TXSeries deployment patterns & Product positioning

TXSeries V7.1 for Multiplatforms
The Next Generation of Distributed CICS
www.ibm.com/CICS

Agenda

- Common deployment patterns
  - Transactional integration server
  - Transaction server
- TXSeries as an Application modernization platform
- TXSeries as a component of SOA
- Product Positioning
- Comparing CICS Transaction Server and TXSeries
  - When to choose what?

Deployment Patterns

Common deployment patterns

TXSeries as a transactional integration server

A consolidating mid-tier terminal server

An intelligent mid-tier gateway between a J2EE™ application server and more than one EIS

A comprehensive mid-tier integration server

TXSeries as a transaction server

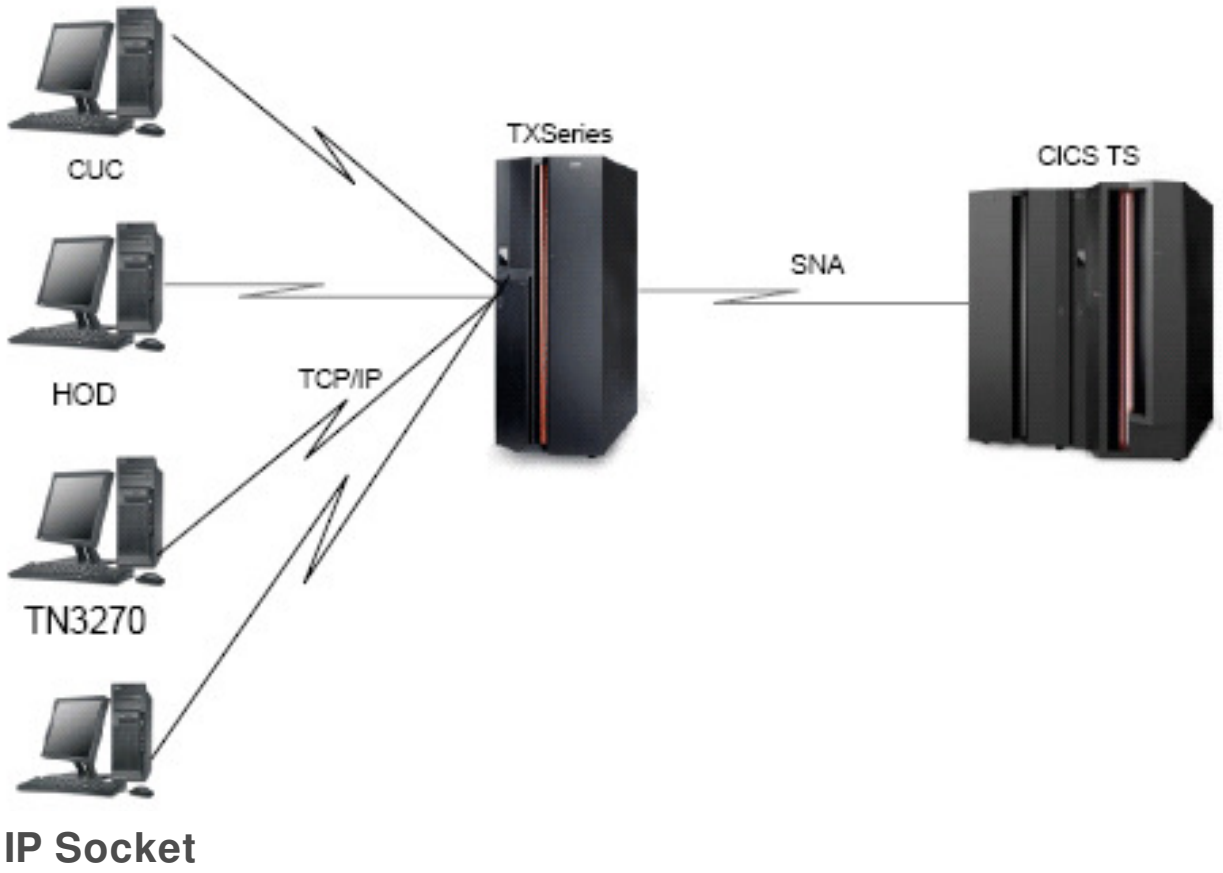
An entry level, stand-alone transaction server

Non-J2EE server in a mixed workload environment

A distributed server with mainframe connectivity

Common deployment patterns – Integration Server

1 **A consolidating mid-tier terminal server**



Key features

Reduced number of connections to the EIS, since many terminal connections
are replaced with one connection from TXSeries.

