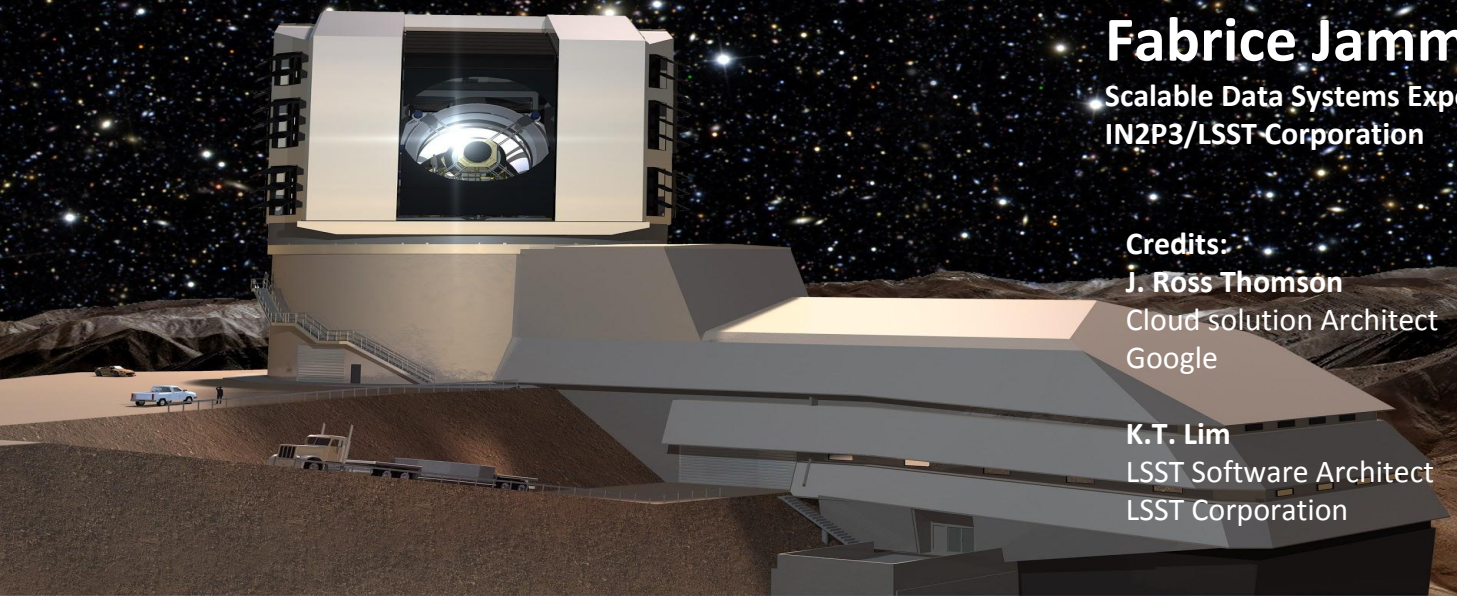
KubeCon CloudNativeCon Europe 2019

Large Synoptic Survey Telescope
From Cloud-Native to Dark Energy



Fabrice Jammes
Scalable Data Systems Expert
IN2P3/LSST Corporation

Credits:
J. Ross Thomson
Cloud solution Architect
Google

K.T. Lim
LSST Software Architect
LSST Corporation

Agenda

1. Large Synoptic Survey Telescope
2. Qserv: LSST Petascale database
3. Benefits of Cloud-Native
4. On-premise vs Public Cloud

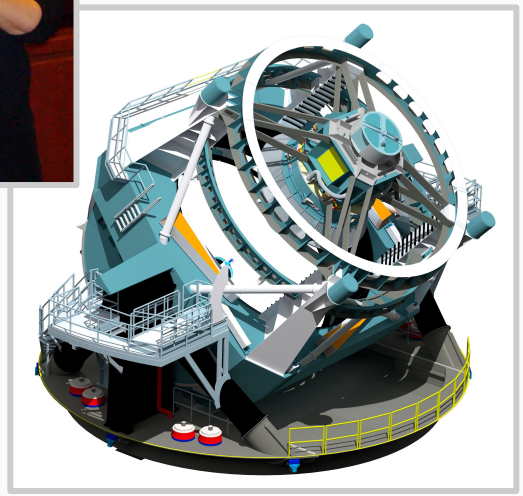
LSST in short

Large Synoptic Survey Telescope

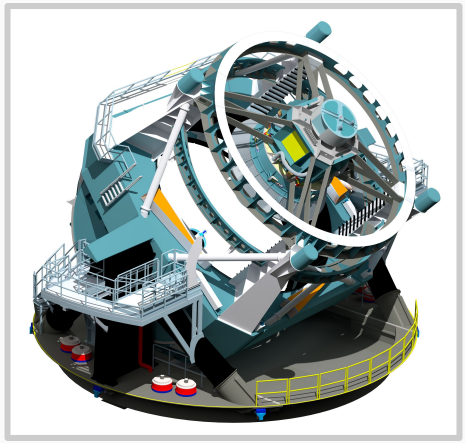
Large aperture, wide-field, ground-based survey telescope
The largest imager ever built for astronomy

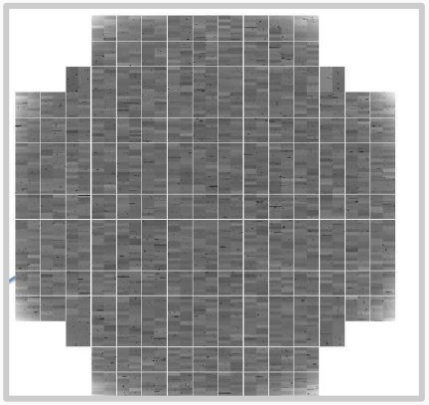
Characteristics

- ★ All visible sky in 6 bands
- ★ ~20000□
- ★ 15 seconds exposures, 1 visit/3 days
- ★ During 10 years!
- ★ **60 PB of raw data**

