IBM System z

z/OS and Linux for System z: *Selecting the Best SOA Platform*

Bill Seubert
System z I/T Architect

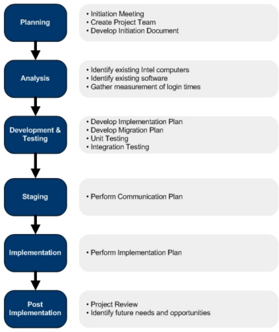
Agenda

- Selecting computing platforms
- A SOA infrastructure reference architecture
- z/OS and Linux on System z
  - Architectural contrast
  - Strengths & weaknesses
- Which one?
  - Application suitability
  - Qualities of service
  - Cost considerations
  - “Intangibles”
- Examples

Agenda

- Selecting computing platforms
- A SOA infrastructure reference architecture
- z/OS and Linux on System z
  - Architectural contrast
  - Strengths & weaknesses
- Which one?
  - Application suitability
  - Qualities of service
  - Cost considerations
  - “Intangibles”
- Examples

What's your selection methodology?


Planning
- Initiation Meeting
- Create Project Team
- Develop Initiation Document

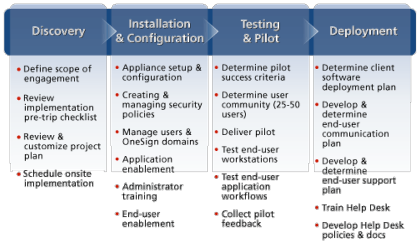
Analysis
- Identify existing Intel computers
- Identify existing software
- Gather measurement of login times

Development & Testing
- Develop Implementation Plan
- Develop Migration Plan
- Unit Testing
- Integration Testing

Staging
- Perform Communication Plan

Implementation
- Perform Implementation Plan

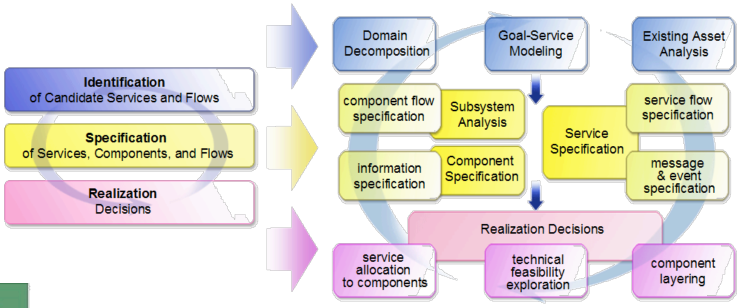
Post Implementation
- Project Review
- Identify future needs and opportunities

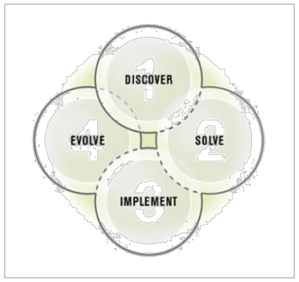

Discovery
- Define scope of engagement
- Review implementation pre-trip checklist
- Review & customize project plan
- Schedule onsite implementation

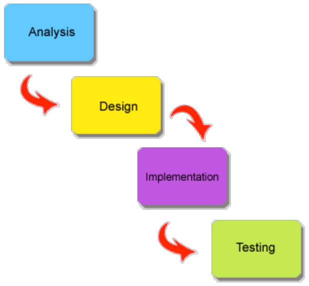
Installation & Configuration
- Appliance setup & configuration
- Creating & managing security policies
- Manage users & OneSign domains
- Application enablement
- Administrator training
- End-user enablement

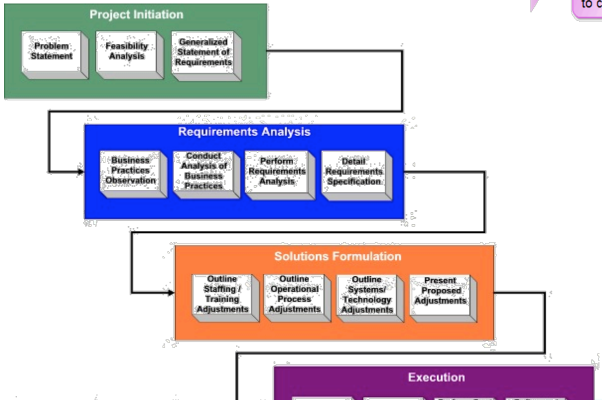
Testing & Pilot
- Determine pilot success criteria
- Determine user community (25-50 users)
- Deliver pilot
- Test end-user workstations
- Test end-user application workflows
- Collect pilot feedback

Deployment
- Determine client software deployment plan
- Develop & determine end-user communication plan
- Develop & determine end-user support plan
- Train Help Desk
- Develop Help Desk policies & docs

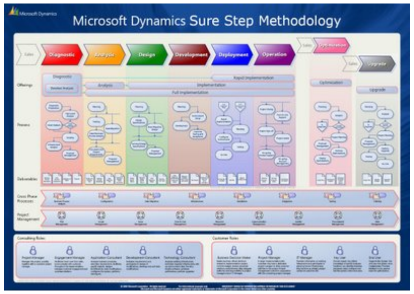

Identification of Candidate Services and Flows
Specification of Services, Components, and Flows
Realization Decisions
Domain Decomposition
Goal-Service Modeling
Existing Asset Analysis
component flow specification
Subsystem Analysis
Service Specification
service flow specification
information specification
Component Specification
message & event specification
Realization Decisions
service allocation to components
technical feasibility exploration
component layering

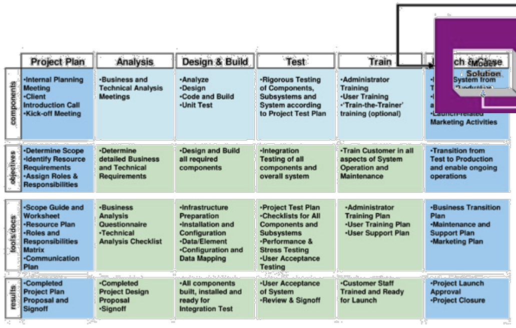




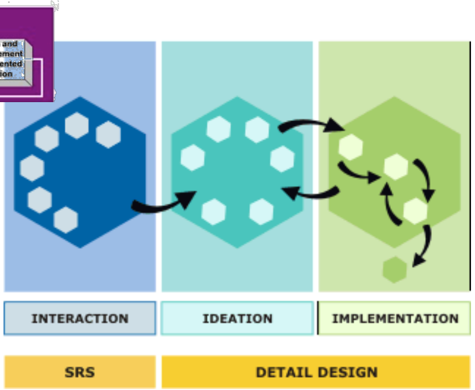

Project Initiation
Problem Statement
Feasibility Analysis
Generalized Statement of Requirements
Requirements Analysis
Business Practices Observation
Conduct Analysis of Business Practices
Perform Requirements Analysis
Detail Requirements Specification
Solutions Formulation
Outline Staffing / Training Adjustments
Outline Operational Process Adjustments
Outline Systems/ Technology Adjustments
Present Proposed Adjustments
Execution
Model Solution
Implement Solution
Perform Post Implement-ation Evaluation
Refine and Improvement Implemented Solution

Is it something like one of these?





	Project Plan	Analysis	Design & Build	Test	Train	ch & Close
components	•Internal Planning Meeting •Client Introduction Call •Kick-off Meeting	•Business and Technical Analysis Meetings	•Analyze •Design •Code and Build •Unit Test	•Rigorous Testing of Components, Subsystems and System according to Project Test Plan	•Administrator Training •User Training •'Train-the-Trainer' training (optional)	•[illegible] System from T[illegible] Production •[illegible] a[illegible] •Launch-related Marketing Activities
objectives	•Determine Scope •Identify Resource Requirements •Assign Roles & Responsibilities	•Determine detailed Business and Technical Requirements	•Design and Build all required components	•Integration Testing of all components and overall system	•Train Customer in all aspects of System Operation and Maintenance	•Transition from Test to Production and enable ongoing operations
tools/docs	•Scope Guide and Worksheet •Resource Plan •Roles and Responsibilities Matrix •Communication Plan	•Business Analysis Questionnaire •Technical Analysis Checklist	•Infrastructure Preparation •Installation and Configuration •Data/Element •Configuration and Data Mapping	•Project Test Plan •Checklists for All Components and Subsystems •Performance & Stress Testing •User Acceptance Testing	•Administrator Training Plan •User Training Plan •User Support Plan	•Business Transition Plan •Maintenance and Support Plan •Marketing Plan
results	•Completed Project Plan Proposal and Signoff	•Completed Project Design Proposal •Signoff	•All components built, installed and ready for Integration Test	•User Acceptance of System •Review & Signoff	•Customer Staff Trained and Ready for Launch	•Project Launch Approval •Project Closure



Or something more like this… ??

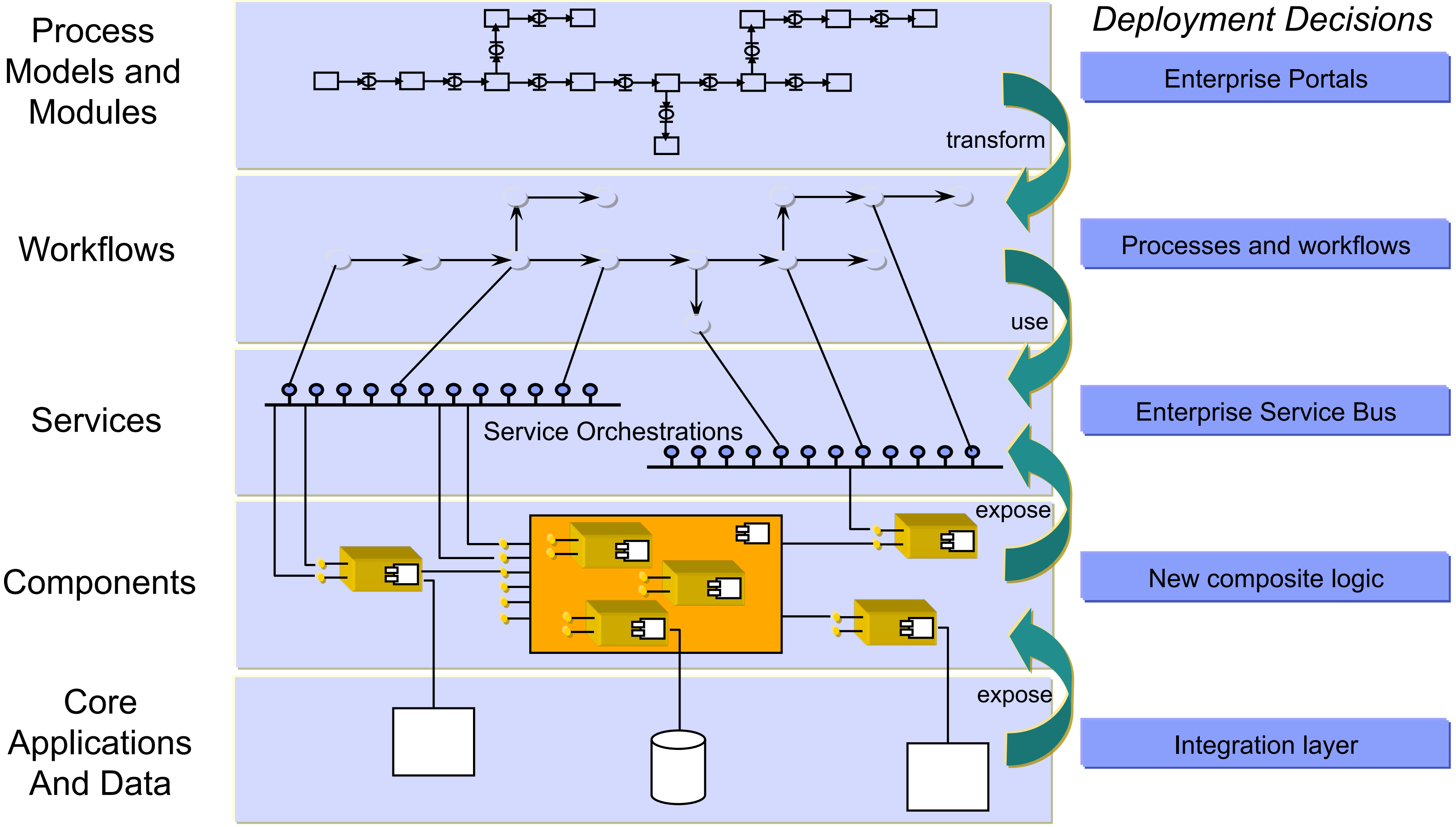
Methodology – how much do I need?

- A selection methodology is important, but how elaborate?

