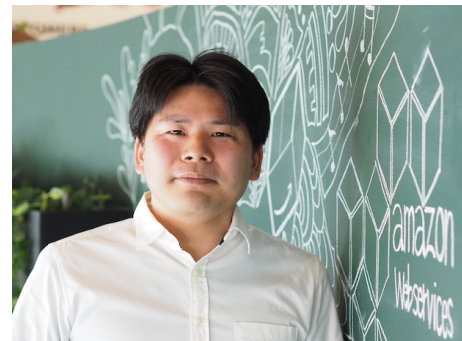


# AWSを利用してIoTビジネスを quickに立ち上げる

# 自己紹介

## 小梁川 貴史(こやながわ たかし)



- パートナーソリューションアーキテクト
  - APNさまへの技術支援とくにIoT向け
    - アーキテクチャの検討支援やレビューなど
- 経歴
  - 電機メーカーにて、自社Webサービスの設計から運用まで経験
    - AWSのユーザとして4年半、開発・運用を経験
- 好きなサービス
  - AWS IoT
  - Amazon Kinesis
  - AWS Lambda

# Amazonのノベーション

## Amazon Robotics



<https://www.youtube.com/watch?v=FQn6aFQwBQU>



Amazon  
Echo



Amazon Dash button

## Amazon PrimeAir





初期投資不要



完全従量課金



柔軟なキャパシティ



市場投入スピード



セキュア



グローバル展開

柔軟なインフラリソースを、  
いつでも、誰でも、安く、手に入れることができる

# 16のリージョン

1. US EAST (Virginia)
2. US EAST2 (Ohio)
3. US WEST (N. California)
4. US WEST 2 (Oregon)
5. EU WEST (Ireland)
6. Canada
7. **JAPAN (Tokyo)**
8. South America (Sao Paulo)
9. ASP 1 (Singapore)
10. ASP 2 (Sydney)
11. GovCloud
12. BJS 1 (Beijing China) limited
13. **EU (Frankfurt)**
14. **London**
15. **Seoul(korea)**
16. **Mubai(India)**



**16** Regions (地域) | **47** Availability Zones(データセンター群)

**72**か所のエッジロケーション

# 2018年 大阪ローカルリージョン開設予定

AWSのローカルリージョンは、旧来の単一データセンターのインフラ設計とは全く異なる、**耐障害性の高い単一**のデータセンター

## 📦 ローカルリージョンに適した用途

- 国内DR対策のバックアップ先
- 日本国内で3-AZが必須なお客様の3-AZ目

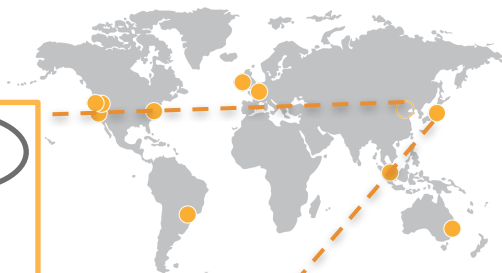
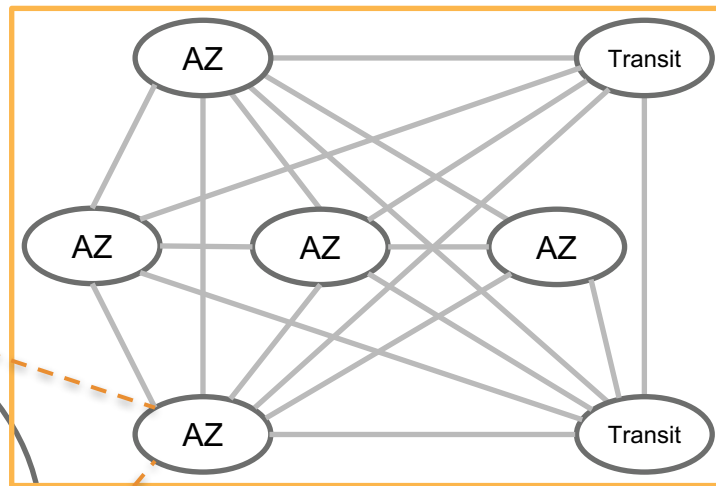
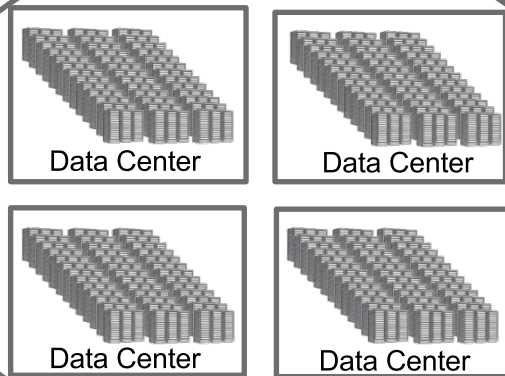
以下のケースの場合は、従来通り海外のregionをご検討頂くほうがよい

- 📦 海外regionへのデータ転送/保管に抵抗がない
- 📦 Tokyo regionでの重大障害が発生した場合でもサービス継続が必要

# 世界中に広がるAWSの拠点

AZ間はAWSの専用線で接続  
AZ間はほぼ1ms以下のレイテンシー

1データセンターあたり5万台以上のサーバ  
データセンター間はAWSの専用線で接続



# 世界中に広がるAWSの拠点

**収集されるデータのロケーションはお客様が選択**

**データは配置されたリージョンから許可なく他  
リージョンへコピーされない**

**むしろ、DR目的であえて自動でコピーする設定を  
行うことも可能**

1デ  
デ



## Enterprise Applications



Virtual Desktop



Sharing & Collaboration

### Analytics



Hadoop



Real-time

### App Services



Queuing & Notifications



Workflow

### Deployment & Management



One-click web app deployment

### Mobile Services



Identity



Sync

サーバー、ストレージ、DBから、アプリケーションまで  
**90**を超えるクラウドサービスを提供



Data Pipelines



Search



Resource Templates



Push Notifications

## Administration & Security



Identity Management



Access Control



Usage Auditing



Key Storage



Monitoring And Logs

## Core Services



Compute  
(VMs, Auto-scaling and Load Balancing)



