



SUMMIT
ONLINE
JAPAN

CUS55

Re : Chatwork レジリエンスなシステムを支えるAWS

~2011年から続く継続的アーキテクチャと組織の変革について~

春日重俊

執行役員兼CTO

Chatwork株式会社



1. 紹介
2. 継続的にシステム構成を変化できる重要性
3. SREの重要性
4. コンテナ技術の重要性と効果
5. 今後の展望

- ▶ こんな人たちに聞いてもらいたい
 - ▶ SaaSスタートアップの開発責任者
 - ▶ 大企業の新規事業開発責任者
- ▶ こんなことがわかる
 - ▶ 事業拡大する上での組織運営上の注意点
 - ▶ 事業ステージにおいて必要な技術スタックの考え方

● 板子（いたご）一枚下は地獄



01

紹介



- 名前：春日 重俊（カスガ シゲトシ） 40歳
- 所属会社：Chatwork株式会社（2016～）
- 役割：執行役員CTO
- 経歴
 - Chatwork
 - メッセージDBリプレイス、k8s導入、組織拡大やら
 - リクルート時代
 - ビッグデータ基盤組織立ち上げ、じゃらんnet検索エンジンリプレイス
 - Sier時代
 - seasar2が立ち上がった会計開発Prj



会社名

Chatwork株式会社

代表取締役CEO

山本 正喜

従業員数

134名 (2020年6月末日時点)

所在地

東京、大阪、ベトナム、台湾

ビジネスチャットツール「Chatwork」とは

- ・ 効率的なチャットによりメール・電話・会議の非効率性を解消
- ・ 2011年3月からサービス提供するパイオニアであり、日本国内で利用者No.1*1
- ・ 導入社数は27.3万社*2を突破



効率的に情報共有できる
グループチャット



仕事の見える化ができる
タスク管理



見落としがなくなる
ファイル管理

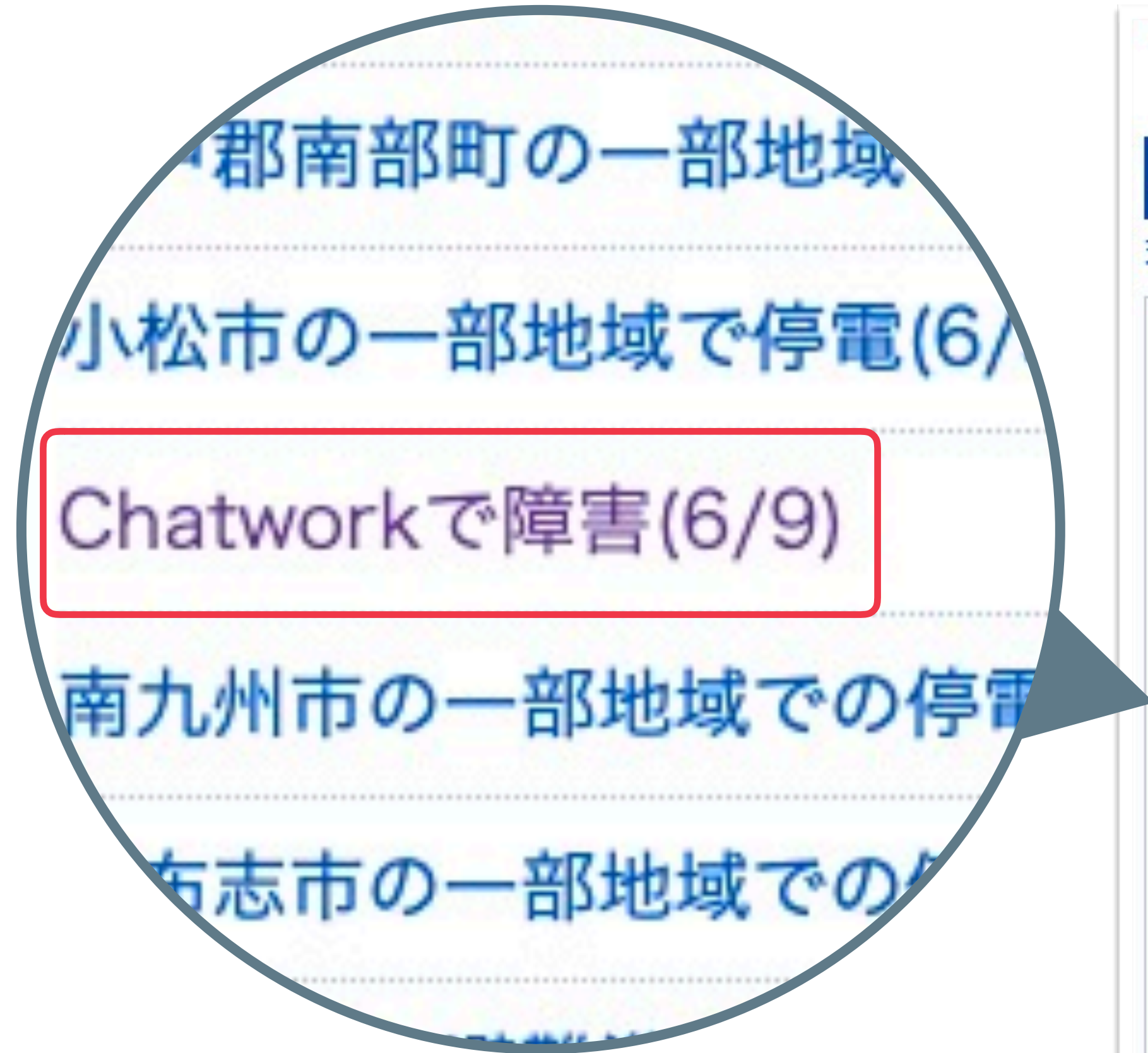


いつでも会議ができる
ビデオ/音声通話

*1 Nielsen NetView 及びNielsen Mobile NetView 「2019年5月調べ 月次利用者（MAU:Monthly Active User）」より。調査対象サービスはChatwork株式会社にて選定

*2 2020年6月末時点

先日Yahoo!さんに天気・災害サービスに認定されました



YAHOO! JAPAN 天気・災害 Masaki Yamamoto 0ポイント 新型コロナ 私たちの暮らしに

地名・施設名・郵便番号を入力 検索 パーソナル天気

天気・災害トップ > 緊急・被害状況

緊急情報

仙台市青葉区の一部地域での停電は復旧(6/9)	2020年6月9日 11時50分
長岡市の一部地域で停電(6/9)	2020年6月9日 11時40分
三戸郡南部町の一部地域で停電(6/9)	2020年6月9日 11時40分
小松市の一部地域で停電(6/9)	2020年6月9日 11時10分
Chatworkで障害(6/9)	2020年6月9日 11時00分
南九州市の一部地域での停電は復旧(6/9)	2020年6月9日 10時20分
志布志市の一部地域での停電は復旧(6/9)	2020年6月9日 10時20分
宮古島市で避難準備・高齢者等避難開始(6/9)	2020年6月9日 8時00分
鹿角郡小坂町の一部地域での停電は復旧(6/9)	2020年6月9日 7時00分
志布志市の一部地域での停電は復旧(6/9)	2020年6月9日 5時30分
鹿角市の一部地域での停電は復旧(6/9)	2020年6月9日 3時50分

誰もが簡単に使える



- ITを専門としないビジネス職をメインターゲットとして、機能やインターフェイスを設計
- 複雑なカスタマイズが不要で誰もが簡単に利用可能

オープンプラットフォーム



- 社内外をひとつのアカウントでシームレスにやりとり可能なオープンプラットフォーム型を採用
- 取引先やお客様との間で利用する事例が多数

フリーミアム



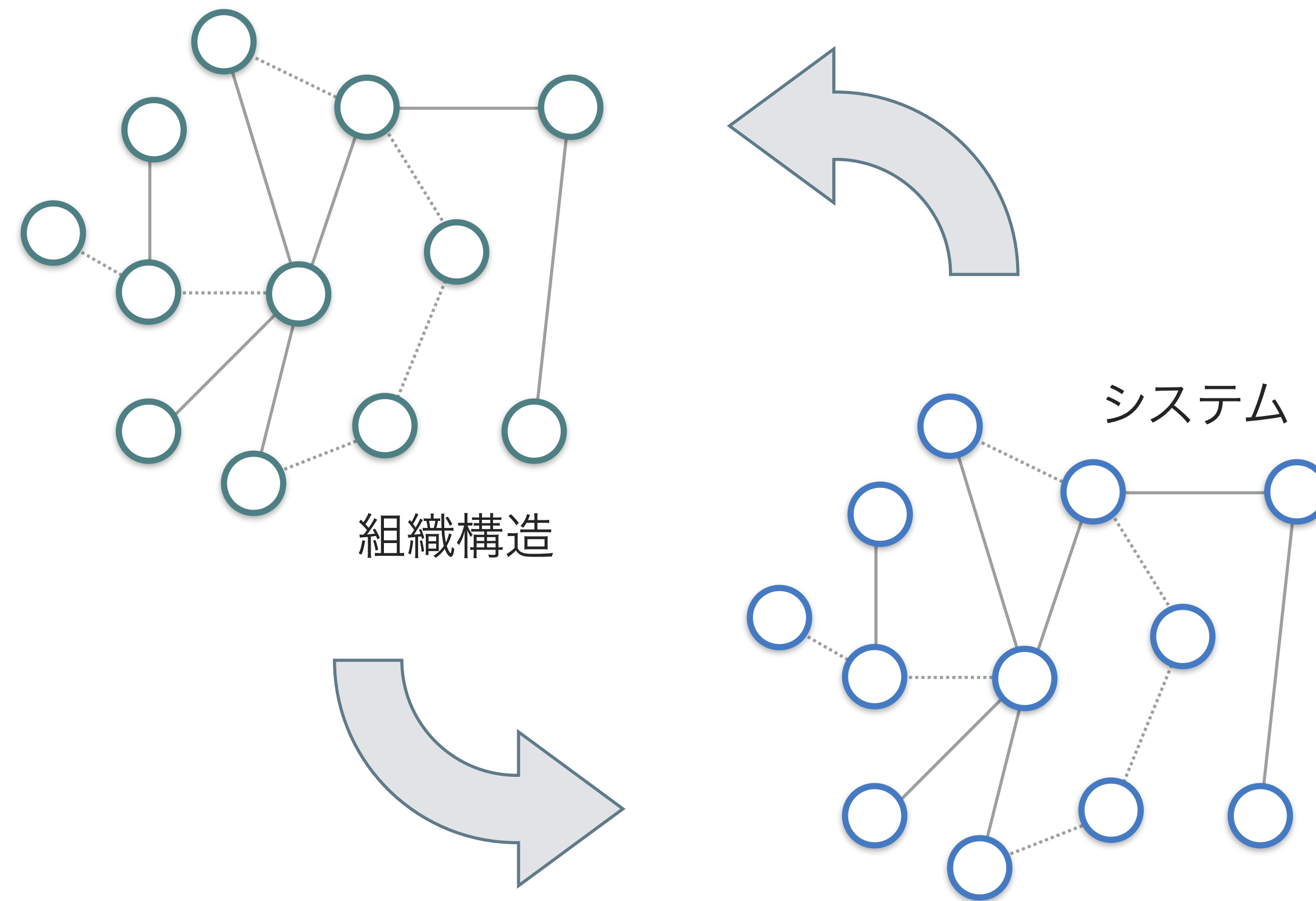
- 無料で期限がなく使い続けられ、活用が進むことで有料となる"フリーミアム"モデルでサービス提供
- 無料のプランがあることで、取引先やお客様にも気軽に勧められる

