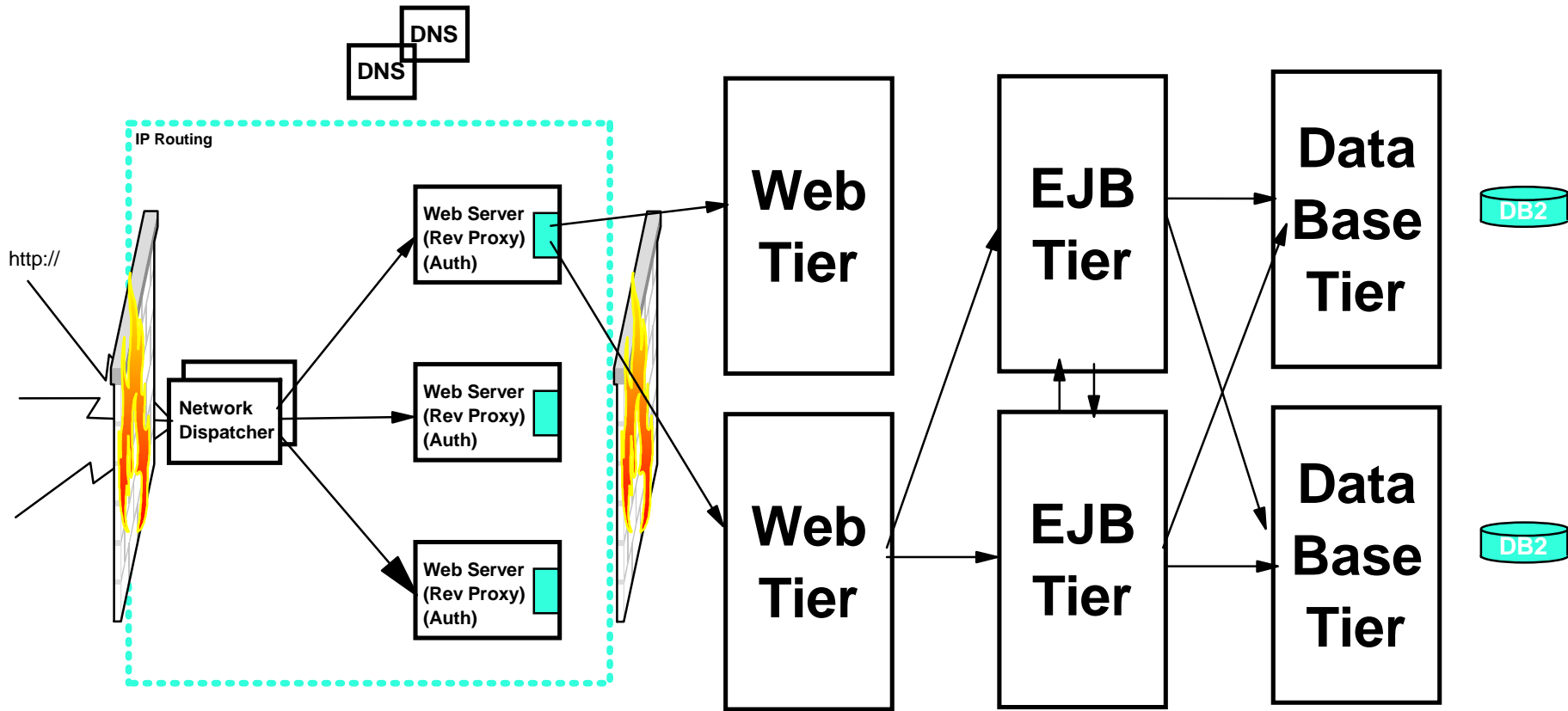# WebSphere for z/OS

A Cool Way to Boost Performance with IBM WebSphere for z/OS Applications
The Performance Advantage
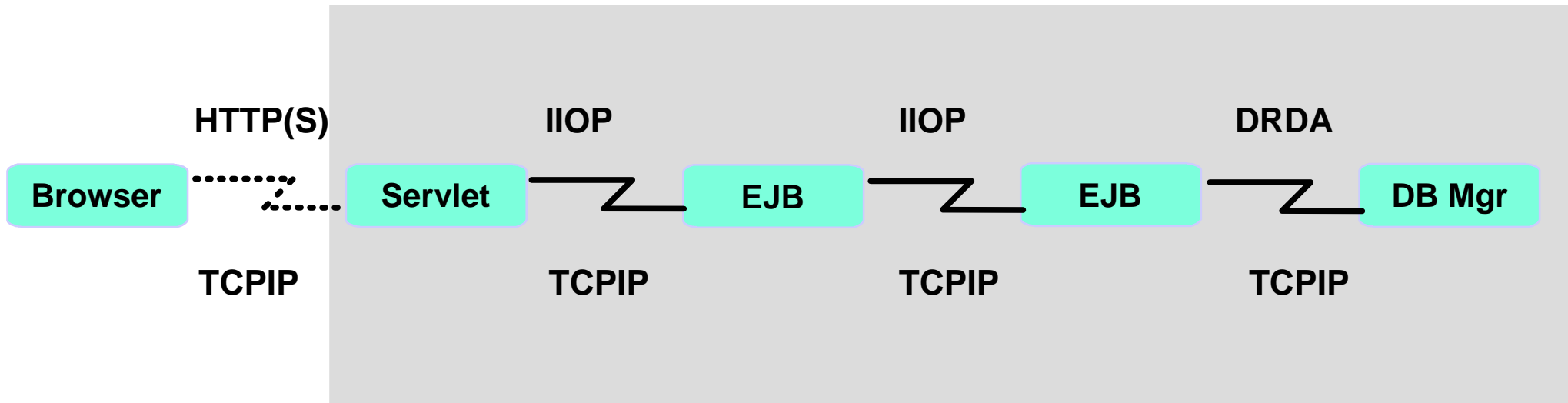
Mike Cox

