



SREのはじめかた

NTTドコモ サービスデザイン部"RAFTEL"が
実践するサービスレベルの計測と可視化

Tsuyoshi Shimizu

Senior Solutions Consultant

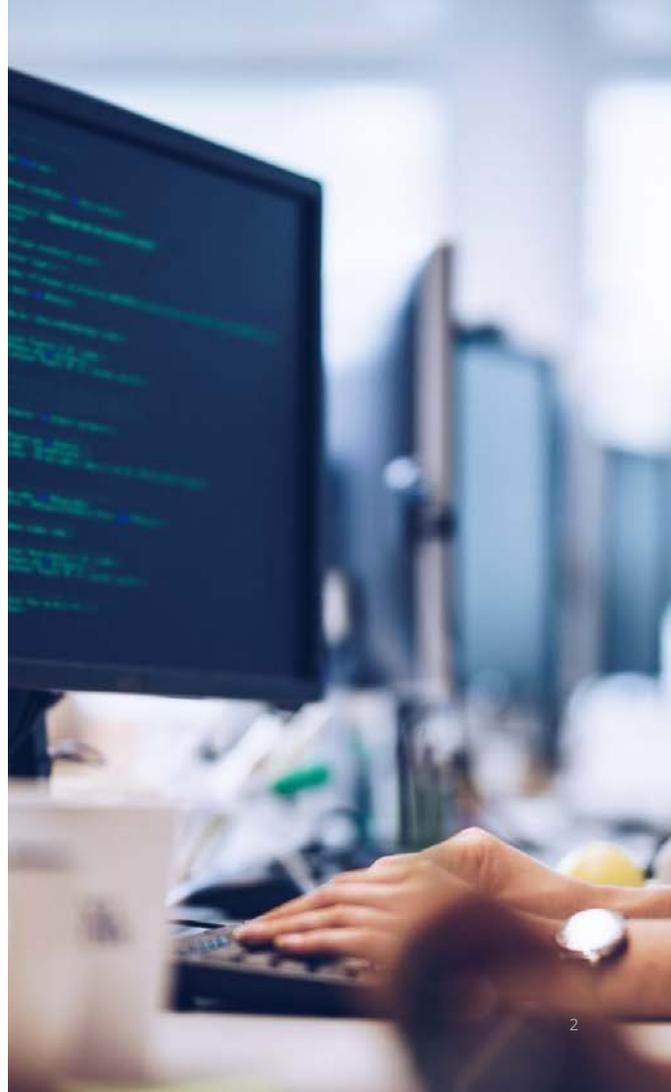
対象者とゴール

対象者

- SREについてこれから学びたいエンジニア
- サービスレベルの計測を始めたいと思っているエンジニア

ゴール

- SREの基本の理解
- **サービスレベル計測の必要性理解**



本セッションの流れ

1. SRE とDevOps
2. サービスレベルの基本
3. NTTドコモ様"RAFTEL"事例紹介
4. まとめ
5. QAセッション



- Package Vender - E-Commerce **Infra**
- Package Vender - E-Commerce **SaaS SRE**
- Public Cloud - **SaaS** Solutions Architect
- New Relic - Solutions Consultant



New Relic®

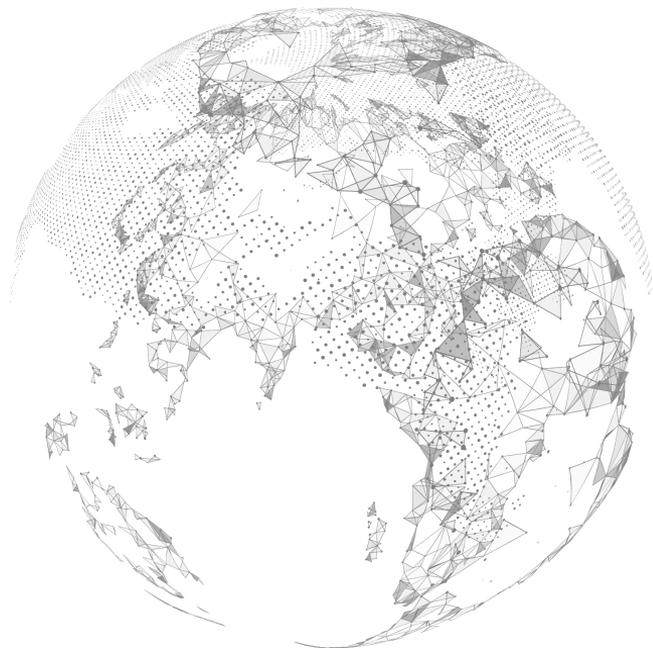
清水 毅
Senior Solutions Consultant
New Relic



**More Perfect
Software.**



デジタル変革は全企業の最優先課題



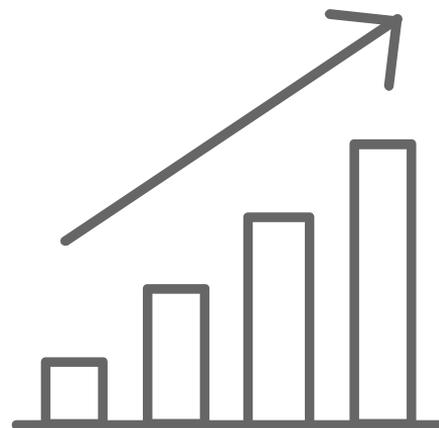
2022年までに
デジタルサービスが
GDPに占める割合

60%以上

IDC Future Scope: Multiplied Innovation Takes Off, Powered by AI, Distributed Public Cloud, Microservices, Developer Population Explosion, Greater Specialization and Verticalization, and Scaling Trust

