INTRODUCTION

When you store your photos online instead of on your home computer, or use webmail or a social networking site, you are using a "cloud computing" service. If you are an organization, and you want to use, for example, an online invoicing service instead of updating the in-house one you have been using for many years, that online invoicing service is a "cloud computing" service.

Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. Doing so may give rise to certain privacy implications.

For that reason the Office of the Privacy Commissioner of Canada (OPC) has prepared some responses to Frequently Asked Questions (FAQs). We have also developed a Fact Sheet that provides detailed information on cloud computing and the privacy challenges it presents.

Cloud Computing

Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications.

The following definition of cloud computing has been developed by the U.S. National Institute of Standards and Technology (NIST):

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.1

Characteristics

The characteristics of cloud computing include on-demand self service, broad network access, resource pooling, rapid elasticity and measured service. On-demand self service means that customers (usually organizations) can request and manage their own computing resources. Broad network access allows services to be offered over the Internet or private networks. Pooled resources means that customers draw from a pool of computing resources, usually in remote data centres. Services can be scaled larger or smaller; and use of a service is measured and customers are billed accordingly.

Why cloud services are popular

Cloud services are popular because they can reduce the cost and complexity of owning and operating computers and networks. Since cloud users do not have to invest in information technology infrastructure, purchase hardware, or buy software licences, the benefits are low up-front costs, rapid return on investment, rapid deployment, customization, flexible use, and solutions that can make use of new innovations. In addition, cloud providers that have specialized in a particular area (such as e-mail) can bring advanced services that a single company might not be able to afford or develop.

Some other benefits to users include scalability, reliability, and efficiency. Scalability means that cloud computing offers unlimited processing and storage capacity. The cloud is reliable in that it enables access to applications and documents anywhere in the world via the Internet. Cloud computing is often considered efficient because it allows organizations to free up resources to focus on innovation and product development.

Another potential benefit is that personal information may be better protected in the cloud. Specifically, cloud computing may improve efforts to build privacy protection into technology from the start and the use of better security mechanisms. Cloud computing will enable more flexible IT acquisition and improvements, which may permit adjustments to procedures based on the sensitivity of the data. Widespread use of the cloud may also encourage open standards for cloud computing that will establish baseline data security features common across different services and providers. Cloud computing may also allow for better audit trails. In addition, information in the cloud is not as easily lost (when compared to the paper documents or hard drives, for example).

Potential privacy risks

While there are benefits, there are privacy and security concerns too. Data is travelling over the Internet and is stored in remote locations. In addition, cloud providers often serve multiple customers simultaneously. All of this may raise the scale of exposure to possible breaches, both accidental and deliberate. Security issues, the need to segregate data when dealing with providers that serve multiple customers, potential secondary uses of the data—these are areas that organizations should keep in mind when considering a cloud provider and when negotiating contracts or reviewing terms of service with a cloud provider. Given that the organization transferring this information to the provider is ultimately accountable for its protection, it needs to ensure that the personal information is appropriate handled.

Developing cloud services

The cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In a Software as a Service model, a pre-made application, along with any required software, operating system, hardware, and network are provided. In PaaS, an operating system, hardware, and network are provided, and the customer installs or develops its own software and applications. The IaaS model provides just the hardware and network; the customer installs or develops its own operating systems, software and applications.

Deployment of cloud services:

Cloud services are typically made available via a private cloud, community cloud, public cloud or hybrid cloud.

Generally speaking, services provided by a **public cloud** are offered over the Internet and are owned and operated by a cloud provider. Some examples include services aimed at the general public, such as online photo storage services, e-mail services, or social networking sites. However, services for enterprises can also be offered in a public cloud.

In a **private cloud**, the cloud infrastructure is operated solely for a specific organization, and is managed by the organization or a third party.

In a **community cloud**, the service is shared by several organizations and made available only to those groups. The infrastructure may be owned and operated by the organizations or by a cloud service provider.

A **hybrid cloud** is a combination of different methods of resource pooling (for example, combining public and community clouds).

Surveying the Role of Cloud Computing

1. **Cloud computing definition**

2.Why Migrate Applications and Services to the Cloud?

- Investment
- Long time planning
- reduce loses
- usage based payments

3.Cloud Computing’s Ancestry (origin)

- Personal Computer
- Network
- Thin client - Web TV (Microsoft)
- Netbooks
- Web hosting , ASP

ASPs gradually became known as *Software as a Service* (*SaaS*) providers. There are five generally accepted ASP market segments:

Specialty ASPs usually deliver a single application, such as credit card or other payment processing, customer relationship management (CRM), human resources management system (HRMS), word processing, spreadsheet, database or timesheet services. Google Apps provide web-based email, calendar, word-processing, spreadsheet and presentation modules to business users for a fixed charge per user per year, while Salesforce.com rents CRM capabilities and Intuit provides its QuickBase RDBMS with per subscriber per month billing.

Enterprise ASPs deliver a broad spectrum of specialty ASP solutions. For example, Microsoft rents Microsoft SharePoint Services, Microsoft Dynamics CRM Services, and Office Business Applications (OBAs), as well as Windows Live services online.

Vertical-market ASPs deliver multiple software solutions for a specific customer category, such as medical or dental practice, insurance brokerage, church congregation, residential or commercial construction, or personal finance management.

Local-market ASPs deliver geocoded marketing services to small service businesses, such as restaurants, pubs and bars, within a limited geographic region.

4.Cloud Computing and Everything as a Service

Files [storage] as a Service: FaaS, often called *Data Storage as a Service* (DaaS), lets users store files of various data types in a highly scalable hierarchical file system and retrieve them over the Internet as various Multipurpose Internet Mail Extension (MIME) types. FaaS was one of the first cloud-based services. Several Internet start-ups, such as SmugMug, DropBox, Ozmo, and HolaServers, use AmazonWeb Services' Simple Storage Service (S3) to hold graphic images and other files, charging users a small or no access fee. Microsoft Live SkyDrive is a FaaS provider that gives users up to 25GB of free file storage at no charge.

The term *Data Storage* or *Database as a Service* implies structured storage with at least some relational database management system (RDBMS) features, such as query capabilities, primary and foreign key indexes, and entity associations through simulated JOINs. Commercial cloud services, such as Amazon Web Services (AWS), Google App Engine (GAE), and Windows Azure, offer indexed Entity-Attribute-Value (EAV) tables and query languages having some relationship to SQL. Microsoft says SQL Azure Database (SADB) ‘‘offer highly scalable and Internetfacing distributed database services in the cloud for storing and processing relational queries.’’ SADB, Amazon SimpleDB, and GAE's DataStore offer advanced features that qualify them as Databases as a Service (DBaaS).

Software as a Service: SaaS delivers a packaged or equivalent commercial software application to end users over the Internet with a subscription or usage-based pricing model, as opposed to a traditional lifetime license for a particular version. Examples include Microsoft Office Live, Microsoft Exchange Online, Microsoft SharePoint Online, Microsoft Dynamics CRM Online, and Salesforce.com. Microsoft was an early SaaS supporter with SOAP-based web services but has gradually migrated to promoting Software plus Services (S+S). *Application as a Service* is a synonym for SaaS.

Infrastructure as a Service: IaaS provides traditional data center resources, such as highly scalable virtualized computing power, memory and storage, over a network (typically, but not necessarily, the Internet) and usually with a subscription or per usage pricing model. IaaS is also called *utility computing*. Internet-delivered cloud examples include AmazonWeb Services, GoGrid, and Flexiscale. IaaS or PaaS delivered over an intranet is called a *private cloud*.

