

# Mission Thread Management for Federated SOA Systems

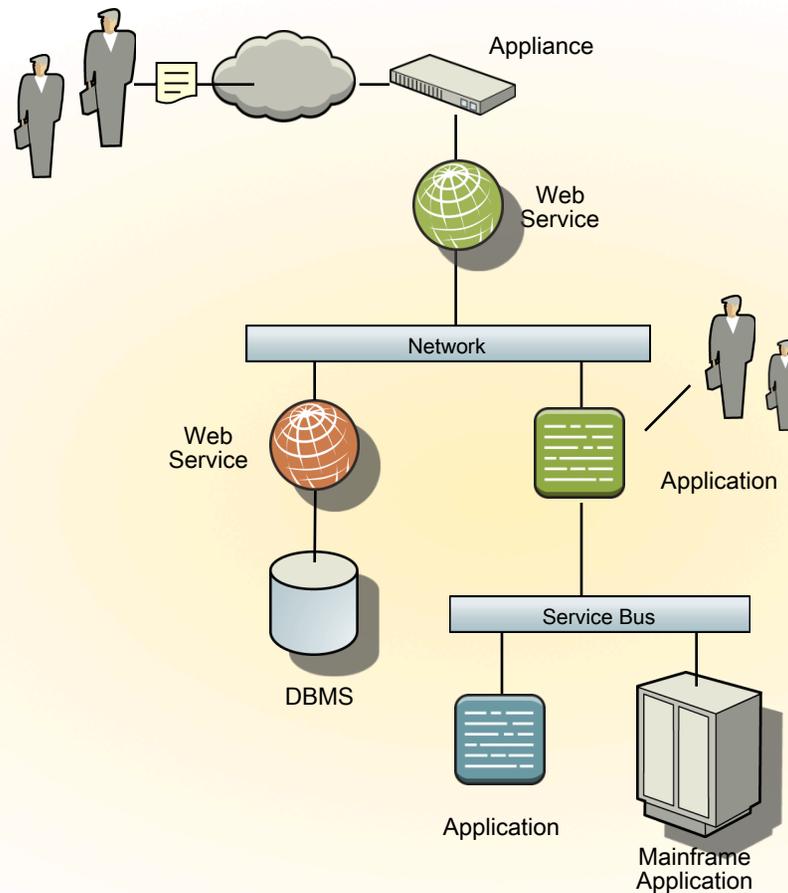
---

Paul Butterworth  
CTO  
AmberPoint, Inc.

## ● Topics

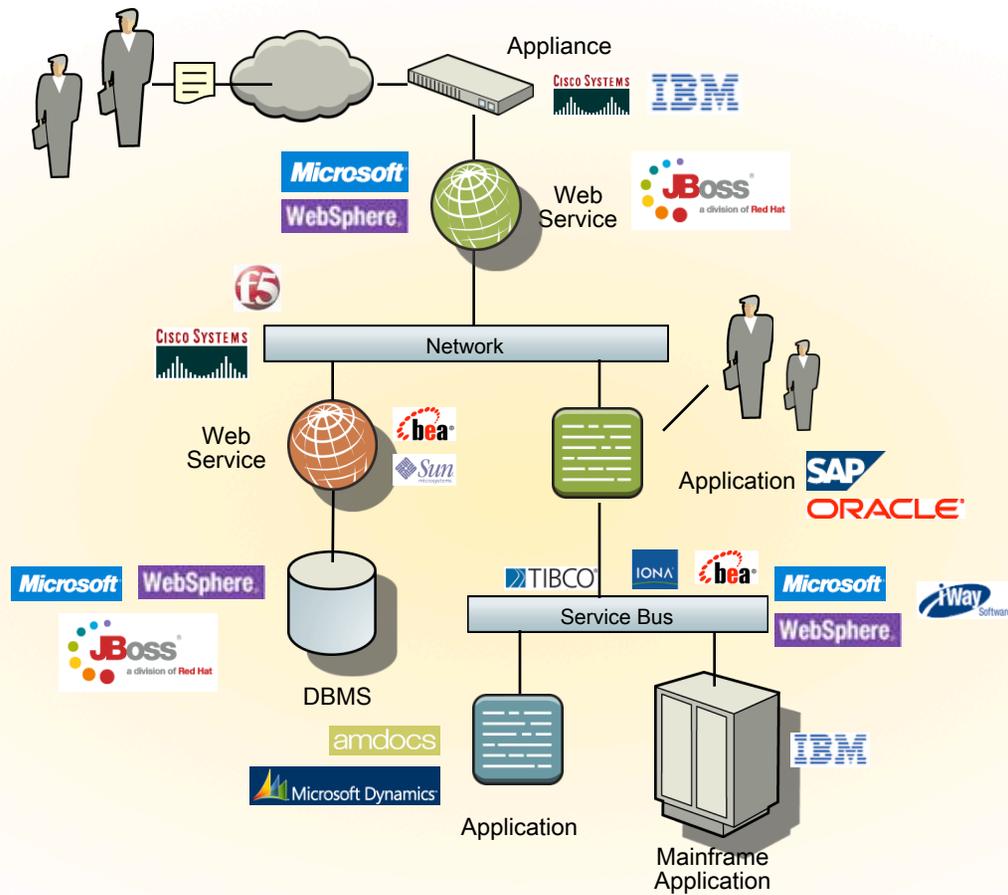
- ◆ SOA Topology
- ◆ Enterprise Service Management
- ◆ Mission Threads
- ◆ Mission Thread Management
  - Requirements
  - Capabilities
  - Benefits
- ◆ Q&A

