

What happened to my Transaction?

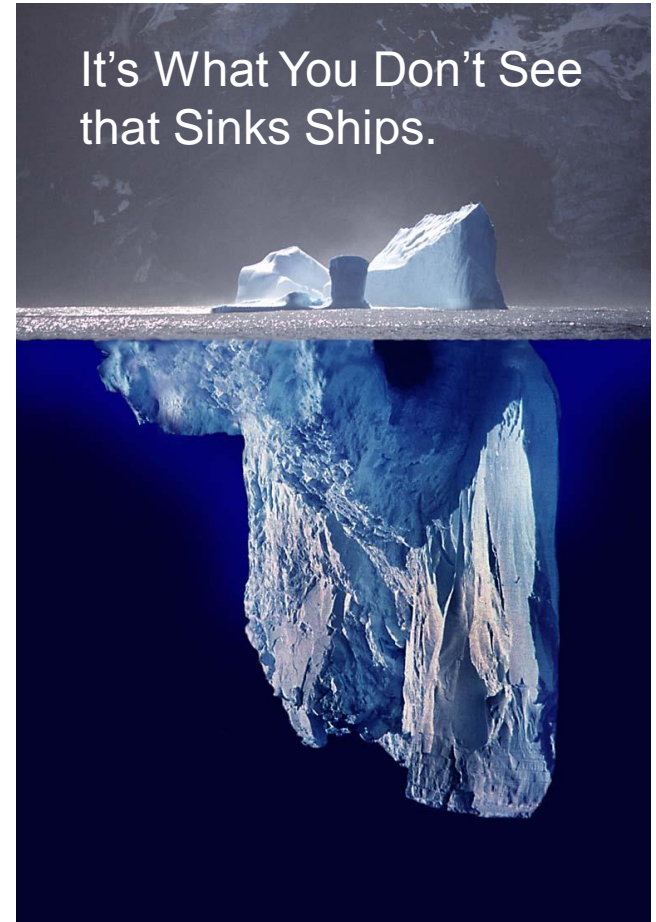
Barry D. Lamkin
Executive IT Specialist
blamkin@us.ibm.com

Transaction Tracking - APM

- Transaction Tracking is a major part of Application Performance Monitoring
- To ensure SLA compliance, monitoring transaction performance is essential for finding and fixing problems before they impact customer satisfaction.
- You need a consolidated view for monitoring resources and activities within your enterprise so performance issues can be detected early and resolved quickly.
- When a problem is detected, is it possible to drill-down to diagnose the root cause of the bottleneck so a solution can be implemented?

Businesses are facing unparalleled challenges

- **Accelerated rate of change** drives the need for increased visibility into the application & IT infrastructure
- **Monitoring IT resources alone** provides an incomplete view of application performance and makes problem isolation and resolution a complex, expensive task
- **Lack of visibility** into end-user experience, component relationships and service levels in **dynamic environments** such as cloud
- **Performance and availability** issues for multi-platform composite applications
- **Lack of drilldown capability** to find the root cause of problems
- **Increased risk** of revenue loss and brand damage

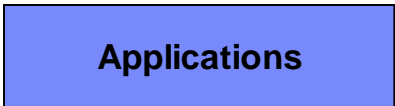
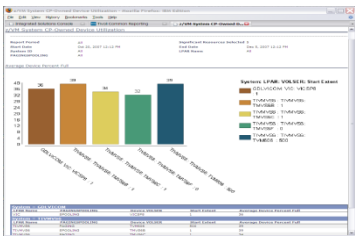
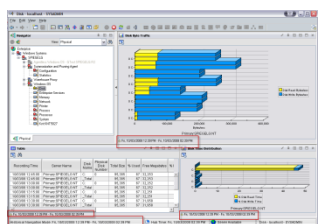
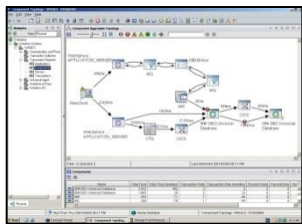


Benefits to Effective Application Performance Management

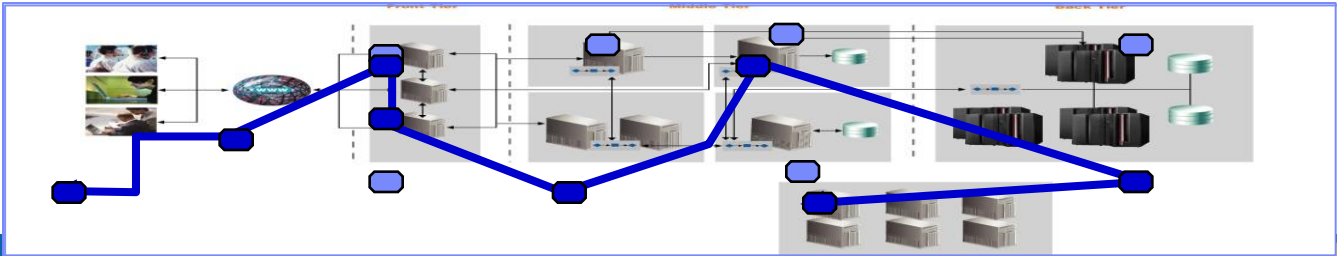
- Ensure application response meets business expectations
- Understand transaction flows over complex topologies
- Drive close collaboration between departments
- Monitor infrastructure performance and availability
- Diagnose application performance issues
- Increase application availability and customer satisfaction
- Improve MTTR and MTBF



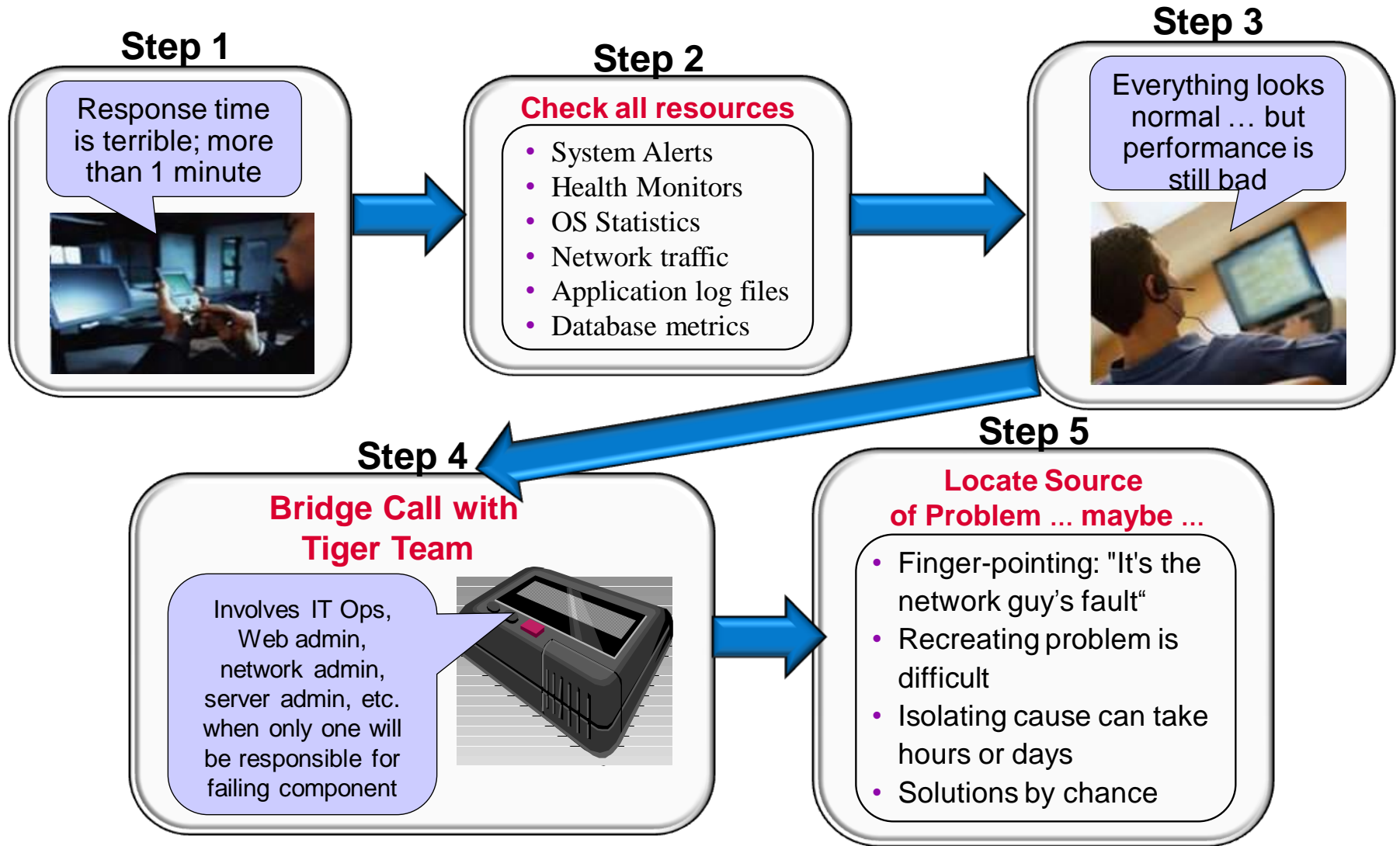
IT Staff



IT Customer



The Issue – Sensing and Isolating a Problem Today



Customer Value – Demonstrating ROI

Money wasted isolating problems

Sev 1 outages/slowdowns per year	12
Average time to isolate (hrs)	8
SME's involved in isolation	15
Avg. loaded hourly rate (/hr)	\$75
Total direct costs	<hr/> \$108,000

Revenue lost during outages

Lost revenue per hour	\$50,000
Hours downtime / yr	96
Total indirect costs	<hr/> \$4,800,000

