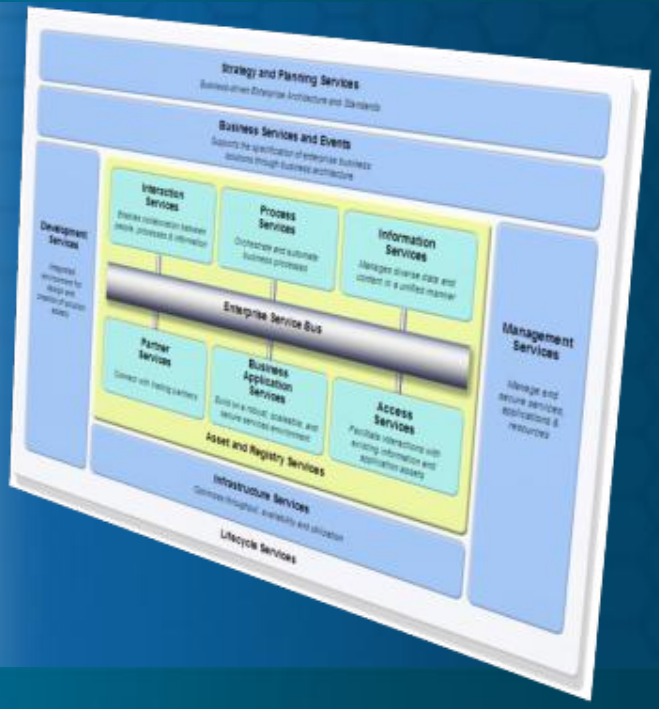
IBM SOA Architect Summit

Infrastructure Architecture:

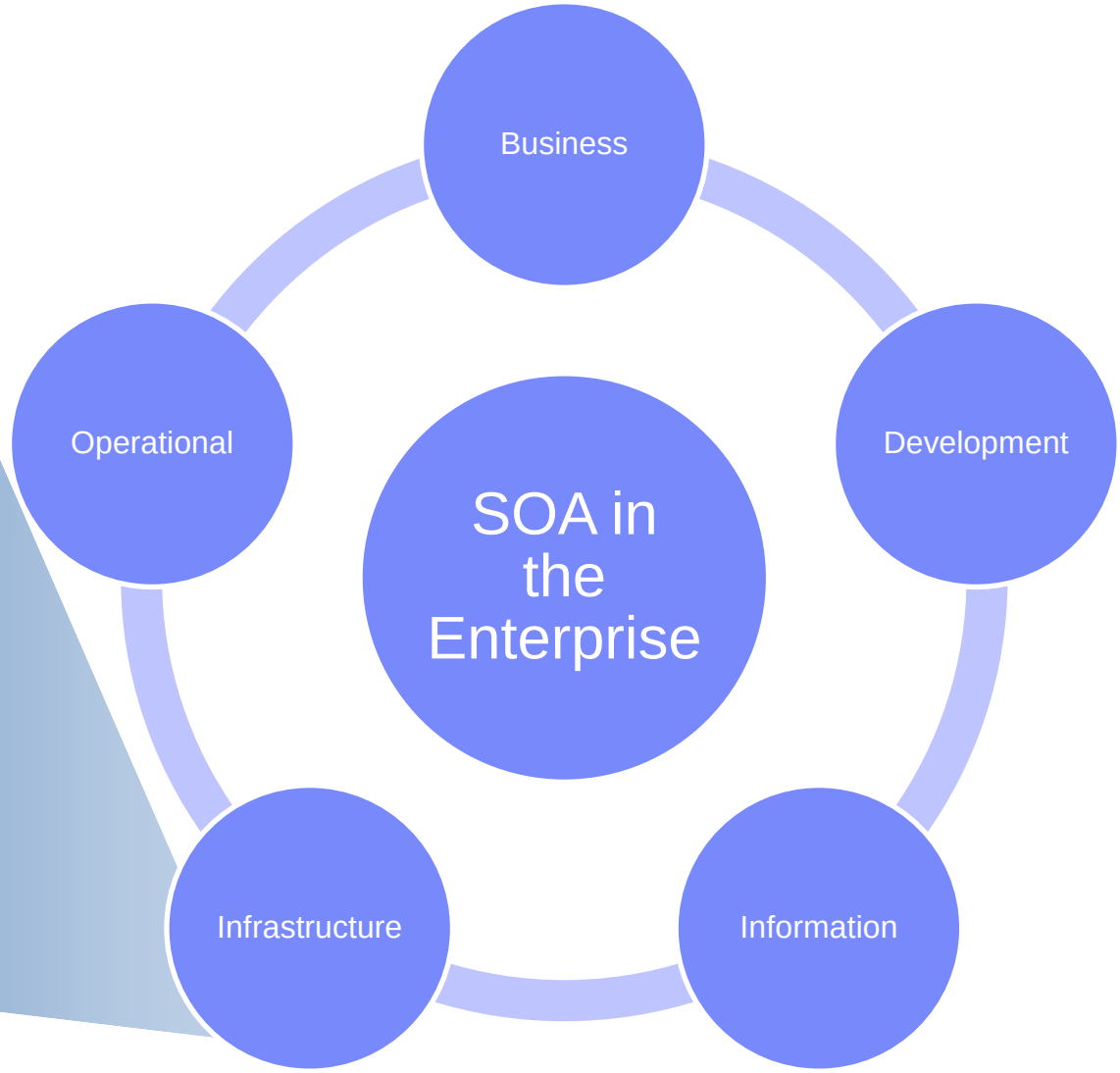
Architecting the Right SOA Infrastructure



Claus Jensen
STSM, SOA Foundation

Service-Oriented **Architecture** in the Enterprise

- **How do you architect the right SOA Infrastructure?**
- **What materials and best practices can be referenced?**
- **How do we get the team to follow and leverage these best practices?**
- **How do non-functional requirements impact infrastructure?**
- **How do we manage our SOA infrastructure?**

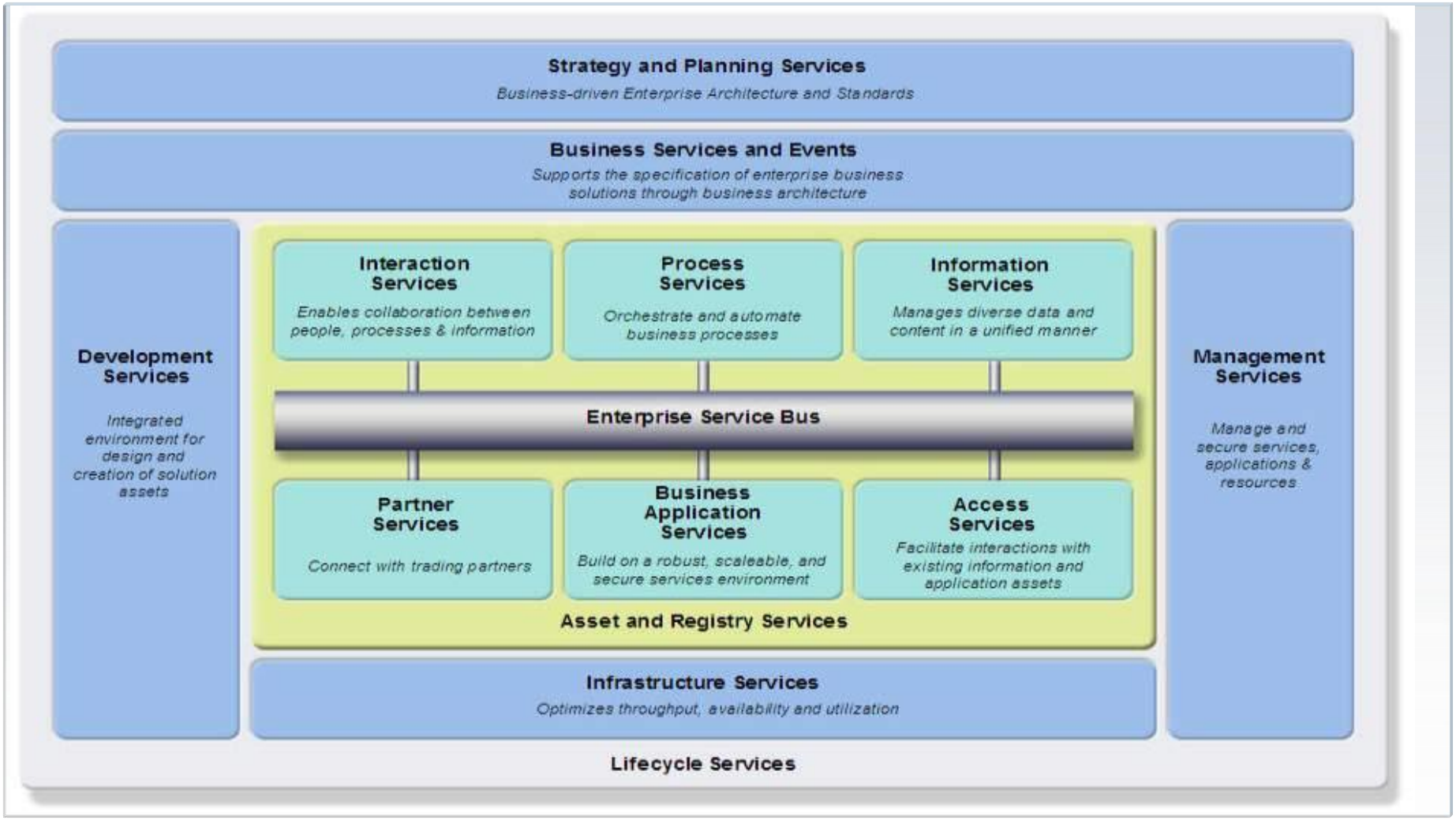


Agenda

- **Planning SOA Infrastructure**
- Infrastructure Considerations for SOA
  - Development Services
  - ESB/Connectivity
  - Non-Functional Requirements
    - Performance
    - Availability
    - Virtualization
    - Security
  - Service Management
- IBM Capabilities to Support SOA Infrastructure Architecture
- Summary

