DIRAC

THE INTERWARE

Dirac Grid

Jennifer2 II Computing Workshop – Task 5.1

February 20th 2024

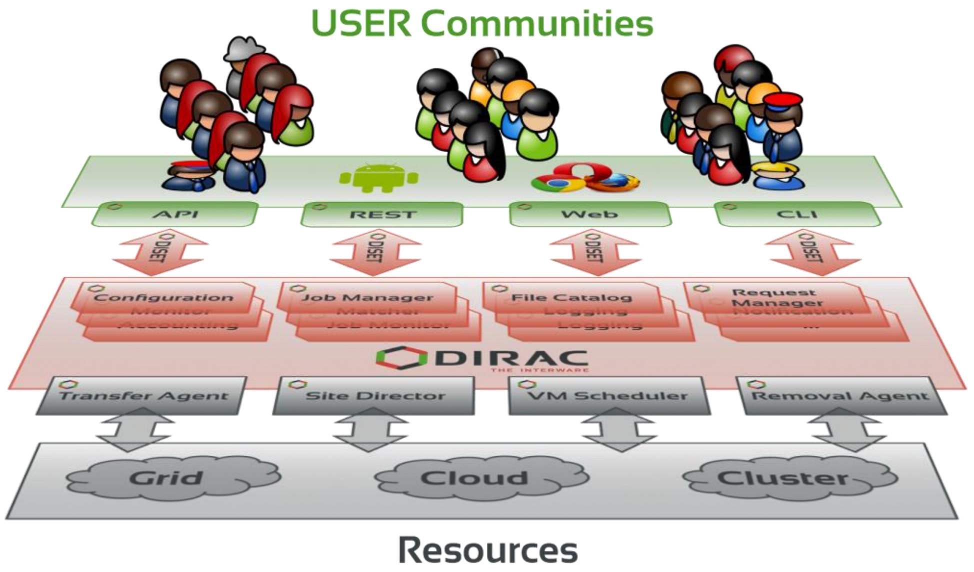
https://agenda.infn.it/event/39895/

Federico Stagni
Technical coordinator
On behalf of the DIRAC consortium

What is Dirac Grid

DIRAC
THE INTERWARE

- A software suite for **distributed computing**
- A **complete** solution to one or more **user community**
- Builds a layer between **users** and **resources**



- Developed by communities, for communities
- Open source (GPL3+)
- github.com/DIRACGrid
- Python 3
- Publicly documented, yearly users workshops, open developers meetings and hackathons

The DIRAC **consortium** as representing body

What you do with Dirac

DIRAC
THE INTERWARE

The "basics" (WMS and DMS)

- Obviously, send jobs to "the Grid"
- interface with different sites
  - With different *Computing Elements*
    - and *batch systems*
  - with different *storage elements*
- interface with different *information systems*
- interface with different *catalogs*, including Rucio
- interface with different *DBs*
- authenticate through different *identity providers*

DataSet Management and more

- manage “productions” (e.g. reconstruction, simulation…)
- manage dataset transfers and removals
  - or delegate the task
- interact with FTS
- provide a failover system
  - your jobs won’t fail because a certain SE is down, nor because the central service are down
- (if possible at all) transfer data from the experiment (the "online") to a Grid SE (the "offline")
- monitor your resources with a policy-based system
- … and more

Used by multiple experiments/projects, inside HEP, astronomy, and life science

Experiment agnostic

Extensible

Flexible

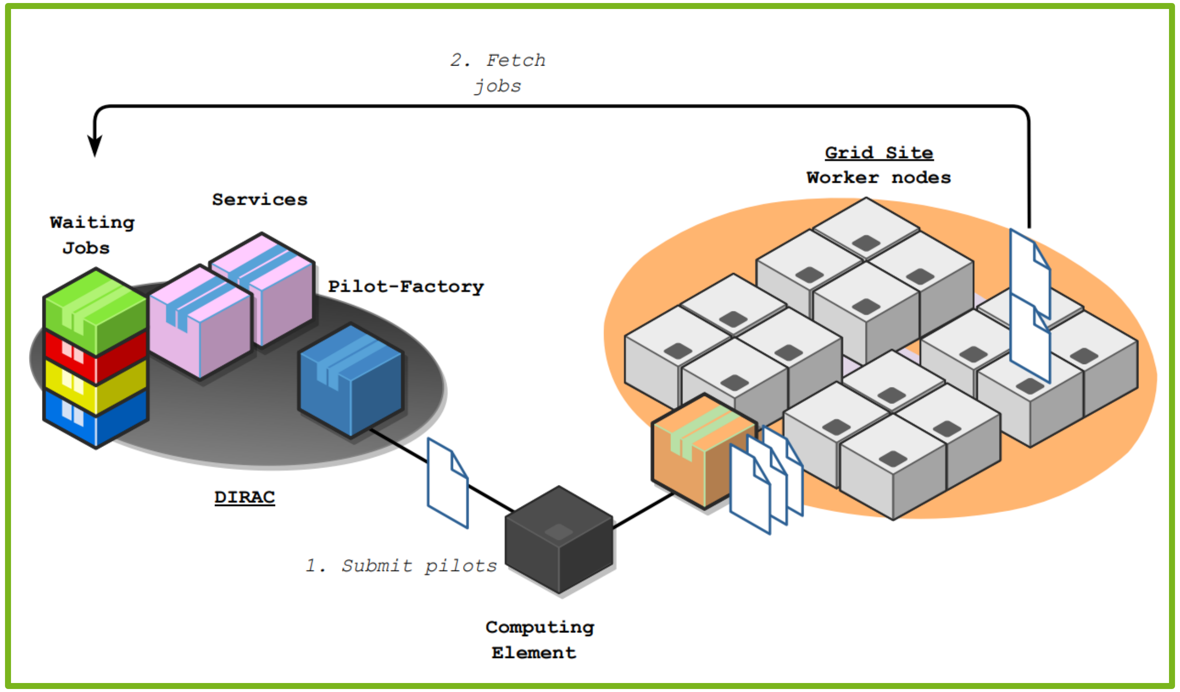
Workload Management (WMS)

