Eucalyptus: An Open-source Infrastructure for Cloud Computing

Shashi Mysore
Eucalyptus Systems Inc.
www.eucalyptus.com

Overview

Eucalyptus

Open-source cloud computing infrastructure

- Implements private and hybrid clouds
- Uses local machines and data without modification
- Transforms your resources into a cloud computing system, such as Amazon EC2

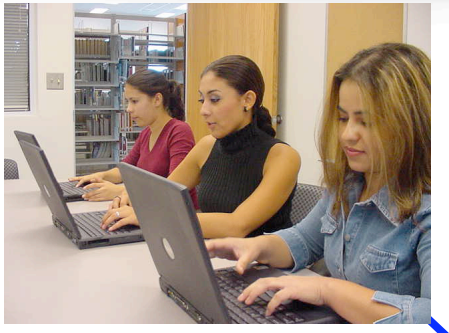
Company Facts

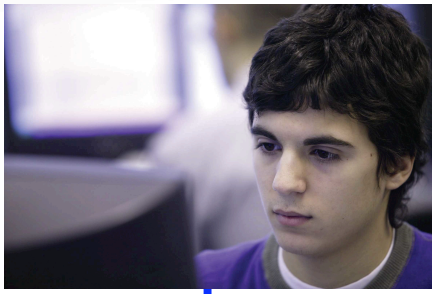
Eucalyptus

- Creators of the leading open-source private cloud platform
- Incorporated January 2009
- Headquartered in Santa Barbara, CA
- Funded by NEA, Benchmark Capital, and BV Capital
- Born from the EUCALYPTUS cloud computing research project from U.C. Santa Barbara (released May 2008)

What is a cloud?

Eucalyptus





SLAs

Web Services

Virtualization

Open-source Cloud Infrastructure

Eucalyptus

- ***Idea:*** **Develop an open-source, freely available cloud platform for commodity hardware and software environments**
  - Stimulate interest and build community knowledge
  - Quickly identify useful innovations
  - Act to dampen the “hype”
- **First-principles cloud implementation**
  - Not a refactorization of previously developed technology

What's in a name?

Eucalyptus

- **Elastic Utility Computing Architecture Linking Your Programs To Useful Systems**
- **Web services based implementation of elastic/utility/cloud computing infrastructure**
  - Linux image hosting ala Amazon
- ***How do we know if it is a cloud?***
  - Try and emulate an existing cloud: Amazon AWS
- **Functions as a software overlay**
  - Existing installation should not be violated (too much)
- **Focus on installation and maintenance**
  - *"System Administrators are people too."*

Goals for Eucalyptus

Eucalyptus

- **Foster greater understanding and uptake of cloud computing**
  - Provide a vehicle for extending what is known about the utility model of computing
- **Experimentation vehicle prior to buying commercial services**
  - Provide development, debugging, and “tech preview” platform for Public Clouds
- **Homogenize local IT environment with Public Clouds**
  - AWS functionality locally makes moving using Amazon AWS easier, cheaper, and more sustainable
- **Provide a basic software development platform for the open source community**
  - E.g. the “Linux Experience”
- **Not designed as a replacement technology for AWS or any other Public Cloud service**

Requirements

Eucalyptus

- **Implement cloud abstractions and semantics**
  - Must be a cloud (inarguably)
- **Simple**
  - Must be transparent and easy to understand
- **Scalable**
  - Interesting effects are observed at scale (e.g. not an SDK)
- **Extensible**
  - Must promote experimentation
- **Non-invasive**
  - Must not violate local control policies
- **System Portable**
  - Must not mandate a system software stack change
- **Configurable**
  - Must be able to run in the maximal number of settings
- **Easy**
  - To distribute, install, secure, and maintain

Open-source Cloud Anatomy

Eucalyptus

- **Extensibility**
  - Simple architecture and open internal APIs
- **Client-side interface**
  - Amazon's AWS interface and functionality (familiar and testable)
- **Networking**
  - Virtual private network per cloud
  - Must function as an overlay => cannot supplant local networking
- **Security**
  - Must be compatible with local security policies
- **Packaging, installation, maintenance**
  - system administration staff is an important constituency for uptake

The Elements of Cloud Style

Eucalyptus

- **SaaS (Software as a Service)**
  - Applications exporting network-facing user interfaces
  - User transfers data to the cloud
- **PaaS (Platform as a Service)**
  - Program or scripting runtime exports network-facing interfaces
  - Internal platform services available
  - User transfers program code and data to the cloud
- **IaaS (Infrastructure as a Service)**
  - Resource provisioning services export network-facing interfaces
  - Internal platform services available
  - User transfers code, data, and environment to the cloud

Why IaaS?

Eucalyptus

- **Applications are often multi-technology**
  - System “images” for different technologies can be combined
  - Multiple language technologies at different revision levels
- **Legacy support**
  - System images that mimic bare metal deployments can be used
  - Legacies are archived with the environment necessary to run them
- **Transparency**
  - Debugging and performance tuning can go down to the hypervisor
- **QoS containers**
  - QoS is implemented in the infrastructure today => familiar
- **Anti-lock in**
  - If clouds fail, a return path to bare metal is available