ビジネスアジリティの必要性

- デジタルサービスを用いてビジネスを推進する必要性の高まり
- 新しいデジタルビジネスへの移行の要求
- ビジネスの迅速な市場への投入



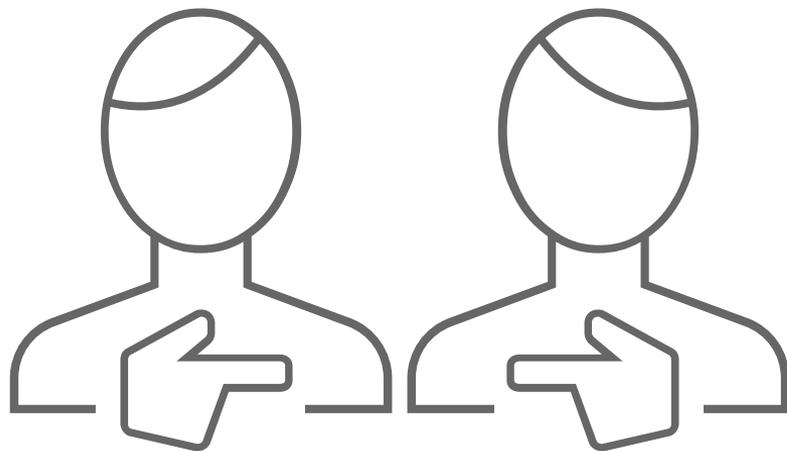
開発と運用の対立

新しい機能をできるだけ多くリリースしたい開発 (Dev)

と

システムを安定化させたい運用
(開発運用やインフラ運用)

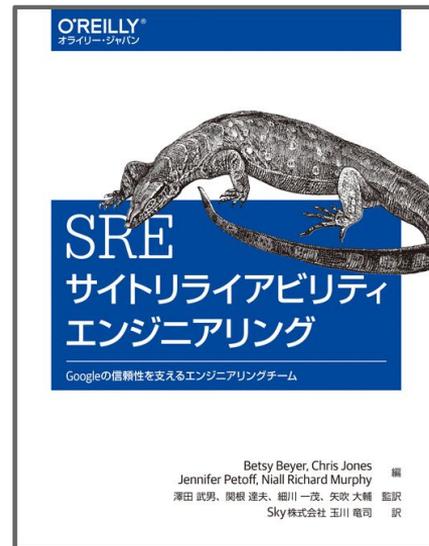
ミッションの違いにより
たびたび対立構造に



DevOpsとは

「DevOps」という言葉は、2008年の終わり頃から業界で使われ始め、本書の執筆時点（2016年初頭）の時点でも、その意味するところは未だに固まっていません。

その中核となるのは、システムの設計と開発の各フェーズにおいて IT に役割を持たせること、人手ではなく自動化に頼ること、運用タスクにエンジニアリングのプラクティスとツールを適用することなど ”



本セッションでの定義:

DevOpsとは文化・ツール・プラクティスの3つで構成され、
開発(Dev)と運用(Ops)のフィードバックサイクルの高速化を行うこと

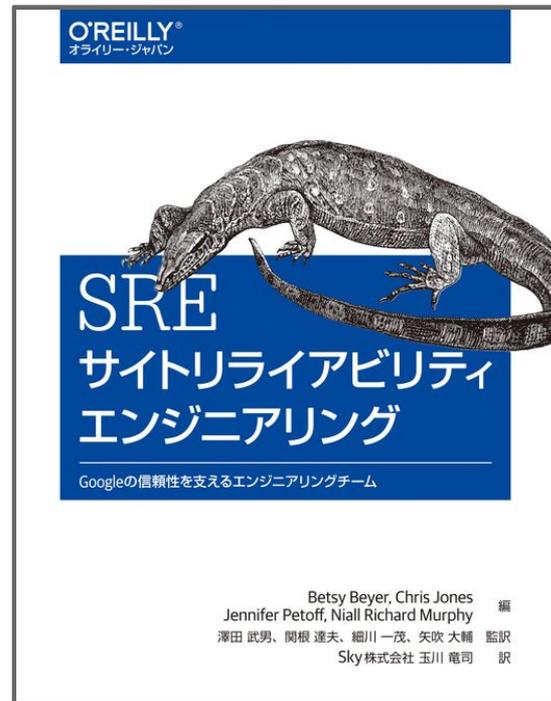
SRE (Site Reliability Engineering) とは

SRE とは “ソフトウェアエンジニアが 運用を設計したらどうなるのか？” ということ

- Benjamin Treynor Sloss, Vice President, Engineering, Google

- “SREはシステムの信頼性に重きをおく”
- “SREとはDevOpsに独自拡張を加えた1プラクティスと考えることもできる”
- “一般的に、SREチームは可用性、レイテンシー、パフォーマンス、効率、変更管理、監視、緊急対応、サービスのキャパシティプランニングに責任をもつ”

※“運用にかけていいリソースは活動の50%まで、50%以上は信頼性を向上するためのコード生成を行うこと”

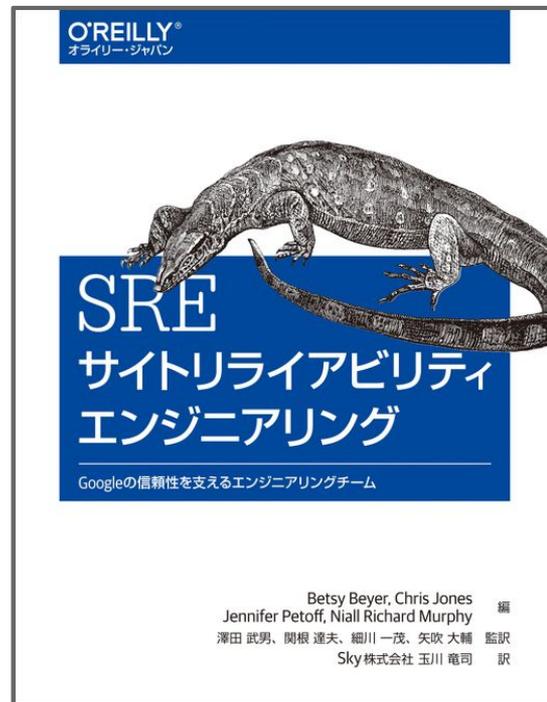


サービスの信頼性とは

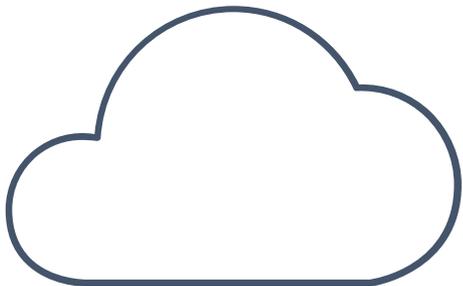
“ここでは、信頼性とは「[システムが]求められる機能を、定められた条件の下で、定められた期間にわたり、障害を起こすことなく実行する確率」です。これは[Oco12]の定義に従っています。”

Oco12:P. O'Connor and A. Kleyner, 「Practical Reliability Engineering」, 5th edition: Wiley, 2012.

