DX スキル・ランドスケープとタイムライン - 2026-08-25

Table of Contents

はじめに

本サイトは、ソフトウェア開発と DevOps の領域を中心に据えながら、DX（デジタルトランスフォーメーション） に関連する概念、技術、ツール、プラットフォーム、フレームワークについて、以下の 12 のカテゴリにわたる包括的な概要を提供することを目指しています。

- 01 - ビジネス変革
- 02 - ウェブアプリケーション開発
- 03 - クラウド & クラウドネイティブコンピューティング
- 04 - セキュリティ・プライバシー
- 05 - データサイエンス・エンジニアリング
- 06 - AI・機械学習・LLM
- 07 - 開発者の基礎スキル
- 08 - OS とネットワークの基礎
- 09 - プログラミングの概念とパラダイム
- 10 - 高度なプログラミング
- 11 - 専門開発領域
- 12 - パーソナルスキル

タイムライン

上記のスキルカテゴリに加えて、本サイトでは 1930 年から現在に至るまでの重要な技術イベント、製品リリース、業界の節目を記録した年代別タイムラインを提供しています。各エントリは絵文字によってドメインごとに分類されています。

- 🏢 経営管理、開発方法論、マネジメント
- 🎨 UX、デザイン、認知科学
- 🌐 ウェブフレームワークを含むウェブ技術
- ☁️ クラウド、クラウドネイティブ、コンテナ、DevOps/SRE
- 🔐 セキュリティ、プライバシー
- 🐛 マルウェア、ウイルス、セキュリティインシデント
- 📊 データサイエンス、データベース、データプラットフォーム
- 🧠 AI、機械学習、大規模言語モデル
- 💻 シェル、ターミナル、IDE、開発者の生産性

- ⚙️ システム管理、OS、VM、ネットワークインフラ
- 📜 プログラミングパラダイム、プログラミングの概念、ライブラリ
- 🧩 その他

これらのタイムラインは、スキルセクションに掲載されている技術の変遷をたどるのに役立ち、ソフトウェア開発とデジタルトランスフォーメーションにおける現在のトレンドを理解するための歴史的な文脈を提供します。

IPA DSS-P のカバレッジ

本サイトのスキルリストは、IPA が定める **デジタルスキル標準（DSS-P）v2.0** を可能な限り網羅することを試みています。各セクションページの各サブセクションの下には、最も関連性の高い DSS-P スキル名が記載されています。

IPA (情報処理推進機構) は、経済産業省 (METI) の所管の下にある政策実施機関です。IPA は、**人材育成** (国家 IT 試験の実施)、**情報セキュリティ対策**、そして**デジタルトランスフォーメーション（DX）に関するガイドライン**の策定など、日本の国家 IT 戦略において中心的な役割を担っています。

本ドキュメントで言及する **DSS-P** は、日本企業の DX を加速するために IPA が策定した公開標準です。人材育成や採用のベンチマークとして、多くの日本企業に広く採用されています。

以下の表は、各スキルセクションが主に対応する DSS-P カテゴリを示しています。

セクション・タイトル	主な DSS-P カテゴリ
01 - ビジネス変革	1. ビジネス変革
02 - ウェブアプリケーション開発	3. テクノロジー
03 - クラウド & クラウドネイティブコンピューティング	3. テクノロジー
04 - セキュリティ・プライバシー	4. セキュリティ
05 - データサイエンス・エンジニアリング	2. データ整備・活用
06 - AI・機械学習・LLM	2. データ整備・活用
07 - 開発者の基礎スキル	3. テクノロジー
08 - OS とネットワークの基礎	3. テクノロジー
09 - プログラミングの概念とパラダイム	3. テクノロジー

セクション・タイトル	主な DSS-P カテゴリ
10 - 高度なプログラミング	3. テクノロジー
11 - 専門開発領域	3. テクノロジー
12 - パーソナルスキル	5. パーソナルスキル

原則

本サイトは、以下の原則に基づいて構築されています。

AI 駆動開発と人間の関わり: AI エージェントがソフトウェア開発においてますます詳細なタスクを担うようになるにつれ、要件を正確に言語化する能力が重要になります。これには幅広い分野横断的な知識基盤が求められます — 技術や概念についての理解が広いほど、AI に正しい成果を生み出させるための指示をより効果的に行えるようになります。同時に、デジタルトランスフォーメーションは組織や個人に大きな変化をもたらすため、社会学や心理学といった人間科学の要素も取り入れています。人間の行動を理解することは、変革を推進する上で極めて重要です。

オープン性の優先: プロプライエタリな代替手段よりも、オープンソースソフトウェア (OSS) やオープンフォーマットが優先されます。これにより、俊敏な意思決定を妨げかねない制約やベンダーロックインを最小限に抑えられます。さらに、ソースコードが入手可能であることは、トラブルシューティングに大いに役立ちます。クラウドサービスは、必要不可欠な場合に限り含まれます。

言語非依存: プログラミング言語は、あくまでツールとして扱われます。現代の開発者は複数の言語を容易に扱えるため、多言語によるワークフローは日常的なものになっています。ここでの焦点は、特定の問題領域や文化的文脈に最も適した言語を選択することにあります。

主要な参照元

以下のリソースは、本サイトのコンテンツにおける主要な参照元です。

- Level Up Coding - ソフトウェアエンジニアリング、DevOps、クラウドに関するトピックを扱う Medium 上の刊行物
- ITNEXT - IT、ウェブ開発、DevOps 実践者向けの Medium 上の刊行物
- FAUN - クラウドネイティブ、DevOps、開発者向けコンテンツに特化した Medium 上の刊行物
- Thoughtworks Technology Radar - 採用・試行・回避すべき技術、ツール、プラットフォーム、言語をまとめたガイド
- Developer Roadmaps - さまざまなエンジニアリング職種向けの、コミュニティ主導によるロードマップと学習パス

- Golang Weekly - Go 言語のニュース、記事、プロジェクトを扱う週刊ニュースレター
- Ruby Weekly - Ruby 言語のニュース、記事、プロジェクトを扱う週刊ニュースレター
- Postgres Weekly - PostgreSQL のニュース、記事、ツールを扱う週刊ニュースレター
- Tony Lixu（Medium） - クラウドインフラ、Kubernetes、DevOps エンジニアリングに関する記事

01 - ビジネス変革

プロジェクトマネジメント

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.3 変革活動のマネジメント > プログラム／プロジェクトマネジメント
> - 1. ビジネス変革 > 1.2 プロダクトのマネジメント > プロダクトスコープと優先順位のマネジメント

プロジェクトプランニング＆見積もり

- Project management - チームの作業を指導してプロジェクトのすべての目標を与えられた制約条件内で達成するプロセスです
  - PMBOK (Project Management Body of Knowledge) - プロジェクトマネジメントの標準的な用語とガイドラインのセットです
  - Project charter - プロジェクトのスコープ、目的および参加者を述べたステートメントです
  - Critical chain project management - プロジェクトタスクを実行するために必要なリソース（人、機器、物理的スペース）に重点を置いたプロジェクト計画および管理方法です
  - Program evaluation and review technique (PERT) - プロジェクト完成に関わるタスクを分析して表現するためにプロジェクトマネジメントで使用される統計的ツールです
  - Work breakdown structure - プロジェクトをより小さな構成要素に分解した成果物指向の分解です
  - RACI matrix - プロジェクトまたはビジネスプロセスのタスクまたは成果物の完成に関わるさまざまな役割の参加を説明した責任割り当てマトリックスです
    - Responsible、Accountable、Consulted、Informed
  - Software development effort estimation - 不完全、不確実、ノイズの多い入力に基づいてソフトウェアを開発または保守するために必要な最も現実的なエフォート

量（人時または金額で表現）を予測するプロセスです

- Hofstadter's law - 複雑なタスク完成に必要な時間を正確に見積もることの難しさについての自己言及的な警句です
- Three-point estimation - プロジェクトマネジメントで活動の予想期間またはコストを見積もるために使用される技術です
- Planning poker - アジャイル原則でのタイムボックスに主に使用されるコンセンサスベースのゲーム化された見積もり技術です
- Fermi problem - 次元解析や極端な科学計算の近似を教えるために設計された、物理学または工学教育における見積もり問題です

- Systems development life cycle (SDLC) - 情報システム開発プロジェクトに関わるステージを説明するプロジェクトマネジメントで使用される概念モデルです

優先順位付け

- Prioritization - 項目または活動を緊急性の順序で配置する活動です
  - RICE - リーチ、インパクト、確信度、エフォートを意味するプロダクト優先順位付けのシンプルなスコアリングシステムです
  - Kano model - 1980 年代に狩野紀昭によって開発されたプロダクト開発と顧客満足度の理論です
  - MoSCoW method - マネジメント、ビジネス分析、プロジェクトマネジメント、ソフトウェア開発で使用される優先順位付けテクニックです

作業・課題追跡

- 課題追跡＆タスク管理ボード
  - Jira - チームがプランニング、割り当て、追跡、レポート、作業管理を行うのに役立つ課題追跡とプロジェクトマネジメント用のソフトウェアアプリケーションです
    - JiraCLI - Atlassian Jira の対話的なコマンドラインツールで、ある程度 Jira UI を避けるのに役立ちます
  - Linear - 課題、サイクル、プロダクトロードマップを備えたプロダクト開発向けの目的志向のツールです
  - Fizzy - バグ、課題、アイデア、小さなプロジェクトなど何かを追跡するための現代的なカンバン式ツールです
  - GitLab Issue Board - ワークフローステータスに対応する列に課題を表示するユーザーインターフェースです
  - GitLab Service Desk - ユーザーに GitLab アカウントを必須としないメール経由でユーザーに接続できる機能です

    - Azure Boards - チームが効果的に協力し、ワークフローを合理化できるようにする作業項目管理用のカスタマイズ可能なプラットフォームを提供するサービスです
    - GitHub Issues - GitHub で作業管理に役立つ追跡ツールです
    - Redmine - 無料でオープンソースの Web ベースのプロジェクトマネジメントおよび課題追跡ツールです
    - OpenProject - クラシック、アジャイル、またはハイブリッドプロジェクトマネジメント向けのオープンソースプロジェクトマネジメントソフトウェアです
- 協調的な作業管理
    - Asana - チームが作業を調整しプロジェクトを進め続けるのに役立つ人間と AI 協調向けのプラットフォームです
    - monday.com - 人とエージェントが部門や使用例全体で結果を推進する安全な作業プラットフォームです
    - ClickUp - アプリ、AI、プロジェクト、チャットをまとめてすべてのソフトウェアに置き換えるオールインワン生産性プラットフォームです
    - Trello - ボード、リスト、カードを使用してタスクを整理し、プロジェクトに関する共有の視点を作成するビジュアル協調ツールです
    - Airtable - スプレッドシートの柔軟性とデータベースの力を組み合わせて、チームが作業管理を行うのに役立つプラットフォームです

チームコラボレーション

- メッセージング＆会議
    - Slack - 会話、ツール、ファイルを 1 か所にまとめたクラウドベースのチーム協調プラットフォームです
    - Microsoft Teams - 職場チャット、会議、ファイルストレージ、アプリケーション統合を組み合わせたコラボレーションプラットフォームです
    - Discord - コミュニティ、友人グループ、ビジネスが接続を保つために使用する音声、ビデオ、テキスト通信サービスです
    - Mattermost - 開発者向けのオープンソースコラボレーションプラットフォーム。セキュアメッセージング、プロジェクトマネジメント、ワークフロー自動化を提供します
    - Zoom - ビデオ会議、音声通話、ウェビナー、チャットを提供するビデオ通信プラットフォームです

プロセス成熟度＆品質基準

- CMMI (Capability Maturity Model Integration) - プロセスレベルの改善トレーニングと評価プログラムです

- ISO 9001 (Quality management systems) - 最も確立された品質フレームワークで、標準を満たすことを希望する組織が満たす必要のある品質管理システムの要件を定義しています
- ISO/IEC 12207 (Software life cycle processes) - ソフトウェアシステムの開発と保守に必要なすべてのプロセスを定義するソフトウェアライフサイクルプロセスの国際標準です
- ISO/IEC 15288 (System life cycle processes) - プロセスとライフサイクルステージをカバーするシステムエンジニアリングの技術標準です
- ISO/IEC 15504 (Process assessment) - コンピュータソフトウェア開発プロセスおよび関連するビジネス管理機能の技術標準ドキュメントのセットです

プロダクトマネジメント

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.2 プロダクトのマネジメント > 要求の分析とマネジメント
> - 1. ビジネス変革 > 1.2 プロダクトのマネジメント > プロダクトビジョン／ロードマップ策定
> - 1. ビジネス変革 > 1.2 プロダクトのマネジメント > マーケティング
> - 1. ビジネス変革 > 1.3 変革活動のマネジメント > プロダクトライフサイクルマネジメント

プロダクト戦略＆ロードマップ

- Product management - プロダクトまたはサービスの計画、開発、発売、管理のビジネスプロセスです
- Lean startup - プロダクト開発サイクルを短縮し、提案されたビジネスモデルが実行可能かどうかを素早く発見することを目的とした企業とプロダクト開発の方法論です
- Jobs to Be Done (JTBD) - 人と組織を意思決定に向かわせたり遠ざかったりする状況を明かすレンズです
- Crowdfunding - 通常はインターネット経由で多くの人からお金を集めてプロジェクトまたはベンチャーに資金を提供する実践です
- Business model - 経済的、社会的、文化的、またはその他のコンテキストにおいて、組織が価値をどのように創造、提供、および捕捉するかの根拠です
  - Direct-to-consumer - 製品を顧客に直接販売し、第三者の小売業者、卸売業者、または仲介者をバイパスするビジネスモデルです
  - Subscription business model - 顧客がプロダクトまたはサービスへのアクセスの

ために定期的な間隔で定期的な価格を支払う必要があるビジネスモデルです

- Business model canvas - 新しい、または既存のビジネスモデルを開発または文書化するための戦略的マネジメントテンプレートです
- Lean Canvas - 起業家がビジネスアイデアを素早く概説するための 1 ページのビジネスモデリングツールです

- Technology roadmap - 短期および長期の目標を特定の技術ソリューションと一致させることで、戦略的および長期計画をサポートする柔軟な計画スケジュールです
  - Aha! - チームが顧客が愛するプロダクトを構築およびマーケティングするのに役立つプロダクト開発ソフトウェアのスイートです

要求分析

- Business analysis - ビジネスニーズを特定し、ビジネス問題への解決策を決定することに焦点を当てた専門分野です
- Requirements analysis - さまざまなステークホルダーの競合する可能性のある要件を考慮して、新規または変更されたプロダクトまたはプロジェクトを満たす必要性または条件を決定するプロセスです
  - Requirement - プロダクトまたはサービスが何であるか、または何をすべきかの文書化されたニーズです
    - Non-functional requirement - 特定の動作ではなく、システムの操作を判断するために使用できる基準を指定する要件です
- 関連標準
  - ISO/IEC 25010 (Systems and software Quality Requirements and Evaluation) - システムおよびソフトウェア品質モデルを定義するシステムおよびソフトウェア品質要件と評価の国際標準です

マーケティング＆カスタマー体験

- Marketing - 通常、さまざまなステークホルダー向けの価値を持つ提供品を作成、通信、配信、交換することを含む、顧客を獲得、満足させ、保持するプロセスです
  - Market research - ターゲット市場と顧客に関する情報を収集するための組織化された取り組みです
  - SEO - 検索エンジンから Web サイトまたは Web ページへの Web サイトトラフィックの品質と量を改善するプロセスです
    - Google Search Central - ユーザーが Google 検索でサイトを見つけるのに役立つために必要なすべてのホームです
  - Marketing mix - ビジネスの基盤モデル。歴史的にはプロダクト、価格、場所、プロモーションを中心としています

    - Fear of missing out (FOMO) - 自分の人生をより良くする可能性のある情報、イベント、経験、または人生の決定について情報を得ていない、または逃している可能性があるという不安感です
    - Fear, uncertainty, and doubt (FUD) - 販売、マーケティング、公共関係、政治、投票、カルトで使用される操作的なプロパガンダ戦術です
    - Mere-exposure effect - 人が、単に慣れ親しんでいるという理由だけで物事に対する好みや嫌悪を抱く傾向がある心理現象です
    - 広告
        - インジケーター
            - Click through rate - 特定のリンクをクリックするユーザー数と、ページ、メール、または広告を表示するユーザー総数の比率です
            - Conversion rate - 望ましいアクションを実行するユーザーの割合です
    - コンセプトとフレームワーク
        - Brand - 1 つの売り手の商品またはサービスを他の売り手の商品またはサービスと区別する名前、用語、デザイン、シンボル、またはその他の特性です
        - Customer experience - プロダクトまたはサービスとの相互作用のすべてのステージ中の顧客の認知的、感情的、感覚的、行動的応答です
        - Customer service - その製品またはサービスを購入または使用する者に対して、企業が対面またはリモートで提供する支援およびアドバイスです
        - Value chain - 最終顧客に価値がある商品およびサービスを提供するためにビジネスまたは企業が実行する活動の進行です
    - 戦略向けツール
        - Value proposition canvas - 顧客が望むプロダクトとサービスを作成するのに役立つツールです

マーケティング技術

- タグマネジメント
    - Google Tag Manager - Web サイトまたはモバイルアプリ上の測定コードと関連コードフラグメント（タグと呼ばれます）を素早く簡単に更新できるタグマネジメントシステムです
- Web 分析
    - Google Analytics - Web サイトおよびアプリパフォーマンスをより深く理解することを求める数百万の Web サイトおよびアプリ所有者向けのプラットフォームです
    - Plausible - 直感的で、軽量で、オープンソースの Web 分析です

    - Umami - Google Analytics に対する、シンプル、高速、プライバシーに焦点を当てた代替案です
    - Ackee - プライバシーに関心のある人のための、自己ホストされたプライバシーに焦点を当てた分析ツールです
- カスタマーデータプラットフォーム
    - Customer data platform - さまざまなタッチポイントからカスタマーデータを集約および整理して統一されたカスタマープロファイルを構築するソフトウェアシステムです
    - Twilio Segment - ビジネスが複数のソースとデスティネーションからカスタマーデータを収集、統一、アクティベートしてパーソナライズされた体験を作成できるカスタマーデータプラットフォームです
- マーケティング自動化
    - Marketing automation - マーケティング部門と組織が反復的なタスクを自動化するために設計されたソフトウェアプラットフォームとテクノロジーです
    - Klaviyo - アメリカのマーケティング自動化プラットフォームおよびメールマーケティングサービスです
- 広告プラットフォーム
    - Google Ads - 広告主が入札して、短い広告、サービス提供、プロダクトリスティング、またはビデオを Web ユーザーに表示するオンライン広告プラットフォームです
    - Google AdSense - Google が実行するプログラムで、Google コンテンツサイトネットワーク内の Web サイトパブリッシャーは、サイトコンテンツとオーディエンスをターゲットとした、テキスト、画像、ビデオ、またはインタラクティブメディア広告を提供します
- 実験＆最適化
    - Optimizely - デジタル体験プラットフォーム（DXP）。スケーラビリティとセキュリティを提供し、ビジネスを未来に進めるのに役立つ単一の統合プラットフォームです
- メールマーケティング＆配信
    - SendGrid - スケールで信頼できるトランザクションおよびマーケティングメール配信を提供するクラウドベースのメール配信プラットフォームです
    - Mailchimp - 小規模ビジネス向けのオールインワンマーケティングプラットフォーム。クライアント、顧客、オーディエンスとのメールマーケティングの管理と通信に役立ちます
    - listmonk - 自己ホストされたニュースレターとメーリングリスト管理です

    - BillionMail - 完全に自己ホストされ、開発者向けなオープンソースメールサーバーおよびメールマーケティングソリューションです

メトリクス＆パフォーマンス

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.2 プロダクトのマネジメント > プロダクト成果指標の設計と運用
> - 1. ビジネス変革 > 1.1 戦略の理解とアーキテクチャ設計 > ビジネス価値定義／投資対効果の試算と意思決定支援

ゴール設定フレームワーク

- Goal setting - 人またはグループをゴールに向かって動機付け導くために設計されたアクションプランを開発するプロセスです
    - SMART goals - 例えば、プロジェクトマネジメント、従業員パフォーマンスマネジメント、個人開発におけるゴール設定を指導するために使用される頭字語です
        - Specific: 改善のための特定の領域をターゲット
        - Measurable: 定量化、または少なくとも進捗インジケーターを提案する
        - Assignable: 責任を明確に定義する
        - Realistic: 利用可能なリソースで達成可能な結果の概要を示す
        - Time-related: 予想される結果のためのタイムラインを含める
    - FAST goals - 頻繁に議論され、スコープが野心的で、メトリクスが具体的で、すべての人が見えるゴールのフレームワークです
    - GROW model - ゴール設定と問題解決のためのシンプルな方法です
    - OKRs - 個人、チーム、組織が測定可能なゴールを定義し、その結果を追跡するために使用するゴール設定フレームワークです
    - KPIs - 組織または特定の活動（プロジェクト、プログラム、プロダクト、その他のイニシアティブなど）の成功を評価するために使用されるパフォーマンス測定の一種です
    - Goodhart’s law - しばしば「測定がターゲットになると、それはもう良い測定ではなくなる」と述べられている警句です

プロダクト＆カスタマーメトリクス

- Net Promoter Score - 企業、プロダクト、またはサービスを友人または同僚に推薦する可能性をレーティングするよう回答者に求める単一の調査質問に基づいた市場調査メトリクスです

- Rubric - レスポンスの品質を評価するために使用されるスコアリングツールです

エンジニアリングパフォーマンスメトリクス

- SPACE framework - 満足度と幸福度、パフォーマンス、アクティビティ、コミュニケーションとコラボレーション、効率とフローを含む、より総合的な方法で開発者生産性について考える方法を提供するフレームワークです
- The Four Keys of DORA - デプロイメント頻度、変更リードタイム、変更失敗率、サービス復旧時間で構成される DevOps パフォーマンスを測定するために使用されるメトリクスのセットです

エンタープライズ戦略＆アーキテクチャ

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.1 戦略の理解とアーキテクチャ設計 > ビジネス環境と経営戦略の理解
> - 1. ビジネス変革 > 1.1 戦略の理解とアーキテクチャ設計 > ビジネスとエンタープライズのアーキテクチャ設計
> - 1. ビジネス変革 > 1.3 変革活動のマネジメント > アーキテクチャマネジメント＆ガバナンス
> - 1. ビジネス変革 > 1.3 変革活動のマネジメント > チェンジマネジメント
> - 1. ビジネス変革 > 1.3 変革活動のマネジメント > リスク＆コンプライアンス

エンタープライズアーキテクチャ

- Enterprise architecture - 戦略の成功した開発と実行のためのすべての時間における包括的なアプローチを使用した、エンタープライズ分析、設計、計画、実装の実施のための実証済みの実践です
  - TOGAF standard - 世界のリーディング組織がビジネス効率を改善するために使用する実証済みのエンタープライズアーキテクチャ方法論およびフレームワークです
  - Zachman Framework - オブジェクトの構造化されたエッセンシャルコンポーネントの存在の理論である本体論です
  - ArchiMate - 異なるツール納入業者およびコンサルティング企業にサポートされたエンタープライズアーキテクチャのためのオープンで独立したモデリング言語です
    - Archi - ArchiMate モデルを作成するための無料でオープンソース、クロスプラットフォームツールおよびエディターです
- Enterprise modeling - プロセスモデル、データモデル、リソースモデルおよび/または新しい本体論を使用したエンタープライズの全体または一部のモデル構築プロセスです

    - BPMN - ビジネスプロセスダイアグラムでビジネスプロセスを指定するためのグラフィカル記法。ビジネスユーザーに理解可能であり、技術ユーザーの複雑なプロセスセマンティクスを表現しています
    - SysML - ハードウェア、ソフトウェア、情報、人員、手順、施設を含める可能性のある複雑なシステムを指定、分析、設計、検証するための汎用グラフィカルモデリング言語です
    - Eclipse Capella - フィールドで実証された言語と方法を活用して複雑なシステムのアーキテクチャを正常に設計する強力で拡張可能な MBSE ソフトウェアツールです

エンタープライズアプリケーション

- Enterprise resource planning - 多くの場合リアルタイムで、ソフトウェアおよびテクノロジーで仲介されたメインビジネスプロセスの統合管理です
    - SAP ERP - プロセスを合理化し、生産性を向上させ、組織全体にわたってリアルタイムの洞察を提供する包括的なソフトウェアシステムです
    - Odoo - CRM、ERP、会計などの領域をカバーするオープンソースビジネスアプリケーションのスイートです
    - ERPNext - 100% オープンソースの ERP。現代的で、包括的で、ユーザーフレンドリーなエンタープライズリソースプランニングソリューションです
- Customer relationship management - 組織が顧客との相互作用を管理、分析、改善するために使用する戦略的なプロセスです
    - EspoCRM - すべての企業関係を管理および評価するためのオープンソース Web アプリケーションです
    - HubSpot - 企業が共有データベースにマーケティング、営業、サービスツールを接続して成長するのに役立つ顧客プラットフォームです
    - Salesforce - 企業と顧客を一緒にもたらし、すべての部門のための統合 CRM プラットフォームを提供する顧客関係管理ソリューションです
    - Zendesk - 企業がマルチチャネルサポートを通じてより良い顧客関係を構築するのに役立つカスタマーサービスソフトウェアおよびサポートチケットシステムです
    - Atlas - ツールとワークフローに合わせた迅速、正確、測定可能なサポートを提供するカスタマーサポート向けのカスタム AI です
    - SuiteCRM - 顧客の 360 度ビューとビジネスを提供するオープンソースおよび無料のカスタマー関係管理（CRM）ソフトウェアソリューションです
- Supply chain management - ビジネスと場所の間における商品とサービスの流れの管理。原材料、進行中の在庫、完成品の移動と保管が含まれます。発生地点から消費地点までです

- Human resource management - 会社または組織の人の効果的で効率的な管理への戦略的かつ首尾一貫したアプローチ。ビジネスが競争優位性を得るのに役立ちます
  - Competence - ジョブの効率またはパフォーマンスを有効にして改善する実証可能な特性とスキルのセットです
- Contract management - 契約作成、実行、分析を体系的かつ効率的に管理し、財務および運用パフォーマンスを最大化し、リスクを最小化するプロセスです
- E-commerce - オンラインサービスまたはインターネット上でプロダクトを電子的に売買する活動です
  - Shopify - オンラインストアと小売ポイントオブセールシステムのための独自の e コマースプラットフォームを提供するカナダの多国籍 e コマース企業です
  - Stripe - e コマース Web サイトとモバイルアプリケーション向けの支払い処理ソフトウェアと API を提供するビジネス向けの金融インフラストラクチャプラットフォームです

組織の変革管理

- Organizational structure - 特定の活動がどのように指向され、組織のゴールを達成するのかについて概説するシステムです
- Diffusion of innovations - 新しいアイデアや技術がどのように、なぜ、どのくらいの速さで広まるのかを説明しようとする理論です
- Kotter’s 8-step change model - 組織が変革を効果的に実装し維持するのに役立つように設計されたツールと戦略のセットです
- Prosci ADKAR Model - 個人および組織の変革をガイドするゴール指向の変革管理モデルです

戦略分析＆ビジネスインテリジェンス

- 戦略分析フレームワーク
  - MECE principle - 項目のセットを相互排他的（ME）かつ集合的に網羅的（CE）なサブセットに分離するグループ化原則です
  - SWOT analysis - 組織またはプロジェクトの強み、弱み、機会、脅威を特定する戦略計画および管理に使用される意思決定技術です
  - PEST analysis - 戦略管理および市場調査で使用される外部マクロ環境要因（政治的、経済的、社会的、技術的）のフレームワークです
  - Porter’s five forces analysis - 業界組織経済学にルーツを置く企業の競争環境を分析し、競争強度と業界の魅力を決定する 5 つの力を特定する方法です
- ビジネスインテリジェンスプラットフォーム

- Tableau - 人々が問題を解決するためにデータを見て、理解し、行動するのに役立つビジュアル分析プラットフォームです
- Metabase - データベースのクエリと可視化層。スタートアップの本番 DB から大規模なデータウェアハウスまで対応します
- Power BI - セルフサービスおよびエンタープライズビジネスインテリジェンスの統合されたスケーラブルなプラットフォームです
  - DAX - 計算列、測定、カスタムテーブルを作成するために Microsoft Power BI 全体で使用されるプログラミング言語です
- Looker Studio - 旧 Google Data Studio として知られる、データをカスタマイズ可能で情報豊富なレポートとダッシュボードに変えるビジネスインテリジェンス及びデータ可視化プラットフォームです
- Exploratory Desktop - さまざまなデータソースへのアクセス、データのクレンジングと変換、データの可視化と分析をシンプルで使いやすい UI で実現します

リスクマネジメント

- Risk management - リスクの特定、評価、優先順位付けの後、不運なイベントの発生確率または影響を最小化し、監視、制御し、または機会実現を最大化するために調整された経済的リソース適用です
  - Business continuity planning - 自然災害またはサイバー攻撃などの潜在的な脅威からの防止と復旧システムを作成するために組織が実施するプロセスです
    - IT disaster recovery - 自然災害、サイバー攻撃、機器障害などの破壊的なイベント後に通常の IT 操作を再開するプロセスです
    - ISO 22301 (Business continuity management systems) - 破壊的なインシデントから保護、対応、復旧するための要件を指定するビジネス継続管理システムの国際標準です
  - Project risk management - プロジェクトライフサイクル全体に発生するリスクを特定、分析、対応し、プロジェクトが進行中で目標を達成するのに役立つプロセスです
  - Financial risk management - 主に信用リスクと市場リスク、および運用リスクのいくつかの側面を管理することにより、企業の経済的価値を保護する実践です
  - ISO 31000 (Risk management) - リスク評価と管理のために一貫した語彙と方法論を提供するリスク管理の国際標準のセットです

IT サービス管理

- IT service management - 顧客に提供される IT サービスを設計、構築、配信、操作、制御する組織が実行するアクティビティです

- ITIL (Information Technology Infrastructure Library) - IT サービス管理の詳細な実践のセット。IT サービスをビジネスニーズに合致させることに焦点を当てています
- ISO/IEC 20000 (Service management) - サービス管理システムを確立、実装、維持、継続的に改善するための要件を指定する IT サービス管理の国際標準です
- ServiceNow - エンタープライズ全体で人、機能、システムを接続するデジタルワークフロー向けのクラウドベース、AI 駆動プラットフォームです

ナレッジ＆コンテンツマネジメント

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.3 変革活動のマネジメント > ビジネスモデリングとコラボレーション

パーソナルナレッジマネジメント

- Zettelkasten - 研究、研究、執筆のためのメモ取りおよびパーソナルナレッジマネジメントシステム。スリップまたはカード上に保存された相互接続された小さな情報項目で構成されています
- ノート取得ツール
  - Obsidian - ユーザーがプラグインとテーマを使用してカスタマイズ可能なインターフェースを通じてデバイスにメモを保存し、アイデアを接続し、ナレッジを整理できるプライベートな思考のための無料で柔軟なアプリケーションです
  - Memos - クイックキャプチャ用に構築されたオープンソース、自己ホストされたメモ取得ツール。マークダウン-ネイティブで軽量です
- フィード集約
  - FreshRSS - RSS および Atom フィード用の無料で自己ホスト可能なアグリゲーターです

協調的なワークスペース＆ Wiki

- Wiki software - ユーザーが Web ブラウザー経由でページまたはエントリを作成および協調編集できるコラボレーティブソフトウェアです
  - MediaWiki - 無料でオープンソースの Wiki ソフトウェアです
  - Ibis - ActivityPub プロトコルを使用するフェデレーション百科事典です。Mastodon や Lemmy と同様です
  - LeafWiki - 重いスタックなしの実際の Wiki アプリ。ディスク上のマークダウンファイルを保存します。データベースサーバーは不要です

- チームワークスペース
  - Notion - メモ、ドキュメント、Wiki、プロジェクト、コラボレーションのオールインワンワークスペース。ナレッジ管理をタスクおよびプロジェクト追跡と組み合わせます
  - Coda - チームとツールをまとめてより整理された作業日を行うオールインワンコラボレーティブワークスペースです
  - Confluence - すべてのアイデア、ドキュメント、ナレッジ、チームメイト向けの 1 つの場所です
  - Outline - チームがドキュメントを整理し、リアルタイムで協力し、AI 駆動の質問応答でワークスペース全体を検索するのに役立つナレッジベースプラットフォームです

コンテンツ管理システム

- Content management system - デジタルコンテンツの作成と変更を管理するために使用されるコンピュータソフトウェアです
  - WordPress - ハイパーテキストプリプロセッサ言語で記述された、MySQL または MariaDB データベースでサポートされた HTTPS が無料でオープンソースのコンテンツ管理システムです
  - Drupal - PHP で記述され、GNU General Public License の下で配布された無料でオープンソースの Web コンテンツ管理システムです

コンテンツコラボレーション＆ファイルシンク

- SharePoint - 組織がコンテンツを安全に保存、共有、管理し、内部および外部ユーザーと協力できるようにする Web ベースのコラボレーションおよびドキュメント管理プラットフォームです
- Nextcloud - 業界をリードしている、完全にオープンソース、オンプレミスのコンテンツコラボレーションプラットフォームです
- Box - 組織が内部および外部ユーザーと協力しながら、コンテンツを安全に管理および共有できるエンタープライズクラウドコンテンツ管理プラットフォームです
- Dropbox - クラウドストレージ、ファイル同期、パーソナルクラウド、クライアントソフトウェアを提供するファイルホスティングサービスです

エンタープライズ AI ＆生産性

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.1 データ・AIの戦略的活用 > データ・AI理解・活用

> • 2. データ整備・活用 > 2.1 データ・AIの戦略的活用 > データ・AI活用戦略設計

自律型 AI ワーカー

- Claude Cowork - ファイル、ドキュメント、Web アプリケーション全体で複数ステップのタスクを自律的に計画および実行するナレッジワーク向けのエージェント型 AI システムです。コンピューター上で直接
- claw-empire - CLI、OAuth、API 接続エージェントを仮想自律企業として調整するローカルファーストの AI エージェントオフィスシミュレーターです
- Sistava - 構造化スプリント、OKRs、KPI を通じて協力する AI ワーカーのチームを管理するための AI 従業員プラットフォーム。永続的なメモリと実際のツールアクセスを備えています

エンタープライズ AI アシスタント

- Amazon Q Business - 企業が情報を見つけ、洞察を得られるようにする生成 AI 駆動のアシスタント。仕事で行動します。会社のデータとアプリケーションと統合します
- Microsoft 365 Copilot - 生産性と創造性を増幅し、ビジネスプロセスを再設計し、AI 駆動の組織にビジネスを変換するのに役立つ仕事向けの AI アシスタントです。安全に
- Notion AI - ワークスペース用の統合 AI アシスタント。ライティング支援、ワークスペース Q＆A、タスク自動化向けの自律エージェントを提供します
- Gemini for Google Workspace - Docs、Gmail、Sheets、Slides 向けの AI 駆動のアシスタント。プラットフォーム全体で作業を書く、視覚化、整理するのに役立ちます
- Claude for Enterprise - 管理者制御、シングルサインオン（SSO）、Claude の最新モデルへのロールベースアクセスを備えた、組織が AI を使用するための安全でスケーラブルな方法です
- ChatGPT Enterprise - より高速なアクセス制限なし GPT-4、より長いコンテキストウィンドウ、高度なデータ分析機能を備えたエンタープライズグレードの AI アシスタントです
- Glean - 企業向けの AI 駆動の検索およびアシスタント。会社のすべてのアプリとデータに接続して、必要なものを正確に見つけます

自己ホストされた AI プラットフォーム

- AnythingLLM - 皆向けのオールインワン AI アプリケーションです
- Dify - オープンソース LLM アプリ開発プラットフォームです
- Flowise - AI エージェント、LLM オーケストレーション、さらに構築するためのオープンソース生成 AI 開発プラットフォームです

- LibreChat - すべての AI 会話を統一されたカスタマイズ可能なインターフェースにまとめるオープンソース AI プラットフォームです
- OpenWebUI - 完全にオフラインで操作するように設計された拡張可能で機能豊富で、ユーザーフレンドリーな自己ホストされた AI プラットフォームです

AI コンテンツ＆ドキュメント生成

- Beautiful.ai - エンタープライズチーム向けに構築された AI プレゼンテーションプラットフォーム。すべてのデッキをブランド内に保ち、企業全体で共有可能です
- Gamma - AI で駆動される、アイデア提示向けの新しい媒体です
- Napkin AI - ビジネスストーリーテリング向けのビジュアル AI。テキストをビジュアルに変換して、アイデア共有を迅速かつ効果的にします
- NotebookLM - AI 研究ツールおよび思考パートナー。ソースを分析し、複雑さを明確にし、コンテンツを変換できます

AI エージェントレジストリ

- エージェントスキルレジストリ
  - The Agent Skills Directory - AI エージェント向けの再利用可能な機能を提供するオープンエージェントスキルエコシステムです
  - Anthropic Agent Skills - 特殊なタスクと反復可能なワークフローを有効にする指示、スクリプト、リソースを含む Claude のスキル実装を含む公開リポジトリです
  - SkillsMP (Skills Management Platform) - オープン SKILL.md 標準に基づいてモジュール AI エージェント機能を検出および共有するためのコミュニティ駆動型マーケットプレイスです
- MCP（モデルコンテキストプロトコル）レジストリ
  - Official MCP Registry - モデルコンテキストプロトコル組織によって保守される公式参照 MCP サーバー実装の集まりです
  - MCP Registry - 公開 MCP サーバーの検索可能な Web ディレクトリです
  - Claude Plugins - ワンクリックインストール向けのツール、スキル、統合をバンドルするディレクトリです

ヒューマン・センタード・デザイン

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.4 デザイン > 顧客・ユーザー/ステークホルダー理解
> - 1. ビジネス変革 > 1.4 デザイン > デジタルプロダクト設計

コア原則＆ユーザーエクスペリエンス（UX）

- User experience - 特定のプロダクト、システム、またはサービスを使用することに関する人物の感情と態度です
- Usability - 指定されたユーザーが指定されたコンテキストで指定されたゴールを効果性、効率性、満足度で達成するために、プロダクトを使用できる程度です
- User interface design - デザイナーがユーザーエクスペリエンスを作成する上で重要な機能を実行する工芸です
- Design thinking - デザインコンセプトが開発される認知的、戦略的、実践的プロセスのセットです
- Accessibility - 障害のある人向けのプロダクト、デバイス、サービス、環境の設計です
  - Accessibility Object Model (AOM) - 開発者が HTML ページのアクセシビリティツリーを変更（および最終的に探索）できるようにする JavaScript API です
  - WAI-ARIA - アクセス可能なリッチインターネットアプリケーション Web 標準スイートです
- プロトタイピング
  - Paper prototyping - ユーザー中心の設計プロセスで広く使用される方法。開発者がユーザーの期待とニーズを満たすソフトウェアを作成するのに役立つプロセスです
  - Website wireframe - Web ページのスケルトン輪郭です

ユーザーリサーチ＆アイディエーション

- リサーチ方法論
  - Extreme users - 使用スペクトラムの極端なユーザーを研究して、エッジケースと典型的なユーザーの両方に役立つデザインソリューションを通知することに焦点を当てたユーザー中心のデザイン方法論です
  - Card sorting - サイトの情報アーキテクチャを設計または評価するのに役立つ方法です
  - A/B testing - 例えば、テスト対象者の変種 A に対する変種 B への反応を比較し、変種の方がより効果的であるかを決定することにより、単一の変数の複数のバージョンを比較する方法です
  - Diary studies - 人々が時間をかけて経験と活動を記録する研究方法です
  - Persona - ユーザータイプの関係を表すために作成されたフィクショナルキャラクターです
- アイディエーションテクニック

- Affinity diagram - アイデアとデータを整理するために使用されるビジネスツールです
- Brainstorming - 特定の問題の結論を見つけるため、メンバーによって自発的に提供されるアイデアのリストを集めることにより、努力がなされるグループ創造性テクニックです
- SCAMPER - 学生が枠外に考えるのを支援し、ナレッジを向上させるための構造化された方法です
- How Might We (HMW) - デザイナーが問題ステートメントを再フレーム化および開くことができるデザイン思考方法。効率的で焦点を当てた、革新的なアイディエーションセッションが可能になり、デザインチャレンジの解決に役立ちます

認知＆行動心理学

- 心理学的モデル
  - Seven stages of action - ゴール達成のためにとる個人の認知的および物理的ステップの理想化された説明です
    - 1: ターゲット形成
    - 2: 意図の形成
    - 3: アクション指定
    - 4: アクション実行
    - 5: 世界の状態の認識
    - 6: 世界の状態の解釈
    - 7: 結果の評価
- 認知プロセス
  - Attention - 他の事柄を無視しながら、環境の 1 つの側面に選択的に集中する認知プロセスです
  - Metacognition - 自分自身の思考プロセスの認識および その背後のパターンの理解です
- インタラクション原則＆法則
  - Principle of least astonishment - 操作の実行結果が、操作の名前および他のコンテキストに基づいて、明白、一貫性、予測可能である必要があることを述べる一般的な原則です
  - Affordance - オブジェクトの使用方法を示すオブジェクトの特性です
  - Stroop effect - タスクの反応時間での干渉のデモンストレーションです
  - Fitts’s law - 人間とコンピューター相互作用とエルゴノミックスで主に使用される人間の動きの予測モデルです

ビジュアル設計＆タイポグラフィ

- タイポグラフィ
  - Typography - 表示された書き言葉を読みやすく、読みやすく、魅力的にするためのタイプの配置の芸術と技術です
  - Web Typography - ワールドワイドウェブでのフォントの使用です
  - Microsoft Typography - フォントテクノロジーとタイプフェイス、技術仕様、開発者ツール、Microsoft プロダクト向けの設計ガイドラインの包括的なリソースです
- ビジュアルファンデーション
  - Color space - 色の特定の組織です
    - ICC profile - 色入力または出力デバイス、または色スペースを特性付ける一連のデータです
      - sRGB - HP と Microsoft が 1996 年に協調して、モニター、プリンター、インターネットで使用するために作成された標準 RGB 色スペースです
      - HSL and HSV - RGB カラーモデルの 2 つの最も一般的な円筒座標表現です
  - Lucide - さまざまなプラットフォームとフレームワーク向けの美しく一貫したアイコンライブラリです
- フォントレンダリング＆テクノロジー
  - フォント標準
    - TrueType - 1980 年代後半に Apple と Microsoft によって開発され、PostScript で使用される Adobe の Type 1 フォントの競争相手として開発されたアウトラインフォント標準です
    - OpenType - TrueType フォント形式の拡張として Microsoft と Adobe によって開発されたスケーラブルなコンピュータフォント形式。高度なタイポグラフィ機能とマルチプラットフォーム互換性をサポートします
    - WOFF (Web Open Font Format) - Mozilla とその他によって開発された Web ページで使用するフォント形式。TrueType および OpenType フォント向けの圧縮ラッパーを提供し、Web パフォーマンスを向上させます
    - Variable Fonts - OpenType フォント仕様の進化。単一のフォントファイルが重量、幅、その他の軸で変動を定義することで、複数のフォントのように動作できます
  - オープンフォント
    - Noto Fonts - すべての現代および古代言語用のグローバルフォントコレクションです

        - Orbitron - ディスプレイ目的のために意図されたジオメトリックサンセリフタイプフェースです
    - ライブラリ＆エンジン
        - FreeType - フォントをレンダリングするために利用可能なソフトウェアライブラリです
        - HarfBuzz - 広く使用されるオープンソーステキストシェーピングエンジン。Unicode テキストを、さまざまなスクリプトと言語全体で正しいレンダリングに必要なグリフとポジションに変換します
        - Pango - テキストのレイアウトおよびレンダリング向けのオープンソースライブラリ。国際化と複雑なスクリプトのサポートが重視されます
        - Fontconfig - フォントアクセスを設定およびカスタマイズするためのライブラリ。主に Linux およびその他の Unix 類似システムで使用され、一貫したフォントマッチングおよび置換を提供します
    - レンダリング技術＆ API
        - ClearType - Microsoft によって開発されたサブピクセルレンダリングテクノロジー。各ピクセルの個別のサブピクセルを使用して液晶ディスプレイ（LCD）上のテキスト読みやすさを向上させます
        - DirectWrite - Microsoft からの高性能テキストレイアウトおよびフォントレンダリング API。ハードウェア高速レンダリングと最新アプリケーション向けの高品質タイポグラフィをサポートします

デザインシステム＆デザインツール

- ビジュアルデザインツール
    - Claude Design - ユーザーが自然会話と反復的な洗練を通じて Claude と協力してポーランド化されたデザイン、プロトタイプ、スライド、マーケティング資料を作成できるビジュアルデザインツールです
    - Figma Design - チーム向けの強力でコラボレーティブなデザインツールです
    - Locofy.ai - デザインからコードへ瞬く間にです
    - Uizard - テキストプロンプトからプロトタイプ、画面、テーマを生成できる AI 搭載デザインツールです
    - Anything - あなたの言葉をコードで構築されたモバイルアプリ、サイト、ツール、製品に変える AI アプリビルダーです
    - v0 - 数分で動作するアプリケーションを生成し、数秒でライブウェブサイトとして公開できる AI 搭載のフルスタック Web アプリケーションビルダーです
- デザインシステム＆ガイドライン

- Material Design - 美しく、使いやすいプロダクトを構築するための Google のオープンソースデザインシステムです
- Apple HIG - すべての Apple プラットフォーム全体でアプリの外観と動作を一貫させるのに役立つ推奨事項のセットです
- GNOME HIG - GNOME デスクトップ向けの高品質で一貫性のある、使いやすいアプリケーション作成ガイドです

Web エクスペリエンス＆パフォーマンス

- Responsive web design - さまざまなデバイスとウィンドウまたは画面サイズで Web ページが適切にレンダリングされるようにすることを目的とした Web 設計へのアプローチです
- Core Web Vitals - すべての Web ページに適用される Web Vitals のサブセット。すべてのサイト所有者が測定する必要があり、すべての Google ツール全体で提示されます
  - Largest Contentful Paint (LCP)
  - Interaction to Next Paint (INP)
  - Cumulative Layout Shift (CLS)

経済学、ゲーム理論＆ファイナンス

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.1 戦略の理解とアーキテクチャ設計 > ビジネス環境と経営戦略の理解
> - 1. ビジネス変革 > 1.1 戦略の理解とアーキテクチャ設計 > ビジネス価値定義／投資対効果の試算と意思決定支援

経済学＆ゲーム理論

- Market - 当事者が交換に従事するシステム、機関、手順、社会関係、またはインフラストラクチャの構成です
- Inflation - 時間経過に伴うエコノミー内の商品およびサービスの一般価格レベルの上昇です
- Prospect theory - 最終結果ではなく損失と利益の潜在的価値に基づいて人々が決定を下すと述べる行動経済学と行動金融の理論です
- Sunk cost - すでに発生しており、もはや回収することのできないコストです
- Choice architecture - 選択肢を意思決定者に提示するさまざまな方法の設計と、その提示が意思決定に与える影響です

    - Nudge theory - 行動経済学および関連する行動科学における概念で、意思決定環境の適応的な設計を、集団や個人の行動と意思決定に影響を与える方法として提案するものです
- Principal–agent problem - 人またはグループと、彼らに代わって行動する権限を与えられた代表者の間の優先順位の競合です
- Information asymmetry - トランザクション内の 1 つのパーティが他方よりもはるかに多くの情報を持っている状況です
- Induced demand - 供給が増加した後、より多くの商品が消費される現象です
- Metcalfe's law - 通信ネットワークの価値はシステムの接続ユーザー数の 2 乗に比例するということです（n2）
    - Network effect - ユーザーが互換性のある製品のユーザー数に依存する商品またはサービスから得る価値またはユーティリティの現象です
- Braess's paradox - 1 つ以上の道路を道路ネットワークに追加することで、全体的なトラフィックフロー全体を遅くすることができるという観察です
- Nash equilibrium - 各プレイヤーが他のプレイヤーの平衡戦略を知っていると仮定される 2 つ以上のプレイヤーを含む非協力的ゲームのソリューションの概念。プレイヤーは自分の戦略のみを変更することで得るものはありません
- Pareto efficiency - リソースの割り当ての状態。再配分することは、少なくとも 1 つの個人または優先順位基準を悪化させずに、個人または優先順位基準をより良くすることは不可能です
- Operations research - マネジメントと意思決定を改善するために分析方法の開発と応用に対処する分野です

コーポレートファイナンス＆マーケット

- Currency - 交換媒体として使用または流通中の任意の形式のお金の標準化です
- Interest - 債務者または預金取得金融機関から貸し手または預金者への支払い。元本返済額（つまり、借用された金額）を超える金額。特定のレートで
- Central bank - 国または通貨連合の金銭政策を管理する機関です
- Revenue model - 金融収入を生成するためのフレームワークです
- Financial capital - 起業家とビジネスがプロダクト製造または提供するサービスに必要なものを購入するために使用する金銭で測定される経済リソースです
    - Venture capital - 高成長可能性があると判断されたスタートアップ、初期段階、新興企業に対してベンチャーキャピタル企業またはファンドによって提供される民間エクイティ資金の形式です
- マーケット＆セキュリティ

    - Stock market - ビジネスの所有権請求を表すストックの買い手と売り手の集約です
    - Stock - 企業の所有権を分割し、通常、利益、清算手続き、または投票力への権利を付与する株式です
    - Dividend - 当年利益または保有利益から株主への企業による利益の配分です
- 契約
    - Credit - 1 つのパーティが別のパーティに金銭またはリソースを提供することを許可する信頼です。2 番目のパーティは最初のパーティにすぐに払い戻しません
    - Debt - 1 つのパーティ（債務者）が別のパーティ（債権者）に金銭またはその他の価値を返す必要があるという義務です
        - Discounting - 債務者が定義された期間にわたって債権者への支払いを遅延する権利を得られる仕組み。手数料または料金と引き換えに
        - Bond - 発行者（債務者）が保有者（債権者）の債務を所有する証券のタイプ。条件に応じて、満期日にボンドの元本を返済し、指定された期間にわたって利息を支払う必要があります
    - Spot - 商品、セキュリティ、または通貨を即座に決済のために売買する契約です
    - Futures - 将来の指定された時間に配信するために、所定の価格でものを売買する標準化された法的契約です
    - Option - 所有者（保有者）に、指定された日付時点またはそれ以前に指定された行使価格で特定の量の基礎となる資産または商品を売却する権利を付与する契約です。義務ではなく
- Cryptocurrency - 金銭としてデジタルファイルを使用する通貨のタイプです

会計＆財務報告

- 会計基礎
    - Asset - ビジネスまたは経済的実体が所有またはコントロールし、正の経済価値を生成するために使用できるリソースです
    - Liability - 金融的実体が過去のイベントから生じた現在の義務を満たすために将来に提供することが期待されている価値の量です
    - Equity - 債務または他の債務を受ける可能性のあるプロパティの所有権利益。負債を資産の価値から差し引いて測定されます
    - Revenue - ビジネスの一次操作に関連する商品およびサービスの販売によって生成される総収入です
    - Depreciation - 資産の価値の低下。有用なライフスパン全体にわたる有形資産のコストを再配分するのに使用される方法です

    - Accrual - 現金受領または支払い時ではなく、獲得または発生時に収益および費用を認識する会計方法です
- 財務諸表＆メトリクス
    - Balance sheet - 個人または組織の財務残高のサマリーです
    - Income statement - 企業の財務諸表の 1 つ。特定期間の企業の財務パフォーマンスを示しています
    - Cash flow statement - バランスシート勘定および収入の変化がどのように現金および現金同等物に影響するかを示す財務諸表です
    - Return on investment - 期間にわたるネット収入とリソースの投資時点から生じる投資コストの比率です
    - Net present value - 資産が生成する将来のすべてのキャッシュフローの現在価値を追加することで、キャッシュフローを持つ資産の価値を測定する方法です
    - EBITDA - 債務の影響、州が要求される支払い、資産基盤を維持するために必要なコストの前に操作ビジネスのプロフィタビリティの測定です
    - Operating margin - 通常パーセント で表現される、純売上高に対する営業収入の比率です
    - Burn rate - 企業がキャッシュを消費する速度です。通常月単位で表現されます。スタートアップが株主資本を使い切る速さを測定するために使用されます
    - Liquidity - マーケットの機能。個人またはファームが資産の価格に激劇的な変化を引き起こさずに資産をすばやく購入または販売できます
    - Valuation - 潜在的投資、資産、またはセキュリティの価値を決定するプロセスです
- 会計基準＆プロセス
    - Generally Accepted Accounting Principles - 実施する必要があるロールアップを詳細に規定する会計基準。財務諸表をどのように提示するのか。どのような追加開示が必要かです
    - Audit - 任意のエンティティの財務情報の独立した検査。その上で意見を表現する見方で実施されます

02 - ウェブアプリケーション開発

ウェブプラットフォーム基礎

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > Webアプリケーション基本技術

ウェブコンセプト

- World Wide Web - ドキュメントおよびその他のウェブリソースが Uniform Resource Locator (URL) で識別され、ハイパーテキストリンクで相互に関連付けられ、インターネットを通じてアクセスできる情報空間です
  - Hypertext - コンピュータディスプレイやその他の電子デバイスに表示されるテキストで、読者が即座にアクセスできる他のテキストへの参照 (ハイパーリンク) を含むテキストです
  - Semantic Web - W3C によって設定された標準を通じてインターネットデータをマシンリーダブルにする World Wide Web の拡張で、自動エージェントがより知的に情報を処理できるようにします
  - URI - 論理的または物理的なリソースを識別する一意の文字列です
  - URL - URL 、ドメイン、IP アドレス、application/x-www-form-urlencoded フォーマット、およびそれらの API を定義する標準です
- コアウェブプロトコル及び言語
  - HTTP - 分散型の協調的なハイパーメディア情報システム用のアプリケーションプロトコルです
    - HTTP cookie - サーバーがユーザーのウェブブラウザに送信する小さなデータです
  - HTML - World Wide Web のコアマークアップ言語です
  - CSS - ウェブドキュメントにスタイル (フォント、色、間隔など) を追加するシンプルなメカニズムです
- リアルタイム及びメッセージングプロトコル
  - WebSockets - ユーザーのブラウザとサーバー間で双方向対話型通信セッションを開くことを可能にするテクノロジーです
  - WebRTC - ウェブブラウザとモバイルアプリケーションにリアルタイム通信 (RTC) を提供する無料のオープンソースプロジェクトです
  - Server-sent events - サーバーが HTTP またはデディケートサーバープッシュプロトコルを使用してウェブページにデータをプッシュできるようにするテクノロジーです
  - MQTT - メッセージキュー / メッセージキューイングサービス用の軽量で publish-subscribe な machine to machine ネットワークプロトコルです
  - AMQP - メッセージ指向ミドルウェア用のオープンな標準アプリケーション層プロトコルです
- データ及びイベント仕様
  - CloudEvents - イベントデータを共通の方法で説明するための仕様です

    - JSON Merge Patch - ターゲット JSON ドキュメントに対して行われる変更を記述する JSON フォーマットです
    - OpenAPI spec - HTTP API への標準的で言語に依存しないインターフェースです
    - TypeSpec - 開発者が使い慣れた方法で API 形状を説明するのに役立つ最小限の言語です
    - API ツーリング
        - Redocly CLI - OpenAPI 定義をリント、バンドル、およびプレビューするのに役立つオープンソースコマンドラインツールです
- ウェブパフォーマンスコンセプト
    - DNS Prefetching - ユーザーがリンクをたどる前にドメイン名を解決するメカニズムです
- ウェブアプリケーションタイプ
    - Progressive web app - HTML 、CSS 、JavaScript 、および WebAssembly を含む共通ウェブテクノロジーを使用して構築されたウェブを通じて配信されるアプリケーションソフトウェアの一種です

ブラウザテクノロジー及び DOM

- ブラウザ
    - Chrome - Google が開発したフリーウェアのクロスプラットフォームウェブブラウザです
    - Chromium - すべてのユーザーがウェブをより安全で高速で安定した方法で体験できるようにすることを目指すオープンソースブラウザプロジェクトです
    - Firefox - Mozilla Foundation によって開発された無料でオープンソースのウェブブラウザです
    - w3m - テキストベースのウェブブラウザおよびページャーです
    - EWW - Emacs Web Wowser で、Emacs 用のウェブブラウザです
- レンダリングエンジン
    - WebKit - アプリケーションに豊富でインタラクティブなウェブコンテンツを表示するためのフレームワークです
    - Gecko - Mozilla が開発したウェブブラウザエンジンです
    - Blink - Chromium で使用されるレンダリングエンジンです
    - Servo - アプリケーションと組み込み使用の両方向けに設計された最新の高性能ブラウザエンジンです
- スクリプトエンジン

    - V8 (JavaScript engine) - Google のオープンソースで高性能な JavaScript および WebAssembly エンジンで、C++ で書かれています
    - JavaScriptCore - Apple プラットフォームの Safari およびその他のアプリケーションを動かす JavaScript エンジンです
- クライアントスクリプティング API
    - XMLHttpRequest (XHR) - クライアントとサーバー間のデータ転送のためのスクリプト付きクライアント機能を提供する API です
    - Fetch Standard - リクエスト、レスポンス、およびそれらをバインドするプロセス (フェッチ) を定義する現行標準です
    - Canvas API - JavaScript および HTML <canvas> 要素を使用してグラフィックスを描画するための手段です
    - WebGL API - プラグインを使用せずに互換性のあるウェブブラウザ内で高性能なインタラクティブ 3D および 2D グラフィックスをレンダリングするための JavaScript API です
    - Web Neural Network API (WebNN) - GPU、CPU、NPU などのオンデバイスハードウェアを利用して、ウェブアプリやフレームワークがディープニューラルネットワークを高速化できるようにする新興のウェブ標準です
- サイト分析ツール
    - Wappalyzer - ウェブサイトが何で構築されているかを表示するテクノロジープロファイラーです

ウェブアプリケーションアーキテクチャ

- Single-page application - ウェブサーバーからの新しいデータを使用して現在のウェブページを動的に書き直すことでユーザーと相互作用するウェブアプリケーションまたはウェブサイトです
- Multi-page application - 複数ページを備えた従来のウェブ構造で、各ページが独自の URL を持ち、ユーザーがリクエストしたときに個別にダウンロードおよび読み込まれます
- Microfrontend - 独立して開発されたフロントエンドが全体に構成されるウェブ開発のためのアーキテクチャパターンです
- Islands Architecture - ページを高速な静的 HTML にレンダリングし、インタラクティビティが必要な場所にのみ JavaScript の選択的な「島」を追加するフロントエンドパターンです
- Backend for Frontend - 異なるフロントエンドアプリケーション向けに特別に作成されたバックエンドサービスの別々のセットが作成されるアーキテクチャパターンです
- Multitier architecture - ソフトウェアアーキテクチャの異なるレベルがプレゼンテー

ション、アプリケーション処理、およびデータマネジメント機能に物理的に分離されている client-server アーキテクチャです

- Server-side rendering - サーバーから静的 HTML をクライアントに送信し、クライアント側の JavaScript がハイドレーションと呼ばれるプロセスでイベントハンドラーを接続してウェブページを動的にする方法です
- Incremental Static Regeneration - サイト全体を再構築する必要なく、ページごとに静的生成を有効にし、デプロイ後に静的コンテンツを更新できるテクニックです
- JAMstack - ウェブエクスペリエンスレイヤーをデータとビジネスロジックから切り離し、柔軟性、スケーラビリティ、パフォーマンス、および保守性を向上させるアーキテクチャアプローチです

フロントエンド開発

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > フロントエンドシステム開発
> - 1. ビジネス変革 > 1.4 デザイン > デジタルプロダクト設計

UI フレームワーク及びコアライブラリ

- コア SPA フレームワーク
  - React - ウェブおよびネイティブユーザーインターフェース用のライブラリです
    - コアコンセプト
      - Component - ユーザーインターフェースを作成するために使用される基本的なビルディングブロックです
      - Props - 親コンポーネントから子コンポーネントへデータを渡すためのメカニズムです
      - Children - コンポーネントを構成できるようにする特別なプロップです
      - Key Props - 要素のリストを作成する際に含める必要がある特別な文字列属性です
      - Rendering - React がコンポーネントに外観を説明するよう要求するプロセスです
      - Event Handler - イベントに応答して実行される関数です
      - State - コンポーネントの動的データを格納する JavaScript オブジェクトです
      - Controlled Component - React state が入力要素の値を制御するコンポーネントです

        - Hooks - 関数コンポーネントから React state およびライフサイクル機能に「フック」できる関数のセットです
        - Strict Mode - アプリケーションの潜在的な問題を強調するためのツールです
        - Side-effect - 実行中の関数の外側の何かに影響を与える操作を指す用語です
        - Refs - レンダーメソッドで作成された DOM ノードまたは React 要素にアクセスする方法を提供する機能です
        - Context - すべてのレベルでプロップを手動で渡す必要なく、コンポーネントツリーを通じてデータを渡す方法です
        - Portals - 親コンポーネントの DOM 階層の外に存在する DOM ノードに子をレンダリングするための第一級の方法を提供する機能です
        - Suspense - コンポーネントツリーの一部に対してローディングインジケーターを指定できるコンポーネントです
        - Error Boundary - 子コンポーネントツリーの任意の場所で JavaScript エラーをキャッチする React コンポーネントです
    - Preact - 同じ最新 API を備えた React への高速な 3kB 代替です
    - Vue.js - ユーザーインターフェースを構築するための JavaScript フレームワークです
    - Angular - 開発者が高速で信頼性の高いアプリケーションを構築できるようにするウェブフレームワークです
    - Svelte - コンパイラーを使用して、HTML 、CSS 、JavaScript という言語を使用して簡潔なコンポーネントを記述し、ブラウザで最小限の処理を行う UI フレームワークです
    - Ember.js - 野心的なウェブ開発者向けのフレームワークです
- HTML -First フレームワーク
    - htmx - HTML 属性を使用して AJAX 、CSS Transitions 、WebSockets 、Server Sent Events に直接アクセスできるライブラリです
    - htm - トランスパイラーが不要な標準的なタグ付きテンプレートを使用した JSX 代替です
- フレームワークに依存しないコアライブラリ
    - TanStack - ウェブ開発向けの高品質でフレームワークに依存しないオープンソースライブラリの集合です
        - TanStack Query - TS /JS 、React 、Solid 、Vue 、Svelte 、Angular 向けの強力な非同期 state management です

        - TanStack Router - 最新のウェブアプリケーションを構築するための強力で型安全でフレームワークに依存しないルーターです
        - TanStack Table - TS /JS 、React 、Vue 、Solid 、Svelte 向けの強力なテーブル及びデータグリッドを構築するためのヘッドレス UI です
        - TanStack Form - React 、Vue 、Solid 、Svelte 向けの型安全でフレームワークに依存しないフォーム state management です
        - TanStack Virtual - React 、Vue 、Svelte 、Solid 、JS で大きなリスト及びグリッドを仮想化するためのヘッドレス UI です

State 、ルーティング及びロジック

- State Management
    - Redux - 予測可能で保守可能なグローバル state management のための JS ライブラリです
        - React-Redux - Redux の公式 React バインディングです
    - Zustand - 簡略化された flux 原則を使用した小型で高速でスケーラブルなベアボーン state management ソリューションです
    - Recoil - React 用の state management ライブラリです
    - XState - 有限状態マシンおよび statechart を作成、解釈、実行するためのライブラリです
- ルーティング
    - React Router - ユーザーを重視した標準に焦点を当てた、どこにでもデプロイできるマルチストラテジールーターです
- 構文及びテンプレート
    - JSX - JavaScript ファイル内に HTML のようなマークアップを記述できる JavaScript の構文拡張です
    - MDX - Markdown ドキュメント内で JSX をシームレスに記述できるようにする作者向けフォーマットです
- WASM ランタイム
    - PyScript - Python アプリケーションの作成、デプロイ、および共有を容易にする無料のオープンソースソフトウェア (OSS) です

スタイリング及び UI コンポーネント

- CSS エコシステム
    - フレームワーク及び UI キット

        - Bootstrap - 世界で最も人気のあるフロントエンドオープンソースツールキットです
        - Tailwind CSS - クラスが充実したユーティリティファースト CSS フレームワークです
        - Oat - 最小限で標準ベースの CSS および JS を提供する超軽量でセマンティックでゼロ依存の HTML UI コンポーネントライブラリです
    - Tailwind コンポーネントライブラリ
        - daisyUI - Tailwind CSS で最も人気のあるコンポーネントライブラリです
    - CSS -in-JS
        - Emotion - JavaScript でスタイルを記述するように設計されたライブラリです
        - Linaria - ゼロランタイム CSS in JS ライブラリです
    - プリプロセッサ
        - Sass language - CSS にコンパイルされるスタイルシート言語です
    - 変換
        - CSS Transforms 1 - 要素を 2 次元空間で変換できる CSS モジュールです
        - CSS Transforms 2 - 要素を 3 次元空間で変換できる CSS モジュールです
- UI コンポーネントライブラリ
    - templUI - Go および templ 向けに美しく設計された UI コンポーネントの増加するコレクションです
    - Material UI - Google の Material Design を実装するオープンソース React コンポーネントライブラリです
    - Chakra UI - 製品を高速に構築するためのコンポーネントシステムです
    - Vuetify - デザインスキルが不要なオープンソース UI ライブラリで、美しく丁寧に作られた Vue コンポーネントです
- 特殊な UI ウィジェット
    - リッチテキストエディター
        - Tiptap - ウェブ開発者向けに設計されたヘッドレスでオープンソースのエディターフレームワークです
    - インタラクション及びメディア
        - Swiper.js - ハードウェアアクセル付きトランジションと素晴らしいネイティブな動作を備えた最新のモバイルタッチスライダーです
        - Hammer.js - マルチタッチジェスチャー用の JavaScript ライブラリです
    - キャンバス及びホワイトボード

        - tldraw - 高性能な Web キャンバスでホワイトボード、ダイアグラム、キャンバスツールを構築するための React ベースの SDK です

ビルド及び開発ツーリング

- 開発環境
  - Storybook - UI コンポーネントおよびページを分離して構築するためのフロントエンドワークショップです
- バンドラー
  - Vite - 最新のウェブプロジェクトのためにより高速でより充実した開発体験を提供することを目指すビルドツールです
  - Parcel - ゼロ設定ビルドツールです
  - webpack - 最新の JavaScript アプリケーション向けの静的モジュールバンドラーです
  - Rspack - Rust で書かれた高性能な JavaScript バンドラーです
  - Rsbuild - Rspack ベースのウェブビルドツールです
- トランスパイラー
  - babel - JavaScript コンパイラーです
- ミニファイアー
  - JSMin - JavaScript ファイルからコメントと不要な空白を削除するミニフィケーションツールです
- リンター及びフォーマッター
  - Biome - JavaScript 、TypeScript 、JSX 、TSX 、JSON 、HTML 、CSS 、GraphQL 用の高速フォーマッター及びリンターで、ウェブプロジェクト向けの統合的なツールチェーンを提供します
  - Knip - JavaScript および TypeScript プロジェクトで未使用の依存関係、エクスポート、およびファイルを検出して修正するツールです

フルスタック及び静的サイトフレームワーク

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > フロントエンドシステム開発
> - 3. テクノロジー > 3.1 ソフトウェア開発 > バックエンドシステム開発

フルスタックフレームワーク

- JS /TS フルスタックフレームワーク
  - Next.js - フルスタックウェブアプリケーションを構築するための React フレームワークです
  - Nuxt.js - Vue.js を使用してタイプセーフで高性能で本番環境対応のフルスタックウェブアプリケーションおよびウェブサイトを作成するための直感的でかつ拡張可能な無料のオープンソースフレームワークです
  - Astro - コンテンツ駆動型ウェブサイト向けのウェブフレームワークです
  - Fresh - 速度、信頼性、シンプルさのために構築された次世代ウェブフレームワークです
- Rust フルスタックフレームワーク
  - Leptos - 最新ウェブ向けの最先端の Rust フレームワークです

静的サイトジェネレーター

- Docusaurus - 静的サイトジェネレーター。Markdown ファイルから単一ページアプリケーションを構築し、React の全力を活用して高速なクライアント側ナビゲーションを活かしてサイトをインタラクティブにします
- mdBook - Markdown ファイルから最新のオンライン書籍を作成するユーティリティです
- VuePress - Vue -powered 静的サイトジェネレーターです
- Hugo - ウェブサイト構築向けの世界で最速のフレームワークです
  - Docsy - テクニカルドキュメンテーションサイト向けの Hugo テーマで、簡単なサイトナビゲーション、構造などを提供します
- Jekyll - 個人、プロジェクト、または組織向けサイトに完璧なシンプルでブログ対応の静的サイトジェネレーターです
- Eleventy - JavaScript で書かれたシンプルな静的サイトジェネレーターです
- Sphinx - インテリジェントで美しいドキュメンテーションを簡単に作成できるツールです
- MkDocs - プロジェクトドキュメント構築向けに重視されたシンプルで美しい静的サイトジェネレーターです
  - Material for MkDocs - MkDocs 静的サイトジェネレーター向けの強力で美しいテーマです
- Nanoc - 小さな個人ブログから大きな企業ウェブサイトまで、何でも構築するのに適した静的サイトジェネレーターです

- gitmal - Git リポジトリ向けに設計された静的ページジェネレーターです

ヘッドレス CMS

- クラウドネイティブ及び API -first CMS
  - Contentful - コンテンツファースト方式でデジタル製品構築を可能にするヘッドレスコンテンツ管理システムです
  - Strapi - 最先端のオープンソースヘッドレス CMS です
  - Sanity - より良いデジタル体験を構築できるようにする構造化コンテンツ用プラットフォームです

バックエンド開発

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > バックエンドシステム開発

API アーキテクチャスタイル

- REST - World Wide Web のアーキテクチャをガイドし開発するために作成されたソフトウェアアーキテクチャスタイルです
- SOAP (legacy) - ウェブサービスの実装で構造化情報を交換するためのメッセージングプロトコル仕様です
- GraphQL - API のクエリ言語および既存データを使用してそれらのクエリを満たすためのランタイムです
- gRPC - あらゆる環境で実行できる最新のオープンソースで高性能なリモートプロシージャコール (RPC) フレームワークです
- json-rpc - ステートレスで軽量なリモートプロシージャコール (RPC) プロトコルです
- Webhook - ウェブページまたはウェブアプリケーションの動作をカスタムコールバックで拡張または変更する方法です

バックエンドフレームワーク

- JS /TS バックエンドフレームワーク
  - Fastify - Node.js 向けの高速で低オーバーヘッドのウェブフレームワークで、最適なパフォーマンスと開発者体験のために設計されています
  - Express.js - 最小限で柔軟な Node.js ウェブアプリケーションフレームワークです

    - Koa - Express の背後にあるチームによって設計された新しいウェブフレームワークです
    - Nest.js - 効率的で信頼性の高いスケーラブルなサーバー側アプリケーション構築向けの進歩的な Node.js フレームワークです
    - Hono - エッジ向けの小型でシンプルで超高速なウェブフレームワークです
- API ツール
    - tRPC - スキーマまたはコード生成なしに完全にタイプセーフな API を簡単に構築して消費できるツールです
- Go バックエンドフレームワーク
    - Echo - 高性能で拡張可能なミニマリスト Go ウェブフレームワークです
    - Fiber - Go 向けの最速の HTTP エンジンである Fasthttp の上に構築された Express にインスパイアされたウェブフレームワークで、パフォーマンスに配慮した開発を容易にするように設計されています
    - Gin Web Framework - Go で書かれたウェブフレームワークです
    - Gorilla web toolkit - HTTP ベースのアプリケーション記述向けに有用で構成可能なパッケージを提供する有益なツールキットです
    - Yokai - バックエンドアプリケーション向けのシンプルでモジュール式で監視可能な Go フレームワークです
    - Kratos - トランスポート、ミドルウェア、レジストリ、設定、ロギング、コード生成のための API を備えた、クラウドネイティブなマイクロサービスを構築するための軽量な Go フレームワークです
- Python バックエンドフレームワーク及びサーバー
    - WSGI - Web Server Gateway Interface
        - Gunicorn - UNIX 向けの Python WSGI HTTP サーバーです
        - Flask - 軽量な WSGI ウェブアプリケーションフレームワークです
    - ASGI - WSGI の精神的後継で、ウェブサーバー、フレームワーク、アプリケーション間の互換性のための長年の Python 標準です
        - Uvicorn - uvloop と httptools を使用した高性能向けの Python 用の稲妻高速 ASGI サーバー実装です
        - Hypercorn - sans-io hyper 、h11 、h2 、wsproto ライブラリベースの ASGI および WSGI ウェブサーバーで、HTTP /1 、HTTP /2 、HTTP /3 のサポート付きです
        - FastAPI - 標準 Python 型ヒントに基づいて Python で API を構築するための最新で高速 (高性能) なウェブフレームワークです

        - SlowAPI - ASGI アプリケーションをレート制限するための小さなライブラリです
- Ruby バックエンドフレームワーク及びサーバー
    - Ruby on Rails - Model - View - Controller (MVC) パターンに従うデータベース-バック型ウェブアプリケーション作成に必要なすべてを含むウェブアプリケーションフレームワークです
    - Rack - モジュール式 Ruby ウェブサーバーインターフェースです
        - Puma - Ruby 及び Rack 向けの高速で並行処理対応のウェブサーバーです
        - Falcon - async 、async -container 、async -http の上に構築された multi -process 、multi -fiber rack -compatible HTTP サーバーです
        - Sinatra - 最小限の手間で Ruby でウェブアプリケーションを迅速に作成するための DSL です
    - Sidekiq - Ruby 向けのシンプルで効率的なバックグラウンド処理ツールです
    - Shrine - Ruby アプリケーション向けのファイル添付ツールキットです
- Perl バックエンドフレームワーク (legacy)
    - クラシック CGI
        - mod_cgi - CGI スクリプト実行向けのモジュールです
        - CGI.pm - Common Gateway Interface リクエストおよびレスポンス処理向けのモジュールです
    - Fast CGI
        - mod_fcgid - mod_cgi または cgid に代わる高性能です
        - FCGI.pm - FastCGI アプリケーション向けのモジュールです
- Java バックエンドフレームワーク
    - Jakarta EE - エンタープライズソフトウェア開発向けの Java API を定義する仕様のセットです
        - Apache Tomcat - オープンソースのウェブサーバーおよびサーブレットコンテナーです
    - Spring - Java をシンプルで最新で生産的でリアクティブでクラウドレディにするプロジェクトです
        - Spring Boot - Spring プラットフォームおよびサードパーティーライブラリに対して意見を提供するツールで、最小限の手間で開始できます
- .NET バックエンドフレームワーク
    - ASP.NET - .NET および C# でウェブアプリケーションおよびサービスを構築するための無料でクロスプラットフォームのオープンソースフレームワークです

- Elixir バックエンドフレームワーク
  - Phoenix - より少ないコードおよびより少ない可動部でより豊かでインタラクティブなウェブアプリケーションを迅速に構築し、API 、HTML5 アプリケーションなど大規模で構築するためのウェブフレームワークです
- GraphQL サーバー
  - Apollo Server - オープンソースで仕様準拠の GraphQL サーバーで、任意の GraphQL クライアントと互換性があります

ウェブインフラストラクチャ

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

ウェブサーバー及びプロキシ

- ウェブサーバー及びリバースプロキシサーバー
  - NGINX - ウェブサービス、リバースプロキシ、キャッシング、ロードバランシング、メディアストリーミング、およびその他のためのオープンソースソフトウェアです
  - Apache HTTP Server - UNIX および Windows を含む最新のオペレーティングシステム向けのオープンソース HTTP サーバーを開発および保守するプロジェクトです
  - Caddy - サイト、サービス、およびアプリケーションを提供するための強力で拡張可能なプラットフォームで、Go で書かれています
  - HAProxy - TCP および HTTP ベースのアプリケーション向けに高可用性、ロードバランシング、およびプロキシを提供する無料で非常に高速で信頼性の高いリバースプロキシです
  - nodejs http-server - シンプルな静的 HTTP サーバーです
  - goshs - HTTP /S 、WebDAV 、SFTP 、SMB 、LDAP /S 、NTLM ハッシュキャプチャ、DNS /SMTP コールバック、TLS 、認証、共有リンクをサポートする赤チーム及び開発者向けの機能豊富なシングルバイナリファイルサーバーです
- API マネジメント
  - Unkey - 開発者が API をセキュア、管理、スケールするのに役立つようにオープンソースの API 管理プラットフォームです
  - Kong API gateway - 軽量で高速で柔軟なクラウドネイティブ API ゲートウェイです

- Azure API Management - すべての環境の API を管理するためのハイブリッドで multicloud マネジメントプラットフォームです
- Amazon API Gateway - 開発者があらゆるスケールで API を作成、公開、保守、監視、セキュアに行うのを容易にする完全に管理されたサービスです
- Google Cloud Apigee - API サービスの開発と管理向けのプラットフォームです
- Gravitee - あらゆるインフラストラクチャ全体の API を管理、セキュア化、統治するための単一ペインガラスを提供する統一 API 可視性及び統治プラットフォームです

CDN 及びエッジコンピューティング

- コンセプト
  - Web cache - 帯域幅使用量、サーバー負荷、および認識遅延を削減するために HTML ページおよび画像などのウェブドキュメントの一時保管 (キャッシング) のための情報技術です
  - Content delivery network - プロキシサーバーおよびそれらのデータセンターの地理的に分散されたネットワークです
    - Point of presence - 通信エンティティ間の人工的な分界点またはインターフェースポイントです
- フォワードプロキシサーバー
  - Squid - HTTP 、HTTPS 、FTP などをサポートするウェブ向けのキャッシングプロキシです
- CDN プロバイダー
  - Cloudflare - インターネットに接続するすべてのものをセキュア、プライベート、高速、信頼性の高くするために設計されたグローバルネットワークです
    - Cloudflare Workers - インフラストラクチャの設定や保守なしで、まったく新しいアプリケーションを作成したり既存のアプリケーションを拡張したりできるサーバーレス実行環境です
    - Cloudflare Workers Bindings - Worker が Cloudflare Developer Platform のリソースと相互作用できるようにするメカニズムで、Worker からリソースにアクセスするための REST API よりも優れたパフォーマンスおよびより少ない制限を提供します
  - Amazon CloudFront - 高性能、セキュリティ、開発者の利便性のために構築されたコンテンツ配信ネットワーク (CDN) サービスです
    - Lambda@Edge - アプリケーションのユーザーに近い場所でコードを実行できるようにする Amazon CloudFront の機能です

    - Google Cloud CDN - ウェブおよびビデオコンテンツ配信を加速化するコンテンツ配信ネットワーク (CDN) です
    - Azure Front Door - グローバルウェブアプリケーション及びコンテンツの高速配信向けのセキュアでスケーラブルなエントリポイントを提供する最新のクラウドコンテンツ配信ネットワーク (CDN) です
- JAMstack ホスティング
    - GitLab Pages - GitLab リポジトリから静的ウェブサイトを直接公開できるようにする機能です
    - Cloudflare Pages - フロントエンド開発者がウェブサイトで協力および デプロイするための JAMstack プラットフォームです

分散型ウェブ

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.2 デジタルテクノロジー > その他先端技術

ブロックチェーンテクノロジー

- Web3 - 分散化、ブロックチェーンテクノロジー、トークンベースの経済学などの概念を組み込む World Wide Web の新しいイテレーションのアイデアです
- Blockchain - レコードの増加するリストを持つ分散台帳です
    - Hashcash - メールスパムおよびサービス拒否攻撃を制限するために使用される proof-of -work システムです
    - Proof of work - 一方の当事者が一定量の特定の計算努力が消費されたことを他者に証明する暗号証明の形式です
- Smart contract - 契約条件に従ってイベントおよびアクションを自動的に実行、制御、またはドキュメント化するコンピュータプログラムまたはトランザクションプロトコルです
- Bitcoin - ピアツーピア bitcoin ネットワークで転送できる分散化されたデジタル通貨です
- Ethereum - 集中管理当局からの許可を必要としないデジタル資産、データ、およびアイデンティティの直接所有権を提供するグローバルな分散化ネットワークです
- Non-fungible token - ブロックチェーンに記録され、所有権および真正性を証明するために使用される一意のデジタル識別子です
- Decentralized autonomous organization - 集中管理上のリーダーシップなしのメンバー所有コミュニティで、投票およびファイナンスをブロックチェーンで処理する分散化コンピュータプログラムによって管理されます

- Solidity - 様々なブロックチェーンプラットフォーム、最も顕著には Ethereum でスマートコントラクトを実装するためのプログラミング言語です
- Web3.js - 開発者が Ethereum および他の EVM -compatible ブロックチェーンに接続および相互作用できるようにする TypeScript /JavaScript ライブラリです
- ethers.js - Ethereum のすべての必要のための シンプルで、コンパクトで完全な JavaScript ライブラリです
- MetaMask - ユーザーがデータおよび資産に対する制御を維持しながら、暗号化通貨の購入、販売、スワップ、および保存を有効にする暗号化ウォレットです
- WalletConnect - モバイル暗号化ウォレットおよびデスクトップベースの分散化アプリケーション間で暗号化接続を確立するオープンソースプロトコルです
- Hardhat - Ethereum および EVM -compatible ブロックチェーン向けの開発環境で、開発者が Solidity スマートコントラクトをコンパイル、デプロイ、テスト、デバッグするのに役立ちます

分散化ソーシャル

- ActivityPub - ActivityStreams 2.0 データフォーマットに基づく分散化ソーシャルネットワーキングプロトコルです
- AT Protocol - ユーザーがアイデンティティを所有し、コンテンツが相互に関連した JSON レコードとして表現されるソーシャルアプリケーション構築向けのオープンデータネットワークです
- Fediverse - ウェブパブリッシングおよびファイルホスティング向けに使用され、相互に通信できる相互接続サーバーのアンサンブルです
- Mastodon - ユーザーにアルゴリズムまたは広告なしでフィードを制御させ、独立したサーバーが ActivityPub プロトコルを通じて相互運用することを許可する無料でオープンソースの分散化ソーシャルメディアプラットフォームです
- Bluesky - アメリカを拠点とする公益企業であるマイクロブログソーシャルメディアサービスおよび公益企業です
- Nostr - 暗号署名を使用して複数の独立したサーバー（ リレー ）と呼ばれる検閲耐性通信を有効にするオープンで分散化ソーシャルプロトコルです
- Matrix - 異なるサービスプロバイダー間でシームレスな通信を可能にするリアルタイム通信向けのオープン標準および通信プロトコルです
- PeerTube - 独立したビデオホスティングプラットフォーム作成向けの無料でオープンソースのツールで、分散化ネットワークを形成するために接続され、集中型サービスに代わるものを提供します
- Lemmy - ユーザーが企業追跡またはコマーケティングなしで経験を制御できるようにする分散化ディスカッションプラットフォームです

- Diaspora - ネットワークを形成するために相互運用する pod と呼ばれる独立して所有されたノードで構成される非営利でユーザー所有の分散ソーシャルネットワークです
- Secure Scuttlebutt - 企業データ収集から自由なローカルコミュニティ開発を有効にする分散化ソーシャルネットワークプラットフォームです

開発及びテストツール

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

ウェブ /HTTP クライアント

- HTTP CLI ツール
  - cURL - URL でデータを転送するためのコマンドラインツール及びライブラリです
  - Wget - HTTP 、HTTPS 、FTP 、FTPS を使用してファイルを取得するための無料ソフトウェアパッケージです
  - curlie - curl のパワーと httpie の使いやすさです
  - hurl - シンプルなプレーンテキストフォーマットで定義された HTTP リクエストを実行するコマンドラインツールです
  - httpie cli - API エラについての シンプルで強力なコマンドライン HTTP および API テストクライアントです
  - wuzz - HTTP 検査向けのインタラクティブ CLI ツールです
  - httptap - Linux プログラムによって行われた HTTP および HTTPS リクエストを表示するツールです
- HTTP クライアントライブラリ
  - Python Requests - 人間のために構築された優雅でシンプルな Python 向け HTTP ライブラリです
  - JS Axios - node .js およびブラウザ向けの promise ベースの HTTP クライアントです
  - Go Resty - Go 向けのシンプルな HTTP および REST クライアントライブラリです
  - Go FastHTTP - Go 向けの高速 HTTP パッケージです
  - Surf - Chrome /Firefox ブラウザ偽装、HTTP /3 と QUIC フィンガープリント、JA3 /JA4 TLS エミュレーション、アンチボット回避を備えた高度な Go HTTP クライアントです

    - Typhoeus - libcurl をラップして高速で信頼性の高いリクエストを行うライブラリです
    - Ruby Net - client -side インターネットプロトコルを実装するクラスの集合です
    - httpx - Ruby プログラミング言語向けの HTTP クライアントライブラリです
    - wreq-ruby - 正確に様々なブラウザを TLS /HTTP2 署名で エミュレートする高度なブラウザフィンガープリント付きの簡単で強力な Ruby HTTP クライアントです
    - Rust reqwest - エルゴノミックで非同期 HTTP クライアントです
- GraphQL ライブラリ
    - URQL - React 、Svelte 、Vue 、またはプレーン JavaScript 向けの高度にカスタマイズ可能で多目的な GraphQL クライアントです
- API テストプラットフォーム
    - Bruno - Git 統合でフルオフラインでオープンソースの API クライアントです
    - Postman/Newman - API 構築および使用向けの API プラットフォームです
- クラシックウェブオートメーション
    - Mechanize - ウェブサイトとの相互作用を自動化するのに役立つモジュールです
    - Mechanize (Ruby) - 自動化されたウェブ相互作用を簡単にする ruby ライブラリです

ウェブデバッグツール

- Chrome DevTools - Google Chrome ブラウザに直接組み込まれたウェブ開発者ツールのセットです
- Firefox Developer Tools - HTML 、CSS 、JavaScript を検査、編集、デバッグできるようにする Firefox に組み込まれたウェブ開発者ツールのセットです
- React Developer Tools - React コンポーネント を検査し、props および state を編集し、React アプリケーション内のパフォーマンス問題を特定できるブラウザ拡張及びスタンドアロンデバッガーです
- Vue.js devtools - コンポーネント検査および state management デバッグを提供する Vue .js アプリケーションデバッグ向けのブラウザ拡張です
- Redux DevTools - ホットリロード、アクション再生、カスタマイズ可能な UI を含む Redux 開発ワークフロー向けの power-ups を提供する開発ツールです
- Lighthouse - パフォーマンス、アクセシビリティ、SEO 、ベストプラクティスを監査することでウェブページの品質を向上させるのに役立つオープンソースで自動化されたツールです

- Fiddler - あらゆるブラウザ、システム、プラットフォーム向けの無料ウェブデバッグプロキシです
- Charles Proxy - 開発者がマシンとインターネット間のすべての HTTP および SSL/HTTPS トラフィック (リクエスト、レスポンス、ヘッダーを含む) を表示できるようにする HTTP プロキシ/モニターです
- mitmproxy - デバッグ、テスト、ペネトレーションテスト向けウェブトラフィックをインターセプト、検査、修正、再生できる無料でオープンソースのインタラクティブ HTTPS プロキシです
- Requestly - 開発者がデバッグとテスト向けにリアルタイムで URL 、ヘッダー、API レスポンスを修正できるようにする HTTP インターセプターです

ウェブテストオートメーションフレームワーク

- ブラウザオートメーション及びテスト
  - Puppeteer - DevTools Protocol を通じて Chrome /Chromium を制御する高レベルの API を提供する Node .js ライブラリです
  - Playwright - Chromium 、Firefox 、WebKit 向けの単一 API で最新のウェブアプリケーションの信頼性の高いエンドツーエンドテスト向けのフレームワークです
    - Playwright for Go - 単一 API で Chromium 、Firefox 、WebKit を自動化する Go ライブラリです
  - Cypress - 開発者が最新のウェブアプリケーション向けブラウザー内でエンドツーエンド及びコンポーネントテストを直接記述、実行、デバッグできるようにするオープンソースの JavaScript ベーステストフレームワークです
  - WebDriver - user agent の遠隔制御インターフェースを有効にする仕様です
    - Selenium WebDriver - ローカルまたはリモートマシンで user agent をネイティブにドライブするツールです
    - WebDriver BiDi - Bidirectional WebDriver Protocol で、user agent のリモート制御向けのメカニズムです
  - Selenium IDE - ウェブ向けのオープンソースレコード及びプレイバックテストオートメーションです
  - Chrome DevTools Protocol (CDP) - Chromium ベースのブラウザーをインストルメント、検査、デバッグ、及びプロファイルするために外部ツールを有効にするする低レベル API です
  - Karma - ウェブサーバーをスポーンし、接続された各ブラウザーのテストコードに対してソースコードを実行するテストランナーです
  - サポーティングツール

        - Chrome for Testing - ウェブアプリケーション テスト及びオートメーション使用例を特別に対象とする Chrome の新しいフレーバーです
- アクセシビリティテスト
    - axe-core - ウェブサイト及びその他の HTML ベースのユーザーインターフェース向けのアクセシビリティテストエンジンです
- AI -powered ウェブオートメーション
    - browser-use - AI エージェントが自然言語を使用してウェブブラウザーと相互作用できるようにするオープンソース Python ライブラリです
- ウェブスクレイピング
    - Crawlee - ウェブスクレイピング及びブラウザオートメーションライブラリです
    - BeautifulSoup - スクリーン-スクレイピングなど迅速な周期プロジェクト向けに設計された Python ライブラリです
    - Scrapy - ウェブサイトから必要なデータを抽出するためのオープンソース及び協調的なフレームワークです
    - Colly - ウェブスクレイパー構築向けの Golang フレームワークです
    - Katana - 次世代クローリング及びスパイダリングフレームワークです
    - Trafilatura - ウェブからテキストを収集するための Python パッケージ及びコマンドラインツールです

