SHARE – Anaheim, CA – February 27 - March 4, 2011

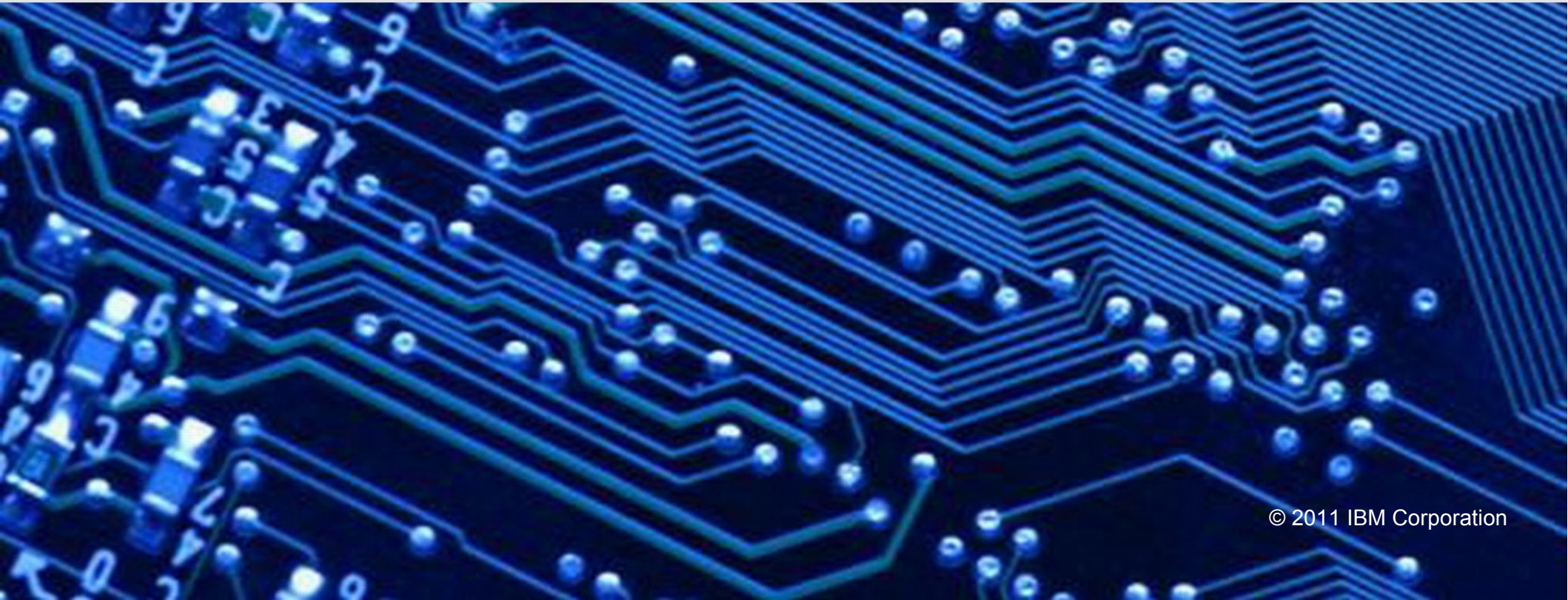
WebSphere Application Server for z/OS

High Availability Considerations

Don Bagwell

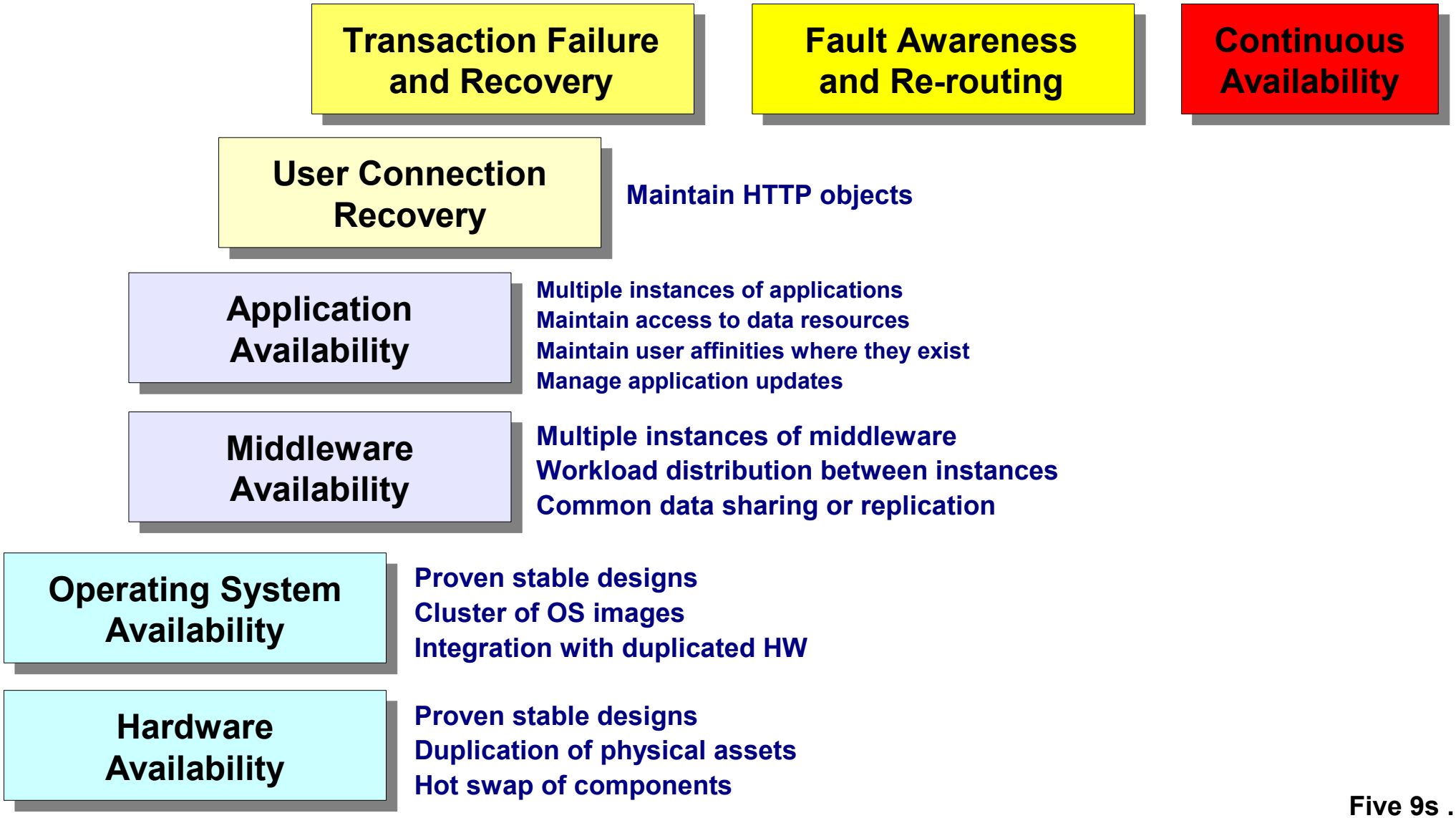
IBM Advanced Technical Skills

dbagwell@us.ibm.com



First Consideration -- What Constitutes an "Outage?"

This is key -- know what you're trying to plan to protect against. This will keep you from under-engineering as well as over-engineering.



Five 9s ...

"Nines to the Right of the Decimal Point"

Just a reminder of about how difficult "100%" is to achieve:

Seconds per year = 60 (Seconds) x 60 (Minutes) x 24 (Hours) x 365 (Days) = 31,536,000

99.00000 %	88 *hours* of downtime per year	1.7 hours / *week*
99.90000 %	9 *hours* of downtime per year	1.5 minutes / *day*
99.99000 %	53 *minutes* of downtime per year	8.7 seconds / day
99.99900 %	5 *minutes* of downtime per year	0.8 seconds / day
99.99990 %	32 *seconds* of downtime per year	2.1 seconds / *hour*
99.99999 %	3 *seconds* of downtime per year	< 2/10^{ths} second / hour

Important questions:

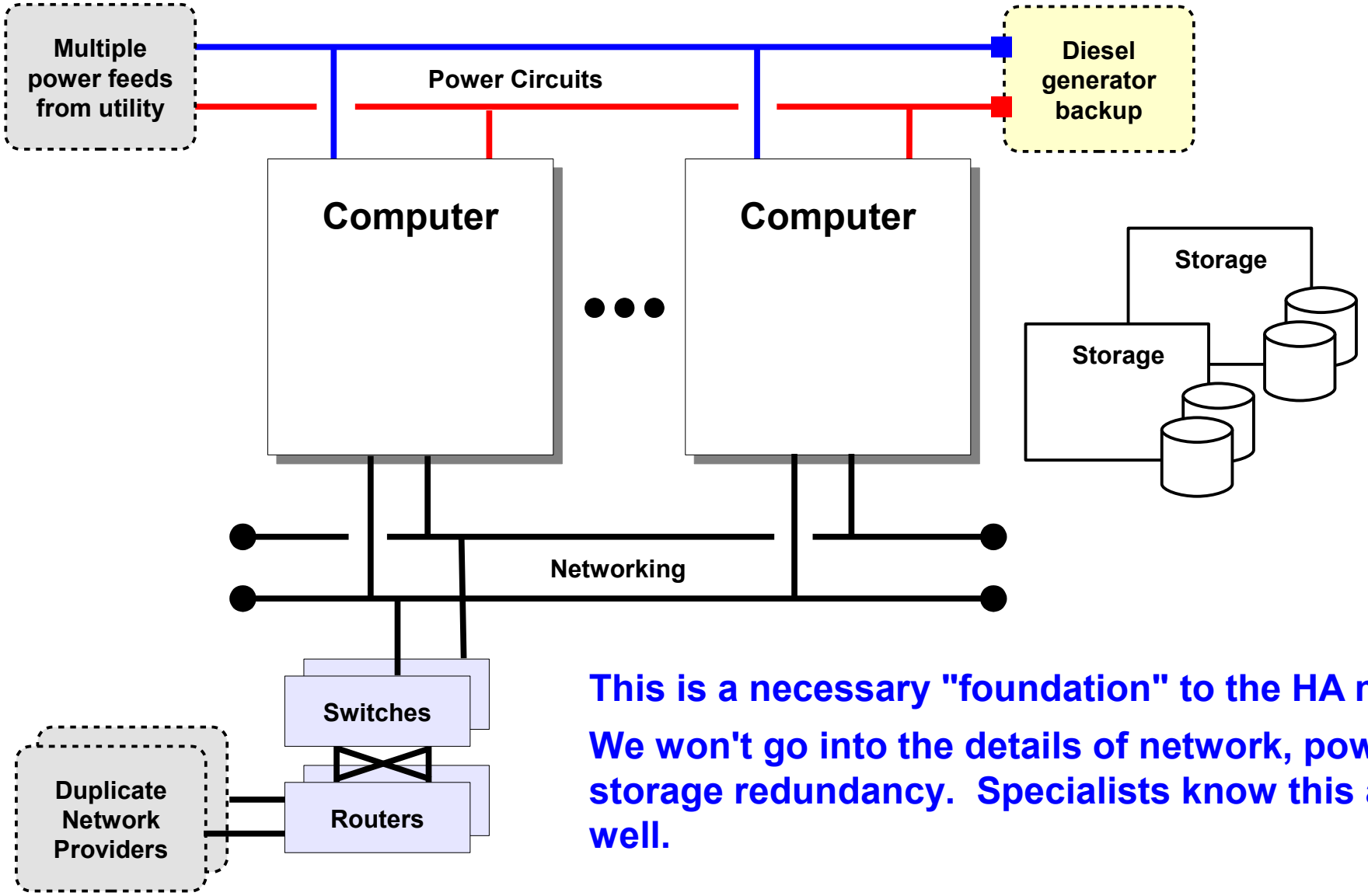
- Again, is everyone in agreement with what an "outage" is?
- What does the business *really* need?
- What are the expectations about *planned* outages?

Redundancy ...

Redundancy and Parallel Sysplex

Eliminating Lower-Level "Single Points of Failure"

This is what most often comes to mind when we think about "High Availability"



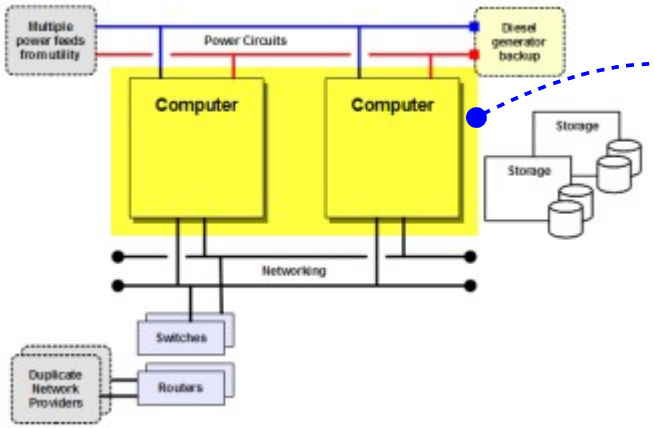
This is a necessary "foundation" to the HA model.

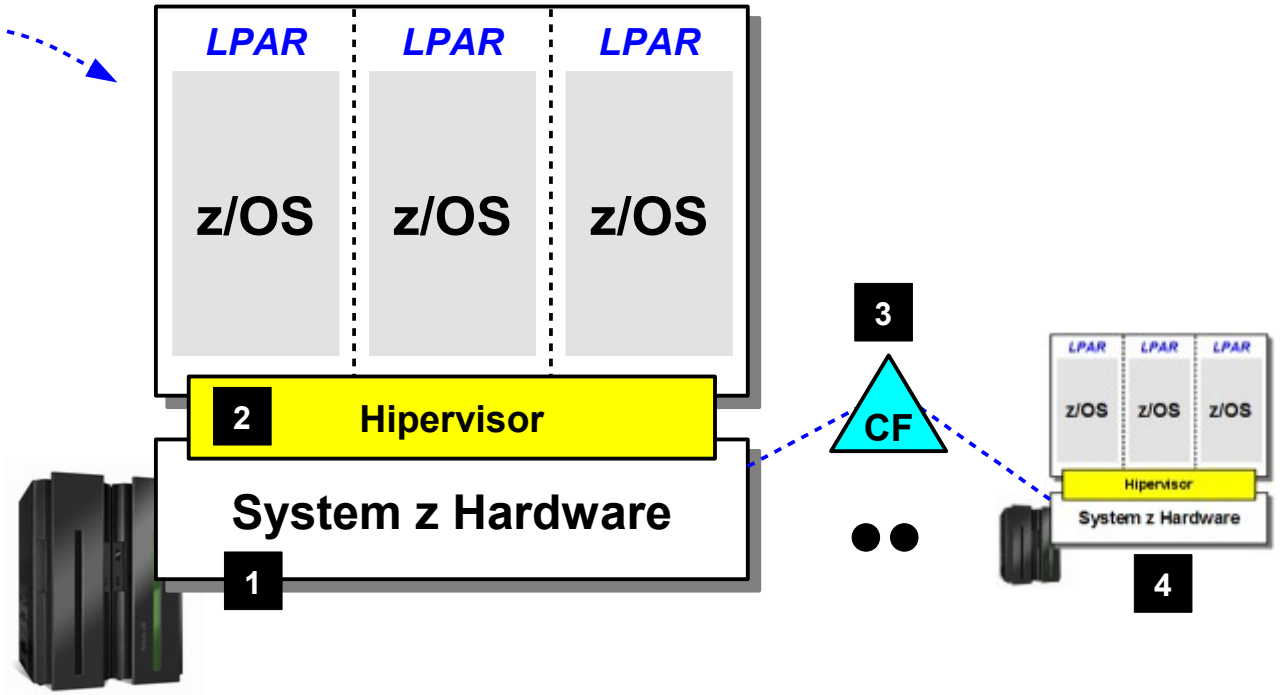
We won't go into the details of network, power and storage redundancy. Specialists know this area quite well.

Parallel Sysplex and z/OS ...

What Does System z and z/OS Bring to the Table

This is where WAS z/OS starts its HA journey ...





1. Hardware

A strong story about designed-in redundancy, hot-swappable components and mean time between failure measured in years.

2. Hipervisor

The virtualization layer that allows multiple logical partitions (LPARs) to be hosted on top of the physical hardware resources. Extremely stable with sophisticated dynamic qualities.

3. Coupling Facility / Parallel Sysplex

This is the heart of the HA story ... this is what provides the shared data, rapid signalling and clustering at the OS level.

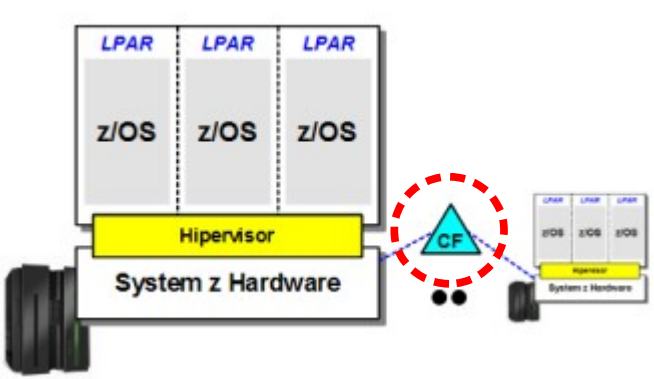
4. Other CECs

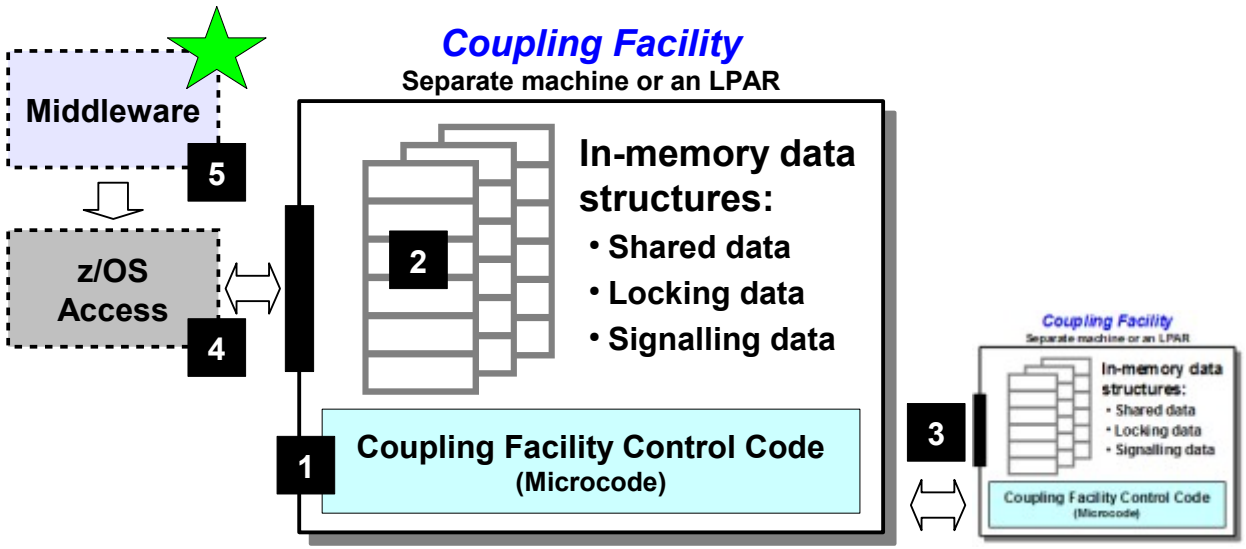
Parallel Sysplex is not limited to a single CEC. Multiple CECs may be joined. The physical distance between the machines in the Sysplex may be expanded to span buildings or cities.

