CDMI Clients

Rich Ramos
Individual

Agenda

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

Definitions

Obligatory "What's Cloud?"

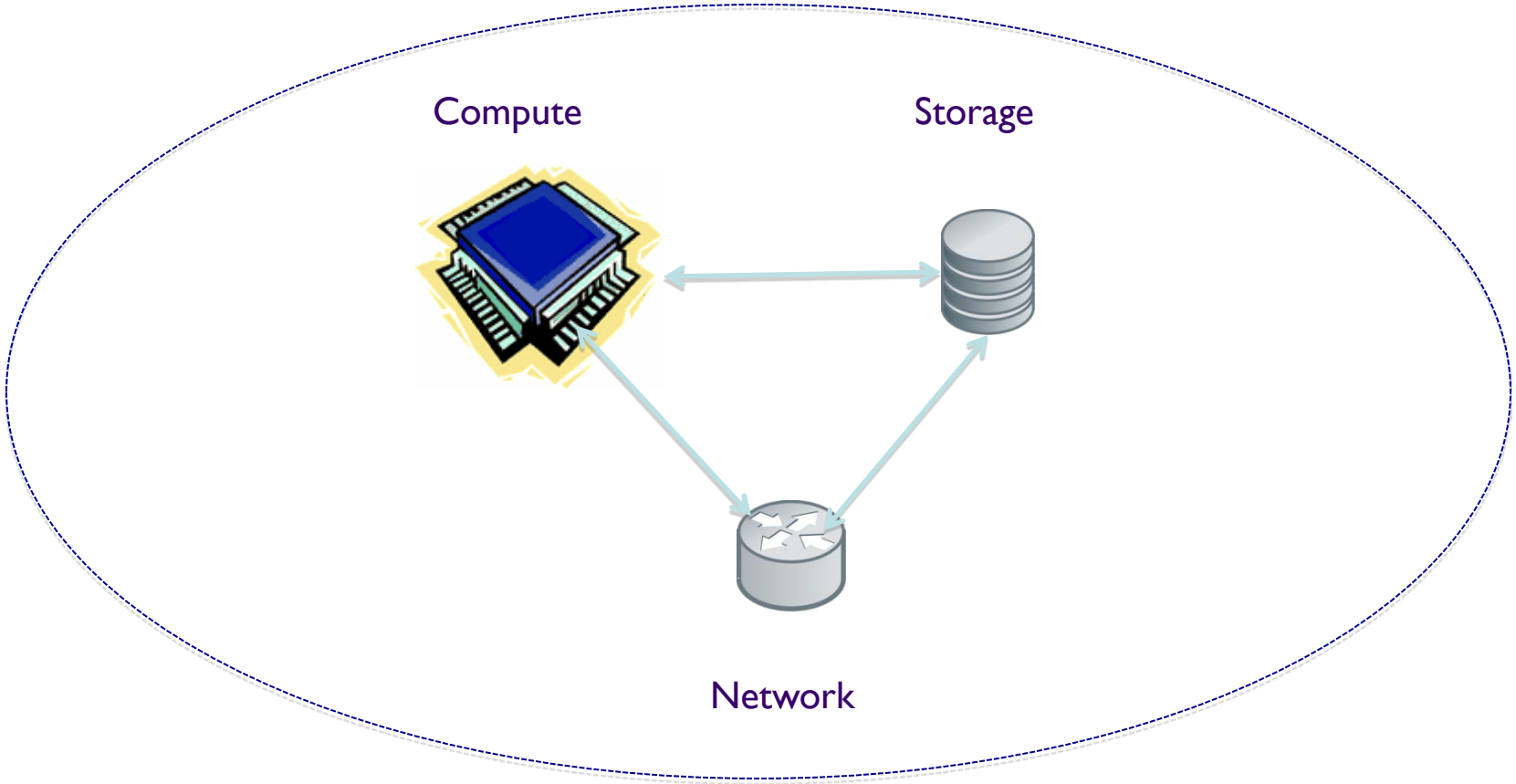
- Remember: Always at least two points of view (usually more):
  - Cloud as a primary business model v.
  - Cloud as a technology
- Note: The Speaker acknowledges that there are multiple, even conflicting, definitions of Cloud. This tutorial is created from the point of view of a technology first. The 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.

