A Blueprint for Better Management from the Desktop to the Data Center

March 2007
Table of Contents

Executive Summary .............................................................................................................................. 3
IT Infrastructure Library (ITIL) ........................................................................................................... 5
Blueprint for Desktop to Data Center Management ................................................................. 8
A New Model of Computing ............................................................................................................. 14
Appendix—Open Standards and Open Source ............................................................................. 16
Executive Summary

There has been no greater change in IT management thinking in the last 30 years than the burgeoning focus on Service-Oriented Architecture and Infrastructure. Simply put, instead of focusing solely on what IT work is done, organizations are now calling on CIOs to focus on how well IT delivers its full spectrum of services to support the bottom line.

If an employee needs access to a service, they don’t care where the persistent data is stored, how the client-side, middleware or server software is instantiated, which servers it’s running on and what operating systems are required. They never did care. They only want the service, preferably every time they require access, with no unexpected interruptions or processing delays.

We live in an on-demand world, and the technobabble IT department excuses and limitations are no longer tolerated by today’s businesses. Service-Oriented Infrastructure creates new challenges for CIOs who already are governed by legislation to ensure and attest to authorized access, control wasteful power consumption and balance over-provisioning with redundancy for disaster tolerance. Now, IT organizations are consolidating their operating systems, servers and storage, and are implementing technologies like virtualization to overcome the static limitations of service deployment and continuity management that exist in the physical world.

Delivering services to users requires successful implementation of all the management disciplines, but you must go further—these disciplines must interact seamlessly on behalf of the service delivery. For example, for the CIO to access SAP resources on the network, the system must understand authorizations, roles and other security issues; check on the configuration and patch levels of the access device; enact Change Management and potentially Release Management processes; track issues or problems with service continuity management; and potentially provision new service-oriented components, requiring licensing and approvals. At any stage in this process, dropping into a manual procedure will violate the service-level objective. All of these silos must work together to achieve CIOs’ dreams of an automated service delivery infrastructure.

Adding fuel to the fire are security risks, regulations (Sarbanes-Oxley [SOX] and the Health Insurance Portability and Accountability Act [HIPAA]), the need to implement IT best practices (the IT Infrastructure Library [ITIL] and Control Objectives for Information and Related Technology [COBIT]) and requirements set forth in new standards (International Organization for Standardization [ISO] 20000, the first international standard for IT service management).

“... IT needs to find clearer, more relevant metrics for showing business alignment and relevance, service quality supportive of that alignment, and cost efficiency in delivering its service products to the business. Moreover, IT is also being asked to support compliance and security initiatives that are adding extra relevance and context to this drive towards accountability. — Enterprise Management Associates, “Getting Started with ITIL’s CMDB...,” Sept. 2006

Most forward-thinking CIOs are basing their process-management improvements around ITIL. Any successful product architecture will need to be built from the ground up with ITIL services as its central core. With the publication of ITIL 3 (ISO 20000) in 2007, a large increase in the adoption of ITIL is expected in the U.S., which currently lags the rest of the world in ITIL adoption.
No single management software vendor will be able to provide all the pieces required to deliver on CIOs’ dreams for a Service-Oriented Infrastructure. Fortunately, standards for product interaction have matured to the point where one can depend on them to provide seamless integration of third-party and partner products as well as extensions customers make themselves. IT Service Management (ITSM) removes traditional IT silos, informal processes and firefighting, and this blueprint for management also identifies viable industry standards and points of integration with existing products and systems.

In this paper, we wish to continue our interaction with two diverse audiences. First, we plan to increase our support of the open source development community by creating an open systems management architecture that encourages innovation with respect to the higher-order problems of distributed systems management. We invite you to think big about code that manages storage for thousands of virtual machines (VMs)—one of many exciting challenges ahead. We think it’ll be fun. Second, this paper is written for those whose job it is to manage information technology. You are faced with the difficult task of figuring out what tools are available—commercially or via open source—and whether, when and how to use them, while keeping track of licensing and integration and providing determinism with respect to your business. And you are expected to adhere to the concepts of ITIL that provide increased service levels for lower cost.

