



StratusLab: Darn Simple Cloud

Charles (Cal) Loomis & Mohammed Airaj

LAL, Univ. Paris-Sud, CNRS/IN2P3

29 August 2013

What is it?

- Complete IaaS cloud distribution
- Open source (Apache 2 license)
- Works well for production private and public IaaS clouds

Focus: Darn Simple Cloud

- Simple to install on commodity hardware
- Simple to use, from any client machine
- Scales down as well as up!

Infrastructure as a Service (IaaS)

- + Customized environment
- + Dynamic (scalable) provisioning
- + Easy access
- Variety of APIs and interfaces
- Image creation is tedious
- Single machine granularity



Why are cloud technologies useful?



Users

- Custom environment: no more porting, revalidation of code
- Pre-installed and configured applications
- Rapid, dynamic provisioning of resources
- Complete control over the requested resources

Developers

- Simple access: use of REST and RPC over HTTP(S)
- Elasticity to respond to peaks in demand for applications

Administrators

- Flexible management: separate mgt. of machines and services
- Separation of responsibilities: Hardware / Services / Platforms / Users

Resource Providers

- Better utilization of shared resources
- Federation (outsourcing) possible

Commercial Provider: Amazon Web Services (AWS)

- Leading and largest IaaS service provider
- Improving and adding new services at a phenomenal rate
- Providers differentiate based on price, SLAs, location, etc.

Commercial Cloud Distribution: VM-ware

- Extremely good and complete
- Very expensive, except for ESXi hypervisor (free)

Open Source Cloud Distributions: Many!

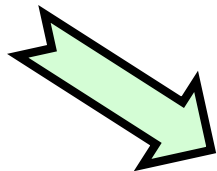
- Essentially none in 2007; now easily a dozen different distributions
- StratusLab, ..., OpenStack, OpenNebula, CloudStack
- Very different levels of maturity, stability, scalability, etc.

IaaS cloud providers all use similar semantics, but different APIs, etc.

Where did it start?

Informal collaboration to investigate running grid services on Amazon EC2 (2007)

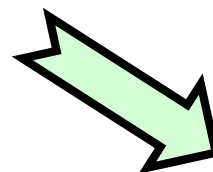
Identified need for open source cloud distribution.



StratusLab Project (6/2010 to 5/2012) co-funded by EC with 6 partners from 5 countries



Production dist. with academic & commercial deployments.



Website: <http://stratuslab.eu>
Twitter: @StratusLab
Support: support@stratuslab.eu
Source: <http://github.com/StratusLab>

Open collaboration
to continue the development and support of the StratusLab software



Post-Project Releases

- V2.1 (16/10): Streamlined release; improved IO perf. with virtio drivers
- V2.1.1 (29/11): Bug fixes; storage upload; better Windows support
- V13.02 (31/1): Support for CloudInit contextualization and bug fixes
- V13.05 (18/6): Initial steps towards new architecture
- V13.09 (30/9): CIMI and new architecture