DIRAC

THE INTERWARE

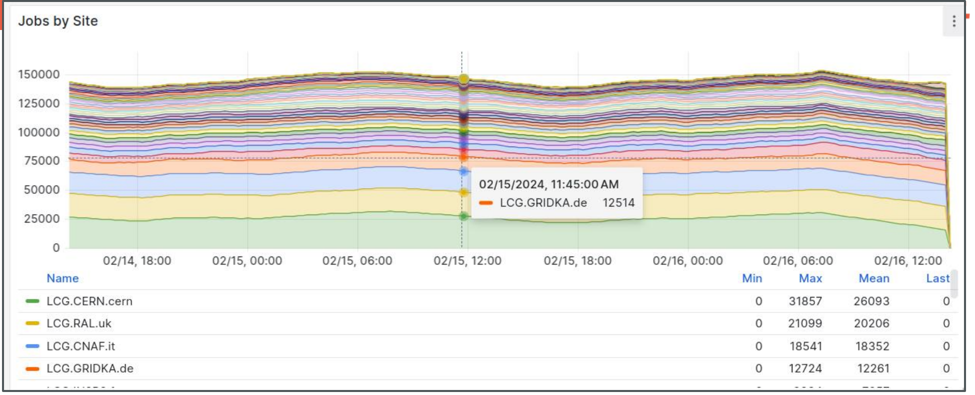
- **Users** define and submit **jobs**. Jobs have **requirements**. Job descriptions are stored in DIRAC's Job DB.
- Independently, **Pilots** are started (1) on the **sites' worker nodes (WN)**
- Pilots will try to **match** (2) the worker nodes' capabilities to the jobs requirements.
- Jobs are started on WNs. DIRAC monitors their progress.

Pilots-based WMS, with late binding



Computing Resources

- Grids (nowadays: HTCondor, AREX)
  - through proxies and/or tokens
- Clusters behind a BS
  - access through SSH/GSISSH tunnel
    - a really thin layer that we call "SSH CE"
  - and then talks with batch system
- VMs scheduler:
  - Based on apache libcloud
  - Contextualization from standard images
    - with, at least, the DIRAC pilot
- Vacuum:
  - VAC/vcycle resources
  - BOINC Volunteer resources
  - HLT farm (LHCb)
- HPC sites
  - it often means at least SSH+Slurm

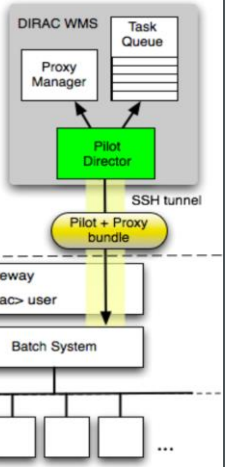


Jobs by Site

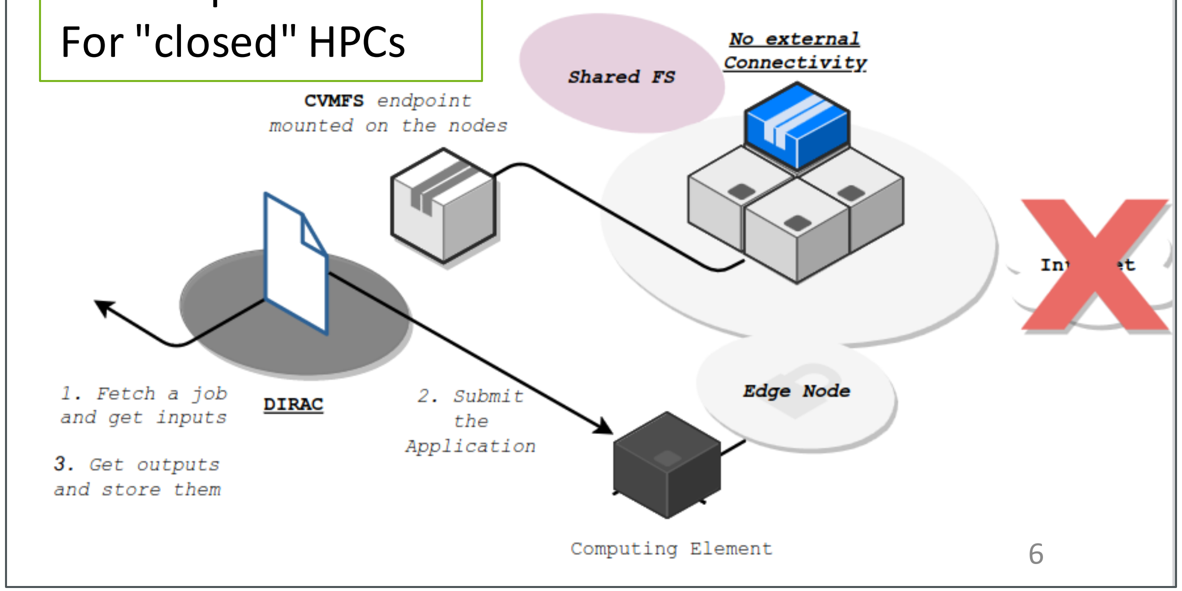
02/15/2024, 11:45:00 AM
LCG.GRIDKA.de 12514

