

# クラウドコンピューティングとどう向き合うか

～クラウドはビジネス系情報システム構築技法の革命となりうるか～

2010年度第1回SWIM研究会

2010年6月4日

宮西洋太郎

宮城大学客員教授

仙台ソフトウェアセンター嘱託

(株)アイエスイーエム

# クラウド出現の背景(情報技術へのニーズ)

- ユーザの観点から1つのコンピュータを多数のユーザが利用するというマルチユーザ(マルチテナント)の考え、
- 1つのコンピュータでは処理能力や処理機能(サービス)が不十分の場合に、複数のコンピュータによってサービスを提供するインターネット経由の広域的な分散システムの考え、さらには、
- 自己(自社)でコンピュータを所有せず、外部でサービスとして提供されているコンピューティング機能を共同利用するという考え、これらが融合してきたものと考えられる。
- ただし、実現技術として、主にGoogle社が大量の情報を検索するために築いてきた従来に比べてきわめて大規模かつ柔軟な規模の拡張可能性(*scalability*)を実現するための様々なソフトウェア技術が用いられている。その点において、やはり新しいコンピューティングの形態の出現ともいえるのではないだろうか。

# 情報技術の進歩(ハードウェア)

- リレー(継電器)によるデジタルコンピュータ
- 真空管によるデジタルコンピュータ
- トランジスタによる(デジタル)コンピュータ
- ICによるコンピュータ
- LSIによるコンピュータ
- ダウンサイジング(どんどん小さくなる)の流れ  
メインフレーム(大型コンピュータ)から  
ミニコン、オフコン、パソコンへ
- マイクロプロセッサ、パーソナルコンピュータ(PC、パソコン)
- 個人が所有するコンピュータ
- 機器に組み込まれたコンピュータ(携帯電話、ゲーム機、家庭電気製品、自動車など)
- 別の流れは、並列コンピュータ、スーパーコンピュータ、グリッドコンピュータなど

# 情報技術の進歩(ネットワーク)

- 専用線による遠隔端末からコンピュータを使用  
(ホスト集中処理)
- 専用線によるコンピュータネットワーク  
(分散処理のはじまり クライアントサーバ処理へ)
- 電話線によるコンピュータネットワーク
- インターネットの出現(ARPAネット)
- LANの出現、高速化 物理層IEEE802.3  
10Mbps 100Mbps 1000Mbps
- プロトコルの自然淘汰(TCP/IPへ)  
ISO対TCP/IP TCP/IPに淘汰された
- 広域ネットワークの帯域幅の拡大(ブロードバンド)  
アナログ回線 ADSL 光ファイバー  
(クライアントサーバ処理 Webアプリケーションへ)
- 無線通信ネットワーク、高速化 物理層IEEE802.11

# 情報技術の進歩(ソフトウェア作成)

- マシン言語によるプログラミング
- アセンブラ言語によるプログラミング
- コンパイラ言語、インタープリタ言語によるプログラミング
- ソフトウェア危機 (Software Crisis)
- ソフトウェアエンジニアリング (Software Engineering) の誕生
- 構造化プログラミング
- 大規模ソフトウェアの開発技法の整理 (ウォーターフォール開発プロセス、構造化分析設計技法)
- オブジェクト指向、データ中心
- サービス指向
- クラウドコンピューティング

# 情報技術の進歩

## (ソフトウェアアーキテクチャ)

- **ホスト集中処理 (ホスト - 周辺端末)**
- **ホスト - 遠隔端末 (リモートターミナル)**
- **分散処理、コンピュータネットワーク**
- **クライアント・サーバ**
- **3層 (three tiered) クライアント・サーバ**  
(クライアント + サーバ + バックエンド (DB))
- **Webアプリケーション**  
(クライアントとサーバとの組み合わせが地理的に自由になったが、論理的には固定的) (URLで接続)
- **クラウドコンピューティング**  
(クライアントとサーバとの組み合わせが完全に自由、クラウド側でサーバとの対応の面倒をみている)

