

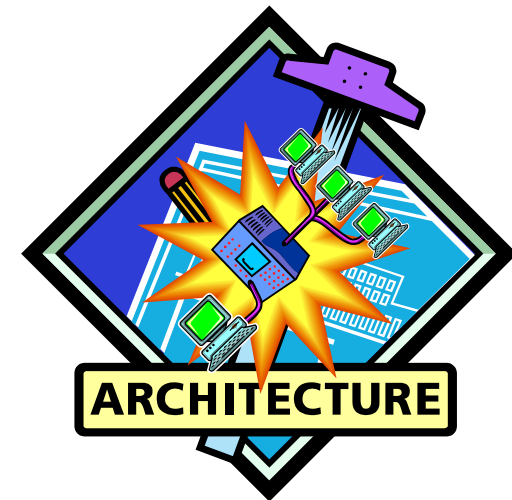


Modeling Service Oriented Architecture

Brian Itow

EVP, AZORA Technologies
brian.itow@azoratech.com

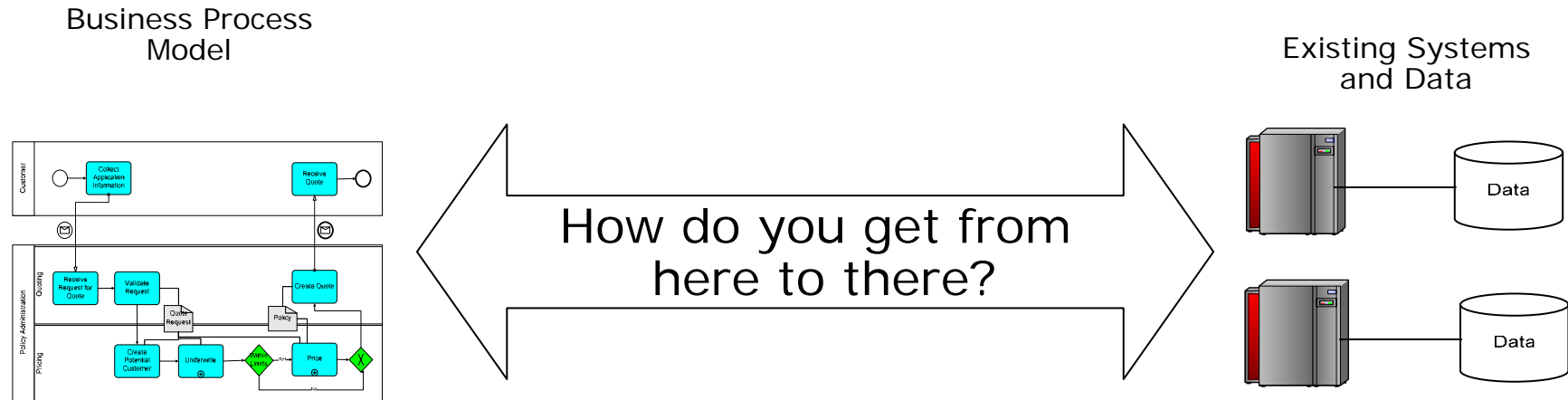
May 19, 2006



Agenda

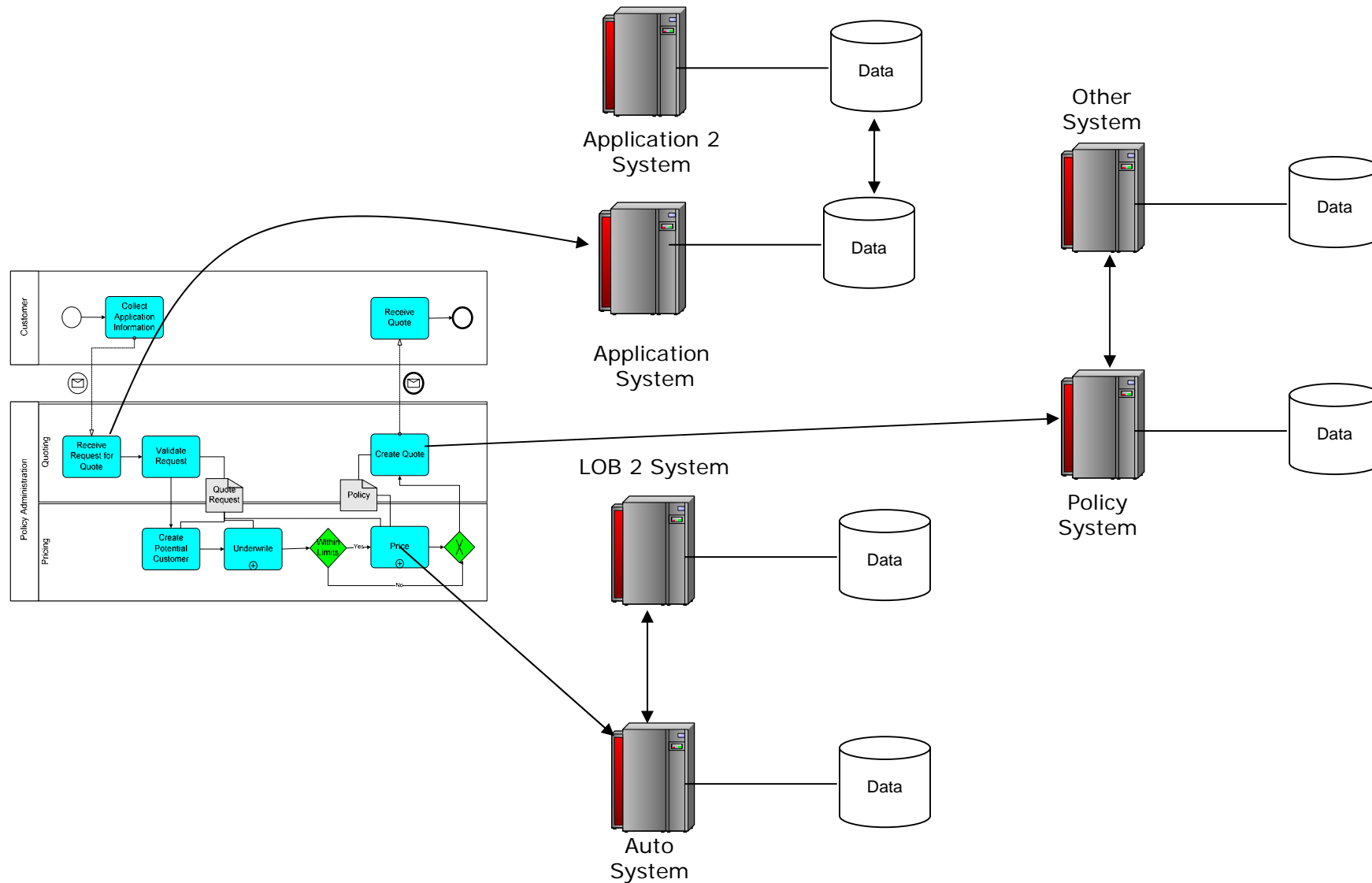
- Typical Integration and Business Process Solution
- Why has this been a problem?
- The service solution
- But SOA is Hard!
- What SOA must address to be successful
- How to achieve SOA

