

JAPAN | 2024

aws SUMMIT



SaaS 開発と プラットフォームエンジニアリングの未来

後藤 健汰

アマゾン ウェブ サービス ジャパン合同会社
技術統括本部 デジタルサービス技術本部 ISV/SaaSソリューション部
ソリューションアーキテクト

自己紹介

後藤 健汰 (Kenta Goto)

ISV/SaaS Solutions Architect

アマゾン ウェブ サービス ジャパン合同会社



□ バックグラウンド

インフラエンジニアとして、SaaS 事業者にて @kennygt51
サービス運用・クラウド移行を経験

□ 好きな AWS サービス

Amazon EKS



本セッションの対象者

- SaaS 事業者の開発組織に責任を持つ方
 - e.g. エンジニアリングマネージャー、VPoE、CTO
- SaaS を開発・運用されている方
 - e.g. インフラエンジニア、開発エンジニア
- 「プラットフォームエンジニアリング」に興味のある方

本日のゴール

- プラットフォームエンジニアリングについて理解する
- SaaS 開発とプラットフォームエンジニアリングの親和性を理解する



プラットフォームエンジニアリングが

SaaS を開発する組織が抱える課題を解決する打ち手の選択肢になる

アジェンダ

1. SaaS ビジネスの潮流
2. SaaS 事業者の開発組織が抱える課題
3. プラットフォームエンジニアリングによる開発組織の変革
4. AWS を活用したプラットフォームエンジニアリングの実践
5. まとめ

SaaS ビジネスの潮流



SaaS ビジネスの概況

- ここ数年、SaaS 事業者のマルチプルは低空飛行
- 先行投資で売上高を伸ばしてきた SaaS 事業者も、利益体質への移行・維持を求められている
- 利益を確保しながら成長を続ける必要性