03 - クラウド & クラウドネイティブコンピューティング

クラウドコンピューティング

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

コンピューティング & ストレージ (IaaS)

- Amazon EC2 - クラウド上で安全かつ拡張可能なコンピューティングキャパシティを提供するウェブサービス
- Amazon EBS - Amazon Elastic Compute Cloud との使用に設計された、使いやすく高パフォーマンスなブロックストレージサービス
- Azure Virtual Machines - Windows および Linux 仮想マシンを数秒でプロビジョニングするサービス
- Azure Disk Storage - Azure 仮想マシン向けの高パフォーマンスで耐久性のあるブロックストレージ

- Google Cloud Compute Engine - Google のインフラストラクチャ上で仮想マシンを作成・実行できるカスタマイズ可能なコンピューティングサービス

ネットワーキング

- Amazon VPC - 自分で定義した論理的に分離された仮想ネットワーク内で AWS リソースを起動できるサービス
- Amazon ELB - 受信するアプリケーショントラフィックを Amazon EC2 インスタンス、コンテナ、IP アドレス、Lambda 関数などの複数のターゲットに自動的に分散するサービス
- Azure Virtual Network - Azure におけるプライベートネットワークの基本的な構成要素で、高パフォーマンスのネットワーキングを利用可能
- Azure Load Balancer - バックエンドの仮想マシンにトラフィックを分散させるサービス
- Azure Application Gateway - プラットフォームが管理する、スケーラブルで高可用性のアプリケーションデリバリーコントローラーサービス
- Google Cloud VPC - Andromeda を使用して Google の本番ネットワーク内に実装された物理ネットワークの仮想版
- Cloud Load Balancing - すべてのトラフィックに対応するフルマネージドの分散型ソフトウェア定義サービス

アプリケーションホスティングプラットフォーム (PaaS)

- Azure App Service - ウェブアプリケーション、REST API、モバイルバックエンドをホストするための HTTP ベースのサービス
- AWS Elastic Beanstalk - ウェブアプリケーションとサービスのデプロイおよびスケーリングを簡単に行えるサービス
- Google Cloud App Engine - スケールでウェブアプリケーションを開発・ホストするためのフルマネージドのサーバーレスプラットフォーム
- Vercel - ウェブを構築、デプロイ、スケールするための開発者体験とインフラを提供するフロントエンドクラウドプラットフォーム
- Netlify - グローバルエッジネットワーク上でモダンなウェブ体験を構築、デプロイ、スケールできるコンポーザブルなウェブプラットフォーム
- Coolify - Vercel、Heroku、Netlify、Railway のオープンソース & セルフホスト可能な代替手段

クラウドコマンドラインインターフェース

- AWS CLI - AWS サービスを管理するための統合ツール

- Azure CLI - インタラクティブなコマンドやスクリプトで Azure リソースを管理するクロスプラットフォームのコマンドラインツール
- Azure Developer CLI (azd) - ローカル開発環境から Azure への移行を加速するオープンソースツール
- Google Cloud CLI (gcloud) - Google Cloud のリソースとサービスを作成・管理するためのツールセット
- Vercel CLI - Vercel プロジェクトの管理と設定、アプリのデプロイ、ローカルでのデプロイ環境の複製に使用するコマンドラインインターフェース
- Netlify CLI - 継続的デプロイの設定、ローカル開発サーバーの実行、Netlify へのサイトデプロイを行うコマンドラインインターフェース

クラウドエミュレーター

- LocalStack - オフラインでクラウドおよびサーバーレスアプリを開発・テストするための完全機能のローカルクラウドスタック

プライベートクラウド & オンプレミス IaaS

- OpenStack - パブリッククラウドとプライベートクラウドの構築・管理のために Infrastructure as a Service (IaaS) を提供するオープンソースのクラウドコンピューティングプラットフォーム
- OpenNebula - クラウドおよび仮想化管理、ハイパーバイザー運用、Kubernetes オーケストレーションにわたるエンドツーエンドのカバレッジ、ベンダー中立性、包括的なサポートを提供するエンタープライズクラウド & 仮想化プラットフォーム

クラウドアーキテクチャフレームワーク

- AWS Well-Architected Framework - クラウドでワークロードを設計・実行するための主要な概念、設計原則、アーキテクチャのベストプラクティスを説明するフレームワーク
- Azure Architecture Center - Azure 上で安全で高パフォーマンス、回復力があり効率的なインフラを構築するためのガイダンス、パターン、ベストプラクティスのセット
- Azure Well-Architected Framework - ソリューションアーキテクトがワークロードの技術的基盤を構築するために設計された、品質重視のテナント、アーキテクチャの意思決定ポイント、レビューツールのセット

コードとしての設定

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

Infrastructure as Code (IaC)

- Hashicorp Terraform - インフラストラクチャを安全かつ効率的に構築、変更、バージョン管理できる Infrastructure as Code ツール
- OpenTofu - 安定したドロップイン代替を提供する、インフラの構築と管理のためのオープンソースのコミュニティ主導の Terraform フォーク
- Pulumi - 使い慣れたプログラミング言語とツールを使用してクラウドインフラを構築、デプロイ、管理できる Infrastructure as Code プラットフォーム

AI 駆動インフラ

- Spacelift Intent - 自然言語を使用してクラウドインフラをプロビジョニングおよび管理できる AI 搭載ツール

構成管理 & 自動化

- Ansible - プロビジョニング、構成管理、アプリケーションデプロイ、オーケストレーション、その他多くの IT プロセスを自動化するオープンソースの IT 自動化エンジン
- SaltStack - イベント駆動の IT 自動化、リモートタスク実行、構成管理のための Python ベースのオープンソースソフトウェア
- Rudder - IT インフラ自動化のためのオープンソースの継続的設定 & コンプライアンスプラットフォーム
- cloud-init - クラウドインスタンスをカスタマイズするための標準規格

イメージビルド

- Hashicorp Packer - 単一のソース設定から複数のプラットフォーム向けに同一のマシンイメージを作成するツール

エコシステム & ベンダーツール

- Terraform/OpenTofu エコシステム
  - Terraform/OpenTofu Provider: Core Functions - コア関数を実行するための Terraform/OpenTofu プロバイダー
  - Terragrunt - 設定を DRY に保ち、複数の Terraform モジュールを扱い、リモート状態を管理するための追加ツールを提供するシンラッパー
  - TerraTest - インフラストラクチャのテストのためのパターンとヘルパー関数を提供する Go ライブラリ
  - Atmos - ワークフローをオーケストレーションし、インフラ管理を簡素化する DevOps とクラウドエンジニアリングのためのユニバーサルツール

    - GitLab-managed Terraform/OpenTofu state - Terraform ステートファイルを GitLab に保存できる機能
    - tf.libsonnet - Terraform コードを生成するための Jsonnet ライブラリのコレクション
    - terraform-docs - さまざまな出力フォーマットで Terraform モジュールのドキュメントを生成するユーティリティ
    - Terraformer - 既存のインフラから Terraform ファイルを生成する CLI ツール
    - Atlantis - Webhook 経由で Terraform プルリクエストイベントをリッスンするセルフホスト型の Golang アプリケーション
- ベンダー固有のツール
    - AWS CloudFormation - Amazon Web Services リソースのモデル化とセットアップを支援するサービス
    - AWS CDK - 使い慣れたプログラミング言語でクラウドアプリケーションリソースを定義するオープンソースのソフトウェア開発フレームワーク
    - AWS SAM - サーバーレスアプリケーションを構築するためのオープンソースフレームワーク
    - SST - サーバーレスとイベント駆動アーキテクチャに特化した、AWS 上でフルスタックアプリケーションを簡単に構築できるフレームワーク
    - Azure Resource Manager - Azure のデプロイおよび管理サービス
        - Bicep language - Azure リソースをデプロイするために宣言的な構文を使用するドメイン固有言語 (DSL)
        - Azure Resource Graph - クラウドリソースを大規模にクエリ、探索、分析するための強力な管理ツール

コンテナ化

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- Containerization - オペレーティングシステムレベルの仮想化の一形態
- コンテナ向け Linux ディストリビューション
    - Alpine Linux - musl libc と busybox をベースにしたセキュリティ重視の軽量 Linux ディストリビューション
        - apk-tools - 元々 Alpine Linux 向けに構築されたパッケージマネージャー
    - Fedora CoreOS - コンテナ化されたワークロードを安全かつ大規模に実行するための、自動更新される最小限の OS

    - Flatcar Container Linux - コンテナ向けの不変 Linux ディストリビューション
- コンテナ内のユーティリティ
    - busybox - 多くの一般的な UNIX ユーティリティの小型版を組み合わせた単一の小さな実行ファイル
- 標準規格
    - The Open Container Initiative (OCI) - コンテナフォーマットとランタイムに関するオープンな業界標準を作成することを明確な目的とするオープンガバナンス構造
    - Compose Specification - クラウドおよびプラットフォームに依存しないコンテナベースアプリケーションを定義するための開発者向け標準
    - Development Containers - コンテナを開発固有の設定、ツール、構成で拡張するためのオープン仕様

エンジン & ランタイム

- コンテナエンジン
    - Docker Engine - アプリケーションを構築・コンテナ化するためのオープンソースのコンテナ化技術
        - Docker Rootless mode - Docker デーモンとコンテナを非 root ユーザーとして実行し、潜在的な脆弱性を軽減する機能
    - podman - コンテナと Pod の構築、管理、実行のための強力なコンテナエンジン
        - podman-static - Linux 向けの Alpine ベースのコンテナイメージと静的リンク（rootless）バイナリ
    - WSL Container - Windows 上で Linux コンテナをビルド、実行、管理するための CLI（wslc.exe）と API を提供する、Windows Subsystem for Linux 組み込みのコンテナエンジン
- コンテナランタイム
    - containerd - シンプルさ、堅牢性、ポータビリティを重視した業界標準のコンテナランタイム
        - nerdctl - containerd 向けの Docker 互換 CLI
        - ctr - containerd デーモンと対話するための非公式のデバッグ・管理クライアント
    - CRI-O - OCI (Open Container Initiative) 互換のランタイムを使用できるようにするための Kubernetes CRI (Container Runtime Interface) の実装
        - cri-tools - CRI 向けのツールセット

- OCI ランタイム
  - runc - OCI 仕様に従ってコンテナを生成・実行するための CLI ツール
  - crun - コンテナを実行するための高速で軽量なフル機能の OCI ランタイムおよび C ライブラリ

イメージ管理

- イメージビルドツール
  - Docker Build - Dockerfile とコンテキストから Docker イメージを作成するプロセスを自動化する Docker Engine の一部
  - buildah - Open Container Initiative (OCI) コンテナイメージの構築を容易にするツール
  - podman build - Containerfile または Dockerfile の指示を解釈し、Buildah を基盤として利用して OCI 互換のコンテナイメージを構築するコマンド
  - Kaniko - コンテナまたは Kubernetes クラスター内で Dockerfile からコンテナイメージを構築するツール
  - Cloud Native Buildpacks - Dockerfile なしで、アプリケーションのソースコードをあらゆるクラウドで実行できるイメージに変換するツール
- イメージ検査 & 管理ツール
  - skopeo - コンテナイメージとイメージリポジトリに対してさまざまな操作を実行するコマンドラインユーティリティ
  - dive - Docker イメージ、レイヤーの内容を探索し、Docker/OCI イメージのサイズを縮小する方法を見つけるためのツール
  - regclient - マルチプラットフォームイメージやミラーリングなどの高度な機能をサポートする、OCI レジストリとイメージの管理・検査のためのコマンドラインツールスイート（regctl、regsync、regbot）
- コンテナレジストリ
  - GitLab Container Registry - Docker イメージのための安全なプライベートレジストリ
  - Project Quay - コンテナのビルド、整理、配布、デプロイのために設計されたオープンソースのコンテナネイティブイメージレジストリ
  - Docker Hub - 開発者やチームが Docker コンテナイメージを保存、共有、配布できるクラウドベースのレジストリサービス
  - Amazon ECR - コンテナイメージとアーティファクトの保存、管理、共有、デプロイを簡単に行えるフルマネージドのコンテナレジストリ
  - Azure Container Registry - コンテナイメージと関連アーティファクトを管理する

プライベートレジストリ

- Harbor - ポリシーとロールベースのアクセス制御でアーティファクトを保護するオープンソースレジストリ

環境 & 管理

- コンテナ管理ツール
  - Podman Desktop - コンテナ管理の簡素化、Kubernetes ワークフローの効率化、ローカル開発から本番環境への移行を支援する、開発者がコンテナと Kubernetes を操作するための最も優れた無料のオープンソースツール
  - lazydocker - docker と docker-compose の両方に対応したターミナル UI
  - Docker Compose - マルチコンテナ Docker アプリケーションを定義・実行するためのツール
- 開発環境プロビジョニング
  - devcontainers CLI - devcontainer.json から開発コンテナを作成・設定し、さまざまなインフラストラクチャにわたって開発コンテナの構築、実行、管理コマンドを提供する仕様のリファレンス実装
  - DevPod - オープン標準の devcontainer.json フォーマットを使用して dev-environments-as-code を実現する、ローカルマシン、Kubernetes クラスター、クラウドプロバイダーなどあらゆるインフラをサポートするクライアントのみのツール
- ローカル環境プロビジョナー（Mac 向け）
  - Colima - macOS（および Linux）上で最小限のセットアップでコンテナランタイムを提供するツール
  - Lima - 自動ファイル共有とポートフォワードを備えた Linux 仮想マシンを起動するツール

WebAssembly

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.2 デジタルテクノロジー > その他先端技術

- 標準規格
  - WebAssembly - スタックベースの仮想マシン向けのバイナリ命令フォーマット
  - WebAssembly System Interface (WASI) - WebAssembly のためのモジュラーシステムインターフェース
  - WASIX - 既存の WASI ABI の長期的な安定化とサポート、および追加の非侵略的

なシステムコール拡張

- WebAssembly Component Model - WASI Preview 2 の一部として開発された、型付きの言語非依存インターフェースを通じてモジュールが相互運用できるバイナリインターフェース標準

- ランタイム
  - wazero - Go で記述された唯一のゼロ依存 WebAssembly ランタイム
  - Wasmtime - WebAssembly 向けの高速で安全なランタイム
  - Wasmer - 非常に軽量なコンテナをあらゆる場所で実行できる、高速で安全な WebAssembly ランタイム
  - WasmEdge - クラウドネイティブ、エッジ、分散型アプリケーション向けの軽量で高パフォーマンスかつ拡張可能なランタイム
  - WebAssembly Micro Runtime (WAMR) - 組み込み、IoT、エッジ、Trusted Execution Environment での使用向けに、小さなフットプリントと高度に設定可能な機能を備えた軽量なスタンドアロンランタイム
- ツールチェーン & 言語
  - Emscripten - 速度、サイズ、ウェブプラットフォームに特化した、LLVM を使用した WebAssembly への完全なコンパイラツールチェーン
  - AssemblyScript - WebAssembly 向けの TypeScript に似た言語
  - TinyGo - 組み込みシステムと WebAssembly に言語をもたらす小さな場所向けの Go コンパイラ
  - Binaryen - C++ で記述された WebAssembly 向けのコンパイラとツールチェーンインフラストラクチャライブラリ
- クラウド & エッジプラットフォーム
  - wasmCloud - クラウド、Kubernetes、エッジをまたいで再利用可能な WebAssembly コンポーネントで構成される多言語アプリケーションを構築するためのオープンソース CNCF プロジェクト
  - Fermyon Spin - WebAssembly マイクロサービスとウェブアプリケーションを構築するための開発者ツール

Kubernetes

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- Kubernetes - コンテナ化されたアプリケーションのデプロイ、スケーリング、管理を自動化するオープンソースシステム

- マスターノード
  - kube-apiserver - API サービスを担当
  - kube-scheduler - スケジューリングを担当
  - kube-controller-manager - コンテナオーケストレーションを担当
- コンピュートノード
  - kubelet - そのノード上の Pod の API サーバーを監視し、それらが実行されていることを確認する
  - cAdvisor - 特定のノードで実行中の Pod に関するメトリクスを収集する
  - kube-proxy - ネットワークを最新の状態に保つために、Pod やサービスの変更について API サーバーを監視する
  - container runtime - そのノードでコンテナイメージを管理し、コンテナを実行する役割を担う
- インターフェース標準
  - CNI (Container Networking Interface)
  - CSI (Container Storage Interface)
  - CRI (Container Runtime Interface)

コアコンセプト & コンポーネント

- K8s の内部構造
  - Workloads - クラスター上でコンテナを管理・実行するために使用するオブジェクト
  - Pod
    - assignment - Pod が特定のノードでのみ実行されるように、または特定のノードで実行されることを優先するように制限するプロセス
    - taint and toleration - Pod が不適切なノードに配置されないようにするメカニズム
    - lifecycle - Pod のライフサイクル
    - liveness probe - kubelet がコンテナをいつ再起動するかを判断するために使用するプローブ
    - requests and limits
    - eviction
  - Deployment、ReplicaSet、StatefulSet、DaemonSet
  - Kubernetes network model - Kubernetes クラスターにおけるネットワーキング

の基本的な要件と原則のセット

        - Service、Ingress、Ingress Controllers
    - Storage - ストレージの提供と消費を抽象化する API を備えた強力なボリュームサブシステム
        - PersistentVolume、PVC、StorageClass
    - Configuration - アプリケーションを実行する Pod に設定データを注入できるさまざまなメカニズム
        - Secret、ConfigMap
    - セキュリティ & ポリシー
        - Kubernetes RBAC - 企業内の個々のユーザーの役割に基づいてコンピュータまたはネットワークリソースへのアクセスを制御する方法
        - PodDisruptionBudget - アプリケーションが経験する同時中断の数を制限し、高可用性を実現するオブジェクト
        - Security context - Pod またはコンテナの権限とアクセス制御設定の定義
- オートスケーリング
    - HPA - 観測された CPU 使用率に基づいて、レプリケーションコントローラー、デプロイメント、レプリカセット、またはステートフルセット内の Pod 数を自動的にスケールするコンポーネント
    - Cluster Autoscaler - Kubernetes クラスターのサイズを自動的に調整するツール
    - Karpenter - 柔軟で高パフォーマンスな Kubernetes クラスターオートスケーラー

運用 & 管理

- K8s オペレーター
    - Prometheus Operator - Kubernetes 上で Prometheus クラスターを作成/設定/管理するオペレーター
        - kube-prometheus - エンドツーエンドの Kubernetes クラスター監視を簡単に操作できるよう、ドキュメントとスクリプトを組み合わせた Kubernetes マニフェスト、Grafana ダッシュボード、Prometheus ルールのコレクション
    - Grafana Operator - Kubernetes と OpenShift 環境でカスタムリソースを通じて Grafana インスタンス、ダッシュボード、データソースの管理を効率化する Kubernetes オペレーター
    - OpenTelemetry Operator - OpenTelemetry のための Kubernetes Operator の実装

    - Elastic Cloud on Kubernetes (ECK) - Kubernetes 上の Elastic Stack の公式オペレーター
    - Rook - Kubernetes 向けのオープンソースのクラウドネイティブストレージオーケストレーター
- ダッシュボード
    - k9s - Kubernetes クラスターと対話するためのターミナルベース UI
    - KDash - Rust で構築されたシンプルな Kubernetes ターミナルダッシュボード
    - Seabird - Kubernetes の操作を簡素化するネイティブデスクトップアプリ
    - Headlamp - 拡張性に焦点を当てたユーザーフレンドリーな Kubernetes UI

CLI & ローカル環境

- CLI プラグイン管理
    - Krew - kubectl コマンドラインツールのプラグインマネージャー
        - kubectl-node-shell - ノード上でルートシェルを実行するための kubectl プラグイン
        - kubectl-tree - Kubernetes オブジェクト間の所有関係を探索するための kubectl プラグイン
        - kubectl-pod-inspect - Pod とコンテナのステータスを一目で確認するための kubectl プラグイン
        - kubepug - Kubernetes API のプリフライトチェックツール
        - rakkess - すべての利用可能なリソースのアクセスマトリックスを表示する kubectl プラグイン
        - ketall - すべてのリソースを取得する kubectl プラグイン
- ローカル K8s ツール
    - Minikube - Kubernetes をローカルで実行できるツール
    - Kind - Docker コンテナを「ノード」として使用してローカル Kubernetes クラスターを実行するツール

エコシステム & 拡張機能

- アプリケーションパッケージング & 設定
    - Helm - Kubernetes のパッケージマネージャー
    - Kustomize - kustomization ファイルを通じて Kubernetes オブジェクトをカスタマイズするスタンドアロンツール

    - Artifact Hub - クラウドネイティブのパッケージと設定の検索、インストール、公開を容易にするウェブベースのアプリケーション
- クラウドリソース管理
    - Crossplane - ユーザーがコントロールプレーンで独自の API とサービスを構築し、Kubernetes を拡張してどこでもリソースを管理できるプラットフォームエンジニアリング向けのクラウドネイティブフレームワーク
- 開発者ワークフローツール
    - Skaffold - コンテナベースアプリケーションの継続的な開発を容易にするコマンドラインツール
- プラットフォーム拡張機能
    - kube-fencing - Kubernetes においてステートフルアプリケーションのノードをフェンシングするためのソリューション
    - KubeVirt - Kubernetes 向けの仮想マシン管理アドオン
- オペレーター & コントローラー開発
    - Kubebuilder - カスタムリソース定義 (CRD) を使用して Kubernetes API を構築するためのフレームワーク
- リソース最適化
    - Goldilocks - リソースリクエストと制限の出発点を特定するのに役立つユーティリティ
- ベンダー固有のツール
    - eksctl - Amazon EKS の公式 CLI
- バッチ & ジョブスケジューリング
    - Kueue - バッチ、HPC、AI/ML ワークロード向けにクォータとリソース共有を管理する Kubernetes ネイティブなシステム

クラウドネイティブコンピューティング

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- Serverless Computing - クラウドプロバイダーがオンデマンドでマシンリソースを割り当てるクラウドコンピューティング実行モデルで、サーバーの管理はクラウドプロバイダーが顧客に代わって行う

Container as a Service (CaaS)

- マネージド Kubernetes
  - Google Kubernetes Engine (GKE) - コンテナ化されたアプリケーションを実行するためのマネージドな本番環境
  - Amazon Elastic Kubernetes Service - AWS およびオンプレミスで Kubernetes を簡単に実行できるマネージドサービス
  - Azure Kubernetes Service (AKS) - コンテナ化されたアプリケーションのデプロイと管理のための完全マネージド Kubernetes サービス
- シンプルなコンテナホスティング
  - Amazon Elastic Container Service - コンテナ化されたアプリケーションのデプロイ、管理、スケーリングを簡単に行えるフルマネージドのコンテナオーケストレーションサービス
  - AWS Fargate - ECS と EKS の両方で動作するコンテナ向けサーバーレスコンピューティングエンジン
  - AWS App Runner - 開発者がインフラストラクチャの事前知識なしに、コンテナ化されたウェブアプリケーションと API を大規模かつ迅速にデプロイできるフルマネージドサービス
  - Azure Container Apps - マイクロサービスとイベント駆動ワークロードのために KEDA、Dapr、Envoy を統合した、Kubernetes 上に構築されたフルマネージドのサーバーレスコンテナサービス
  - Google Cloud Run - 自動スケーリングされるコンテナを実行できるマネージドコンピューティングプラットフォーム

Function as a Service (FaaS)

- AWS Lambda - サーバーのプロビジョニングや管理なしに、事実上あらゆる種類のアプリケーションまたはバックエンドサービスのコードを実行できるサーバーレスのイベント駆動型コンピューティングサービス
- Azure Functions - 好みのプログラミング言語を使用してより効率的に開発できるイベント駆動型のサーバーレスコンピューティングプラットフォーム
- Google Cloud Run Functions - クラウドサービスの構築と接続のためのサーバーレス実行環境

高度なランタイム & 分離

- サンドボックス化されたランタイム
  - Kata Containers - コンテナのような感覚とパフォーマンスを持ちながら、仮想マシンのワークロード分離とセキュリティを提供する軽量仮想マシンの標準実装を構

築するオープンソースプロジェクト

  - gVisor - Linux カーネルとそのネットワークスタックを実装し、システムコールをインターセプトしてホストをコンテナ化されたアプリケーションから保護する Linux 互換サンドボックス
  - libkrun - 仮想化ベースのプロセス分離機能を提供する動的ライブラリ
  - Microsoft eXecution Container (MXC) - Windows、Linux、macOS にわたり、複数の封じ込めバックエンドとポリシー駆動のセキュリティ制御によって信頼できないコードを実行する、クロスプラットフォームなサンドボックス化コード実行システム
  - Cloud Hypervisor - 最小限のハードウェアエミュレーションで現代のクラウドワークロードの実行に焦点を当てた、Rust で実装されたオープンソースの仮想マシンモニター (VMM)
  - Firecracker - セキュアなマルチテナントのコンテナおよびファンクションベースのサービスの作成と管理を目的として構築されたオープンソースの仮想化技術
  - AWS Lambda MicroVMs - Firecracker 仮想化技術を基盤に、ほぼ瞬時の起動と再開を備えた VM レベルの分離を提供するサーバーレスコンピューティングサービス
  - QEMU microvm - PCI や ACPI サポートのない最小限のマシンタイプで、短期ゲスト向けに設計され、起動時間とフットプリントの両方に最適化されている
  - Docker Sandboxes - セキュリティとシステム保護を強化するために、軽量なマイクロ VM 上で AI コーディングエージェントを実行するように設計された、隔離された使い捨て環境
  - Daytona - 90ミリ秒未満の作成時間で、隔離されたサンドボックス環境で AI 生成コードを実行するための安全なインフラプラットフォーム
  - Modal - エンジニアが数千の隔離されたサンドボックス上でコンピューティング集約型アプリケーションを構築・スケールするためのサーバーレスクラウドを提供する AI インフラプラットフォーム

- 仮想化 & コンテナストレージ
  - virtiofs - 仮想マシンがホスト上のディレクトリツリーにアクセスできる共有ファイルシステム
- イメージサービス & 配布
  - Nydus - コンテナイメージ、ソフトウェアパッケージなどのクラウドネイティブワークロード向けに高効率なイメージ配布システムを形成する強力なオープンソースのファイルシステムソリューション

クラウドネイティブインフラ

- アプリランタイム & スケーリング
  - KEDA (Kubernetes Event-driven Autoscaling) - 任意のクラスターに追加して、環境で実行されている任意のコンテナにイベント駆動のスケールを提供できる単一目的の軽量コンポーネント
  - Dapr (Distributed Application Runtime) - クラウドとエッジで実行される回復力のあるステートレスおよびステートフルアプリケーションを任意の開発者が構築しやすくし、言語と開発者フレームワークの多様性を受け入れるポータブルなイベント駆動ランタイム
  - V8 isolates - 独自のヒープと独自のガベージコレクターを持つエンジンの独立したインスタンス
- サーバーレスコンピューティング
  - Knative - 現代のサーバーレスワークロードを構築、デプロイ、管理するための Kubernetes ベースのプラットフォーム
- サービスメッシュ & ディスカバリー
  - Gateway API - Kubernetes Ingress、ロードバランシング、サービスメッシュ API の次世代
  - Istio - 既存の分散アプリケーションに透過的にレイヤーされるオープンソースのサービスメッシュ
    - Kiali - Istio 向けのサービスメッシュのオブザーバビリティと設定ツール
  - Linkerd - Kubernetes 向けの超軽量でセキュリティファーストのサービスメッシュ
  - Hashicorp Consul - あらゆるランタイムプラットフォームおよびパブリック/プライベートクラウドにわたってサービスを接続・保護するサービスネットワーキングソリューション
- エッジプロキシ & Ingress
  - Envoy Proxy - オープンソースのエッジ & サービスプロキシ
  - Traefik proxy - 主要なモダンオープンソースのリバースプロキシと Ingress コントローラー
  - Contour - Envoy エッジ & サービスプロキシのコントロールプレーンを提供する Kubernetes 向けの高パフォーマンス Ingress コントローラー
  - Apache APISIX - マイクロサービスの管理、すべての API とマイクロサービスのための究極のパフォーマンス、セキュリティ、スケーラブルなプラットフォームを提供するオープンソース API ゲートウェイ
- クラウドネイティブネットワーキング

    - Project Calico - コンテナ、仮想マシン、ネイティブホストベースのワークロード向けに安全なネットワーク接続、ネットワークセキュリティ、オブザーバビリティを提供するオープンソースプロジェクト
    - Cilium - クラウドネイティブ環境向けのネットワーキング、セキュリティ、オブザーバビリティを提供するオープンソースプロジェクト
    - Flannel - Kubernetes 向けに設計されたレイヤー3ネットワークファブリックを設定するためのシンプルで簡単な方法

CI/CD & GitOps

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > SREプロセス

デリバリー & デプロイメント

- 継続的デリバリーツール
    - Harness - AI と自動化を使用して CI/CD、GitOps、クラウドコスト管理を効率化するモダンなソフトウェアデリバリープラットフォーム
    - Jenkins - 世界中の開発者がソフトウェアを確実に構築、テスト、デプロイできるオープンソースの自動化サーバー
        - Blue Ocean for Jenkins Pipelines - Jenkins のユーザーエクスペリエンスを再考するプロジェクト
        - Python Jenkins - Jenkins REST API の Python ラッパー
        - JenkinsPipelineUnit - Groovy または Declarative で記述された Jenkins パイプラインをユニットテストするためのテストフレームワーク
    - GitLab CI/CD - ソースコードのビルド、統合、検証を自動化するために使用できる GitLab の一部
    - GitHub Actions - すべてのソフトウェアワークフローを自動化しやすくする機能
        - actionlint - GitHub Actions ワークフローファイルの静的チェッカー
        - act - GitHub Actions をローカルで実行するためのツール
    - Azure Pipelines - コードプロジェクトを自動的にビルド・テストし、他のユーザーが利用できるようにするクラウドサービス
- アプリケーションデプロイ
    - Kamal - どこにでもウェブアプリをデプロイするツール

GitOps & クラウドネイティブ

- GitOps スタイルの CD
  - ArgoCD - Kubernetes 向けの宣言的な GitOps 継続的デリバリーツール
  - FluxCD - Kubernetes クラスターを設定のソース（Git リポジトリなど）と同期させ、デプロイする新しいコードがある場合に設定の更新を自動化するツール
- クラウドネイティブアプリケーションデリバリー
  - Open Application Model - あらゆるプラットフォームにデプロイ・管理できるようにアプリケーションを記述するための仕様
  - KubeVela - 今日のハイブリッドなマルチクラウド環境全体へのアプリケーションのデプロイと運用をより簡単、高速、信頼性高くするモダンなソフトウェアデリバリープラットフォーム
  - Flagger - Kubernetes 上で実行されるアプリケーションのリリースプロセスを自動化するプログレッシブデリバリーツール

インテグレーション & レジストリ

- プライベートパッケージレジストリ
  - JFrog Artifactory - ソフトウェアアーティファクトとその依存関係を保存、管理、配布できるユニバーサルな DevOps リポジトリマネージャー
  - GitLab Package Registry - さまざまなサポートされているパッケージマネージャー向けにパッケージを公開・共有できる機能
  - GitHub Packages - ソフトウェアパッケージをプライベートまたはパブリックにホストできるソフトウェアパッケージホスティングサービス
  - Nexus Repository Manager 3 - 高度なリポジトリマネージャー
  - Azure Artifacts - パブリックおよびプライベートソースから Maven、npm、NuGet、Python パッケージフィードを作成・共有できるサービス

システムオブザーバビリティ

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > SREプロセス

インストゥルメンテーション & プラットフォーム

- コンセプト
  - Observability - システムの外部出力に関する知識からシステムの内部状態がどれだけ推測できるかの尺度

- インストゥルメンテーションライブラリ
  - OpenTelemetry - トレース、メトリクス、ログなどのテレメトリデータをインストゥルメント化、生成、収集、エクスポートするためのベンダー中立のオープンソースオブザーバビリティフレームワーク
  - Jaeger - 複雑な分散システムのワークフローを監視・トラブルシューティングするために使用するオープンソースの分散トレーシングプラットフォーム
  - Micrometer - JVM ベースアプリケーション向けのメトリクスインストゥルメンテーションライブラリ
- モニタリングツール
  - Uptime Kuma - 使いやすいセルフホスト型モニタリングツール
- マネージドプラットフォーム
  - Azure Monitor - クラウドおよびオンプレミス環境からのテレメトリを収集、分析、対応するための包括的なソリューション
    - Kusto Query Language - データを探索してパターンを発見し、異常や外れ値を特定し、統計モデルを作成するための強力なツール
    - App Insights - 開発者と DevOps プロフェッショナル向けの Azure Monitor の機能で、拡張可能なアプリケーションパフォーマンス管理 (APM) サービス
  - AWS CloudWatch - DevOps エンジニア、開発者、サイトリライアビリティエンジニア (SRE)、IT マネージャー向けに構築された監視・オブザーバビリティサービス
  - Datadog - 監視 & セキュリティのための統合プラットフォーム
  - Sentry - 開発者が重要なことを確認し、問題をより迅速に解決するためのエラートラッキングとパフォーマンスモニタリングを提供するアプリケーションモニタリングプラットフォーム
- セルフホスト型（上級者向け）
  - SigNoz - ログ、メトリクス、トレース、ダッシュボード、アラートなどのためのオープンソースの Datadog または New Relic 代替
- 可視化ツール
  - Grafana - オープンソースのデータ可視化 & モニタリングソリューション
    - Grafonnet - Grafana ダッシュボードを生成するための Jsonnet ライブラリ
    - gcx - Grafana Cloud、Enterprise、セルフホスト環境全体で、ダッシュボード、アラート、SLO、メトリクス、ログ、トレースといった Grafana リソースへの構造化されたアクセスを提供し、オブザーバビリティ主導の開発ワークフローのために AI コーディングエージェントと統合する CLI ツール

- Kibana - Elasticsearch データを可視化し、Elastic Stack を操作できる無料のオープンユーザーインターフェース

テレメトリ送信

- データシッパー
  - Prometheus exporters - Prometheus メトリクスを公開するサービス
    - node-exporter - *NIX カーネルが公開するハードウェアと OS メトリクスのエクスポーター
    - blackbox-exporter - HTTP、HTTPS、DNS、TCP、ICMP、gRPC 経由でエンドポイントのブラックボックスプロービングを可能にするツール
  - Grafana Alloy - Prometheus パイプラインが組み込まれ、メトリクス、ログ、トレース、プロファイルをサポートするオープンソースの OpenTelemetry コレクター
  - Fluent Bit - 超高速で軽量かつ高度にスケーラブルなログ、メトリクス、トレースのプロセッサーとフォワーダー
  - Fluentd - データの収集と消費を統一してデータの活用と理解を向上させるオープンソースのデータコレクター
  - Filebeat - ログデータの転送と集中化のための軽量シッパー
  - Logstash - 多数のソースからデータを取り込み、変換してから好みの「stash」に送信するオープンソースのサーバーサイドデータ処理パイプライン
  - Telegraf - スタック、センサー、システムからメトリクスを収集するためのオープンソースのサーバーエージェント
  - Metricbeat - オペレーティングシステムとサーバー上で実行されているサービスから定期的にメトリクスを収集するためにサーバーにインストールできる軽量シッパー
  - rsyslog - ログ処理のための高速システム
- ベンダー固有のツール
  - Azure Monitor Agent - Azure およびハイブリッド仮想マシンのゲスト OS から監視データを収集するエージェント
  - Cloudwatch Agent - Amazon EC2 インスタンスとオンプレミスサーバーからシステムレベルのメトリクスとログファイルの両方を収集するエージェント

テレメトリ収集 & ストレージ

- データストア & アラートツール
  - Prometheus - オープンソースのシステム監視 & アラートツールキット

        - PromQL - Prometheus クエリ言語
        - promtool - Prometheus サーバーのコマンドラインユーティリティ
        - Awesome Prometheus Alerts - 90 以上のサービスとエクスポーターをカバーするコピー可能な Prometheus アラートルールのコレクション
    - Alertmanager - Prometheus サーバーなどのクライアントアプリケーションから送信されるアラートを処理するツール
        - amtool - Alertmanager API と対話するための CLI ツール
    - InfluxDB - 高い書き込みおよびクエリ負荷を処理するためにゼロから構築された時系列データベース
        - InfluxQL - InfluxDB のデータと対話するための SQL に似たクエリ言語
        - influx cli - InfluxDB 2.0 のコマンドラインインターフェース
    - Grafana Mimir - Prometheus 向けのオープンソースで水平スケーラブルかつ高可用性のマルチテナントの長期ストレージ
    - Grafana Loki - Prometheus にインスパイアされた水平スケーラブルで高可用性のマルチテナントログ集約システム
        - LogQL - Loki のクエリ言語
            - LogCLI - Loki のコマンドラインインターフェース
    - Grafana Tempo - オープンソースで使いやすく大規模な分散トレーシングバックエンド
        - TraceQL - トレースの選択のために設計されたクエリ言語
    - ElasticSearch - オープンソースの分散型 RESTful 検索 & 分析エンジン、スケーラブルなデータストア、ベクターデータベース
        - Elastic Common Schema - Elastic ユーザーコミュニティのサポートを受けて開発されたオープンソース仕様
        - Ingest pipelines - インデックス作成前にデータに対して一般的な変換を実行できる機能
        - Dissect and Grok - 単一のテキストフィールドから構造化フィールドを抽出できるプロセッサー
    - Graphite - 高度にスケーラブルなリアルタイムグラフシステム
    - Grafana Alerting - データに対するアラートを作成・管理できる機能
    - OpenObserve - モダンアプリケーション向けに設計されたオープンソースのオブザーバビリティプラットフォーム

SRE（サイトリライアビリティエンジニアリング）

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > SREプロセス

- Site Reliability Engineering - ソフトウェアエンジニアリングの側面を取り入れ、インフラストラクチャと運用の問題に適用するディシプリン
  - Service Level Objectives (SLOs) - サービスレベルインジケーター（SLI）によって測定されるサービスレベルの目標値または値の範囲
  - Dickerson's Hierarchy of Service Reliability - 信頼性の高いサービスを構築・維持するために必要な基礎的要素を示すモデルで、ピラミッドとして可視化されることが多い
  - The Four Golden Signals - Google の SRE がユーザー向けシステムの監視に使用する4つの主要なメトリクス（レイテンシ、トラフィック、エラー、サチュレーション）
- Ishikawa diagram - 石川馨が考案した、特定の事象の潜在的な原因を示す因果図

フリート管理 & 運用

- フリート管理
  - AWS Systems Manager - AWS、マルチクラウド、ハイブリッド環境のリソースに対するセキュアなエンドツーエンドの管理ソリューション
  - Azure Automation - Azure および非 Azure 環境全体で一貫した管理をサポートするクラウドベースの自動化 & 設定サービス
    - Azure Update Manager - すべてのマシンの更新を管理・統制するための統合サービス
- バックアップ
  - ベンダー固有のツール
    - AWS Backup - AWS サービス、クラウド、オンプレミス全体のデータ保護を一元化・自動化するフルマネージドサービス
    - Azure Backup - Microsoft Azure クラウドからデータをバックアップし、回復するためのシンプル、セキュア、コスト効率の高いソリューションを提供するサービス
  - K8s 固有のツール
    - Velero - Kubernetes クラスターのリソースと永続ボリュームを安全にバックアップ・リストアし、ディザスタリカバリを実行し、移行するためのオープンソースツール

    - 汎用ツール
        - Barman - オンラインホットバックアップを簡素化することでビジネス継続性を確保するように設計された PostgreSQL データベースのディザスタリカバリソリューション
        - Restic - 高速、安全、効率的なバックアッププログラム
- ランブック自動化
    - RunDeck - データセンターやクラウド環境での日常的な運用手順の自動化を支援するオープンソースの自動化プラットフォーム
- AIOps & 自律エージェント
    - Azure SRE Agent - 監視、診断、インシデント解決の支援によって SRE プラクティスを自動化するために設計された AI 搭載サービス
    - Mezmo Aura - SRE と本番 AI 運用のために特別に設計されたオープンソースのエージェント型ハーネス

カオスエンジニアリング

- コンセプト
    - Chaos Engineering - 本番環境での乱流状態に耐えるシステムの能力への信頼を構築するためにシステムで実験する手法
    - Principles of chaos engineering - カオスエンジニアリングのプラクティスを定義する原則
- カオスエンジニアリングツール
    - Litmus - Kubernetes 向けのクラウドネイティブなカオスエンジニアリングフレームワーク
    - Chaos Mesh - Kubernetes 環境でカオスをオーケストレーションするクラウドネイティブなカオスエンジニアリングプラットフォーム
    - Toxiproxy - カオスと回復力テストのためにネットワークとシステムの状態をシミュレートする TCP プロキシ
    - kubeinvaders - Kubernetes のゲーミファイされたカオスエンジニアリングツール

FinOps

- コンセプト
    - FinOps principles - クラウドの変動費用モデルに財務的説明責任をもたらす文化的プラクティス
- FinOps ツール

- FinOps toolkit - 組織での FinOps 実装のためのツール、リソース、ベストプラクティスのコレクション
- AWS Cost Explorer - コストと使用状況を表示・分析できるツール
- Infracost - Terraform、CloudFormation、その他のインフラストラクチャアズコードプロジェクトのクラウドコスト見積もりを表示するツール
- OpenCost - Kubernetes 支出を監視するためのオープンソースソリューション
- Cloud Custodian - パブリッククラウドのアカウントとリソースを管理するためのルールエンジン

パフォーマンス & 負荷テスト

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > SREプロセス

- Performance Testing - 特定のワークロード下でのシステムの応答性と安定性を評価する手法

パフォーマンステストツール

- Hyperfine - コマンドラインベンチマーキングツール
- Locust - 使いやすい分散型のユーザー負荷テストツール
- Grafana k6 - エンジニアリングチームのパフォーマンステストを簡単かつ生産的にするオープンソースの負荷テストツール
- Gatling - エンジニアリングチームがパフォーマンスの懸念を左にシフトするのを助けるプログラマー向けの負荷テストツール
- Apache Jmeter - 機能的な動作を負荷テストしパフォーマンスを測定するために設計された純粋な Java アプリケーション
- ab - Apache HTTP サーバーをベンチマークするためのツール
- stress-ng - 設定可能な量の CPU、メモリ、I/O、ディスクストレスをシステムに与えるツール
- sysbench - LuaJIT に基づくスクリプト可能なマルチスレッドベンチマークツール
- fio - ユーザーが指定した特定の種類の I/O アクションを実行するスレッドまたはプロセスを多数生成するツール
- iPerf - TCP、UDP、SCTP の究極のスピードテストツール
- Speedtest CLI - Web ブラウザーに依存せず、ダウンロード、アップロード、レイテンシー、パケットロスといったインターネット接続のパフォーマンス指標をネイティブに測定するコマンドラインインターフェース

- plow - Web UI とターミナルインターフェースの両方でリアルタイムのパフォーマンスメトリクスを表示しながら並行接続で負荷テストを実施する、Golang で書かれた HTTP(S) ベンチマークツール
- loadgen-rs - コマンドラインモードと分散モードをサポートする、HTTP/1.1、HTTP/2、HTTP/3 (QUIC) に対応した h2load 互換の Rust 製 HTTP ベンチマーククライアント

04 - セキュリティ・プライバシー

セキュリティ基礎

> **Relevant DSS-P Skills**
>
> - 4. セキュリティ > 4.1 セキュリティマネジメント > セキュリティ体制構築・運営
> - 4. セキュリティ > 4.1 セキュリティマネジメント > セキュリティマネジメント

- Information security - 情報リスクを低減することにより情報を保護する実践です
- Vulnerability - 脅威アクターによって悪用される可能性がある弱点です
- Threat - 脆弱性によって促進される潜在的な悪影響またはイベントです
- Shared Responsibility Model - クラウドサービスプロバイダー（CSP）と顧客がクラウド環境のあらゆる側面を保護する責任を定めるセキュリティ・コンプライアンスフレームワークです

一般的な脅威と攻撃ベクトル

- Malware - コンピュータ、サーバー、クライアント、またはコンピュータネットワークに混乱を引き起こすために意図的に設計されたソフトウェアです
- Ransomware - 被害者の個人データを公開するか、永遠にそのデータへのアクセスをブロックすることで脅迫し、身代金を要求するタイプのマルウェアです
- Social engineering - 人々を行動させたり、機密情報を漏らさせたりするための心理的操作です
- Phishing - 攻撃者が機密情報を明かすようにして人をだまそうとする不正なメッセージを送付するタイプのソーシャルエンジニアリングです
- Business Email Compromise (BEC) - 攻撃者が高管経営者になりすまし、従業員または顧客を騙してお金または機密データを転送させるようにするタイプのフィッシング攻撃です
- Infostealer - システムから情報を収集するように設計されたトロイの木馬のタイプで

す

- Mirai (malware) - Linux を実行しているネットワークデバイスを遠隔制御可能なボットに変え、大規模ネットワーク攻撃で botnet の一部として使用できるマルウェアです
- Think before you Click(Fix) - ユーザーが軽微な技術的問題を解決しようとする傾向を利用して、デバイス上で悪意あるコマンドを実行するように騙すソーシャルエンジニアリング技法です
- Evilginx - ログイン認証情報とセッション Cookie をフィッシングするために使用される中間者攻撃フレームワークです

モダンセキュリティアーキテクチャ

- Zero trust security model - 信頼が暗黙的に付与されることなく、すべてのユーザーに対して検証が必要とされる IT システムの設計と実装に対するアプローチです

セキュリティトレーニングと競技

- Capture the flag (cybersecurity) - 参加者が専門的な知識と技法を使用して隠された「フラグ」（解答）を探し、最高得点を競うハッキング競技です
- プラットフォーム
  - CTFd - 独自のサイバーセキュリティワークショップをホストするための最も簡単なキャプチャザフラッグプラットフォームで、テーマとプラグインで簡単にカスタマイズできる堅牢なベースを提供します

暗号化とデータ保護

> **Relevant DSS-P Skills**
>
> - 4. セキュリティ > 4.2 セキュリティ技術 > セキュア設計・開発・構築

コア暗号化

ハッシング

- ハッシュ関数
  - MD5 - 暗号論的に破られているが、それでも広く使用されている 128 ビットのハッシュ値を生成するハッシュ関数です
  - SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512) - 米国国立安全保障局（NSA）によって設計された暗号ハッシュ関数のセットです
  - Bcrypt - Blowfish 暗号に基づいたパスワード ハッシュ関数です

    - Scrypt - Colin Percival によって作成されたパスワードベースの鍵導出関数です

共通鍵暗号

- Symmetric-key algorithm - 平文の暗号化と暗号文の復号化の両方に同じ暗号鍵を使用する暗号化アルゴリズムです
- ブロック暗号
    - AES - 2001 年に米国国立標準技術研究所（NIST）によって確立された電子データ暗号化の仕様です
- ストリーム暗号
    - Salsa20 ChaCha - Salsa20 のバリエーションで、ラウンドあたりの拡散を増加させながら同じまたはわずかに改善されたパフォーマンスを達成します
- MAC（メッセージ認証コード）
    - HMAC - 暗号ハッシュ関数と秘密の暗号鍵を含むメッセージ認証コード（MAC）の特定のタイプです
- 動作モード
    - CBC (Cipher block chaining) - ブロック暗号の動作モードで、平文のブロックが暗号化される前に前の暗号文ブロックと XOR されます
    - GCM (Galois/Counter Mode) - 対称鍵暗号ブロック暗号の動作モードで、そのパフォーマンスの観点から広く採用されています
    - CCM - 認証と機密性の両方を提供するように設計された暗号ブロック暗号の動作モードです

公開鍵暗号

- Digital signature - デジタルメッセージまたはドキュメントの真正性を検証するための数学的スキームです
- Public-key cryptography - 鍵のペアを使用する暗号システムです
    - RSA - 安全なデータ送信に広く使用される公開鍵暗号システムです
    - EdDSA - ツイスト Edwards 曲線に基づく Schnorr 署名のバリエーションを使用するデジタル署名スキームです
- 鍵合意
    - Diffie-Hellman key exchange - 公開チャネルを介して暗号鍵を安全に交換する方法です
    - Elliptic-curve Diffie-Hellman - 2 者が公開されていないチャネルを介して共有秘密を確立することを可能にする key agreement プロトコルです

- 暗号化スキーム
  - RSAES-PKCS1-v1_5 - PKCS #1 のバージョン 1.5 で最初に標準化された古い暗号化・復号化スキーム（ES）で、脆弱性が知られています
  - RSAES-OAEP - ランダム性を追加し、部分的な復号化を防止することで RSA 暗号化を強化するパディングスキームで、PKCS#1 v2 と RFC 2437 で標準化されました
- 署名スキーム
  - RSASSA-PKCS1-v1_5 - PKCS #1 のバージョン 1.5 で最初に標準化された署名付き署名スキーム（SSA）で、Jager et al.（2018）によると偽造不可と見なされています
  - DSA - デジタル署名のための公開鍵暗号システムおよび連邦情報処理標準で、モジュール指数化と離散対数問題の数学的概念に基づいています
  - ECDSA - 楕円曲線暗号を活用するデジタル署名アルゴリズム（DSA）のバリエーションです
- 鍵フォーマット
  - PKCS #1: RSA Cryptography Specifications - 公開鍵暗号の RSA アルゴリズムの実装の基本的な定義と推奨事項を提供する標準です
  - PKCS #12: Personal Information Exchange Syntax - 単一ファイルに複数の暗号オブジェクトを格納するためのファイル形式です
- 暗号標準とフォーマット
  - Cryptographic Message Syntax - デジタル署名、ダイジェスト、認証、またはあらゆる形式のデジタルデータの暗号化に使用される暗号化スキームとプロトコルのための IETF の標準です

公開鍵基盤（PKI）

- Public Key Infrastructure (PKI) - デジタル認証書を作成、管理、配布、使用、保存、失効させるために必要な役割、ポリシー、ハードウェア、ソフトウェア、および手順のセットです
- Certificate authority (CA) - デジタル認証書を保存、署名、および発行するエンティティです
- 登録機関（RA）
- 検証機関（VA）
- プロトコルと標準
  - Simple Certificate Enrollment Protocol - X.509 認証書を安全かつ自動化された方法で登録するためのプロトコルです

- 検証と登録
  - Domain Control Validation - 認証局（CA）が認証書を要求している個人または組織が認証書にリストされているドメインを制御しているかを検証するために使用されるプロセスです
- トラストストア
  - Certifi - TLS ホストの ID を検証しながら SSL 認証書の信頼性を検証するための慎重に厳選されたルート認証書のコレクションです
- Public key certificate - 公開鍵の有効性を証明するための電子ドキュメントです
  - ドメイン検証（DV）
  - 組織検証（OV）
  - 拡張検証（EV）
- Let’s Encrypt - TLS 認証書を提供する非営利認証局です
  - certbot - Let’s Encrypt 認証書を手動で管理する Web サイトで自動的に使用できるようにするための無料のオープンソースソフトウェアツールです
  - lego - Go で書かれた Let’s Encrypt クライアントおよび ACME ライブラリです
- ACME (Automatic Certificate Management Environment) - 認証局とそのユーザーの Web サーバー間の相互作用を自動化するための通信プロトコルです
- mkcert.org - ローカルで信頼された開発認証書を作成するためのシンプルなツールです
- cert-manager - Kubernetes と OpenShift 用の強力で拡張可能な X.509 認証書コントローラーです
- cfssl - Cloudflare の PKI ツールキットです

シークレット管理

- Vault - API キー、パスワード、認証書などのシークレットに安全にアクセスするためのツールです
- OneCLI - AI エージェント用のオープンソース認証情報コンテナとプロキシゲートウェイで、シークレットを暗号化コンテナに保存し、鍵を公開することなくエージェント要求に挿入します
- SOPS - YAML、JSON、ENV、INI、BINARY フォーマットをサポートする暗号化ファイルエディターです
- git-secret - git リポジトリ内に機密データを保存するための bash ツールです
- Kubernetes エコシステム

    - Sealed Secrets - 単方向暗号化されたシークレット用の Kubernetes コントローラーとツールです
    - Secrets Store CSI Driver - Kubernetes がエンタープライズグレード外部シークレットストアに保存されている複数のシークレット、鍵、認証書をポッドにボリュームとしてマウントできるようにするドライバーです
    - External Secrets Operator - AWS Secrets Manager、HashiCorp Vault、Google Secrets Manager、Azure Key Vault、IBM Cloud Secrets Manager などの外部シークレット管理システムを統合する Kubernetes オペレーターです
- ベンダーサービス
    - Azure Key Vault - クラウドアプリとサービスで使用される暗号鍵と他のシークレットを保護するためのクラウドサービスです
    - Google Cloud Secret Manager - API キー、パスワード、認証書、およびその他の機密データのための安全で便利なストレージシステムです
    - AWS Key Management Service - 暗号鍵を簡単に作成・管理できるサービスです
    - AWS Secrets Manager - アプリケーション、サービス、IT リソースへのアクセスを保護するためのシークレット管理サービスです

応用暗号化とツール

- 高度な暗号化トピック
    - Post-quantum cryptography - 量子コンピュータによる暗号解析攻撃に対して安全であると考えられている暗号化アルゴリズムです
    - 情報隠蔽
        - Steganography - ファイル、メッセージ、画像、またはビデオを別のファイル、メッセージ、画像、またはビデオ内に隠す実践です
        - Digital watermarking - オーディオ、ビデオ、画像データなどのノイズに耐性のある信号に密かに埋め込まれたマーカーの一種です
- エンドツーエンド暗号化ツール
    - age - シンプルで最新で安全なファイル暗号化ツール、フォーマット、Go ライブラリです
    - Pretty Good Privacy (PGP) - データ通信の暗号化プライバシーと認証を提供するデータ暗号化・復号化コンピュータプログラムです
    - OpenPGP - 公開鍵と暗号化されたメッセージを交換するための非所有プロトコルです
    - keys.openpgp.org - OpenPGP 用の公開鍵サーバーです
    - GnuPG - PGP 暗号化ソフトウェアスイートの無料ソフトウェア置き換えです

    - Gpg4win - OpenPGP の助けを借りてメールとファイルの安全な転送を容易にする Windows ソフトウェアパッケージです
- 暗号化ライブラリ
    - PyCryptodome - 暗号化プリミティブの自己完結した Python パッケージです
    - Python cryptography - 暗号化プリミティブとレシピを Python 開発者に公開するために設計されたパッケージです
    - Go Cryptography - Go 暗号化ライブラリのコレクションです
    - Botan - C++ で書かれた暗号化ライブラリです

アイデンティティとアクセス管理（IAM）

> **Relevant DSS-P Skills**
>
> - 4. セキュリティ > 4.2 セキュリティ技術 > セキュア設計・開発・構築
> - 2. データ整備・活用 > 2.3 データマネジメント > データの品質・安全性向上

統合 IAM

- Identity management - エンタープライズ内の適切な人々が技術リソースへの適切なアクセスを持つようにするためのポリシーとテクノロジーのフレームワークです
- セルフホスト IAM プラットフォーム
    - FusionAuth CE - FusionAuth のセルフホスト型コミュニティサポート版です
    - KeyCloak - オープンソースのアイデンティティとアクセス管理ソリューションです
    - FreeIPA - Linux、389 Directory Server、MIT Kerberos、NTP、DNS、認証システムを組み合わせた統合セキュリティ情報管理ソリューションです
- クラウド IAM サービス
    - Microsoft Entra ID - クラウドベースのアイデンティティとアクセス管理サービスです
    - AWS IAM - AWS リソースへのアクセスを安全に制御するのに役立つサービスです
    - Amazon Cognito - Web およびモバイルアプリにユーザーサインアップ、サインイン、アクセス制御を追加できるサービスです
    - Auth0 - アプリケーションに認証・認可サービスを追加するための柔軟で組み込みが簡単なソリューションです
- Directory service - ネットワークリソースの名前をそれぞれのネットワークアドレスにマッピングするサービスです

    - LDAP - 分散ディレクトリ情報サービスにアクセスし、保守するためのオープンで独立したベンダー中立のインダストリスタンダード応用プロトコルです
    - OpenLDAP - Lightweight Directory Access Protocol のオープンソース実装です
    - 389 Directory Server - Linux システム用に Red Hat によって開発された無料でオープンソースのソフトウェアプロジェクトです
- 仕様
    - Decentralized Identifiers (DIDs) - 検証可能で分散的なデジタル ID を有効にする新しいタイプの識別子です
    - System for Cross-domain Identity Management (SCIM) - クラウドベースのアプリケーションとサービスでのユーザーアイデンティティ管理を簡単にするために設計された仕様です

認証（AuthN）

- Authentication - コンピュータシステムユーザーのアイデンティティなど、アサーションを証明する行為です
- Mutual authentication - 通信リンク内の両者が互いに認証するプロセスです
- Multi-factor authentication (MFA) - アクセスのために複数の検証方法を必要とする方法です
- 3-D Secure - オンラインクレジットカードと デビットカード取引のための追加のセキュリティレイヤーであるように設計されたセキュリティプロトコルです
- Single sign-on (SSO) - 複数のアプリケーションに対して 1 回のログインを許可するサービスです
- プロトコルと標準
    - OpenID Connect - OAuth 2.0 プロトコルの上にシンプルなアイデンティティレイヤーです
    - SAML - ユーザーをアプリケーションにログインさせるための標準です
    - WS-Federation - 参加している Web サービス間の信頼を仲介し、アイデンティティ、属性、認証を管理するためのメカニズムを定義する仕様です
    - FIDO2 (WebAuthn、CTAP、Passkeys) - ユーザーが一般的なデバイスを利用して、オンラインサービスに簡単かつ安全に認証することを可能にする仕様のセットです
        - 依存当事者 - ユーザーのアイデンティティを検証したい Web サイトまたはオンラインサービス（例：お客様の銀行の Web サイト）です
        - 認証器 - 暗号鍵を安全に保存し、ユーザーの認証を実行するデバイスまたはソフトウェアです

        - クライアント - ユーザーのデバイス上のソフトウェア（通常は Web ブラウザーまたはオペレーティングシステムコンポーネント）で、依存当事者と認証器の間を通信します
        - WebAuthn - 公開鍵認証情報にアクセスするための API です
        - CTAP - 外部認証器がクライアントプラットフォームと通信することを可能にするプロトコルです
        - Passkeys - パスワードのためのフィッシング耐性置き換えです
    - SPIFFE - Secure Production Identity Framework for Everyone です
    - Kerberos - チケットの概念に基づいて動作するコンピュータネットワーク認証プロトコルです
    - SSPI (Security Support Provider Interface) - アプリケーションがセキュリティシステムへのインターフェースを変更することなく、コンピュータまたはネットワークで利用可能なさまざまなセキュリティモデルを使用できるようにする Win32 API です
- 認証情報とトークン
    - Basic authentication - HTTP ユーザーエージェントがリクエストを行う際にユーザー名とパスワードを提供する方法です
    - JSON Web Token (JWT) - オプションの署名およびまたはオプションの暗号化を持つデータを作成するためのインターネット標準で、ペイロードはいくつかの主張を主張する JSON を保持します
        - nodejs jsonwebtoken - Node.js 用の JSON Web Token の実装です
    - TOTP (Time-Based One-Time Password) - コンピュータシステムへのアクセスを認証するために使用される、アルゴリズムによって生成された一時的なパスコードです
    - AWS Signature Version 4 (SigV4) - HTTP で送信された AWS API リクエストに認証情報を追加するプロセスです
- プラットフォームとツール
    - Dex - フェデレーション OpenID Connect プロバイダーです
    - Firebase Authentication - ユーザーをアプリに認証するためのバックエンドサービス、使いやすい SDK、既成の UI ライブラリを提供するサービスです
    - Supabase Auth - Supabase プロジェクトのユーザー管理とアクセス制御を提供するサービスです
    - ReCAPTCHA - Web ホストが人間とコンピュータによる自動アクセスを Web サイトに区別できるようにする CAPTCHA システムです
    - Microsoft Authentication Library (MSAL) - 開発者が認証・認可をアプリケーショ

ンに統合するのに役立つライブラリです

- Application Default Credentials (ADC) - Google Cloud クライアントライブラリがアプリケーション環境に基づいて認証情報を自動的に見つけるために使用するメカニズムです
- Limen - Go 用の軽量でコンポーザブルな認証・認可ライブラリで、セッション、パスワードハッシング、OAuth、CSRF 保護を提供します

認可（AuthZ）

- Authorization - リソースへのアクセス権・特権を指定する機能です
- アクセス制御モデル
  - Access control list (ACL) - システムリソースに関連する権限のリストです
  - Attribute-based access control (ABAC) - ユーザー属性に基づいてアクセスを許可するモデルです
  - Discretionary access control (DAC) - ユーザーが自分のリソースへのアクセスを制御することを可能にするモデルです
  - Mandatory access control (MAC) - セキュリティラベルに基づいてアクセスポリシーを適用するモデルです
  - Role-based access control (RBAC) - 役割と特権の周りに定義されたポリシー中立的なアクセス制御メカニズムです
    - Azure RBAC - Azure リソースのきめ細かいアクセス管理を可能にするシステムです
      - セキュリティプリンシパルの種類：ユーザー、グループ、サービスプリンシパル、マネージド ID
    - Entra ID RBAC - Microsoft Entra リソースのきめ細かいアクセス管理を提供するシステムです
- プロトコルと標準
  - OAuth 2.0 Authorization Framework - アクセスデリゲーションのためのオープンスタンダードです
    - リソース所有者 - アクセスされているデータまたはリソースを所有するユーザーです
    - リソースサーバー - 保護されたリソースをホストするサーバーです
    - クライアント - リソース所有者に代わってリソースにアクセスしたいアプリケーションまたはサービスです
    - 認可サーバー - クライアントにアクセストークンを発行するサーバーです

- プラットフォームとツール
  - Permify - Golang API を使用してあらゆる種類の認可システムを作成するのに役立つオープンソース認可サービスです
  - Azure Shared Access Signature (SAS) - 1 つ以上のストレージリソースを指し、アクセスの権限とインターバルを指定するトークンを含む署名付き URI です

セキュアな開発ライフサイクル（DevSecOps）

> **Relevant DSS-P Skills**
>
> - 4. セキュリティ > 4.2 セキュリティ技術 > セキュア設計・開発・構築

セキュアな設計とモデリング

- Threat modeling - 潜在的な脅威を識別、列挙、優先順位付けできる仮想の攻撃者の観点からのプロセスです
- OWASP Threat Modeling - OWASP セキュリティカルチャープロジェクトの開発ライフサイクルへの脅威モデリング統合に関するガイダンスです
- STRIDE model - コンピュータセキュリティ脅威を 6 つのカテゴリに分類するニーモニックです
- MITRE ATT&CK - 敵の戦術と技法のグローバルアクセス可能なナレッジベースです
  - 戦術：攻撃中に敵が達成することを目指す高レベルの目標または目的です
  - 技法：敵が戦術的目標を達成するための特定のメソッドまたは方法です
  - 手順：敵が操作で利用する技法の特定の実装またはバリエーションです
- モデリングツール
  - OWASP Threat Dragon - 無料でオープンソースのクロスプラットフォーム脅威モデリングアプリケーションです
  - threatspec - 脅威モデルをコードとして定義できるツールです

セキュアな開発実践

- Secure Software Development Framework (SSDF) - 基本的で健全でセキュアなソフトウェア開発実践のセットです
- Microsoft Security Development Lifecycle (SDL) - 開発者がより安全なソフトウェアを構築し、セキュリティコンプライアンス要件に対応しながら開発コストを削減するのに役立つソフトウェア開発プロセスです
- OWASP Application Security Verification Standard (ASVS) - アプリケーションレベルのセキュリティ検証を実行するための標準です

- OWASP Security Champions - 開発チーム内にセキュリティの専門知識とカルチャーを組み込むためのプログラムです
- OWASP Cheat Sheet Series - さまざまなセキュリティトピックに関する簡潔なチートシートのコレクションです
- OWASP LLM Top 10 - 開発、デプロイ、管理ライフサイクル全体にわたり、生成 AI と大規模言語モデルアプリケーションの開発とセキュリティに関する上位 10 のリスク、脆弱性、軽減策のガイドです
- コーディング標準
  - MISRA C - 安全クリティカル組み込みシステム（元々自動車産業向けに開発）における C プログラミング言語の使用に関するガイドラインのセットです
  - CERT Secure Coding Standards - Carnegie Mellon の CERT によって発行された C、C++、Java、Perl その他の言語のプログラミングセキュリティガイドラインのコレクションです

Web アプリケーションセキュリティ

- セキュリティメカニズムとポリシー
  - SOP (Same-origin policy) - Web アプリケーションセキュリティモデルにおける重要な概念です
  - CORS (Cross-Origin Resource Sharing) - Web ページ上の制限されたリソースを別のドメインからリクエストできるようにするメカニズムです
  - CSP (Content Security Policy) - クロスサイトスクリプティング（XSS）およびデータインジェクション攻撃を含む特定の種類の攻撃を検出・軽減するのに役立つセキュリティの追加レイヤーです
  - HSTS (HTTP Strict Transport Security) - プロトコルダウングレード攻撃と Cookie ハイジャックから Web サイトを保護するのに役立つ Web セキュリティポリシーメカニズムです
  - Cross-origin isolation - Web ページが SharedArrayBuffer と performance.measureUserAgentSpecificMemory()などの強力な機能を使用できるようにする Web セキュリティ機能です
- 一般的な脆弱性と攻撃
  - Cross-site request forgery (CSRF) - Web アプリケーションが信頼するユーザーから不正なコマンドが送信される Web サイトの悪意あるエクスプロイトです
  - Cross-site scripting (XSS) - 通常は Web アプリケーションで見られるセキュリティ脆弱性のタイプです
  - DNS rebinding - 悪意のある Web ページが Domain Name System を悪用することで、同一生成元ポリシーをバイパスできる攻撃のタイプです

    - SSRF (Server-side request forgery) - 攻撃者がサーバー上の機能を悪用して、内部リソースを読み取り・変更できるエクスプロイトのタイプです
- プライバシーと透明性
    - Privacy sandbox - ユーザーのプライバシーを保護し、企業と開発者に栄える デジタルビジネスを構築するためのツールを提供する Google のイニシアティブです
    - security.txt - Web サイトがセキュリティ研究者のためのセキュリティポリシーを定義できるようにする提案された標準です

アプリケーションセキュリティテスト（AST）

- 統合セキュリティプラットフォーム
    - GitHub Advanced Security - シークレット保護でリークを開始する前に停止し、コードセキュリティでコード内の脆弱性を修正する、開発速度で移動するセキュリティを提供する Suite です
- 静的解析（SAST）
    - SonarQube Server - 自動コードレビューツールで、クリーンコード配信を体系的にサポートします
    - GitLab SAST - ソースコードの既知の脆弱性をチェックするツールです
    - Bandit (Python の場合) - Python コード内の一般的なセキュリティの問題を見つけるために設計されたツールです
    - Semgrep OSS - バグを見つけ、コード標準を強制するための高速でオープンソース、静的解析ツールです
    - Fluid attacks - ソースコード、コンテナ、依存関係の脆弱性を見つけることができるセキュリティツールです
    - CodeQL - コードをデータのようにクエリして脆弱性とそのバリエーションを見つけることができるセマンティックコード解析エンジンです
- 動的解析（DAST）
    - ZAP - 世界中で最も広く使用されている Web アプリスキャナーで、無料でオープンソースであり、誰もが貢献できるコミュニティベースの GitHub Top 1000 プロジェクトです
    - Nuclei - グローバルセキュリティコミュニティによって駆動され、シンプルな YAML ベースの DSL に基づいて構築された高速でカスタマイズ可能な脆弱性スキャナーです
    - sqlmap - SQL インジェクションの欠陥を検出・悪用し、データベースサーバーを引き継ぐプロセスを自動化するオープンソースペネトレーションテストツールです
- シークレット検出

  - GitLab Secret Detection - リポジトリ履歴でシークレットをスキャンするツールです
  - Gitleaks - git リポジトリ内のパスワード、api キー、トークンなどの硬いシークレットを検出・防止する SAST ツールです
  - secretlint - 認証情報のコミットを防止するためのプラグイン可能なリントツールです
  - Talisman - 機密情報またはセンシティブ情報がコミットされないようにリポジトリにフックをインストールするツールです
  - TruffleHog - 環境をスキャンしてシークレットを発掘し、コミット履歴とブランチに深く掘り下げるツールです
  - Whispers - ハードコードされた認証情報の検索で、さまざまな一般的なデータ形式を解析するために設計された静的コード解析ツールです
- AI オーケストレーション型ペネトレーションテスト
  - PentestGPT - 大規模言語モデル（LLM）によって駆動される自動ペネトレーションテストフレームワークです
  - PentAGI - 自動セキュリティテスト用に設計された完全に自律的な AI エージェントシステムです
  - Strix - 実際のハッカーのように動作する自律型 AI エージェントのセットで、コードを動的に実行し、脆弱性を見つけ、実際の概念実証を通じて検証します
  - CAI - セキュリティプロフェッショナルが AI による攻撃・防御自動化を構築・デプロイできるようにする軽量でオープンソースのフレームワークです
  - HexStrike AI - AI エージェントが自律的に 150 以上のサイバーセキュリティツールを実行して、自動ペネトレーション、脆弱性発見、バグ報奨自動化、セキュリティ研究を行うことができる高度な MCP サーバーです
  - Zen AI Pentest - 最先端の言語モデルとプロフェッショナルセキュリティツールを組み合わせた自律的な AI 駆動型ペネトレーションテストフレームワークです
  - Project Glasswing - 防御的なサイバーセキュリティのための Anthropic の Claude Mythos Preview モデルへのアクセスを提供するイニシアティブで、重要なソフトウェアインフラストラクチャの脆弱性を自律的に発見できます

インフラストラクチャアズコード（IaC）セキュリティ

- Trivy - 包括的で汎用のセキュリティスキャナーです
- Defender for Cloud CLI - CI/CD パイプライン内のセキュリティスキャンをオーケストレーションし、結果を Microsoft Defender for Cloud にアップロードして姿勢管理と優先順位付けを行う開発者ファーストコマンドラインツールです

- checkov - インフラストラクチャアズコード（IaC）ファイルの設定ミスをスキャンするための静的コード解析ツールです
- Haskell Dockerfile Linter - ベストプラクティスの Docker イメージを構築するのに役立つスマート Dockerfile linter です
- kube-score - Kubernetes オブジェクト定義の静的コード解析を実行するツールです
- kubesec - Kubernetes リソースのセキュリティリスク解析です
- PSRule - インフラストラクチャアズコード（IaC）をテストおよび検証するコマンドを含む クロスプラットフォーム PowerShell モジュールです
  - PSRule for Azure - PSRule を使用して Azure リソースとインフラストラクチャアズコード（IaC）を検証するルールのスイートです
- ComplianceAsCode - SCAP、Bash、Ansible などのさまざまな形式でセキュリティ自動化コンテンツを提供するプロジェクトです
- complyctl - OSCAL を使用してコンプライアンス評価活動を合理化するコマンドラインツールです

ソフトウェアサプライチェーンセキュリティ（SSCS）

- コンポジション解析（SCA）
  - SBOM 生成
    - Syft - コンテナイメージとファイルシステムからソフトウェア部品表（SBOM）を生成するための CLI ツールおよび Go ライブラリです
    - OWASP CycloneDX format - アプリケーションセキュリティコンテキストでの使用を目的とした軽量ソフトウェア部品表（SBOM）標準です
    - SPDX format - ソフトウェア部品表（SBOM）情報を通信するためのオープンスタンダードです
  - 脆弱性スキャン
    - Grype - コンテナイメージとファイルシステムの脆弱性スキャナーです
    - OSV-scanner - OSV の公式脆弱性スキャナーです
    - Safety - インストール済みの依存関係の既知のセキュリティ脆弱性をチェックするツールです
    - Clair - アプリケーションコンテナの脆弱性の静的解析のためのオープンソースプロジェクトです
    - GitLab Container Scanning - Docker イメージの既知の脆弱性をチェックするツールです
    - JFrog Xray - ソフトウェアサプライチェーン全体にわたり脆弱性とライセンスコンプライアンスの問題を識別するアプリケーションセキュリティツールです

    - ライセンスと依存関係解析
        - Feluda - Python プロジェクトの高速依存関係グラフジェネレーターです
    - 自動化された依存関係更新
        - Dependabot Core - Dependabot セキュリティ・バージョン更新の中心となるライブラリです
- フレームワークとアセスメント
    - SLSA framework - タンパリング防止、整合性向上、パッケージとインフラストラクチャのセキュリティを目指す標準とコントロールのセキュリティフレームワークです
    - in-toto - ソフトウェアサプライチェーン整合性を保護するためのフレームワークです
    - OpenSSF Scorecard - ソフトウェアセキュリティと関連する多数の重要な ヒューリスティック（「チェック」）を評価し、各チェックに 0〜10 のスコアを割り当てる自動ツールです
- プロヴェナンスとアーティファクトメタデータ
    - GUAC - ソフトウェアセキュリティメタデータを高忠実度グラフデータベースに集約するオープンソースツールです
- セキュアな配布と更新
    - The Update Framework (TUF) - ソフトウェア更新システムを保護するためのフレームワークで、リポジトリまたは署名鍵を侵害する攻撃者に対する保護も提供します
- コード署名と整合性
    - Sigstore (Fulcio、Rekor、Cosign) - ソフトウェアの署名、検証、保護のための新しい標準です
- 注目すべき攻撃
    - Shai-Hulud npm Supply Chain Attack - post-install スクリプトを使用してセンシティブデータを収穫し、到達可能なパッケージの悪意あるバージョンを自動的に公開してさらに拡散する自己伝播型ワームです

ランタイムと運用セキュリティ

> **Relevant DSS-P Skills**
>
> - 4. セキュリティ > 4.1 セキュリティマネジメント > インシデント対応と事業継続
> - 4. セキュリティ > 4.2 セキュリティ技術 > セキュリティ運用・保守・監視

クラウドネイティブアプリケーション保護（CNAPP）

- The 4 Cs of Cloud-Native Systems - クラウドネイティブアプリケーションの多層保護を提供するために、セキュリティ戦略を 4 つの異なるレイヤーに分割する多層防御アプローチです
- Microsoft Defender for Cloud - クラウドベースのアプリケーションを保護するために設計されたセキュリティ対策とプラクティスのセットを備えたクラウドネイティブアプリケーション保護プラットフォーム（CNAPP）です
- クラウドセキュリティ姿勢管理（CSPM）
  - AWS Security Hub - セキュリティベストプラクティスチェックを実行し、アラートを集約し、自動修復を可能にするクラウドセキュリティ姿勢管理（CSPM）サービスです
  - cnquery - インフラストラクチャ全体をデータとしてクエリできるクラウドネイティブのグラフベースセキュリティツールです
- クラウドワークロード保護プラットフォーム（CWPP）
  - Amazon Inspector - AWS にデプロイされたアプリケーションのセキュリティとコンプライアンスを向上させるのに役立つ自動化されたセキュリティ評価サービスです
  - Falco - クラウドネイティブランタイムセキュリティプロジェクトです
  - Tracee - Linux のための強力なランタイムセキュリティとフォレンジックスツールです
  - ClamAV - トロイの木馬、ウイルス、マルウェア、その他の悪意のある脅威を検出するためのオープンソースアンチウイルスエンジンです
  - YARA - マルウェア研究者のためのパターンマッチングスイスアーミーナイフです

セキュリティ運用と監視（SecOps）

- 検出と対応
  - Endpoint detection and response (EDR) - 高度な脅威への継続的な監視と対応の必要性に対処するサイバーセキュリティテクノロジーです
  - Extended detection and response (XDR) - SaaS ベースのベンダー固有のセキュリティ脅威検出とインシデント対応ツールです
  - Managed detection and response (MDR) - 脅威ハンティングサービスを組織に提供し、脅威が発見されたら対応する外部委託サービスです
- SIEM と SOAR
  - Security orchestration, automation and response (SOAR) - 組織がセキュリティ脅威に関するデータを収集できるようにする互換性のあるソフトウェアプログラム

のスタックです

- Microsoft Sentinel - スケーラブルで、クラウドネイティブで、セキュリティ情報イベント管理（SIEM）およびセキュリティオーケストレーション自動化応答（SOAR）ソリューションです
- Amazon GuardDuty - 悪意のある活動と不正な行動を継続的に監視する脅威検出サービスです

- 検出と監査
  - Sigma Detection Format - 関連するログイベントを簡潔に記述できる汎用でオープンな署名フォーマットです
  - AWS CloudTrail - AWS アカウントの運用およびリスク監査、ガバナンス、コンプライアンスを有効にするのに役立つ AWS サービスです
  - AWS Config - AWS リソースの構成を評価、監査、確認できるサービスです

ポリシー適用

- Open Policy Agent (OPA) - スタック全体のポリシー適用を統一するオープンソース汎用ポリシーエンジンです
  - Rego - OPA のポリシーを作成するために使用される高レベル宣言型言語です
  - Conftest - 構造化された構成データに対してテストを書くのに役立つユーティリティです
- クラウドポリシーエンジン
  - Azure Policy - リソースガバナンスと一貫性を備えたリアルタイムでクラウドコンプライアンスを実現するサービスです
- Kubernetes ポリシーエンジン
  - Gatekeeper - Open Policy Agent（OPA）によって実行されるポリシーを強制する、カスタマイズ可能な検証 webhook です
  - Kyverno - Kubernetes 用に設計されたポリシーエンジンです

デジタルフォレンジックスとインシデント対応（DFIR）

- 概念
  - Computer security incident management - コンピュータまたはコンピュータネットワーク上のセキュリティイベントの監視と検出、およびそれらのイベントへの適切な対応の実行です
  - Digital forensics - モバイルデバイスとコンピュータ犯罪に関連することが多い、デジタルデバイスで見つかった物質の回復、調査、検査、分析を含むフォレンジック科学の一分野です

    - Computer forensics - コンピュータおよびデジタルストレージメディアで見つかった証拠に関連するデジタルフォレンジック科学の一分野です
- ツールとプラットフォーム
    - Volatility - 揮発性メモリ（RAM）サンプルからデジタル成果物を抽出するための世界で最も広く使用されているフレームワークです
    - Autopsy - The Sleuth Kit およびその他のデジタルフォレンジックスツールのデジタルフォレンジックスプラットフォームおよびグラフィカルインターフェースです

セキュアな通信とネットワーキング

> **Relevant DSS-P Skills**
>
> - 4. セキュリティ > 4.2 セキュリティ技術 > セキュア設計・開発・構築

トランスポート層セキュリティ（TLS）

- Transport Layer Security (TLS) - コンピュータネットワーク上の通信セキュリティを提供するために設計された暗号プロトコルです
- Server Name Indication (SNI) - Transport Layer Security（TLS）コンピュータネットワーキングプロトコルへの拡張です
- ツールとライブラリ
    - testssl.sh - サーバーのサービスがポートで TLS/SSL 暗号とプロトコルのサポートをチェックする無料コマンドラインツールです
    - OpenSSL library - ネットワーク盗聴に対するコンピュータネットワーク上のセキュアな通信のためのアプリケーション用ソフトウェアライブラリです
    - stunnel - 既存のクライアントとサーバーに TLS 暗号化機能を追加するように設計されたプロキシです
    - Squid SSL Bump - Squid プロキシの機能で、SSL/TLS トラフィックを傍受、復号化、再暗号化します
- 脆弱性
    - Lucky Thirteen attack - TLS プロトコルに対するタイミング攻撃で、攻撃者が暗号文を復号化できます

セキュアシェル（SSH）

- Secure Shell (SSH) - セキュリティで保護されていないネットワーク上でネットワークサービスを安全に運用するための暗号化ネットワークプロトコルです
- ツールとライブラリ

- OpenSSH - SSH プロトコルでリモートログインするための随一の接続ツールです
- PuTTY - Windows および Unix プラットフォーム用の SSH と Telnet の無料実装です
- ssh-audit - SSH サーバーとクライアントの構成監査用ツールです
- keychain - 通常は ~/.bash_profile から開始される ssh-agent マネージャーです

ファイアウォールとネットワーク保護

- Web アプリケーションファイアウォール（WAF）
  - AWS WAF - Web アプリケーションまたは API を一般的な Web エクスプロイトと bot から保護するのに役立つ Web アプリケーションファイアウォールです
  - Azure Web Application Firewall - 一般的な Web ハッキング技法と脆弱性から Web アプリを保護するクラウドネイティブサービスです
- ネットワークレベルの保護
  - AWS Shield - AWS 上で実行されているアプリケーションを保護する分散サービス拒否（DDoS）保護サービスです
  - Azure DDoS Protection - 最も洗練された DDoS の脅威に対する対策を提供するサービスです
  - Fail2ban - コンピュータサーバーをブルートフォース攻撃から保護する侵入防止ソフトウェアフレームワークです
  - Snort (IPS) - 世界中で最初のオープンソース侵入防止システム（IPS）です
- ホストベースのファイアウォール
  - netfilter (iptables、nftables) - パケットフィルタリング、ネットワークアドレス変換、およびその他のパケット操作を可能にする Linux カーネル内フレームワークです
  - Uncomplicated Firewall (ufw) - netfilter ファイアウォールを管理するためのプログラムです

メールと DNS セキュリティ

- メールセキュリティ
  - STARTTLS - プレーンテキスト通信プロトコルが暗号化接続にアップグレードできるようにするメカニズムです
  - SASL (Simple Authentication and Security Layer) - インターネットプロトコルの認証とデータセキュリティのためのフレームワークです

    - SPF (Sender Policy Framework) - メール配信中に送信者アドレスの偽装を検出するように設計されたメール認証方法です
    - DKIM (DomainKeys Identified Mail) - メール内の偽造送信者アドレスを検出するように設計されたメール認証方法です
    - DMARC (Domain-based Message Authentication, Reporting & Conformance) - メール認証、ポリシー、レポート プロトコルです
    - S/MIME - 認証、メッセージ整合性、否認防止、プライバシー、電子メッセージングアプリケーション用データセキュリティなどの暗号セキュリティサービスを提供する標準です
- DNS セキュリティ
    - DNSSEC - ドメイン名ルックアップへの応答を認証する Domain Name System（DNS）の機能です
    - DNS over TLS (DoT) - Transport Layer Security（TLS）プロトコルを介して Domain Name System（DNS）クエリと回答を暗号化する security protocol です
    - DNS over HTTPS (DoH) - HTTPS プロトコルを介してリモート Domain Name System（DNS）解決を実行するプロトコルです
- ツールとライブラリ
    - OpenDKIM - DomainKeys Identified Mail 対応アプリケーションおよび DomainKeys Identified Mail サービスを提供するための milter を開発・保守するコミュニティの取り組みです

ガバナンス、リスク、コンプライアンス（GRC）

> **Relevant DSS-P Skills**
>
> - 1. ビジネス変革 > 1.3 変革活動のマネジメント > リスク＆コンプライアンス
> - 4. セキュリティ > 4.1 セキュリティマネジメント > セキュリティマネジメント
> - 4. セキュリティ > 4.1 セキュリティマネジメント > プライバシー保護

データガバナンス

- Unity Catalog - さまざまなフォーマットとプラットフォーム全体での相互運用性、開放性、統一ガバナンスを提供するデータと AI のための普遍的なカタログです
- Microsoft Purview - 異種データエステート全体で組織がデータを保護・管理するための統合的アプローチです
- Amazon DataZone - 顧客が AWS、オンプレミス、サードパーティのソースに保存されているデータをカタログ化、検出、共有、管理できるデータ管理サービスです

AI ガバナンスとセキュリティ

- ISO/IEC 42001 - 組織内の人工知能管理システム（AIMS）を確立、実装、保守、継続的に改善するための要件を規定する世界初の AI 管理システム標準です
- METR - AI システムが社会に壊滅的な害をもたらす可能性があるかどうか、いつか測定する研究非営利団体です
- Microsoft Agent 365 - エンタープライズ環境内での自律 AI エージェントを監督・管理するための一元化されたガバナンスおよび管理プラットフォームです

規制と標準

- 法律と規制
  - General Data Protection Regulation (GDPR) - 世界で最も厳しいプライバシーおよびセキュリティ法です
    - データ主体の要求（DSR）
    - 違反通知
    - データ保護影響評価（DPIA）
  - California Consumer Privacy Act (CCPA) - カリフォルニア州の住民のプライバシー権と消費者保護を向上させることを目的とした州法です
  - Cyber Resilience Act - デジタルコンポーネントを持つ製品またはソフトウェアの購入・使用を行う消費者と企業を保護し、強制的なサイバーセキュリティ要件を導入する規制です
- セキュリティとプライバシーフレームワーク
  - NIST SP 800-53 - 国家安全保障に関連するもの以外のすべての U.S. 連邦情報システムのセキュリティ・プライバシーコントロールのカタログです
  - OSCAL - セキュリティコンプライアンスプロセスを自動化するために、NIST が主導するオープンで機械可読形式（XML、JSON、YAML）を提供する Open Security Controls Assessment Language です
  - ISO/IEC 27001 - 情報セキュリティ管理システムの国際標準です
- 業界と監査標準
  - PCI-DSS - ペイメントカードデータセキュリティのためのグローバル標準です
  - SOC 2 - 組織が顧客データを管理する方法を規定するサービス組織のための任意のコンプライアンス標準です
  - FIPS 140-2 - 暗号化モジュールを承認するために使用される米国政府コンピュータセキュリティ標準です
- 強化とガイドの実装

  - Security Technical Implementation Guides (STIGs) - DOD IA および IA 対応デバイス/システムの構成標準です
  - CIS Controls and Benchmarks - コンピュータセキュリティのための最善実行ガイドラインの出版です
  - NIST SP 800-190 - アプリケーションコンテナセキュリティガイドで、コンテナの使用に関連する潜在的なセキュリティ上の懸念を説明し、それらに対処するための推奨事項を提供しています

脆弱性管理とレポート

- Bug bounty program - 脆弱性またはバグを正常に発見しアプリケーション開発者に報告するエシカルハッカーに提供される金銭的報酬です
- 識別子と列挙
  - CVE (Common Vulnerabilities and Exposures) - 公開されている情報セキュリティの脆弱性と暴露の参照方法を提供するシステムです
  - CWE (Common Weakness Enumeration) - ソフトウェアの弱点と脆弱性のためのカテゴリシステムです
  - OSV (Open Source Vulnerability) - オープンソースプロジェクト向けの脆弱性データベースと分類インフラストラクチャです
- スコアと優先順位付け
  - CVSS (Common Vulnerability Scoring System) - コンピュータシステムセキュリティの脆弱性の重大度を評価するための無料でオープンな業界標準です
  - EPSS (Exploit Prediction Scoring System) - ソフトウェア脆弱性が野生で悪用される可能性を推定するためのデータドリブンフレームワークです
  - KEV (Known Exploited Vulnerabilities) - 野生で悪用されている脆弱性を含むカタログです
  - SSVC (Stakeholder-Specific Vulnerability Categorization) - 脆弱性が組織に対してもたらすリスクを評価し、対応する意思決定プロセスを提供する脆弱性管理方法論です
- プロトコルとデータベース
  - Security Content Automation Protocol (SCAP) - 自動化された構成、脆弱性、パッチチェックをサポートする仕様の多目的フレームワークです
  - NVD (U.S. National Vulnerability Database) - 標準ベースの脆弱性管理データの米国政府リポジトリです
  - SARIF - 静的解析ツールの出力の標準形式です

システムとパーソナルセキュリティ

> **Relevant DSS-P Skills**
>
> - 4. セキュリティ > 4.1 セキュリティマネジメント > セキュリティ体制構築・運営
> - 4. セキュリティ > 4.2 セキュリティ技術 > セキュア設計・開発・構築

OS とエンドポイントセキュリティ

- Address space layout randomization (ASLR) - メモリ脆弱性の悪用を防止するために、プロセスのアドレス空間の主要なデータ領域の位置をランダムに配置することで、メモリ保護に関連するコンピュータセキュリティ技術です
- W^X - プロセスのアドレス空間内のすべてのページが書き込み可能または実行可能である（ただし両方ではない）ことを保証するセキュリティ機能です
- Control-flow integrity - プログラムの実行フローをリダイレクトするマルウェア攻撃の多くを防止するコンピュータセキュリティ技術の一般用語です
- TPM (Trusted Platform Module) - 統合暗号鍵を通じてハードウェアをセキュアに保つために設計された専用マイクロコントローラーである、セキュアな暗号プロセッサーの仕様です
- Linux 強制アクセス制御
  - SELinux - さまざまな Linux ディストリビューションに追加されているカーネル変更とユーザースペースツールのセットです
  - AppArmor - プログラムパー・プログラムプロファイルでプログラムの機能を制限することをシステム管理者に許可する Linux カーネルセキュリティモジュールです
  - bubblewrap - Flatpak およびそれに似たプロジェクトで使用される低レベルの非特権サンドボックスツールです
- Linux 微粒度アクセス制御
  - Linux capabilities - スーパーユーザーの権限の一部をプロセスに付与するが、すべてを付与しない機能です
- 一般的なスキャン
  - OpenSCAP - Security Content Automation Protocol（SCAP）のオープンソース実装です
  - Lynis - Linux、macOS、または Unix ベースのオペレーティングシステムで実行されるシステムのセキュリティ監査ツールです

パーソナルセキュリティツール

- パスワードマネージャー
  - 1Password - 人間と AI エージェント全体にわたり、アイデンティティ、認証情報、シークレットの password manager および secure vault プラットフォームです
  - pass - 標準 unix password manager です
  - gokey - Go の簡単な vault なし password manager です
  - Buttercup - 無料でオープンソースでクロスプラットフォームの password manager です

05 - データサイエンス・エンジニアリング

基礎概念

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.1 データ・AI の戦略的活用 > データ・AI 理解・活用