Insurance Example

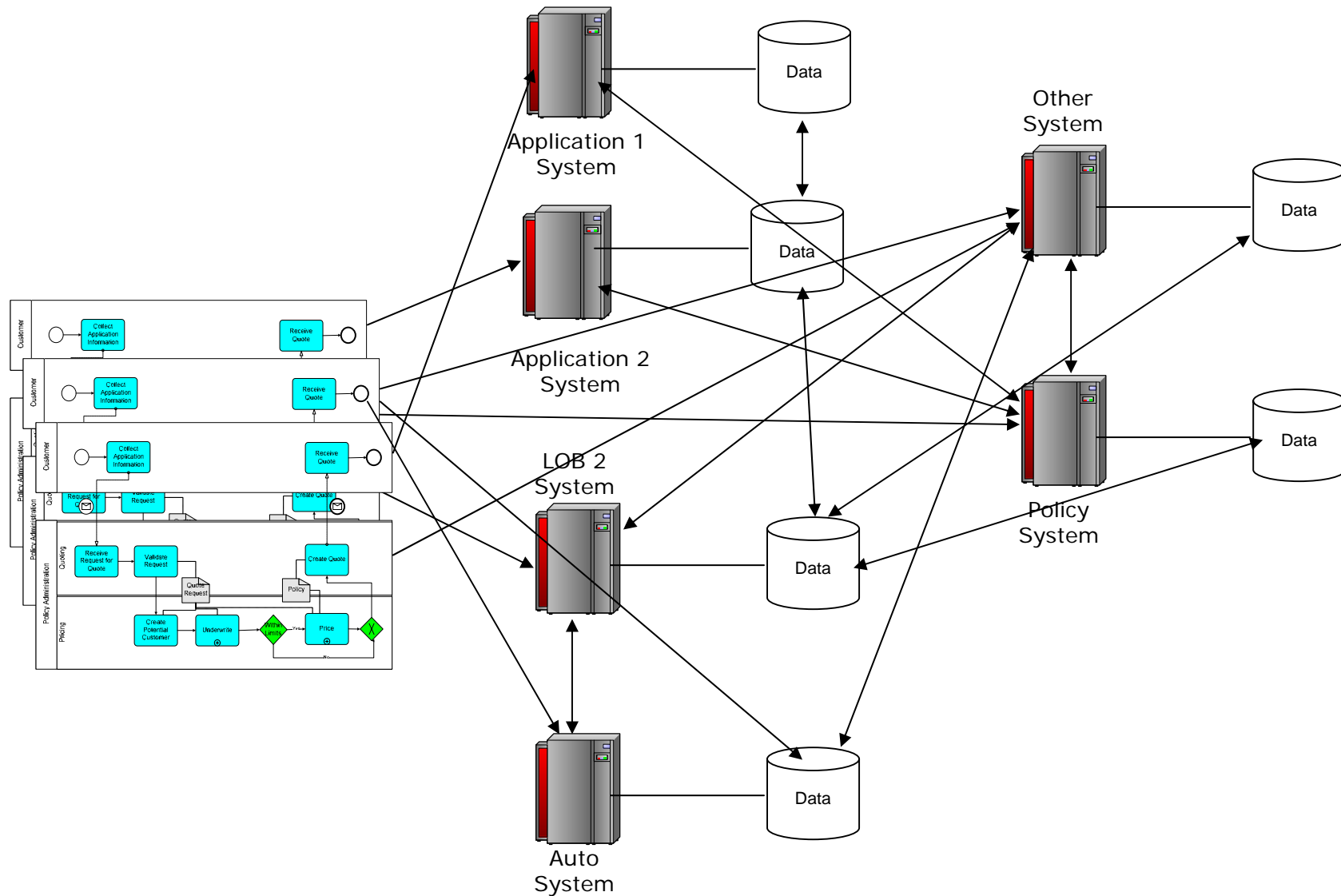


- Merger of two insurance companies with different lines of business
- Need new pricing mechanism that go across lines of business
 - 5% discount if 2 policies
 - 10% discount if 3 policies
- Pricing implemented on existing mainframe applications

