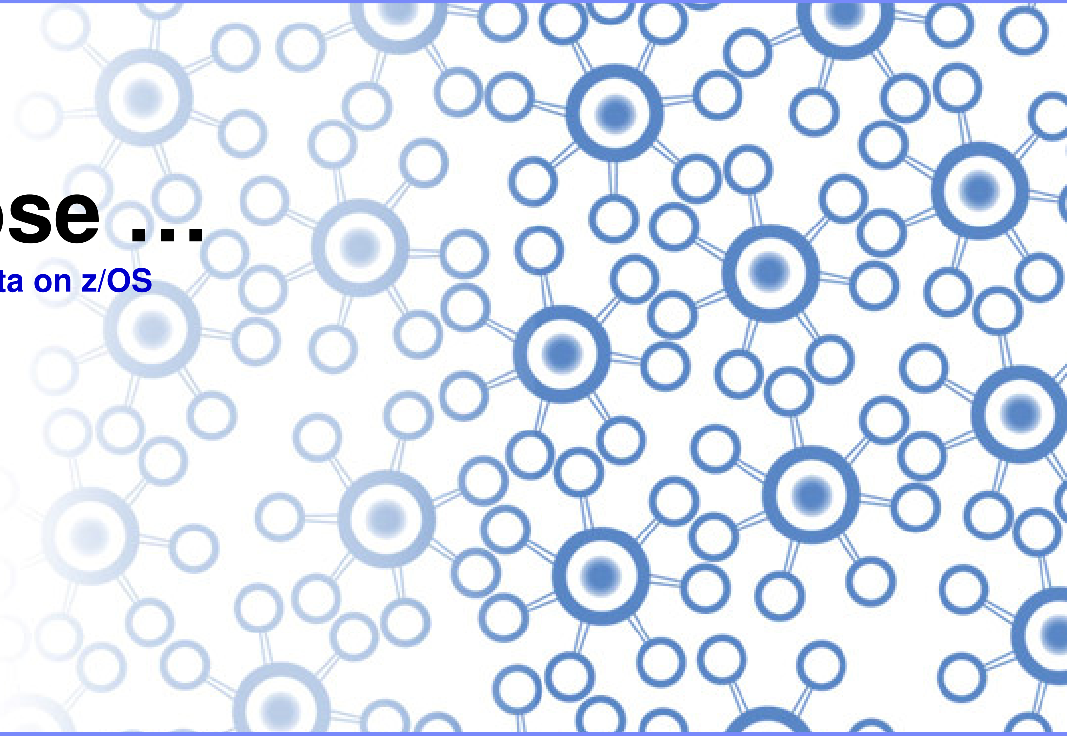
WebSphere Application Server for z/OS:

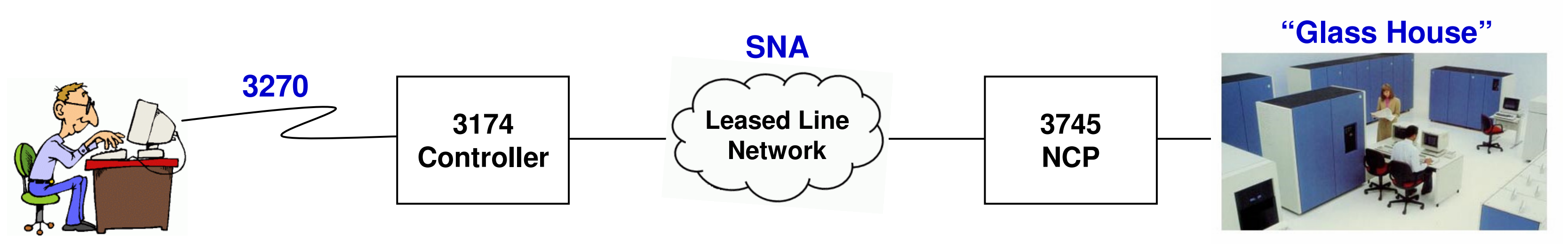
Keeping Things Close …

The practicalities of collocating business logic & data on z/OS



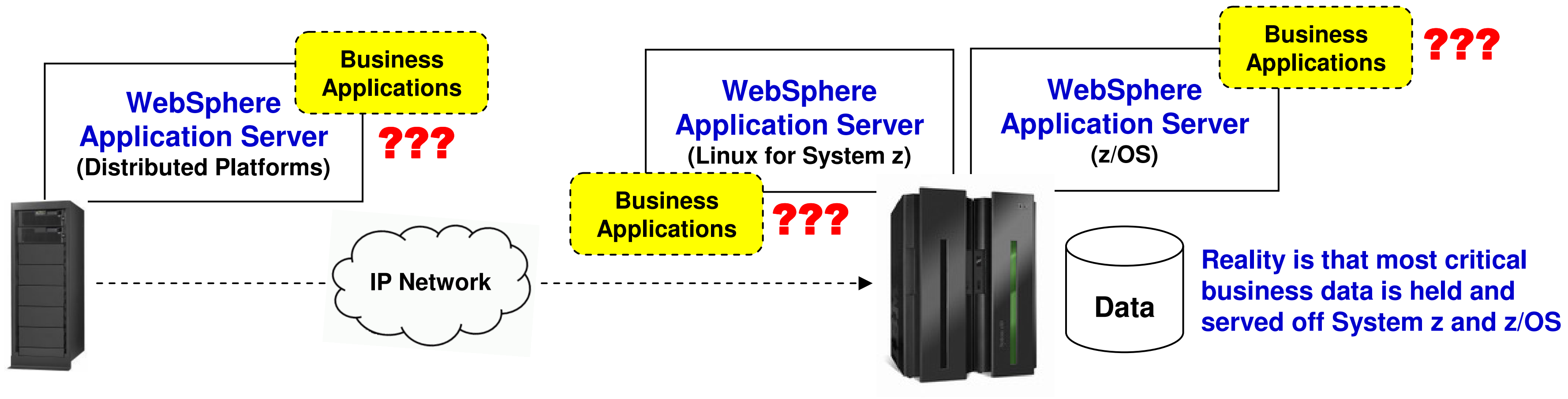
Introduction

In the "old days," things were relatively simple and easy to understand:



This was the world … there was no issue of where to locate the applications vs. the data … it *all* was on the mainframe.

But things are different now … we live in a "distributed" world with servers running on the mainframe and elsewhere.



WebSphere Application Server (Distributed Platforms)
Business Applications
???
IP Network
WebSphere Application Server (Linux for System z)
Business Applications
???
WebSphere Application Server (z/OS)
Business Applications
???
Data

Reality is that most critical business data is held and served off System z and z/OS

Many questions spring from this …

No Shortage of Questions ...

- What does “local” mean in today’s world?
- What do I gain? What do I lose?
- What other evaluation criteria should I consider?

