

SOAに基づく クラウド間連携アーキテクチャの提案

南山大学 数理情報学部 情報通信学科
小島弘誉 岩本勝也 大倉奨貴 中道上 青山幹雄

シナリオ

背景と問題点

アプローチ

提案方法と適用技術

クラウドブローカーアーキテクチャ

プロトタイプ

評価と考察

今後の課題

まとめ

背景と問題点

➤ 背景: クラウドコンピューティングの普及

□ 業務システムをクラウドに移行する企業が増加

➤ 問題点: クラウド-オンプレミス間やクラウド間の連携が未確立

□ 現状のクラウド間連携方法

✓ 振舞いと型を一括で整合

⇒ 組み合わせ毎にコネクタが必要

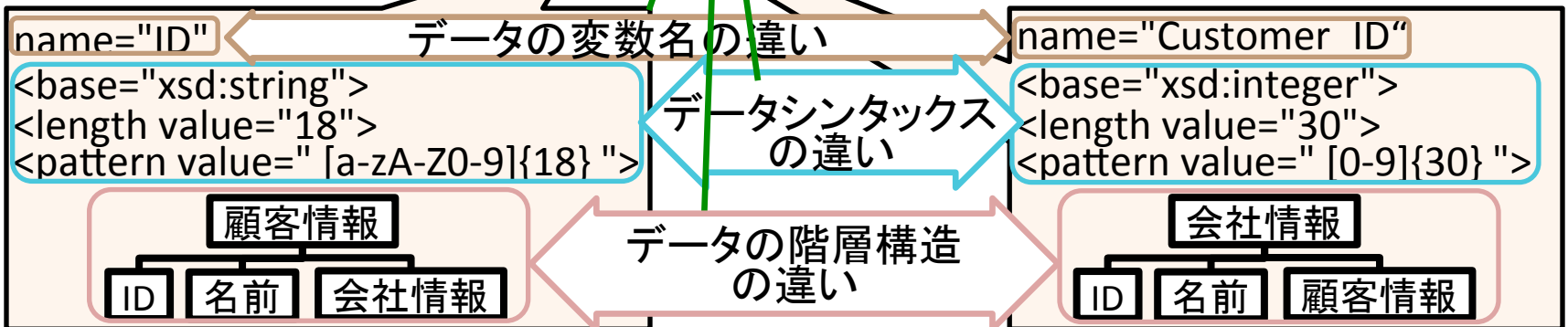
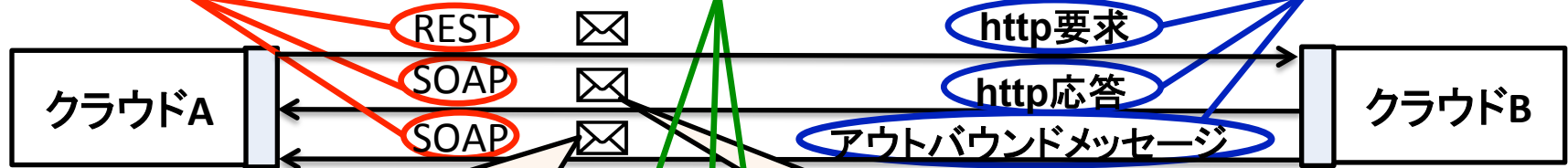


➤ 研究課題: 異なるメッセージの振舞いと型の整合

(1) メッセージプロトコル

(2) メッセージの型

(3) メッセージ交換のタイミング



アプローチ

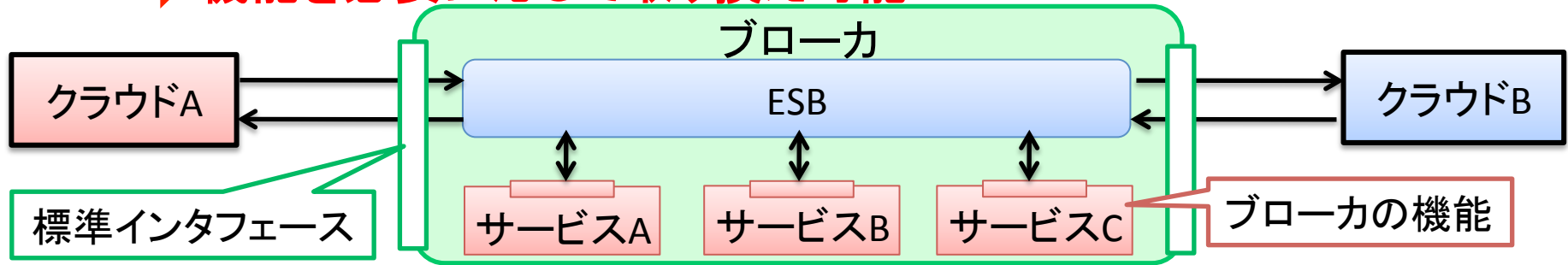
1. SOA (Service-Oriented Architecture)の適用[1]

