The SOA Magazine

Feature Article

Understanding Cloud Computing and Cloud-Based Security

by David Chou

Published: March 9, 2010 (SOA Magazine Issue XXXVII: March 2010)

Download PDF

Digg This • De.licio.us • Slashdot • Technorati • StumbleUpon • Google Bookmark

Abstract: This article provides a concise introduction to cloud computing, its relationship with SOA, and further highlights common security considerations for cloud-based services. The article is comprised of excerpts from the upcoming book "SOA with .NET & Windows Azure: Realizing Service-Orientation with the Microsoft Platform"
[REF-1].

Introduction: What is Cloud Computing?

Cloud computing enables the delivery of scalable and available capabilities by leveraging dynamic and on-demand infrastructure. By leveraging these modern service technology advances and various pervasive Internet technologies, the "cloud" represents an abstraction of services and resources, such that the underlying complexities of the technical implementations are encapsulated and transparent from users and consumer programs interacting with the cloud.

At the most fundamental level, cloud computing impacts two aspects of how people interact with technologies today:

- how services are consumed
- how services are delivered

Although cloud computing was originally, and still is often associated with Web-based applications that can be accessed by end-users via various devices, it is also very much about applications and services themselves being consumers of cloud-based services. This fundamental change is a result of the transformation brought about by the adoption of service-oriented architecture (SOA) and Web-based industry standards, allowing for service-oriented and Web-based resources to become universally accessible on the Internet as on-demand services.

One example has been an approach whereby programmatic access to popular functions on Web properties is provided by simplifying efforts at integrating public-facing services and resource-based interactions, often via RESTful interfaces. Architectural views, such as this, assisted in establishing the Web-as-a-platform concept, and helped shed light on the increasing inter-connected potential of the Web as a massive collection (or cloud) of ready-to-use and always-available capabilities.

This view can fundamentally change the way we work with services, as we reuse not only someone else's code and data, but also their infrastructure resources, and further can leverage them as part of our own service implementations. We do not need to understand the inner workings and technical details of these services; Service Abstraction, as a principle, is applied to its fullest extent by hiding implementation details behind clouds.

With regards to service delivery, we are focused on the actual design, development, and implementation of cloud-based services. Let's begin by establishing high-level characteristics that a cloud computing environment can include:

- generally accessible
- always available and highly reliable
- elastic and scalable
- abstract and modular resources
- service-oriented
- self-service management and simplified provisioning

Fundamental topics regarding service delivery pertain to the cloud deployment model used to provide the hosting environment and the service delivery model that represents the functional nature of a given cloud-based service. The next two sections explore these two types of models.

Cloud Deployment Models

There are three primary cloud deployment models. Each can exhibit the previously listed characteristics; their differences lie primarily in the scope and access of published cloud services, as they are made available to service consumers.

Let's briefly discuss these deployment models individually.

Public Cloud

Also known as external cloud or multi-tenant cloud, this model essentially represents a cloud environment that is openly accessible. It generally provides an IT infrastructure in a third-party physical data center that can be utilized to deliver services without having to be concerned with the underlying technical complexities.

Essential characteristics of a public cloud typically include:

- homogeneous infrastructure
- common policies
- shared resources and multi-tenant
- leased or rented infrastructure; operational expenditure cost model
- economies of scale

Note that public clouds can host individual services or collections of services, allow for the deployment of service compositions and even entire service inventories.

Private Cloud

Also referred to as internal cloud or on-premise cloud, a private cloud intentionally limits access to its resources to service consumers that belong to the same organization that owns the cloud. In other words, the infrastructure that is managed and operated for one organization only, primarily to maintain a consistent level of control over security, privacy, and governance.

Essential characteristics of a private cloud typically include:

- heterogeneous infrastructure
- customized and tailored policies
- dedicated resources

- in-house infrastructure (capital expenditure cost model)
- end-to-end control

Community Cloud

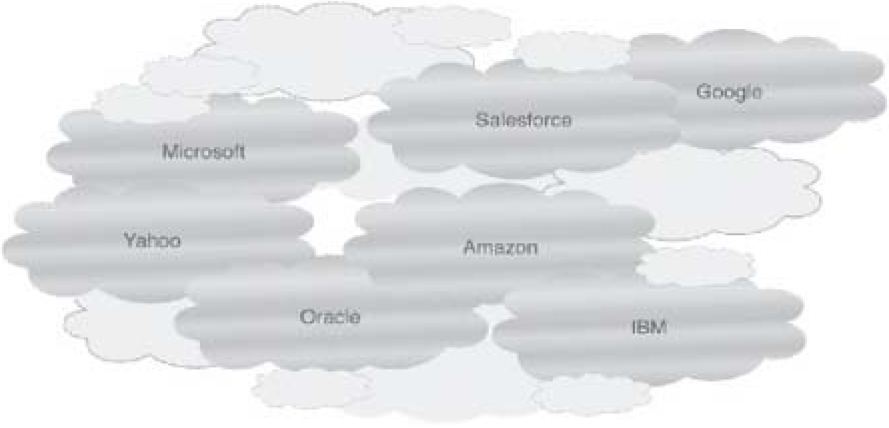
This deployment model typically refers to special-purpose cloud computing environments shared and managed by a number of related organizations participating in a common domain or vertical market.