# ● Variety of Infrastructure and Legacy Systems

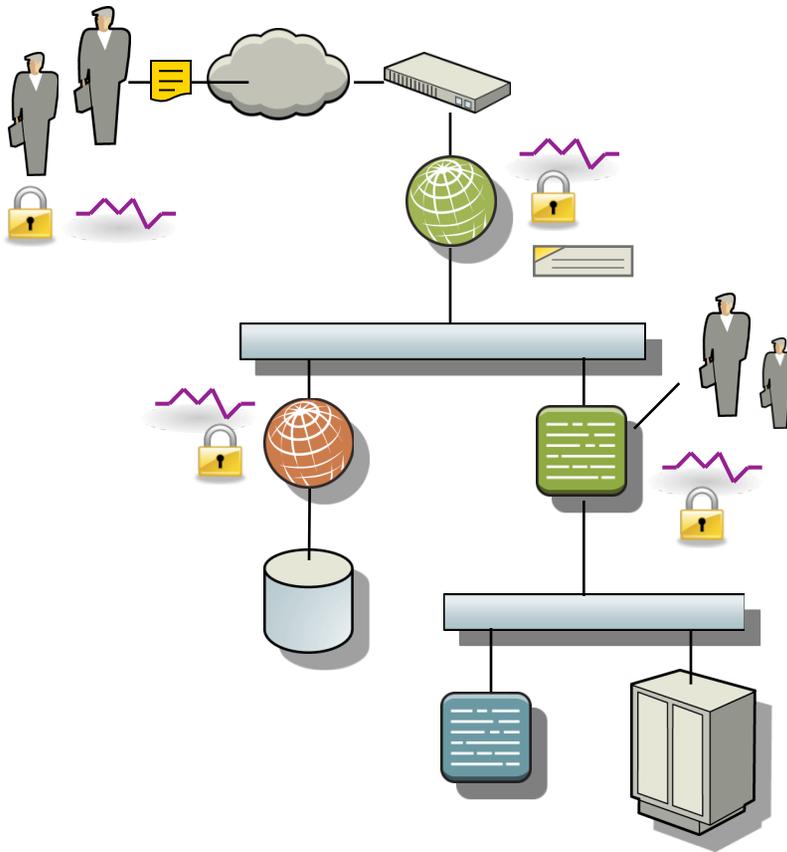


In all but the newest of environments, “SOA” ≠ “Just Web Services & XML”

