ペイメントプラットフォームにおけるAWSの活用

楽天株式会社 グローバルテクノロジー統括部
國谷 彩 (Sayaka Kunitani)
2020年9月8日

Rakuten

自己紹介

楽天株式会社 グローバルテクノロジー統括部

國谷 彩 (Sayaka Kunitani)

経歴

- 2008年 楽天入社
- 2016年9月まで楽天市場のDBAとして従事
- 2016年10月よりペイメントプラットフォーム課ペイメントクラウドグループにてクラウドインフラエンジニアとして従事

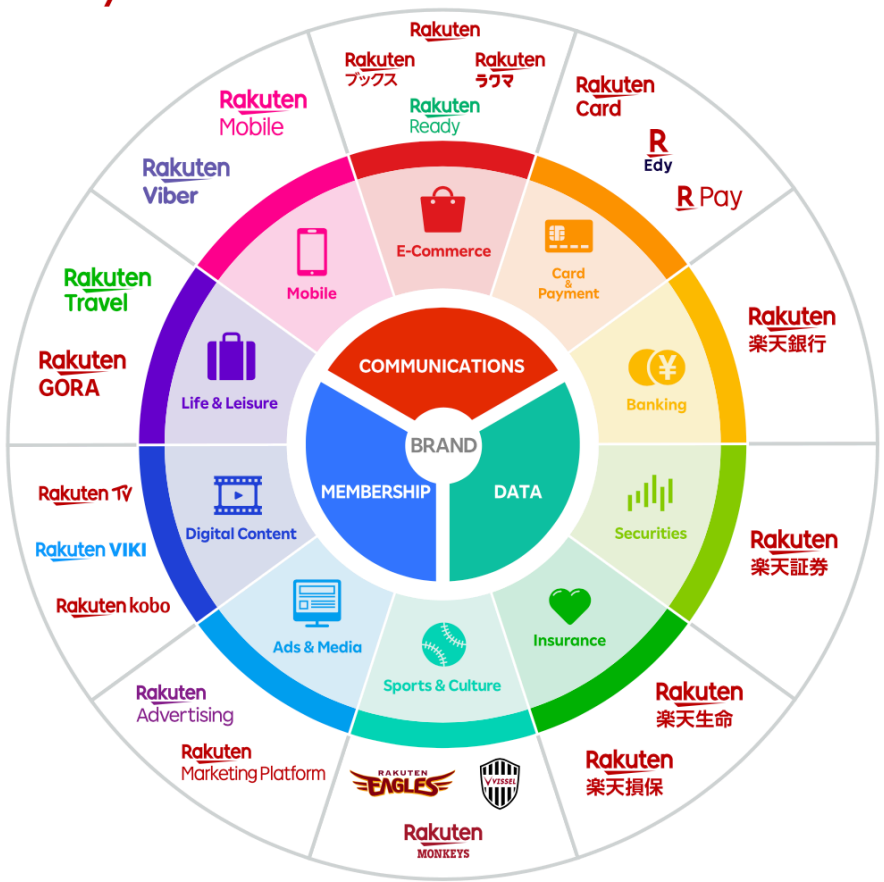
アジェンダ

- 楽天グループについて
- ペイメントプラットフォームについて
- ペイメントプラットフォームにおける
  アマゾンウェブ サービス(AWS)の歴史
- まとめ

楽天グループについて

楽天グループについて

Rakuten Ecosystem

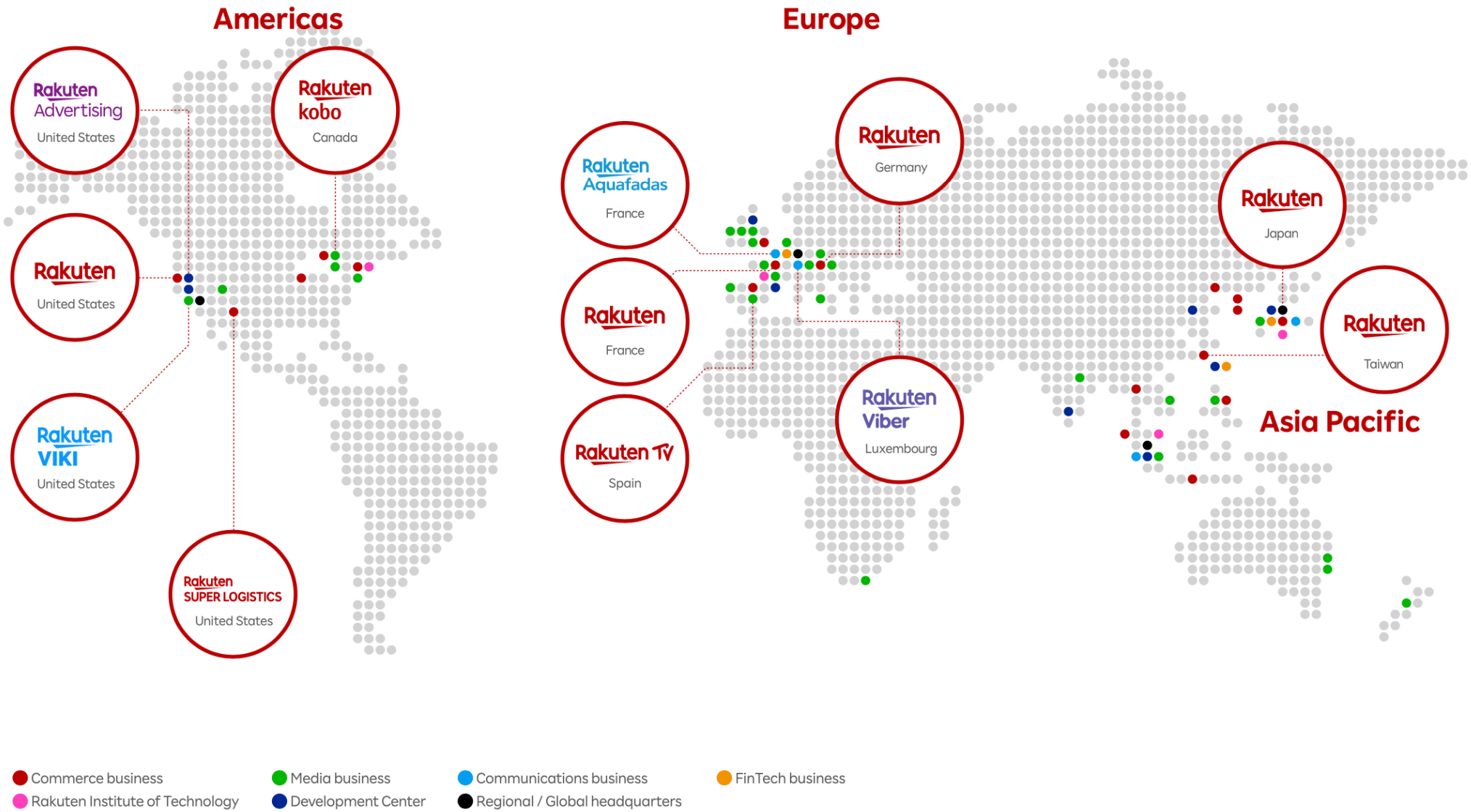


サービス展開	グループサービス利用者数	サービス数	グローバル流通総額
30カ国・地域	約14億	70超	19.0兆円
			*2019年度

楽天会員数	楽天市場出店店舗数	クロスユース率
1億以上※1	48,661店舗	71.1%※2
連結売上高	グローバル流通総額	国内EC流通総額
1.1兆円	15.4兆円※3	3.9兆円
楽天カード取扱高	楽天銀行口座数	
7.5兆円※5	800万口座突破	

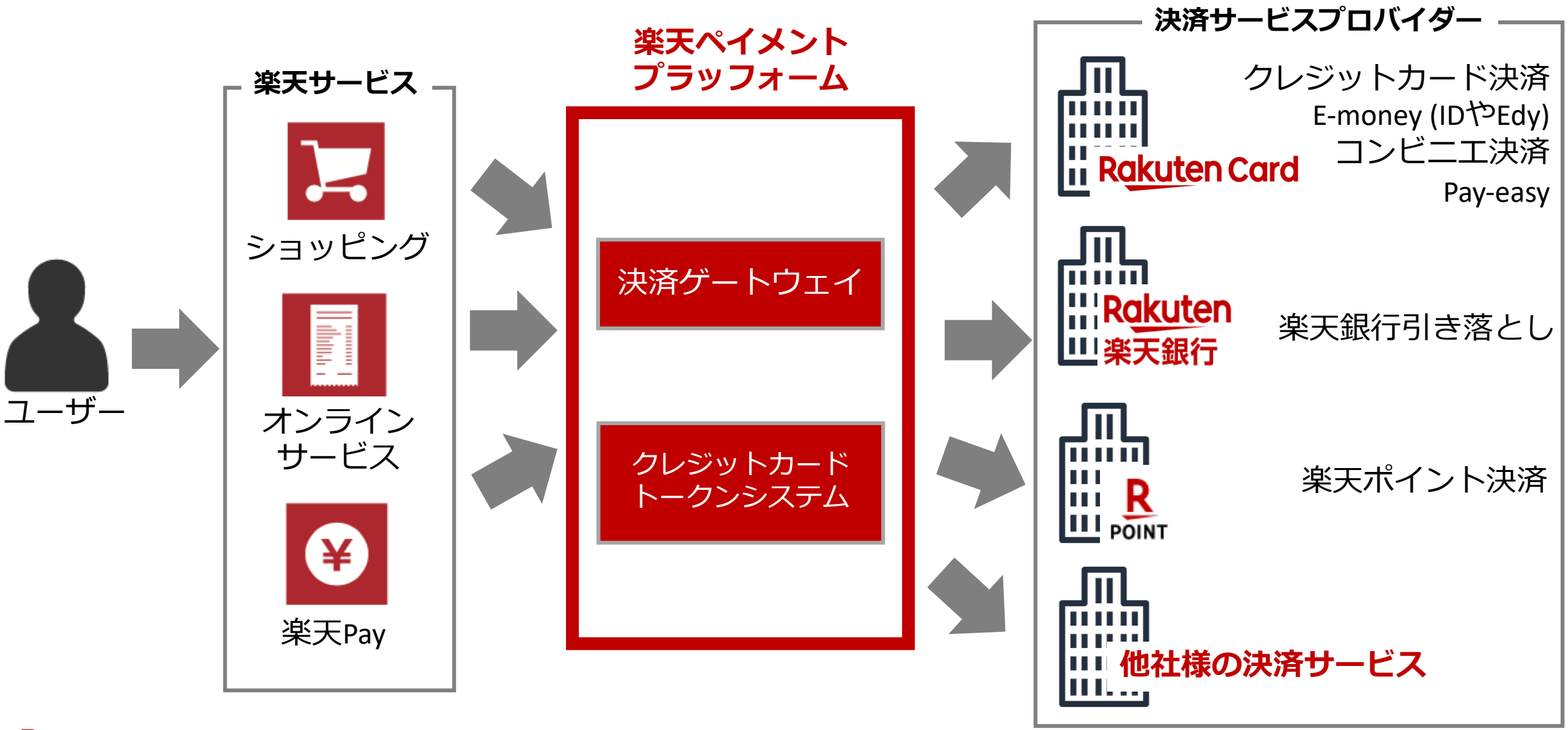
楽天会員数、楽天市場出店店舗数、クロスユース率、楽天銀行口座数は2019年9月時点。それ以外の数値は、2018年の累計。

グローバル展開

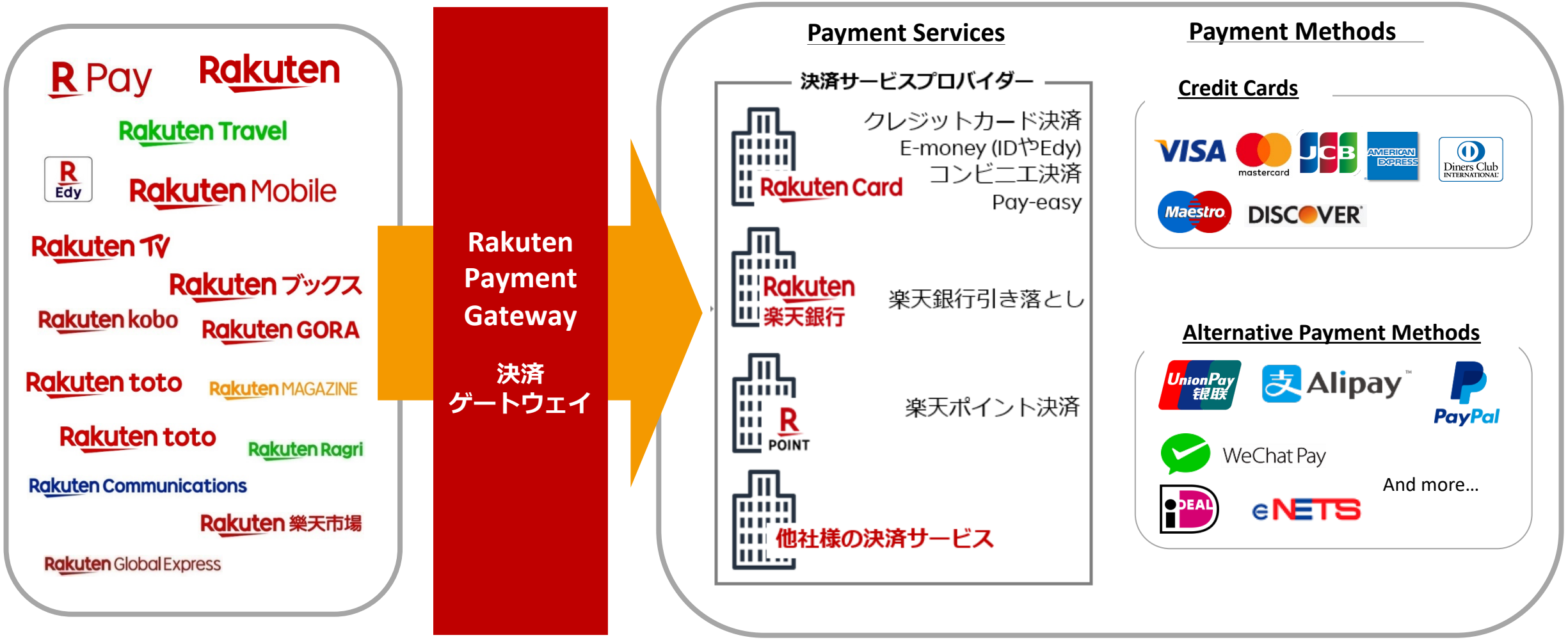

Americas
Rakuten Advertising
United States
Rakuten kobo
Canada
Rakuten
United States
Rakuten VIKI
United States
Rakuten SUPER LOGISTICS
United States
Europe
Rakuten Aquafadas
France
Rakuten
France
Rakuten TV
Spain
Rakuten
Germany
Rakuten Viber
Luxembourg
Rakuten
Japan
Rakuten
Taiwan
Asia Pacific
Commerce business
Media business
Communications business
FinTech business
Rakuten Institute of Technology
Development Center
Regional / Global headquarters