IBM Washington Systems Center

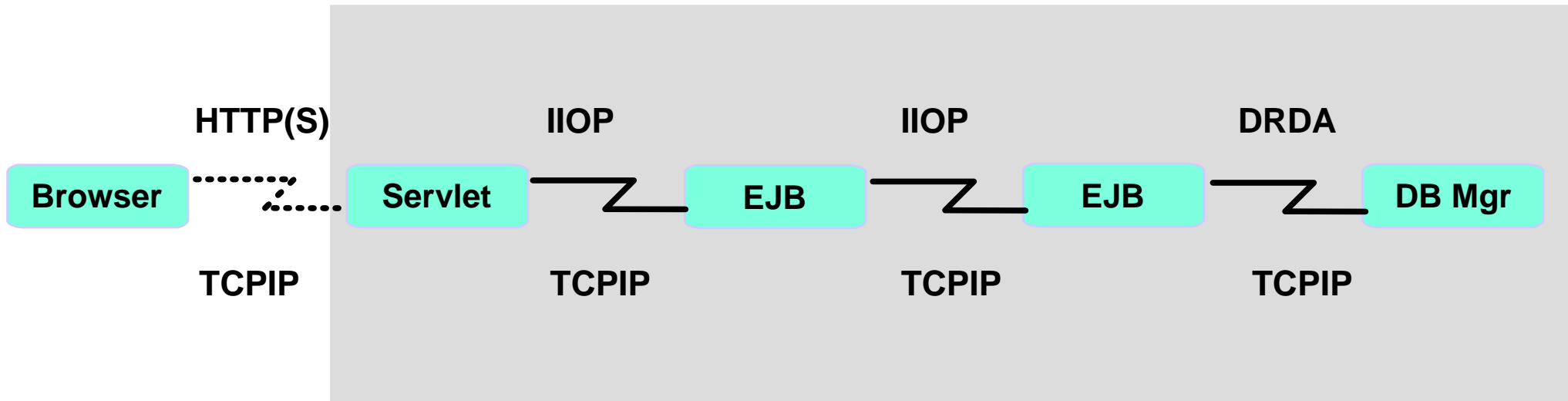© 2005 IBM Corporation

Server Topology Decisions

Flows

HTTP(S)   IIOP   IIOP   DRDA

| Browser | | Servlet | | EJB | | EJB | | DB Mgr |

TCPIP   TCPIP   TCPIP   TCPIP

The Problem

HTTP(S)   IIOP   IIOP   DRDA

| Browser | | Servlet | | EJB | | EJB | | DB Mgr |

TCPIP   TCPIP   TCPIP   TCPIP
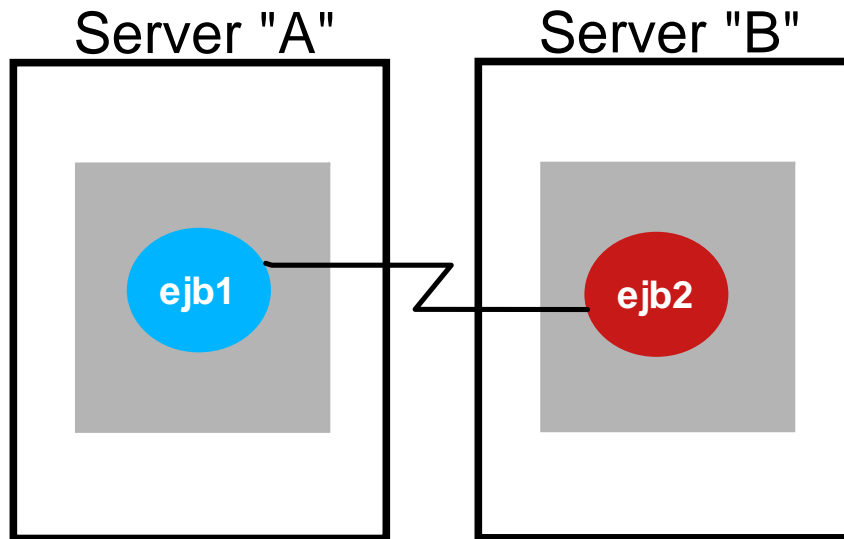
- **There is an increase in latency**
  - ► **There is a re-dispatch at every layer**
- **CPU cost associated with flowing method parameters and arguments between the layer**
  - ► **There is a network bandwidth cost as well**
- **The importance of the unit of work is not carried across all the layers consistently.**
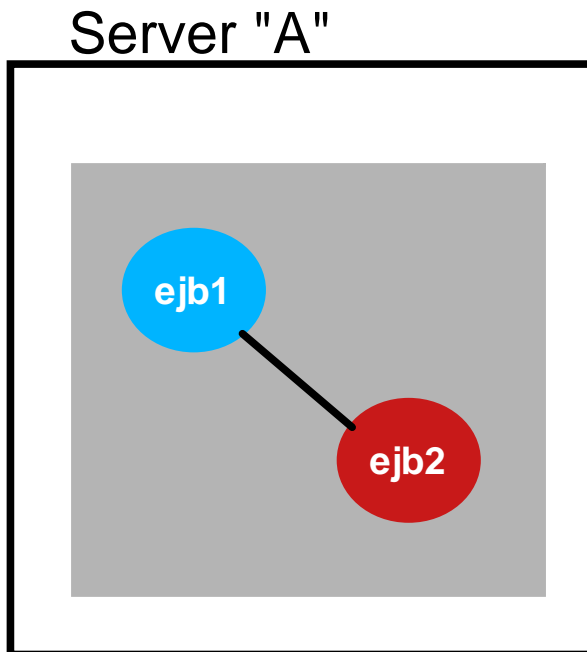
Remote EJB Invocation

