Engineer

アーキテクチャ

🧩 システムの各部分の組み合わせ方と、その理由

どのようなものか

ソフトウェアアーキテクチャ、システムアーキテクチャとも呼ぶアーキテクチャは、ある環境にあるシステムの基本的な概念と性質の集合です。要素、それらの関係、設計と変化を支配する原則に表れます。この説明は、2022年11月に公開されたNIST SP 800-160 Volume 1 Revision 1『Engineering Trustworthy Secure Systems』の定義に従っています。

定義の各部分に意味があります。要素は、サービス、データベース、キュー、フロントエンド、その内側のモジュールといった部品です。関係は、部品同士の依存と、その間を行き来するものです。原則は、新しい仕事を置く場所を決め、成長しても要素と関係がずれないようにするルールです。「その環境における」は、利用者、隣接システム、制約も一部であることを示し、「進化」は、描いた当日だけ正しい説明では不十分であることを意味します。

すべての設計選択が、アーキテクチャの判断ではありません。Architectural Decision Record（ADR）の形式を管理する団体は、「アーキテクチャ上重要な機能・非機能要件に対処する、理由のある設計選択」と説明します。重要な要件とは、構造と品質への影響を測れるものです。arc42テンプレートはさらに明快で、判断の節は「重要な判断、費用や手間の大きい判断、リスクの高い判断、意見が分かれる判断」のためとしています。

一枚の図にすべては表せないため、複数の視点で記述します。Simon BrownのC4モデルは、地図の縮尺を変えるような図を提供します。利用者や周囲のシステムとの位置関係を示すシステムコンテキスト図、内側のアプリケーションとデータ保管先を示すコンテナー図、一つのコンテナー内のコンポーネント図、役立つ場合のコード図です。arc42は、目標と制約から判断、リスク、技術的負債まで、十二の節で文章を整理します。

概念を判断に生かす

「アーキテクチャとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 まず全体像を設計し、各部分のつながりと費用を、コードを書き始める前に明らかにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 14部門を支える共通インフラ計画 — 14 共通インフラ計画の対象となった研究部門

14部門を支える共通インフラ計画 お問い合わせ →

なぜ重要なのか

チームにとっての問いは、どの構造を選ぶか以上に、どの選択を今、意識的に決めて記録し、どれを未決のまま残すかです。

構造上の問題を早く決めると、まだよく分からないうちに、その負担を引き受けます。プロジェクトで情報が最も少ない時点で選び、最初の数週間が遅くなります。未決のままでも構造は決まり、最後にコードに触れた人が、その都度便利な変更で固めます。解決は、選択を二種類に分けることです。コードが依存してからでは変更に大きな負担がかかる選択も、少数あります。システムの分け方、業務ルールの所有者、データへのアクセス、すべての変更が越える基準です。それらは早く意図して決めるべきです。残りの大半は少ない負担で再検討でき、コードから必要性が分かるまで待てます。

記録は判断の後半で、理由が残る場所です。ADRは、一つの判断と、その理由、利点・不利益、結果を記録し、プロジェクトのADR群が判断の履歴になります。ADRのサイトは、Michael Nygardの2011年の「Documenting Architecture Decisions」が、この考えを広めたとしています。必要な手間は、判断ごとに一ページを書き残すことです。得られるのは、後の人が、意図した選択と偶然を区別し、その理由がまだ成り立つか確認できることです。

どれも動く構造の間では、品質目標が選択を決めます。arc42は、主要な関係者が最も重視する品質目標を、上位三つ、最大五つ挙げるよう求め、プロジェクトの目標とは別だと警告します。機能が同じ二つの設計も、壊れ方、拡張、変更の方法は大きく異なり、どの違いを重視するかは品質目標だけが示します。

実際の活用例

プラットフォームのリポジトリは、2025年11月25日に空の状態で始まり、初期の技術判断のページは、そうした選択を「三日目なら安くでき、六か月目にはほぼ取り消せない」と論じます。当時の三つのうち二つが、構造を形づくりまし