Total costs of poor problem isolation capability

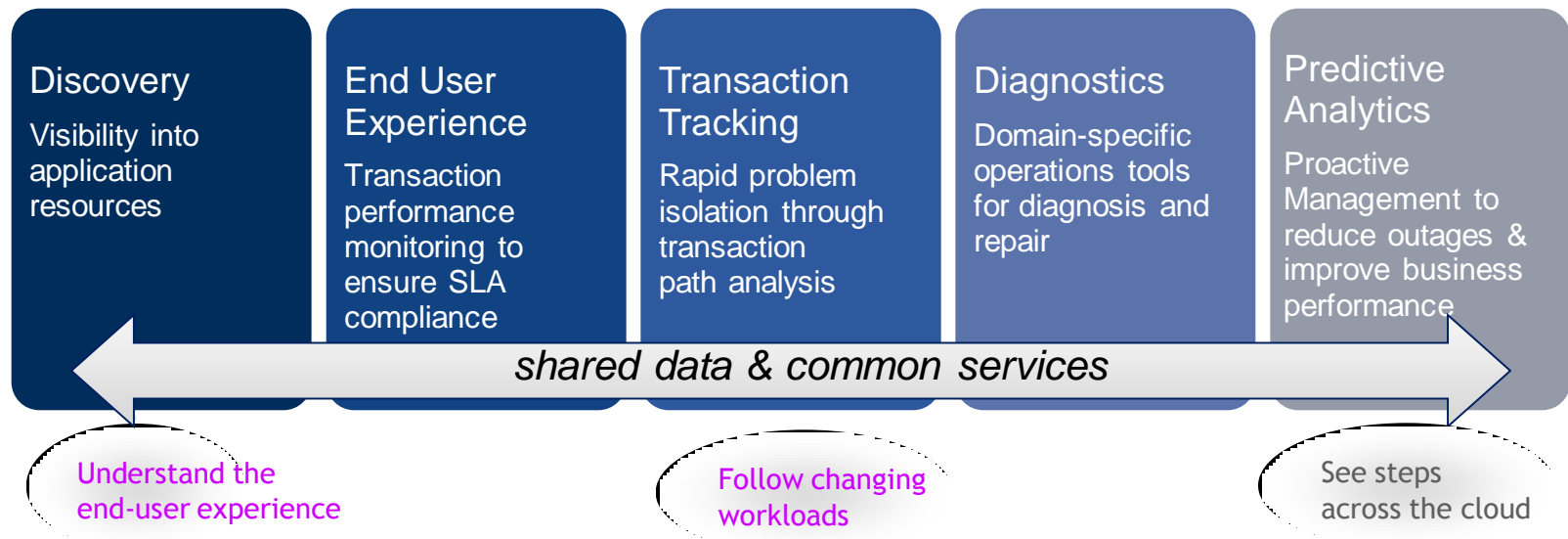
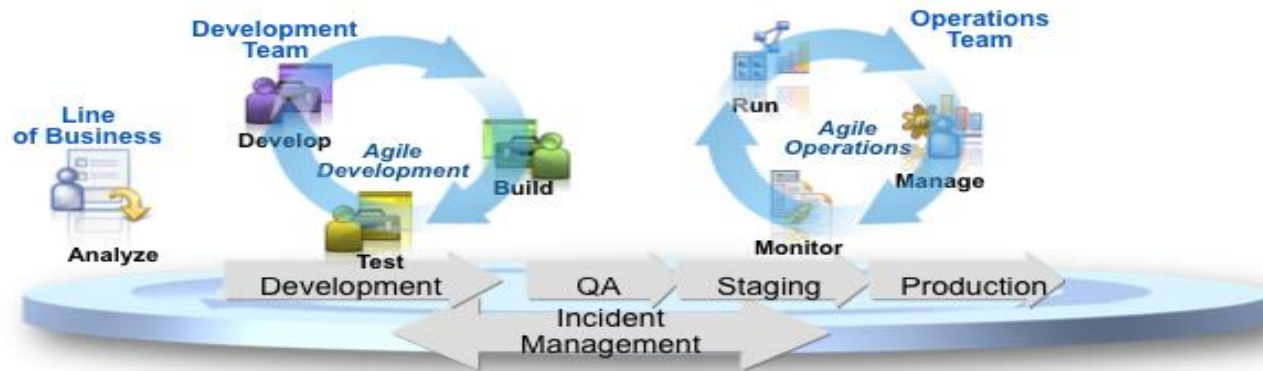
Total lost / yr	<hr/> \$4,908,000
-----------------	-------------------

Every customer case will be different ...

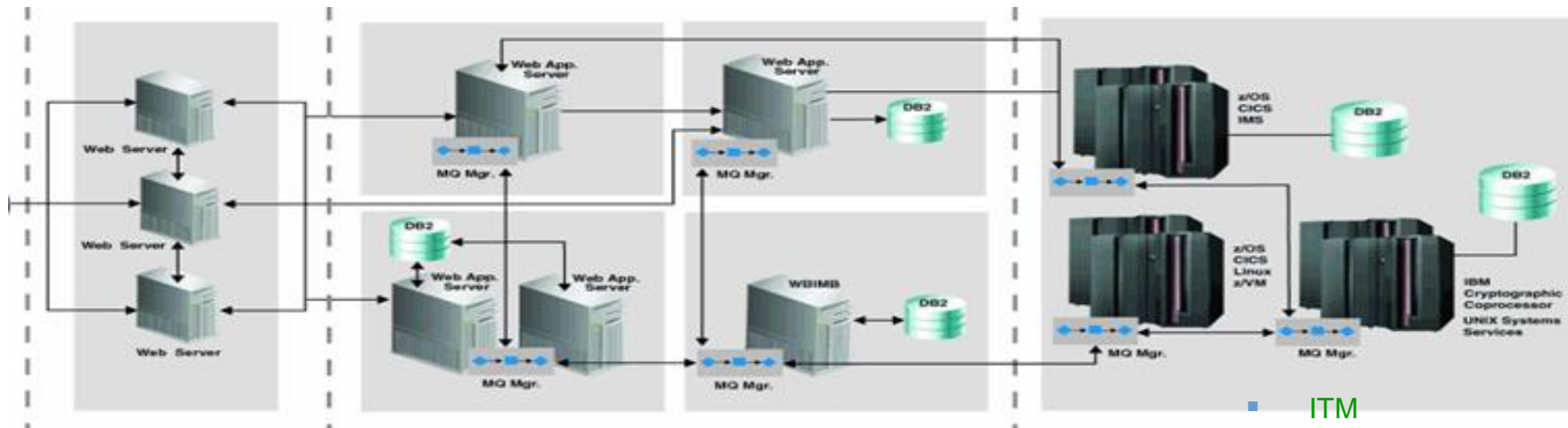
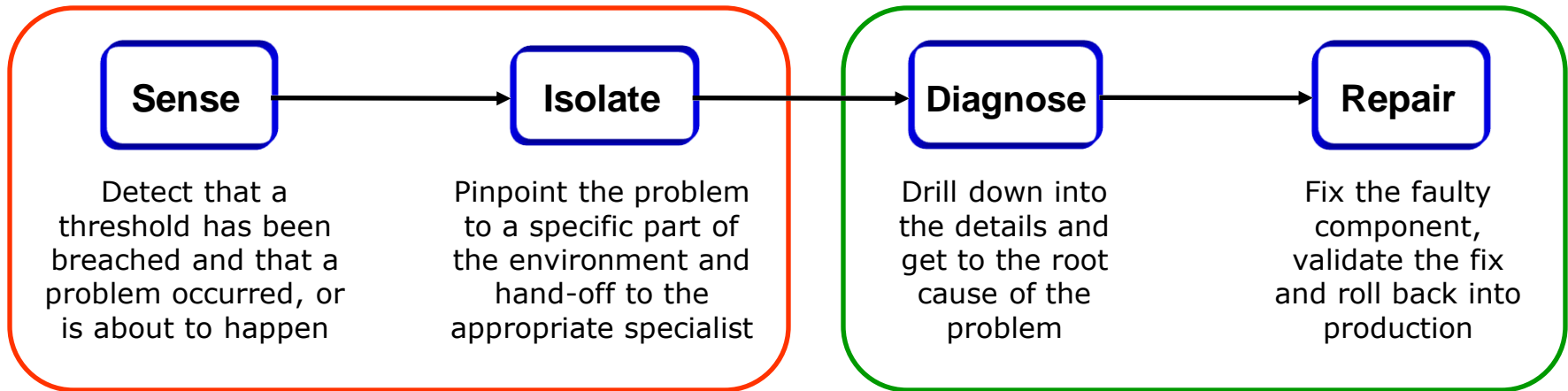
...what do **you** lose each year due to poor performance?

Application Performance Management provides...

Visibility, control and automation to intelligently manage critical applications in cloud and hybrid environments.