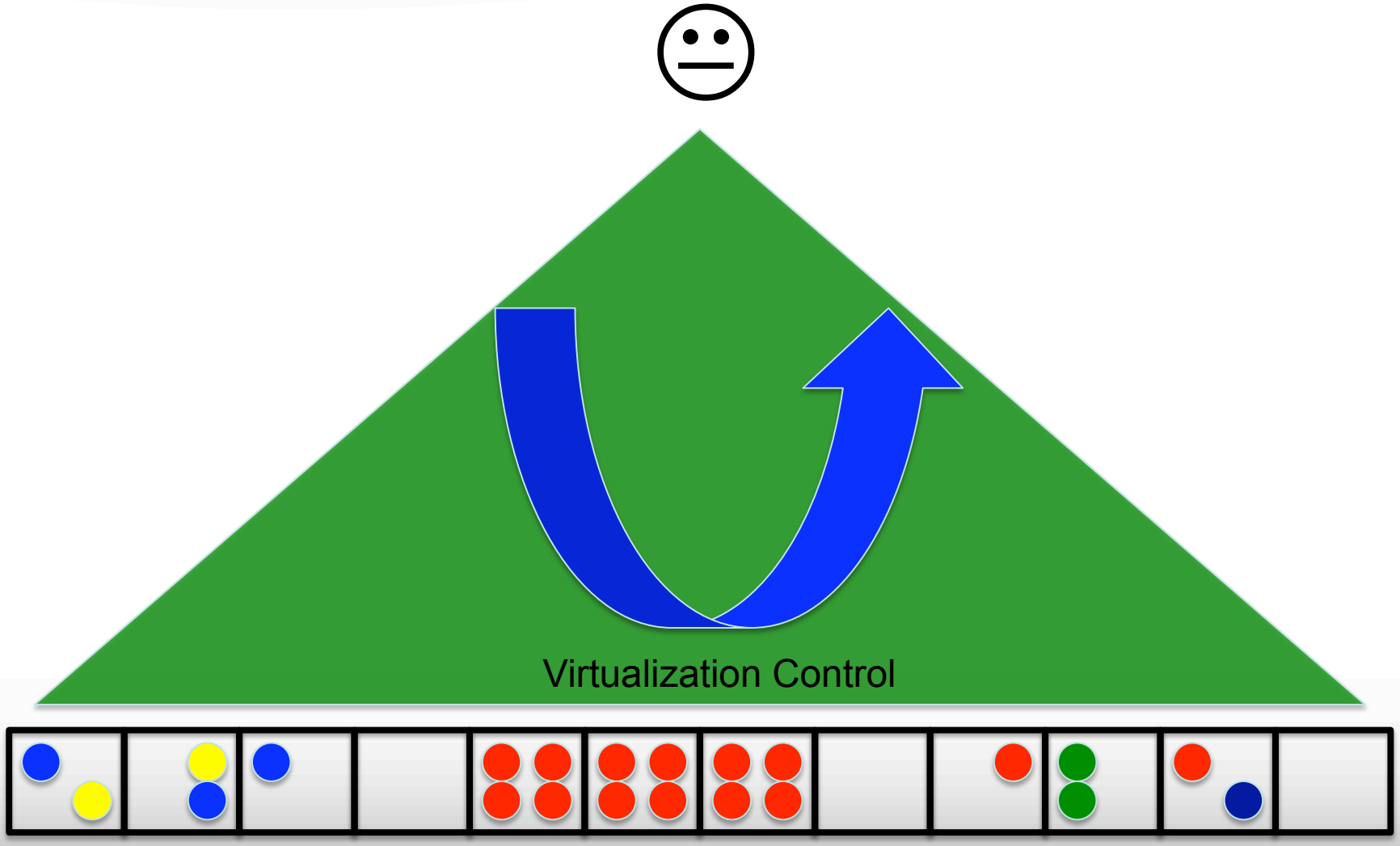
Clouds and Virtualization

Eucalyptus

- ***Isn't a virtualized data center a private cloud?***
  - "I'm running Xen in my data center – I'm running a private cloud."
- **Web service interfaces to hypervisors are useful**
  - Clouds require transactional operations across multiple hypervisors
- **Data center virtualization platforms are synchronous**
  - Data center administrators typically requires sequential consistency