ペイメントプラットフォームについて

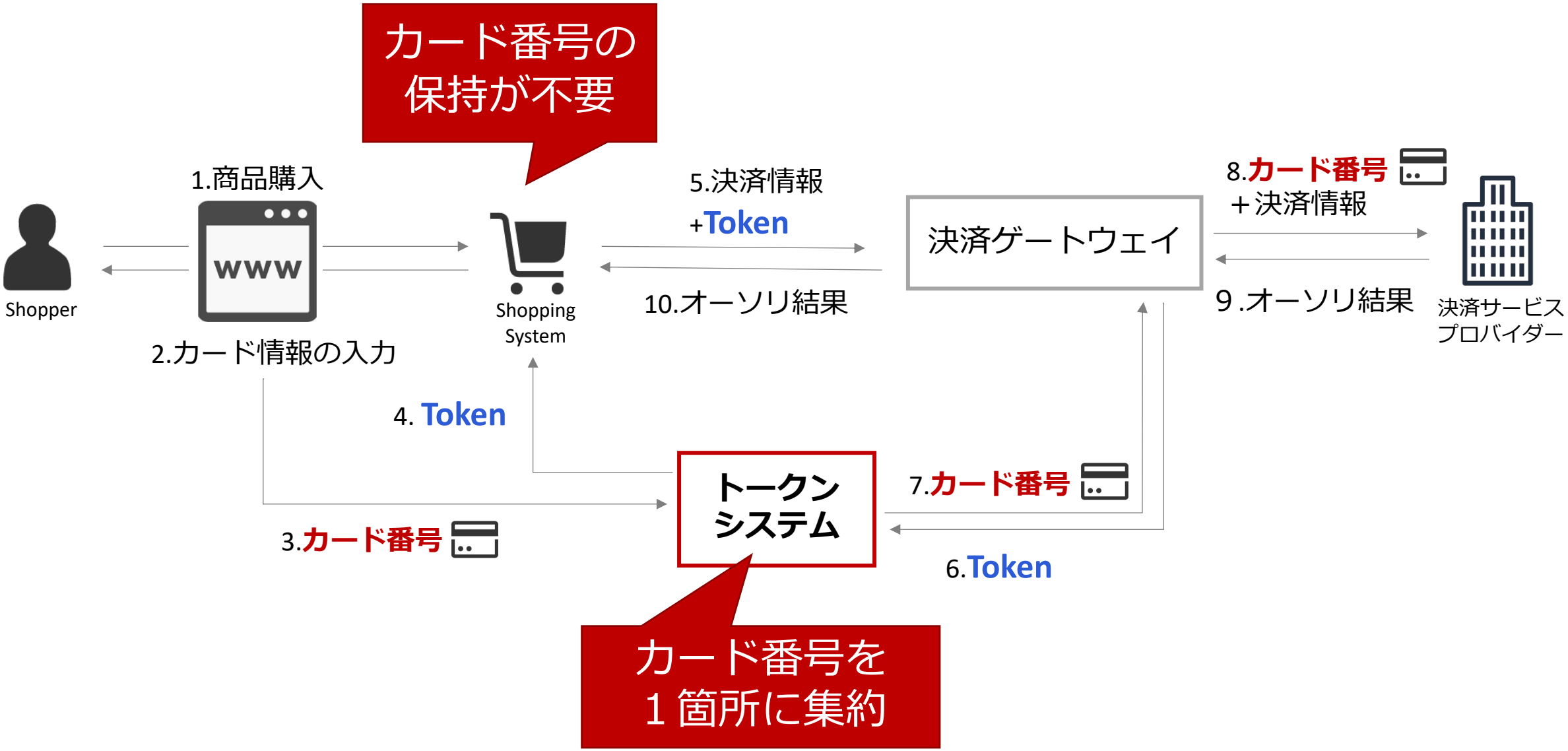
ペイメントプラットフォームとは？



決済ゲートウェイ


R Pay
Rakuten
Rakuten Travel
R Edy
Rakuten Mobile
Rakuten TV
Rakuten ブックス
Rakuten kobo
Rakuten GORA
Rakuten toto
Rakuten MAGAZINE
Rakuten toto
Rakuten Ragri
Rakuten Communications
Rakuten 楽天市場
Rakuten Global Express
Rakuten Payment Gateway
決済 ゲートウェイ
Payment Services
決済サービスプロバイダー
Rakuten Card
クレジットカード決済
E-money (IDやEdy)
コンビニエ決済
Pay-easy
Rakuten 楽天銀行
楽天銀行引き落とし
R POINT
楽天ポイント決済
他社様の決済サービス
Payment Methods
Credit Cards
VISA
mastercard
JCB
AMERICAN EXPRESS
Diners Club INTERNATIONAL
Maestro
DISCOVER
Alternative Payment Methods
UnionPay 銀联
Alipay
PayPal
WeChat Pay
iDEAL
eNETS
And more...

トークンシステム

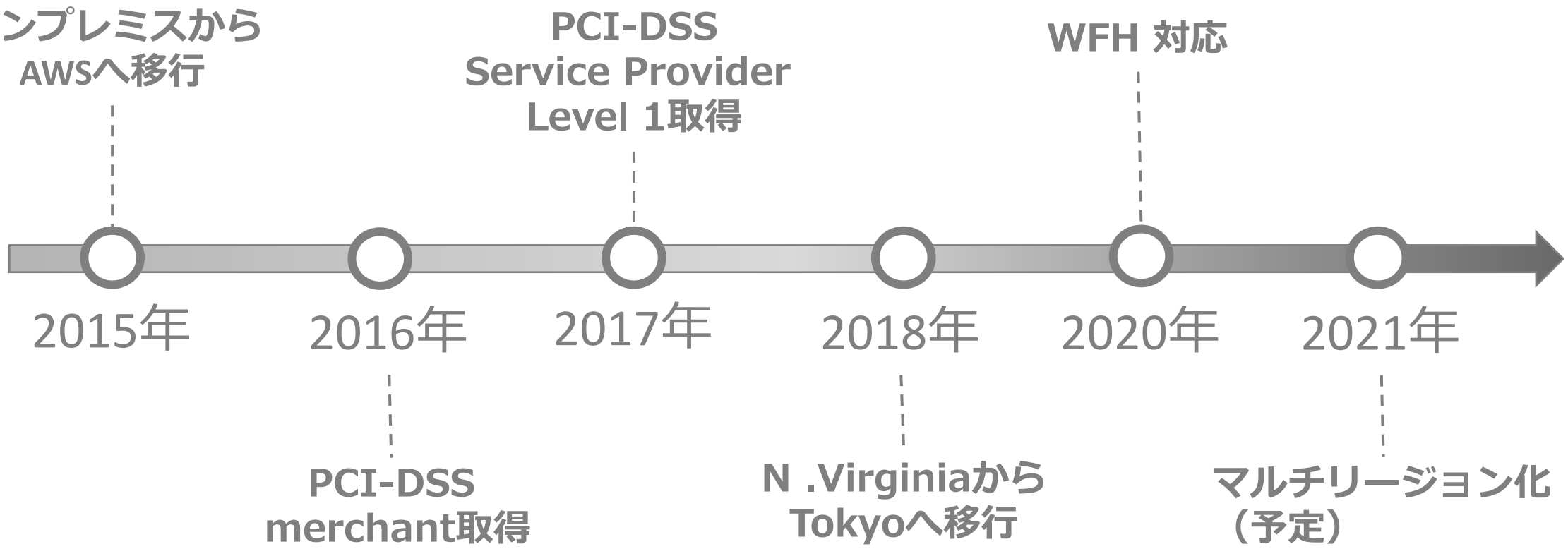

カード番号の
保持が不要
1.商品購入
Shopper
www
2.カード情報の入力
Shopping
System
5.決済情報
+Token
10.オーソリ結果
決済ゲートウェイ
8.カード番号
＋決済情報
9 .オーソリ結果
決済サービス
プロバイダー
4. Token
3.カード番号
トークン
システム
7.カード番号
6.Token
カード番号を
1 箇所に集約

実績

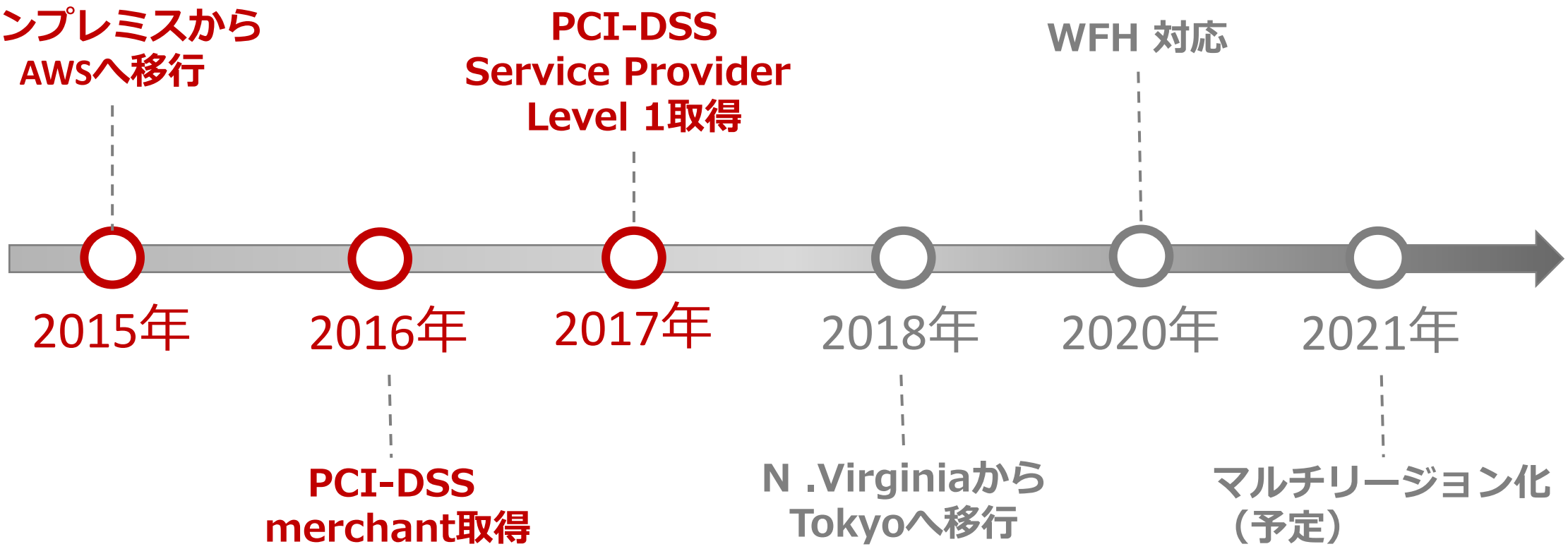

Rakuten toto
Rakuten 樂天市場
Rakuten プレミアム
R Pay
Rakuten Delivery
Rakuten MAGAZINE
Rakuten Global Express
Rakuten RAXY
Rakuten Travel
Rakuten クラウド
Rakuten ブックス
Rakuten ラクマ
Rakuten kobo
Rakuten Car
Rakuten 楽天生命
Rakuten 楽天損保
Rakuten
Rakuten Mobile
Rakuten SEIYU ネットスーパー
Rakuten Super English
Rakuten STAY
Rakuten チケット
Rakuten RaCoupon
Rakuten LIVE
Rakuten Energy
Rakuten TV
Rakuten コレクション

ペイメントプラットフォームにおける アマゾンウェブ サービス(AWS)の歴史

ペイメントプラットフォームにおけるAWSの歴史



2015年 – 2017年


オンプレミスから
AWSへ移行
PCI-DSS
Service Provider
Level 1取得
WFH 対応
2015年
2016年
2017年
2018年
2020年
2021年
PCI-DSS
merchant取得
N .Virginiaから
Tokyoへ移行
マルチリージョン化
（予定）