一般的なデータコンセプトと原則

- Data - 1 つ以上のシンボルの任意のシーケンス。datum は単一のデータシンボルです
- Metadata - その他のデータに関する情報を提供する情報であり、データの内容ではありません
- Big data - 従来のデータ処理アプリケーションソフトウェアでは処理するには大きすぎたり複雑すぎたりするデータセット
- Unstructured data - 事前定義されたデータモデルを持たない、または事前定義されたやり方で編成されていない情報
- Data model - データの要素を編成し、それらが相互にどのように関連するか、および実世界のエンティティのプロパティとどのように関連するかを標準化する抽象モデル
  - Entity–relationship model - 特定の知識領域での関心のある相互関連するものの抽象的な説明
- Data orientation - データ自体を強調するデータの視点であり、データを使用するアプリケーションではありません
- DIKW pyramid - データ、情報、知識、および知恵の間の構造的および/または機能的な関係を表す一連のモデルのクラス

- Garbage in, garbage out - 出力の品質は入力の品質によって決定されるというコンピュータサイエンスと情報通信技術の概念
- Data cleansing - レコードセット、テーブル、またはデータベースから破損または不正確なレコードを検出して修正する（または削除する）プロセス
- Data lifecycle management - 情報システムのデータをそのライフサイクル全体を通じて管理する流れを管理するポリシーベースのアプローチ
- Master data - ビジネストランザクションの文脈を提供するビジネスエンティティについてのデータ
- Master data management - ビジネスと IT が連携して、企業の公式共有マスターデータアセットの均一性、精度、管理、セマンティック一貫性、および説明責任を確保する技術対応規律
- Data quality - 精度、完全性、一貫性、信頼性、および最新性などの要因に基づいて、データの状態を測定する指標
- Single source of truth - すべてのデータ要素が 1 つの場所でのみマスター（または編集）されるように情報モデルと関連データスキーマを構築する実践

コアデータエンジニアリングおよびデータベース概念

- Concurrency control - 同時実行操作の正しい結果が効率的に生成されることを確保するメカニズム
- CRUD operations - 永続的なストレージの 4 つの基本操作：作成、読み取り、更新、削除
- Shard - データベースまたは検索エンジンのデータの水平パーティション
- ETL - データが入力ソースから抽出され、変換され、出力データコンテナに読み込まれる 3 段階のプロセス
- ELT - 生データをソースシステムからデータウェアハウスなどの宛先リソースに移動し、その後使用のために変換するデータ統合プロセス
- Data pipeline - 直列に接続されたデータ処理要素のセット。1 つの要素の出力は次の要素への入力です
- Data governance - データのライフサイクル全体を通じて高いデータ品質が存在することを組織が確保できるようにするデータ管理概念
- Data lineage - データソースから消費まで、データが流れる様子を理解、記録、可視化するプロセス
- Online transaction processing (OLTP) - 同時に発生する多数のトランザクションを実行するデータ処理タイプ
- Online analytical processing (OLAP) - 多次元分析クエリに迅速に対応するコンピューティング内のアプローチ

- Search engine indexing - 高速で正確な情報検索を容易にするためのデータの収集、解析、および保存

データガバナンス、品質、およびアーキテクチャ

- Data Catalog - 組織がデータアセットを管理し、発見するのを支援する一元化されたメタデータリポジトリ
- Data Stewardship - 品質、セキュリティ、および準拠を確保するために組織のデータアセットを管理するための実践およびプロセスのセット
- Data Privacy - 個人が自身に関する情報に何が起こるかを決定する権利と能力
- Data Security - デジタル情報をオンライン脅威から保護するためのセキュリティ保護プロセス
- ISO 8000 - データ品質およびマスターデータの国際規格
- Data Contract - データプロデューサーとコンシューマー間のデータ構造、品質、およびセマンティクスに関する明示的な合意
- Schema Evolution - 既存のデータおよびアプリケーションとの互換性を維持しながらデータベーススキーマを変更するプロセス
- Dimensional Modeling - 事実と次元を使用して分析クエリの対象のデータウェアハウスを最適化するために使用されるデータベース設計手法

データサイエンスツールキット

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 数理統計・多変量解析・データ可視化

プログラミング言語とライブラリ

- Python - 素早く作業し、システムをより効果的に統合できるプログラミング言語
  - Awesome Python - 素晴らしい Python フレームワーク、ライブラリ、ツール、リソースのキュレーションされたリスト
  - Pandas - 高速で強力で柔軟で使いやすいオープンソースのデータ分析および操作ツール
  - Polars - 構造化データを操作するためのかなり高速な DataFrame ライブラリ
  - Narwhals - Python の遅延評価優先、型非依存、フレームワーク非依存の dataframe ライブラリ
  - NumPy - Python での科学計算の基礎パッケージ

    - SciPy - Python での科学計算のための基本的なアルゴリズム
    - SymPy - シンボリック数学用の Python ライブラリ
    - SageMath - GPL ライセンスの無料オープンソース数学ソフトウェアシステム
    - statsmodels - 多くの異なる統計モデルの推定用のクラスと関数を提供する Python モジュール。統計テストの実施と統計データの探索用
- R - 統計計算とグラフィックス用のフリーソフトウェア環境
    - Tidyverse - データサイエンス用に設計された R パッケージの意見的なコレクション
- GNU Octave - 主に数値計算を目的とした高級言語
- Wolfram Language - シンボリック言語であり、強力なプログラムを迅速に開発するために必要な幅と統一性を意図的に備えています

特殊科学ツール

- latexify - Python ソースコードの断片を対応する LaTeX 式にコンパイルする Python パッケージ
- handcalcs - Python 計算コードを LaTeX で自動的に描画する Python ライブラリ。ただし、鉛筆で書いた場合の形式を模倣しています
- NetworkX - 複雑なネットワークの構造、ダイナミクス、および機能を作成、操作、および研究するための Python パッケージ
- JAX - アクセラレーター指向の配列計算およびプログラム変換用の Python ライブラリ

データソースおよび地理空間情報

- GeoLite2 - ダウンロード可能なデータベースおよび Web サービス形式の無料の地理位置情報および ASN データのセット

スプレッドシートおよびコラボレーティブデータプラットフォーム

- Microsoft Excel - 業界最高のスプレッドシートソフトウェアプログラムおよび強力なデータ可視化および分析ツール
- Grist - スプレッドシートの親しみやすいインターフェイスとリレーショナルデータベースのパワーと構造を組み合わせたリレーショナルスプレッドシート
- NocoBase - 複雑なビジネスアプリケーションと内部ツールを構築するために設計されたスケーラビリティ優先のオープンソースノーコードプラットフォーム
- NocoDB - あらゆるデータベースをスマートスプレッドシートに変え、リレーショナルデータベース用のコラボレーティブインターフェイスを提供するオープンソースノーコードプラットフォーム

- Airtable - スプレッドシートの柔軟性とデータベースの強力さを組み合わせて、チームが自分たちの仕事を管理するのを支援するプラットフォーム

インタラクティブコンピューティング環境

- JupyterLab - ノートブック、コード、およびデータ用の Web ベースの対話的開発環境
- Jupyter Notebook - 計算ドキュメントを作成して共有するための元の Web アプリケーション
  - VSCode Jupyter Extension - 環境でサポートされている言語カーネルの基本的なノートブックサポートを提供する VS Code 拡張機能
- nbviewer - Jupyter Notebook を共有する簡単な方法
- R Markdown - コード、描画出力、および散文を組み合わせて動的な分析ドキュメントを作成するのを支援するオーサリングフレームワーク
- Wolfram Notebooks - 強力な環境であり、テキスト、識字的なプログラミング、グラフィックス、カスタムインタラクティブ要素を組み合わせた探索と通信
- Voila - Jupyter ノートブックをスタンドアロン Web アプリケーションに変えるツール

データ可視化

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 数理統計・多変量解析・データ可視化

一般的なチャートタイプ

- Histogram - 数値データの分布の表現
- Scatter plot - デカルト座標を使用してデータセットの通常 2 つの変数の値を表示するプロットまたは数学図の類型
- Box plot - 四分位数を通じた数値データのグループの位置、スプレッド、スキューを図的に実証する方法
- Error bar - グラフで使用される、報告された測定の不確実性を示すデータの可変性のグラフィック表現
- Heat map - 現象の大きさを 2 次元の色として表示する手法
- Choropleth map - 事前定義された領域のセットが統計変数に比例して色または図案化される主題図の類型

- Proportional symbol map - 定量変数を表すために異なるサイズのシンボルを使用する主題図の類型
- Tag cloud - テキストデータの目新しいビジュアル表現

可視化ツールとライブラリ

- Python ライブラリ
  - matplotlib - Python での静的、アニメーション、対話的な可視化を作成するための包括的なライブラリ
  - seaborn - matplotlib に基づく Python データ可視化ライブラリ
  - Plotly - Python 用のインタラクティブでオープンソースでブラウザベースのグラフィングライブラリ（Plotly Express を含む）
  - WordCloud for Python - Python での少し単語クラウドジェネレーター
- JavaScript ライブラリ
  - D3 - カスタムデータ可視化用の JavaScript ライブラリ
  - GoJS - Web ブラウザでインタラクティブな図を簡単に作成できる JavaScript ライブラリ
  - Chart.js - 最新の Web 用のシンプルで柔軟な JavaScript チャートライブラリ
  - Recharts - React コンポーネント上に構築された構成可能なチャートライブラリ
  - Tabulator - テーブルとデータグリッドを作成するための、使いやすく、コーディングが簡単で、フル機能でインタラクティブな JavaScript ライブラリ
- 文法とその他
  - gnuplot - ポータブルなコマンドラインドリブングラフユーティリティ
  - ggplot2 - グラフィックス文法に基づいた宣言的にグラフィックスを作成するシステム
  - Vega - 可視化文法であり、インタラクティブな可視化デザインを作成、保存、共有するための宣言型言語
  - Vega-Lite - インタラクティブグラフィックスの高レベルの文法

ダッシュボーディングと Web アプリ

- Dash - Python、R、Julia、F# でデータアプリを迅速に構築するための元のローコードフレームワーク
- Panel - インタラクティブな Web アプリとダッシュボードを作成できる強力な Python ライブラリ
- Streamlit - データアプリをより速く構築して共有する方法

分散システム

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.3 データマネジメント > データエンジニアリング（設計・収集・統合・提供）

分散コンピューティングの原則

- Distributed computing - このようなシステムを研究するコンピュータサイエンスの分野
- Single point of failure - システムの一部。失敗すると、システム全体が動作を停止します
- Fault tolerance - システムのコンポーネントの一部が失敗した場合でも、システムが適切に動作し続ける機能
- Load balancing - タスクのセットをリソースのセットに配分する処理。全体的な処理をより効率的にする目的で
- Fallacies of distributed computing - 分散アプリケーションに新しいプログラマーが必然的に行う誤った仮定を説明する一連の主張
- Byzantine fault - 分散システムの状態であり、コンポーネントが失敗する可能性があり、コンポーネントが失敗したかどうかに関する不完全な情報があります
  - Consensus - 分散プロセスまたはシステム間で単一のデータ値について必要な同意を達成するために分散システムで使用される耐障害性メカニズム
- CAP theorem - 分散データストアは、一貫性、可用性、およびパーティション耐性の 3 つの保証のうち 2 つのみを提供できるという定理
- BASE properties - 一貫性よりも可用性を優先するデータベースモデル
- Amdahl's law - リソースがシステムに追加されるにつれて、タスクのスピードアップを制限する式

コンセンサスとレプリケーション戦略

- Raft Consensus Algorithm - Paxos より理解しやすいように設計されたコンセンサスアルゴリズムであり、クラスタ全体での安全な状態マシンレプリケーションを有効にします
- Paxos Algorithm - 信頼できないまたは非同期プロセッサのネットワークでコンセンサスを解決するためのプロトコルファミリー
- Data Replication - コンピュータまたはサーバーから別の場所、コンピュータ、またはサーバーへのデータの頻繁な電子コピー

    - Master-Slave Replication - 1 つのプライマリノードが書き込みを受け入れ、スレーブがデータをレプリケートするパターン
    - Consensus - 分散プロセスまたはシステム間で単一のデータ値について必要な同意を達成するために分散システムで使用される耐障害性メカニズム

分散パターンと監視可能性

- Circuit Breaker Pattern - 分散システムでのカスケード障害を防ぐための設計パターン
- Distributed Tracing - アプリケーション、特にマイクロサービスアーキテクチャを使用して構築されたアプリケーションのプロファイリングおよび監視方法
- Event Sourcing - アプリケーション状態へのすべての変更が不変イベントのシーケンスとして保存されるパターン

分散ストレージシステム

- 分散ファイルシステム
    - HDFS - コモディティハードウェアで実行するために設計された分散ファイルシステム
    - IPFS - Web をより高速、より安全、よりオープンにすることを目的とした P2P ハイパーメディアプロトコル
        - Kubo - IPFS の Go 実装
- Object storage - オブジェクトとしてデータを管理するコンピュータデータストレージアーキテクチャ
    - Amazon S3 - 業界をリードするスケーラビリティ、データ可用性、セキュリティ、パフォーマンスを備えたオブジェクトストレージサービス
    - Azure Blob Storage - 大量の非構造化データの保存に最適化された Microsoft のクラウドオブジェクトストレージソリューション
    - Azure Data Lake Storage (ADLS) - 高性能分析ワークロード用のスケーラブルで安全なデータレイク
    - Google Cloud Storage - Google Cloud Platform インフラストラクチャ上のデータを保存およびアクセスするための RESTful オンラインファイルストレージ Web サービス
    - Cloud Storage for Firebase - ユーザー生成コンテンツ（画像やビデオなど）をアップロードして共有できるサービス
    - Supabase Storage - 写真やビデオなどの大きなファイルの保存と配信を簡単にするサービス

- 自己ホスト型（高度）
  - Ceph - オープンソースの分散ストレージシステム
  - MinIO - 高性能で S3 互換のオブジェクトストア
- ツール
  - s5cmd - 非常に高速な S3 とローカルファイルシステム実行ツール
  - Rclone - クラウドストレージ上のファイルを管理するためのコマンドラインプログラム
  - Azure Storage Explorer - Windows、macOS、Linux 上で Azure Storage データを簡単に操作できるスタンドアロンアプリ
  - Azurite - オープンソースの Azure Storage エミュレーター

数学と統計

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 数理統計・多変量解析・データ可視化

基本数学

- Algebra - 代数的構造として知られる抽象システムと、これらのシステム内の式の操作を扱う数学の一分野
  - Boolean algebra - 変数の値が true と false の真理値（通常は 1 と 0 で示される）であり、論理演算子（and、or、not）を使用するという点で初等代数とは異なる代数の分野
  - Elementary algebra - 代数の基本概念を含む数学の一分野
    - Equation - 2 つの式の等式を等号 = で接続することで表現する数式
    - Logarithm - その数を生成するために別の固定値である底を上げる必要がある指数
  - Abstract algebra - 要素に特定の操作を行う代数的構造のセットの研究
  - Linear algebra - 線形方程式、線形マップ、およびベクトル空間での表現を通じたそれらの表現に関する数学の分野
    - Vector space - 要素（多くの場合ベクトルと呼ばれる）を追加して、スカラーと呼ばれる数で「スケール」できるセット
    - Matrix - 行と列に配置された要素またはエントリを持つ数値またはその他の数学的オブジェクトの矩形配列。通常、加算と乗算の特定の特性を満たしています

        - Sparse matrix - ほとんどの要素がゼロである行列
        - Rank - その列によって生成（またはスパン）されるベクトル空間の次元
        - Determinant - 正方行列のエントリのスカラー値関数
- Calculus - ジオメトリが図形の研究である方法と同じように、代数が算術操作の一般化の研究である方法と同じように、継続的な変化の数学的研究
    - Differential calculus - 量が変化する速度を研究するカルキュラスの分野
    - Integral calculus - 合計の連続類似物であり、面積、体積、およびそれらの一般化を計算するために使用されます
    - Differential equation - 1 つ以上の未知の関数とそれらの導関数を関連付ける方程式
- Geometry - 距離、形状、サイズ、図の相対的な位置などの空間の特性に関する数学の一分野
    - Trigonometry - 三角形の角度と辺の長さの関係に関する数学の分野
    - Coordinate system - ユークリッド空間などの多様体上の点またはその他の幾何学的要素の位置を一意に決定して標準化するために 1 つ以上の数値または座標を使用するシステム
    - Euclidean distance - ユークリッド空間内の 2 つのポイント間の線分の長さ
- Category theory - 数学的構造とそれらの関係の一般理論
    - Functor - カテゴリ間のマッピング
- Root mean square - 数値のセットの二乗の平均の平方根
- 変換
    - Discrete cosine transform - 異なる周波数で振動するコサイン関数の合計の観点から、有限の一連のデータポイントを表現する変換
    - Discrete Fourier transform - 均等に配置されたサンプル関数の有限シーケンスを離散時間フーリエ変換の均等に配置されたサンプルの同じ長さのシーケンスに変換するフーリエ変換のディスクリート版
- 関連リソース
    - NIST Digital Library of Mathematical Functions - 応用数学の特殊関数の決定的なリファレンス
        - Notations - ライブラリで使用されている表記法のリスト

確率と情報理論

- Probability theory - 確率に関する数学の分野

    - Bayes' theorem - 条件付き確率を反転させるための数学的規則であり、その効果が与えられて原因の確率を見つけることができます
    - Central limit theorem (CLT) - 適切な条件の下で、サンプル平均の正規化されたバージョンの分布が標準正規分布に収束するという定理
- Information theory - デジタル情報の定量化、保存、および通信の科学的研究
    - Entropy - ランダム変数の可能な結果に固有の「情報」、「驚き」、または「不確実性」の平均レベル

統計と数値方法

- Statistics - データの収集、組織化、分析、解釈、および提示に関する規律
    - Sampling - 統計母集団全体の特性を推定するために、統計母集団内の個人のサブセットの選択
        - Sampling error - 全体の母集団ではなく、サンプルを観測することによって引き起こされるエラー
    - Errors and residuals - 統計サンプルの要素の観測値とその「真の値」からの偏差の測定
    - Frequency - 観察が実験またはスタディで発生または記録された回数
        - Contingency table - 変数の多変量頻度分布を表示する行列形式のテーブルの類型
    - Confounding - 従属変数と独立変数の両方に影響を与え、見せかけの関連を引き起こす変数
    - Standard deviation - 変数の値の平均に関する変動量の測定
    - Root mean square deviation - 予測値と実際の値の間の二乗差の平均の平方根
    - F-score - 二値分類および情報検索システムの統計分析の予測パフォーマンスの測定
    - Correlation - 2 つの確率変数またはビバリエートデータ間の統計関係
        - Pearson correlation coefficient - 2 つのデータセット間の線形相関を測定する相関係数
    - Hypothesis testing - データが特定の仮説を拒否するために十分な証拠を提供するかどうかを決定するために使用される統計的推論の方法
        - Null hypothesis - 統計的関係と有意性が、単一の観測変数の特定のセットの間、または 2 つの観測データセットと測定現象の間に存在しないことを示唆する典型的な統計理論
        - Confidence interval (CI) - 母集団平均など、未知の統計パラメータの真の値を含む可能性がある（反復サンプリング）値の範囲

        - P-value - null 仮説が正しいという仮定の下で、実際に観測された結果と同等に極端な検査結果を取得する確率
- 数値方法
    - Significant figures - 位置標記で書かれた数値内の特定のデジタルであり、特定の量の伝達における信頼性と必要性の両方を持ちます
- リソース
    - Openstax Introductory Statistics - 統計入門コース向けのオープンソースの教科書
    - OpenIntro Statistics - 従来のカリキュラムの動的な見方であり、コミュニティカレッジから Ivy League までで正常に使用されています

データ形式とアーキテクチャ

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.3 データマネジメント > データエンジニアリング（設計・収集・統合・提供）
> - 2. データ整備・活用 > 2.3 データマネジメント > データの品質・安全性向上

データ形式とテーブル形式

- Apache Parquet - 効率的なデータストレージと検索のために設計されたオープンソースの列指向データファイル形式
- Apache ORC - Hadoop ワークロード用の最小で最速の列ストレージ
- Apache Arrow - 普遍的な列フォーマットおよび高速データ交換と in-memory 分析用のマルチ言語ツールボックス
- BSON - JSON のような文書のバイナリエンコード シリアル化
- Apache Avro - レコードデータ用の主要なシリアル化形式、ストリーミングデータパイプラインの最初の選択肢
- Delta Lake - コンピュートエンジンでフォーマット非依存の Lakehouse アーキテクチャを構築できるオープンソースストレージフレームワーク
- Apache Iceberg - 巨大な分析データセット用のオープンテーブル形式
- Apache Hudi - ストリーミングデータレイクプラットフォーム

データアーキテクチャと方法論

- Data warehouse - レポートとデータ分析に使用されるシステムであり、ビジネスインテリジェンスのコアコンポーネント

- Data lake - その自然/生フォーマット（通常のオブジェクト blob またはファイル）で保存されたデータのシステムまたはリポジトリ
- Data lakehouse - データレイクとデータウェアハウスの最高の要素を組み合わせた新しいオープンアーキテクチャ
- Medallion Architecture - レイクハウスのデータを論理的に編成するために使用されるデータ設計パターン
- CRISP-DM - データマイニングの専門家が使用する一般的なアプローチを説明するオープン標準プロセスモデル
- PPDAC (Problem, Plan, Data, Analysis, Conclusion) - データを使用した実世界の課題を解決するための統計識字率と循環フレームワークへのよく確立されたアプローチ
- Data architecture - どのデータが収集され、データシステムおよび組織で保存、配置、統合、および使用されるかを管理するモデル、ポリシー、ルール、および標準のセット
- DAMA-DMBOK - データ管理知識体ガイド。データ管理の 13 の機能領域全体のフレームワークと用語を概説しています

データガバナンスとメタデータ管理

- Apache Atlas - 企業が準拠要件を満たすことを可能にするスケーラブルで拡張可能なコア基礎ガバナンスサービスのセット
- Collibra - データ管理用の共通言語を提供するエンタープライズデータガバナンスプラットフォーム
- Informatica Metadata Manager - エンタープライズデータガバナンス用の包括的なメタデータ管理ソリューション
- OpenMetadata - データ検出、ガバナンス、およびコラボレーション用のオープンソースメタデータ管理プラットフォーム

データ品質と検証

- Great Expectations - データ品質を定義、ドキュメント化、テストするための Python ライブラリ
- Apache Griffin - 分散データ品質測定用に Apache Spark と Apache Hadoop 上に構築されたデータ品質ソリューション
- Soda - 最新のデータスタックと統合するデータ品質監視ソリューション

データバージョニングとスキーマ管理

- Schema Registry - Kafka トピック用のスキーマを一元化するホストスキーマ管理サービス

- Git ベースのスキーマ管理 - Git リポジトリを使用してデータベーススキーマをバージョン制御
- DBT Contracts - 入出力データ要件を定義する明示的なデータコントラクト

リレーショナルデータベース（SQL）

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.3 データマネジメント > データエンジニアリング（設計・収集・統合・提供）

SQL 基礎

- 基礎的な概念
  - Relational model - 一階述語論理と一貫した構造と言語を使用してデータを管理するアプローチ
  - ACID properties - エラー、電源障害、およびその他の不幸にもかかわらず、データの有効性を保証することを意図した一連のデータベーストランザクションプロパティ
    - Atomicity、Consistency、Isolation、Durability
  - Codd's Twelve Rules - リレーショナルデータベース管理システムが関連すると見なされるために必要なことを定義するために Edgar F. Codd によって提案された 13 の規則のセット
  - Database normalization - リレーショナルデータベースのデータ冗長性を最小化するために、列（属性）およびテーブル（リレーション）を編成するプロセス
- 言語とダイアレクト
  - Structured Query Language (SQL) - リレーショナルデータベース管理システムに保持されているデータを管理するために使用されるドメイン固有言語
    - コマンドカテゴリ
      - DDL - データ定義言語
      - DQL - データクエリ言語
      - DML - データ操作言語
      - DCL - データ制御言語
      - TCL - トランザクション制御言語
    - SQL Join - リレーショナルデータベース内の 1 つ以上のテーブルから列を組み合わせるクローズ

        - Aggregate function - 複数の行の値がグループ化されて、単一の要約値を形成する関数
    - Transact-SQL - SQL Server をプログラムおよび管理するために使用される SQL への独自の拡張

データベース管理システム（DBMS）

- クライアント サーバー RDBMS
    - PostgreSQL - ユニバーシティ オブ カリフォルニア バークレー コンピュータ サイエンス部門で開発されたバージョン 4.2 に基づく POSTGRES に基づくオブジェクト関連のデータベース管理システム（ORDBMS）
    - MySQL - 最も人気のあるオープンソース SQL データベース管理システムであり、Oracle Corporation によって開発、配布、サポートされています
    - MariaDB community Server - MySQL のコミュニティが開発したオープンソース リレーショナルデータベースのフォーク
- 分散 SQL
    - TiDB - ハイブリッドトランザクション分析処理（HTAP）ワークロードをサポートするオープンソース分散 SQL データベース
- 組み込み/インプロセス
    - SQLite - 小さく、高速で、自己完結型で、高信頼性とフル機能のデータベースエンジンを実装する C 言語ライブラリ
    - PGlite - ブラウザ、Node.js、および Bun でデータベースを実行できるようにする TypeScript/JavaScript クライアントライブラリにパッケージ化された WASM ビルド
    - DuckDB - インプロセス SQL OLAP データベース管理システム
- ストレージエンジン
    - Storage Engine - データベース管理システムが、データベースからデータを作成、読み取り、更新、削除（CRUD）するために使用するソフトウェアコンポーネント
    - InnoDB - MySQL および MariaDB 用のトランザクショナルストレージエンジン

クラウドおよび管理サービス

- 管理データベースサービス
    - Amazon RDS - クラウドでデータベースを設定、操作、スケーリングすることを簡単にする管理サービスのコレクション
    - Amazon Aurora - PostgreSQL、MySQL、および DSQL 用にグローバルスケールで

高性能と可用性を提供する完全に管理されたリレーショナルデータベースエンジン

- Azure SQL Database - クラウド用に構築された知的でスケーラブルなリレーショナルデータベースサービス
- Azure HorizonDB - 最大 3,072 vCores と 128 TB ストレージをサポートするスケールアウトアーキテクチャを備えた、高スループット AI 対応アプリケーション向けに設計された完全に管理された PostgreSQL 互換クラウドネイティブデータベースサービス
- Google Cloud SQL - Google Cloud 上のリレーショナルデータベースを設定、保守、管理、実装するのに役立つ完全に管理されたデータベースサービス
- Neon - サーバーレスで耐障害性があり、スケーラブルな Postgres。寛容な無料層を備えています
- Turso - SQLite の完全な書き直しで構築された SQLite 互換のデータベース。どこでも実行するのに十分に軽く、十分に高速です

接続とツール

- 接続 API と ORM
  - Connection pool - 将来のデータベースへのリクエストが必要な場合に接続を再利用できるように、保持されるデータベース接続のキャッシュ
  - ODBC - データベース管理システムにアクセスするための標準アプリケーションプログラミングインターフェイス
  - JDBC - Java プログラミング言語からほぼすべてのテーブル形式のデータソースにアクセスできるようにする API
    - Jdbi - Java データベース接続 API をより慣用的な方法で使用する方法を提供するライブラリ
  - Object-Relational Mapping (ORM) - オブジェクト指向プログラミング言語を使用して、互換性のない型システム間でデータを変換するプログラミング技術
    - Prisma - 次世代 ORM。データベース付きの信頼性とスケーラビリティに優れたアプリケーションを簡単に構築できます
    - Drizzle ORM - 開発者の経験を念頭に置いた軽量でパフォーマンスの高い TypeScript ORM
    - Hibernate - Java プログラミング言語用のオブジェクト関連マッピングツール
    - SQLAlchemy - Python SQL ツールキットおよびオブジェクトリレーショナルマッパー。アプリケーション開発者に SQL の完全なパワーと柔軟性を提供します
    - GORM - Golang 用の素晴らしい ORM ライブラリであり、開発者に優しいことを目指しています

        - XORM - Go 用のシンプルで強力な ORM
        - Diesel - Rust 用の安全で拡張可能な ORM とクエリビルダー
- 開発者ライブラリとドライバー
    - Vanna.AI - 検索拡張を使用して LLM を使用して正確な SQL クエリを生成するのに役立つ Python パッケージ
    - Psycopg - Python プログラミング言語用の最も一般的な PostgreSQL アダプター
- データベースクライアントと IDE
    - pgAdmin - PostgreSQL の最も一般的でフル機能のオープンソース管理および開発プラットフォーム
    - SSMS (SQL Server Management Studio) - SQL Server から Azure SQL Database まで、あらゆる SQL インフラストラクチャを管理するための統合環境
    - DB Browser for SQLite - SQLite と互換性のあるデータベースファイルを作成、設計、編集するための高品質でビジュアルなオープンソースツール
    - Azure Data Studio - データランドスケープを簡略化するために設計された最新のオープンソースのクロスプラットフォームハイブリッドデータ分析ツール
    - Beekeeper Studio - 最新で使いやすく、見栄えの良い SQL エディターおよびデータベースマネージャー
- コマンドラインおよびデプロイメントユーティリティ
    - sqlcmd utility - Transact-SQL ステートメントおよびスクリプト用のアドホック対話的実行のコマンドラインユーティリティ。および T-SQL スクリプティングタスクの自動化
    - sqlpackage - いくつかのデータベース開発タスクを自動化するコマンドラインユーティリティ
    - DAC (Data-tier Applications) - ユーザーのデータベースに関連する SQL Server オブジェクトのすべてを定義する論理データベース管理概念
    - pgroll - PostgreSQL のゼロダウンタイムで反転可能なスキーマ移行ツール
- 監視と分析
    - pgBadger - 速度のために構築された PostgreSQL ログアナライザーであり、完全に詳細なレポートと専門的なレンダリング
- パフォーマンスベンチマーク
    - TPC-H - ビジネス指向のアドホッククエリおよびデータ変更に対する大量のデータを通じてデータベースシステムを評価する意思決定サポートベンチマーク
    - TPC-DS - 複雑なデータベースクエリとビッグデータ環境をシミュレートして、システムパフォーマンスと価格/パフォーマンスメトリクスを評価する意思決定サポー

トベンチマーク

NoSQL と特殊なデータベース

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.3 データマネジメント > データエンジニアリング（設計・収集・統合・提供）

NoSQL データモデル

- Object-relational impedance mismatch - オブジェクト指向プログラミング言語またはスタイルで書かれたプログラムでリレーショナルデータベース管理システム（RDBMS）が使用されている場合、しばしば遭遇する概念的および技術的な困難
- ドキュメントデータベース
  - MongoDB - アプリケーション開発とスケーリングの容易さのために設計されたドキュメントデータベース
  - DocumentDB - 最新のアプリケーション用に構築された強力でスケーラブルなオープンソースドキュメントデータベース
- キー値ストア
  - etcd - 分散システムの最も重要なデータのための分散で信頼性の高いキー値ストア
  - Redis - 数百万の開発者によってキャッシュ、ベクトルデータベース、ドキュメントデータベース、ストリーミングエンジンとして使用される in-memory データストア
  - Dragonfly - Redis の代替ドロップイン
- グラフデータベース
  - Neo4j - 無制限のスケール、セキュリティ、データ整合性を備えた高速グラフデータベース
    - Cypher - プロパティグラフデータベース用の宣言型クエリ言語
  - LadybugDB - 高度に規制された業界向けに構築された組み込み列グラフデータベース
- ワイド列データベース
  - Apache Cassandra - オープンソースの NoSQL 分散データベース
  - Apache HBase - Hadoop データベース。分散型でスケーラブルなビッグデータストア
  - ClickHouse - リアルタイム分析用に設計された高速でオープンソースの OLAP（

オンライン分析処理）データベース管理システム

ベクトルと AI データベース

- 概念
  - HNSW (Hierarchical Navigable Small Worlds) - ベクトル類似性検索のための最高パフォーマンスのインデックス
- ベクトルデータベース
  - Pinecone - あらゆる規模で関連する結果を提供するために構築されたベクトルデータベース
  - pgvector - Postgres のオープンソースベクトル類似性検索
  - ElasticSearch vector database - 世界で最も広く展開されているオープンソースベクトルデータベース
  - Weaviate - AI アプリケーション開発を簡素化するオープンソースベクトルデータベース
  - Milvus - 数十億のベクトルを処理するために構築された高性能なオープンソースベクトルデータベース
  - Chroma - AI ネイティブオープンソース埋め込みデータベース
  - Qdrant - あらゆる規模での AI 検索構築を支援するために Rust で完全に構築された高性能ベクトル検索エンジン

クラウド NoSQL サービス

- マルチモデルデータベース
  - Azure Cosmos DB - 最新のアプリケーション開発用の完全に管理されたサーバーレス分散データベース
  - Amazon DynamoDB - あらゆる規模で高性能アプリケーションを実行するために設計された完全に管理された、サーバーレス、キー値 NoSQL データベース
- ドキュメントデータベース
  - Cloud Firestore - Apple、Android、Web アプリがネイティブ SDK を通じて直接アクセスできるクラウドホストの NoSQL データベース
- グラフデータベース
  - Amazon Neptune - 高度に接続されたデータセットで動作するアプリケーションを簡単に構築および実行できる高速で信頼性が高く、完全に管理されたグラフデータベースサービス
- ワイド列データベース

    - Google Cloud Bigtable - 大規模な分析および運用ワークロード用の NoSQL ワイド列データベースサービス

データ処理とメッセージング

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.3 データマネジメント > データエンジニアリング（設計・収集・統合・提供）

エンタープライズ統合

- Enterprise Integration Patterns - 分散アプリケーションの設計と構築、または既存のパターン言語である 65 統合パターンの支援
- Apache Camel - 様々なシステムの消費または生産するデータを統合するすばやく簡単に統合できるオープンソース統合フレームワーク

メッセージキューイングとイベントストリーミング

- 概念
    - Message Brokers - 送信者の正式なメッセージング プロトコルから受信者の正式なメッセージング プロトコルにメッセージを翻訳する中間コンピュータプログラムモジュール
    - Dead-letter queue - 正常に配信または処理できなかったメッセージを保存するためにメッセージキューイングシステムで使用される特殊なキュー
- メッセージングとストリーミングプラットフォーム（ソフトウェア）
    - Apache Kafka - オープンソースの分散イベントストリーミングプラットフォーム
        - Apache Kafka エコシステム
            - Kafbat UI - Apache Kafka クラスタを監視および管理するために設計された多機能で、高速で、軽量で柔軟な Web インターフェイス
    - RabbitMQ - 信頼性の高い成熟したメッセージングおよびストリーミングブローカー
- クラウドサービス
    - Amazon Kinesis - リアルタイムストリーミングデータを簡単に収集、処理、分析できるようにするサービス
    - Azure Event Hubs - 秒単位で数百万のイベントを取り込むことができるスケーラビリティに優れた信頼性の高いイベントストリーミングプラットフォーム
    - Azure Service Bus - メッセージキューと発行購読トピック付きの完全に管理されたエンタープライズメッセージブローカー

バッチ処理（ETL/ELT）

- ベースフレームワーク
  - Apache Hadoop - 大規模なデータセットの分散処理を可能にするフレームワーク
    - mrjob - フレームワーク上で実行される Python プログラムを書く最も簡単なルート
  - Apache Spark - 大規模データ分析の統合エンジン
    - PySpark - エンジン用の Python API。言語でビッグデータ処理を許可します
  - RAY - AI と Python ワークロードのスケーリングを簡単にするオープンソース統合計算フレームワーク
  - Joblib - Python で軽量なパイプラインを提供するためのツールのセット
- ワークフロー編成および ETL ツール（ソフトウェア）
  - Apache NiFi - データを処理および配布する簡単で強力で信頼性の高いシステム
  - Apache Airflow - ワークフローをプログラムで作成、スケジュール、監視するプラットフォーム
  - dbt - チームが大規模に信頼できるガバナンスの効いたデータを提供できるようにする、信頼できるデータ配信のための統合プラットフォーム
  - Dagu - ラップトップから分散クラスタにスケーリングする単一のバイナリからタスクをオーケストレーションする宣言的でファイルベースの自己完結型プラットフォームを提供するローカルファーストワークフローエンジン
- 管理された ETL およびデータ統合サービス
  - Azure Data Factory - スケールアウトサーバーレスデータ統合およびデータ変換用のクラウド ETL サービス
  - AWS Glue - 複数のソースからデータを検出、準備、移動、統合することを容易にするサーバーレスデータ統合サービス
  - Google Cloud Data Fusion - ユーザーが ETL/ELT データパイプラインを効率的に構築および管理するのに役立つ完全に管理されたクラウドネイティブデータ統合サービス

ストリーム処理

- ストリーム処理エンジン（ソフトウェア）
  - Spark Structured Streaming - Spark SQL エンジン上に構築されたスケーラブルで耐障害性のあるストリーム処理エンジン
  - Apache Storm - 無料でオープンソースの分散リアルタイム計算システム
  - Apache Flink - 非バウンドおよびバウンドデータストリーム上の状態計算用のフ

レームワークおよび分散処理エンジン

- クラウドサービス
  - Google Cloud Dataflow - 自動スケーリングとバッチ処理を通じてレイテンシ、処理時間、コストを最小化する完全に管理されたストリーミング分析サービス

データ分析と検索

> **関連する DX 推進スキル標準のスキル**
>
> - 2. データ整備・活用 > 2.1 データ・AI の戦略的活用 > データ・AI 理解・活用

検索エンジンとプラットフォーム

- Web 検索エンジン
  - Google Search - Webページ、画像、ビデオなど、世界の情報を検索できる検索エンジン
  - DuckDuckGo - あなたを追跡しない検索エンジン
- 回答エンジン
  - Wolfram|Alpha - 革新的なアルゴリズム、ナレッジベース、AI テクノロジーを使用して専門家レベルの回答を計算する計算知識エンジン
  - Perplexity AI - あらゆる質問に正確で信頼できるリアルタイムの回答を提供する AI 駆動型回答エンジン
- 検索プラットフォームとツール
  - Azure AI Search - エンタープライズおよび Web コンテンツへのアクセスを統合し、AI 駆動検索と検索拡張生成を行う完全に管理されたクラウドホスト型サービス
    - Reciprocal Rank Fusion (RRF) - 複数の以前に実行されたクエリからの検索スコアを評価して、統一された結果セットを生成するアルゴリズム
    - BM25 relevance scoring - フルテキスト検索の一致ドキュメントの関連度スコアを計算するために使用される Okapi BM25 ランキング関数
  - ElasticSearch - オープンソースの分散型、RESTful 検索と分析エンジン。スケーラブルなデータストアおよびベクトルデータベース
    - Painless - エンジンでの使用専用に設計されたシンプルで安全なスクリプト言語
    - ES|QL - エンジンに保存されたデータをフィルタ、変換、分析できるパイプライン言語

        - Kibana - エンジンに保存されているデータをクエリ、分析、可視化、管理するためのオープンソースインターフェイス
        - Kibana Query Language - データをフィルタリングするためのシンプルなテキストベースのクエリ言語
    - Apache Solr - Apache Lucene 上に構築された人気のあり、かなり高速でオープンソースのエンタープライズ検索プラットフォーム
        - Apache Lucene - 強力なインデックス作成と検索機能を提供する Java ライブラリ
    - Faiss - 高密度ベクトルの効率的な類似性検索とクラスタリングのためのライブラリ
    - Meilisearch - アプリ、Web サイト、ワークフローに簡単にフィットする高速検索エンジン
    - TypeSense - 高速で、オープンソースで、喜ばしい検索体験を構築するための検索型エンジン

分析エンジンとプラットフォーム

- ソフトウェアと管理サービス
    - Apache Hive - 分散で耐障害性のあるデータウェアハウスシステムであり、大規模な規模での分析を実現します
    - Presto - あらゆる規模での高速で信頼できる効率的な分析のために設計された分散 SQL クエリエンジン
    - Trino - 1 つ以上の異機種データソースに分散する大規模なデータセットをクエリするように設計された分散 SQL クエリエンジン
    - Amazon EMR - 大規模な分散データ処理ジョブ、対話型 SQL クエリ、機械学習アプリケーション実行用のクラウドビッグデータプラットフォーム
    - Amazon Redshift - クラウドの完全に管理されたペタバイトスケールデータウェアハウスサービス
    - Amazon Athena - Amazon S3 およびその他のデータストアのデータを標準 SQL を使用して簡単に分析できるインタラクティブクエリサービス
    - Databricks - 組織全体がデータと AI を使用できるようにするプラットフォーム
        - Declarative Automation Bundles - 以前は Databricks Asset Bundles として知られていた、データと AI プロジェクト用のソフトウェア エンジニアリング ベストプラクティス（ソースコントロール、コードレビュー、テスト、継続的統合と配信（CI/CD））の採用を促進するツール
    - Snowflake - 分析、アプリケーション、AI を単一の完全に管理されたプラットフォームで実行するための近くの無制限のスケールでデータを動員する AI データ

クラウド

- Microsoft Fabric - データ移動、データレイク、データエンジニアリング、データ統合、データサイエンス、リアルタイム分析、ビジネスインテリジェンスなどの全サービス機能を備えたエンドツーエンド分析ソリューション
  - Microsoft OneLake - 組織全体向けの単一で統一された論理的なデータレイク
  - Real-Time Intelligence - 動きのストリーミングデータから洞察を抽出するサービス。摂取、変換、ストレージ、分析、可視化、および時間ベースイベントでのリアルタイムアクション用のエンドツーエンドソリューション
  - Rayfin CLI - プロジェクトスキャフォルディング、リモート配置、構成管理機能を備えた Fabric アプリケーションを作成、デプロイ、管理するためのコマンドラインツール
  - Lakehouse vs Data Warehouse - データ量、構造、処理要件に基づいてレイクハウスとデータウェアハウスを選択するためのガイド
- Azure Synapse Analytics - データウェアハウスおよびビッグデータシステム全体で洞察への時間を加速するエンタープライズ分析サービス
- Google Cloud BigQuery - データから価値を最大化するのに役立つ、マルチエンジン、マルチフォーマット、マルチクラウド向けに設計された完全に管理された AI 対応データ分析プラットフォーム
- Amazon QuickSight - ユーザーがデータを分析し、可視化を作成し、様々なエンタープライズデータソースから洞察を得ることができる AI 駆動型ビジネスインテリジェンスサービス

セマンティックレイヤー

- Cube - AI エージェントをデプロイしてデータをモデル化、分析、レポートするエージェント分析プラットフォーム
- Open Semantic Interchange (OSI) - 分析、AI、BI プラットフォーム全体でセマンティックメタデータ交換を可能にするセマンティックモデル交換の普遍的な標準

06 - AI・機械学習・LLM

AI と機械学習の基礎

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.1 データ・AIの戦略的活用 > データ・AI理解・活用

- AI パラダイム
  - Symbolic AI - 問題、ロジック、検索の高レベルシンボリック(人間が読み取れる)

表現に基づいた人工知能研究におけるすべてのメソッドの集合です
    - Neuro-symbolic AI - ニューラル方式とシンボリック方式を統合して両方の強みを組み合わせる人工知能のサブフィールドで、生データから学習可能でありながら説明可能性と明示的推論を保持する AI システムを実現します
        - AlphaGeometry - ニューラル言語モデルとシンボリック演繹エンジンで構成されたニューロシンボリックシステムであり、複雑な幾何定理の証明を見つけるために協力して動作します
    - Generative AI - 生成モデルを使用してテキスト、画像、ビデオ、またはその他の形式のデータを生成する人工知能のサブセットです
    - Causal AI - 因果モデルを構築し、相関だけでなく因果関係を使用して推論できる人工知能の技法です
    - Connectionism - コネクショニストネットワークまたは人工ニューラルネットワークとして知られた数学モデルを利用する人間の精神プロセスと認識の研究へのアプローチです
    - Expert system - 人間の専門家の意思決定能力をエミュレートするコンピュータシステムです
    - Artificial general intelligence - ほぼすべての認知タスクにおいて人間の能力に匹敵または超える人工知能の仮説的なタイプです
- 基本概念
    - Embedding - 複雑な高次元データを数値ベクトルの低次元ベクトル空間にマップする表現学習手法です
    - Transfer learning - あるタスクから得た知識を異なるが関連するタスクのパフォーマンス向上に再適用する機械学習手法です
    - Mathematical model - 数学的概念と言語を使用して具体的なシステムを抽象的に記述したものです
    - Mathematical optimization - 利用可能な選択肢のセットから、いくつかの基準に関して最良の要素を選択することです
    - Gradient descent - 微分可能な多変数関数を最小化するための1階反復アルゴリズムです
    - Loss function - イベントまたは1つ以上の変数の値を実数にマップし、直感的にイベントに関連する何らかのコストを表す関数です
    - Pattern recognition - データから抽出されたパターンに基づいて観測値にクラスを割り当てるタスクです
    - Feature engineering - 教師あり機械学習と統計モデリングの前処理ステップで、生データをより効果的な入力セットに変換します

- Overfitting - 特定のデータセットに密接に対応する分析の生成であり、追加データへのフィットまたは将来の観測の予測が信頼できない可能性があります
- Bias–variance tradeoff - モデルの複雑さ、予測精度、以前に見たことのないデータに対する予測能力の関係です
- Inductive bias - 学習者が遭遇していない入力の出力を予測するために使用する仮定のセットです
- Curse of dimensionality - 低次元設定では発生しない高次元空間でのデータ分析と整理の際に生じる現象です

機械学習

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 機械学習・深層学習

学習パラダイム

- Supervised learning - ラベル付きデータからアルゴリズムが学習する機械学習のパラダイムです
  - Classification - トレーニングデータセットに含まれる観測に基づいて、新しい観測がどの一連のカテゴリ(部分集団)に属するかを特定する問題です
    - Logistic regression - イベント発生の確率をモデル化する統計モデルで、イベントの対数オッズが1つ以上の独立変数の線形結合です
    - Support vector machine - データを分類および回帰分析するために分析する関連学習アルゴリズムを備えた教師あり学習モデルのセットです
    - Naive Bayes classifier - 機能間の強い(素朴な)独立性仮定でベイズの定理を適用した単純な確率分類器のファミリーです
    - Decision tree learning - 項目に関する観測からその目標値に関する結論に移行するために予測モデルとして決定木を使用する方法です
    - Ensemble learning - 複数の学習アルゴリズムを使用して、構成する学習アルゴリズムのいずれからも単独では得られるより良い予測パフォーマンスを取得する方法です
      - Random forest - トレーニング時に多数の決定木を構築することによって機能する分類、回帰、およびその他のタスク用のアンサンブル学習方法です
    - ROC curve - 判別閾値が変化すると二項分類器システムの診断能力を示すグラフィカルプロットです
  - Regression - 従属変数と1つ以上の独立変数の関係を推定するための統計プロセスのセットです

        - Ordinary least squares - 線形回帰モデルで未知のパラメータを選択するための線形最小二乗法のタイプです
        - Generalized linear model - 通常の最小二乗回帰の柔軟な一般化です
        - ARIMA model - 自己回帰移動平均(ARMA)モデルの一般化で、時系列データに適合して、データをより良く理解したり、シリーズの将来のポイントを予測したりします
- Unsupervised learning - ラベルなしのデータセットを使用してモデルを学習し、以前のトレーニングなしでそのデータに対して機能させることが許可される機械学習のタイプです
    - Principal component analysis - 探索的データ分析、視覚化、およびデータ前処理でアプリケーションを持つ線形次元削減手法です
    - K-means clustering - n個の観測をk個のクラスタに分割することを目指すベクトル量子化の方法で、各観測は最も近い平均を持つクラスタに属します
- Reinforcement learning - インテリジェントエージェントが累積報酬の概念を最大化するために環境で行動を取る方法に関する機械学習の領域です
    - Markov chain - 各イベントの確率が前のイベントで達成した状態に依存するイベントのシーケンスを説明する確率過程です
    - Markov decision process - 結果が部分的にランダムで、部分的に意思決定者の管理下にある状況で意思決定をモデル化するための数学的フレームワークです
    - Hidden Markov model - モデル化されているシステムが観測されない(隠された)状態を持つマルコフ過程と仮定される統計マルコフモデルです
    - Multi-armed bandit - 固定限定リソースのセットを競合する(代替)選択肢間に割り当て、期待ゲインを最大化する必要がある問題です
    - Value function - 数学的最適化と強化学習で使用される関数で、状態または行動に望ましさの尺度を割り当てます
- ML 概念と技法
    - Hyperparameter - 学習プロセスを制御するために使用される値を持つパラメータです
    - Hyperparameter optimization - 学習アルゴリズムの最適なハイパーパラメータセットを選択する問題です
    - Early stopping - 勾配降下法などの反復的な方法で学習者をトレーニングする場合にオーバーフィッティングを回避するための正則化の形式です
    - Cross-validation - 統計分析の結果が独立したデータセットにどの程度一般化されるかを評価するためのさまざまな同様のモデル検証手法です
    - Anomaly detection - データの大部分と大幅に異なることで疑いを生じさせる稀なアイテム、イベント、または観測の特定です

        - One-class classification - そのクラスのオブジェクトのみを含むトレーニングセットから主に学習することによって、すべてのオブジェクトの中から特定のクラスのオブジェクトを識別しようとする手法です
    - Recommender system - ユーザーがアイテムに与える「評価」または「好み」を予測しようとする情報フィルタリングシステムです
- ML フレームワーク & プラットフォーム
    - scikit-learn - Python プログラミング言語用の無料のソフトウェア機械学習ライブラリです
        - libsvm - サポートベクターマシンのライブラリです
    - ML.NET - .NET 開発者向けのオープンソースで、クロスプラットフォーム対応の機械学習フレームワークです
    - Crab - 推奨システムを構築するための Python ライブラリです
    - mlxtend - 日々のデータサイエンスタスクのための便利なツールの Python ライブラリです
    - Prophet - 時系列データの予測手順で、高速で完全に自動化された予測を提供します
    - Azure Machine Learning - モデルをより速く構築してデプロイするためのエンタープライズグレードの機械学習サービスです
    - Amazon SageMaker - 完全に管理されたインフラストラクチャ、ツール、およびワークフローを備えた任意のユースケースに対して機械学習(ML)モデルを構築、トレーニング、およびデプロイするサービスです
    - Gradio - フレンドリーなウェブインターフェイスを備えた機械学習モデルのデモを作成する最速の方法で、誰でもどこでも使用できます

ニューラルネットワークと深層学習

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 機械学習・深層学習

- ニューラルネットワークの基礎
    - Neural network - データのパターンを見つけるための機械学習で使用される計算モデルです
    - Tensor - 機械学習で使用される多次元配列として表現される数学的オブジェクトです
    - 活性化関数

        - Sigmoid function - 特性的な「S」字形またはシグモイド曲線を持つ数学関数です
        - Softmax function - K個の実数のベクトルをK個の可能な結果の確率分布に変換する関数です
    - Backpropagation - フィードフォワードニューラルネットワークのトレーニングに広く使用されているアルゴリズムです
    - Autoencoder - ラベルなしデータの効率的なコーディングを学習するために使用される人工ニューラルネットワークのタイプ(教師なし学習)です
    - Vanishing gradient problem - 勾配ベースの学習方法とバックプロパゲーションでニューラルネットワークをトレーニングするときに発生する困難で、勾配がバックプロパゲートされると縮小します
- 深層学習の概念とトレーニング
    - Deep Learning - 表現学習に基づいた人工ニューラルネットワークを使用した、より幅広い機械学習方法のファミリーの一部です
    - Stochastic gradient descent - 適切な平滑性プロパティを持つ目的関数を最適化するための反復的な方法です
    - Dropout (neural networks) - 訓練データに対する複雑な共適応を防ぐことで、人工ニューラルネットワークの過学習を低減する正則化手法です
    - Fine tuning - 転移学習へのアプローチで、事前学習されたモデルの重みが新しいデータで学習されます
    - LoRA (machine learning) - 事前学習されたモデルを特定のタスクに適応させるためのパラメータ効率的なファインチューニング技法で、大幅に少ない計算リソースを使用します

アーキテクチャ

- Recurrent neural network - ノード間の接続がサイクルを作成できる人工ニューラルネットワークのクラスで、一部のノードからの出力が同じノードへの後続入力に影響を与えることができます
    - LSTM - 人工知能と深層学習のフィールドで使用される人工ニューラルネットワークで、フィードバック接続によって区別されます
- Convolutional neural network (CNN) - 最も一般的に視覚画像を分析するために適用される人工ニューラルネットワークのクラスです
- Attention - ニューラルネットワークのコンテキストでの技法で、認知注意を模倣し、入力データの重要な部分を強化し、残りをフェードアウトします
    - FlashAttention - 高速でメモリ効率的な正確な注意メカニズムです

    - Transformer - マルチヘッド注意メカニズムに基づいた深層学習アーキテクチャです
- Mixture of experts - 複数のエキスパートネットワーク(学習者)を使用して、問題空間を均一な領域に分割する機械学習技法です
- 深層学習フレームワーク & 可視化
    - TensorFlow - 機械学習のためのエンドツーエンドのオープンソースプラットフォームです
        - TFDS - TensorFlow または Jax などの他の Python ML フレームワークで使用できるようにすぐに利用できるデータセットのコレクションです
        - Keras - 機械ではなく人間のために設計された Python ディープラーニング API です
    - PyTorch - 研究プロトタイピングから本番デプロイメントへのパスを加速するオープンソース機械学習フレームワークです
    - ONNX Runtime - クロスプラットフォーム、高パフォーマンス機械学習モデル推論エンジンで、多様なハードウェアとプラットフォーム全体でトレーニングと推論の両方を加速します
    - Windows ML - ONNX Runtime によって駆動され、NPU、GPU、CPU 全体でハードウェアアクセラレーションを使用したオンデバイスモデル実行を可能にする、Windows 用の統合された高パフォーマンスのローカル AI 推論フレームワークです
        - WinML CLI - AI モデルを Windows ML 用に変換して最適化し、多様な Windows デバイス全体でハードウェア最適化デプロイメントを実現するコマンドラインツールです
    - AttentionViz - Transformer 注意の全体ビューです
    - BertViz - NLP モデルの注意を視覚化するツールです
- 参考書
    - Neural Networks and Deep Learning - ニューラルネットワークと深層学習の背後にある核となるアイデアを説明する無料のオンライン書籍です
    - Deep Learning, MIT Press - 学生と開発者が機械学習の分野、特に深層学習に入るのを支援することを目的とした教科書です

自然言語処理 (NLP)

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 機械学習・深層学習

- 言語学の基礎
  - Computational linguistics - 自然言語の計算モデリング、および言語的な質問への適切な計算的アプローチの研究に関する学際的な分野です
  - Morphology - 言葉がどのように形成されるか、および同じ言語の他の言葉との関係の研究です
  - Syntax - 言葉と形態素がどのようにして句や文などのより大きな単位を形成するかの研究である言語学的分野です
  - Semantics - 言葉がどのように意味を獲得するか、および複雑な表現が構成要素から意味をどのように導出するかを調べる言語的意味の研究です
  - Distributional semantics - 言語データの大規模サンプルの分布プロパティに基づいて言語アイテム間の意味的類似性を定量化および分類するための理論と方法を開発および研究する研究領域です
  - Symbol grounding problem - 抽象シンボルを、それらが表現する実世界のオブジェクトまたは概念に接続するという課題です

概念とベクトル表現

- Okapi BM25 - 検索エンジンが確率的検索フレームワークに基づいて特定の検索クエリへのドキュメントの関連性を推定するために使用されるランキング関数です
- Levenshtein distance - 一つを他に変更するために必要な単一文字編集の最小数を数えることによって 2 つのシーケンス間の差を測定するための文字列メトリックです
- n-gram - 特定の順序で n 個の隣接するシンボルのシーケンスで、自然言語処理および計算生物学などの分野で使用されます
- tf-idf (term frequency-inverse document frequency) - 情報検索で使用される統計尺度で、一般的な頻度を考慮して、コレクションまたはコーパスに対する相対的なドキュメント内の単語の重要性を評価します
- Word embedding - 自然言語処理における単語の表現で、通常は単語の意味をエンコードする実数値ベクトルで、ベクトル空間で近い単語は意味で類似していると予想されます
  - Word2vec - 自然言語処理での単語の周囲の単語に基づいて意味に関する情報をキャプチャする単語のベクトル表現を取得するための技法です
  - fastText - 効率的なテキスト分類と表現学習のためのライブラリです
  - GloVe - 単語表現のためのグローバルベクトルです
- Sentence embedding - 有意な意味情報をエンコードする文の数値ベクトル表現です
- NLP ライブラリ
  - Natural Language Toolkit - 人間言語データを操作する Python プログラムを構築するための主要なプラットフォームです

    - Gensim - ドキュメントを意味ベクトルとして表現するための無料のオープンソース Python ライブラリです
    - Kuromoji - Java で記述されたオープンソース日本語形態素解析器です
    - Kagome - 純粋な golang で記述されたオープンソース日本語形態素解析器です
    - mecab-python3 - 日本語テキスト用の MeCab 形態素解析器の Python ラッパーです
    - jieba - 中国語テキスト分割用の Python モジュールです

コンピュータビジョン

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 機械学習・深層学習

画像処理と基礎

- OpenCV - オープンソースのコンピュータビジョンおよび機械学習ソフトウェアライブラリです
    - GoCV - OpenCV 4 のバインディングを備えた Go プログラミング言語用のパッケージです
- Pillow - 画像処理機能を追加する Python 画像処理ライブラリです
- scikit-image - Python での画像処理用のアルゴリズムのコレクションです
- ImageMagick - デジタル画像を作成、編集、構成、または変換するためのツールです

3D ビジョン & ジオメトリ

- Open3D - 3D データを扱うソフトウェアの迅速な開発をサポートするオープンソースライブラリです
- Point Cloud Library (PCL) - 2D/3D 画像とポイントクラウド処理用のスタンドアロンで大規模なオープンソースプロジェクトです

物体検出と認識

- YOLO (You Only Look Once) - リアルタイム検出用に設計されたエンドツーエンドのオブジェクト検出モデルです
- Detectron2 - Facebook の次世代ライブラリで、最先端の検出およびセグメンテーションアルゴリズムを提供します
- Mask R-CNN - Faster R-CNN の拡張で、バウンディングボックスと並行してオブジェクトマスク予測用のブランチを追加します

ポーズと身体推定

- MediaPipe - ポーズ推定、手の追跡、顔検出などの知覚タスク用のマルチモーダル ML パイプラインを構築するためのフレームワークです
- OpenPose - リアルタイムマルチユーザーキーポイント検出が可能なライブラリです

顔と生体認証認識

- face_recognition - Python またはコマンドラインから顔を認識および操作する Python ライブラリです

OCR とテキスト認識

- Tesseract OCR - オープンソーステキスト認識(OCR)エンジンです
  - gosseract OCR - Tesseract C++ ライブラリを使用した OCR(光学文字認識)用の Go パッケージです
- EasyOCR - 80 以上のサポートされている言語とすべての一般的なスクリプトを備えた、すぐに使用できる OCR です
- OCRmyPDF - PDF ファイルに検索可能な OCR テキストレイヤーを追加するツールです
- オープンモデル
  - LLaVA - ビジョンエンコーダと Vicuna を結合した、エンドツーエンドでトレーニングされた新しい大規模マルチモーダルモデルで、汎用的な視覚的および言語理解を実現し、マルチモーダル GPT-4 の精神を模倣した印象的なチャット機能を実現し、Science QA で最先端の精度を設定します

音声とオーディオ処理

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.2 AI・データサイエンス > 機械学習・深層学習

自動音声認識 (ASR)

- Kaldi - 音響モデリングと言語モデリング用の C++ で記述された音声認識ツールキットです
- DeepSpeech - 深層学習で駆動されるスピーチテキスト変換のオープンソースエンジンです
- wav2vec 2.0 - 音声認識事前学習の自己教師あり学習アプローチです
- オープンモデル

    - Whisper - 大規模な弱い監督を通じてトレーニングされた堅牢な音声認識モデルです

テキスト音声変換 (TTS) & 音声合成

- gTTS (Google Text-to-Speech) - Google のテキストから音声への API とインターフェイスする Python ライブラリおよび CLI ツールです
- Tacotron 2 - 注意を持つシーケンスツーシーケンスモデルに基づいた音声合成用のニューラルネットワークアーキテクチャです

オーディオ処理とツール

- librosa - オーディオおよび音楽分析用の Python ライブラリです
- PyAudio - クロスプラットフォームオーディオ I/O ライブラリである PortAudio の Python バインディングです
- SoX (Sound eXchange) - 音声処理プログラムのスイスアーミーナイフです
- Praat - 音声学での音声分析用のソフトウェアプログラムです

音声活動検出 & オーディオ強化

- webrtcvad - WebRTC ボイスアクティビティディテクタの Python ラッパーです
- SileroVAD - 大規模なデータセットでトレーニングされたオープンソースボイスアクティビティディテクタです

スピーカー識別と音声生体認証

- Resemblyzer - スピーカーエンベディングベースのスピーカー認識用の Python パッケージです
- PyannoteAudio - スピーカーダイアライゼーションとスピーカー変化検出用の Python ツールキットです

生成 AI & 大規模言語モデル (LLMs)

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.1 データ・AIの戦略的活用 > AI実装・運用

- 基本概念
    - Vision Language Models (VLM) - 画像とテキストを理解できる興味深いクラスのモデルです
    - Diffusion model - 段階的なノイズプロセスを逆にして新しいデータを生成するこ

とを学習する機械学習の潜在変数生成モデルのクラスです

- Multimodal learning - テキスト、オーディオ、画像、またはビデオなどの多様なデータタイプを組み合わせて処理し、複雑な情報のより全体的な理解を実現する深層学習アプローチです

モデルプロバイダーとアグリゲーター

- Anthropic - Anthropic の Claude モデルへのアクセスを提供する API です
- OpenAI - OpenAI のモデルでアプリケーションを構築するためのプラットフォームです
- Gemini Developer APIs - Google の最新 Gemini モデルへのアクセスを提供する API です
- Hugging Face Serverless Inference API - Hugging Face Hub でホストされているモデルで推論を許可する API です
- OpenRouter - LLM 向けの統合インターフェイスです
- オープンモデル
  - Llama - どこでもファインチューニング、蒸留、デプロイできるオープンソース AI モデルです
  - Gemma - Gemini モデルの作成に使用されたのと同じ研究と技術から構築された軽量で最先端のオープンモデルのファミリーです
  - Mistral - オープンソースおよび商用の生成 AI モデルのファミリーです
  - OLMo - 最先端で本当にオープンな言語モデルと言語モデルの科学を構築および研究するためのフレームワークです
  - Kimi K3 - 2.8 兆パラメータと 100 万トークンのコンテキストウィンドウを備えた、オープンウェイトのネイティブマルチモーダルエージェントモデルです
  - DeepSeek V4-Pro - 1.6 兆パラメータ (アクティブ 490 億) と 100 万トークンのコンテキストウィンドウを備えた、MIT ライセンスでオープンウェイトの Mixture-of-Experts (MoE) 言語モデルです

標準とモデル形式

- GGUF - GGML および GGML ベースのエグゼキューター推論用のモデルを保存するためのファイル形式です
- ONNX - 機械学習モデルを表すために構築されたオープン形式です
- Safetensors - テンサーを安全に保存するためのシンプルな形式です

モデルインターフェース SDK とクライアント

- ライブラリ & SDK
  - Go OpenAI - OpenAI API の Go クライアントライブラリです
  - Google Gen AI SDK - Google の生成 AI モデル用の Python SDK です
  - Instructor - 大規模言語モデル(LLM)から構造化された検証済みデータを抽出するために設計された Python ライブラリです
  - OmniAI - LLM とのインターフェイスのための最小限のライブラリです
  - Outlines - JSON スキーマ、正規表現、またはコンテキストフリー文法への準拠を強制することにより、任意の大規模言語モデルからの生成中に構造化出力を保証するライブラリです
  - Ruby OpenAI - OpenAI API 用の Ruby ラッパーです
  - RubyLLM - GPT、Claude、Gemini など向けの 1 つの美しい Ruby API です

技術と方法

- Retrieval-augmented generation (RAG) - 大規模言語モデルが外部データソースから新しい情報を取得および組み込むことを可能にする技法です
  - dsRAG - 非構造化データ用の高パフォーマンス検索エンジンです
- モデル圧縮
  - Knowledge Distillation - 大規模な機械学習モデルから小さいモデルへの知識転送のプロセスです
  - AWQ (Activation-aware Weight Quantization) - LLM 圧縮と加速のための効率的で正確な低ビット重み量子化(INT3/4)です
  - Pruning (artificial neural network) - 既存の人工ニューラルネットワークからパラメータを削除する慣行です
- GraphRAG - LLM の力を使用して非構造化テキストから有意で構造化されたデータを抽出するように設計されたデータパイプラインおよび変換スイートです
- Prompt Engineering Guide - 大規模言語モデルを効果的に利用し、AI エージェントを構築するためのプロンプトエンジニアリング技法の学習と適用のための包括的なリソースです
- CRAFT framework - コンテキスト、役割、アクション、形式、トーンを定義することで、明確で正確な AI プロンプトを作成するための構造化された方法です

開発ツール & ユーティリティ

- LiteLLM - 統一された OpenAI 入出力形式を使用して 100 以上の大規模言語モデル(LLM)を呼び出すことができる Python SDK および AI ゲートウェイ(プロキシ)で

す

- RedCandle - Rust の Candle を使用してローカルで最先端の言語モデルを実行するための Ruby Gem です
- Unsloth AI - 大規模言語モデル(LLM)を簡単にファインチューニングおよびトレーニングして、より速く、より効率的な AI トレーニングを実現するためのツールおよびサービスを提供するプラットフォームです

ベンチマークと分析

- ARC-AGI - 人間には直感的であるが AI システムには困難な新しいタスクでのスキル取得効率をテストすることで、人工汎用知能への進歩を測定するベンチマークです
- OSWorld - Ubuntu、Windows、macOS にわたってウェブアプリケーション、デスクトップソフトウェア、クロスアプリケーションワークフロー全体にまたがる 369 のオープンエンドタスクでマルチモーダルエージェントを評価するためのスケーラブルな実際のコンピュータ環境です
- FrontierMath - 専門の数学者が作成した非常に困難な数学的問題から構成される AI ベンチマークで、学部レベルから研究レベルの難易度の範囲から非常に困難な数学的問題を含むオープン研究問題を含みます
- MRCR v2 - LLM が拡張マルチターン会話履歴から情報を検索および使用する能力を評価するマルチラウンドコンテキスト呼び出しベンチマークです
- Artificial Analysis - AI モデルと API プロバイダーの独立した分析で、ユーザーが AI ランドスケープを理解するのに役立ちます
- Arena - 大規模言語モデル(LLM)とビジョン言語モデル(VLM)を含む様々な AI モデルをベンチマークして比較するために設計されたプラットフォームです

エージェント AI

> **Relevant DSS-P Skills**
>
> - 2. データ整備・活用 > 2.1 データ・AIの戦略的活用 > データ・AI活用戦略設計
> - 2. データ整備・活用 > 2.1 データ・AIの戦略的活用 > AI実装・運用

標準とプロトコル

- AGENTS.md - AI エージェントを定義して実行するためのオープン標準です
- Agent Skills - エージェントに新しい機能と専門知識を付与するためのシンプルでオープンな形式です
- llms.txt - LLM が推論時に Web サイトを利用するのに役立つ情報を提供するため、/llms.txt ファイルの使用を標準化する提案です

- Model Context Protocol (MCP) - AI アプリケーションを外部システムに接続し、データソース、ツール、ワークフローへのアクセスを可能にするためのオープンソース標準です
- A2A Protocol - ウェブアプリケーションと AI エージェント間の双方向通信を可能にするプロトコルです
- Agent Name Service (ANS) - 公開鍵インフラストラクチャ(PKI)を活用した安全な DNS に着想を得た AI エージェント発見のためのフレームワークで、通信用の構造化 JSON スキーマ、および A2A、MCP、ACP プロトコルをサポートするプロトコルアダプタレイヤーを備えています
- GitAgent - Git リポジトリ内のバージョン管理されたファイルのセットとして AI エージェントを定義できるフレームワークに依存しない標準です
- AG-UI (Agent–User Interaction Protocol) - AI エージェントがユーザー向けアプリケーションに接続する方法を標準化するオープンで軽量なイベントベースのプロトコルです

エージェントパターンと技法

- ReAct Prompting - 言語モデルで推論と行動を相乗効果させるプロンプト技法です
  - Reason、Act、Thought、Observation
- Recursive Language Models - 言語モデル(LM)が無制限の長さの入力コンテキストを分解して再帰的に相互作用できる推論戦略です
- Effective Context Engineering for AI Agents - LLM 推論中に最適なトークン(情報)セットをキュレーションおよび維持するための戦略のセット(プロンプト以外の他の情報を含みます)です

エージェントオーケストレーション & フレームワーク

- オーケストレーションフレームワーク
  - Agno - マルチエージェントフレームワーク、ランタイム、およびコントロールプレーンです
  - crewAI - 開発者がハイパフォーマンス AI エージェントを簡単かつスケーリングして調整できるようにするオープンソースのマルチエージェント調整フレームワークです
  - Deep Agents - LLM を使用したエージェントとアプリケーションの構築を開始する最も簡単な方法で、タスク計画、コンテキスト管理用ファイルシステム、サブエージェント生成、および長期メモリ用の組み込み機能を備えています
  - Hermes Agent - 経験からスキルを作成し、使用中に改善し、永続的なメモリを維持し、セッション全体でユーザーの深まるモデルを構築する組み込み学習ループを備えた自己改善 AI エージェントです

    - LangGraph - LLM を使用してステートフルなマルチアクターアプリケーションを構築するためのライブラリで、エージェントランタイムの循環グラフを作成します
    - Mastra - ワークフロー、RAG、メモリ、および可観測性の組み込みサポートを備えた、AI エージェントを構築、反復、デプロイするためのオールインワンのオープンソース TypeScript フレームワークです
    - Microsoft Agent Framework - 技術的進歩とともに進化する堅牢で将来性のある Agentic AI ソリューションを構築するためのリソースです
    - Microsoft 365 Agents SDK - Microsoft 365 Copilot、Teams、サードパーティプラットフォーム、カスタムアプリケーション、ウェブサイト全体でシームレスに動作するフルスタックのマルチチャネルエージェントを構築するための包括的なフレームワークです
- アプリケーションフレームワーク
    - Chainlit - 本番対応の会話型 AI を構築するためのオープンソース Python パッケージです
    - DSPy - 構造化コードでの高速反復を可能にし、AI プログラムを言語モデルの効果的なプロンプトと重みに変換するアルゴリズムを提供する、モジュラー AI ソフトウェア構築用の宣言的フレームワークです
    - Genkit - エージェント、RAG、ツール使用の統合サポートを備えた、フルスタックアプリケーション構築用の AI フレームワークです
    - LangChain - 大規模言語モデルを使用したアプリケーション開発のためのフレームワークです
    - LlamaIndex - 信頼できる低および高レベルの抽象化を備えた GenAI アプリケーションの本番投入までの時間を急速に加速する開発者優先のエージェントフレームワークです
        - LlamaParse - 複雑なレイアウト、表、グラフ、手書き、チェックボックス、および画像をきれいな markdown に変換するドキュメントパーサーです
    - PydanticAI - 型安全性と構造化出力を強調する生成 AI を使用した本番グレードアプリケーション構築用の Python エージェントフレームワークです

エージェント開発キット (ADK)

- Agent Package Manager (APM) - スキル、プロンプト、および指示の単一の情報源を提供する AI エージェント向けのオープンソース依存関係マネージャーです
- Agent Development Kit (ADK) - AI エージェント開発とデプロイメント向けの柔軟でモジュラーなフレームワークです
- Claude Agent SDK - Claude Code を強力にするのと同じツール、エージェントループ、コンテキスト管理を提供する Agent SDK で、Python と TypeScript でプログラム可能です

- Claude Agent Skills - 命令、メタデータ、およびオプションのリソースをパッケージ化することで、エージェントの機能を拡張するモジュラー機能です
- Fantasy - 単一の API を通じて複数のプロバイダーとモデルで AI エージェントを構築するための Go ライブラリです
- Go Micro - 1 つのランタイムでエージェント、サービス、およびワークフローを構築できる Go 用のエージェントハーネスです
- Microsoft 365 Agents Toolkit - Microsoft 365 Copilot、Teams、Office、ウェブ、およびその他のサードパーティメッセージングチャネル全体で動作するエンタープライズ対応のエージェントおよびアプリを構築するためのツールスイートです
- OpenAI Agents SDK - モデルが追加のコンテキストとツールを使用し、特殊なエージェントに手渡し、結果をストリーミングできるエージェント的アプリケーション構築用のライブラリです

サポートサービス & プラットフォーム

- エージェントプラットフォームとサービス
  - Foundry Agent Service - エンタープライズ変革のためのガバナンスと可観測性を備えた AI エージェントを安全に設計、デプロイ、スケールするためのプラットフォームです
  - Moltbook - AI エージェントが共有、議論、投票するソーシャルネットワークです
- 相互運用性
  - FastMCP - Model Context Protocol(MCP)サーバーとクライアントをビルドするための Python フレームワークです
- 事前構築済みエージェント & コレクション
  - agency-agents - ユニークな音声、証明されたワークフロー、測定可能な成果物を持つ特殊な専門家として機能するように設計された、精密に作成された AI エージェント個性の成長するコレクションです
- メモリシステム
  - Mem0 - LLM アプリケーション向けの AI メモリレイヤーで、パーソナライズされた AI エクスペリエンスの提供を目指しています
  - Graphiti - 時間認識コンテキストグラフを構築するためのオープンソース Python フレームワークです
  - Hindsight - AI エージェント専用に設計されたメモリシステムで、マルチ戦略検索と自動知識統合を使用して複数のセッション全体で情報を保持、リコール、および推論できるようにしています
- 検索とデータ抽出

    - Firecrawl - URL を取得し、クロールして、クリーンな markdown または構造化データに変換する API サービスです
    - Tavily Search - LLM 向けに最適化された検索エンジンで、効率的で迅速で永続的な検索結果を目的としています
    - SWIRL AI Search - 100 以上のエンタープライズプラットフォームに接続し、コストのかかるデータ変換や移行を必要とせずに、データおよびドキュメントサイロ全体でリアルタイムの可視性を提供するフェデレーション AI 検索ソリューションです

- セキュリティツール
    - skill-scanner - AI Agent Skills 向けのベストエフォート型セキュリティスキャナーで、プロンプトインジェクション、データ流出、悪意あるコードパターンを検出します
    - Toxic Flow Analysis (TFA) - AI アプリケーションの攻撃面を削減し、間接的なプロンプトインジェクションおよびその他の MCP 攻撃ベクトルを軽減するための初の原理的アプローチです

MLOps & LLMOps

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > SREプロセス
> - 2. データ整備・活用 > 2.1 データ・AIの戦略的活用 > AI実装・運用

ML ライフサイクルと MLOps プラットフォーム

- ML ライフサイクルツール
    - DVC - 機械学習プロジェクト用のオープンソースデータバージョン管理です
    - CML - 機械学習プロジェクトで継続的インテグレーション & デリバリー(CI/CD)を実装するためのオープンソースツールです
    - MLFlow - ML ライフサイクル管理のためのオープンソースプラットフォームで、実験、再現性、デプロイメント、および中央モデルレジストリを含みます
    - KubeFlow - Kubernetes 用機械学習ツールキットで、Kubernetes 上の ML ワークフローのデプロイメントをシンプル、ポータブル、スケーラブルにすることに専念しています
    - Weights & Biases - AI エージェント、アプリケーション、モデルを自信を持って構築するための AI 開発者プラットフォームです
    - ZenML - トレーニングパイプラインからエージェント評価まで、ローカルから Kubernetes へのオーケストレーション、バージョン管理、ガバナンス用のオープ

ンソースフレームワークです

- BentoML - 任意のモデルを任意の場所にデプロイするためのオープンソースフレームワークで、カスタマイズされた最適化、効率的なスケーリング、および合理化された操作を備えています

- 管理対象 MLOps プラットフォーム
  - Microsoft Foundry - AI アプリケーションとエージェントを構築、最適化、管理するための統合された相互運用可能なプラットフォームで、ビジネスコンテキストを理解し、ビジネスインパクトを提供します
  - Vertex AI - ML モデルと AI アプリケーションをトレーニングしてデプロイするための機械学習(ML)プラットフォームです
  - Nebius - AI および機械学習ワークロード向けの目的別インフラストラクチャを提供する専門 AI クラウドプラットフォームです
  - Amazon SageMaker - データ、分析、AI を一堂に集め、スケールでモデルを構築、トレーニング、デプロイするための完全管理型サービスです

LLM サービングとランタイム

- ローカル LLM ランタイム
  - LM Studio - コンピュータでローカルに LLM を開発し、実験するためのデスクトップアプリです
  - LocalAI - 無料のオープンソース OpenAI 代替です
  - Ollama - LLM をローカルにデプロイして管理するために設計されたツールです
  - llama.cpp - Apple Silicon から NVIDIA GPU まで、ハードウェア全体で LLM 推論を実行するための依存関係なし C/C++ 実装です
  - Jan - コンピュータでローカルに AI モデルを実行する ChatGPT 代替のオープンソースです
- 本番サービングエンジン
  - vLLM - 大規模言語モデル(LLM)の高スループットでメモリ効率的な推論とサービングエンジンです
  - Hugging Face TGI - テキスト生成推論用の Rust、Python、gRPC サーバーで、Hugging Chat と Inference API を強力にするために本番環境で使用されます
  - SGLang - 大規模言語モデルとマルチモーダルモデル向けの高パフォーマンスサービングフレームワークです
  - KServe - Kubernetes 上での機械学習モデルの標準化された分散推論プラットフォームで、スケーラブルな、マルチフレームワークデプロイメント用です

    - Modular MAX - GPU カーネルから API エンドポイントまで、多様なハードウェア全体で完全な最適化を提供する統合 AI 推論プラットフォームです

LLMOps & 可観測性

- 管理対象モデルサービス
    - Amazon Bedrock - 高パフォーマンスな基盤モデルの選択肢を提供する完全管理型サービスです
        - Amazon Bedrock Agents - 基盤モデルの推論、API、およびデータを使用してユーザーリクエストを分割し、関連情報を収集し、タスクを効率的に完了するサービスです
- エージェント可観測性
    - Mission Control - AI エージェントのフリートを管理、監視、調整するための中央オペレーションコントロールプレーンです
- LLM 可観測性 & 評価
    - Langfuse - LLM アプリケーションをデバッグおよび改善するためのトレース、評価、プロンプト管理、メトリクスを提供するオープンソース LLM エンジニアリングプラットフォームです
    - OpenLIT - LLM および GenAI 可観測性のためのオープンソース OpenTelemetry ネイティブツールです
    - LangSmith - LLM アプリケーションの開発、協調、テスト、デプロイメント、監視のための統合 DevOps プラットフォームです
    - Helicone - 開発者が生成 AI アプリケーションを監視および最適化するための、オープンソース LLM 可観測性プラットフォームです
    - Arize Phoenix - LLM 向けのオープンソース AI 可観測性および評価プラットフォームです
    - Braintrust - LLM でビルドする開発者向けの評価および可観測性プラットフォームを提供するエンタープライズ AI プラットフォームです
    - Ragas - AI アプリケーションの体感チェックから体系的な評価ループへの移行を支援するライブラリです
    - DeepEval - 研究を支持するメトリクスを備えた LLM 評価フレームワークで、CI/CD で実行される pytest ネイティブ評価です

07 - 開発者の基礎スキル

ソフトウェア開発手法

> **Relevant DSS-P Skills**

> - 3. テクノロジー > 3.1 ソフトウェア開発 > ソフトウェア開発プロセス
> - 5. パーソナルスキル > 5.2 コンセプチュアルスキル > 適応力

アジャイル開発

- Agile software development - The Agile Alliance が合意した価値観と原則を反映した、ソフトウェア開発アプローチの総称です
  - Agile Manifesto - プロセスやツールよりも個人と対話を、包括的なドキュメントよりも動くソフトウェアを、契約交渉よりも顧客との協調を、計画に従うことよりも変化への対応を重視することで、より良いソフトウェア開発の方法を宣言した文書です
  - Software prototyping - 開発中のソフトウェアプログラムの不完全なバージョン、すなわちソフトウェアアプリケーションのプロトタイプを作成する活動です
  - Minimum viable product - 将来のプロダクト開発のためのフィードバックを提供できる初期の顧客が使用できるだけの、十分な機能を備えたプロダクトのバージョンです
  - User story - ソフトウェアシステムの機能を非公式な自然言語で記述したものです
    - Card, Conversation, Confirmation - Card が要件を表すトークンであり、Conversation が詳細を引き出す場であり、Confirmation がストーリーの受け入れテストであるプラクティスです
  - INVEST of PBI - 質の高いプロダクトバックログ項目 (PBI) の特徴を思い出すために Bill Wake が作成した記憶術です
    - Independent：PBI は自己完結しているべきです
    - Negotiable：草案の PBI は明示的な契約ではなく、議論の余地を残すべきです
    - Valuable：PBI はステークホルダーに価値をもたらさなければなりません
    - Estimable：PBI のサイズは常に見積もれなければなりません
    - Small：PBI はある程度の精度で計画・タスク化・順序付けができなくなるほど大きくあってはいけません
    - Testable：PBI またはその関連する記述は、テスト開発を可能にするために必要な情報を提供しなければなりません
- 主要な方法論
  - Extreme Programming - ソフトウェアの品質と、顧客要件の変化への対応力を向上させることを目的としたソフトウェア開発方法論です
  - Scrum - 人々が生産的かつ創造的に、可能な限り最高価値のプロダクトを届けながら、複雑で適応を要する問題に対処できるフレームワークです

    - Acceptance test-driven development - ビジネス顧客、開発者、テスターの間のコミュニケーションに基づいた開発方法論です
        - Three Amigos - プロダクトオーナー、開発者、品質テスターが集まり、プロジェクトのスコープについて明確化を図るミーティングです
    - Behavior driven development - ソフトウェアプロジェクトにおいて、開発者、品質保証テスター、顧客代表者間の協調を促すアジャイルソフトウェア開発プロセスです
        - Specification by example - 抽象的な記述の代わりに現実的な例を用いて要件を捉え説明することに基づいて、ソフトウェアプロダクトの要件とビジネス指向の機能テストを定義する協調的アプローチです
- 主要なプラクティス
    - Refactoring - 既存のコードベースを再構築するための規律あるテクニックで、外部の振る舞いを変えずに内部構造を変更します
        - Software rot - ソフトウェアが時間の経過とともに品質、パフォーマンス、または有用性が劣化していく傾向です
        - Technical debt - より時間のかかるより良いアプローチを使う代わりに、今簡単な（限定的な）解決策を選ぶことで生じる追加的な手戻りの暗黙的なコストを反映した、ソフトウェア開発における概念です
            - Technical Debt Ratio - コードベース全体をゼロから開発するコストと比較して、既存の技術的負債を修正するコストを測定するために使用されるメトリクスです
    - Test driven development - ソフトウェアが完全に開発される前にソフトウェア要件をテストケースに変換し、すべてのテストケースに対してソフトウェアを繰り返しテストすることであらゆるソフトウェア開発を追跡する、ソフトウェア開発プロセスです
- ATDD/BDD 向けツール
    - Gauge - 受け入れテストの作成と保守の手間を取り除く、無料でオープンソースのテスト自動化フレームワークです
    - Cucumber - プレーンテキストで書かれた実行可能な仕様を読み取り、ソフトウェアがその仕様どおりに動作することを検証することで、ビヘイビア駆動開発 (BDD) をサポートするツールです
        - Gherkin Syntax - Cucumber が理解できるほど構造化されたプレーンテキストにするための一連の文法規則です
        - cucumber-ruby - Cucumber の Ruby 実装です
    - RSpec - ビヘイビア駆動開発 (BDD) 向けに作成された、Ruby プログラミング言語用のテストツールです

- Behave - Python コードに裏打ちされた自然言語スタイルで書かれたテストを使用する、Python でのビヘイビア駆動開発 (BDD) 向けのツールです

リーン開発

- Lean software development - リーン生産方式の原則と実践をソフトウェア開発領域に翻案したものです
  - Continual improvement process - プロダクト、サービス、またはプロセスを改善するための継続的な取り組みです
  - OODA loop - 利用可能な情報をフィルタリングし、それを文脈に当てはめ、より多くのデータが利用可能になるにつれて変更が可能であることを理解しながら、迅速に最も適切な意思決定を行うことに焦点を当てた 4 段階の意思決定アプローチです
- Lean manufacturing - 生産システム内の時間だけでなく、サプライヤーからおよび顧客への応答時間を削減することを主な目的とした生産方式です
  - The 7 Wastes：顧客に価値を付加しない活動です
  - Value-stream mapping - プロダクトまたはサービスが特定のプロセスの開始から顧客に届くまでの一連の出来事について、現状を分析し将来の状態を設計するリーンマネジメント手法です
  - ECRS (Eliminate, Combine, Rearrange, Simplify) - 生産ラインへの解決策の即時適用を可能にする柔軟な継続的改善戦略です
- Toyota Production System - トヨタが開発した経営哲学と実践から構成される、統合された社会技術システムです
  - Kanban - 人間系全体で作業を管理・改善するためのリーン手法です
  - Kaizen - CEO から組立ライン作業員まで全従業員が関わる、あらゆる機能の継続的かつ漸進的な改善に焦点を当てた哲学です
  - Autonomation - プロセスを部分的に自動化し、問題が検出されると自動的に停止するように機器を設計することで、オペレーターが問題を即座に修正できるようにする実践です
  - Heijunka - 生産量と製品構成の両方を平準化することで生産を平滑化する手法です
  - Genchi Genbutsu - 状況を真に理解するには、作業が行われている「現場」に行き、プロセスを観察し、自ら事実を確認する必要があるとする原則です
  - Andon (manufacturing) - 生産ラインの状態を示すために使用される視覚的管理システムです
  - Muda (Japanese term) - リソースの最適な配分からの逸脱を表す 3 種類のうちの 1 つとして、リーンプロセスマネジメントにおける重要な概念であり、無駄、

無用、浪費を意味します

- Theory of Constraints - 管理可能などんなシステムも、ごく少数の制約によってその目標達成がより制限されているとみなす経営パラダイムです

DevOps とエンジニアリング生産性

- 概念
  - CALMS framework - Culture（文化）、Automation（自動化）、Lean（リーン）、Measurement（測定）、Sharing（共有）の略で、DevOps へのアプローチのための概念モデルです
- 文化的・組織的な基盤
  - Generative organizational culture - 高度な信頼と協調、ミッションに対する責任の共有意識、そして学習と継続的改善への注力を特徴とする文化の一種です
- 技術的なプラクティス
  - Feature Toggles - コードを変更せずにシステムの動作を修正できるようにする強力なテクニックです
  - Blue-Green Deployment - Blue と Green と呼ばれる 2 つの同一の本番環境を稼働させることで、ダウンタイムとリスクを削減するテクニックです
  - Canary Release - 変更を全インフラストラクチャに展開する前に、少数のユーザーサブセットに徐々に展開することで、本番環境に新しいソフトウェアバージョンを導入するリスクを削減するテクニックです
  - Everything as code - ネットワークインフラストラクチャ、ドキュメント、構成を含む開発ライフサイクルのあらゆる側面の保守性とスケーラビリティを向上させるために、バージョン管理、テスト、デプロイメントと同じ原則を適用しようとするソフトウェア開発プラクティスです

リリース自動化

- semantic-release - 形式化されたコミットメッセージに基づいて次のバージョン番号を決定し、リリースノートを生成し、パッケージを公開する、完全に自動化されたバージョン管理・パッケージ公開ツールです
- Release Please - Conventional Commits に基づいて、プロジェクトの変更履歴生成、GitHub リリースの作成、バージョンアップを自動化するツールです
- GoReleaser - Go プロジェクト向けのリリース自動化ツールです
- Changesets - モノレポに重点を置いたバージョニングと変更履歴を管理するツールです

リリースの慣習と標準

- keep a changelog - プロジェクトの各バージョンにおける注目すべき変更点を、時系列順に整理してまとめたファイルです
- Conventional Commits - 明示的なコミット履歴を作成するための簡単なルールセットを提供する、コミットメッセージ上の軽量な規約です
- Semantic Versioning - バージョン番号の割り当て方と増分方法を規定する、シンプルなルールと要件のセットです
- CalVer - 恣意的な番号の代わりに、プロジェクトのリリースカレンダーに基づいたバージョニング規約です

オープンエコシステム

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

オープンソース

- Open Source Initiative - オープンソースソフトウェアとそのコミュニティの推進と保護に専念する非営利団体です
- 主要なパブリックライセンス
  - MIT - 1980 年代後半にマサチューセッツ工科大学 (MIT) で生まれた寛容なソフトウェアライセンスです
  - BSD - 対象ソフトウェアの使用と配布に最小限の制限を課す、寛容なフリーソフトウェアライセンスのファミリーです
  - Apache - Apache Software Foundation (ASF) によって書かれた寛容なフリーソフトウェアライセンスで、ロイヤリティを気にすることなくソフトウェアを使用、配布、修正することをユーザーに許可します
  - GPL - ソフトウェアを実行、研究、共有、修正する自由をエンドユーザーに保証する、広く使用されている一連のフリーソフトウェアライセンス、すなわちコピーレフトライセンスです
  - LGPL - Free Software Foundation (FSF) が公開したフリーソフトウェアライセンスで、開発者や企業が自身の（プロプライエタリなものであっても）ソフトウェアに、このライセンスの下でリリースされたソフトウェアコンポーネントを使用・統合する際、自身のコンポーネントのソースコードを公開する必要がないようにします
  - AGPL - ソフトウェアやその他の種類の著作物向けのフリーでコピーレフトなライセンスで、サーバー上で修正済みのプログラムを実行し、他のユーザーがそこでそのプログラムと通信できるようにする場合、サーバーはそこで実行されている修正