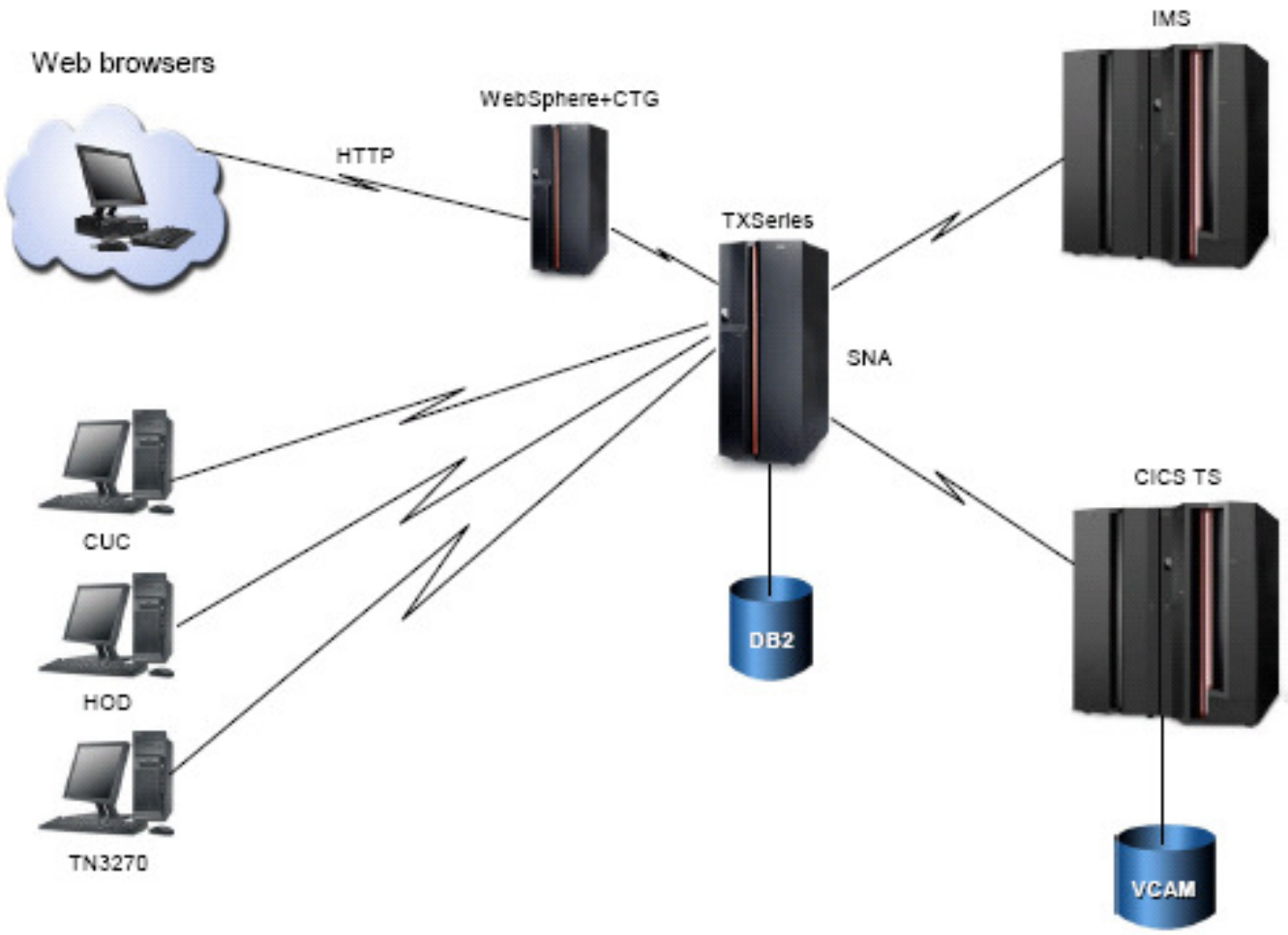
EIS is protected from client originated network disruptions.

Takes advantage of the wide range of terminals supported by TXSeries.

No duplication of business data on the TXSeries environment.

Common deployment patterns – Integration Server

2 **An intelligent mid-tier**



Key features

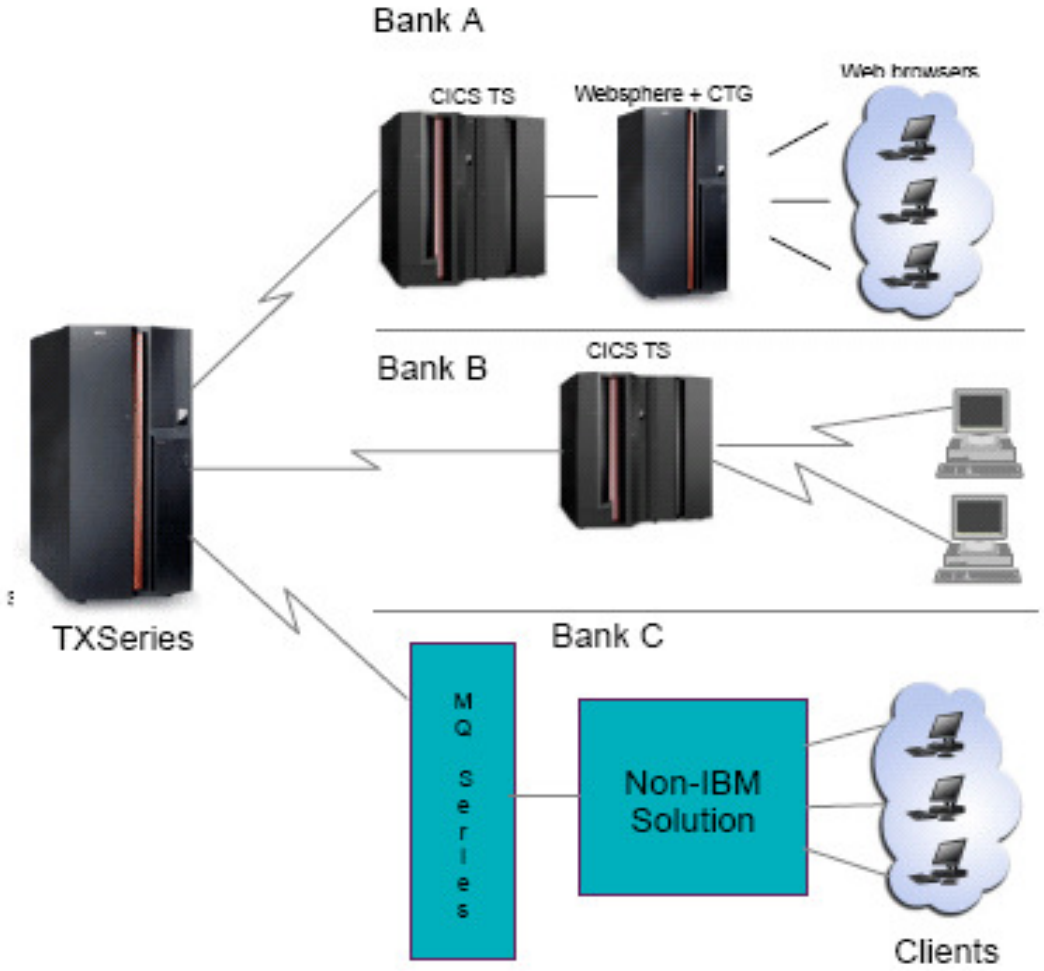
Reduced resource consumption on the EIS as TXSeries takes over a number of business operations.

Intuitive and rich Web based interfaces are now available to users.

