



Linked Dataを用いた クラウドサービス連携アーキテクチャ

M2011MM041

小島 弘誉

+ 概要

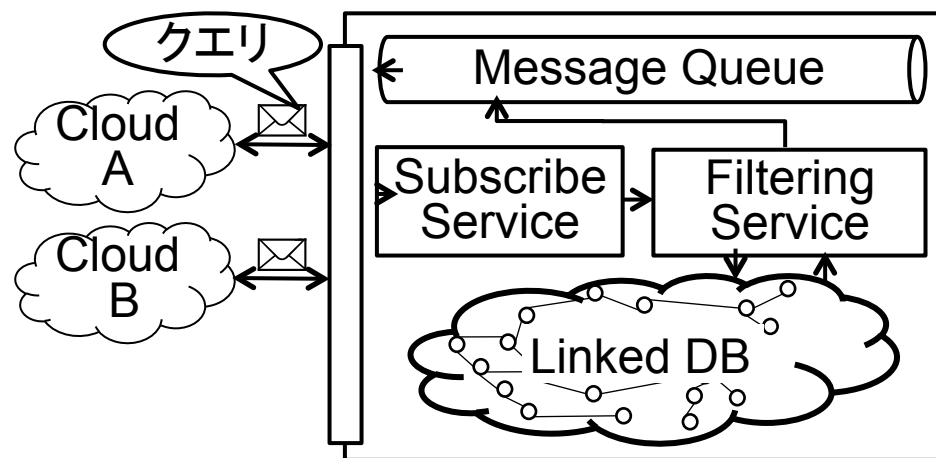
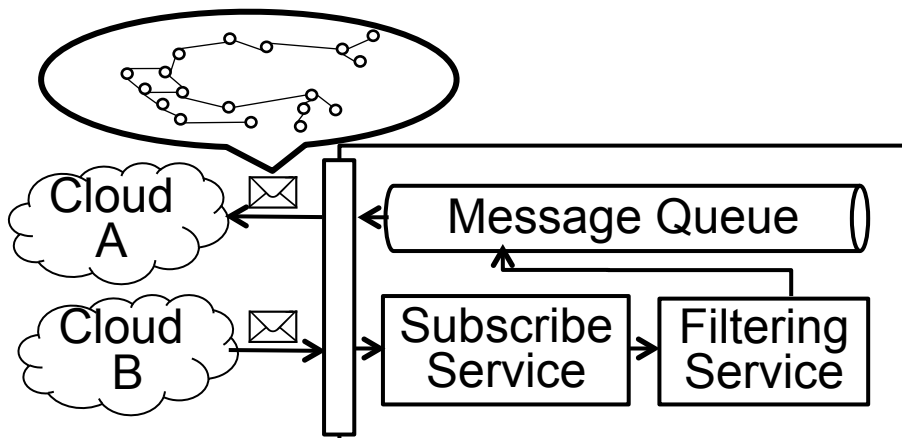


- 前回の復習と今回の流れ
- 研究概要
- 研究の方向性再検討
- アーキテクチャ比較
 - ❖ Publish/Subscribe Architecture
 - ❖ Linked Data Architecture
 - ❖ Publish/Subscribe with Linked Data
- 今後の課題
- 参考文献

+ 前回の復習と今回の流れ

■ 前回の復習

- ❖ 具体的なアーキテクチャ提案
- ❖ アーキテクチャの比較, 検討



■ 今回の流れ

- ❖ 研究方向性の見直し
- ❖ Linked Dataの可能性
- ❖ 3つのアーキテクチャの比較

+ 研究概要

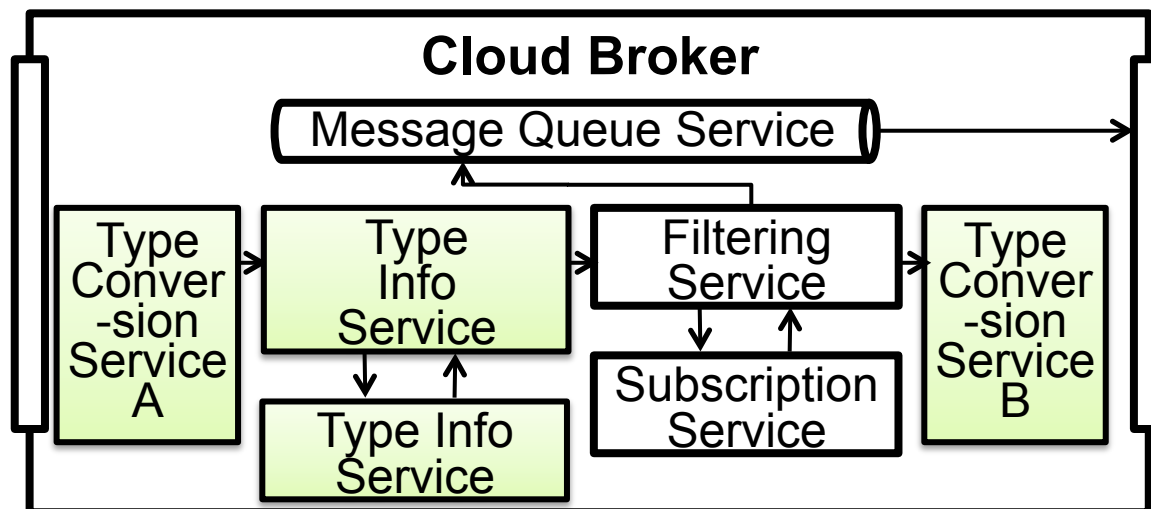
■ 背景

- ❖ クラウドの普及
- ❖ サービス連携の必要性

■ 問題: クラウドがサービスデータ連携方法が未定義

■ 研究課題

- ❖ 型の整合 (構造の整合, 意味の整合)



