6th Slide Set Cloud Computing

Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Faculty of Computer Science and Engineering
christianbaun@fb2.fra-uas.de

Agenda for Today

- Solutions for running private cloud infrastructure services
  - Focus: Eucalyptus and OpenStack
- Solutions for running private platform services
  - Focus: AppScale

Solutions for running Private Cloud Infrastructure Services

- Several free solutions exist run infrastructure services

abiCloud (Abiquo)	`http://www.abiquo.com`
CloudStack (Citrix)	`http://cloudstack.apache.org`
Enomaly ECP	`http://src.enomaly.com`
Eucalyptus	`http://open.eucalyptus.com`
Nimbus	`http://www.nimbusproject.org`
OpenECP	`http://openecp.sourceforge.net`
OpenNebula	`http://www.opennebula.org`
OpenStack	`http://www.openstack.org`
Tashii (Intel)	`http://www.pittsburgh.intel-research.net/projects/tashi/`

- These solutions are used mainly for the construction of private clouds
- Some solutions can also be used for the construction of public cloud services

Project Status of the Solutions

abiCloud (Abiquo)	???
Apache CloudStack	`https://github.com/apache/cloudstack`
Enomaly ECP	† (The company discontinued its former open source strategy)
Eucalyptus	(†) `https://github.com/eucalyptus/eucalyptus`
Nimbus	(†) `https://github.com/nimbusproject/nimbus`
OpenECP	† (Fork of Enomaly ECP. The development is stopped)
OpenNebula	`https://github.com/OpenNebula/one`
OpenStack	`https://github.com/openstack`
Tashi (Intel)	† (Intel is strongly committed with OpenStack)

Eucalyptus

- EUCALYPTUS – Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems
- Allows execution and control of virtual instances (Xen or KVM) on different physical resources
- Developed at UC Santa Barbara
  - Further development by Eucalyptus Systems, Inc.
- Interface compatible to AWS
  - EC2 + EBS + ELB + AutoScaling and S3
- Use of poplar AWS-compatible tools is possible:
  - e.g.: S3 Curl, Elasticfox, s3cmd,...
- Free software: GPLv3 (until 11/2017). Now BSD

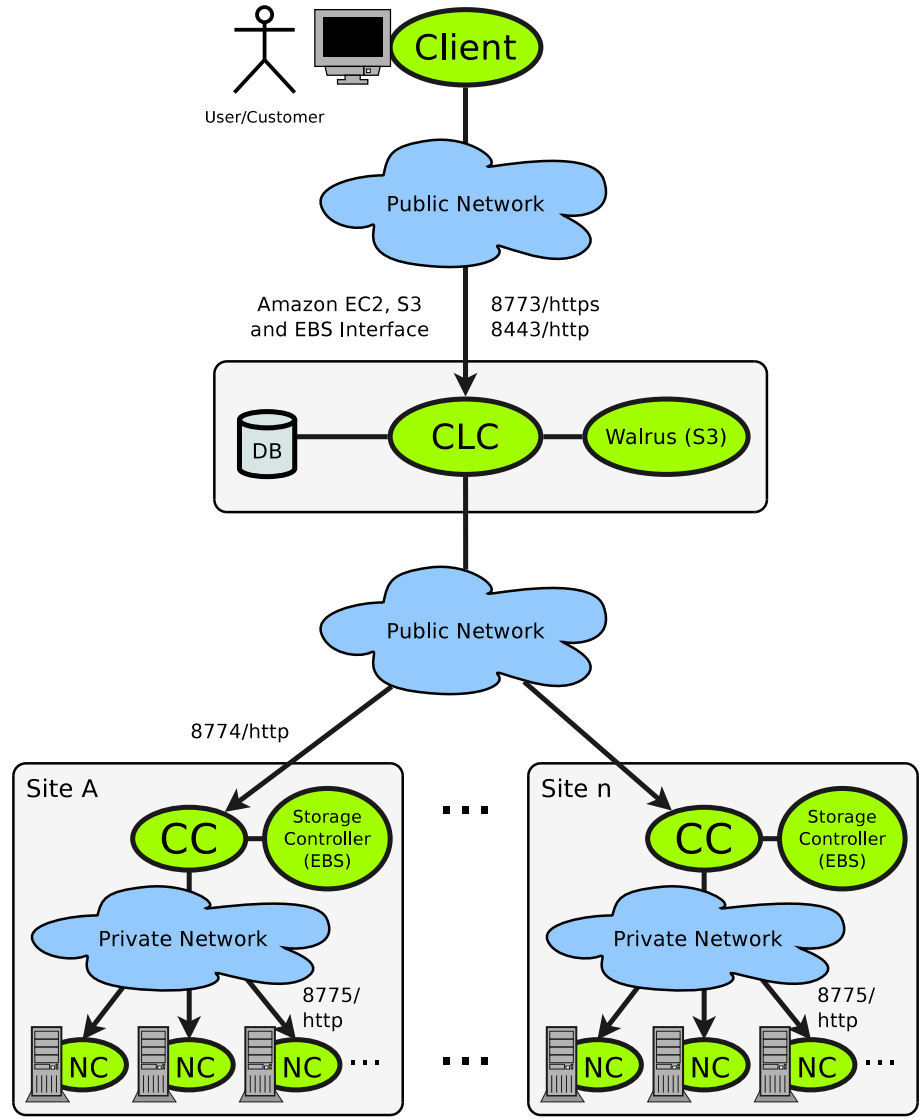
In 2008 + 2009, Eucalytus was a major step forward in establishing an API standard for cloud infrastructure services

Eucalyptus Services

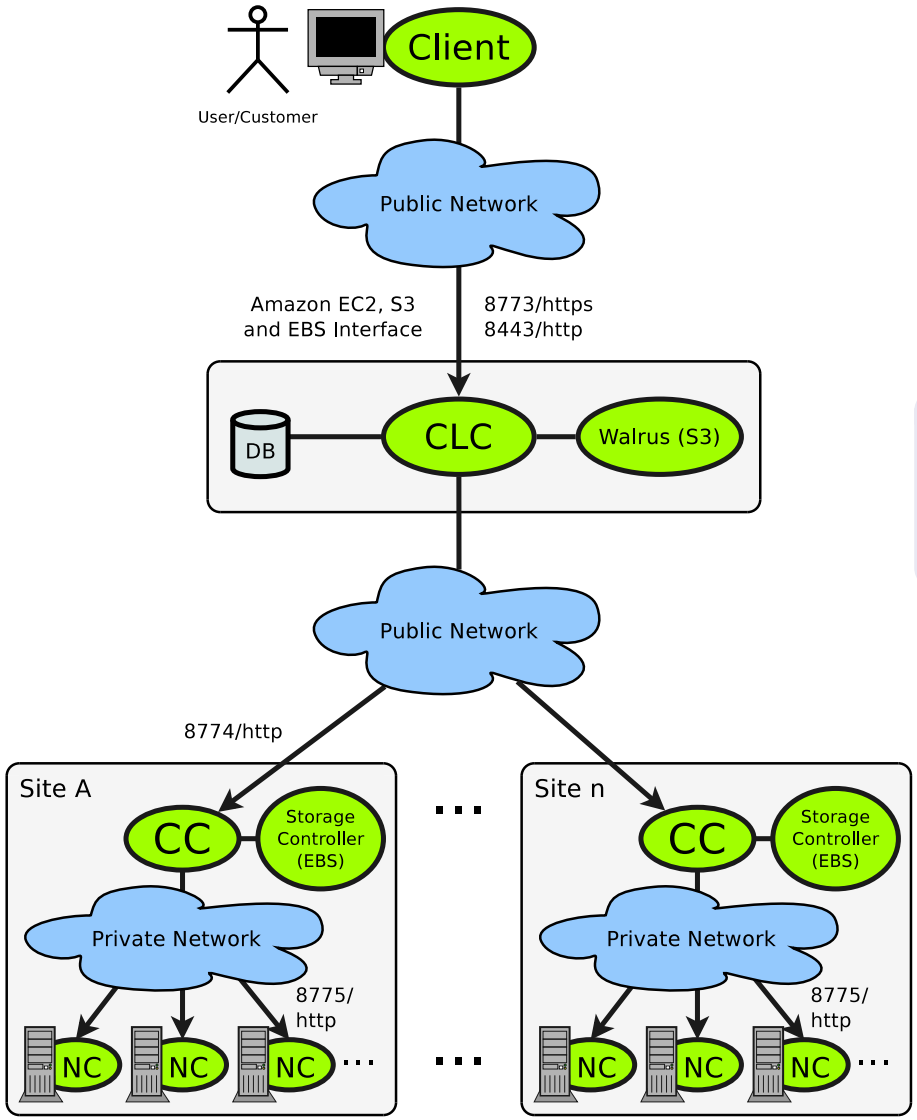
- Eucalyptus consists of several UNIX services
  - Cloud Controller (CLC)
  - Cluster Controller (CC)
  - Node Controller (NC)
  - Walrus
  - Storage Controller (SC)
- The services communicate via web services (SOAP+REST)
- Eucalyptus infrastructures consist of one or more sites