Name	Min	Max	Mean	Last
LCG.CERN.cern	0	31857	26093	0
LCG.RAL.uk	0	21099	20206	0
LCG.CNAF.it	0	18541	18352	0
LCG.GRIDKA.de	0	12724	12261	0

SSH CE



The no-pilot model
For "closed" HPCs



Shared FS
No external Connectivity
CVMFS endpoint mounted on the nodes
Int   et
1. Fetch a job and get inputs
DIRAC
2. Submit the Application
3. Get outputs and store them
Edge Node
Computing Element

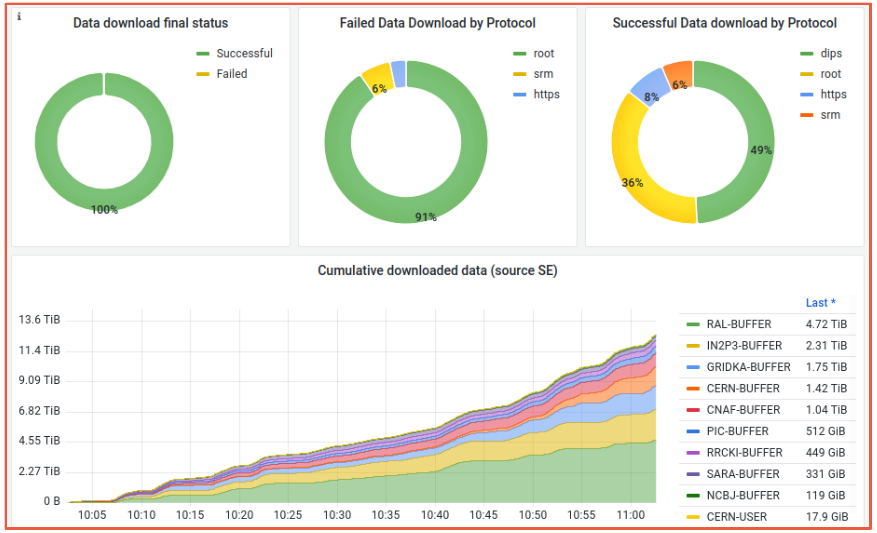
Data Management (DMS)

DIRAC

THE INTERWARE

basically, it’s about **files**:

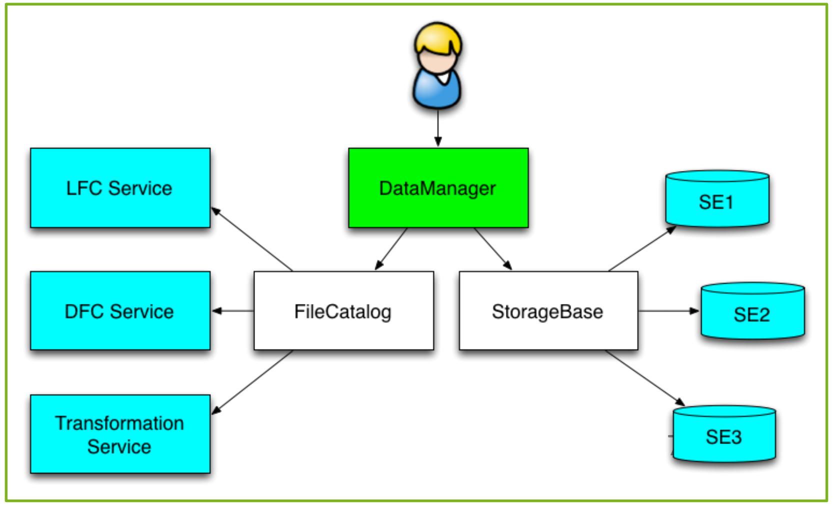
⇒ placing, replicating, removing files



	Last *
RAL-BUFFER	4.72 TiB
IN2P3-BUFFER	2.31 TiB
GRIDKA-BUFFER	1.75 TiB
CERN-BUFFER	1.42 TiB
CNAF-BUFFER	1.04 TiB
PIC-BUFFER	512 GiB
RRCKI-BUFFER	449 GiB
SARA-BUFFER	331 GiB
NCBJ-BUFFER	119 GiB
CERN-USER	17.9 GiB

Implementation details:

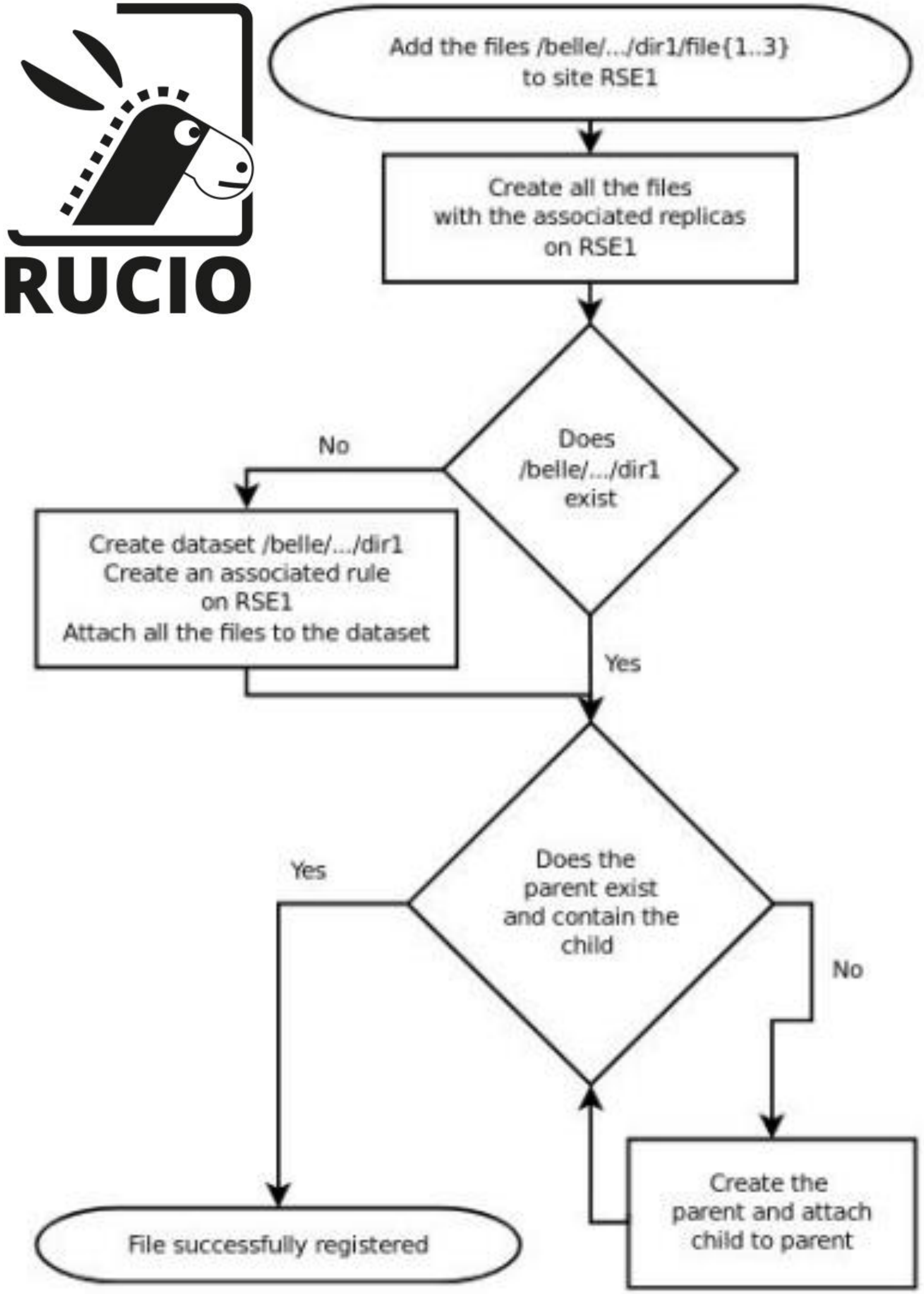
- there are **LFN**s (logical file names) → and users ONLY work with these
  - e.g. /lhcb/MC/2018/SIM/00155979/0013/00155979_00132406_1.sim
    - if you ever worked with files in LHCb, you recognize the above format
- LFNs are registered in **catalog**(s)
  - where are the LFNs? (DIRAC File Catalog)
  - what are their metadata? (LHCb Bookkeeping)
- LFNs *may* have **PFN**s, stored in **SE**s.
- You can access those PFNs with several protocols.



Data Management, but with Rucio

The "entry point" is the Rucio FileCatalog:

- Developed in DIRAC RucioFileCatalog (RFC)
  - similar to LFC or DFC
- No change for the download/upload, which is still done via the DIRAC DataManager
- All the replication policies, 3rd party copy are handled by Rucio subscriptions and rules
- The synchronization between DIRAC and Rucio is done via DIRAC agents



Add the files /belle/.../dir1/file{1..3} to site RSE1

Create all the files with the associated replicas on RSE1

Does /belle/.../dir1 exist

No

Create dataset /belle/.../dir1
Create an associated rule on RSE1
Attach all the files to the dataset