Server "A"   Server "B"


```
...
myResults = myEjb2.doIt(obj1,obj2);
...
```

- **The runtime in Server "A" must serialize the arguments to the doit( ) method into wire format (byte array).**
- **The byte array is passed to Server "B"**
- **The runtime in Server "B" de-serializes the objects from the byte array**
- **The doIt( ) method is dispatched and does what it has to do to create the results.**
- **The runtime in Server "B" must serialize the results of the doit( ).**
- **The byte array is passed to Server "A"**
- **The runtime in Server "A" de-serializes the results**
- **Ejb1 is re-dispatched with myResults and continues on.**

Trace of EJB remote method

Jinsight trace of entire dispatch of EJB doIt( ) method driven by remote client.
Interested in three phases of processing.

CDRInputStream - Object Deserialization

doIt() method - produces the result set to be returned to client

CDROutputStream - Object Serialization

"Local" EJB Invocation
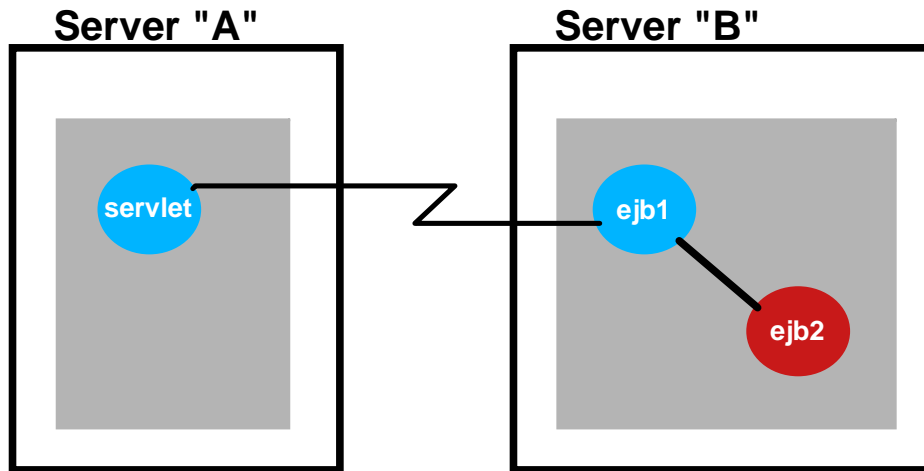
Server "A"

```
ejb1
ejb2
```

- **Placing the two EJBs in the same server does not necessarily mean that the serialization and deserialization CPU costs are gone.**
- **Serialization/de-Serialization goes away when method parameters and results are "passed by reference"**
- **Two ways of having objects passed by reference.**
  1. **Create Local Interfaces on the EJBs**
  2. **Configure the runtime to pass objects by reference when components are in the same server.**
- **WebSphere supports the EJB Specification (option 1) and a "NoLocalCopies" optimization on Remote Interfaces (option 2)**
  - ▶ **WebSphere also supports pre-fer local when there is co-resident and remote target bean.**

```
...
myResults = myEjb2.doIt(obj1,obj2);
...
```

The Entire Flow
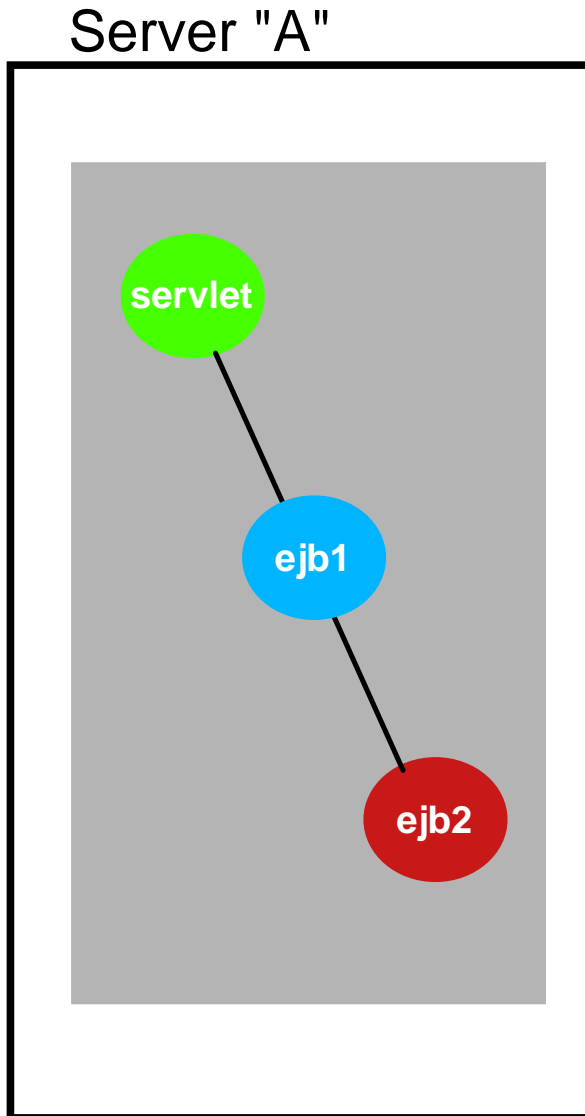
Server "A"

- **TheApplication Components are deployed in one server**
- **The server/application is configured to pass objects by reference.**
  - ► **The serialization/de-serialization CPU cost are gone.**
- **The servlet will invoke ejb1 on the same thread of execution AND ejb1 will invoke ejb2 on this same thread of execution**
  - ► **There is no redispatch of individual components. There is one unit of work.**
  - ► **All the processing by all components is managed toward the service classification assigned to the servlet.**

What if...

