

Una perspectiva de cloud computing con énfasis en seguridad y arquitectura

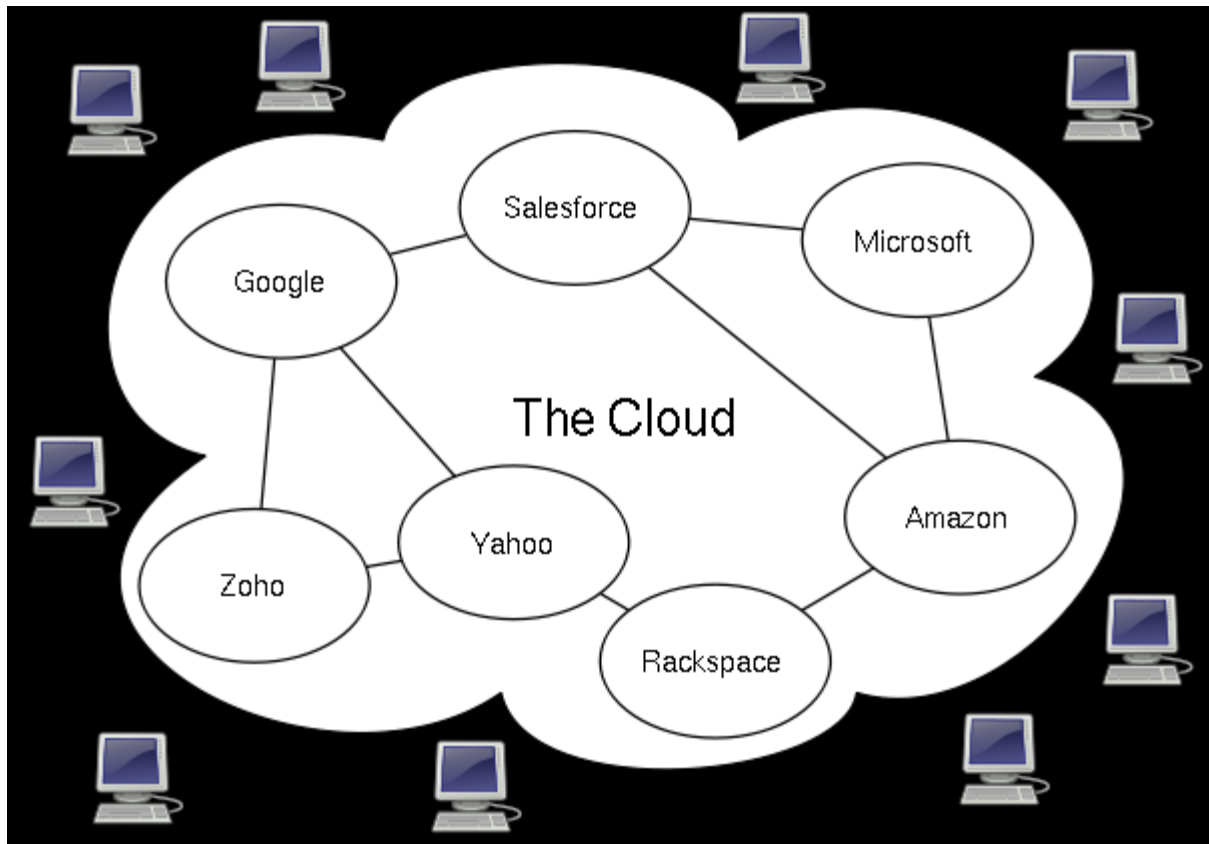
Eduardo B. Fernandez
Florida Atlantic University
Boca Raton, FL, USA

Objetivos

- Se expondrá una visión de ciertos aspectos claves de cloud computing; en particular, su arquitectura general y seguridad. La arquitectura es importante para ver que se puede hacer con los clouds. La seguridad limita lo que se puede hacer en ciertos casos. Se explicará el concepto de modelar arquitecturas, ataques, y defensas usando patrones de software.

The cloud is based on two old ideas

- *Utility computing*—an infrastructure provides a set of resources to be shared by applications which use them as needed and pay only for what they use. This approach lets applications have access to a variety of almost unlimited resources (at the cost of giving up parts of the control about where and how data is processed).
- *Virtual machines*—this is a concept used for supporting the execution of operating systems sharing the same hardware. The virtual machine seen by a given user is created from one or more servers in the cloud and has apparently unlimited power and storage capacity.



Cloud services

- A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access)
- Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

User view

- The illusion of infinite computing resources available on demand. Cloud providers allow users to increase/decrease resources depending on their needs.
- The reduction of an up-front commitment by users. Minimize up-front costs associated with on-premise hardware and software license.
- The ability to pay only for use of computing resources as needs. Clouds offer pay-as-you-go services to the public where users only pay for the services consumed.

Types of services

- *Software as a Service (SaaS)*, An infrastructure that runs customer services and provides those services for the customer itself or for its own customers
- *Platform as a Service (PaaS)*, A platform where a customer can run his applications in a configured execution infrastructure.
- *Infrastructure as a Service (IaaS)*. A hardware service where customer applications can be executed.