Application Performance Management Workflow

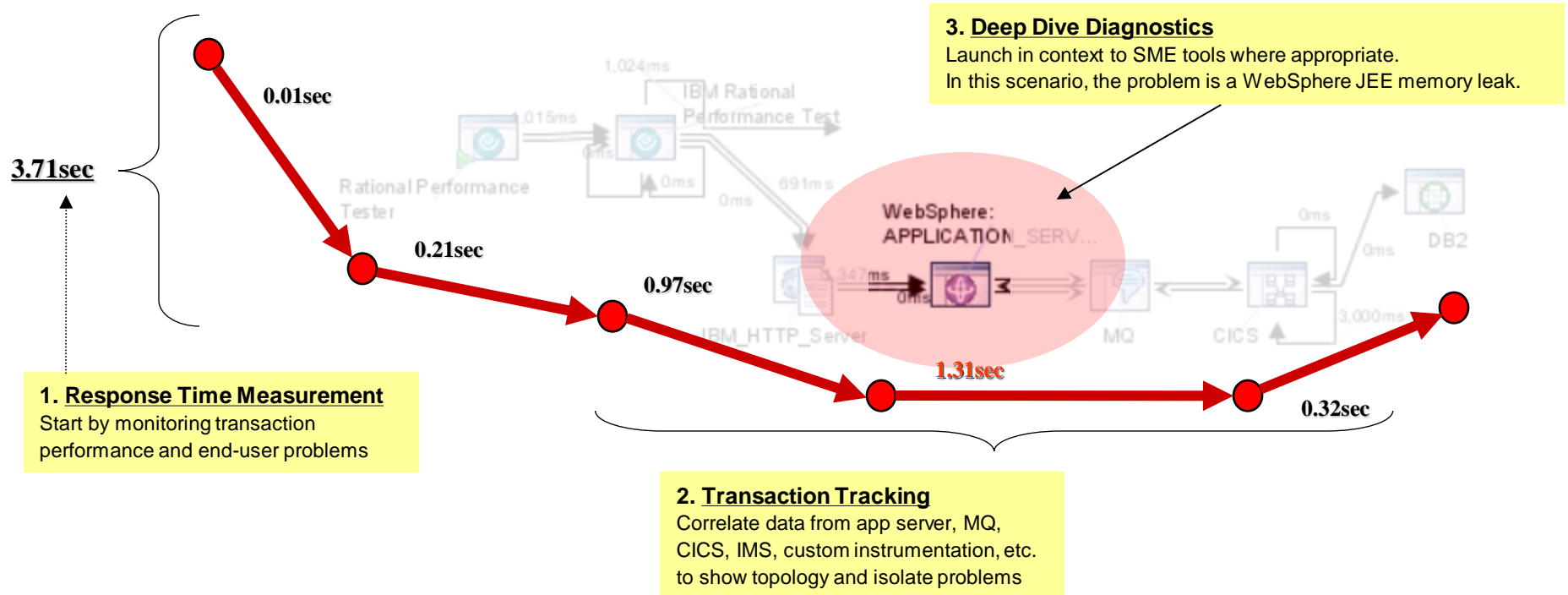


Transaction Tracking

Deep-dive tools

- ITM
- ITCAM for AD
- ITCAM for SOA
- OMEGAMONs

End-to-End Monitoring, Tracking and Diagnosis



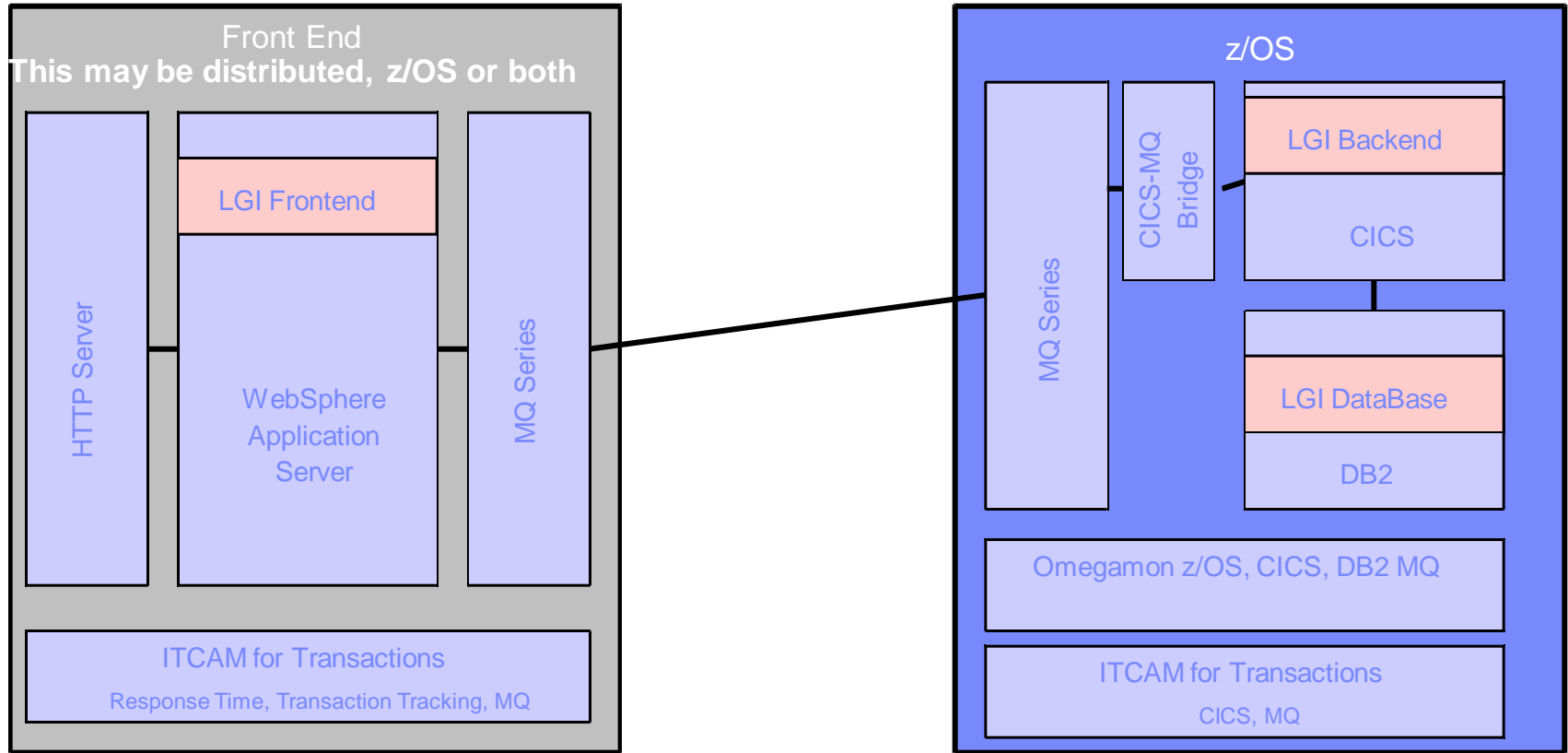
Transaction Root Cause Analysis

1. Sense End User Experience and alert on threshold violation

1. Isolate by measuring performance data against baseline through entire infrastructure

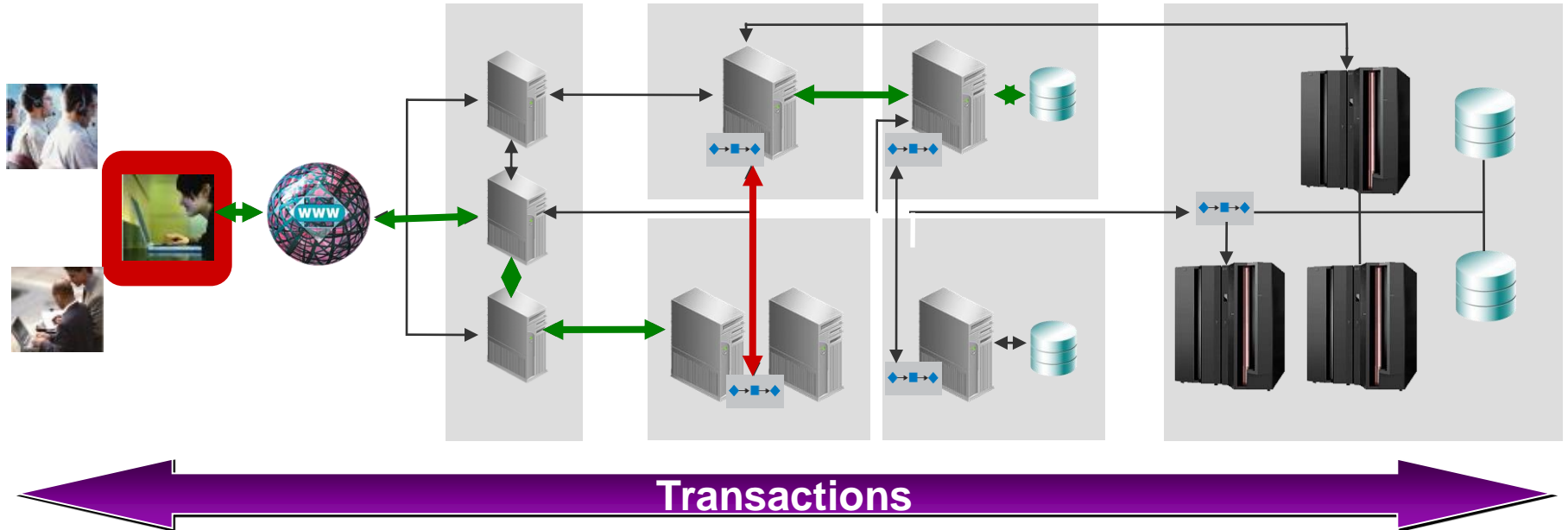
1. Diagnose and repair through launch-in-context into deep-dive diagnostics

End-to-End infrastructure - System z backend



The front-end may be either zLinux, distributed or native z/OS or a mix: customers may deploy any of these configurations.

Why Monitor End-User Response?



- See what your users are experiencing
- Validate production system performance
- Identify problems before they affect SLA's
- **If you have a problem, find out about it before the customers start complaining**

A majority of IT problems are still being **identified by customer complaints**

End User Monitoring

Ensure end user's experience always meets their expectations

- **See what your users are experiencing**
- **Identify problems before they affect SLAs:**
 - Real-User monitoring
 - Robotics monitoring
- **Continuously validate** production system performance
 - Captures performance and availability data of actual users for SLA reporting
- Monitors network traffic for HTTP(S) requests to the web server
 - Completely non-invasive, agentless monitoring
- **If you have a problem, find out** about it **before** the **customers complain**

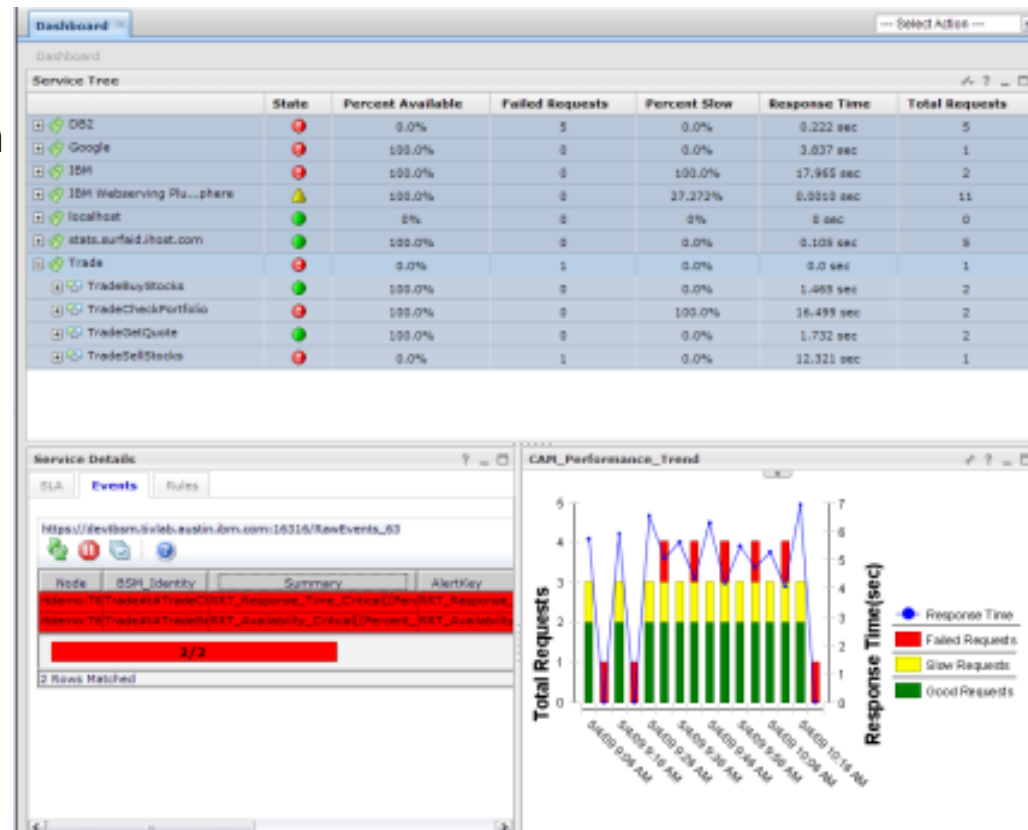


50% of problems found through customer complaints to help desk!