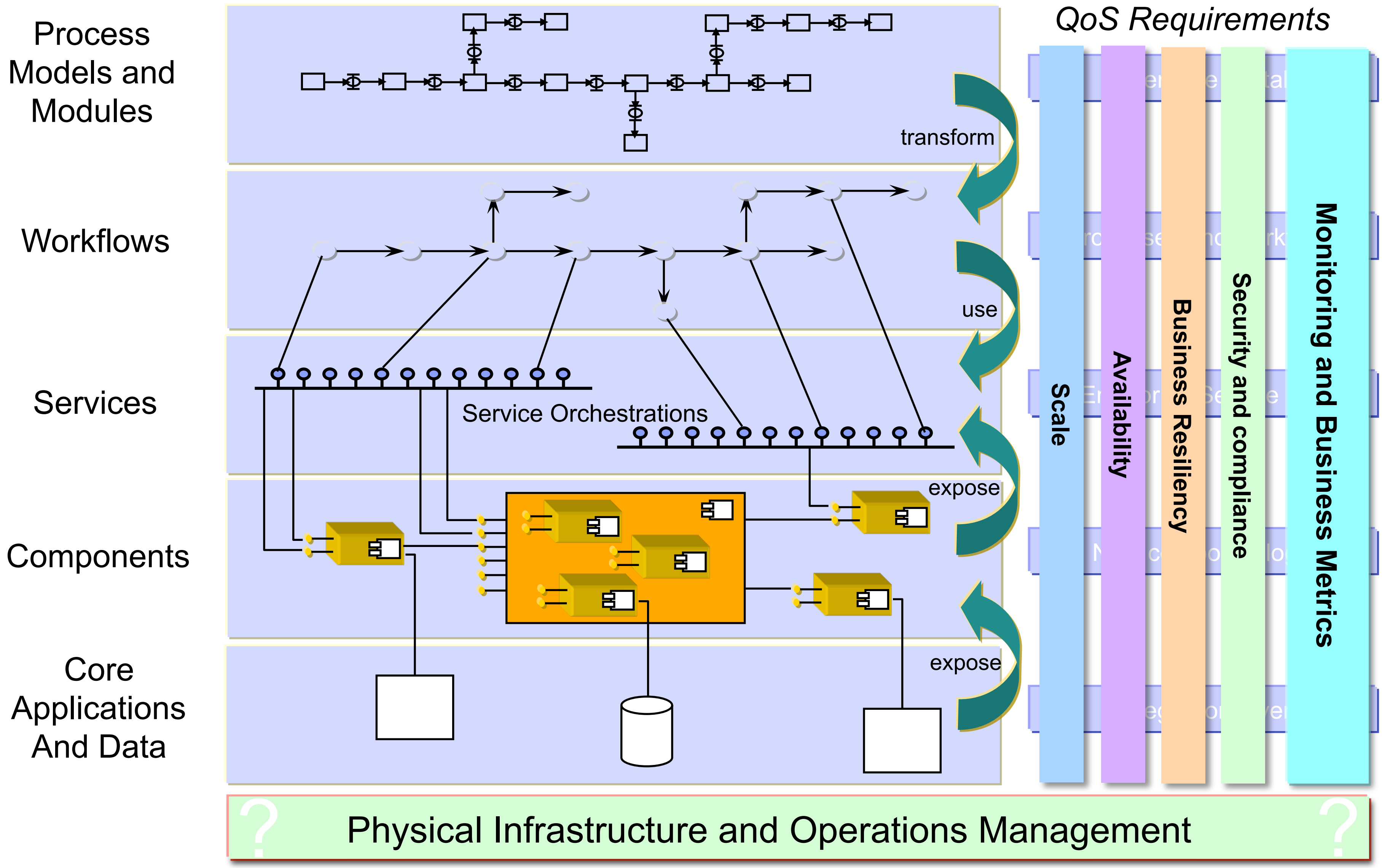
- The “Level of Ceremony” is based on
  - Maturity Level
  - Scope and impact of the project
  - Duration of the project

a methodology is like a Smörgåsbord…

SOA Design – application & infrastructure deployment



QoS requirements impact those decisions



Process Models and Modules

Workflows

Services

Components

Core Applications And Data

transform

use

expose

expose

Service Orchestrations

QoS Requirements

Scale

Availability

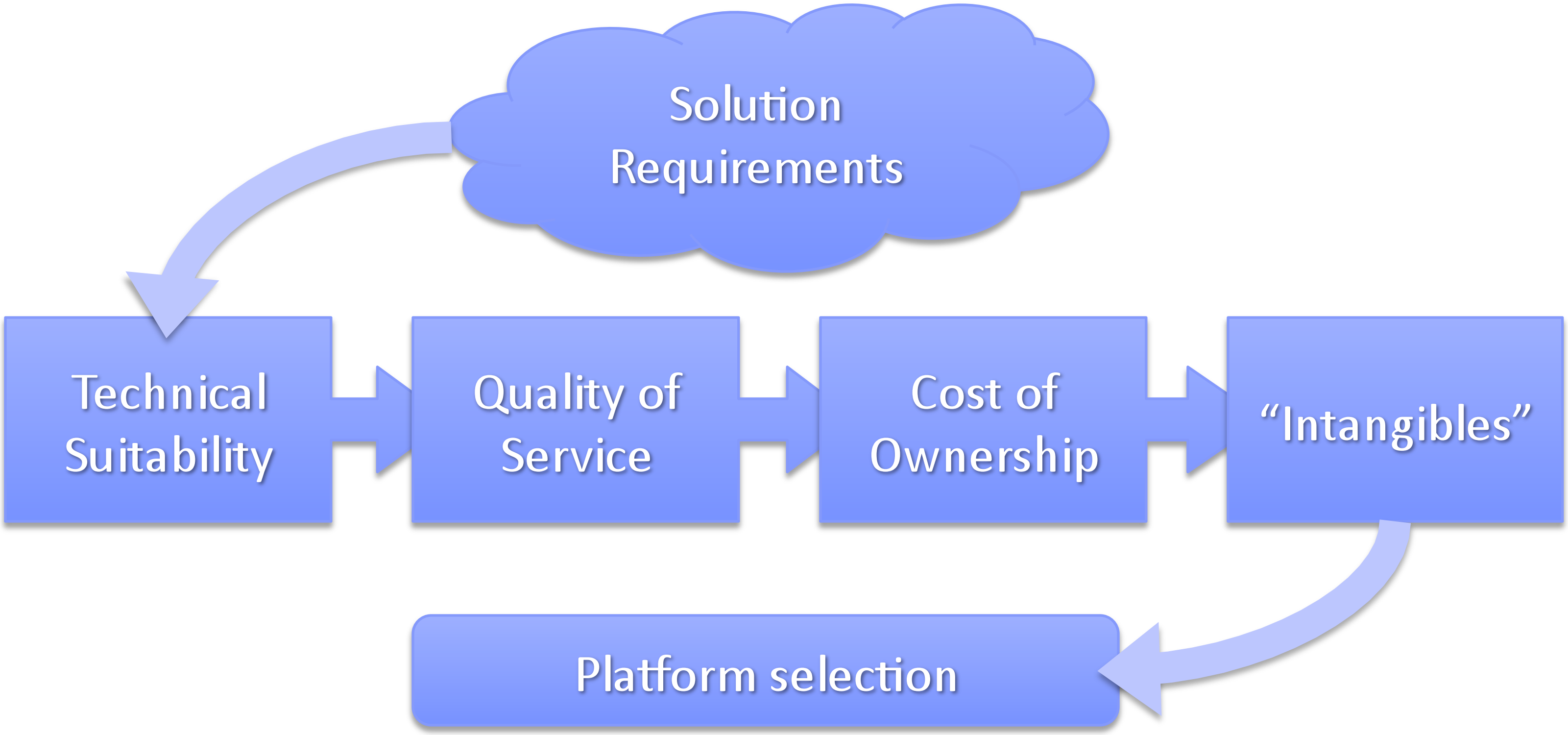
Business Resiliency

Security and compliance

Monitoring and Business Metrics

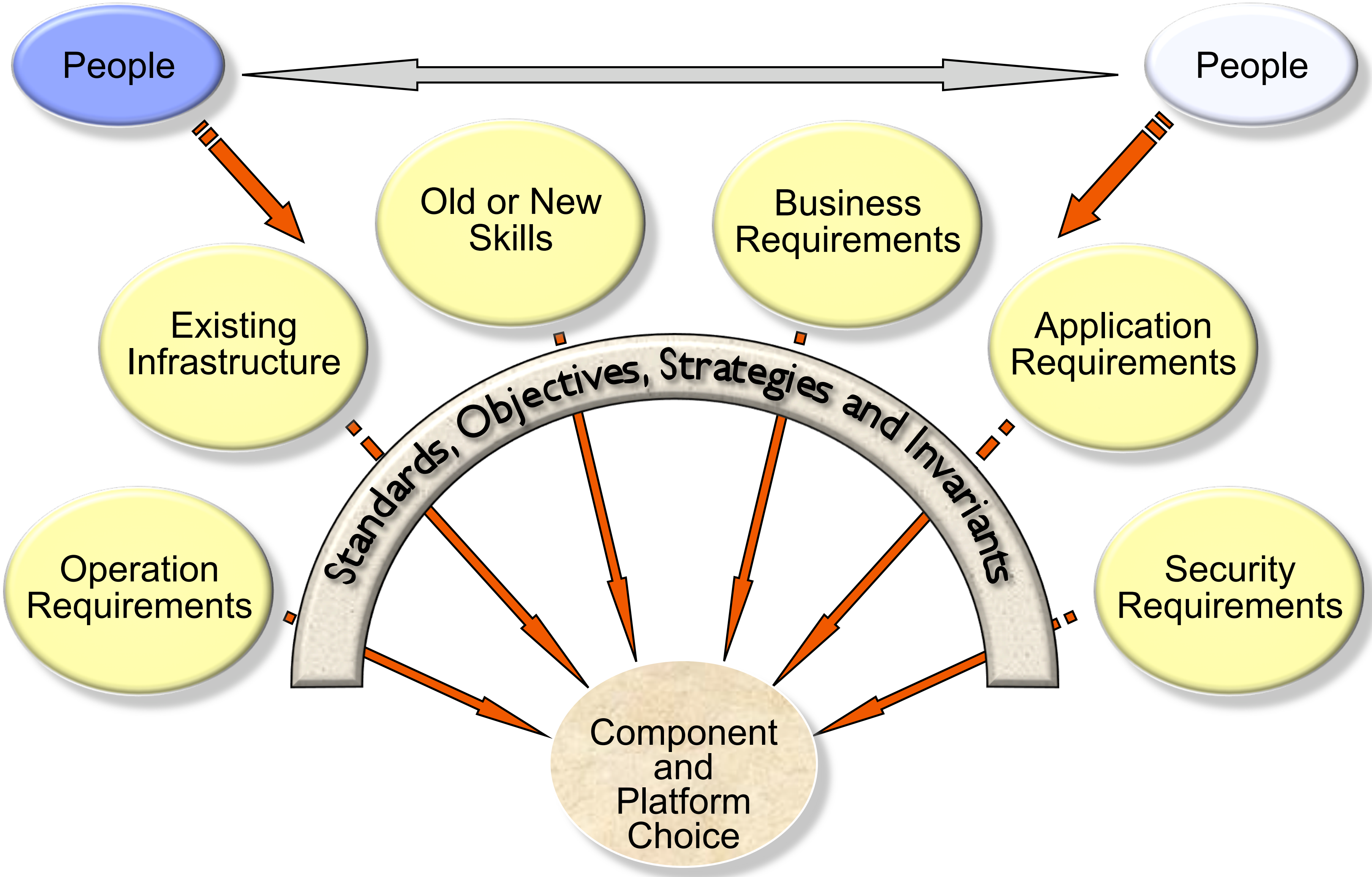
Physical Infrastructure and Operations Management

A high-level platform selection process/methodology



- Low-medium "ceremony" in the process
- Lots of details behind each of the boxes – we'll elaborate later
- There is no standard methodology for platform selection
  - Many different opinions, no particular consensus
  - This high-level approach is compatible with most architectural methods