中小企業のビジネスコミュニケーション課題を解決します



02
継続的にシステム構成を
変化できる重要性

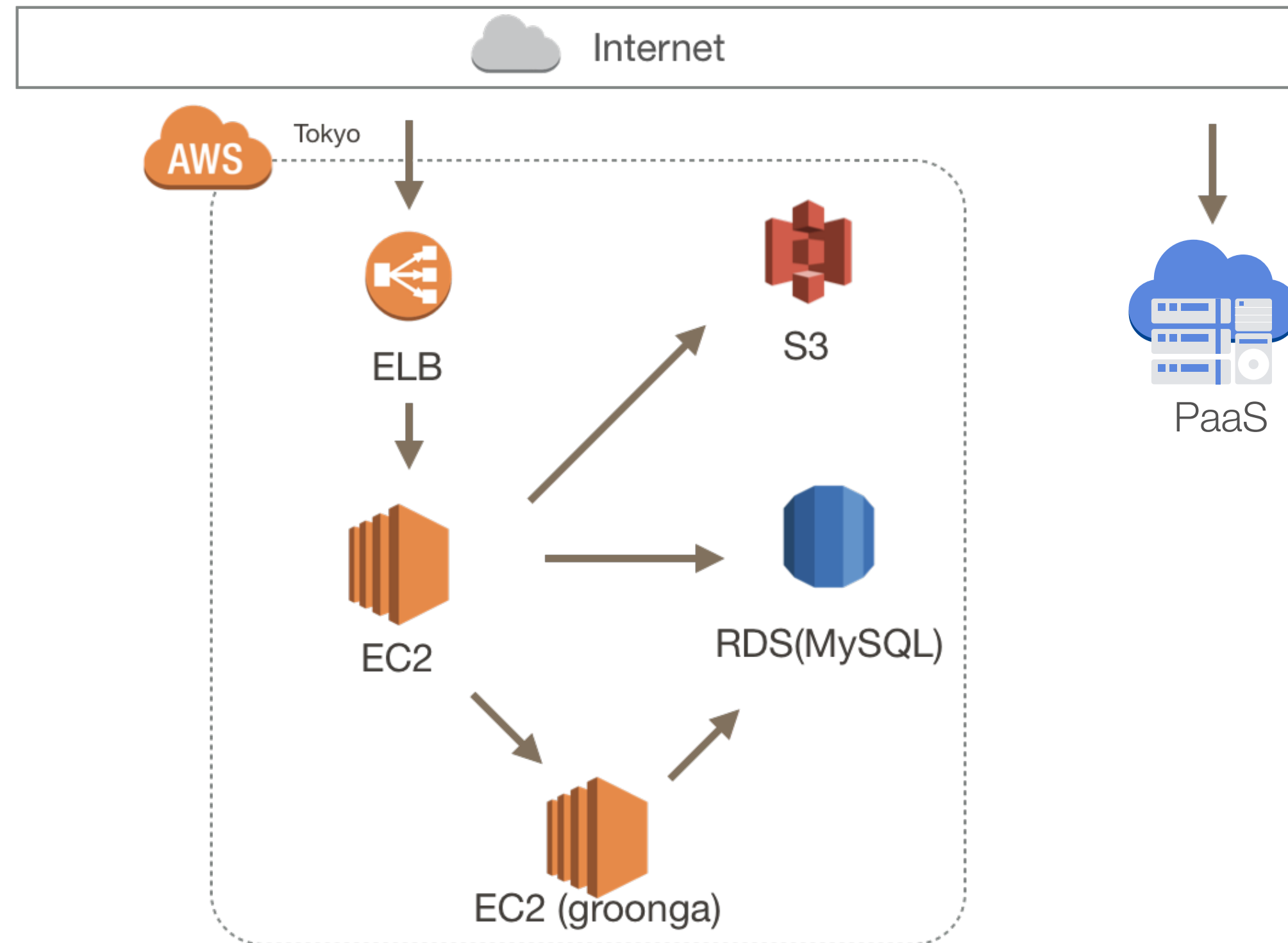
技術と組織は切り離せない関係性である By コンウェイの法則



元々は社内ツールの予定で開発していた



システム構成



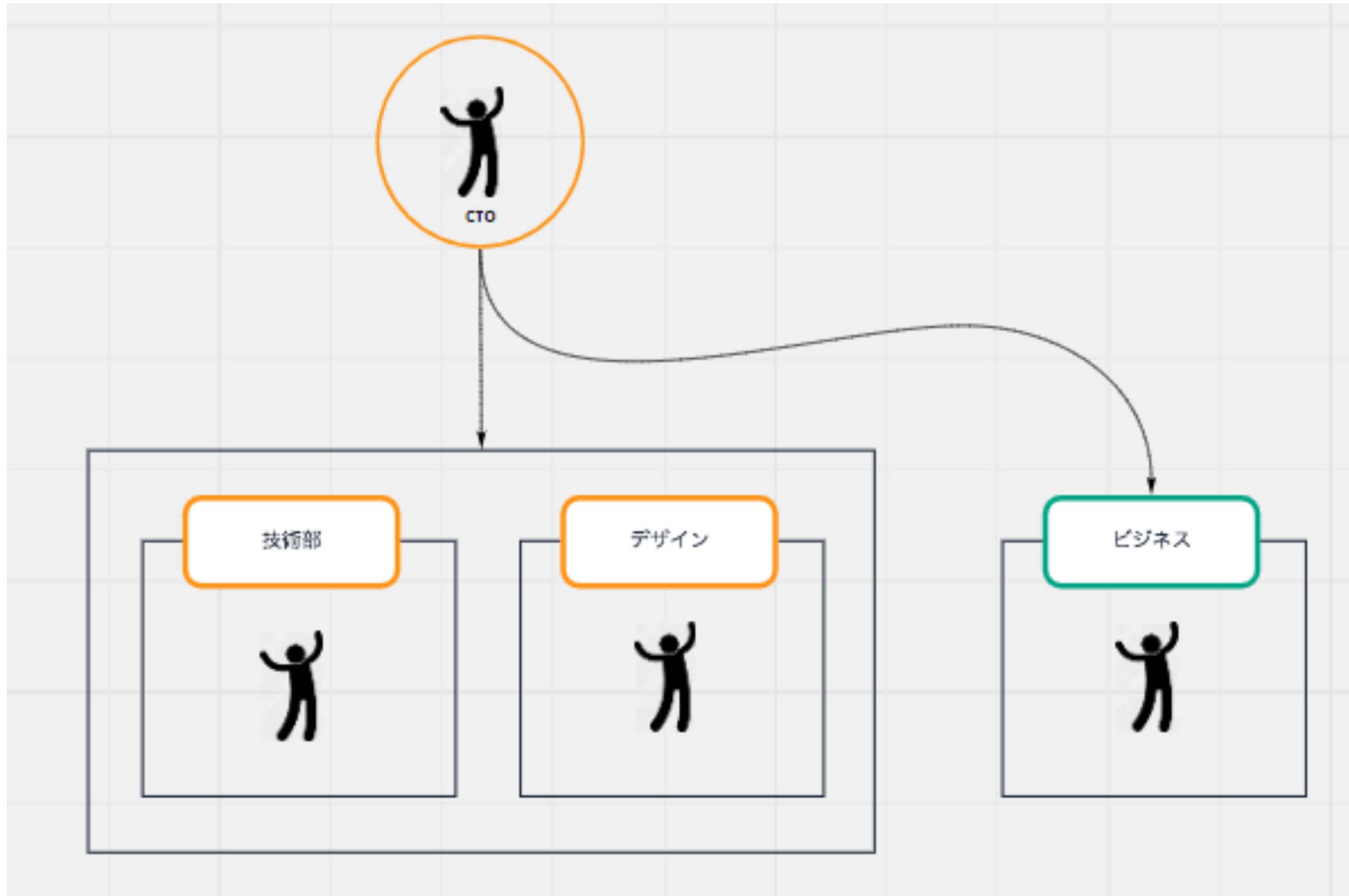
特徴

スタートアップらしく、スピード重視の

シンプルなLAMP構成。

AWS利用が9年前なので、早かった

組織構成

