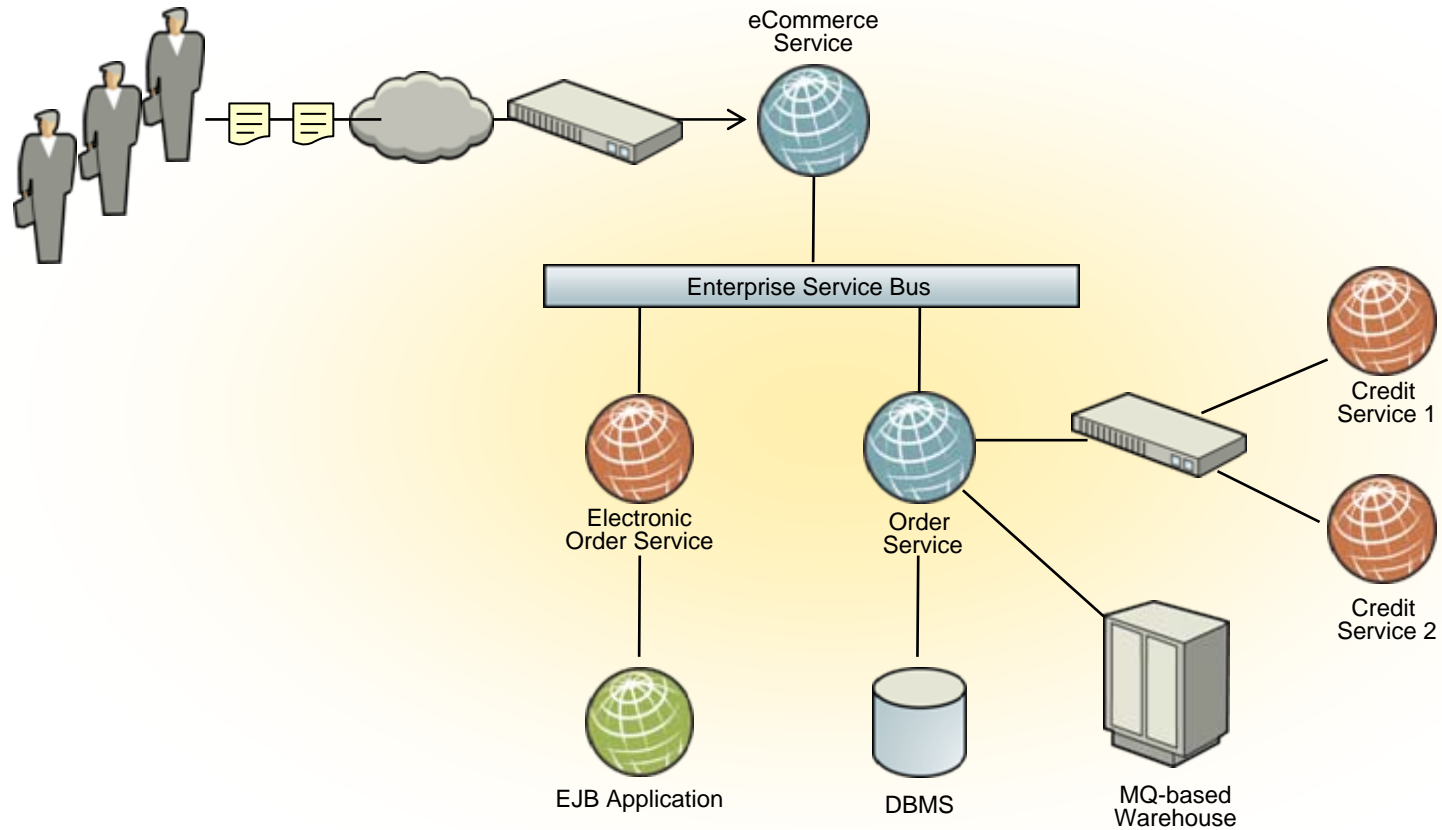


AMBERPOINT

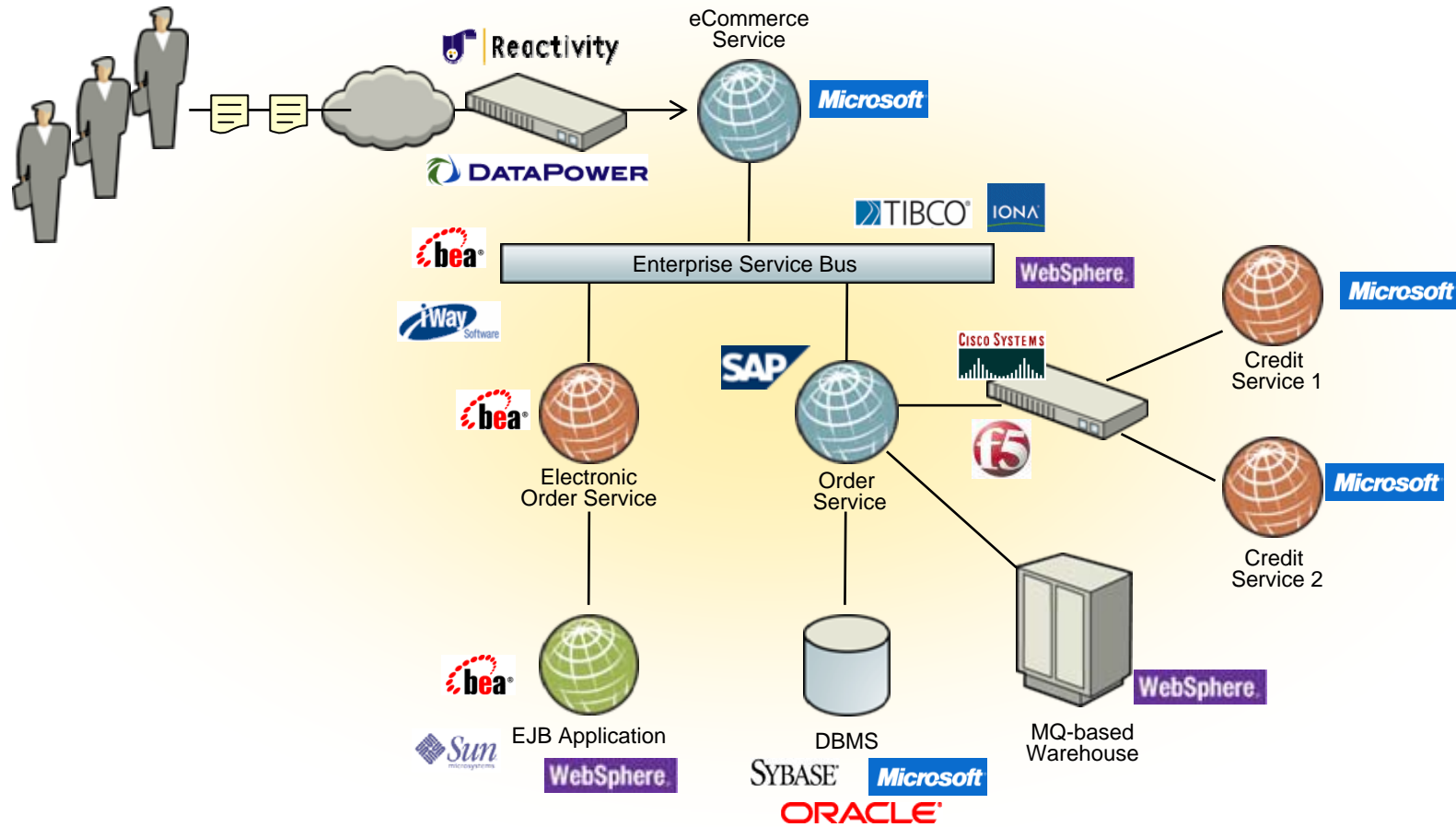
Securing and Managing a Web Services (SOA) Environment

