Cloud Computing Research

USENIX LISA'10

RC2 : A Living Lab for Cloud Computing

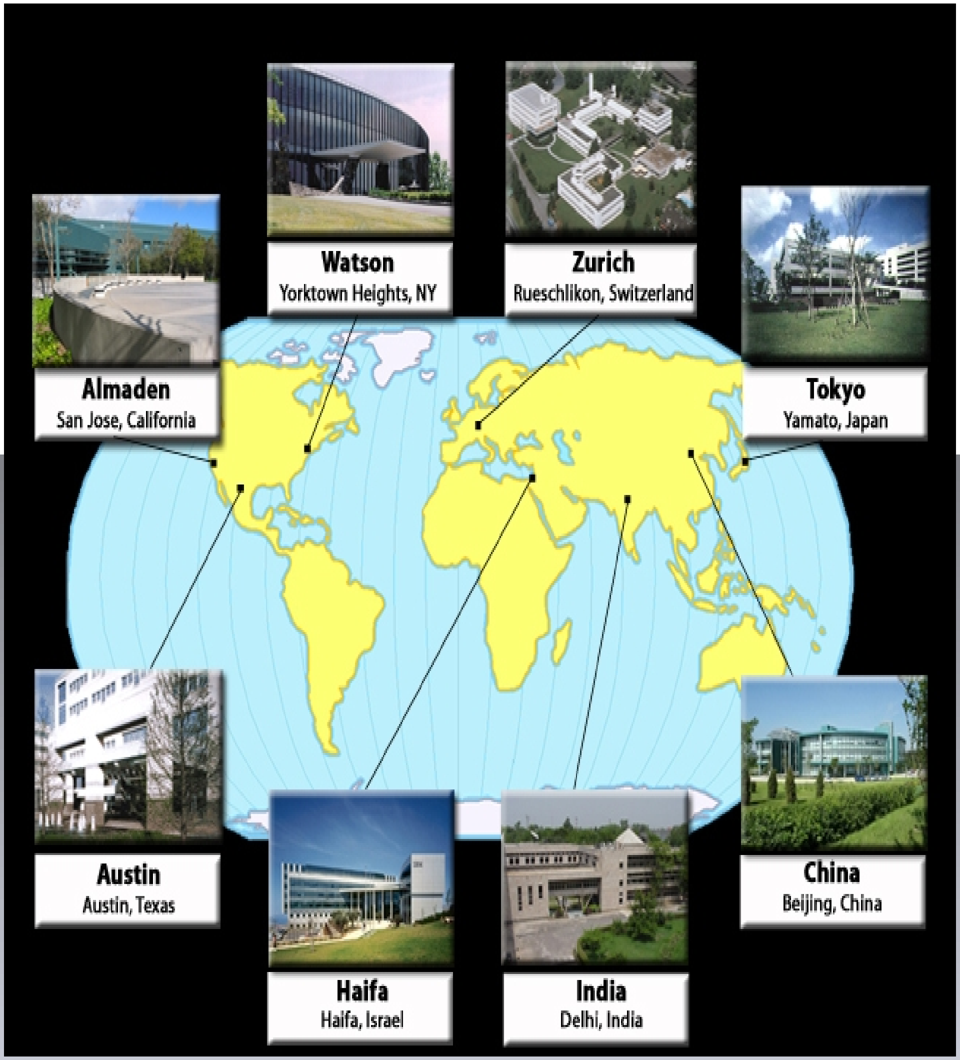
Kyung Ryu, Xiaolan Zhang, Glenn Ammons, Vasanth Bala, Stefan Berger, Dilma M Da Silva, Jim Doran, Frank Franco, Alexei Karve, Herb Lee, James A Lindeman, Ajay Mohindra, Bob Oesterlin, Giovanni Pacifici, Dimitrios Pendarakis, Darrell Reimer, Mariusz Sabath

IBM T.J Watson Research Center

Nov 11th 2010

Research Compute Cloud for IBM Research Worldwide

- 9 research centers distributed around world
- A variety of IT experiment labs on each site
- Lots of lab machines used for experiments and getting dusted
- Extensive cloud computing research and experiments



Objectives and Challenges

Mission (Maybe Not) Impossible:

"Chase two rabbits and catch both"

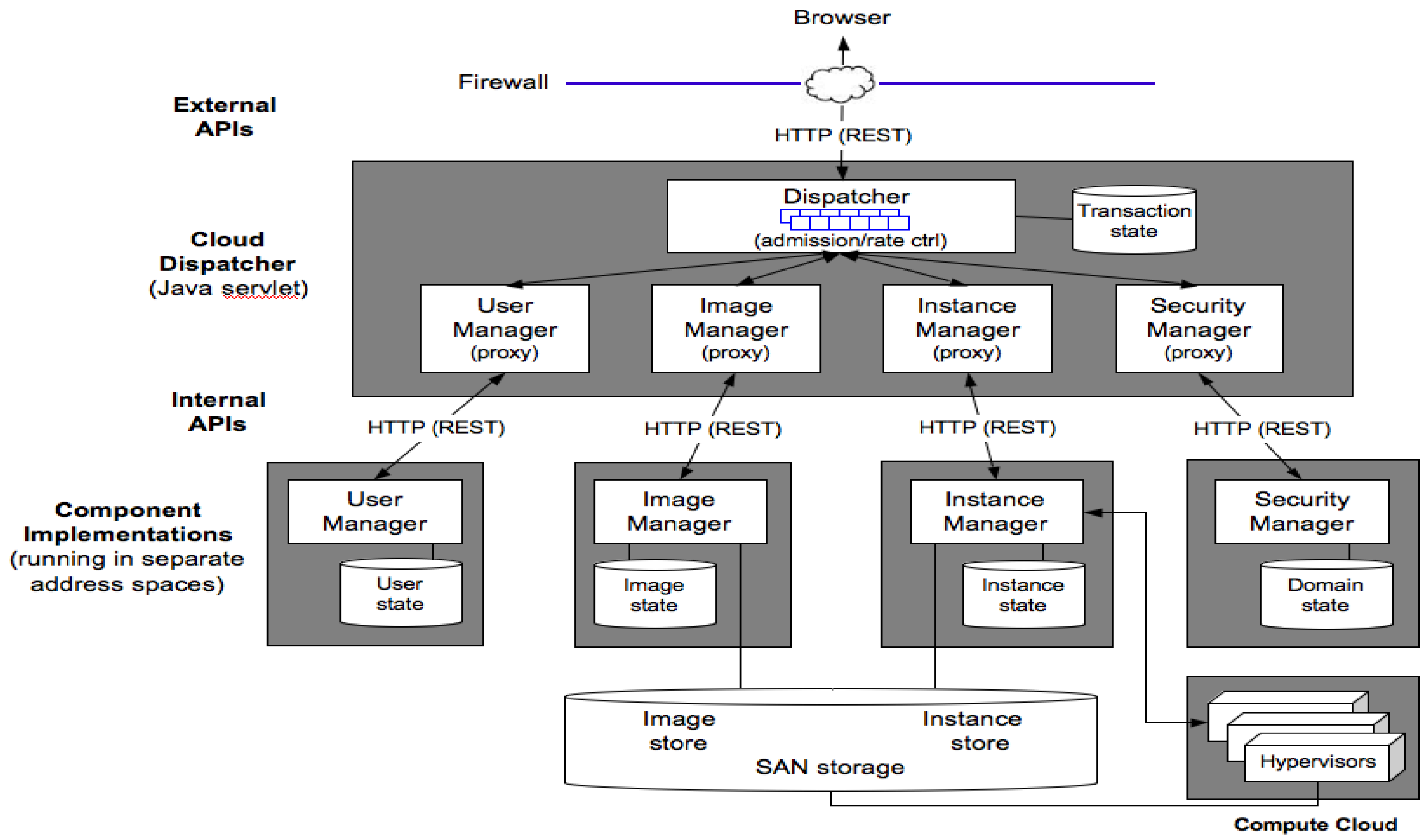
- Commercial-grade IaaS for Semi-public Cloud
  - Serve worldwide IBM research community (and beyond)
- Playground for quick Cloud technology experiments and transfer
  - IBM's cloud computing initiatives and businesses

