Fit For Purpose:

Selecting the Right Platform

S. Michael Benson
Executive I/T Architect
smb@us.ibm.com
845-435-1554

Some Fundamentals

- **Because they are programmable servers and are fundamentally "general purpose"**
  - *it is "a small matter of programming" to make any server do any task.*
- **Because of the above there is a great deal of overlap in server functionality.**
- **There are 3 fundamentals of differentiation**
  - Fitness for functionality – does the code support this platform and/or that platform
  - Fitness for non-functional requirements – how well does it run here and/or there
  - Fitness to meet local needs – How well does this or that meet MY needs HERE
- **Any rational and reasonably objective view of this subject will determine that "one size does not fit all".**

The right ‘tool’…All of these tools can move a person from one place to another…real fast….

Lear Jet 60 (Corporate)

Capacity = 7 (8 with belted toilet)

Range = 2,691 miles

Cruise Speed = 514 mph

MD – 90 (Regional)

Capacity = 153

Range = 2,400 miles

Cruise Speed = 503 mph

Boeing 747-400 (Large Capacity)

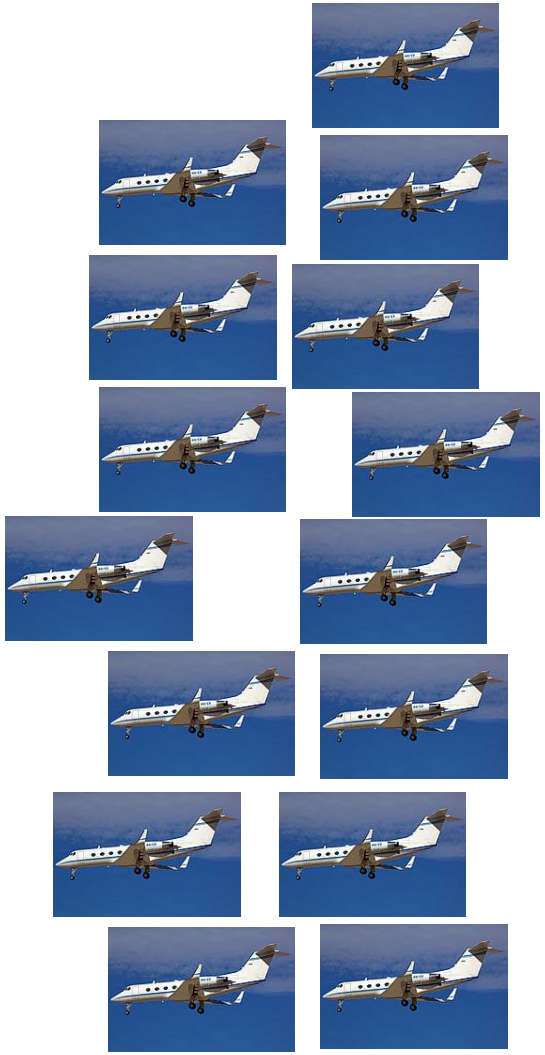
Capacity = 420

Range = 8,827 Miles

Cruise Speed = 563 mph

Each tool offers varying levels of capabilities…

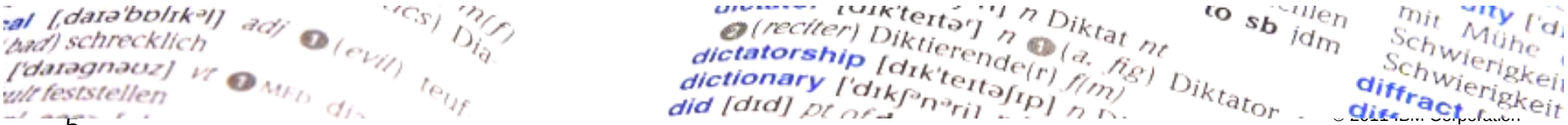
But…which is the right tool… to move 1 person? 100 people? 400 people?



