IBM

IBM SOA Architect Summit

GET PRACTICAL HELP TO MEET THE DEMANDS OF YOUR BUSINESS.

SOA and the Non-Functionals

Mark Ketteman

16th September 2008

Non-Functional Requirements

Performance

Availability

Service Management

Security

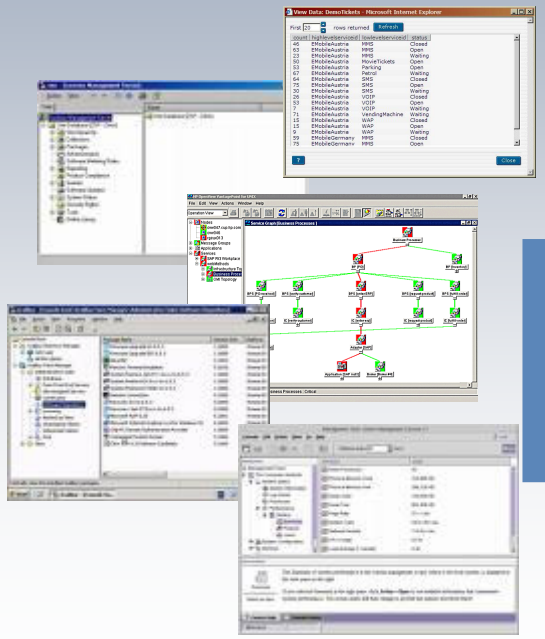
Accessibility
Audit and control
Certification
Dependency on other parties
Documentation
Efficiency
Effectiveness
Escrow
Extensibility
Legal and licensing issues
Interoperability
Maintainability
Open Source
Platform compatibility
Price
Quality
Reliability
Resilience
Resource constraints
Robustness
Scalability
Software, tools, standards etc. Compatibility
Stability
Supportability
Testability
Usability

SOA Represents a Marked Change in IT Prioritization

And Requires a New Way of Thinking

Old Thinking

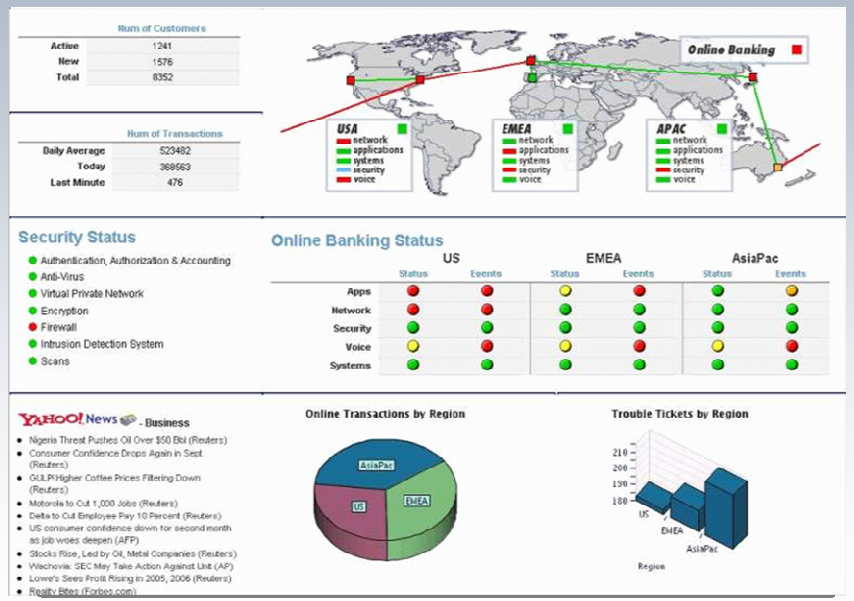
IT *maintains* IT **resources** that support the business



From Silos …

New Thinking

IT *delivers* **services** designed to meet business **goals**



… to Services

IBM

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

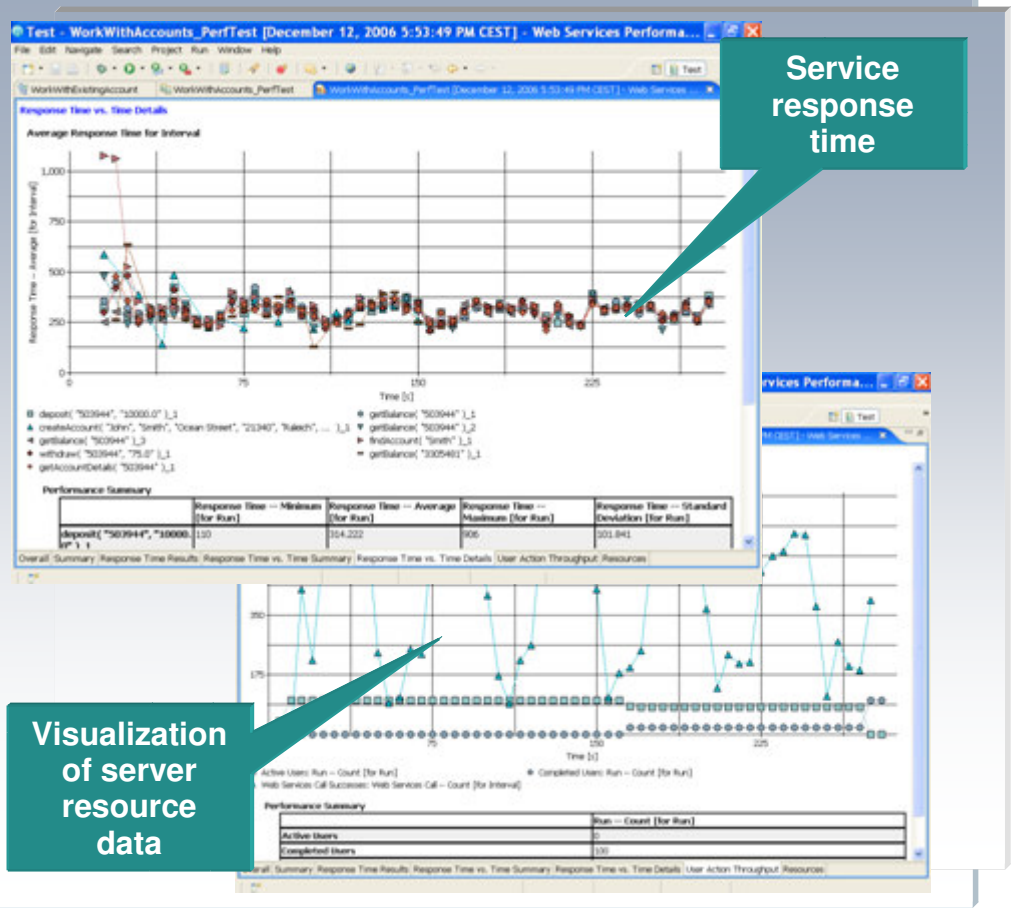
- Authentication, Authorization, Encryption