サービスの信頼性とは
“障害を起こすことなく
実行する確率”



Service Levelとは



SaaS

SLA
Service
Level
Agreement

Service Level のタイプ

SLI

Service Level Indicator

計測値

SLO

Service Level Objective

目標値

SLA

Service Level Agreement

契約値

SREがフォーカスするのはSLIとSLO

SLIのタイプ

SLIには種類があり、
代表的なものは可用性や
レイテンシー。

システムによって使い分
ける。

| サービスの種類 | SLIのタイプ |
|-----------|---------|
| リクエストドリブン | 可用性 |
| リクエストドリブン | レイテンシー |
| リクエストドリブン | 品質 |
| パイプライン | 新鮮さ |
| パイプライン | 正確性 |
| パイプライン | カバレッジ |
| ストレージ | 耐久性 |



顧客視点のSLO設定

- "SLOは、サービス顧客の目標レベルで信頼性を設定します。"
- "このしきい値を下回ると、ユーザーは不平を言い始めるか、サービスの利用を停止する可能性があります。"
- "最終的に、ユーザーの幸せが重要です。(中略)

お客様の満足を維持するためにサービスの信頼性を維持しています。"



SLOとエラーバジェット

それぞれの目的には個別のエラーバジェットがあり、
100%からそのSLOを引いたものとして定義されます。



100% - SLO = エラーバジェット

SLI - SLO = "残"エラーバジェット

New Relic SLIダッシュボード例

Dashboards

Service Level Dashboard - tshimizu ★ ⓘ

Demotron V2



🕒 Since 7 days ago ▾

☰ ▾ Search for any attribute or value.

SLI 可用性 (SLO 99%)

Since 1 week ago

100 %

Verify all APIs are working

100 %

Create alert violations data

100 %

Load homepage and assets

100 %

Verify homepage is working

69.66 %

Verify checkout flow is working

SLI レイテンシー (SLO 3秒以内99%)

Since 1 week ago

100 %

Verify homepage is working

99.97 %

Verify all APIs are working

99.17 %

Create alert violations data

97.02 %

Load homepage and assets

0 %

Verify checkout flow is working

可用性・エラーバジェット

Since 1 week ago

Verify all APIs are working 1 %

Create alert violations data 1 %

Load homepage and assets 1 %

Verify homepage is working 1 %

Verify checkout flow is working -29.34 %

レイテンシー・エラーバジェット

Since 1 week ago

Verify homepage is working 1 %

Verify all APIs are working 0.97 %

Create alert violations data 0.17 %

Load homepage and assets -1.98 %

Verify checkout flow is working -99 %

事例：
株式会社NTTドコモ様
サービスデザイン部

自己紹介

宮川 倫

所属:

ネットワーク本部 サービスデザイン部
第一クラウド推進担当



経歴

2018年～ 一般ユーザ向けサービス (dカーシェア・dエンジョイパス) のシステム開発

2020年～ 複数サービスが利用するサービス共通 API 基盤 (RAFTEL) の運用

SRE やってみようとしてる

株式会社 NTTドコモ

営業収益 4.7 兆円 (2020 年度)

モバイル通信事業に加え
様々な事業を展開



サービス デザイン部

Mission

テクノロジーの民主化

スキルやノウハウの脱属人化

ユーザファーストでサービス創出

ペルソナの潜在ニーズをくみ取ったサービスの創出

サービス提供基盤の再構築

CloudNative 化でサービスを早く、安く、柔軟に提供

NTT ドコモが提供する
多様なサービスに携わる



ビジネス部と連携して
要件定義、実装はSler



RAFTEL チームのミッション

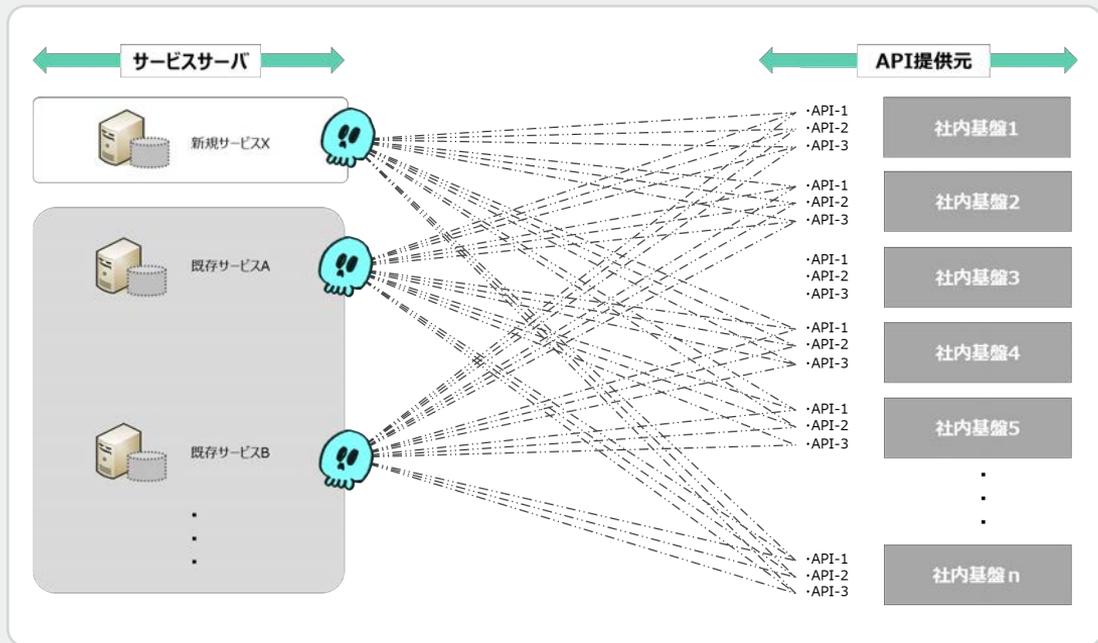
- 早く、安い開発！
 - 市場投入までの時間を最短に
- 自動化やツール使用の推進！
 - トイル削減。人は本来やるべき価値創出を全力でやる



