# A Route to Standardized Operating Environments

*Pedro Gómez*

*April 2007*

*Sun Microsystems, Inc.*
*Sun Venezuela*

# Table of Contents

## Index of Tables

# Index of Figures

# Introduction

Regardless of the complexity and variety existing in biological beings, living organisms are built from basic building blocks or fundamental units of life: cells. These small units execute functions, such as transportation and conversion, using a set of standardized structures (for example, the nucleus, cytoplasm, and organelles) that work together towards the proper functioning of the biological being to which they belong.

Similarly, an enterprise IT architecture uses building blocks or operating environments to create, process, and transform information that supports the operations of the business organization. In this context, standardized operating environments (SOEs) represent a key concept in the simplification and normalization stages of data center optimization.

Standardized operating environments act like the cells of a biological being by forming the building blocks of the IT architecture, by providing processing power, storage, software, and services. These operating environments are defined according to their place in the architecture's logical tiers, and they are shaped according to the systemic qualities (availability, performance, reliability, security, and so on) defined by the organization.

The purpose of this article is to provide a better understanding of the following topics:

- A definition of standardized operating environments (SOEs)
- The place of SOEs within the enterprise IT architecture
- Templates for SOEs
- A practical guide for developing and transitioning to SOEs
- Provisioning alternatives
- Governance linkages

# What Are Standardized Operating Environments?

Operating environments are computing and operational units that belong to the overall architecture of a data center. A standardized operating environment is a unit that is plugged into a tier of the IT architecture and has the responsibility of of moving transactions, executing instructions, and providing a service.

A unit is self-contained and has processing power and memory, an operating system, middleware, and business applications or services configured through the following four categories (the "Ss"):

- Systems, which consist of hardware resources (the servers, I/O, memory, computing power, and so on)
- Software, which is divided into three areas:
    - Operating systems (what actually runs and controls the hardware resources)
    - Infrastructure software (management and monitoring, messaging, administrative, and so on)
    - Business applications and executing logic
- Storage, which is related to the internal storage of the environment and the connectivity to external storage and a storage area network (SAN)
- Services, such as the support services and skills required to administer the corresponding environment. Operating environments need to be installed, supported, and maintained, and the service level varies.

The concept of standardized operating environments derives from several architecture methodologies and frameworks. For instance, the Open Group Architecture Framework (TOGAF, http://www.opengroup.org/togaf/), defines architecture building blocks as elements that are used in the definition of an enterprise architecture, such as hardware, operating systems, and applications. These blocks are compiled by the organization to provide reusable components and economies of scale.

In the SunTone$^{SM}$ Architecture Methodology (http://www.sun.com/2002-0212/feature/side1.html), the categories (the four Ss) are used to establish the operational environment of applications. The standardized operating environment theory and construction is mainly an application and extension of the three-dimensional model (3DM) of the SunTone Architecture Methodology with the following additions:

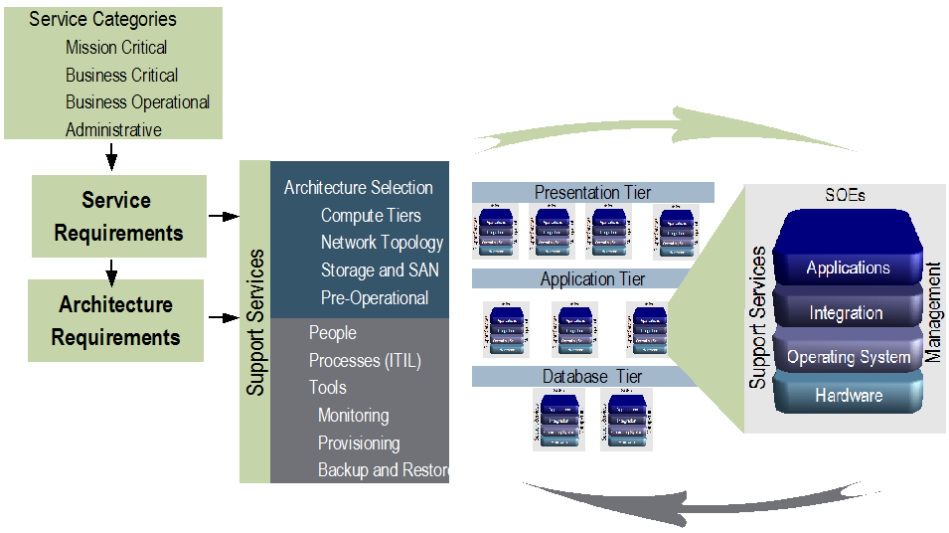
- The scope is the architecture of the data center and covers the functional, preoperational, and management tiers.
- The stack is extended to cover the services and support aspect of the environments.
- Governance is accounted for during the definition of the standardized operating environment.

## SOEs Within the Enterprise IT Architecture

Although each standardized operating environment could potentially work as a standalone unit, the standardized operating environment definition and implementation depends on the IT architecture design and implementation, as shown in Figure 1.

SOEs follow the architectural principles defined in the IT architecture. In turn, the IT architecture exists to support business services and their requirements. The relationship among the business strategy, the business services, the IT architecture, and the standardized operating environments is achieved through an understanding of the requirements, and the requirements translate into systemic qualities (performance, availability, scalability, reliability, security, manageability, and so on).

*Figure 1: Standardized Operating Environments Live Inside the IT Architecture Tiers*

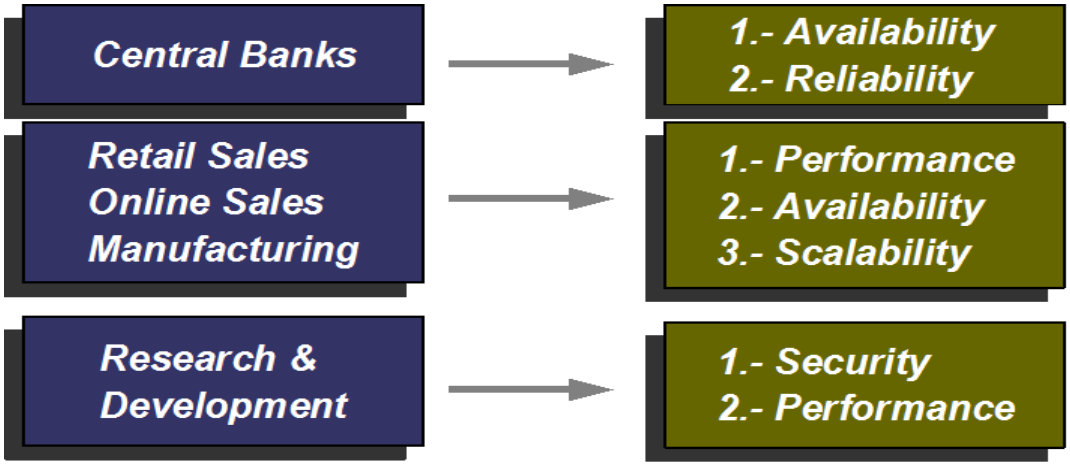



When an IT organization discusses the characteristics of the business services that the infrastructure must support, it must have a thorough understanding of the main service requirements in terms of the systemic qualities. The nature of the required business services dictates the characteristics of the IT design.