□クラウド間が独立

⇒クラウド間の疎結合な連携

□ブローカの機能のサービス化

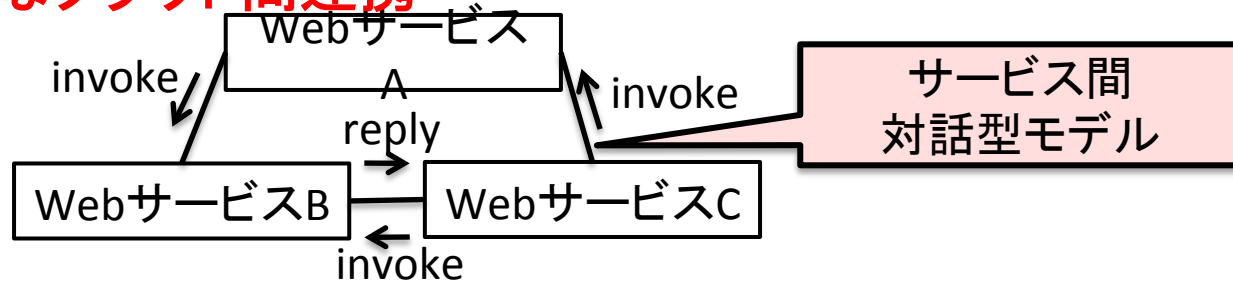
⇒機能を必要に応じて取り換え可能



2. コレオグラフィ連携モデルの適用[2]

□集中制御を持たない連携

⇒疎結合なクラウド間連携



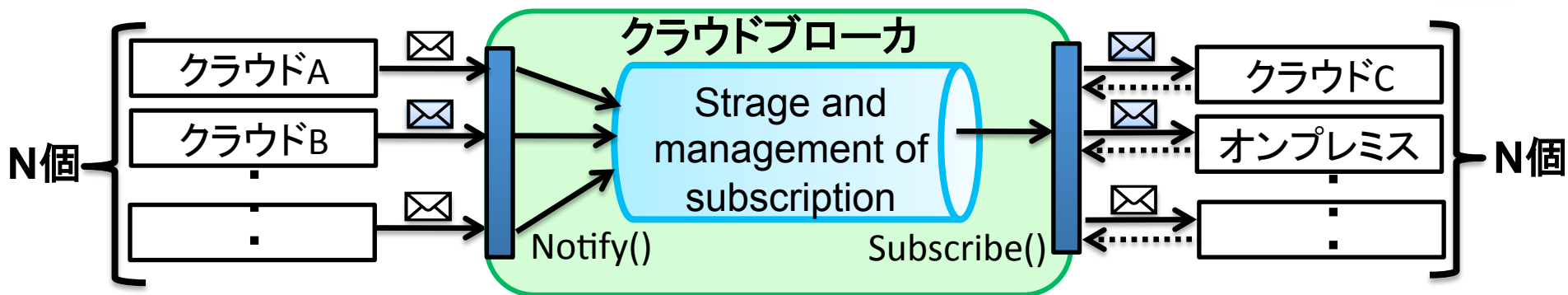
[1] D. K. Krafcig, et al., Enterprise SOA, Prentice Hall, 2005[山下 真澄 (監訳), SOA大全, 日経BP社].

[2] M. B. Juric, Business Process Execution Language for Web Services, 2nd ed., PACKT, 2006.