Our strategy is to build an open management architecture that offers distinct value through sophisticated integration of otherwise isolated components:

“One of the overall design goals is to create a computing system which is capable of meeting all of the requirements of a large computer utility. Such systems must run continuously and reliably 7 days a week, 24 hours a day... and must be capable of meeting wide service demands.... Because the system must ultimately be comprehensive and able to adapt to unknown future requirements, its framework must be general, and capable of evolving over time.” – Corbató and Vyssotsky on Multics, http://www.multicians.org/fjcc1.html, 1965
**IT Infrastructure Library (ITIL)**

ITIL provides best-practice guidelines and architectures on all aspects of end-to-end service management to ensure that IT processes are closely aligned with business processes and that IT delivers the correct and appropriate business solutions.

ITIL is neither a technology standard, nor is it regulations, and therefore there are no entities—including tools, and people—that can be deemed “ITIL compliant.” Processes and organizations, however, can be assessed against the British Standard Institution’s BS 15000, the ISO 20000 standard or COBIT. COBIT and ITIL are not mutually exclusive and can be combined to provide powerful control of and IT governance for IT service management as well as a best-practice framework. Enterprises that want to put their ITIL program into the context of a wider control and governance framework should use COBIT.

![Figure 1: A comprehensive management blueprint helps align IT processes with user needs and business goals.](image)

ITIL is a comprehensive set of best practices that focus on determining IT service delivery and business efficiency. ITIL outlines methods for IT planning, models and processes, and it establishes the required roles and relationships to execute those processes.

The ITIL framework also establishes the working relationship among an organization’s service providers, which could include the service desk, application developers, roll-out teams, network managers, building technicians and outside contractors. It calls for unified processes for all service providers in an organization, helping them work together and coordinate projects more easily.
Today's IT manager is less interested in technology as a single means to solve problems and save money. IT technology and products alone don't yield the desired end result and return on investment. Both people and processes must be aligned for maximum benefit. Good processes comprise both technology and people to define work flow, operations, decision making and approvals.

Think of it in terms of rolling out a new desktop operating system. Tools may automate the physical delivery of the operating system and software, but if the local building technicians learn about the roll post-facto, the results will be disastrous. Several organizations must work together to ensure minimal disruption to service and to maintain high user satisfaction.

ITIL establishes a common language and terminology across both internal and external IT groups. For example, a Change Advisory Board (CAB) comprises representatives from various IT and service organizations and is responsible for analyzing and approving changes to the standardized environment. Decisions made by the CAB, along with reported incidents and their resolutions are captured in the Change Management Database (CMDB). This database of knowledge is made available to all of an organization's service providers for better communication and cooperation.

The ITIL framework provides an effective foundation for quality IT service management. It is, however, only a framework. The methodologies have been defined, but as you implement them you need to refine them to fit your organization and goals. If one of the processes is bad, it will affect service quality until you resolve the issue.

Defining and documenting your processes is an ongoing effort. It takes time, but you can consider it time well spent if you’re serious about implementing ITIL. In addition to helping provide swift service to your users, you need such best practices in place to help you capture and assess your corporate asset data for both financial and regulatory compliance needs—no matter how large or small your organization.

**ITIL Components**

ITIL’s two major components are Service Delivery and Service Support. Service Delivery and Service Support cover more of the day-to-day operational processes of IT management. Some of the most common ITIL components are:

- Configuration Management
- Release Management
- Change Management
- Incident Management
- Problem Management
- Availability Management

**Configuration Management**

Configuration Management provides the foundation for successful IT service management and underpins every other process. The fundamental deliverable is the CMDB, comprising one or more integrated databases detailing all of the organization’s IT infrastructure components and other important associated assets. It is these assets, known as Configuration Items (CIs), that deliver IT services. What sets a CMDB apart from an ordinary asset register are the relationships, or links, that define how each CI is interconnected and interdependent with its neighbors. These relationships allow activities such as impact analyses and “what if?” scenarios to be carried out efficiently.
out. Ideally, the CMDB also contains details of any incidents, problems, known errors and changes associated with each CI.

**Release Management**
The Release Management process takes a holistic view of changes to IT services, considering all aspects of a release, both technical and non-technical. Release Management is responsible for all legal and contractual obligations for all hardware and software the organization uses. In order to meet this responsibility and protect the IT assets, it establishes secure environments for hardware in the Definitive Hardware Store (DHS) and for software in the Definitive Software Library (DSL).

**Change Management**
A change is initiated to resolve a problem, and a proposal is submitted for approval. A detailed plan is prepared to implement change, with a rollback plan acting as a safety net. After implementing the change, the requestor needs to verify that the change was successful.