Server "A"   **Server "B"**

servlet

ejb1

ejb2

```
...
myResults = myEjb1.doIt(obj1,obj2);
...
```

What if the application components cannot be placed in the same server.

- and -

The serialzaton costs are "high".

Modify the application to implement:

1. **java.io.Serializable , or**
2. **java.io.Externalizable**

Serialize only the data needed to construct the object again.

► **Have seen cases where default processing created 35K byte array and customer serializer create 60 byte array.**

► **See "More About High-Volume Web Sites" SG24-6562, Chapter 6**

Remote Data Access

Server "A"

Server "B"

ejb1

DRDA

D
D
F DB2

Database
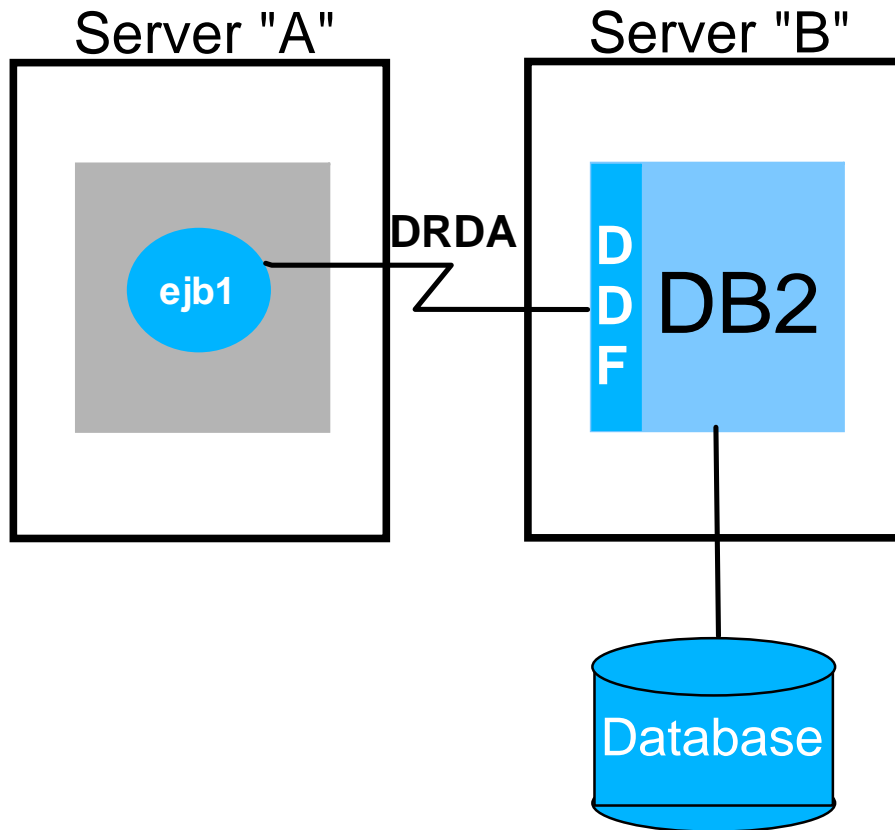
When using the jcc type-4 or type-4 (XA) JDBC driver

For SQL requests by ejb1 to DB2 the following must occur

- ► **The query is serialized into a byte array for transmission to DDF**
- ► **DDF must get dispatched and drive the request into DB2**
- ► **Result set is serialized and returned to Server "A"**
- ► **Ejb1 is dispatched with results**

Local Data Access

Server "A"
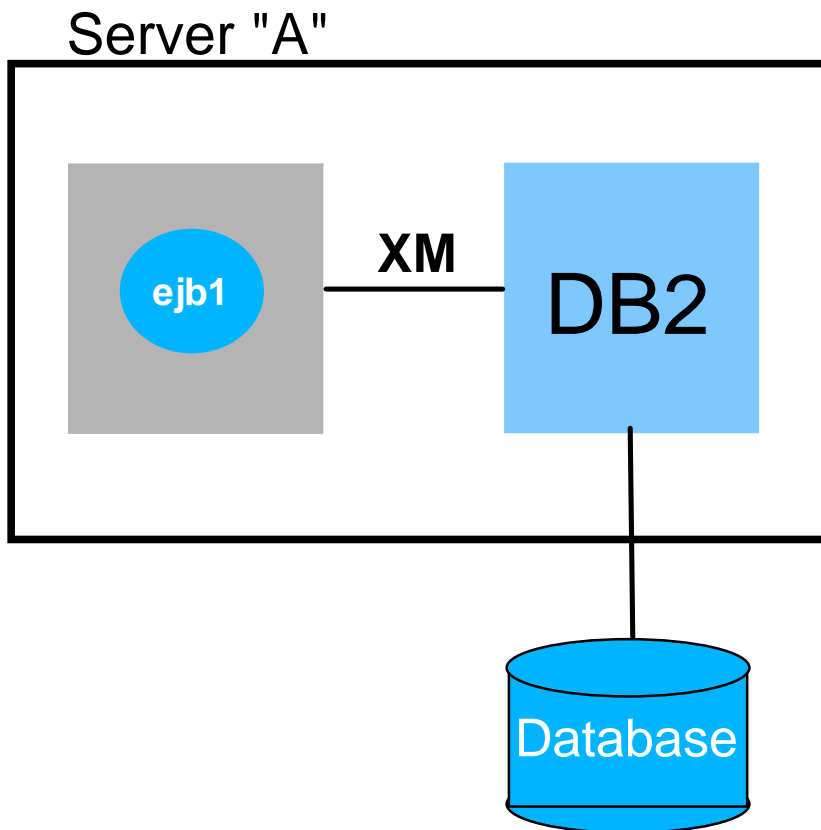
When using the jcc type-2 JDBC driver