Dept 1 App

Dept 2 App

Dept 3 App

Built by each department

Shared Components Self-Service Interface Built by IT

SOA BPM UI Identity Mgt System Mgt Middleware

Application Server

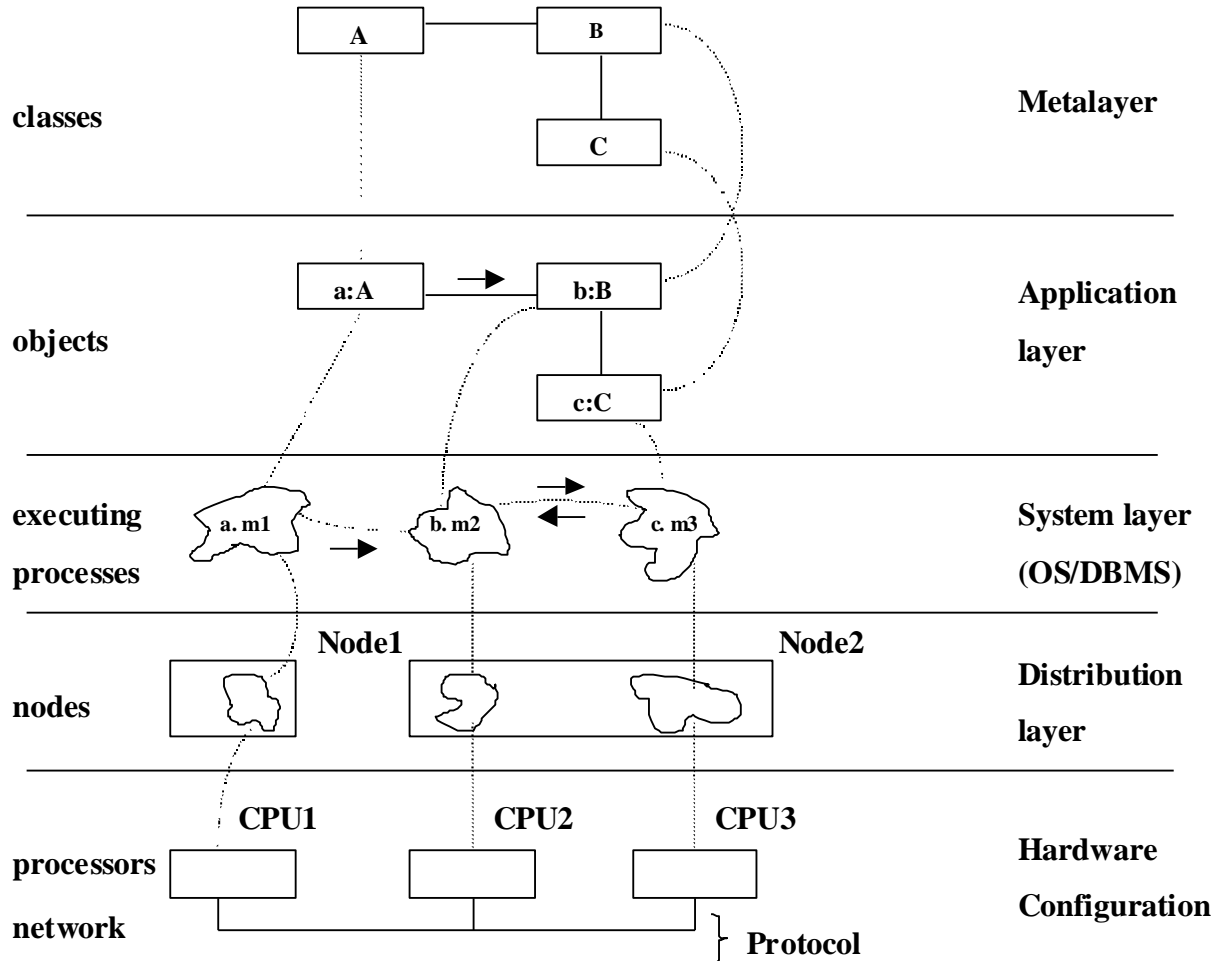
OS, Virtualization

Provided by IT

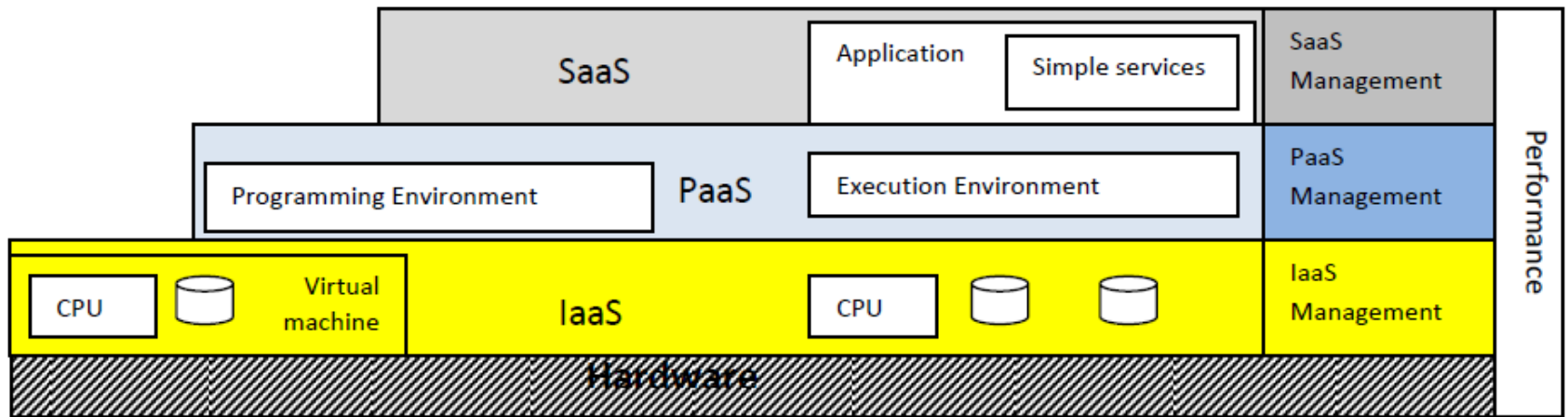


DB Integration Legacy

Architectural levels



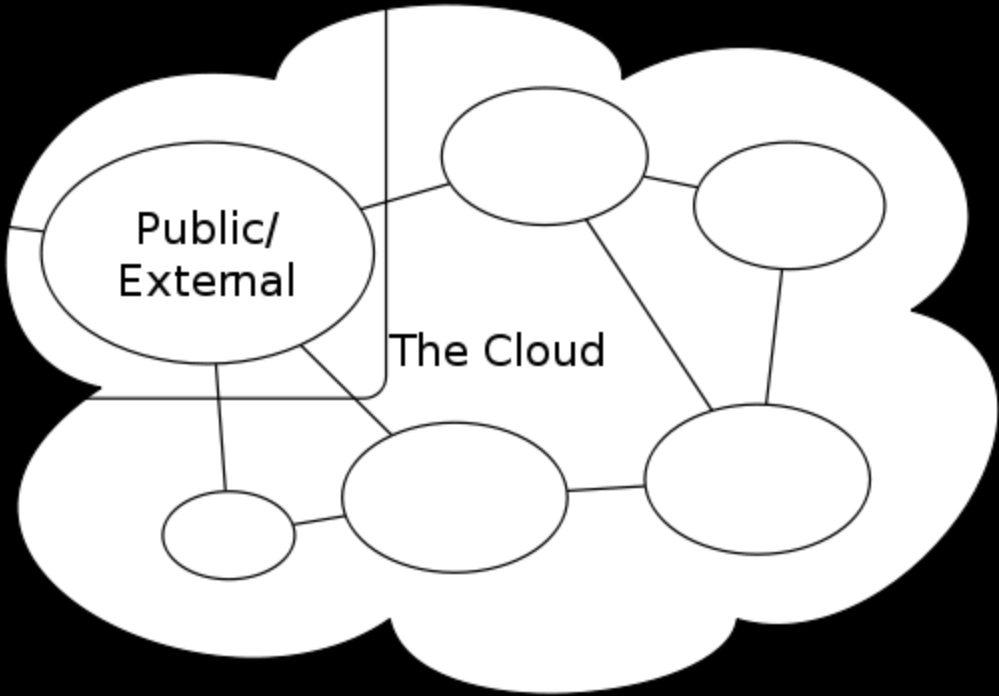
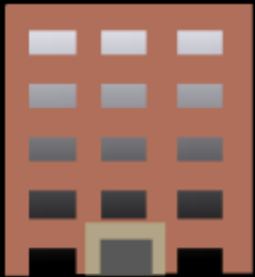
Cloud layers



Deployment

- Public – public clouds are run by third party providers. They provide services to any client who pays for their services. These services can be accessed across the Internet or a private network. Public clouds are preferable for non-critical applications that are related to few core processes of the organization.
- Private – private clouds are built for the exclusive use of an institution, providing strong control over data, security, and quality of service. Private clouds can be built and managed by the organization or by a cloud provider. Their use is significantly more expensive than public clouds.
- Hybrid – combines the two previous approaches. Hybrid clouds are environments in which an organization manages some resources in-house and uses external clouds for other services.
- Community – community clouds might be established where different organizations that share similar requirements want to share the infrastructure. Community clouds can be deployed using any of the three models mentioned before.

Private/
Internal



Applications

- Medical health record sharing – provides storage of personal health-care information online such as Microsoft Health Vault [Mic] and Google Health [Goo].
- Web-based email – it is accessed through a web browser, and it uses the cloud for processing and data storage. Users can access their webmail from anywhere. For example, Yahoo and Gmail provide web-based email service.
- Social Networks – provides a platform to facilitate communication and sharing between users to post profile, create collaboration groups, and share contents. For example Facebook users can build scalable Cloud based applications hosted by Amazon Web Services [Ama].
- Data Analytics—Analysis of sales, data mining, statistical analysis of customer preferences.
- End-user applications – these applications are delivered on a PC such as word processing, databases, spreadsheets, media editing, presentations, collaboration, and many others. For instance, Google Docs [Wika] offers web-based applications that allow users to create and edit documents online.

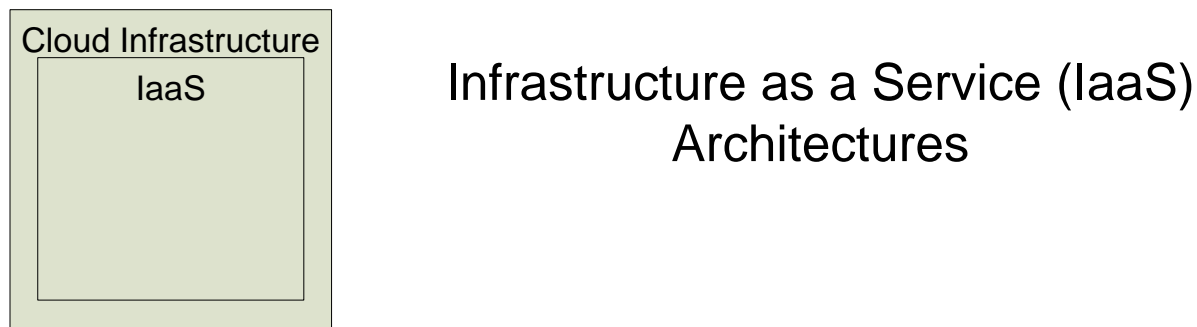
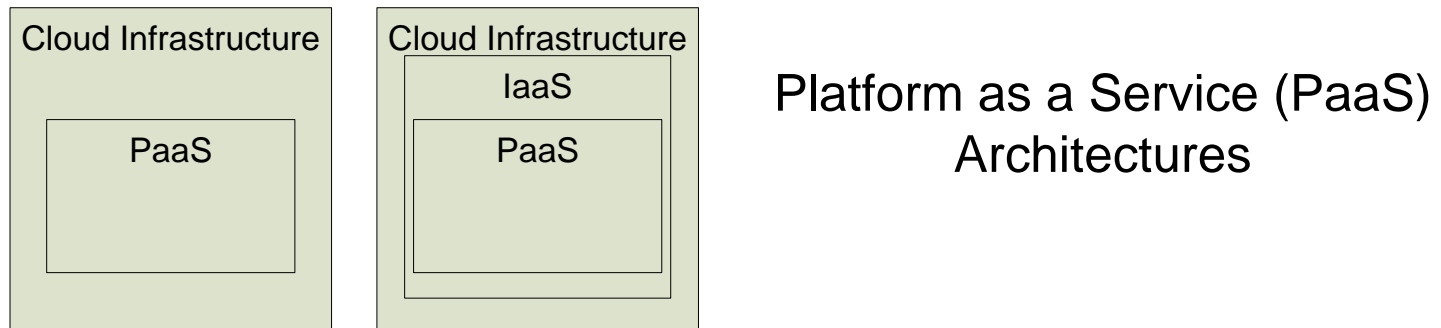
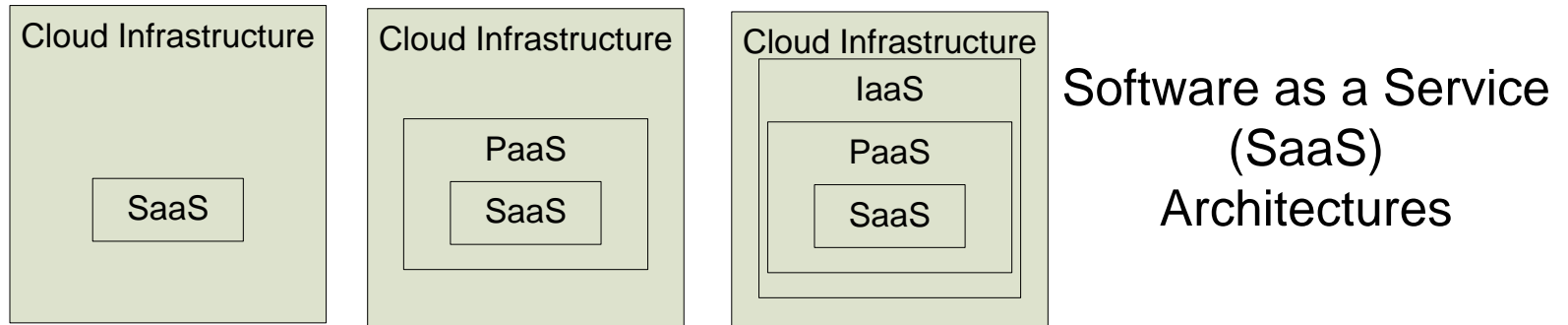
Possible uses

- Gaming, extended to personalized medication, product development,...
- Repair information for field technicians
- Comparison shopping and make sure you find a product
- Suggest activities, movies,...
- Cheap computational power for planning, stock market selection,...

Current Market

- From “The Economist”
- SaaS: \$11.7 B
- PaaS: \$311 M
- IaaS: over \$1 B
- Amazon has 80-90% of market, 90,000 VMs created every day
- Overall: \$56 B by 2020

Service Model Architectures



Service Level Agreement

- SLA is a service contract that defines the relationship between the provider and client, and it determines the benefits and responsibilities of each party
- The only means the provider can gain trust of client is through the SLA
- Due to the dynamic nature of the cloud, continuous monitoring on Quality of Service (QoS) attributes is necessary to enforce SLAs [Pat09]
- IBM proposed a framework for specifying and monitoring SLA for web services named Web Service Level Agreement (WSLA). Although WSLA [Lud03] was designed for web services, it can be applicable to Cloud Computing as well.

WSLA

- Parties: It is composed of two parties: supporting parties and signatory parties. Signatory parties are the service provider and the consumer. Supporting parties are the third parties that act in behalf of the service provider or the consumer.
- Service Definitions: describe the services to be delivered in terms of operations and service's metrics.
- Obligations: define the service level that is guaranteed with respect to the metrics defined in service definitions.