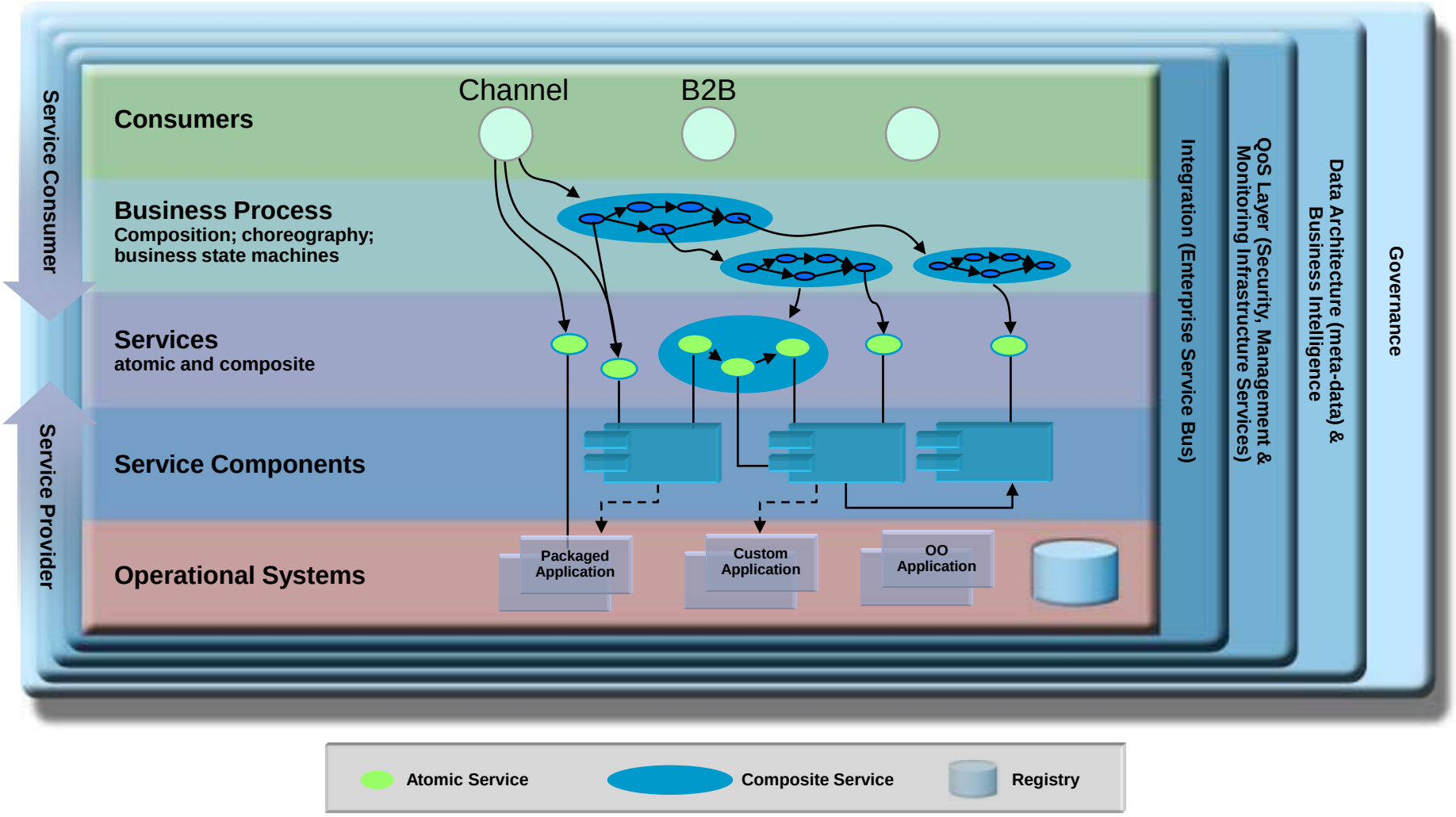
SOA Foundation Reference Model

Guidance in Planning Infrastructure



SOA Solution Layering

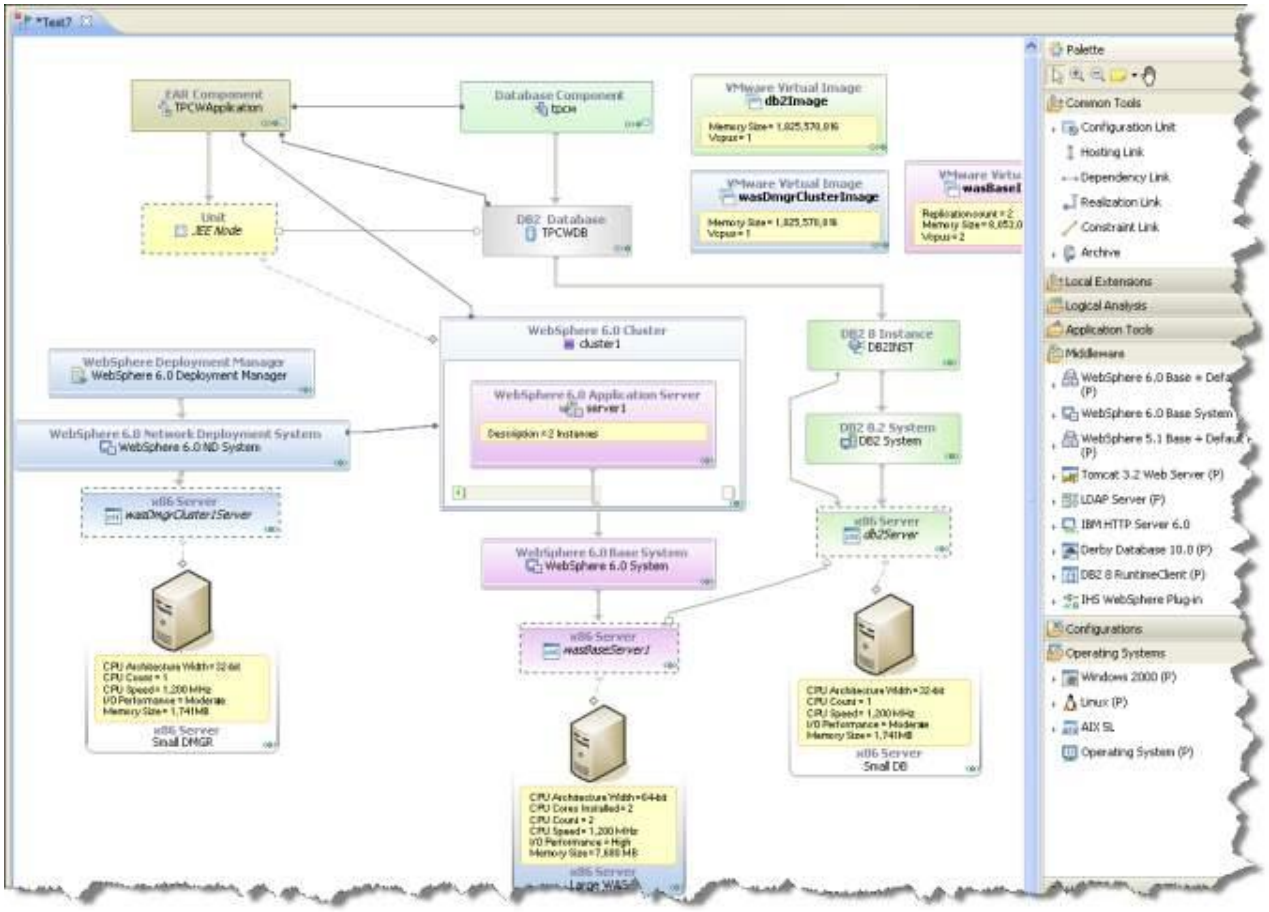
Leveraging the SOA Reference Architecture