# ● Variety of Infrastructure and Legacy Systems

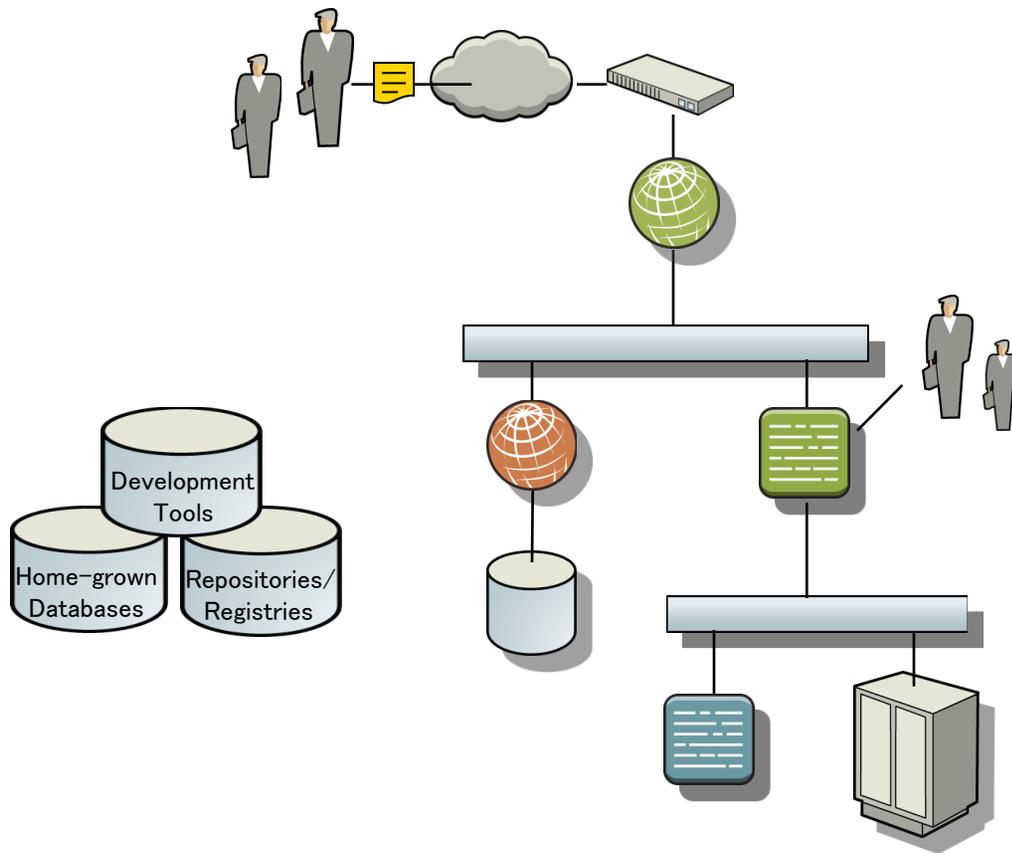


# ● Keys to Successful Governance and Management



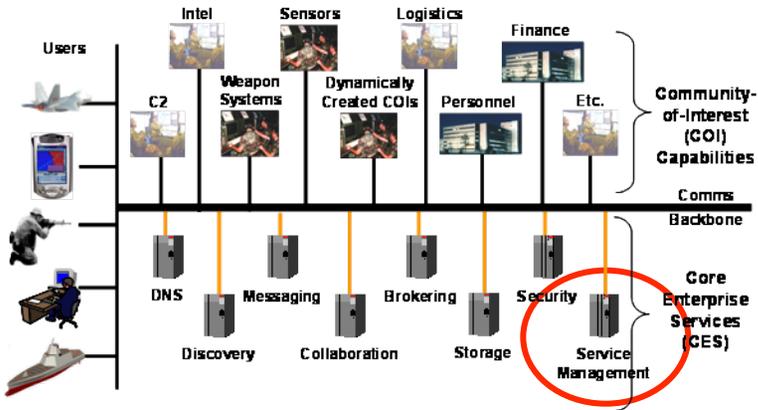
- ◆ Continuous Discovery
- ◆ Service Management & Security

# ● Keys to Successful Governance and Management

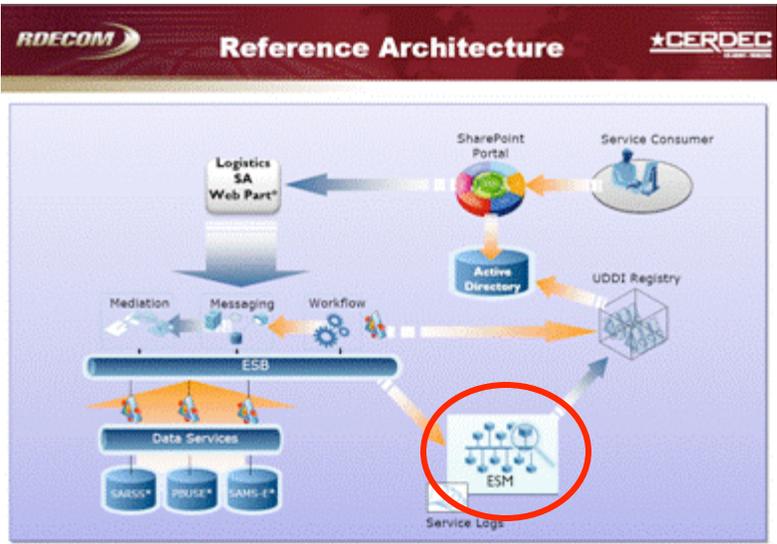


- ◆ Continuous Discovery
- ◆ Service Management & Security
- ◆ Mission-Thread Management
- ◆ Closed-Loop Governance

# ESM – Core Component of SOA Foundation Architectures throughout DoD and Intel



## MCEITS RFP



## ● Example Missions

- ◆ Warfighter
  - Cursor on Target
  - Missing Marine
  - Chemical Plume Analysis and Response
- ◆ Intel – Situational Awareness (i.e., current status of the environment)
- ◆ Air – Air Tasking Order
- ◆ Homeland Security/First Responder
  - Secure the Border
  - Maritime Interdiction
  - Disaster Response

*Complex Processes with Lives on the Line -  
Will Require LSIs and Domain Experts*

# ● The Complex Nature of Mission Threads...

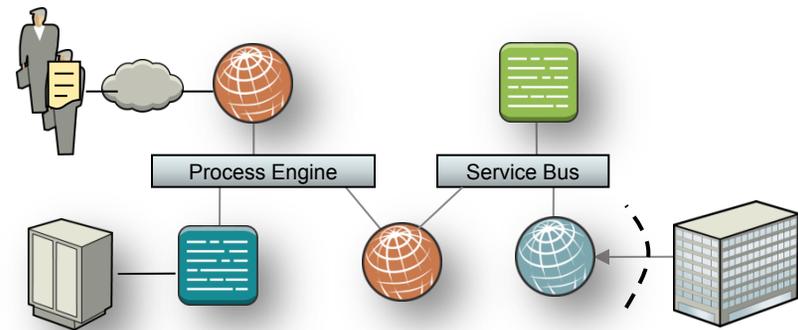
## ◆ Federated SOA

- Multiple platforms
- Variety of communication methods
- Includes components you don't own or control
- Includes integrations of existing systems