Ed Horst, AmberPoint
INTEROP
May 23, 2007

● Wide Variety of SOA Infrastructure



Wide Variety of Vendors for SOA Applications

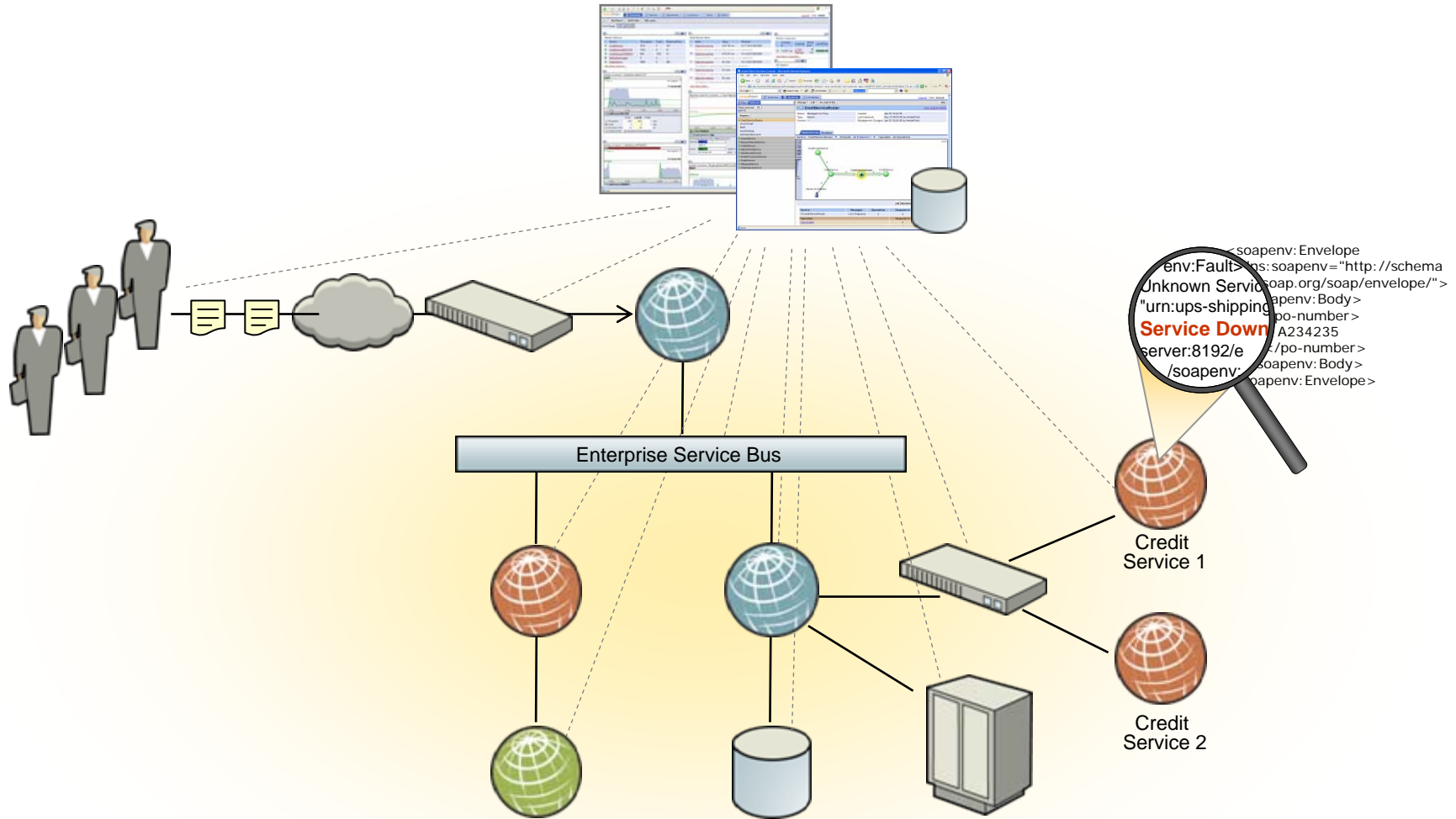


● Keys to Successful SOA Runtime Governance

- ◆ Knowing What's Out There and What's Going On...
 - What services, SOA components, and infrastructure are installed and in use
 - Who's using it
 - Is it meeting expectations / obligations
- ◆ Controlling It...
 - Define and enforce runtime policies – make sure proper policies are active
 - Diagnose failures / prevent them
- ◆ Ensuring Integrity...
 - Automatically check for the correctness of the running system
 - Detect and validate changes before they impact users and partners
- ◆ And, do it across the entire infrastructure
- ◆ Do this all as automatically as possible
 - Reduces risks and costs
 - Automation is the single most important thing that makes SOA scaleable