背景

2015年

システム管理者　セキュリティ担当者

PCI-DSSをオンプレで取るのはリードタイムやコスト面から厳しい

2016年

PCI-DSS merchant

システム管理者　セキュリティ担当者

トランザクションの増加により次のレベルの監査が必要

2017年

PCI-DSS Service Provider Level1

システム管理者　セキュリティ担当者　外部監査

PCI-DSS監査会社の監査を受ける必要がある。

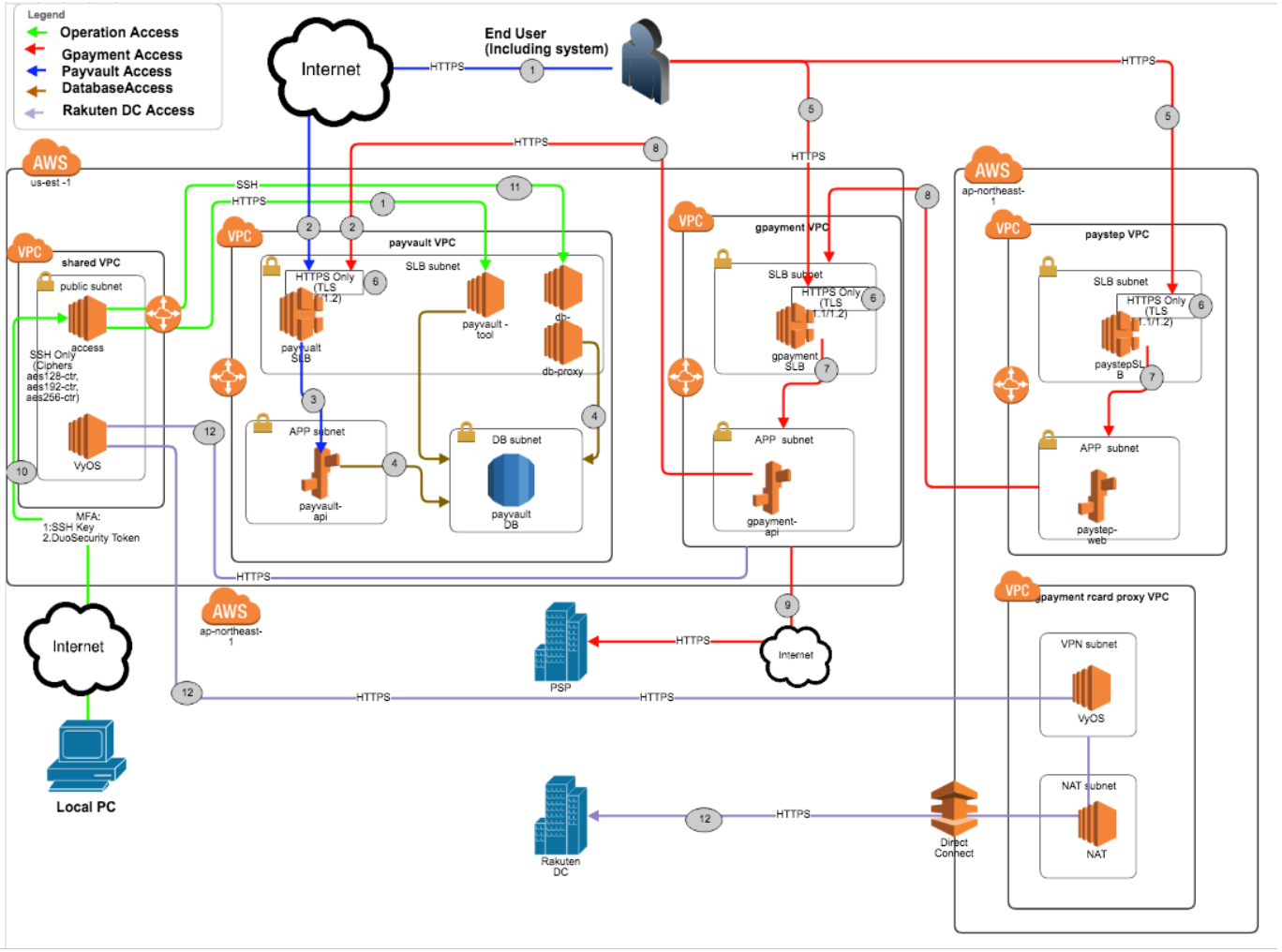
要件と課題

要件　**PCI-DSSに準拠**

課題

- **ネットワークセグメンテーション**
- **ネットワーク設定の可視化, 敏速な設定変更**
  - サーバーが何台あろうとも手元のコードで仕様やセキュリティの状況を把握できる必要がある
- **カードデータの保護**
  - 業務上必要な範囲内に制限
  - データの暗号化
- **本番前の確実なテストと再現性**

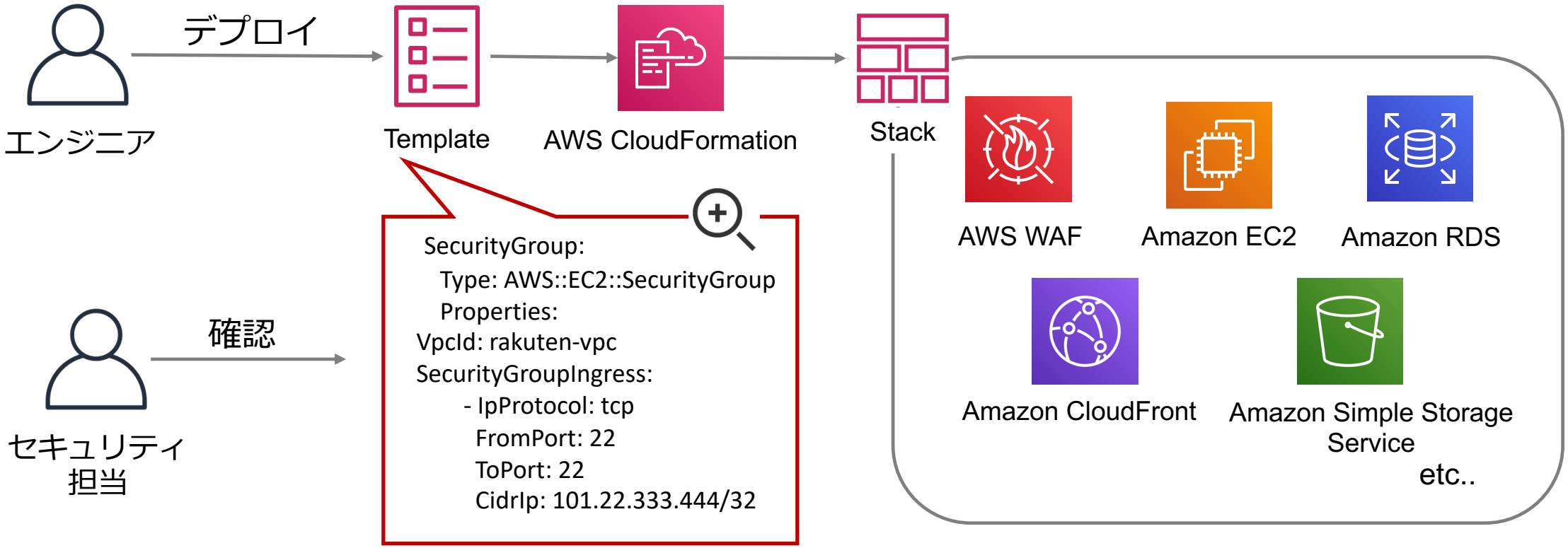
解決策 – AWSへの移行と細かいセグメンテーションの実施



- サービスに応じてネットワーク境界線(VPCレベル)を分離
- インスタンス単位でのセキュリティグループ適用
- IPアドレスとポート番号単位でのアクセス制御
- WAFの導入
- KMSの導入

