



Linked Dataを用いた クラウドサービス連携アーキテクチャ

南山大学大学院 数理情報研究科
数理情報専攻 ソフトウェア工学専修

M2011MM041 小島 弘誉

指導教員: 青山 幹雄

+シナリオ



- 研究の全体概要
- 前回のおさらいと今回の狙い
- 実装の目的
- 実装での検証項目
- 実装最終ゴール
- 今回の実装
- グラフ操作サービスとは
- データ連携時間の比較
- ユースケースパターン
- 今後の課題

+ 研究の全体概要

- 研究課題: 不特定多数のクラウドサービス連携
 - ❖ クラウドサービスの疎結合な連携
 - ❖ スケーラブルなデータ連携
- 提案方法: クラウドサービス連携アーキテクチャ
 - ❖ パブリッシュ/サブスクライブに機能拡張
 - ❖ Linked Dataの適用
- 期待効果
 - ❖ ブローカに負荷が集中しないデータ連携が可能
 - ❖ 通知型, リクエストレスポンス型の連携が可能
- 今後の課題
 - ❖ アーキテクチャの詳細化
 - ❖ プロトタイプ作成による妥当性の確認

+ 前回のおさらいと今回の狙い

■ 前回のおさらい

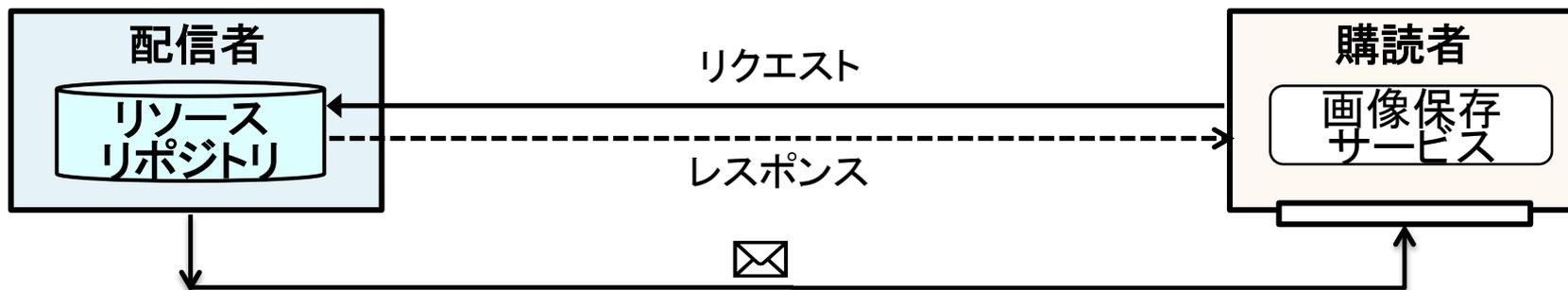
- ❖ 中間発表の質問と解答
- ❖ プロトタイプ作成計画
- ❖ SOAPを用いた画像取得サービスの作成

■ 今回の狙い: 配信者, 購読者のプロトタイプの作成

- ❖ 実装のユースケース
 - ✓ 視点: 配信者
- ❖ Linked Data用いた配信者, 購読者の作成
- ❖ SOAP with Attachment

+ 実装のユースケース

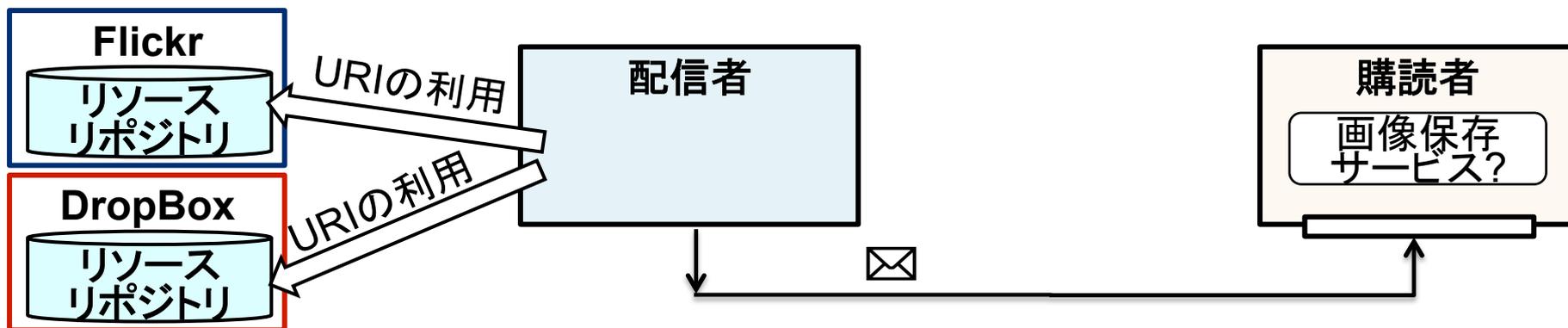
- 前提条件: 不特定多数の連携
- 配信者が保持しているデータを配信 ➡ リンクを活かしきれっていない?



