AWSおよびパートナーサービスを使った、データの集約および活用設計パターン

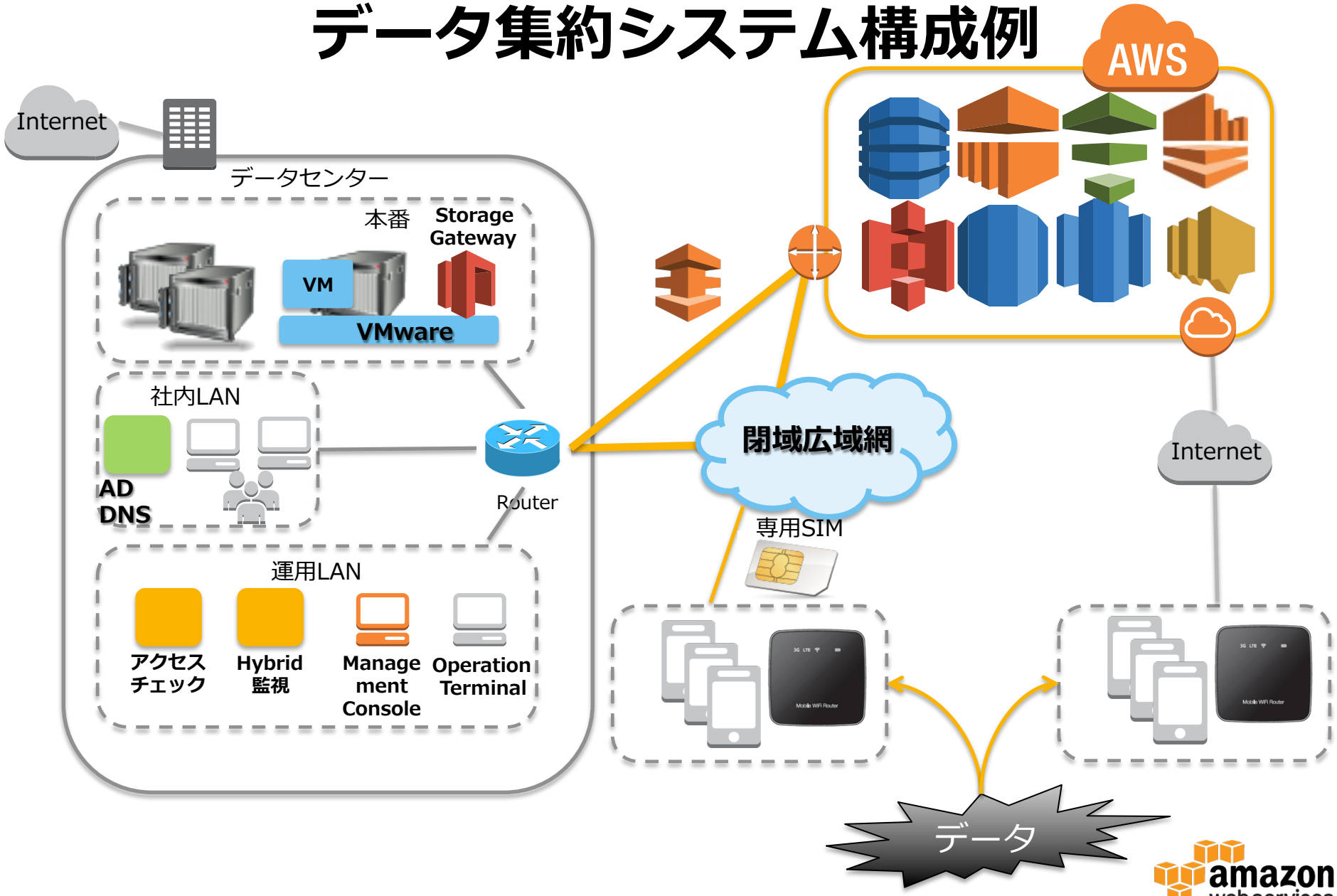
荒木靖宏
アマゾンデータサービスジャパン株式会社
プリンシパルソリューションアーキテクト

自己紹介

- 名前
  - 荒木 靖宏
- 所属
  - アマゾンデータサービスジャパン株式会社
    プリンシパルソリューションアーキテクト
- 好きなAWSサービス
  - Amazon Virtual Private Cloud
  - AWS Direct Connect

