Cloud computing

Antonio Corradi

DEIS, University of Bologna, ITALY

antonio.corradi@unibo.it

What is cloud computing?

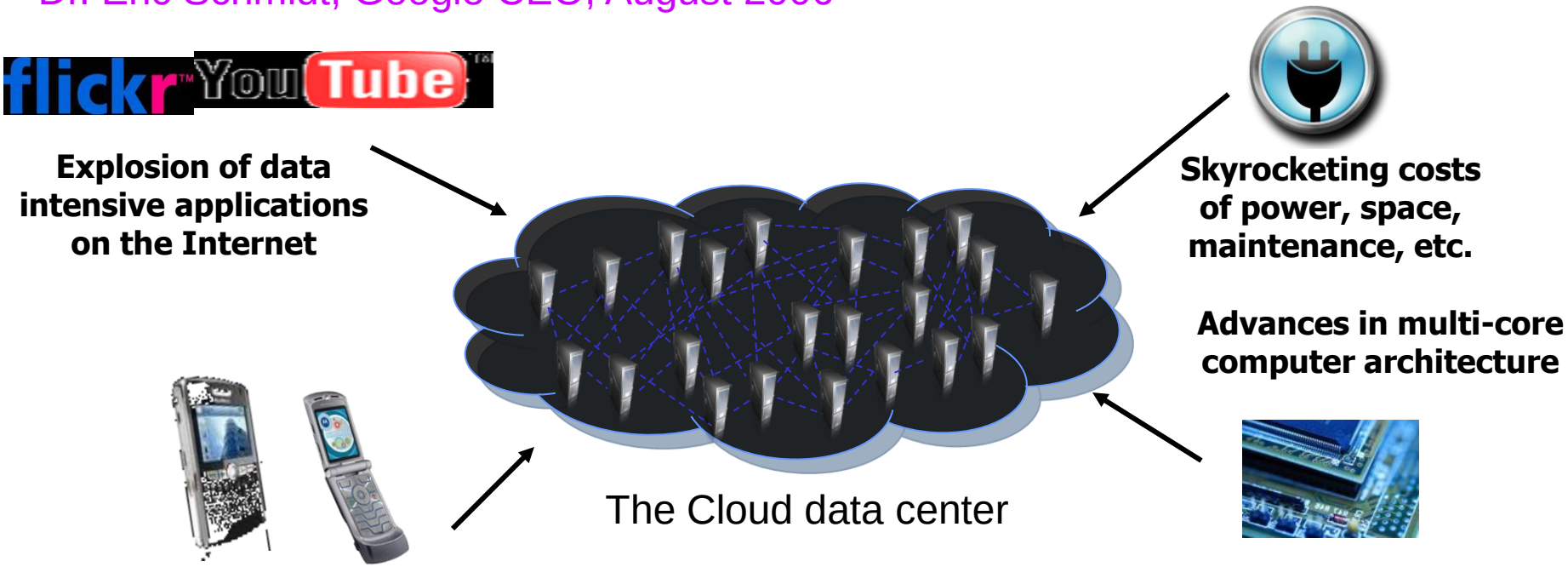
"The architecture and terminology of cloud computing is as clearly and precisely ***defined as, well, a cloud****."*

Source:
www.opencloudmanifesto.org

Cloud Computing Problem Space

"It starts with the premise that the **data services and architecture** should be on **servers**. We call it **cloud computing** – they should be in a 'cloud' somewhere. And that if you have the right kind of **browser** or the right kind of access, it doesn't matter whether you have a PC or a Mac or a mobile phone or a BlackBerry or what have you – or new devices still to be developed – you can get access to the cloud…"

Dr. Eric Schmidt, Google CEO, August 2006



Cloud Concepts

- IT on demand pricing
- Best benefits in a reliable context
- Pool of virtualized computer resources
- Rapid live providing while demanding
- Systems on scaling architecture

Cloud keywords

on demand,
reliability,
virtualization,
provisioning,
scalability

What is a Cloud

One Cloud is capable of providing IT resources 'as a service'

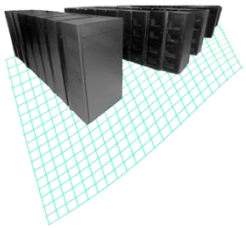
One Cloud is an **IT service** delivered to users that have:

- a **user interface** that makes the infrastructure underlying the service transparent to the user
- reduced **incremental management costs** when additional IT resources are added
- **services oriented management** architecture
- **massive scalability**

A bit of history

Grid Computing

- Solving large problems with parallel computing



Utility Computing

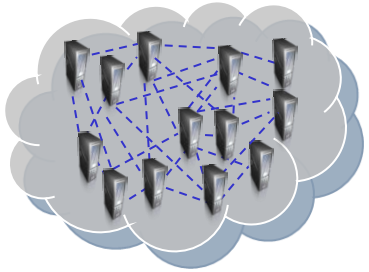
- Offering computing resources as a service

Software as a Service

- Network-based subscriptions to applications

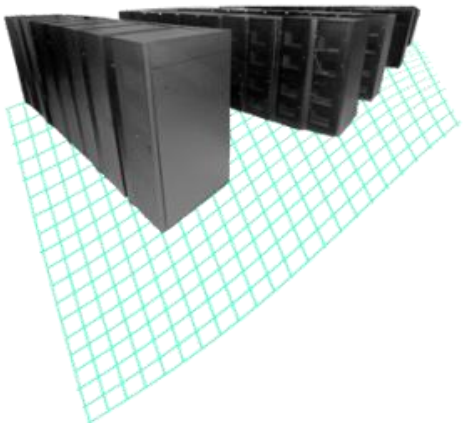
Cloud Computing

- Anytime, anywhere access to IT resources delivered dynamically as a service.



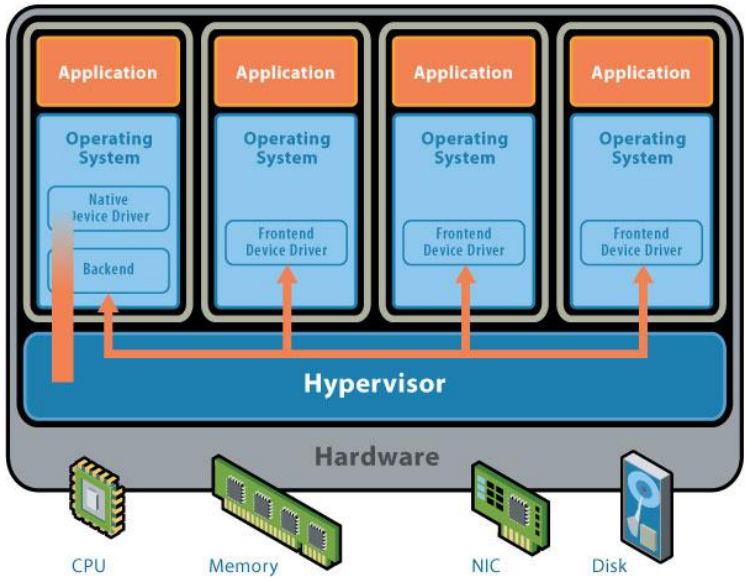
Before Cloud computing: GRID

- **Grid computing**
  - Sharing of **heterogeneous resources** (computer, software, data, memory, computational power,, …) in **highly distributed environments** with the goal of **creating a virtual organization scalable** (***by need!***)
  - Interfaces (for management), often **too fine grained, with low level of abstraction,** and **non self-contained** ☹
  - Application areas very **limited and specific** (parallel computation for scientific, engineering scenarios, …)



Before the Cloud: Virtualization

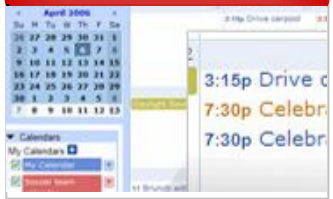
- **Virtualization**
  - Technologies for **virtualization** (either system-based or hosted), as in a server farm**:** Vmware, Xen, …
  - **Isolation** & **personalized infrastructure and/or SW platform** (O.S. and some additional applications)
  - Tool for the **efficient management** of computing infrastructures (IBM Tivoli suite, Xen monitoring tools, …)



Before Cloud computing: Web 2.0

- **Web 2.0**
  - **Usage of asynchronous protocols not visible** to users to ask only **really required info** and not the **whole web pages: Asynchronous Javascript And XML (AJAX)**
  - New ways of **using Web services coupled with new applications easier to use, collaboration based** and **openly available, without requiring any installation** by interested users: new business model, **very, very cooperative** (**S**oftware **a**s **a S**ervice ☺)

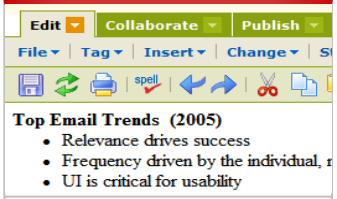
Office



https://www.google.com/hosted
http://smallbusiness.yahoo.com/email/
www.zimbra.com

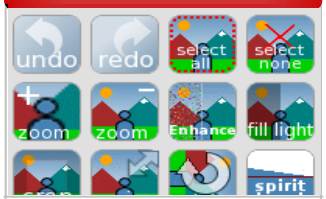
many others…

Word



www.writely.com
www.writeboard.com
www.inetword.com
many others…

Graphics

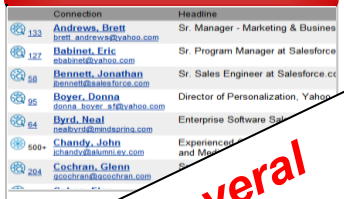


www.pxn8.com
www.pixoh.com
many others…

Database

www.dabbledb.com
www.Lazybase.com
www.quickbase.com
many others…

Contacts



And several others…

Before Cloud computing: Utility computing

- Huge computational and storage capabilities available from **utilities**, the same as for energy and electricity, and on pay-per-use base.
- "**Computing may someday be organized as a public utility**" - John McCarthy, MIT Centennial in 1961
- **Metered billing** (pay for what you use)
- **Simple to use interface** to access the capability (e.g., plugging into an outlet)

Software as a Service (SaaS)

Traditional Software

Build Your Own

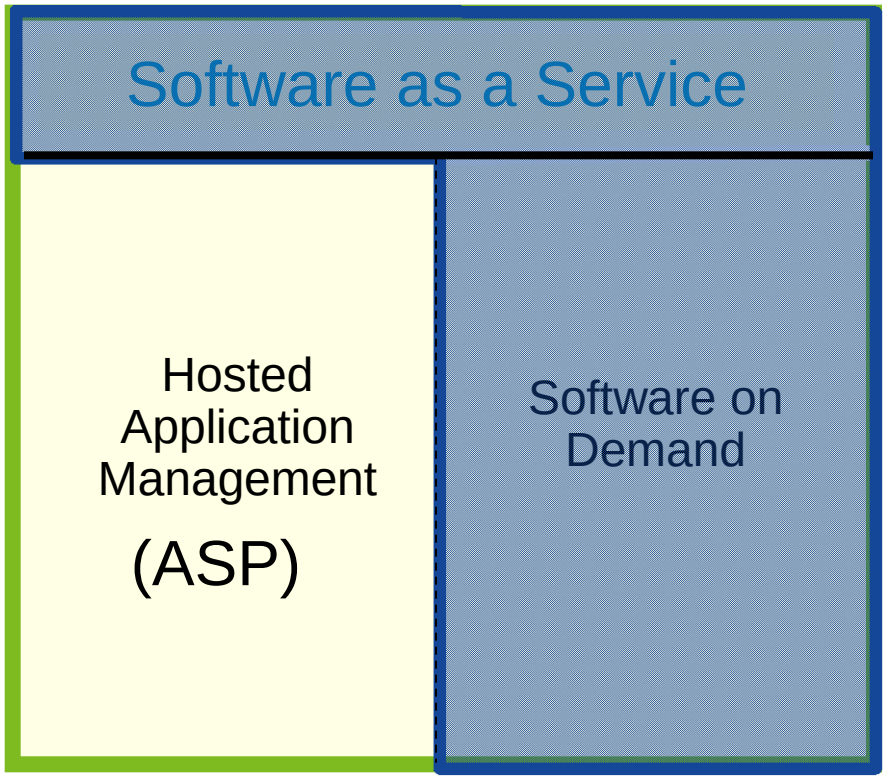
On-Demand Utility

Plug In, Subscribe

Pay-per-Use

Software as a Service (SaaS)

- Built for one-to-one delivery or management
- Applications deployed in-house
- Dedicated infrastructure/ environments
- Highly customized
- One-time license and recurring maintenance or support fee

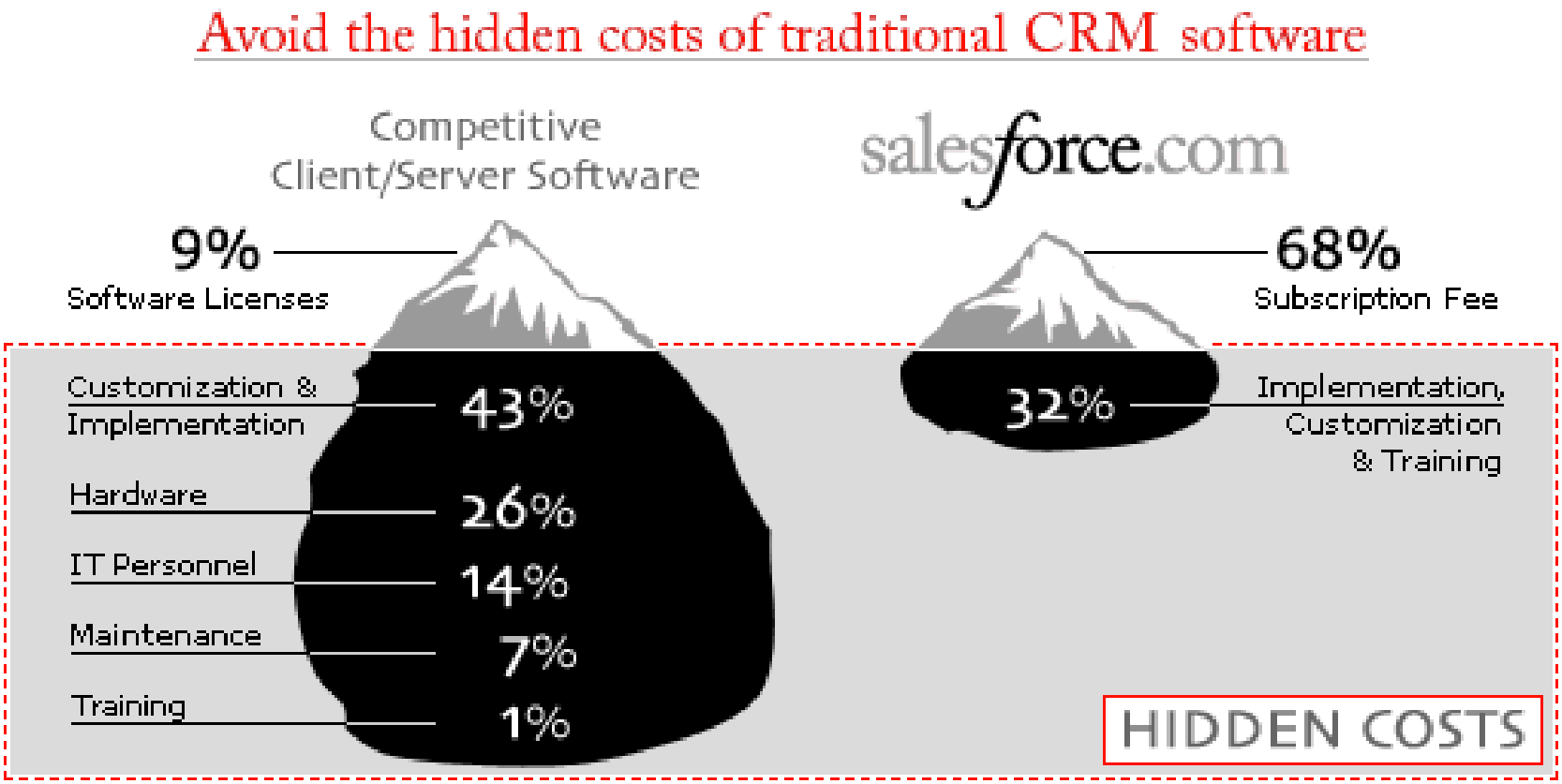


- Built for one-to-many delivery over Web
- Applications not deployed in-house
- Shared public infrastructure
- Little customization
- Subscription fee or advertising-supported