In organizations such as central banks, there is a significant need for **availability**. This need prevails over the need for performance and other business drivers, because central banks have a relatively low number of transactions per day with a high monetary value per transaction (which is easy to explain given the fact that the owner of the checking account is a country). The availability might not necessarily require a service with 24x7 agreement, but the service must be available when the banks need to transact with other international banks at specific times of the day or week. The main need is to ensure that the IT service is available when the banks need it.

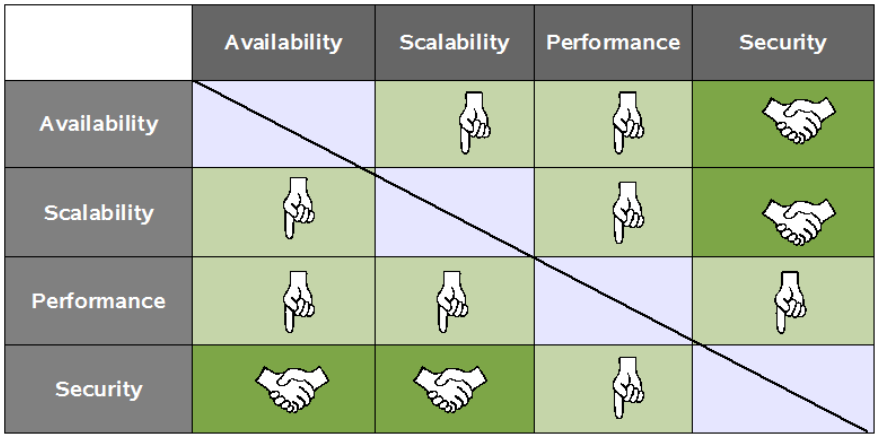
As shown in Figure 2, in other situations, such as retail sales and online auctions, **performance** is the most important differentiation mechanism in order to compete, and availability is the second most important requirement. In research and development firms, for which the business is the creation of intellectual capital, **security** is considered before any other systemic quality.

*Figure 2: Different Businesses Have Different Architecture Requirements*




The presence of different systemic qualities or drivers in the service requirements creates an interesting situation for IT architects, because some qualities have positive or negative impacts in others. Availability, for instance, can be enhanced at the expense of performance. A solution that requires redundancy of data and services usually requires additional layers of software (for example, cluster applications and middleware), hardware (for example, heartbeat networks and redundant components), and storage operations (for example, RAID 1). These layers could make the overall performance of the system slower. Similar situations could occur when security requirements introduce additional check points in the overall architecture. Figure 3 shows the relationship among the different systemic qualities.

*Figure 3: Relationship Among Systemic Qualities*



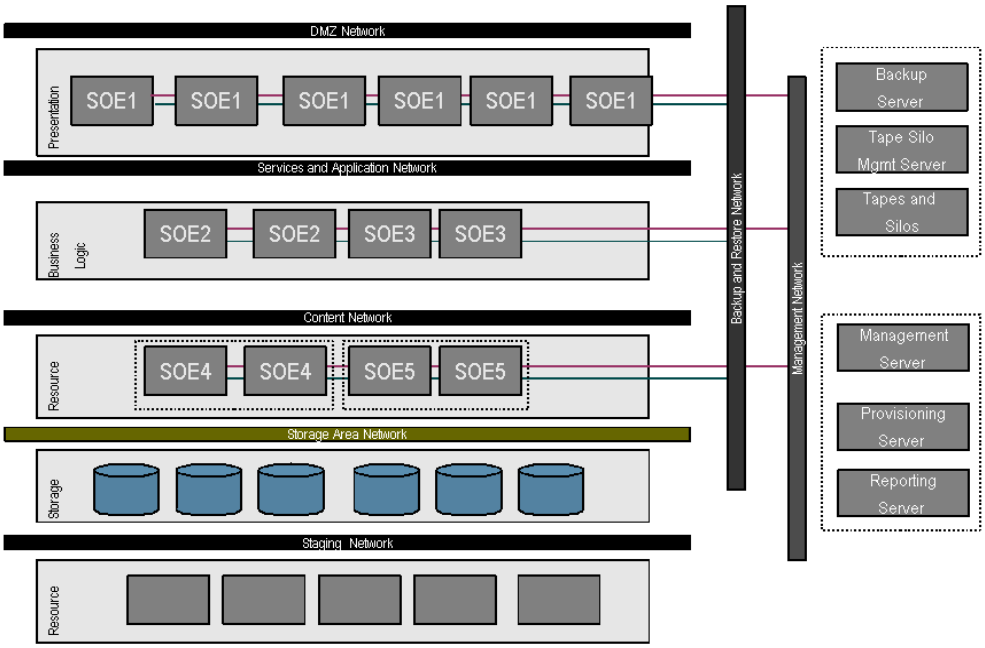

The definition of an enterprise IT architecture that accounts for these qualities and their trade-offs is outside the scope of this article. The Service Definition Workshop (http://www.sun.com/service/sdw/) from Sun Microsystems is an excellent resource for defining an enterprise IT architecture, bridging the

gap between business and technical requirements, and establishing architecture blueprints for IT. The process could result in the definition of an enterprise IT architecture that includes the following subarchitectures:

- Functional architectures, such as production environments
- Prefunctional architectures, such as mirrors and staging areas that are used to validate the quality of the applications and changes before they are released to the production environment, for example:
    - Development areas
    - Sandbox area
    - Quality assurance area
- Integration architectures, such as design hubs for establishing new atomized services
- Management architectures, such as:
    - Backup and restore architectures
    - Administration and monitoring architectures
    - Provisioning architectures

Each of these subarchitectures requires an operating environment to function, execute, and communicate with the remaining tiers. Figure 4 shows an example of an N-Tier production environment with standardized operating environments in each tier. Note that in some tiers, there is more than one standard stack.

*Figure 4: IT Architecture Tiers With Standardized Operating Environments*



## Templates for Standardized Operating Environments

Each of the four Ss (systems, software, storage, and services) of a standardized operating environment has several detailed components of information that are required to create a computing and operational unit.

### Systems

In order to create a unit, the systems category is used to group the information related to the server environment. The components included in this category are used to configure a server with the corresponding processing power, memory, and I/O connectivity.

- Basic information. The brand and model of the server are the mandatory information. Optionally, you could gather information on the department that will own the environment, the services this server will be supporting, and the specific configuration data, such as the host name and the host ID. These last two pieces of information could be used if you want to map an actual environment to a standardized operating environment.
- Processing power. The CPU architecture, the model, the speed, the number of cores, the number of threads, and the number of CPUs configured for this environment are the relevant pieces of information required here.
- Memory configuration. The specific memory density, the standard, and the total memory configured are registered here. Detailed information on the density can be used to draw conclusions on scalability and performance.
- Capacity. This information is usually directly related to the model and brand of the server; however, some solutions offer flexible and configurable solutions with dynamic I/O boards, slots, and a maximum number of configurable CPUs. Note that the capacity you are interested in is related to the potential growth that the server can achieve without requiring significant engineering changes to the design.
- Environmental. This information is used in the evaluation of power constraints and rack space. You need to know about rack units, power type, and the number of power supplies.
- Network connectivity. For this parameter, you interested in the configuration of cards and ports for the different networks of the server. Speed, port counts, bus type, and interface type are relevant.

