

# ATLAS-Kubernetes integration for workloads

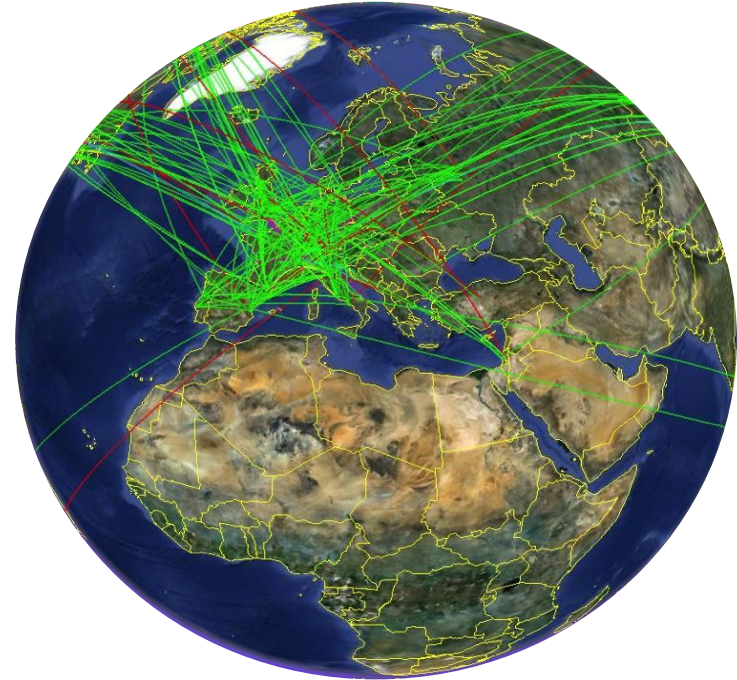
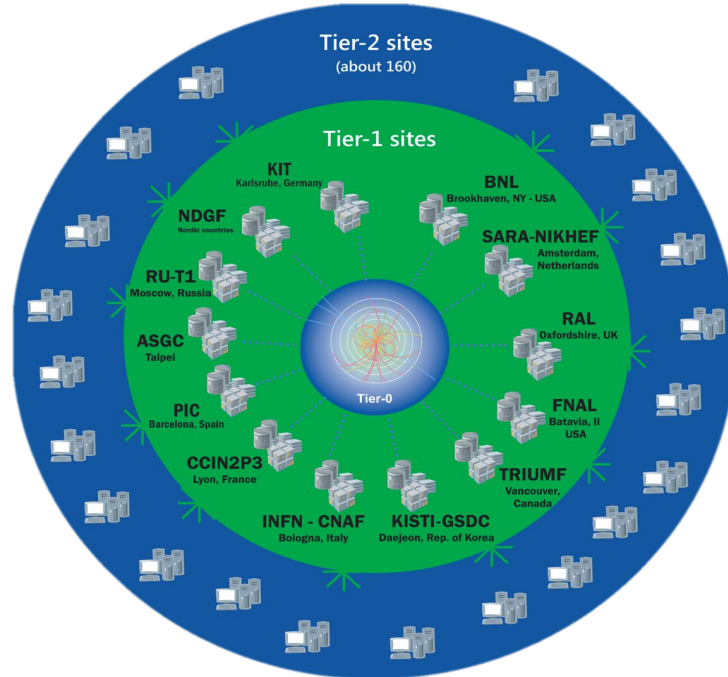
Fernando Barreiro Megino  
HEP-Google Technical Interchange Meeting  
(25 March 2020)



