Cloud Computing

Abdullah Albarrak, Andres Felipe Aguirre Quintero, Romaan Khadeer Ahmed, and Svetlana Nogina

EAIT, The University of Queensland,
Brisbane, QLD, Australia
{abdullah.albarrak, andres.aguirrequintero,
romaan.ahmed, svetlana.nogina}@uqconnect.edu.au

1 Introduction

This report discusses five different cloud computing inventions from major companies in the IT sector: IBM, EMC, Microsoft, Google and Amazon. The objective of this report is to illustrate the diverse implementation of the cloud services and how each company interprets the concepts of cloud computing. Firstly, a general background about cloud computing is provided. It demonstrates the basic components and architecture of the cloud computing environments and introduces new terms. Then, the cloud application from IBM is presented with features and details about two of its cloud services. Next, the report demonstrates the cloud storage solution from EMC and how it's structured. Then, the report provides a brief discussion of the cloud computing service Windows Azure from the major software company Microsoft. Afterwards, the report introduces the cloud service from Google, Google App Engine, which focuses on facilitating the programming tasks. Next, the report is concluded by a brief overview for Amazon cloud computing solution. Finally, a conclusion is given to shortly summarize the report.

2 Background

Cloud computing is a computing model introduced to utilize distributed computing resources and to reach higher throughput from these resources [RCL09]. The major customers of such service are enterprises customers with large amount of data to process and store and the research sector where large scale calculations and processing are required. Cloud computing services are required to provide fault tolerance, scalability, load balancing and security as standard services.

For now, there is no clear classification for the cloud services. However, the major IT firms relativity agree on categorizing their cloud services into three categories: *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) and *Software as a Service* (SaaS). In Infrastructure as a Service, the hardware and storage are provided as a service and the customer has to install all necessary software. Platform as a service is a higher layer that sits above the IaaS layer and uses the underlying layer service [Cha10]. It provides fully geared development platform for developers to ease their job. The software as a service category contains any application being provided through the web protocols or interfaces [Siia01]. Its key advantage persists on the unburden of installing, upgrading and maintain the software.

There are three deployment modes for cloud computing: *public cloud*, *private cloud*, and *hybrid cloud*. The most common one is the public mode. The cloud is hosted off-site and can be reached through the Internet only. The private mode is the least common mode. Many experts do not consider it as a cloud service. It basically means the services are hosted within the company's data center. The last mode is the hybrid mode where part of the services is hosted off-site and the rest are hosted on-site [Siia01].

3 IBM

IBM offers Cloud solutions for Enterprise, including IaaS, SaaS, and PaaS systems with public, private and hybrid cloud delivery models. The IBM cloud computing product, Cloud Burst is presented below with a brief description of features, architecture and their advantages.

3.1 Cloud Burst

IBM CloudBurst includes hardware, software, and services to help grow private cloud environment. Built on the IBM Blade Center platform, IBM CloudBurst provides pre-installed, fully integrated services management across hardware, middleware and applications[LM12]. CloudBurst is a prepackaged and self-contained service delivery platform that can be easily and quickly implemented in a data center environment. It provides capabilities essential to a cloud model. Some of the features of CloudBurst are fully integrated hardware and software platform with networking and storage for fast roll out, a self-service portal interface for reservation of computing, storage, and networking resources, including virtualized resources, automated provisioning and de-provisioning of resources, prepackaged automation templates and workflows for most common resource types, such as VMware virtual machines, Service management for cloud computing, Real time monitoring, Backup and recovery, Integrated usage and accounting chargeback to track and optimize system usage, facility administrators with energy monitoring and management to optimize energy consumption for higher efficiency and finally supports ability to manager other heterogeneous resources outside the IBM [LM12]. The basic hardware conceptual view that makes up IBM CloudBurst is presented in the figure 1.

