

クラウドで手軽に始めるモダンなモニタリング

株式会社クアンド 西 孝幸

自己紹介

西 孝幸

テックリード



QUANDO

沖縄にUターンして7年
自動化が大好きです





地域産業・レガシー産業のアップデート
update the industry

トピック

- クラウド時代の**モニタリング**の課題
- ログの重要性とモダンな**モニタリング**
- クラウドで手軽に始めるモダンな**モニタリング**
- モダンな**モニタリング**を活用してサービスの成長を支える

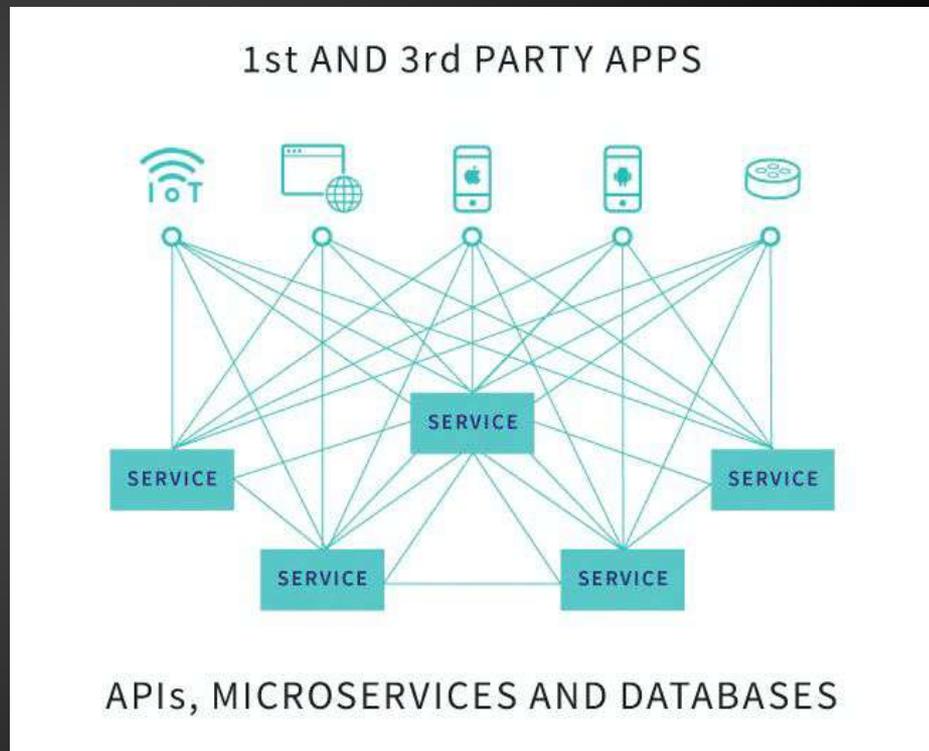
クラウド時代のモニタリングの課題



時はマイクロサービス全盛期

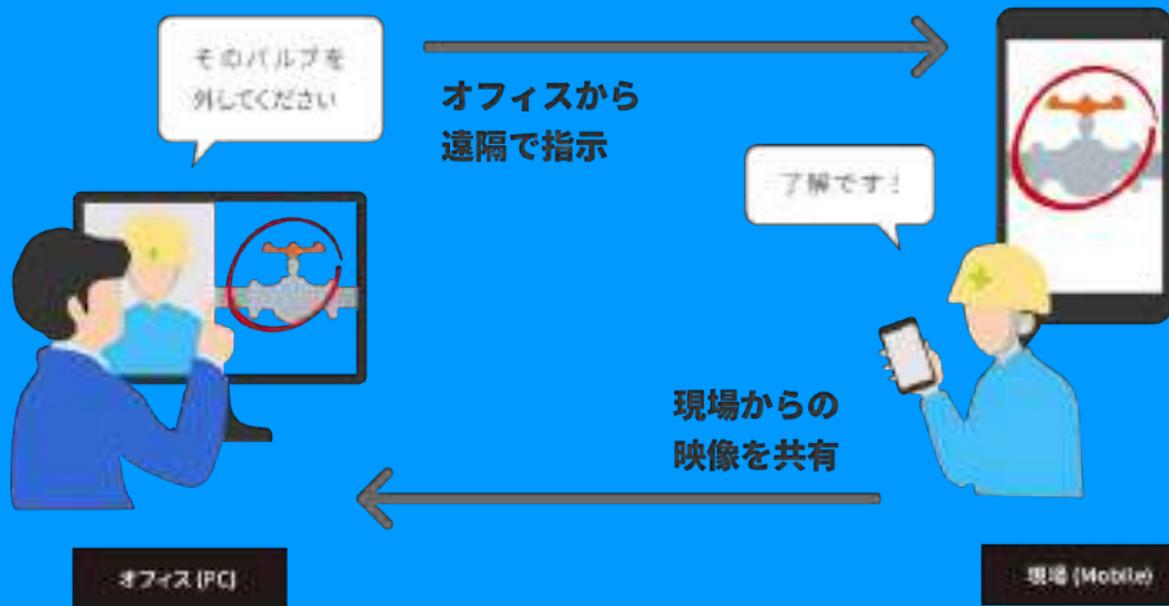
マイクロサービスが普及したことで、

- サービスを跨いでログをトレースする必要がでてきた
- 分散したサーバ全てを俯瞰してモニタリングするコスト増大という課題が発生した。



SynQ Remote におけるモニタリングの課題

現場に最適なりモートワークツール



通話機能の紹介



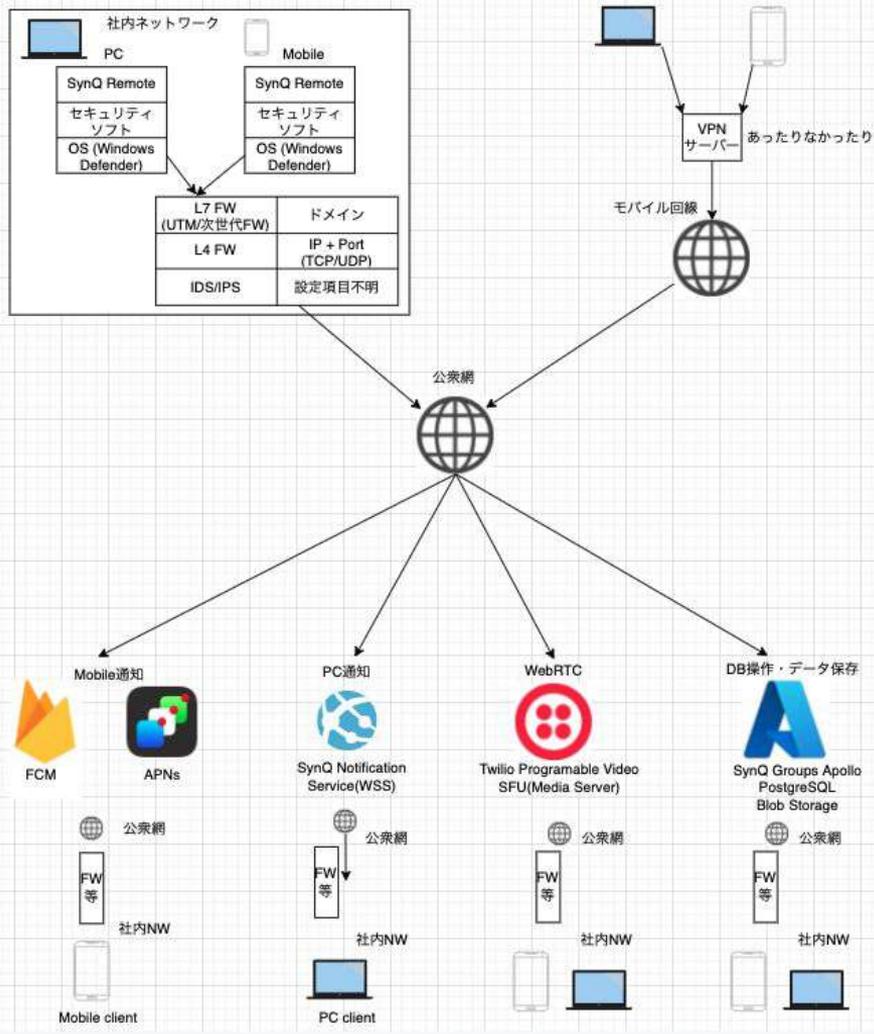