That is what this presentation is all about … to present a framework of evaluation so you can weigh the considerations and make a reasoned decision regarding where to locate the applications relative to the data

Agenda …

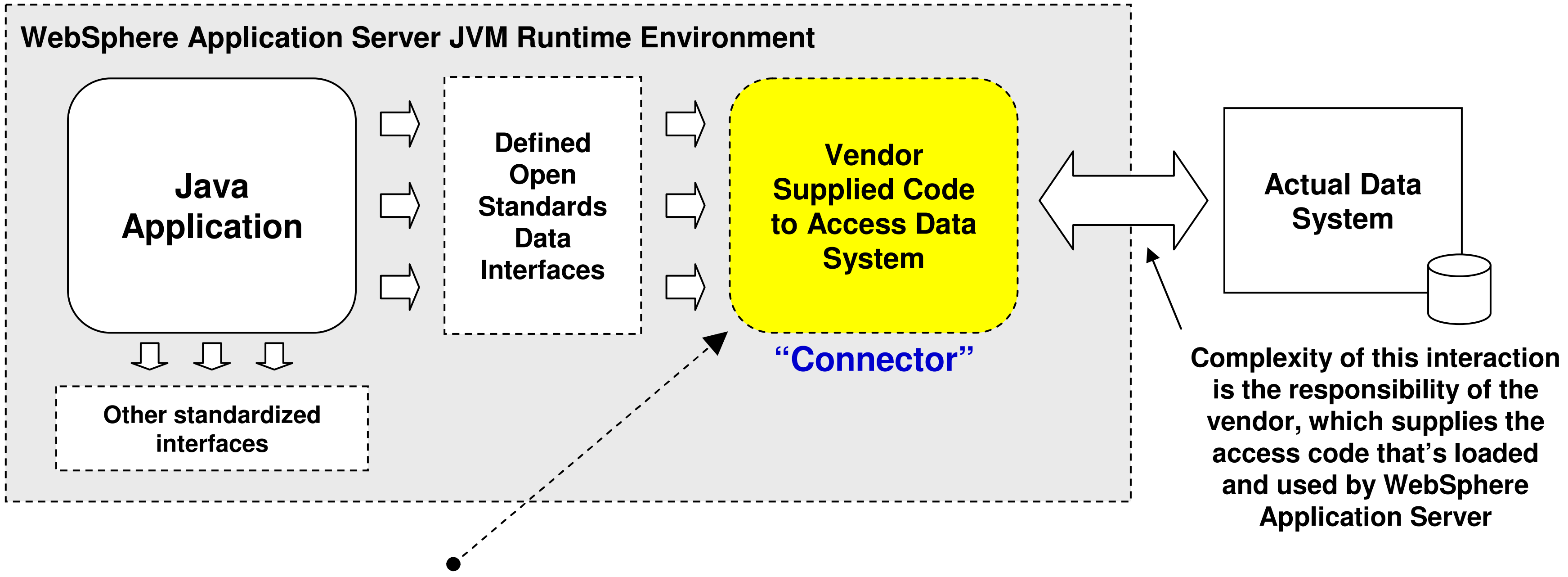
Agenda

- **Connector Considerations**
  Exploring the different connector/connectivity options between WebSphere Application Server and data on z/OS
- **Application Considerations**
  Application-level efficiencies when application and data are in the same operating system image
- **Security Considerations**
  Security efficiencies when application and data are in the same operating system image
- **zIIP and zAAP Considerations**
  The role of the specialty engines based on the connectivity configuration
- **Management Considerations**
  Some advantages of having the application-data system in the same place

Overview of Connecting Applications To Data From WebSphere

Basics of WebSphere's Connector Framework

WebSphere Application Server's design allows for different access code to be used when connecting to different backend data resources:


WebSphere Application Server JVM Runtime Environment
Java Application
Defined Open Standards Data Interfaces
Vendor Supplied Code to Access Data System
"Connector"
Other standardized interfaces
Actual Data System

Complexity of this interaction is the responsibility of the vendor, which supplies the access code that's loaded and used by WebSphere Application Server

This is installed and configured by the WebSphere Application Server Administrator, based on what backend data systems are being utilized

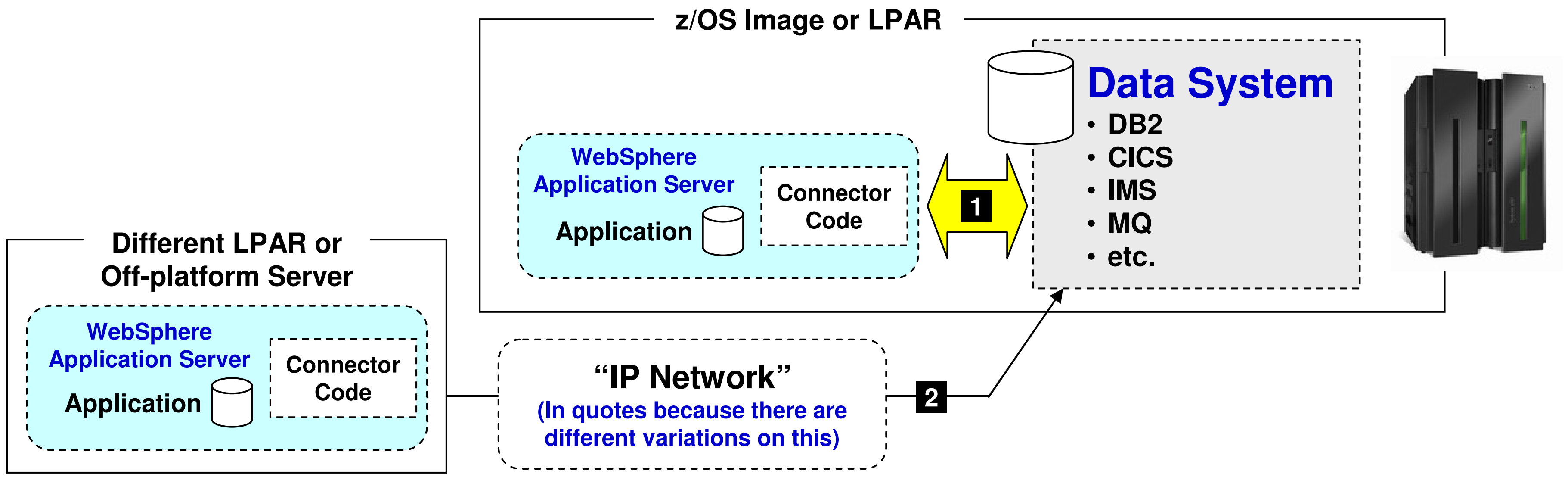
Examples: DB2 and JDBC drivers; or CICS and JCA resource adapter

Let's explore the arrow between the connector and the data system

"Local" vs. "Remote" …

Communications May be “Local” or “Remote”

There are two basic ways in which the vendor connector code can access the backend data system -- cross memory local or “remote” using network



z/OS Image or LPAR
WebSphere Application Server
Application
Connector Code
1
Data System
• DB2
• CICS
• IMS
• MQ
• etc.
Different LPAR or Off-platform Server
WebSphere Application Server
Application
Connector Code
“IP Network”
(In quotes because there are different variations on this)
2