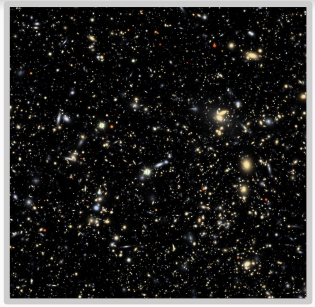


80+ PB of astronomical catalog

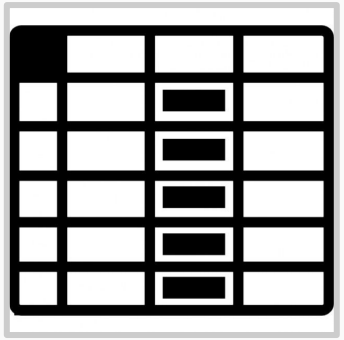




Raw data



Processed image

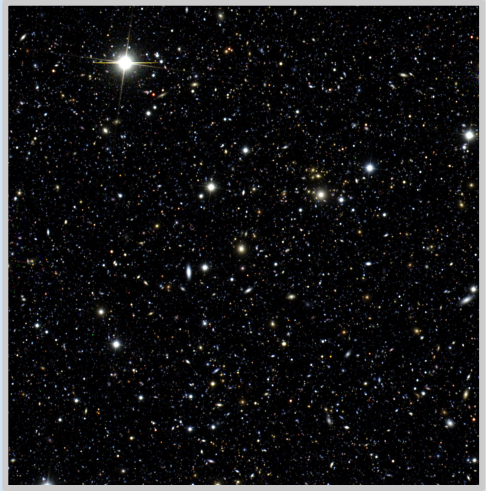


Catalog (stars, galaxies, objects, sources,transients, exposures, etc.)

LSST will build a catalog of 20 billion galaxies and 17 billion stars and their associated physical properties

Data

Images
Persisted: **~38 PB**
Temporary: **~½ EB**



- ★ ~3 million “visits”
- ★ ~47 billion“objects”
- ★ ~9 trillion “detections”

- ★ Largest table: **~5 PB**
- ★ Tallest table: ~50 trillion rows
- ★ Total (all data releases, compressed): **~83 PB**

Ad-hoc user-generated data
Rich provenance

Qserv The LSST Petascale database

Who we are

Database and Data access team

- ★ 10 engineers at Stanford University + 1 IN2P3
  - *Software development*

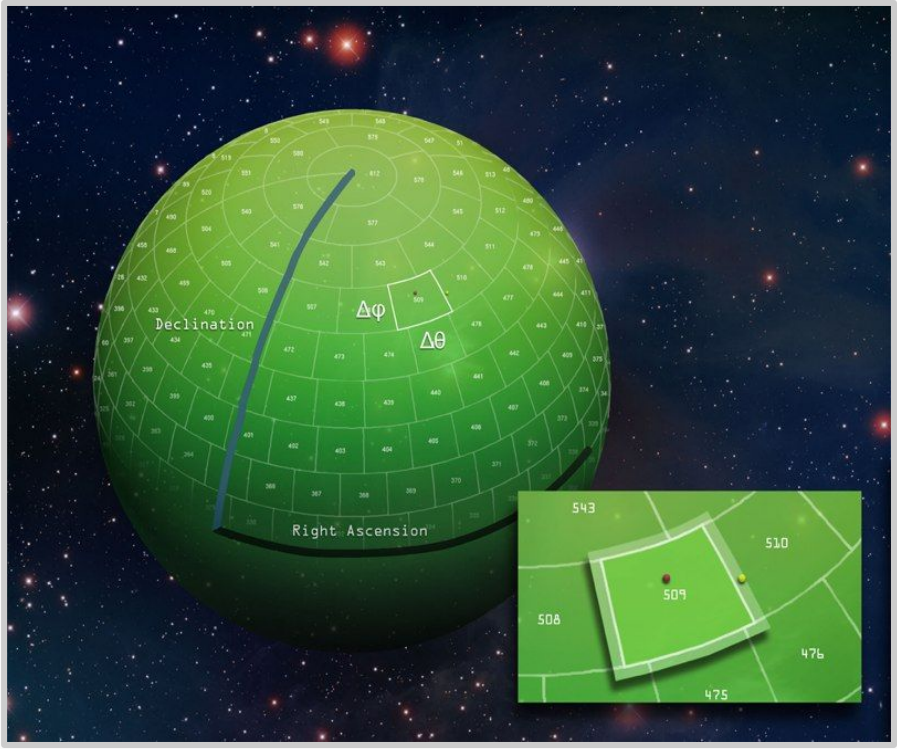
Operations teams

- ★ 5 sysadmins at NCSA/IN2P3
  - *Large Scale development platform*
  - *Prototype Data Access Center*
  - *Cloud Native / Kubernetes*
  - *System administration, Monitoring*

What we do

Data Access and **Database**

- ★ Data and metadata
- ★ Images and databases
- ★ Persisting and querying
- ★ For pipelines and users
- ★ Real time Alert Production and annual Data Release Production
- ★ For Archive Center and all Data Access Centers
- ★ For USA, France and international partners
- ★ Persisted and virtual data
- ★ **Estimating, designing, prototyping, building, and productizing**