CF and Sysplex ...

High-Level -- The Coupling Facility and Parallel Sysplex

This is where shared data and locking is achieved, as well as providing a very efficient signalling mechanism:





1. Coupling Facility Itself

This can either be its own System z machine or an LPAR. In either case the CF itself runs as single-purpose microcode

2. CF Structures

These are in-memory structures to hold data for the three purposes listed on the chart. This, coupled with the function provided by the CF, is the very heart of the Parallel Sysplex.

3. Duplicated CFs

Is part of the design for avoiding single point of failure of the CF itself.

4. Access for the z/OS Members

The CF provides a high-speed interface for participating z/OS members to access the services and shared data in the CF

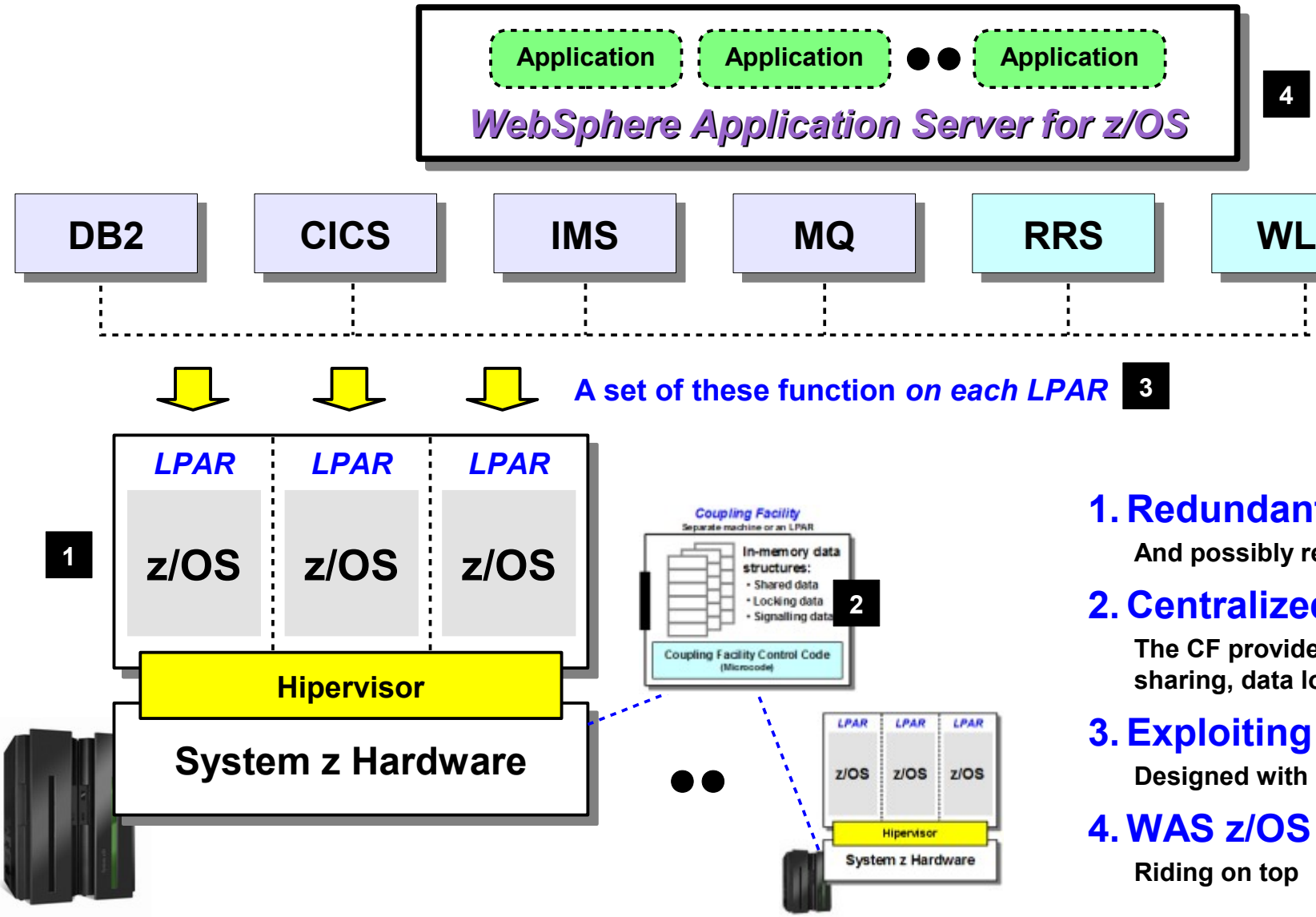
5. Exploitation by Middleware

This is the key ... the underlying Parallel Sysplex environment can be exploited by the higher level middleware, and thus by the application layer as well.

Middleware exploitation ...

z/OS Middleware Components that Exploit the Parallel Sysplex

Let's do a quick survey of the key middleware components that are "Sysplex Aware" ... this then sets the stage for the discussion of WAS z/OS that rides on top:



1. Redundant z/OS images

And possibly redundant CECs as well

2. Centralized Clustering

The CF provides a mechanism for data sharing, data locking and fast signalling

3. Exploiting Middleware

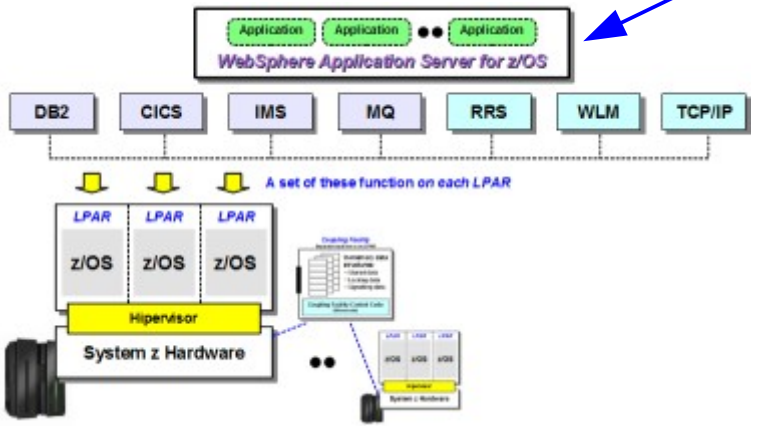
Designed with Parallel Sysplex in mind

4. WAS z/OS

Riding on top

WAS z/OS ...