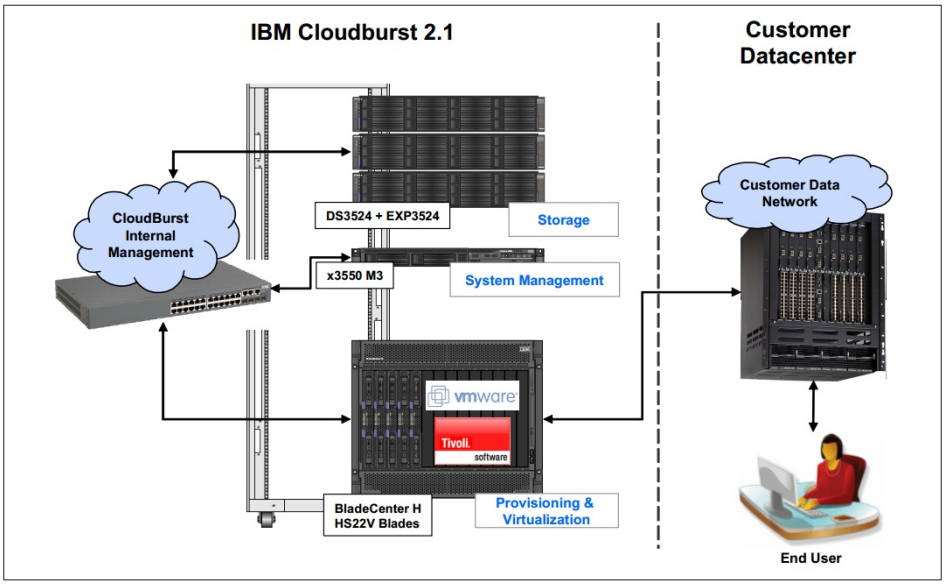

Fig. 1. IBM CloudBurst[LM12]

The IBM CloudBurst for System X provides software solution for service management automation in a virtual data center environment. It is delivered as an

appliance containing a pre-integrated software stack that is deployed as a set of virtual images. This automates IT service deployment and provides resource monitoring, cost management, and provisioning of services in a cloud, enabling the data center to accelerate the creation of service platforms for a wide spectrum of workload types. It provides a high degree of integration, flexibility, and resource optimization.

4 EMC

EMC launched the product Atmos, which is a multi-petabyte offering for information storage and distribution. Atmos allows to build Cloud Optimized Storage by combining scalability with automated data placement to deliver content and information services anywhere in the world [Weh11].

This product delivers a more modern technology than traditional file or block based constructs. It delivers the linear scale, elasticity, geographic distribution of unstructured data that eliminates the repeated need to provision and consume storage capacity. It makes the capacity available based on demand and enables users accessing an entire infrastructure collectively rather than in a frame by frame monolithic approach common with file or block based storage technologies [Weh11]. It also solves the constraints of rigid, mount point based interaction between storage and consumer by presenting a singular access point to the entire storage infrastructure. This enables storage to be automatically distributed across the infrastructure in an active/active everywhere methodology.

Next, to streamline the delivery of storage as a service on a shared infrastructure, Atmos Cloud Delivery Platform leverage a built in multi tenancy model where all users and consumers of storage can be logically isolated from one another without any additional software or complete silos of infrastructure to provide isolation as has been the case with legacy storage technology.

It is a cloud storage platform that enables enterprises and service providers to store, manage and protect globally distributed, unstructured content at scale. It is a global information management solution designed to automate and manage data placement. It also provides protection and access for rich, unstructured content as a single system across distributed storage environments. Atmos operates as a single entity, regardless of how it is physically distributed, which distributes content in an active/active paradigm rather than in an hierarchical approach common with file system based structures. It uses customizable, value driven metadata to drive storage placement, protection and life cycle policies. This ensures information gets to the right location, at the right time - automatically. Atmos can operate as the foundation of a Cloud infrastructure, natively serving and metering isolated tenants (multi-tenancy) from a single system to maximize utilization across multiple customers and applications. The EMC Atmos service architecture is shown in the figure 2. It provides the hardware, software and the access layer to applications or operating systems. Thus EMC Atmos helps to manage more efficiently distributed unstructured big data in the cloud by providing single system access across entire storage environment.