- 配信者が保持しないデータを配信

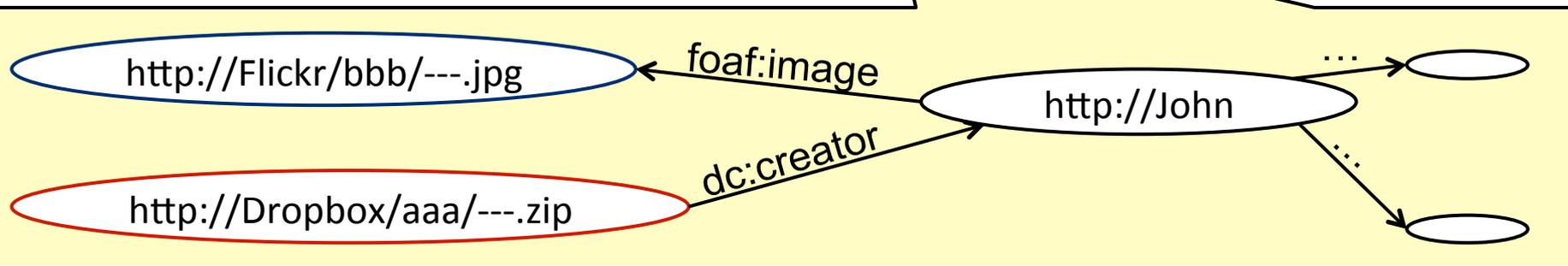
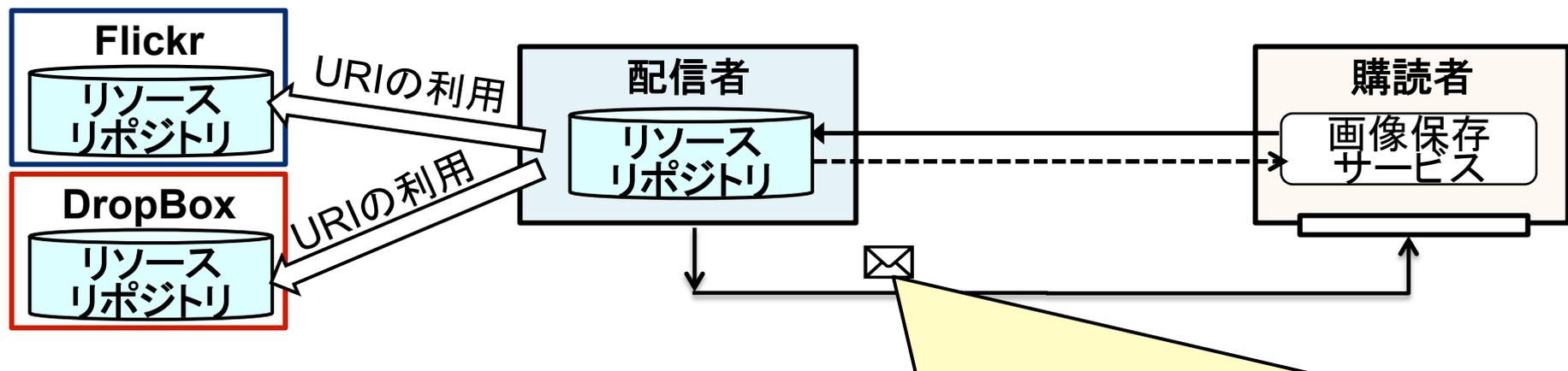
❖ (例) データを保持しないアプリケーション ➡ 研究課題との矛盾

✓ HTML5: ``



+ 実装のユースケース

■ 配信者が保持しないデータと保持するデータの配信



➡ 利用シナリオ具体化が必要

+ 実装の目的

- 連携アーキテクチャの妥当性の確認
 - ❖ スケールアウト実現性の確認
 - ✓ 負荷の計測方法は未決定
 - ❖ 連携時間の計測
- スケーラビリティの効果の確認
 - ❖ ブローカの負荷を計測
 - ❖ パブリッシュ/サブスクライブアーキテクチャと比較