# 情報システム構築技法の変遷(1)

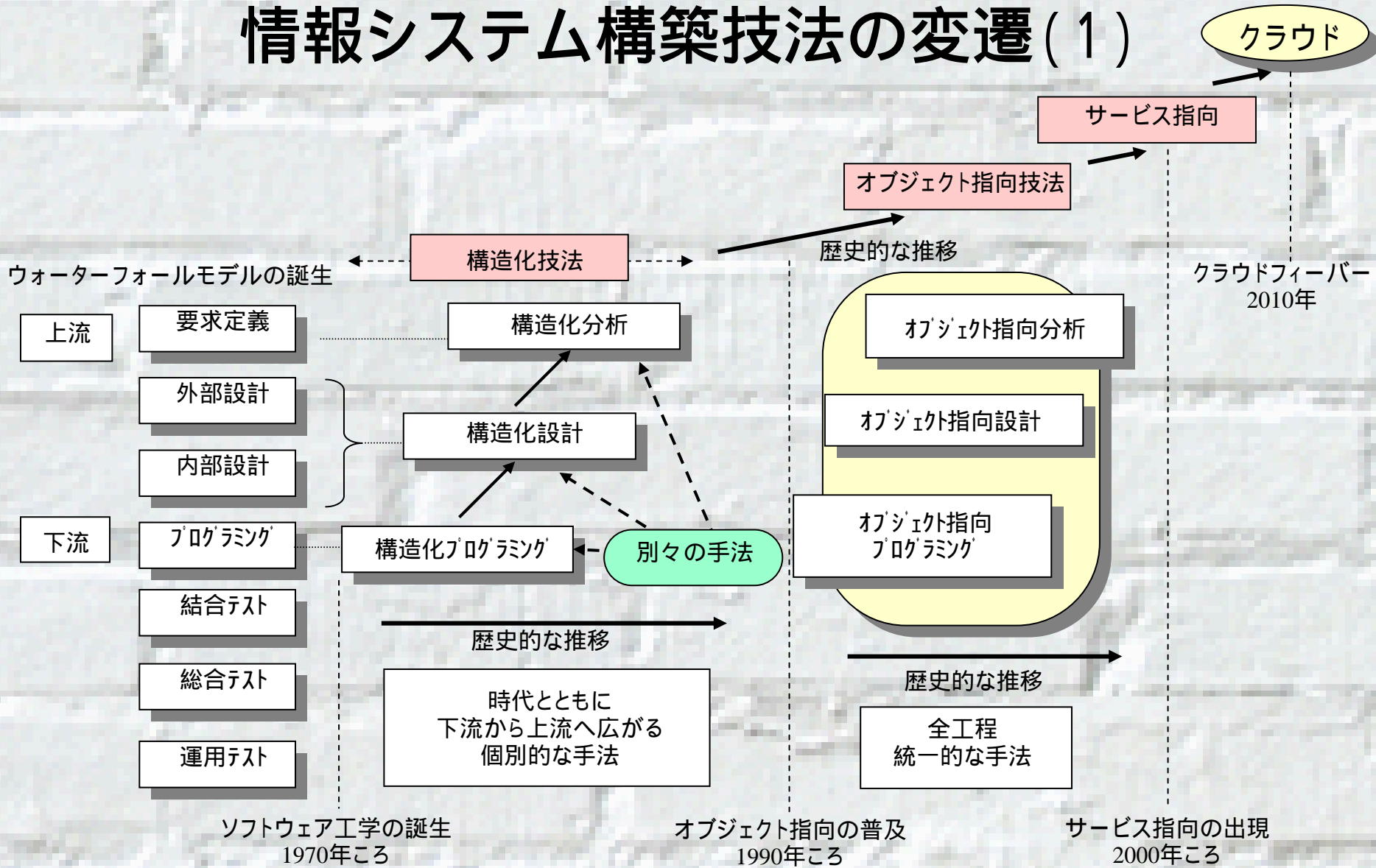


図 ソフトウェア工学の誕生から、構造化技法、オブジェクト指向技法、サービス指向、クラウドへ

# 情報システム構築技法の変遷(2)

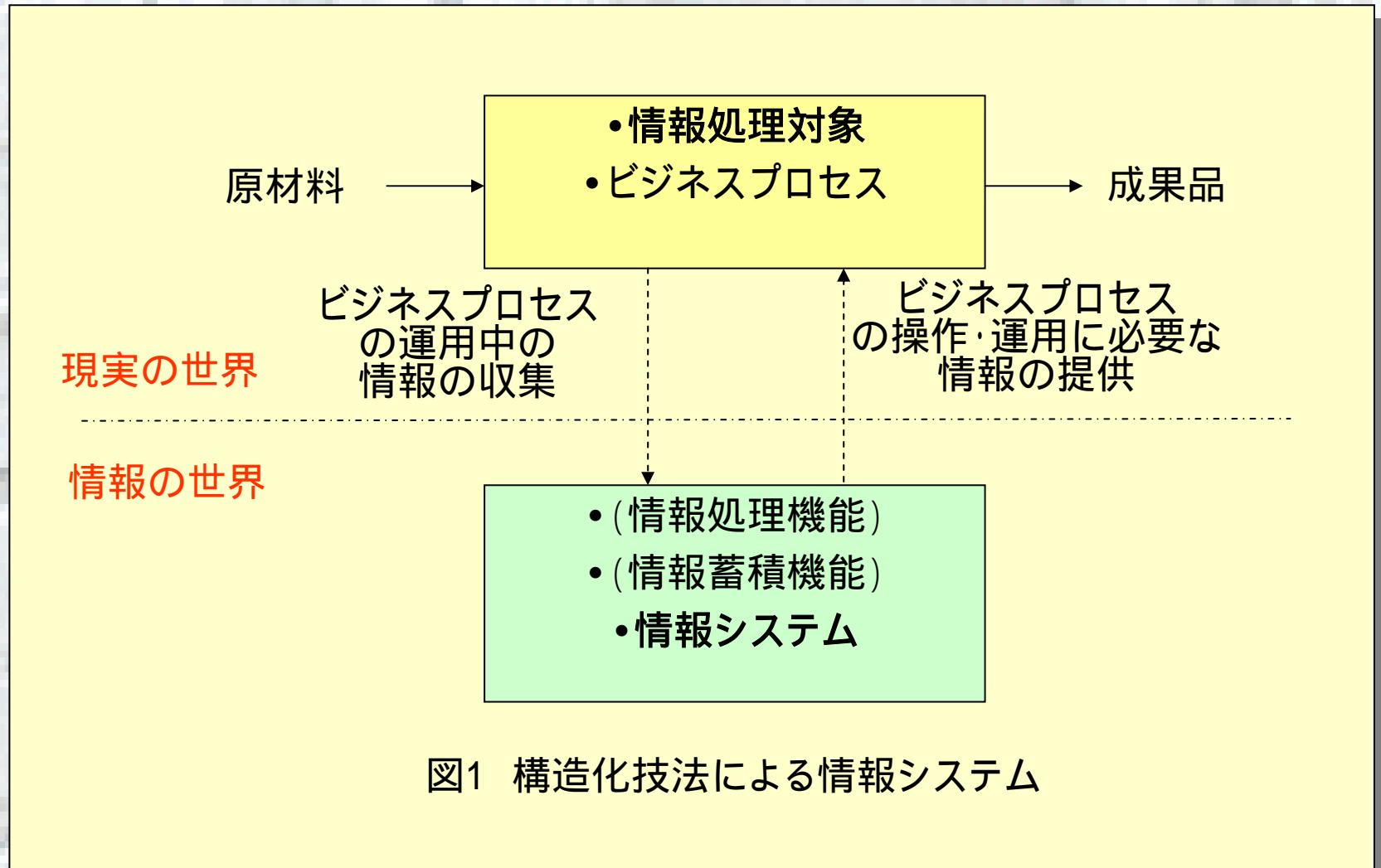


図1 構造化技法による情報システム



# 情報システム構築技法の変遷(3)

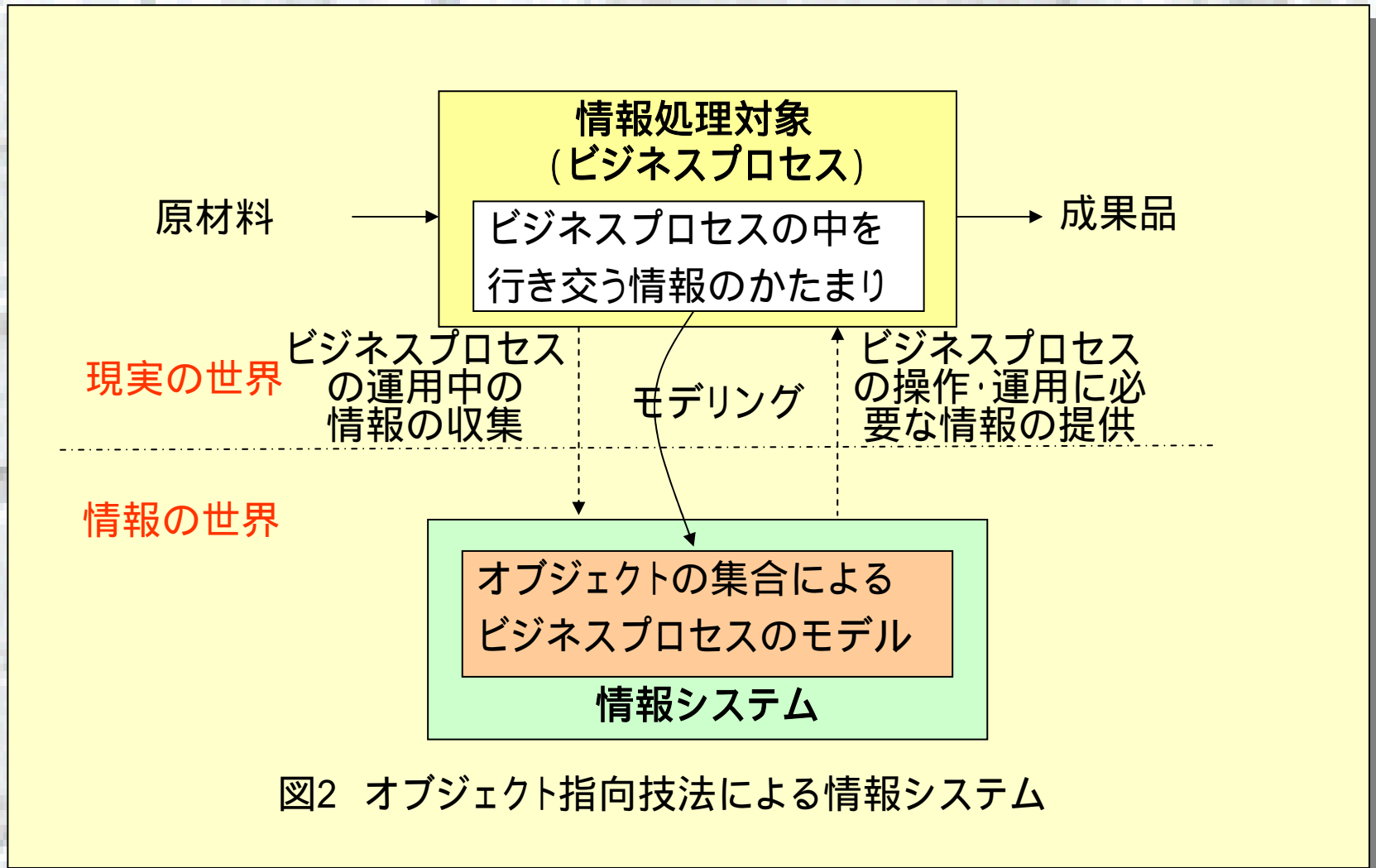
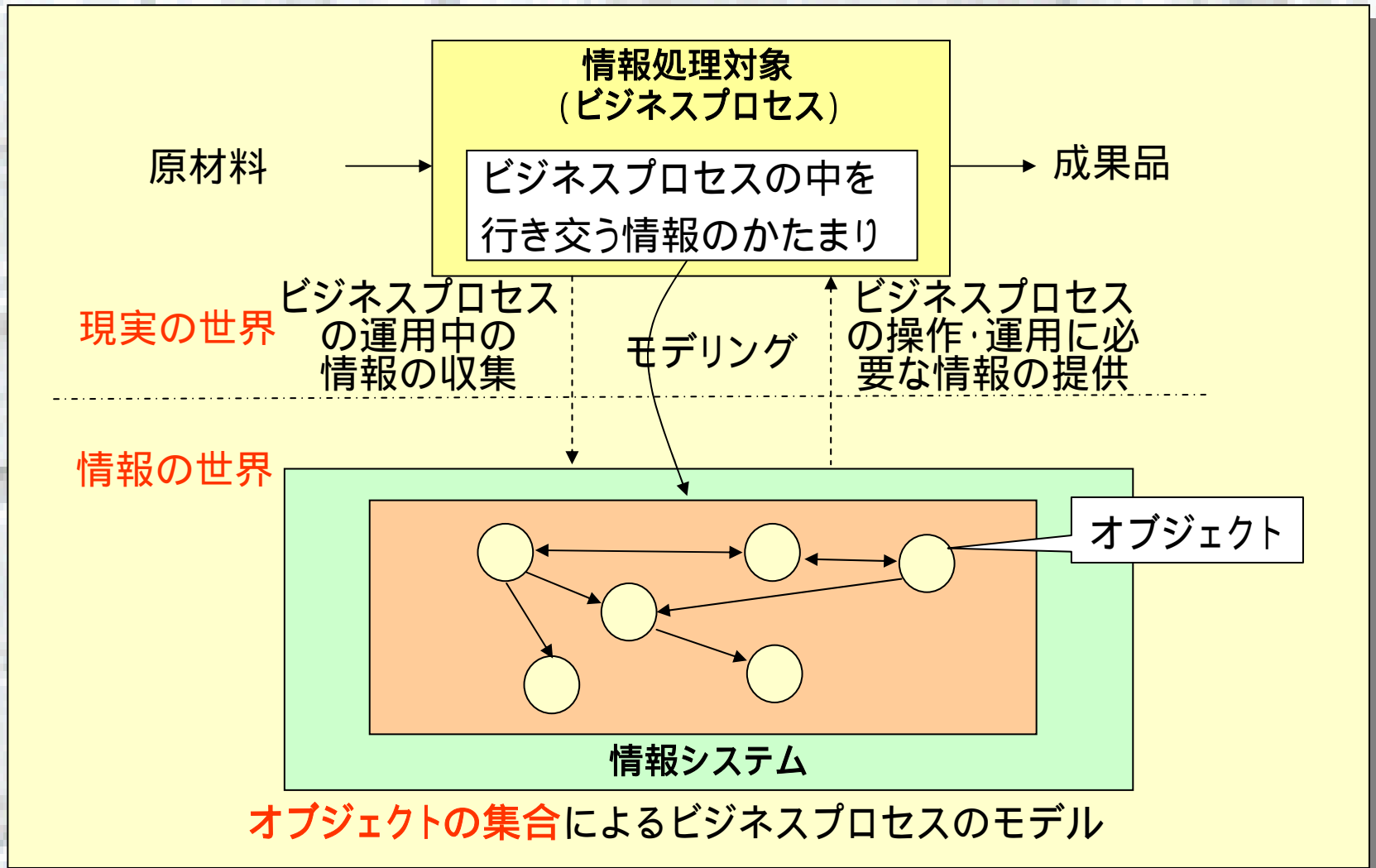