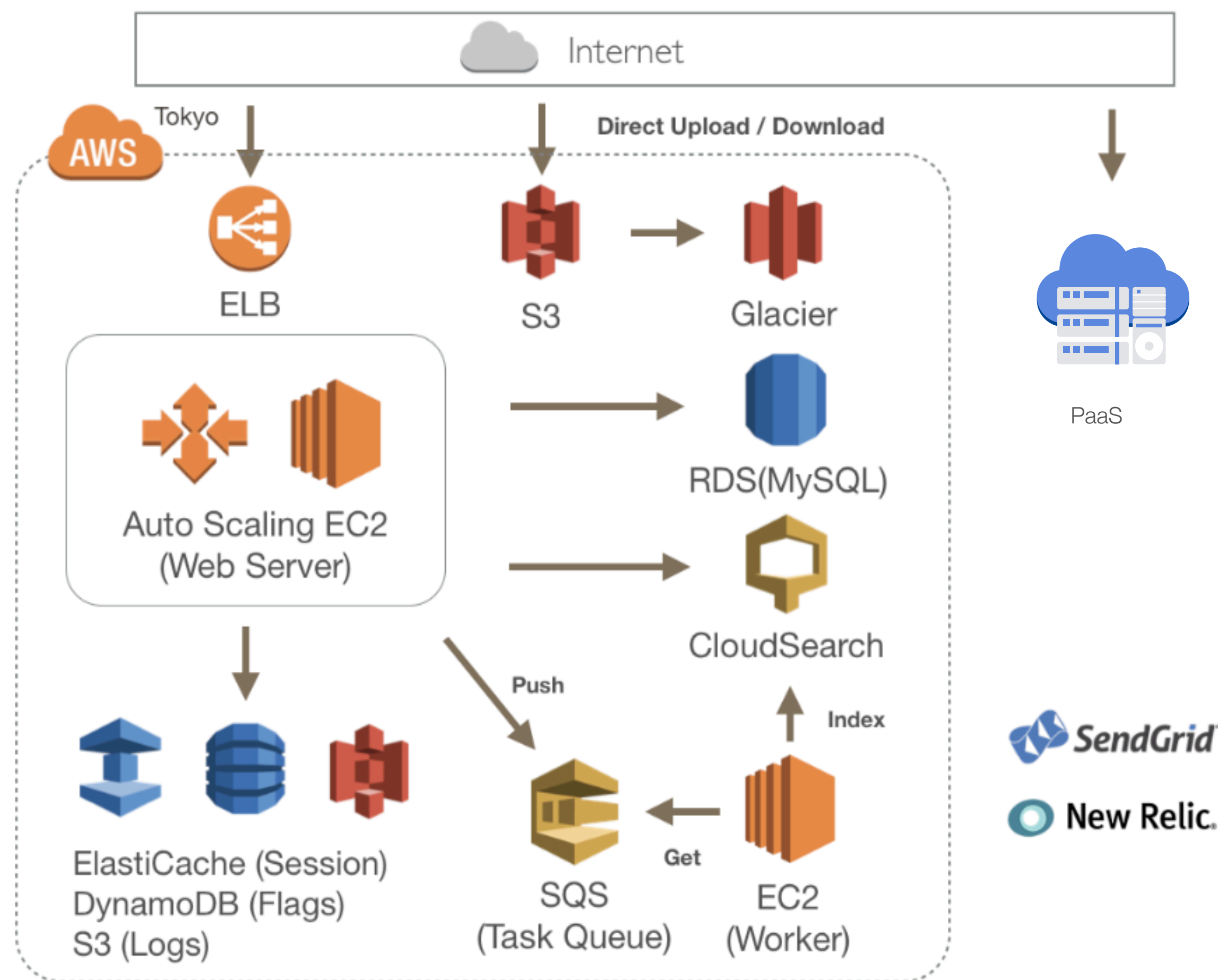


特徴

CTOが立ち上げた事業なので、ビジネス機能も全て担当しているフェーズ。



システム構成



特徴

導入企業数が5万社突破。

メッセージ数が5億以上になって

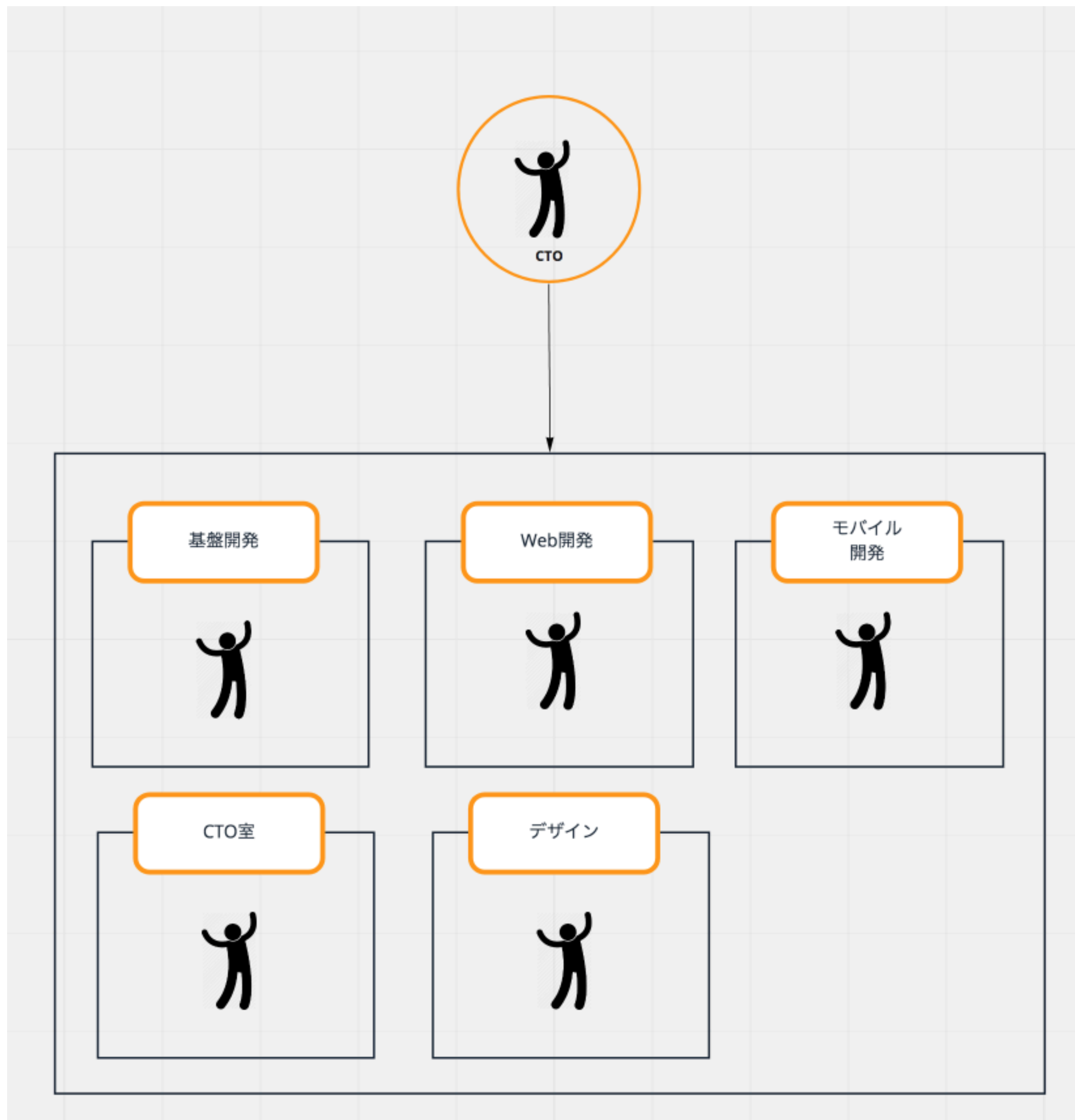
データレイヤー負荷爆発。

以下施策で乗り切る

- SQS(非同期システムの導入)
- Cloud Search(全文検索エンジン)
- ElasticCache (キャッシュ)