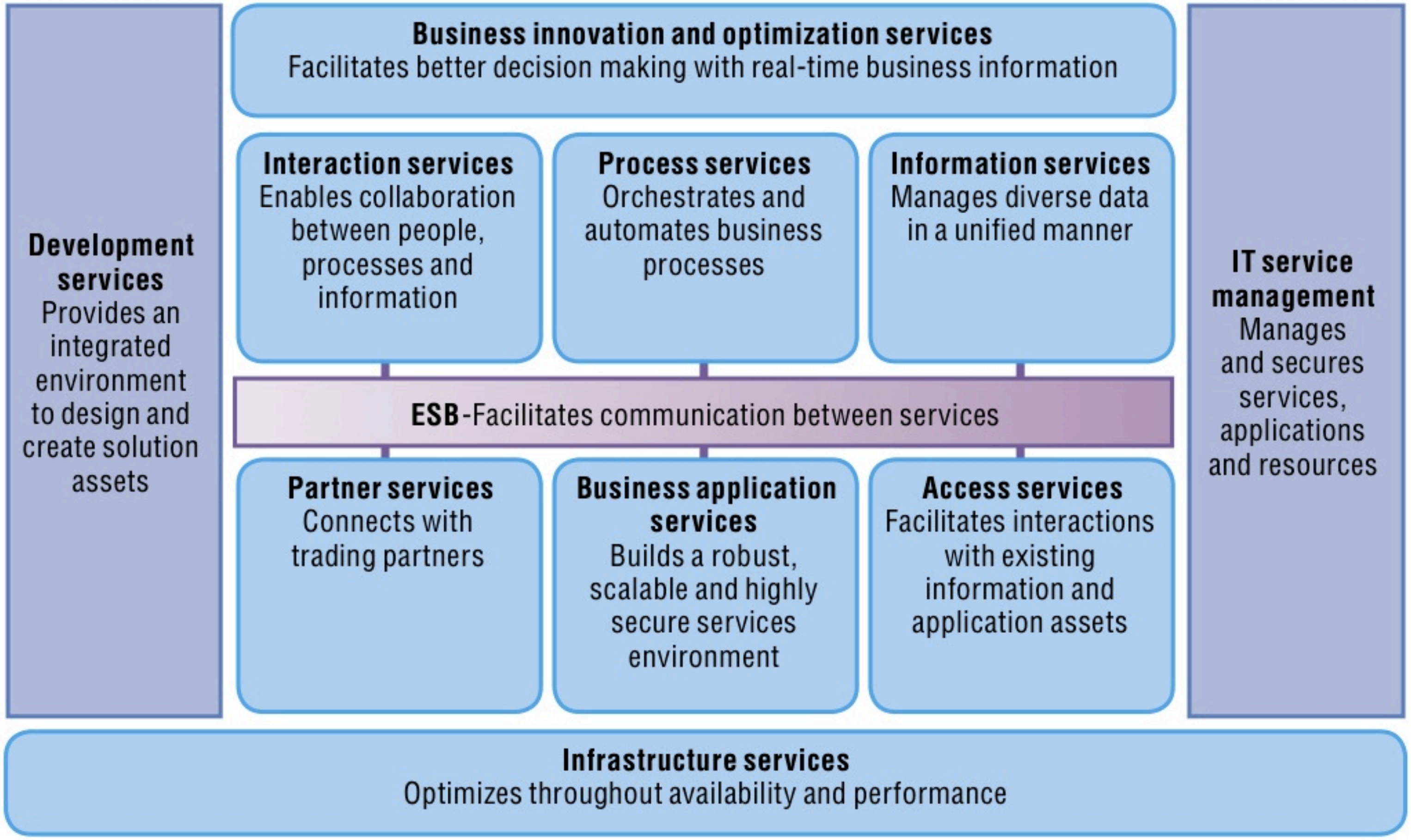
Another way of looking at it…



Agenda

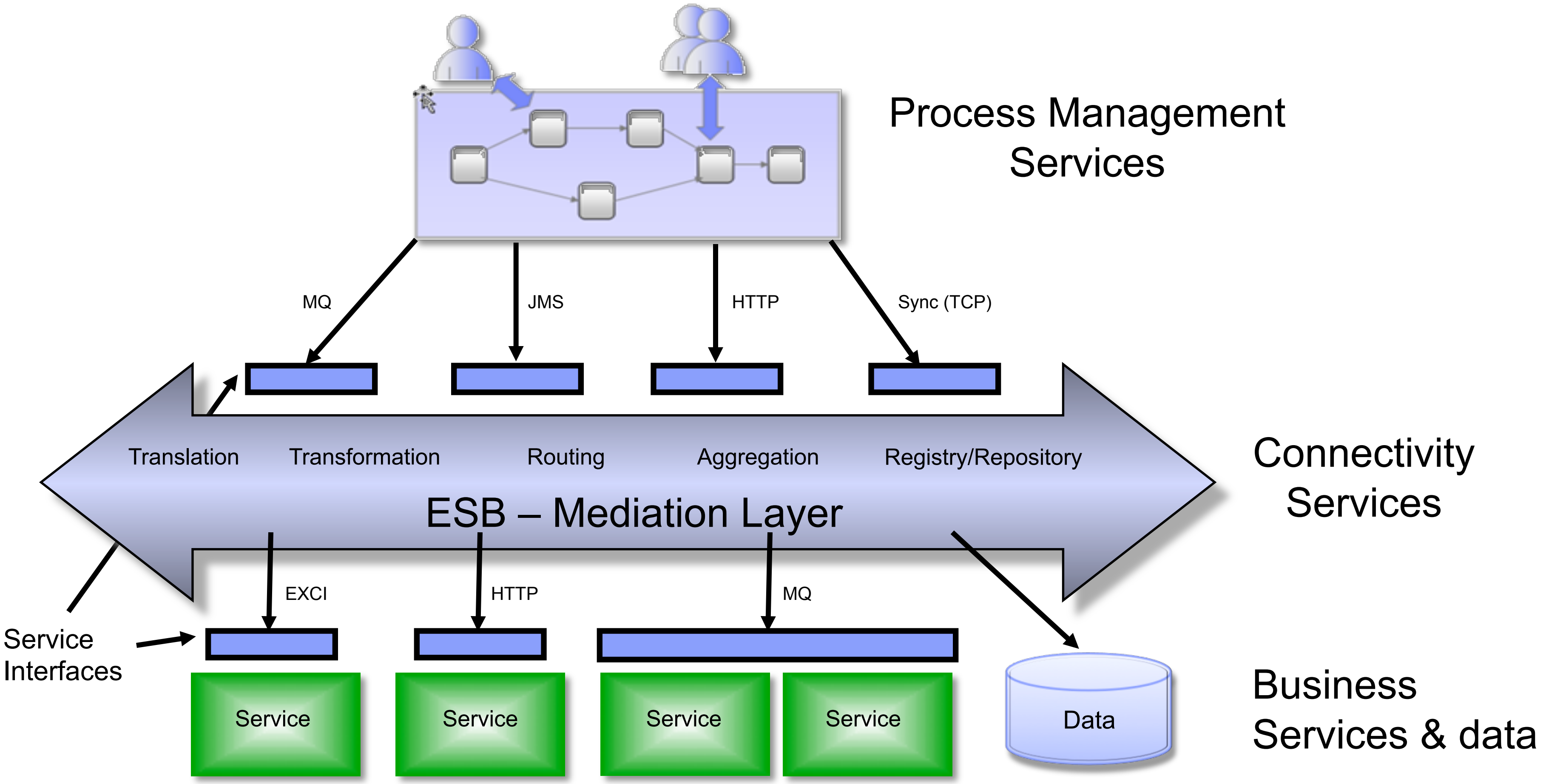
- Selecting computing platforms
- A SOA infrastructure reference architecture
- z/OS and Linux on System z
  - Architectural contrast
  - Strengths & weaknesses
- Which one?
  - Application suitability
  - Qualities of service
  - Cost considerations
  - “Intangibles”
- Examples

The IBM SOA Reference Architecture


Business innovation and optimization services
Facilitates better decision making with real-time business information
Development services
Provides an integrated environment to design and create solution assets
Interaction services
Enables collaboration between people, processes and information
Process services
Orchestrates and automates business processes
Information services
Manages diverse data in a unified manner
IT service management
Manages and secures services, applications and resources
ESB-Facilitates communication between services
Partner services
Connects with trading partners
Business application services
Builds a robust, scalable and highly secure services environment
Access services
Facilitates interactions with existing information and application assets
Infrastructure services
Optimizes throughout availability and performance

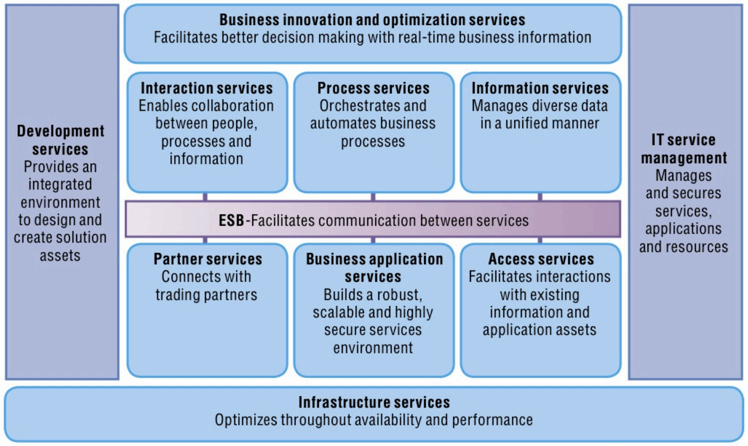
- The only one – there is not a special one for z or any other platform
- We will examine how System z is best used to host the products that back the functional components of the Reference Architecture

The basic infrastructure model for this discussion



Which products are we considering?

- We will not necessarily address these products individually, but they represent the products/workloads to keep in mind
  - **Interaction Services**
    - WebSphere Portal Server
  - **Process Services**
    - WebSphere Process Server
    - WebSphere Business Services Fabric
  - **Enterprise Service Bus**
    - WebSphere ESB
    - WebSphere Message Broker
    - WebSphere Service Registry & Repository
  - **Business Application Services and Access Services**
    - WebSphere Application Server
    - IMS, CICS & MQ Connectivity
  - **Information Services**
    - DB2, Enterprise Content Management



Business innovation and optimization services
Facilitates better decision making with real-time business information

Development services
Provides an integrated environment to design and create solution assets

Interaction services
Enables collaboration between people, processes and information

Process services
Orchestrates and automates business processes

Information services
Manages diverse data in a unified manner

IT service management
Manages and secures services, applications and resources

ESB-Facilitates communication between services

Partner services
Connects with trading partners

Business application services
Builds a robust, scalable and highly secure services environment

Access services
Facilitates interactions with existing information and application assets

Infrastructure services
Optimizes throughout availability and performance

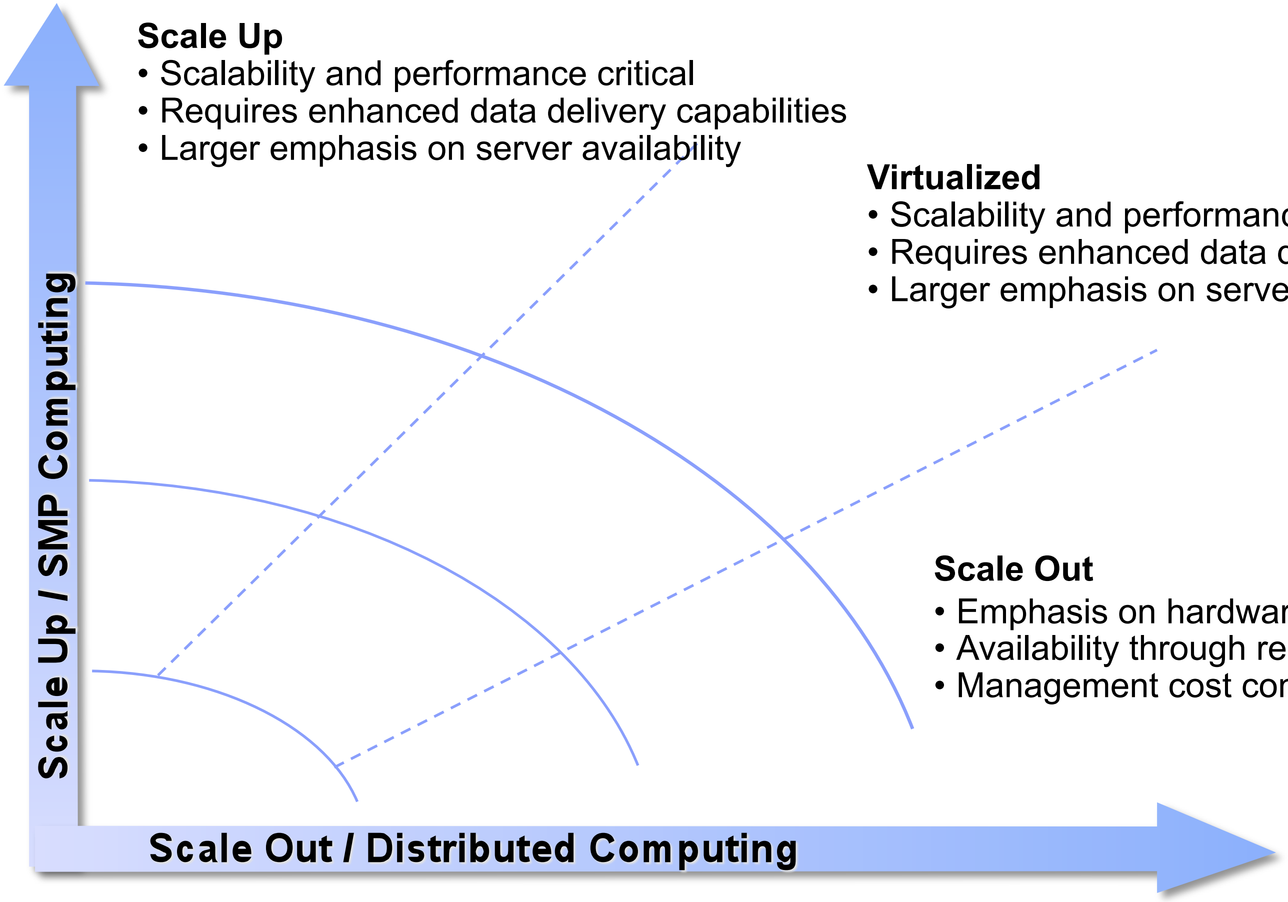
Lots of moving parts – what are we looking for?

