New Relic

[New Relic Solutions]

インフラ監視の
アンチパターンとデザインパターン

Tsuyoshi SHIMIZU

Senior Solutions Consultant

New Relic KK

GoToWebinarの活用

本Webinarは**質問**が可能です。

音声**不具合**などについても
こちらからご連絡ください。

適宜**回答**いたします。

最後の**Q&A**の時間にて
可能な範囲で回答いたします。

Safe Harbor

This presentation and the information herein (including any information that may be incorporated by reference) is provided for informational purposes only and should not be construed as an offer, commitment, promise or obligation on behalf of New Relic, Inc. ("New Relic") to sell securities or deliver any product, material, code, functionality, or other feature. Any information provided hereby is proprietary to New Relic and may not be replicated or disclosed without New Relic's express written permission.

Such information may contain forward-looking statements within the meaning of federal securities laws. Any statement that is not a historical fact or refers to expectations, projections, future plans, objectives, estimates, goals, or other characterizations of future events is a forward-looking statement. These forward-looking statements can often be identified as such because the context of the statement will include words such as "believes," "anticipates," "expects" or words of similar import.

Actual results may differ materially from those expressed in these forward-looking statements, which speak only as of the date hereof, and are subject to change at any time without notice. Existing and prospective investors, customers and other third parties transacting business with New Relic are cautioned not to place undue reliance on this forward-looking information. The achievement or success of the matters covered by such forward-looking statements are based on New Relic's current assumptions, expectations, and beliefs and are subject to substantial risks, uncertainties, assumptions, and changes in circumstances that may cause the actual results, performance, or achievements to differ materially from those expressed or implied in any forward-looking statement. Further information on factors that could affect such forward-looking statements is included in the filings New Relic makes with the SEC from time to time. Copies of these documents may be obtained by visiting New Relic's Investor Relations website at ir.newrelic.com or the SEC's website at www.sec.gov.

New Relic assumes no obligation and does not intend to update these forward-looking statements, except as required by law. New Relic makes no warranties, expressed or implied, in this presentation or otherwise, with respect to the information provided.

Q1/3: みなさまのお役割

Q2/3: 現在取り組まれているインフラ監視では
どのようなツールを
利用していますか？

Q3/3: 現在取り組まれているインフラ監視では『どこまで』実施していますか？

- 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.

対象者とゴール

対象者

- モダンWEBシステムのための

  モダン監視を学びたいエンジニア

- 次世代WEBインフラエンジニアになりたい方

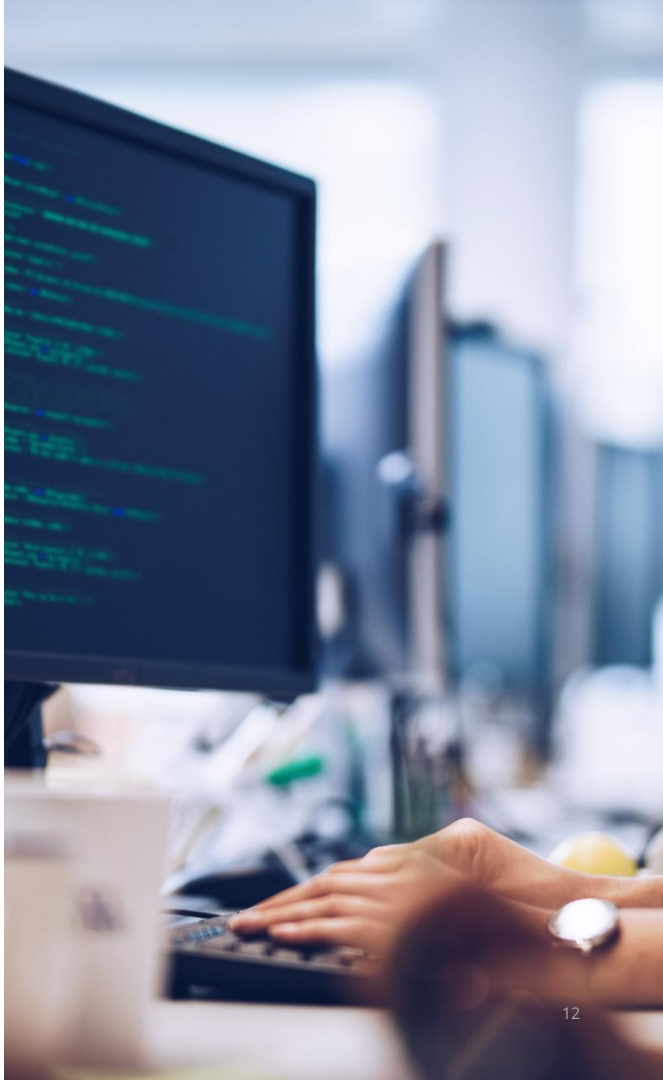
ゴール

- インフラエンジニアにおける監視を

  歴史からおさらいしモダン監視の理解

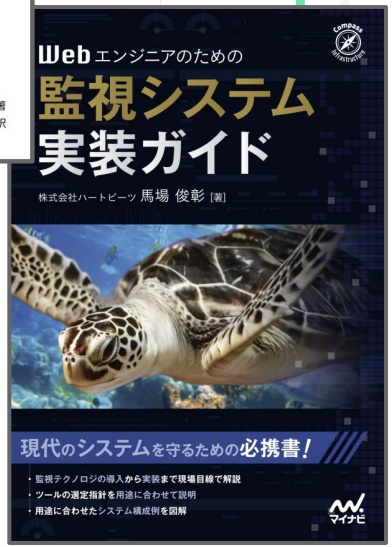
- モダンWEBシステムをモダンに

  監視する方法を理解し一歩を踏み出せる



Agenda

1. インフラ監視の基本
2. インフラ監視のアンチパターンと
モダンなデザインパターン
3. インフラ監視設計への
オブザーバビリティの組み込み方
4. デモ
5. まとめ



