2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises

The Virtual Business Services Fabric: an integrated abstraction of Services and Computing Infrastructure

Pankaj Goyal, Ph.D. *Senior Member, IEEE*

Abstract— **This paper introduces (i) the concept of Business services as a fabric that is a network of networks consisting of service components; (ii) proposes partitioning and composition schemes for creating a virtualized services fabric; and (iii) an agent based infrastructure services fabric to implement knowledge based policies and business workflows. The proposed VBSF – a model for heterogeneous virtualization and abstraction of services, applications, policies, capabilities, resources, infrastructure and, even, people – allows dynamic allocation of resources and dynamic assembly (composition) of business workflows. This model is applicable to services and infrastructural environments that transcend public and private clouds, services, applications and infrastructures.**

Index Terms—**cloud computing, virtualization, services**

I. Introduction

The evolution of computing is the evolution of abstracting away from underlying resources – from storage to network, a separation of concerns – from interaction management to business rules and their implementation. Recent evolution allows orchestration of independent services and, meshing of services by the user to create new “virtual application” or change the behavior of existing applications.

In the infrastructure area, the evolution from stand-alone servers to the grid and beyond, wide area networks with latencies equivalent of local area networks, massive storage arrays is enabling cloud computing that makes IT to be capacity and location independent. Cloud computing delivers computing infrastructure as a service based on the use of virtualized resources instead of physical resources. Cloud computing utilizes technologies and concepts from grid computing, virtualization, utility computing, and software as a service.

The emerging computing infrastructure fabric – an infrastructural resources constellation – promises integration between private and public clouds and private computing infrastructures (data centers and corporate network), and client devices (for example, desktop, PDA). Public and private clouds integrated with private computing infrastructures, provide the capacity for unprecedented business agility.

Traditional IT applications execute on specific hardware, operating system, middleware and/or on a specific network. The applications provide a pre-defined set of services invoked in hard-coded sequence – logic, business rules and process. This makes application adaptability to changes expensive and the applications limit the user to essentially a “read and write” (or a “get and send”) interaction.

The new IT is more user-centric and collaborative providing an a la carte choice of services, compatible services, ability to quickly create new “meshed up” services and the ability to modify/adapt processes. A process specifies the sequencing of and interactions between services, where the process, services, the interactions and the deployment and operational resources are all “services.” These “services” include the applications, business processes, and the resources that they consume. The collection of all services constitutes a business services fabric. The evolving business services fabric permits the use of services with different deployments and different non-required manageability and operability capabilities.

This paper uses the term infrastructure in its broadest sense as “an underlying base or foundation especially for an organization or system; and The basic facilities, services, and installations needed for the functioning of a community or society, such as transportation and communications *systems*, …” [1] and not in the narrower sense where infrastructure refers to only the computing infrastructure.[1] While fabric is defined as a frame or structure of connected parts, in usage not all components of a logical frame may be explicitly connected but they are connectable, for example, by other services.

Computing infrastructure consists of:

- Equipment: includes enterprise servers, storage, network, security devices
- Facilities: that house, protect and power the equipment including data centers, power and cooling systems, backup generators and security systems
- Management systems: that monitor performance of the infrastructure and to make changes as necessary. Typically, management software is provided with individual infrastructure elements, but it is hard to get a complete integrated view across the elements.

A Business Fabric integrates the functions and capabilities of cloud computing, private computing infrastructure, computing middleware, services middleware and the services themselves to enable a highly dynamic, agile, adaptive “infrastructure” for business applications/processes. A Business Fabric provides a general purpose infrastructure for

P.Goyal (pgoyal@ieee.org) is a principal with MicroMega, Denver, CO 80250-1496 USA.

[1] The term computing infrastructure is used to include computing elements, storage elements and interconnecting network.