Storage  
(Object, Block and Archival)



CDN



Databases  
(Relational, NoSQL, Caching)



Networking  
(VPC, DX, DNS)

## Infrastructure



Regions



Availability Zones



Points of Presence

## 本日のKey word

- Serverless
- IoT
- Security

**Quickな環境構築のためにマネージド・サービスを積極的に利用する**

# Quickに??

- インフラの構築
- プログラム開発時間
- 運用後のメンテナンスの考慮

# オンプレミス、EC2+ミドルウェア、マネージドサービスの比較

アプリからの利用
スケーラビリティ
可用性
バックアップ
ミドルウェアのパッチ
ミドルウェアの導入
OSのパッチ
OSの導入
サーバメンテナンス
ラック導入管理
電源、ネットワーク

オンプレミス

アプリからの利用
スケーラビリティ
可用性
バックアップ
ミドルウェアのパッチ
ミドルウェアの導入
OSのパッチ
OSの導入
サーバメンテナンス
ラック導入管理
電源、ネットワーク

ミドルウェアon EC2

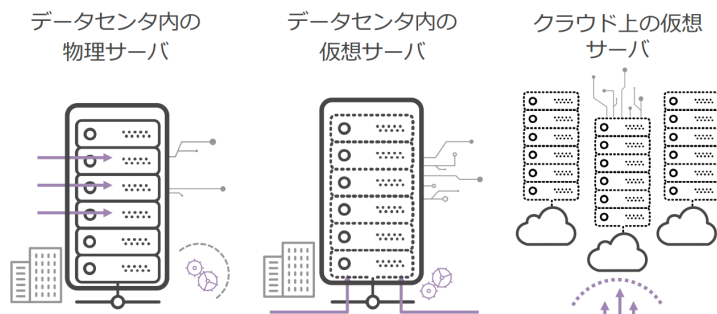
アプリからの利用
スケーラビリティ
可用性
バックアップ
ミドルウェアのパッチ
ミドルウェアの導入
OSのパッチ
OSの導入
サーバメンテナンス
ラック導入管理
電源、ネットワーク