Agentless Real-User (Passive) Monitoring

Monitor every end user's experience

- See what your users are experiencing and immediately identify problems
- Agentless – no impact to production machines
- User/session tracking – observe individual user experience
- Multi-protocol support (beyond HTTP/S)



Robotic Monitoring

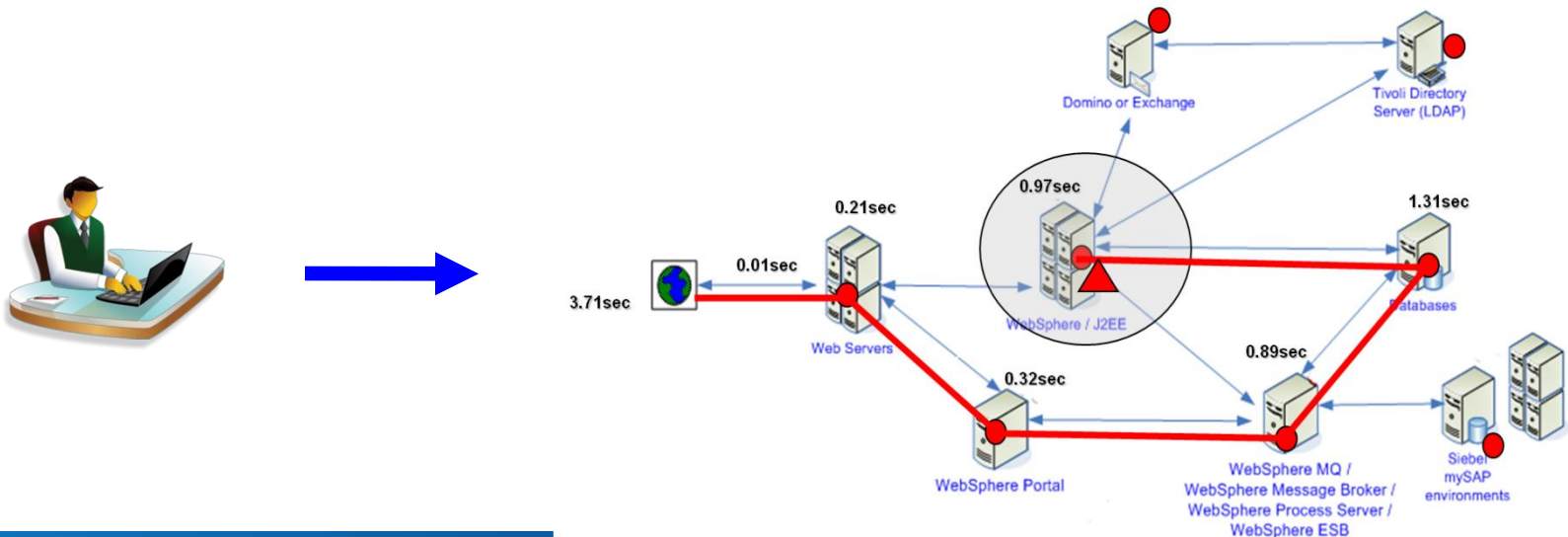
- Verification Points for content matching and response code checking
- HTTP transactions correlate with downstream instrumentation for problem isolation
- Improved scalability for more concurrent playbacks on a single agent
- Support for a growing list of protocols: HTTP(S), Siebel, Citrix, web services



Transaction Tracking

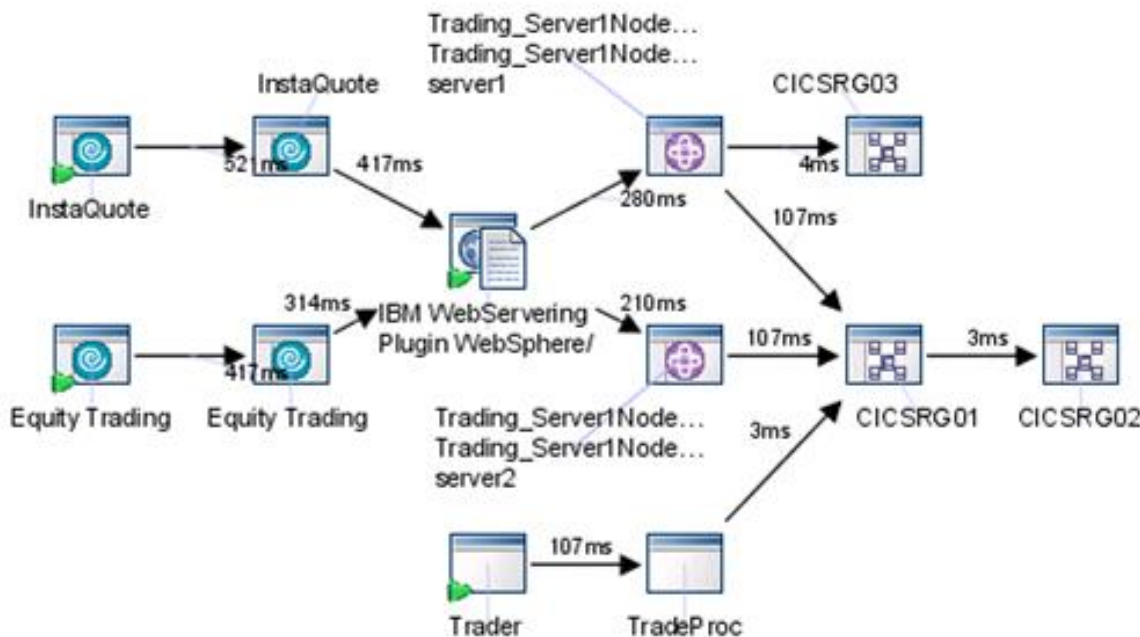
Quickly isolate the failing component in the application

- Follow path of user transactions across application infrastructure domains, making it easier to evaluate a transaction in its entirety
 - Agentless: Track flows through network traffic
 - Agent Based: Detailed, Instance-level Transaction Tracking
- Visibility into how IT infrastructure delivers business critical applications
- End-to-End view of response times across multiple domains helps quickly isolate problems and hand problem off to appropriate specialist



Problem Isolation Through Transaction Tracking

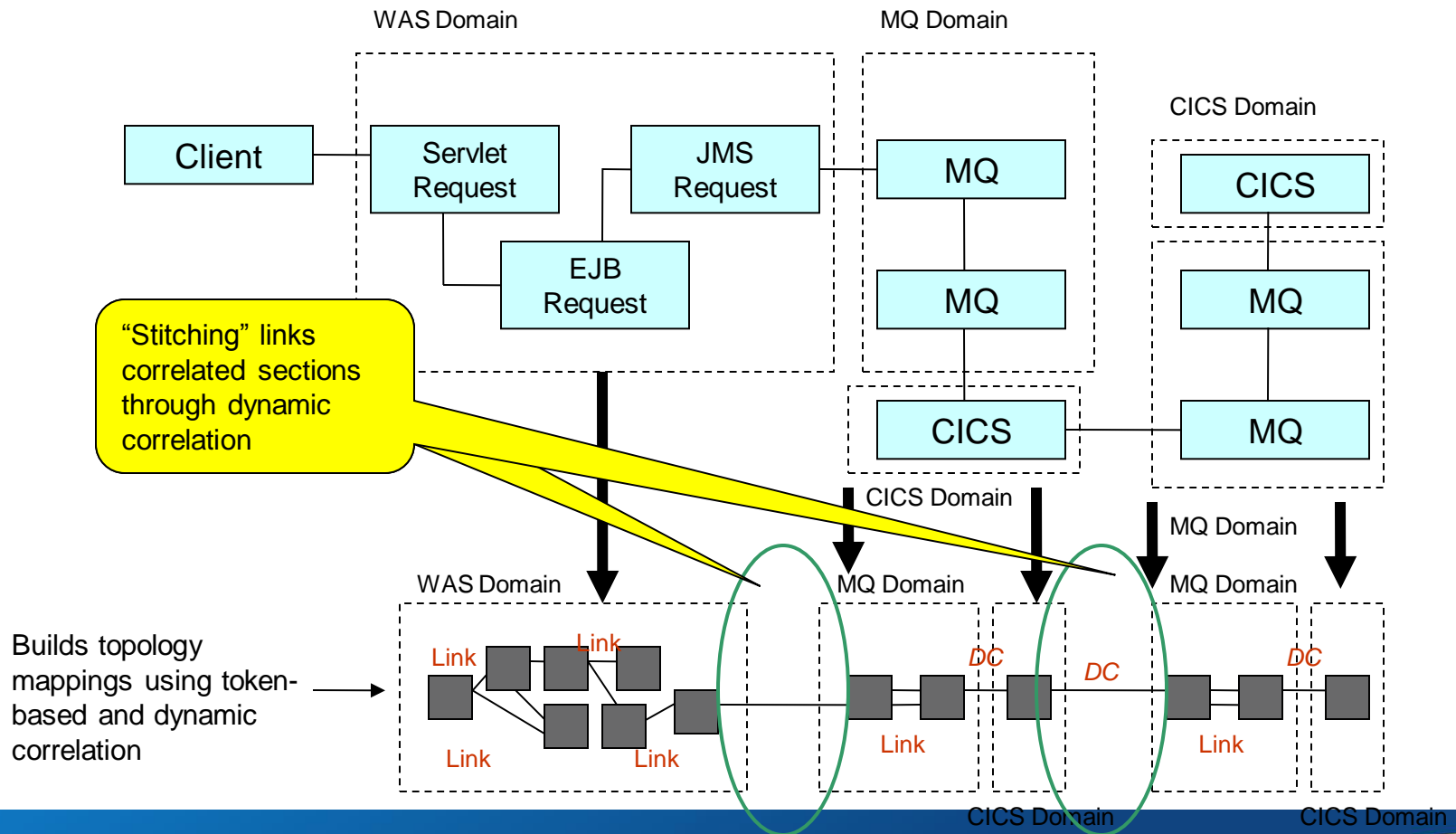
- Unified, end-to-end transaction tracking
- Heterogeneous environments
 - fully integrated across distributed and System z



- Support for asynchronous transactions
- Extensible, modular framework
- Integrated response time and transaction tracking