Other Deployment Models

There are variations of the previously discussed deployment models that are also worth noting. The hybrid cloud, for example, refers to a model comprised of both private and public cloud environments. The dedicated cloud (also known as the hosted cloud or virtual private cloud) represents cloud computing environments hosted and managed off-premise or in public cloud environments, but dedicated resources are provisioned solely for an organization's private use.

The Intercloud (Cloud of Clouds)

The intercloud is not as much a deployment model as it is a concept based on the aggregation of deployed clouds. Just like the Internet, which is a network of networks; intercloud refers to an inter-connected global cloud of clouds. Also like the World Wide Web, intercloud represents a massive collection of services that organizations can explore and consume.



From a service consumer's perspective, we can look at the intercloud as an on-demand SOA environment where useful services managed by other organizations can be leveraged and composed. In other words, services that are outside of an organization's own boundaries and operated and managed by others can become a part of the aggregate portfolio of services of those same organizations.

Cloud Service Delivery Models

Many different types of services can be delivered in the various cloud deployment environments. Essentially, any IT resource or function can eventually be made available as a service. Although cloud-based ecosystems allow for a wide range of service delivery models, three have become most prominent:

Infrastructure-as-a-Service (IaaS)

This service delivery model represents a modern form of utility computing and outsourced managed hosting. IaaS environments manage and provision fundamental computing resources (networking, storage, virtualized servers, etc.). This allows consumers to deploy and manage assets on leased or rented server instances, while the service providers own and govern the underlying infrastructure.

Platform-as-a-Service (PaaS)

The PaaS model refers to an environment that provisions application platform resources to enable direct deployment of application-level assets (code, data, configurations, policies, etc.). This type of service generally operates at a higher abstraction level so that users manage and control the assets they deploy into these environments. With this arrangement, service providers maintain and govern the application environments, server instances, as well as the underlying infrastructure.

Software-as-a-Service (SaaS)

Hosted software applications or multi-tenant application services that end-users consume directly correspond to the SaaS delivery model. Consumers typically only have control over how they use the cloud-based service, while service providers maintain and govern the software, data, and underlying infrastructure.

Other Delivery Models

Cloud computing is not limited to aforementioned delivery models. Security, governance, business process management, integration, complex event processing, information and data repository processing, collaborative processes - all can be exposed as services and consumed and utilized to create other services.

An Analogy

An on-premise infrastructure is like having your own car. You have complete control over when and where you want to drive it, but you are also responsible for its operation and maintenance. IaaS is like using a car rental service. You still have control over when and where you want to go, but you don't need to be concerned with the vehicle's maintenance. PaaS is more comparable to public transportation. It is easier to use as you don't need to know how to operate it and it costs less. However, you don't have control over its operation, schedule, or routes.

Cloud Computing and SOA

Service-oriented architecture (SOA) is a distinct distributed technology architectural model defined by a set of concrete characteristics. Service-orientation is a distinct paradigm comprised of a set of design principles that, when applied to a meaningful extent, shape software programs into units of service-oriented solution logic called services. A primary emphasis of service-orientation is to deliver services as inherently flexibly and interoperable software programs that can be readily and repeatedly aggregated into a variety of complex service compositions.

Service-orientation can be viewed as a method of achieving a specific target state that is represented by a set of strategic goals and benefits associated with service-oriented computing and further supported by establishing service-oriented technology architecture. Service-oriented computing, as a distinct form of distributed computing, encompasses SOA and service-orientation. However, the "SOA" acronym has been historically used to brand what is represented by service-oriented computing.

Cloud computing technology advances can be leveraged in support of applying service-orientation and building service-oriented technology architecture. Where appropriate, cloud computing deployment models and cloud service delivery models can be utilized for specific services or entire service compositions. This provides the option for services to be deployed within and accessed via cloud environments where they may benefit from increased scalability and reliability provided by the cloud environments underlying infrastructure and resources.

Cloud computing models can extend and build upon service-oriented architectural models. However, it is important to note that for a software program to be service-oriented, it does not require cloud technology - and - for a software program to be cloud-based, it does not need to have been shaped by the service-orientation design paradigm. Non-service-oriented architectural models, such as silo-based and monolithic application architectures, can also be used to create cloud-based systems.