5 Windows Azure

For Microsoft, cloud computing is leading to a transformation on the industry making necessary to provide IT as a service, in this way customers can focus more on the business itself and leave the infrastructure issues to the service provider. Microsoft provides every aspect of IT as a service (SaaS, PaaS, and IaaS) [Mic10]. For achieving Cloud Architecture benefits, the following features should be provided:

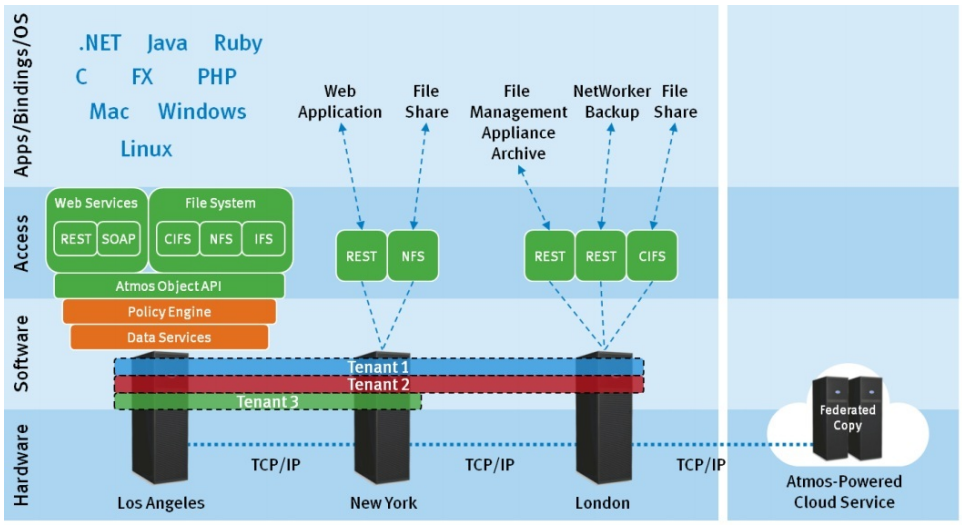

Fig. 2. EMC Atmos Service Architecture[Weh11]

- A virtualized compute fabric, where a hypervisor can manage and assign resources dynamically.
- A virtualized network that groups virtual machines and isolate them from unwanted traffic.
- Software, Data and Node State are viewed as cache.
- One node state can be reconstructed in a different one.
- Automation for OS and application updates, automatic failure handling and recovery.
- Automate application life cycle management.
- Rich platform services accessible online.

Windows Azure is Microsoft PaaS offering, the main components of this platform are Windows Azure and SQL Azure. The first component takes away much of the infrastructure efforts to run the applications, allowing developers to focus on building them; it also provides virtualized compute and network infrastructure, each server contains multiple virtual machines where one of them is the manager controling the incoming and outgoing traffic. This service grants compute, storage, networking, and middleware capabilities to manage applications on the internet located physically on Microsoft data centers. It is possible to use the standard Microsoft programming tools and development environments or any other programming languages.

SQL Azure is a cloud based relational database, developers can use their existing knowledge on T-SQL and use the same tools for developing and managing that they use for on-site databases.

Windows Azure introduces a new architecture component called the *fabric controller* (FC), that has two main functions:

- Managing the hardware, configuring the network, reacting and recovering of hardware failures.
- Managing a repository that contains the configurations and resources needed per application, then it maps the resources according to the needs, this repository is highly redundant and is based on developers models containing the topology, required software, and configuration settings for each client.

Windows Azure allows you to make changes on your application and publishing it without having to stop the service. Microsoft also provides development and

management tools, database, storage, monitoring and logging services, metering and billing services, and a rich developer environment with support from Visual Studio and Eclipse [Kha11]. Windows Azure offers the option to move just some parts of the application and keep the rest on the site, some applications where it can be adopted are:

- Adding web functionality to existing business databases and applications.
- Offer short term affordable capacity for big applications.
- Easily migrate databases into the cloud.
- High Performance Computing applications by using pools of computing power
- Efficient storage and distribution of massive amounts of data for network users

The following table 1 lists the advantages and some known limitations of Windows Azure.

Advantages	Disadvantages
Quickly scale up applications for peaks of demand	Security issues in access to data and missing true deletion
No infrastructure expertise needed	Governance data protection issues, right now you can not specify the location of your data
Compatible with existing .net skills	Poor debugging and logging support
No hardware compatibility issues	The platform is few tested compared to the Google and Amazon Cloud Computing products.
	Pricing changes unexpected over time
	Microsoft could terminate the service in the future

Table 1. Advantages and Disadvantages of Windows Azure [Mac10]

6 Google AppEngine

App Engine is a cloud-platform designed by Google for hosting real-time dynamic applications supporting many simultaneous users. It provides auto-scalable implementations of technologies important for web services, such as messaging, key-value data storage, multi-tasking, web server support, elasticity, and resource management. The infrastructure automatically adjusts number of application instances to the request rate [MKWB11]. Resources and services are available on a "pay-per-use", on a free or low-cost rental basis [San11].