- An SOA implementation often includes many different infrastructure components
  - Multiple artifacts to deploy to multiple places
  - Several potential points of failure and/or "slowdown"
  - What are the required service levels (SLA)?
    - Performance/response time
    - Availability window – serviceability and planned outages
    - Recovery time – consider component failures and major disasters
    - Scalability & responsiveness to capacity fluctuations
    - Skill & usability requirements – development & systems management
- Ideally, one should select the OS that supports the optimal match between the QoS requirements and what can be supported by that platform

Agenda

- Selecting computing platforms
- A SOA infrastructure reference architecture
- z/OS and Linux on System z
  - Architectural contrast
  - Strengths & weaknesses
- Which one?
  - Application suitability
  - Qualities of service
  - Cost considerations
  - “Intangibles”
- Examples

Server Design Concepts



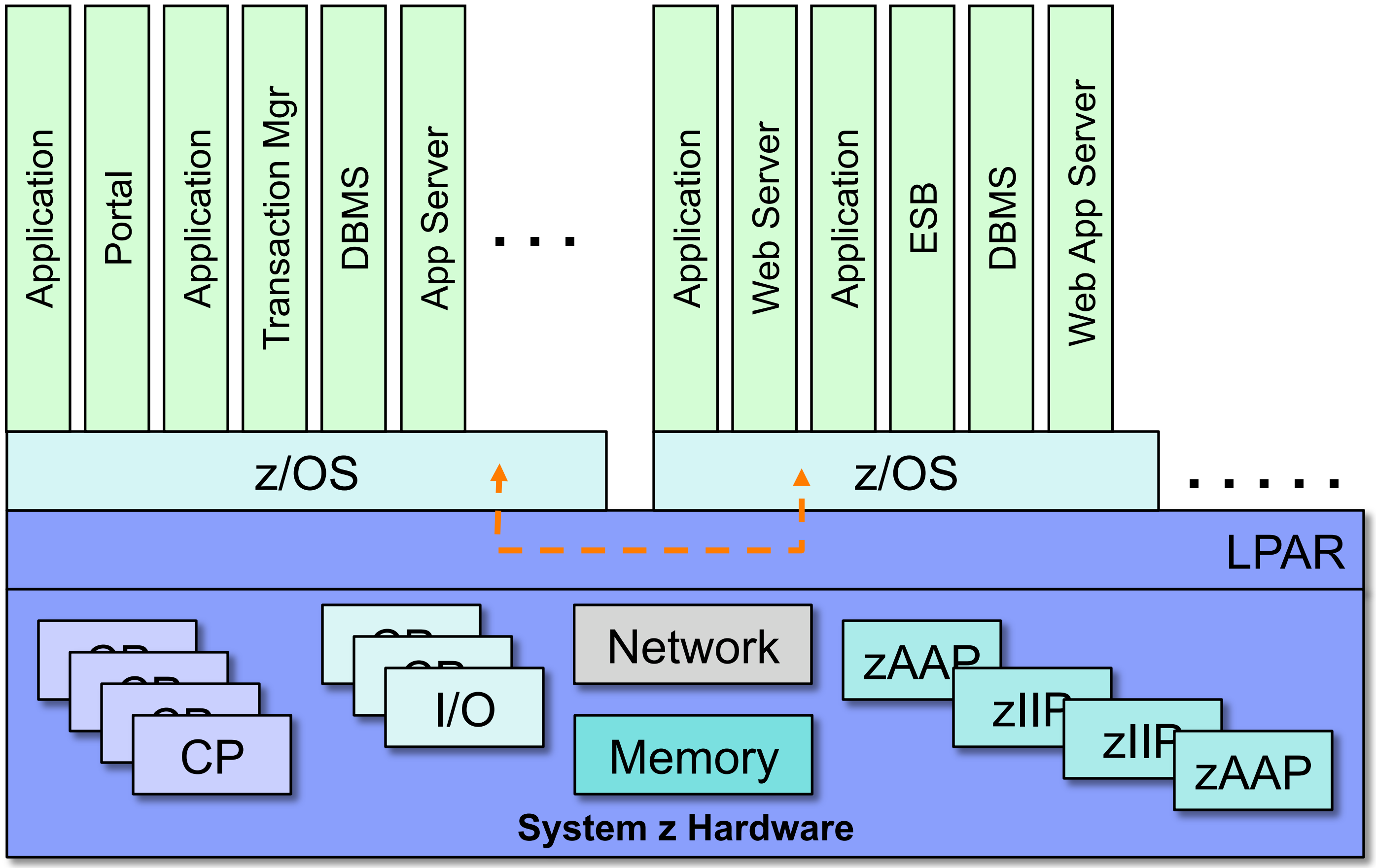
Scale Up
- Scalability and performance critical
- Requires enhanced data delivery capabilities
- Larger emphasis on server availability

Virtualized
- Scalability and performance critical
- Requires enhanced data delivery capabilities
- Larger emphasis on server availability

Scale Out
- Emphasis on hardware acquisition costs
- Availability through replication
- Management cost considerations

System z qualifies in all three categories – scale up with large single image z/OS (or even Linux), scale out with Parallel Sysplex, and virtualize with LPARs and z/VM

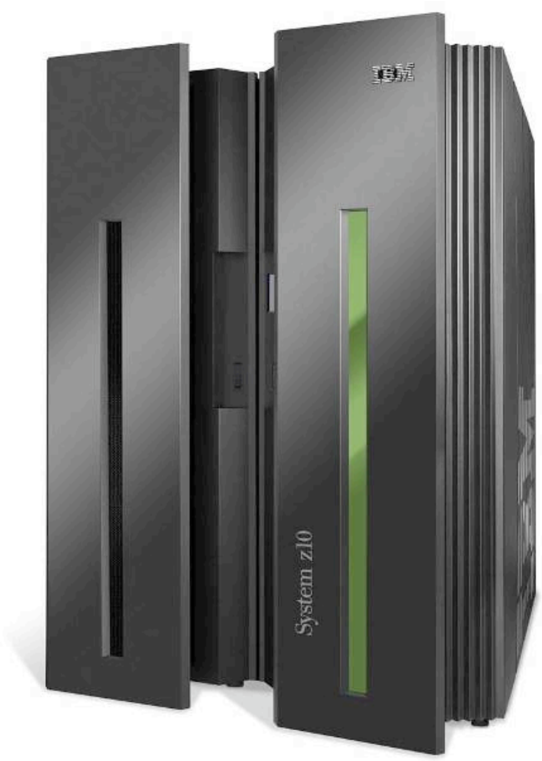
z/OS – an Application Virtualization architecture



- Multiple apps and middleware per OS image
  - Benefits from proximity between components – performance, simplicity, reliability
- Multiple OS instances (LPARs) per System z box/frame
- Networking between LPARs with Hipersockets (< - - - - - - - >)

"Specialty Engines" provide lower total cost

- Special assist processors for System z
  - For Java workloads (zAAP)
  - For selected DB2 workloads (zIIP)
  - For Linux workloads (IFL)
  - For Sysplex Coupling (ICF)



- Lower hardware, software pricing
  - Hardware is $125K per processor one time charge
    - $125K for a ~920 MIP processor (z10)
  - No charge for IBM software running on zAAP/zIIP
  - IBM software running on IFL pays 120 PVUs
  - Free upgrade to next generation!

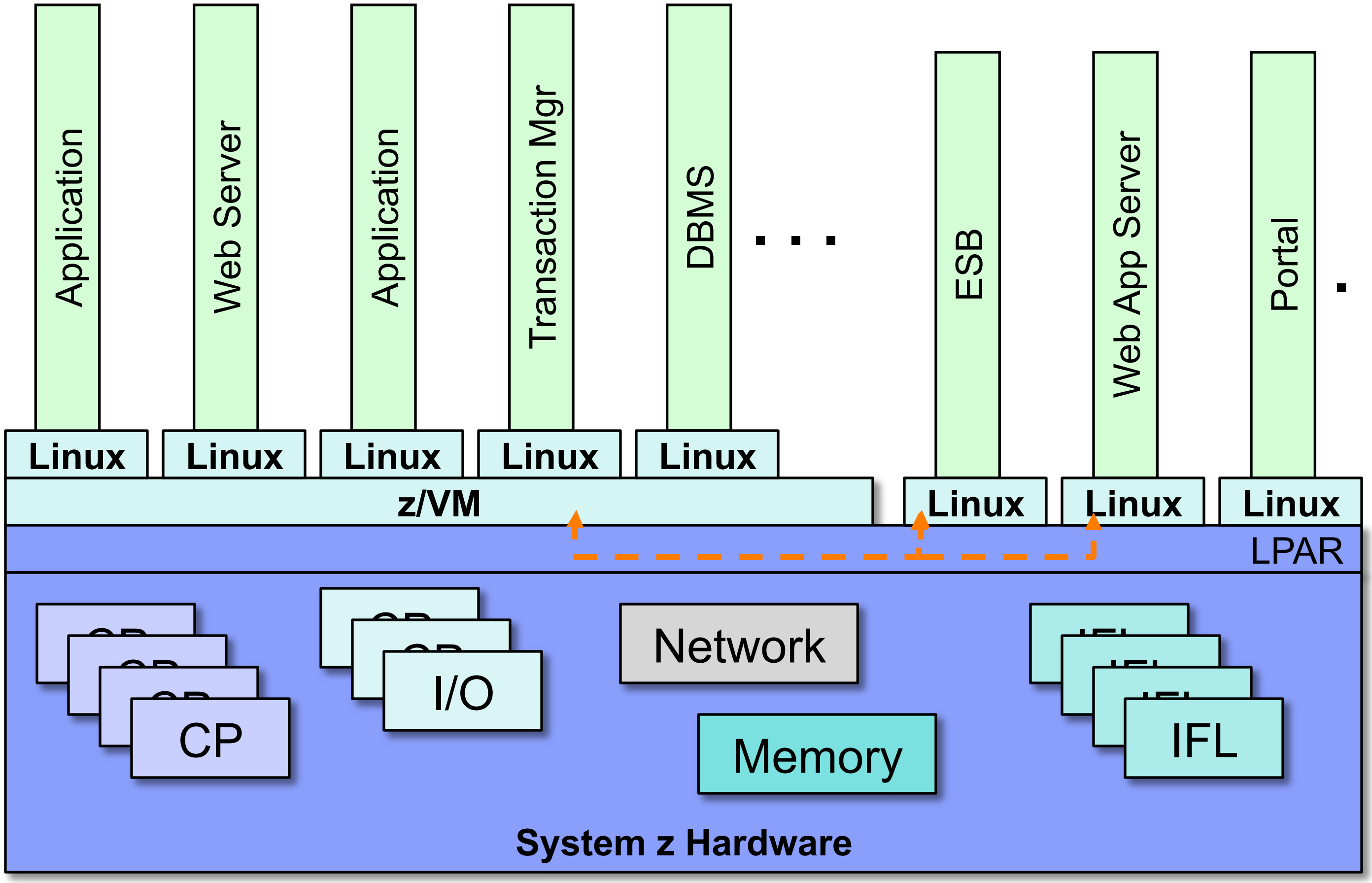
- Specialty Processors are for pricing – they do NOT make the work run faster
  - Same "silicon" on the chip
  - Microcode changes to limit the workloads on the processor

Benefits of z/OS

z/OS delivers an unprecedented experience for enterprise worthy businesses, featuring:

- Automated business resiliency & disaster recovery
  - Parallel Sysplex, ARM, GDPS, Capacity on Demand
- OS designed for high availability and outage avoidance (planned and unplanned)
- z/OS WLM for prioritization and classification of workloads
- Vertical scalability with support for 64+ procesors
- Dynamic response to changing LPAR demands with Intelligent Resource Director (IRD)
- Granular control and centralized logging for detailed snapshots of the Sysplex
- Advanced change management and support for production and back-up environments
- Support for zAAP and zIIP specialty engines to optimize cost of new workloads
- Integration with core middleware
  - DB2, CICS, WebSphere, IMS, WebSphere MQ, etc.