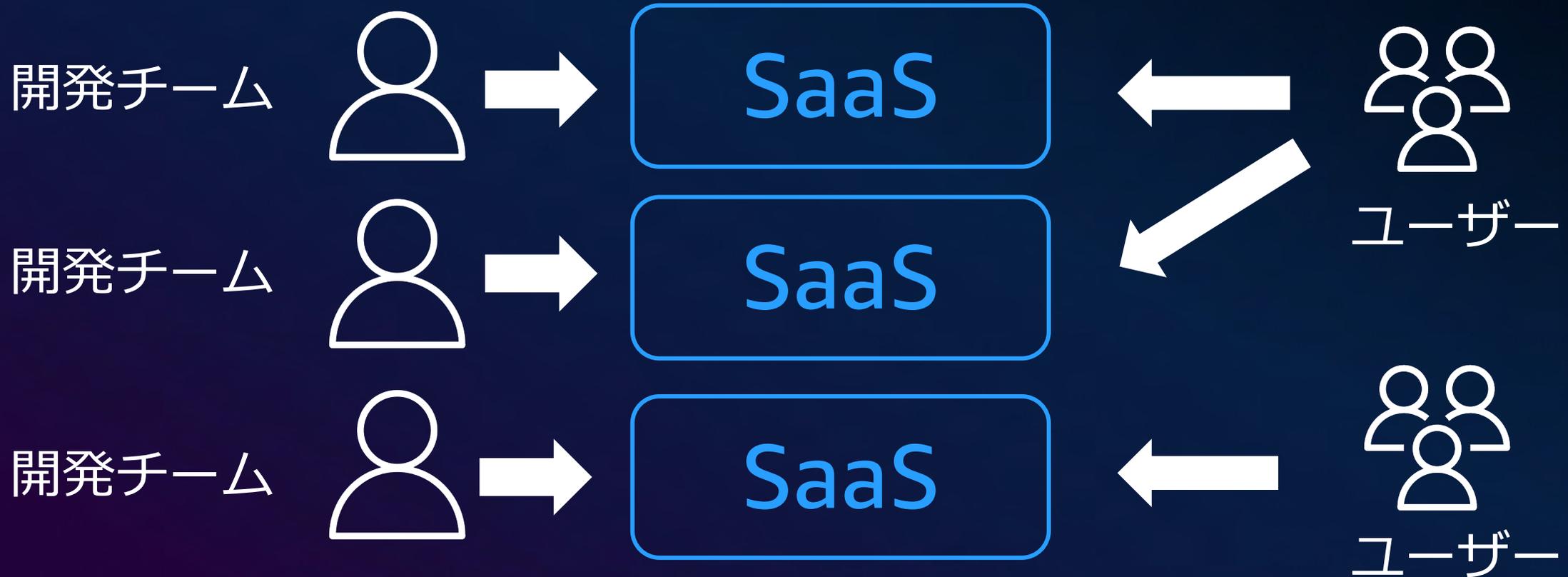
SaaS 事業者の成長に向けた戦略

1. ターゲットの拡大



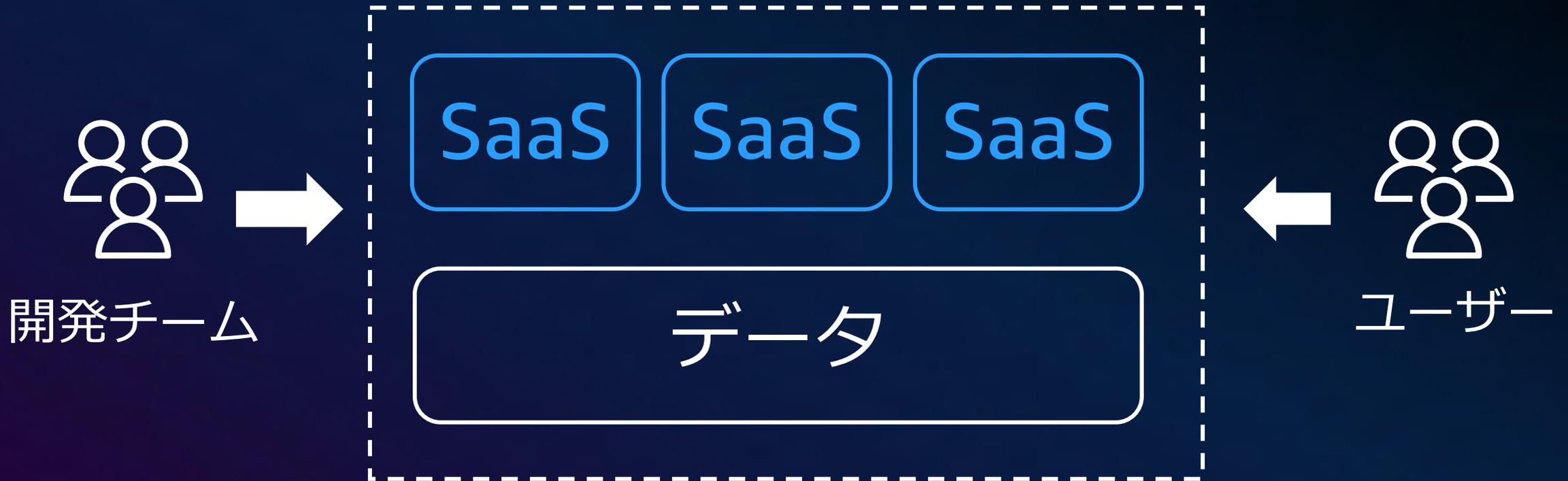
SaaS 事業者の成長に向けた戦略

2. マルチプロダクトの展開



SaaS 事業者の成長に向けた戦略

3. コンパウンド・スタートアップ



SaaS 事業者の開発組織が抱える課題

SaaS のさらなる成長のために

ターゲットの拡大

- 新しいターゲットが求める機能の追加
- プロダクトの規模拡大

マルチプロダクト の展開

- 新規プロダクト開発
- 開発チームの増加

コンパウンド・ スタートアップ