SOA Performance Testing Concepts

- Test SOA applications throughout the development lifecycle
- Test tools must be separate and external to the SOA environment
- Use multiple diverse datasets that are representative of an SOA workload
- Stress test the solution to detect latency issues
- Run tests in a comparable environment to the deployment environment
- Use multiple test tools – similar results from multiple test tools using identical data sets validates the tests

SOA Performance Testing and Problem Analysis Tools

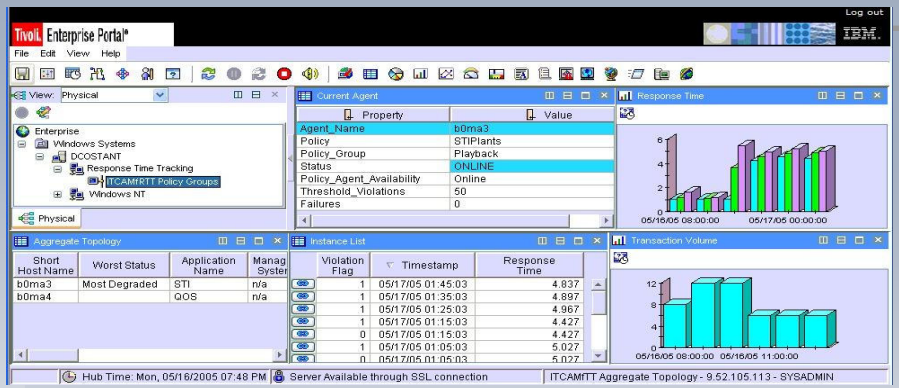
- Validate system scalability
  - Workload modeling for automated generation of test clients
  - Automated generation of performance tests
  - Real-time reporting of server response time and throughput
- Isolate performance bottlenecks and resolve problems
  - Monitoring support for services across multiple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

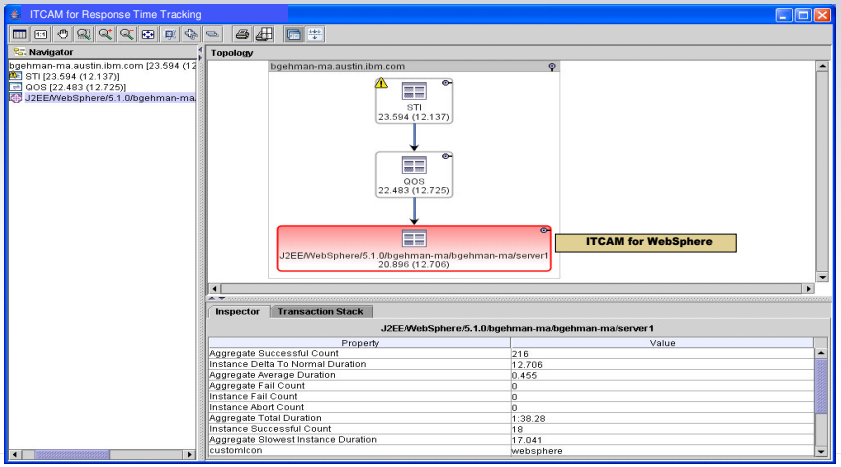


Monitoring Transaction Performance in SOA

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
- Use industry-standard ARM-based instrumentation to isolate the problem





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

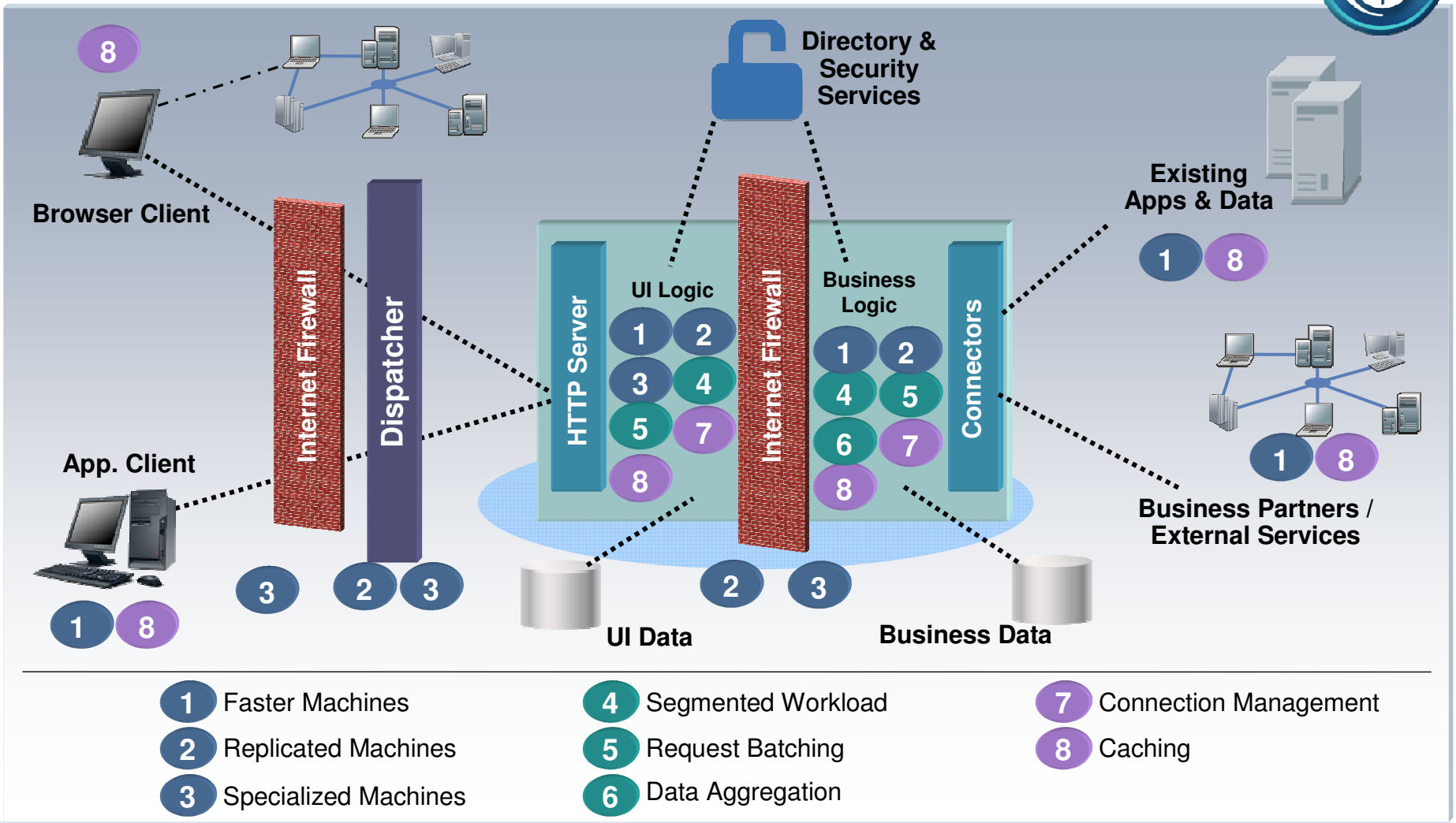
Implement an integrated solution that will automatically monitor, analyze and resolve response time problems