RAFTEL 誕生以前

遅い&面倒 🙄

- 基盤ごとにバラバラかつ、複雑な API をそれぞれ仕様の理解から実施
- 基盤システム利用開始までの時間がかかることで開発スケジュールへの影響大
- 基盤システムの API 更新に追従した開発が全てのサービスサーバに必要



RAFTEL 爆誕



RAFTEL (サービス共通API基盤)

～ RAPid development system For TEam Launch ～

ドコモのサービス開発を「はやく」「やすく」するためのAPIプロダクト

サービスのユースケースを考慮してドコモの多くの基盤(認証基盤や決済基盤など)のAPIをラッピング

開発の高速化

サービス開発は2~4週に1回のペースでリリース、RAFTELも同等のスピード開発を実現する

開発したアプリケーションの有効活用(コンテナ化による流用)

高信頼性

dアカウント認証機能は24時間365日無停止が前提

決済機能も年に数回の計画停止以外は24時間稼働

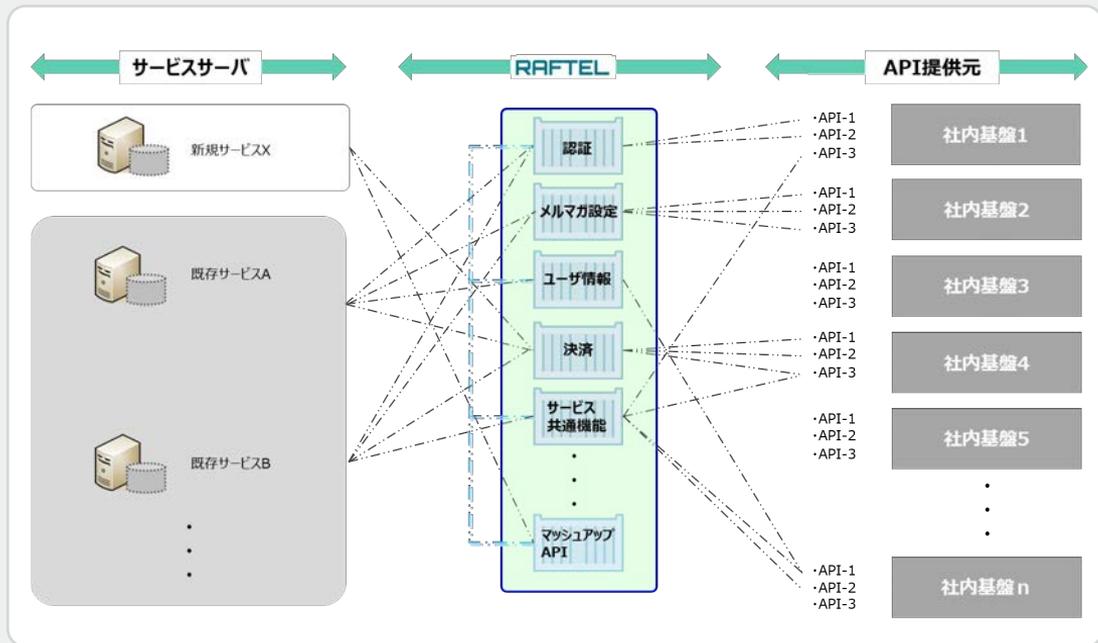
運用稼働削減

インフラ保守作業やOS・ソフトウェアバージョンアップ作業極力削減

RAFTEL 誕生以後

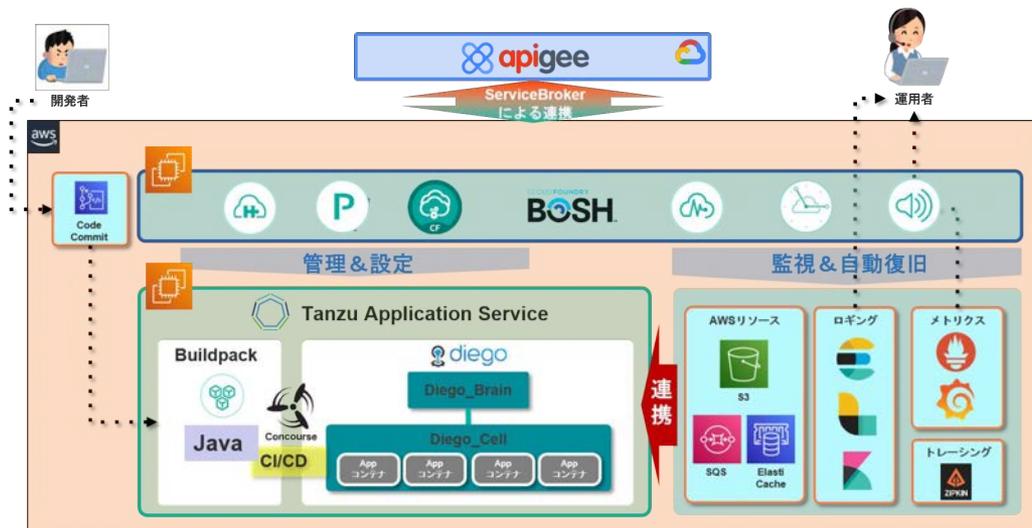
簡単にすぐ使える 😊

- 接続基盤の単一化が開発効率化を実現
- 社内基盤の API 更新は RAFTEL が吸収することでサービスサーバは開発削減を実現
- ユースケースに応じて複数 API をマッシュアップして提供するので、サービスサーバの開発削減に貢献