amazon web services

データ集約システム構成例

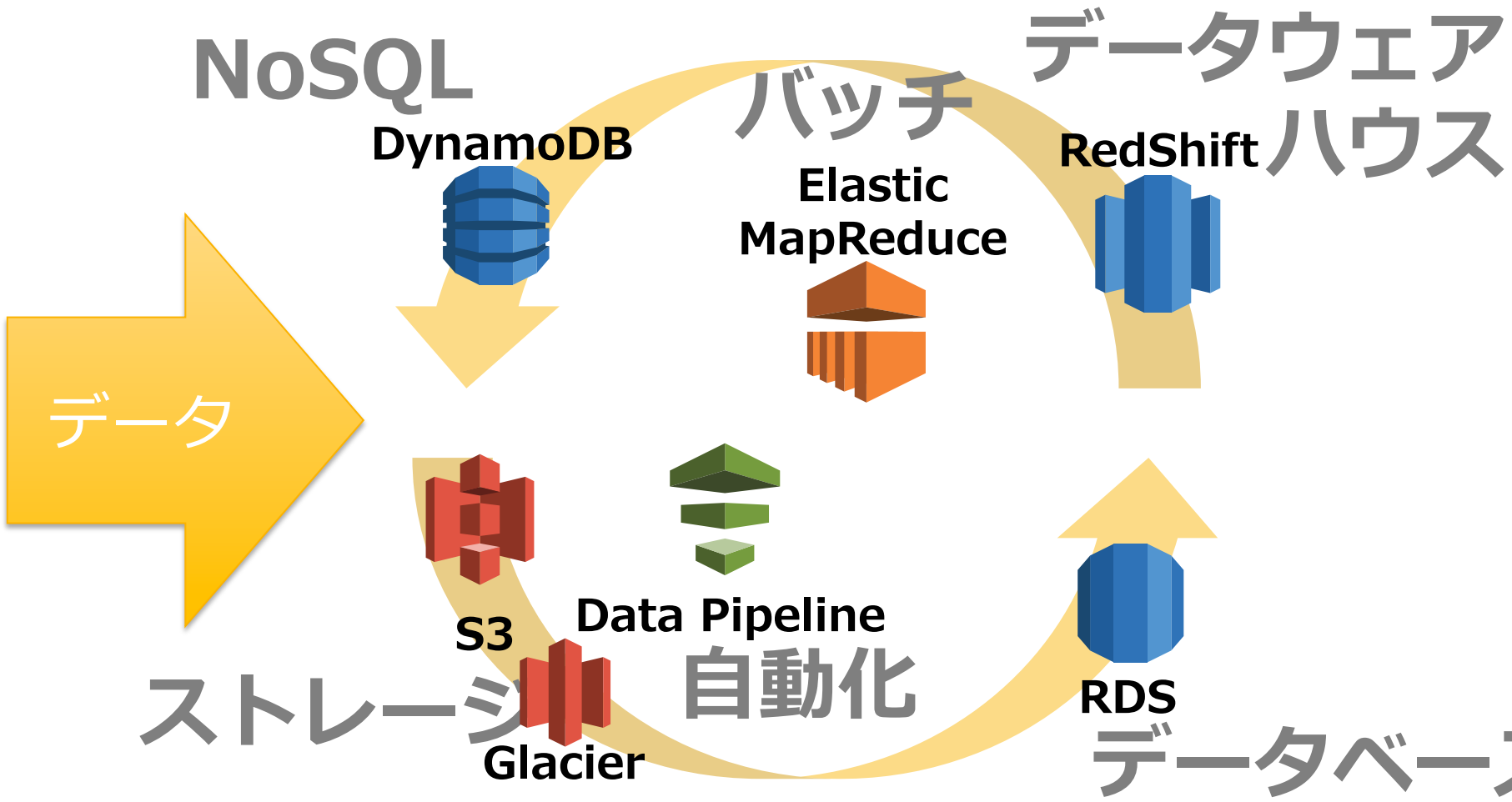


データ集約の基本方針

あらゆるデータの発生源に対応する
既存のアセットを最大限に活かす
AWSのメリットを享受する

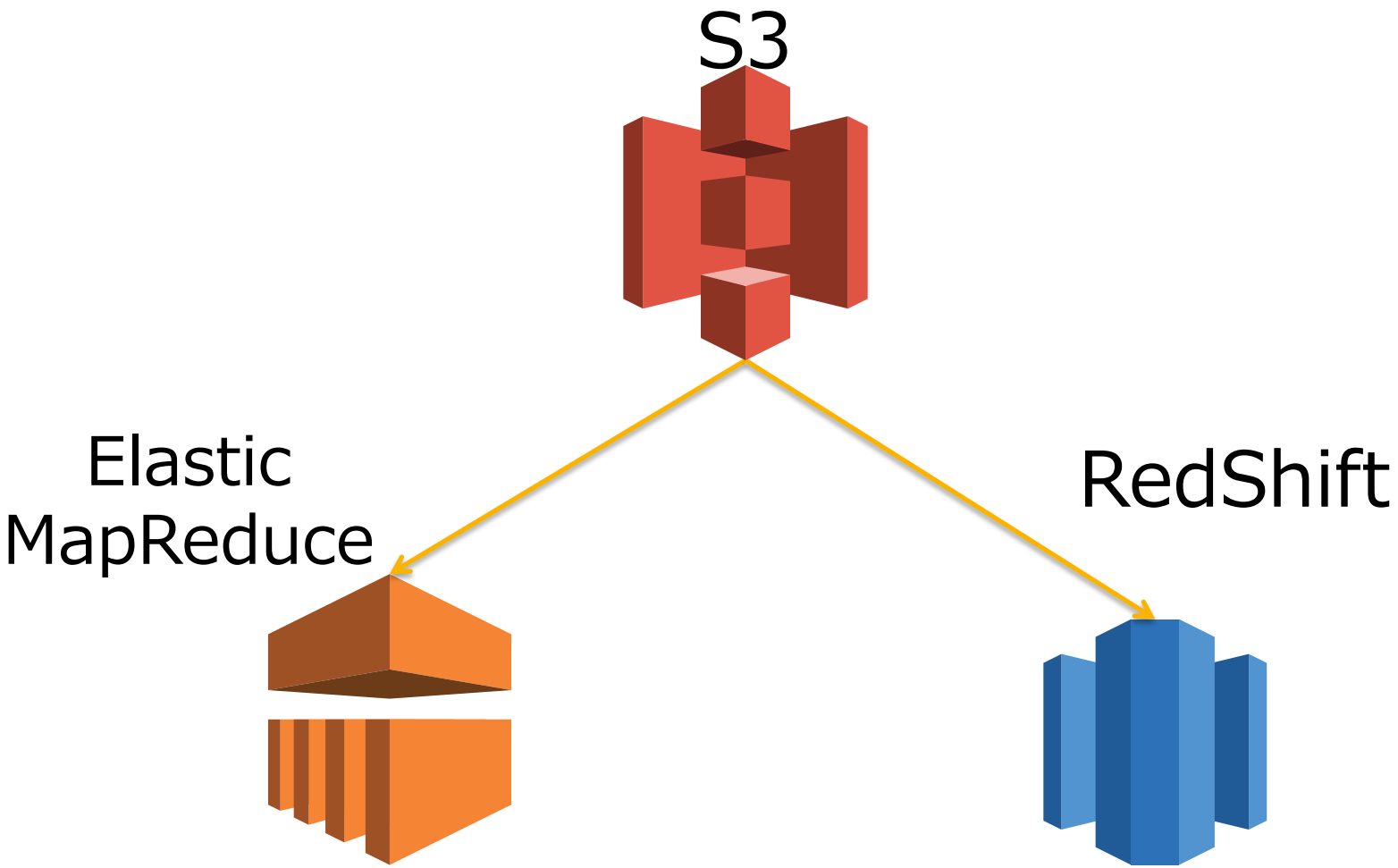
AWSのデータ関連サービス

AWSのデータ関連サービス

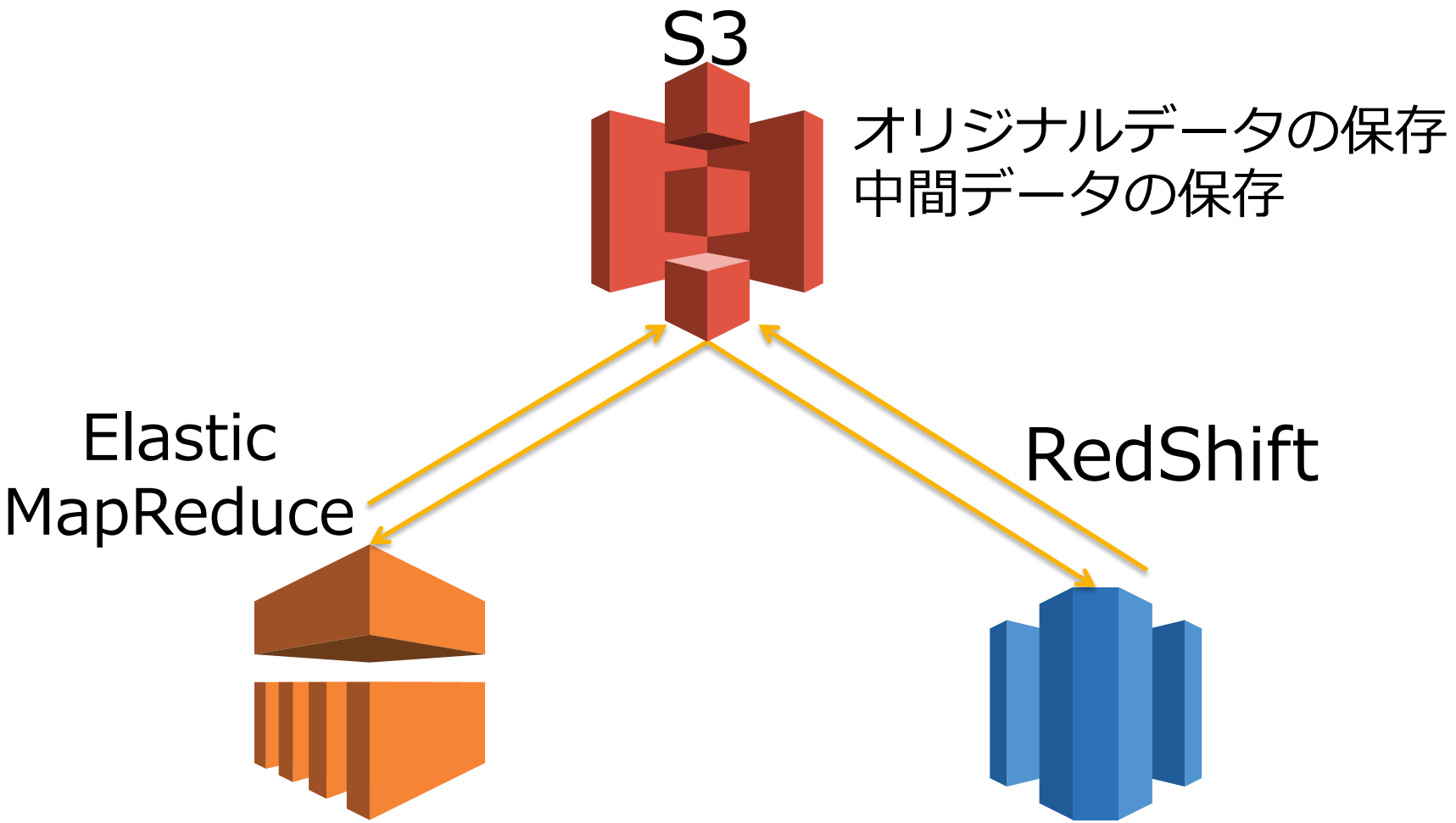


amazon web services

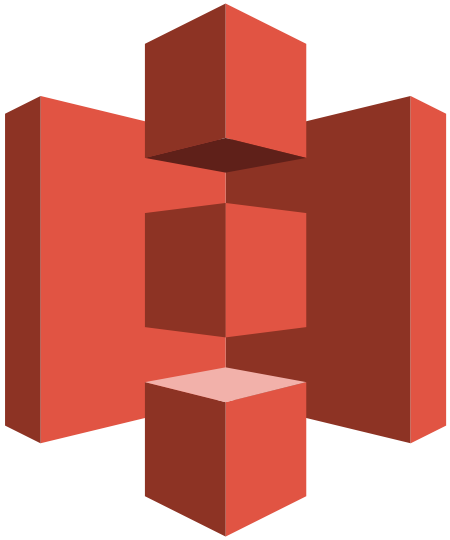
データ処理の主要3サービス



データ処理の主要3サービス



S3

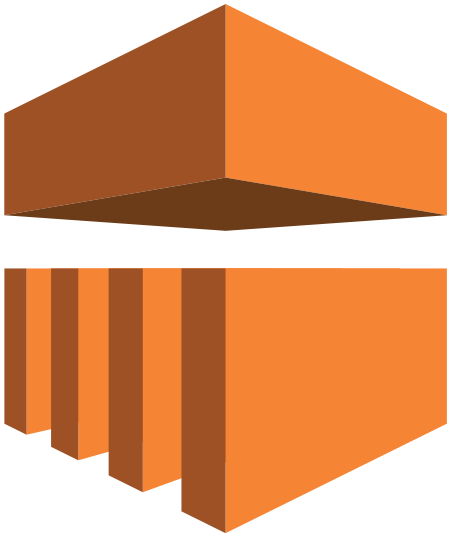


- 非常に耐久性の高い(99.999999999%)のオンラインストレージサービス
- 格納容量に制限がない
- ストレージ用サーバやディスクの運用から解放される
- 1GB約10円程度からの従量課金により低コストにデータを保存
- アクセスコントロール、暗号化などセキュリティ機能も万全

ストレージ

amazon web services

Elastic MapReduce(EMR)