Enterprise-Wide Tracking

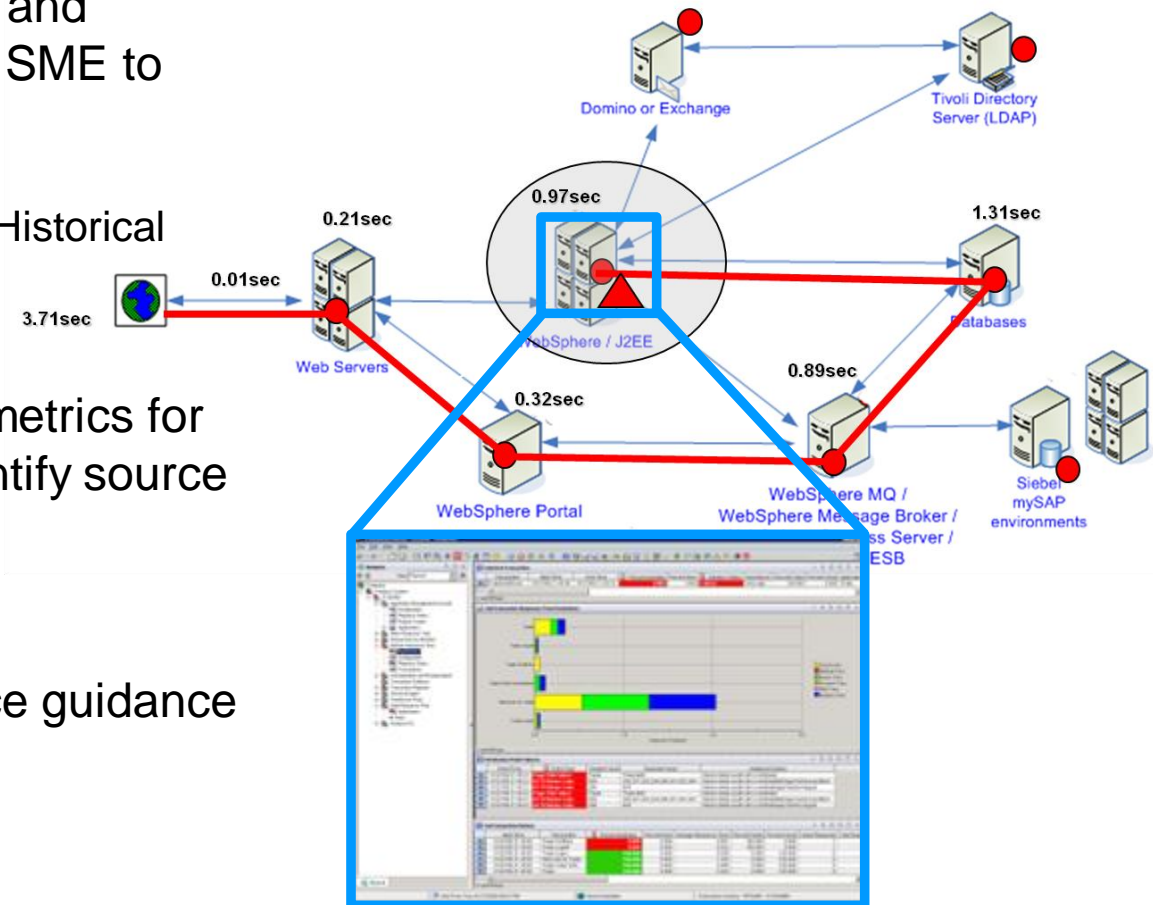
- Track inside domains with correlated techniques
- Track between domains through stitching



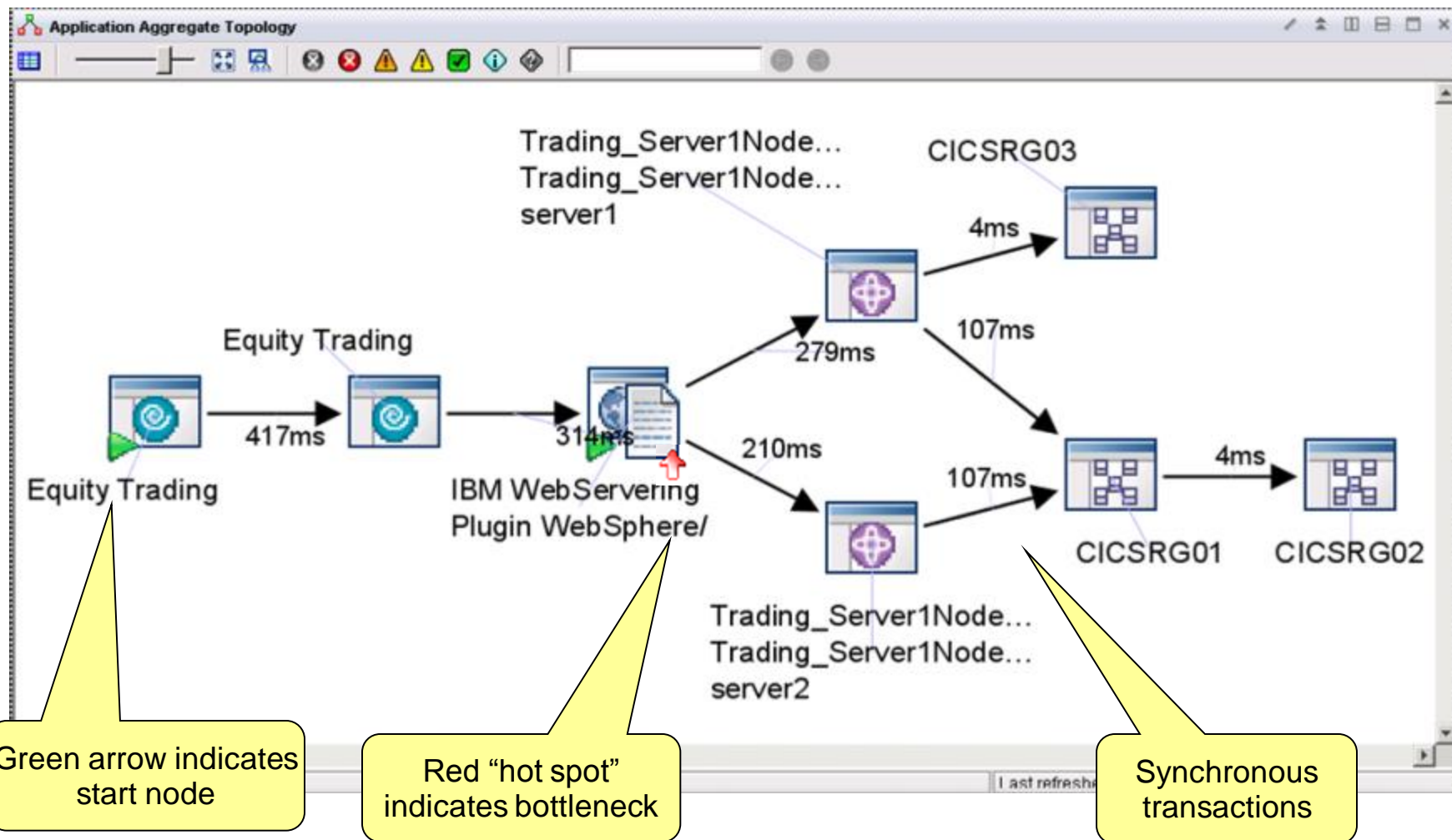
Diagnostics

Quickly identify the source of the failure within the component

- When the failing component has been isolated, detailed performance and availability metrics provided to SME to troubleshoot
 - Access to both Real-time and Historical data within a single UI
- Access to key performance metrics for each resource to quickly identify source of failure
- Expert Help and Best Practice guidance for optimum performance



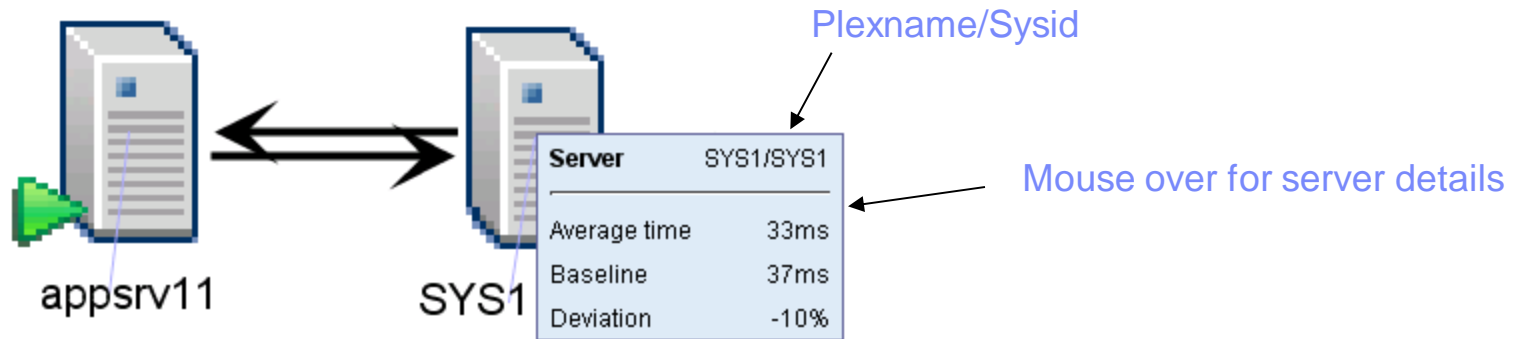
Transaction Tracking Topology



Transaction Tracking – Topology Workspace Views

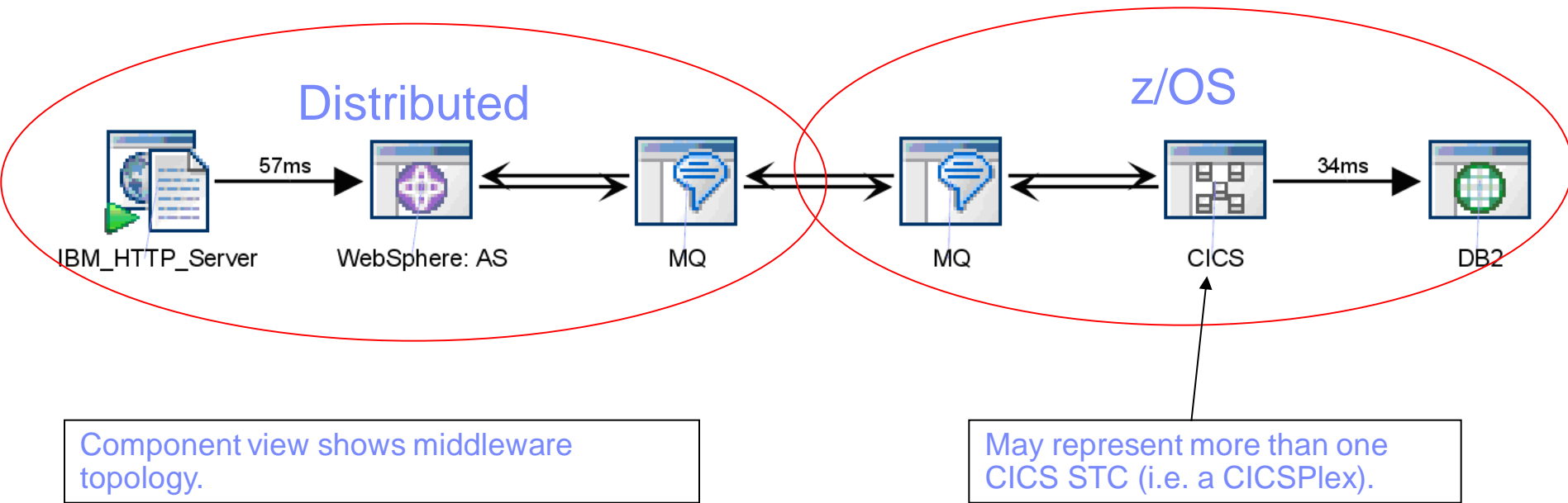
- There are 4 topology workspace views available in the TEP.
 - Server:
 - A topology showing monitored servers (Sysplex name/SMFID or shortname).
 - Component
 - A topology showing the monitored components (Eg: WAS, CTG, CICS, IMS, Connect IMS, WMQ).
 - Application:
 - A topology showing monitored applications (jobnames, STC names, subsystem names).
 - Transactions
 - A topology showing the monitored transactions (Eg: CICS transaction name, IMS transaction name, WAS jsp).