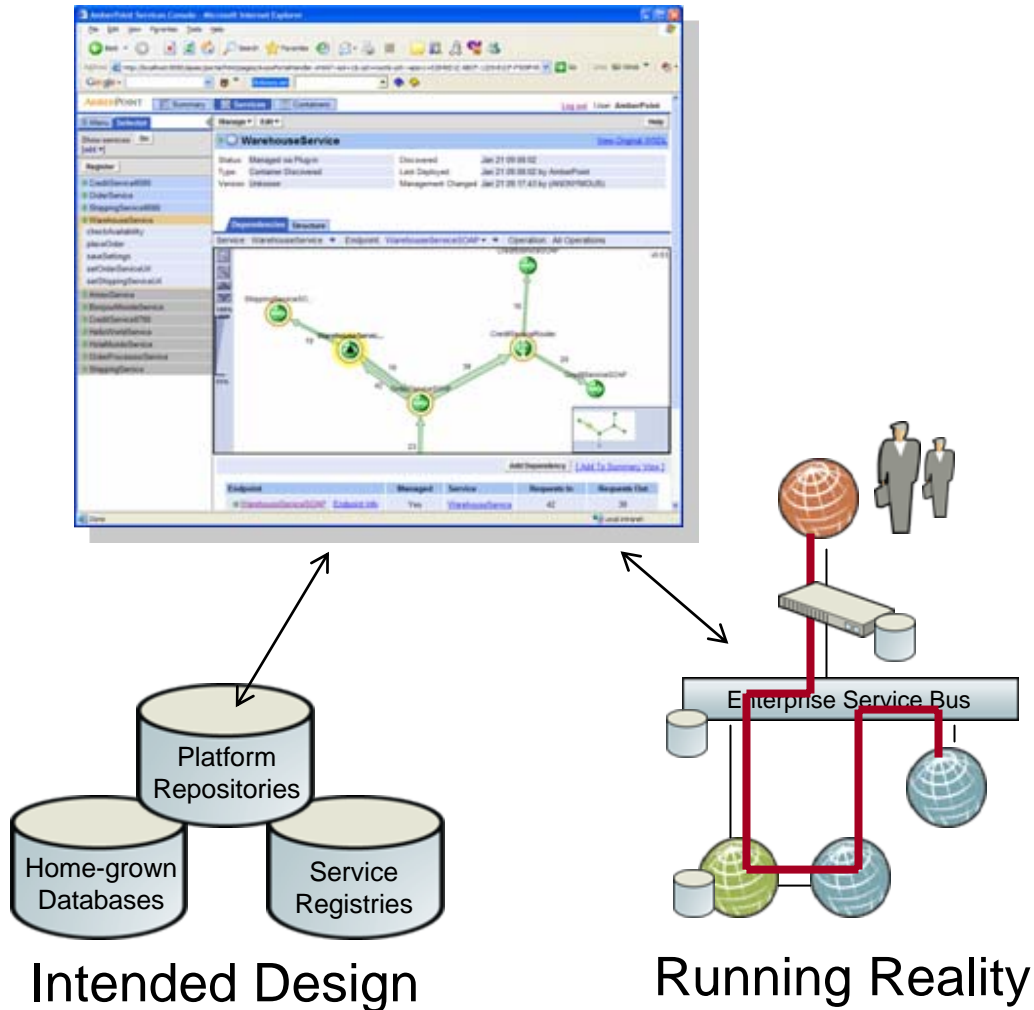
SOA Runtime Governance...

Real-time Visibility, Control and Validation



Visibility, Control, and Validation regardless of SOA Infrastructure Decisions

● Automatic End-to-End Visibility



- ◆ Dynamic discovery of SOA environment...
 - Services
 - Consumers
 - Dependencies
 - Transactions
 - Runtime Policies
 - Runtime Metadata
- ◆ ...across heterogeneous infrastructure
 - Containers
 - ESBs
 - Appliances
 - Registries / Repositories
- ◆ Needs to be non-invasive; no message modifications
- ◆ Should work for development, QA/staging, and production

Real-time World View

What's out there and what's going on...

The screenshot displays the AmberPoint Services Console interface. On the left, there are 'FILTER' and 'STATISTICS' sections. The main area shows a network diagram with nodes for ReservationService, RegistrationService, OrderService, BuildingService, and Clients (250). A 'SERVICE SUMMARY' table is visible at the bottom, and a 'Failure' popup is shown over the 'Faults in last 1 hour' section.

Filters (pointing to the left sidebar):

- Managed
- New to the system in the last 24 hrs
- With Policy of type Security
- With Violations
- With Exceptions
- Used by customers
- Currently limiting traffic
- Name

Statistics (pointing to the left sidebar):

- Messages last 1 hrs
- Faults last 1 hrs
- Avg. Response Time last 1 hrs
- Customer Usage
- Alerts
- Exceptions

Service Summary Table (pointing to the bottom section):

Container	Deployment	Management	EndPoints	Operations
OrderService	thalia:8080	KestrelTestbed	2	12

Failure Details Table (pointing to the popup):

Time	Details	Target	All
15:32, 09.21.2006	Maximum Response time is 523ms	400ms	Details
15:32, 09.21.2006	Average Response time is 65ms	60ms	Details