図2 オブジェクト指向技法による情報システム

# 情報システム構築技法の変遷(4)



# 情報システム構築技法の変遷(5)

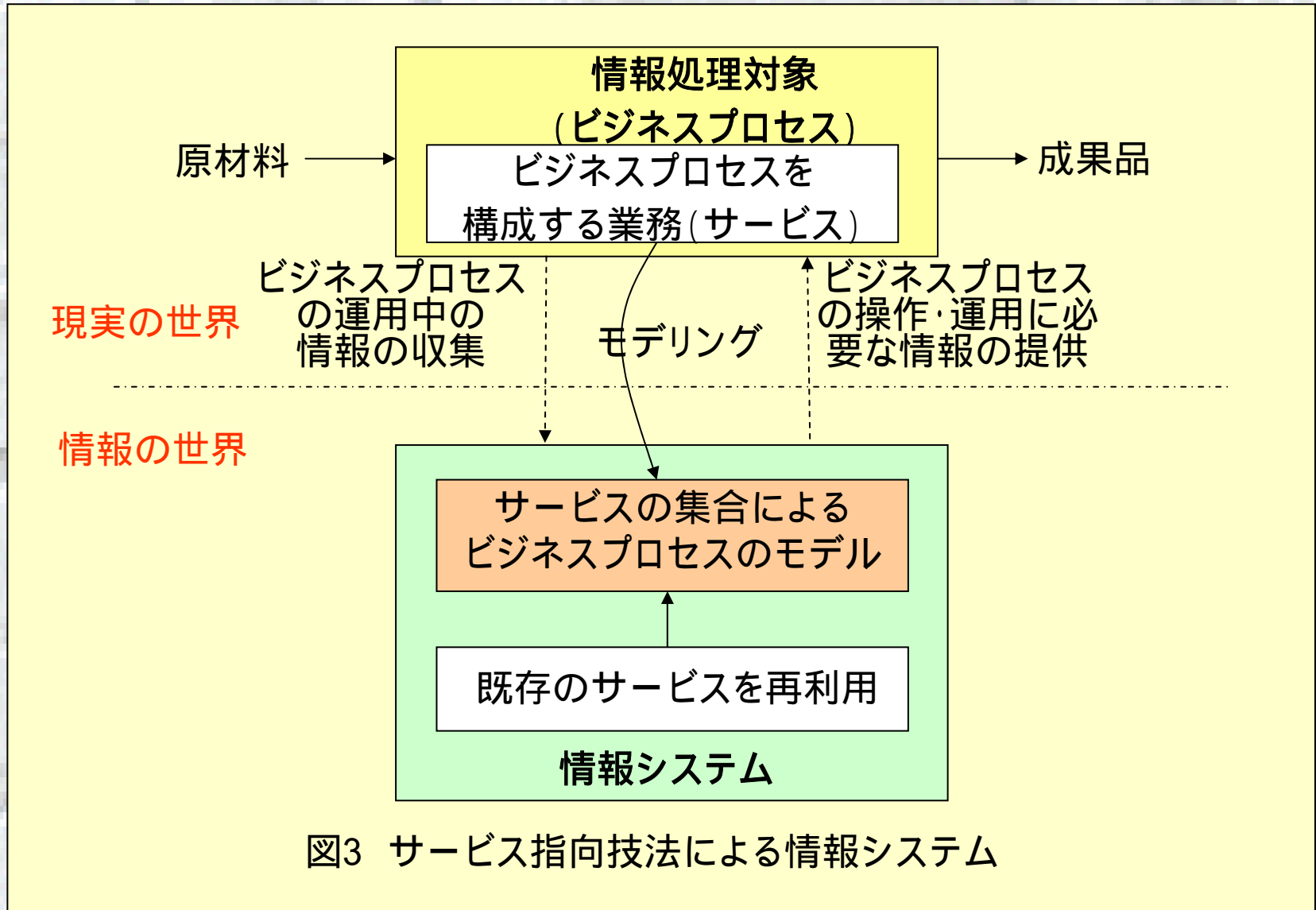
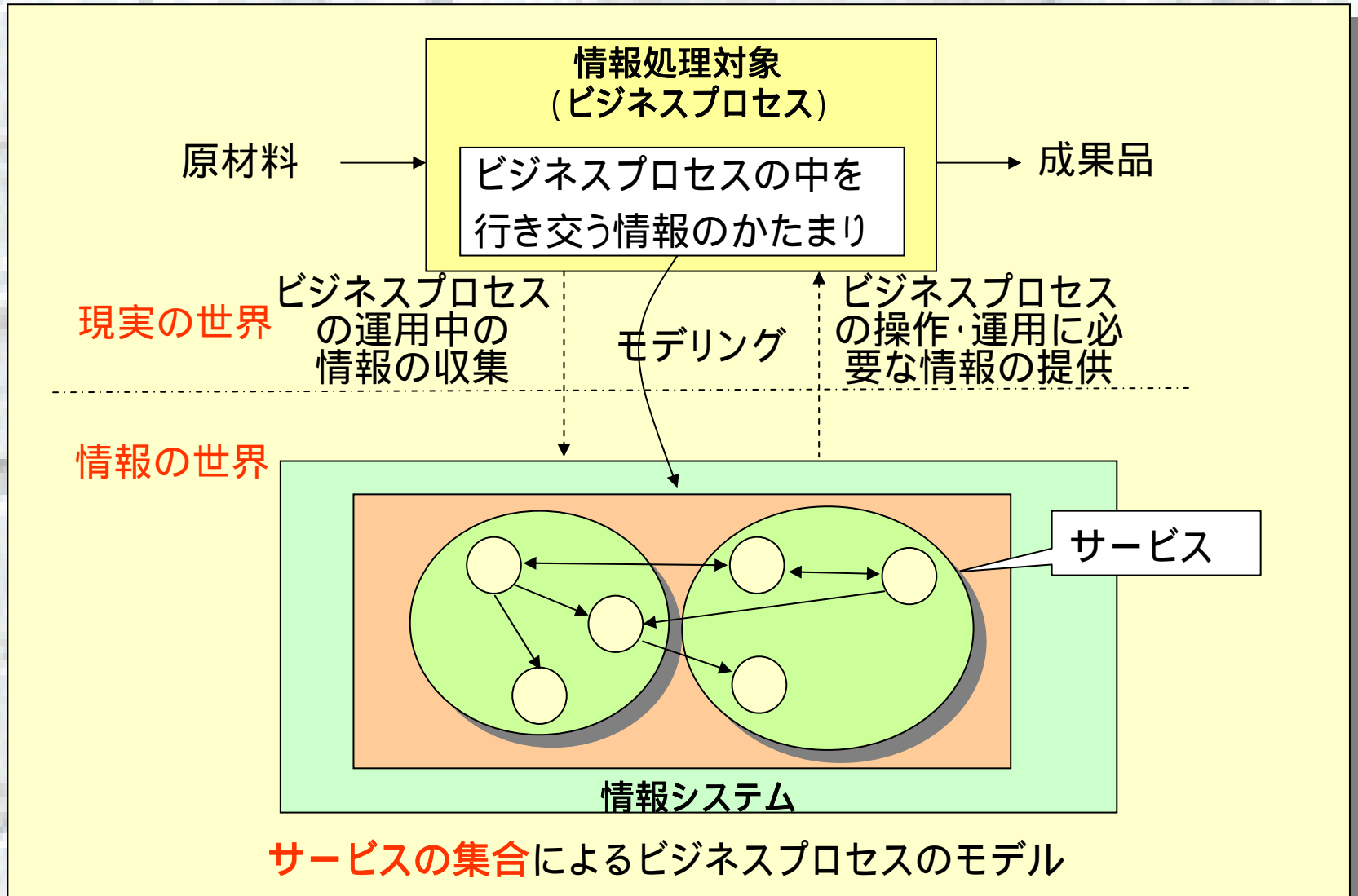