Server topology



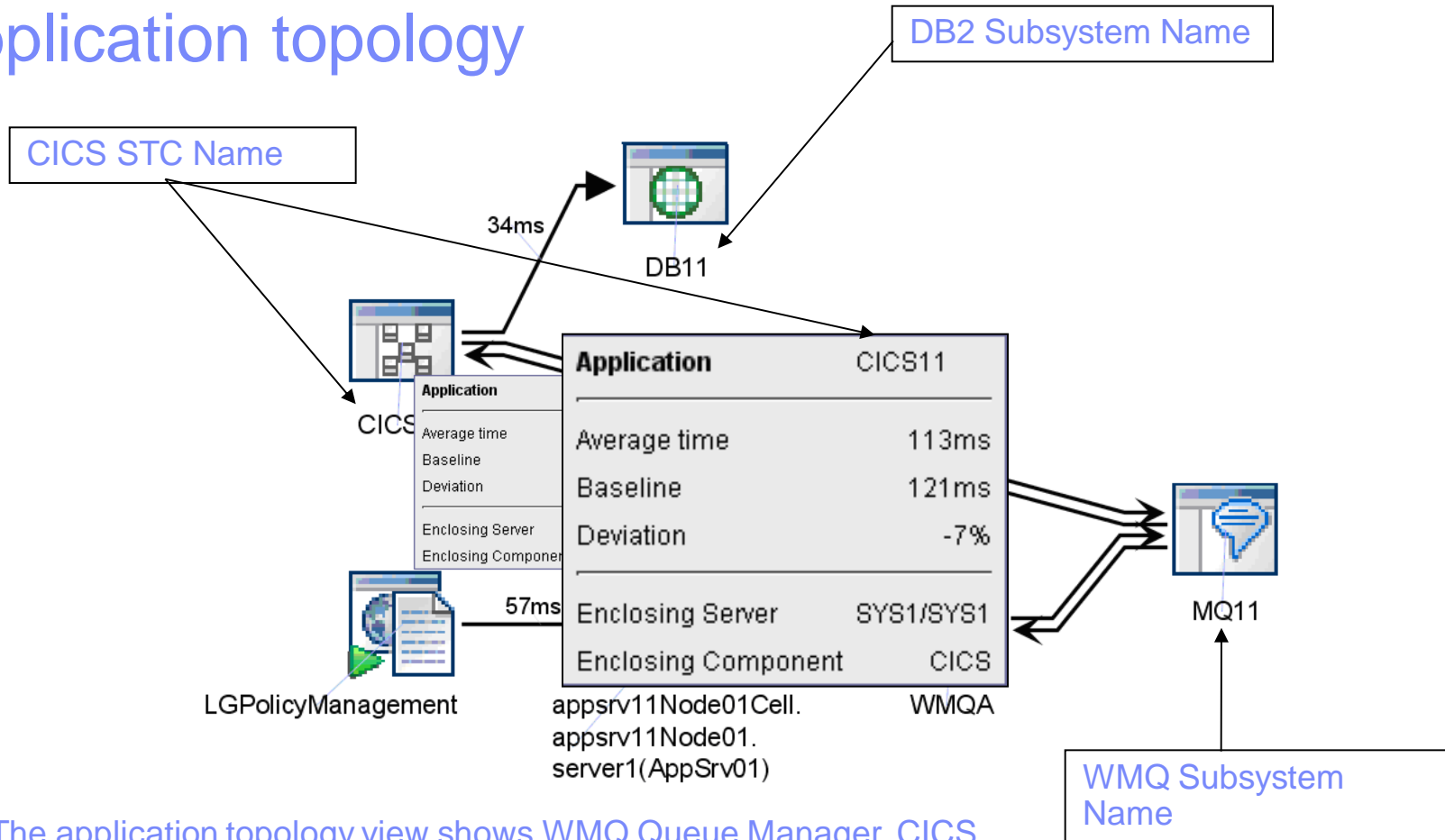
Simple server topology shows single distributed server interacting with a single z/OS server.

Component topology



Component view provides more insight into the deployment. On z/OS, WMQ into CICS (via the bridge) with a typical CICS to DB2 backend. A z/OS monoplex was chosen for simplicity.

Application topology

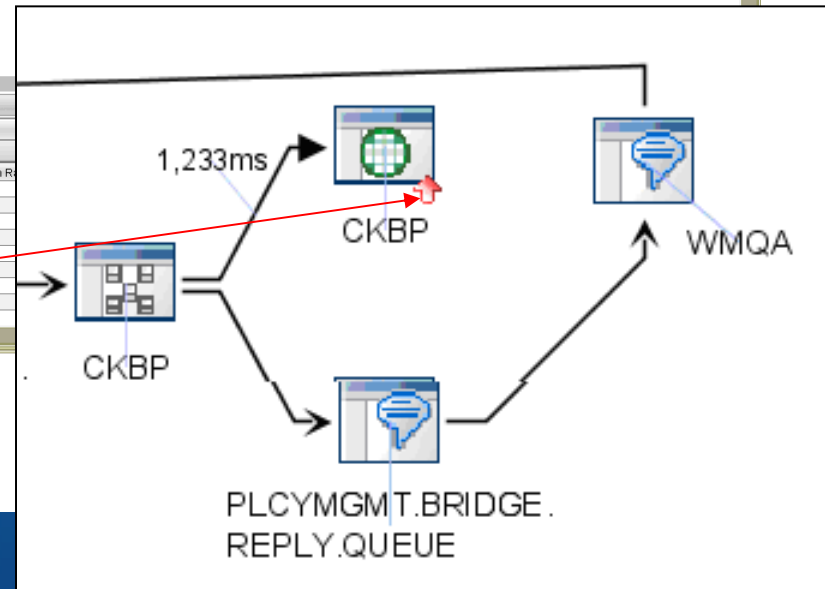
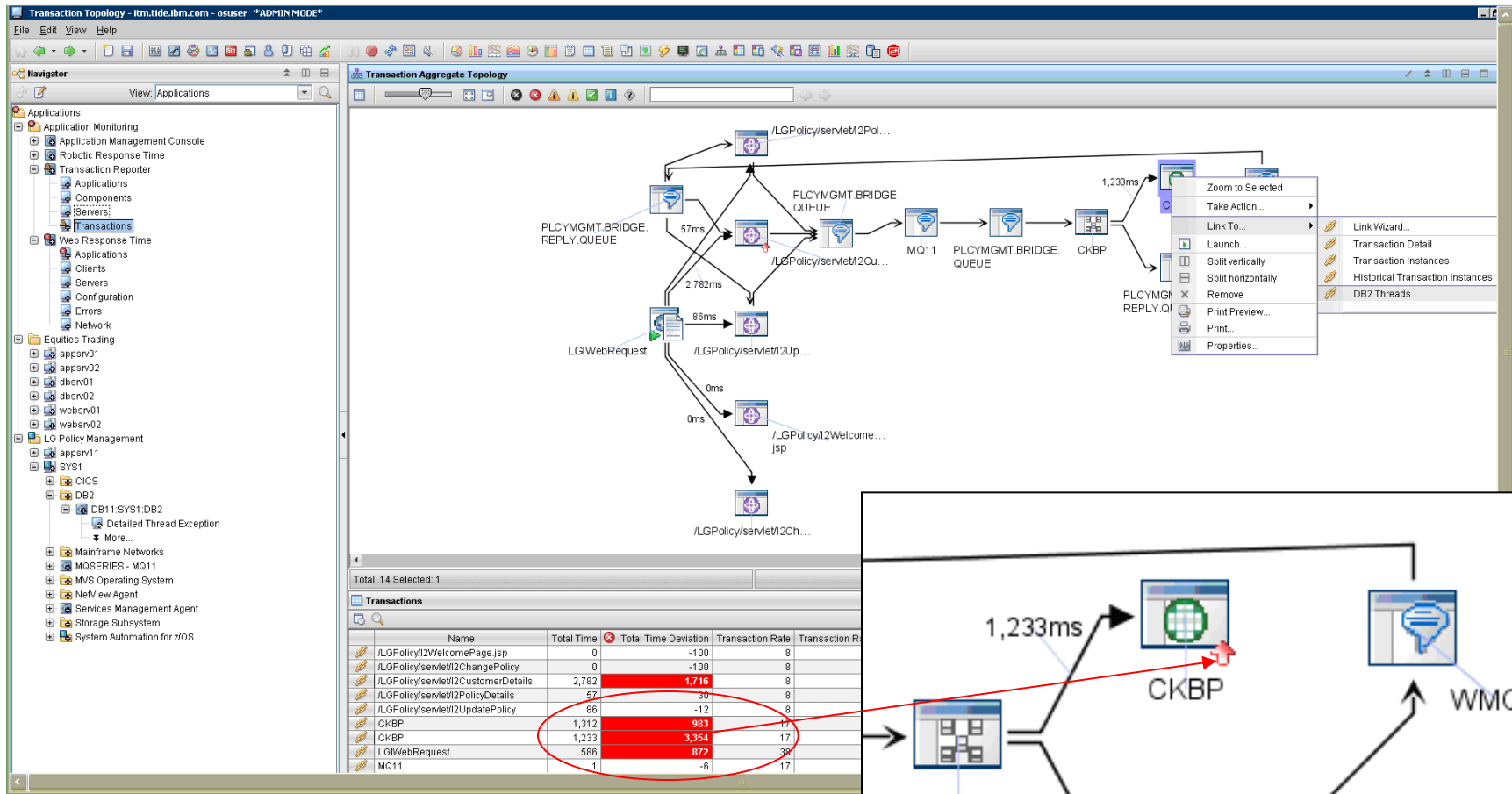


The application topology view shows WMQ Queue Manager, CICS STC, and the DB2 subsystem.

Mouse over shows z/OS host system and response time details.