提案方法と適用技術

➤ 提案方法: 任意のクラウド間N対N連携アーキテクチャ

- 連携トポロジ: N対N
- クラウド間の疎結合な連携



➤ 適用技術

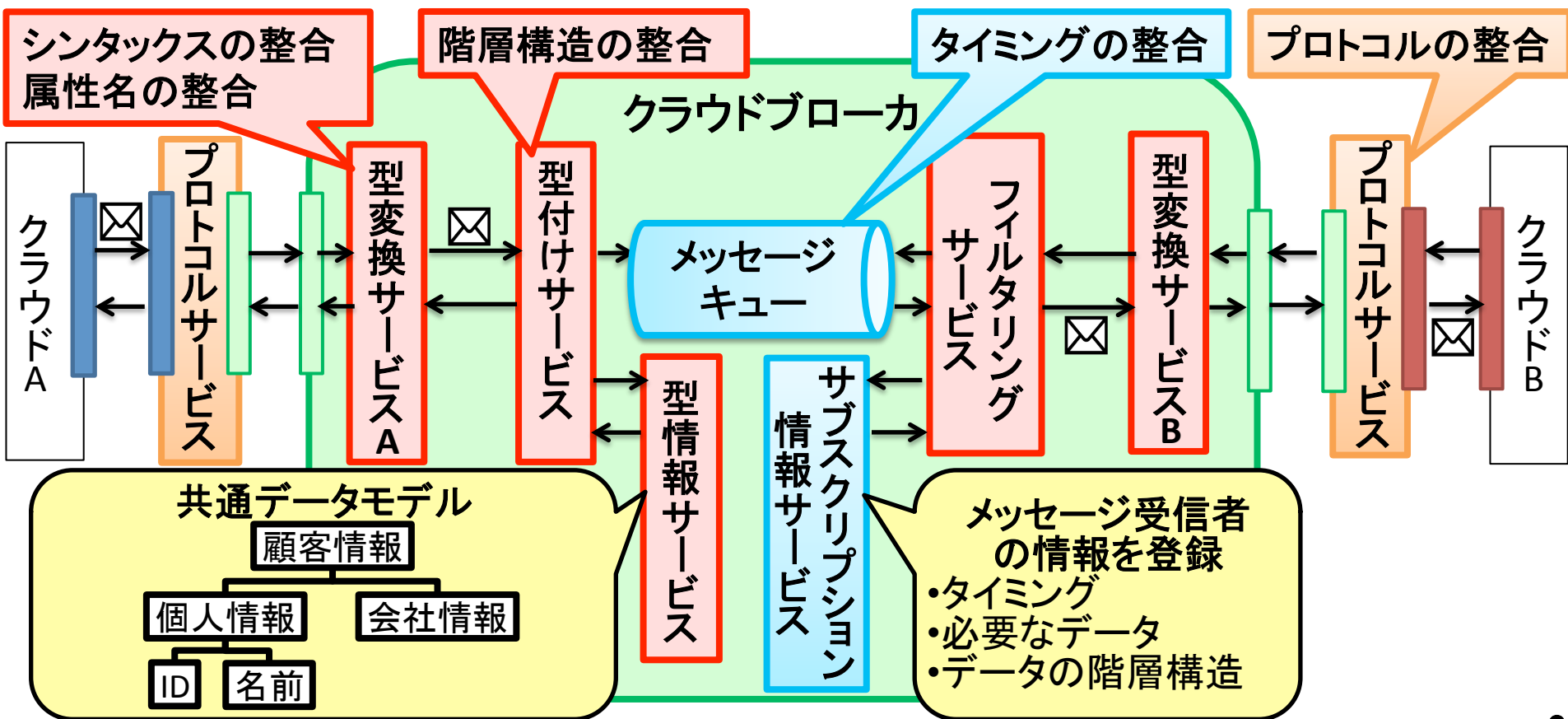
- パブリッシュ/サブスクライブアーキテクチャ
 - ✓ 適用理由: 三つの非依存性(同期的分離, 時間的分離, 空間的分離)
 - ⇒ クラウド間の疎結合な連携を実現
- タイプベースフィルタリング
 - ✓ 適用理由: 静的にセマンティクス, 動的にシンタックスのフィルタリング
 - ⇒ 不特定多数の型の整合

クラウドブローカーアーキテクチャ(1/4)

振舞いと型の整合

➤ 整合方法

- ✓ 振る舞いと型を分離
- ✓ データセマンティクスを属性名と階層構造に分離
- ✓ クラウド間の共通データモデルを定義

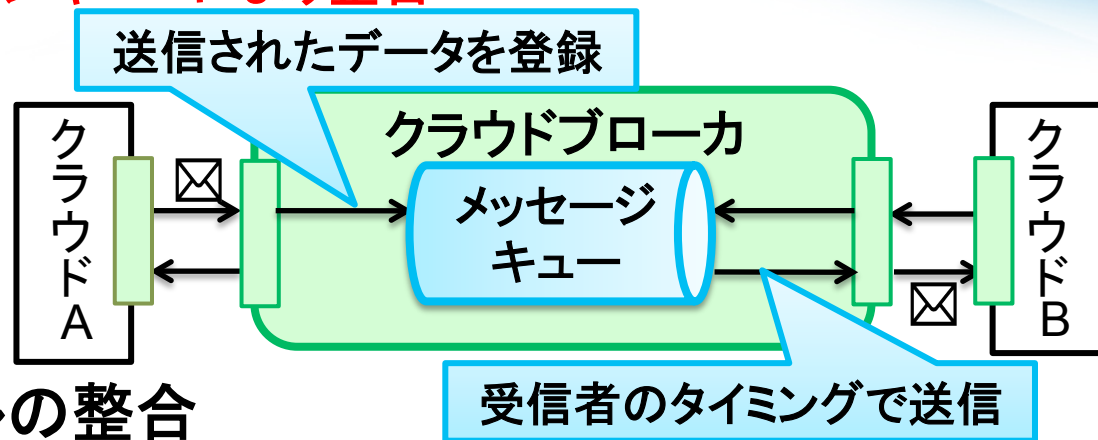


クラウドブローカーアーキテクチャ(2/4)