組織構成



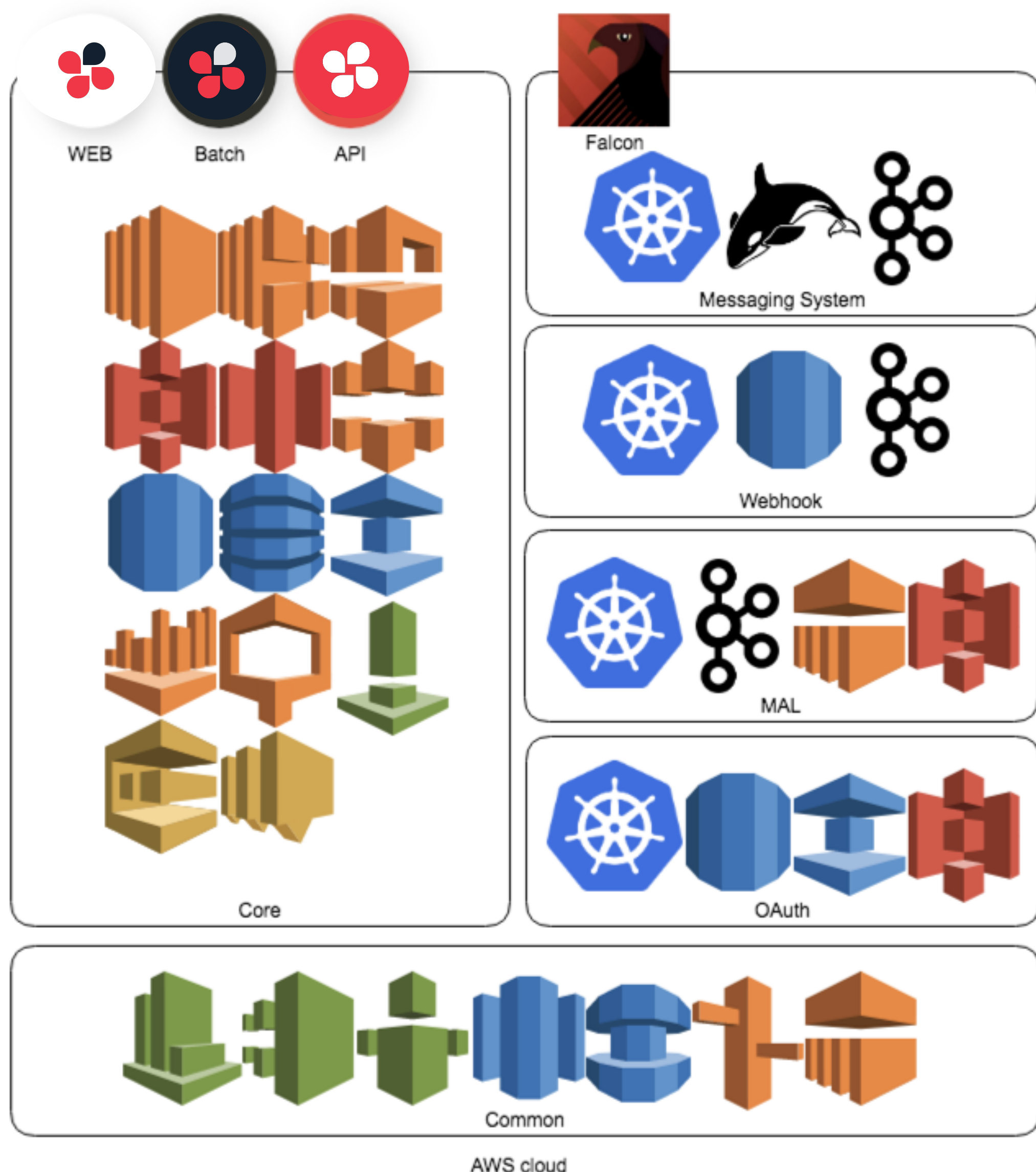
特徴

エンジニアが20名を超えてきたので、
提供機能別に組織を分割。

障害が多く、インフラエンジニアが
1名のみであった

分散システムの導入

AWS



特徴

導入企業数が10万社突破。

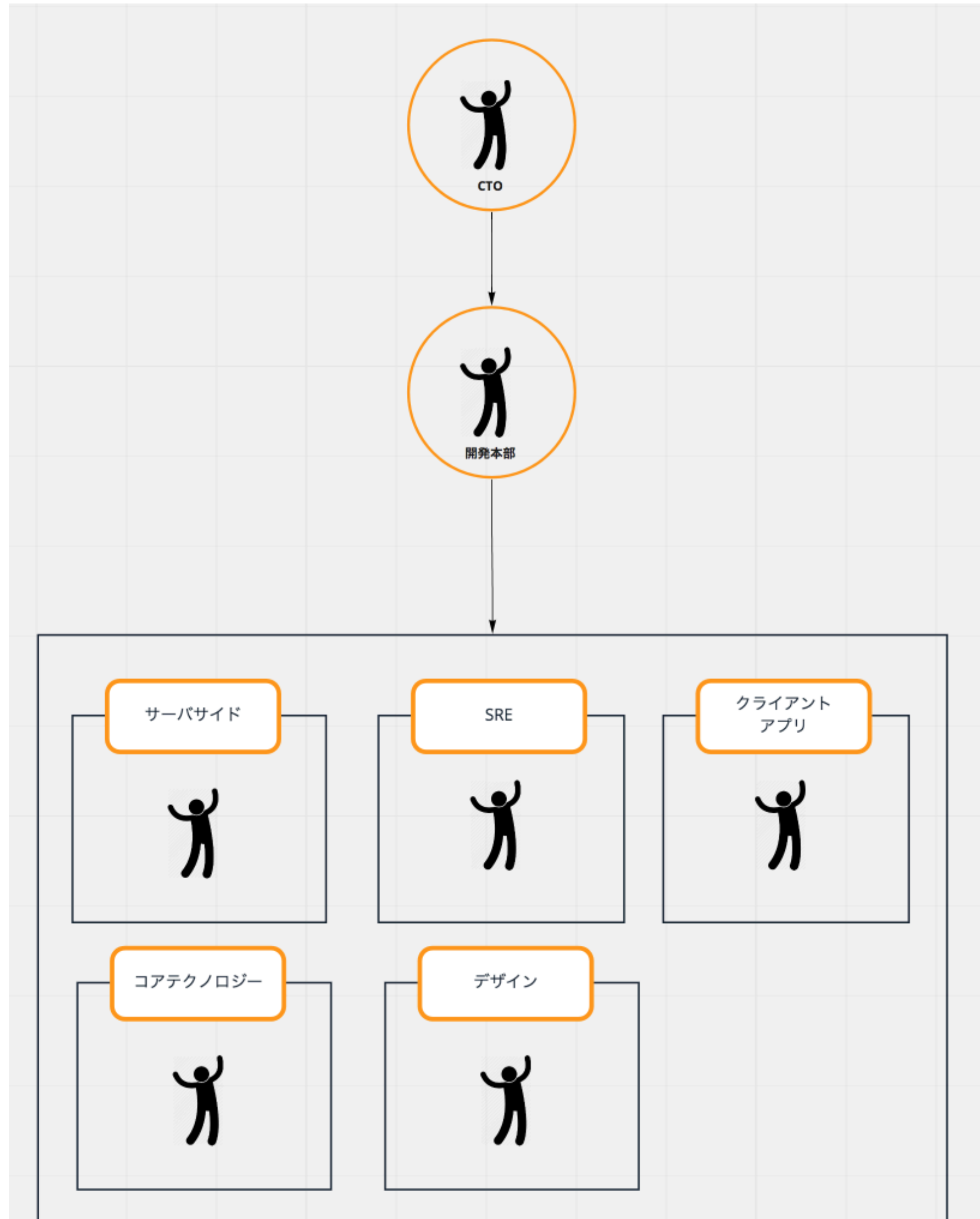
メッセージ数が20億以上になって

メッセージDB部分に分散システム導入

以下アーキテクチャで乗り切る

- Kafka
- Hbase
- Akka
- CQRS+Event Sourcing

組織構成



特徴

CTOが経営レイヤーの仕事が増えたので

開発本部機能を導入。

また、インフラ安定化を担う

SREを専属部署として設立。

クライアントとサーバサイドは意図的に一度集約。

全アプリ Kubernetes導入



特徴

導入企業数が27万社突破。

メッセージ数が60億以上になっているが

継続的なアーキテクチャ刷新により

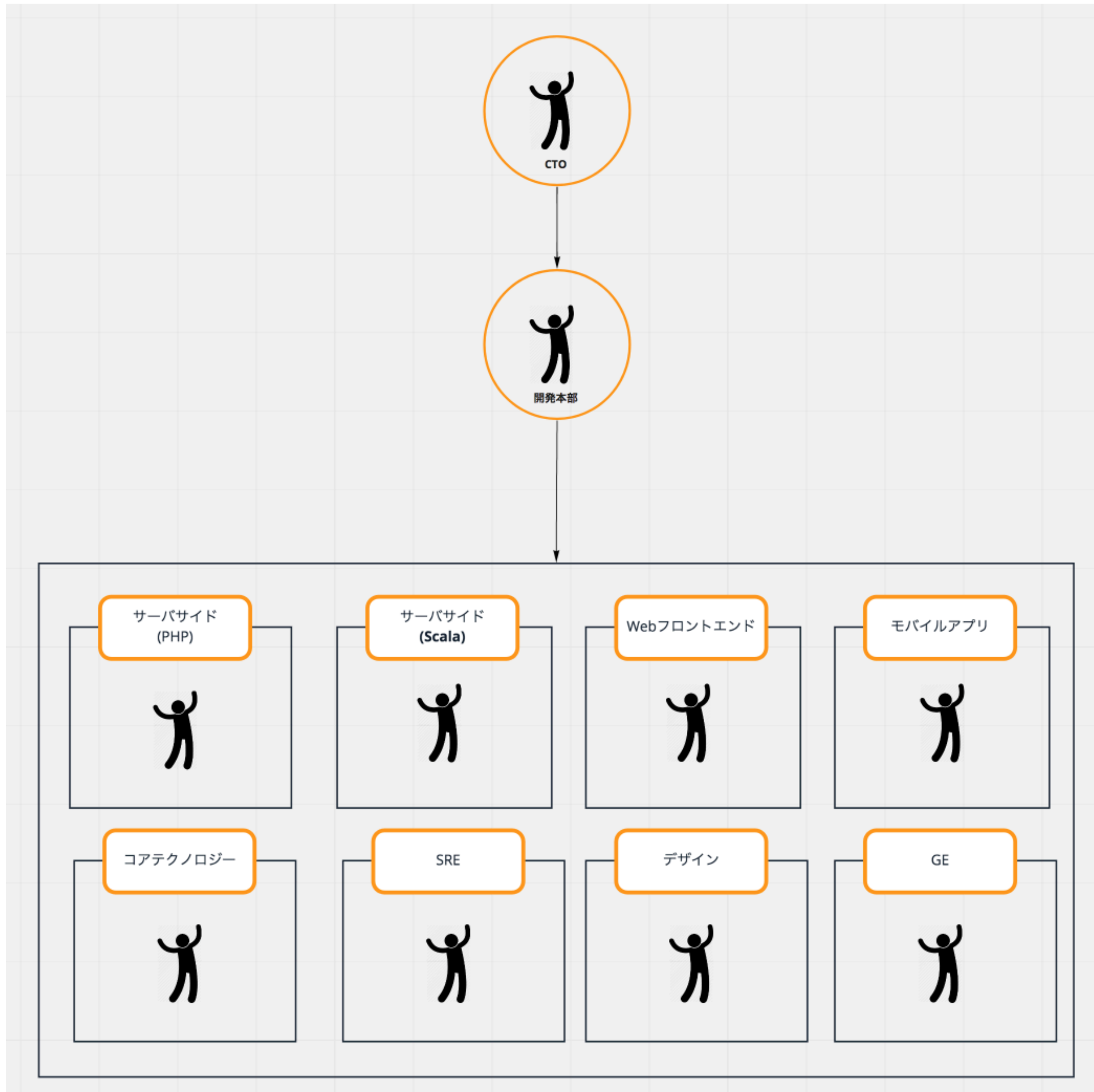
安定的なサービス運営実現

業務ロジックの大半を占める

PHPアプリケーションにもKubernetes

導入済み

組織構成



特徴

CEOがCTO兼務状態から解消。

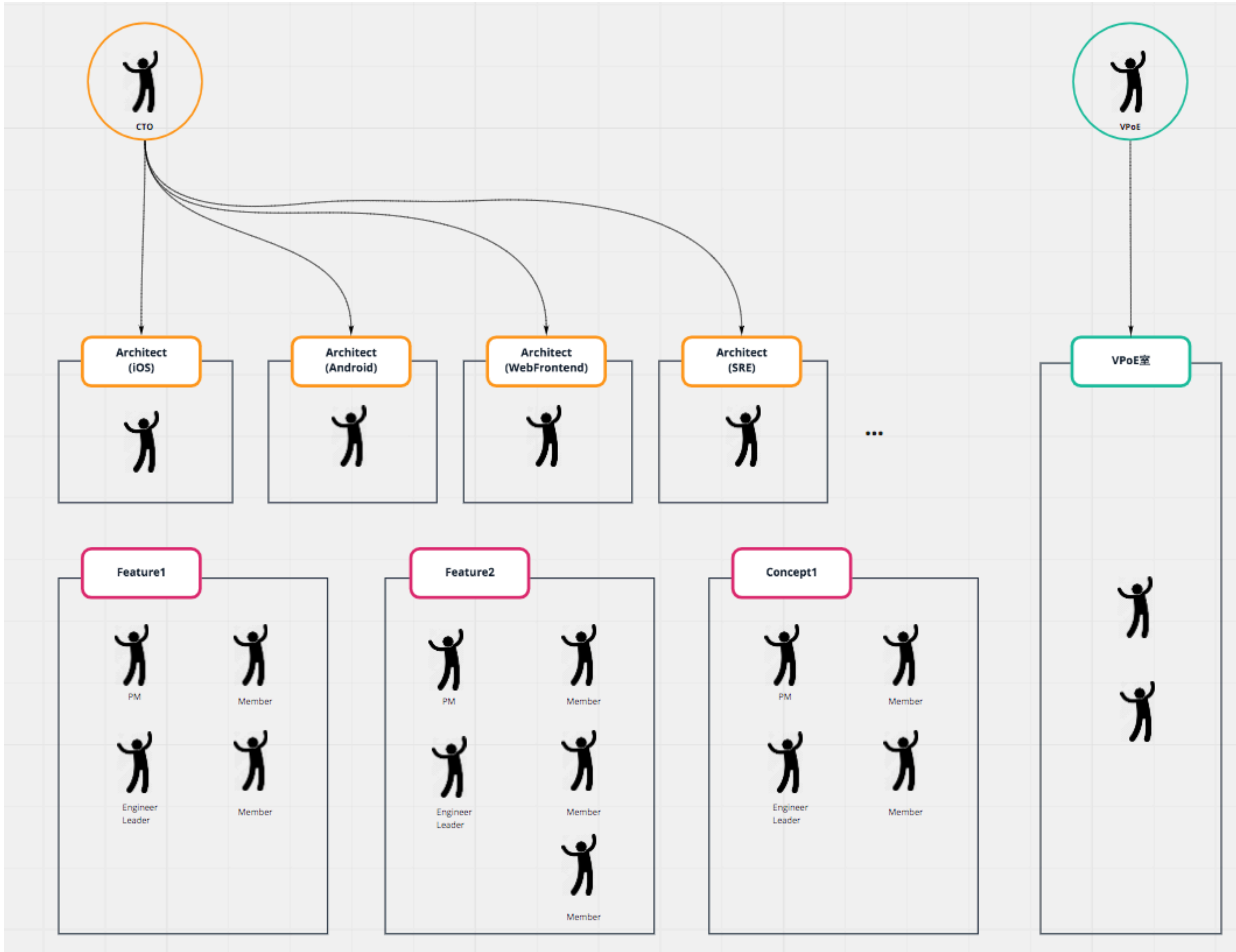
メンバーも60名近くなってきたので

技術縦割り組織から

ユーザグロース部分を一部機能組織化に

チャレンジ中

組織構成



特徴

機能別組織・技術支援組織・ピープル

マネジメントの3つの役割で

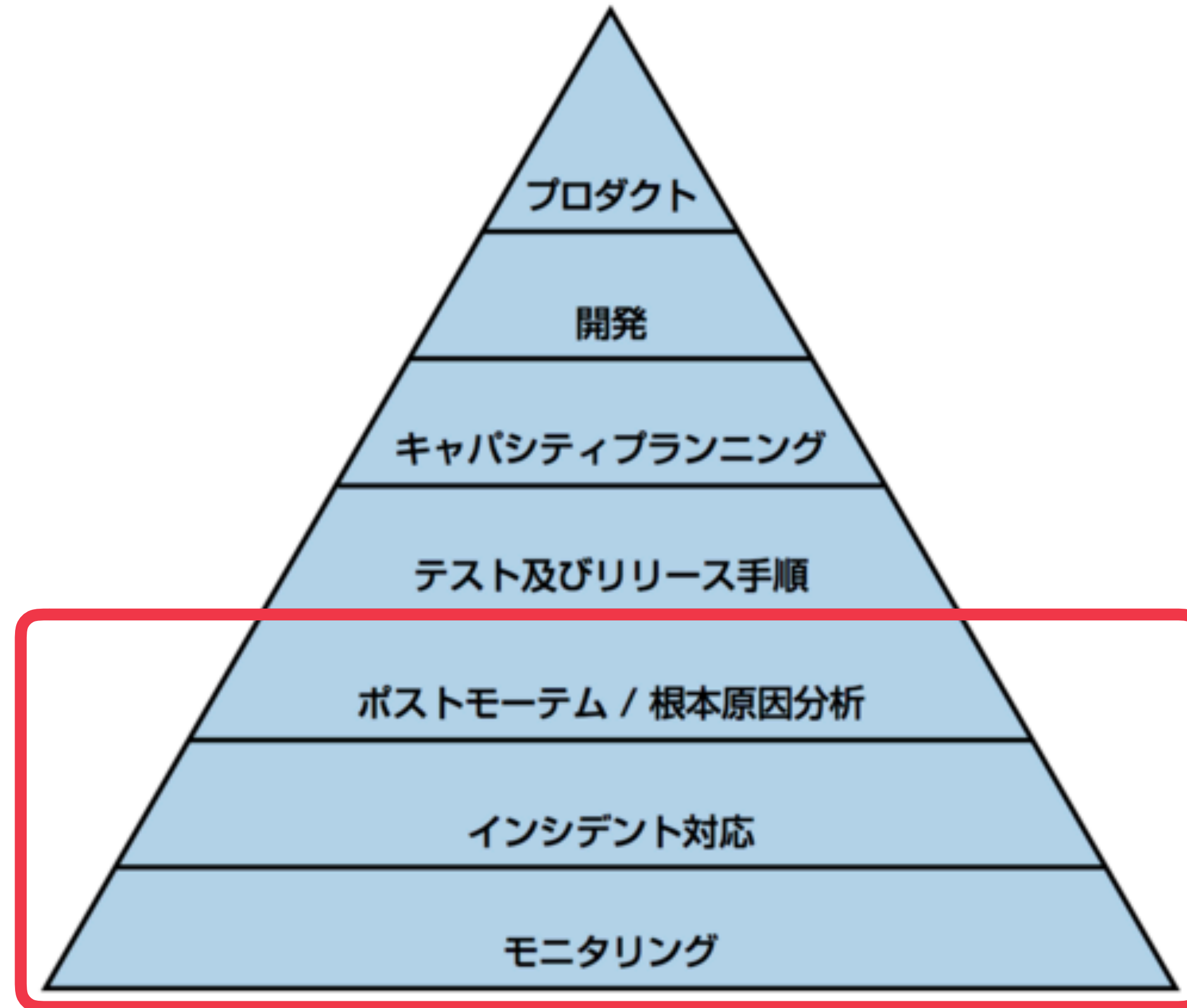
組織が分割されていて、スケールできる

体制が構築されている

03

SREの重要性

サービス信頼性階層



Source: "SRE サイトリライアビリティエンジニアリング", オライリージャパン, p108図III -1サービスの信頼性の階層)



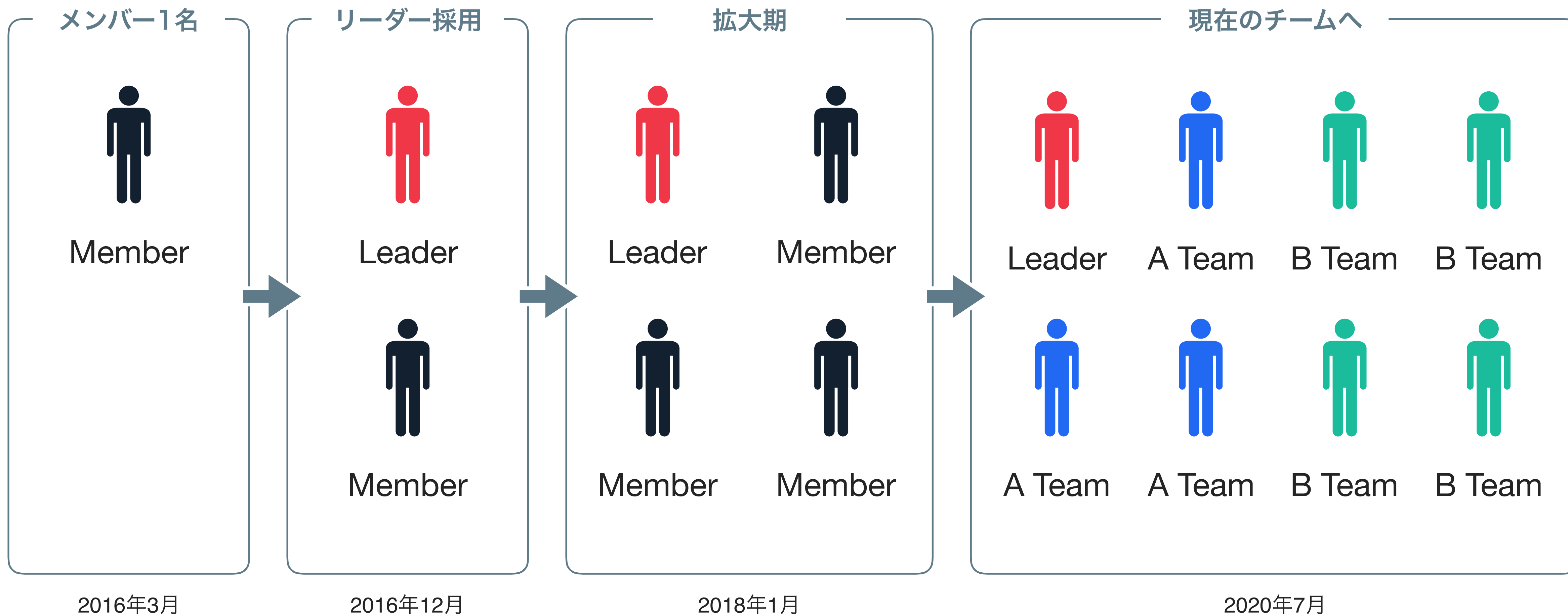
Why is SRE important in chatwork?

ビジネスチャットというサービスの特性上低レイテンシーで要求が厳しい





- ソフトウェアをインストール・稼働させるノードの数が増え、100や1000ノードになることもある。
- 扱うデータ量もギガバイト単位から、ペタバイト単位に増えた。
- 以前はデータが変化するタイミングは限定的で日次のバッチ処理などで処理されていたが、常にデータが変化するようになり、障害が起こればその間に処理できなかったデータを処理して追いつくのは困難なこともある。