### Software

The software elements include all components used to run, monitor, back up, and restore the server, as well as the applications and services for which the environment has been designed:

- Operating environment. The operating system, volume management, and file system layers are accounted for in this category. The information required includes the level of patch that needs to be applied to the environment in order to be able to provision, manage, and control the environment.
- Development layer. This category is necessary to understand the runtime environment for services that will run in the server.

- Management layer. The monitoring agents that will help govern the standardized operating environment are included here. Also included in this category are provisioning, administration, and data backup agents.
- Integration layer. This layer could be optional if integration hub middleware is part of the architecture.
- Application layer. The business application is accounted for here. The name, versions, and file system layout are the basic elements to establish, but additional elements could be included depending on the nature of the application.

### Storage

The standardized operating environment is a computing unit, and the storage category exists because of the internal disks and the connectivity to external storage.

- Internal disks. This category consists of the number of disks, the capacity, the speed, the form factor, the interface type, and the RAID level (implemented either through software or hardware), and total space configured.
- Direct-attached storage (DAS). It is important to understand that this information is necessary to configure the server, not the external storage. You are interested in the interfaces, connectivity, ports speed, models, and the total addressable space for the environment.
- SAN. Similar to DAS information, the information in this category is related to the connectivity of the server to the external storage, including host bus adapters (HBAs), switch brands and models, and storage brands and models.

### Services

Services consist of the following:

- Support services. What level of support will be contracted for the environment? Because a standardized operating environment consolidates hardware, the OS, volume manager software, and applications, you need to account for the service support level in each of these elements.
- Skills required. When a standardized operating environment is deployed in a production environment, system administrators must have knowledge about the environment to be able to manage it. Therefore, you must establish the skill set required to manage the standardized operating environment. This information is useful to determine the organizational components (training, profiles, and so on) that will govern the architecture.

## Creating Instances of SOEs

In the SunTone Architecture Methodology, IT architects establish the overall information flow model with the definition of tiers and characteristics to support the operation. Architectural principles establish the behavior of the implementation and its growth. Table 1 depicts examples of decisions made by the architects and the expected actions that will govern the model.

*Table 1: Systemic Qualities per Tier*

| | Scalability | Availability | Service Isolation | Load Management |
|---|---|---|---|---|
| **Presentation Tier** | Add Protocol Servers with Load Balancer Distribution | Load Balancer failover redirection | Separate servers for different protocols or user cases | Load balancer |
| **Application Tier** | Vertical scalability to maximum limit of app servers | Typically application server agent selection of alternative application server instances | Separate application server instances for different user / application classes | Typically application server agent load balancing to alternative application server instances |
| **Integration Tier** | Vertical scalability to maximum limit of application servers (Horizontal replication of services, for example master slave) | May use load balancer type service-level failover or messaging agent alternative resources | Dedicating services to servers or resource sharing such as srm and containers | Often intrinsic to scaling model of service |
| **Resource Tier** | Maximum vertical scaling with partitioning or replication | Clustered servers with custom failover | Dedicated resources or possible use of containers/srm | Some limited use service agent-based balancing |

As soon as there is a clear understanding of the expected behavior for the architecture, the standardized operating environment stacks are built in an iterative way by analyzing each tier of the selected architectures and establishing the candidate stacks that apply in each tier.

Using the systemic qualities as anchors for the definitions, you can use Table 2 as a guideline for understanding what to consider in each component of the standardized operating environment in the tier being evaluated. This matrix would be also used to support the decision criteria used to instantiate the environments in the future, when new technology is available.

Because standardized operating environments are designed to fit within an architecture tier, it is extremely important to have an outline of the expected behavior with regards to the systemic qualities of each tier before focusing on creating the standardized operating environments. The decisions about what elements will be included in the environment depends on the approach established for the tier.

For example, the availability strategy for the presentation tier is usually based on load balancing of stateless servers, for which the redundancy of components is not necessarily a priority. The environments in this tier tend to be designed with a greater emphasis on performance and throughput but with loose restrictions on the availability components of each operating environment. That doesn't mean that the tier itself lacks the ability to be available, but rather that the applications and transactions passing through this tier are made available by a combination of external factors not considered in the standardized operating environment (for example, load balancing).

In the resource tier, on the other hand, availability is designed within the operating environment as a main characteristic, with careful consideration of the hardware layer up to the business applications layer, based on the use of cluster software and redundant components.

*Table 2: Guideline Matrix for Systemic Qualities per Standard Operating Environment*

| SOEs | Performance | Availability | Reliability | Recoverability | Scalability |
|---|---|---|---|---|---|
| **SYSTEMS** | | | | | |
| Basic Information Server Model Sever Brand Hostname Physical Location Department Owner IT Services Delivered Business Services Delivered | | HA failover ready Redundant internal interconnect Redundant system clocks | | Automatic System Recovery Automatic system controller failover | Vertical/Horizontal/Diagonal Max Number CPUs Max Memory Config |
| Processing Power CPU Architecture CPU Model Clock Speed Cores per CPU Number of CPUs | Usually a higher clock speed yields higher performance, but a higher thread count/cores per cpu can have better results in multithreaded applications. Cache Size and the presence of L2 and L3 cache enhances performace. HPC computing requires a CPU with strong floating point performance | Redundant Number of CPUS / Cores Failure Isolation | | | |
| Memory Configuration Memory Size Memory Density Memory Standard | Memory Standards have a heavy impact on performance. DDR2 has double the bandwith and transfer rate of DDR. Large amount of nemory contributes to increased performance | ECC/Parity correction on critical components I/O boards/trays . Spread Memory Size in small dimm size | | | |
| Capacity CPU Mem Boards I/O Slots I/O Bandwith I/OBoards SSL Adapter Max Ops/sec SSL Adapter Interface Type | | Redundant CPU/Memory boards Redundant I/O Boards Cluster Software Clustered | | | Capacity is directly related to scalability, Vertical Scalability refers to adding more components inside the same configuration. Watch for number of I/O slots maximum number of CPUs, and addressable number of disks. |
| Environmentals Power Supplies Rack Units Input Voltage Input Current Power Consumption BTU's | N/A | Redundant (N+1) power supplies, Redundant FANS | | | |
| Networking Conectivity Network Tier Bus type Interface Type Network ports connectors Number of ports Number of Network Cards Port Speed | Port speed is the main concern for performance, but also the number of ports and the capability to use load balancing and trunking technologies | Redundant network interface cards, redundant fans | | Automatic failover for network interface cards | |