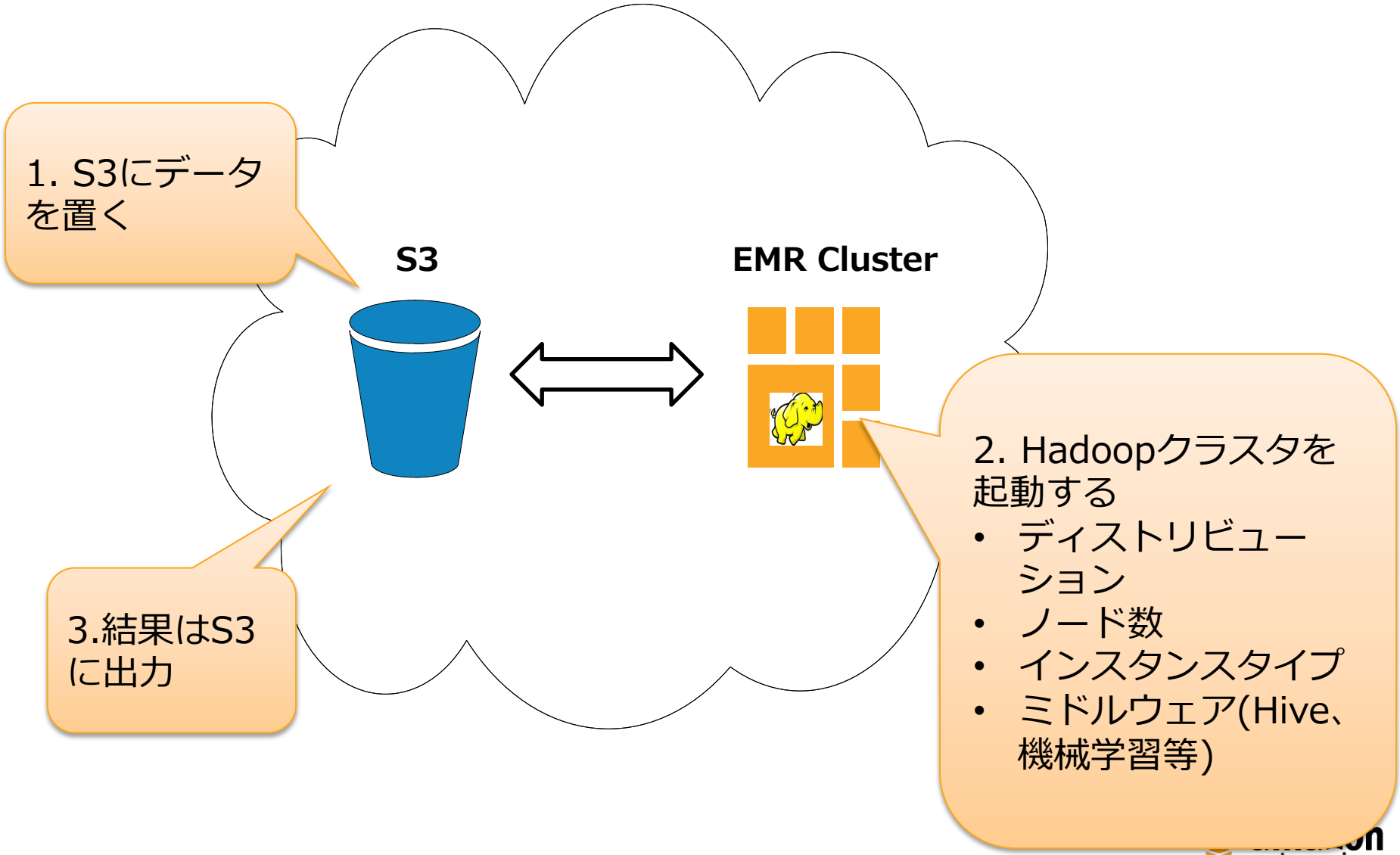


- Hadoopクラスタのサービス
- 大量のバッチ計算処理をサーバを大量に並べて短時間で解決する
- 1時間単位の従量課金
  - 使い終わったら課金は停止する
  - 勿論ずっと稼働も可能

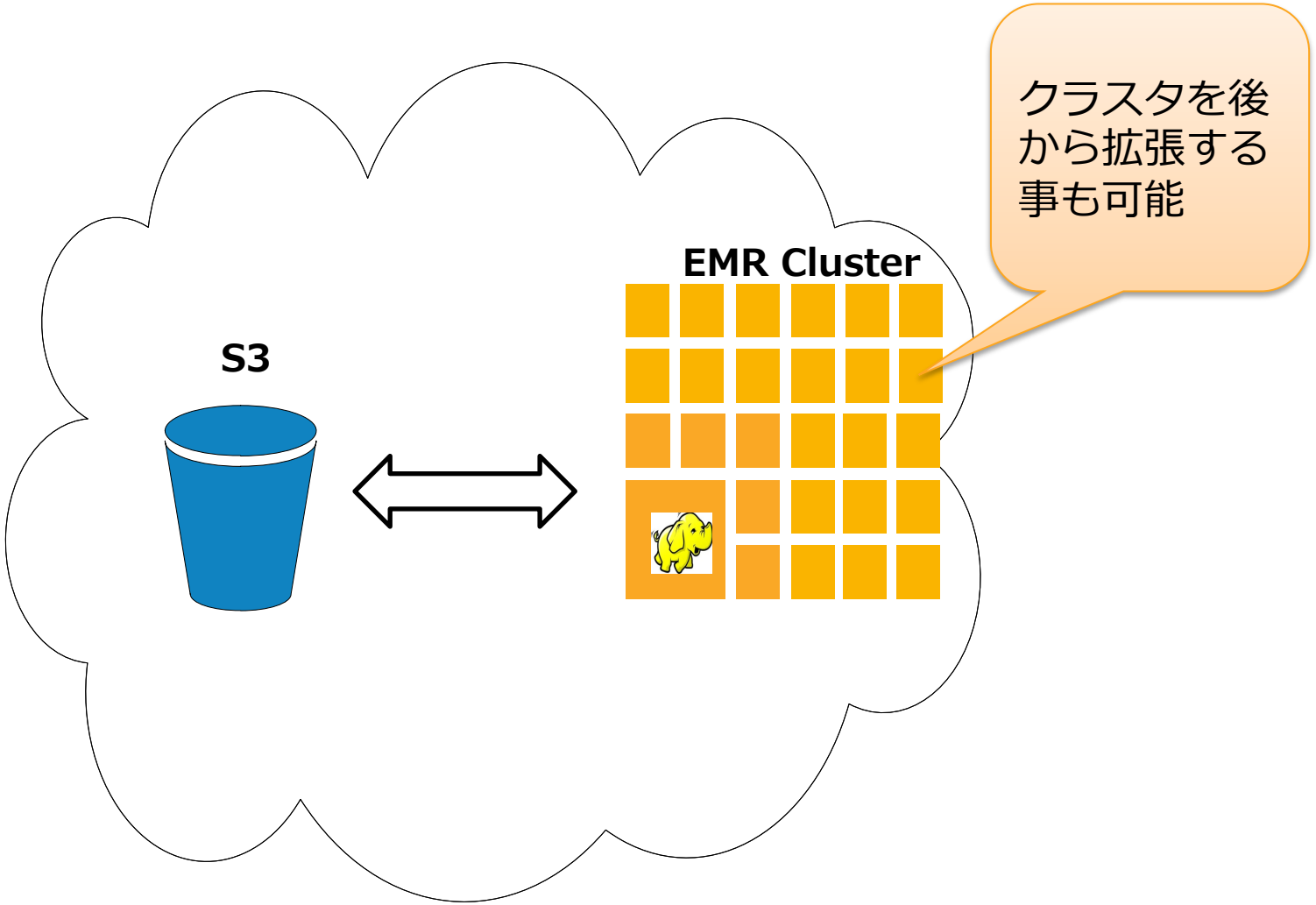
- 組み合わせ爆発、データ変換など利用例は多数

バッチ

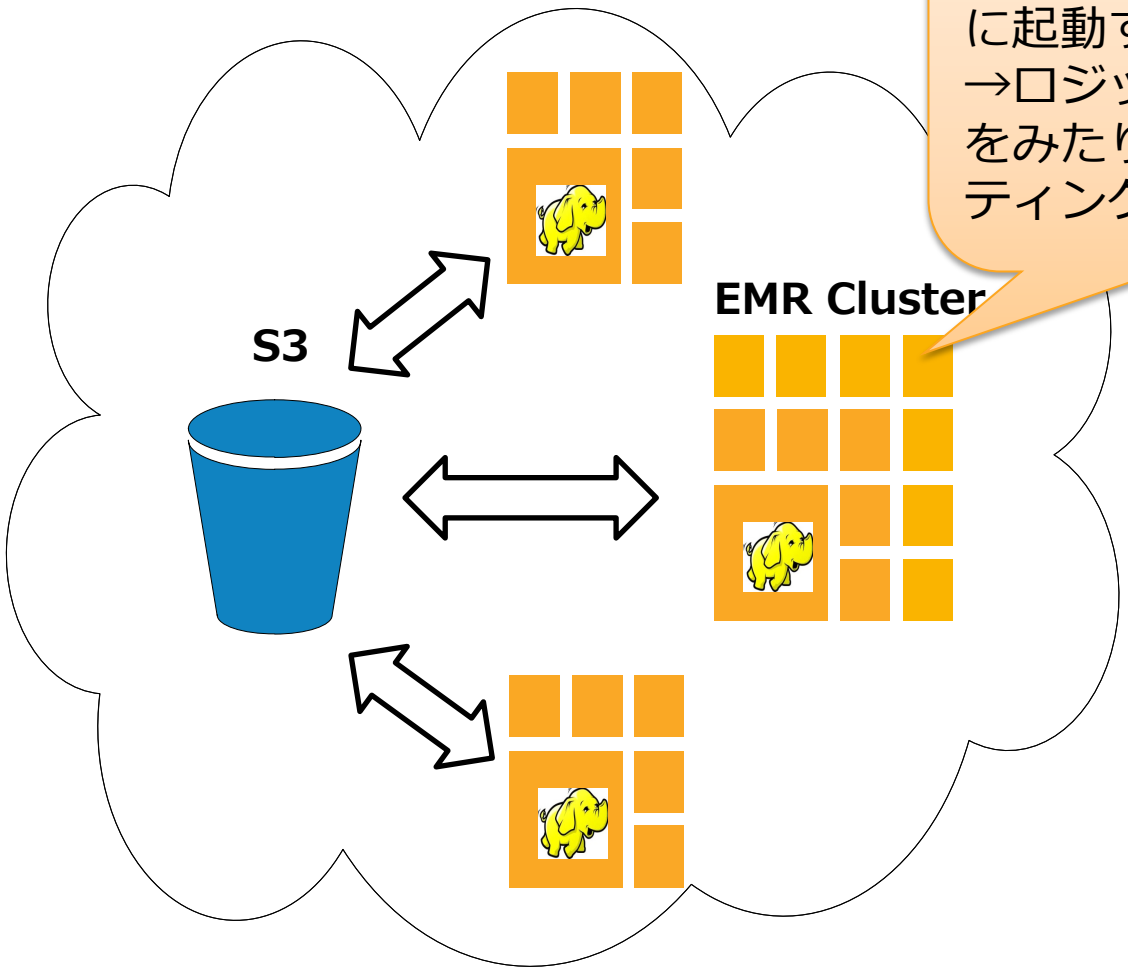
EMRの仕組み


1. S3にデータを置く
S3
EMR Cluster
2. Hadoopクラスタを起動する
• ディストリビューション
• ノード数
• インスタンスタイプ
• ミドルウェア(Hive、機械学習等)
3.結果はS3に出力

EMRの仕組み(2)