インフラ監視の基本

システムはビジネスそのものになりつつある

1980年～
メインフレーム・オープンシステム

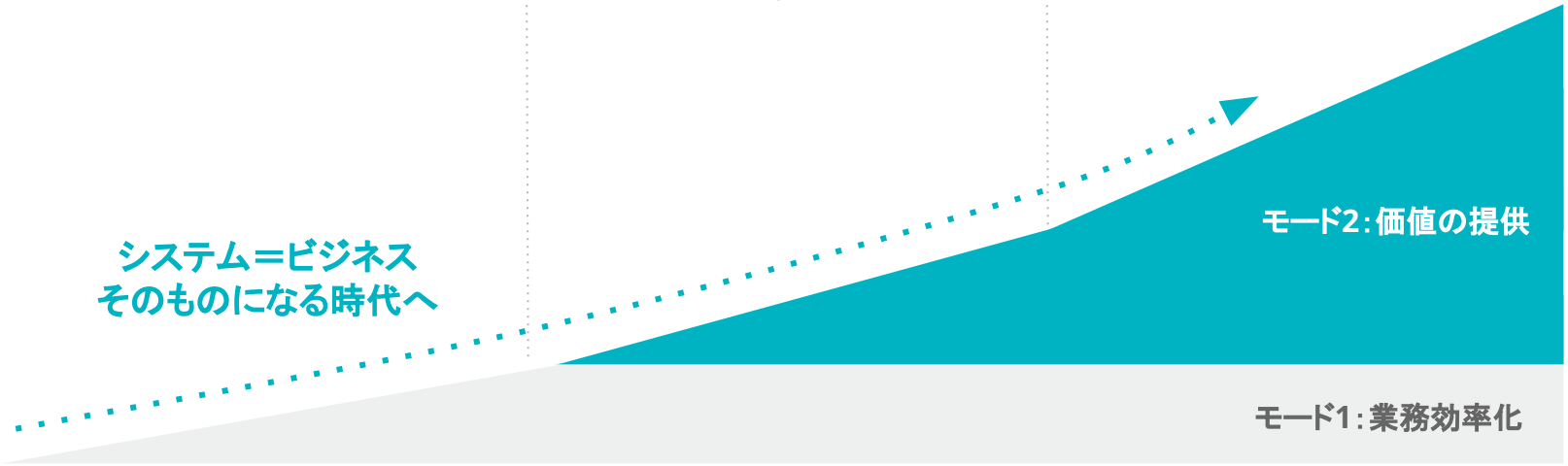
業務効率化
メールやドキュメントなど

2000年～
インターネット・WEB・ガラケー

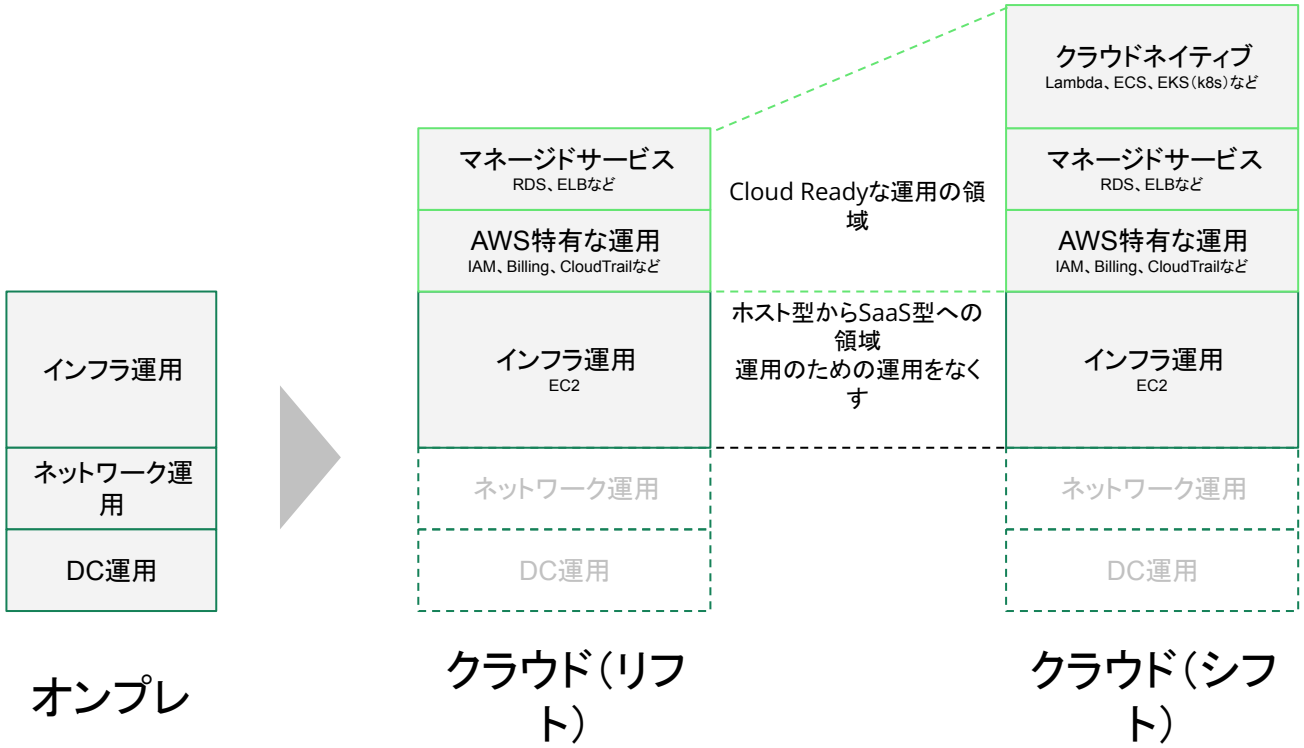
自動化/情報分析
インターネット検索やeコマース

2010年～
クラウド・スマートフォン・ AI・IoT

事業創造
システム＝ビジネス

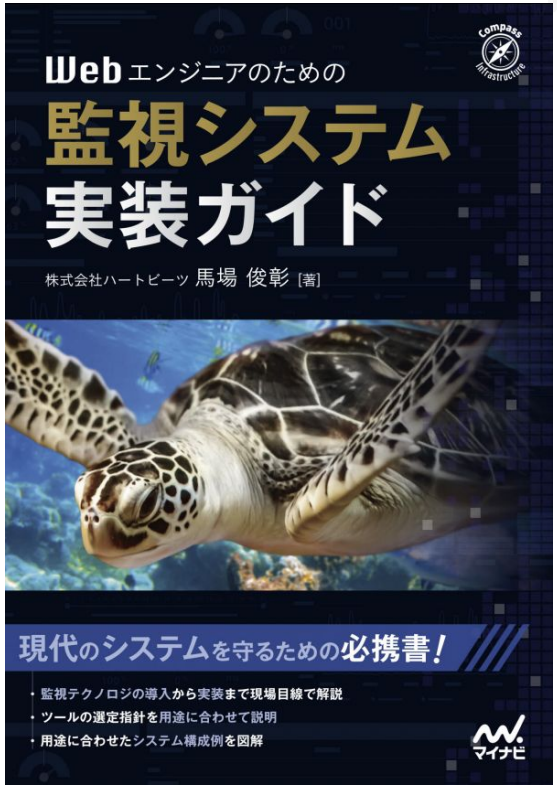


インフラのモダン化と監視の変遷

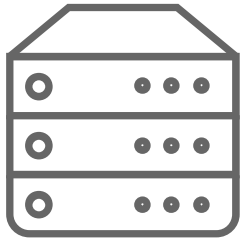


インフラ監視技術の変遷

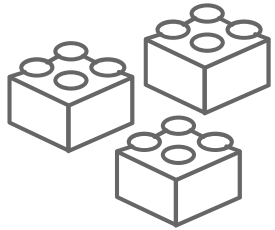
- ～2000年代: **OSSを活用したチェック・メトリクス監視**
  - 5分に一回チェックとメトリクス収集
  - NagiosやZabbix、MRTG、Cacti etc
- 2000年代後半～: **システムの大規模化とDevOps**
  - 1~3分に1回、データ量増大
  - Chef、Puppet、Ansible、Serverspec etc
- 2010年代～: **クラウド・ペットから家畜へ・Disposable**
  - 1分1回 各種メトリクスを収集可視化
  - Sensu、Amazon CloudWatch、Monitoring SaaSの登場
- 2010年代中盤～: **Microservice、Observability、SRE**
  - 1分に1回以上 収集
  - Prometheus、Monitoring SaaSの定着



インフラ監視って？

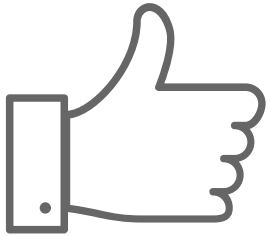


OSリソース監視

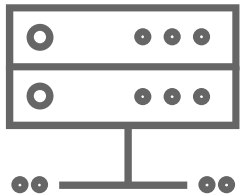


プロセス監視

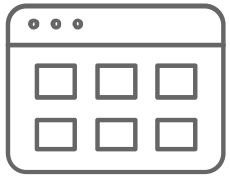
ログ監視



死活監視



機器監視



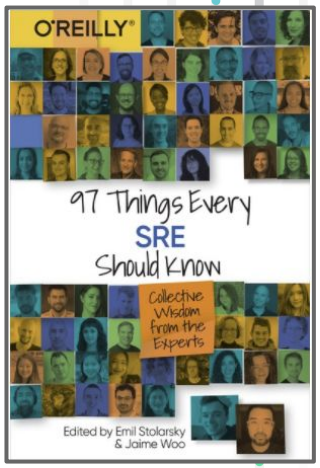
URL監視

Optimize for MTTBTB
”Mean Time to Back to Bed”

Spike Lindsey

Shopify

- *“97 Things Every SRE Should Know” 2020 O’reilly*



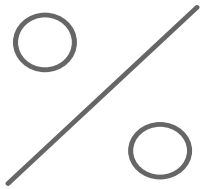
- オンコールの課題

「あなたが目覚めたばかりのときに、その話を理解できるでしょうか」
「これで早くベッドに戻れるか（そしてずっとベッドにいられるか）？」

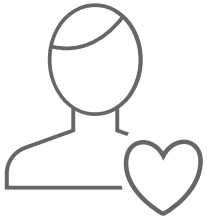
インフラ監視に関する不安の例（実体験）

1. アラートが多すぎるし、受け取って何をしていいかわからない
2. インフラ監視ツールの運用のための運用が大変
3. 開発現場とのコミュニケーションにギャップある
4. クラウド監視がこれでいいのか悩ましい
5. コンテナやサーバーレスの何を監視していいかわからない
6. 今のままオンプレミスの延長線上でインフラ監視をやっていていいのか？
7. インフラエンジニアとしてのキャリアはどこに向かうのか？
8. SREって最近聞くけどどう違うのか

モダンインフラ監視って？



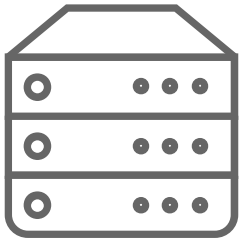
サービスレベル監視



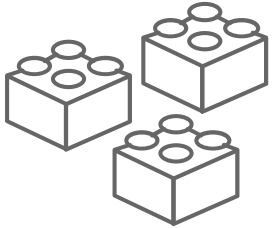
UI/UX監視

アプリケーション監視

クラウド・コンテナ監視

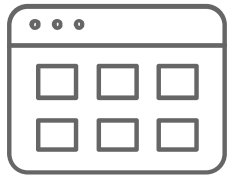


OSリソース監視



プロセス監視

ログ監視



URL監視

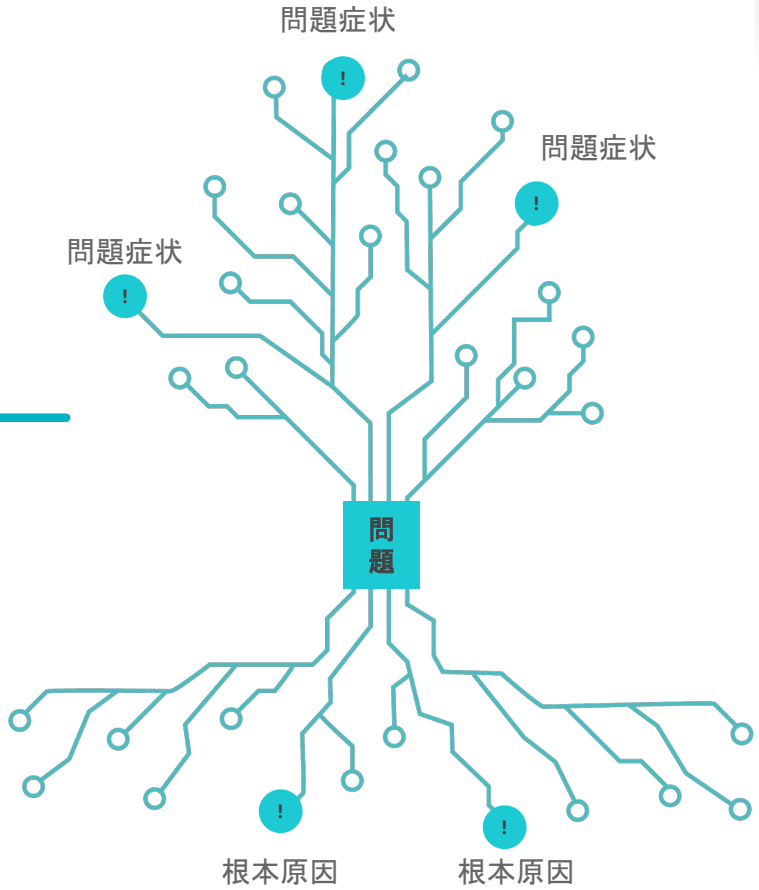
監視
Monitoring

**"監視とは、あるシステムやそのシステムの
コンポーネントの振る舞いや出力 を
観察しチェックし続ける行為である。**
Greg Poirier - Monitorama 2016"

"入門監視" - Mike Julian 著 松浦隼人 訳
オライリー・ジャパン

オブザーバビリティ
Observability

システムのメトリクス・イベント・
ログ・トレースのデータを
リアルタイムに取得し続け、
常にシステム全容の
状態把握と改善ができる状態



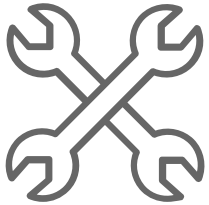
インフラ監視の
アンチパターンとデザインパターン

New Relic.