Redundant operation of the services CLC, CC, Storage Controller and Walrus became a feature with Eucalyptus v3.0 in 2011

This feature was removed with v4.2 in 2015



Node Controller (NC)

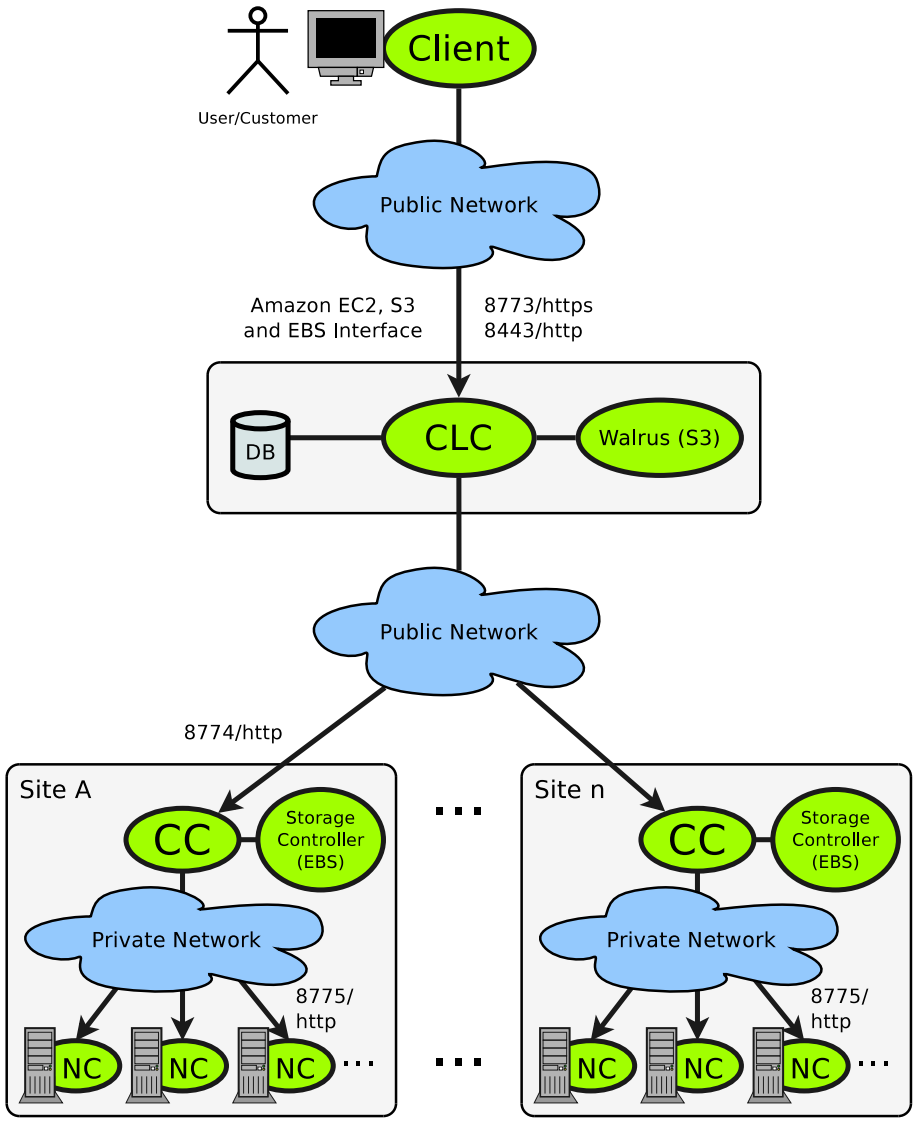


- Runs on every physical node, where instances are planned to run
- Controls the KVM hypervisor

- Xen is not supported any longer since v4.0
- VMware ESX(i) is not supported any longer since v4.1

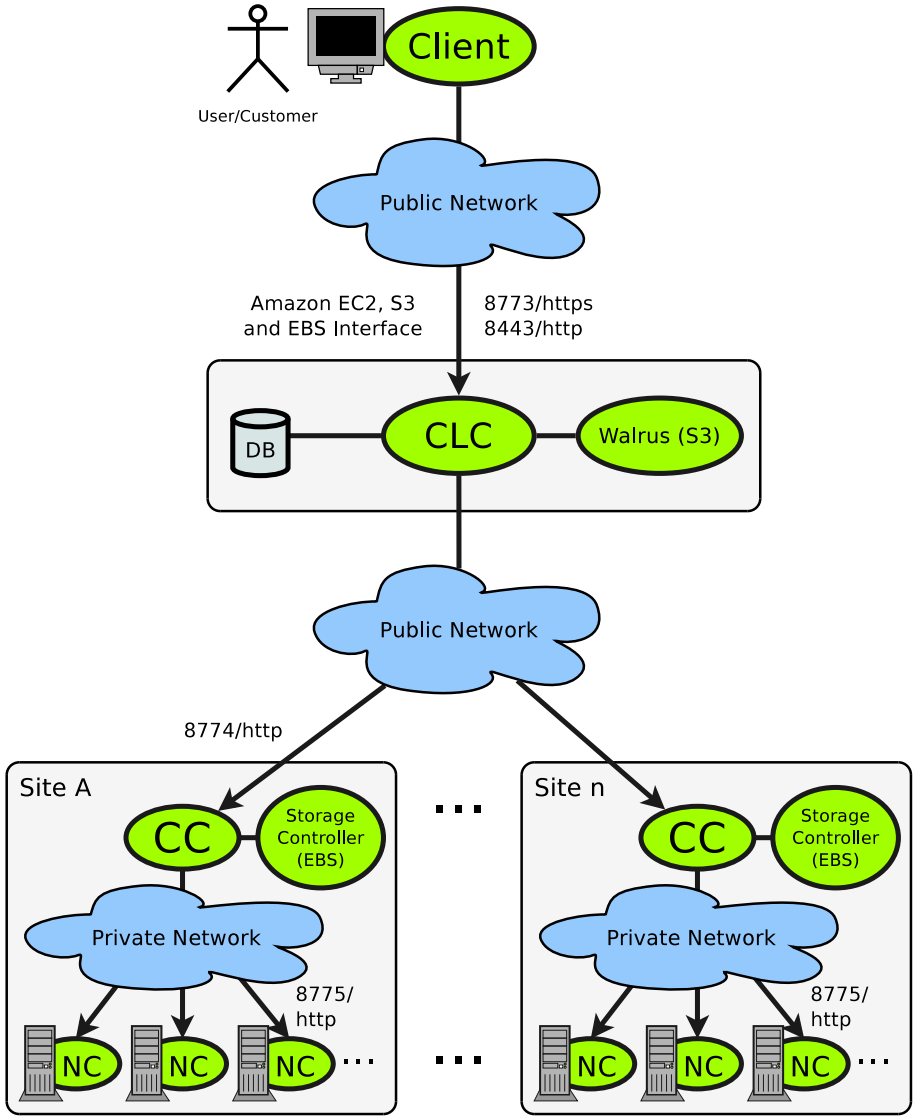
- Each NC transmits information about the utilization of their own resources to the CC of the site
  - Number of virtual processors
  - Free memory
  - Free storage

Cluster Controller (CC)