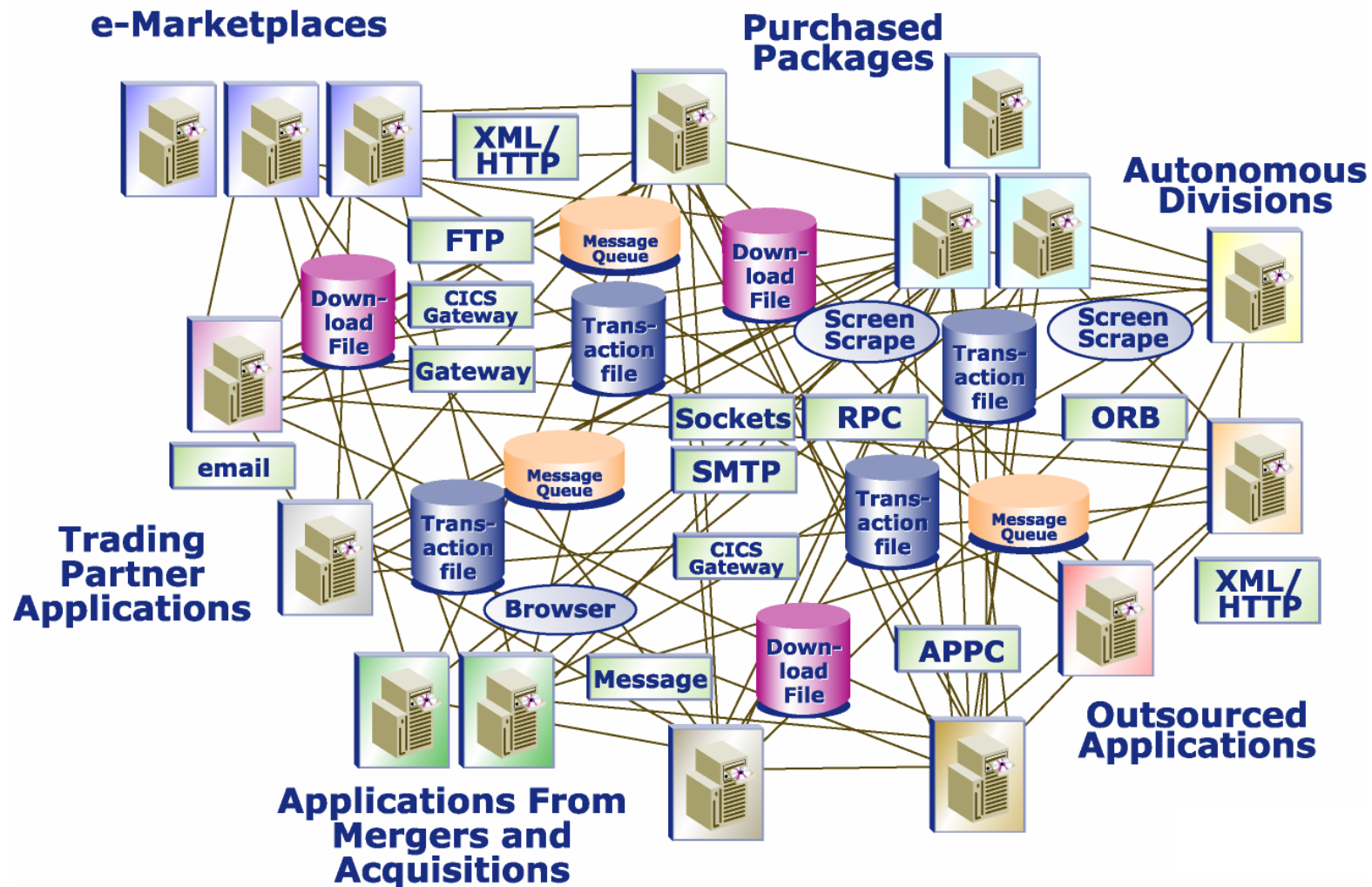
A Typical BPM Integration Scenario



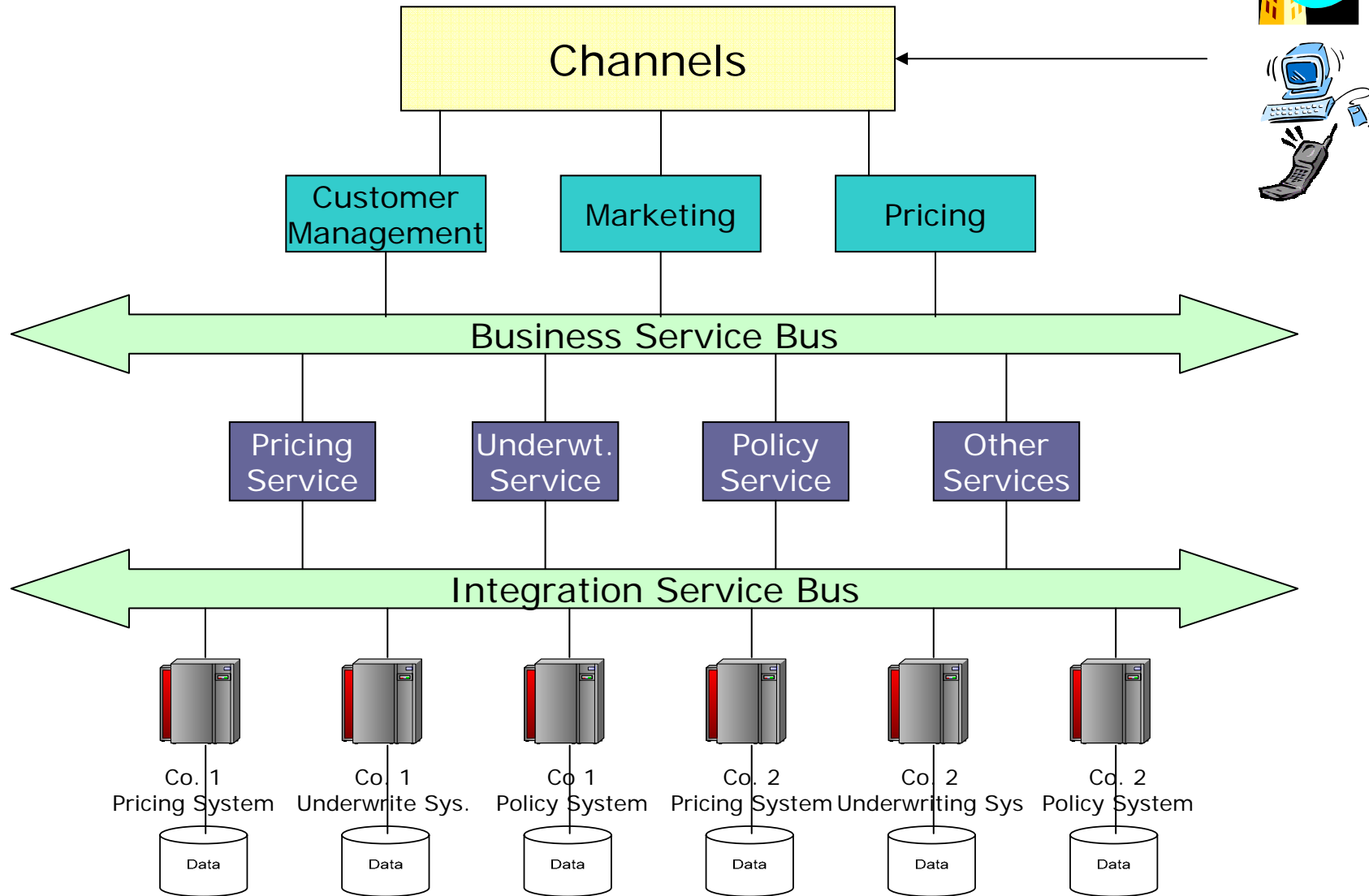
This Doesn't Scale



The Result: Enterprise Application “Spaghetti”



SOA: A Better Solution



Benefits of SOA Integration Approach

- Integrate once, connect many
 - Each system is integrated once into the service bus, rather than many time for each point-to-point connection
 - Less cost, consistent access
- Build up higher level business services
 - Combine lower level operations into business services that align with the goals and strategy of the new enterprise, rather than of the old systems
 - Quickly construct high-level, high-value business processes from the business services in response to new initiatives, competitive pressures, regulatory changes, ...
- Flexibility
 - Multiple services can be easily constructed from the integration of existing applications
 - New processes can be constructed from the service

Benefits of SOI Approach (2)

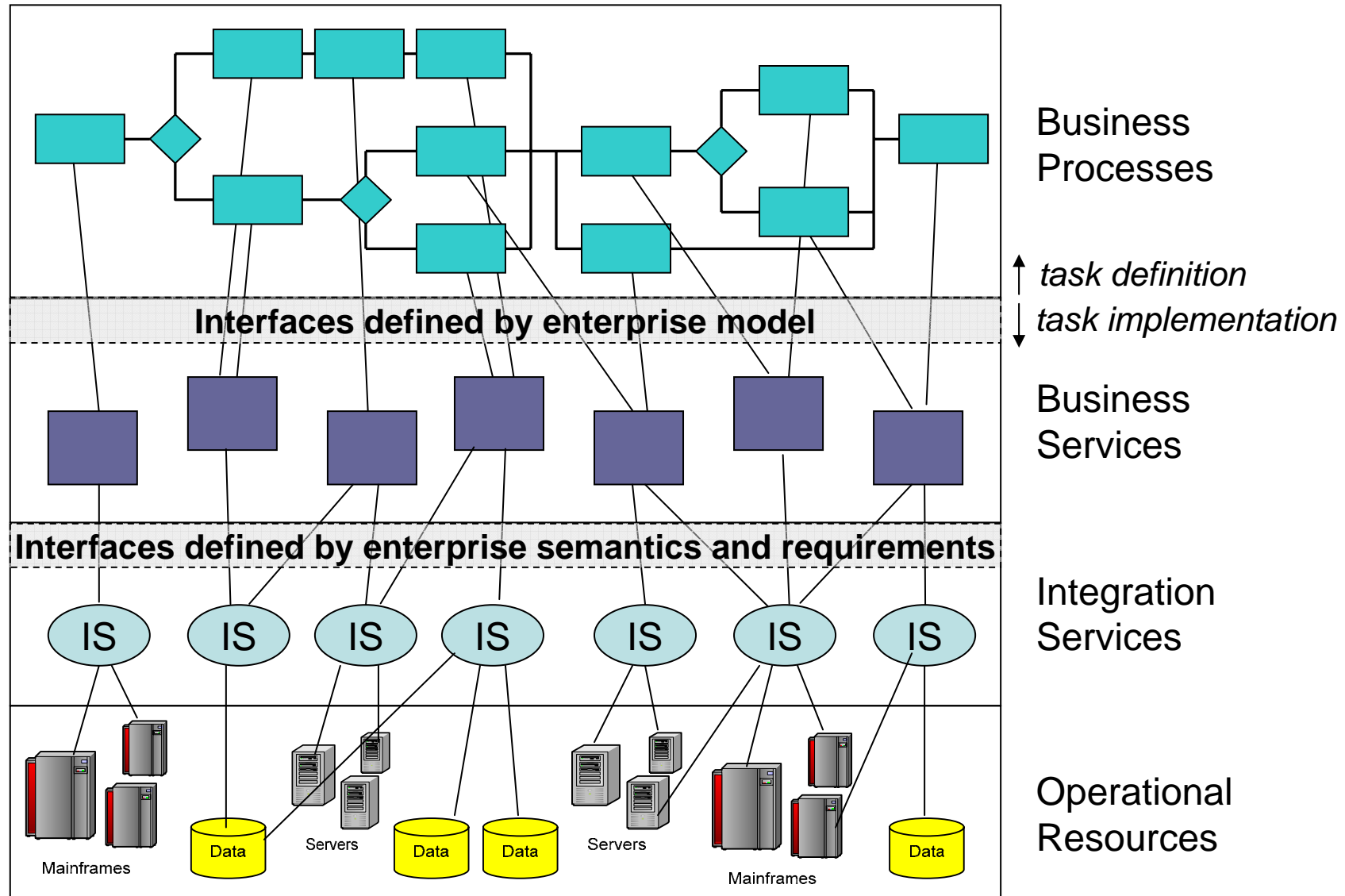
■ Adaptability to change

- Business Processes change quickly
- Operational Systems are difficult, costly and slow to change
 - Layered SOI approach enables quickly reconfiguring processes or services without needing to change operational systems
- Operational systems are retired or replaced
 - Layered SOI approach allows operational systems to change without affecting business processes