❌ 監視アンチパターン5

1. ツール依存
2. 役割としての監視
3. チェックボックス監視
4. 監視を支えにする
5. 手動設定

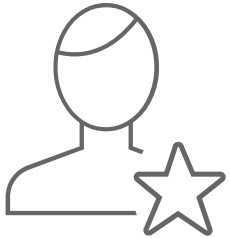
❌ アンチパターン①:ツール依存



1. ツールに依存している
    a. 監視とは**複雑な問題をひとくくり**にしたもの
    b. カーゴ・カルト(**不要なものを儀式的**に含める)な
       ツールを**避ける**
    c. **自分でツール**を作らなければならない時もある
    d. 「一目で分かる」は迷信

❌ アンチパターン②：役割としての監視

1. **役割として監視の専門チームがある**



   a. 監視とは**役割ではなくスキル**

      i. チーム内でスキルを身につける。監視専門チームだけが知ればいいということではない

   b. **監視の責任は全員**（監視を最優先項目の一つに）

   c. **監視してない**システムは**本番環境ではない**

   d. **セルフサービスモニタリング**の提供は有効

❌ アンチパターン③:チェックボックス監視

1. OSメトリクスは取得しているが、**システムダウンがわからない**
2. **誤検知が多すぎ**て無視
3. 各メトリクスを**5分以上の長い間隔**で取得
4. **メトリクスの履歴**を保存していない
5. **動いているかどうか**をみていない
6. **OSメトリクス・アラート発報している**

❌ アンチパターン④:監視を支えにする

1. **監視を追加しつづけている**
    a. **監視は解決ではない**
    b. 監視を増やしても**壊れたシステムは直らない**
    c. 監視通知を受け取ったら**問題の修正が必須**
    d. 監視を追加していくようなら立ち止まって

    回復力向上に時間を

❌ アンチパターン⑤:手動設定

1. **監視を手動設定**している
    a. クラウド環境と従来型環境の監視の**違い**
        i. **ペットから家畜へ**
        ii. クラウド監視は全体を監視し、**自動化が必須**
    b. 監視設定は**100%自動化すべき**
        i. **自動化なし**で適切な監視は現代では**不可能**
        ii. 監視設定の追加は**容易に**しておこう

❌ アンチパターンまとめ

1. **ツールに依存**しても監視の仕組みは良くなりません
2. **監視は全員がやるべき**仕事でチームや部署内での役割ではない
3. 「これは監視してます」と**チェックをいれるだけ**で済むものではない
4. 監視するだけでは、**壊れたものは直せません**。
5. **自動化**が足りてないということは、何か重要なことを

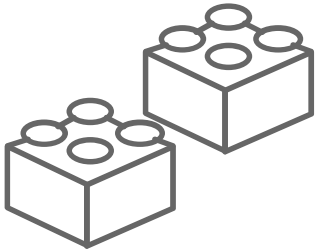
   見落としている

監視デザインパターン4

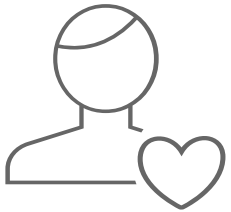
1. 組み合わせ可能な監視
2. ユーザー視点での監視
3. 作るのではなく買う
4. 継続的改善

デザインパターン①：組み合わせ可能な監視

1. **データ収集**
   a. プッシュ型orプル型、メトリクス、ログ
2. **データストレージ**
   a. メトリクス、ログ、コスト
3. **可視化**
   a. 自由にダッシュボードを作れることはとても重要
   b. 素晴らしいダッシュボードの特徴
      i. ある場面での持った疑問を回答するために最もよく理解している人たちによって作られ、『運用』されているときが最高のダッシュボード
4. **分析とレポート**
   a. 可視化をこえて分析レポートを意識すると有益な場合あり
5. **アラート**
   a. 監視とアラートはセットではない
   b. アラートは一つの手段



デザインパターン②:ユーザー視点での監視



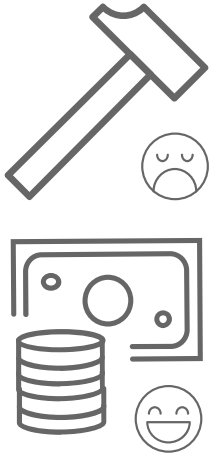
1. さまざまな要素が絡み合う現在**どこが壊れても**おかしくない
2. 最初に手を付ける計測箇所は**ユーザー視点**
3. 内部がどうなっているかではなく**ユーザー視点を優先した可視化**が必要
    a. 例）データベースサーバーのCPU使用率が急上昇してもユーザーが困っていなければ問題がない
4. 各メトリクスの**ユーザー影響**を常に検討

デザインパターン③：作るのではなく買う

1. 社内の**監視システム成熟度**を把握

   SaaS（少なくとも5年以上はSaaSでOK）

   →**FOSS**（Free and Open Source Software）→**独自開発**



『成熟度を無視してジャンプしてはいけない』

『監視ツールを気にする前に基礎的なことからまずは取り組もう』

2. 安いから：FOSS利用は意外と高い（人件費・運用費）
3. あなたは（おそらく）**監視ツール設計専門家ではない**
4. SaaSを使うと**プロダクトにフォーカス**できるから

デザインパターン④：継続的改善

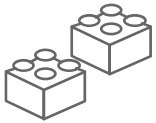
1. Google, Facebook, Twitter, Netflixといった

   先進的企業の監視は素晴らしい

   a. が、何年も時間と投資をしてどんんどん変わりながら成熟しながら発展している。**各組織の成熟度に合った監視**

   b. 素晴らしい監視は計測した**注意深さと改善**から生まれる

監視デザインパターンまとめ



1. **組み合わせ可能な監視の仕組み**は
   モノリシックな仕組みよりも効果的

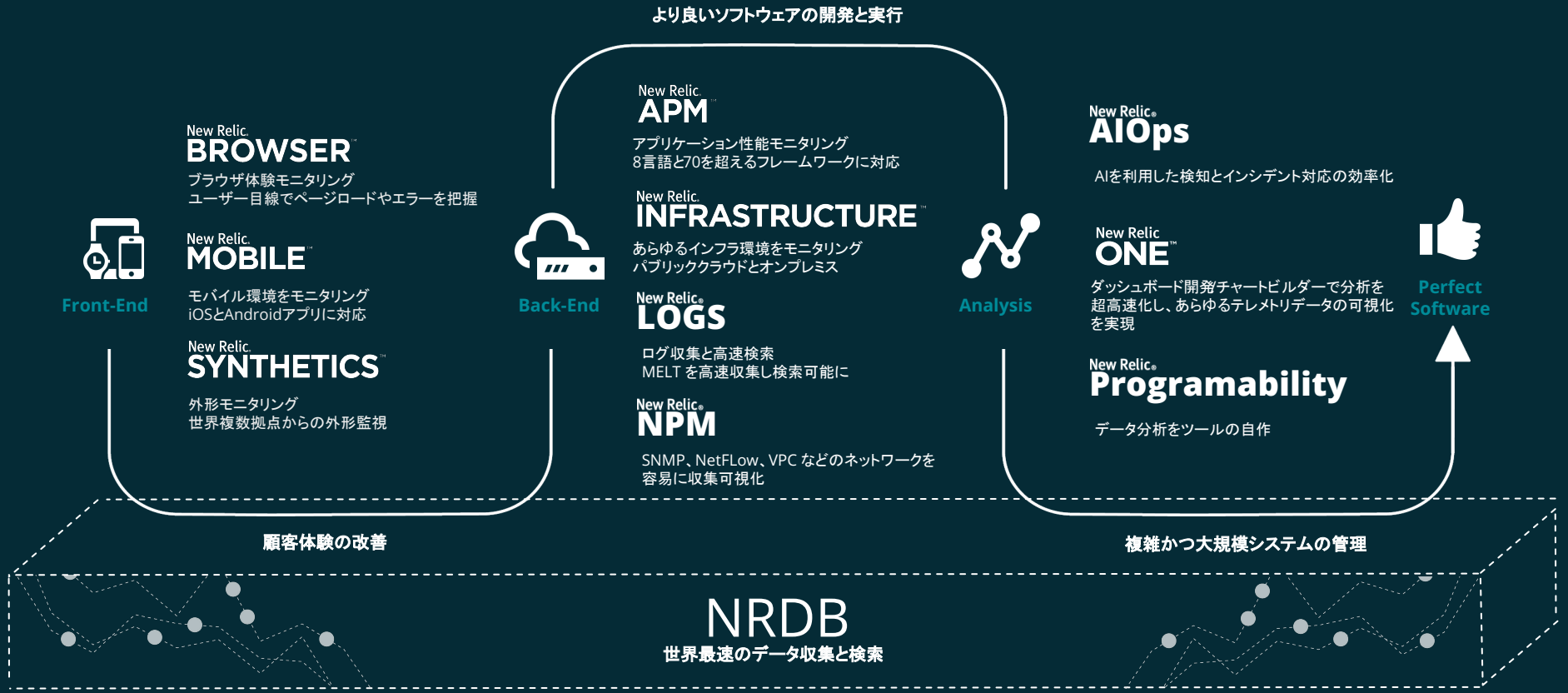
2. **ユーザー視点優先**での監視によって
   より効果的な監視ができる

3. 監視の仕組みは可能な限り
   自分で**作るのではなく買う**こと

4. 組織にあわせ**常に改善**し続ける

インフラ監視にNew Relicで
オブザーバビリティを組み込む方法

New Relic One オブザーバビリティプラットフォーム

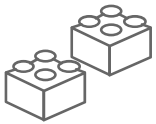


❌ アンチパターンまとめ（再掲）

1. **ツールに依存**しても監視の仕組みは良くなりません
2. **監視は全員がやるべき**仕事でチームや部署内での役割ではない
3. 「これは監視してます」と**チェックをいれるだけ**で済むものではない
4. 監視するだけでは、**壊れたものは直せません**。
5. **自動化**が足りてないということは、何か重要なことを