**Incident Management**
Incident Management is responsible for the management of all incidents from detection and recording through to resolution and closure. The objective is the restoration of normal service as soon as possible with minimal disruption to the business.

**Problem Management**
Problem Management assists Incident Management by managing all major incidents and problems while endeavoring to record all workarounds and “quick fixes” as known errors where appropriate, and raising changes to implement permanent structural solutions wherever possible. Problem Management also analyzes and trends incidents and problems to proactively prevent future issues.

**Availability Management**
Availability Management ensures that each service meets or exceeds its availability targets and is proactively improved on an ongoing basis. In order to achieve this, Availability Management monitors, measures, reports on and reviews the availability, reliability, maintainability, serviceability and security of each service and component.
Novell® has engaged in many months’ research with hundreds of CIOs and service partners. The result of this research is a blueprint for solutions that can attack the overall problem while still being useful in the individual silos. The blueprint looks at the fundamental elements from the point of view of both the CIO and ITIL.

Business Process and Technology

The ITIL framework seeks to bring both business processes and technology together via a series of interrelated management disciplines. The blueprint acknowledges this effort by echoing the need for business process and technology to work together. To simplify how this can be achieved, we present a set of blueprint blocks that provide a solution in CIO terms, followed by a mapping of the ITIL services used to answer those questions.

The blueprint has to cover all the computing resources in a typical organization, including personal, handheld and telecommunications devices as well as desktops, servers, storage and network connections. It must recognize and create virtual environments that emulate any or all of these resources. Finally, it must deal with applications and their virtual instantiations, with full knowledge of usage and licensing implications.

Discover

The first and foremost problem for CIOs is identifying what they have in their infrastructure at any given point in time. Although many tools are capable of discovering devices, there are often multiple entries for the same device and varying sets of data about the device.
Discovery includes accurately identifying and being able to describe resources. This can be as low level as processor type, reboot capability, virtualization capabilities, out-of-band management, hardware components and even server power supply rating. The services available to both users and other services must continually be discovered and put into a real-time service catalog; and just as important are application dependencies, in terms of both configuration and their interaction with active services in the operating environment. The only way for Change Management processes to understand the downstream impacts of possible changes is to discover the dependencies in the environment.

The discovery process can be restricted to various classes of IP subnets, and uses both agent and agentless techniques—including Internet Control Message Protocol (ICMP) ping, Simple Network Management Protocol (SNMP) Get and TCP port probing—to discover the IT infrastructure and its applications.

Applications are discovered by matching discovered artifacts with defined application attributes including file locations, registry settings and service signatures—a process called application fingerprinting.

Relate

Once all the resources have been identified, it is equally important to know how they interact with each other in terms of dependencies and capacity and bandwidth requirements. With the introduction of virtualization this issue becomes more acute. VMs require physical devices they can run on, “pinning” information such as IP addresses and storage location and control of the lifecycle itself. These relationships are captured in a “model” and stored in databases.

The model must build in permanent relation facilities for discovered resources. We are suggesting that the systems management blueprint have an evolving model of relationships, and that this be accomplished through a federated CMDB (FCMDB).

A CMDB is a database that contains all the details about employees, workstations, devices, incidents, problems and changes as well as complete details of all the components in the business. It provides the basis for a public knowledgebase of known errors and solutions, which helps employees resolve minor incidents themselves without contacting the helpdesk. It also provides a private knowledgebase where the support staff can get detailed reports about all assets, with problem histories, workarounds and temporary fixes included.

As mentioned earlier, within this context, components of an information system are referred to as CIs. The processes of Configuration Management seek to specify, control and track CIs and any changes made to them in a comprehensive and systematic fashion.

We suggest that the federated model, where many systems can be originators (and maintainers) of CIs, is much more practical for any but the smallest of organizations. This approach avoids the biggest pitfall of early ITIL implementations—the creation and maintenance of the universal CMDB. The federated approach allows the individual systems to continue as they always have, while also cooperating in a “virtual” CMDB that has access to CIs and their relationships.

The Discovery processes will populate the CMDB with CIs according to the evolving models. Outside systems such as human resources systems will also create CIs. For example, a typical policy would be “all employees and contractors can only be created by PeopleSoft.” The FCMDB will appear as a single database to the higher layers of the model without the pitfalls of actually creating a single, enormous, centralized duplicative database.
Contain and Instantiate
In the new service-oriented world, virtualization is critical. However, with virtualization comes a new set of management challenges. The introduction of virtual machine operating system “images” as a first-class IT asset necessitates OS image lifecycle management—for instantiation, usage and retirement. Additionally, cloning new OS images from templates versus creating and deploying new images is also required.