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
  - Applications need 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

Techniques for High Availability and Scalability



Guidance for SOA Availability

There are an increased number of components in an SOA infrastructure, so test rigorously for availability

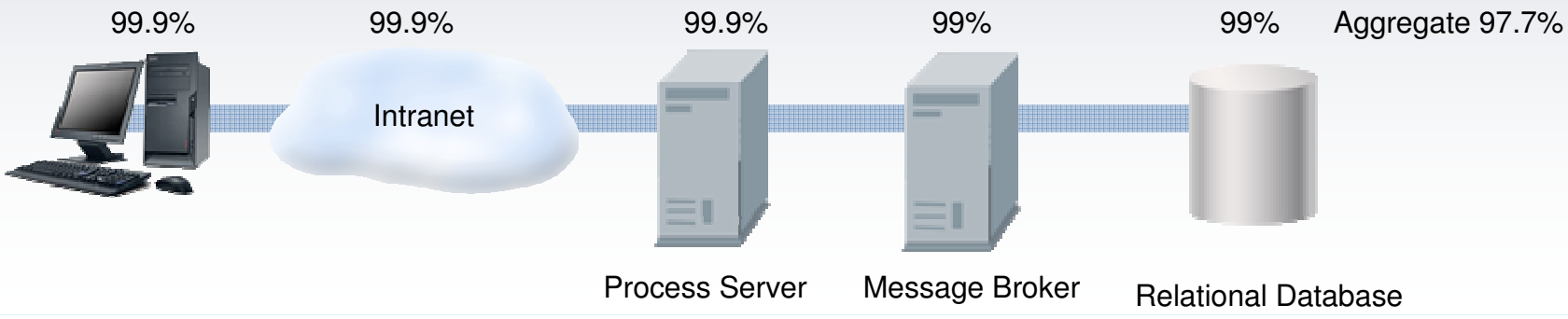
Create failover plans based on criticality of applications and services

Take advantage of established availability techniques

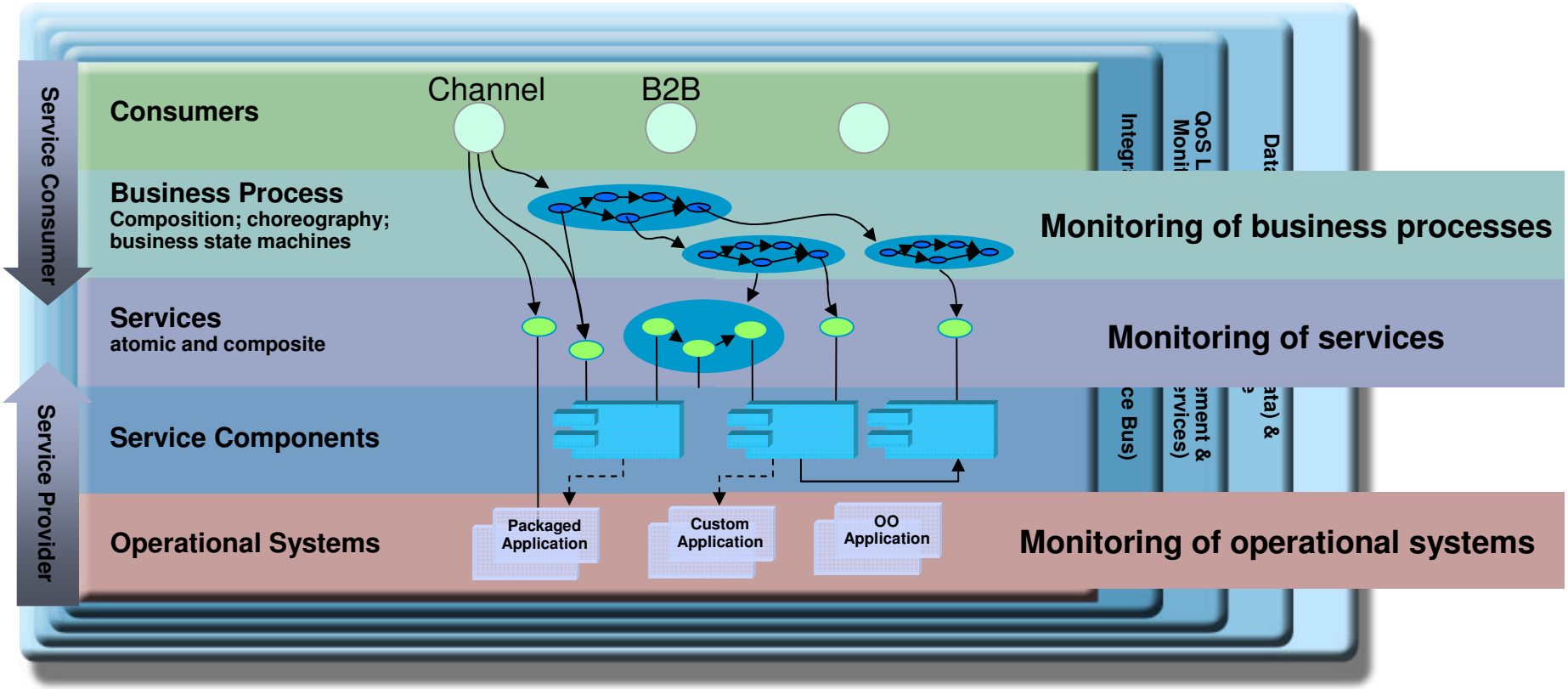
- Each component requires its own availability architecture
- Leverage capabilities like Workload Management, High-Availability Manager, Deployment Manager, etc.