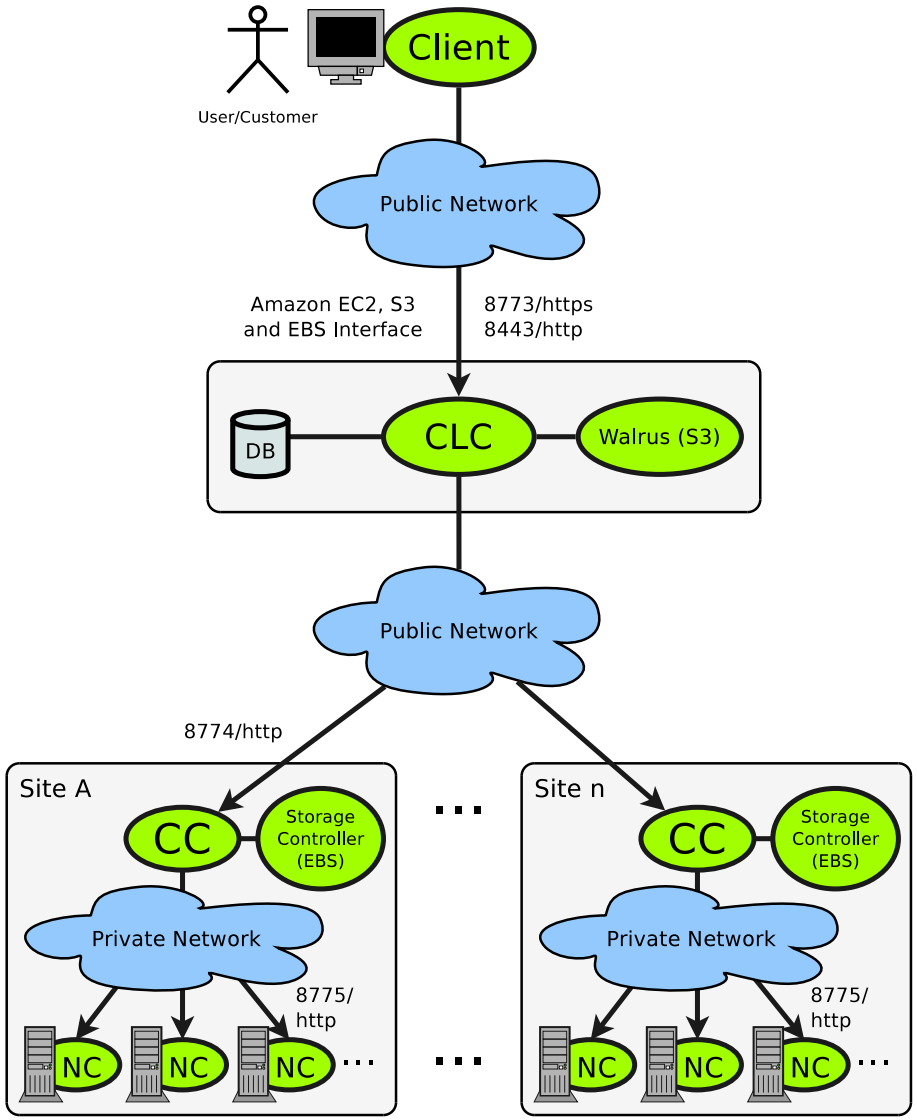
- Exactly a single CC per site is required
- Controls the distribution of the virtual machines to the NCs
- Collects free resource information from the NCs
- In small infrastructures CLC and CC usually run on the same physical server
- In each site, the NCs communicate with the CC via a virtual network (VLAN)
  - The VLAN ensures that all instances within a site share the same subnet

Cloud Controller (CLC)



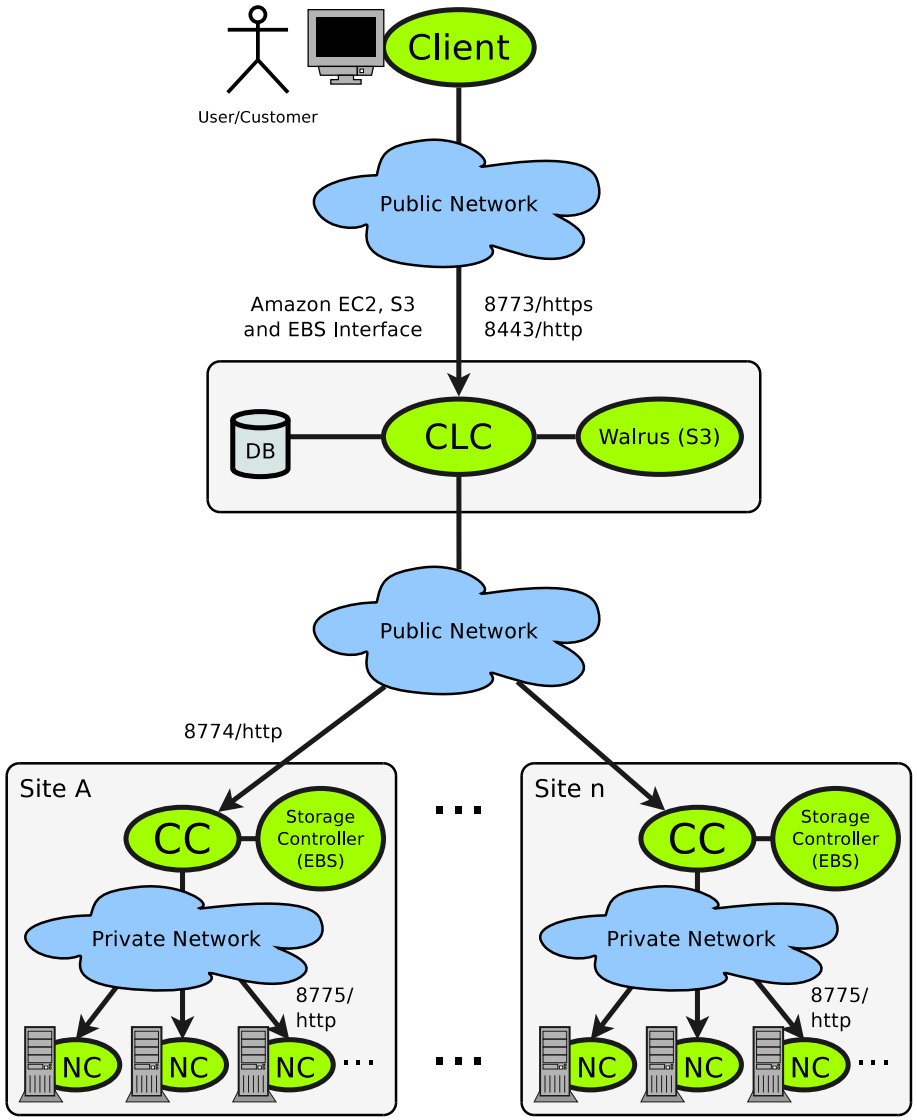
- Exactly a single CLC per Eucalyptus infrastructure is required
- Acts as a meta-scheduler in the cloud infrastructure
- Collects resource information from the CCs
- Runs per default on the same physical server as the storage services Walrus and Storage Controller