- 以前は長いシステムメンテナンス時間があったが、今ではそれは受け入れられない。
- 解決したいのは技術的な課題ではなく、ユーザーの期待の変化である。
- 社会インフラ
- 利用できなかつたり、不満があれば、ユーザーは他のよりよい選択肢を探し始める





Postmortems Culture

障害記録: Chatwork瞬断

概要
www.chatwork.comでの10%リリース中のSTRIXへALB経由でのアクセスができなくなり、そちらに繋がったChatworkユーザーがサービスを利用できなくなった。

臨時チャット

インシデント種類
システム障害

インシデントレベル
軽度障害
NewRelicの外形監視的には、のサービスダウン

1 violation

Severity	Location	Alert condition	Opened	Closed	Duration	Alert policy
CRITICAL	Tokyo, JP	SLA check (web)	11:58:53	11:59:21	27s	SLA check default no Pager D...

事象発生期間

事象発見経緯

- 発見日:
- 発見者:
- 経緯: オフラインになったので、何度か試したところまたオフラインになったのでSlackで報告

影響範囲

- ECWユーザーの10%
- TTLが300s (5m) のため の間にアクセスした10%のユーザーが影響を受けた

The image shows a screenshot of a NewRelic incident report for a 'Chatwork瞬断' (Chatwork outage). The report includes a summary, incident details (severity: CRITICAL, location: Tokyo, JP, duration: 27s), a table of violations, and performance graphs. The graphs show a spike in web transactions time and a drop in Apdex score during the outage period, with throughput also showing a slight dip. The incident was caused by a 10% release of STRIX to ALB, which resulted in 10% of users being affected for 5 minutes due to a 300s TTL.