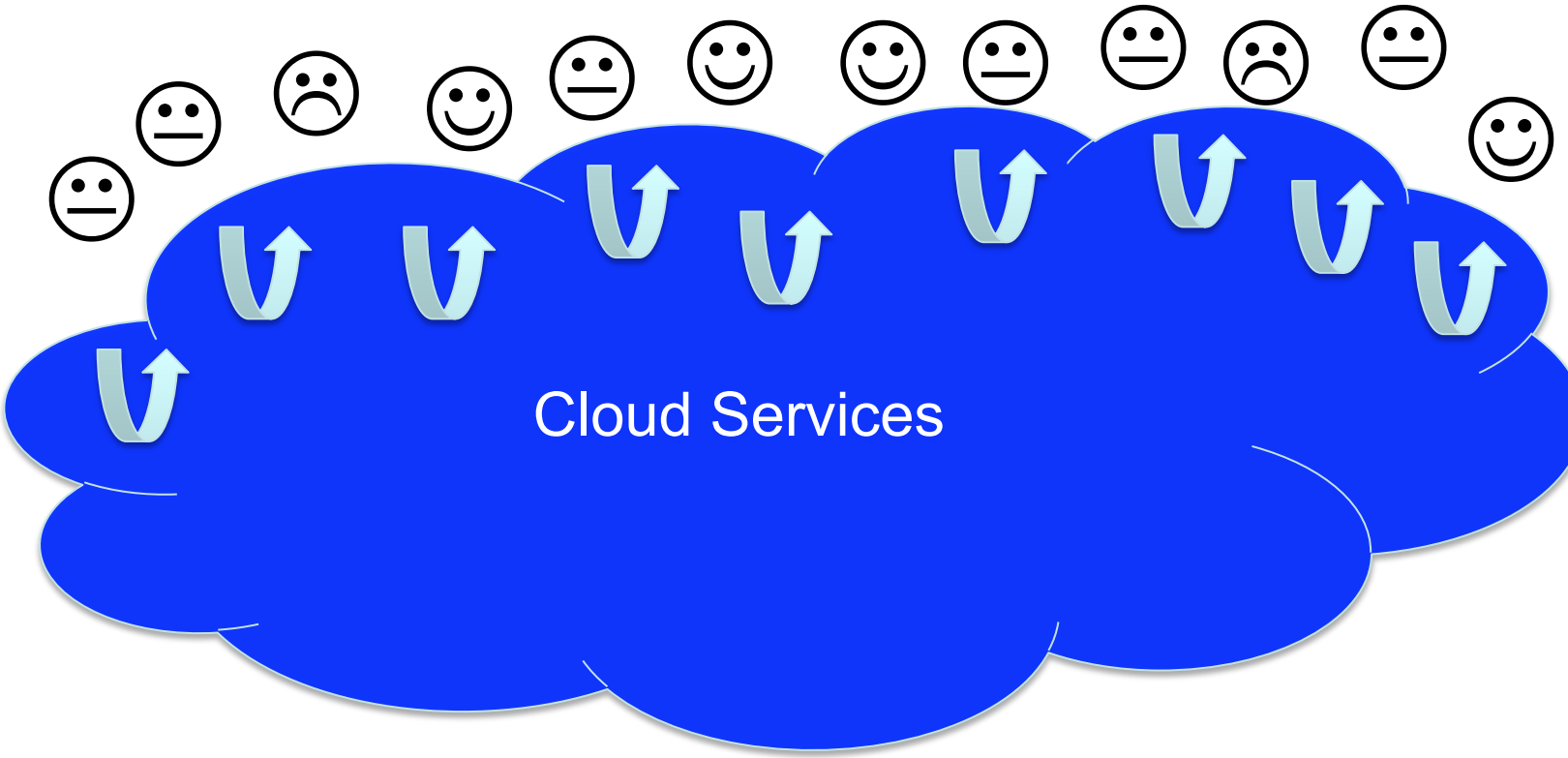
Datacenter Virtualization

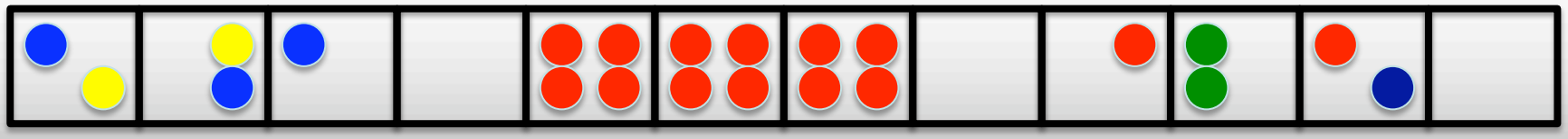
Eucalyptus



A Cloud

Eucalyptus





Cloud QoS Factors

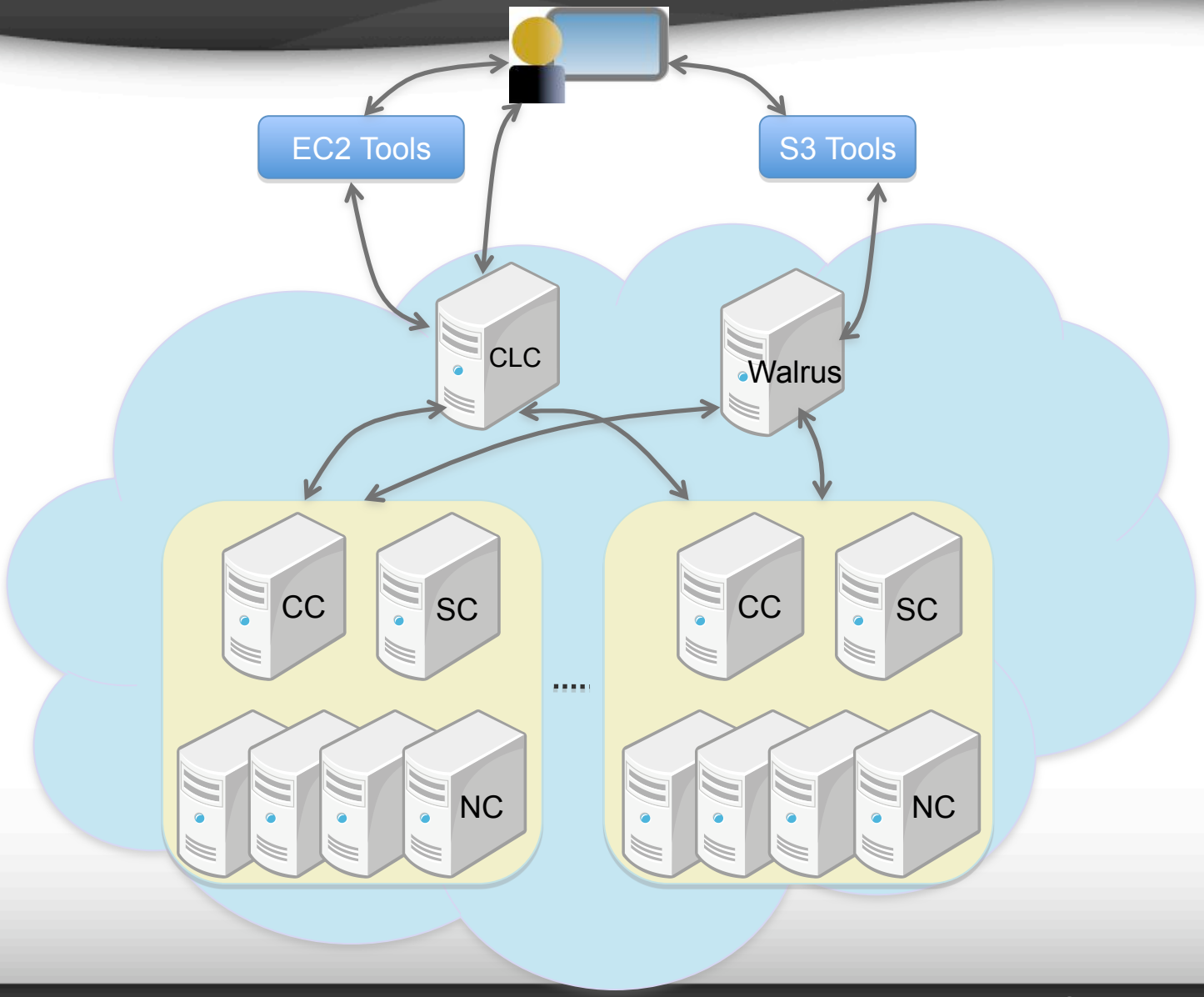
Eucalyptus

- **Scalability**
  - VM scalability
  - User scalability
  - Storage scalability
  - Node scalability
  - Multi-tenancy
- **User Experience**
  - VM provisioning
  - Request-response times
- Application Performance
  - Application I/O characteristics
  - Application architecture
- Cloud specific
  - Image management
  - Elastic block storage
  - Security groups
  - Walrus
  - Network configurations

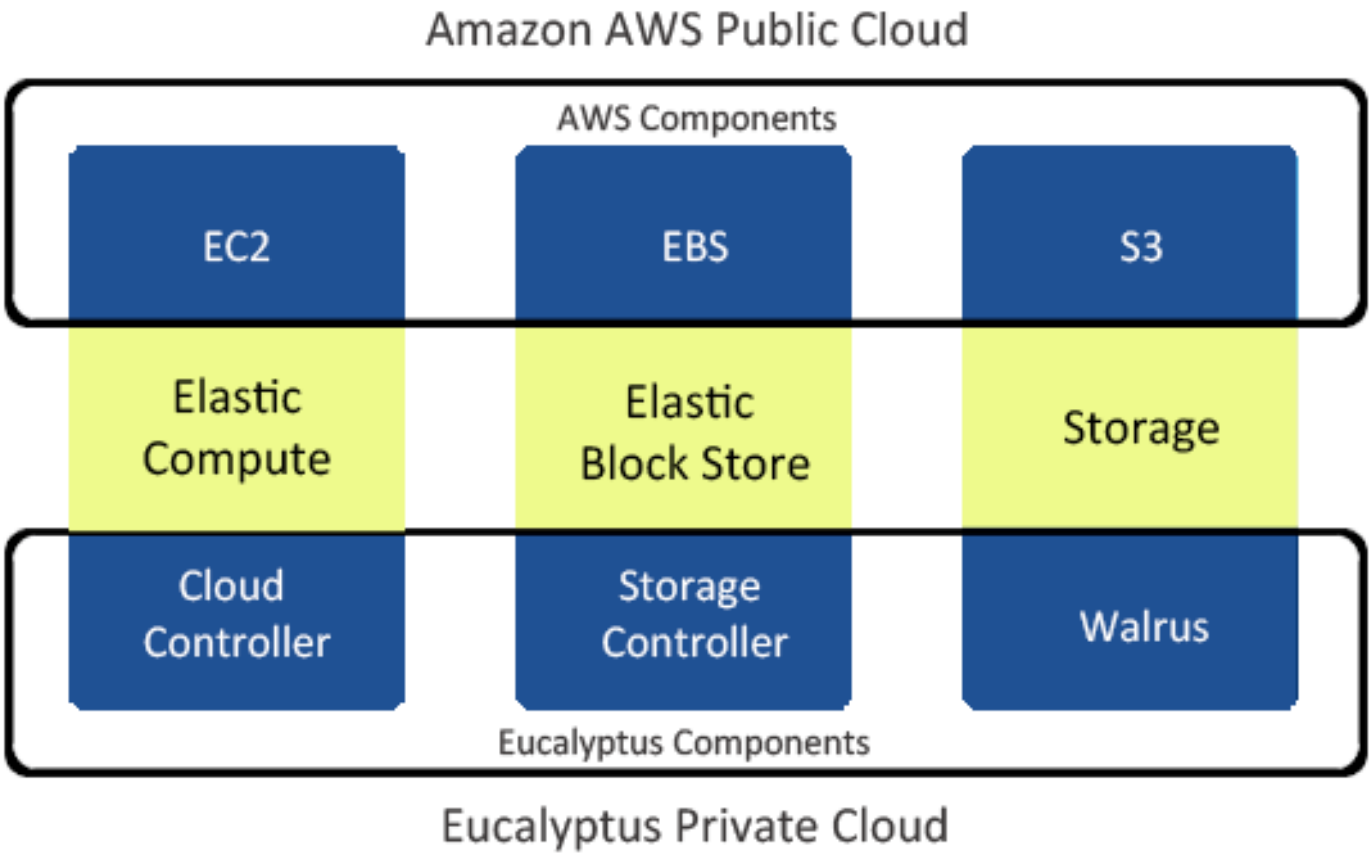
Eucalyptus Architecture

- **Distributed system**
  - Components implemented as web services
- **Components run on top of existing resources**
  - Linux distribution agnostic
    - Ubuntu, RHEL, CentOS, SLES, openSUSE, Debian
  - Hypervisor agnostic
    - Xen, KVM, VMware