AWS IoT, RDS等  
マネージドサービス

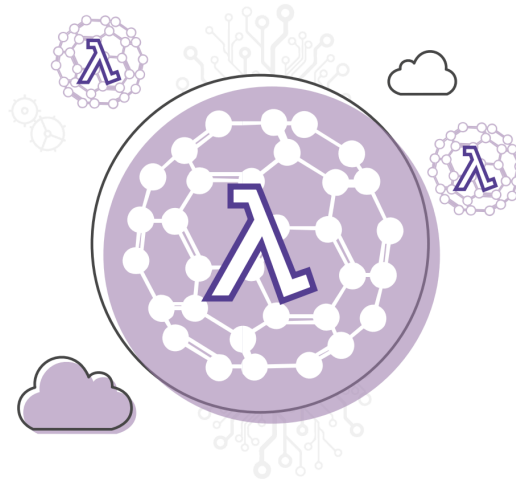
# サーバレス

# サーバレス

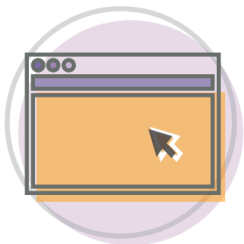
サーバについて考慮すべきことが減り、アプリケーションをビルド実行していくこと。



## サーバレス

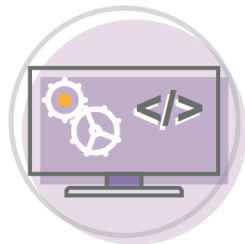


# サーバレスユースケース



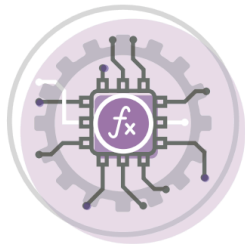
## Web アプリケーション

- 静的Webサイト
- 動的Webアプリ
- FlaskとExpressのためのパッケージ



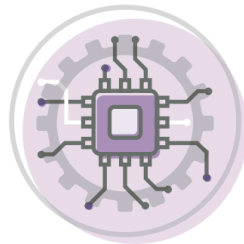
## Backends

- アプリとサービス
- モバイル
- IoT



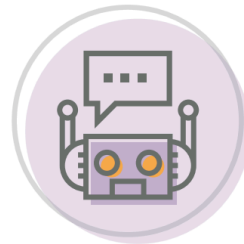
## Big Data

- MapReduce
