

CEATEC Japan 2010

Webサービス, クラウドの先へ: サービスコンピューティングが拓く世界  
エンタープライズにおけるクラウドコンピューティングの適用可能性

## パネル討論

# クラウドコンピューティングの現状と今後

青山 幹雄

南山大学 情報理工学部 ソフトウェア工学科

[mikio.aoyama@nifty.com](mailto:mikio.aoyama@nifty.com)

<http://www.nise.org/>

We are **NISE**: Network Information and Software Engineering

2010年 10月 7日

# クラウドの3段階進化

☞ クラウドの進化モデル[Gartner 2008]

☞ 第3段階: 水平連携(Horizontal Federation)

☝ サービスブローカが鍵

クラウド統合(クラウドインテグレーション)  
クラウド間, クラウドとオンプレミスの連携  
による価値, 負荷耐力, 経済性などの向上  
**主役: サービスブローカ**

第3段階 水平連携  
(2012年)

メガプロバイダ+エコシステム  
(PaaS(Azure, Force.com, Google App Engine) 上にISVがSaaSとして集積): ベンダ固有アーキテクチャ  
**主役: エコシステム**

第2段階  
垂直サプライ  
チェーン  
(2010年)

エンタープライズ  
クラウド?

現在の状況?

第1段階  
モノリシック  
(2008年)

独立メガプロバイダ(Amazon, Google, 他)  
ベンダ固有アーキテクチャ  
**主役: メガプロバイダ**

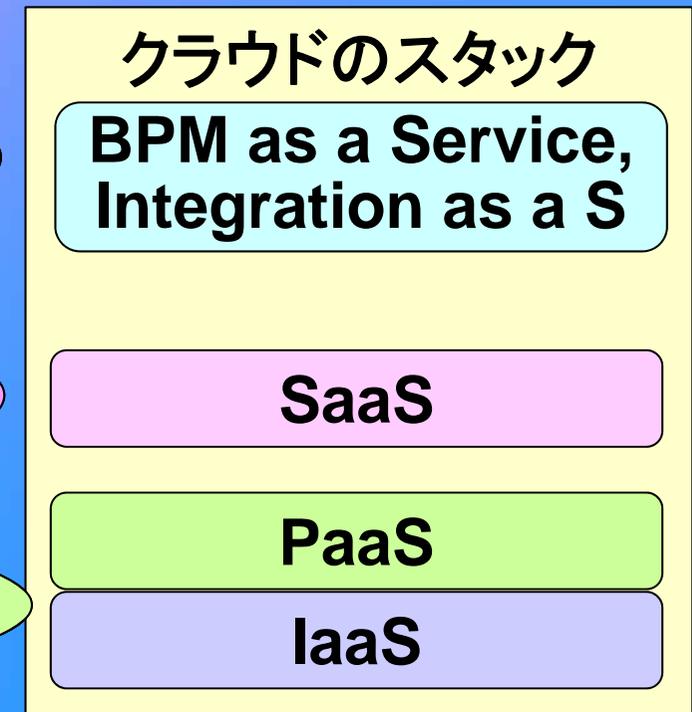
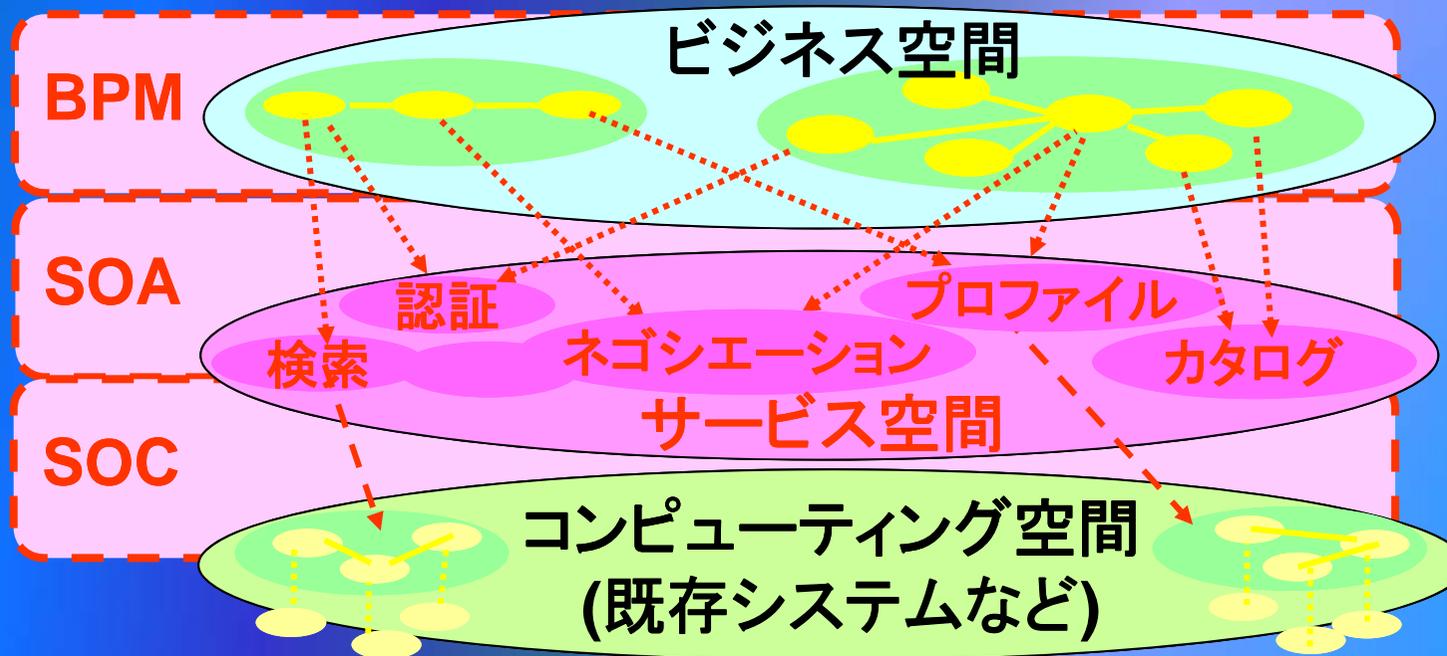
# クラウドとサービス指向アーキテクチャ(SOA)