Cloning is optimal for transient workloads, compared to OS images that are version controlled as part of a change-control strategy. OS images must be managed, leaving IT to ask key questions: “How do I control who defines and creates an OS image?”, “What is the process for rolling out to production?” and “How do I manage change for virtual machines?”

A second challenge is that once multiple virtual machines have been deployed to a physical server, multiple applications and business systems are critically affected in the event of a server failure. It is no longer enough to just replace a downed server and suffer a short application outage. The blueprint describes clustered virtual machine servers for hosting services in virtual machines—with rapid redeployment of services should a physical server fail.

Today, businesses have to define “blackout” periods, the windows of time that IT “owns” in order to patch or update systems to better secure them or to support new application features. These blackout periods mean that some or all of the business must come to a halt for the blackout period. In our global economy, where business runs 24x7, this may cost a company millions in revenue. The business needs a way to assign priority between the update and the needs of the business. The ability to dynamically allocate resources through pre-emption, or starving can offer the business the best case for maintaining the balance between business process and systems maintenance.

The IT Service Management Forum (iTSMF) has published Software Asset Management processes that concentrate on the specific demands of managing an organization’s software assets and all issues related to asset use. Software Asset Management is comprised of three processes: Core Asset Management, Logistics and Relationships.

Core Asset Management Processes
Core Asset Management processes identify software assets, maintain information about them throughout their lifecycles and manage physical assets related to software. With the advent of virtualization, lifecycle control becomes far more complex.

In the asset model of the past, IT was challenged with recording asset “tags” for physical devices and tracking their whereabouts until they were fully depreciated and retired. Software was tracked by version and license count, and retired mostly by an upgrade.

In today’s virtualized environment, IT needs to know where an image comes from, why it was created, what state it is in, how to store it and make it available for use, and who can use it, for how long and when. The Contain and Instantiate blocks address these issues by combining new image-creation services with proposed standards for describing the lifecycle of the image-creation process.
**Logistic Processes**

The management data that describes the creation portion of the lifecycle covers only a portion of the VM lifecycle. Once the OS image has been created, it is necessary to store and “stage” the image for use. The storage process requires similar data collection: where is the image stored, who stored it there, and when, and how it is to be used? There are two types of OS images at this point: templates and deployable images. Templates are “gold master” images that are copied (cloned), and then deployed. Deployable images are directly moved to the target resource where they are run.

The Systems Virtualization, Partitioning and Cluster (SVPC) Working Group at the DMTF has defined a management model that describes these stages of VM lifecycle management. The blueprint uses this proposed standard to ensure consistent lifecycle management for all VMs, regardless of their vendors.

**Relationship Processes**

Relationship processes manage all relationships within the business and with partners and suppliers, according to agreed-upon contractual, legal and documented service terms and targets relating to software use. The blueprint addresses service terms and licensing compliance. Tools like Novell ZENworks® Asset Management provide both license checking and reporting. Policies govern the use of software and can be configured to provide alerts or restrict usage when one is out of compliance with licensing terms.

**Manage**

The fundamental core of the entire system are the individual “spokes,” the ITIL functional processes required—all or in part—to manage any sizeable IT organization. For our purposes, we will separate them into two categories: first, Service Management, which consists of both Service Delivery and Service Support, and second, Application Management.

**Service Delivery and Service Support**

The Service Management components of ITIL, as we stated earlier, deal more with the day-to-day support and maintenance processes of Service Support and Service Delivery. In this category, we also place IT Provisioning for both physical and virtual environments, self-service, storage management and license management. All of these processes create and/or deploy identities and CIs, and make changes to them according to policy. These processes can also accommodate human intervention, of course, as automated management policies cannot cover all contingencies.

A combination of two or three different systems are normally engaged to enable these tasks, and depending on the number of platforms supported, potentially several more: configuration and storage management systems are closely tied to particular platforms because small details can greatly improve the usefulness of a particular product. We expect that in the near future, standards maturation will allow for single systems to manage resources ranging from personal devices to the data center; specialized large systems will be brought along later.