+ 理論的な評価と仮説



■ 評価

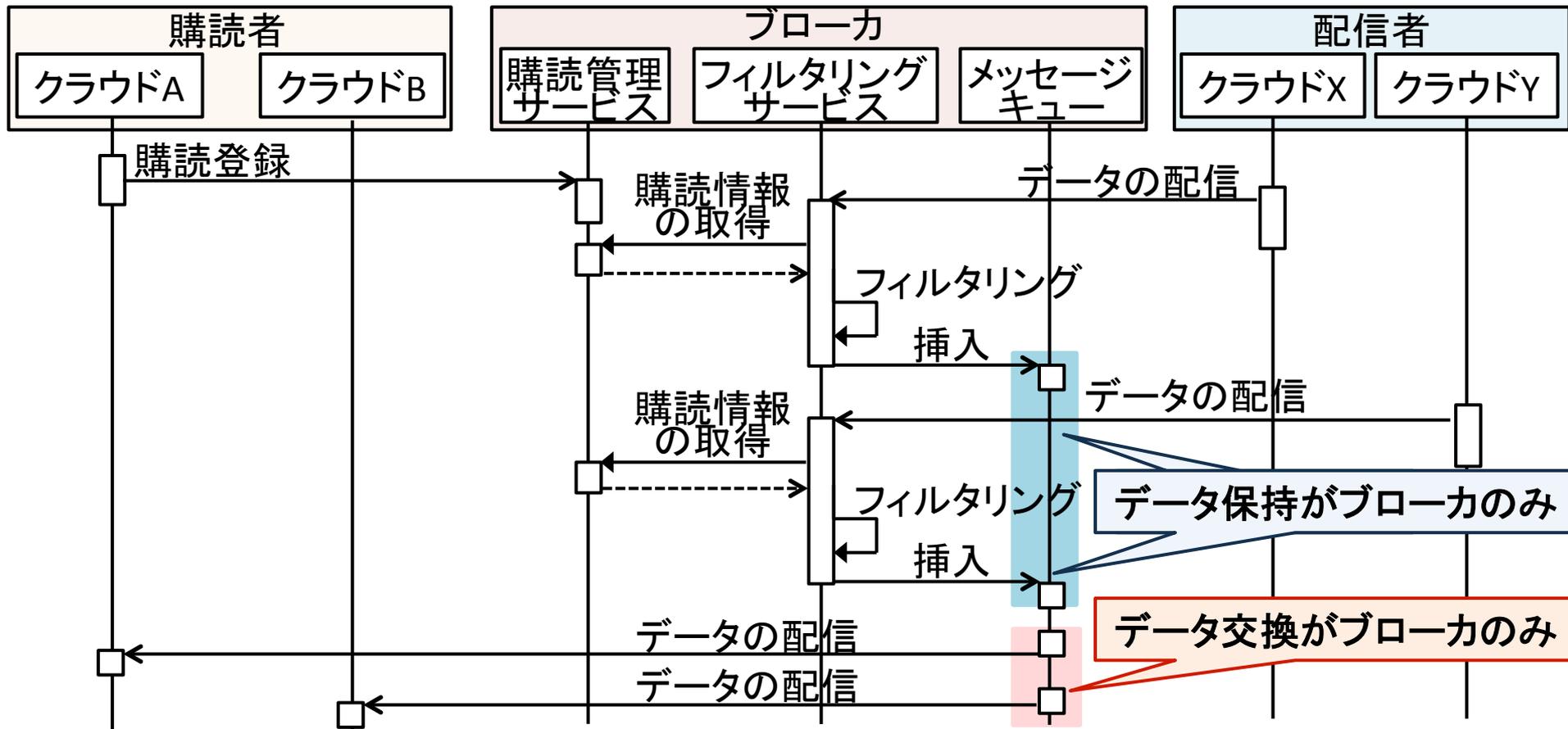
アーキテクチャ	データ交換	データの保持	データ連携	連携可能な振舞い
パブリッシュ/サブスクライブ	ブローカ	ブローカ	集中	通知型
Liked Data	ブローカ	各クラウドサービス	分散	リクエストレスポンス
提案アーキテクチャ	各クラウドサービス	各クラウドサービス	(1)非集中	(2)通知型, リクエストレスポンス