Some components may require both hardware and software clustering

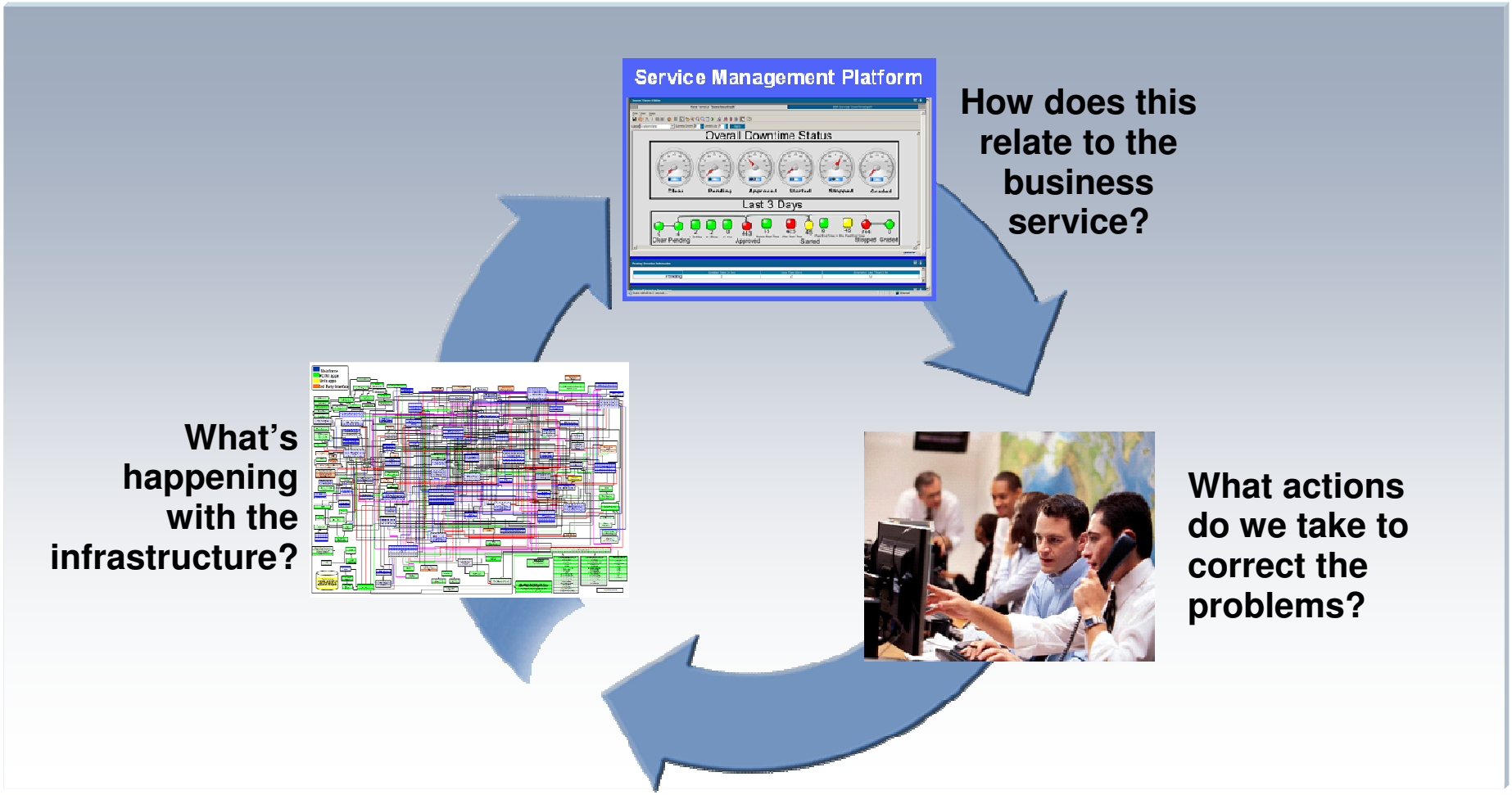
- Databases, enterprise messaging infrastructure, SOA appliances



The Challenges of Managing SOA

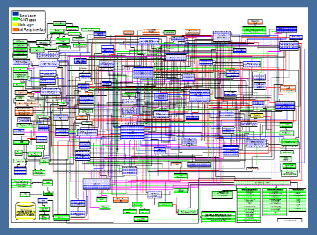


Service Management
Requires a Closed-Loop Approach



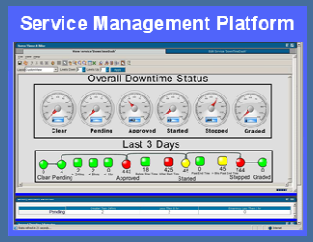
IBM Service Management

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

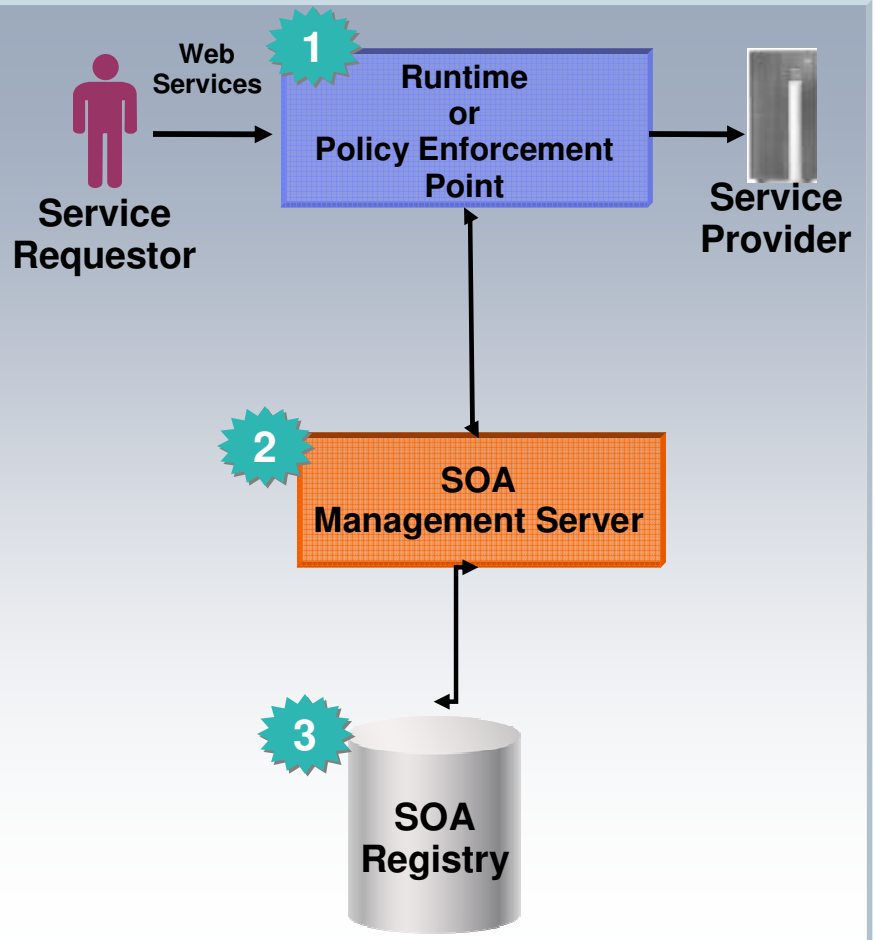
What actions do we take?