Time →
Perpetual license → Subscription
One-to-few → One-to-many
Private infrastructure → Public infrastructure

Source: IDC, 2006

Hidden Cost of IT



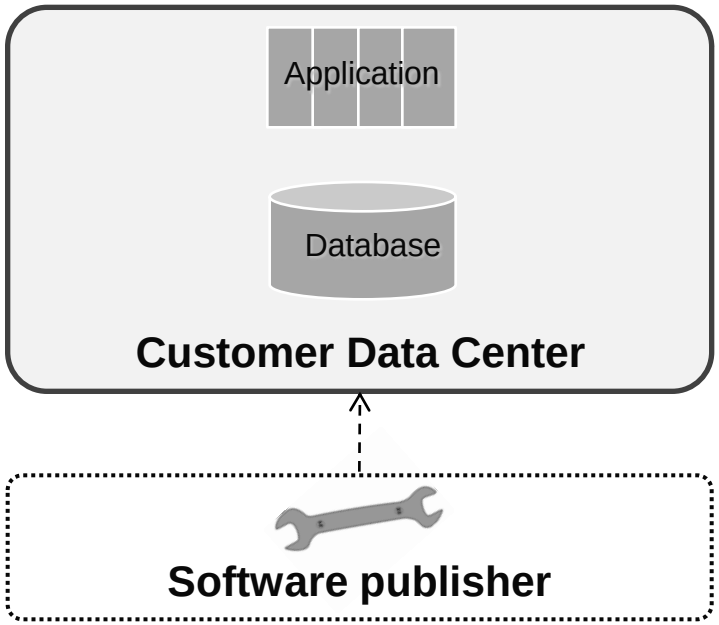
SaaS - Software as a Service

- **Software ownership costs pushed to vendor** - hardware, software, system security, disaster recovery, maintenance, monitoring
- **Return to core competency** - organizations shift resources to core competencies, vendors focus on managing their SaaS
- **More efficient deployment** - instant evaluation, more collaboration between vendor and IT organization, much faster deployments
- **Eliminate shelfware & maintenance** - pay for what you use
- **Always on current version** - version-free software means the latest for the customer
- **Modern, Web 2.0 interface** - drive technician usage and better customer interaction with IT
- **SaaS homogeneity costs less** - one version for the vendor to support means lower costs for everybody

Applicazioni areas suitable for SaaS

- **ERP vertical business** applications, both specialized and very specific
- **General-purpose** applications without any adaptation (potentially sharable)
  - self-service provisioning and ad-hoc personalization
  - applications available to several different users
- **Business B2B** applications **domain specific**
  - no need of third party hosting and involvement
- **Customer/Supplier** applications
  - applications where most of users and access is externally to the organization and where ubiquitous access via Web is critical and intrinsic
- **Business** applications **even critical**,
  but not the **core business ones**

Traditional on-premise Deployment at the client site



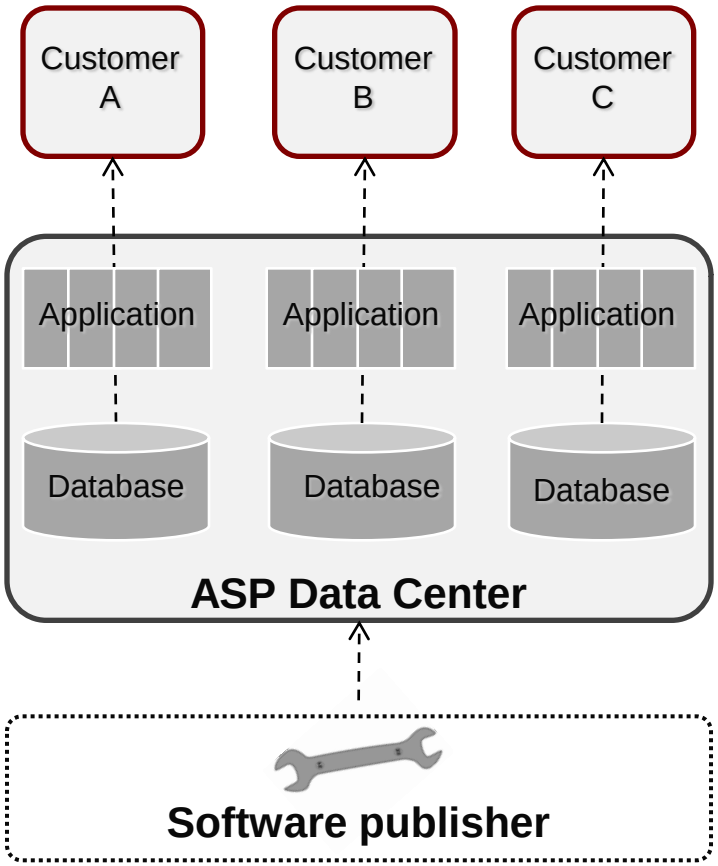
Details

- Full ownership
- Significant implementation
- Customizable
- Difficult to upgrade / maintain

Examples

- HP Service Manager
- BMC Remedy
- CA Service Desk
- EMC Infra

Application Service Provider (ASP)



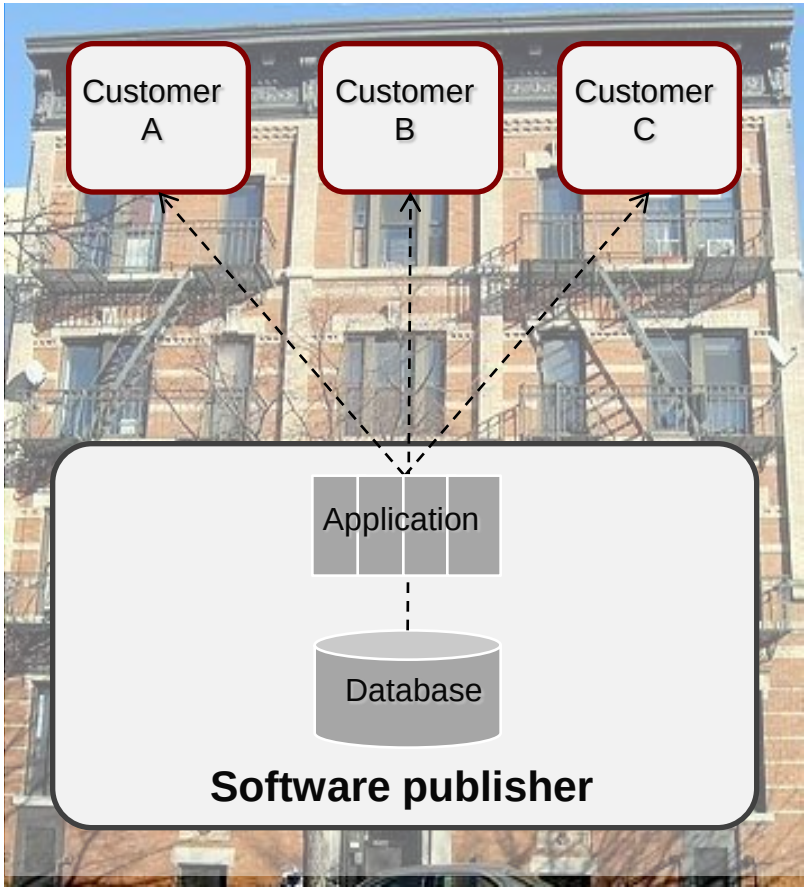
Details

- Procures app and resells service
- Broker between customer and publisher
- Focus on ‘out-of-box’

Examples

- IBM GS
- HP Services
- BMC AAS
- CSC

SaaS multi-tenant



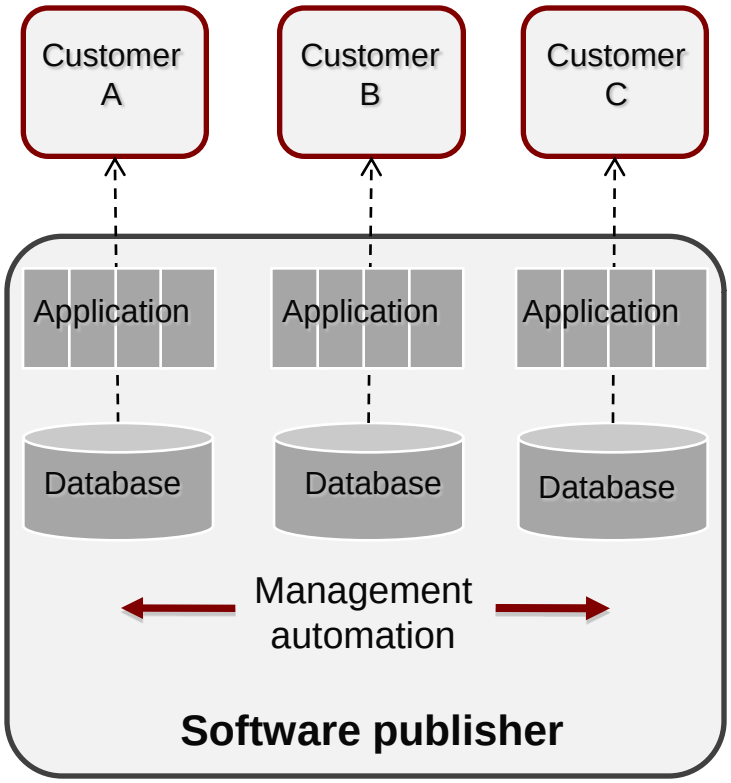
Details

- Hosted by software publisher
- Many customers to one application set
- Thought to be inflexible

Examples

- Salesforce.com
- Workday
- Innotas

SaaS single-tenant



Details

- Hosted by software publisher
- Customers receive their own app and database
- Auto-upgrades
- Extensive customization

Examples

- Service-now.com
- InteQ
- Eloqua

Modelli SaaS in evolution

Some increasing resources models for providing some resources as a service, X**aaS**

SaaS **S**oftware **a**s **a S**ervice

- Resources are simple **applications available** via remote Web access

PaaS **P**latform **a**s **a S**ervice

- Resources are **whole software platforms available** for remote execution, i.e., several programs capable of interacting with each other

Iaas **I**nfrastructure **a**s **a S**ervice

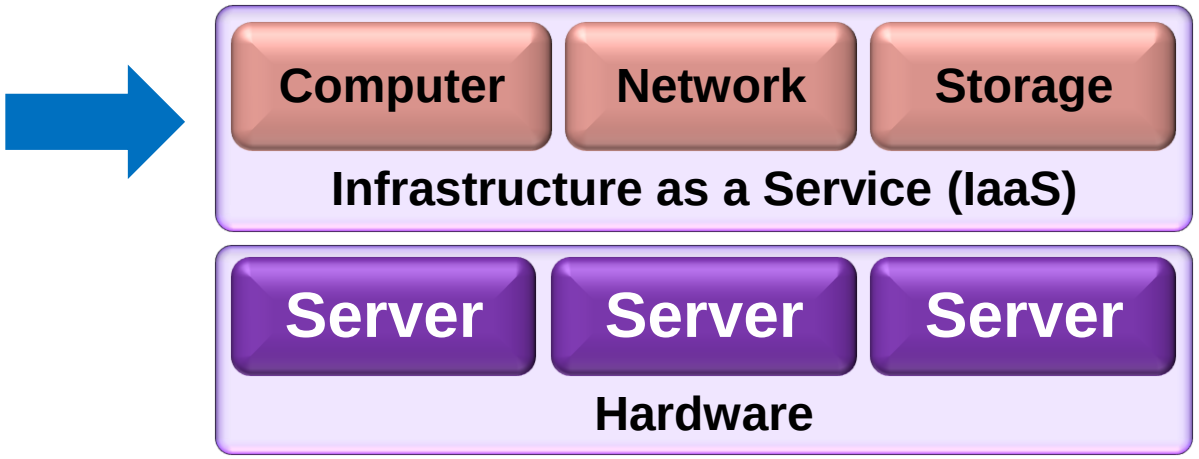
- Resources are intended **in a wider and complete way,** from **hardware platforms**, to **operating systems,** to **support** to final **applications**: usually via virtualization up to **Cloud Computing**