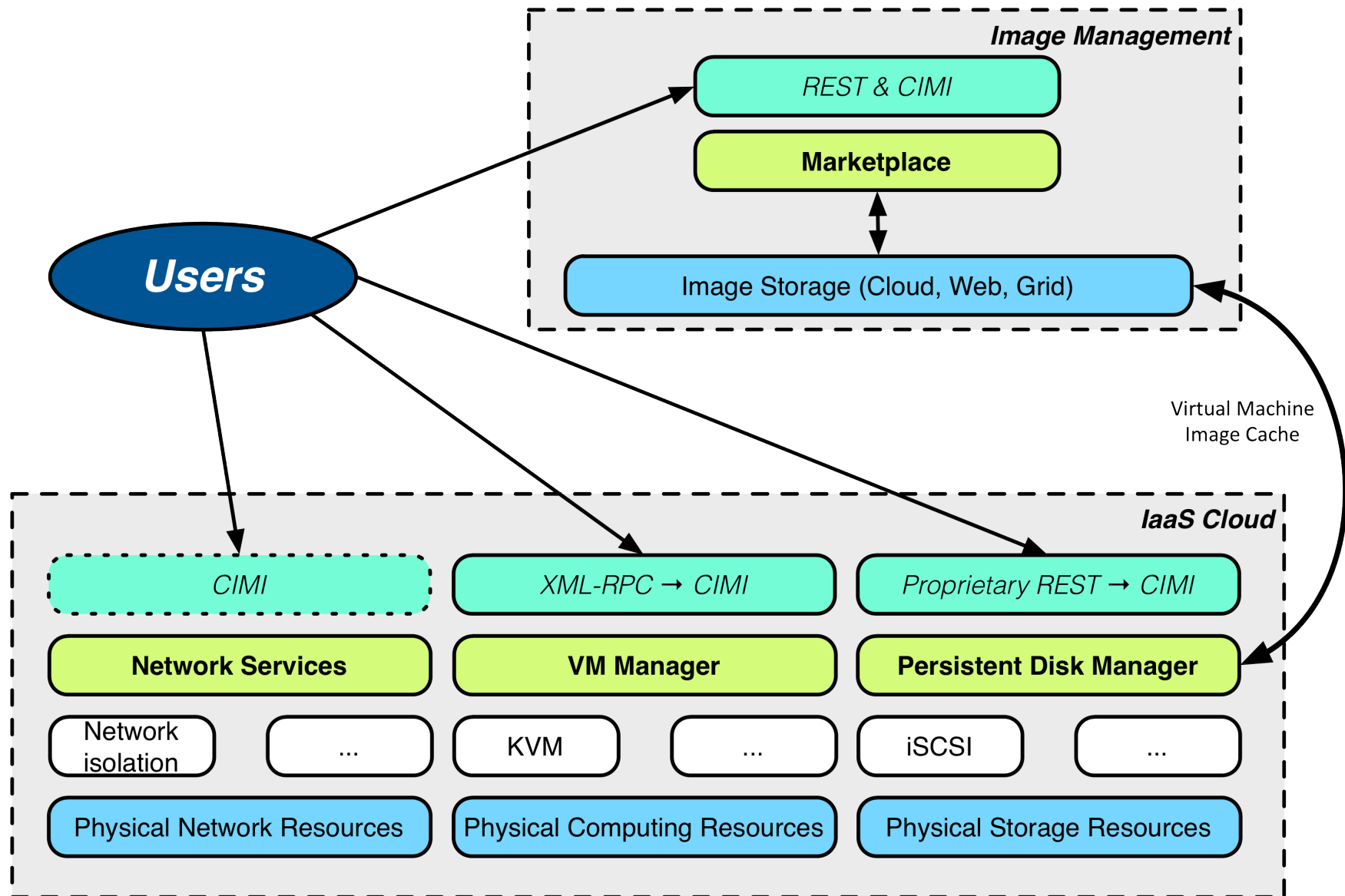
Release Policy

- Quarterly timed releases (13.02, 13.05, ...)
- Intermediate bug fix releases as needed
- Roadmap (6-month) available describing the StratusLab evolution

Support Policy

- Best-effort support for all recent releases, emphasis on latest

StratusLab Services





Services

- Compute: Virtual machine management (currently uses OpenNebula)
- Storage: Volume-based storage service
- Network: Simple configuration for public, local, and private VM access
- Image mgt.: Complete system for trusted sharing of VM images

Tools

- Python CLI and APIs (Libcloud) to facilitate use of cloud
- CLI to facilitate the installation of services

Service Details

Compute



Features

- Fast provisioning of VMs, with low latency start-up



Contextualization

- HEPiX & OpenNebula CDRom contextualization by default
- CloudInit (disk based) also supported

Implementation

- API: XML-RPC interface of OpenNebula
- OpenNebula (C++, Ruby) with customized hooks
- Hooks primarily for caching, snapshots, and storage access



Features

- Volume abstraction for storage service
- Provide users with persistent storage for data
- Serves also as cache of images for VM instances
- (No file-based or object-based storage service)



Implementation

- API: Proprietary REST interface with CRUD actions
- Java-based service using MySQL database for state information
- Can use iSCSI or shared file system for physical storage
- Can use simple files or LVM volumes for disk content