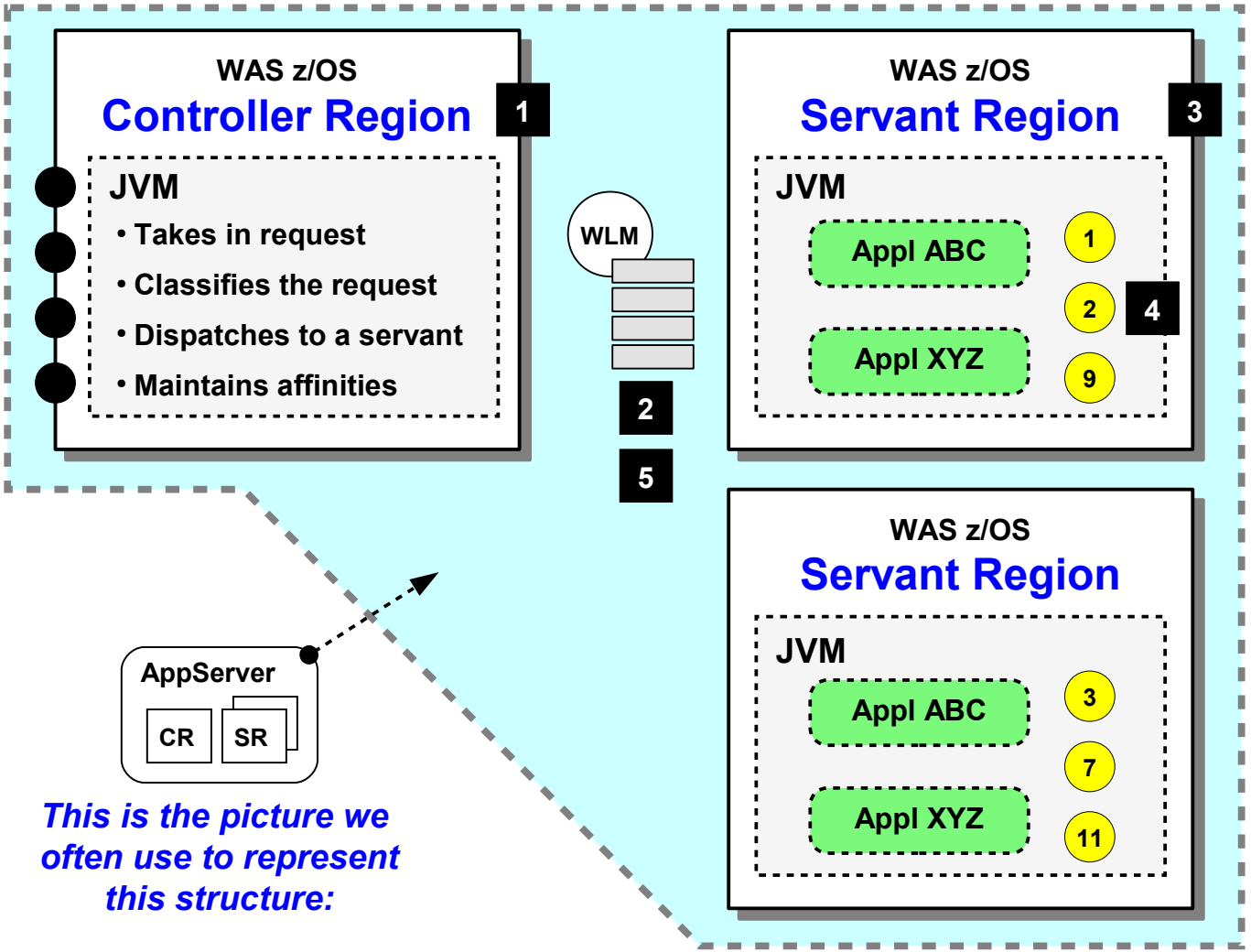
WAS z/OS and HA



The Parallel Sysplex "Foundation" to WAS z/OS HA

The Split JVM Model -- Redundant JVMs Behind the Listener Ports

This is the first line of defense * -- redundant JVMs per application servers provides nearly seamless protection against JVM outages:



1. Controller

Consider this IBM plumbing code. It's primary role is summarized by the bullets to the left

2. WLM Work Queues

The controller makes use of WLM work queues between CR and SR. This provides a way to segment by classification as well as a queuing point to buffer against overruns.

3. Servant(s)

This is where the applications run. Multiple concurrent servants is possible and is what provides the redundancy.

4. User Session Objects

Not replicated to each servant, which means no unnecessary usage of heap. Sessions maintained in z/OS data spaces, so lost servant does not mean lost sessions.

5. Auto Restart

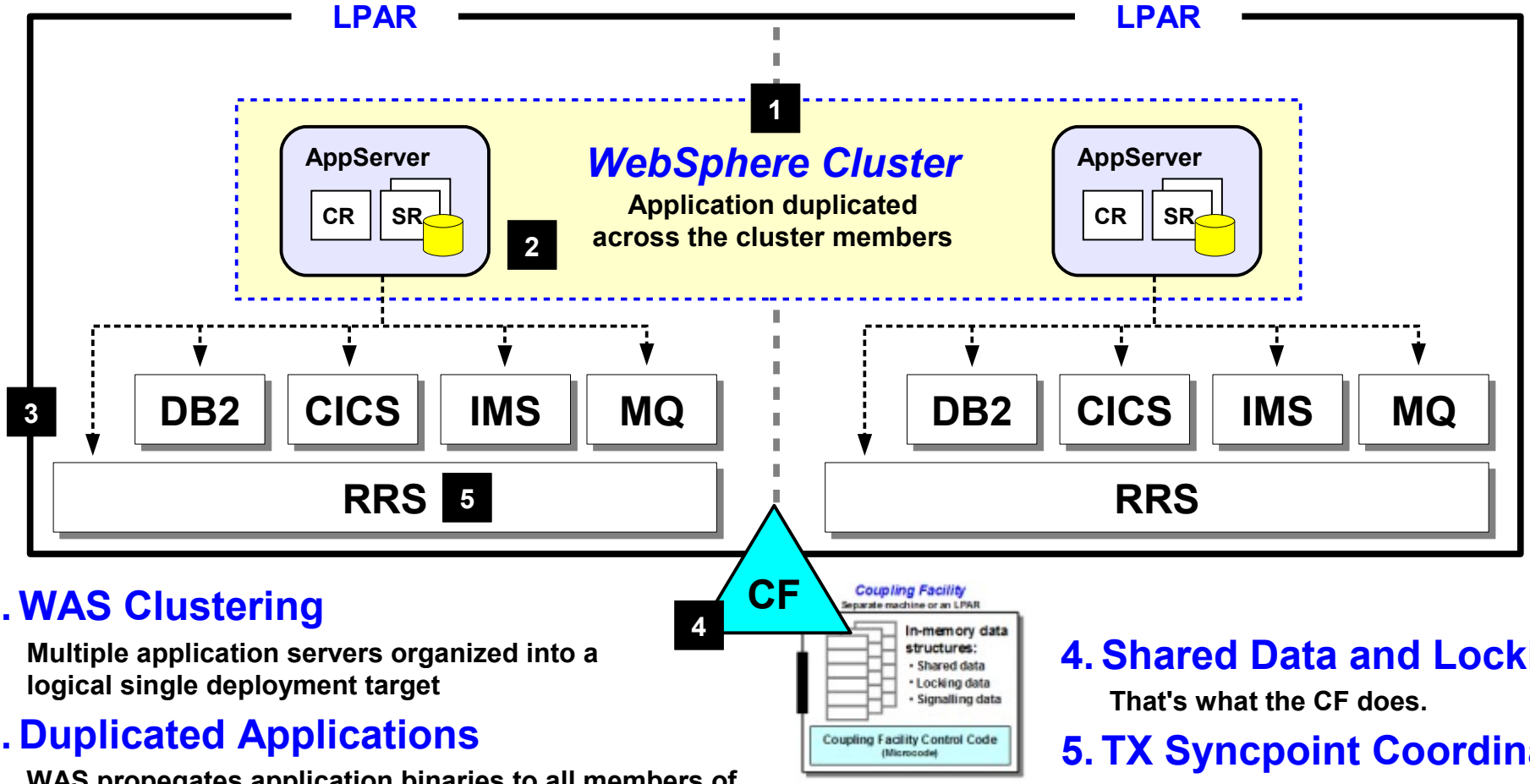
WLM will automatically restart any failed servant regions

* One could say that the System z hardware and the z/OS operating system are the true first lines of defense against outage

Clustering ...

Clustering Across LPARs -- the Second Line of Defense

Clustering is a feature available on all platforms of WAS. The difference is the access to the Sysplex-enabled middleware components:



1. WAS Clustering

Multiple application servers organized into a logical single deployment target

2. Duplicated Applications

WAS propegates application binaries to all members of the cluster.

3. Redundant Sysplex-aware Middleware

Physically separate from other LPARs but sharing common data structures in the CF.

4. Shared Data and Locking

That's what the CF does.

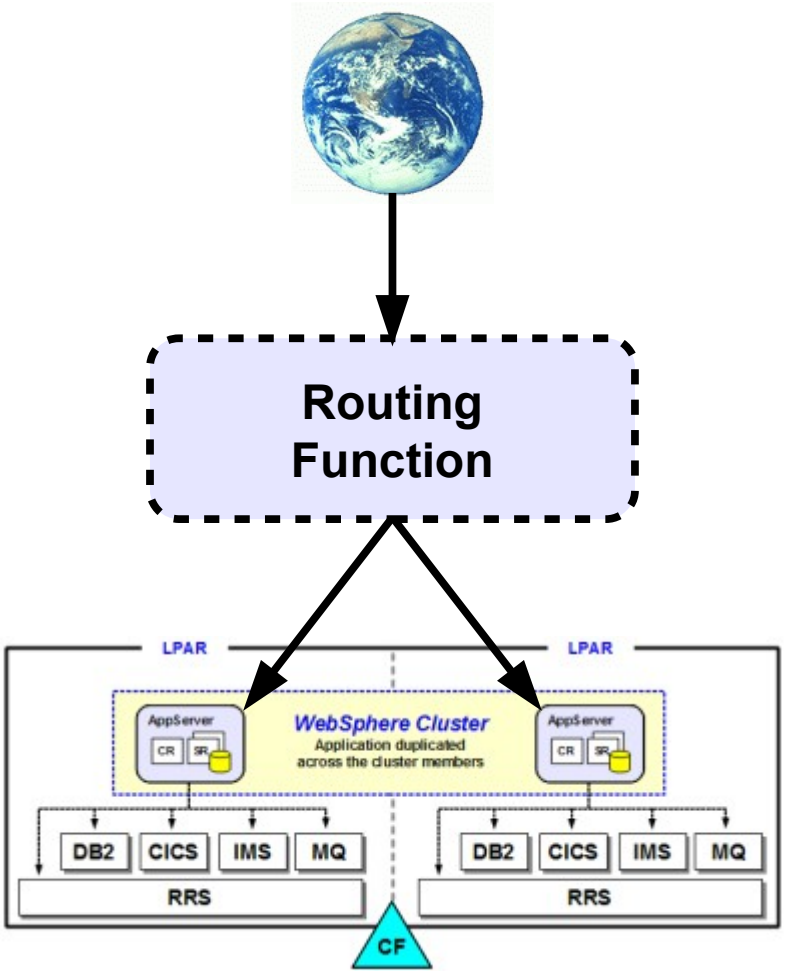
5. TX Syncpoint Coordinator

Resource Recovery Services (RRS) is a transaction syncpoint coordinator. All the major z/OS subsystems make use of it to coordinate 2PC processing.

Routing from outside in ...