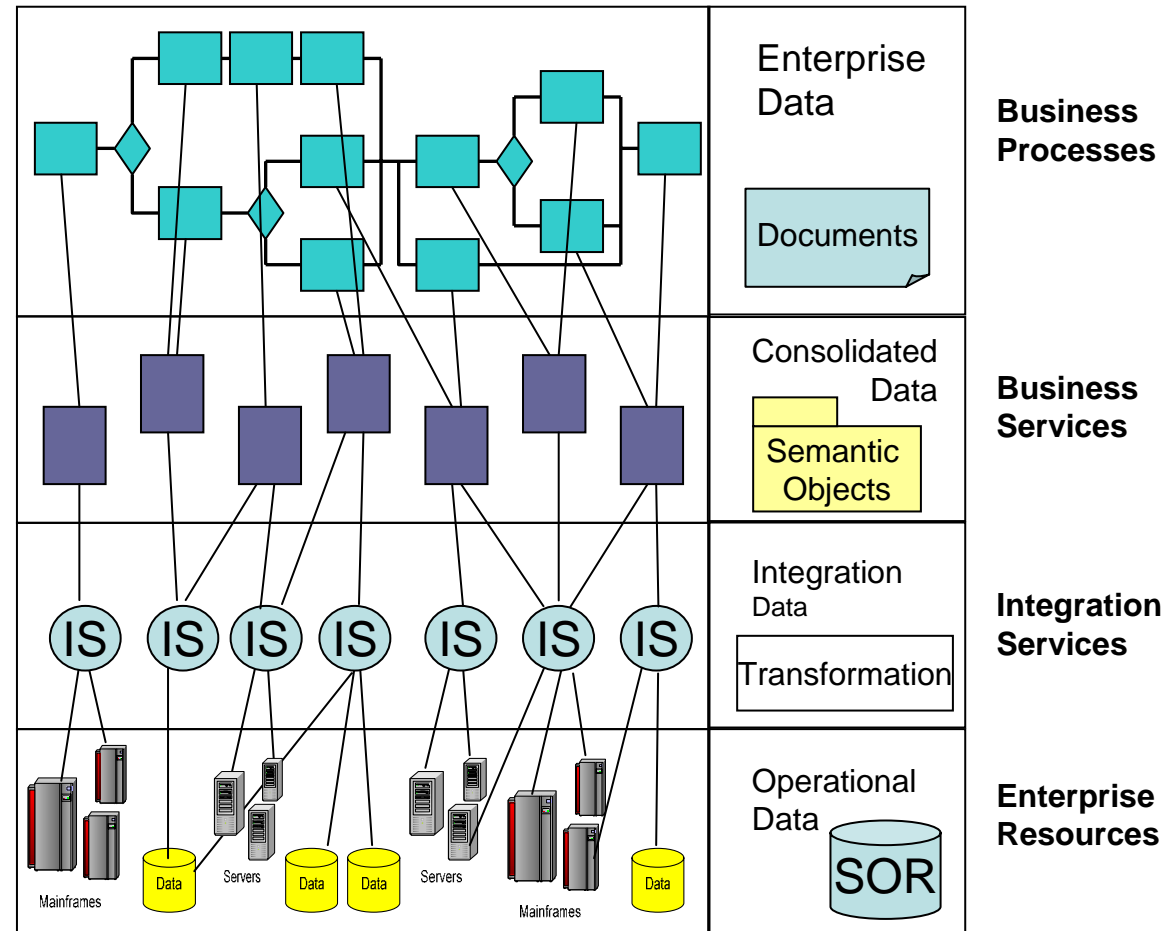
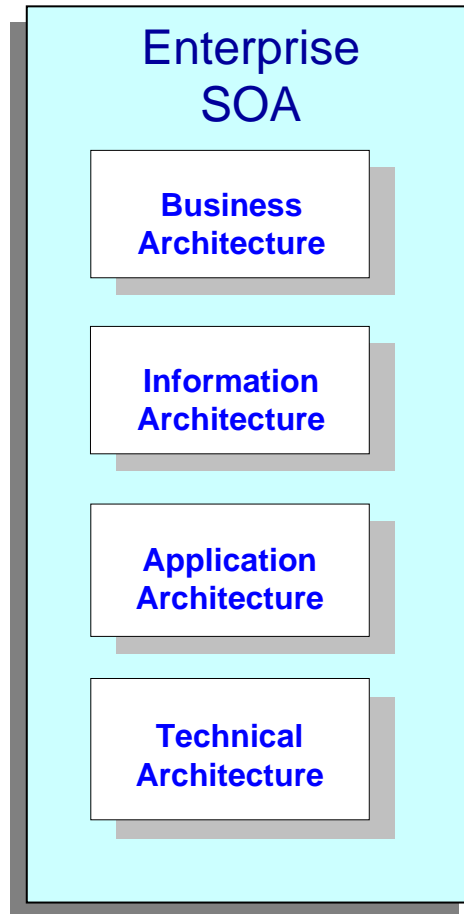
■ Incremental Approach

- Start small
- Add new integration services, business services and processes over time, as part of specific projects, in response to specific business needs
- Flexibility and capabilities increase exponentially with each new service.