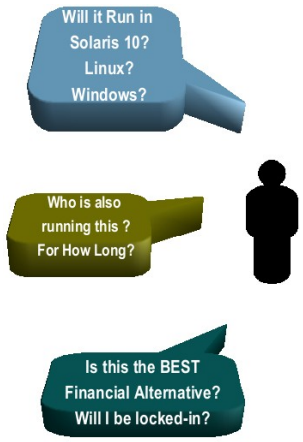
## Constraints and External Considerations Influencing SOE Selection

The systemic qualities correspond to the internal qualities of the architecture. As shown in Figure 5, there are some external elements that must be also considered in the analysis and design of standardized stacks:

- Certification of third-party independent software vendors (ISVs). Third-party ISVs dedicate resources to ensure that their applications run without problems in specific environments. If a problem occurs, the engineers in the ISV support center have specific knowledge of what issues might arise in the platforms they have tested. However, if you choose to standardize an operating environment for which the ISV has not issued a certification "stamp," there are risks associated with solving problems. These risks could jeopardize the entire standardization process.
- Platform certification. Hardware, storage, applications, drivers, and HBAs are built independently of each other, even when they are all provided from a single vendor. The certification of the entire stack must be carefully screened before you make a commitment to move to a standardized operating environment.
- Maturity. Even when all the certifications for hardware and software have been carefully considered, it is important to consider other aspects, such as install-base configurations and the experiences of customers using the platform, which could enhance the robustness of the stacks over the time. These items also could result in lower incidence of bugs and critical patches.
- Total cost of ownership (TCO). TCO is a critical aspect to consider. It includes the up-front purchasing price of the stacks, the cost for supporting and maintaining the stacks, and the personnel costs for acquiring the skills that are required to manage and grow the environment. Brian Down provides good insights on this topic in his Sun BluePrints™ article *Protecting Investments Through Technology Advancements* (http://www.sun.com/blueprints/1005/819-3931.html).

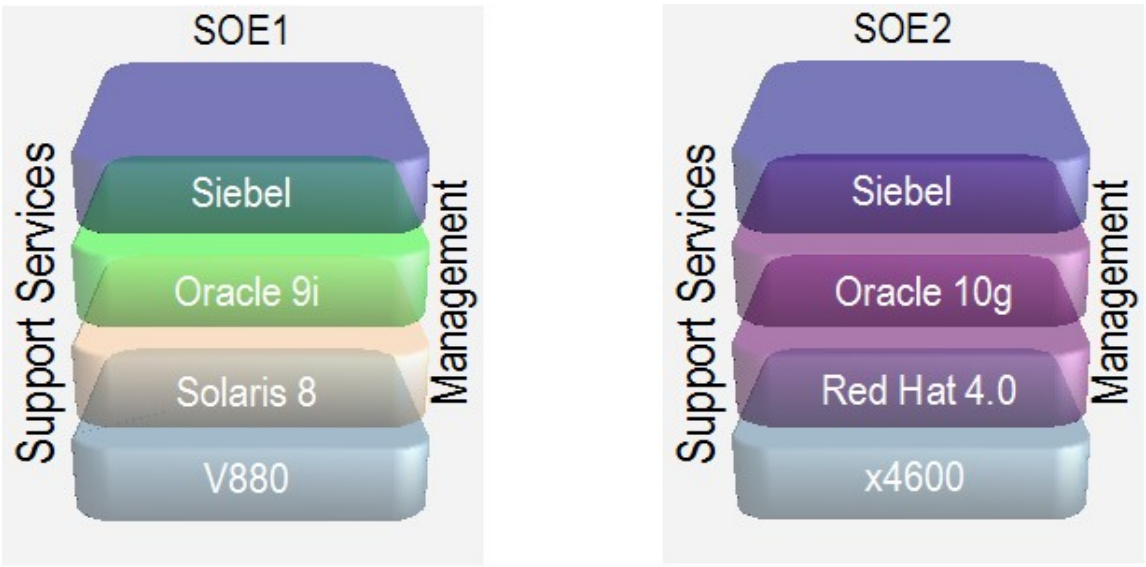
Figure 5 shows some of the considerations involved in creating SOEs, such as whether the services and applications in the SOE will be certified in the Solaris Operating System or another OS.

*Figure 5: Additional Questions to Consider When Creating SOEs*



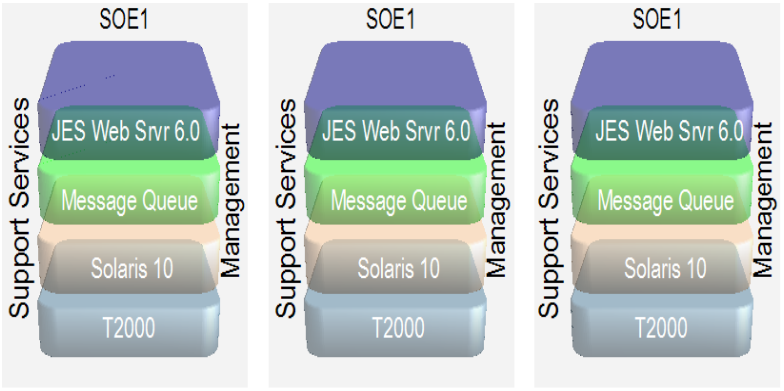

IT architects must evaluate and balance these external factors against the required systemic qualities of the infrastructure in order to make the appropriate adjustments to the standardized operating environment components and to the number of different standardized operating environments that will tentatively constitute the build specifications. Figure 6 shows an example of two different SOEs.

*Figure 6: Different Standardized Operating Environments Require Different Stack Elements*




It is important to distinguish between the number of standardized operating environments defined and the number of *instances* of SOEs per tier. The number of instances of a standardized operating environments is the number of times the same SOE is provisioned according to the workload of the services for which the environment is designed, as shown in Figure 7. The number of required instances is calculated through sizing exercises, typically using ISV guidelines (in the case of third-party software) and through the capacity planning process of the organization.

*Figure 7: Three Instances of the Same Standardized Operating Environment*




*Key:*
*JES Web Srvr= Sun Java™ System Web Server*
*Solaris 10 = Solaris 10 OS*
*T2000 = Sun Fire™ T2000 Server*

As soon as the stacks for each of the corresponding tiers are created, the design decisions are put into a standardized operating environment matrix, as shown in Table 3.

