



# ReC<sup>2</sup>S: Reliable Cloud Computing System

Alecsandru Pătrașcu – Master SCPD, An II

Scientific lead:

Prof. Dr. Ing. Valentin Cristea

As. Dr. Ing. Cătălin Leordeanu

---

# Contents

- Introduction
- Goals
- Architecture
- Implementation
- Results
- Future work
- Conclusion

# Introduction

*“**Cloud computing** is a model for enabling ubiquitous, convenient, **on-demand network access** to a **shared pool** of configurable computing resources that can be **rapidly provisioned and released** with minimal management effort or service provider interaction.”*

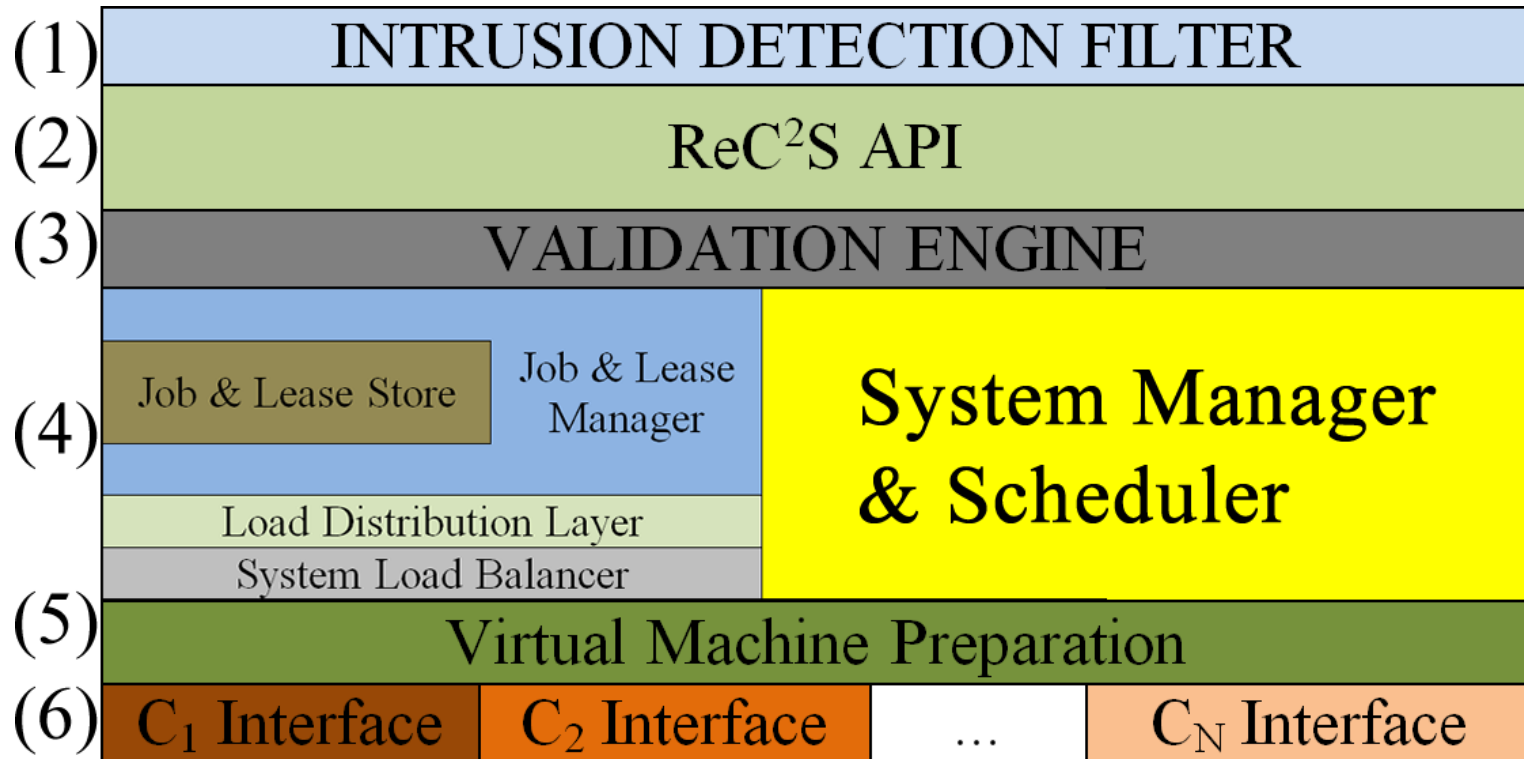
**NIST**

# Goals

- To create a unified and self-contained platform and framework for resource management
- To help users develop applications that automatically scale horizontally and vertically
- To implement custom security policies
- Seamlessly interconnect with other Clouds