Architecture



Amazon AWS Compatibility

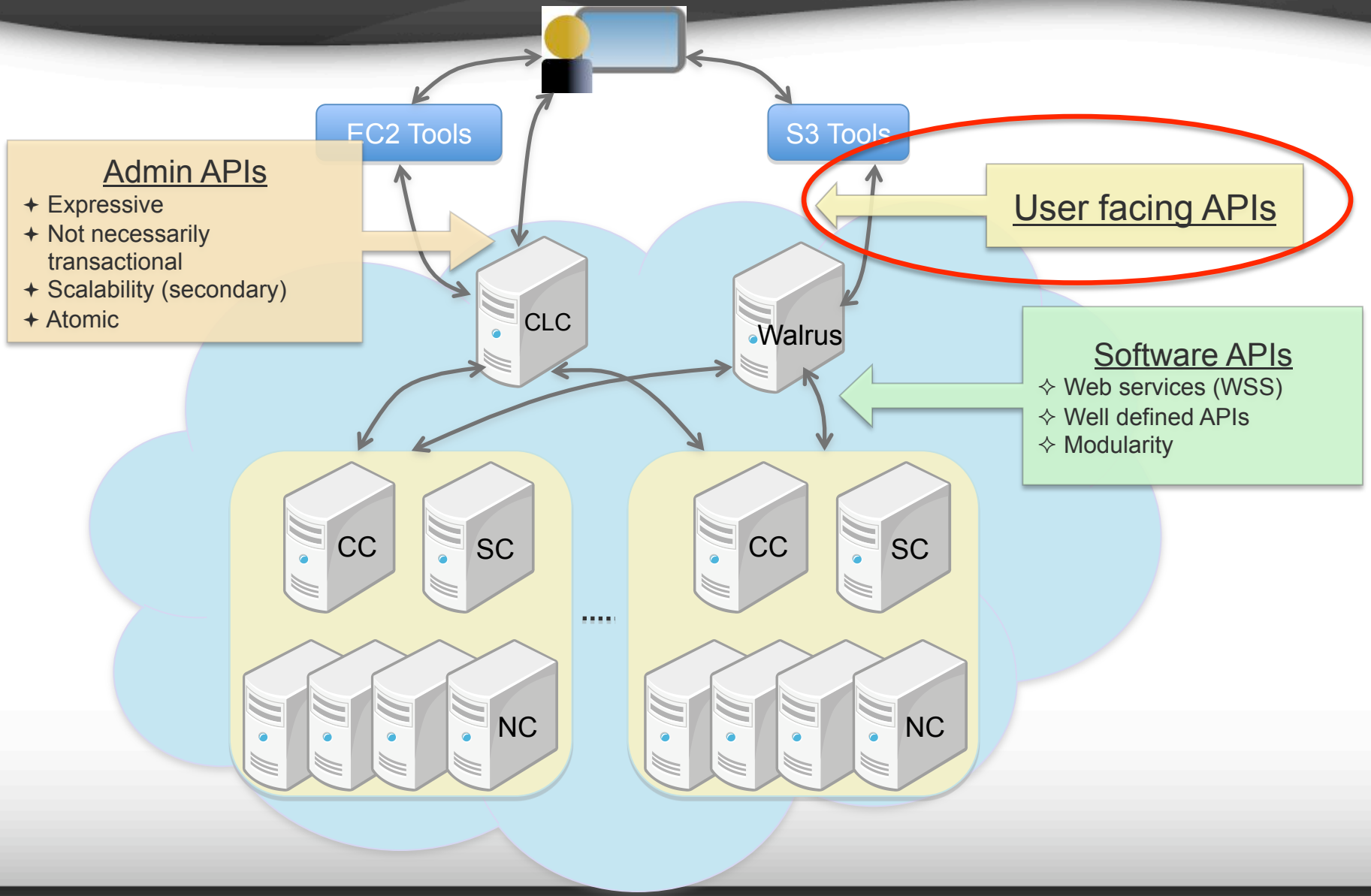


Cloud API Requirements

Eucalyptus

- **Scalability**
  - Resources
  - Users
- **Transactional**
  - Ease of use
  - Failure management
  - Low cost

Distributed Cloud Computing APIs



API Characteristics - I

Eucalyptus

- **Asynchronous APIs**
  - Deadline driven
  - Client-server agree on a time frame for a response
  - Applications have a defined way to check health/ status

  - *User scale*

API Characteristics - II

Eucalyptus

- **API avoids aggregate operations**
  - Requests refer to “singleton” resources
  - Transactional semantics should scale
  - Simple scheduling
  - No state-locking needed