Routing from Outside In

WAS cluster members represent physically separate application servers hosting separate TCP ports. So *something* has to be "out front" routing:



There is a great deal to consider in this space:

- **Does it terminate the SSL connection?**
  - This allows for a more flexible routing to the backend.
- **Do server affinities need to be honored?**
  - The most common is affinity based on session object location.
- **Where does this routing function reside -- inside the DMZ or behind the secure firewall?**
  - Inside the DMZ suggests a minimum of protocols and ports punching through the back secure firewall.
- **How much knowledge of the environment do you desire the function to possess?**
  - Server up or down? Or more -- J2EE application status?
- **How intelligent do you want the routing to be?**
  - Three basic levels -- pure round robin; weighted round robin, intelligent placement based on advice.
- **Plus other criteria not listed above**

This is a challenging task -- the "ideal" solution would have *full awareness* of the *everything* and intelligently route around *any* problems it detects.

If such a perfect solution exists, I'm not aware of it.

Brief Summary of Routing Mechanisms

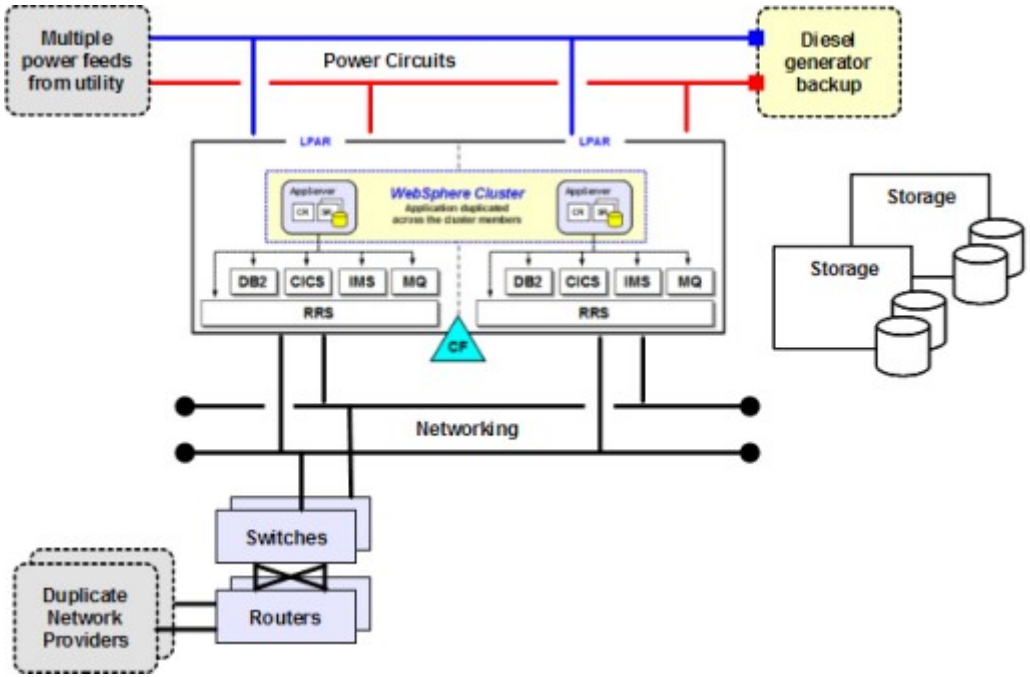
By no means an exhaustive review of all the capabilities:

	SSL Termination	Affinity Routing	DMZ Friendliness	Java EE Awareness	Intelligent Routing
Sysplex Distributor	No	Client yes, session no.	Yes, but few have it out in the DMZ	No	TCP connection placement based on WLM advice
WAS Plugin	Yes	Yes	Yes	No	Weighted round robin
WAS Proxy	Yes	Yes	No	Yes	Yes
WAS Secure Proxy	Yes	Yes	Yes, but lose Java EE awareness	Only if you allow WAS management flows to occur	Yes if aware of environment, weighted round robin if not.
WAS VE ODR	Yes	Yes	No	Yes	Yes; Proxy + more functionality
DataPower	Yes	Not sure	Yes	No	Configurable yes, WLM assist ... not sure
OEM Solutions	Many do	Some do	Yes	Maybe -- not aware of any	Yes, some participate in WLM advice

Combinations of these are possible ... and likely needed to craft robust architecture

We'll revisit this issue when we get to planned maintenance and application updates

Recap of the Foundation HA Structure



We're assuming smart people have worked out the power and storage redundancy

We're assuming the networking folks have done their thing up front and have all the routers and switches in place

We're assuming all the normal Parallel Sysplex best practices are put in place with respect to redundancy and availability

We saw how WAS z/OS provides a kind of "vertical clustering" with the split JVM model

We spoke of the challenge of request routing

Now it's time to focus in on the application and what it's doing

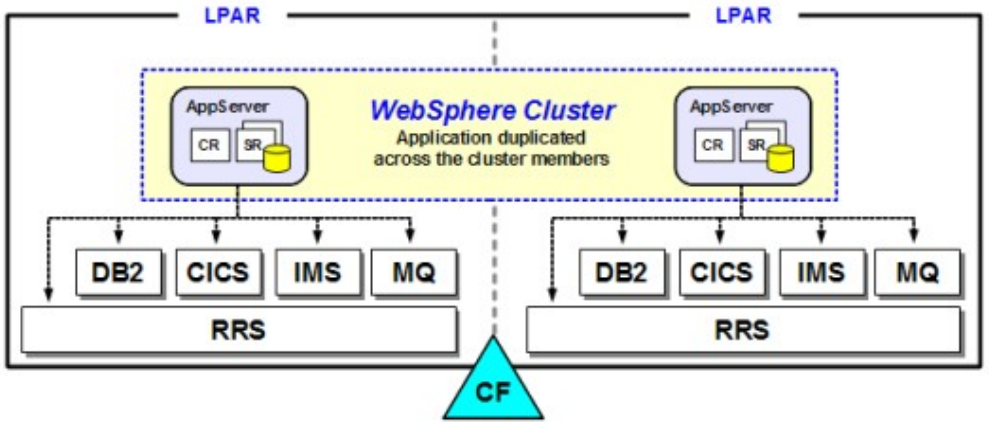
Application layer ...

Application Layer HA

Including data access topologies

Four Topics to Consider

They are ...



Server Affinities

The most common are the result of creating HTTP session objects.

Affinity restricts request routing flexibility

Data Access Approaches

It's a trade-off ... the benefits of co-location vs. the flexibility of IP-based re-routing.

Other Application Dependencies

The application may be up and accessible but not "working" because of some other element in the design not available.

What do you do about this other than keep close tabs on the relationships and enforce careful change control?

Transactions and Transaction Recovery

WAS has a mechanism to roll back in-doubt transactions. RRS is central to this.

You can start to see why that third, fourth and fifth nine past the decimal point becomes increasingly difficult

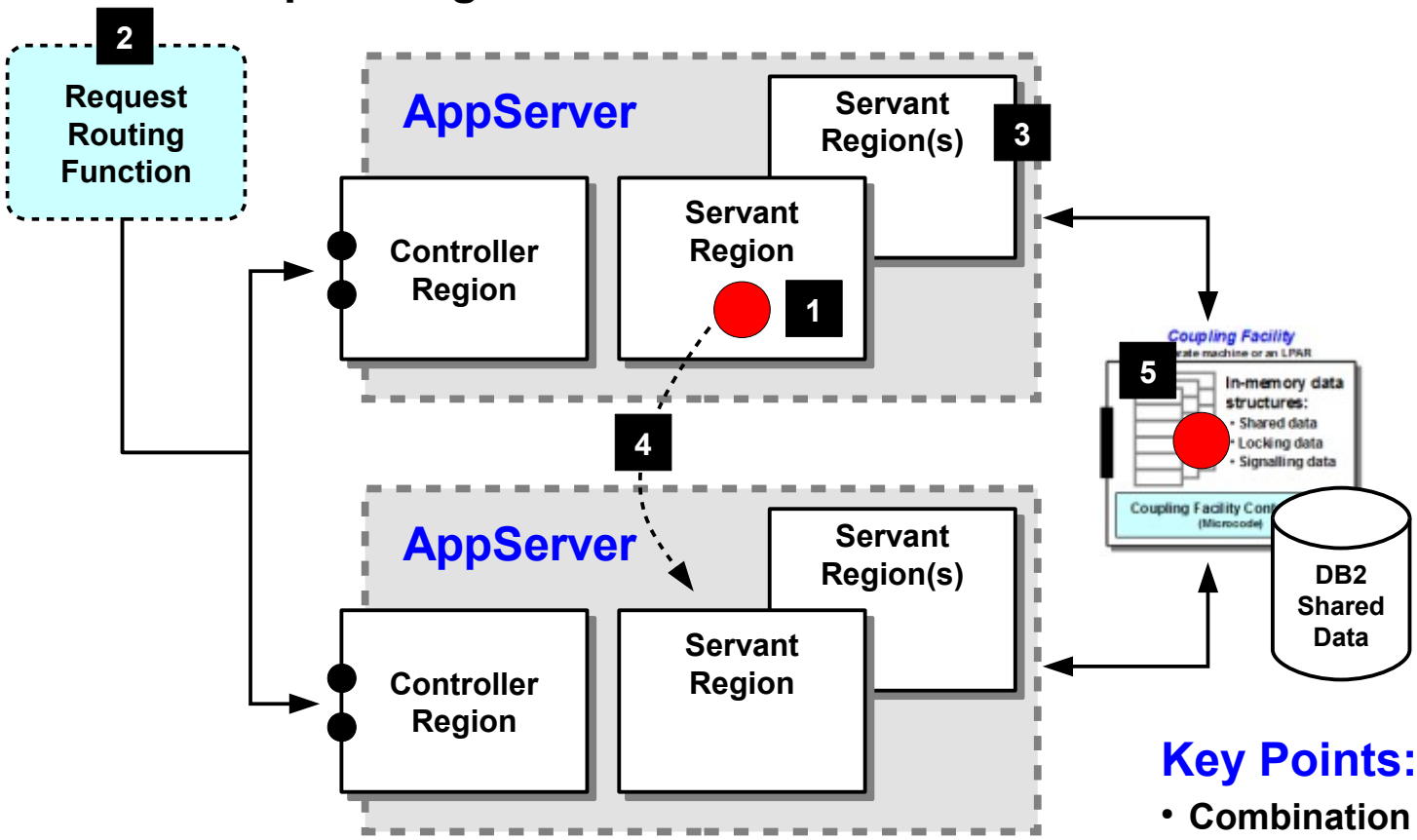
This is also why it's really important to lock down the definition of what constitutes an outage

It may well be that user access to a *part* of the functionality is acceptable

Server affinities ...