SQL requests are:

- ► **Processed on the same thread of execution**
  - – **Managed to one WLM Goal.**
- ► **Access to DB2 engine is done cross memory**
  - ● **TCPIP and DDF eliminated**
  - ● **Reduced latency - consistent response times**
- ► **RRS is used rather XA flows for 2PC processing**
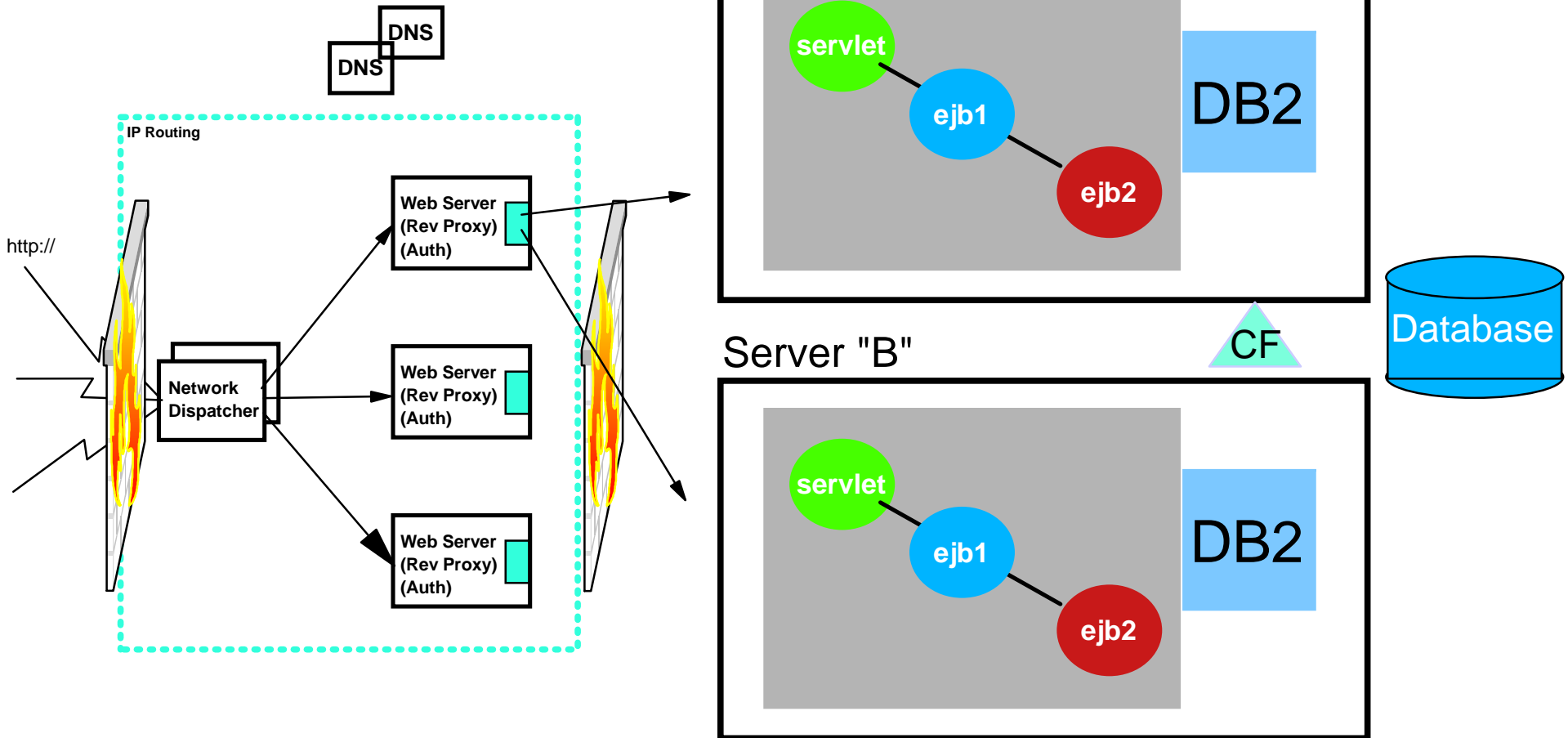
So what is this worth?

- **Refactoring Application to eliminate passing of large amounts of data between tiers**
  - ► CPU time in EJB tier: 11.73 ms -> 2.64 ms
  - ► Amount of data serialized: 54Kb -> 0.5 Kb

- **Using jcc type-4 vs type-2**
  - ► Trade3 Application: ~30% CPU increase
  - ► Customer Application: ~50% CPU increase

Caveat: Your mileage may differ. Just as there is no "typical" application, there is no "one" answer.