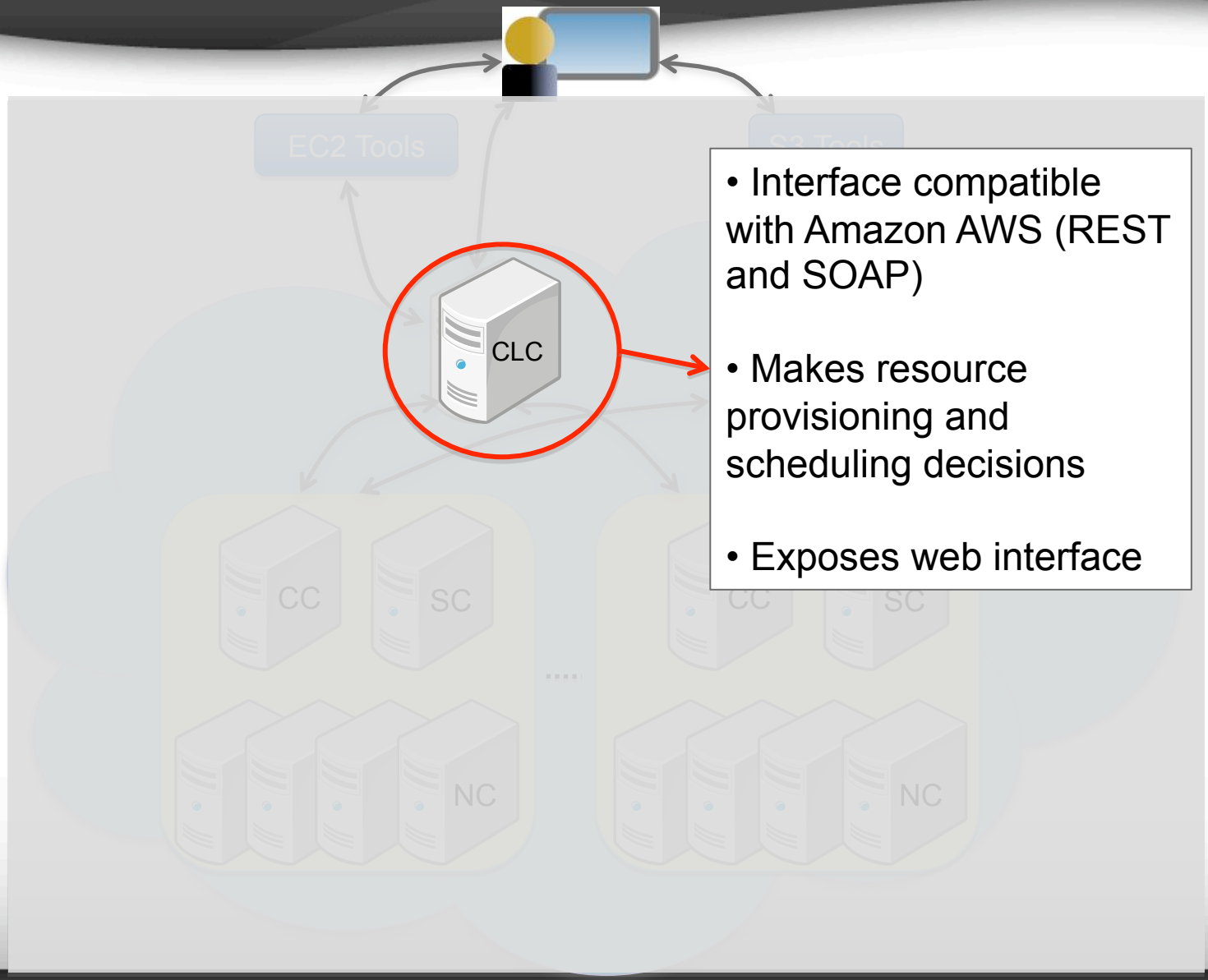
  - *Resource scale*

API Characteristics - III

Eucalyptus

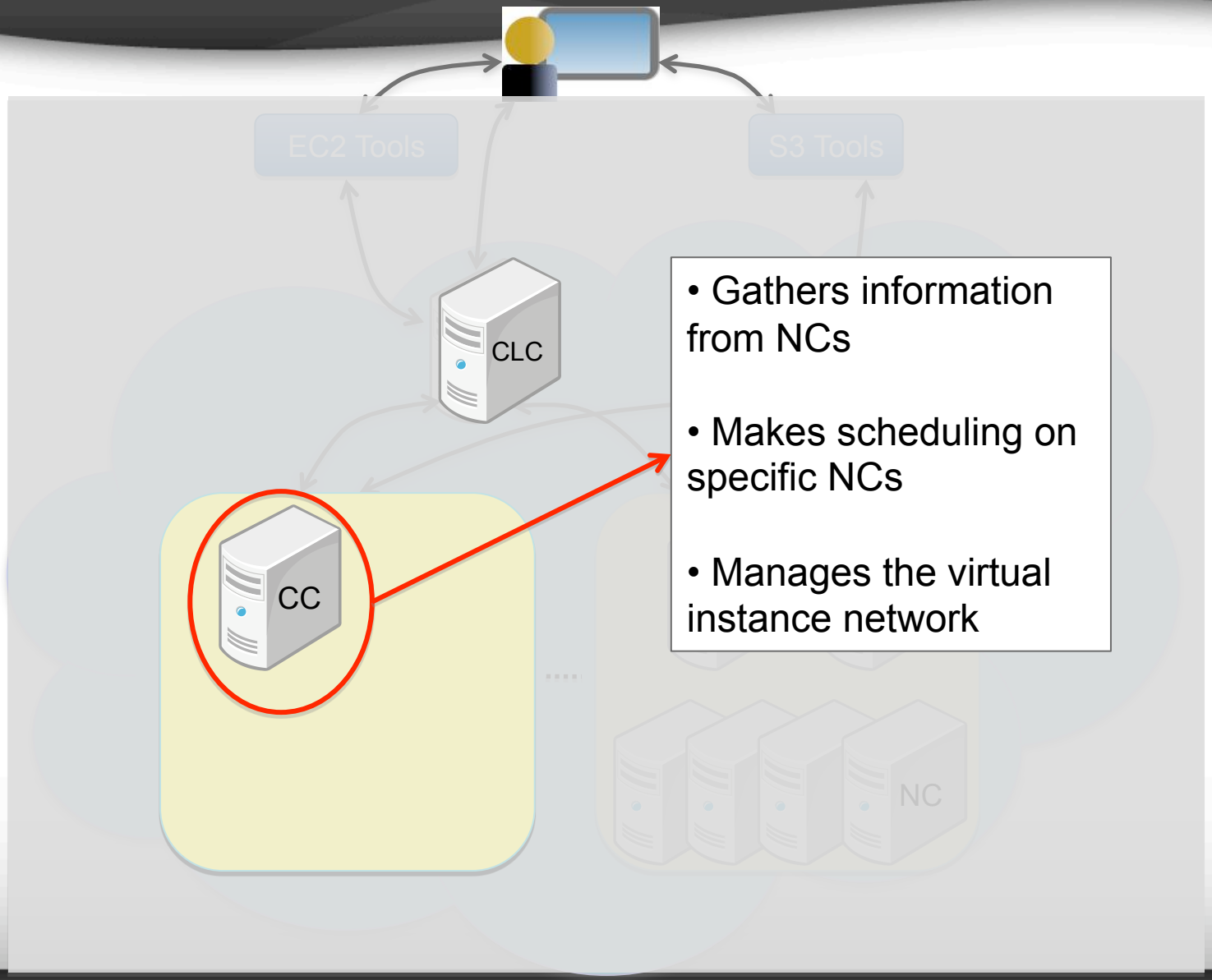
- **API mandates scalable persistent abstractions**
  - S3
    - Eventual consistency
    - Transactional writes
  - EBS
    - Limited scope to an availability zones
    - No sharing within the cloud semantics
  - Ephemeral
    - Very scalable – ‘cos you just destroy them!

Architecture

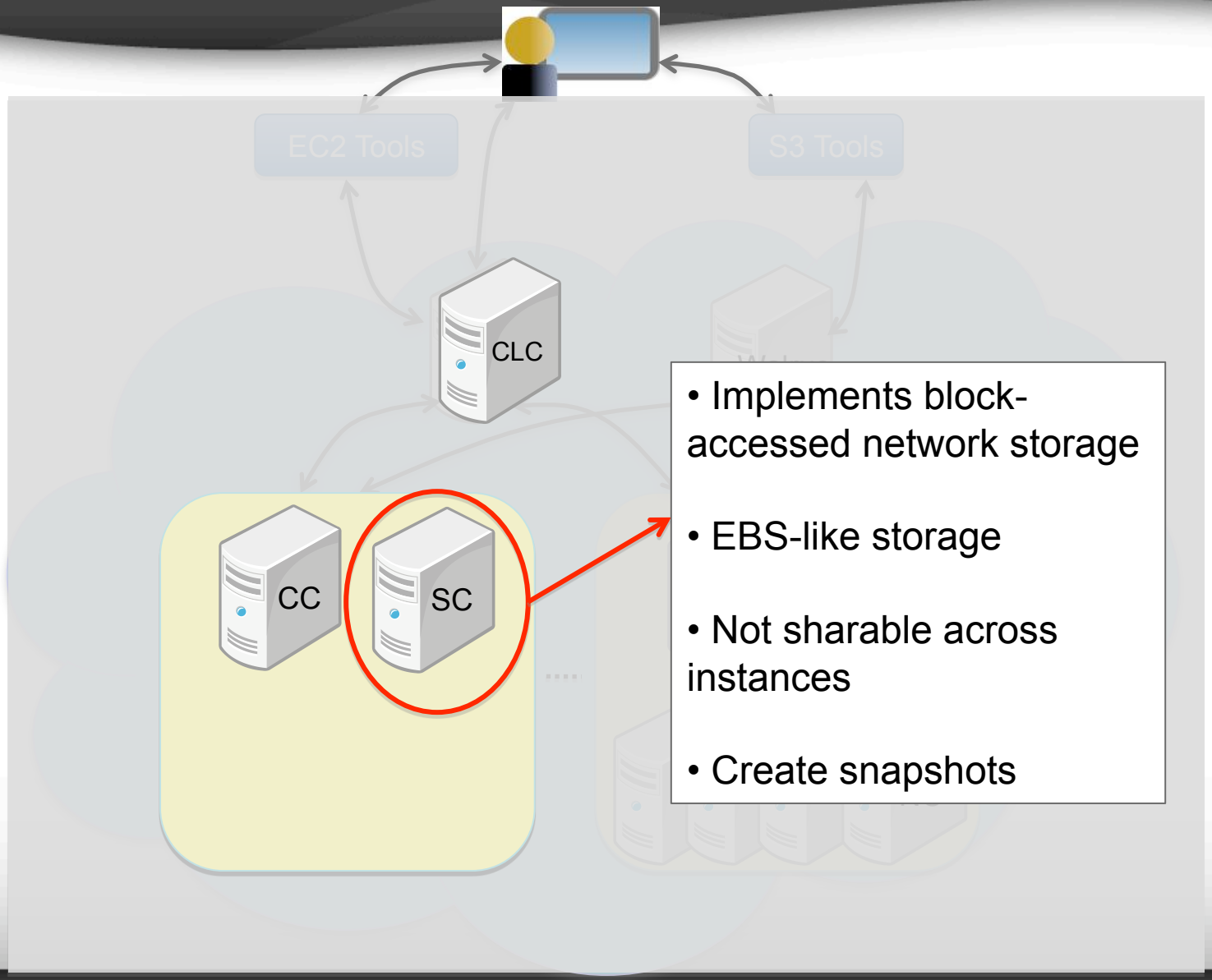


- Interface compatible with Amazon AWS (REST and SOAP)
- Makes resource provisioning and scheduling decisions
- Exposes web interface

