



# 「ITアーキテクチャのセオリー」

～ システム構築の大前提はアーキテクチャを決めること～



2018年10月25日 スクエアfreeセミナー第96回  
“情報システムにセオリーはあるか？”  
(株) アイ・ティ・イノベーション 中山 嘉之



## ～ I Tアーキテクチャのセオリー～

### 第 I 部 何をなすべきか

- 第 1 章 問題の所在
- 第 2 章 取り組むべき課題は何か
- 第 3 章 取り組みにあたっての留意点

### 第 II 部 アーキテクチャ

- 第 4 章 EA：エンタープライズ・アーキテクチャ
- 第 5 章 DA：データ・アーキテクチャ
- 第 6 章 AA：アプリケーション・アーキテクチャ
- 第 7 章 TA：テクノロジー・アーキテクチャ

### 第 III 部 戦略ソリューション

- 第 8 章 エンタープライズ・データHUB
- 第 9 章 マスタ・データHUB
- 第 10 章 トランザクション・データHUB
- 第 11 章 情報系データウェアハウス

### 第 IV 部 戦術ソリューション

- 第 12 章 リポジトリで情報を可視化
- 第 13 章 ゆるやかなシステム移行へ
- 第 14 章 近未来へ先手を打つ



# I. 何をなすべきか？

# ・問題の所在、取り組むべき課題は？

- ・ エンタープライズ・システムはコモディティではない（個々の部品はコモディティ化が顕著だが）
- ・ ビジネスが差別化を志向するなら、エンタープライズ・システムも差別化に向かう（脱コモディティ化によるケイパビリティ向上）
- ・ 適材適所なハイブリッド・システムは、平均的なモノリシック・システムを機能面で上回る（ERP-onlyの終焉）
- ・ システムの適用範囲の拡大は、単位面積当たりのコスト効果の増大に繋がる（汎化 [= 抽象化] の恩恵）
- ・ 大規模な密結合システムを疎結合化して行くことは、今後の大企業のEAにとって重要な課題である

# ・ 取り組みにあたっての留意点

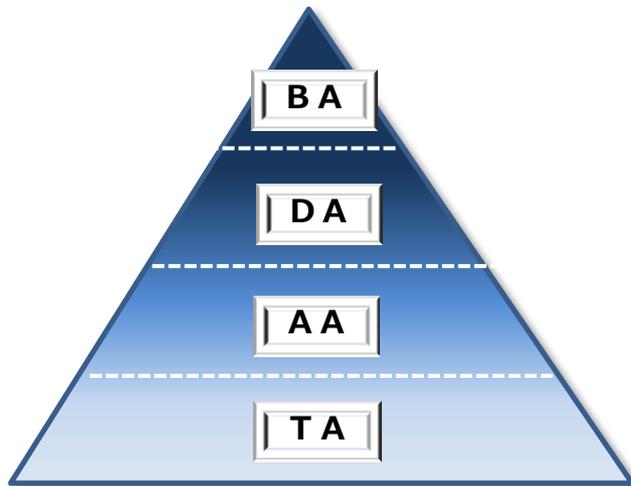
- ・ 大きく全体をモデリングし、小さく実装を始める  
(Think Big [Model] , Start Small [implementation] ! )
- ・ システムのデザインスコープ≠プロジェクトスコープ (WF型開発ではこれを混同するので魅力的なシステムが生まれにくい)
- ・ 全てのエンタープライズデータは何らかの関係を持ちながら企業活動を形成しており、ツール・サービスの種類によって連携が分断されてはならない (システムのサイロ化は企業活動の効率を妨げる)
- ・ SoRとSoEはプレイヤーも適用技術も異なるが、企業システムでは両者が融合されてはじめてインベーションが生まれる (お互いに食わず嫌いをしない)



## Ⅱ. アーキテクチャ（EA）を創る

# ・ ITアーキテクチャの定義

## ■ EAにおけるITアーキテクチャの分類



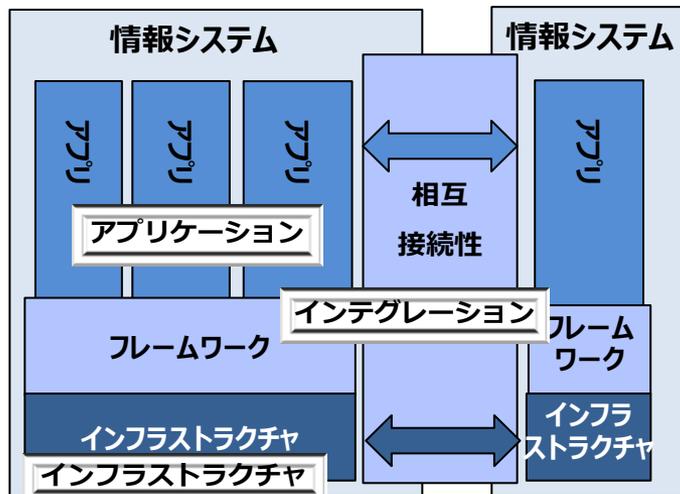
**ビジネスアーキテクチャ：**  
企業全体の組織機能や役割の関連構造が見える化