Server Topology Decision Revisited

Server "A"

servlet

ejb1

ejb2

DB2

DNS

DNS

IP Routing

http://

Network
Dispatcher

Web Server
(Rev Proxy)
(Auth)

Web Server
(Rev Proxy)
(Auth)

Web Server
(Rev Proxy)
(Auth)

Server "B"

servlet

ejb1

ejb2

DB2

CF

Database

WebSphere for z/OS

A Cool Way to Boost Performance with IBM
WebSphere for z/OS Applications
   The Security Advantage

Mike Kearney
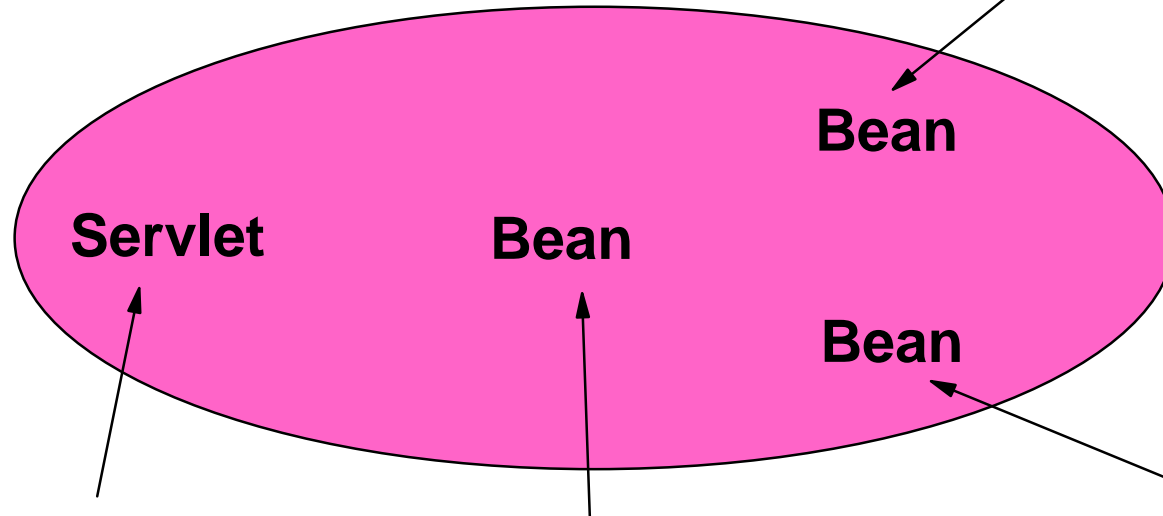**IBM Washington Systems Center**

Topics

- **Overview of WebSphere's security environment.**
- **Compare remote versus local environments.**
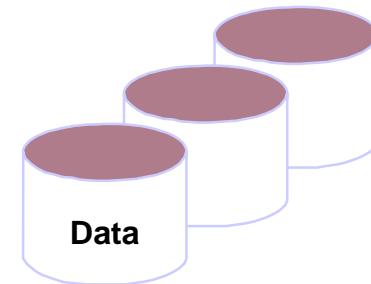- **Discuss options for propagating an identity.**

J2EE Application Security Environment

★ **When an EJB method calls another method, a check is made to see if the method identity has the required role.**

User

Servlet

Bean

Bean

Bean

Data

★ **In most business applications, authentication of the client is required.**

★ **In order to run the application, the client must have the required role.**

★ **A Servlet typically runs with the identity of the client.**

★ **When a Servlet calls an EJB method, a check is made to see if the Servlet identity has the required role.**

★ **EJB methods can run with the identity of the Caller, the Servant's identity or an identity associated with a role.**

★ **Connectors provide a standard way to communicate with existing information systems like DB2, CICS and IMS.**