Yet, Another Challenge:

- Support various types of systems and virtualization mechanisms
  - xSeries (x86): virtual machines through xen, kvm, etc
  - pSeries (Power): LPARs through phyp
  - zSeries (mainframe): VMs through native virtualization

RC2 Architecture

- Extensible and Pluggable



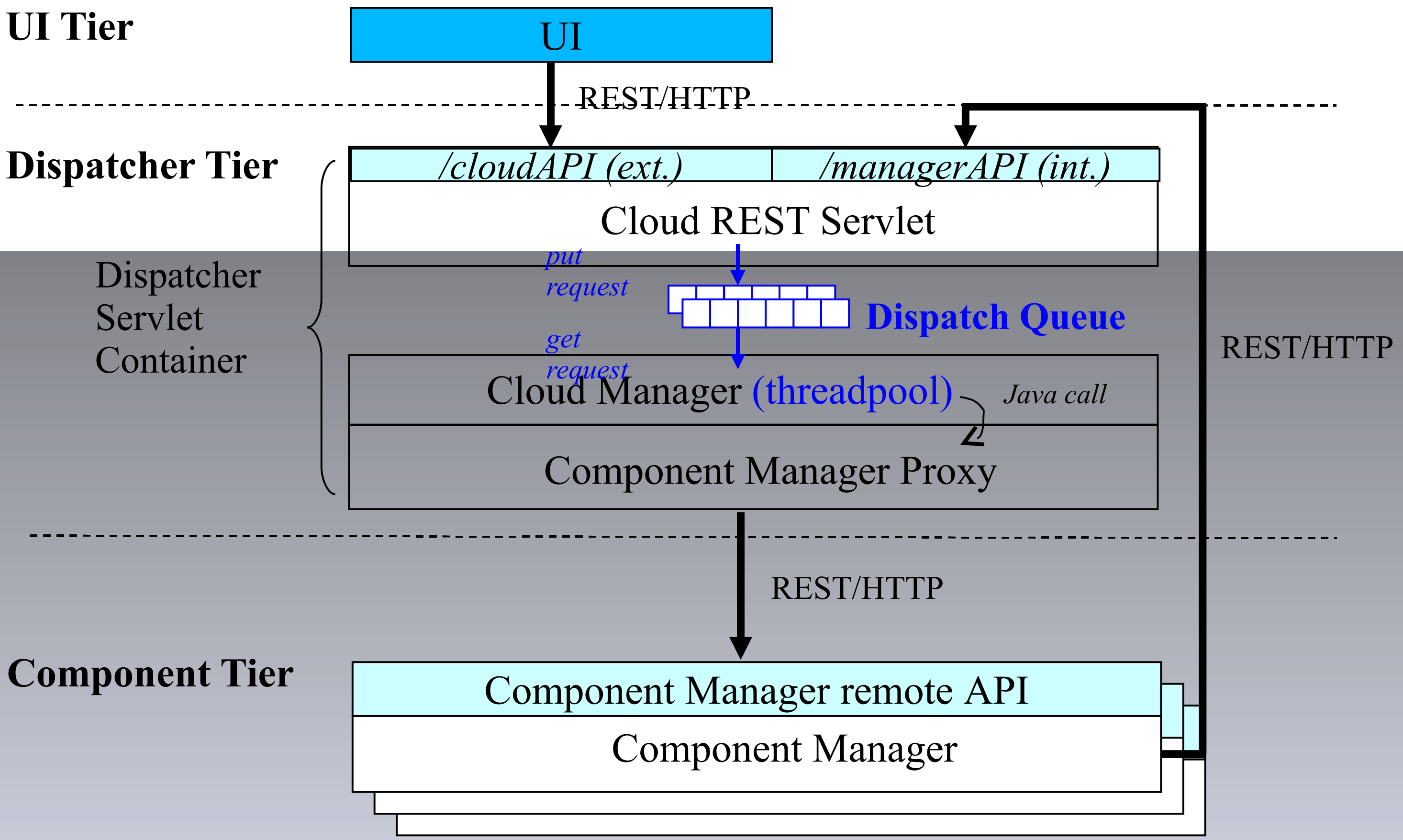
Smart Cloud Dispatcher

Make RC2 Operating Infrastructure More Scalable and Reliable.

