Technical vision, challenges, plans

Andrew McNab
University of Manchester
LHCb and GridPP

Eiger, Monch, Jungfrau: Nov ‘16

Vision

Vision

- “A common e-Infrastructure for STFC sciences”

- “Common” means we avoid unnecessary duplication of services, management effort, support effort
- “Common” means we avoid artificial technical barriers between pools of resources, allowing us to use available capacity and to create mixed HPC/HTC workflows
- “e-Infrastructure” means transparent access to resources, with workflows and data managed by software rather than manual recipes
- “STFC sciences” means we accommodate the full breadth of sciences funded by STFC (eg astronomy) or supported by STFC facilities (eg materials science)

Vision

- We are proposing to do this within today's paradigms of cloud computing
- This gives us access to well-supported software ecosystems like OpenStack
- Allows us to work with commercial providers who now present their services as clouds
- Cloud paradigms like Infrastructure-as-a-Service allow us to give users new options
  - Like creating transient, virtual "sites" with their own workers and services
- Existing e-Infrastructure we rely on still fits in
  - eg Looking at "Grid" as a Platform-as-a-Service cloud
- **Cloud language is just easier to explain to people today**

Challenges

Challenges

- We have to integrate resources based on different technologies and “cultures”
  - High Throughput Computing farms
  - High Performance Computing clusters
  - Infrastructure-as-a-Service clouds
  - Large scale data stores
  - Large scale databases
- We have to support user communities with
  - Very different scales: Prof+Postdoc+Phd vs 5000-person international collaborations
  - Very different cultures: “computing is in-house and hands-on” vs “computing is outsourced and turn-key”