Service-Oriented Architecture (SOA)

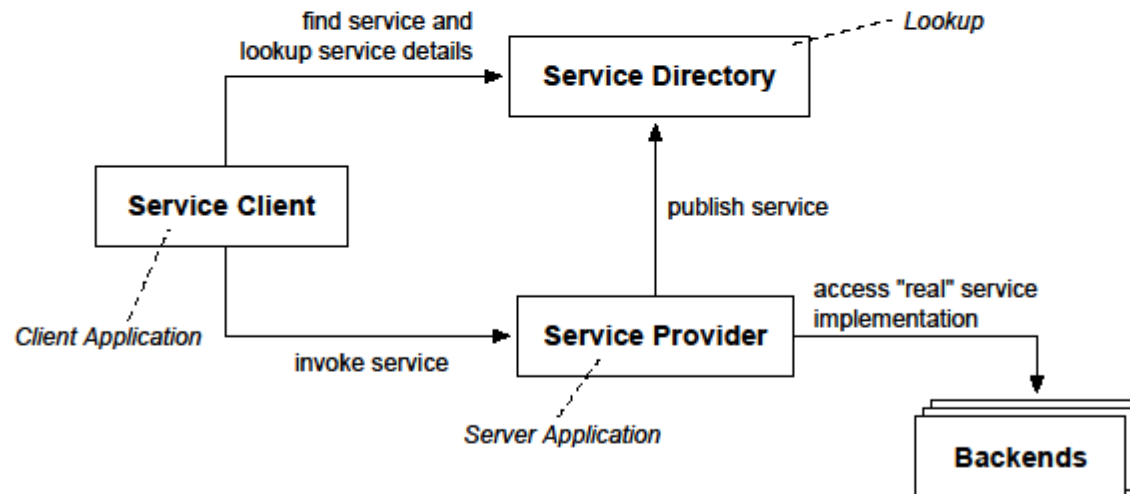
- SOA is an architectural style in which a system is composed from a set of loosely coupled services that interact with each other by sending messages
- In order to interoperate, each service publishes its description, which defines its interface and expresses constraints and policies that must be respected in order to interact with it.
- A service (set of services) is thus a building block for service-oriented applications
- Applications are built by coordinating and assembling services
- A service is a logical representation of a business activity that has a specified outcome
- A key principle about services is that they should be easily reusable and discoverable, even in an inter-organizational context
- Furthermore, the channels of communication between the participating entities in a service-oriented application are much more vulnerable than in operating systems or within the boundaries of an organization's intranet, since they are established on public networks

SOA properties

- SOA has been adopted as a main direction by IBM and Microsoft among others
- It is a high-level architectural approach that attempts to glue and unify disjoint pieces of software
- SOA can be implemented using ad hoc architectures, CORBA, Jini, web services, clouds, or other distributed architectures, now in clouds
- A mail service is a basic type of SOA (with only one service)
- SOA can be seen as the generalization of client-server architectures.

Basic SOA concept

- A service offers a remote interface with a well-defined INTERFACE DESCRIPTION
- The interface description contains all interface details about the service
- The service advertises itself at a central service, the LOOKUP service



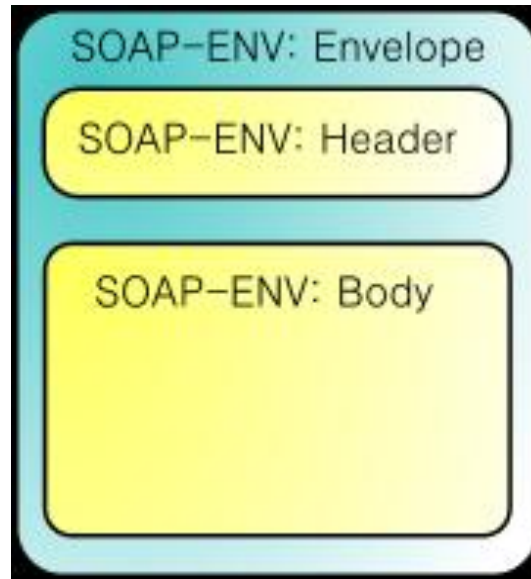
Web Services

- *A Web Service is a type of component that is available on the web and can be incorporated in applications or used as a standalone service*
- *Requires a standard supporting framework*
- *The web could become a marketplace of web services (not there yet)*

SOAP

- Originally defined as **Simple Object Access Protocol**, is a protocol specification for exchanging structured information between web services
- It relies on [Extensible Markup Language](#) (XML) for its message format, and usually relies on other [Application Layer](#) protocols, most notably [Remote Procedure Call](#) (RPC) and [Hypertext Transfer Protocol](#) (HTTP), for message negotiation and transmission
- SOAP can form the foundation layer of a [web services protocol stack](#), providing a basic messaging framework upon which web services can be built. This XML based protocol consists of three parts: an envelope, which defines what is in the message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing procedure calls and responses.

SOAP



REST

- **Representational State Transfer (REST)** is a style of [software architecture](#) for [distributed hypermedia](#) systems
- REST exemplifies how the Web's architecture emerged by characterizing and constraining the macro-interactions of the four components of the Web, namely [origin servers](#), [gateways](#), [proxies](#) and [clients](#), without imposing limitations on the individual participants.
- REST-style architectures consist of [clients](#) and [servers](#). Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of representations of resources. A [resource](#) can be essentially any coherent and meaningful concept that may be addressed. A [representation](#) of a resource is typically a document that captures the current or intended state of a resource.

REST and web services

A RESTful web service (also called a RESTful [web API](#)) is a simple web service implemented using HTTP and the principles of REST. It is a collection of resources, with three defined aspects:

- the base URI for the web service, such as `http://example.com/resources/`
- the [Internet media type](#) of the data supported by the web service. This is often [JSON](#), [XML](#) or [YAML](#) but can be any other valid Internet media type.
- the set of operations supported by the web service using [HTTP methods](#) (e.g., POST, GET, PUT or DELETE).

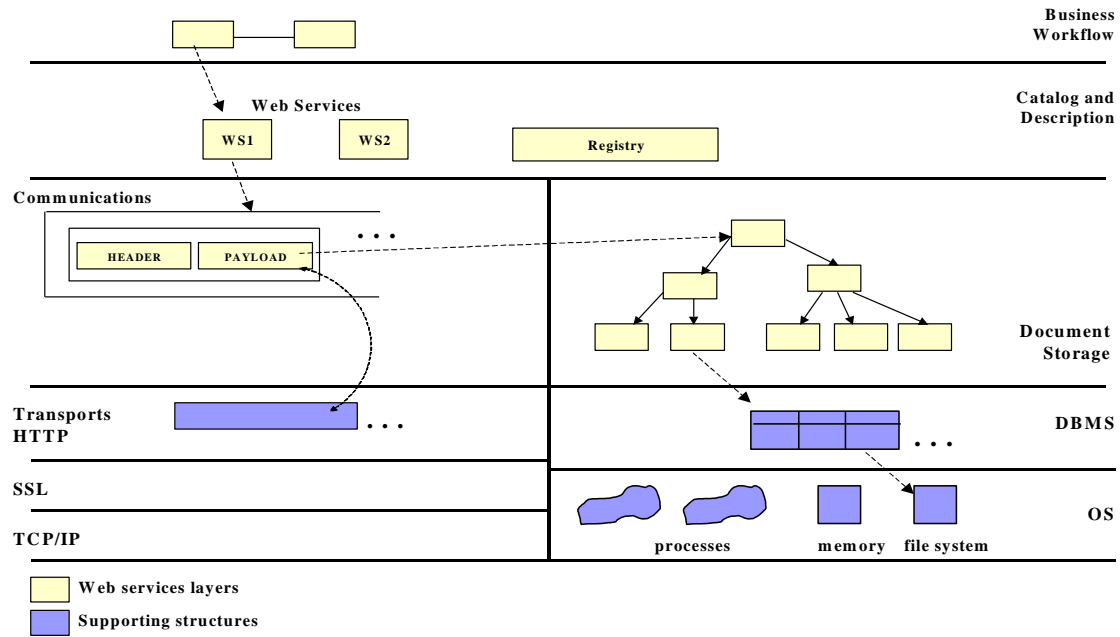
REST or SOAP?

- Most of the web functionality on the Internet now uses REST: Twitter, Yahoo's web services use REST, others include Flickr, del.icio.us, pubsub, bloglines, technorati, and several others. Both eBay and Amazon have web services for both REST and SOAP.
- SOAP is mostly used for Enterprise applications to integrate wide types and large numbers of applications. Google implements their web services using SOAP, with the exception of Blogger, which uses XML-RPC.
- REST is a low-level protocol and cannot enforce precise security

New layers

- Web services introduce three more architectural layers
- These layers run on top of communication protocols (SOAP or REST) for transmission and on top of databases and operating systems for storage and processing
- New layers add functionality but also complexity

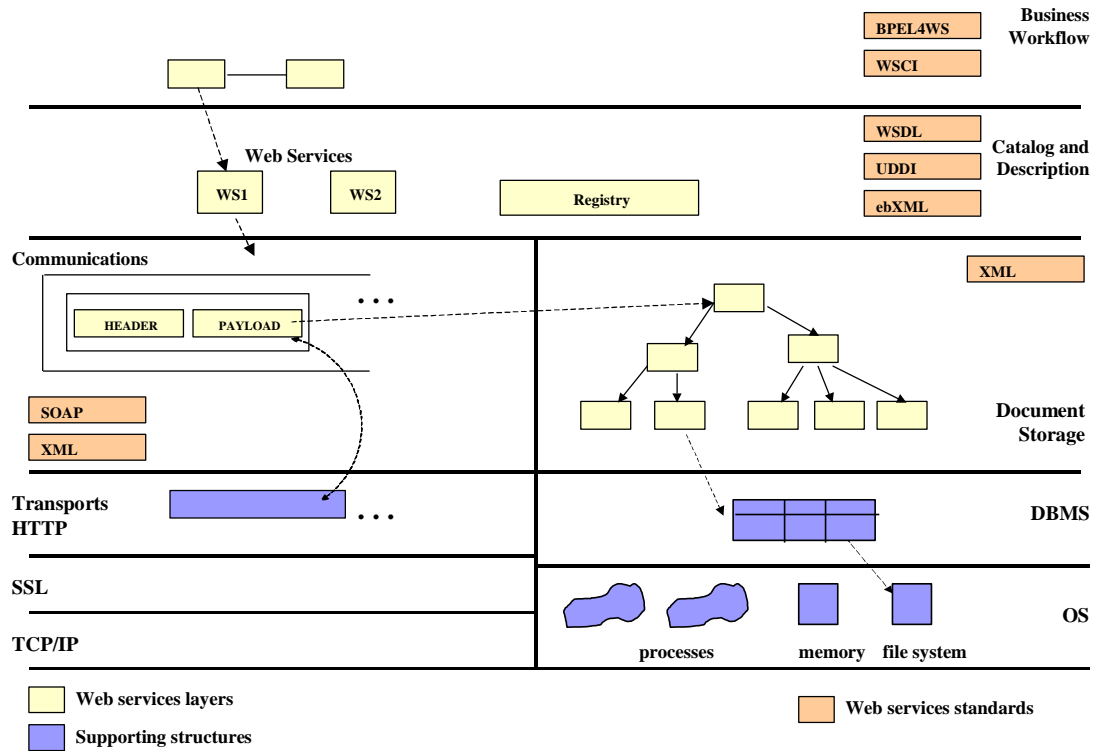
Web services layers



Functionality standards

- Each layer has a set of standards
- BPEL Business Processing Language
- WSDL and UDDI to find services
- SOAP/REST for communication
- XML to describe documents
- There are also protocols for security, reliability, trust, and policies (non-functional aspects)

Web services standards



Clouds and SOA

- Often SOA has failed because of the need to set up a large framework of business services at the same time
- Clouds offer a way to deploy incrementally, especially using SaaS. Business services can be deployed one by one
- If the appropriate cloud services are available, e.g. security and reliability, we can put important services in the cloud without the need to be secure and reliable in our own installation. SOA can also now be more scalable and more cost effective.