Layered Architecture: IaaS, PaaS & SaaS

- Below the real architecture:
  hardware components
  &
  software products

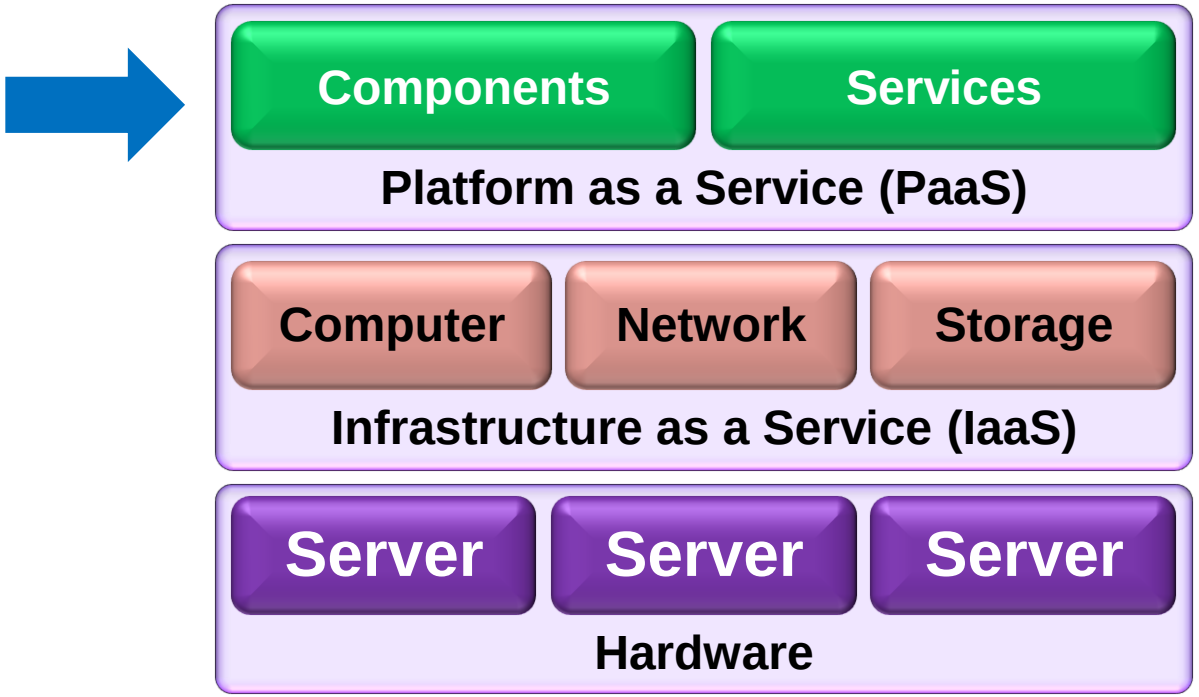
Layered Architecture: IaaS, PaaS & SaaS

- **Infrastructure**: layer to enable the distribution of Cloud services, typically realized by a virtualization platform



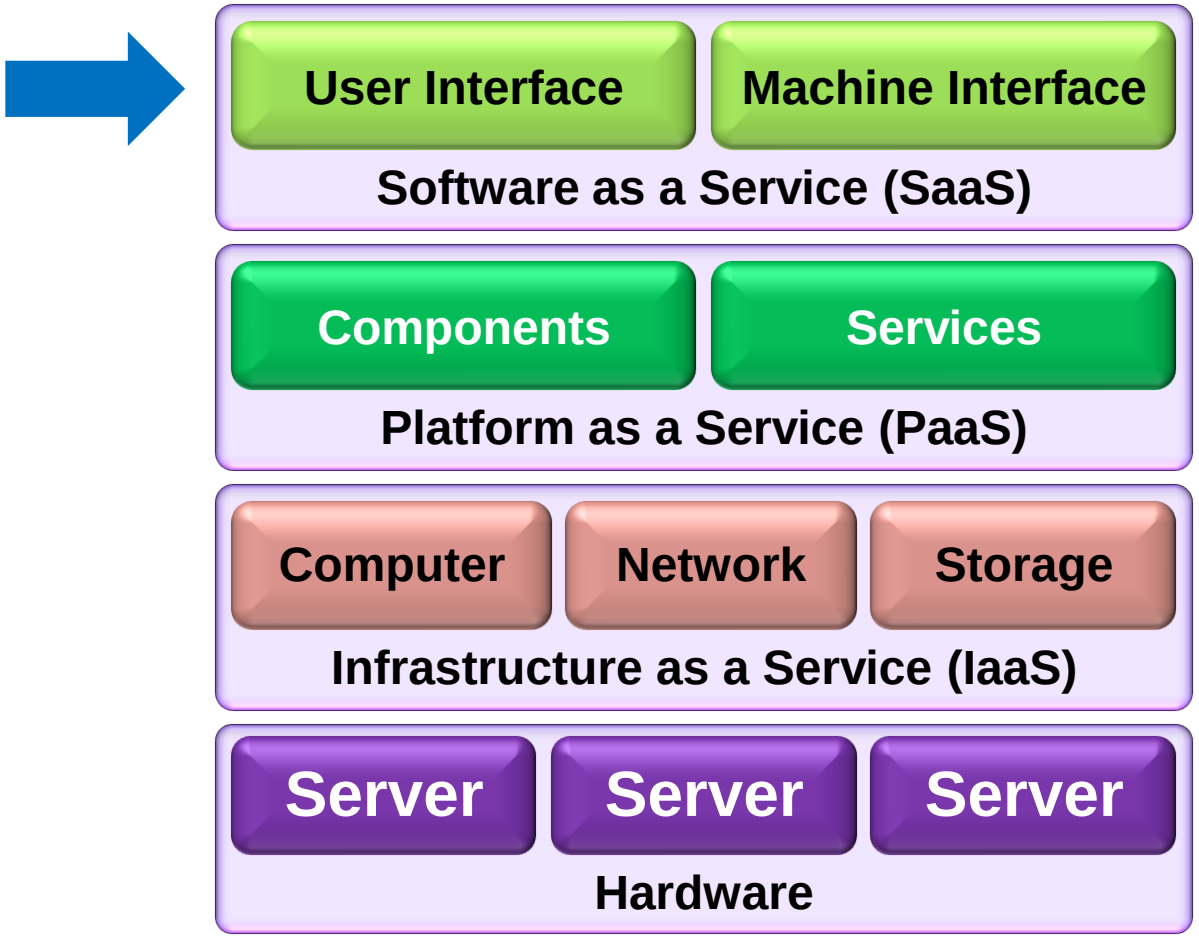
Layered Architecture: IaaS, PaaS & SaaS

- **Platform:** layer to provide to upper layers a set of services and components remotely available



Layered Architecture: IaaS, PaaS & SaaS

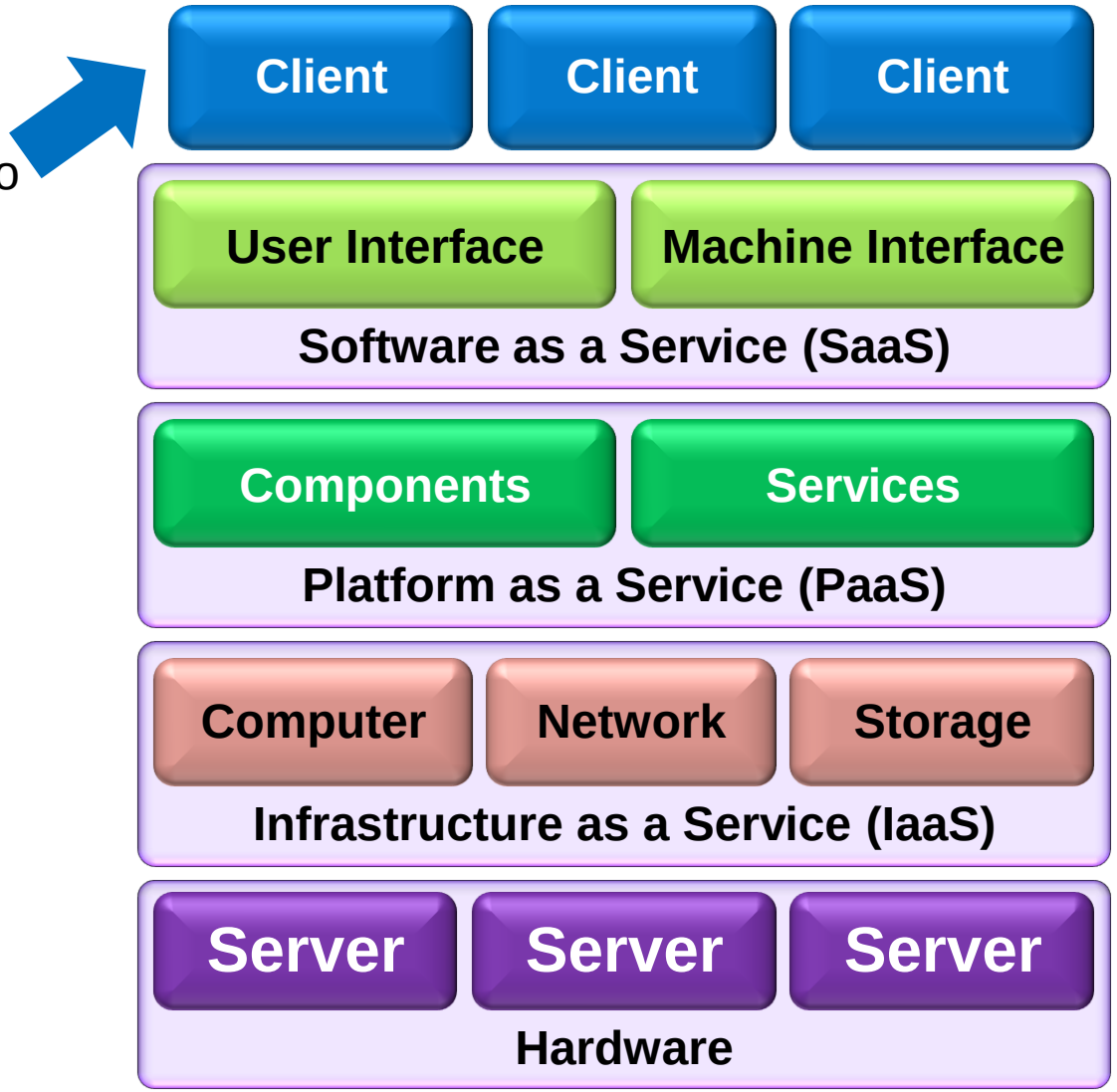
- **Application**: layer to install applications, to be available via Web and Internet via Cloud



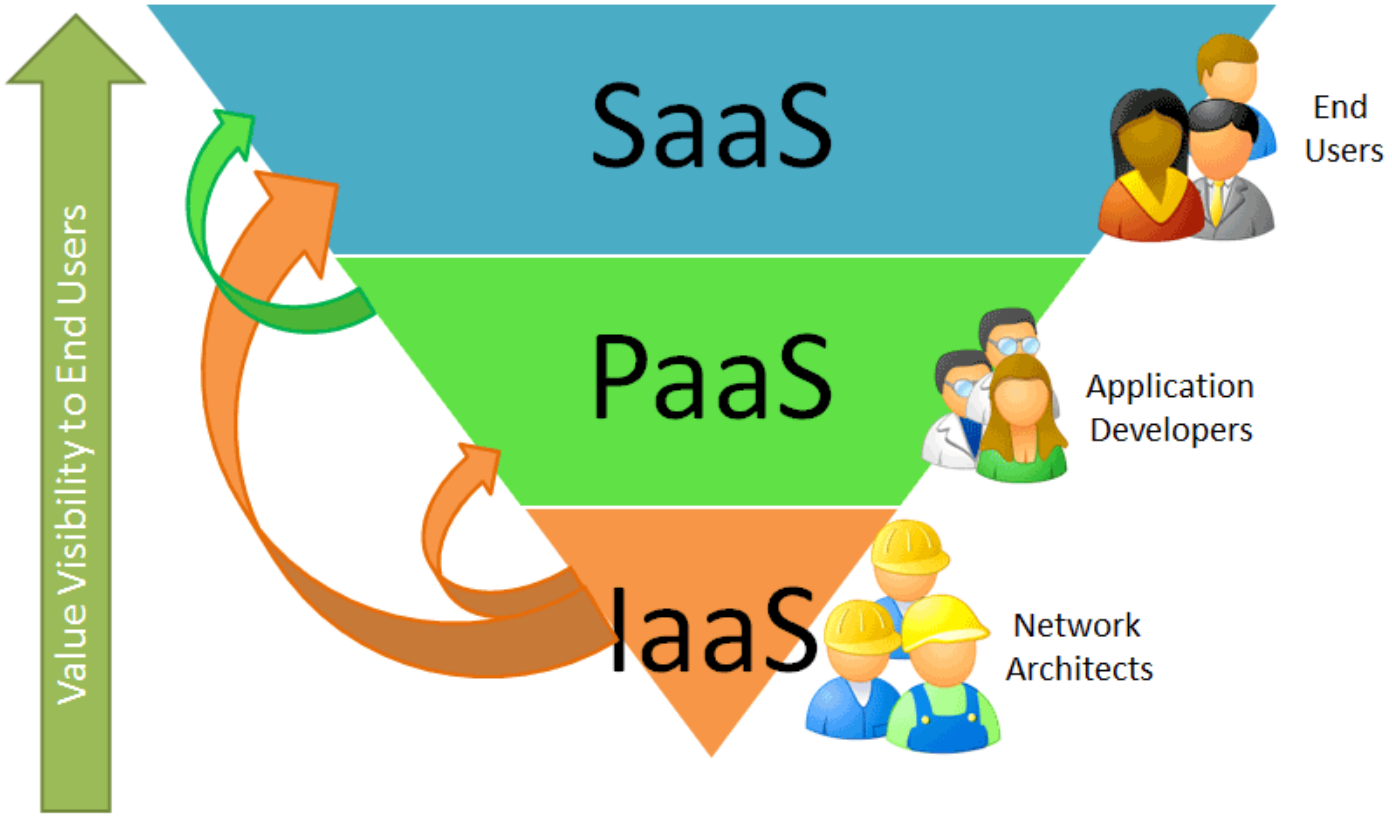
Layered Architecture: IaaS, PaaS & SaaS

- **Client** software to get access to the system.
  Those applications execute on the **client physical platforms** (remote computers) owned by the final remote user

  they can communicate with the Cloud via the **available interfaces**



Layered Architecture: Actors



Some SaaS and aaS examples

SaaS

From desktop applications: **Google Apps** (Gmail, Google calendar & docs), **Microsoft Window live** (Hotmail, Messenger, …) to search engines, Google, Yahoo, Several **social networks** **(**Facebook, LinkedIn, Twitter, …)

PaaS typically accessed via Web service

Services available internally to and interacting with other applications, as **Google Maps**