図3 サービス指向技法による情報システム

# 情報システム構築技法の変遷(6)



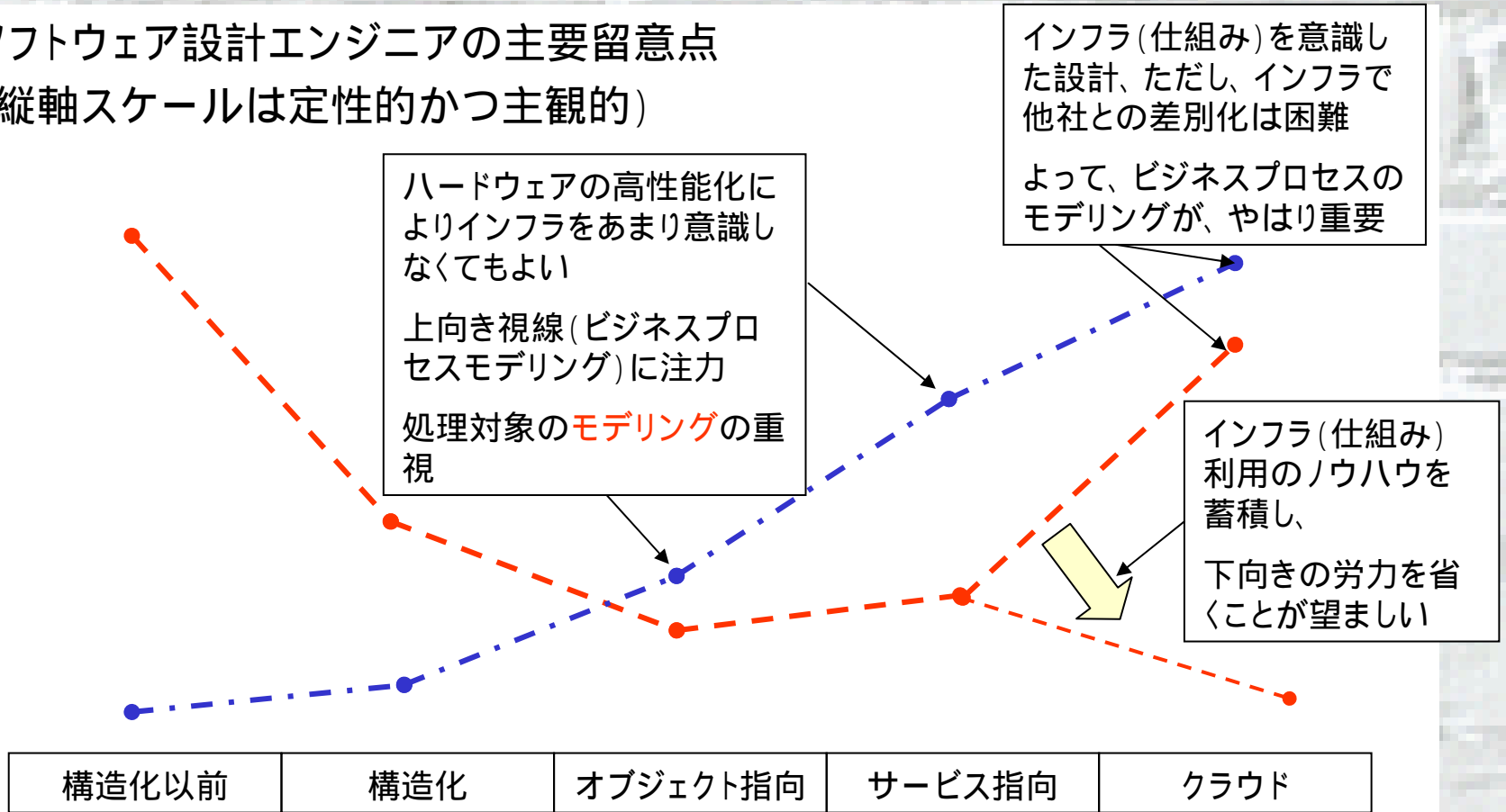
# 情報システム構築技法の変遷(7)

## ソフトウェア設計エンジニアの主要留意点

1. **ソフトウェア工学以前 (Before Software Engineering)の時代**  
ハードウェアリソース(主メモリの容量、実行ステップ数)に留意。
2. **構造化の時代**  
スパゲッティプログラムの排除(goto less)、ソフトウェアの構造に留意、「**接続、選択、反復**」のパターンにはめる。
3. **オブジェクト指向の時代**  
ソフトウェア体系の中を動き回る情報の塊(オブジェクト)に留意、できるだけ自律的単位(属性、メソッドをもつ)としてまとめ、不要な情報は隠蔽する。  
人間社会のモデリング(モデル化)、情報の擬人化、そのモデリング。
4. **サービス指向の時代**  
オブジェクトの粒度をさらに大きく、サービスという単位で捉える。  
3~4の流れで、アプリケーションプログラムの視線は、上へ上へと、用途(業務)のほうを向いてきた。ビジネスプロセスモデリングなど。
5. **クラウドコンピューティングの時代**  
再び、インフラ(仕組み、メカニズム)を意識する必要がでてきた。  
(PaaS、SaaSを利用した情報システムの構築)

# 情報システム構築技法の変遷(8)

ソフトウェア設計エンジニアの主要留意点  
(縦軸スケールは定性的かつ主観的)



ハードウェアの高性能化によりインフラをあまり意識しなくてもよい  
上向き視線(ビジネスプロセスモデリング)に注力  
処理対象の**モデリング**の重視

インフラ(仕組み)を意識した設計、ただし、インフラで他社との差別化は困難  
よって、ビジネスプロセスのモデリングが、やはり重要

インフラ(仕組み)利用のノウハウを蓄積し、  
下向きの労力を省くことが望ましい

● - - - ●

下向きの視線(インフラ(仕組み)、ハードウェア)への注力度

● · · · · · ●

上向きの視線(ビジネス、業務、プロセス)への注力度

# 情報システム構築技法の変遷(9)

## 分散システム構築方法が目指してきた方向(原則)の1つ

透過性(Transparency)