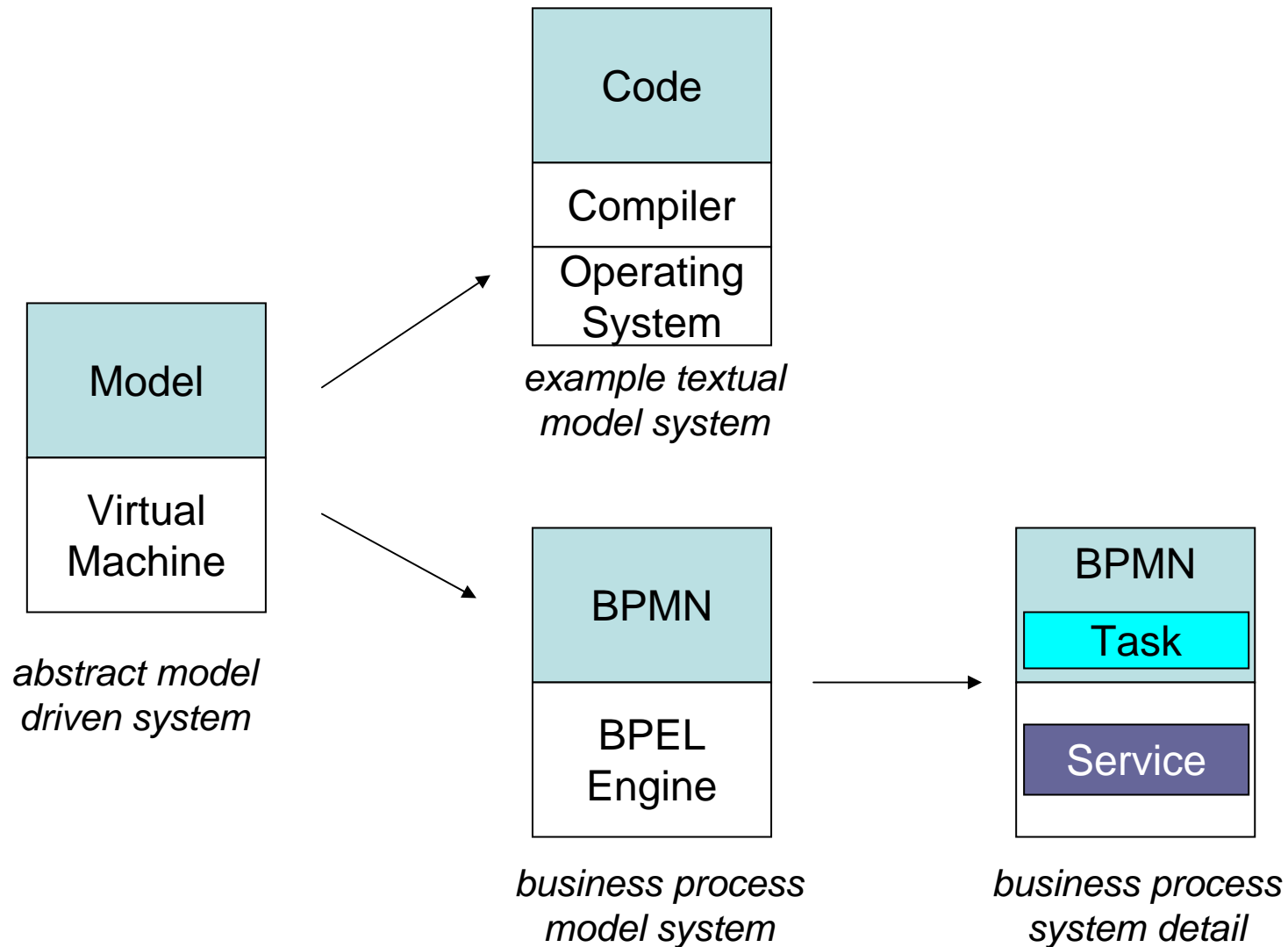
Layered Enterprise SOI Architecture



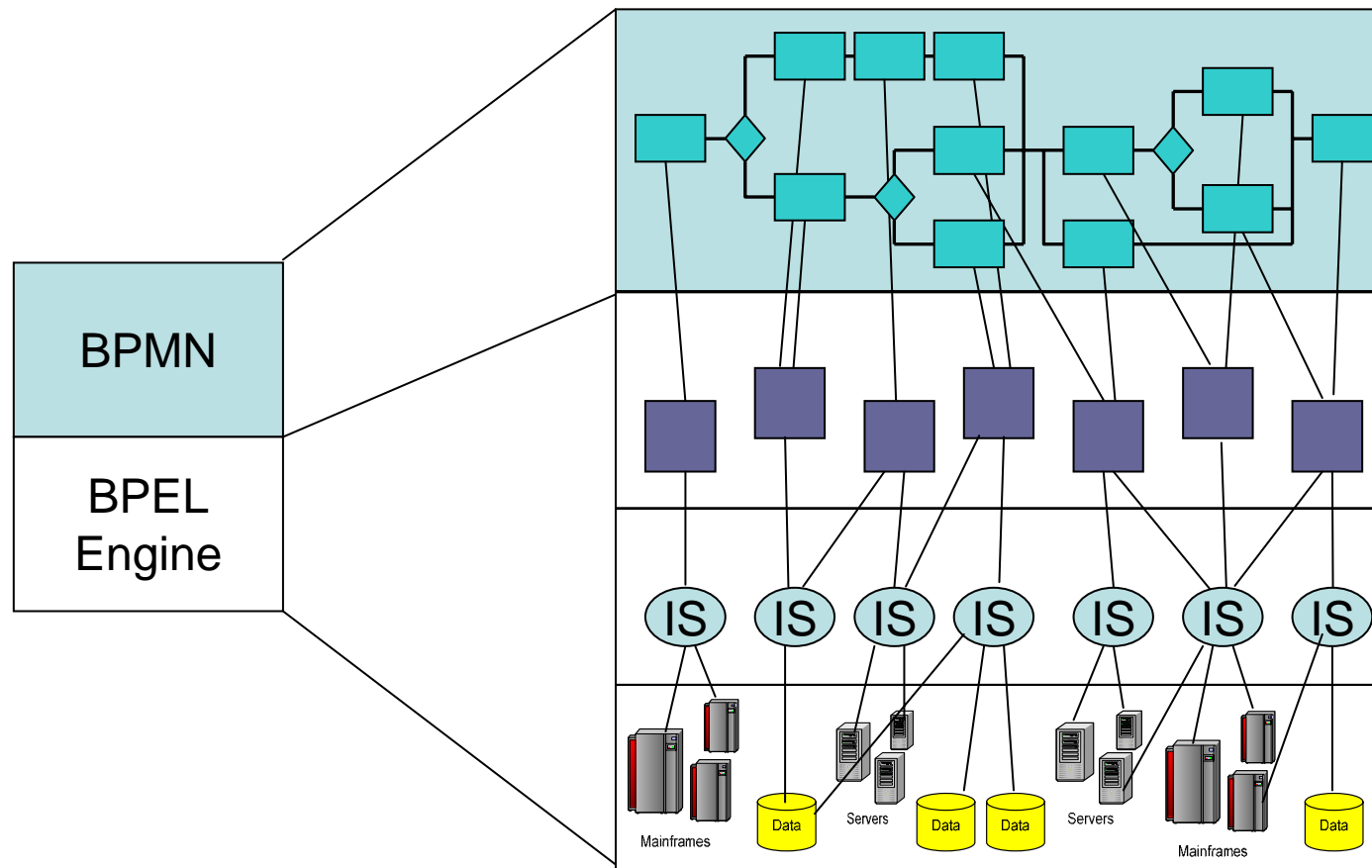
BPM and Enterprise Architecture



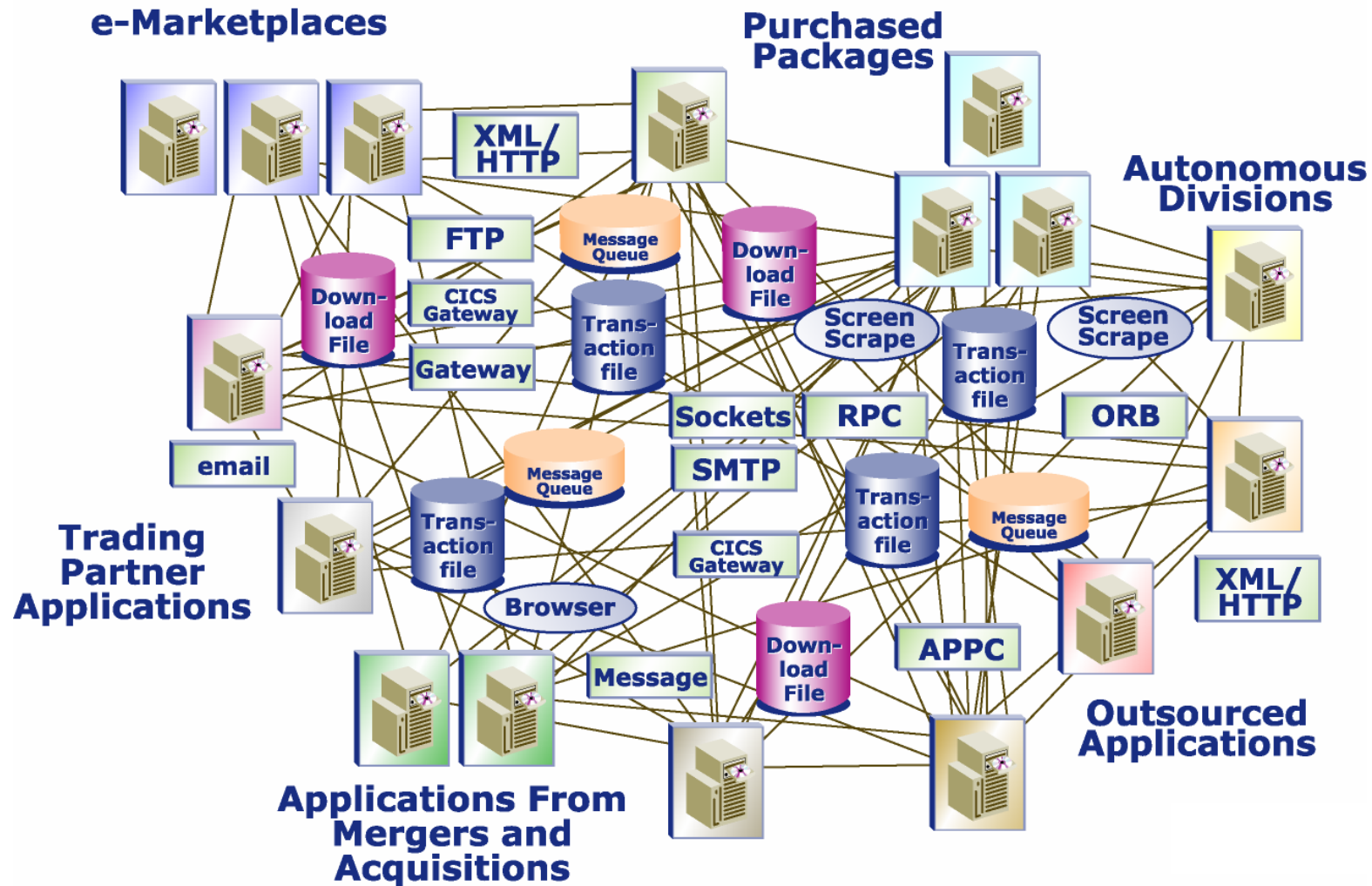
Model Driven Systems



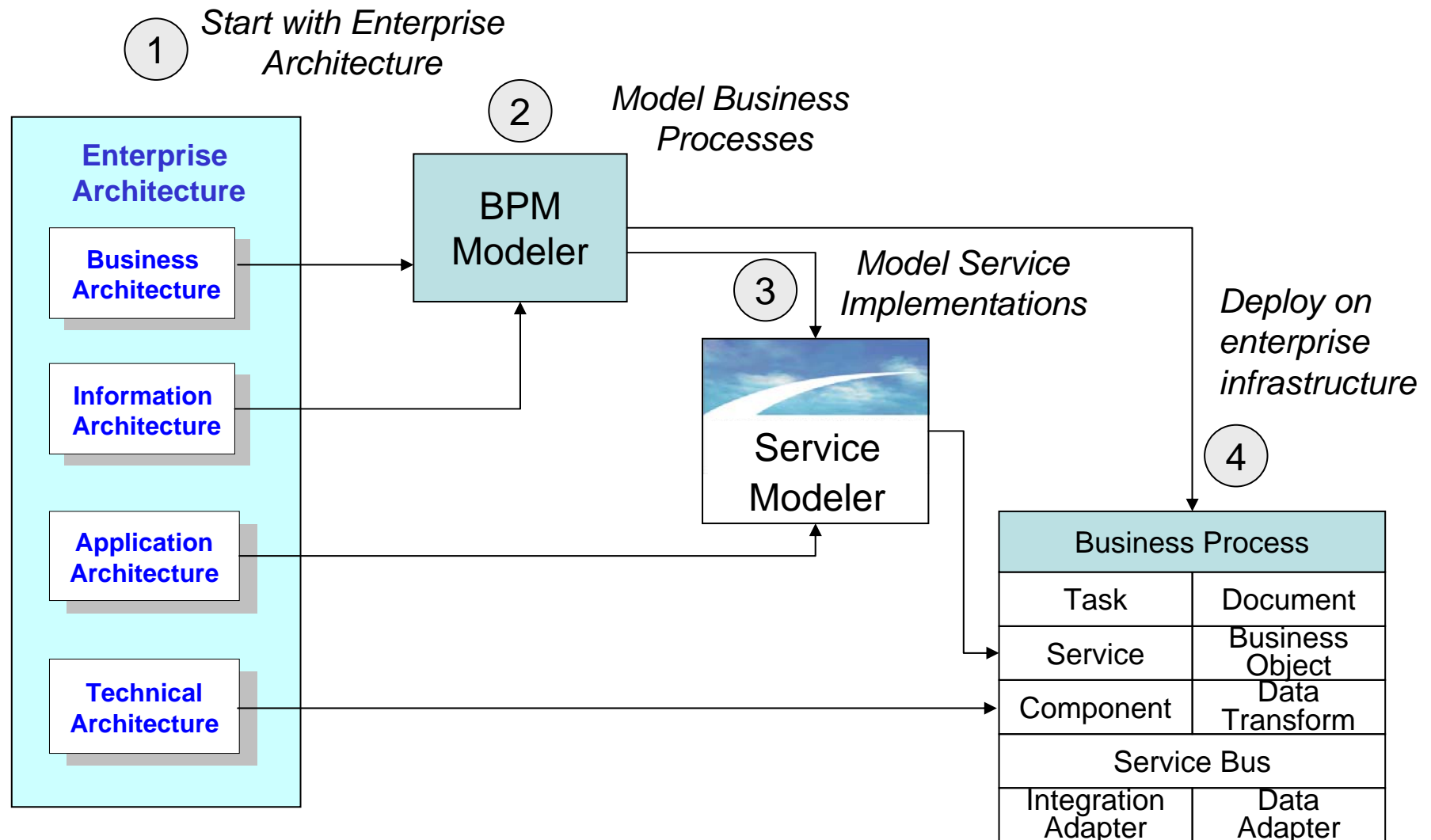