**データアーキテクチャ：**  
企業内の情報をデータ部品の組み合わせ構造で見える化

**アプリケーションアーキテクチャ：**  
データを更新・参照する情報システム（プロセス）の関連が見える化

**テクノロジーアーキテクチャ：**  
情報システムを構築運用するIT基盤（ハード、ソフト）が見える化

## ■ IPAにおけるITアーキテクチャの分類



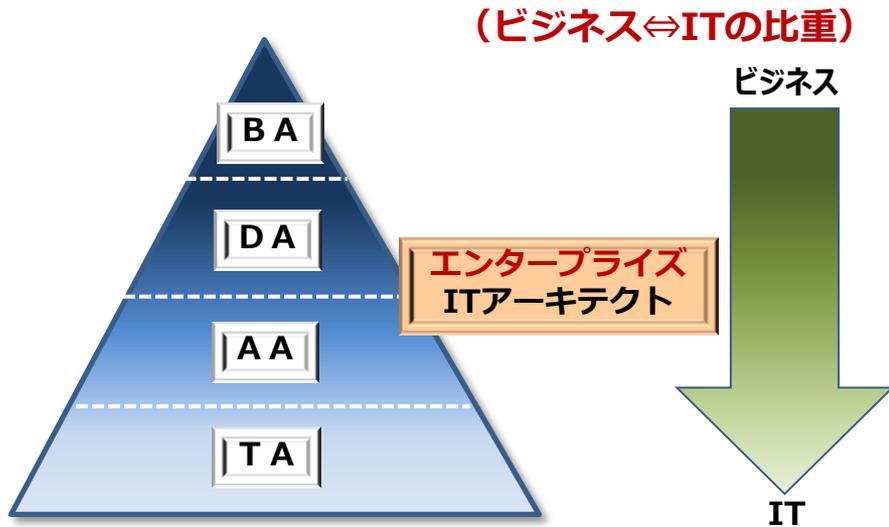
**アプリケーションアーキテクチャ：**  
ビジネス及びIT上の課題を分析し、機能要件として再構成する。機能属性、仕様を明らかにし、コンポーネント構造、論理データ構造等を設計

**インテグレーションアーキテクチャ：**  
全体最適の観点から、情報システム間の統合及び連携要求を分析し、連携要件として再構成する。フレームワークおよび相互接続性を設計

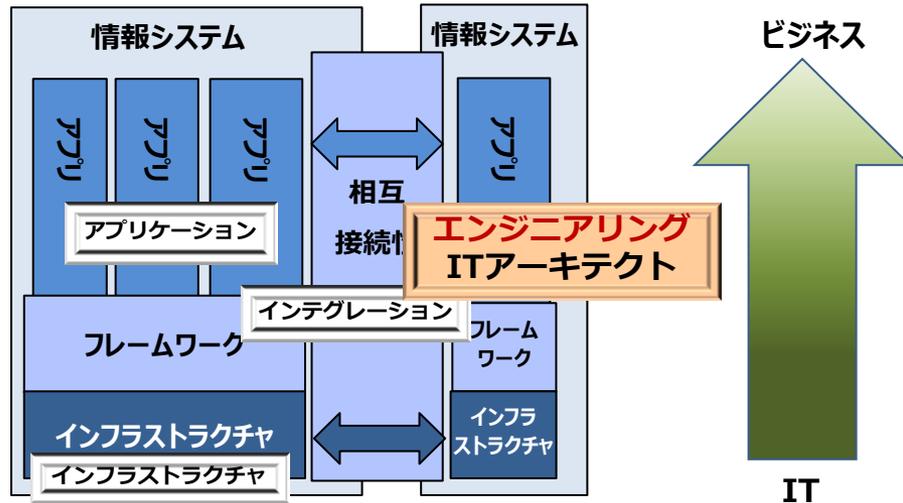
**インフラストラクチャアーキテクチャ：**  
ビジネス及びIT上の課題を分析し、システム基盤要件として再構成する。システム属性、仕様を明らかにし、ネットワーク、プラットフォーム等の基盤を設計