Architecture

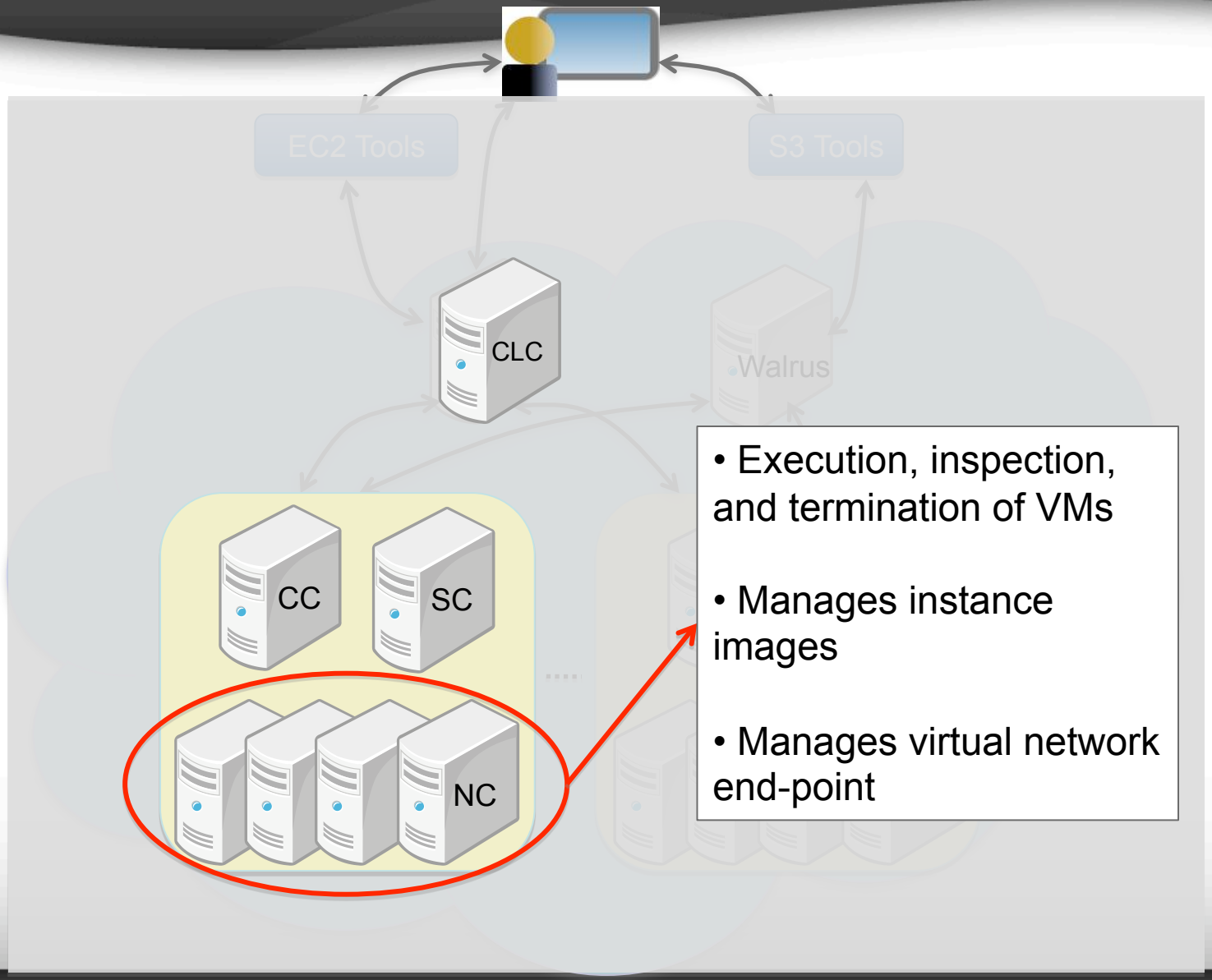


Architecture

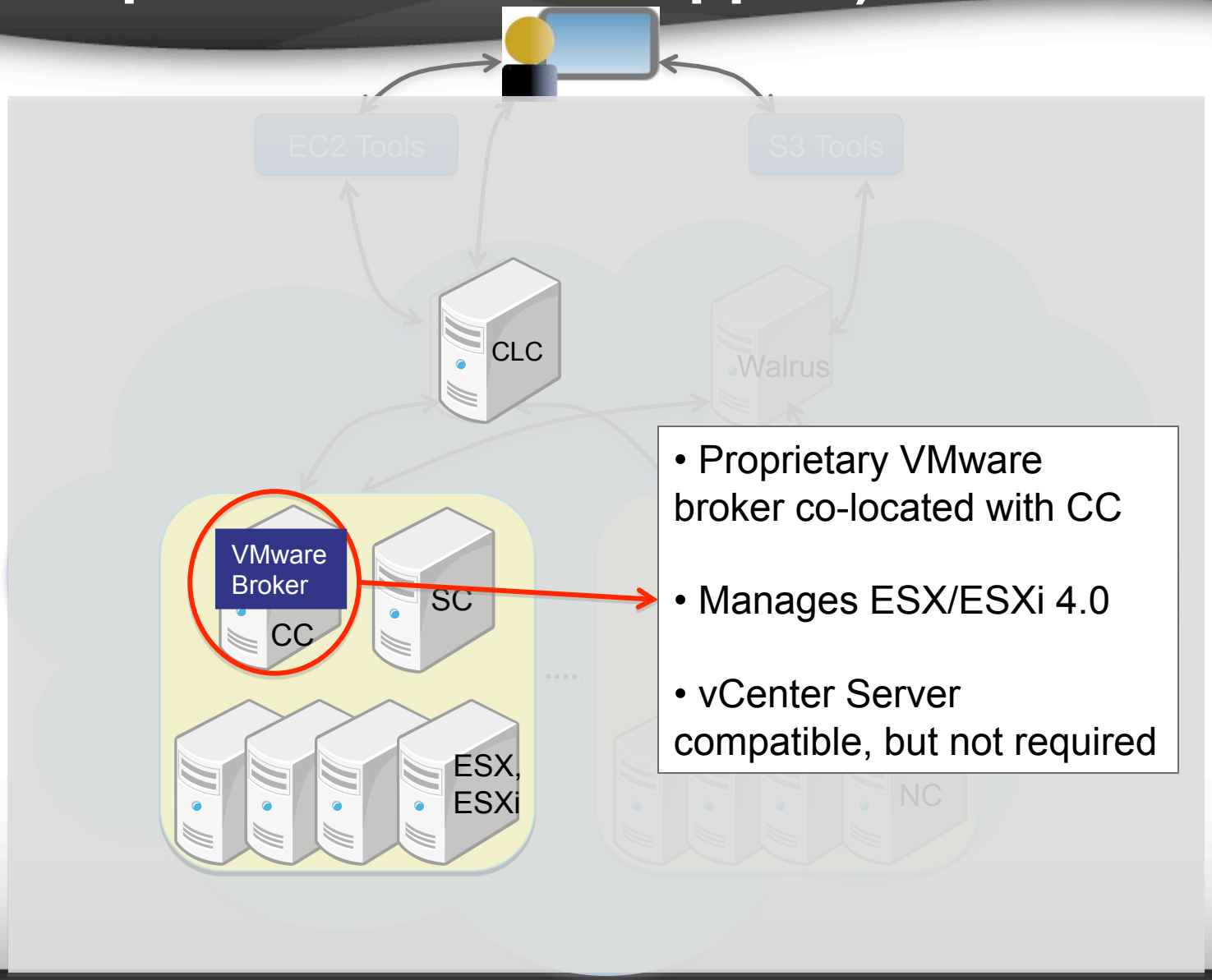


- Implements block-accessed network storage
- EBS-like storage
- Not sharable across instances
- Create snapshots