Communication as a Service: CaaS provides communication capability that is service-oriented, configurable, schedulable, predictable, and reliable, as well as network security, dynamic provisioning of virtual overlays for traffic isolation or dedicated bandwidth, guaranteed message delay, communication encryption, and network monitoring. CaaS is critical to meeting Service Level Agreements (SLAs) but usually is considered to be a component of SaaS, S+S, or IaaS.

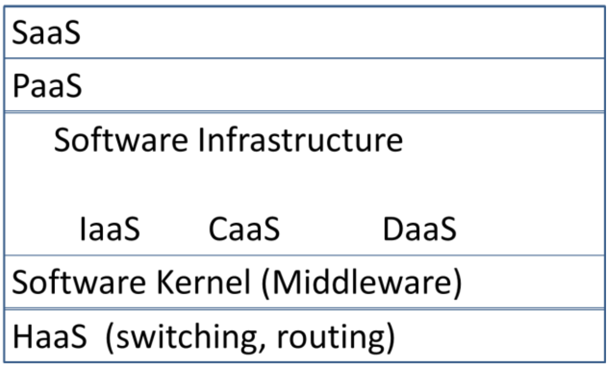
Monitoring as a Service: MaaS notifies the user of cloud computing or network outages, errors, or slowdowns. For example, Cloud Status is a simple iPhone application that monitors the status of Amazon Web Services, Google App Engine, and Twitter and reports whether service is normal, has problems, or is down. MaaS can contain auditing components for network vulnerability assessment or to verify SLA conformance and the accuracy of monthly usage charges.
Some suppliers of MaaS services, such as RightScale, also provide instance deployment automation for increasing the number of running AMI instances during demand peaks and reducing the number as demand subsides.

Platform as a Service: PaaS usually comprises at least these three distinct elements:

Tools as a Service (TaaS), which provides Web-based development tools and languages, such as Microsoft Visual Studio (for Visual C#, Visual Basic, IronPython, and IronRuby) or open-source Eclipse (primarily for Java). The Windows Azure Tools for VS 2008 include templates for creating Web, Worker, Web and Worker, and Cloud Sequential Workflow Services that can run under a local (developer) or cloud (production) Windows Azure instance (fabric). Google App Engine offers a hosted Python variant as well as webapp and Django frameworks.

5. Cloud Computing Ontologies

Cloud Computing Ontologies : 5 layer Model



SaaS

PaaS

Software Infrastructure

IaaS CaaS DaaS

Software Kernel (Middleware)

HaaS (switching, routing)

Cloud application developers use the Cloud Software Environment Layer, which provides support for a programming language and a set of application programming interfaces (APIs) ‘‘to facilitate the interaction between the environments and the cloud applications,’’ which leads to the Platform as a Service moniker.

The Cloud Software Environment Layer is built on the Software
Kernel and Firmware/Hardware layers and provides Computational Services (IaaS), Data Storage (DaaS), and Communication (CaaS) services. Virtual machines (VMs) commonly deliver IaaS, although Windows Azure offers the option of a dedicated server running Windows Server 2008. However, it’s arguable that CaaS capabilities belong at the lower Firmware/Hardware (HaaS) level because off-premises HaaS isn’t practical without CaaS.

The Software Kernel can be implemented as an OS kernel, hypervisor, virtual machine monitor and/or clustering middleware, or various combinations of these systems. Although grid applications played a significant role in early cloud computing implementations, the grid has given way to the hypervisor as the preferred software kernel for cloud computing because the latter abstracts hardware idiosyncrasies from the service. Adding CaaS makes this layer equivalent to traditional VPS Web hosting.

The Firmware/Hardware layer is the physical computing, switching, and routing hardware that forms the cloud's backbone. The HaaS provider operates, manages, and upgrades the hardware on behalf of its lessees, who supply their own operating system and application software, and charges by the GB for data ingress and egress, similar to web server colocation. Leasing eliminates users' need to invest in building and managing data centers and might reduce the cost of power and insurance.

6. Cloud Computing Concerns

Define Your Governance Needs: Are they internal, external, legal? List the requirements and how they're satisfied.

Classify Your Data: Before you can determine what data you can safely put in the cloud, you first have to classify and label it according to sensitivity and type.

Choose Wisely: Identify cloud vendors that can satisfy your processing and governance needs. Direct business leaders to walk away from the rest, no matter how attractive pricing is.

Set Limits: Define what the service provider can do with your data. Prohibiting the outsourcing of processing to a third party without your consent is basic.

Put Rules in Writing: Publish policies and procedures stating which cloud vendors can receive which types of data.

Developing Cloud Services

The cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In a Software as a Service model, a pre-made application, along with any required software, operating system, hardware, and network are provided. In PaaS, an operating system, hardware, and network are provided, and the customer installs or develops its own software and applications. The IaaS model provides just the hardware and network; the customer installs or develops its own operating systems, software and applications.

Deployment of cloud services:

Cloud services are typically made available via a private cloud, community cloud, public cloud or hybrid cloud.

Generally speaking, services provided by a **public cloud** are offered over the Internet and are owned and operated by a cloud provider. Some examples include services aimed at the general public, such as online photo storage services, e-mail services, or social networking sites. However, services for enterprises can also be offered in a public cloud.

In a **private cloud**, the cloud infrastructure is operated solely for a specific organization, and is managed by the organization or a third party.

In a **community cloud**, the service is shared by several organizations and made available only to those groups. The infrastructure may be owned and operated by the organizations or by a cloud service provider.

A **hybrid cloud** is a combination of different methods of resource pooling (for example, combining public and community clouds).

Advantage of Auxiliary Cloud Services

The Windows Azure Platform incorporates three sets of auxiliary services .NET Services, SQL Azure Database, and Live Services.

.NET services

Microsoft .NET Services are a set of scalable, developer-oriented services that are hosted by Windows Azure alongside Azure Storage Services in Microsoft data centers. The Microsoft .NET Services SDK (March 2009 CTP) provides class libraries, samples, and documentation for building connected applications with the .NET platform. Access Control, Service Bus, and Workflow services take advantage of Web-standard HTTP protocols, so any application that has reliable Internet access can use them and they're compatible with other popular programming languages, such as Java and Ruby (Ruby is a object-oriented, general-purpose programming language).

Access control services

Microsoft claims that Access Control Services (ACS) ''provide an easy way to control web applications and services while integrating with standards-based identity (identity services running in the cloud that you can use for managing access by employees, partners, and customers to your corporate assets, including both on-premises and cloud assets) providers. The advantage of ACS is that you can write a set of declarative rules that can transform incoming security claims into a claims-based, federated identity to minimize developer effort.

Service Bus

Microsoft states that the .NET Service Bus (SB), which was originally known as *BizTalk Services*, makes it easy to connect applications together over the Internet. Services that register on the Bus can easily be discovered and accessed, across any network topology. When a new service connects to the bus all other applications and services on the bus can connect with it, even if they could not connect directly with one another.

Example:

If you deploy your system locally within 1 region, e.g. East Australia, then pretty-much straight forward. Now, let's assume that you have a growing user-base, and after further analysis, you can split your users origins into Oceania & Europe. To give better performance, it's better to deploy your system to the Europe & East Australia datacentres. However, now you have an issue of consistency between the databases. Yes you can rely on SQL Server's (or whatever your database flavour is) replication feature, however, those can be expensive - you generally have to use top end flavour of SQL Server as well as transport costs as well, to name a few.

Service bus can be a real help here. If you put Service Bus + Worker Role between Service/API layer and Data-access/Database layers; then instead of directly updating the database, the information is passed on the service bus. And the worker role will then fetch the message from the queue and do both data updating of that region database (e.g. Europe) as well as pushing that same message into the service bus on the other region (e.g. East Australia) - of which the worker role in that region will fetch that message and updates the East Australia region database.