Consolidated data from multiple EIS systems before presentation to users.

Common deployment patterns – Integration Server

3 **A comprehensive mid-tier integration server**



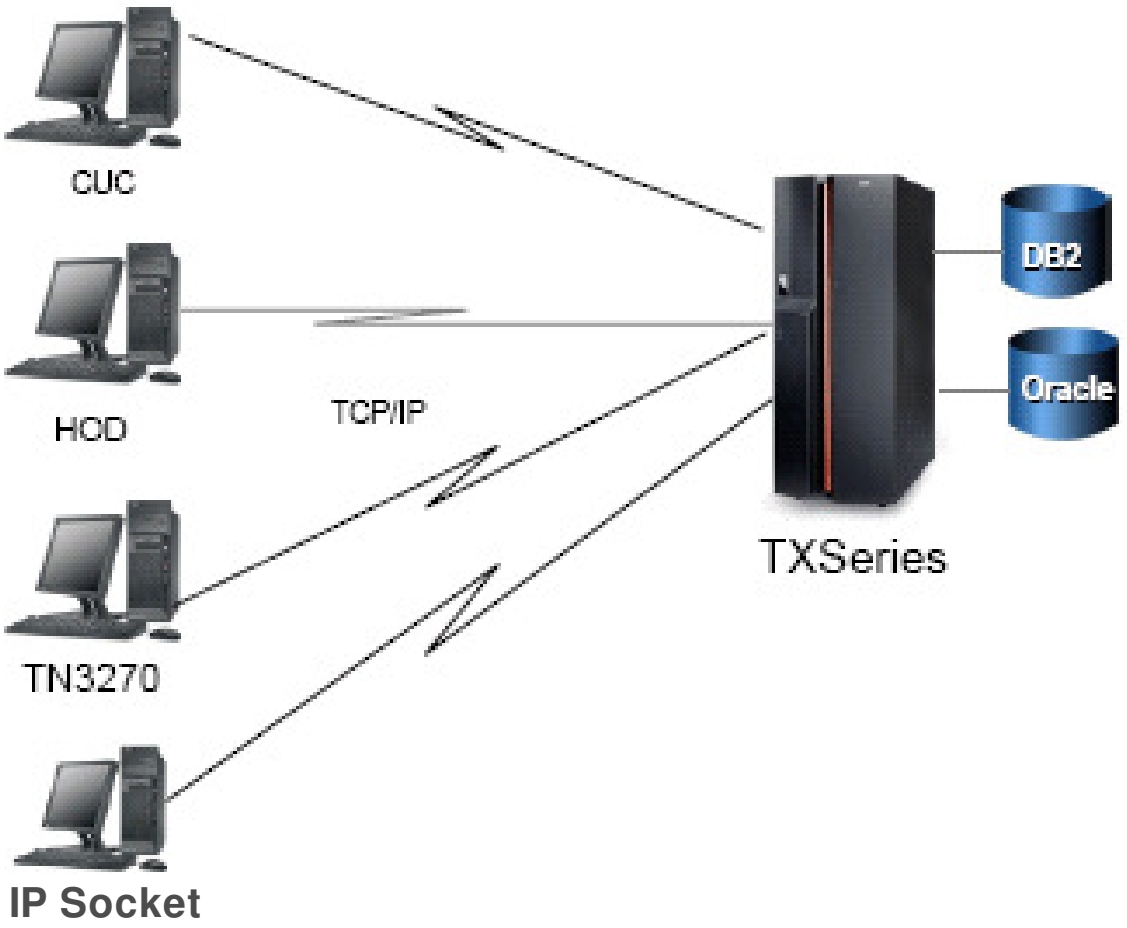
Key features

Essential business information taken from multiple EIS systems are consolidated, summarized, and made available in one place.

Rapid integration of heterogeneous EIS systems without the need to consolidate these systems.