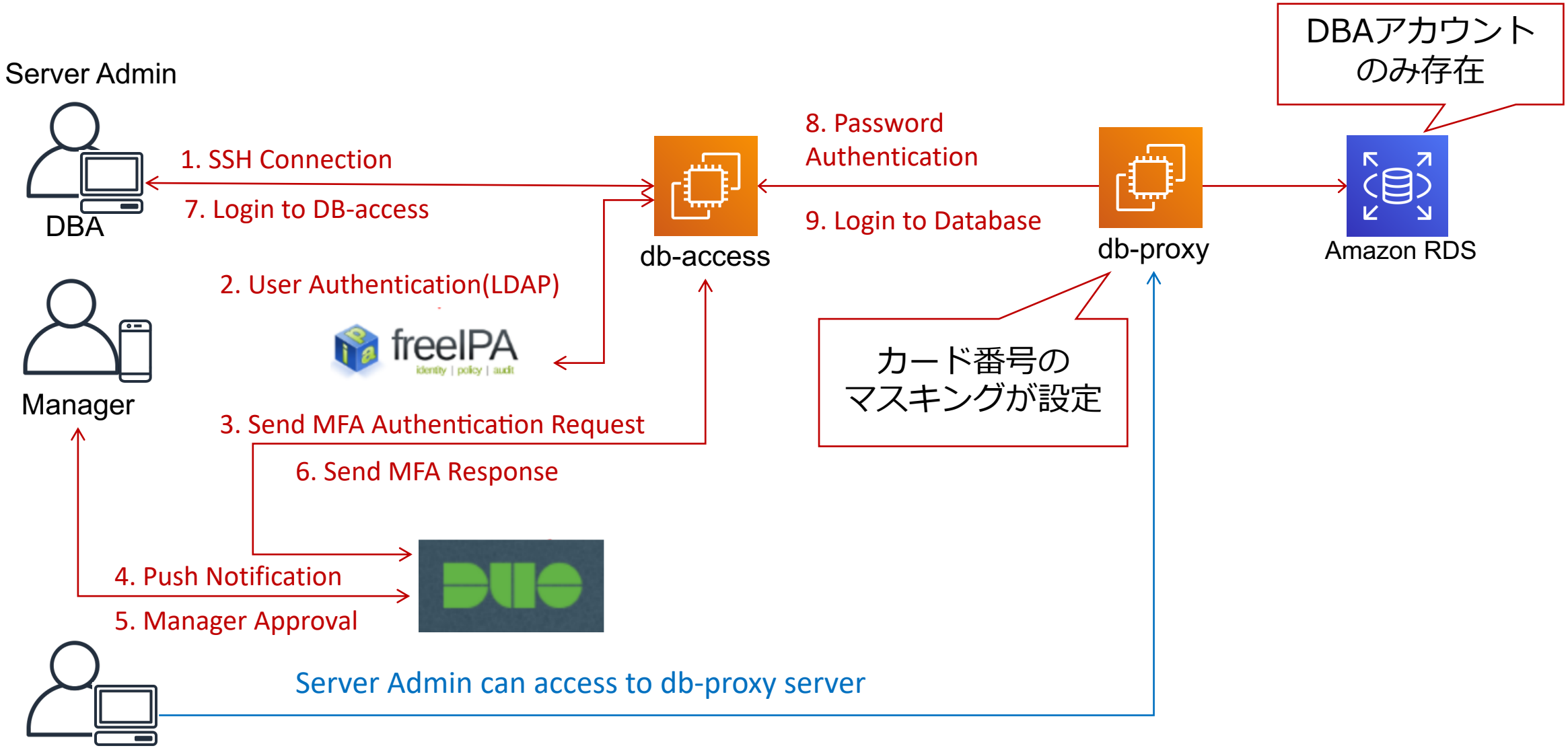
解決策 - CloudFormation の導入

- サーバーが何台あろうとも手元のコードで仕様やセキュリティの状況を把握できる
- Git のバージョン管理により、いつ誰が何を変更したのか履歴が追える



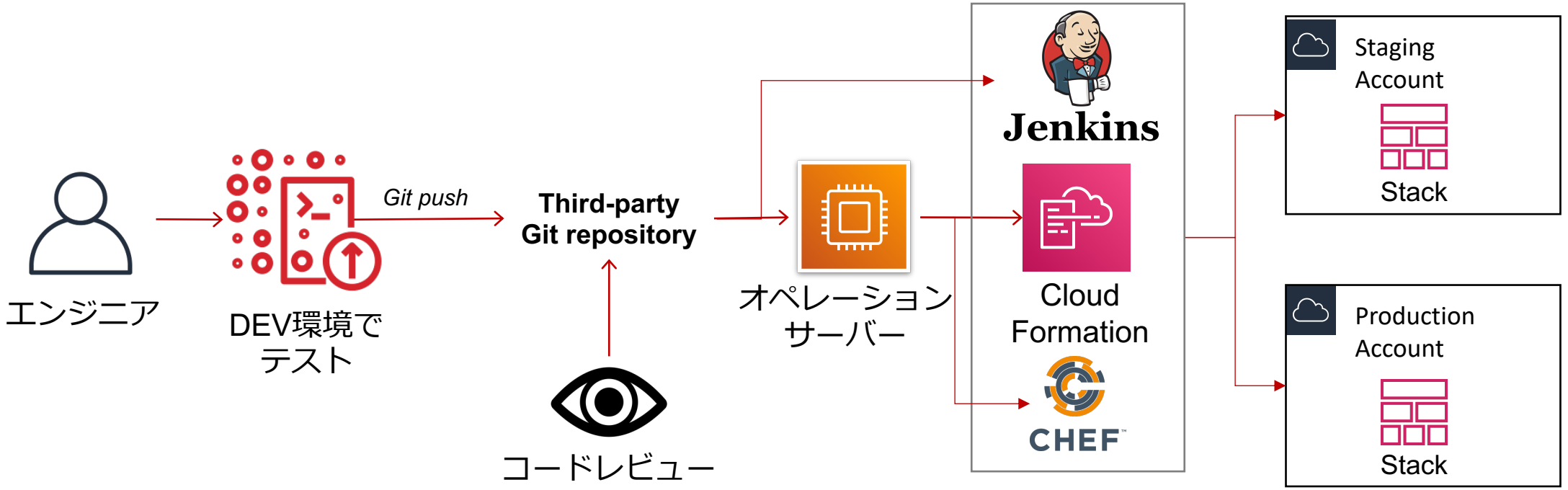
解決策 – MFAによるアクセス制御とクレジットカード番号の暗号化

責任点を分けることで内部に悪意ある人間がいても互いに牽制することができる

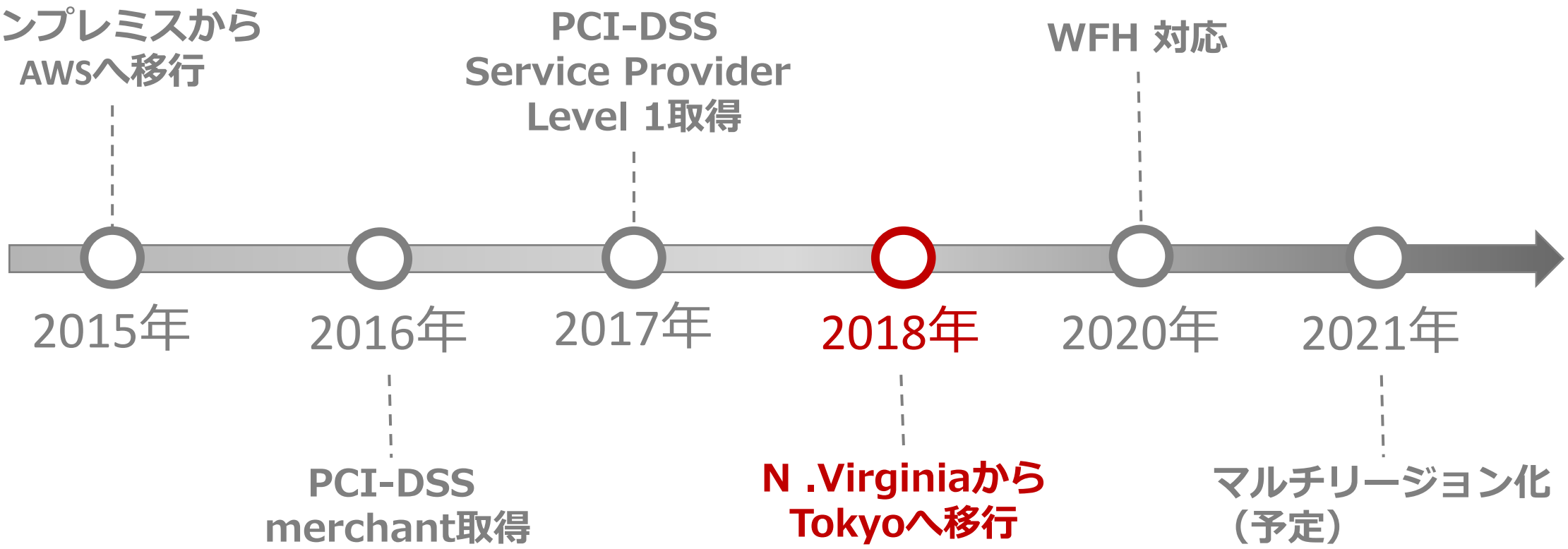


解決策 – コードベースでのリリース

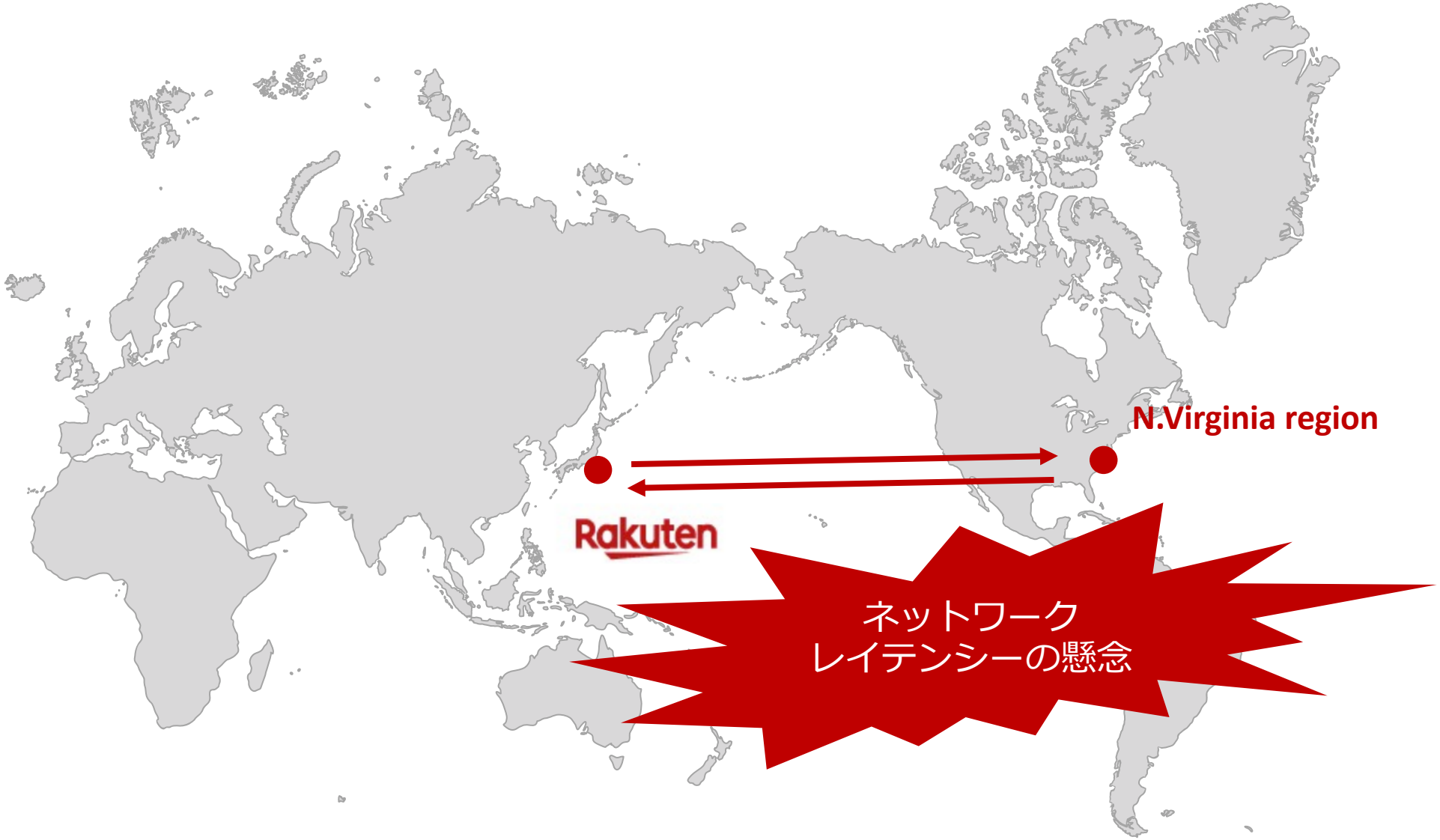
ステージング環境と本番環境で必ず同じ内容をデプロイできる



2018年


オンプレミスから
AWSへ移行
PCI-DSS
Service Provider
Level 1取得
WFH 対応
2015年
2016年
2017年
2018年
2020年
2021年
PCI-DSS
merchant取得
N .Virginiaから
Tokyoへ移行
マルチリージョン化
（予定）

背景



要件と課題

要件　**楽天市場のサービスレベルへコミット**

課題

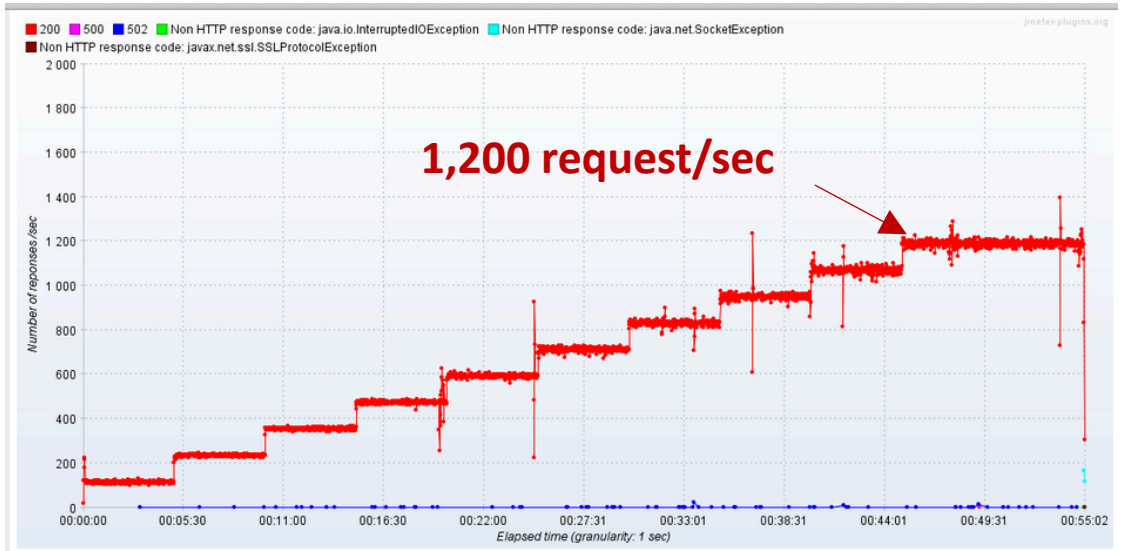
- 楽天市場のトラフィックを捌くためのシステム導入
- 安定したネットワークの提供
- ネットワークレイテンシーの改善

解決策 – RDSからAuroraへの移行

楽天スーパーセール時のトラフィック 1,200 request/sec に耐えられるシステムが必要

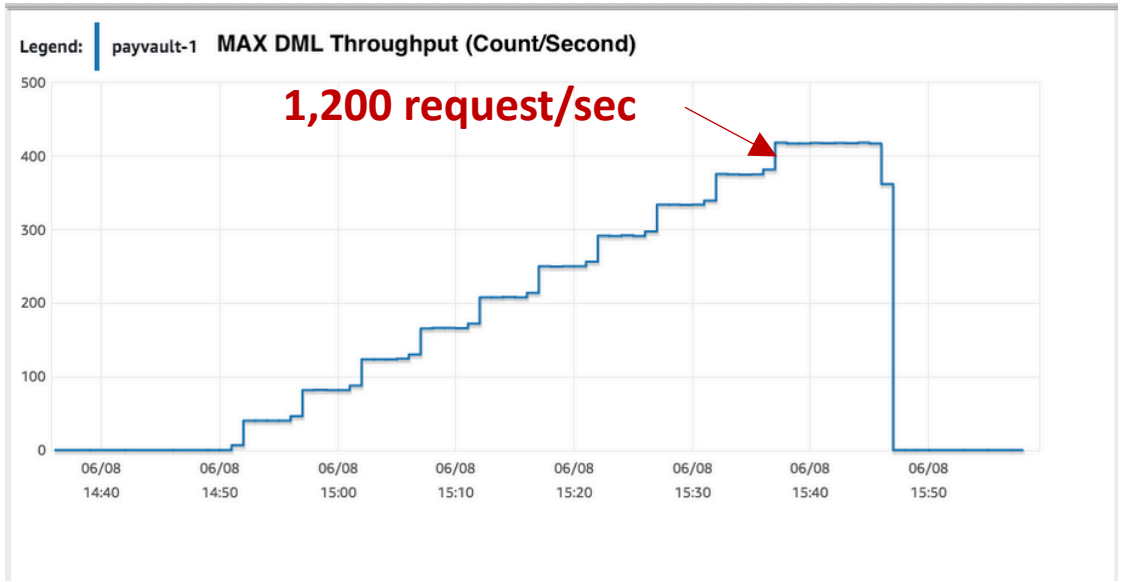
既存のRDSだとパフォーマンス要件は満たせなかった。
並行処理に強くRDSと比較するとDisk I/Oの負荷軽減が見込まれるAuroraへ移行した。

アプリケーション



DBがボトルネックとならずを目標値を達成

Aurora



目標値までスループット（パフォーマンス）が上昇し続けた

解決策 – スーパーセール時の暖気申請、BLUE-GREEN デプロイメント 対応

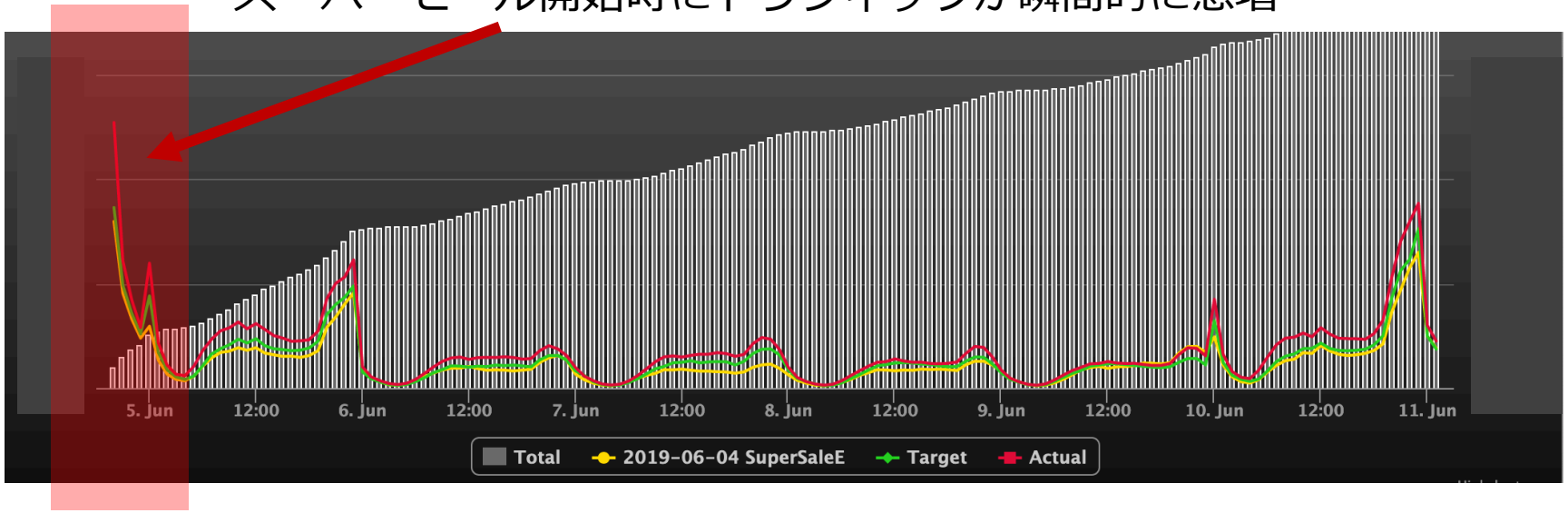
暖気申請

ALBは負荷に応じて自動スケールするが、急激なトラフィックで瞬間的に急増する場合は、
ALBのスケーリングが間に合わないので事前に暖気申請でALBをWarm up しておく