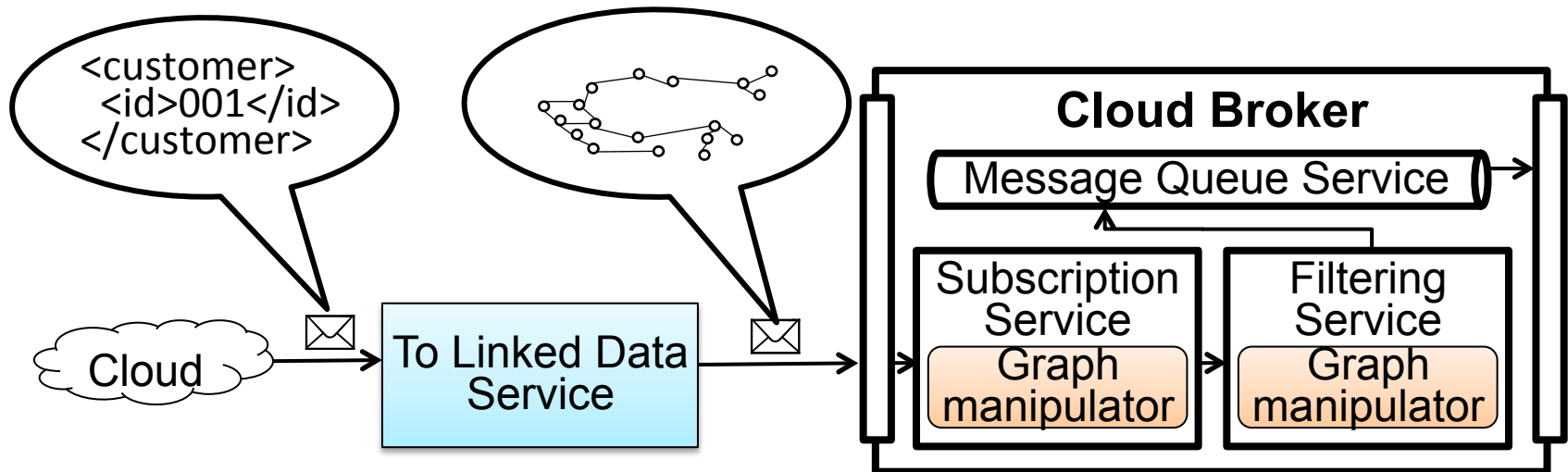
+ 研究方向性の再検討

■ 研究の概要

- ❖ Linked Dataを用いることでデータ構造を意識しない連携を実現
- ❖ URIにより全てのデータを機械処理可能な連携を実現

■ 研究の問題点: Linked Dataへの変換が行われている前提

- ❖ Linked Dataへの変換アダプタは既存研究のアダプタとほぼ同値?



➡ 研究の方向性を再検討する必要性大

+ 研究概要



■ 背景

- ❖ クラウドの普及
- ❖ サービス連携の必要性

■ 問題: クラウドがサービスデータ連携方法が未定義

■ 研究課題

- ❖ 型の整合 (構造の整合, 意味の整合)
- ❖ 振舞いの整合
- ❖ スケーラビリティ
 - ✓ 多量のデータ処理
 - ✓ 様々なサイズのデータ処理

+ なぜスケーラビリティが必要か

■ クラウドコンピューティングの特徴[1]

- ❖ On-demand: 利用者がクラウド・プロバイダーと人を介したやり取りを一切行わず、に必要なに応じてクラウド・サービスを使用可
- ❖ Reliability: 必要なに応じてリソースを拡張または縮小できる機能