UNIVERSITY OF  
**TEXAS**  
ARLINGTON



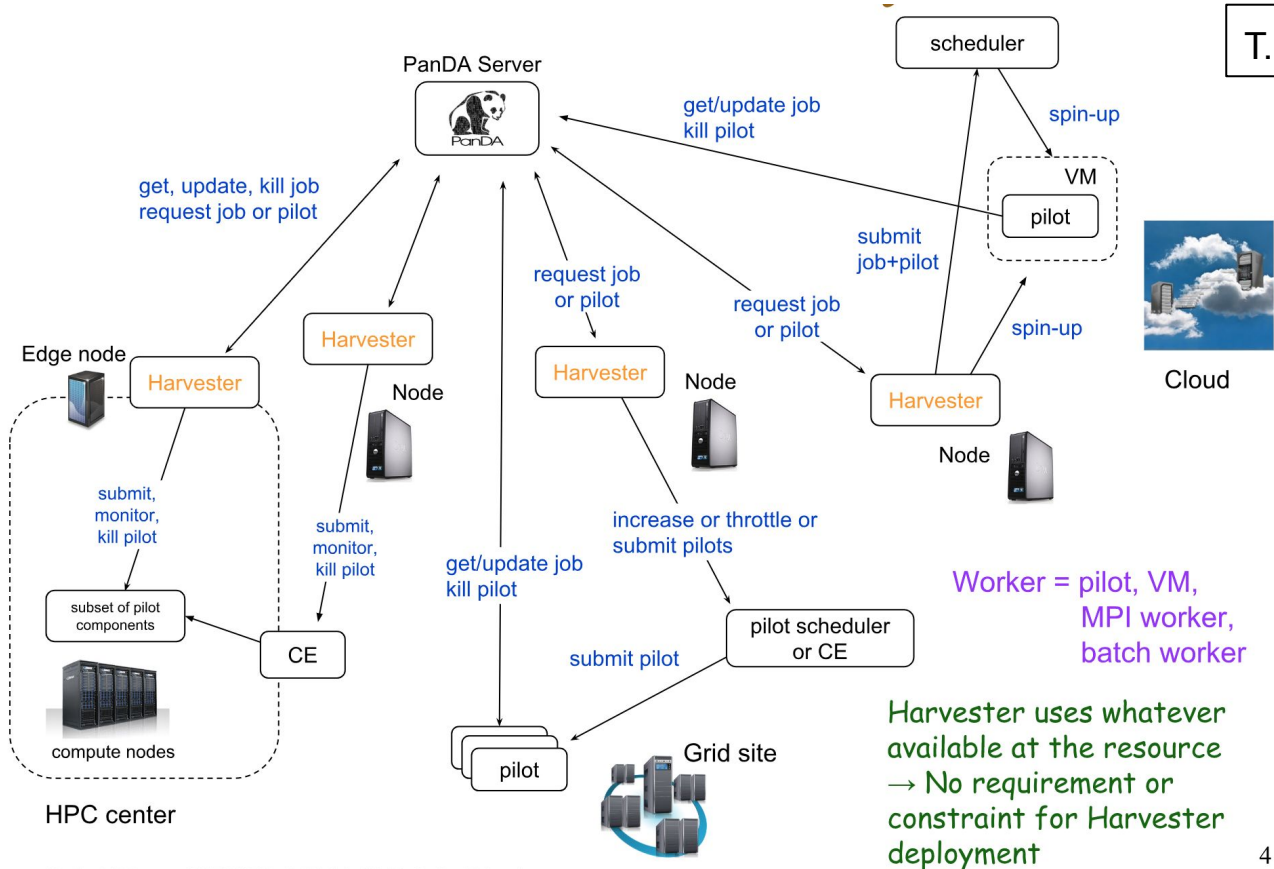
# Worldwide LHC Computing Grid (WLCG)