版に対応するソースコードのダウンロードもユーザーに許可しなければならないことを保証するために特別に設計されています

    - SSPL - ライセンスされたソフトウェアをサービスとして提供する事業者に対し、サービス全体の完全なソースコードの公開を義務付ける、強力なコピーレフトソフトウェアライセンスです

- 原則と格言
    - Linus's law - 「十分な目玉があれば、すべてのバグは深刻ではない」という主張であり、オープンソース開発における重要な原則です
- ソースリポジトリ
    - GitHub - 安全なソフトウェアを構築、拡張、提供するための AI 駆動の開発者プラットフォームです
    - GitLab.com - 組織がソフトウェア開発の全体的な投資対効果を最大化できるようにする DevSecOps プラットフォームです
- パッケージレジストリ
    - CTAN - Comprehensive TEX Archive Network です
    - CPAN - Comprehensive Perl Archive Network です
    - CRAN - Comprehensive R Archive Network です
    - PyPI - Python プログラミング言語向けのソフトウェアリポジトリです
    - RubyGems.org - Ruby コミュニティの gem ホスティングサービスです
    - npm Registry - 世界最大のソフトウェアレジストリです
    - JSR - モダンな JavaScript と TypeScript 向けのオープンソースパッケージレジストリです
    - pkg.go.dev - Go のパッケージとモジュールに関する情報源です
    - crates.io - Rust コミュニティの crate レジストリです
    - LuaRocks - Lua モジュール向けのパッケージマネージャーです
    - Hackage - Haskell コミュニティのオープンソースソフトウェアの中心的なパッケージアーカイブです
        - Stackage - Hackage からのパッケージの厳選されたセットです
    - NuGet Gallery - .NET 向けのパッケージマネージャーです
    - Maven Central - 世界最大かつ最古のコンポーネントリポジトリです
    - ConanCenter - コミュニティによって作成されたすべてのオープンソースパッケージを見つけられる中心的なリポジトリです
    - Anaconda Hub - データサイエンスと AI コラボレーションのためのハブです

オープンデータ

- ツールとライセンス
  - Creative Commons - 世界の差し迫った課題に取り組むために、知識と創造性の共有に対する法的障壁を克服する手助けをする非営利団体です
  - Open Data Commons - オープンデータの公開、提供、利用を支援する一連の法的ツールとライセンスの拠点です
- オープンデータレジストリ
  - Hugging Face Hub - 90 万を超えるモデル、20 万のデータセット、30 万のデモを備えたプラットフォームで、人々が ML ワークフローで簡単に協業できます
  - Data.gov - 米国政府のオープンデータの拠点です
  - Kaggle - データサイエンスの目標達成を支援する強力なツールとリソースを備えた、世界最大のデータサイエンスコミュニティです
  - Registry of Open Data on AWS - AWS サービスを通じて公開されているデータセットを人々が簡単に見つけられるようにするサービスです
  - OpenML - オープンで協調的、摩擦のない自動化された機械学習環境です
  - OpenStreetMap - あなたのような人々によって作成され、オープンライセンスの下で自由に使用できる世界地図です
- データ検索エンジン
  - Google Dataset search - Web 上の何千ものリポジトリに保存されているデータセットをユーザーが見つけられるようにする検索エンジンです

コミュニティとガバナンス

- 傘下組織となるオープンソース財団
  - Linux Foundation - 開発者がオープンテクノロジープロジェクトのコーディング、管理、拡張を行うための中立で信頼できるハブを提供することで、Linux を支援、保護、標準化する非営利団体です
  - Apache Software Foundation - Apache HTTP Server を含む Apache ソフトウェアプロジェクトを支援する非営利法人です
  - Eclipse Foundation - 個人と組織のグローバルコミュニティのために、オープンソースソフトウェアの協業とイノベーションのためのビジネスフレンドリーな環境を提供する組織です
- 技術特化型財団
  - OpenJS Foundation - Appium、Dojo、jQuery、Node.js、webpack を含む 40 を超えるオープンソースプロジェクトの中立的な拠点です

    - Rust Foundation - Rust プログラミング言語とそのエコシステムの管理と発展に専念する独立した非営利団体です
    - Python Software Foundation - Python プログラミング言語を支える慈善団体です
    - PyTorch Foundation - オープンソース AI のためのコミュニティ主導のハブです
- クラウドと AI
    - Cloud Native Computing Foundation - クラウドネイティブコンピューティングを普遍的で持続可能なものにすることに専念するオープンソースソフトウェア財団です
    - Agentic AI Foundation (AAIF) - この重要な能力が透明かつ協調的に、そして主要なオープンソース AI プロジェクトの採用を促進する形で進化することを保証するための、中立でオープンな財団です
- Web とデータの標準
    - World Wide Web Consortium - Web の長期的な成長を保証するオープンスタンダードを開発する国際的なコミュニティです
    - WHATWG - HTML と関連技術の進化に関心を持つ人々のコミュニティです
    - The Open Group - 技術標準を通じてビジネス目標の達成を可能にするグローバルコンソーシアムです
- 倫理とデジタル権利
    - Free Software Foundation - コンピューターユーザーの自由を推進するという世界規模の使命を持つ非営利団体です
- コミュニティガバナンスと行動規範
    - Debian Constitution - Debian プロジェクトにおける意思決定のための組織構造を説明した文書です
    - Ubuntu Code of Conduct - Ubuntu コミュニティの一員としての行動を対象とした一連のガイドラインです
    - Mozilla Community Participation Guidelines - Mozilla コミュニティ内の参加者に対する期待事項をまとめた一連のガイドラインです
    - Contributor Covenant - Coraline Ada Ehmke によって作成された、フリー/オープンソースソフトウェアプロジェクトの貢献者向けの行動規範です

シェルとターミナル

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

Bash とその他のシェル

- Bash - Korn shell (ksh) と C shell (csh) の便利な機能を取り入れた sh 互換シェルです
  - Line editing - GNU コマンドライン編集インターフェイスの基本機能です
  - History - Bash の履歴展開機能です
  - Shell expansions - コマンドラインが単語に分割された後に実行される処理です
  - Pipelines - 制御演算子 '|' または '|&' のいずれかで区切られた 1 つ以上のコマンドのシーケンスです
  - Built-in commands - 新しいプロセスをフォークすることなく、シェルプロセス自体の中で実行されるコマンドです
  - Special variables - シェルによって設定または使用されるシェル変数のリストです
  - Built-in job control - プロセスの実行を選択的に停止（一時停止）し、後で実行を継続（再開）できる機能です
- Zsh - 対話的な使用のために設計されたシェルですが、強力なスクリプト言語でもあります
- fish-shell - Linux、macOS、その他一連のプラットフォーム向けのスマートでユーザーフレンドリーなコマンドラインシェルです
- PowerShell - コマンドラインシェル、スクリプト言語、構成管理フレームワークで構成されたクロスプラットフォームのタスク自動化ソリューションです
- nushell - 新しいタイプのシェルです

シェルユーティリティ

- 汎用シェルユーティリティ
  - coreutils = cat、ls、rm など Unix 系オペレーティングシステムに必要な多くの基本ツールを含む GNU ソフトウェアパッケージです
  - GNU parallel - 1 台以上のコンピューターを使用してジョブを並列実行するためのシェルツールです
  - rlwrap - readline ラッパーです
  - bash-completion - bash 向けのプログラム可能な補完関数のコレクションです
  - direnv - 現在のディレクトリに応じて環境変数をロード・アンロードできるシェルの拡張機能です
  - zoxide - よりスマートな cd コマンドです

    - Atuin - 既存のシェル履歴を SQLite データベースに置き換え、コマンドの追加コンテキストを記録し、オプションですべてのマシン間でシェル履歴を同期するツールです
    - wttr.in - curl、httpie、wget などのコマンドラインツールを通じて気象情報を提供する、コンソール指向の天気予報サービスです
- 検索ツール
    - findutils - GNU オペレーティングシステムの基本的なディレクトリ検索ユーティリティです
    - fzf - 汎用のコマンドラインファジーファインダーです
    - fd - find に代わるシンプルで高速、かつユーザーフレンドリーなツールです
    - grep - 正規表現に一致する行をプレーンテキストデータセットから検索するコマンドラインユーティリティです
    - ripgrep - 現在のディレクトリを再帰的に検索し、正規表現パターンを探す行指向の検索ツールです
    - silversearcher-ag - ack に似たコード検索ツールですが、より高速です
- シェルフレームワークとカスタマイズ
    - starship - あらゆるシェル向けの、ミニマルで高速、無限にカスタマイズ可能なプロンプトです
    - oh-my-bash - BASH 構成を管理するための、オープンソースでコミュニティ主導のフレームワークです
    - oh-my-zsh - Zsh 構成を管理するための、愉快でオープンソース、コミュニティ主導のフレームワークです
    - Zim Framework - 高速性とモジュール式の拡張機能を備えた Zsh 構成フレームワークです
    - Powerlevel10k - Zsh 向けのテーマです
    - Pure - 美しくミニマルで高速な ZSH プロンプトです
- シェルチュートリアル
    - LinuxCommand.com - Linux コマンドラインの使い方を学ぶのに役立つよう設計された、書籍やその他の資料を掲載したサイトです

ターミナルエミュレーター

- Terminal Emulators - 他の表示アーキテクチャの中でビデオ端末をエミュレートするコンピュータープログラムです
    - WaveTerm - オープンソースでクロスプラットフォーム、AI 統合されたターミナルです

    - kitty - 高速で機能豊富な GPU ベースのターミナルエミュレーターです
    - Rio Terminal - 21 世紀のためのモダンなターミナルです
    - Alacritty - 適切なデフォルト設定を備えつつ、広範なカスタマイズも可能なモダンなターミナルエミュレーターです
    - Terminator - 複数の GNOME ターミナルを 1 つのウィンドウにまとめ、好みのスタイルに合わせてターミナルを組み合わせ、再構成できるようにするターミナルエミュレーターです
    - Windows Terminal - 新しい Windows Terminal であり、オリジナルの Windows コンソールホストです
    - Mintty - Cygwin、MSYS または Msys2 とその派生プロジェクト、および WSL 向けのターミナルエミュレーターです
    - xterm - X Window System 向けのターミナルエミュレーターです
    - wterm - Zig+WASM コアを使用して DOM に直接レンダリングする Web 向けのターミナルエミュレーターで、ネイティブなブラウザーのテキスト選択、コピー/ペースト、アクセシビリティを実現します
- 技術とプロトコル
    - Pseudoterminal - プログラムが端末をエミュレートするために使用する、端末のようなインターフェイスを提供する一対の疑似デバイスです
    - ANSI escape code - ビデオテキスト端末上でカーソル位置、色、フォントスタイル、その他のオプションを制御するためのインバンドシグナリングの標準です
    - kitty keyboard protocol - 端末がその中で実行されているアプリケーションにキーボードイベントを送信するためのプロトコルです
    - iTerm2 image protocol - ターミナル内にインラインで画像を表示するためのカスタムエスケープコードです
- ターミナルフォント
    - Nerd Fonts - 開発者向けフォントに多数のグリフをパッチ適用するプロジェクトです
    - Share Tech Mono - Share ファミリーに基づいた等幅サンセリフ体です
    - Cascadia Code - プログラミング用合字を含む、楽しく新しい等幅フォントです

ターミナルユーティリティ

- マルチプレクサとセッション管理
    - screen - 複数のプロセス間で物理端末を多重化するフルスクリーンウィンドウマネージャーです

  - tmux - ターミナルマルチプレクサです
  - byobu - GPLv3 のオープンソースであるテキストベースのウィンドウマネージャー兼ターミナルマルチプレクサです
  - zellij - 必要なものがすべて揃ったターミナルワークスペースです
  - asciinema - ターミナルセッションを記録し、Web 上で共有するための無料でオープンソースのソリューションです
  - ttyd - Web 経由でターミナルを共有するためのシンプルなコマンドラインツールです
- コンソールファイルマネージャー
  - midnight commander - ビジュアルファイルマネージャーです
  - ranger - コンソール向けの VIM に着想を得たファイルマネージャーです
  - superfile - 非常に洗練されたモダンなターミナルファイルマネージャーです

Windows 上の Linux/Unix 系環境

- winpty - Windows コンソールプログラム向けに Unix ライクな VT100 コンソールインターフェイスを提供する Windows ソフトウェアパッケージです
- WSL - 別の仮想マシンやデュアルブートを必要とせずに、Windows マシン上で GNU/Linux 環境を実行できるようにする Windows の機能です
  - WSLg - 完全に統合されたデスクトップ体験の中で、Windows 上で Linux GUI アプリケーション (X11 と Wayland) を実行するサポートを可能にするプロジェクトです
- Git for Windows - Git SCM の全機能セットを Windows にもたらす、軽量なネイティブツール群です
- MSYS2 - ネイティブな Windows ソフトウェアのビルド、インストール、実行のための使いやすい環境を提供するツールとライブラリのコレクションです

バージョン管理システム

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

- 概念
  - Distributed Version Control - 完全な履歴を含むコードベース全体が、すべての開発者のコンピューターにミラーリングされるバージョン管理の形態です
- コア VCS とクライアント

    - Git - 小規模から非常に大規模なプロジェクトまで、あらゆるものを高速かつ効率的に扱うために設計された、無料でオープンソースの分散型バージョン管理システムです
        - local repository、remote repository
        - branch、tag、worktree
        - push、pull、fetch、rebase、reset、stash
        - staging、commit
    - Jujutsu (jj) - シンプルかつ強力な Git 互換の VCS です
    - gitoxide - イディオマティックで無駄がなく、高速かつ安全な Git の純粋な Rust 実装です
    - TortoiseGit - Git 向けの Windows シェルインターフェイスで、TortoiseSVN をベースにしています
    - git lfs - 大きなファイルをバージョン管理するためのオープンソース Git 拡張機能です
- ターミナル & UI ツール
    - Informative git prompt for bash and fish - 現在の git リポジトリに関する情報を表示する bash プロンプトです
    - lazygit - git コマンド向けのシンプルなターミナル UI です
    - gitui - git GUI の快適さを、そのままターミナルで提供するツールです
    - giff - ブランチ間の変更を表示・管理できる、対話的なリベース機能を備えたターミナルベースの Git 差分ビューアーです
    - Git Interactive Rebase Tool - Git 向けの改良されたシーケンスエディターです
- 履歴とメンテナンスツール
    - BFG Repo-Cleaner - Git リポジトリ履歴から不要なデータを除去するための、git-filter-branch に代わるよりシンプルで高速なツールです
    - git filter-repo - 履歴を書き換えるための汎用的なツールです
    - degit - シンプルなプロジェクトのひな形作成です
- コミットと変更履歴のツール
    - Git Lint - クリーンで読みやすく、デバッグ可能で保守可能なプロジェクト履歴を維持することで、Git コミットをリントするコマンドラインインターフェイスです
    - commitlint - コミットメッセージをリントすることで、チームがコミット規約を遵守する助けとなるツールです
    - git cliff - 高度にカスタマイズ可能な変更履歴生成ツールです

- フック管理
  - pre-commit - 多言語対応の pre-commit フックを管理・保守するためのフレームワークです
  - Lefthook - あらゆる種類のプロジェクト向けの、高速で多言語対応の Git フックマネージャーです
  - Husky - 超高速でモダンなネイティブ git フックです
  - lint-staged - ステージされた Git ファイルに対して、フォーマッタやリンターなどのタスクを実行するツールです

Git ホスティングサービス

- GitLab SCM - コードとプロジェクトの共同作業のための単一の信頼できる情報源です
  - GitLab CLI - GitLab をターミナルにもたらすオープンソースツールで、すでに git やコードで作業している場所のすぐそばで使用できます
- Gitea - Git ホスティング、コードレビュー、チームコラボレーション、パッケージレジストリ、CI/CD を含む、手間のかからないセルフホスト型のオールインワンソフトウェア開発サービスです
- Codeberg - フリー/オープンソースプロジェクト向けに Git ホスティングやその他のサービスを提供する、コミュニティ主導の取り組みです
- Forgejo - セルフホスト型の軽量なソフトウェアフォージです
- Soft Serve - コマンドライン向けの、美味しくセルフホスト可能な Git サーバーです
- Azure Repos - コードを管理するために使用できる一連のバージョン管理ツールです
- GitHub - 安全なソフトウェアを構築、拡張、提供するための AI 駆動の開発者プラットフォームです
  - GitHub CLI - プルリクエスト、Issue、GitHub Actions、その他の GitHub 機能をターミナルにもたらすオープンソースツールで、すべての作業を一箇所で行えます

ブランチングモデル

- Git Flow - プロジェクトのリリースを中心に構築されたブランチモデルで、長期間存続する develop ブランチと main ブランチに加え、feature・release・hotfix ブランチを用いて並行開発を整理します
- Trunk Based Development - 開発者が「trunk」と呼ばれる単一のブランチでコードを協業し、文書化されたテクニックを用いて他の長期的な開発ブランチを作成する圧力に抵抗する、ソースコード管理のブランチングモデルです
- GitHub Flow - 頻繁にデプロイするチーム向けに設計された軽量なブランチベースのワークフローです

- GitLab Flow - 機能駆動開発と機能ブランチを Issue トラッキングと組み合わせた、GitFlow に代わるよりシンプルな方法です

コードレビュー

- Google Engineering Practices Documentation - レビュアーと変更作成者の両方に向けた、Google のコードレビュープロセスとポリシーの包括的なガイドです
- Conventional Comments - 明瞭性の向上、誤解の低減、コメントの機械可読化のために、コードレビューフィードバックの構造化されたフォーマットを提供する標準です
- Danger - コードレビューにおけるチームの規範を自動化するツールです
- Gerrit - Git ベースの開発のための、オープンソースのパッチセット単位のコードレビュープラットフォームです

統合開発環境 (IDE)

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

- GUI ベース
  - Eclipse IDE - 業界に支えられコミュニティ主導で開発され、豊富なプラットフォームプラグインによる高い拡張性を備えた、無料でオープンソースの IDE です
  - Visual Studio Code - デスクトップ上で動作し、Windows、macOS、Linux で利用できる、軽量ながら強力なソースコードエディターです
    - Awesome VS Code - 素晴らしい VS Code パッケージとリソースの厳選リストです
    - GitLens - Visual Studio Code に組み込まれた Git 機能を強化する拡張機能です
    - Git Graph - リポジトリの Git グラフを表示し、そのグラフから Git アクションを実行するための拡張機能です
  - AI アシスタンスプラグイン
    - GitHub Copilot - より速く、より少ない労力でコードを書く手助けをする AI ペアプログラマーです
      - Awesome GitHub Copilot - 体験を向上させるための、コミュニティが貢献したエージェント、指示、スキル、フック、ワークフロー、プラグイン、キャンバス拡張機能、ツールのコレクションです
    - Gemini Code Assist - 開発ライフサイクル全体を対象とした AI 駆動のアシスタントです

        - Amazon Q Developer - ソフトウェア開発向けの、最も高性能な生成 AI 駆動のアシスタントです
        - Cline - 最先端の AI モデルを VS Code エディターに直接もたらすオープンソースの AI コーディングエージェントです
    - AI 統合 IDE
        - Cursor - プロプライエタリモデルおよび最先端モデルとのシームレスでエージェント的な統合のために設計された、VS Code の AI ネイティブなフォークです
        - Windsurf - 開発者と AI の作業が本当に一体となって流れ、まさに魔法のようなコーディング体験を可能にする場です
        - Zed - 人間と AI との高パフォーマンスな協業のために設計された次世代コードエディターです
- Web ベース
    - code-server - リモートサーバー上で動作し、あらゆる Web ブラウザーからアクセスできる VS Code インスタンスです
    - GitHub Codespaces - GitHub にネイティブな、完全に構成済みの安全なクラウド開発環境で、より速くコーディングを始められます
    - Replit - コーディング不要で、アイデアを数分でアプリに変えるプラットフォームです
- ターミナルベース
    - Vim - あらゆる種類のテキストの作成と変更を非常に効率的に行えるように構築された、高度に設定可能なテキストエディターです
        - motion and operators - カーソルを移動するコマンドと、テキストを削除・変更するために使用されるコマンドです
        - vim-plug - Vim の事実上の標準プラグインマネージャーです
        - NERDTree - vim 向けのツリーエクスプローラープラグインです
    - Neovim - 超拡張可能な Vim ベースのテキストエディターです
        - LazyVim - 💤 lazy.nvim を利用した Neovim セットアップで、構成のカスタマイズと拡張を容易にします
            - lazy.nvim - Neovim 向けのモダンなプラグインマネージャーです
        - NvChad - 堅実なデフォルト設定と美しい UI を提供する超高速な Neovim 構成です
        - neo-tree.nvim - ファイルシステムやその他のツリー状構造を管理するための Neovim プラグインです

        - colorful-winsep.nvim - Neovim 向けのカラフルなウィンドウ区切りです
        - mason.nvim - LSP サーバー、DAP サーバー、リンター、フォーマッターなどの外部エディターツールを単一のインターフェイスで簡単に管理できる Neovim プラグインです
        - telescope.nvim - リストに対する高度に拡張可能なファジーファインダーです
        - flash.nvim - 検索ラベル、強化された文字モーション、Treesitter 統合でコードナビゲーションを助けるプラグインです
        - nvim-llama - Neovim 向けの Ollama へのシンプルなインターフェイスです
    - Helix - モーダルエディターであり、異なるタスクに対して異なるモードを持つことを意味します
    - GNU Emacs - 拡張可能でカスタマイズ可能な、フリー/リブレなテキストエディター、そしてそれ以上のものです
        - MELPA - Milkypostman's Emacs Lisp Package Archive です
        - doomemacs - 頑固な火星人ハッカーのための Emacs フレームワークです
        - neotree - Emacs 向けのツリーエクスプローラーです
        - Treemacs - Emacs 向けのツリーレイアウトファイルエクスプローラーです
        - Spacemacs - コミュニティ主導の Emacs ディストリビューションです
    - チュートリアルとチートシート
        - OpenVim - インタラクティブな Vim チュートリアルです
        - Vim Adventures - VIM のキーボードショートカットに基づいたオンラインゲームです
        - Vim Cheat Sheet - Vim コマンドのクイックリファレンスガイドです

言語サーバー

- LSP - 自動補完、定義へのジャンプ、すべての参照の検索などの言語機能を提供するために、エディターまたは IDE と言語サーバーの間で使用されるプロトコルです
- pyright - Python 向けの静的型チェッカー兼言語サーバーです
    - Pylance - Visual Studio Code の Python 拡張機能と連携して高性能な言語サポートを提供する拡張機能です
- Ruby LSP - Ruby 向けの、独自の意見を持った言語サーバーです
- TypeScript Language Server - スタンドアロンの TypeScript および JavaScript 言語サーバーです
- Gopls - Go 言語向けの公式言語サーバーです

- rust-analyzer - Rust プログラミング言語向けの言語サーバーです
- Eclipse JDT Language Server - Eclipse JDT に基づいた Java 言語サーバーです
- clangd - C++ のコードを理解し、コード補完、コンパイルエラーの表示、定義へのジャンプなどのスマートな機能をエディターに追加する言語サーバーです

コード品質とリファクタリング

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

- 概念
  - SQALE method - ソフトウェアソースコードの品質評価を支援するための手法です
  - Cyclomatic complexity - プログラムの複雑さを示すために使用されるソフトウェアメトリクスです
  - Code refactoring - 外部から見た動作を変えることなく、既存のソースコードを再構成するプロセスです
  - Code smell - より深い問題を示唆するソースコードの特徴です
- コードメトリクスツール
  - scc - 多くのプログラミング言語でコード行数をカウントするツールです
  - cloc - 多くのプログラミング言語で空白行、コメント行、ソースコードの物理行をカウントするツールです

分析プラットフォーム

- SonarQube Server - 30 以上の言語、フレームワーク、IaC プラットフォームでコーディング上の問題を検出するために設計された、オンプレミスの分析ツールです
- CodeScene - 静的解析にとどまらず、Code Health メトリクスとチームのコードへの関わり方に関する行動分析を通じてソフトウェアの品質を可視化するコード分析プラットフォームです
- Codecov - コードカバレッジを測定し、あらゆる段階でコード品質の向上を支援するツールです
- GitLab Code Coverage - テストでカバーされているコードの割合を示すレポートです
- GitLab Code Quality - CodeClimate Engines を使用してプロジェクトのコード品質分析を提供する機能です

フォーマッター

- EditorConfig - コーディングスタイルを定義するためのファイル形式であり、エディターがそのファイル形式を読み取り、定義されたスタイルを遵守できるようにするテキストエディタープラグインのコレクションです
- Prettier - 独自の意見を持ったコードフォーマッターです
- Ruff - Rust で書かれた、非常に高速な Python リンター兼コードフォーマッターです
- Biome - JavaScript、TypeScript、JSX、JSON、HTML、CSS、GraphQL 向けの高速なフォーマッターと高性能なリンターを提供する Web 向けツールチェーンです
- gofmt - タブでインデントを行い、空白で位置を揃えて Go プログラムを整形するツールです
- clang-format - C/C++/Java/JavaScript/JSON/Objective-C/Protobuf/C# のコードを整形するツールです

リンター

- ESLint - JavaScript コードの問題を見つけて修正する手助けをするオープンソースプロジェクトです
- JSHint - JavaScript 向けの静的コード解析ツールです
- Biome - JavaScript、TypeScript、JSX、JSON、HTML、CSS、GraphQL 向けの高速なフォーマッターと高性能なリンターを提供する Web 向けツールチェーンです
- Pylint - Python 2 または 3 向けの静的コード解析ツールです
- Ruff - Rust で書かれた、非常に高速な Python リンター兼コードフォーマッターです
- Staticcheck - Go プログラミング言語向けの最先端のリンターです
- revive - Go 向けの高速で拡張可能な静的コード解析フレームワークです
- golangci-lint - Go 向けの高速なリンターランナーです
- RuboCop - Ruby の静的コード解析ツール（別名リンター）兼コードフォーマッターです
- Rust Clippy - よくある間違いを検出し、Rust コードを改善するためのリントのコレクションです
- PSScriptAnalyzer - PowerShell モジュールとスクリプト向けの静的コードチェッカーです
- ShellCheck - bash/sh シェルスクリプトに対して警告と提案を行う GPLv3 のツールです

- Stylelint - エラーを回避し、規約を徹底するのに役立つ強力な CSS リンターです
- vacuum - 超高速で軽量な OpenAPI リンター兼品質チェックツールです
- yamllint - YAML ファイル向けのリンターです
- ls-lint - 非常に高速なファイル・ディレクトリ名リンターです

コーディングスタイルガイド

- Google Style Guides - さまざまなプログラミング言語でソースコードを書くための一連の規約を提供する文書のコレクションです
- Style Guide for Python - メインの Python ディストリビューションの標準ライブラリを構成する Python コードのコーディング規約を示す文書です
- Ruby Style Guide - Ruby プログラミング言語向けのコミュニティ主導のスタイルガイドです
- Airbnb JavaScript Style Guide - JavaScript に対する、おおむね理にかなったアプローチです

開発者 AI と生産性

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

AI 生産性ツール

- Mods - AI の助けを借りてプログラムを書く手助けをするシンプルなツールです
- gptcli - ChatGPT 向けのコマンドラインインターフェイスです
- ShellGPT - AI 大規模言語モデル (LLM) を活用したコマンドライン生産性ツールです
- LLM - 大規模言語モデルと相互作用するための CLI ユーティリティと Python ライブラリです
- Fabric - AI を使って人間を拡張するためのオープンソースフレームワークです
- OpenCommit - 意味のあるコミットを一瞬で自動生成します
- AI Commits - AI であなたに代わって git コミットメッセージを書く CLI です
- lootbox - 「Code Mode」に着想を得た CLI で、LLM がツール呼び出しを使う代わりに TypeScript コードを書いて API を呼び出します

開発エージェント

- CLI コーディングエージェント

- Claude Code - コードベースを読み取り、ファイルを編集し、コマンドを実行し、開発ツールと統合する、エージェント的なコーディングツールです
  - Plan Mode - 読み取り専用の操作でコードベースを分析して計画を立てるよう Claude に指示するモードで、コードベースの探索、複雑な変更の計画、安全なコードレビューに最適です
  - Hooks - エージェントのライフサイクルの特定のポイントで自動的に実行される、ユーザー定義のシェルコマンドまたは LLM プロンプトです
  - Subagents - 特定の種類のタスクを処理する専門化された AI アシスタントで、独自のコンテキストの中でカスタムプロンプトとツールアクセスを備えて実行されます
  - Sandboxing - より安全で自律的なエージェント実行のためにファイルシステムとネットワークの分離を提供する機能で、OS レベルのプリミティブを使用してこれらの分離を実施し、絶え間ない権限プロンプトを削減します
  - Auto-memory - ビルドコマンド、デバッグの知見、アーキテクチャに関するメモ、ユーザーの好みに関するノートを手動介入なしで保存することで、エージェントがセッションをまたいで自動的に知識を蓄積できるようにする機能です
- Gemini CLI - Gemini の力を直接ターミナルにもたらすオープンソースの AI エージェントです
  - Conductor - Gemini CLI 向けの公式プロジェクトマネジメントツールです
  - Sandboxing - 潜在的に危険な操作をホストシステムから分離する機能で、macOS Seatbelt またはコンテナベースの分離手法を使用して、AI 操作と環境の間にセキュリティバリアを提供します
- Antigravity CLI - コードベースを理解し、許可を得て編集を行い、ターミナルから直接コマンドを実行する、ターミナルベースの AI コーディングエージェントです
- GitHub Copilot CLI - Copilot コーディングエージェントの力を直接ターミナルにもたらすツールです
- Cursor CLI - ターミナルから直接開発を進められるよう構築されたツールで、どの環境でも同じコマンドを使用できます
- Amp - あらゆる場所で動作するコーディングエージェント兼開発環境です
- Aider - 新しいプロジェクトを始めたり、既存のコードベース上に構築したりできる、ターミナル内の AI ペアプログラミングツールです
- Letta Code - ターミナルに常駐する、メモリファーストのコーディングエージェントです
- Deep Agents CLI - Deep Agents SDK 上に構築されたターミナルコーディングエージェントです

    - OpenCode.ai - 実際のリポジトリ内でコードの理解、編集、出荷を支援する対話的な TUI を提供する、ターミナル向けのオープンソース AI コーディングエージェントです
    - OpenAI Codex - ターミナルで動作する軽量なコーディングエージェントで、ローカルなコーディングアシスタントを提供します
    - Kimi Code - あらゆる開発ワークフローに組み込めるよう設計された Moonshot AI 製の CLI コーディングエージェントで、コードベース分析、ファイル操作、Web 検索、並列サブエージェントタスク処理をサポートし、kimi-k2.6 モデルを搭載しています
    - Crush - お気に入りのターミナル向けの華やかな AI コーディングエージェントです 💘
    - ForgeCode - 調査、計画、実行のためのマルチエージェントアーキテクチャを備えた ZSH 統合コーディングハーネスで、複数の LLM プロバイダーをサポートし、Terminal-Bench で最上位にランクされています
- デスクトップコーディングエージェント
    - goose - エンジニアリングタスクをシームレスに自動化するローカル AI エージェントです
    - Open Interpreter - コンピューター上でドキュメントを読み取り、編集し、作成できるエージェントと共に作業できるオープンソースのデスクトップエージェントです
    - Agent Zero - コードを書いて実行し、自身のコンピューター環境を管理し、自らの行動から学習することで複雑なタスクを解決する動的なツールセットを使用する、オープンソースの個人向け AI エージェントです
    - AionUI - ドキュメント生成機能を備えた、AI コーディングエージェント向けの Cowork アプリです
- 自律型コーディングエージェント
    - SWE-agent - 一連の言語モデルを搭載した、GitHub リポジトリのバグや問題を修正するツールです
    - mini-swe-agent - SWE-agent のより小型でアクセスしやすいバージョンです
    - Devin - 独自のサンドボックス環境内で複雑なエンジニアリングタスクを支援なしで処理できる自律型 AI ソフトウェアエンジニアです
    - Jules - 自律型コーディングエージェントです
    - Replit Agent - IDE 内であなたと共に学習し作業できる最初の開発者エージェントです

サポートツールとインフラストラクチャ

- プラットフォーム
  - OpenHands - AI を搭載したソフトウェア開発エージェント向けのプラットフォームです
  - Port - エンジニアリングのあらゆる側面を加速する自律的なワークフローを構築するためのエージェント的な開発者ポータルです
  - Antigravity - エージェント的な開発プラットフォームです
  - Warp - 開発を自動化するためのオープンプラットフォームです
- ベンチマーク
  - SWE-bench - GitHub から収集された実世界のソフトウェア問題で大規模言語モデルを評価するためのベンチマークです
  - Terminal-Bench - ソフトウェアエンジニアリング、機械学習、セキュリティ、データサイエンスをカバーする、AI エージェントのターミナル習熟度を定量化するための harbor ネイティブなベンチマークタスクのコレクションです
- コンテキストプロバイダー
  - Context7 - LLM と AI コードエディター向けに最新のドキュメントを提供するために設計された AI エージェント兼ツールです
  - LeanCTX - ファイル読み取りとシェル出力を最大 99% 圧縮することで AI コーディングアシスタントのトークン使用量を削減する、オープンソースのコンテキスト圧縮ツールで、Cursor や Claude Code など 29 以上の AI ツールで動作します
  - Context Mode - AI コーディングエージェント向けのコンテキストウィンドウ最適化です。ツール出力をサンドボックス化し、98% の削減を実現します
  - RTK - 一般的な開発コマンドで LLM のトークン消費を 60〜90% 削減する CLI プロキシです
- セマンティックコード検索
  - Serena - セマンティック検索と編集機能を提供する強力なコーディングエージェントツールキットです
- セッショントラッキング
  - Entire - git ワークフローにフックし、プッシュのたびに AI エージェントセッションをキャプチャして、コードがどのように書かれたか、各コミットの背後にある意図の検索可能な記録を作成する CLI ツールです

08 - OS とネットワークの基礎

OS の基本概念

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- System call - コンピュータプログラムが実行されているオペレーティングシステムのカーネルにサービスを要求する、プログラム的な方法です
- Protection ring - 障害や悪意ある動作からデータと機能を保護する仕組みです
- Daemon - 対話的なユーザーの直接の制御下ではなく、バックグラウンドプロセスとして実行されるコンピュータプログラムです
- Environment variable - 通常はオペレーティングシステムやマイクロサービスに組み込まれた機能を通じて、プログラムの外部で値が設定される名前付き変数です
- POSIX standard - オペレーティングシステム間の互換性を維持するために IEEE Computer Society が規定した標準規格群です

プロセス管理

- Process - 1 つ以上のスレッドによって実行されているコンピュータプログラムのインスタンスです
  - Thread - スケジューラによって独立に管理できる、プログラムされた命令列の最小単位です
  - Scheduling - タスクを実行するためにリソースを割り当てる行為です
  - Context switch - 後で復元して実行を再開できるように、プロセスまたはスレッドの状態を保存する処理です
  - Interrupt - イベントを適時に処理できるように、プロセッサに現在実行中のコードを中断させる要求です

プロセス間通信 (IPC)

- パイプ
  - Anonymous pipe - 単方向のプロセス間通信に使用できる、単方向 FIFO 通信チャネルです
  - Named pipe - Unix および Unix 系システムにおける従来のパイプ概念を拡張したもので、プロセス間通信の手法の 1 つです
- Shared memory - 複数のプログラム間の通信や重複コピーの回避を目的として、それらのプログラムから同時にアクセスできるメモリです
- Signal - 発生したイベントを通知するために、プロセスまたは同一プロセス内の特定のスレッドに送信される非同期通知です

- Unix domain socket - 同一ホストオペレーティングシステム上で実行されるプロセス間でデータを交換するためのデータ通信エンドポイントです

メモリ管理

- Virtual memory - あるマシンで実際に利用可能なストレージリソースの理想化された抽象化を提供するメモリ管理技術です
  - Memory paging - コンピュータがメインメモリで使用するデータを二次記憶装置に保存し取得するメモリ管理方式です
  - Page fault - 実行中のプログラムが、メモリ管理ユニットによってプロセスの仮想アドレス空間に現在マッピングされていないメモリページにアクセスした際に、コンピュータハードウェアが発生させる例外の一種です
  - Resident set size (RSS) - プロセスが占有するメモリのうち、メインメモリに保持されている部分です
  - Working set size (WSS) - プロセスの仮想アドレス空間内のページのうち、現在メインメモリに常駐しているものの集合です
- Page cache - 将来のデータ要求をより高速に処理できるように、データを保存するハードウェアまたはソフトウェアコンポーネントです

ストレージ管理

- Disk partitioning - 各領域を個別に管理できるように、二次記憶装置上に 1 つ以上の領域を作成することです
- Loop device - ファイルをブロックデバイスとしてアクセス可能にする擬似デバイスです
- File system - オペレーティングシステムがデータの保存や取得の方法を制御するために使用する方式とデータ構造です
  - Journaling file system - ファイルシステムの主要部分にまだコミットされていない変更を記録する循環ログ、すなわちジャーナルを保持するファイルシステムです
  - Path - ファイルシステム内の一意の場所を指定する、ファイルまたはディレクトリの名前の一般的な形式です
  - Glob pattern - ワイルドカード文字を用いてファイル名の集合を指定するパターンです
  - File handle/descriptor - ファイルや、パイプ・ネットワークソケットなどの他の入出力リソースを識別する一意の識別子です
  - Symbolic link - 絶対パスまたは相対パスの形式で他のファイルやディレクトリへの参照を含み、パス名解決に影響を与えるファイルの総称です

    - Permissions - コンピュータのユーザーがファイルシステムの内容を閲覧・変更・移動・実行する能力を制御する、多くの現代的なファイルシステムの機能です
        - Setuid - ユーザーが実行可能ファイルの所有者またはグループの権限で実行ファイルを実行できるようにする、Unix のアクセス権フラグです
        - Sticky bit - Unix 系システムのファイルやディレクトリに割り当てることができる、ユーザー所有権に関するアクセス権フラグです
    - Inode - ファイルやディレクトリなどのファイルシステムオブジェクトを記述する、Unix 形式のファイルシステムにおけるデータ構造です
- RAID - データの冗長化、性能向上、またはその両方を目的として、複数の物理ディスクドライブコンポーネントを 1 つ以上の論理ユニットに統合するデータストレージ仮想化技術です

ネットワークの基本概念とプロトコル

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- The OSI Model - システム間相互接続を目的とした標準規格開発の連携のための共通基盤を提供する概念モデルです

リンク層 (L2)

- Ethernet - 有線コンピュータネットワーキング技術のファミリーです
    - ARP - 特定のインターネット層アドレスに関連付けられた、MAC アドレスなどのリンク層アドレスを発見するために使用される通信プロトコルです
        - MAC address - ネットワークセグメント内の通信でネットワークアドレスとして使用するために、ネットワークインターフェースコントローラに割り当てられる一意の識別子です
    - VLAN - コンピュータネットワークにおいて、データリンク層で分割・分離されたブロードキャストドメインです

インターネット層 (L3)

- The Internet - ネットワークやデバイス間の通信にインターネットプロトコルスイートを使用する、相互接続されたコンピュータネットワークの世界的なシステムです
- IP - インターネットプロトコルスイートにおけるネットワーク層の通信プロトコルです
    - Link-local address - ホストが接続されているネットワークセグメントまたはブロードキャストドメイン内の通信でのみ有効なネットワークアドレスです

  - IP-multicast - 単一の送信で興味を持つ受信者グループにインターネットプロトコルデータグラムを送信する方式です
  - IPv6 - ネットワーク上のコンピュータに識別・所在システムを提供し、インターネット上でトラフィックを経路制御する通信プロトコルであるインターネットプロトコルの最新版です
    - Unique local address - アドレスブロック fc00::/7 内の IPv6 アドレスです
  - ICMP - インターネットプロトコルスイートにおける補助プロトコルです
  - ICMPv6 - インターネットプロトコルバージョン 6 向けの Internet Control Message Protocol の実装です
  - DHCP - ネットワークに接続されたデバイスに IP アドレスやその他の通信パラメータを自動的に割り当てるために、インターネットプロトコルネットワーク上で使用されるネットワーク管理プロトコルです
  - DHCPv6 - IPv6 ネットワークで動作するために必要な IP アドレス、IP プレフィックス、その他の設定データを、インターネットプロトコルバージョン 6 ホストに設定するためのネットワークプロトコルです
  - NAT - トラフィックルーティングデバイスを通過中のパケットの IP ヘッダー内のネットワークアドレス情報を変更することで、ある IP アドレス空間を別の空間にマッピングする手法です
  - NAT64 - IPv6 ホストと IPv4 ホスト間の通信を仲介する IPv6 移行メカニズムです
  - NDP - インターネットプロトコルバージョン 6 で使用される、インターネットプロトコルスイート内のプロトコルです
- ルーティング
  - Routing table - 特定のネットワーク宛先への経路を列挙した、ルーターまたはネットワークホストに保存されるデータテーブルです
  - CIDR - IP アドレスの割り当てと IP ルーティングのための手法です

トランスポート層 (L4)

- Network socket - コンピュータネットワークのネットワークノード内にあり、ネットワーク越しのデータ送受信のエンドポイントとして機能するソフトウェア構造です
- TCP - インターネットプロトコルスイートの主要プロトコルです
  - TCP window scale option - Transmission Control Protocol で許容される受信ウィンドウサイズを、従来の最大値である 65,535 バイトより大きくするためのオプションです
- UDP - インターネットプロトコルスイートの中核をなすメンバーです

- QUIC - UDP ベースの、ストリーム多重化された暗号化トランスポートプロトコルです

ネットワークアーキテクチャ

- Peer-to-peer - 参加者が集中管理システムに頼ることなく、直接リソースを共有する分散コンピューティングまたはネットワーキングアーキテクチャです

ドメインネームシステム (DNS)

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- DNS - インターネットや他のインターネットプロトコルネットワークを通じて到達可能なコンピュータ、サービス、その他のリソースを識別するために使用される、階層的かつ分散化されたネーミングシステムです
- mDNS - ローカルネームサーバーを含まない小規模なネットワーク内で、ホスト名を IP アドレスに解決するプロトコルです

ドメイン登録と検索

- IANA WHOIS Service - ドメイン名または IP アドレスの登録データを検索するサービスです
- Registration Data Access Protocol (RDAP) - ドメイン名レジストリや地域インターネットレジストリから登録データを配信する、コンピュータネットワーク通信プロトコルです
- Domain drop catching - ドメイン名の登録が失効した直後に、そのドメイン名を登録する行為です

サーバーとリゾルバの実装

- BIND (dnsutils) - 非常に柔軟でフル機能を備えた DNS システムです
- dnsmasq - 軽量で設定が容易な DNS フォワーダー、DHCP、ルーターアドバタイズメントサーバーです
- CoreDNS - プラグインを連鎖させる DNS サーバーです
- systemd-resolved - ローカルアプリケーションにネットワーク名前解決を提供するシステムサービスです
- mDNS の実装
  - Avahi - mDNS/DNS-SD プロトコルスイートを介して、ローカルネットワーク上でのサービス発見を容易にするシステムです

    - Bonjour - ゼロコンフィグレーションネットワーキングの Apple による実装です

クライアントツール

- BIND の一部
    - dig - DNS ネームサーバーに問い合わせるための柔軟なツールです
    - nslookup - インターネットドメインネームサーバーに問い合わせるプログラムです
- dog - コマンドライン DNS クライアントです
- Doggo - Go で書かれた、モダンなコマンドライン DNS クライアント (dig のようなもの) です
- マネージド DNS サービス
    - Amazon Route53 - 高可用性でスケーラブルなクラウド Domain Name System Web サービスです
    - Google Cloud DNS - コスト効率よくドメイン名をグローバル DNS に公開する、高性能でレジリエントなグローバル Domain Name System サービスです

メールシステム

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- Email - 電子デバイスを使用して人々の間でメッセージを交換する方法です
- SMTP - 電子メール送信のための通信プロトコルです
- POP - メールクライアントがメールサーバーから電子メールを取得するために使用される、アプリケーション層のインターネット標準プロトコルです
- IMAP - メールクライアントが TCP/IP 接続を介してメールサーバーから電子メールメッセージを取得するために使用される、インターネット標準プロトコルです
- MIME - ASCII 以外の文字セットのテキストをサポートするために、電子メールメッセージの形式を拡張する標準です
    - Quoted-printable encoding - 7 ビットのデータパスを使用して送信できるように、8 ビット ASCII 文字セットでデータを表現するエンコーディングです
    - Base64 - バイナリデータを基数 64 表現に変換することで ASCII 文字列形式として表現する、バイナリ-テキストエンコーディング方式群です

メールボックス形式

- Unix Mbox
- Maildir

サーバーソフトウェア (MTA/MDA)

- Postfix - 広く使われている Sendmail プログラムの代替として IBM の研究部門で誕生したメールサーバーです
- Maddy Mail Server - メールサービスを運用するために必要なすべての機能を実装した、オールインワンのメールサーバーです
- テスト用サーバー
  - Mailpit - 小型で高速、低メモリ、依存関係ゼロのマルチプラットフォームな、開発者向けメールテストツール兼 API です
- IMAP
  - Cyrus IMAP - 小規模から大規模の企業環境での使用向けに設計された、高いスケーラビリティを持つエンタープライズメールシステムです
  - Dovecot - Linux/UNIX 系システム向けのオープンソース IMAP・POP3 メールサーバーです

クライアントソフトウェアとユーティリティ

- TUI クライアントとユーティリティ
  - mailutils - 電子メールを扱うためのライブラリとユーティリティのセットです
    - mail command - メールを送受信するコマンドです
  - Himalaya - Rust で構築された CLI アプリケーションで、IMAP、Maildir、SMTP、OAuth 2.0、PGP 暗号化をサポートし、シェルコマンドを通じてメールを操作できます
  - Mutt - Unix オペレーティングシステム上で電子メールを読み書きするための、小さいながら非常に強力なテキストベースのプログラムです
  - swaks - 機能豊富で柔軟、スクリプト可能なトランザクション指向の SMTP テストツールです
  - msmtp - 配送のためにメールを SMTP サーバーへ送信する SMTP クライアントで、メールクライアント向けの sendmail 代替として設定できるように設計されています
  - Pop - ターミナルからメールを送信するためのライブラリです
  - GNU sharutils - シェルアーカイブを作成・展開するためのユーティリティー式です

- ライブラリ
  - go-mail - シンプルに使えて機能豊富な、Go 向けメールライブラリです
- GUI クライアント
  - Thunderbird - セットアップとカスタマイズが容易な無料のメールアプリケーションです
  - Sylpheed - シンプルで軽量ながら機能豊富、使いやすい電子メールクライアントです

スパムテストとレピュテーション

- mail-tester - スパム、不正な形式のコンテンツ、メールサーバー設定の問題についてメールをテストできる無料のオンラインサービスです
- Spamhaus Project - スパムおよび関連するサイバー脅威を追跡する非営利組織です
- クラウドサービス
  - Amazon SES - 開発者が任意のアプリケーション内からメールを送信できる、コスト効率が良く柔軟でスケーラブルなメールサービスです
  - SendGrid - 大規模なトランザクションメールおよびマーケティングメールの信頼性の高い配信を提供する、クラウドベースのメール配信プラットフォームです

Unix 系オペレーティングシステム

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- The Linux Kernel - Linux オペレーティングシステムの主要コンポーネントであり、コンピュータのハードウェアとプロセスの間の中核的なインターフェースです
  - スレッド
    - Pthreads - プログラミング言語から独立して存在する実行モデルであり、並行実行モデルでもあります
  - ファイルシステム
    - Ext4 - 多くの主要な Linux ディストリビューションのデフォルトファイルシステムです
    - XFS - Silicon Graphics, Inc が開発した、高性能なジャーナリングファイルシステムです
    - Btrfs - スナップショット、RAID、自己修復などの高度な機能を備えた、Linux 向けの copy-on-write ファイルシステムです

        - UnionFS - 他のファイルシステムに対する union mount を実装する、Linux、FreeBSD、NetBSD 向けのファイルシステムサービスです
        - OverlayFS - Linux 向けの union mount ファイルシステム実装です
        - proc.5 - カーネルのデータ構造へのインターフェースを提供する仮想ファイルシステムです
        - sysfs.5 - さまざまなカーネルサブシステム、ハードウェアデバイス、関連するデバイスドライバに関する情報をエクスポートする仮想ファイルシステムです
    - コンテナサポート
        - cgroups - プロセスを階層的なグループに編成し、さまざまな種類のリソースの使用量を制限・監視できるようにする Linux カーネルの機能です
        - namespaces - グローバルなシステムリソースをラップする抽象化であり、名前空間内のプロセスからはそのグローバルリソースの独立したインスタンスを持っているかのように見せます
        - lxc/rootfs - Linux カーネルのコンテインメント機能に対するユーザースペースインターフェースです
        - nsenter - 他のプロセスの名前空間内でプログラムを実行するコマンドです
    - FUSE (Filesystem in Userspace) - ユーザースペースプログラムがファイルシステムを Linux カーネルにエクスポートするためのインターフェースです
        - s3fs - Amazon S3 バケットをローカルファイルシステムとしてマウントできる FUSE ファイルシステムです
    - eBPF (Extended Berkeley Packet Filter) - 特権的なコンテキストでサンドボックス化されたプログラムを実行できる、Linux カーネルに起源を持つ革新的な技術です

Linux ディストリビューション

- 汎用
    - Arch Linux - シンプルで軽量なディストリビューションです
    - Debian - 完全な自由なオペレーティングシステムです
    - Fedora - ハードウェア、クラウド、コンテナ向けにコミュニティによって構築された、革新的でフリーかつオープンソースのオペレーティングシステムプラットフォームです
    - Gentoo - カスタマイズ性とパフォーマンスを重視した、非常に柔軟なソースベースの Linux ディストリビューションです
    - NixOS - パッケージと設定の管理に独自のアプローチを採用した Linux ディストリビューションです

    - openSUSE - デスクトップ、サーバー、コンテナ向けの無料 Linux オペレーティングシステムです
- サーバー向け
    - Ubuntu server - パブリッククラウド、オンプレミス、IoT デバイス向けの標準プラットフォームです
- デスクトップ向け
    - Debian ベース
        - Ubuntu desktop - デスクトップからクラウド、インターネットに接続されたあらゆるモノまで動作する、Linux ベースのオペレーティングシステムです
        - BunsenLabs Linux - 軽量で簡単にカスタマイズできる Openbox デスクトップを提供するディストリビューションです
    - Arch ベース
        - Manjaro Linux - 独立して開発された Arch オペレーティングシステムをベースにした、使いやすい Linux ディストリビューションです
        - Mabox Linux - 高速で軽量、機能的な Linux Desktop の「リラックスした」ローリングリリースで、Manjaro をベースに Openbox Window Manager を採用しています

BSD ディストリビューション

- FreeBSD - Berkeley Software Distribution (BSD) の流れを汲む、フリーでオープンソースの Unix 系オペレーティングシステムです
- NetBSD - フリーで高速、安全、高い移植性を備えた Unix 系のオープンソースオペレーティングシステムです
- OpenBSD - 極めて高いセキュリティと暗号技術を重視した、NetBSD のフォークとしてのフリーでオープンソースな Unix 系オペレーティングシステムです

システムサービスと認証

- Systemd - Linux オペレーティングシステム向けのシステムおよびサービスマネージャーです
    - journald - ログデータを収集し保存するシステムサービスです
    - hostnamed - ユーザープログラムからホスト名や関連するマシンのメタデータを制御するために使用できるシステムサービスです
    - networkd - ネットワークを管理するシステムサービスです
    - resolved - ローカルアプリケーションにネットワーク名前解決を提供するシステムサービスです

    - timesyncd - ローカルシステムクロックをリモートの Network Time Protocol サーバーと同期するために使用できるシステムサービスです
- linux-pam - Linux システム内のアプリケーションとサービスの認証タスクを処理するライブラリ群です

マシン仮想化

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- Virtualization - 仮想コンピュータハードウェアプラットフォーム、ストレージデバイス、コンピュータネットワークリソースなど、何かの仮想版を作成する行為です
- libvirt - 仮想化プラットフォームを管理するためのツールキットです

Type-1 ハイパーバイザー

- KVM - 仮想化拡張機能を備えた x86 ハードウェア上の Linux 向けの完全仮想化ソリューションです
- Hyper-V - Microsoft のハードウェア仮想化製品です
- Proxmox VE - エンタープライズ仮想化向けの完全なオープンソースサーバー管理プラットフォームです

Type-2 ハイパーバイザー

- VirtualBox - 企業利用にも家庭利用にも適した、強力な x86・AMD64/Intel64 仮想化製品です
- QEMU - 汎用のオープンソースマシンエミュレーター兼仮想化ソフトウェアです

CPU エミュレーター

- QEMU - 汎用のオープンソースマシンエミュレーター兼仮想化ソフトウェアです

コンピュータハードウェア

- 処理装置
    - Central processing unit (CPU) - 算術・論理・制御・入出力操作を実行することでプログラム命令を処理する、コンピュータの主要なプロセッサです
    - Graphics processing unit (GPU) - グラフィックスレンダリングと並列コンピューティングタスクの高速化のために設計された特殊な電子回路で、3D グラフィックス、AI/ML ワークロード、動画処理に広く使用されています

    - Neural processing unit (NPU) - ニューラルネットワークとコンピュータビジョンタスクを効率的に実行することで、人工知能と機械学習アプリケーションを高速化するために設計された特殊なハードウェアです
- CPU アーキテクチャ
    - x86-64 - x86 命令セットの 64 ビット版です
    - ARM64 - ARM アーキテクチャファミリーの 64 ビット拡張です
- CPU 拡張機能
    - x86 virtualization
    - Intel AMX

Linux ホスト管理

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

コアユーティリティ

- util-linux - Linux ユーティリティの雑多なコレクションです
    - lsblk - 利用可能なすべてのブロックデバイス、または指定したブロックデバイスの情報を一覧表示するコマンドです
    - lsns - 現在アクセス可能なすべての名前空間、または指定した名前空間の情報を一覧表示するコマンドです
    - swapon - ページングとスワッピングを行うデバイスを指定するために使用されるコマンドです
- rsync - 高速な増分ファイル転送を提供するオープンソースユーティリティです
- sudo - システム管理者が特定のユーザーに root または他のユーザーとしていくつかのコマンドを実行する権限を委任するためのものです
- shadow-utils - UNIX パスワードファイルをシャドウパスワード形式に変換するために必要なプログラムと、ユーザーおよびグループアカウントを管理するプログラムを含みます
    - useradd - ユーザーを追加するための低レベルユーティリティです
- jc - 一般的なコマンドラインツールやファイル形式の出力を JSON または Dictionary に変換する CLI ツール兼 Python ライブラリです
- aha - Ansi HTML Adapter です
    - NO_COLOR - コマンドラインソフトウェアで ANSI カラーを無効化するための環境変数です

- Vixie Cron - POSIX Cron のオープンソース実装です
  - Crontab.guru - cron のスケジュール式のための、素早くシンプルなエディタです
- logrotate - ログファイルの自動ローテーション、圧縮、削除、メール送信を可能にします
- Syslog - メッセージロギングのための標準です

プロセスとシステムの監視

- procps - 通常 /proc にある擬似ファイルシステムから情報を提供する、コマンドラインおよびフルスクリーンユーティリティのセットです
  - ps - 選択したアクティブなプロセスの情報を表示するコマンドです
  - top - 稼働中のシステムの動的なリアルタイムビューを提供するプログラムです
  - free - システム内の空きおよび使用中の物理メモリとスワップメモリの合計量を表示するコマンドです
  - vmstat - プロセス、メモリ、ページング、ブロック IO、トラップ、ディスク、CPU 活動に関する情報を報告するコマンドです
- psmisc - proc ファイルシステムを使用する小さなユーティリティのパッケージです
  - pstree - 実行中のプロセスをツリー形式で表示するコマンドです
  - killall - 指定したコマンドのいずれかを実行しているすべてのプロセスにシグナルを送信するコマンドです
- lsof - オープンファイルを一覧表示する (LiSt Open Files) コマンドです
- strace - Linux 向けの診断・デバッグ・学習用ユーザースペースユーティリティです
- inxi - フル機能を備えたシステム情報スクリプトです
- witr - 実行中のプロセスの因果的な系統と目的を説明するツールです
- パフォーマンスモニター
  - sysstat - Linux 向けのパフォーマンス監視ツールのコレクションです
    - iostat - システムの入出力デバイスの負荷を監視するために使用されるコマンドです
  - Monit - Unix システムを管理・監視するための小さなオープンソースユーティリティです
  - atop - Linux 向けの ASCII フルスクリーンパフォーマンスモニターです
  - smem - Linux システムにおけるメモリ使用状況について多様なレポートを生成できるツールです

時刻同期

- NTP - パケット交換方式で遅延が可変のデータネットワーク越しに、コンピュータシステム間でクロックを同期するためのネットワーキングプロトコルです
- chrony - Network Time Protocol の汎用的な実装です
- pool.ntp.org - 何百万ものクライアントに信頼性が高く使いやすい NTP サービスを提供する、タイムサーバーの大規模な仮想クラスタです

モダンな CLI 代替ツール

- gdu - Go で書かれた、コンソールインターフェースを備えた高速なディスク使用量アナライザーです
- lsd - 色、アイコン、ツリービュー、その他の書式オプションなど多くの機能を追加した GNU ls の書き換え版です
- eza - ls のモダンな代替です
- broot - ディレクトリツリーを閲覧・移動する新しい方法です
- bat - 翼の生えた cat(1) のクローンです
- dust - Rust で書かれた、より直感的な du のバージョンです
- dua - ディスク容量の使用状況を確認し、不要なデータを高速に削除するツールです
- duf - より優れた 'df' の代替です
- procs - Rust で書かれた ps のモダンな代替です
- htop - Unix システム向けの対話的なプロセスビューアです
- btop++ - Linux、macOS、FreeBSD 向けのリソースモニターです
- glances - curses または Web ベースのインターフェースを通じて大量の監視情報を提示することを目指す、クロスプラットフォームな監視ツールです
- neofetch - コマンドラインシステム情報ツールです

パッケージ管理ツール

- dpkg - Debian の基本パッケージ管理システムです
  - apt - Ubuntu、Debian、および関連する Linux ディストリビューションで deb パッケージのインストール、更新、削除などの管理を行うコマンドラインユーティリティです
- Pacman - Linux でソフトウェアパッケージを管理するユーティリティです
  - Yay - Go で書かれた AUR ヘルパーです
- yum - rpm システム向けの自動アップデーターおよびパッケージインストーラー/リムーバーです

- dnf - yum の次世代版です
- Homebrew - macOS (または Linux) に欠けていたパッケージマネージャーです
- pipx - 分離された環境で Python アプリケーションをインストール・実行するツールです
- Flatpak - Linux 上でサンドボックス化されたデスクトップアプリケーションを構築・配布・実行するためのシステムです
- Snapcraft - Linux カーネルを使用するオペレーティングシステム向けに Canonical が開発した、ソフトウェアのパッケージングとデプロイのシステムです
- arkade - お気に入りの DevOps CLI をダウンロードしたり、Kubernetes クラスタに helm チャートをインストールしたりできる、ポータブルなマーケットプレイスです
- Plaza - pacman、AUR、apt、dnf、Flatpak にまたがってパッケージを検索・インストール・管理できる、カスタマイズ可能で装飾可能なターミナル UI です

Linux ネットワーク管理

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

設定と管理

- iproute2 - Linux で TCP/IP ネットワーキングとトラフィック制御を行うためのユーティリティ集です
  - ip - ルーティング、ネットワークデバイス、インターフェース、トンネルを表示・操作するための主要なコマンドです
  - ss - ソケットを調査するユーティリティです
- net-tools (legacy) - Linux カーネルのネットワークサブシステムを制御するプログラム集です
  - ifconfig - ネットワークインターフェースを設定するために使用されるコマンドです
  - netstat - ネットワーク接続、ルーティングテーブル、インターフェース統計、マスカレード接続、マルチキャストメンバーシップを表示するコマンドです
- NetworkManager - libudev やその他の Linux カーネルインターフェースの上に位置し、ネットワーク設定のための高レベルインターフェースを提供するデーモンです
- Ubuntu NetPlan - ネットワーク設定の抽象化レンダラーです

分析とセキュリティ

- tcpdump - 強力なコマンドラインパケットアナライザーです
- wireshark - 世界で最も定評あるネットワークプロトコルアナライザーです
- nmap - ネットワーク探索とセキュリティ監査のためのオープンソースツールです
  - ncat - コマンドラインからネットワーク越しにデータを読み書きする、機能満載のネットワーキングユーティリティです
- traceroute - インターネットプロトコルネットワーク越しのパケットの経路を表示し、通過遅延を測定するコンピュータネットワーク診断ツールです

プロキシとゲートウェイ

- SOCKS Proxy - プロキシサーバーを介してクライアントとサーバー間でネットワークパケットを交換するインターネットプロトコルです
  - Dante - RFC 1928 および関連標準を実装した SOCKS サーバー兼 SOCKS クライアントです
  - tun2socks - TUN デバイスからのすべての接続を処理する、TCP・UDP 向けの SOCKS プロキシです
  - proxychains - 任意のアプリケーションが行う TCP 接続を、TOR や他の SOCKS4、SOCKS5、HTTP(S) プロキシなどのプロキシ経由に強制するツールです
- トンネリング
  - cloudflared - Cloudflare ネットワークからオリジンへトラフィックをプロキシするデーモンである Cloudflare Tunnel のコマンドラインクライアントです
  - ngrok - ローカルアプリケーションをインターネットに公開するために、リバースプロキシ、API ゲートウェイ、セキュアなトンネルを提供する統合イングレスプラットフォームです

ファイル共有とリモートアクセス

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

ファイルサーバーとプロトコル

- SMB - ネットワーク上のノード間でファイル、プリンター、シリアルポートへの共有アクセスを提供するネットワーク通信プロトコルです
  - Samba - Linux と Unix 向けの標準的な Windows 相互運用プログラム群です
- FTP - コンピュータネットワーク上でサーバーからクライアントへコンピュータファ

イルを転送するために使用される標準通信プロトコルです
    - vsftpd - Linux を含む UNIX 系システム向けの GPL ライセンスの FTP サーバーです
- SFTP - 信頼性のあるデータストリーム越しにファイルアクセス、ファイル転送、ファイル管理を提供するネットワークプロトコルです
    - SFTPGo - フル機能で高度に設定可能な SFTP サーバーで、オプションで HTTP/S、FTP/S、WebDAV をサポートします

リモートアクセスサーバーとプロトコル

- SSH - 保護されていないネットワーク越しにネットワークサービスを安全に運用するための暗号化ネットワークプロトコルです
    - openssh - SSH プロトコルによるリモートログインのための第一級の接続ツールです
- RDP - ネットワーク接続越しに他のコンピュータへ接続するためのグラフィカルインターフェースをユーザーに提供する、Microsoft が開発した独自プロトコルです
    - xrdp - オープンソースの Remote Desktop Protocol サーバーです
- RFB - グラフィカルユーザーインターフェースへのリモートアクセスのためのシンプルなプロトコルです
    - x11vnc - X11 向けの VNC サーバーです
    - TightVNC - 無料のリモートデスクトップアプリケーションです
- Mosh - 対話的な SSH ターミナルの代替です
- メッシュ VPN
    - Tailscale - レガシーな VPN、SASE、PAM を置き換え、リモートチーム、マルチクラウド環境、CI/CD パイプライン、エッジ・IoT デバイス、AI ワークロードを接続する、ID ベースのゼロトラスト接続プラットフォームです

09 - プログラミングの概念とパラダイム

ソフトウェア設計とアーキテクチャ

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > ソフトウェア設計手法

設計原則

- Orthogonality and DRY principle - システム内のあらゆる知識は単一の、曖昧さのない、権威ある表現を持たなければならないという原則です

- Separation of concerns - コンピュータプログラムを個別のセクションに分離するための設計原則です
- Design by Contract - ソフトウェアコンポーネントに対して形式的で正確かつ検証可能なインターフェース仕様を規定する、ソフトウェア設計のアプローチです
- Law of Demeter - ソフトウェア、特にオブジェクト指向プログラムを開発するための設計ガイドラインです
- Hyrum's Law - 暗黙のインターフェースの法則であり、API の利用者が十分な数に達すると、システムの観測可能なすべての振る舞いに誰かが依存するようになるというものです
- SOLID - The principle of OOD - オブジェクト指向設計をより理解しやすく、柔軟で保守しやすいものにすることを意図した、5 つの設計原則の頭字語です
  - Single responsibility
  - Open–closed
  - Liskov substitution
  - Interface segregation
  - Dependency inversion
- The Reactive Manifesto - アプリケーションが応答性、耐障害性、弾力性を持ちメッセージ駆動であることを目指す、システムアーキテクチャに対する首尾一貫したアプローチです
- Unix philosophy - ソフトウェア開発に対する文化的規範と哲学的アプローチの集合です
- KISS principle - ほとんどのシステムは複雑にするよりもシンプルに保つ方がうまく機能するとする設計原則です
- Wirth's law - ソフトウェアはハードウェアが高速化する以上に急速に低速化しているとするコンピュータ性能に関する格言です

設計のベストプラクティス

- Resource acquisition is initialization (RAII) - リソースのライフサイクルをオブジェクトの寿命に結びつけるプログラミングイディオムです
- Rob Pike's 5 Rules of Programming - 最適化の努力をどこに向けるべきかについての一連のルールであり、計測とデータ構造の重要性を強調するものです
- The Zen of Python - コンピュータプログラムを書くための 19 の指導原則を集めたもので、Python プログラミング言語の設計に影響を与えたものです
- The twelve-factor app - 現代的なクラウドプラットフォームでの展開に適した Software-as-a-Service アプリケーションを構築するための方法論です

デザインパターン

- Software design pattern - ソフトウェア設計において特定の文脈内で頻繁に発生する問題に対する、一般的で再利用可能な解決策です
- Entity–control–boundary - オブジェクト指向システムにおけるクラスの責務の構造化を助ける、ソフトウェア設計・分析で使用されるアーキテクチャパターンです
- Command Query Responsibility Segregation - データストアの読み取り操作と更新操作を分離するパターンです
- Fluent interface - メソッドチェーンに基づいてオブジェクト指向 API を設計する手法であり、ソースコードの可読性を通常の文章に近づけることを目標としています
- Model-view-controller pattern - 関連するプログラムロジックを 3 つの相互接続された要素に分割する、ユーザーインターフェース開発によく使われるソフトウェア設計パターンです
- Dependency injection - オブジェクトや関数が依存する他のオブジェクトや関数を受け取る設計パターンです

アーキテクチャスタイル

- Three-tier architecture - プレゼンテーション、アプリケーション処理、データ管理の各機能が論理的に分離された、クライアントサーバーアーキテクチャです
- Microservices architecture - 単一のアプリケーションを、それぞれが独自のプロセスで動作し軽量な仕組みで通信する小さなサービス群として開発するアプローチです
- Event-driven architecture - イベントの生成、検知、消費、およびそれへの反応を促進するソフトウェアアーキテクチャパラダイムです
- Resource-oriented architecture - リソースのネットワークという形でソフトウェアを設計・開発するための、ソフトウェアアーキテクチャおよびプログラミングパラダイムのスタイルです
- Background processing - メインアプリケーションの応答性を維持しながら、バックグラウンドでタスクを実行することです

アーキテクチャ記述

- System - 一定の規則に従って相互作用または相互関連し、統一された全体を形成する要素の集合です
  - Systems architecture - システムの構造、振る舞い、その他の視点を定義する概念モデルです
    - 4+1 architectural view model - 「複数の並行するビューの使用に基づいてソフトウェア集約型システムのアーキテクチャを記述する」ために使用されるビューモデルです

        - The C4 model - 学びやすく開発者に優しい、ソフトウェアアーキテクチャ図の作成アプローチです
        - UML - ソフトウェア集約型システムの成果物を可視化、仕様化、構築、文書化するためのグラフィカル言語です
        - Flowchart - ワークフローやプロセスを表現する図の一種です
    - Conway's law - 組織は自らのコミュニケーション構造を反映したシステムを設計するという格言です
- 関連標準
    - ISO/IEC/IEEE 42010 (Architecture description) - システムおよびソフトウェアのアーキテクチャ記述に関する国際標準規格です

ドメイン駆動設計 (DDD)

- Domain-driven design - そのドメインの専門家からの入力に従って、ドメインに一致するようにソフトウェアをモデル化することに重点を置いた、主要なソフトウェア設計アプローチです
    - Ubiquitous Language - 開発者とユーザーの間で共通かつ厳密な言語を構築する実践です
    - Anemic domain model - ドメインオブジェクトが検証、計算、ルールなどのビジネスロジックをほとんど、あるいはまったく含まないプログラミングのアンチパターンです
- Object-oriented analysis and design - オブジェクト指向プログラミングを適用し、ソフトウェア開発プロセス全体を通じて視覚的モデリングを使用することで、アプリケーション、システム、またはビジネスを分析・設計するための技術的アプローチです
    - Use case - 目標を達成するために、役割 (Unified Modeling Language ではアクターと呼ばれる) とシステムとの間の相互作用を定義する、一連の行動またはイベントステップの一覧です
- Ontology - 1 つ、複数、またはすべての議論領域を構成する概念、データ、実体間のカテゴリ、性質、関係を表現し、形式的に命名・定義したものです
    - Semantic network - ネットワーク内の概念間の意味的関係を表現するナレッジベースです
        - WordNet - 英語の大規模な語彙データベースです
- Database design - データベースモデルに従ったデータの編成です

コアプログラミング概念

> **Relevant DSS-P Skills**

> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

言語の仕組みと実行

- Source code - 通常はプレーンテキストとして、人間が読めるプログラミング言語を使って書かれた、コメントを含む場合もあるコードの集合です
- Statement - 実行すべき何らかの動作を表現する、命令型プログラミング言語の構文単位です
- Expression - 評価してその値を決定できる、プログラミング言語における構文的実体です
  - 演算子とオペランド
- Literal - ソースコード内で固定値を表現するための表記法です
  - テンプレート文字列またはリテラル
  - Heredoc - ソースコードの一部を、あたかも別のファイルであるかのように扱う、ファイルリテラルまたは入力ストリームリテラルです
- Constant - プログラムの通常の実行中に変更できない値です
- Variable - 値と呼ばれる既知または未知の量の情報を含む、シンボリックな名前と対応付けられた抽象的な記憶場所です
- Scope - コンピュータプログラム内で、名前とエンティティの束縛 (名前束縛) が有効となる領域です
- Data type - 通常は可能な値の集合と許される操作によって指定される、データ値の集合または分類です
  - Primitives - プログラミング言語が基本的な構成要素として提供する、他のデータ型を用いずに定義されるデータ型です
  - Nominal type system - データ型の互換性と同等性が明示的な宣言や型名によって決定される、型システムの主要な分類の 1 つです
  - Structural type system - 型の互換性と同等性が型の実際の構造や定義によって決定される、型システムの主要な分類です
  - Duck typing - 特定のメソッドやプロパティの有無に基づいて型の互換性を判定する、ダックテストの応用です
  - Union type - そのインスタンスに格納できる複数の許容されたプリミティブ型を指定するデータ型定義です
  - Type variance - 複合型のサブタイプとその構成要素のサブタイプとの間の関係です
  - Type safety - プログラミング言語が型エラーをどの程度抑止または防止するかの度合いです

- Reference - プログラムがコンピュータのメモリや他の記憶装置内の特定のデータに間接的にアクセスできるようにする値です
  - Null pointer - ポインタや参照が有効なオブジェクトを参照していないことを示すために保存された値です

メモリ管理

- Reference counting - リソースへの参照、ポインタ、ハンドルの数を保持するプログラミング技法です
- Garbage collection - コレクタがもはや使用されていないオブジェクトが占有するメモリの回収を試みる、自動メモリ管理の一形態です
- Smart pointer - ポインタを模倣しつつ、自動メモリ管理や境界チェックなどの追加機能を提供する抽象データ型です
- Memory safety - メモリアクセスを扱う際に、様々なソフトウェアのバグやセキュリティ脆弱性から保護されている状態です

制御フロー構造

- Control flow - 命令型プログラムの個々の文、命令、関数呼び出しが実行または評価される順序です
- Continuation - プログラムの計算処理の、ある時点以降の残りを表すデータ構造です
  - call-with-current-continuation - 特に Scheme プログラミング言語において継続を捕捉し呼び出すために使用される、制御フロー演算子です
- Exception handling - プログラムの実行中に発生する例外への対応処理です
- Finite-state machine - ある時点で有限個の状態のうちのちょうど 1 つを取り得る抽象機械を表現する、計算の数学的モデルです

基礎的な技法と特性

- State - ある瞬間において、コンピュータプログラムまたはシステムがアクセスできる、保存された情報です
- Function - 特定のタスクを実行する、1 つの単位としてまとめられたプログラム命令の並びです
  - Parameter - 入力として与えられるデータの一部を参照するために、サブルーチンや関数で使用される特別な種類の変数です
  - Anonymous function - 識別子に束縛されていない関数定義です
- Immutable object - 作成後に状態を変更できないオブジェクトです
- Generic Programming - 必要に応じて後からインスタンス化される、後で指定される型を用いてアルゴリズムを記述するプログラミングスタイルです

- Assertion - コード内のある地点で、述語（ブール値を返す関数）が常に真であることが期待されるという表明です
- Autovivification - 新しい変数やデータ構造が初めて使用される際に必要に応じて自動的に作成されることです

モジュール構造と組織化

- Cohesion - モジュール内の要素がどの程度まとまっているかの度合いです
- Coupling - ソフトウェアモジュール間の相互依存の度合いであり、2 つのルーチンやモジュールがどの程度密接に結びついているか、およびモジュール間の関係の強さを測る尺度です

プログラミングパラダイム

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

オブジェクト指向プログラミング

- Object-oriented Programming - オブジェクト、すなわちデータと関数をカプセル化するソフトウェアエンティティに基づくプログラミングパラダイムです
  - Abstraction - 問題に適したクラスをモデル化し、最も関連性の高い詳細レベルで作業することで、システムの複雑さを隠す過程です
  - Encapsulation - データとそのデータを操作するメソッドをひとまとめにすること、またはオブジェクトの一部の構成要素への直接アクセスを制限することです
  - Polymorphism - 異なる型のエンティティに対して単一のインターフェースを提供することです
    - Dynamic dispatch - 実行時にポリモーフィックな操作（メソッドまたは関数）のどの実装を呼び出すかを選択する処理です
  - Inheritance - 類似した実装を保持したまま、あるオブジェクトやクラスを別のオブジェクトやクラスに基づいて構成する仕組みです
  - Class - 状態の初期値と振る舞いの実装を提供する、オブジェクトを作成するための拡張可能なプログラムコードのテンプレートです
  - Interface - データを含まず、メソッドシグネチャとして振る舞いを定義する抽象型です
  - Method - オブジェクトに関連付けられ、暗黙的にそのオブジェクトに対して作用する手続きです

    - This keyword - 現在の関数呼び出しやメソッド呼び出しに関連付けられたオブジェクトを参照するために、多くのオブジェクト指向プログラミング言語で使用されるキーワードです
    - Prototype-based programming - プロトタイプとして機能する既存のオブジェクトを再利用する過程を通じて、振る舞いの再利用を行うオブジェクト指向プログラミングのスタイルです

関数型プログラミング

- Functional Programming - 関数の適用と合成によってプログラムを構築するプログラミングパラダイムです
    - Algebraic data type - 他の型を組み合わせて形成される複合データ型です
    - Pattern matching - あるパターンの構成要素が存在するかどうかを、与えられたトークン列に対して確認する行為です
    - First-class function - 関数を第一級オブジェクトとして扱う (例えば変数に代入したり引数として渡したりできる) プログラミング言語の性質です
        - Map - 与えられた関数を並びの各要素に適用し、その結果を含む並びを返す高階関数です
        - Filter - データ構造を処理し、与えられた述語が真を返す要素のみを含む新しいデータ構造を生成する高階関数です
        - Reduce - 結合演算を再帰的に適用することでデータ構造を単一の値に縮約する高階関数 (fold とも呼ばれます) です
    - Referential transparency - 式をプログラムの振る舞いを変えることなくその対応する値に置き換えられるという、式の性質です
    - Closure - 関数と、その関数の非局所変数のための参照環境とを組み合わせたものです
    - Side-effect - 操作、関数、式が、そのローカル環境の外部にある状態変数の値を変更する、観測可能な影響です
    - Monad - プログラム断片 (関数) を組み合わせ、その戻り値を追加の計算を伴う型でラップする構造を持つ、ソフトウェア設計パターンです
    - Currying - 複数の引数を取る関数を、それぞれが単一の引数を取る関数の並びに変換する技法です

リアクティブプログラミングとその他

- Reactive Programming - データストリームと変化の伝播を扱う宣言型プログラミングパラダイムです

    - Functional Reactive Programming (FRP) - 関数型プログラミングの構成要素を用いたリアクティブプログラミングのためのプログラミングパラダイムです
    - 言語とフレームワーク
        - RO - Go 向けのストリームとリアクティブプログラミングのためのライブラリです
        - ReactiveX - 観測可能なストリームを用いた非同期プログラミングのための API です
        - Elm - 信頼性の高い Web アプリケーションのための心地よい言語です
- Aspect-oriented Programming - 横断的関心事の分離を可能にすることでモジュール性を高めることを目指すプログラミングパラダイムです
    - Cross-cutting concern - プログラムのある側面が、そのいずれにもカプセル化されることなく複数のモジュールに影響を与えることです

スクリプト言語

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

Python

- Python - 素早く作業を行いシステムをより効果的に統合できるようにするプログラミング言語です
    - コア機能
        - Python import system - Python コードをモジュールやパッケージに整理し、コードの再利用と大規模アプリケーションの構造化を容易にする仕組みです
        - Special method names - 先頭と末尾に二重アンダースコアが付くことで識別される、特殊な構文によって呼び出される操作をクラスが実装できるようにするメソッドです
        - Type Hints - 変数、関数パラメータ、戻り値の型注釈のための標準的な構文であり、静的解析に使用されます
            - typing module - 型ヒントに対する実行時サポートを提供する標準ライブラリモジュールです
            - Mypy - 動的型付けと静的型付けの利点を組み合わせることを目指す、Python 用のオプションの静的型チェッカーです
        - f-string - 'f' または 'F' を接頭辞に持つ文字列リテラルの一種で、最小限の構文で文字列定数の中に式を埋め込むことができます

        - with statement - リソース管理のための try/finally 文の標準的な使用をカプセル化することで、例外処理を簡略化する文です
            - contextlib - with 文を伴う一般的なタスクのためのユーティリティを提供するモジュールです
        - Generators - yield 文を持つ関数を用いて定義される、イテレータを作成するためのシンプルで強力な方法です
        - Decorators - 関数やメソッドを変換するための '@' 記号を用いた構文で、多くの場合それらを非侵襲的に変更・強化するために使用されます
        - Coroutine - async def で定義される特殊なジェネレータ関数であり、実行を中断・再開でき協調的マルチタスクを可能にします
        - Lambda - lambda キーワードを用いて定義される、単一の式に限定された小さな無名関数です
        - Data Classes - 主にデータを保存するために使用されるクラスを簡潔に作成する方法を提供する、モジュールおよびデコレータであり、特殊メソッドを自動的に生成します
        - Pattern Matching - シーケンス、マッピング、オブジェクト構造を含む複雑なパターンに対する値のマッチングを可能にする、switch 文に類似した機能を提供する機能です
        - Unpacking Operator - によるイテラブルのアンパック演算子と* による辞書のアンパック演算子の拡張された使用法であり、より多くの位置で、任意の回数、かつ追加の状況においてアンパックを可能にします
    - 主要ライブラリ
        - pathlib - 異なるオペレーティングシステムに適したセマンティクスを持つファイルシステムパスを表現するクラスを提供するモジュールです
        - dotenv - .env ファイルからキーと値のペアを読み込み、それらを環境変数として設定できるライブラリです
        - Pydantic - Python のためのデータ検証・設定管理ライブラリです
        - Tenacity - Python 向けの汎用リトライライブラリです
    - 開発ツール
        - IPython - 履歴機能、タブ補完、特殊なマジックコマンドを備えた、Python に対するリッチな対話型インターフェースです

JavaScript と TypeScript

- JavaScript/ECMAScript - ECMAScript 言語を定義する標準規格です
    - モジュールシステム

        - CommonJS - ブラウザ外での JavaScript のためのエコシステムを規定することを目標とするプロジェクトです
        - ES modules - 再利用のために JavaScript コードをパッケージ化する公式の標準形式です
        - UMD - ブラウザ、および AMD や CommonJS ベースのシステムで使用するための、Universal Module Definition のパターン群です
    - コア機能
        - Event-driven - ユーザーの操作、センサーの出力、他のプログラムからのメッセージといったイベントによってプログラムの流れが決定されるプログラミングパラダイムです
        - Spread and rest operators - 配列式や文字列などのイテラブルを、0 個以上の引数や要素が期待される場所で展開できるようにする構文です
        - Generator - ジェネレータ関数によって返されるオブジェクトで、イテラブルプロトコルとイテレータプロトコルの両方に準拠します
    - 主要ライブラリ
        - Lodash - モジュール性、パフォーマンス、追加機能を提供するモダンな JavaScript ユーティリティライブラリです
        - dax - zx に触発された、Deno と Node.js のためのクロスプラットフォームシェルツールです
        - Bun Shell - シェルスクリプトを実行するための組み込みのシェルライクなインターフェースです
        - zx - より良いスクリプトを書くためのツールです
        - Zod - 静的型推論を備えた TypeScript ファーストのスキーマ検証ライブラリです
        - yup - 実行時の値の解析と検証のためのスキーマビルダーです
    - Typescript - JavaScript を基盤として構築される強く型付けされたプログラミング言語で、あらゆる規模でより優れたツールを提供します
        - Union Types - 複数の型を 1 つに組み合わせる方法です
        - Type Aliases - 任意の型に対する名前です
        - Type Assertions - コンパイラに「信じてほしい、自分が何をしているか分かっている」と伝える方法です
        - Mapped Types - PropertyKeys の共用体を使用して別の型のキーを反復処理し、新しい型を作成する汎用型です
        - Nominal typing techniques - デフォルトでは構造的型システムを持つ TypeScript において、公称型をシミュレートする方法です

        - Declaration Files - ライブラリの型を定義するファイルです
        - Decorators - クラス宣言、メソッド、アクセサ、プロパティ、パラメータに付与できる特殊な種類の宣言です
    - TS 型ユーティリティ
        - json-schema-to-typescript - JSONSchema を TypeScript の型宣言にコンパイルするツールです
        - Json Schema to TS - FromSchema メソッドにより JSON スキーマから TS の型を直接推論できます
- チュートリアルと実践
    - 33 JS Concepts - すべての JavaScript 開発者が知っておくべき 33 の概念に関する記事をまとめたリポジトリです
    - JS Project Guidelines - JavaScript プロジェクトのためのベストプラクティス集です
    - Callback Hell - 非同期処理を扱う際にコールバック関数がネストしていくことです
    - NodeSchool - Web ソフトウェアのスキルを教える一連のオープンソースワークショップです
    - Node.js Best Practices - Node.js のベストプラクティスに関する上位ランクのコンテンツをまとめたものです

Ruby、Perl とその他

- Ruby - シンプルさと生産性に重点を置いた動的なオープンソースプログラミング言語です
    - コア機能
        - Percent notation - パーセント記号と区切り文字を使用して、文字列、配列、正規表現などの様々なリテラル型を生成するための簡潔な構文です
        - Fiber - 実行の一時停止と再開によって協調的マルチタスクを可能にする、軽量な並行処理プリミティブです
        - proc - コードのブロックを、保存、渡し、実行できるオブジェクトにカプセル化する仕組みです
        - lambda - 厳密な引数チェックとローカライズされた return の振る舞いを強制する、特殊なブロックオブジェクトです
        - then - オブジェクト自身をブロックに渡してその結果を返すメソッドで、関数型スタイルのメソッドチェーンを容易にします
        - define_method - define_method を使用して実行時にメソッドを作成・登録

する機能で、コードの柔軟性を高め重複を減らします
        - instance_eval - 特定のオブジェクトインスタンスのコンテキスト内でブロックや文字列を評価し、その内部スコープとプライベートメソッドへのアクセスを可能にするメソッドです
    - ライブラリ
        - io-event - イベントループを構築するための低レベルなクロスプラットフォームプリミティブです
        - Async - io-event に基づく、Ruby 用の組み合わせ可能な非同期 I/O フレームワークです
        - ruby-git - git バイナリへのシステムコールをラップすることで Git リポジトリを作成、読み取り、操作できる Ruby ライブラリです
    - 開発ツール
        - IRB (Interactive Ruby) - 標準入力から読み込んだ Ruby の式を対話的に実行するツールです
- Perl - 2 つの高水準で汎用的、インタプリタ型かつ動的なプログラミング言語のファミリーです
    - コア機能
        - Special variables - Perl にとって特別な意味を持つ変数です
        - Built-in regex - Perl における正規表現の構文です
        - Context - 式が評価される際にどのように振る舞うかを決定する式の性質です
        - Scalar values - 単一のデータ項目です
            - Reference - 他のデータを「指し示す」スカラーデータ型です
        - Quote-like operators - 一連の汎用的な引用演算子です
        - I/O operators - ファイルハンドルからの読み取りなど、入出力操作に使用される演算子です
- Tcl - 幅広い用途に使用される動的プログラミング言語とグラフィカルユーザーインターフェースツールキットです
    - Event-driven by design - GUI とネットワーキングに最適な組み込みのイベントループです
- Lua - 強力で効率的、軽量かつ組み込み可能なスクリプト言語です
- Emacs Lisp - Emacs テキストエディタを拡張・カスタマイズするために使用されるプログラミング言語です
    - S-expression - 入れ子になったリスト (木構造) データのための記法です

    - Homoiconicity - 一部のプログラミング言語が持つ性質で、プログラムの主要な表現がその言語自体のプリミティブな型のデータ構造でもあるというものです

非同期処理と並行性

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- Concurrent computing - 複数の計算を逐次的にではなく並行して実行する計算の形態です
    - Coroutine - 実行を中断・再開できるようにすることで、非プリエンプティブなマルチタスクのためにサブルーチンを一般化したコンピュータプログラムの構成要素です
    - Async/await - 非同期でノンブロッキングな関数を、通常の同期的な関数と似た方法で構造化できるようにする構文的機能です
    - Futures and promises - プログラムの実行を同期させるために使用される構造体であり、最初は未知である結果のプロキシを表現します
    - Semaphore - 並行システムにおいて複数のスレッドによる共通リソースへのアクセスを制御するために使用される変数または抽象データ型です
    - Mutex - 複数の実行スレッドによって状態が同時に変更またはアクセスされることを防ぐ、同期プリミティブです
    - Channel - メッセージパッシングによるプロセス間通信と同期のためのモデルです
    - Thread safety - マルチスレッド環境に適用可能なコンピュータコードの性質で、共有データ構造の正しい操作を保証するものです
    - Deadlock - あるグループのエンティティの各メンバーが、他のメンバーが行動を起こすのを待っているために誰も先に進めなくなる、並行計算における状況です

言語解析

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- 概念
    - Formal language - 単語、すなわち文字、記号、トークンの有限列の集合です
        - Well-formed formula - 形式言語の一部である、与えられたアルファベットからの記号の有限列です

    - Formal grammar - 形式言語における文字列の生成規則の集合です
    - Chomsky hierarchy - 形式文法のクラスの包含階層です
    - Rewriting - 式の部分項を他の項で置き換える手法の幅広い範囲です
        - Confluence - 書き換えシステムの性質であり、そのシステム内のどの項が複数の方法で書き換えられて同じ結果に至るかを記述するものです
    - Automata theory - 抽象機械とオートマトン、およびそれらを用いて解決できる計算上の問題を研究する分野です
        - Finite-state machine - ある時点で有限個の状態のうちのちょうど 1 つを取り得て、何らかの入力に応じてある状態から別の状態へ遷移する、計算の数学的モデルです
    - Lexical analysis - テキストを、レキサープログラムによって定義されたカテゴリに属する意味のある字句トークンに変換することです
    - Parsing - 形式文法の規則に従う記号列を部分に分解して解析する処理です
    - BNF syntax - 文脈自由文法のための記法技法であり、計算機で使用される言語の構文を記述するためによく使用されます
    - AST - プログラミング言語で書かれたソースコードの抽象的な構文構造を木で表現したものです
- パーサージェネレーター
    - ANTLR - 構造化されたテキストやバイナリファイルの読み取り、処理、実行、変換のための強力なパーサジェネレータです
    - Bison - 文脈自由文法の文法記述を、その文法を解析する C プログラムに変換する汎用パーサジェネレータです
- レキサージェネレーター
    - Flex - 高速なレキシカルアナライザ (スキャナジェネレータ) です
    - Ragel - 状態機械コンパイラです
- パーサー/ライブラリ
    - tree-sitter - パーサジェネレータツールおよびインクリメンタルパーシングライブラリです
    - ts-morph - TypeScript Compiler API のラッパーです