   見落としている

監視デザインパターンまとめ（再掲）



1. **組み合わせ可能な監視の仕組み**は
   モノリシックな仕組みよりも効果的

2. **ユーザー視点優先**での監視によって
   より効果的な監視ができる

3. 監視の仕組みは可能な限り
   自分で**作るのではなく買う**こと

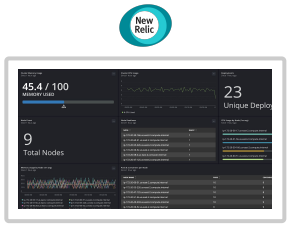
4. **常に改善**し続ける

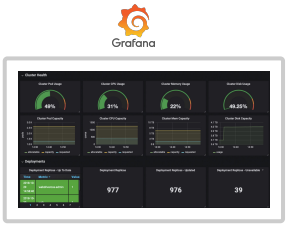
デザインパターン①
組み合わせ可能な監視をNew Relicで

組み合わせ可能な仕組み

様々なテレメトリデータをさまざまに組み合わせて1箇所に集約と可視化

可視化





蓄積

New Relic Database (NRDB)

収集

New Relic エージェント
- APM
- Infrastructure
- Browser
- Mobile

OSSツール、プロトコル、仕様

New Relic APIs

New Relic Telemetry SDKs

New Relic Integrations

Metric API | Event API | Trace API | Log API

Prometheus, etc | .NET | PIXIE | OpenTelemetry | ZIPKIN, etc | fluentd, etc

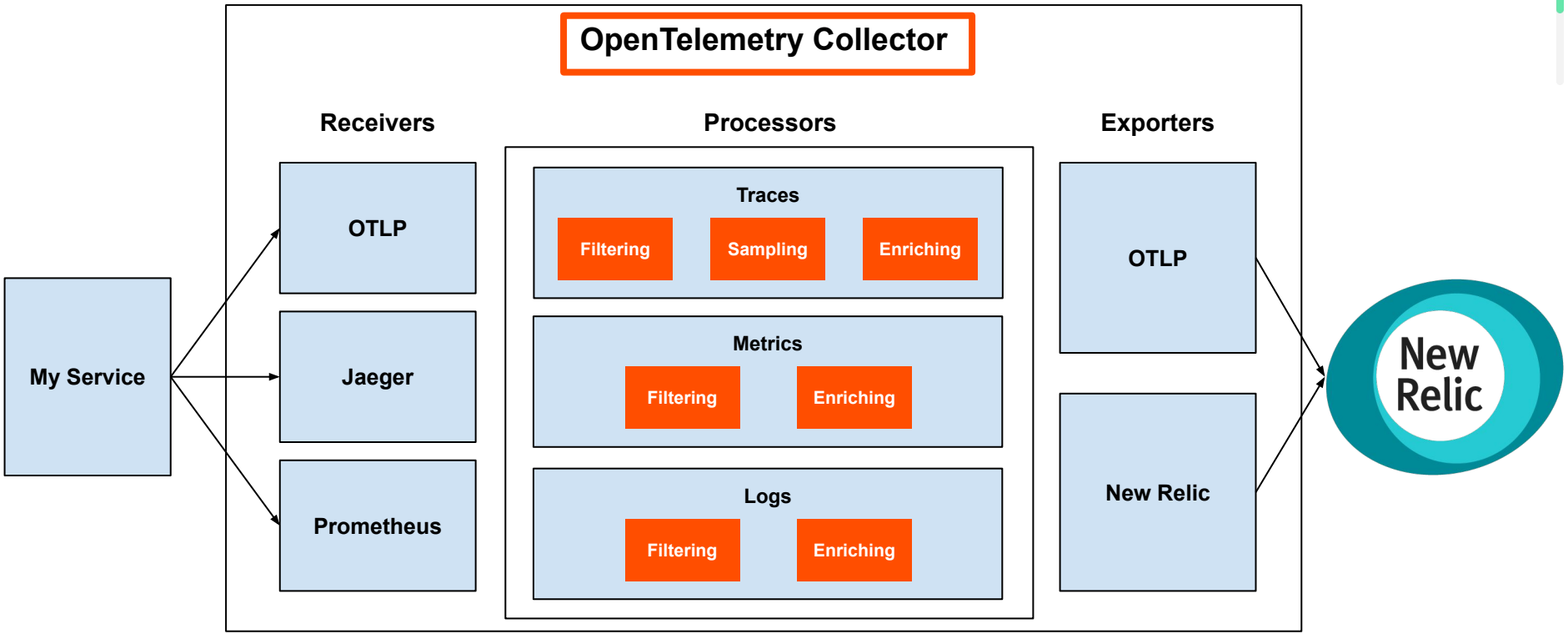
リアルユーザーモニタリング

テレメトリデータソース

アプリケーション、マイクロサービス

インフラ、クラウドプラットフォーム

メトリクス、トレース、ログ対応（OpenTelemetry）



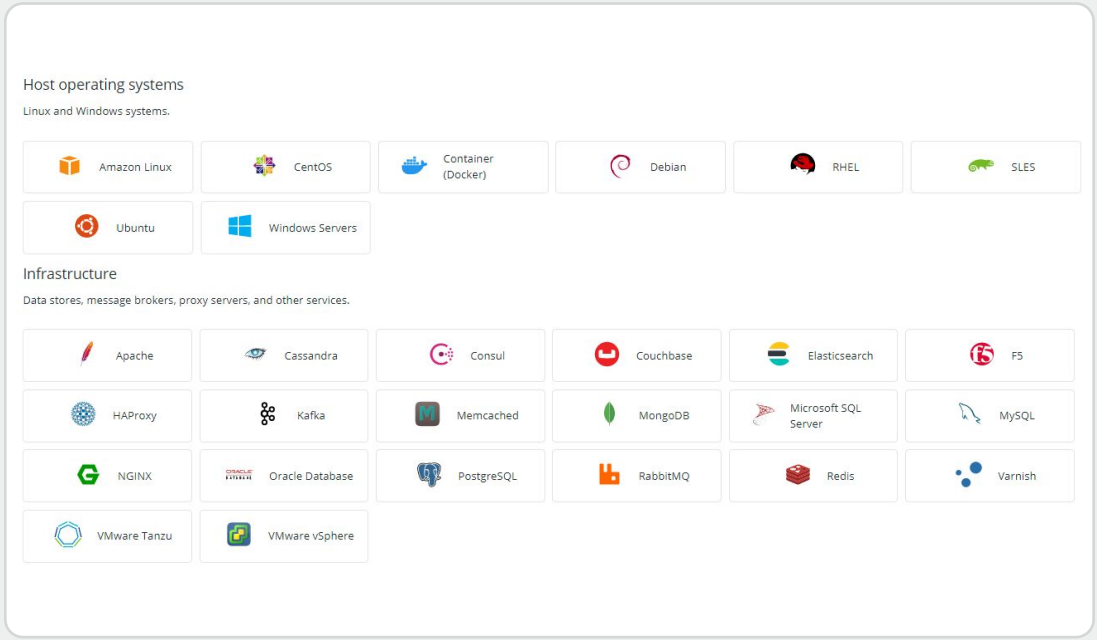
OTLP - OpenTelemetry Line Protocol

New Relic
インフラモニタリング

New Relic Infrastructure

New relic InfrastructureとはNew Relic によるインフラモニタリングサービスです。

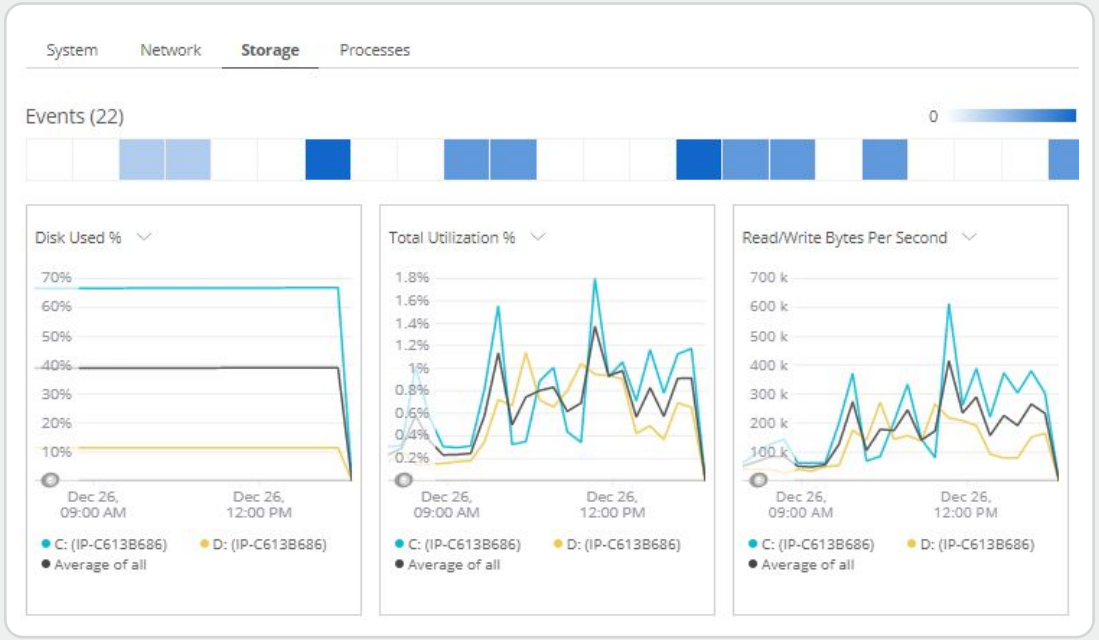
パブリッククラウド, コンテナ, OS,ミドルウェア, Network, OSS等の情報を収集します。