Walrus



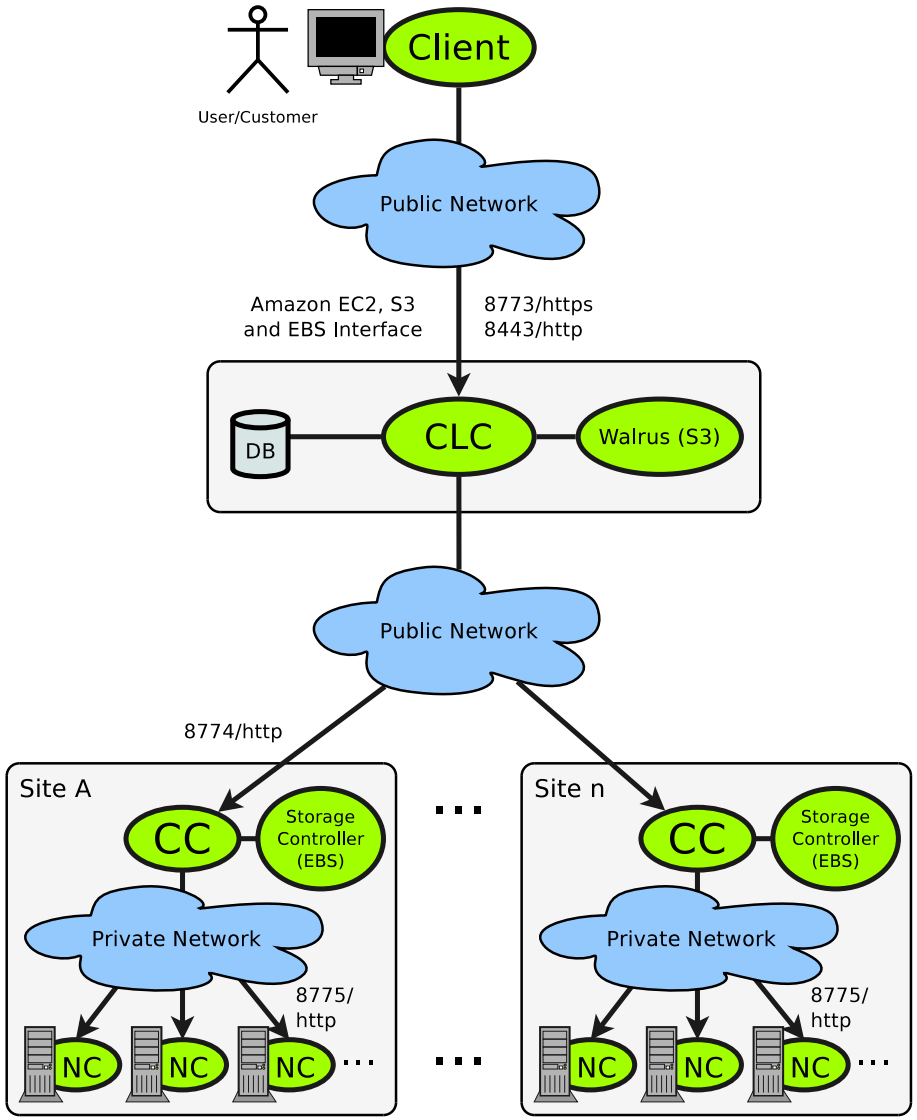
- Storage service, which implements the S3 REST API
- Eucalyptus stores here the images
- Usually runs on the physical server which hosts the CLC/CC
  - Can be outsourced from the CLC since v1.6
- Walrus is not a distributing service
  - Operates only in single-node mode
- In order to improve the read/write performance of the object-based storage, Walrus can be replaced by a Riak Cloud Storage (Cluster)

Storage Controller



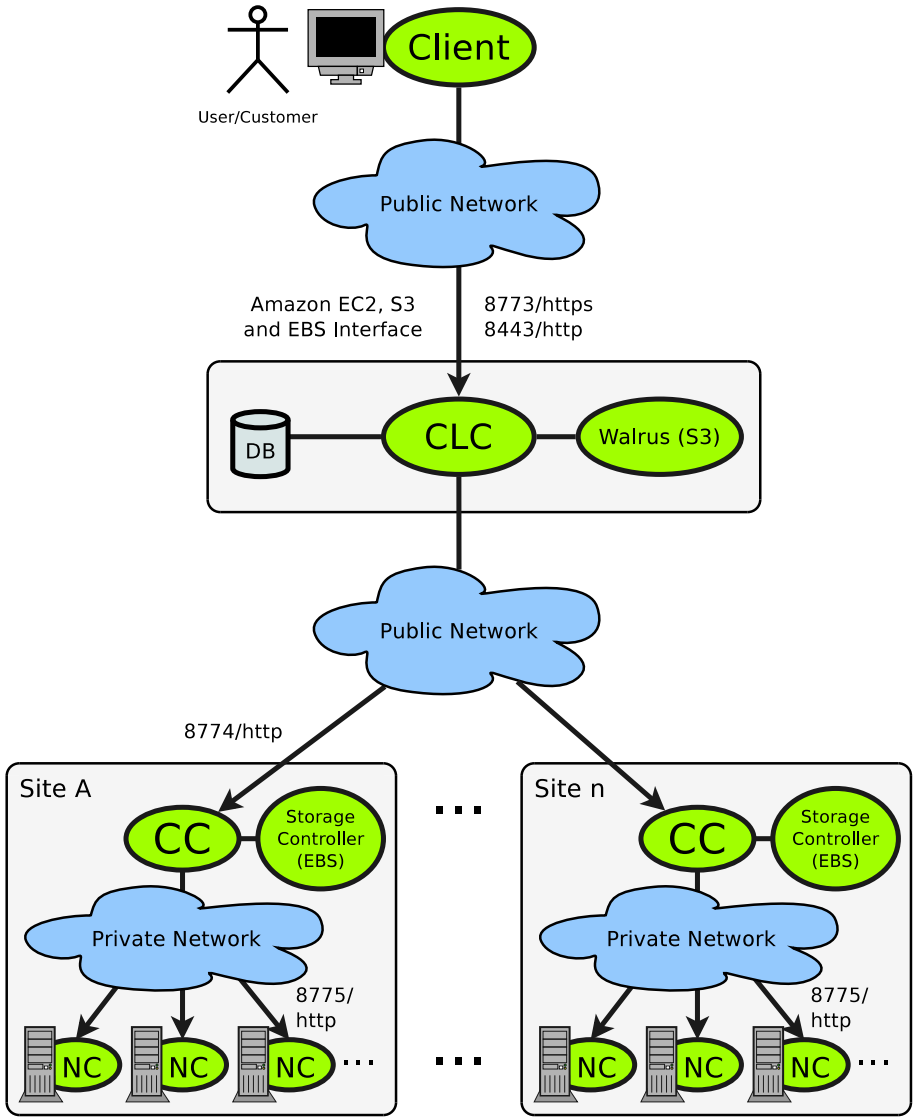
- Storage service, which implements the EBS API
- In infrastructures with only a single site, the Storage Controller usually runs on the physical server which hosts the CLC/CC
  - Can be outsourced from the CLC since v1.6
- If the infrastructure contains multiple sites, each site has its own storage controller

Launch of an Instance in Eucalyptus (1/2)



1. A user or applications, which tries to start instances, provides the CLC these parameters:
   - Image
   - Iinstance type
   - Number of instances
2. CLC selects a CC with enough free resources in its cluster
3. CC selects in the local cluster one (or more) NC(s), with enough free resources and commands the start of the instance(s)

Launch of an Instance in Eucalyptus (2/2)



4. If the required image is not available on the NC, the NC requests the image from the CLC
5. CLC transmits the image from Walrus via an encrypted transmission via Secure Copy (SCP) to the NC
6. The transmission duration for images from Walrus to the NCs depends of:
   - Network technology used
   - Number of required transmissions
   - Size of the images

Some Facts about Eucalyptus

- The Installation of Eucalyptus is simple when CentOS or Red Hat Enterprise Linux are used
  - and it is horror when other Linux distributions are used
- Stable operation of an Eucalyptus infrastructure is possible
  - If administrators are willing to invest some time...
  - Single services need to reboot from time to time
  - Help of the developers is not always helpful
  - Commercial support is (was?!) available
  - Commercial interests seem(ed) to be contrary to community concept sometimes
- Extensions and modifications in the source code are difficult
  - Source code of services appears obscure
  - No assistance from the developers can be expected
- Unclear future
  - HP aquired Eucalyptus systems in 9/2014
  - 2015: Eucalpytus became a part of HPE, which separated from HP
  - 2017: Eucalpytus became a part of DXC Technology $\Longrightarrow$ † (?!)

A new Hope for Eucalpytus?

- DXC stopped developing the product in late 2017
- AppScale Systems forked the code in 2018 and started supporting the product
- `https://www.eucalyptus.cloud`
- Latest version: 4.4.5 from December 2018
- `https://github.com/corymbia/eucalyptus/`

Eucalyptus Installation (the simple way on a single node)

- Check the installation tutorial
  `https://docs.eucalyptus.cloud/eucalyptus/4.4.5/index.html#shared/install_section.html`
- Create a virtual machine with CentOS 7.3 minimal
- Execute:
  `bash <(curl -Ls https://eucalyptus.cloud/install)`
- The script will ask a few questions (e.g. about spare IP addresses)
- Hope the best

- A faststart iso image „Cloud in a Box“ existed until 2015/2016
- The last revision came with Eucalyptus 3.4.2 and CentOS 6
- It was the most simple way to install an Eucalyptus IaaS on a single node or a cluster mode
- It is not available any more

Review of NASA regarding Eucalyptus

NASA drops Ubuntu's Koala food for (real) open source

Open core is not open source: a cautionary tale

By **Cade Metz in San Francisco** • **Get more from this author**

Posted in Software, 20th July 2010 05:09 GMT

Free whitepaper – Hosted apps

NASA is dropping Eucalyptus from its Nebula infrastructure cloud not only because its engineers believe the open source platform can't achieve the sort of scale they require, but also because it isn't entirely open source.