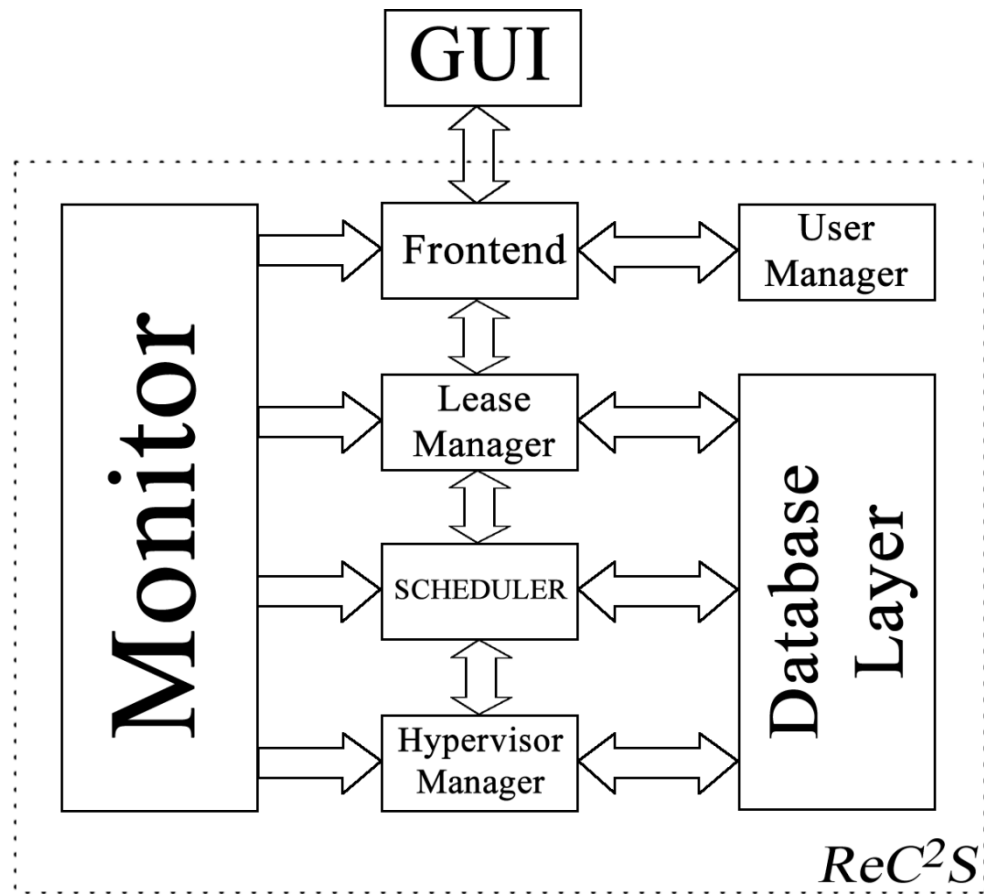
# Architecture – Top view

6 layers:



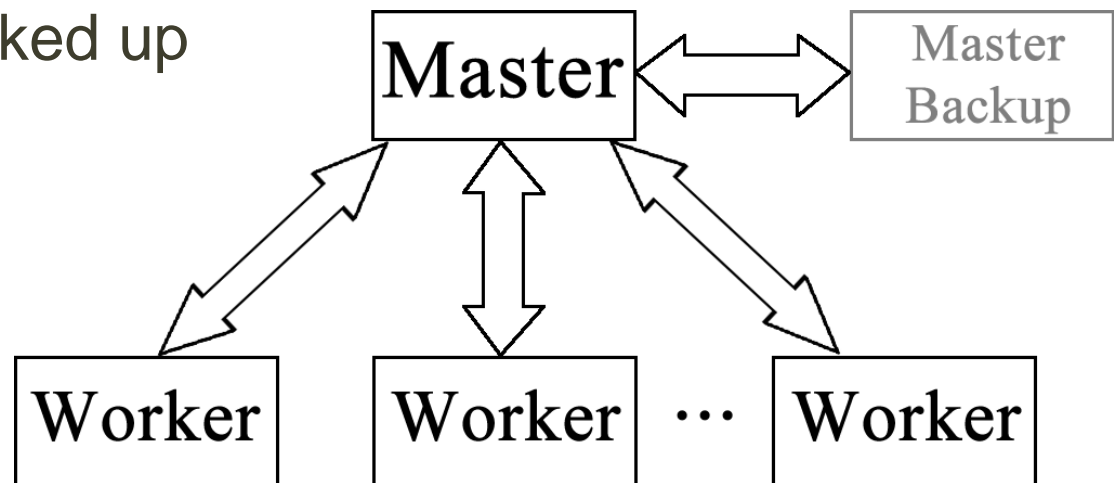
# Architecture - Implementation

- 7 modules:



# General building block

- Master-slave framework
- Process based parallelism
- One worker per CPU
- Workers are automatically restored on crash
- Master is backed up



# GUI

- Responsible for user interaction
- Web interface

[index](#) [Add lease](#) [Lease status](#)

Virtual Machine details

Minimum instance count 1 Maximum instance count 5

Hardware configuration

processorArchitecture x86\_64  
processorVendor Intel  
processorSpeed 2000 Mhz  
numberOfCores 2  
memorySize 1024 MB  
storageCapacity 5 GB  
networkBandwidth 1000

Software configuration

templateName Ubuntu Server 10.04 + MySQL

Preemptible ☒

Time

Start 04/11/2012 @ 10:22  ☐ Right now

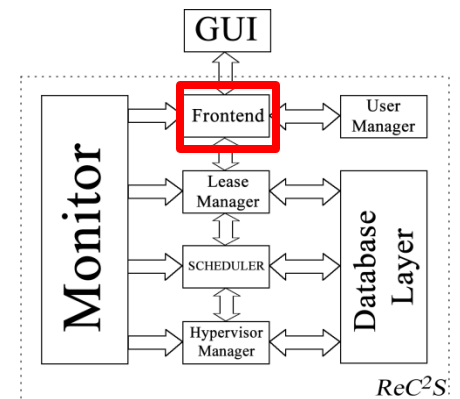
Stop 04/12/2012 @ 14:41  ☐ Infinite

Alocate



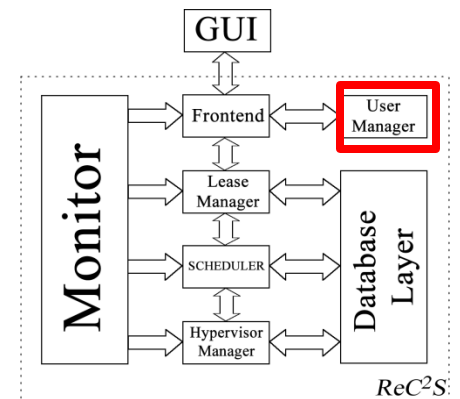
# Frontend

- Maps over the “Intrusion detection filter” and “ReC<sup>2</sup>S API”
- Receives authentication requests from GUI
- Legit requests are passed to the “User Manager”



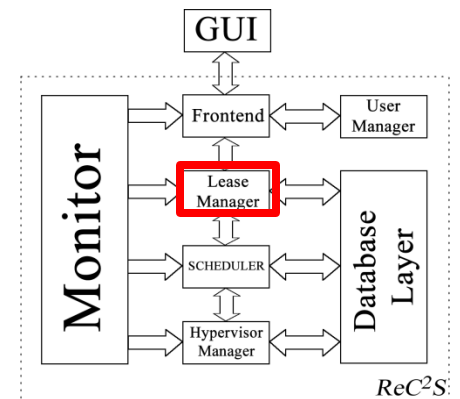
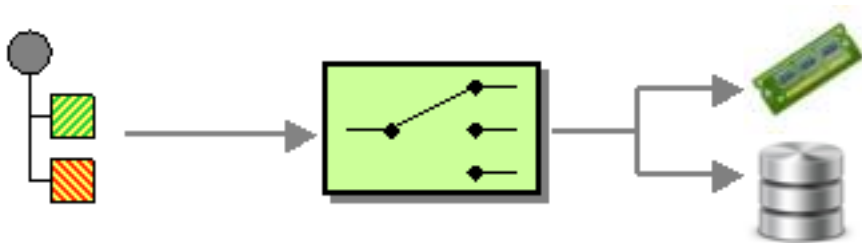
# User Manager

- Maps over the “Validation engine”
- Authentication and lease validation
- Stores details about the user
  - Username
  - Password
  - etc