Drill Down

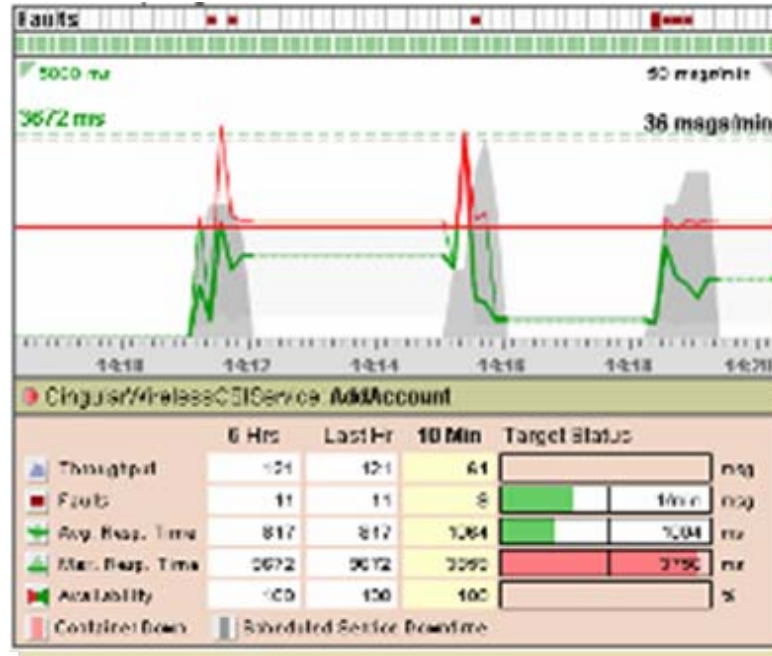
Summaries

- ◆ Quick filters to rapidly isolate areas of interest
- ◆ Point-and-click drill down with float over summaries & detail

Performance Monitoring

Real-time tracking of basic service performance

Maximum Response Time



Faults

Alerts at different warning levels

Detailed Metrics

- ◆ Knowledge of basic service performance
 - Throughput
 - Faults
 - Average and Max Response Time
 - Availability

● *Now What?*

- ◆ A “Monitoring only” definition of “management” ends here.
- ◆ Control is required to do anything more...

Management without Control is like a police force that sits in their cars all day and just writes down the crimes that they see happening...

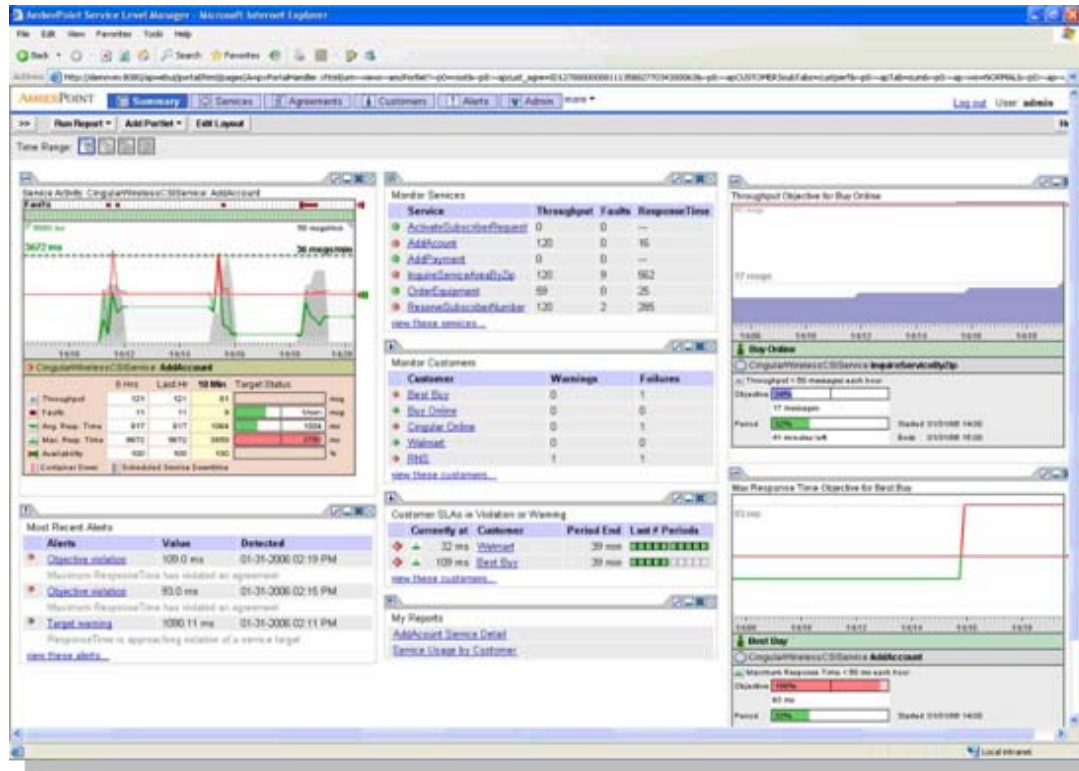
Successful, scalable management of SOA requires the ability to take action and intervene