Deployment Architecture in Rational Software Architect

Communicate, Guide, Validate

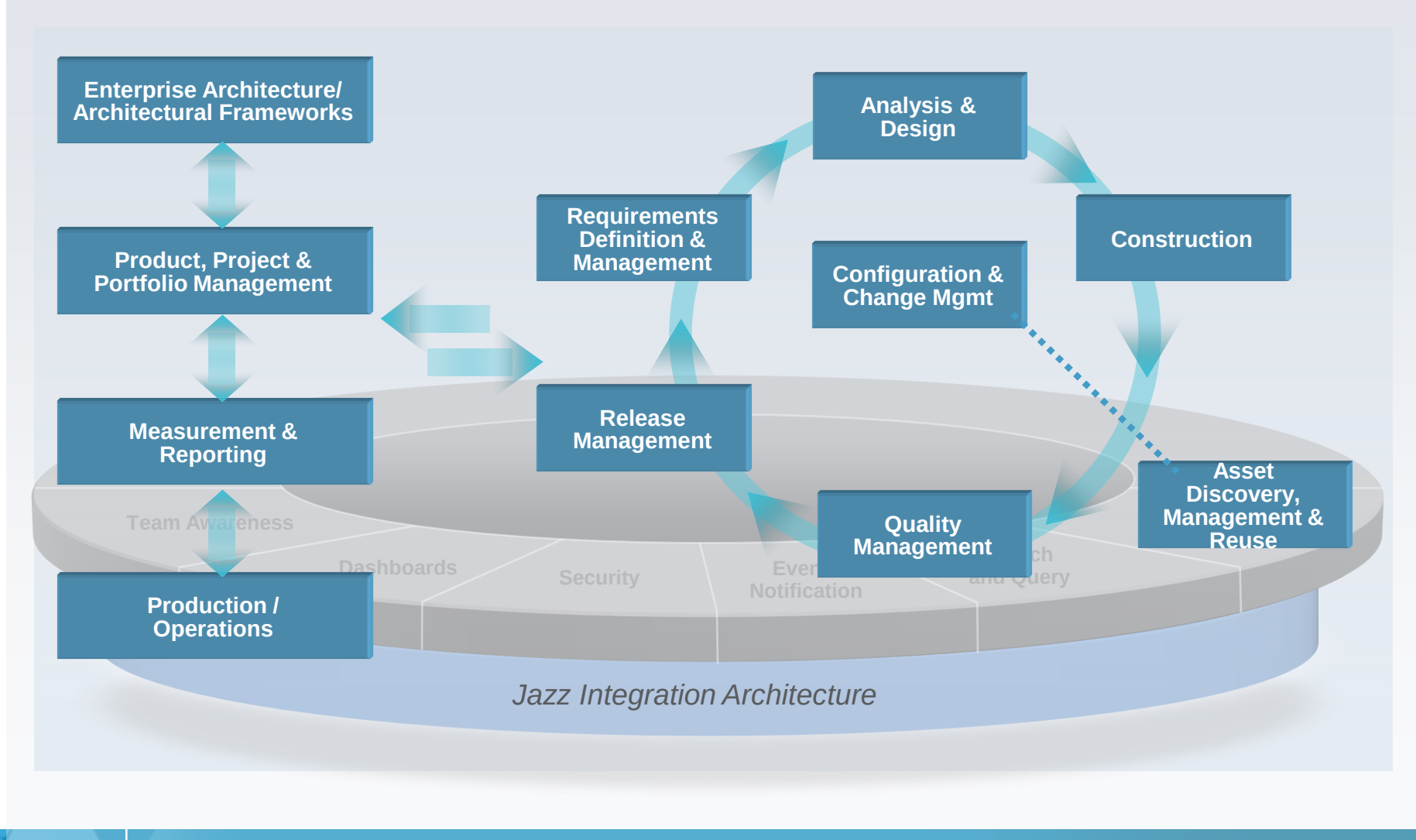
- Facilitates communication between development and operation organizations
- Promotes an open IT resource modeling format
- Creates reusable IT templates for capturing standards and best practices
- Ensures successful deployments by validating deployment designs that are mapped to your environment



Agenda

- Planning SOA Infrastructure
- **Infrastructure Considerations for SOA**
  - Development Services
  - ESB/Connectivity
  - Non-Functional Requirements
    - Performance
    - Availability
    - Virtualization
    - Security
  - Service Management
- IBM Capabilities to Support SOA Infrastructure Architecture
- Summary

Infrastructure for Delivering SOA Solutions



The Enterprise Service Bus ***Pattern***

Connectivity Infrastructure

An ESB is a flexible connectivity infrastructure that enables dynamic, adaptable service interaction.

The ESB *connects* requestor and provider, enabling the *decoupling of interactions* between them

The ESB fulfils *two core principles* in support of *separation of concerns*:

Service Virtualization

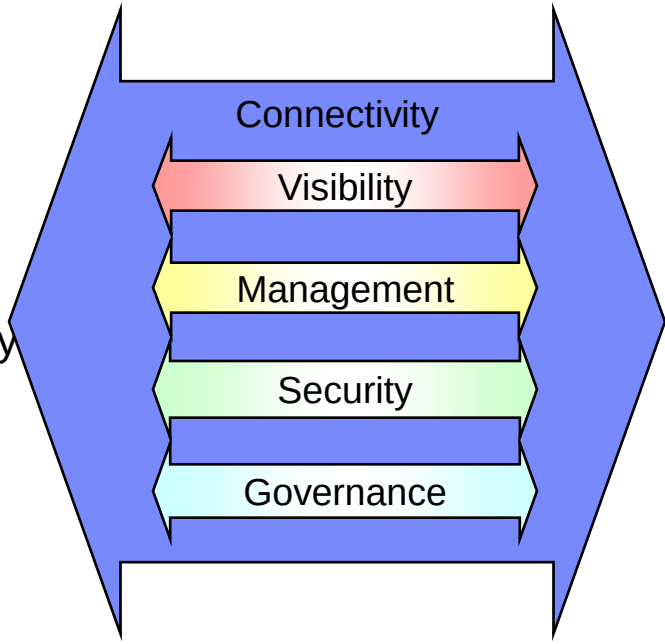
- *Protocol and Interaction Pattern* … via conversion
- *Interface* … via transformation
- *Identity* … via routing

Aspect Oriented Connectivity

- *Security*
- *Management*
- *Logging*
- *Auditing*

The Connectivity Infrastructure

- Service Visibility (& Interoperability)
  - Allows a service consumer to interact with a service provider
  - Service registry and service bus (service virtualization)
- Service Management (& Monitoring)
  - Enables understanding of and dynamic adaptation to the changing service conditions
  - Management products, often facilitated by service registry and service bus (aspect-oriented connectivity)
- Service Security
  - Guards integrity by securing access to services
  - Security products, sometimes facilitated by service registry and service bus (aspect-oriented connectivity)
- Service Governance
  - Manages the configuration of the other parts of the Connectivity Infrastructure, supporting the connectivity goals of the enterprise
  - Derives from cooperative parts of the other infrastructures



An Emerging Solution is a Federated ESB

Which Can Link Together Multiple, Disparate Domains

A Federated ESB is an enterprise-spanning connectivity infrastructure of multiple ESBs working together to extend service reuse across as well as within domains.

SOA and Non-Functional Requirements

SOA Characteristics

- Applications reused in new dynamic ways
- Services combined from multiple sources
- Rapid deployment
- Services route to any available resource
- Distributed access

Key Infrastructure and Management Considerations

SOA Introduces Performance Challenges

- Measuring performance across organizational boundaries can be more difficult than in siloed applications
- Response time estimation is more challenging in a more distributed environment
  - Performance costs can be difficult to predict
  - Performance testing an SOA application requires the use of new techniques
- Increased requirement for XML processing may impact performance

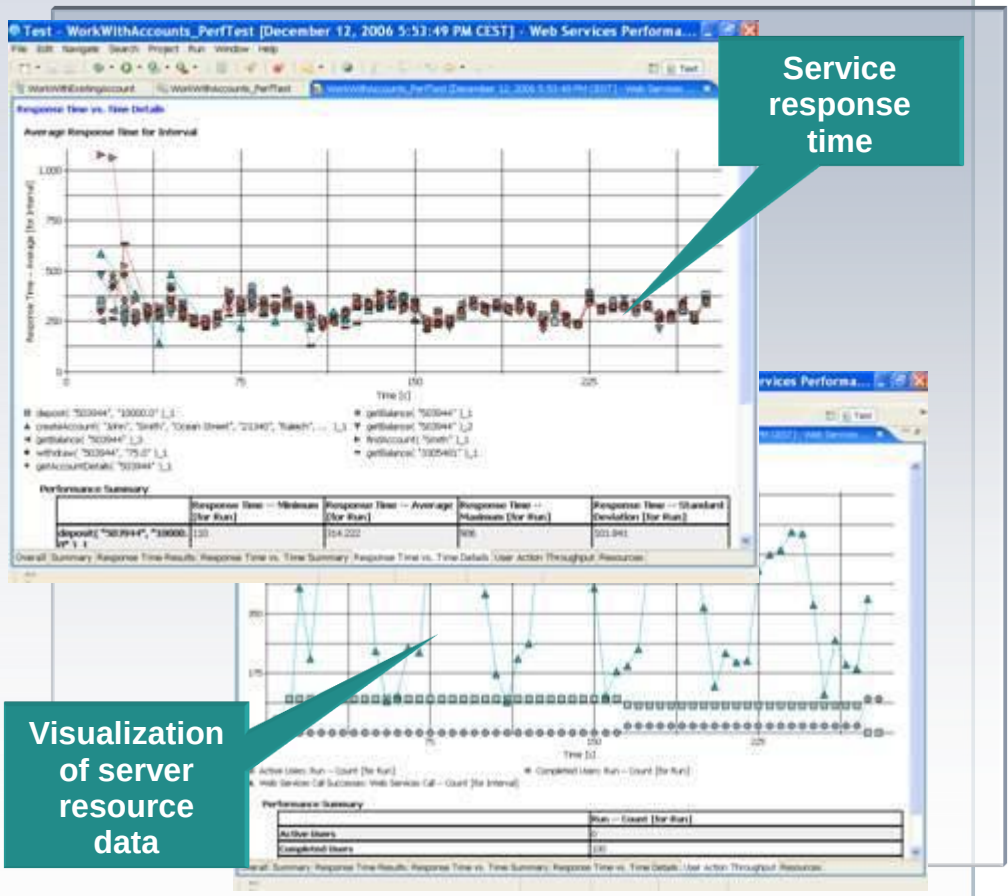
Performance Should Not be an Afterthought