Business Process Run Time Platform



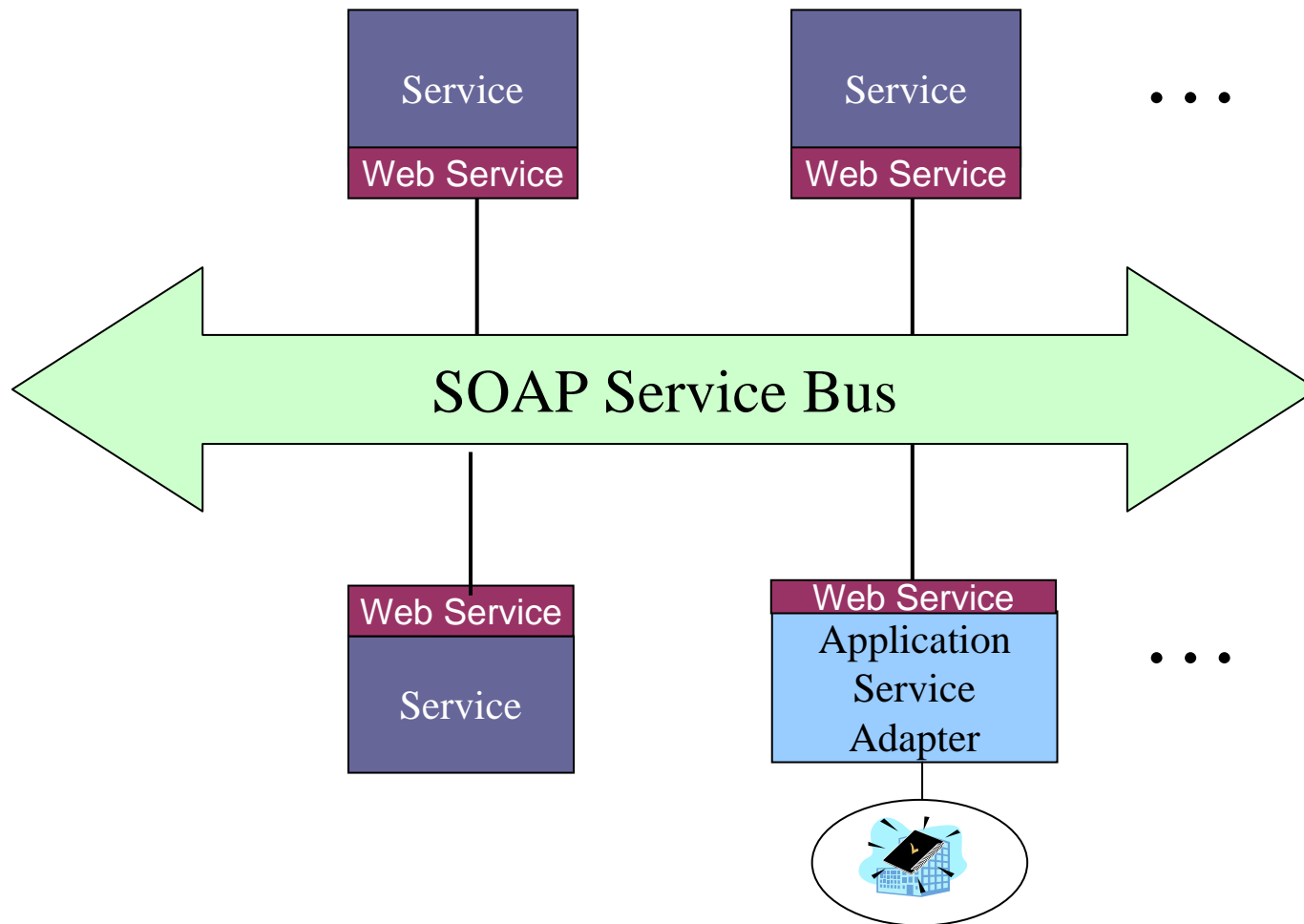
Oversimplified BPM Results in Spaghetti



Architecture Driven BPM Development

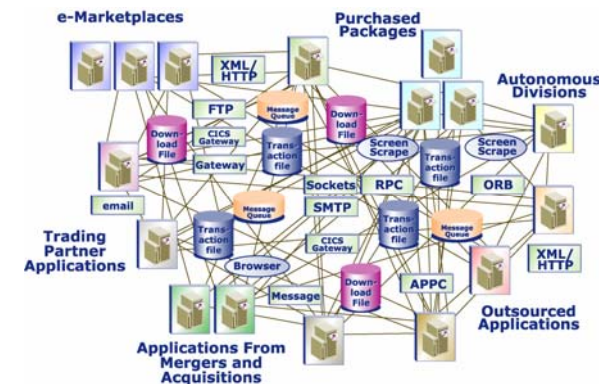


Web Services to the Rescue?