初期 RAFTEL の性能と効果

API利用の利便性向上、かつ安定稼働を実現し接続サービスが順調に拡大



40以上

RAFTELを利用するサービスの総数

4億以上

RAFTELが1日に捌くトランザクション

2営業日

API利用開始まで必要な日数

70%削減

APIの仕様学習コストと実装工数

15分

アプリケーション完成からリリースまで

再利用可

コンテナベースでマッシュアップ

無停止

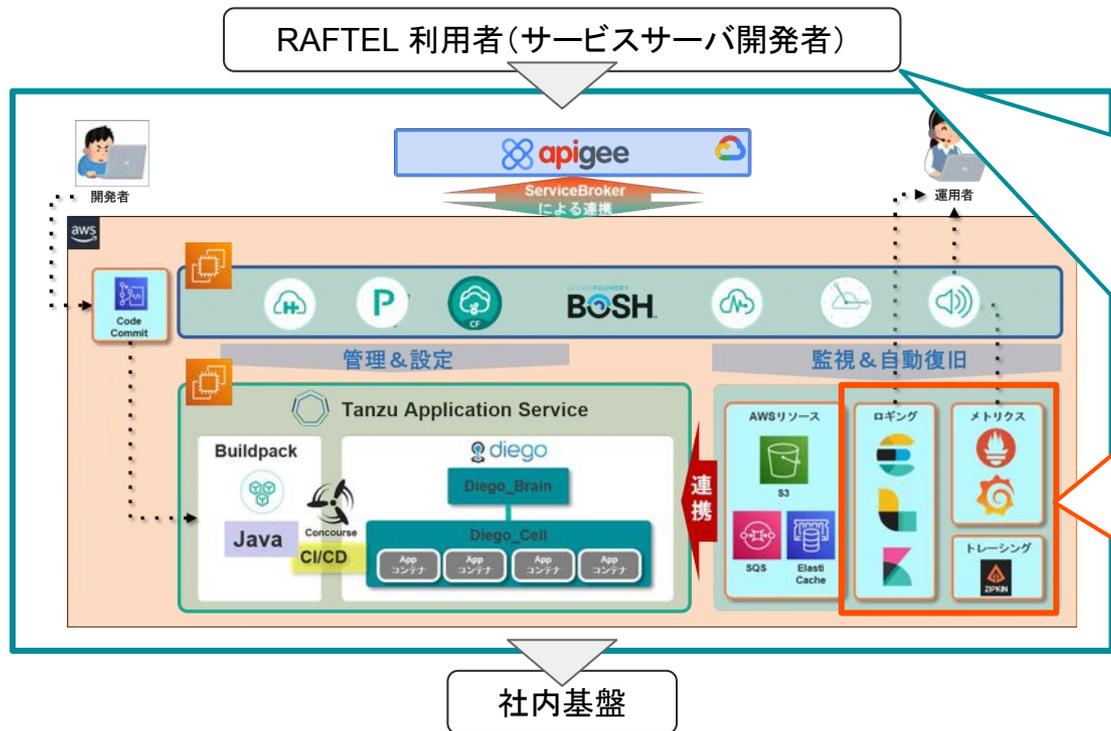
ローンチから現在まで一度も停止なし

運用なし

IaaS/PaaSレイヤは自動復旧なのでノータッチ

出てきた課題

当初想定通りの開発・運用はできてきた👍しかし、課題が出てきた...



課題2: 問い合わせ多すぎ問題

利用が増えるにあたって RAFTEL チームへの問い合わせが増える
👉 開発者向けポータルサイトを用意して利用者自身で解決できるようにしているが問い合わせは多い

RAFTEL の QA ではなく社内基盤 (他部署管理) 原因の問い合わせも来るように😞
👉 現状は RAFTEL チームにて回答している状況

運用も大事だが、開発にもっと稼働を使っていきたい😞

NewRelic との出会い

- まずは “ツール多すぎ問題” を解決すべく SaaS 導入を検討
- そこで New Relic と出会う
- 導入検証をしていく中で New Relic 清水さんからの様々なインプットにより RAFTEL チームの考え方が変わっていく ...
 - Observability
 - ユーザ体験の向上
 - SLO・エラーバジェットの考え方
- New Relic の RAFTEL への導入
 - メトリクス・ロギング・トレースが New Relic 1つで完結
 - 👉 課題1: ツール多すぎ問題 **解決** 🎉
 - SLO 設定・モニタリング実施
 - 外形監視 (New Relic Synthetics) 実施