Service Level Management

SLA enforcement for services, users, and groups

Selectable Time Settings

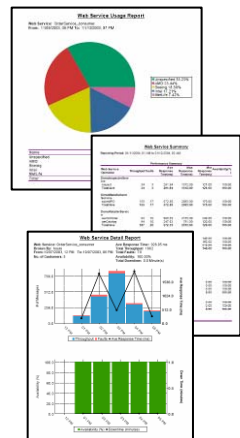
Service Level Violations



Visibility by User

Multiple Objectives per Agreement

Historical Reporting



- ◆ Enforces agreements based on business criteria
 - “Gold” users, Accounting systems at the end of quarter, etc.
- ◆ Preventative and corrective actions; E.g.:
 - Activate more resources (blade servers, load balancing, etc.)
 - Redirect traffic
 - Make “bronze” users wait until “gold” users complete transactions

Security

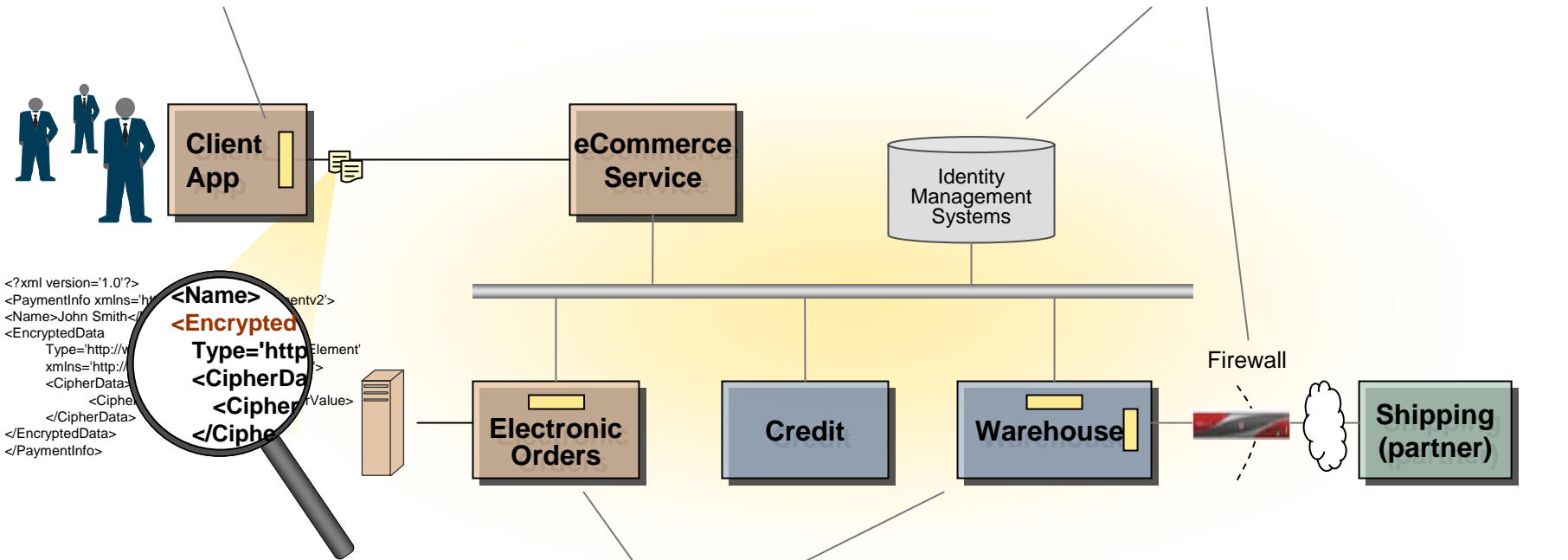
First and Last Mile Enforcement

First Mile Security

- Client-side agent
- Automatic enforcement of out-bound security

Extensive Integration

- Identity Management Systems
- Security Appliances
- App Server / ESB / OS Security



Last Mile Security

- Plug-ins provide endpoint protection
- No ability to circumvent

Complete Policy Library

- Authentication
- Authorization
- Credential Mapping
- Censorship
- Crypto

Automatic Client-side Compliance

Eliminates client-side coding and re-coding

Manually recode, retest, redeploy on each platform as policies change



Client App

```
New Security Policy #3 - .Net Version
New Security Policy #2 - Java Version
New Security Policy #2 - .NET Version
Security Policy #1 - Java Version
Security Policy #1 - .NET Version
...
SoapContext requestContext = Proxy.RequestSoapContext;
// Get an X.509 certificate & sign the SOAP message.
Sign();

// Get an X.509 certificate for encrypting the SOAP message.
X509SecurityToken encryptionToken = GetSecurityToken(false);
if (encryptionToken == null)
{
    throw new FalconException("No token found for encrypting message");
}
else
    Console.WriteLine("Got encryptionToken...");

// Add the X.509 certificate to the WS-Security header.
requestContext.Security.Tokens.Add(encryptionToken);

// Specify that the SOAP message is encrypted using
// this X.509 certificate.
EncryptedData enc = new EncryptedData(encryptionToken);
requestContext.Security.Elements.Add(enc);

// Set the TTL to 1 minute.
requestContext.Security.Timestamp.TtlInSeconds = 60;
}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
}
}

public void Sign()
{
    try
    {
        SoapContext requestContext = Proxy.RequestSoapContext;

        // Get an X.509 certificate for signing the SOAP message.
        X509SecurityToken token = GetSecurityToken(true);
        if (token == null)
        {
            throw new FalconException("No token found for signing message");
        }
    }
    ...