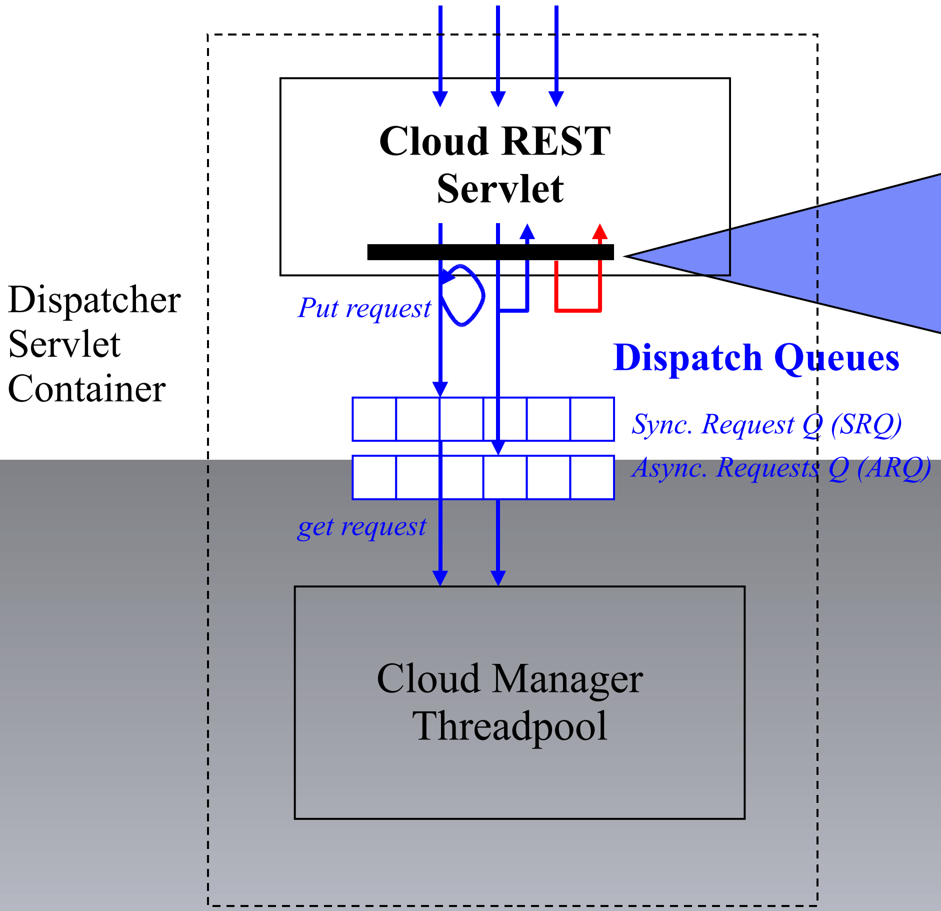
- **Handle impedance mismatch** between user requests and back-end component managers and avoid overload and crash
- **Respond quickly** to light-weight requests
- Provide a request delivery mechanism that **allows component manager to scale-out**
- Extensible to **scale dispatcher itself**

Smart Cloud Dispatcher

- Request Queues for Dispatcher and Component Managers



Dispatch Queue

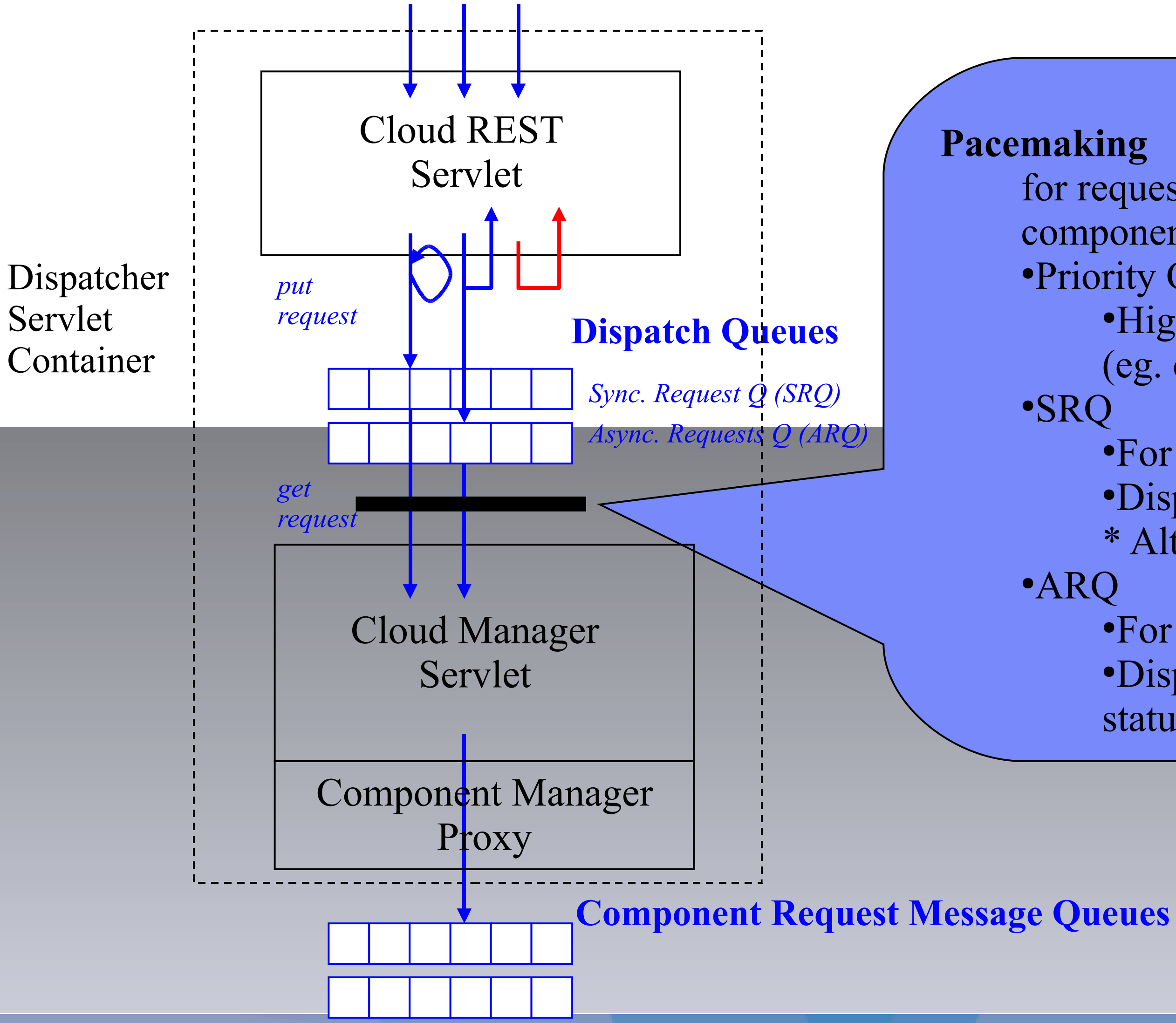


Gate Keeping
for early and simple Admission Control

- Admit
  into Queue with Request(TR) ID and blocks for result for sync. requests
- Reject
  if queue is full or component is down

* **time-to-live** setting for request expiration
* **priority** setting for internal requests
* **cancel** waiting requests in queue

Dispatch Queue (cont.)