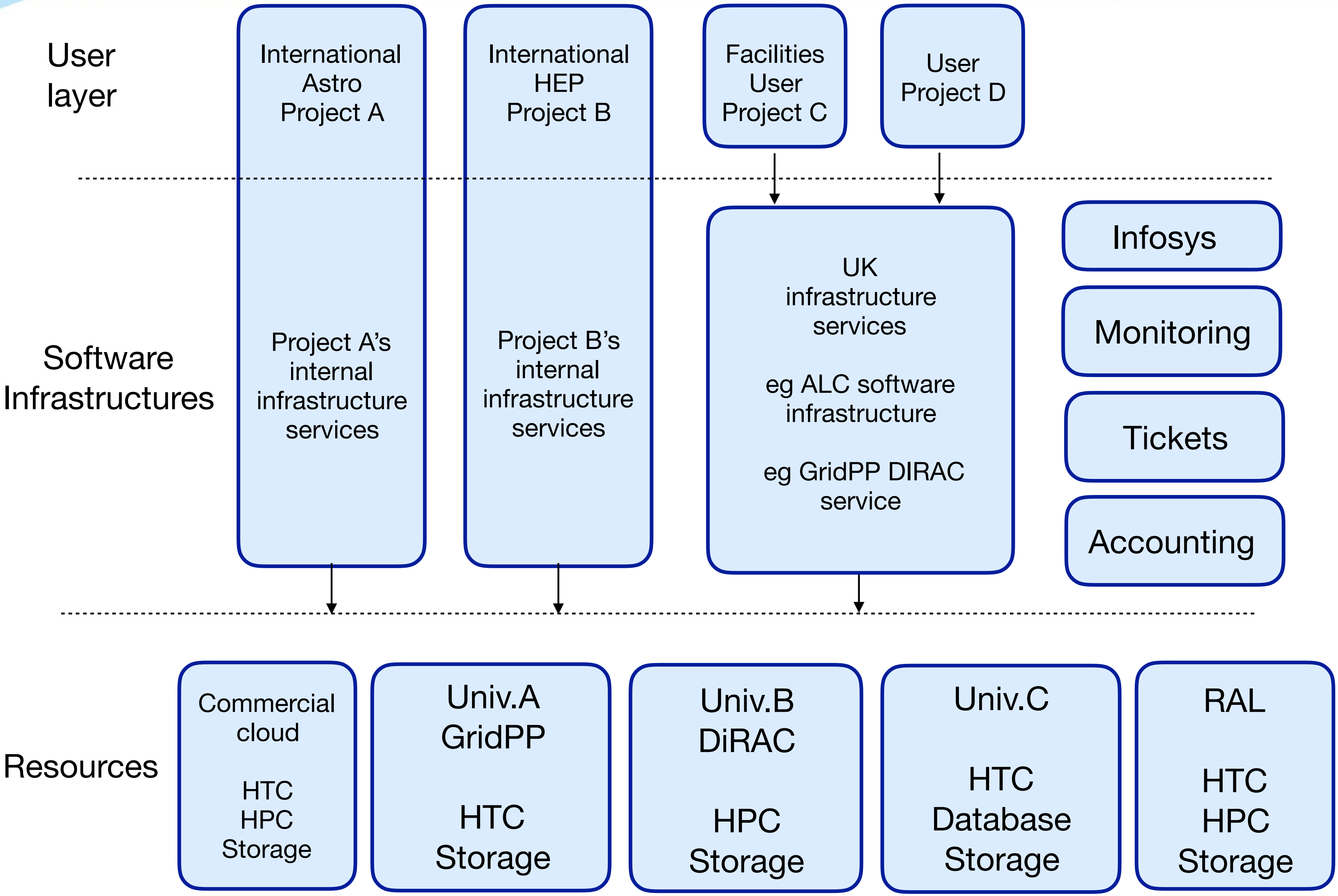
Challenges

- We have to understand how to build and support new platforms for scientific computing that go beyond batch
  - Not just big HTC batch farms with grid interfaces
  - Not just big HPC clusters with fast interconnects
  - VMs and containers with user-supplied images
  - Users orchestrating their own workers and transient services on our infrastructures
- Users demand convenient single sign-on interfaces
- And all of this at a large scale and distributed across multiple sites
  - At labs, universities and commercial providers
- And we already have £1.2million hardware to get working!

Plans

Where would we like to be in ~2022?

- Transparent access to HTC, HPC, Storage and Databases
- Able to use resources located at commercial providers, universities, and labs
- To do this we need “software infrastructures” to federate resources
  - Large international projects (eg LHC experiments) already have software infrastructures to do this, and typically want direct access to resources
  - But smaller user communities usually don’t have this kind of thing in place, and use private resources or a single site (an account on a big HPC machine somewhere?)
- We also need services that support operations (tickets, monitoring) and administration (accounting systems)


User layer
International Astro Project A
International HEP Project B
Facilities User Project C
User Project D
Software Infrastructures
Project A's internal infrastructure services
Project B's internal infrastructure services
UK infrastructure services
eg ALC software infrastructure
eg GridPP DIRAC service
Infosys
Monitoring
Tickets
Accounting
Resources
Commercial cloud
HTC
HPC
Storage
Univ.A
GridPP
HTC
Storage
Univ.B
DiRAC
HPC
Storage
Univ.C
HTC
Database
Storage
RAL
HTC
HPC
Storage

Software Infrastructures

- SCD work already ongoing and supported by the 2017 capital grant
  - Data Movement Service and Virtual Machine Manager
- ALC for facilities users
  - eg Data Analysis as a Service
- International Projects with their own infrastructures
  - eg PanDA/Rucio for ATLAS, DIRAC for LHCb/CTA/ILC/…
- International Projects intending to build their own
  - eg SKA European Science Data Centre
- Others which already exist
  - eg GridPP DIRAC service
- New infrastructures user communities develop themselves?