Common deployment patterns – Transaction Server

1 **A stand-alone distributed transaction server**



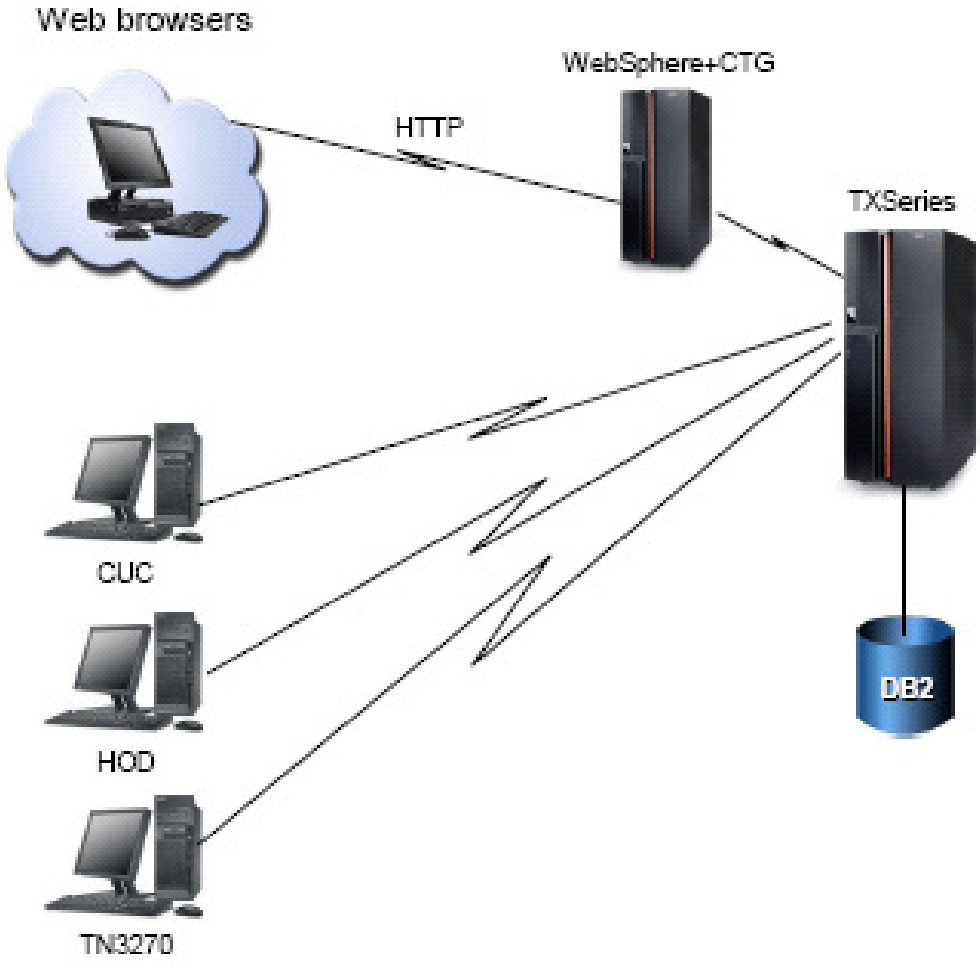
Key features

Cost effective and robust solution for deploying and running business transactions.

Ideal solution for customers who do not have CICS or large scale EIS systems but require the transaction handling qualities of service offered by CICS.

Common deployment patterns – Transaction Server

2 A non-J2EE server in a mixed workload environment



Key features

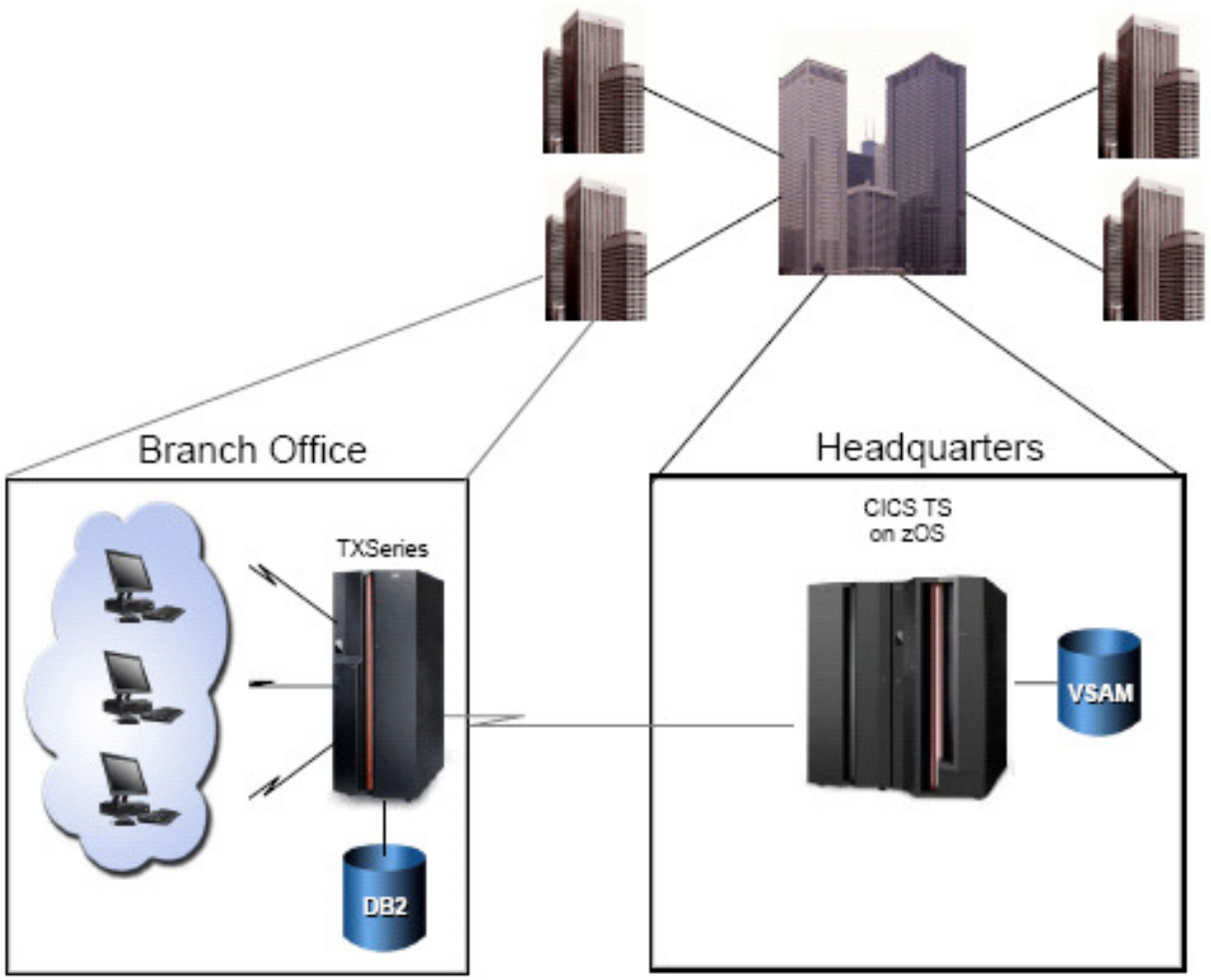
Access to TXSeries applications from a Java-based environment.

Transactional integrity is propagated automatically by the CICS Transaction Gateway from WebSphere Application Server to TXSeries. This ensures that Web-based users receive the same, or better, qualities of service from TXSeries as conventional terminal-based users.

Business applications are written using a combination of J2EE and traditional CICS programming languages.

Common deployment patterns – Transaction Server

3 **A distributed server with mainframe connectivity**



Key features

There is full end-to-end integration between WebSphere Application Server, CICS Transaction Gateway, CICS, other EIS systems, and TXSeries.

