
Auditing as a Service Part 3

CS528 - Cloud Computing

— Mentors: Sahil Tikale - Naved Ansari —

Ali Raza - Rushi Patel - Chenxi Li - Kevin Liang

Background

- **Bare Metal Clouds**

- Bare metal machines instead of virtual machines
- Current bare metal clouds also do provisioning
- Tenant does not have the flexibility
- Each tenants may have different requirements
- How do we ensure different tenants can share the cloud?

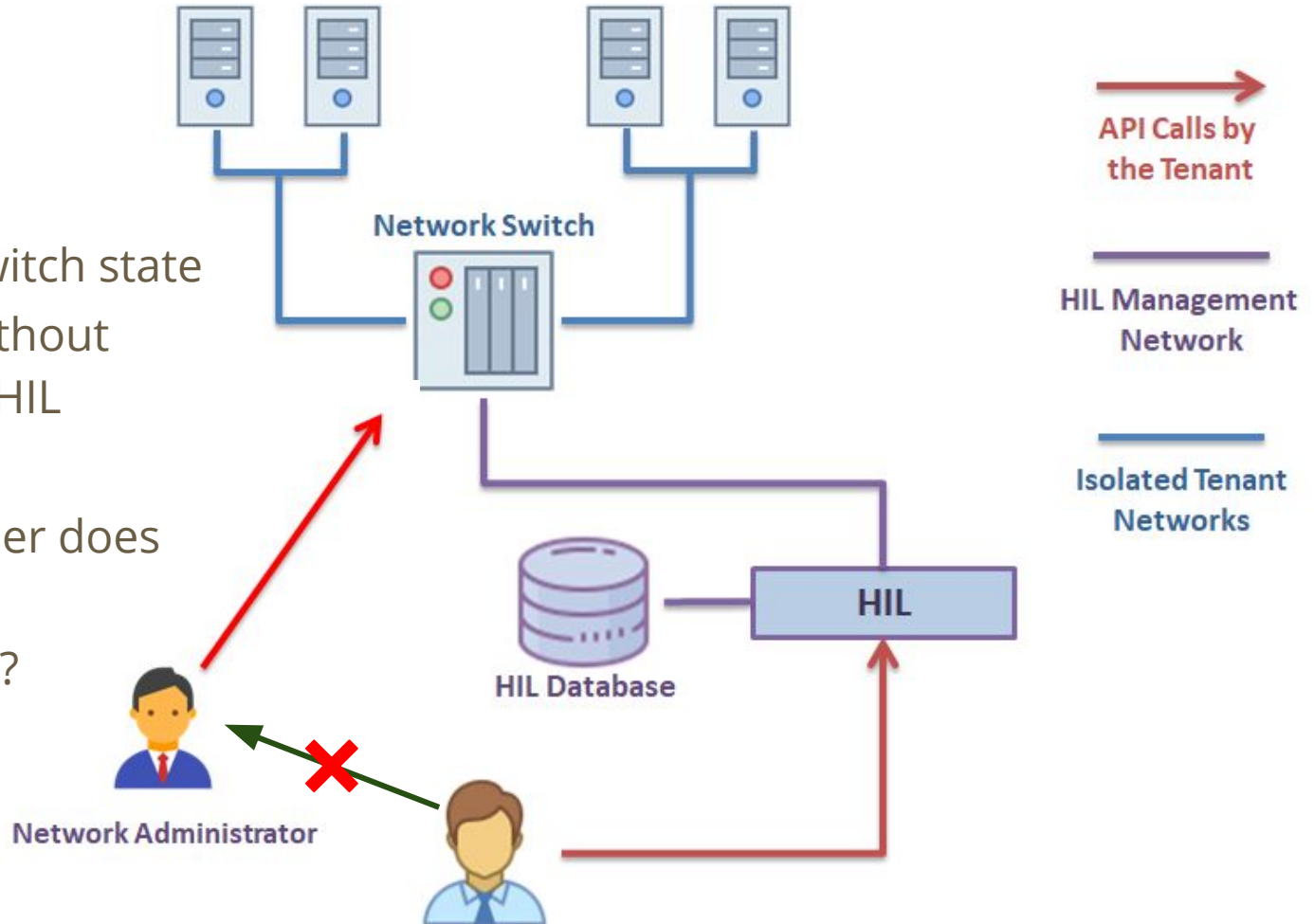
Background

- **Hardware Isolation Layer (HIL)***
 - Provides isolated network of nodes
 - Exokernel for the cloud
 - Provides as little abstractions as possible
 - Allows tenants to choose provisioning system and other services
 - Move resources between multiple clusters
 - Allows security sensitive applications to use public clouds

* Hennessey, Jason, et al. "HIL: designing an exokernel for the data center." *Proceedings of the Seventh ACM Symposium on Cloud Computing*. ACM, 2016.