プログラム翻訳

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- 概念
  - Compiler - あるプログラミング言語で書かれたコンピュータコードを別の言語に翻訳するコンピュータプログラムです
  - Transpiler - あるプログラミング言語で書かれたプログラムのソースコードを入力として受け取り、同じまたは別のプログラミング言語で等価なソースコードを生成する翻訳器の一種です
    - Intermediate representation - コンパイラや仮想マシンがソースコードを表現するために内部的に使用するデータ構造またはコードです
  - Program optimization - ソフトウェアシステムのある側面をより効率的に動作させたり、使用するリソースを削減したりするために変更する過程です
  - Machine code - コンピュータの中央処理装置（CPU）が直接実行できる機械語命令で書かれたコンピュータプログラムです
  - Cross compiler - コンパイラが動作しているプラットフォーム以外のプラットフォーム向けに実行可能コードを生成できるコンパイラです
  - Linker - 1 つ以上のオブジェクトファイルを受け取り、単一の実行可能ファイルに結合するコンピュータシステムプログラムです

主要なコンパイラ基盤

- LLVM Compiler Infrastructure - モジュール化され再利用可能なコンパイラおよびツールチェーン技術の集合です
  - Clang - LLVM のための C 言語ファミリーフロントエンドです
  - LLD - LLVM のリンカです
- gcc - C、C++、Objective-C、Fortran、Ada、Go、D のフロントエンドを含む GNU Compiler Collection です
- rustc - Rust プログラミング言語のコンパイラです

個別の翻訳系とビルドツール

- MinGW-w64 - Windows システム上で GCC コンパイラをサポートするために作られた、元の mingw.org プロジェクトの発展形です
- Go build command - Go のソースコードを管理するためのツールです
  - 静的バイナリ実行ファイル
- GopherJS - Go から JavaScript へのコンパイラです
- Bunster - シェルスクリプトを自己完結型の実行可能プログラムに変換するシェルコンパイラです

リンカ (スタンドアロン)

- mold - モダンなリンカです
- ランタイムライブラリ
  - glibc - GNU システムおよび GNU/Linux システムのコアライブラリを提供する GNU C Library プロジェクトです
  - musl libc - Linux カーネルをベースとしたオペレーティングシステム向けの C 標準ライブラリです

プログラム実行

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- 概念
  - Runtime System - プログラムが書かれた言語のために、コンピュータ上で実行されるプログラムの部分です
  - Bytecode - ソフトウェアインタプリタによる効率的な実行のために設計された命令セットの一形態です
  - Just-in-time compilation - プログラムの実行中にコンパイルを行うことを伴う、コンピュータコードの実行方式です
  - Global interpreter lock - Python オブジェクトへのアクセスを保護し、複数のスレッドが同時に Python バイトコードを実行するのを防ぐミューテックスです

ランタイム実装

- Javascript
  - Node.js - 無料でオープンソースのクロスプラットフォーム JavaScript ランタイム環境です
    - libuv - 非同期 I/O に重点を置いたマルチプラットフォーム対応サポートライブラリです
  - Deno - TypeScript と JavaScript のためのモダンなランタイムです
    - Deno Deploy - JavaScript、TypeScript、WebAssembly をエッジで実行できる分散型 HTTP サービスです
    - Deno Subhosting - SaaS プロバイダが V8 アイソレートを使用して信頼できない顧客のコードを大規模に安全に実行するためのプラットフォームです
  - Bun - JavaScript と TypeScript の実行、ビルド、テスト、デバッグのための高速なオールインワンツールキットです

    - WinterJS - SpiderMonkey エンジンと Tokio ランタイムを使用して Rust 上に構築された、非常に高速な JavaScript ランタイムです
- Python
    - CPython (default)
    - pypy - Python の高速で準拠したオルタナティブ実装です
    - Pyodide - WebAssembly をベースとした、ブラウザと Node.js のための Python ディストリビューションです
- Ruby
    - CRuby (default)
    - JRuby - Java Virtual Machine 上で動作する Ruby プログラミング言語の実装です
- Java SE - 現代的なアプリケーション開発のための、最も実績があり信頼性が高く安全な開発プラットフォームです
    - Java HotSpot VM - Oracle Corporation が提供する、デスクトップとサーバー向けの主要な Java Virtual Machine です
    - JMX API - Java Platform の標準的な一部である Java Management Extensions 技術です
    - JDK tools - アプリケーションを作成しビルドするためのコマンドラインツールです
    - GraalVM - アヘッドオブタイムの Native Image コンパイルを備えた先進的な JDK です
    - OpenJDK - Java Platform, Standard Edition のオープンソース実装を共同開発するための場です
    - Eclipse Temurin - OpenJDK のオープンソースでエンタープライズ対応かつ TCK 認定済みのビルドです
- .NET - モダンなアプリと強力なクラウドサービスを構築するための、無料でオープンソースのクロスプラットフォームフレームワークです
    - CLR - .NET Framework の仮想マシンコンポーネントです
- ランタイムユーティリティ
    - PM2 - アプリケーションをオンラインに保つ管理を助けるデーモンプロセスマネージャです
    - PyCall - Ruby から Python の関数を呼び出せるようにする Ruby ライブラリです
    - VisualVM - オールインワンの Java トラブルシューティングツールです

アルゴリズムと計算複雑性

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- 概念
  - Complexity class - 関連するリソースベースの複雑性を持つ計算問題の集合です
- 参考リソース
  - NIST Dictionary of Algorithms and Data Structures - アルゴリズム、アルゴリズム的技法、データ構造、典型的な問題、関連する定義の辞書です

アルゴリズム

- Algorithm - 特定の種類の問題を解決したり計算を実行したりするために通常使用される、厳密な命令の有限列です
  - 解析技法
    - Amortized analysis - 与えられたアルゴリズムの複雑性を分析するための手法です
    - Big O notation - 引数が特定の値または無限大に近づくときの関数の極限的な振る舞いを記述する数学的記法です
  - アルゴリズムパラダイム
    - Recursion - 解が同じ問題のより小さなインスタンスの解に依存する、計算問題を解決する方法です
    - Divide and conquer - アルゴリズム設計パラダイムです
    - Dynamic programming - 複雑な問題をより単純な部分問題の集合に分解することで解決する方法です
    - Backtracking - いくつかの計算問題に対する解を見つけるための一群のアルゴリズムです
    - Greedy algorithm - 各段階で局所的に最適な選択を行うという問題解決のヒューリスティックに従うアルゴリズムパラダイムです
  - ソートアルゴリズム
    - Quicksort - インプレースソートアルゴリズムです
    - Merge sort - 効率的で汎用的な、比較に基づくソートアルゴリズムです
    - Heapsort - 比較に基づくソートアルゴリズムです
  - 探索アルゴリズム

        - Binary search - ソート済み配列内で目的の値の位置を見つける探索アルゴリズムです
        - Interpolation search - キーに割り当てられた数値によって順序付けられたソート済み配列内でキーを探索するアルゴリズムです
    - 文字列アルゴリズム
        - Knuth–Morris–Pratt algorithm - メインのテキスト文字列 T の中に「単語」W が出現する箇所を探索する文字列検索アルゴリズムです
        - Boyer–Moore algorithm - 実用的な文字列検索の文献における標準的なベンチマークとなっている文字列検索アルゴリズムです
        - Longest common subsequence - ある列の集合内のすべての列に共通する最長の部分列を見つける問題です
    - グラフアルゴリズム
        - 走査
            - Breadth-first search - 木構造やグラフ構造のデータを走査または探索するアルゴリズムです
            - Depth-first search - 木構造やグラフ構造のデータを走査または探索するアルゴリズムです
        - 最短経路
            - Dijkstra's algorithm - 重み付きグラフ内のノード間の最短経路を見つけるアルゴリズムです
            - Bellman–Ford algorithm - 重み付き有向グラフにおいて単一の始点から他のすべての頂点への最短経路を計算するアルゴリズムです
        - Minimum Spanning Tree - 連結で辺に重みが付いた無向グラフのすべての頂点を接続する辺の部分集合です
            - Prim's algorithm - 重み付き無向グラフに対する最小全域木を見つける貪欲アルゴリズムです
            - Kruskal's algorithm - 森の中の任意の 2 つの木を接続する、可能な限り重みの小さい辺を見つける最小全域木アルゴリズムです
        - その他
            - Tarjan's strongly connected components algorithm - 有向グラフの強連結成分 (SCC) を見つけるためのグラフ理論のアルゴリズムです
            - Topological sorting - 有向非巡回グラフ (DAG) の頂点の線形順序付けです
    - ハッシュアルゴリズム

        - Hash function - 任意サイズのデータを固定サイズの値に写像するために使用できる任意の関数です

データ構造

- Abstract Data Types - データ型のための数学的モデルです
    - String - アルファベットと呼ばれる集合から選ばれた記号の有限列です
    - List - 有限個の順序付けられた値を表現する抽象データ型です
    - Associative array - (キー, 値) のペアの集まりを保持できる抽象データ型です
    - Stack - push と pop という 2 つの主要な操作を持つ要素の集合として機能する抽象データ型です
    - Queue - enqueue と dequeue という 2 つの主要な操作を持つ要素の集合として機能する抽象データ型です
        - Priority queue - 通常のキューやスタックのデータ構造に似ていますが、さらに各要素に「優先度」が関連付けられている抽象データ型です
    - Tree - 接続されたノードの集合を持つ階層的な木構造を表現する抽象データ型です
    - Graph - 数学における無向グラフと有向グラフの概念を実装することを意図した抽象データ型です
        - Directed acyclic graph (DAG) - 有向閉路を持たない有向グラフです
- Data Structures - 効率的なアクセスと変更を可能にするよう設計された、データの編成・管理・保存形式です
    - Passive data structure - 公開データフィールドのみを含み、フィールドの読み書き以外のメソッドを暗黙的にしか提供しないレコードデータ構造です
    - Array - 要素 (値または変数) の集まりからなるデータ構造です
        - Array slicing - 配列から要素の部分集合を抽出し、それを別の配列としてパッケージ化する操作です
        - Sparse matrix - ほとんどの要素がゼロである行列であり、メモリと処理時間を節約するための専用の記憶方式とアルゴリズムを可能にします
    - Hash table - 連想配列という抽象データ型を実装するデータ構造です
        - 衝突解決
            - Cuckoo hashing - ハッシュテーブル内のキーのハッシュ衝突を解決するためのコンピュータプログラミングの手法です
            - Linear probing - ハッシュテーブルにおける衝突を解決するためのコンピュータプログラミングの手法です

    - Linked data structure - 一連のデータレコード（ノード）が参照によって連結され組織化されたデータ構造です
    - Persistent structure - 変更されても常に以前のバージョンの自身を保持するデータ構造です
    - Disjoint-set data structure - 互いに素な（重複しない）集合の集まりを格納するデータ構造です
    - ツリーベース
        - Search tree - ある集合の中から特定のキーを探索するために使用される木構造のデータ構造です
            - Binary search tree (BST) - 各内部ノードのキーが、そのノードの左部分木のすべてのキーより大きく、右部分木のキーより小さい、根付き二分木のデータ構造です
        - Merkle tree - すべての葉ノードがデータブロックの暗号学的ハッシュでラベル付けされる木です
        - Heap - ヒープ性質を満たす木構造のデータ構造です
        - Trie - ある集合の中から特定のキーを探索するために使用される探索木のデータ構造です
        - Fenwick tree - 数値の表において要素の更新と接頭辞和の計算を効率的に行えるデータ構造です
    - グラフベース
        - Adjacency matrix - 有限グラフを表現するために使用される正方行列です
        - Adjacency list - 有限グラフを表現するために使用される順序なしリストの集まりです

10 - 高度なプログラミング

手続き型・システムプログラミング

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

Rust 言語

- Rust - 誰もが信頼性が高く効率的なソフトウェアを構築できるようにするプログラミング言語です
    - Ownership and borrowing - Rust プログラムがメモリをどのように管理するかを規定する一連の規則です

    - Interior mutability - そのデータへの不変参照が存在する場合でも、データを変更できるようにする Rust の設計パターンです
    - Closure - 変数に保存したり、他の関数への引数として渡したりできる匿名関数です
    - Trait-based generics - 型が提供しなければならない振る舞いを定義する方法であり、指定された振る舞いを実装する任意の型に対して動作する汎用的なコードを可能にするものです
    - Lifetime - すべての借用が有効であることを保証するためにコンパイラが使用する構成要素です
    - Module Pin - データをメモリ上の位置に固定する型を提供するモジュールです
    - ツール
        - C2Rust - ほとんどの C モジュールを意味的に等価な Rust コードに変換できるツールです
    - チュートリアル
        - Rust by Example - さまざまな Rust の概念と標準ライブラリを説明する、実行可能なサンプルのコレクションです

Go 言語

- Go - Google がサポートするオープンソースのプログラミング言語です
    - コア機能
        - Go Modules - Go プログラミング言語のための依存関係管理システムです
        - Defer, panic and recover - Go における強力ですが特異な制御フロー機構です
        - Pointer receiver - 型へのポインタに対して動作し、レシーバーが指す値を変更できるようにするメソッドです
        - Interface - メソッドシグネチャの集合として定義される型です
        - Goroutine - Go ランタイムによって管理される軽量スレッドです
        - Channel - チャネル演算子 ← を使って値を送受信できる、型付けされた導管です
    - ライブラリ
        - Awesome Go - 優れた Go のフレームワーク、ライブラリ、ソフトウェアを厳選したリストです
        - lo - Lodash スタイルの Go ライブラリです
        - fp-go - 関数型プログラミングのヘルパー集です

        - shortuuid - 簡潔で曖昧さがなく URL セーフな UUID を生成するライブラリです
    - ツール
        - Go binary size SVG treemap - Go 実行ファイルのサイズのツリーマップを作成する CLI ツールです
        - mvm - コンパイルなしでソースコードから直接 Go プログラムを実行できるようにする、Go 用の高速な仮想マシンです
    - チュートリアル
        - Effective Go - 明確でイディオマティックな Go コードを書くためのヒントを提供するドキュメントです
        - Go by Example - 注釈付きのサンプルプログラムを使った Go の実践的な入門です
        - Learn Go with tests - 初日からテストを含む Go の基礎を教えるリソースです

C とその他の手続き型言語

- C - 構造化プログラミング、レキシカルな変数スコープ、再帰をサポートする、静的型システムを備えた汎用手続き型コンピュータプログラミング言語です
    - Macros - 名前が付けられたコードの断片です
- Zig - 堅牢で最適かつ再利用可能なソフトウェアを保守するための汎用プログラミング言語およびツールチェーンです
    - Comptime - コンパイル時にコードを実行できるようにする仕組みです

関数型・ハイブリッドプログラミング

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

関数型優先・マルチパラダイム言語

- F# - 簡潔で堅牢かつ高性能なコードを書くための汎用プログラミング言語です
    - 不変データ構造
    - Discriminated union - いくつかの異なる、しかし固定された型のいずれかの値を格納できる型です
    - Active pattern - 入力データを分割する名前付きのパーティションを定義できる機能であり、これらの名前をパターンマッチング式で使用できるようにするものです

    - Computation expression - 制御フロー構造とバインディングを使って順序付けたり組み合わせたりできる計算を記述するための、便利な構文を提供する機能です
- OCaml - 表現力と安全性を重視した、実用に耐える関数型プログラミング言語です
    - Functor - 別のモジュールによってパラメータ化されたモジュールです
- Haskell - 高度な純粋関数型プログラミング言語です
    - 純粋関数型
    - Lazy evaluation - 式の値が必要になるまで、その評価を遅らせる評価戦略です
- Elixir - スケーラブルで保守しやすいアプリケーションを構築するための動的な関数型言語です
    - Process - 分離されており、メッセージを介して情報をやり取りする軽量な実行スレッドです
- Power Fx - Microsoft Power Platform 全体で使用される、汎用的で強く型付けされた宣言型かつ関数型のローコード言語です

マルチパラダイム・ハイブリッド言語

- Java - 第 1 位のプログラミング言語であり開発プラットフォームです
    - Built-in concurrency support - 並行プログラミングをサポートするために根本から設計された Java プラットフォームの機能です
- C# - モダンでオブジェクト指向かつ型安全なプログラミング言語です
    - Language-Integrated Query (LINQ) - クエリ機能を C# 言語に直接統合することに基づく一連の技術の名称です
    - Delegate - 特定のパラメータリストと戻り値の型を持つメソッドへの参照を表す型です
    - Lambda expression - 匿名関数を作成する方法です
- Scala - 一般的なプログラミングパターンを簡潔でエレガントかつ型安全な方法で表現できるように設計された、モダンなマルチパラダイムプログラミング言語です
    - Hybrid OO/functional - 静的型付けの環境でオブジェクト指向プログラミングと関数型プログラミングを融合させた言語の特性です
- Groovy - 静的型付けと静的コンパイルの機能を備えた、Java プラットフォーム向けの強力でオプション型付け可能な動的言語です
- Dart - あらゆるプラットフォームで高速なアプリを実現する、クライアント最適化言語です

データ・フォーマット標準

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

テキスト形式と文字コード

- ASCII - 電子通信のための文字符号化標準です
- Unicode - 普遍的な文字符号化標準です
  - UTF-8 - 電子通信に使用される可変長文字符号化方式です
  - Unicode Emoji - 顔文字のように使用される、標準化された文字の集合です
- CSV - 値をカンマで区切る区切りテキストファイルです
- TSV - 表内の値をタブ文字で区切る区切りテキストファイル形式です
- ライブラリ
  - ICU - Unicode とグローバリゼーションを提供する、成熟し広く使われている C/C++ と Java のライブラリ群です
  - Python emoji - Python 用の絵文字ライブラリです
  - Go emoji - Go 用のミニマルな絵文字パッケージです

日時形式

- UTC - 時計と時刻を規制するために世界的に使用されている主要な時間標準です
- ISO 8601 - 日付・時刻に関連するデータの世界的な交換と伝達を対象とする国際標準です
- Unix time - ある時点を記述するための方式です
- ライブラリ
  - Ruby Time - 日付と時刻の抽象化です
  - Python delorean - Python で日時を扱う際に生じる不都合な真実を解消するためのライブラリです
  - Python arrow - 日付、時刻、タイムスタンプの作成、操作、フォーマット、変換に対して、理にかなった人間に優しいアプローチを提供する Python ライブラリです
  - Luxon - JavaScript の日付と時刻のための強力でモダン、かつ使いやすいラッパーです
  - date-fns - ブラウザと Node.js で JavaScript の日付を操作するための、最も包括的でありながらシンプルで一貫性のあるツールセットを提供する、モダンな

JavaScript 日付ユーティリティライブラリです
    - Day.js - Moment.js とほぼ互換性のある API を持ち、モダンブラウザ向けに日付と時刻の解析、検証、操作、表示を行う、ミニマルな JavaScript ライブラリです
    - Go time - 時間の計測と表示のための機能を提供するパッケージです
    - Go when - 依存関係のない自然言語の日時パーサーです
- iCalendar - カレンダーとスケジュールの情報を保存・交換できるようにするメディアタイプです

データ交換言語

- JSON - 軽量なデータ交換フォーマットです
    - jq - 軽量で柔軟なコマンドライン JSON プロセッサです
    - gojq - jq の純粋な Go 実装です
    - gron - JSON を個別の代入文に変換し、探したいものを grep しやすくし、その絶対的な「パス」を確認しやすくするツールです
    - JMESPath - JSON のためのクエリ言語です
    - JSON::Tiny - 依存関係のないミニマルな JSON モジュールです
    - Python json - JSON のエンコーダーとデコーダーを実装するモジュールです
    - Jackson - ストリーミング JSON パーサー・ジェネレーターライブラリと、それに対応するデータバインディングライブラリを含む、Java 向けのデータ処理ツール群です
- XML - SGML (ISO 8879) から派生した、シンプルで非常に柔軟なテキストフォーマットです
    - XPath - XQuery および XPath データモデルに準拠した値の処理を可能にする式言語です
    - DOM - イベント、処理の中断、ノードツリーのためのプラットフォーム中立なモデルです
    - Python xml.etree.ElementTree - XML データの解析と生成のためのシンプルで効率的な API を実装するモジュールです
    - Nokogiri - ドキュメントの読み取り、書き込み、変更、クエリのためのツールを提供し、XML と HTML の操作を簡単で苦労なく行えるようにする Ruby ライブラリです
- logfmt - シンプルで高速、かつ人間にもマシンにも解析しやすいログフォーマットです
- JSON Lines - 一度に 1 レコードずつ処理できる構造化データを保存するための便利なフォーマットです

- 関連ツール
  - fx - ターミナル向けの JSON ビューアーです
  - jnv - JSON のナビゲーション用に設計された、対話型の JSON ビューアーおよび jq フィルターエディタです

設定言語

- JSON Superset
  - Jsonnet - アプリとツールの開発者向けのデータテンプレート言語です
  - Hjson - JSON のためのユーザーインターフェースです
  - YAML - すべてのプログラミング言語向けの、人間に優しいデータシリアライゼーション言語です
    - yq (python) - YAML、XML、TOML ドキュメント向けの、コマンドライン YAML、XML、TOML プロセッサ兼 jq ラッパーです
    - yq (go) - ポータブルなコマンドライン YAML、JSON、XML、CSV、TOML、プロパティプロセッサです
    - YAML::Tiny - できる限り少ないコードで書かれた、YAML スタイルのファイルを読み書きするための Perl クラスです
    - PyYAML - Python 用の YAML パーサーおよびエミッターです
  - StrictYAML - YAML 仕様の制限されたサブセットを解析・検証する、型安全な YAML パーサーです
  - JSON with comments - JSONC (コメント付き JSON) を解析・文字列化する JS ライブラリです
  - CUE - 論理プログラミングにルーツを持つ、オープンソースのデータ検証言語兼推論エンジンです
- その他の設定言語
  - TOML - 読みやすいミニマルな設定ファイルフォーマットです
    - TOML::Tiny - ミニマルな、純粋な Perl 製の TOML パーサー兼シリアライザです
    - Python tomllib - TOML を解析するためのインターフェースを提供するモジュールです
  - HCL - 人間にもマシンにも優しい構造化された設定言語を作成するためのツールキットです
- 関連ツール
  - yj - YAML、TOML、JSON、HCL 間の変換を行うコマンドラインインターフェースツールです

- 汎用式言語
  - CEL - 高速でポータブル、かつ安全に実行できるように設計された汎用式言語です

テキスト処理と操作

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

正規表現

- Regex - テキスト内の検索パターンを指定する文字列です
  - PCRE - Perl 5 と同じ構文と意味論を用いて正規表現パターンマッチングを実装するライブラリです
  - Onigmo - Oniguruma からフォークされた正規表現ライブラリです
  - Python re - Perl にあるものと同様の正規表現マッチング操作を提供するモジュールです
  - Go regexp - 正規表現検索を実装するパッケージです
  - RE2 - バックトラック型の正規表現エンジンに代わる、高速で安全かつスレッドフレンドリーな代替エンジンです
- PRegEx - 正規表現をプログラム的に作成できるようにする Python ライブラリです
- 正規表現ツール
  - Rubular - Ruby ベースの正規表現エディタです
  - rubree - Rubular にインスパイアされた Ruby の正規表現エディタです
  - Wubular - Rubular にインスパイアされた、Javascript ベースの正規表現エディタです
  - RegEx101 - オンラインの正規表現エディタ兼デバッガーです

テキストツール

- 汎用ツール
  - GNU sed - 入力ストリームに対して基本的なテキスト変換を行うために使用されるストリームエディタです
  - sd - 直感的な検索・置換コマンドラインツールです
  - GNU diffutils - ファイル間の差分を見つけるための複数のプログラムからなるパッケージです
  - colordiff - diff と同じ出力を、可読性を高める色付きのシンタックスハイライト付

きで生成するツールです

表形式データ

- CLI ツール
  - csvkit - CSV への変換や CSV を扱うためのコマンドラインツール群です
  - xsv - Rust で書かれた高速な CSV コマンドラインツールキットです
  - qsv - CSV ファイルのインデックス作成、スライス、分析、分割、拡充、変換、結合を行うコマンドラインプログラムです
  - GNU awk - ファイル内の特定のレコードを選択し、それらに対して操作を行うために使用できるプログラムです
  - GoAWK - CSV をサポートする、Go で書かれた POSIX 準拠の AWK インタプリタです
- ライブラリ
  - Text::CSV - カンマ区切り値 (CSV) の操作を行うモジュールです (XS または PurePerl を使用)
  - Python csv - CSV 形式の表形式データを読み書きするクラスを実装するモジュールです
  - Ruby csv - CSV ファイルとデータへの完全なインターフェースです
  - smarter_csv - CSV ファイルの読み書きを便利に行うための Ruby Gem です
  - Go csv - カンマ区切り値 (CSV) ファイルを読み書きするパッケージです
  - Papa Parse - JavaScript 向けの強力なブラウザ内 CSV パーサーです
  - Python tabulate - データを視覚的に見やすい形式で表示するライブラリ兼コマンドラインユーティリティです
  - Text::MarkdownTable - MultiMarkdown 構文でフォーマットされた表形式でデータを書き出すために使用できるモジュールです
  - Terminal Table - シンプルで機能豊富な、Ruby 向けの ASCII テーブル生成ライブラリです
  - go-pretty - golang 製のテーブルライターほか多機能を備えたライブラリです

テンプレートエンジン

- テンプレート言語とエンジン
  - gomplate - 多数のデータソースと数百の関数をサポートする高速なテンプレートレンダラーです
  - Go template - テキスト出力を生成するためのデータ駆動型テンプレートを実装す

るパッケージです

- sprig - Go のテンプレート言語向けのテンプレート関数を提供するライブラリです

- mustache - ロジックレスなテンプレート構文です
- Jinja - Python 向けの高機能なテンプレートエンジンです
- Perl Text::Template - 定型文の作成、HTML ページの構築など、あらゆる用途に使えるライブラリです
- Perl HTML::Template - HTML テンプレートを作成するためのシステムです
- Template Toolkit - 高速で柔軟、かつ高度に拡張可能なテンプレート処理システムです
- ERB - 使いやすくパワフルな、Ruby 向けのテンプレートシステムです
- Haml - インラインコードを使用せずに、あらゆる Web ドキュメントの HTML をクリーンかつシンプルに記述するために使用されるマークアップ言語です
- Liquid - 柔軟な Web アプリ向けの、安全で顧客向けのテンプレート言語です
- envsubst in gettext - 環境変数の値を置換するプログラムです

マークアップ・ドキュメント処理

- unified - コンテンツの作成と操作のために構築されたプラグインのエコシステムに支えられた、使いやすいインターフェースです
  - remark - プラグインによって駆動される Markdown プロセッサです
- markdown-it - 100% の CommonMark サポート、拡張機能、シンタックスプラグインを備えた Markdown パーサーです
  - markdown-it-py - markdown-it プロジェクトの Python 移植版です
- markdownify - タグ、フォーマット、スタイリングの扱いをカスタマイズできるオプションを備えた、HTML を Markdown に変換する Python ライブラリです

デバッグ

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- デバッガー
  - Python
    - VSCode Python extension - Python 言語を豊富にサポートする拡張機能です

        - debugpy - Python 3 向けの Debug Adapter Protocol の実装です
    - Node.js
        - VSCode built-in debugger - 編集・コンパイル・デバッグのループを高速化するのに役立つ、組み込みのデバッガーです
        - Node.js built-in inspector - デバッグとプロファイリングのために Chrome DevTools を Node.js インスタンスにアタッチできるインスペクタです
    - Go
        - VSCode Go extension - Go プログラミング言語向けの豊富な言語サポートを提供する拡張機能です
        - Delve - Go プログラミング言語向けのデバッガーです
    - Ruby
        - VSCode rdbg Ruby Debugger - debug.gem をベースにした Ruby デバッガー拡張機能です
        - debug.rb - Ruby 向けのデバッグ機能です
    - その他
        - VSCode Bash Debug - bashdb をベースにした bash デバッガーの GUI フロントエンドです
        - BASH Debugger - bash シェルのコマンドラインデバッガーです
        - GDB - GNU プロジェクトのデバッガーです
- デバッガープロトコル
    - DAP - 開発ツール (IDE やエディタなど) とデバッガーの間で使用される抽象プロトコルです
    - V8 V8 Inspector Protocol - JavaScript アプリケーションのデバッグとプロファイリングのために、ツールが V8 を計測できるようにするプロトコルです

ロギング

- ロギングライブラリ
    - Python
        - Python logging - アプリケーションやライブラリのための柔軟なイベントロギングシステムを実装する関数とクラスを定義するモジュールです
        - loguru - Python で楽しいロギング体験をもたらすことを目指すライブラリです
    - Javascript/Typescript

        - bunyan - node.js サービス向けのシンプルで高速な JSON ロギングライブラリです
        - winston - ほぼあらゆる用途に使えるロガーです
        - debug - Node.js コアのデバッグ手法をモデルにした、小さな JavaScript デバッグユーティリティです
    - Go
        - Charmbracelet log - ミニマルでカラフルな Go ロギングライブラリです
        - Go log - シンプルなロギング機能を実装するパッケージです
        - zap - Go における非常に高速で構造化された、レベル分けされたロギングです
        - Logrus - Go (golang) 向けの構造化ロガーであり、標準ライブラリのロガーと完全に API 互換です
        - Zero Allocation JSON Logger - JSON 出力に特化した、高速でシンプルなロガーを提供するパッケージです
    - その他
        - logger - システムログにメッセージを記録するツールです
        - log4j - 汎用性の高い、実運用レベルの Java ロギングフレームワークです
        - log4sh - シェルスクリプト向けの高度なロギングフレームワークです
        - log4net - 優れた Apache log4j フレームワークを Microsoft .NET ランタイムに移植したものです

テストフレームワークとツール

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > ソフトウェア開発プロセス

- テストの概念とベストプラクティス
    - Test Pyramid - バランスの取れたポートフォリオを作るために、異なる種類のテストをどのように使うべきかについての考え方です
    - Test case - 1 つのテストを定義する、入力、実行条件、テスト手順、期待される結果の仕様です
    - Test double - テストにおいて実際のオブジェクトの代役を務めることができるオブジェクトです
    - Arrange-Act-Assert Pattern - テスト対象のオブジェクトをセットアップし、何らかのミューテーターを通じてそれに作用させ、そのオブジェクトや協調オブジェクト、パラメータについて主張を行うという形でテストを構造化するパターンです

    - DAMP (Descriptive And Meaningful Phrases) - 単体テストを書く際に、DRY よりも重視すべき原則です
    - Unit testing best practices with .NET - 堅牢で保守しやすいテストを書く助けとなる一連のベストプラクティスです
    - JS Testing Best Practices - JavaScript 向けの主要なテストプラクティスをまとめたものです
- テストプロトコル
    - Test Anything Protocol - テストモジュールとテストハーネスの間のシンプルなテキストベースのインターフェースです
        - tappy - Test Anything Protocol (TAP) を扱うための一連のツールです
        - Node-Tap - JavaScript 向けの Test-Anything-Protocol ライブラリです

テストフレームワーク

- Bash
    - Bats-core - Bash 用の自動テストシステムです
    - shUnit2 - Bourne 系のシェルスクリプト向けの単体テストフレームワークです
    - shellspec - dash、bash、ksh、zsh、そしてすべての POSIX シェル向けの高機能な BDD 単体テストフレームワークです
- Ruby
    - Minitest - TDD、BDD、モック、ベンチマークをサポートする、完全なテスト機能一式です
    - RSpec - Ruby プログラミング言語向けのテストツールです
    - aruba - Cucumber-Ruby、RSpec、Minitest を使ってコマンドラインアプリケーションをテストするツールです
- Python
    - Python unittest - 「PyUnit」とも呼ばれる単体テストフレームワークで、JUnit の Python 言語版です
    - pytest - 小さく読みやすいテストを簡単に書けるようにし、複雑な機能テストをサポートするためにスケールできるフレームワークです
    - pytest-mock - 標準の unittest.mock パッケージの薄いラッパーである mocker フィクスチャを提供する pytest プラグインです
- Javascript/Typescript
    - Vitest - Vite が搭載された、非常に高速な単体テストフレームワークです

    - Jest - シンプルさに重点を置いた、心地よい JavaScript テストフレームワークです
    - Mocha - Node.js とブラウザ上で動作する、機能豊富な JavaScript テストフレームワークです
    - ランタイム統合型
        - bun test - Bun に組み込まれた、高速で Jest 互換のテストランナーです
        - deno test - JavaScript と TypeScript のコードをテストするために使用できる、組み込みのテストランナーです
- Go
    - Go testing - Go パッケージの自動テストをサポートするパッケージです
    - Ginkgo - Go 向けの BDD スタイルのテストフレームワークです
- その他
    - JUnit - Java と JVM 向けのプログラマーフレンドリーなテストフレームワークの、第 5 メジャーバージョンです
    - xUnit.net - .NET Framework 向けの、無料でオープンソースかつコミュニティ主導の単体テストツールです

アサーションライブラリ

- Chai - node とブラウザ向けの BDD / TDD アサーションライブラリです
- Gomega - Go 向けのマッチャー兼アサーションライブラリです

コードカバレッジツール

- Go cover - Go プログラムのコードカバレッジ統計を提供するツールです
- Istanbul - もう 1 つの JS コードカバレッジツールです
- cobertura - テストによってアクセスされたコードの割合を計算する、無料の Java ツールです
- LCOV - プログラムのどの部分が実際に実行されたかについての情報を提供する GNU ツール GCOV の拡張です
- kcov - コンパイルされたプログラム向けのコードカバレッジテスターです
- SimpleCov - 強力な設定ライブラリと、テストスイート間でのカバレッジの自動マージ機能を備えた、Ruby 向けのコードカバレッジツールです

テスト支援ツール

- モックライブラリ

    - Jest / Vitest 組み込みのもの
    - mock - HTTP レスポンスをモックするためのコマンドラインツールです
    - unittest.mock - テスト対象システムの一部をモックオブジェクトに置き換えられるようにする、Python のテスト用ライブラリです
    - sinon.js - スタンドアロンでテストフレームワークに依存しない、JavaScript のテストスパイ・スタブ・モックです
    - mockery - Golang のインターフェースのモック実装を作成するプロジェクトです
- テストデータジェネレーター
    - Databricks Labs Data Generator - Spark を使って Databricks 環境内で合成データを生成するための Python ライブラリです
    - generatedata.com - 強力で機能豊富な、ランダムテストデータジェネレーターです
    - gofakeit - go で書かれたランダムデータジェネレーターです
    - Fake-rs - Rust で偽データを生成するためのライブラリです
- マルチ環境テストランナー
    - nox - tox と同様に、複数の Python 環境でのテストを自動化するコマンドラインツールです
    - tox - Python 向けの、コマンドラインで駆動する自動テストツールです

ミューテーションテスト

- Mutation testing - ソースコードの特定の文を変更し、テストスイートがそのエラーを検出できるかどうかを確認する、ソフトウェアテストの一種です
    - cargo-mutants - テストを失敗させることなくバグを挿入できる箇所を見つけることで、プログラムの品質向上を助ける Rust 向けのツールです
    - Stryker Mutator - ミュータントを使ってテスト自体をテストするツールです
    - PIT Mutation Testing - Java と JVM に対してゴールドスタンダードなテストカバレッジを提供する、最先端のシステムです

ビルド自動化

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス
> - 3. テクノロジー > 3.1 ソフトウェア開発 > クラウドインフラ活用

- ビルド自動化ツール
  - GNU Make - プログラムの実行可能ファイルやその他の非ソースファイルの生成を制御するツールです
    - Remake - エラーレポートの改善、トレーシングとプロファイリングの向上、そしてデバッガーを追加した GNU Make の拡張版です
    - makefile-graph - GNU Make の内部データベースを解析してグラフを生成する、Go モジュール兼 CLI アプリケーションです
  - Bazel - 高速でスケーラブル、マルチ言語対応かつ拡張可能なビルドシステムです
  - Gradle - ほぼあらゆる種類のソフトウェアをビルドできるほど柔軟に設計された、オープンソースのビルド自動化ツールです
  - Maven - ソフトウェアプロジェクトの管理と理解のためのツールです
  - Task - GNU Make よりもシンプルで使いやすいことを目指す、タスクランナー兼ビルドツールです
  - CMake - ソフトウェアのビルド、テスト、パッケージ化のために設計された、オープンソースのクロスプラットフォームなツール群です
    - CPack - バイナリインストーラーとソースパッケージ用のジェネレーターを設定するツールです
  - Meson - 非常に高速であり、さらに重要なこととして可能な限りユーザーフレンドリーであることを目指した、オープンソースのビルドシステムです
  - Dune - OCaml、OCaml 関連言語、Coq 向けの構成可能なビルドシステムです
  - Rake - Ruby で実装された Make のようなプログラムです
  - fpm - Debian、Ubuntu、Fedora、CentOS、RHEL、Arch Linux などのパッケージを簡単に作成できるツールです
- チュートリアル
  - Makefile Tutorial by Example - Makefile の基礎を教えるチュートリアルです

モノレポ管理

- モノレポツール
  - Turborepo - JavaScript と TypeScript のモノレポ向けの高性能ビルドシステムです
  - Nx - ファーストクラスのモノレポサポートと強力な統合機能を備えた、スマートで高速かつ拡張可能なビルドシステムです
  - Lerna - JavaScript/TypeScript 向けの元祖モノレポツールです
- リソース

- Monorepo Tools - モノレポ向けのツールとリソースを掲載したウェブサイトです

プログラムドキュメント

- プログラムドキュメントツール
  - apiDoc - ソースコード内の API 記述からドキュメントを作成するツールです
  - JSDoc - JavaScript 向けの API ドキュメントジェネレーターです
  - perldoc - perl のインストールツリーに埋め込まれた .pod 形式のドキュメントを検索するツールです
    - Pod - Perl、Perl プログラム、Perl モジュールのドキュメントを書くために使用される、使いやすいマークアップ言語です
  - pydoc - Python モジュールからドキュメントを自動生成するツールです
    - Docstring - モジュール、関数、クラス、メソッドの定義において最初の文として現れる文字列リテラルです
  - godoc - Go プログラム向けのドキュメントを抽出・生成するツールです
  - rustdoc - Rust プロジェクト向けのドキュメントを生成するツールです
  - RDoc - Ruby プロジェクト向けの HTML およびコマンドラインドキュメントを生成するツールです
  - Javadoc - ソースコード内のドキュメントコメントから HTML 形式の API ドキュメントを生成する、Oracle 製のツールです

パッケージ依存関係管理

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- JavaScript と Web エコシステム
  - npm CLI - 世界最大のソフトウェアレジストリです
    - npm-check-updates - package.json の依存関係を最新バージョンにアップグレードできるコマンドラインツールです
    - npmgraph - npm の依存関係グラフを探索するためのツールです
  - yarn - プロジェクトマネージャーとしても機能するパッケージマネージャーです
  - pNPm - 高速でディスク容量効率の良いパッケージマネージャーです
  - Corepack - Node.js プロジェクトと、開発中に使用されることを意図したパッケージマネージャーとの橋渡しをする、ランタイム依存のない Node.js スクリプトです

    - dpmland - Deno のモジュールと依存関係を管理するための、シンプルでモダン、かつ簡単な方法です
    - Bun package manager - Bun に組み込まれた、高速で npm 互換のパッケージマネージャーです
    - orogene - JavaScript エコシステム向けの次世代パッケージマネージャーです
    - vlt - npm 互換パッケージを安全な npm ミラーおよび完全に解決された依存関係グラフとともに提供する、素早く動くチーム向けの JavaScript パッケージレジストリです
- Python 開発
    - pip - Python 向けのパッケージインストーラーです
    - poetry - Python における依存関係管理とパッケージングのためのツールです
    - pdm - 最新の PEP 標準をサポートする、モダンな Python パッケージ・依存関係マネージャーです
    - uv - Rust で書かれた、非常に高速な Python パッケージ・プロジェクトマネージャーです
- システム言語・ネイティブ言語
    - go mod - Go のソースコードを管理するためのツールです
    - Cargo - Rust のパッケージマネージャーです
    - Conan - C 言語と C++ 言語向けの依存関係・パッケージマネージャーです
    - vcpkg - ライブラリの取得と管理のための、無料の C/C++ パッケージマネージャーです
    - bpkg - 軽量な bash パッケージマネージャーです
- JVM と .NET フレームワーク
    - Maven - ソフトウェアプロジェクトの管理と理解のためのツールです
    - Gradle - ほぼあらゆる種類のソフトウェアをビルドできるほど柔軟に設計された、オープンソースのビルド自動化ツールです
    - NuGet CLI - .NET 向けのパッケージマネージャーです
- その他の言語エコシステム
    - RubyGems CLI - Ruby 公式のパッケージマネージャーです
    - Bundler - Ruby プロジェクトに一貫した環境を提供するツールです
    - LuaRocks CLI - Lua モジュール向けのパッケージマネージャーです
    - cpanminus - CPAN からモジュールを取得・展開・ビルド・インストールするツールです

- stack - Haskell プロジェクトを開発するためのクロスプラットフォームプログラムです
- opam - OCaml コミュニティ向けのソースベースのパッケージマネージャーです
- Pub - Dart と Flutter 公式のパッケージマネージャーです
- Hex - Erlang エコシステム向けのパッケージマネージャーです

仮想環境

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス

- 仮想環境マネージャー
  - Python venv - 仮想環境の作成のためのモジュールです
  - pyenv - シンプルな Python バージョン管理のためのツールです
  - nodeenv - 分離された node.js 環境を作成するツールです
  - nvm - 複数のアクティブな node.js バージョンを管理するための、POSIX 準拠の bash スクリプトです
  - nvm-windows - Windows 向けの node.js バージョンマネージャーです
  - rv - Rust で書かれた、シンプルで強力な Ruby バージョンマネージャーです
  - frum - Rust で書かれた、高速でモダンな Ruby バージョンマネージャーです
  - g - シンプルな Go バージョンマネージャーで、グルテンフリーです
  - perlbrew - $HOME ディレクトリ内で複数の perl インストールを管理するツールです
  - asdf - ツールバージョンマネージャーです
  - mise - 多言語対応のツールバージョンマネージャーです
  - tenv - OpenTofu、Terraform、Terragrunt、Atmos 向けの多用途バージョンマネージャーです

11 - 専門開発領域

ビジネス・生産性アプリケーション SDK

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > サービス活用

プロジェクト・業務管理

- Python Jira - A Pythonic interface to the JIRA REST APIs の Python 版インターフェースです
- Notion SDK for JavaScript - Notion API 用の公式クライアントです
- Linear SDK - プロダクトの計画・構築のための Linear の GraphQL API と連携する、型付き TypeScript SDK です
- Asana Python SDK - Asana のプロジェクト管理プラットフォームへプログラムからアクセスできる、Asana API 用の公式 Python クライアントライブラリです
- py-trello - Trello のオブジェクト（ボード、リスト、カード）をキャッシュ属性付きの対応する Python オブジェクトとして表現する Python API ラッパーです

コラボレーション・コミュニケーション

- Python Slack SDK - Python 開発者が Slack と連携するアプリを構築するのを支援するソフトウェア開発キットです
- Slack API in Go - Slack API 用の Go ライブラリです
- discord.js - Discord API と非常に簡単にやり取りできる強力な Node.js モジュールです
- Microsoft Teams JavaScript client library - アプリのコンテンツを iFrame 内でホストしながら、Teams、Microsoft 365 アプリ、Outlook 上でホスト型の体験を作成できるライブラリです
- Notify - さまざまなメッセージングサービスへ通知を送る、非常にシンプルな Go ライブラリです
- Twilio Node.js SDK - 開発者が Twilio の通信サービスを統合するためのツールを提供する、Twilio API 用の公式 Node.js ヘルパーライブラリです
- Zoom Meeting SDK - Angular、React、Vue.js に対応し、Zoom のミーティング・ウェビナー体験を Web アプリケーションへ組み込む WebAssembly ベースの SDK です

クラウドストレージ・ファイル管理

- Dropbox JavaScript SDK - OAuth 対応と TypeScript 互換性を備え、Node.js とブラウザアプリケーション向けに API V2 アクセスを提供する、Dropbox 用の公式 SDK です
- Box Node SDK - 認証方式や自動リトライなどの機能を含め、Box の API エコシステムを完全にカバーする、Box API と連携するための JavaScript インターフェースです

- Airtable.js - RESTful API を通じて Airtable のデータへ簡単にアクセス・連携できる公式 JavaScript ライブラリです

メール・マーケティング自動化

- SendGrid Node.js SDK - SendGrid の Web API v3 を素早く簡単に利用できる、Twilio SendGrid の公式 Node.js API ライブラリです
- Mailchimp Marketing Node.js SDK - Basic Auth または OAuth2 による認証をサポートする、Mailchimp Marketing API 用の公式 Node.js クライアントライブラリです

決済・ファイナンス

- Stripe Node.js SDK - TypeScript 対応と自動リトライを備え、サーバーサイドの JavaScript アプリケーションから Stripe API へ便利にアクセスできるライブラリです

CRM・カスタマーサポート

- JSforce - JavaScript アプリケーション向けの Salesforce API ライブラリです
- HubSpot Node.js SDK - CRM オブジェクト、ファイル、OAuth などのプラットフォーム機能を管理するための HubSpot の V3 API へアクセスできる公式 SDK です
- node-zendesk - Zendesk のカスタマーサポートプラットフォームとシームレスに連携できる、Node.js およびブラウザ向けの信頼性の高い API クライアントライブラリです

エンタープライズワークスペース

- Microsoft 365 開発
  - PnPjs - SharePoint、Graph、Office 365 の REST API を利用するための流暢なライブラリ群です
  - SharePoint Framework (SPFx) - クライアントサイドの SharePoint 開発を完全にサポートし、SharePoint データとの簡単な連携や Microsoft Teams・Microsoft Viva の拡張を可能にする、ページ・Web パーツモデルです
  - CLI for Microsoft 365 - あらゆるプラットフォームで Microsoft 365 テナントと SharePoint Framework プロジェクトを管理できる、クロスプラットフォームのコマンドラインインターフェースです
  - Microsoft Graph - Microsoft 365 のデータとインテリジェンスへのゲートウェイです
  - Work-IQ - Microsoft 365 のデータへアクセスするための MCP (Model Context Protocol) サーバーおよび CLI です

- Google Workspace 開発
  - Google Workspace CLI - Google Discovery Service から動的に構築され、AI エージェントスキルも含む、Drive、Gmail、Calendar、Sheets、Docs、Chat、Admin などを対象とするコマンドラインツールです
  - Google APIs Node.js Client - OAuth 2.0、API キー、JWT トークン認証をサポートし、Google API を利用するための Node.js クライアントライブラリです

開発ツール連携 SDK

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

バージョン管理・DevOps

- Octokit.js - ブラウザ、Node.js、Deno 向けの、あらゆる機能を含む GitHub SDK です
- PyGithub - GitHub REST API を通じてリポジトリ、ユーザープロファイル、組織などの GitHub リソースを管理できる Python ライブラリです
- go-github - GitHub の各種エンドポイントを包括的にカバーする、GitHub v3 REST API へアクセスするための Go クライアントライブラリです
- python-gitlab - Python で書かれた GitLab API のラッパーです
- GitLab Go Client - GitLab のリソースを包括的にサポートする、GitLab API と連携するための公式 Go ライブラリです
- GitLab Ruby Gem - 開発者が GitLab とプログラムで連携できるようにする、GitLab REST API 用の Ruby ラッパーおよび CLI です
- Atlassian Python API - Jira、Confluence、Bitbucket を含む Atlassian 製品の REST API と連携するためのシンプルなラッパーを提供する Python ライブラリです

コンテナ・オーケストレーション

- Docker SDK for Python - Python アプリケーション内から docker コマンドと同等の操作を行える、Docker Engine API 用の Python ライブラリです
- Docker SDK for Go - Docker Engine API 用の公式 Go クライアントです
- Kubernetes client-go - clientset、動的クライアント、コントローラー構築ツールを提供する、Kubernetes クラスターと連携するための公式 Go クライアントライブラリです

- Kubernetes Python Client - プログラムによるクラスター管理とリソース操作を可能にする、Kubernetes API と連携するための公式 Python ライブラリです

CI/CD・自動化

- python-jenkins - Jenkins サーバーを Python らしい方法で制御・自動化できる、Jenkins REST API 用の Python ラッパーです
- GitHub Actions Toolkit - 入出力機能やファイル操作機能などを含む、GitHub Actions の作成を容易にする関数・ユーティリティ群のパッケージです
- Terraform Plugin SDK - サービスプロバイダーや自社独自のソリューションを管理する Terraform プラグイン (プロバイダー) を構築できるフレームワークです

コンピュータグラフィックス・ゲーム開発

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > フロントエンドシステム開発

3D グラフィックス

- Three.js - WebGL を用いて Web ブラウザ上でアニメーション付きの 3D コンピュータグラフィックスを簡単に作成・表示できる JavaScript 製 3D ライブラリです
- GSAP - 開発者をアニメーションのスーパーヒーローに変える堅牢な JavaScript ツールセットです

2D グラフィックス

- PixiJS - Web 上で素晴らしいビジュアル体験を作り出すために設計された、先進的なオープンソースの 2D レンダリングエンジンです

グラフィックス API

- WebGL - OpenGL ES をベースとした低レベルの 3D グラフィックス API のための、クロスプラットフォームでロイヤリティフリーな Web 標準です
- OpenGL - 2D および 3D のベクターグラフィックスをレンダリングするための、言語・プラットフォームを問わないアプリケーションプログラミングインターフェースです
- Vulkan - 低オーバーヘッドでクロスプラットフォームな 3D グラフィックス・コンピューティング API です
  - nvk - Go 言語向けの Vulkan ヘッダーです

バイナリ・メディア処理

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > バックエンドシステム開発

バイナリフォーマットツール

- file (command) - ファイル種別を推測するツールです
- hexdump - 指定したファイルまたは標準入力を、ユーザー指定の形式で表示するフィルターです
- xxd - よく知られた hex-dump 系のユーティリティです
- bed - Go で書かれたバイナリエディタです
- fq - jq に着想を得た、バイナリ形式を調べられるツールです
- ELF format - 実行可能ファイル、オブジェクトコード、共有ライブラリ、コアダンプ向けの共通標準ファイル形式です
- BinData - Ruby で構造化バイナリデータを読み書きするための宣言的な方法です

データシリアライゼーション

- Protobuf - 構造化データをシリアライズするための、言語・プラットフォームを問わない拡張可能な仕組みです
- MessagePack - 効率的なバイナリシリアライゼーションフォーマットです

画像・メディアフォーマット

- JPEG - デジタル画像の非可逆圧縮によく使われる方式です
- PNG - 可逆データ圧縮をサポートするラスターグラフィックスファイル形式です
- Webp - JPEG、PNG、GIF の後継として Google が開発したラスターグラフィックスファイル形式です
- MPEG-4 - 音声・映像のデジタルデータの圧縮を定義する方式です
- High Efficiency Video Coding - 広く使われている Advanced Video Coding (AVC) の後継として設計された動画圧縮規格です

画像・メディア処理

- ツール
  - Swatchify - k-means クラスタリングを用いて画像から主要な色を抽出するツールです

    - exiftool - ファイル内のメタ情報を読み書きするためのコマンドラインアプリケーションおよび Perl ライブラリです
    - ImageMagick - デジタル画像の編集・加工に使われる、無料のオープンソースソフトウェアスイートです
    - FFmpeg - 音声・動画の記録、変換、ストリーミングを行う完全なクロスプラットフォームソリューションです
- ライブラリ
    - go-mp4 - MP4 用の低レベル I/O インターフェースを提供する Go ライブラリです
    - Native WebP for Go - libwebp などの外部ライブラリに依存しない、Go で完全にネイティブに書かれた WebP エンコーダーです
    - Pillow - Python インタープリタに画像処理機能を追加する、親しみやすい PIL (Python Imaging Library) のフォークです
    - RMagick - Ruby から ImageMagick 画像操作ライブラリへのバインディングです
    - pure_jpeg - ネイティブ依存のない純粋な Ruby 製 JPEG エンコーダー・デコーダーライブラリです

圧縮・アーカイブ

- ツール
    - GNU Gzip - 人気のあるデータ圧縮プログラムです
    - GNU tar - tar アーカイブの作成や、その他さまざまな操作を行う機能を提供するプログラムです
    - Info-Zip - ZIP アーカイブを扱うためのオープンソースソフトウェア一式です
    - P7ZIP - POSIX システム向けの 7za.exe の移植版です
- ライブラリ
    - Python Data Compression and Archiving libs - データ圧縮とアーカイブの作成・読み込みをサポートするモジュール群です
    - Go compress libs - 圧縮・解凍アルゴリズムに共通のインターフェースを定義するパッケージです
    - Go archive libs - アーカイブファイル形式へのアクセスに共通のインターフェースを定義するパッケージです
    - JSZip - .zip ファイルの作成、読み込み、編集を行う JavaScript ライブラリです
    - Ruby module Zlib - ストリームの圧縮・解凍や gzip 形式のファイルを扱うクラスを含むモジュールです

- zlib - あらゆるコンピュータハードウェア・OS で利用できる、無料で汎用的、かつ法的制約のない可逆データ圧縮ライブラリです
- zlib-rs - より安全な zlib です
- snappy - 非常に高速で妥当な圧縮率を目指す圧縮・解凍ライブラリです
- rubyzip - zip ファイルの読み書きを行う Ruby ライブラリです
- klauspost/compress - zstandard、S2、最適化された deflate (gzip、zip、zlib)、snappy、エントロピー符号化など、さまざまな圧縮アルゴリズムを提供する Go 用パッケージです

ドキュメント処理

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > バックエンドシステム開発

ビジネス文書フォーマット

- PDF - アプリケーションソフトウェア、ハードウェア、OS に依存しない形で文書を提示するために Adobe が開発したファイル形式です
- PDF/A - 電子文書のアーカイブ・長期保存用途に特化した、Portable Document Format (PDF) の ISO 標準化版です
- Office Open XML - スプレッドシート、チャート、プレゼンテーション、ワードプロセッサ文書を表現するための、zip ベースの XML ファイル形式です
- OpenDocument - スプレッドシート、チャート、プレゼンテーション、ワードプロセッサ文書向けの、zip 圧縮された XML ベースのファイル形式です

PDF 処理

- ツール
  - Docling - 多様なフォーマットを解析し、文書処理を簡素化する強力なライブラリです
  - Ghostscript - Adobe Systems の PostScript および Portable Document Format ページ記述言語向けのインタープリタを基盤とするソフトウェアスイートです
  - qpdf - PDF ファイルに対して内容を保ったまま変換を行う、コマンドラインツールおよび C++ ライブラリです
  - pdftk server - PDF を扱うためのコマンドラインツールです
  - pdfcpu - Go で書かれた PDF プロセッサです

    - MinerU - PDF を Markdown や JSON へ変換する高品質なツールです
    - Nano-PDF - AI ビジョンモデルを利用し、マルチページの並行編集と OCR による非破壊的なテキストレイヤー保持を備えた、自然言語の指示で PDF スライドを編集できるコマンドラインツールです
- ライブラリ
    - Folio - レイアウトエンジン、HTML から PDF への変換、フォーム、電子署名、バーコード、PDF/A 準拠のサポートを含む、Go 向けのモダンな PDF ライブラリです
    - Poppler - xpdf-3.0 のコードベースを基にした PDF レンダリングライブラリです
    - PDF.js - PDF の解析・レンダリングを行う、Web 標準ベースの汎用プラットフォームです
    - pypdf - PDF ファイルのページ分割、結合、切り抜き、変換ができる純粋な Python 製 PDF ライブラリです
    - PyMuPDF - PDF（およびその他のドキュメント）のデータ抽出、解析、変換、操作を行う高性能な Python ライブラリです
    - pdfplumber - 各文字、矩形、線に関する詳細情報を PDF から取り出せる Python ライブラリで、表の抽出や視覚的なデバッグ機能も備えています
    - Prawn PDF - Ruby 向けの高速で軽快な PDF ジェネレーターです
    - ReportLab - PDF やグラフィックスを生成するためのオープンソース Python ライブラリです

オフィス文書処理

- ツール
    - libreoffice cli - LibreOffice オフィススイート用のコマンドラインインターフェースです
    - markitdown - LLM や関連するテキスト分析パイプラインで使用するために、さまざまなファイルを Markdown へ変換する軽量な Python ユーティリティです
    - xlsx2csv - XLSX ファイルを CSV へ高速かつ簡単に変換する方法です
    - docx2txt - docx ファイルからテキストを抽出する、純粋な Python 製のコマンドラインツールです
    - pptx2md - pptx を markdown へ変換するシンプルなツールです
- ライブラリ
    - python-pptx - PowerPoint（.pptx）ファイルを作成・更新するための Python ライブラリです

- PptxGenJS - Node.js、React、Web ブラウザで動作する、PowerPoint プレゼンテーションを構築するための JavaScript ライブラリです
- Excelize - XLSX/XLSM/XLTM ファイルを読み書きする Go ライブラリです
- Roo - 各種スプレッドシートファイルの内容にアクセスできるライブラリです

CLI/TUI 開発

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > コンピュータサイエンス
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発

- Bash
  - built-in getopts etc. - シェル自体に組み込まれたコマンド群です
  - tput - ターミナルを初期化したり terminfo データベースを問い合わせたりするコマンドです
  - dialog - シェルスクリプト用の見栄えの良いユーザーインターフェースを作成できるプログラムです
  - Gum - 華やかなシェルスクリプトを作るためのツールです
  - FIGlet - 通常のテキストから大きな文字を作るプログラムです
  - lolcat - ファイルまたは標準入力を標準出力へ連結し、虹色の着色を加えるプログラムです
  - cfonts - コンソールにセクシーなフォントを出力するツールです
- Perl
  - Getopt::Long - GetOptions() と呼ばれる拡張 getopt 関数を実装したモジュールです
  - Term::ANSIColor - ANSI エスケープシーケンスを使ってテキストを色付けするモジュールです
  - Text::ANSITable - ASCII 文字と ANSI カラーを使ってフォーマット済みの表を作成するモジュールです
- Python
  - argparse - コマンドライン引数を解析するためのモジュールです
  - getopt - コマンドラインオプション用の C スタイルパーサーです
  - click - 必要最小限のコードで組み合わせ可能な、美しいコマンドラインインターフェースを作成するための Python パッケージです

    - Colorama - Python から色付きのターミナルテキストを出力するためのシンプルなクロスプラットフォーム API です
    - Typer - ユーザーに愛され、開発者も作るのが楽しくなる CLI アプリケーションを構築するためのライブラリです
    - Asciimatics - テキストベースのユーザーインターフェースを構築するための、クロスプラットフォームなフルスクリーンターミナル API を提供するパッケージです
    - Python Prompt Toolkit - Python で強力なインタラクティブコマンドラインおよびターミナルアプリケーションを構築するためのライブラリです
    - Questionary - インタラクティブなコマンドラインプロンプトを構築するための Python ライブラリです
    - Urwid - Python 向けのコンソールユーザーインターフェースライブラリです
    - Textual - Textualize.io によって作られた、Python 向けの Rapid Application Development フレームワークです
        - Rich - ターミナルでのリッチテキストと美しいフォーマットのための Python ライブラリです
- Ruby
    - OptionParser - コマンドラインオプション解析用のクラスです
    - colorize - ANSI エスケープシーケンスを使ってテキストを色付けする gem です
    - TTY - インタラクティブなコマンドラインアプリケーションを構築するための幅広いツール群を提供する gem 一式です
    - thor - 強力なコマンドラインインターフェースを構築するためのツールキットです
    - dry-cli - コマンドをオブジェクトとして表現し、引数、オプション、サブコマンドへの可変長引数の転送をサポートする、Command Line Interface (CLI) アプリケーション開発のための汎用フレームワークです
    - Clamp - コマンドライン引数の解析とヘルプ生成を行う、コマンドラインユーティリティ向けの最小限のフレームワークです
    - TableTennis - 自動テーマ設定、レイアウト調整、数値カラムの色分けを備えた、ターミナルに洗練された表を出力するライブラリです
- Javascript
    - yargs - 引数を解析し、洗練されたユーザーインターフェースを生成することでインタラクティブなコマンドラインツールを構築するライブラリです
    - minimist - 引数オプションを解析するツールです

  - chalk - ターミナル文字列のスタイリングツールです
  - cli-progress - コマンドライン・ターミナルアプリケーション向けの使いやすいプログレスバーです
  - FIGLet.js - FIGfont 仕様を完全に実装することを目指した、JavaScript で書かれた FIG ドライバーです
  - Ink - CLI アプリケーションを構築するための React ベースのライブラリです
  - gradient-string - ターミナル出力に美しいグラデーションを作成するためのライブラリです
- Go
  - Fang - CLI スターターキットです。バッテリー同梱の Cobra アプリケーション向けの小さな実験的ライブラリです
  - Bubble Tea - The Elm Architecture に基づいてターミナルアプリを構築するための Go フレームワークです
    - Lip Gloss - ターミナルアプリケーションのスタイルとレイアウトを宣言的に定義するライブラリです
    - Bubbles - 一般的なターミナルユーザーインターフェースコンポーネント集です
  - Huh - ターミナルフォームやプロンプトを構築するための、シンプルで強力かつエレガントな TUI ライブラリです
  - pflag - POSIX/GNU スタイルの --flags を実装した、Go の flag パッケージの代替です
  - color - ANSI エスケープシーケンスによる色付き出力を Go で利用できるパッケージです
  - Cobra - 強力なモダン CLI アプリケーションを作成するためのフレームワークです
  - urfave/cli - Go でコマンドラインアプリを構築するためのシンプルで高速、楽しいパッケージです
  - viper - Go アプリケーション向けの完全な設定ソリューションです
  - Wish - あなたのプログラム向けの小さな SSH サーバーです
  - Wishlist - 秘密の鍵とお気に入りの SSH コマンドをまとめる SSH ディレクトリです
  - go-tui - Go で宣言的なターミナルユーザーインターフェース (TUI) を構築するためのフレームワークです
  - asciigraph - コマンドラインアプリで軽量な ASCII 折れ線グラフを作成する、他に依存関係のない Go パッケージです

- Rust
  - clap - Rust 向けの、あらゆる機能を備えた高速なコマンドライン引数パーサーです
  - Ratatui - おいしいターミナルユーザーインターフェースを作り上げるための Rust ライブラリです
  - R3BL - Rust でモダンなターミナルアプリを構築するためのライブラリ一式です
  - Ansic - Rust 向けの、モダンで効率的なコンパイル時 ANSI マクロ・ユーティリティクレートです
- C
  - ncurses - プログラマがターミナルに依存しない形でテキストベースのユーザーインターフェースを記述できるアプリケーションプログラミングインターフェース (API) を提供するプログラミングライブラリです

デスクトップアプリ開発

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > フロントエンドシステム開発

クライアント OS・環境

- Windows 環境
  - パッケージ管理と運用管理
    - Chocolatey - Windows 向けのパッケージマネージャーです
    - Scoop - Windows 向けのコマンドラインインストーラーです
    - WinGet - Windows 上でアプリケーションの検索、インストール、アップグレード、削除、設定を可能にするコマンドラインツールです
    - gsudo - オリジナルの Sudo と似たユーザー体験を持つ、Windows 向けの Sudo です
  - 生産性と自動化
    - AutoHotKey - ユーザーが小規模から複雑なものまでスクリプトを簡単に作成できる、Windows 向けの無料オープンソーススクリプト言語です
    - Clavier+ - キーボードショートカットでアクションをトリガーできるようにします
    - WinSSHTerm - PuTTY、WinSCP、VcXsrv を組み合わせた、Windows 向けのタブ対応 SSH ソリューションです

- Linux 上の Android
  - Waydroid - Wayland ベースのデスクトップ環境が動く通常の GNU/Linux システム上で、完全な Android システムを起動するコンテナベースのアプローチです

GUI システム・ウィンドウイング

- ディスプレイサーバー
  - X.org - X Window System のオープンソース実装です
  - Wayland - X11 ウィンドウシステムのプロトコルとアーキテクチャの後継です
- デスクトップ環境
  - GNOME - あなたがコンピュータをコントロールし物事を成し遂げられるよう設計された、簡単でエレガントなコンピュータの使い方です
  - Xfce - UNIX 系オペレーティングシステム向けの軽量なデスクトップ環境です
- ウィンドウマネージャー
  - openbox - 高度に設定可能な次世代ウィンドウマネージャーです

デスクトップ GUI ツールキット

- 標準ツールキット
  - Tk - グラフィカルユーザーインターフェースツールキットです
    - tkinter - Tcl/Tk GUI ツールキットへの標準的な Python インターフェースです
    - CustomTkInter - Tkinter をベースにした Python UI ライブラリで、新しくモダンで完全にカスタマイズ可能なウィジェットを提供します
  - GTK - グラフィカルユーザーインターフェースを作成するための、無料でオープンソースなクロスプラットフォームウィジェットツールキットです
    - pygobject - GLib、GObject、GIO、GTK のオブジェクト指向 C ライブラリ向けの Python バインディング一式です
- コンパイル・キャンバスベース (Custom Rendering)
  - Slint - Rust、C++、JavaScript アプリ向けのネイティブユーザーインターフェースを構築する宣言的な GUI ツールキットです
  - Gio - Go でクロスプラットフォームな即時モード GUI を書くためのライブラリです
  - Fyne - デスクトップ、モバイル、Web 向けのグラフィカルアプリを作成する、習得しやすいツールキットです
- Web 技術ベース

- Chromium バンドル型
  - Electron - JavaScript、HTML、CSS を使ってデスクトップアプリケーションを構築するためのフレームワークです
- System WebView (Hybrid)
  - Tauri - 開発者が主要なデスクトッププラットフォーム向けのアプリケーションを作成するのを助けるツールキットです
  - Wails - 開発者が Go と Web 技術を使ってデスクトップアプリケーションを構築できるようにするツールです
  - Microsoft Edge WebView2 - Microsoft Edge をレンダリングエンジンとして使い、Web 技術 (HTML、CSS、JavaScript) をネイティブアプリへ組み込めるコントロールです

デスクトップアプリのテスト・自動化

- xa11y - macOS、Windows、Linux でネイティブデスクトップアプリを操作する、Playwright スタイルのライブラリです

インストール・パッケージング

- NSIS - Windows インストーラーを作成するためのプロフェッショナルなオープンソースシステムです
- Windows App CLI (winapp) - Windows SDK の管理、MSIX を用いたアプリのパッケージング、クロスプラットフォームフレームワーク向けのアプリ ID、マニフェスト、証明書の生成を行うコマンドラインインターフェースです
- PyInstaller - Python アプリケーションとその依存関係すべてを 1 つのパッケージにまとめるツールです

モバイルアプリ開発

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > フロントエンドシステム開発

モバイルプラットフォーム・ネイティブ SDK

- Android - 改変された Linux カーネルとその他のオープンソースソフトウェアを基にしたモバイルオペレーティングシステムです
  - Android Studio - Google の Android オペレーティングシステム向けの公式統合開発環境です
- iOS - Apple Inc. が自社ハードウェア専用に作成・開発したモバイルオペレーティング

システムです

モバイルクロスプラットフォームフレームワーク

- キャンバスベース (Custom Rendering)
  - Flutter - Google が作成したオープンソースの UI ソフトウェア開発キットです
- ネイティブウィジェットブリッジ
  - React Native - React でネイティブアプリを構築するためのフレームワークです
    - Expo - Android、iOS、Web でネイティブに動作するユニバーサル React アプリケーション向けのフレームワークおよびプラットフォームです
  - .NET MAUI - C# と XAML でネイティブなモバイル・デスクトップアプリを作成するためのクロスプラットフォームフレームワークです
- Web 技術ベース
  - Capacitor - 高性能なモバイルアプリ、デスクトップアプリ、プログレッシブ Web アプリを簡単に構築できるクロスプラットフォームネイティブランタイムです
  - Ionic Framework - Web 技術を使って高性能で高品質なモバイル・デスクトップアプリを構築するためのオープンソース UI ツールキットです

モバイル DevOps・テスト

- fastlane - iOS と Android アプリのベータ配布とリリースを自動化する最も簡単な方法です
- Appium - ネイティブ、ハイブリッド、モバイル Web アプリで利用できるオープンソースのテスト自動化フレームワークです

アプリケーションサービス・機能

- 通知
  - Firebase Cloud Messaging - 無料で確実にメッセージを送信できるクロスプラットフォームメッセージングソリューションです
  - Apple Push Notification service - 開発者のサーバーから Apple デバイス上のアプリへプッシュ通知を伝播させるサービスです
- デバイスハードウェア/OS 連携
  - GPS - アメリカ合衆国政府が所有し、United States Space Force が運用する衛星測位システムです
  - QR code - 1994 年に日本のデンソーウェーブ社が発明したマトリックス型バーコードの一種です

        - libqrencode - 高速でコンパクトな QR コードエンコーディングライブラリです
        - Pure python QR Code generator - Python で QR コードを簡単に作成できるライブラリです
        - QR code payment - モバイルアプリから QR コードをスキャンして決済を行う非接触型の決済方法です
- Backend-as-a-Service (BaaS)
    - Firebase - ユーザーに愛されるアプリやゲームの構築・成長を支援するアプリ開発プラットフォームです
    - Supabase - データベース、認証、インスタント API、エッジ関数、リアルタイムサブスクリプション、ストレージ、ベクトル埋め込みを提供する Postgres 開発プラットフォームです
    - AWS Amplify - フロントエンドの Web・モバイル開発者が AWS 上でフルスタックアプリケーションを簡単に構築、出荷、ホストできる完全なソリューションです

モノのインターネット (IoT)

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.2 デジタルテクノロジー > フィジカルコンピューティング

IoT の概念

- Internet of things (IoT) - センサー、ソフトウェア、その他の技術が組み込まれ、インターネットを通じて他のデバイスやシステムと接続・データ交換を行う目的を持つ物理的な物 (もの) のネットワークです
- Edge computing - 計算とデータストレージをデータの発生源に近づける分散コンピューティングパラダイムです
- Machine to machine - 有線・無線を含む任意の通信チャネルを使ったデバイス間の直接通信です
- Cyber-physical system - インターネットとそのユーザーに緊密に統合された、コンピュータベースのアルゴリズムによって制御・監視される仕組みです
- Digital twin - シミュレーション、テスト、監視、保守などの目的でデジタルの対応物として機能する、意図された、または実際の現実世界の物理的な製品、システム、プロセスのデジタルモデルです
- Firmware - デバイス固有のハードウェアに対して低レベルの制御を提供する、特定のクラスのコンピュータソフトウェアです

- Over-the-air update - モバイルデバイスへ新しいソフトウェア、ファームウェア、その他のデータを無線で配信することです

通信規格

- Wi-Fi - IEEE 802.11 系の規格に基づく無線ネットワークプロトコル群で、デバイスのローカルエリアネットワーキングやインターネットアクセスに広く使われています
- Bluetooth Low Energy - 従来の Bluetooth と同程度の通信範囲を維持しながら、消費電力とコストを大幅に削減するよう設計された無線パーソナルエリアネットワーク技術です
- Zigbee - 小型で低電力なデジタル無線を使ったパーソナルエリアネットワークを構築するために使われる、IEEE 802.15.4 をベースとした高レベル通信プロトコル群の仕様です
- Near-field communication - 4 cm (1+1⁄2 in) 以下の距離で 2 つの電子機器間の通信を可能にする通信プロトコル群です

IoT ハードウェアプラットフォーム

- Raspberry Pi - 使って学ぶための、小型で手頃な価格のコンピュータです
- Arduino - 電子工作プロジェクトにおける革新と創造性を可能にする、使いやすいオープンソースのハードウェア・ソフトウェアプラットフォームです
- ESP32 - Wi-Fi と Bluetooth 接続を統合した、IoT アプリケーション向けに設計された超低消費電力の高機能マイクロコントローラユニットです
- ESP8266 - 32 ビットの Tensilica プロセッサを搭載し低消費電力を特徴とする、IoT アプリケーション向けの費用対効果の高い高集積 Wi-Fi MCU です
- BeagleBone - 1 GHz の ARM Cortex-A8 プロセッサを搭載し、10 秒未満で Linux を起動する、低コストでコミュニティに支えられた開発プラットフォームです
- STM32 - 高性能、リアルタイム性能、低消費電力動作を提供するよう設計された、Arm Cortex-M プロセッサをベースにした 32 ビットマイクロコントローラファミリーです
- Nordic nRF52 - マルチプロトコル無線接続をサポートする 64 MHz の ARM Cortex-M4 プロセッサを中心に構築された、Bluetooth Low Energy System-on-Chip ファミリーです
- Particle - インテリジェントなデバイスの開発、接続、管理のためのエッジからクラウドまでのインフラを提供する、統合 IoT プラットフォームアズアサービスです
- BBC micro:bit - 実践的なコーディングと創造的なテクノロジープロジェクトを通じて、子どもたちが最高のデジタルな未来を作り出すよう促すために設計された、ポケットサイズのプログラム可能なコンピュータです

- Teensy - Arduino ソフトウェアと互換性があり、多くの種類のプロジェクトを実装できる、コンパクトな USB ベースのマイクロコントローラ開発ボードです
- Adafruit Feather - 内蔵バッテリーコネクタとモジュール式の拡張機能を備えた、組み込みプロジェクト向けの標準として設計されたコンパクトで携帯性の高いマイクロコントローラボードのファミリーです

IoT クラウドプラットフォーム

- Azure IoT Hub - IoT アプリケーションと接続されたデバイス間の通信のための中央メッセージハブとして機能する、マネージドクラウドベースサービスです
- AWS IoT Core - デバイスをクラウドへ簡単かつ安全に接続し、デバイス群を大規模に管理できるマネージドクラウドサービスです
- AWS IoT Greengrass - デバイスソフトウェアの構築、デプロイ、管理を行う、オープンソースのエッジランタイムおよびクラウドサービスです
- ThingWorx - 組織がデバイスを接続し、データを分析し、カスタマイズ可能なアプリケーションを通じてリアルタイムの洞察を提供できるよう支援する、包括的な産業用 IoT アプリケーション開発プラットフォームです
- Balena - セキュアな OTA アップデートとリモートアクセスを備え、開発者が Linux デバイスの群れを構築、デプロイ、管理、拡張できるコンテナベースの IoT デバイス管理プラットフォームです
- Arduino Cloud - ダッシュボードとリモートアクセスを使い、どこからでも接続されたプロジェクトを構築、制御、監視できるオールインワンの IoT プラットフォームです
- ThingSpeak - MATLAB との連携により、ライブデータストリームをクラウド上で集約、可視化、分析できる IoT 分析プラットフォームサービスです
- Losant - デバイスやシステムからのデータをリアルタイムに収集、統合、可視化することで、組織がスケーラブルな接続ソリューションを構築できるようにするエンタープライズ IoT プラットフォームです
- Adafruit IO - あなたのプロジェクトをモノのインターネットに乗せる最も簡単な方法で、Web ベースのマイクロコントローラ操作とデータロギングのプラットフォームとして機能します
- ThingsBoard - 業界標準プロトコルを介したデバイス接続を可能にする、データ収集、処理、可視化、デバイス管理のためのオープンソース IoT プラットフォームです
- Ubidots - 企業がカスタマイズ可能なダッシュボードと自動化されたワークフローを通じて、センサーデータをリアルタイムで接続、監視、可視化、対応できるようにするクラウドベースの産業用 IoT プラットフォームです

ローコード・ノーコード開発

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > サービス活用

ビジネスアプリケーションプラットフォーム

- Microsoft Power Apps - あらゆるデバイス向けのプロフェッショナルグレードのアプリを迅速かつ効率的に構築するプラットフォームです
- AppSheet - 生産性を高める強力なアプリケーションと自動化の構築を支援する、ノーコードプラットフォームです
- OutSystems - 重要なエンタープライズアプリケーションの開発を加速する、高性能なローコードプラットフォームです

ワークフロー・連携自動化

- n8n - AI 機能とビジネスプロセス自動化を組み合わせた、フェアコードライセンスのワークフロー自動化ツールです
- Microsoft Power Automate - ビジネスプロセスを最適化するために構築された、企業向けのエンドツーエンド自動化ソリューションです
- Zapier - コーディングの知識なしにアプリを連携させ、ワークフローを自動化できるツールです
- Microsoft Copilot Studio - 生成 AI と、事前構築済みコネクタ・アクションのライブラリを使って Copilot を作成・保守するグラフィカルなローコードツールです

Web コンテンツ・ポータルビルダー

- Microsoft Power Pages - 組み込みエージェントを備えたエンタープライズグレードの AI 駆動ビジネスポータルを素早く作成するプラットフォームです
- Webflow - コーディングなしで本番運用可能な Web サイトを構築する力を与える、ブラウザベースのデザインツールです

12 - パーソナルスキル

基礎的思考と論理学

> **Relevant DSS-P Skills**
>
> - 5. パーソナルスキル > 5.2 コンセプチュアルスキル > 創造的な問題解決
> - 5. パーソナルスキル > 5.2 コンセプチュアルスキル > 批判的思考

論理学

- Logic - 正しい推論を研究する学問です
  - Logical reasoning - 厳密な方法で結論に到達することを目指す精神活動です
    - 演繹的、帰納的、アブダクション的、類推的
  - First principle - 他のいかなる命題や仮定からも演繹できない基本的な命題または仮定です
- 論理学の分野
  - Informal Logic - 日常生活で用いられる議論を分析・評価するさまざまな手法を指す広範な用語です

非形式論理学

- Argument - 非形式論理学における中心的な研究対象であり、ある陳述 (結論) の真偽の程度を決定することを意図した一連の陳述 (前提) です
  - Enthymeme - 前提の一つが明示的に述べられていない議論であり、現実世界の推論によく見られる特徴です
- 議論評価の基準
  - Fallacy - 議論の構築において無効な、あるいは欠陥のある推論を用いることであり、気づかれなければ理にかなっているように見えることがあります
  - Category mistake - ある性質を持ちえない物事にその性質を帰属させてしまう推論上の誤りです
- 関連分野
  - Rhetoric - 説得の技術です
  - Critical Thinking - 健全な結論や十分な情報に基づく選択を行うために、利用可能な事実、証拠、観察、議論を分析するプロセスです

数理論理学

> **Relevant DSS-P Skills**
>
> - 5. パーソナルスキル > 5.2 コンセプチュアルスキル > 批判的思考

基礎概念

- Formal system - 公理体系を抽象化・形式化したものであり、一連の推論規則を用いて公理から定理を導出するために使われます
- Gödel's incompleteness theorems - 基本的な算術をモデル化できるあらゆる形式的公理系に内在する限界を示す、数理論理学における 2 つの定理です

- 論理法則
  - De Morgan's laws - いずれも有効な推論規則である、一対の変換規則です
  - Law of noncontradiction - 任意の命題について、その命題とその否定が同時に真であることはあり得ないとする法則です
  - Law of excluded middle - すべての命題について、その命題かその否定のいずれかが真であるとする原則です
  - Peirce's law - 古典論理において、任意の命題に対して排中律が成り立つとする原則です
  - Proof by contradiction - ある命題が偽であると仮定すると矛盾に至ることを示すことで、その命題が真であることを証明する間接証明の一形式です

論理体系

- Propositional calculus - 命題（真か偽かのいずれかである）と命題間の関係を扱う論理学の一分野であり、それらに基づく議論の構築も含みます
  - 連言、選言、含意、双条件、否定
  - Tautology - 論理定数のみが固定された意味を持ち、構成要素の解釈にかかわらず常に真となる式です
- First order logic - 数学、哲学、言語学、コンピュータサイエンスで用いられる形式体系の集合です
  - 全称量化と存在量化
- Higher order logic - 追加の量化子や、時にはより強い意味論によって一階論理と区別される論理の形式です
- Modal logic - 可能性と必然性に関する陳述を表現するために用いられる論理の一種です

数理論理学の分野

- Set theory - 集合を研究する数理論理学の分野であり、集合は非形式的にはものの集まりとして記述できます
- 素朴集合論
  - Set - 異なるものの集まりであり、その要素は集合の要素またはメンバーと呼ばれ、通常はあらゆる種類の数学的対象です
  - Function (a.k.a. Map) - 2 つの集合の間の二項関係であり、最初の集合のすべての要素を 2 番目の集合のちょうど 1 つの要素に関連付けるものです
  - Operation - ある集合からそれ自身への関数です

        - Idempotence - 最初の適用を超えて結果を変えることなく複数回適用できる、特定の演算の性質です
    - Partition of a set - 集合の要素を、すべての要素がちょうど 1 つの部分集合に属するように、空でない互いに素な部分集合 (「ブロック」または「セル」と呼ばれる) にグループ分けすることです
    - Equivalence relation - 反射的・対称的・推移的な二項関係であり、集合を互いに素な同値類に分割するものです
- 公理的集合論
    - Zermelo–Fraenkel set theory - ラッセルのパラドックスのような矛盾のない集合論を定式化するために 20 世紀初頭に提案された公理体系です
        - 順序数と基数
- Type Theory - (集合論と同様に) 数学に対する代替的な基礎を提供する形式体系であり、型付き関数型プログラミングや証明支援系の基盤です
    - Curry-Howard correspondence - コンピュータプログラムと数学的証明の間の直接的な対応関係です
- Proof Theory - 証明を形式的な数学的対象として表現し、数学的手法による分析を可能にする、数理論理学の主要な分野です
    - Sequent calculus - 定理を証明するための演繹体系です
    - Natural deduction - 論理的推論が「自然な」推論方法に密接に関連した推論規則によって表現される、証明計算の一種です
- Computability Theory - 計算可能関数とチューリング次数の研究として 1930 年代に始まった、数理論理学、コンピュータサイエンス、計算理論の分野です
    - Lambda calculus - 関数の抽象化と適用に基づいて計算を表現するための、数理論理学における形式体系です
    - Turing machine - 規則の表に従ってテープ上の記号を操作する抽象機械を記述する、計算の数学的モデルです
- Model Theory - 形式理論 (形式言語における文の集合) とそのモデル (文が真となる構造) との関係を研究する分野です
- 論理学の応用
    - Constraint satisfaction problem - 状態がいくつかの制約や制限を満たさなければならない対象の集合として定義される数学的問題です
    - Satisfiability modulo theories - ある数学的な式が充足可能かどうかを判定する問題です
    - Automated theorem proving - コンピュータプログラムによって数学的定理を証明することを扱う、自動推論と数理論理学の一分野です

    - Formal verification - 数学の形式的手法を用いて、システムがある形式的仕様や性質に関して正しいかどうかを証明または反証する行為です
        - Hoare logic - コンピュータプログラムの正しさを厳密に推論するための、一連の論理規則を持つ形式体系です
- 形式論理ツール
    - Stanford Encyclopedia of Philosophy - 世界中の哲学および関連分野の研究者を組織し、最新のコンテンツを作成・維持する参考資料です
    - SMT-LIB - テキストインターフェースを介して SMT ソルバーとやり取りするためのコマンド言語です
    - MiniZinc - 無料でオープンソースの制約モデリング言語です
    - P - 複雑な分散システムを形式的にモデル化・仕様化するための、状態機械ベースのプログラミング言語です
    - Lean - Calculus of Constructions に基づく対話型定理証明支援系兼プログラミング言語です

ドキュメンテーション

> **Relevant DSS-P Skills**
>
> - 3. テクノロジー > 3.1 ソフトウェア開発 > チーム開発
> - 5. パーソナルスキル > 5.1 ヒューマンスキル > コラボレーション

- Technical writing - 指示、説明、解説を必要とする特定の主題について著者が執筆する文章の一種です
- Divio Documentation System - すべてのドキュメントはその目的に応じてチュートリアル、ハウツーガイド、技術リファレンス、解説の 4 つの明確な種類に構造化されるべきだと提案するフレームワークです

アーキテクチャ文書

- 作図ツール
    - draw.io - 作図アプリケーションを構築するための技術スタックであり、世界で最も広く使われているブラウザベースのエンドユーザー向け作図ソフトウェアです
- コードとしての作図
    - D2: Declarative Diagramming - テキストを図に変換する、モダンな図表記述言語です
    - Diagrams - Python コードでクラウドシステムアーキテクチャを描画するための Python パッケージです

    - PlantUML - シンプルなテキスト記述から図を作成できるツールです
    - Mermaid - Markdown にインスパイアされたテキスト定義をレンダリングして図を動的に作成・変更する、JavaScript ベースの作図・グラフ作成ツールです
    - Kroki - プレーンテキストの図を画像に変換する、無料でオープンソースのサービスです
    - Graphviz - オープンソースのグラフ可視化ソフトウェアです
        - DOT language - プレーンテキストのグラフ記述言語です
        - sfdp - 大規模な無向グラフに対して、辺の交差を最小化し節点の重なりを回避する、スケーラブルなマルチスケール力学的レイアウトエンジンです
        - haphviz - DOT 形式でグラフを表現・操作・整形出力するための Haskell ライブラリです
    - ditaa - アスキーアートで描かれた図を適切なビットマップグラフィックに変換できる、小さなコマンドラインユーティリティです
- アーキテクチャ決定記録
    - Architectural Decision Records (ADRs) - 行われた重要なアーキテクチャ上の決定を、その背景と結果とともに記録する文書です
        - MADR (Markdown Architectural Decision Records) - アーキテクチャ上の決定を Markdown で記録するための簡潔なテンプレートであり、決定を書き留めてバージョン管理することをできるだけ容易にするものです
        - adr-tools - アーキテクチャ決定記録の管理を助けるコマンドラインツールです
        - Log4brains - 開発者が IDE から直接決定を記録し、静的サイトとして公開できるようにする、アーキテクチャ知識ベースです

軽量マークアップとライティングスタイル

- 軽量マークアップ
    - Markdown - プレーンテキストエディタを使って整形されたテキストを作成するための軽量マークアップ言語です
        - CommonMark - Markdown 構文を合理化したバージョンであり、仕様と、C 言語および JavaScript による BSD ライセンスのリファレンス実装を備えています
        - GFM (GitHub Flavored Markdown) - CommonMark 仕様に基づき、GitHub 独自の Markdown 方言の構文と意味を定義する正式な仕様です
            - github-markdown-css - GitHub 上でレンダリングされた Markdown をスタイリングする CSS です
        - markdownlint - Markdown/CommonMark ファイル向けの Node.js スタイ

ルチェッカー兼リントツールです
        - Glow - ターミナルベースの Markdown リーダーです
        - mdterm - Rust で書かれたターミナルベースの Markdown ビューアであり、シンタックスハイライト、スタイル付きフォーマット、対話的なナビゲーションを備えて Markdown ファイルをレンダリングします
        - Grip - README ファイルを GitHub にプッシュする前にローカルでレンダリングするコマンドラインサーバーアプリケーションです
        - markmap - Markdown とマインドマップを組み合わせたものです
        - Marp - プレーンな Markdown で書ける、最もシンプルな Markdown プレゼンテーション作成ツールです
            - Markdown all-in-one - Markdown 向けのオールインワンツールです (キーボードショートカット、目次、自動プレビューなど)
            - Markdown Preview Enhanced - Visual Studio Code 向けの超強力な Markdown 拡張機能です
            - Markdown Preview for (Neo)vim - (neo)vim 向けの Markdown プレビュープラグインです
        - ガイド
            - Markdown Guide - Markdown の使い方を説明する、無料でオープンソースのリファレンスガイドです
    - DocUtils - プレーンテキストのドキュメントを HTML、LaTeX、man ページ、OpenDocument、XML などの有用な形式に変換するための、オープンソースのテキスト処理システムです
        - reStructuredText - 読みやすく WYSIWYG なプレーンテキストマークアップ構文とパーサーシステムです
    - Asciidoc - メモ、ドキュメント、記事、書籍、電子書籍、スライドショー、Web ページ、man ページ、ブログを書くための軽量マークアップ言語です
        - Asciidoctor - AsciiDoc コンテンツを HTML5、DocBook 5 (または 4.5)、その他の形式に変換するための、高速でオープンソースのテキストプロセッサ兼パブリッシングツールチェーンです
    - Org Mode - GNU Emacs 向けの執筆ツール兼 TODO リスト管理ツールです
        - nvim-orgmode - Lua で書かれた Neovim 向けの Org Mode クローンです
    - Wikitext - MediaWiki ソフトウェアがページを整形するために使用する構文とキーワードからなるマークアップ言語です
- スタイルガイド

    - Microsoft Writing Style Guide - アプリやウェブサイトを含むさまざまな種類のコンテンツを作成する執筆者向けのガイドです
    - Google documentation style guide - ソフトウェア開発者などの技術的な読者に向けて、明確で一貫性のある技術文書を書くための編集ガイドラインです
    - Red Hat documentation style guide - Red Hat の製品文書や製品横断的なソリューション文書に対するスタイルガイドラインを提供するガイドです
    - Microsoft Terminology - 特定の言語に関する言語表現とスタイルの規則を定義するルール集です
    - List of English words - 46 万 6,000 語を超える英単語を含むテキストファイルです
- 文章リンター
    - vale - 自然言語・文章向けのリンターです
    - retext - 拡張可能な自然言語処理系です
    - alex - 性差別的、対立を煽る、人種に関する、宗教に配慮を欠く、その他不平等な表現を文章中から見つけるのに役立つツールです
    - write-good - 素朴な英文リンターです
    - textlint - テキストや Markdown 向けのプラグイン可能なリントツールです

ドキュメンテーションツール

- 組版システム
    - Typst - LaTeX と同等の強力さを持ちながら、学習も使用もはるかに容易であるように設計された、新しいマークアップベースの組版システムです
    - Troff/Groff - 整形コマンドが混在したプレーンテキストを読み込み、整形済みの出力を生成する組版システムです
    - LaTeX - 高品質な組版システムであり、技術文書や科学文書の作成のために設計された機能を含んでいます
        - TexLive - TeX 組版システム向けの、クロスプラットフォームなフリーソフトウェアディストリビューションです
        - PGF/TikZ - グラフィックを生成するための TeX マクロパッケージです
    - KaTeX - Web 向けの最速の数式組版ライブラリです
        - sphinxcontrib-katex - KaTeX を使って Sphinx ドキュメント内の数式をレンダリングできるようにする Sphinx 拡張です
- 検証と保守
    - lychee - Rust で書かれた高速な非同期リンクチェッカーです