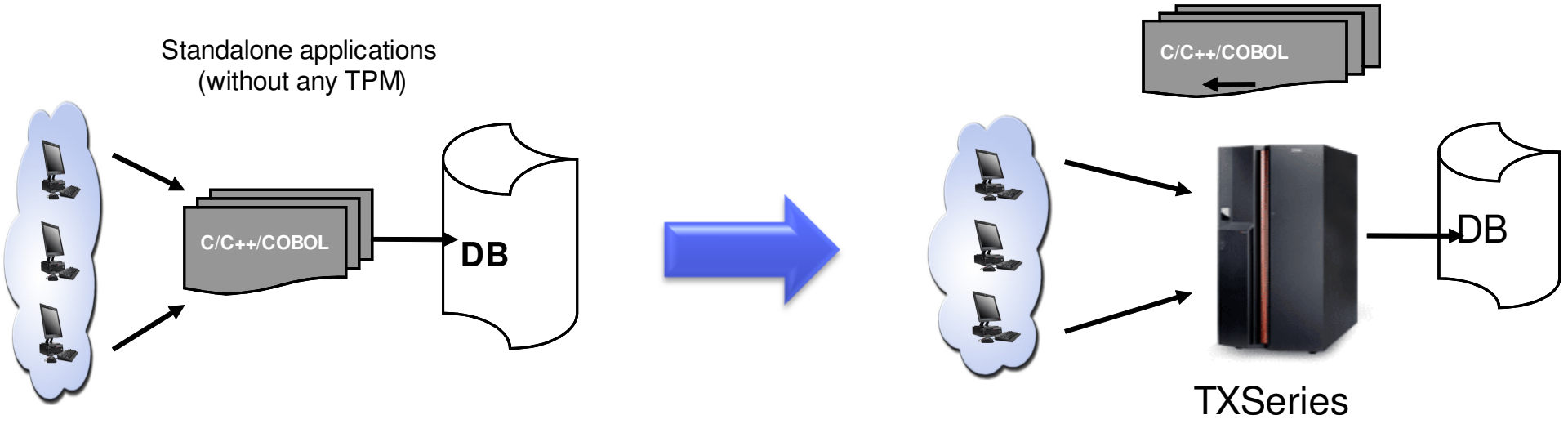
Solutions can be simple or complex depending upon the needs of the organization and the demands of the users.

As the organization grows or changes the architecture can remain constant while utilizing the flexibility of the CICS family to select the most appropriate combination of products for the users' needs.

TXSeries – As an application modernization platform

Modernize your standalone COBOL/C/C++ applications by hosting on to TXSeries

Extend the life of applications by providing transactional capability and ability to expose them into the SOA architecture by leveraging the capability of TXSeries



Only Retain core business logic within application and leverage TXSeries for all other enterprise scale requirements

Leverage TXSeries to:

Improve the reliability, and availability of your mission critical applications

Create a highly scalable environment using Work Load Manager (WLM) to handle future business growth

Expose existing applications into a SOA environment

Support data integrity, transaction recovery, and a host of other TPM functionality

Extend enterprise data to the web

TXSeries as a component of SOA

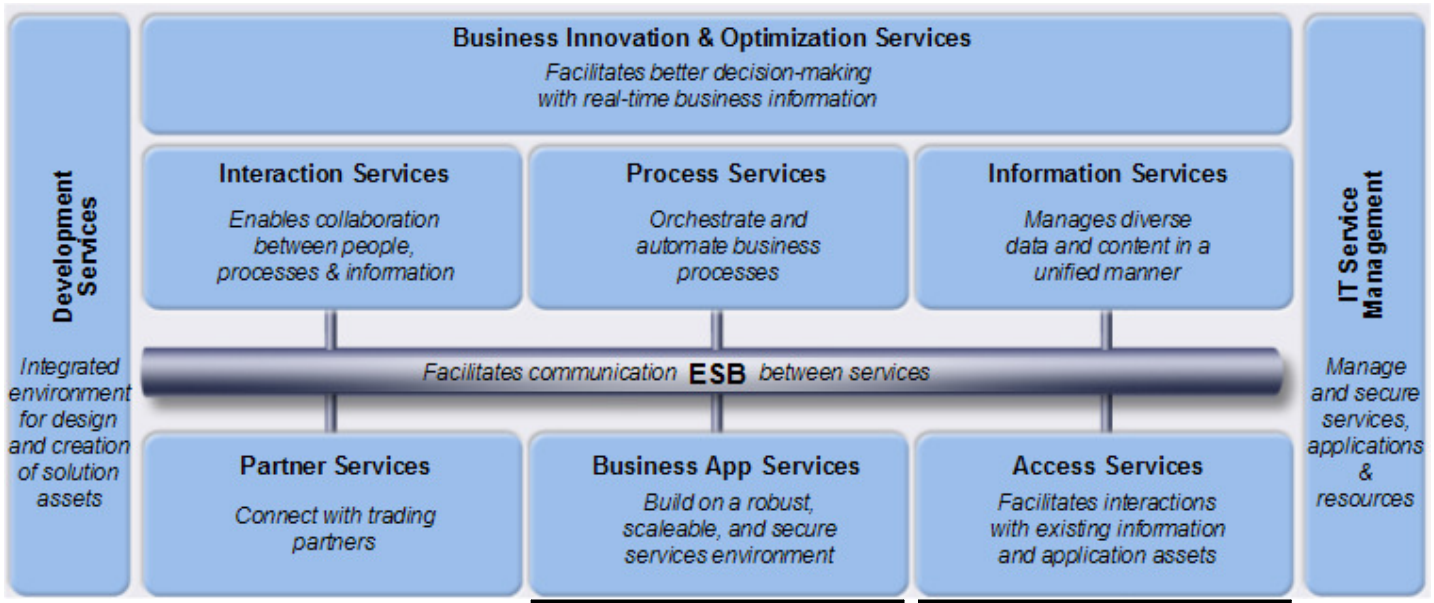
TXSeries enables end-to-end, distributed, mixed-language SOA through integration with WebSphere and CICS Transaction Server for z/OS

The JCA interface provided in the CICS TG connects TXSeries to the following WebSphere SOA server products:

WebSphere ESB
WebSphere Application Server
WebSphere Process Server

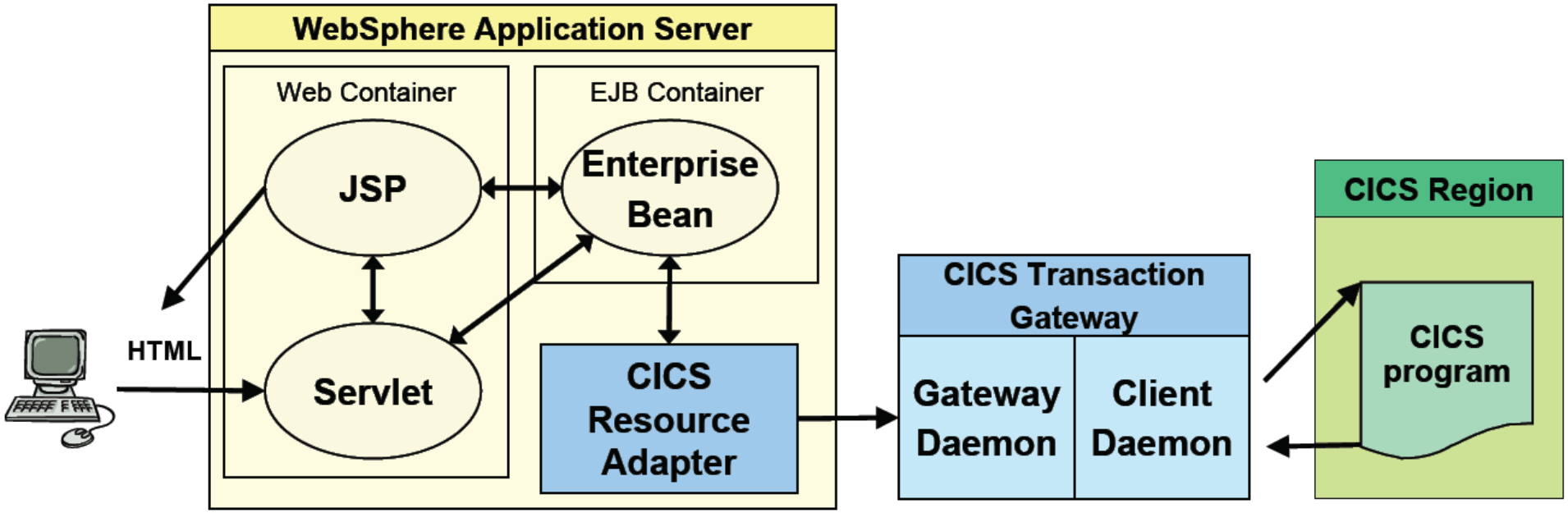
TXSeries with WebSphere MQ can connect to:

WebSphere Message Broker
Any other product that supports native MQSeries transport



TXSeries as a component of SOA – How it works ?

Transforming existing application into Web Services



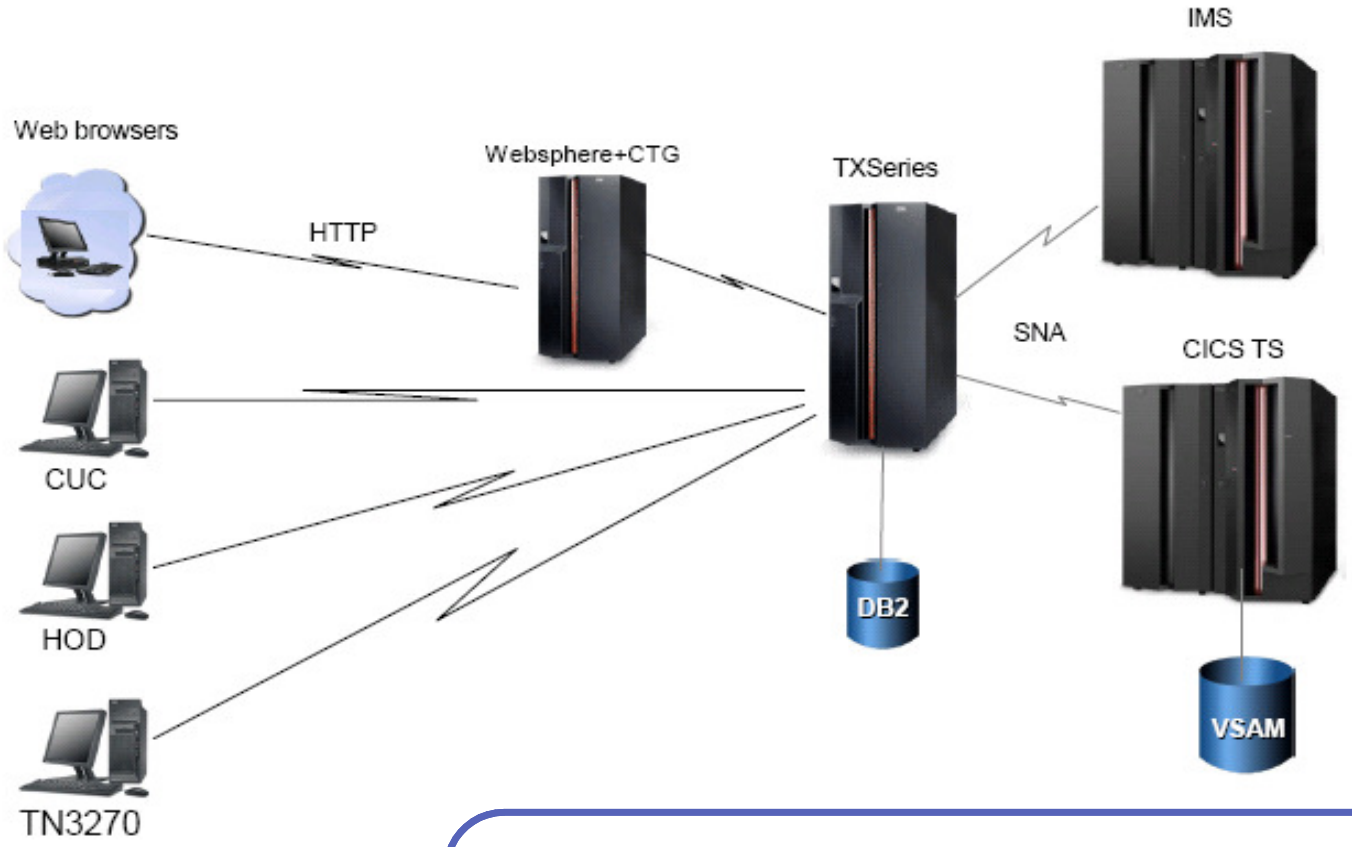
WebSphere Application Server:

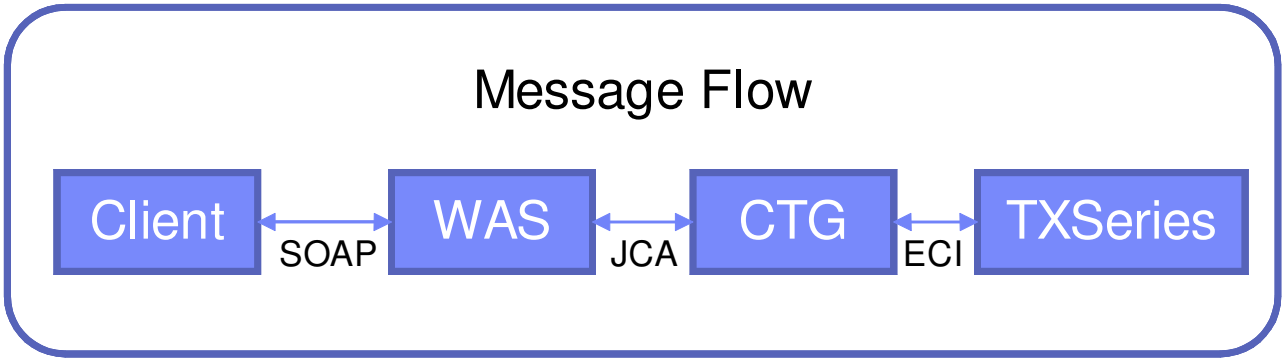
J2EE programming model
JSP, Servlets, EJB and Web Services
J2EE Connector Architecture (JCA)

CICS Transaction Gateway:

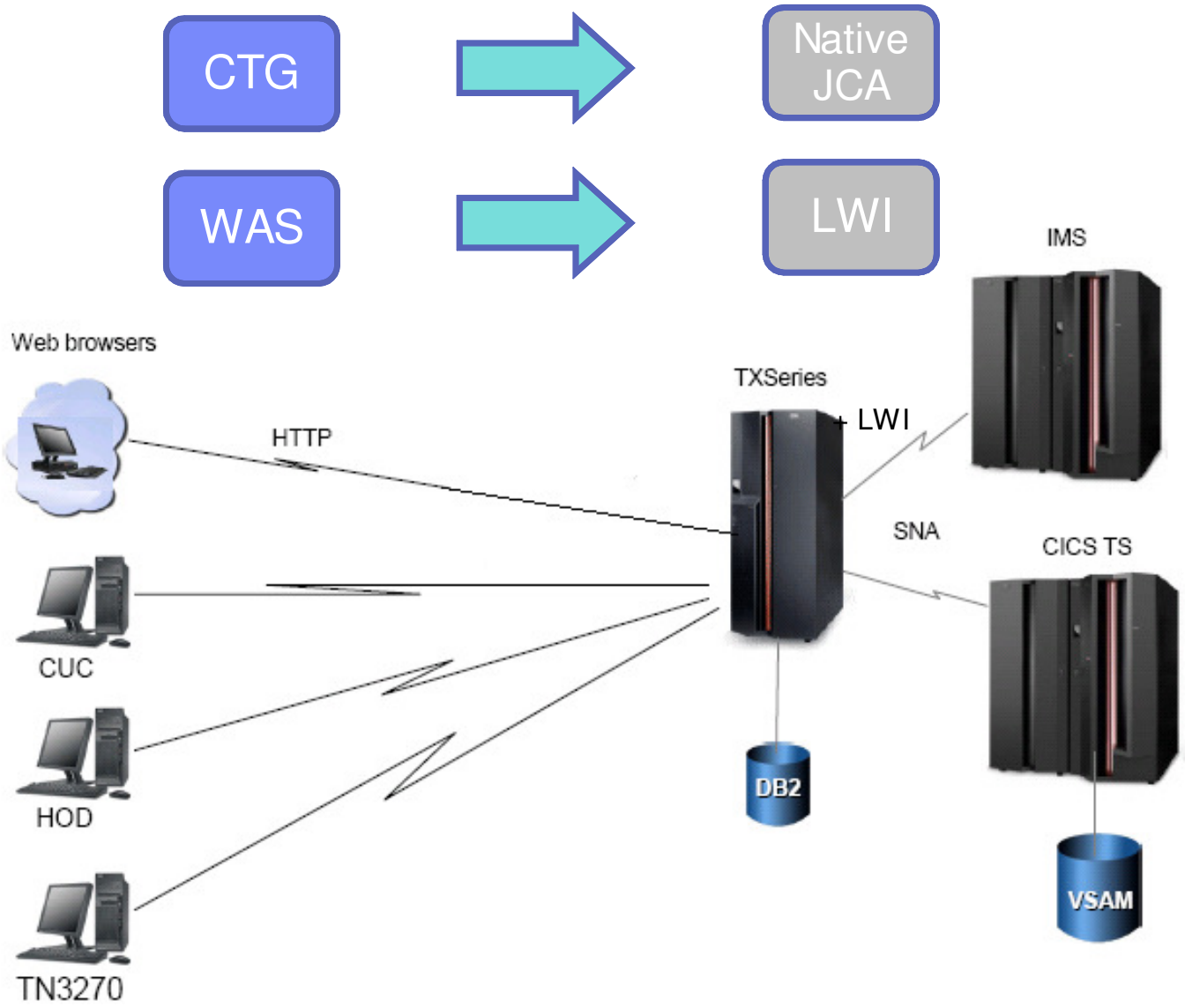
JCA adapter for TXSeries
Gateway daemon
functions as a server to Java clients
Supports TCP, SSL, HTTP, HTTPS
Client daemon
Provides connectivity to TXSeries/CICS-TS