- 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



Dynamic Service Support

Dynamically Changing Service and Application Relationships

Support Change Process

- Initial state
- Use to validate that planned changes were executed and that the results are as expected

Pre-Change Validation

- What applications does a component support?
- Reduce unintended consequences

Improve MTTR (Mean Time to Resolution)

Accurate application maps show you what is important

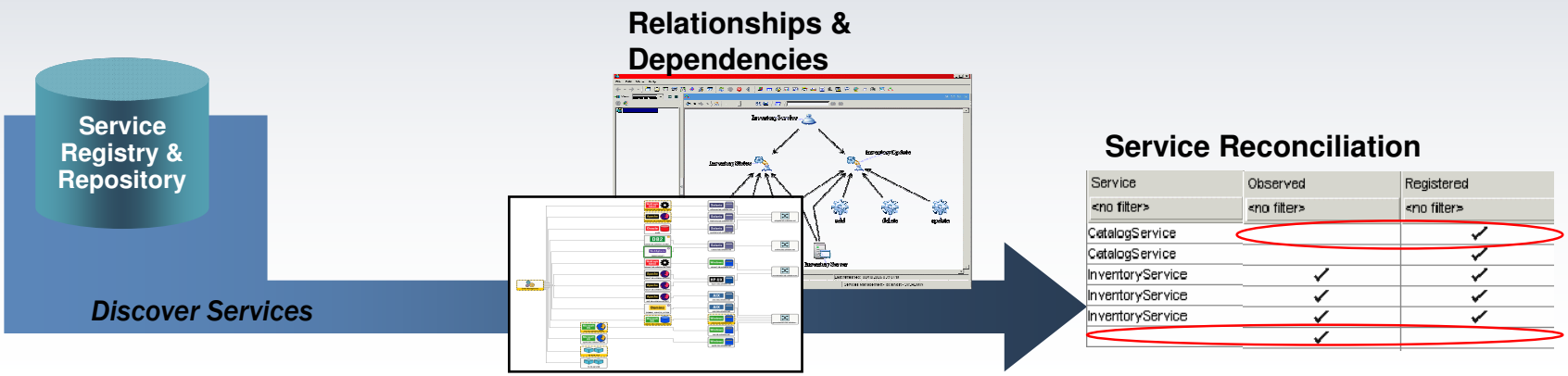
Find the “Last Change” before a problem shows up