Host/Process モニタリング

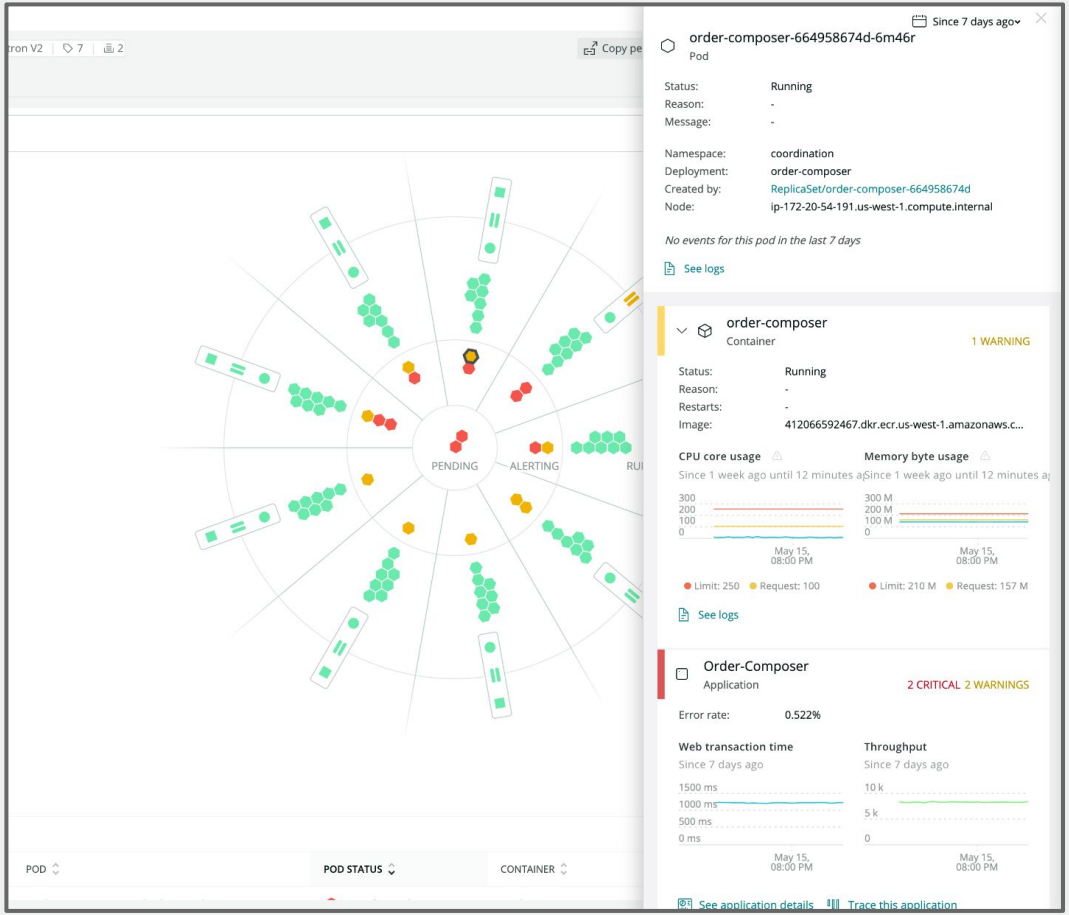
CPU,Memory,Network,Storageの値を確認することができます。

ProcessSampleを有効にすることでプロセス毎のリソース使用量を確認することもできます。



コンテナ
モニタリング

kubernetesのクラスターを視覚化し、どのPodで何がおきているかすぐに辿り着ける。



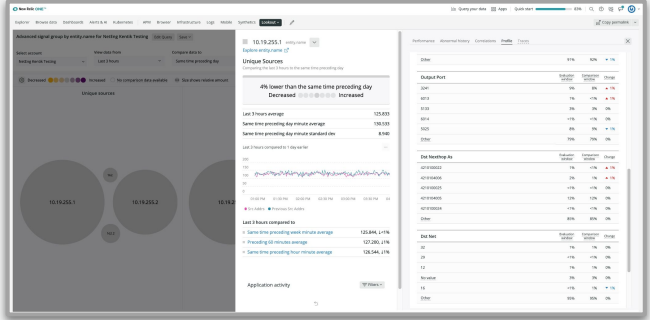
Network Performance Monitoring NEW

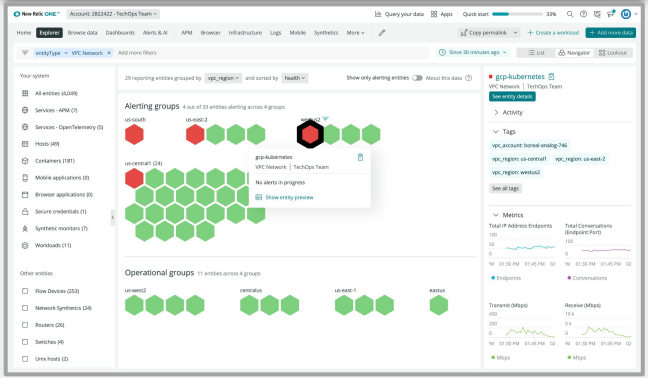
Network Performance Monitoring（NPM）は、運用チームや開発者が、ネットワークへの侵入や排除を迅速に行うことで検知を効率化し、MTTRを短縮することができます。SNMPやNetFlow、VPCなどに対応しオンプレミスからモダンクラウドネットワークまで包括的にモニタリング可能に

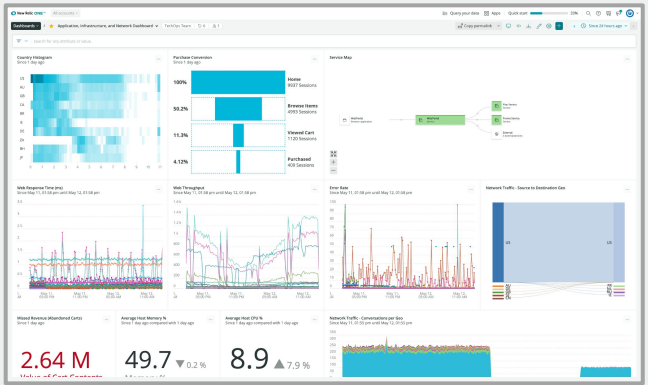
メリット:

- ✓ Streamlined detection: ネットワークチームに連絡することなく、ネットワークのルールインやルールアウトが可能
- ✓ Improved MTTR: 適切なリソースを適切なレイヤーでより早く適用する。
- ✓ Increase operational efficiency: AIと自動アラートでネットワークの異常をデフォルトで検出。

https://newrelic.com/platform/network-observability







デザインパターン②
ユーザー視点（SLAとRUM）監視をNew Relicで実現！

サービスレベル 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

フォーカスするべきは**SLIとSLO**

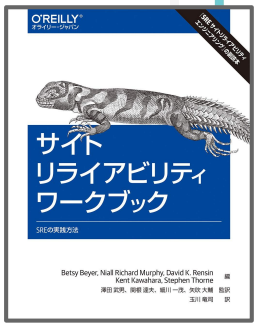
SLIのタイプ

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。

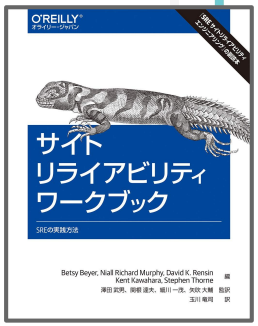
システムによって使い分ける。

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



New Relic.

顧客視点のSLO設定

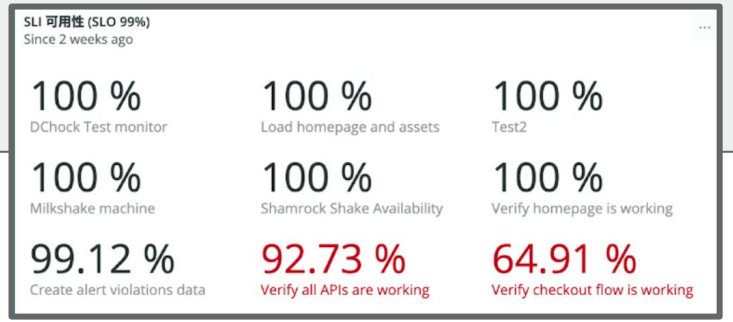


- SLOは、**サービス顧客の目標レベルで信頼性を設定**します。
- このしきい値を下回ると、ユーザーは不平を言い始めるか、サービスの利用を停止する可能性があります。
- 最終的に、ユーザーの幸せが重要です。（中略）

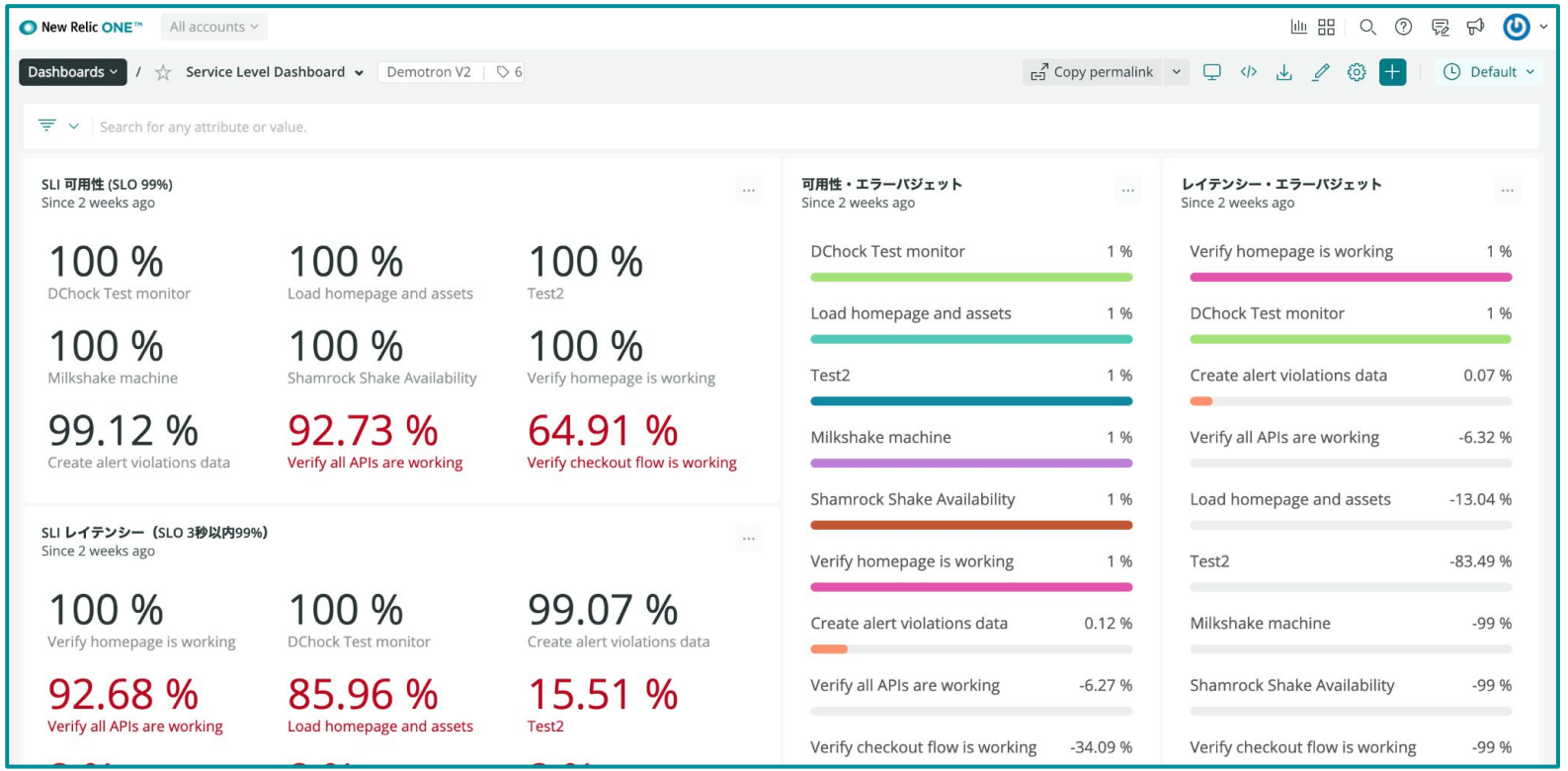
  お客様の満足を維持するためにサービスの信頼性を維持しています。

可用性サービスレベルを NRQL（New Relic Query Language）で表す

```
FROM SyntheticCheck
 SELECT percentage(count(*), where result = 'SUCCESS')
 facet monitorName
 since this month
```



