

30分でわかる アーキテクチャ設計の基本

ビジネスの価値を高めるクラウドサービス目利きのポイント

2023/09/28

Masatsuna Kawamata

川又 眞綱 (かわまた まさつな)

所属

BIPROGY株式会社

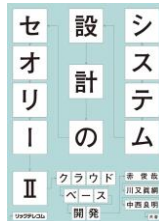
経歴

MIDMOST for .NET Marisの開発を担当。
鉄道、流通、小売、金融などの大規模基幹系
システム開発に参画。

趣味

競技スキー、自作パソコン

著書



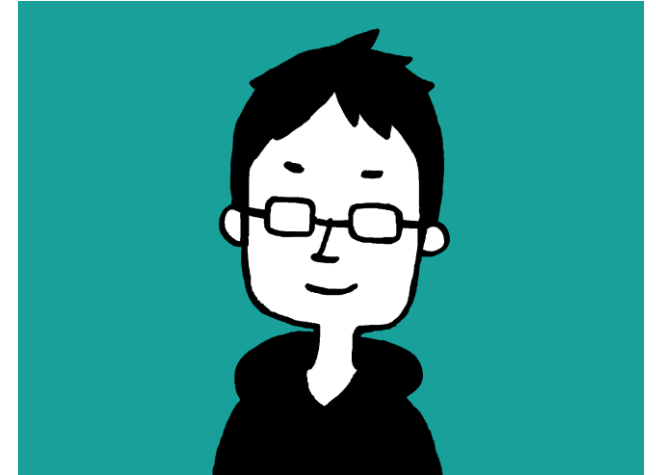
システム設計のセオリーⅡ
クラウドベース開発

<https://amzn.asia/d/5yRpkjs>



Visual Studioで学ぶ
C#プログラミング

https://zenn.dev/tsuna_can/books/basic-csharp-programming



✕ @tsuna_can_se

🔗 <https://github.com/tsuna-can-se/>

📖 <https://tsuna-can.hateblo.jp/>



本日お話しする内容は個人の見解であり
所属する組織を代表するものではありません

みなさんクラウドは使っていますか？

- ✓ 私はほぼ毎日触っています。
 - PowerPoint、Excel、TeamsなどのMicrosoft 365製品
 - Azure
 - AWS
 - GitHub、Azure DevOps、Jira(ALMツール)
 - ChatGPT、Google Bard
 - GitHub Copilot
 - Bing Image Creator

