Introduction to Cloud Technology

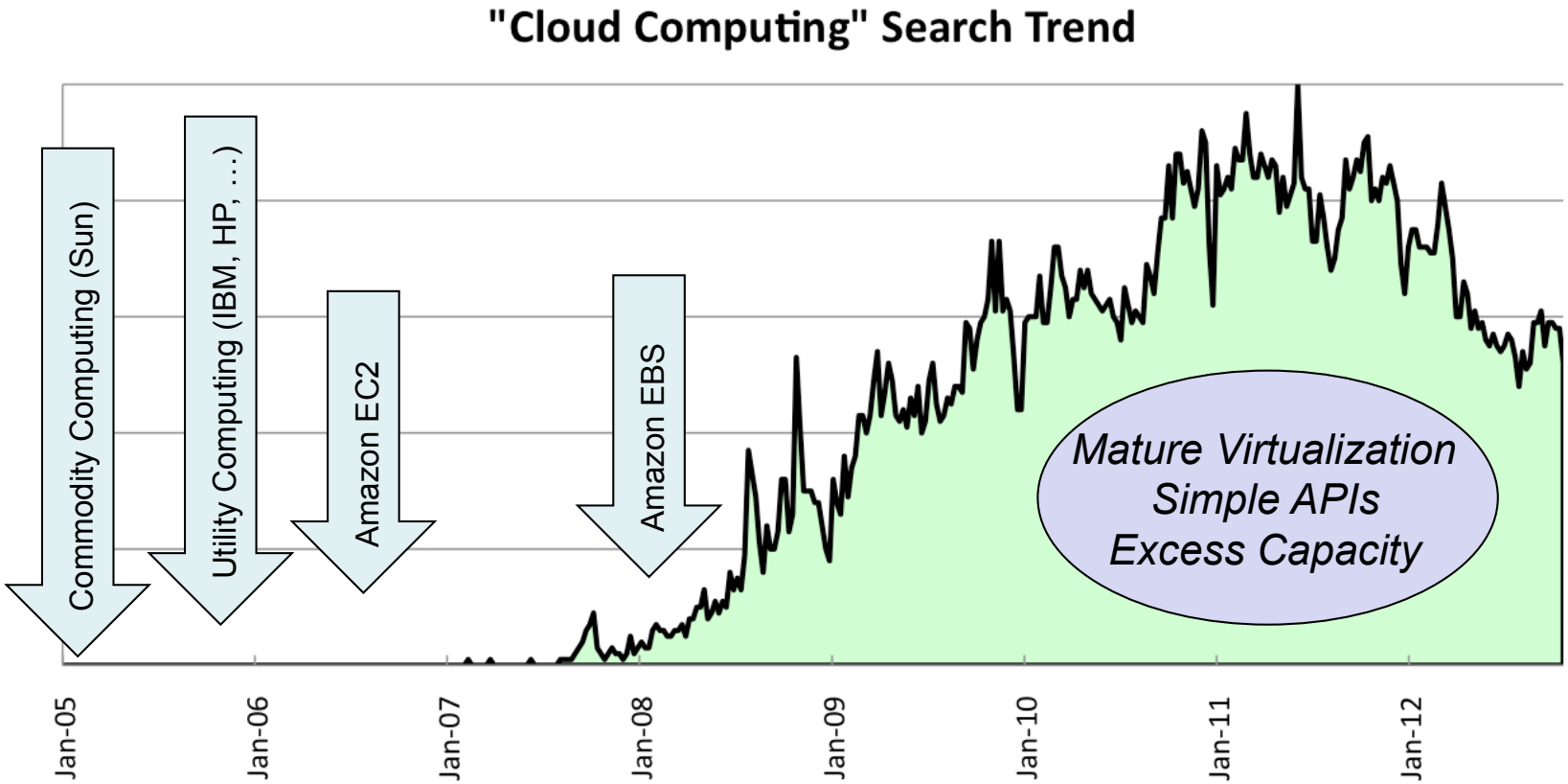
StratusLab Tutorial (Orsay, France)