- バッチ



## Media & Log Processing

- リアルタイムデータ
- ストリーミングデータ



## Chatbots

- チャットボットのロジック
- Amazon Echo用 Alexa Skill



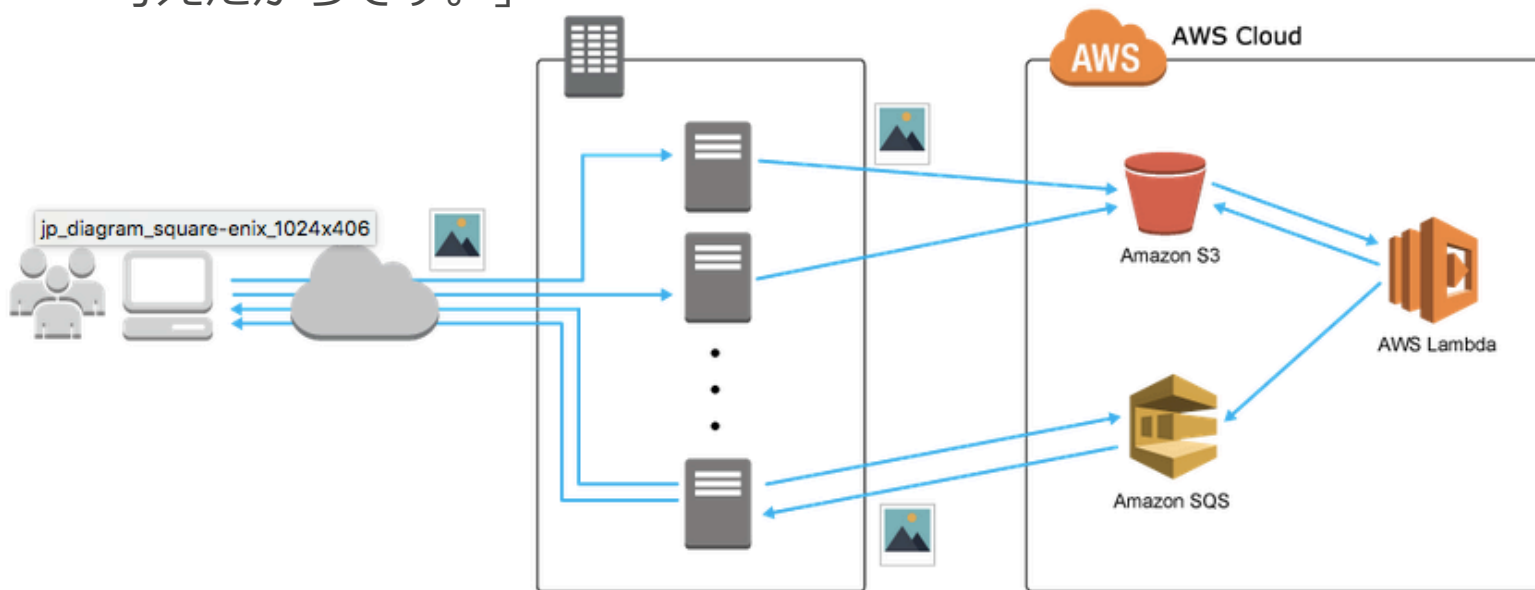
# サーバレスの使い所

- Restで定義できる
- トランザクションが不要
- 非同期処理が許容できる
- 最大5分の実行時間制約が許容できる

従来のアーキテクチャの考え方とは違う考え方  
既存から移行する際に、現行アーキテクチャに引きずられず  
にサーバレスで考え直すことも重要

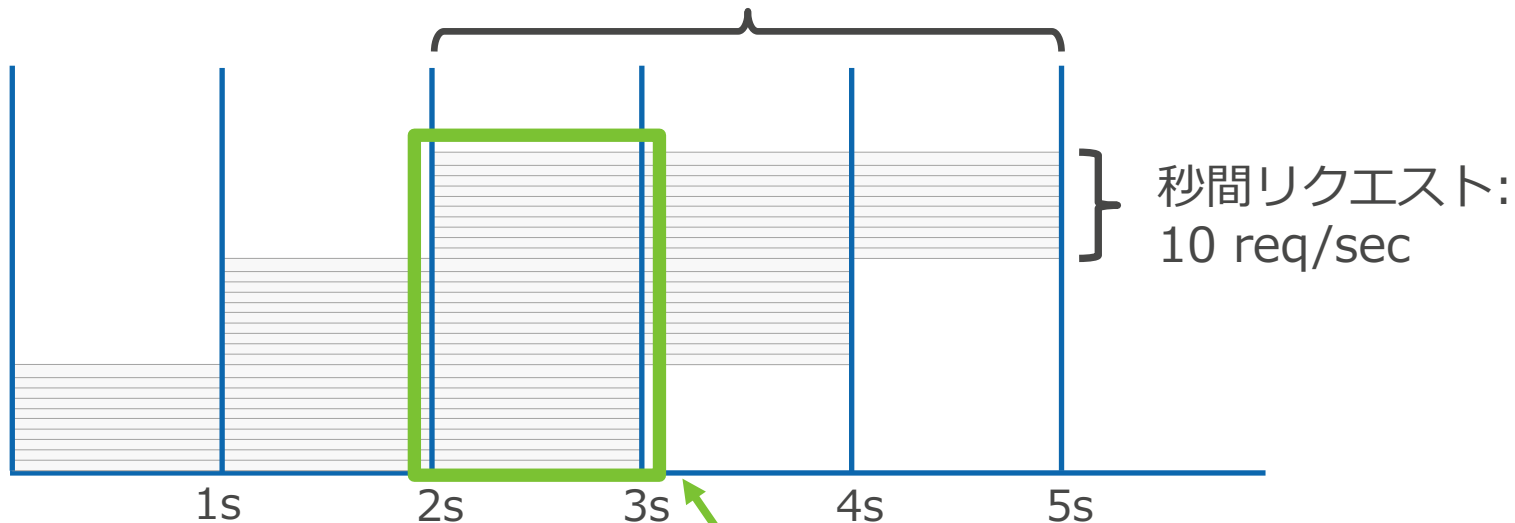
# 直列処理から並列処理へ

画像処理を高速化するため、最初は 64 コア程度のスペックのサーバーを新たに用意し、処理するスレッド数を増やすことが検討されました。「1回の処理速度を速くするよりも、処理の並列度を上げるほうが効果的だと考えたからです。」