Simplify impact analysis

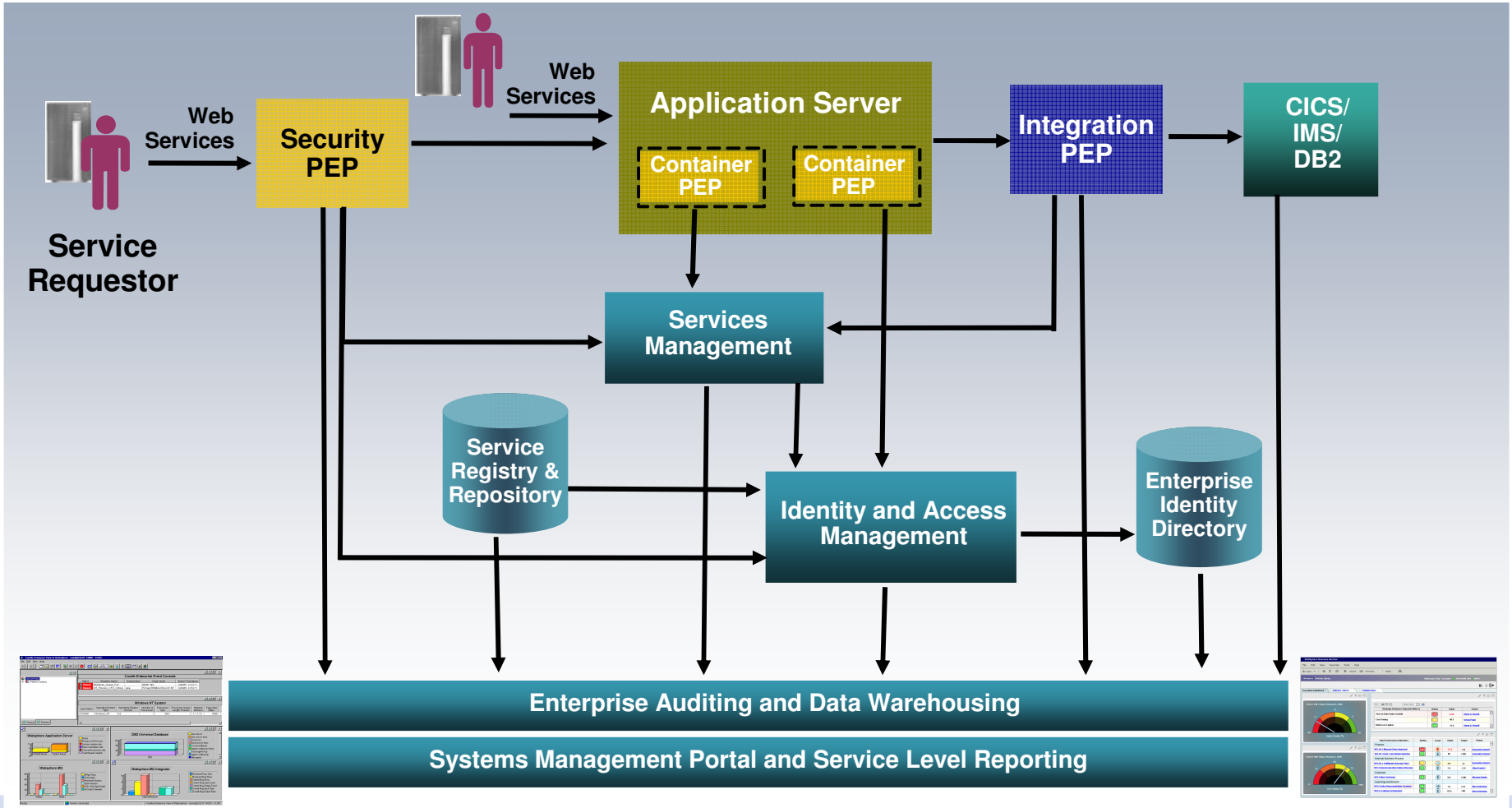
Configuration Drift

Notice when Configurations change and notify operations

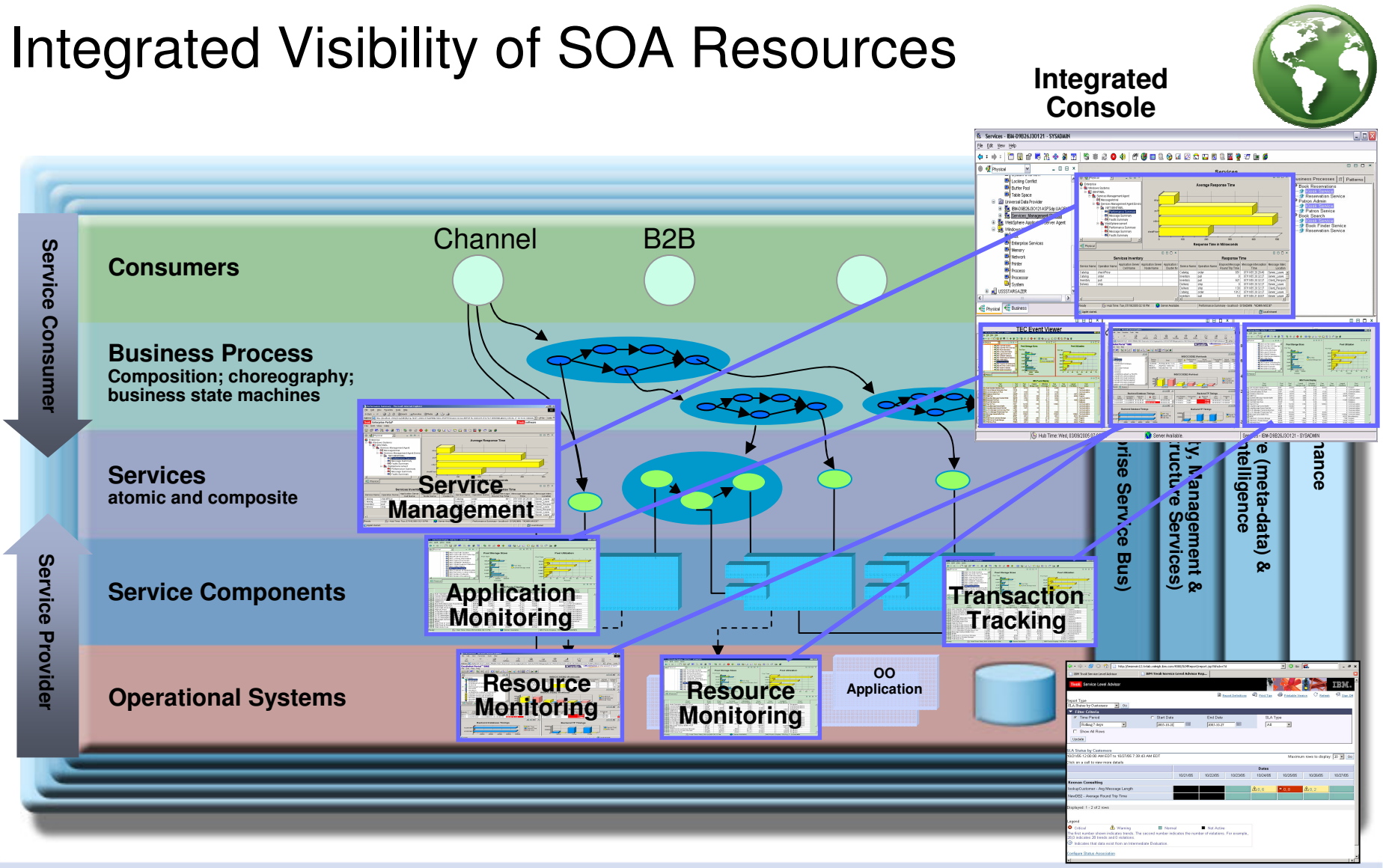
Keep “bit-rot” from impacting operational readiness