た。Next.jsのフロントエンド、検証・転送するGo API、業務ルールを持つPostgreSQL関数層に分け、各部分に一つの仕事を持たせました。後付けでは「全ハンドラーの書き直しが必要になっていたはず」なので、最初からストアド関数経由のデータアクセスにしました。九か月、数千のコミットの後も、層は曖昧にならず、テーブルへ直接アクセスするハンドラーはないと報告します。初期判断の試験は、よく聞こえたかではなく、その後の仕事量に耐えたかです。

偶然に任せた構造が、偶然に決まる様子は、GIS地図開発のページにあります。地図は、PostgreSQLの空間データ、Mapboxの描画、ReactJSのフロント、NodeJSのバックと、別々に育っていました。個々は動いても、「組み合うようにつくられていなかった」のです。仕事はフルスタック開発と技術的な方向付けを含み、最初に構造を定めました。その後は、「責任が隣の層へ漏れ出すたびに、本来の層へ戻し続け」、多くの作業は「小さな不整合が構造として固まる前に捕まえること」でした。

コンテナー実行基盤のページは、理由の記録を本当の成果と捉えます。業界標準のエンジンは、ホストのファイアウォールルールより前でポートを公開します。そのため、OS標準のサービス管理機構が生成するユニットを、デーモンのないエンジンで動かす構成とし、公開ポートを一切持たせません。一つの選択を明記しています。ユニットをrootlessではなくシステムレベルで実行し、rootlessなら得られるより厳密な分離と引き換えに、自動化を単純にしました。コンテナー脱出がより重要になったら、どちらへ移るかも述べています。これがADRの持つもの、判断、理由、利点・不利益、結果です。「次の人は、読む記事すべてから標準を使うよう勧められる」ため、そのページは、選択自体より理由と一緒に書き残すことを重視します。

問題が起きるところ

慎重な新任者が、意図した判断を覆します。 理由が誰かの頭だけにあると、選択は誤りに、どの入門記事も勧める標準は修正に見えます。覆すのに大きな負担がかかる判断には、それぞれADRを書きます。背景、比較した案、選んだもの、結果、再検討する条件を記載します。コードの横に置き、双方の変更を一緒にレビューします。

業務ルールが三層に同時に現れます。 各変更のロジックを、その日の便利な場所に置き続け、同じルールがフロント、API、データベースで少しずつ違って検査されます。各層の責任を名付けて記録し、漏れたら、そのコピーが小さいうちに戻します。可能なら、APIには関数呼び出しだけを許し、テーブルの権限を与えないなど、境界も強制します。

六か月目に入れたルールが、整理のプロジェクトになります。 全体が依存する判断を未決のままにし、それ以後のコミットが、その時点の状態を前提としてきました。変更の負担が大きい少数の選択を先に決め、残りは意識的に未決のまま残します。判断の履歴にそう書き、未決事項が、次の変更で黙って決まらず、未決と見えるようにします。

構成図が、誰にも何も説明しません。 利用者、サービス、クラス、テーブルを一枚に混ぜ、線に名前がないか、「使う」としか書いてありません。C4のように縮尺ごとに図を分け、すべての線に関係の向きと意図を示します。コンテナー間では技術やプロトコルも記載します。C4の表記指針もそれを求め、「Uses」など一語だけのラベルを避けるよう勧めています。

参考資料

- NIST CSRC用語集のarchitecture：ここで使う定義と、同じ項目のFIPS・セキュリティアーキテクチャの定義です。
- NIST SP 800-160 Vol. 1 Rev. 1：定義の出典『Engineering Trustworthy Secure Systems』です。
- Architectural Decision Records：判断が構造上のものとなる条件、記録用テンプレート、補助ツールです。
- C4モデル：コンテキストからコードまでの四段階の図、表記の助言、レビュー項目です。
- arc42：arc42テンプレートの十二の節それぞれの、実用的な助言と例です。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「アーキテクチャとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/architecture/