- 複数のプロダクト開発
- 複数の開発チーム

SaaS のさらなる成長のために

ターゲットの拡大

- 新しいターゲットが求める機能の追加
- プロダクトの規模拡大

マルチプロダクト の展開

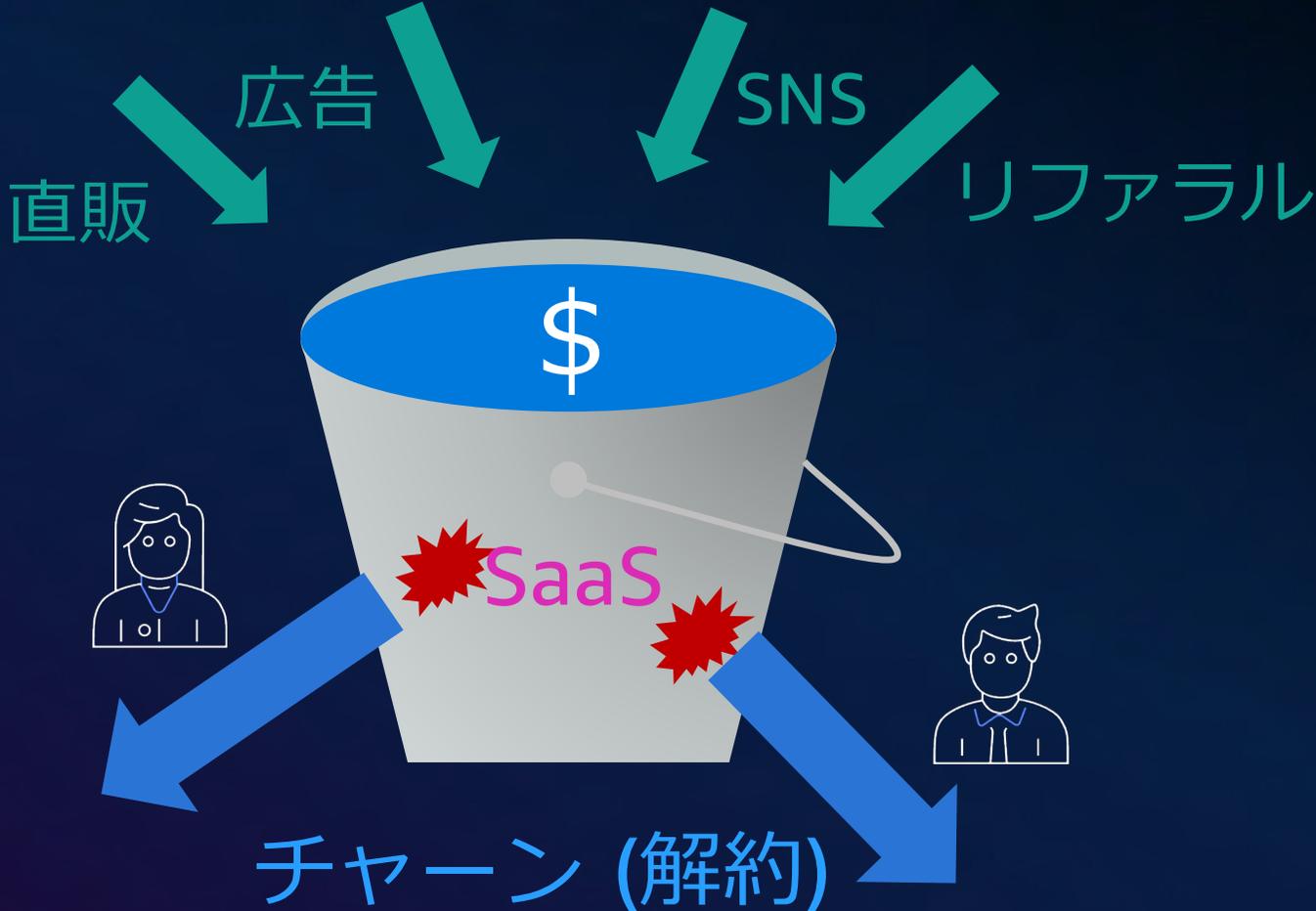
開発組織の拡大

- 開発チームの増加

コンパウンド・ スタートアップ

- 複数のプロダクト開発
- 複数の開発チーム

継続的な顧客体験の改善



継続的な顧客体験の改善

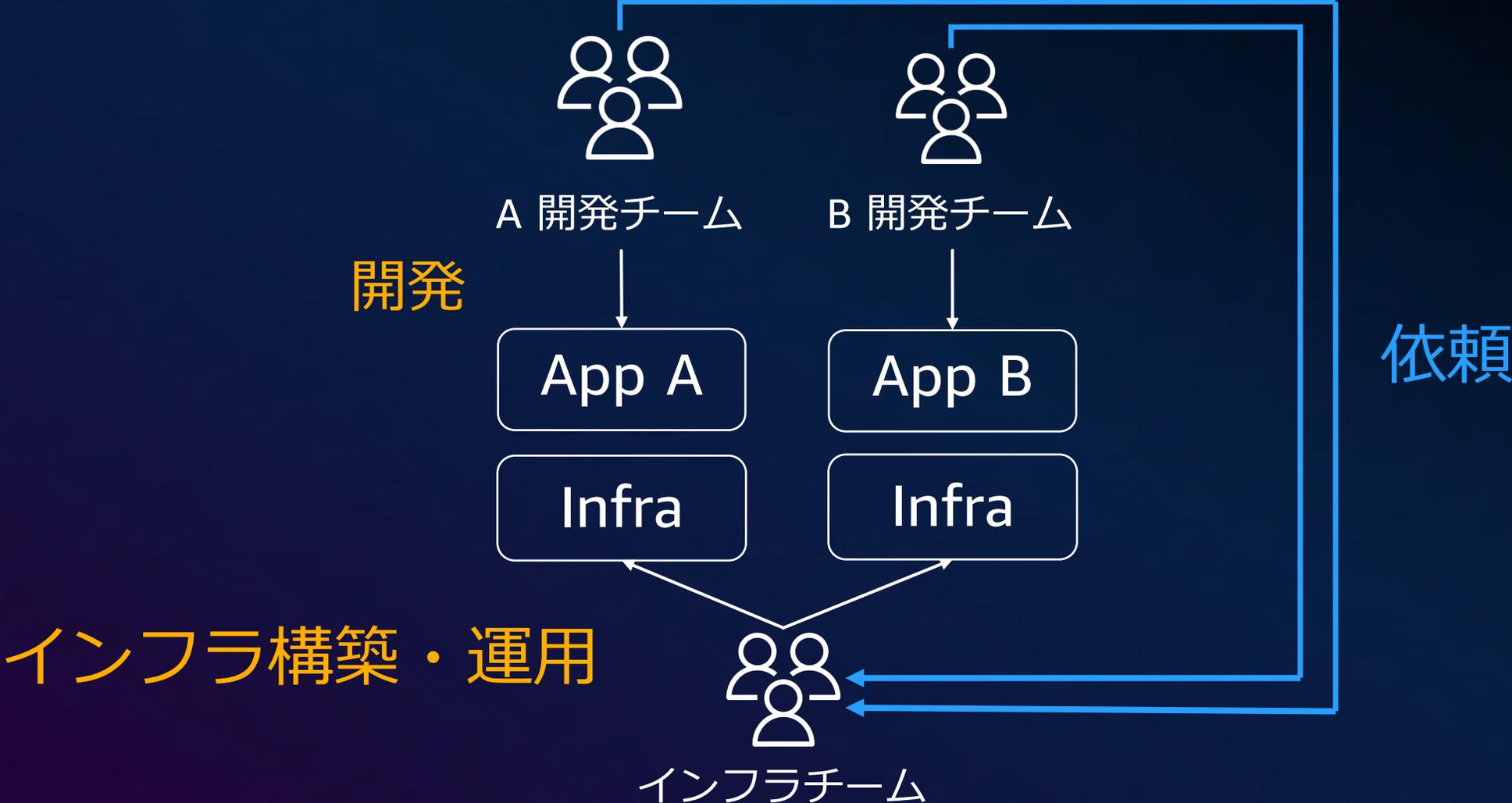
開発生産性の 継続的な向上

SaaS 開発組織に求められる難しい舵取り

開発組織の規模が拡大する中で、