⇒ スケーラブルな(多量)データ交換の想定

- ❖ Resource Pool: リソースをプール化することにより、クラウド・プロバイダはマルチテナントモデルを使用して利用者にサービスを提供可

⇒ 多量のデータ or 大きいデータサイズ交換の想定



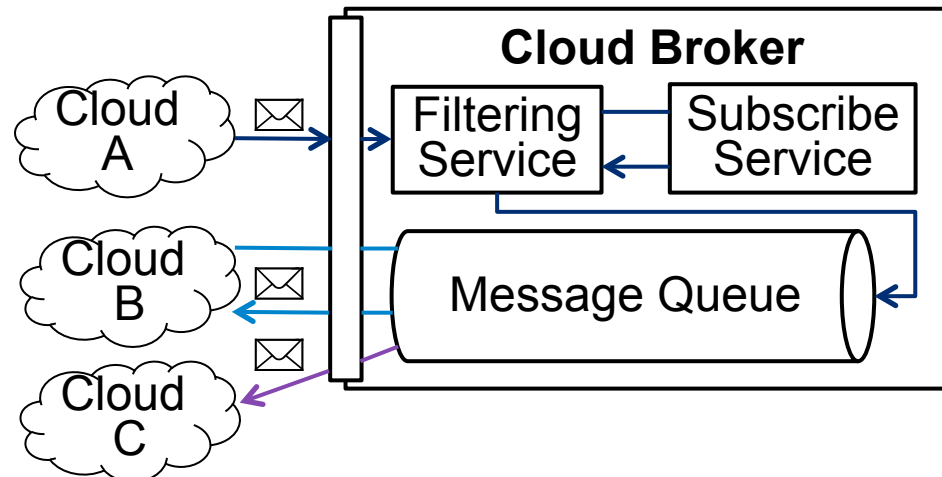
+ Publish/Subscribe Architecture



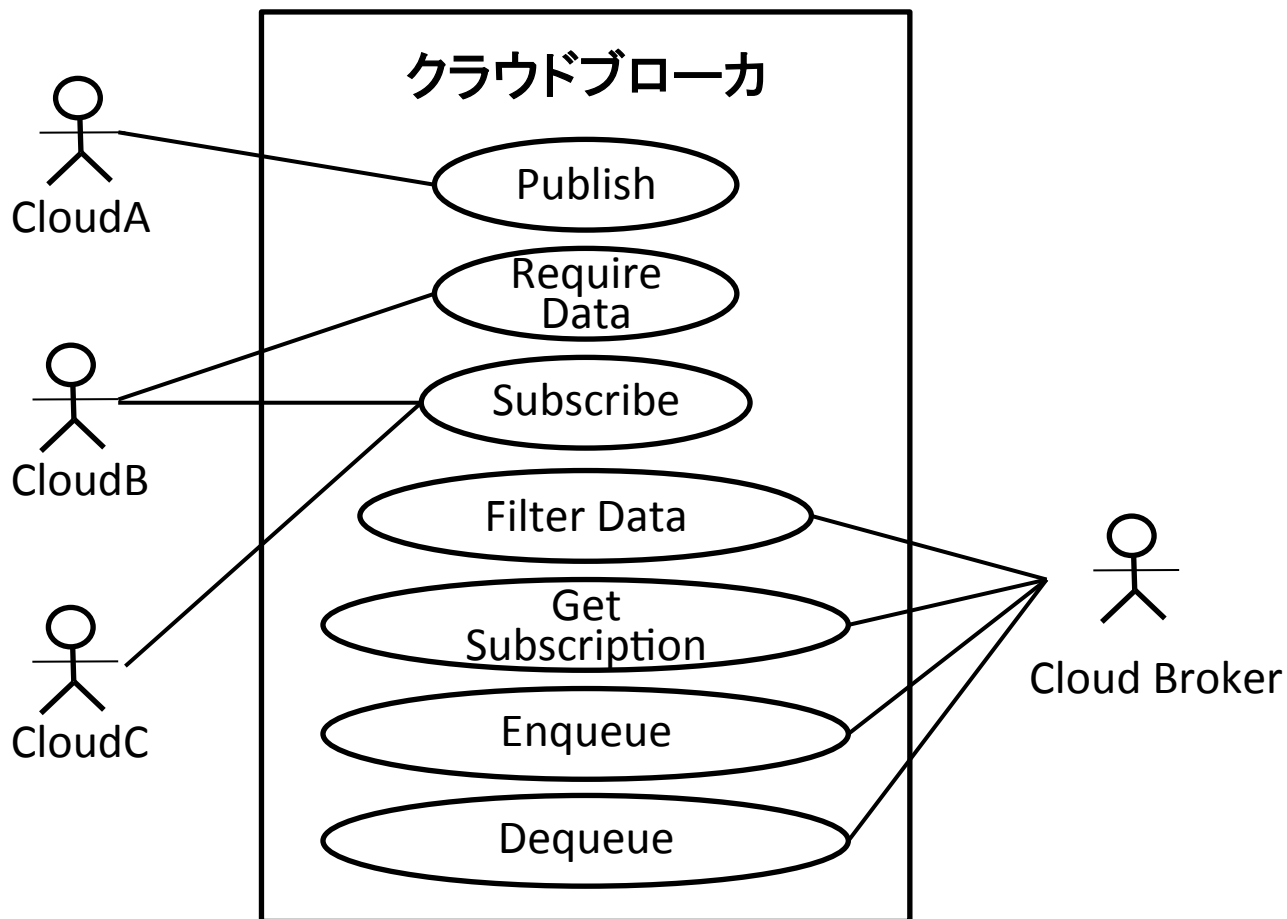
■ 非依存性の性質

- ❖ 空間的分離: エンドポイント情報の保持が不必要
- ❖ 時間的分離: インタラクシオンへの同時の参加が不必要
- ❖ 同期的分離: インタラクシオンがメッセージの送受信中にロックされない