The main components of App Engine are the runtime environment, the datastore, and the scalable services. The abstract structure of App Engine is illustrated in the figure 3. The following sections give an overview of the single parts.

6.1 Runtime environment

Google App Engine provides two possible runtime environments for applications: a Java environment and a Python environment. All requests are routed to an app server, which runs the application. The server handles the request and returns the response to the client. The interpreter (the JVM or the Python interpreter) is executed with sandbox restrictions, such that any attempt to use a feature of the language or a library that would require access outside of the sandbox fails with an exception.

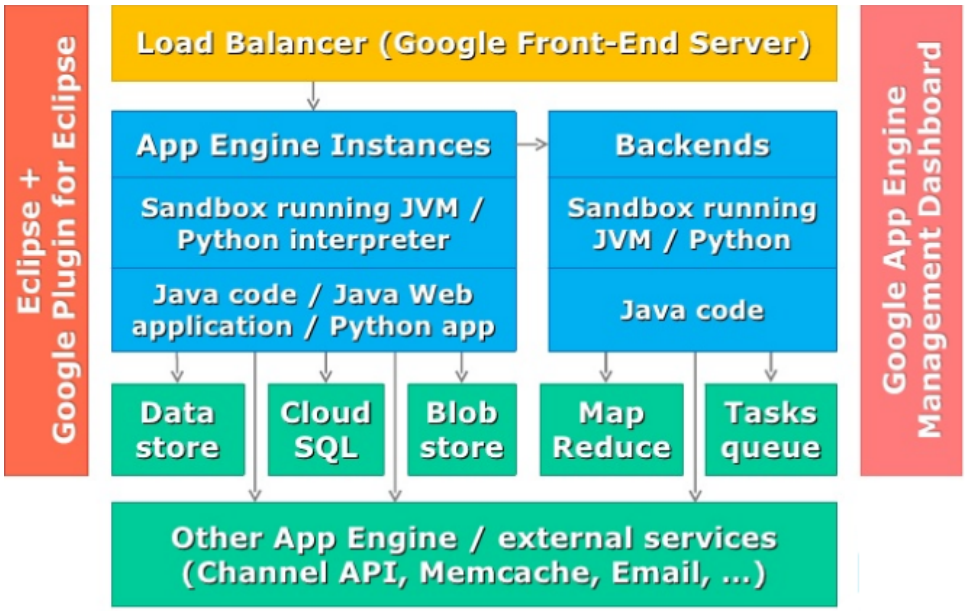

Fig. 3. Google App Engine [Nak12]

6.2 Datastore

Google App Engine's uses the schemaless NoSQL object database. It differs from join-query relational databases in multiple ways. An App Engine application stores its data as one or more datastore entities with a number of properties like name and value. Every datastore entity has a unique key, which is ,unlike in a relational database, not a property, but an independent parameter of the entity, allowing you to fetch an entity or perform queries on key values. An entity's key cannot be changed after the entity has been created. An entity is categorized by its kind, which is used to efficiently perform queries. Entities of the same kind can have different properties. Even if two entities of the same kind have the same properties, the values saved in the properties can be of different types. In this way, datastore entities are schemaless, which on the one hand provides powerful flexibility, on the other hand however causes some maintenance challenges.

Transactions An update of a single entity occurs in an atomic transaction. Reading requests never fail due concurrency, if the requested entity is being updated. Instead the most recent stable state is returned. In most cases, retrying a transaction on a contested entity will succeed. Concurrency failures become more likely with an increasing number of users.

6.3 Services

App Engine includes several self-scaling services.

Memory Cache Service The memory cache (or *memcache*) is a short-term key-value storage service. Its main advantage over the datastore is that it is much faster than the datastore for simple storage and retrieval. The memcache stores values in memory instead of on disk for faster access. It is distributed like the datastore, so

every request sees the same set of keys and values. However, it is not persistent like the datastore: if a server goes down, such as during a power failure, memory is erased. It also has a more limited sense of atomicity and transactionality than the datastore.