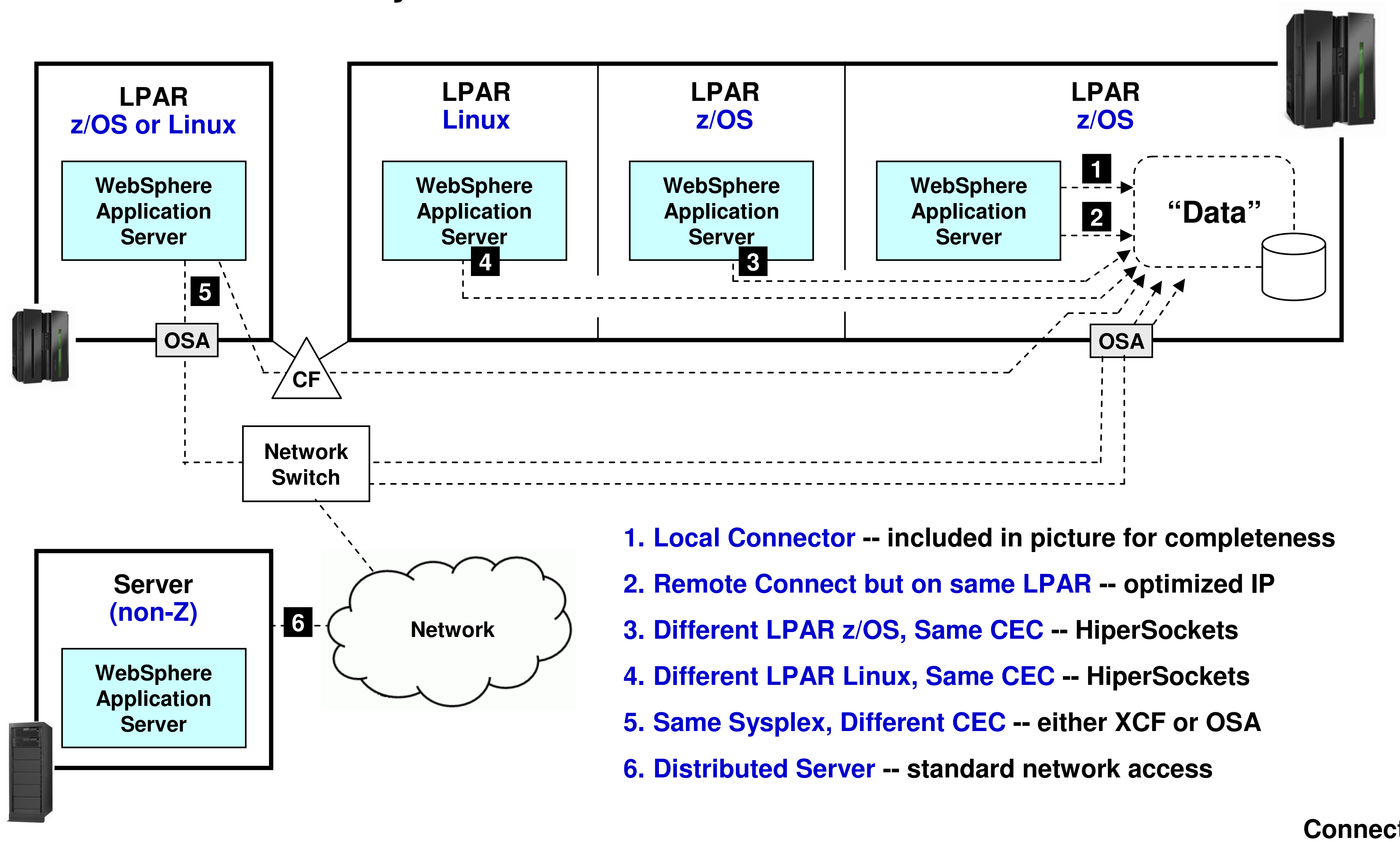
1 “Local” -- uses native code to directly access the data system cross-memory. This is often referred to as “Type 2” though that phrase is really for JDBC connectors. CICS, IMS and MQ have same thing, but they have different labels.

2 “Remote” -- uses Java code to form up TCP/IP requests that flow over the “network” to the data system, where it’s picked up and acted upon. Network is in quotes because, as we’ll see, there are different variations on this that can make a difference.
Note: it is possible to do “remote” even though you’re on the same z/OS instance.

Networks …

Different Variations on "Network"

They all make use of IP, but the degrees of optimization and the inherent bandwidth and latency are different:



LPAR
z/OS or Linux
WebSphere Application Server
LPAR
Linux
WebSphere Application Server
LPAR
z/OS
WebSphere Application Server
LPAR
z/OS
WebSphere Application Server
"Data"
1
2
3
4
5
6
OSA
OSA
CF
Network Switch
Network
Server
(non-Z)
WebSphere Application Server

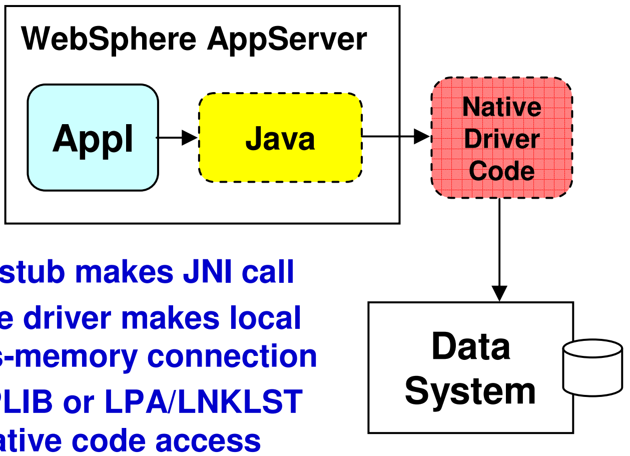
1. Local Connector -- included in picture for completeness
2. Remote Connect but on same LPAR -- optimized IP
3. Different LPAR z/OS, Same CEC -- HiperSockets
4. Different LPAR Linux, Same CEC -- HiperSockets
5. Same Sysplex, Different CEC -- either XCF or OSA
6. Distributed Server -- standard network access

Connectors …

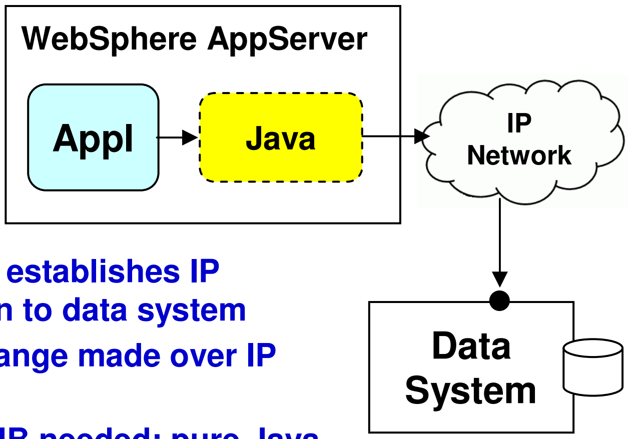
Cross-Memory vs. TCP-based Connectivity

This chart summarizes the terminology used by different data systems for access:

Coding Language	Some Java, Mostly Native Code	Pure Java
Type of Access	Direct cross memory	Network (IP)
JDBC - DB2, other relational	"Type 2"	"Type 4"
JCA - CICS, IMS, other non-relational	"Local"	"Remote"
JMS - MQ when defined as JMS provider	"Bindings Mode"	"Client Mode"



- Java stub makes JNI call
- Native driver makes local cross-memory connection
- STEPLIB or LPA/LNKLST for native code access



- Java code establishes IP connection to data system
- Data exchange made over IP network
- No STEPLIB needed; pure Java

Networks …

The Continuum of Network Access Quality

We've all experienced good network connections and bad network connections. Connections for data access are really no different …

Worst

- On other side of planet from WebSphere
- Using satellite uplink with long latency delays
- Uplink shared with neighbor who downloads high-def DVDs daily
- Awful response times, impacted transaction rates

Better

- On T1 or better link to Internet
- But still through shared public infrastructure (though VPN)
- Several router hops and through firewalls
- Getting better, but things in place that might impact response and tran rates

Better Still

- Inside the data center
- Directly connected to the same network switch as System z
- Very good connectivity

Even Better

- Inside the same LPAR
- Utilizing HiperSockets
- Extraordinary IP connectivity

Best

- Inside the same z/OS image
- Using local connectors
- No IP … direct memory to memory transfer
- Lowest overhead, lowest latency solution

The point is that remote data connectivity can be impaired by sub-optimum bandwidth, throughput and latency

The impact is dependent on the nature of the application

Update on Type 2 support …

Recent Updates to DB2 Type 2 Connector Support

2008 saw delivery of some updates to the DB2 Type 2 connector support to further enhance the performance:

JCC 3.5.1

- **DB2 Version 8 … APAR PK63584**
- **DB2 Version 9 … APAR PK68428**

A bunch of fixes as well as the capability to do "Multi-Row Fetch"

Type 4 connectors had the ability for some time to fetch back multiple rows of a result set in one response, but Type 2 did not. Now Type 2 does as well.

