



SOA REFERENCE ARCHITECTURE: SERVICE ORIENTED ARCHITECTURE

SOA Blueprint

A structured blog by Yogish Pai

Service Oriented Infrastructure (SOI)


As the infrastructure to support SOA, service-oriented infrastructure (SOI) requires the resources to compute, network, and store data: networks, routers, network appliances, servers, storage, data centers, VoIP, and wireless capabilities.

Currently, storage products have consolidated manageability models and operation procedures through the SNIA SMI-S standard. SOI will leverage that standard as part of SOI architecture. In contrast, SOI network integration relies primarily on proprietary protocols. There is no industry-wide interoperable management standard available yet, except the outdated simple network management protocol (SNMP).

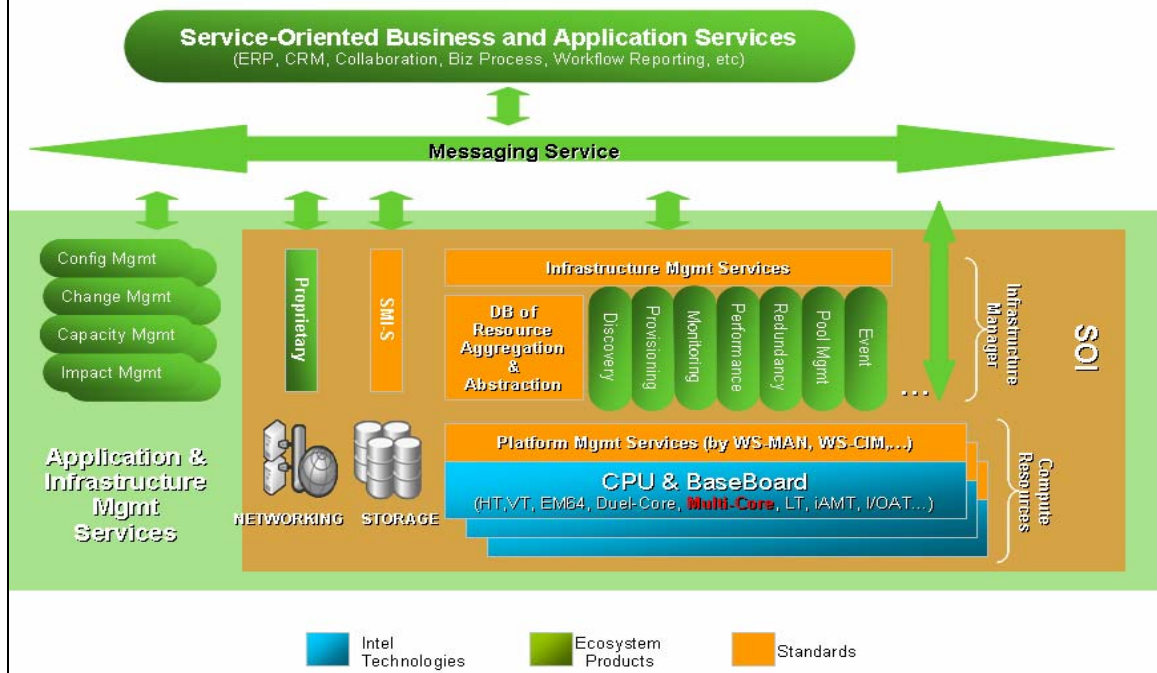
For computing resources, there are two work-in-progress standards developed by DMTF. The server manageability standard defined by DMTF Server Management Working Group (SMWG) focuses at the individual system level to define the server resource modeling before pre-boot. The Utility Computing Working Group (UCWG) defines the standard that deals with modeling management of the virtualized server pool. Architects can leverage these work-in-process standards as a foundation and propose additional extensions to take advantage of technologies for better performance and further capabilities.

There are two major components for SOI computing resources. The first is the infrastructure manager, software to aggregate computing resources and abstract them into a virtualized computing pool. The second is the computing resource itself, which combines software and hardware modules that enable the autonomic capabilities for compute resource virtualization.