Iaas some experimental infrastructures

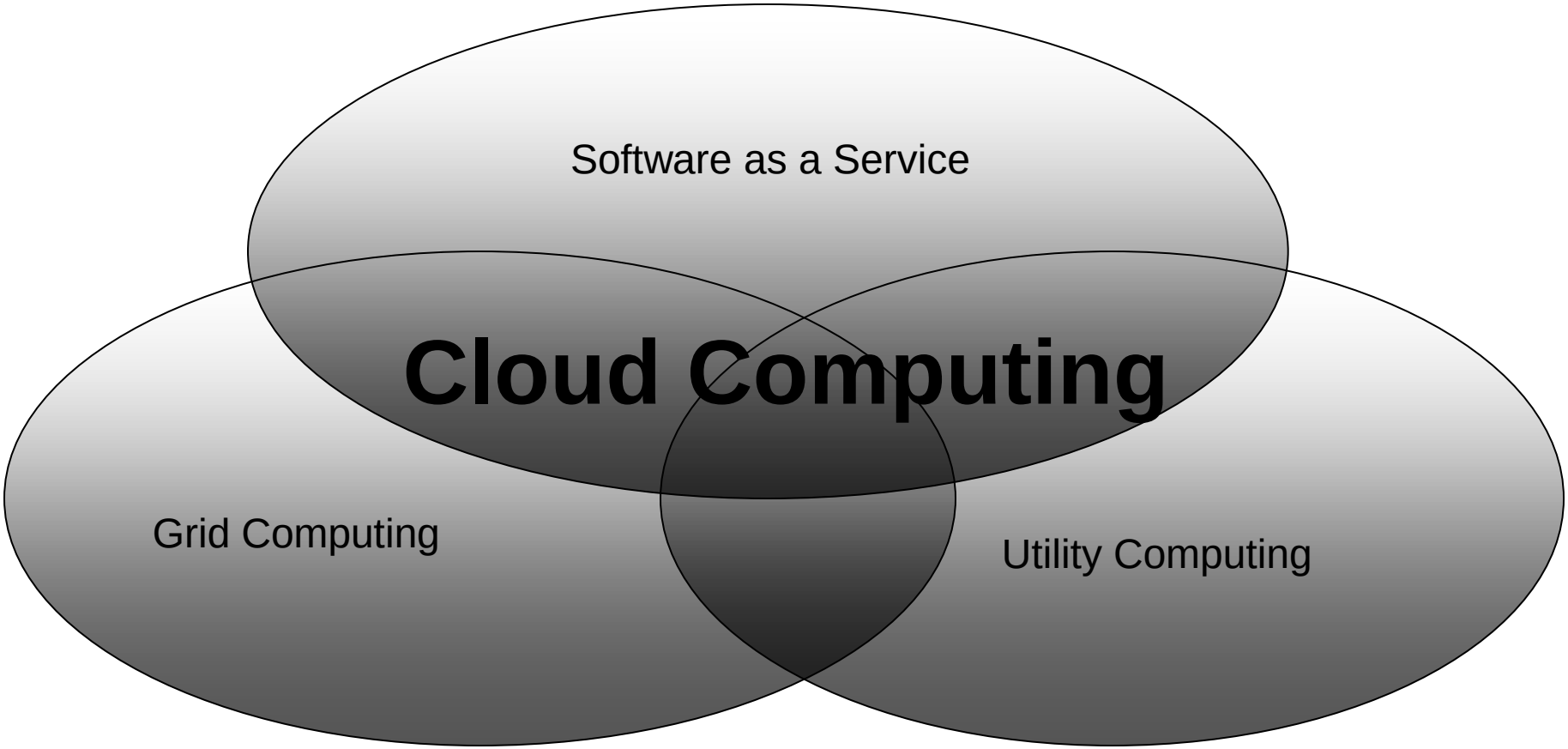
Several examples, with virtualization services, **Amazon Web Services** (S3), **Elastic Computing Cloud** (EC2), to several management and monitoring desktops to control execution (Sun global desktop, Zimdesk, …)

Cloud different from …

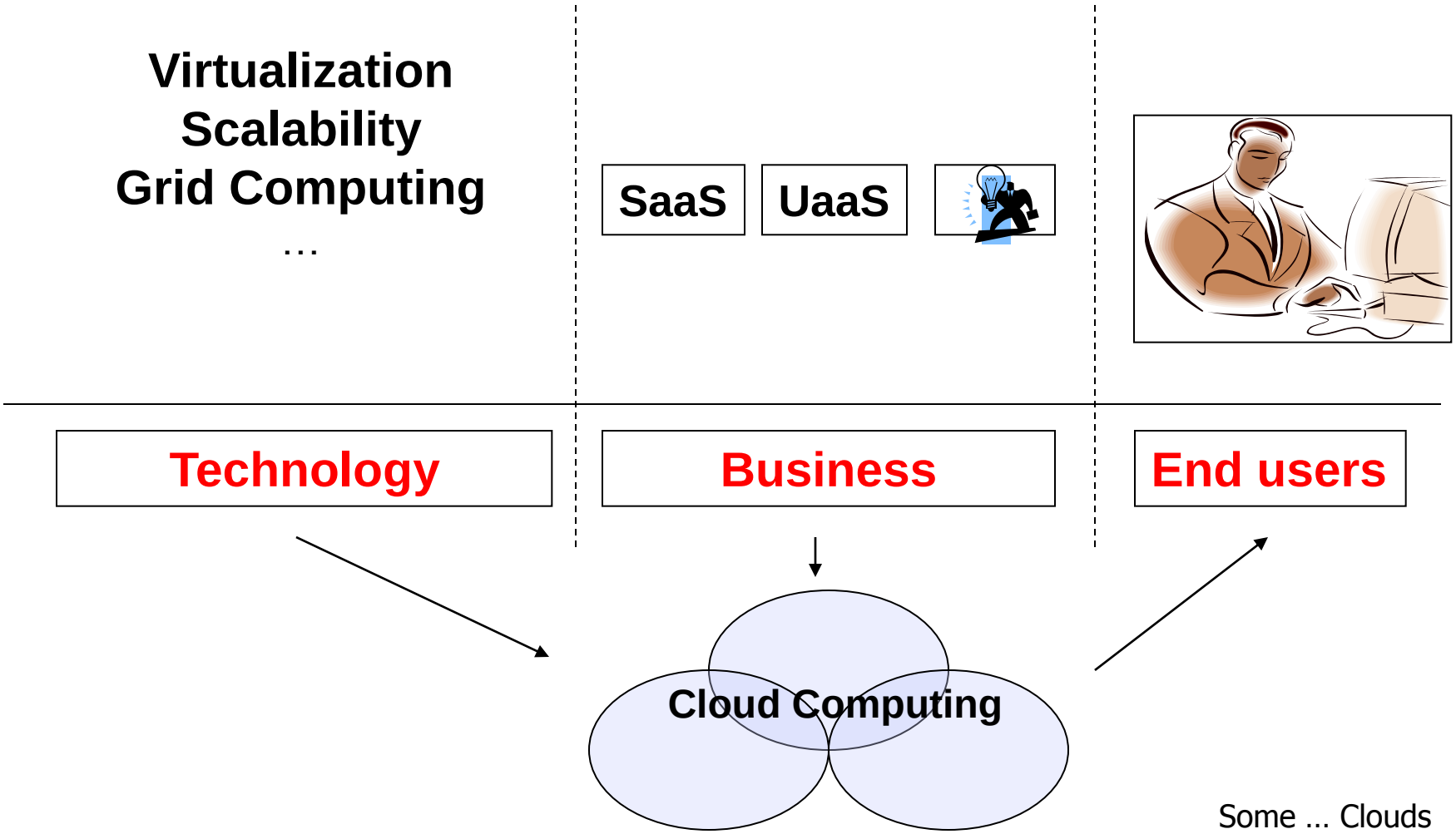
- **Grid Computing**
  - A cloud is more than a **collection of computer resources** because a cloud provides a mechanism to manage those resources

    Provisioning, change requests, workload balancing, monitoring
  - Cloud computing is an infrastructure that sits on top of a data centre for efficiency
- **Utility Computing**
  - Service that allows users to **deploy, manage, and scale** online services using the provider's resources and pay for resources they consume
  - Users want to be in control of what runs on each server
  - **Cloud users** want to avoid infrastructure. The provider is in complete control.
- **SaaS**
  - Software that is **owned, delivered, and managed remotely** by one or more providers
  - Software that allows a sharing of application processing and storage resources in a one-to-many environment on a pay-for-use basis, or as a subscription

Evolution of Cloud Computing



Technology & Business



Cloud Key Goals
Infrastructure Perspective

- How can we provide flexible compute resources quickly to promote **rapid prototyping**?
- How do we deploy applications that **scale up** to meet increasing demands over time?

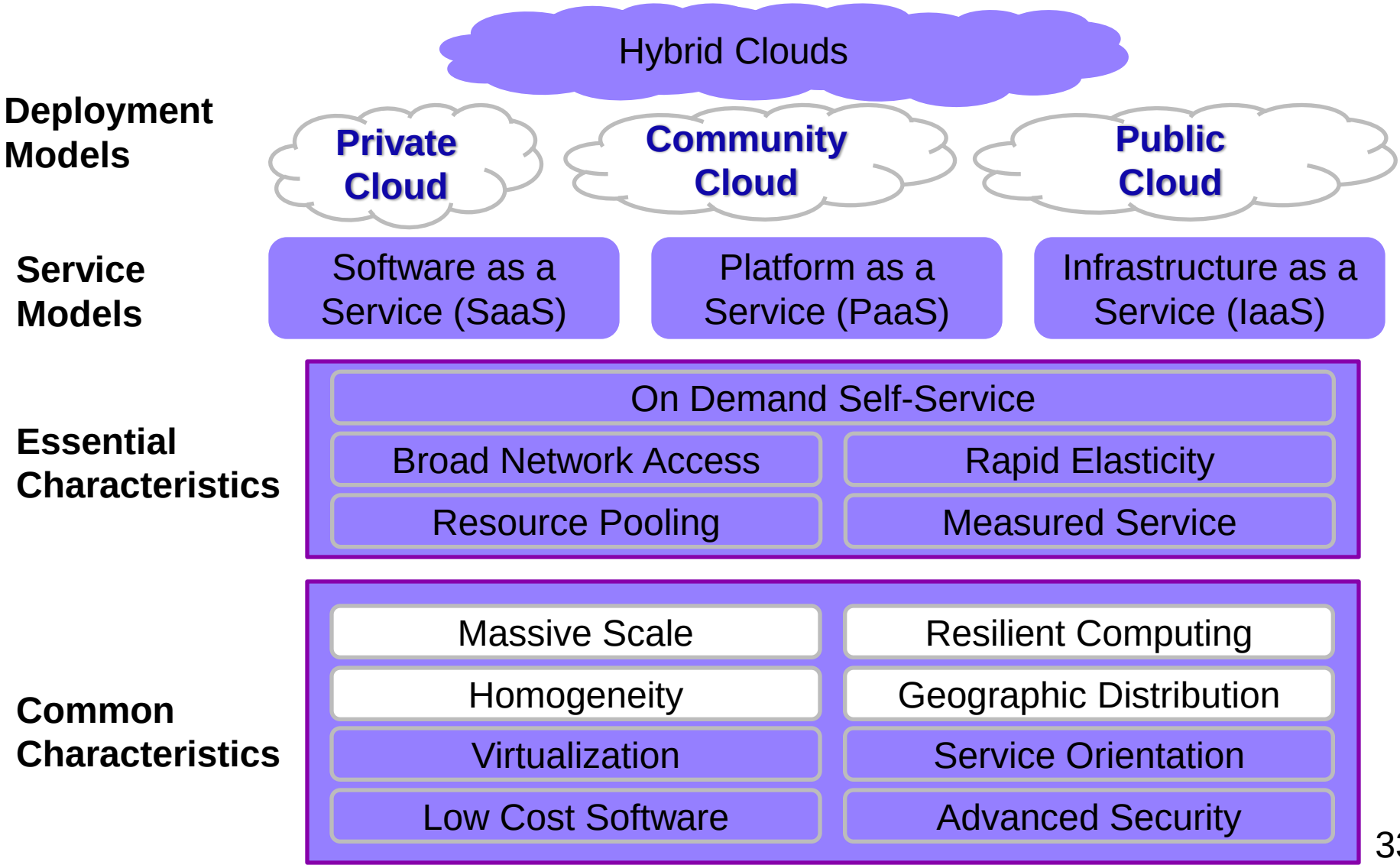
- How do we manage 100,000's of machines with **minimal human intervention**?
- How can we make the most **efficient** use of all the compute resources in a data center?

Cloud Deployment Models

Typically three models

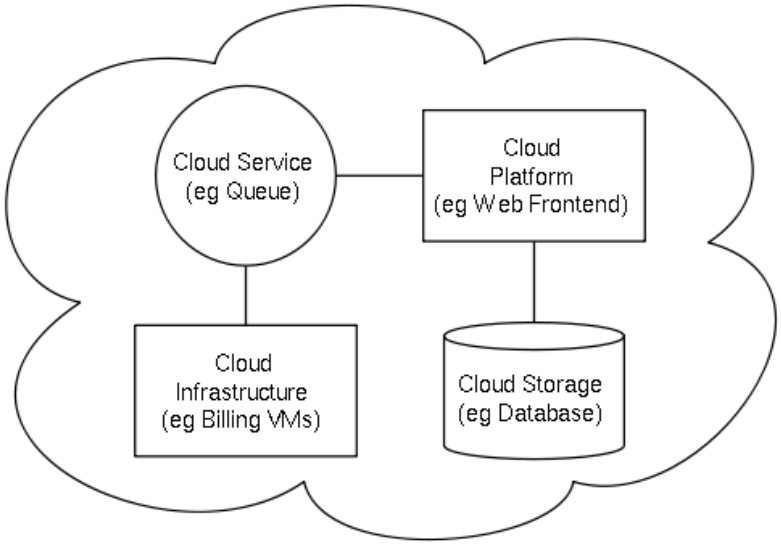
- **Private cloud**
  - enterprise owned or leased
- **Community cloud**
  - shared infrastructure for specific community
- **Public cloud**
  - sold to the public, mega-scale infrastructure
- Hybrid cloud
  - composition of two or more clouds

The NIST Cloud Definition Framework



Cloud components

Cloud Computing software systems have a typical **structure based on components** that can communicate with each other via well defined **interfaces** (often Web Services)