It Should be Engineered into the Solution

- Performance in SOA systems should be a combination of performance engineering and performance management
- SOA-based applications can change the way an infrastructure performs
  - XML message transformation, location, message size, frequency
  - More complex applications and transactions
- Each of the components should be used to build a performance budget, transaction models and use cases
- Middleware and server sizing need to be done with the application teams
  - How many, how available, virtualized, system platform
- Don’t forget about security overhead
  - Authentication, Authorization, Encryption

SOA Performance Testing and Problem Analysis Tools

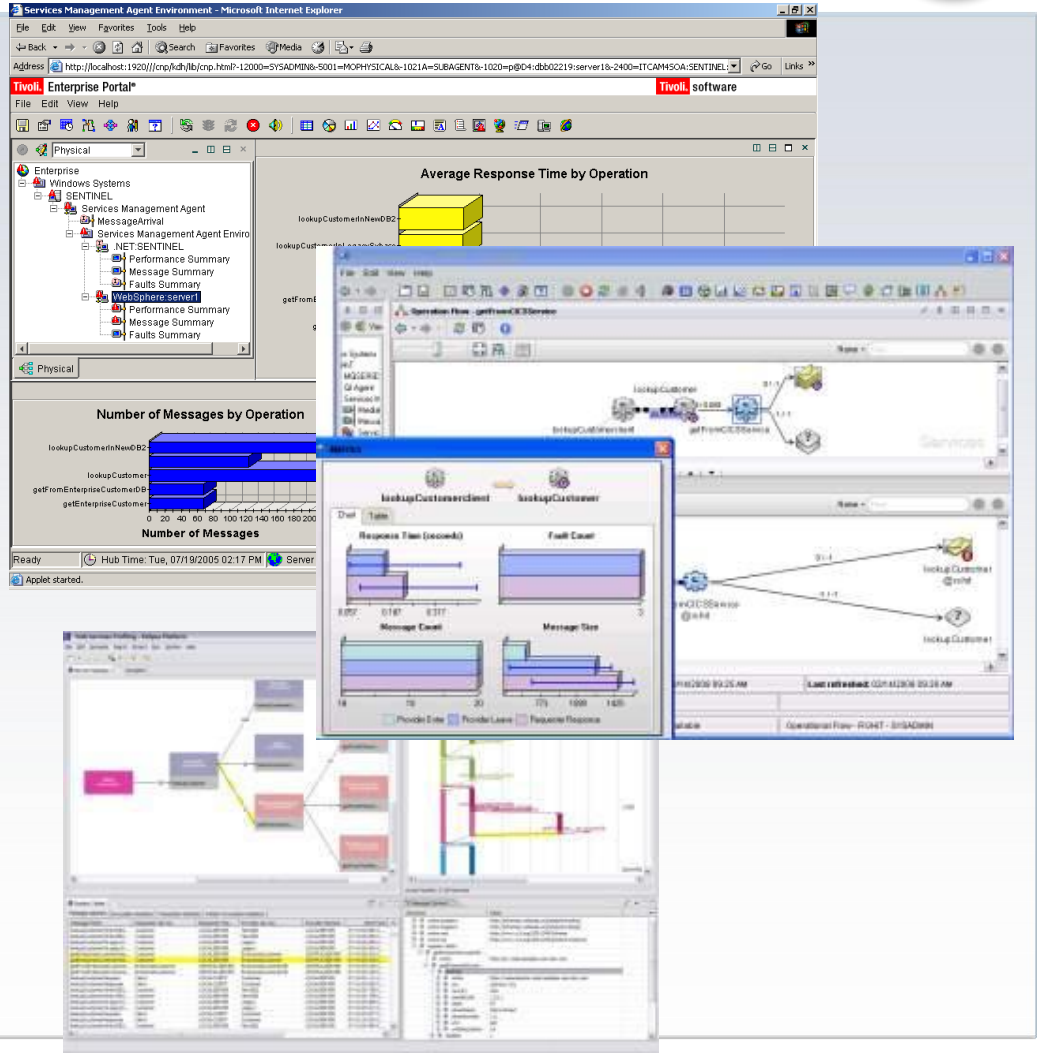
- Validate system scalability
  - Workload modeling for automated generation of test clients
  - Automated generation of performance tests
  - Real-time reporting of server response time and throughput
- Isolate performance bottlenecks and resolve problems
  - Monitoring support for services across multiple platforms
  - Collection and visualization of server resource data – root cause analysis



Monitoring Transaction Performance in SOA

Response Time Metrics in a Distributed Environment

- Composite applications span technology and platform boundaries
- Can be difficult to identify and isolate performance bottlenecks
- Use lightweight instrumentation that can be dynamically configured to proactively identify performance problems
- Correlate data from infrastructure using appropriate instrumentation to show topology and isolate problems



Guidance for SOA Performance

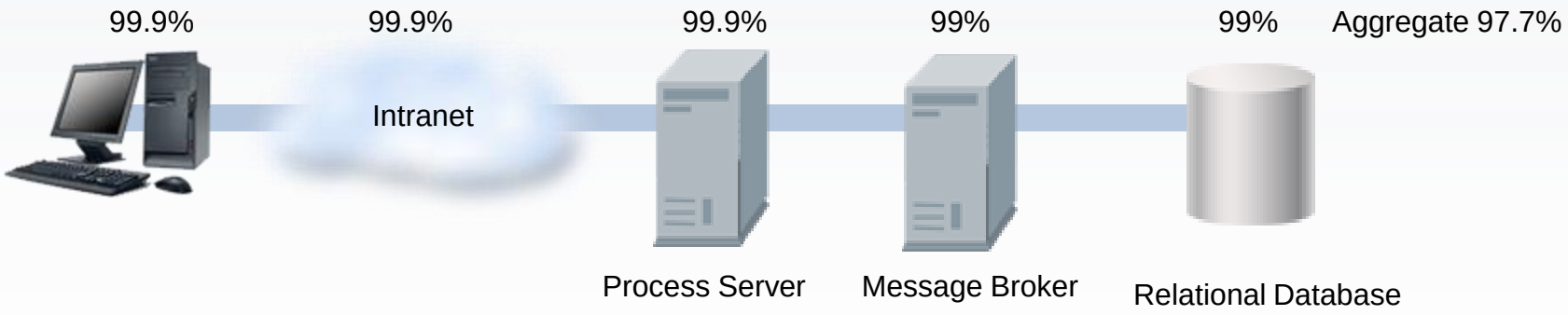
- The SOA performance model should be created and maintained throughout the lifecycle as the application is built
- Performance testing needs to obtain sufficient metrics to validate that services meet performance expectations
- Use established techniques to meet SOA performance requirements
- Design, test, and retest to confirm that non-functional requirements are met
- Implement an integrated solution that will automatically monitor, analyze and resolve response time problems
- Consider dedicated network appliances to optimize and accelerate XML parsing and security processing