Architecture

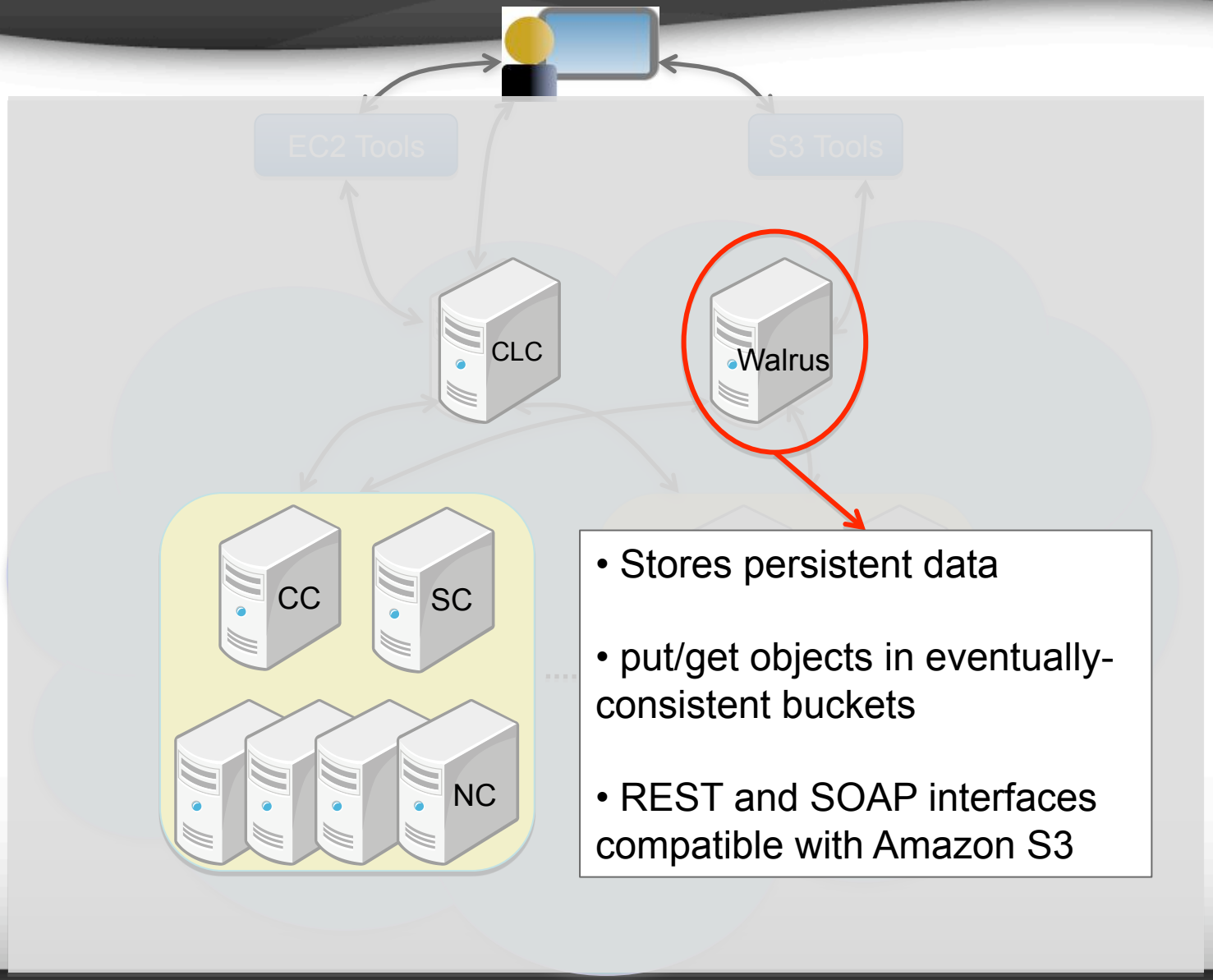


Architecture (Enterprise – VMware support)



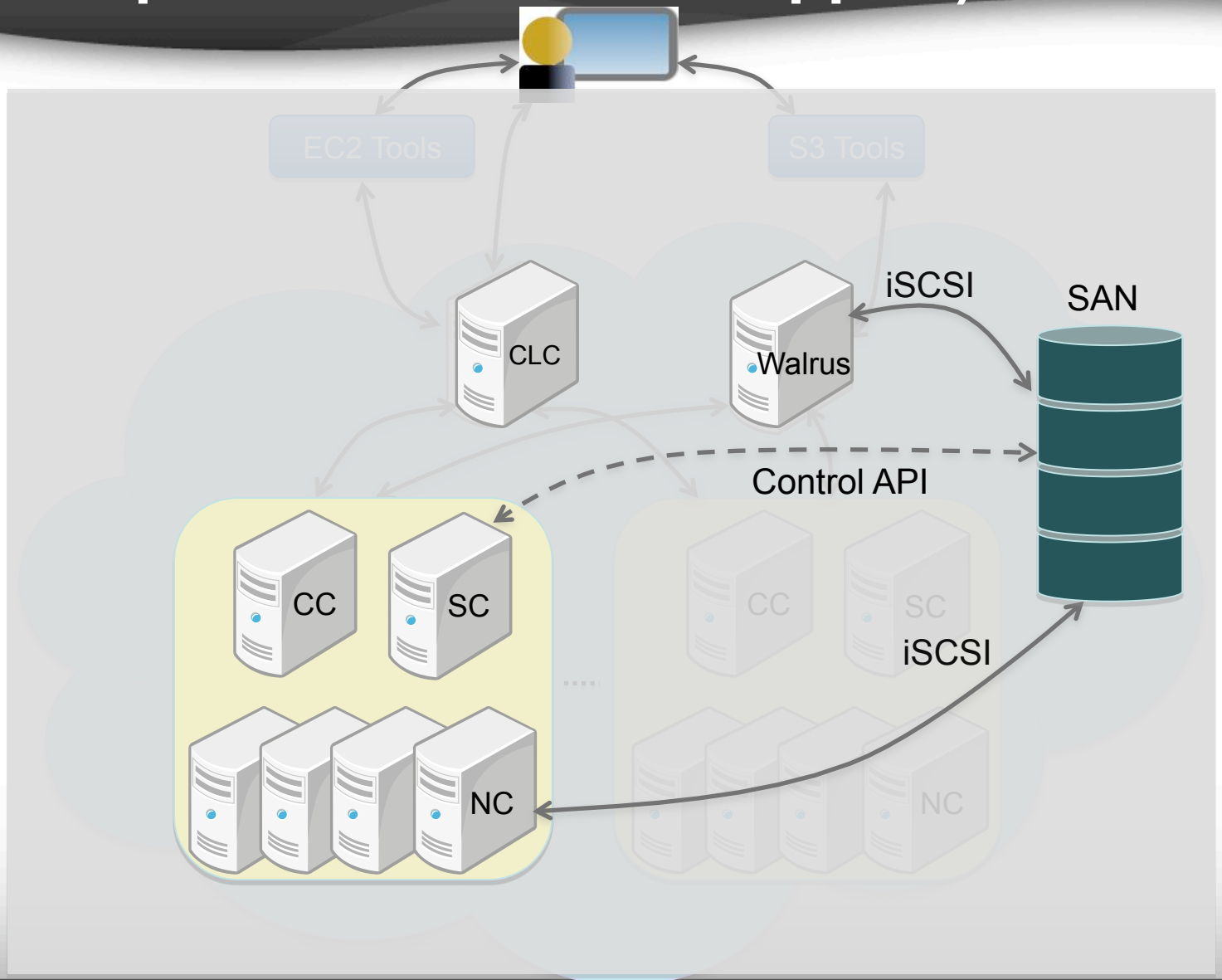
- Proprietary VMware broker co-located with CC
- Manages ESX/ESXi 4.0
- vCenter Server compatible, but not required

