

IBM Business Process Manager Scalability and Availability

Dave Hay

Smarter Process Infrastructure Consultant

IBM Software Services for WebSphere

(ISSW)

david_hay@uk.ibm.com

+44 7802 91423



IBM
Software Services
for WebSphere



Introduction

- *In this session, you will discover how an IBM BPM infrastructure can be designed and built to meet some of the more common non-functional requirements for performance, scalability and resilience.*
- *This session assumes some prior knowledge of the IBM BPM solution, as well as some of the key infrastructure building blocks, including HTTP load balancing, WebSphere Application Server and relational database systems*

Thanks for inspiration and support from ...



- Nguyen-Anh Le
- Karri Carlson-Neumann
- Dawn Ahukanna
- Chris Richardson
- Mike Collins
- Dennis Parrott
- Paul Pacholski
- Andy Garratt

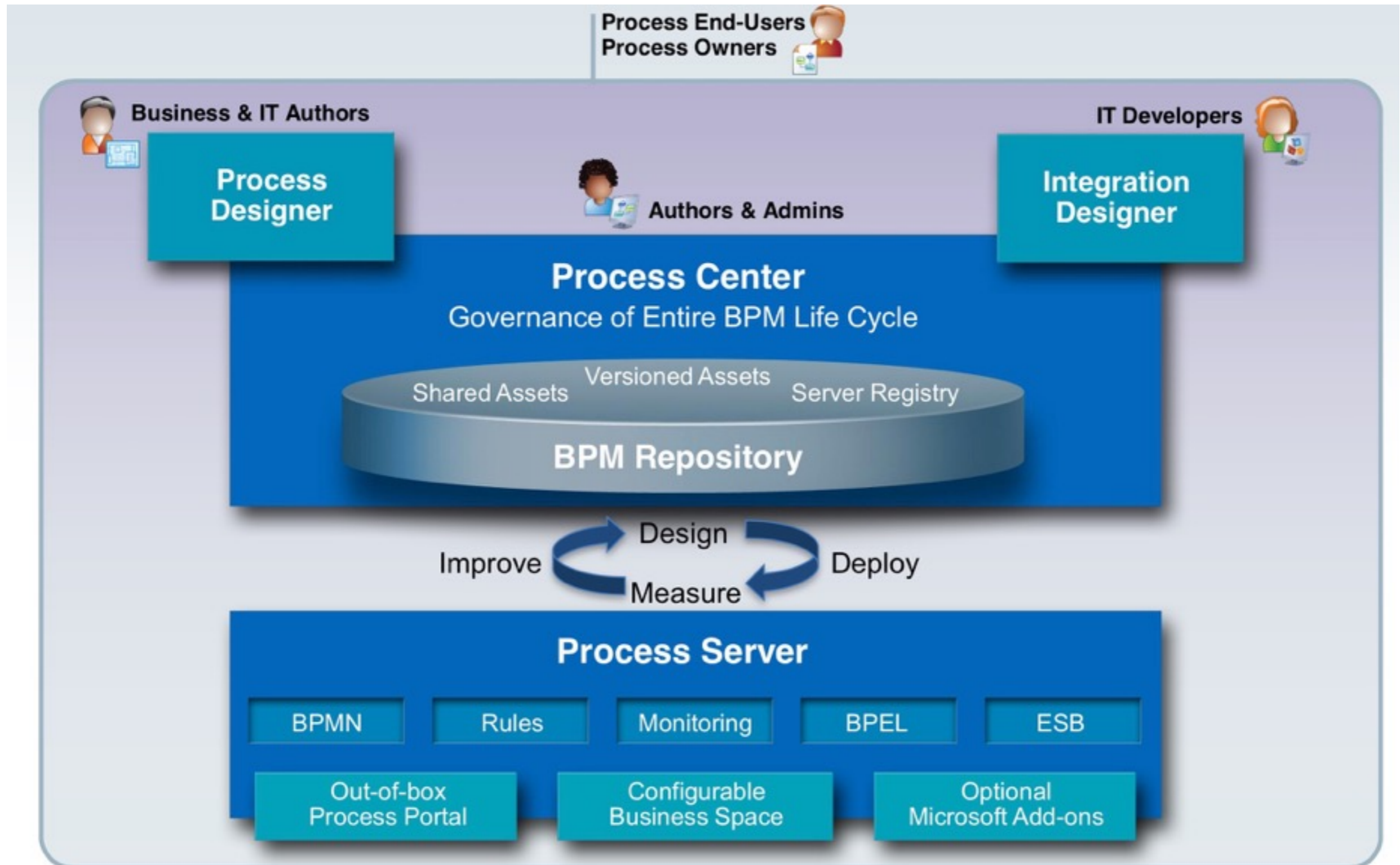


Introduction

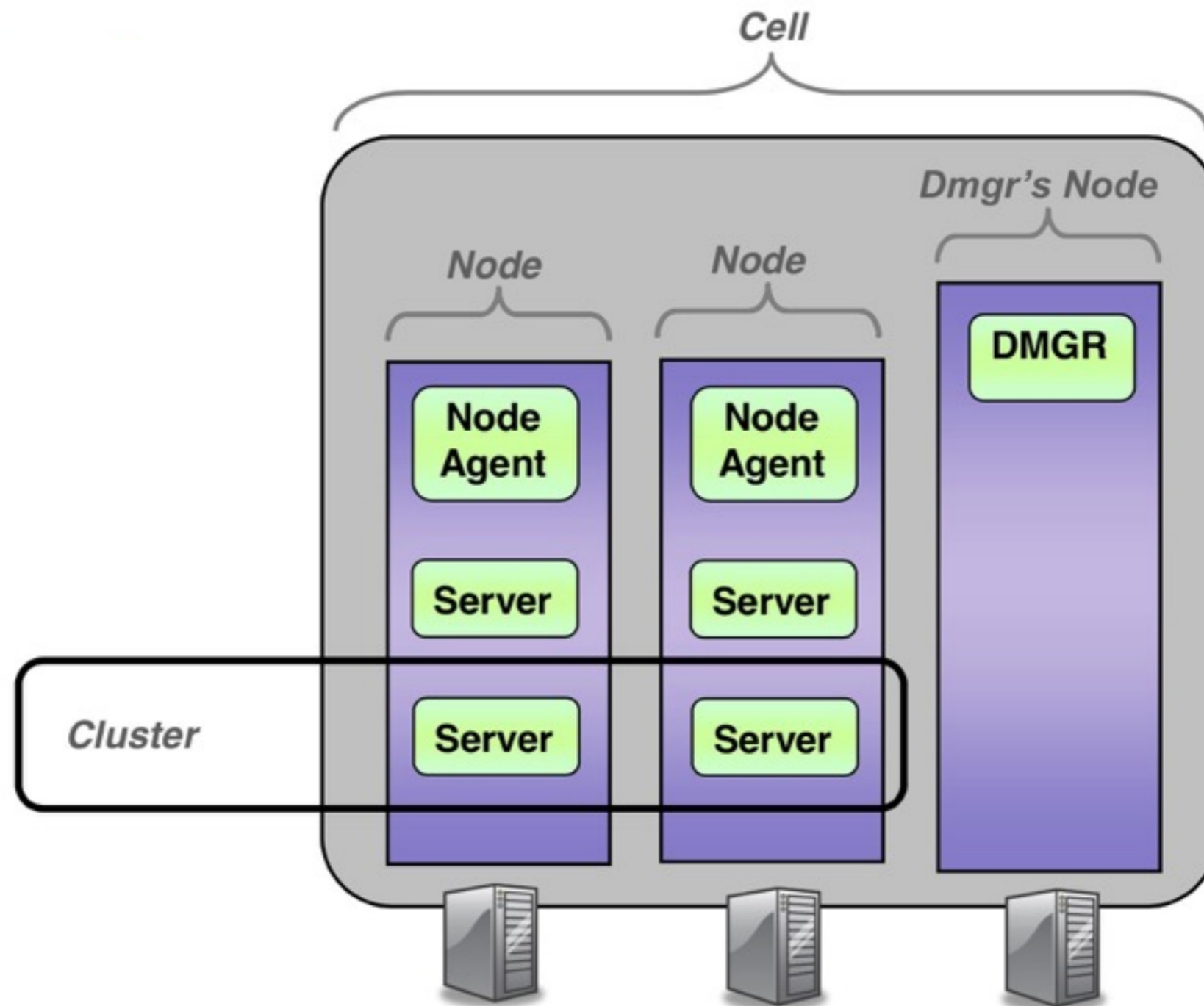
- BPM leverages WAS capabilities
 - Scalability
 - Resilience