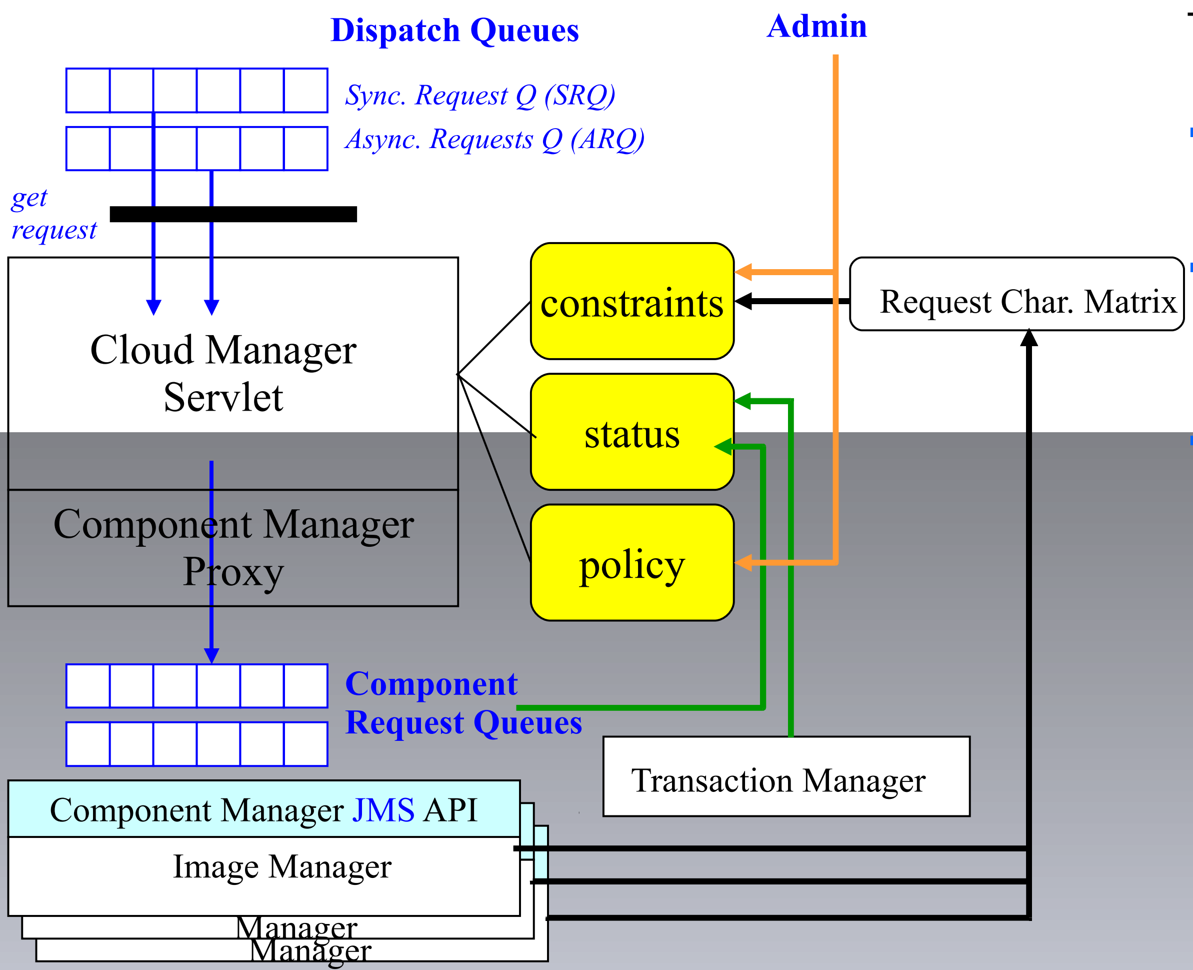


Pacemaking

for request throttling to keep pace with component managers

- Priority Queues
  - High priority for internal requests (eg. checkin/checkoutImage)
- SRQ
  - For light requests
  - Dispatch requests immediately
  - * Alt. use blocking call w/o queue
- ARQ
  - For heavy requests
  - Dispatch based on constraints, status, policy on components load

End-to-End Pacemaking (Ongoing)



Throttling model based on constraints-status-policy:

- **Constraints:** maximum 256 concurrent checkoutImage requests allowed
- **Status:** number of active and outstanding (queued) checkoutImage requests are monitored
- **Policy:** dispatch if number of active/outstanding checkinImage requests < 256 – 25 (10% safety buffering)

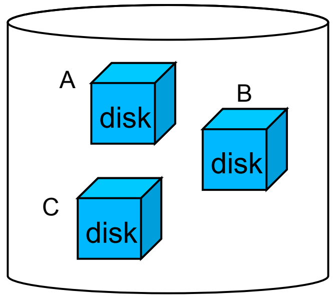
Image Management

- Catalogs, accesses, and maintains VM images
  - ListImages
  - DescribeImage *imageId*
  - AddImage *directoryURL imageName*
  - *CheckoutImage *imageId directoryURL*
  - *CheckinImage *directoryURL imageName*
  - DeprecateImage *imageId*
  - PublishImage *imageId*
  - UnpublishImage *imageId*

- *Metadata and Provenance*
  - *Name, description, owner, ACL, parent ID*

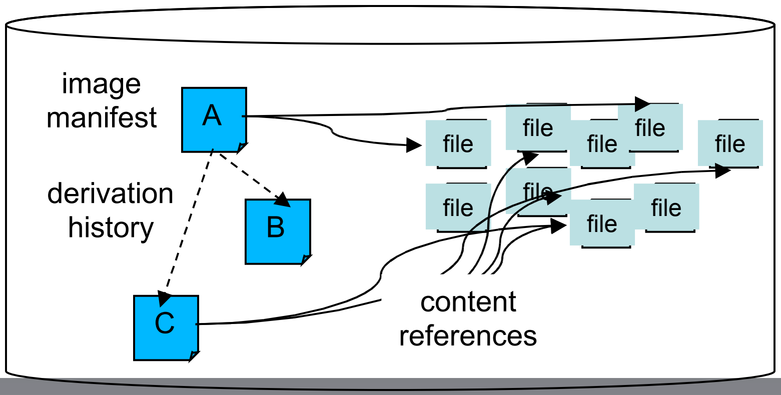
Mirage Image Library

Conventional image library



Disk granularity store

Mirage image library



Content addressable, file granularity store

- Disk based representation
- No image relationships
- Hypervisor-dependent
- Merely a storage system for image disks