- コンバーター
  - Pandoc - 汎用のドキュメント変換ツールです
    - Eisvogel - Markdown ファイルを PDF や LaTeX に変換するための pandoc LaTeX テンプレートです

対人関係とチームリーダーシップ

> **Relevant DSS-P Skills**
>
> - 5. パーソナルスキル > 5.1 ヒューマンスキル > リーダーシップ
> - 5. パーソナルスキル > 5.1 ヒューマンスキル > コラボレーション

チームダイナミクスとコミュニケーション

- チームダイナミクス
  - Team building - チーム内の社会的関係を強化し役割を明確にするために使われる、さまざまな種類の活動の総称であり、しばしば協働タスクを伴います
    - Tuckman's stages of group development - 1965 年に Bruce Tuckman によって最初に提唱された、集団発達のモデルです
  - Group dynamics - 社会集団の内部または集団間で生じる行動と心理的プロセスの体系です
  - 研究とモデル
    - Google Rework: Understand team effectiveness - 心理的安全性、信頼性、構造と明確さ、意義、影響力といった主要なダイナミクスが、成功する集団協働に不可欠であると特定した研究イニシアチブです
    - GRPI Model - チームの有効性を設計・診断するための基礎的なフレームワークであり、高パフォーマンスなチームの構成要素を目標、役割、プロセス、対人関係という 4 つの層に整理するものです
- 対人コミュニケーション技法
  - Storytelling - 時に即興や演出、脚色を伴いながら物語を共有する社会的・文化的活動です
  - Facilitation - 成功する会議やワークショップを設計し進行する行為です
  - Active listening - 聞く準備を整え、どのような言語的・非言語的メッセージが発せられているかを観察し、提示されているメッセージへの注意を示すために適切なフィードバックを行う実践です
  - Negotiation - 相違点を解決し、個人または集団のために優位性を得たり、さまざまな利害を満足させる結果を作り出したりするための、2 者以上の間の対話です

- 企業理念と価値観
  - Amazon's Leadership Principles - Amazon の従業員が日々の議論、意思決定、行動の指針として用いる中核的な信条の集合です
  - GitLab Values - 企業文化とチームメンバーの働き方を定義する指導原則の集合です
- プロフェッショナル宣言
  - Manifesto for Software Craftsmanship - ソフトウェア開発における、質の高いソフトウェア、継続的な価値の追加、専門家コミュニティ、生産的なパートナーシップの重要性を強調する宣言です
- フィードバックモデル
  - DESC feedback model - 行動を説明し、その影響を伝え、望む変化を明示し、結果を説明することによって建設的なフィードバックを行うためのコミュニケーションツールです

組織行動

- Stakeholder management - プロジェクトや事業によって影響を受ける個人や集団を特定し、その関心事や懸念を理解し、その期待や影響力を管理するプロセスです
- Contingency theory - 企業を組織し、会社を率い、意思決定を行うための唯一最善の方法は存在しないとする理論であり、最適な行動方針は内的・外的状況に依存すると主張します
- Holacracy - 分散型のマネジメントと組織統治の手法であり、権限と意思決定を管理階層に委ねるのではなく、自己組織化するチームのホラーキーを通じて分散させると主張します
- Expectancy theory - 個人の行動はその行動から期待される結果によって動機づけられるとする理論であり、結果の望ましさが特定の行動の選択を決定するとします
- Intrinsic motivation - 楽しさ、好奇心、達成感といった内的要因から生じる動機づけの一種であり、個人が活動そのもののためにその活動に取り組むことです
- Management 3.0 - 組織を管理し業務システムを改善するのを助けるために設計された、常に進化し続けるマインドセットとゲーム・ツール・実践の集合です
- PM Theory of Leadership - リーダーを成果志向型と維持志向型の 2 つのカテゴリーに分類するリーダーシップ理論です
- Theory X and Theory Y - Douglas McGregor によって開発された、人間の労働意欲とマネジメントに関する理論です
- Two-factor theory - 心理学者 Frederick Herzberg によって開発された理論であり、職務満足と不満足はそれぞれ独立に作用する動機づけ要因と衛生要因という別々の要因の組によって影響を受けると仮定します

リーダーシップスタイル

- Servant leadership - 個人の成功や従来型の階層的権威に焦点を当てるのではなく、責任者の第一の目的をチームメンバーのニーズを優先し、その成長とパフォーマンスを育むことに置くリーダーシップ哲学です
- Shared leadership - 責任を広く分散させ、チームや組織内の個人が互いに主導し合えるようにするリーダーシップスタイルです
- Situational leadership - 唯一普遍的に適切なアプローチは存在しないと認識し、効果的なリーダーがそれぞれの状況にスタイルを適応させるリーダーシップモデルです
- Transformational leadership - リーダーが従業員に対し、会社の将来の成功を成長させ形作るイノベーションと変化を促すよう奨励し、鼓舞し、動機づけるリーダーシップスタイルです

個人の心理とパフォーマンス

> **Relevant DSS-P Skills**
>
> - 5. パーソナルスキル > 5.2 コンセプチュアルスキル > 適応力

個人のパフォーマンス

- メンタルヘルス
  - Mindfulness - 完全に今この瞬間に存在し、自分がどこにいて何をしているかを意識し、周囲で起きていることに過度に反応したり圧倒されたりしない、人間の基本的な能力です
    - Zen - 唐代の中国で生まれた大乗仏教の一派です
  - Flow - 何らかの活動を行っている人が、活動のプロセスに対するエネルギーに満ちた集中、完全な没入、楽しさの感覚に完全に浸っている精神状態です
  - Defence mechanism - 内的葛藤や外的ストレス要因に関連する不安を引き起こす思考や感情から自己を守る、無意識の心理的プロセスです
  - Psychological resilience - 危機に精神的・感情的に対処する能力、または危機前の状態に速やかに戻る能力です
  - Occupational burnout - うまく管理されなかった慢性的な職場ストレスに起因する、仕事に関連した現象です
- 認知パフォーマンスと意思決定
  - Maslow's Hierarchy of Needs - 人間の行動を動機づける欲求（または目標）の概念化です
  - Cognitive bias - 判断における規範や合理性からの逸脱の体系的なパターンです

        - Anchoring effect - 個人の判断や意思決定が、まったく無関係であり得る基準点すなわち「アンカー」に影響される心理現象です
    - Zeigarnik effect - 人が完了したタスクよりも未完了・中断されたタスクをよく記憶しているという現象です
    - Default mode network - 大規模な脳ネットワークであり、人が外の世界に集中しておらず脳が覚醒状態で休息しているときに活動することで知られています
    - Situation awareness - 環境とその要素、および時間やその他の要因に応じてそれがどのように変化するかについての理解です
        - 1: 環境内の要素の知覚
        - 2: 状況の把握または理解
        - 3: 将来の状態の予測
    - Vertical thinking - 合理的評価と外部データに依拠することが多い、選択的・分析的・逐次的であることを特徴とする問題解決アプローチです
    - Lateral thinking - 直接的には明らかでない推論を通じて、間接的で創造的なアプローチを用いて問題を解決する方法です
- 関連する思想
    - Three Virtues - 優れたプログラマーの資質、すなわち怠惰・短気・傲慢です
- 関連書籍
    - Thinking, Fast and Slow - 心理学者 Daniel Kahneman による 2011 年の書籍です

社会的パフォーマンス

- 社会心理学
    - Psychological safety - アイデアや質問、懸念、間違いを口にしても罰せられたり恥をかかされたりしないという信念です
    - Trust - 他者が期待通りのことをするという信念です
    - Collective intelligence - 多くの個人の協働、集団的努力、競争から生まれ、合意形成による意思決定に現れる、共有された、あるいは集団の知性です
    - Groupthink - 集団内の調和や同調への欲求が、不合理または機能不全な意思決定の結果をもたらす、人々の集団内で生じる心理現象です
    - Bystander effect - 他の人々が居合わせているとき、個人は被害者を助ける可能性が低くなるとする社会心理学の理論です
    - Dunbar's number - 人が安定した社会的関係を維持できる人数について提唱された認知的な限界です

    - Norm of reciprocity - 他者が自分にしてくれたことと同種のことを返すという社会的期待です
- 具体例となる概念
    - Broken windows theory - 犯罪、反社会的行動、市民の秩序崩壊の目に見える兆候が、重大な犯罪を含むさらなる犯罪と秩序崩壊を助長する都市環境を作り出すとする犯罪学の理論です
    - Stone soup story - 空腹の見知らぬ人々が、食事を作るために町の人々それぞれに少量の食料を分けてもらうよう説得する、ヨーロッパの民話です
    - Boiling frog apologue - カエルがゆっくりと生きたまま茹でられる様子を描いた寓話です

タイムライン - 1930-89

1930s

📜 ラムダ計算は、変数の束縛と置換を用いた関数の抽象化と適用に基づいて計算を表現するための、数理論理学における形式体系です。数学の基礎に関する研究の一環として 1930 年代に数学者 Alonzo Church によって導入されましたが、当初の体系は 1935 年に論理的に矛盾していることが示されました。

📜 チューリングマシンは、規則の表に従ってテープ上の記号を操作する抽象機械を記述した計算の数学的モデルであり、1936 年に Alan Turing が考案し、後に博士課程の指導教員であった Alonzo Church が書評の中で名付けました。

1940s

🎨 Maslow の欲求階層説は、Abraham Maslow が雑誌 Psychological Review に掲載された 1943 年の論文 "A Theory of Human Motivation" で提唱した心理的健康に関する理論であり、人間の欲求は基本的な生理的欲求から自己実現に至るまでの階層に分類できると論じました。

📜 コンピュータプログラミングにおいて、アセンブリ言語とは、その言語の命令とアーキテクチャの機械語命令との間に非常に強い対応関係がある低水準プログラミング言語の総称であり、機械語命令を表現するために用いられた最初のアセンブリコードは Kathleen Booth と Andrew Donald Booth による 1947 年の研究に登場しました。

1950s

🧠 チューリングテストは、1950 年に Alan Turing が当初イミテーションゲームと呼んだもので、人間と同等の、あるいは人間と見分けのつかない知的な振る舞いを機械が示せるかどうかを測るテストです。

📜 正規表現の起源は 1951 年にさかのぼり、数学者 Stephen Cole Kleene が正規事象と呼ぶ自身の数学的記法を用いて正規言語を記述したことに始まります。

⚙ セシウム原子時計は 1955 年に英国の National Physical Laboratory (NPL) の Louis Essen によって初めて製作され、天文観測よりもはるかに安定的かつ精密な計時の手段をもたらし、現代の SI 秒の定義と協定世界時 (UTC) の基礎を築きました。

🎨 Parkinson の法則は「仕事の量は、その完成のために利用できる時間をすべて満たすまで膨張する」という格言であり、Cyril Northcote Parkinson が英国の官僚機構の肥大化に関する自身の経験を踏まえて 1955 年に The Economist に発表したユーモラスなエッセイの中で初めて述べたものです。

📜 Lisp は、長い歴史と、特徴的な完全に括弧で囲まれた前置記法を持つプログラミング言語のファミリーであり、1958 年に John McCarthy が Massachusetts Institute of Technology (MIT) で開発した際に最初の仕様が定められ、現在も一般に使われているものとしては 2 番目に古い高水準プログラミング言語となっています。

📜 ALGOL (「Algorithmic Language」の略) は、1958 年に最初に開発された命令型コンピュータプログラミング言語のファミリーであり、他の多くの言語に大きな影響を与え、アルゴリズム記述の標準的な手法となりました。

🧠 機械学習において、パーセプトロン (または McCulloch-Pitts ニューロン) は二値分類器の教師あり学習のためのアルゴリズムであり、その最初の実装は 1958 年に Cornell Aeronautical Laboratory で Frank Rosenblatt が製作した機械でした。

🧠 機械学習という用語は、1959 年に IBM の社員でありコンピュータゲームと人工知能の分野の先駆者であった Arthur Samuel によって作られました。この時期には自己学習型コンピュータという同義語も使われていました。

1960s

🔐 パスワードはコンピューティングの最初期からコンピュータで使われてきました。1961 年に MIT で導入されたオペレーティングシステムである Compatible Time-Sharing System (CTSS) は、パスワードによるログインを実装した最初のコンピュータシステムでした。

⚙ 1961 年 5 月、IBM のエンジニアであった Bob Bemer は American Standards Association (ASA) の X3.2 小委員会に提案を提出し、ASCII (American Standard Code for Information Interchange) の正式な標準化作業の口火を切りました。エスケープシーケンスも考案した Bemer は、後に「ASCII の父」として知られるようになりました。

📜 Simula は、1960 年代に Oslo の Norwegian Computing Center で Ole-Johan Dahl と Kristen Nygaard によって開発された 2 つのシミュレーション用プログラミング言語の名称で、1962 年に初めて登場しました。このうち Simula 67 は、オブジェ

クト、クラス、継承、サブクラス、仮想手続き、コルーチン、離散事象シミュレーション、ガベージコレクションを導入しました。

🧠 機械学習において、バックプロパゲーション (誤差逆伝播法) はニューラルネットワークモデルの訓練に用いられる勾配推定の手法です。「back-propagating error correction」という用語は 1962 年に Frank Rosenblatt によって導入されましたが、彼自身はそれを実装する方法を知りませんでした。もっとも、Henry J. Kelley は制御理論の文脈で 1960 年にすでにバックプロパゲーションの連続版の先駆けを得ていました。

🎨 イノベーションの普及は、新しいアイデアや技術がどのように、なぜ、どれくらいの速さで広まるのかを説明する理論であり、社会学者 Everett Rogers が 508 を超える普及研究を統合して 1962 年の著書 Diffusion of Innovations にまとめ、採用者を革新者、初期採用者、初期多数派、後期多数派、遅滞者に分類する枠組みを導入しました。

⚙ ASCII の初版 (ASA X3.4-1963) は、Teletype Model 33 の登場と時を同じくして 1963 年に発行されました。ASCII は最初、AT&T の TWX (TeletypeWriter eXchange) ネットワーク向けの 7 ビットのテレプリンタ用符号として商用に使われました。

💻 1964 年、Multics オペレーティングシステムのために、Louis Pouzin は「コマンドをある種のプログラミング言語のように使う」というアイデアを着想し、それを表すシェルという用語を作りました。

🧠 エキスパートシステムは、Edward Feigenbaum が率いる Stanford Heuristic Programming Project によって 1965 年ごろに正式に導入されました。Feigenbaum は「エキスパートシステムの父」と呼ばれることもあります。最初期のものの一つが Dendral で、質量分析データから未知の有機化合物の分子構造を同定するために設計されたこのシステムは、Bruce Buchanan および Joshua Lederberg とともに開発されました。

📜 Amdahl の法則は、あるタスクを実行するシステムに資源を追加していったときの高速化を制限する式であり、Gene Amdahl が 1967 年 4 月の AFIPS Spring Joint Computer Conference において論文 "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities" で発表しました。

🎨 Conway の法則は、組織は自らのコミュニケーション構造を反映したシステムを設計するという格言です。1967 年にこの考えを示したコンピュータプログラマの Melvin Conway にちなんで名付けられました。

🔐 PIN は、銀行が顧客に現金を渡す効率的な手段として 1967 年に現金自動預け払い機 (ATM) が導入されたことに始まります。最初の ATM システムは 1967 年に London の Barclays が導入したもので、カードではなく機械可読の符号が記された小切手を受け付け、PIN と小切手を照合するものでした。

⚙ 擬似端末は早くも 1967 年に DEC PDP-6 Timesharing Monitor に登場し、そこで

は主にバッチ処理を実装するために使われていました。

🎨 The Mother of All Demos は、1968 年 12 月 9 日に San Francisco で開催された ACM/IEEE Fall Joint Computer Conference で Douglas Engelbart が行った 90 分間のライブデモンストレーションであり、oN-Line System (NLS) を紹介するとともに、ウィンドウ、ハイパーテキスト、コンピュータマウス、ビデオ会議、ワードプロセッシング、リアルタイムの共同編集エディタを実演しました。

💻 ed は Unix および Unix 系オペレーティングシステム向けのラインエディタであり、1969 年 8 月に Ken Thompson が AT&T Bell Labs の PDP-7 上で開発し、アセンブラおよびシェルと並ぶ Unix オペレーティングシステムの最初の 3 つの重要な構成要素の一つとなりました。

📜 Hoare 論理は、コンピュータプログラムの正しさを厳密に推論するための一連の論理規則からなる形式体系です。1969 年に英国のコンピュータ科学者であり論理学者でもある Tony Hoare によって提案され、その後 Hoare や他の研究者たちによって洗練されていきました。

⚙ Unix (商標としては UNIX) は、1969 年に最初にリリースされたオリジナルの AT&T Unix に由来する、マルチタスクかつマルチユーザーのコンピュータオペレーティングシステムのファミリーです。

⚙ Advanced Research Projects Agency Network (ARPANET) は 1969 年に構築された、分散制御を備えた最初の広域パケット交換ネットワークであり、TCP/IP プロトコル群を実装した最初のネットワークの一つでもありました。その双方がインターネットの技術的基盤となりました。

⚙ コンピューティングにおける「server」という語は、少なくともインターネットの前身である ARPANET を記述した最初期の文書の一つである RFC 5 (1969) にまで遡り、「user」と対比されて、「server-host」と「user-host」という 2 種類のホストを区別していました。

⚙ Telnet (「teletype network」の略) は、ローカルエリアネットワークやインターネット上で遠隔システムの仮想端末へのアクセスを提供するクライアント／サーバ型のアプリケーションプロトコルであり、1969 年の RFC 15 を皮切りに秘密技術として開発されました。

⚙ TENEX は 1969 年に BBN が PDP-10 向けに開発したオペレーティングシステムであり、後に Digital Equipment Corporation の TOPS-20 オペレーティングシステムの基礎となりました。TENEX は、BBN が独自に設計したハードウェアページャによって実装された、すべてのプログラムが 262,144 ワードのアドレス空間にアクセスできる完全な仮想記憶システムを導入し、その EXEC コマンドインタプリタは、ユーザーがエスケープキーを押すと入力途中のコマンド語を完全な語に展開するコマンド補完を先駆けて実現しました。この機能は tcsh のような現代のシェルにも今なお見られます。

1970-74

🏗️「リレーショナルデータベース」という用語は、1970 年に IBM の E. F. Codd によって初めて定義されました。Codd は自身の研究論文 "A Relational Model of Data for Large Shared Data Banks" でこの用語を導入しました。

⚙️ File Transfer Protocol (FTP) は、コンピュータネットワーク上でサーバからクライアントへコンピュータファイルを転送するために用いられる標準的な通信プロトコルであり、1971 年 4 月 16 日に登場しました。

⚙️ mail は Unix および Unix 系オペレーティングシステム向けのコマンドライン電子メールクライアントであり、1971 年 11 月 3 日に最初にリリースされました。

📜 roff は組版用のマークアップ言語であり、Unix 最初のテキスト整形プログラムとして、nroff および troff 文書処理システムの前身にあたるもので、1971 年 11 月 3 日に最初にリリースされました。

⚙️ 1971 年に最初の ARPANET のネットワークメールが送信され、ユーザーのシステムアドレスを示す「@」記号を用いる、いまではおなじみのアドレス記法が導入されました。一連の RFC を通じて、File Transfer Protocol 上でメールメッセージを送信するための規約が整えられていきました。

📜 yacc (Yet Another Compiler Compiler) は、文脈自由文法のためのパーサジェネレータであり、LALR パーサ用の C コードを生成します。1971 年に Bell Labs の Stephen C. Johnson が Donald Knuth の LR 構文解析に関する研究に触発されて最初に開発したもので、その自嘲的な名称は、Multics プロジェクトに由来する他のコンパイラコンパイラが Bell Labs に存在していたことを反映しています。1973 年までに、のちの C 実装にそれと分かるほど近い形にまで到達していましたが、初期のバージョンは極めて遅く、Johnson は性能をおよそ 1 万倍改善するために十数回にわたって書き直しました。

📜 ML (Meta Language) は、1972 年に Stanford (後に Cambridge) の Robin Milner によって、LCF (Logic for Computable Functions) 証明支援系のためのメタ言語としてリリースされ、OCaml の基礎となる型システムをもたらしました。

📜 C は、Unix 上で動作するユーティリティを構築するために 1972 年から 1973 年にかけて Bell Labs の Dennis Ritchie が作成した中水準の汎用コンピュータプログラミング言語であり、対象とする CPU の能力を素直に反映した設計のおかげで、今なお非常に広く使われ、大きな影響力を持ち続けています。

📜 Smalltalk は、Xerox Palo Alto Research Center (PARC) で Alan Kay が主導した研究の成果として教育用途向けに設計された、オブジェクト指向で動的型付けのリフレクティブなプログラミング言語であり、最初のシステム (Smalltalk-72) は Xerox Alto 上で動作し、Kay の新しいオブジェクト指向プログラミングパラダイムを支えるよう設計され

ていました。

🏢 1972 年 6 月、AI 部門出身の 5 人の IBM エンジニアが、ドイツ民法上の私的パートナーシップとして SAP (Systemanalyse und Programmentwicklung、「システム分析とプログラム開発」) を設立し、1973 年に最初の商用製品である財務会計システム RF を発売しました。

🎨 Xerox PARC で開発され、1973 年 3 月 1 日に最初のマシンが登場した Xerox Alto は、デスクトップメタファーとビットマップ方式のグラフィカルユーザーインターフェースを実証した最初のコンピュータであり、後に Macintosh の設計に大きな影響を与えました。

⚙️ ローカルエリアネットワーク向けのネットワーク技術である Ethernet は、Xerox PARC の Robert Metcalfe によって発明され、1973 年 5 月 22 日に 2.94 Mbps のビットレートで初めて動作しました。Metcalfe はその頃に書いたメモの中でこの構想を「Ethernet」と名付けました。

⚙️ 1973 年、Version 4 Unix はより高水準の言語である C で書き直されました。これは、オペレーティングシステムはその複雑さと高度さゆえにアセンブリ言語で書かれる必要があるという当時の一般的な考え方に反するものでした。

💻 sed (「stream editor」) は、単純でコンパクトなプログラミング言語を用いてテキストを解析し変換する Unix のユーティリティであり、1973 年から 1974 年にかけて Bell Labs の Lee E. McMahon によって開発されました。

⚙️ TCP は、IP ネットワークを介して通信するホスト上で動作するアプリケーションの間で、オクテット (バイト) のストリームを信頼性が高く、順序どおりに、かつ誤り検査を伴って配送します。1974 年 5 月、Vint Cerf と Bob Kahn は、ネットワークノード間でパケット交換を用いて資源を共有するためのインターネットワーキングプロトコルを記述し、その結果生まれたプロトコル (TCP/IP) の仕様は Vint Cerf、Yogen Dalal、Carl Sunshine によって書かれ、1974 年 12 月に公表されました。

📊 Structured Query Language (SQL) は、リレーショナルデータベース管理システム (RDBMS) に保持されたデータを管理するため、あるいはリレーショナルデータストリーム管理システム (RDSMS) におけるストリーム処理のために設計された、プログラミングで用いられるドメイン固有言語であり、1974 年に初めて登場しました。

🧠 MYCIN は、菌血症や髄膜炎といった重篤な感染症を引き起こす細菌を人工知能によって同定し、患者の体重に合わせて用量を調整した抗生物質を推奨する初期の後方連鎖型エキスパートシステムであり、1970 年代前半に Stanford University において、Stanford Heuristic Programming Project の Bruce G. Buchanan らの指導のもとで Edward H. Shortliffe によって 5 年から 6 年かけて開発されました。

1975-79

🔐 Data Encryption Standard はデジタルデータを暗号化するための共通鍵アルゴリズムであり、鍵長が 56 ビットと短いため現代の用途には安全性が不十分ですが、暗号技術の発展に大きな影響を与えました。DES の起源は 1972 年にさかのぼり、米国政府のコンピュータセキュリティに関する National Bureau of Standards の調査により、機密指定外の重要情報を暗号化するための政府全体の標準が必要であることが明らかになりました。1975 年 3 月 17 日、提案された DES が連邦官報に公表され、パブリックコメントが募集されるとともに、翌年には提案された標準について議論する 2 回の公開ワークショップが開催されました。

🏢 Fred Brooks による The Mythical Man-Month: Essays on Software Engineering は 1975 年に初版が出版され、IBM の OS/360 プロジェクトを管理した経験に基づいて、すでに遅延しているソフトウェアプロジェクトに人員を追加するとさらに遅延すると論じました。この考えは Brooks の法則として知られるようになりました。

⚙️ cron コマンドラインユーティリティは Unix 系オペレーティングシステムにおけるジョブスケジューラであり、cron ジョブとも呼ばれるジョブ (コマンドやシェルスクリプト) を、決まった時刻・日付・間隔で定期的に実行するようスケジュールするために使われ、1975 年 5 月に最初にリリースされました。

📜 lex は字句解析器生成系であり、トークンの仕様を読み込んで字句解析器の C コードを生成します。Bell Labs の Mike Lesk と Eric Schmidt によって開発され、1975 年 10 月に Computing Science Technical Report #39 として最初に文書化されました。同じ年に Stephen C. Johnson が yacc の正式な論文を発表しており、yacc で実装された最初期の言語には AWK、C++、eqn、Pic がありました。

💻 Make はビルド自動化ツールであり、対象プログラムをどのように導出するかを指定した Makefile と呼ばれるファイルを読み込むことで、ソースコードから実行可能プログラムやライブラリを自動的にビルドします。1976 年 4 月に登場しました。

⚙️ Metcalfe と Boggs は 1976 年 7 月に Communications of the ACM で "Ethernet: Distributed Packet Switching for Local Computer Networks" を発表しました。

💻 vi はもともと Unix オペレーティングシステム向けに作られたスクリーン指向のテキストエディタであり、そのオリジナルのコードは 1976 年に Bill Joy によって、Joy が Chuck Haley と共に書いた ex というラインエディタのビジュアルモードとして書かれました。

🔐 Diffie–Hellman 鍵交換は公開された通信路上で暗号鍵を安全に交換する数学的手法であり、1976 年に Diffie と Hellman によって発表され、秘密鍵とそれに対応する公開鍵という考え方を提唱した最初期の公知の成果となりました。

📜 AWK はテキスト処理のために設計されたドメイン固有言語であり、一般にデータ抽出とレポート作成のツールとして使われます。1977 年に Alfred Aho (egrep の作者)、Peter J. Weinberger (小規模なリレーショナルデータベースに取り組んでいた人物)、Brian Kernighan によって最初に開発されました。

🔐 RSA は安全なデータ伝送に広く使われている公開鍵暗号方式であり、"RSA" という頭字語は 1977 年にこのアルゴリズムを公に記述した Ron Rivest、Adi Shamir、Leonard Adleman の姓に由来します。

🔐 Data Encryption Standard は 1977 年 1 月に標準化されました。鍵長が 56 ビットと短いため現代の用途には安全性が不十分ですが、暗号技術の発展に大きな影響を与えました。

💻 Bill Joy の ex 1.1 は 1978 年 3 月に最初の Berkeley Software Distribution (BSD) Unix リリースの一部として公開されました。このビジュアルモードの多くのアイデアは、Xerox PARC が Alto 向けに開発したバイモーダルなテキストエディタ Bravo から取り入れられたものです。

📜 1978 年に Brian Kernighan と Dennis Ritchie は The C Programming Language の初版を出版しました。この本は C プログラマの間で K&R として知られ、長年にわたって同言語の非公式な仕様として機能しました。この本が記述する C のバージョンは一般に "K&R C" と呼ばれます。

📜 TeX は Donald Knuth が設計・作成した組版システムであり、1978 年に最初にリリースされました。TeX78 と呼ばれる TeX の最初のバージョンは、Stanford の WAITS オペレーティングシステム上の PDP-10 で動作するよう SAIL プログラミング言語で書かれました。

🏢 大野耐一は 1978 年に代表的な著書『トヨタ生産方式 — 脱規模の経営をめざして』を出版し、その哲学を世界の読者に向けて体系化しました。

📜 ビザンチン合意を得るという問題は Robert Shostak によって考案・定式化され、彼はこれを対話型整合性問題と名付けました。この研究は 1978 年に、SRI International の Computer Science Lab における NASA 出資の SIFT プロジェクトの文脈で行われました。

💻 C シェル (csh) は、Bill Joy が 1970 年代後半に University of California, Berkeley の大学院生だったときに作った Unix シェルであり、Joy が 1978 年に最初に配布した Berkeley Software Distribution の 2BSD リリースから配布が始まりました。その構文は C プログラミング言語をモデルにしており、アイデアやコードに貢献した初期のメンバーには Michael Ubell、Eric Allman、Mike O’Brien、Jim Kulp がいました。

📜 Model-view-controller（MVC）はユーザーインターフェースの開発に広く使われるソフ

トウェアデザインパターンであり、関連するプログラムロジックを相互に接続された 3 つの要素に分割します。Trygve Reenskaug は 1970 年代後半に Xerox Palo Alto Research Center (PARC) の客員研究員として Smalltalk-79 に取り組む中で MVC を生み出しました。

💻 Bourne シェル sh は、Bell Labs の Stephen Bourne が開発した新しい Unix シェルであり、1979 年に UNIX Version 7 のシェルとして配布されました。

⚙️ Unix および Unix 系オペレーティングシステムにおける chroot は、現在実行中のプロセスとその子プロセスに対して見かけ上のルートディレクトリを変更する操作です。chroot システムコールは 1979 年の Version 7 Unix の開発中に導入されました。

📜 lex と yacc はいずれも 1979 年の Version 7 Unix のリリースによって Unix ツールチェーンの標準ツールとなり、Unix システム上でのコンパイラおよび言語開発の基礎的な構成要素としての地位を確立しました。

📊 Oracle Database (一般に Oracle DBMS、Oracle Autonomous Database、あるいは単に Oracle と呼ばれます) は Oracle Corporation が製造・販売するプロプライエタリなマルチモデルデータベース管理システムであり、1979 年に最初にリリースされました。

⚙️ Apple の社員だった Jef Raskin は 1979 年に Macintosh プロジェクトを立ち上げ、大衆向けの手頃で使いやすいコンピュータを構想しました。彼は自分の好きなリンゴの品種 McIntosh にちなんでプロジェクトを名付けました。

🧠 1979 年に Stanford Medical School で MYCIN の独立した評価が実施され、このシステムは 65% の受容性評価を得ました。これは教員である医師に与えられた 42.5% から 62.5% という評価に匹敵するか上回るものであり、ルールベースのエキスパートシステムが特定の診断タスクにおいて人間の専門家に匹敵する性能を達成できることを示しました。

🎨 プロスペクト理論は、人々が確率的な選択肢の間でどのように選択を行うかを記述する行動経済学・判断・意思決定の理論であり、1979 年の Econometrica 誌の論文 "Prospect Theory: An Analysis of Decision under Risk" において、Daniel Kahneman と Amos Tversky によって期待効用理論に代わる記述的な理論として提示されました。

🧩 世界規模の分散型ディスカッションシステムである Usenet は、1979 年に Duke University の Tom Truscott と Jim Ellis がローカルな告知プログラムの代替として思いついたアイデアに始まり、Steve Daniel と Truscott が書いたニュースソフトウェアが公開されました。

1980-84

⚙️ Digital Equipment Corporation、Intel、Xerox (DIX) は 1980 年 9 月 30 日に、"Blue Book" として知られる "The Ethernet: A Local Area Network — Data Link

Layer and Physical Layer Specifications" のバージョン 1.0 を公開しました。

📜 Smalltalk-80 は PARC の外部で利用可能になった最初の言語バリアントであり、まず Smalltalk-80 Version 1 として少数の企業と大学に提供されました。

🧩 Usenet は 1980 年に UC Berkeley を通じて ARPANET に接続され、元々の UUCP ダイヤルアップ接続を超えて、より広いネットワークへ初めてつながることとなりました。

⚙️ Steve Jobs は Lisa チームから外された後、1981 年に Macintosh プロジェクトを引き継ぎ、コンピュータを数百万台規模で生産するという Jef Raskin のビジョンを保ちながら、設計を Xerox PARC に触発されたグラフィカルユーザーインターフェースの方向へと転換しました。

⚙️ Berkeley r-commands は、ある Unix システムのユーザーが TCP/IP ネットワーク経由で別の Unix コンピュータにログインしたりコマンドを発行したりできるようにするために設計された一連のプログラムであり、1981 年 6 月に最初にリリースされました。

🧩 1981 年 6 月 1 日までに、UUCP/Usenet ネットワークは大学や研究機関にまたがる数十の接続システムを含むまでに拡大していました。

⚙️ IPv4 は RFC 791 (1981) で記述されています。

💻 拡張版 C シェルである tcsh は、Ken Greer が Carnegie Mellon University で TENEX 風のファイル名補完を実装しようとした取り組みに由来します。彼は 1975 年 9 月にこれに着手し、1981 年 12 月についに C シェルへマージしました。tcsh の "t" は、そのコマンド補完機能で Greer に着想を与えたオペレーティングシステム TENEX の "T" に由来します。

⚙️ 1982 年 3 月、米国国防総省は TCP/IP をすべての軍用コンピュータネットワーキングの標準として宣言しました。

🏢 The Mythical Man-Month は 1982 年に修正を加えて再版され、その時点で Brooks の法則はソフトウェアプロジェクト管理における基礎的な原則となっていました。

📊 Microsoft が開発した初期の表計算プログラム Microsoft Multiplan は 1982 年 8 月にリリースされました。

📜 Revision Control System (RCS) はバージョン管理システム (VCS) の初期の実装であり、複数のユーザーがプログラムコードや文書を開発・保守できるようにする一連の UNIX コマンドです。RCS は 1982 年に Purdue University の Walter F. Tichy によって最初にリリースされ、現在は GNU Project によって保守されています。

💻 John Warnock と Charles Geschke は Xerox PARC を去り、1982 年 12 月に

Adobe Systems を設立しました。彼らは、印刷ページをデバイスに依存しない方法で記述する手段を提供することを目標に、Warnock の Xerox における以前の Interpress 研究を踏まえたページ記述言語 PostScript の開発を始めました。

📜 TeX をゼロから書き直した新しいバージョンである TeX82 が 1982 年に公開されました。他の変更に加えて、元のハイフネーションアルゴリズムは Frank Liang が書いた新しいアルゴリズムに置き換えられました。

⚙ ARPANET の NCP から TCP/IP への移行は、新しいプロトコルが恒久的に有効化されたフラグデーである 1983 年 1 月 1 日に正式に完了しました。

📊 Lotus 1-2-3 は 1983 年 1 月 26 日に IBM PC 向けに正式にリリースされ、表計算・グラフ作成・データ管理の機能を x86 アセンブリで書かれた単一の高性能プログラムに統合することで、たちまちこのプラットフォームの「キラーアプリ」となりました。

💻 KornShell (ksh) は 1980 年代初頭に Bell Labs の David Korn が開発した Unix シェルであり、1983 年 7 月 14 日に USENIX で発表されました。Bourne シェルと後方互換性があり、Bell Labs のユーザーからの要望に触発された C シェルの多くの機能を備えています。

⚙ r-commands (rlogin、rsh、rcp を含みます) は 1983 年 8 月に 4.2BSD へ完全に統合されました。このリリースは成熟したプロセス間通信 (IPC) プリミティブを提供し、それによって r-commands は 1980 年代を通じて Unix ネットワーキングの事実上の標準となりました。

📜 GNU オペレーティングシステムの開発は、Richard Stallman が MIT Artificial Intelligence Laboratory に在籍していたときに始められました。これは GNU Project と呼ばれ、1983 年 9 月 27 日に公に発表されました。

⚙ Internet Engineering Task Force は 1983 年 11 月に DNS の当初の仕様を RFC 882 および RFC 883 として公開しました。

⚙ Simple Mail Transfer Protocol (SMTP) プロトコルは 1983 年に ARPANET 上で実装されました。

⚙ 現代の Unix の擬似端末インターフェースは 1983 年の Eighth Edition Unix の開発中に生まれ、4.2BSD (Berkeley Software Distribution) リリースに含まれたことで広く普及しました。

⚙ 1983 年、IEEE は最長 500 メートルの同軸ケーブル上で 10 Mbps の Ethernet を実現する 802.3 標準をドラフトとして公開しました (10BASE5)。

💻 1983 年春、Steve Jobs は Adobe を訪問し、Macintosh に対する PostScript の可能性に魅了されました。Apple と Adobe は 1983 年 12 月に PostScript のライセンス契約を締結し、Apple は Adobe に出資するとともに、開発中のレーザープリ

ンタに PostScript を採用することに同意しました。

⚙ 最初の Macintosh 128K は 1984 年 1 月 24 日に Steve Jobs によって 2,495 米ドルの価格で発表されました。これはマウス操作のグラフィカルユーザーインターフェースを備えた最初の商業的に成功したパーソナルコンピュータであり、Motorola 68000 プロセッサと 9 インチのモノクロディスプレイを使用していました。

⚙ 1984 年、UC Berkeley の 4 人の学生 Douglas Terry、Mark Painter、David Riggle、Songnian Zhou が、一般に BIND と呼ばれる Berkeley Internet Name Domain のための最初の Unix ネームサーバ実装を書きました。

🏢 X/Open group は、情報技術分野におけるオープンな標準を特定し推進するために、1984 年に複数のヨーロッパの UNIX システムメーカーによって設立されたコンソーシアムです。

🏢 Eliyahu M. Goldratt は 1984 年、ビジネス小説 The Goal で制約理論を広く一般の読者に紹介しました。

📜 TeX は 1984 年以来 GNU オペレーティングシステムの公式な組版パッケージとなっています。

🧠 画期的な著書 Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project は 1984 年に Bruce Buchanan と Edward Shortliffe によって出版されました。同書は MYCIN システムと、医学知識ベースを取り除いたエキスパートシステムシェルである EMYCIN (Essential MYCIN) を記録しており、EMYCIN は多様な領域にわたるルールベースのエキスパートシステムの構築を可能にし、1980 年代を通じたエキスパートシステムの急速な普及に寄与しました。当時は Fortune 500 企業の 3 分の 2 がこの技術を日常の業務活動に適用していました。

1985

💻 Apple と Adobe は 1985 年 1 月 23 日、Apple の年次株主総会で LaserWriter を発表しました。これは PostScript を搭載して出荷された最初のプリンタです。同時に Aldus が PageMaker を発表し、デスクトップパブリッシング革命の引き金となりました。

📜 GNU Emacs は、GNU Project の創設者 Richard Stallman が Unix オペレーティングシステム向けに開発された Emacs エディタをもとに作成したフリーソフトウェアのテキストエディタであり、C で書かれ、拡張言語として (同じく C で実装された) Emacs Lisp を提供します。最初の公開リリースはバージョン 13 で、1985 年 3 月 20 日に公開されました。

🏢 GNU 宣言は Richard Stallman による行動の呼びかけであり、GNU フリーコンピュータオペレーティングシステムを開発するという GNU Project の目標への参加と支援

を促すものです。1985 年 3 月に公開されました。

💻 Aldus PageMaker 1.0 は 1985 年 7 月に Macintosh 向けにネイティブな PostScript サポートを備えてリリースされ、最初に広く普及したデスクトップパブリッシングアプリケーションとなり、PostScript をプロフェッショナルな印刷における標準のページ記述言語として確固たるものにしました。PageMaker は 1986 年に Best New Use of a Computer 部門で Codie Award を受賞しました。

📊 Microsoft Excel は 1985 年 9 月 30 日に Macintosh 向けに最初にリリースされ、ユーザーがスプレッドシートの見た目を定義できるようにした最初の表計算ソフトでした。

📊 Lotus 1-2-3 Release 2.0 は 1985 年 9 月に登場し、マクロ、アドイン、拡張メモリ (EMS) のサポートを導入して、企業向け表計算市場における優位を確立しました。

⚙️ Microsoft Windows は 1983 年 11 月 10 日に Bill Gates によって MS-DOS 用のグラフィカルユーザーインターフェースとして初めて発表され、Windows 1.0 は 1985 年 11 月 20 日に正式にリリースされました。

📜 LaTeX は 1980 年代初頭に Leslie Lamport が Stanford Research Institute (SRI) に在籍していたときに作られました。もともとは彼自身の TeX マクロの必要を満たすために書かれましたが、他の人も使える汎用パッケージにする意図で設計されており、1984 年と 1985 年にバージョンがリリースされました。

🏢 Free Software Foundation は 1985 年に、フリーソフトウェアの開発を支援する非営利法人として設立されました。

📜 C はデンマークの計算機科学者 Bjarne Stroustrup が C プログラミング言語の拡張 (あるいは "C with Classes") として作った高水準の汎用プログラミング言語であり、1985 年に登場し、1998 年に ISO/IEC 14882:1998 として最初に標準化された後、C03、C11、C14、C++17 の各標準によって改訂されました。

📜 フリーで yacc 互換のパーサジェネレータである GNU Bison は、1985 年に Robert Corbett によって最初に書かれました。その後 Richard Stallman が GNU Project の一環として完全に yacc 互換にし、Carnegie Mellon University の Wilfred Hansen が複数文字の文字列リテラルなどの機能を追加しました。

1986

🏢 竹内弘高と野中郁次郎は、1986 年 1 月・2 月号の Harvard Business Review の論文 "The New New Product Development Game" において、製品開発の文脈で「スクラム」という用語を導入しました。

📜 GDB は GNU General Public License の下でリリースされたフリーソフトウェアで

あり、GNU Emacs が「そこそこ安定した」状態に達した後、1986 年に Richard Stallman が自身の GNU システムの一部として最初に書きました。

🌐 Standard Generalized Markup Language (SGML; ISO 8879:1986) は、文書のための汎用マークアップ言語を定義するための標準です。

📜 Vrije Universiteit Amsterdam の Dick Grune は、複数の開発者が同じファイルを並行して扱えるようにするため、1986 年 7 月に RCS をラップする一連のシェルスクリプトとして CVS の初期の形を開発しました。

🧠 機械学習において、バックプロパゲーション (誤差逆伝播法) は順伝播型人工ニューラルネットワークや微分可能なノードを持つその他のパラメータ化されたネットワークを訓練するために広く使われるアルゴリズムです。David E. Rumelhart らが 1986 年に発表した実験的分析はその普及に寄与し、多層パーセプトロンに関する研究の活発な時代を切り開く助けとなりました。

🧠 Deep Learning という用語は、1986 年に Rina Dechter によって機械学習コミュニティに導入されました。

📊 gnuplot は、関数、データ、データフィッティングの 2 次元および 3 次元のプロットを、主要なあらゆるコンピュータとオペレーティングシステム
(Linux、Unix、Microsoft Windows、macOS、FreeDOS など) で生成できるコマンドラインおよび GUI のプログラムであり、1986 年に最初にリリースされました。

📊 POSTGRES プロジェクトは 1986 年に Michael Stonebraker の主導のもと UC Berkeley で正式に開始され、オブジェクトリレーショナルの概念を導入することでリレーショナルモデルの限界を解決することを目指しました。

🧩 1986 年に公開された RFC 977 は、Usenet の記事を TCP/IP 上で配信するための Network News Transfer Protocol (NNTP) の仕様を提供しました。

1987

⚙️ Macintosh SE と Macintosh II は、1987 年 3 月 2 日に Los Angeles で開催された AppleWorld カンファレンスでともに発表されました。Macintosh II は Apple にとって最初のモジュラー型かつカラー対応の Macintosh であり、それ以前のモデルのオールインワン設計から脱却したものでした。

📜 GNU Compiler Collection (GCC) は GNU Project が開発する最適化コンパイラであり、さまざまなプログラミング言語、ハードウェアアーキテクチャ、オペレーティングシステムをサポートします。1987 年 3 月 22 日に Richard Stallman が最初にリリースした (MIT から FTP で入手可能でした) 当時は C プログラミング言語のみをサポートしていたため GNU C Compiler と名付けられていましたが、その後 GCC ははるかに多くの言語をサポートするよう発展しました。

🏢 "No Silver Bullet — Essence and Accidents of Software Engineering" は Fred Brooks によるエッセイであり、技術や管理手法における単一の進展が 10 年以内にソフトウェア生産性を桁違いに向上させることはないと論じ、問題に固有の「本質的」複雑性と、ツールやプロセスによって持ち込まれる「偶有的」複雑性を区別しました。1986 年の IFIP Tenth World Computing Conference で最初に発表され、1987 年 4 月に IEEE Computer に再掲されました。

📜 flex (Fast Lexical Analyzer Generator) は lex のフリーな再実装であり、スキャナ向けにより高速で効率的な C コードを生成します。1987 年頃に UC Berkeley の Vern Paxson が Van Jacobson から着想を得て書いたもので、商用 Unix 環境の外でのオリジナルの AT&T lex の普及を制限していたプロプライエタリなライセンス上の制約を取り除きました。

🎨 Tom DeMarco と Tim Lister による Peopleware: Productive Projects and Teams は 1987 年に初版が出版されました。

📜 1987 年 9 月、Oregon 州 Portland で開催された Functional Programming Languages and Computer Architecture (FPCA) 会議での会合において、同程度の表現力を持つ非正格な純粋関数型言語が 10 種類以上も生まれていたことを受けて、共通の言語を設計する委員会を設けるべきだという強い合意が形成されました。委員会はこの言語を、論理学者 Haskell B. Curry にちなんで Haskell と名付けました。

⚙ 1987 年 12 月 9 日にリリースされた Windows 2.0 は、重なり合うウィンドウ、デスクトップアイコン、キーボードショートカットを導入し、前バージョンからユーザー体験を大きく向上させました。

📜 Perl は高水準・汎用・インタプリタ型・動的なプログラミング言語のファミリーであり、レポート処理を容易にするための汎用 Unix スクリプト言語として 1987 年に Larry Wall によって開発され、1987 年 12 月 18 日に登場しました。

📊 Excel の最初の Windows 版は 1987 年 11 月 19 日にリリースされ、Macintosh 版と揃えるためバージョン 2.0 という番号が付けられ、Windows のランタイム版が同梱されていました。

📜 Self はプロトタイプの概念に基づくオブジェクト指向プログラミング言語であり、1986 年に Xerox PARC に在籍していた David Ungar と Randall Smith によって主に設計されました。動的型付け、ジャストインタイムコンパイル (JIT)、プロトタイプベースのオブジェクトへのアプローチを備えた Smalltalk の方言として始まり、1987 年に登場しました。

📜 Caml (Categorical Abstract Machine Language) の最初の実装は、Categorical Abstract Machine のコンパイル手法に基づいて 1987 年に INRIA の Ascánder Suárez によって作られました。

🧮 SQL は 1986 年に ANSI によって SQL-86 として、また 1987 年に ISO によって標準として採用されました。

1988

🎨 認知負荷理論は、認知負荷を内在的負荷・外在的負荷・学習関連負荷に区別する教授理論であり、教育心理学者の John Sweller によって構築されました。彼は 1988 年 4 月に Cognitive Science 誌で "Cognitive Load During Problem Solving: Effects on Learning" を発表し、手段-目標分析による問題解決方略がスキーマ獲得を妨げる認知的負担を課しうることを示しました。

🧮 Wolfram Mathematica は、機械学習、統計、記号計算、データ操作、ネットワーク分析、時系列分析、NLP、最適化、プロット、アルゴリズム実装、ユーザーインターフェース作成、他言語のプログラムとのインターフェースなど、技術計算のための組み込みライブラリを備えたソフトウェアシステムであり、1988 年 6 月 23 日に最初にリリースされました。

🧩 Xerox の Joe Becker は 1988 年 8 月、暫定的に "Unicode" と名付けられた国際的・多言語対応のテキスト文字符号化システムのドラフト提案を Unicode 88 と題した文書で発表し、Apple の Lee Collins および Mark Davis と共に開発した 16 ビットの文字方式を提案しました。

🧩 テキストベースのチャットシステムである Internet Relay Chat (IRC) は、1988 年 8 月にフィンランドの University of Oulu で Jarkko Oikarinen によって作られ、当初は OuluBox という BBS 上の MUT (MultiUser Talk) というプログラムを置き換えることを意図していました。

🐛 1988 年 11 月 2 日の Morris ワーム (あるいは Internet ワーム) は、インターネット経由で拡散した最も古いコンピュータワームの 1 つであり、主要メディアから大きな注目を集めた最初のものです。

🔐 X.509 は公開鍵証明書の形式を定める International Telecommunication Union (ITU) の標準であり、1988 年 11 月 25 日に最初に公開されました。

📜 GNU Make (略して gmake) は Linux と macOS における Make の標準実装であり、条件分岐、makefile のルール内でシェルスクリプトを不要にする組み込み関数、変数を操作する機能など、オリジナルの Make に対するいくつかの拡張を提供します。1988 年に最初にリリースされました。

🧮 GNU Octave は、1988 年に University of Texas at Austin の John W. Eaton によって数値計算のための高水準言語として構想され、もともとは化学反応器設計の講義の副教材として意図されていました。

📜 AWK は 1985 年から 1988 年にかけて大幅に改訂・拡張され、その成果として

Paul Rubin、Jay Fenlason、Richard Stallman が書いた GNU AWK の実装が 1988 年にリリースされました。

🏢 Open Software Foundation (OSF) は、Unix オペレーティングシステム実装のためのオープンな標準を作ることを目的に 1988 年に結成された非営利の業界コンソーシアムです。

📜 Portable Operating System Interface (POSIX) は、オペレーティングシステム間の互換性を維持するために IEEE Computer Society が策定した標準のファミリーであり、システムレベルとユーザーレベルの両方のアプリケーションプログラミングインターフェース (API) を定義しています。1988 年から開発が始まりました。

📊 データウェアハウジングの概念は 1980 年代後半にさかのぼり、IBM の研究者 Barry Devlin と Paul Murphy が「ビジネスデータウェアハウス」を考案し、1988 年に論文 "An architecture for a business and information system" を発表してこの用語をより広い読者に紹介しました。

🎨 The Design of Everyday Things は認知科学者 Donald Norman による書籍であり、もともと 1988 年に The Psychology of Everyday Things (POET) という題で出版されました。認知心理学に根ざしたデザイン原則を説明し、人間のニーズと認知に基づくユーザー中心設計を提唱するもので、2013 年には改訂版がリリースされました。

💻 オリジナルの KornShell である ksh88 の機能は POSIX.2 標準の基礎として使われました。1988 年にちなんで名付けられた ksh88 は、AIX や他のいくつかの商用 Unix システムでデフォルトのシェルとなりました。

1989

🔐 Kerberos はチケットを用いて動作するコンピュータネットワーク認証プロトコルであり、安全でないネットワーク上で通信するノードが互いに自身の身元を安全に証明できるようにします。Project Athena が提供するネットワークサービスを保護するために、1988 年に Massachusetts Institute of Technology (MIT) で開発されました。最初の公開版である Kerberos version 4 は 1989 年 1 月 24 日にリリースされました。

📊 Microsoft SQL Server は Microsoft が開発したプロプライエタリなリレーショナルデータベース管理システムであり、他のソフトウェアアプリケーションからの要求に応じてデータを保存・取得することを主な目的とするデータベースサーバとして機能します。最初のリリースは 1989 年 4 月 24 日です。

📜 Brian Berliner は 1989 年 4 月に CVS を C で再設計・再実装し、Grune のオリジナルのシェルスクリプト実装に比べて性能と信頼性を大幅に向上させました。

💻 Bash は、Bourne シェルのフリーソフトウェアによる置き換えとして Brian Fox が GNU Project のために書いた Unix シェルおよびコマンド言語であり、1989 年 6

月 8 日に最初にリリースされ、ほとんどの Linux ディストリビューションでデフォルトのログインシェルとして使われてきました。

📊 Lotus 1-2-3 Release 3.0 は 1989 年 6 月に C 言語による大規模な書き直しとしてリリースされ、単一ファイル内に複数のワークシートを持つ「3D スプレッドシート」を導入しましたが、動作には 80286 ベースのより新しいハードウェアが必要でした。

📊 POSTGRES Version 1 は 1989 年 6 月に少数の外部ユーザーに向けてリリースされ、Berkeley で開発された拡張可能なデータベースシステムの最初の公開実装となりました。

⚙ NeXTSTEP 1.0 は、Steve Jobs が 1985 年に Apple を去った後に設立した NeXT Computer が開発したオブジェクト指向のマルチタスクオペレーティングシステムであり、1989 年 9 月 18 日にリリースされました。Mach カーネルと BSD の上に構築されており、後に Mac OS X および現代の macOS の基盤となりました。

🌐 Hypertext Transfer Protocol (HTTP) は、分散協調型のハイパーメディア情報システムのための、インターネットプロトコルスイートモデルにおけるアプリケーション層のプロトコルです。HTTP の開発は 1989 年に CERN の Tim Berners-Lee によって始められ、0.9 と名付けられた最初の HTTP プロトコルバージョンを用いるクライアントとサーバの振る舞いを記述した簡潔な文書にまとめられました。

タイムライン - 1990-99

1990

📜 Haskell の最初の言語定義である Haskell version 1.0 Report が、Paul Hudak と Philip Wadler の編集により 1990 年 4 月 1 日に公開されました。同時に、誰でも参加できる Haskell メーリングリストが開設されました。

⚙ 1990 年 5 月 22 日に発売された Windows 3.0 は、完全に再設計されたユーザーインターフェースと 386 プロセッサ向けに改善されたメモリ管理を特徴とし、広範な商業的成功を収めた最初の Windows となりました。

📜 groff (GNU troff とも呼ばれます) は、書式設定コマンドを混在させたプレーンテキストを与えると整形された出力を生成する組版システムで、バージョン 0.3.1 が 1990 年 6 月に最初にリリースされました。

📜 CVS (Concurrent Versions System) のバージョン 1.0 は 1990 年 11 月 19 日に公開され、1990 年代を通じて主流のバージョン管理システムの一つとなりました。

💻 Zsh は、対話的なログインシェルとしても、シェルスクリプトのためのコマンドインタプリタとしても使用できる Unix シェルです。Paul Falstad は Princeton University の学生だった 1990 年に最初のバージョンを書き、1990 年 12 月 14 日に

alt.sources という Usenet ニュースグループへ 8 つに分けて投稿しました。この名前は Princeton のティーチングアシスタントであった Zhong Shao に由来し、Falstad は彼のログイン名である "zsh" がシェルの名前にふさわしいと考えました。バージョン 1.0 の時点での目標は、ksh と tcsh の中間、すなわち ksh のようによく設計されて論理的でありながら、tcsh のように人間のために作られた強力な「コマンドおよびプログラミング言語」となることでした。

🌐 CERN httpd (後に W3C httpd としても知られます) は、初期の、現在は開発が終了した Web サーバ (HTTP) デーモンで、もともと 1990 年以降に CERN で Tim Berners-Lee、Ari Luotonen、Henrik Frystyk Nielsen によって開発されました。C 言語で実装され、最初の Web サーバソフトウェアであり、1990 年 12 月 24 日に初版がリリースされました。

🏬 書籍 "The Machine That Changed the World" が 1990 年に出版され、トヨタ生産方式の手法を西洋に説明するために「リーン生産方式 (Lean Manufacturing)」という用語を生み出しました。

⚙ Microsoft は 1990 年に OLE 1.0 (Object Linking and Embedding) をリリースし、ソフトウェアコンポーネント同士が連携できるようにする、Windows 向けとして最初のドキュメントのリンクと埋め込みの技術を導入しました。

📜 flex は 1990 年に University of California の理事会 (Regents) によって BSD スタイルのライセンスの下で公開され、プロプライエタリな AT&T lex を置き換える、自由に利用できる最初の実装となって、オープンソースの Unix プロジェクト全体へ広く普及する道を開きました。

🌐 1990 年、Tim Berners-Lee のハイパーテキストに関する提案は、ハイパーリンクの参照先となるリソースを表す短い文字列としての URL という考え方を暗黙のうちに導入しました。

1991

🧩 Unicode Consortium は、Unicode 標準の開発、拡張、利用促進を行う非営利団体として、1991 年 1 月 3 日に California で法人化されました。

🌐 最初の Web ブラウザである WorldWideWeb は、1990 年に Tim Berners-Lee によって NeXT Computer 向けに開発され、1991 年 3 月に CERN の同僚たちに紹介されました。

⚙ Linux カーネルは、フリーかつオープンソースの、モノリシックでモジュール式、マルチタスクの Unix ライクなオペレーティングシステムカーネルです。1991 年 4 月、University of Helsinki の 21 歳の計算機科学の学生であった Linus Torvalds は、自身の i386 ベースの PC 向けに UNIX に触発された簡素なオペレーティングシステムの開発に着手し、Intel 80386 のアセンブリ言語によるタスクスイッチャと端末ドライバ

から始めました。

⚙ Linus Torvalds は 1991 年 8 月 25 日に comp.os.minix ニュースグループで自身のプロジェクトを発表し、それを「ただの趣味であり、gnu のように大規模でプロフェッショナルなものにはならない」と表現したことで知られています。

⚙ 1991 年 9 月 17 日、Torvalds は Linux カーネルのバージョン 0.01 を FUNET の FTP サーバ上で公開しました。これはまだ実行可能ではなく、コンパイルには MINIX が必要でした。

⚙ 1991 年 10 月 5 日、Torvalds は Linux の最初の「公式」バージョンであるバージョン 0.02 を発表しました。この時点でカーネルは Bash シェルと GCC を動作させることができました。

🧩 Unicode 1.0 は 1991 年 10 月に公開され、現代のすべての表記体系を包含することを意図した統一的な 16 ビット文字集合として、24 の文字体系から 7,161 文字を符号化しました。

📜 Vim (Vi IMproved の短縮形) は、フリーかつオープンソースの、画面ベースのテキストエディタプログラムであり、Bill Joy の vi を改良したクローンです。Vim の作者である Bram Moolenaar は Amiga 向けの Stevie エディタの移植版から Vim を派生させ、1991 年 11 月 2 日に最初のバージョンを一般に公開しました。

📜 Python は高水準のインタプリタ型汎用プログラミング言語であり、意味を持つインデントによってコードの可読性を重視する設計哲学を持ちます。Guido van Rossum は 1980 年代後半に ABC プログラミング言語の後継として Python の開発を始め、1991 年に Python 0.9.0 として最初にリリースしました。

🧠 1991 年、オートエンコーダが主成分分析 (PCA) の非線形な一般化として Kramer により初めて提案されました。

🔐 Pretty Good Privacy (PGP) は、データ通信に暗号によるプライバシーと認証を提供する暗号化プログラムです。PGP はテキスト、電子メール、ファイル、ディレクトリ、ディスクパーティション全体の署名、暗号化、復号に用いられ、電子メール通信のセキュリティを高めるために使われます。Phil Zimmermann が 1991 年に PGP を開発しました。

⚙ 1991 年、rlogin プロトコルが RFC 1282 で正式に定義され、Unix コミュニティで 10 年以上にわたり広く採用されてきた既存の慣行が文書化されました。

🎨 Crossing the Chasm は Geoffrey A. Moore によるマーケティングの書籍で、1991 年に初めて出版されました。Everett Rogers のイノベーションの普及理論を技術製品に特化して適用し、新しい技術が主流での成功を収めるために越えなければならない、アーリーアダプターとアーリーマジョリティの間にある「キャズム (chasm)」を説明しています。改

訂版は 1999 年と 2014 年に出版されました。

1992

⚙ 1992 年 4 月 6 日にリリースされた Windows 3.1 は、TrueType フォント、Windows レジストリを導入し、リアルモードのサポートを廃止して、より現代的なオペレーティングシステムアーキテクチャへの大きな一歩となりました。

⚙ Linux 0.12 は 1992 年 2 月 1 日に GNU General Public License (GPL) の下でリリースされ、当初の制限的なライセンスからの決定的な転換となりました。

🔐 MD5 は、MIT の Ronald Rivest 教授が設計した一連のメッセージダイジェストアルゴリズムの一つです。解析的な研究により MD5 の前身である MD4 が安全でない可能性が高いことが示されたため、Rivest は 1991 年に安全な代替として MD5 を設計し、1992 年 4 月に最初に公表しました。

🧩 UTF-8 は 1992 年 9 月 2 日、Ken Thompson と Rob Pike が New Jersey のダイナーでランチョンマットに走り書きして設計しました。二人はそれを実装し、Plan 9 from Bell Labs 全体で UTF-8 を使うよう更新して、Unicode のための自己同期型で後方互換性のある可変長符号化を生み出しました。

🌐 libwww (Library World Wide Web) は、Unix と Windows 向けのモジュール式のクライアントサイド Web API であり、libwww API のリファレンス実装です。1991 年から 1992 年にかけて、Tim Berners-Lee と CERN の Jean-François Groff という学生は、World Wide Web の可能性を示すために、NeXTstep オペレーティングシステム向けの元の WorldWideWeb ブラウザのさまざまなコンポーネントを移植性のある C 言語のコードで書き直しました。バージョン 1.0 は 1992 年 11 月に初めてリリースされました。

🏛 1992 年 11 月、IETF の「URI Working Group」が初めて会合を開きました。

📜 CTAN ("Comprehensive TeX Archive Network" の頭字語) は、TeX 関連の資料やソフトウェアをダウンロードできる権威ある場所です。1992 年にドイツの Rainer Schöpf と Joachim Schrod、英国の Sebastian Rahtz、米国の George Greenwade によって構築され、1993 年に Aston University で開催された EuroTeX カンファレンスで正式に発表されました。WEB サーバ自体は Gerd Neugebauer によって保守されています。

🏛 技術的負債という用語は、開発において安易な解決策を選んだことに起因する、システムを維持するためのコストを定性的に表現したものです。この用語は 1992 年に Ward Cunningham によって作られました。Metaphors We Live By を読んだ後、Ward は自分たちが取り組んでいた金融商品をリファクタリングする必要性を上司に説明するために、この負債のメタファーを考案しました。

⚙ オリジナルの Berkeley Packet Filter (BPF) は、1992 年に Steven McCanne と Van Jacobson による研究論文で紹介されました。カーネル内でネットワークパケットをフィルタリングする手段を提供し、カーネルとユーザー空間の間のデータのコピーを最小限に抑えました。

1993

🌐 National Center for Supercomputing Applications (NCSA) で開発された Web ブラウザの最初のアルファ／ベータ版である NCSA Mosaic 0.5 は、1993 年 1 月 23 日に Marc Andreessen によって X Window System 向けに発表されました。

📜 PDF (Portable Document Format) は、1993 年 1 月に Windows and OS/2 Conference において Adobe Systems によって公に紹介されました。Adobe は PDF 1.0 仕様を無償で利用可能にし、最初のリーダー／ライターとして Acrobat 1.0 をリリースして、あらゆるプラットフォームにわたる一貫した文書交換を実現することを目指しました。

⚙ NetBSD のソースコードリポジトリは 1993 年 3 月 21 日に開設されました。このプロジェクトは、386BSD に対するより開かれたコミュニティ主導の代替として、Chris Demetriou、Theo de Raadt、Adam Glass、Charles Hannum によって設立されました。

🌐 X Window System 向けの NCSA Mosaic 1.0 は 1993 年 4 月 21 日に正式にリリースされました。これは別ウィンドウではなく <img> タグを使って画像をテキストとインラインで表示した最初のブラウザであり、World Wide Web の普及に大きく寄与しました。

⚙ FreeBSD という名称は 1993 年 6 月 19 日に David Greenman によって正式に選ばれました。このプロジェクトは、同年初めに 386BSD の開発が停滞した後、「非公式の 386BSD パッチキット」を維持する取り組みとして始まりました。

📊 R は、University of Auckland で入門的な統計学を教えるためのプログラミング言語として Ross Ihaka 教授と Robert Gentleman 教授によって始められ、1993 年 8 月に初めて登場しました。

🌐 NCSA は 1993 年 9 月に Microsoft Windows と Apple Macintosh 向けの Mosaic の最初のバージョンをリリースし、技術者ではない家庭のユーザーにも Web を利用できるようにしました。

⚙ Debian (Debian GNU/Linux としても知られます) は、フリーかつオープンソースのソフトウェアで構成される Linux ディストリビューションで、コミュニティに支えられた Debian Project によって開発されています。Debian の最初のバージョン (0.01) は 1993 年 9 月 15 日にリリースされました。

⚙ FreeBSD 1.0-RELEASE は、4.3BSD-Net/2 と 386BSD に基づいて 1993 年 11 月 1 日に公開されました。この最初のリリースは、PC アーキテクチャ向けに堅牢な BSD ベースのオペレーティングシステムを提供しました。

🌐 Unix 向けの NCSA Mosaic 2.0 は 1993 年 11 月 10 日にリリースされました。これは HTML フォームのサポートを導入し、最初の動的で対話的な Web ページの作成を可能にしました。

⚙ Component Object Model (COM) は 1993 年に OLE 2.0 の基盤となるバイナリアーキテクチャとして正式に導入され、ソフトウェアコンポーネントの相互運用性に関する言語非依存の標準を確立しました。

⚙ 1993 年、Microsoft は OLE Automation を導入し、Microsoft Excel などのアプリケーションが Visual Basic のような外部のスクリプト言語に対して自身の機能を公開できるようにしました。

📊 Microsoft は 1993 年に Excel 5.0 をリリースし、これには Visual Basic for Applications (VBA) が含まれていました。VBA は作業を自動化しユーザー定義関数を提供する機能を追加したプログラミング言語です。

⚙ 1993 年、Microsoft は「New Technology」(NT) 系列の最初のバージョンである Windows NT 3.1 を発表しました。これはコンシューマ向けの DOS ベースの系列とは別に、ハイエンドのワークステーションとサーバ向けに設計された堅牢な 32 ビットカーネルを備えていました。

⚙ CFEngine は Mark Burgess によって書かれたオープンソースの構成管理システムで、大規模なコンピュータシステムの自動的な構成と保守を提供します。CFEngine プロジェクトは 1993 年、作者の Mark Burgess が理論物理学科の少数のワークステーション群の管理を自動化して自分の本来の仕事を片付けるための手段として始まりました。

🌐 HyperText Markup Language すなわち HTML は、Web ブラウザで表示されるように設計された文書のための標準的なマークアップ言語であり、1993 年に初めてリリースされました。

🌐 Charlie Jackson、Jonathan Gay、Michelle Alsip-Welsh が設立した FutureWave Software は、1993 年にペンコンピュータ向けのベクター描画アプリケーションである SmartSketch を公開しました。これは後に Adobe Flash となるものの最も初期の前身でした。

🌐 Common Gateway Interface (CGI) は、Web サーバが外部プログラムを実行できるようにするインターフェース仕様であり、通常はユーザーのリクエストを処理するために用いられます。1993 年、National Center for Supercomputing Applications (NCSA) のチームが、www-talk メーリングリスト上でコマンドラインの実行可能ファイルを呼び出すための仕様を書きました。

🌐 NCSA HTTPd は、初期の、現在は開発が終了した Web サーバで、もともと University of Illinois at Urbana–Champaign の NCSA で Robert McCool らによって開発され、1993 年に最初にリリースされた、最も初期の Web サーバの一つです。

📜 Lua は、主にアプリケーションへの組み込み用途向けに設計された軽量で高水準なマルチパラダイムのスクリプト言語で、1993 年にブラジルの Pontifical Catholic University of Rio de Janeiro のコンピュータグラフィックス技術グループである Tecgraf において、Roberto Ierusalimschy、Luiz Henrique de Figueiredo、Waldemar Celes によって作られました。その名前はポルトガル語で「月」を意味します。

🎨 Don Norman は、Apple Fellow として Apple に加わり User Experience Architect’s Office を設立した後の 1993 年に「ユーザーエクスペリエンス」という用語を生み出し、インダストリアルデザイン、グラフィックス、インターフェース、物理的な相互作用を包含するために「ヒューマンインターフェース」や「ユーザビリティ」よりも広い用語を求めていたと説明しています。

💻 1993 年にリリースされた KornShell の ksh93 は、連想配列や組み込みの浮動小数点演算といった機能に加え、動的に読み込まれる組み込みコマンドなど、ksh88 に対するさまざまな拡張を追加しました。

1994

📊 GNU Octave 1.0 は 1994 年 2 月 17 日に正式にリリースされ、GNU プロジェクトにおける科学技術計算と工学のための堅牢なツールとしての地位を確立しました。

⚙️ Linux のバージョン 0.95 は X Window System を動作させることができた最初のバージョンでした。Linux 1.0.0 は 1994 年 3 月 14 日にリリースされ、176,250 行のコードで構成されていました。これは本番環境での利用に適した最初のバージョンでした。

🏢 1994 年 6 月、IETF は Berners-Lee による、URL と URN の存在を認めた最初の Request for Comments を公開しました。

🏢 World Wide Web Consortium (W3C) は World Wide Web のための主要な国際標準化団体であり、1994 年 10 月 1 日に設立され、Tim Berners-Lee が率いています。

📜 Perl 5.000 は 1994 年 10 月 17 日にリリースされました。これはインタプリタのほぼ完全な書き直しであり、オブジェクト、リファレンス、レキシカル (my) 変数、モジュールなど、多くの新機能を言語に追加しました。

🌐 Netscape Navigator は、プロプライエタリな Web ブラウザであり Netscape 系列の最初のブラウザ (バージョン 1 から 4.08、および 9.x) で、Netscape Communications Corp の主力製品として、1990 年代に利用シェアの点で支配的な Web ブラウザとなり、1994 年 12 月 15 日に最初にリリースされました。

🧩 QR コードのシステムは 1994 年に日本の自動車製品企業である Denso Wave で発明されました。

📜 最初のユニットテストフレームワークである SUnit は、1994 年に Kent Beck によって Smalltalk 向けに作られました。これは setUp、テストメソッド、tearDown というテストフィクスチャのパターンを導入し、それが xUnit 系のフレームワーク全体の基礎となっています。

🎨 Jakob Nielsen と Rolf Molich は、ユーザーインターフェース設計のための一連のユーザビリティヒューリスティックを開発しました。初期の版は 1989 年から 1990 年に発表された論文に現れ、Nielsen は 1994 年にシステム状態の可視性、エラーの防止、一貫性といった原則を扱う洗練された 10 個のヒューリスティックを発表し、これはユーザビリティエンジニアリングにおけるヒューリスティック評価の広く使われる枠組みとなりました。

1995

🔐 Transport Layer Security (TLS) は、コンピュータネットワーク上で通信のセキュリティを提供するために設計された暗号プロトコルです。元々の SSL プロトコルは Netscape が開発し、1995 年から 1998 年まで Netscape Communications のチーフサイエンティストを務めた Taher Elgamal は「SSL の父」と評されてきました。SSL バージョン 2.0 は 1995 年 2 月にリリースされた後、数多くのセキュリティ上および使い勝手上の欠陥を含むことがすぐに判明しました。

📜 Wirth の法則は、ソフトウェアが低速化する速度はハードウェアが高速化する速度よりも速いという、コンピュータの性能に関する格言であり、1995 年 2 月に Computer 誌に掲載された記事 "A Plea for Lean Software" でこれを論じた Niklaus Wirth にちなんで名付けられました。

📜 Java は、実装依存性を可能な限り少なくすることを目指して設計された、高水準でクラスベースのオブジェクト指向プログラミング言語です。当初 Sun Microsystems の James Gosling によって開発され、Java SE は幅広い汎用 API を定義し、Java Language Specification と Java Virtual Machine Specification を含んでおり、1995 年 5 月 23 日に初めて登場しました。

🌐 Java Applet は Java 1.0a2 と HotJava ブラウザとともに導入され、1995 年 5 月 23 日の SunWorld で公開のデモンストレーションが行われました。これにより小さな Java プログラムを Web ページに埋め込み、ブラウザプラグイン (NPAPI) が提供するサンドボックス化された JVM 内で実行できるようになり、Web 上でのインタラクティブなコンテンツが可能になりました。

📶 MySQL はオープンソースのリレーショナルデータベース管理システム (RDBMS) であり、GNU General Public License の条件の下でフリーかつオープンソースのソフトウェアとして利用できるほか、さまざまなプロプライエタリライセンスの下でも利用で

き、1995 年 5 月 23 日に初版がリリースされました。

🌐 PHP は Web 開発に向けられた汎用スクリプト言語であり、元々 1994 年にデンマーク系カナダ人プログラマの Rasmus Lerdorf によって作られ、1995 年 6 月 8 日に初めて登場しました。

⚙ 1995 年 8 月 24 日にリリースされた Windows 95 は、象徴的なスタートメニュー、タスクバー、プラグアンドプレイ技術を導入するとともに、コンシューマ向け製品ラインを 32 ビットアーキテクチャへと移行させた、大規模なコンシューマ向けリリースでした。

⚙ Theo de Raadt は 1995 年 10 月 18 日に OpenBSD の CVS リポジトリを作成し、NetBSD のフォークとしてプロジェクトが正式に始まりました。このプロジェクトは de Raadt と NetBSD のコアチームとの意見の相違を受けて設立され、プロアクティブなセキュリティとコードの正しさに新たな焦点を置きました。

📜 CPAN は 1993 年に構想され、CTAN のモデルに基づいて 1995 年 10 月からオンラインで活動しており、散在していた Perl のアーカイブの構造を統一する場として始まりました。1995 年 10 月 26 日、Perl 言語と Perl モジュールのリポジトリとして Comprehensive Perl Archive Network (CPAN) が設立されました。

📜 Ruby は、複数のプログラミングパラダイムをサポートする、インタプリタ型で高水準の汎用プログラミング言語です。Ruby 0.95 の最初の公開リリースは 1995 年 12 月 21 日に日本国内のニュースグループで発表されました。

🌐 1995 年 12 月、Sun Microsystems と Netscape はプレスリリースで JavaScript を発表しました。最初の JavaScript エンジンは 1995 年に Netscape Navigator Web ブラウザ向けに Brendan Eich によって作られました。これは、Eich が生み出したばかりの言語のための初歩的なインタプリタでした。

⚙ IPv6 を標準化した最初の RFC は 1995 年の RFC 1883 であり、1995 年 12 月に IPv4 の後継としてこのプロトコルを正式に導入し、アドレス空間を 32 ビットから 128 ビットへ拡張しました。

📊 NumPy の前身である Numeric は、元々 Jim Hugunin が他の数名の開発者の貢献を得て作成したもので、1995 年に初版がリリースされました。

⚙ 100 Mbps で動作する Fast Ethernet は、1995 年に IEEE 802.3u 標準として導入されました。

🌐 Apache HTTP Server はフリーかつオープンソースのクロスプラットフォームな Web サーバソフトウェアであり、Apache License 2.0 の条件の下でリリースされ、Apache Software Foundation の支援の下でオープンな開発者コミュニティによって開発・保守されており、1995 年に初版がリリースされました。

🔐 SHA-1 (Secure Hash Algorithm 1) は、入力を受け取ってメッセージダイジェストとして知られる 160 ビット (20 バイト) のハッシュ値を生成するハッシュ関数であり、通常は 40 桁の 16 進数として表記されます。アメリカ国家安全保障局によって設計され、1995 年に初めて公開されたアメリカ連邦情報処理標準です。

🔐 SSH は、Telnet や、Berkeley の rsh および関連する rlogin、rexec といった保護されていないリモートシェルプロトコルの代替として設計されました。1995 年、フィンランドの Helsinki University of Technology の研究者であった Tatu Ylönen は、自身の大学のネットワークで発生したパスワード盗聴攻撃をきっかけに、このプロトコルの最初のバージョン (現在は SSH-1 と呼ばれます) を設計しました。

🏢 The Mythical Man-Month は 1995 年に 4 つの章を追加した 20 周年記念版として再出版され、Brooks の 1986 年のエッセイ "No Silver Bullet" を、彼の新たな解説 "'No Silver Bullet' Refined" とともに再録しました。

🏢 Ken Schwaber と Jeff Sutherland は 1995 年、Texas 州 Austin で開催された OOPSLA '95 カンファレンスで Scrum フレームワークを正式に発表しました。

1996

🧩 UTF-8 は 1996 年 1 月に RFC 2044 として初めて正式に標準化され、US-ASCII との互換性を保ちながら MIME やインターネットプロトコルで使用するために Unicode 文字を符号化する変換形式として定義されました。

🌐 Netscape Navigator 2.0 は 1996 年 3 月にリリースされ、NPAPI プラグインアーキテクチャを介した Java Applet のサポートを備えており、アプレットを広く利用できるものにして、Web 全体での急速な普及のきっかけとなりました。

⚙️ Debian の最初の安定版 (1.1) は 1996 年 6 月 17 日にリリースされました。

📊 PostgreSQL は Postgres としても知られ、拡張性と SQL への準拠を重視したフリーかつオープンソースのリレーショナルデータベース管理システム (RDBMS) です。1996 年 7 月 8 日にオープンソースプロジェクトとして初めてリリースされ、1996 年 10 月 22 日には SQL のサポートを反映してプロジェクトが正式に改名されるとともに、Web サイト PostgreSQL.org が開設されました。

🌐 Cascading Style Sheets (CSS) は、HTML や XML などのマークアップ言語で書かれた文書の体裁を記述するために使われるスタイルシート言語であり、1996 年 12 月 17 日に初版がリリースされました。

🌐 1996 年 12 月、Macromedia は FutureWave Software を買収し、FutureSplash Animator を Macromedia Flash 1.0 として再ブランド化しました。そして Flash Player を無料のブラウザプラグインとして配布することで急速にシェアを獲得し、Flash をインタラクティブな Web コンテンツの主要なプラットフォームとし

て確立しました。

🌐 Jakarta Servlet (旧称 Java Servlet) は、サーバの機能を拡張する Java ソフトウェアコンポーネントであり、最も一般的には Web アプリケーションのリクエストとレスポンスを処理するために使われます。Servlet API は 1996 年 5 月に第 1 回 JavaOne カンファレンスで初めて公に発表され、Servlet 1.0 は 1996 年 12 月に Java Servlet Development Kit の一部として出荷されました。最初の仕様は Sun Microsystems の Pavni Diwanji によって作成されました。

🏢 The Open Group は、「オープンでベンダー中立な技術標準と認証」を策定することによって「ビジネス目標の達成を可能にする」ことを目指す世界的なコンソーシアムです。1996 年に X/Open が Open Software Foundation と合併して設立されました。

📜 Objective Caml (OCaml) は 1996 年に初めてリリースされました。このとき Didier Rémy と Jérôme Vouillon が強力で静的に型安全なオブジェクトとクラスのシステムを Caml Special Light に統合し、この言語に "Objective" という接頭辞が与えられました。

💻 IntelliSense は Microsoft によるコード補完の実装であり、Visual Studio におけるものが最もよく知られています。コード補完と構文チェックについてすでに考案されていた多くの概念を基に、1996 年に Microsoft の主要な製品の機能として初めて導入されました。

🌐 Microsoft は 1996 年、Web および Internet Explorer との統合のために特別に最適化された COM ベースの技術群を再ブランド化・簡素化したものとして ActiveX を発表しました。

🌐 HTTP/1 は 1996 年に (バージョン 1.0 として) 確定し、完全に文書化されました。

🌐 1996 年、iframe タグが Internet Explorer によって導入されました。object 要素と同様に、コンテンツを非同期に読み込んだり取得したりできます。

🔐 より新しいバージョンの SSL/TLS は、1996 年にリリースされた SSL 3.0 に基づいています。

1997

📊 PostgreSQL 6.0 は 1997 年 1 月 29 日に、新しい名称での最初の正式リリースとしてリリースされ、ユニークインデックスと pg_dumpall ユーティリティを導入しました。

⚙️ Apple Computer は 1997 年 2 月 4 日に NeXT を 4 億 2700 万ドルで買収し、OPENSTEP オペレーティングシステムを獲得するとともに Steve Jobs を Apple

に呼び戻しました。NeXTSTEP から派生したコードベースは、後に Mac OS X の基盤となります。

💻 Visual Studio は Microsoft の統合開発環境 (IDE) であり、コンピュータプログラム、Web サイト、Web アプリ、Web サービス、モバイルアプリの開発に使われます。Microsoft は 1997 年 3 月 19 日に Visual Studio を初めてリリースし、多くのプログラミングツールを Visual Studio 97 として初めて 1 つにまとめました。

🏢 Eric S. Raymond によるエッセイ "The Cathedral and the Bazaar" は、1997 年 5 月 27 日に Linux Kongress で初めて発表されました。

🌐 ECMA-262 (ECMAScript) の初版は 1997 年 6 月に Ecma 総会で採択されました。

⚙️ 無線ローカルエリアネットワークのための IEEE 802.11 標準の最初のバージョンは 1997 年 6 月に承認され、1 Mbit/s および 2 Mbit/s のデータレートを提供しました。

🏢 統一モデリング言語 (UML) は、システムの設計を視覚化する標準的な方法を提供することを目的とした汎用のビジュアルモデリング言語です。UML 1.1 は 1997 年 8 月に OMG に提出され、1997 年 11 月に OMG によって採択されました。

📊 Comprehensive R Archive Network (CRAN) は R の中心的なソフトウェアリポジトリであり、R Foundation によって支えられています。CRAN は 1997 年に Kurt Hornik と Friedrich Leisch によって作られ、その名称は TeX の CTAN (1992 年リリース) や Perl の CPAN (1995 年リリース) といった初期の他のパッケージングシステムに倣ったものです。

🌐 Zeev Suraski と Andi Gutmans は 1997 年にパーサを書き直して PHP 3 の基礎を作り、この言語の名称を再帰的頭字語である PHP: Hypertext Preprocessor に変更しました。

🧠 リカレントニューラルネットワーク (RNN) は、ノード間の結合が循環を作ることができ、一部のノードの出力が同じノードへの後続の入力に影響を与えることを可能にする人工ニューラルネットワークの一種です。これにより時間的な動的挙動を示すことができます。Long short-term memory (LSTM) ネットワークは 1997 年に Hochreiter と Schmidhuber によって発明され、複数の応用分野で精度の記録を打ち立てました。

🌐 Google Search は Google が運営する検索エンジンで、検索クエリとの関連性に基づいて Web サイトを分析しランク付けするアルゴリズムを使用しており、1997 年に開始されました。

📜 JUnit は、Kent Beck と Erich Gamma が Zurich から Atlanta で開催される OOPSLA 1997 へ向かうフライトの中で、ペアプログラミングとテストファースト開発を

用いて Beck の SUnit フレームワークを Java に移植して作成したものです。JUnit は Java の主要なテストフレームワークとなり、自動化されたユニットテストを標準的なエンジニアリングの実践として確立しました。

1998

🏢 Netscape Communications Corporation は 1998 年 1 月 22 日、主力製品である Netscape Communicator のソースコードをフリーソフトウェアとして公開すると発表しました。

🌐 Extensible Markup Language (XML) は、任意のデータを保存、伝送、再構成するためのマークアップ言語およびファイル形式であり、人間にも機械にも読める形式で文書を符号化するための一連の規則を定義しています。1998 年 2 月 10 日に初めて公開されました。

🏢 Open Source Initiative (OSI) は、オープンソースソフトウェアを定義する一連の規則である Open Source Definition の管理者です。この組織は 1998 年 2 月下旬に Bruce Perens と Eric S. Raymond によって設立されました。二人は、Netscape がブラウザスイートをオープンソース化するという発表に触発されたグループの一員でした。

🏢 Netscape Communications Corporation は 1998 年 3 月 31 日、"The Cathedral and the Bazaar" に影響を受けて Netscape Communicator のソースコードを公開し、Mozilla プロジェクトを開始しました。

⚙ Advanced package tool (APT) は、Debian および Debian ベースの Linux ディストリビューションでソフトウェアのインストールと削除を扱うためにコアライブラリと連携して動作するフリーソフトウェアのユーザーインターフェースであり、1998 年 3 月 31 日に初版がリリースされました。

⚙ 1000 Mbps で動作する Gigabit Ethernet は、1998 年 6 月 25 日に IEEE 802.3z 標準として正式に承認されました。

⚙ iMac G3 は 1998 年 8 月 15 日に出荷が始まり、Jony Ive がデザインした半透明の Bondi Blue の筐体を特徴としていました。SCSI やフロッピードライブといった従来のポートを廃して USB を採用し、Steve Jobs の復帰後に Apple を倒産寸前の状態から救ったと広く評価されています。

🌐 Document Object Model (DOM) は、XML または HTML 文書を、各ノードが文書の一部を表すオブジェクトであるツリー構造として扱う、クロスプラットフォームかつ言語非依存のインターフェースです。1998 年 10 月 1 日に初めて公開されました。

🌐 Apache Tomcat は、Jakarta Servlet、Jakarta Pages および関連する Java Web 技術のフリーかつオープンソースの実装です。これは 1998 年 11 月に、Sun Microsystems のソフトウェアアーキテクトであった James Duncan Davidson による

サーブレットのリファレンス実装として始まりました。彼が「tomcat」という名前を選んだのは、この動物が自力で生きていけるものを象徴していたからです。

🌐 1998 年 11 月にリリースされた Servlet 2.1 は最初の公式なサーブレット仕様であり、RequestDispatcher インターフェースと、Web アプリケーション全体で情報を共有するための ServletContext を追加しました。

🔐 OpenSSL は、コンピュータネットワーク上の通信を盗聴から保護したり、通信相手を識別する必要があったりするアプリケーションのためのソフトウェアライブラリで、HTTPS Web サイトの大多数を含むインターネットサーバで広く使われています。OpenSSL プロジェクトは、インターネットで使われるコードのための無料の暗号化ツール一式を提供するために 1998 年に設立され、1998 年 12 月 23 日に初版がリリースされました。

🌐 LAMP (Linux, Apache, MySQL, PHP/Perl/Python) は、Web で最も人気のあるアプリケーションの多くで使われる、最も一般的なソフトウェアスタックの 1 つを表す頭字語です。LAMP という頭字語は、ドイツのコンピュータ雑誌 Computertechnik の 1998 年 12 月号で Michael Kunze が、フリーかつオープンソースのソフトウェアの組み合わせが「高価な商用パッケージの実現可能な代替となり得る」ことを示した際に作られました。

⚙️ 1998 年 12 月、RFC 1883 に取って代わる RFC 2460 の公開により、IPv6 は IETF の Draft Standard となりました。

📜 1998 年、まつもとゆきひろによって Ruby Application Archive が、Ruby の簡素な英語版ホームページとともに開設されました。

🏢 Halloween documents は、フリーソフトウェア、オープンソースソフトウェア、とりわけ Linux に関連する潜在的な戦略についての Microsoft の一連の社外秘メモと、それらのメモに対する一連のメディアの反応から成ります。流出した文書とそれへの反応は、いずれも 1998 年にオープンソースソフトウェアの提唱者である Eric S. Raymond によって公開されました。

🌐 1998 年、Microsoft Outlook Web Access チームは XMLHttpRequest スクリプティングオブジェクトの背後にある概念を開発しました。XMLHttpRequest (XHR) は、Web ブラウザと Web サーバの間でデータを転送するメソッドを持つオブジェクトの形をした API です。このオブジェクトはブラウザの JavaScript 環境によって提供されます。

🔐 AppArmor (「Application Armor」) は Linux カーネルのセキュリティモジュールであり、システム管理者がプログラムごとのプロファイルによってプログラムの権限を制限できるようにします。プロファイルでは、ネットワークアクセス、raw ソケットアクセス、一致するパス上のファイルを読み取り・書き込み・実行する許可といった権限を指定できます。1998 年に初版がリリースされました。

🌐 Perl 5 は 1990 年代後半に CGI スクリプト言語として広く普及しました。その一因は強力な正規表現と文字列解析の能力にありました。

⚙ 1998 年から 2004 年にかけて、コンピューティングプラットフォームとしての Linux の人気の高まりとともに、CFEngine の採用も拡大しました。

1999

🔐 TLS 1.0 は 1999 年 1 月に、SSL バージョン 3.0 のアップグレードとして RFC 2246 で初めて定義されました。

📜 Haskell 98 Report: Language and Libraries が、Simon Peyton Jones と John Hughes の編集により 1999 年 2 月に公開されました。これは安定性への表明であり、Haskell の特定のインスタンスを Haskell 98 と名付けて実装者が無期限にサポートすることを約束する一方、その自由奔放な後継たちは引き続き Haskell を名乗れるようにするものでした。Haskell 委員会は同じ年に解散しました。

⚙ Mac OS X Server 1.0 は 1999 年 3 月 16 日にリリースされ、NeXT から取得した NeXTSTEP のコードベース上に構築された Apple 初のオペレーティングシステムとなりました。クラシック Mac OS の GUI を改変したものを使用しており、コンシューマ向けの Mac OS X の先駆けとなりました。

🏢 Salesforce, Inc. は、California 州 San Francisco に本社を置くアメリカのクラウドベースのソフトウェア企業です。営業、カスタマーサービス、マーケティングオートメーション、e コマース、アナリティクス、人工知能、アプリケーション開発に焦点を当てたアプリケーションを提供しています。Salesforce は 1999 年 3 月 8 日、元 Oracle 幹部の Marc Benioff によって、Parker Harris、Dave Moellenhoff、Frank Dominguez とともに software-as-a-service (SaaS) 企業として設立されました。

🏢 Apache Software Foundation (ASF) は、多数のオープンソースソフトウェアプロジェクトを支援するアメリカの非営利法人です。ASF は Apache HTTP Server の開発者グループから形成され、1999 年 3 月 25 日に法人化されました。

🐛 Melissa ウイルスは、1999 年 3 月 26 日ごろに初めて出現した、急速に拡散するマクロウイルスでした。このウイルスは主に Microsoft Word と Outlook を使用するコンピュータを攻撃しました。

📜 HotSpot は、Java HotSpot Performance Engine としてリリースされた、デスクトップおよびサーバコンピュータ向けの Java 仮想マシンであり、Sun Microsystems によって開発され、現在は Oracle Corporation によって保守・配布されています。ジャストインタイムコンパイルや適応的最適化といった手法によって性能を向上させているのが特徴です。Java HotSpot Performance Engine は 1999 年 4 月 27 日にリリースされ、プログラミング言語 Smalltalk の実装である Strongtalk に由来する技術の上に構築されました。当初は Java 1.2 のアドオンとして提供されていた HotSpot は、Java