## ◆ Constantly evolving

## ◆ As a result, Mission Threads often break without warning

## ◆ Failures are difficult to detect, diagnose and repair



## ● Lack of Mission Thread-level Visibility Results in Resource Intensive Fire-drills

- ◆ You don't know what to look for
  - Minor, data-oriented anomalies are often the root cause
    - Typos, missing data, wrong codes etc.
- ◆ Application users cannot provide sufficient context
  - “The command was sent at 10:00 AM”
- ◆ Service logs are scattered across the environment
  - Unclear which log to examine first
  - May or may not contain the information you need
- ◆ Some mission thread paths might not be within your control
  - Thread instances may flow through appliances, 3<sup>rd</sup> party applications or legacy systems
- ◆ Need a safety-net for trapping mission thread fallout from unforeseen events

# ● Look Familiar?

## Server Error in '/HARMANCENTER/' Application

Cannot have multiple items selected in a DropDownList.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Web.HttpException: Cannot have multiple items selected in a DropDownList.

Source Error:

The source code that generated this unhandled exception can only be viewed when debugging is turned on in debug mode. To enable this, please follow one of the below steps, then request the page again.

1. Add a "Debug=true" directive at the top of the file that generated the error.

Example:

```
<! Page Language="C#" Debug="true" %>
```

or:

- 2) Add the following section to the configuration file of your application:

```
<configuration>  
  <system.web>  
    <compilation debug="true" />  
  </system.web>  
</configuration>
```

Note that this second technique will cause all files within a given application to be compiled in debug mode. The first technique will cause only that particular file to be compiled in debug mode.

Important: Running applications in debug mode does incur a memory/performance overhead. You should make sure that an application has debugging disabled before deploying into production scenarios.

Stack Trace:

```
[HttpException (0x80004005): Cannot have multiple items selected in a DropDownList.]  
System.Web.UI.WebControls.DropDownList.VerifyMultiSelect() +107  
System.Web.UI.WebControls.ListControl.RenderContents(HtmlTextWriter writer) +182597  
System.Web.UI.WebControls.webControl.Render(HtmlTextWriter writer) +29  
System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter a  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +22  
System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection child  
System.Web.UI.WebControls.webControl.RenderContents(HtmlTextWriter writer) +7  
System.Web.UI.WebControls.webControl.Render(HtmlTextWriter writer) +29  
System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter a  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +22  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer) +22  
ASP.checkout_payment_aspx.__Render__control14(HtmlTextWriter __w, Control parameterC  
System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection child  
System.Web.UI.Control.RenderChildren(HtmlTextWriter writer) +20  
System.Web.UI.Control.Render(HtmlTextWriter writer) +7  
System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter a  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +22  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer) +22  
ASP._include_template_aspx.__RenderForm1(HtmlTextWriter __w, Control parameterCont  
System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection child  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +37  
System.Web.UI.HtmlControls.HtmlForm.RenderChildren(HtmlTextWriter writer) +59  
System.Web.UI.HtmlControls.HtmlForm.Render(HtmlTextWriter writer) +68  
System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter a  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +121  
System.Web.UI.HtmlControls.HtmlForm.Render(HtmlTextWriter writer) +37  
ASP._include_template_aspx.__Render__control16(HtmlTextWriter __w, Control parameterContainer) +760  
System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection children) +98  
System.Web.UI.Control.RenderChildren(HtmlTextWriter writer) +20  
System.Web.UI.Control.Render(HtmlTextWriter writer) +7  
System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter adapter) +25  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +121  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer) +22  
System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection children) +199  
System.Web.UI.Control.RenderChildren(HtmlTextWriter writer) +20  
System.Web.UI.Page.Render(HtmlTextWriter writer) +26  
System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter adapter) +25  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter) +121  
System.Web.UI.Control.RenderControl(HtmlTextWriter writer) +22  
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +2558
```

Version Information: Microsoft .NET Framework Version:2.0.50727.1433; ASP.NET Version:2.0.50727.1433