Yes

Does the parent exist and contain the child

Yes

No

Create the parent and attach child to parent

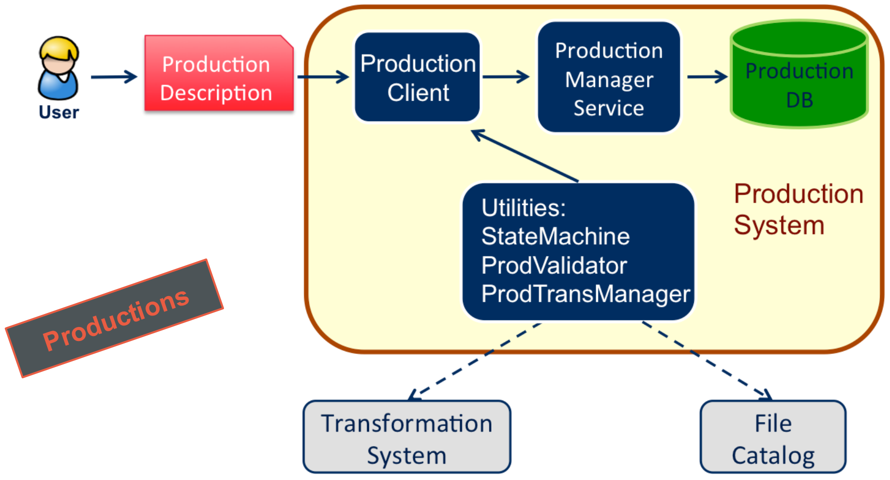
File successfully registered

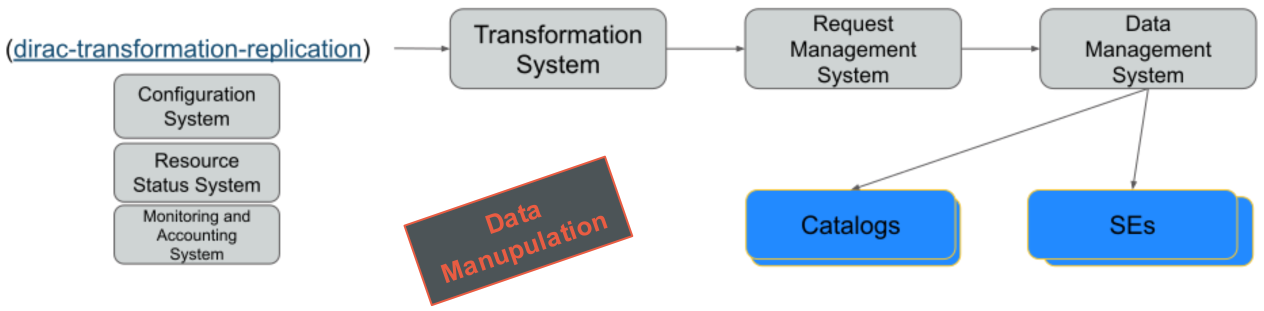
Productions and dataset management

DIRAC
THE INTERWARE

The **Transformation System (TS)** is used to automate common tasks related to production activities

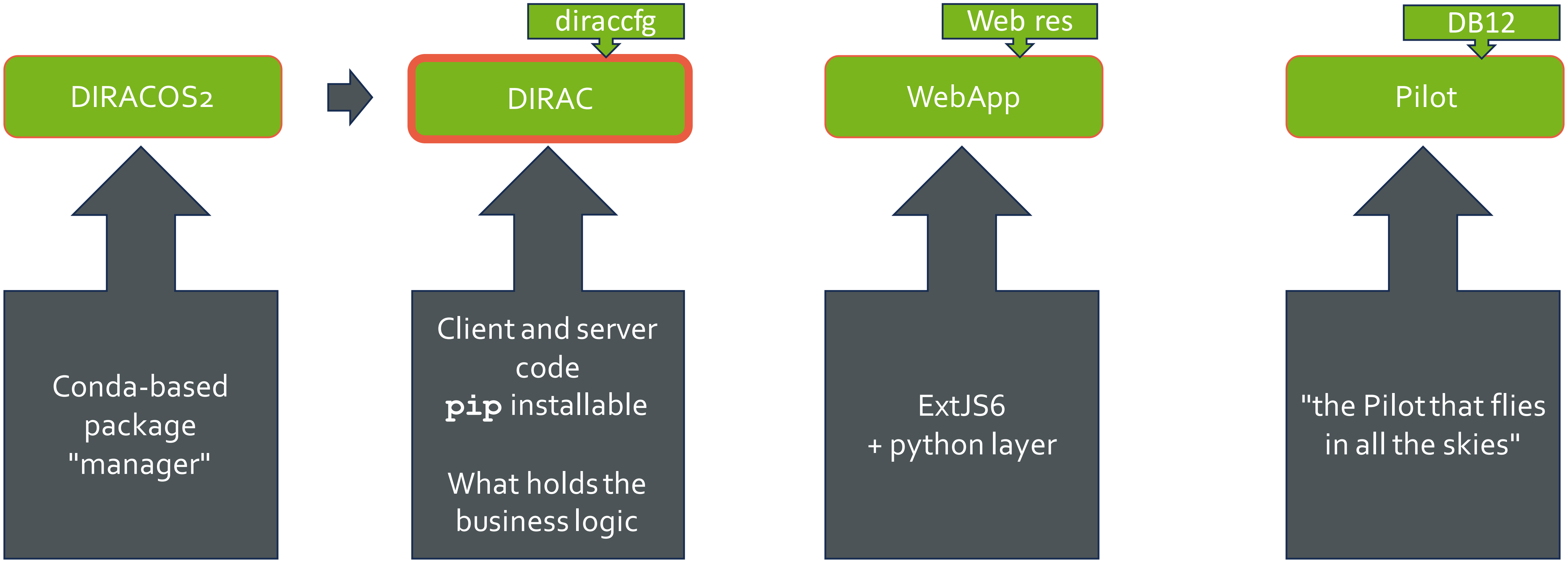
- A "*production*" is a transformation managed by the TS that is a "Data Processing" transformation (e.g. Simulation, Merge, DataReconstruction...). A Production ends up creating jobs in the WMS.
- (Unless you are using Rucio) a "Data Manipulation" transformation replicates, or remove, data from storage elements. A "Data Manipulation" transformation ends up creating requests in the RMS (Request Management System), which feeds the DMS.





Production SW stack (py3)

DIRAC
THE INTERWARE



DIRACOS2

Conda-based package "manager"

DIRAC

diraccfg

Client and server code
pip installable

What holds the business logic

WebApp

Web res

ExtJS6
+ python layer

Pilot

DB12

"the Pilot that flies in all the skies"