★ **When an EJB method uses a connector, an identity is provided.**

★ **The identity depends on the type of connector being used.**
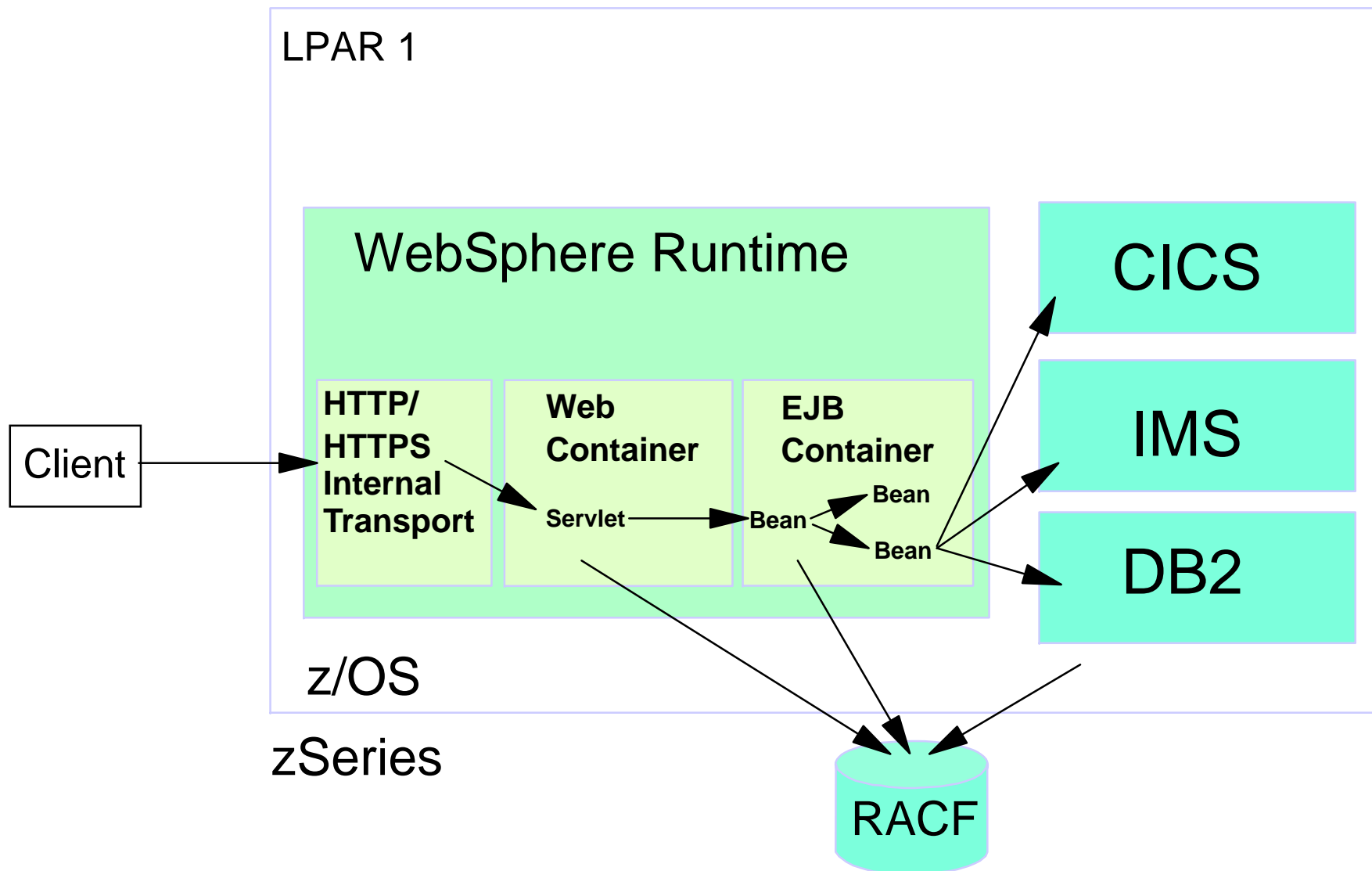
WebSphere V5/V6 'Remote' Environment
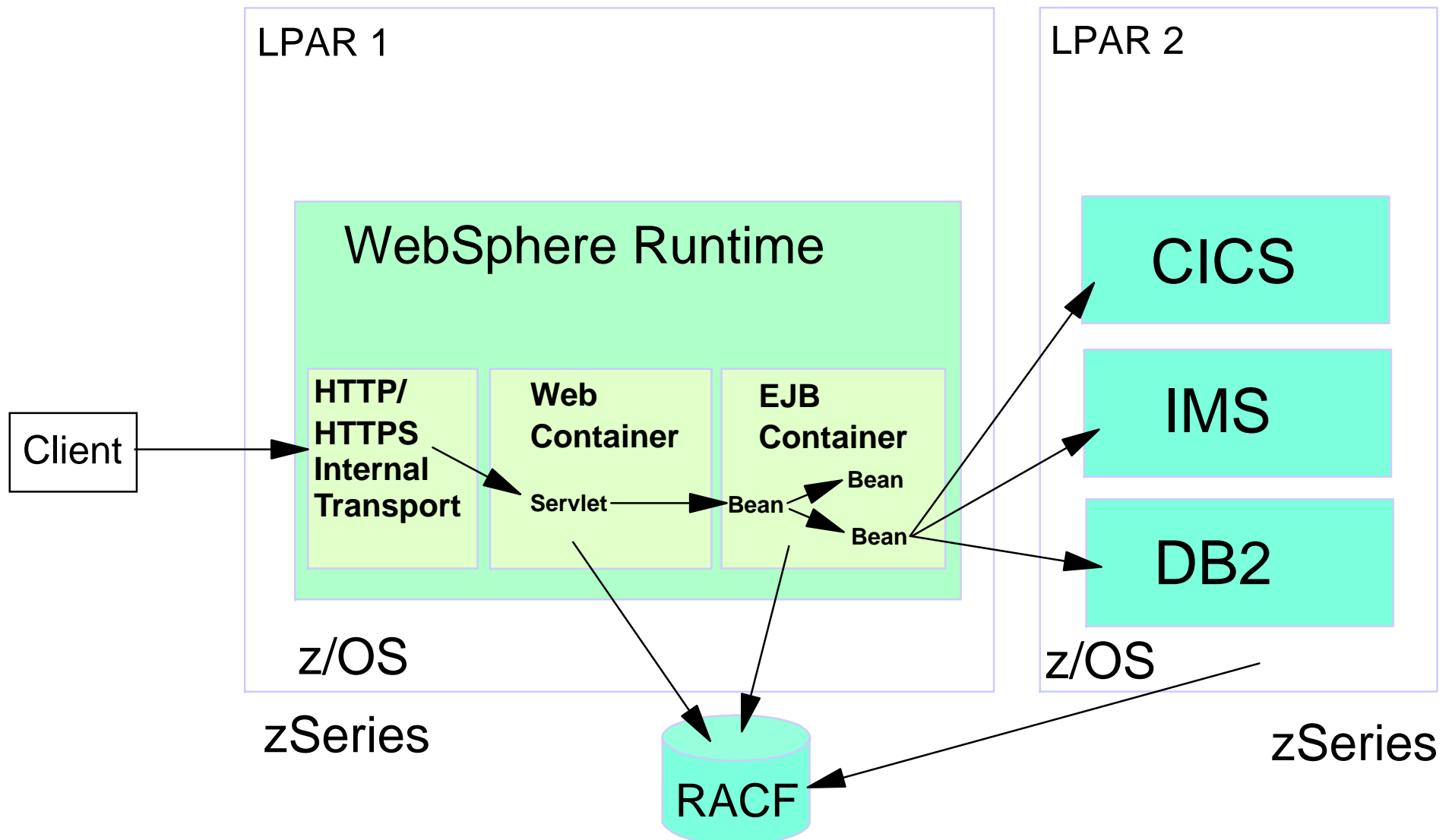
Propagating Identities using Connectors

■ **If WebSphere is remote to CICS, IMS or DB2, you can**

► Define a Userid and Password in an Authentication Alias.
  − Frowned upon by mainframe security auditors.
  − 1 userid/password per Alias = no security granularity.