28 November 2012

Cloud Marketing

“Cloud” is currently very trendy, used everywhere

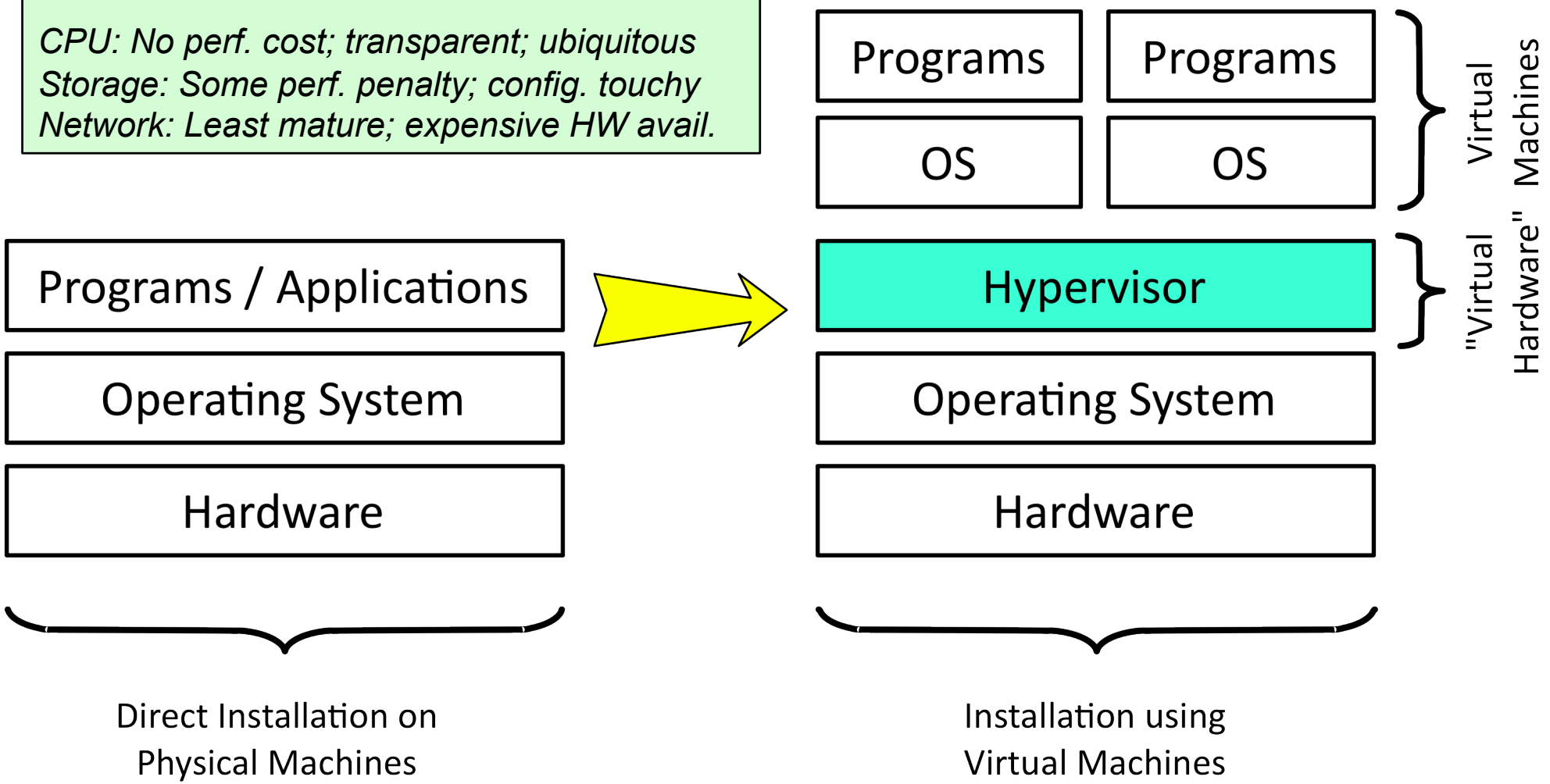
- Many definitions that are often incompatible
- Used (often) to market pre-existing (non-cloud) software



Virtualization

Different Maturity Levels

CPU: No perf. cost; transparent; ubiquitous
Storage: Some perf. penalty; config. touchy
Network: Least mature; expensive HW avail.