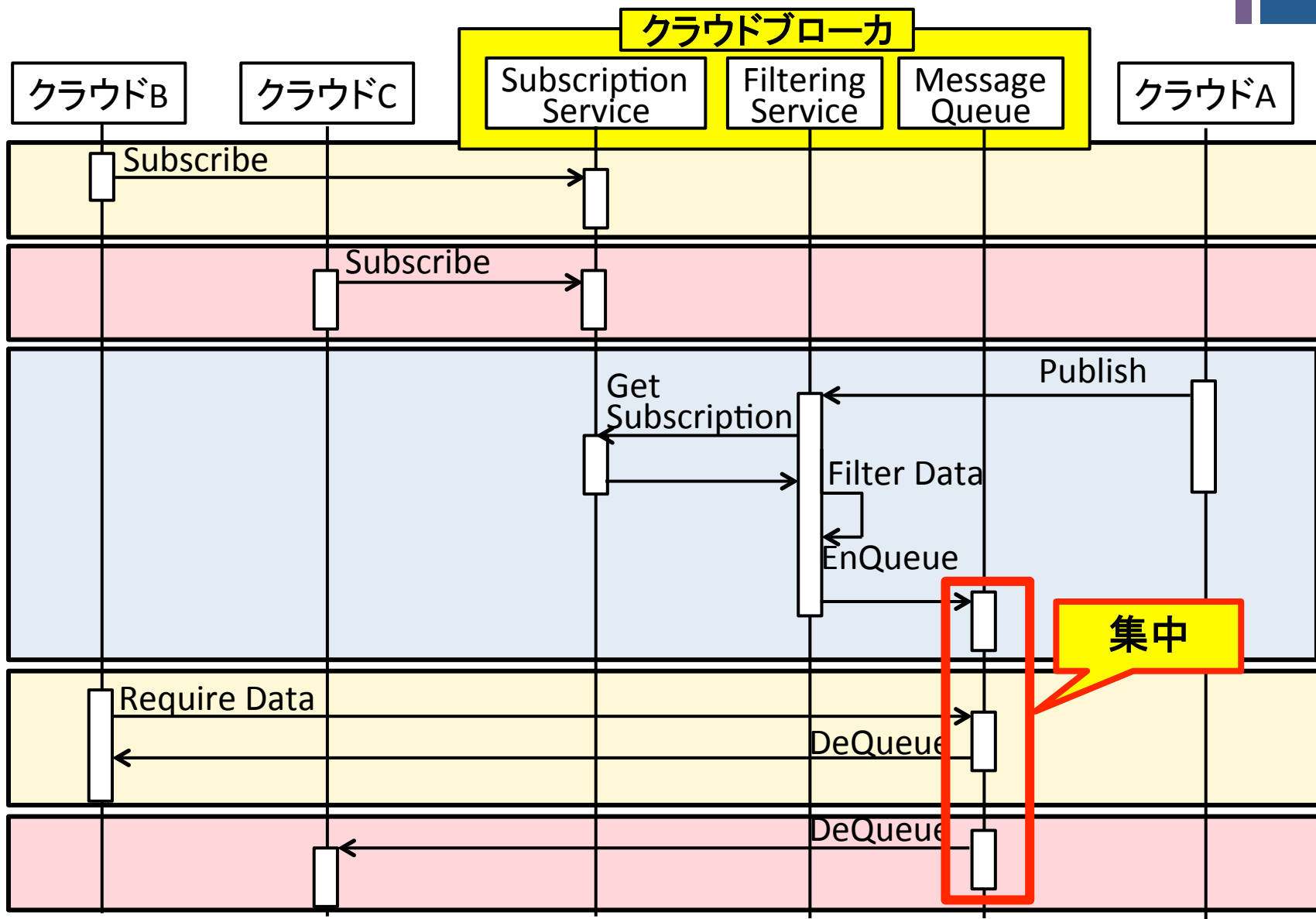
⇒ 不特定多数連携の振る舞いの整合に向いている



+ ユースケース図



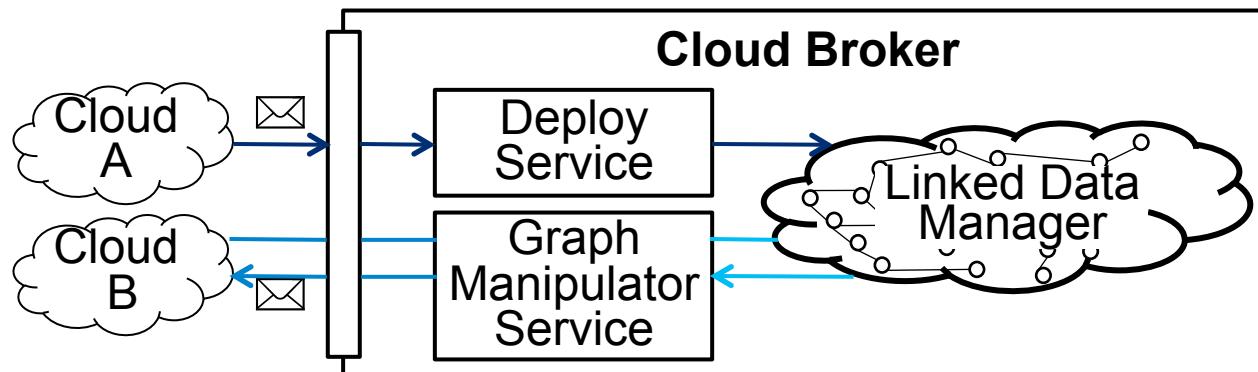
+シーケンス図



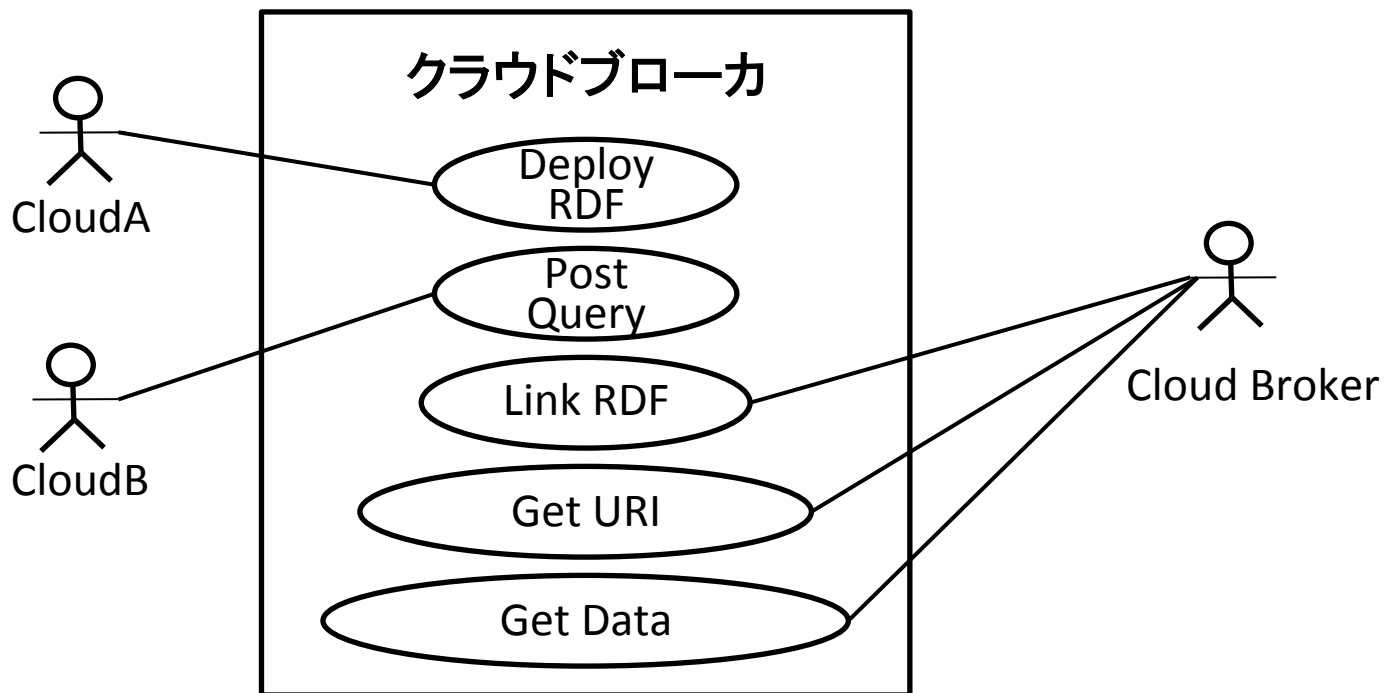