 - Security
 - Standards
 - Performance
- This deck outlines how: -
 - Deployment Environments
 - Vertical and Horizontal Scaling
 - HA vs. DR
 - Cross-data centre resilience
 - Managing transaction logs



Context



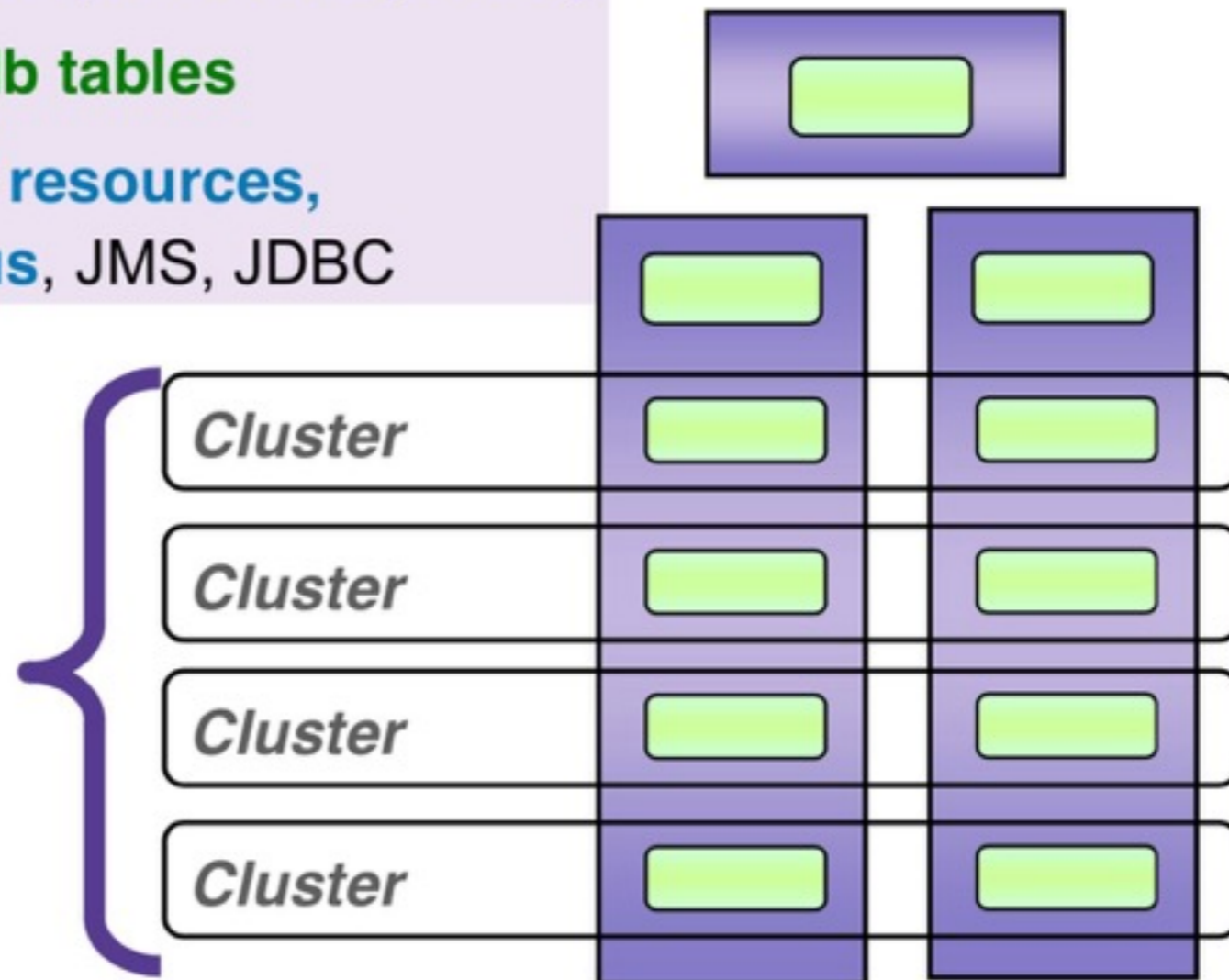
The Elements of a WAS Topology





Introducing the Deployment

- Well known patterns for deployment
- Things included in a pattern:
 - BPM functionality
 - **Some number of clusters which work together to provide capability**
 - **Collection of db tables**
 - **Infrastructural resources, including SIBus, JMS, JDBC**





What's a Deployment Environment ?

- We call it DE for short :-)
- Introduced in BPM 7.0
- Comprises a single BPM configuration
- Allows for multiple BPM environments in one cell
- Separates WAS and BPM roles
 - e.g *wasadmin* for cell, *deAdmin* for DE

Deployment Environment Revealed



PCCell1

Aliases Bus

CellIDB

bpm855.uk.ibm.com : Port number

Dmgr

bpm855.uk.ibm.com : Port number

De1

ProcessServ...

Performanc...