タイミングとプロトコルの整合

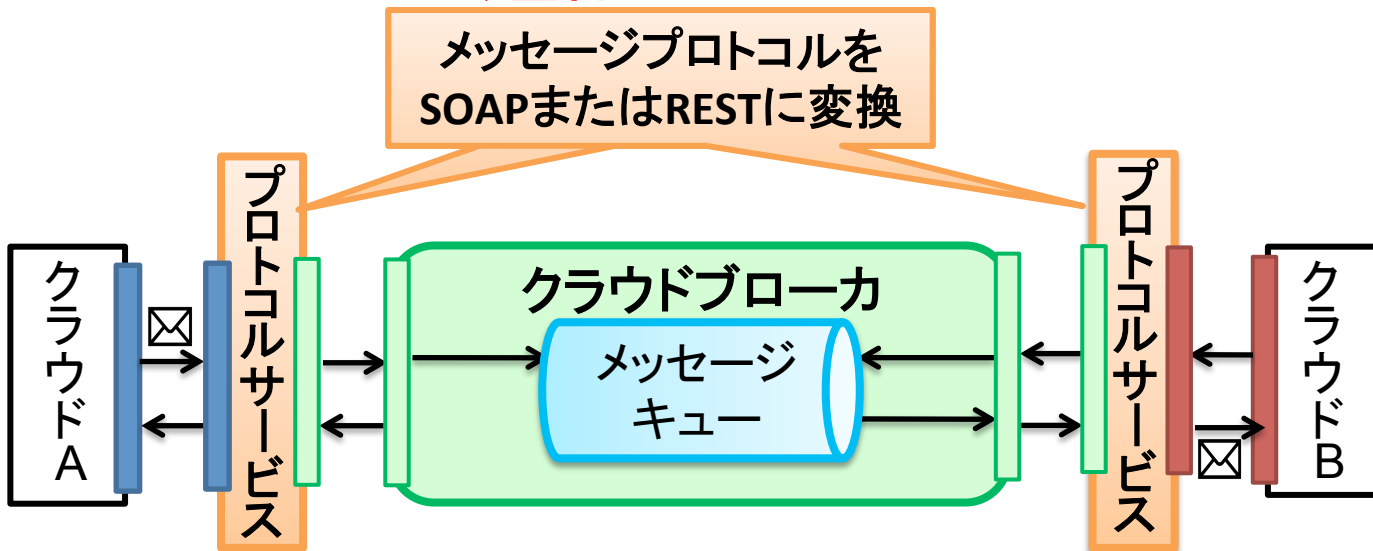
➤ タイミングの整合

⇒ メッセージキューにより整合



➤ プロトコルの整合

⇒ プロトコルサービスにより整合

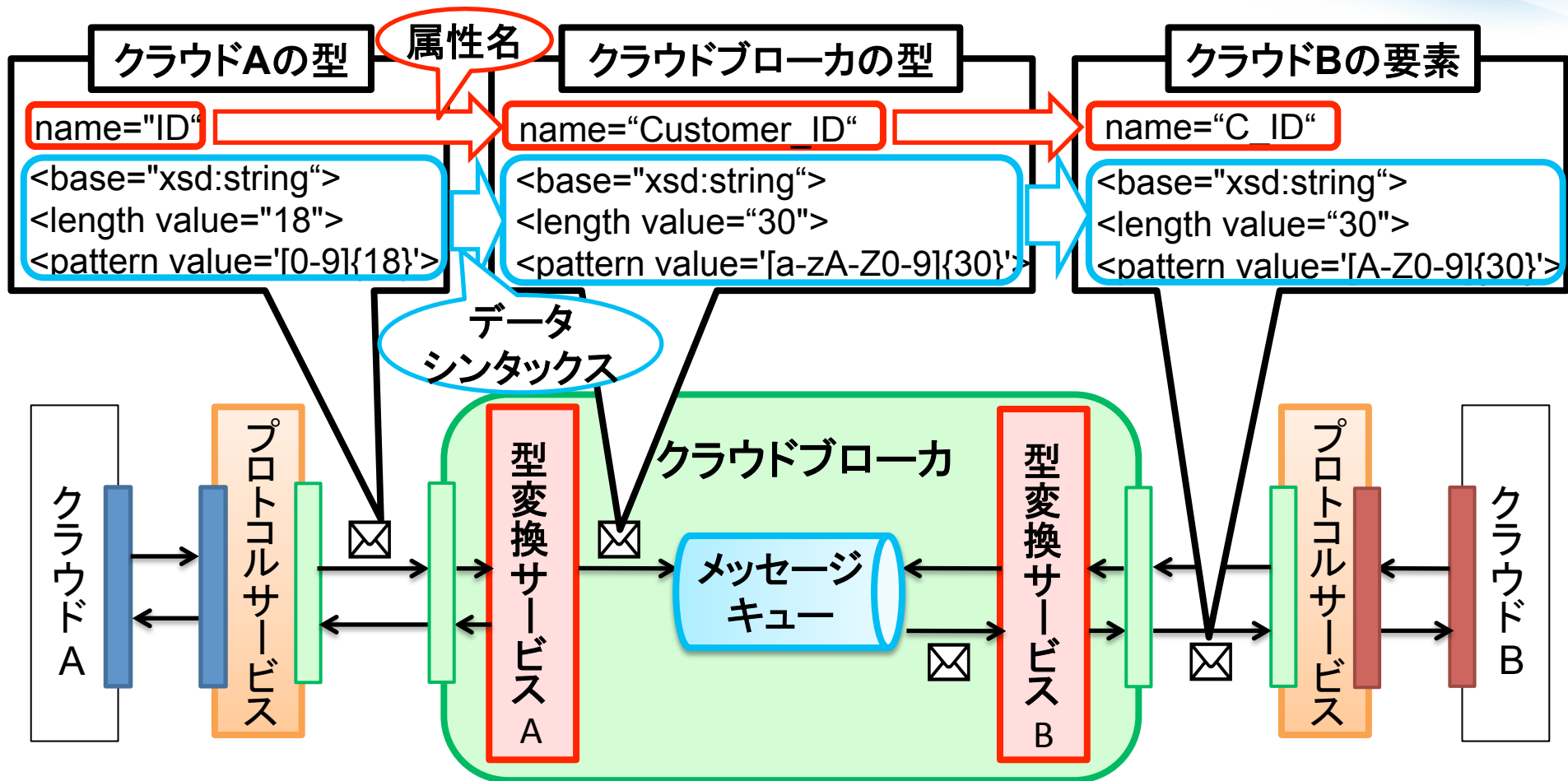


クラウドブローカーアーキテクチャ(3/4)