JCC 3.5.2

- **DB2 Version 8 and 9 … APAR PK65069**

More fixes, as well as "pureQuery" support

pureQuery makes it easier to program and access relational data from an object-oriented environment like Java.

Article on pureQuery:

`http://www.ibm.com/developerworks/db2/library/techarticle/dm-0708ahadian/`

Summary …

Checkpoint Summary

- **Two basic data access methods: local cross memory and network based "remote"**
- **Local connectors often referred to as "Type 2" and remote connectors often referred to as "Type 4" … though that language is from the JDBC connector world**
  (CICS, IMS and MQ use different phrases, but concepts the same)
- **Local connectors employ native code; remote are pure Java**
  (Which means with local you have to give WebSphere access to the native libraries -- STEPLIB or LPA/LNKLIST)
- **Not all "IP networks" are the same in terms of bandwidth, latency and overall quality of service**
  (Depending on the application -- the frequency and quantity of data -- that can make a big difference in the perceived performance/quality of the application)

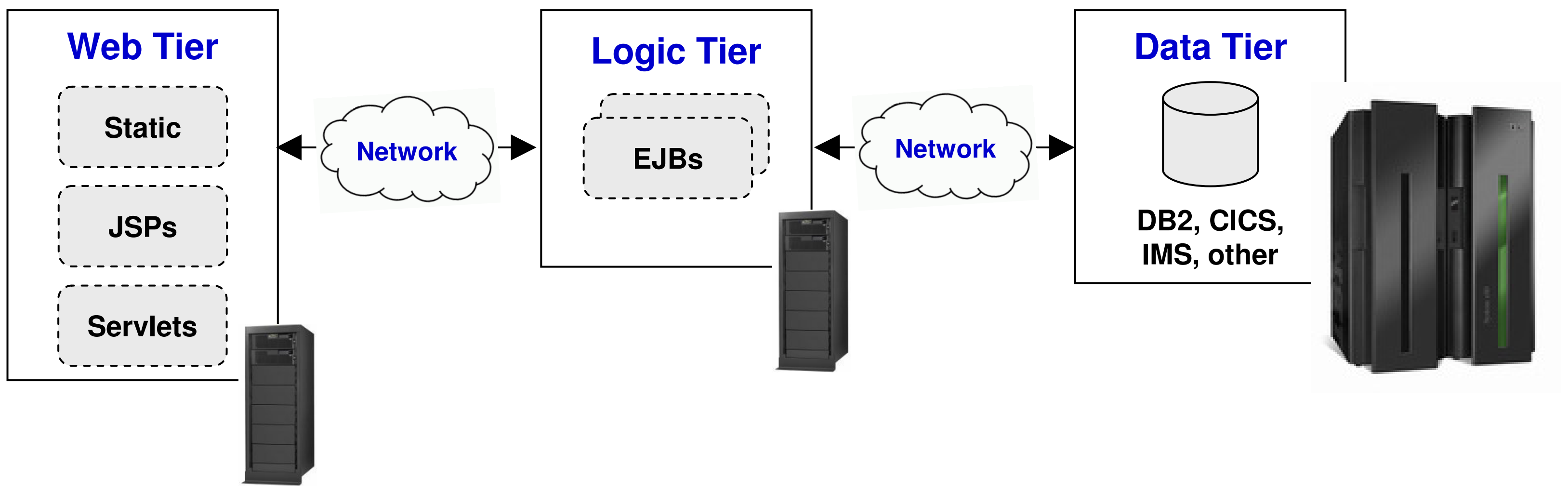
There's more to this story than simple data transfer speed

Application considerations …

Application Considerations

Setting the Context: The Three-Tier Architecture Model

The three-tier architecture model is very common in today's environment:



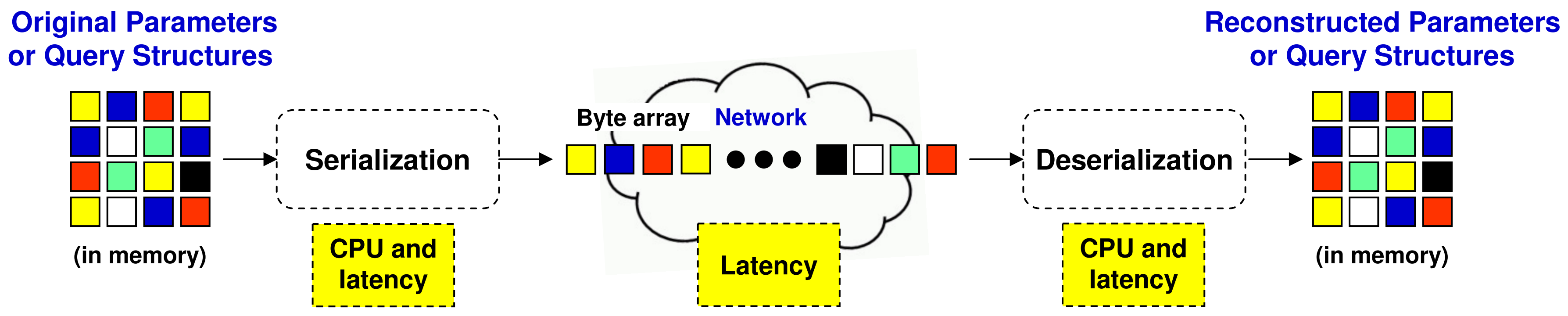
We've already explored the issue of networks

There's another piece to this that isn't commonly known … it has to do with serializing and de-serializing objects and queries/result-sets to pass them across the network

