



GRNET Cloud Computing Services An Overview

Panos Louridas
louridas@grnet.gr

GN3 Innovation Workshop
Copenhagen, October 10 – 11 2011



Outline

- ◆ ~okeanos IaaS
- ◆ pithos Storage Service
- ◆ Aquarium Accounting Service

All code developed by GRNET for its cloud services is available through a BSD license.



okeanos



Motivation

- ◆ Deliver IaaS to GRNET's customers
 - ➔ direct: IT departments of connected institutions
 - ➔ indirect: university students, researchers in academia
- ◆ Other IaaS efforts
 - ➔ Amazon EC2 not an end-user service
 - ➔ Need to develop custom UI, AAI layers
 - ➔ Vendor lock-in
 - ➔ Gain know-how, build on own IaaS → new services



~okeanos IaaS

◆ Infrastructure...

- ➔ Compute (Virtual Machines)
- ➔ Network (Virtual Networks)
- ➔ Storage (Virtual Disks)

◆ ... as a Service

◆ Users manage resources over

- ➔ a simple, elegant UI, or
- ➔ a REST API, for full programmatic control

IaaS – Compute (1)

◆ Virtual Machines

➔ powered by KVM

- Linux and Windows guests, on Debian hosts
- Windows Servers customized so as to look as much close to Windows 7 as possible

➔ Google Ganeti for VM cluster management

➔ accessible by the end-user over the Web or programmatically (OpenStack Compute v1.1)



IaaS – Compute (2)

◆ User has full control over own VMs

➔ Create

- Select # CPUs, RAM, System Disk
- OS selection from pre-defined Images
- popular Linux distros (Fedora, Debian, Ubuntu)
- Windows Server 2008 R2

➔ Start, Shutdown, Reboot, Destroy

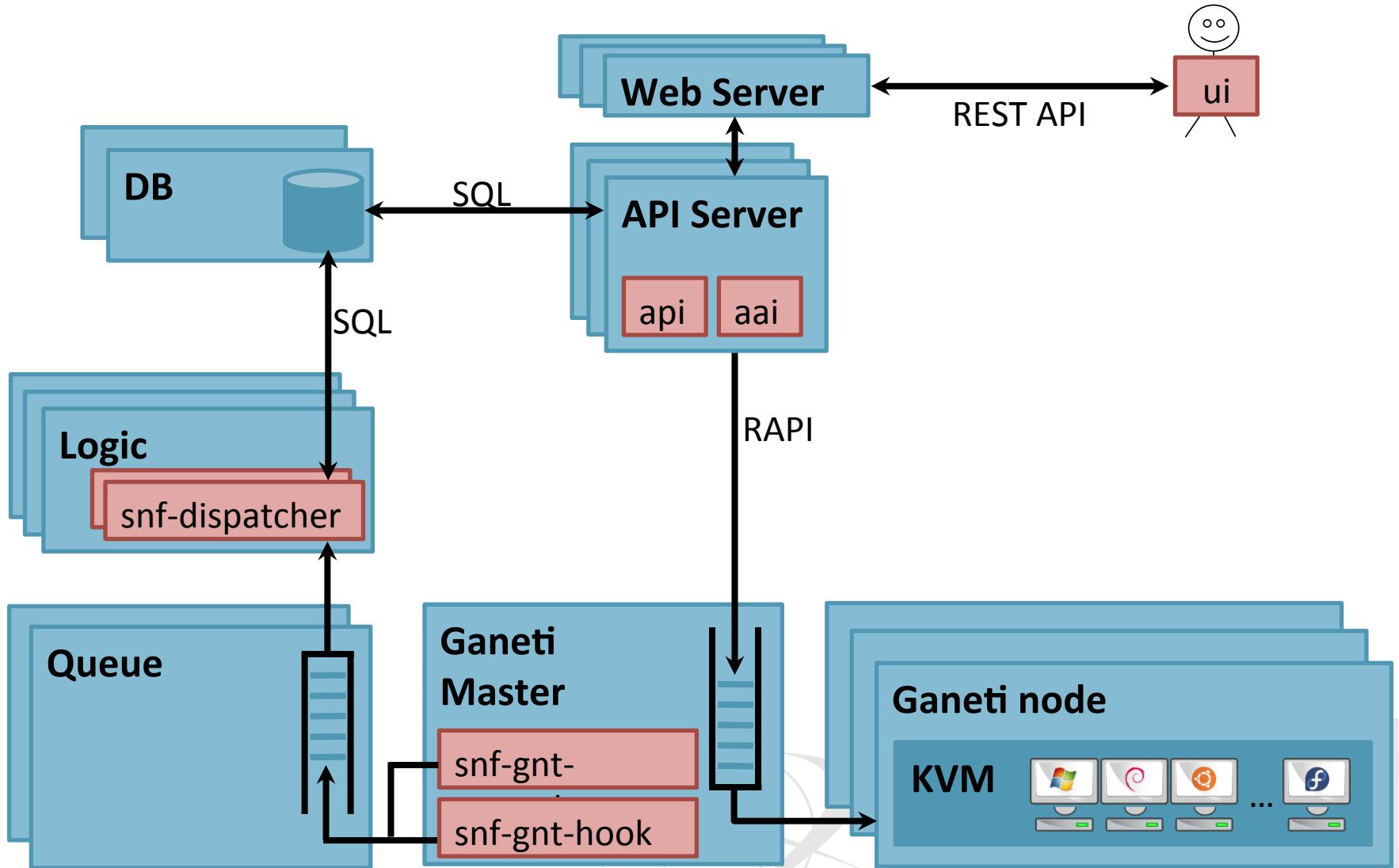
➔ Out-of-Band console over VNC for troubleshooting