1524-4547/09 $25.00 © 2009 IEEE
DOI 10.1109/WETICE.2009.19

arbitrary business applications, primarily aimed at providing business flexibility efficiently. Business Fabrics include

- the business services fabric (also the applications fabric)
- the services middleware fabric – tools for assembling services, discovering services, interaction of services, service managers (e.g., Google AppEngine)
- the services infrastructure fabric (e.g., the resource independent service abstractions of the underlying fabrics – data service, queuing service – and also the available services),
- the computing middleware fabric (e.g., DBMS, ESB, J2EE, .Net)
- the computing infrastructure fabric.

This paper presents the Business Services Fabric – a model for heterogeneous virtualization and abstraction of services, applications, policies, capabilities, resources, infrastructure and, even, people. The applicability extends to, for example, future internet; [2] presents a services model and capabilities for next generation internet. Goyal et al [3] present how Virtual Business Services Fabric simplifies manageability and operability.

II. SERVICE MODEL

A number of models for services exist including an extensive model developed for web services [4]. The service model presented here, an extension to the web services model, models everything as a service; specialized terms such as process, agent, action, task, and message, are used only for ease of communication. Only model elements that differ from the web services model [4] are presented here.

In this model, a service is an abstraction of anything that can perform a task or a collection of tasks; for example, a person, a computer, a storage device, a message. The service may be provided (or in computing realized) by one or more provider service agents that act on behalf of the provider entity (or owner); for example, a customer service agent acting on behalf of the company provides an authorized set of services (or performs a certain set of tasks). In the example of the customer service agent, the agent must be *authorized* to perform the services but the customer must also be *eligible* to receive those services or have those services performed on its behalf.

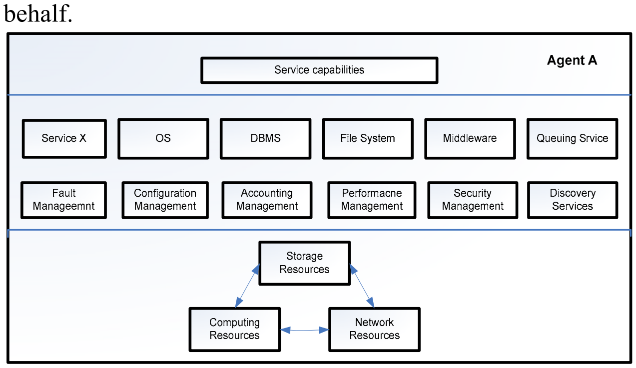

Figure 1. Conceptual Model of a Service Agent

The service description details the service interface – the data types, operations, transport protocol information, and identifier. The service description also describes the manageability and operability capabilities of the service. The service Interface defines the messages that the service makes available, and the message exchange patterns for interacting with the service.

It is important to note that service descriptions and interfaces may be private, and may have multiple implementations depending upon the capabilities of the requestor and provider service agents. For example, a requestor service agent broadcasts a query that includes its security credentials and a specification of the nature of services it desires. Provider service agents, that meet these requirements, may then respond to the Requestor service agent with a service description, including the interface available to the Requestor service agent where the service description and interface specifications may depend upon the security credentials of the requestor service agent. This permits differentiation, for example, between the messages that requestor service agents invoke for the performance of some task (or the "business" operations) from the messages that, say, a management agent invokes for the manageability and operability of services and their agents; the term "management interface" to qualify the latter is used here purely for convenience. The messages of the management interface can only be invoked by agents with the requisite administrative privileges and are subject to the policies regarding such management. For example, if a management agent, even with the appropriate permissions, changes a managed service agent's configuration, the policy with respect to such a change may require the execution of configuration change management; save current configuration, validate new configuration, obtain approvals and/or report changes.

An agent[2] realizes or implements a service – they perform the tasks, actions, and policy management of the service's capabilities; a service can have one or more realizations. Multiple agents can realize a service, where the agents have certain capabilities and may utilize a different set of services/resources in implementing the service (Figure 1). Agents may implement service features in different ways, for example on different underlying resources. In this model, agents may also implement or facilitate different management or performance capabilities, for example, monitoring and QoS. The concept of these capabilities is broader than that specified in SOAP. For example, a Service S_0 realized by agents A_{01} and A_{02}, where A_{01} guarantees a better performance than A_{02}. Or it could be that A_{01} is an implementation that conforms to laws of country X whereas A_{02} can be used in all other countries except X. In these implementations the service signature, messages, description, semantics etc. may all be the same; the difference maybe in the use of, say, local resources or a particular resource.