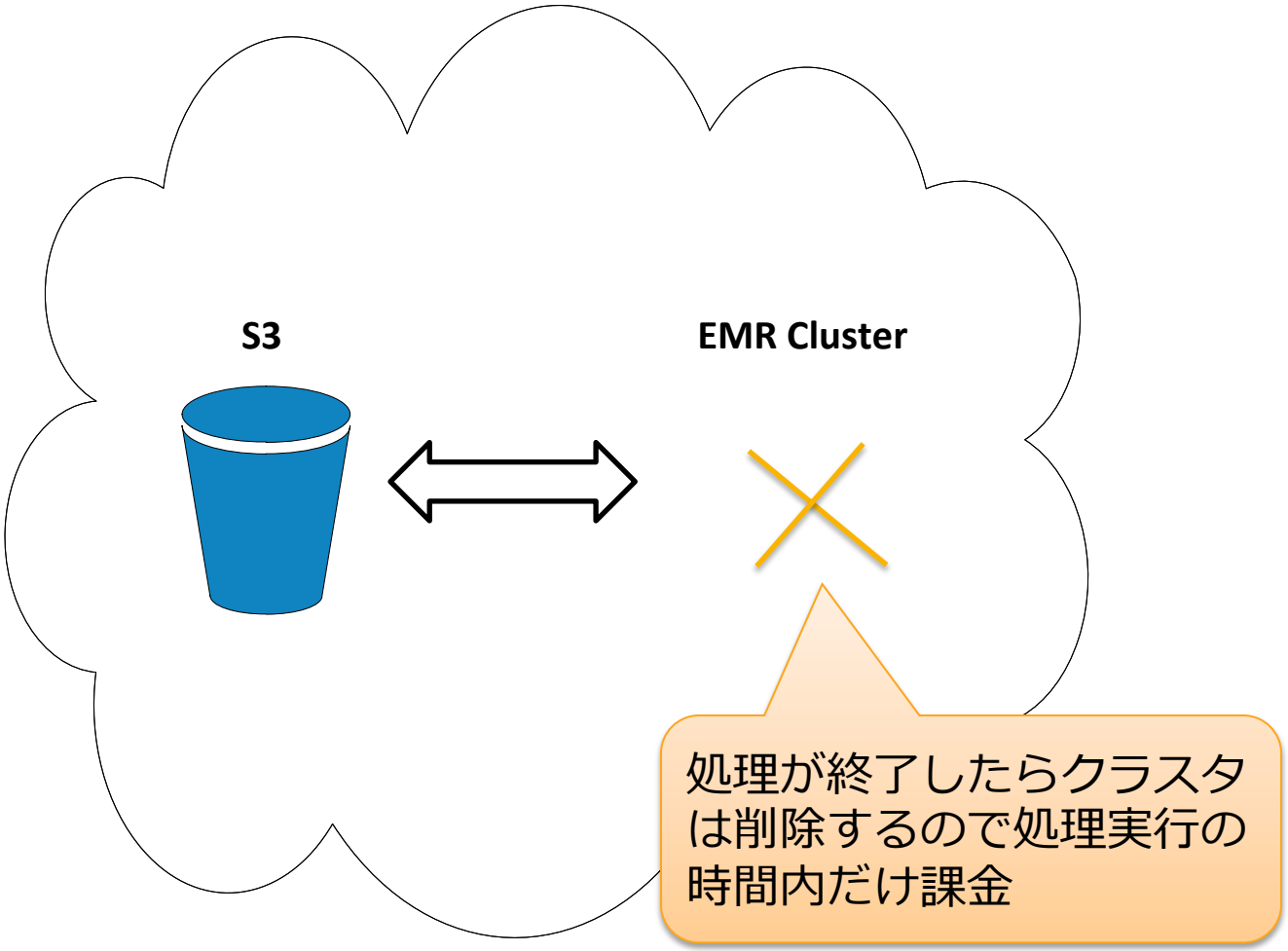


EMRの仕組み(3)



同じデータに対して、複数のクラスタを並列に起動する事が可能
→ロジック変更の影響をみたり、A/Bテスティングが容易

EMRの仕組み(4)



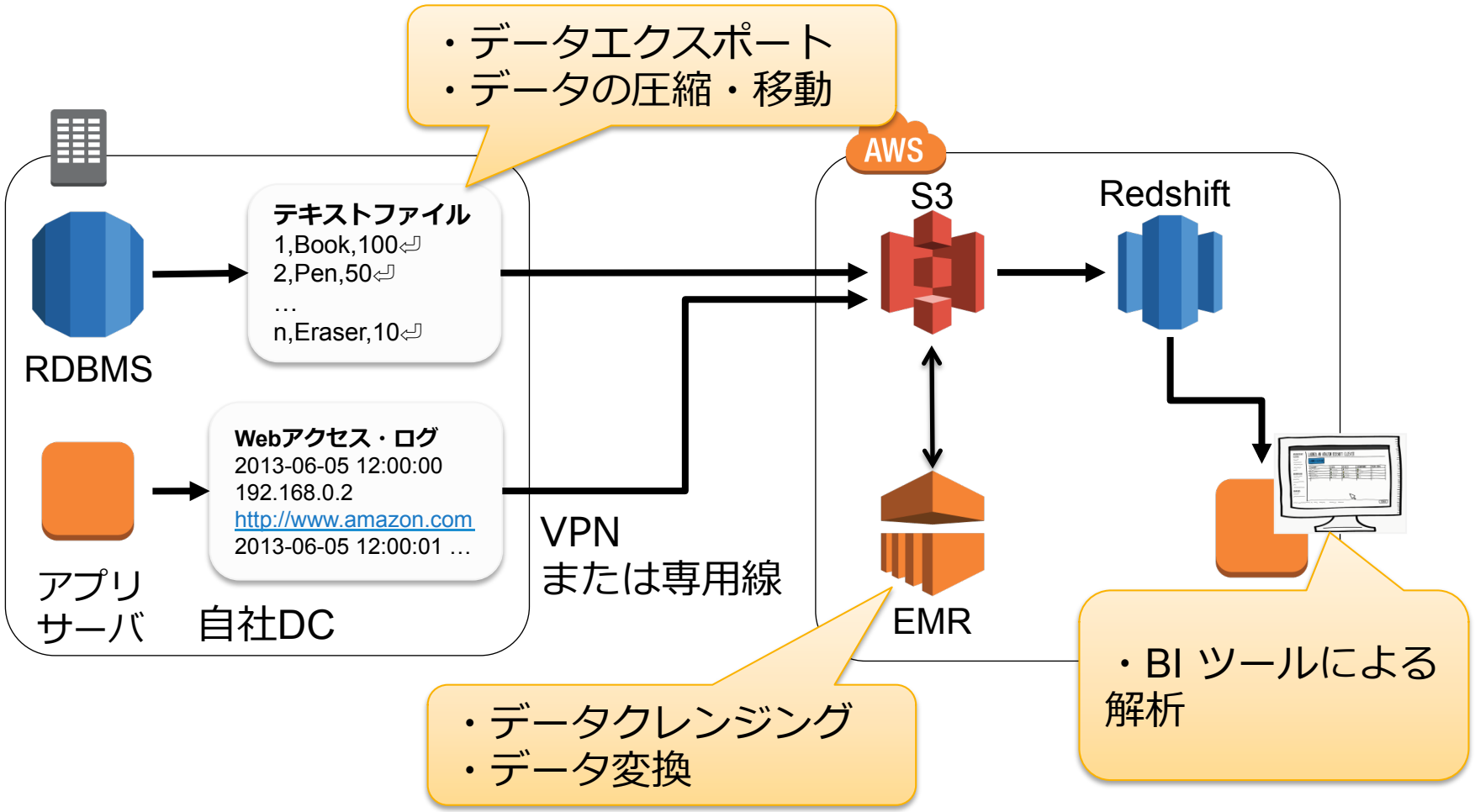
RedShift

- データウェアハウスサービス
- 運用管理の手間がほぼない
- 初期費用ゼロ、低価格
- SQLが使えるため、既存ツールなどの連携も容易
- S3連携が可能で、データのロード・アンロードが容易
- ペタバイトまでもスケール可能

データウェアハウス

amazon web services

データ処理の流れ



amazon web services

RedShiftへのデータロード

- データロードツールも充実

FlyData

ATTUNITY CloudBeam

DataSpider® Servista

talend*
*open data solutions

JDBC/ODBC
S3からのLoad

Amazon Redshift

amazon web services

BIツールとの連携

- 既存のBIツールとの連携が可能

Amazon Redshift

JDBC/ODBC

PostgreSQL.org で
提供されるドライバで接続

ACTUATE The BIRT Company™
1ST WingArc
birst
INFORMATICA The Data Integration Company™
JASPERSOFT the intelligence inside
JReport JINFONET SOFTWARE
sas™
MicroStrategy
COGNOS®
PERVASIVE
Better Decisions Every Day™
pentaho
tableau SOFTWARE

amazon web services

ビッグ・データのソリューション

- **ビッグ・データへの一般的なアプローチ**
  - クエリー・エンジン（データ・ウェアハウス、YesSQL、NoSQL）
    - 構造化データに対して、クエリーの繰り返し実行
    - 索引、ディメンションによるクエリー性能の向上
  - バッチ・エンジン（Map-Reduce）
    - 非構造化データに対して、低い頻度でクエリー・解析の実行
- **ビッグ・データへのストリーム処理**
  - コンテンツ（データ・ストリーム）へのリアルタイムな応答
  - 比較的にシンプルなデータ処理（集約、フィルター、スライドウィンドウ等）
  - 他のデータ・ストアへの移動によるデータのライフ・サイクル