分散していることを意識しなくても良い、位置透過性

## オブジェクト指向が目指してきた方向(原則)の1つ

情報隠蔽(Information hiding)

オブジェクト内部のことは外にださない、自律動作

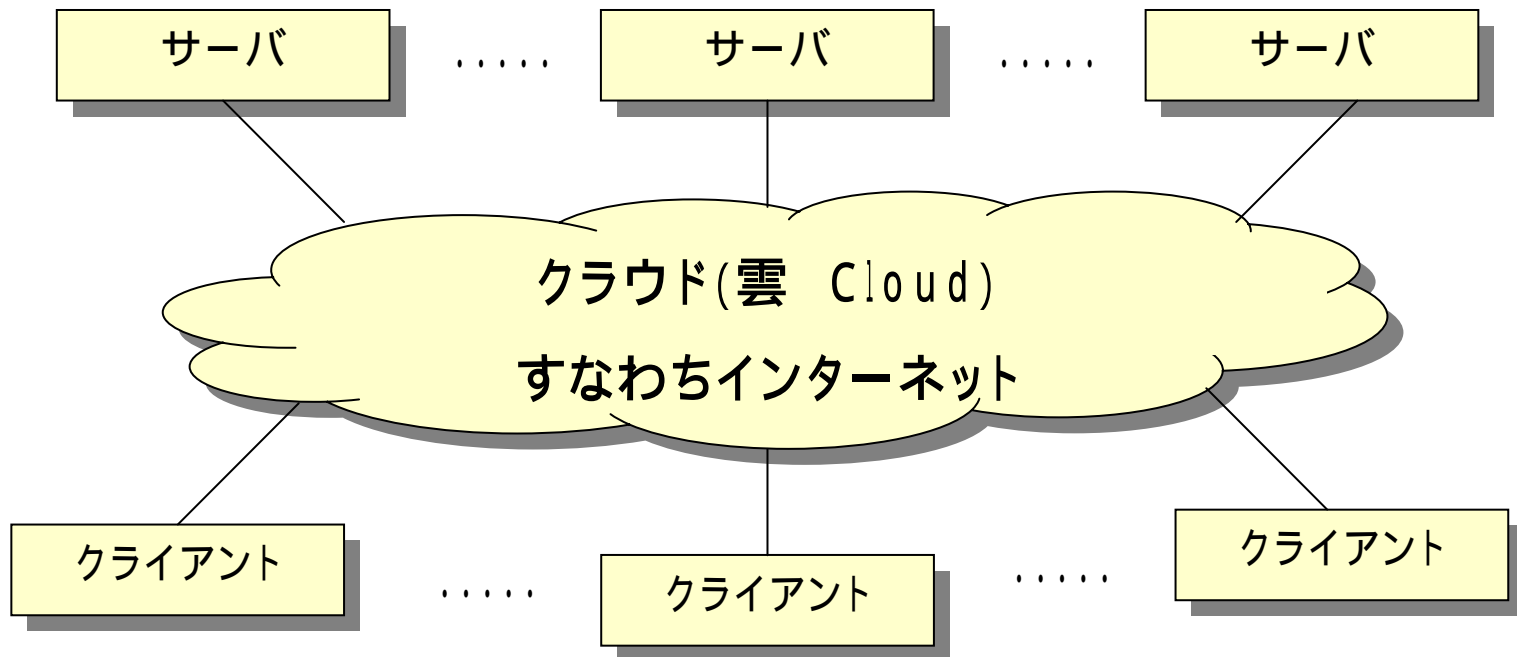
どちらも、上位レベルの機能を実現するために働いている下位レベルの事柄は、上位レベルから見え(意識し)なくてもよい。見えないようにする。

## クラウドコンピューティングでは、

サーバの所在場所はユーザからまったく見えない。その意味では透過的である。ただしソフトウェア構築の際、インフラ(仕組み)を意識する必要がある。

# クラウドコンピューティングとは(1)

## 雲の上のコンピュータ

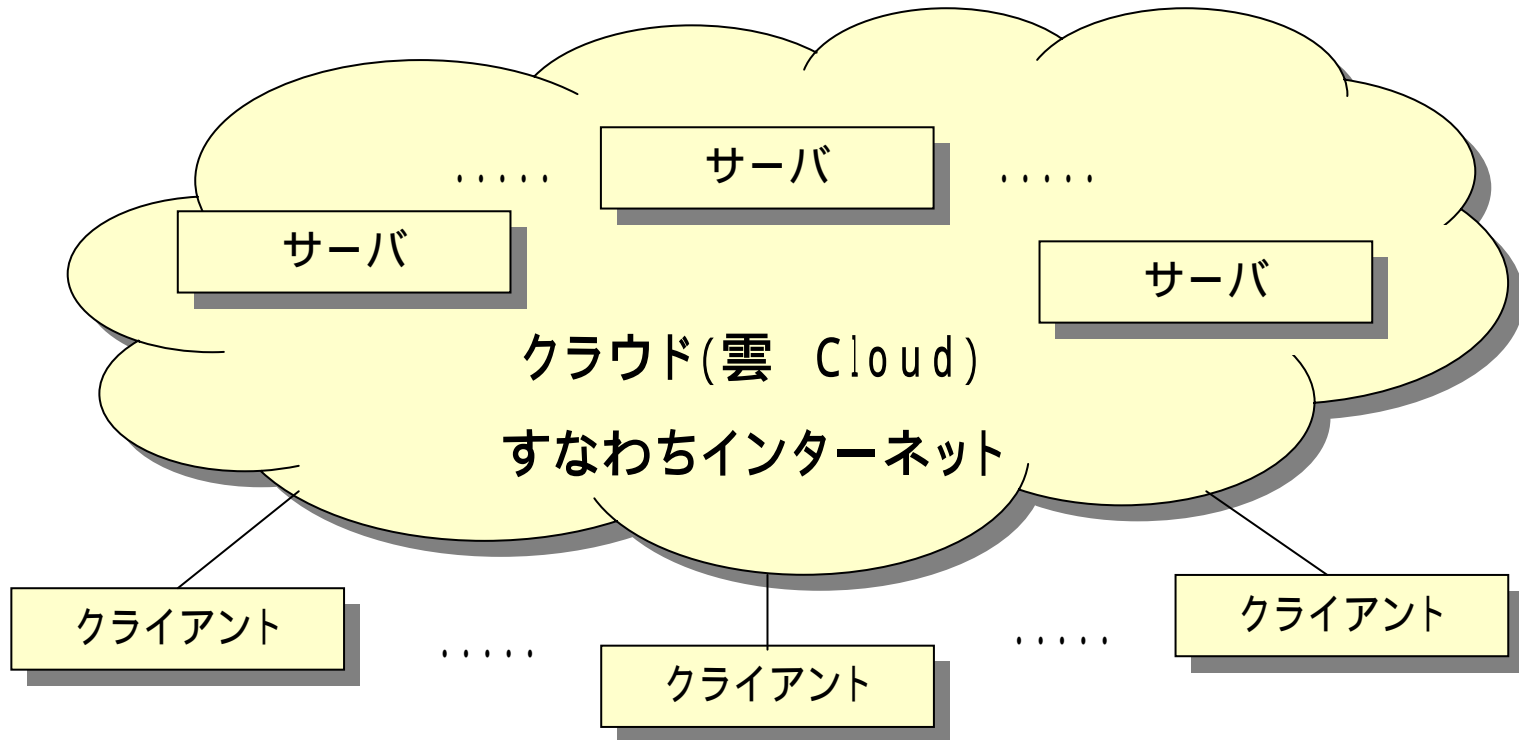


クライアントからの情報処理要求は、雲の上(インターネット上)にある、場所を意識しなくても良いどこかのサーバで処理されて、結果が戻ってくる。このような形態での情報処理をクラウドコンピューティングという



# クラウドコンピューティングとは(2)

## 雲の中のコンピュータ



Wikipedia クラウドコンピューティングとは、ネットワーク、特にインターネットをベースとしたコンピュータの利用形態である。ユーザはコンピュータ処理をネットワーク経由で、サービスとして利用する。

# クラウドコンピューティングとは(3)

## NISTでのクラウドコンピューティングの定義

5つの本質的特徴(Essential Characteristic)

- 「On-demand self-service」
- 「Broad network access」
- 「Resource pooling」
- 「Rapid elasticity」
- 「Measured Service」