SLO / SLI 定義と計測の実践

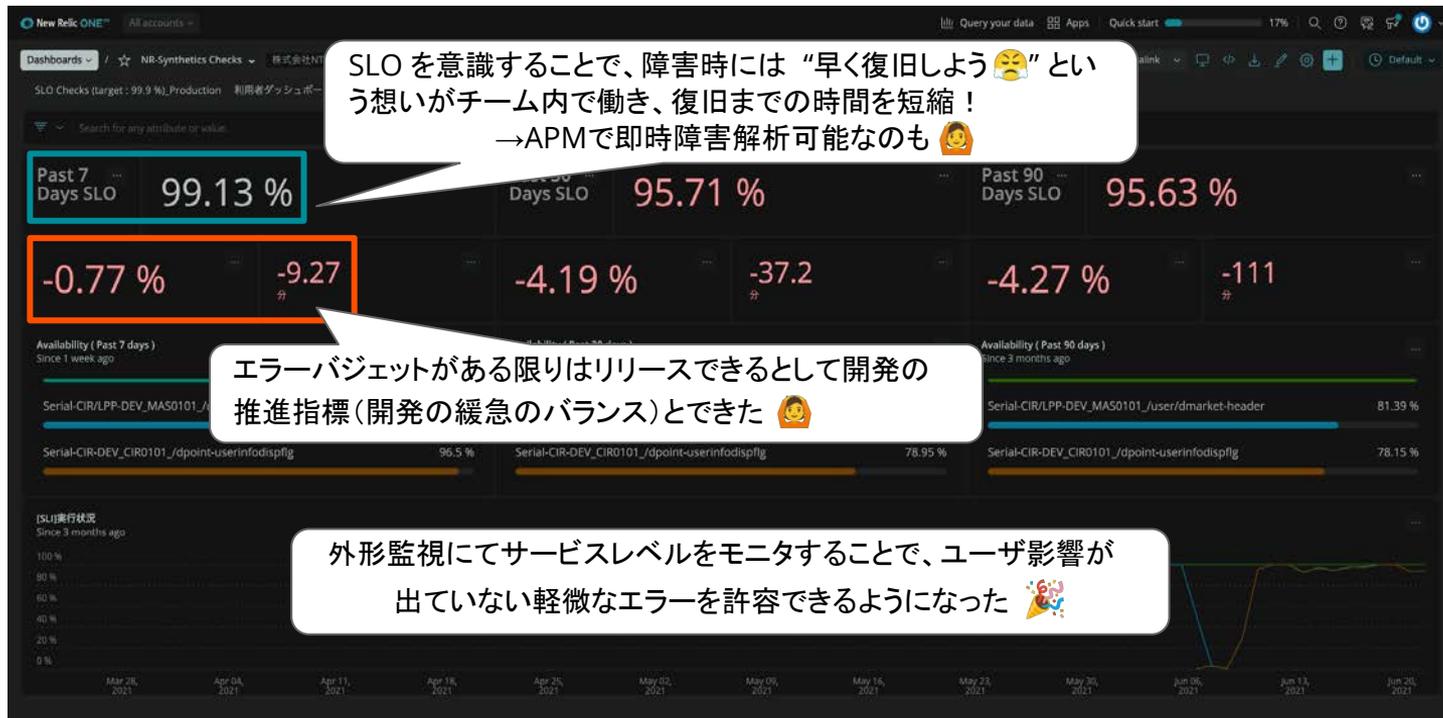
実際に SLO / SLI を定義し、計測する取組みをやってみた



SLOダッシュボード@商用環境

SLO / SLI 定義と計測の実践

実際に SLO / SLI を定義し、計測する取組みをやってみた



SLOダッシュボード@検証環境

利用者向け API ダッシュボードの公開

- 利用者が API 状態を確認できるダッシュボードを作成し公開
- 利用者が自分で状況確認ができることにより RAFTEL チームへの問い合わせ減



課題2: 問い合わせ多すぎ問題 **解決** 🎉

運用稼働を削減することで、
より開発に注力できるように 🍷

| TIMESTAMP | RESULT | ERROR DETAIL |
|------------------------|--------|--------------|
| June 23, 2021 12:11:51 | FAILED | 社内基盤メンテナンス |
| June 23, 2021 12:11:51 | FAILED | 社内基盤メンテナンス |

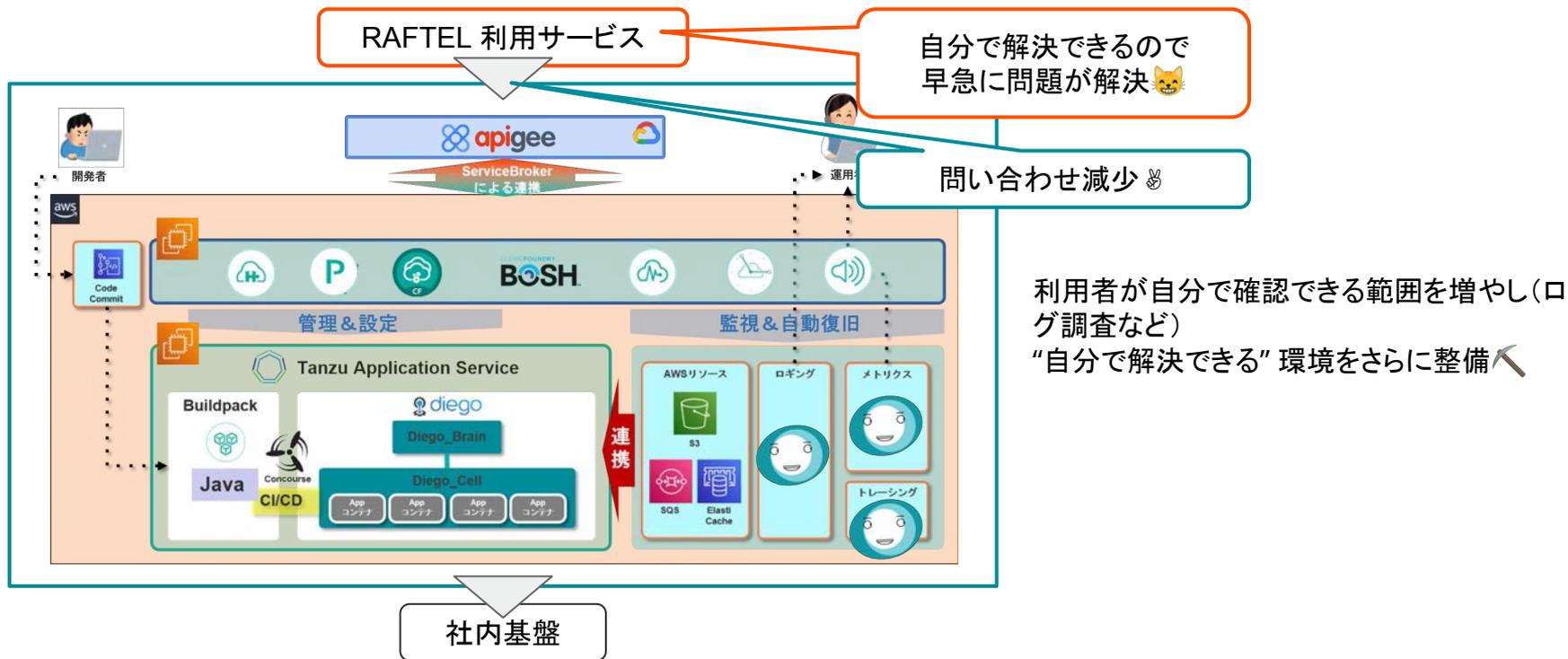
エラーのAPIとエラー原因がわかる

RAFTEL APIのみのエラーでなく社内基盤起因のエラーであることもわかる 🐱

利用者向けダッシュボード@**検証環境**

今後の展望

今後、利用者／トランザクション数のさらなる増加が見込まれる...