Outline plan

- We already have working systems that most user communities can use at some level
  - So evolution rather than revolution
- The 2017 capital grant gives us “elbow room” to start trying new things out too
  - Without interfering with existing production services
- We need to be doing demonstrably useful work
  - While developing, deploying and testing new technologies
  - And bringing new user communities on board

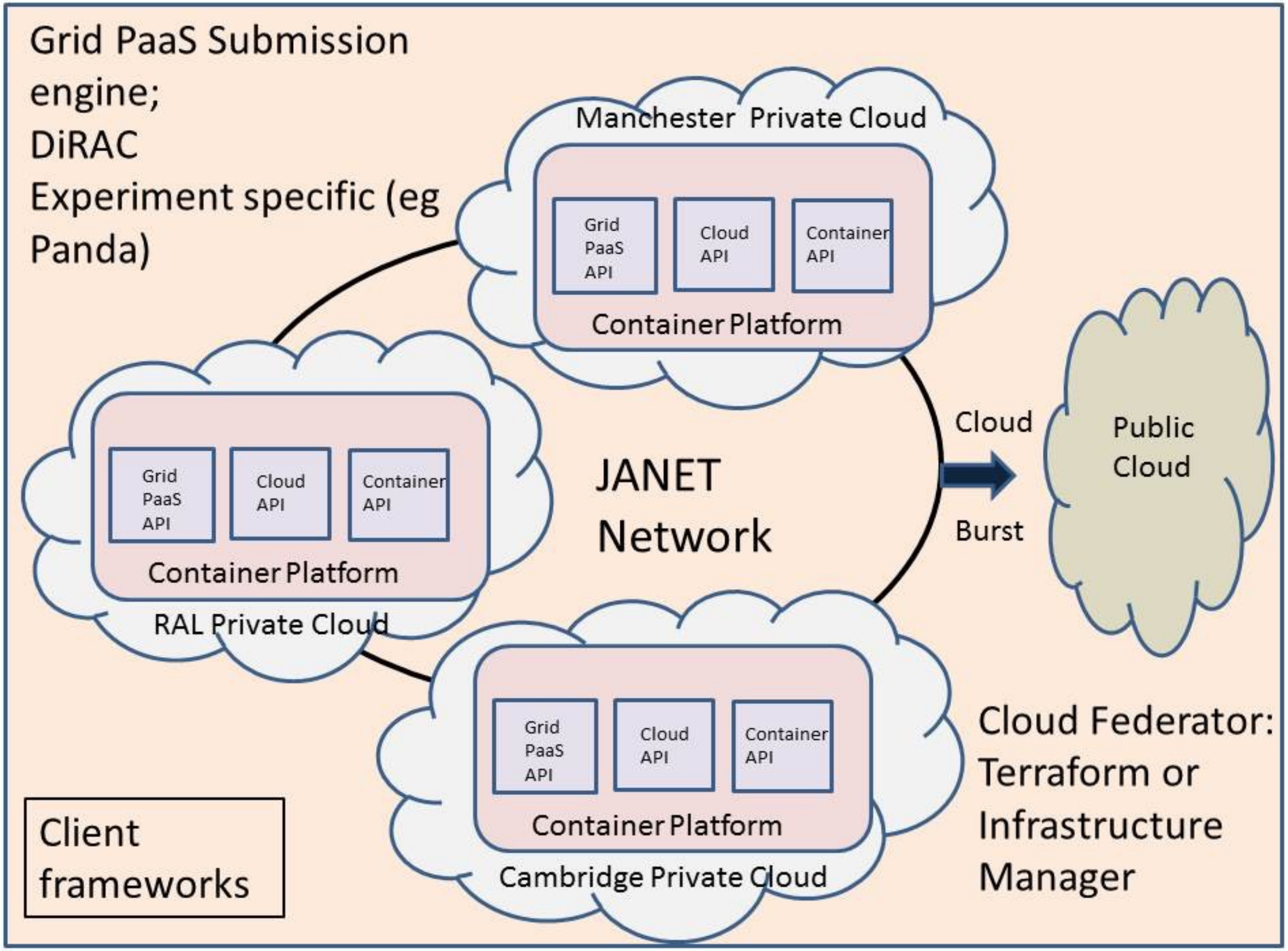
How to manage evolution?