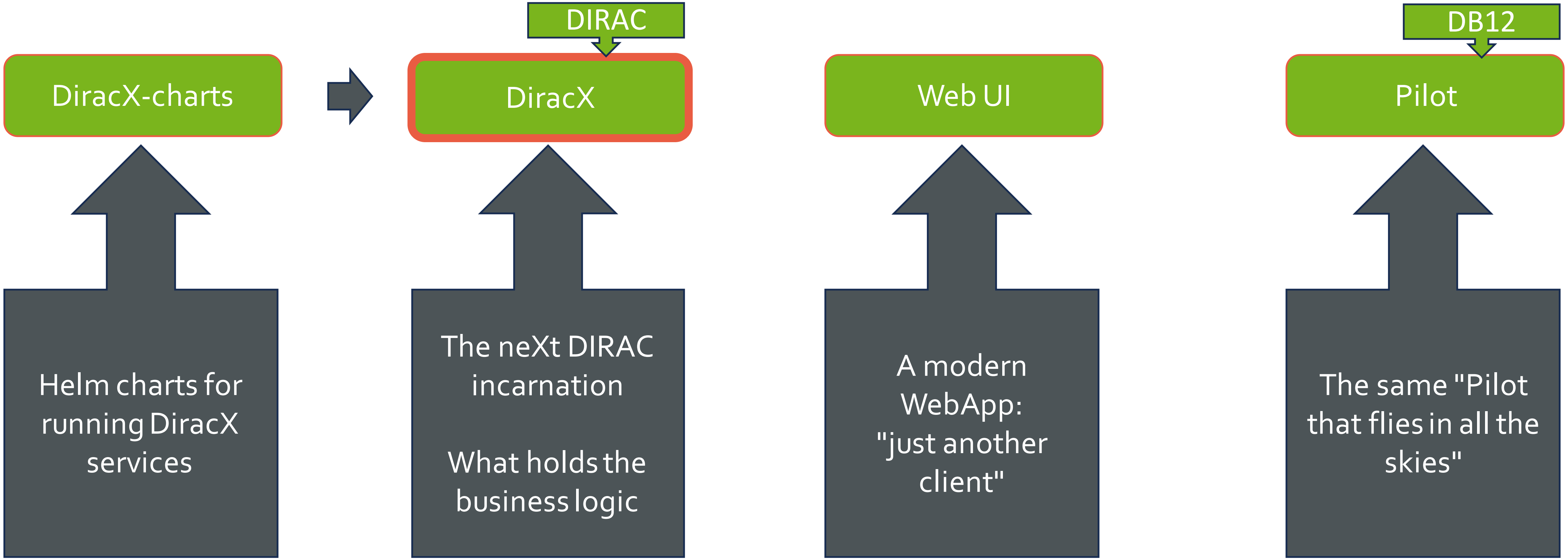
Project status

- The current production version (v8) is at the moment the only supported version
  - Introduced Token authentication for CEs (use case: HTCondor CEs, and AREX), will also introduce Token authentication for FTS (tested with WLCG DC24)
  - Monitoring capabilities extended
  - Expanded HPCs support
- A new software stack, dubbed **DiracX**, is in active development
  - Will replace the current software stack, which will become legacy
- The in-testing version (v9) will serve as jumping point to DiracX

DiracX SW stack

DIRAC

THE INTERWARE



DIRAC

DB12

DiracX-charts

DiracX

Web UI

Pilot

Helm charts for running DiracX services

The neXt DIRAC incarnation

What holds the business logic

A modern WebApp: "just another client"

The same "Pilot that flies in all the skies"

DiracX in just one slide

DIRAC

THE INTERWARE

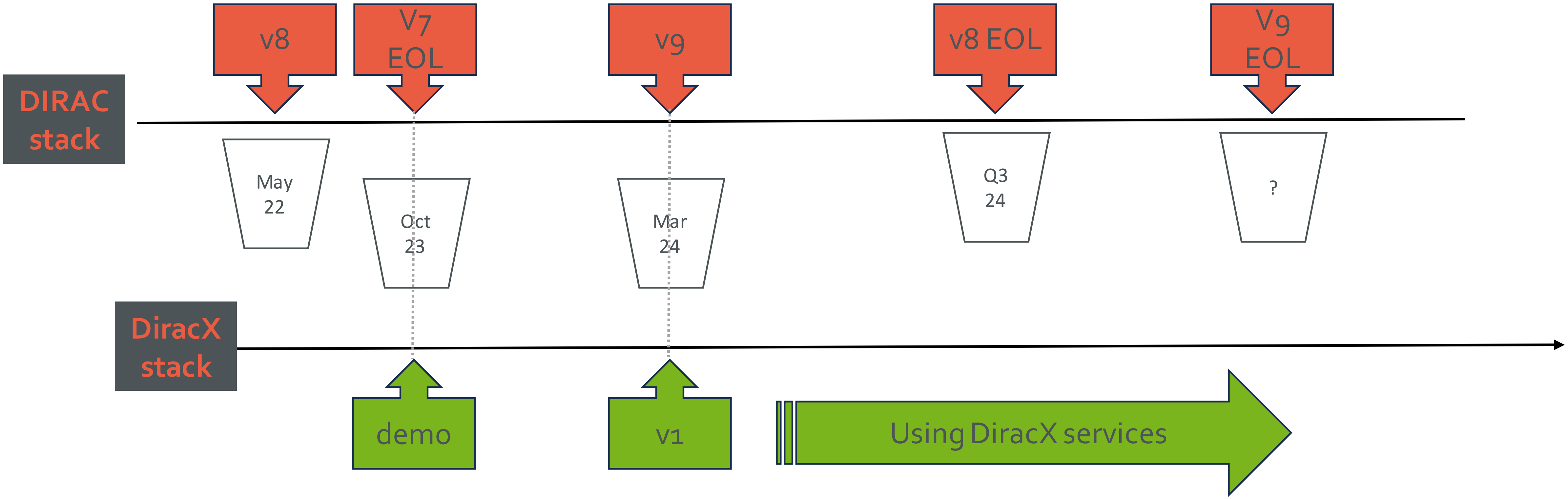
- Current **DIRAC functionalities will be fully retained**
- **Multi-VO** from the get-go