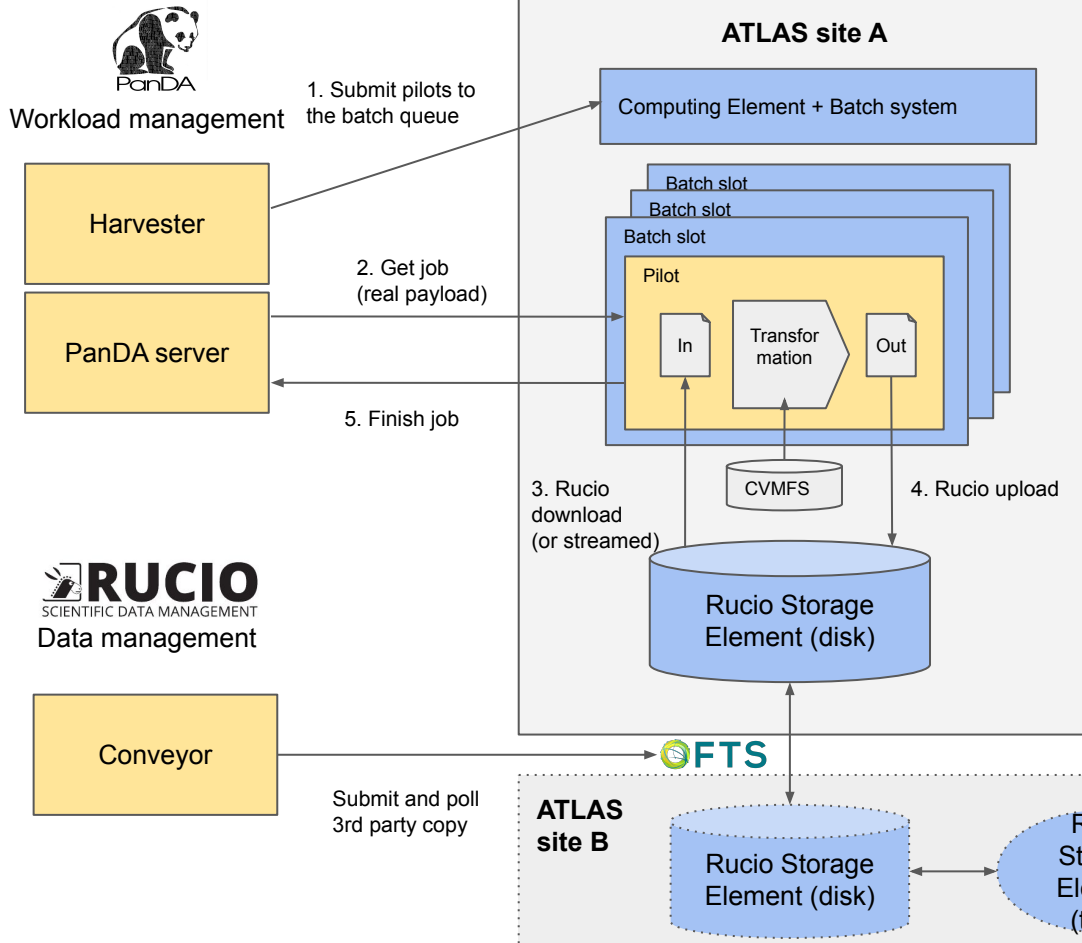
- Shared by LHC experiments (ATLAS, CMS, LHCb, ALICE)
- ATLAS: 150 sites, 400k CPU cores, 500 PB data (including tape archive)
  - Heterogeneous resources: grid sites, cloud computing, HPC, volunteer