Web Service Interfaces

‘Traditional’ Web Services

- Prioritizes easier implementation by developers
- Very complex specifications, with (sadly) limited interoperability
- Similarly complex tooling for developers
- Supports only RPC architecture

REST, XMLRPC, HTTP Query APIs

- Prioritizes easy access by clients
- Universal language support by relying on std. HTTP protocol
- Both RPC and Resource Oriented Architectures possible

Excess Computing Capacity

Amazon

- Dimensioned to handle Christmas rush
- Idle machines/resources other times of year
- Monetize investment in these services
- *Allowed resources to be offered at excellent prices*

Dedicated Data Centers

- Moved from monetizing existing investment to profit center
- Now Amazon and others have dedicated centers for the cloud!

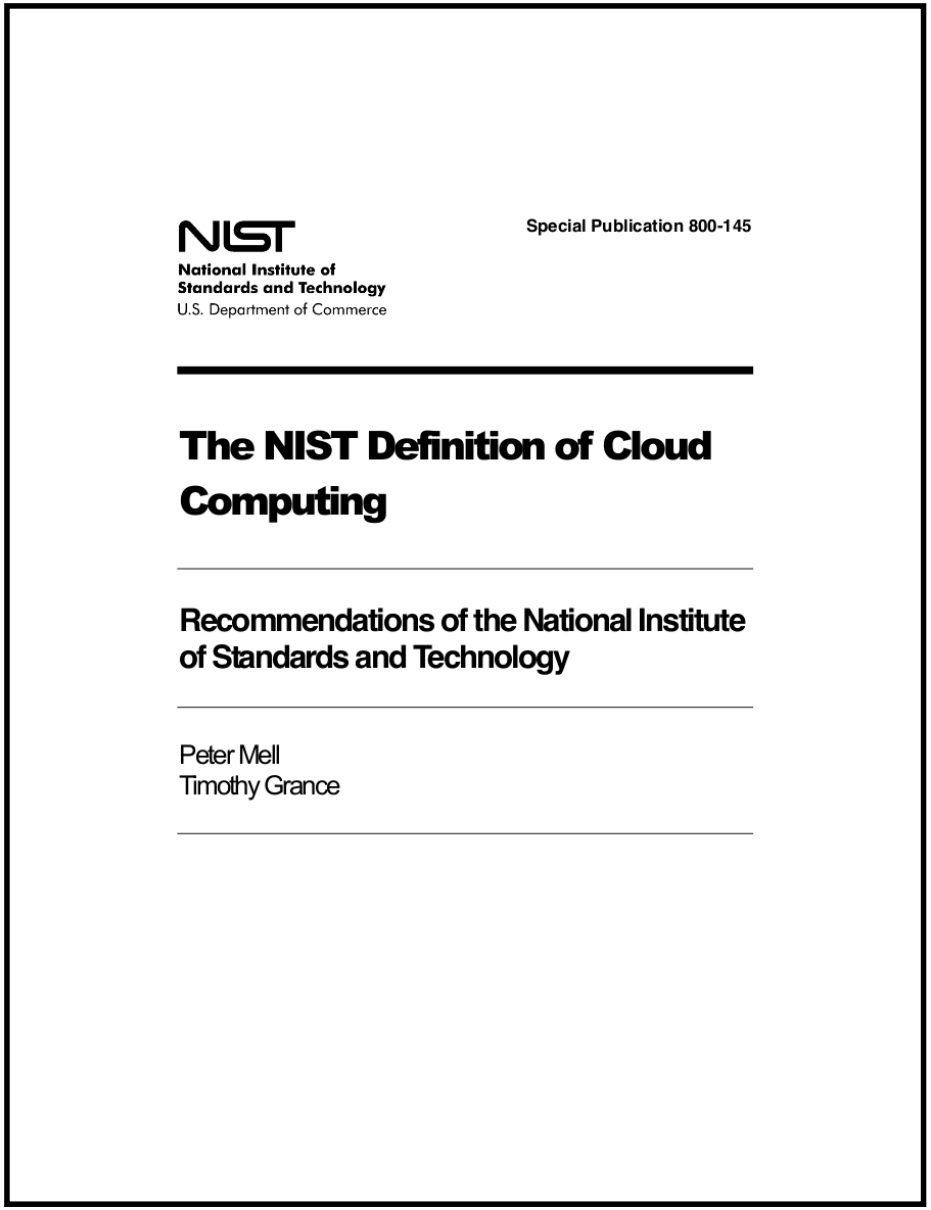
What is a Cloud?