サービスのレイヤ(Service Model)によって、

- IaaS(Infrastructure as a Service) ,
- PaaS(Platform as a Service) ,
- SaaS(Software as a Service)

サーバ配置(Deployment Model)によって、

- プライベートクラウド、
- コミュニティクラウド、
- パブリッククラウド、に分類され、これらの混合である
- ハイブリッドクラウドの形態もある

# クラウドコンピューティングの メリット / デメリット

## メリット

- 自社にサーバを設けなくてもよい(初期費用、維持費用の節減)
- スケール(ユーザ数、データ量などの規模)について気にしなくてもよい(従来とは桁違いのスケラビリティをもつ)
- 広い範囲からサービス(機能)を借用することにより、開発期間を短縮できる可能性がある(開発の高速化)
- 総合的に安価に情報システムを構築できる可能性がある(使用期間にわたる総合費用の節減) TCO(Total Cost of Ownership)からTCU(Total Cost of Usage)へ
- 大規模計算を並列処理で高速化できる可能性がある(処理の高速化)

## デメリット

- 外部にデータが漏れる恐れがある
- サービス利用に費用がかかる(運用費用)
- ブラックボックスの利用で、トラブル時の対応に不安
- ブラックボックスの利用で、テストのやりかたに注意が必要
- 従来のデータベース、トランザクションの考えを使えない
- インフラを意識したシステム構築が必要 ノウハウ獲得の要あり
- 任意の情報処理と言うより、元はと言えば、やはり大量の情報を検索するしくみから発生していることがいろいろな面での制約になっている(ようだ)

# クラウドコンピューティングは本命であろうか

では、一体、クラウドコンピューティングは、エンタープライズ用途(ビジネス系、業務系)の情報システムに本格的に使用できるのでしょうか？

- ・ビジネスの遂行に役立つのでしょうか？
- ・さらに一歩進んで、ビジネスの革新(イノベーション)を推進する原動力になるのでしょうか？

もし、YESならば、クラウドコンピューティングを用いたエンタープライズ用の情報システムの構築方法を早急に獲得する必要がある。

従来の分散システム構築方法と、根本的に異なってくる可能性がある。早くノウハウを獲得すれば、アドバンテージになる。

電子情報通信学会SWIM研究会での取り組み

「クラウドコンピューティングトライアルプロジェクトCCTP」

# クラウドコンピューティングの要素技術(1)

## ユーザ(クライアント)とプロバイダ(サーバ)間のインタフェース技術

ユーザとプロバイダの間は、通常は標準化されたインターネットの技術が使用されるが、専用の技術(プロトコル、ソフトウェア、ハードウェア)を使用するものもある。(Wikipediaから引用)

## プロバイダ内部の技術

### 従来技術を利用可能とする考え

オープン標準に準拠したソフトウェアや、ユーザ数や処理量の増減に対応できる仮想化技術が使用される。(Wikipediaから引用)

### クラウド特有の技術を前面に押し出す考え

Googleなどは、スケーラビリティ確保のために自社独自技術を多様している。(Wikipediaから引用)

この場合、クラウドの特長を最大にひきだすため、クラウド特有技術を利用して情報システムを構築するには、独自技術(クラウドを実現している仕組み)をよく理解し、その条件下で構築することが必要となる。

# クラウドコンピューティングの要素技術(2)

## データベース

リレーショナルDBからKey Valueデータストア (NoSQL)

「スケーラビリティ」を厳しく優先する場合、リレーショナルDBは使用できない

## トランザクション

ACIDトランザクションからBASEトランザクションへ

上記のデータベースと関係、(スケーラビリティ優先の場合)

## 並列処理

MapReduce

## 仮想化技術

1つの物理的なCPUに複数の論理的なCPUを保持する

複数の物理的なCPUを1つの論理的なCPUとして働く 分散処理

従来型(非クラウド)情報システムを比較的容易にクラウド化する方法で、従来システムへの影響は少ない

## スケールアウト

担当サーバの割り当て、探索 (ハッシング)(コンシステントハッシング)

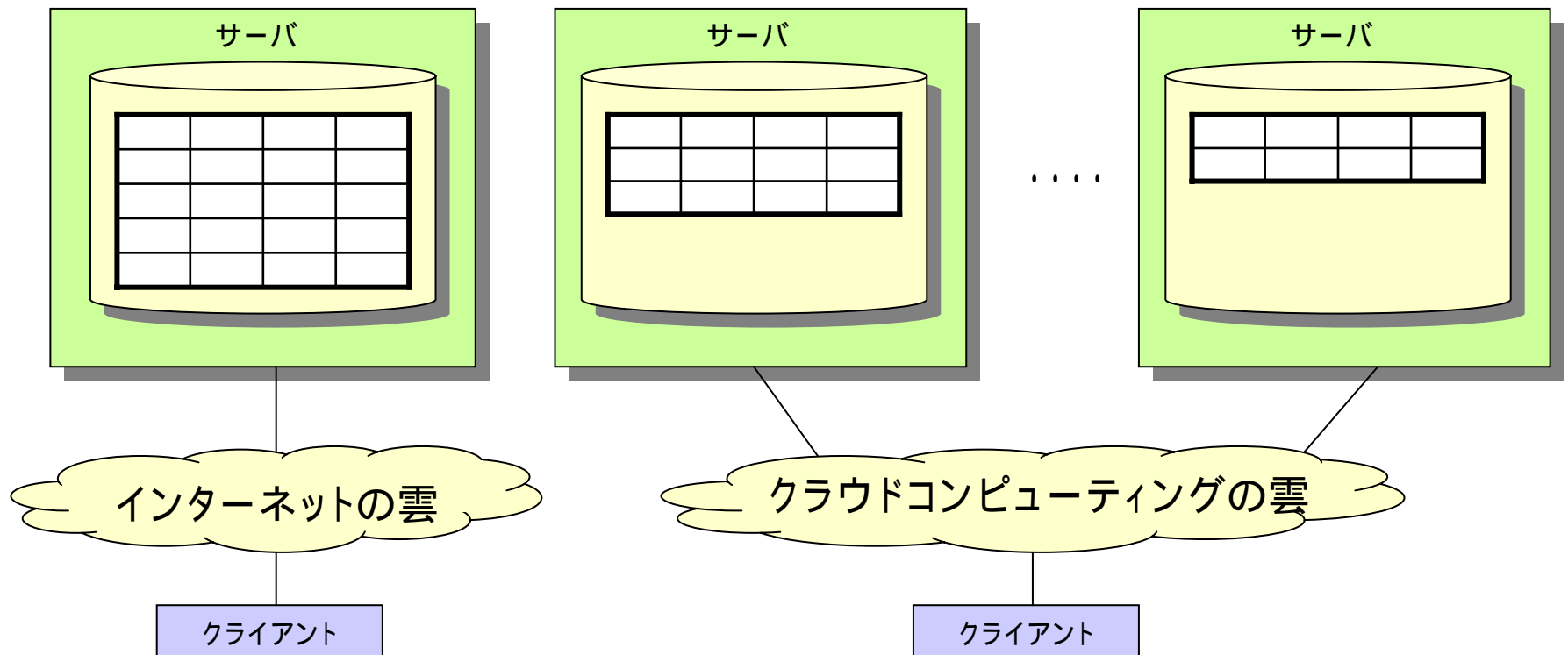
担当サーバへの到達、ルーティング(構造化オーバーレイ)

# クラウドコンピューティングの要素技術(3)

データベース リレーショナルDBからKey Valueデータストア (NoSQL)

クラウド前

クラウドコンピューティング



i 行(タプル)

PK i	Vi1	Vi2	Vi3
------	-----	-----	-----

基本的に、すべての行が1つのサーバに配置される (PK: Primary Key)