ビッグ・データのソリューション

蓄積された「過去」データの処理

「今」流れているデータの処理

amazon web services

Amazon Kinesis

フルマネージドなリアルタイムデータ処理サービス

使いやすさ

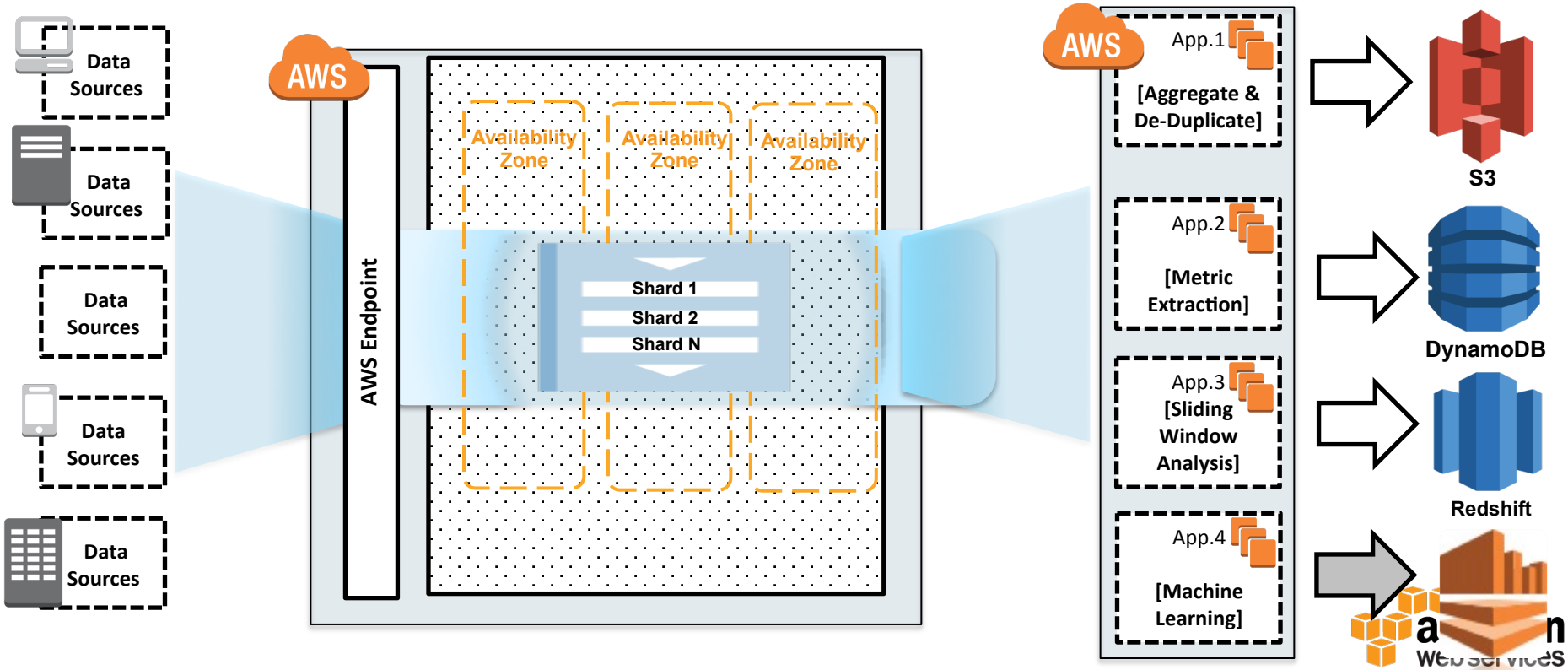
リアルタイム処理

高スループット／伸縮自在性

インテグレーション

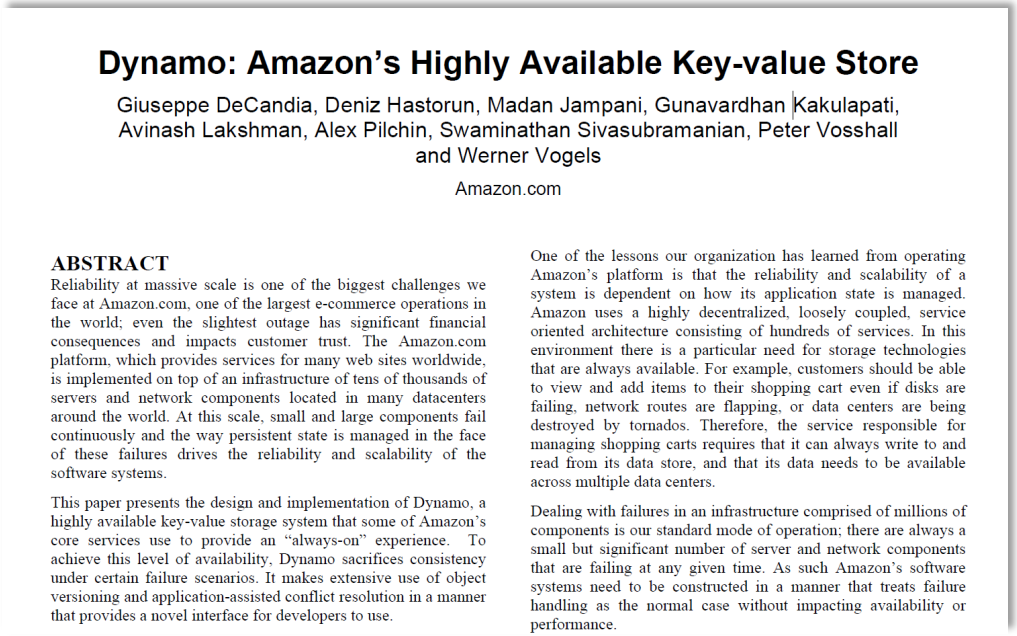
アプリ開発SDK

低コスト



DynamoDBとは

- NoSQL as a Service
- 超高速・予測可能な一貫したパフォーマンス
- 高いスケーラビリティ、そして低コスト



Dynamo: Amazon's Highly Available Key-value Store

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels

Amazon.com

ABSTRACT

Reliability at massive scale is one of the biggest challenges we face at Amazon.com, one of the largest e-commerce operations in the world; even the slightest outage has significant financial consequences and impacts customer trust. The Amazon.com platform, which provides services for many web sites worldwide, is implemented on top of an infrastructure of tens of thousands of servers and network components located in many datacenters around the world. At this scale, small and large components fail continuously and the way persistent state is managed in the face of these failures drives the reliability and scalability of the software systems.

This paper presents the design and implementation of Dynamo, a highly available key-value storage system that some of Amazon's core services use to provide an "always-on" experience. To achieve this level of availability, Dynamo sacrifices consistency under certain failure scenarios. It makes extensive use of object versioning and application-assisted conflict resolution in a manner that provides a novel interface for developers to use.

One of the lessons our organization has learned from operating Amazon's platform is that the reliability and scalability of a system is dependent on how its application state is managed. Amazon uses a highly decentralized, loosely coupled, service oriented architecture consisting of hundreds of services. In this environment there is a particular need for storage technologies that are always available. For example, customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornados. Therefore, the service responsible for managing shopping carts requires that it can always write to and read from its data store, and that its data needs to be available across multiple data centers.

Dealing with failures in an infrastructure comprised of millions of components is our standard mode of operation; there are always a small but significant number of server and network components that are failing at any given time. As such Amazon's software systems need to be constructed in a manner that treats failure handling as the normal case without impacting availability or performance.