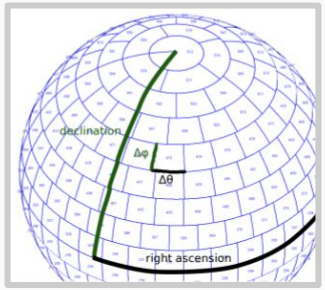


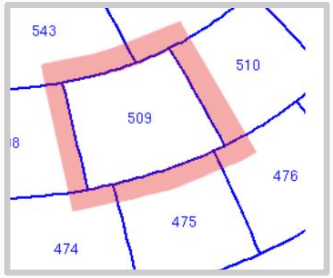
Qserv design

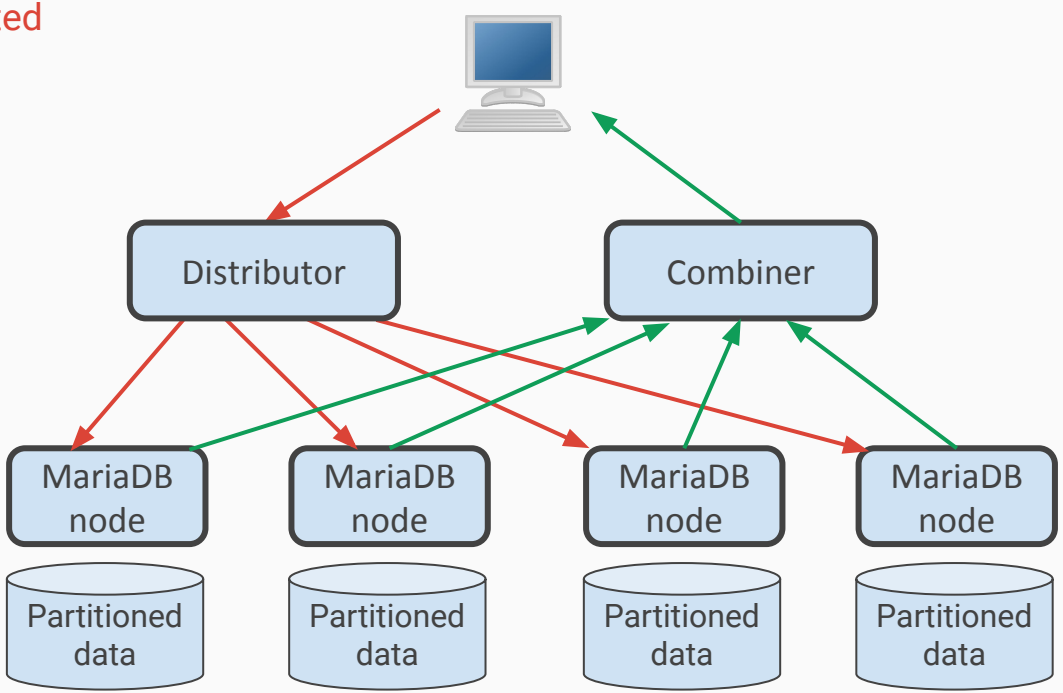
Relational database, 100% open source

Spatially-sharded with overlaps

Map/reduce-like processing, highly distributed







From Cloud-Native to Bare-Metal

Target for production

~500 nodes clusters in 2 international Academic data-centers

Running now

Development platform (CC-IN2P3)

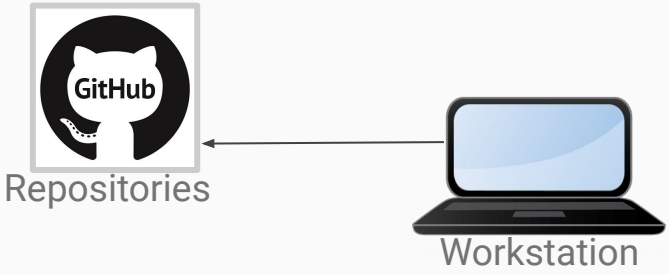
400 cores, 800 GB memory
500 TB storage (upgrade in progress)
=> +250 TB of synthesized data

Prototype Data Access Center (NCSA)

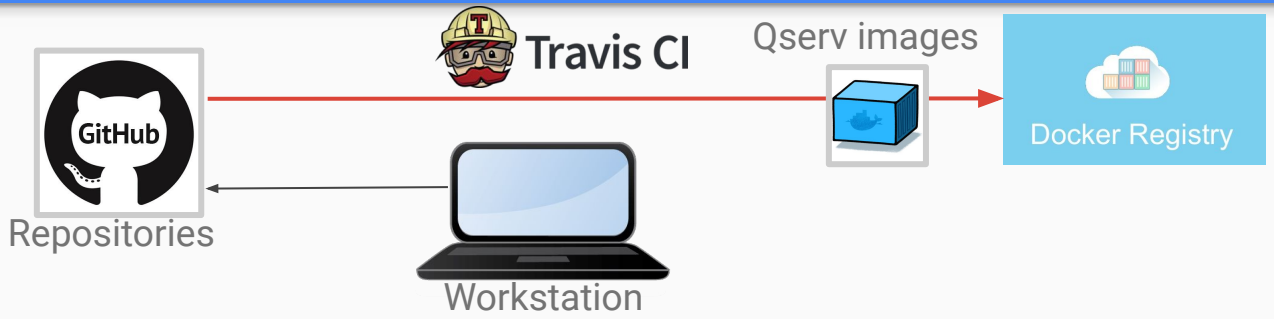
500 cores, 4 TB memory
700 TB storage,
=> WISE catalog ("real" dataset)