# ・ 2種類のアーキテクトの融合

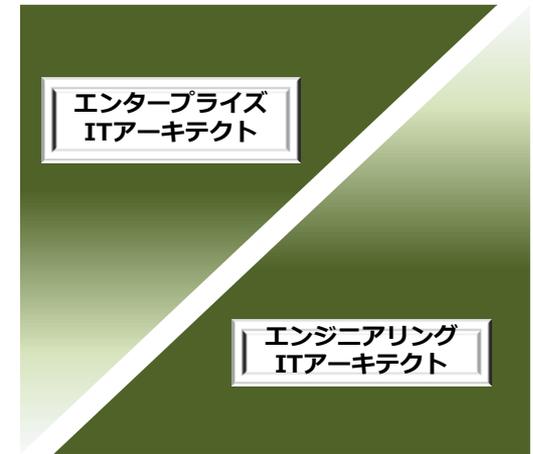
E  
A



I  
P  
A



## 二種類の ITアーキテクトの融合



## • EA (DA、AA、TA) - ①

- **EA** : BA⇒DA⇒AA⇒TAへと役割を継承する4層の視点毎に、対象企業を部品化・見える化することである
- **EA** : 成功の秘訣は、その対象領域にメリハリをつけることにある（こだわる部分は精緻に、そうでない部分は大まかに）
- **DA** : データのライフサイクルはアプリケーション（プロセス）のそれよりも長いゆえに中核に据えるべきである
- **DA** : COPY&PASTE（コピペ）は最もパワフルなIT機能であるが、度が過ぎると、企業システム自らを死に至らしめる
- **DA** : 「One fact in one place」は論理的なものであり、物理的なものではない（疎結合アーキテクチャでは参照用の複製を肯定）

## ・ EA (DA、AA、TA) - ②

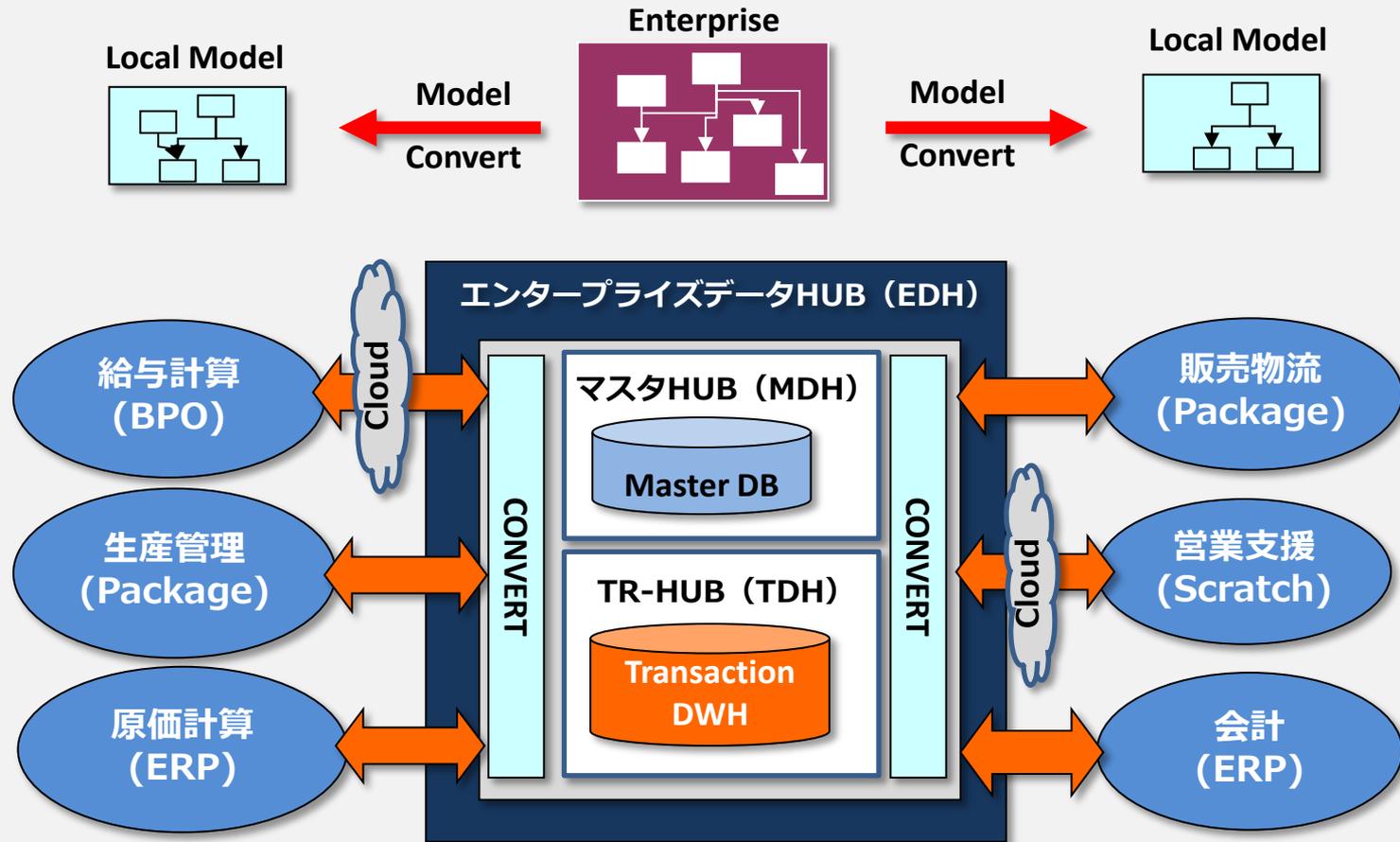
- ・ **AA** : システムの見える化には、「鳥の眼 (バースアイ) 」で、まずは上空から森の全容を掴み、その後に木を見て詳細を理解する
- ・ **AA** : パッケージはその原始的価値を損なうことなく導入すべき (パッケージに自社のシステムを適合させるべくデータモデルを変換)
- ・ **AA** : 疎結合化のポイントは、2つの業務間で共通するイベントトランザクションを遅延同期し、各々が保有する残高を個別更新すること
- ・ **TA** : 将来のシステムを設計する際は、進化の早いTA (ハード・ソフトのインフラ部分) の非機能要件をネックに持ってゆくようなデザインを心掛ける
- ・ **TA** : モダナイゼーションとはビジネス・イノベーションをもたらすものを言い、異種プラットフォームへの単純移行より広範囲な取組みである