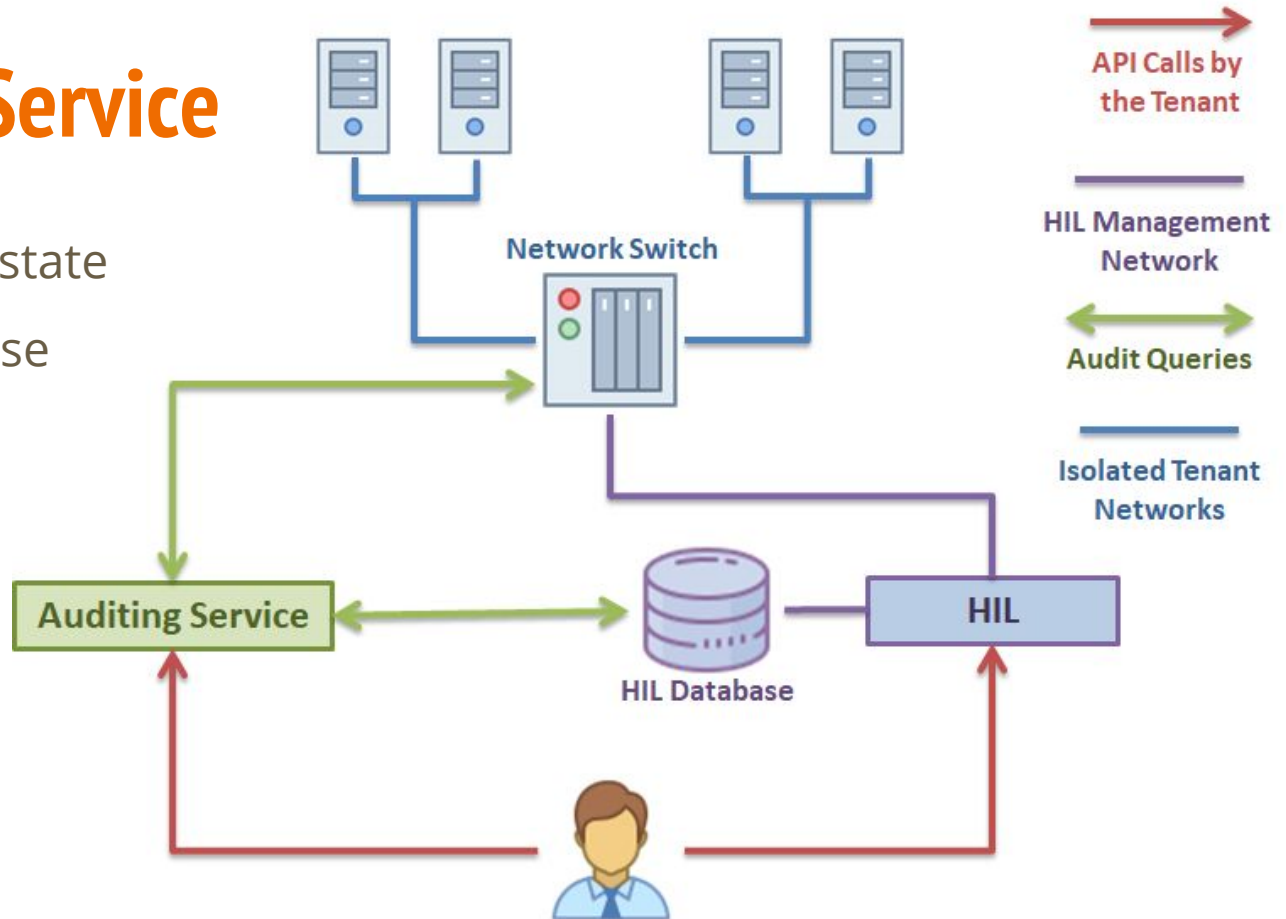
Motivation

- What if the switch state is changed without updating the HIL database?
- What if the user does not trust the administrator?



Auditing as a Service



- Query the switch state