# Harvester: universal resource interface

T. Maeno

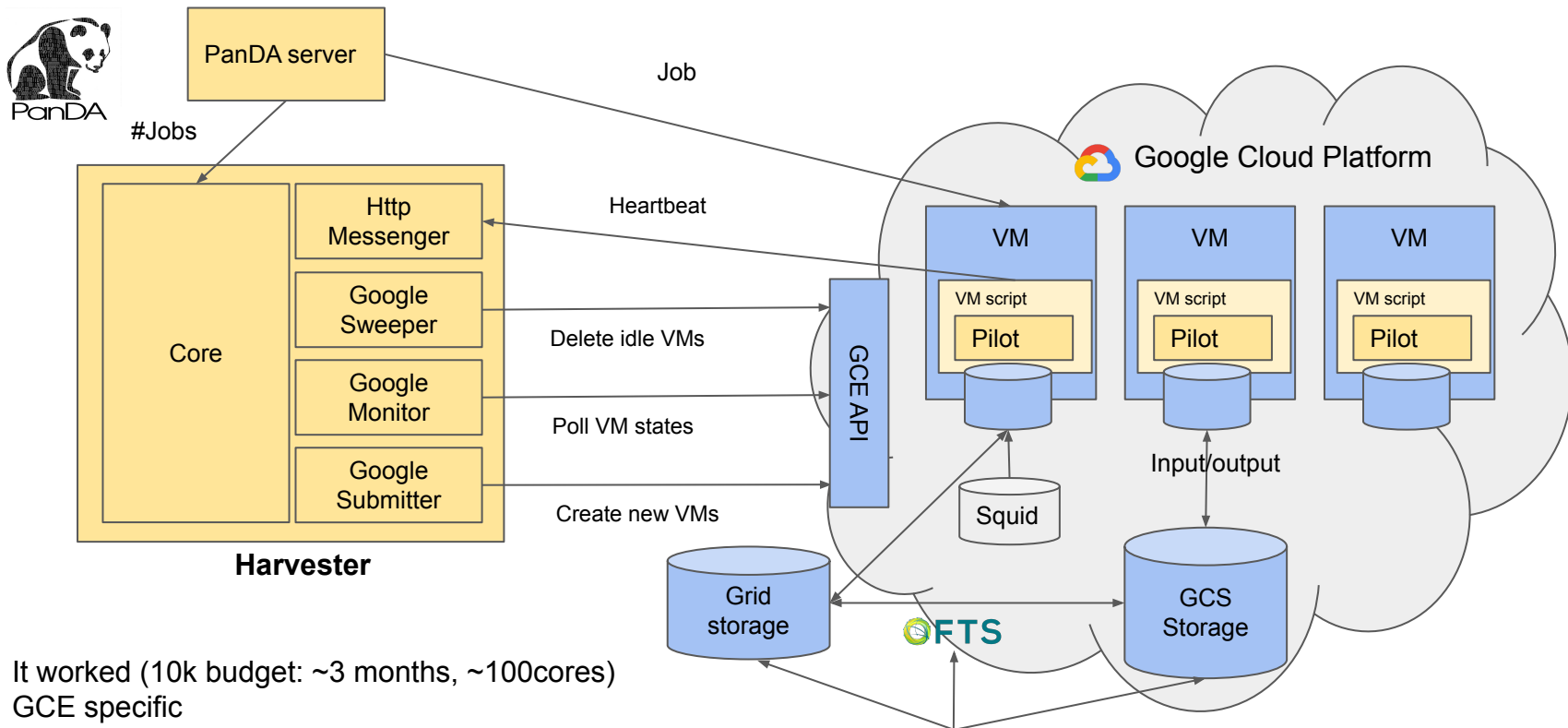


# ATLAS Distributed Computing: basic concepts at a site



- Computing Element: WLCG interface to batch systems
- CVMFS: CERN VM File System. Read only file system distributed hierarchically through Squids. Contains all the ATLAS software
- Rucio Storage Element (RSE): contains ATLAS simulated and detector data
- FTS: File Transfer Service, part of WLCG middleware
- There can be many RSE implementations (dcache, EOS, GCS...). Third party copy will use understandable protocol for both (gridftp, xrootd, http)