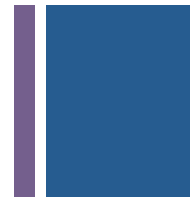
+ Linked Data Architecture

■ Linked Dataの特徴

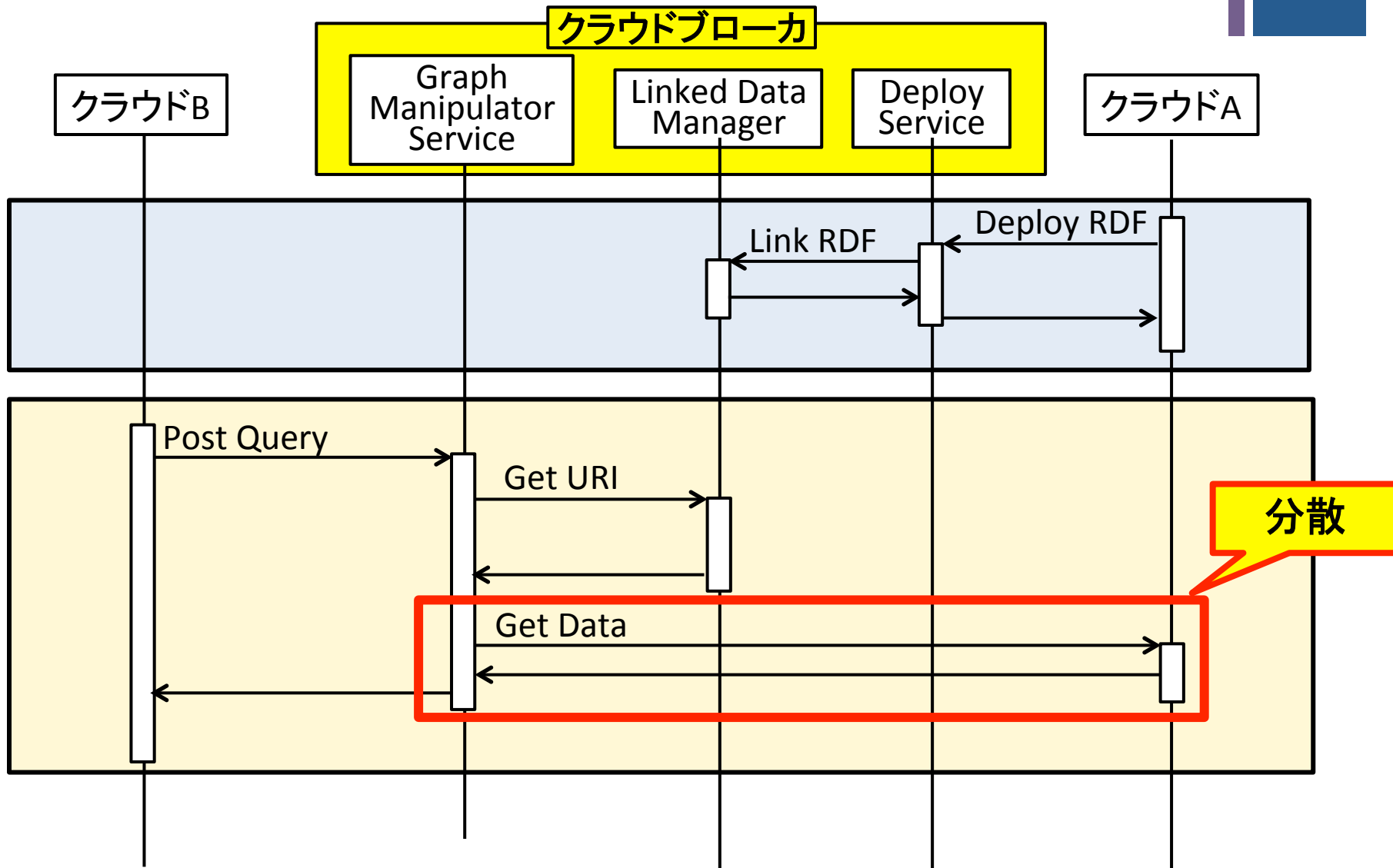
- ❖ リソースをURIで記述
 - ✓ 世界に一つの識別子
- ❖ 明示的かつ統一的な意味表現
 - ✓ 機械処理可能