特徴

発生した重大インシデントはポストモータム導入済み。

特に重要なのは振り返り中では

ポジティブなフィードバックで統一！！！！

次回防止のアクションまで時間かけても

定義することが大事



ADR (Architectural Decision Records)

特徴

継続してシステムを成長させるために
 アーキテクチャを常にアップデートし続ける
 必要があるが、そのアーキテクチャが採用され
 た背景を明記することで、途中参加メンバー
 も選択背景を理解してもらえる。

008. メトリクス収集

目次

- 目次
- コンテキスト
 - Datadog with Kubernetes
 - 収集しているメトリクスと代替方法
- 決定事項
- ステータス
- 結果
- 関連リンク
- 関連チケット

コンテキスト

現在、CW WebではPHPと同一のEC2インスタンス上にDatadogAgentを起動して、そこでメトリクスを収集している。
 STRIXではコンテナになるため、コンテナのベストプラクティスから同じコンテナ内にAgentを配置することはできない。

Chatwork社ではインフラ系のメトリクスは全てDatadogに集約しており、アラートの作成なども行なっているため、本ADRでは切り替えは切り替えは検討しない。

Datadog with Kubernetes

DatadogではKubernetesに対応しているため、基本的にここに乗りかかっていく形になる。

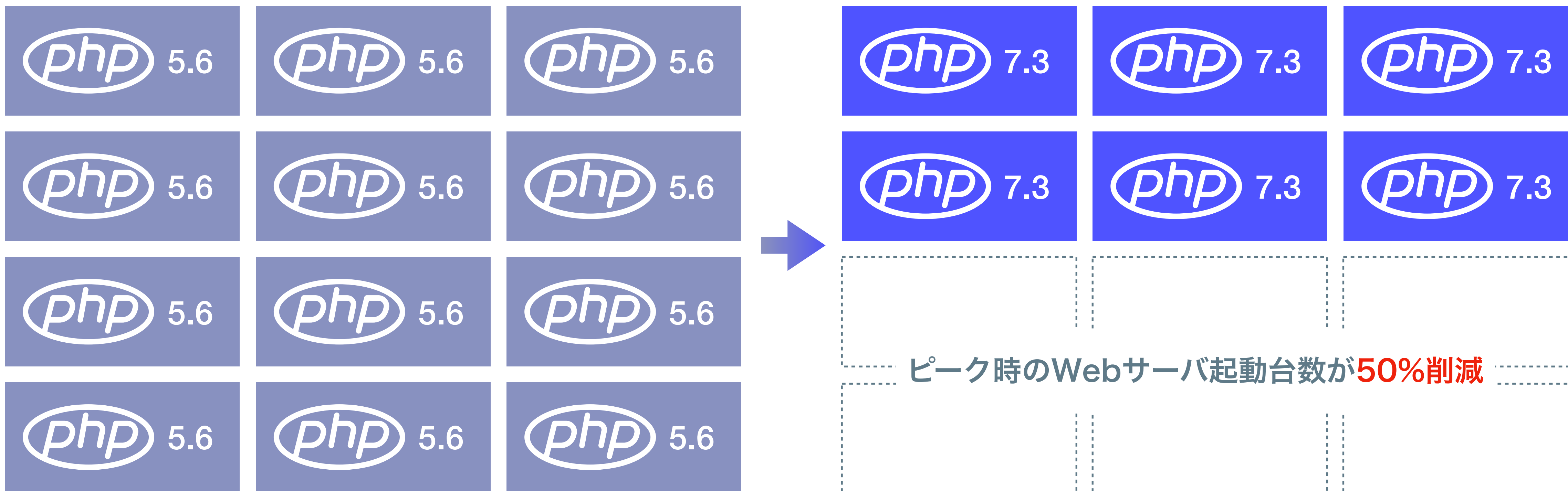
収集しているメトリクスと代替方法

現行の設定:

		代替方法	
System Metrics	<ul style="list-style-type: none"> CPU Memory Network Disk 	<ul style="list-style-type: none"> Kubernetes Pod Monitoring 	
NGINXのメトリクス	<ul style="list-style-type: none"> Connection 	<ul style="list-style-type: none"> CloudFront、ALB、NewRelicのメトリクスで代替可能なため不要 	
PHP-FPMのメトリクス	<ul style="list-style-type: none"> Backlog queue Process Request 	<ul style="list-style-type: none"> PHP FPM Exporter 自作Exporter <ul style="list-style-type: none"> PHPアプリにエンドポイントを追加 	Twistlock、NewRelicの影響によりコンテナ数、CPU数が増えると料金に影響が出るという課題がある
Fluentd	<ul style="list-style-type: none"> Buffer Retry Count 	<ul style="list-style-type: none"> Kubernetesのログ収集の仕組みで別途考える 	
Postfix	<ul style="list-style-type: none"> Queue 	<ul style="list-style-type: none"> Postfix Exporter 	
TCP	<ul style="list-style-type: none"> DBのEndpoint <ul style="list-style-type: none"> cw-vpc-application-cluster Writer Endpoint cw-vpc-application-cluster Reader Endpoint ec-vpc-application ec-vpc-readreplica ec-vpc-readreplica-writabl 	<ul style="list-style-type: none"> TCP Checkを各ノードで実行する 	