SharedDb

Node1

AppCluster [Application]

AppClusterMember1 2

MECluster [Messaging]

MEClusterMember1 2

SupCluster [Support]

SupClusterMember1 2

More on this later ...

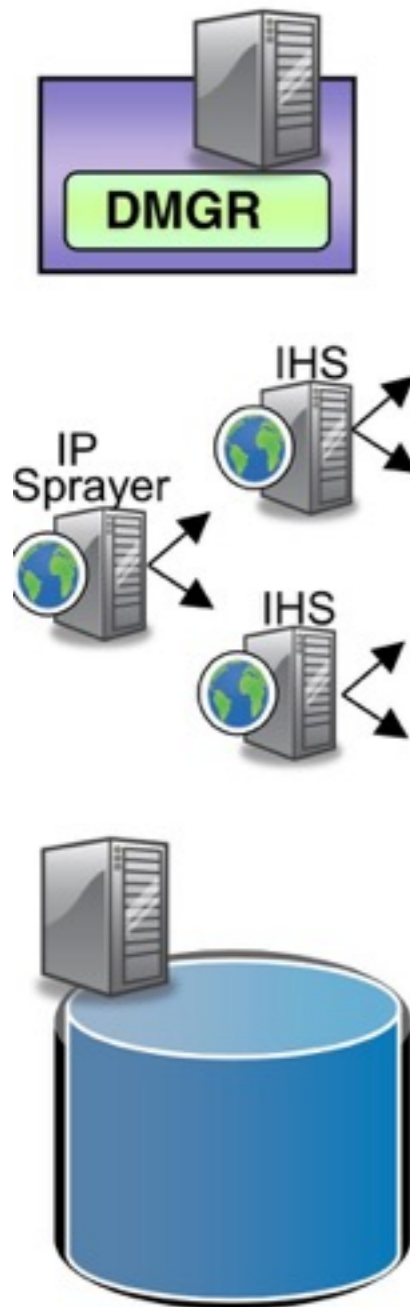


Scaling - Up and Out

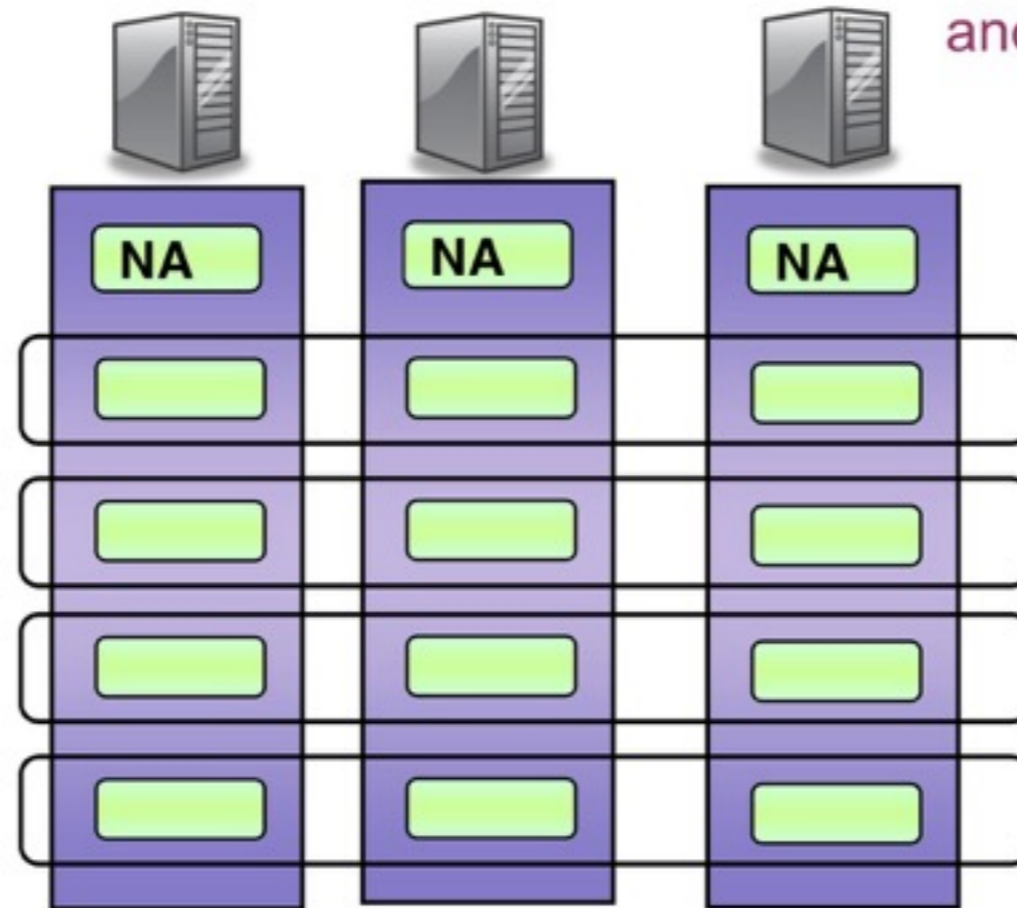
- Deployment Environment spans Cell
 - Allows for horizontal scaling
 - Cell spans multiple virtual/physical nodes
- Multiple Cells or Multiple DEs
 - Allows for vertical scaling
 - "Sweat the assets"