運用管理必要なし

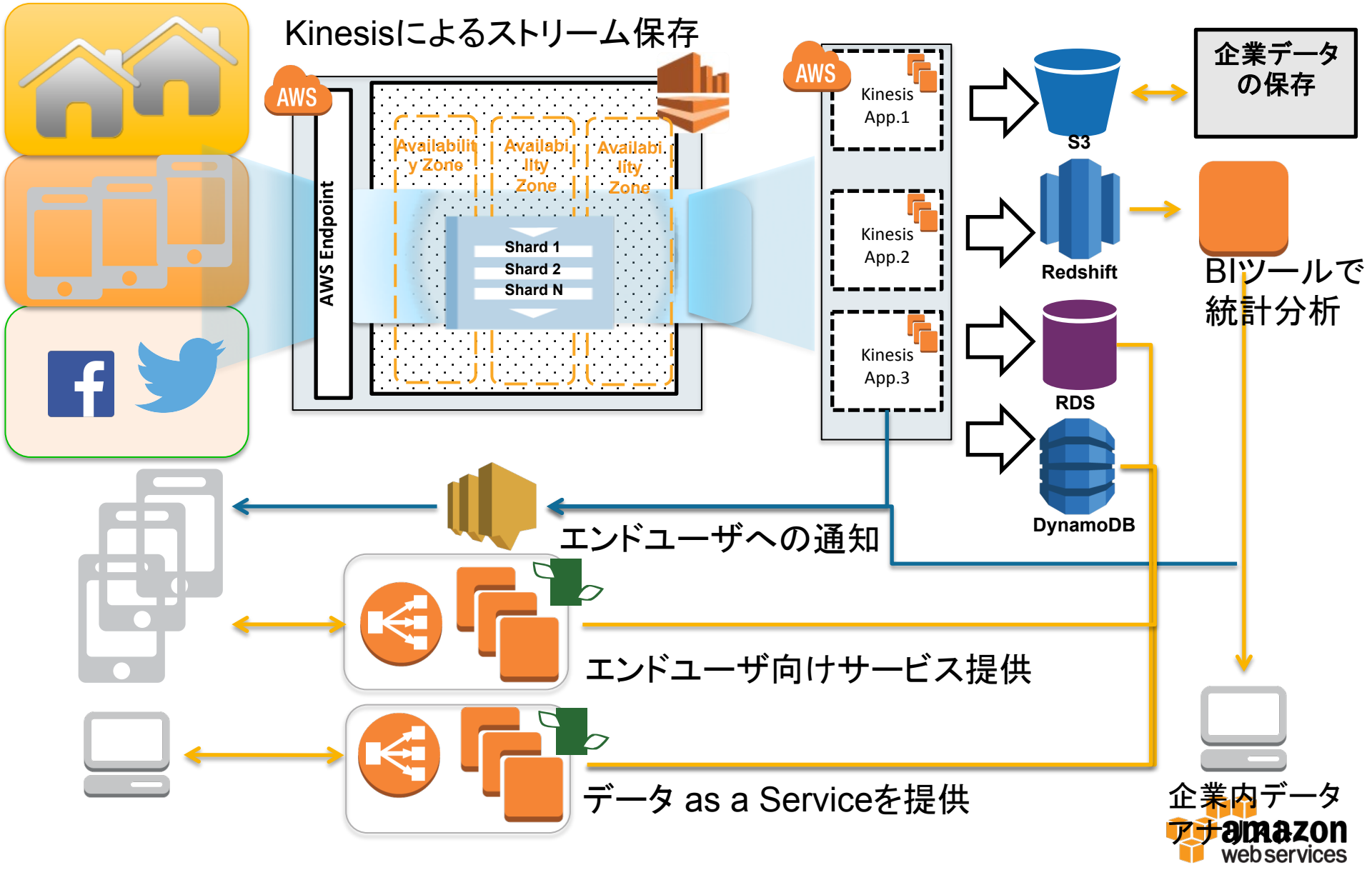
低レイテンシ、SSD

プロビジョンスループット

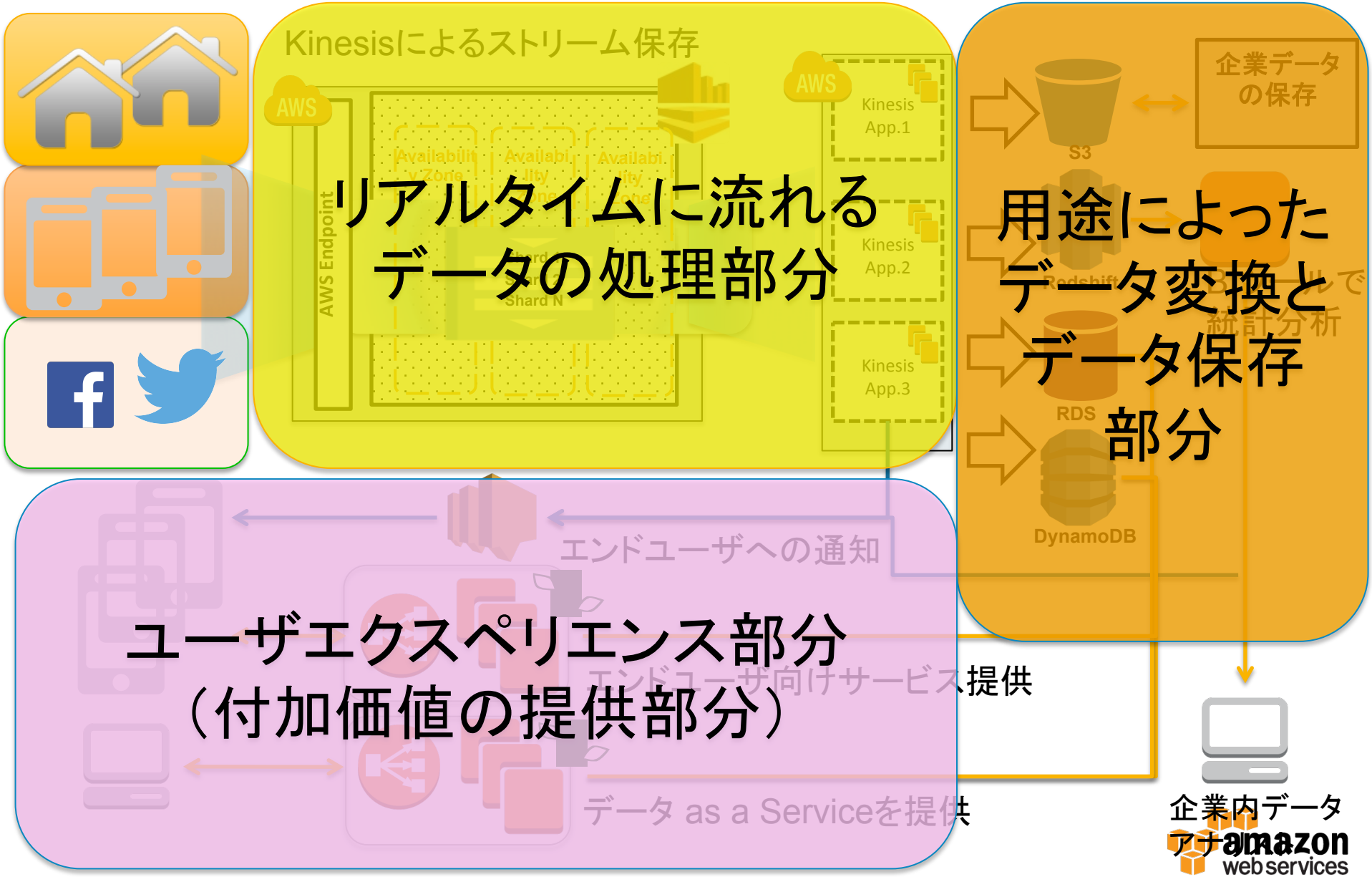
無限に使えるストレージ

amazon web services

AWSを使ったIoTアーキテクチャイメージ



AWSを使ったIoTアーキテクチャイメージ

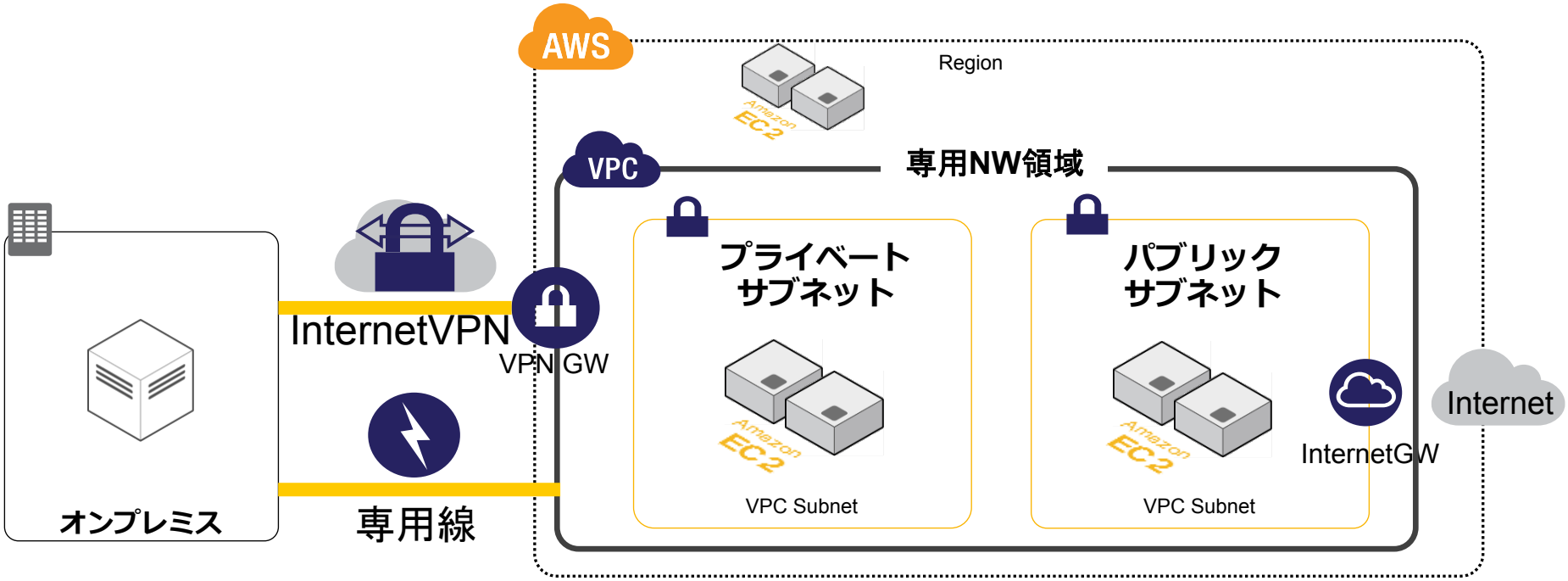


amazon web services

データ処理を支えるサービス

Virtual Private Cloud(VPC)

- クラウド内に、専用のネットワークを構築
- VPN接続することで、ハイブリッド環境が可能

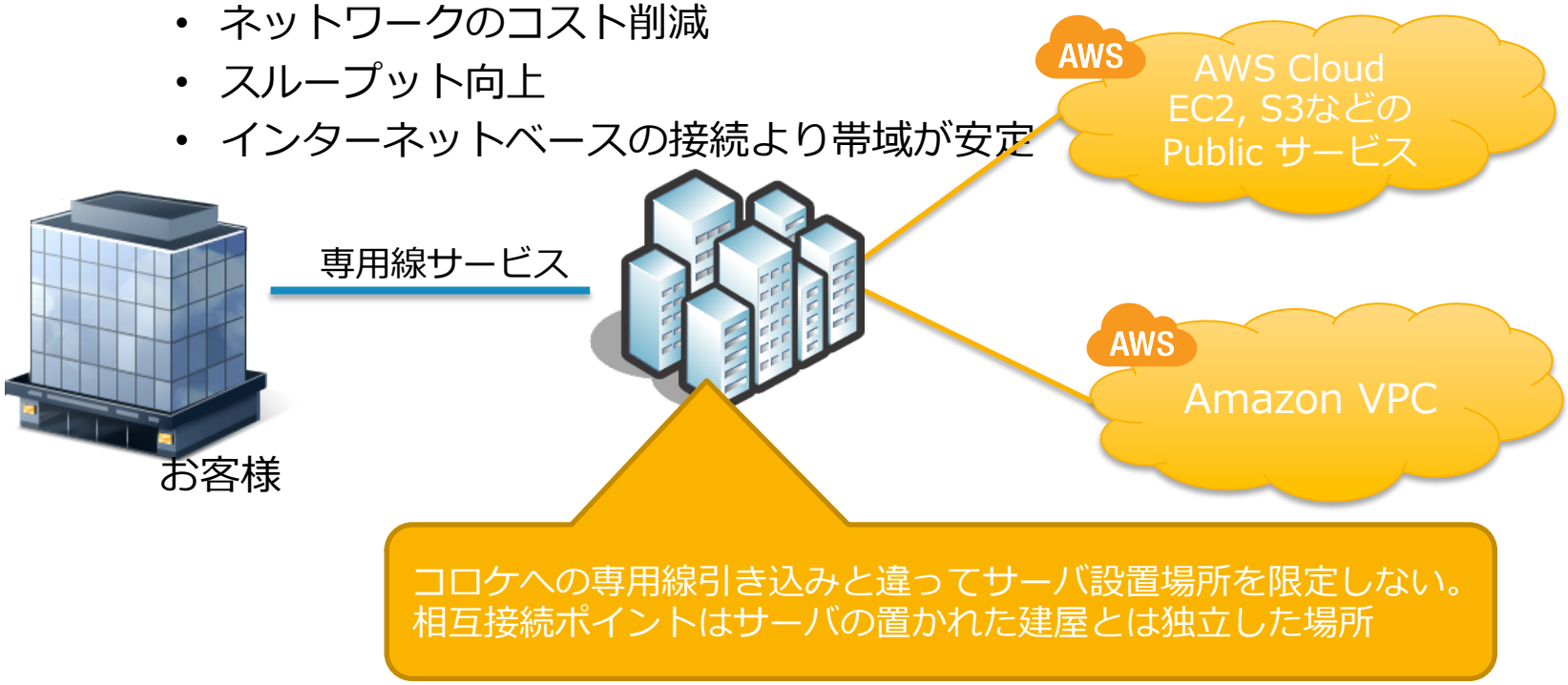


VPC内のシステム構築のために

- ネットワーク分割のベストプラクティス
  - ログインする必要のないELB，RDS, Elasticache用のサブネット
    - 目的別には分けずに、/22や/24など、わかりやすく大きめのネットワークを指定する。
  - ログインする必要のあるEC2は目的別に。
- AWSのAPI使用にはインターネット接続が必要
  - EIPを使用
  - NATインスタンスを使用
  - オンプレ側インターネット線の使用
- AWSのリソースは原則ホスト名を使ってアクセス