NIST: Best Definitions

- Essential characteristics
- Service models
- Deployment models

- Just 2 pages of text!

http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf


NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 800-145

The NIST Definition of Cloud Computing

Recommendations of the National Institute of Standards and Technology

Peter Mell
Timothy Grance

Essential Characteristics

On-demand self-service

- Users provision computing resources without human intervention

Broad network access

- Fast, reliable access to remote (cloud) resources via the network

Rapid elasticity

- Ability to scale the resources rapidly based on application needs

Resource pooling

- Multi-tenant sharing of resources

Measured service

- Control and optimization of resources through measured use

Other Distributed Computing Systems

Remote Services

- RackSpace, etc.
- Separates service management from hardware management

Volunteer Computing

- BOINC, XtremWeb, etc.
- Takes advantage of idle, private, and volatile resources

Batch Systems

- LSF, PBS, etc.
- Permits worker nodes on different sites, but centrally managed

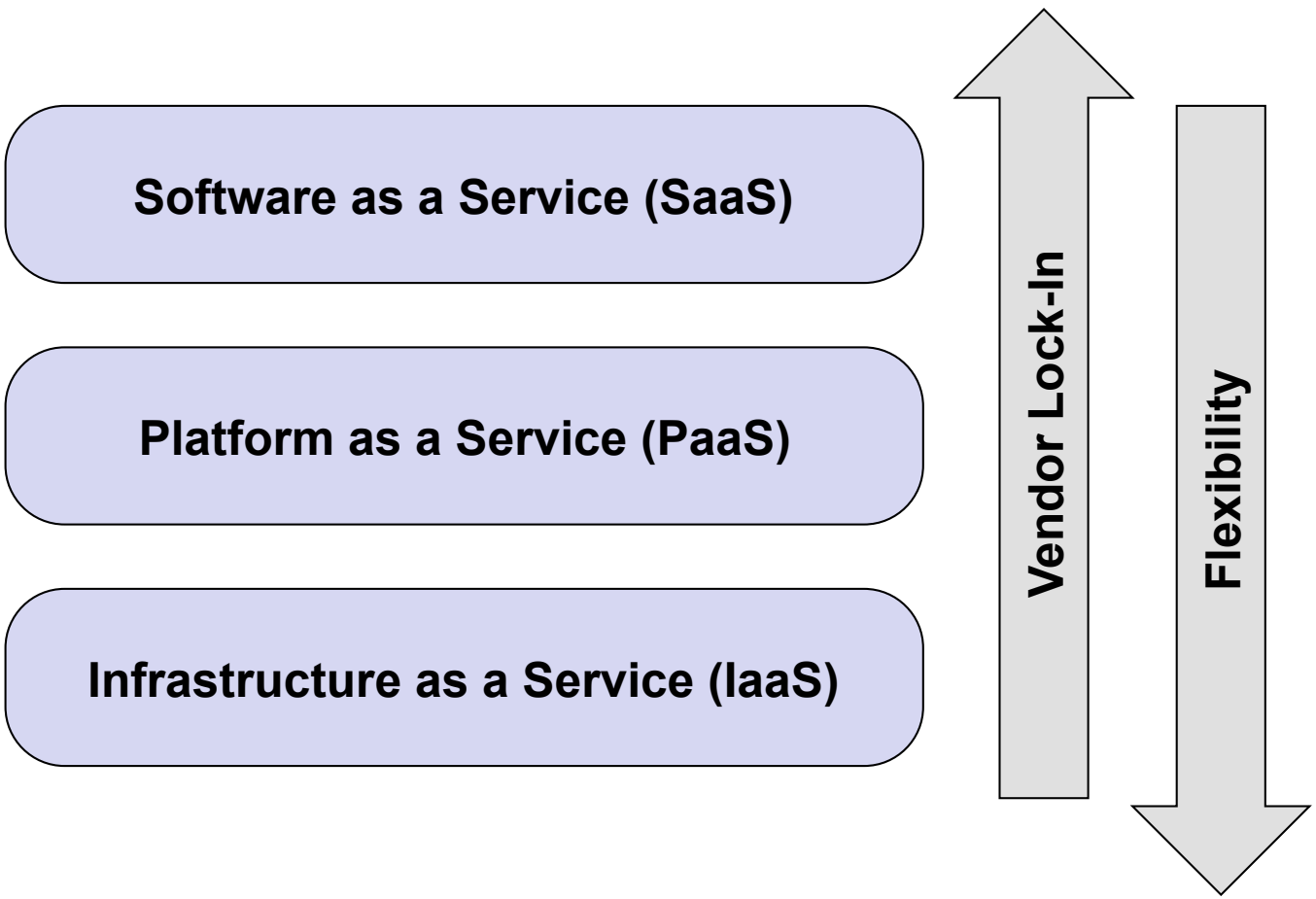
Grid Computing

- European Grid Infrastructure (EGI), Open Science Grid (OSG)
- Federating distributed data centers for easier access, better efficiency

Service Models

What resources are offered
to customers or clients?

Service Models



Software as a Service (SaaS)

Abstraction

- Essentially web-hosting
- Aimed at End-users

Advantages

- Very simple use: web interface with no software installation
- Very accessible: laptop, smartphone, …

Disadvantages

- Questions about data: access, ownership, reliability, etc.
- Integration of different services and novel uses of data are often difficult

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

Doodle®

salesforce.com
Success On Demand.

Platform as a Service (PaaS)

Abstraction

- Platform and infrastructure for creating web applications
- Aimed at developers

Advantages

- Load balancing, automatic failover, etc.
- Programmers can forget about the low-level “plumbing”

Disadvantages

- Restricted number of languages
- Applications are not portable between different providers

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

GridGain CLOUD COMPUTING

Google App Engine

Windows Azure

Infrastructure as a Service (IaaS)