☞ SOC/SOAとクラウドコンピューティングのスタック(SPI)

☞ クラウド統合スタックの出現

☞ BPM(Business Process Management) as a Service

☞ Integration as a Service



参考文献: 青山 幹雄, サービス指向アーキテクチャの誕生と進化, ソフトウェアエンジニアリング 最前線 2008, 近代科学社, Sep. 2008, pp. 9-16.

# クラウド統合の課題

## ☞ 統合のレベル

- ☞ SaaS, IaaS, PaaSのどのレベルで統合するか
- ☞ 統合のもたらす価値

## ☞ 統合のモデル

- ☞ プレゼンテーション(Mashup), データ(メッセージ), 制御(機能)
- ☞ ビジネスプロセス

## ☞ 統合の形態

- ☞ 連携のトポロジ:1対1, 1対N, N対N

## ☞ 統合の手段

- ☞ メッセージ交換: SOAP, POX
- ☞ インタフェースラッピング:WSDL

# SOAによるクラウド統合

## SOAによるアーキテクチャミスマッチの整合

### SOAによるアーキテクチャミスマッチの整合

- ブローカによる振舞い, 機能粒度の調整: コレオグラフィ(Choreography)
- Webサービス標準インターフェース: SOAP/REST



インターフェース整合問題  
SOA標準インターフェースによるオンプレミス/クラウド間インターフェース整合

マルチテナント問題  
シングルインスタンス/マルチテナントのユーザ定義と管理

カスタマイズ問題  
テナント毎の差異の定義, 管理

# SOAによるクラウド統合

## クラウドのアーキテクチャと処理モデル

クライアント/サーバ, Webサービス, クラウドのアーキテクチャ  
ギャップ

半構造的,  
非構造的  
データ  
(ファイル)

動的リンク

- クラウドコンピューティング
- ・アーキテクチャ: 分散データフロー
  - ・UI: Web,
  - ・非ACID/長寿命トランザクション
  - ・NoSQL DB

動的負荷

分散アーキテクチャ  
(分散処理, 集中制御)

アーキテクチャ  
ギャップ

非集中アーキテクチャ  
(分散処理, 分散制御)