Linux on System z – a server virtualization architecture



- A distributed architecture implemented in a System z frame
  - Generally one function per Linux instance, like most distributed server implementations
- Benefits derived from drastically lower environmental, floor space expense, network efficiency & performance
  - Networking between Linux instances with Hipersockets (< - - - - - - > above) or z/VM VLAN,
- z/VM virtualization flexibility, ease of instance management (provisioning, monitoring), security

Benefits of Linux on System z

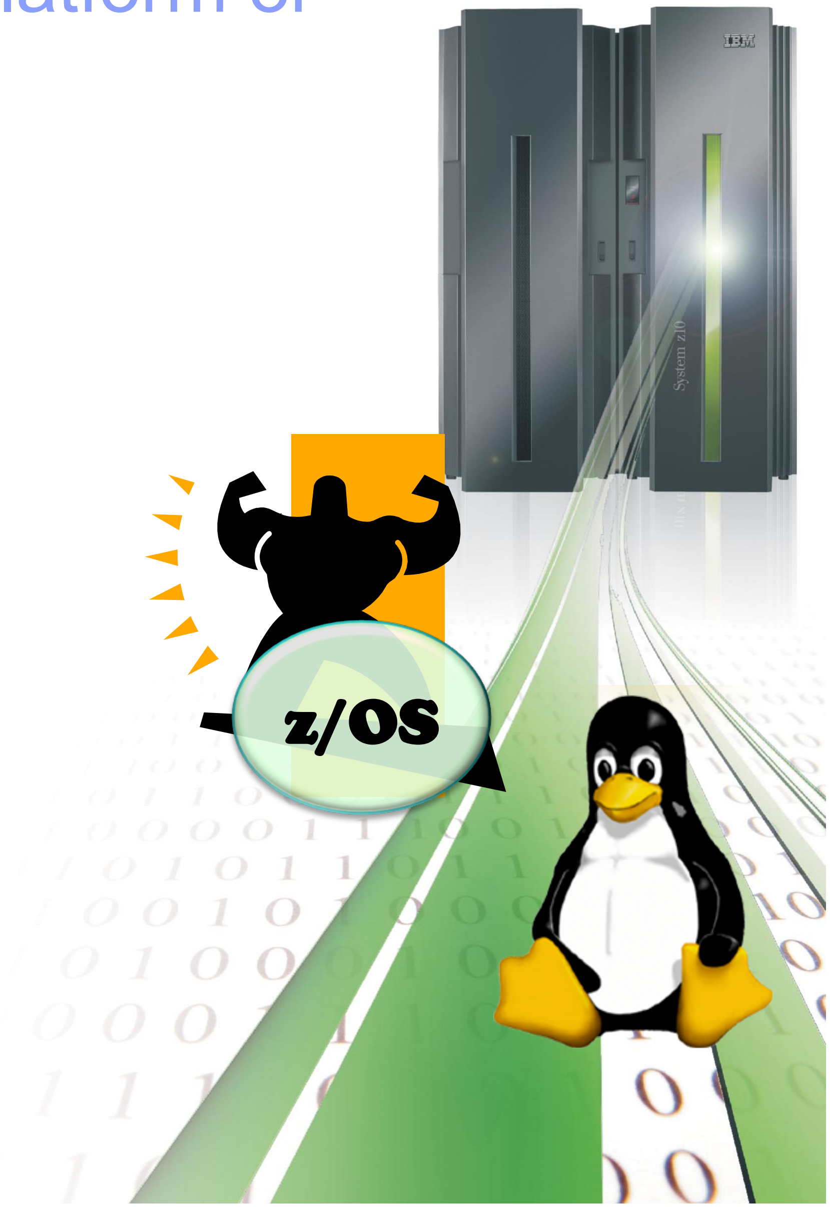
- Provides an industry-leading virtualization platform that delivers numerous benefits:
  - Linux enables you to run a common operating system on every server in your enterprise
  - Hundreds of Linux guests on a single System z box
    - Dramatic cost savings from environmentals – floor space, power, cooling, etc.
  - Support for many ISV applications, middleware, out of the box
  - The same Linux distributions as on distributed – familiar environment to administrators and developers
    - Easy to administer and use – well-known tools & utilities
  - Leverages inherent features of System z hardware and underlying VM and LPAR virtualization
    - System z hardware reliability
    - System z hardware support (DASD, networking, tape, etc.)
  - Disaster recovery of "distributed" Linux guests far easier than a distributed server farm
  - When combined with z/VM, easy creation and provisioning of Linux guest server instances

Agenda

- Selecting computing platforms
- A SOA infrastructure reference architecture
- z/OS and Linux on System z
  - Architectural contrast
  - Strengths & weaknesses
- Which one?
  - Application suitability
  - Qualities of service
  - Cost considerations
  - “Intangibles”
- Examples

Common factors considered in platform or OS selection

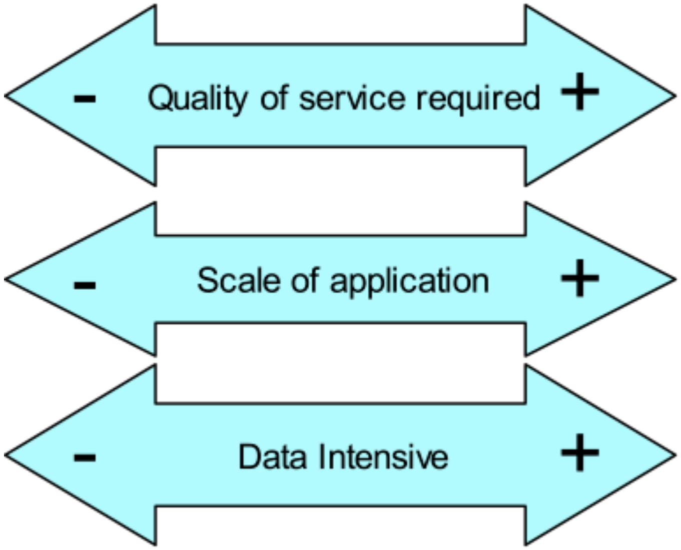
- Performance / scalability
- Existing application heritage
- Proximity to existing applications & data
- Reliability/Availability
- Security
- Skills
- Cost
- Politics/”ideology”



Some general considerations

Hardware considerations

Intel Unix System z

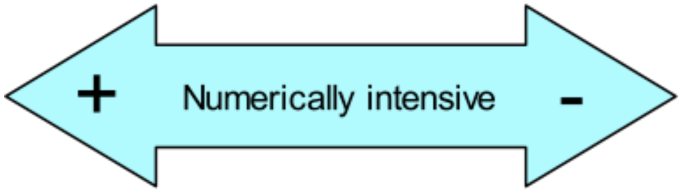


- Quality of service required +

- Scale of application +

- Data Intensive +

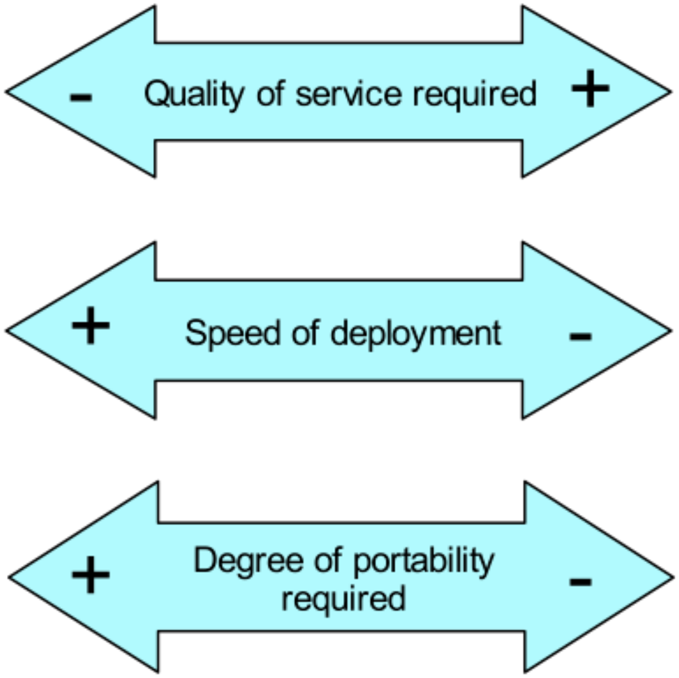
Unix/RISC Intel System z



+ Numerically intensive -

Software/OS considerations

Linux on System z z/OS



- Quality of service required +

+ Speed of deployment -

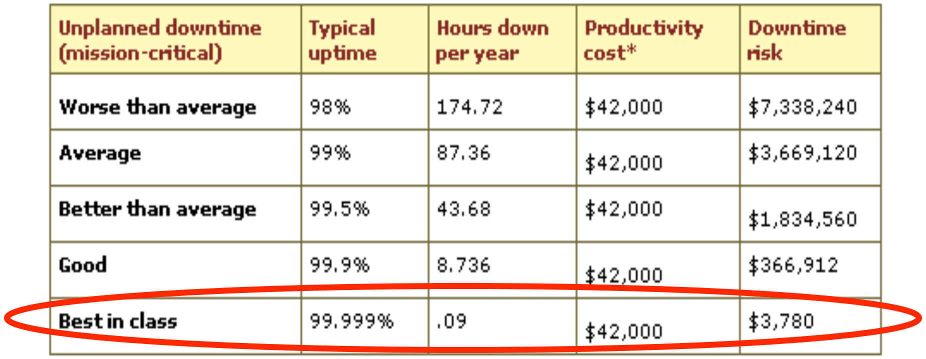
+ Degree of portability required -

System z Qualities of Service and Nonfunctional requirements

- The mainframe is known for delivering high levels of reliability, scalability, security and manageability
- But, “all roads lead to cost of ownership”
  - Delivering higher levels of QoS usually maps back to cost considerations
  - It costs more to deliver higher service levels
  - Outages, security breaches, and wasted resources – all cost money

- What are the QoS requirements for an SOA implementation?
  - What level of performance, security, etc. is required for the services themselves?
    - Often represented by a Service Level Agreement, or SLA
  - What is the SLA for the SOA infrastructure?
    - For example, the ESB is the cornerstone of the SOA infrastructure – should the ESB not reside on the most reliable, secure infrastructure available?

Parallel Sysplex & GDPS deliver superior “five nines” availability and extreme horizontal scalability for z/OS



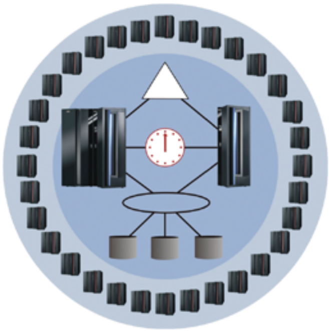
Unplanned downtime (mission-critical)	Typical uptime	Hours down per year	Productivity cost*	Downtime risk
Worse than average	98%	174.72	$42,000	$7,338,240
Average	99%	87.36	$42,000	$3,669,120
Better than average	99.5%	43.68	$42,000	$1,834,560
Good	99.9%	8.736	$42,000	$366,912
Best in class	99.999%	.09	$42,000	$3,780

SOURCE: ALINEAN *Per unplanned downtime hour for typical user group

Single System

- Built In redundancy
  - Spare chips, cards, processors, etc.
- Capacity upgrade on Demand
- Capacity backup
- Hot pluggable I/O

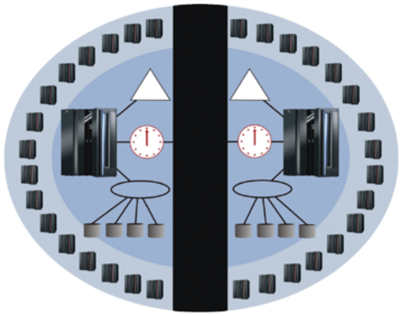
Parallel Sysplex



Up to 32 Systems

- Addresses planned/unplanned HW/SW outages
- Flexible, non-disruptive growth
  - Capacity beyond largest CEC
  - Scales better than SMPs
- Dynamic workload/Resource management
- Verifiable & repeatable recovery

