

November 15-18, 2010



Santa Clara Marriott
Santa Clara, CA

CDMI Cloud Clients

Rich Ramos
Individual

Agenda

- Definitions
- Additional Concepts
- Development Nuts and Bolts
- Example: iOS iPad Demo

November 15-18, 2010



Santa Clara Marriott
Santa Clara, CA

Definitions



Obligatory “What’s Cloud?”

- Remember: Always at least two points of view (usually more):
 - Cloud as a primary business model v.
 - Cloud as a technology
- Note: The Speaker acknowledges that there are multiple, even conflicting, definitions of Cloud. This tutorial is created from the point of view of a technology first. The audience should be aware that some information available today is not created from that point of view.

Cloud Storage Defined (SNIA)

- A cloud in networking conceptually represented any to any connectivity in a network, plus an ***abstraction*** of concerns of how connectivity and services are accomplished.
- Thus cloud storage is simply the delivery of virtualized storage on demand. The formal term we proposed for this is *Data Storage as a Service* (**DaaS**).
- ***Data Storage as a Service***
 - *Delivery over a network of appropriately configured virtual storage and related data services, based on a request for a given service level.*
 - Typically, DaaS hides limits to scalability, is either self-provisioned or provisionless and is billed based on consumption.

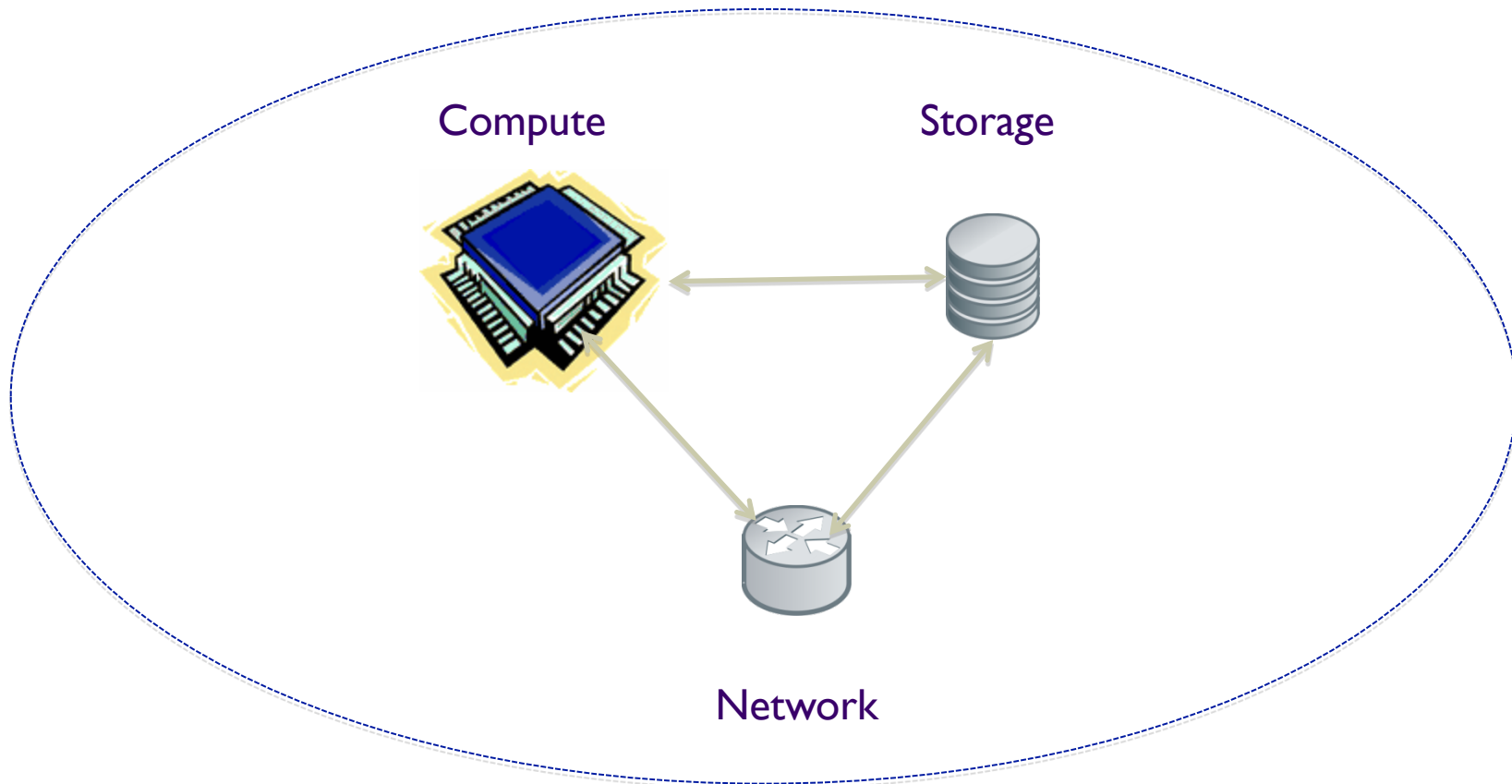
November 15-18, 2010



Santa Clara Marriott
Santa Clara, CA

(Additional) Concepts

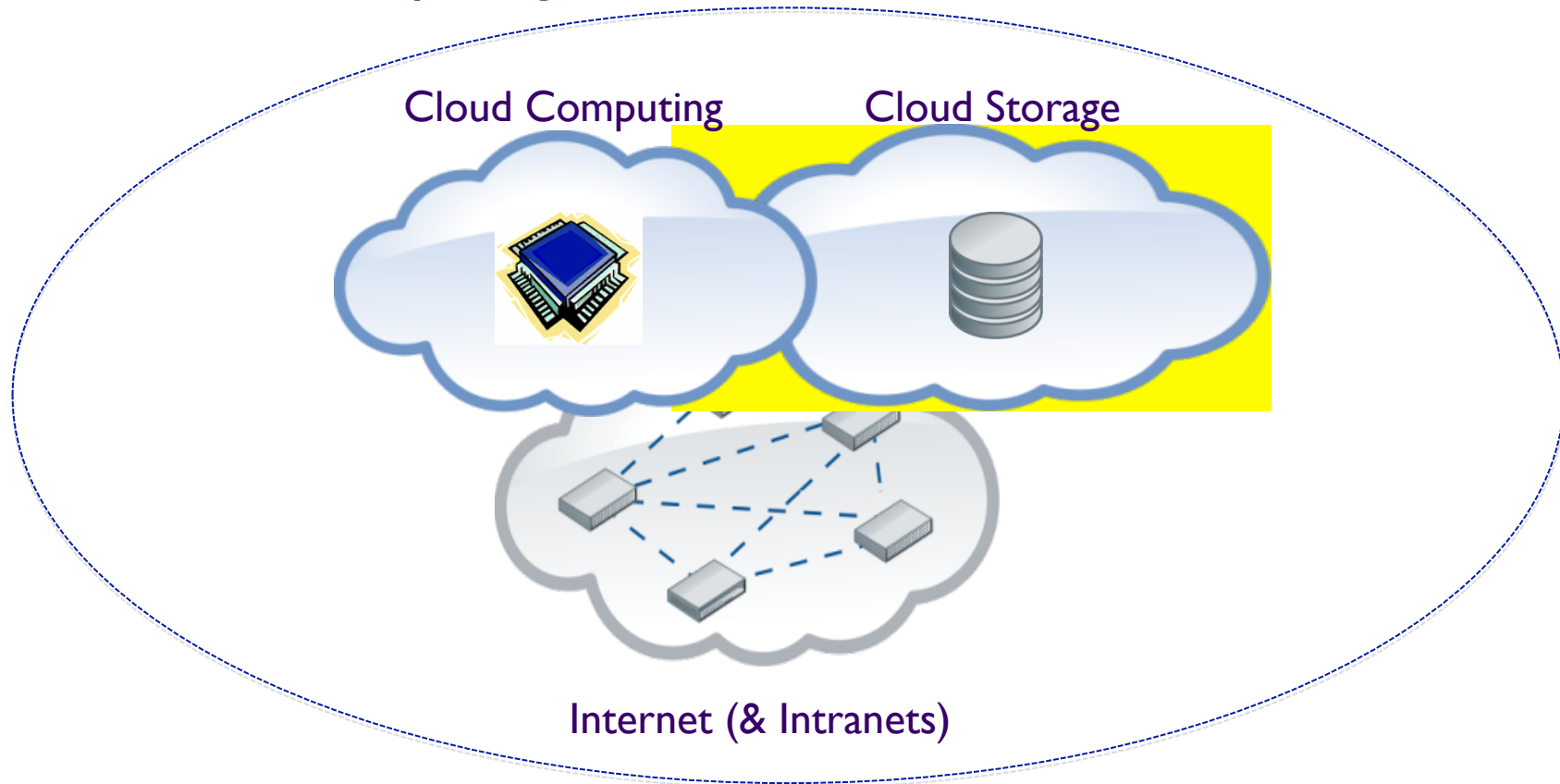
Generic Computing Environment



Basic building blocks of a modern computing environment, 3 legs

Roadmap for Concepts

Computing Environment “Cloud-ification”

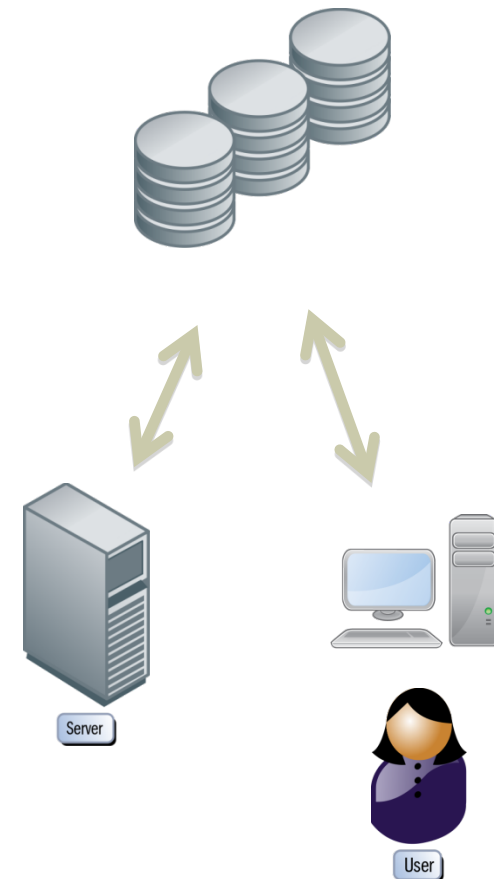


Cloud as an architecture & abstraction

“Traditional” Storage: Client/Server Model

Today’s Persistent Storage Paradigm...

- NAS: File-based
 - CIFS
 - NFS
- SAN: Block-based
 - FibreChannel
 - iSCSI
- Relational Database
 - SQL
- Notes:
 - Servers are very monolithic, even when clustering
 - Applicable to non-storage client server as well