- クライアント/サーバ  
(状態を持つ)
- ・アーキテクチャ: MVC
  - ・UI: Web/GUI
  - ・ACIDトランザクション,
  - ・SQL DB

動的発見

- Webサービス(SOA)  
(状態を持たない)
- ・アーキテクチャ: パブリッシュ/  
サブスクライブ
  - ・UI: Web
  - ・ACIDトランザクション,
  - ・SQL DB

構造的  
データ  
(RDB)

# SOAによるクラウド統合

## SOC/SOAとクラウド

### ☞ サービスコンピューティング(SOC/SOA)

- ☞ システム=サービス+メッセージ
- ☞ 状態を持たない(Stateless)サービスモデル
- ☞ 組合せ可能(Composable)
- ☞ LRT(Long-Running Transaction)

### ☞ クラウドコンピューティング

- ☞ システム=サービスプール+デリバリ(?)
- ☞ ネットワーク上の処理モデル: ACID/SQL + BASE/NoSQL (?)
  - ☞ NoSQL: キーバリュー型, カラム指向, ドキュメント指向
  - ☞ BASE(Basically Available, Soft state, Eventually consistent):  
[弱一貫性(Weak Consistency)]
- ☞ サーバクラウド/スケールアウトクラウド
- ☞ マルチテナント + 単一コードベース

Some old assumptions may now be misconceptions.  
- Strict locking & consistency  
- Relational data  
- Transaction  
[B. Neidercker-Lutz, SAP, LADIS2010]

# SOAによるクラウド統合

## SOAによるクラウド統合の課題

### ☞ SOC/SOAによるクラウド間/クラウド・オンプレミス間水平連携

- ☞ メッセージ形式: Webサービス標準(WSDL, SOAP/REST)
- ☞ コレオグラフィ(Choreography) [メッセージプロトコル]による連携
- ☞ ESB(Enterprise Service Bus) [Publish/Subscribe+メタデータ]