BLUE-GREEN デプロイメント対応

スーパーセール中にメモリーリークが発生してもすぐにeb swapできるよう事前に環境準備

スーパーセール開始時にトラフィックが瞬間的に急増

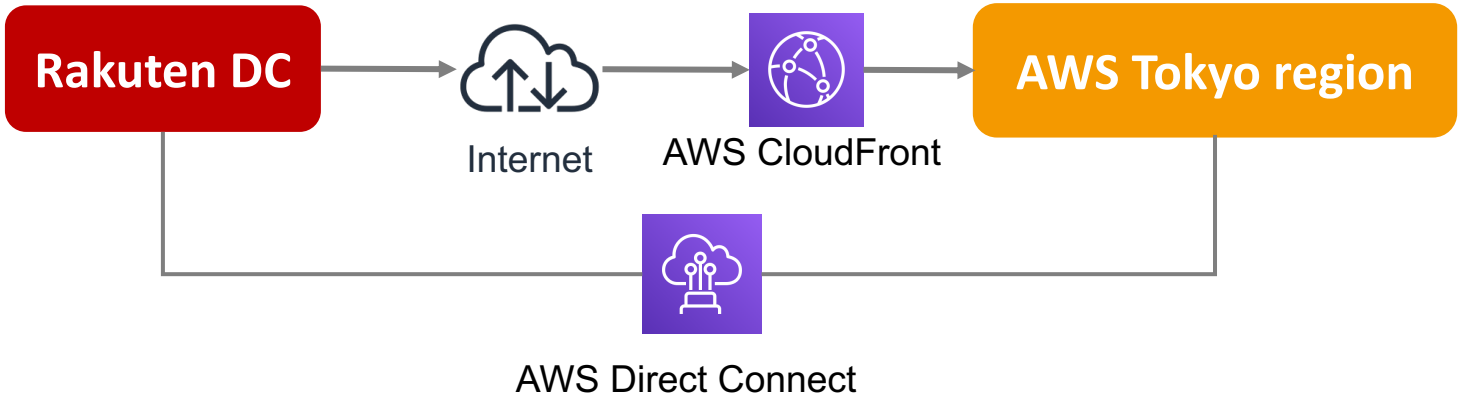


解決策 - Direct Connect の導入とTokyo regionへの移行

As Is

- Rakuten DCとAWS N.Virgina regionが離れているためパフォーマンスが良くない
- Rakuten DCとAWS N.Virgina region のネットワークが不安定
- ネットワークのルートが一つしかなく、 耐障害性に弱い

To Be

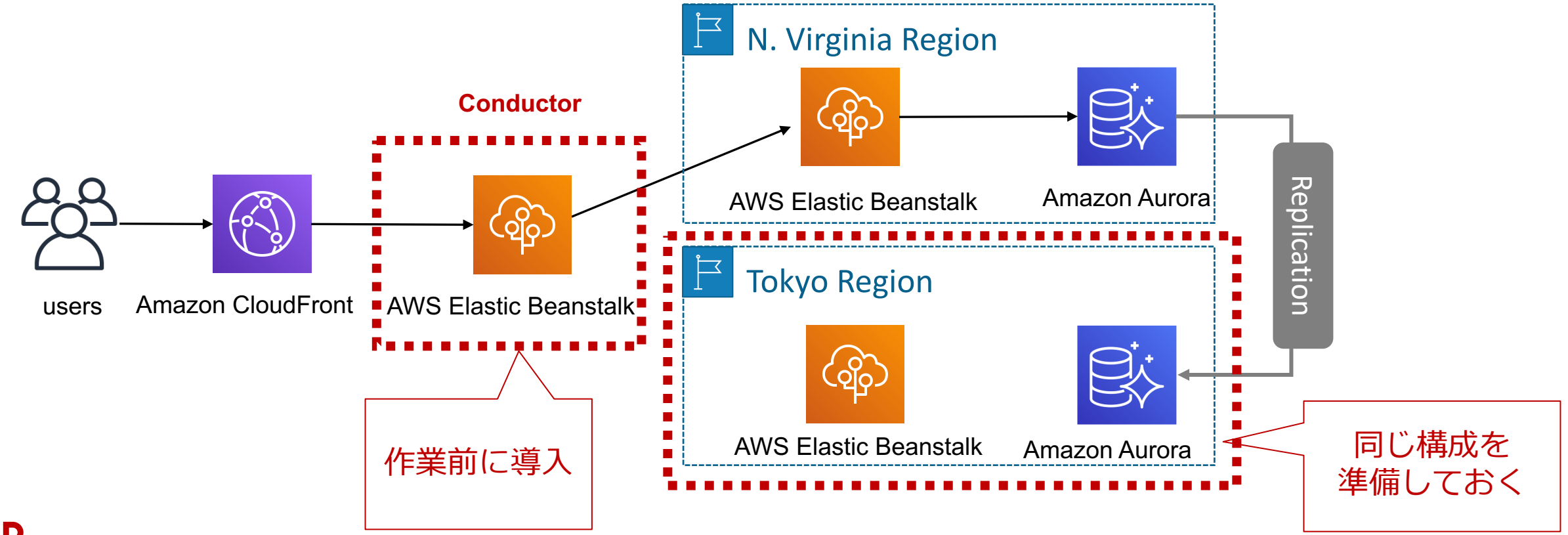


- Rakuten DCとAWSの距離が近くなることでレイテンシーが改善
- Rakuten DCとAWS Tokyo regionのネットワークが安定した（AkamaiからCloudFrontへの切り替え）
- インターネットとプライベートの両方のルートを用意し耐障害性を強化