開発生産性を向上し続ける必要がある

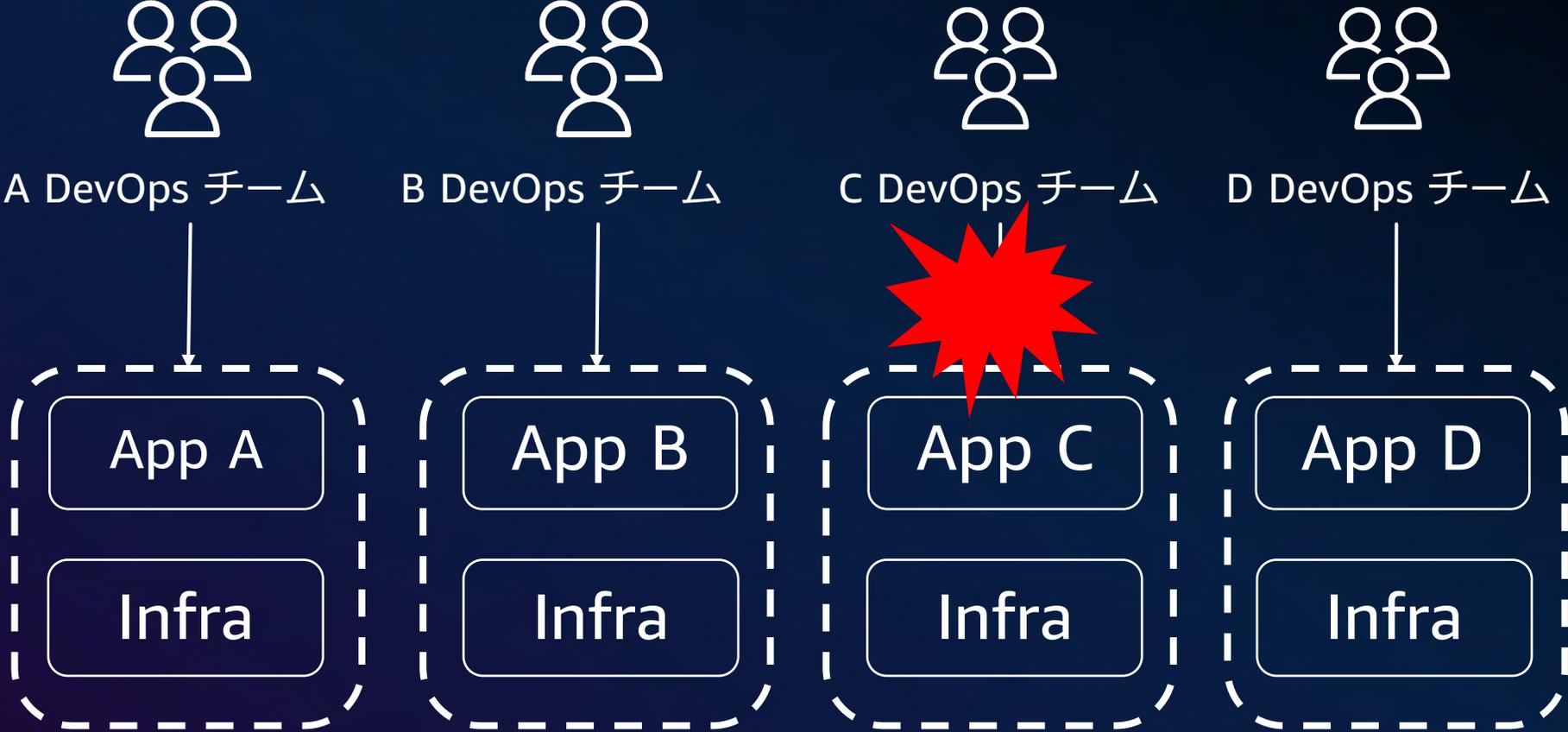
拡大する開発組織の変遷



拡大する開発組織の変遷



拡大する開発組織の変遷



開発組織が拡大する中で開発生産性を高め続けるには

- ✓ SaaS 事業者の成長に伴う、開発組織の拡大
- ✓ 開発チームの増加によって、他チームの負担が増加
- ✓ 権限委譲によって解決を図ると、開発チームの負担が増加



開発生産性の低下



プラットフォームエンジニアリング

プラットフォームエンジニアリングによる SaaS 事業者の開発組織の変革

プラットフォームエンジニアリングとは



認知負荷の軽減



認知負荷 (Cognitive Load)