Of course, the commercially available systems will cross over the theoretical lines drawn by ITIL. Provisioning, for example, is accomplished in change-, configuration-, storage- and identity-management software. The single-product process paradigm is becoming rarer, and even then, interoperability will be a basic requirement.
**Applications Management**

A key issue that has existed for some time is the problem of moving application developers and IT service management closer together. Service-management considerations within all phases of the application lifecycle have been seriously deficient. Applications need to be deployed with service-management requirements included: that is, they need to be designed and built for operability, availability, reliability, maintainability, performance and manageability. Moreover, they should be tested for compliance to specifications.

**Application Development** is concerned with the activities needed to plan, design and build an application that can ultimately be used by some part of the organization to address a business requirement. The blueprint introduces the notion of a creation service for images. This service will contain management data about the build process but will not address the need to “instrument” applications for better service management.

IT cannot guarantee the absence of problems in a production environment, so it must have established remediation procedures to discover problems, alert remediation systems and actually fix the problems. These systems are usually a combination of automation and human intervention. Some issues, like disaster failover, can be fully automated; others, like diagnosing a particularly tricky application problem, are almost entirely staff-centric. But most problems require a combination of automation and human intervention, if only to discover the problem in the first place.

This is also an area in which a pervasive identity security and management fabric can enable substantial evolution: as assets and individuals are persistently recognized, many problems will disappear, including authorization issues (passwords) and security violations. As these problems are resolved, the IT staff will have more time to discover and diagnose difficult problems.

**Orchestration**

The next challenge on the horizon is automating resources. CIOs are faced with the mandate to “do more with less.” To meet this objective, they resort to draconian cuts in the numbers or types of devices supported and in the people that maintain those resources. Simple “trusted” tasks are left to the management software to fix, or a failing device is replaced with another “hot spare.” Several management tools are trying to address these issues by introducing workflows or best practice policies that CIOs can trust and a smaller IT staff can easily use.

The panacea for management is to not have to know about it at all. The notion here is that one can define the business need in either policy or workflow—or perhaps in plain language—and the management system will simply figure out what resources need to be deployed when and where. Although this point has not been reached, given virtualization, such a system may be realized in the not-too-distant future.

The Orchestration block introduces the notion of workloads. A workload is a task that the business process requires to meet the objectives of the business. Within this block, we look at all the compute resources as the total compute power in the enterprise, both desktops and servers. When a workload is desired, it is deployed to the best location, either physical or virtual, in which it can be run. The orchestrator allocates or reserves compute resources for the workload. For example, there are times when the business needs to prioritize work to meet certain business objectives. At these times the orchestrator can preempt other workloads or move them to allow for the higher priority workload to execute. Should all the compute resources be busy at the time when the higher-priority workload is needed, the orchestrator can starve another workload by taking its resources and later giving them back.
This orchestration allows the organization to do what it does best, while understanding the service-level expectations of both the business and IT. Or in other words, it keeps the business moving!

**Visualize**
Finally, there is the notion that one console can do it all. In reality, there are three kinds of needs for which consoles may be developed. The dashboard view is targeted at the CIO who wants to know the overall state of the system. This management dashboard provides either color-coded event strings or colored-coded images that represent servers or applications. In the operator's console, on the other hand, workflows can be launched or a VM deployed or some other “operation” or remediation performed to keep the system up and running according to the Service Level Agreement (SLA). Finally, there is the developer’s console, which allows policies to be written, and VMs and workflows to be defined.

Any or all of these consoles may be graphical user interfaces (GUIs), command-line interpreters or Web pages. In most cases, they are a combination of all three. The CIO gets a simple overall view from a business perspective, while the IT personnel have to decide which console to use, and when.

**Monitor**
In the area of resource monitoring there are already many ways to observe information about common resources. In fact, many resource providers support several of the current standard application programming interfaces (APIs) in an effort to publish information to the widest audience possible. However, there is a constant tradeoff in performance and other factors when providing all the data to anyone requesting it. This has led to support for standardized, purpose-driven APIs that try to focus the amount of data being requested. However, although these APIs have often been agreed upon and implemented, the data made available is not always consistent. The units of measure as well as the meaning of the terms are critical. For example is "available memory" of 2000 a measure of megabytes of RAM or gigabytes of disk space?

Historically, each data consumer leveraged some mechanism—often a local agent—that managed its access to the resource through a published API. These agents were in conflict with each other, causing customers to pick specific vendor solutions that implicitly excluded others. Because use of the data varied, and vendors typically did not cover every use case, end users encountered significant problems. The users were also left with the cost of ownership and administration of these agents, as they had to mix and match agents depending on the tools they wanted to use. End users are now demanding agentless environments when environments provide native collection systems.