 **SynQ Remote**

現場情報共有プラットフォーム

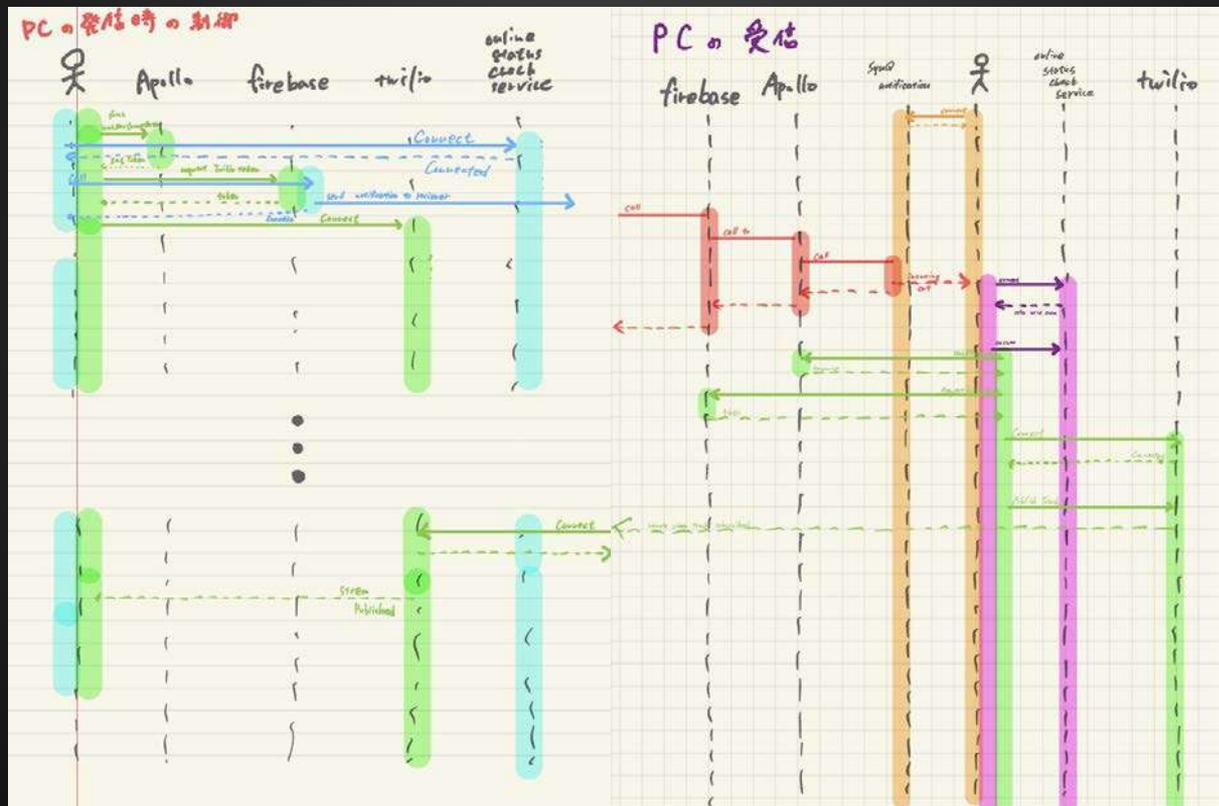


発着信を実装

WebRTCを利用した通話を実現しており、特に発着信（通話開始のプッシュ通知）を実装した為、トラブルシュー트가困難だった。



シーケンス図を見ると複雑さが一目瞭然



全てをトレースするのは難しい

通話開始ボタンを押してから着信通知が届くまで、ネットワークをまたいで、

- APIサーバ
- Twilio
- プッシュサービス
- ステータスチェックサービス

間にて相互に通信が発生し、ネットワーク起因などのトラブルが発生した場合、原因追求が難しかった。

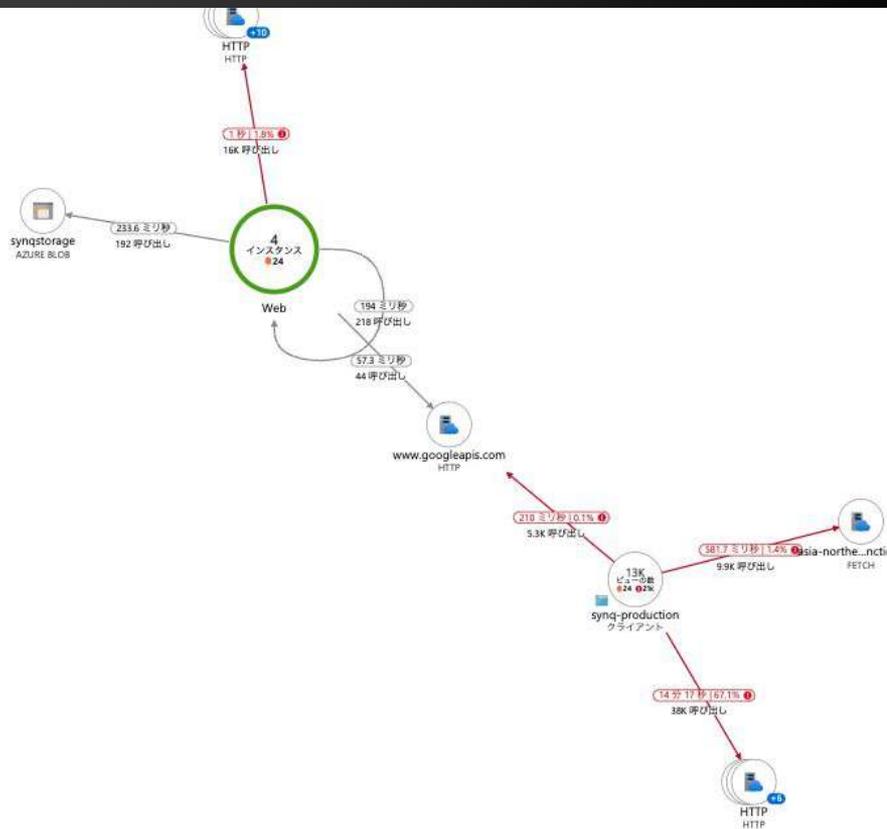


全てをモニタリングするのも大変

同様に、

- バックエンドサーバ
- Webサーバ
- 認証サーバ
- ステータスチェックサーバ

といった複数のサーバのモニタリングも行う必要があり結構大変。



マイクロサービスが生み出したカオス

マイクロサービスでサービス毎の責務を分割し開発しやすくなったが、同時にリクエストの「あみだくじ」が生まれ、起きていることの把握が難しくなった。