ユーザー視点サービスレベルモニタリング

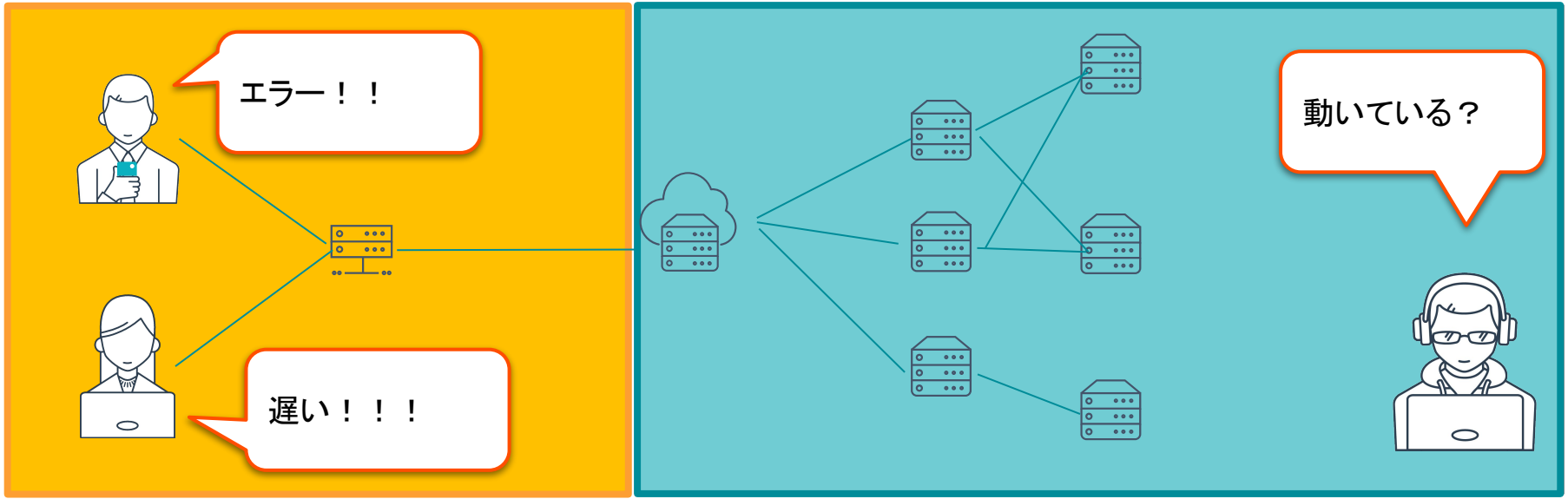


New Relic SYNTHETICS™

New Relic.

リアルユーザーモニタリング（RUM）

ユーザーにとってはリクエストを投げて画面に描画されるまでが応答時間です。
データセンター外で発生するエラーや遅延はクライアントから見ないとわかりません。



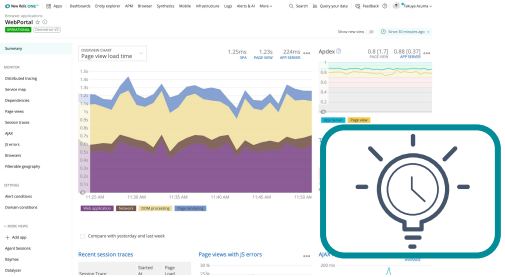
ユーザー側

サービス提供側

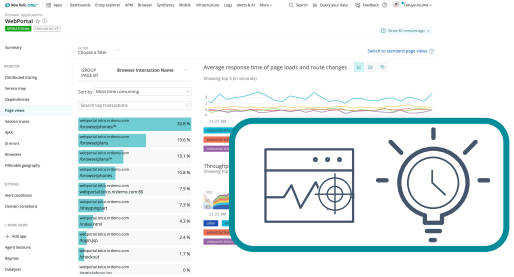
New Relic.

New Relic Browserで実現するRUM

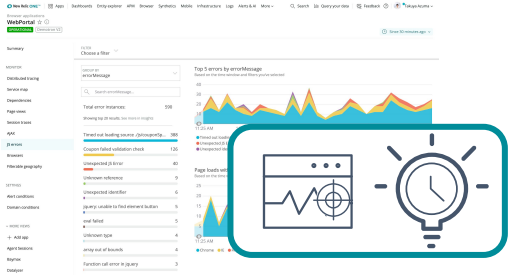
パフォーマンス概要の把握



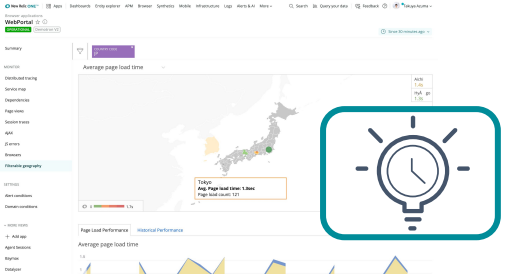
ページ毎の状況確認



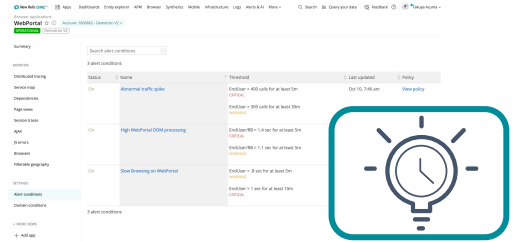
エラー分析



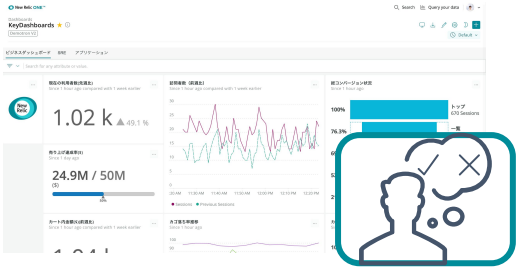
ブラウザ / Geo



アラート



行動分析



早期解決
プロアクティブな対応

予期せぬ課題に
高速にアプローチ

顧客体験とシステムの
因果関係を把握し、理解を得る

New Relic.

Core Web VitalにNew Relicの活用

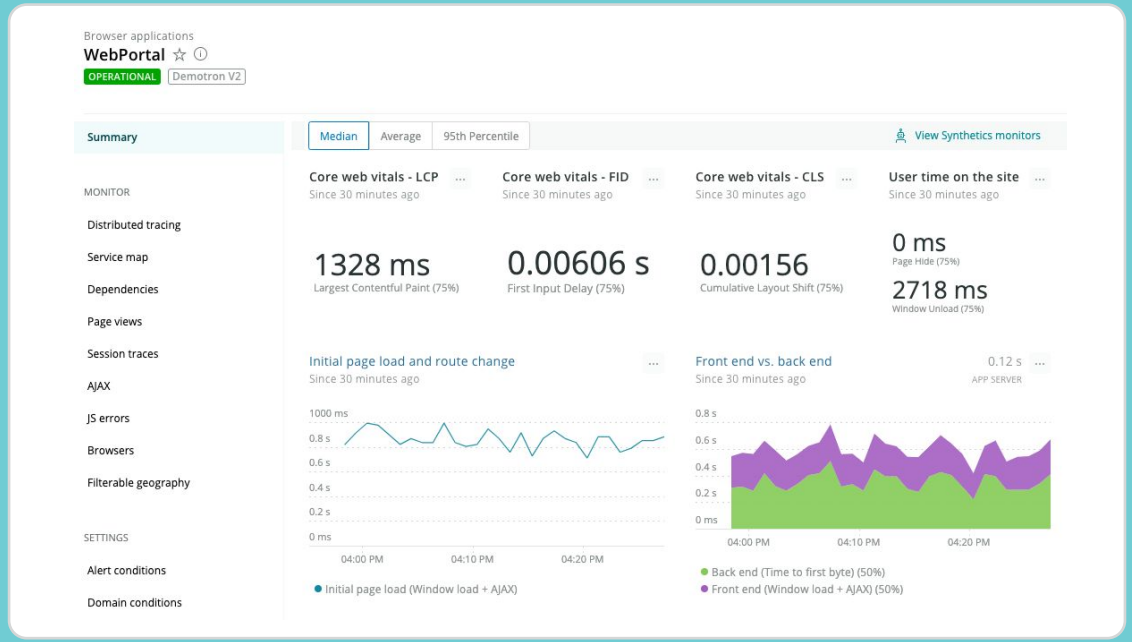
Core Web VitalsはSEO対策だけでなく、フロントエンジニアがよく気にするGoogleの「Lighthouse」などのスコアと同等。

”ユーザー体験”