```

Security Policies



Client App

Look for client-side agents that automatically adjusts to implement matching client-side request.

Security Policies



Exception Management

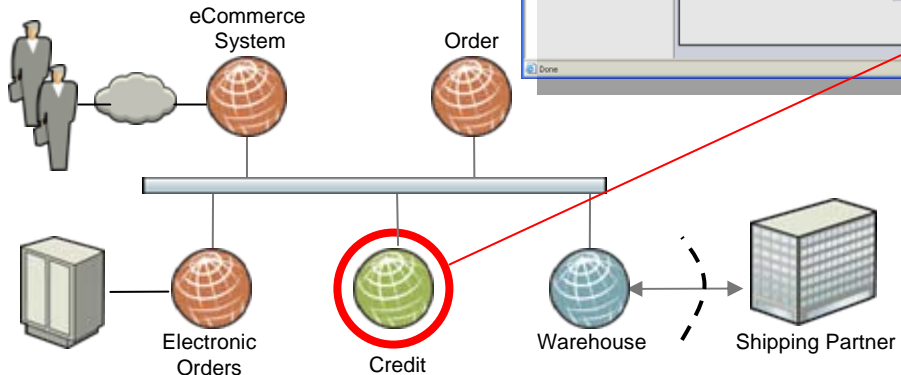
Automatic Transaction Tracking and Diagnosis

Business Exceptions

Process Flow
 - Exception context
 - Response times

Technical Faults

Drill into Exception Content & Context



The screenshot shows the AmberPoint Exception Manager interface. The top part displays a table of exceptions:

Status	Total
Open	5
Closed	0
Resolved	0
All	5

The detailed view for a 'DebitReject' exception shows the following information:

- Correlation:** OrderTransaction
- Expression:** NOT CreditServiceSOAP.debit.response.debitReturn.successful[false]
- Status:** Open
- Priority:** High
- Captured:** 2006-01-31 18:37:19.870
- ID:** 43

Below the text is a process flow diagram with nodes like 'submit', 'checkAvailability', 'checkCredit', 'placeOrder', 'ship', 'ShippingComplete', and 'debit'. The 'debit' node is circled in red.

This screenshot shows a process flow diagram with a detailed view of an exception. The flow includes steps like 'submit', 'checkAvailability', 'checkCredit', 'placeOrder', 'ship', 'ShippingComplete', and 'debit'. A red circle highlights the 'debit' step, which is linked to the exception details shown in the adjacent screenshot.

Augments System-level View

System Mgmt [Close Icon]

- Machines
- App Servers
- Network

- ◆ Business visibility using exception content and context
 - Credit check failure
 - Alert when “no order confirmation within 3 minutes after completion”
- ◆ Visibility in operational issues – services, transactions, operations, messages
 - SOAP faults, database errors, etc.

● Keys to Successful SOA Runtime Governance

- ◆ Knowing What's Out There and What's Going On...
 - What services, SOA components, and infrastructure are installed and in use
 - Who's using it
 - Is it meeting expectations / obligations
- ◆ Controlling It...
 - Define and enforce runtime policies – make sure proper policies are active
 - Diagnose failures / prevent them
- ◆ Ensuring Integrity...
 - Automatically check for the correctness of the running system
 - Detect and validate changes before they impact users and partners
- ◆ And, do it across the entire infrastructure
- ◆ Do this all as automatically as possible
 - Reduces risks and costs
 - Automation is the single most important thing that makes SOA scaleable

AMBERPOINT

Management & Security: The Road Ahead

● The Road Ahead for SOA Management & Security

- ◆ Standards will likely be slow in ratification and implementation
 - Must be able to “roll with the punches”
 - Look for solutions that can support standards as they are implemented, but doesn’t wait for them...
- ◆ Most attempts to form a “single system of record” will fail except for the most simple definitions
 - Federated approaches will dominate

