

# Cloud Systems BoF

Steven Newhouse

Microsoft

# Evolving State of (my) Networking

- 1970s: Walked across campus to submit jobs
  - Carrying a stack of punch cards!
- 1980s: Cross-campus networking
  - Distributed Computing. International networks
- 1990s: Limited home connectivity
  - 56K modems to ADSL
- 2000s: Internet everywhere
  - Massive bandwidth to the home & to the mobile

# Evolving State of Computing

- 1970s: Time sharing mainframes
  - A few machines per campus
- 1980s: Affordable Distributed Computing
  - Home & Office desktops. Research workstations
- 1990s: High Performance Computing
  - Vector processors, high-speed interconnects, COTS
- 2000s: The Grid
  - Enough connected computing to do interesting ‘stuff’

Universal ubiquitous network  
connectivity...

How does this change the way we work?

# Changing Software Use

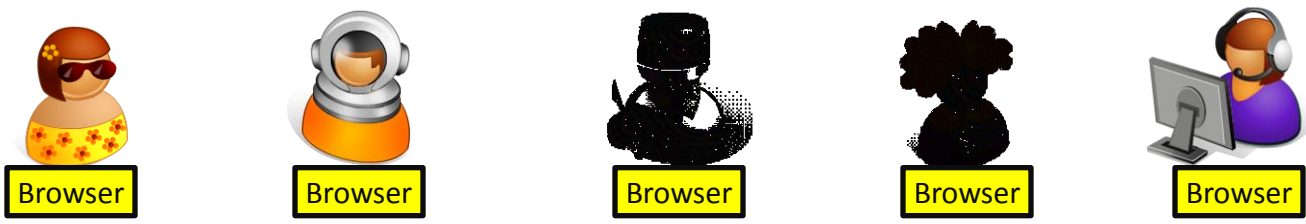
- Standalone Software (e.g. Traditional MS)
  - Your hardware & staff
  - Isolated Desktop installation
  - Networked workgroup licensing
- Hosted Software (e.g. MS Online)
  - Service Agreement
  - Not your hardware, software or staff
- Cloud Services (e.g. MS Live)
  - Software as a service
  - Potential pay per use model

# Changing the way we work

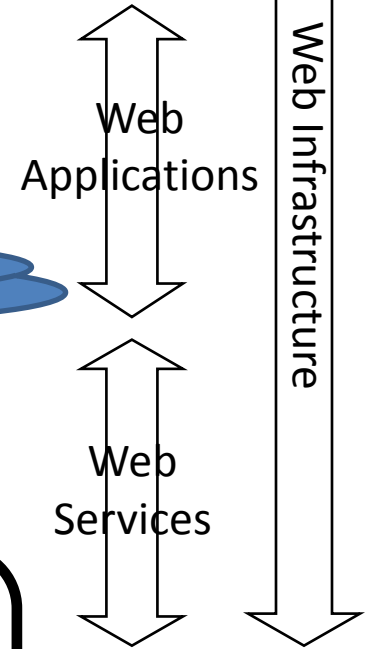
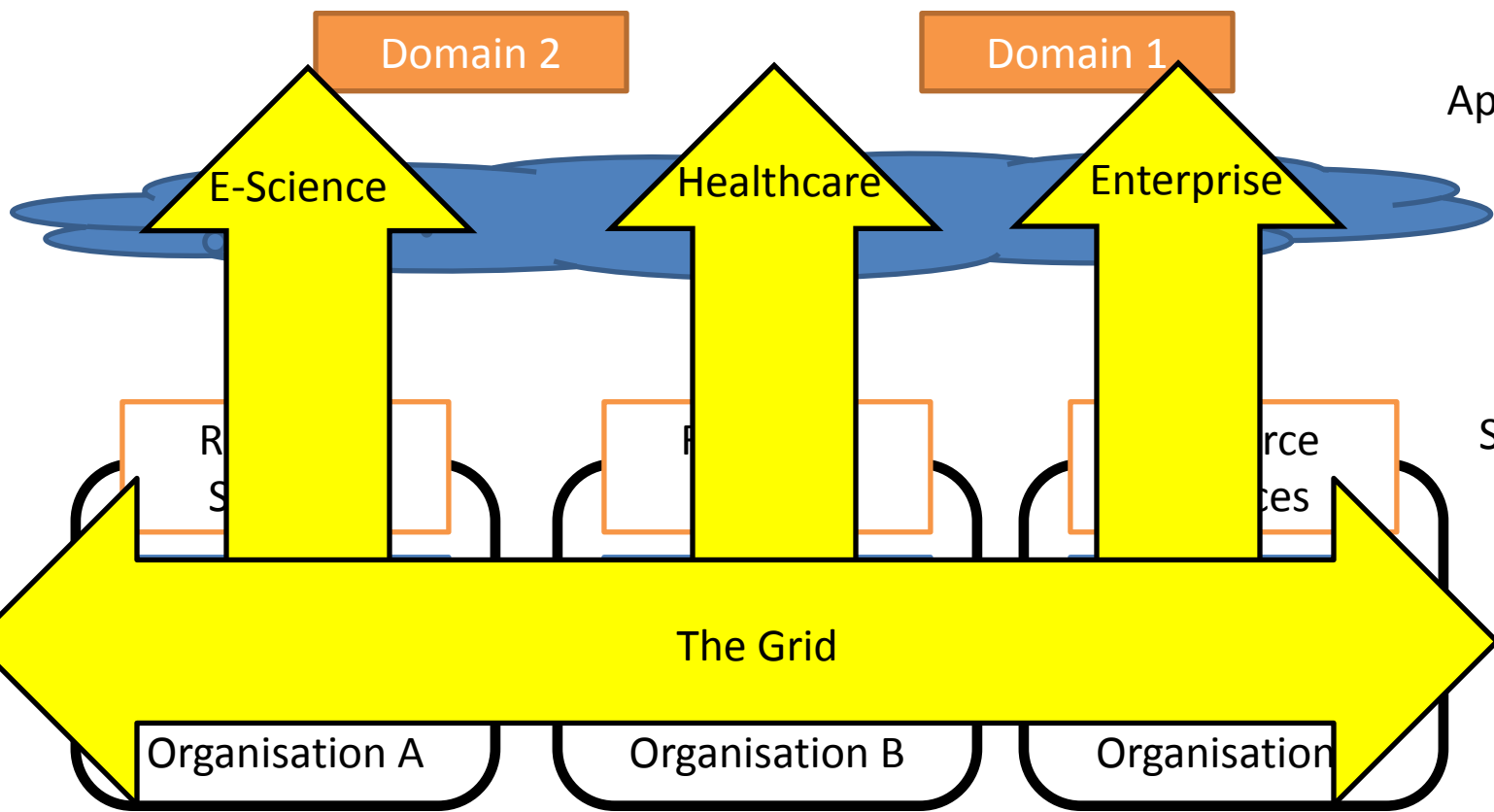
- Collaboration, collaboration, collaboration
  - IM
  - Email
  - Shared documents (e.g. Sharepoint, Wikis)
  - ....
- Information, information, information
  - Generating documents (of all forms)
  - Searching documents (of all forms)
  - Gaining knowledge to exploit information