型の整合

➤ データシンタックス, 属性名の整合

⇒ 型変換サービスにより整合

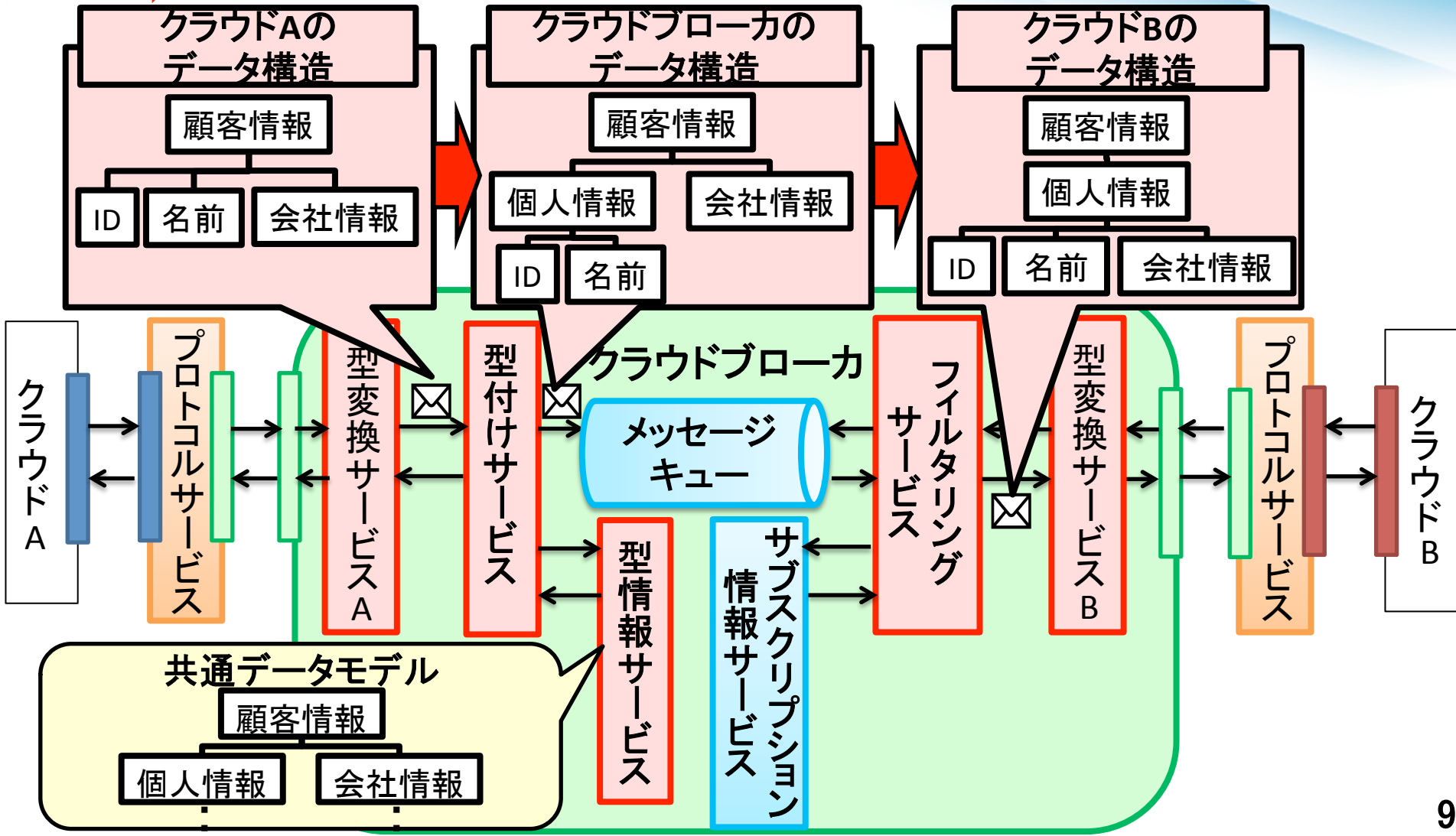


クラウドブローカーアーキテクチャ(4/4)

型の整合

データの階層構造の整合

型付けサービス, フィルタリングサービスにより整合



プロトタイプ(1/4)