Mail Service Emails can be used to send notifications and confirmations from applications to users. Applications are also able to receive emails as HTTP requests.

6.4 Task Queues

Task Queues allow scheduling of jobs outside the scope of the web request. Tasks are performed in parallel on multiple application servers. Assigning jobs to multiple servers according to the current load greatly improves scalability and load balancing. Queues ensure that every task gets done eventually and at least once. If a task fails, the queue retries the task until it succeeds. This may lead to re-executing of some tasks in case of a failure. Therefore the executed tasks are required to be idempotent. Tasks can also be executed on a regular schedule or at a specified time with no user interaction, which is useful for sending periodic notifications or maintenance activities[San11]. The task execution is completely transparent for the user. App Engine does not provide APIs for inter-task data exchange or any concurrency control [MKWB11].

6.5 Limitations

App Engine is targeted at web applications and many limitations concern therefore only general purpose applications. In some cases workarounds are possible, however it often reduces scalability and maintainability. The following list presents some of the disadvantages of App Engine.

- **No background processes** - App Engine does not host long-running background processes. Task queues and scheduled tasks can invoke request handlers outside of a user request, and can initiate batch processing to execute a series of small tasks.
- **Missing streaming connections** - App Engine does not support streaming or long-term connections. Delivering state updates to a client can be done using a polling technique, where the client regularly asks the application for updates. However polling is difficult to scale with the increasing number of users and is not appropriate for all applications.
- **Tools limitations** - Google App Engine requires application developers to use only a limited set of supported programming languages, APIs and frameworks which are mostly dedicated to Web applications [San11]. It is not possible to install new applications and acquire a full access to the running virtual machine.

7 Amazon Web Services

The American online retail sales giant, Amazon, offers a cloud platform called: Amazon Web Services, which covers two important aspects of IT: the computing aspect and the storage aspect. The Amazon Elastic compute cloud (Amazon EC2) is one of those services. It is a service to host server instances in the cloud that can be scaled up or down in any time [Ama12a], adding more flexibility for the end users or developers in using computing resources. Regarding the storage services, Amazon Simple Storage Service (Amazon S3) is their leading cloud storage service which Amazon itself uses to store its own data [Ama12b].

7.1 Amazon Elastic Compute Cloud

Amazon EC2 is classified as a PaaS architecture. It consists of server instances, which is called Amazon Machine Image (AMI). The developer has the choice to configure an image for specific applications or use the preconfigured image which includes vast number of operating systems, Databases, Application Servers, Content Management Applications and Business Intelligence Systems. Using the latter is recommended since it saves the end users' time and the users do not have to worry about the licenses for the software they are using. Realizing the different requirements of the compute cloud customers and how they use the cloud for diverse applications, Amazon provides various types of server instances ranging from High-Memory and High-CPU instances to Low-Memory and Low-CPU instances which cover most of the customers' needs. The Elastic Compute Cloud consists of heterogeneous hardware with miscellaneous performance power. Not to mention how fast the hardware is updated. Amazon wants to deliver a constant processing power that is independent of the hardware. That processing power is defined as Compute Unit (CU) [Ama12a]. This tactic aims to facilitate comparing the performance of different instances and calculating the cost of using the service. The problem of IPs dynamic allocation is present in Amazon Cloud EC2. Any instance is allocated a local and a public IP address and these addresses change once a reboot of that instance is done [Ama12c]. The aim is to map a public IP address for any instance to an Elastic IP address that does not change if the instance reboots or fails for any reason. Amazon's EC2 cloud provides five EIPs for one account and a change in mapping is reflected immediately. The key advantage of using the cloud is to fully utilize the processing resources. Amazon introduced the auto scaling feature to give its product more competent value among other products. The auto scaling can be done up or down, to keep up with any peaks of requests or to save money while ideal [Ama12a]. Moreover, load balancing is a key factor in a stable system. It is used in Amazon Cloud EC2 to mask any failure and to balance the load between server instances in one pool [Ama12a].