The price to be paid for increased automation is that we have to monitor the system and bring exceptions to light for remediation on a nearly instantaneous basis. We must continually monitor servers, OSes, applications and services to make sure SLAs are met and that the policies established do not result in over-allocation of capacity, power or storage. We must also be aware that systems break and that business processes may be poorly defined, allowing automation to run in ways that we might not anticipate. Monitoring (with control and remediation capabilities) is critical to keep the system from producing unwanted results.

**Security and Identity**
IT Security Management is the process of managing a defined level of security for information, IT services and infrastructure. IT Security Management ensures that:
Security controls are implemented and maintained to address changing circumstances such as changed business and IT service requirements, IT architecture elements, threats and so forth.

Audit results show the adequacy of security controls and measures taken.

Reports are produced to show the status of information security.

Under ITIL, the functions that we think of as “security” are separated into several processes. Identity management, security event management and patch management are part of security in most people’s minds, but these are separate processes under ITIL. The Security Management process involves taking standard policies and enforcing them in real time.

The core of our vision rests on an identity fabric that is woven throughout the entire model. All of the participants—employees, managers, customers, partners, devices and locations—must be identified and have roles assigned in order for policies to be effective. The only way to reconcile multiple discoveries on the same devices is to have a permanent identity that persists throughout the lifecycle of the entity.

For example, we may have a desktop machine that has certain components in a central database, including a biometric authorization component (fingerprint reader). When the CFO signs onto this machine, we want to check that the machine is authorized and is configured according to policy: it would be OK for a new application to be added (as long as it’s not a keystroke scanner!), for example, but not for the fingerprint reader to disappear. We need persistent identity of the machine, user and important components to reconcile ongoing discovery for our policy engine to make accurate decisions.

The Bandit project, [http://www.bandit-project.org](http://www.bandit-project.org), is a set of loosely coupled components that provides consistent identity services for authentication, authorization and auditing.

Bandit implements open standard protocols and specifications such that identity services can be constructed, accessed and integrated from multiple identity sources. Bandit components support many authentication methods and provide user-centric credential management. And Bandit is building additional services needed for Role Based Access Control (RBAC) and for the emission of records to verify compliance with higher-level policies.

One area where we are currently extending identity is in the realm of VMs. Not only will we identify potential VMs in a repository, but we will also be able to identify and track instantiated VMs in the production environment—as well as the applications running under them—for license and usage analysis.

**Audit and Compliance**

All IT systems, but especially those that touch customer records, financials and other strategic corporate data, must be auditable. Today’s standards for IT audits require a compliance monitoring system. These systems are constantly testing events to see if any activity violates corporate IT policies, whether those relate to outside regulations or simply are internal IT governance choices. Auditors do not prescribe how companies should comply, but they do insist on compliance processes. The blueprint must have compliance built in, and it must pervade all elements to succeed.

**A New Model of Computing**

The classic computer has CPUs, memory and disks to hold data when the power is turned off. Virtual memory gave computers the ability to present the illusion to applications of more main memory than was physically available. Virtual disks create the illusion of a larger or more fault-tolerant disk compared to the many physical
disks they comprise. VMs present the illusion of a whole computer that is actually contained by a real computer sharing its physical resources among competing virtual machines. Clusters present the illusion of a single reliable computer by coupling together and masking the failures of physical computers.

Virtualization eliminates physically imposed static boundaries: CPU, memory and disks are allocated dynamically. Services and data gain mobility: the freedom to optimally consume physical resources and the ability to rapidly switch to alternate physical resources while adapting to workload demands. High availability is a natural consequence of virtualized systems.

Today, data center computers (servers) are connected to disks over a storage area network (SAN). By removing and relocating storage from individual servers to a central network location, server form factors have shrunk. Blade servers are now popular. Blades are granted access to virtual disks (named storage containers) located inside SAN disk arrays. When a server fails, processing fails over to another server with access to the same SAN virtual disks. When a service (running on a server) runs out of storage, more space can be allocated from the SAN, using standard management APIs. When services themselves are virtualized, by hosting inside a VM, they gain the flexibility to migrate from one physical server to another.