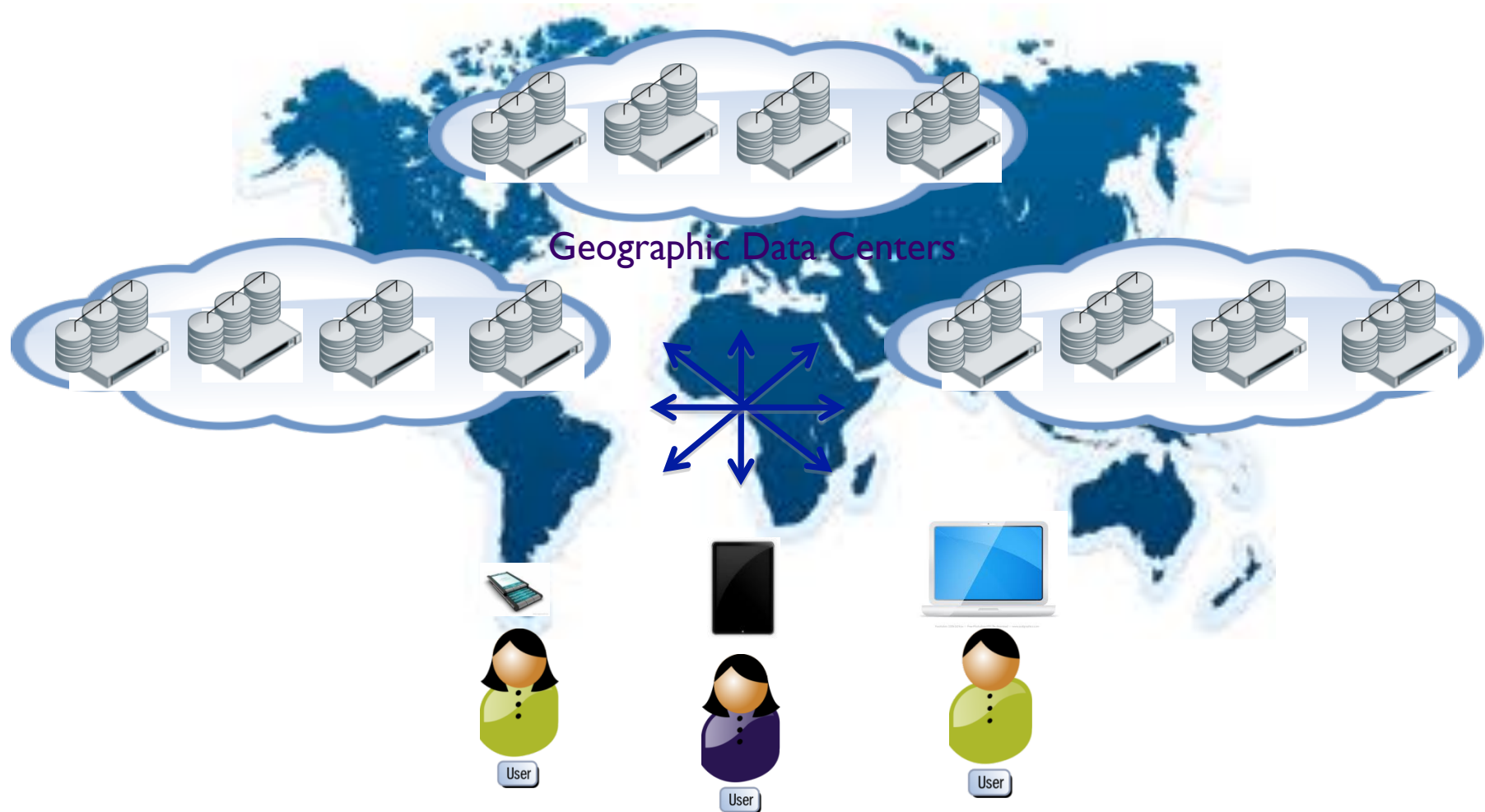


Rise of Non-Traditional Storage

- Factors
 - Mobile Workforce
 - Standardized x86 Hardware
 - Very High Capacity Disk Drives
 - The Web
 - Data Explosion
 - Moore's Law
- Results – New Applications for Storage
 - Many more server nodes: scale-out, not up
 - More disconnected client devices: laptops, smartphones, tablets
 - Everything online
 - Everything kept forever
 - Most data is fixed content & semi-structured

Results: "Big Data"

Cloud Storage






Non-Traditional Data Stores

- Web Service APIs Everywhere: Application Programmer Interfaces
- Examples: (many many out there)
 - Public: Amazon S3, Rackspace Cloud Files, Google GData, Box.net, Dropbox, Twitter API, flickr API, Facebook APIs
 - Private: Iron Mountain, Bycast, Caringo SCSP
 - List: <http://www.programmableweb.com> (2,346 APIs Nov 17, 2010)
- NOSQL: Not Only SQL
 - A correlated movement around non-relational databases
 - Store Types: Key Value, Column, Document, Graph
 - Many have REST-style APIs
- Notes:
 - In most cases, non-traditional stores **complement** traditional data stores.
 - Internet based Cloud **Services** get the most press, but there are many other types. “Cloud Storage Services” are a subset of Cloud Storage (Think Intranets)
 - Some companies are applying traditional APIs to problem, which implies traditional clients (NFS, CIFS, SQL, ...)