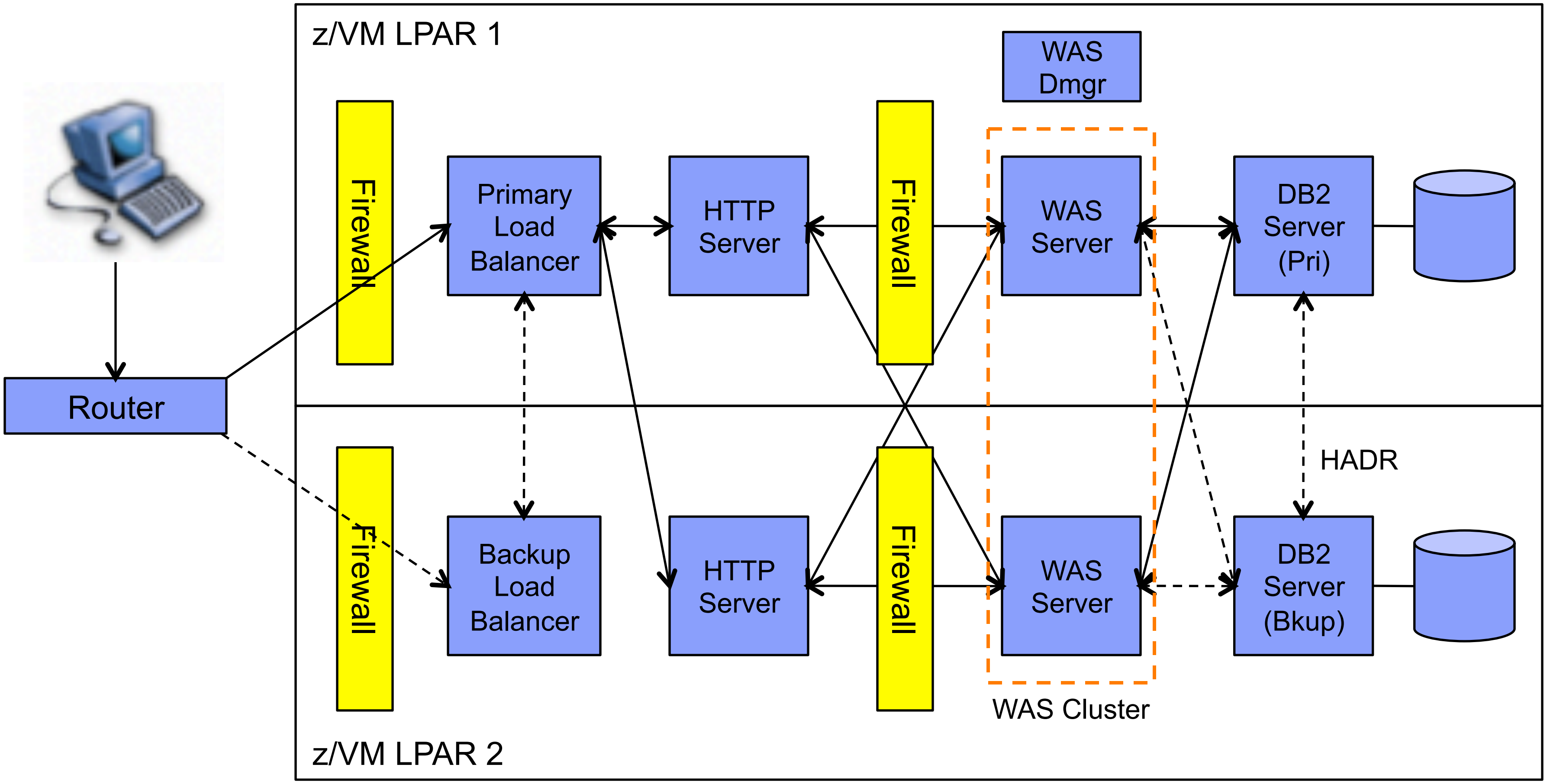
GDPS



Site 1 Site 2

- Addresses site failure & maintenance across GEOs
  - Regular test validation
- Sync/Async data mirroring
  - Eliminates tape/disk SPOF
  - No/some data loss
- Application independent
- Addresses global distance recovery

Clustering with Linux on System z – a cluster and network in the box



Each colored rectangle is a separate Linux guest (VM or LPAR)

- Clustering/load balancing using inherent functions of middleware components
- No built-in OS clustering/WLM feature (like Parallel Sysplex on z/OS)

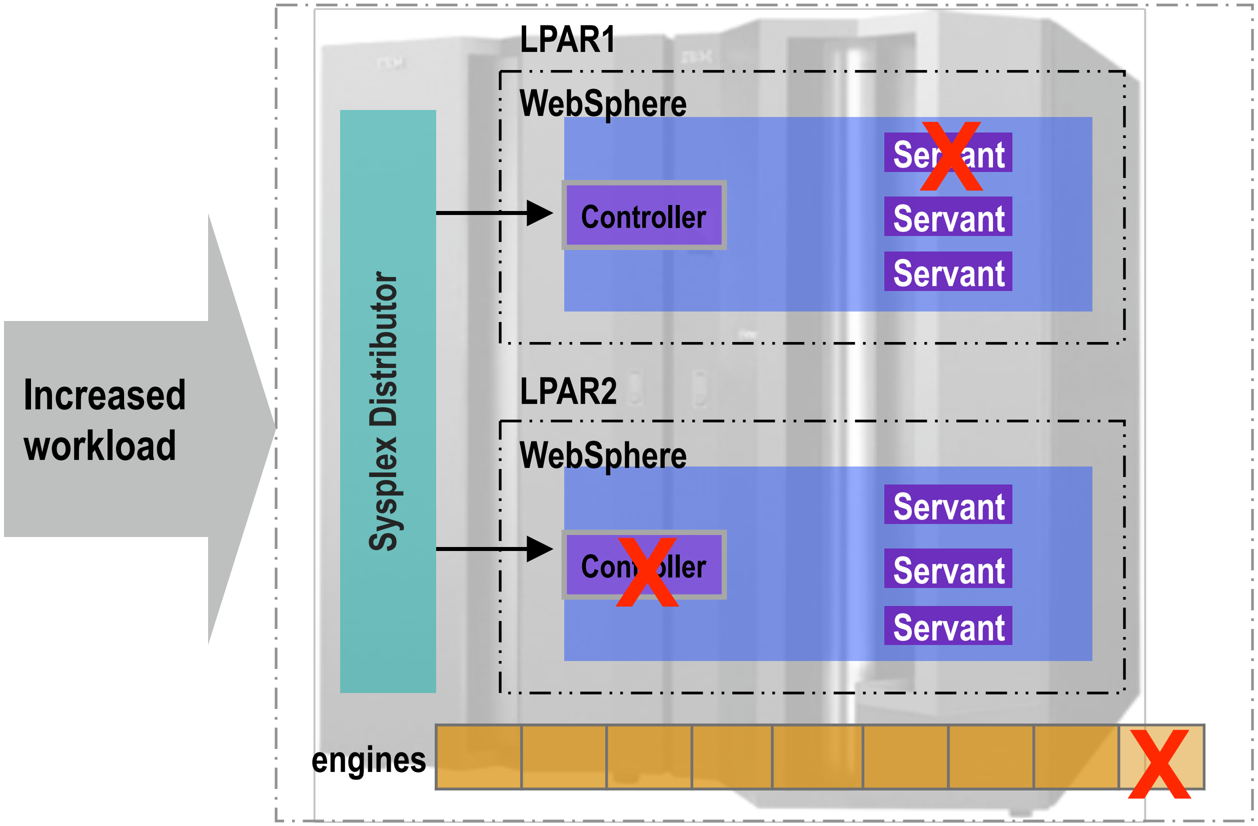
What does resiliency mean to your business

Main Entry: **re·sil·ient**

a: capable of withstanding shock without permanent deformation or rupture

b: tending to recover from or adjust easily to misfortune or change

Using an application server middleware example, on z/OS:



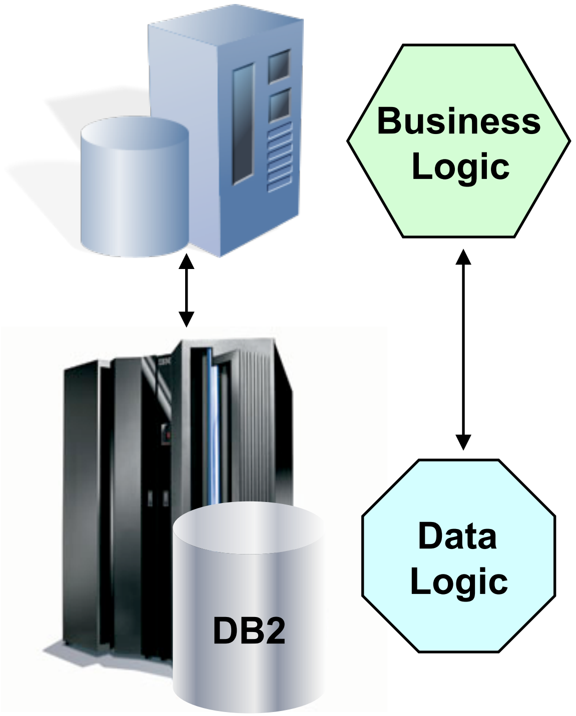
Self Optimizing:

- Auto-responsive to changes in demand
- Support for differentiation of workload priorities

Self Healing:

- Auto-recover from failures with integrity
- Isolation and response to component failures

The implications of proximity (or lack thereof…)

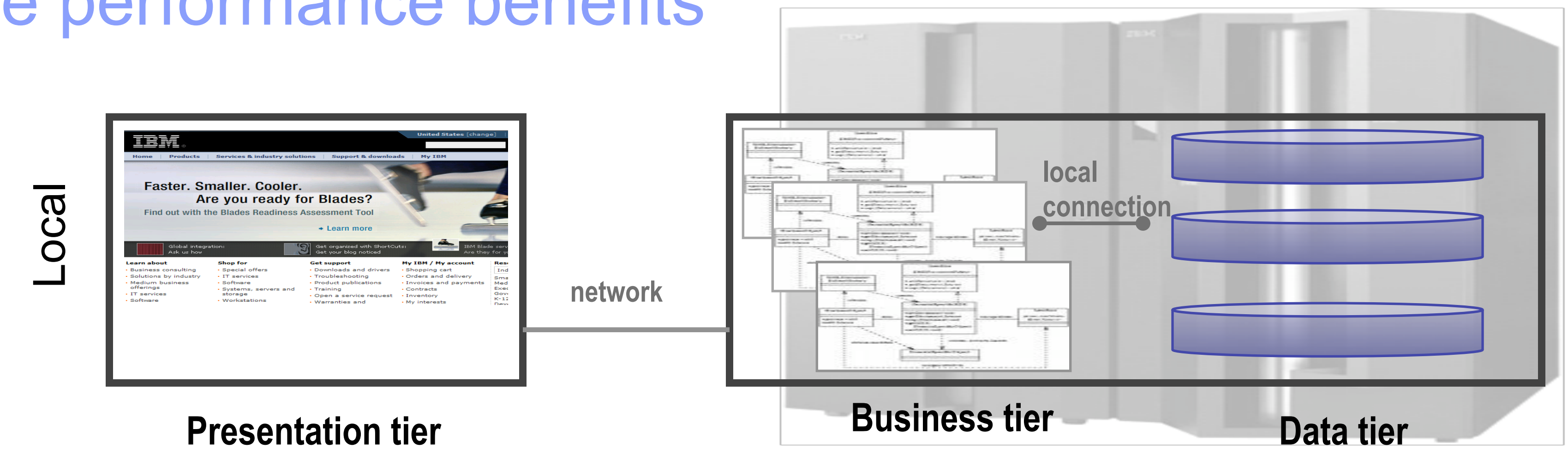


- Objects are converted into byte array at the requester (CPU, time)
- **Network latency is incurred (time)**
- More latency is incurred as service is dispatched (CPU, time)
- Objects are reconstructed at the server (CPU, time)
- Requested data is retrieved
- Objects are converted into byte array at the server (CPU, time)
- **Network latency is incurred (time)**
- Objects are reconstructed at the requester (CPU, time)

Some other considerations:

- # of interactions between the tiers, volume of data passed
- No local optimizations of the access protocol
- Effect on server memory requirements due to locking

The performance benefits



- **No network time**
- **Less network protocol construction / deconstruction**
  - DRDA (DB2); CICS Transaction gateway (CICS); RMI / IIOP (IMS & J2EE)
- **Reduced CPU utilization**
- **High speed access to cache without network**
  - Parallel Sysplex infrastructure infrastructure for DB2, CICS, IMS, MQ
  - Guaranteed integrity and currency of data (Pessimistic data access)
- **Improved two phase commit performance**

Mainframe security in action

The **Network**

- z/OS Communications Server with integrated intrusion detection capabilities

The **System z9** server

- EAL 5 Common Criteria certified LPARs
- System z cryptographic technology featuring the Crypto Express2
- HiperSockets security benefits compared to communication alternatives like TCP/IP
- Storage Protection Keys can granularly protect address spaces
- IP Security with zIIP engines

The **z/OS** operating environment

- EAL 4+ Common Criteria certified multi-level security
- Public Key Infrastructure (PKI) services
- Centralized system level auditing to know who is trying to access your system and when

The **z/OS** operating environment cont’d

- Resource Access Control Facility (RACF) integrates with z/OS to provide centralized enterprise level security & auditing up through the software stack
- EAL4+ for Controlled Access Protection Profile (CAPP) and Labeled Security Protection Profile (LSPP).
- Tivoli support for z/OS security, including access, identity, compliance, et al

The **WebSphere** application infrastructure

- Java 2 Platform Enterprise Edition (J2EE) security
- EAL4 Common Criteria certified
- Federal Information Processing Standards (FIPS) 140-2 Compliance