- Query HIL database
- Check and report discrepancies



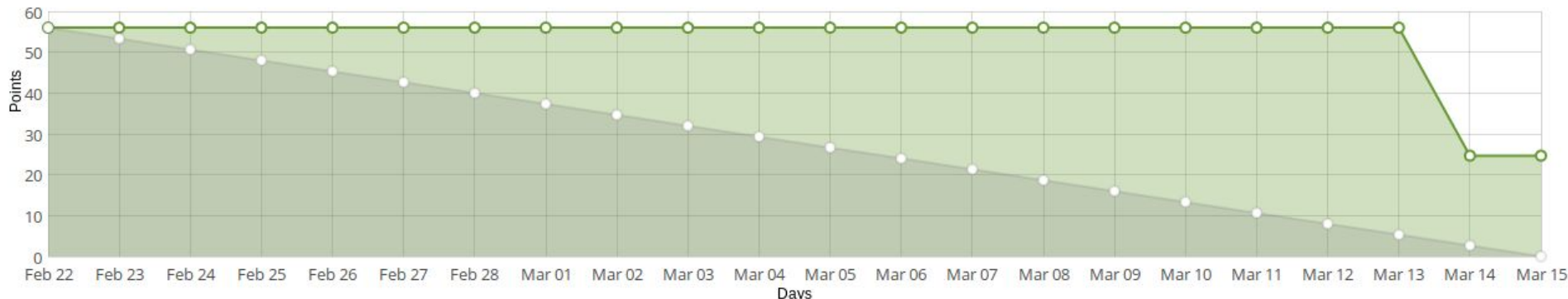
Current Epic

- Epic 1

- Minimum Viable Product
- Current state of HIL and network switches - Simple Yes or No at the start