# In Moving to the Cloud...

- It's nothing new!
  - Software running remotely on a computer
  - Various access methods
    - Thin client web browser (e.g. traditional server side)
    - Active thin client (e.g. smarter client – Silverlight, AJAX,...)
    - Fat clients (e.g. MS Office & Sharepoint)
- It's a whole new paradigm!
  - Everything is done & stored on remote resources
  - Your (business) success depends 100% on someone else
  - Breaks the enterprise silo to enable collaboration



**Web 2.0 / Portals / APIs**





# So, What is the Cloud?

- A User oriented access layer to the distributed resources
  - A layer above the resource services
  - Focused on what users want to do
    - Not on how they are provisioned
- Different clients for different communities:
  - From the Browser
    - ‘Traditional’ Portals
    - New Portals (Web 2.0)
  - From the Operating System
    - Transparently access the grid from desktop Windows & Linux
      - E.g. from with shell environments
- Both need APIs (from Browser or OS)
  - e.g. C#, Java, JSON, Perl, PowerShell

# Hierarchical Decomposition

- Users will want to go to multiple clouds
  - User → Cloud: Bespoke solution
  - User → Clouds:
    - Interoperability through the UI (Hotmail vs. Gmail)
    - Interoperability through standards (IMAP, SMTP, ...)
- Cloud providers will want multiple resources
  - Swap providers of resource services (no lock in)
  - Both internal and external resource providers

# Do Clouds Need Standards?

- Standards at two levels:
  - Infrastructure Services
  - Interface Services
- Infrastructure
  - Low-level compute & data services
  - Management & Monitoring services
- Interface
  - Authentication & Authorization management
  - Functional Application or domain services

# Standard APIs?

- APIs to the clouds
  - Help build a client ecosystem
  - Avoid provider lockin
    - Not a concern now... but later?
- APIs to the resources
  - Helps build portable cloud services
  - Portability between resource provider
    - And across 'ensembles' from a resource provider

# What can OGF do?

- What is it **you** want to do within OGF?
- OGF has a set of resource specifications
  - Computing, data, ...
- Looking at how to apply OGF specifications
  - Specification adoption – Andrew Grimshaw
  - Vendors have technology that can be exposed
    - MS HPCS 2008, LSF, PBS
- Looking at practical gaps in current OGF activity
  - Be consumer (community) driven through use cases

# Summary

- The Cloud is a new perspective on the Grid
- Focused on its **use** rather than its **provision**
- The need for Grid standards continues
  - Strong base: HPCBP, BES, JSDL, RNS, Naming, ...
- Need for standards in the access layer
  - Simple API for Grid Applications
  - Accounting... someone will HAVE to pay!