目の前の情報を処理する (認知) する際にかかる負荷

課題内在性負荷

Intrinsic Cognitive Load

学習対象の難解さによる負荷

課題外在性負荷

Extraneous Cognitive Load

学習に無関係な余分な負荷

学習関連負荷

Germane Cognitive Load

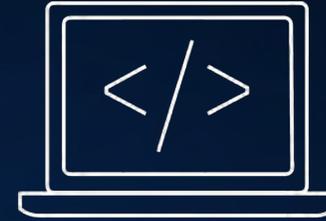
適切な学習のための必要な負荷

「仕組み」によって解決すべき

プラットフォームエンジニアリングのポイント

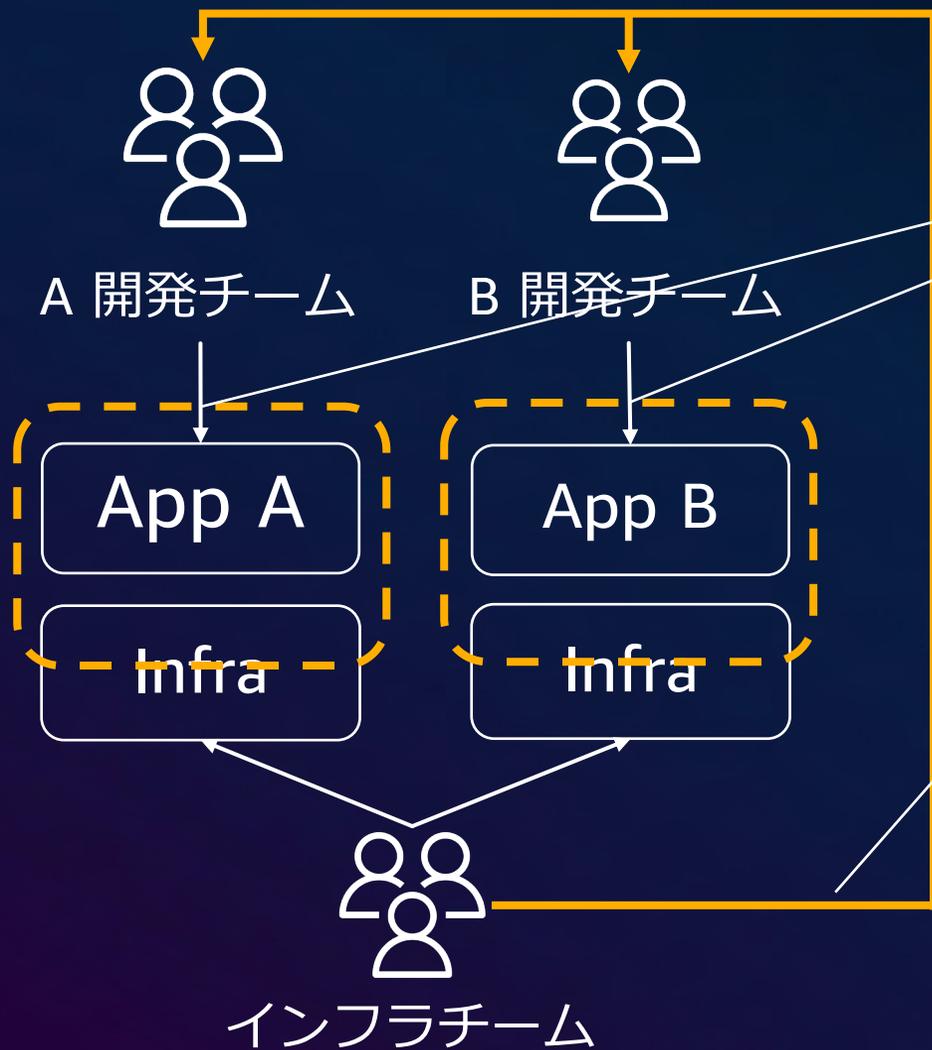


ゴールデンパス



セルフサービス

ゴールデンパス

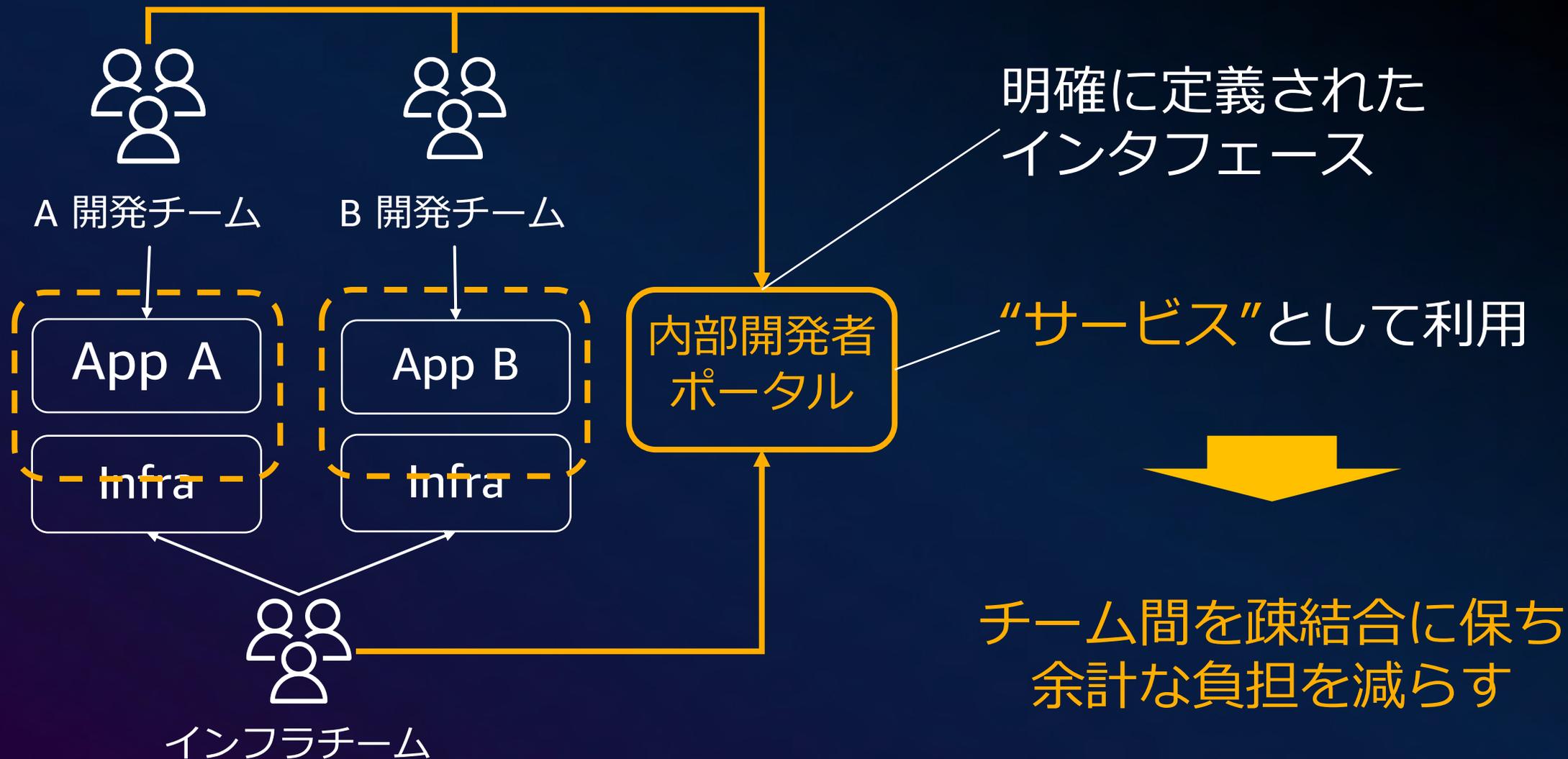


責務の委譲

IaC テンプレートや
オンボーディングの整備

開発タスクとは関係ない
余計な負担を減らす

セルフサービス



プラットフォームエンジニアリングによる変革

自律的な開発チームによって構成される開発組織

プラットフォームエンジニアリングによる
開発チームの認知負荷の軽減



開発生産性の継続的な向上

AWS を活用した プラットフォームエンジニアリングの実践

プラットフォームエンジニアリングの実践



“使われないプラットフォーム”という失敗



コンセプト
設計

チーム組成

設計と実装

成果の計測

改めて、SaaS を提供するとは

SaaS をサービスとして提供し続けるために必要なことは？

- 提供する SaaS のミッションやビジョンを策定
- ターゲットユーザー、提供する価値の検討
- 機能開発における優先順位を整理
- MVP (Minimum Viable Product) による価値検証
- サービスを正しく使ってもらうための様々な支援