Server Affinities

The most common is HTTP session affinities, which are used to hold transient data. But without planning can create a need to route users back to the initial server



1. Session Object

Created by application if designed to do so. It is a data object in JVM memory

2. Affinity Routing

One option is to provide affinity routing. WAS Plugin does this, as does Proxy. Sysplex Distributor does not

3. Within Appserver

If the servant in which a user object resides goes down, WAS z/OS automatically takes care of that and places user into a surviving servant

4. Replication Domain

A feature of WAS across platforms. This serializes the object and copies it over the network to other servers in the defined domain.

5. Session Persistence

Object persisted to DB2 and fetched back as needed

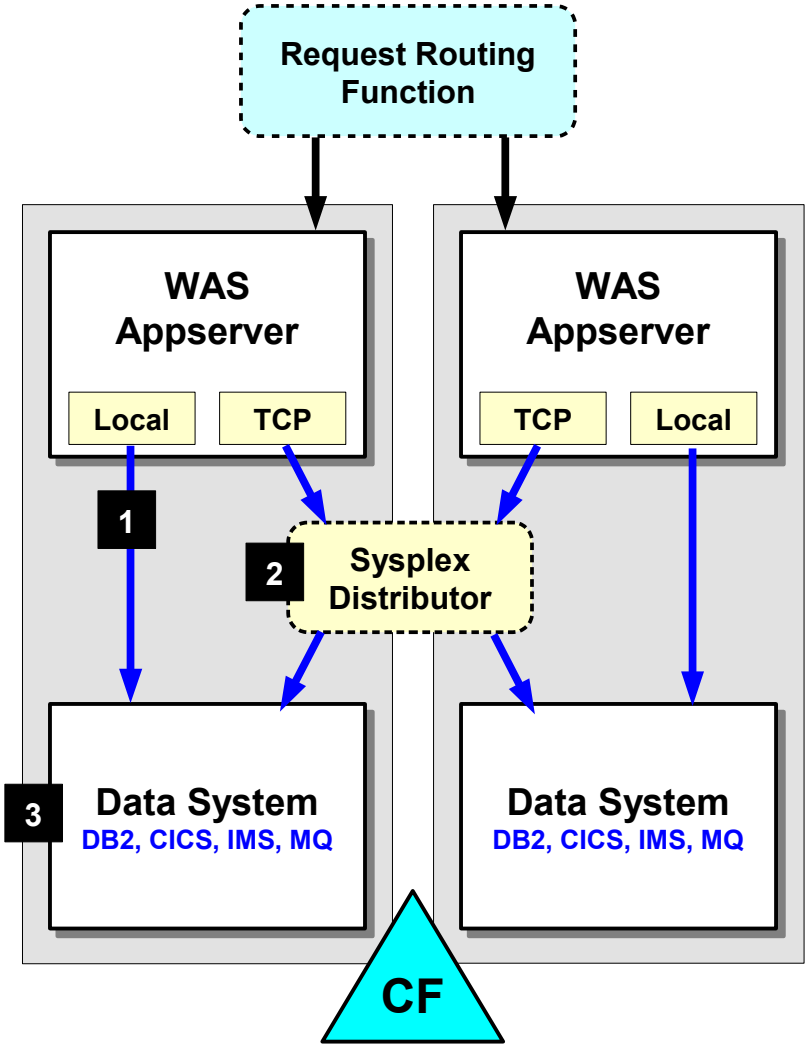
Key Points:

- Combination of affinity routing and replication or persistence is common
- On z/OS persistence performs as well or better than replication
- Create affinities only where necessary; be careful of object size

T2 / T4 debate ...

Data Access Approaches - The T2 vs. T4 Debate

You gain a degree of flexibility with an TCP-based connection but lose some of the advantages of a local cross-memory connection:



1. Local Connectors

Uses the cross-memory native interfaces. Available for DB2, CICS, IMS and MQ.

Advantages: Speed, avoid serialization, assert identity, single thread of execution, propegate enclave for DB2

Disadvantages: Loss of data system means application has no access to data unless alternative connections are made available. Routing function may not know backend data system is gone.

2. TCP-based Connectors

Uses the TCP network to flow requests to target listener. Available for DB2, CICS, IMS and MQ.

Advantages: Loss of TCP connection typically signals retry; SD will connect to surviving member. DB2 T4 takes this even further.

Disadvantages: Potential loss in performance, generally implies alias ID and PW.

Data locks may exist ... other work may proceed but work related to held data can not until failure subsystem restarted so locks can be freed.

3. What is Being Protected Against?

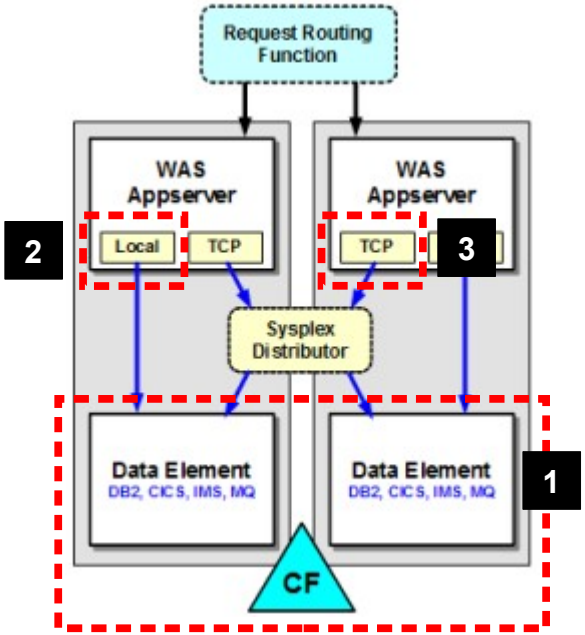
If you are concerned about loss of data system while WAS server stays up, then the TCP based with intermediate routing is a consideration.

If your primary concern is loss of the LPAR "tower" then the use of TCP connectors becomes less important.

Variations ...

Variations on the Connector Theme

A few things to be aware of:



1. CICSPlex

It is common to have WAS connect to a TOR, and have CICPlex handle routing to best AOR hosting program.

Provided local TOR stays up you have local access along with intelligent routing within CICS

2. MQ Connectors in WAS V7

It has a "Bindings, then Client" option -- if local QMGR not available, then go client to named host:port. That can be Sysplex Distributor DVIPA.

Advantages of local when it's present; availability via distributed client mode when not.

3. JDBC Type 4 Connector for z/OS

Two things:

a) TCP to target on same LPAR uses "fast local sockets" -- avoids IP layer code path.

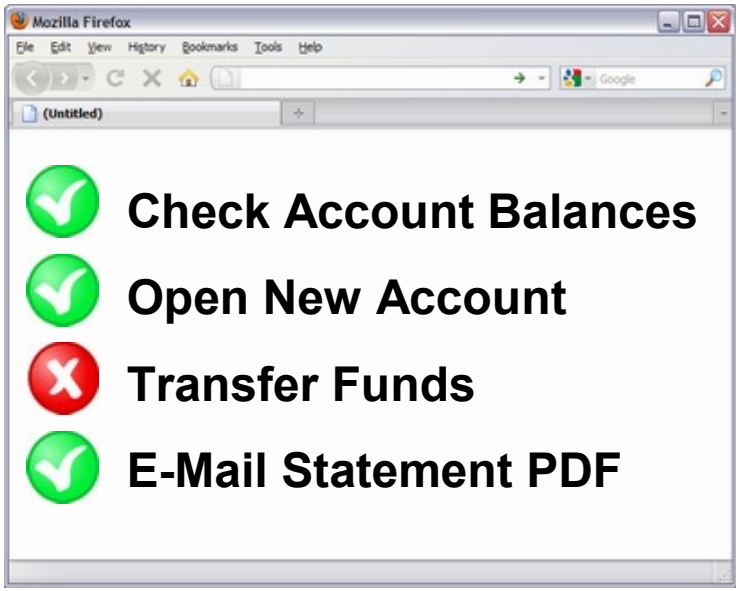
b) DB2 z/OS JDBC Type 4 driver 2.7 or higher has "enableSysplexWLB" function. This is what DB2 Connect has had before that.

Initial connection to DB2 data sharing group returns to the adapter the host:port of *all the members*. Adapter opens TCP connections to *all*. WLM feeds health metrics to the adapter. The adapter connection pool code spreads work across the established connections.

Connector summary ...

Final Point on Data Connectors

Again ... what constitutes an outage?