Vertical Scaling - 1/3



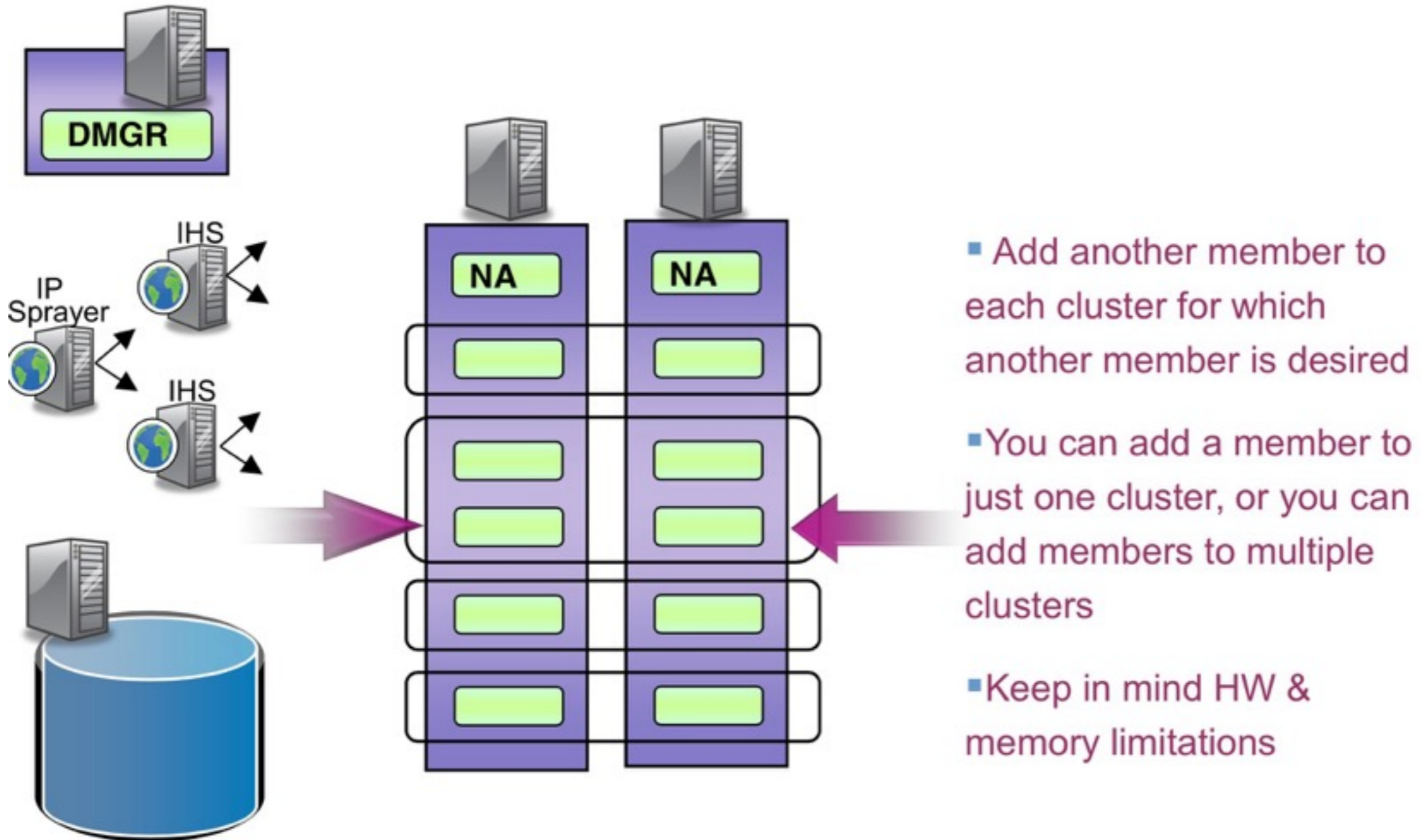
- Add another node
- Add another member to each cluster for which another member is desired



- You can add another member to just one cluster, or you can add another member to multiple clusters