コンセプト
設計

チーム組成

設計と実装

成果の計測

Platform as a Product

プラットフォームを「**プロダクト**」として捉え、
広めていく活動をおこなう



コンセプト
設計

チーム組成

設計と実装

成果の計測

プロダクトとしてのコンセプトの整理



Why

ユーザーは誰で、どんなペインを解決するのか？

What

Why を実現する解決策はなにか？
何をつくり、どのような優先度で取り組むのか

How

What をどのように実現するのか？

コンセプト
設計

チーム組成

設計と実装

成果の計測

最小限のプラットフォームを維持する

TVP (Thinnest Viable Platform)

価値提供を実現する、最小限のプラットフォームを継続的に維持する

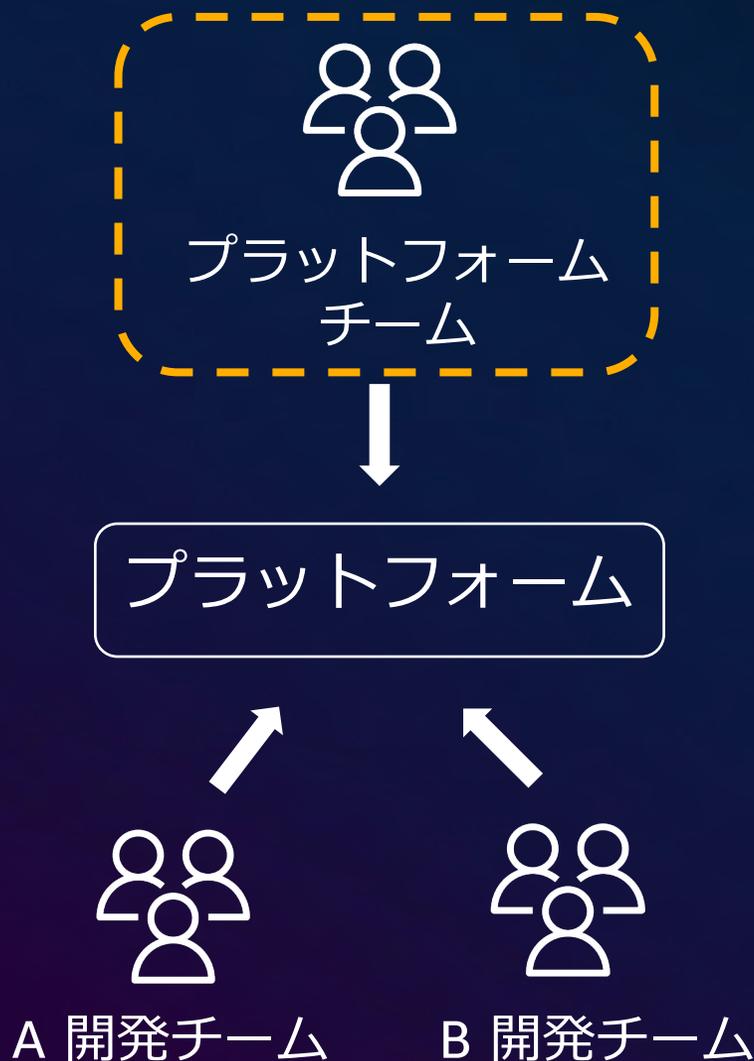
コンセプト
設計

チーム組成

設計と実装

成果の計測

プラットフォームチームを組成する



- プロダクトを開発するチーム
- チームとしての責務
 - 機能開発と継続的な改善
 - ユーザーの要件調査
 - ロードマップの策定
 - 価値の提唱

コンセプト
設計

チーム組成

設計と実装

成果の計測

プラットフォームを設計する

1. 認知負荷がかかる領域を特定
2. 開発チームが使っているツールやプロセスを棚卸し
3. ゴールデンパスを設計
4. ゴールデンパスを作成するうえで必要な、セキュリティ・ガードレールを特定
5. セルフサービスを実現するプロセスの策定