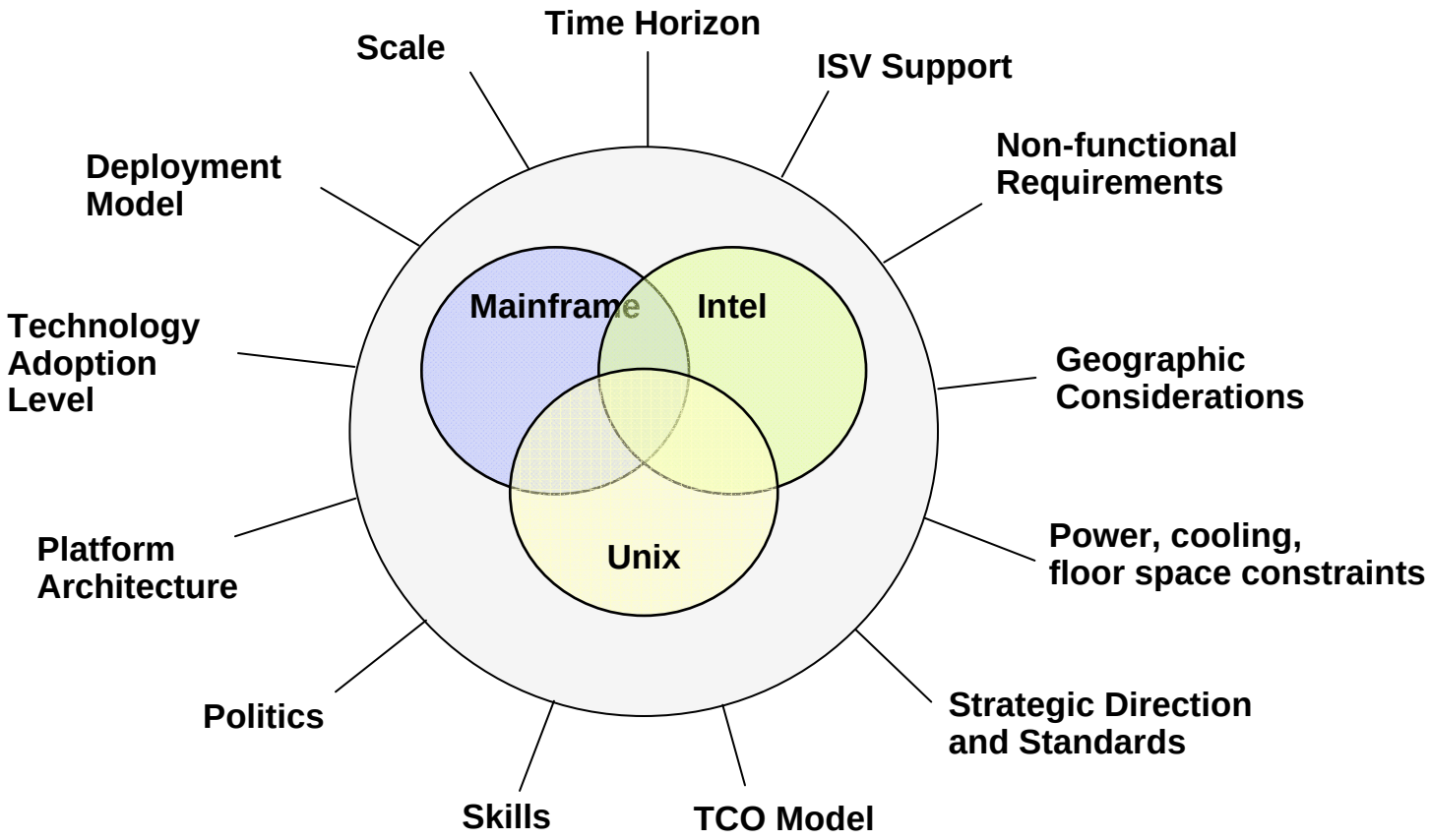
High Level Workload Definition

- **Workloads are a combination of:**
  - Application function: What it does and how it does it
  - Data structure: Data residency, topology, access model
  - Usage pattern: Utilization profile over time, mix of use cases
  - Service level: Non-functional requirements
  - Integration: Interaction between application & data components

- **The workload requirements will create varying demands when determining server alternatives**