もはやクラウドを使わない日はない

クラウドサービスの利用は拡大傾向



AWS、Azure、GCPを筆頭に、
毎年20%以上の成長が続く



IaaSやPaaSの活用が
当たり前となって久しい

クラウドを使わないシステムは少なくなっている

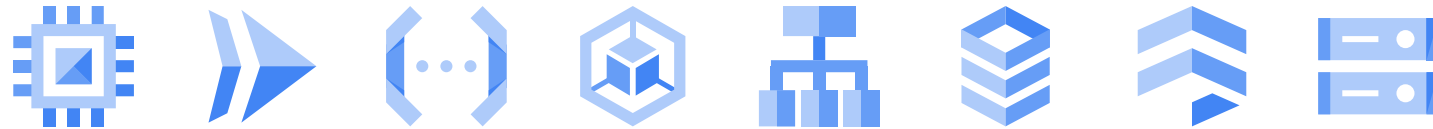
クラウドを活用するアーキテクチャ設計に必要なものとは？

大量のサービスがクラウドベンダーから提供されている

AWS



Google Cloud



Microsoft Azure



いつ・何を・どこで使うか**判断する**ことが大切

歴史を紐解きながら、基本原則を見直そう

メガクラウドの誕生

AWSとAzure爆誕！

2000年代のITシステムの課題感

✓ スケーラビリティ

- ウェブサービスとオンラインビジネスが急速に成長
- 急増するトランザクションを支えるスケーラブルなインフラが必要

✓ セキュリティとプライバシー

- サイバーセキュリティの脅威が増加
- 個人情報保護に関する法規制も強化

✓ コスト管理

- ハードウェアとソフトウェアの購入、アップグレード、維持コストが負担に
- コスパ重視の風潮

✓ データ管理とストレージ

- データ量が急速に増加
- 大量データの効率的な保存、分析に対する需要が拡大

✓ システム統合

- システムのサイロ化
- 異なるシステム間でのデータ共有



「IT革命」終盤の時代

クラウドサービスの登場

✓ AWS

- 2004年 ローンチ
- SQS(2004)
- S3、EC2(2006)
- IaaS中心



✓ Azure

- 2008年 ローンチ
- Xxx Service
- PaaS中心



クラウドベンダーからは数多のサービスが提供される時代

クラウドサービスの提供価値

オンデマンドで適切な量のリソースを素早く確保できる

- 自動的なスケールアウトにより、事前のリソース確保が不要
- スケールアップ、スケールダウンも条件次第で実現可能
- 災害に強いシステム構成を現実的なコストで実現できる