In this model, the requester agent may specify what it needs but the provider service agent may impose constraints on the

[2] Nothing in the model precludes agents to be people despite language that imply a computational entity.

requester agent. For example, a provider service agent may impose requirements for the requester agent to encrypt all message exchanges or to ensure that it is operating in a specified domain (secure, country, etc.).

In this service abstraction where everything is a service, including messages – messages are subject to policies, which themselves are services. A service is composed of other services and interacts with some other set of services through messages. A message can be externally visible or maybe an internal message. The model allows services to interact and collaborate with each other; collaboration, for example, is useful for manageability and operability, including recoverability.

Both, choreography and orchestration prescribe the sequence of interactions for the completion of some function, for example, the placement of the order; choreography is modeled as a composition of services, whereas orchestration is modeled as a management service. Choreography allows one service to invoke the next service(s) interaction in the sequence; in orchestration the completion of one service results in the orchestration service invoking the next service(s) interaction.

The resources, the relationships between these resources (and services), and their virtualization are all modeled as services; resources can be servers, data stores, network, people, facilities, services, etc. Agents reference and use resources. Policies constrain the behavior and utilization of resources, and apply to (or govern) Agents.

Policy is a constraint on the behavior of agents as they perform actions or access resources. There are many kinds of policies, some relate to accessing resources in particular ways, others relate more generally to the allowable actions an agent may perform. "Policy" agents establish policies like defining security properties, quality of service properties, management properties and service configurability properties. Management agents can enforce policies. Logically, two types of policies exist: management and governance. Governance policies are usually obligations, for example, the obligation for employees to work a certain amount of time. Obligations can by their nature not be proactively enforced; they can only be validated for compliance/violations. The management policies govern permissions and management capabilities such as fault, configurability, accounting, performance and security (FCAPS). The management policies can and must be proactively enforced and/or managed. The permissions aspect of security can be handled from both within the agent and the outside – mist likely a combination of the two. Permissions proscribe rights to perform some action. An action is not performed if the requestor agent does not have the necessary permission to perform the action in that specific environment. Some capabilities are managed internally and some externally; some capabilities require external management agents.

Management Services, realized by agents, enforce policies; an agent A_0 realizing a service S_0, thus, may be composed with a number of management service agents that manage the set of capabilities of the agent A_0, and the agent A_0 may also be subject to management by a set of external management service agents within the same or a higher layer. This is very important as only certain capabilities can be managed internally while others may require external management. Put another way, a management service manages agents at its layer and performs "consolidation/correlation" from the lower layers for its handling or escalating to the next upper layer; figure 2 shows the scoping of a service (or its realizing agent).

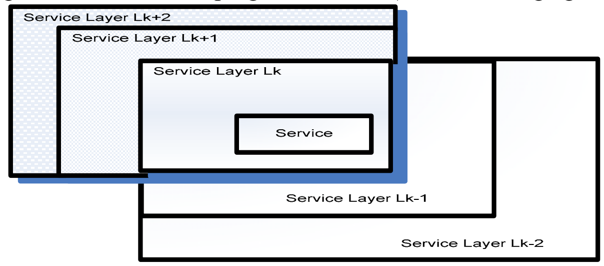

Figure 2. Scope of Management Messages

A service can be "composed" from other services by combining them in some fashion; service and agent composition allows for dynamic deployment of new and modified functionality (Figure 3). Similarly, the realizing service agent implemented as a composition of other service agents and custom functionality. When a service agent, *Ax*, is composed in this manner, the constituent agents gain access to the internal messages and have visibility into the implementation of *Ax*. This composition capability allows for the dynamic deployment of new/changed functionality by, say, modifying a constituent agent.