*Table 3: Standardized Operating Environment Matrix (Only Systems Shown)*

| Components | Functional Tiers | | | |
|---|---|---|---|---|
| PART NUMBERS | Sun-S1-RES | Sun-S2-BUS | Sun-S3-INT | Sun-S4-PRE |
| | Resource | Business | Integration | Presentation |
| SYSTEMS | | | | |
| **Basic Information** | | | | |
| Server Model | E25K | E25K | | T2000 |
| Server Brand | Sun Fire | Sun Fire | | Sun Fire |
| Hostname | | | | |
| Physical Location | CCS-1 | CCS-1 | | CCS-1 |
| Department Owner | IT | IT | | IT |
| IT Services Delivered | Database | Application Server | | Web Server |
| Business Services Delivered | Payroll | Payroll | | Payroll |
| **Processing Power** | | | | |
| CPU Architecture | RISC | RISC | | RISC |
| CPU Model | US IV + | US IV + | | US T1 |
| Clock Speed | 1.5 Ghz | 1.5 Ghz | | 1.4 Ghz |
| Cores per CPU | 2 | 2 | | 8 |
| Number of CPUs | 12 | 8 | | 1 |
| **Memory Configuration** | | | | |
| Memory Size | 48 | 32 | | 32 |
| Memory Density | 2GB | 2GB | | 2GB |
| Memory Standard | DDR | DDR | | DDR |
| **Capacity** | | | | |
| CPU Mem Boards | 3 | 2 | | N/A |
| I/O Slots | 8 | 8 | | N/A |
| I/O Bandwith | 9.6 Gbps | 9.6 Gbps | | 4.8 |
| I/O Boards | 2 | 2 | | N/A |
| SSL Adapter Max Ops/sec | | | | On Chip |
| SSL Adapter Interface Type | | | | On Chip |
| Max Number of Internal Disks | | | | 4 |
| **Environmentals** | | | | |
| Power Supplies | | | | 2 |
| Rack Units | | | | 2 |
| Input Voltage | 200VAC,47-63Hz | 200VAC,47-63Hz | | 240VAC |
| Input Current | 45.6A @200VAC | 45.6A @200VAC | | 2A @ 200VAC |
| Power Consumption | 9120W(Max) | 9120W(Max) | | 400W |
| BTU's | 31,113 BTU/hr max | 31,113 BTU/hr max | | 1365 BTU/Hr max |
| **Networking Connectivity** | | | | |
| Network Tier | Resource | Apps | | Presentation |
| Bus type | PCI-X | PCI-X | | PCI-E |
| Interface Type | Copper | Copper | | Copper |
| Network port connectors | RJ45 | RJ45 | | RJ45 |
| Number of ports | 12 | 8 | | 4 |
| Number of Network Cards | 3 | 2 | | On board |
| Port Speed | 10/100/1000 Mbps | 10/100/1000 Mbps | | 10/100/1000 Mbps |

## Impact of Systemic Qualities on SOE Components

During the design process and after all the elements of the stack have been defined, it is convenient to validate how the stacks comply with the design principles. The Component Impact Analysis Matrix shown in Table 4 is a tool that can be used as a tollgate to:

- Validate the compliance of the standardized operating environments with the systemic qualities
- Evaluate the impact that would be caused to the entire stack if a component failed
- Analyze the potential introduction of new technology or the substitution of components

*Table 4: Component Impact Analysis Matrix*

| SOEs / SYSTEMS | Performance | Reliability | Availability | Scalability | Security | Accessability | Recoverability | Manageability | Usability | Serviceability | Reusability | Flexib |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Basic Information** | | | | | | | | | | | | |
| Server Model | H | H | H | H | H | | | | | M | | |
| Server Brand | H | M | H | | | | | | | | | |
| Hostname | | | | | | | | | | | | |
| Physical Location | | | | | | | | | | | | |
| Department Owner | | | | | | | | | | | | |
| IT Services Delivered | | | | | | | | | | | | |
| Business Services Delivered | | | | | | | | | | | | |
| **Processing Power** | | | | | | | | | | | | |
| CPU Architecture | H | M | L | | | | | | | | | |
| CPU Model | H | M | L | H | | | | | | | | |
| Clock Speed | H | M | M | | | | | | | | | |
| Cores per CPU | H | M | M | H | | | | | | | | |
| Number of CPUs | H | | M | H | | | | | | | | |
| **Memory Configuration** | | | | | | | | | | | | |
| Memory Size | H | L | L | M | | | | | | | | |
| Memory Density | H | M | L | H | | | | | | | | |
| Memory Standard | H | M | H | H | | | | | | | | |
| **Capacity** | | | | | | | | | | | | |
| CPU Mem Boards | H | | H | H | | | | | | | | |
| I/O Slots | H | | H | H | | H | | | | | | |
| I/O Bandwith | M | | | H | | | | | | | | |
| I/O Boards | L | | H | | | | | | | | | |
| SSL Adapter Max Ops/sec | H | | | | H | | | | | | | |
| SSL Adapter Interface Type | H | | | | H | | | | | | | |
| **Environmentals** | | | | | | | | | | | | |
| Power Supplies | | | | | | | | | | | | |
| Rack Units | | | | | | | | | | | | |
| Input Voltage | | | | | | | | | | | | |
| Input Current | | | | | | | | | | | | |
| Power Consumption | | | | | | | | | | | | |
| BTU's | | | | | | | | | | | | |
| **Networking Conectivity** | | | | | | | | | | | | |
| Network Tier | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| Bus type | H | M | | H | | | | | | | | |
| Interface Type | H | | | | | | | | M | | M | M |
| Number of ports | M | | H | H | | | | | | | | |
| Number of Network Cards | L | | H | H | | | | | | | | |
| Port Speed | H | | | M | | | | | | | | |

To categorize the impact of components on each of the systemic qualities, you use three values: high impact (H), medium impact (M), and low impact (L). The rationale for placing each component in a particular category is explained in Table 5.

*Table 5: Systemic Qualities and Impact Categories*

| | | GUIDELINES TO DETERMINE CATEGORY OF IMPACT ON SYSTEMIC QUALITIES | | |
|---|---|---|---|---|
| **SYSTEMIC QUALITIES** | **DESCRIPTION** | **HIGH** | **MEDIUM** | **LOW** |
| **SERVICE LEVEL** | | | | |
| **Performance** | Relates both to the specific performance metrics (e.g., responsiveness, latency) and to the users' expectations about performance. | * HW. An increase in the number of this component has a linear relationship to performance of OE. Component has been designed to offload operations from other components for performance SW: Typical Benchmarks on this component are related to speed and throughput. Total or partial failure of this component will have a direct impact on the performance of the OE. | * HW. An increase in the number of this component would benefit the overall performance of the OE | * HW. An increase or reduction in the number of this component would have little or no impact on the overall performance of the OE |
| **Availability** | Essentially the percentage of time that the system is available for use. Total availability is also impacted by factors beyond the architecture such as latency within a user's ISP or the general Internet. | Failure of this component could bring down the OE. - Adding another instance of this component (Redundancy) could prevent the SOE from failing (even if it requires other components to work). The presence of this element allows the OS to continue running in the case of failure of another instance. | Failure of this component could impact the availability of the OE without bringing it down. | Failure of this component has little or no impact on the availability of the operating environment. |
| **STRATEGIC LEVEL** | | | | |
| **Reliability** | Related to the reliability of the underlying individual components in servers. System reliability describes the likelihood of any component failures. | Component has implemented verifiable mechanisms that contribute to a consistent output for the task it has been assigned. Failure of this component or part of it could degrade the system's overall performance | | |
| **Scalability** | Scalability relates to the ability to add capacity and thus add users over time. Scalability will usually require the addition of resources, but scalability should not require changes in the architecture, redesign or loss of service due to the time required to scale. | The component has been designed to handle operations considering throughput and multithreading. The component design has the potential to accommodate additional capacity to allow for increase in throughput without changes in the architecture of the OE. | The component has been designed to handle operations considering throughput and multithreading. The component design has the potential to accommodate additional capacity to allow for increase in throughput without changes in the architecture of the OE. | Component presence or absence has little impact on the ability of the SOE to accommodate additional transactions/users. |
| **SYSTEM LEVEL** | | | | |
| **Security** | Levels of authentication supported, the granularity of authorization controls, the mechanisms for provided auditability and the techniques utilized to ensure the integrity of resources, and the resistance to unauthorized access. | The component or its elements contribute directly to enhance the security of the operating environment, including physical elements and/or software components, isolation mechanisms, filtering capabilities, etc. | The component or its elements could contribute to the overall security of the environment if used in conjunction with other elements in the stack. | The component or its elements have no particular participation in the security of the environment. |
| **Accessibility** | Usability of applications across the full range of user capabilities, languages, and devices. | HW: The component design is to be accessible without the use of mechanical tools. The component contains visual indicators of functions and problems. SW: Several tools are provided to configure and maintain the software elements, including command-line interface and graphical user interface. | HW: The component design simplifies the maintenance activities of the environment. SW: At least one tool is provided to configure and maintain the software elements. | The component design does not consider accessibility features. |

