



# CompF4: Analysis Facility Discussion & Priorities

Mark Neubauer, Oksana Shadura, Gordon Watts



## Definition of future AF

- What is a definition of nowadays “Analysis Facility” and how we see it for next 5 years?

A working version of an AF definition from the [March HSF AF Forum kick-off meeting](#):

*“The infrastructure and services that provide integrated data, software and computational resources to execute one or more elements of an analysis workflow. These resources are shared among members of a virtual organization and supported by that organization.”*

How might we think about AFs 5-10 years from now?

Still < 10% of HL-LHC computing resources?



## Building blocks for AF

- Modern authentication
- Data delivery and data management tools
- Integration of the object storage
- Intensive data caching
- Support of new ecosystems (python) and columnar analysis
- Introducing DevOps techniques for providing
  - Lab vs University environments?
- Reusability/Sharing of existing computing resources
- Shared Usage Controller
  - Fair-share without compromising fast turn-around?
  - More than what a batch system provides?

**What we need to add in this list?**



# Analysis facilities and preparation for HL-LHC

- Analysis frameworks and its integration in the existing facilities - how to make it successful?
- “Test-driven development” for Analysis Facilities?
  - The 10-minute benchmark?
- Lack of users feedback, “don’t expect tuning and cooperating” - how we can make facilities more user friendly?
  - Monitoring to discover usage patterns?



# New features to take in an account while designing Analysis Facilities

- Portability as it is envisioned in L. Bryant talk
  - Building blocks of services
  - Stitched together uniquely for each site
- Interoperability
- Providing Enhanced Security (Doug Benjamin)
- “Cloud Native” design (Maria Acosta)
  - Service based architecture
  - Api-driven communication between services
  - Container based infrastructure
  - declarative system to manage the configuration and operations of every element of the platform
- User-defined containers for customizing and sharing python environments (Mark Neubauer)
  - Configuration management, injection of secrets, etc.
- Enabling analysis teams to work together
  - Shared data, shared resource usage, shared queries, shared caches, etc.



# Discussion Points

## Interoperability

- Labs vs Universities
  - Labs - putting services together that are individually managed and bespoke
  - Universities - putting together a chart that spins up everything
  - Can we create a common experience in an AF for all users with these two approaches? Should we?
- What do we use as the substrate - how do we automate our infrastructure?
  - Universities - K8S
  - Labs, other places - OpenShift
- Different cloud providers?
  - University, Lab, Amazon, Azure, Google?
- Experiment environment vs Modern Python Environment - Compatibility?
  - Experiment environment (C++ development)
  - Modern analysis environment (newest ROOT, python, DASK, etc.)
  - Containers with same view of shared disk resources (user/scratch)?

How do we make sure we don't have islands of development?