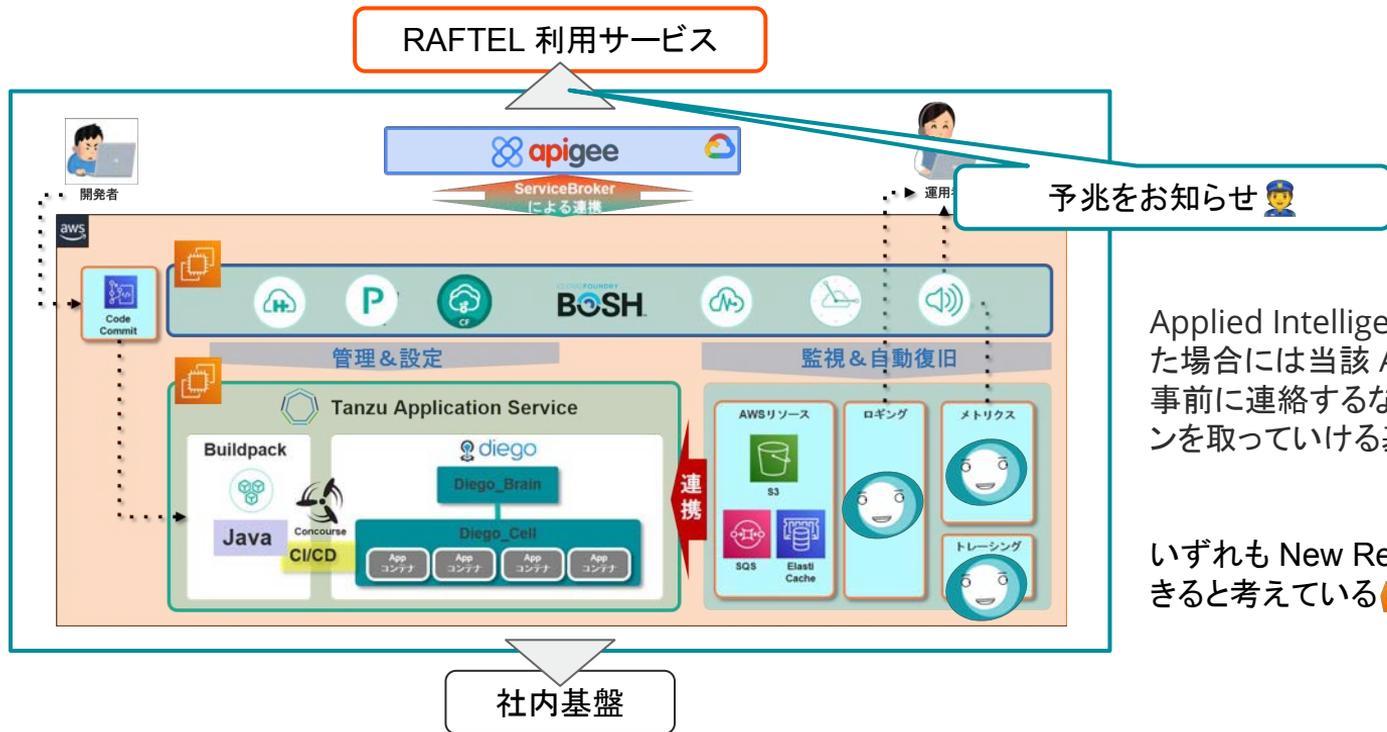


利用者が自分で確認できる範囲を増やし(ログ調査など)

“自分で解決できる” 環境をさらに整備 🗡️

今後の展望

今後、利用者／トランザクション数のさらなる増加が見込まれる...



Applied Intelligence も利用し、予兆を検知した場合には当該 API を使用するサービスへ事前に連絡するなど、プロアクティブなアクションを取っていただける基盤としたい

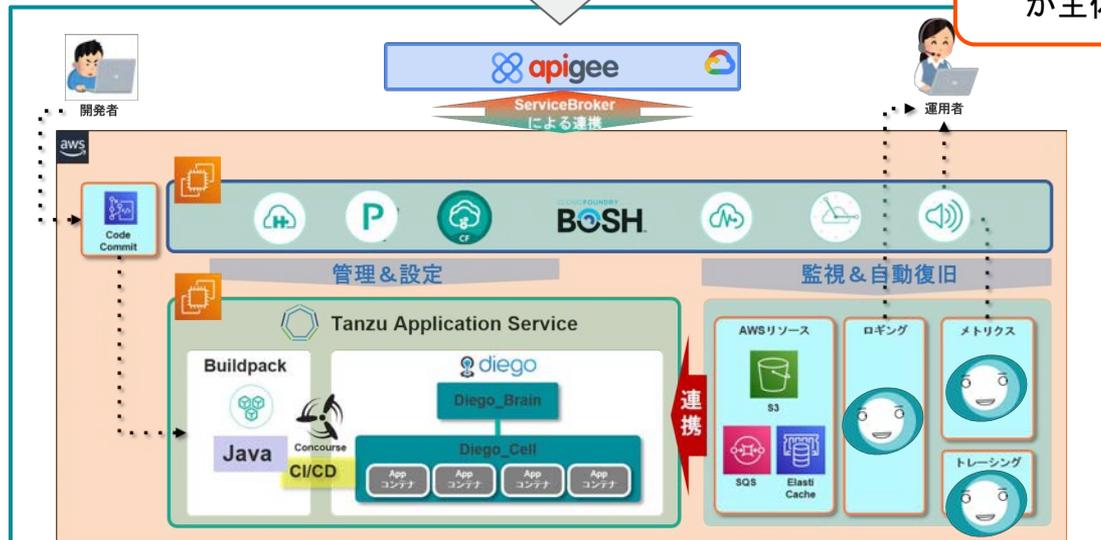
いずれも New Relic を活用することで実現できると考えている 🧑

今後の展望

今後、利用者／トランザクション数のさらなる増加が見込まれる...