Benefits of Cloud-Native

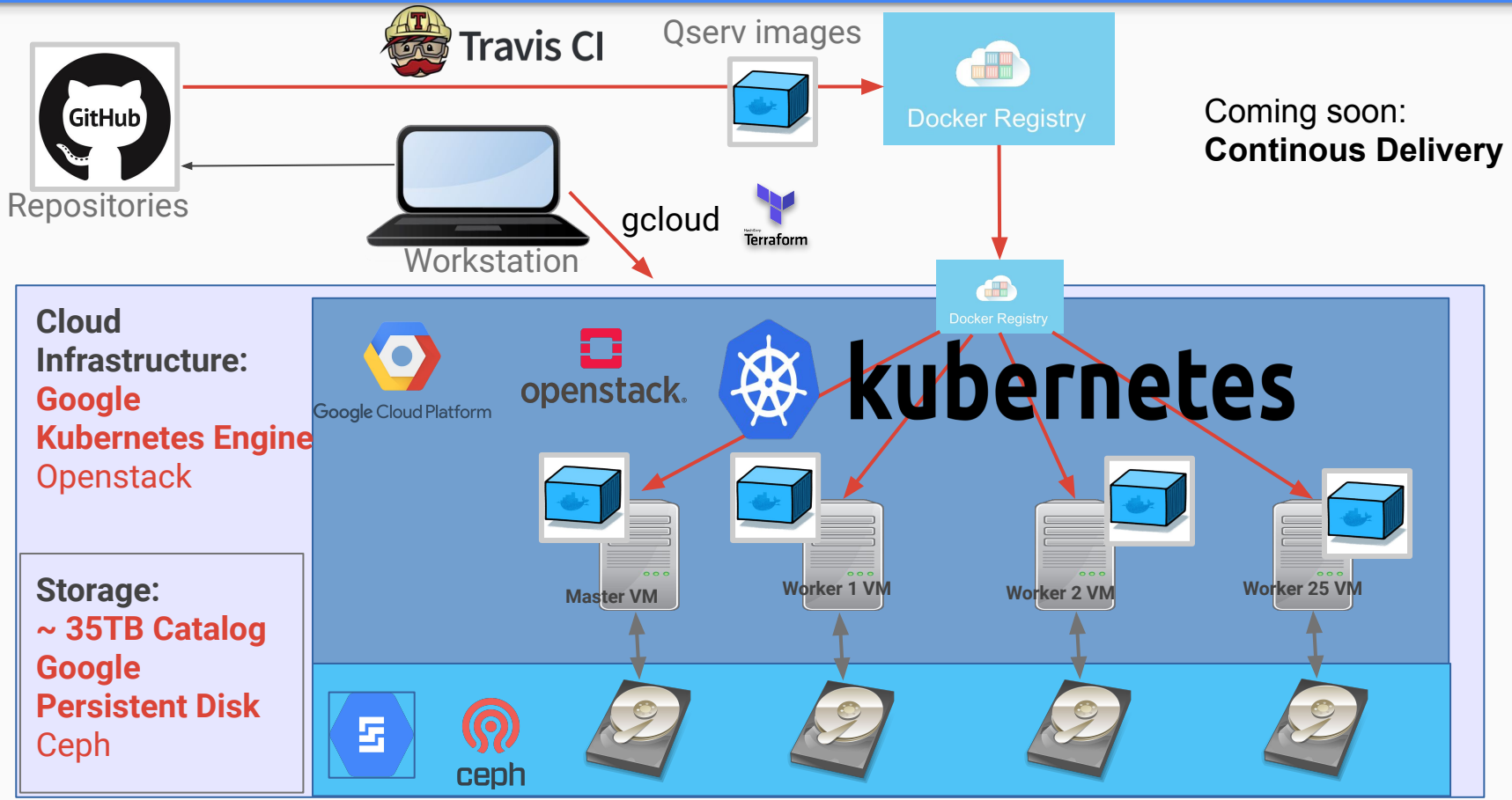
Automated Qserv deployment



Automated Qserv deployment

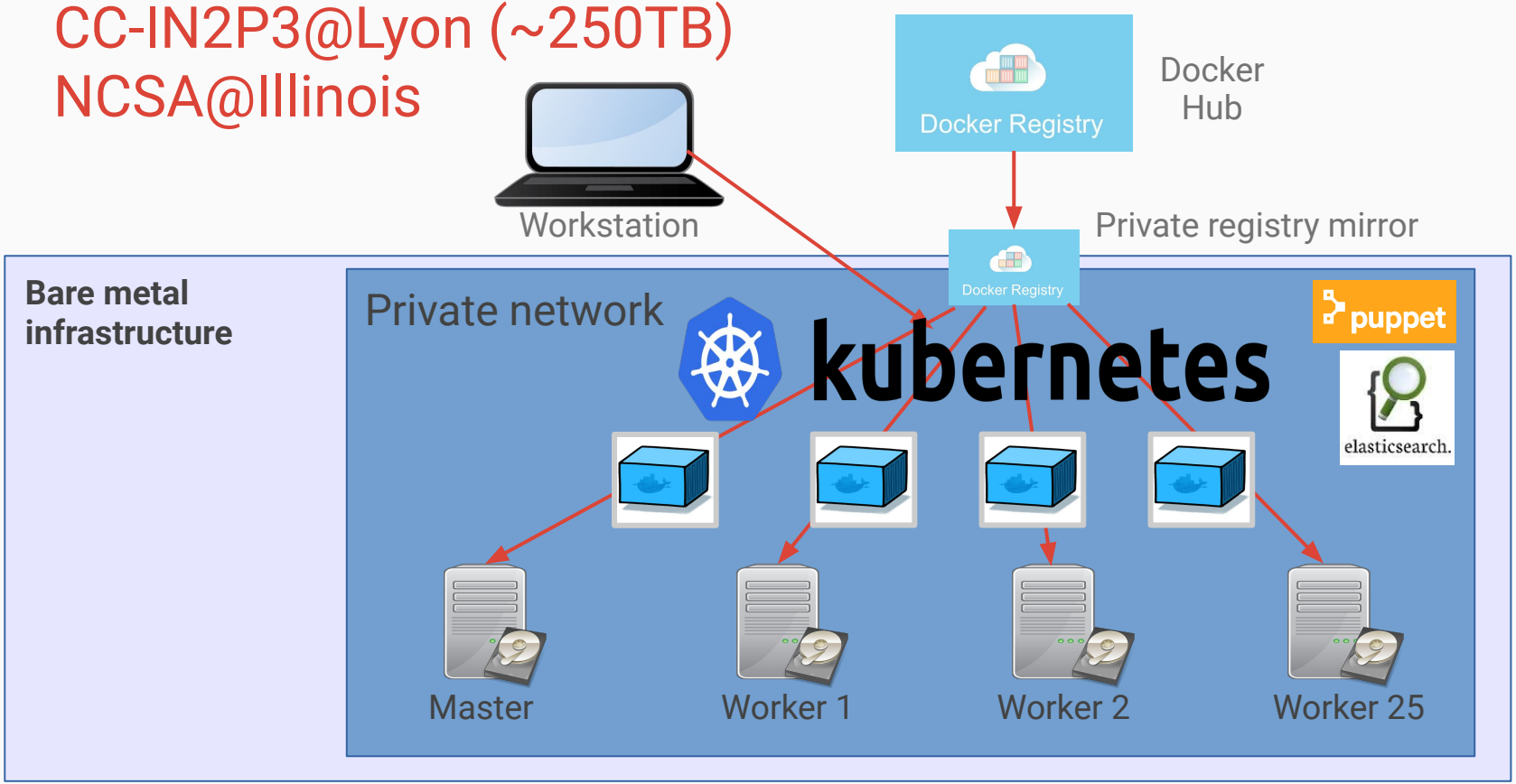


Automated deployment: Cloud Native

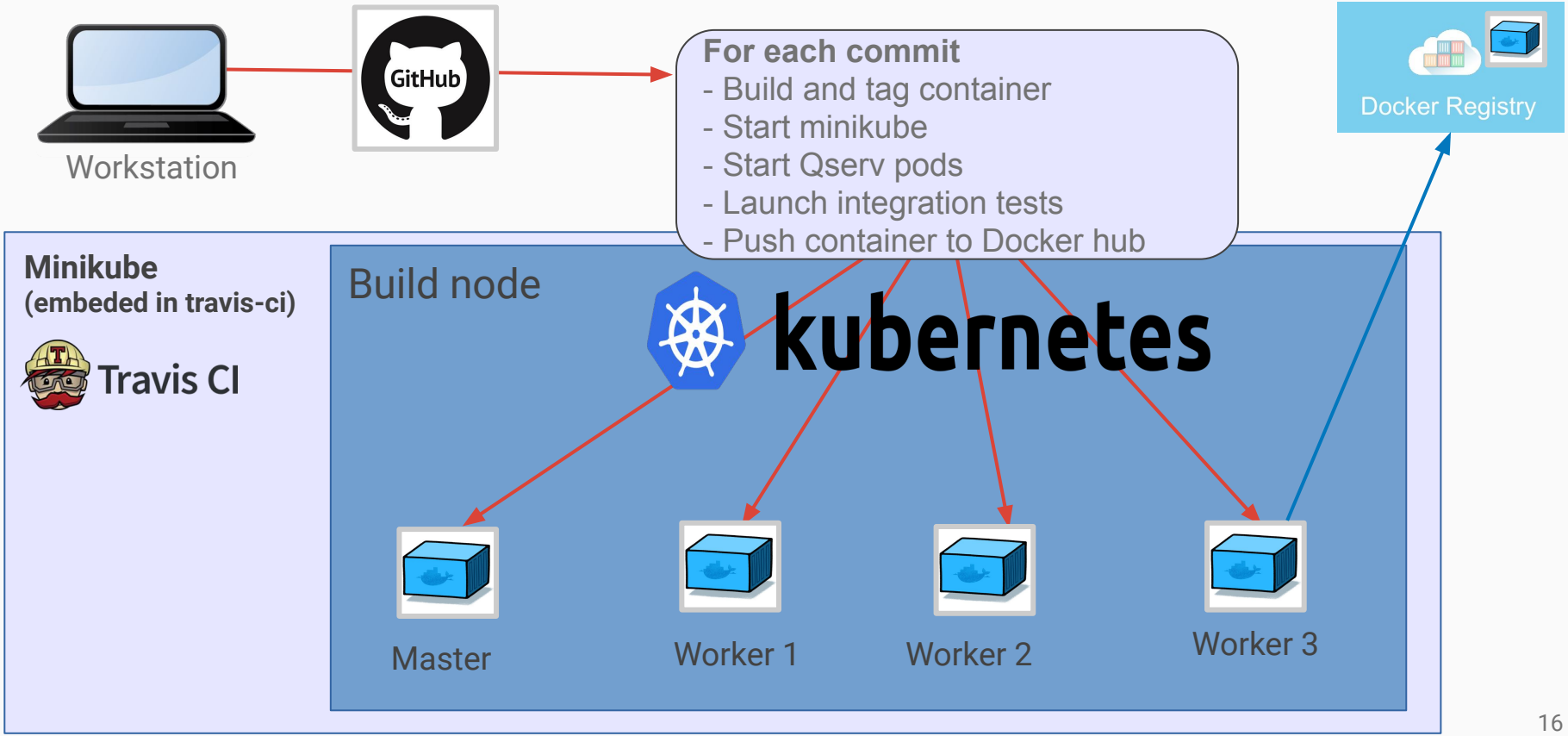


Automated deployment: bare-metal

CC-IN2P3@Lyon (~250TB)
NCSA@Illinois

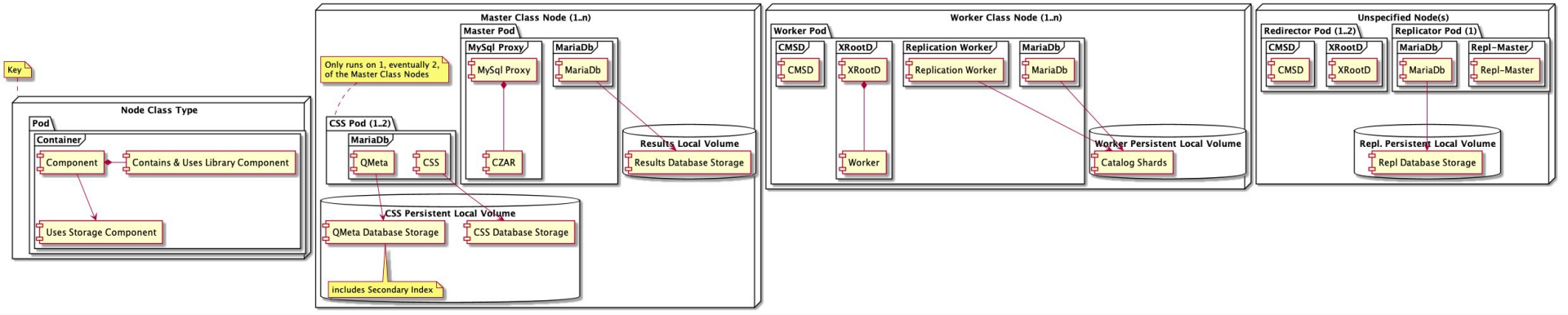


Automated deployment: CI



K8s + Microservice features

- ★ Automated scaling