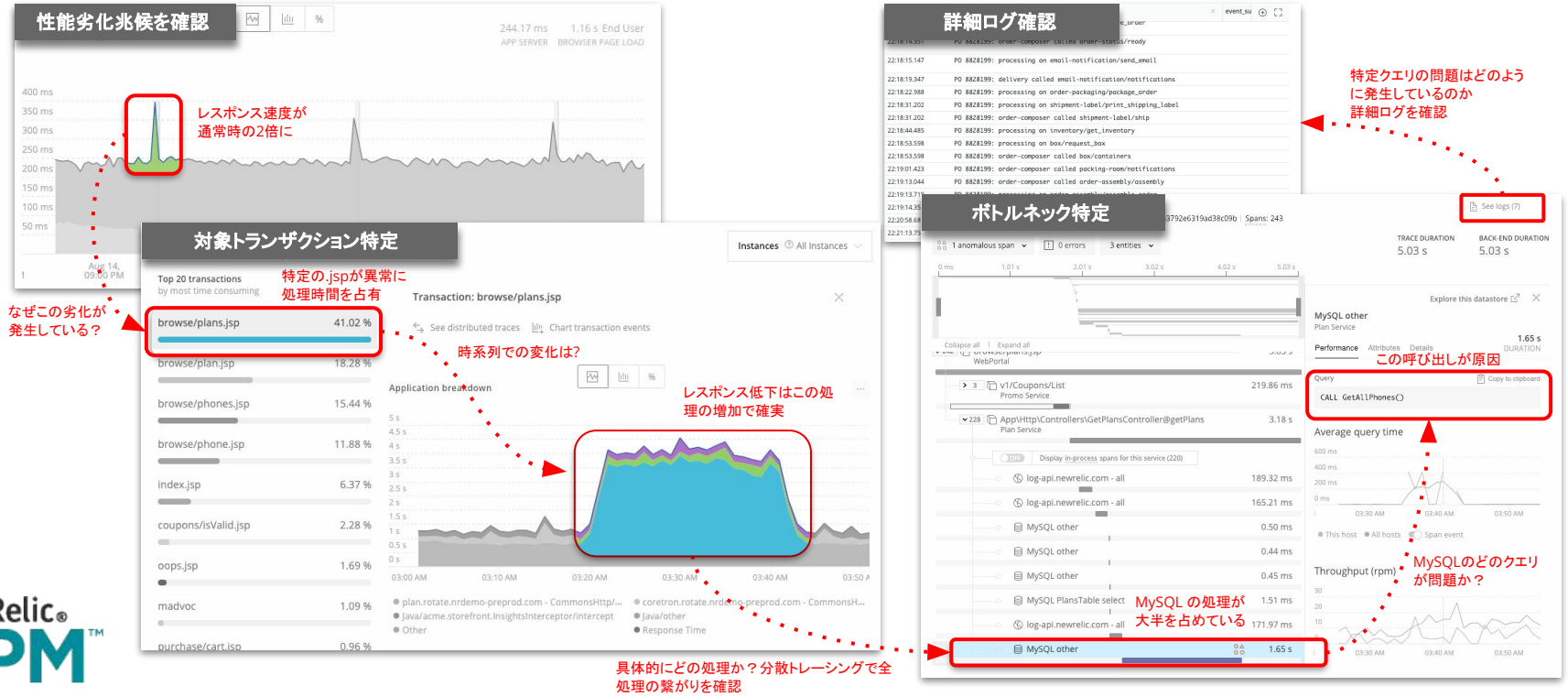
Lighthouseなどを使っても静的にしか値が分からないが、New Relicだと
リアルな情報が**リアルタイム**にわかる

Googleが始めたユーザー体験向上のための指標群

LCP: 一番大きなコンテンツ（背景画像など）の描画
FID：初回入力の遅延速度
CLS:JSによるコンテンツ追加などでの画面崩れ



ユーザーの体験したエラーや遅延から原因の特定



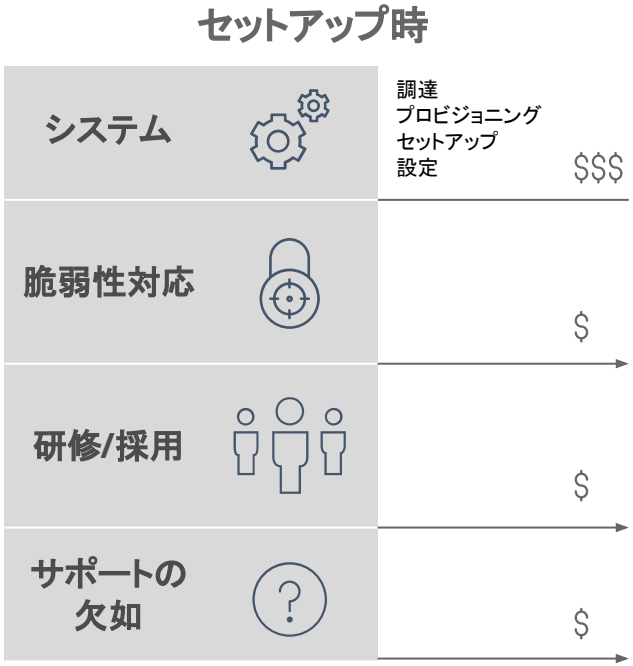
New Relic APM™

性能劣化問題をコードレベルで特定する

デザインパターン③
作るのではなく買う

OSSツールの拡張は、高コストの”運用のための運用”

フリーなOSSのデプロイと保守に必要な時間とリソースは、拡大してしまえばとてもコストがかかります。



OSSへの投資の**50%以上**は、
トータルコストに大きなメリットが
得られていない※1

※1: Source: Gartner, "What Innovation Leaders Must Know About Open Source Software," Arun Chandrasekaran and Mark Driver, 26 August 2019

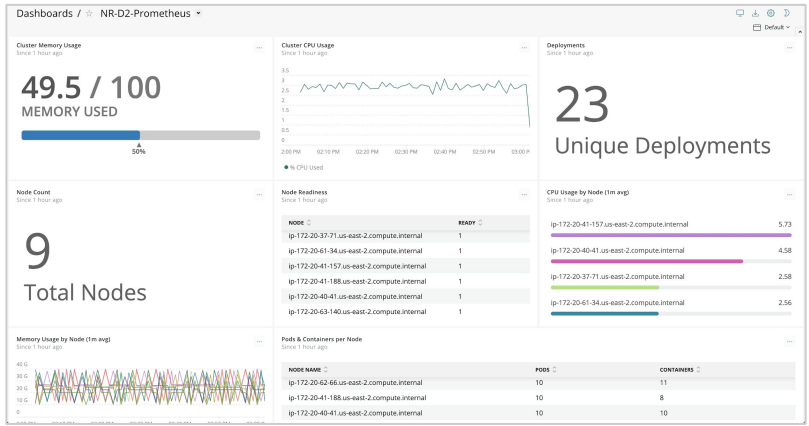
”運用のための運用からの解放”