+ ユースケース図



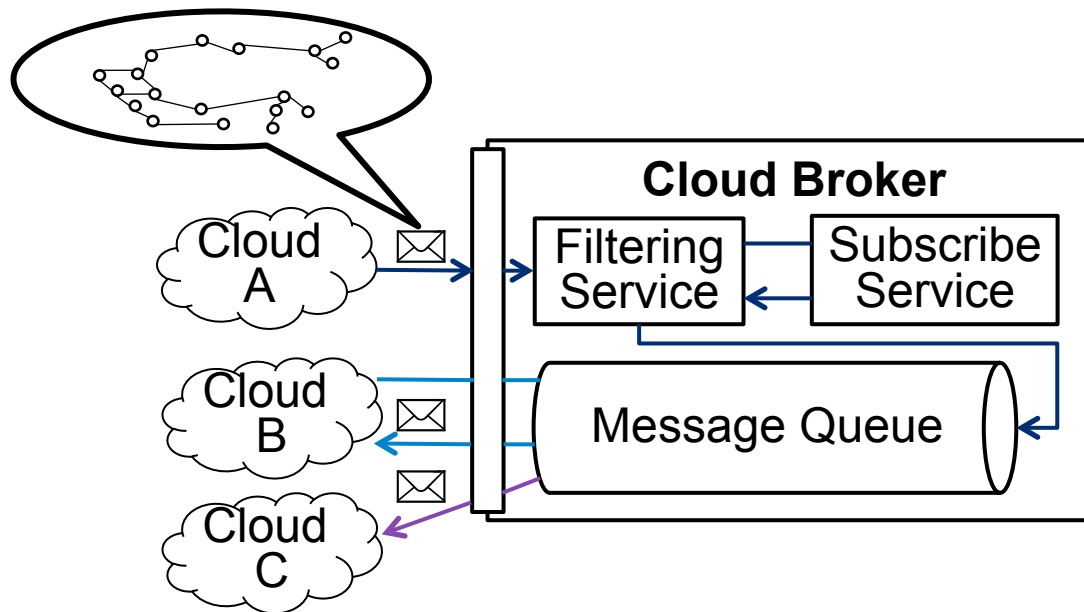
+シーケンス図



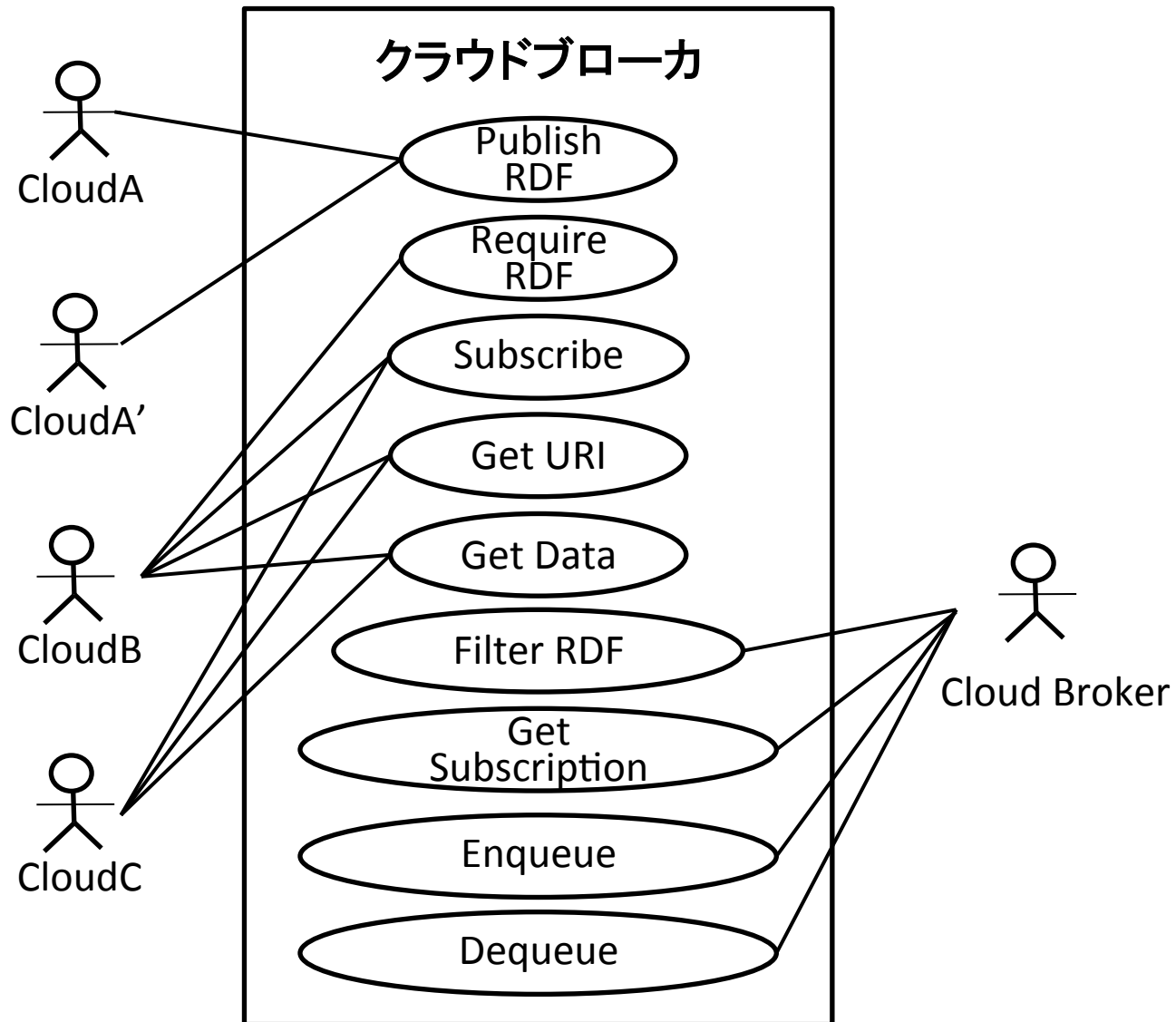
+ Publish/Subscribe with Linked Data