プロトタイプの構成

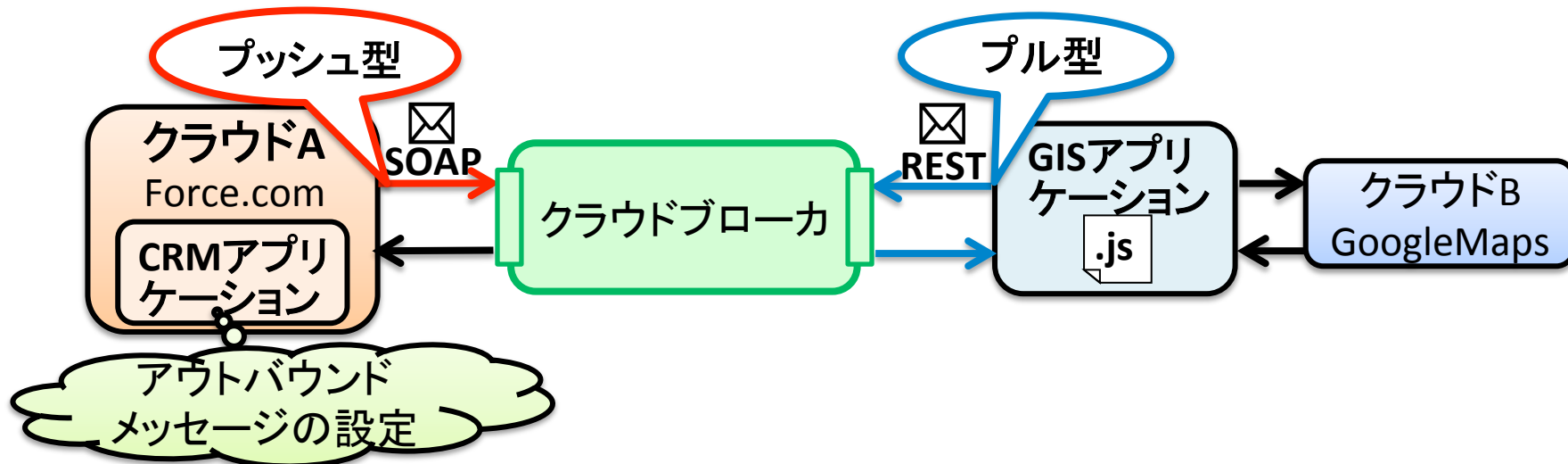
➤ 目的: 振舞いと型の整合の妥当性確認

➤ 整合対象とするクラウド

- ✓ Force.com: Salesforce.com社が提供するPaaS
- ✓ GoogleMaps: Google社が提供するSaaS