# 同時実行数

関数の平均実行時間: 3s / exec

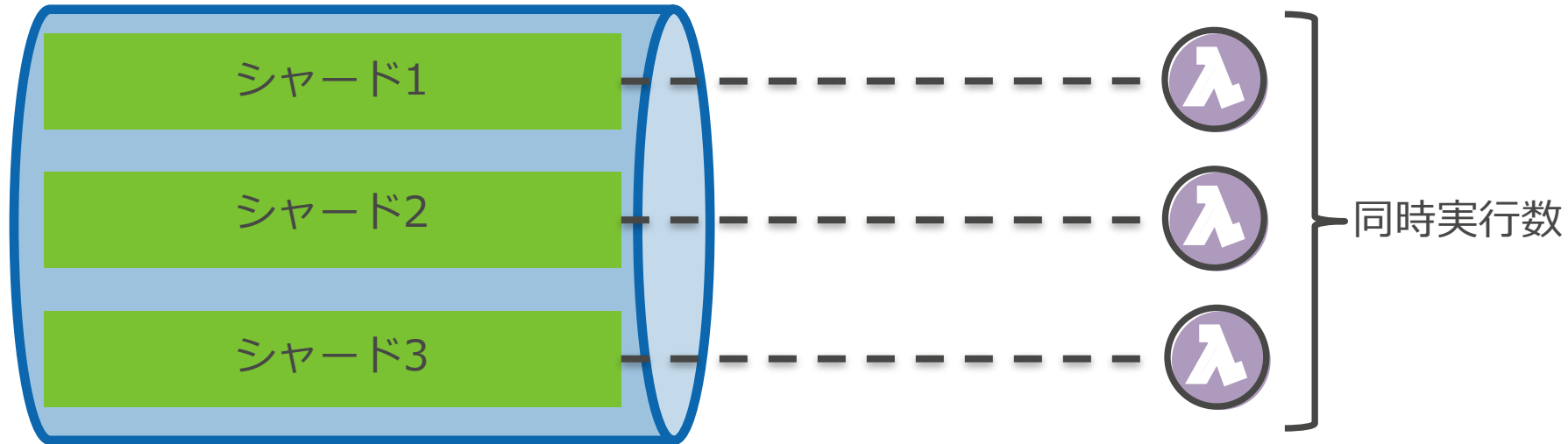


同時実行数

= “同時”に実行されているタイミング

# 同時実行数 - ストリームベースの場合

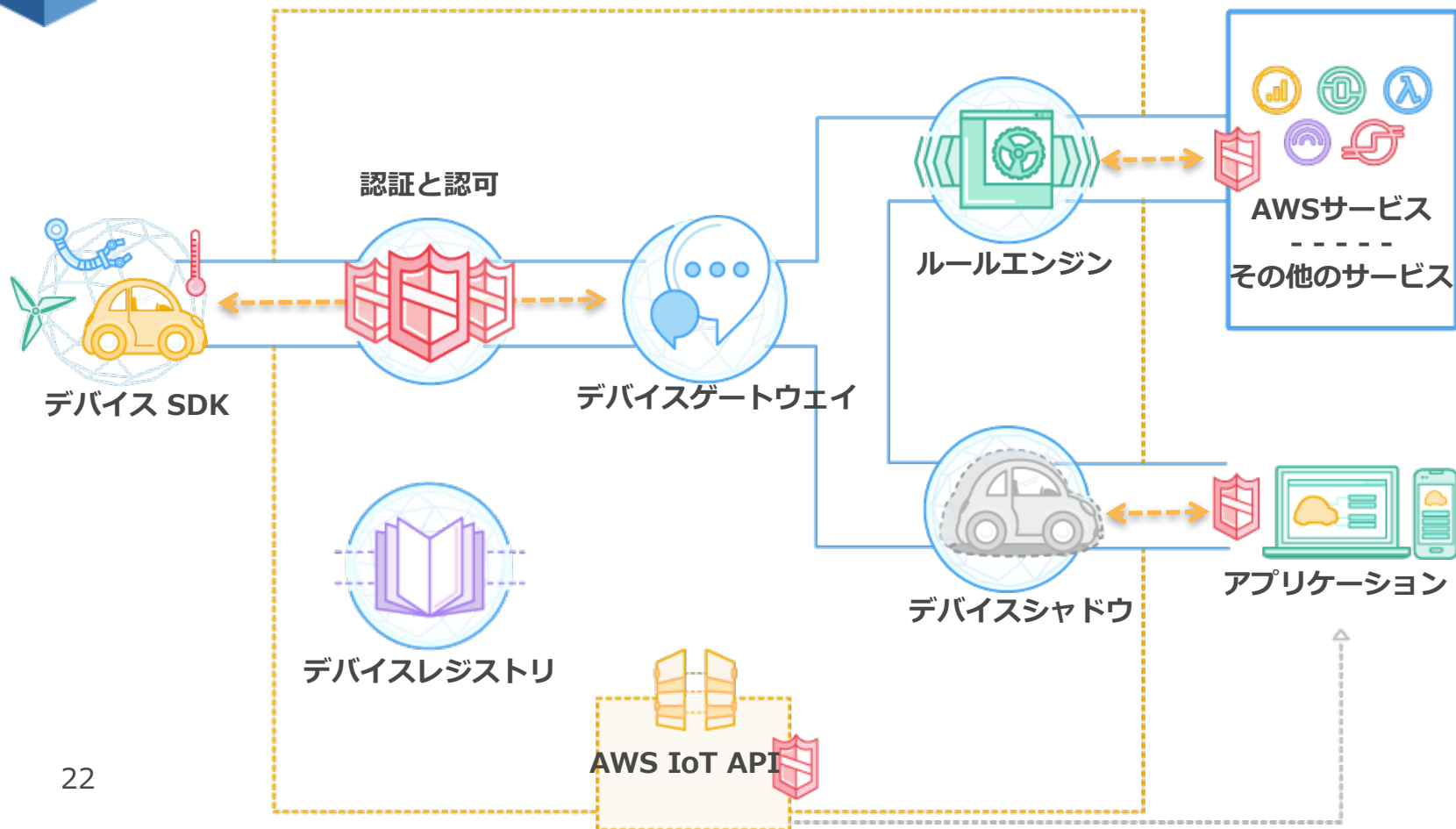
ストリーム



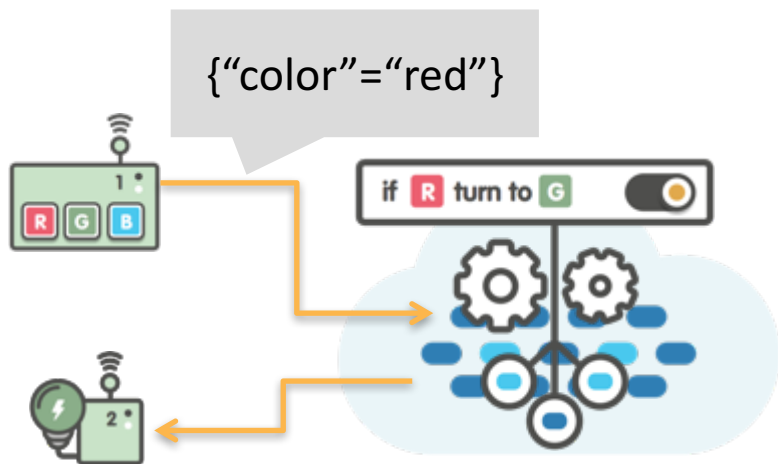
# IoT



# AWS IoT



# ルールエンジン



## ■ シンプル&慣れた構文

- ・ SQL文を使ったトピックのフィルタ
- ・ オプションのWHERE句で条件を記述することが可能
- ・ JSONサポート

## ■ メッセージ変換機能

- ・ 文字列操作 (正規表現サポート)
- ・ 算術計算
- ・ コンテキストベースのヘルパー
- ・ 暗号
- ・ UUID, Timestamp, 乱数など.

SELECT **Data** FROM **topic** WHERE **条件**

```
SELECT * FROM 'things/thing-2/color'  
WHERE color = 'red'
```