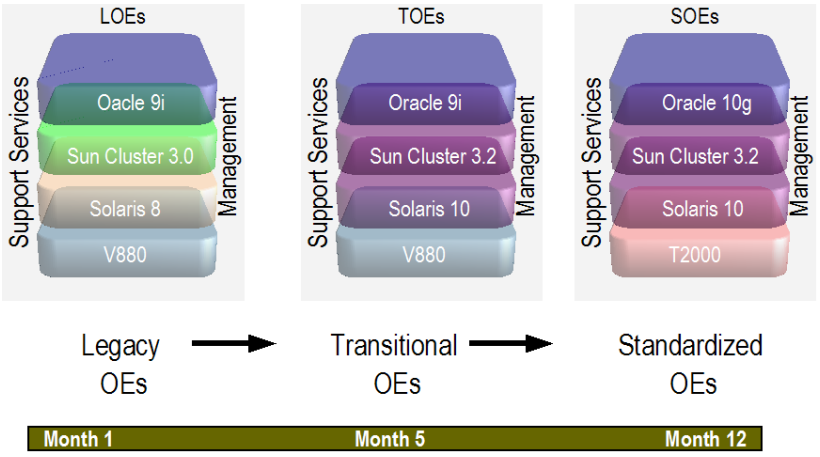
## Transitioning From Legacy Operating Environments to SOEs

If the environments you create are for a new data center, the logical next step after the design and validation of the standardized operating environments would be the implementation (provisioning) and governance. In an optimization process, however, you have existing environments that would transition to the standardized stacks. The process described so far can also be used to establish the route from a current, legacy operating environment (LOE) to a standardized operating environment.

As discussed earlier, an important element of the transition process is the certification of applications and services from both internal sources and from third-party vendors. It is fundamental that you work within the boundaries of the ISV and independent hardware vendor (IHV) ecosystem by honoring platform certifications. Because this limitation exists, you must make an initial assessment of the current ecosystem, which is also called an **operational baseline**, by using templates that are similar to the ones used to define the standardized operating environments in each tier. This process will help you understand the gap between the legacy operating environments and the standardized operating environments.

The Open Group Architecture Framework suggests a phased approach to close the gap from the "as-is" legacy operating environments, through the transitional operating environments (TOEs), to the "to-be" SOEs by using incremental adjustments to the platform until the final standardization goal is achieved, as shown in Figure 8.

*Figure 8: Transitions to Standardization*




*Key:*
*Sun Cluster = Sun™ Cluster software*
*Solaris 10 = Solaris 10 OS*
*V880 = Sun Fire™ V880 Server*
*T2000 = Sun Fire™ T2000 Server*

This transitional strategy is particularly relevant in those scenarios with a high dependency on third-party providers, such as ISVs and IHVs. For example, one issue might be a software vendor that has not yet certified its applications on the selected operating system, and another issue might be hardware adapters for which storage and server manufacturers have not agreed on specific software configurations. An incremental approach could also be an alternative to reduce the risk of migration. The idea is to keep most of the stack intact and make a few validated changes in each step with sufficient time to evaluate the impacts of the adjustments. As soon as the new environment is stabilized, additional adjustments are introduced until the SOE is obtained.

An operational baseline captures the following information:

- **Legacy Services and Applications View**, which consists of the current IT services and business services with their corresponding application architecture
- **Legacy Operating Environments View**, which consists of the current legacy operating environments and the IT architecture
- Optionally, the Legacy Processes and Organization View, which consists of the organizational data and IT processes to manage the data center operations

The Legacy Services and Applications View contains the following information:

- Business services
- IT services created to support the business services
- Applications
- Components of these applications by tiers

There is a hierarchical relationship among the business services, the IT services, and the applications used to support the services, as shown in Figure 9. This information is used to understand the current relationships among these elements and to focus the information that is captured about the current environments so it is really relevant to the business and IT operations.

*Figure 9: Services View: Business and IT Services, Applications, and Their Components*

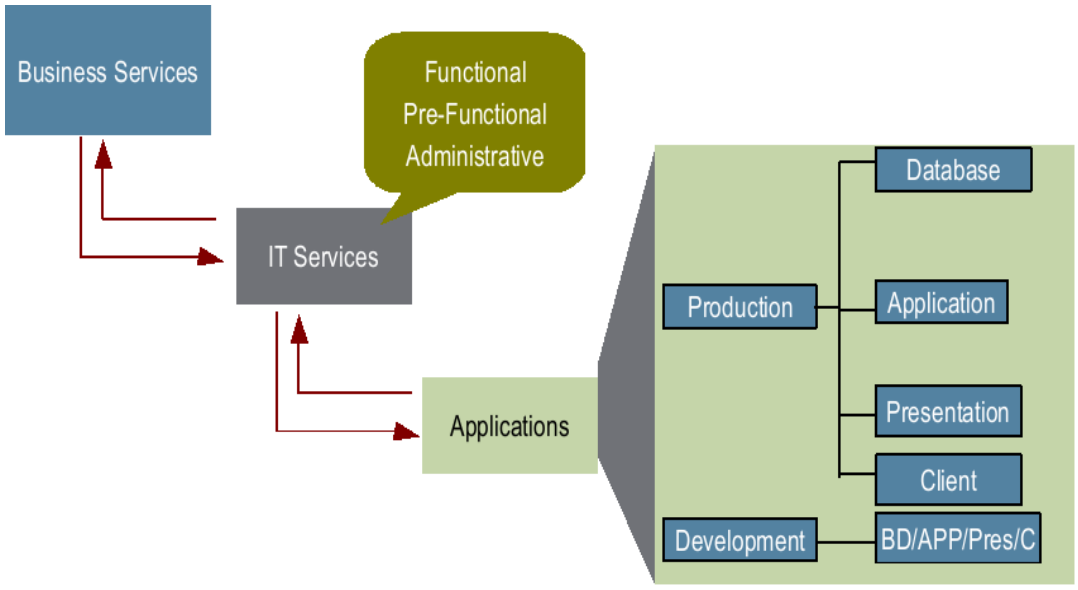



Table 6 outlines business services, IT services, and applications per tier.