Features

- Support 3 specific use cases: public service (public), batch system (local), and BOINC-like worker (private)
- Dynamic configuration of network switches not needed
- Uses usual services for VM network configuration



Implementation

- No API: manual, static configuration of network
- Rec. configuration: VLAN for cloud services separate VLAN for VMs
- All classes of IP addresses are optional, can create other classes
- Uses DHCP for VM network configuration
- Users responsible for protecting their machines

Marketplace & Image Handling



Priorities

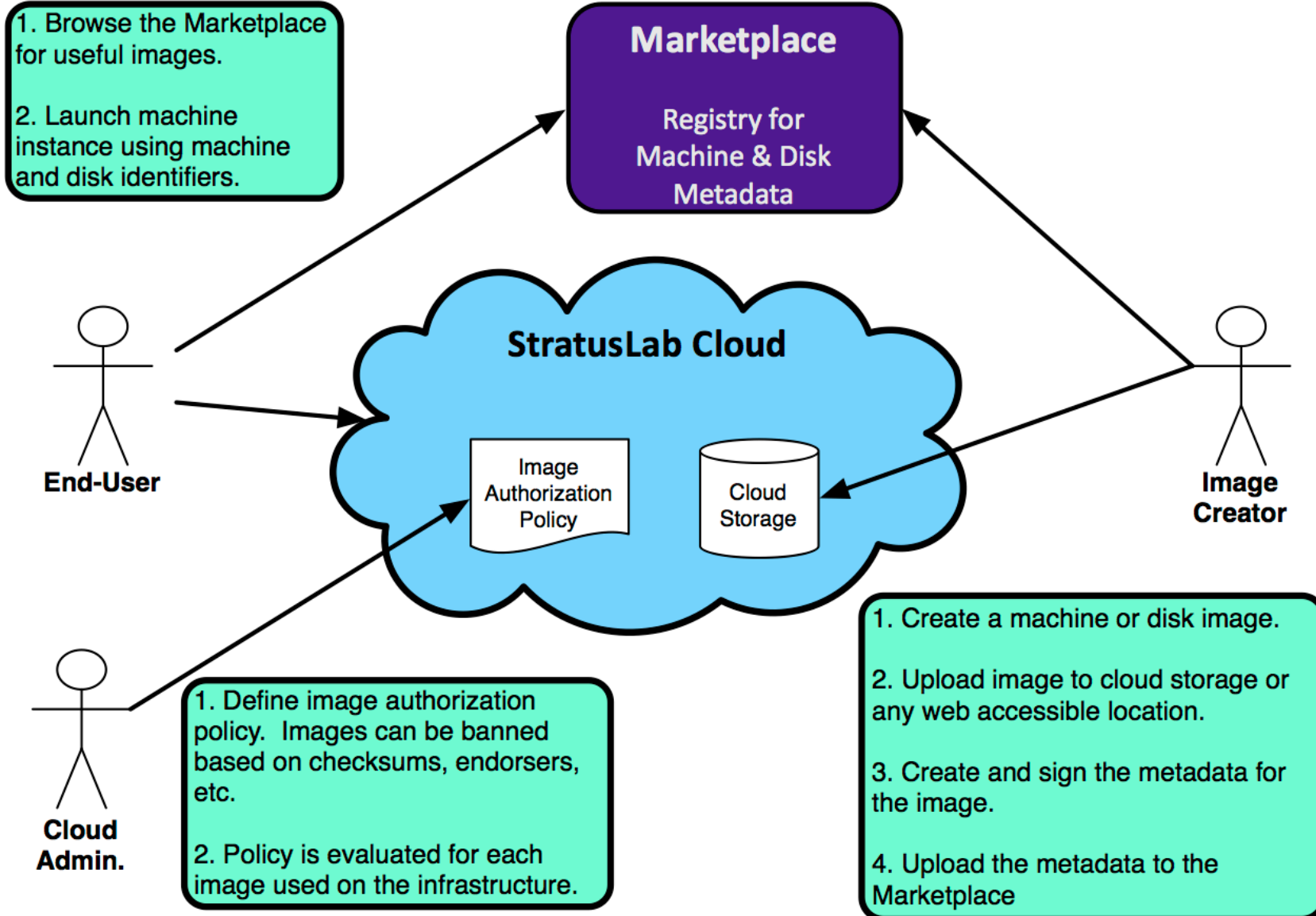
- Mechanism for sharing and trusting images
- Possible to distribute fixed, read-only data sets as well
- Split the storage of image metadata and image contents
- Availability of VM images of common operating systems