Abstraction

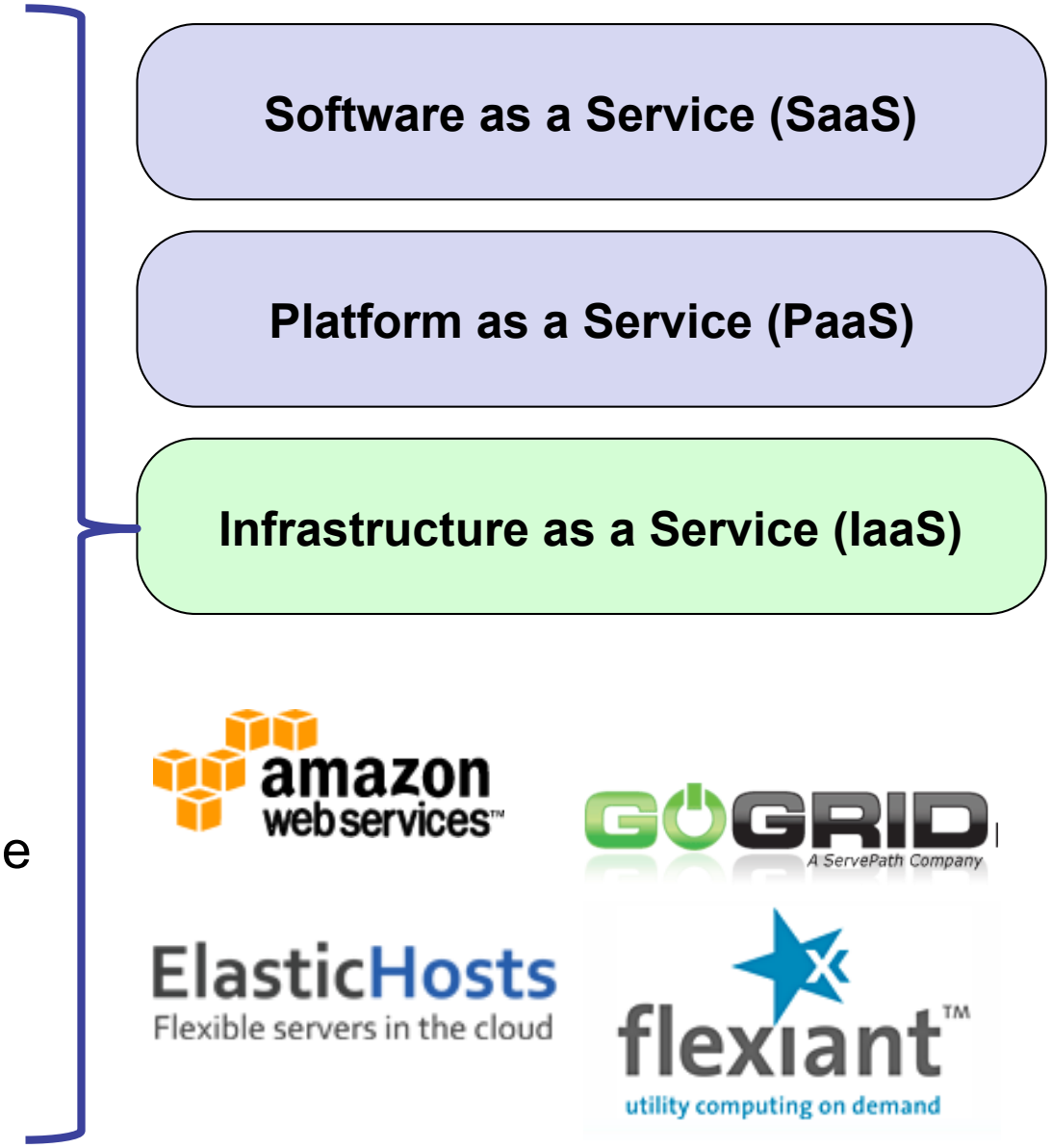
- Access to remote virtual machines
- Aimed at service providers

Advantages

- Customized environment
- Simple and rapid access
- Access as “root”
- Pay-as-you-go model

Disadvantages

- Non-standardized and multiple interfaces (vendor lock-in)
- Virtual machine creation is difficult and time-consuming



Deployment Models

Who are the users and
what ties them together?

Deployment Models

Private

- Single administrative domain, limited number of users
- Resource allocation usually ‘informal’, hallway conversations
- E.g. site uses cloud for standard site services, managed by sysadmins

Community

- Different administrative domains but with common interests/procedures
- Resource allocation usually formalized ‘horse trading’
- E.g. high-energy physics community