High Availability in the SOA World

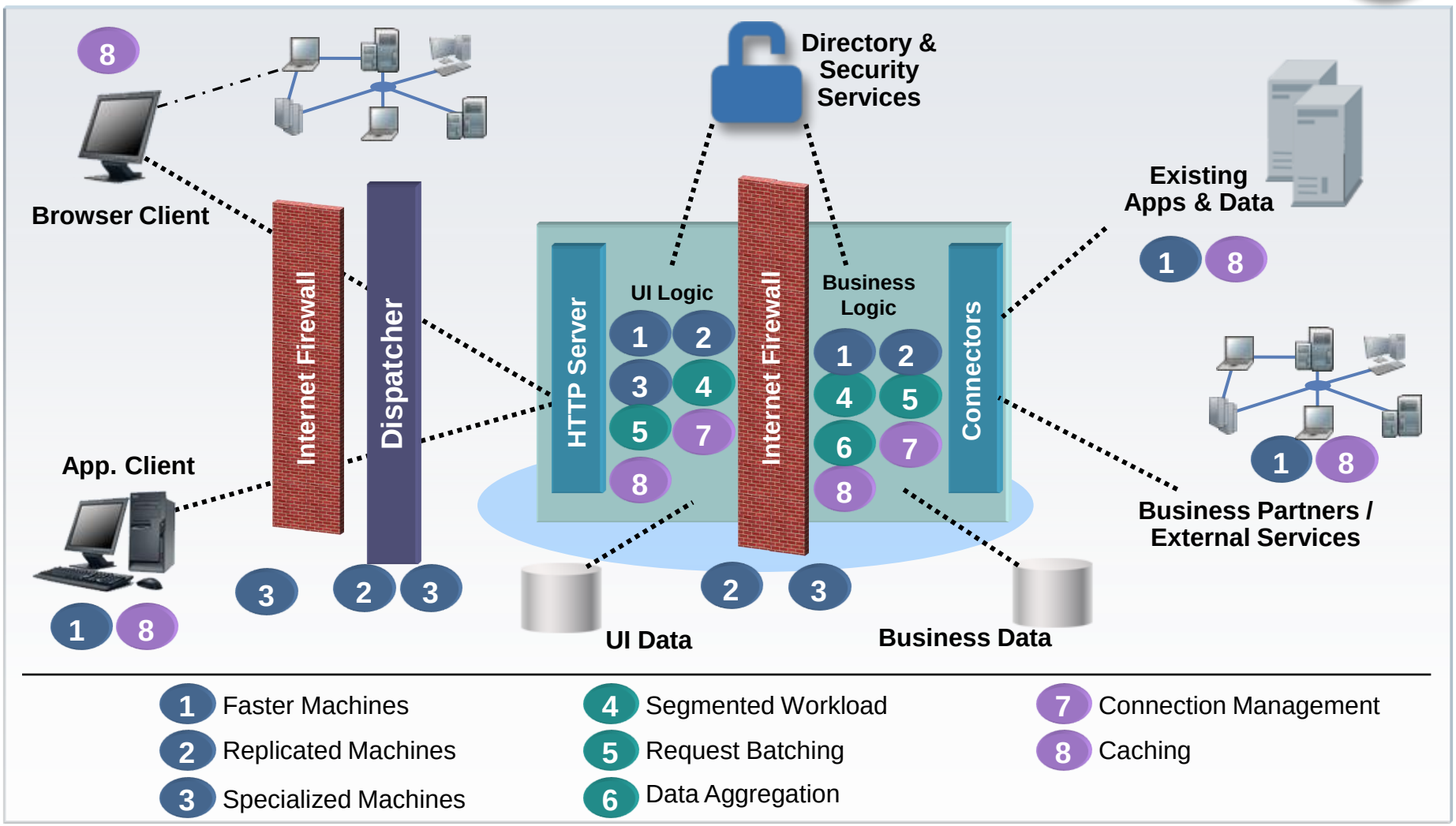
- An application may exist on multiple servers in different locations
  - Infrastructure needs to be “availability” aware in case a service within the workflow is unavailable
- SOA applications impact service availability levels
  - SOA introduce new application dependencies, including externally provided services
  - Need to understand the end-to-end view
- Monitoring, management and reporting is required to achieve predictable availability in an SOA environment
- Plan for the unexpected
  - What are the non-functional requirements? What systems are you using? Distributed? Mainframe? Where are they located? How will they be accessed?
  - The more components in the transaction, the greater the risks for failure or human error

Guidance for SOA Availability

- There are an increased number of components in an SOA infrastructure, so test rigorously for availability
- Create failover plans based on criticality of applications and services
- Take advantage of established availability techniques
  - Each component requires its own availability architecture
  - Leverage capabilities like Workload Management, High-Availability Manager, Deployment Manager, etc.
- Some components may require both hardware and software clustering
  - Databases, enterprise messaging infrastructure, SOA appliances



Techniques for High Availability and Scalability

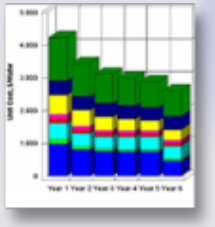


Infrastructure Optimization & Virtualization

Optimize infrastructure investment and prioritize applications and users in a mission-critical manner

- Provide high availability and redundancy for business-critical applications
- Increase server utilization to optimize capital & administration costs
- Ensure that the most important applications and users are given priority according to business and IT policies
- Flexibly respond to unforeseen application demand

Resource Optimization



Utilization

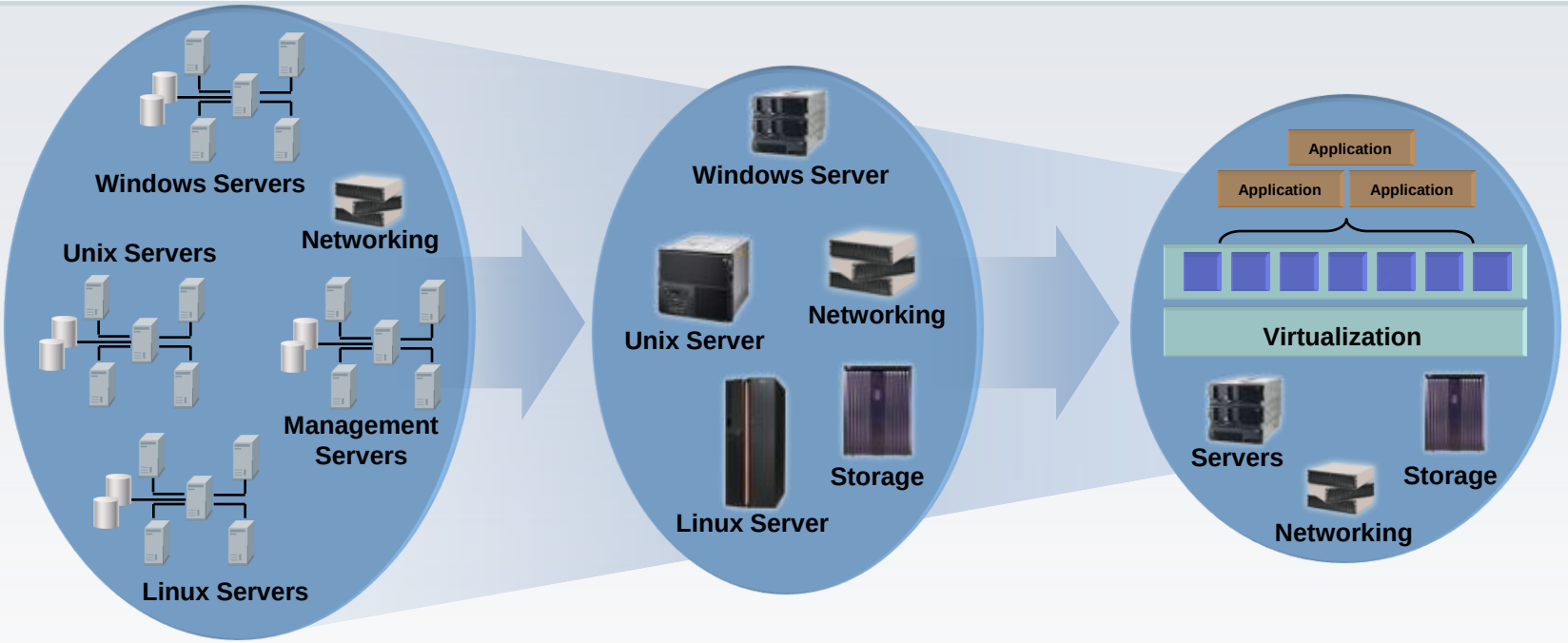
Application Prioritization

Importance

High Availability

Assurance

Virtualization Decouples IT Infrastructure from Applications



Complex

- Islands of computing and data
- Physical resources are bound to applications
- Disparate management tools
- Manual provisioning

Consolidated

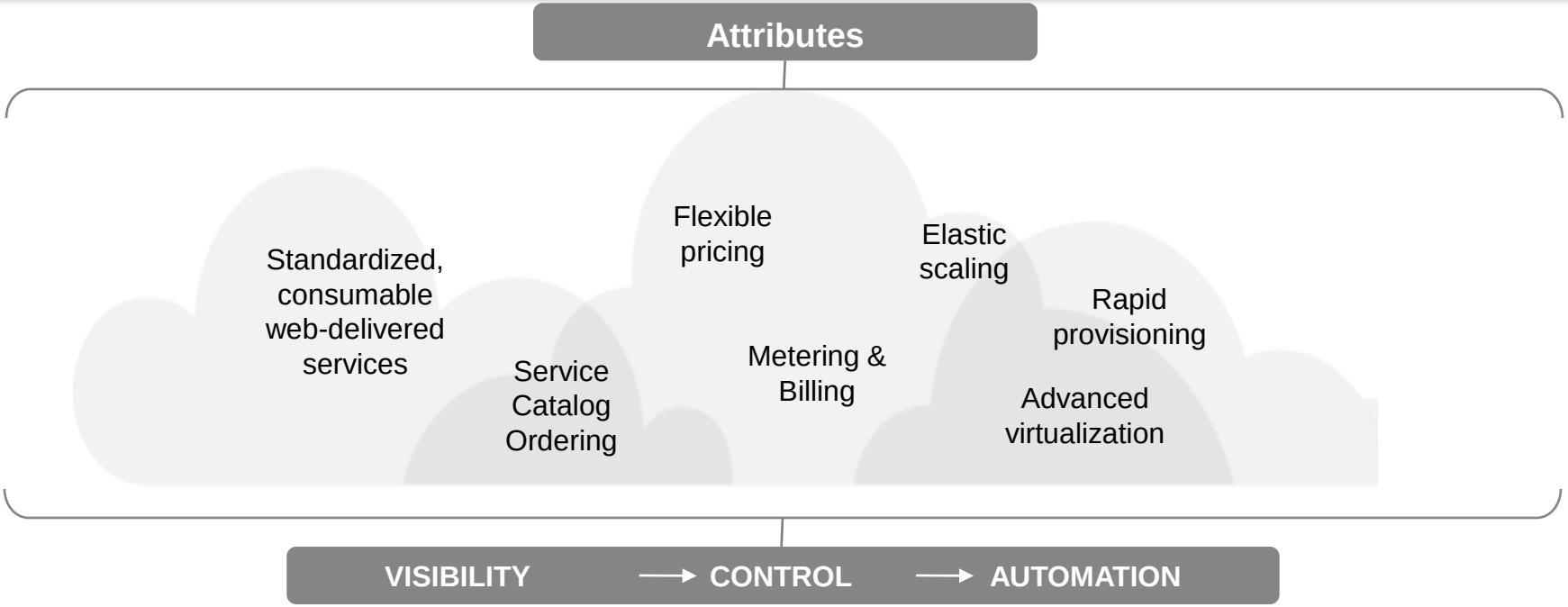
- Fewer devices and licenses
- Increased utilization
- Physical resources still bound to applications
- disparate management tools
- Labor intensive provisioning