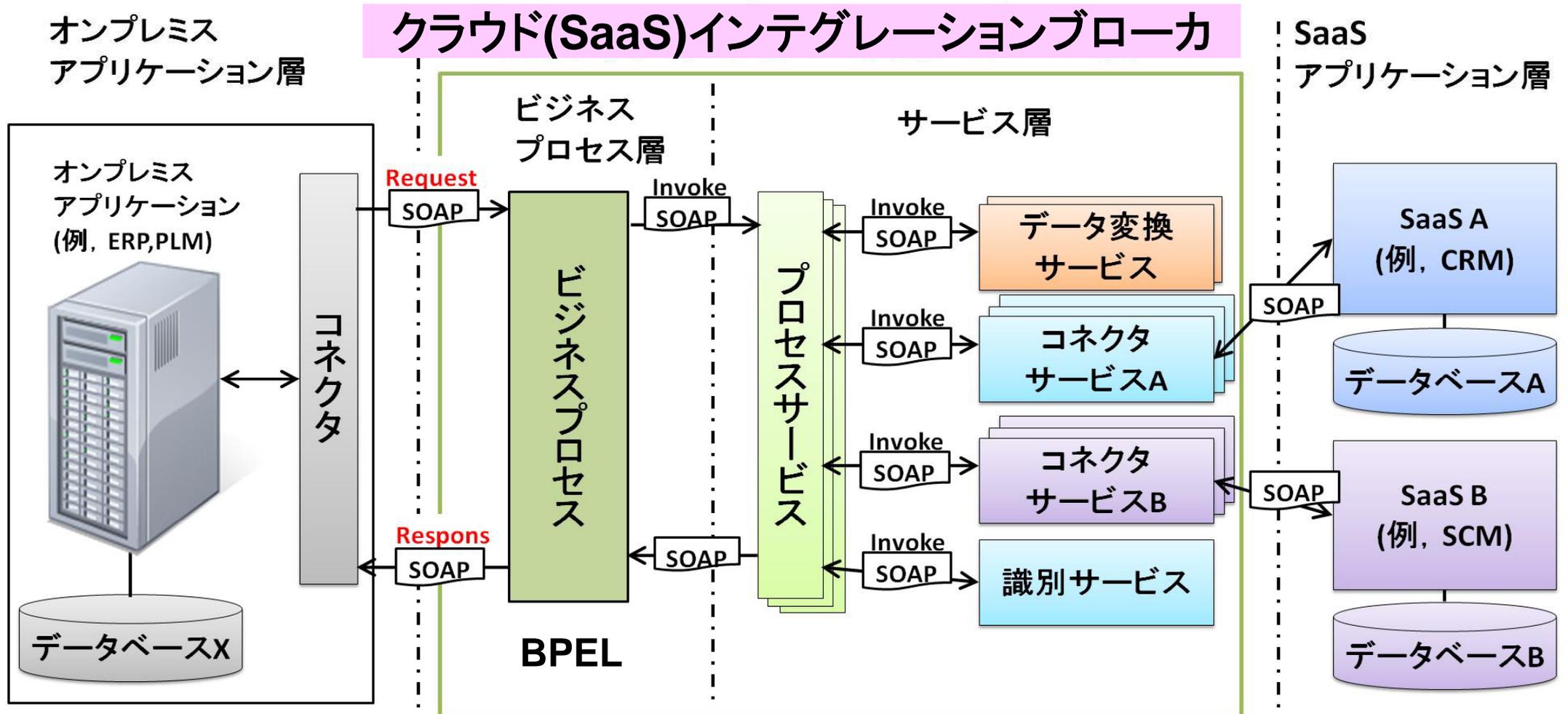
### ☞ 統合における問題: 相互運用性

- ☞ アーキテクチャミスマッチ ⇒ 振舞いの整合
  - ☞ 例: Salesforce.comのoutbound message: トランザクション処理単位にメッセージ送信