# Lease Manager

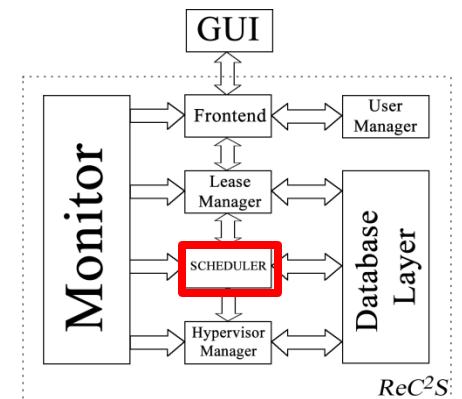
- Maps over the “Job and Lease Manager” layer
- Save the leases into *jobs*
- Plug-and-play store engine
- Automatic lease backup and restore



# Scheduler

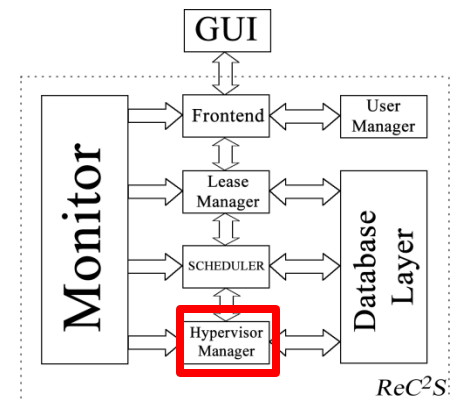
- Maps over the “Scheduler & System manager” layer
- Checks for lease resource availability
- Plans leases to run on the attached Cloud interfaces
- It runs the *ReC2Sched* algorithm which proved to be about **35%** faster than FIFO/SJF

A. Pătraşcu, C. Leordeanu, C. Dobre and V. Cristea,  
“*ReC<sup>2</sup>S: Reliable Cloud Computing System*”,  
European Concurrent Engineering Conference,  
Bucharest, April 2012



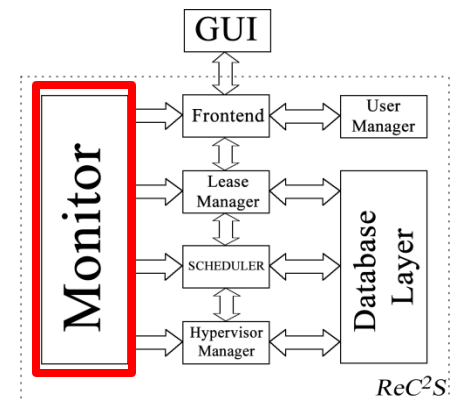
# Hypervisor Manager

- Maps over the “Virtual machine preparation” and “Cloud specific interface” layers
- Receives requests to start/check a new lease
- Plug-and-play hypervisor engine
  - VMware ESX
  - VirtualBox
  - OpenNebula
  - Eucalyptus
  - Amazon EC2



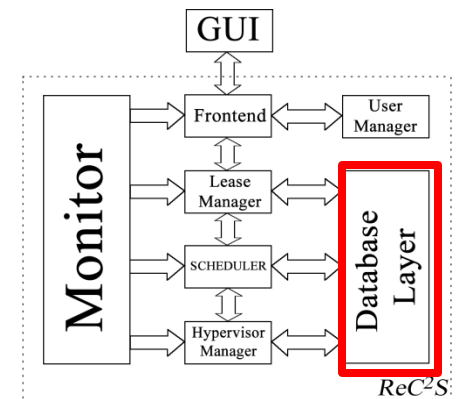
# Monitor

- Monitors the entire activity of the system
- Decides when to:
  - Preempt a lease
  - Increase the number of VMs from the lease
  - Decrease the number of VMs from the lease



# Database Layer

- DB management for the entire system
- Plug-and-play DB engine



# Implementation

Element	Programming Language
Backend modules	Server-side JavaScript (Node.JS)
Frontend module	Grails
Hypervisor drivers	Java



# Results

- Lease Manager

#	Lease creation time (ms)	Lease check-up (ms)	Lease store (ms)
1	204	10	1
2	205	11	1
3	289	11	1
4	208	10	1
5	262	10	1

# Results

- Hypervisor manager

#	Lease retrieve (ms)	Lease resource check (ms)
1	3	25
2	3	28
3	2	51
4	2	26
5	3	39

# Future work

- Further optimize the whole framework
- Security improvements
  - Trust management
  - Information security and privacy (homomorphic encryption schemes)
- Integration with other Cloud infrastructures

# Conclusions

- We presented a reliable framework for cloud infrastructures
- We provide performance and security
- We provide compatibility with different cloud systems



**Thank You!**