Deploying Web Services using WAS & CTG





Deploying Web Services through native support



Product Positioning

Product Positioning

Complementary to J2EE and mainframe environments

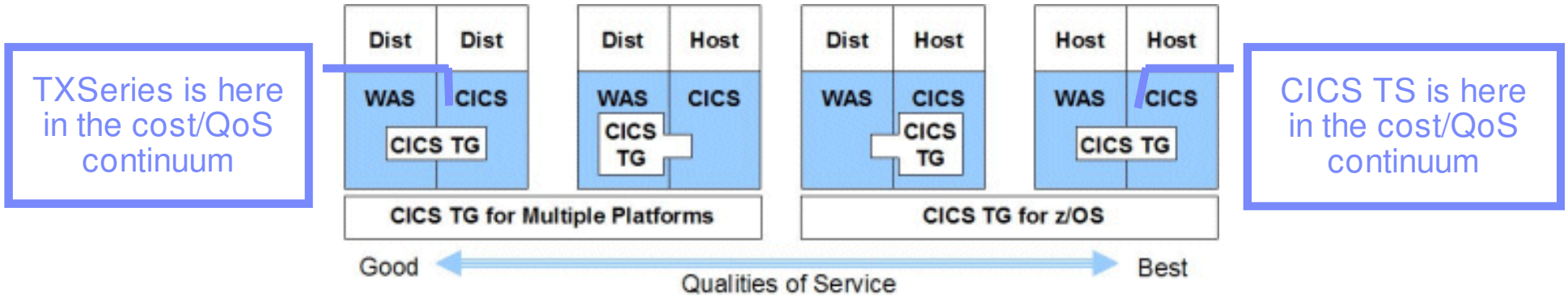
Stand-alone: distributed CICS server for business critical workloads

Integration: with WebSphere and CICS TS

Supports JCA based connectivity to WebSphere, via CICS Transaction Gateway

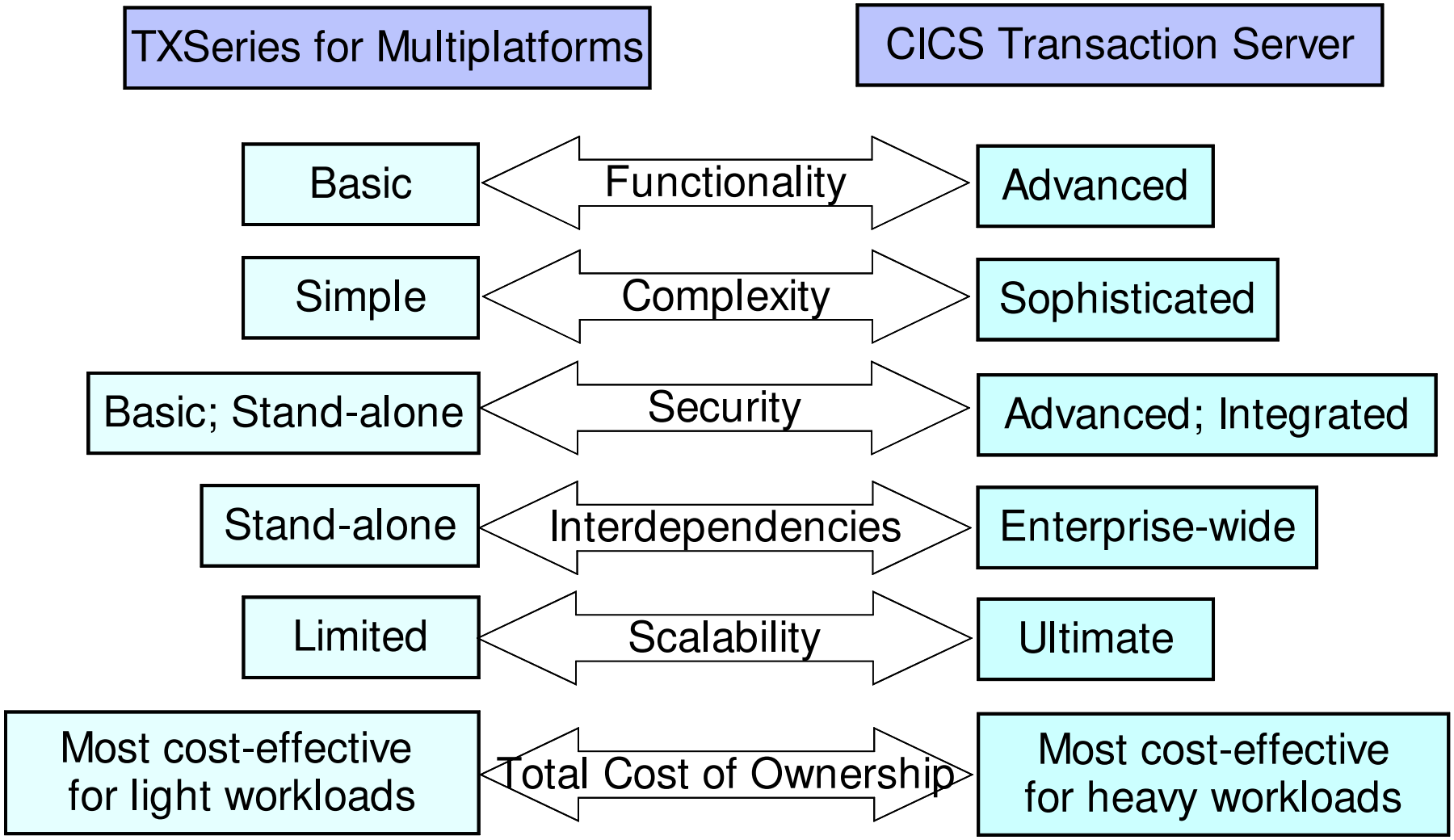
Enables end to end distributed mixed language solutions

Often used in support of CICS TS for z/OS because of it's excellent CICS interconnectivity



Comparing CICS Transaction Server with TXSeries

When to choose what ?



Thank you