(Additional) Concepts

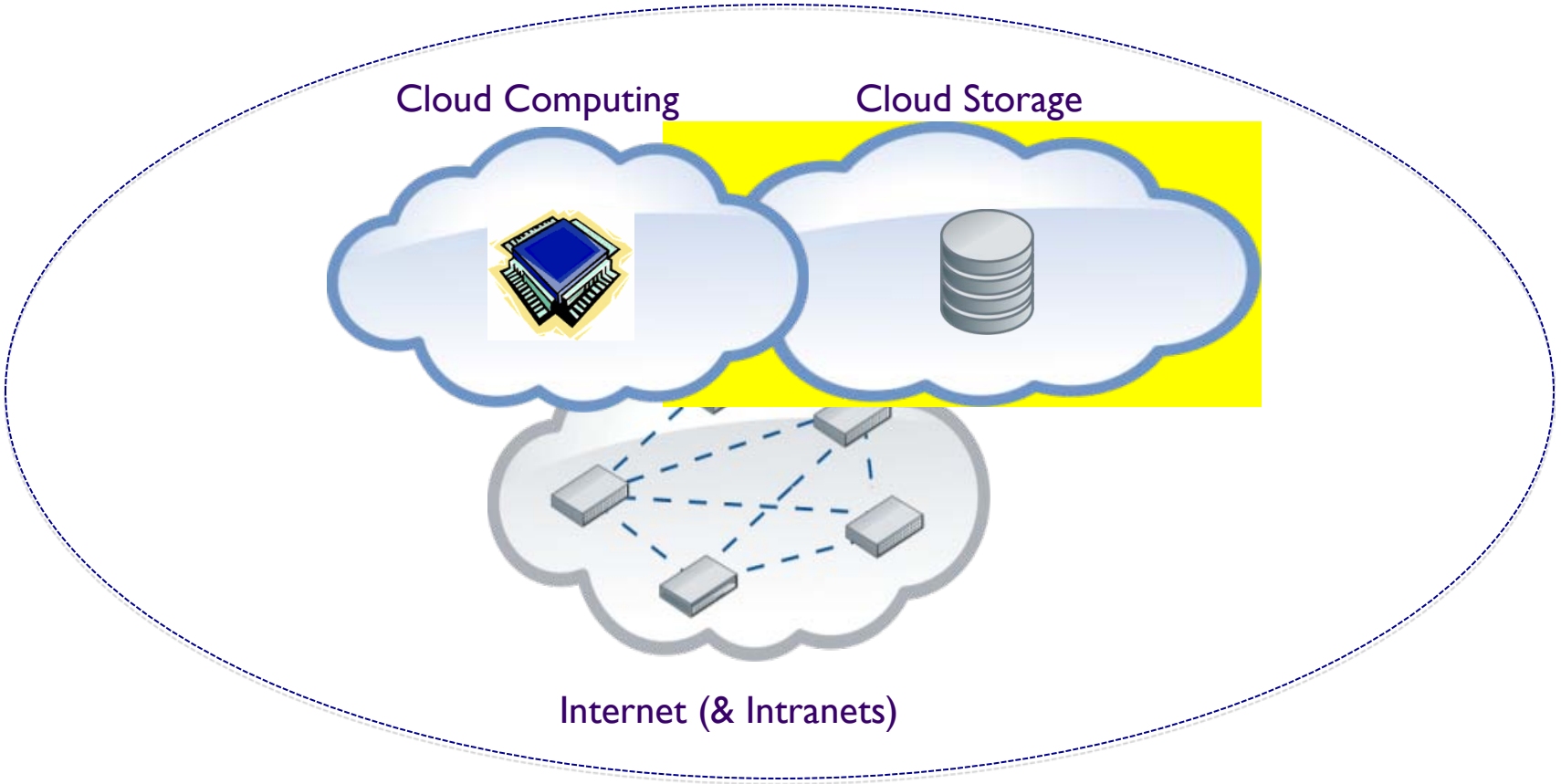
Generic Computing Environment



Basic building blocks of a modern computing environment, 3 legs

Roadmap for Concepts

Computing Environment "Cloud-ification"