Key Valueデータ

i 番目

Key i	Vi1	Vi2	Vi3
-------	-----	-----	-----

基本的に、この単位で、多数のサーバにバラバラに配置される

# クラウドコンピューティングの要素技術(4)

## トランザクション

ACID (Atomicity, Consistency, Isolation, Durability)トランザクションから

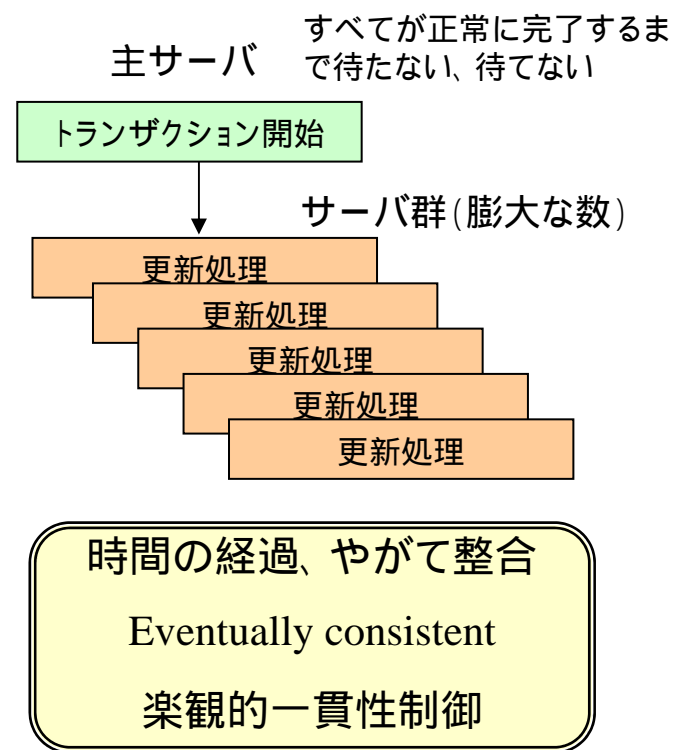
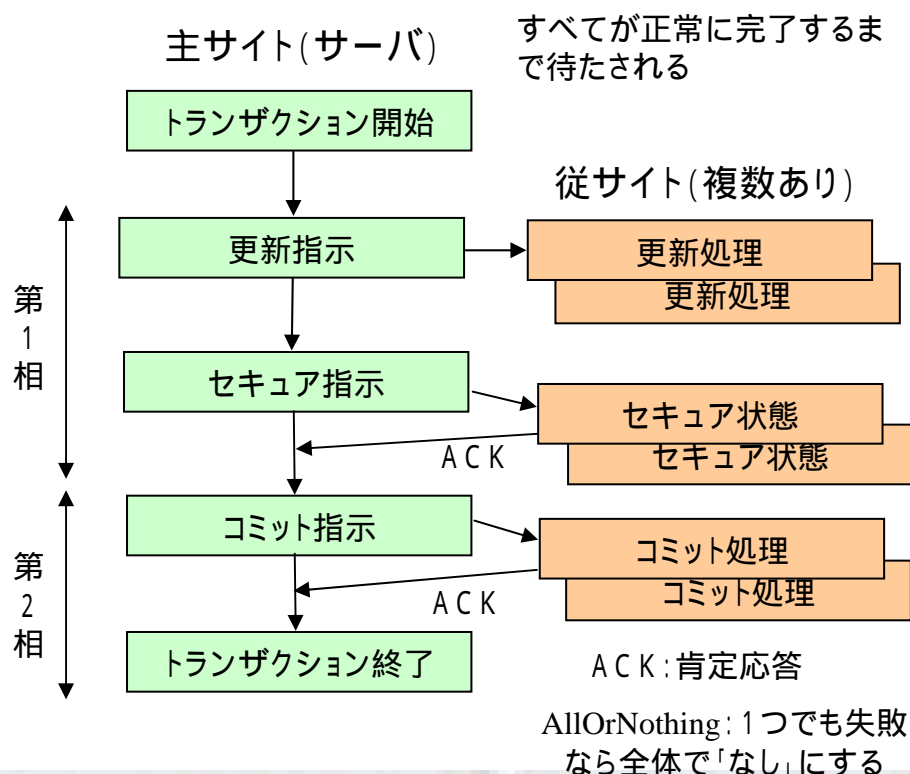
BASE (Basically Availability, Soft state, Eventually consistency)トランザクションへ

一貫性(整合性)の維持

強い一貫性制御方式 ACID, 2PC, WAL

一貫性(整合性)の維持

弱い一貫性制御方式 BASE



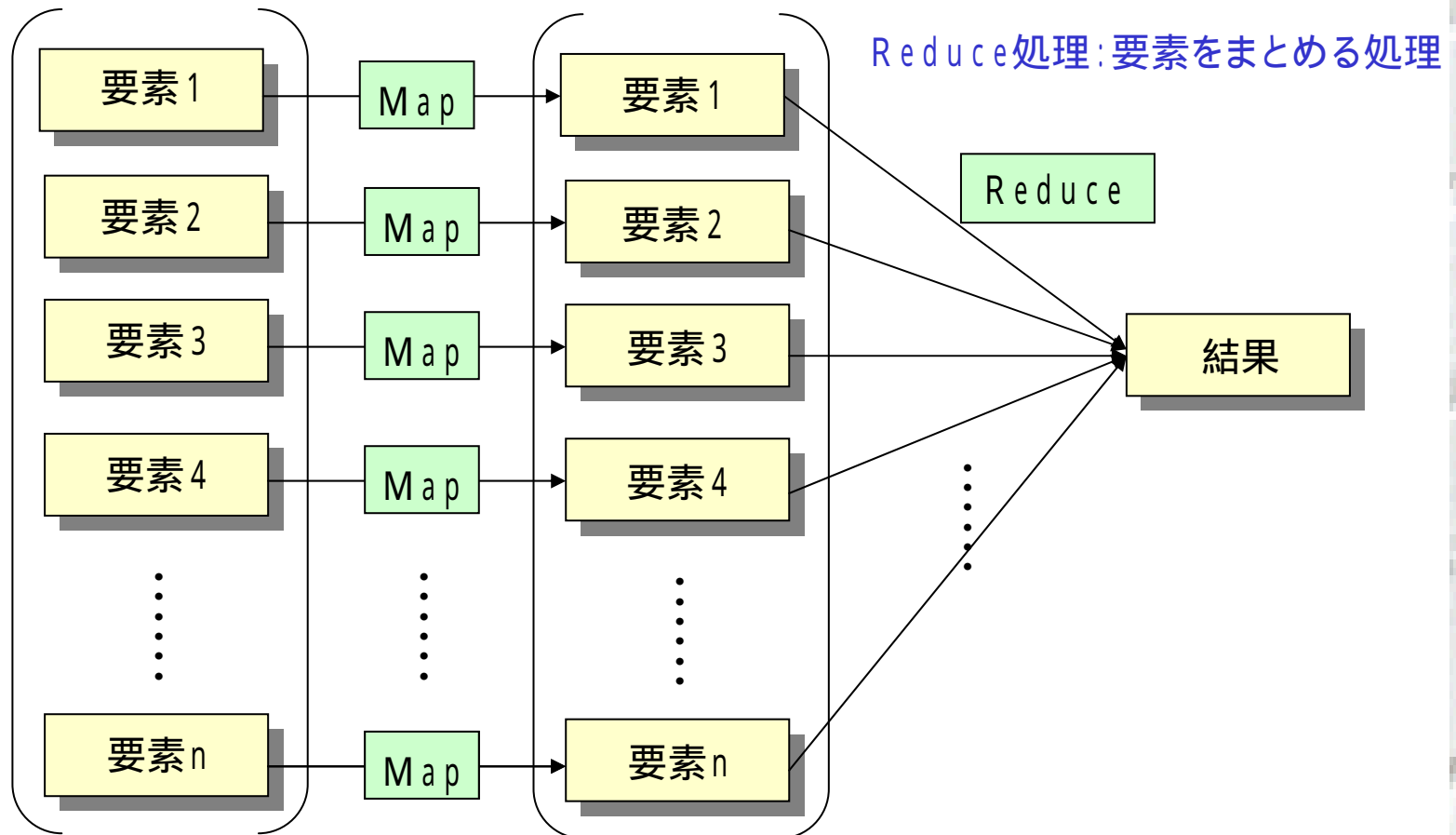


# クラウドコンピューティングの要素技術(5)

## 並列処理

MapReduce

Map処理:要素ごとに行う処理



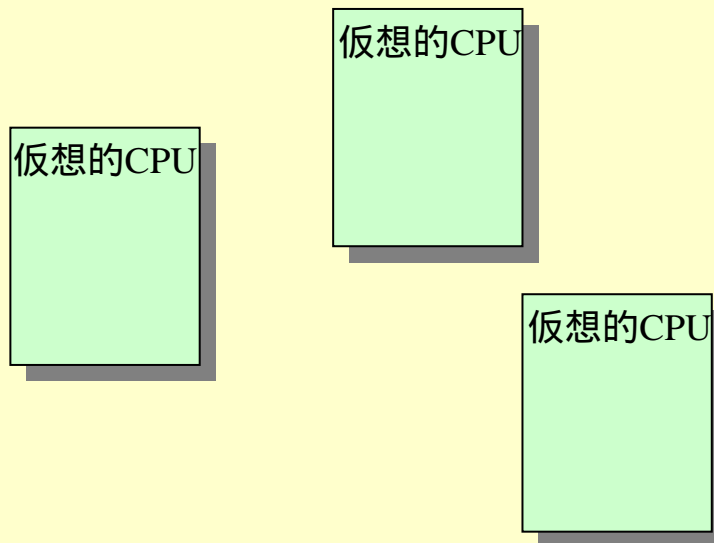
# クラウドコンピューティングの要素技術(6)