Once the infrastructure manager aggregates and abstracts the computing resources, IT can choose its partitioning, such as a traditional one-server-one-OS partition, multiple OS partitions inside a compute system, or an OS partition span across multiple systems. Software features within the infrastructure manager enable IT to implement SOI services:



SOI Architecture



Service-Oriented Infrastructure Architecture

- Database of resource aggregation and abstraction:** this repository is the information foundation of an SOI architecture. It stores computing, network, and storage resources once they've been abstracted into logical models based on the common information model (CIM) of DMTF. Depending on the implementation, the repository might offer only simple resource configuration or it might include interdependencies among compute, storage, and network resources and their operation processes.

Once IT has defined the common abstractions of the resources and their interdependencies, this repository can share and synchronize this information among internal and external applications to ensure the interoperability and consistency of resource management. The database also includes operation processes, the generic IT micro-operations that serve as building blocks to standardize IT processes. For example, server provisioning can be simplified by a generic operation that orchestrates configuration procedures across server OS installation, network switch VLAN settings, and SAN storage partitions. Currently, the DMTF Utility Computing Working Group is working on these generic micro-operation processes for server virtualization. As the scope expands, the UCWG or other standard consortiums will standardize more resource operation processes.

IT staff can leverage these building blocks to develop more complex and custom processes for their own environment. Product vendors can extend operation processes to make better use of their product features. Vendors could differentiate themselves by extending the operation process for "bare metal" provisioning based on active management technology (AMT).

- Infrastructure management modules:** these software modules provide the execution of services. The services can include automatic resource discovery, provisioning of an OS virtual partition, resource health monitoring and its event notification, and resource pool management. An SOI SDK can help accelerate the development of these management modules by providing software examples and libraries that show developers how to interact with the abstraction and



aggregation resource database and how to register or advertise WSDL services to the external applications.

- **Infrastructure management protocol:** this set of XML protocols enables the upper-layer business or system management applications, either web services-enabled or legacy, to interact with underlying resources. SOI management modules encapsulate the complexity, keeping it from the business or system management applications, and maintain a runtime detail internally. This management protocol specification defines the details of the resource object handlers that manage the interaction.

For example, a J2EE application server may reach its performance threshold and need more computing resource to comply with its SLA. Then, it can send an XML resource request based on the needed jAppServer SPEC benchmark to the SOI to ask for additional computing resources. The request may specify the application configuration and the operation process to join the J2EE application server cluster. However, there is no need to specify the detail system and OS configuration in this XML request because SOI will automatically validate the available J2EE application server platforms and reply with available options in the SOI. This can be based on either a dedicated Xeon server or a shared Itanium 2 4-way server using a specified jAppServer SPEC benchmark and available OS and availability. The J2EE application can select one of the options depending on availability or cost of resources. SOI will maintain the configuration context across computing, network, and storage resources, and encapsulate the detail from the business applications. This is just a starting point for resource performance metrics. It still needs many improvements and extensions with the current SPEC benchmark metrics in order to fulfill the final goal of SOI.

Beyond the infrastructure manager mentioned above, several key hardware and software features on the computing resource itself enable implementation of SOI services.

- **Asset identifiers:** provide a dependable way to remotely and uniquely identify and catalog physical resources attached to a network. This enables an enterprise management console to discover and account for all IT assets connected to the corporate network, including wireless devices.
- **Out-of-band (OOB) accessibility:** allows network access to platform management functions even if the platform is otherwise non-functional because it has not yet had operating system or application software loaded, or due to a software bug, system failure, malware attack or other problem. This does not require another network cable, but provides alternate paths and intelligence within the platform itself. This feature enables remote "bare metal" provisioning, reloading software that has become corrupted, running remote diagnostics, remote reconfiguration, and many other functions.
- **Service processor:** like OOB facilities, a dedicated management service processor supports functionality even when the primary platform processor(s) are not yet provisioned with software or are not functioning properly. A separate service processor also enables more effective monitoring, failure detection, and diagnostics in any platform system state. It also allows the platform to perform autonomous functions such as periodic self-test and diagnostics, automatic hardware and software configuration discovery and reporting, and self-repair actions, such as isolating itself from the network and sending an alert when it detects a virus.