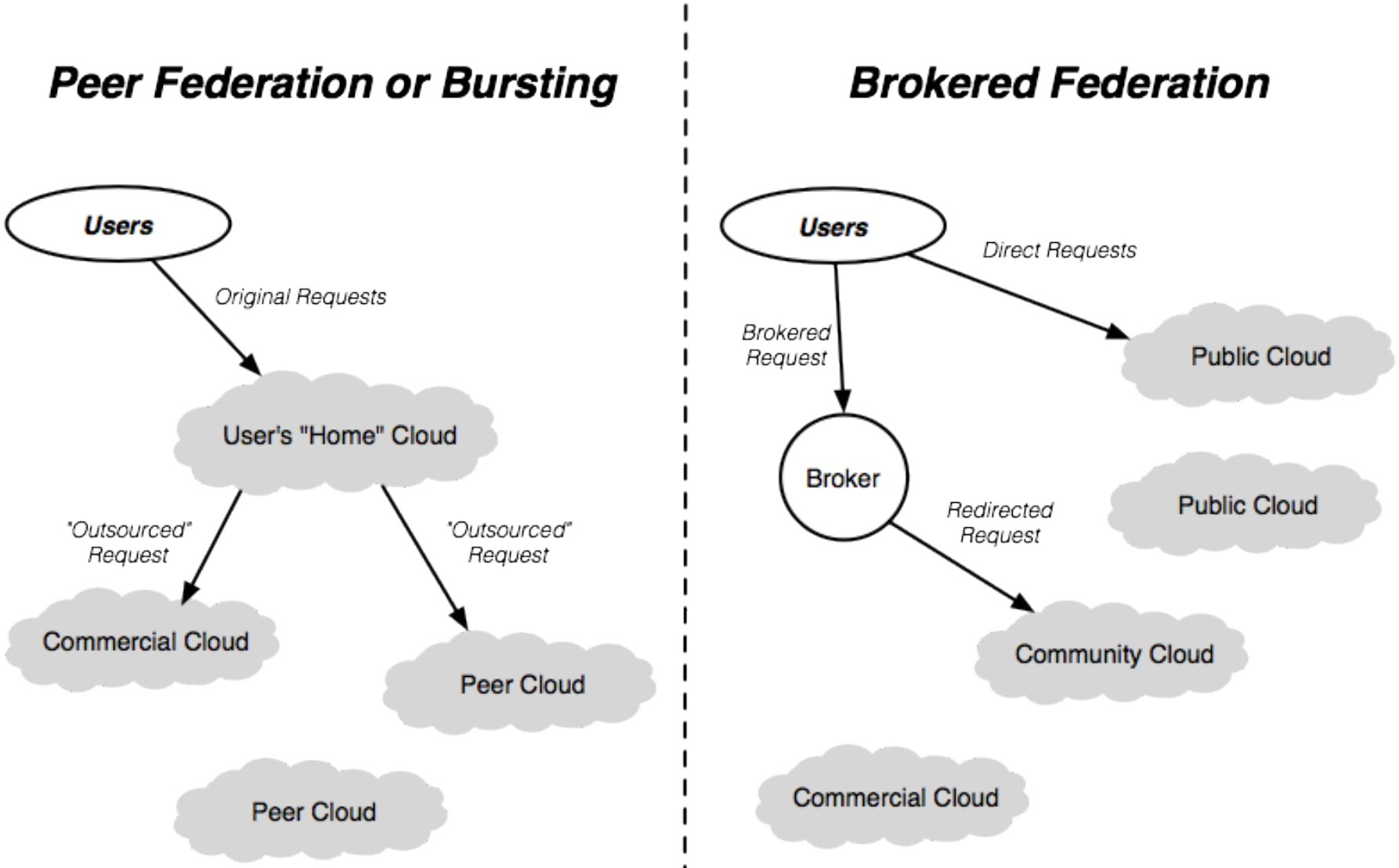
Public

- People outside of institute’s administrative domain, general public
- Resource allocation by payment
- E.g. Amazon Web Services (EC2, S3, …)

Hybrid

- Combination of other deployment models to federate resources

Hybrid Clouds and “Sky” Computing



Questions and Discussion

Exercise: Your Interest in Clouds

Researchers and Engineers (End-users)

- Use existing academic and/or commercial software on cloud
- What scientific domains?

Developers

- Modify existing software to use cloud resources
- Create new software for the cloud
- What types of software?

Administrators

- Provide cloud resources to researchers, engineers, and/or developers
- What types of users? Local, multi-institute, …?

Exercise: Commercial Services

What are the characteristics of clouds?

- Are these clouds: gmail, facebook, twitter, dropbox, iCloud?
- What characteristics does each have?
- What limitations does each have?
- How would these work with scientific computation?

http://www.stratuslab.eu

Copyright © 2012, Members of the StratusLab collaboration.

This work is licensed under the Creative Commons Attribution 3.0 Unported License (http://creativecommons.org/licenses/by/3.0/).