高セキュリティな物理環境の管理をクラウドベンダーにお任せできる

- データセンターの施錠や入退室管理はベンダーにおまかせ
- ストレージ交換や配線など、物理的なインフラ設計・運用が不要

所有から利用へ

- 各種機器の購入が不要で、資産を持つ必要がなく、経費として処理できる

ビジネス上の問題解決やイノベーションに集中できる

- 定型業務からエンジニアを解放できる
- システムがビジネスの価値を阻害せず、けん引できる存在になる

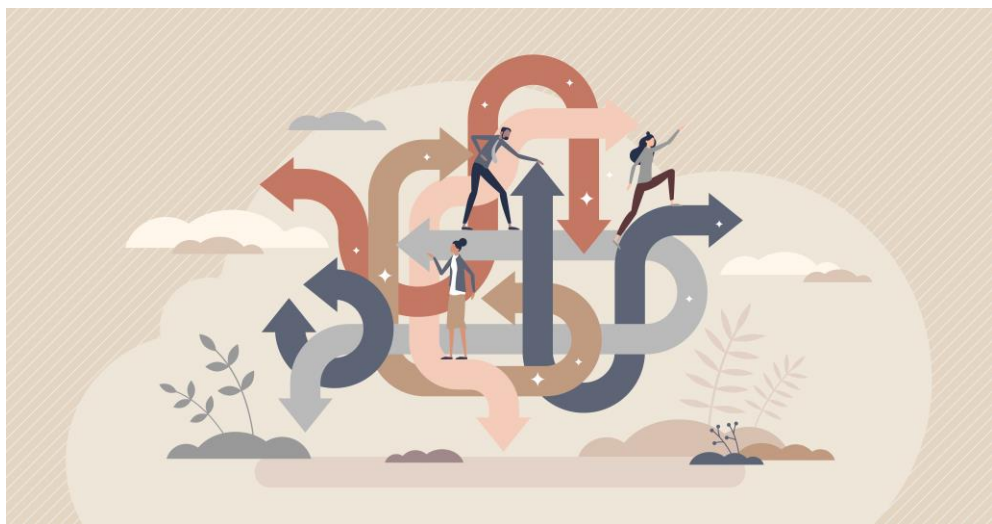
ビジネス上の課題を
解決するものだったから

2020年代のクラウド

クラウドネイティブ？

個人的にいま危惧していること

- ✓ クラウドサービスの利用や、最新技術の導入がビジネス上の課題を解決しているのか？



私が最近よく見る意思決定

- 理由の不明確な先進技術の導入
- 複雑なシステムアーキテクチャの採用
- 非常に高い可用性
- アーキテクチャ・開発プロセスと組織構造・商習慣のミスマッチ

「いい感じなシステムを作ること」が目的になっていないか？

『小さな独立した複数のサービスで
ソフトウェアを構成する、
ソフトウェア開発に対する
アーキテクチャ的、組織的アプローチ』

出典: AWS マイクロサービスの概要(2023/7/28)
<https://aws.amazon.com/jp/microservices/>

アーキテクチャを変えるだけではメリットが生きないことを示唆

マイクロサービスアーキテクチャのメリットとビジネス上の価値

俊敏性、機敏性が高い

▶ そんなに高い俊敏性はそのビジネスに必要ですか？

サービスごとにスケーリングできる

▶ 基盤コストは最適化されても管理コストのほうが高くない？

サービスの管理範囲が狭まり
オーバーヘッドが下がる

▶ 管理範囲を狭めるためには後方互換性の担保が必須だし、そのコストのほうが高かついたりしません？

技術の組み合わせが自由

▶ 最終的に技術的負債が発生する未来が見えませんか？

サービスの共有が可能

▶ ライブラリを共有するだけで十分だったりしません？

耐障害性を高く保てる

▶ システムは無停止でも全体のビジネスは止まりませんか？

サービス毎にデータが分離される

▶ 整合性維持にかかる設計や運用のコストって考えてます？

GitHubの元CTOのつぶやき



Jason Warner

@jasoncwarner

I'm convinced that one of the biggest architectural mistakes of the past decade was going full microservice

On a spectrum of monolith to microservices, I suggest the following:

Monolith > apps > services > microservices

So, some thoughts

[ツイートを翻訳](#)