- ★ Container scheduling
- ★ Auto-healing
- ★ Continuous deployment
- ★ Volume management (storage)
- ★ Easy monitoring
- ★ Healthcheck
- ★ Security

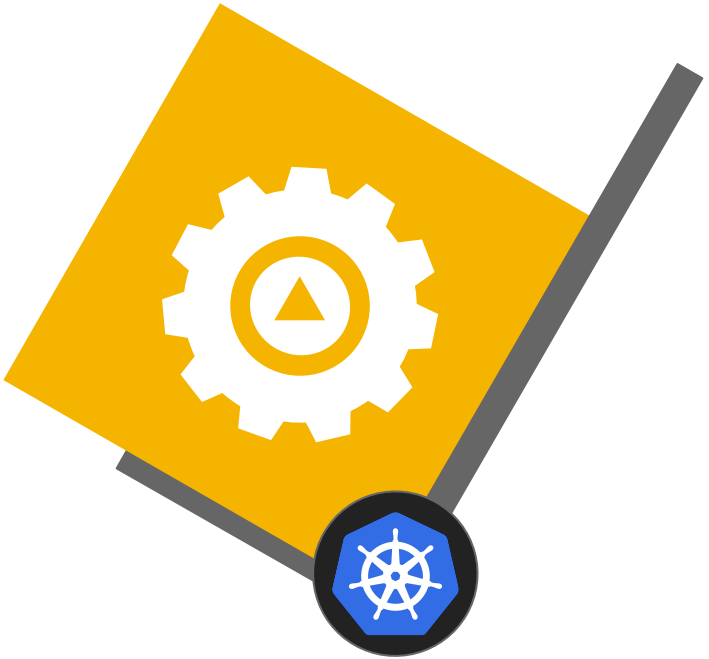


The killer feature: workload portability

Result: Portability

Put your app on wheels and move it whenever and wherever you need

Easily move your distributed application anywhere Kubernetes is supported, in seconds.

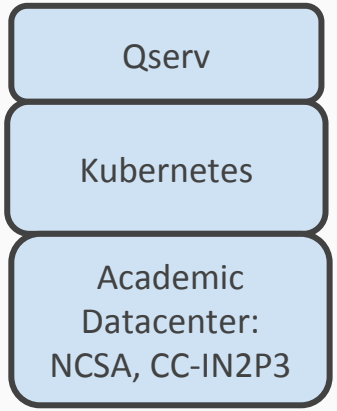