# 便利な機能がいろいろ



ルールエンジンと  
アクション

SQLライクな構文でルールの設定  
ルールに合致した場合、AWSの各種サー  
ビスとのインテグレーション



デバイスシャドウ

デバイスが物理的に接続されてなくて  
もコマンドを伝達できる



デバイスとクラウドとの  
相互認証

TLS1.2を用いた相互認証  
証明書に対するポリシー設定も可能

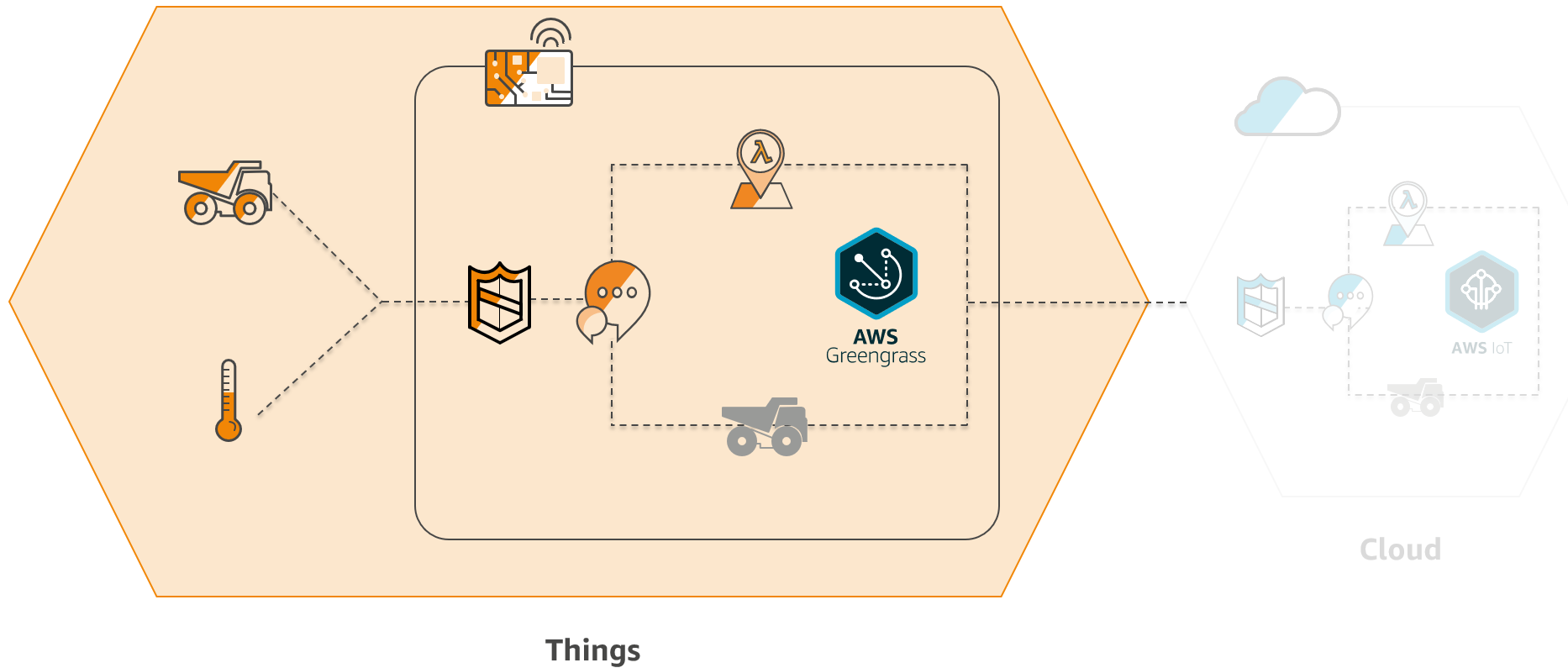


デバイスレジストリ

多くのデバイスを管理  
Key-Value形式でファームバージョンなど  
の属性情報も管理可能



# Moving to the edge



# ローカルで処理することの価値



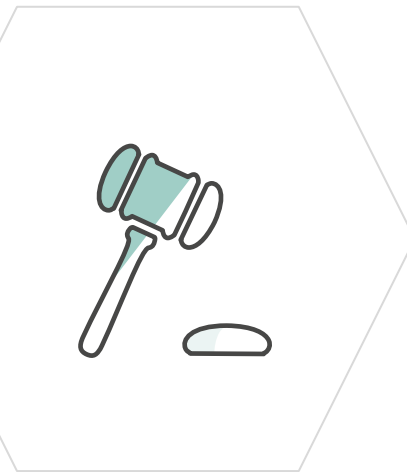
## 物理法則

- 通信速度
- Latency



## 経済の法則

- データを保管するコスト
- 回線敷設のコスト



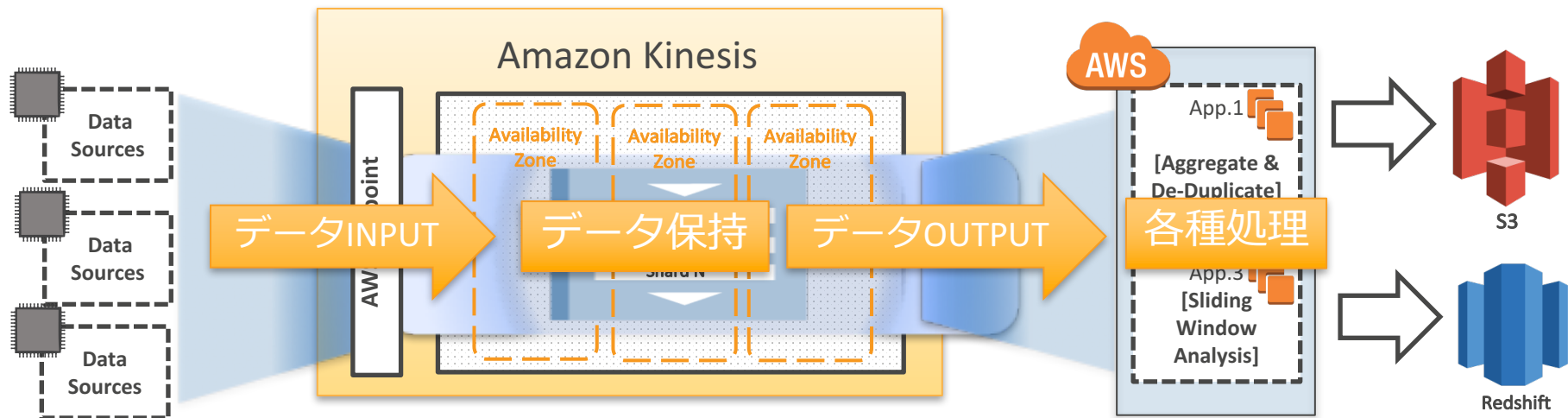
## 法律 (決まり)

- 長期間の保管



# Amazon Kinesis Streams

大量でかつ連続したセンサーデータを信頼性高く格納し、低レイテンシーでデータ処理に伝送が可能。

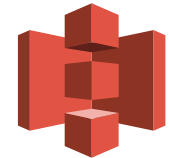


低レイテンシー

高信頼性  
(3AZ, 7days)