Implementation

- Marketplace API: Proprietary REST API for create, read, search
- Marketplace acts as image registry and handles only metadata
- Image contents can be located on any public (web) server
- 'Private' images can also be held in cloud storage
- CentOS, Ubuntu, OpenSuSE, Debian, Fedora, ScientificLinux images created and supported by StratusLab

Image Handling Workflow



Command Line Client

- Administrator: simplifies StratusLab installation
- Users: access StratusLab cloud from anywhere

Administration

- Quarantine for stopped virtual machines
- Monitoring of cloud activity and resources

Authentication and Authorization

- Supports username/password, certificates, cert. proxies
- Specification in local file and/or LDAP

Support



Information

- Web site documentation
- Recorded tutorials

Mailing List

- support@stratuslab.eu

Meetings

- Live tutorials (usually 2-3 per year)
- Workshops (2+ per year)

Priorities for Evolution



Interfaces

- Adopt CIMI as the standard interface to services
- Provide complete browser interface for all services



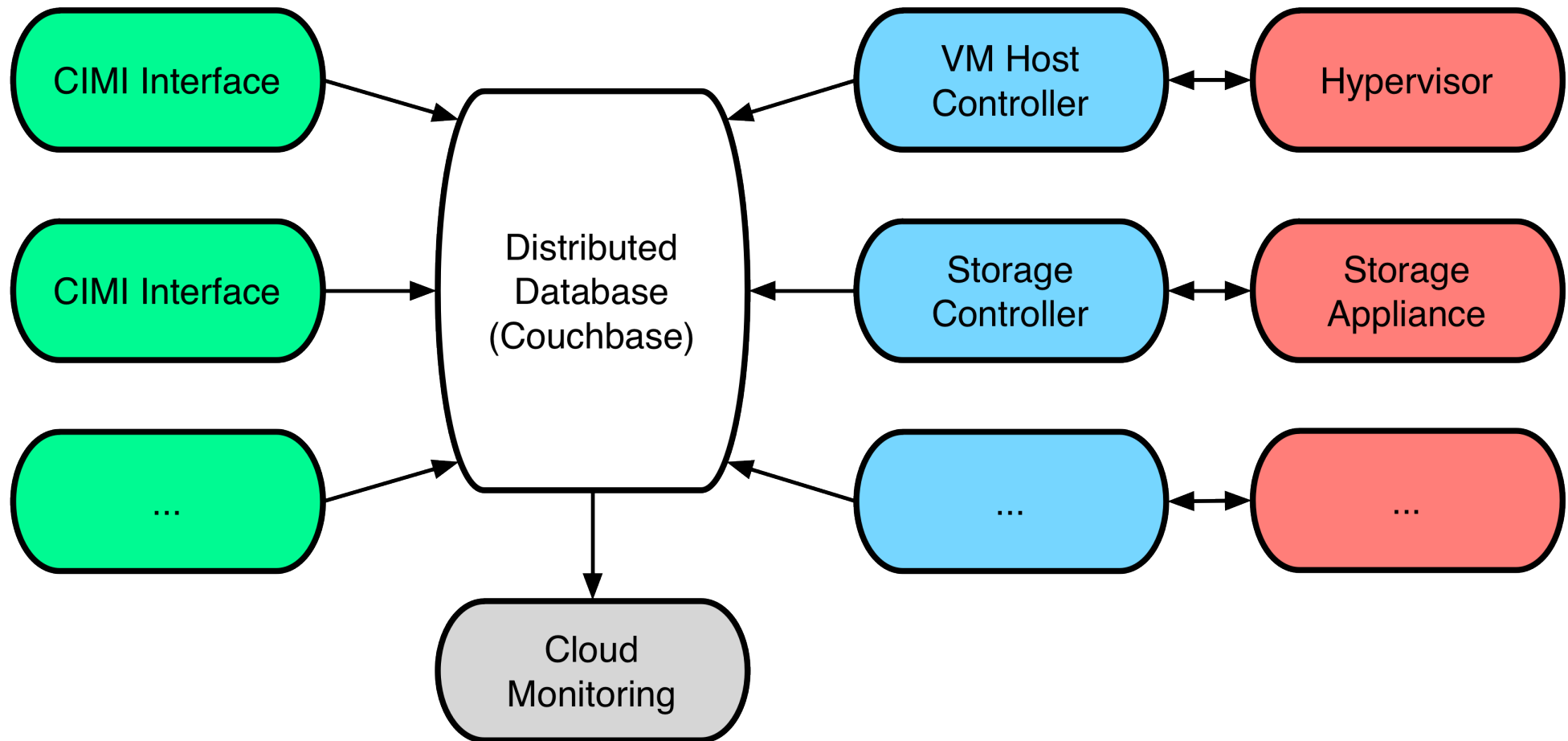
Simplicity, Scalability, & Robustness

- Direct use of libvirt as VM manager
- Distributed database (Couchbase) as information 'bus'

Better services for system administrators

- Improved overview and monitoring of infrastructure
- Fine-grained accounting for all resources
- Migration control

New Architecture



Running Clouds in Production

Reference Cloud Services

- (~)Open infrastructures for using StratusLab and providing feedback
- Operated on a first-come, first-serve, best-effort basis
- In production 2+ years, with 250+ registered users
- Two sites: LAL (Orsay, France) and GRNET (Athens, Greece)

Other deployments...

- Academic: France, Ireland, UK, Vietnam, South Africa, ...
- Commercial: Atos, Helix Nebula, ...

Building on top...

- SlipStream from SixSq: cloud based systems deployment and testing

Cloud Experience at LAL



Private cloud for laboratory services

- Works well, plan to migrate all services including grid worker nodes and experiment-specific servers
- Services switched to VMs without users being aware of change
- Very different way of working, need to change administrator habits
- Have seen some stability issues related to SL6 kernel/virtualization

Public cloud open to university

- Very positive reaction to cloud; LAL resources nearly 100% used
- Fields: biology, software eng., stats, astrophysics, bioinformatics, ...
- After initial introduction, users require only low level of support
- Other labs offering StratusLab training without our direct involvement

Majority of problems from machine room & hardware, not software.