NASA chief technology officer Chris Kemp tells *The Reg* that as his engineers attempted to contribute additional Eucalyptus code to improve its ability to scale, they were unable to do so because some of the platform's code is open and some isn't. Their attempted contributions conflicted with code that was only available in a partially closed version of platform maintained by Eucalyptus Systems Inc., the commercial outfit run by the project's founders.

Instead, Kemp's team built their own compute engine and fabric controller from scratch. The new platform — dubbed Nova — has been open sourced under the Apache 2.0 license and is now part of the OpenStack project announced today by Rackspace.

Source: `http://www.theregister.co.uk/2010/07/20/why_nasa_is_dropping_eucalyptus_from_its_nebula_cloud/`

2011: Ubuntu switches from Eucalpytus to OpenStack

Ubuntu Cloud: OpenStack Wins, Eucalyptus Loses

Joe Panettieri | *Talkin' Cloud* *May 10, 2011*

Stated politely, Canonical is transitioning the Ubuntu Cloud to OpenStack -- an open source cloud standard -- as a foundation technology. Stated bluntly, Ubuntu is de-emphasizing Eucalyptus, another open source cloud standard. The obvious question: Is the open source cloud industry's balance of power shifting from Eucalyptus to OpenStack?

According to an official statement from Canonical:

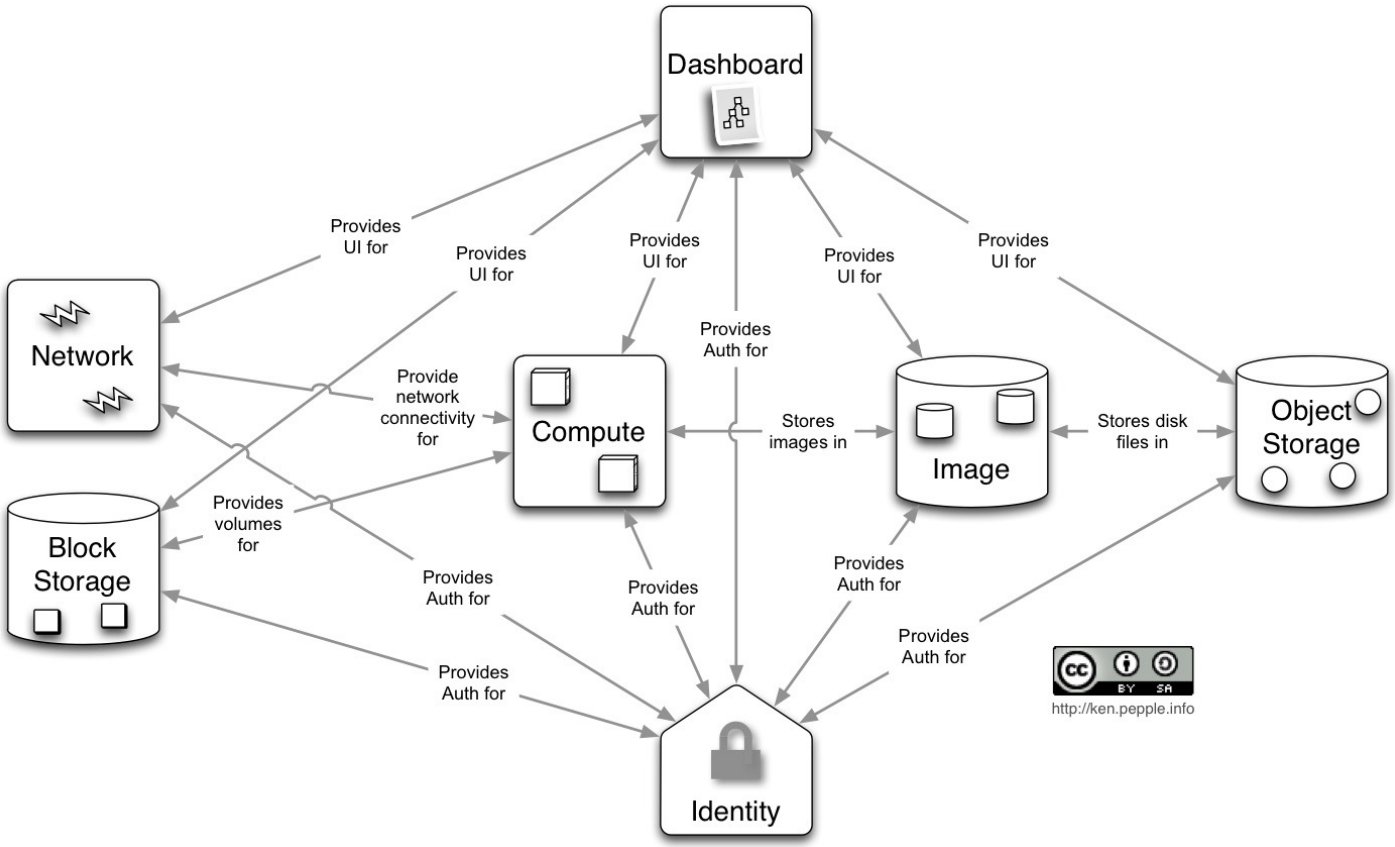
> "Today, the Ubuntu project announced that future versions of the Ubuntu Cloud will use OpenStack as a foundation technology. OpenStack, the rapidly growing, open-source, cloud platform effort founded by Rackspace and NASA in 2010, has secured more than 53 commercial companies including Dell, Internap, Intel and Cisco to join the IaaS cloud computing initiative since launching. Ubuntu officially joined the OpenStack initiative in February of this year, but it is not currently providing commercial services for it as part of the releases of its most recent Linux-based operating system, Ubuntu Server and Ubuntu Enterprise Cloud. However, with this announcement, OpenStack will become a core part of future releases. While no longer the foundation technology for the Ubuntu Cloud, Eucalyptus will remain within Ubuntu and will be available for users who prefer this technology."

Source: `http://talkincloud.com/ubuntu-cloud-openstack-wins-eucalyptus-loses`

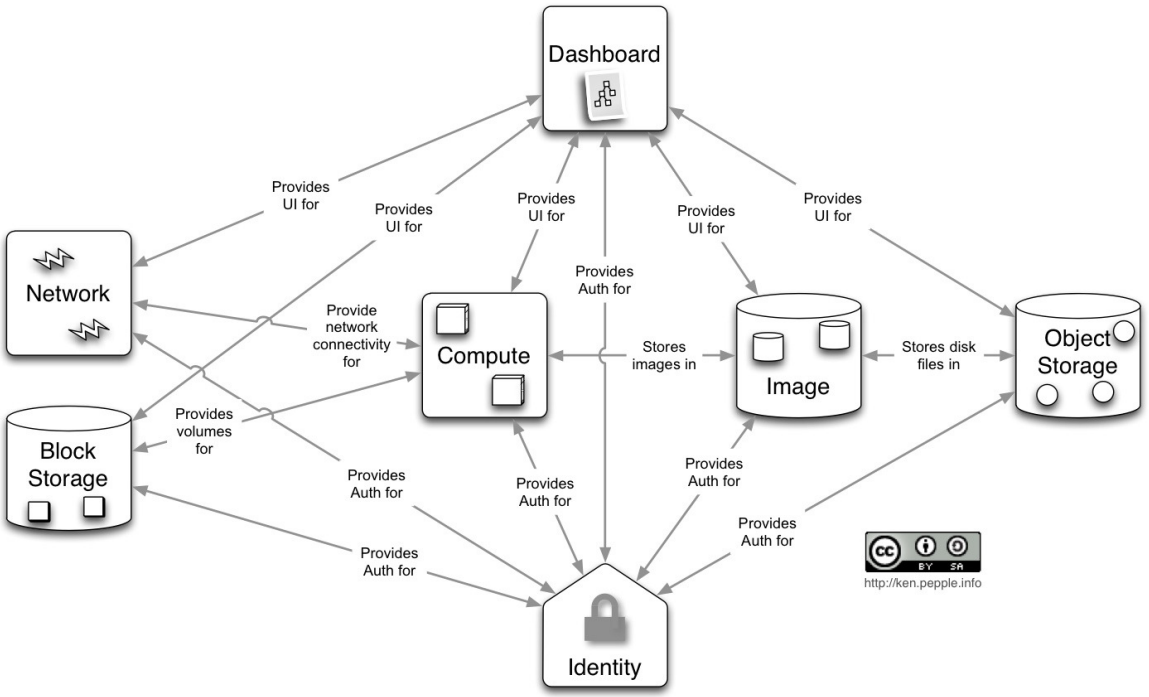
OpenStack

Image Source: OpenStack

- Initiated by NASA and Rackspace Cloud
  - Supported by AMD, Dell, IBM, Intel, Red Hat, SuSE, Yahoo and more
- Free software (Apache License v2.0)
- Contains several services which communicate via REST