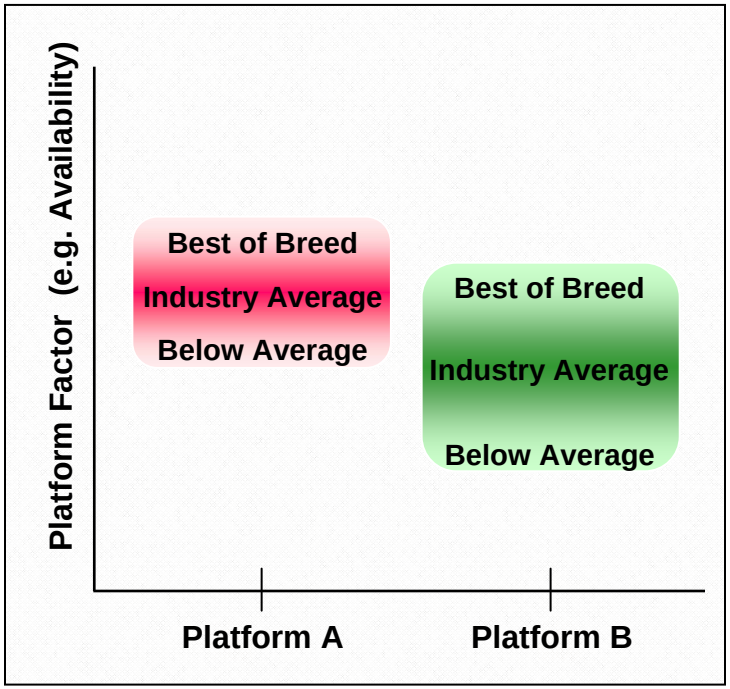
Selecting a Platform



There are many factors that influence platform selection making it difficult to develop a simple platform selection matrix

Local Factors are Important

- **Platform and workload type**
- **Local factors (constraints)**
  - Skills
  - Technology adoption levels
  - Platform management practices
  - Number of servers
  - Organization considerations
- **Service Level Agreements**
  - Non-functional requirements



Functional and Non-Functional Requirements

Select or design applications based on functional requirements driven by business process, and non-functional requirements

Functional
"What it does"

- Correct business results
- Inputs
- Outputs
- Behaviors
- External interfaces
- Screen layouts

Non-Functional
"How well it does it"

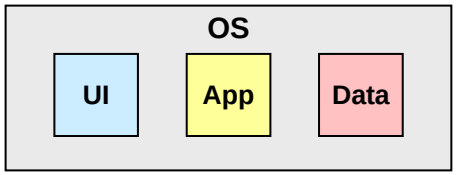
- Availability requirements
- Transactions per minute
- Security requirements
- Ease of provisioning and support
- Disaster recovery requirements
- Future growth

Select platforms based upon non-functional requirements driven by business value

Platform Strengths

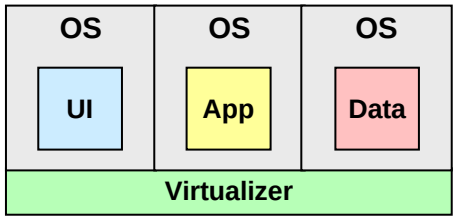
- **x86**
  - Granularity
  - User interface
  - Commodity servers
- **POWER**
  - Compute intensive
  - Parallel processing
  - High performance
- **System z**
  - Mixed workloads
  - High I/O
  - Scalability
  - Security

Common Deployment Models



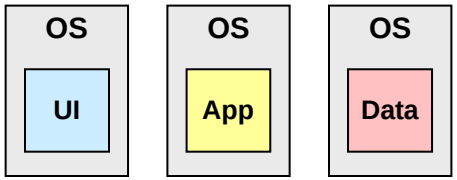
Shared

- Components are all together
- Very granular resource sharing
- OS workload management
- Strongly integrated and stacked



Virtualized

- Components split across virtual images
- Coarser grained resource sharing
- Virtualizer workload management
- Stacked and integrated over network