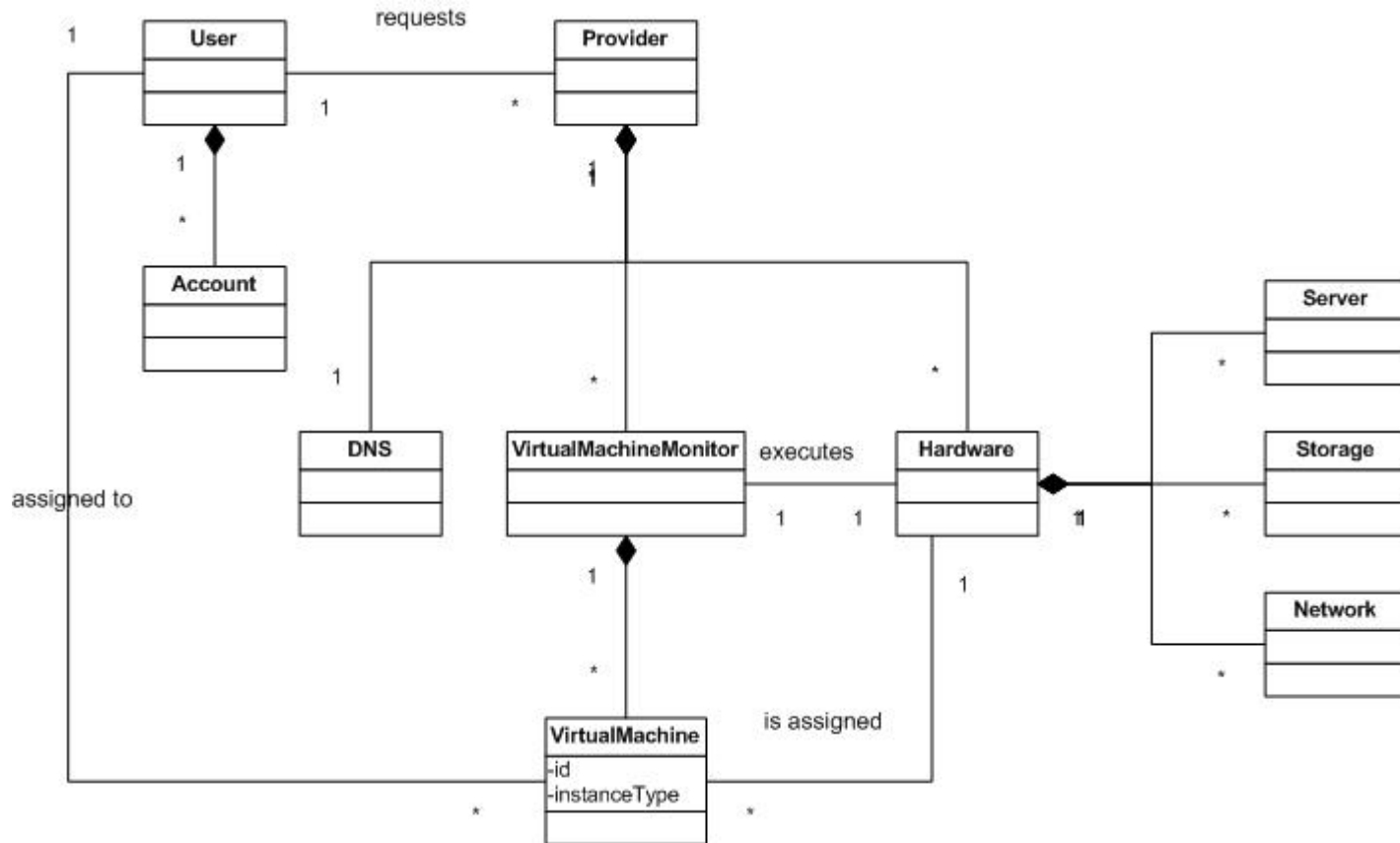
SOA and Clouds

- Clouds provide services so most of this applies to them
- Web services interoperate well because of their standards
- There are no similar standards for cloud services
- NIST is developing interim standards, starting from use cases

System architecture for clouds

- A **User** creates one or more **Accounts** in order to use the **Provider's** infrastructure. The **Provider** is composed of a Hypervisor, Hardware (server, storage and network), and **DNS** (Domain Name System). The **Virtual Machine Monitor** (VMM) creates **Virtual Machines** (VM) and assigns their instances to the users who requested them. When the instance is launched, it is assigned to a physical server and given other hardware resources. The **Virtual Machine** passes system calls to the **Virtual Machine Monitor** which executes those calls in the **Hardware**.

IaaS architecture

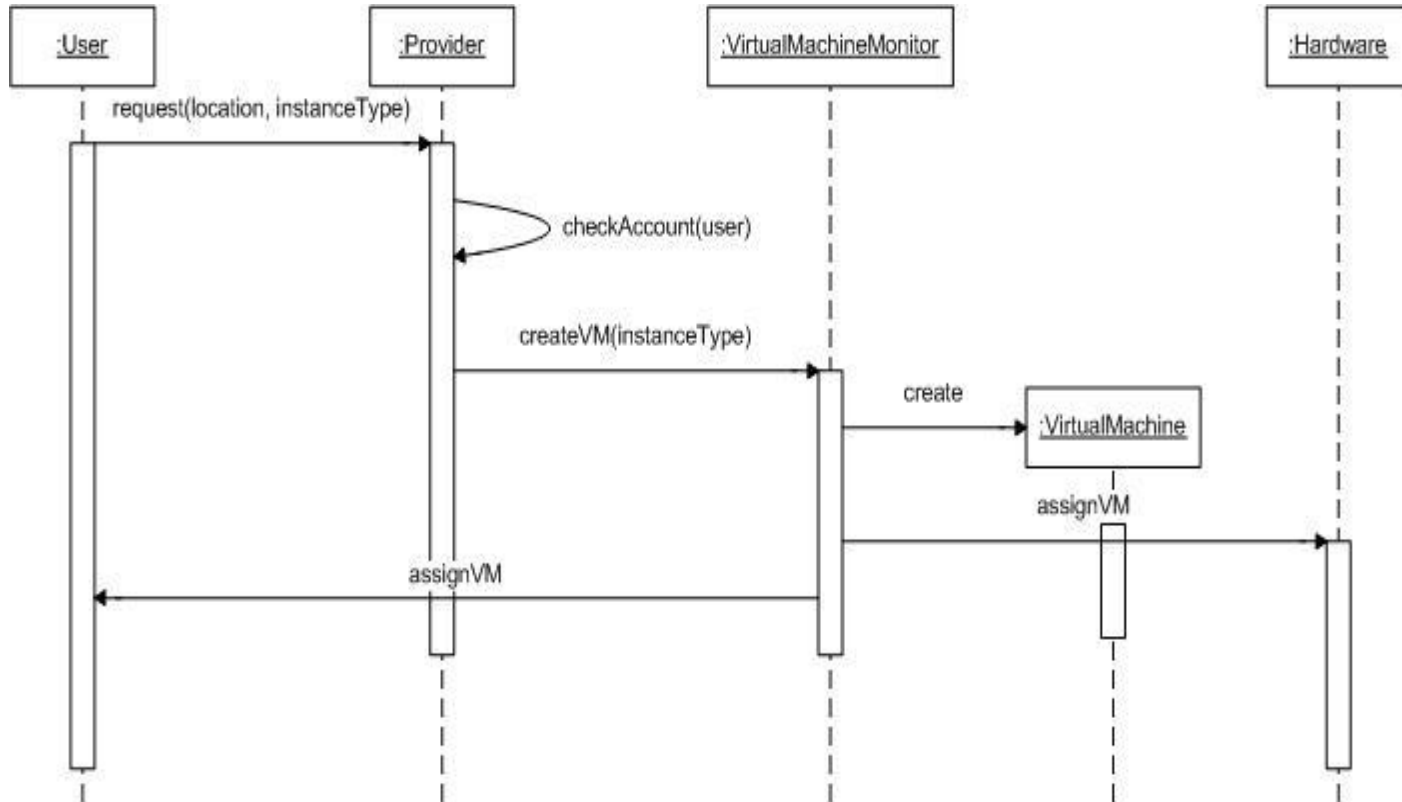


UC Create a VM

UC1: Create a Virtual Machine for a user

- Summary: The Provider creates a Virtual Machine for a user.
- Actor: User
- Precondition: The user must have an account with the Provider
- Description:
 - The User requests to the Provider to create a virtual machine. He specifies the physical location and the type of the instance.
 - The Provider checks if the user has an account and redirects the request to the Hypervisor.
 - The Virtual Machine Monitor creates an instance of the Virtual Machine and assigns it to a server and to the user.
- Postcondition: A Virtual Machine is created in the specified location and assigned to a server and to the user.

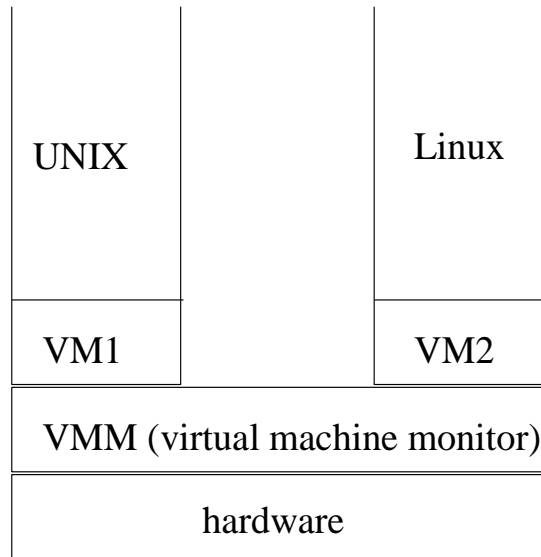
UC Create a VM



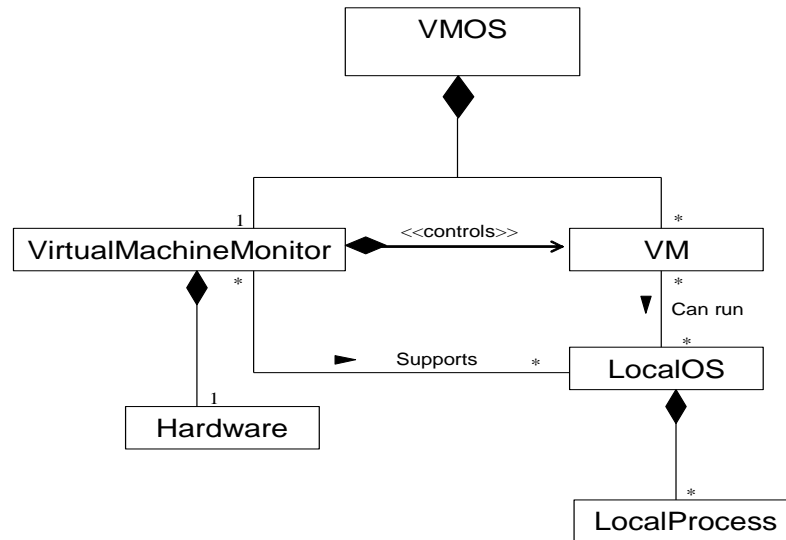
The Virtual Machine Operating System Architecture

- Provides a set of replicas of the hardware architecture (Virtual Machines), that can be used to execute (maybe different) operating systems with a strong isolation between them
- Context: Mutually suspicious sets of applications that need to execute in the same hardware. Each set requires isolation from the other sets