従来のトレーシング手法では、「クモの巣」とも表現される複雑なリクエストの実態を把握し、トラブルの原因を突き止めることは難しい。



自前で構築することも可能だが

それはサービスが大きくなり、従量課金のサービスメリットがなくなった場合。

スタートアップや通常のサービスでは データストアを自前で運用するよりは、解決したい問題にフォーカスできるマネージドサービスに比較優位がある。



ログの重要性とモダンなモニタリング



なぜログが必要か？

ログはサービスの健康状態を示す。

- 障害対応の手がかりや障害発生の予兆を見つける
- サービス改善に役立てる
- 性能検証
- ユーザ行動の可視化

といった、サービスを安定して動かし続けるために必要な情報を得ることができる。



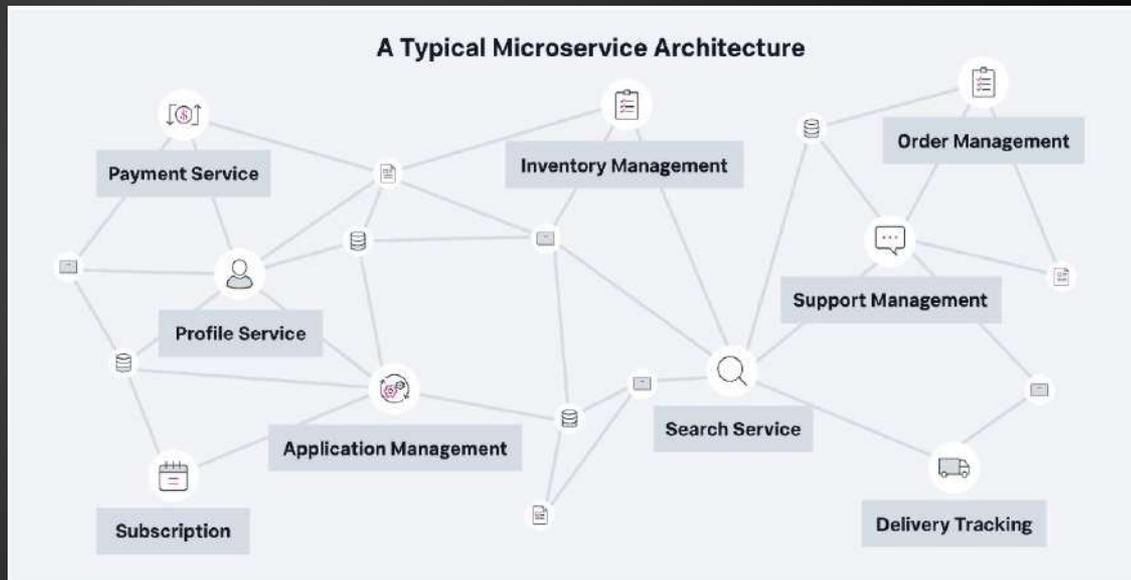
サービスを安定して動かし続けるために

サービスを動いたではなく、動かし続けるためには、サービスの各機能で起きていることをログとして収集し、把握する必要がある！



分散トレーシングとは

分散トレーシングとは、マイクロサービスアーキテクチャで構築されたアプリケーションを監視する手法のこと。