To deliver interoperability among all the hardware and software components that make up the service-oriented enterprise, the SOI must be based on open, industry-wide standards that define common functions, data structures, and interfaces.





The Business Value of Services Oriented Infrastructure (SOI)

SOI delivers bottom-line benefits to the enterprise. It provides the basis for greater IT automation which results in higher IT productivity and lower operational costs. SOI also makes it possible to move from the static, “one-box-per-application” approach to dynamic resource allocation in which virtual processing, storage and network resources are allocated to applications as needed. This results in reduced capital costs through better resource utilization. It also yields higher reliability, with fewer service outages and delays, since applications can fail over to available resources without disruption of the application. IT can deliver more consistent service levels since software can automatically allocate additional resources in real time as an application’s workload increases.

Even more important in the long run is SOI’s contribution to overcoming the “complexity barrier” that limits organizations’ ability to design and implement new information-supported business processes. With an SOI in place, a business can reallocate IT overhead dollars to investment in new innovative systems that create true business differentiation. For example, SOI can help enterprises take full advantage of autonomic data sources such as RFID or eForms in real-time when the data becomes available. SOI enhances the ability to implement and manage event-driven, message-based services and applications inside and across the corporate firewall. Better management of mobile wireless devices allows the business to push services to the edge of the network to make employees more productive and to deliver value-added services to mobile customers. SOI also enables greater business accountability, for example, improving Sarbanes-Oxley compliance in the area of IT asset inventories for accurate depreciation schedules, and providing timely software version and license management. These are high-impact capabilities—one high-tech company saved 40,000 person-hours in just five quarters by using automated IT asset inventory techniques.

From the unique perspective of IT management, SOI offers many benefits:

- Reduced operations workload due to reduced manual reconfiguration of hardware and software.
- High productivity resulting from the ability to perform enterprise-wide platform, network, data, and applications management from a single, standardized management console.
- Better resource utilization leading to reduced capital expenditures to accomplish IT objectives.
- Greater flexibility through dynamic resource allocation, as well as the option to focus on core competencies and move commodity IT services, such as corporate e-mail, outside the firewall to 3rd party providers. The ability of SOI to operate across firewalls means that IT can manage services consistently whether they are sourced within the corporation, or outsourced to external providers through telco or ISP connectivity.
- Simple and cost-effective upgrades thanks to a modular, loosely-coupled SOI architecture. This allows IT to refresh its infrastructure to take advantage of new technologies without complications resulting from hard-wired management dependencies. This eliminates painful “rip-and-replace” (aka “forklift”) upgrades, allowing IT to respond to change in a rapid and graceful manner, and reducing the inertia that causes IT to resist incorporation of new technologies because of expected disruptions
- Support for on-demand usage models which allow more accurate alignment of costs to benefits through pay-for-use chargeback methods.
- Support for managing all client platforms ranging from desktops to mobile hand-held devices, as an integral part of the infrastructure with common platform features and access interfaces.

Developers also benefit directly from evolution to a service-oriented infrastructure:

- SOI removes resource management and security responsibilities from the application code. Infrastructure management protocols ensure these functions are architected into web service-enabled hardware, SOI middleware, and applications building blocks from the bottom up and the inside out in a standardized, consistent manner. Rather than having to make critical assumptions that will invariably change, or embed logic to take care of this in the application





code, developers need only specify virtual application requirements. They can then rely on the SOI to ensure that these are met dynamically.

- Developers can use pre-built services for distributed, real-time, event-driven, multi-platform applications because these elements are already architected to be consistent with, and take advantage of, a standardized SOI. Because developers can easily reuse components they have built for other services and applications, they are able to build and modify applications more quickly, and therefore can deliver more innovation and business value.
- Developers work more efficiently in heterogeneous environments, including cross-enterprise Internet/Web applications, because they're using building blocks that are web-based, service-oriented, and virtualized, meaning they conform to industry-standard capabilities and interfaces.

Copyright

Copyright © 2007.

The author grants a non-exclusive licence to everyone to publish this document in full or part by acknowledging the source. Any other usage is prohibited without the express permission of the authors.