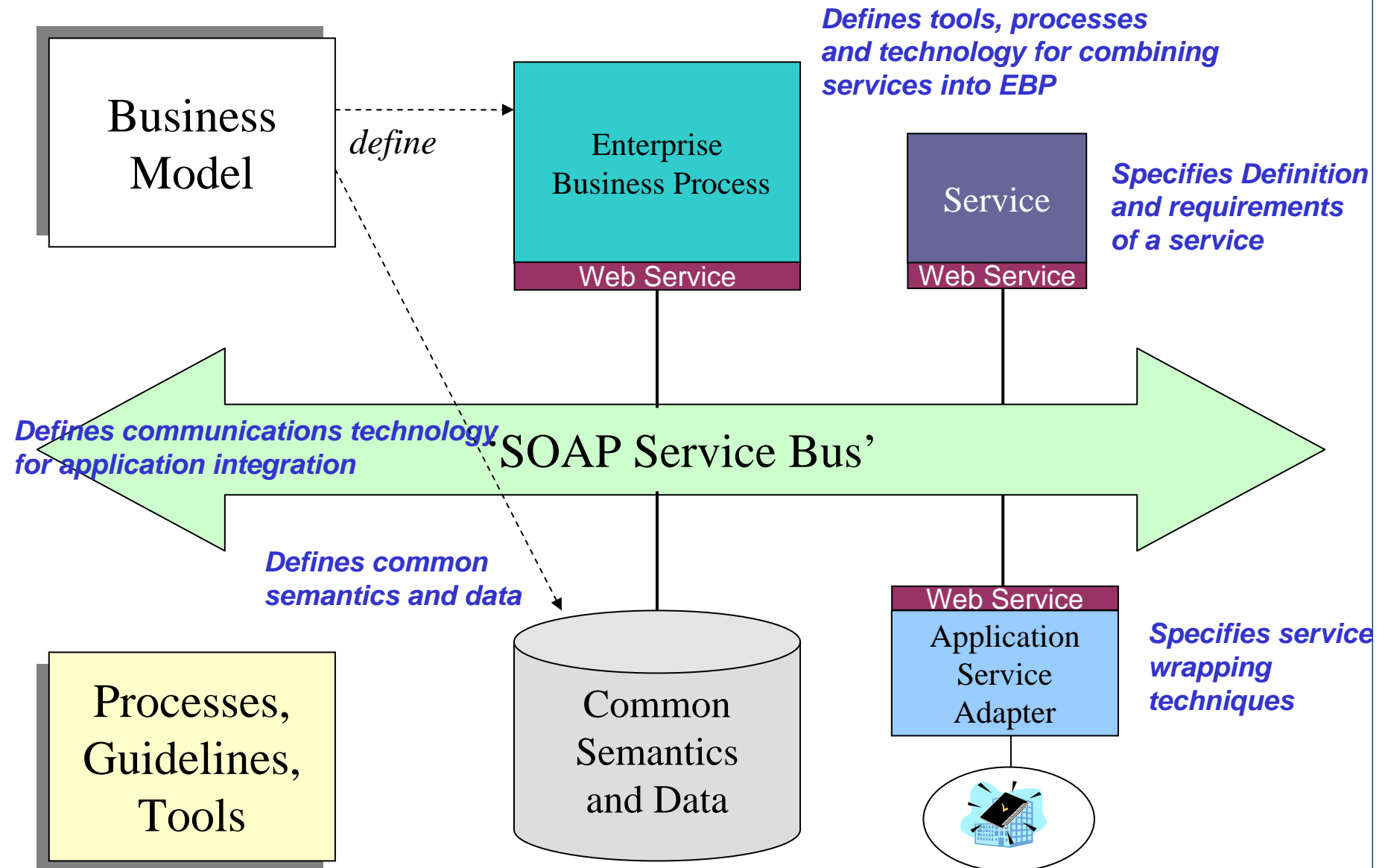


SOA is Hard!

- Previous technical infrastructures were very difficult to master
- We did not adequately understand the characteristics of services and service design
- Requires an understanding of the business and information and a strategic vision
- Requires an architectural based approach
 - But architecture is hard too!
- Requires an appropriate methodology
- Requires a supporting organizational structure



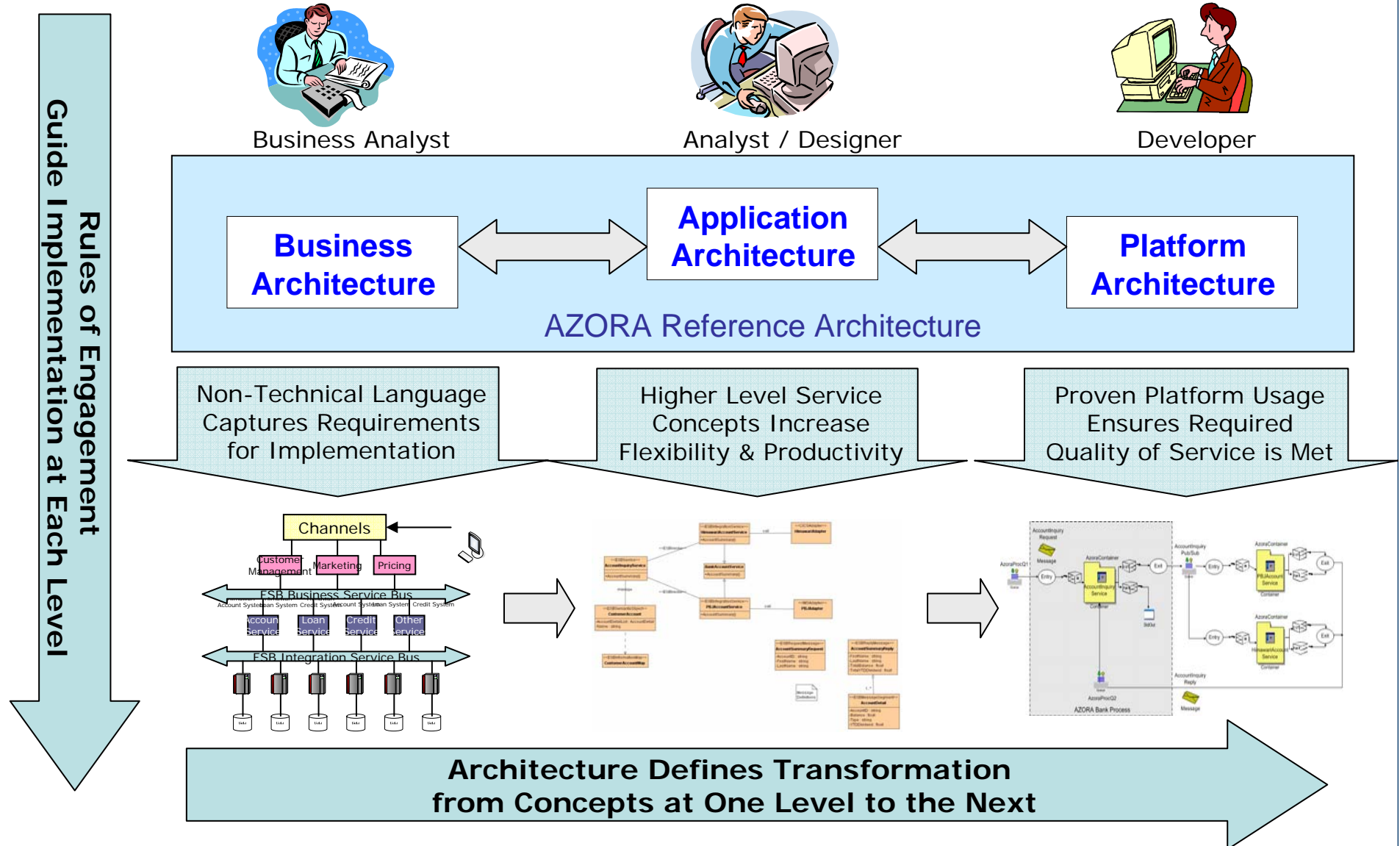
SOA Requires More



SOA Defined Again

- SOA is concerned with the *independent* construction of services which can be *combined* into meaningful, higher level business processes within the *context of the enterprise*.
- A Service Oriented Architecture describes several aspects of services within an enterprise:
 - The granularity and types of services
 - How services are constructed
 - How the services communicate at a technical level
 - How the services are combined together (i.e. orchestrated)
 - How the services interoperate at a semantic level (i.e. how they share common meanings)
 - How services contribute to IT and Business Strategy