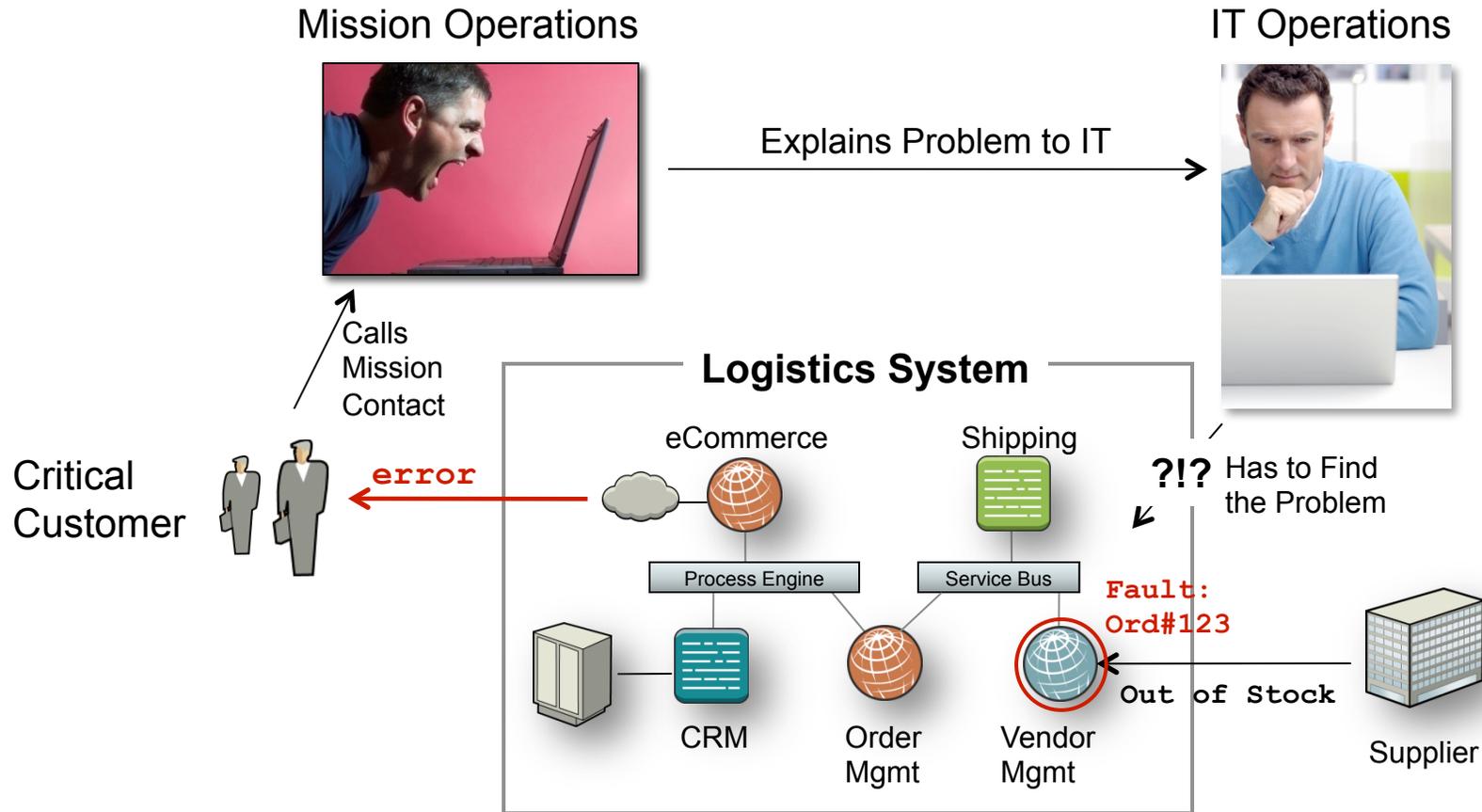
## Server Error

An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information...

## Stack Trace

```
System.Web.UI.Control.RenderControl(HtmlT  
extWriter writer) +22  
System.Web.UI.Control.RenderChildrenInter  
nal(HtmlTextWriter writer, ICollection  
children) +199  
System.Web.UI.Control.RenderChildren(Html  
TextWriter writer) +20
```