あらゆるソースからのテレメトリーデータの分析、可視化、アラートを一箇所で実施

New Relic Agents

Applications and microservices
APM
Mobile
Browser

Infrastructure and services
Hosts
Hybrids
Multi-cloud



Metrics | Events | Logs | Traces

Third-Party Data

Integrations
SDKs
APIs

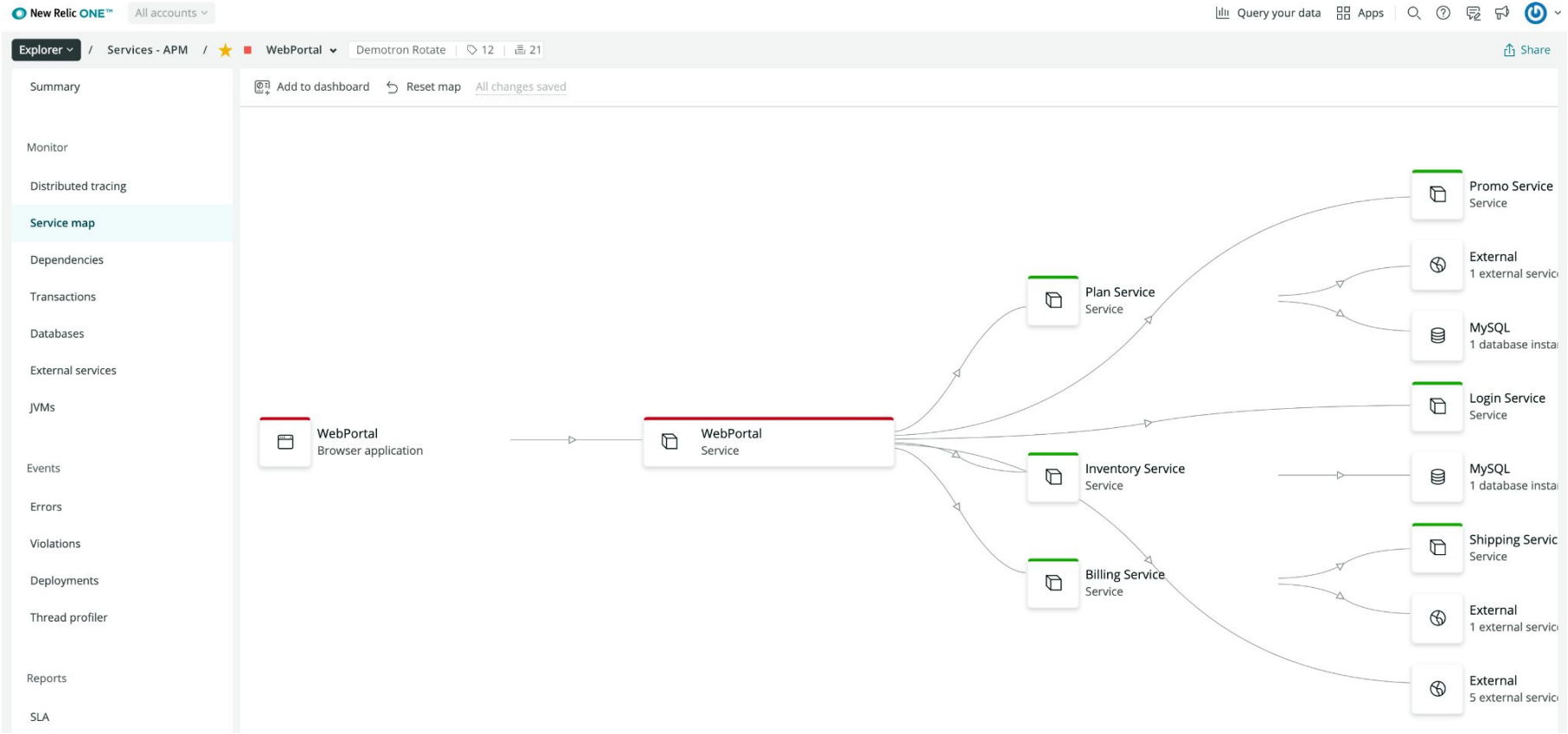
Metrics
Prometheus

Traces
ZIPKIN
OpenTelemetry

Logs
logstash
fluentd

デザインパターン④
継続的改善をNew Relicで

サービスマップをつかった全貌の可視化

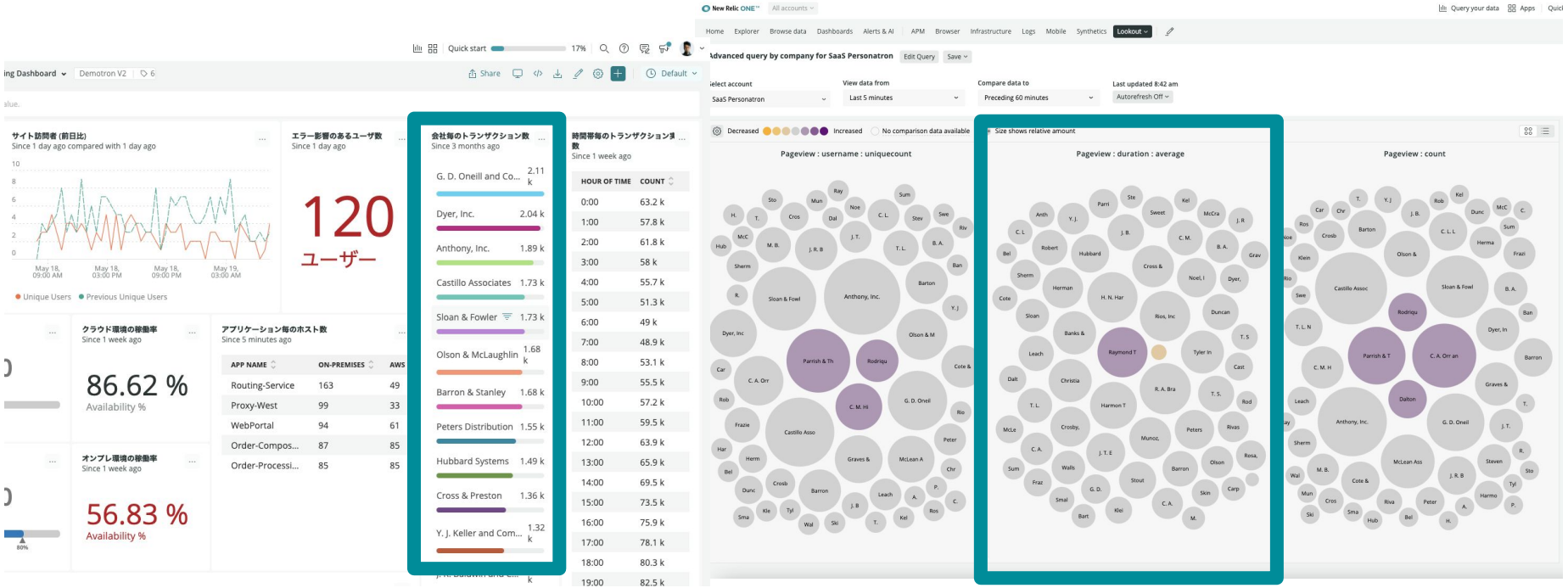


New Relic APM™

Applicationの品質改善 (システム全体の把握と問題箇所の明確化)

New Relic.

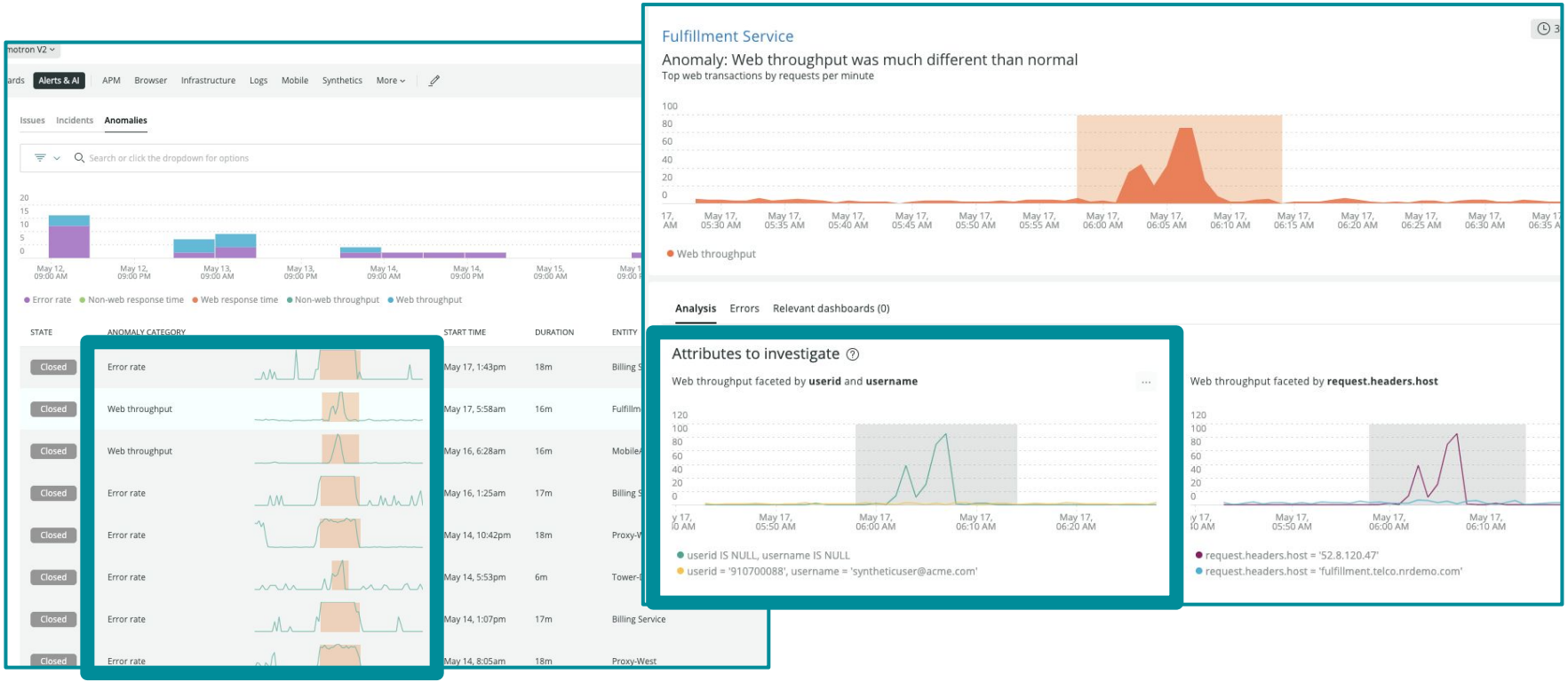
論理Key（ユーザー名や企業名）から分析が可能に！



Multi Tenant/User分析によって
どの企業のどのユーザーに不満があるのかを把握

New Relic.

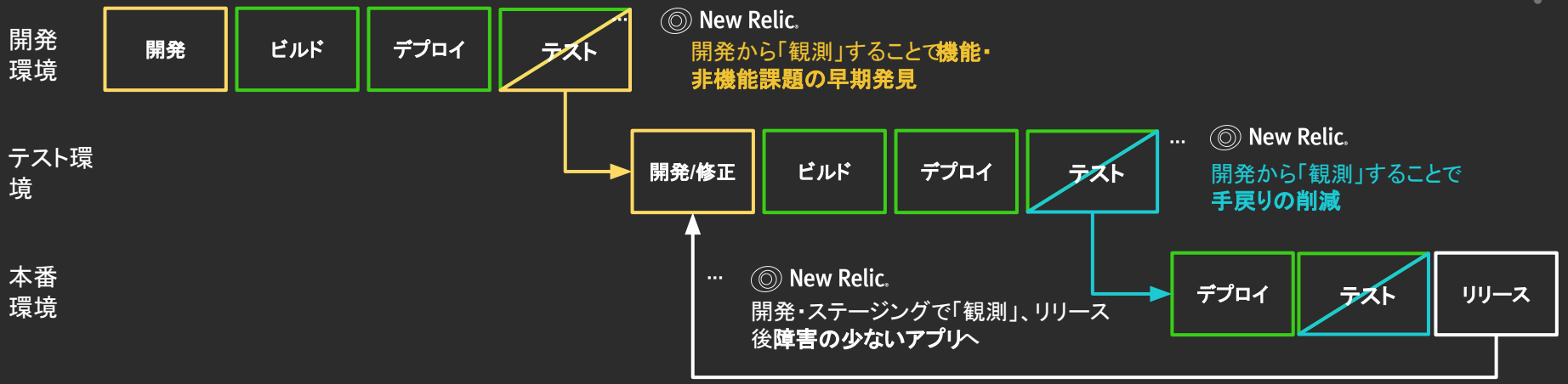
AIOpsで未知の問題発見と原因特定



AIOpsによる問題の自動検知と特定

New Relic.

全環境にNew Relicを！



… 開発者

… CI/CD

… 品質評価

New Relic

テストフェーズやリリース後の問題早期発見で手戻り少なく、さらにデプロイごとのアプリ性能を観測し信頼性向上。

New Relic One Demo

DEMO

まとめ

- アンチパターン5
  - **ツールに依存**せず、**監視は全員が取り組み、目的ある監視**をし、**監視と改善をセット**とし、監視設定を**自動化**しよう！
- デザインパターン4
  - **組み合わせ可能な監視**で、**ユーザー視点優先**での監視を、**監視SaaSを買って**実現し、自分たちに合った**監視を模索**しよう！

- **インフラエンジニアのスキルを活用して、モダン監視へ踏み出そう！**

Q&A

Thank You

tshimizu@newrelic.com
@photographed

New Relic.