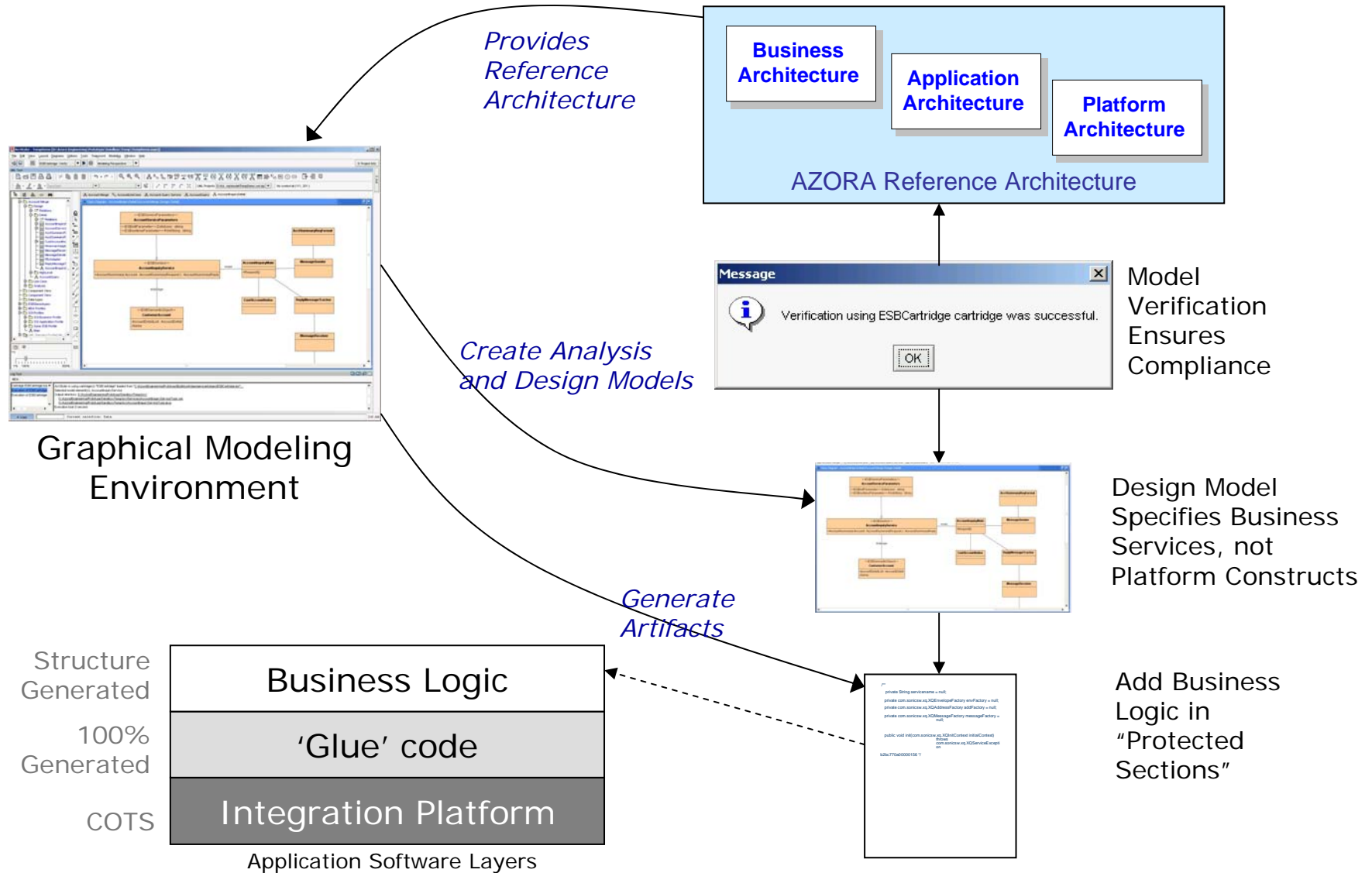
Proven Reference Architecture



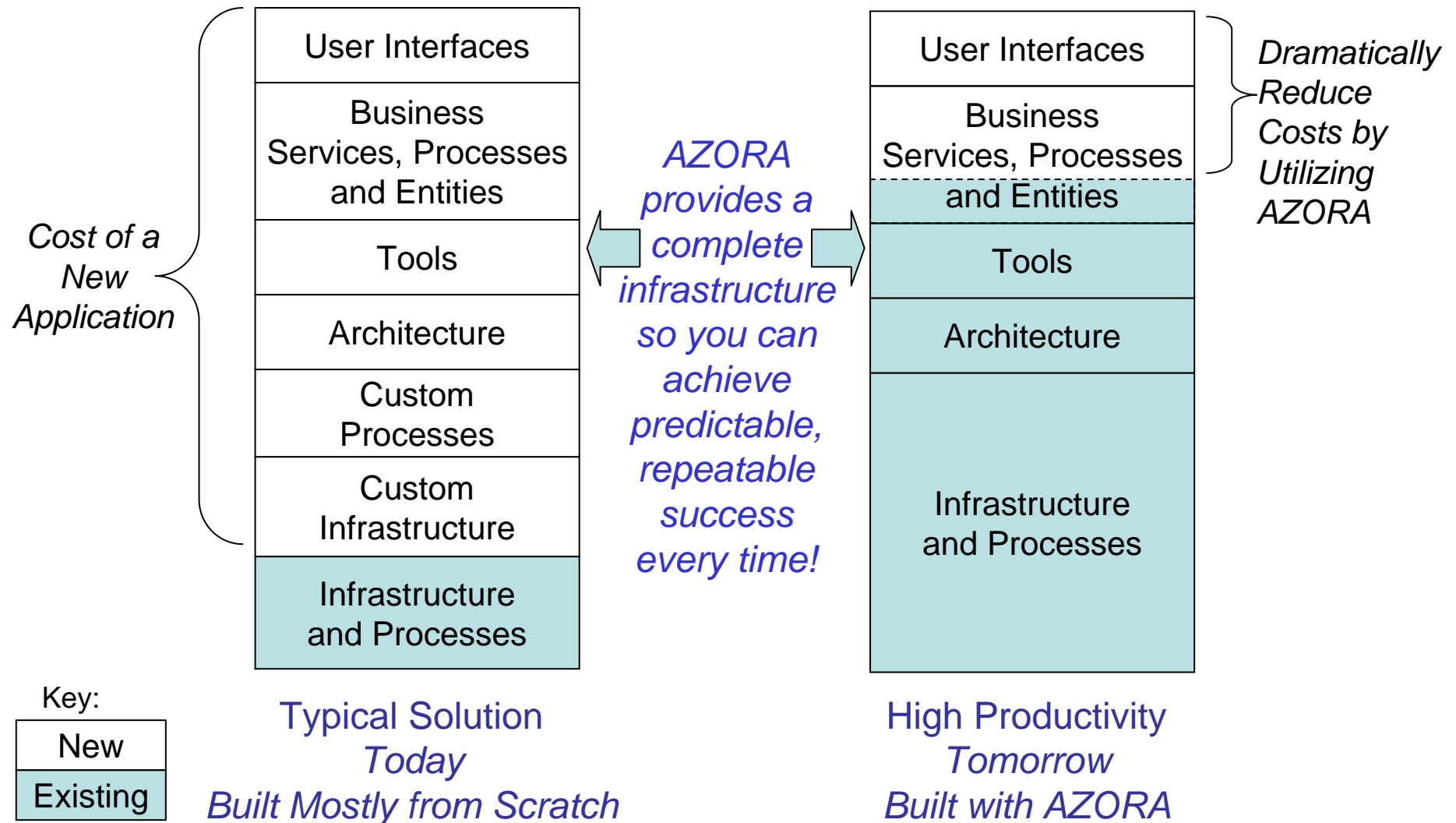
Reference Architecture Reduces Risk

- Most companies have a critical shortage of architects
- But, architecture determines project success:
 - Flexibility of the Application
 - Ability of Applications to Integrate
 - Consistency of Implementation, Operations and Maintenance
 - Correct Use of Platform Capabilities
 - Performance, Security, Reliability, Availability
- Worse, most of these key success factors cannot be tested until after the application is fully deployed!
- AZORA's Reference Architecture provides a proven solution to these requirements:
 - Eliminate the Need to Re-Invent Architecture for Each Project
 - Reduce the Need for Critical Resources
 - After Deployment, You See Success, Not Surprises

Automation and Artifact Generation



Optimization and Predictability



How to model and implement SOA successfully

- Generation of Artifacts that Solve Customer Problems
 - Create Application Design Model
 - Validate the Correctness of the Model
 - Generate the Necessary Platform Artifacts to Support the Design
 - Understand how the Reference Architecture Supports the Design and Artifact Generation
- Repeatable Implementation
 - Step-by-Step Instructions for Creating Analysis and Design Models
 - Overall Support for Development Process and Architecture
 - Guidance and Tracking of Complete Lifecycle
- Platform Support
 - Platforms Require Difficult, Platform-Specific Artifacts and Configurations
 - Generate the Artifacts and Configurations Automatically, so your Developers Do Not Have to be Platform Experts

Benefits of Achieving This

- Reduced Implementation Cost and Risk
- Improved Implementation Quality
- Increased Solution Predictability and Repeatability
- Provides Framework for Easy Implementation of SOA Solutions
- Enables Accurate Translation of Customer Requirements into a Technical Solution
- Directly Reduces Amount of Work and Re-Work Required; Ensures Predictability through Consistent Development Approach; Enhances Solution Re-Use



Questions



日本語の資料をご要望の方は、下記までお問い合わせください。

Mail: info@azoratech.com

TEL:03-3406-0380

Benefits of SOA

- Applications can be exposed to diverse clients and channels
 - Because services can be accessed via standard Web services
- Applications and business processes can be more easily constructed and modified
 - Because they're built from autonomous, composable services with loosely coupled interfaces. BPM provides for high level of application abstraction and productivity.
- Applications can be deployed or enhanced incrementally
 - Because services can be rolled out independently and implementations can be migrated transparently
- Application Integration can be aligned with business requirements
 - Because of mediation of service layer
- Reusable services provide improved time-to-market and quality, and reduced costs and maintenance

Insurance Quoting BPMN Model