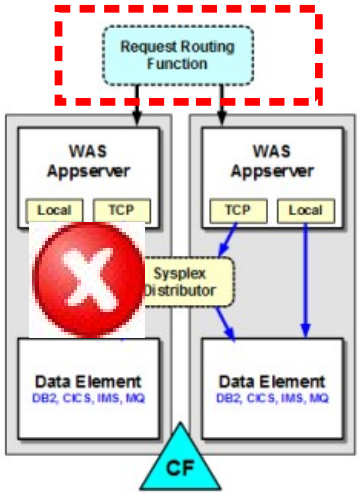


Depending on the nature of the local data system loss, you may well have only partial loss of function.

That may be unacceptable.

Or it may be okay.

Which means you should get agreement on this to avoid over-engineering



Remember ... the total loss of an LPAR -- planned or unplanned -- is an issue separate from the loss of a particular data subsystem.

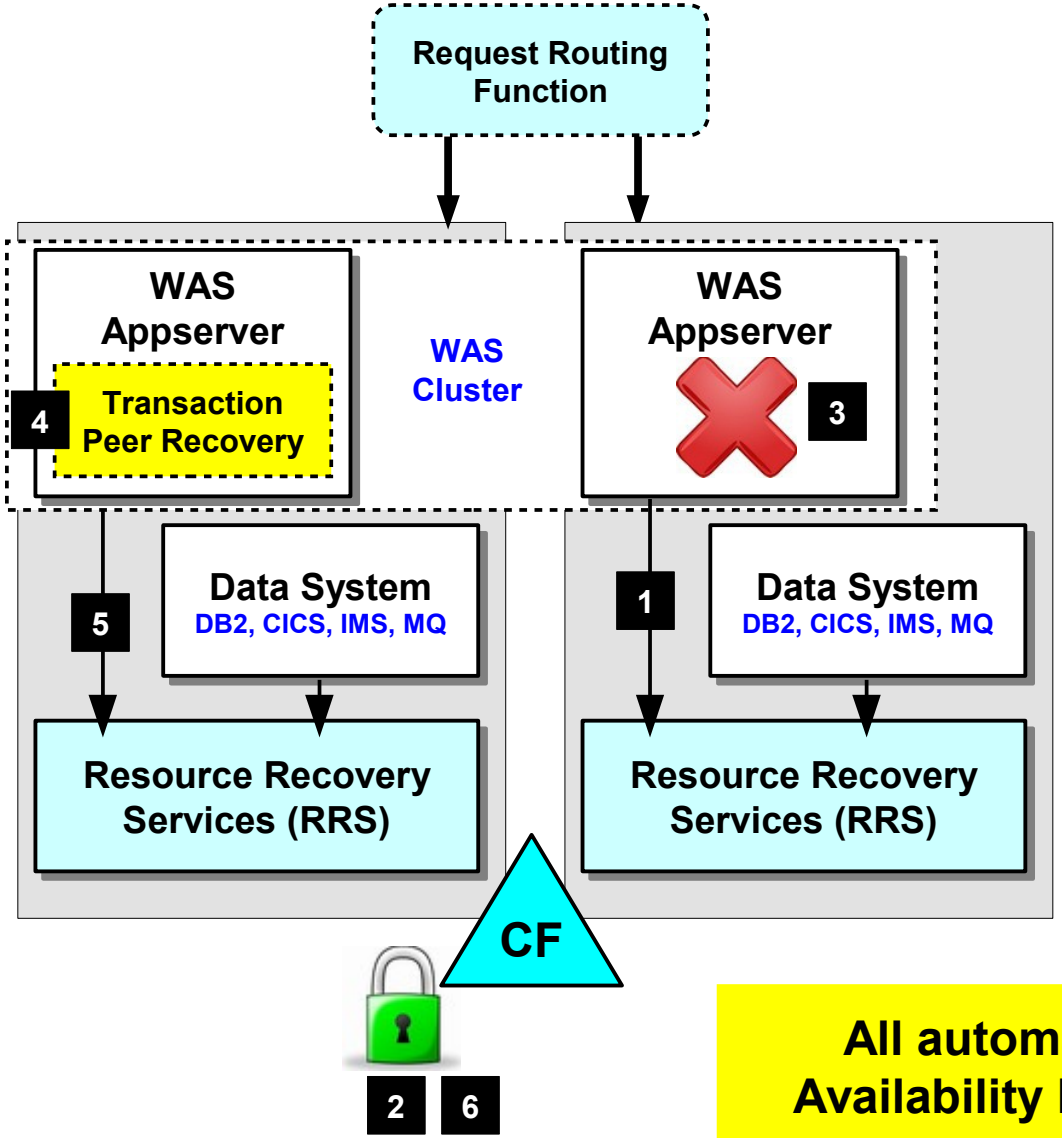
Losing the whole LPAR implies losing the WAS cluster member as well. That means the front-end routing should easily detect and route to surviving members.

In this case using a TCP-based connector doesn't buy you anything since the whole LPAR is gone.

Transactions ...

Transaction Rollback

This is where WAS and RRS come into play:



1. WAS on LPAR B starts TX

The application in WAS on LPAR A starts a global TX. WAS on LPAR A registers this in RRS.

2. Some Data Lock(s) Registered

One of the participating data systems is called upon and it locks some data as part of the TX. Locks managed in RRS.

3. WAS on LPAR B crashes

And leaves the TX in-doubt

4. TX Peer Recovery on LPAR A

In the failed server's cluster partner detects the loss. It acts on behalf of the failed server to roll back the TX through RRS.

5. TX Rolled Back

RRS signals to all participants in the TX to roll the TX back.

6. Data Locks Released

Participating data systems release their held locks.

All automatic provided you leave the High Availability Manager (HAM) function turned on

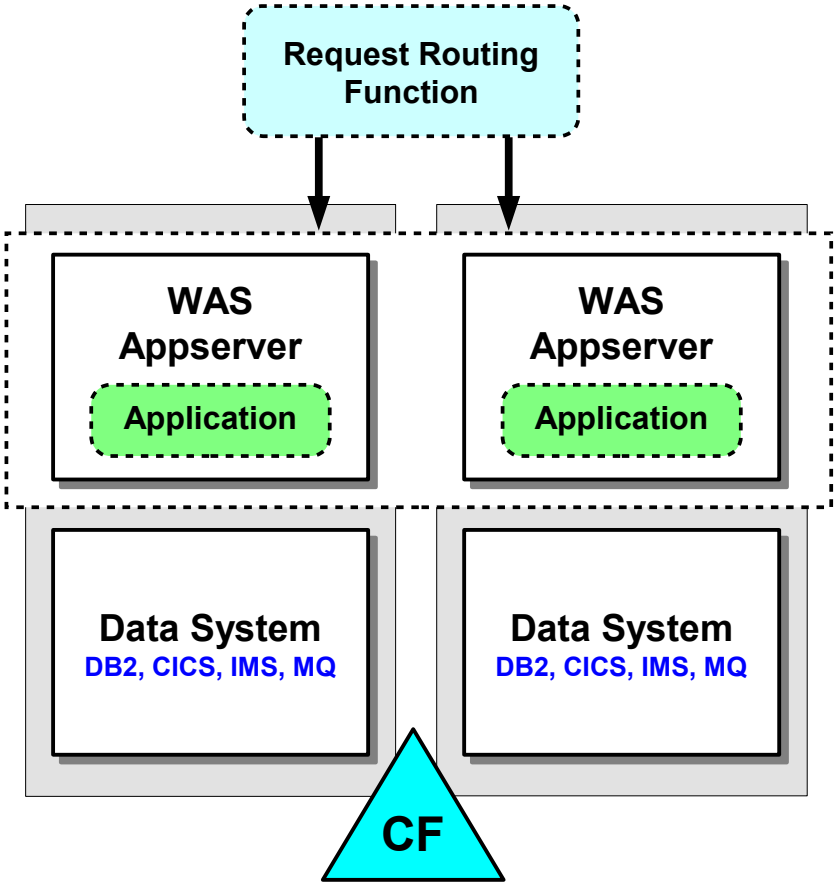
Planned outages ...

Planned Outages

System, Subsystem, WAS and Application Maintenance

Four Broad Activities Here

They are ...



1. Plan for Isolation

The design of the system has to allow the isolation of pieces of the overall.

This also includes having sufficient remaining capacity to handle the work

2. Quiesce Inbound Work and Flush

The flow of inbound work has to be routed away from the portion you are isolating, and all inflight work within that portion has to flush out.