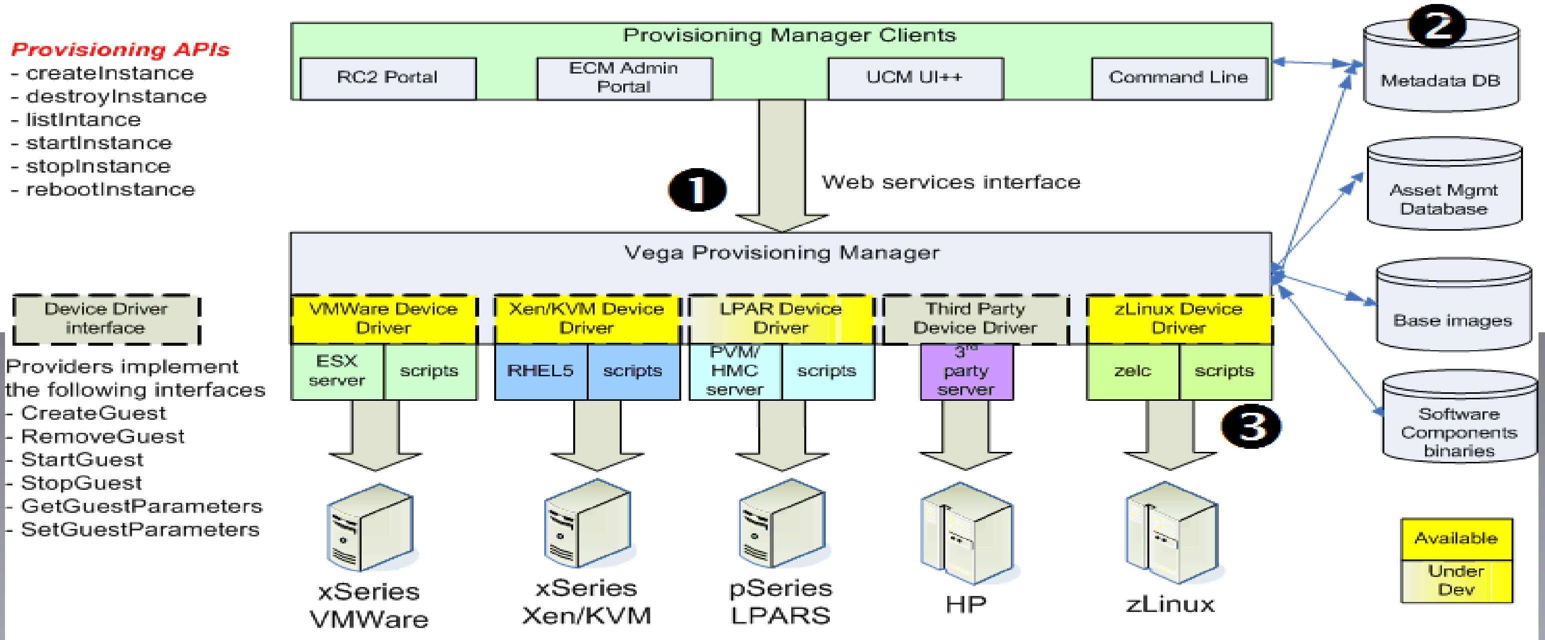
- File based representation
- Image relationships (think CVS)
- Hypervisor-agnostic
- A sophisticated store with APIs to directly manipulate images without deploying them as instances or fully assembling their disks
- Conventional disk is reconstituted when an image is checked out

Instance Management

Provisioning Steps to create Instance

1. Reserve resources - Placement Advisor
   1) Select HostPlatform - Match capabilities with requirements of image
   2) Reserve guest IP address and resources for Instance
1. Register instance parameters with TPM for tracking the instance
2. Request Image Manager to checkout/clone the image to the HostPlatform
3. Fixup the image before boot – Copy ssh keys
4. Setup the Activation Engine parameters on Activation Device (floppy/cdrom). These parameters are for fixup during boot
5. Register the VM with Hypervisor
6. Start the VM – This will complete the fixup
7. Wait for VM to start (ping/ssh)
8. Notifications
   1) Notify User with Email
   2) Message for Compliance Tracking

VEGA provisioning abstraction layer



Provisioning APIs
- createInstance
- destroyInstance
- listIntance
- startInstance
- stopInstance
- rebootInstance

Providers implement the following interfaces
- CreateGuest
- RemoveGuest
- StartGuest
- StopGuest
- GetGuestParameters
- SetGuestParameters

Provisioning Abstraction Layer

❶ Provisioning API	Web- services API for requesting provisioning services
❷ Provisioning Metadata	Metadata to capture provisioning requirements including virtual image definitions (based on OVF)
❸ Provisioning DeviceDriver	Service Provider Interface to plug provisioning manager implementations targeting specific infrastructure &virtualization technology; essentially consists of a set of workflows (uses embedded TPM capabilities)

Image definition is an xml file consisting of the description of the composite image with parameters that provide additional details of the images
Image content is a tarball of the file system and a set of content and configuration scripts to configure the OS and software stack on the image.