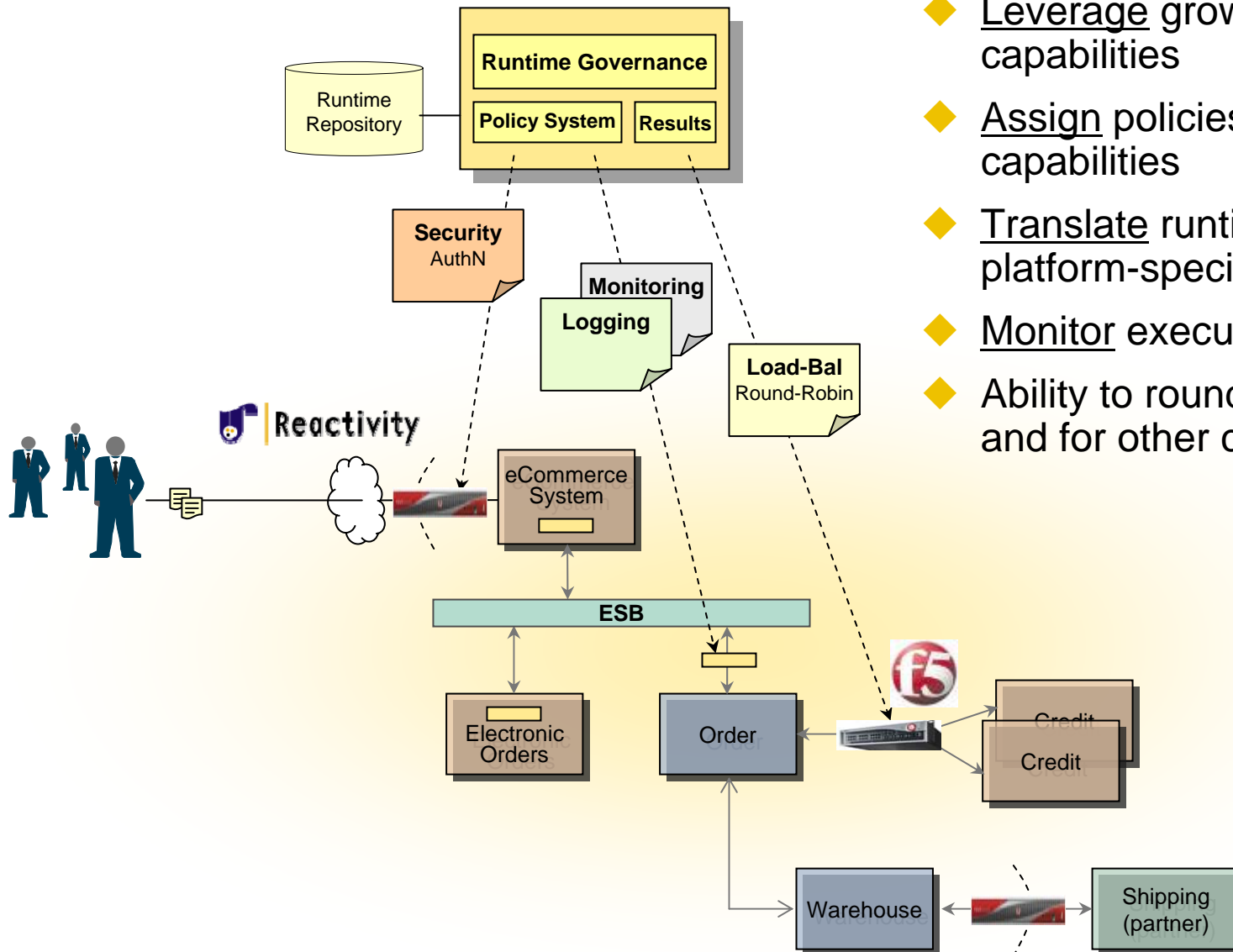
● Agentless Policy Execution

Directly Leverage New, More Capable SOA Infrastructure

- ◆ Increasing ability to execute XML and management requests
 - Appliances – F5, Reactivity, DataPower
 - App Servers – BEA WebLogic, IBM WebSphere
 - ESBs – BEA AquaLogic, IBM WebSphere ESB, IONA Artix, Microsoft BizTalk
 - Operating Systems – Microsoft Windows Communication Foundation (aka “Indigo”)
- ◆ Leverage intrinsic and increasing SOA capabilities of various components wherever possible
- ◆ Look for system that decouples policy definition from policy enforcement, but must have automatic provisioning of enforcement...
 - Enforce runtime governance without installing an agent

Consistent, comprehensive policy enforcement
regardless of platform decisions

Capability-based Delegation based on Platform Capabilities



- ◆ Leverage growing platform capabilities
- ◆ Assign policies based on capabilities
- ◆ Translate runtime policy into platform-specific interfaces
- ◆ Monitor execution
- ◆ Ability to round out capabilities and for other components