VM execution



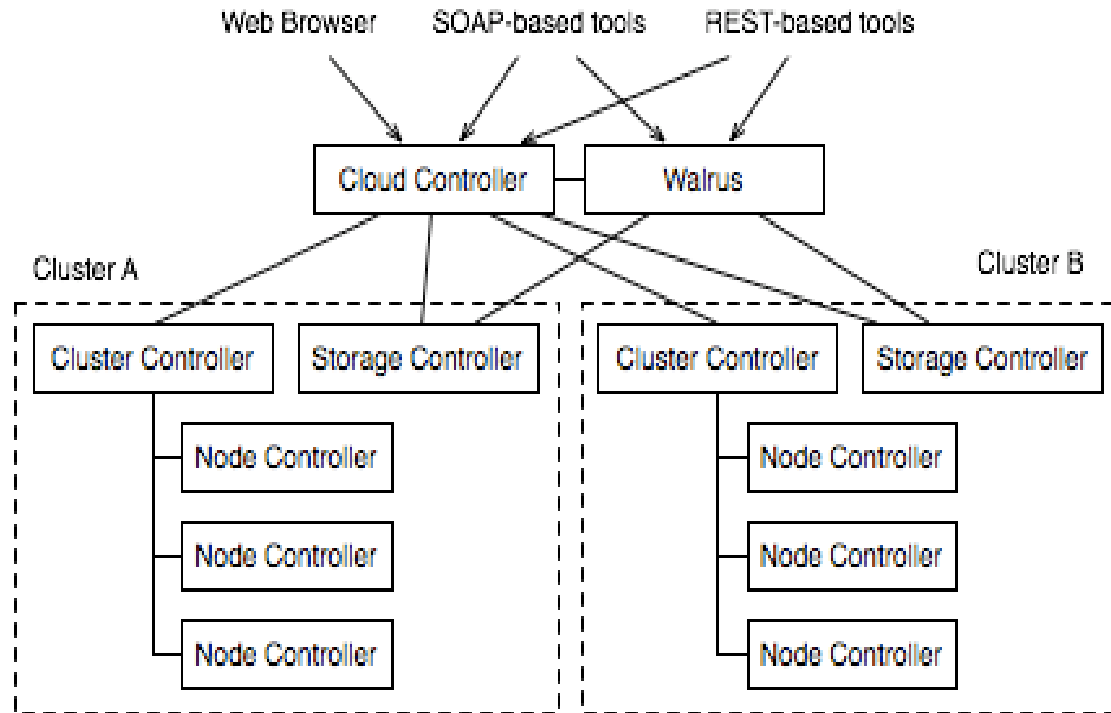
VM operating system

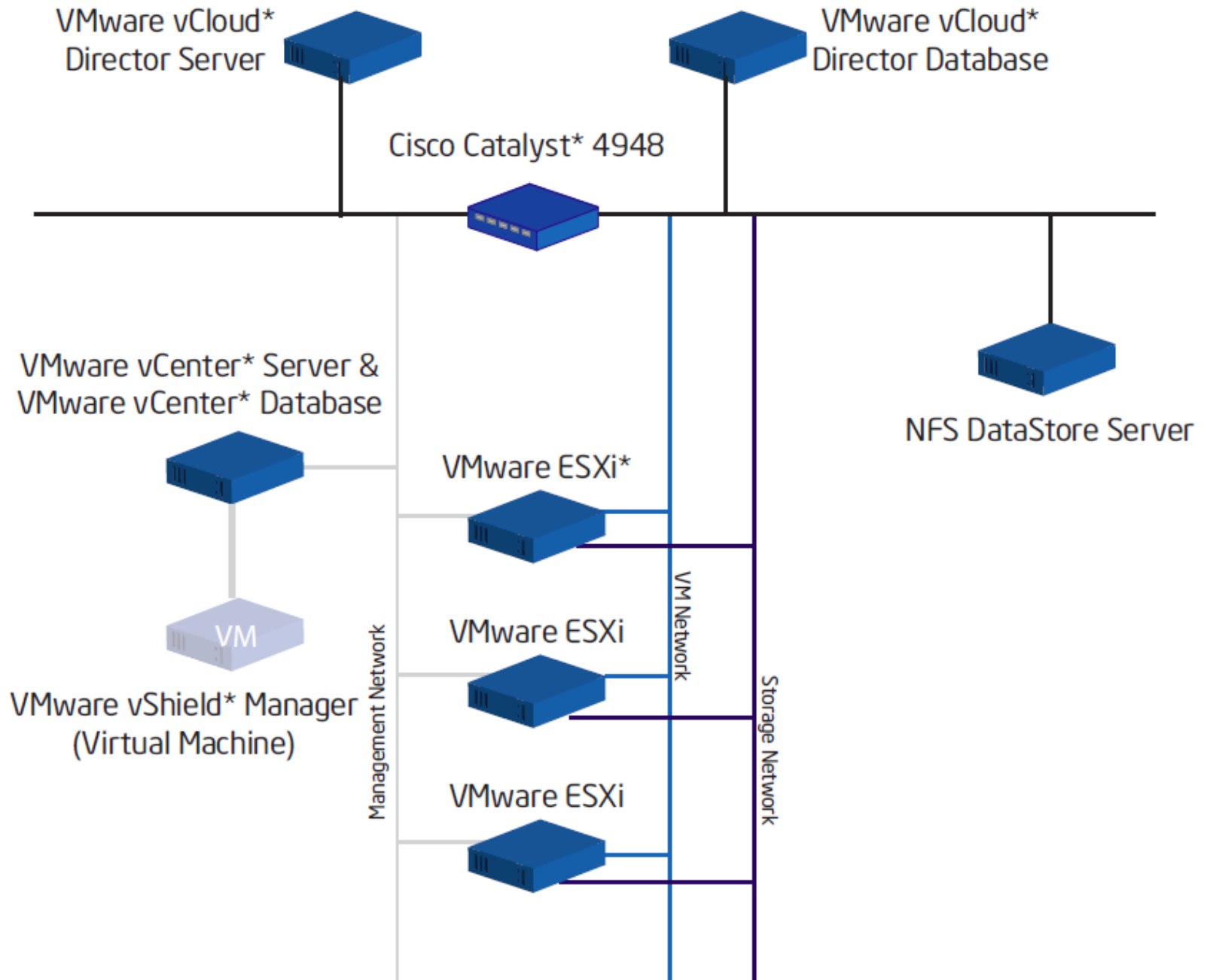


Examples

- Virtualization in portable and embedded devices
- Eucalyptus Cloud
- Amazon EC-2 Cloud
- VMWare
- Hypervisors

Eucalyptus architecture





Virtual Machine Monitors

- A VMM provides a *uniform view* of underlying hardware, making machines from different vendors with different I/O subsystems look the same, which means that virtual machines can run on any available computer.
- Administrators can view hardware simply as a pool of resources that can run arbitrary services on demand.
- Because the VMM also offers complete *encapsulation* of a virtual machine's software state, the VMM layer can map and remap virtual machines to available hardware resources at will and even migrate virtual machines across machines.
- Load balancing among a collection of machines thus becomes trivial, and there is a robust model for dealing with hardware failures or for scaling systems
- When a computer fails and must go offline or when a new machine comes online, the VMM layer can simply remap virtual machines accordingly.
- Virtual machines are also easy to replicate, which lets administrators bring new services online

VMM advantages II

- Encapsulation also means that administrators can suspend virtual machines and resume them at arbitrary times or checkpoint them and roll them back to a previous execution state.
- With this general-purpose undo capability, systems can easily recover from crashes or configuration errors.
- Encapsulation also supports a very general mobility model, since users can copy a suspended virtual machine over a network or store and transport it on removable media.

VM sprawl

- Since VMs are easy to create it may happen that many VMs may be running with little control or supervision.
- This may produce security and reliability problems.
- A systematic system administration is needed to control executing VMs

Mediation

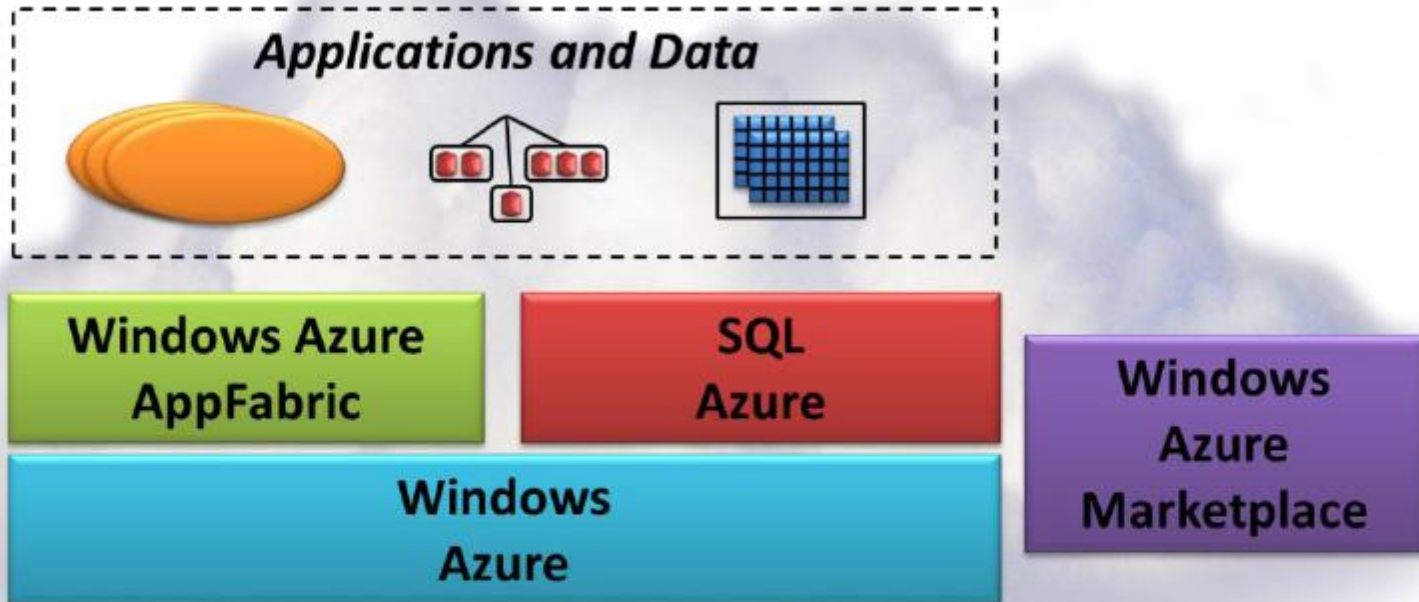
- The VMM can also provide total mediation of all interactions between the virtual machine and underlying hardware, thus allowing strong isolation between virtual machines and supporting the multiplexing of many virtual machines on a single hardware platform.

The VMM can then consolidate a collection of virtual machines with low resources onto a single computer, thereby lowering hardware cost and space requirements

Agnostic

- The cloud middleware platform should be designed to be agnostic in all three dimensional axes: frameworks, languages, and target clouds.
- In particular, we want to allow users to focus on their applications, their target levels of quality needs, their business needs, and be able to deploy these applications using state of the art best practices that can help guarantee repeatable and optimizing deployments.
- The deployments and associated management tooling should be independent of cloud providers and application platforms.