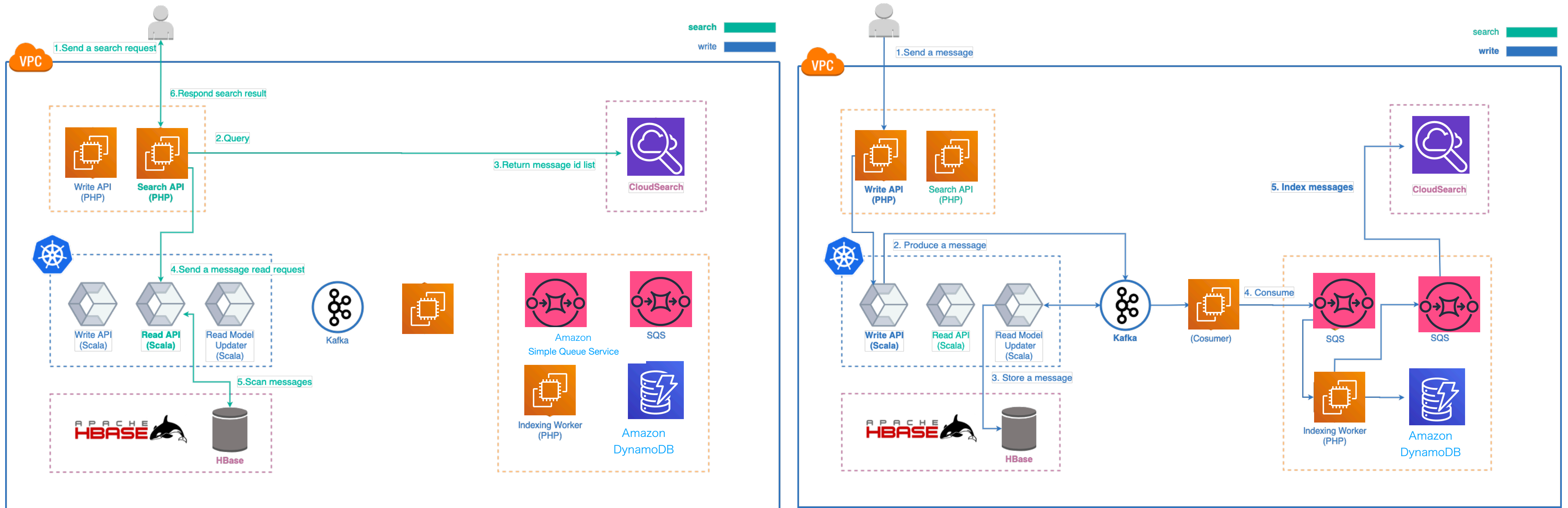
php7対応



レスポンスタイムが**1/2**に

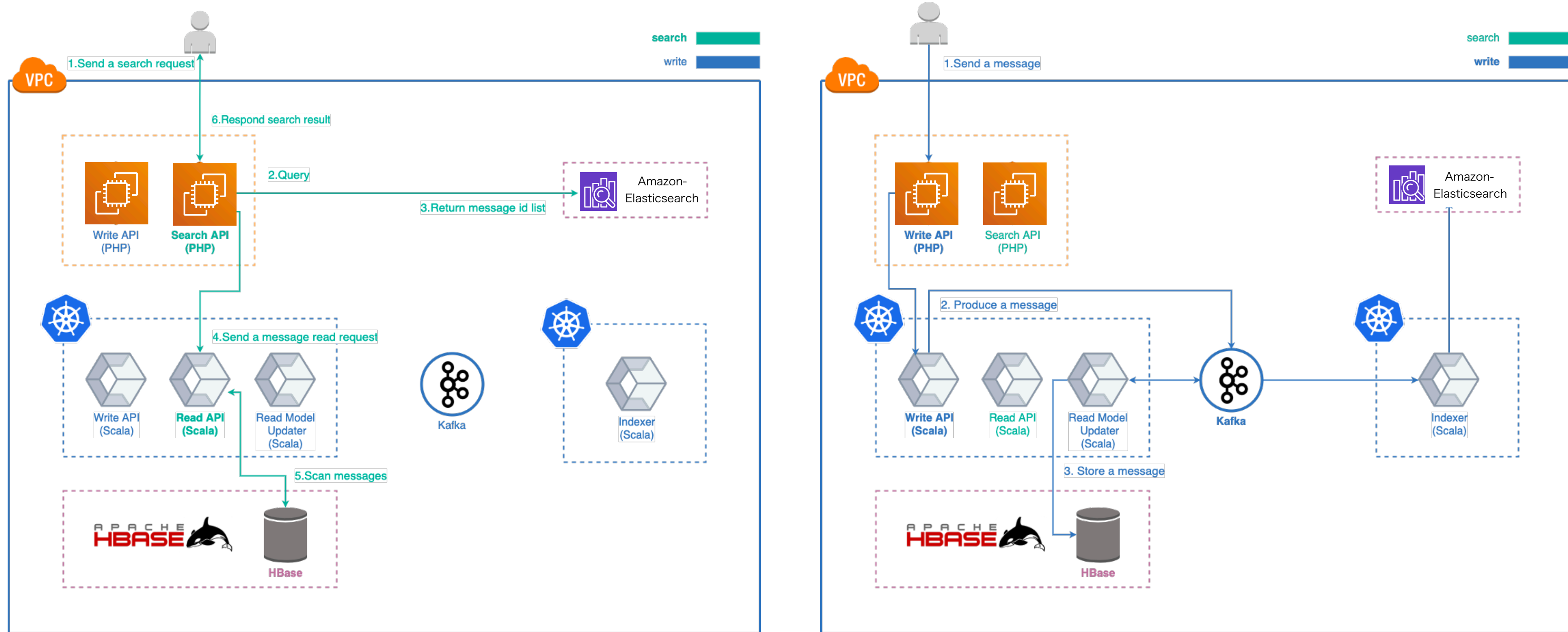
検索エンジンリプレース

Before



検索エンジンリプレース

After



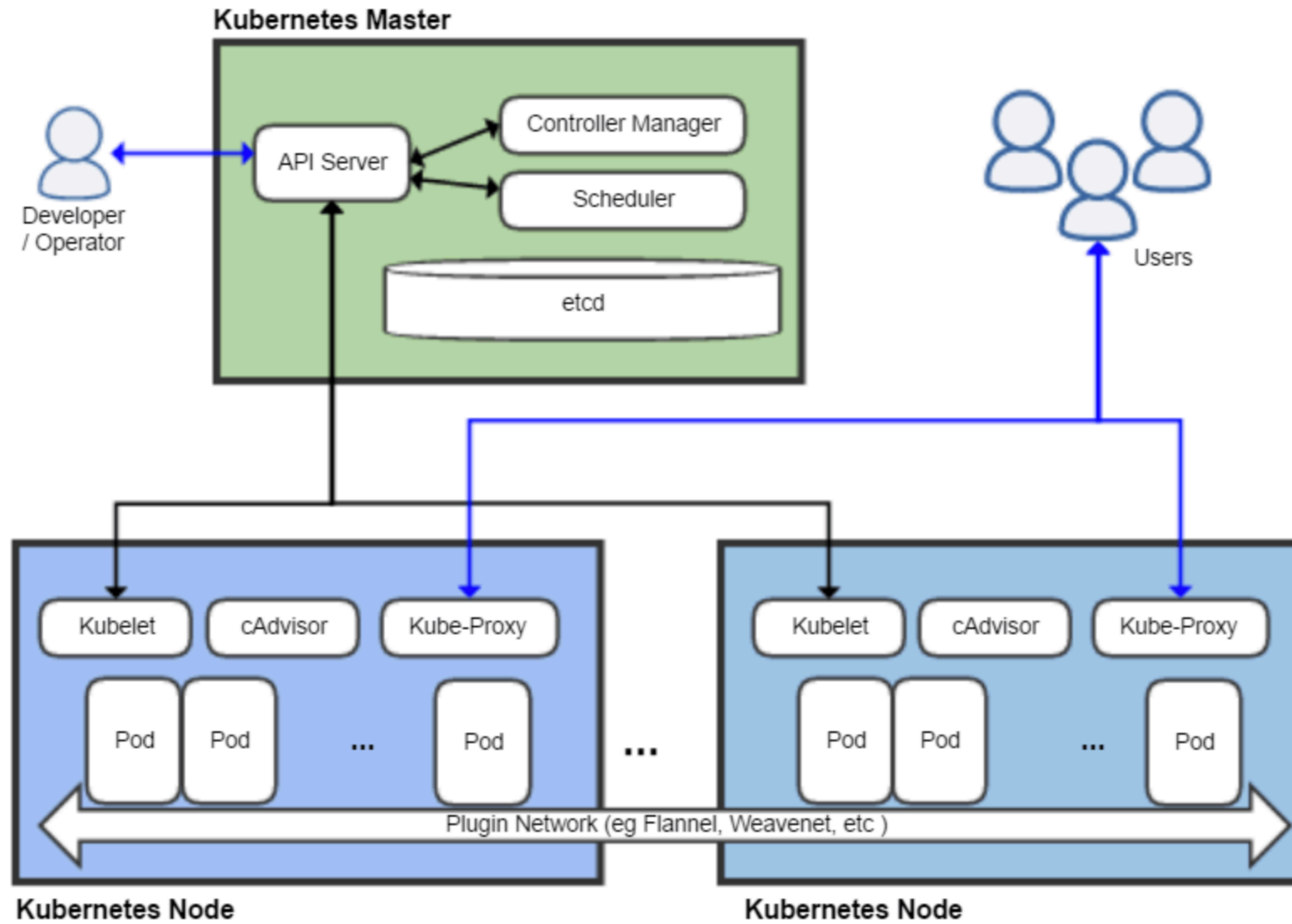
サーバ台数が **1/10** になりランニングコストが **1/6** & Index 反映時間が

ほぼリアル

04

コンテナ技術の重要性と効果

What is kubernetes?



Source:wikipedia kubernetesのアーキテクチャ図



What is the intrinsic value of kubernetes?

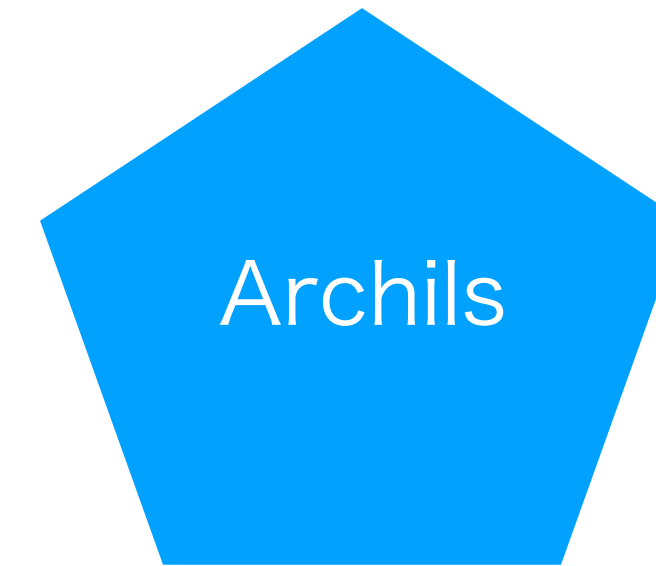
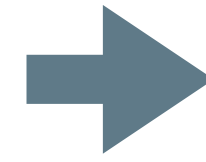


Speed!Speed!Speed!Speed!Speed!Speed!

管理



kube-aws



CD



+



flux

構成管理



セキュリティ





柔軟なデプロイ技術(k8s)

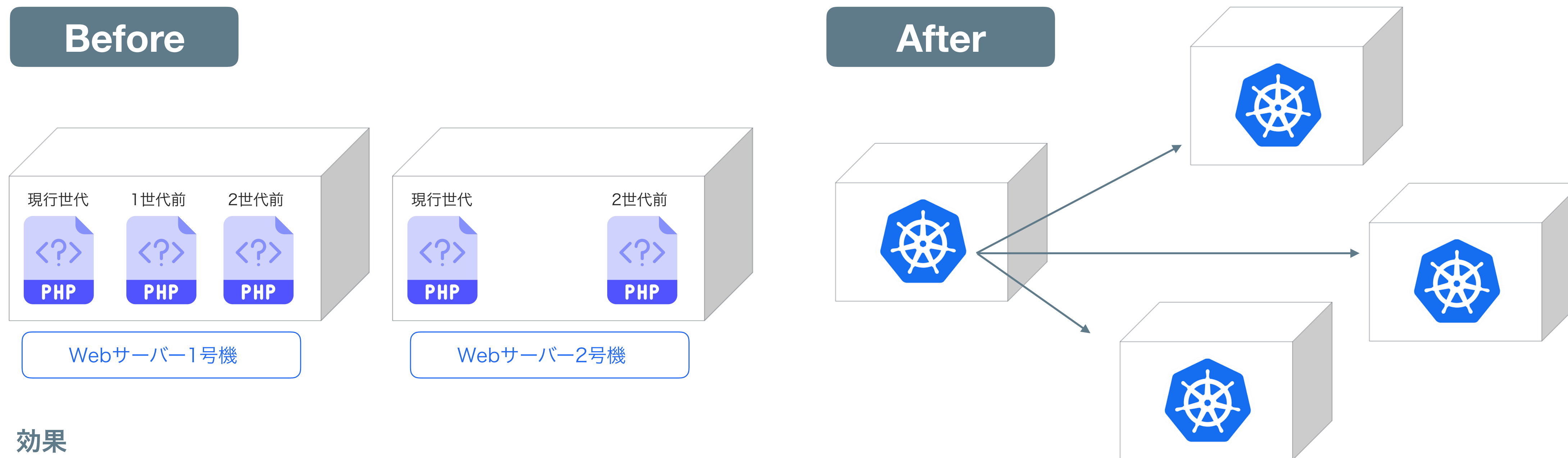
```
./achlis born --aws-profile=chatwork --flux-git-branch=eks
```