Workflow services

Microsoft describes the .NETWorkflow Services (WFS) as ‘‘**a high-scale host** for running workflows in the cloud.’’ WFS orchestrates the sending, receiving, and manipulating of HTTP and Service Bus messages. It also provides hosted tools to deploy, manage, and track the execution of workflow.

Deploying Application and Services to the Azure Cloud.

Windows Azure Cloud Services allows you to quickly create, deploy, and manage multitier applications in the cloud. You can define multiple roles for your application to distribute processing and allow flexible scaling of your application. Cloud Services applications can be built using almost any popular development framework including .NET, Node.js, PHP, Java, Python, and Ruby. You can also integrate Windows Azure Mobile Services and Media Services with your cloud application.

With Cloud Services, you can focus your attention on building, testing, deploying, and managing your application instead of focusing on the underlying infrastructure on which your application runs. You don’t need to worry about patching servers, dealing with hardware failures, or troubleshooting network issues since Windows Azure is designed to allow applications to be available even in the event of hardware failures or system upgrades.

You can use the Windows Azure Management Portal to monitor the health and availability of applications running on Cloud Services. You can configure alerts so you can be notified in real-time should a service interruption or service degradation occur. And you can use the new Autoscale feature to allow your application to automatically scale up or down as demand changes. This helps minimize the cost of running your applications in the cloud since you only pay for the execution resources that you actually use.

To create a cloud service, you first need to understand a number of concepts. A cloud service role, which consists of your application files and XML configuration files, can be either a web role or a worker role. A web role provides a dedicated IIS web server and is typically used for hosting front-end web applications or mid-tier service layers. Worker roles, on the other hand, host applications that can run asynchronously and are generally used to perform long-running data processing tasks that are independent of user input or interaction.

A role instance is the virtual machine on which the application code and role configuration run. Each role can have one or several instances as defined in the cloud service configuration file (.csdef) which defines the service model for the application. The cloud service configuration file (.cscfg) specifies configuration settings for the cloud service and its individual roles, including the number of role instances. Finally, the service package (.cspkg) contains the actual application code along with the service definition file.

To create a new cloud service in Windows Azure, open the Windows Azure Management Portal, select the Cloud Services tab on the left and click the New button in the command bar at the bottom. The command bar expands and displays two options for creating cloud services: Quick Create or Custom Create.

Once you’ve created your cloud service using the Quick Create option, you can install the appropriate Windows Azure SDK needed for running your application and any SSL certificates needed by your application. The next step is to decide which environment you want to deploy your new application to. Windows Azure provides two deployment environments for cloud services: staging and production. The staging environment is where you can test your deployment before you “swap” it into your production environment by switching the virtual IP addresses (VIPs) by which your cloud service is accessed.

You have the choice of deploying your application even if one or more roles contain a single instance, but you should generally ensure that every role has at least two instances since Windows Azure can only guarantee 99.95 percent uptime if this is the case. The Custom Create option is similar to Quick Create but also enables you to deploy your cloud service package when you create the new cloud service. Just keep in mind these two things:

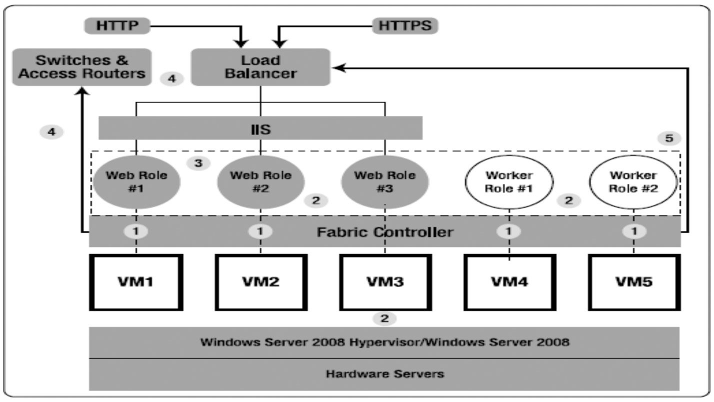
1. Avoid using the Windows registry.
2. If you are using a web.config or app.config files, you should instead consider using a service configure (.cscfg) file.

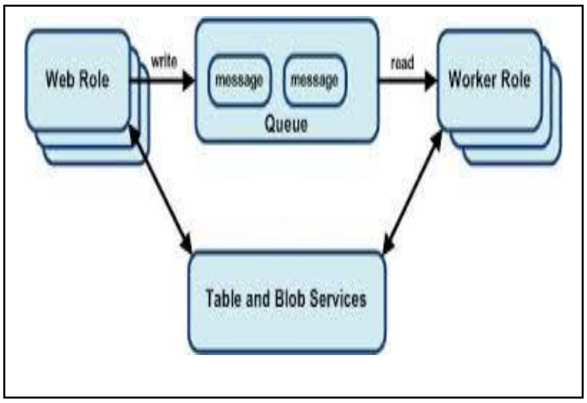
Unit – 2

Life cyle - Windows Azure Platform

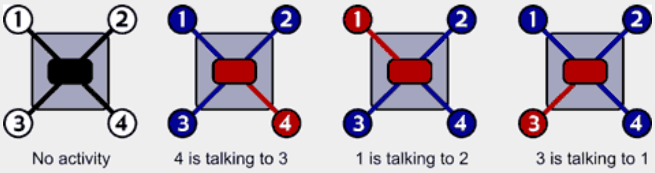
Windows Azure is a ‘‘cloud layer’’ operating system that runs on thousands of Windows Server 2008 physical instances in Microsoft data centers**.** Scaling and reliability (replication) are controlled by the **Microsoft Azure Fabric Controller** so the services and environment do not crash if one of the servers crashes within the Microsoft data center and provides the management of the user's web application like memory resources and load balancing. The data center’s physical servers run an advanced, custom version of Microsoft’s Hyper-V hypervisor technology that virtualizes the physical instances.

Roles are runnable components of an application. WebRole instances accept HTTP or HTTPS requests via Internet Information Services (IIS) 7and respond with an ASP.NET, ASP.NET MVC, or Silverlight UI. WorkerRoles provide batch computing services in response to request messages received from WebRoles or .NET Services in Azure Queues. Each WebRole or WorkerRole is assigned to its own guest VM and server core to isolate the tenant’s data.

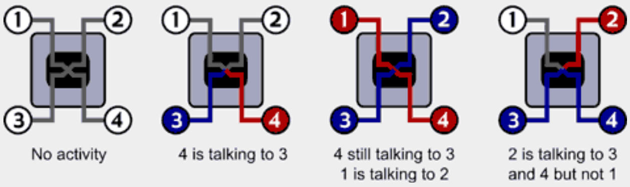




Both hubs and switches serve that purpose, and from the outside, they function identically: they allow the connected computers to exchange data among themselves. However, the way they handle data internally is very different. You can think of a ***hub*** like a house with 4 rooms, 4 people, and 4 phones but only one phone number. Each person has the phone to his ear, and they can converse with each other, but if one person speaks, everyone can hear it regardless if the statement was intended for them or not. So, if person 4 wanted to send a message to person 3, he would have to tell everyone to be quiet, say "this message is for person 3" and then say the message.



Imagine the same situation except that each room has its own telephone number. This situation describes a ***switch***. If person 4 wanted to send a message to person 3, he could call directly to that room without disturbing the people in rooms 1 or 2. That means that at the same time 3 and 4 are talking, room 1 and 2 could have a conversation without disrupting any other conversations.