Legacy line-of-business (LOB) applications are also being virtualized. Static, monolithic client/server software is being augmented or replaced with Web services. Web-based Service-Oriented Architecture (SOA) replaces earlier distributed object systems. There are new WS- protocols for anything that wasn’t XML-based before. And LOB applications now comprise a number of cooperating services. Infrastructure services provide naming, discovery, and via XML, a data integration and exchange format. LOB components execute in VMs and communicate using Web services protocols. SOA and WS- protocols are creating a new platform for distributed computing.

Finally, with so many distributed moving parts, identity management creates the infrastructure necessary to securely name and associate, authenticate and authorize service consumers with producers regardless of service type. Identity is the context that binds a flow of service requests all the way from the end user through multiple processing tiers to data on disks. Users are granted rights to services, and services are granted rights to other services. And if we haven’t experienced enough virtualization yet, identity itself has been virtualized by the notion of “role.”

The blueprint, based on ITIL, embraces this new model of computing and turns the CIO’s dream of service on demand as part of business solutions into a reality.
Appendix—Open Standards and Open Source

IPMI, SMASH, SMI-S, and SVPC-V are all DMTF CIM-based management profiles that describe models and APIs for server, storage and virtualization management. WS-Management provides Web-services-based access to these and other existing management profiles, via translation of model objects and relationships into XML schema and SOAP for transport-level interoperability.

Bandit
Bandit implements open standard protocols and specifications such that identity services can be constructed, accessed and integrated from multiple identity sources. Bandit components support many authentication methods and provide user-centric credential management. On this base of a common identity model, Bandit is building additional services needed for Role-Based Access Control (RBAC) and for the emission of records to verify compliance with higher-level policies. Since every IT resource has an identity, Bandit's implementation of open access control will provide the foundation for trust when managing virtual resources and their relationships.

Common Information Model (CIM)
A normative definition (and model) of management information for systems, networks, applications and services. CIM enables systems and management application vendors to exchange and act on semantically rich management information (www.dmtf.org).

Systems Management Architecture for Server Hardware (SMASH)
Novell was a founding member of the Open Management with CIM project, which promotes an open source implementation of SMASH and related profiles (www.omc-project.org).

Storage Area Network (SAN)
A storage-specific network that connects multiple servers to centralized storage subsystems. Fibre Channel and iSCSI are common technologies for creating a SAN. SUSE® Linux Enterprise 10 includes the latest open source support for iSCSI and Virtual N_Ports necessary for virtual machine mobility on Fibre Channel networks. Novell is also working to instrument storage components as described by SMI-S.

Storage Management Initiative Specification (SMI-S)
An ANSI standard for managing storage (network) devices (www.snia.org/smi).

Novell is a member of SNIA and participates in both the SMI-Lab and the Aperi open source SMI-S platform project (www.eclipse.org/aperi).

Virtual Machine Monitor (VMM)
Support for a number of execution environments on a single computer, each of which emulates the host computer. This provides services with the illusion of owning an entire computer, but one that is a "private" machine, isolated from others, sharing a single physical machine. The software that provides this capability is called a virtual machine monitor or hypervisor. Novell is working with the open source Xen hypervisor project and via the DMTF System Virtualization, Partitioning and Cluster working group, to help define an interoperable standard for virtual machine lifecycle management. For more information about this collaboration, visit: http://wiki.xensource.com/xenwiki/XenCim
**Web-Based Enterprise Management (WBEM)**
A set of standards developed to unify the management of distributed computing environments. WBEM provides the foundation for the industry to deliver well-integrated standards-based management tools, facilitating the exchange of data across otherwise disparate technologies and platforms ([www.dmtf.org/standards/wbem/](http://www.dmtf.org/standards/wbem/)).

**WS-Management**
WS-Management addresses the cost and complexity of IT management by providing a common way for systems to access and exchange management information across the entire IT infrastructure. By using Web services to manage IT systems, deployments that support WS-Management will enable IT managers to remotely access devices on their networks. Novell is working with the [www.Openwsman.org](http://www.Openwsman.org) project with the goal to provide a complete WS-Management stack, using CIM to provide the system management information. The main focus is to support mapping CIM data into WS-Management and reuse existing agents (providers) currently available.

© 2007 Novell, Inc. All rights reserved. Novell, the Novell logo, the N logo, ZENworks and SUSE are registered trademarks of Novell, Inc. in the United States and other countries.

*Linux is a registered trademark of Linus Torvalds. All other third-party trademarks are the property of their respective owners.*

© Copyright itSMF, 2004