Virtualized

- Pools of resources
- Logic and physical resources decoupled
- Standardized, automated infrastructure management
- Automated provisioning

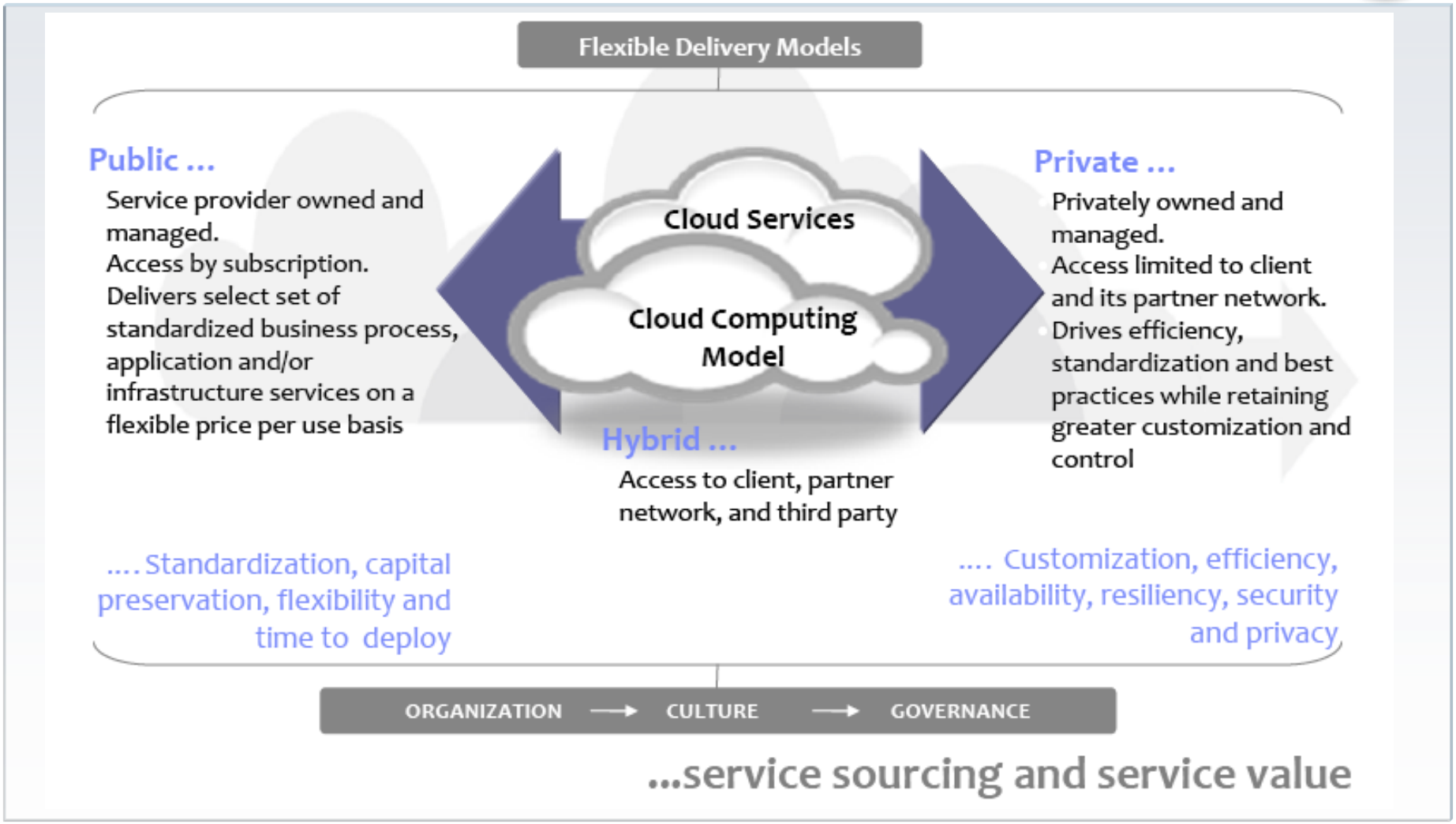
Cloud Computing …

"Cloud" is an emerging consumption and delivery model for many IT-based services, in which the user sees only the service, and has no need to know anything about the technology or implementation