■ 前述の2つのアーキテクチャとの違い

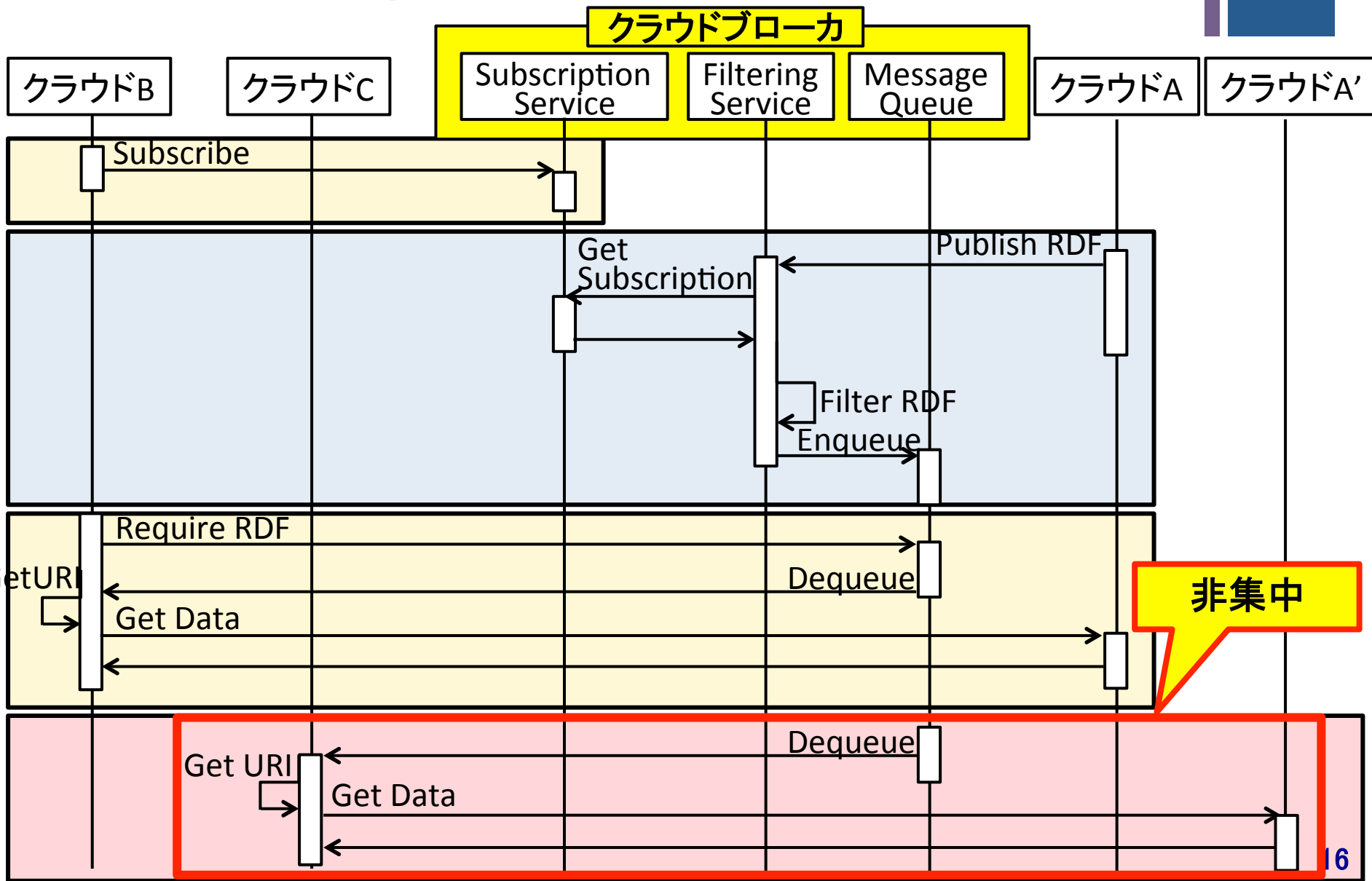
- ❖ Linked Dataでありながら振舞い整合の実現
- ❖ クラウドサービスがデータを取得する振舞いの変更