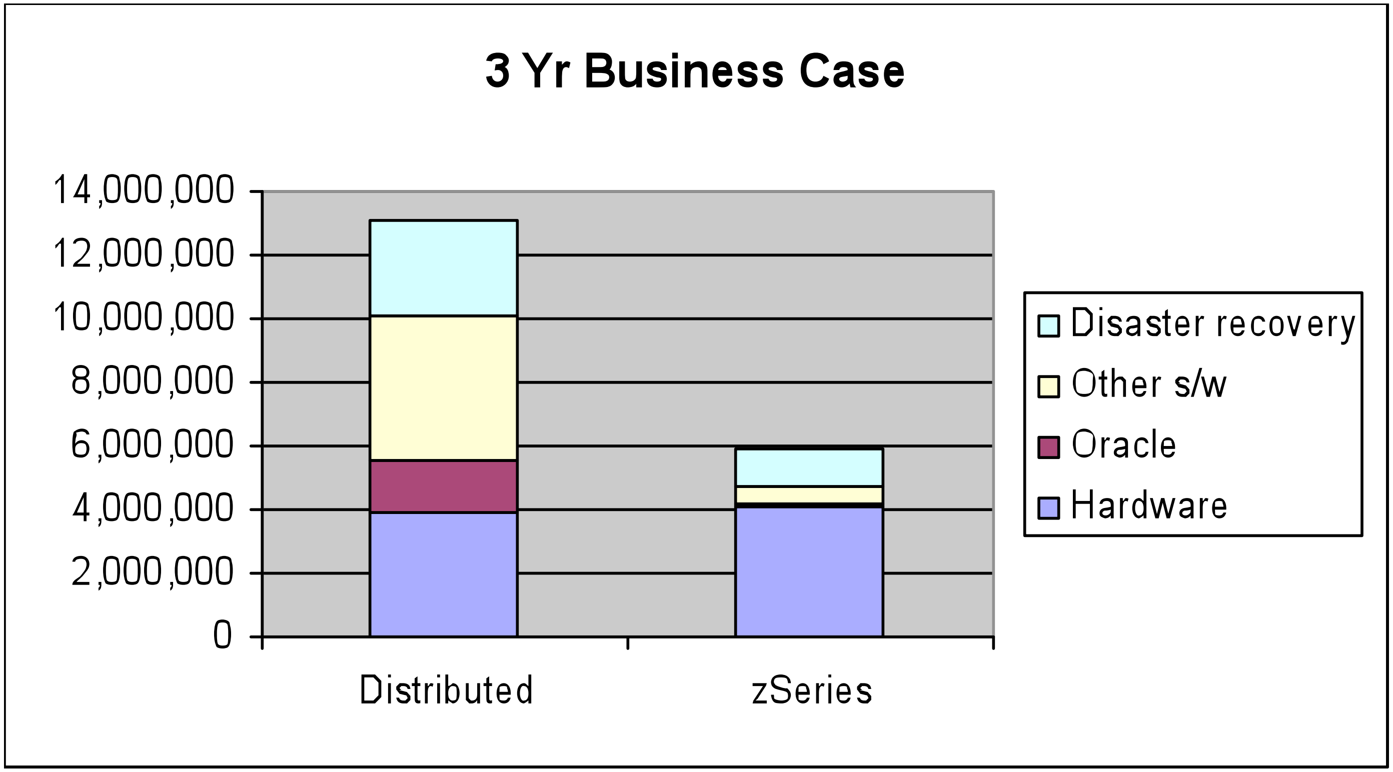
The **Data Protection** realm

- Data encryption solutions for data at rest and data in flight
- Data integrity

Total I/T Cost of the solution

- Every decision on platform selection somehow relates to cost of the solution
  - Does the solution require 100% availability? Are you prepared to pay for that?
  - Does the solution require encryption at all levels, including data-at-rest? How much will that implementation cost?
  - Do you require the qualities of service, given the cost required?
- System z is often considered the most expensive platform option, but “cost of acquisition” is usually the evaluation point
  - When other costs are considered, including management cost, environmentals, cost of outages/security breaches, etc. are considered, System z often (usually) emerges as the low-cost option.
- Many IBM clients cite software cost (often ISV packages) as their biggest cost pain
  - Customers are often unaware of the actual cost of distributed software and management cost of distributed implementations.
- In some cases, System z is not only lowest cost of ownership, but can even be the lowest in cost of acquisition (hardware & software)
  - Specialty Engines (zIIP, zAAP, IFL) have changed the equation considerably – more on those later…

A sample client business case for platform cost



- 226 distributed CPs to 16 z9 IFLs
- Net new all-IFL z9. Adding IFLs to existing System z would reduce the cost
- Refresh of old distributed machines not included
- People, floor space, power, cooling not included

Source: Presented by IBM & Confirmed by Customer at SHARE, San Diego, Aug/07, Session 1842

The “intangibles”

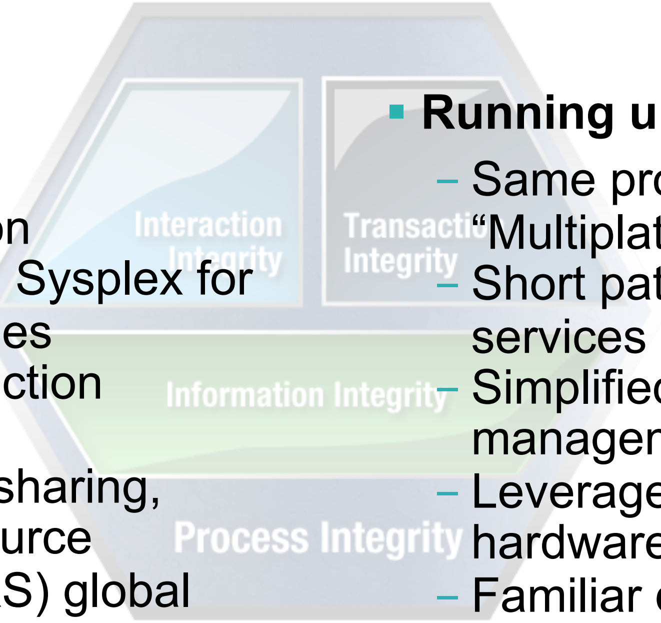
- Skills
  - Is the tech support staff sufficiently skilled on z/OS vs. Linux?
  - Do the application & infrastructure architects have a thorough understanding of System z and the operating systems?
  - Are you hiring new hires that understand Linux (or z/OS)?
- Strategy
  - Some organizations have a documented platform placement strategy, and sometimes it’s simply “habitual”
    - What is the official <u>and</u> unofficial (or unspoken) platform strategy?
    - Is there any existing Linux or z/OS implementation on the mainframe?
- Standards
  - Examine your documented I/T standards for specifications on OS or platform
  - Are there defacto standards in place that drive decisions
- Ideology & personal preference
  - Often the strongest driving force in platform and OS selection
  - Could be as simple as a key decision-maker’s personal experiences
  - This presentation is intended to equip you to remove some of the subjectivity in the decision process – but the ideology issue will never go away…

Agenda

- Selecting computing platforms
- A SOA infrastructure reference architecture
- z/OS and Linux on System z
  - Architectural contrast
  - Strengths & weaknesses
- Which one?
  - Application suitability
  - Qualities of service
  - Cost considerations
  - “Intangibles”
- Examples

Process and Connectivity with SOA software for System z

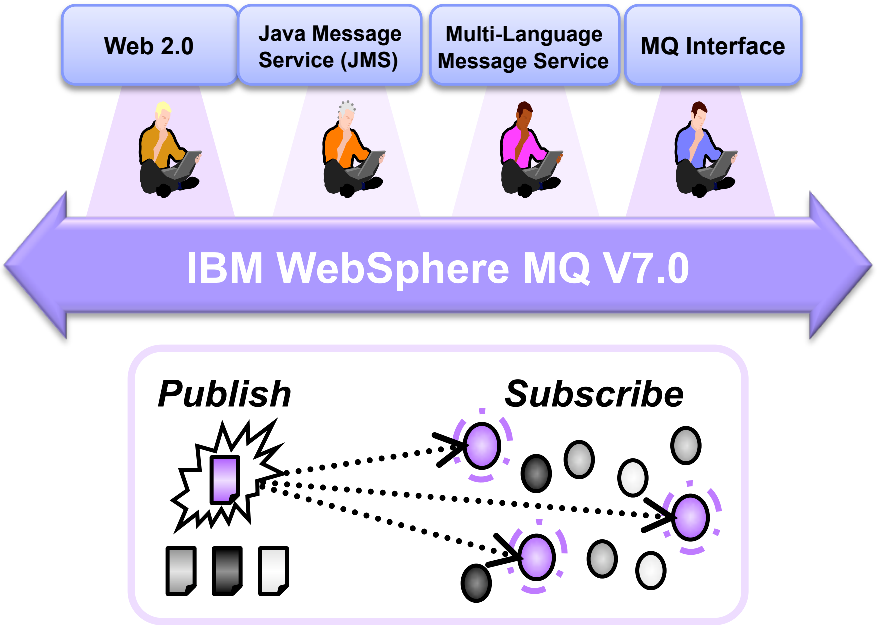
– WebSphere MQ for z/OS, WebSphere Message Broker for z/OS, WAS for z/OS, WebSphere ESB, WebSphere Process Server:



- **Running under z/OS:**
  - Fully ARM-enabled
  - z/OS WLM participation
  - Exploitation of Parallel Sysplex for with MQ Shared Queues
  - Heterogeneous transaction coordination
  - Support for DB2 data sharing, CICS EXCI, and Resource Recovery System (RRS) global transaction coordination
  - RACF for integrated security
  - Reporting and Chargeback with SMF

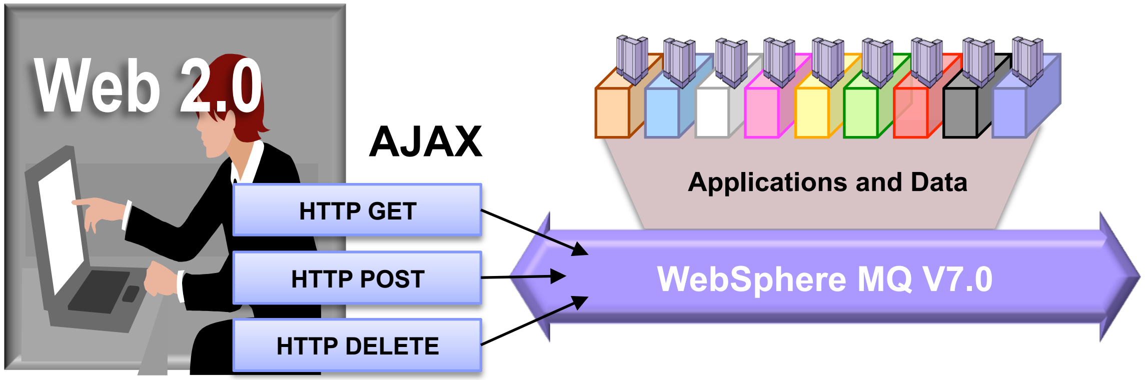
- **Running under Linux on z:**
  - Same products as LUW "Multiplatform"
  - Short path length to z/OS services and infrastructure
  - Simplified distributed server management
  - Leverage reliability of z hardware platform
  - Familiar distributed architecture
  - Flexible deployment options
  - Familiar Linux OS environment and admin tools

WebSphere MQ as a SOA transport

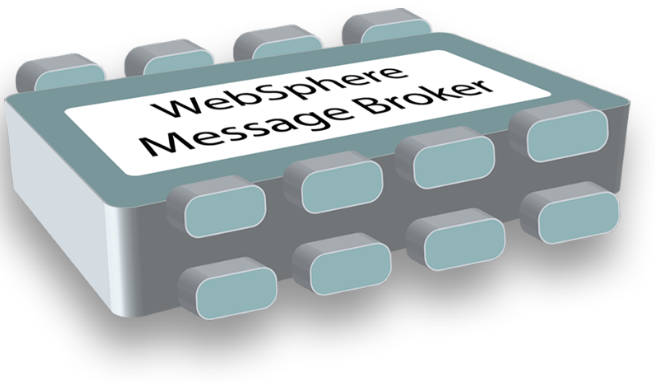


Universal Messaging Backbone for SOA and Web 2.0

- Either z/OS or Linux on z
- Point-to-point or ESB protocol
- Now supports HTTP access to queues
- Continues to leverage z/OS facilities
- MQ a good candidate for Linux on z

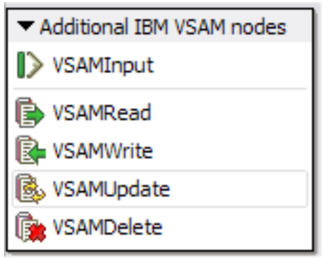


Advantages of WebSphere Message Broker for z/OS



- **High Availability**
  - Takes full advantage of Parallel Sysplex aware resource managers (e.g. WebSphere MQ)
  - Fully ARM-enabled
  - Supports WebSphere MQ clustering
- **z/OS-Specific Connectivity:**
  - VSAM
  - QSAM
  - CICS

CICS Request


▾ Additional IBM VSAM nodes

- VSAMInput
- VSAMRead
- VSAMWrite
- VSAMUpdate
- VSAMDelete

- **Workload Management**
  - Goal-oriented resource allocation
  - Workload scaling, workload isolation
- **Reporting and Chargeback**
  - SMF
  - Coordinated reporting (ENF37)
- **Option to extend transformation options with WebSphere Transformation Extender for Message Broker**
  - Data enhancement
  - Complex, many-to-many transformation

Candidate workload examples for Linux on System z

- Customers frequently consider these for movement from distributed systems to System z
  - WebSphere MQ Series queue managers
  - WebSphere Application Server (WAS)
  - Domino/Notes servers
  - “Gateway servers”:
    - DB2 Connect
    - CICS Transaction Gateway,
    - IMS Connect
  - SAP
  - WebSphere and JAVA applications development environments
  - Network Infrastructure (FEP replacement)
  - “Utility servers” – FTP, NFS, DNS, etc..,
  - Oracle Database
  - Infrastructure with rigid disaster recovery req’s
  - Virtualization and Security Services

- If these are already running on z/OS, there is seldom any reason to move them, and there is often feature/function advantage to running on z/OS

Industry: Education
URL: http://cms.bsu.edu/

"SOA has been such a gift to us. It enables us to embrace a new technology that provides services at a level that we couldn't even imagine before."
–Dr. O'Neal Smitherman

BALL STATE UNIVERSITY
EDUCATION REDEFINED

Ball State University

Ball State University bridges disparate systems and solves key administrative issue with IBM SOA solution.