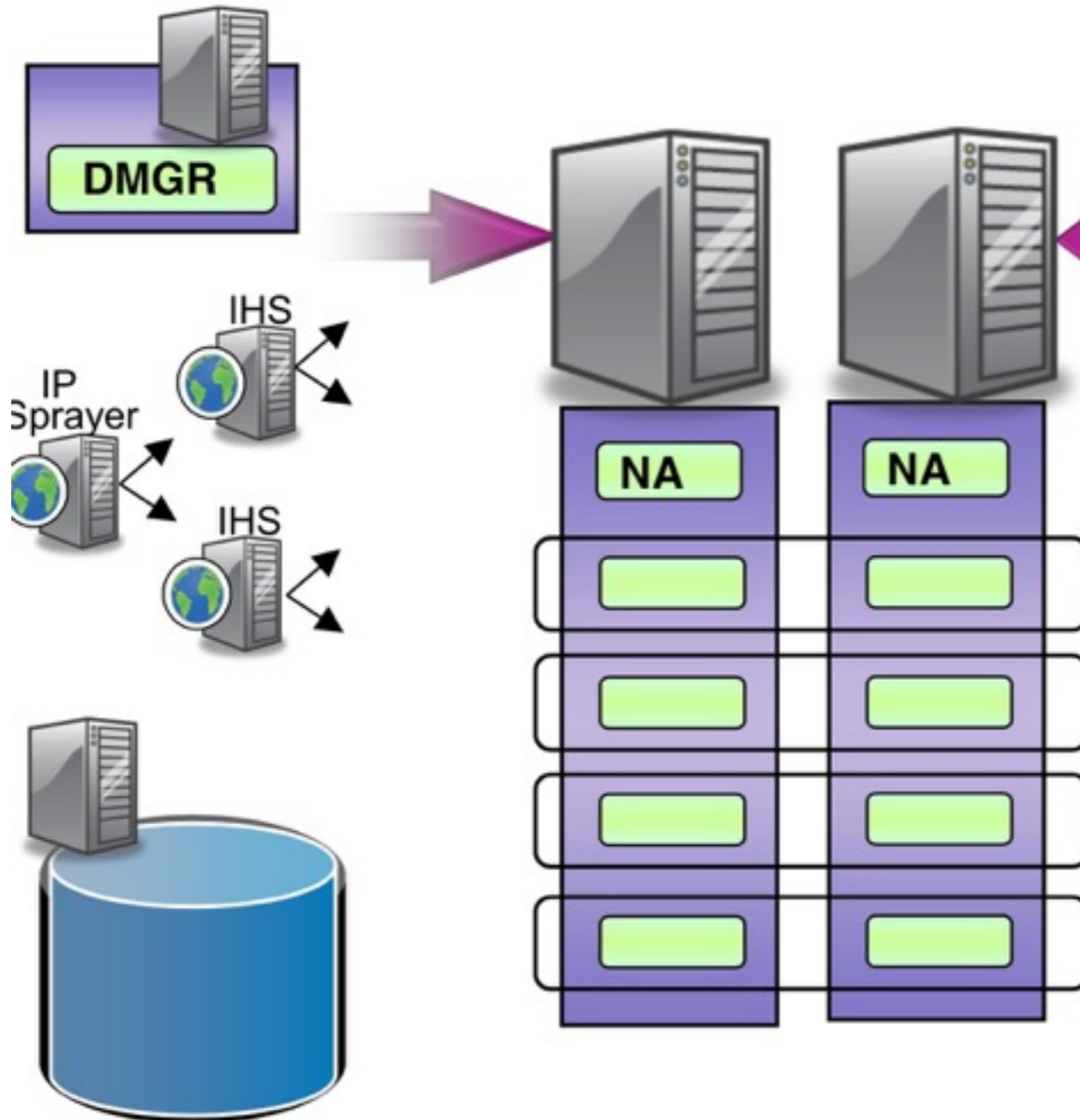


Vertical Scaling - 2/3





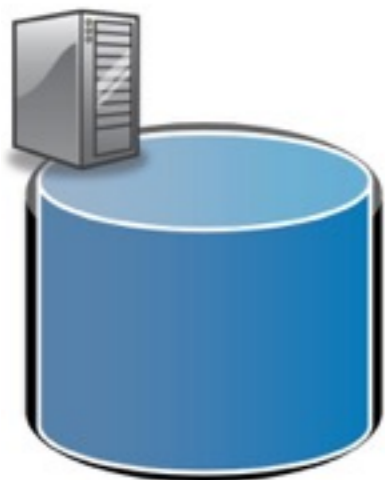
Vertical Scaling - 3/3



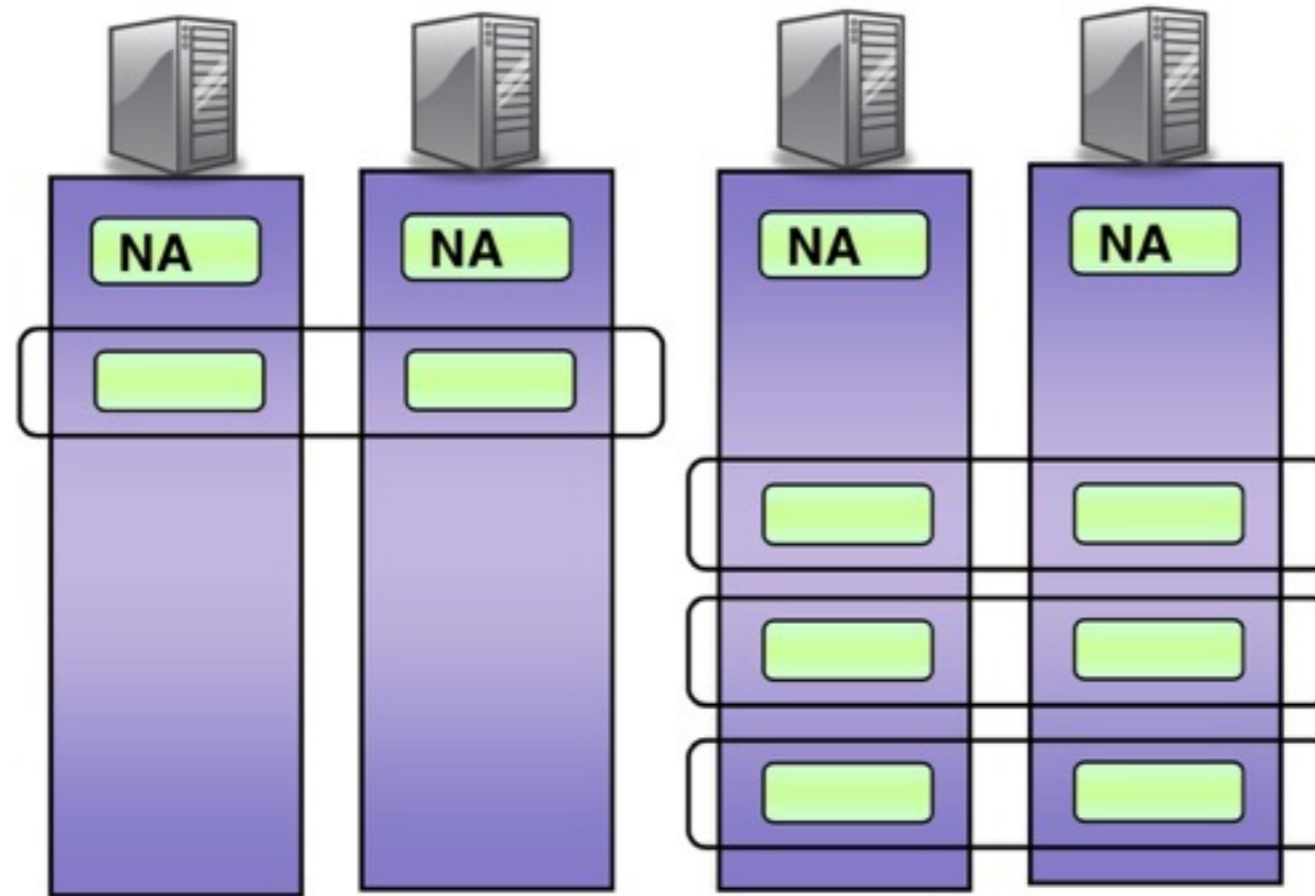
- Some resource constraints may be alleviated by increasing the available resources