Logical Elements of an SOA Management Solution



Integrated Visibility of SOA Resources



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

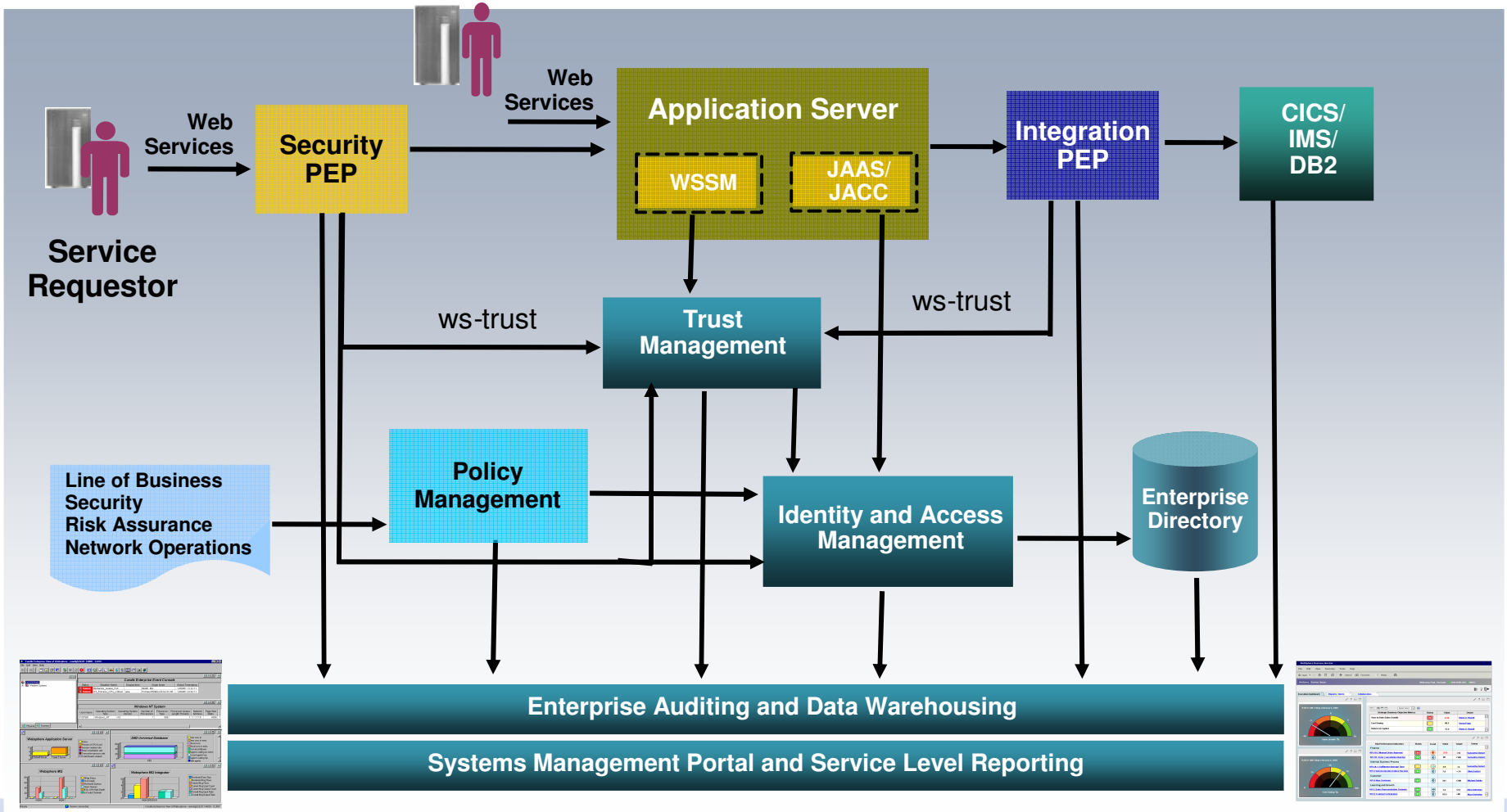
Make use of tools to improve application availability

Track/predict change to reduce costs and downtime

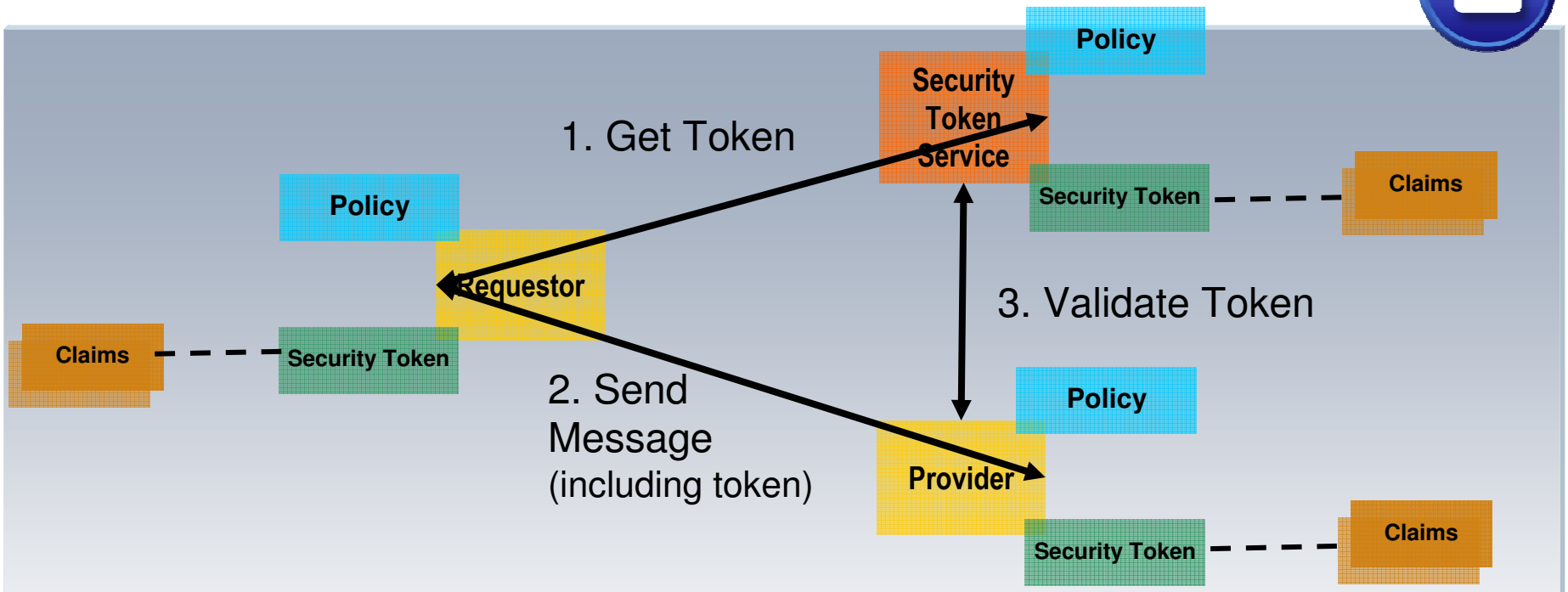
SOA Security Considerations

- SOA introduces raise additional security issues
  - How do we identify and authenticate the service requester?
  - How to we identify and authenticate the source of the message?
  - Is the client authorized to send this message?
  - Can we ensure message integrity & confidentiality?
  - How do we audit the access to services?
  - How do we leverage Web services security standards?
  - How do we propagate identities with trusted service providers?
- XML Web services may expose backend systems in unintended ways
- SOA security may require multiple layers of enforcement – perimeter, gateway, app server, application
- Traditional security devices do not secure XML/SOAP