MS Azure components



SQL Azure

- SQL Azure Database provides a cloud-based database management system (DBMS). This technology lets on-premises and cloud applications store relational data on Microsoft servers in Microsoft data centers.
- SQL Azure Reporting allows creating and publishing standard SSRS reports on cloud data.
- SQL Azure Data Sync allows synchronizing data between SQL Azure Database and on-premises SQL Server databases. It can also be used to synchronize data across different SQL Azure databases in different Microsoft data centers.

NOSQL (Not Only SQL)

- DBMSs that differ from classic [relational database management systems](#) (RDBMSes) in some way
- These data stores may not require fixed [table schemas](#), and usually avoid [join](#) operations and typically [scale horizontally](#)
- Papers typically refer to these databases as **structured storage**, a term that would include classic relational databases as a subset.
- Some NoSQL advocates promote very simple interfaces such as [associative arrays](#) or key-value pairs.
- Other systems, such as native [XML databases](#), promote support of the [XQuery](#) standard. Newer systems such as CloudTPS also support [join queries](#).

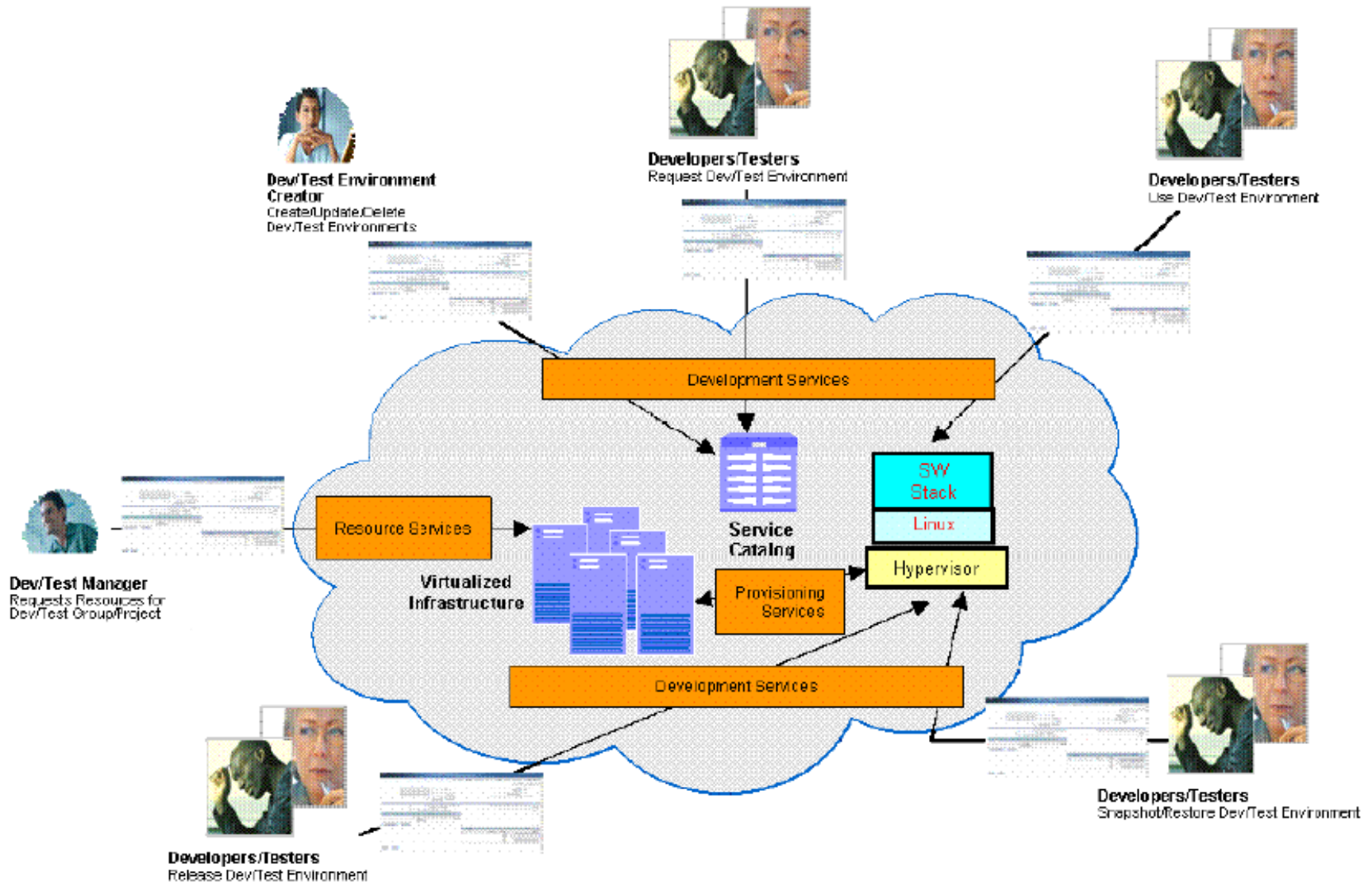
IBM Tivoli Development and Test cloud

- Mahesh Dodani (IBM), JOT, May/June 2010
- Developers/testers reserve dev/test environments from a service catalog, use and release the environments that are handled and managed from "nearby" virtualized infrastructures.
- A central site monitors the geographically dispersed cloud environments and manages the cloud environment in areas such as performance, availability, utilization, and capacity. A key objective here is to use IBM Service Management capabilities to monitor and manage the cloud environment.
- Capacity is increased by "plugging in" a virtualized infrastructure anywhere in the world.

Developer/tester use cases

- Facilitate interactions with the dev/test environments through the entire lifecycle, including the ability to manage dev/test environments, request these environments, use them, snapshot/restore the environments, and release them.
- In addition, a dev/test manager has the ability to reserve resources to cover all the development and test for the projects under their control