7.2 Amazon Simple Storage Service (Amazon S3)

One of the most desired services is to store data in the Internet to access it anytime and anywhere. Amazon provides an IaaS product which allows customers to store any sort of data in any format with high availability. Basically, data are stored in buckets. Those buckets have global addresses and can contain objects of five gigabytes maximum size [Gar08]. Because of the high availability and durability Amazon promises, the S3 is well suited for backup and recovery purposes [Ama12b]. Also it uses internet protocols to write and read data from the buckets, e.g., SOAP [Haz08]. Businesses-related information, like Amazon's online store data, are sensitive and require high standard of security measures when stored in Amazon's S3 cloud. Thus, support for user authentication and data integrity is implemented in S3. Access Control Lists (ACL) is used to control permissions on who can view any data objects [Ama12b]. Also MD5 is used to check the integrity of the written or read data [Gar08].

7.3 Limitations

Amazon S3 service had several failures that lasted for almost 2 hours. The root of the problem was a failure in the load balancer which caused the resources to be unavailable to many customers over the internet [Gar08]. Moreover, there are limitations in controlling objects within buckets in S3 cloud. It does not support operations to rename, copy or move an objects directly. Instead, a combination of basic primitives must be done in order to perform one of these operations [Gar08].

8 Conclusion

In conclusion, the IT industry designs the cloud computing services to serve the customers and they rarely agree on the architecture of the cloud. Five different cloud services from leading IT companies are discussed in this report which reflects how these companies understand the key elements of cloud services differently.

Offerings range from budget solutions suitable for start-up companies to comprehensive custom build infrastructures. Important factors in the product choice are scalability, availability, level of control over the system as well as the pricing strategy. The PaaS App Engine has excellent scaling and high-availability mechanisms, but lacks flexibility. The IaaS Service EC2 allows the user the full control over the virtual machine, but it requires also much more sophisticated approaches to provide scalability and fault tolerance, because the system processes are application-dependent.

Some of the problems associated with cloud computing are shared by all solutions. One concern is the security of the data, located in the cloud. Another disadvantage is the instability of the performance, caused by I/O delays or scheduling [AFG+10].

On the whole the Cloud Computing remains a highly attractive field, providing elastic affordable resources for users and new business models for Cloud vendors.

References

AFG+10. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A View of Cloud Computing, 2010.

Ama12a. Amazon. Amazon Elastic Compute Cloud (Amazon EC2), 10.05.2012. `http://aws.amazon.com/ec2/`.

Ama12b. Amazon. Amazon Simple Storage Service (Amazon S3), 10.05.2012. `http://aws.amazon.com/es3/`.

Ama12c. Amazon. Feature Guide: Amazon EC2 Elastic IP Addresses, 10.05.2012. `http://aws.amazon.com/articles/1346`.

Cha10. D. Chambers. Windows Azure: Using Windows Azure's Service Bus to Solve Data Security Issues. July 2010.

Gar08. Simson L. Garfinkel. An Evaluation of Amazon's grid computing services EC2, S3 and SQS. Technical report, Harvard University, May 2008.

Haz08. Scott Hazelhurst. Scientific computing using virtual high-performance computing: a case study using the Amazon Elastic Computing Cloud. 2008.

Kha11. Y. Khalidi. Building a Cloud Computing Platform for New Possibilities. *IEEE Computer*, 44(3):29–34, 2011.

LM12. Armando Lemos and Rafael Molerio. IBM Cloud Burst on System X. April 2012.

Mac10. Alex Mackey. *Introducing .NET 4.0 With Visual Studio 2010.* APress, 2010.

Mic10. Microsoft. IT is a Service. Technical report, Microsoft, 2010.

MKWB11. M. Malawski, M. Kuzniar, P. Wojcik, and M. Bubak. How to Use Google App Engine for Free Computing. *Internet Computing, IEEE*, PP(99):1, 2011.

Nak12. Svetlin Nakov. Cloud for Developers: Azure vs. Google App Engine vs. Amazon vs. AppHarbo, 2012.

RCL09. Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A Taxonomy and Survey of Cloud Computing Systems. 2009.

San11. Dan Sanderson. *Programming Google App Engine.* O'Reilly, 2011.

Siia01. Software and information industry association. Software as a service strategic backgrounder. February 2001.

Weh11. Michael Wehle. EMC Atmos Cloud Storage Architecture. November 2011.