Logical Elements of SOA Security



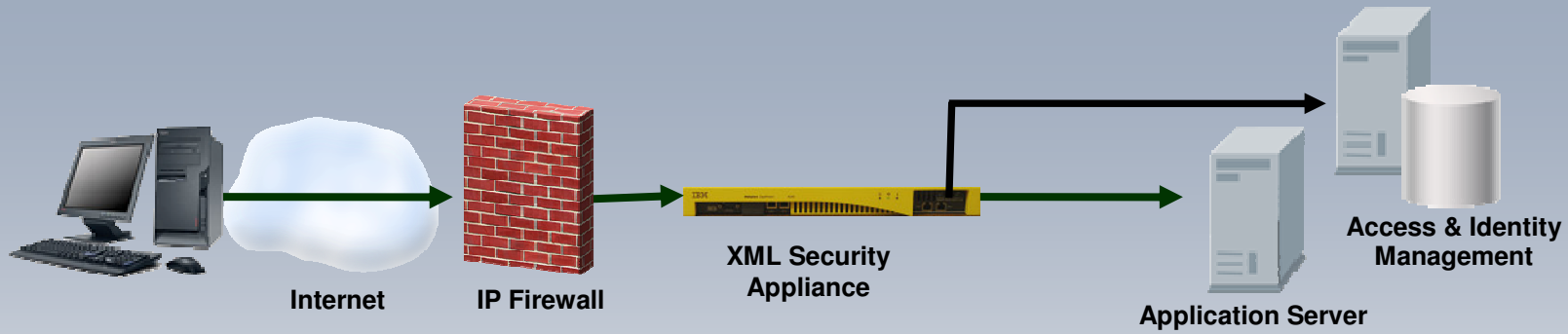
SOA Security – Trust Model



- 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

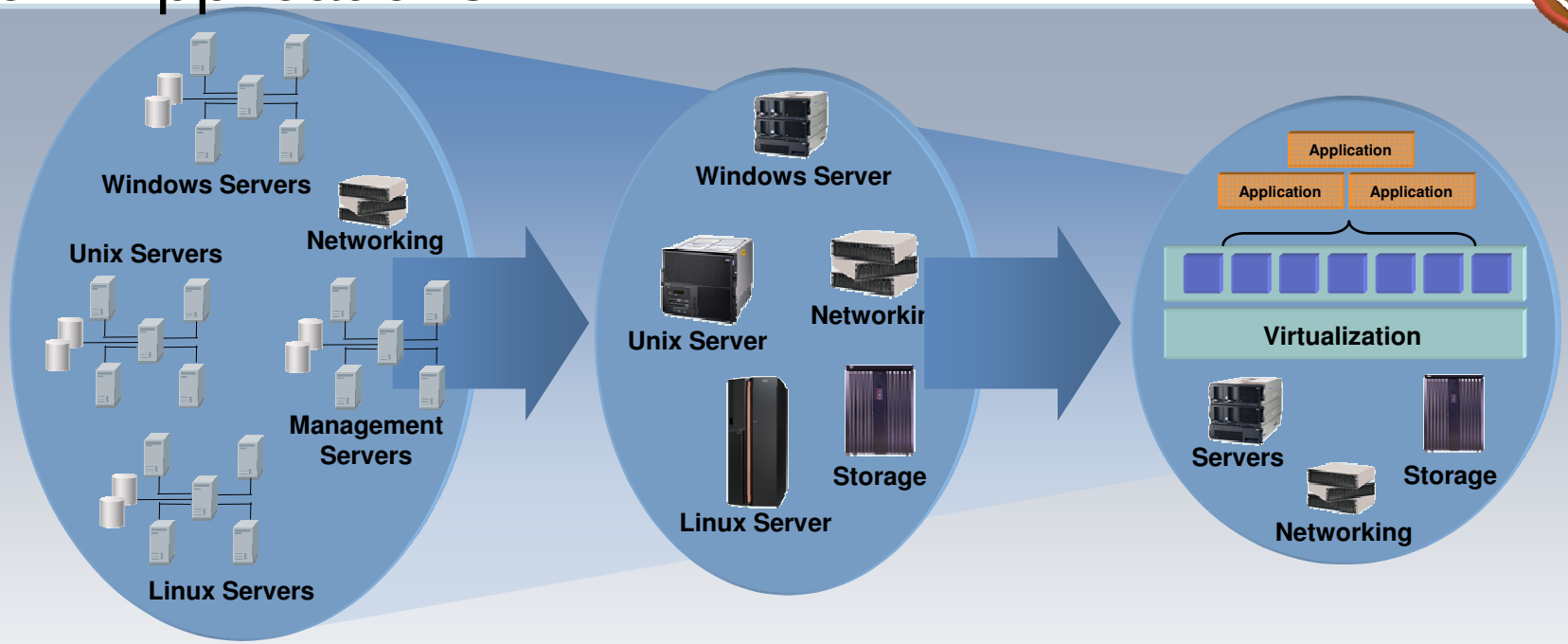
Understand the trade-offs of security, performance and cost

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

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
- Labour intensive provisioning

Virtualized

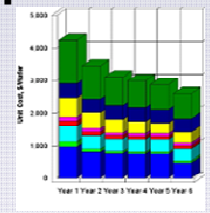
- Pools of resources
- Logic and physical resources decoupled
- Standardized, automated infrastructure management
- Automated provisioning

Infrastructure Optimization & Virtualization

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

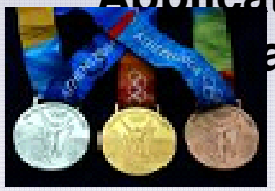
- Provide high availability and redundancy for business-critical applications
- Increase server utilization to optimize capital & administration costs
- Ensure that the most important applications and users are given priority according to business and IT policies
- Flexibly respond to unforeseen application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

Resource Optimization



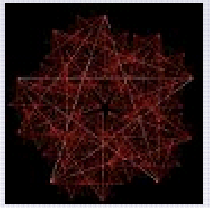
Utilization

Application ation



Importance

High Availability



Assurance

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

The Keys to Architecting an SOA Infrastructure

In the real-world, SOA-based applications put a lot of stress on a typical infrastructure

From a business view, the application layer is geared towards simplification but the infrastructure can become complex

The IT Infrastructure/Middleware Architect cannot let the SOA application become a *"black box"* within the infrastructure

Visibility of quality of service metrics within the SOA application is crucial to achieving performance and availability goals

As an IT Infrastructure Architect, one needs to know what is in the toolbox and how to build the best infrastructure for the SOA application