Thus, the difference between a switch and a hub is that a switch can handle multiple communications between the computers attached to it whereas a hub handle one at a time. If there are only two computers transmitting data across a network, a hub would perform identically to a switch. However, if more than two computers were trying to transmit across the network at the same time, the switch would perform far better. Whereas a hub and switch serve the same function, a ***router***serves a slightly different function. A router is explicitly designed to connect two networks together, usually a *Local Area Network, or LAN* (like a single small office) to a *Wide Area Network, or WAN*(like the Internet). A router also has additional "smart" software with security features that disallow unauthorized access to the computers in the LAN from the outside.

Messaging

Business applications are often multitier in nature, and code running in each tier needs to be able to communicate with code in other tiers in a way that is fast, secure, and reliable. For applications deployed in the cloud, this can become a key issue since different application components often run on physical servers in datacenters located in different geographical locations, sometimes even on separate continents.

Windows Azure provides several ways for the different components of a cloud-based application to effectively communicate with one another: Windows Azure Queue, Windows Azure Service Bus, and Windows Azure Notification Hubs.

Windows Azure Queue

Let’s say that one piece of application code sends a message to another piece of code. What if the receiving code isn’t ready to process the message? In that case, a simple solution is to temporarily store the message in a queue until the receiving code is ready to process it.

For cloud-based applications that can use this kind of approach, Windows Azure Queue can provide exactly what they need. For example, when PHP code running in a Web role needs to communicate with code running in a Worker role of the same cloud service, simple message queuing using Windows Azure Queue enables the Worker role to perform asynchronous processing of the message when the role is ready to do so.

Message queuing has several advantages when building cloud-based applications. For one thing, it’s easy to implement. It also scales well, for in the previous example you could easily increase the number of Web role instances or Worker role instances to meet increasing demand.

Windows Azure Service Bus

While simple message queuing is good for one-to-one communications between application components, it doesn’t provide a good solution in scenarios where one-to-many communication is needed between components. To address this need, Windows Azure provides another service called Windows Azure Service Bus. Windows Azure Service Bus supports both basic queuing and publish-and-subscribe forms of messaging.

The publish-and-subscribe approach enables one piece of code to send a message on a topic and have multiple other pieces create subscriptions to the topic. Service Bus thus enables Windows Azure applications to communicate with other Windows Azure applications, with applications running on some other cloud platform, or even with applications running outside the cloud.

Tables

Tables is a Windows Azure data management capability that can store large amounts of unstructured data. You can then access this data using REST application programming interfaces (APIs) either from within a service running in Windows Azure or directly over the Internet using HTTP/HTTPS request/response.

BLOB storage

BLOBs provide a simple mechanism for storing large amounts of text or binary data such as images, audio, or visual files. Windows Azure BLOB Storage can autoscale up to 200 terabytes and can be accessed using REST APIs in the same way as Tables. Applications running in Windows Azure can also mount a BLOB formatted as a single volume NTFS virtual machine which you can then move between private and public clouds using Windows Azure Drive.

The life cycle proceeds in the following steps.

1. Creating VM on physical server. The process of adding new VMs is wrapped in a transaction. If any operation in the process fails, all previous operations are rolled back.
2. Maintaining Role Instance Health
   The FC is responsible for keeping services running by inspecting their state and adding or removing role instances. Following are the primary FC responsibilities for maintaining service availability:
   - ❑ The FC maintains a state machine for each node.
   - ❑ The FC maintains a cache of last state of each node.
   - ❑ Load balancers probe the nodes to determine that each is operable and reports failures to the FC.
   - ❑ The FC detects when the node reaches its goal state.
   - ❑ If the FC can’t recover a failed node, it finds or creates a suitable replacement on other hardware and notifies other role instances of the configuration change.
3. Upgrading Service Software and Windows Azure: Rolling service software upgrades and patches to the Windows Azure OS take place within transactions on running services in one Upgrade Domain at a time. The FC deploys resources to all nodes of the Upgrade Domain in parallel, so all updated services in the Upgrade Domain go offline temporarily during the upgrade while new logical role instances bind to physical nodes.

Securing and Isolating Services and Data

IT managers simply said that the threat to network security and data privacy presented an outrageous risk to the organizations’ survival.

IT departments’ focus was on *perimeter security* to ‘‘keep the bad guys out’’ of the corporate network by using stringent Network Access Control (NAC) parameters. NAC is a mix of hardware and software technology that dynamically controls client system access to networks based on their compliance with policy.

The need to accommodate an increasingly mobile sales force, support telecommuting employees, and acquiesce to on-premises workers’ demands for high-speed Internet access to business-related information gradually overcame IT departments’ nay-saying. Firewalls that permitted users and their applications to connect to the Internet with a limited number of open TCP ports provided the illusion of security to IT and corporate management.

Most IT departments currently view that allowing services to be delivered by third parties means they lose control over how data is secured, audited, and maintained and they can’t enforce what they can’t control.

The key to acceptance of third-party security, auditing, and maintenance of customers’ data in the cloud is *transparency*. Cloud-computing vendors, such as Microsoft, must fully detail their security-related practices and incorporate guaranteed levels of data security, auditing, availability, and reliability in their service-level agreements (SLAs.)

Security for applications’ data against access by unauthorized users from other organizations using the same service is one of the most important incentives for adopting virtualization in cloud computing by independent software vendors (ISVs). Windows Azure implements some *multitenant computing* also called *multitenancy* features.

Multitenancy refers to a principle in software architecture where a single instance of the software runs on a software-as-a-service (SaaS) vendor’s servers, serving multiple client organizations (tenants). Multitenancy is contrasted with a multi-instance architecture where separate software instances (or hardware systems) are set up for different client organizations. With a multitenant architecture, a software application is designed to virtually partition its data and configuration so that each client organization works with a customized virtual application instance.

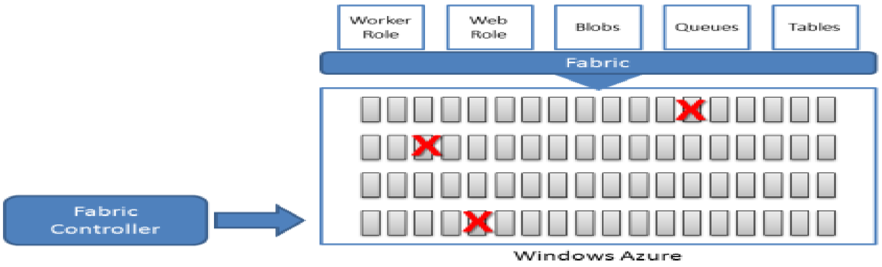
In a pure multi-tenant architecture a single instance of the hosted application is capable of servicing all customers (tenants). Unlike more classical web applications or web services ‘‘in the cloud’’ which behave the same way for each requests, a multi-tenant architecture is designed to allow tenant-specific configurations at the UI (branding),business rules, business processes and data model layers. This is has to be enabled without changing the code as the same code is shared by all tenants, therefore transforming customization of software into configuration of software. As you can imagine, this drives the clear need for ‘‘metadata driven everything.’’ The other main challenge is being able to co-locate (mingle and ‘‘de-mingle’’) persistent data of multiple tenants in the same data infrastructure.

In other words, the challenge for the multi-tenant application is to behave as if it was fully dedicated to a single tenant but is actually serving all of them in parallel on the same code base. I call this ‘‘you are my #1 customer’’ approach; which means every customer believes they are the #1 customer but in reality they are all served by a talented customer rep. The main advantage of this architecture is (at least) twofold (a) the underlying infrastructure is shared, allowing massive economy of scale with optimal repartition of load and (b) because the very costly nfrastructure and application development costs are shared, the ‘‘enterprise grade’’ application can be offered to very small businesses as well, permitting [it] to address the long tail of the market.