出典: [分散トレーシングとは](#) : splunk

分散トレーシングの効能 - 1

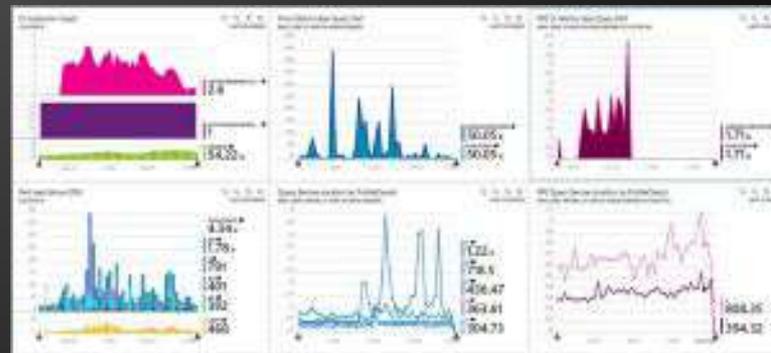
分散トレーシングを使用すると、監視中のアプリケーションを通過するリクエストやトランザクションの経過を観察し、アプリケーションのパフォーマンスに影響するボトルネックやバグなどの問題を正確に特定できるようになる。



分散トレーシングの効能 - 2

分散トレーシングは、各サービスやモジュールを通過するリクエストを追跡して、リクエストが辿ったすべての軌跡をエンドツーエンドで明らかにし、追跡の問題と数パフォーマンスの問題を解決できる。

どのインスタンスがアプリケーションの遅延や失敗を招いているのかを特定したり、その解決方法を確認して、パフォーマンスを監視できるようになる。



分散トレーシングに必要な要素

必須

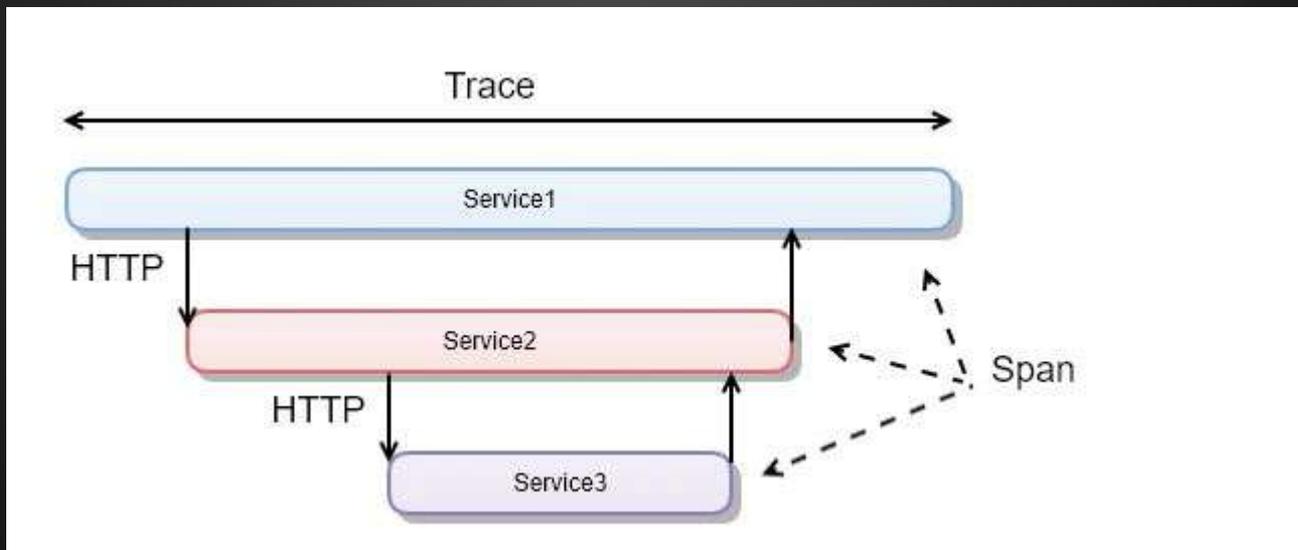
- Operation Name
 - Start/Finish Timestamp
 - Span Context
- Baggage Items
trace / span ID