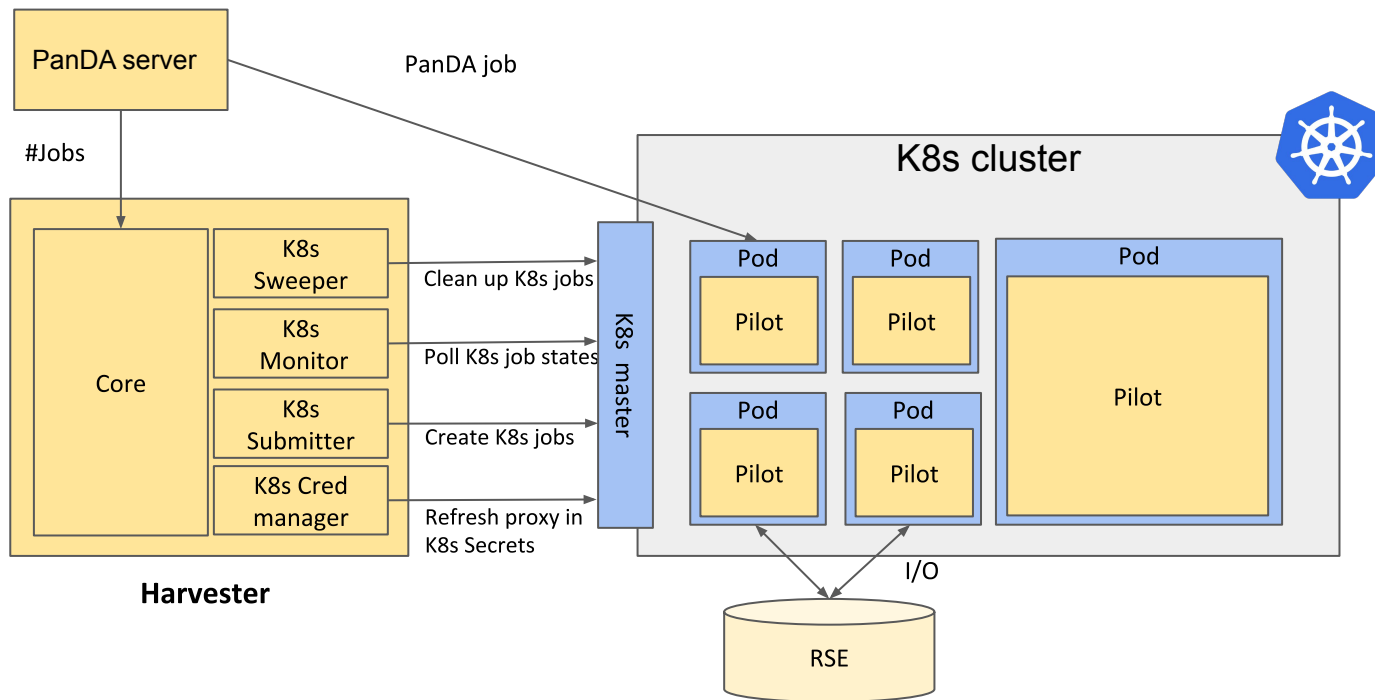
# What about other resources? GCE integration 2018



- + It worked (10k budget: ~3 months, ~100cores)
- GCE specific
- VM overhead

→ Karan was talking about containers continuously

# What about other resources? K8S integration 2019



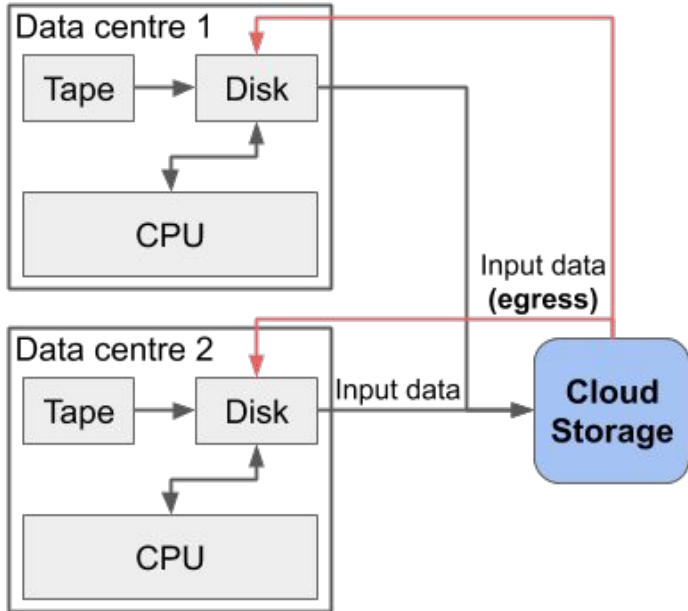
There are still limitations in current model, but we are working on improving it