■ 仮説

- ❖ データ連携の負荷が小さい
 - ✓ 非集中なデータ連携を実現
- ❖ データ連携時間が大きい
 - ✓ サービスの呼び出し回数が多い

+ 提案アーキテクチャの評価(2)

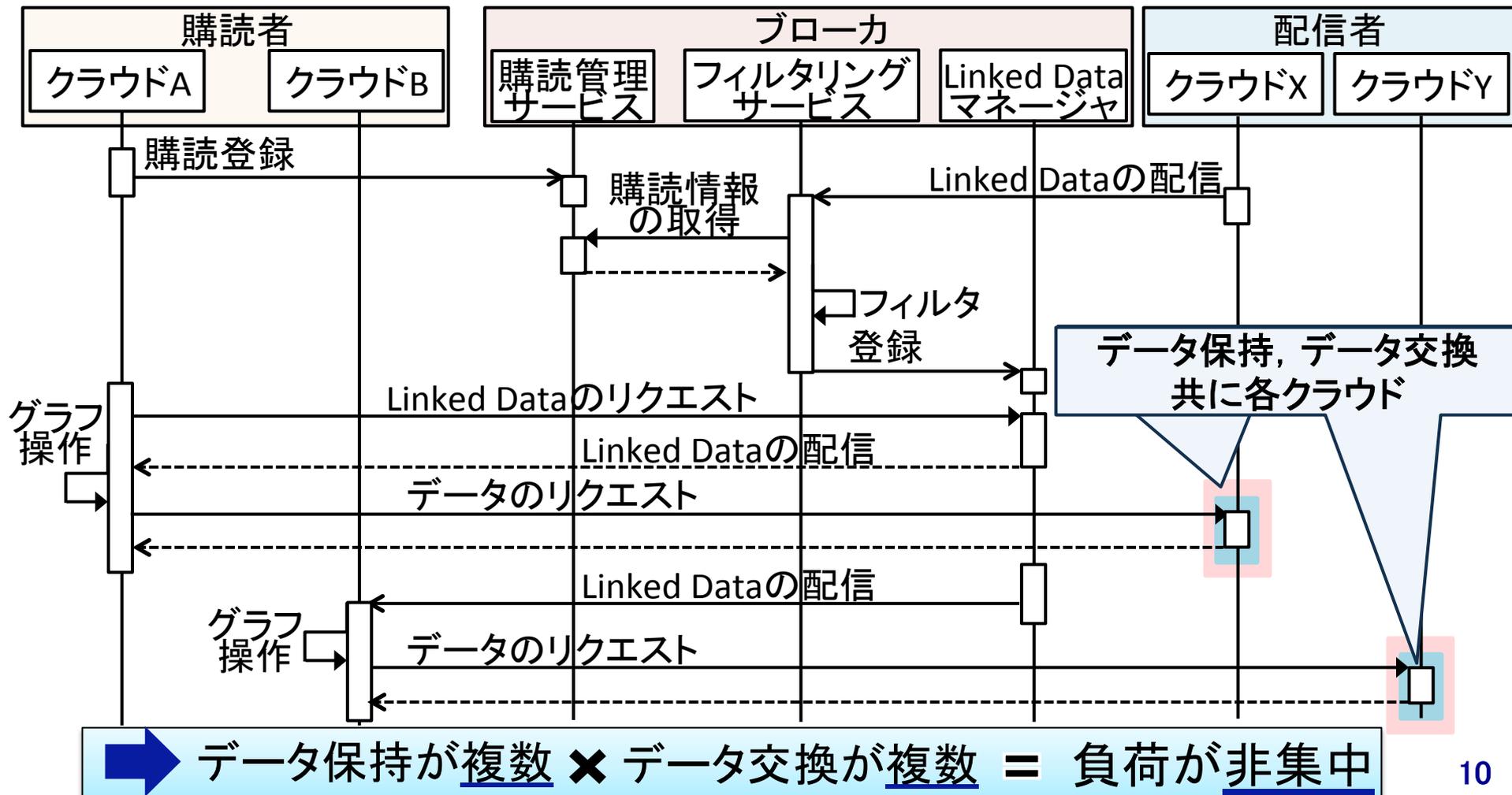
■ パブリッシュ/サブスクライブアーキテクチャ