サービス無停止でのシステム移行

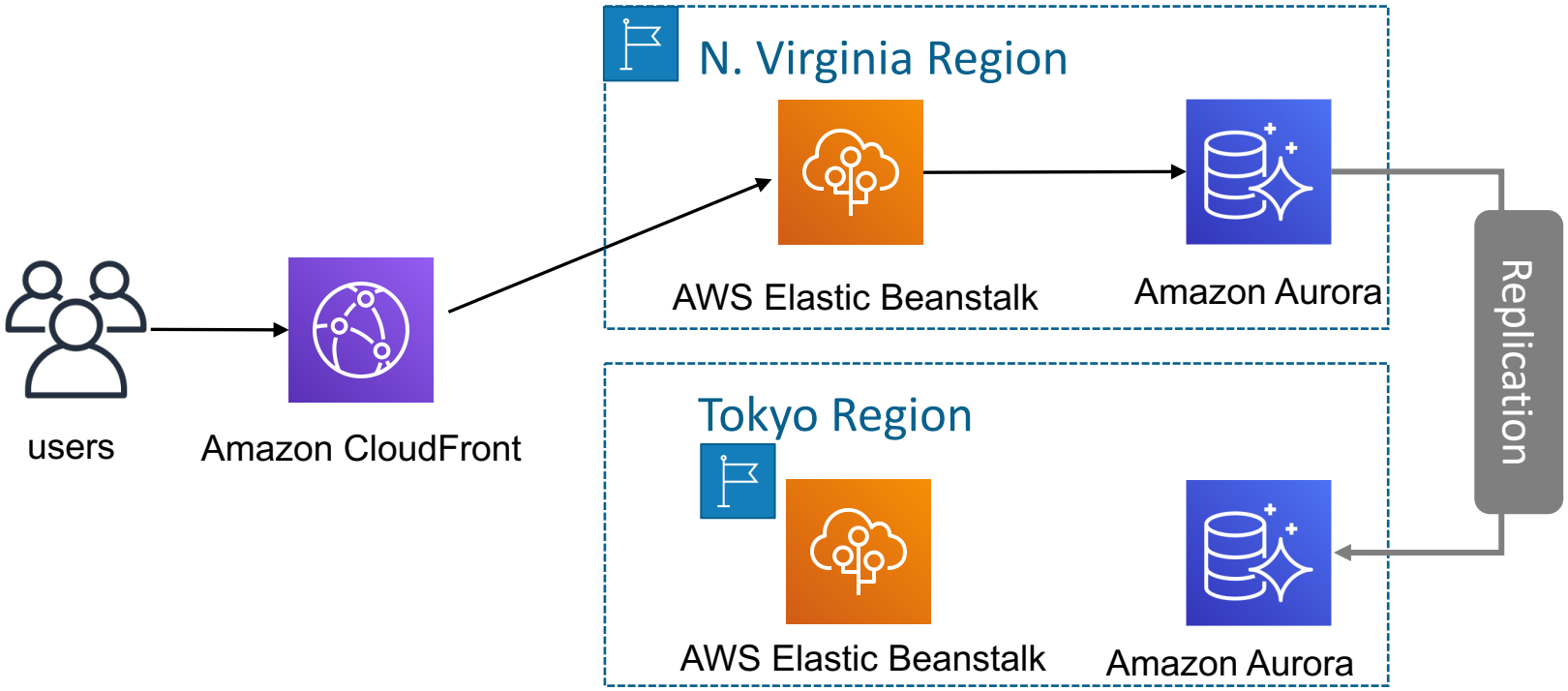
サービス無停止の定義

- アプリケーションにエラーを返さない
- ユーザーからはレスポンスが少し遅いかな？と感じる程度に止める（＞30sec）



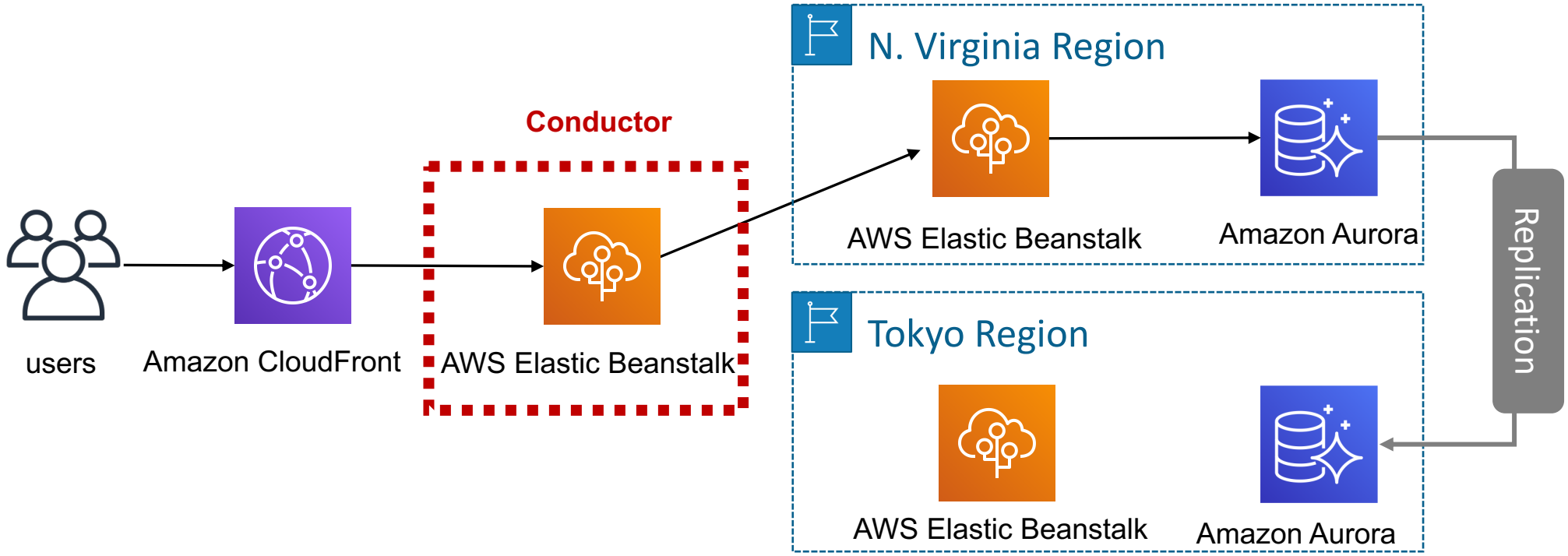
コンダクターの実装 #0

コンダクターの導入 → サスペンドモード → DBの移行 → トラフィックの切り替え → コンダクターを外す



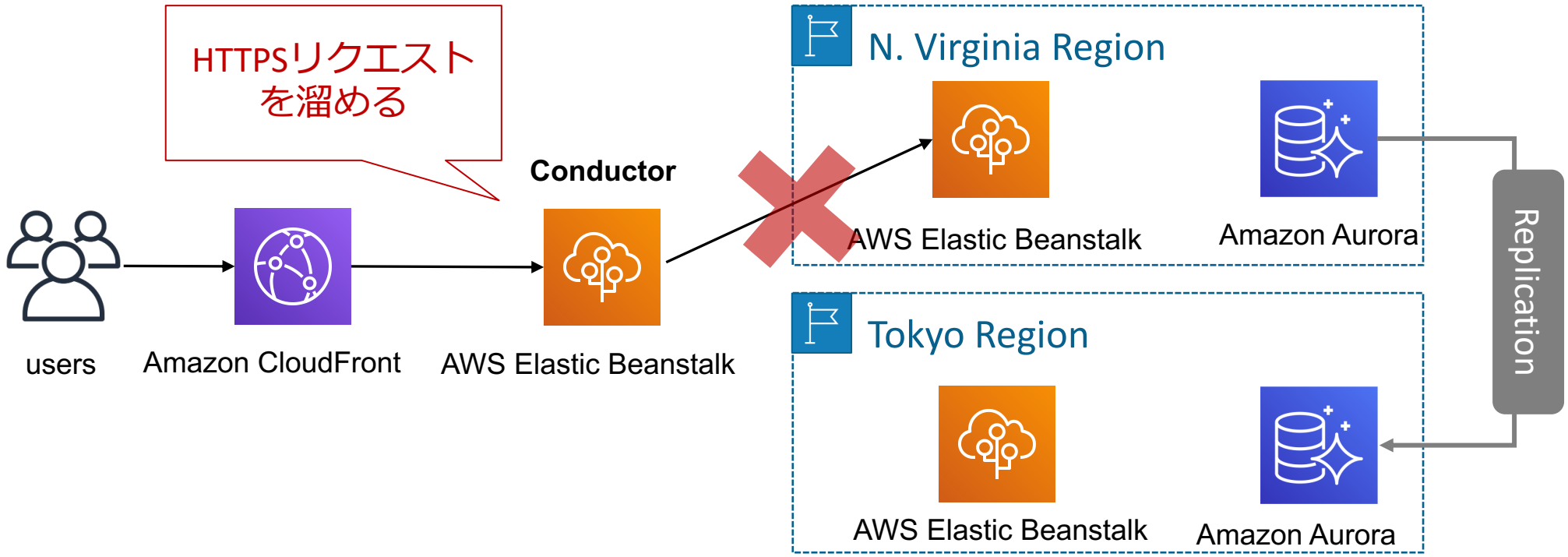
コンダクターの実装 #1

コンダクターの導入 → サスペンドモード → DBの移行 → トラフィックの切り替え → コンダクターを外す



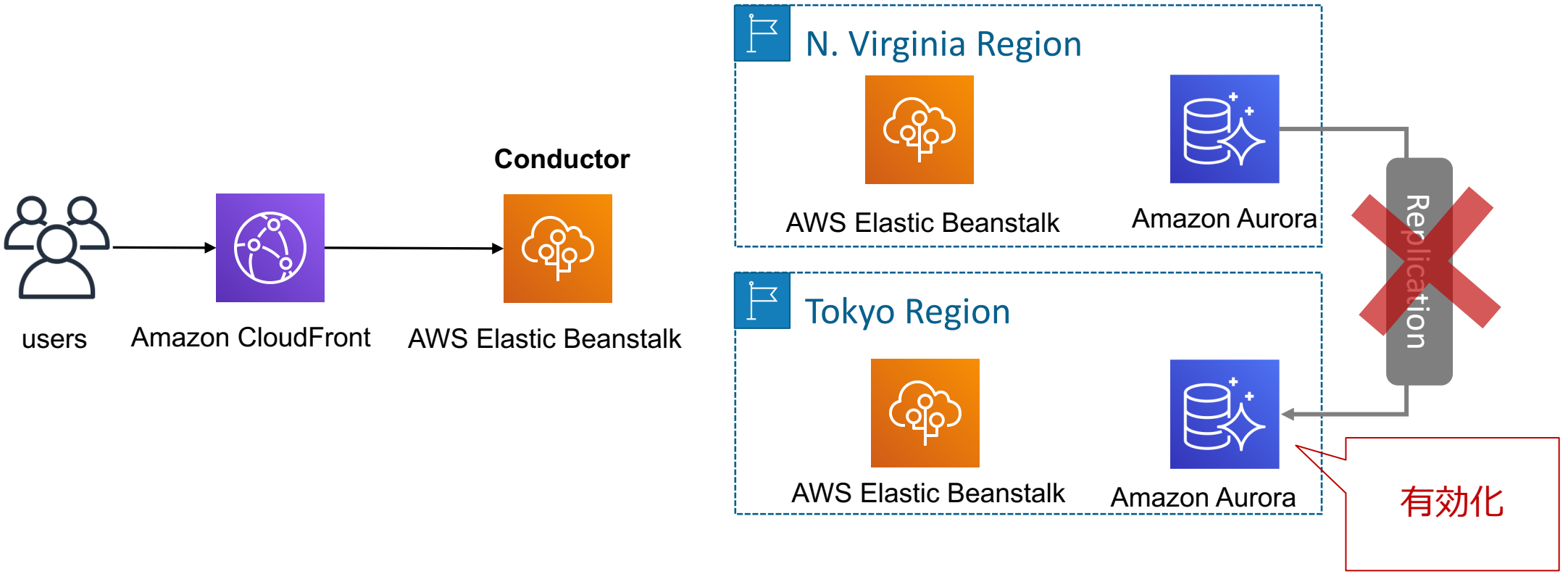
コンダクターの実装 #2

コンダクターの導入 → **サスペンドモード** → DBの移行 → トラフィックの切り替え → コンダクターを外す



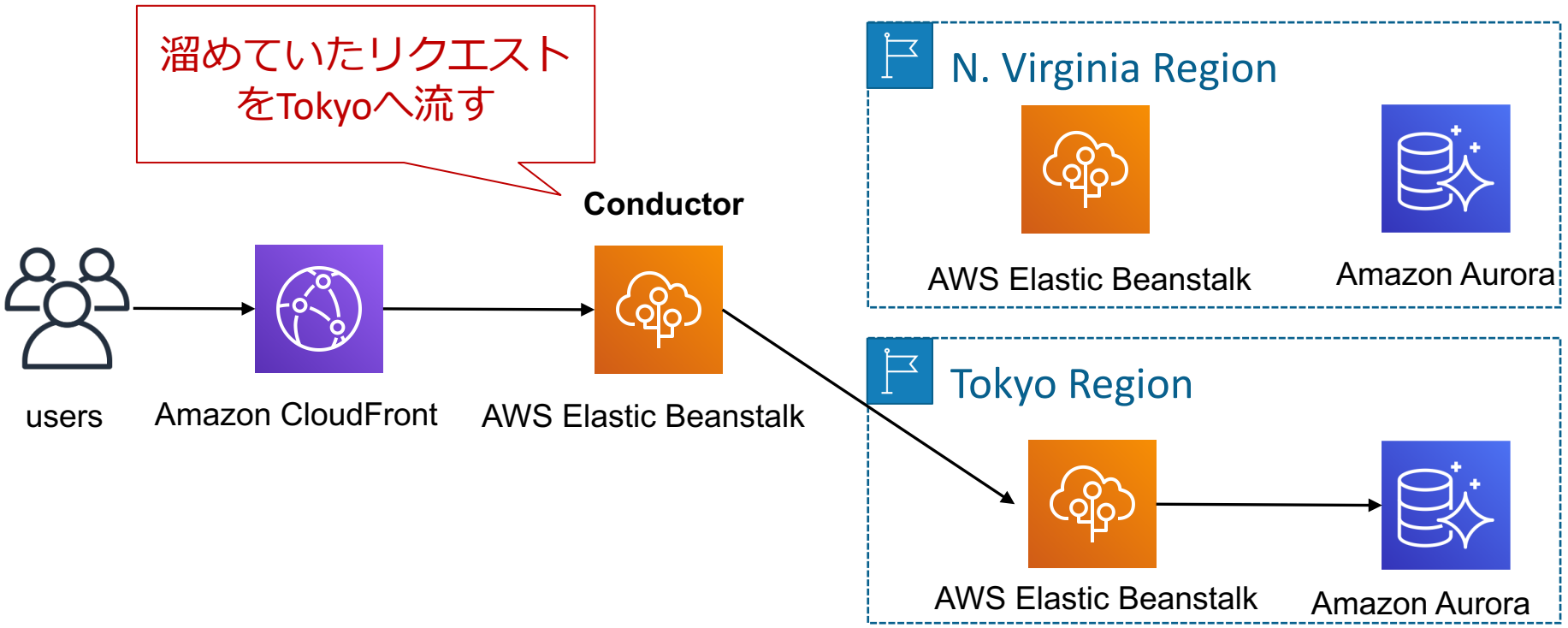
コンダクターの実装 #3

コンダクターの導入 → サスペンドモード → **DBの移行** → トラフィックの切り替え → コンダクターを外す



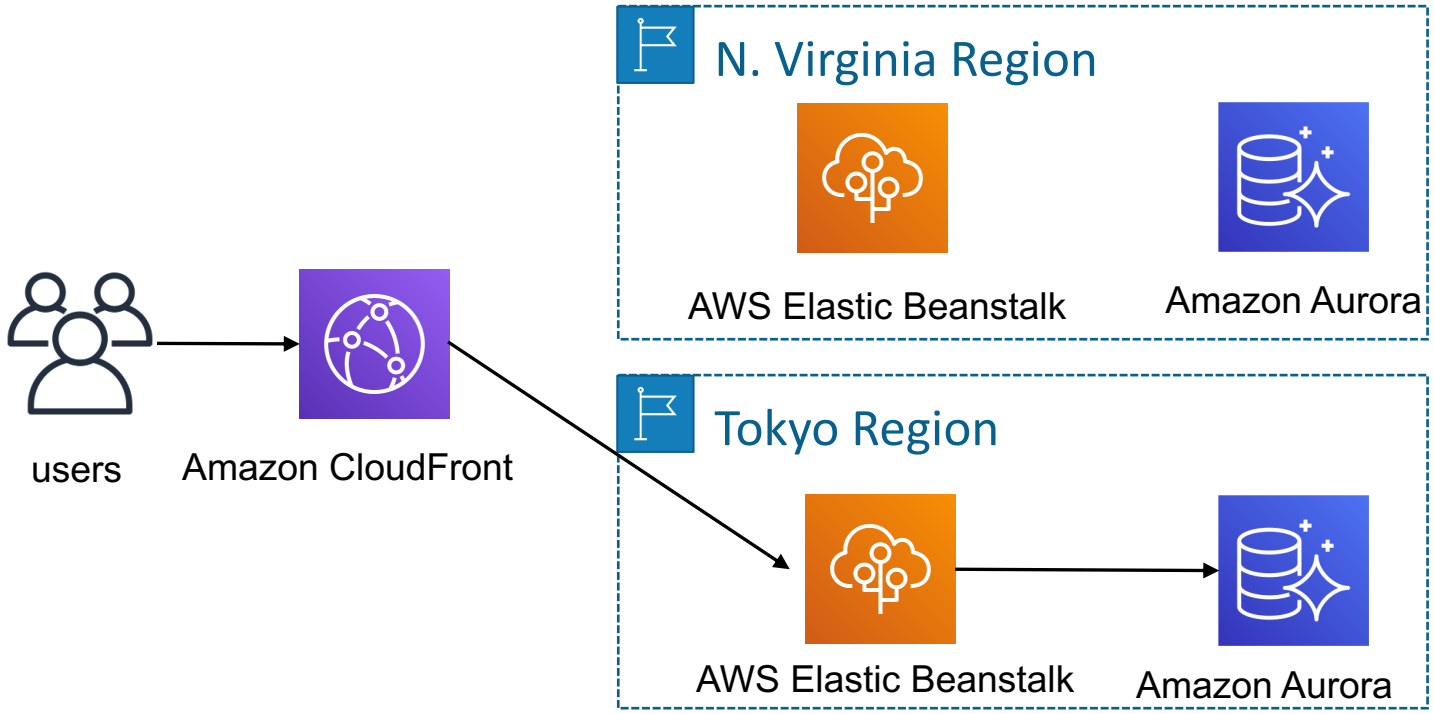
コンダクターの実装 #4

コンダクターの導入 → サスペンドモード → DBの移行 → **トラフィックの切り替え** → コンダクターを外す

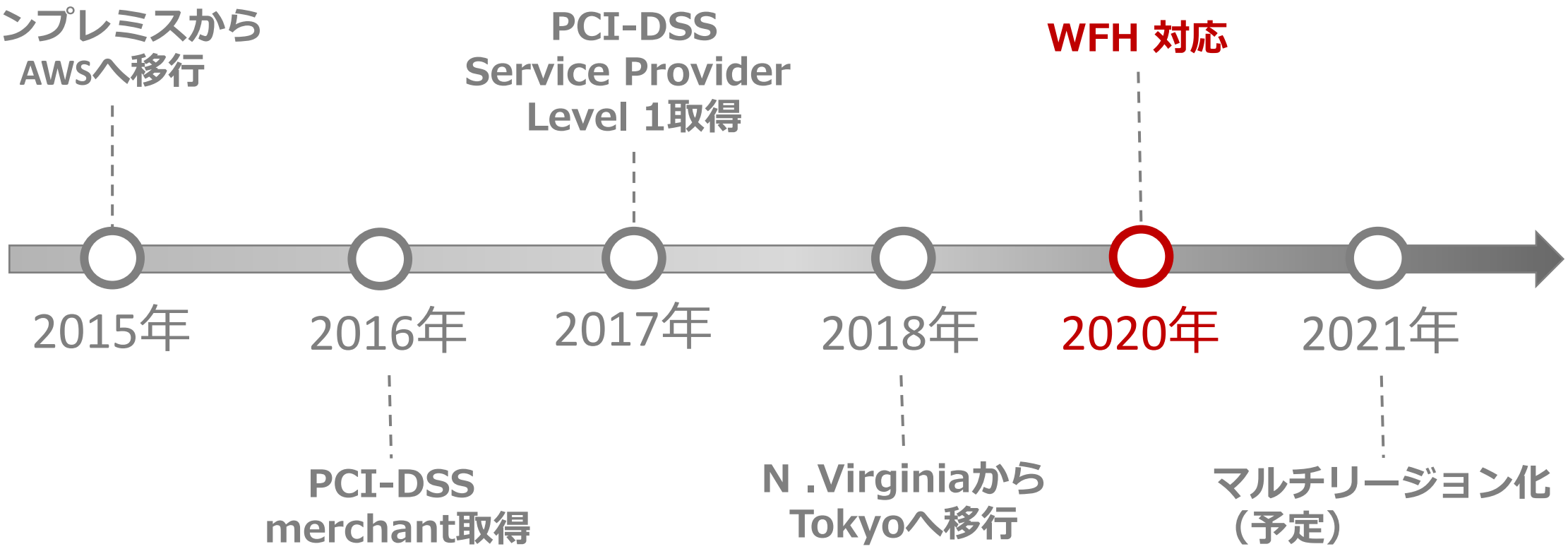


コンダクターの実装 #5

コンダクターの導入 → サスペンドモード → DBの移行 → トラフィックの切り替え → **コンダクターを外す**

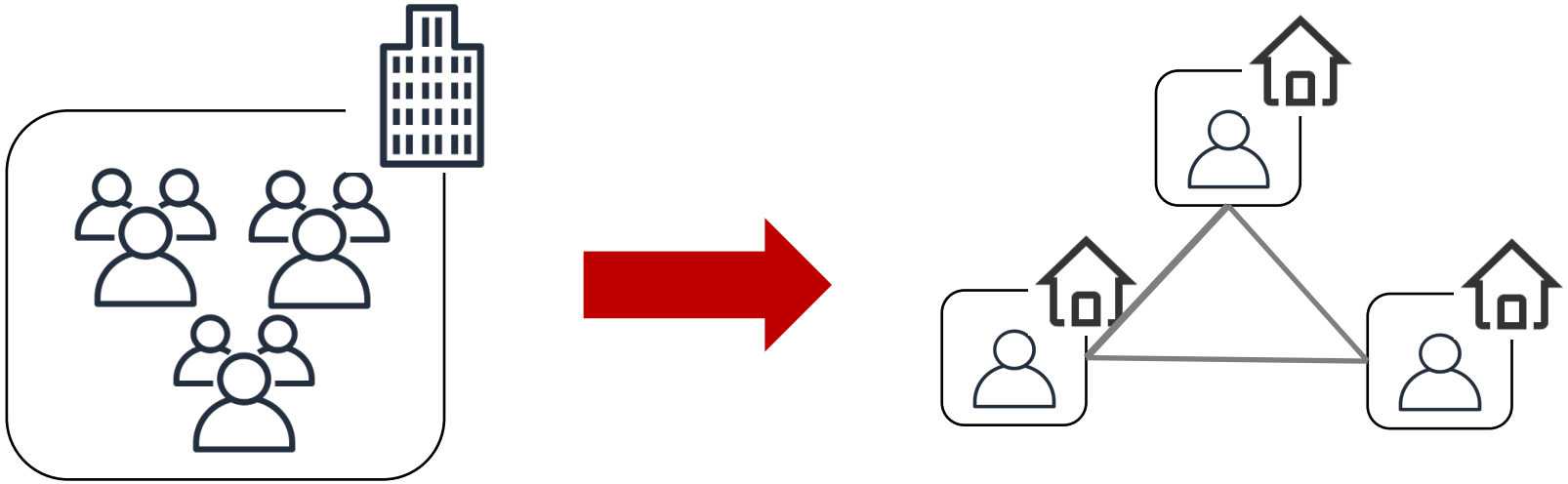


2020年


オンプレミスから
AWSへ移行
PCI-DSS
Service Provider
Level 1取得
WFH 対応
2015年
2016年
2017年
2018年
2020年
2021年
PCI-DSS
merchant取得
N .Virginiaから
Tokyoへ移行
マルチリージョン化
（予定）