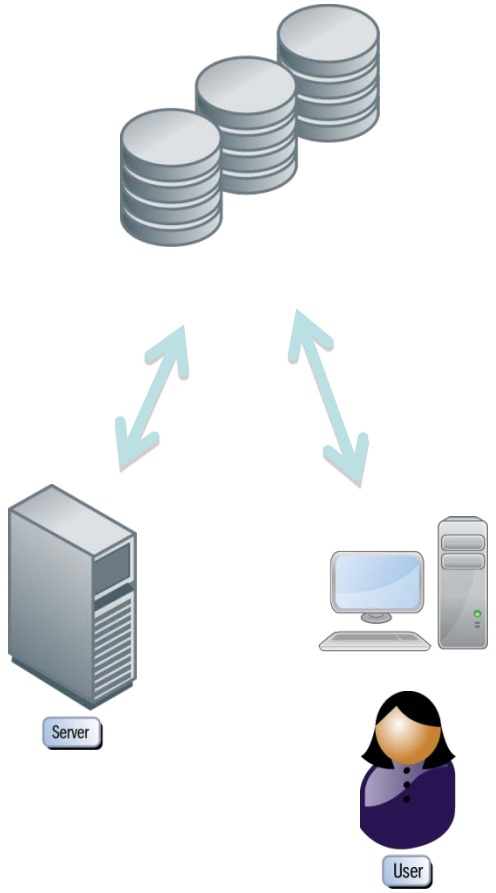


Cloud as an architecture & abstraction

"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

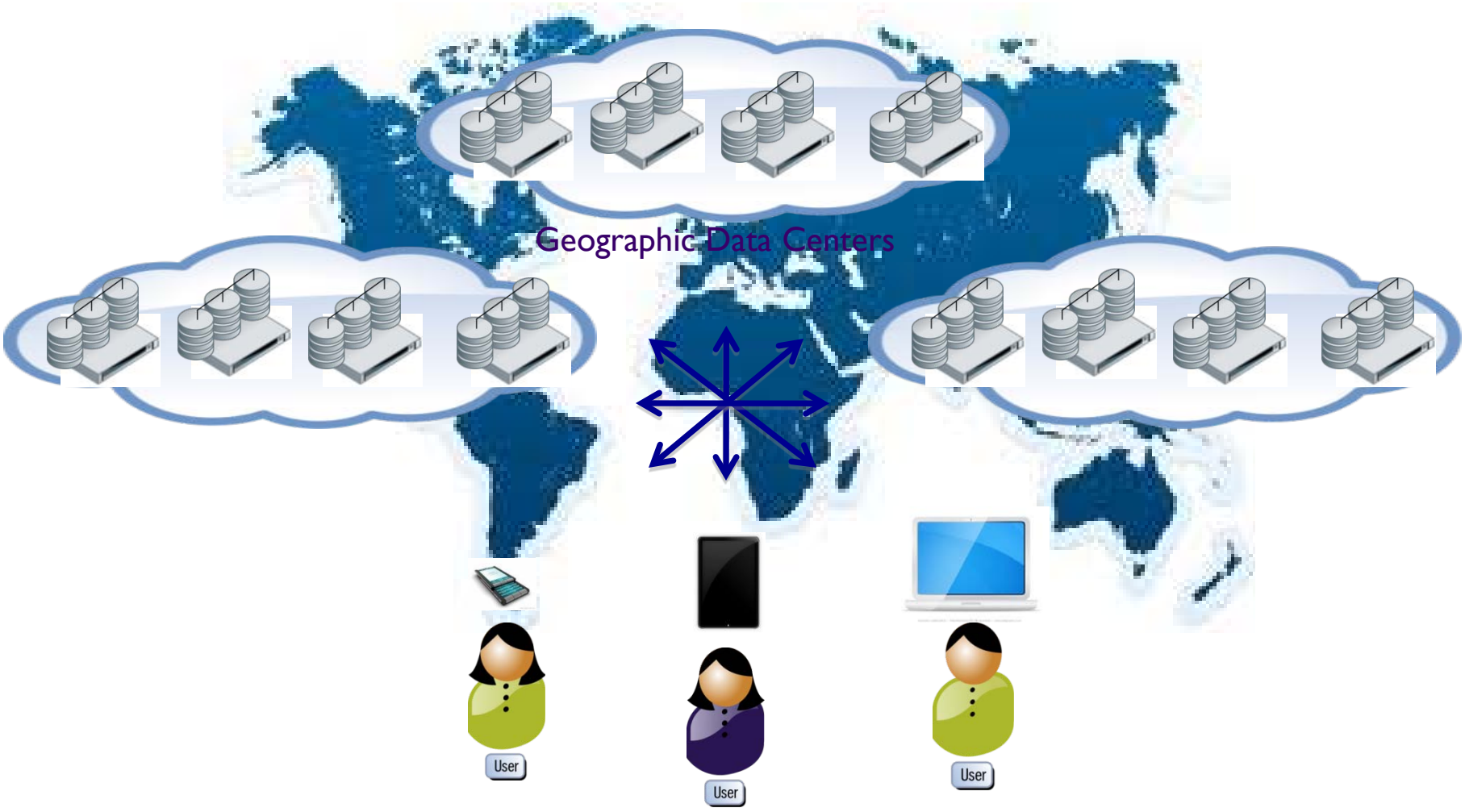


Rise of Non-Traditional Storage

- 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

- APIs Everywhere: Application Programmer Interfaces
- Early Days: Proprietary
  - Examples: Amazon S3, Rackspace Cloud Files, Google GData, Iron Mountain, Twitter API, flickr API, Sun Cloud API, Facebook APIs, Bycast, Caringo SCSP
- NOSQL: Not Only SQL
  - A whole movement around non-relational databases
  - Store Types: Key Value, Column, Document, Graph
- Notes:
  - In most cases, non-traditional stores ***compliment*** 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

REST

- **Re**presentation **S**tate **T**ransfer
  - 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

Development Nuts and Bolts

APIs
Libraries
Frameworks
Platforms
IDEs

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/Libraries:
  - iOS*
    - Apple: URL Loading System (roll your own, not a framework)
    - ASIHTTPRequest: 3rd party open source (library, not a framework)
    - ObjectiveResource: 3rd party open source (Ruby on Rails specific)
  - Android*: cannot find an official Web Services framework
  - Microsoft: .Net, Silverlight
  - Adobe: Flash/Flex
  - Java: JavaFX, Spring, AJAX
  - Ruby on Rails
  - Python: Django
- Note: *Mobile platforms still evolving, but will be focus of future

Integrating CDMI

- Still very early days
- CDMI Server Reference Implementation:
  - PROTOTYPE / Example Only
  - Not a framework or library
  - Uses Spring Framework
- iOS:
  - Goal to open source a minimal CDMI object creation client library by SDC

Example
iOS / iPad Client Demo

Summary

- Cloud & Cloud Storage viewed as a technology here
- 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

Thanks!

Q&A