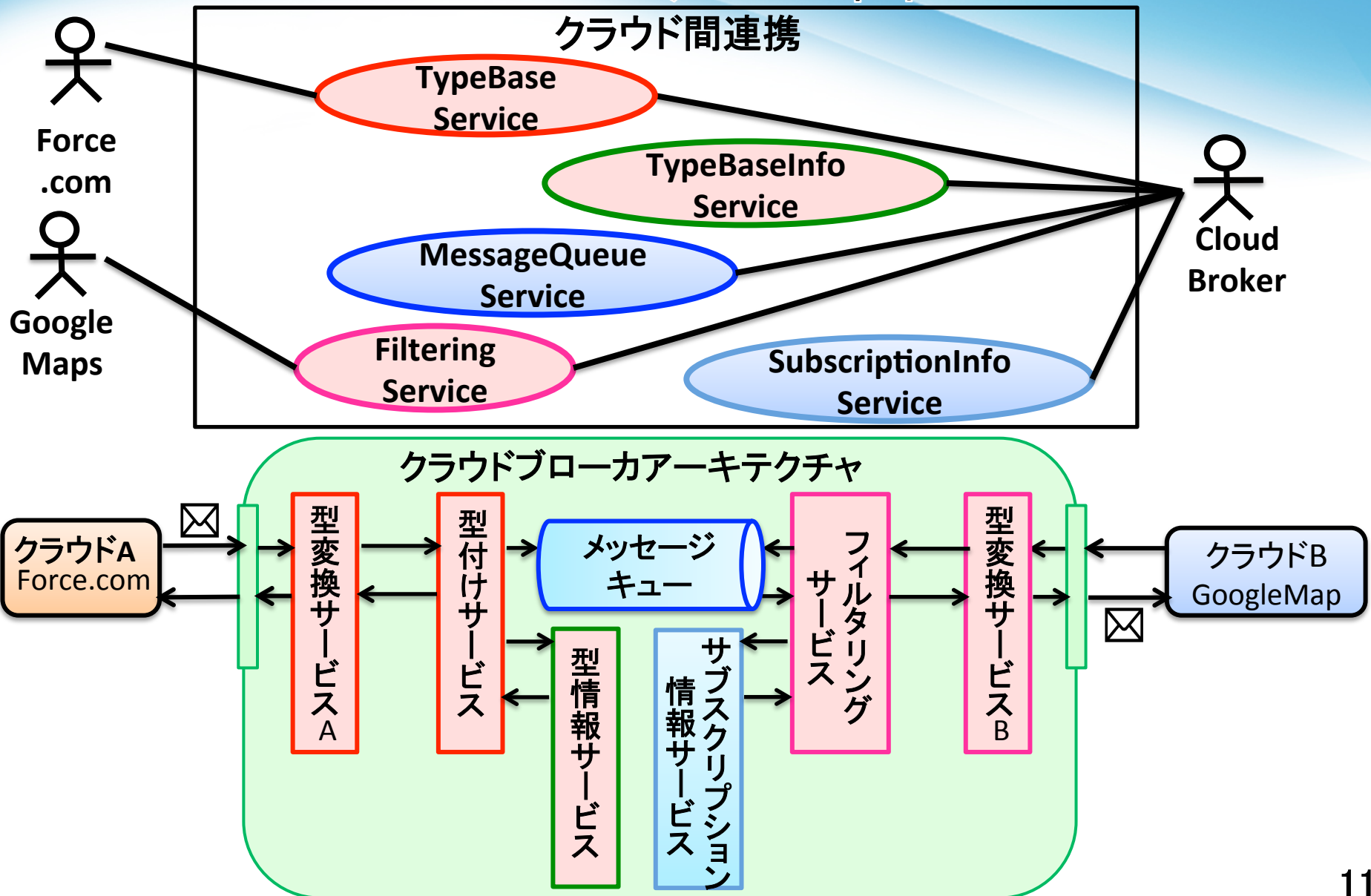
➤ クラウド間の振舞いの違い

- ✓ プッシュ型: 自立的にメッセージを送信
- ✓ プル型: ユーザの要求に応じたメッセージ交換

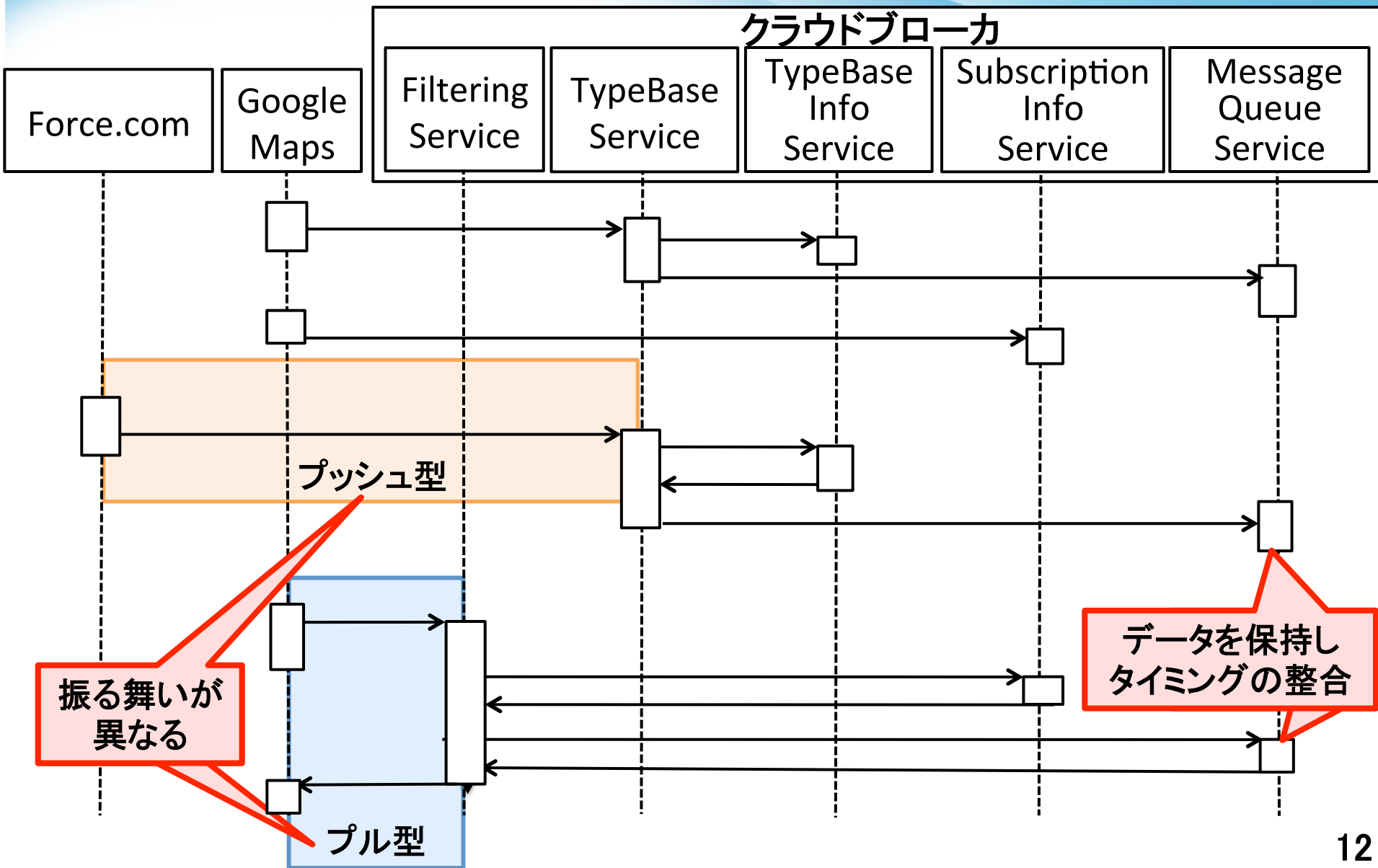


プロトタイプ(2/4)