*Table 6: Business Services, IT Services, and Applications per Tier*

| | | | | Functional Architecture Tiers | | | | | | | Development Architecture Tiers | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Business Service | IT Service | Applications | Category | DB | App | Present | Client | High Availability | Monitoring | Provisioning | DB | App | Present | Client | High Availability | Monitoring | Provisioning |
| Communications | | | MC | | | | | | | | | | | | | | |
| | **Email** | **MS Exchange** | MC | Exchange Server | | | | | | | | | | | | | |
| | **Intranet** | **Intranet Services** | BO | MySql | Jboss | Apache | Browsers (MS IE, Mozilla) | | | | | | | | | | |
| | **Internet** | | | MySql | Jboss | Apache | | | | | | | | | | | |
| Inventory Control | **Inventory Control** | | MC | | | | | | | | | | | | | | |
| | | **SAP R3** | MC | Oracle 8i | Web-Logic | Apache | | In DB Tier Sun[tm] Cluster 3.1 | | | | | | | In DB Tier Sun Cluster 3.1 | | |
| | | | | | | | | | | | | | | | | | |

For the Legacy Operating Environments View, you map all the current applications to the corresponding infrastructure elements or environments according to the legacy operating environment templates, as shown in Table 7. You also include information that can help you evaluate which components of the stack can be standardized and which should be monitored closely to ensure they comply with the certifications of the third-party providers.

To achieve this objective, you establish three labels of information as shown in Table 7:

- Relevancy: This color relates to the relevancy of the component for certification. If the application is from a third party, and this component is part of the certification process, you label it "Relevant." The Relevant tag helps you to understand what components are critical for the certification of the platform. This information is especially important when third-party software is included in the solutions.
- Supported options are shown by this color: Among the alternatives available in the market, list all the options that are supported to run for each component.
- Installed: Using this label, you list the actual installed items that are currently functioning within the existing architecture.

*Table 7: Operational Baseline for Legacy Operating Environments (for Systems Only)*

| SOEs | RESOURCE | | |
|---|---|---|---|
| | SAP R/3 | Siebel | HR Intranet |
| **SYSTEMS** | | | |
| **Basic Information** | | | |
| Server Model | E25K | X4200 | X4200 |
| Server Brand | Sun Fire | Sun Fire | Sun Fire |
| Hostname | | | |
| Physical Location | | | |
| Department Owner | | | |
| IT Services Delivered | | | |
| Business Services Delivered | Payroll | CRM | NEWS |
| **Processing Power** | | | |
| CPU Architecture | Relevant | Relevant | N/A |
| | RISC | RISC | RISC |
| | X86-32 | X86-32 | X86-32 |
| | X86-64 | X86-64 | X86-64 |
| | EPIC | EPIC | EPIC |
| CPU Model | N/A | N/A | N/A |
| | US IV | AMD Opteron | AMD Opteron |
| Note: US stands for UltraSPARC® | US IV+ | US IV+ | US IV+ |
| | Itanium | Itanium | Itanium |
| | Power PC | Power PC | Power PC |
| Clock Speed | N/A | N/A | N/A |
| | 1.35 | 2.2 | 2.2 |
| Cores per CPU | 2 | 2 | 2 |
| Number of CPUs | 16 | 4 | 4 |
| **Memory Configuration** | N/A | N/A | N/A |
| Memory Size | 32 | 8 | 8 |
| | | | |
| Memory Density | | | |
| Memory Standard | DDR | DDR2 | DDR2 |

| | |
|---|---|
| Relevant | Relevant |
| Supported | Supported by application owner |
| Installed | Installed |

The following process is the suggested approach for analyzing the data:

1. Identify environments in each tier that have common elements, which can be targeted for simplification.
2. Analyze the installed elements of components that are relevant for certification in each tier. Because you gathered information on what is installed and what options are supported for the components, you can draw conclusions about potential transitional environments.
3. Create potential transitional operating environment candidates.
4. Evaluate transitional operating environment candidates against standardized operating environments.
5. Create a transitional road map that shows the different transitional operating environments with the corresponding action plans derived, which could include:
    a) Purchasing plans
    b) Installation and migration activities
    c) Maintenance time windows
    d) Rollback plans
    e) Architecture recertification plans
    f) Project management information
    g) Required training skills

## Provisioning Standardized Operating Environments

As soon as the design specification for a standardized operating environment is complete, there are several alternatives to you can use to install, configure, and maintain the standardized operating environment. The options range from installing factory-built systems to manually installing each of the components.

### Factory-Built Systems (Customer Ready Systems)

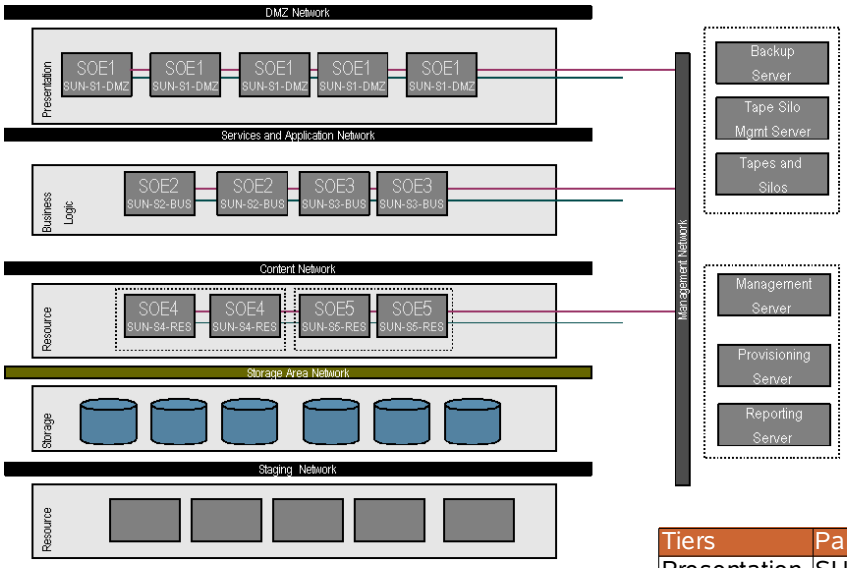
One of the principles governing the idea of a standardized operating environment is that a standardized operating environment must be a configurable computing and operational unit. What this means is that after the standardization process is performed, you should be able to transform the specifications into part numbers and send the part numbers somewhere, so the system can be built internally or externally. Today, technology providers can provide customer ready systems, which are factory-integrated solutions, and ship them to the data center where the final localization takes place before going into production.

Sun Microsystems implements factory-integrated services through its Customer Ready Systems (CRS) Program (http://www.sun.com/service/crs/index.html). Customers have the option of requesting preconfigured solutions, such as a complete stack including third-party software.