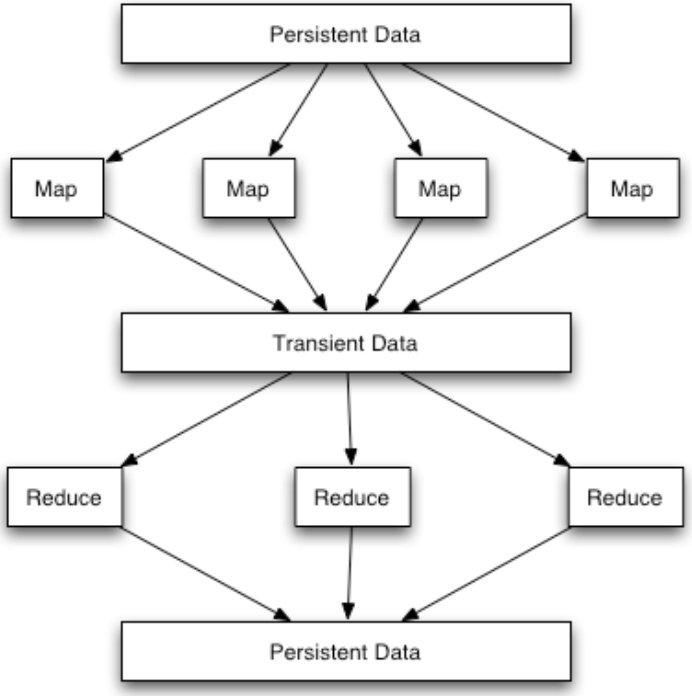


- **Four main components**:
  - one **Cloud platform**, with an externally available **interface accessed via web** to cooperate with the real or virtual internal infrastructure
  - one **virtualization infrastructure** and the management system for the control, monitoring, and billing for **client requests**
  - one internal **memory system** **typically via** a **database**
  - one internal **manager** to handle **external requests** (management, queuing, and controlling)

MapReduce Programming Model Hadoop

- **Functional programming** that is easily parallelizable
- **Split into two phases**:
  - Map – Perform custom function on all items in an array
  - Reduce – Collate map results using custom function
- **Scales well** – computation separated from processing dataflow
- Illustrative example:
  - Map that squares the value of numbers in an array

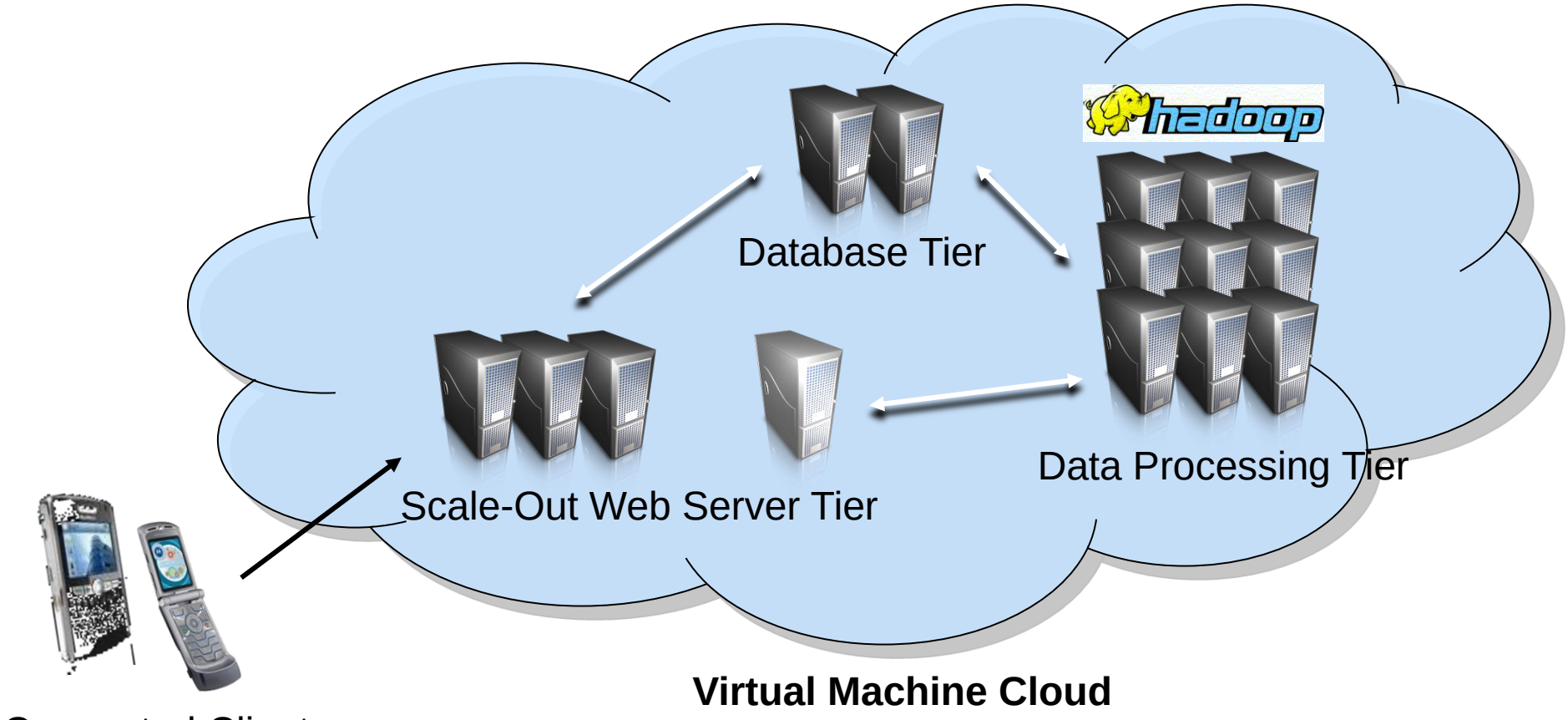
    {1, 2, 3, 4} -> {1, 4, 9, 16}
  - Reduce that sums the squares : 30



Apache Hadoop

- Open source **MapReduce** software platform
- **Automatically provides a framework** for developing MapReduce applications
  - Handling mapping and reducing logistics
  - Programmer provisioning of custom functionality
- Currently takes custom functionality in Java and Python
- IBM developed open source Eclipse plug-in to interface with Hadoop

From the Scale-Out Model to Cloud Computing … pervasive



Cloud computing: reality check

- **Amazon Elastic Computing – EC2**: virtualized images (DB+Software and middleware+OS), Xen, simple SLA console
- **Google App Engine** (Software as a Service, web applications, Google App Engine, sandbox for management and security)
- **IBM Blue Cloud**: virtualized images (DB+Software and middleware+OS), Xen, Tivoli (monitoring and management), simple SLA console
- **HP/Yahoo/Intel Test Bed**: virtualized images, Xen, simple SLA console
- **Microsoft Azure**: recently launched by Microsoft
- **Research initiatives** (***RESERVOIR EU FP7 project***, previous projects on grid computing such as EEGE, …)

Others ongoing projects: **Eucalyptus**, **3Tera**, …

Google App Engine

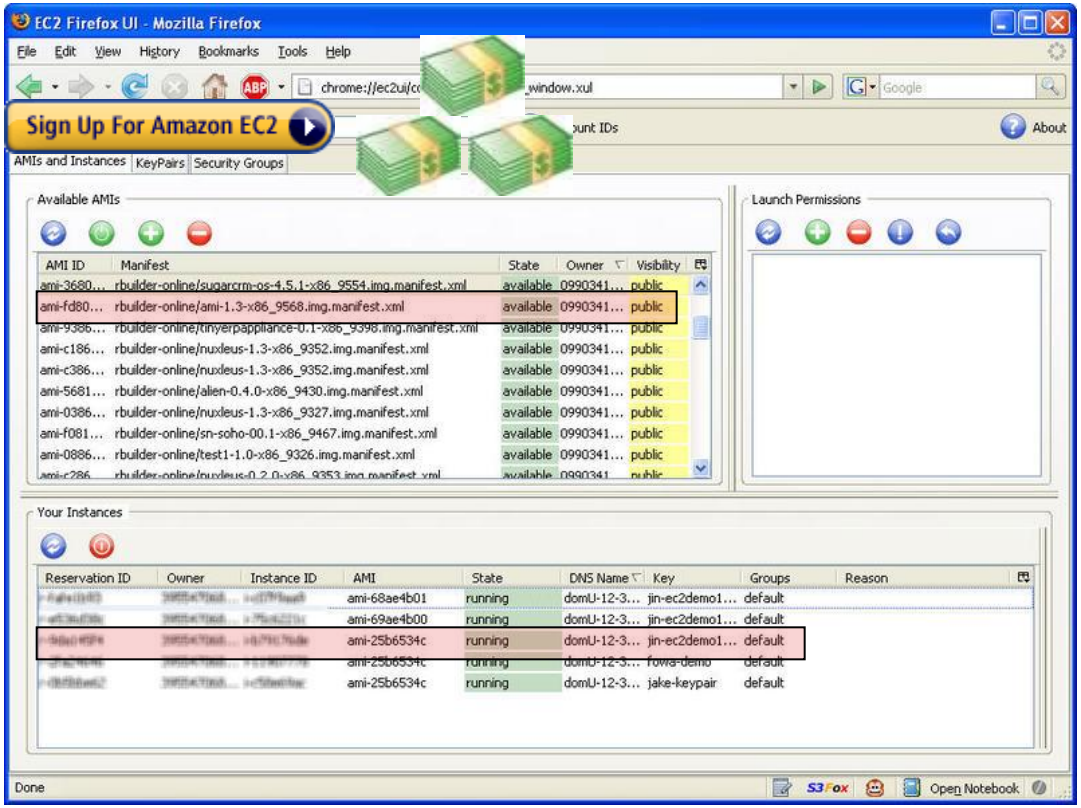
- **Web Application on Google's infrastructures**
- **Application Environment**
  - Sandbox: **secure environment** that distributes **web requests for the application across multiple servers** and starts/stops servers to meet traffic demands
  - Python runtime environment
  - Datastore service
  - Google Accounts Integration
- **Preview period**, only free accounts are available
  - 500 MB and up to 5 million page views a month
  - Up to 3 applications
  - Scalable quotas
- **What you need is**
  - **Google App Engine SDK**
  - **Google Account**
  - Text Editor

Amazon EC2

- **Features**
  - AMI Amazon Machine Image
    - Use pre-configured, templated images to get up and running immediately.
    - Create image containing applications, libraries, data and associated configuration settings
    - Restriction: Linux-based Images
  - Amazon S3 (Simple Storage Service)
    - Providing safe (?), reliable (?) and fast (?) repository to store the AMIs
  - Amazon EC2 (Elastic Computing Cloud)
    - Web service that lets the user requisition AMIs
- **Price**
  - Pay only for the resources that are used
    - Different SLAs
      - Small, Large, Extra Instances
      - Data Transfer Levels
    - Different prices
  - SLAs example
    - Small Instance
      - $0,10 per instance-hour
      - 1.7 GB of memory
      - 1 EC2 Compute Unit
      - 160 GB of instance storage
      - 32-bit platform

Amazon EC2

- How does it work?
  - Subscribe account
  - Get Firefox Plug-In
  - Run your image
    - Example
      - Fedora Core 4
      - Apache
      - MySQL
  - Manage it

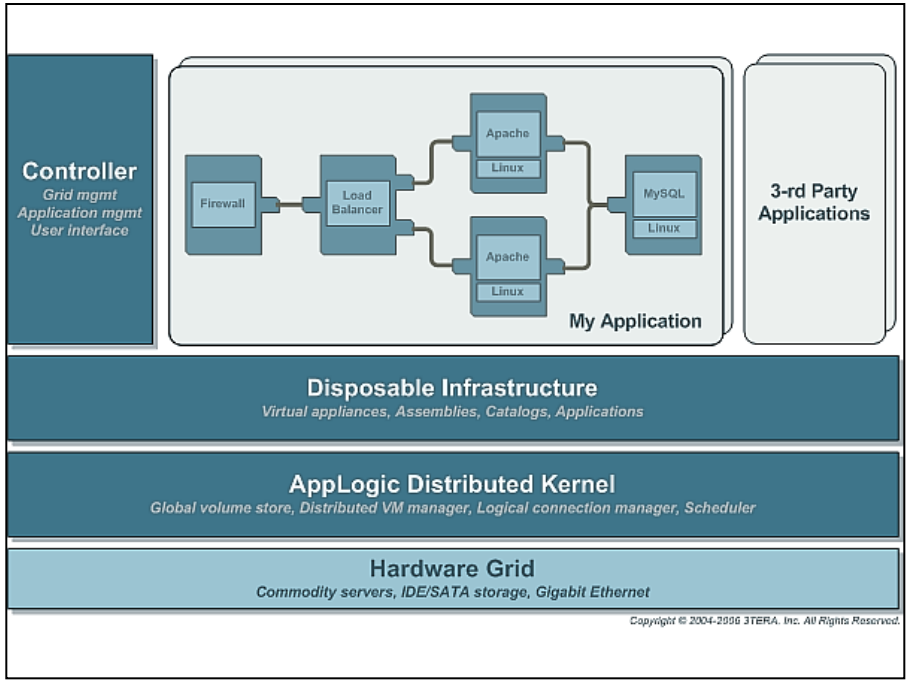


3Tera

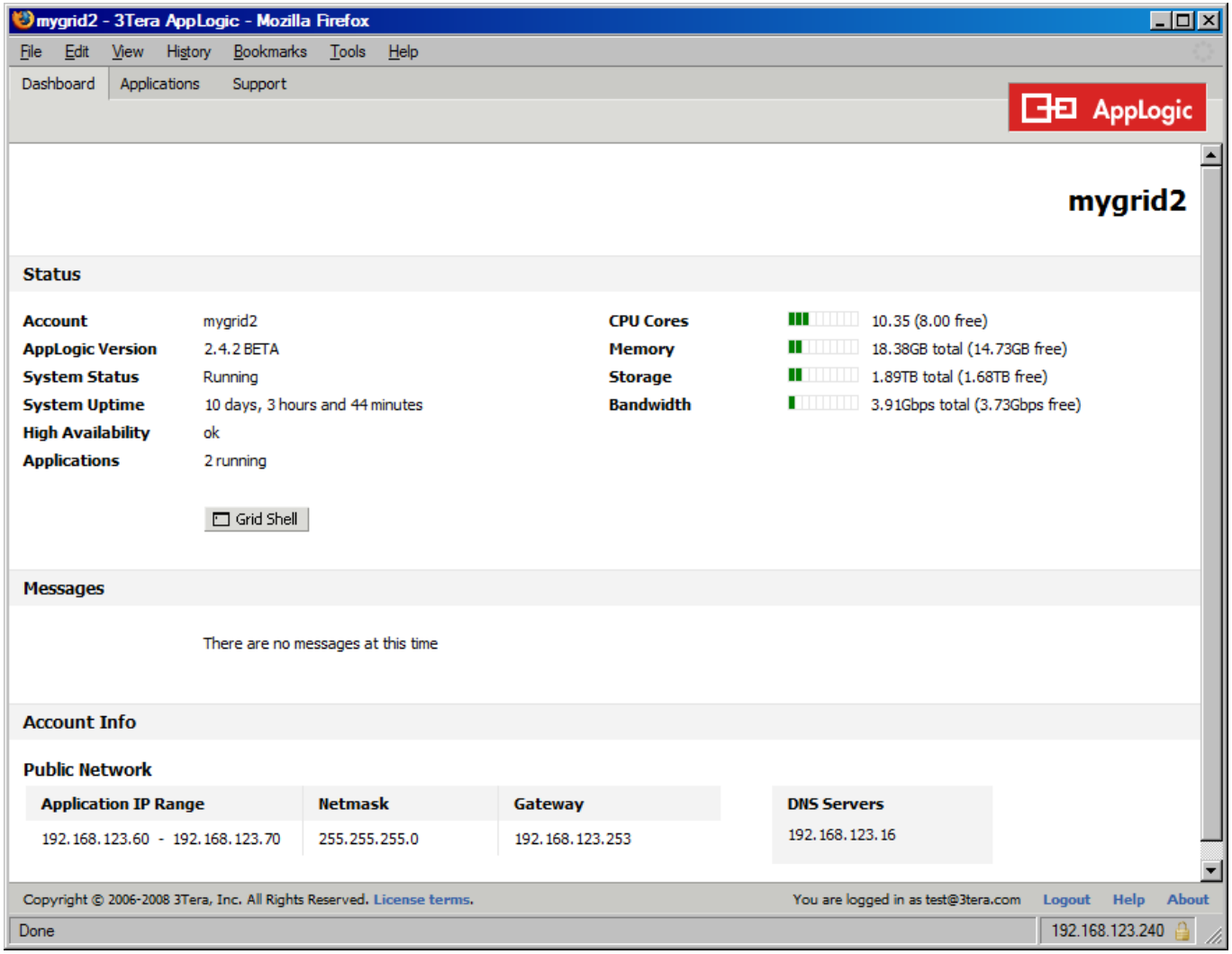
Architecture intended to provide an **open framework to** allow the **development of a cloud computing environment** that's rigorous enough to take both web or enterprise application