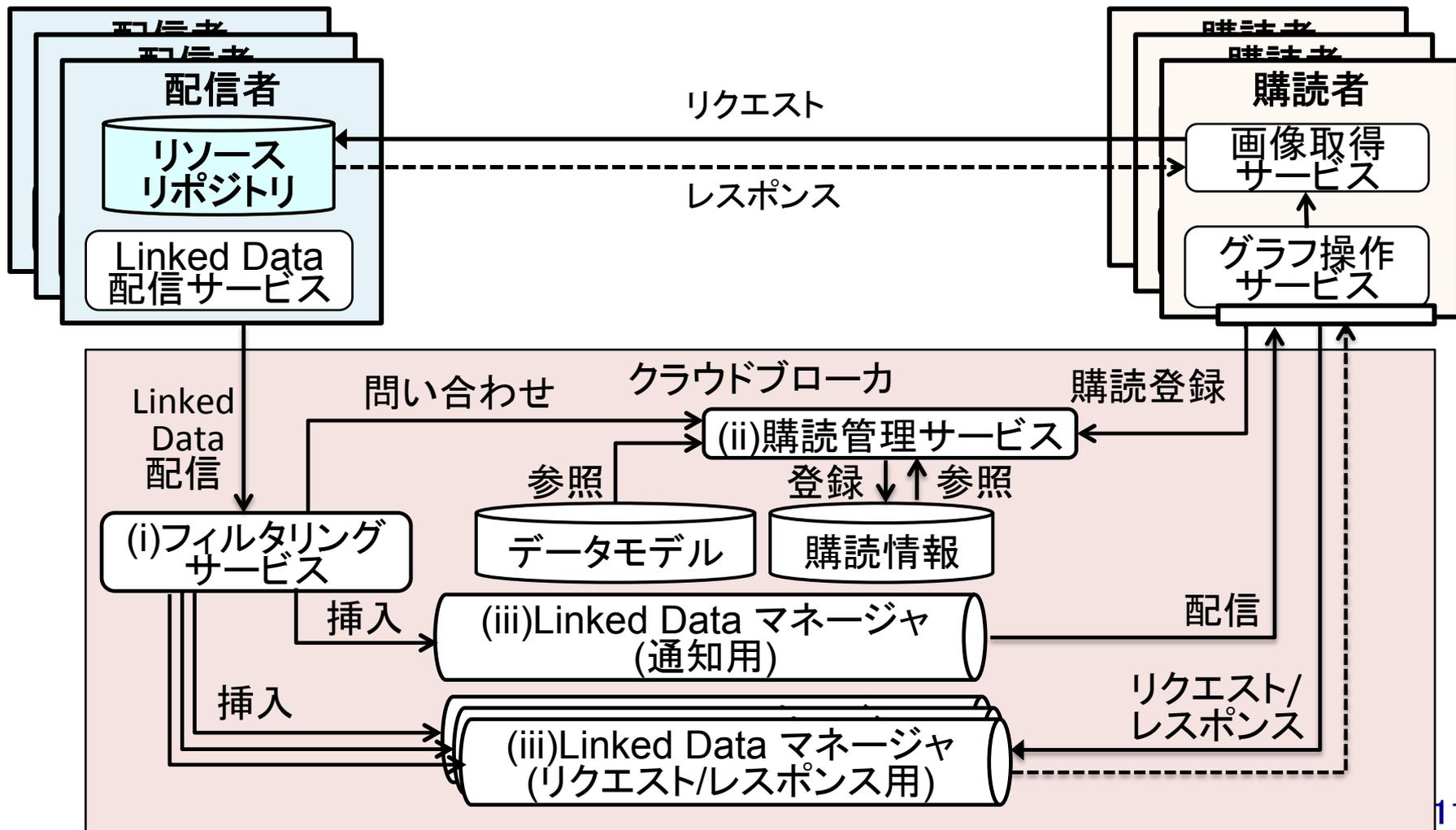
➡ データ保持が一つ × データ交換が一つ = 負荷が集中

+ 提案アーキテクチャの評価(3)

■ 提案アーキテクチャ

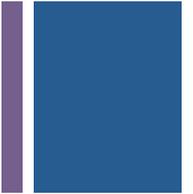


+ 実装最終ゴール(仮)