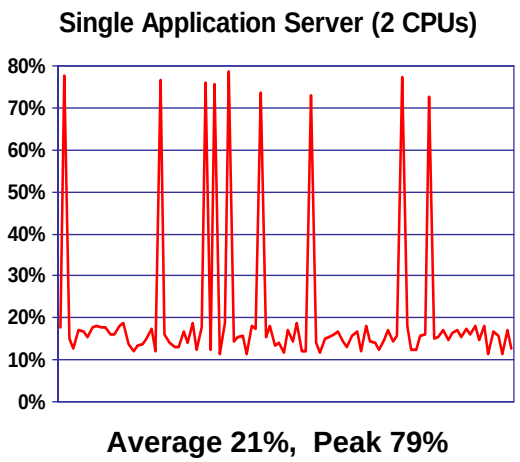


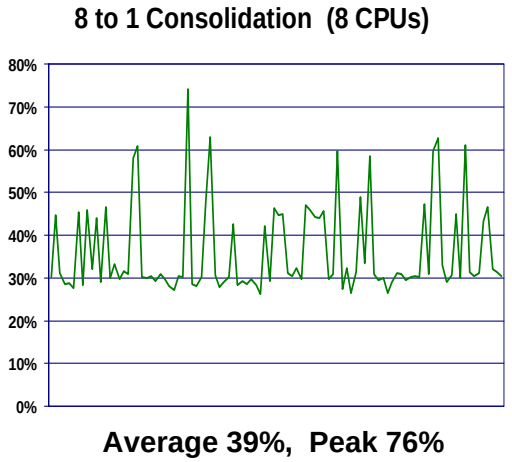
Dedicated

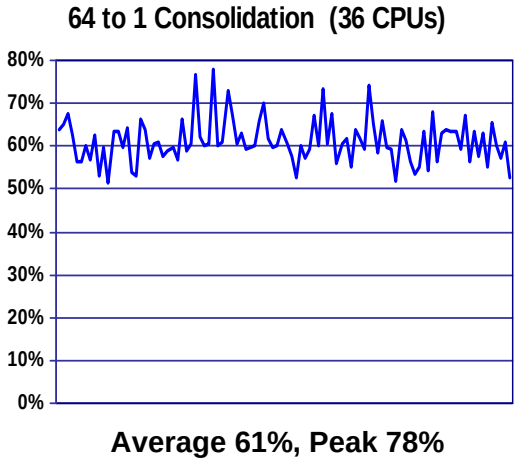
- Components split across servers
- No resource sharing between servers
- Limited workload management
- Integrated over physical networks

Consolidating Workloads Optimizes Efficiency

- **Single workload model**
  - Average: 21%; Peak: 79%
  - Random arrival rate
- **As copies are added**
  - Average approaches peak
  - Total CPU grows at slower rate







Workload Attributes and Market Segmentation

Transaction Processing and Database

- High Transaction Rates
- High Quality of Service
- Peak Workloads
- Resiliency and Security

Analytics and High Performance

- Compute or I/O intensive
- High memory bandwidth
- Floating point
- Scale out capable

Business Applications

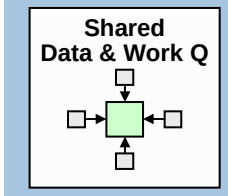
WebSphere

- Scale
- High Quality of Service
- Large memory footprint
- Responsive infrastructure

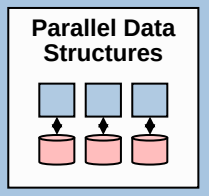
Web, Collaboration and Infrastructure

- Highly threaded
- Throughput-oriented
- Scale out capable
- Lower Quality of Service

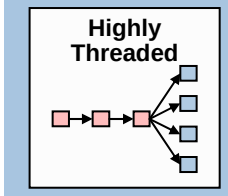
Workload Architectures – More Technical View



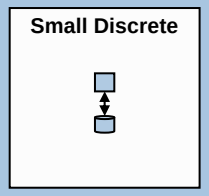
- Shared data and work queues



- Parallel data structures

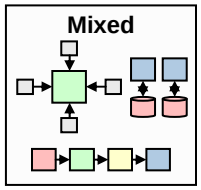


- Highly threaded

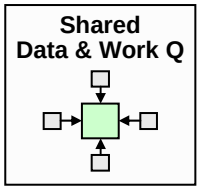
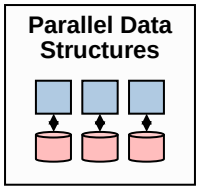
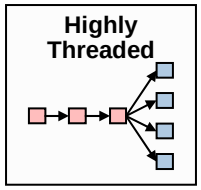
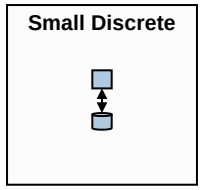
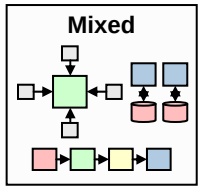