The services, more correctly, the very loosely coupled agents hide the underlying implementation. In this way, different agents can implement a service and if changes in an implementation are compatible with the existing interface then they would not impact the interactions with any connected services; implementation changes may impact performance and/or other message independent characteristics.

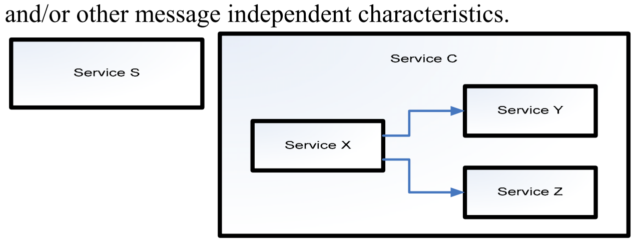

Figure 3. Service Compositions

A simple Service task that takes a small known finite "short" time and that does not need collaboration can do with the typical single message (or I/O) channel. A Service performing a task that takes a long time (most likely asynchronously) would need to be communicated with. It may be to access the status or to signal external environmental changes. This requires a control channel for the Service so that it can then receive the necessary messages and respond to them.

A Service collaborates with other Services through a "channel pipe" – the collection of a set of control channels and the associated set of message channels; the implementation consists of logically independent control/signaling and information/message channels (in the theoretical BPM, multiple channels exist between interacting processes).

The same concept of collaboration can manage a Service for fault, capacity, accounting, provisioning and security management (FCAPS). These FCAPS management services set up a collaborative network with their managed Services.

A. *Clouds*

At their most basic, cloud-computing provides computing infrastructure consisting of networked (typically, Internet-connected) servers that house applications and data. They also include virtualization, grid, management, database, and other types of software; user interfaces; APIs; a communications infrastructure for connecting to users over the Internet or a private network.

The cloud provides almost infinite flexible and scalable external computing and processing services that not only offer significant cost benefits, but also provide the ability to connect with customers, partners and suppliers like never before. Cloud computing is the IT environment – encompassing all elements of the full "stack" of IT and network products (and supporting services)—that enables the development, delivery and consumption of cloud services.

Cloud services are applications (or services) that are delivered using cloud computing. Clients (or users) generally use browsers or dedicated client applications to access cloud services.

In addition to Amazon, EMC/VMWare, IBM, Google and Microsoft, more than a dozen other companies are actively engaged in offering or in developing cloud computing [5]. A number of companies are also working on cloud management tools; Google and IBM are even collaborating in this area.

B. *Data Service*

This section presents, as an example, an abstraction of an underlying resource – data resources. An abstraction of the underlying data, as a data service, may provide different interfaces to operations for modifying or accessing the underlying data, different performance characteristics and different manageability capabilities. The data service may abstract or virtualize data from one or more data resources.

The abstraction or virtualization of the underlying data is provided by mapping from the underlying data resource, a subset of the data resource or from multiple data resources and/or data services; the mappings may be one-to-one, many-to-one, one-to-many, or many-to-many, and may involve computational transformations of the underlying data. The data service implements the mapping from the underlying data resource(s) implementation. The data service encapsulates the implementation and is not visible or accessible to users of that data other than what is permitted by the service's interface. The management interfaces, however, may provide greater visibility to the data service implementation.

A data service may allow concurrency, and/or providing for concurrency control mechanisms in its interface.

III. BUSINESS, SERVICES AND INFRASTRUCTURE FABRIC

The entire global interconnection of computing infrastructure, services and applications is encompassed in the Infrastructure, Services and Business fabrics, respectively. An executing business process (or application) does not have to be cognizant of all of the underlying components that enable its execution. The layering of the components is a logical, and, in many cases, an implementation mechanism that hides lower level component services from their higher level component service users. A service in a lower level layer may be used by a service in a higher level layer, but never vice versa. Figure 4 shows one possible layering – the organization of the component services – of the fabrics with business-related services in the higher layers and hardware related services in the lower layers; this is not the only possible layering and the boundaries are not well-defined. While services maybe used in any combination, some services are inter-related and hence the use of the term fabric in the layer names.