背景

新型コロナウィルス感染症対策のため、2020年2月から段階的に原則在宅勤務へ



- 常に周囲に複数の第三者がいる状態
- 本番作業の立会い体制も万全

- 周囲に会社関係者はおらず独立した環境
- 本番作業の立会いはリモートで実施するしかない
- 社内ルールの規制緩和（モバイルPCの持ち帰りなど）

要件と課題

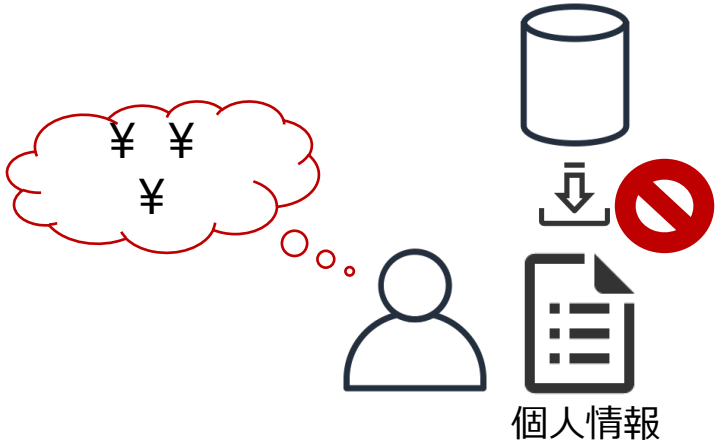
要件　**WFHでのリスクを減らす**

課題

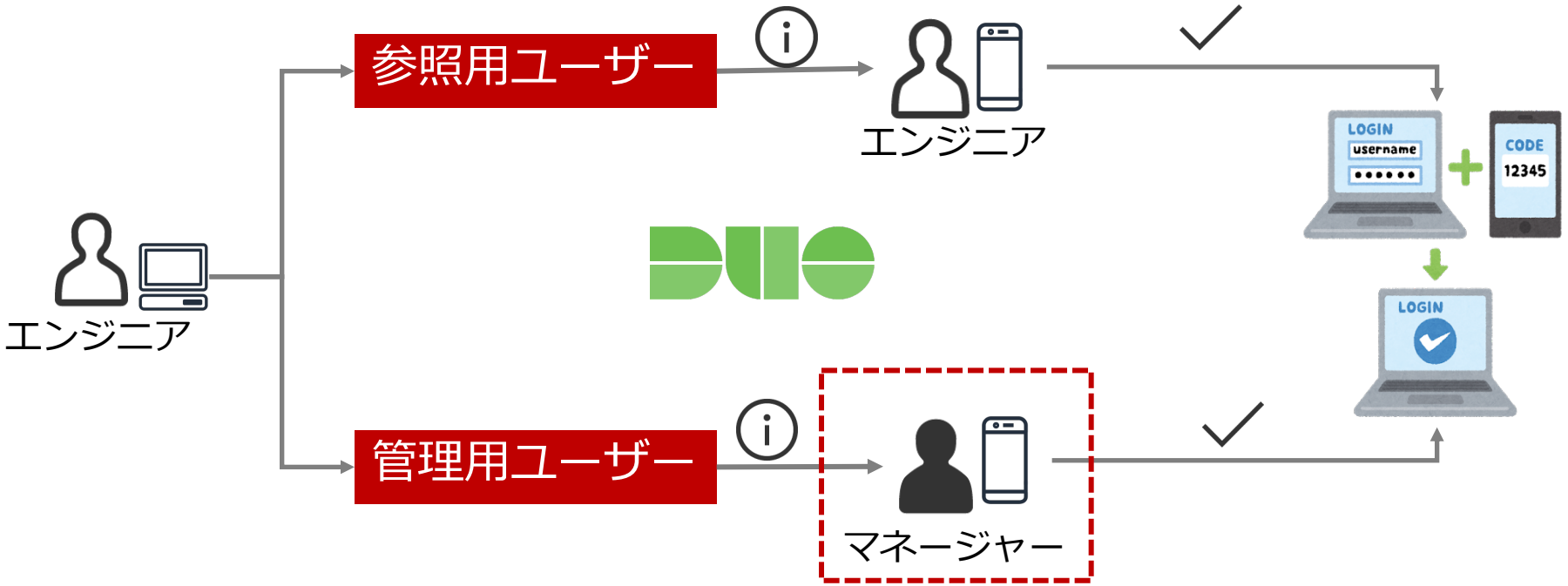
オペレーションミスによるサービス停止

- 環境間違えによる予期せぬ修正
- 本番環境を確認するつもりだけだったのにまちがって更新をしてしまう

予期せぬデータ漏洩

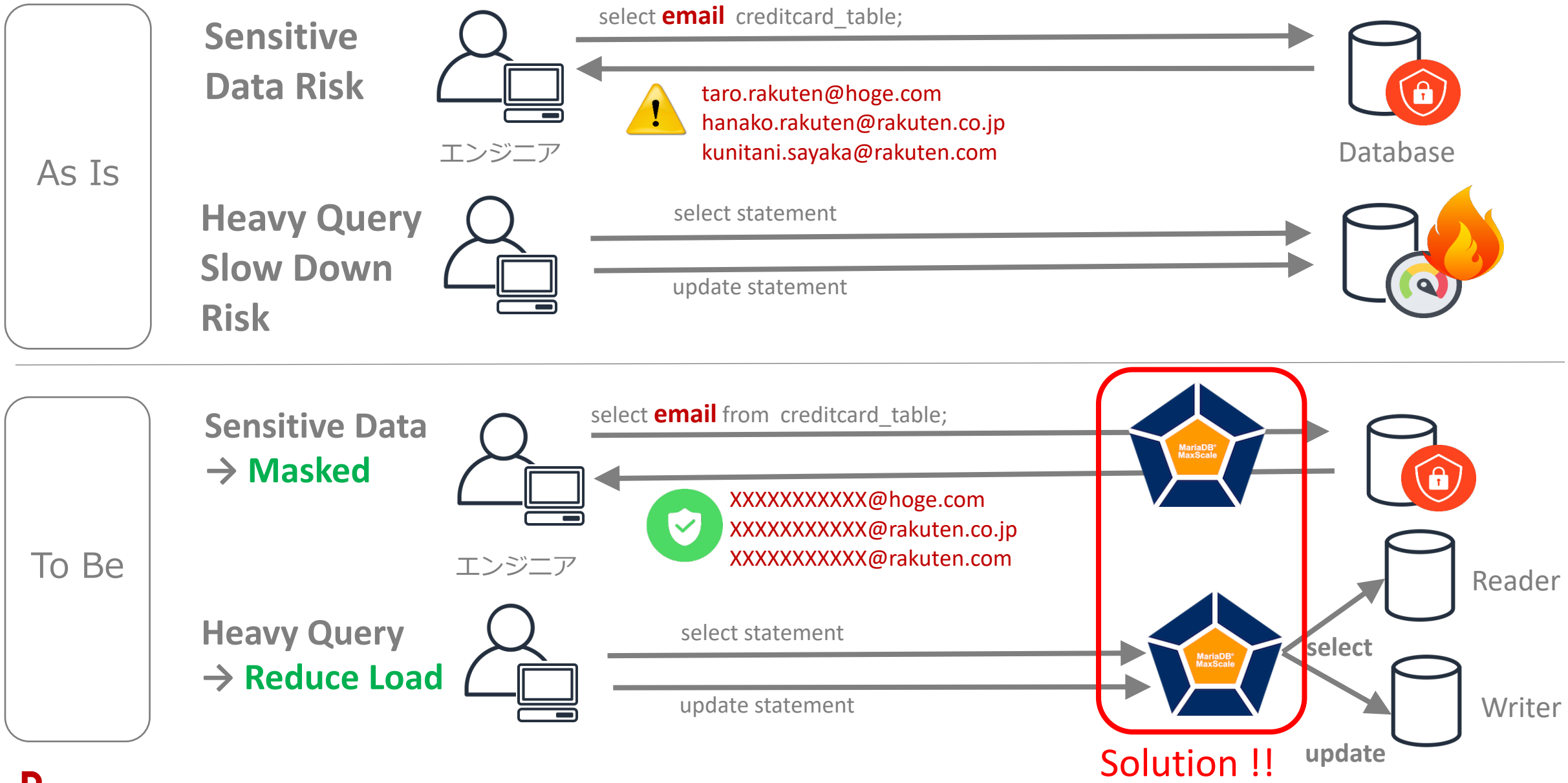


解決策 – 参照用ユーザーと管理用ユーザーの分離

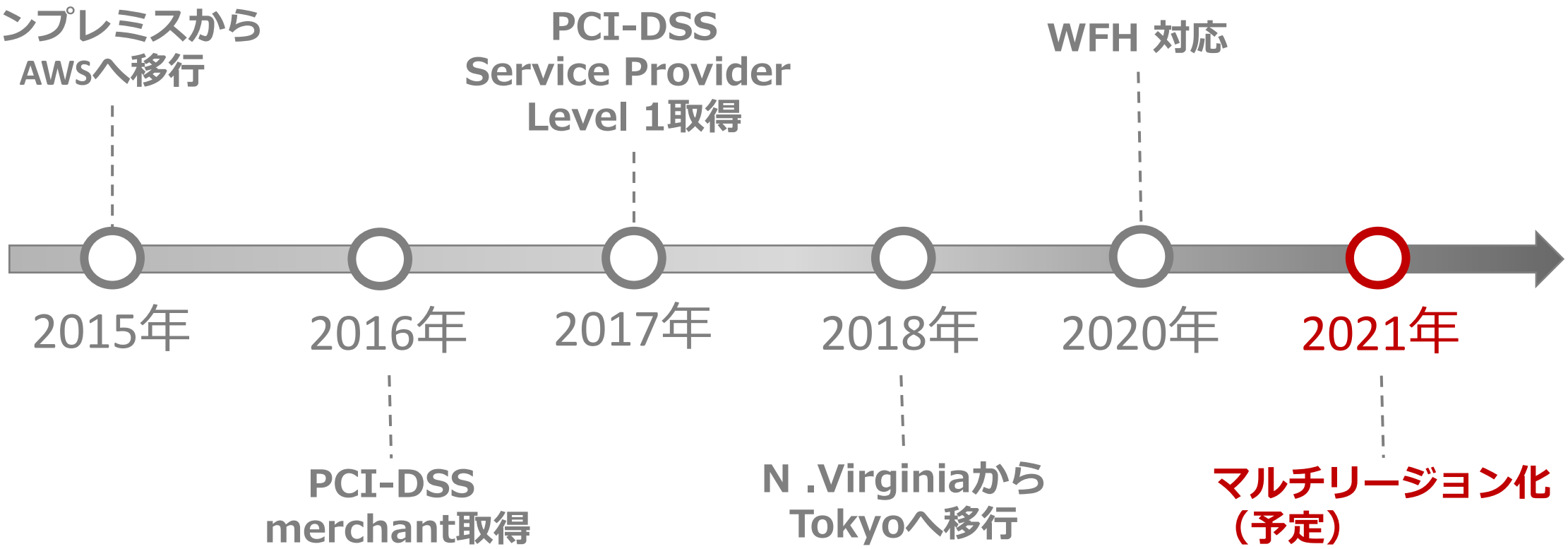


	管理用ユーザー	参照用ユーザー
Root userへの切り替え	✓	✕
サービスステータスの確認	✓	✓
サービスの起動/停止	✓	✕
ファイルの参照	✓	✓
ファイルの更新	✓	✕