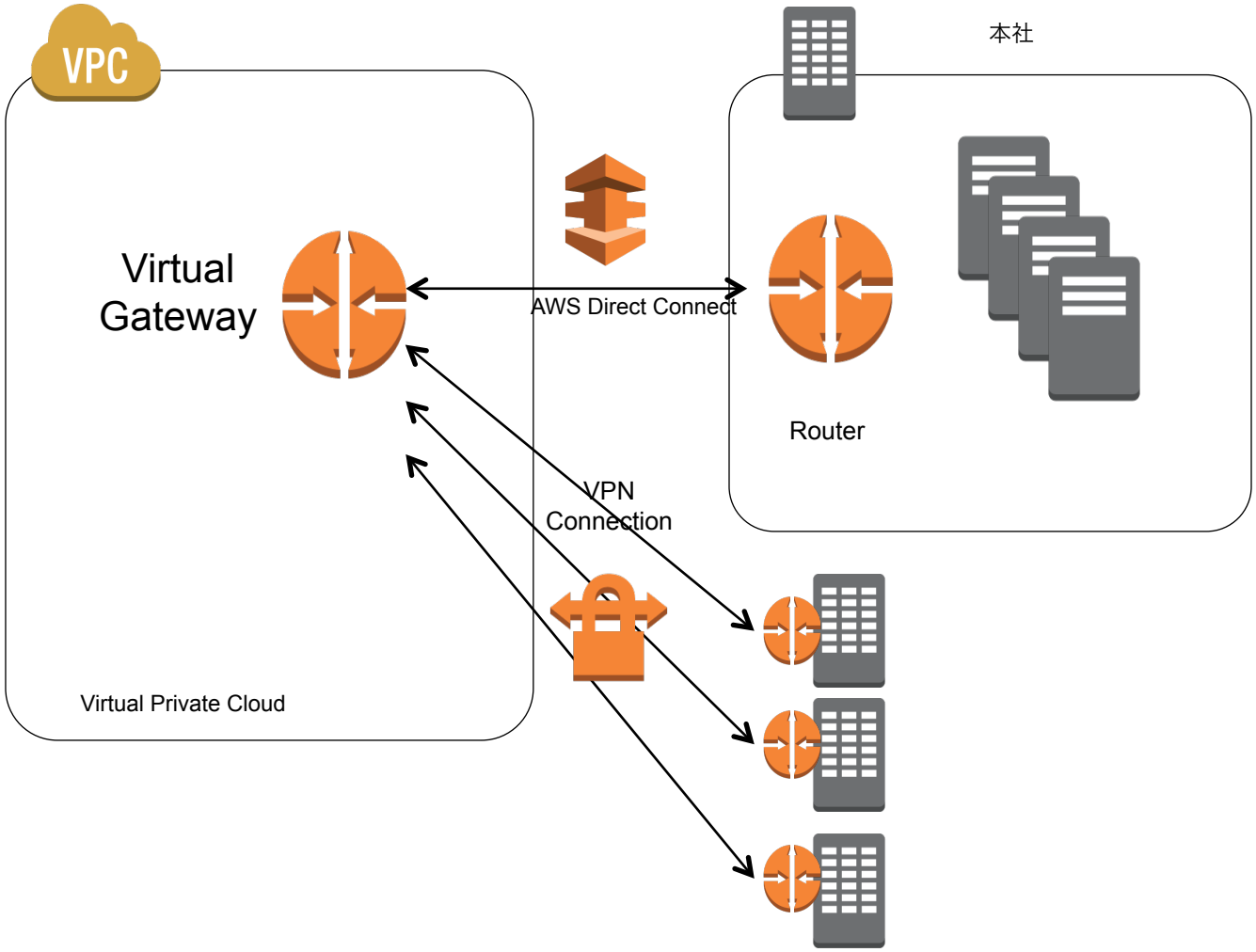
AWS Direct Connect

- AWSとデータセンター、オフィス、コロケーション環境間を専用線で接続
- 特徴
  - ネットワークのコスト削減
  - スループット向上
  - インターネットベースの接続より帯域が安定



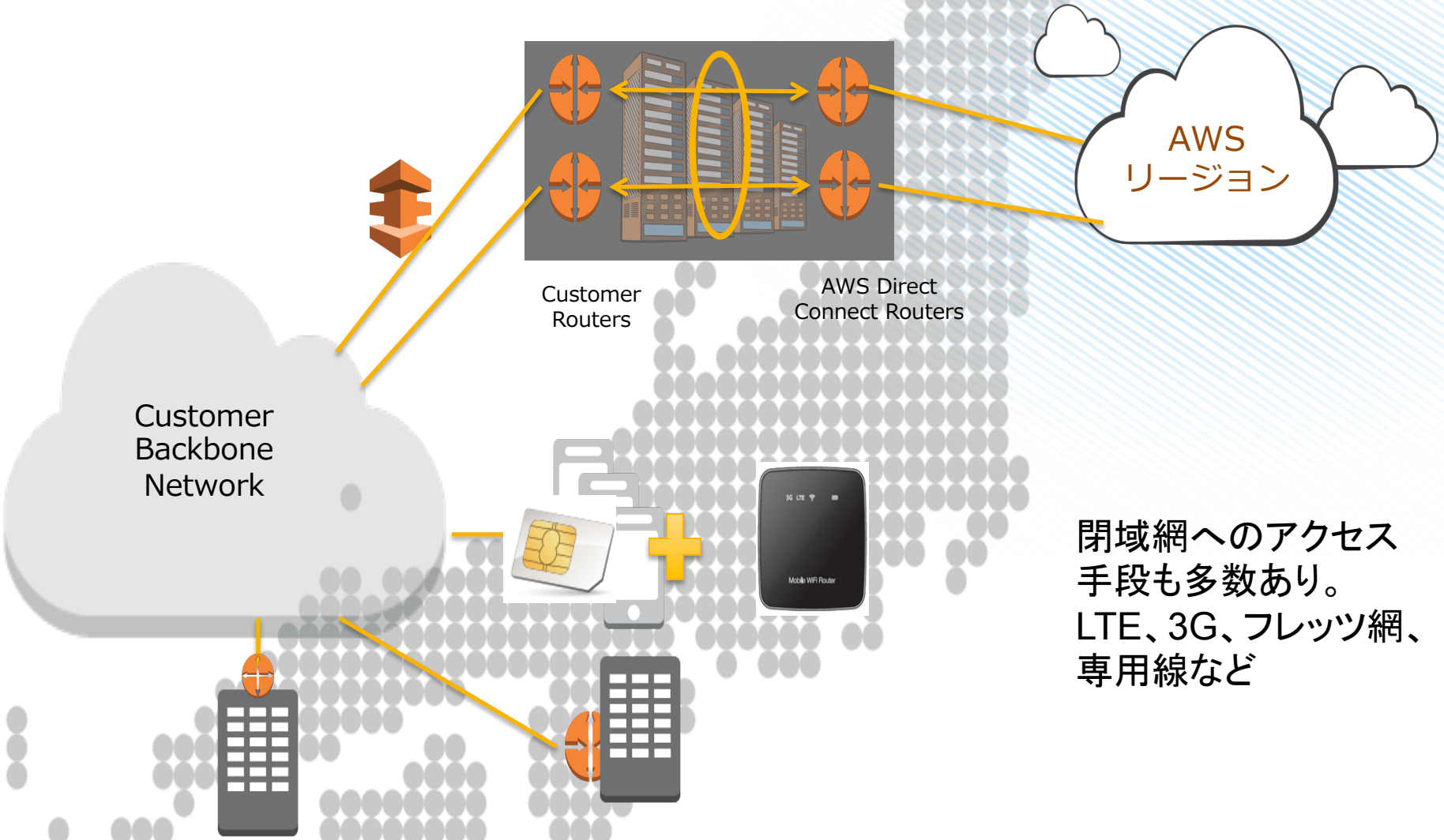
amazon web services

CloudHubのHubとしてのVPC



amazon web services

ネットワークキャリアによる拡張



閉域網へのアクセス
手段も多数あり。
LTE、3G、フレッツ網、
専用線など

amazon web services

S3へのファイルコピー

S3へ

ファイル転送高速化のために

- バケットのリージョンを確認
- 数十MBを超えるならばマルチパート化
- 並列転送
- 数十TPSを超えるならばキー名を分散化
- 無駄なオペレーションは使わない

支援プログラム例

S3への転送に特化

- Attunity CloudBeam

ATTUNITY CloudBeam

- ExpeDat S3 Gateway

Data Expedition, Inc.