# Tracking Mission Threads in Distributed Environments

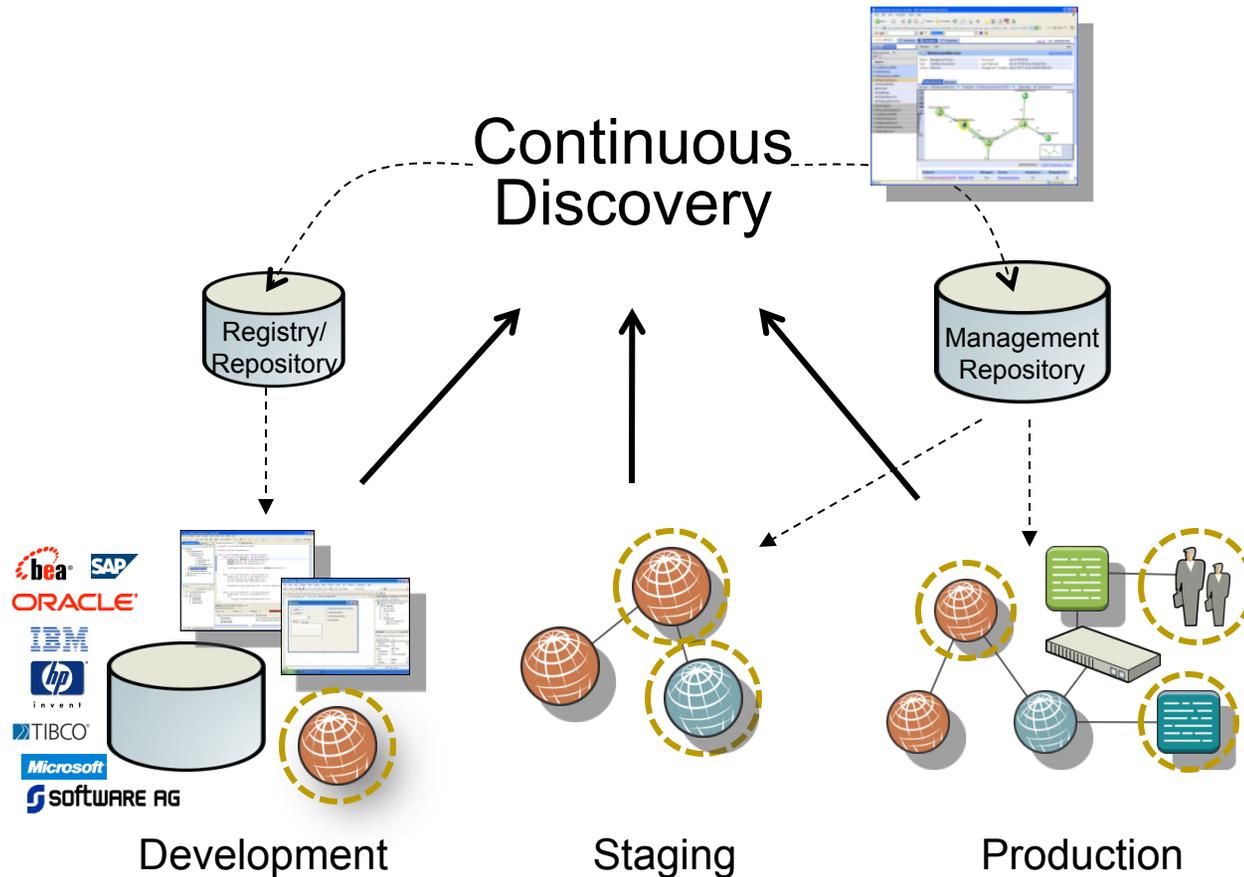


Most fire drills and finger pointing are rooted in Mission Thread failures.

Must ensure that the user experience matches agency objectives.

# Continuous Discovery

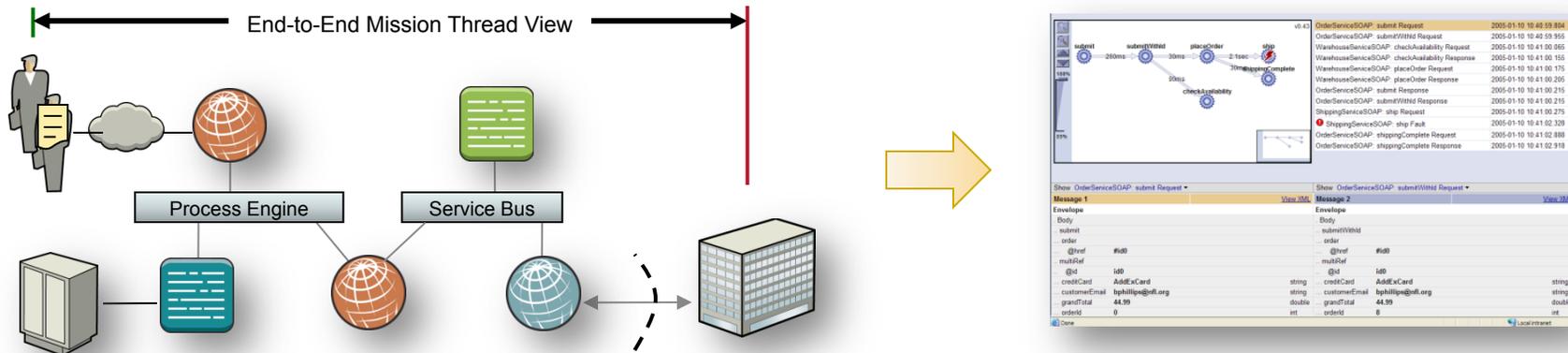
## Ensures a Complete Accounting of Your Applications



- ◆ **Discovers:**
  - Services
  - Dependencies
  - Transaction Flow
  - Consumers
- ◆ **Across:**
  - Containers
  - ESBs
  - Process Engines
  - Packaged Applications
  - Appliances
  - Registries / Repositories
- ◆ **No application coding or message/header modifications**
- ◆ **Benefits**
  - Quick start to projects
  - Closes loopholes in governance
  - Eliminates manual data entry



# Mission Thread Management



- ◆ Real-time tracking of each mission thread instance
- ◆ Follows mission thread instances across all infrastructure and applications, not just inside of a single app server
  - App servers, applications, ESBs, BPMs, etc.
  - Synchronous and asynchronous
  - Long running transactions – hours, days ...
- ◆ Proactively monitors for SLA violations and breakages
- ◆ Alerts when mission threads are out-of-order

# Mission Thread Management

## Real-time Visibility

Mission-Specific Instrumentation

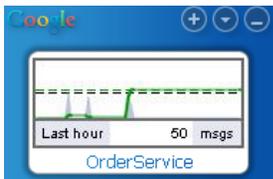
Transaction Summary (Today)		
Correlation	Successful	Unsuccessful
FulfillmentTransaction	18	3
InventoryTransaction	10	1
OrderTransaction	21	6

Bookmark Revenue by Time Period		
Time Period	Revenue	
Today	\$4,073.89	
Last Hour	\$4,073.89	
Last 10 Min	\$134.97	
Last 1 Min	\$44.99	