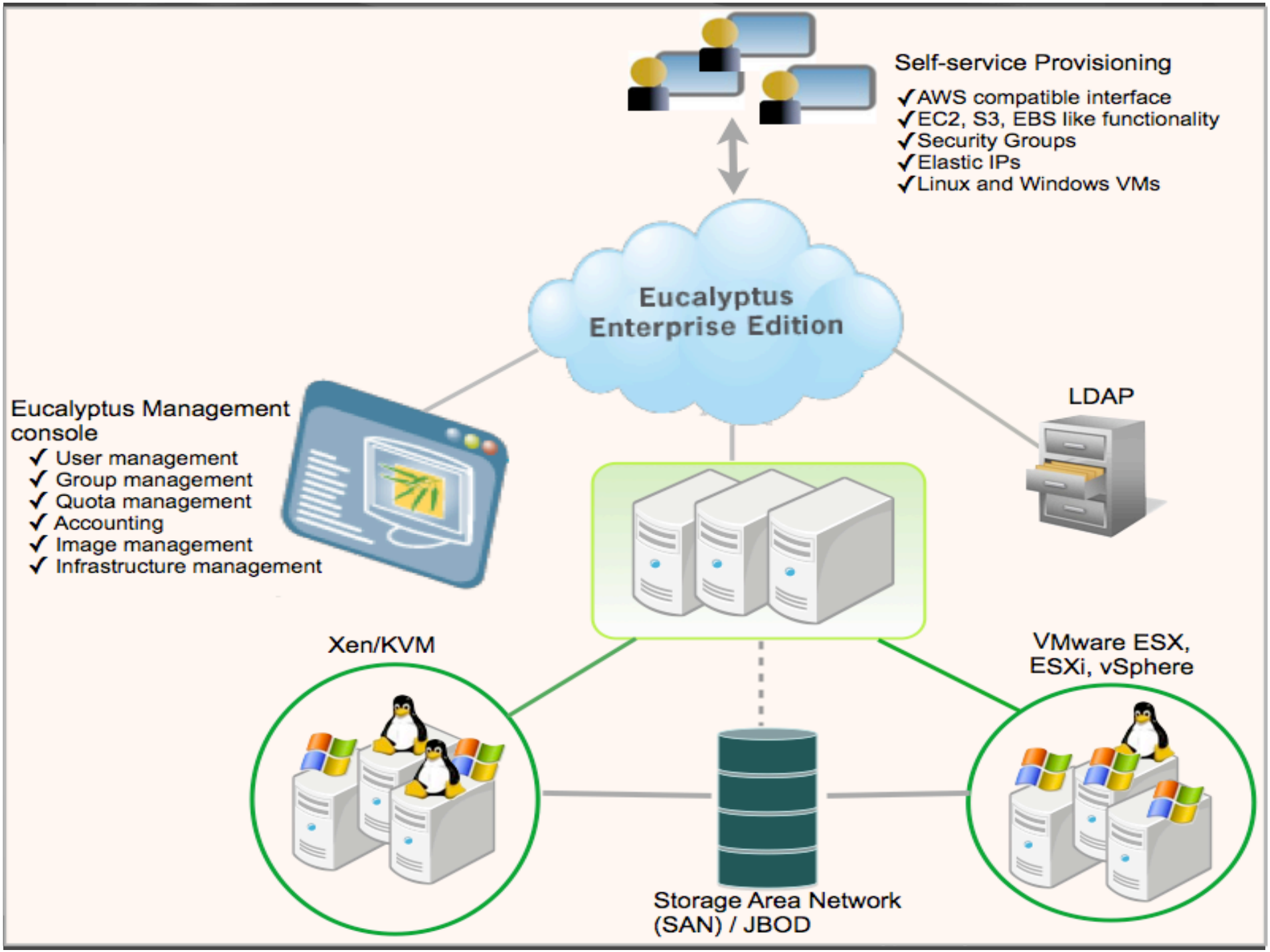
Architecture



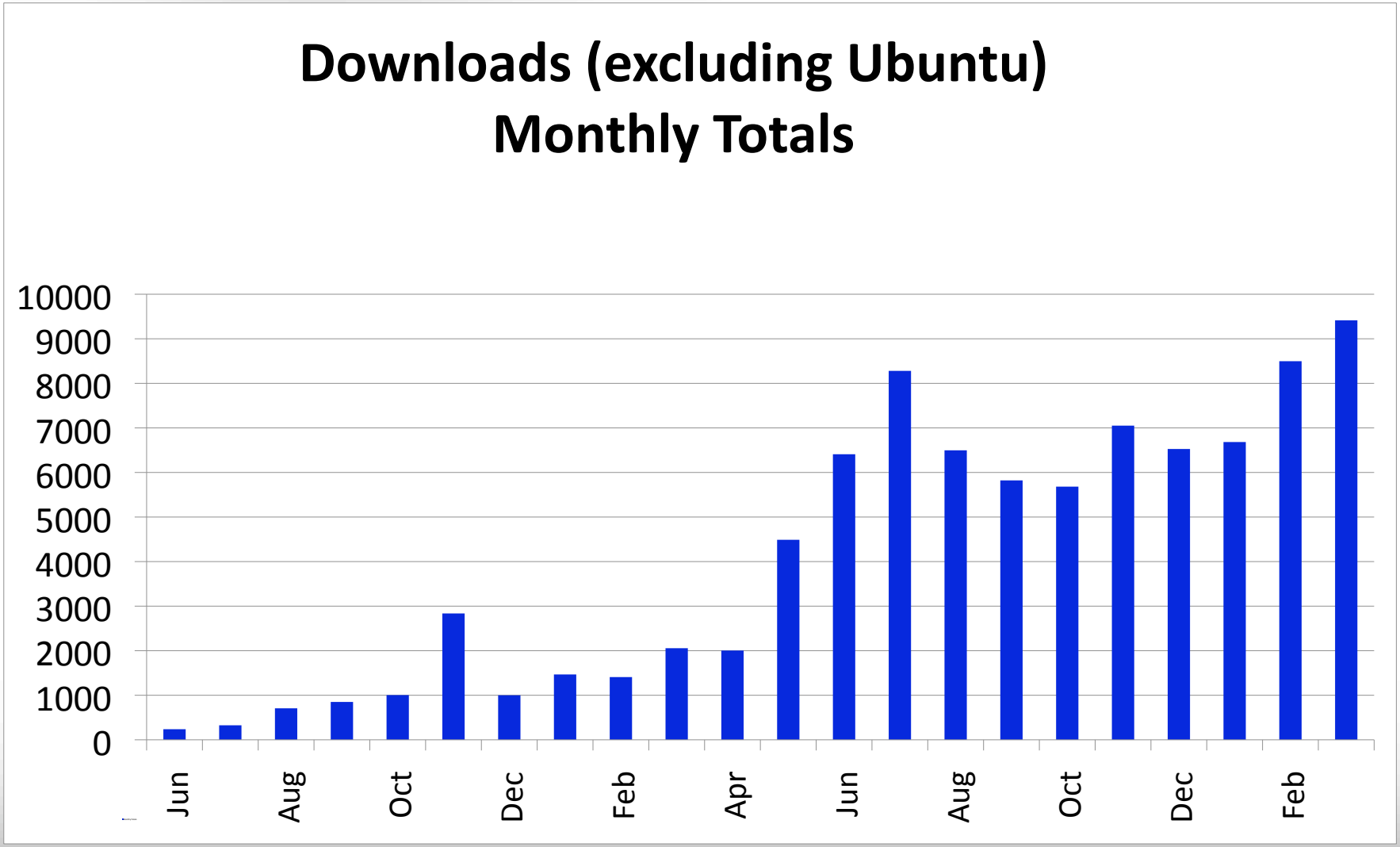
- Stores persistent data
- put/get objects in eventually-consistent buckets
- REST and SOAP interfaces compatible with Amazon S3

Architecture (Enterprise – iSCSI SAN support)




Self-service Provisioning
✓AWS compatible interface
✓EC2, S3, EBS like functionality
✓Security Groups
✓Elastic IPs
✓Linux and Windows VMs
Eucalyptus
Enterprise Edition
Eucalyptus Management console
✓ User management
✓ Group management
✓ Quota management
✓ Accounting
✓ Image management
✓ Infrastructure management
LDAP
Xen/KVM
VMware ESX, ESXi, vSphere
Storage Area Network (SAN) / JBOD

Uptake


Downloads (excluding Ubuntu)
Monthly Totals
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Jun
Aug
Oct
Dec
Feb
Apr
Jun
Aug
Oct
Dec
Feb

No Eucalyptus in Antarctica (yet)

Eucalyptus

Open-source Distribution

Eucalyptus

Linux Distribution: *Ubuntu* and Eucalyptus

- Jaunty Jackalope "Powered by Eucalyptus"
  - April 23, 2009
  - Complete build-from-source
- Karmic Koala
  - October 23, 2009
  - Full-featured Eucalyptus
- Fundamental technology
  - "Ubuntu Enterprise Cloud" ecosystem surrounding Eucalyptus
- 10,000,000 potential downloads
- *Debian* "squeeze"
  - Source release packaging under way
- Packaged for *CentOS*, *OpenSUSE*, *Debian*, and *Ubuntu* as "binary" release as well

> "Make Eucalyptus the open source reference implementation for cloud computing."
>
> Simon Wardley (head of cloud strategy), Canonical

Customer Pain Points

Eucalyptus

- Self-service procurement of IT resources not available
- Automation of resource delivery isn't possible
- Procuring machines takes weeks or months
- End-users want access to new and legacy resources
- More efficient use of resources is required

Use Cases

Eucalyptus

- **Testing/Development**
- **Education/Training**
- **Web Services deployment**
- **Quickly create and tear-down IT environments**
- **Hybrid clouds**
  - Cloud bursting
  - Disaster recovery
  - Back up

Thanks!

Eucalyptus

- Thanks to our original research sponsors…

- …and to our new commercial friends

www.eucalyptus.com
805-845-8000
shashi@eucalyptus.com