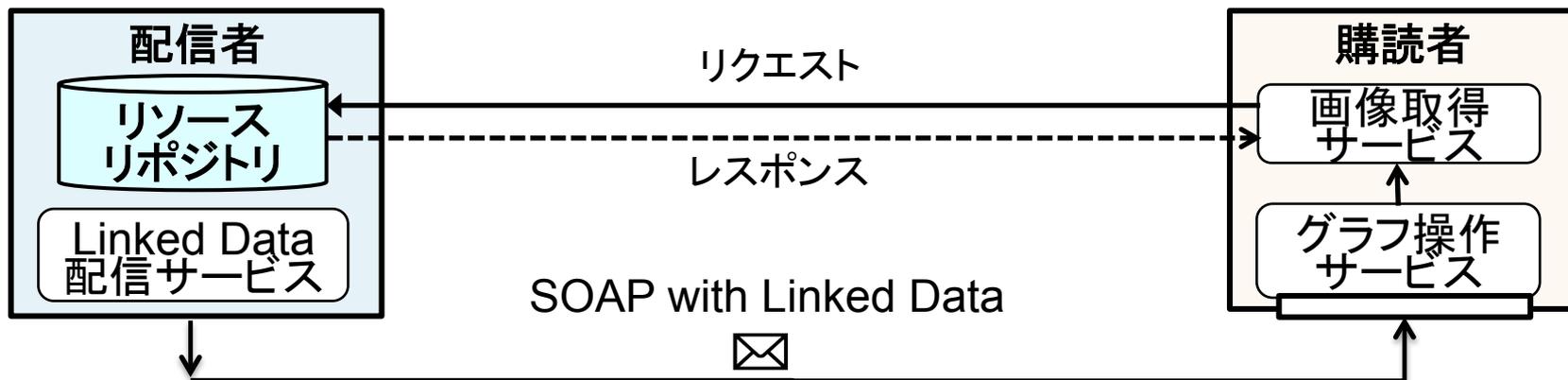
+ 今回の実装:実装環境

- OS: Mac OSX 10.8.2
- Web Server: Apache Tomcat v6.0.35
- Web Service Engine: Apache Axis2 v1.6.0
- Java Framework: Apache Jena v2.6.4