- CloudBerry

CloudBerry Lab Cloud Storage Tools

amazon web services

インスタンスへのファイルコピー

任意のプロトコル、
任意のアプリケーションを利用可

転送高速化プログラムも

- Tsunami-UDP
- Aspera
- Skeed

ブロックデバイスコピー

AWS Storage Gateway

キャッシュ型:32TB/ボリューム

保管型:1TB/ボリューム

インスタンスへ

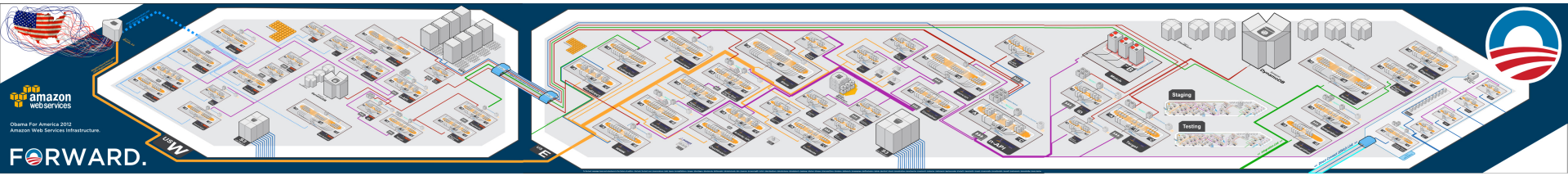
DRBD

DRBD Proxy

DataKeeper

転送最適化サービスを使った例

西海岸 ⟷ 東海岸



- データ圧縮
- 重複排除
- TCP最適化
- 伝送路暗号化

- OFAではCloudOptを利用
- Market Place でAMIも
  - CloudOpt (時間課金有り)
  - Silver Peak (BYOL)

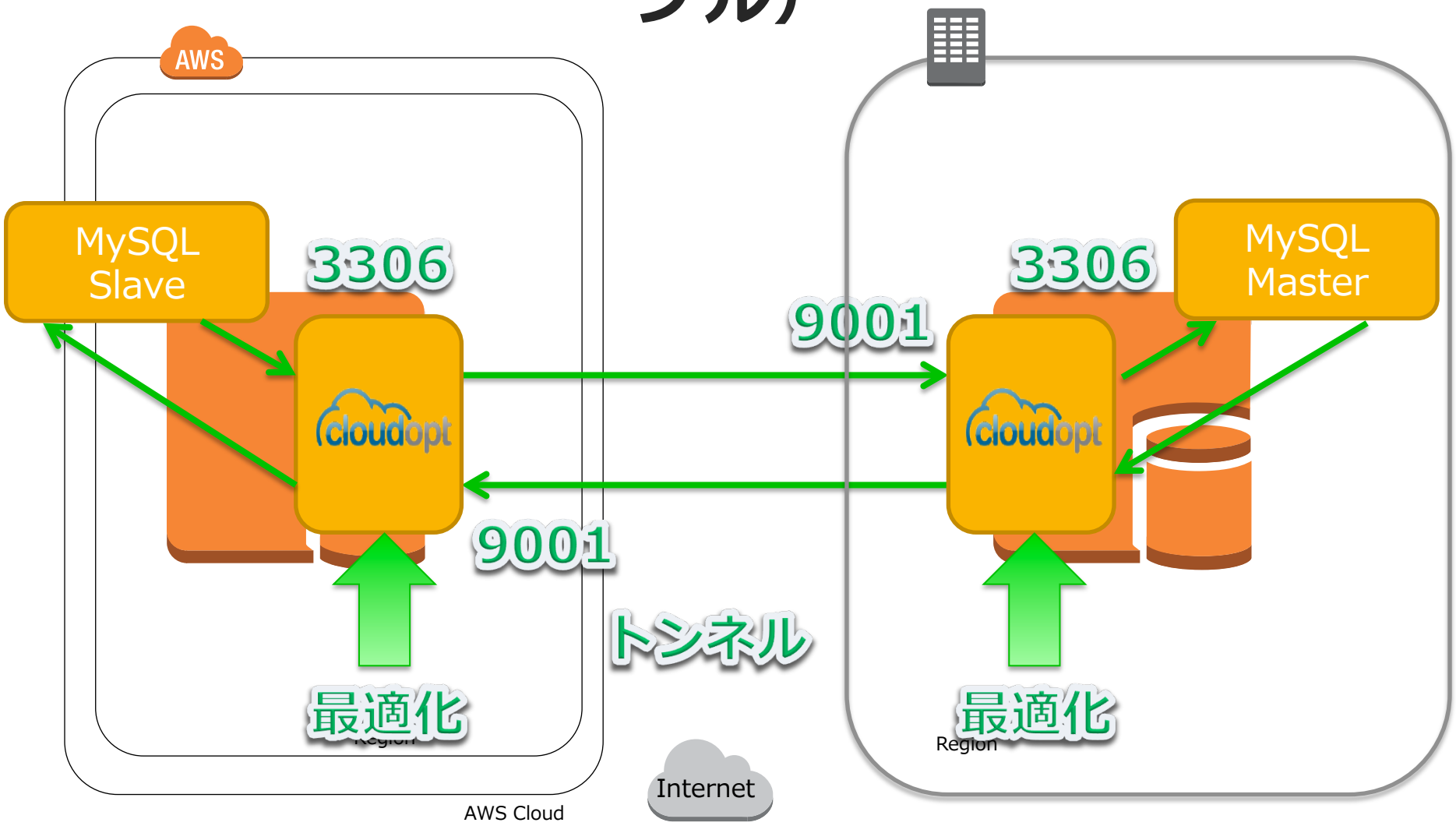
awsofa.info

cloudopt

Silver Peak

amazon web services

CloudOptの動作イメージ（MySQLのサンプル）



データ活用のためのシステム設計

インフラ

- 対象アプリ決定
- リージョン設計
- VPC設計
- インターネット VPN
- 専用線

データ集約

- S3
- EC2
  - ブロックデバイス
  - ファイル
  - VM
- Kinesis
- Storage Gateway
- Glacier

データ活用

- Redshift
- Elastic Map Reduce
- RDS
- Dynamo
- Data Pipeline


まとめ

AWSの特徴

初期投資が不要

低額な変動価格

実際の使用分のみ支払い

インフラコストの削減

柔軟性の確保

調達・構築が容易なインフラ

スケールアップ・ダウンが容易

市場投入と俊敏性の改善

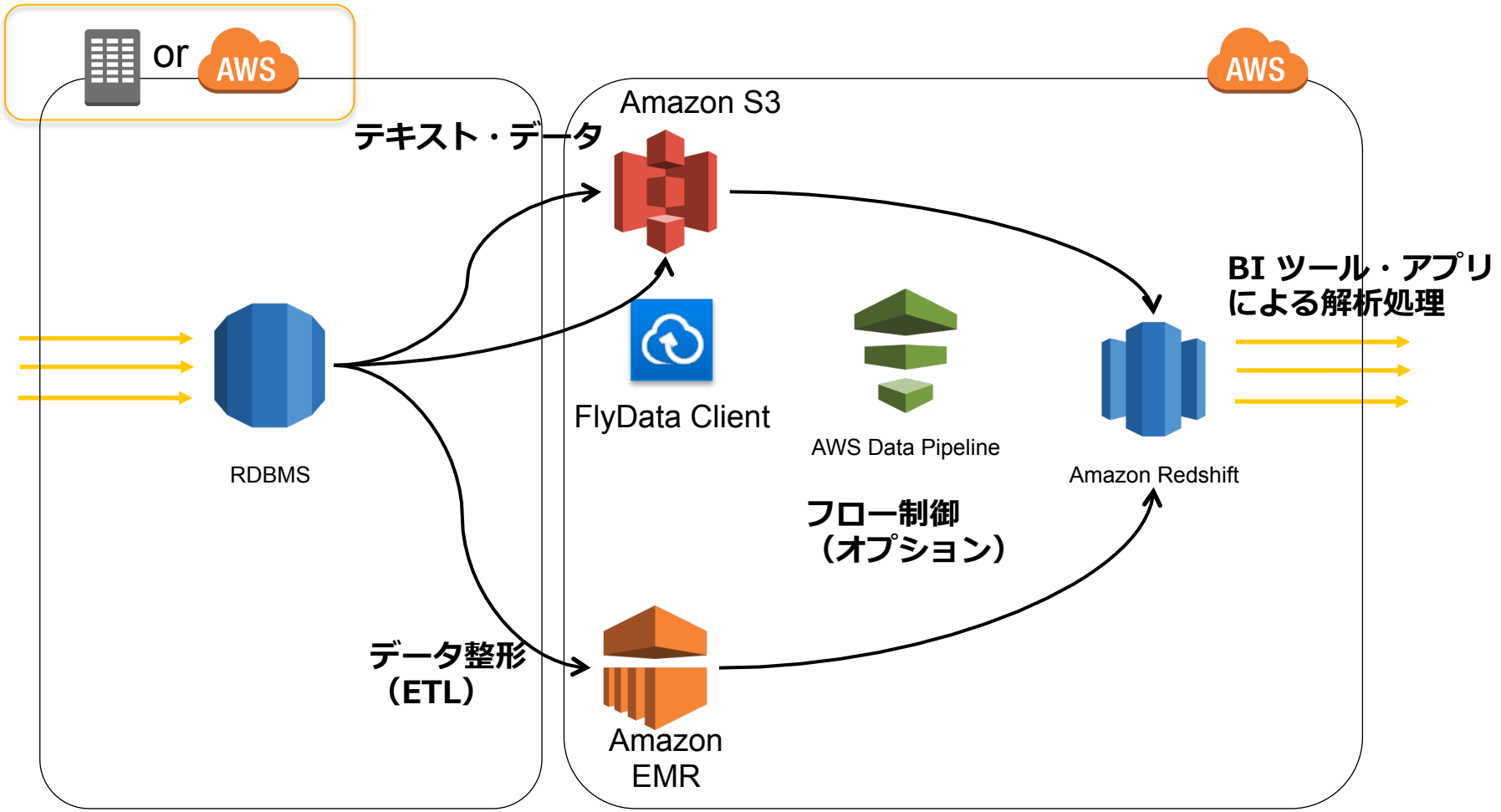
不確実なものへのチャレンジ

まとめ

- AWSはデータ収集、保存、解析、共有まで
トータルでカバーできるプラットフォーム
  - S3、EMR、Redshiftの組み合わせは強力
  - 更にKinesis, DynamoDBなどあわせるとリアルタイムからバッチまで全体をカバー
  - しかも低コストで！
- AWSを使いこなす様々なツールの存在
  - データの扱いソフトウェアからネットワークキャリアによるサービスまで。


backup

S3/EMR/Redshiftの利用イメージ



FlyData Sync for MySQL