- ☞ メッセージ/データ形式ミスマッチ ⇒ メッセージ変換
- ☞ 非機能要求ミスマッチ: セキュリティ, SLA, ほか
- ☞ 制約/非標準性: プログラミング言語の制約

# SOAによるクラウド統合の事例

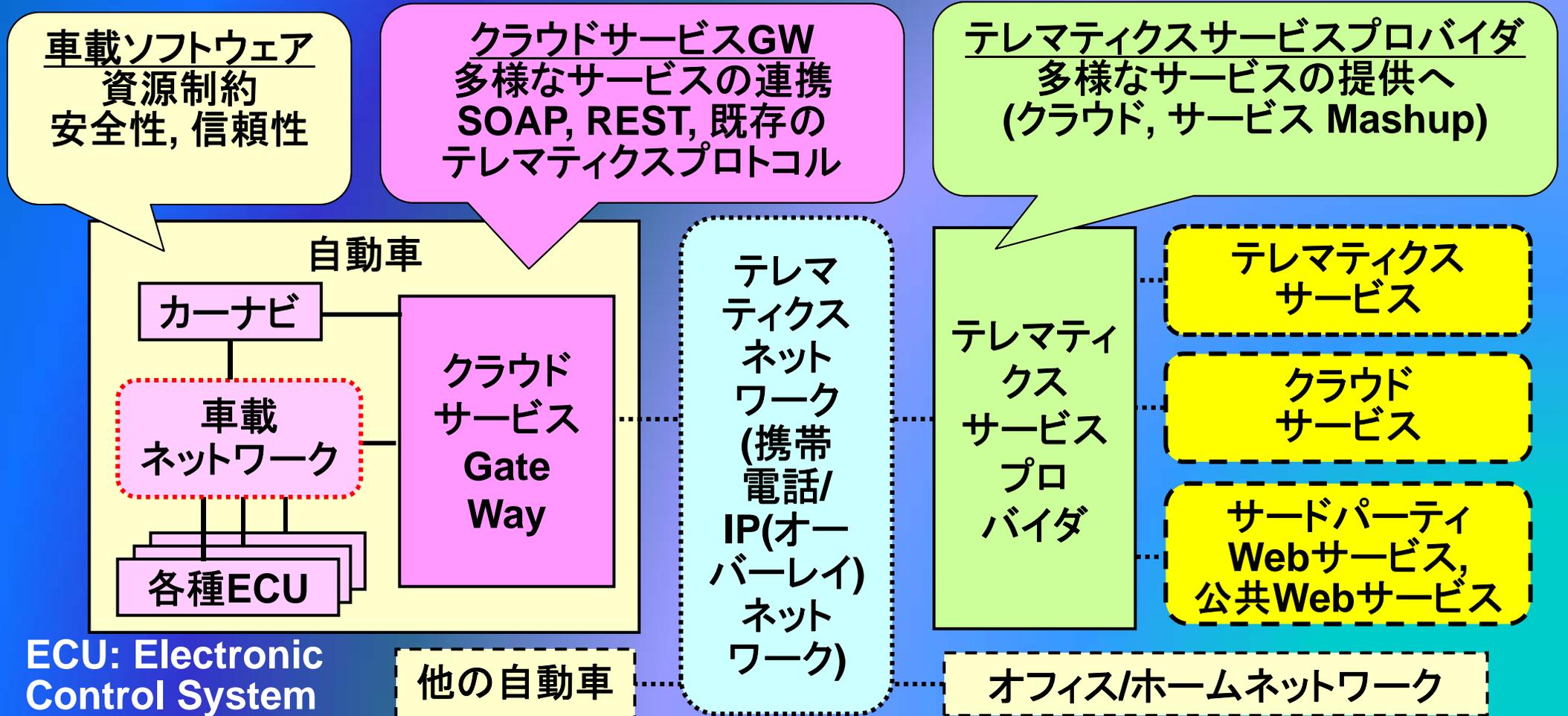
## BPMによる統合例(1/2)