....service oriented and service managed

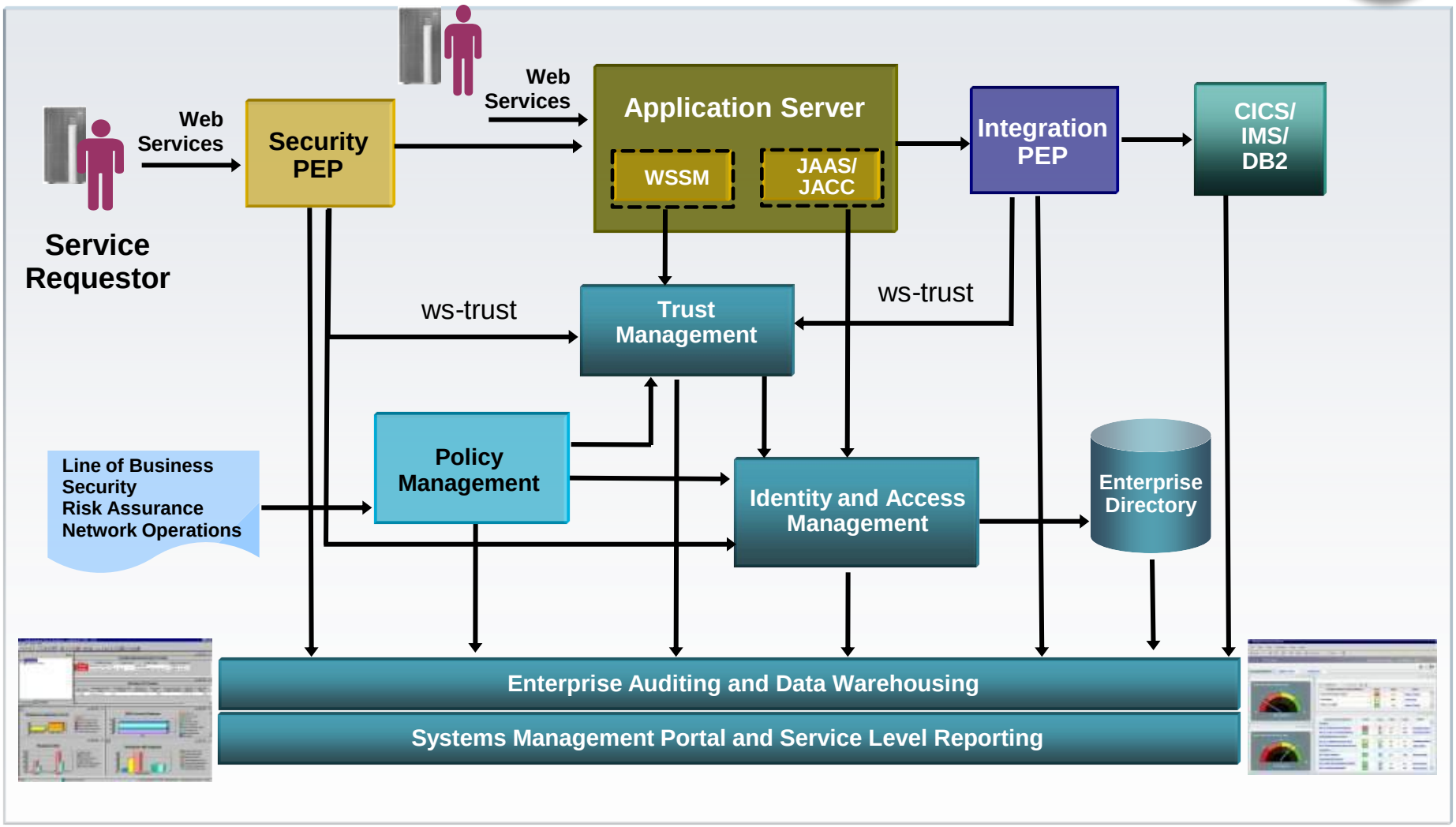
Cloud Computing Delivery Models



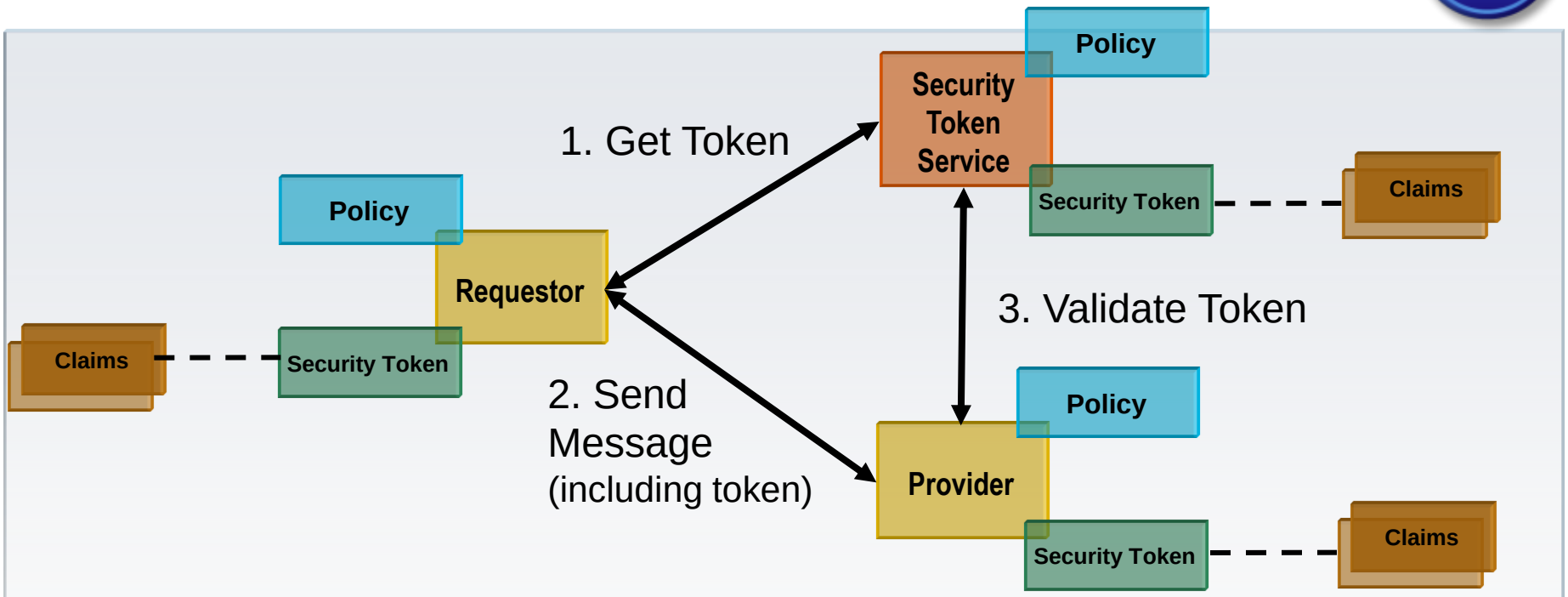
Guidance for Virtualization

- Consolidate servers, storage & network assets for greater efficiency & reduced complexity
- IT resources should be used across applications without regard to where they physically reside
- Replace error-prone manual tasks & repetitive IT resource/capacity management tasks with automated capabilities
- Dynamically allocate IT capacity to meet business goals for increased infrastructure agility and readiness for growth
- Measure resource usage in order to accurately allocate costs

Logical Elements of SOA Security



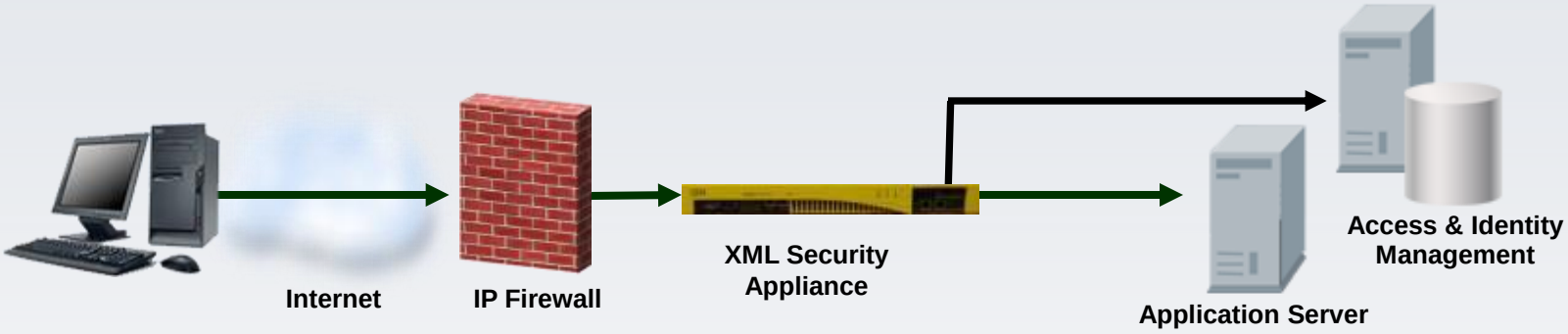
SOA Security – Trust Model



- Identity Federation and Web Services requires trust
  - This trust is based on agreements between partners & expressed as policies
- Trust can be enabled by technology
  - Trust requirements expressed as infrastructure policies and requirements
  - Security tokens include identity information; Cryptographic keys used to sign Security Tokens
- Technology needs to be standards based
  - Standard ways to express and exchange policies that reflect trust relationships
  - Agreed token format, information content, signing and encryption methods

XML Security Appliances

Can Simplify and Accelerate SOA Security



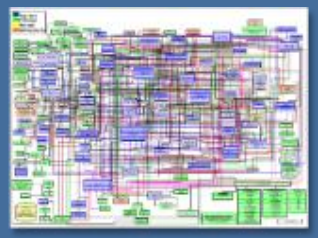
- XML/SOAP firewall enables filtering on any content, metadata or network variables
- Incoming and outgoing XML and SOAP is validated at wire speed
- Security can be performed at the field level
  - WS-Security
  - Encrypt & sign individual fields
  - Non-repudiation
- Provides XML/Web services access control

Guidance for SOA Security

- Security authorization needs to be granular at the service level
- Understand existing corporate security policies (especially approval and audit process) and apply them in the SOA environment
- Work with the SOA application teams to understand the requirements
- Understand the trade-offs of security, performance and cost
- Choose policy-based over programmatic approaches to allow security decisions to be implemented at service invocation
- Evaluate performance implications of security implementations
- Consider XML appliances to accelerate security processing

IBM Service Management

What's happening with the infrastructure?



- Infrastructure and application discovery
- Server monitoring
- Storage monitoring
- Network monitoring
- Data monitoring
- Application monitoring
- Service monitoring

How does this relate to the business service?

- Dashboard
- Application dependency mapping
- Business service management
- Service level management

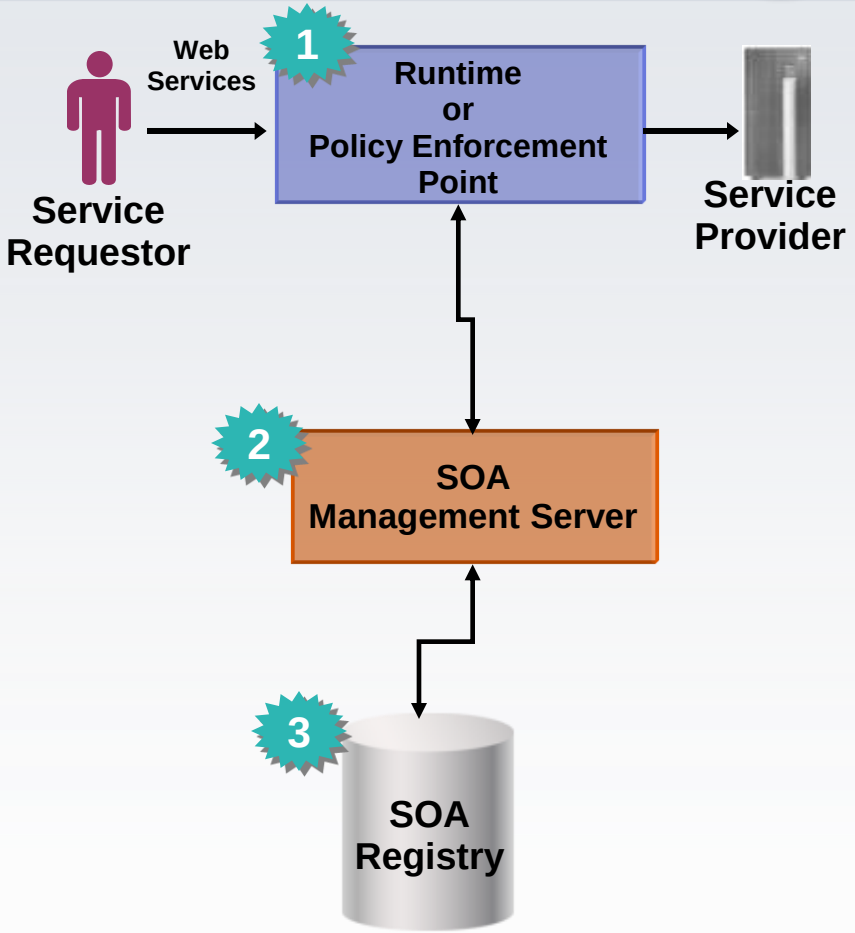
What actions do we take?