Horizontal Scaling

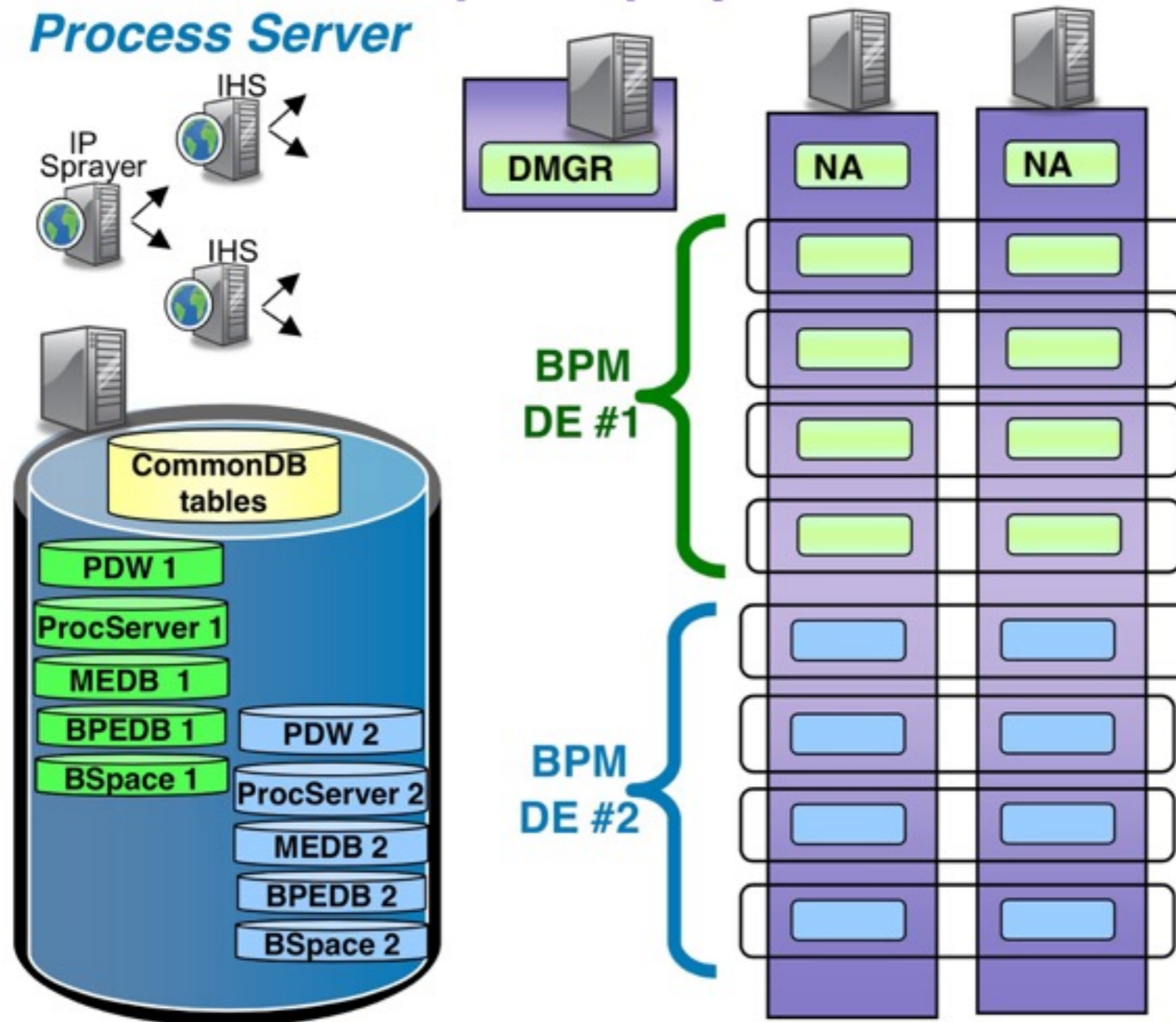


- Every node does NOT have to host a member of every cluster





The Multiplier Effect



Multiple Cells vs Multiple DEs



- Considerations
 - How many sets of binaries ?
 - How many WAS profiles ?
 - How many Deployment Managers & Node Agents ?
 - Maintenance & Downtime
 - Configuration Complexity



What's the answer ?

- Is it a case of "Six and Two Threes" ?
- What does IBM support / suggest ?



(C) Universal Pictures, 1993

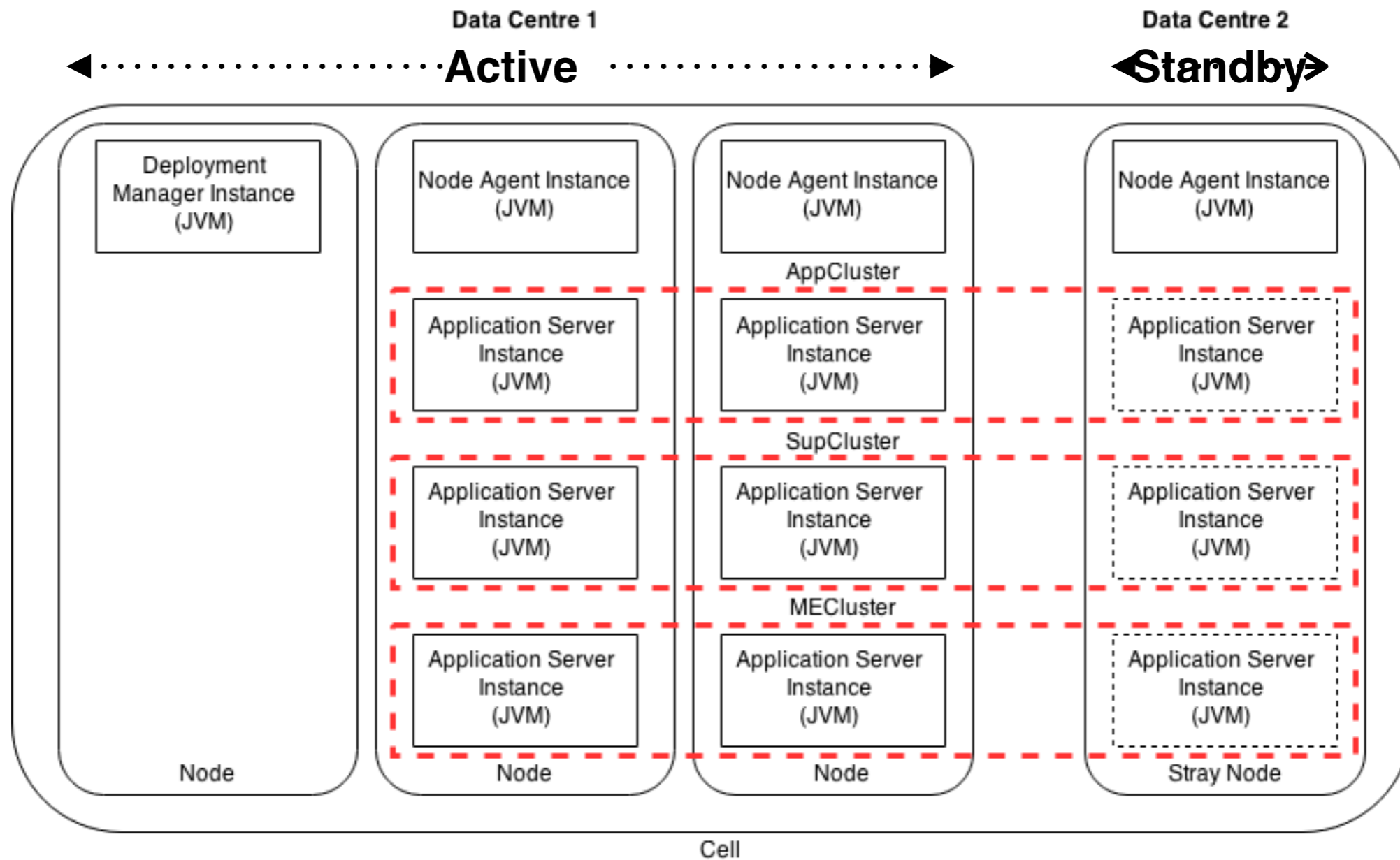


Multiple Data Centres

- Whilst active cell could span multiple data centres ...
 - We don't want to do that
 - Remember what Tom Alcott says
 - There are risks: -
 - Split Brain
 - Replicated Corruption
 - Network Load across the WAN
- Plan for cell \leq data centre boundaries
 - Still provides resilience across multiple nodes (think servers, racks, frames)
- For cross-data centre resilience, look at