+ ユースケース図



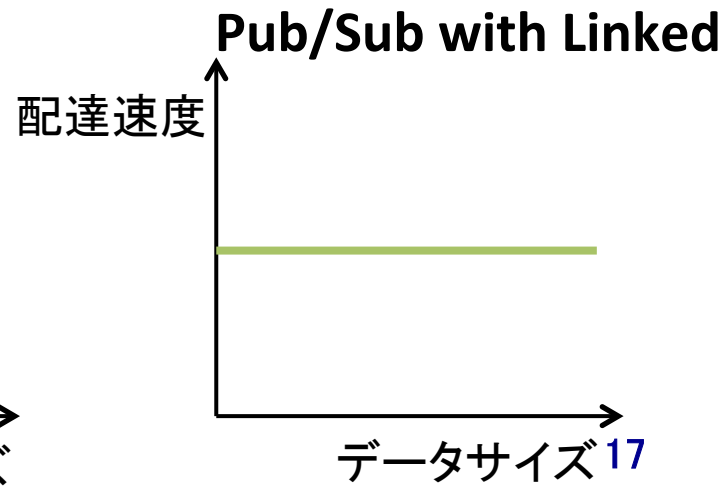
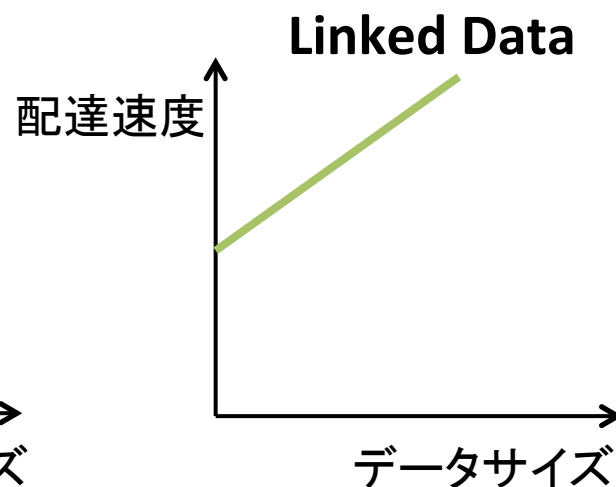
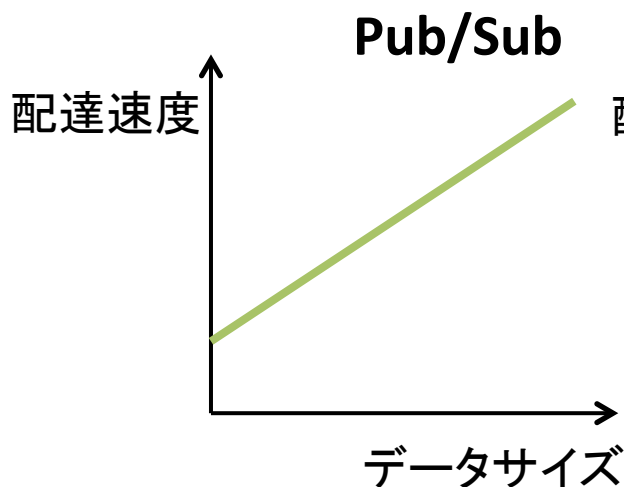
+シーケンス図



+アーキテクチャの比較と考察

■ アーキテクチャの比較

	振舞いの整合		スケーラビリティ	
	Push	Pull	データの位置と制御	サービス呼出し数
Pub/Sub	Yes	Yes	集中	5
Liked Data	No	Yes	分散	5
Pub/Sub with Linked Data	Yes	Yes	非集中	6



+ 今後の課題

- [実装] URI / Literalの連携時間の違いの計測
 - ❖ Apache Jenaを用いたLinked Dataの実現(2週間)
 - ❖ REST or SOAPでのLinked Dataの交換(4週間)
 - ❖ 受信したLinked Dataの処理(6週間)
 - ❖ データ交換時間の計測
- [調査] スケーラビリティに着目したデータ連携
 - ❖ 分散処理: Cloud Service Bus
- [提案] データモデル

+ 参考文献



- [1] Cloud Computing Use Cases Whitepaper, the Cloud Computing Use Case Discussion Group, July 2010 .
- [2] P. T. Eugster, et al., The Many Faces of Publish/Subscribe, ACM Computing Survey, Jun. 2003.
- [3] D. Allemang and J. Hendler, セマンティックWeb, May. 2010