Security Management

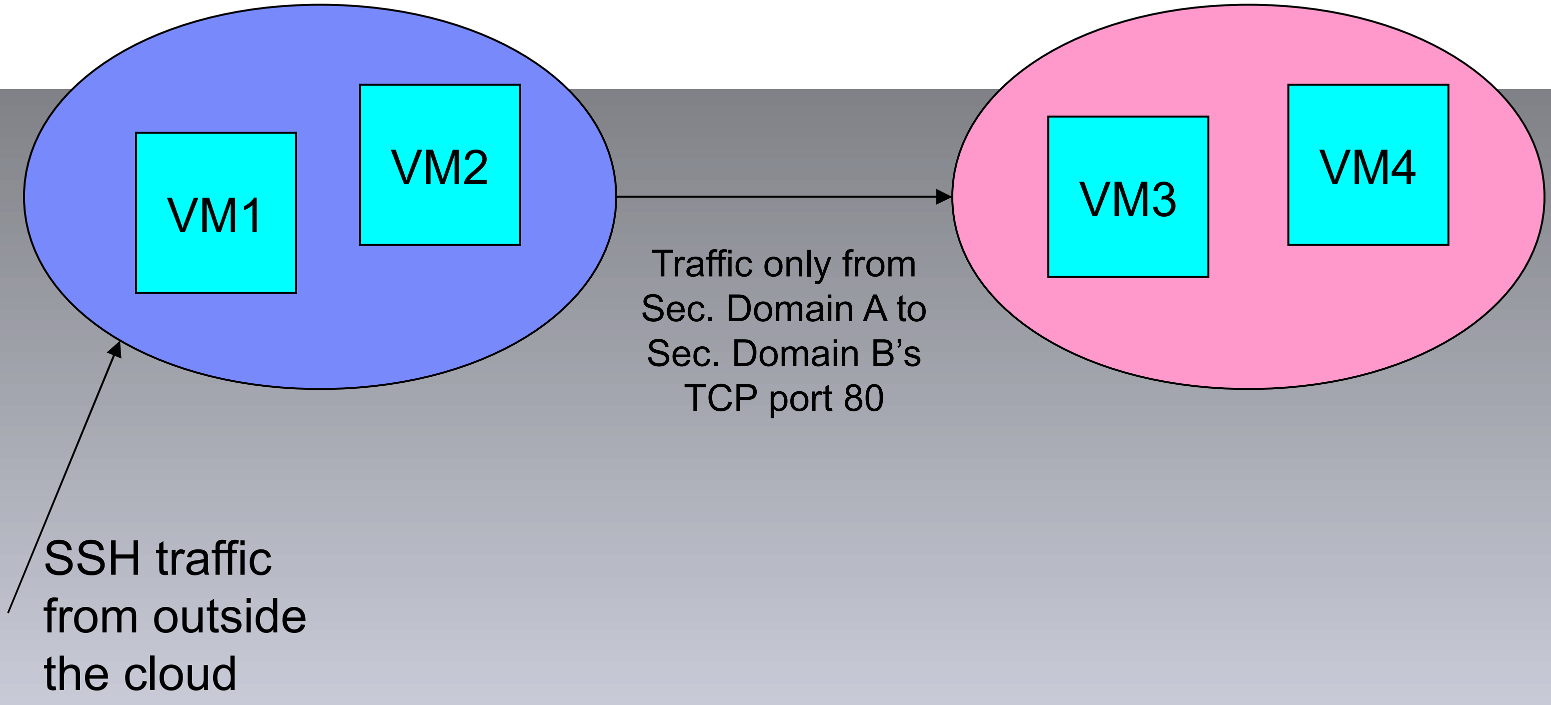
- Goals of the Security Manager
  - Realization of Trusted Virtual Domain (TVD)
    - Isolation between different cloud users' workloads
    - Grouping of VMs of the same or different users (security domains)
    - Enable controlled collaboration between users

- Layered approach
  - Xen daemon extended for applying filtering rules for layer 2 to layer 4 traffic
    - Based on Linux filtering : ebtables and iptables
    - Prevent MAC, IP and ARP spoofing
    - Filtering of traffic with other VMs inside the cloud
    - Filtering of traffic with IP addresses outside the cloud
  - Security Manager implements VM grouping support
    - Calculates per-VM filtering policies
    - Pushes policies to Xen daemons

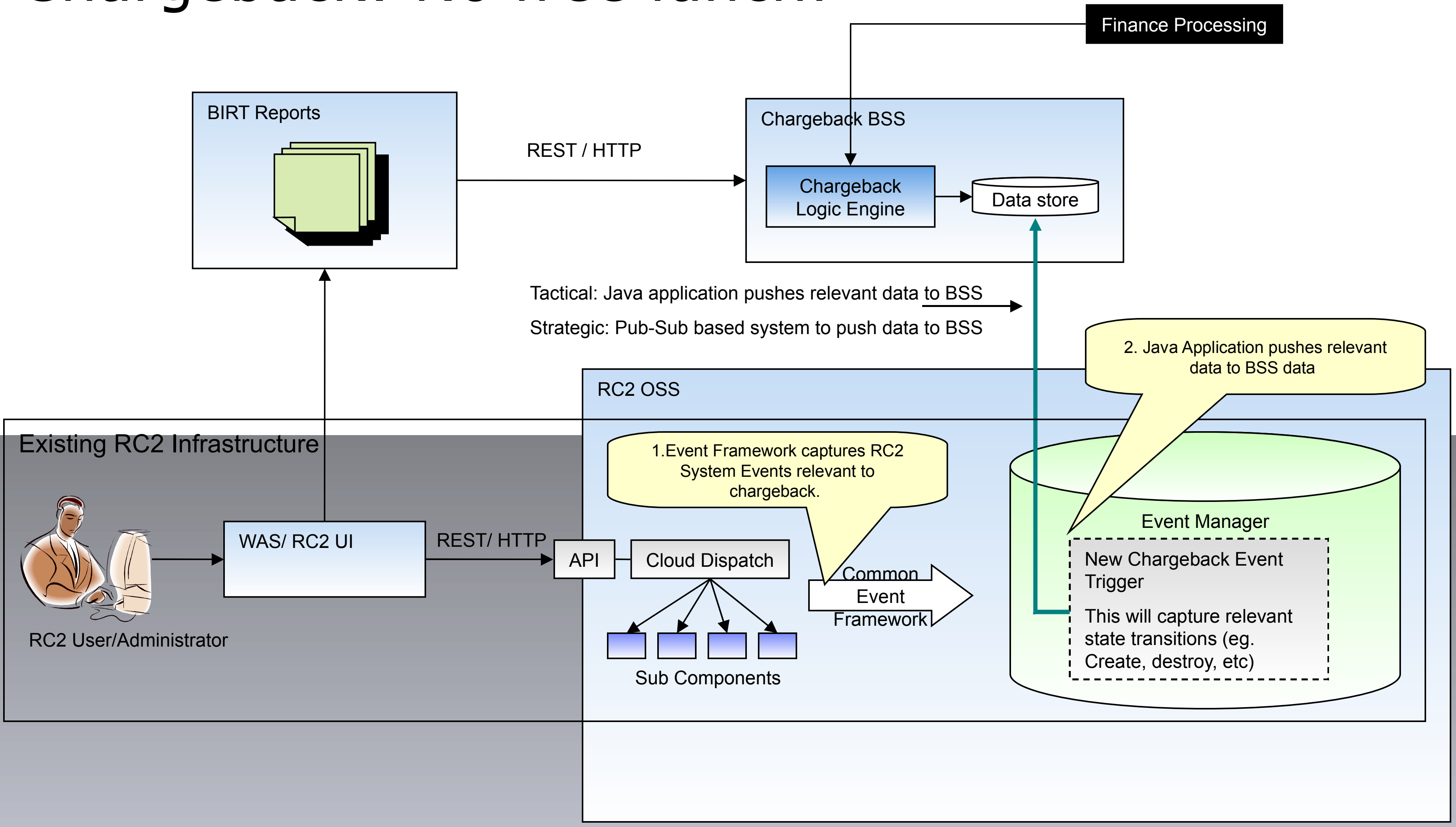
Trusted Virtual Domains

Security Domain A
Owner: User A
Filter: Allow traffic to SSH and VNC ports from outside the cloud

