

# Distributed Monitoring Solutions for the Seattle Testbed

Hieu Sy Do, Albert Rafetseder, Justin Cappos

## Introduction

### ➤ Seattle Testbed:

- an open-source, peer-to-peer cloud platform
- operates on shared resources donated by users worldwide
- provides a sandboxed environment (vessel)

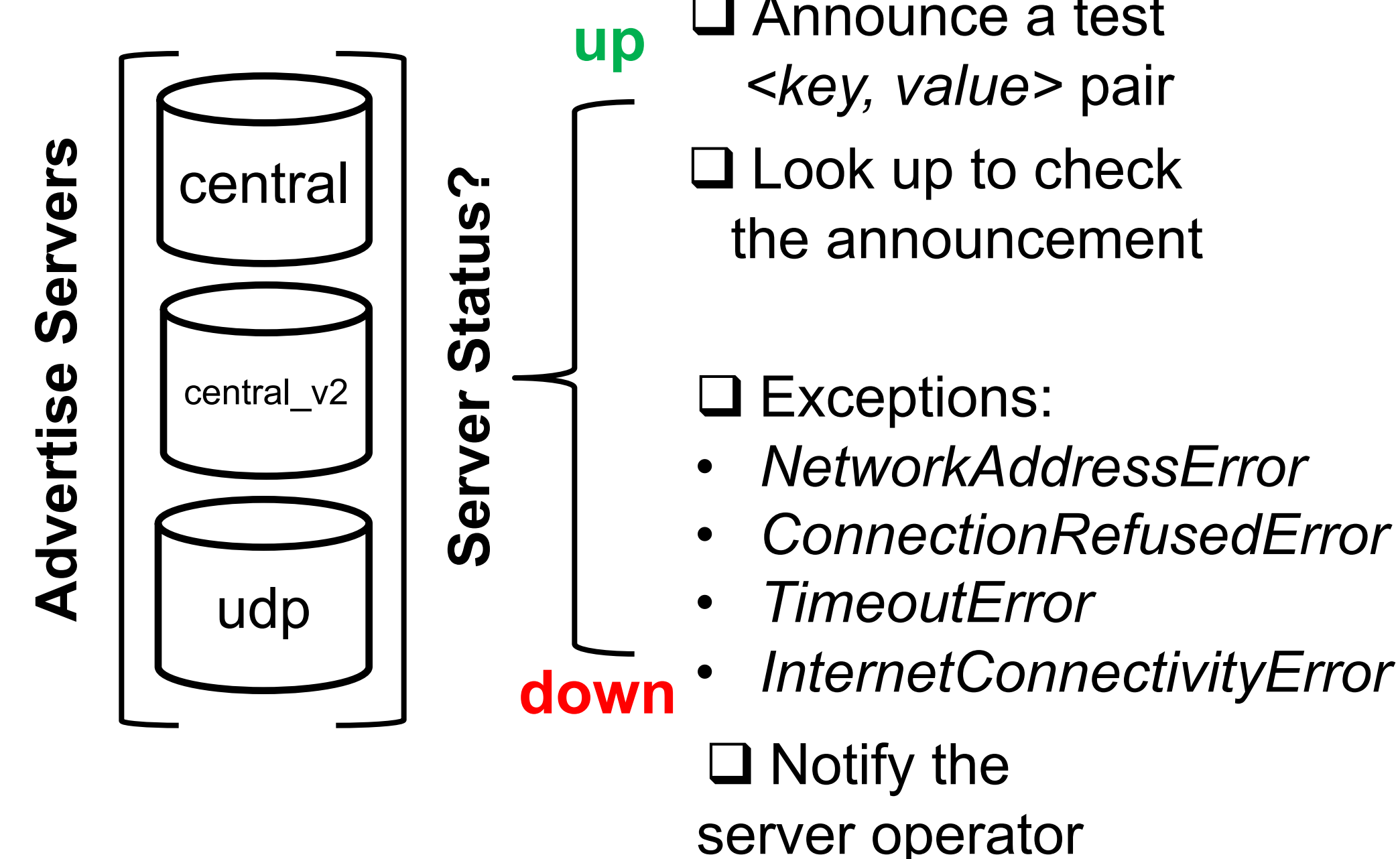
### ➤ Monitoring Seattle:

- ❑ Targets to be monitored:
  - Seattle Clearinghouse: keeps track and propagates the donated resources.
  - Custom Installer Builder: allows the users to customize the configuration of vessels
  - Software Updater: dispatches the latest version
- ❑ Design criteria:
  - Metrics: availability, performance, usage, etc.
  - Location to run: local machine, node manager, servers, etc.
  - Users: operators, contributors, public users
  - Notification system

→ Easy Access: public users can run the tool

## Implementation

### ➤ Advertise Service:

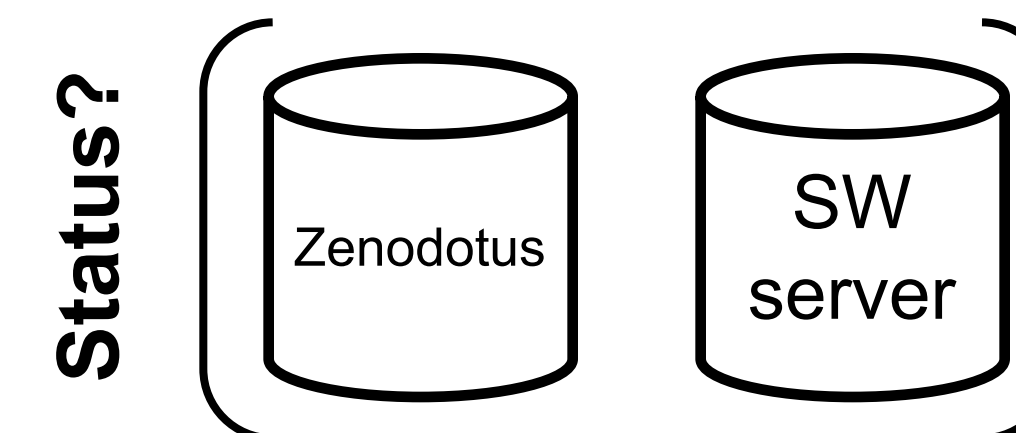


### ➤ Software Updating site:

- ❑ <https://seattle.poly.edu/updatesite/>

### ➤ Zenodotus

- ❑ Advertise service: multiples nodes' IP addresses registered to a single hostname



### ➤ Time: NTP, TCP query → returned time v. local time

### ➤ Custom Installer

#### Builder site:

<https://sensibilityclearinghouse.poly.edu/custominstallerbuilder/>

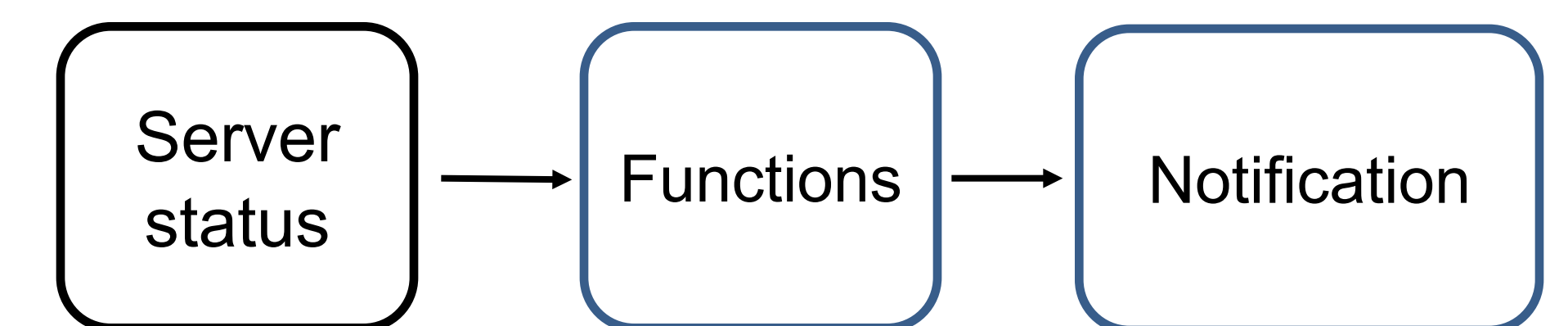
### ➤ Seattle Clearinghouse site:

<https://seattleclearinghouse.poly.edu/>

## Conclusion & Further Work

### ➤ Conclusion:

- ❑ More access to the monitoring tools
  - less control, but more scalability
  - good for Seattle platform (over 20,000 computers worldwide)
- ❑ Principle of Separating Concerns:
  - Monitoring server status
  - Monitoring functions
  - Notification



Modular codes for each process

→ reuse any process for other purposes.

Ex: notification for integration test framework

### ➤ Further work:

- ❑ Monitoring dashboard: interactive & real-time