コンセプト
設計

チーム組成

設計と実装

成果の計測

プラットフォームを実装する

1. ゴールデンパスの実装
2. セルフサービスの実装

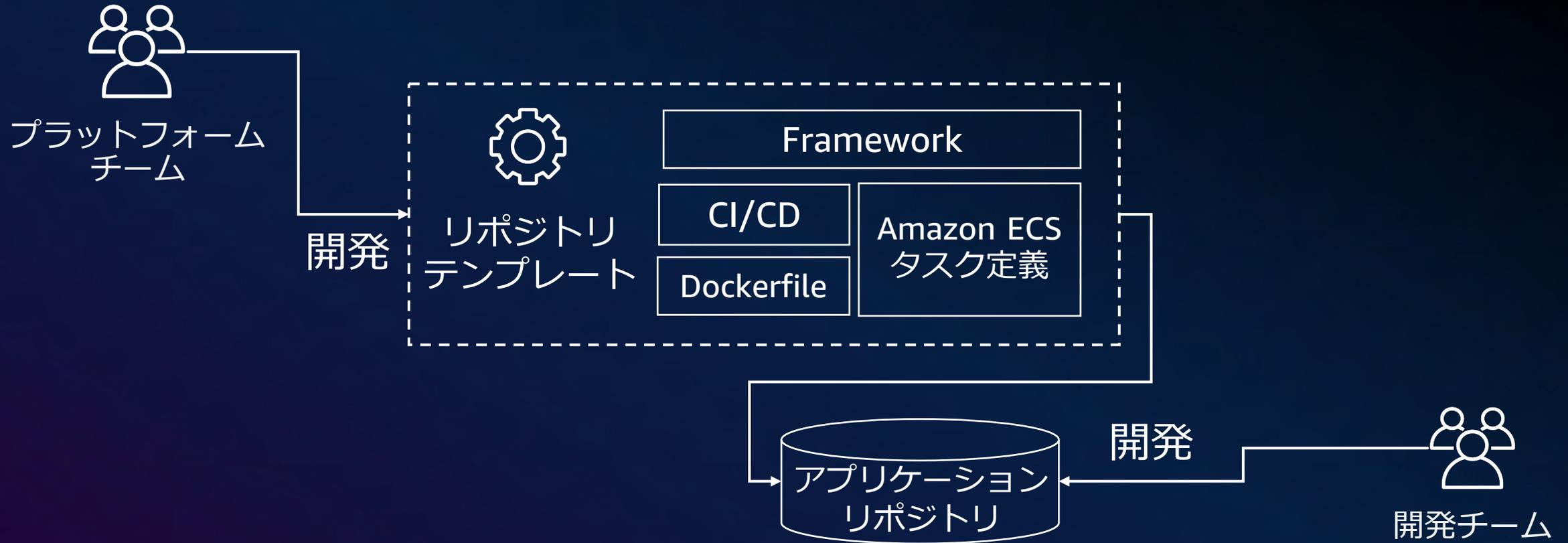
コンセプト
設計

チーム組成

設計と実装

成果の計測

ゴールデンパスの実装例：リポジトリリテンプレート



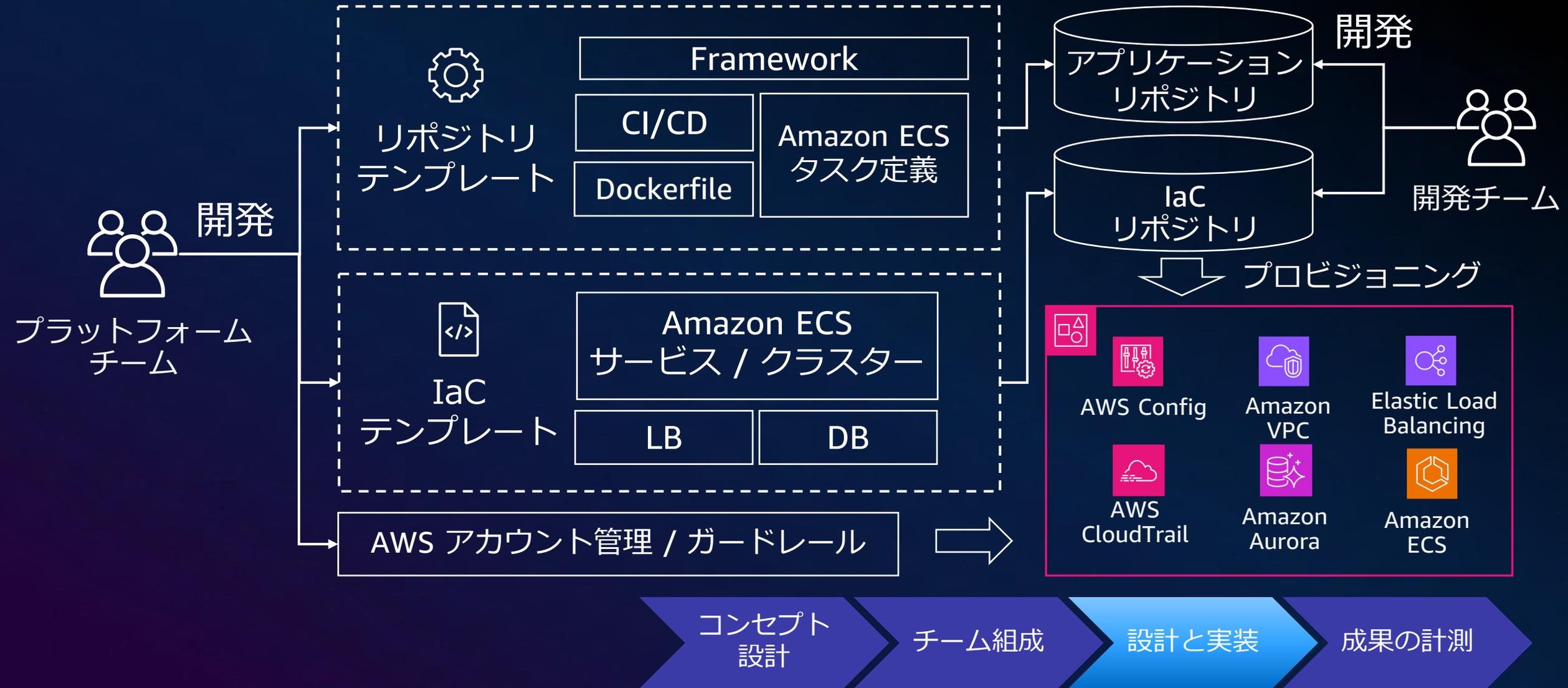
コンセプト
設計

チーム組成

設計と実装

成果の計測

ゴールデンパスの実装例：IaC テンプレート



セルフサービスの実装

セルフサービス ≠ 完全な自動化

e.g. IaC テンプレートの利用方法が書かれたドキュメントを見ながら開発チームが作業を進める

“ いちばんシンプルなプラットフォームは、
下位のコンポーネントやサービスについて書いた
単なる Wiki ページ上のリストだ。

マシュー・スケルトン、マニュエル・パイス著
『チームトポロジー 価値あるソフトウェアをすばやく届ける適応型組織設計』
日本能率協会マネジメントセンター, 2021

コンセプト
設計

チーム組成

設計と実装

成果の計測

Backstage

The screenshot displays the Backstage interface for a 'Demo Company Catalog'. The left sidebar contains navigation options: Home, AWS, APIs, Docs, Create..., and Tech Radar. The main content area shows a list of 'Owned components (6)'. The components are listed in a table with columns for NAME, SYSTEM, OWNER, TYPE, LIFECYCLE, DESCRIPTION, TAGS, and ACTIONS. The components are:

NAME	SYSTEM	OWNER	TYPE	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
@ demo-app		Everyone	aws-app	experimental	demo-app	aws, nodejs	[edit] [delete] [star]
@ demo-app-2		Everyone	aws-app	experimental	demo-app-2	aws, nodejs	[edit] [delete] [star]
@ demo-app-3		Everyone	aws-app	experimental	demo-app-3	aws, nodejs	[edit] [delete] [star]
@ demo-app-4		Everyone	aws-app	experimental	demo-app-4	aws, nodejs	[edit] [delete] [star]
@ demo-app-5		Everyone	aws-app	experimental	demo-app-5	aws, nodejs	[edit] [delete] [star]
@ demo-application		Everyone	aws-app	experimental	demo-application	aws, nodejs	[edit] [delete] [star]

Filters on the left include Kind (Component), Type (all), PERSONAL (Owned: 6, Starred: 0), DEMO COMPANY (All: 6), OWNER, TAGS, and PROCESSING STATUS.

- 内部開発者ポータルを実現するオープンソース
- 社内のソフトウェア資産を「サービスカタログ」として管理
- プラグインを活用したセルフサービス化の促進

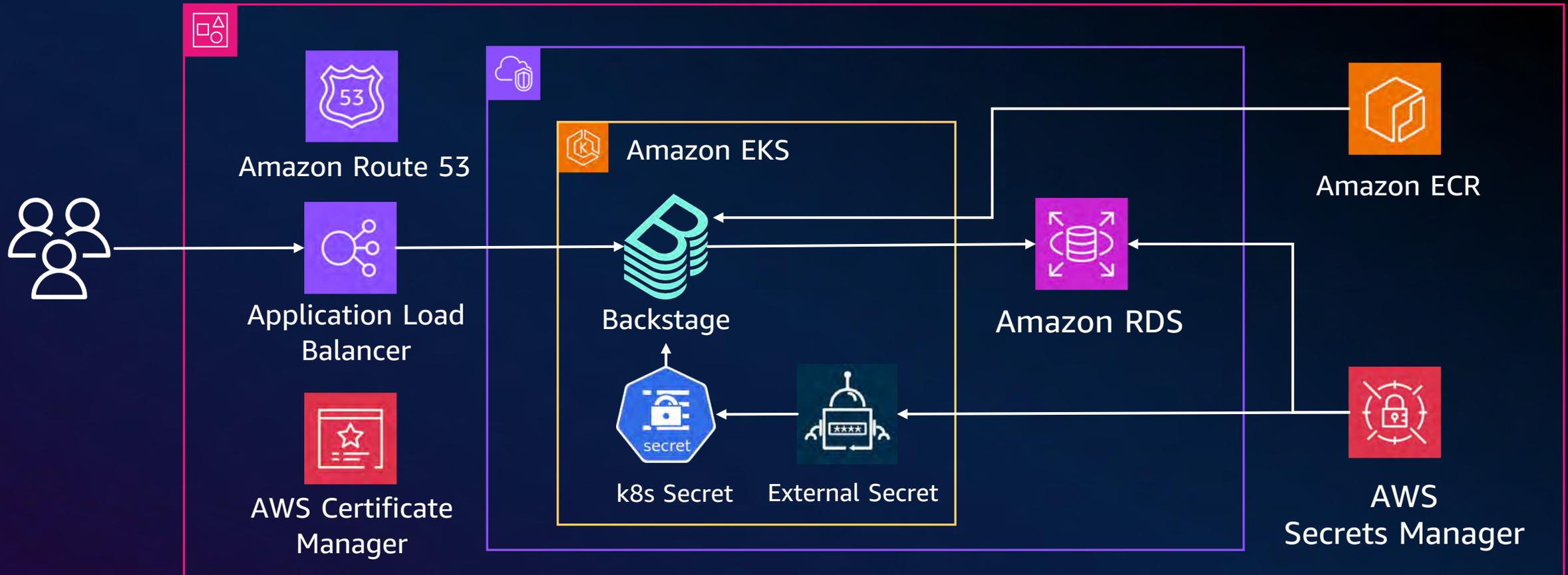
コンセプト
設計

チーム組成

設計と実装

成果の計測

Backstage on AWS



<https://aws.amazon.com/jp/blogs/news/building-developer-portals-with-backstage-and-amazon-eks-blueprints/>

コンセプト
設計

チーム組成

設計と実装

成果の計測

プラットフォームの成果を計測する

定量的な評価

- 開発チームの生産性
 - e.g. SPACE / DORA
- チームの疎結合さ
 - e.g. 依頼回数 / 待ち時間
- 組織の効率性
 - e.g. サービスリリースまでの期間

定性的な評価

- ユーザーヒアリング
- 定期的な開発者サーベイ

コンセプト
設計

チーム組成

設計と実装

成果の計測

まとめ

SaaS 開発とプラットフォームエンジニアリングの未来

プラットフォームエンジニアリングによる
さらなる開発生産性の向上



SaaS 開発を通じて得たプラクティスを
自社のプラットフォームに適用



ユーザーに対して、価値を継続して提供し続ける

おわりに

- プラットフォームエンジニアリングの目的は、
認知負荷の軽減による開発生産性の向上
- 様々なツールやサービスを組み合わせ、開発組織に
あった形で取り組む必要がある
- **AWS のソリューションアーキテクトにご相談ください！**

Thank you!

後藤 健汰 (Kenta Goto)
@kennygt51

アマゾン ウェブ サービス ジャパン合同会社
技術統括本部 デジタルサービス技術本部 ISV/SaaSソリューション部
ソリューションアーキテクト