解決策 – プロキシによる個人情報マスキングとクエリの自動振り分け



2021年

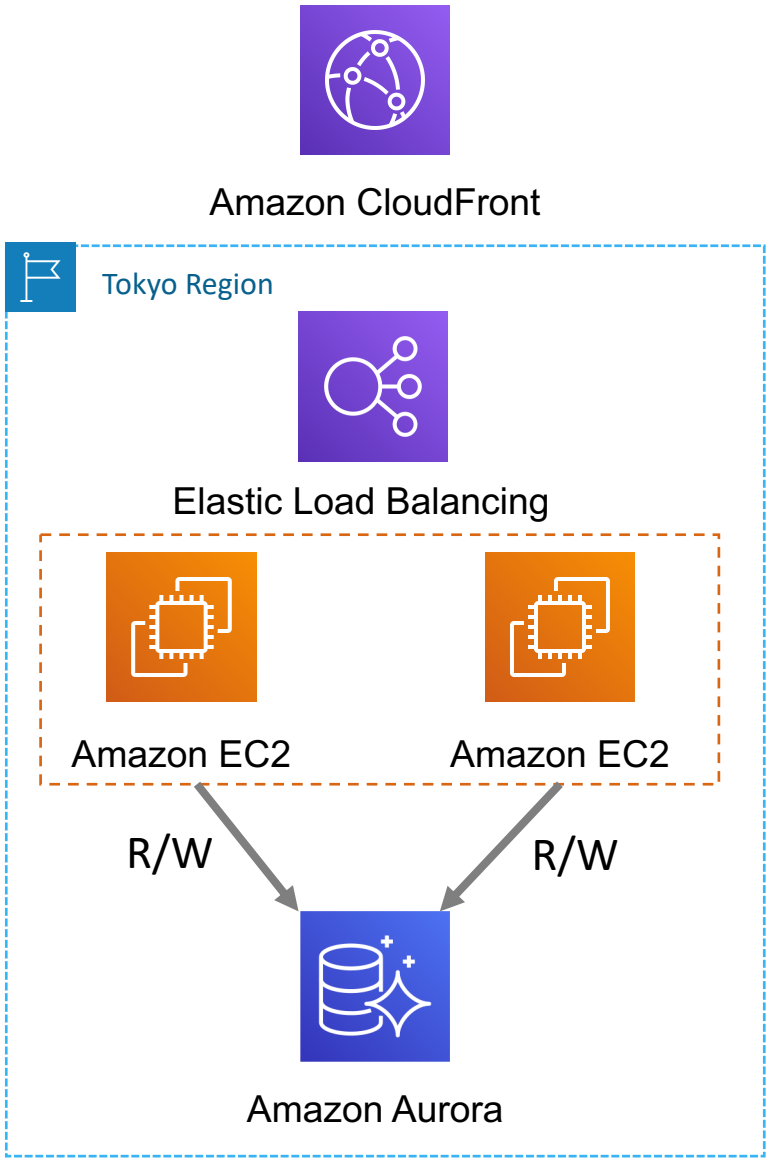

オンプレミスから
AWSへ移行
PCI-DSS
Service Provider
Level 1取得
WFH 対応
2015年
2016年
2017年
2018年
2020年
2021年
PCI-DSS
merchant取得
N .Virginiaから
Tokyoへ移行
マルチリージョン化
（予定）

要件

- **Global展開**
  - 世界中に点在する楽天グループの各サービスそれぞれに対して、低レイテンシを実現する。
  - カード情報が全てのリージョンで同期され、どの海外サービスからでもそのカード情報を利用可能にする。

- **BCP**
  リージョン災害時の切り替えを可能にし、災害時のF/Oのダウンタイムを限りなくゼロにする。

現状



システム構成

- **Tokyo region のみ利用**
- **3Tier構成のシステム**
  - Web-Appレイヤには、Elastic Beanstalkを利用
  - RDBにAurora MySQL 5.7を利用
    （AuroraのWriterとReaderは１台ずつで、
    決済処理のR/Wすべてを Writerに向けている。
    ReaderはF/O用途）

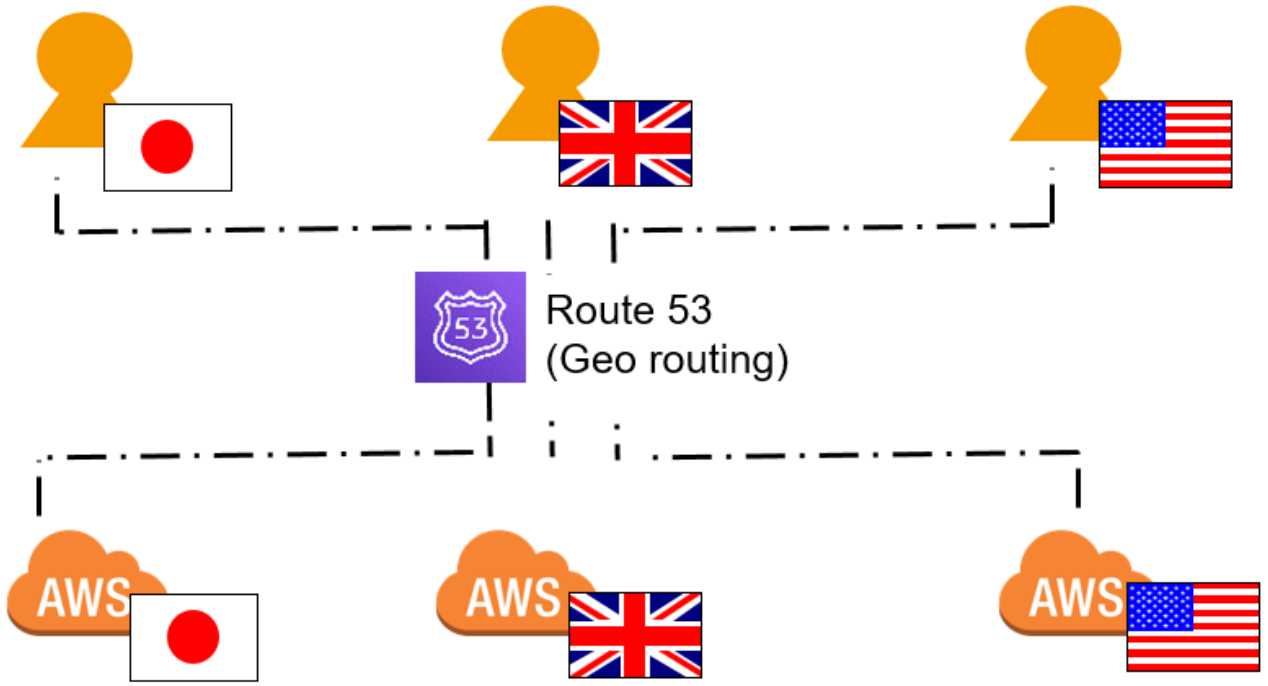
課題

- **クエリーのレイテンシーの改善**
  - エンドユーザーやサービスに地理的に近いロケーションにシステムが必要とされる 。
- **日本、ヨーロッパ、アメリカの3リージョンで構成**
  - メインターゲットのサービスがこの３拠点に存在している
- **リージョン単位でのデータを分割ができない**
  - あるリージョンで登録したクレジットカードは、海外からでも利用できることが要件のため分割できない。
- **ゼロダウンタイムを目指す**
  - F/Oなどによるサービス影響をユーザーに感じさせないレベルまで極小化したい。

解決策 – Route 53 Geo routing

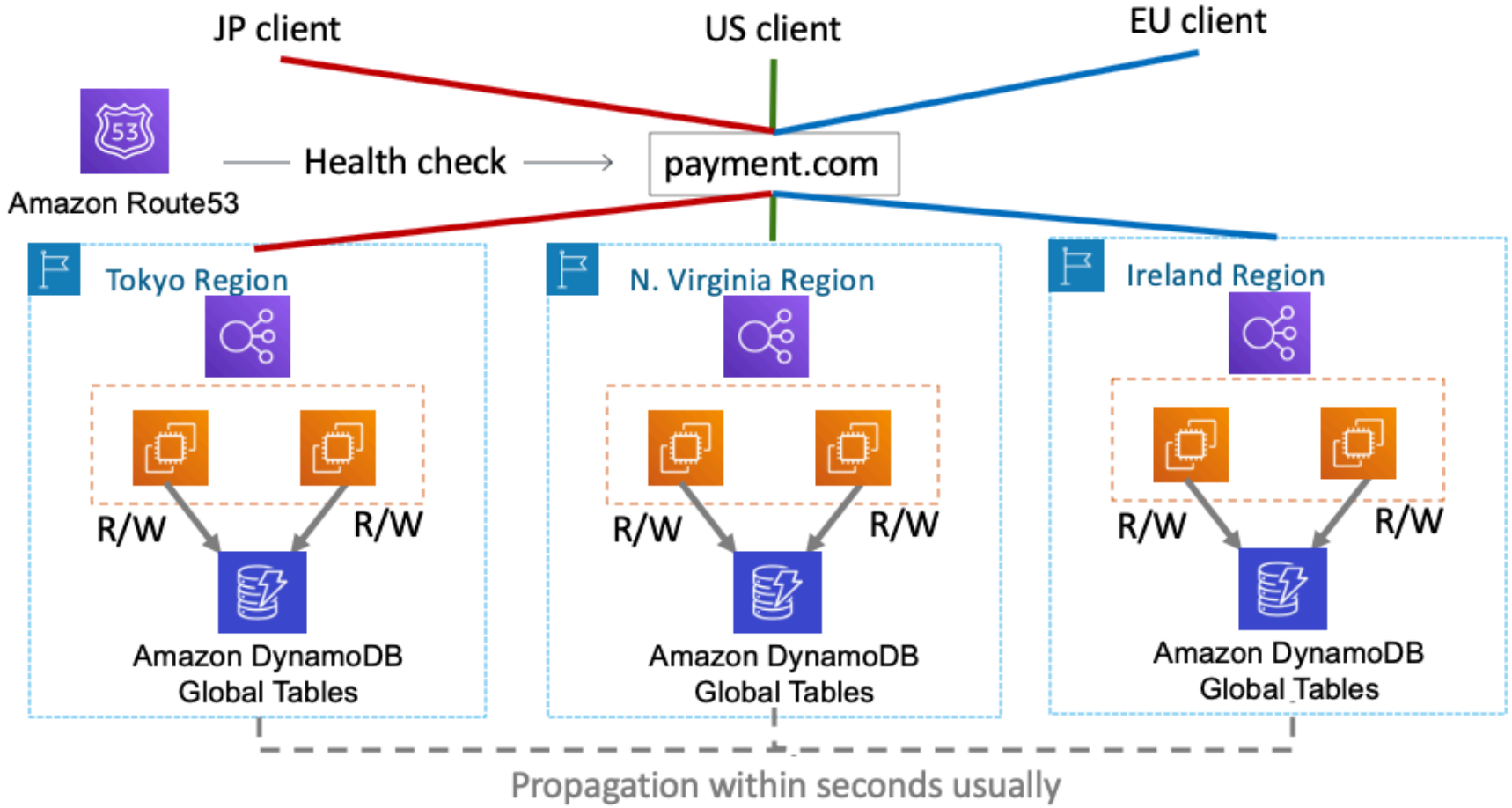
- 接続元の位置情報に基づいてトラフィックをルーティングする機能
- アプリケーションレイヤーが地理的位置を意識する必要がない。

-> **クライアントから最も近いリージョンに自動的にアクセスさせ、レイテンシーの課題をクリア**



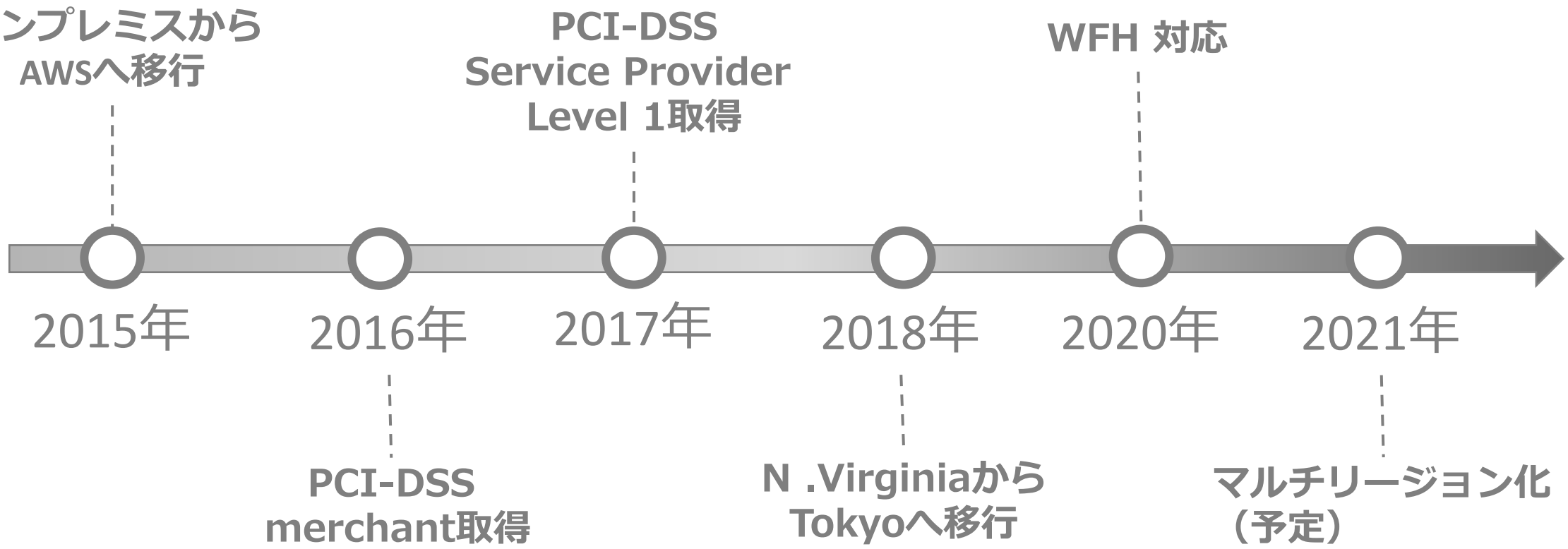
解決策 – Amazon Dynamo DB Global Tablesを利用したR/L-W/L 構成

- **3リージョン**を利用
- データベースは**DynamoDB**を利用し、各リージョンに配置
- リージョン間のデータプロパゲーションはDynamoDB streamsによって**双方向に同期**
- **READ LOCAL - WRITE LOCAL** (ReadとWriteの両方のプロセスは各リージョンのローカルで実行)



まとめ

振り返り


オンプレミスから
AWSへ移行
PCI-DSS
Service Provider
Level 1取得
WFH 対応
2015年
2016年
2017年
2018年
2020年
2021年
PCI-DSS
merchant取得
N .Virginiaから
Tokyoへ移行
マルチリージョン化
（予定）

テクノロジーと工夫の融合

テクノロジーに頼るのではなく、テクノロジーに工夫を融合して課題を解決してきました。
今後もこのようにして、より良いシステムを作りを目指します。

楽天株式会社
國谷 彩 (Sayaka Kunitani)