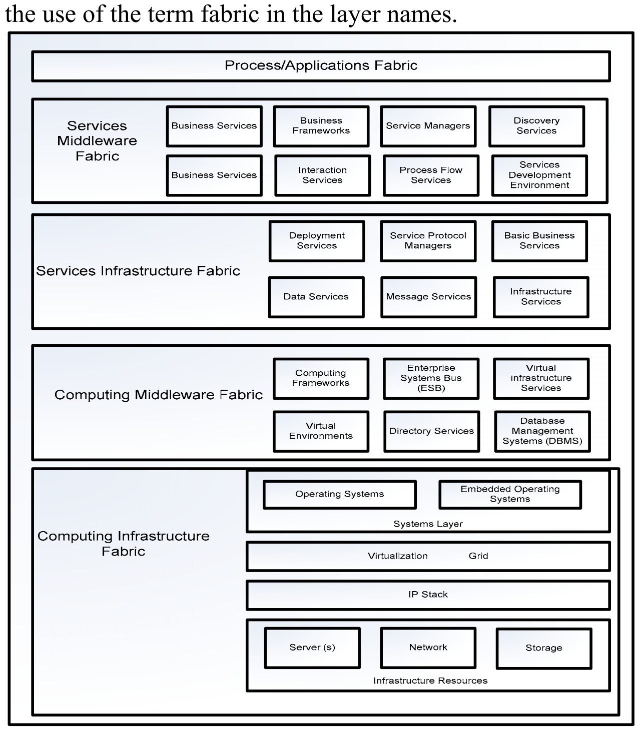

Figure 4. A high level model of a Business Fabric

Computing Infrastructure Fabric: houses the basic services (resources) such as compute power, storage, network, sensors, actuators ...) together with associated network stack, resource virtualization and grid capabilities, and low-level software components, for example, operating systems components. Operating systems offer a number of services, including but not limited to, management of memory and processing resources, multitasking, device access and control, file systems, networking and security. In the services composition model, a service realization may only incorporate operating system services needed by the service; the same is true for other system capabilities such as database management systems, and file management systems.

Computing Middleware Fabric: Traditionally, middleware covers all software that is an intermediary between the traditional application software and the underlying systems software and hardware resources (those provided by the computing infrastructure fabric). The computing middleware fabric covers the traditional middleware software functionality. For example, application frameworks (e.g. J2EE, .Net), database management systems/services, messaging services and portal frameworks and also the virtual computing infrastructure services useful for hiding the Computing Infrastructure Fabric (computing technology) complexity, such as distribution, heterogeneity, operating systems dependency) from higher level business and utility services that are ultimately used in applications. The computing middleware fabric also contains frameworks with services that enforce policies; services that allocate, access, distribute, monitor, optimize and secure computing services (resources); and services that manage faults, performance, and configurations provided by the Computing Infrastructure Fabric.

Services Infrastructure Fabric: is the services equivalent to the computing infrastructure fabric. It provides a set of basic service-utilities that are the building blocks of "computing" independent services and allow for the assembly of process flows/applications independent of all underlying hardware and software environments. The fabric consists of services like data services (data abstraction, data manipulation capabilities), message services (synchronous, asynchronous, guaranteed delivery, logging), (computing) infrastructure services, service protocol managers, and management services (manage faults, enforce policies, performance, configure, allocate, access, distribute, monitor and security).

Services Middleware Fabric: is the services equivalent of the computing middleware fabric; Google Application Engine or Microsoft's Azure are early and evolving examples of services middleware. It provides all of the intermediary services that mediate between the applications (service process flows) and the underlying services infrastructure services. Examples of the capabilities of a services middleware include:

- Interaction services enables links to be established between services independent of the underlying message services (synchronous, asynchronous); Interactions can be of the following types: Query, Initiating (for example, the first step of a sequence), Defining (for example, specifying the sequence of actions), Informational, Response and Control (for example, alarms, abnormal action). The underlying implementation, at the physical network level, may be a single cable, a bus or multiple cables.
- Business services development environment enables new business services to be created, debugged and simulated.
- Process Flow Services allows for the creation and execution of different types of process flows (allow workflows to be defined); process flow services, thus, incorporates a process execution engine – the creation and execution environment can easily be separated.
- Business Services frameworks are services that enforce services level policies; services that allocate, access, distribute, monitor, optimize and secure business services; and services that manage business services faults, business services performance, and business services configurations. The business services frameworks may also include orchestration and choreography services.
- Specific Composite and General Business Services: are reusable compositions of services that are used repeatedly in many business processes.
- The fabric controller and fabric mediator services for the management of virtual business services environments.

Process/Applications Fabric: covers the set of processes and applications assembled from the available services and built upon the capabilities of the underlying fabrics. Manager services control the actual initiation, execution and termination of the agents that realize these services.

A. *Fabric Partitioning*

The implementation of a business process/application utilizes a subset of the available services through the partitioning of the fabric into virtual fabrics: Virtual Process/Application Fabric (VPAF) that utilizes some services from a Virtual Services Middleware Fabric (VSMF) and is realized using some services from a Virtual Services Infrastructure Fabric (VSIF). The VSIF is deployed using certain services from the Virtual Computing Middleware Fabric (VCMF) and certain services (resources) of a Virtual Computing Infrastructure Fabric (VCIF). These virtual fabrics allow, for example, restriction on the possible interactions, the type of resources, their location, manageability and operability options. Mobility can be restricted to well-specified service implementations (agents), network segments, client devices, compute servers and data servers.

A VCMF restricts the uses of a computing framework (for example .Net), the type of DBMS (say, MS SQL). The VCIF provides a partitioned infrastructure that seamlessly integrates the public and private clouds, company owned and partner owned data center resources. The VCIF, itself being a service, provides a public interface, is governed by policies that include restrictions, for example, on visibility and access. In the example, where a VCMF utilizes .Net and MS SQL, the VCIF would ideally consist of Microsoft Windows environment. A particular business process can be, thus, implemented using services, where these services are optimized to execute on different sub-fabrics based on their needs and policy specifications. The VPAF and VSMF restrict the services available, in particular, their realizing agents for use in the process flows and applications. For example, if the client service is running in a "noisy," "limited bandwidth" and "unsecure" environment, then the client services and their possible usage may be curtailed to prevent chatty or secure interactions.

The partitioning of Fabrics into sub-fabrics – virtual fabrics – provides tremendous flexibility in business process implementation under will-architected safeguards. The sub-fabrics restrict the permissible interactions between services. Consider, for example, a VPAF V that consists of, say, two sub-VPAFs V_a and V_b. Thus, a broadcast message from Agent A_{a0} will only be seen by other agents of the sub-VPAF V_a.

In the infrastructure area, a sub-VCIF creates, say, a proprietary and secure capability over the company and/or partner resources; this sub-VCIF can be specified for reliability to only be replicated in another similar proprietary and secure company or partner owned facility. The sub-fabric mechanism, for example a sub-VCIF, permits different infrastructural capabilities, such as bandwidth, communication protocols, nature and quantity of computing resources. Note that this mechanism allows fine tailoring of the bandwidth requirements to the needs of the interacting services and a customer does not have to pay for the maximum bandwidth utilized but the bandwidth utilized in each segment. The provider can also better manage its resources and finely tune them to the expected customer demands. This sub-fabric mechanism can restrict proprietary data and/or sensitive processing without loosing the power and benefits of other service capabilities. The sub-fabric mechanism allows integration of heterogeneous infrastructure resources, existing applications and emerging services.