Fabric (Framework) controller

The generic term fabric is a synonym for framework. The Azure Fabric Controller functions as the kernel of the Azure operating system[fabric].

It provisions, stores, delivers, monitors and commands the virtual machines (VMs) and physical servers that make up Azure. Microsoft data center stores all the data of Windows Azure storage and all Windows Azure applications. Windows Azure Fabric Controller controls and manages the servers, the set of machines which are dedicated to Windows Azure and the software that runs on the Microsoft Data Center. The Windows Azure Fabric Controller is a distributed application that is replicated among the group of machines.



It has its own set of resources in its own environment like computers, load balancers, switches etc. Since Windows Azure Fabric Controller can communicate with the fabric agent on each machine; it keeps a track of all Windows Azure application in the fabric [The Fabric is an abstract model of the massive number of servers in the Azure data center. The Fabric Controller manages everything].

This helps the Windows Azure Fabric Controller to perform useful activities like monitoring all the running applications. The Windows Azure Fabric Controller decides where new applications will run, and also selects the physical server so that the hardware is utilized optimally. This is achieved using the configuration information which is uploaded with each Windows Azure application. Windows Azure Fabric Controller achieves this using the configuration information which is uploaded with each application on Windows Azure. The configuration file is an XML file which explains the various instances of the application, number of virtual machines to be created for the application, etc.

The Windows Azure Fabric Controller also manages the operating system in each instance for Web and Worker roles which includes OS Patch updates, system updates and software updates. This allows developers to focus exclusively on the creation of applications and not worry about the management of the platform.

One thing to keep in mind is that the Windows Azure Fabric Controller always assumes that at least two instances of each function are underway. This allows you to turn one of them to update their software without stopping the entire application. This is one of the reason why running a single instance of any feature of Windows Azure is generally a bad idea.

Allocating resources

One of the key jobs of the FC is to allocate resources to services. It does this by analyzing the service model of the service, including the fault and update domains, and the availability of resources in the Fabric. Using a greedy resource allocation algorithm, it finds which nodes can support the needs of each instance in the model. Once it has reserved the capacity, it updates the FC data structures in one transaction. Once this is done, the goal state of each node has been changed, and the FC starts moving each node towards its goal state by deploying the proper images and bits, starting up services, and issuing other commands through the driver model to all of the resources needed for the change.

Instance management

The FC is also responsible for managing the health of all of the nodes in the Fabric as well as the health of the services running. If it detects a fault in a service, it will try to remediate that fault, perhaps by restarting the node or taking it offline and replacing it with a different node in the Fabric. When a new container is added to the data center, the FC performs a series of burn-in tests to ensure that the hardware delivered is working properly. Part of this process results in the new resources being added into the inventory for the data center, making it available to be allocated by the FC. If hardware is ever to be determined to be faulty, either during installation or during a fault, the hardware is flagged as unusable in the inventory and left alone until later. Once a container has enough failures, the remaining workloads are moved to different containers and then the whole container is taken offline for repair. Once the problems have been fixed, the whole container is retested and returned into service.

The service model and you

The driving force behind what the FC does is the service model that you define for your service . You define the service model in an indirect manner. When you are developing a service, you define the following:

- Some configuration on what the pieces to your service are
- How the pieces communicate
- Expectations you have about the availability of the service

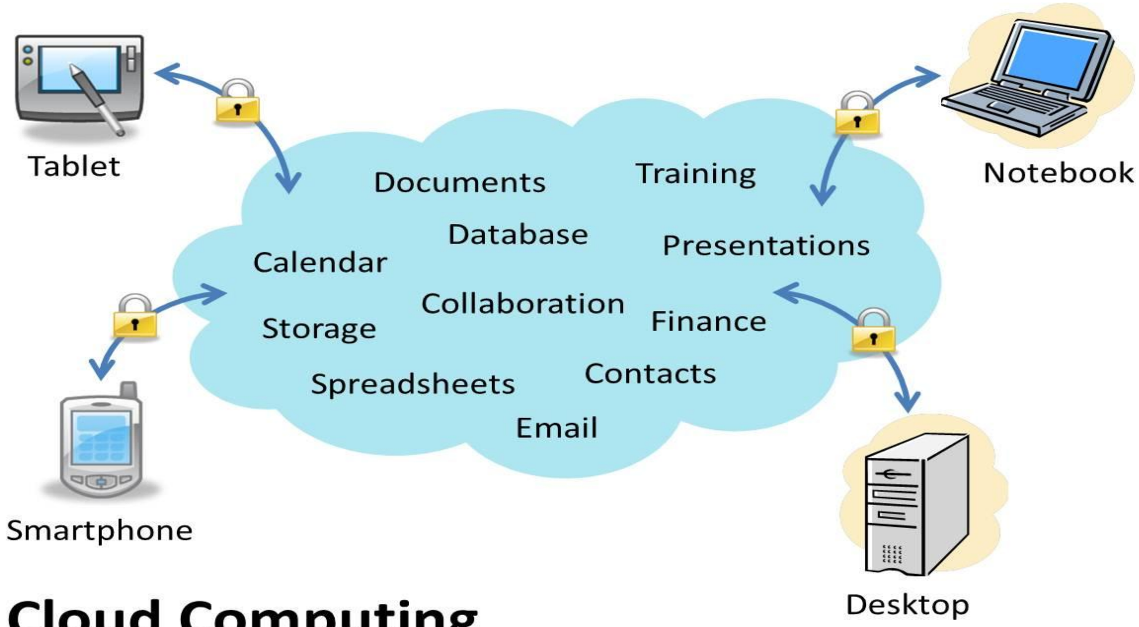
Virtualization windows server for Azure

Virtualization is a proven software technology that makes it possible to run multiple operating systems and applications on the same server at the same time. It's transforming the IT landscape and fundamentally changing the way that people utilize technology. Virtualization can increase IT agility, flexibility, and scalability while creating significant cost savings. Workloads get deployed faster, performance and availability increases and operations become automated, resulting in IT that's simpler to manage and less costly to own and operate.

Benefits:
Reduce capital and operating costs.
Deliver high application availability.
Minimize or eliminate downtime.
Increase IT productivity, efficiency, agility and responsiveness.
Speed and simplify application and resource provisioning.
Support business continuity and disaster recovery.
Enable centralized management.
Build a true Software-Defined Data Center.

Refer the figure at page.58 in the prescribed text book.

Host partition, also called the parent partition, is dedicated to running the Host OS. In Hyper-V v1, the host partition is the root (boot) partition and there can be only one host partition. *Host OS* is a lightweight server operating system (Windows Server 2008 Core for Azure) controls access to the hardware of the underlying server, and provides a mechanism for other guest VMs (where our customers applications are deployed) to safely communicate with the outside world. *Guest partitions*, also called child partitions, are created and owned by the host OS and are dedicated to running guest OSes. *Guest OS* is a server operating system for applications and services (Windows Server 2008 Enterprise with IIS 7, .NET Fx 3.5, and other extensions for Azure). *Services* are custom-written (Azure) applications and services that run on the guest OS. Child partitions do not have direct access to hardware resources, but instead have a virtual view of the resources, in terms of virtual devices. Any request to the virtual devices is redirected via the VMBus to the devices in the parent partition, which will manage the requests. The VMBus is a logical channel which enables inter-partition communication. The response is also redirected via the VMBus. If the devices in the parent partition are also virtual devices, it will be redirected further until it reaches the parent partition, where it will gain access to the physical devices. Parent partitions run a Virtualization Service Provider (VSP), which connects to the VMBus and handles device access requests from child partitions. Child partition virtual devices internally run a Virtualization Service Client (VSC), which redirect the request to VSPs in the parent partition via the VMBus. This entire process is transparent to the guest OS. *NIC*s are physical network interface card(s).*CPU*s are physical central processing units, which have one or usually more cores. *Disk*(s) are the physical fixed disk(s) for the root and guest partitions.