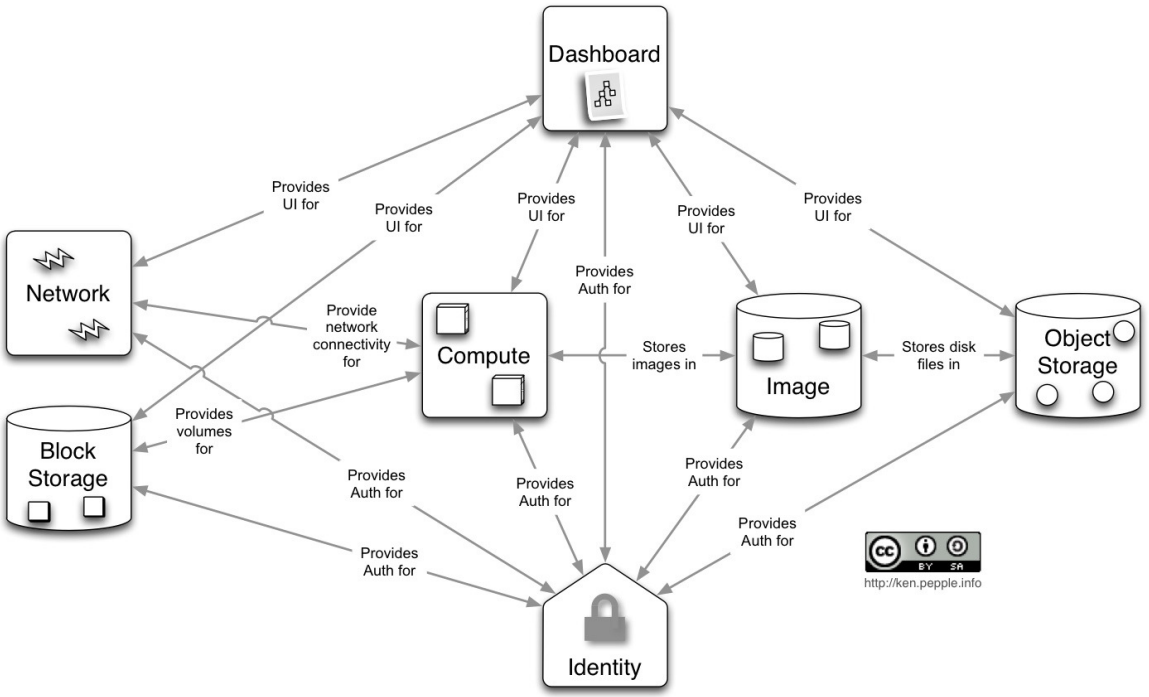
OpenStack – Services (1/4)



- **Compute** (*Nova*)
  - Infrastructure service
  - Implements the EC2 API
  - Highly scalable (up to tens of thousands of nodes)

- **Object Storage** (*Swift*)
  - Redundant, highly scalable (petabyte range), object-based storage service
  - Objects are stored on multiple hardware
  - Automatic replication when nodes fail or are added
  - Implements the S3 API

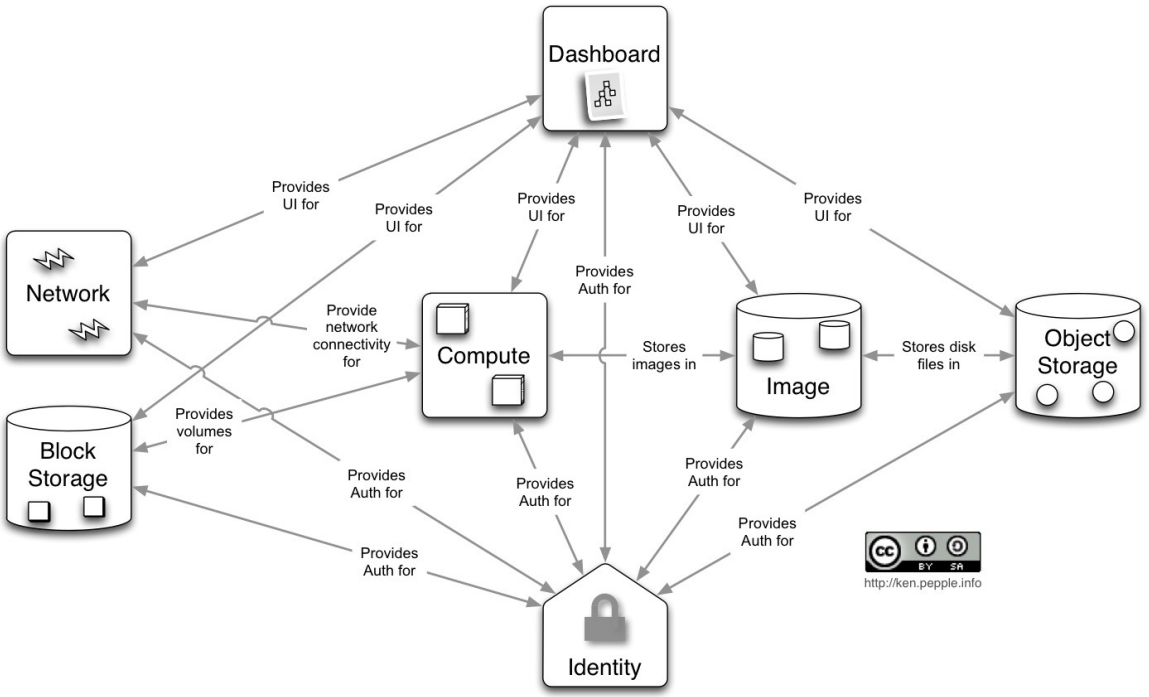
OpenStack – Services (2/4)



- **Image Service** (*Glance*)
  - Service for the search, register and request of images
  - Supported image formats: Raw, AMI, VHD (Hyper-V), VDI (VirtualBox), qcow2 (Qemu/KVM), VMDK and OVF (VMWare)