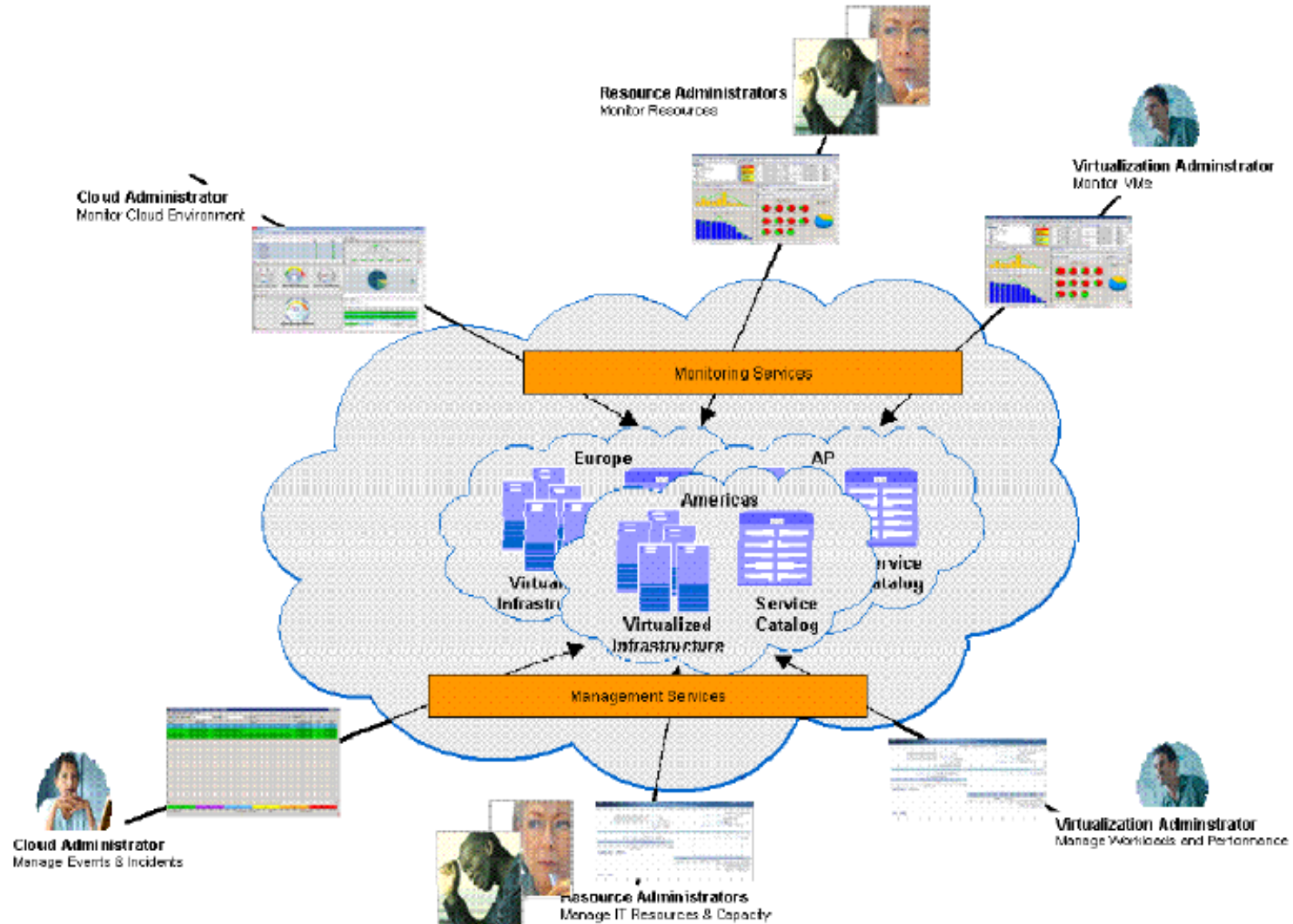
Developer/tester use cases



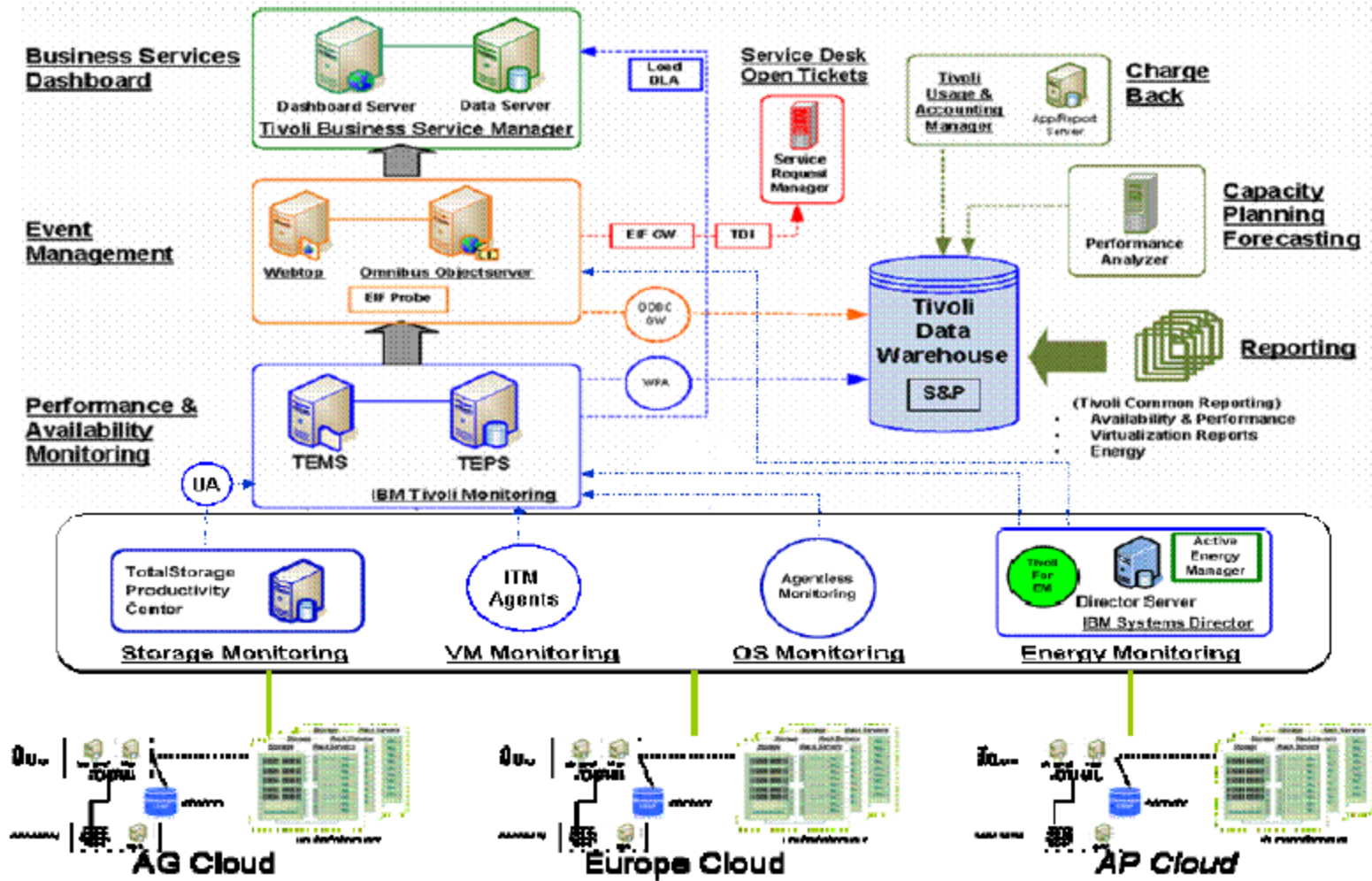
Administrator use cases

- The administrator requirements define how the cloud environment is monitored and managed from the perspective of IT resources, virtualized environments, and cloud services.
- The resource administrator is interested in monitoring different types of IT resources involved in delivering cloud services, including compute, memory and storage; and the ability to manage different aspects of the IT resources, including utilization and capacity.
- The virtualization administrator focuses on monitoring VMs, and manages workloads associated with the cloud services as well as the performance (by increasing the IT resources allocated to the VM.)
- The cloud administrator monitors the entire cloud environment, and manages any incidents and events to ensure efficient and effective delivery of the cloud services to the established SLAs

Administrator use cases



IBM Tivoli Cloud



Cost of security attacks

- An attack on Google cost about \$500,000
- Produced by the Santi worm, which infected computers in 2004-2005 to cause them to enter search queries and overwhelmed the search engines (Denial of Service attack)
- A Google engineering team studied the worm and fixed it
- Cost considers the engineering team and the lost revenue

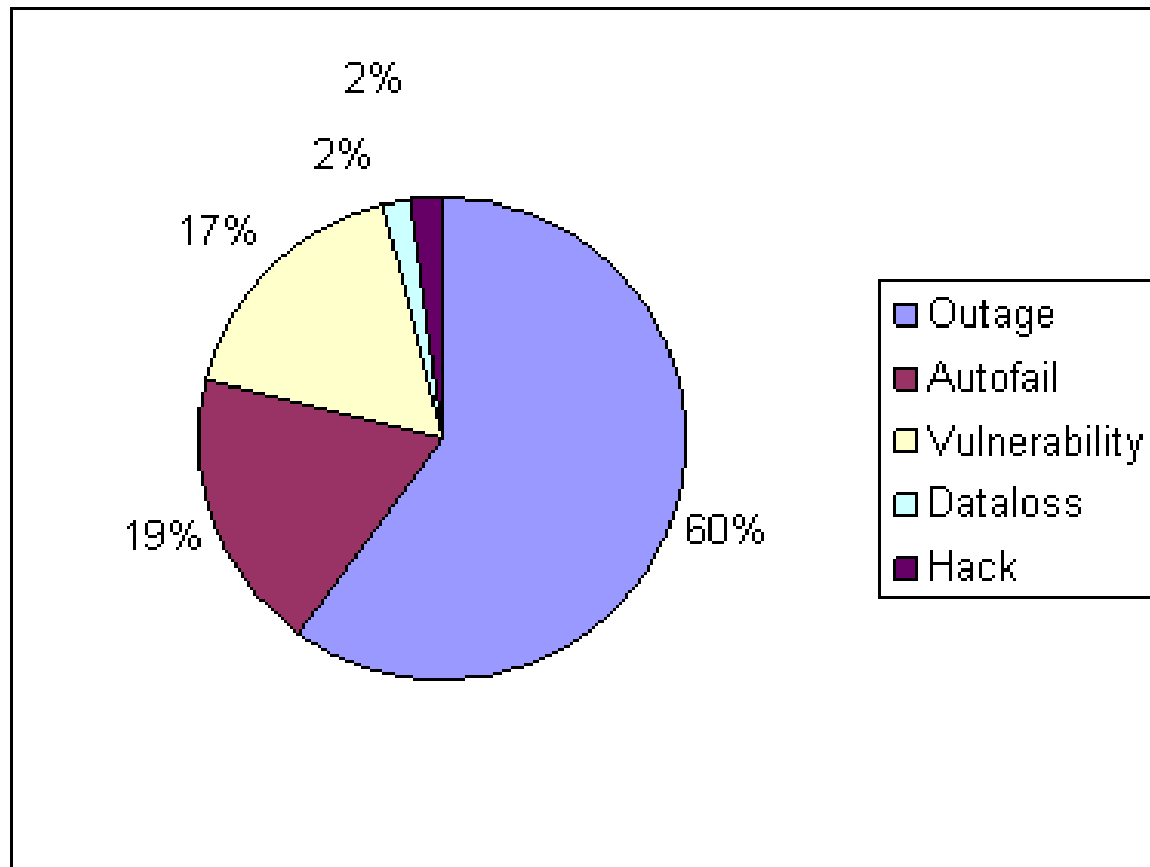
Security is the protection against:

- Unauthorized data disclosure (confidentiality or secrecy).
- Unauthorized data modification (integrity). Unauthorized modification of data may result in inconsistencies or erroneous data. Data destruction may bring all kinds of losses.
- Denial of service—Users or other systems may prevent the legitimate users from using their system. Denial of service is an attack on the availability of the system.
- Lack of accountability—Users should be responsible for their actions and should not be able to deny what they have done (non-repudiation).
- Protection of intellectual property









Countermeasures (defenses)

- Identification and Authentication (I&A)—Identification is a user or system action where the user provides an identity. Authentication implies some proof that a user or system is the one he/it claims to be. The result of authentication may be a set of credentials, which later can be used to prove identity and may describe some attributes of the authenticated entity
- Authorization and Access control (A & A)—Authorization defines permitted access to resources depending on the accessor (user, executing process), the resource being accessed, and the intended use of the resource. Access control requires some mechanism to enforce authorization
- Logging and Auditing—Implies keeping a log of actions that may be relevant for security. These functions can be used to collect evidence for prosecution (forensics) and to improve the system by analyzing why the attack succeeded.
- Hiding of information—It is usually performed by the use of cryptography but steganography is another option. The idea is to hide the information in order to protect it.
- Intrusion detection—Intrusion Detection Systems (IDS) alert the system when an intruder is trying to attack the system.

Cloud computing incidents: 128, 40, 37, 4, 4



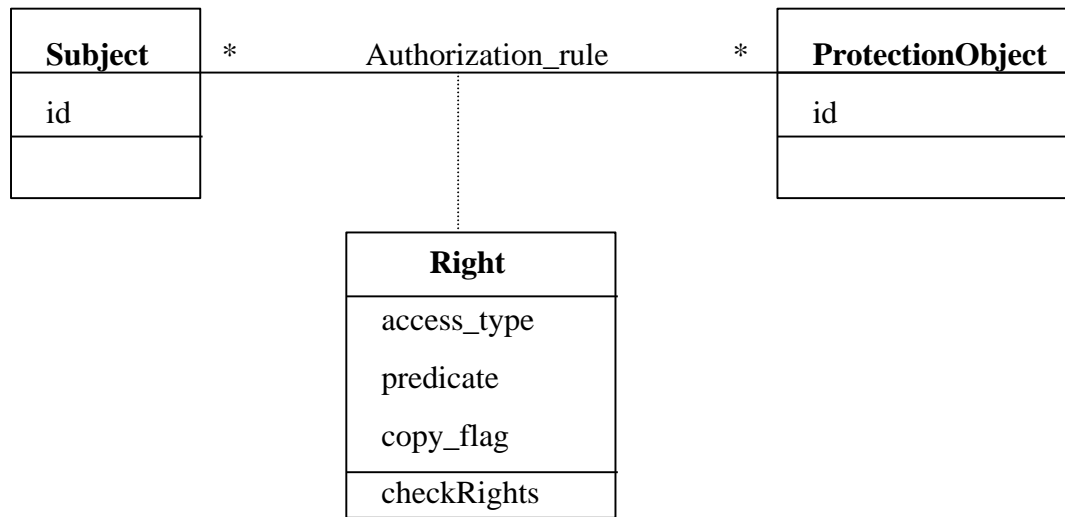
Cloutage.org

ID	Type	Reported	Summary	Organization	Services
500	 hack	2011-03-24	TripAdvisor Member Data Stolen in Possible SQL Injection Attack	Expedia	TripAdvisor
443	 hack	2011-01-21	Whirlpool Forum Hit with DDOS	Whirlpool	Forum
416	 hack	2010-11-17	sitelutions Suffers DDoS	sitelutions	DNS
409	 hack	2010-11-04	www.websites.intuit.com DoS	Intuit.com	Web Hosting
387	 hack	2010-11-01	DreamHost cardiff DDOS	DreamHost	Web Hosting
413	 hack	2010-10-25	User Details Posted in ISP Hack	MWEB	ISP
417	 hack	2010-10-25	Barclay's Bank Fails to Catch Malicious Automated Transactions	Barclays Bank	Barclays Bank
410	 hack	2010-10-19	Kaspersky Download Site Redirects Users to Fake AV	Kaspersky Lab	Antivirus
418	 hack	2010-10-01	Hacker Steals \$600K From TD Bank	TD Bank	Online Banking

Security patterns

- We can use security patterns to build systems that have systematic defenses against attacks
- We are developing a methodology to build such systems
- Now building also a catalog of patterns, over 70, and a catalog of cloud attacks
- Each pattern provides solutions with UML diagrams, maybe extended with formal specifications (OCL)