オプション

- Span Tags
 - Span Logs
 - References
- 他のSpanとの関連性

分散トレーシングの基本概念 - 1

- Span: 1サービス内の処理を表す。
- Trace: Requestのstart-endを含むSpanの集合をあらわす。



分散トレーシングの基本概念 - 2

Visualization

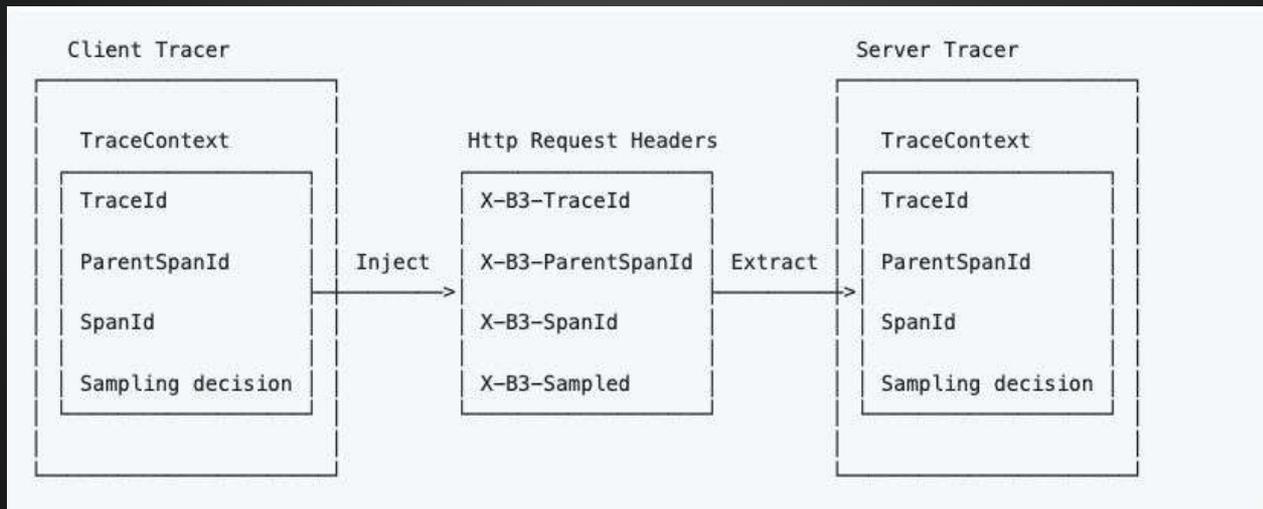
有効非巡回グラフ(DAG)

タイムラインビュー



分散トレーシングの基本概念 - 3

各Spanのデータの伝搬(Propagate)



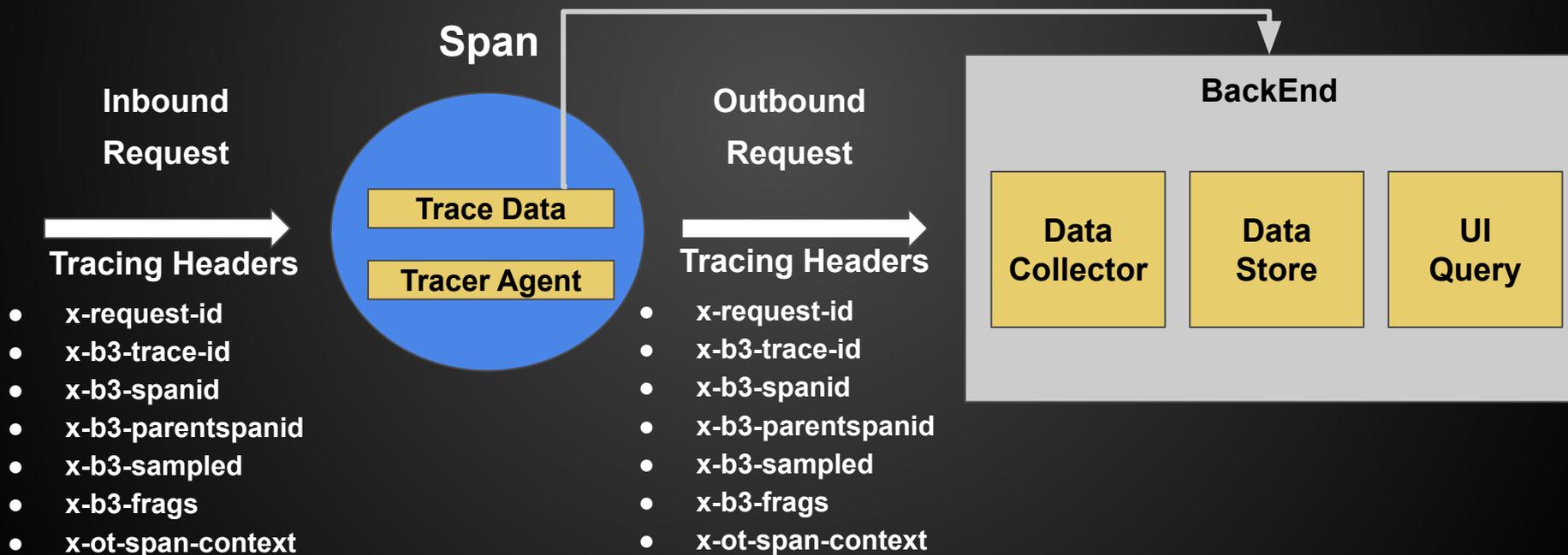
- Trace ID = xxx1
- Span ID = ccc
- Parent Span ID = **aaa**