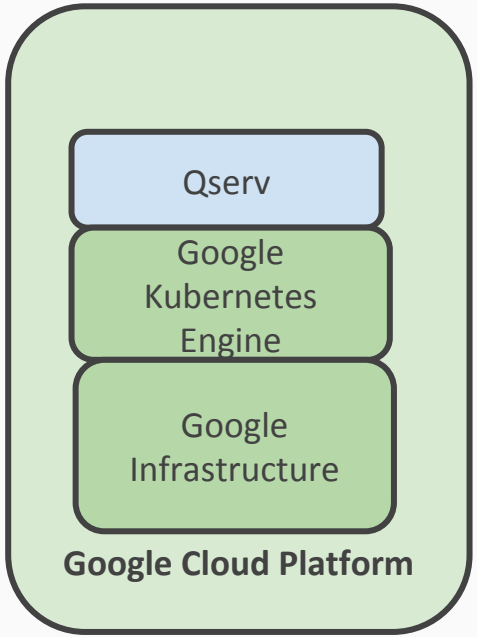
On-premise
vs
Public Cloud

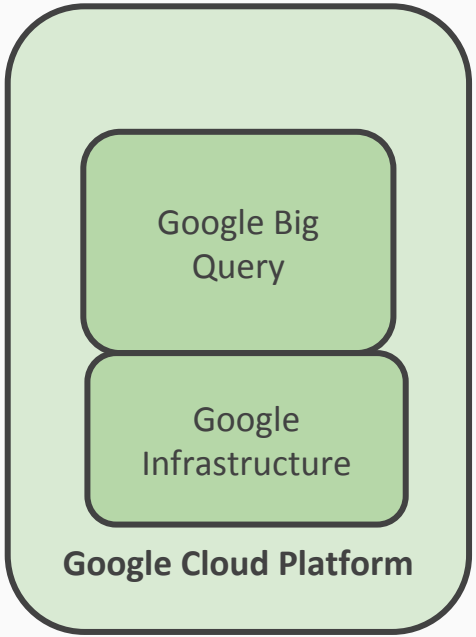
Qserv deployment schemes

Legend:

Managed by Google
Managed by LSST







GKE Overview

Containers at Google

Each week, Google launches more than four billion containers across its data centers around the world. These containers house the full range of applications Google runs, including user-facing applications such as Search, Gmail, and YouTube.