- Small discrete applications

Mixed

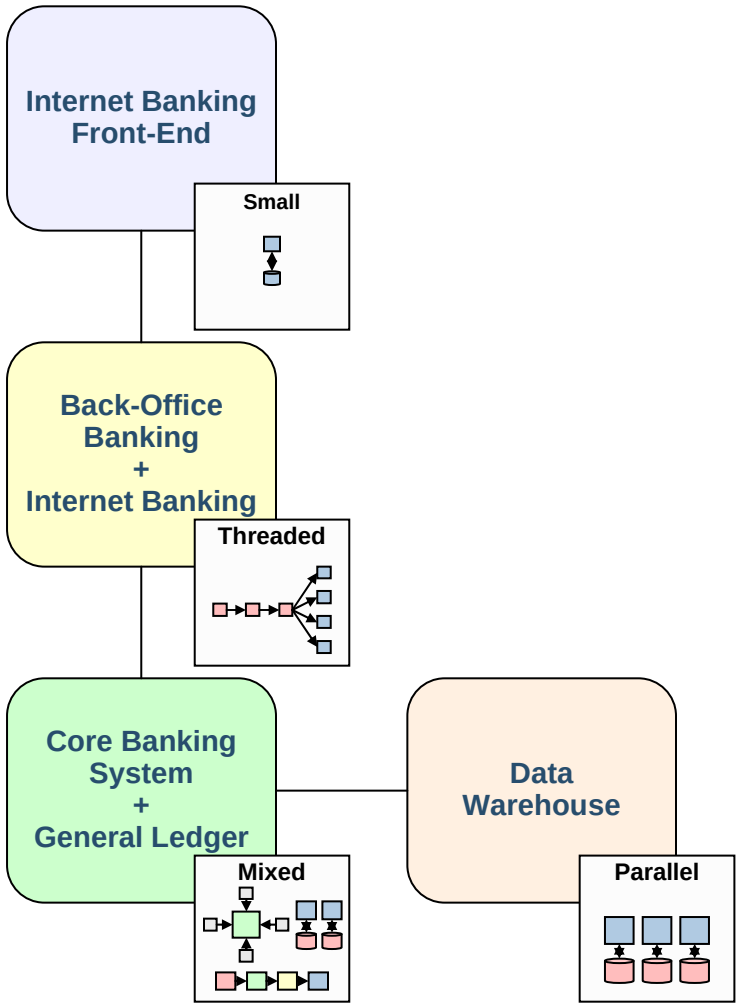


Workload Characteristics and Platform Requirements

	Shared Data & Work Q	Parallel Data Structures	Highly Threaded	Small Discrete	Mixed
					
Examples	▪ OLTP databases ▪ N-Tier transaction processing	▪ Structured BI ▪ XML parsing ▪ HPC applications	▪ Web app servers ▪ SAP app servers	▪ HTTP, FTP, DNS ▪ File and print ▪ Small end user apps	▪ z/OS and IBM i ▪ Hypervisors with virtual guests, WPAR
Characteristics	▪ Thread interaction raises contention & coherence delays ▪ Coherency traffic increases memory & cache bus utilization ▪ High context switch rates	▪ Low thread interaction ▪ High memory bandwidth ▪ Low context switch rates	▪ Lots of software threads ▪ Modest thread interaction	▪ Does not pressure any resource ▪ Requires minimal memory footprint ▪ Inefficient on dedicated resources ▪ No shared data	▪ Different SLAs ▪ Varying sizes and number of threads ▪ May be N-Tier or independent ▪ Variable context switch rates
Platform Considerations	▪ Scale on robust SMP ▪ Cluster technology dependent ▪ Large shared caches and wide busses ▪ Fewer, bigger threads	▪ Scale well on clusters ▪ Large private caches ▪ High thread count ▪ High memory and I/O bandwidth ▪ Often on dedicated machines	▪ Scale on large SMP ▪ Can scale on clusters ▪ High thread count ▪ Large number of memory busses ▪ Large private caches	▪ Scale on almost any hardware ▪ Ripe for virtualization ▪ Can exist on low cost hardware	▪ Scale on robust SMP ▪ High internal bandwidth ▪ Thread speed and number is workload dependent ▪ Large, close caches ▪ High memory bandwidth