- We want to be able to try new things and allow change
  - But “no surprises” to people running production work
- Proposal:
  - Define a current baseline of “blessed” interfaces which resources will expose (eg OpenStack Nova API for VM management)
  - Software infrastructures know what they can rely on
  - Notifying people of changes to the baseline manages change
  - Other interfaces can be tried by cooperation between software infrastructures and resources
  - We can drop interfaces from the baseline if it becomes clear they aren’t needed

Proposed 2018 plan

- Keep the hardware busy by backfilling with existing workloads
- Identify and support some pathfinder “new” user communities
  - “New” means user communities who have not used distributed computing infrastructures in the UK before
  - But to be doing production work in 2018 at scale, they probably need to have some workflows in place already (that they run on a big single site at the moment?)
- Expose Infrastructure-as-a-Service resources, and work with user communities wanting to use non-batch ways of working
- Start/continue work on the software systems which can support large numbers of small user communities
  - See other talks yesterday and today

CPU in the e-Infrastructure bid



Proposed **initial** baseline of interfaces

- Grid / Platform-as-a-Service
  - ARC CE (HTCondor or Slurm batch)
  - *?HTCondor-CE (HTCondor batch)?*
- Cloud Infrastructure-as-a-Service
  - KeyStone/Nova/Glance
  - Federated KeyStone to reduce admin burden to sites
- Vacuum Cloud
  - Run VM/Container definitions published by users
- Storage
  - GridFTP, *?xrootd?*, WebDAV, *?S3?*, *?Swift?*
- Databases
  - ODBC

Authentication and Authorization

- See Jens talk for the details of this

- Initially the baseline naturally continues to use X.509
  - X.509 user/host certificates, proxy certificates delegated to jobs, attribute certificates (VOMS) for group membership
  - We could apply X.509 to OpenStack too
- But we aim to allow users to use systems like JISC Assent to avoid the need to handle X.509 credentials
  - We can use translation services to allow users to acquire X.509 credentials
  - Just initially? Indefinitely? What will international projects require?

“Upstairs/Downstairs” factorisation

- We are aiming to have access mediated by software infrastructures
- This means the users will usually authenticate to their software infrastructure
- Infrastructures will authenticate to resources
- In existing software infrastructures of the LHC experiments, only a handful of official identities need to authenticate to the compute resources
  - Users’ jobs are running inside some kind of “box” (Pilot Job, VM, Container)
- This means we probably don’t need to decide how to allow thousands of users to create VMs at each site, for instance
- User community developers and operations teams may though

“Inspiration”

Jupyter notebooks on the SWAN service at CERN

- This is an example of the kind of thing we could build on the e-Infrastructure

- Jupyter notebooks provide a convenient way of doing data analysis
  - Interactive shell in a browser window
  - Container technology behind this
- SWAN at CERN does the execution in multiple containers run on the CERN OpenStack cloud
- With an e-Infrastructure, we could use the e-Infrastructure as the backend instead of just a single site
- The next two slides suggest required components (like AAI)