Kubernetes was directly inspired by Google's cluster manager, internally known as Borg. Borg allows Google to direct hundreds of thousands of software tasks across vast clusters of machines numbering in the tens of thousands — supporting seven businesses with over one billion users each. Borg and Kubernetes are the culmination of Google's experience deploying resilient applications at scale.

But getting started can be challenging

Kubernetes The Hard Way

This tutorial walks you through setting up Kubernetes the hard way. This guide is not for people looking for a fully automated command to bring up a Kubernetes cluster. If that's you then check out Google Kubernetes Engine, or the Getting Started Guides.

- Prerequisites
- Installing the Client Tools
- Provisioning Compute Resources
- Provisioning the CA and Generating TLS Certificates
- Generating Kubernetes Configuration Files for Authentication
- Generating the Data Encryption Config and Key
- Bootstrapping the etcd Cluster
- Bootstrapping the Kubernetes Control Plane
- Bootstrapping the Kubernetes Worker Nodes
- Configuring kubectl for Remote Access
- Provisioning Pod Network Routes
- Deploying the DNS Cluster Add-on
- Smoke Test
- Cleaning Up

And don't forget "Day 2" ops

- Managing components
- Encrypting and securing etcd
- Configuring HA
- Rolling out security patches
- Backups and disaster recovery
- Bootstrapping TLS
- Managing users and policies