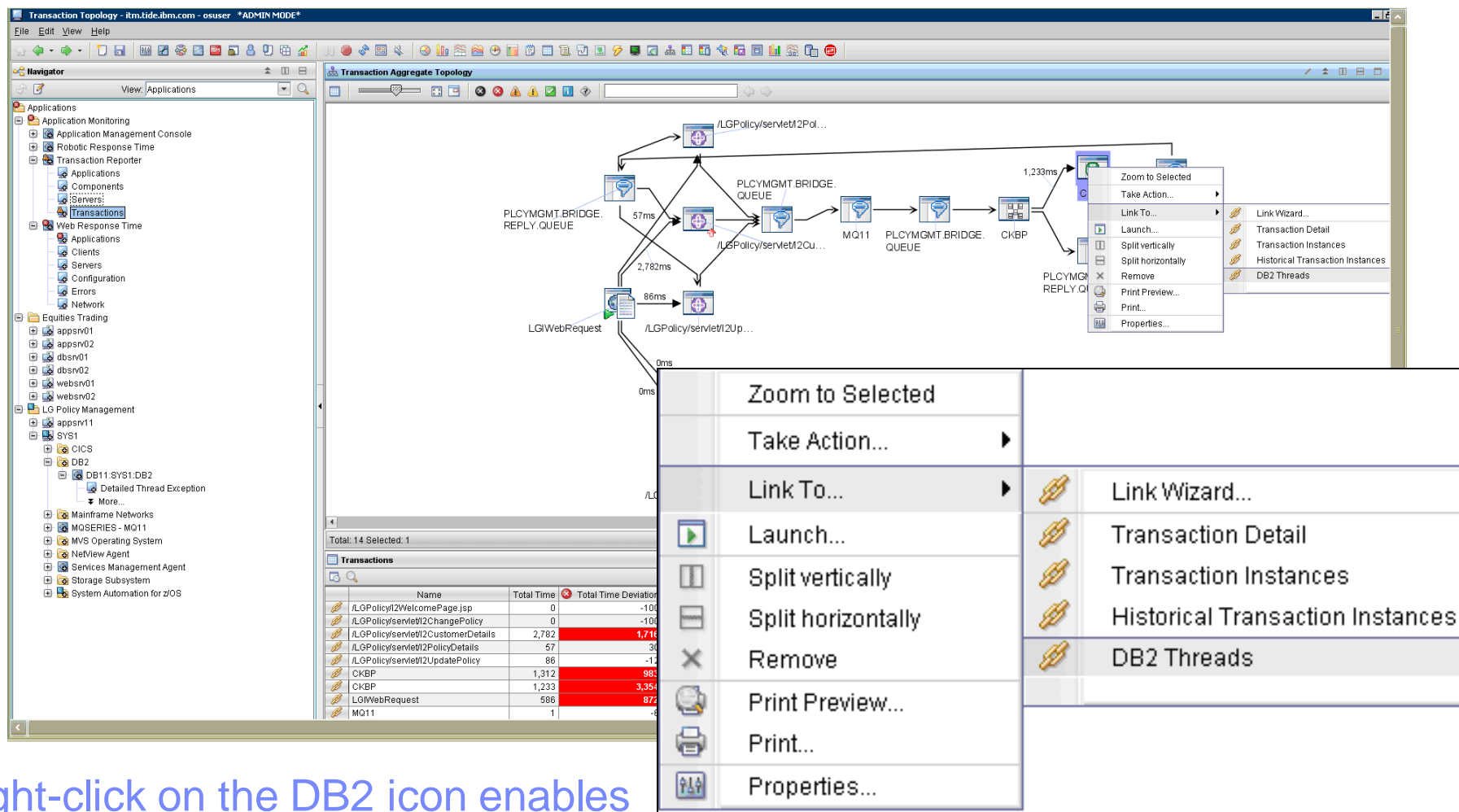
In a sysplex (shown later) we can use this view to locate a specific WMQ/CICS/DB2 on the associated sysplex member.

Transaction topology



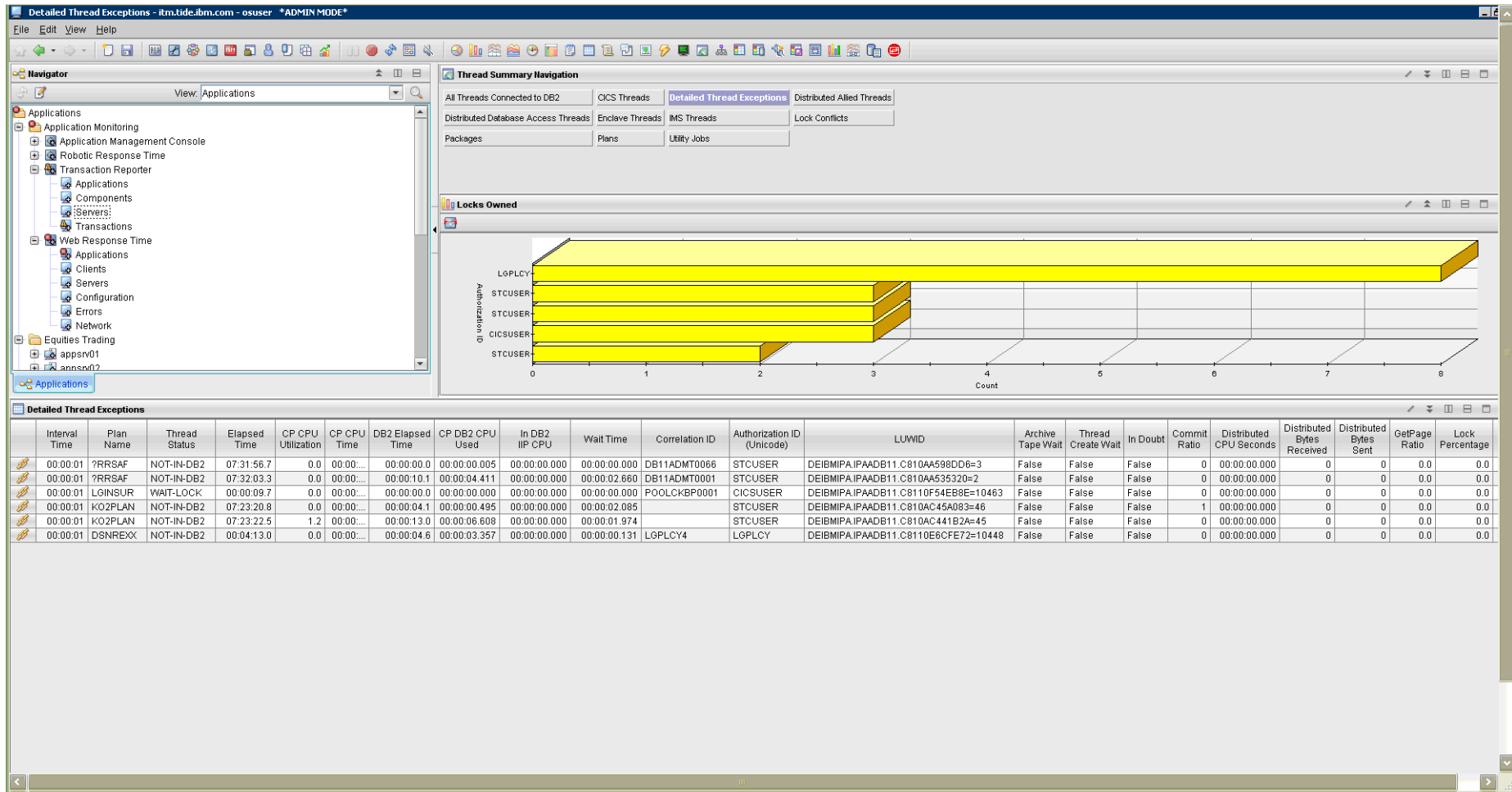
Red arrow indicates a response time deviation above 100%.
The high-lighted icon indicates a problem with DB2
associated with the CICS transaction CKBP

Transaction topology – Launch to Omegamon for DB2



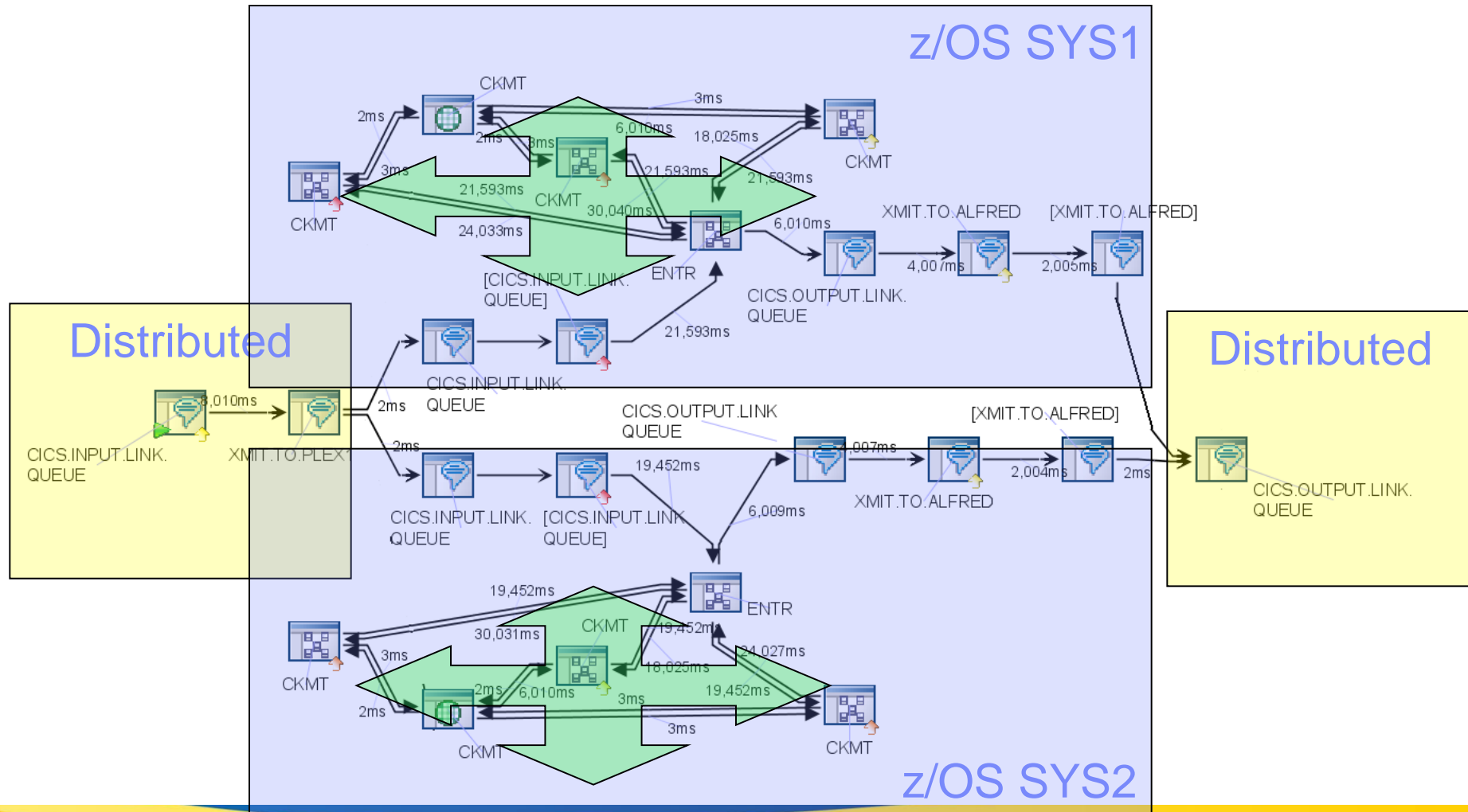
Right-click on the DB2 icon enables launch to Omegamon for DB2