- 👉 ビジネスプロセス層: インタラクションの整合
- 👉 サービス層: 機能の整合



# SOAによるクラウド統合の事例:自動車サービス連携 ACSS(Automotive Cloud Service System)

☞ クラウドと連携して多様なサービスを提供

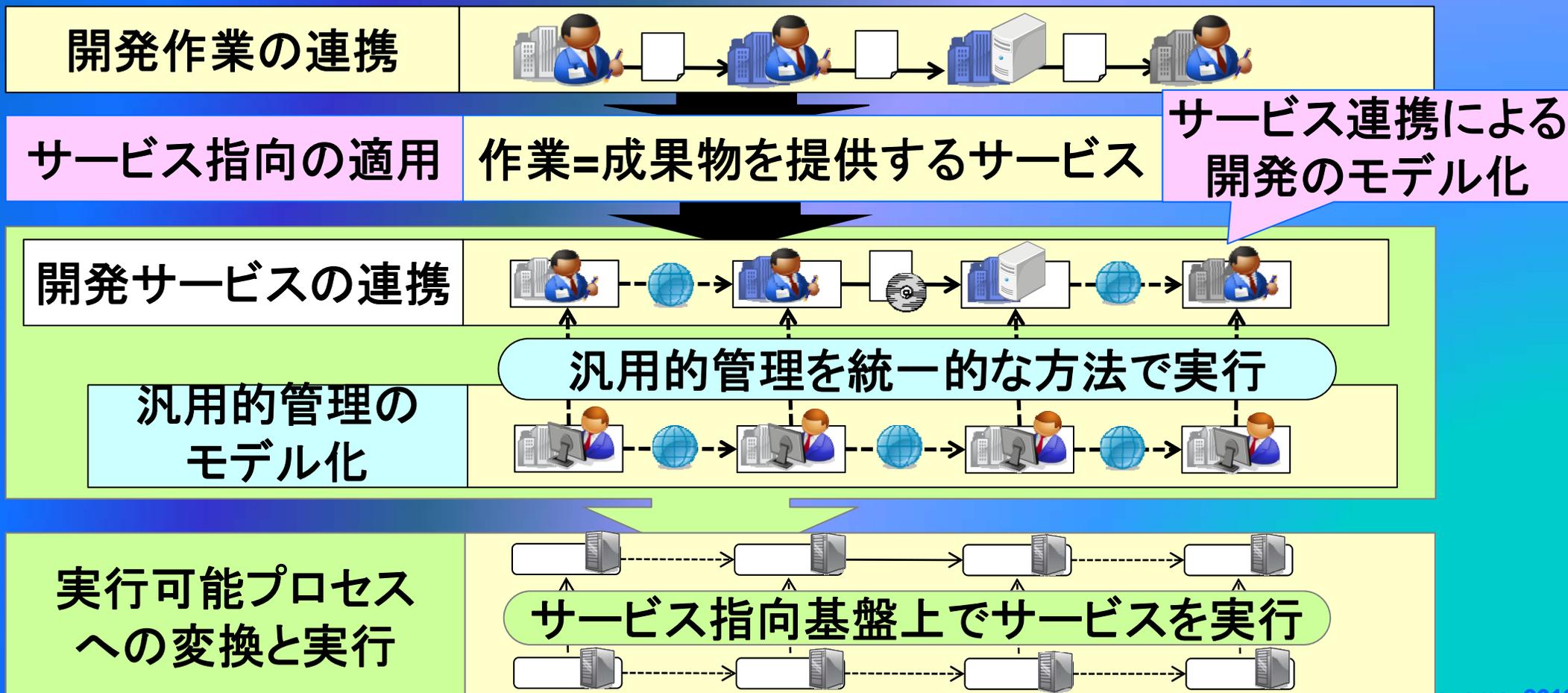


参考文献:青山 幹雄 ほか, 車載ソフトウェアのサービスプラットフォームのモデルとアーキテクチャ,  
自動車技術会2008年秋季大会 学術講演会前刷集, No. 97-08, Oct. 2008, pp. 21-26.

# SOAによるクラウド統合 サービスとしてのソフトウェア開発(1/3)

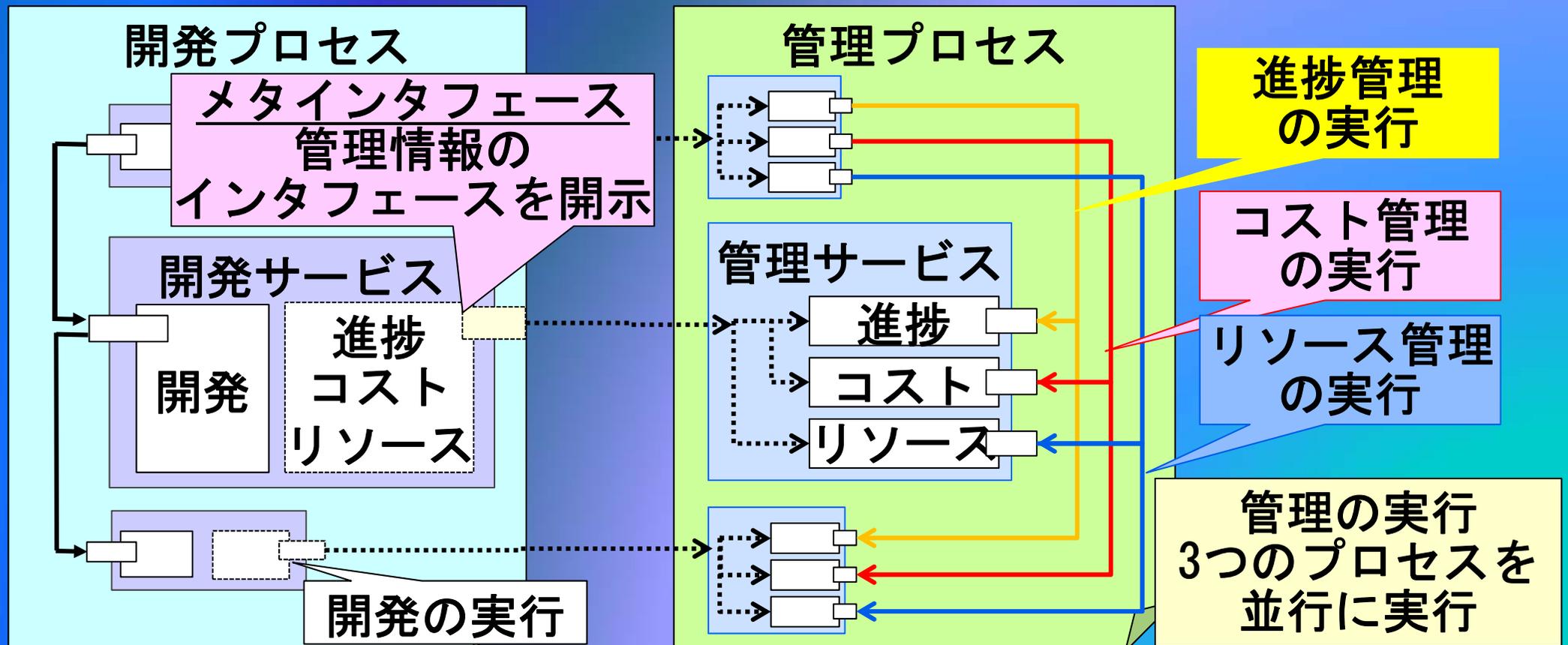
- ☞ ソフトウェア開発へのサービス指向の適用 ⇒
- ☞ 汎用的管理のモデル化 ⇒
- ☞ 定義したプロセスを実行可能な記述へ変換 ⇒