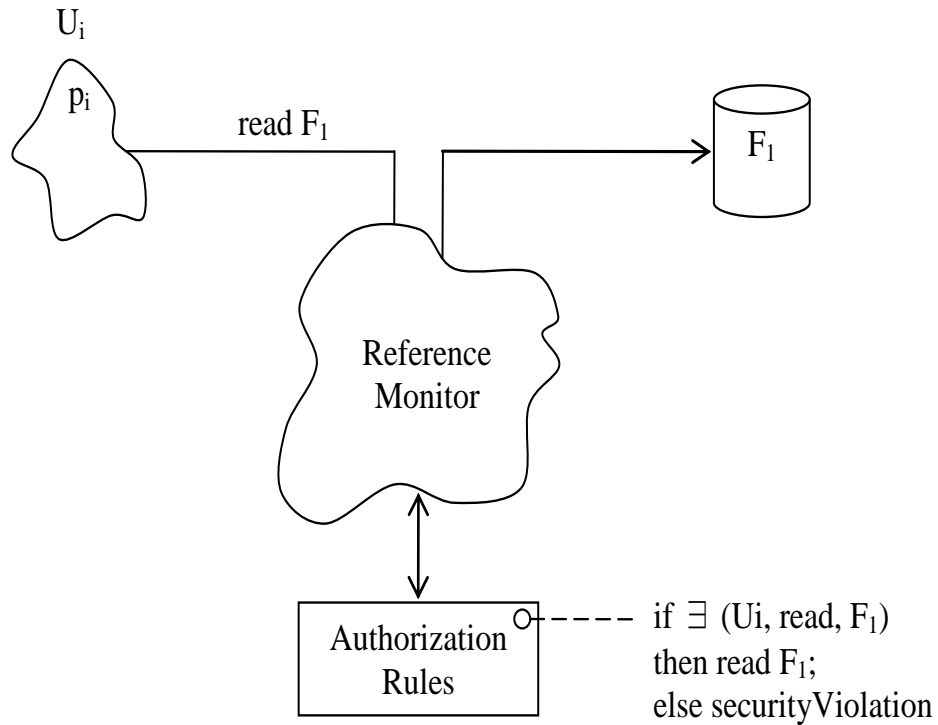
Authorization pattern

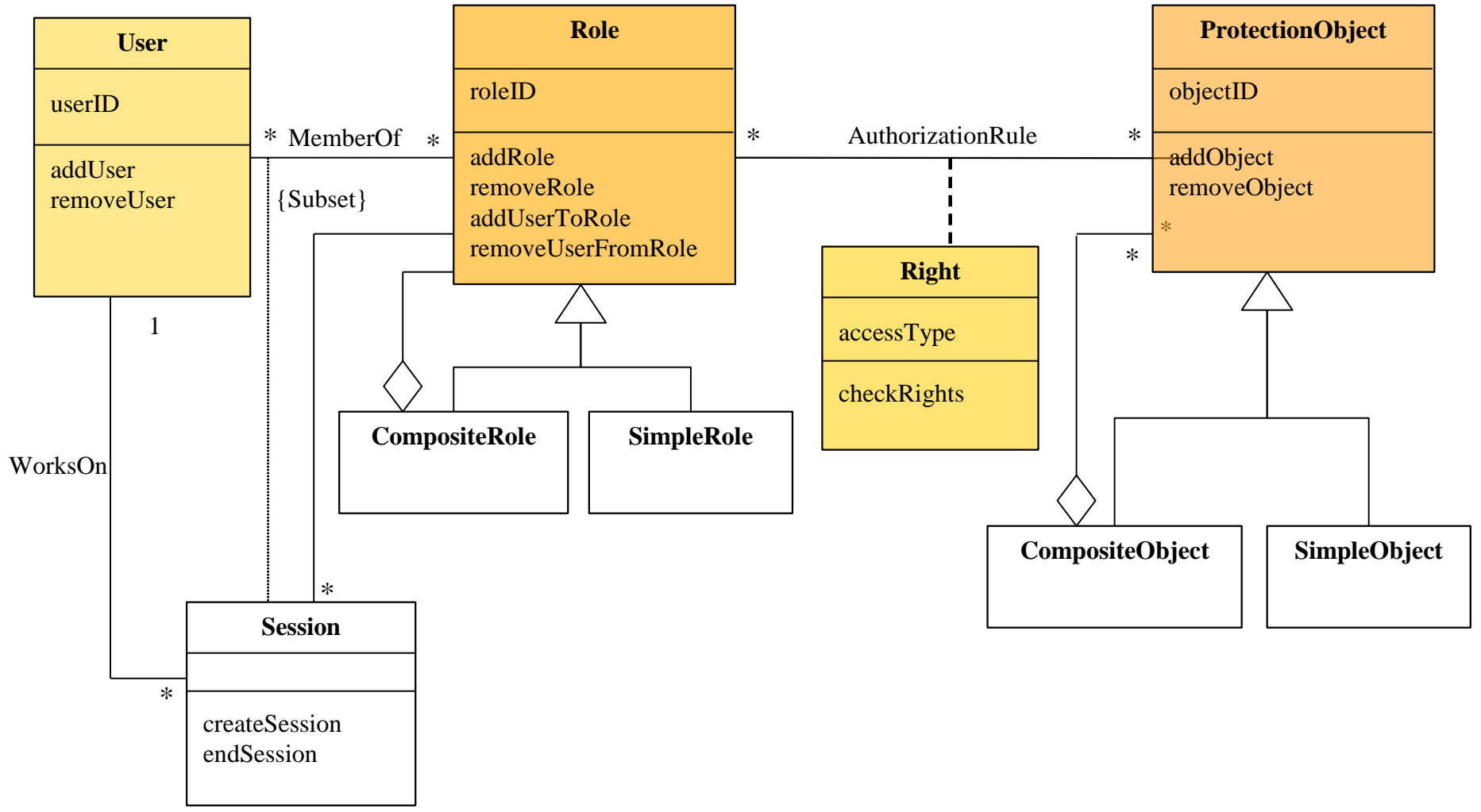


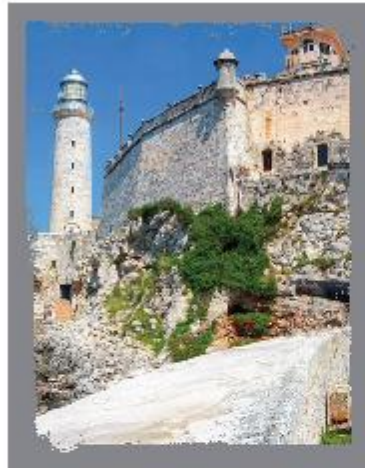
Reference Monitor

- Each request for resources must be intercepted and evaluated for authorized access
- Abstract concept, implemented as memory access manager, file permission checks, CORBA adapters, etc.

Reference Monitor idea







Eduardo Fernandez-Buglioni

SECURITY PATTERNS IN PRACTICE

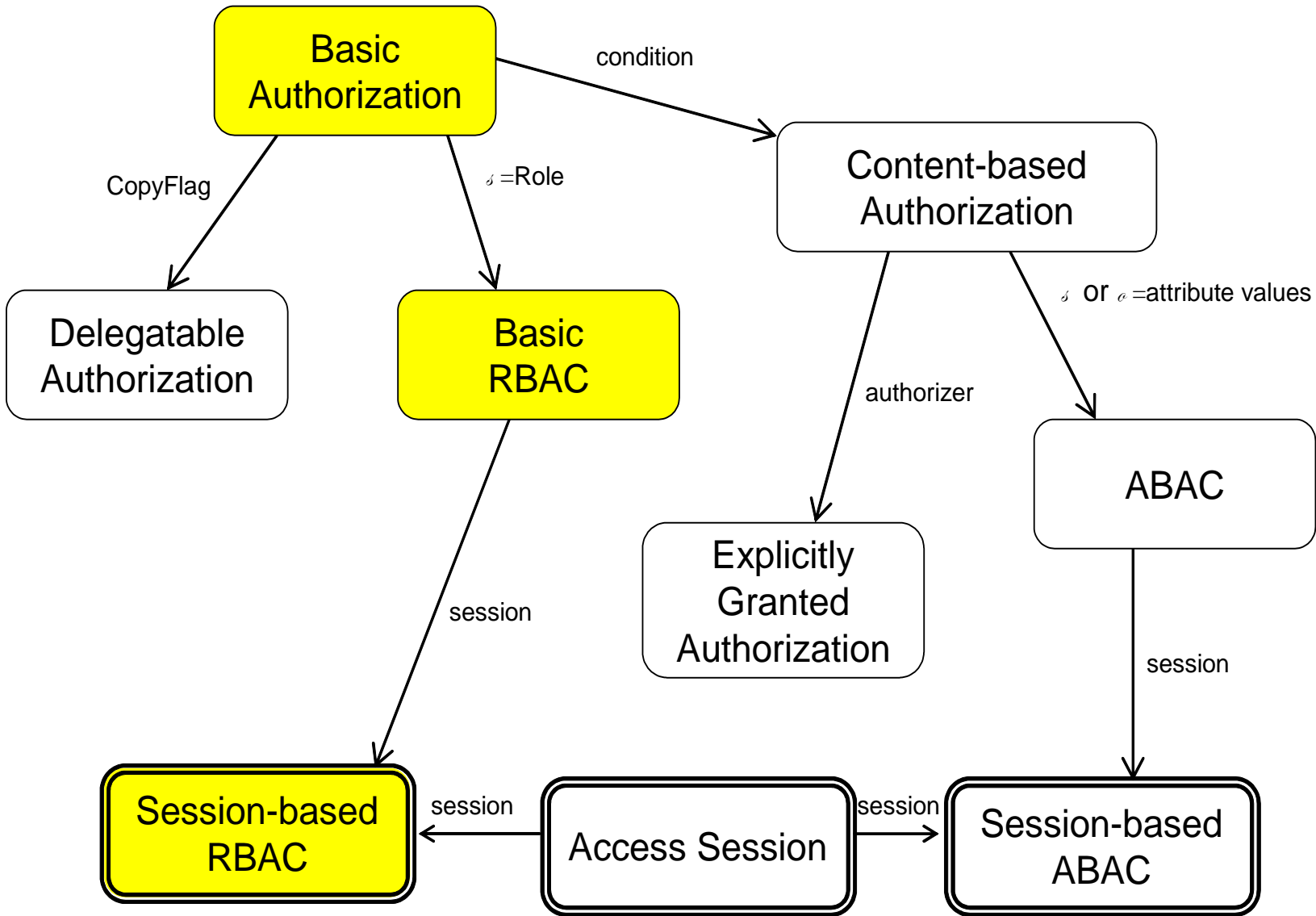
Designing Secure Architectures
Using Software Patterns



WILEY SERIES IN
SOFTWARE DESIGN PATTERNS

My view of security

- I see security as a systems/ software architecture concern
- The organization of the units and their interconnection is fundamental to define its security properties
- Code checking helps but only for specific products, every time we need to do it again
- Theoretical models provide boundaries but cannot be used to build real systems



Conclusions

- New concepts: SLA (Service Level Agreement), SOAP/REST, NoSQL databases, 3 levels of service, several deployment modes
- Security is a big concern
- Patterns cannot prevent attacks that happen through code flaws but can make their effect much less harmful
- Can be made more formal: OCL
- Security patterns are now accepted by many companies, Microsoft, Sun, and IBM have books, papers, and web pages on this subject.