“Standards” Emerge

- RESTful Web Services (RESTful HTTP)
- Cloud Data Management Interface (CDMI)
- Java Script Object Notation (JSON)
- Notes:
 -  These are just a few, but most relevant to Cloud Storage
 - There is an entire tutorial on CDMI
 - (<http://www.snia.org/education/tutorials>)

REST

- **Representation State Transfer**
 - Started with [Dissertation by Roy Fielding](#) outlining the principles
- **Addressability**
 - Every object (resource) is addressable through a unique identifier
- **Uniform, Constrained Interface**
 - Use only HTTP verbs and model other semantics in the data model
 - Allows for Familiarity (low learning curve), Interoperability and Scalability
- **Representation Oriented**
 - Complexity is in the representations
- **Communicate Statelessly**
 - No persistent client-server connections, no



Cloud Storage Clients

- Characteristics
 - Hybrid: Web + Local (App-like, also Rich Internet Application)
 - RESTful HTTP
 - Disconnected Operations
 - Local Caching
 - Data Synchronization
 - Data as Objects with Metadata

Cloud Storage Clients (cont)

- Just a Few Examples (there are many more):
 - Mac & iPhone: Apple iDisk
 - Windows: Microsoft Live Mesh / Windows Live Sync
 - Linux: Ubuntu One
 - Firefox + Gears (Google Docs*)
 - Rich Media apps, iPad Magazines/Newspapers
 - Social Apps, Facebook, ...
- Notes:
 - Other Servers can also be cloud storage clients, however that use case is not the main topic of this tutorial.
 - Again, it's not always easy to separate compute from storage, hence overlap
 - * Temporarily disabled

November 15-18, 2010



Santa Clara Marriott
Santa Clara, CA

Development Nuts and Bolts

Platforms
Frameworks
Libraries
Tools



Development Platforms

- Determined by your “Camp”, which determines most other things...
- Close ties between; Platform, Framework, Library, Language, IDE
- Audience Poll, which do you use?
- Most common*
 - Apple iOS
 - Google Android
 - Microsoft: .Net, Silverlight
 - Adobe: Flash Suite
 - “Javas”: Java, JavaFX, Javascript, AJAX
 - Ruby (on Rails)**
 - Python**
 - HTML5**
 - PaaS: Google App Engine, Heroku, Engine Yard, Microsoft Azure
- Notes:
 - *there’s not a clean delineation, two main models: 1) server/client & 2) fat client
 - ** “platform’ish”



Development Frameworks

- Most hands on development will happen inside a framework
- Relevant Web Services Frameworks:
 - iOS*, Android*: Mobile platforms still evolving, but will be focus of future
 - Microsoft: .Net, Silverlight
 - Adobe: Flash/Flex
 - Java: JavaFX, Spring, AJAX
 - Ruby on Rails
 - Python: Django



SDKs, Libraries, Tools

- Examples
 - Language or Platform Specific
 - These are just a few, many others out there
- SDKs/Libraries
 - Web Service Specific: Dropbox, Box.net
 - 3rd Party: jclouds, Fog (Ruby), Simple Cloud (PHP), Deltacloud
 - iOS:
 - Apple's URL Loading System (roll your own)
 - ASIHTTPRequest
 - ObjectiveResource: Ruby on Rails specific
 - ShareKit: Extendable via plugins for new Web Services
- Tools
 - Client: FireFox HttpFox, Safari Developer
 - Cloud: apigee



Integrating CDMI

- Still very early days
- CDMI Server Reference Implementation:
 - PROTOTYPE / Example Only
 - Uses Spring Framework
- iOS/Cocoa Client Library
 - Goal to open source a minimal CDMI object creation client library
 - If interested contact me
- Remember: RESTful HTTP
 - Leverage Existing Stuff Already Listed

November 15-18, 2010



Santa Clara Marriott
Santa Clara, CA

Example iOS / iPad Client Demo

Summary

- Cloud & Cloud Storage technologies focused
- Emergence of non-traditional data stores
- RESTful HTTP Cloud Storage Clients matching non-traditional data stores
- Platform, Framework, Libraries, Language, IDE are interrelated, determined by rest of dev environment.
- Use RESTful Web Service Frameworks when possible
- CDMI integration still early days, more code coming

November 15-18, 2010



Santa Clara Marriott
Santa Clara, CA

Thanks!

Contact:

mandevconf@RichRamos.com