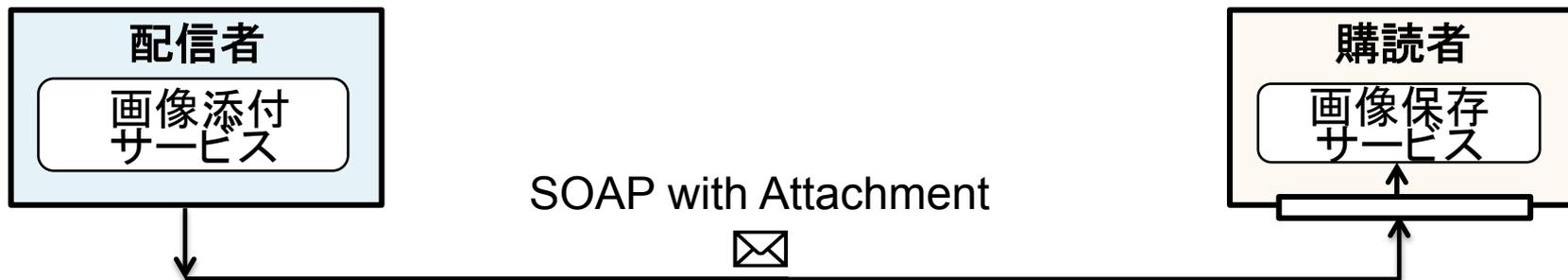


+ 今回の実装: 構造

■ 提案: Linked Dataを用いた配信者, 購読者の実装

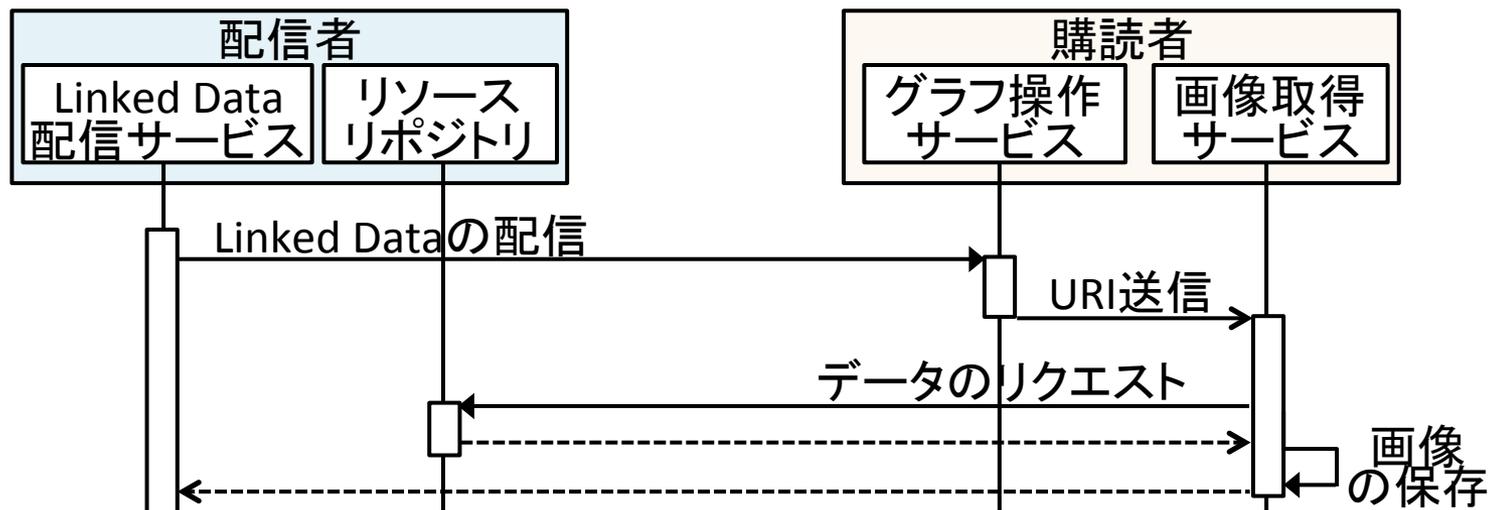


■ 比較対象: SOAP with Attachmentの実装

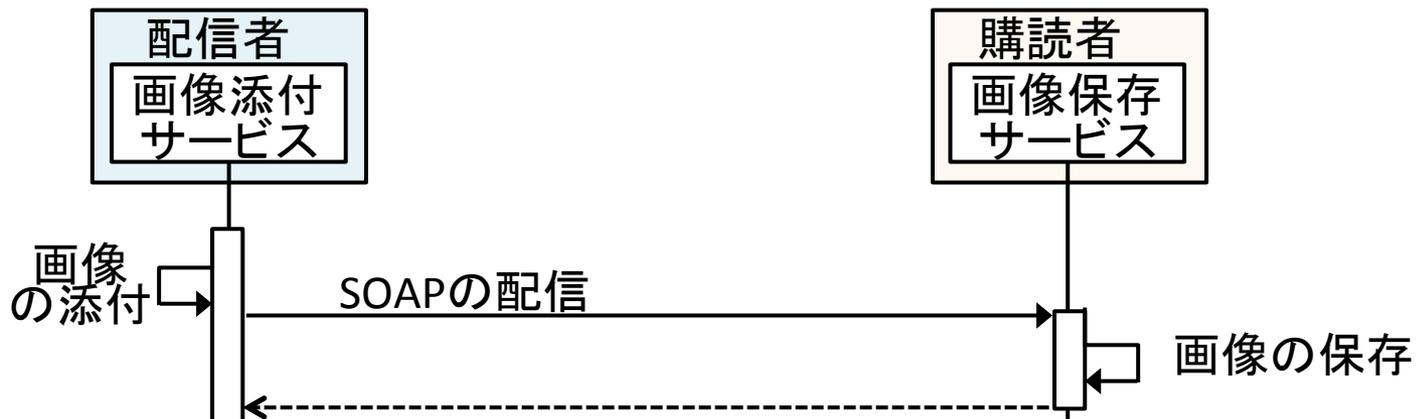


+ 今回の実装: 振舞い

■ 提案: Linked Dataを用いた配信者, 購読者の実装



■ 比較対象: SOAP with Attachmentの実装



+ 今回の実装: グラフ操作の詳細



■ 送信されるLinked Data

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:about="https://dl.dropbox.com/u/8824332/John.JPG">
    <foaf:name>John Smith</foaf:name>
  </rdf:Description>
</rdf:RDF>
```



■ モデル作成方法