Security Domain B
Owner: User B
Filter: Sec. Domain A may reach TCP port 80

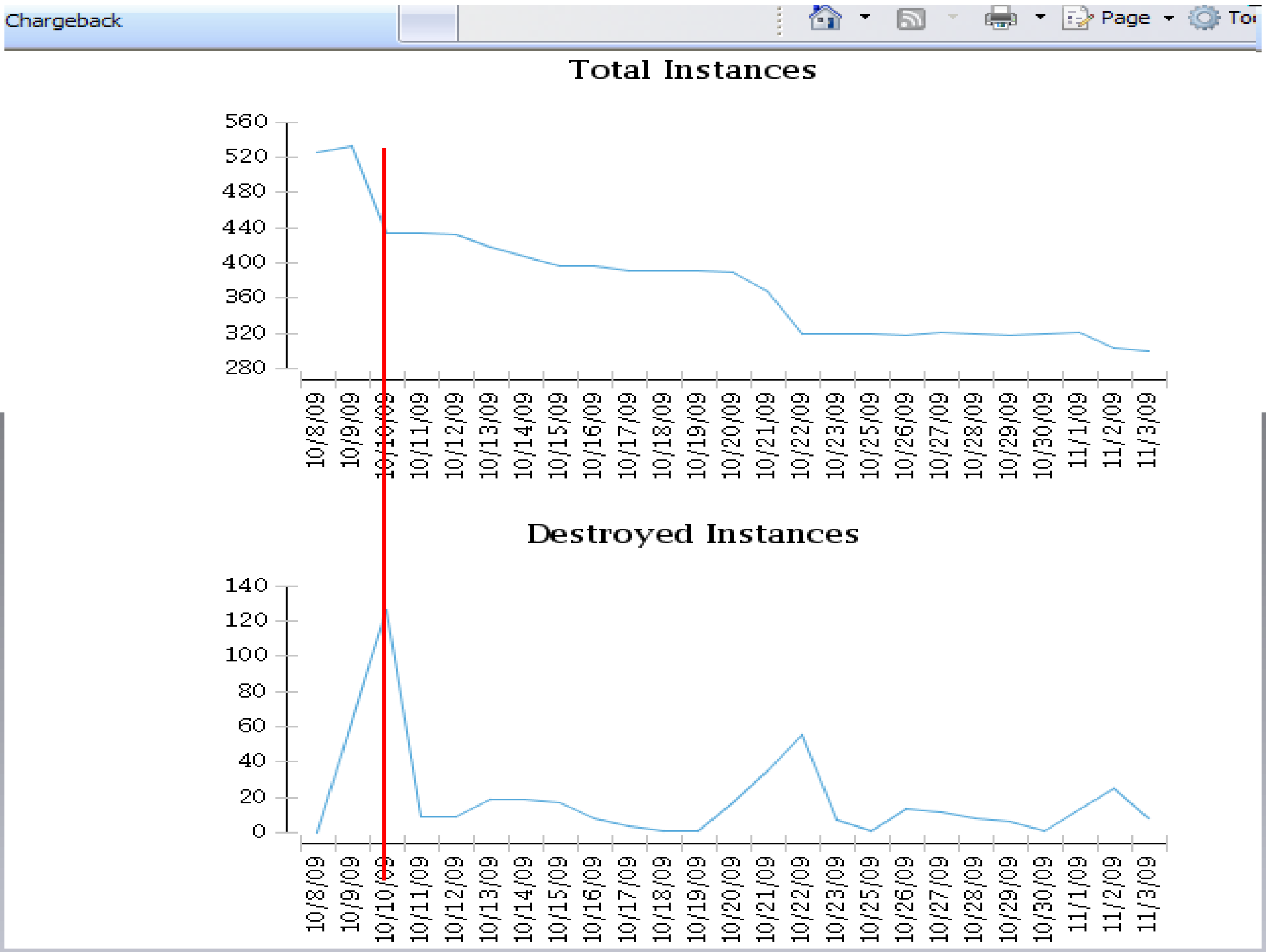

VM1
VM2
VM3
VM4
Traffic only from Sec. Domain A to Sec. Domain B's TCP port 80
SSH traffic from outside the cloud

Chargeback: No free lunch!


Finance Processing
BIRT Reports
REST / HTTP
Chargeback BSS
Chargeback Logic Engine
Data store
Tactical: Java application pushes relevant data to BSS
Strategic: Pub-Sub based system to push data to BSS
2. Java Application pushes relevant data to BSS data
RC2 OSS
Existing RC2 Infrastructure
1.Event Framework captures RC2 System Events relevant to chargeback.
WAS/ RC2 UI
REST/ HTTP
API
Cloud Dispatch
Common Event Framework
Event Manager
New Chargeback Event Trigger
This will capture relevant state transitions (eg. Create, destroy, etc)
Sub Components
RC2 User/Administrator