Bookmark Revenue by Account		
Account	Total Orders	Total Revenue
Amazon.com	2	\$359.90
Barnes & Noble	1	\$279.92
Books-A-Million	2	\$349.90
Borders	1	\$44.99
Half Price Books	20	\$2,679.28
Walmart	2	\$359.90



Monitor summary from your desktop

Exception Summary		Total
Open		17
Closed		0
Resolved		0

Most Frequent Exceptions		Total
Condition		
OrderTransaction.SecurityException	4	
InventoryTransaction.LowInventory	3	
OrderTransaction.OutOfStock	3	
FulfillmentTransaction.Fault	2	
FulfillmentTransaction.DebitReject	2	
OrderTransaction.DebitCardUsed	2	
FulfillmentTransaction.LostShipment	1	

5 Most Recent Open Exceptions for OrderTransaction			
	Condition Name	Correlation Name	Capture Time
!	SecurityException	OrderTransaction	Aug 20 20:36:54
!	SecurityException	OrderTransaction	Aug 20 20:36:53
♦	OutOfStock	OrderTransaction	Aug 20 20:31:04
●	DebitCardUsed	OrderTransaction	Aug 20 20:25:18
♦	OutOfStock	OrderTransaction	Aug 20 20:24:46

Transaction Exceptions

Detail for Exceptions

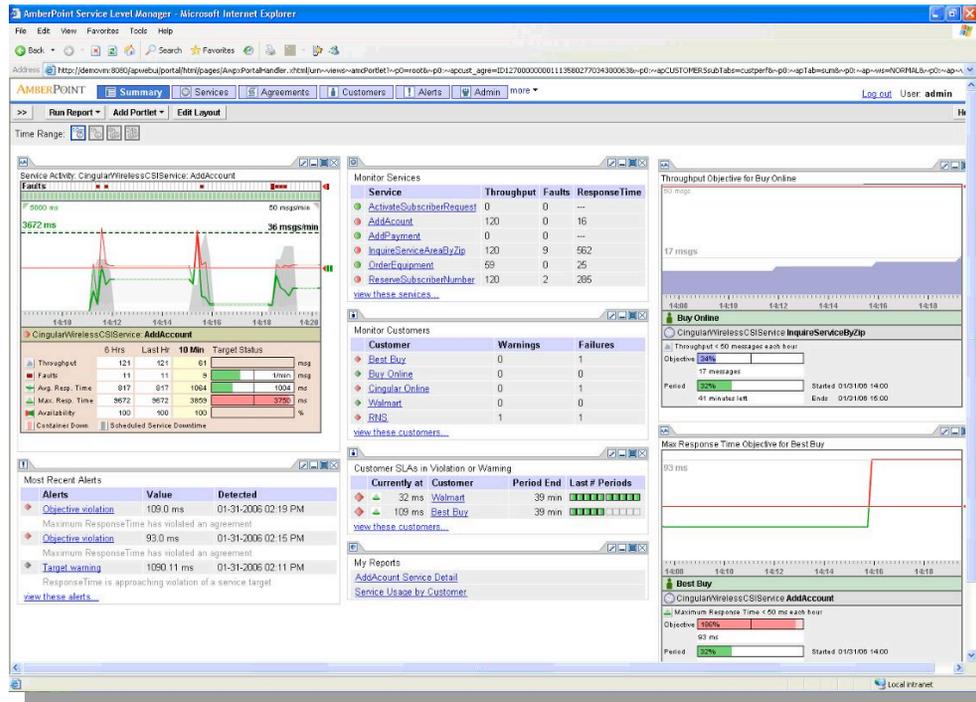
- ◆ Track mission-centric value flowing through the system
  - Track logistics, total orders, etc.

# Mission Thread Management

## Meeting SLAs for Responsiveness, Availability & Throughput

Transaction Level Agreements

Violations

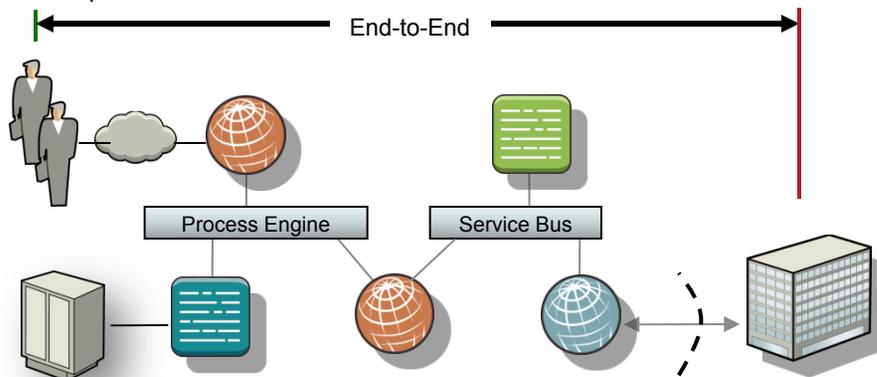


Various Groups

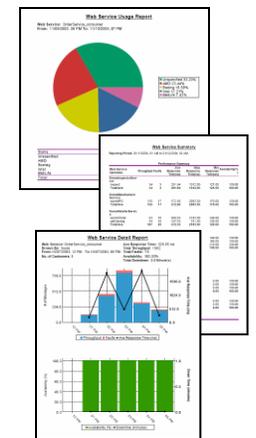
- Platinum, Gold, etc.
- Different Departments or Agencies, etc.