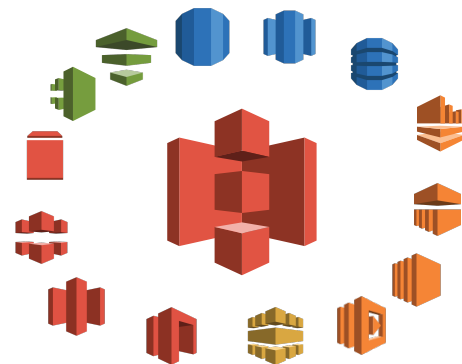
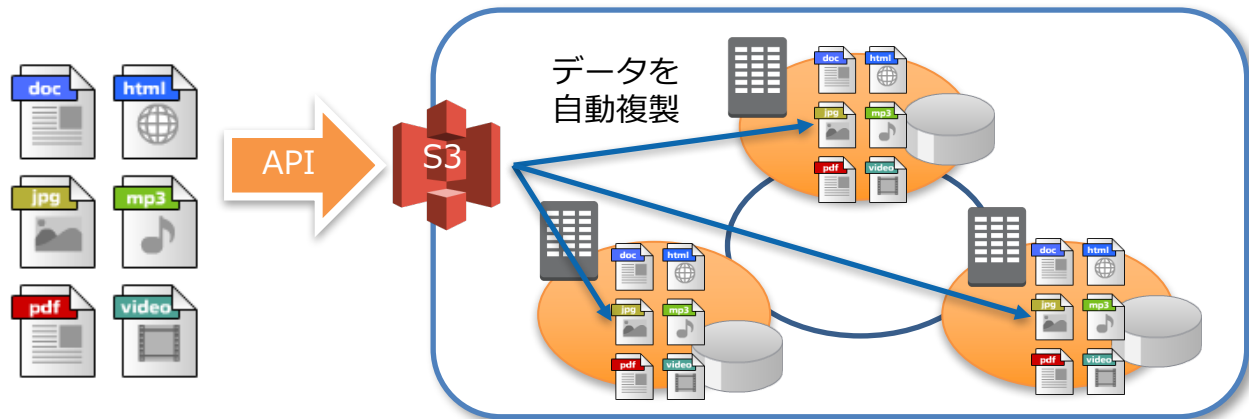
容易な  
インテグレーション

Pluggable  
アーキテクチャ



# Amazon S3 (Simple Storage Service)

センサーデータを耐久性が高く低コストなクラウドストレージに格納。  
データ処理などのデータハブに。



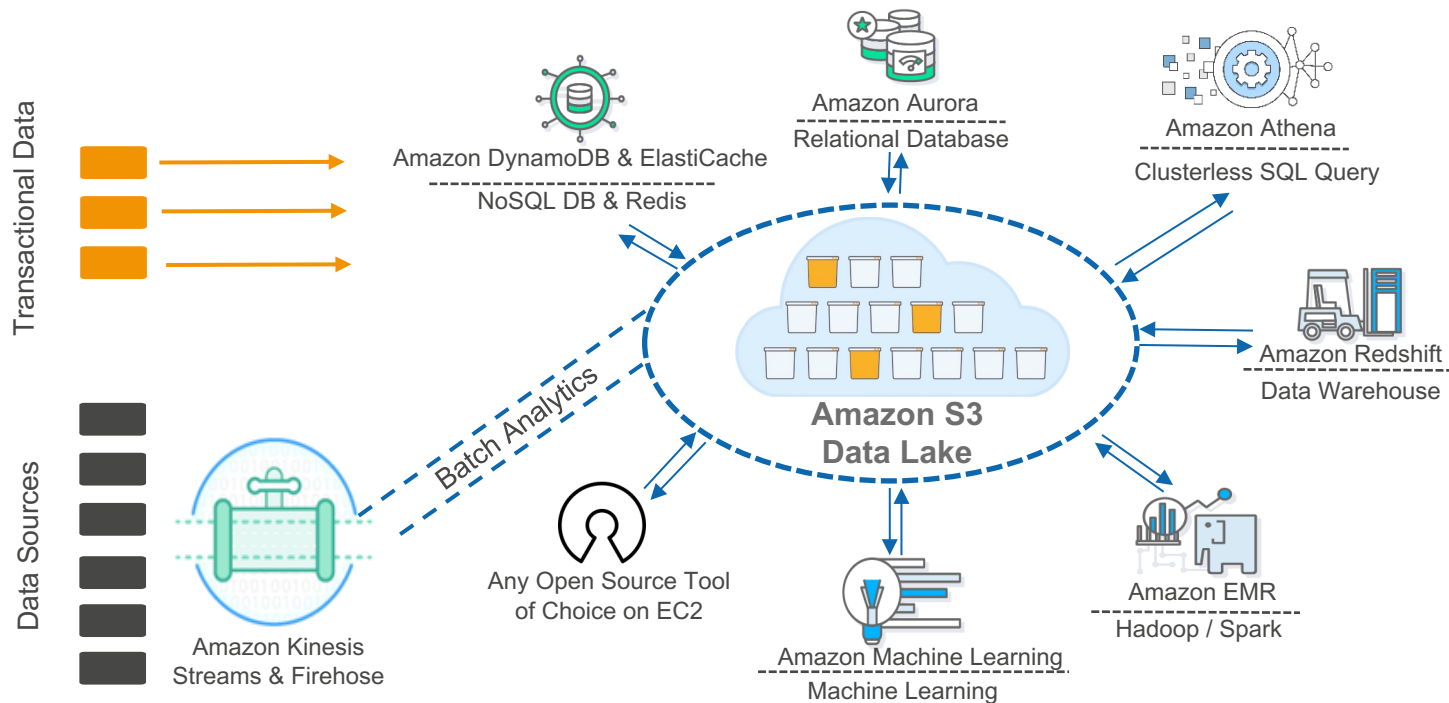
- データを3つ以上のデータセンターに自動複製
- 設計上のデータ耐久性は、99.999999999%
- 1GByteあたり月額 \$0.033 - 容量は無制限

# データ送受信(Ingest)



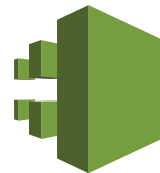
	AWS IoT	Amazon Kinesis	Amazon S3
ペイロード長	1メッセージあたり128KB	1レコードあたり1MB	1オブジェクトあたり5TB
プロトコル	MQTT/HTTPS /Websocket	HTTPS	HTTPS
料金	512Byteを1メッセージとして \$8/百万メッセージ	1シャード\$0.0195/時 ペイロードユニットを25KB として\$0.0215/百万ペイ ロード(Streams)	\$0.025/GB(最初の50TB, 月)
認証	クライアント証明書 SigV4	SigV4	SigV4
使いどころ	ペイロードが小さく送信頻度も 低い デバイスとクラウド間の双方向 通信が必要 高いセキュリティを求められる	ペイロードが大きく、送信 頻度が高い	メディアなどデータサイズ が大きい場合、ファイル単 位でデータを扱う場合