# Advantages and possibilities of K8S integration

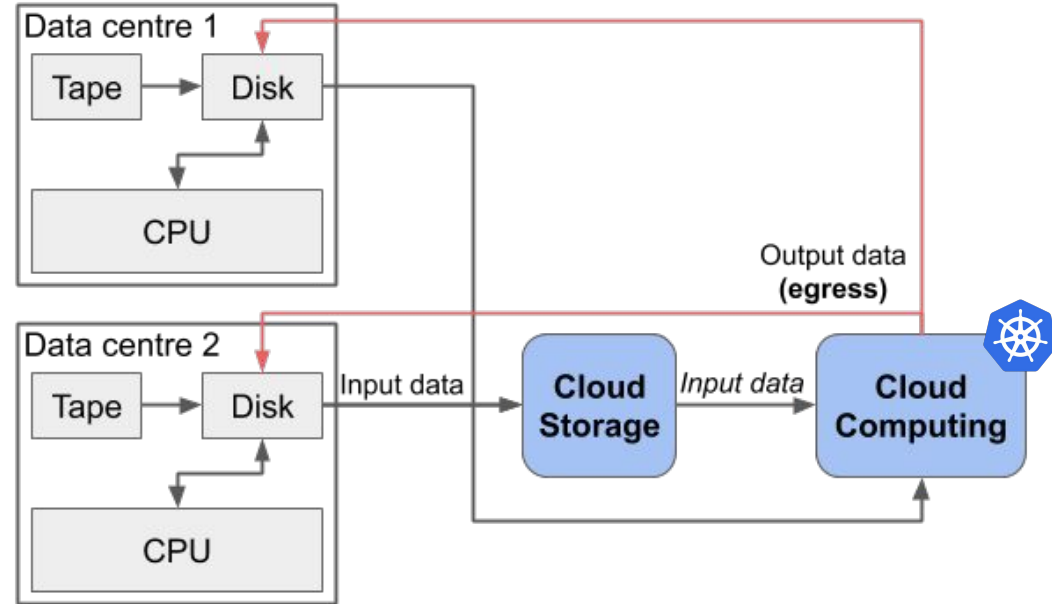
- Native container environment
- In theory standard interface across major cloud providers and WLCG clusters
  - CRITICAL!!!
- Massive infrastructure simplification compared to Grid-batch sites
  - However also losing Grid features/experience
  - Discovering many new behaviours
- Last month mini Kubernetes-grid with central Harvester starting to grow
  - Couple hundred cores each
    - Academia Sinica (Taiwan)
    - CERN (Switzerland)
    - University of Chicago (US)
    - University of Victoria (Canada)
- New idea: implement a Kubernetes ML/GPU cluster