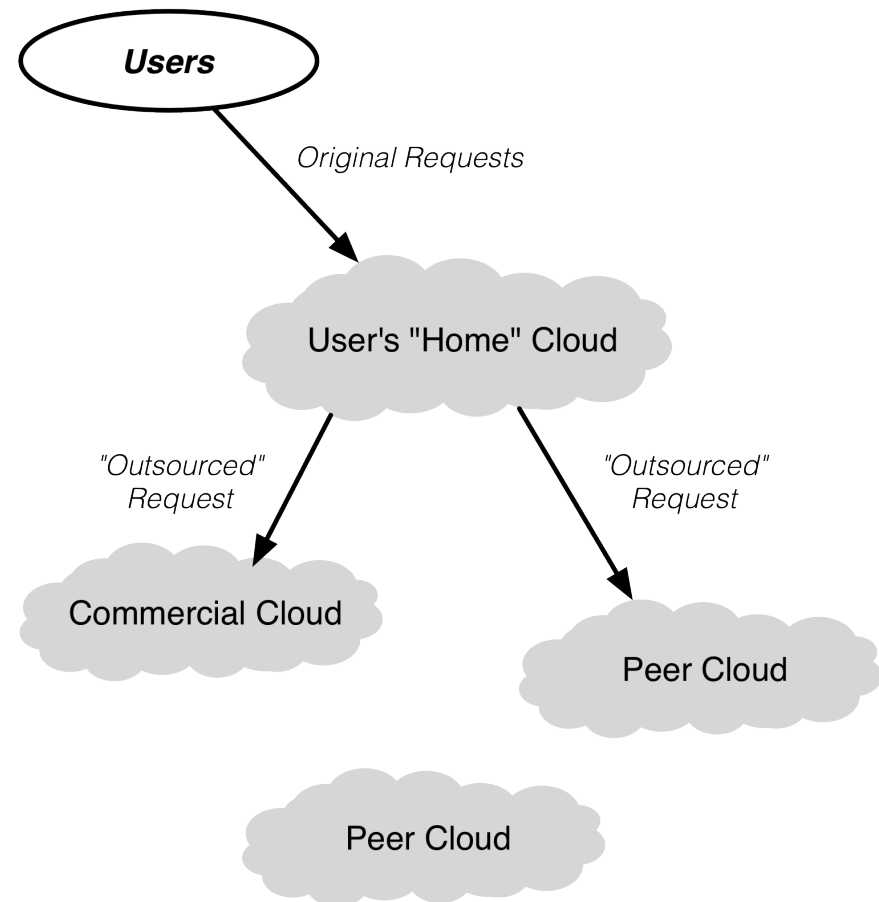
Federated Clouds

Federation Models (Hybrid Cloud & “Sky” Computing)

Transparent Federation

- Site operators “outsource” to other providers
- Completely transparent to end users
- Difficult to achieve in practice because of concerns about data protection, network access and performance

Peer Federation or Bursting



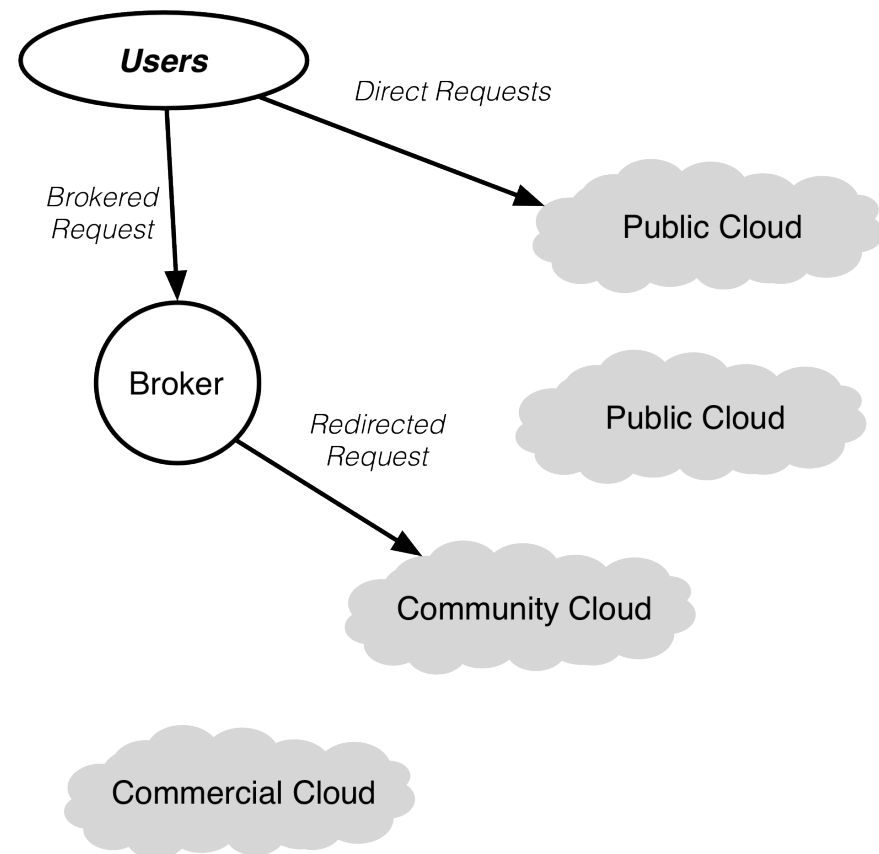
Federation Models (Hybrid Cloud & “Sky” Computing)

Brokered Federation

- Variety of different cloud infrastructures are visible to users
- Users choose to place virtual machines in particular locations
- Simple clients can handle federation if differences are small
- Orchestrators are needed for larger differences between clouds