1.3 で Sun の標準の JVM となりました。

⚙ VMware Workstation Pro (2015 年の VMware Workstation 12 のリリースまでは VMware Workstation として知られていました) は、Windows および Linux オペレーティングシステムの x64 版で動作するホスト型ハイパーバイザであり、1999 年 5 月 15 日に初版がリリースされました。

⚙ RRDtool (ラウンドロビンデータベースツール) は、ネットワーク帯域幅、温度、CPU 負荷といった時系列データを扱うことを目的としており、1999 年 7 月 16 日に初版がリリースされました。

🌐 1999 年 8 月にリリースされた Servlet 2.2 は Java 2 Platform, Enterprise Edition (J2EE) の一部となり、標準的なパッケージングおよびデプロイの形式として Web アプリケーションアーカイブ (.war) ファイルを導入しました。

🔐 GnuPG は当初 Werner Koch によって開発され、最初の製品版 (1.0.0) は、最初の GnuPG リリース (バージョン 0.0.0) からほぼ 2 年後の 1999 年 9 月 7 日にリリースされました。

🏢 エクストリームプログラミング (XP) は、ソフトウェアの品質と、変化する顧客要求への対応力を向上させることを目的としたソフトウェア開発方法論です。Kent Beck は自身の仕事の中でエクストリームプログラミングを開発しました。彼はプロジェクトで用いていた開発方法論を洗練し始め、その方法論についての本 (Extreme Programming Explained、1999 年 10 月出版) を執筆しました。

🏢 The Pragmatic Programmer: From Journeyman to Master は、Andrew Hunt と David Thomas によって書かれ 1999 年 10 月に出版された、コンピュータプログラミングとソフトウェアエンジニアリングについての本です。

💻 GNU nano は、コマンドラインインターフェースを使用する Unix 系のコンピューティングシステムまたはオペレーティング環境向けのテキストエディタであり、1999 年 11 月 18 日に初版がリリースされました。

🔐 OpenBSD 2.6 は 1999 年 12 月 1 日にリリースされ、OpenSSH の最初のリリースを含んでいました。元々はプロプライエタリな SSH スイートのフリーな代替として開発された OpenSSH は、Secure Shell (SSH) プロトコルに基づいており、クライアント・サーバ型のアーキテクチャにおいて保護されていないネットワーク上で安全な通信路を提供します。

📜 Jakarta EE は、旧称を Java Platform, Enterprise Edition (Java EE) および Java 2 Platform, Enterprise Edition (J2EE) といい、分散コンピューティングや Web サービスといったエンタープライズ向け機能の仕様によって Java SE を拡張する一連の仕様であり、最初の仕様は 1999 年 12 月 17 日にリリースされました。

🎨 Peopleware: Productive Projects and Teams は 1999 年に第 2 版として改訂されました。

🏢 The Cathedral and the Bazaar という書籍は 1999 年に出版され、同年に Open Publication License v2.0 の下で公開されました。

🏢 Nikolai Bezroukov は 1999 年、Eric Raymond のオープンソースソフトウェアに関する見解について 2 編の批判的なエッセイを発表しました。

📊 Apache Lucene はフリーかつオープンソースの検索エンジンライブラリであり、元々 1999 年に Doug Cutting によって書かれました。Lucene という名前は Doug Cutting の妻のミドルネームであり、彼女の母方の祖母のファーストネームでもあります。

🌐 1999 年、James Duncan Davidson は Tomcat のオープンソース化に貢献し、Sun Microsystems から Apache Software Foundation への寄贈で重要な役割を果たしました。Tomcat はそこで Jakarta Project の一部となりました。Apache による最初のリリースである Tomcat 3.0.x は、Servlet 2.2 と JSP 1.1 の仕様を実装していました。

🌐 JavaServer Pages（JSP）は、HTML、XML、その他の文書型に基づいて動的に生成される Web ページをソフトウェア開発者が作成するのを助ける技術の集合です。1999 年に Sun Microsystems によってリリースされた JSP は PHP や ASP に似ていますが、Java プログラミング言語を使用します。

🌐 1999 年に VA Software によって設立された SourceForge は、フリーかつオープンソースのソフトウェア開発者がソフトウェア開発を制御・管理するための中央集約的な場所を初めて提供し、しかもそれを無償で行いました。

タイムライン - 2000-09

2000

🌐 Flash 5 は 2000 年にリリースされ、ECMAScript に基づく完全なスクリプト言語である ActionScript 1.0 を導入しました。これにより複雑なインタラクティブ性が実現できるようになり、Flash はリッチインターネットアプリケーション、ゲーム、ビデオプレイヤーのためのプラットフォームとなりました。

⚙ jail は FreeBSD の OS レベル仮想化の実装であり、システム管理者が FreeBSD 派生のコンピュータシステムを jail と呼ばれる複数の独立したミニシステムに分割できるようにする仕組みで、2000 年 3 月 14 日の FreeBSD バージョン 4.0 で初めて導入されました。jail は仮想化、セキュリティ、権限委譲の容易さという 3 つの主要な目標を掲げています。

🌐 2000 年 3 月、Jimmy Wales と Larry Sanger によって、フリーで査読を経た百

科事典として Nupedia プロジェクトが開始されました。これは Wikipedia の前身にあたるプロジェクトで、その成長の遅さが翌年の Wikipedia 誕生に直接つながりました。

🌐 2000 年 5 月 22 日、Zend Engine 1.0 を搭載した PHP 4 がリリースされました。

📶 SQLite は C プログラミング言語で書かれたデータベースエンジンで、2000 年 8 月 17 日に最初にリリースされました。単体のアプリケーションではなく、ソフトウェア開発者が自身のアプリに組み込むライブラリであり、そのため世界で最も広く使われているデータベースエンジンとなっています。

📜 Python 2.0 は 2000 年 10 月 16 日にリリースされ、循環参照を検出するガベージコレクタや Unicode のサポートなど、多くの重要な新機能を備えていました。

📜 Apache Subversion (SVN) はソフトウェアのバージョン管理・リビジョン管理システムです。CollabNet は 2000 年、CVS と同じように動作しながらそのバグを修正し、不足していた機能を追加したオープンソースのバージョン管理システムを作るためにこのプロジェクトを立ち上げ、2000 年 10 月 20 日に最初のリリースを行いました。

🔐 Security-Enhanced Linux (SELinux) は Linux カーネルのセキュリティモジュールで、強制アクセス制御 (MAC) を含むアクセス制御セキュリティポリシーをサポートする仕組みを提供します。2000 年 12 月 22 日に最初にリリースされました。

📜 C# (see sharp と発音) は、2000 年に Microsoft の Anders Hejlsberg によって設計された汎用のマルチパラダイムプログラミング言語です。その後、2002 年に Ecma (ECMA-334)、2003 年に ISO/IEC (ISO/IEC 23270) によって国際標準として承認されました。

📜 CMake はクロスプラットフォームのビルドシステムジェネレータで、Insight Segmentation and Registration Toolkit (ITK) のための統一されたビルド環境が必要とされたことを受けて開発され、2000 年に最初に実装されて 2001 年にさらに開発が進められました。

🏢 Linux Foundation (LF) は、Linux を標準化し、その成長を支え、商業的な採用を促進するために、Open Source Development Labs と Free Standards Group の合併によって 2000 年に設立された非営利の技術コンソーシアムです。

🌐 2000 年、Roy Fielding は Web サービスを設計するためのアーキテクチャ上のアプローチとして Representational State Transfer (REST) を提唱しました。REST はハイパーメディアに基づいて分散システムを構築するためのアーキテクチャスタイルです。

🏢 Microsoft は 2000 年 6 月の Forum 2000 カンファレンスで .NET イニシアチブを正式に発表し、COM のようなバイナリレベルのコンポーネントモデルが抱える複雑さに対処することを狙った新しい「マネージド」なソフトウェアプラットフォームを提示し

ました。

2001

⚙ 2001 年 1 月 4 日にリリースされた Linux バージョン 2.4.0 は、ISA Plug and Play、USB、PC Card のサポートを含んでいました。Linux 2.4 では Pentium 4 と Itanium、そして新しい 64 ビットの MIPS プロセッサのサポートが追加されました。

🏛 Creative Commons (CC) は、教育へのアクセスと、他者が合法的に利用・共有して次の創作につなげられる作品の範囲を広げることに取り組むアメリカの非営利組織であり国際的なネットワークで、2001 年 1 月 15 日に設立されました。

🌐 Wikipedia は世界中のボランティアによって作成・編集されるフリーのオンライン百科事典です。誰もが編集できる、包括的でアクセスしやすく信頼できる情報源を作るという原則に基づいて、2001 年 1 月 15 日に Jimmy Wales と Larry Sanger によって設立されました。Wikipedia は、その必要性に合わせて特別に作られたソフトウェアである MediaWiki 上で動作しています。

🏛 2001 年 2 月 11 日から 13 日にかけて、ソフトウェア業界の 17 人のリーダーが集まり、語り合い、スキーをし、くつろぎながら共通点を見いだそうとして、Manifesto for Agile Software Development を共同で発表しました。署名者には Scrum の共同開発者である Ken Schwaber と Jeff Sutherland も名を連ねています。

⚙ VMware ESXi (旧 ESX) は、仮想コンピュータを展開・提供するために VMware が開発したエンタープライズクラスの Type-1 ハイパーバイザで、2001 年 3 月 23 日に最初にリリースされました。

⚙ Mac OS X 10.0 Cheetah は 2001 年 3 月 24 日にリリースされ、NeXTSTEP を土台として構築された Apple の Unix ベースのオペレーティングシステムとして最初のデスクトップ版となりました。Aqua ユーザーインターフェースを導入し、クラシック Mac OS のルック&フィールに取って代わりました。

☁ CruiseControl は BSD スタイルのライセンスの下で配布される Java ベースの継続的ビルドフレームワークで、2001 年 3 月 30 日に、最初期のオープンソース継続的インテグレーションツールの一つとしてリリースされました。

📊 2001 年 4 月 13 日にリリースされた PostgreSQL 7.1 は、データの永続性とクラッシュリカバリを確保するうえで重要な機能である Write-Ahead Log (WAL) を導入しました。

📜 YAML は人間が読みやすいデータシリアライゼーション言語で、設定ファイルやデータの保存・転送に広く使われています。2001 年 5 月 11 日に最初にリリースされました。

📜 reStructuredText (RST) は、主に Python コミュニティで技術文書のために使われるテキストデータのファイル形式です。Docutils プロジェクトの一部であり、Java における Javadoc に似た文書化ツールを Python にもたらすために、2001 年 6 月 1 日に最初にリリースされました。

🌐 WebKit は Apple が開発したブラウザエンジンで、主に同社の Safari Web ブラウザとすべての iOS 向け Web ブラウザで使われており、2001 年 6 月 25 日に Apple 社内で Don Melton が KHTML と KJS のフォークとして開始しました。

🐛 Code Red は 2001 年 7 月 15 日にインターネット上で観測されたコンピュータワームで、Microsoft の IIS Web サーバを実行しているコンピュータを攻撃し、企業ネットワークを標的として成功した最初の大規模な複合脅威型攻撃となりました。

🐛 Nimda ウイルスは、急速に拡散したファイル感染型の悪意あるコンピュータワームで、Code Red など過去の流行による経済的被害を上回りました。最初のアドバイザリは 2001 年 9 月 18 日に公開されました。

📊 2001 年 9 月、Lucene は Apache Software Foundation のオープンソース Java 製品群である Jakarta ファミリーに加わりました。

🌐 2001 年 9 月にリリースされた Tomcat 4.0 は、Servlet 2.3 と JSP 1.2 を実装した完全な書き直しであり、Catalina (サーブレットコンテナ)、Coyote (HTTP コネクタ)、Jasper (JSP エンジン) という、その後長く続くアーキテクチャを導入しました。

⚙️ 2001 年 10 月 25 日に発売された Windows XP は、コンシューマ向けの製品ラインを安定した NT カーネルへ移行させることで、Microsoft のコンシューマ向けとビジネス向けのオペレーティングシステムをついに統合しました。再設計された「Luna」インターフェースを特徴とし、その後 10 年以上にわたってデスクトップ市場を支配することになります。

💻 Eclipse は、基本のワークスペースと拡張可能なプラグインシステムを備えた統合開発環境 (IDE) で、Smalltalk ベースの VisualAge シリーズの IDE に着想を得ています。Eclipse 1.0 は 2001 年 11 月 29 日にリリースされ、2016 年まで最も人気のある Java IDE でした。

📊 SciPy は科学技術計算のためのフリーかつオープンソースの Python ライブラリで、Travis Oliphant、Eric Jones、Pearu Peterson がそれぞれ書いていたコードを統合し、単一のパッケージ名の下にまとめた 2001 年頃に生まれました。

⚙️ VMware Server (旧 VMware GSX Server) は VMware, Inc. が開発した、提供が終了している無償の仮想化ソフトウェアサーバスイートで、ESX 1.0 (Type 1 ハイパーバイザ) と GSX 1.0 (Type 2 ハイパーバイザ) がいずれも 2001 年に登場しました。

📊 IPython (Interactive Python) は、もともと Python 向けに開発された対話的コンピ

ューティングのためのコマンドシェルで、イントロスペクション、リッチメディア、シェル構文、タブ補完、履歴といった機能を提供します。2001 年に最初にリリースされました。

🔐 SHA-2 (Secure Hash Algorithm 2) はアメリカ国家安全保障局 (NSA) が設計した暗号学的ハッシュ関数群で、2001 年に初めて公開されました。専用のブロック暗号から作られる Davies–Meyer 構造に基づく一方向圧縮関数を、Merkle–Damgård 構成によって組み立てたものです。

🔐 Advanced Encryption Standard (AES) は、元の名称である Rijndael としても知られ、アメリカ国立標準技術研究所 (NIST) が 2001 年に策定した電子データの暗号化に関する仕様です。

🌐 2001 年、最初の Java Specification Request (JSR) は javax.servlet.ui という名前のパッケージを開発することを提案しました。2001 年 6 月には JavaWorld が、Amy Fowler のチームによる「JavaServer Faces API」(「Moonwalk」とも呼ばれます) の設計を、「Web ベースのユーザーインターフェースを作成するためのアプリケーションフレームワーク」として報じました。

🏢 Toyota は 2001 年、社内の経営上の価値観を「The Toyota Way」として明文化しました。これは継続的改善 (Kaizen) と人間の尊重という 2 つの柱から成ります。

2002

🌐 ASP.NET は動的な Web ページを構築するためのサーバサイド Web アプリケーションフレームワークで、2002 年 1 月 5 日に .NET Framework 1.0 の一部としてリリースされました。Microsoft の Active Server Pages (ASP) 技術の後継にあたります。

🏢 .NET Framework 1.0 は 2002 年 2 月 13 日、Visual Studio .NET とともに正式にリリースされ、Common Language Runtime (CLR) と C# プログラミング言語を導入しました。

⚙ Arch Linux は独立して開発されている x86-64 向けの汎用 Linux ディストリビューションで、ローリングリリースモデルを採用することでほとんどのソフトウェアについて最新の安定版を提供することを目指しています。Judd Vinet によって始められ、2002 年 3 月 11 日に最初にリリースされました。

⚙ Gentoo Linux は Portage パッケージ管理システムを用いて構築される Linux ディストリビューションで、バージョン 1.0 が 2002 年 3 月 31 日にリリースされ、2004 年には非営利の Gentoo Foundation が設立されてすべての著作権と商標が移管されました。

🌐 Mozilla プロジェクトは nsIXMLHttpRequest というインターフェースを Gecko レイアウトエンジンに開発・実装し、それを XMLHttpRequest という JavaScript オブ

ジェクトとして公開するラッパーを作りました。XMLHttpRequest オブジェクトは Gecko バージョン 0.6 (2000 年 12 月 6 日リリース) の時点ですでに利用できましたが、完全に機能するようになったのは Gecko 1.0 (2002 年 6 月 5 日リリース) からでした。

🌐 Mozilla Firefox は Mozilla Foundation が開発するフリーかつオープンソースの Web ブラウザで、Gecko レンダリングエンジンを使って Web ページを表示します。2002 年 9 月 23 日に最初にリリースされました。

📜 Spring Framework は Java プラットフォーム向けのアプリケーションフレームワークであり制御の反転 (IoC) コンテナで、2002 年 10 月 1 日に最初にリリースされました。中核となる機能はあらゆる Java アプリケーションで利用でき、Java EE の上に Web アプリケーションを構築するための拡張も備えています。

🧠 Torch は Lua プログラミング言語をベースとしたオープンソースの機械学習ライブラリ兼科学計算フレームワークで、2002 年 10 月に最初にリリースされました。

📜 Kent Beck による "Test-Driven Development: By Example" は 2002 年 11 月に Addison-Wesley から出版され、レッド・グリーン・リファクタのサイクルを広めるとともに、テストを通すための本番コードを書く前に自動テストを書くという規律としてテスト駆動開発を提示しました。

⚙️ Linux の名前空間は、2002 年にカーネル 2.4.19 でマウント名前空間に関する作業から始まりました。それ以外の名前空間は 2006 年以降に追加され、その後も追加が続いています。

📜 AsciiDoc は人間が読める文書形式で、意味的には DocBook XML と同等でありながらプレーンテキストのマークアップ記法を用います。2002 年に Stuart Rackham が、プレーンテキストファイルを一般的な出版用文書形式へ変換する Python ベースのツール ('asciidoc' と 'a2x') とともに作成しました。

🌐 JSON は JavaScript から派生した言語非依存のデータ形式で、標準的なデータ交換形式として確立するために 2002 年に JSON.org の Web サイトが開設されました。

📊 Lotus 1-2-3 の最後のメジャーリリースとなる Release 9.8 は、2002 年に Lotus SmartSuite オフィスパッケージの一部として発売されました。

🏢 Ken Schwaber は 2002 年、世界規模で Scrum の研修と認定を提供するために Scrum Alliance を設立しました。

🏢 Kent Beck は 2002 年 11 月 8 日に "Test Driven Development: By Example" を出版し、失敗するテストを書き、最小限のコードでそれを通し、その後リファクタリングするというレッド・グリーン・リファクタのサイクルとともに、TDD を独立した実践として体系化しました。

2003

📊 Tableau Software, LLC はビジネスインテリジェンスに重点を置くアメリカの対話的データ可視化ソフトウェア企業で、2003 年 1 月に Pat Hanrahan、Christian Chabot、Chris Stolte によって正式に設立され、2004 年に本社を Washington 州 Seattle へ移しました。

🌐 その後ずっと Wikipedia を運営している非営利組織である Wikimedia Foundation は、2003 年 6 月 20 日に Florida 州 St. Petersburg で法人化されました。

💻 Monad (Microsoft Shell、MSH) が初めて公開の場で実演されたのは、2003 年 9 月の Professional Developers Conference (PDC) でした。

⚙️ Android, Inc. は「所有者の位置や好みをよりよく認識する、より賢いモバイルデバイス」を開発するために、2003 年 10 月に California 州 Palo Alto で Andy Rubin、Rich Miner、Nick Sears、Chris White によって設立されました。

⚙️ Linux バージョン 2.6.0 は 2003 年 12 月 17 日にリリースされました。2.6.x の開発では、シリーズを通じて新機能を取り込んでいく方向へとさらに舵が切られました。

📜 Python Package Index (PyPI) は Cheese Shop とも呼ばれる Python 公式のサードパーティソフトウェアリポジトリで、2003 年に開設されました。そのメタデータ標準は 2001 年 3 月に PEP 241 で定義され、包括的なカタログの提案は 2002 年 11 月に確定しました。

📜 ドメイン駆動設計 (DDD) は、ドメインエキスパートからの入力に従ってソフトウェアをドメインに合わせてモデル化することに重点を置く、主要なソフトウェア設計手法です。この用語は Eric Evans が 2003 年に出版した同名の著書で作りました。

⚙️ Google Borg は Google が使用しているクラスタマネージャです。Docker や Kubernetes といった同様のアプローチが広く使われるきっかけとなりました。Google が 2015 年に公開した研究論文によれば、Borg は 2003 年に開発されました。

⚙️ Xen は、複数のコンピュータオペレーティングシステムを同じハードウェア上で同時に実行できるようにするサービスを提供する Type-1 ハイパーバイザで、Ian Pratt と博士課程の学生であった Keir Fraser が率いた University of Cambridge の研究プロジェクトとして生まれました。最初の公開リリースは 2003 年、v1.0 は 2004 年で、現在は Linux Foundation が開発を進めています。

📊 Matplotlib は Python プログラミング言語と NumPy のための描画ライブラリで、Tkinter、wxPython、Qt、GTK といった GUI ツールキットを使ってアプリケーションにプロットを埋め込むためのオブジェクト指向 API を提供します。2003 年に最初にリリースされました。

🔐 CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans

Apart) は、利用者が人間かどうかを判定してボットによる攻撃やスパムを防ぐためにコンピューティングで用いられるチャレンジ・レスポンス型のチューリングテストで、この用語は 2003 年に Luis von Ahn、Manuel Blum、Nicholas J. Hopper、John Langford によって作られました。

🧩 RFC 3629 は 2003 年 11 月に発行され、UTF-8 仕様を更新して U+0000..U+10FFFF の範囲に制限し、UTF-8 を標準的なインターネットプロトコル要素として確立しました。

🏢 Dan North は 2003 年後半、テストの語彙を取り除いて振る舞いの検証を中心とした語彙に置き換えた JUnit の代替である JBehave を書く中で、ビヘイビア駆動開発 (BDD) を提唱しました。さらに Chris Matts とともに、「Given/When/Then」というシナリオの語彙を通じてこのアプローチを要求分析へと広げました。

📜 2003 年に GameDev.net が実施した調査では、Lua がゲームプログラミングにおいて最も人気のあるスクリプト言語であることが示されました。

2004

📜 Scala は静的に強く型付けされた汎用プログラミング言語で、オブジェクト指向と関数型の両方のプログラミングパラダイムをサポートします。2004 年 1 月 20 日に最初にリリースされました。

🌐 JavaServer Faces 1.0 は 2004 年 3 月 11 日に、コンポーネントベースの Web アプリケーションフレームワークの最初の仕様としてリリースされました。

📜 RubyGems は Ruby プログラミング言語のパッケージマネージャで、Ruby のプログラムやライブラリを配布するための標準的な形式を提供します。開発は 2003 年 11 月に始まり、2004 年 3 月 14 日 (円周率の日) に公開されました。

🌐 Google は 2004 年 4 月 1 日、限定ベータ版として Gmail を公開し、当時の競合他社を大きく上回る 1 ギガバイトの保存容量を提供しました。

📜 2004 年 4 月、Windows Installer XML (WiX) は Microsoft のプロジェクトとして初めて、オープンソースライセンスである Common Public License の下で公開されました。当初は SourceForge でホストされており、外部でホストされた最初の Microsoft プロジェクトでもありました。

⚙ 2004 年 4 月、南アフリカの起業家 Mark Shuttleworth は 12 人ほどの Debian 開発者を London の自宅に招き、そこで彼らは後に Ubuntu となるものの特徴をブレインストーミングによって描き出しました。プロジェクトの資金を賄うため、Shuttleworth は Thawte を Verisign へ売却して得た資産で開発者を雇用すべく Canonical Ltd. を設立しました。

🌐 Web Hypertext Application Technology Working Group (WHATWG) は 2004 年 6 月 4 日に発表されました。HTML を発展させるための共同提案が W3C のワークショップで却下されたことを受けて、Apple、Mozilla、Opera が設立したものです。WHATWG は独自に新しい HTML 仕様の策定を始め、それが最終的に HTML5 となりました。

🌐 2004 年 7 月 1 日、新しい Zend Engine II を搭載した PHP 5 がリリースされました。PHP 5 には、オブジェクト指向プログラミングのサポート改善、PHP Data Objects (PDO) 拡張、数多くの性能改善といった新機能が含まれていました。

📜 Maven は主に Java プロジェクトで使われるビルド自動化ツールです。Jason van Zyl が作り、2002 年に Apache Turbine のサブプロジェクトとして始まり、2003 年に Apache Software Foundation のトップレベルプロジェクトとして受け入れられました。Maven 1.0 は 2004 年 7 月 13 日にリリースされました。

🌐 Ruby on Rails 0.5.0 は、プロジェクト管理ツール Basecamp から切り出された最初のオープンソースリリースで、2004 年 7 月に David Heinemeier Hansson によって公開されました。

🌐 Nginx は Web サーバであり、リバースプロキシ、ロードバランサ、HTTP キャッシュとしても使えます。C10k 問題を解決するために Igor Sysoev が作り、2004 年 10 月 4 日に最初に公開され、2008 年 9 月までに Rambler ポータルで 1 日あたり 5 億リクエストを処理するようになっていました。

⚙ Ubuntu 4.10 (Warty Warthog) は 2004 年 10 月 20 日に Ubuntu の最初のリリースとして公開されました。Debian を基盤としており、6 か月ごとに新しいリリースを出し、その後 18 か月のサポートを提供する計画が立てられていました。

⚙ Unionfs は Linux、FreeBSD、NetBSD 向けのファイルシステムサービスで、他のファイルシステムに対するユニオンマウントを実装し、別々のファイルシステムのファイルやディレクトリを単一の一貫したファイルシステムとして透過的に重ね合わせることができます。バージョン 1.0.2 は 2004 年 11 月 9 日にリリースされました。

🌐 Firefox のバージョン 1.0 は 2004 年 11 月 9 日にリリースされました。その後、2005 年 11 月にバージョン 1.5、2006 年 10 月にバージョン 2.0、2008 年 6 月にバージョン 3.0、2009 年 6 月にバージョン 3.5 が続きました。

📜 Markdown は、プレーンテキストエディタを使って書式付きテキストを作成するための軽量マークアップ言語です。John Gruber と Aaron Swartz は 2004 年、ソースコードの形のままでも読みやすいことを意図したマークアップ言語として Markdown を作りました。

📊 MapReduce は、クラスタ上の並列分散アルゴリズムによって大規模なデータセットを処理・生成するためのプログラミングモデルと、それに対応する実装で、2004 年に Google によって発表されました。

🧠 2004 年、K. S. Oh と K. Jung は、標準的なニューラルネットワークが GPU によって大幅に高速化できることを実証し、CPU による実装の 20 倍の速さを達成しました。2005 年に続いた論文は、機械学習における GPGPU の価値をさらに強調しました。

🌐 Selenium は 2004 年、Chicago の ThoughtWorks で Jason Huggins が、社内の勤怠・経費アプリケーションのテストを自動化するために「JavaScriptTestRunner」として作りました。同僚の Paul Hammant がオープンソース化を提案し、あらゆる言語からネットワーク越しにブラウザを自動操作できる Selenium Remote Control (RC) を共同開発しました。

2005

📊 PostgreSQL 8.0 は 2005 年 1 月 19 日にリリースされ、Microsoft Windows をネイティブにサポートした最初のバージョンとして注目されたほか、セーブポイントとポイントインタイムリカバリを追加しました。

☁️ Hudson は Java で書かれた継続的インテグレーションサーバで、Sun Microsystems の川口耕介が作り、2005 年 2 月 7 日に最初にリリースされました。2010 年に Oracle が Sun を買収した後、商標をめぐる争いからコミュニティはプロジェクトをフォークし、2011 年に Jenkins へと改名しました。

💻 fish (「friendly interactive shell」の頭字語) は対話性と使いやすさに重点を置いた Unix シェルで、2005 年 2 月 13 日に Axel Liljencrantz によって最初にリリースされました。他のほとんどのシェルと異なり、設計上あえて POSIX のシェル標準に従わず、代わりに構文の色分けや入力候補の自動提示といった機能によってユーザー体験を優先しています。

🌐 Prototype JavaScript Framework は、Ruby on Rails の Ajax サポートの一部として 2005 年 2 月に Sam Stephenson が作成した JavaScript フレームワークです。

📊 2005 年 2 月、Lucene は Jakarta のサブプロジェクトではなく、Apache のトップレベルプロジェクトになりました。

🌐 2005 年 4 月 1 日、Gmail の 1 周年に合わせて、Google は保存容量を 2 GB へと倍増させました。

🔐 Security Assertion Markup Language (SAML) は、特に ID プロバイダとサービスプロバイダの間で、認証・認可のデータを当事者間で交換するためのオープンな標準で、SAML 2.0 は 2005 年 3 月に OASIS 標準となりました。

⚙️ 2005 年 6 月 6 日の WWDC で、Steve Jobs は Macintosh の製品ライン全体を PowerPC から Intel のプロセッサへ移行するという Apple の計画を発表しました。これは 1994 年の Motorola 68000 から PowerPC への移行に続く、Mac の歴

史で 2 度目となる大きなプロセッサアーキテクチャの移行でした。

⚙ collectd は、コンピュータやネットワーク機器の性能データを収集・転送・保存する Unix のデーモンで、2005 年 7 月 8 日に最初にリリースされました。

⚙ Google は 2005 年 7 月 11 日に Android, Inc. を買収して完全子会社とし、モバイルオペレーティングシステムの開発を継続するために同社の主要な従業員を引き留めました。

🌐 Django は model–template–views (MTV) アーキテクチャパターンに従う、フリーかつオープンソースの Python ベース Web フレームワークで、2005 年 7 月 21 日に最初にリリースされました。

📜 RSpec は 2005 年 8 月、Steven Baker が Dave Astels、Aslak Hellesøy、David Chelimsky とともに、Ruby のためのビヘイビア駆動開発 (BDD) フレームワークとして始めました。JBehave に着想を得て、ソフトウェアの振る舞いを実行可能な例として表現する describe/it の DSL を導入しました。2006 年に David Chelimsky がプロジェクトの主導権を引き継ぎ、RSpec 1.0 は 2007 年 5 月 18 日にリリースされました。

🏢 2005 年 12 月、Yahoo! は自社の Web サービスの一部を JSON で提供し始めました。

🌐 JSONP、あるいは JSON-P (JSON with Padding) は、通常の JavaScript を読み込むための要素である <script> 要素を読み込むことでデータを要求する、歴史的な JavaScript の手法です。JSONP は同一オリジンポリシーを回避してデータを共有できるようにします。JSONP の元となる提案 (パディングがコールバック関数であるもの) は、2005 年 12 月に Bob Ippolito が行ったと見られています。

🌐 Ruby on Rails 1.0 は 2005 年 12 月 13 日にリリースされ、このフレームワークにとって最初の主要な安定版となりました。

📜 F# (F sharp と発音) は、F# Software Foundation、Microsoft、およびオープンな貢献者によって開発されている、関数型を第一とする汎用で強く型付けされたマルチパラダイムプログラミング言語で、2005 年に最初にリリースされました。

💻 Git は 2005 年、Linux カーネル開発のために Linus Torvalds が他のカーネル開発者の協力を得て作りました。2005 年 7 月 26 日に保守は濱野純へ引き継がれ、濱野は 2005 年 12 月 21 日にバージョン 1.0 をリリースし、その後もプロジェクトの中心的な保守者であり続けています。

☁ Puppet は 2005 年に Luke Kanies が立ち上げた構成管理ツールで、プロビジョニング、パッチ適用、設定を含む IT インフラのライフサイクルを宣言的な言語で管理します。Puppet 自体は Ruby で書かれており、2005 年に最初にリリースされました。

🏢 Adobe Systems は 2005 年 12 月 3 日、Macromedia を約 34 億ドルで買収し、Flash、Dreamweaver、Director、Fireworks、Authorware を手に入れました。この製品はその後 Adobe Flash として再ブランド化されました。

🧩 AOL は 2005 年、ブログ、チャットフォーラム、オンライン会議の人気が高まっていることを理由に、Usenet へのアクセス提供を終了しました。

2006

⚙️ 2006 年 1 月 10 日、Steve Jobs は Macworld Conference & Expo で最初の Intel ベースの Mac である iMac と MacBook Pro を発表し、PowerPC からの移行が始まりました。この移行は 9 か月のうちに全製品ラインで完了することになります。

📊 SQLAlchemy は Python プログラミング言語のためのオープンソースの SQL ツールキットおよびオブジェクト関係マッパーで、2006 年 2 月に最初にリリースされました。

☁️ Amazon S3 (Amazon Simple Storage Service) は、Amazon Web Services (AWS) が Web サービスのインターフェースを通じて提供するオブジェクトストレージサービスで、2006 年 3 月 14 日に米国で提供が始まりました。

📊 Apache Hadoop は、コンピュータのクラスタ全体で大規模なデータセットを分散処理するためのフレームワークで、Doug Cutting と Mike Cafarella が作り、2006 年 4 月 1 日に最初にリリースされました。Google の MapReduce と Google File System の論文に着想を得ており、中核は Hadoop Distributed File System (HDFS) と MapReduce の処理モデルから成ります。Cutting は息子のおもちゃの象のぬいぐるみにちなんで名付けました。

🏢 2006 年 4 月 5 日、World Wide Web Consortium (W3C) は、公式な Web 標準を作ろうとする試みとして、XMLHttpRequest オブジェクトの最初のドラフト仕様を公開しました。

💻 Microsoft は 2006 年 4 月 25 日、Monad を正式に Windows PowerShell へ改名し、このツールの最初のリリース候補 (RC1) を公開しました。

📜 2006 年 5 月 11 日にリリースされた Java EE 5 では、プラットフォームの名称が J2EE から Java EE に変更され、Java のアノテーションとジェネリクスの採用に加えて EJB 3.0 と JPA によって開発が大幅に簡素化されました。

⚙️ Upstart は従来の init デーモンを置き換えるイベントベースの仕組みで、2006 年 8 月 24 日に最初にリリースされ、コンピュータの起動時にタスクを実行するためにいくつかの Unix ライクなオペレーティングシステムで使われました。

☁️ Amazon Elastic Compute Cloud (EC2) は、利用者がアプリケーションを実行するため

の仮想コンピュータを借りられる Amazon Web Services (AWS) 内のクラウドコンピューティングサービスで、2006 年 8 月 25 日に限定公開ベータとして発表され、当初は Xen による仮想化のみを使用していました。

🌐 Apple は 2006 年 8 月 7 日、Mac OS X v10.5「Leopard」に Ruby on Rails を同梱すると発表しました。Leopard は 2007 年 10 月にリリースされ、このフレームワークをプリインストールした最初の主要なオペレーティングシステムとなりました。

🌐 jQuery は、HTML DOM の走査と操作、イベント処理、CSS アニメーション、Ajax を簡素化するために設計された JavaScript ライブラリです。Dean Edwards の cssQuery ライブラリに着想を得て、2006 年 1 月に BarCamp NYC で John Resig が作り、2006 年 8 月 26 日に最初にリリースされました。

💻 Windows PowerShell 1.0 は 2006 年 11 月 14 日、Windows XP SP2、Windows Server 2003、Windows Vista 向けにリリースされました。

📊 NumPy は、大規模な多次元配列や行列のサポートと高水準の数学関数を Python プログラミング言語に追加するライブラリで、2006 年に Travis Oliphant が SciPy プロジェクトの一環として Numarray の機能を Numeric へ移植し、単一の配列パッケージのもとにコミュニティを統合したことで生まれました。

🧠 2006 年、Geoffrey Hinton は多層の深いオートエンコーダを訓練するためのディープビリーフネットワークの手法を開発しました。

🔐 2006 年、SSH プロトコルの改訂版である SSH-2 が標準として採用され、元の SSH-1 プロトコルとの非互換性が持ち込まれました。

🏢 Dan North は 2006 年に "Introducing BDD" を発表し、ビヘイビア駆動開発という用語を作るとともに、ビジネス上の関係者と開発者をつなぐ架け橋として Given/When/Then というシナリオの構造を定式化しました。

2007

🌐 2007 年 4 月 10 日、Mozilla、Apple、Opera は、W3C の新しい HTML Working Group が WHATWG の HTML5 ドラフトを出発点として採用することを提案し、並行して進んでいた 2 つの標準化作業を HTML5 の名の下に正式に統合しました。

⚙️ Steve Jobs は 2007 年 1 月 9 日、Macworld Conference & Expo の基調講演で、当時「OS X」のバージョンとして宣伝されていた OS を搭載した初代 iPhone を発表しました。

📜 Apache Groovy は、Java の構文と互換性があり静的にも動的にも型付けできる、Java プラットフォーム向けのオブジェクト指向プログラミング言語で、Python、Ruby、Smalltalk に似た機能を備えています。2003 年 8 月に James

Strachan によって初めて提唱され、Java Community Process による標準化作業を経て、2007 年 1 月 2 日にバージョン 1.0 として正式にリリースされました。

🌐 Simon Stewart は 2007 年 1 月 3 日に WebDriver の最初のコミットをプッシュしました。これは JavaScript を注入するのではなくブラウザをネイティブに制御するブラウザ自動化ツールで、Selenium RC が抱えていたセキュリティサンドボックスの制約を克服しました。

⚙ Oracle VM VirtualBox (旧 Sun VirtualBox、Innotek VirtualBox) は Oracle Corporation が開発した x86 仮想化のための Type-2 ハイパーバイザで、2007 年 1 月 17 日に最初にリリースされました。

📊 2007 年 1 月 30 日にリリースされた Excel 2007 は、リボンインターフェースと XML ベースの新しいファイル形式 .xlsx を導入し、それまでのバージョンで使われていたバイナリ形式の .xls を置き換えました。

⚙ Kernel-based Virtual Machine (KVM) は、カーネルをハイパーバイザとして機能させる Linux カーネルの仮想化モジュールです。2007 年 2 月 5 日にリリースされた Linux カーネルのバージョン 2.6.20 で本流に取り込まれました。

🏢 Sun は 2006 年 11 月 13 日、Java HotSpot 仮想マシンとコンパイラを GNU General Public License の下でフリーソフトウェアとして公開し、Java Runtime Environment を含む JDK の残りの部分も 2007 年 3 月までに GPL の下に置くと約束しました。

📜 Rake は Ruby で実装された Make ライクなビルド自動化プログラムで、タスクと依存関係を標準的な Ruby の構文で記述します。バージョン 0.7.3 は 2007 年 4 月 21 日にリリースされました。

📜 OpenJDK (Open Java Development Kit) は Java SE のフリーかつオープンソースの実装で、Sun Microsystems が 2006 年に始めた取り組みの成果です。HotSpot 仮想マシン、Java クラスライブラリ、javac コンパイラを生み出しており、最初のリリースは 2007 年 5 月 8 日でした。

🧠 scikit-learn プロジェクトは、フランスのデータサイエンティスト David Cournapeau による Google Summer of Code のプロジェクト scikits.learn として始まり、2007 年 6 月に最初にリリースされました。

⚙ Apple のモバイルオペレーティングシステムの最初のバージョンである iPhone OS 1 は、初代 iPhone とともに 2007 年 6 月 29 日にリリースされました。

☁ Heroku は 2007 年 6 月、James Lindenbaum、Adam Wiggins、Orion Henry によって、当初は Ruby の Web アプリケーションを対象とするクラウドの Platform as

a Service (PaaS) として設立され、プロトタイプの開発には約 6 か月を要しました。同プラットフォームは 2008 年 5 月に 300 万ドルの資金を調達しました。

📜 PyPy は、標準実装である CPython に代わる Python プログラミング言語の実装で、ジャストインタイムコンパイラを備えるため多くの場合より高速に動作します。研究プロジェクトとして始まり、2007 年半ばに成熟した 1.0 リリースに到達して、その後は本番環境での利用可能性と CPython との互換性の向上に力が注がれました。

🌐 Sinatra は、Ruby で Web アプリケーションを構築するための軽量なドメイン固有言語 (DSL) として、2007 年 9 月 9 日に Blake Mizerany が作成し、オープンソースとして公開しました。

🌐 2007 年 10 月、Gmail は競合他社の動きを受けて IMAP のサポートを追加し、保存容量を 4 GB に増やしました。

🌐 GitHub.com プラットフォームの開発は 2007 年 10 月 19 日に始まりました。数か月前からベータ版として提供された後、サイトは 2008 年 4 月に Tom Preston-Werner、Chris Wanstrath、P. J. Hyett、Scott Chacon によって公開されました。

📜 Language Integrated Query (LINQ) は、.NET 言語にネイティブなデータクエリ機能を追加する Microsoft .NET Framework のコンポーネントで、当初は .NET Framework 3.5 の主要な一部として 2007 年 11 月 19 日にリリースされました。

📜 2007 年 11 月に .NET Framework v3.5 とともにリリースされた C# 言語 v3.0 は、匿名関数の完全なサポートも備えています。

📜 F# は 2007 年、バージョン 2.0 で await ポイントを備えた非同期ワークフローを追加しました。これは C# に追加された async/await の仕組みに影響を与えました。

📜 Go は 2007 年後半、C++ の複雑さとコンパイルの遅さへの不満を動機として、Google の Robert Griesemer、Rob Pike、Ken Thompson によって構想されました。

2008

📊 Pandas はデータ操作と分析のための Python ライブラリで、数値の表や時系列を扱うためのデータ構造と演算を提供します。2008 年 1 月 11 日に最初にリリースされました。

🌐 HTML5 は World Wide Web 上でコンテンツを構造化し提示するために使われるマークアップ言語です。最初の公開 W3C ドラフトは 2008 年 1 月 22 日に公開され、<canvas>、<video>、<audio>、ローカルストレージ、ジオロケーション、Web Workers、WebSocket といった画期的な機能を導入しました。これらは全体として、ブラウザプラグインなしでリッチな Web アプリケーションを実現するものです。

⚙ コントロールグループの機能は、2008 年 1 月にリリースされた Linux カーネルのバージョン 2.6.24 で本流に取り込まれました。

⚙ FreeBSD 7.0-RELEASE は 2008 年 2 月 27 日にリリースされました。このバージョンでは ZFS ファイルシステムの実験的なサポートが追加され、マルチコアシステムでの性能を改善するために ULE スケジューラが導入されました。

📜 Sphinx は Python で書かれた文書生成ツールで、Python コミュニティで広く使われており、2008 年 3 月 21 日に最初にリリースされました。reStructuredText のファイルを HTML の Web サイトや、PDF、EPub、Texinfo、man ページといった他の形式に変換します。

📊 HBase は Google の Bigtable をモデルとし Java で書かれた、オープンソースの非リレーショナル分散データベースで、2008 年 3 月 28 日に最初にリリースされました。Apache Hadoop プロジェクトの一部として HDFS や Alluxio の上で動作し、Bigtable のような機能を Hadoop にもたらします。

📜 Gradle は多言語のソフトウェア開発に対応したビルド自動化ツールで、コンパイルとパッケージングからテスト、デプロイ、公開まで、ライフサイクル全体を扱います。2008 年 4 月 21 日に最初にリリースされました。

☁ 2008 年 4 月 7 日、Google は、同社が管理するデータセンターで Web アプリケーションを開発・ホスティングするためのプラットフォームである App Engine を発表しました。これは Google にとって最初のクラウドコンピューティングサービスでした。

📜 PDF は 2008 年 7 月 1 日に ISO 32000-1:2008 としてオープンな ISO 標準として発行され、Adobe によるこの形式の独占的な管理は終わりました。標準化されたバージョンは PDF 1.7 に基づいており、ISO の第 171 専門委員会によって承認されました。

⚙ iPhone OS 2 は 2008 年 7 月 11 日にリリースされ、App Store とサードパーティ製アプリケーションのサポートを導入して、プラットフォームの能力を大きく広げました。

🌐 Jinja は Python プログラミング言語のための Web テンプレートエンジンです。Armin Ronacher が作成し、BSD License の下でライセンスされており、最初のリリースは 2008 年 7 月 17 日でした。

📊 Apache Cassandra は、多数のコモディティサーバにまたがって大量のデータを扱うために設計された、フリーかつオープンソースの分散型ワイドカラムストア NoSQL データベース管理システムで、単一障害点のない高い可用性を提供します。2008 年 7 月に最初にリリースされました。

⚙ Linux Containers (LXC) は、単一のカーネル上で複数の隔離された Linux システム

を動かすための OS レベルの仮想化方式で、2008 年 8 月 6 日に最初にリリースされました。カーネルの cgroups と名前空間による隔離を組み合わせて軽量なコンテナを提供し、Docker が当初その上に構築された基盤となりました。Docker は 2014 年にこれを独自の libcontainer へ置き換えました。

🧩 ドメイン名 bitcoin.org は 2008 年 8 月 18 日に登録されました。

☁️ 2008 年 8 月 20 日、Amazon は永続的なストレージを提供する Elastic Block Store (EBS) を追加しました。これは EC2 の登場以来欠けていた重要な機能でした。

🔐 TLS 1.2 は 2008 年 8 月に RFC 5246 で定義されました。

🌐 Google Chrome は Google が開発するクロスプラットフォームの Web ブラウザです。2008 年 9 月 2 日に Microsoft Windows 向けに最初にリリースされ、Apple の WebKit と Mozilla Firefox のフリーソフトウェアコンポーネントを用いて構築されました。

🌐 V8 は当初 Google Chrome のために構築された JavaScript の実行エンジンです。その後 2008 年に Google によってオープンソース化され、最初のバージョンは Chrome の最初のバージョンと同じ 2008 年 9 月 2 日にリリースされました。V8 の開発の多くは Sun Microsystems が開発した Java HotSpot 仮想マシンから強い影響を受けており、より新しい実行パイプラインは HotSpot のものと非常によく似ています。

💻 Stack Overflow はコンピュータプログラマ向けの質問・回答サイトで、2008 年 9 月 15 日に Jeff Atwood、Joel Spolsky、Jarrod Dixon によって公開されました。Stack Overflow はプログラマにとって最大のオンラインコミュニティとなり、利用者が質問し、回答を寄せ、投稿の質に投票できる、プログラミング知識の中心的なリポジトリとして機能しています。

⚙️ オペレーティングシステムの最初の商用バージョンである Android 1.0 は 2008 年 9 月 23 日にリリースされ、Google のサービスとの統合と Android Market の導入を特徴としていました。

📶 DuckDuckGo はオンラインプライバシーに重点を置くアメリカのソフトウェア企業で、主力製品は DuckDuckGo という名前の検索エンジンです。2008 年 9 月 25 日に公開されました。

⚙️ Open Virtualization Format (OVF) は、仮想アプライアンス、より一般的には仮想マシン上で実行されるソフトウェアをパッケージ化して配布するためのオープンな標準で、2008 年 9 月に最初にリリースされました。

🧩 2008 年 10 月 31 日、Satoshi Nakamoto を著者とする "Bitcoin: A Peer-to-Peer Electronic Cash System" と題されたホワイトペーパーへのリンクが、暗号技術のメ

ーリングリストに投稿されました。

📜 Python 3.0 は 2008 年 12 月 3 日にリリースされました。明確さを高め、過去のしがらみによる複雑さを取り除くために、あえて後方互換性を壊した言語の大きな改訂版です。

🏛️ 2008 年、Microsoft は Apache Software Foundation に加わり、Google、Facebook、Sun、IBM、Apache などとともに Open Web Foundation を共同で設立しました。

⚙️ Graphite は、コンピュータシステムの性能のような数値の時系列データを監視し、グラフ化するフリーのオープンソースソフトウェア (FOSS) ツールです。Orbitz Worldwide, Inc. が開発し、2008 年にオープンソースソフトウェアとして公開されました。

🧩 UTF-8 は 2008 年、ASCII やその他の従来の文字符号化を抜いて、World Wide Web で最も一般的な文字符号化となりました。

📜 Cucumber は 2008 年、RSpec Story Runner (旧 RBehave) の後継として Aslak Hellesøy が作成し、平易な言葉で書かれた Gherkin のシナリオを開発者と技術者以外の関係者の間で共有できるようにしました。この名前は、当初 Hellesøy がプロジェクトを「Stories」と呼んでいたのを受けて、彼の婚約者が提案したものです。

2009

☁️ 2009 年 1 月、Heroku は 3 か月かけて作り直したプラットフォームの新しいバージョンを公開しました。

☁️ Progress Chef (旧 Chef) は Ruby と Erlang で書かれた構成管理ツールです。システム構成の「レシピ」を書くために、純粋な Ruby のドメイン固有言語 (DSL) を使用します。2009 年 1 月に最初にリリースされました。

🌐 Cross-Origin Resource Sharing (CORS) は、Web ページ上の制限されたリソースを、最初のリソースが配信されたドメインとは別のドメインから要求できるようにする仕組みで、ドラフト仕様が最終的な名称に改称されたのは 2009 年 3 月でした。

⚙️ Android 1.5 Cupcake は 2009 年 4 月 27 日にリリースされ、メジャーバージョンに菓子の名前を付ける慣習の始まりとなったほか、画面上のキーボードやサードパーティ製ウィジェットのサポートといった重要な機能を導入しました。

📊 WolframAlpha は Wolfram Research が開発した回答エンジンです。このエンジンは、同社の以前の製品である技術計算プラットフォーム Wolfram Mathematica を基盤としています。2009 年 5 月 18 日に公開されました。

⚙️ Homebrew は、Apple のオペレーティングシステムである macOS や Linux でのソフトウェアのインストールを簡単にする、フリーかつオープンソースのソフトウェアパ

ッケージ管理システムです。もともと Max Howell が書いたもので、Ruby on Rails コミュニティで人気を集め、その拡張性が高く評価されてきました。2009 年 5 月 21 日に最初にリリースされました。

🌐 JavaServer Faces 2.0 は 2009 年 7 月 1 日、使いやすさ、機能の強化、性能の面で重要なリリースとして、Java EE 6 に合わせて公開されました。Java EE 6 では Facelets が JSP に代わる公式のビュー技術として採用されました。

🌐 Gmail は 2009 年 7 月 7 日にベータ版の位置づけを終えました。

💻 Windows PowerShell 2.0 は 2009 年 7 月、Windows 7 と Windows Server 2008 R2 に不可欠な構成要素としてリリースされ、PowerShell Remoting やバックグラウンドジョブといった重要な機能を導入しました。

💻 Zsh の設定を管理するコミュニティ主導のフレームワークである Oh My Zsh は、2009 年 8 月 28 日、Robby Russell が自身のシェル設定を Git リポジトリにまとめて GitHub に置き、毎日一緒に働く同僚たちが自分とともに改善できるようにしたことで誕生しました。2024 年までに貢献者は 2,300 人を超え、300 以上のプラグインと 140 以上のテーマを擁するまでに成長しました。

📜 CommonJS は、Web ブラウザの外部で使われる JavaScript のモジュールエコシステムについて規約を定めることを目的としたプロジェクトです。Mozilla のエンジニアである Kevin Dangoor が 2009 年 1 月に開始し、当初は ServerJS という名前でした。2009 年 8 月、API の適用範囲の広さを示すためにプロジェクトは CommonJS へと改名されました。

💻 2009 年 9 月、Joel Spolsky の会社である Fog Creek Software は Stack Exchange 1.0 プラットフォームのベータ版を公開し、Stack Overflow の質問・回答モデルをより広いサイト網へと広げました。

📊 Amazon Relational Database Service (Amazon RDS) は Amazon Web Services (AWS) による分散型のリレーショナルデータベースサービスです。2009 年 10 月 22 日に MySQL のデータベースをサポートして最初にリリースされ、その後 2011 年 6 月に Oracle Database、2012 年 5 月に Microsoft SQL Server、2013 年 11 月に PostgreSQL のサポートが続きました。

⚙ 野心的ながらしばしば批判された Windows Vista (2007 年) に続いて、Windows 7 が 2009 年 10 月 22 日にリリースされました。ユーザーインターフェースを洗練させ、性能を大幅に改善したことで、企業でも家庭でも広く普及しました。

⚙ VMware Server は 2009 年 10 月 26 日に最後のリリースを迎え、その後 VMware は商用の ESXi ハイパーバイザを優先してこの製品の提供を終了しました。

📊 MariaDB は、MySQL リレーショナルデータベース管理システム (RDBMS) からコ

ミュニティ主導で開発され商用サポートも提供されるフォークで、GNU General Public License の下でフリーかつオープンソースのソフトウェアであり続けることを目指しています。2009 年 10 月 29 日に最初にリリースされました。

⚙️ IEEE 802.11n 追補 (後に Wi-Fi 4 として名付けられました) は 2009 年 10 月に承認され、発行されました。

🌐 Node.js は 2009 年に Ryan Dahl によって最初に書かれました。これは、最初のサーバサイド JavaScript 環境である Netscape の LiveWire Pro Web の登場から約 13 年後のことでした。Dahl は 2009 年 11 月 8 日、第 1 回のヨーロッパ JSConf でこのプロジェクトを実演しました。

📜 Go は、Google の Robert Griesemer、Rob Pike、Ken Thompson によって設計された静的型付けのコンパイル型プログラミング言語です。2009 年 11 月 10 日にオープンソースプロジェクトとして公開が発表されました。

📜 Haskell 98 標準の最初の改訂版である Haskell 2010 は、2009 年 11 月に発表され、2010 年 7 月に公開されました。それまでコンパイラ固有のフラグで有効化されていた、広く使われ異論の少ない機能として、外部関数インターフェース (FFI)、階層的なモジュール名、パターンガードが追加された一方、いわゆる n+k パターンは使用できなくなりました。

🏢 DevOps という用語は 2009 年、O’Reilly Velocity Conference での "10+ Deploys per Day: Dev and Ops Cooperation at Flickr" と題された講演をきっかけに生まれました。John Allspaw と Paul Hammond は、当時のソフトウェア開発ライフサイクルが抱えていた苦労のいくつかを詳しく説明しました。

🏢 2009 年、devopsdays という名前の最初のカンファレンスがベルギーの Ghent で開催されました。このカンファレンスは、ベルギーのコンサルタントであり、プロジェクトマネージャー、アジャイル実践者でもある Patrick Debois が立ち上げました。

🏢 Microsoft は 2009 年に初めて Linux カーネルへの貢献を始めました。

🌐 Google Chrome 4 は 2009 年 12 月、完全な WebSocket サポートをデフォルトで有効にして出荷した最初のブラウザとなり、ブラウザとサーバの間で持続的な双方向通信を行うことが現実に可能であることを示しました。

🌐 SPDY は、Web コンテンツを転送するために開発された、現在は廃止されたオープン仕様の通信プロトコルです。Google は 2009 年後半に SPDY を発表し、2010 年にデプロイしました。

🧩 Nakamoto は bitcoin のソフトウェアをオープンソースのコードとして実装し、2009 年 1 月に公開しました。

タイムライン - 2010-14

2010

📜 npm は JavaScript プログラミング言語のためのパッケージマネージャで、すべて JavaScript で書かれています。Isaac Z. Schlueter が「モジュールのパッケージングがひどいやり方で行われているのを見てきた」問題に対処するために開発し、PEAR (PHP) や CPAN (Perl) といった類似のプロジェクトから着想を得ました。2010 年 1 月 12 日に最初にリリースされました。

⚙ Steve Jobs は 2010 年 1 月 27 日、San Francisco の Yerba Buena Center for the Arts で初代 iPad を発表し、スマートフォンとノートパソコンの間に位置する新しいカテゴリのデバイスだと説明しました。Wi-Fi モデルは 2010 年 4 月 3 日にリリースされ、iPhone OS 3.2 を搭載していました。

🏢 Windows Azure Platform は 2010 年 2 月 1 日に商用利用が可能になりました。

📊 Elasticsearch は Apache Lucene の上に構築された分散型で RESTful な検索・分析エンジンです。Shay Banon は 2004 年にその前身である Compass を作り、2010 年 2 月 8 日に Elasticsearch の最初のバージョンを「You Know, for Search」という言葉とともにリリースしました。

🏢 Ken Schwaber は 2009 年に Scrum.org を設立し、"The Scrum Guide" の最初の公式版は 2010 年 2 月に彼と Jeff Sutherland によって公開されました。

⚙ systemd は、ディストリビューション間でサービスの設定と挙動を統一するために設計された、Linux オペレーティングシステム向けの一連のシステムコンポーネントを提供するソフトウェアスイートです。Red Hat のソフトウェアエンジニアである Lennart Poettering と Kay Sievers が、Linux 従来の System V init を置き換えるプロジェクトとして開発し、2010 年 3 月 30 日に最初にリリースされました。

🌐 Sinatra 1.0 は 2010 年 3 月 23 日にリリースされ、このフレームワークにとって最初の主要な安定版となるとともに、推奨されるテストツールとして Rack::Test を導入しました。

⚙ Vagrant は、VirtualBox、KVM、Hyper-V、Docker コンテナ、VMware、AWS にまたがって可搬性のある仮想的なソフトウェア開発環境を構築・維持するためのオープンソースソフトウェア製品で、2010 年 1 月に Mitchell Hashimoto の個人的なサイドプロジェクトとして始まり、2010 年 3 月に最初にリリースされました。

📊 2010 年 3 月、Apache Solr 検索サーバが Lucene のサブプロジェクトとして加わり、2 つの開発者コミュニティが統合されました。

🌐 Flask は特定のツールやライブラリを必要としない Python 製のマイクロ Web フ

レームワークで、Python 愛好家の国際的なグループである Pocoo の Armin Ronacher が作り、2010 年 4 月 1 日に最初にリリースされました。

🌐 WebSocket プロトコルの開発は、2010 年 2 月に WHATWG/W3C から IETF の HyBi ワーキンググループへ移されました。初期のドラフトにプロキシキャッシュポイズニングというセキュリティ脆弱性が見つかり、いくつかのブラウザは一時的に WebSocket のサポートを無効にしました。

🌐 2010 年 4 月、Steve Jobs は "Thoughts on Flash" という公開書簡を発表し、セキュリティ上の欠陥、性能の低さ、独占的な管理、バッテリーの消耗を挙げて、Apple が iOS で Flash をサポートしない理由を説明しました。この書簡は Flash の衰退を加速させ、その「死の一撃」だったと広く見なされています。

🔐 OAuth はアクセス委譲のためのオープンな標準で、インターネットの利用者がパスワードを共有することなく、他の Web サイト上にある自分の情報へのアクセスを Web サイトやアプリケーションに許可する用途で広く使われています。OAuth 1.0 プロトコルは 2010 年 4 月に RFC 5849 として公開されました。

💻 2010 年 5 月 3 日、Stack Overflow が Union Square Ventures の率いる投資家グループから 600 万ドルのベンチャーキャピタル資金を調達したことが発表されました。

☁️ Google Cloud Storage は、Google Cloud Platform のインフラ上でデータを保存し、アクセスするためのオンラインファイルストレージ Web サービスで、2010 年 5 月 19 日に提供が始まりました。

📊 BigQuery は、大量のデータに対するスケーラブルな分析を提供する Google のマネージドかつサーバレスなデータウェアハウス製品で、数兆行にわたる高速なクエリを可能にした Google 社内の Dremel 技術から生まれ、2010 年 5 月 19 日に提供が始まりました。

⚙️ Google は 2010 年 5 月 20 日に Android 2.2 Froyo をリリースしました。ジャストインタイム (JIT) コンパイラの導入によって性能が大幅に向上し、Wi-Fi ホットスポット機能のサポートが追加されました。

⚙️ Apple は 2010 年 6 月 21 日の iOS 4 のリリースに合わせて、オペレーティングシステムの名称を「iPhone OS」から「iOS」へ改めました。このリリースではマルチタスクとホーム画面のフォルダも導入されました。

🎨 Sketch 1.0 は、Adobe Illustrator の UI 作業における複雑さへの不満を解消するために Pieter Omvlee が構想した macOS 向けベクターグラフィックエディタで、2010 年 9 月 7 日に Bohemian Coding によってプライベートベータ版としてリリースされ、2010 年代半ばにかけてインターフェースデザインの主流ツールとなりました。

📜 Jasmine 1.0 は 2010 年 9 月 14 日、JavaScript のためのビヘイビア駆動開発 (BDD) テストフレームワークとして Pivotal Labs によってリリースされ、DOM に依存せずブラウザと Node.js の両方で動作します。RSpec、JSpec、JSSpec の影響を受けています。

🔐 ZAP (Zed Attack Proxy) は Apache License の下で公開されている動的アプリケーションセキュリティテストツールで、HTTPS で暗号化されたものも含め、通過するすべてのトラフィックを操作できます。最初のリリースは 2010 年 9 月に Bugtraq で告知され、数か月後に OWASP のプロジェクトとなりました。

🧩 Unicode 6.0 は 2010 年 10 月にリリースされ、日本の携帯電話事業者に由来する 722 の絵文字を取り込んで、標準化された絵文字を収録した最初の Unicode 標準のバージョンとなりました。

📊 Apache Hive は、データのクエリと分析を提供するために Apache Hadoop の上に構築されたデータウェアハウスのソフトウェアプロジェクトで、2010 年 10 月 1 日に最初にリリースされました。

📜 NuGet は、開発者が再利用可能なコードを software as a service として共有できるようにするために設計されたパッケージマネージャで、フリーかつオープンソースのクライアントアプリを備えており、当初は Outercurve Foundation が NuPack という名前で作りました。もともとは Visual Studio の拡張機能として配布されていましたが、Visual Studio 2012 以降は Visual Studio と Visual Studio for Mac のいずれもが NuGet パッケージをネイティブに扱えるようになっています。プロジェクトは 2010 年 10 月 5 日に最初にリリースされました。

📜 RSpec 2.0 は 2010 年 10 月 10 日にリリースされ、一枚岩だった gem を rspec-core (実行機構とフォーマッタ)、rspec-expectations (マッチャ)、rspec-mocks (モックとスタブ) という別々のコンポーネントへ分割して、プロジェクトが必要なものだけを取り込めるようにしました。

📜 Pillow は Python プログラミング言語のためのフリーかつオープンソースの追加ライブラリで、さまざまな画像ファイル形式を開き、加工し、保存する機能を追加します。Python Imaging Library (PIL) のフォークで、2010 年 10 月 18 日に最初にリリースされました。

🏢 Microsoft は 2010 年 10 月 19 日、当初はプライベートベータとして Office 365 を発表し、Office Web Apps、SharePoint Online、Exchange Online といった既存のクラウドサービスを単一のサブスクリプション型サービスにまとめました。

🌐 AngularJS はシングルページアプリケーションを開発するための、開発が終了したフリーかつオープンソースの JavaScript ベース Web フレームワークで、主に Google と個人や企業から成るコミュニティによって保守されていました。2010 年 10 月 20 日に最初にリリースされました。

🌐 Express.js は Node.js で RESTful API を構築するためのバックエンド Web アプリケーションフレームワークです。TJ Holowaychuk が立ち上げ、MIT License の下でフリーかつオープンソースのソフトウェアとして公開されており、バージョン 0.12 は 2010 年 11 月 16 日にリリースされました。

☁️ Salesforce.com は 2010 年 12 月 8 日、Heroku を約 2 億 1200 万ドルの現金で買収し、このクラウド PaaS プラットフォームを完全子会社としました。

🔐 JSON Web Token (JWT) は、任意で署名や暗号化を施したデータを作成するためのインターネット標準の提案で、そのペイロードには複数のクレームを主張する JSON が入ります。2010 年 12 月 28 日に最初に公開されました。

🧩 2010 年 12 月、QR コードを用いた決済に関する最初の文書化された記述が、Shaun Cooley と Andrew Charles Payne の出願した 2 件の特許として現れました。これは Symantec の Norton Labs 向けに開発された Norton Mobile Pay というプロトタイプシステムに基づくものでした。

📜 David Chelimsky、Dave Astels、Zach Dennis、Aslak Hellesøy、Bryan Helmkamp、Dan North による "The RSpec Book: Behaviour-Driven Development with RSpec, Cucumber, and Friends" は 2010 年 12 月に Pragmatic Bookshelf から出版され、Ruby におけるビヘイビア駆動開発の決定版となる手引きとなって、ユニットレベルのスペックには RSpec を、受け入れテストには Cucumber を組み合わせるワークフローを記録しました。

📜 Rust は、ガベージコレクタや参照カウントを必要とせずにメモリ安全性を保証するマルチパラダイムの汎用プログラミング言語で、Mozilla の従業員であった Graydon Hoare が 2006 年に始めた個人プロジェクトから育ちました。Mozilla は 2009 年にこのプロジェクトの支援を始め、2010 年に正式に発表しました。

2011

💻 Stack Exchange は 2011 年 1 月に 33 の Web サイトを擁して一般公開されました。当時の従業員は 27 人、利用者は 150 万人でした。

📶 Apache Kafka は分散型のイベントストアかつストリーム処理プラットフォームです。Apache Software Foundation が開発した、Java と Scala で書かれたオープンソースのシステムで、リアルタイムのデータフィードを扱うための統一された高スループット・低遅延のプラットフォームを提供するよう設計されています。2011 年 1 月に最初にリリースされました。

☁️ Jenkins は、ビルド、テスト、デプロイに関わるソフトウェア開発の一部を自動化し、継続的インテグレーションと継続的デリバリーを支援するオープンソースの自動化サーバです。もともとは Hudson という名前でしたが、Oracle との争いを経て 2011 年に Jenkins へ改名され、2011 年 2 月 2 日に最初にリリースされました。

⚙ タブレット向けに特化して最適化されたバージョンである Android 3.0 Honeycomb は 2011 年 2 月 22 日にリリースされ、新しい「Holographic」ユーザーインターフェースとハードウェアアクセラレーションを導入しました。

⚙ 未割り当てだった最後の最上位アドレスブロック (1600 万個の IPv4 アドレス) は 2011 年 2 月に IANA から各地域インターネットレジストリへ割り当てられ、IPv6 導入への圧力が一段と高まりました。

⚙ Chocolatey は Windows ソフトウェアのためのマシンレベルのコマンドラインパッケージマネージャ兼インストーラです。Rob Reynolds が作り、2011 年 3 月 23 日に最初にリリースされました (v0.9.0)。NuGet の基盤の上に構築されており、Windows における apt-get のような体験から着想を得ています。

📜 Package Installer for Python (pip) は、ソフトウェアパッケージをインストールし管理するための事実上の標準であり推奨されているパッケージ管理システムで、Python で書かれています。2011 年 4 月 4 日に最初にリリースされました。

⚙ World IPv6 Day は Internet Society が後援・主催した技術検証と広報のためのイベントで、2011 年 6 月 8 日に Google、Facebook、Yahoo、Akamai Technologies、Limelight Networks といった主要な組織が 24 時間にわたり主要サービスで IPv6 を有効にしました。

🏢 Office 365 は 2011 年 6 月 28 日に一般提供が始まり、当初は Microsoft Online Services Productivity Suite (BPOS) の後継として企業の利用者を対象としていました。

🔐 時刻ベースのワンタイムパスワード (TOTP) は、一意性の源として現在時刻を用いるワンタイムパスワード (OTP) を生成するコンピュータアルゴリズムで、2011 年 5 月に正式に RFC 6238 となりました。

⚙ 2011 年 5 月、Fedora Linux は systemd をデフォルトで有効にした最初の主要な Linux ディストリビューションとなり、Upstart を置き換えました。

📊 Apache Flink は Apache Software Foundation が開発した、ストリーム処理とバッチ処理を統合したオープンソースのフレームワークです。中核は Java と Scala で書かれた分散ストリーミングデータフローエンジンで、2011 年 5 月に最初にリリースされました。

🌐 Selenium 2.0 は 2011 年 7 月にリリースされ、元の Selenium RC と Simon Stewart の WebDriver プロジェクトを統合しました。Java、Python、Ruby、C# に向けた WebDriver のネイティブなブラウザ制御 API が新しい標準インターフェースとなりました。

⚙ Mac OS X 10.7 Lion は 2011 年 7 月 20 日にリリースされました。Apple はこのリリースから製品名を「Mac OS X」から「OS X」へ短縮し、他のオペレーティングシス

テムとブランディングを揃えました。

⚙ Linux 3.0 は 2011 年 7 月 21 日にリリースされました。2.6 から 3.0 への移行は、大規模なアーキテクチャ変更を意味するものではなく、主に Linux の 20 周年を祝い、バージョン番号を簡素にするためのものでした。

📜 Kotlin は型推論を備えたクロスプラットフォームで静的型付けの汎用プログラミング言語で、Java と完全に相互運用できるよう設計されています。Kotlin の JVM 版の標準ライブラリは Java クラスライブラリに依存していますが、型推論によって構文をより簡潔にできます。2011 年 7 月 22 日に初めて登場しました。

☁ 2011 年 7 月、Heroku は当初の Ruby に加えて Node.js と Clojure に対応し、多言語サポートを拡大しました。また、Ruby の作者であるまつもとゆきひろが、Ruby のチーフアーキテクトとして同社に加わりました。

🌐 Google Native Client (NaCl) は、Intel x86、ARM、MIPS のネイティブコードの部分集合、あるいは可搬な実行ファイルをサンドボックス内で実行するためのサンドボックス化技術でした。利用者のオペレーティングシステムに依存せず Web ブラウザからネイティブコードを安全に実行できるようにし、ChromeOS に関する Google の構想に沿って、Web アプリがネイティブに近い速度で動作することを可能にしました。2011 年 9 月 16 日に最初にリリースされました。

🖩 Apache Storm は主に Clojure プログラミング言語で書かれた分散ストリーム処理の計算フレームワークです。もともと BackType で Nathan Marz とそのチームが作り、Twitter による買収の後にオープンソース化されました。2011 年 9 月 17 日に最初にリリースされました。

🌐 Ruby on Rails 3.1 は 2011 年 8 月 31 日にリリースされ、Asset Pipeline、デフォルトの JavaScript ライブラリとしての jQuery、そして CoffeeScript と Sass のサポートを導入しました。

☁ GitLab は 2011 年、ウクライナのプログラマ Dmytro Zaporozhets が Ruby on Rails で書いたサイドプロジェクトとして生まれました。GitLab プロジェクトへの最初のコードコミットは 2011 年 10 月 8 日に行われました。

⚙ Android 4.0 Ice Cream Sandwich は 2011 年 10 月 18 日にリリースされ、洗練された「Holo」デザイン言語のもとで、電話向けとタブレット向けに分かれていたオペレーティングシステムを単一のプラットフォームへ統合することを目指しました。

📜 Microsoft は 2011 年、async/await を備えた C# を Async CTP で初めて公開しました。これらは後に 2012 年の C# 5 で正式にリリースされました。

🖩 当時 Pentaho の最高技術責任者であった James Dixon は、2011 年までに data lake という用語を作り、生データから抽出した興味深い属性を集めたより小さなリ

ポジトリである data mart と対比させました。

☁ Google App Engine は 2011 年 11 月に一般提供 (GA) となりました。

📜 Mocha は TJ Holowaychuk が作り、Node.js とブラウザのための柔軟な JavaScript テストフレームワークとして 2011 年 11 月に公開されました。BDD と TDD の両方のインターフェースを提供し、非同期処理を正確に報告することに重点を置いています。

🌐 WebSocket プロトコルは 2011 年 12 月に IETF によって RFC 6455 として確定・公開され、定められたハンドシェイク、フレーミング形式、セキュリティモデルとともに、単一の TCP 接続上での完全な双方向通信プロトコルを標準化しました。

🌐 Locust は 2011 年、ESN の Jonatan Heyman と Carl Bystrom が、Battlefield 3 のコンパニオンアプリである Battlelog の負荷テストを行う中で作りました。現実的な利用者の振る舞いを再現するのに適したツールが見つからなかったため、gevent を使って Python で Locust を構築し、単一のプロセスで数千の同時利用者を扱えるようにしました。

🎨 InVision は、静的な Sketch や Photoshop の画面をクリック可能なプロトタイプに変換するクラウドベースのプロトタイピング・コラボレーションプラットフォームで、2011 年に Clark Valberg と Ben Nadel によって設立され、デザインハンドオフの業界標準となり、評価額はピーク時に 19 億ドルに達しました。

2012

📜 コンパイラの最初の番号付きプレアルファ版である Rust 0.1 は 2012 年 1 月にリリースされました。

🌐 Gatling は 2012 年 1 月 13 日に Stéphane Landelle によって、Scala と Netty の上に構築されたオープンソースの負荷・性能テストフレームワークとして最初にリリースされ、DSL によるシナリオ定義によって性能テストを本番コードと同じように扱います。

📜 Matt Wynne と Aslak Hellesøy による "The Cucumber Book: Behaviour-Driven Development for Testers and Developers" は 2012 年 1 月に Pragmatic Bookshelf から出版され、Cucumber の決定版となる手引きとして、ビジネス上の関係者と開発者の意思疎通を橋渡しするために、平易な言葉の Gherkin 構文で実行可能な受け入れテストを書く方法を示しました。

☁ Ansible は Michael DeHaan が書き、2015 年に Red Hat に買収されました。最初のリリースは 2012 年 2 月 20 日です。

🐛「うるう日バグ」は 2012 年 2 月 29 日、Windows Azure の利用者に大きなサー

ビス障害を引き起こしました。

⚙️ Raspberry Pi は Linux ベースのオペレーティングシステムが動作する小型のシングルボードコンピュータです。Raspberry Pi Foundation が設計し、コンピュータ科学教育を広めるための安価なプラットフォームとして 2012 年 2 月 29 日に最初にリリースされました (発表日。実際の提供は 2012 年 5 月に始まりました)。最初の Model B は ARM プロセッサと 256 MB の RAM を備えて 35 ドルという価格で、世界中の愛好家、教育者、開発者が手に取れるツールとなりました。

📜 Go 1.0 は 2012 年 3 月 28 日に最初の安定版としてリリースされ、Go 1 言語仕様に対する長期的な後方互換性の保証が伴いました。

📜 2012 年 4 月 24 日、José Valim は "A peek inside Elixir’s Parallel Compiler" を公開し、Plataformatec におけるこの言語の初期の開発と設計目標を示しました。

☁️ Google Drive は 2012 年 4 月 24 日、Google Apps スイート全体でファイル管理を統合するクラウドストレージ・同期サービスとして登場しました。

☁️ Microsoft は 2012 年 6 月、仮想マシン (Linux を含む)、Web サイト、Windows Azure 向け Python SDK の本格的なサポートを発表しました。

☁️ Google Compute Engine (GCE) は Google Cloud Platform の Infrastructure as a Service (IaaS) にあたるコンポーネントで、Google の検索エンジン、Gmail、YouTube などのサービスを動かしている世界規模のインフラの上に構築されています。Google は 2012 年 6 月 28 日、Google I/O 2012 で Compute Engine を限定プレビューとして発表しました。

📜 TypeScript は Microsoft が開発・保守しているフリーかつオープンソースのプログラミング言語で、Microsoft 社内での 2 年間の開発を経て、2012 年 10 月 1 日にバージョン 0.8 として初めて公開されました。

🌐 QUIC は汎用のトランスポート層ネットワークプロトコルで、Google の Jim Roskind が最初に設計し、2012 年に実装・展開されました。正式な発表は 2012 年 10 月 12 日です。

📶 Apache Lucene バージョン 4.0 は 2012 年 10 月 12 日にリリースされ、差し替え可能なコーデックやニアリアルタイム検索の改善といった主要な機能を導入しました。

📶 Amazon Redshift は、より大きなクラウドコンピューティングプラットフォームである Amazon Web Services の一部を成すデータウェアハウス製品で、大規模なデータセットとデータベース移行を扱うために、超並列処理 (MPP) のデータウェアハウス企業 ParAccel (後に Actian が買収) の技術の上に構築されました。2012 年 10 月に最初にリリースされました。

🌐 2012 年 10 月に Windows 8 とともにリリースされた Internet Explorer 10

は、Windows の利用者に WebSocket のサポートをもたらし、Firefox 4 (2011 年) と Safari 5 (2010 年) に続いて主要ブラウザでの対応が出そろいました。

🔐 OAuth 2.0 は 2012 年 10 月に RFC 6749 として公開されました。

🌐 Emscripten は LLVM/Clang をベースとしたコンパイラで、C や C++ のソースコードを WebAssembly (あるいは、2017 年に WebAssembly が登場する前の当初のコンパイル対象であった、asm.js と呼ばれる JavaScript の部分集合) にコンパイルします。主に Web ブラウザでの実行を目的としており、最初のリリースは 2012 年 11 月 11 日です。

🔐 HTTP Strict Transport Security (HSTS) は、プロトコルのダウングレード攻撃やクッキーの乗っ取りといった中間者攻撃から Web サイトを守るためのポリシー機構です。HSTS の仕様は、2012 年 10 月 2 日に IESG が Proposed Standard の RFC としての公開を承認した後、2012 年 11 月 19 日に RFC 6797 として公開されました。

☁ GitLab CI は Dmitriy Zaporozhets が作った継続的インテグレーションのツールで、2012 年 11 月 22 日に単独のサービスとして最初にリリースされました。その後 2015 年 9 月に GitLab 本体のアプリケーションへ統合され、統一された DevOps プラットフォームを形作りました。

🌐 2012 年 12 月、W3C は HTML5 を Candidate Recommendation に指定し、仕様が機能面で完成して最終レビューに入ったことを示しました。同時に WHATWG は、バージョン番号を持たず継続的に更新される「Living Standard」として HTML を保守し始めました。

🏢 2012 年以降、Microsoft は GitHub の主要な利用者となり、.NET Core、Chakra Core、MSBuild、PowerShell、PowerToys、Visual Studio Code、Windows Calculator、Windows Terminal といったオープンソースプロジェクトや開発ツール、そして製品ドキュメントの大半 (現在は Microsoft Docs にあります) をホストするために使いました。

☁ Prometheus はイベントの監視とアラートに使われるフリーソフトウェアのアプリケーションで、HTTP のプルモデルで構築された時系列データベースにリアルタイムのメトリクスを記録します。Prometheus は、StatsD と Graphite を用いた既存のメトリクス・監視の仕組みでは自社の要求に十分応えられないと SoundCloud が気づいたことをきっかけに、2012 年から同社で開発が始まりました。

📊 Elastic NV は、Elasticsearch と関連ソフトウェアをめぐる商用のサービスと製品を提供するために 2012 年に設立されました。

🎨 Dylan Field と Evan Wallace は、Brown University で計算機科学を学んでいた 2012 年に Figma の開発を始めました。Field は同年 Thiel Fellow に選ばれ、大学

を休学することと引き換えに 234,460 ドルを得ました。

2013

📜 AsciiDoc の Ruby 実装である Asciidoctor は 2013 年 1 月 30 日にリリースされ、GitHub と GitLab で使われています。

📜 TOML は設定ファイルのためのファイル形式で、「最小限」であることを目指した明快な意味づけによって読み書きしやすいことを意図しており、辞書へ曖昧さなく対応づけられるよう設計されています。最初のリリースは 2013 年 2 月 23 日です。

📜 Ruby 2.0 は Ruby 1.9.3 と完全に後方互換であることを意図しており、公式の 2.0.0 リリースは 2013 年 2 月 24 日でした。

💻 Meson はソフトウェアのビルド（コンパイル）を自動化するソフトウェアツールで、プログラマの生産性を高めることを全体の目標としています。Meson は Apache License 2.0 の下で Python で書かれたフリーかつオープンソースのソフトウェアで、最初のリリースは 2013 年 3 月 2 日です。

🎨 Peopleware: Productive Projects and Teams は 2013 年に第 3 版として改訂されました。

📊 Apache Parquet は Twitter と Cloudera によって作られ、2013 年 3 月に初めて発表されました。

☁️ Docker は、OS レベルの仮想化を用いてコンテナと呼ばれるパッケージでソフトウェアを提供する Platform as a Service (PaaS) 製品群です。Docker は 2013 年 3 月 20 日、Santa Clara の PyCon で Solomon Hykes によって公に紹介され、オープンソースプロジェクトとして公開されました。当時はデフォルトの実行環境として LXC を使っていました。

🏢 2013 年 10 月 29 日、dotCloud, Inc. は正式に Docker, Inc. へ社名を変更し、プロジェクトの急速な成長と会社の重心の移動を反映させました。

🌐 asm.js は JavaScript の部分集合で、C などの言語で書かれたソフトウェアを、標準的な JavaScript よりかなり優れた性能特性を保ちながら Web アプリケーションとして実行できるように設計されており、2013 年 3 月 21 日に初めて登場しました。

🌐 W3C は 2013 年に WebSocket API の仕様を公開し、基盤となる RFC 6455 のプロトコルとは別に JavaScript のインターフェース (new WebSocket()) を定義して、Web プラットフォームの機能としての WebSocket の標準化を完成させました。

🏢 2013 年 4 月 15 日、Xen Project が Linux Foundation の Collaborative Project として同財団の下に移されたことが発表されました。

🔐 2013 年 4 月の Java SE 7 Update 21 以降、署名のない Java アプレットはセキュリティ警告を表示するようになり、Java 7 Update 51 からはデフォルトでブロックされるようになって、セキュリティを理由とするこの技術の衰退が始まりました。

⚙️ Google Compute Engine は 2013 年 5 月に一般提供として公開されました。

🐛 SSL 証明書の期限切れにより、2013 年 2 月 22 日に Windows Azure と Xbox Live で広範囲にわたる障害が発生しました。

📜 2013 年 5 月 28 日にリリースされた Java EE 7 は、WebSocket、JSON 処理、HTML5 のサポートを追加し、プラットフォームを Web 向けに近代化しました。

🌐 React は UI コンポーネントに基づいてユーザーインターフェースを構築するための、フリーかつオープンソースのフロントエンド JavaScript ライブラリで、Meta (旧 Facebook) と個人の開発者や企業から成るコミュニティによって保守されています。最初のリリースは 2013 年 5 月 29 日です。

📊 Plotly は Python のための対話的でオープンソースの、ブラウザベースの描画ライブラリです。会社は 2012 年に設立され、Python ライブラリは 2013 年 6 月に最初にリリースされました。

📊 Seaborn は matplotlib をベースとした Python のデータ可視化ライブラリで、見栄えがよく情報量の多い統計グラフィックスを描くための高水準のインターフェースを提供します。2013 年 12 月に最初にリリースされました。

⚙️ IEEE 802.11ac 追補 (後に Wi-Fi 5 として名付けられました) は 2013 年 12 月に公開されました。

🌐 Electron (旧称 Atom Shell) は GitHub が開発・保守するフリーかつオープンソースのソフトウェアフレームワークで、Chromium ブラウザエンジンの一種でレンダリングし、バックエンドに Node.js のランタイム環境を用いて、Web 技術でデスクトップアプリケーションを作れるよう設計されています。2013 年 7 月 15 日に最初にリリースされました。

🌐 QUIC のコードは 2012 年から Google Chrome で実験的に開発され、2013 年 8 月 20 日にリリースされた Chromium バージョン 29 の一部として発表されました。

📊 ビジネスインテリジェンスとセルフサービスのデータ可視化ツールを集めた Power BI は、2013 年 7 月 8 日に Office 365 向けに発表され、Excel の中でデータを分析し可視化できるようにしました。

🏢 "Searchable Log of All Conversation and Knowledge" の頭字語である Slack は、2013 年 8 月に正式に一般公開され、最初の 24 時間で 8,000 件の登録を集めました。

🌐 2013 年 9 月、Google は Chrome での NPAPI サポートを 2014 年中に段階的に廃止すると発表し、その「90 年代のアーキテクチャ」がハングやクラッシュ、セキュリティ事案の主な原因になっていると述べました。これをきっかけに主要ブラウザ全体で NPAPI の削除が連鎖し、現代のブラウザにおける Java アプレットのサポートは事実上終わりました。

📊 InfluxDB は InfluxData 社が開発したオープンソースの時系列データベース (TSDB) で、最初のリリースは 2013 年 9 月 24 日です。

🌐 2013 年 10 月、Ecma International は JSON の標準である ECMA-404 の初版を公開しました。

📊 Presto は SQL クエリ言語を用いる大規模データ向けの分散クエリエンジンで、そのアーキテクチャにより Hadoop、Cassandra、Kafka、AWS S3、Alluxio、MySQL、MongoDB、Teradata といったデータソースに問い合わせでき、1 つのクエリの中で複数のデータソースを使うこともできます。Hive は Facebook の規模には遅すぎると判断され、その隙間を埋めて高速なクエリを実行するために Presto が生み出されました。当初の開発は 2012 年に始まり、同年のうちに Facebook で運用が始まりました。2013 年 11 月 10 日に最初にリリースされました。

🏢 2013 年 11 月 13 日、Microsoft は Microsoft Azure プラットフォーム上で Visual Studio を software as a service として提供すると発表しました。当時 Microsoft はこれを Visual Studio Online と呼んでいました。

📊 Amazon Kinesis は、大規模なリアルタイムのストリーミングデータを処理・分析するために Amazon Web Services (AWS) が提供するサービス群で、2013 年 11 月に提供が始まりました。

⚙️ 2013 年、User namespaces の導入によって、カーネルバージョン 3.8 でコンテナに必要な機能が一通りそろいました。

🌐 MEAN (MongoDB、Express.js、AngularJS (または Angular)、Node.js) は、動的な Web サイトや Web アプリケーションを構築するためのフリーかつオープンソースの JavaScript ソフトウェアスタックです。MERN と呼ばれる派生では Angular が React に置き換わります。MEAN という頭字語は Valeri Karpov が作ったもので、彼は 2013 年のブログ記事でこの用語を紹介しました。

🏢 FIDO (「Fast IDentity Online」) Alliance は 2013 年 2 月に発足したオープンな業界団体で、「世界のパスワードへの過度な依存を減らす助けとなる」認証標準を開発し普及させることを使命として掲げています。

🔐 Adversarial Tactics, Techniques, and Common Knowledge、すなわち MITRE ATT&CK は、サイバー攻撃や侵入を分類し記述するための指針です。Mitre Corporation が作成し、2013 年に公開されました。

🏢 DevOps Research and Assessment (DORA と略されます) は Google Cloud に属するチームで、ソフトウェアエンジニアへの意識調査を行い、DevOps 運動のための研究を進めています。DORA チームは 2013 年から State of DevOps Report を公開し始めました。

2014

⚙ FreeBSD 10.0-RELEASE は 2014 年 1 月 20 日にリリースされました。このバージョンでは Clang/LLVM が GCC に代わってデフォルトのシステムコンパイラとなり、bhyve ハイパーバイザが導入されました。

💻 MkDocs はプロジェクトの文書を構築するために設計された静的サイトジェネレータで、Python で書かれ、他の環境でも使われています。Markdown のファイルを HTML のページに変換し、文書を収めた静的な Web サイトを手軽に作れます。最初のリリースは 2014 年 1 月 24 日です。

🌐 2014 年 1 月、CORS が W3C 勧告として受け入れられました。

🌐 Webpack はフリーかつオープンソースのモジュールバンドラで、主に JavaScript のために作られましたが、対応するローダーを組み込めば HTML、CSS、画像といったフロントエンドの資材も変換できます。最初のリリースは 2014 年 2 月 19 日です。

🔐 OpenID Connect は OpenID 技術の第 3 世代にあたり、2014 年 2 月に OpenID Foundation によって公開されました。OAuth 2.0 認可フレームワークの上に載る認証のレイヤーです。

⚙ 2014 年 2 月、Mark Shuttleworth は Debian の決定に従って、Ubuntu が従来の Upstart init システムに代えて systemd を採用すると発表しました。これは Debian Technical Committee 内で「広く報じられた議論」を引き起こし、最終的には論争による心労から複数の著名な開発者が辞任する結果を招きました。

⚙ eBPF (Extended BPF) は 2014 年 3 月、バージョン 3.15 で Linux カーネルに取り込まれました。Berkeley Packet Filter (BPF) の後継にあたり、より洗練された命令セットと、tracepoint や kprobe といったネットワーク以外のさまざまなカーネルフックにプログラムを結び付ける機能によって BPF を拡張しました。

🏢 .NET Foundation は、.NET Framework をめぐるオープンソースソフトウェアの開発と協働を改善するために、2014 年 3 月 31 日に Microsoft によって設立された組織です。

📜 TypeScript 1.0 は 2014 年 4 月 12 日、Microsoft の開発者向けカンファレンス Build でリリースされ、Visual Studio 2013 Update 2 が TypeScript の組み込みサポートを提供しました。

☁ 2014 年 4 月、Windows Azure は Microsoft Azure へ改称されました。

☁ Microsoft は 2014 年 4 月の Build 2014 カンファレンスで Azure Resource Manager (ARM) ポータルを発表しました。

🐛 Heartbleed は、Transport Layer Security (TLS) プロトコルの広く使われている実装である OpenSSL 暗号ライブラリの、一部の古いバージョンに存在したセキュリティ上のバグです。2012 年にソフトウェアへ持ち込まれ、2014 年 4 月に公表されました。

⚙ 2014 年 4 月下旬、systemd のアーキテクチャが「Unix 哲学に反する」うえ「フィーチャークリープ」に陥っていると主張する批判者たちによって、systemd のボイコット運動が始まりました。

📊 Apache Spark は大規模なデータ処理のためのオープンソースの統合分析エンジンで、暗黙的なデータ並列性と耐障害性を備えたクラスタプログラミングのためのインターフェースを提供します。Spark は 2009 年に UC Berkeley の AMPLab で Matei Zaharia が始め、2010 年に BSD ライセンスの下でオープンソース化されました。2013 年にプロジェクトは Apache Software Foundation へ寄贈されてライセンスを Apache 2.0 に切り替え、最初のバージョン (v1.0) は 2014 年 5 月 26 日にリリースされました。

💻 Git 2.0 は 2014 年 5 月 28 日にリリースされました。

📜 RSpec 3.0 は 2014 年 6 月 1 日にリリースされ、rspec-mocks に検証付きダブル (テストダブルにメソッドの存在を強制するもの) を、rspec-expectations に組み合わせ可能なマッチャを導入する一方、2.x で非推奨となっていた機能を削除しました。

🌐 Cypress は 2014 年、Selenium の限界への反動として Brian Mann が立ち上げました。最初のコードコミットは 2014 年 6 月 5 日に行われ、MVP に到達するまでおよそ 18 か月の開発を経て、プロジェクトは 2015 年にプライベートベータに入りました。

📜 Jest は 2014 年 6 月に Facebook によってオープンソース化されました。それ以前は 2011 年ごろから、Web チャットアプリケーションのテストのために社内で開発されていました。モジュールの自動モック化とテストの並列実行を導入し、React アプリケーションにとって主要なテストフレームワークとなりました。