CHALLENGE

- Coordinate 40 name and address systems to streamline administrative processes and ensure information integrity for users

SOLUTION

- SOA with WebSphere Enterprise Service Bus on z/OS to connect silo'd applications without hand-coding individual API calls (WESB, CICS TS, System z)

BENEFITS

- Ability to develop and implement services in an SOA environment for resolving name and address discrepancies in 10 months, as opposed to several years for hand-coding individual application connections
- Confidence that IBM solution can lead to wider use of SOA to further streamline administrative business processes
- Services created here can be reused in later SOA efforts

Nationwide® On Your Side™

…embraces mainframe economics & saves $15M over three years with Linux on System z

Before adopting a mainframe Linux strategy …

- 78% of Nationwide's distributed servers had peak utilization of less than 50%
- Server provisioning time was measured in weeks
- Dynamically allocating processing power was difficult, for example:
  - Some applications needed more processing power on Fridays
  - Development wanted performance test environment for only 8 weeks
- Data center power and floor space scarcity – 10s of millions of dollar investment

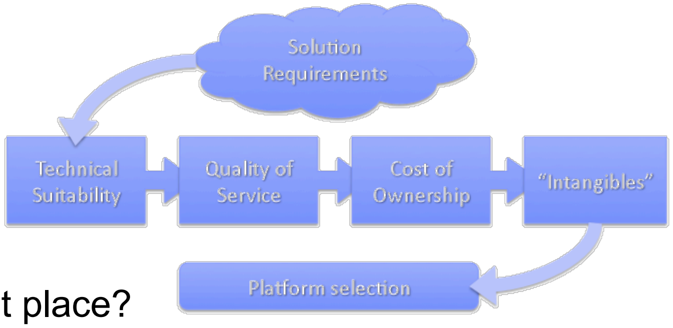
The benefits of Linux on System z enabled Nationwide to …

- Virtualize 350 servers (HTTP, WebSphere, Portal, etc.) on 2 mainframes with 15 IFLs and supported by a staff of 3
- Reduce Web hosting monthly costs by 50%
- Conserve floor space & power consumption by 80%
- Reduce hardware & OS support efforts by 50%
- Significantly save on middleware software costs
- Fast deployment (4 months) and significantly faster provisioning speed
- Dynamic allocation of compute power eliminates need to "over-provision"

Listen to the LinuxWorld Extreme Virtualization session hosted by **Guru Vasudeva**, ***Nationwide VP & Chief Architect***: http://www.linuxworld.com/events/keynotes/lwsf06-guru.html

To summarize…

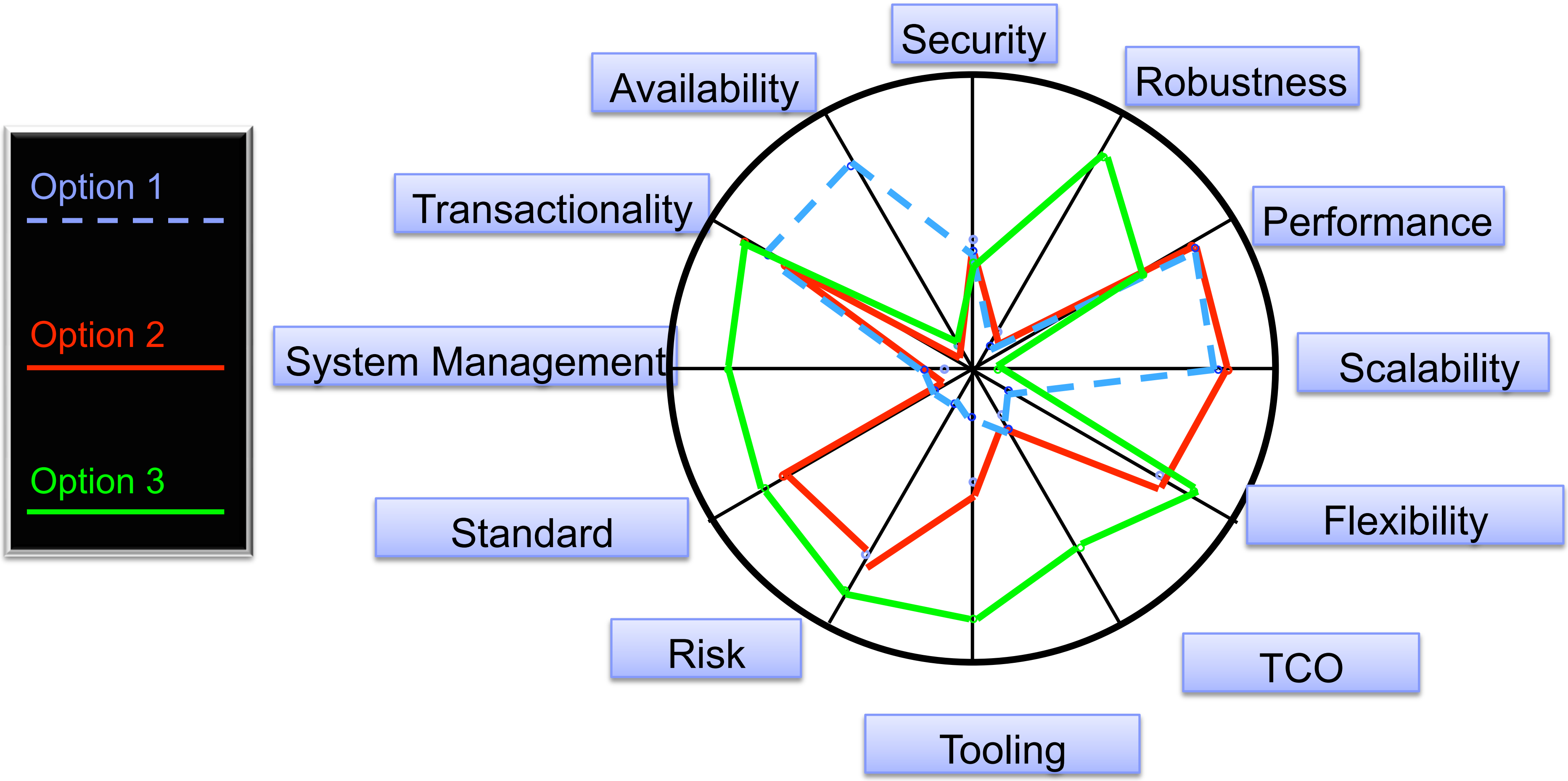
Making the decision



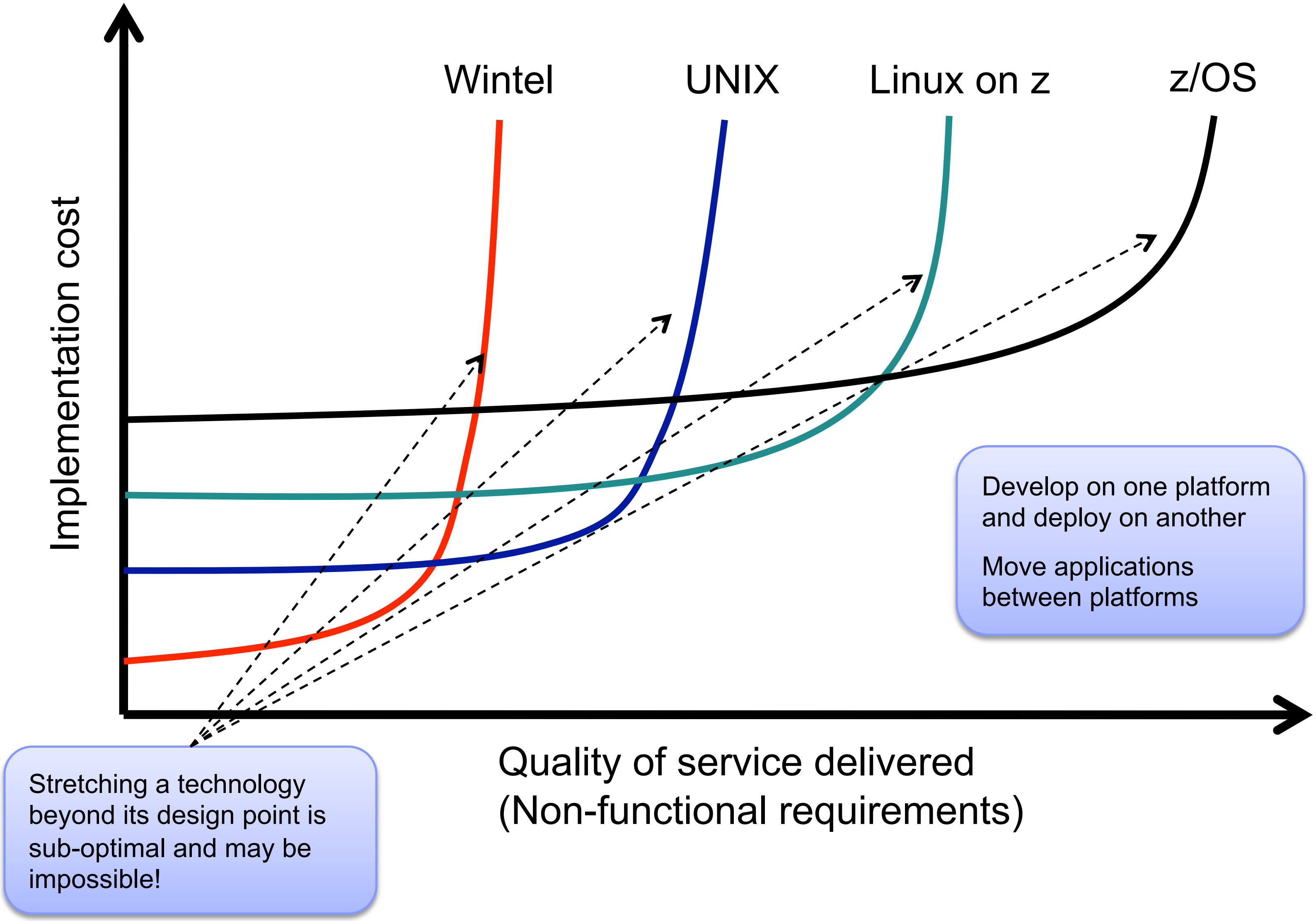
- *Think back to the high-level evaluation flowchart:*
  - **Technical suitability**
    - Is this a component that's well-suited for System z in the first place?
      - Either z/OS or Linux can be evaluated once the System z platform is confirmed
  - **Quality of Service**
    - What are the security, availability, scalability, disaster recovery, etc. requirements for our SOA implementation?
      - z/OS generally considered higher QoS operating system than Linux
  - **Cost of ownership**
    - What is the true cost of operating the SOA solution on z/OS versus Linux on z?
    - What is the business cost impact of higher qualities of service delivered versus a lower-cost infrastructure?
    - Are we evaluating ALL costs, or just the acquisition cost?
  - **"Intangibles"**
    - What are our standards – documented and defacto?
    - What skills are available on z/OS versus Linux?
    - What's the ideological climate and position on things like open source, Linux, etc.?

- Do you have your architectural decisions sufficiently documented?
  - Most architecture methodologies suggest a format for documenting decisions

Comparing using evaluation criteria



Use the right technology for the right job…



Develop on one platform and deploy on another

Move applications between platforms

Stretching a technology beyond its design point is sub-optimal and may be impossible!

* - A single server. A large solution with many servers may be a different story…

Thank you…

Questions?