- Configuration options
- Architecture

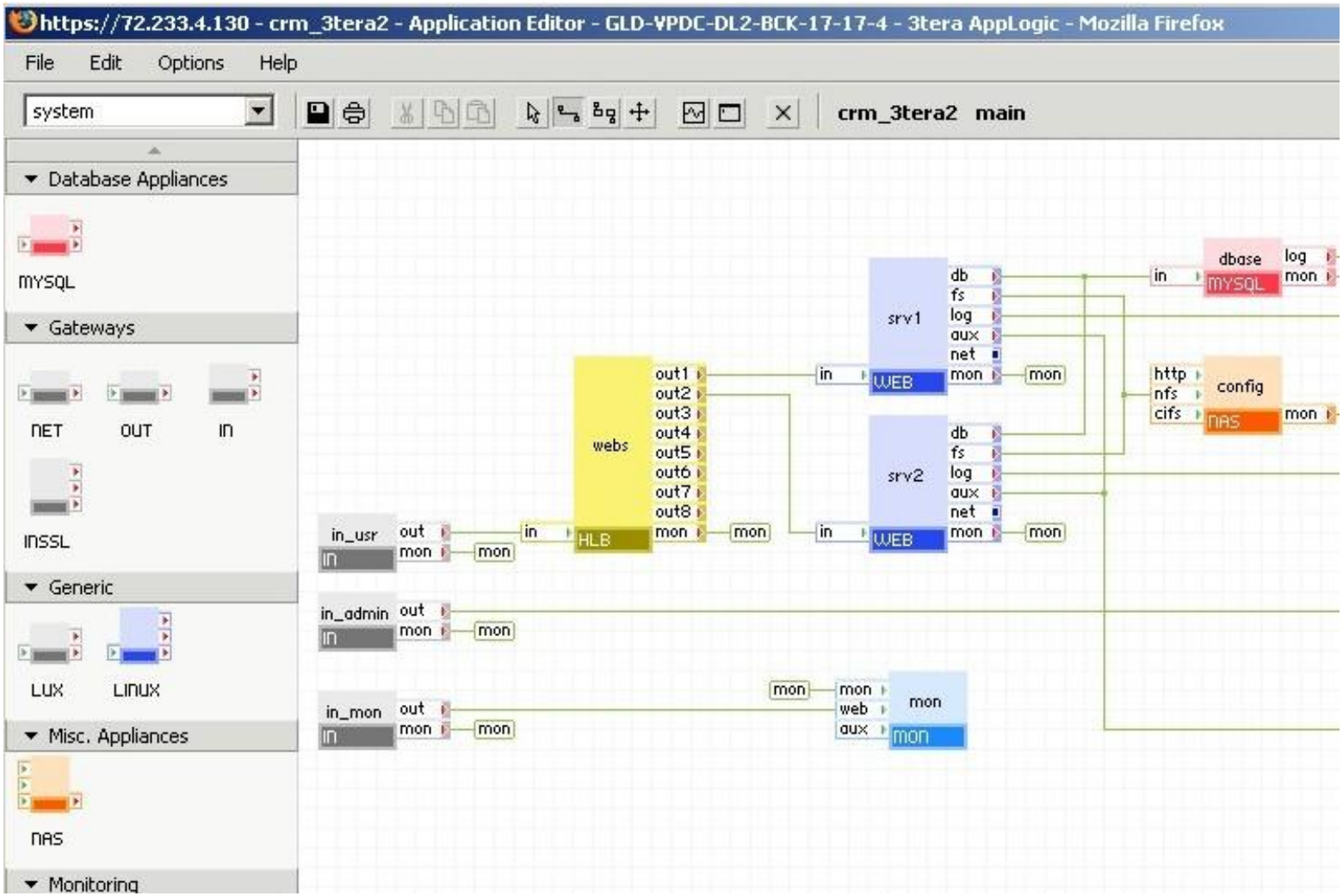
Resource	Min	Max
CPU's	2	1024
RAM, GB	2	2048
Storage, GB*	750	512,000
IP addresses	32	1024



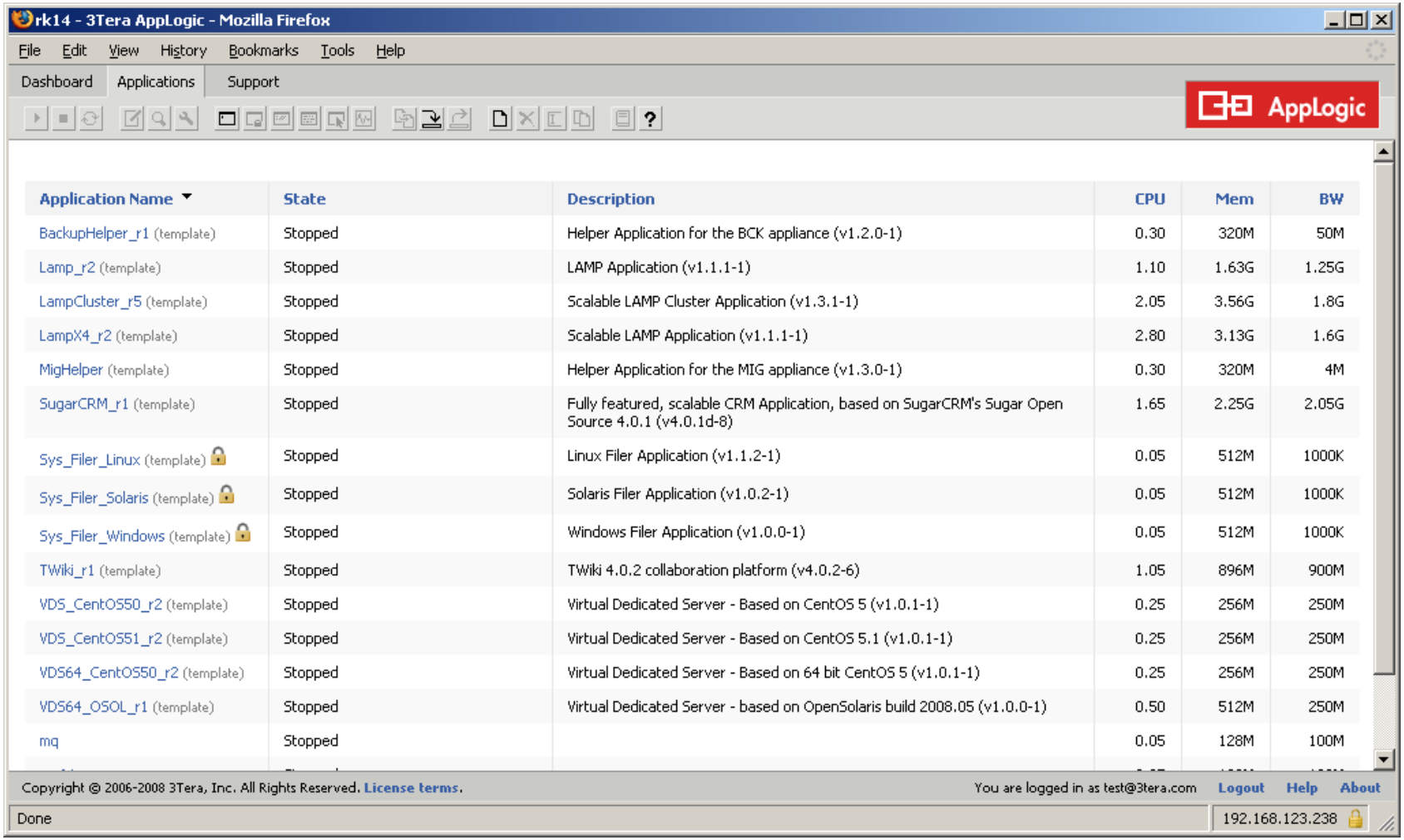
3Tera


mygrid2 - 3Tera AppLogic - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Dashboard Applications Support
AppLogic
mygrid2
Status
Account mygrid2
AppLogic Version 2.4.2 BETA
System Status Running
System Uptime 10 days, 3 hours and 44 minutes
High Availability ok
Applications 2 running
CPU Cores 10.35 (8.00 free)
Memory 18.38GB total (14.73GB free)
Storage 1.89TB total (1.68TB free)
Bandwidth 3.91Gbps total (3.73Gbps free)
Grid Shell
Messages
There are no messages at this time
Account Info
Public Network
Application IP Range
Netmask
Gateway
DNS Servers
192.168.123.60 - 192.168.123.70
255.255.255.0
192.168.123.253
192.168.123.16
Copyright © 2006-2008 3Tera, Inc. All Rights Reserved. License terms.
You are logged in as test@3tera.com Logout Help About
Done
192.168.123.240

3Tera


https://72.233.4.130 - crm_3tera2 - Application Editor - GLD-VPDC-DL2-BCK-17-17-4 - 3tera AppLogic - Mozilla Firefox
File
Edit
Options
Help
system
crm_3tera2 main
Database Appliances
MYSQL
Gateways
NET
OUT
IN
INSSL
Generic
LUX
LINUX
Misc. Appliances
NAS
Monitoring
srv1
srv2
webs
HLB
WEB
dbase
MYSQL
config
NAS
in_usr
in_admin
in_mon
db
fs
log
aux
net
mon
out
in
http
nfs
cifs
web

3Tera


rk14 - 3Tera AppLogic - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Dashboard Applications Support
AppLogic
Application Name | State | Description | CPU | Mem | BW
BackupHelper_r1 (template) | Stopped | Helper Application for the BCK appliance (v1.2.0-1) | 0.30 | 320M | 50M
Lamp_r2 (template) | Stopped | LAMP Application (v1.1.1-1) | 1.10 | 1.63G | 1.25G
LampCluster_r5 (template) | Stopped | Scalable LAMP Cluster Application (v1.3.1-1) | 2.05 | 3.56G | 1.8G
LampX4_r2 (template) | Stopped | Scalable LAMP Application (v1.1.1-1) | 2.80 | 3.13G | 1.6G
MigHelper (template) | Stopped | Helper Application for the MIG appliance (v1.3.0-1) | 0.30 | 320M | 4M
SugarCRM_r1 (template) | Stopped | Fully featured, scalable CRM Application, based on SugarCRM's Sugar Open Source 4.0.1 (v4.0.1d-8) | 1.65 | 2.25G | 2.05G
Sys_Filer_Linux (template) | Stopped | Linux Filer Application (v1.1.2-1) | 0.05 | 512M | 1000K
Sys_Filer_Solaris (template) | Stopped | Solaris Filer Application (v1.0.2-1) | 0.05 | 512M | 1000K
Sys_Filer_Windows (template) | Stopped | Windows Filer Application (v1.0.0-1) | 0.05 | 512M | 1000K
TWiki_r1 (template) | Stopped | TWiki 4.0.2 collaboration platform (v4.0.2-6) | 1.05 | 896M | 900M
VDS_CentOS50_r2 (template) | Stopped | Virtual Dedicated Server - Based on CentOS 5 (v1.0.1-1) | 0.25 | 256M | 250M
VDS_CentOS51_r2 (template) | Stopped | Virtual Dedicated Server - Based on CentOS 5.1 (v1.0.1-1) | 0.25 | 256M | 250M
VDS64_CentOS50_r2 (template) | Stopped | Virtual Dedicated Server - Based on 64 bit CentOS 5 (v1.0.1-1) | 0.25 | 256M | 250M
VDS64_OSOL_r1 (template) | Stopped | Virtual Dedicated Server - based on OpenSolaris build 2008.05 (v1.0.0-1) | 0.50 | 512M | 250M
mq | Stopped | | 0.05 | 128M | 100M
Copyright © 2006-2008 3Tera, Inc. All Rights Reserved. License terms.
You are logged in as test@3tera.com Logout Help About
Done
192.168.123.238

Eucalyptus

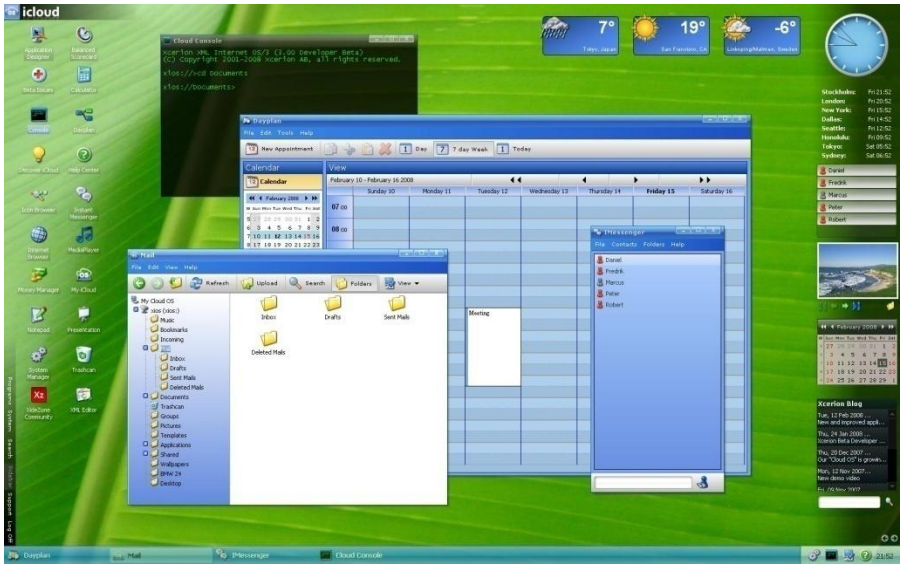
- **Open-source software infrastructure for implementing Cloud computing on clusters**
- Linux systems
- Xen (versions 3.*) for virtualization
- Rocks based (open-source cluster manager)
- Virtual Machines Provisioning