- A **cloud native app**
- Standards-based

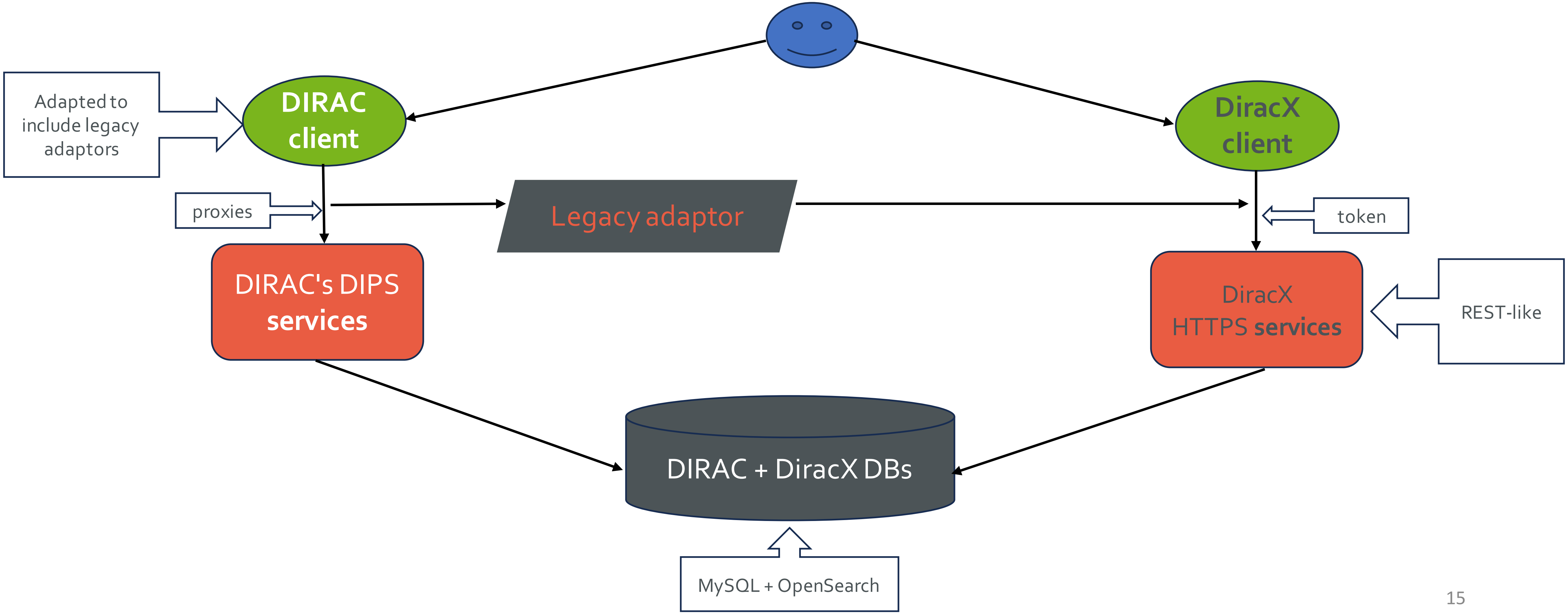
Effectively, **DiracX is a DIRAC rewriting**, with modern technologies

Transitioning to DiracX

DIRAC
THE INTERWARE



Transitioning to DiracX



Transitioning stages
(extreme summary)

DIRAC
THE INTERWARE

- Update to DIRAC v9
  - this, effectively, means also installing DiracX
- Run few services in DiracX
- Activate the legacy adaptor
  - traffic for the selected services will be redirected to DiracX services
  - proxy → token behind the scene
- You can now remove the legacy DIRAC services

This will likely be a multi-year transition!

Development

DIRAC
THE INTERWARE

- Dirac(X) is written in Python
  - Using 3.11 atm, and we try to keep being on track with latest python versions
- WebApp (will be legacy) is written with ExtJS6, and with Python 3.11
- DiracX-Web uses NextJS13 framework, and typescript

- ~6 FTE as core developers, a dozen contributing developers
- We use GitHub Actions "massively"
- We support Puppet profiles for DIRAC stack and Kubernetes charts for DiracX

Past outreach – DR 23

DIRAC AND RUCIO

Workshop 2023

16-20 October 2023
KEK, Tsukuba Campus, Japan

The workshop will be devoted to the information exchange between the DIRAC and the Rucio developers, service administrators and users.

PROGRAM COMMITTEE

Martin Barisits, CERN (Co-Chair)
Cedric Serfon, BNL
Federico Stagni, CERN (Co-Chair)
Andrei Tsaregorodtsev, IN2P3
Ikuo Ueda, KEK/IPNS
Eric Vaandering, FNAL

LOCAL ORGANISERS

Ikuo Ueda, KEK/IPNS
Takanori Hara, KEK/IPNS

ABSTRACTS
The call for abstracts is open.

Registration deadline: 11 September 2023
Registration fee: JPY 9,000
Registration to the workshop is necessary.
Payment of the fee is mandatory before the deadline.

https://indico.cern.ch/e/DR23

- We were in KEK, Tsukuba, Japan for the "DIRAC & Rucio Workshop 2023
  - https://indico.cern.ch/event/1252369/
  - the first of these workshop types
- One combined workshop, not two workshops
  - Community & User talks
  - Technology talks
  - Q&A, Hackathons, Tutorials
- Why a combined workshop:
  - Several experiments are interested in both DIRAC and Rucio. A few are already using both of them in production.
  - Check for possible further collaborations.
  - Mutual developers' interests.