Figure 10 shows a multitier environment with several standardized operating environments defined by tier and a layout. Using the CRS program, you could set up specific part numbers for each one of these environments, and then you could issue a request for the design configurations and the number of standardized operating environments required for each tier. The CRS program would integrate all the systems and components and ship them to the final destination.

Additional configuration would be required to handle localization aspects, such as IP addresses, host names, time zones, and so on.

*Figure 10: Multitier Environment With Several Standardized Operating Environments*




| Tiers | Part Numbers | Quantity |
|---|---|---|
| Presentation | SUN-S1-PRE | 4 |
| Business | SUN-S2-BUS | 2 |
| | SUN-S3-BUS | 2 |
| Integration | N/A | N/A |
| Resource | SUN-S4-RES | 2 |
| | SUN-S5-RES | 2 |

## Sun N1™ Service Provisioning System

The Sun N1 Service Provisioning System automates the process of installing operating environments, including installing hardware, operating systems, and applications. The software of the N1 Service Provisioning System holds a repository of profiles that can be deployed on any number of servers across the data center infrastructure. SOEs are great candidates for using the N1 Service Provisioning System because these environments contain the design specifications of the operating system and the applications to be deployed. The N1 Service Provisioning System can deploy any number of standardized operating environments and essentially provision the entire data center infrastructure.

### JumpStart™ Technology

JumpStart is a common technology that is widely used by Sun customers. It is included with the Solaris Operating System distribution. Online Sun BluePrints articles provide details on how to configure and create a provisioning environment: See John S. Howard's *Performing Network Installations Without a Local Boot Server* (http://www.sun.com/blueprints/0504/817-7288.pdf) and Pierre Reynes' *Configuring JumpStart Servers to Provision Sun x86-64 Systems* (http://www.sun.com/blueprints/0205/819-1692.pdf).

JumpStart can provision the operating system and applications so that installation can be automatically installed using scripts. If the software elements in the upper part of the stack can be installed using the Solaris standard packaging formats, the complete stack could be provisioned with using JumpStart technology. Also useful are Live Upgrade, Flash, and the JumpStart Enterprise Toolkit (JET).

## Control Objectives for Information and Related Technology (COBIT) and the Governance of Stacks

The entire IT Infrastructure must be managed to keep it current, and standardized operating environments are not an exception. Managing these environments to keep the standards updated and in sync with technology developments ensures the continuity of the technology plan and the advantages of the optimization process.

The most popular governance framework for IT organization is the COBIT, which is described at the Information Systems Audit and Control Association (ISACA) web site (http://www.isaca.org/).

The creation and maintenance of SOEs contributes to the governance process by facilitating information and standards. SOEs also depend on the following processes to maintain value for the organization and the data center operations.

- Plan and organize. SOEs contribute to the creation of technology standards that fit in the infrastructure plan. Standardized stacks not only contribute to the simplification the data center, they also introduce a good level of predictability that assists in the creation of budgetary estimates for IT investments.
- Acquire and implement:
    - The definition of a standardized operating environment incorporates applications, and although an SOE is not specifically tied to the applications, it is tied to the overall environment for service delivery. The creation of an SOE promotes the mapping between business requirements and technology investments. Therefore, SOEs help create a compliance mechanism (standards) for applications.
    - The acquisition plan can be simplified significantly with the stack. SOEs could actually be part numbers that are provisioned directly by an internal or external integrator, which could ease the process.
    - The architecture principles established during the design process allow designers to review, compare, and evaluate new technologies and their contribution to the systemic qualities. Because each element of an SOE has a relationship with the systemic qualities that it supports, technology updates can be incorporated easily by following the systemic quality guidelines that supported the creation of the standardized operating environment.

- Deliver and support:
    - The architecture principles definition and the standardized operating environment implementation facilitate the capacity planning and performance planning processes. After the standardization process is achieved, scaling the infrastructure becomes a process of adding additional SOEs (for those tiers with horizontal scalability, per the architecture design), modifying existing SOEs to increase their vertical computing reach, or both.
    - Configuration management is also greatly improved because there are fewer environments for which the configurations are specifically detailed and integrated. Because the stacks are built from a known specification, variations, adjustments, and changes are replicated easily.

      SOEs and Problem Management have a symbiotic relationship. SOEs are refined, updated and matured by the process of identifying the root cause of anomalies in the stacks and by incorporating changes that do away with the problems. Having standards across the organization on the other hand allows for better data gathering and analysis, since there is a controlled set of variables that narrow the possibilities for identifying the causes of problems.
    - SOEs also help to increase security in the data center. Glenn Brunette discusses this concept in his blueprint *Toward Systemically Secure IT Architectures* (http://www.sun.com/blueprints/0206/819-5605.pdf). Brunette suggests that SOEs add security to the infrastructure in the following ways:
        - Using SOEs means that all systems will be built against a known specification. It becomes easier to detect variations from the baseline specification that could indicate a security incident or a violation of change management or configuration control.
        - Using SOEs means that organizations will be able to respond more effectively to zero-day exploits and security-critical patches, because the organizations will know more quickly and easily what systems may be patched, what mitigating controls are in place, and so on.
        - Using SOEs also offers a greater chance that there will be less variation in the infrastructure with respect to vendors, product versions, product patch levels, and configurations. This means that the environments can be assessed more easily for compliance and managed to ensure that security controls are enforced consistently throughout the enterprise.

## Summary

This article elaborated an approach for simplifying data center infrastructure through the creation of standardized operating environments. It proposes a model that includes the elements used to build the stacks, defines how they fit in the overall enterprise IT architecture, and explains what steps organizations can take to transition their current environments and keep their new environments updated after standardization has been achieved.

## References


- *Configuring Boot Disks* by John S. Howard, Enterprise Engineering and David Deeths, Enterprise Engineering
  http://www.sun.com/blueprints/1201/config-bootdisks.pdf

- *Building Secure N-Tier Environments* by Alex Noordergraaf - Enterprise Engineering
  http://www.sun.com/solutions/blueprints/1000/ntier-security.pdf

- *Protecting Investments through Technology Advancements* by Brian Down, Client Solutions Organization - Global Data Center Practice
  http://www.sun.com/blueprints/1005/819-3931.html

- The Sun Service Optimized Data Center (SODC) Program
  http://www.sun.com/products-n-solutions/sodc/SODCwp.pdf

- Service Definition Workshop
  http://www.sun.com/service/sdw/

- The Open Group Architecture Framework
  http://www.opengroup.org/togaf/

- SunTone Architecture Methodology
  http://www.sun.com/2002-0212/feature/side1.html

- Customer Ready Systems Program
  http://www.sun.com/service/crs/index.html

- *Performing Network Installations Without a Local Boot Server* by John S. Howard, Reference Architecture Engineering

  http://www.sun.com/blueprints/0205/819-1692.pdf

- COBIT Framework

  http://www.isaca.org/

- *Toward Systematically Secure Architectures* by Glenn Brunette

  http://www.sun.com/blueprints/0206/819-5605.pdf