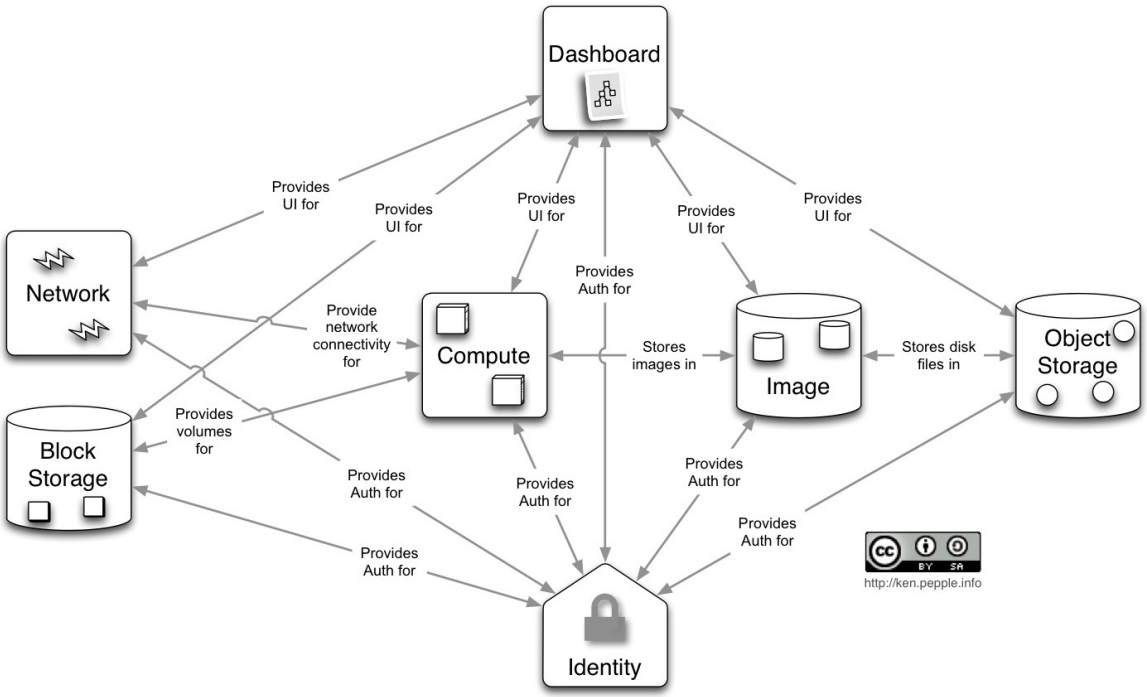
- **Block Storage** (*Cinder*)
  - Storage service for persistent block-based storage devices
  - Virtual storage devices can be created, erased, attached to and detached from instances
  - Implements the EBS API

OpenStack – Services (3/4)



- **Networking** (*Neutron*)
  - Service for managing IP addresses and distributing them to instances
  - Administrators specify, if all instances are connected to the same network, or if they are separated from each other via VLAN
- **Dashboard** (*Horizon*)
  - Provides a graphical web-interface for administrators and users

OpenStack – Services (4/4)



- **Identity Service** (*Keystone*)
  - Central directory of users for the other OpenStack services
  - Provides user authentication
  - Can interact (query) existing user directory services (e.g. LDAP)

Infrastructure Services and their Compatibility to the AWS

	AWS APIs implemented		
Project/Solution	EC2	S3	EBS (EC2)
abiCloud (Abiquo)	partly	—	—
CloudStack (Citrix)	partly	—	—
Enomaly ECP	—	—	—
Eucalyptus	partly	partly (Walrus)	partly (SC)
Nimbus	partly	partly (Cumulus)	—
OpenECP	—	—	—
OpenNebula	partly	—	partly
OpenStack	partly (Nova)	partly (Swift)	partly (Cinder)
Tashi (Intel)	—	—	—

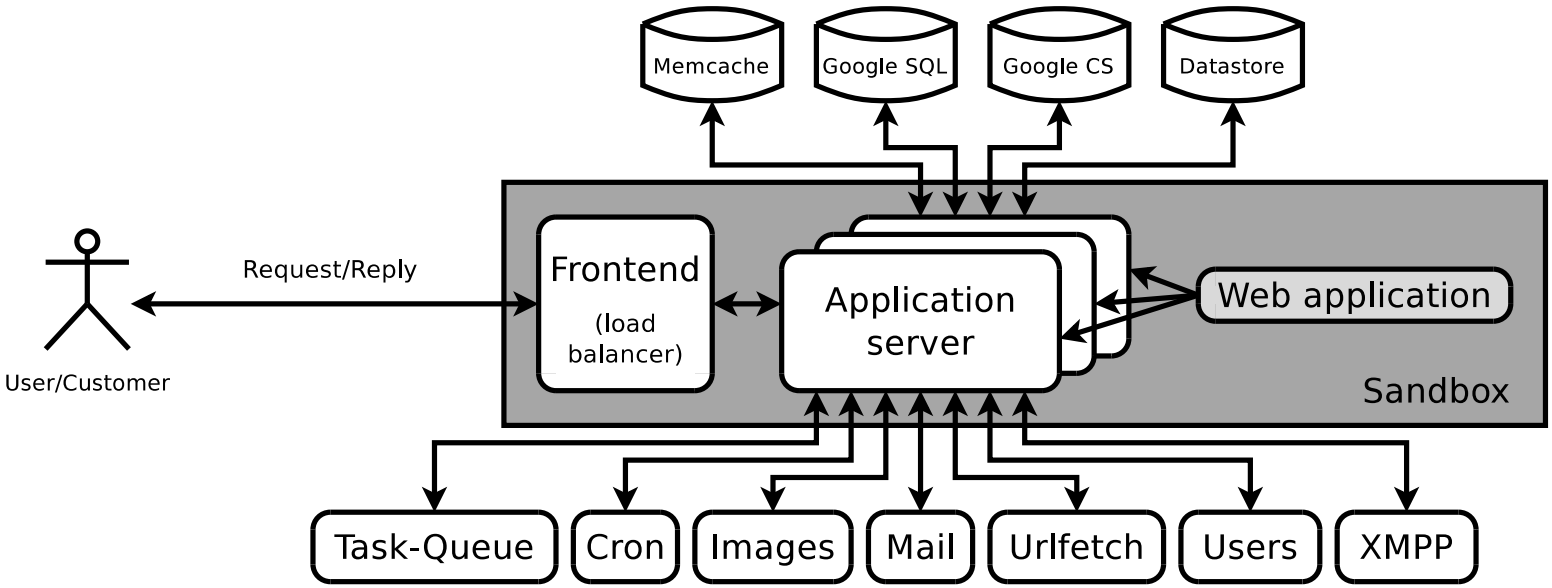
- Many free private cloud solutions exist, which implement an object-based storage service and provide the S3 API
  - Ceph-RGW, Fake S3, Minio, Riak CS, S3ninja, S3rver, Scality S3 Server,...

Freie Objektspeichersoftware mit S3-API, *Christian Baun*. iX 9/2017, P.76-79
`https://www.heise.de/ix/heft/Eimerweise-3807215.html`

AppScale

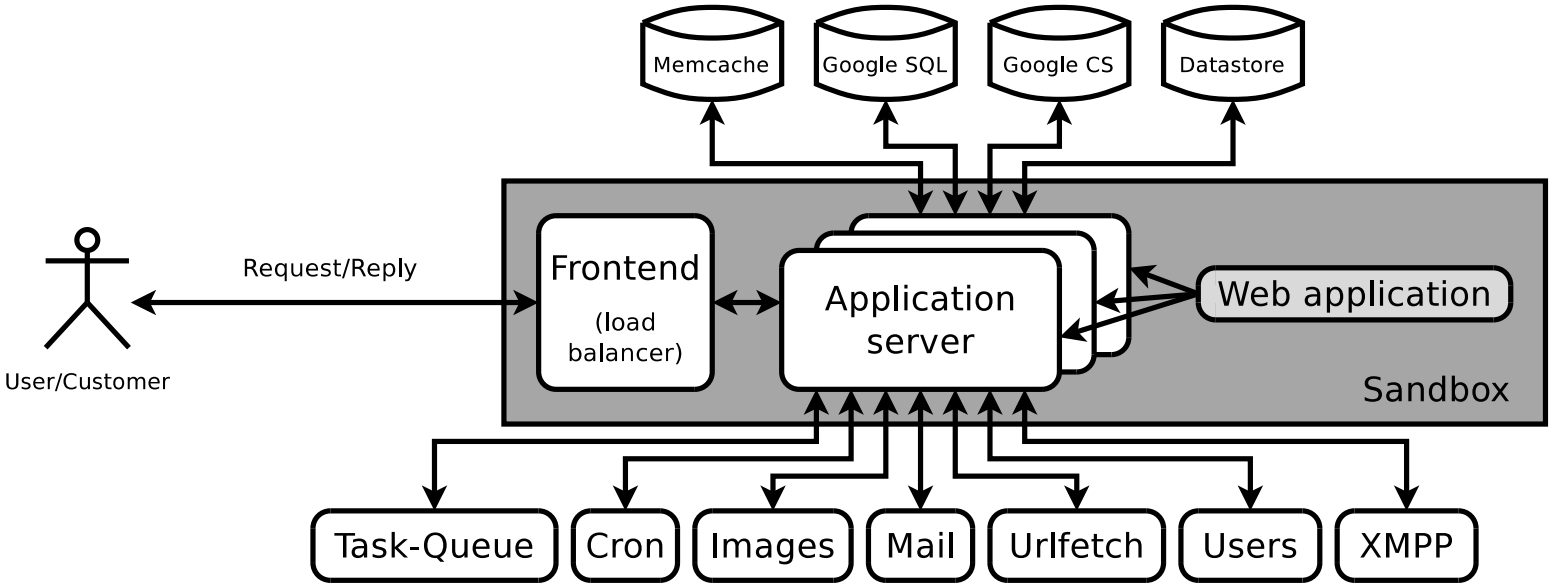
- Free reimplementation of the Google App Engine
  - Allows the operation of a GAE-compatible platform service
- Developed at UC Santa Barbara since 2008
- Free software (Apache 2.0 license)
- GAE-compatible applications can be developed, executed and tested with AppScale inside a...
  - public cloud with EC2, Microsoft Azure, Google Compute Engine or Alibaba Cloud
  - private cloud with Eucalyptus, OpenStack and CloudStack
- Can also run directly on Docker, Xen, KVM and VirtualBox
- Actively developed
- `https://github.com/AppScale/appscale`