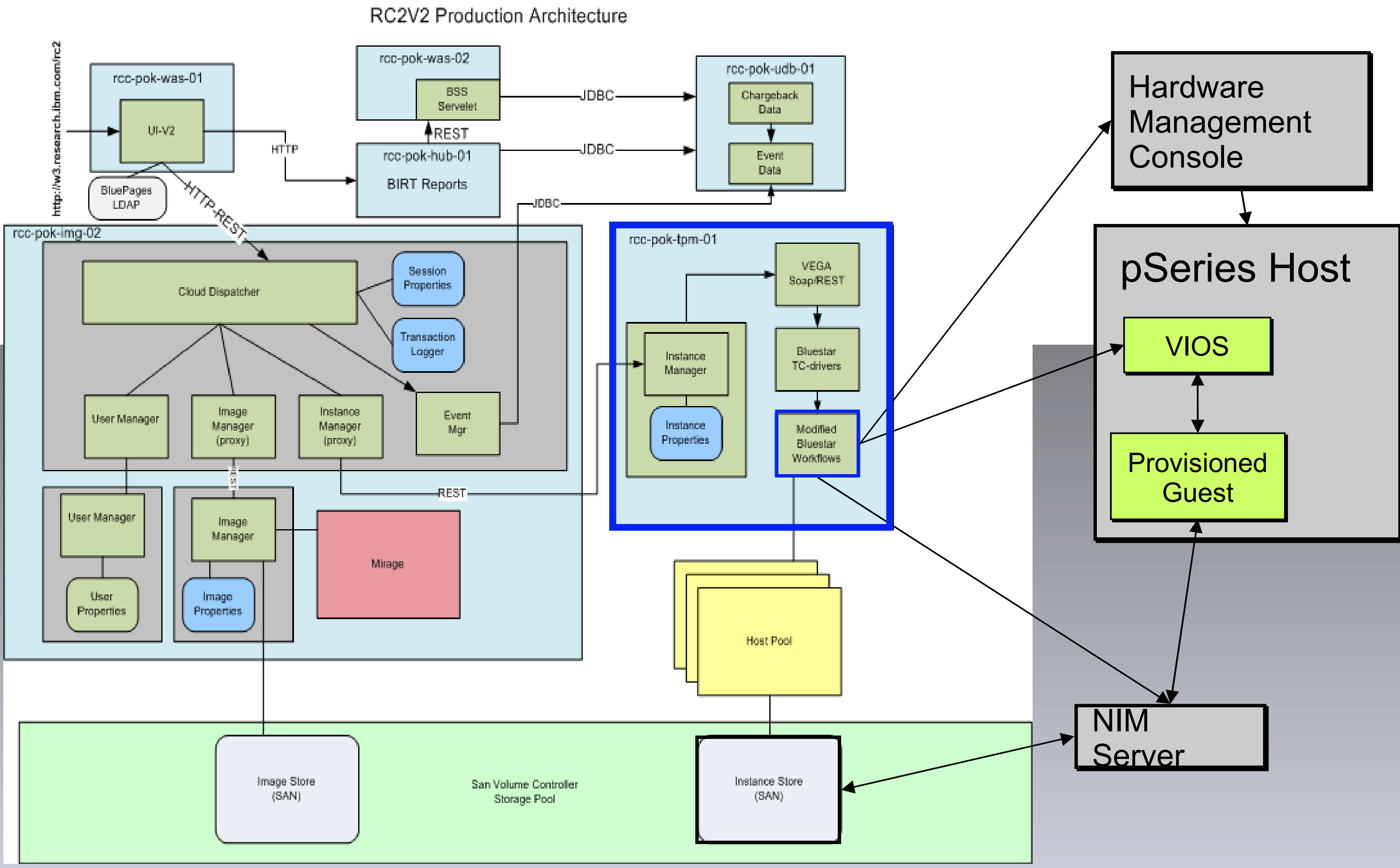
Chargeback Impact

- Money talks!!!



Challenge Case 1: Heterogeneous Cloud

- Provision pSeries (Power) System Instance (LPAR/Phyp)



Challenge Case 2: Strategic Hypervisor Switch

Strategic move from Xen to KVM

- Migrate existing Xen VM images to KVM compatible images
- Xen images contain paravirtualized Linux OSes

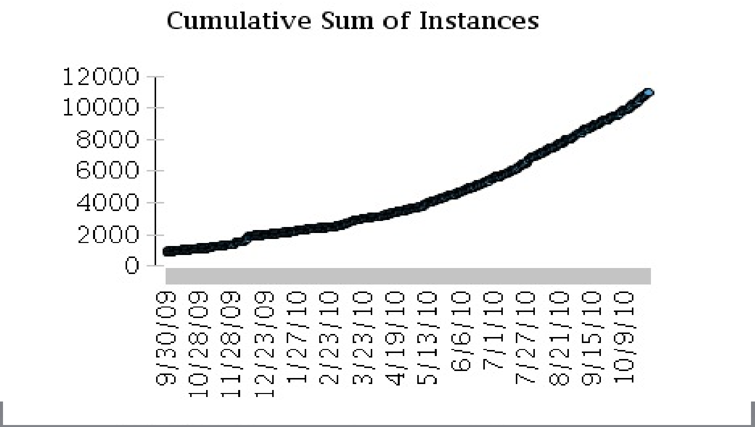
Requirements

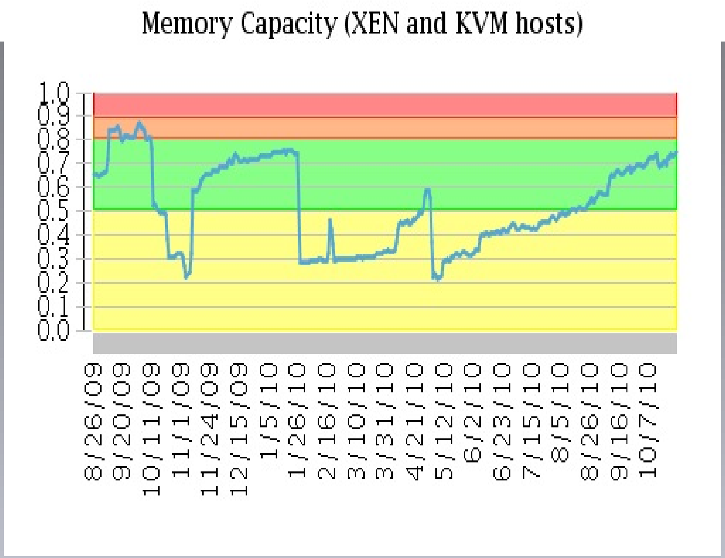
- Zero Downtime
  - RC2 production system was continuously running with all functionalities enabled and no noticeable performance slowdown.
- Efficiency (both in storage and in time)
  - Consumed another 293 GB storage (cf. 9.5TB with flat file approach)
  - Took only 20 seconds to convert an image (cf 4 minutes in native way)
- Transparency to end users
  - End users did not notice any change of their images until the "conversion" day

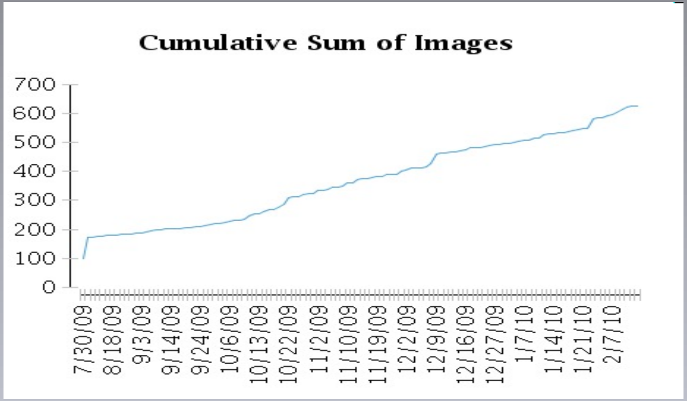
RC2 Value: Usage Growth

In 1 year of RC2 production operation

- 631 users from 34 countries
- Fast grow of VM images and instances
- Matching capacity grow required







Conclusions and Ongoing Work

- RC2
  - Delivers high-quality cloud computing for IBM research community (and beyond)
  - Provides effective framework for quick integration of novel ideas in to real cloud platform
- Ongoing
  - Extending to include at least two other RC2 zones in two different continents
  - Adding many research PaaS (Web App Platform, Elasticity Service) and SaaS (dev/test service cloud) technologies

Thanks…

“Cloud at IBM T.J. Watson Research Center”, photograph copyrighted by K Ryu, 2009