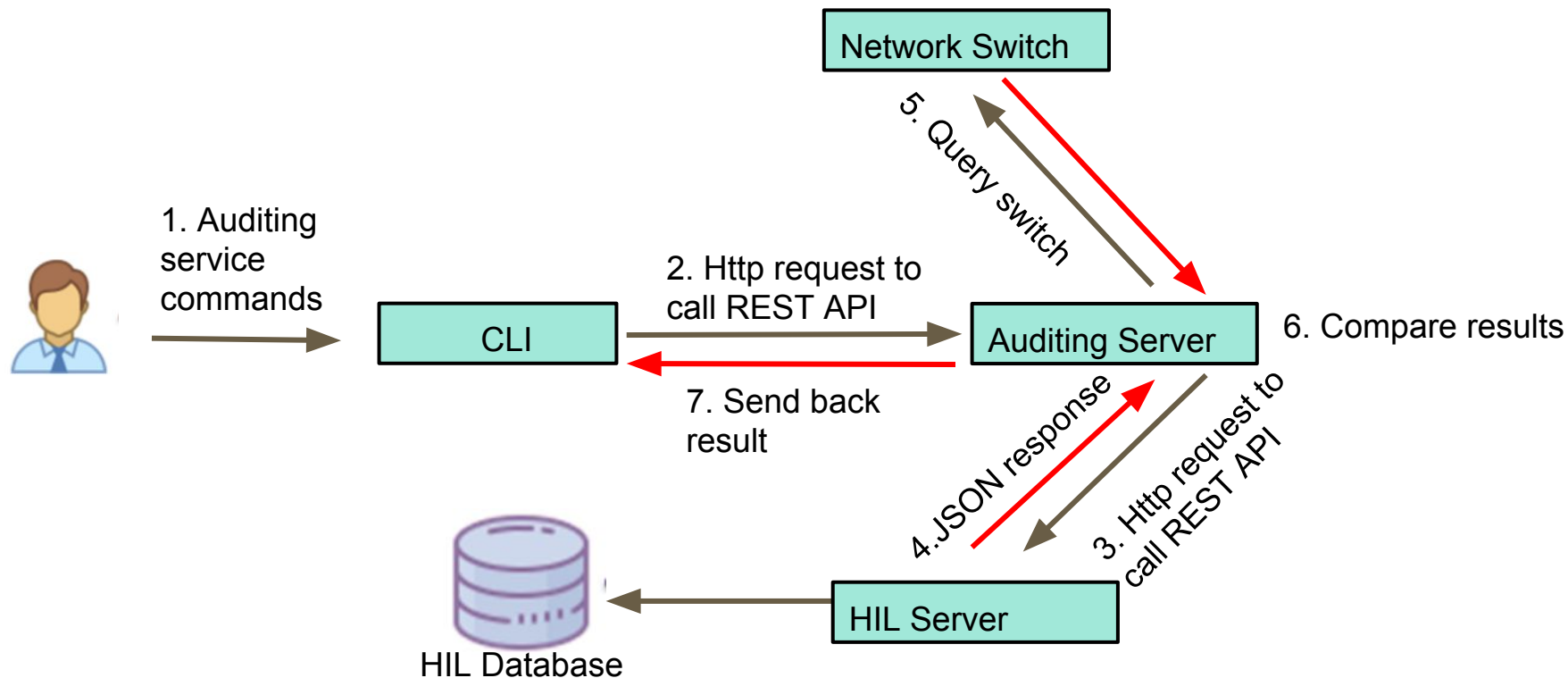
Votes	Name	Project	Sprint	Assigned	Status	Progress	View options ▾
▲ 0	#41 Audit system returns yes or no (Current State) EPIC ^				New ▾	<div><div></div></div>	
▲ 0	#50 Audit system history and automated EPIC ^				New ▾	<div><div></div></div>	

Third Sprint - MVP



- Create Backend audit system - Part 2
 - Simple calls to HIL database (Rushi)
 - Abstract class for switches (Ali)
- Connect CLI with REST (Kevin)
- Connect REST with Backend (Mark/Rushi)

Auditing Service Workflow



Demo

CLI Functions

1. list_nodes_for_project

parameter *project_id*
return *array<node>*

2. get_port_for_node

parameter *node_id*
return *port*

3. list_projects_for_node

parameter *node_id*
return *array<project>*

4. list_projects_for_switch

parameter *switch_id*
return *array<project>*

5. list_nodes_for_switch

parameter *switch_id*
return *array<node>*

6. check_project_info_consistency

parameter *project_id*
return *BOOL*

Next Sprint - MVP with additional functions

User Stories:

- Switch communication still needs work
 - Want to get a robust system working with Open vSwitch
 - Also want to adopt other switches present in MOC
- Implement new functions and better audit design based on user stories
 - Mentors as our customer, we want to include all functionality that must be done in our MVP before moving to the second Epic
- Begin thinking about changes needed for second Epic
 - Design new workflow of system
 - Think about changes for always running system with database

Thank you!