Future outreach

DIRAC

THE INTERWARE

- DIRAC+X **hackathon**:
  - https://indico.cern.ch/event/1376672/
  - 9th-10th April 2024
  - 2 days, around a table, to talk and hack with the developers
- **10th DUW** (DIRAC Users' Workshop 2024):
  - We will be in Lyon (CC-IN2P3) for the 10th DIRAC Users' Workshop
    - https://indico.cern.ch/event/1341205/
    - 19th -> 21st June 2024
  - Poster, registration, and further details will follow in the coming weeks

Get in touch (daily)

DIRAC
THE INTERWARE

- All doc should be on dirac.readthedocs.io
  - For release updates: github.com/DIRACGrid/dirac/wiki
- For questions and issues, through https://github.com/DIRACGrid
  - Have an issue? Feel free to open it
  - We are happy to answer your Discussions, still on github (DIRAC and diracx)
  - Vulnerability reports should also go through GitHub (DIRAC and diracx)
- Mattermost channel: https://mattermost.web.cern.ch/diracx/
- If you want to participate to weekly developers meetings, drop me an email

Summary

- Dirac Grid is a fully-capable software suite for distributed computing
- A single service/installation can handle several Virtual Organizations
- DIRAC is being rewritten with modern technologies
  - Its functionalities will be ported to the new implementation (DiracX)
  - DIRAC v9 will be the way we get there
  - From the users' point of view, the transition to DiracX will happen with minimal disruptions

DIRAC

THE INTERWARE

?

Backup

DIRAC

THE INTERWARE

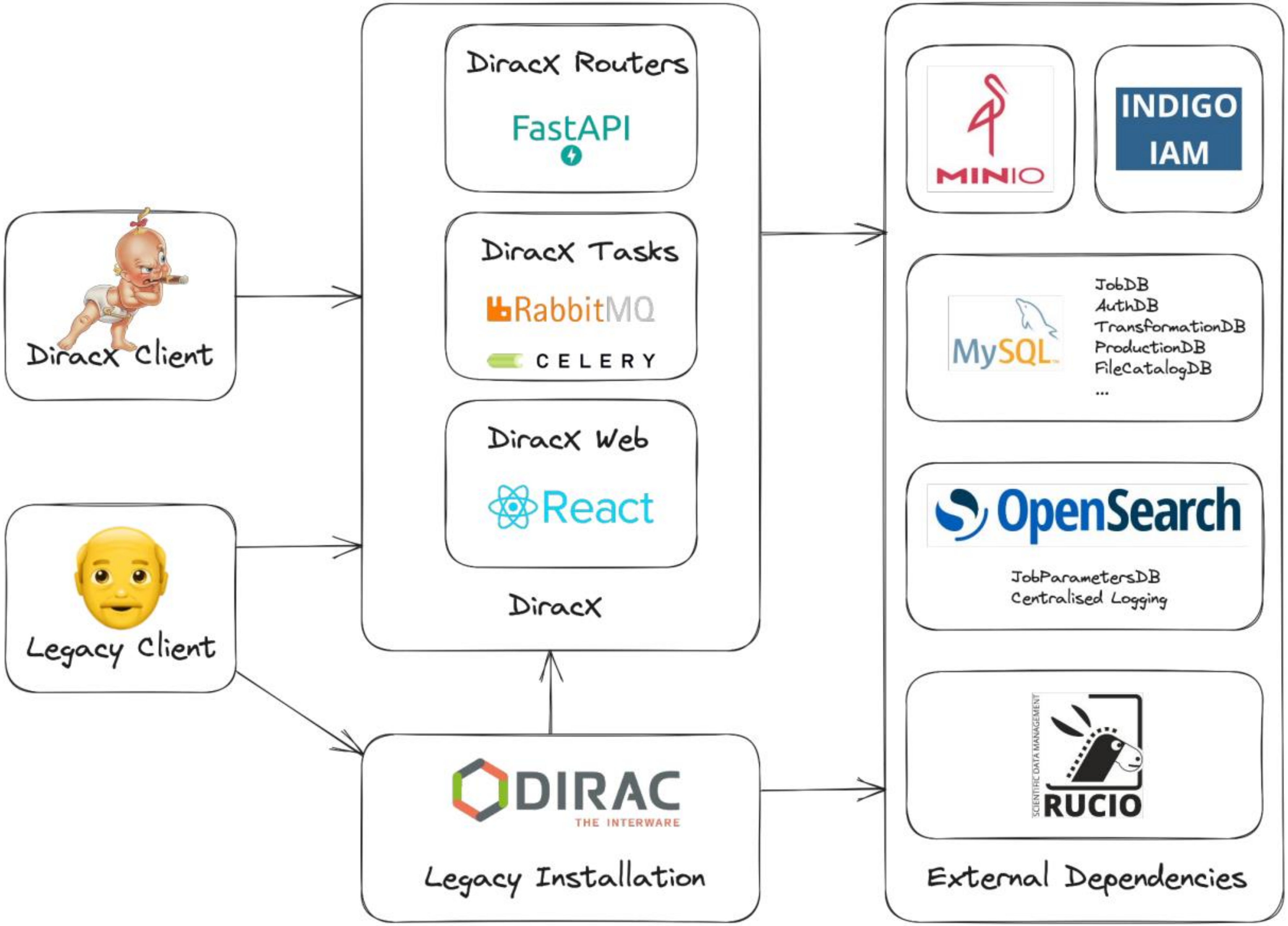
On tokens

DIRAC

THE INTERWARE

In the "Dirac world", tokens can be used for:

- Submitting pilots
  - In DIRAC v8 since a while
- Data access
  - the model is not fully clear yet
- Verifying a user's identity
  - In DiracX tokens will be the only way we will do that
  - Purely internal



DiracX Client

Legacy Client

DiracX Routers

DiracX Tasks

DiracX Web

DiracX

Legacy Installation

JobDB
AuthDB
TransformationDB
ProductionDB
FileCatalogDB
...

JobParametersDB
Centralised Logging

External Dependencies