☁ Docker 1.0 は 2014 年 6 月 9 日にリリースされ、このプラットフォームが企業や本番環境に向けて「本番利用可能」になったことを示しました。

☁ Kubernetes は 2014 年 6 月 10 日、DockerCon で Google によってオープンソースプロジェクトとして正式に発表されました。プロジェクトは Joe Beda、Brendan Burns、Craig McLuckie によって作られ、Google 社内のクラスタ管理システムである Borg と Omega から着想を得ています。

🎨 Google は 2014 年 6 月 25 日の Google I/O で、クロスプラットフォームのビジュアルデザイン言語である Material Design を発表しました。

☁️ Terraform は HashiCorp が作ったオープンソースの infrastructure as code ソフトウェアツールで、最初のリリースは 2014 年 7 月 28 日です。

📜 Elixir v1.0 は 2014 年 9 月 18 日にリリースされ、Erlang VM (BEAM) 上での本番利用に向けた言語の安定性と成熟を確かなものにしました。

🐛 Shellshock (Bashdoor とも呼ばれます) は Unix の Bash シェルに存在した一連のセキュリティ上のバグで、最初のものは 2014 年 9 月 24 日に公表されました。Shellshock により、攻撃者は Bash に任意のコマンドを実行させ、リクエストの処理に Bash を使う Web サーバなど、インターネットに面した多くのサービスへ不正にアクセスできる恐れがありました。

🌐 Babel はフリーかつオープンソースの JavaScript トランスコンパイラで、主に ECMAScript 2015 以降 (ES6 以降) のコードを、古い JavaScript エンジンでも動く後方互換性のある JavaScript へ変換するために使われます。最初のリリースは 2014 年 9 月 28 日です。

🌐 2014 年 10 月 28 日、HTML5 が W3C 勧告として公開されました。

🧠 2014 年 11 月 6 日、Amazon は Amazon Lab126 が開発したスマートスピーカー端末 Echo とともに Alexa を発表しました。

📜 2014 年 11 月 12 日、Microsoft は .NET Core を発表し、Linux や macOS を含めて .NET にクロスプラットフォームのサポートをもたらすとともに、.NET Foundation の統括の下で従来型の (「バザール」的な) オープンソース開発モデルを採り入れようとしました。

⚙️ Android 5.0 Lollipop は 2014 年 11 月 12 日にリリースされ、「Material Design」言語に基づくユーザーインターフェースの大幅な再設計に加えて、性能を高めるために Dalvik 仮想マシンを置き換える Android Runtime (ART) を導入しました。

☁️ 2014 年 11 月 13 日、AWS は EC2 Container Service (ECS) のプレビューを公開し、AWS 上でコンテナのインフラを利用しやすくしました。リリース時点で、Docker との連携などサードパーティとの統合も利用できました。

☁️ 2014 年 11 月 13 日、AWS は Functions as a Service (FaaS) のツールである AWS Lambda を公開しました。Lambda により、AWS の利用者は特定のトリガーと実行コードを持つ関数を定義してアップロードできます。

🐛 ストレージのアップグレードにより、2014 年 11 月 18 日に Microsoft Azure で大規模な世界的障害が発生しました。

🔐 Let’s Encrypt は Internet Security Research Group (ISRG) が運営する非営利の認証局で、Transport Layer Security (TLS) 暗号化のための X.509 証明書を無償で提供します。Let’s Encrypt は 2014 年 11 月 18 日に公に発表されました。

📊 PostgreSQL 9.4 は 2014 年 12 月 18 日にリリースされ、半構造化データを効率よく保存し索引付けするための JSONB データ型を導入しました。

🌐 HTML5 は 2014 年 10 月 28 日に W3C 勧告として公開され、10 年近くにわたる開発が完成するとともに、ビデオ、オーディオ、キャンバス、ローカルストレージ、セマンティック要素をネイティブにサポートする現代の Web の標準が確立されました。

☁️ Google Kubernetes Engine (GKE、当初は Google Container Engine と呼ばれていました) のアルファリリースは 2014 年 11 月に発表されました。

🏢 2014 年、Satya Nadella が Microsoft の新しい CEO に指名されました。Microsoft はオープンソースを中核事業に取り込み始めます。Ballmer の姿勢とは対照的に、Nadella は「Microsoft loves Linux」と書かれたスライドを掲げました。

☁️ 2014 年、バージョン 0.9 のリリースで Docker は LXC を、Go プログラミング言語で書かれた自前のコンポーネントである libcontainer に置き換えました。

⚙️ OverlayFS は Linux 向けのユニオンマウントファイルシステムの実装です。複数の異なる下位のマウントポイントを 1 つにまとめ、すべての元の場所にあるファイルとサブディレクトリを含む単一のディレクトリ構造を作り出します。2014 年、カーネルバージョン 3.18 で Linux カーネル本流に取り込まれました。バージョン 4.0 で改良され、たとえば Docker の overlay2 ストレージドライバに必要な改善がもたらされました。

☁️ Grafana は、対応するデータソースに接続すると Web 上でチャート、グラフ、アラートを提供する、マルチプラットフォームでオープンソースの分析・対話的可視化 Web アプリケーションです。Grafana は 2014 年、Orbitz でのプロジェクトから派生したものとして Torkel Ödegaard によって最初にリリースされ、InfluxDB、OpenTSDB、Prometheus といった時系列データベースを対象としています。Grafana のユーザーインターフェースは当初、Kibana のバージョン 3 をもとにしていました。

🧠 seq2seq は機械翻訳 (より一般的にはシーケンス変換) への一つのアプローチで、通信を符号化・伝送・復号の過程として捉え、機械翻訳を通信の特殊な場合として研究できるとする情報理論にその源流があります。エンコーダ・デコーダによるシーケンス変換という考え方は 2010 年代初頭に発展し、seq2seq を生み出した起点として最もよく引用されるのは 2014 年の 2 本の論文です。そこで提案された seq2seq ではエンコーダもデコーダも LSTM であり、「ボトルネック」問題を抱えていました。2014 年に提案されたアテンション機構が、このボトルネック問題を解決しました。

💻 Neovim プロジェクトは 2014 年に Thiago de Arruda によって、時代遅れになっ

た Vim のソースコードを全面的に作り直す取り組みとして開始され、2014 年 3 月には少なくとも 1 名の専任開発者を支えるためのクラウドファンディングを成功させました。

タイムライン - 2015-19

2015

🏢 2013 年までに、Prometheus は SoundCloud の本番環境の監視に導入されていました。公式な発表は 2015 年 1 月に行われました。

📊 Project Jupyter は、複数のプログラミング言語にまたがる対話的コンピューティングのためのオープンソースソフトウェア、オープンな標準、サービスを開発するプロジェクトです。2014 年に Fernando Pérez と Brian Granger が IPython からスピンオフさせ、2015 年 2 月に正式に発足しました。

🧠 Keras は、人工ニューラルネットワークのための Python インターフェースを提供し、TensorFlow ライブラリのインターフェースとしても機能するオープンソースのソフトウェアライブラリで、2015 年 3 月 27 日に最初にリリースされました。

💻 2015 年 4 月 8 日、GitHub は Git Large File Storage (Git LFS) を発表しました。これにより、利用者は大きなバイナリファイルを Git で保存し扱えるようになります。

💻 Visual Studio Code は Microsoft が Windows、Linux、macOS 向けに開発したソースコードエディタで、2015 年 4 月 29 日に Build カンファレンスで初めて発表され、その直後にプレビュービルドが公開されました。

⚙️ 2015 年 4 月、Debian 8.0 (Jessie) が systemd をデフォルトの init システムとしてリリースされました。同じ月に Ubuntu 15.04 もリリースされ、正式に systemd へ切り替えました。

🌐 Discord は 2015 年 5 月 13 日に一般公開されました。ゲームコミュニティが一緒に遊びながら会話するために特化して設計されたプラットフォームです。

📜 最初の安定版リリースである Rust 1.0 は 2015 年 5 月 15 日に発表されました。

📜 第 6 版である ECMAScript 6 (ES6)、後に ECMAScript 2015 と改称されたものは 2015 年 6 月に確定し、クラス宣言や ES6 モジュールなど、複雑なアプリケーションを書くための重要な新しい構文が加わりました。ES6 は「アロー関数」構文をサポートしており、無名関数の引数リストと本体を ⇒ 記号で区切ります。

☁️ Open Container Initiative (OCI) は、オペレーティングシステムレベルの仮想化のためのオープンな標準を設計する目的で、2015 年 6 月 22 日に DockerCon において Docker、CoreOS をはじめとする業界の主要企業により Linux Foundation の下で発足

しました。OCI は、自らの仕様を実装し、より上位のツールの基盤となるコンテナランタイム runc を開発しています。runc は 2015 年 7 月にバージョン 0.0.1 として最初にリリースされました。

🔘 Amazon Echo と Alexa は、2014 年 11 月の限定リリースを経て、2015 年 6 月 28 日に米国で本格的に発売されました。

🏢 Kubernetes 1.0 は 2015 年 7 月 21 日にリリースされました。Google は Linux Foundation と協力して Cloud Native Computing Foundation (CNCF) を設立し、Kubernetes をその出発点となる技術として提供しました。

⚙️ 2015 年 7 月 29 日にリリースされた Windows 10 は、継続的な更新による「Windows as a Service」モデルを導入し、Windows 8 でのタイル中心の試みの後にスタートメニューを復活させ、Microsoft Edge ブラウザと音声アシスタント Cortana を統合しました。

📜 Go 1.5 は 2015 年 8 月 19 日にリリースされました。コンパイラとランタイムのツールチェーン全体が C から Go へ書き直され、言語がセルフホスティングとなった画期的なリリースでした。

🌐 Phoenix v1.0 は 2015 年 9 月 1 日にリリースされ、生産性が高く信頼できる Web フレームワークとして、Elixir エコシステムの土台となりました。

🔐 FIDO2 Project は、Web のための強力な認証を実現することを目的とした、FIDO Alliance と World Wide Web Consortium (W3C) の共同の取り組みです。その中核は W3C の Web Authentication (WebAuthn) 標準と FIDO Client to Authenticator Protocol 2 (CTAP2) から成り、FIDO 2.0 Proposed Standard は 2015 年 9 月 4 日に公開されました。

🏢 2015 年 9 月 14 日、Let’s Encrypt は最初の証明書を helloworld.letsencrypt.org というドメインに対して発行しました。同じ日、ISRG はルート証明書プログラムへの申請を Mozilla、Microsoft、Google、Apple に提出しました。

🏢 2015 年 9 月、Node.js v0.12 と io.js v3.3 が Node v4.0 として再び統合されました。

🌐 Vue.js は、ユーザーインターフェースやシングルページアプリケーションを構築するための、Evan You が作ったオープンソースの model–view–viewmodel フロントエンド JavaScript フレームワークです。プロジェクトへの最初のソースコードコミットは 2013 年 7 月にさかのぼり、Vue は 2014 年 2 月に最初にリリースされ、Vue.js 1.0 は 2015 年 10 月 27 日にリリースされました。

☁️ Serverless Framework は Node.js で書かれたフリーかつオープンソースの Web

フレームワークで、AWS Lambda 上にアプリケーションを構築するために開発された最初のフレームワークです。Austen Collins が開発し、専任のチームが保守しており、2015 年 10 月に最初にリリースされました。

💻 Neovim の最初の公式リリースは 2015 年 11 月 1 日に行われ、現代的な機能と組み込みやすさに重点を置いて再構成された、拡張性の高い Vim のフォークをもたらしました。

🧠 TensorFlow は機械学習と人工知能のためのフリーかつオープンソースのソフトウェアライブラリで、幅広い用途で使えますが、とりわけ深層ニューラルネットワークの学習と推論に重点を置いています。2015 年 11 月 9 日に最初にリリースされました。

🏢 2015 年 11 月 18 日、Visual Studio Code のソースコードが MIT License の下で公開され、GitHub で入手できるようになりました。

🏢 2015 年 11 月 18 日、Microsoft は Visual Studio Online を「Visual Studio Team Services (VSTS)」へ改称すると発表しました。

📜 Python は 2015 年、バージョン 3.5 で async/await のサポートを追加しました。

📜 TypeScript は 2015 年、バージョン 1.7 で async/await のサポートを追加しました。

🏢 Cloud Native Computing Foundation (CNCF) は、コンテナ技術の発展を後押しし、その進化をめぐって技術業界の足並みをそろえるために 2015 年に設立された Linux Foundation のプロジェクトです。

🌐 2015 年に公開された HTTP/2 は、HTTP の意味づけを「ネットワーク上で」より効率よく表現します。

☁ Envoy は、ネットワークをアプリケーションから見えないものにする高性能なオープンソースのエッジ・サービスプロキシで、アーキテクチャをモノリスから脱却させるために Lyft で作られました。開発は 2015 年に始まり、2015 年の晩秋には Lyft のトラフィックの 100% を処理するようになりました。

2016

☁ 2016 年 2 月、Kubernetes 向けのパッケージマネージャ Helm がリリースされました。

☁ サーバレスの実行環境である Google Cloud Functions は、2016 年 2 月にアルファ版として利用できるようになりました。

☁ Azure IoT Hub は 2016 年 2 月 4 日に一般提供となり、IoT アプリケーション

とデバイスの間の双方向通信のための中心的なメッセージハブを提供しました。

☁ 2016 年 3 月にリリースされた Kubernetes 1.2 は、Deployment と、最大 1,000 ノードのクラスタのサポートを導入しました。

🧩 Azer Koçulu が 2014 年 8 月 15 日に npm へ公開した 11 行の JavaScript ユーティリティパッケージ left-pad は、月に 1,500 万回以上ダウンロードされ、Babel、Webpack、React といった主要なプロジェクトから依存されるまでになっていました。npm が彼の kik というパッケージ名の所有権を 2016 年 3 月 18 日に Kik Interactive へ移した紛争の後、Koçulu は抗議として 2016 年 3 月 22 日に自身の 273 個の npm パッケージすべてを公開停止し、数時間のうちに Facebook、PayPal、Netflix、Spotify のものを含む数千のプロジェクトのビルドが JavaScript エコシステム全体で壊れました。

🧩 npm は、2016 年 3 月 22 日に公開停止された left-pad パッケージを、その数時間後にバックアップからバージョン 0.0.3 を手動で再公開して復旧させました。同社はその後、他のパッケージが依存しているパッケージは公開停止できないようにポリシーを変更しました。

💻 Visual Studio Code は公開プレビューの段階を終え、2016 年 4 月 14 日にバージョン 1.0 に到達しました。その時点で 1,000 を超える拡張機能と 50 万人の実利用者を擁していました。

🔐 General Data Protection Regulation (略して GDPR、フランス語では RGPD) は、欧州連合 (EU) および欧州経済領域 (EEA) における情報プライバシーに関する欧州連合の規則です。欧州議会と欧州連合理事会は 2016 年 4 月 14 日に GDPR を採択し、2018 年 5 月 25 日に発効しました。

☁ 2016 年 4 月 20 日、Jenkins バージョン 2 が Pipeline プラグインをデフォルトで有効にしてリリースされました。このプラグインにより、Apache Groovy に基づくドメイン固有言語でビルド手順を記述できます。

⚙ Snap は、Linux カーネルと systemd init システムを使うオペレーティングシステム向けに Canonical が開発したソフトウェアのパッケージング・配備システムです。最初の snapd 2.12 は Ubuntu 16.04 (xenial-updates) と Fedora 24 に向けてリリースされました。

🔐 Web Authentication (WebAuthn) は World Wide Web Consortium (W3C) が公開した Web の標準です。WebAuthn は FIDO Alliance の指導の下にある FIDO2 Project の中核となる構成要素です。2016 年 5 月 31 日に最初に公開されました。

🏛 2016 年 5 月、Cloud Native Computing Foundation は Kubernetes に続く 2 番目のインキュベーションプロジェクトとして Prometheus を受け入れました。

🌐 ASP.NET Core は Microsoft が開発したフリーかつオープンソースの Web フレームワークで、ASP.NET の後継にあたります。2016 年 6 月 7 日に最初にリリースされました。

📜 Yarn は、Node.js の JavaScript ランタイム環境のために 2016 年に Facebook が開発したソフトウェアパッケージングシステムで、2016 年 6 月 18 日に最初にリリースされました。

📜 .NET Core 1.0 は 2016 年 6 月 27 日、.NET Core の開発を可能にする Microsoft Visual Studio 2015 Update 3 とともにリリースされました。

💻 Language Server Protocol (LSP) は、ソースコードエディタや統合開発環境 (IDE) と、プログラミング言語固有の機能を提供するサーバとの間で使う、JSON-RPC に基づくオープンなプロトコルです。もともと Microsoft Visual Studio Code のために開発され、現在はオープンな標準となっています。2016 年 6 月 27 日、Microsoft はこのプロトコルの仕様を標準化するために Red Hat および Codenvy と協力すると発表しました。

💻 VS Code の Go 拡張機能は Go プログラミング言語に対する充実した言語サポートを提供するもので、2016 年 7 月に最初にリリースされました。

☁ Prometheus 1.0 は 2016 年 7 月にリリースされました。その後も 2016 年から 2017 年にかけてバージョンが重ねられ、2017 年 11 月の Prometheus 2.0 へとつながりました。

⚙ Windows Subsystem for Linux (WSL) は、ELF 形式の Linux バイナリ実行ファイルを Windows 上でネイティブに実行するための互換レイヤーです。WSL のベータ版は 2016 年 8 月 2 日、Windows 10 バージョン 1607 (Anniversary Update) で導入されました。サポートされていたのは Ubuntu (デフォルトのシェルは Bash) だけでした。

💻 Microsoft は 2016 年 8 月 18 日、PowerShell を MIT License の下でオープンソース化しクロスプラットフォーム化すると発表し、Linux と macOS 向けのアルファ版を公開しました。

☁ Lyft は 2016 年 9 月 14 日に Envoy をオープンソース化し、大規模なサービス指向アーキテクチャ向けに設計された C++ 製の L7 プロキシおよび通信バスとして発表しました。バージョン 1.0.0 はその 2 日前にタグ付けされていました。

⚙ macOS 10.12 Sierra は 2016 年 9 月 20 日にリリースされました。Apple はオペレーティングシステムの名称を「OS X」から「macOS」へ改め、他のオペレーティングシステム (iOS、watchOS、tvOS) とブランディングを揃えました。また、Mac に初めて Siri を導入しました。

☁ Cilium 0.1 は 2016 年 9 月にリリースされました。eBPF を活用して、コンテナ

化されたワークロードに高性能なネットワーク、セキュリティ、可観測性をもたらすクラウドネイティブのプロジェクトです。

🌐 Angular は、Google の Angular Team と、個人や企業から成るコミュニティが主導する TypeScript ベースのフリーかつオープンソースの Web アプリケーションフレームワークです。Angular 2.0 は 2014 年 10 月 22 日から 23 日の ng-Europe カンファレンスで発表されましたが、2.0 での大幅な変更は開発者の間でかなりの論争を呼びました。バージョン 2.0 は 2016 年 9 月 14 日にリリースされました。

🧠 PyTorch は Torch ライブラリをベースとした機械学習フレームワークで、コンピュータビジョンや自然言語処理といった用途に使われています。もともと Meta AI が開発し、現在は Linux Foundation の傘下にあります。最初のリリースは 2016 年 9 月です。

🎨 Figma は 2016 年 9 月 27 日に初めて一般公開され、ブラウザベースの共同インターフェースデザインツールとして登場しました。

🏢 Google Apps は 2016 年 9 月 29 日に正式に G Suite へ改称され、クラウドベースの生産性ツールが持つ協働的で知的な性質が強調されました。

🌐 Next.js は Vercel が作ったオープンソースの Web 開発フレームワークで、React ベースの Web アプリケーションでサーバサイドレンダリングを行い、静的な Web サイトを生成できるようにします。2016 年 10 月 25 日に最初にリリースされました。

🌐 Nuxt.js は Vue.js、Node.js、Webpack、Babel.js に基づくフリーかつオープンソースの JavaScript ライブラリで、React.js をベースとした同様の目的のフレームワークである Next.js に着想を得ています。2016 年 10 月 26 日に最初にリリースされました。

🌐 HTML 5.1 は 2016 年 11 月に W3C 勧告として公開され、レスポンシブ画像のための <picture> 要素、<details> 要素と <summary> 要素、inputmode 属性といった機能が追加されました。

🏢 2016 年 11 月 15 日、Microsoft は Azure Functions の一般提供を発表しました。

🏢 Hugging Face, Inc. は、機械学習を用いてアプリケーションを構築するための計算ツールを開発する、New York 市に拠点を置くアメリカの企業です。同社は 2016 年に、当初は 10 代の若者向けのチャットボットアプリを開発する企業として設立されました。Hugging Face Hub は、モデルなどをホストする中心的な Web サービスとして機能するプラットフォームです。

2017

🌐 k6 は 2017 年 2 月に Load Impact によってオープンソース化されました。Go で書かれ JavaScript でスクリプトを記述する現代的な負荷テストツールで、開発者のワークフローや CI/CD への組み込みを念頭に設計されています。Load Impact はその後 2020 年に k6 へ社名を変更し、2021 年に Grafana Labs に買収されました。

🌐 WebAssembly (Wasm と略されることもあります) は、可搬なバイナリコード形式と、実行可能なプログラムのための対応するテキスト形式を定義しています。2015 年に発表され、2017 年 3 月に最初にリリースされました。

🏢 Google は 2017 年 3 月、データサイエンティストと機械学習の実践者による世界最大のコミュニティである Kaggle を買収しました。

☁ Kubernetes 1.6 は 2017 年 3 月にリリースされ、ロールベースアクセス制御 (RBAC) を導入するとともに、etcd v3 をデフォルトのストレージバックエンドとしました。

🧠 DVC は、データ、機械学習モデル、実験のためのフリーかつオープンソースでプラットフォームに依存しないバージョン管理システムです。ML モデルを共有可能にし、実験を再現可能にし、モデル・データ・パイプラインのバージョンを追跡できるように設計されており、2017 年 5 月 4 日に最初にリリースされました。

📊 世界規模で分散するマルチモデルのデータベースサービスである Azure Cosmos DB は、2017 年 5 月 10 日に一般提供となりました。

☁ 2017 年 5 月 24 日、Google、IBM、Lyft はマイクロサービスを統一的に接続、保護、管理、監視する方法を提供するオープンソースプロジェクト Istio の最初の公開リリースを発表しました。Lyft の Envoy プロキシを用いて構築されており、アプリケーションコードを変更することなくトラフィックの可視性と制御を提供します。

☁ Google Cloud IoT Core は、IoT デバイスを接続し管理するためのサービスとして 2017 年 5 月にベータ版が公開されました。

🐛 WannaCry ランサムウェア攻撃は、2017 年 5 月に WannaCry ランサムウェア暗号ワームによって引き起こされた世界規模のサイバー攻撃です。Microsoft Windows オペレーティングシステムが動作するコンピュータを標的とし、データを暗号化して暗号通貨 Bitcoin による身代金の支払いを要求しました。

☁ 2017 年 6 月にリリースされた Kubernetes 1.7 は、Third Party Resources (TPR) に代わるものとして Custom Resource Definitions (CRD) を導入しました。

🏢「Microsoft 365」というブランドは、2017 年 7 月 10 日の Microsoft Inspire で、Office 365 に Windows 10 Enterprise のライセンスと Enterprise Mobility + Security (EMS) を組み合わせた企業向けのサブスクリプション製品として初めて登場しま

した。

⚙ 2017 年 7 月 14 日、RFC 2460 に取って代わる RFC 8200 の公開により、IPv6 は Internet Standard として承認されました。

☁ OCI v1.0 は 2017 年 7 月 19 日に正式にリリースされ、Runtime Specification と Image Specification が安定版となりました。OCI Runtime Specification のリファレンス実装である runc 1.0 も同じ日にリリースされました。

🌐 2017 年 7 月、Adobe はセキュリティ更新も含めて 2020 年末までに Flash Player を完全に終了させる意向を発表し、すべての主要なブラウザとプラットフォームが削除計画を固める契機となりました。

📊 Apache Iceberg は巨大な分析用テーブルのためのオープンソースの高性能フォーマットで、ビッグデータを SQL のテーブルとして扱えるようにしつつ、Spark、Trino、Flink、Presto、Hive、Impala、StarRocks、Doris、Pig といったエンジンが同じテーブルを同時に安全に扱えるようにします。2017 年 8 月 10 日に最初にリリースされました。

📜 2017 年 9 月 12 日、Oracle Corporation は Java EE を Eclipse Foundation へ移管すると発表しました。Oracle が「Java」の商標を保持したため、移管先で後に Jakarta EE へ改称されました。

☁ Lyft は 2017 年 9 月 13 日に Envoy を Cloud Native Computing Foundation へ寄贈し、インキュベーションの成熟度レベルで受け入れられました。

🌐 Java 9 は 2017 年 9 月 21 日にリリースされ、Java Applet API (java.applet パッケージ) を正式に非推奨として、Java プラットフォームにおけるアプレットサポートの公式な終わりを告げました。

🌐 Cypress は 2017 年 5 月 4 日にオープンソース化を発表し、2017 年 10 月 10 日に公開ベータへ移行しました。単一のモノレポへ再構成され、npm でインストールできる Node.js モジュールとして提供されるようになり、3 年間の非公開の開発に区切りが付きました。

⚙ WSL は、2017 年 10 月 17 日にリリースされた Windows 10 バージョン 1709 (Fall Creators Update) でベータではなくなりました。複数の Linux ディストリビューションをインストールでき、Windows Store からも入手できるようになりました。

⚙ Ubuntu 17.10 (Artful Aardvark) は 2017 年 10 月 19 日にリリースされ、GNOME Shell のインターフェースを採用し、デフォルトで X11 ウィンドウシステムを Wayland ディスプレイサーバに置き換えた最初のリリースとなりました。この変更は、Unity をデフォルトのデスクトップとして使い続けたおよそ 6 年半の後に行われました。

🏢 2017 年 10 月末、Microsoft は Azure 上のマネージド Kubernetes サービスである AKS (Azure Container Service) のプレビューを発表しました。

⚙️ 2017 年 11 月 6 日、Amazon は KVM ハイパーバイザを中心とした Nitro と呼ばれる独自アーキテクチャに基づく、新しい C5 インスタンスファミリーを発表しました。

🧠 Amazon SageMaker は、開発者がクラウド上で機械学習 (ML) モデルを作成、学習、デプロイできるようにするクラウドの機械学習プラットフォームで、組み込みシステムやエッジデバイスへの ML モデルの配備も可能にします。2017 年 11 月 29 日に公開されました。

☁️ Google Kubernetes Engine (GKE) は 2017 年 11 月に CNCF の認定を受け、より広いエコシステムとの互換性が確かなものになりました。

🏢 Databricks は、Scala の上に構築されたオープンソースの分散コンピューティングフレームワークである Apache Spark を生み出すことに関わった、University of California, Berkeley の AMPLab プロジェクトから育ちました。2017 年 11 月、同社は Azure Databricks としての統合を通じて、Microsoft Azure のファーストパーティサービスになると発表されました。

☁️ コンテナと同じような使い勝手と性能を持つ軽量な仮想マシンを提供するオープンソースプロジェクトである Kata Containers は、2017 年 12 月 5 日に発表されました。Intel Clear Containers と Hyper.sh runV の統合によって生まれたものです。

🌐 HTML 5.2 は 2017 年 12 月に W3C 勧告として公開され、<dialog> 要素と Payment Request API のサポートが追加される一方、廃れた機能が削除されました。

📜 ISO は 2017 年、PDF 2.0 を ISO 32000-2 として発行しました。これは Adobe の独自拡張を含まず、完全に ISO によって策定された最初の PDF 仕様のバージョンで、デジタル署名の改良、関連ファイル、地理空間データのより良いサポートといった新機能を導入しました。

🌐 WebAssembly System Interface (WASI) は Mozilla が設計した簡潔なインターフェース (ABI と API) で、あらゆるプラットフォームへ移植できることを目指しています。ケイパビリティベースのセキュリティで制約されたファイル I/O のような、POSIX に似た機能を提供します。

📜 JavaScript は 2017 年、ECMAScript 2017 版の一部として async/await のサポートを追加しました。

🧠 トランスフォーマーは自己注意 (self-attention) の仕組みを採り入れた深層学習モデルで、入力データの各部分の重要性に差をつけて重み付けします。主に自然言語処理 (NLP) とコンピュータビジョン (CV) の分野で使われています。トランスフォーマーは 2017 年

に Google Brain のチームによって発表され、long short-term memory (LSTM) のような RNN モデルに取って代わり、NLP の問題で第一に選ばれるモデルになりつつあります。

🌐 Uvicorn は Python 向けの非常に高速な ASGI (Asynchronous Server Gateway Interface) サーバ実装で、Tom Christie が作り、2017 年後半に最初にリリースされました。

🌐 Puppeteer は 2017 年、Google の Chrome DevTools チームによってオープンソース化されました。Chrome DevTools Protocol (CDP) の上に高水準の API を提供し、ヘッドレスの Chrome と Chromium を制御する Node.js のライブラリです。Puppeteer 1.0 は 2018 年 1 月にリリースされ、JavaScript エコシステムにおけるヘッドレスブラウザ自動化の主要なツールとなり、後の Playwright の誕生にもつながりました。

2018

💻 PowerShell Core 6.0 は 2018 年 1 月 10 日に一般提供となりました。.NET Core の上に構築されており、Windows、macOS、Linux をまたぐクロスプラットフォームでの利用を前提に設計された最初のバージョンです。

☁ 2018 年 1 月 15 日、AWS は AWS Lambda のサポート言語として Go を発表しました。

🐛 Meltdown は、最初に見つかった 2 つの一時的実行に関する CPU 脆弱性のうちの 1 つです (もう 1 つは Spectre)。Meltdown は Intel の x86 マイクロプロセッサ、IBM の POWER プロセッサ、および一部の ARM ベースのマイクロプロセッサに影響します。権限のない不正なプロセスがすべてのメモリを読み取れてしまいます。発見されたのは 2018 年 1 月です。

🐛 Spectre は、最初に見つかった 2 つの一時的実行に関する CPU 脆弱性のうちの 1 つ (もう 1 つは Meltdown) を指し、マイクロアーキテクチャレベルのタイミングを利用したサイドチャネル攻撃を伴います。分岐予測やその他の投機的実行を行う現代のマイクロプロセッサが影響を受けます。発見されたのは 2018 年 1 月です。

⚙ OpenSSH をベースとしたクライアントとサーバのプログラムは、バージョン 1803 (April 2018 Update) 以降の Windows 10 に含まれています。

💻 Visual Studio Code の Python 拡張機能は、Python 言語に対する充実したサポートを備えた Visual Studio Code の拡張機能で、2018 年 2 月 2 日に最初にリリースされました。

☁ Argo CD は Kubernetes 向けの宣言的な GitOps 継続的デリバリーツールで、大企業のニーズに適合する Kubernetes ネイティブなソリューションとして Intuit で開発が

始まりました。リポジトリは 2018 年 2 月 9 日に作成され、バージョン 0.1.0 が 2018 年 3 月 13 日にリリースされました。

🌐 Nest (NestJS) は、効率的でスケーラブルな Node.js のサーバサイドアプリケーションを構築するためのフレームワークで、2018 年 2 月 16 日に GitHub で最初にリリースされました。

🧠 Cloud TPU (Tensor Processing Unit) は 2018 年 2 月にベータ版として利用できるようになり、機械学習のためのハードウェアアクセラレーションを提供しました。

📜 Deno は、V8 JavaScript エンジンと Rust プログラミング言語に基づく JavaScript、TypeScript、WebAssembly のランタイムで、Node.js も生み出した Ryan Dahl が共同で作りました。2018 年 5 月 13 日に最初にリリースされました。

☁️ Google One は 2018 年 5 月 14 日に登場し、Google Drive の有料サービスを置き換えて、このプログラムが複数の Google のサービスにまたがって使われるものであることを打ち出しました。

🌐 W3C は 2018 年 5 月 31 日、WebDriver の仕様を勧告として公開し、Selenium に由来するブラウザ自動化プロトコルを標準化しました。実装は Google (Chrome)、Mozilla (Firefox)、Apple (Safari) によって支えられています。

🌐 Hypercorn は HTTP/1、HTTP/2、HTTP/3 に対応した高性能な ASGI・WSGI の Web サーバで、2018 年 6 月 2 日にオープンソースとして最初に公開されました。

🏢 2018 年 3 月、Kubernetes は Cloud Native Computing Foundation (CNCF) を卒業した最初のプロジェクトとなり、その成熟と幅広い普及が示されました。

🏢 2018 年 6 月 5 日、AWS Elastic Kubernetes Service (EKS) が US East (N. Virginia) と US West (Oregon) の各リージョンで利用できるようになりました。

📜 Node.js の作者である Ryan Dahl は、2018 年 6 月 6 日に Berlin で開かれた JSConf EU での講演 "10 Things I Regret About Node.js" で Deno を正式に発表しました。

🏢 2018 年 6 月 13 日、Microsoft は Azure Kubernetes Service (AKS) の一般提供を発表しました。

🏢 Slack は 2018 年 7 月 26 日に Atlassian から HipChat と Stride の知的財産を取得し、その後これらの利用者を Slack のプラットフォームへ移行させました。

☁️ Istio 1.0 は最初の 0.1 リリースから 1 年あまりを経た 2018 年 7 月 31 日にリリースされ、ユーザーが本番利用で依拠できる中核的な機能群が構築されたことを示すものとなりました。

☁ Kubernetes 上に構築されたオープンソースのサーバーレスかつイベント駆動型アプリケーションプラットフォームである Knative は、2018 年 7 月に Google によってリリースされ、VMware、IBM、Red Hat、SAP との緊密な連携のもとで開発されました。

☁ 2018 年 8 月 15 日、Google Cloud Run（フルマネージド）がアルファ版としてリリースされ、Knative 上に構築されたステートレスなリクエスト駆動型コンテナを実行するためのサーバーレス実行環境を提供しました。

📜 Go 1.11 は 2018 年 8 月 24 日にリリースされ、新しい依存関係管理の仕組みであるモジュールへの実験的なサポートを導入しました。これは後に Go プロジェクトの依存関係を管理する標準的な方法となりました。

🔐 TLS 1.3 は 2018 年 8 月に RFC 8446 で定義されました。

☁ 2018 年 9 月 10 日、Microsoft は VSTS のさらなる改称を発表し、今度は「Azure DevOps Services」となりました。

⚙ Podman は 2018 年 10 月に最初にリリースされました。

⚙ Microsoft は 2018 年、Windows 10 の October 2018 Update の一部として Windows Pseudo Console (ConPTY) を導入し、OpenSSH や WSL のような端末指向のソフトウェアをより適切に扱えるよう、Windows コンソールに標準化された PTY 相当のインターフェースを提供しました。

🏢 2018 年、Microsoft はオープンソースプロジェクトの基盤として最大のホストである GitHub を買収しました。買収は 2018 年 10 月 26 日に完了しました。

☁ 2018 年 11 月 26 日、AWS は Firecracker を発表しました。これは、コンテナの速度とリソース効率のよさに、仮想マシンのセキュリティと分離性を組み合わせることで、マルチテナントのコンテナベースのサービスを運用できるようにするオープンソースの仮想化技術です。Firecracker は AWS Lambda と AWS Fargate を支えるために AWS 社内で開発され、Linux KVM に基づくマイクロ仮想マシン (microVM) を用い、Rust で書かれています。

☁ 2018 年 11 月 28 日、AWS re:Invent で発表され、Envoy は Kubernetes と Prometheus に続いて Cloud Native Computing Foundation を卒業した 3 番目のプロジェクトとなりました。

☁ 2018 年 11 月 29 日、AWS は AWS Lambda のサポート言語として Ruby を発表しました。

☁ 2018 年 12 月にリリースされた Kubernetes 1.13 では、Container Storage Interface (CSI) と CoreDNS が一般提供となりました。

☁ GitHub Actions は 2018 年 10 月 16 日、GitHub Universe カンファレンスで

限定公開ベータとして発表され、ワークフローの自動化機能を GitHub のリポジトリに直接もたらしました。

🌐 FastAPI は Python で API を構築するための現代的で高性能な Web フレームワークで、Sebastián Ramírez が作り、2018 年 12 月 24 日にオープンソースとして最初に公開されました。

💻 Visual Studio Code は 2018 年の Stack Overflow Developer Survey で首位に立ち、回答者の間で最も人気のある開発環境となりました。

💻 Max Brunsfeld によって開発されたインクリメンタル構文解析ライブラリ兼パーサジェネレータである Tree-sitter は、当初 GitHub が Atom テキストエディタで使うために開発し、2018 年に初めてリリースされました。ソースコードから具象構文木を構築し、シンタックスハイライト、コードナビゲーション、静的解析に利用されます。

2019

⚙ 2019 年 1 月、Linuxbrew が Homebrew へ再統合され、Linux と Windows Subsystem for Linux に対するベータ版のサポートが Homebrew の機能に加わりました。

📊 Trino は、1 つ以上の異種のデータソースにまたがって分散した大規模なデータセットに問い合わせるために設計された、オープンソースの分散 SQL クエリエンジンです。2019 年 1 月、Presto を最初に作った Martin Traverso、Dain Sundstrom、David Phillips が Presto プロジェクトのフォークを作りました。当初は Presto という名前を保ったまま、元の PrestoDB プロジェクトと区別するために PrestoSQL という Web 上の名前を使っていました。

🧠 Generative Pre-trained Transformer 2 (GPT-2) は、2019 年 2 月に OpenAI が作ったオープンソースの人工知能です。GPT-2 はテキストを翻訳し、質問に答え、文章を要約し、ときに人間のものと見分けがつかない水準でテキストを生成しますが、長い文章を生成すると同じ内容の繰り返しになったり意味をなさなくなったりすることがあります。

☁ 2019 年 4 月 9 日、Google は Google Cloud Next 2019 で Cloud Run をベータ版として発表し、Knative serving API を使用してステートレスコンテナを実行するフルマネージドサーバーレスプラットフォームを導入しました。

💻 Visual Studio Code Remote Development 拡張機能パックは、開発者がコンテナ、リモートマシン、Windows Subsystem for Linux (WSL) を本格的な開発環境として使えるようにするもので、2019 年 5 月 2 日に初めて発表されました。

⚙ Microsoft は 2019 年 5 月 6 日に WSL 2 を発表し、Windows 10 バージョン 2004 で提供されました。

🏢 2019 年 5 月 7 日、Google は Kotlin プログラミング言語が Android アプリ開発者にとって推奨の言語になったと発表しました。

☁ Argo CD 1.0 は 2019 年 5 月 16 日にリリースされ、指定された対象環境において望ましいアプリケーション状態のデプロイを自動化するツールとして最初の安定版となりました。

⚙ 2019 年 6 月 3 日の WWDC で、Apple は iPad 上で動作する iOS の派生版を「iPadOS」として別ブランド化すると発表し、iPad というプラットフォームを iPhone とは異なる方向へ発展させる意向を示しました。

🌐 Tailwind CSS はオープンソースの CSS フレームワークです。Bootstrap のような他のフレームワークと違い、ボタンやテーブルといった要素向けにあらかじめ定義されたクラス群を提供しません。その代わりに、組み合わせて各要素の見た目を整えられる「ユーティリティ」CSS クラスの一覧を用意します。最初のリリースは 2019 年 5 月 13 日です。

💻 2019 年 9 月 11 日にリリースされた cURL 7.66 は、HTTP/3（したがって QUIC）をサポートしています。

⚙ iPadOS 13.1 は 2019 年 9 月 24 日、新たにブランド化された iPad 向けオペレーティングシステムの最初のバージョンとしてリリースされ、ウィジェットを備えた新しいホーム画面、Split View と Slide Over による改善されたマルチタスク、デスクトップ並みの Safari でのブラウジング、マウスとトラックパッドのサポートを特徴としていました。

💻 2019 年 10 月 7 日にリリースされた macOS Catalina は、Zsh をデフォルトのログインシェルとして採用し、Mac OS X Panther（2003 年）以来デフォルトであった GPLv2 ライセンスの Bash を置き換えました。Apple は 2019 年 6 月 3 日の WWDC でこの変更を発表し、当時の最新版である Zsh 5.7.1 を同梱しました。

☁ GitHub Actions は 2019 年 11 月 13 日に一般提供となり、DevOps のための完全な CI/CD 機能とネイティブなパッケージ管理を備えた、コミュニティ主導のソフトウェア自動化のあり方を提供しました。

☁ 2019 年 11 月 14 日、Google Cloud Run（フルマネージド）が一般提供（GA）となり、月間稼働率 99.95% の SLA を備えた Knative ベースのサーバーレスコンピュートプラットフォームを提供しました。

⚙ 2019 年 11 月 25 日、RIPE NCC は IPv4 アドレスを完全に使い果たしたと発表し、IPv6 への移行の必要性が改めて浮き彫りになりました。

🌐 WebAssembly は 2019 年 12 月 5 日に World Wide Web Consortium の勧告となりました。

🏢 2019 年、JS Foundation と Node.js Foundation が合併して OpenJS Foundation が発足しました。

📜 Rust は 2019 年、バージョン 1.39.0 で async/await のサポートを追加しました。

🏢 The Pragmatic Programmer の 20 周年版は 2019 年にリリースされ、ソフトウェア設計の根幹をなすメタ原則として「ETC」(Easier To Change) を打ち出しました。これは、優れた設計の本質はすべて、変更しやすくすることにあると説くものです。

タイムライン - 2020-24

2020

🌐 現代的な Web アプリケーションのための信頼性の高いエンドツーエンドテストのフレームワークである Playwright は、2020 年 1 月 31 日に Microsoft によってオープンソースとして最初に公開されました。Google で以前 Puppeteer に携わっていたエンジニアたちが作ったもので、単一の API のもとでクロスブラウザ自動化を Chromium、Firefox、WebKit へと広げました。

💻 PowerShell 7.0 は 2020 年 3 月 4 日にリリースされ、「Core」という接尾辞をやめて、Windows PowerShell 5.1 と PowerShell Core 6.x の両方の後継となり、Windows のモジュールとの互換性も改善されました。

☁ Istio 1.5 は 2020 年 3 月 5 日にリリースされ、「モノリスを受け入れる」という方針でコントロールプレーンを Istiod という単一の新しいバイナリに統合し、Istio のインストール、実行、アップグレードの体験を大幅に簡素にしました。

📜 Hyrum の法則は、「API の利用者が十分に多ければ、契約で何を約束しているかは問題にならない。システムの観測可能な挙動はすべて、誰かに依存されることになる」という観察で、Google のエンジニアである Hyrum Wright が言葉にし、同僚の Titus Winters が 2020 年 3 月刊行の著書 Software Engineering at Google の中で名付けました。

☁ 2020 年 4 月 7 日、CNCF の技術監督委員会 (TOC) は、Kubernetes 上でジョブとアプリケーションを実行・管理するための Kubernetes ネイティブなツール群である Argo を、インキュベーションレベルのホストプロジェクトとして受け入れることを決議しました。

🌐 Evan You が作ったフロントエンドのビルドツールである Vite は、2020 年 4 月 20 日に初めて発表されました。ネイティブの ES Modules を活用して開発中のバンドルをなくし、ほぼ瞬時のホットモジュールリプレースメントを実現しています。フレームワークに依存しないよう設計し直された Vite 2.0 は 2021 年 2 月にリリースされました。

💻 GitHub Codespaces は 2020 年 5 月 6 日、GitHub Satellite 2020 で限定公開ベータとして発表され、Visual Studio Code を基盤とする本格的なクラウド上の開発環境を GitHub の中で直接提供しました。

🔐 GitHub Advanced Security は 2020 年 5 月の GitHub Satellite 2020 で発表され、セキュリティ対策を開発者のワークフローに直接組み込んで自動化できるよう、コードスキャンをベータ版で、プライベートリポジトリ向けのシークレットスキャンを限定公開ベータで導入しました。

📜 単一の実行ファイルとして設計された、既定で安全な JavaScript と TypeScript のランタイムである Deno v1.0.0 は、2020 年 5 月 13 日に正式にリリースされました。

🌐 2020 年 5 月、Discord は主要なドメインを discordapp.com から discord.com へ移し、ゲーマーだけでなくあらゆるコミュニティの場になるという、より広い目標を示しました。

🧠 Generative Pre-trained Transformer 3 (GPT-3) は 2020 年 6 月に公開された自己回帰型の言語モデルで、深層学習を用いて人間が書いたようなテキストを生成します。最初のテキストをプロンプトとして与えると、その続きとなるテキストを生成します。

⚙️ Apple は 2020 年 6 月 22 日の WWDC で、Mac を Intel のプロセッサから自社製の Apple Silicon へ移行する計画を発表しました。これは Mac の歴史で 3 度目となる大きなプロセッサアーキテクチャの移行です。

📜 Rust for Linux プロジェクトは 2020 年 7 月に Linux カーネルのメーリングリストで発表され、Rust のメモリ安全性を活かしてカーネルドライバを書く際のバグを減らすことを目標としています。

💻 Visual Studio Codespaces は 2020 年 9 月 4 日に GitHub Codespaces へ統合されました。Microsoft は、2 つの異なる体験が混乱を招いていること、そして Azure ベースの Visual Studio Codespaces のポータルを 2021 年 2 月 17 日に終了することを併せて発表しました。

🔐 GitHub のコードスキャンは、ベータ版の公開から 4 か月後の 2020 年 9 月に一般提供となりました。それまでに 12,000 を超えるリポジトリをスキャンし、リモートコード実行、SQL インジェクション、クロスサイトスクリプティングの欠陥を含む 20,000 件超のセキュリティ上の問題を検出していました。

⚙️ 2020 年 9 月 21 日、GitHub Releases 経由のボトルタップ (バイナリパッケージのリポジトリ) をサポートした Homebrew バージョン 2.5.2 がリリースされました。

🏢 2020 年 10 月 6 日、Google は G Suite を Google Workspace へ改称し、コミュニケーションとコラボレーションのアプリ群に、新しい統合された利用体験と統一され

たブランドをもたらしました。

🌐 Jakarta Server Faces 3.0.0 は 2020 年 10 月 28 日にリリースされ、Jakarta EE への移行の一環としてパッケージ名が javax から jakarta へ変更されました。

⚙ 最初の Apple Silicon 搭載 Mac である MacBook Air、13 インチ MacBook Pro、M1 チップ搭載 Mac mini は 2020 年 11 月 10 日に発表され、2020 年 11 月 17 日に出荷が始まりました。Intel を搭載した従来機に比べて、性能と電力効率が大きく向上しました。

🏢 Microsoft は 2020 年 11 月 10 日に .NET 5.0 をリリースしました。「Core」というブランドは外され、Windows 固有の製品として残る .NET Framework との混同を避けるためにバージョン 4.0 は飛ばされました。.NET Framework に関する特許上の懸念にも対応しています。

⚙ macOS 11 Big Sur は 2020 年 11 月 12 日にリリースされ、2001 年の最初の Mac OS X 以来、メジャーバージョン番号が 10 を超えた最初のバージョンとなりました。Apple Silicon (M1) 搭載 Mac をサポートした最初の macOS でもあり、iOS に着想を得て再設計されたユーザーインターフェースを特徴としています。

🏢「The Scrum Guide」の直近の大きな改訂は、2020 年 11 月 18 日に Ken Schwaber と Jeff Sutherland によって公開されました。

📜 Ruby 3.0.0 は 2020 年のクリスマスにリリースされました。

📊 2020 年 12 月、PrestoSQL は Trino へ改称されました。

📊 2020 年、LAMBDA 関数の導入により Microsoft Excel はチューリング完全なプログラミング言語となり、利用者は Excel 自身の数式言語で書いた新しい関数を定義できるようになりました。

🌐 2020 年にリリースされた Tomcat 10.0 は Servlet 5.0 と JSP 3.0 を実装しており、Java EE が Jakarta EE として Eclipse Foundation へ移管されたことを受けて、パッケージの名前空間を javax. から jakarta. へ移行した最初のバージョンでした。

2021

📊 2021 年 1 月、Elastic は Elasticsearch と Kibana のライセンスを Apache License 2.0 から、Server Side Public License と Elastic License のデュアルライセンスへ変更すると発表しました。Amazon Web Services が十分な協力なしに Elasticsearch と Kibana をサービスとして提供していたことへの対応でした。

📊 2021 年 1 月 21 日、Amazon Web Services は Apache License 2.0 の下での Elasticsearch と Kibana のフォークを発表し、商標上の問題から 2021 年 4 月にこれを OpenSearch へ改称しました。

🏢 2021 年 2 月 8 日、Rust Foundation の設立が、創設企業である 5 社 (AWS、Huawei、Google、Microsoft、Mozilla) によって発表されました。

☁️ Google Kubernetes Engine (GKE) Autopilot は 2021 年 2 月に登場し、クラスタをマネージドで運用するモードを提供しました。

⚙️ IEEE 802.11ax-2021 規格 (Wi-Fi 6) は 2021 年 2 月 9 日に承認されました。

📶 2021 年 3 月、Lucene はロゴを変更し、Apache Solr は Lucene から独立して再びトップレベルの Apache プロジェクトになりました。

🔐 プライベートリポジトリ向けのシークレットスキャンは、2021 年 3 月 31 日に GitHub Enterprise Cloud のすべての GitHub Advanced Security 利用者に向けて一般提供となりました。対応パターンの提供パートナーは 24 社から 37 社へ拡大し、ベータ期間中に 5,000 件を超えるシークレットの発見と失効を支援していました。

🌐 2021 年 5 月、IETF は RFC 9000 で QUIC を標準化しました。

🌐 Firefox での QUIC のサポートは 2021 年 5 月に実現しました。

🧠 統合されたマネージドの機械学習プラットフォームである Vertex AI は、2021 年 5 月の Google I/O で発表されました。

☁️ あらゆるデータベースを賢いスプレッドシートに変えるオープンソースのノーコードプラットフォームである NocoDB は、2021 年 5 月 26 日に正式に公開されました。

☁️ Terraform 1.0 は 2021 年 6 月 8 日にリリースされました。

☁️ JavaScript、TypeScript、WebAssembly をエッジで実行できるようにする分散型の HTTP サービスである Deno Deploy は、2021 年 6 月 23 日に公開されました。

🧠 GitHub Copilot は、Visual Studio Code、Visual Studio、Neovim、JetBrains の統合開発環境 (IDE) の利用者をコードの自動補完によって支援するために GitHub と OpenAI が開発したクラウドベースの人工知能ツールで、2021 年 6 月 29 日にテクニカルプレビューとして発表されました。

💻 2021 年 7 月 2 日の Neovim 0.5 リリースにより、Language Server Protocol (LSP)、Tree-sitter、およびより完全な Lua サポートが標準で組み込まれました。

🏢 Salesforce は 2021 年 7 月 21 日、約 277 億ドルで Slack の買収を正式に完了し、ソフトウェア業界で最大級の買収の 1 つとなりました。

🏢 Microsoft は 2021 年 7 月 31 日にホスト型の Skype for Business Online のサービスを終了し、コミュニケーションとコラボレーションの機能を完全に Microsoft Teams

へ移しました。

🧠 OpenAI Codex は、自然言語を解釈してそれに応じたコードを生成する、OpenAI が開発した人工知能モデルで、GitHub Copilot の基盤となっています。OpenAI の GPT-3 モデルを受け継ぎ、プログラミング用途に微調整されたもので、2021 年 8 月 10 日に公開されました。

💻 Visual Studio Code を基盤とするクラウド上の開発環境である GitHub Codespaces は、2021 年 8 月 11 日に GitHub Team と Enterprise Cloud の利用者向けに一般提供となりました。

🏢 eBPF Foundation は 2021 年 8 月に Linux Foundation の下で発足しました。創設メンバーには Google、Facebook、Isovalent、Microsoft、Netflix が名を連ね、業界における eBPF の重要性の高まりを映し出していました。

🔐 OpenSSL 3.0 は 2021 年 9 月 7 日にリリースされました。

⚙️ Windows 11 は 2021 年 10 月 5 日に正式にリリースされ、中央に配置されたタスクバーとスタートメニュー、改善されたウィンドウの整列（「Snap Layouts」）、強化されたセキュリティ要件への注力といった、大きな視覚的再設計を特徴としています。

💻 2021 年 10 月 27 日、GitHub は GitHub Copilot の Neovim プラグインを公開リポジトリとして公開しました。

☁️ GitHub Actions は 2021 年 10 月 27 日に OpenID Connect (OIDC) のサポートを追加し、デプロイのたびに自動で更新される短命なトークンによる安全なクラウドへのデプロイを可能にしました。

🐛 Log4Shell は、広く使われている Java のロギングフレームワーク Log4j に存在した任意コード実行のゼロデイ脆弱性で、2013 年以来気づかれないまま存在していました。Alibaba Cloud のセキュリティチームの Chen Zhaojun によって、2021 年 11 月 24 日に Apache Software Foundation へ非公開で報告されました。

☁️ GitHub Actions の再利用可能なワークフローは 2021 年 11 月 24 日に一般提供となり、ワークフロー全体をアクションと同じように再利用して重複を減らせるようになりました。

☁️ Kubernetes 上に構築されたフルマネージドのサーバーレスコンテナサービスである Azure Container Apps は、2021 年 11 月 2 日にパブリックプレビューとして登場しました。

☁️ Knative は 2021 年 11 月にバージョン 1.0 に到達し、すべてのリポジトリがコミュニティによって安定版かつ商用利用に適したものと位置付けられました。

🏢 Grafana は 2021 年に k6 を買収しました。

🌐 Ruby on Rails 7.0 は 2021 年 12 月 15 日にリリースされ、既定で Node.js と Webpack を import maps に置き換え、Hotwire と呼ばれる一連のライブラリを導入しました。

📜 Vitest は 2021 年 12 月、JavaScript のための Vite ネイティブなユニットテストフレームワークとして最初にリリースされました。Vite の変換パイプラインと設定をそのまま活かし、Vite ベースのプロジェクトでほぼ瞬時にテストを開始できます。Vitest 1.0 は 2023 年 12 月に本番利用可能な水準に達しました。

🧠 LaMDA (Language Model for Dialogue Applications) は Google が開発した会話向け大規模言語モデルのファミリーです。もともと 2020 年に Meena として登場し、2021 年の Google I/O 基調講演で発表された第 1 世代の LaMDA へと発展し、その翌年に第 2 世代が続きました。

2022

☁️ 2022 年 3 月 2 日、Knative は Cloud Native Computing Foundation (CNCF) のインキュベーティングプロジェクトとして受理されました。これは Google が 2021 年 11 月に申請したことを受けたものです。

📜 Go 1.18 は 2022 年 3 月 15 日にリリースされ、ジェネリックプログラミング (型パラメータ) のサポートを追加しました。これは Go 1.0 以来の最も重要な言語変更であり、コミュニティから長く求められていた機能です。

⚙️ Chocolatey 1.0.0 は 2022 年 3 月 18 日に正式にリリースされ、10 年以上の開発を経て、バージョンの安定性と企業での利用への備えという点で大きな節目となりました。

🧠 PaLM (Pathways Language Model) は Google AI が開発した 5,400 億パラメータのトランスフォーマーベースの大規模言語モデルで、2022 年 4 月に初めて発表され、Google が 2023 年 3 月に公開 API を提供するまでは非公開のままでした。

🧠 2022 年 5 月 11 日、Google は 2022 年の Google I/O 基調講演で、LaMDA の後継となる LaMDA 2 を発表しました。

🏢 Meta は 2022 年 5 月に Jest を OpenJS Foundation へ移管しました。これは、このフレームワークが Meta 社内での利用を超えて、Node.js や jQuery、webpack と並んでホストされるコミュニティ主導のプロジェクトへと成長したことを示すものでした。

☁️ 2022 年 5 月にリリースされた Kubernetes 1.24 は「dockershim」を削除し、Container Runtime Interface (CRI) を採る形で、コンテナランタイムとしての Docker の組み込みサポートを正式に終了しました。

☁ Kubernetes 上に構築されたフルマネージドのサーバーレスコンテナサービスである Azure Container Apps は、2022 年 5 月 24 日に一般提供となりました。

🧠 2022 年 6 月 21 日、GitHub は Copilot が「テクニカルプレビュー」を終え、個人の開発者向けのサブスクリプション型サービスとして利用できるようになったと発表しました。

🌐 Deno のためのフルスタック Web フレームワークである Fresh 1.0 は、設定不要で高い性能を実現する「islands」アーキテクチャを採用し、2022 年 6 月 28 日にリリースされました。

🌐 Zig で書かれ JavaScriptCore を基盤とする高速なオールインワンの JavaScript ランタイム兼ツールキットである Bun は、2022 年 7 月 5 日のバージョン 0.1.0 のリリースで公開ベータに入りました。

☁ 2022 年 8 月 25 日、Heroku は不正利用と濫用を主な理由として無料プランの廃止を発表し、その廃止は 2022 年 11 月 28 日に予定されました。

⚙ systemd のサポートは 2022 年 9 月 21 日に WSL で利用できるようになりました。

☁ Istio は 2022 年 9 月 30 日、インキュベーションの成熟度レベルで Cloud Native Computing Foundation に受け入れられました。

🧠 ReAct (Synergizing Reasoning and Acting in Language Models) は、大規模言語モデル (LLM) による推論と行動を組み合わせてタスクを遂行する汎用的な枠組みで、2022 年 10 月 6 日に Princeton University と Google Research の研究者によって arXiv へ初めて投稿されました。

🏢 2022 年 10 月 13 日、Microsoft は「Microsoft Office」ブランドを段階的に終了して「Microsoft 365」へ寄せると発表し、その後 Office.com と Office のモバイルアプリをこの方針に沿って改称しました。

⚙ Linux 6.0 は 2022 年 10 月 2 日にリリースされ、現代的な CPU アーキテクチャでの性能改善に重点を置くとともに、カーネルでの Rust サポートに向けた最初の下地を含んでいました。

📜 2022 年 10 月、Rust for Linux の実装を受け入れるプルリクエストが Torvalds によって承認されました。

🧠 大規模言語モデル (LLM) を使うアプリケーションの作成を容易にするために設計されたオープンソースのフレームワークである LangChain は、2022 年 10 月に Harrison Chase によって公開されました。

💻 GitHub Codespaces は 2022 年 11 月 9 日にすべての個人利用者が使えるよう

になり、無料枠には毎月の無償利用量も含まれました。

☁️ 2022 年 12 月 6 日、Cloud Native Computing Foundation は Argo が卒業 (Graduated) したことを発表し、Argo は Kubernetes、Prometheus、Envoy と並ぶトップレベルのプロジェクトとなりました。

📜 OCaml 5.0 は 2022 年 12 月 16 日にリリースされ、共有メモリによる並列処理と、ロックフリーな並行処理のためのエフェクトハンドラをサポートするために、ランタイムを全面的に書き直したことを特徴としています。

⚙️ Linux 6.1 は 2022 年 12 月にリリースされ、eBPF プログラムを Rust で書けるようになるなど、安全で高性能なカーネルレベルの可観測性とネットワーク処理に向けたエコシステムがさらに広がりました。

🧠 ChatGPT (Chat Generative Pre-trained Transformer) は OpenAI が開発した人工知能のチャットボットで、2022 年 11 月に公開されました。OpenAI の GPT-3.5 と GPT-4 という大規模言語モデルのファミリーの上に構築され、教師あり学習と強化学習の両方の手法で微調整されています。

📜 npm: 指定子による npm パッケージのサポートは、2022 年 11 月 14 日の Deno v1.28 で安定版となり、このランタイムから既存の膨大な JavaScript エコシステムを利用できるようになりました。

🌐 HTTP/2 の後継である HTTP/3 は 2022 年に公開されました。

2023

🧠 2023 年 2 月 7 日、Microsoft は「新しい Bing」として売り出したチャットボット機能の追加を含む、Bing の大規模な刷新を発表しました。

🧠 Azure OpenAI Service は 2023 年 1 月 17 日に一般提供となり、OpenAI の大規模言語モデルを企業が利用できるようにしました。

🧠 Llama (Large Language Model Meta AI の頭字語で、以前は LLaMA と表記されていました) は、2023 年 2 月から Meta AI が公開している自己回帰型の大規模言語モデル (LLM) のファミリーです。

🧠 Notion ワークスペースに統合された AI アシスタントである Notion AI は、2022 年 11 月に始まったプライベートアルファ期間を経て、2023 年 2 月 22 日に一般提供となりました。

🧠 Generative Pre-trained Transformer 4 (GPT-4) は、OpenAI が GPT 基盤モデルシリーズの 4 番目として作ったマルチモーダルな大規模言語モデルで、2023 年 3 月 14 日に公開され、有料の ChatGPT Plus、OpenAI の API、無料の Microsoft Copilot を通じて一般に利用できるようになりました。

🧠 Yohei Nakajima が作ったタスク駆動型の自律エージェントである BabyAGI は、2023 年 3 月 28 日に Twitter で公開されました。GPT-4、Pinecone、LangChain を用いた 105 行の Python スクリプトとして書かれ、自律的なタスクの計画と実行を実演したことで急速に広まり、16,000 を超える GitHub のスターを集めました。

🧠 Significant Gravitas Ltd. の Toran Bruce Richards が作ったオープンソースの自律型 AI エージェントである AutoGPT は、2023 年 3 月 30 日に GitHub で公開されました。GPT-4 の自律的な能力を示した最初のアプリケーションの 1 つで、Web ブラウジング、ファイル管理、API 呼び出しといったツールを使って利用者の定めた目標をサブタスクへ分解し、数週間で 10 万を超えるスターを集めて GitHub 史上最も急成長したオープンソースプロジェクトとなりました。

🧠 Amazon Bedrock は、主要な AI スタートアップ各社と Amazon の基盤モデル (FM) を API 経由で利用できるようにするフルマネージドのサービスで、2023 年 4 月 13 日に限定プレビューとして初めて発表されました。

🧠 Bard は Google が開発した会話型の人工知能チャットボットで、LaMDA という大規模言語モデルのファミリーに基づいています。OpenAI の ChatGPT の台頭を受けて開発され、2023 年 3 月に限定的な形で公開されましたが、反応は芳しくありませんでした。

🧠 Anthropic が開発した最初の大規模言語モデルである Claude 1 は 2023 年 3 月に公開され、同社の会話型 AI モデルのファミリーの礎を築きました。

⚙️ Chocolatey 2.0.0 は 2023 年 5 月 31 日にリリースされ、基盤となる NuGet のクライアントライブラリを大きく更新するとともに、Semantic Versioning (SemVer) 2.0.0 のサポートを導入しました。

☁️ Istio は 2023 年 7 月 12 日に Cloud Native Computing Foundation 内で卒業の成熟度レベルへ移行しました。インキュベーター入りから 1 年足らずでの卒業でした。

🔐 Azure Active Directory (Azure AD) は 2023 年 7 月 15 日に正式に Microsoft Entra ID へ改称されました。

🧠 LLM ベースのマルチエージェント連携に効率的な人間のワークフローを組み込む画期的なメタプログラミングフレームワークである MetaGPT は、2023 年 8 月 1 日に発表された論文で紹介されました。Standardized Operating Procedures (SOP) をプロンプトシーケンスにエンコードし、アセンブリラインのパラダイムを用いて多様な役割をエージェントに割り当てることで、複雑なタスクを多くのエージェントが協調して取り組むサブタスクへと分解します。

📜 Go 1.21 は 2023 年 8 月 8 日にリリースされ、組み込みの min、max、clear 関数、新しい構造化ロギングのパッケージ log/slog、そして正式なツールチェーン管理を導

入しました。

☁️ HashiCorp は 2023 年 8 月 10 日に Terraform をはじめとする製品のライセンスを Mozilla Public License v2.0 から競合利用を禁じるソースアベイラブルな Business Source License (BUSL) v1.1 へ変更し、これが OpenTF マニフェストを生むコミュニティの反発を招きました。

☁️ ローカルファーストで DAG に基づくワークフローエンジンである Dagu は、Apache Airflow のようなより複雑なオーケストレーターに代わる「ゼロオプス」の選択肢として設計され、2023 年 8 月 11 日に GitHub でオープンソースとして最初に公開されました。

☁️ HashiCorp が Terraform を非オープンソースのライセンスへ移すと発表したことを受けて、OpenTF プロジェクト (後に OpenTofu へ改称) が 2023 年 8 月 25 日に正式にフォークされました。同プロジェクトは 2023 年 9 月 20 日に Linux Foundation に加わりました。

🌐 統合されたバンドラとテストランナーを備え、このランタイムが安定した本番利用可能なバージョンへ移行したことを示す Bun 1.0 は、2023 年 9 月 8 日に正式にリリースされました。

🧠 Amazon Bedrock は 2023 年 9 月 28 日に一般提供 (GA) となり、AI21 Labs、Anthropic、Cohere、Stability AI、Amazon の基盤モデルのサポートとともに提供が始まりました。

🧠 2023 年 11 月、GitHub Copilot Chat は OpenAI の GPT-4 モデルを使うように更新されました。

🧠 Google Gemini は、LaMDA と PaLM 2 の後継として Google DeepMind が開発したマルチモーダルな大規模言語モデルのファミリーで、Gemini Ultra、Gemini Pro、Gemini Flash、Gemini Nano から成ります。2023 年 12 月 6 日に、OpenAI の GPT-4 に真っ向から対抗するものとして発表されました。

📜 Mojo は現在開発が進められている Python ファミリーのプログラミング言語で、Jupyter ノートブック経由でブラウザから、また Linux と macOS ではローカルで利用でき、2023 年に初めて登場しました。

🧠 LocalAI は OpenAI の代替となるフリーかつオープンソースのソフトウェアで、OpenAI API の仕様と互換性のある REST API としてそのまま置き換えて使え、ローカルでの推論を可能にします。2023 年に初めて登場しました。

💻 Anysphere が開発した Visual Studio Code (VS Code) の AI ネイティブなフォークである Cursor は、2023 年に一般公開されました。

2024

🧠 2024 年 2 月、Google は Gemini 1.5 を限定的な形で公開し、1.0 Ultra よりも強力で能力の高いモデルと位置付けました。

🧠 LangGraph は LangChain の上に構築されたライブラリで、循環のあるグラフを作ることによって、LLM を用いた状態を持つマルチアクターのアプリケーションを作れるようにします。2024 年 1 月 17 日に発表されました。

☁ OpenTofu は 2024 年 1 月 10 日の 1.6.0 リリースで一般提供 (GA) となり、最初の安定した本番利用可能なバージョンとなりました。

🧠 2024 年 2 月 8 日、Google は Google One の AI プレミアムプランを導入し、Gemini Advanced と、Gmail、ドキュメント、スライド、スプレッドシート、Meet の中の Gemini を利用できるようにしました。

🧠 2024 年 2 月 27 日、GitHub Copilot Enterprise が月額 39 ドルで一般提供となり、組織内部のコードベースやナレッジベースを参照できる機能で Business プランを拡張しました。

🧠「世界初の完全に自律したソフトウェアエンジニアの AI」と銘打たれた Devin は、2024 年 3 月 12 日に Cognition AI から正式に公開されました。

☁ Envoy Gateway 1.0 は 2024 年 3 月 13 日にリリースされ、Kubernetes Gateway API の高性能で実用的な実装として一般提供に達しました。

🔐 GitHub は 2024 年 3 月 20 日、GitHub Advanced Security 向けの Copilot Autofix を公開ベータとして発表しました。コード中の脆弱性を分析し、開発者がそれを修正する助けとなる提案を示す、AI による修復機能です。

🧠 大規模言語モデルの Claude 3 ファミリーは 2024 年 3 月 4 日に Anthropic から発表され、Claude 3 Opus (最高の能力)、Claude 3 Sonnet (性能のバランス)、Claude 3 Haiku (低遅延・低コスト) という 3 つのモデル階層が導入されました。Claude 3 ファミリーは Anthropic にとって初めてのマルチモーダルなモデルであり、3 つの階層すべてに視覚能力 (画像の理解) が加わりました。Opus と Sonnet は 2024 年 3 月 4 日に、Haiku は 2024 年 3 月 13 日に一般提供となりました。

🌐 Windows への安定したサポートと、よく使われる処理の大幅な性能改善をもたらした Bun 1.1 は、2024 年 4 月 1 日にリリースされました。

🧠 2024 年 4 月 29 日、GitHub は Copilot Workspace をテクニカルプレビューとして公開しました。自然言語で書かれた GitHub の issue を、仕様、計画、コードの変更へと変換する、Copilot を前提とした開発環境です。

🧠 GPT-4o (GPT-4 Omni) は OpenAI が設計した多言語対応でマルチモーダルな生成事

前学習トランスフォーマーで、2024 年 5 月 13 日にライブ配信のデモンストレーションの中で OpenAI の CTO である Mira Murati によって発表され、同日に公開されました。

🧠 AI を活用したアプリケーションを構築・デプロイするための Google のオープンソースフレームワークである Firebase Genkit は、2024 年 5 月 14 日の Google I/O 2024 の基調講演で初めて紹介されました。

🔐 CodeQL のコードスキャンによる警告に対する Copilot Autofix は 2024 年 8 月 14 日に一般提供となり、手作業なら 1.5 時間かかるところを中央値 28 分と、3 倍以上速くコードの脆弱性を修正できるようになりました。

📶 2024 年 8 月 29 日、Elastic は Elasticsearch の 3 つ目のライセンスの選択肢として GNU Affero General Public License を追加し、このソフトウェアが「Open Source, Again」であると宣言しました。

🧩 Unicode 16.0 は 2024 年 9 月 10 日にリリースされ、7 つの新しい用字系と 7 つの新しい絵文字を含む 5,185 の文字が追加されて、収録文字数は合計 154,998 となりました。

🌐 Deno v2.0.0 は 2024 年 10 月 9 日に正式にリリースされ、Node.js および npm との完全な後方互換性と、企業での利用に向けた長期サポート (LTS) に重点が置かれました。

🧠 2024 年 9 月 16 日、Microsoft は「Microsoft 365 Copilot Wave 2」を発表し、Copilot Studio または SharePoint の中で直接構築できる専門特化した AI アシスタント「Copilot agents」を導入しました。これらのエージェントは 2024 年 10 月 17 日に一般提供となりました。

📶 PostgreSQL 17 は 2024 年 9 月 26 日にリリースされ、バキューム処理と I/O の性能を大きく高めるとともに、JSON の機能を拡充しました。

⚙️ IEEE 802.11be-2024 追補 (Wi-Fi 7) は 2024 年 9 月 26 日に IEEE SA Standards Board によって承認されました。

🧠 元の BabyAGI フレームワークを大きく書き直した BabyAGI v2 は、Yohei Nakajima によって 2024 年 9 月に公開されました。関数とメタデータをデータベースに保存する functionz フレームワークを導入し、最初の公開以降の GPT-4o、Claude のツール利用、構造化出力といった進展を取り入れています。

🧠 Anthropic は 2024 年 10 月 22 日、Claude 3.5 Sonnet 向けに Computer Use 機能を公開ベータとして導入し、スクリーンショットを解釈してマウスとキーボードの入力を模倣することで、モデルがデスクトップ環境を操作できるようにしました。

🧠 Model Context Protocol (MCP) は、大規模言語モデルのような人工知能システムが外部のツールやシステム、データソースと連携しデータを共有する方法を標準化するため

に、2024 年 11 月 25 日に Anthropic が公開したオープンな標準かつオープンソースのフレームワークです。

🧠 OpenAI o3 は OpenAI が OpenAI o1 の後継として開発した内省的な生成事前学習トランスフォーマーモデルで、順を追った論理的な推論を要する問いに対してより多くの思考時間を割くよう設計されており、2024 年 12 月 20 日に発表されました。

🧠 Devin は 2024 年 12 月に一般提供となり、席数の制限なしに月額 500 ドルからでエンジニアリングチームに提供され、あわせて Slack との連携、IDE 拡張機能、API も用意されました。

⚙️ 2024 年、Single UNIX Specification Issue 8 は擬似端末についてより包摂的な用語を採用し、従来の「master」と「slave」という呼称をそれぞれ正式に「manager」と「subsidiary」へ置き換えました。

📜 Elixir v1.18 は 2024 年 12 月 19 日にリリースされ、JSON の組み込みサポートと、関数呼び出しに対する型検査の最初の試みを特徴としています。

タイムライン - 2025-現在

2025

🧠 Anthropic の Computer Use に対する重要な API アップデートが 2025 年 1 月 24 日に公開され (ヘッダーは anthropic-beta: computer-use-2025-01-24)、複雑な UI 環境を操作する際のモデルの正確さと信頼性が向上しました。

🧠 2025 年 1 月 30 日、Google は Gemini 2.0 Flash を新しい既定のモデルとして公開し、Gemini 1.5 Flash も引き続き利用できるようにしました。続いて 2025 年 2 月 5 日に Gemini 2.0 Pro が公開されました。

🧠 GitHub は 2025 年 2 月 6 日、開発者に向けた新しい自律的なワークフローと生産性の向上を打ち出した大きな発表の一環として、Copilot Agent Mode を正式に導入しました。続いて 2025 年 2 月 24 日、GitHub は続報のブログ記事で VS Code の Stable と Insiders での提供を明らかにし、アーリーアクセスを超えた最初の本格的な展開となりました。

💻 fish 4.0.0 は 2025 年 2 月 27 日にリリースされ、それまでの C++ 実装から Rust へと書き直され、このシェルの歴史上最大のアーキテクチャ変更となりました。

🧠 2025 年 2 月、Amazon は生成 AI を搭載した音声アシスタントの最新版「Alexa+」を発表し、すべての Prime 会員に無料提供されることになりました。同発表では、Anthropic の Claude モデルが Alexa+ に組み込まれていることも明らかにされました。

🧠 Devin 2.0 は 2025 年 4 月 3 日にリリースされ、複数の Devin を並列に動かせる新しいエージェント前提の IDE 体験を導入するとともに、開始価格を月 500 ドルから月 20 ドルへ引き下げました。

🔐 GitHub は 2025 年 4 月 1 日に GitHub Advanced Security を分割し、GitHub Secret Protection（アクティブなコミッター 1 人あたり月 19 ドル）と GitHub Code Security（アクティブなコミッター 1 人あたり月 30 ドル）という 2 つの独立した製品として提供を始め、GitHub Team プランの利用者にも門戸を広げました。

☁️ Cloud Native Buildpacks やネイティブの OpenTelemetry といったクラウドネイティブ技術の上に構築された次世代の Heroku プラットフォームである Heroku Fir は、2025 年 4 月 14 日に一般提供となりました。

🧠 GitHub は 2025 年 4 月 4 日、Agent Mode をすべての利用者へ展開すると正式に発表しました。これには Model Context Protocol (MCP) のサポートと、Copilot のコードレビューエージェントの導入が含まれていました。

☁️ 2025 年 5 月 6 日、9 年にわたる改良と数え切れないコミュニティからの貢献を経て、Grafana k6 v1.0.0 がリリースされました。

🧠 2025 年 5 月 7 日の Config 2025 で、Figma は Figma Sites、Figma Buzz、Figma Draw、Figma Make といった AI を活用したツール群を発表しました。Figma Make はプロンプトから動作するアプリを作り出す AI ツールで、Anthropic の Claude 3.7 Sonnet モデルを基盤としています。

🧠 2025 年 5 月 17 日、GitHub は GitHub Copilot コーディングエージェントを発表しました。これはより自律的に動作するモードで、issue を割り当てられ、開発者に代わってプルリクエストを作成できます。

🧠 2025 年 5 月 19 日の Microsoft Build 2025 で、GitHub は GitHub Copilot に非同期のコーディングエージェントが加わったことを発表しました。このエージェントは GitHub に直接組み込まれ、VS Code からも利用でき、GitHub Actions を基盤とする安全な開発環境を立ち上げることで、低〜中程度の複雑さのタスクを得意とします。

💻 Anthropic の AI 搭載コマンドラインコーディングアシスタントである Claude Code は、2025 年 2 月 24 日に Claude 3.7 Sonnet とともにベータ研究プレビューとして初めて導入されました。その後、2025 年 5 月 22 日に一般提供となりました。

🧠 2025 年 5 月 22 日の Claude 4 (Opus と Sonnet) のリリースにより、コンピュータ操作の機能がモデルの中核となるエージェント的ワークフローへさらに統合され、ピクセル数の把握が改善されて遅延も短縮されました。

🧠 2025 年 6 月 17 日、Google は 2.5 Pro と Flash の一般提供を発表しました。同じ日に、速度とコスト効率に最適化されたモデルである Gemini 2.5 Flash-Lite も

発表されました。

💻 Gemini CLI は 2025 年 6 月 25 日に Google から正式に公開されました。Gemini 2.5 Pro のためのオープンソースのコマンドラインインターフェースで、AI を活用したコーディング、デバッグ、調査などをターミナル環境で直接行えるようにします。

📜 2025 年 6 月 26 日にリリースされた Jakarta EE 11 は、Java SE 21 および 17 を中心にプラットフォームを現代化し、データアクセスを簡潔にする Jakarta Data 仕様を追加しました。

🧠 エージェントの機能を支える GitHub Copilot Chat 拡張機能は、2025 年 6 月 30 日に MIT License の下でオープンソースソフトウェアとして公開されました。

🧠 GitHub Copilot コーディングエージェントと MCP のサポートは、2025 年 7 月 9 日の Visual Studio Code の 6 月安定版リリース (v1.102) で一般提供となりました。

🧠 2025 年 8 月 6 日、Jules は正式にベータを終えて一般に公開され、Gemini 2.5 を基盤としています。

🧠 GPT-5 は OpenAI が開発したマルチモーダルな大規模言語モデルで、生成事前学習トランスフォーマー (GPT) の基盤モデルシリーズでは 5 番目にあたります。シリーズでは GPT-4 に続くもので、2025 年 8 月 7 日に公開され、推論を伴う機能と伴わない機能を共通のインターフェースのもとにまとめました。

🧠 2025 年 9 月 12 日にリリースされた Visual Studio Code の 8 月アップデート (v1.104) は、「Auto」によるモデル選択を追加し、GitHub Copilot Agent へ独自の指示を与えるための AGENTS.md ファイルのサポートを導入しました。

🐛 Shai-Hulud と呼ばれる npm のサプライチェーン攻撃は、機密データを収集し、アクセスできるパッケージの悪意あるバージョンを自動的に公開する自己増殖型のワームで、2025 年 9 月 15 日に初めて検出されました。

🧠 Notion 3.0 は 2025 年 9 月 18 日に公開され、AI 機能を「Notion AI Agents」として位置づけ直すとともに、ワークスペース内で自律的かつ多段階のエージェント的ワークフローを導入しました。

📊 PostgreSQL 18 は 2025 年 9 月 25 日にリリースされ、非同期 I/O (AIO) を導入して並行する I/O 処理の性能を大きく高めました。

💻 GitHub Copilot CLI は 2025 年 9 月 25 日に公開プレビューへ入ることが発表され、ターミナルに根ざした開発、GitHub との連携、エージェント機能、MCP による拡張性、利用者による制御といった特徴を備えた Copilot コーディングエージェントを、ターミナルに直接もたらしました。

🧠 2025 年 9 月 29 日、Anthropic は Claude Code v2.0 でこのツールを強化し、チェックポイント、ネイティブの VS Code 拡張機能、ターミナルでの操作性の改善、そして Claude Agent SDK を導入しました。

🧠 Anthropic の Skills API (anthropic-beta: skills-2025-10-02) は 2025 年 10 月 2 日に正式に導入され、ワークフローの指示、実行可能なコード、ドキュメントを SKILL.md ファイルによって Claude の再利用可能な機能としてまとめる、標準化された方法を提供しました。

☁️ 2025 年 10 月 8 日、Cloud Native Computing Foundation は、Kubernetes 上のサーバーレスかつイベント駆動型アプリケーションレイヤーである Knative の卒業 (Graduation) を発表し、広範な本番利用に対応できる成熟度に達したことを示しました。

🧠 2025 年 10 月 15 日、Anthropic は Skills Registry を公開しました。これは「Anthropic が検証した」スキルやコミュニティから寄せられたスキルを見つけて共有し、Claude の機能を高めるための中心的なプラットフォームです。

🐛 2025 年 10 月 29 日に起きた世界規模の DNS 設定ミスにより、Microsoft 365、Xbox Live、および Azure の主要な法人利用者に障害が生じました。

🧠 LangChain と LangGraph は 2025 年 10 月 22 日に正式にバージョン 1.0 の節目に到達しました。このリリースは LangGraph が API の安定性を約束する「永続的なエージェントフレームワーク」へ移行したことを示すもので、状態の永続化、人間が介在する (HITL) ワークフローの組み込みサポートと、再設計されたドキュメントサイトを備えています。

💻 Cursor 2.0 は 2025 年 10 月 29 日にリリースされ、「Composer」を導入しました。これは、複数エージェントによるワークフローの統率と、コードベース全体にわたる高速なコード生成に最適化された AI ネイティブのコーディングインターフェースです。

☁️ 2025 年 11 月 14 日、AWS Lambda は Rust によるサーバーレスアプリケーションの構築のサポートを追加しました。

🧠 Google は 2025 年 11 月 18 日に Gemini 3 を正式に公開しました。強化された推論、マルチモーダルな理解、エージェント機能を Gemini アプリ、AI Search、Google AI Studio、Vertex AI などにもたらした大型のリリースです。

🧠 Google Antigravity プラットフォームは 2025 年 11 月 18 日、Gemini 3 のリリースと同時に発表されました。Visual Studio Code に大幅な改変を加えた AI エージェント前提の IDE であり、AI を活用した開発ツールへの Google の参入を象徴するものです。

🧠 Anthropic の Skills フレームワークにおける Model Context Protocol (MCP) のサ

ポートは 2025 年 11 月 20 日に実現し、スキルが標準化された MCP サーバを介して外部のツールやデータソースと滑らかにやり取りできるようになりました。

🌐 Anthropic は 2025 年 12 月 2 日に Bun の買収を発表し、このランタイムを自社のエージェント的ワークフローと開発者向けツールの基盤として使うことを狙いに掲げました。

🏢 Agentic AI Foundation (AAIF) は 2025 年 12 月 9 日、急速に進化する AI エージェントのエコシステムに中立的なガバナンスをもたらすために、Linux Foundation の傘下の新しい財団として設立されました。

🧠 GPT-5 の段階的なアップデートである GPT-5.2 は 2025 年 12 月 11 日にリリースされ、推論能力の向上、遅延の短縮、ツール連携の強化を特徴としています。このリリースは後方互換性を保ちながら、用途ごとにより特化したモードを導入しました。

📜 Ruby 4.0.0 は 2025 年のクリスマスにリリースされました。

💻 2025 年、Neovim は Stack Overflow の開発者調査で 5 年連続となる「最も憧れる開発環境」に選ばれました。

2026

🧠 AI エージェントの構築、改良、デプロイをまとめて扱えるオープンソースの TypeScript フレームワークである Mastra 1.0 は、2026 年 1 月 20 日にリリースされました。Gatsby を手がけたチームが開発したもので、ワークフロー、RAG、メモリ、可観測性のための統合されたツールを提供します。

🧠 GitHub Copilot Agent での Anthropic の Claude モデルのサポートと、強化されたマルチエージェントのセッション管理は、2026 年 1 月の Visual Studio Code のリリース (v1.109) で導入されました。

🧠 GPT-5.3-Codex は OpenAI による大規模言語モデルで、コード生成、速度、そしてリポジトリの検索、ターミナルコマンドの実行、コードのデバッグを行う能力に重点を置いており、2026 年 2 月 5 日にリリースされました。Anthropic の Claude Opus 4.6 に対抗する位置づけで、技術的なベンチマークでは 25% 高速と報告され、当初は ChatGPT Pro の契約者のみが利用できました。

☁️ 2026 年 2 月 6 日、Salesforce は Heroku を保守専念のエンジニアリング体制へ移し、新機能の開発を取りやめて、保守、セキュリティパッチ、安定性の修正だけに限定しました。

🧠 2026 年 2 月 17 日、Figma は Anthropic と提携して Code to Canvas を発表しました。これは、Claude Code のような AI ツールが生成したコードを、完全に編集可能なネイティブの Figma レイヤー、コンポーネント、オートレイアウトのグループ

へ変換する機能です。

🧠 Visual Studio Code の 2026 年 2 月のリリース (v1.110) は、「Agent Plugins」、実験的な「Agentic Browser Tools」、そして GitHub Copilot のエージェントが異なるセッションをまたいでメモリを共有できる機能を導入しました。

💻 GitHub Copilot CLI は、利用者からのフィードバックに基づいて数百の改善が重ねられた 5 か月の公開プレビューを経て、2026 年 2 月 25 日にすべての Copilot 契約者に向けて一般提供となりました。

🧠 Microsoft は 2026 年 3 月 9 日に「Microsoft 365 Copilot Wave 3」を発表し、自律的なエージェント型ワークフローの仕組みである「Copilot Cowork」と、全社的なエージェントの管理と統制のための専用コントロールプレーンである「Agent 365」を導入しました。

🧠 Claude Mythos という名前のモデルの存在は、ブログ記事の草稿が流出したことにより 2026 年 3 月 26 日に世に知られることとなりました。

🧠 自律的なエージェント型ワークフローを構築するために設計された現代的な AI オーケストレーションフレームワークである Microsoft Agent Framework 1.0 は、2026 年 4 月 3 日にオープンソースプロジェクトとして最初に公開されました。

🧠 2026 年 4 月 7 日、Anthropic は Project Glasswing を発表し、Mythos Preview を 11 社へ提供しました。

⚙️ Linux 7.0 は 2026 年 4 月 12 日に最新のメジャー安定版としてリリースされ、先進的なハードウェアのサポートとメモリ安全なプログラミング言語のさらなる統合によって、カーネルの進化を引き続き前へ進めました。

🧠 Anthropic は 2026 年 4 月 17 日に Claude Design を公開しました。Claude Opus 4.7 を基盤とする新しい Anthropic Labs の製品で、デザイン、プロトタイプ、スライド、ワンページャーといった完成度の高いビジュアル作品を Claude と共同で作れます。Pro、Max、Team、Enterprise の各プランでリサーチプレビューとして提供されています。

🧠 OpenAI の GPT-5 シリーズに対する大きなアップデートである GPT-5.5 は 2026 年 4 月 23 日にリリースされ、推論、マルチモーダルな理解、エージェント機能の改善によって能力を大きく前進させました。API での提供は 2026 年 4 月 24 日に発表され、OpenAI の API エンドポイントを通じて最新のモデルが開発者に届けられました。

🧠 Google は 2026 年 5 月 19 日の Google I/O で Antigravity 2.0 を発表し、Gemini CLI を置き換える Go 製のターミナルエージェントとして Antigravity CLI を導入しました。これはデスクトップアプリと同じエージェントランタイムへの軽量なインターフェースを提供します。

🧠 2026 年 6 月 2 日、Anthropic はサイバーセキュリティ向けに Claude Mythos の提供範囲を広げ、150 の組織が利用できるようにしました。

🧠 2026 年 6 月 4 日、Anthropic は「When AI builds itself」と題する Anthropic Institute の論文を発表し、自社コードベースにマージされたコードの 80% 以上が現在 Claude によって書かれていること (2025 年 2 月の Claude Code のリサーチプレビュー開始時点では一桁台前半でした) を明らかにし、AI システムが自律的に自らの後継モデルを設計・構築・訓練しうる「再帰的自己改善 (recursive self-improvement)」への道筋を示した上で、アライメント研究や社会制度が技術の進歩に追いつけるよう、検証可能なグローバルな一時停止 (ポーズ) の仕組みを求めました。

🧠 2026 年 6 月 9 日、Anthropic は一般利用に向けて安全性を確保した Mythos クラスのモデルである Claude Fable 5 を、利用が制限された Claude Mythos 5 とともに公開しました。このモデルは 100 万トークンのコンテキストウィンドウと、1 リクエストあたり最大 128,000 の出力トークンを備えています。

🧠 2026 年 6 月 12 日、Anthropic は Fable 5 のジェイルブレイクが White House に報告されたことを受け、外国籍の利用者によるアクセスを停止せよという米国商務省からの指示に従うため、Mythos クラスのモデルである Claude Fable 5 と Claude Mythos 5 へのアクセスをすべて停止しました。

☁️ 2026 年 6 月 22 日、AWS は Lambda MicroVM を発表しました。Firecracker の仮想化技術の上に構築され、VM レベルの分離をほぼ瞬時の起動・再開とともに提供するサーバーレスのコンピュートサービスです。このサービスは US East (N. Virginia、Ohio)、US West (Oregon)、Europe (Ireland)、Asia Pacific (Tokyo) で直ちに利用可能となり、利用者が書いたコードや AI が生成したコードを実行するための隔離されたサンドボックスを、最大 8 時間の状態保持とともに提供します。

🧠 2026 年 7 月 16 日、Moonshot AI は Kimi K3 を公開しました。これは 2.8 兆パラメータの Mixture-of-Experts 大規模言語モデルで、896 個のエキスパートからトークンごとに 16 個が選ばれ、1 トークンあたり 1,040 億パラメータが活性化されます。Kimi Delta Attention と Attention Residuals を基盤とする 100 万トークンのコンテキストウィンドウを備えています。Artificial Analysis の AI リーダーボードでは Anthropic の Claude Fable 5 と OpenAI の GPT-5.6 Sol に次ぐ第 3 位でデビューし、Arena.ai のフロントエンド Web 開発ベンチマークでは競合モデルを上回りました。

🧠 2026 年 7 月 27 日、Moonshot AI は Kimi K3 の完全な重みを 1.4 TB のダウンロードとして Hugging Face で公開し、その時点で最大のオープンウェイトモデルとなりました。重みは MIT 由来の独自の Kimi K3 ライセンスの下で提供され、研究、社内利用、および小規模な商用利用を許可する一方で、Model-as-a-Service の収益が年間 2,000 万ドルを超えた場合には Moonshot との個別の商用契約を必要とします。