Security Considerations

Cloud-based services and service-oriented solutions deployed on cloud platforms can typically leverage and be designed with existing security frameworks. However, the fact that some or all parts of a given service composition reside in an environment external to the controlled IT enterprise raises several additional security considerations.

Some of the most common distinct security considerations for cloud-based services and service compositions include the following:

- *data privacy* - Your data is being hosted in someone else's data center.
- *shared and virtualized resources* - The service's physical infrastructure, to varying degrees between cloud platforms, may be shared among multiple tenants.
- *multi-tenancy* - The service hosting processes and the exchanged data are executed and managed in shared environments.
- *heterogeneity* - The service may be implemented in a cloud hosting platform that uses highly generic policies and lowest-denominator frameworks, on top of a heterogeneous infrastructure.
- *Internet transit* - Distributed communications are mostly transmitted over public Internet protocols and transports.
- *lack of control* - Cloud platforms abstract infrastructure complexity, which can include hiding the control and administrative mechanisms necessary to meet specific security requirements.
- *lack of standards* - Cloud platforms are mostly specialized implementations. Standards-based data exchange is often supported, but management, governance, and security controls are often abstracted into the implementation and not standardized.

When these types of security issues surface they need to be addressed early on to ensure that services and data remain functional and protected, especially within hybrid architectures that require regular communication within a service composition comprised of services that reside inside and outside of enterprise boundaries.

Cross-Domain Access Control

Cloud computing adds to the service-oriented technology architectural model the dimension of managing access control over services deployed across highly distributed environments. This can be a hybrid cloud model for one organization having assets in both internal and external environments, but it can encompass multi-enterprise collaboration scenarios across different security domains.

Cross-domain access control is about authorization, not authentication. This means that even when spanning domain boundaries, we can reuse existing user authentication and identity management systems, and then extend them to accommodate cloud-based services access requirements.

Hybrid Cloud Security

Hybrid clouds represent the most relevant model when trying to extend an on-premise or internal service-oriented solution to a cloud platform. In general, this points to scenarios where an organization has assets deployed in both on-premise and external cloud environments. The challenge is, now that there are assets deployed outside of the organization's own network environment and security domain, how can we provide secured, single sign-on access to those cloud-based services?

The key is in understanding how we can leverage already centralized security and identity infrastructure without replicating it in another environment and without having distributed services access it over the public Internet.

Traditional practices generally advocate replicating and caching identity directories in different environments across which we want to enable cross-domain interoperability. Such is the case especially when a single IT enterprise has two or more established domain service inventories. However, when exploring this approach with cloud-based services, it can be error-prone, complex and expensive to implement. It would further have to deal with policy distribution, systems management, and auditing. As a result, it may be more practical and effective to try and establish a VPN connection with the cloud provider. But even that option can raise a whole other set of issues.

Ideally, an external cloud environment in a hybrid cloud model would represent a separate security domain. Considerations for access control for cloud-based services would then center on approaches to bridge the separate security domains.

Inter-Organization Service Composition Security

Facilitating service compositions across organizational boundaries is nothing new. It has been a consistent focus area in various past service-oriented architecture implementations, as well as B2B integration architectures and multi-enterprise collaboration frameworks. However, similar to cloud computing, managing access control over automated systems and processes has been challenging.

For example, most implementations today use highly specialized connections (such as FTP, EDI networks, VPNs, dedicated circuits, Web services endpoints exposed in the network perimeter, etc.). These point-to-point models can be brittle in that they suffer from the classic "one-off" syndrome that has plagued silo-based application development and integration architectures. These problems are further compounded when we create individual single-purpose channels for individual partner organizations (or even individual applications with larger partner organizations) and then need to further assume the responsibility of managing identities and their lifecycles.

Different options exist for addressing these issues, including the use of generic or system accounts shared between services and/or individual users. However, ultimately, the considerations in this area also boil down to bridging separate security domains.

External Identity Providers

Online digital identity providers, such as cloud-based authentication services (Windows Live ID, Google Account, Yahoo ID, OpenID, etc.) can be used with various consumer-facing architectures. These implementations evolved from organizations providing their own consumer identity management systems for other organizations to use as a service.

Externalizing consumer-centric membership and identity management systems enables organizations to reuse already provided systems instead of investing in and running their own. It further helps consumers to reduce the number of digital identities they have to manage. However, external identity providers also represent separate security domains, and the same considerations will need to be applied when implementing access control to services intended for these identities to access.

References

[REF-1] "SOA with .NET & Azure", Chou, deVadoss, Erl, Gandhi, Hogan, King, Kommalapati, Loesgen, Schittko, Wilhelmsen, Williams, Prentice Hall, 2010.

Home Past Issues Contributors What is SOA? SOA Glossary SOA School SOA Books About/RSS
Legal