```
String personURI = "http://somewhere/JohnSmith";
String fullName = "John Smith";
Model model2 = ModelFactory.createDefaultModel();
model2.setNsPrefix("foaf", FOAF.NS);
Resource John = model2.createResource("https://dl.dropbox.com/u/8824332/John.JPG");
John.addProperty(FOAF.name, fullName);
```

+ 今回の実装: グラフ操作の詳細

■ グラフ操作

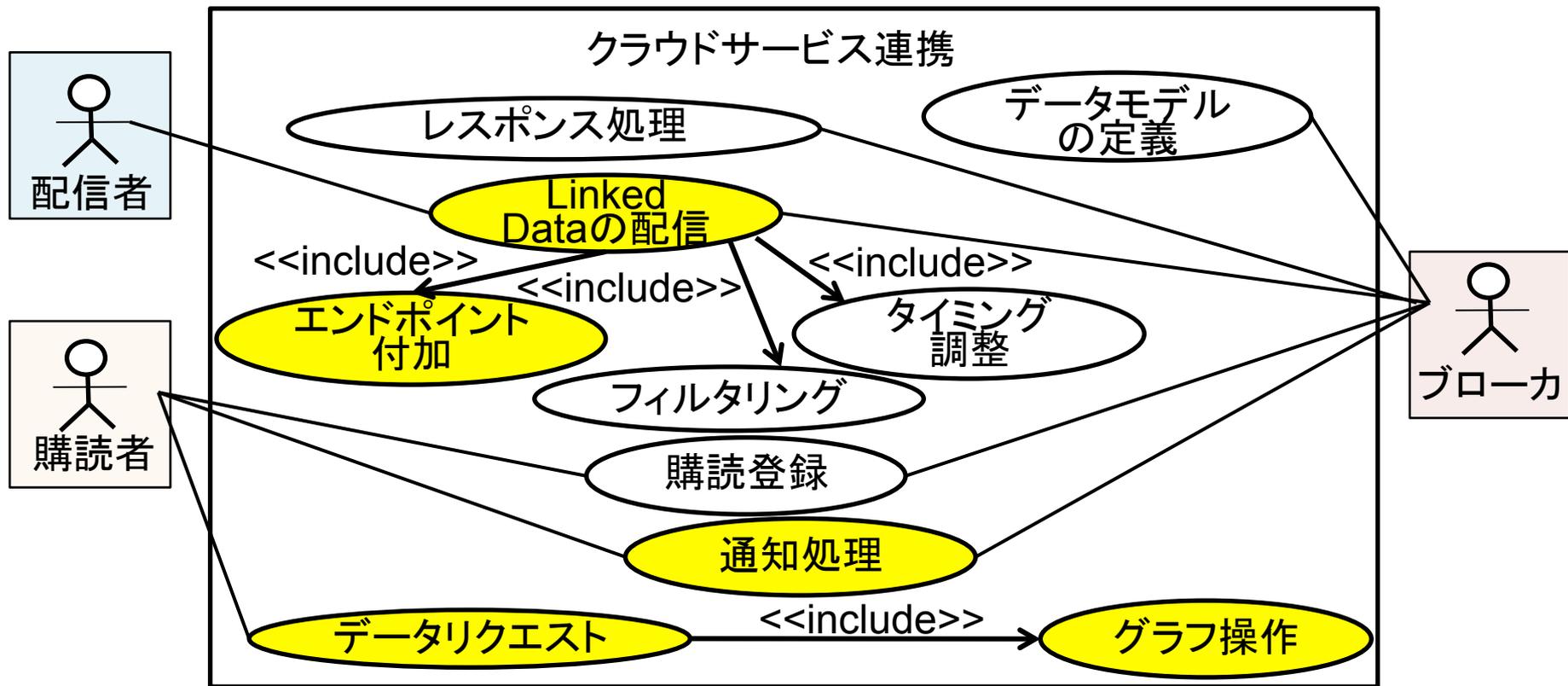
```
String queryString = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"  
    + "SELECT ?picture ?name WHERE{ ?picture foaf:name ?name }";  
Query query = QueryFactory.create(queryString);  
QueryExecution qe = QueryExecutionFactory.create(query, model2);  
ResultSet results = qe.execSelect();  
String URL = null;
```

picture	name
<http://Company-A/member/JohnSmith>	"John Smith"

作成

```
String name = null;  
while (results.hasNext()) {  
    QuerySolution row= results.next();  
    RDFNode picture= row.get("picture");  
    RDFNode Name = row.get("name");  
    Literal label= row.getLiteral("label");  
    URL = picture.toString();  
    name = Name.toString();  
}  
qe.close();
```

+ 実装が完了した機能



+ 今後の課題

- シナリオの決定 (~11/6)
- データの構造の考察 (~11/10)
- パブリッシュ/サブスクライブの実装 (~11/27)
- 連携時間の計測(~12/7)

LINKED DATAを用いた クラウドサービス連携アーキテクチャ

南山大学大学院 数理情報研究科
数理情報専攻 ソフトウェア工学専修
M2011MM041 小島 弘誉
指導教員: 青山 幹雄

END