- System reconfiguration
- Data restore
- User identity provisioning
- System and application restart
- Infrastructure deployment
- Service mediation

Key Elements for Managing Services

There are 3 key components in services management:

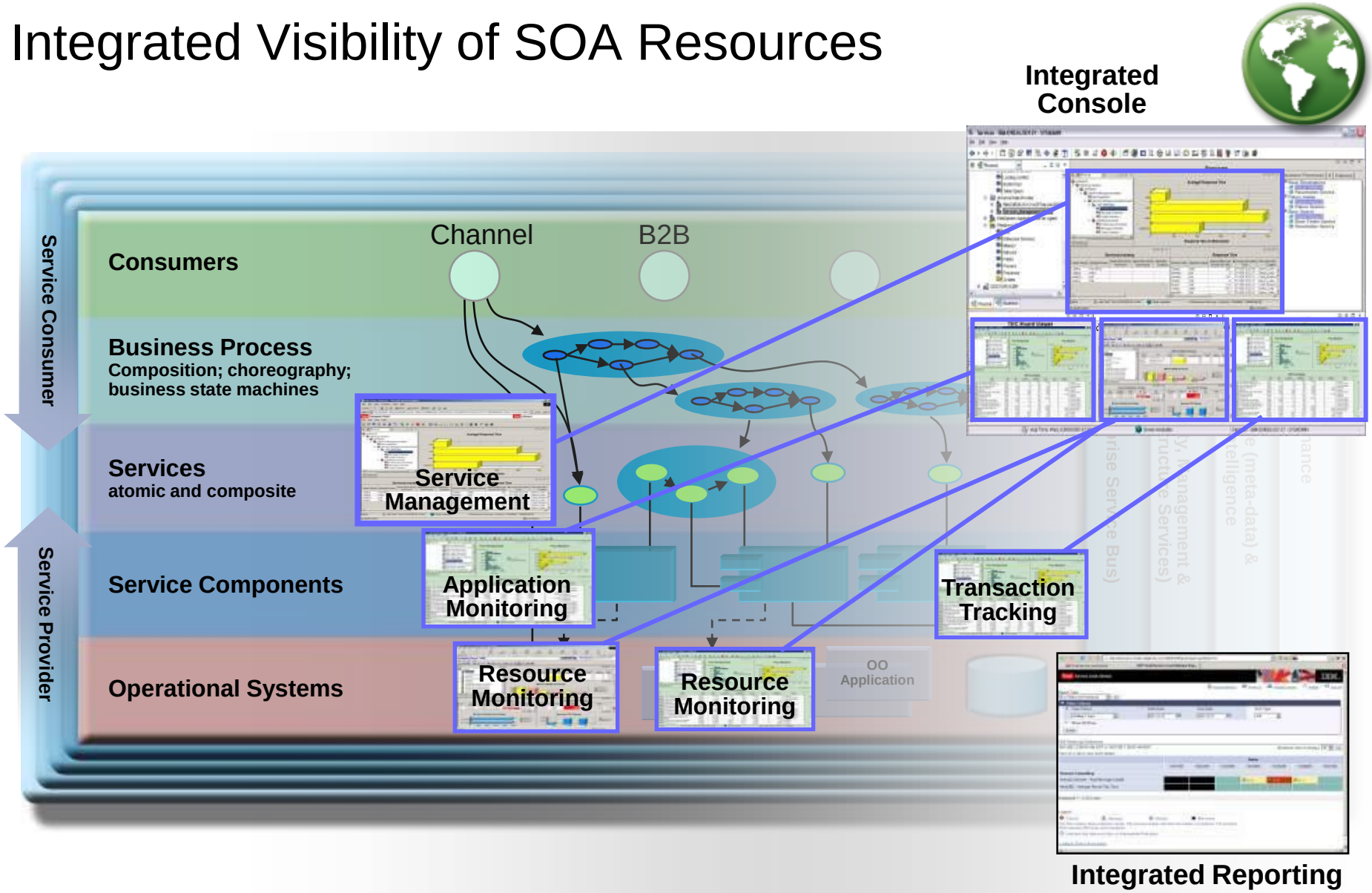
1. The runtime environment – this is where messages are routed, secured, transformed, filtered and logged
2. The management server – aggregates the data from all of the endpoints and runtimes and sends configuration changes based on policy
3. The registry – stores meta data about services and policies



Guidance for Service Management

- Establish operational and business-focused management and monitoring perspectives
- Monitor the end-to-end solution to isolate and fix problems
- Automate provisioning and control of services to meet SLAs
- Make use of tools to improve application availability
- Track/predict change to reduce costs and downtime

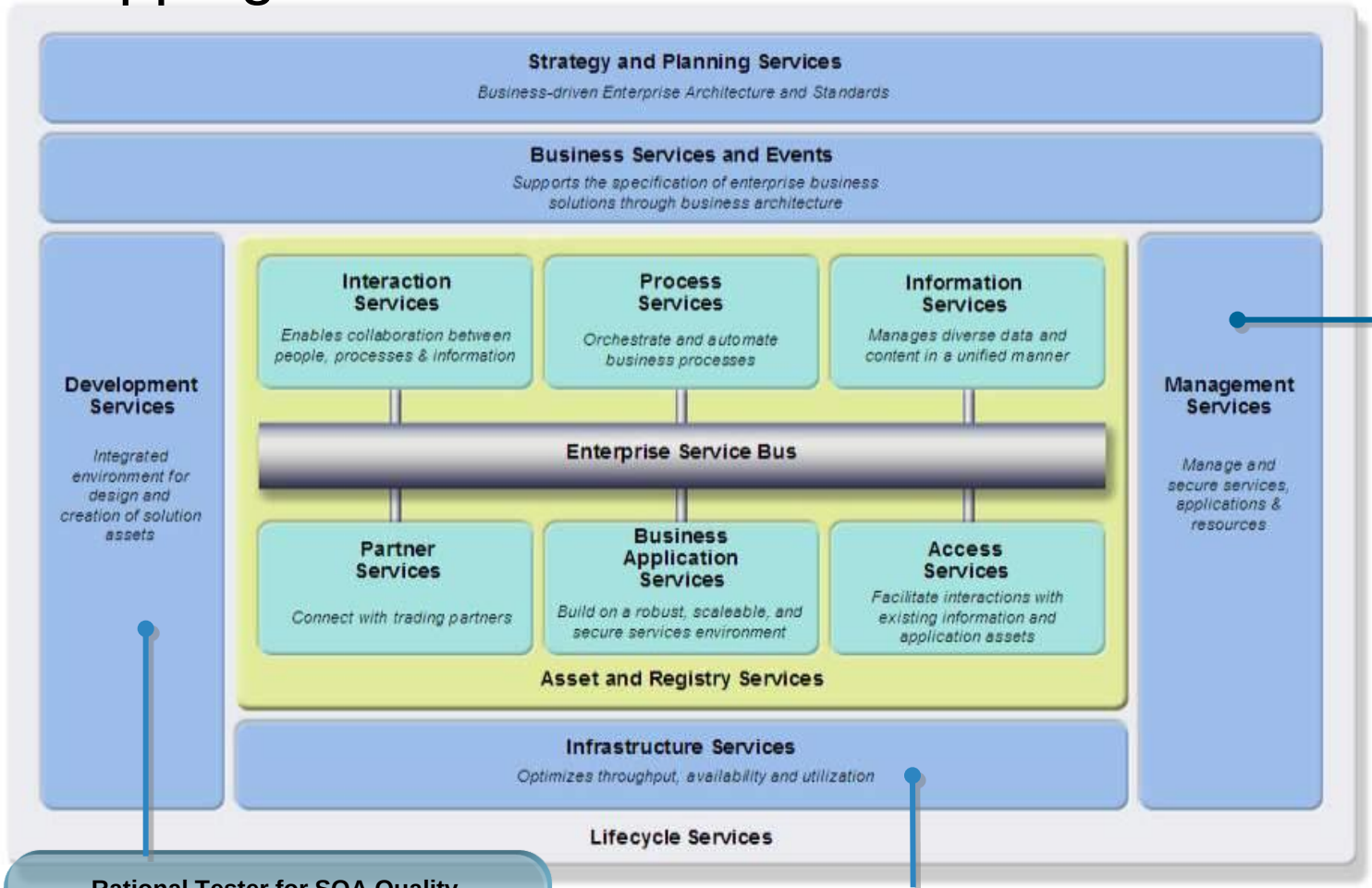
Integrated Visibility of SOA Resources



Agenda

- Planning SOA Infrastructure
- Infrastructure Considerations for SOA
  - Development Services
  - ESB/Connectivity
  - Non-Functional Requirements
    - Performance
    - Availability
    - Virtualization
    - Security
  - Service Management
- **IBM Capabilities to Support SOA Infrastructure Architecture**
- **Summary**

Mapping to IBM Products



Tivoli Composite Application Managers

Tivoli Federated Identity Manager

Tivoli Access Manager

Tivoli Security Policy Manager

Tivoli Provisioning Manager

Tivoli Change & Configuration Management

Tivoli Usage and Account Manager

Rational Tester for SOA Quality

Rational Performance Tester Extension for SOA Quality

Rational Functional Tester

WebSphere Virtual Enterprise

WebSphere DataPower SOA Appliances

WebSphere CloudBurst

धन्यवाद

Hindi

多謝

Traditional Chinese

Teşekkür ederim

Turkish

Спасибо

Russian

Gracias

Spanish

شكرا

Arabic

Thank You

English

Obrigado

Portuguese

Grazie

Italian

Danke

German

Merci

French

Multumesc

Romanian

多谢

Simplified Chinese

감사합니다

Korean

ありがとうございました

Japanese