出典: [b3-propagation](#)

- Trace ID = xxx1
- Span ID = ccc
- Parent Span ID = **bbb**

分散トレーシングの基本概念 - 4

バックエンド = 最終的にデータを集約して集計する箇所
Agent Send Trace Data to backend



分散トレーシングの標準化仕様



Open Census

- 分散トレーシングの為のライブラリ群 (多言語)
- GoogleOSSコミュニティの一部
- 対応バックエンド(Exporter):
Jaeger、Zipkin、AWS-XRay、
Stackdriver Tracing、
Azure Monitor



Open Tracing

- 分散トレーシングの為のAPI仕様、
Framework/ライブラリ
- CNCFプロジェクト
- 対応バックエンド:
Jaeger、Zipkin、ElasticAPM、
Datadog、
Apache SkyWalking

ベンダーニュートラルな標準化仕様へ



2019年3月、それぞれ分散トレーシングの標準化として開発されていたOpenCensusとOpenTracingがOpenTelemetryとして統合され、ベンダーニュートラルな分散トレーシングの標準化として再スタート！

今後の動向に注目。

クラウドで手軽に始めるモダンなモニタリング



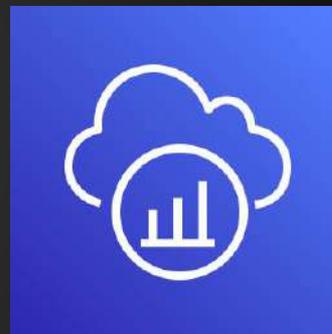
分散トレーシングの紹介 - 1

AWS-XRay

AWSサービス向けに提供されている分散トレーシングサービス。

アプリケーションで使用されているAWSサービスの実行状況を可視化し、パフォーマンス異常やエラーの原因追求などが可能。

当然、AWSで提供されている、EC2、ECS、Lambda、Elastic Beanstalkなどと統合されており、用意に構築できる



分散トレーシングの紹介 - 2

Google Cloud Trace

GCPが提供するマネージドな分散トレーシングサービス。

Googleのデータウェアハウス、Google Bigqueryとの親和性が高く、データウェアハウスの機能をフル活用して分散トレーシングのログを解析・活用できることが特徴。



分散トレーシングの紹介 - 3

DataDog

Datadog .Inc が提供する、SaaS型運用監視サービス。AWS、GCP、Azureと主要なクラウドサービス向けのインテグレーションを提供しており、クラウドを横断してモニタリングできる点がポイント。

これから分散トレーシングやモニタリングを導入されるならこちらがおすすめ。



分散トレーシングの紹介 - 4

Application Insights

Azureが提供しているモニタリングサービスの Azure Monitor に統合されている分散トレーシングサービス。

エージェントがパッケージ形式で配布されており、Node.js、Python、Dart(Flutter)などのパッケージを組み込むことで、Azure以外のサービスを含めた分散トレーシングを構築することが可能。