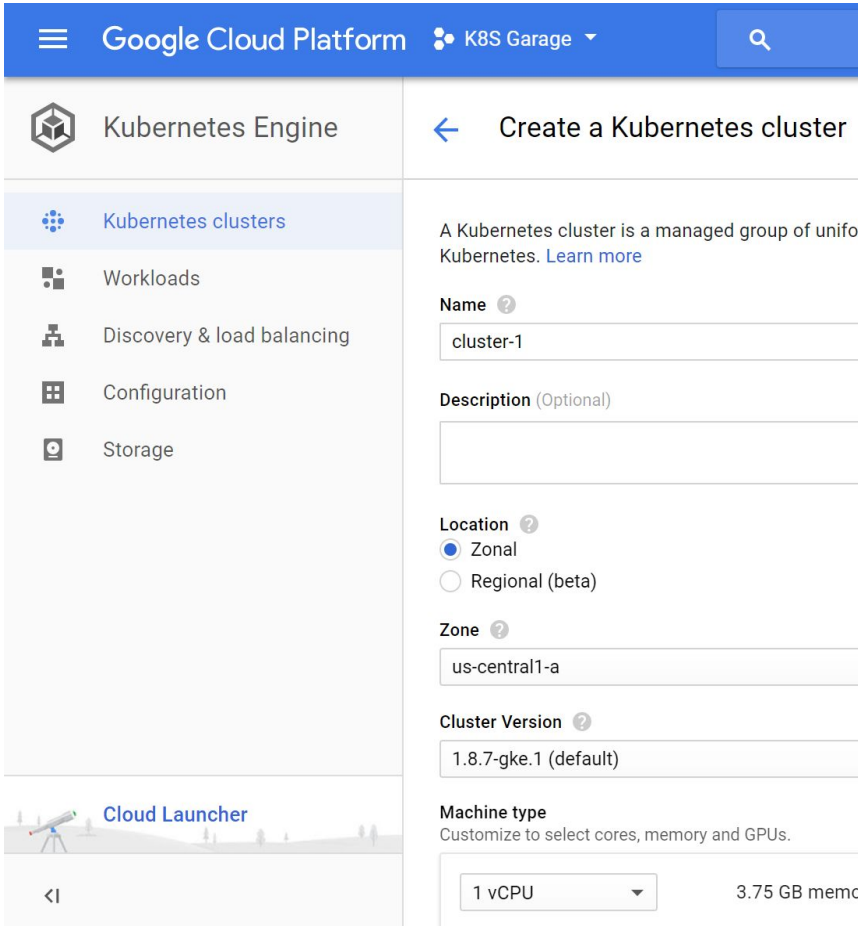
Kubernetes the Easy Way

Start a cluster with one-click

View your clusters and workloads in a single pane of glass

Google keeps your cluster up and running

Google Cloud



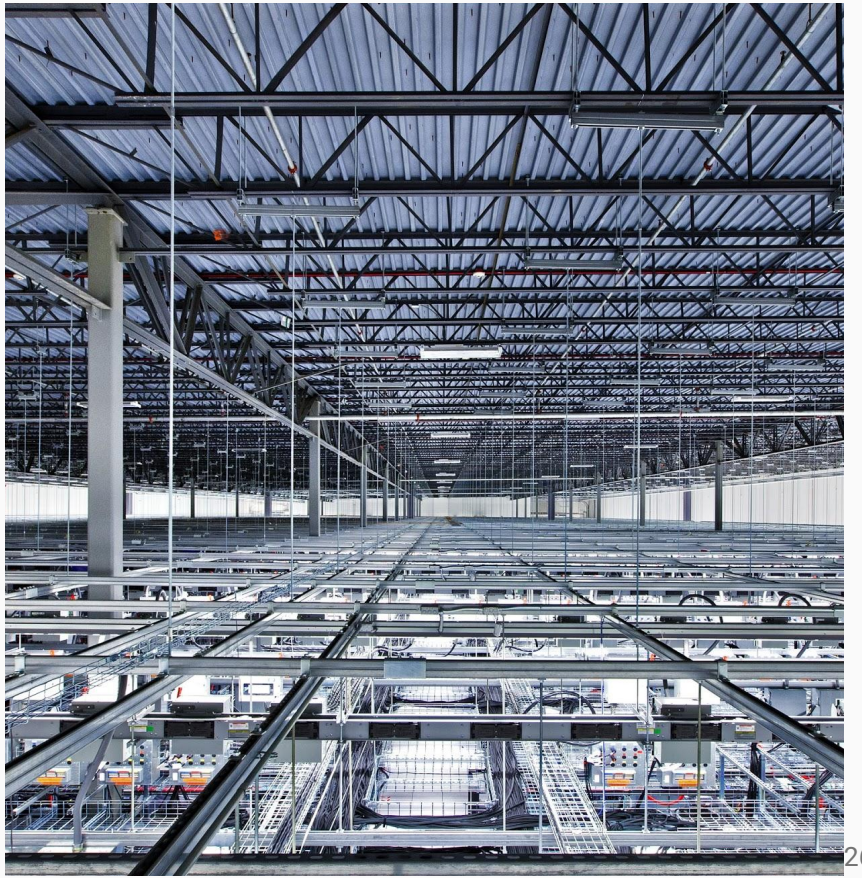
Public cloud: pros and cons

Pros

- ★ Flexibility for infrastructure provisionning:
  - setup a 40 nodes Qserv cluster in 0.5 days
  - extend it to 50 nodes in 10 seconds
- ★ Excellent support from Google engineers
- ★ Easy to setup development clusters with few maintenance
- ★ Cool proprietary features

Cons,

- ★ Expensive for production platform
  - 100K in 3 months for LSST
- ★ Easy to get stuck with proprietary features
- ★ Hide Kubernetes internals so may be difficult to setup
- ★ Run slower than bare-metal (~25%)



On-premise: pros and cons

Pros

- ★ Flexibility on cluster setup
  - ○ DIY Kubernetes
  - ○ Fine-tune your components (local HDD)
- ★ Require skilled engineers
- ★ Ease to guarantee your workload portability
- ★ Run faster than private cloud

Cons

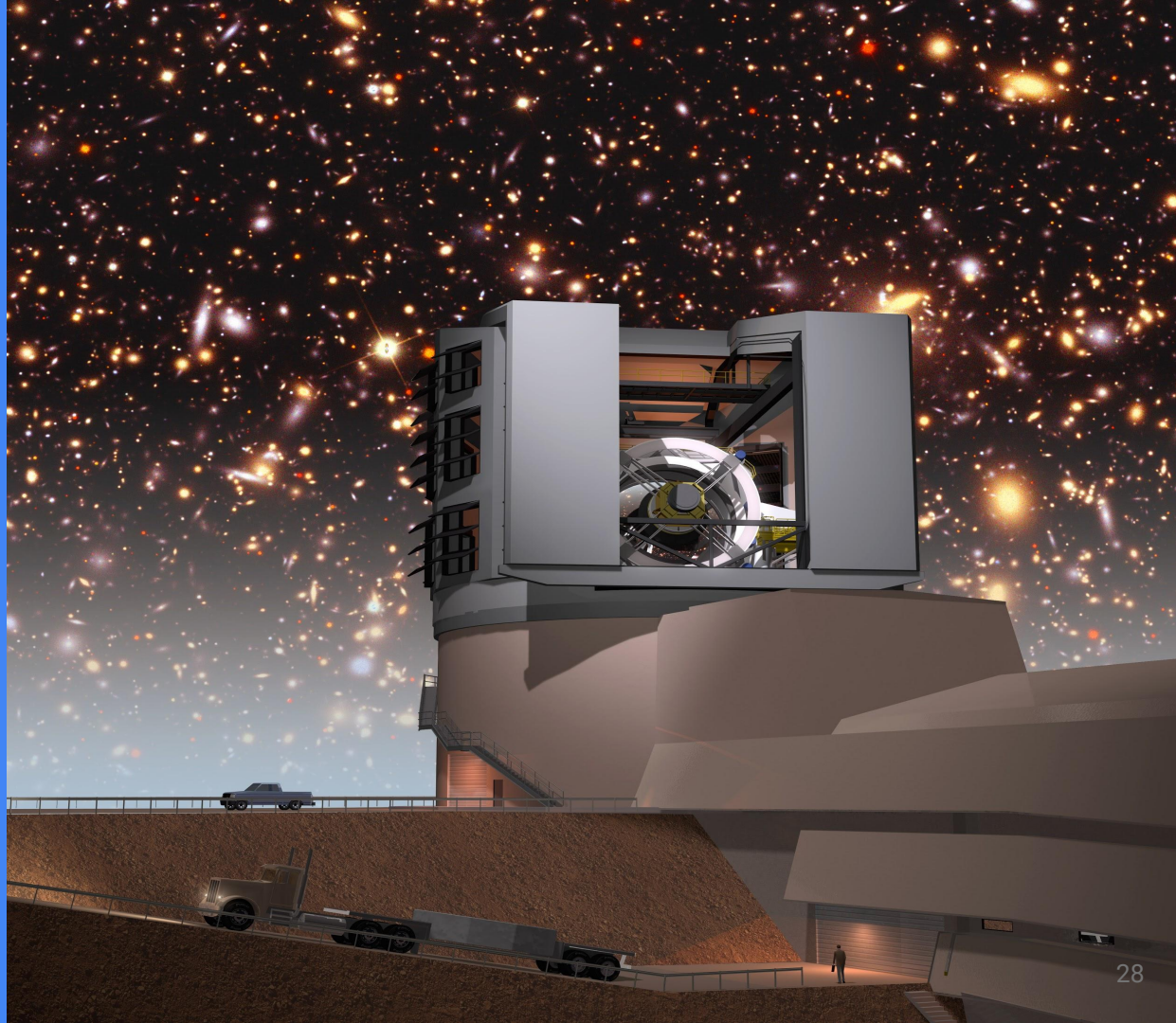
- ★ Difficult to retrieve the global cost
- ★ Require manpower for setup and maintenance
- ★ Hardware upgrade are cost-effective and slow
- ★ Difficult to rebuild the cluster from scratch

Thanks!

Contact:

Fabrice JAMMES
LPC
Clermont-Ferrand

fabrice.jammes@in2p3.fr



Implementation Details