Cloud Computing

Having secure access to all your applications and data from any network device

UNIT - 3

Barriers of Cloud Computing

- High Speed Internet Connectivity
- Latency
- Consistency
- Availability
- Security

  Live ID, Card Space, SSL
- Interoperability

  Data Lock-in , Standardization

Maximizing Data Availability and Minimizing Security Risks

30 seconds **downtime** per month

Microsoft announced in mid-July 2009 that its SLA would cover 99.95% **uptime** guarantee for two or more Azure service instances and 99.9% **availability** for storage services

IT-Related Risk

Unauthorized (malicious or accidental) disclosure, modification, or destruction of information.

Failure to exercise due care and **diligence** in the implementation and operation of the **IT system**.

laws

information security in enterprises

Security in Health Information Technology

Payment Card Industry-Data Security Standard (PCC-DSS)

Implementing the Secure Sockets Layers Transmission

https://www.youtube.com/watch?v=4nGrOpo0Cuc

Encryption for Web Roles- Encrypting Personal Information in Azure Storage Services : Encrypting Strings with AES

```
public static string Encrypt(string input)
{
try
{
// Plaintext string input
string data = input;

// Convert to an array of UTF-8 bytes
byte[] utfData = UTF8Encoding.UTF8.GetBytes(data);
byte[] saltBytes = UTF8Encoding.UTF8.GetBytes("S0d1umChl0r1de");

// Use the PBKDF2 standard for password-based key generation
Rfc2898DeriveBytes rfc = new Rfc2898DeriveBytes("K3yPassw0rd!", saltBytes);

// Advanced Encryption Standard symmetric encryption algorithm
AesManaged aes = new AesManaged();
```

```
// Set AES parameters
aes.BlockSize = aes.LegalBlockSizes[0].MaxSize;
aes.KeySize = aes.LegalKeySizes[0].MaxSize;
aes.Key = rfc.GetBytes(aes.KeySize / 8);
aes.IV = rfc.GetBytes(aes.BlockSize / 8);

// Encryption
ICryptoTransform encryptTransf = aes.CreateEncryptor();
// Output stream, can be also a FileStream
MemoryStream encryptStream = new MemoryStream();
CryptoStream encryptor =
new CryptoStream(encryptStream, encryptTransf,
CryptoStreamMode.Write);
```

```
// Write, flush, clear and close the encryptor
encryptor.Write(utfData, 0, utfData.Length);   Unicode Transformation
Format
encryptor.Flush();
encryptor.Clear();
encryptor.Close();

// Create a byte array and convert it to a
https://www.youtube.com/watch?v=4nGrOpo0Cuc
 string
byte[] encryptBytes = encryptStream.ToArray();
string encryptedString = Convert.ToBase64String(encryptBytes);
return encryptedString;
} }
```

https://en.wikipedia.org/wiki/Base64

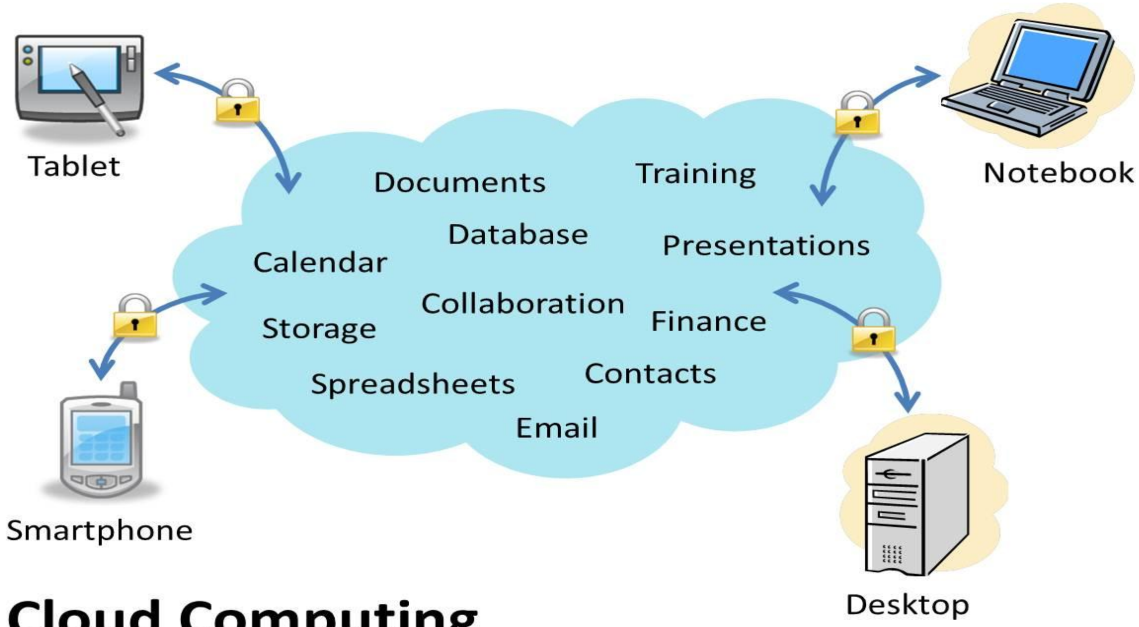
Decrypting Strings with AES

```
public static string Decrypt(string base64Input)
{
Try {
//byte[] encryptBytes = UTF8Encoding.UTF8.GetBytes(input);
byte[] encryptBytes = Convert.FromBase64String(base64Input);
byte[] saltBytes = UTF8Encoding.UTF8.GetBytes("S0d1umChl0r1de");
// Use the PBKDF2 standard for password-based key generation
Rfc2898DeriveBytes rfc = new Rfc2898DeriveBytes("K3yPassw0rd!", saltBytes);
// Advanced Encryption Standard symmetric encryption algorithm
AesManaged aes = new AesManaged();

// Set AES parameters
aes.BlockSize = aes.LegalBlockSizes[0].MaxSize;
aes.KeySize = aes.LegalKeySizes[0].MaxSize;
aes.Key = rfc.GetBytes(aes.KeySize / 8);
aes.IV = rfc.GetBytes(aes.BlockSize / 8);
// Decryption
ICryptoTransform decryptTrans = aes.CreateDecryptor();
// Output stream, can be also a FileStream
MemoryStream decryptStream = new MemoryStream();
CryptoStream decryptor =
new CryptoStream(decryptStream, decryptTrans, CryptoStreamMode.Write);

// Write, flush, clear and close the encryptor
decryptor.Write(encryptBytes, 0, encryptBytes.Length);
decryptor.Flush();
decryptor.Clear();
decryptor.Close();

// Create UTF string from decrypted bytes
byte[] decryptBytes = decryptStream.ToArray();
string decryptedString =
UTF8Encoding.UTF8.GetString(decryptBytes, 0, decryptBytes.Length);
return decryptedString;   }
```



Cloud Computing

Having secure access to all your applications and data from any network device

UNIT - 4

Creating the .NET Services Solution.

Provision for

Access control services

Service Bus

workflow services

Installing the .NET Services SDK and other Tools

.NET Services SDK

AzureManagement Tools

- The Azure Services Management Tools provide a Microsoft Management Console (MMC) and Windows PowerShell cmdlets.

- enable users to configure and manage .NET Access Control Services,and the .NET Workflow Service.

Azure Services Training Kit

- The Azure Services Training Kit includes hands-on labs (HOLs), PowerPoint presentations, and demonstrations that are designed to help you learn how to use the Windows Azure Platform.