午前3:45 · 2022年11月15日

私は、過去10年で最大のアーキテクチャ上の間違いのひとつは、完全なマイクロサービスに移行したことだと確信している。

新しいアーキテクチャがビジネスの課題を解決するとは限らない

参考：<https://tech-blog.rakus.co.jp/entry/20201218/architecture>

サービスやアーキテクチャが
あなたのビジネス上の課題を
解決するのか、よく考えよう

オンプレへの回帰

クラウドが悪いの？

オンプレミスへの回帰

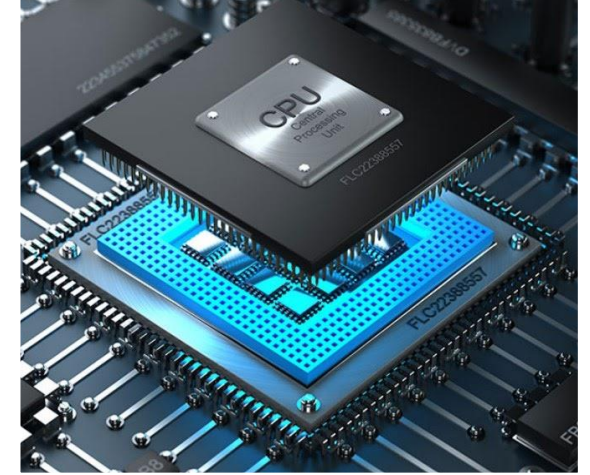
- ✓ 2010年代後半から、オンプレミス回帰の動きがみられる



コストの問題



セキュリティの
問題



パフォーマンスの
問題

クラウドがビジネスを阻害するケースは一般企業ならほぼない

コストの問題

- ✓ 利用料にはクラウドベンダーの利益がのっている
- ✓ 特に大規模なシステム基盤を構築すると高コストになりやすい

サーバーの調達コスト、物理的な場所のコスト、
節税効果など、加味して比較しているか？

無駄に大きく高価なリソースを使ってないか？

もっと安いサービスが使えないか？

セキュリティの問題

- ✓ クラウドベンダー内で正しい運用がされているかわからない
- ✓ データの保全がされていることを保証しきれない

オンプレミスなら安心な理由とは？

クラウドベンダーより厳格な運用が
適切なコストで実現できるか？

利用者としての責任を果たしているか？

パフォーマンスの問題

- ✓ 大量のコアや高性能GPUを積んだマシンがない
- ✓ クラウドで実行するとI/O性能がネックになりやすい

リソースの共有や、増減する要求にあわせて動的にスケーリングできるのが利点では？

その高スペックなマシンは常時必要？

オンプレ回帰ではなく
そもそものユースケース誤り？

自分たちの使い方の問題なのか
クラウドの問題なのか
切り分けてよく考えよう

クラウドサービスの活用に向けて

アーキテクチャ設計の基本とはなにか

中心となるクラウドサービスを見極める

汎用型サービス

アプリケーションの実行

データの保存

メッセージ連携

ネットワーク構築

セキュリティ

システム運用

こちらを中心に据える

特化型サービス

AI/機械学習

AR/VR

IoT

ロボット

動画/メディア配信

ゲーム

特定業界向けサービス

アーキテクチャの提供する価値を考える



オンプレでは実現できなかった
スピード感でサービスを組合せる
ことができるようになった



コスト面で折り合いがつかなかったシステムアーキテクチャを
実現できる可能性が広がった

ビジネス上の要求や課題に対する解決策なのか確認する価値がある

非機能要求/非機能要件を軽視しない

機能要求/機能要件

実現できるのが当たり前なもの

ビジネスを進めるのに必要なもの

非機能要求/非機能要件

現行システムに対するビジネス面の課題が潜んだもの

ビジネスの拡大に必要なもの

- ✓ 非機能要求/非機能要件が「おまけ」の時代は過ぎ去った
- ✓ 非機能要件はビジネスと密接に関係し、ビジネスの価値に直結する

アーキテクチャはビジネスの価値を高める非機能要件から導くもの

リファレンスアーキテクチャを研究する

✓ AWS リファレンスアーキテクチャ図

➤ <https://aws.amazon.com/jp/architecture/reference-architecture-diagrams/>

✓ Azure アーキテクチャ センター

➤ <https://learn.microsoft.com/ja-jp/azure/architecture/>

✓ GCP Cloud アーキテクチャ センター

➤ <https://cloud.google.com/architecture>

✓ Oracle Architecture Center

➤ <https://docs.oracle.com/solutions>

リファレンスの組合せで非機能要件が実現できないか検討

利用者の責任範囲はどこか必ず確認する

まとめ

クラウドの歴史と最近の動向

- ✓ クラウドはビジネスの課題を解決する存在
 - コスト、セキュリティの課題にアプローチ
 - 所有から利用へ
- ✓ 最近ではビジネスに寄与しないアーキテクチャ導入が見られる
 - 猫も杓子もマイクロサービスアーキテクチャはおかしい
 - ビジネスの課題を解決するのかよく考える
- ✓ 脱クラウド、オンプレミス回帰が一部見られる
 - ほとんどのケースで、クラウドサービスの使い方に問題がある
 - もともとクラウドの利用シナリオに存在しないものをクラウド上でやろうとしていた

クラウドベースのシステムアーキテクチャ設計の基本

- ✓ 汎用型のサービスを中心に据えてシステムアーキテクチャを組み立てる
- ✓ 構築するシステムの課題と、アーキテクチャの解決する課題が一致していることを確認する
- ✓ ビジネスの課題に根差した非機能要件からシステムアーキテクチャを導く
- ✓ クラウドベンダーの出しているリファレンスアーキテクチャを研究する
 - ただしベンダーに踊らされてはいけない

■ 免責事項

- 本資料は可能な限り正確な情報を掲載するよう努めております。しかし、誤情報が入り込んだり、情報が古くなったりすることもあり、必ずしもその内容の正確性および完全性を保証するものではありません。当該情報に基づいて被ったいかなる損害について、一切責任を負うものではありませんのであらかじめご了承ください。

■ 他社所有商標に関する表示

- 本書に記載の商品名、製品名、社名などは、各社が商標として使用している場合があります。
- Amazon Web Services、AWS、Powered by AWS ロゴ、およびかかる資料で使用されるその他のAWS商標は、Amazon.com, Inc. またはその関連会社の商標です
- MIDMOST、Marisは、BIPROGY株式会社の登録商標です。
- 「Azure」「Visual C#」「Visual Studio」は米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- その他記載されている会社名・商品名などは、各社の商標および登録商標です。
- 本書に記載の商品名、製品名、社名などには、必ずしも商標表示((R)、TM)を付記していません。

Thank you.



✕ @tsuna_can_se

🔗 <https://github.com/tsuna-can-se/>

🔗 <https://tsuna-can.hateblo.jp/>