► Develop applications that perform Client authentication, saving the userid/password and passing it with connection requests.
  − Really frowned upon by mainframe security auditors.
  − Applications shouldn't be handling user passwords.

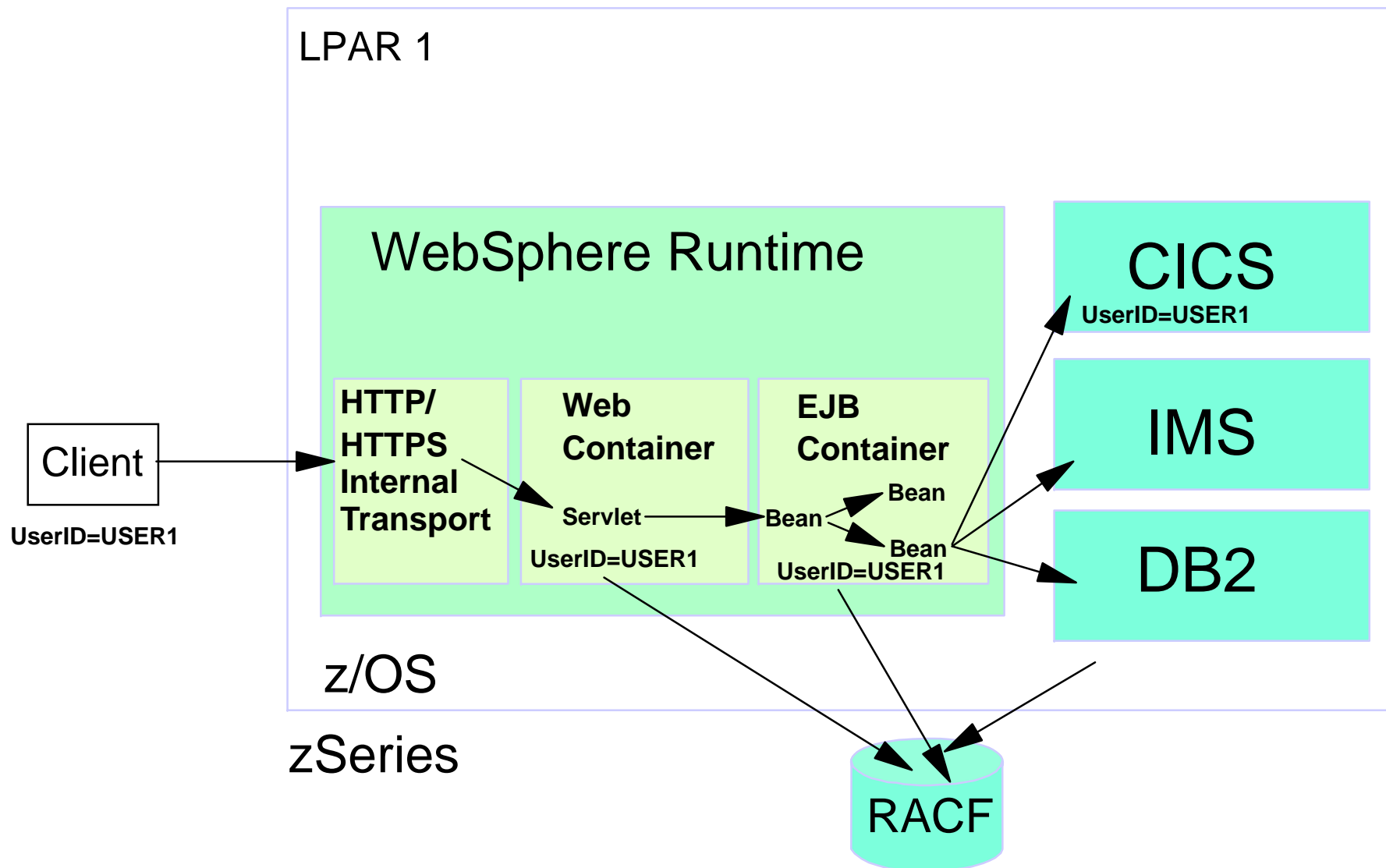# WebSphere V5/V6 'Local' Environment

Propagating Identities using Connectors

- **If WebSphere is local to CICS, IMS or DB2, you can**
  - ► Define a Userid and Password in an Authentication Alias.
    - – Again, frowned upon by mainframe security auditors.
  - ► WebSphere can pass the identity of the Client, the Servant Region, or an identity associated with a role.
    - – No password required.
    - – Enables passing the client's identity to CICS, DB2 or IMS for downstream security checks.
    - – Leverages legacy security protection.
      - • e.g. WebSphere to CICS utilizes MRO security.
    - – Connector request is cross memory, versus cross network.

WebSphere V5/V6 'Local' Environment

Example:

LPAR 1

WebSphere Runtime

HTTP/ HTTPS Internal Transport

Web Container
Servlet
UserID=USER1

EJB Container
Bean
Bean
Bean
UserID=USER1

Client
UserID=USER1

CICS
UserID=USER1

IMS

DB2

z/OS

zSeries

RACF

Summary

- **Putting WebSphere in the same LPAR as CICS, IMS or DB2 provides more options for passing a user identity to the backend server.**
- **This is due to the specialized function found in Type 2 connectors.**
- **Benefits include:**
  - ▶ More Granular Security.
  - ▶ Better Auditing.
  - ▶ Fewer passwords exposed.