Omegamon for DB2



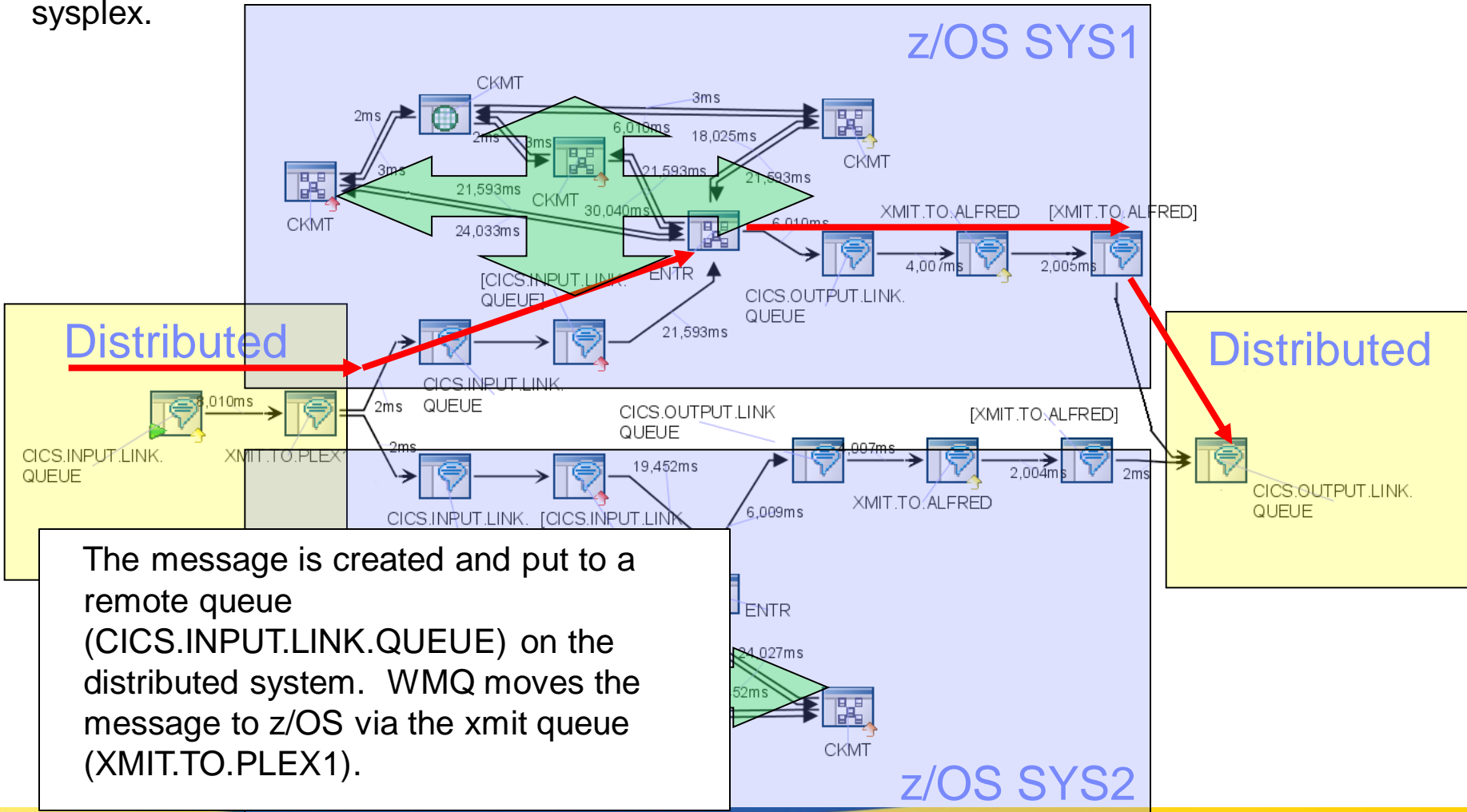
4-Way Sysplex WMQ → CICS → DB2

- Here is an example of a 4-way sysplex with distributed systems connected to z/OS using WMQ. For simplicity we are only showing 2 systems from a 4-way sysplex.



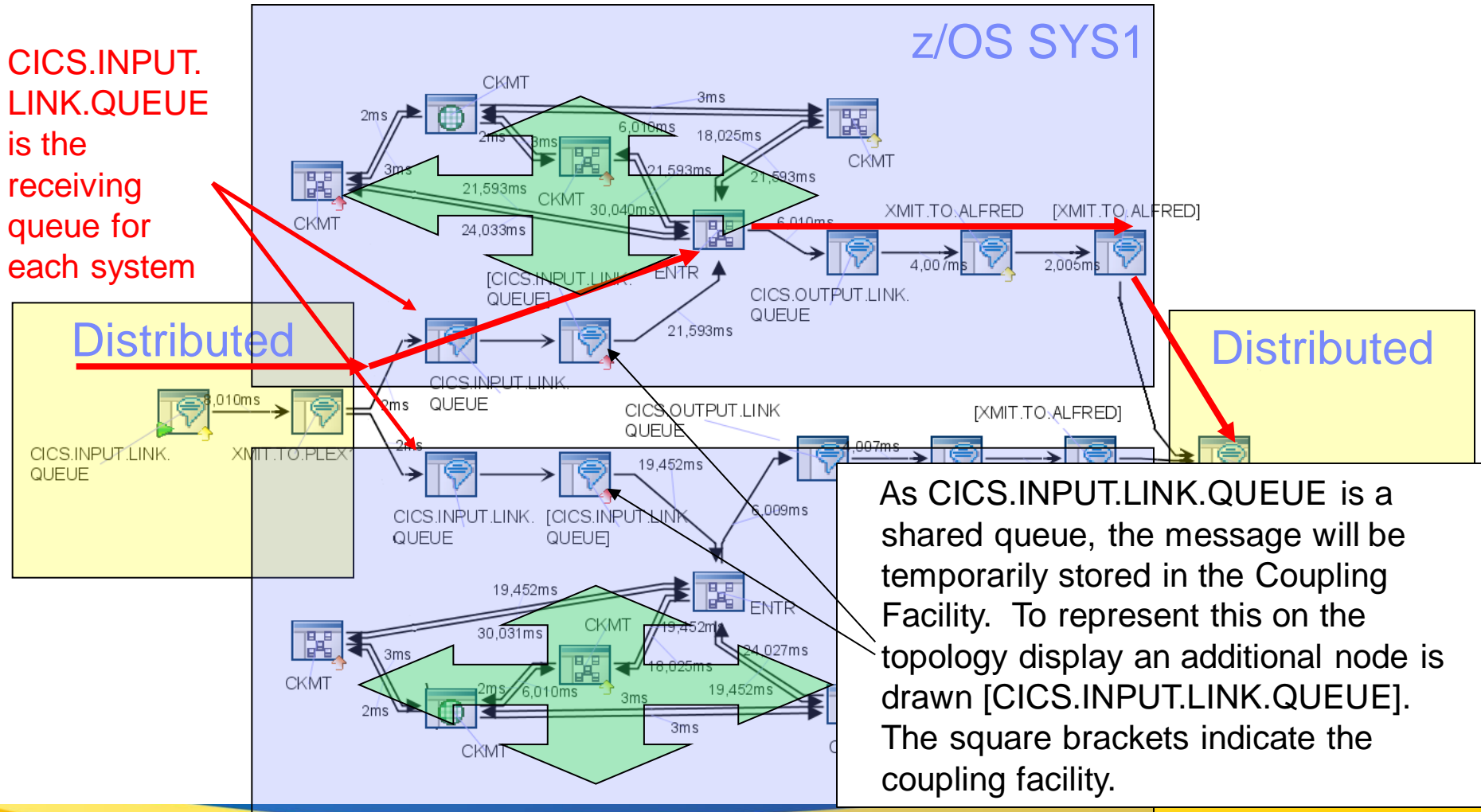
4-Way Sysplex WMQ → CICS → DB2

- The distributed application drives CICS and DB2 on the host. The red arrow shows the path of a WMQ message originating on a distributed system and flowing through to the z/OS sysplex.



4-Way Sysplex WMQ → CICS → DB2

- The z/OS system has a shared queue configuration so the message may be directed to any of the 4 systems in the sysplex. In our example, a message is transferred to system SYS1.

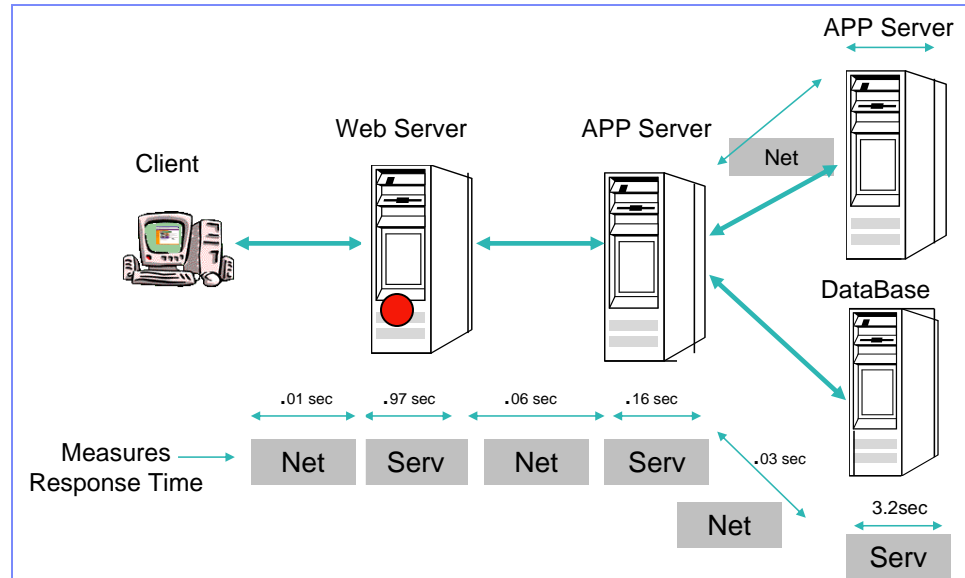


Transaction Tracking

Focused on simplicity and integration

Response Time Measurement

Monitors transaction performance and identifies end-user problems



Transaction Tracking

Consumes data from app server, MQ, CICS, IMS and custom instrumentation to show topology and isolate problems

Deep dive Analysis

Supports launch in context to SME capabilities including SME level tracking

Summary

- End User Monitoring is critical
 - ▶ Real User Monitoring
 - ▶ Synthetic Transaction Monitoring
- Transaction Tracking in today's complex environments is critical
- Performance needs to include all platforms/systems
 - ▶ Leads to closer collaboration between departments
- Standardize on tooling on each platform
- Standardize on metrics provided & measured where possible
- Standardize on a single pane of glass application view
- Standardize on dashboard design
- Applications should not go into production without adequate monitoring

Questions & Answers