**SDaS:  
Software  
Development  
as a Service**



# SOAによるクラウド統合 サービスとしてのソフトウェア開発(2/3)

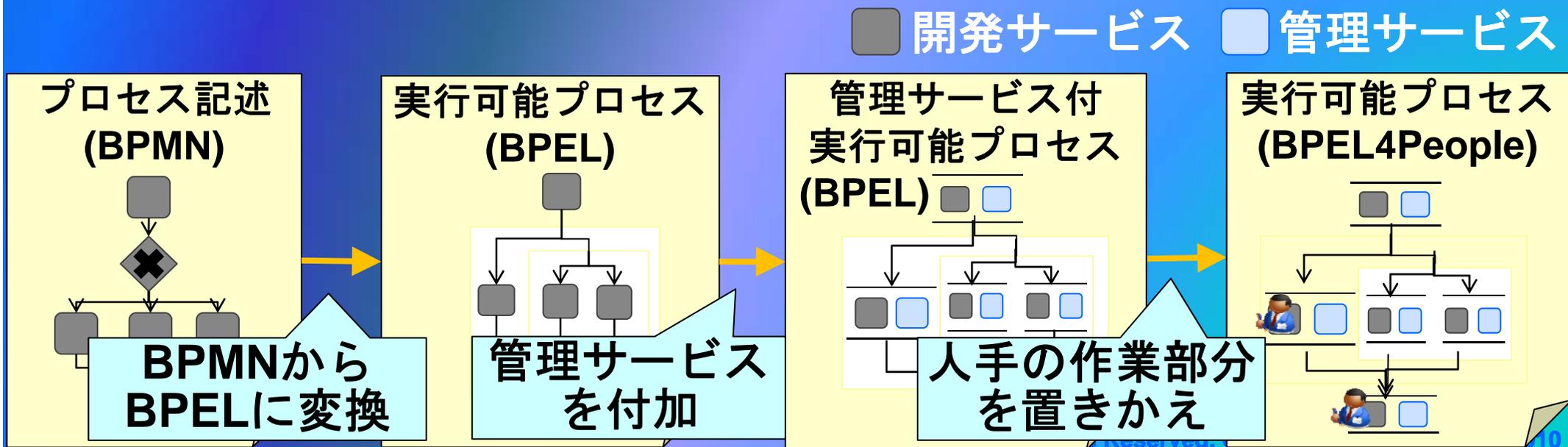
- ☞ ソフトウェア開発=サービス
- ☞ 開発プロセスに対する標準管理プロセスを自動生成
- ☞ BPEL for People, WS Human Taskによる実行



# SOAによるクラウド統合 サービスとしてのソフトウェア開発(3/3)

## 変換方法

- 管理モデルに基づき管理サービスを付加
  - BPELのflow要素で管理サービスと開発サービスを並行実行
- 実行可能SoSDモデル(BPEL4People)に変換
  - 人手の作業をWS-Human Taskを用いてサービス化
  - 記述したサービスで人手の作業を置きかえ



# 今後の課題

- 👉 オンプレミスとクラウド/複数クラウドの統合モデルの構築
- 👉 クラウド統合アーキテクチャの構築
- 👉 統合モデルとユースケースに基づくビジネス/サービス品質モデルの構築
- 👉 ビジネス/サービス品質のモデルとメトリクス
- 👉 ビジネス/サービス品質の管理モデル, 管理技術, 管理システムの設計方法
- 👉 ビジネス/サービス品質の測定と評価の方法
- 👉 ビジネス/サービス品質の改善方法
- 👉 実システムへの技術の応用や開発, 運用の経験の収集, データの収集と分析