Card Space Credentials at Federated identity .Net - Managed Card Space Credential with ACS

Identity : passport, license (many identities based on context)

digital identity

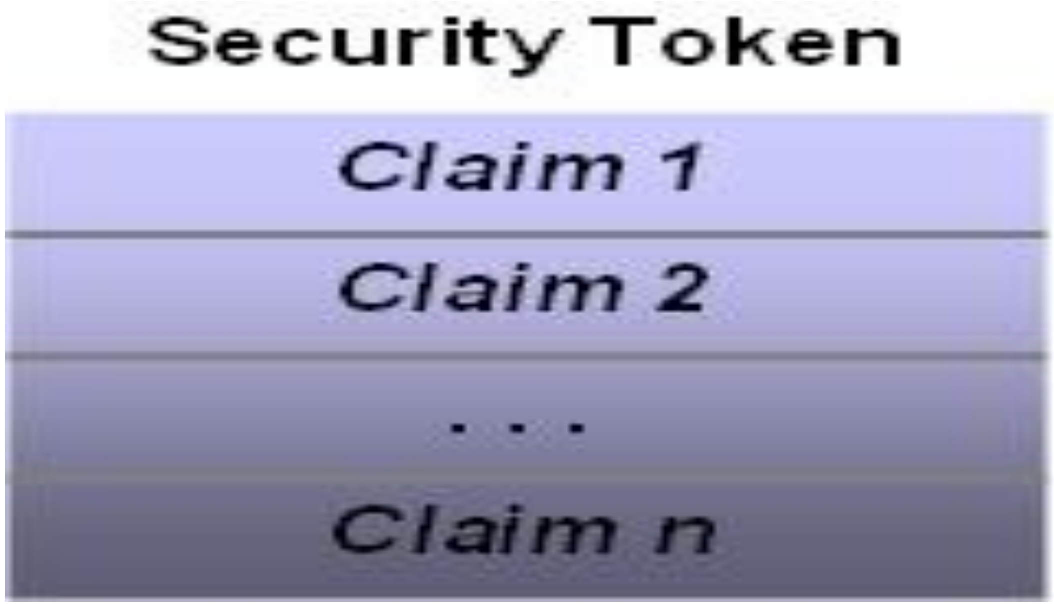
Microsoft identity meta system – windows cardspace (info card)

different digital identities in different contexts

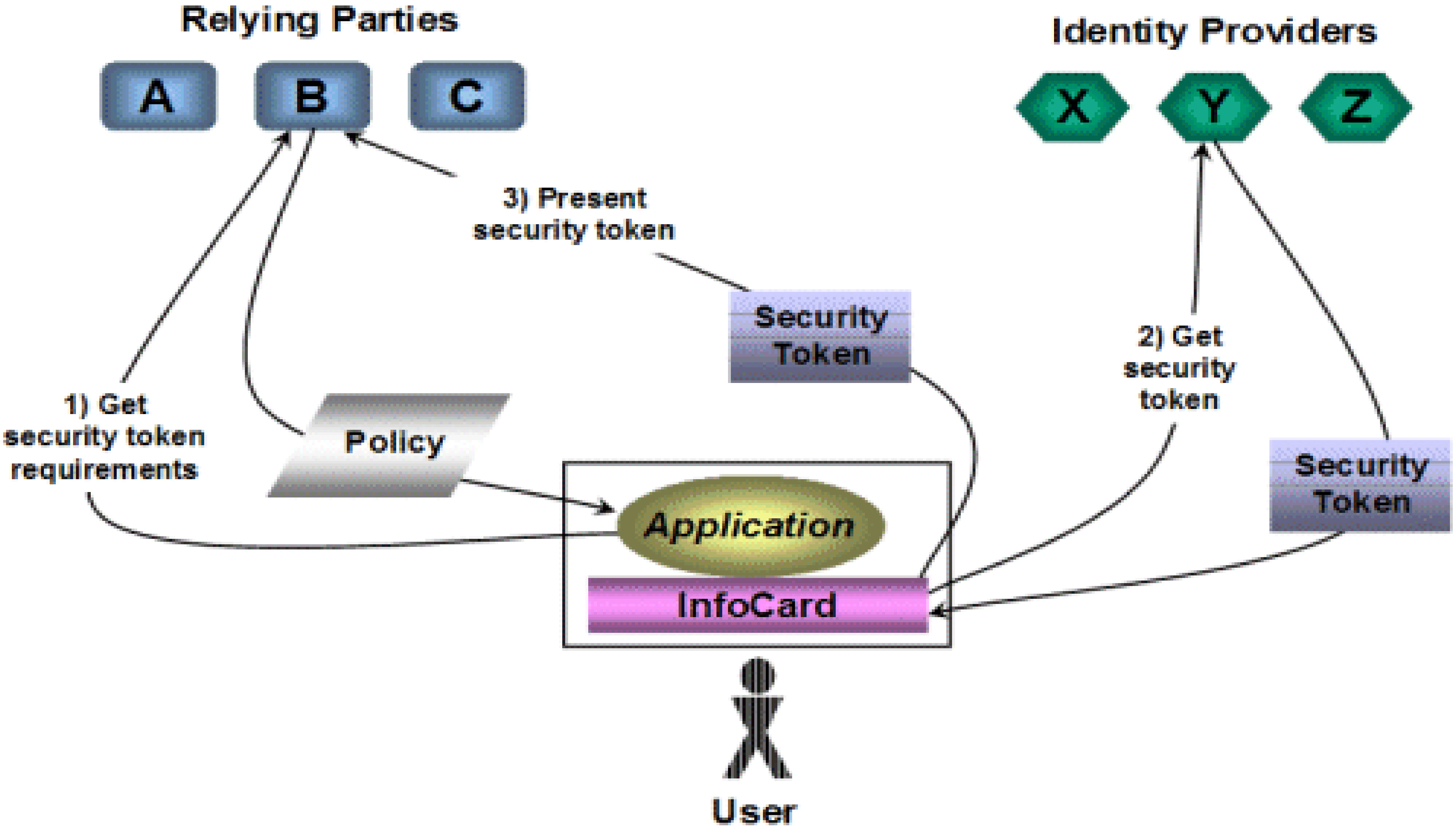
card number (identity) and username,password (identity)

Representing Digital Identity: Security Tokens

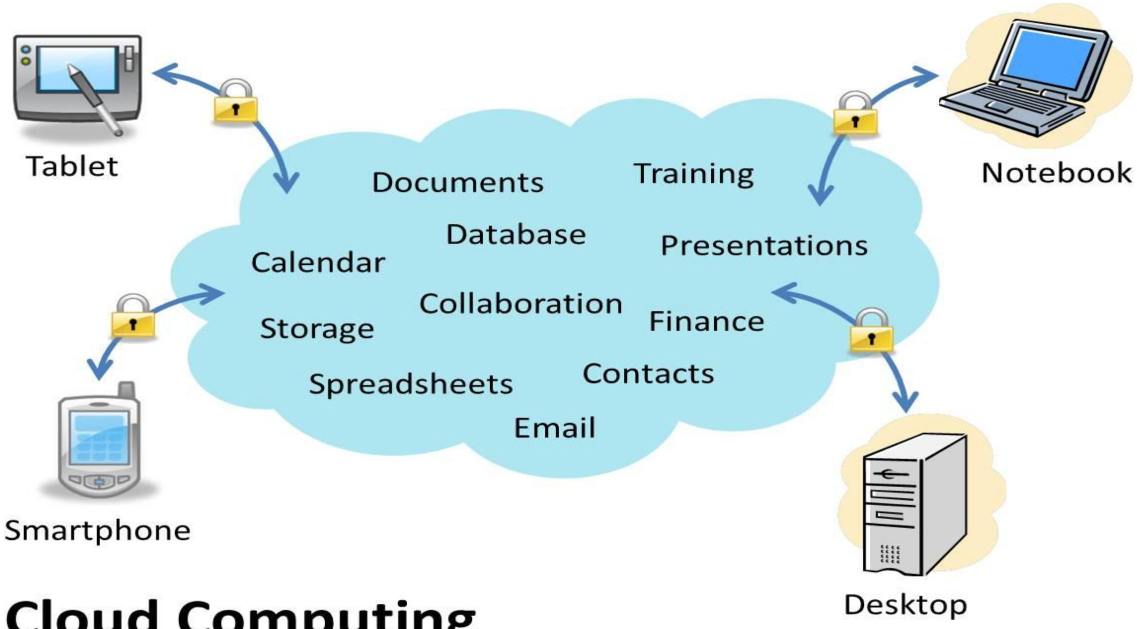
claims



security token is just a set of bytes that expresses information about a digital identity