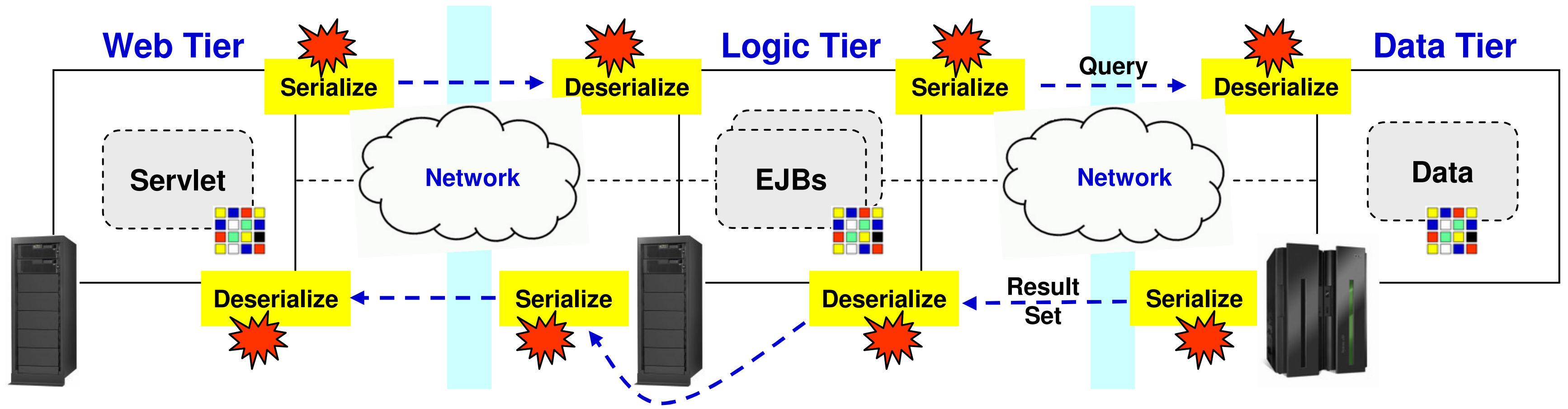
Serialization / de-serialization …

Overview of Object Serialization / Deserialization

In an object-oriented environment, invoking objects across a network involves "serializing" to send across the wire and deserializing on the other side:



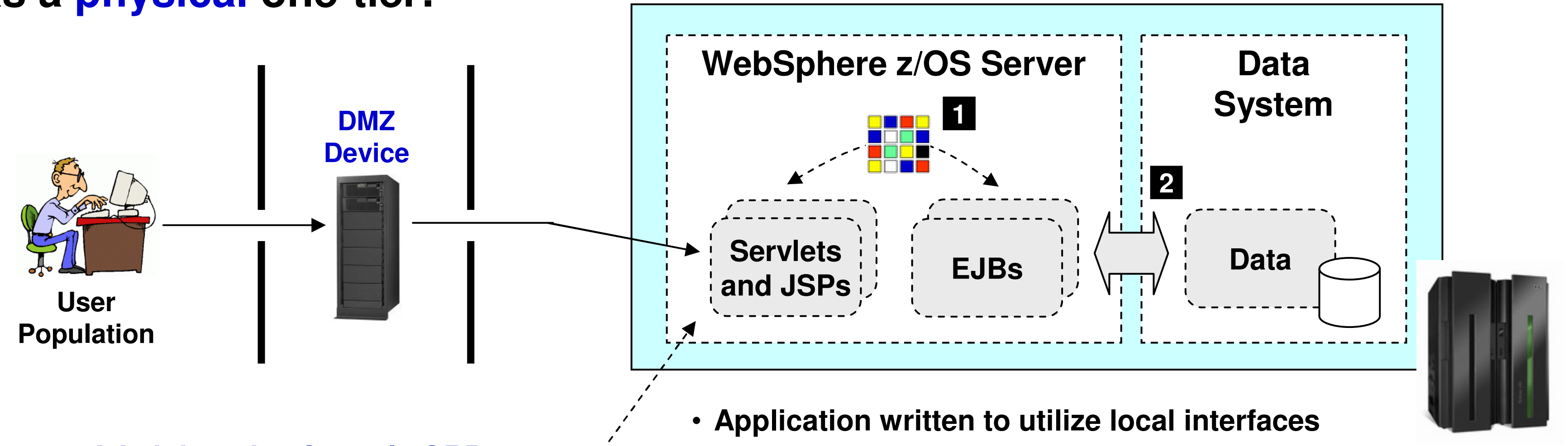
Now consider again the "three-tier" architecture:



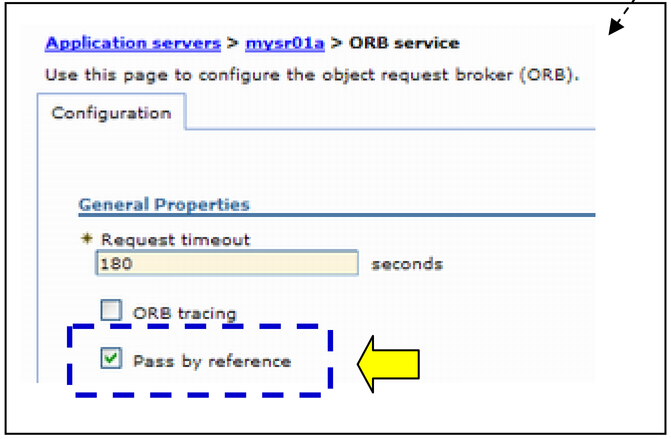
Logical three / physical one …

Collapsing to a Logical Three-Tier Architecture

It's possible to have a logical three-tier architecture, but have it be implemented as a physical one-tier:



Administrative Console ORB setting for "Pass by Reference"


Application servers > mysr01a > ORB service

Use this page to configure the object request broker (ORB).

Configuration

General Properties

* Request timeout
180 seconds

☐ ORB tracing

☑ Pass by reference

1
- Application written to utilize local interfaces
- Container ORB service set to utilize "Pass by Reference"
- Serialization/de-serialization costs removed
- Same thread of execution
- Processing by all components managed to service classification of the servler

2
- Cross memory -- query and result set does not need to be serialized
- Same thread of execution as EJB, which means managed to one WLM goal
- RRS used for two-phase commit, which is more efficient than XA processing
- Security context can be passed (no need for coded aliases)

Summary …

Checkpoint Summary

- **Collapsing to a physical one-tier (but logical three) allows for the elimination of object and query serialization/de-serialization costs**
  The impact can be significant (25% or more) depending on the nature of the application.

- **Avoids network latency as well**
  Depending on the nature of the network between hops, and the frequency and size of the objects being passed back and forth, this could add up to worthwhile savings.

- **One thread of execution -- managed to one WLM goal within a single managed environment**
  This gets to the question of leveraging the single management point value of z/OS

- **Security context can be passed**
  Avoids coding aliases and the exchange of passwords -- both of which are greatly frowned upon by security administrators. More on this in the next section.