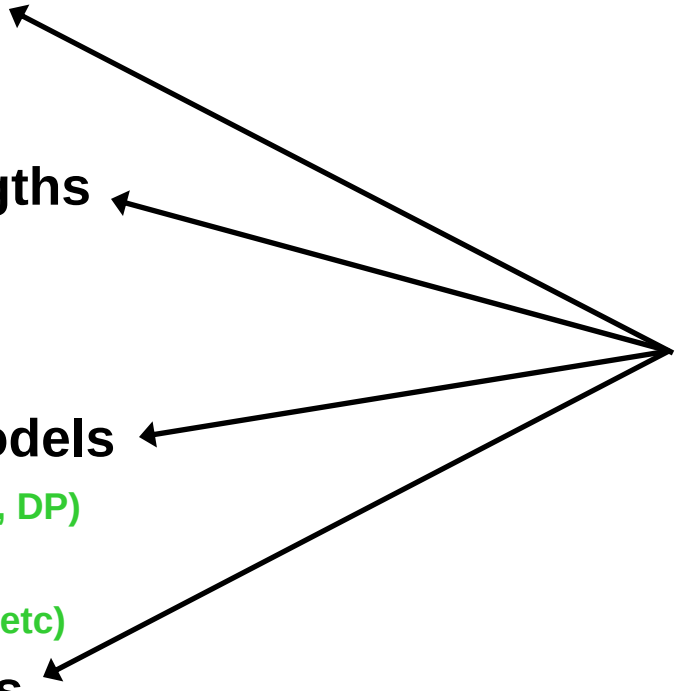
Multiple Platforms May be Appropriate

- **A workload**
  - May have multiple types
  - Can exhibit multiple types based on usage patterns
- **A mix of optimized platforms may be more cost effective**
- **Other local factors and non-functional requirements apply**



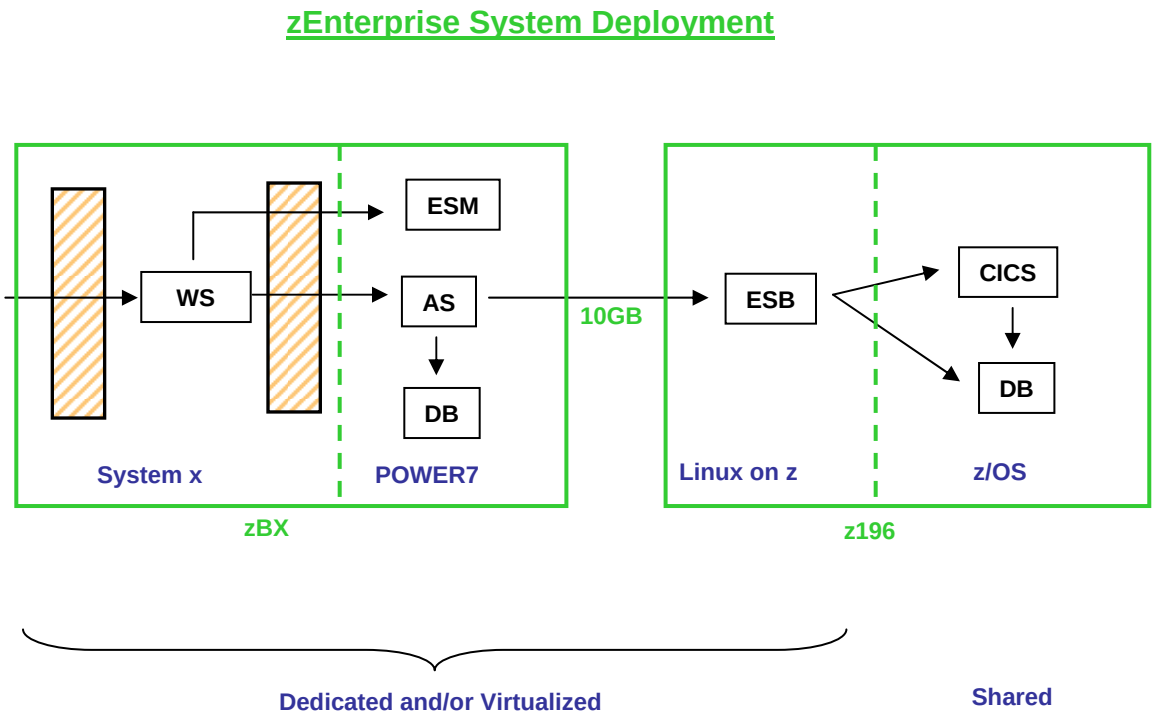
Platform Selection Considerations

- **Local Factors**
  - Technical
  - Non-technical
- **Platform Strengths**
  - x86 **(zBX)**
  - Power **(zBX)**
  - System z **(z196)**
- **Deployment Models**
  - Dedicated **(ISAO, DP)**
  - Shared **(zOS)**
  - Virtualized **(zVM, etc)**
- **Workload Types**
  - Transaction processing and DBs
  - Business applications
  - Web and Collaboration
  - Analytics and High Performance



zEnterprise addresses each of these!

Sample Application



- Performance
- Scale
- Availability
- Manageability
- Security