# Kubernetes in Hot-Cold storage context

Tobias Wegner



**Scenario 1.** Cloud only as intermediate storage



**Scenario 2.** Cloud also for processing



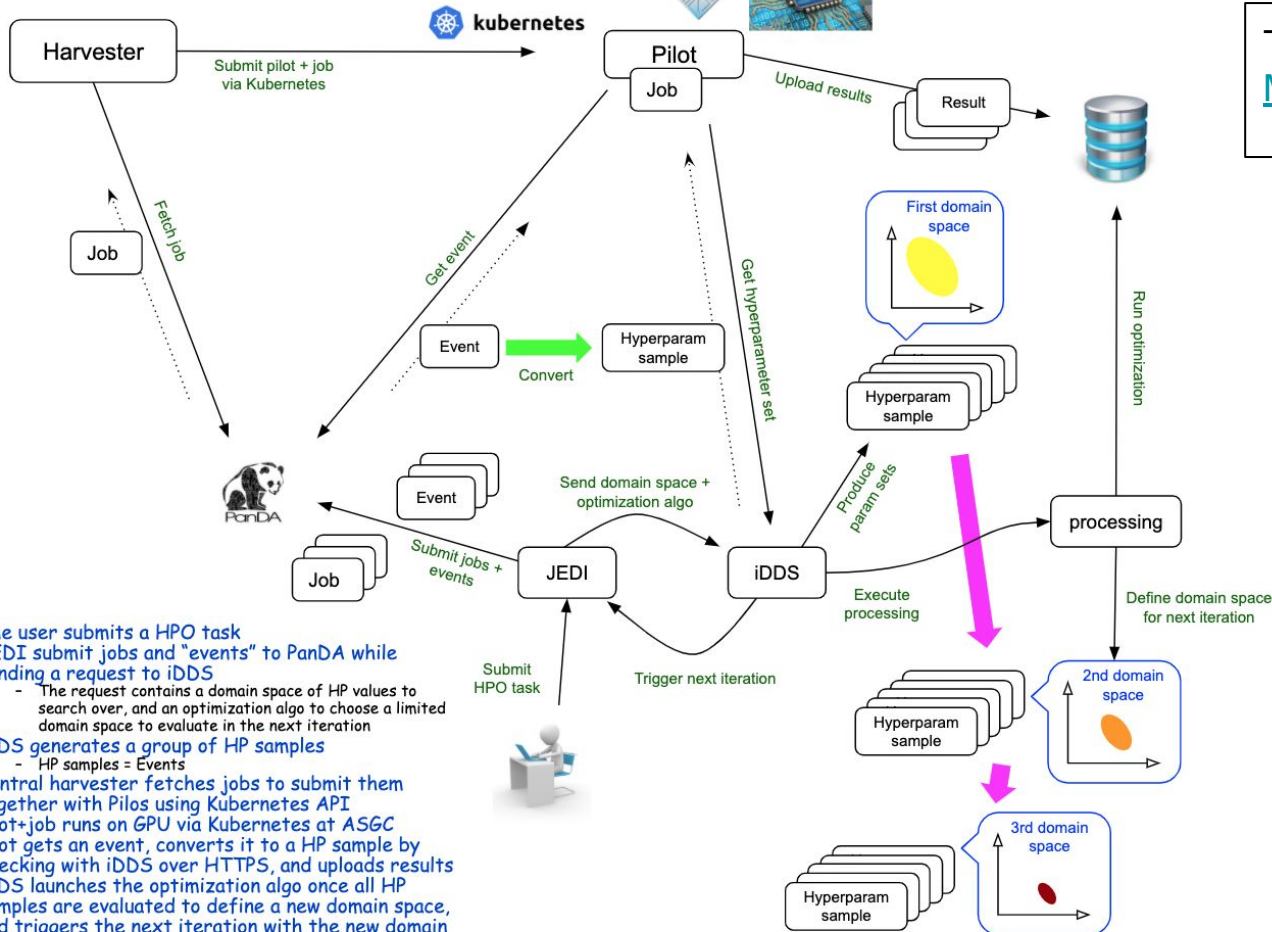
# Scenario 2: discussion

- Technical point of view
  - First integration should be quick if GKE doesn't require special adaptations
  - GKE features/products that would be interesting
    - Autoscaling
    - GKE batch
      - Universal interfaces are critical
  - However no experience in scaling up beyond 2000 cores across 4 clusters
- Hot/Cold strategy and cost model
  - Strategy for data in GCS
    - Cache for one-time campaign?
    - Or longer term archival to reuse over multiple campaigns
    - Can we shorten the GCS data lifetime by processing it immediately?
      - In theory we could run many thousand cores on GKE
  - Egress cost: input vs output size
- Need to get to Step 3 in Alexei's introduction

Backup slides



# Schematic with ASGC GPU



Tadashi Maeno proposal for [ML service for HPO](#)

- The user submits a HPO task
- JEDI submit jobs and "events" to PanDA while sending a request to iDDS
  - The request contains a domain space of HP values to search over, and an optimization algo to choose a limited domain space to evaluate in the next iteration
- iDDS generates a group of HP samples
  - HP samples = Events
- Central harvester fetches jobs to submit them together with Pilos using Kubernetes API
- Pilot+job runs on GPU via Kubernetes at ASGC
- Pilot gets an event, converts it to a HP sample by checking with iDDS over HTTPS, and uploads results
- iDDS launches the optimization algo once all HP samples are evaluated to define a new domain space, and triggers the next iteration with the new domain space