クアンドではこちらを採用。



Application Insights

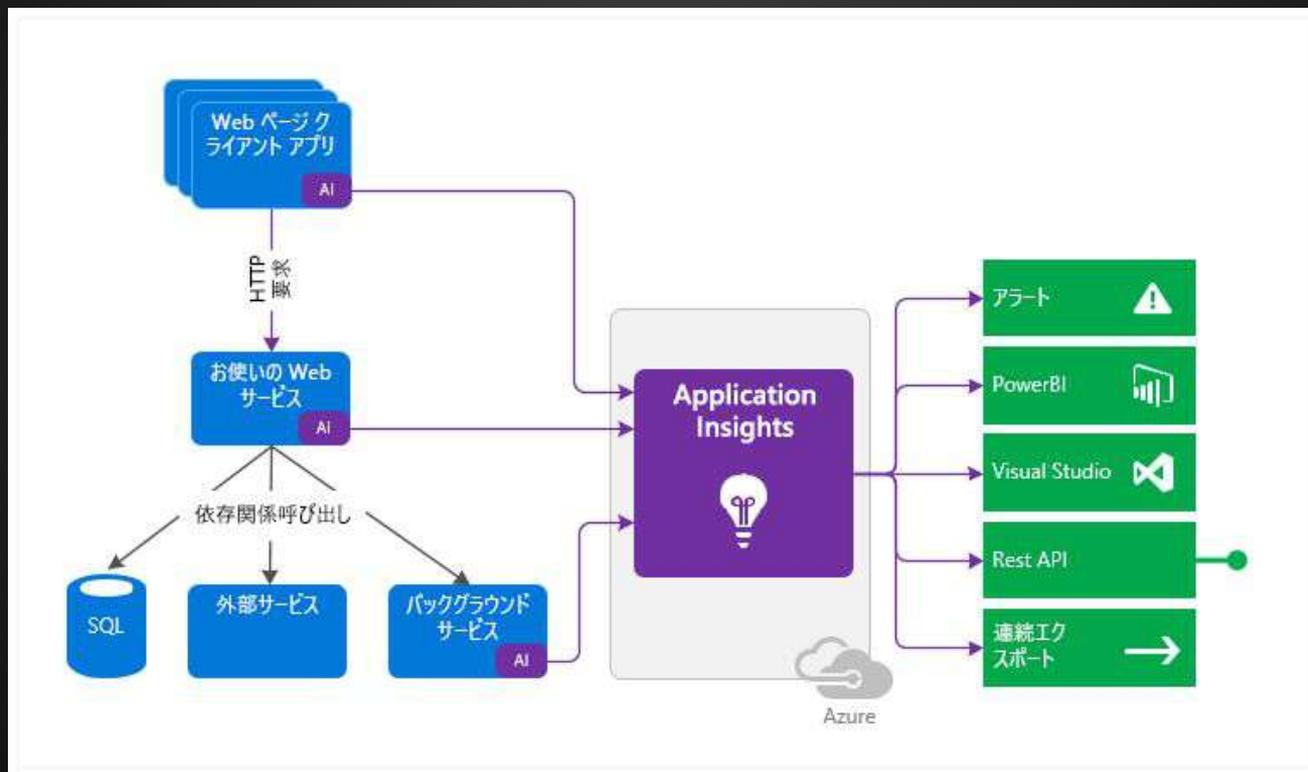
SynQ RemoteへApplication Insightsを組み込む



Application Insights



Application Insightの仕組み



出典: [Application Insights とは何か?](#)

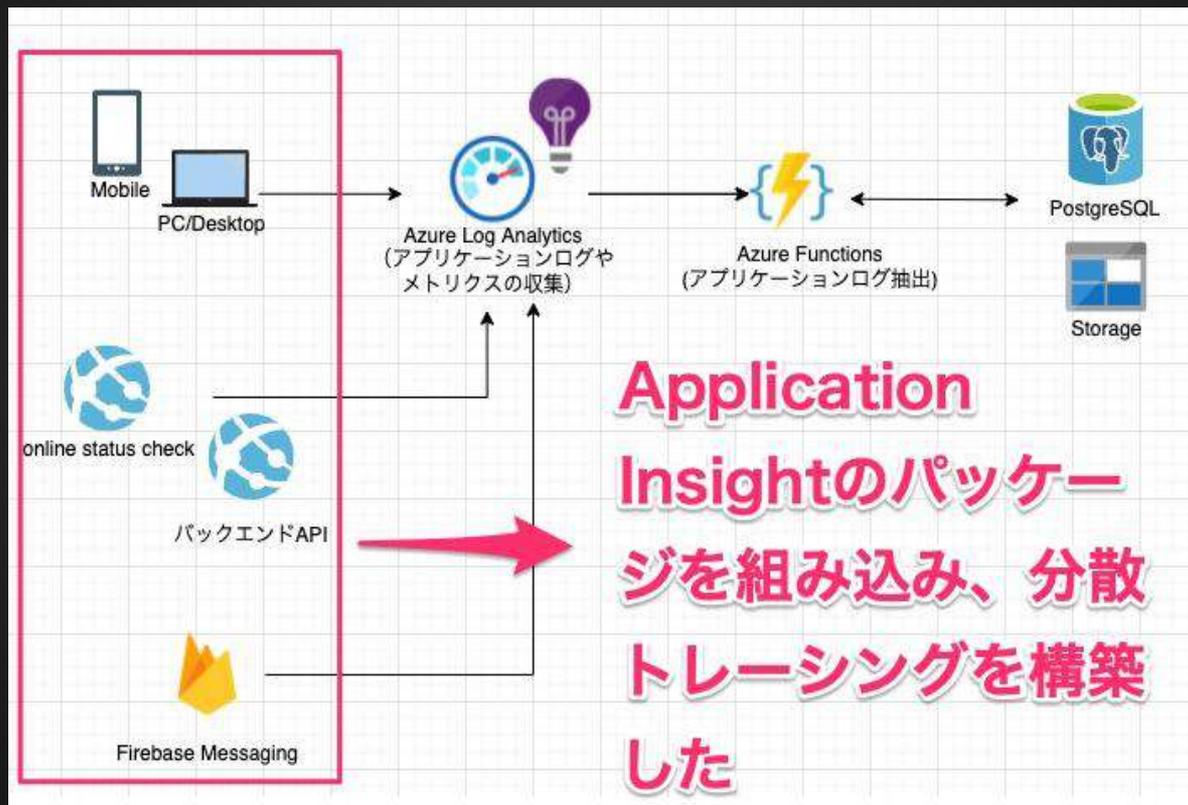
Application Insightの主要機能

- スマート検出と手動のアラート
- アプリケーションマップ
- プロファイラー
- 利用状況分析
- インスタンスデータのトランザクションの検索
- 集計データのメトリックエクスプローラー
- ダッシュボード
- ライブメトリックストリーム
- 分析
- Visual Studio
- スナップショットデバッガー
- PowerBI
- REST API
- 連続エクスポート

SynQ Remoteへ分散トレーシング導入

- パッケージを利用した分散トレーシングの組み込み
 - Node.js、Dartなどのパッケージをソースに組み込み、ログ、メトリクス情報をAzure上で収集できるように設定
- ログの永続化
 - 収集したログをさかのぼって分析できるように、ストレージへのログエクスポートを構築
- アラートルールの設定
 - 通話関連のエラーを検知した際のアラート通知を構築