Eucalyptus

- **Eucalyptus Features (1.2 and 1.3)**
  - Installation
    - Rocks-based "almost-one-button" binary install
    - Experts-only "you-are-on-your-own" source install
    - RPM packages for "non-Rocks" RPM based systems
  - Administration
    - Adding/approving/disabling/deleting users (via the Web interface)
    - Adding/listing/disabling images (Web interface with command line use)
    - Adding/deleting nodes and clusters (via edit of configuration files)
  - Amazon's EC2 compatibility:
    - In terms of command-line tools

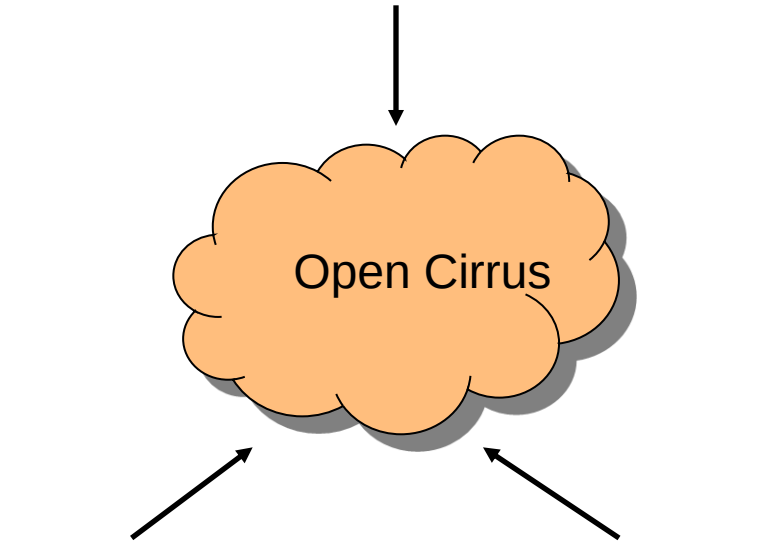
Xcerion

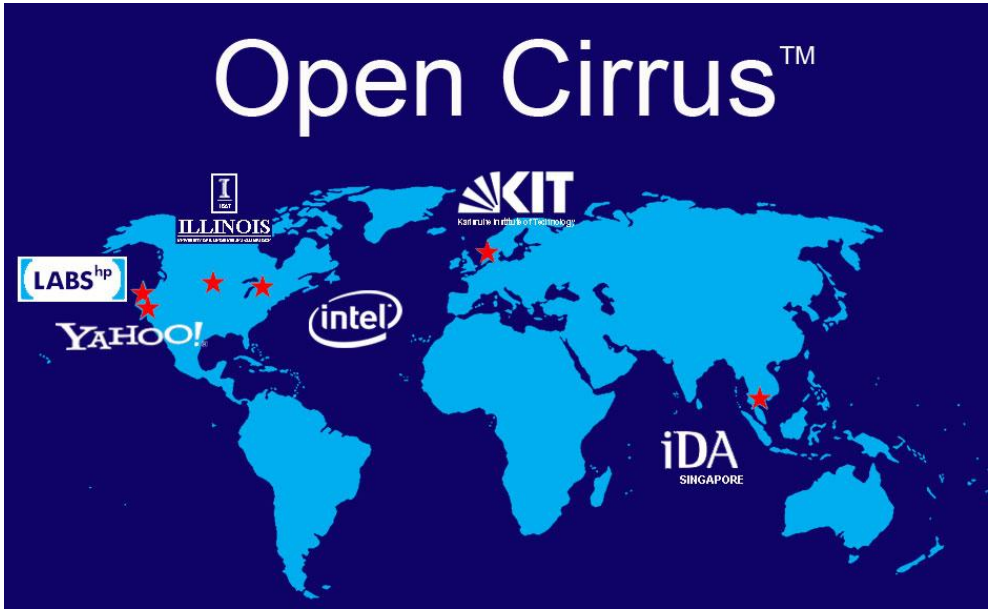
- icloud is an Internet service providing a virtual desktop and OS for free
- Technology
  - XIOS/3 XML Internet Operating System
  - XML Virtual Machine executes the applications locally instead of in the cloud
  - Cloud used for data persistence, storing the users files



Yahoo / Intel / HP test bed

HP (Server e Dispositivi)





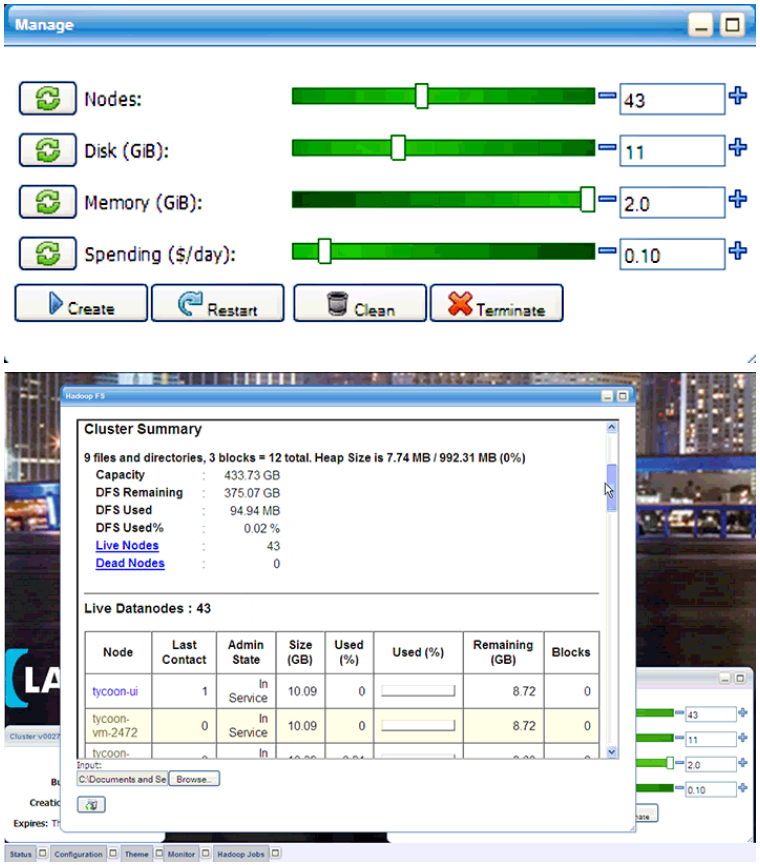
- HP Labs(USA)
- Intel Research (USA)
- Yahoo (USA)
- University of Illinois (USA)
- Karlsruhe institute of Technology (Germany)
- Infocomm Development Autority (Singapore)

Open Cirrus: a Cloud testbed

Merging of various efforts (even research)

- **Goal**: develop an **open-source stack** and **API for Cloud**
- Geographically distributed and **federated** testbed (no centralized management)
- Common services: **Global Single Sign-on**, **Hadoop, Hadoop Distributed File System (HDFS)**
- **Other services: cluster management,** application framework (Pig, MPI, …)
- **System** vs. **application-only:** physical computers (not only virtual machines)
- Validation via heterogeneity

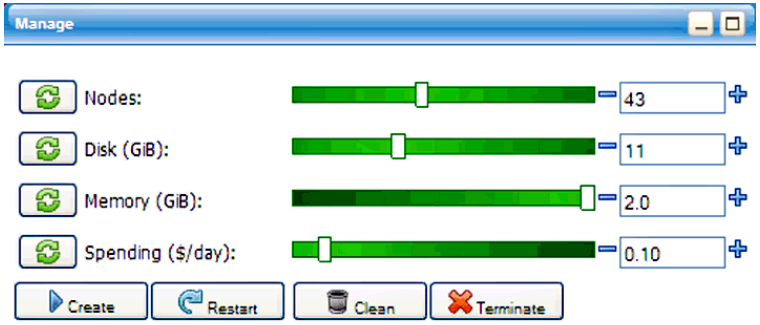
Management Cluster service

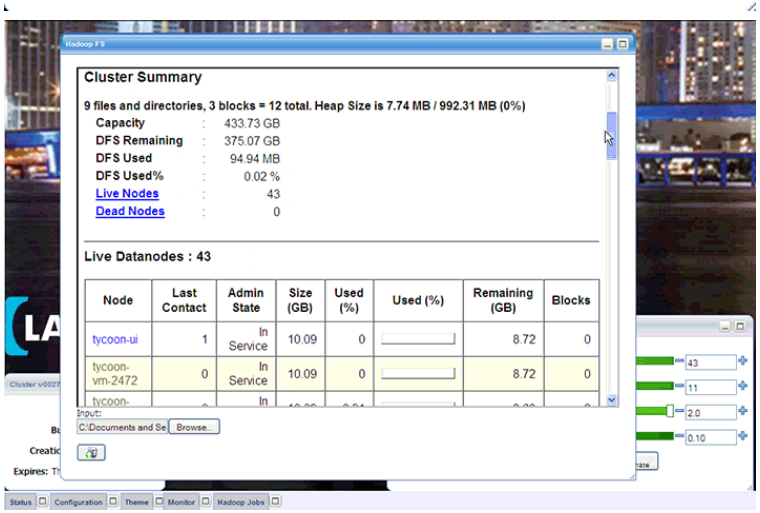


HP/Yahoo/Intel Test Bed

Different projects on Cloud Computing

- Configure the **number of hosts** and their memory and disk
- Specify how much it's willing **to pay for the virtualized cluster** through a **spending rate**
- All of these variables **can be changed at any point** without interrupting running jobs. Increasing the spending rate will immediately increase the CPU share on the cluster nodes
- Possible **Hadoop** Integration



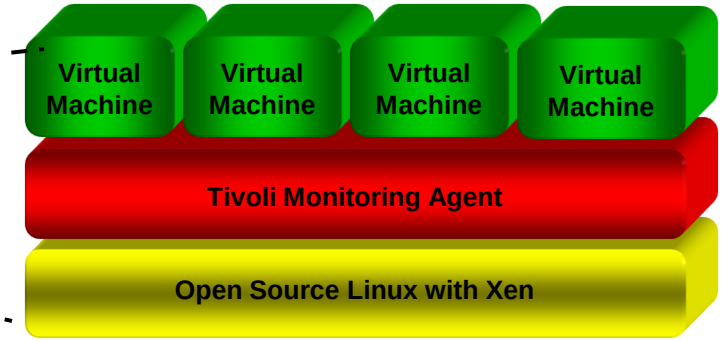


IBM Blue Cloud

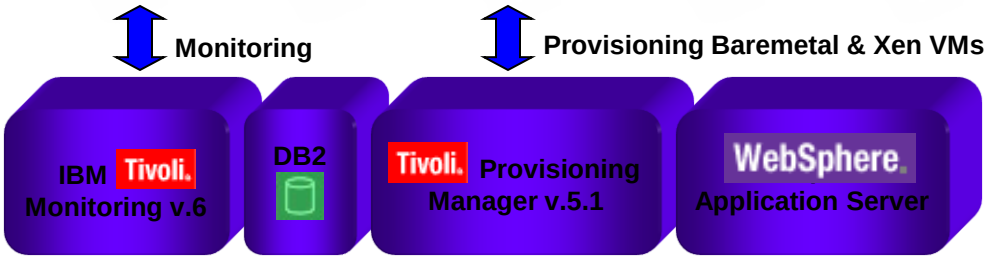
- "**self-service**" model for the requested and managed computational resources
- **automatic image-based provisioning of Xen VMs** (Virtual machinesTivoli)
- **completely automatic** real-time **monitoring** (Tivoli)
- big pool management of **virtual shared resources**
- plug-and-play for the **Xen hosting platform**: auto discovery & configuration (Tivoli)
- **automatic** and **easily configurable software provisioning** (console for simple SLA)
  - WebSphere Application Server Network Deployment
  - DB2 Enterprise Server Edition
  - PHP, MySQL, IIS, MS SQL Server …

Basic Cloud Computing Architecture

Data Center



Virtualized Infrastructure Based on Open Source Linux & Xen



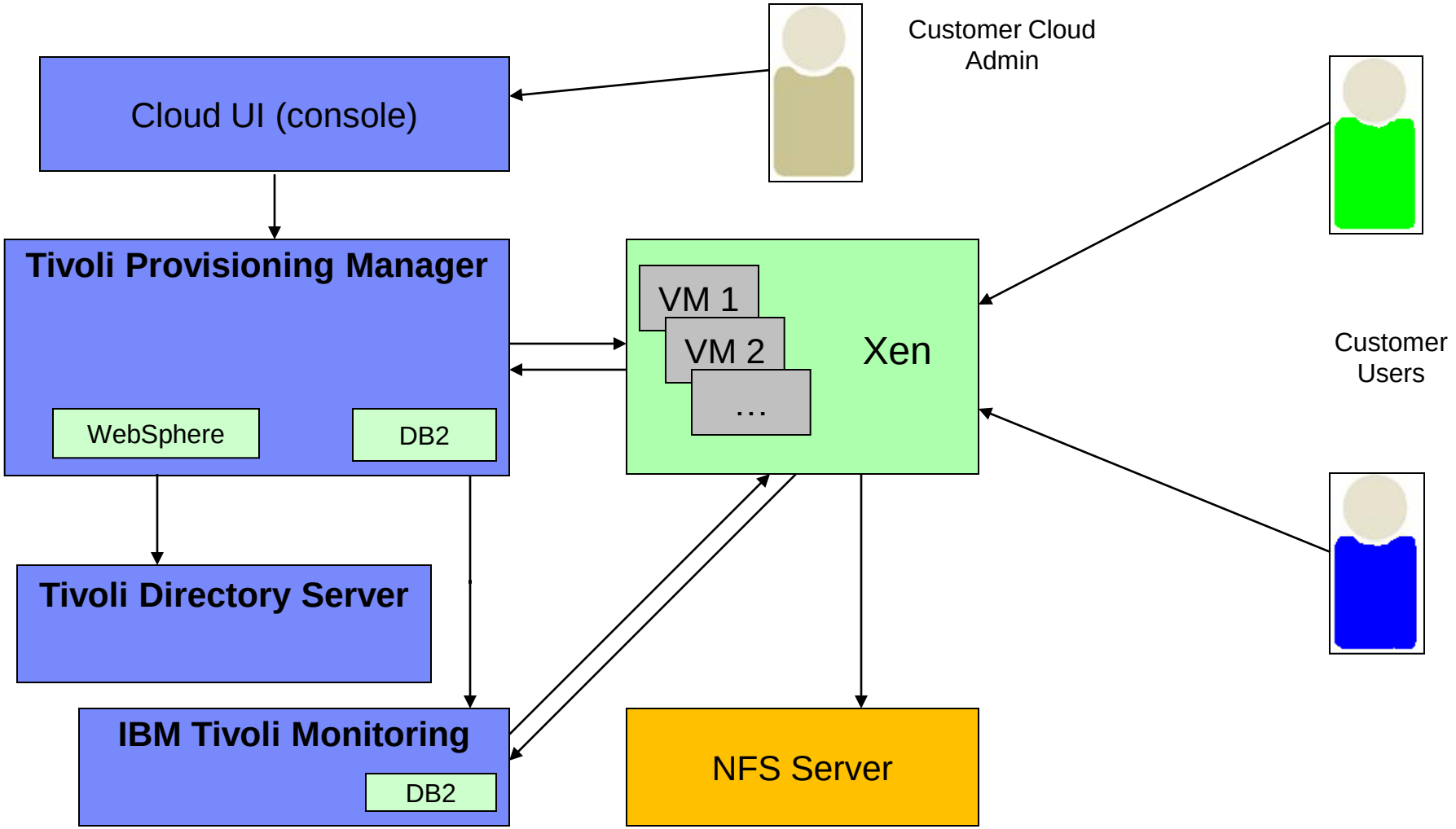
Provisioning Management Stack

Virtualization: all physical machines act as virtual machine hosts; all workloads run on virtual machines