これだけでEKS環境OK

```
./helmctl apply --aws-profile=chatwork
```

これだけでKubernetes内部もOK

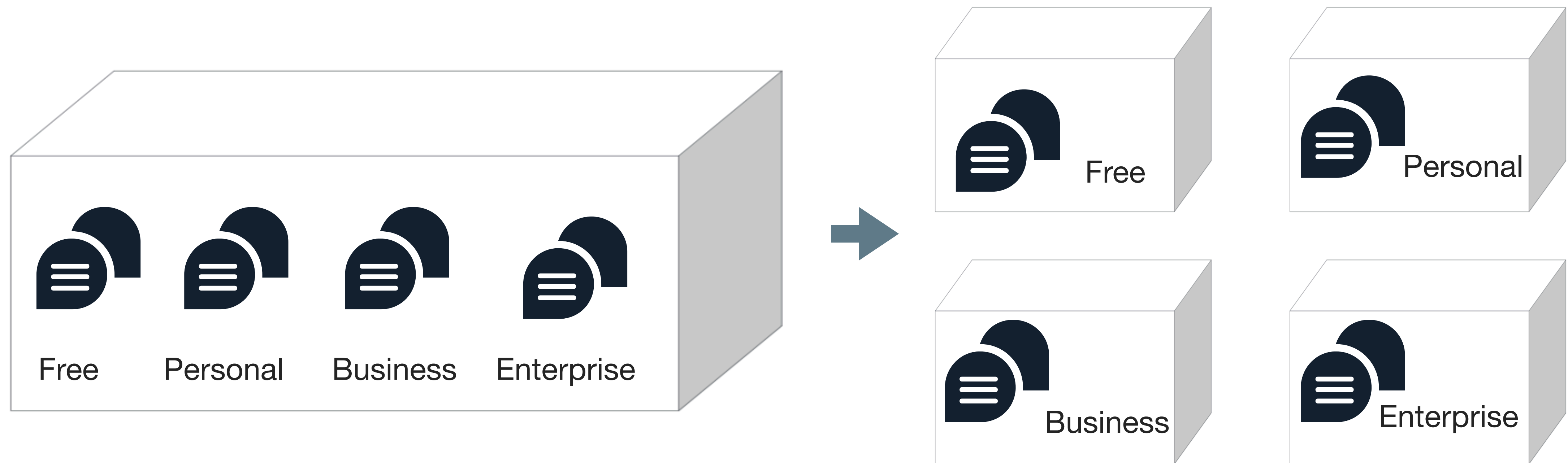
止められないサービスに適した高可用性な基盤



効果

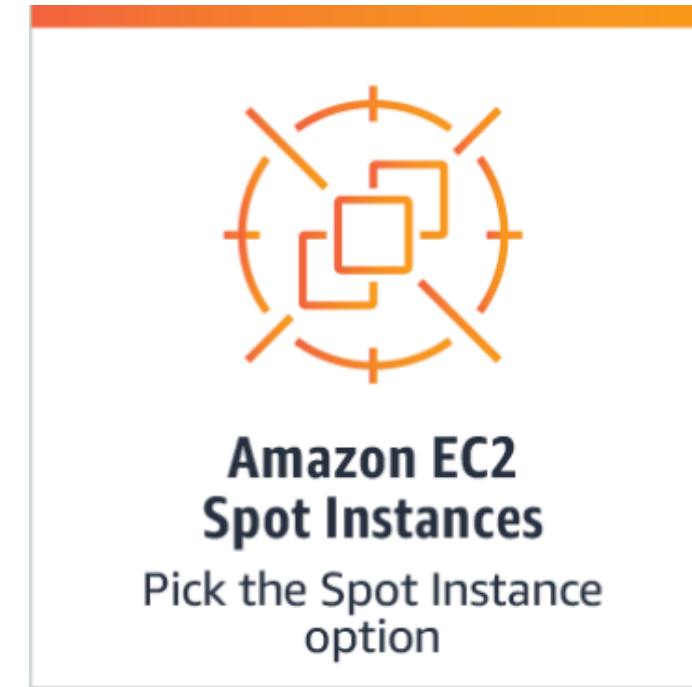
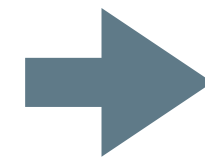
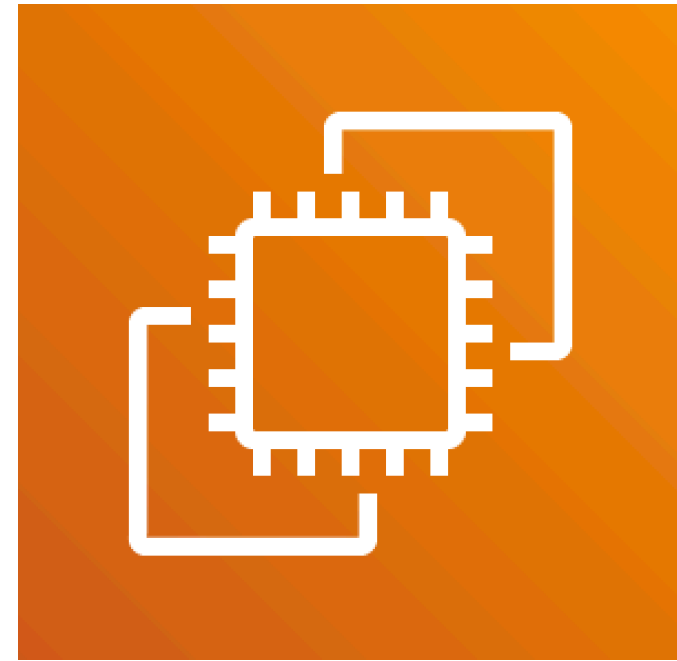
- 1) リリース障害切り戻し時間の短縮化 **(1/20)**
- 2) リリースに係るオペレーションコスト **(1/10)**
- 3) 急激なトラフィック増加に耐えられる運用の柔軟さ

コンテナで開発したアプリの本番運用に適する運用基盤



実際の効果：環境差分の障害発生件数の減少、アプリ諸々運用簡素化

豊富な管理APIによるシステムコストの最適化を実現

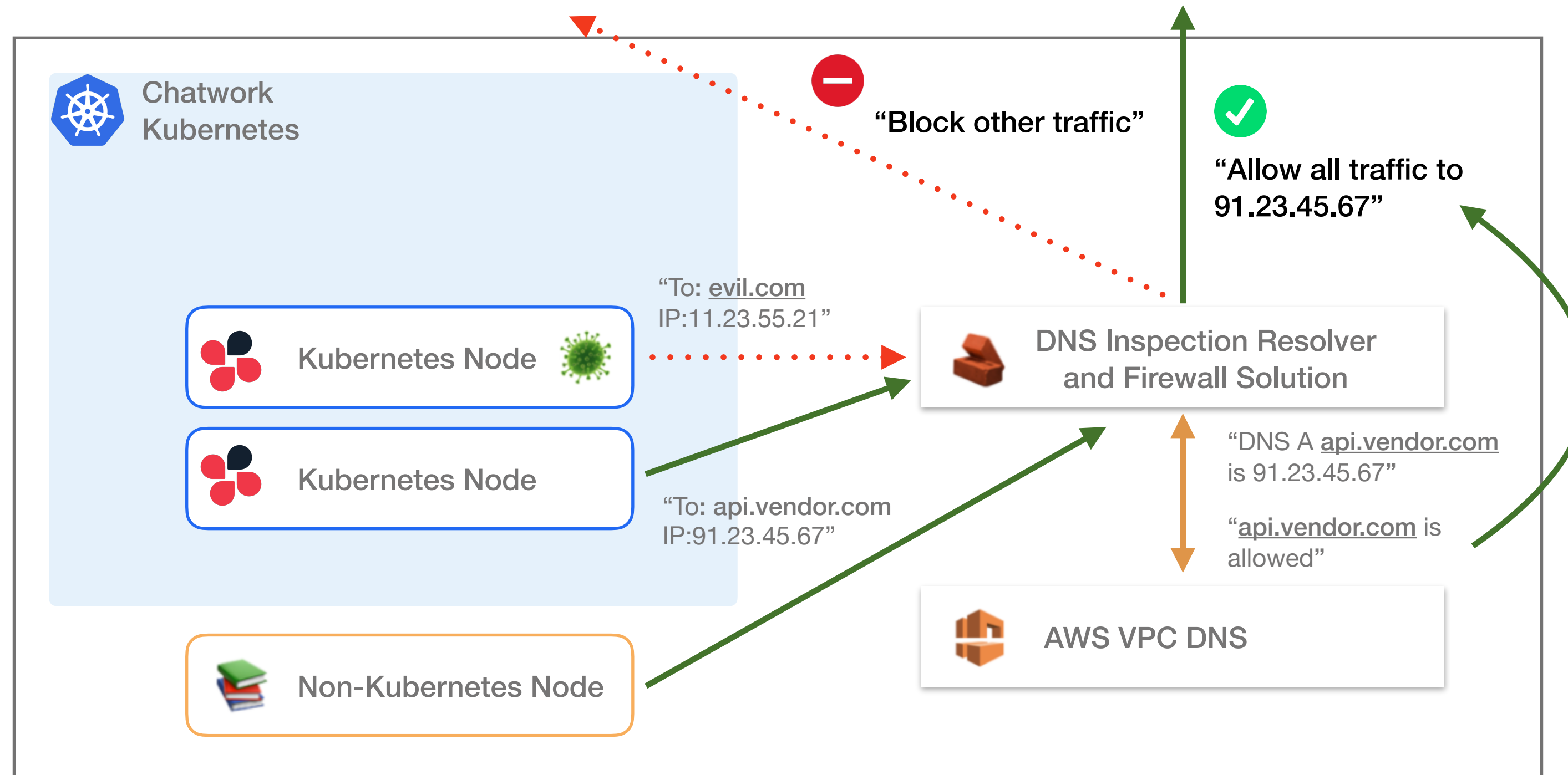


実際の効果：EC2のspotインスタンス化によるコスト削減

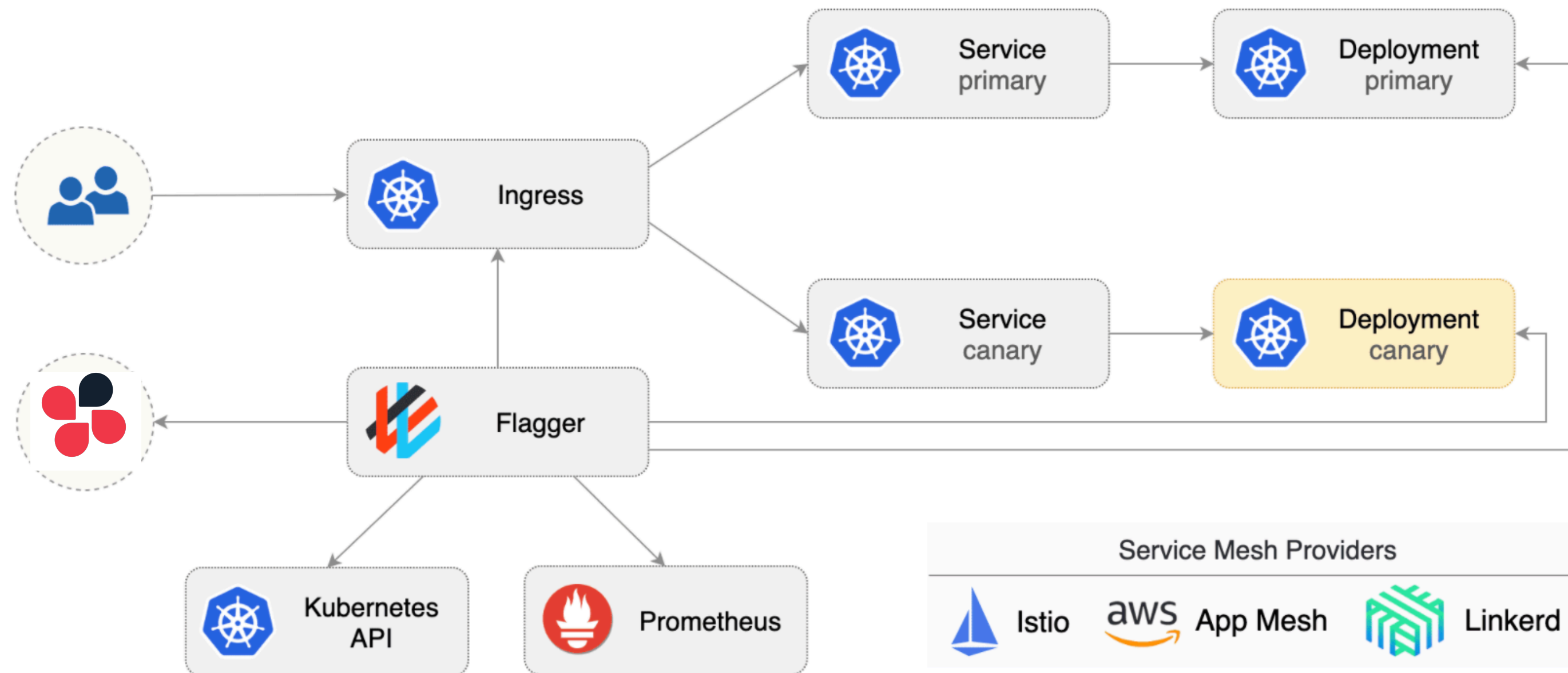
05

今後の展望

許可していない外部サービスへの通信をブロック

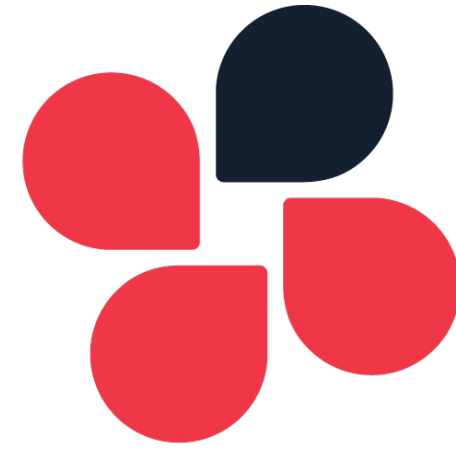


3ヶ月に一度のKubernetes Version Up 自動更新



Source: <https://github.com/weaveworks/flagger>引用

働くをもっと楽しく、創造的に



Chatwork

SRE部メンバー募集中！

<https://corp.chatwork.com/ja/recruit/>

Thank you!