RAFTEL 利用サービス

状況が即時確認できる、サービス側が主体的に問題解決できる 🙌



RAFTel API を利用するサービス側、RAFTel が連携している社内基盤側にも New Relic を導入して横串で運用をしていくチャレンジをしたい！

社内基盤

社内基盤の状況を即時確認できる 🙌

最後に

このような取り組みの結果、ドコモ全体(サービス / RAFTEL / 社内基盤)として開発・運用のスピードを上げることができると思っている

- サービス自体の改善(解決速度向上・開発スピード向上)に繋がることで、その先のエンドユーザに価値あるサービスを提供できる🎉👏🙌
- RAFTELが掲げていたミッション
ドコモのサービスを「はやく」「やすく」するを達成できる🎉👏🙌

まとめ

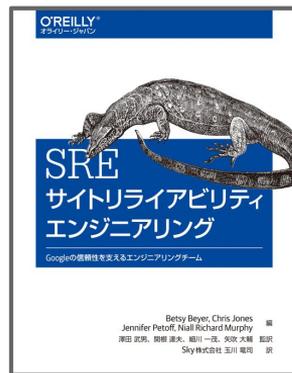
まとめ

- SREをはじめるステップ

- SREの理解
- サービスレベルを計測
- 顧客起点のSLO設定

- 株式会社NTTドコモ様”RAFTEL”SRE

- SREとしてサービスレベル計測し問
い合わせの大幅削減！



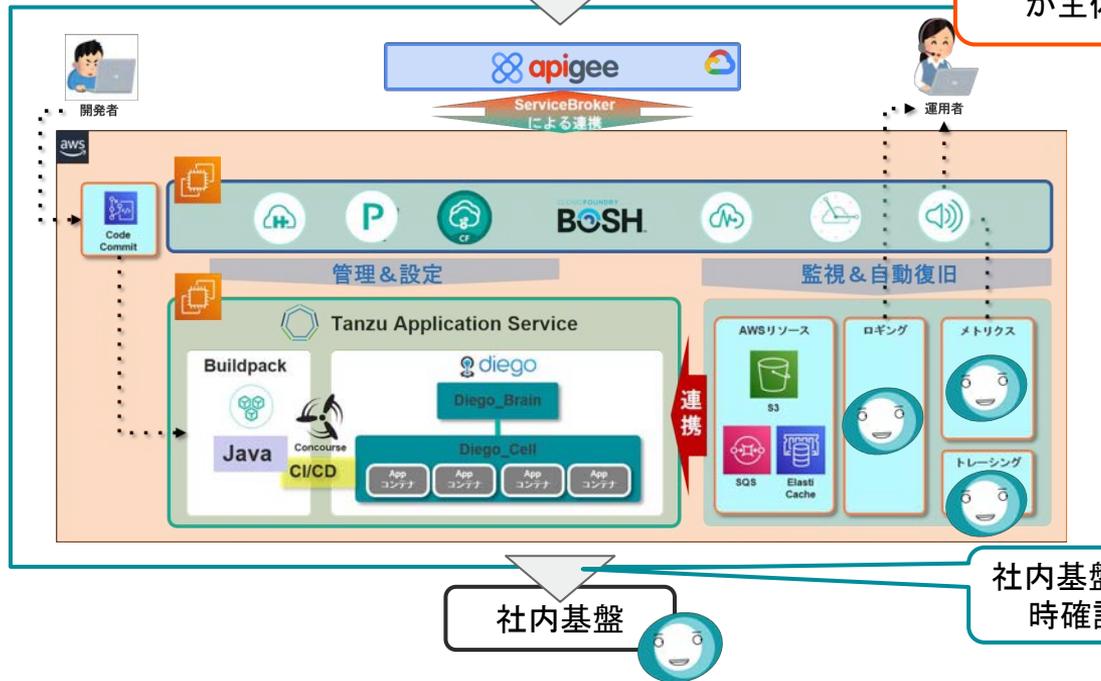
QA

今後の展望 (ドコモ様資料再掲)

今後、利用者／トランザクション数のさらなる増加が見込まれる...

RAFTEL 利用サービス

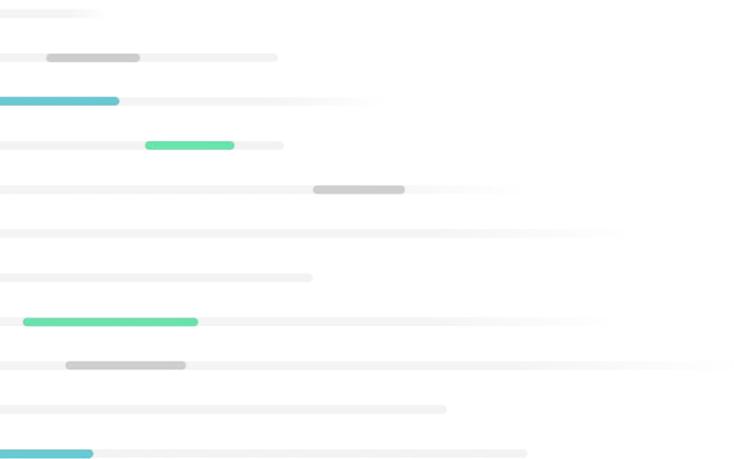
状況が即時確認できる、サービス側が主体的に問題解決できる 🙌



RAFTEL API を利用するサービス側、RAFTEL が連携している社内基盤側にも New Relic を導入して横串で運用をしていくチャレンジをしたい！

社内基盤

社内基盤の状況を即時確認できる 🙌



Thank You

tshimizu@newrelic.com

@photographed