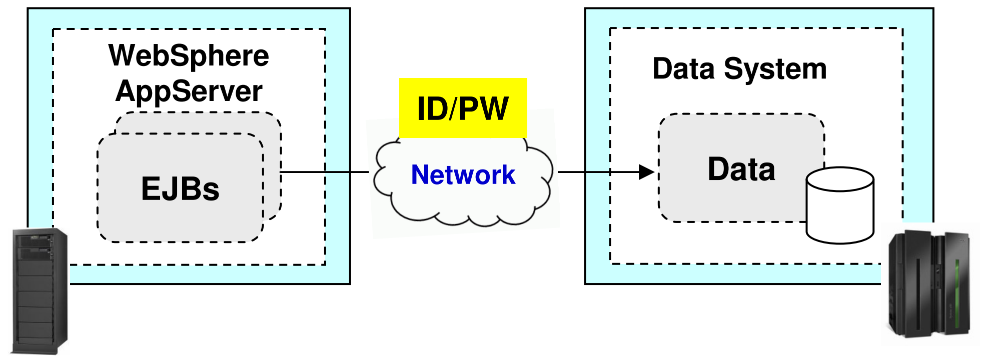
Security considerations …

Security Considerations

Flowing Identity to the Backend Data System

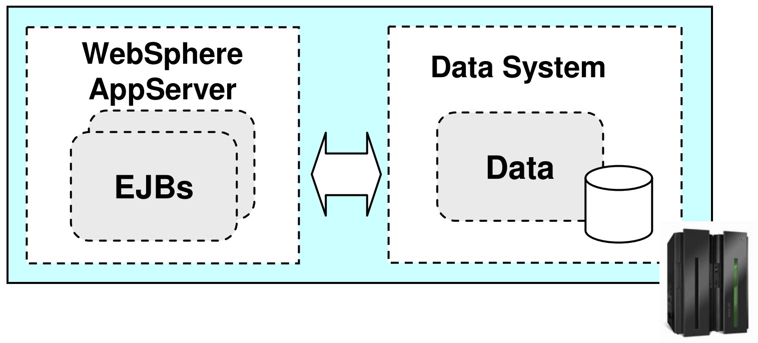
Backend data system require some knowledge of who is asking for the data. How this identity flows depends on the nature of the data connector:

If Remote …



- Flow authentication alias (ID/PW pair) to the data system's listening port
- Some security administrators do not like this:
  - Represents yet another place where userids and passwords are maintained
  - One userid per defined alias, which means ID will likely *not* be the requester ID but a common alias. Reduces granularity of audit.

If Co-Located …



- WebSphere z/OS has more options for passing identity:
  - Client requester, servant region ID, identity associated with a role
  - Data access can be more easily restricted by ID
  - Greater granularity for auditing
- No password is required; no password is transmitted
- Cross-memory as opposed to cross-network
- All flows inside of a trusted environment, which minimizes the security system checking that's required (no password checks needed)

Summary …

Checkpoint Summary

- **Everything is within a single, integrated security domain**
  Better control and security process management

- **Allows avoidance of authentication aliases**
  Which are frowned upon by many security administrators -- they reduce granularity of auditing, and they imply storing userid/password pairs in WebSphere Application Server (encrypted, but it's still "yet another place" where they're maintained)

- **Allows for greater granularity of security access and auditing**
  Authentication aliases typically result in some common ID being flowed back to data systems, where co-locating allows for the flow of the requester ID, the servant region ID, or an ID associated with an EJB role

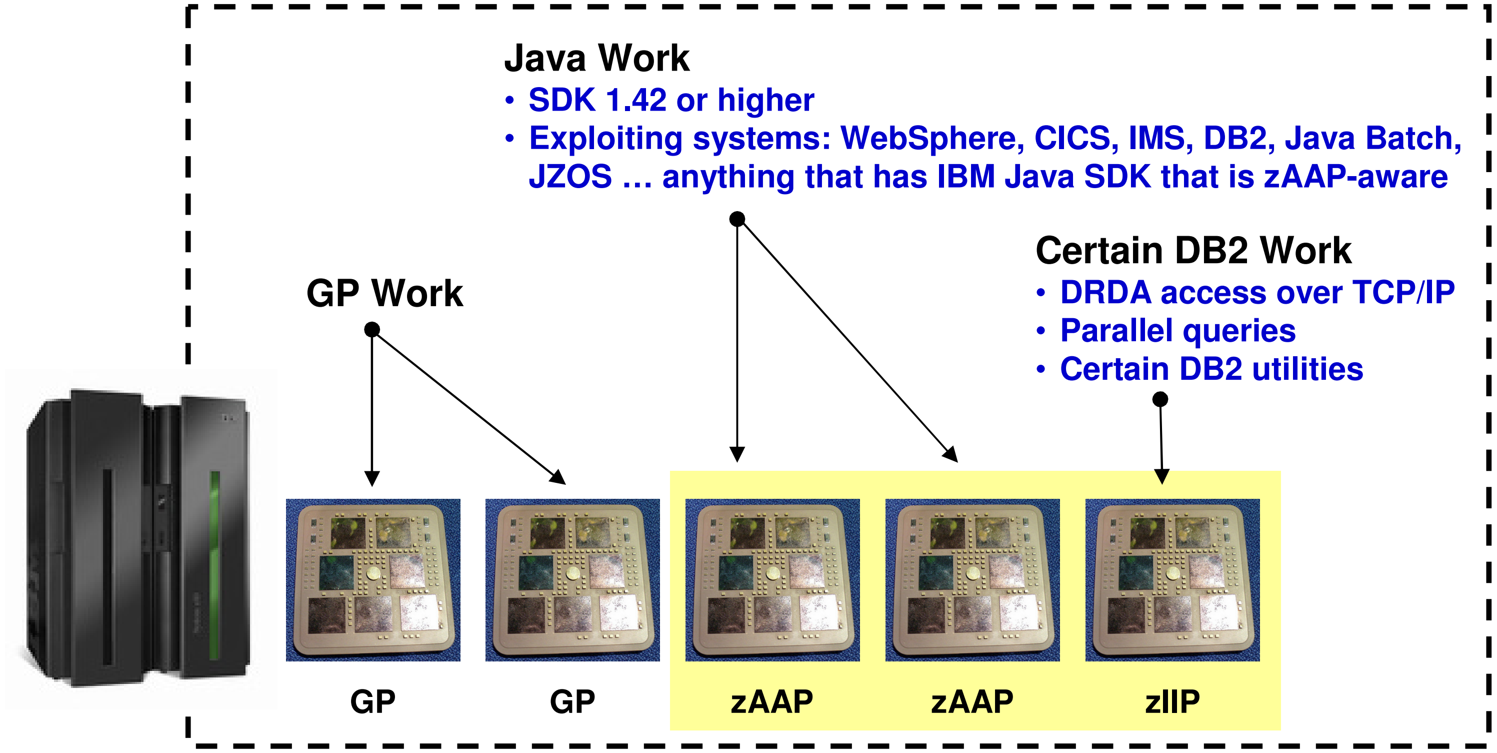
- **Reduces or eliminates the flow of passwords on network wires**
  Generally frowned upon, though encryption in some cases makes this palatable.

zIIPs and zAAPs …

zIIP and zAAP Considerations

Refresher: zAAPs and zIIPs

These "speciality engines" offload certain types of work from the general processors:

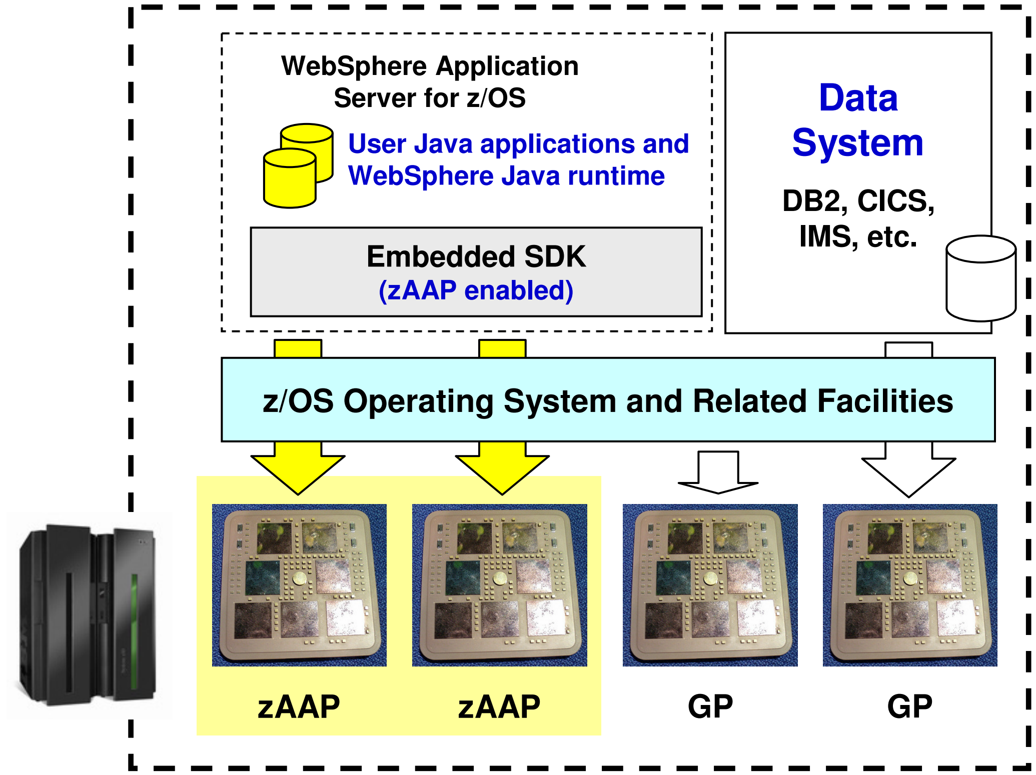

Java Work
• SDK 1.42 or higher
• Exploiting systems: WebSphere, CICS, IMS, DB2, Java Batch, JZOS … anything that has IBM Java SDK that is zAAP-aware
Certain DB2 Work
• DRDA access over TCP/IP
• Parallel queries
• Certain DB2 utilities
GP Work
GP
GP
zAAP
zAAP
zIIP

Advantages: frees GP capacity for critical work; offloads eligible work to value-priced speciality engines; provides processor capacity "hidden" from other sofware license charges

zAAPs …

Offloading Java Work to zAAPs

The cost of bringing Java workload to z/OS can be reduced by offloading that Java work to zAAP processors.



WebSphere Application Server for z/OS
User Java applications and WebSphere Java runtime
Embedded SDK (zAAP enabled)
Data System
DB2, CICS, IMS, etc.
z/OS Operating System and Related Facilities
zAAP
zAAP
GP
GP

Benefits

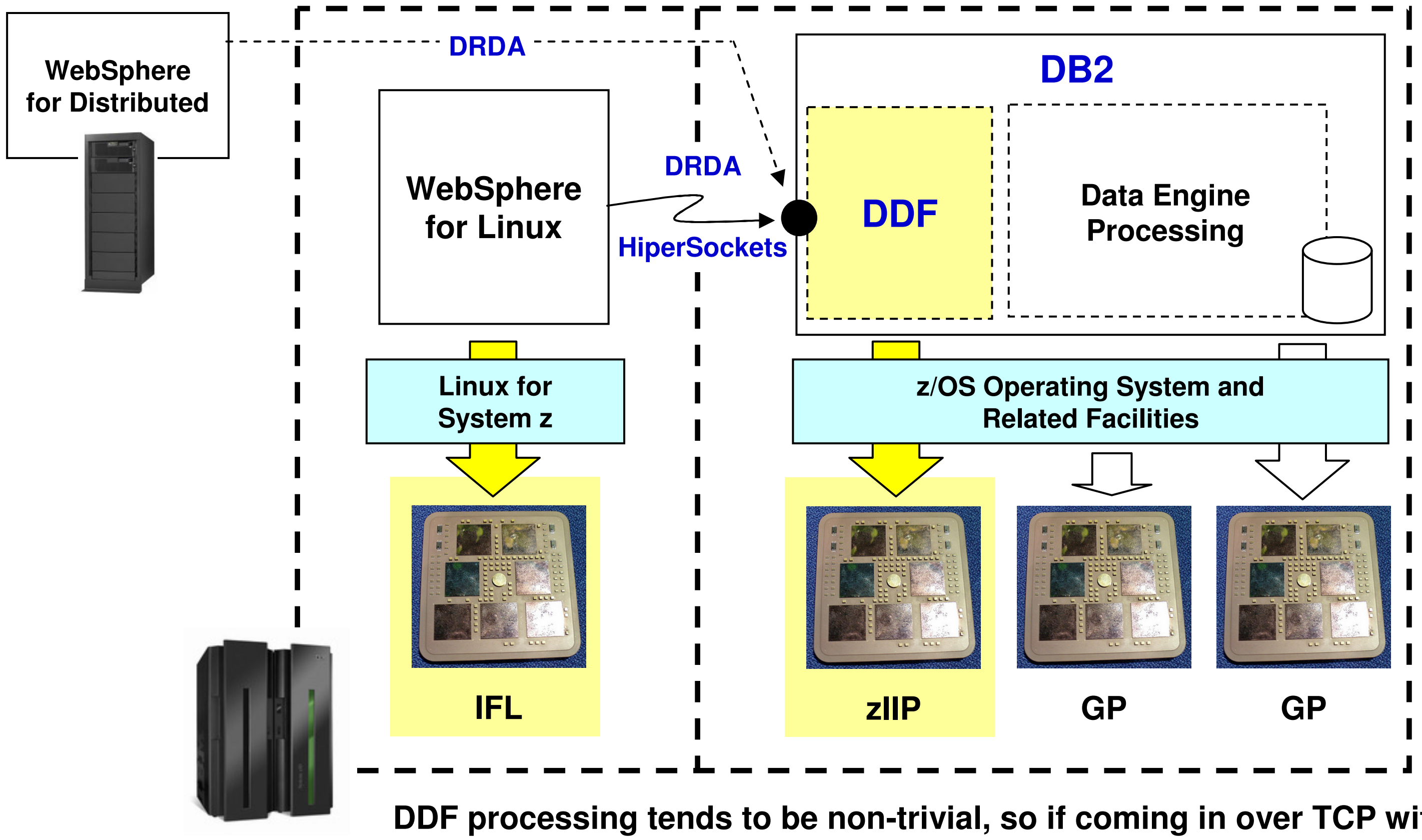
- License manager "sees" the GPs but not the zAAPs. Key software costs do not increase despite processing offloaded to zAAP
- Potentially avoid acquiring additional GPs to handle additional workload. Reduces acquisition costs and software costs
- Expansion of overall capacity at a lower cost/cycle since zAAPs are a lower cost of acquisition

zAAP engines have proven to be very popular and very successful with those customers who have acquired and made use of them

zIIPs …

Offloading DDF Processing to zIIP

If you have remote connection to DB2, that implies cycles spent by DDF to handle the requests. Those can be offloaded to zIIP processors.



Similar statement of benefits as enjoyed by the zAAP

zIIPs and zAAPs can be used together within an LPAR

DDF processing tends to be non-trivial, so if coming in over TCP with Type 4, using zIIP for offload is a good way to avoid GP use for that work

Summary …

Checkpoint Summary

- **zAAP speciality engines help with Java workload on z/OS**
  Which can make justifying bringing WebSphere to z/OS easier.