構築のイメージ



モダンな**モニタリング**を活用してサービスの成長を支える



分散トレーシングを導入することで

ログをためる → エラーを見つける → 改善するというサイクルを回すことができるようになった。

→ 原因がわかれば、対処できる！！



事例 1: モバイルのバリデーション

ふりがなに英字が入るトラブルがあり、分散トレーシング上でログを追いかけていくと、モバイルからバックエンドに英字が入力された状態でリクエストが飛んでおり、バリデーションに問題があることがわかった。

The screenshot shows a distributed tracing tool interface. The main area displays a table with columns for 'client_ip', 'client_CountryOrRegion', and 'client_Browser'. A red arrow points to a value in the 'client_Browser' column, which is highlighted in blue and contains the text 'ふりがなに英字が入っている'. Below the table, a log entry is visible, with a red box highlighting a field containing the text 'lastNameReading: "CSO"'. The interface also includes a search bar, filters, and a sidebar with 'Application insights' and various metrics.

事例 2: 発着信のログ

分散トレーシングを導入したことで、通話開始ボタンが押されてから通話終了ボタンが押されるまでの一連のトレースログを取得、分析できるようになった。

The screenshot displays a distributed tracing interface. At the top, a query editor contains the following SQL-like query:

```
1 traces
2 | where customDimensions contains "XXXXXXXXXXXXXXXXXXXX"
3 | order by timestamp desc
```

Below the query editor, there are two red arrows pointing to specific features:

- クエリ形式でログ抽出できる (Can extract logs in query format)
- クエリからアラートルールを作成することも可能 (It is also possible to create alert rules from queries)

The results section shows a table of traces. The table has columns for timestamp, message, itemType, and cus. The results are filtered to show traces from the last 3 days.

timestamp [大阪、札幌、東京]	message	itemType	cus
> 2021/12/2 18:37:28.953	switchStream	trace	{\"u
> 2021/12/2 18:37:16.233	switchStream	trace	{\"u
> 2021/12/2 18:36:45.376	addTextMessage	trace	{\"u
> 2021/12/2 18:36:27.499	toggleLocalAudioStatus	trace	{\"u
> 2021/12/2 18:36:27.499	Success to unmute local audio	trace	{\"u
> 2021/12/2 18:35:42.449	switchStream	trace	{\"u
> 2021/12/2 18:35:33.494	addTextMessage	trace	{\"u
> 2021/12/2 18:35:18.162	switchStream	trace	{\"u
> 2021/12/2 18:34:35.767	toggleLocalAudioStatus	trace	{\"u
> 2021/12/2 18:34:35.766	Success to mute local audio	trace	{\"u

ログと仲良くなることで

ユーザに安心してサービスを使ってもらうため、裏側で
頑張る

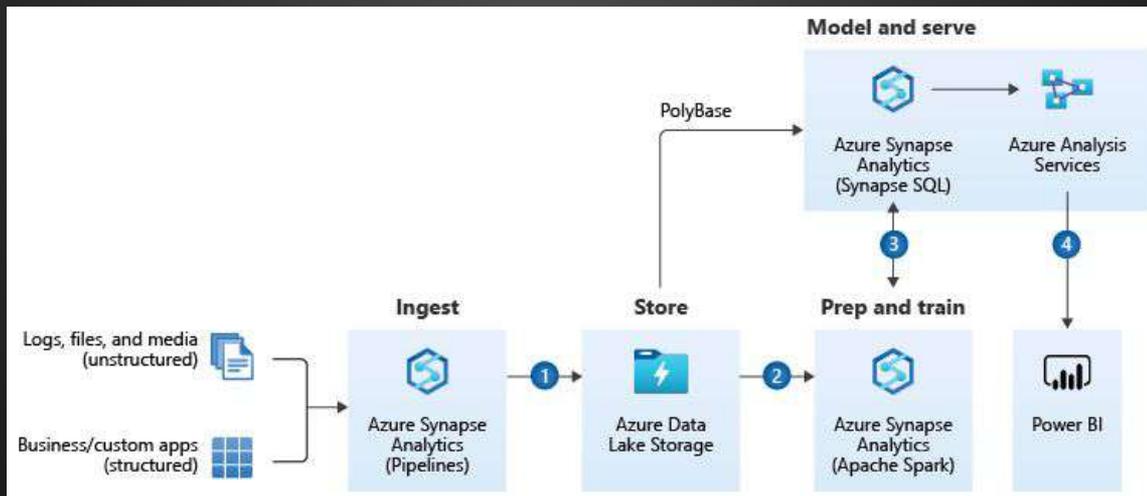
サービスを「動いたではなく動かす」側に回ることができる

→夜も安心して眠れるようになる



今後の拡張案

分散トレーシングだけでなく、サービスに関連するデータを収集・分析するための分析基盤を構築し、サービスの成長を支えるプラットフォームへ拡張したい。



出典: [エンタープライズ データ ウェアハウスのアーキテクチャ](#)

まとめ

- 分散トレーシングは面白い
 - ログを見るだけでは難しい、サービス間の通信などをトレース・分析することが可能
 - サービスで起きていること全てが可視化され、分析できるようになることで、いろいろと見えてくることも多い
- クラウドベンダが用意しているサービスを利用して手軽に始めることができる
 - 今後のOpenTelemetryの発展も注目
- 守りの監視から攻める監視へ
 - サービスの成長や負荷を検知・把握、次の一手を先んじて打てるようになる

ご静聴ありがとうございました



QUANDO