The Stray Node Pattern



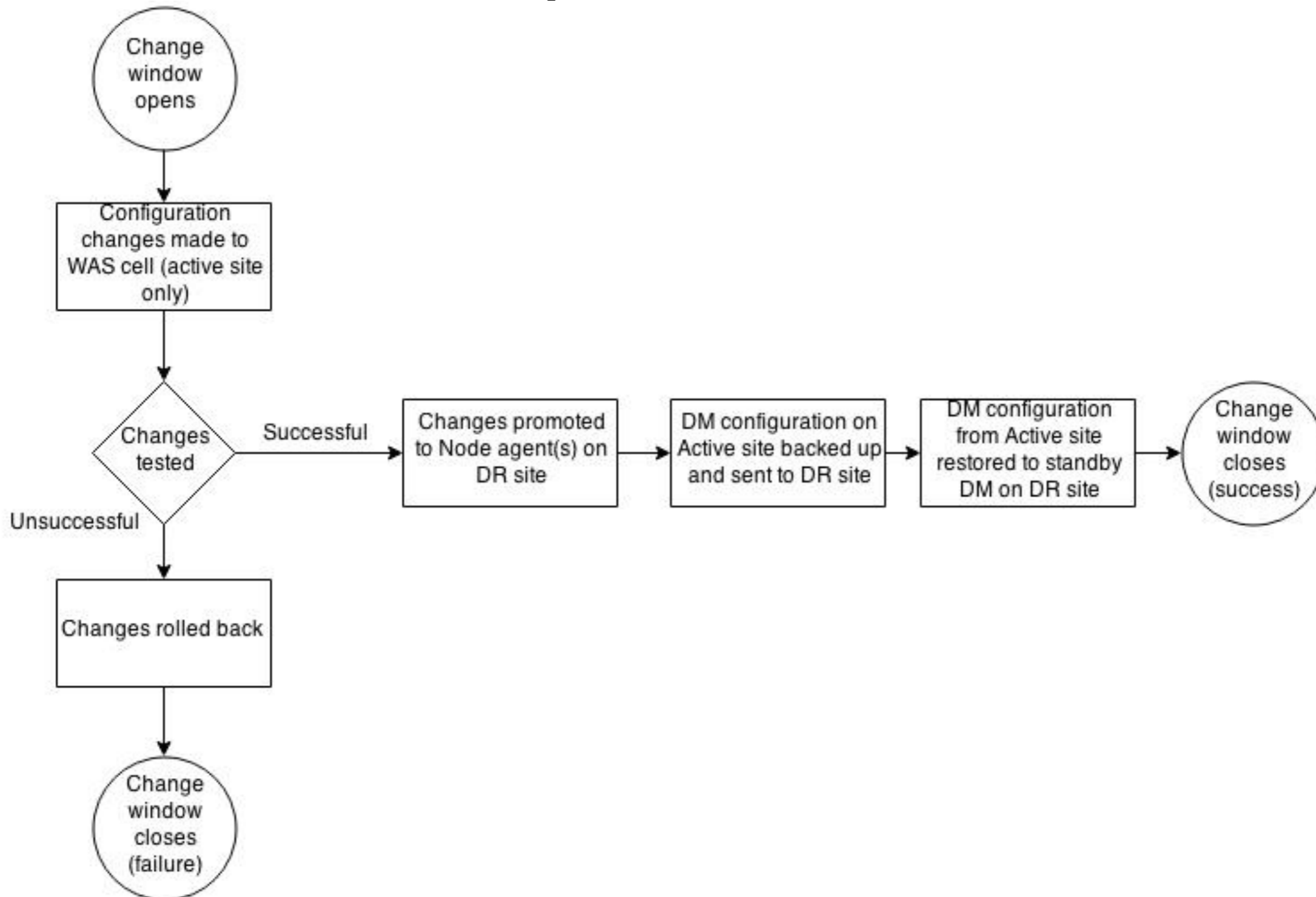


The Stray Node Pattern

- Offers warm standby across DCs
 - Stray Node hosts active Node Agent, but passive Cluster Members
- Cell **does** span DC boundaries but ...
 - active workload only runs in DC1 or DC2, but not both
 - uses core groups to separate active and stray nodes
- Need to plan for Deployment Manager resilience (backup DM)
- Need a decent change management workflow ...



Example Workflow





HA vs DR

- **High Availability**

The ability of an application to deal with the unplanned loss of a single hardware or software component (for instance, due to a bug or failure).

- **Disaster Recovery**

Ensuring that the system can be reconstituted and/or activated at another location and can process work after an unexpected catastrophic failure at one location.



HA and DR are different

- HA is: -
 - Achievable
 - Testable / Provable
- DR is: -
 - Typically +1 Data Center
 - Expensive
 - Harder to test & prove
- HA != DR

Summary

- *HA (component-level resilience) should be first priority*
- *Invoking DR relates to major business impact*
 - *Think "We've lost a data centre" or "Major data corruption"*
- *In the event of a disaster, is BPM the main consideration ?*
- *Let the NFRs guide ...*
- *Consider cascade failure - losing multiple components e.g. double jeopardy, triple jeopardy*
- *Think end-to-end service*