## 仮想化技術

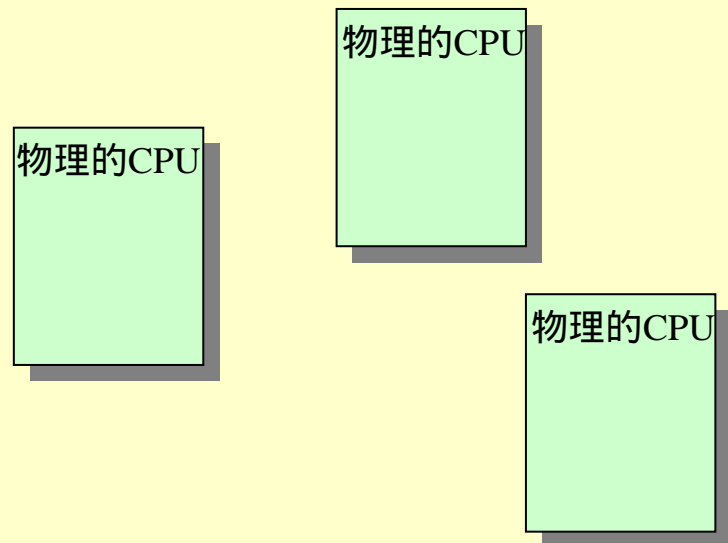
1つの物理的なCPUに複数の論理的なCPUを保持する

複数の物理的なCPUを1つの論理的なCPUとして働く 分散処理

CPU(物理的なサーバマシン)



仮想的CPU(論理的なサーバマシン)



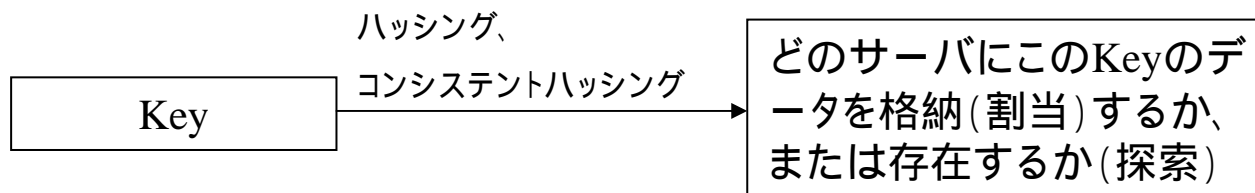
# クラウドコンピューティングの要素技術(7)

## スケールアウト

担当サーバの割り当て、探索 (ハッシング)(コンシステントハッシング)

担当サーバへの到達、ルーティング(構造化オーバーレイ)

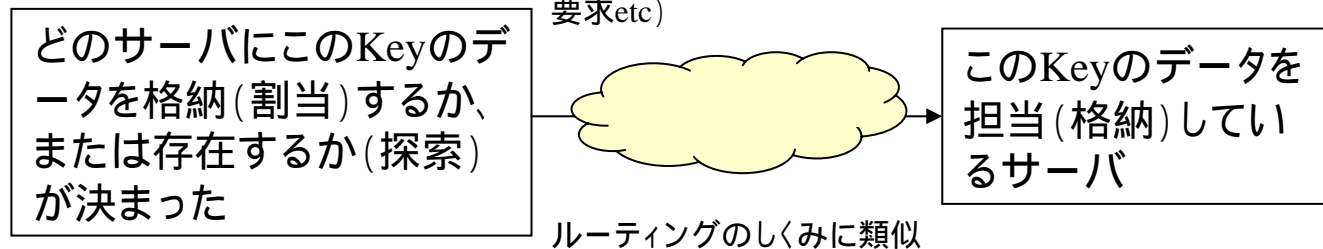
担当サーバの割り当て、探索



クラウド(インターネット)の中を通過して、どうやって、到達するか

担当サーバへの到達

(データ格納の要求、データ送信の要求、データ処理の要求etc)



# クラウドコンピューティングの要素技術(8)

(各社の比較、参考文献(3)から作成、要継続調査)

	Google	Amazon EC2	Windows Azure	Salesforce.com
Key Value データストア	GFS (Google File System), Bigtable	Amazon Simple DB	Windows Azure Table	×
RDB	×	Amazon RDS(MySQL)	SQL Azure	Force.com Database
ACID トランザクション	限定的な対応(同一ノード上のデータ間)			
BASE トランザクション				
並列処理	MapReduce	Amazon Elastic MapReduce		

# クラウドコンピューティングサービス提供者

## マイクロソフト

(PaaS)

.Net、Azure

(SaaS)

Microsoft Online Services

## グーグル

(PaaS)

Google App Engine

(SaaS)

Google Apps

SaaS
PaaS
IaaS

## アマゾン

(IaaS)

Amazon EC2、 Amazon S3

(SaaS : Software as a Service)

(PaaS : Platform as a Service)

(IaaS : Infrastructure as a Service)

## セールスフォース

(PaaS)

Salesforce AppExchange

(SaaS)

Salesforce CRM

(SaaS : アプリケーションの提供)

(PaaS : アプリ構築の基盤提供)

(IaaS : HW、基本ソフトウェア提供)

# 参考文献

- (1) Wikipedia, “クラウドコンピューティング,”  
<http://ja.wikipedia.org/wiki/%E3%82%AF%E3%83%A9%E3%82%A6%E3%83%89%E3%82%B3%E3%83%B3%E3%83%94%E3%83%A5%E3%83%BC%E3%83%86%E3%82%A3%E3%83%B3%E3%82%B0>  
2010年5月5日確認
- (2) 松本正雄編著, “Webサービス時代の経営情報技術,” 電子情報通信学会, 2009年2月
- (3) <http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc>
- (4) 丸山不二夫, “クラウドの成立過程とその技術的特徴について,” 情報処理, Vol.50, No.11巻, 号, pp.1055 - 1061, Nov. 2009.
- (5) 中田秀基, “Googleのクラウド技術,” 情報処理, Vol.50, No.11, pp.1062 - 1067, Nov. 2009.
- (6) 石田愛, “Amazon EC2,” 情報処理, Vol.50, No.11, pp.1068 - 1073, Nov. 2009.
- (7) 藤田昭人, “クラウド技術とオープンソース,” 情報処理, Vol.50, No.11, pp.1074 - 1079, Nov. 2009.
- (8) 首藤一幸, “スケールアウトの技術,” 情報処理, Vol.50, No.11, pp.1080 - 1085, Nov. 2009.
- (9) 萩原正義, “クラウドアプリケーションの分析と開発手法,” 情報処理, Vol.50, No.11, pp.1092 - 1098, Nov. 2009.
- (10) 浦本真彦, “クラウドコンピューティングにおけるセキュリティとコンプライアンス,” 情報処理, Vol.50, No.11, pp.1099 - 1105, Nov. 2009.
- (11) たとえば, 2010年5月12日日経新聞一面記事, “パナソニック クラウド使い生産管理 システム費, 4割削減”
- (12) <http://www.ieice.org/~swim/jpn/CCTPPlan.pdf>

# おわりに

クラウドコンピューティングは、本当に、企業用または行政用の情報システムに役立つものかどうかを見極めなければなりません。

もし、そうならば、そのための知識と実践方法を身につけなければなりません。

一般論では、世の中の変化に常に敏感で、世の中の変化に取り残されないようにしましょう。

さらに願わくば、私たちが世の中の変化を惹き起こす原動力になれるように精進しましょう。

ご清聴(ご注目)、ありがとうございました。