ユースケース図



プロトタイプ(3/4) シーケンス図



プロトタイプ(4/4)

プロトタイプの実装

➤クラウドブローカ

□プラットフォーム: ServiceMix(オープンソースESB)

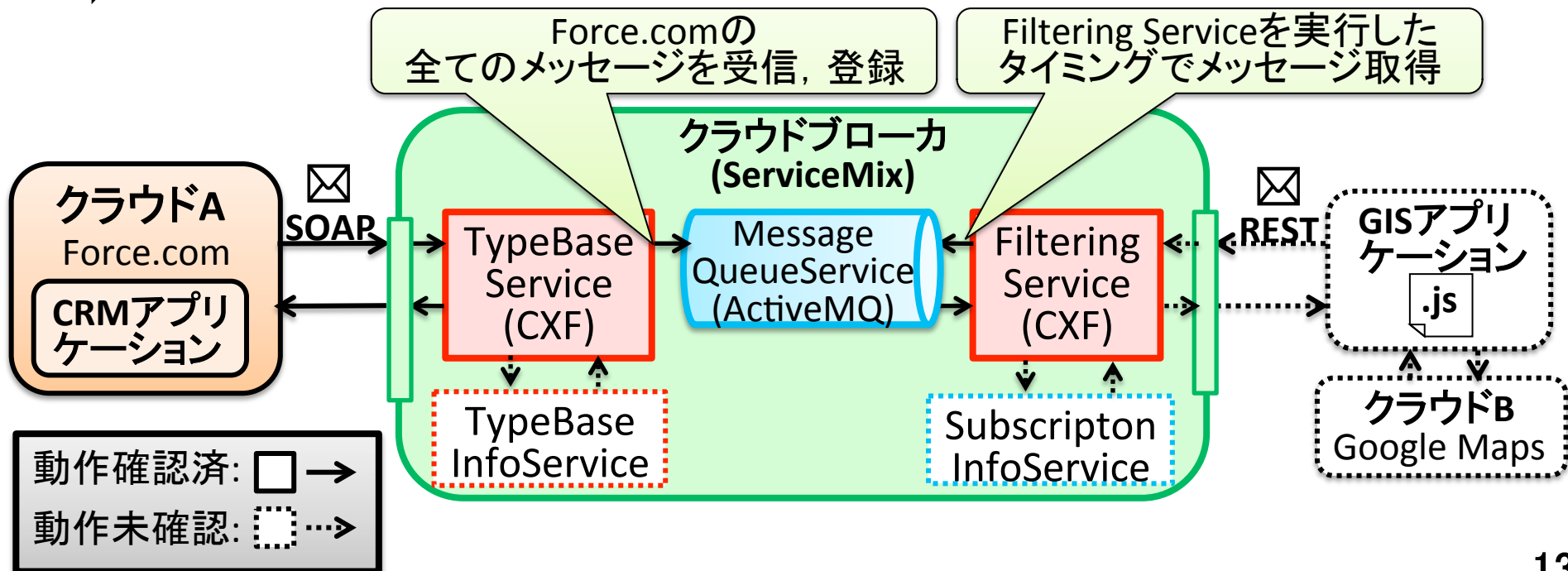
□コンポーネント

✓ Apache CXF(Web Service)

✓ Apache ActiveMQ(Message Queue)

➤アーキテクチャ実現可能性の確認

⇒ タイミングの整合を確認



評価と考察

□プロトタイプによる考察

✓アーキテクチャ実現可能性

振舞いの整合をとるプロトタイプ作成

⇒タイミングの整合の確認

□現状の連携方法との比較

✓整合方法

型と振舞いに分離

型をセマンティクスとシンタックスに分離

⇒必要に応じてサービスの組み替え可能な連携

✓連携トポロジ

N対Nの連携

⇒共通データモデルの型に変換するのみで

不特定多数の連携を実現

今後の課題

□ 型の整合の拡張

- ✓ 共通データモデル定義
- ✓ 整合方法の具体化

□ プロトタイプの拡張

- ✓ クラウドブローカ内のサービスを拡張

□ クラウドブローカのオーバーヘッドを計測

- ✓ 連携アーキテクチャとしての妥当性の確認

まとめ

研究課題

- クラウド毎に振舞いと型が異なるため連携が困難
 - ✓ メッセージプロトコル
 - ✓ メッセージ交換のタイミング
 - ✓ メッセージの型

提案アーキテクチャ

- クラウド間N対N連携アーキテクチャの提案
 - ✓ 連携トポロジ: N対N
 - ✓ 型と振舞いを分離して整合
 - ✓ 型をセマンティクスとシンタックスに分離して整合

評価と考察

- プロトタイプによる評価
 - ✓ タイミングの整合を確認
- 従来の連携方法との比較
 - ✓ 振る舞いと型(シンタックス, セマンティクス)に分離して整合

END

ありがとうございました