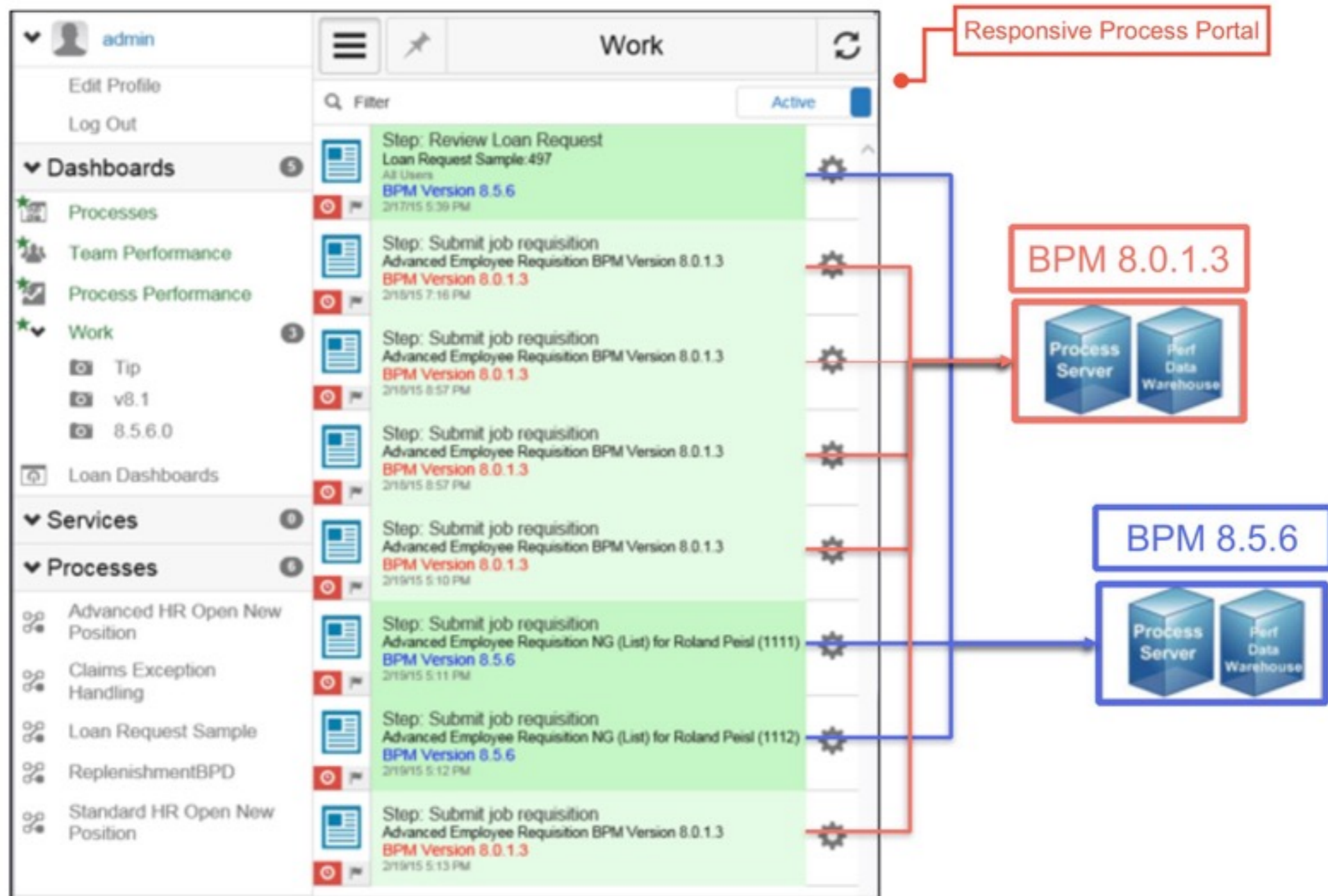


Active/Active BPM

- Consider two Deployment Environments, both active
 - Regardless of whether via one cell or two cells
- Each DE has a set of web UIs - Process Portal etc.
- Two DEs = two Process Portals
- What about aggregation ?
 - BPM 8.5.6 has the answer ...

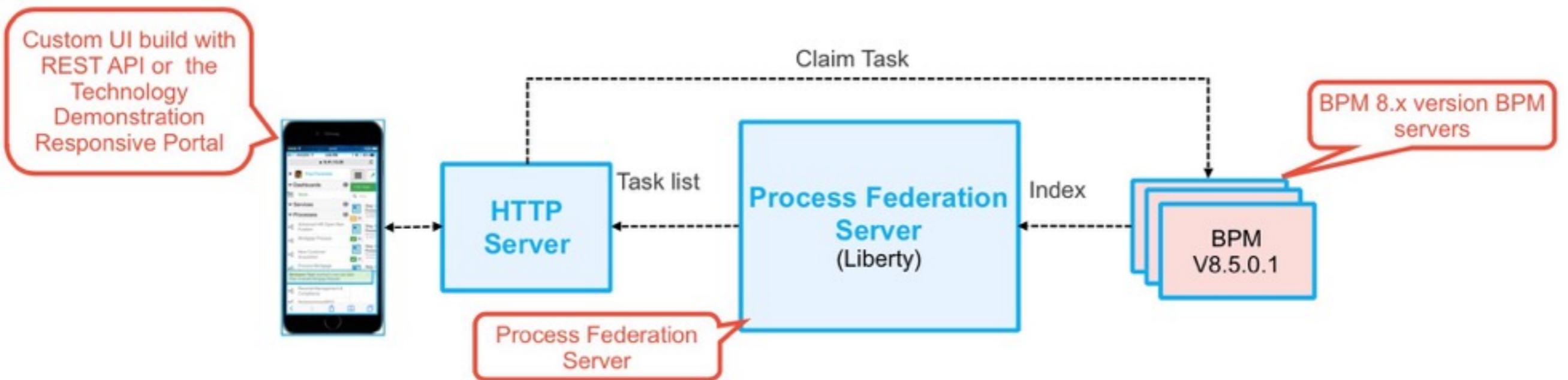


Process Federation Server





Process Federation Server





Sharing State - 1/2

- Consider Long-Running Processes
 - Typically implemented via BPMN
 - Need to share state between AppCluster members
- Consider Straight-Through Processing
 - Typically implemented via SCA (BPEL microflows etc.)
 - Need to share state between AppCluster members
- Let requirements dictate solution



Sharing State - 2/2

- BPM relies upon Transaction, Partner and Compensation Logs (aka TranLogs)
 - Out-of-the-box WAS capability
- For multiple node, these need to be shared
 - Supports HAManager recovery
 - Via disk (NFS, GPFS etc.) or in DB
 - Allows for consistent recovery group
 - Required for rollback and completion
 - Aim is to avoid "in doubt transactions"



In Conclusion

- BPM
 - can be scaled to meet the NFRs of the business
 - builds upon the capabilities of WAS
 - scales vertically to maximise infrastructure potential
 - scales horizontally to help eliminate SPOF
- Remember: -
 - Requirements in, design and implementation out



Questions ?





Want to provide feedback?

- Then please visit here: -

ibm.biz/wug15feedback



References

- [Business Process Management Deployment Guide Using IBM Business Process Manager V8.5](#)
- [IBM Business Process Manager Version 8.0 Production Topologies](#)
- [Faster disaster recovery in IBM Business Process Manager](#)
- [Application, Remote Messaging, and Remote Support topology pattern](#)
- [Storing transaction and compensation logs in a relational database for high availability and disaster recovery in IBM Business Process Manager](#)
- [Comment lines: Tom Alcott: Everything you always wanted to know about WebSphere Application Server but were afraid to ask -- Part 3](#)



Useful Links

- IBM Business Process Manager Developer Center

<https://developer.ibm.com/bpm>

- IBM Operational Decision Manager Developer Center

<https://developer.ibm.com/odm>



Backup - Cells vs. DEs

| | Single Cell, Single DE | Single Cell, Multiple DE | Multiple Cells, 1 DE per cell |
|---|----------------------------|---|---|
| Application (& dependencies) isolation | None. | Pretty good, per group of apps per DE | Absolute best. |
| Share JVM with other apps? | Yes, all other apps | Yes, but only with the apps also deployed to same target | Yes, but only with the apps also deployed to same target |
| Share tables? | Yes, all tables are shared | Each DE has own CMNDB, PDWDB, BPMDB tables. CellDB tables shared by all in cell. | Each DE has own CMNDB, PDWDB, BPMDB tables. Each cell has own CellDB tables. |
| Share SIBus? | One ME per each SIBus. | (v8.5) No, each DE has own SIBus. One ME per each SIBus (prior to v8.5) Multiple members per each SIBus → Multiple MEs per each SIBus. | One ME per each SIBus. |
| Share product binaries? | Yes | Yes | No (HIGHLY prefer separate install roots for separate cells, even if sharing HW) |
| Apply product ifix impacts all JVMs? | Yes | Yes (ifixes, fixpacks apply to install root, not clusters. Therefore all clusters in the cell are impacted) | No (just the cell to which the ifix has been applied) |

↑
 Note that total # of JVMs is not significantly different
 ↑