Relying Parties
A
B
C
Identity Providers
X
Y
Z
3) Present security token
Security Token
2) Get security token
Security Token
1) Get security token requirements
Policy
Application
InfoCard
User

https://www.youtube.com/watch?v=JGDsOhGbgNg



Cloud Computing

Having secure access to all your applications and data from any network device

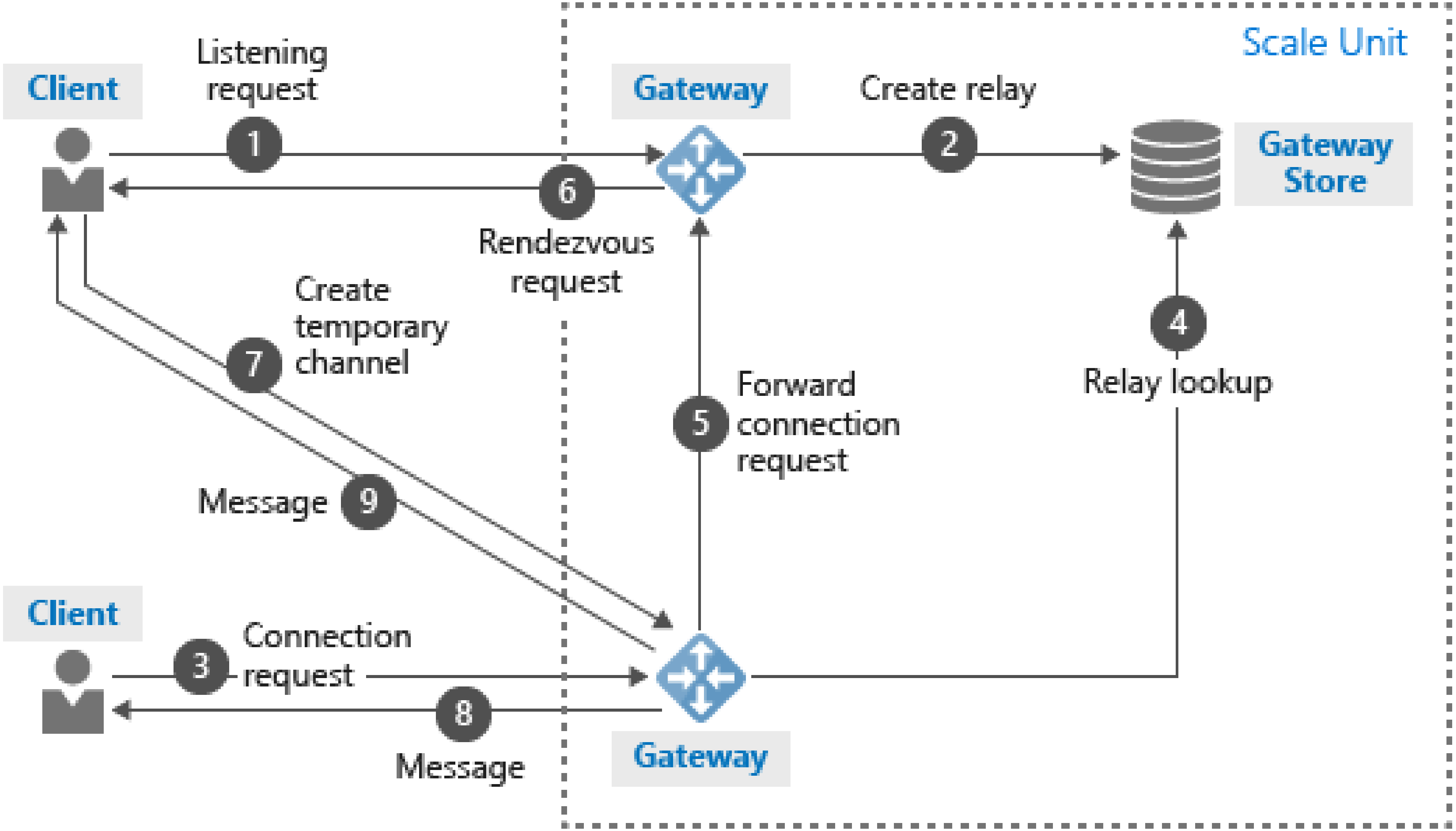
UNIT - 5

Relaying Message with SB

Service Bus allows communication between on-premises solutions to Microsoft Azure solutions and even Microsoft Azure solutions to other solutions within the cloud.

Relayed Messaging: It helps disparate applications and **services communicate through firewalls, NAT gateways, and other network boundaries**

Azure Service Bus Relay is conceptually like a router hosted in the cloud.


Scale Unit
Client
Listening request
1
Gateway
Create relay
2
Gateway Store
6
Rendezvous request
Create temporary channel
7
4
Relay lookup
5
Forward connection request
Message
9
Client
3
Connection request
8
Message
Gateway

Listening client sends a listening request to the Azure Relay service. The Azure load balancer routes the request to one of the gateway nodes.

The Azure Relay service creates a relay in the gateway store.

Sending client sends a request to connect to the listening service.

The gateway that receives the request looks up for the relay in the gateway store.

The gateway forwards the connection request to the right gateway mentioned in the gateway store.

The gateway sends a request to the listening client for it to create a temporary channel to the gateway node that's closest to the sending client.

The listening client creates a temporary channel to the gateway that's closest to the sending client. Now that the connection is established between clients via a gateway, the clients can exchange messages with each other.

The gateway forwards any messages from the listening client to the sending client.

The gateway forwards any messages from the sending client to the listening client.

The relay service supports the following scenarios between on-premises services and applications running in the cloud or in another on-premises environment.

- Traditional one-way, request/response, and peer-to-peer communication

- Event distribution at internet-scope to enable publish/subscribe scenarios

- Bi-directional and unbuffered socket communication across network boundaries.

Persisting Messages in Service- Bus Queues

Service Bus Queues (SBQs) provide a durable first-in, first-out (FIFO) data structure to which senders can add and from which listeners can retrieve messages.

The primary application for the **SB is acting as a transient relay** between senders and active listeners.

If the listener isn't active when the sender issues a message, the message is lost. Message loss also occurs if the sender issues messages faster than the listener can process them.

To overcome these issues, the .NET Services team introduced SB Queues and SB Routers in the .NET Services

https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-dotnet-get-started-with-queues

Diagram : refer book 326

Delivering Message with Service Bus Routers.

Service Bus Routers (SBRs) handle delivery of durable messages to all (multicast) or individual subscribers. Listeners, including queues, can subscribe to these messages.

Page no 337