# Amazon S3を中心としたデータレイクを構成



# セキュリティ

# AWS CloudTrail



## AWS上のAPI操作を記録するサービス



### 特徴 [\(http://aws.amazon.com/jp/cloudtrail/\)](http://aws.amazon.com/jp/cloudtrail/)

- AWSマネジメントコンソール、コマンドライン、サードパーティ製品等AWS APIの呼び出しを記録
- セキュリティの分析、リソース変更の追跡、およびコンプライアンスの監査に利用

### 価格体系 [\(http://aws.amazon.com/jp/cloudfront/pricing/\)](http://aws.amazon.com/jp/cloudfront/pricing/)

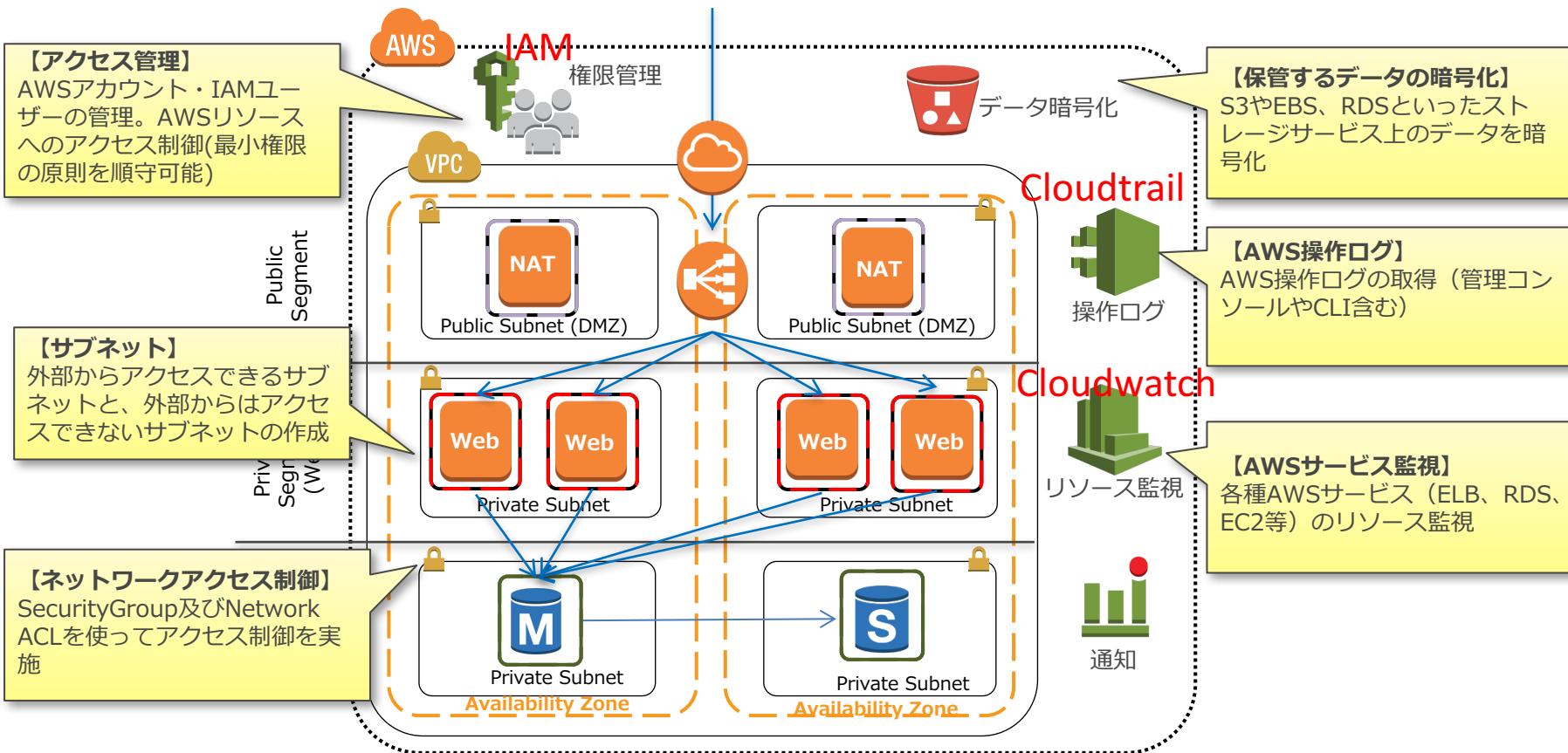
- AWS CloudTrail自体の料金は無料
- Amazon S3、Amazon SNSの標準の使用料金が必要

AWS CloudTrail サポートサービス一覧

[http://docs.aws.amazon.com/ja\\_jp/awscloudtrail/latest/userguide/w hat\\_is\\_cloud\\_trail\\_supported\\_services.html](http://docs.aws.amazon.com/ja_jp/awscloudtrail/latest/userguide/w hat_is_cloud_trail_supported_services.html)



# すべてのレイヤーでセキュリティを確保



# まとめ

マネージドサービスを利用する意味を理解

- ビジネス価値を産まない部分での利用
- アプリケーション開発 / 運用の速度を上げる/楽にするための選択
- 開発後の運用フェーズでも楽をする