AppScale Implementation (1/3)



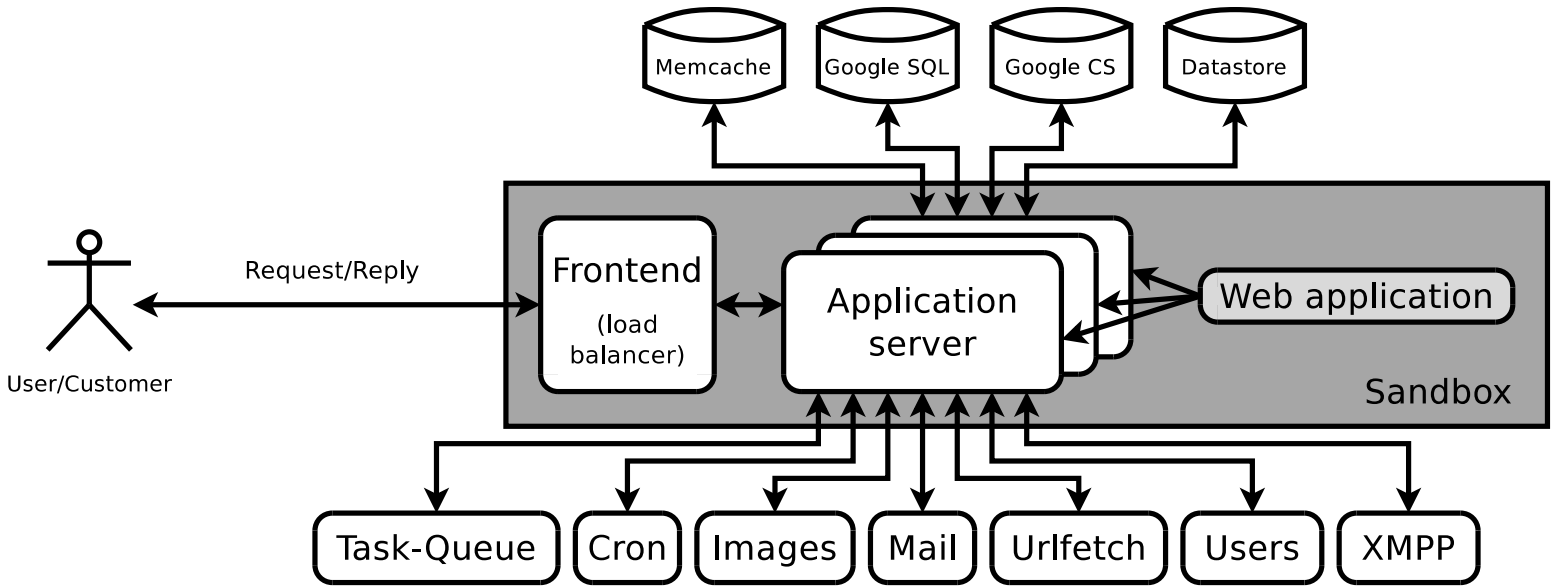
- AppScale emulates Google's infrastructure services by using free software
- The Application server with its APIs is implemented by using the App Engine SDK (which is free software)
  - The SDK contains a resource-saving development server, that implements the core functionality and the APIs of the PaaS
  - But this development server does not scale per default!

AppScale Implementation (2/3)



- The web server functionality is implemented with **nginx**
- The load balancing functionality is implemented by **HAProxy**
- A Datastore implements the distributed NoSQL db **Apache Cassandra**
- The coordination service **ZooKeeper** synchronizes all services
  - Apache ZooKeeper is a distributed configuration service, synchronization service, and naming registry for large distributed systems

AppScale Implementation (3/3)



- The messaging functionality provides the free XMPP server **ejabberd**
- The memcache storage is emulated with the cache server **Memcached**
- The execution of regularly scheduled tasks that operate at defined times or regular intervals is done by **cron**
- The image manipulation functios provides the **Python Imaging Library**
- Message queues implements AppScale with **RabbitMQ**

AppScale Installation

- As first step, the AppScale Tools need to be deployed
  - This can be done via `pip`, which is the package manager for python packages

```
$ sudo pip install appscale-tools
```

- Now, the command line tool `appscale` is available
- Next, the AppScale PaaS service need to be deployed
  - This can be done either inside...
    - a VM inside a public cloud service offering
    - a local VM or container
    - directly inside the local host operating system

`https://www.appscale.com/try-appscale`

- $\geq$ 4 GB of main memory are required because of the Cassandra DB

Deployment of AppScale inside a local VirtualBox VM

```
$ mkdir -p ~/appscale
$ cd ~/appscale
$ vagrant init appscale/releases
$ curl -o Vagrantfile https://s3.amazonaws.com/appscale_CDN/files/
Vagrantfile_template
$ vagrant up
```

- More information:

  `https://github.com/AppScale/appscale/wiki/AppScale-on-VirtualBox`

- The installation with Docker is even more simple:
  `docker run -i -t appscale/appscale:latest /bin/bash`

AppScale als Alternative zu Googles App Engine, *Christian Baun*. iX 12/2016, P.72-75

Working with AppScale

- Create an AppScalefile

  ```
  $ appscale init cluster
  ```

- Modify your AppScalefile (this is for a single machine configuration)

  ```
  ips_layout:
    master : <IP-Node-1>
    appengine: <IP-Node-1>
    database: <IP-Node-1>
    zookeeper: <IP-Node-1>
  ```

- Start AppScale

  ```
  $ appscale up
  ```

- Deploy a web application

  ```
  $ appscale deploy guestbook.tar.gz
  ```

  The application is online now via port 8080 (HTTP) and 4380 (HTTPs)

- Shut down AppScale

  ```
  $ appscale down
  ```

- Erase the web application and all data

  ```
  $ appscale clean
  ```

Create an AppScale Cluster

- Use 4 or 8 nodes
- When using 4 nodes, assign each node one of these roles: `master`, `appengine`, `database` and `zookeeper`
  - Drawback: No redundant storage or services
- When using 8 nodes, only the master is the Single Point of Failure

```
ips_layout:
 master : <IP-Node-1>
 appengine:
 - <IP-Node-2>
 - <IP-Node-3>
 database:
 - <IP-Node-4>
 - <IP-Node-5>
 zookeeper:
 - <IP-Node-6>
 - <IP-Node-7>
 - <IP-Node-8>
```

- More information:

`https://github.com/AppScale/appscale/wiki/Supported-Deployments`

Summary

- The discussed solutions for running private cloud infrastructure and platform services all have advantages and drawbacks
- Drawbacks:
  - Stability and functionality are usually worse in contrast to public cloud service offerings
- Benefits:
  - Independence from the behavior of public cloud services providers
    - Examples: business model changes (usage fee increases), functionality of the services change, service quality changes
    - Precondition: APIs must be compatible with public cloud services offerings
  - Free software $\Longrightarrow$ high level of flexibility
  - Potential way to build hybrid clouds
- Conclusion:
  - Even if the stability and functionality are lower, private cloud services can protect against unwanted changes