3. Roll the Updates

Take the necessary steps to roll in the desired changes

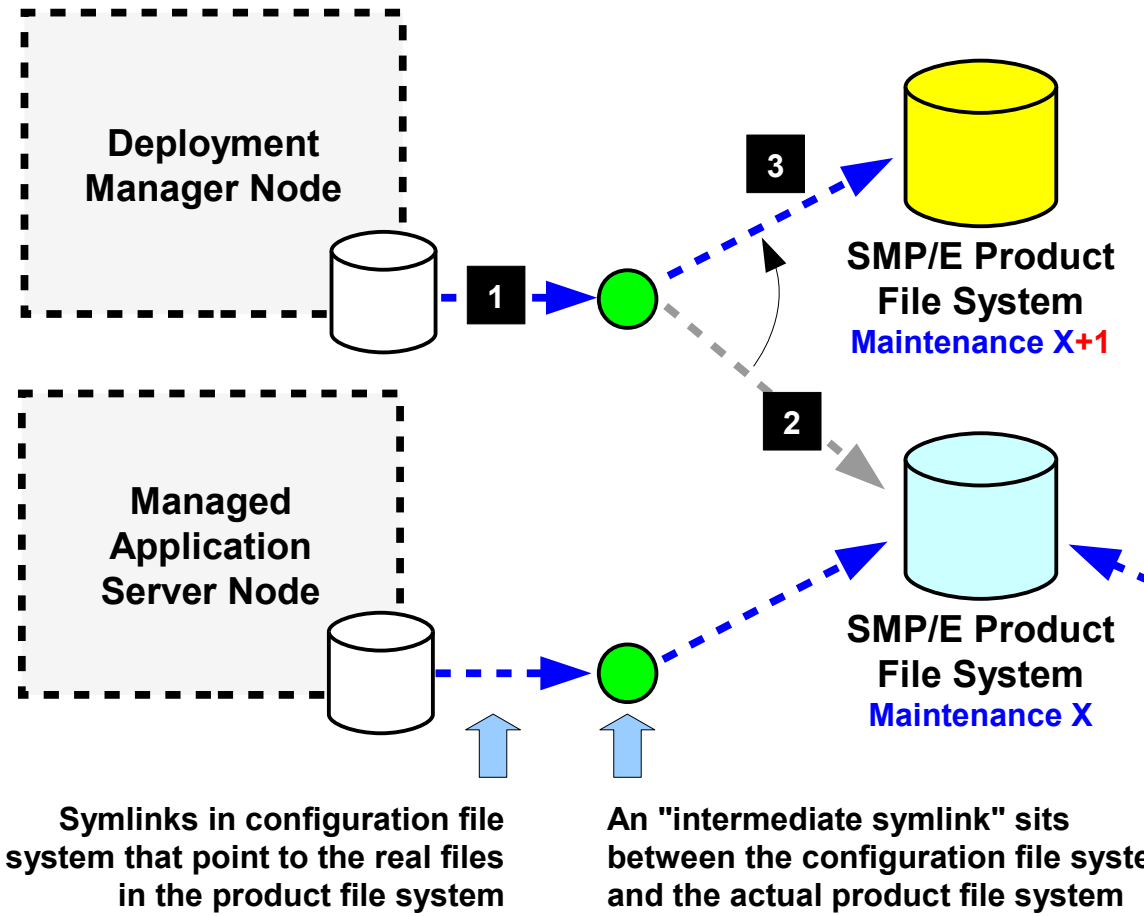
4. Resume Inbound Work

Ready the environment and resume the inbound work dispatching.

Node isolation ...

The Keys -- Nodes and Isolating Nodes for Rolling Updates

The WAS "node" is the unit of update for maintenance and migration. It's important to insure your nodes can be isolated from one another:



1. Config Symlinks

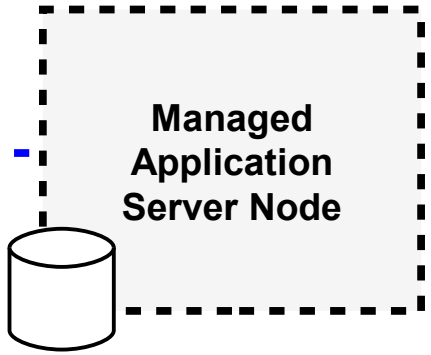
The node is configured to point to an intermediate symlink rather than the actual SMP/E product file system

2. Intermediate Symlink

Is configured to point to the actual SMP/E file system mount point

3. Update the Node

"Swing" the symlink to the new level of product code. Other nodes remain at the previous level. Allws you to "roll" through a Sysplex.



Concept explained in considerable detail:

`ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100396`

Utility to introduce intermediate symlinks into an existing configuration:

`ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1558`

Quiescing work ...

Quiescing Inbound Work to the Cluster Member

You have to starve off the incoming work and make sure all the in-flight work is flushed out before you shut down any servers and perform updates.

1. Reroute inbound work at front-end distribution device

Device may recognize listeners removed from service, but an overt action at the front-end is a better first step to insure that work is re-routed in an orderly fashion

Remember -- not all "work" comes through that device … internal WebSphere IIOP flows will continue, which is why the next step is important

2. Use `PAUSELISTENERS` or `STOP` at application server level

This will remove the cluster member from consideration for internal IIOP routing as well as Message Drive Bean (MDB) traffic

STOP will do same thing as PAUSELISTERS, but will proceed immediately to server shutdown. In-flight work should flush out before server is stopped

3. Use `MODIFY <server>,DISPLAY,WORK` to insure work flushed

If you used PAUSELISTENERS, you can use `/F <server>,DISPLAY,WORK` to see what work is still in process

If Version 7.0, `/F <server>,DISPLAY,THREADS` is a finer view of in-process work

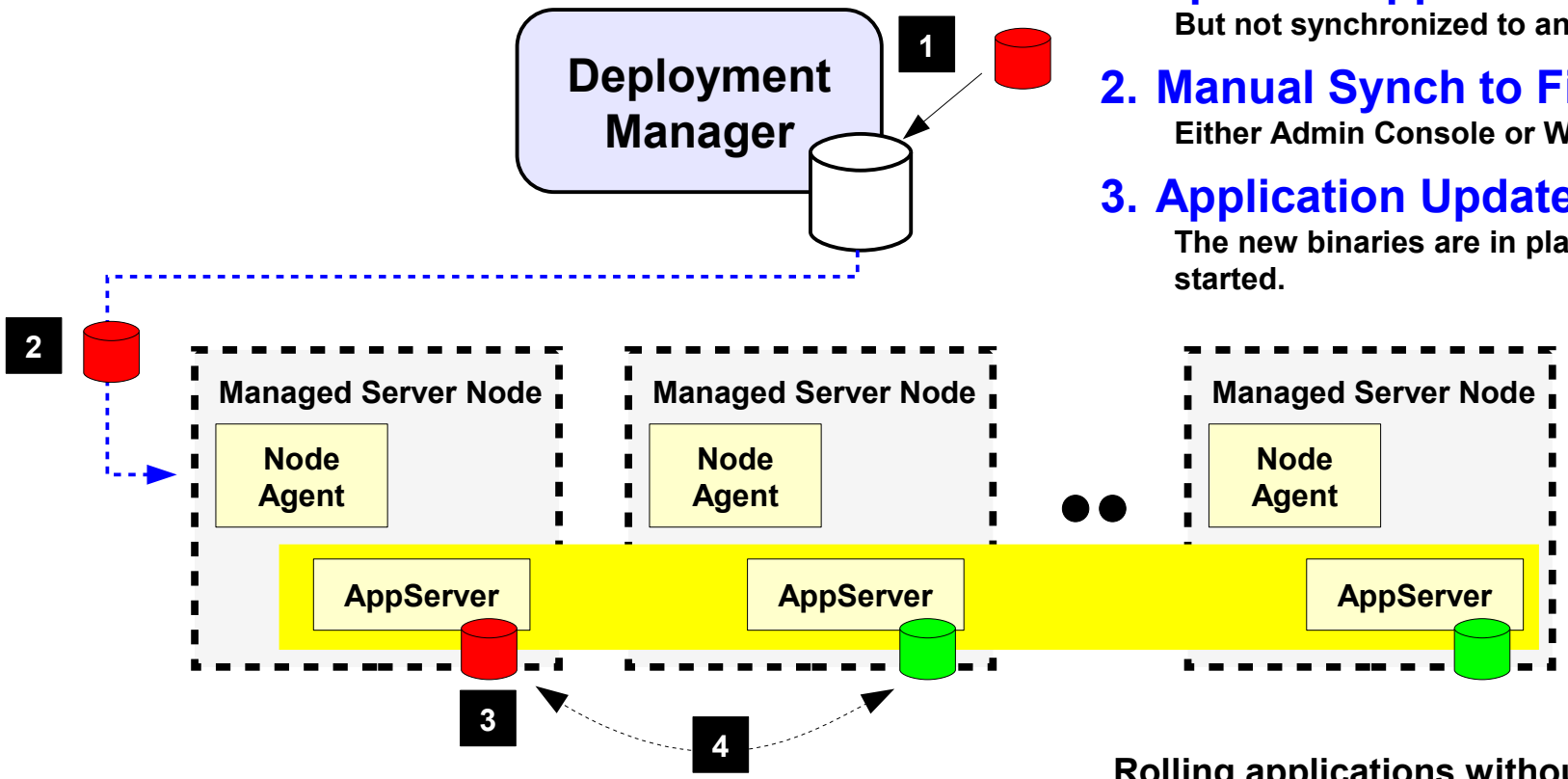
4. Initiate planned outage

With all inbound work re-routed and all current work flushed, you may proceed with the planned outage

Rolling application updates ...

Rolling Application Updates

Operationally not that difficult ... but a question arises when you turn on traffic to the updated application when the prior level is still active in other cluster members:



1. Updated Application Deployed

But not synchronized to any nodes yet

2. Manual Synch to First Node

Either Admin Console or WSADMIN Script

3. Application Updated / Started

The new binaries are in place and the application started.

4. The Question

Do you open up work distribution to this node? That implies two levels of application available at the same time. Is that an issue?

Rolling applications without any concurrent exposure of old and new applications implies a brief period of buffering requests while final application is switched. IBM Virtual Enterprise On Demand Router features that capability.

Techdoc explaining how to perform a staged rollout:

`ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101641`

Summary ...

Summary

- Get consensus on the level of availability truly needed
- Engineer in appropriate redundancy in both hardware as well as software
- Engineer in sufficient isolation so maintenance and updates can be rolled
- Execute with appropriate care and precision ... change control and process management is key