Consumer SLAs

Historical Reporting



- Enforces agreements in real time
- Enables preventative and corrective actions
  - Not just reporting violations after its too late





# Mission Thread Management

## Ops/Development: Rapid Problem Isolation

Transaction History

Message Flow - Per Transaction

The screenshot shows the AmberPoint Exception Manager interface. At the top, there's a navigation bar with 'Summary', 'Exceptions', 'Correlations', and 'Services'. Below that is a table of transactions:

Status	ID	Condition Name	Correlation Name	Property Values	Capture Time
Open	97	OrderService s...	OrderServiceSOAP	grandTotal=44.9900016784668, custome...	Jan 31 06:45:43
Open	77	LostShipment	OrderTransaction	grandTotal=44.9900016784668, cardType...	Jan 31 06:43:26
Open	88	ShippingServic...	ShippingServiceSOAP		Jan 31 06:43:26
Open	43	DebitReject	OrderTransaction	orderId=324, grandTotal=44.9900016784...	Jan 31 06:37:19
Open	8	LostShipment	OrderTransaction	orderId=316, grandTotal=0, cardType=De...	Jan 31 06:32:22

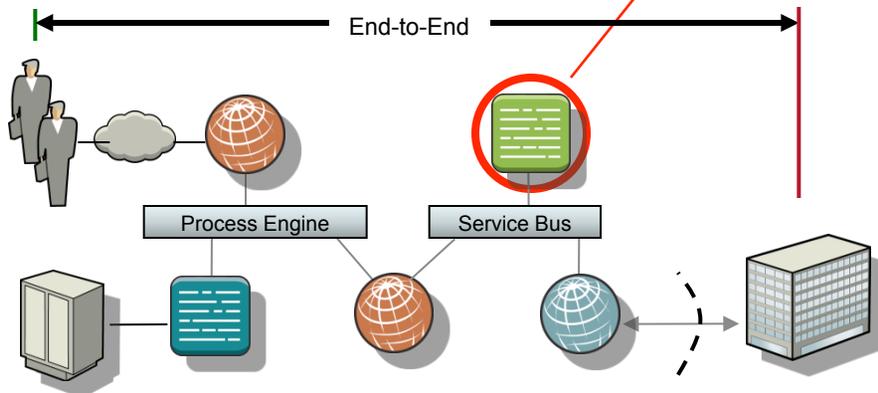
Below the table, a detailed view for the 'DebitReject' transaction is shown. It includes a 'Message Flow' diagram with nodes like 'submit', 'checkAvailability', 'checkCredit', 'placeOrder', 'ship', 'shippingComplete', and 'debit'. The 'debit' node is highlighted with a red circle. A red arrow points from this circle to the 'Service Bus' component in the diagram below.

Drill into Transaction Content & Context

This screenshot shows a detailed view of a transaction message flow. It includes a sequence diagram with nodes like 'submit', 'checkAvailability', 'checkCredit', 'placeOrder', 'ship', 'shippingComplete', and 'debit'. Below the diagram, there are two message envelopes, 'Message 1' and 'Message 2', showing their headers and bodies. The 'debit' node in the diagram is highlighted with a red circle.

Unstructured Search for Specific Transactions

This screenshot shows the search interface for filtering messages. It includes a search bar with the text 'A4315-2' and a dropdown menu set to 'that occurred in the last 1 day'. There are also checkboxes for 'anotherprop', 'orderId', and 'PropertyToTestDuplicateName'. A 'Hide Advanced Text Search' button is visible at the bottom.



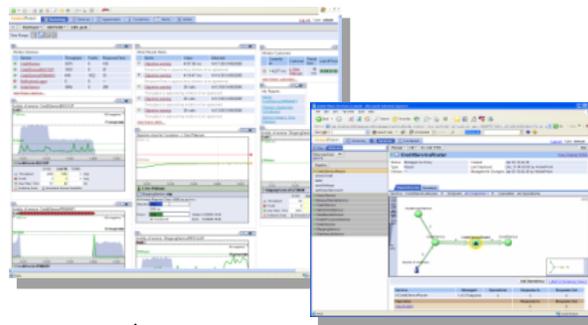
- ◆ Eliminates costly manual search through log files on multiple machines
  - Unified view of transaction even across different keys, IDs, etc.
- ◆ Comprehensive searching
  - Content & context
  - Time range
  - Specific faults

# ● Who Benefits from Mission Thread Management?



## Mission-Focused Users

- Real-time visibility
- Mission thread management



## Architects & Development

- Remedies the brittle nature of SOA
- Identifies "hotspots" within system
- Detailed insight into problems



## Operations

- Ensures SLAs are met
- Rapid problem isolation
- Greater system uptime

# Q&A

---

Paul Butterworth  
CTO  
AmberPoint, Inc.  
[pbutterworth@amberpoint.com](mailto:pbutterworth@amberpoint.com)