## Ⅲ. ソリューション（手段）

# ・エンタープライズデータHUB (EDH)

- ・ 自社独自モデルのエンタープライズデータHUBが中心
- ・ パッケージ等の業務システムは取り替え可能！



## ・ EDHに求められる役割と、あるべき姿

- ・ EDHは、長年のCOPY&PASTEで汚れたシステムを、「共通DBを複数業務でSHAREするアーキテクチャ」に“浄化”する役割を担う
- ・ EDHは、長年の増改築で肥大化した密結合システムを、DBを介したアプリ間疎結合アーキテクチャに転換する要（かなめ）となる
- ・ 自社のデータモデルはパッケージ等の業務システムのモデルに影響を受けてはならない（両者の違いはEDH内のコンバータで吸収する）
- ・ EDHはデータの変換機能は有するが、業務アプリケーションロジックは載せてはならない（EDHはあくまでデータEXCHANGE）
- ・ EDHは、それ自体を説明するシステム（リポジトリシステム）を有し、企業の基本的データの可視化に徹する必要がある

# ・ EDHを活用した緩やかなシステム移行

- ・ EDHを導入する際は、社員が関与できる最後の工程である“要件定義工程”で、メタデータ定義を実施することが前提となる
- ・ EDHは、大規模システムをビッグバンせずに、順次再構築する為の常設のブリッジの役割を果たす（ROIの早期実現とリスク回避）
- ・ レガシーダウンサイジングにおいては、オープン環境に取り出したデータHUB（マスタ、トランザクション）を活用して緩やかに移行する
- ・ SoR、SoEの両者はエンタープライズデータHUBを介して、独立した並行開発を可能にするとともに、相互にデータ連携される
- ・ 企業のM&A対応においては、両社システムのデータモデル、プロセスモデルの比較からはじめる（モデルファースト）

# ・アーキテクトに求められる振舞い

- ・プロジェクトよりプロダクトの成功を、システムを“作る”から“創る”ことを重視する。⇒自ずと開発工程にアジャイル手法を取り入れざるを得ない
- ・アーキテクトは、多かれ少なかれシステムの出来映えと納期・コストとのはざまに葛藤があるが、1つずつ辛抱強く乗り越えて行かねば、ならない
- ・プロマネの悪魔のささやきに屈することなく、最後まで、当初考えたアーキテクチャを諦めてはならない（どんなに角が丸くなくても譲れないイノベーションのネタを残す「肉を切らせて骨を断つ」）
- ・不幸にもプロジェクトでやり残したアーキテクチャは二次開発もしくは別のプロジェクトにてリベンジを図ること



**ご清聴ありがとうございました。  
(詳しくはぜひ本書をご覧ください!)**