Provisioning: dispense preloaded virtual machines in a few minutes

Monitoring: ensure that systems that go down are recycled quickly

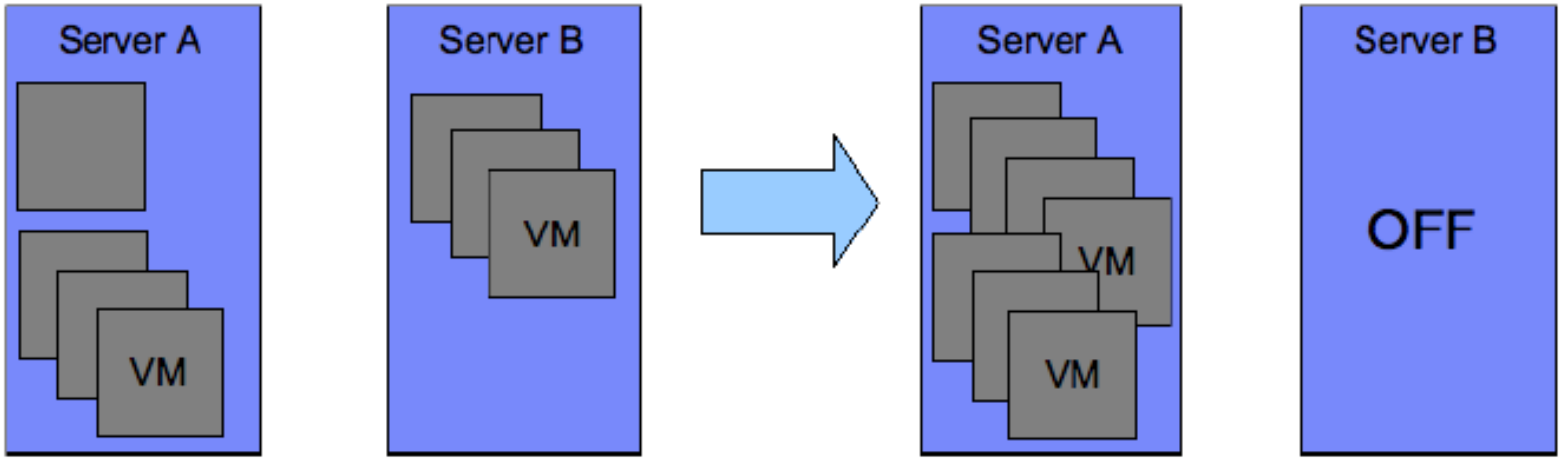
IBM Cloud: architettura di base



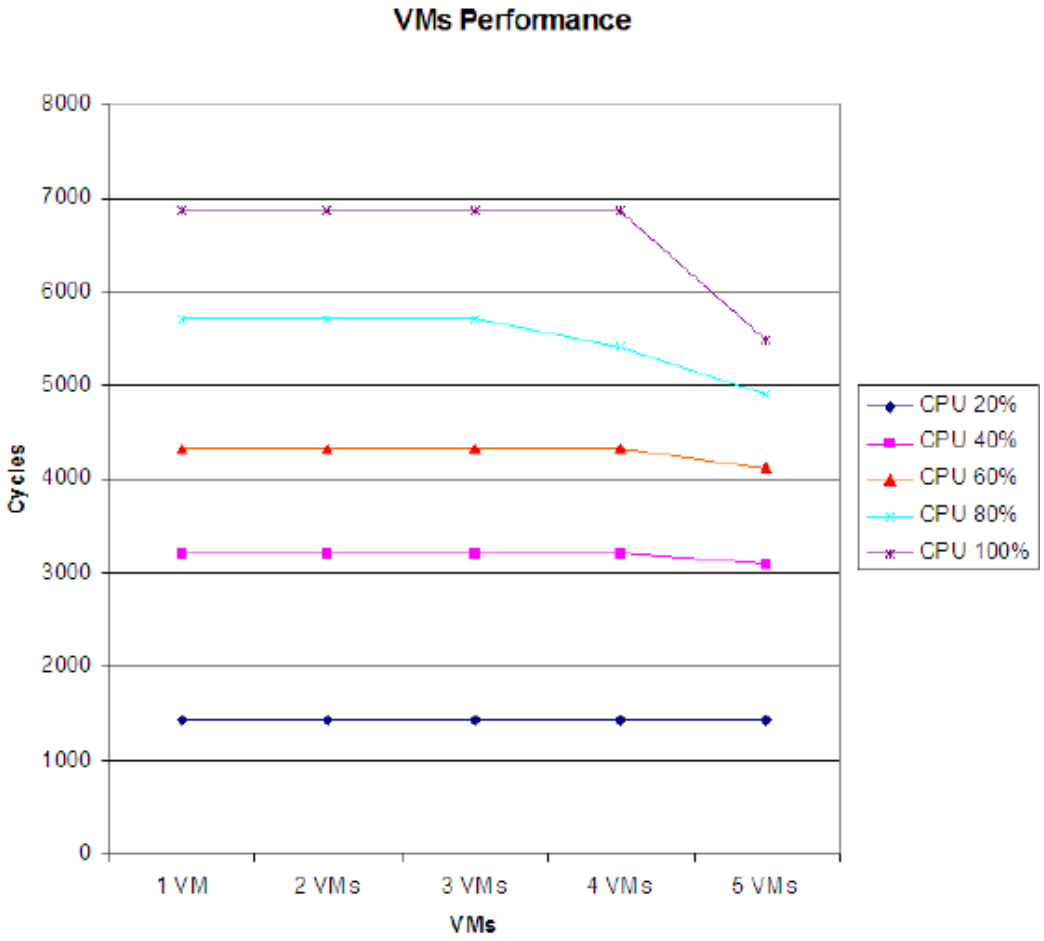
Our experience with the cloud

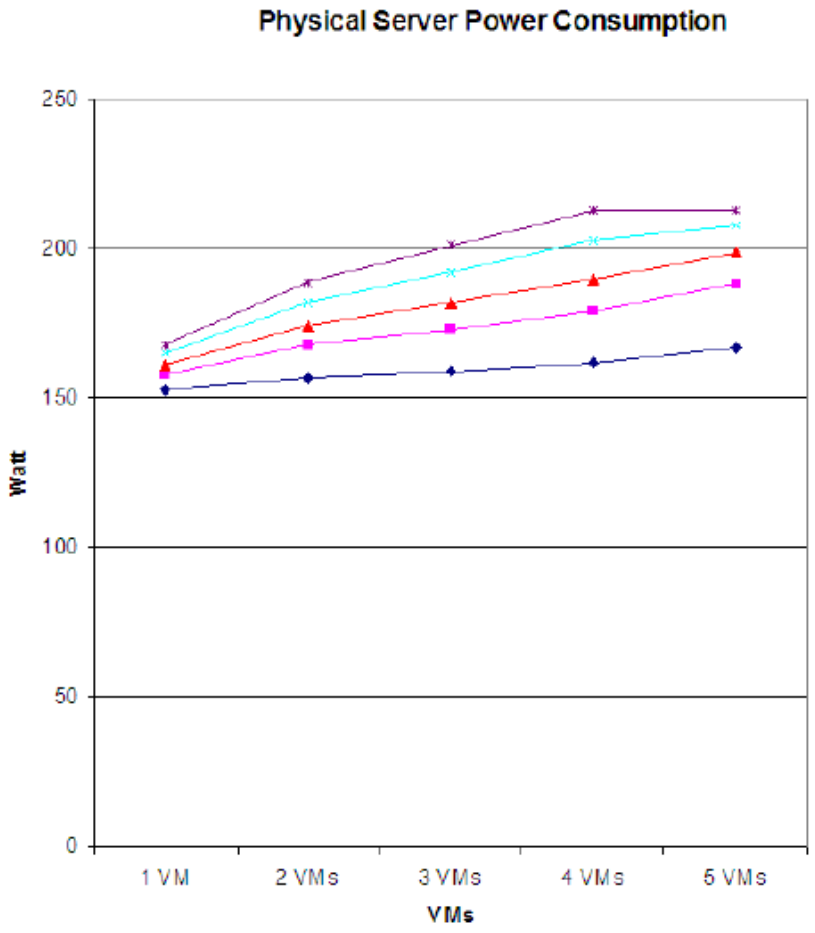
We are working with the **IBM Cloud Computing Center** at Dublin

- Virtual machine (VM) consolidation for power-saving
- Pervasive computing environment with a high number of VMs, e.g., one proxy-VM for each node…



Our experience with the cloud:
some preliminary results

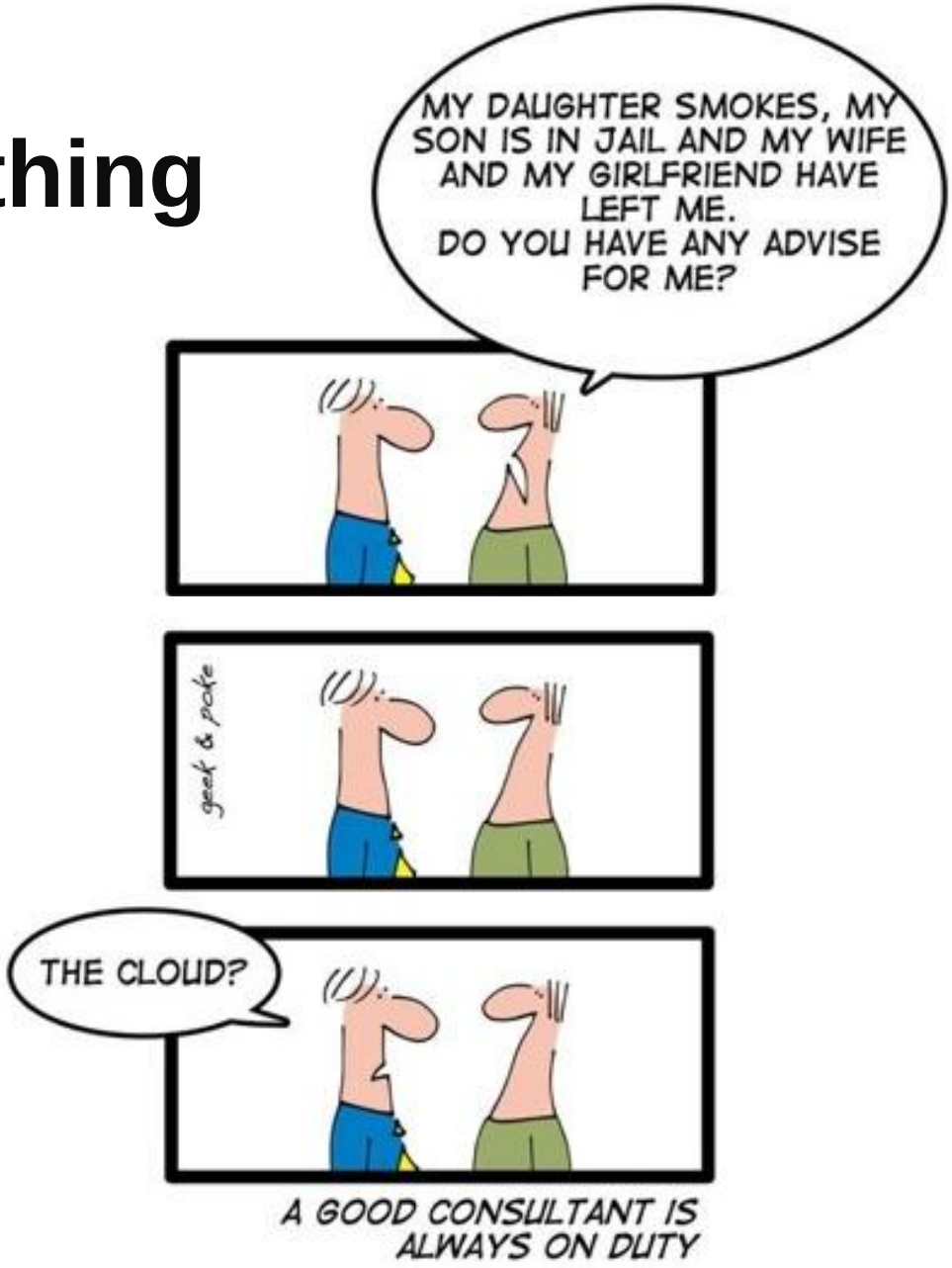




Cloud computing should be…

- **Main requirements**
  - **Scalability on demand** (elastic and highly virtualized resources/images, Service Level Agreements – SLA, …)
  - **Automated provisioning and ease-of-use** (utility computing + infrastructure, platform, and software as a service)
  - **Cost efficiency** (minimized startup costs, energy-saving,…)
- **Challenges**
  - **Management** (system resources, **power-saving**, …)
  - **Interoperability and portability** (data, applications, and virtualized images)
  - **Metering and monitoring** (dynamic monitoring of used resources, accounting, …)
  - **Security**

Cloud for everything



Not exactly
for everything ☺

The fog has gone…

… and **clouds** are disclosed into the **sky**!

Thanks for your attention!