IaaS – Compute (3)

- ◆ REST API for VM management
 - ➔ OpenStack Compute v1.1 compatible
 - ➔ 3rd party tools and client libraries
 - ➔ custom extensions for yet-unsupported functionality
 - ➔ Python & Django implementation; RabbitMQ for plumbing
- ◆ Full-featured UI in JS/jQuery
 - ➔ UI is just another API client
 - ➔ All UI operations happen over the API

Synnefo deployment





Motivation

◆ Evolve existing Pithos Service

- ➔ 10,000 Users
- ➔ 650,000 Files
- ➔ 3.7 Terabytes

◆ Goals

- ➔ Use standardized API
- ➔ Seamless integration with native clients
- ➔ Synchronization capabilities

Architecture

◆ Block-based Storage

- ➔ All user data split into blocks, stored using content-based addressing based on their hashes
- ➔ Metadata stored in database

◆ Extending OpenStack Object Store

- ➔ Metadata used for extra functionality
- ➔ Data sharing
- ➔ Full versioning
- ➔ Deduplication
- ➔ Instant uploads and synchronization



Implementation

- ◆ Backend in Python / Django, MySQL or Postgres
- ◆ Web-based
 - ➔ Written in GWT
- ◆ Mac OS, iOS clients
 - ➔ Native, using common code base, possible fork of OpenStack GitHub client
- ◆ MS-Windows
 - ➔ Native
- ◆ Command-Line Interface
 - ➔ Full implementation of OpenStack Object Storage API v. 1.0
- ◆ All clients use the same REST API

Outline

- ◆ ~okeanos IaaS
- ◆ Pithos Storage Service
- ◆ **Aquarium Accounting Service**
- ◆ Upcoming goals



Motivation

- ◆ Integrated accounting and “billing”
- ◆ One-stop shop for both users and administrators
- ◆ Flexibility + Control in providing resources to our users
 - ➔ Users must be able to use and share resources to the full (or resources remain idle)
 - ➔ Users shall not abuse



Architecture

- ◆ Credit-based use model
 - ➔ GRNET is the sole issuer of credit, on a periodic basis
 - ➔ Credit may be fully transferrable, hierarchically transferrable, not transferrable
- ◆ All user-related events are recorded for accounting and billing
- ◆ A comprehensive accounting service bills user depending on pricing model, checks permissions and quotas



Thank You!

Questions?