- **zIIP speciality engines help with certain kinds of DB2 workload**
  Which can help when DDF processing takes place for inbound queries from remote WebSphere servers

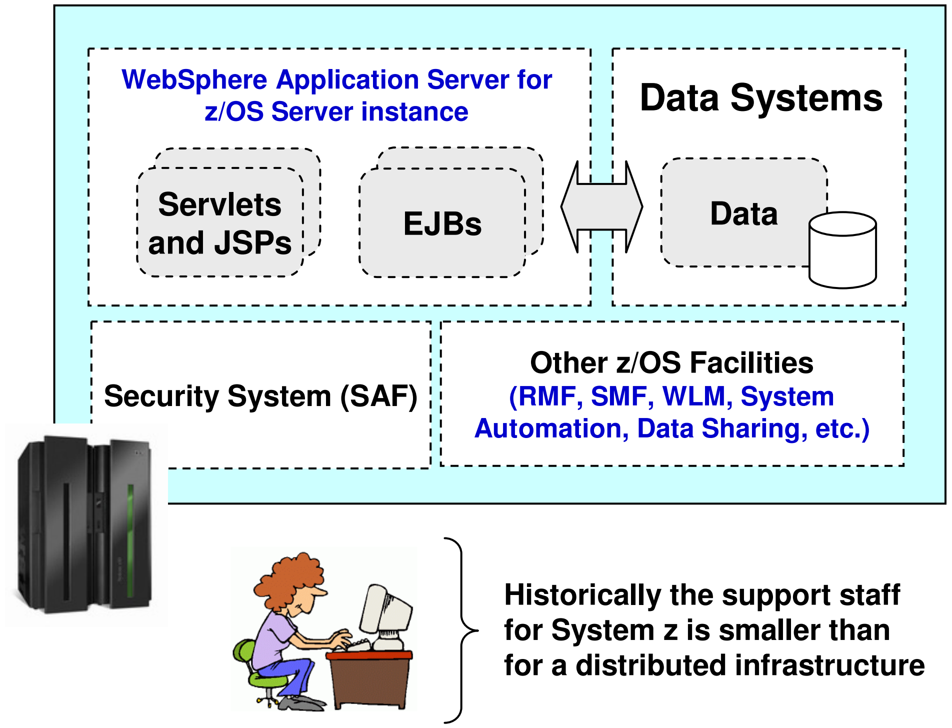
- **Reduced use of General Processors for those workloads means potential for fewer GPs**
  Potential for lower software costs; potential to avoid capital investment in additional GPs to handle growth in processor demands

Management considerations …

Management Considerations

Tighter Management Control; Do More With Less

One of the benefits of an integrated environment is that management of the environment is more controlled and less widely dispersed:


WebSphere Application Server for z/OS Server instance
Servlets and JSPs
EJBs
Data Systems
Data
Security System (SAF)
Other z/OS Facilities
(RMF, SMF, WLM, System Automation, Data Sharing, etc.)
Historically the support staff for System z is smaller than for a distributed infrastructure

- **Better capacity planning**
- **Better accountability and chargeback**
- **WLM helps meet SLAs**
- **Tighter control of security infrastructure**
- **Less downtime due to network component outages**
- **Less controversy over who is responsible for problem determination**

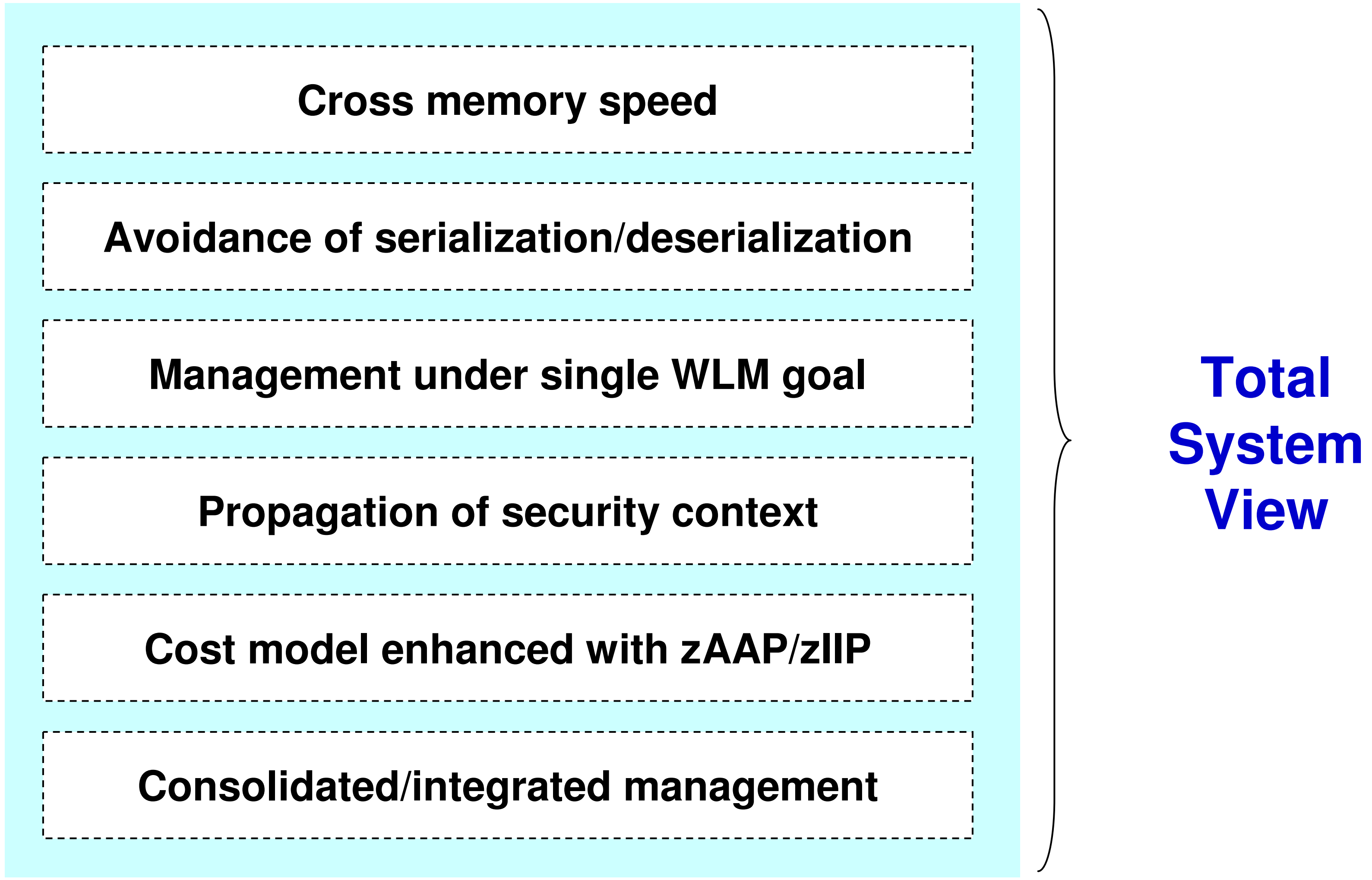
These are often viewed as "soft" values because they're not *directly* expressed in performance numbers … but the value *is* there, and it adds up.

Overall summary …

Overall Summary

A Layered Picture

The co-location story is not simply one of network access speeds, or reduced cycle times. It's a cummulative picture of value adding up to a whole:



Cross memory speed
Avoidance of serialization/deserialization
Management under single WLM goal
Propagation of security context
Cost model enhanced with zAAP/zIIP
Consolidated/integrated management
Total System View