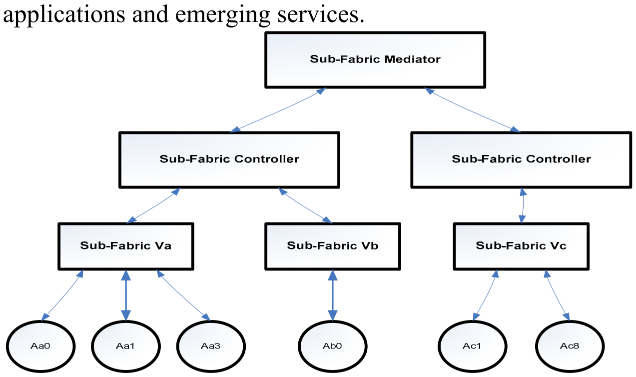

Figure 5. Sub-fabric Organization

The fabric controller and mediator services are responsible for intra- and inter-fabric (sub-fabric) interaction control respectively (Figure 5). These virtual and sub-fabric concepts draw heavily from *virtualization*, including virtual networks and virtual environments that are analogous to separated and protected instances. For example, the sub-fabrics are akin to the sub-networks of VLAN (see Figure 6). The LAN switch (L2 layer) controls the message flows between the hosts on a sub-network under its control, while the Router (L3 layer) controls the message flows between different sub-networks.

The function of the controller service is to:

- Manage interactions within the services of a sub-fabric;
- Restrict certain interactions to services within the sub-fabric scope;
- Provide a level of scope visibility (for example, names, messages, broadcast messages);
- Control and monitor service mobility;
- Manage aggregation/filtering of certain type of faults and alerts, and manage the interaction with mediator services.

The function of the mediator services is to:

- manage and control inter-sub-fabric interaction
- monitor and manage the underlying sub-fabrics
- manage protocols, including protocol conversion.

The same service agent can implement both the mediator and controller services.

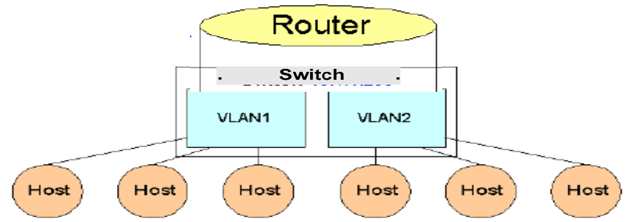

Figure 6. Virtual Local Area Network (VLAN)

These virtual and sub-fabrics provide “complete” end-to-end isolation, with end-to-end operability and manageability, through several levels (application to infrastructure) of virtual environments. The virtual and sub-fabrics improve, for example, security and access control, and allow for the implementation of two nearly identical virtual fabrics with different Quality of Service, security and other capabilities. They also restrict visibility of proprietary services, agents and resources.

IV. Conclusion

An extended enterprise is the result of links between the enterprise and its partners who themselves are connected to their partners. To ease business operations, the computing infrastructures are networked together creating a private cloud. The migration of applications to public and private clouds, and the interconnectivity of public and private clouds, with private computing infrastructures results in a highly complex environment. The deployment and manageability of applications and services in such an environment becomes unmanageable. This paper presented concepts and mechanisms that reduce the complexity of such environments and the applications that they support. In addition, the mechanisms improve the operability and manageability, including security, of the entire environment or fabric. In addition, the Business Fabrics offer dynamic allocation and assembly of services using other services, middleware and infrastructure services, and knowledge-assisted mapping of services to other services and infrastructure requirements and capabilities.

References

[1] The American Heritage® Dictionary of the English Language, Fourth Edition. © 2006 by Houghton Mifflin Company.

[2] “Management and Service-aware Networking Architectures for Future Internet”: Position Paper, Future Internet Assembly, Editors: Alex Galis (UCL), Marcus Brunner (NEC), Henrik Abramowicz (ERICSSON); December 2008.

[3] P. Goyal, R. Mikkilineni, M. Ganti, “Manageability and Operability of the Virtual Business Services Fabric,” in Proceedings of WETICE 2009: 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (accepted; to be published).

[4] Web Services Architecture, Editors D. Booth, F. McCabe, E. Newcomer, et. al., W3C Note, 11 February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/. Latest version available at http://www.w3.org/TR/ws-arch/.

[5] P. Wayner, “Cloud versus Cloud: A guided tour of Amazon, Google, AppNexus and GoGrid,” *Infoworld*, July 21, 2008.