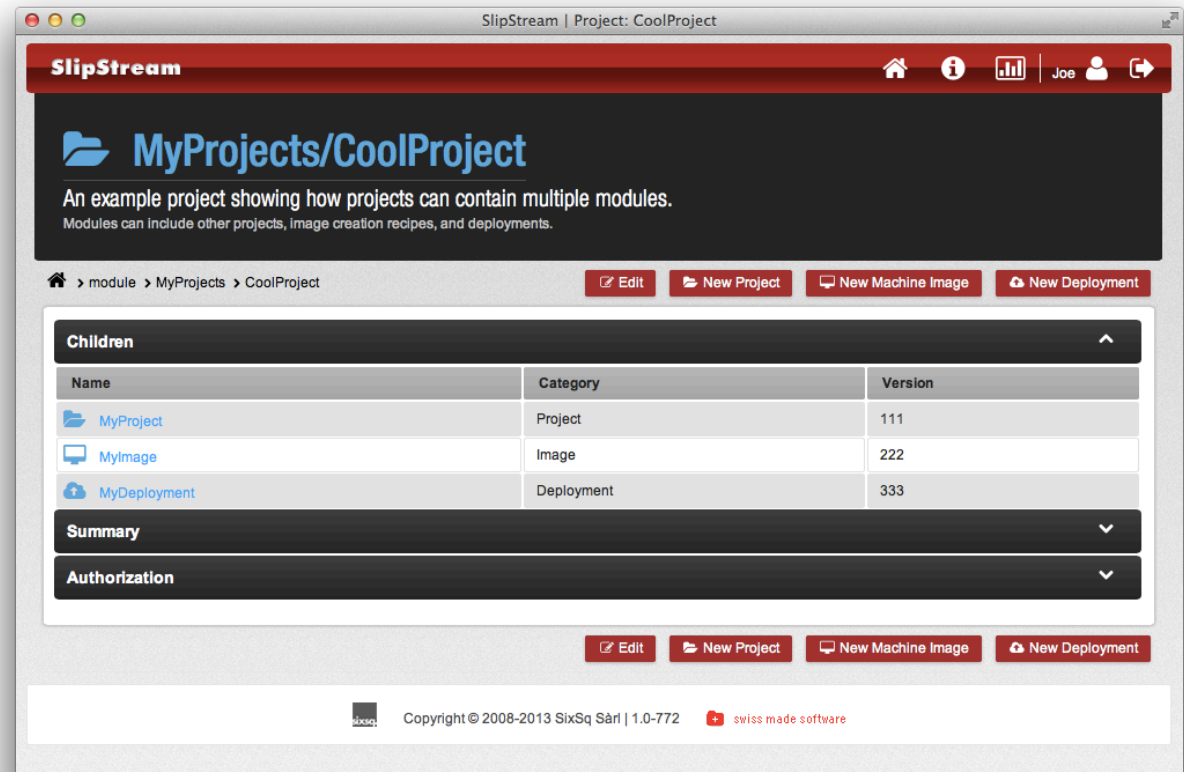
Both Helix Nebula and EGI take the brokered approach

Brokered Federation



Cloud orchestrator and deployment engine

- Facilitates testing, deployment, and maintenance of complex systems
- Transparent access to multiple cloud infrastructures
- Allows automated multi-cloud deployment of systems



Conclusions



StratusLab Cloud Distribution

- Supported, stable, and production-quality IaaS cloud distribution
- Used for reference cloud service for 2+ years
- Other academic and commercial deployments
- Defined, ambitious roadmap for its continued evolution
- Frequent administrator and user tutorials and workshops

StratusLab Collaboration

- New collaborators welcome: developers and documenters!
- Weekly phone conference between developers
- Biannual StratusLab workshops

Questions and Discussion

website <http://stratuslab.eu>

twitter [@StratusLab](#)

support support@stratuslab.eu

StratusLab source <http://github.com/StratusLab>

SlipStream source <http://github.com/slipstream>



<http://stratuslab.eu/>

Copyright © 2013, 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/>).