SWAN in the CERN Ecosystem

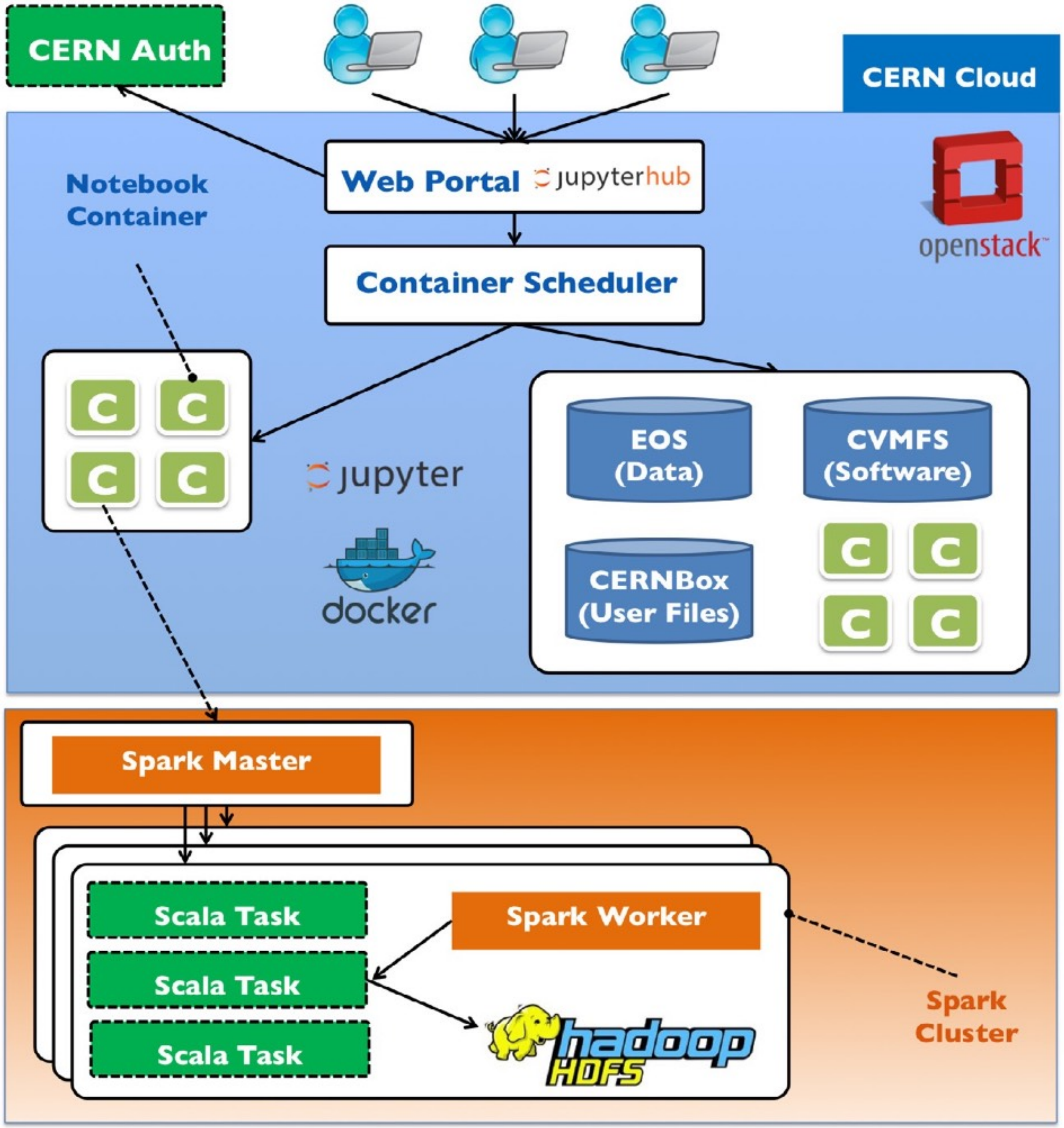
SWAN relies on production technologies at CERN:

- Authentication with CERN credentials
- Infrastructure: virtual machines in OpenStack Cloud
- Software distribution: CVMFS
- Storage access: CERNBox, EOS
  - All experiment data potentially available!

Plus some external technologies:

- JupyterHub
- Docker

5


CERN Auth
CERN Cloud
Web Portal jupyterhub
Notebook Container
Container Scheduler
openstack
C C C C
jupyter
docker
EOS (Data)
CVMFS (Software)
CERNBox (User Files)
C C C C
Spark Master
Scala Task
Scala Task
Scala Task
Spark Worker
hadoop HDFS
Spark Cluster

Summary

- Clear outline plan emerging for creating "A common e-Infrastructure for STFC sciences"
  - Lots of details to be understood of course
- Workable 2018 plan exists to exploit the hardware from the 2017 capital grant
- Evolutionary approach will allow us to manage change as we involve more user communities and technologies
- We get efficiencies of scale etc from all this, but it's not just that:
  - We're opening up lots of new ways of working