Status and future plans of DIRAC

Federico Stagni

DIRAC technical coordinator

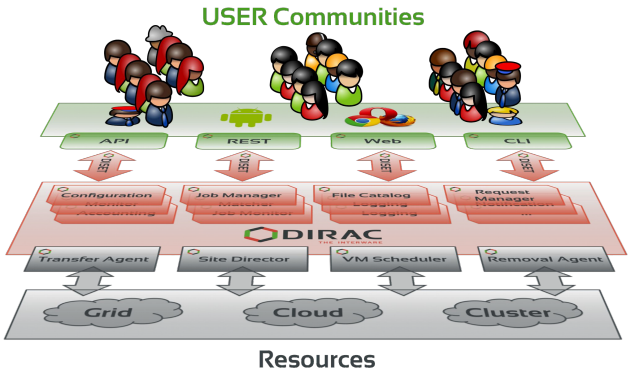
CEPC 23, 26th October 2023

- The previous week we were in Tsukuba for the DIRAC&Rucio Workshop 2023
  - the first of these workshop types
- This presentation builds on the workshop's content
  - I won't talk much about Rucio

What's DIRAC?

Slide that I have been presenting for years, with minimal variations

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

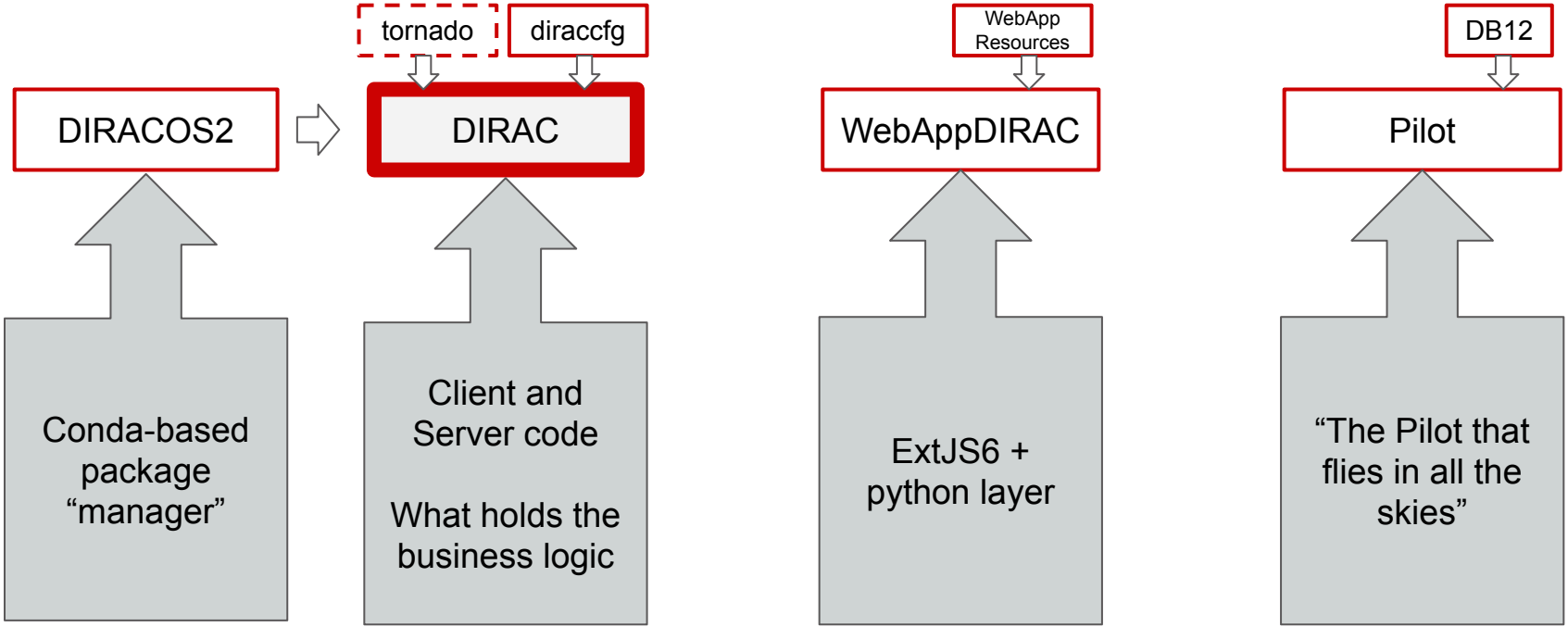


- Developed by communities, for communities
  - Open source (GPL3+), GitHub hosted
  - Python 3
  - Publicly documented, yearly users workshops, open developers meetings and hackathons
  - Deployed mostly via Puppet on VMs (really, not bound to any specific technologies)
- The DIRAC consortium as representing body

Are things about to change?

Yes, but not fully

Today's DIRAC (py3) stack



NB: the py2 stack is deprecated

DIRAC v8.0 (production)

- Abandoned Python 2
- Added support for IdPs (IaM, Check-IN)
  - Can use tokens for submitting pilots to CEs
- Monitoring capabilities expanded
- Expanded support for HPCs
- (computing) clouds support leveraging libcloud

Python3 and PyPI

- DIRAC releases using standard `pip` package manager, found on PyPI
  - extensions had to adapt (already in DIRAC v7.3)
- Deployed in a conda environment created by DIRACOS2 installer
  - which provides Python 3.11
- Support for platforms `ppc64le` and `aarch64` (in addition to the more common `x86_64`) have also been added
  - through conda/mamba

Timeline

Milestone ID	Date	Description	Dependencies	Teams
M.1	Sep 2022	IAM is also in production for ALICE and LHCb.		CERN IT, IAM devs
M.2	Dec 2022	DIRAC versions supporting job submission tokens deployed for concerned VOs (LHCb, Belle-II, …).		DIRAC, LHCb, Belle-II, …
M.3	Feb 2023	VOMS-Admin is switched off for one or more experiments. Prerequisites: • Significant VO admin functionality issues in IAM have been resolved • User registration, group and management have been swi IAM • IAM services are sufficiently • CERN IAM team is sufficientl • Remaining VOMS-Admin use have been moved or will be dropped		IAM devs, CERN IT, experiments
M.4	Mar 2023	HTCondor installations at EGI sites have been upgraded to supported versions > 9.0.x. Prerequisites: • DIRAC versions supporting job submission tokens have been deployed for the concerned VOs (LHCb, Belle-II, EGI catch-all, …) • HTCondor CE supports (adjusted) EGI Check-in tokens • IAM or equivalent in production for ALICE, LHCb, Belle-II, …	M.1 M.2	HTCondor Dev Team, WLCG ops, EGI ops, sites
M.5	Mar 2023	End of HTCondor support for GSI Auth (link).		
M.6	Mar 2023	Some storage endpoints provide support for tokens (at least one per service type).		WLCG ops, storage devs, sites
M.7	Feb 2024	Rucio / DIRAC / FTS have sufficient token support in released versions to perform DC24 using token authorization.	M.6	Rucio, DIRAC, FTS, experiments

https://doi.org/10.5281/zenodo.7014668

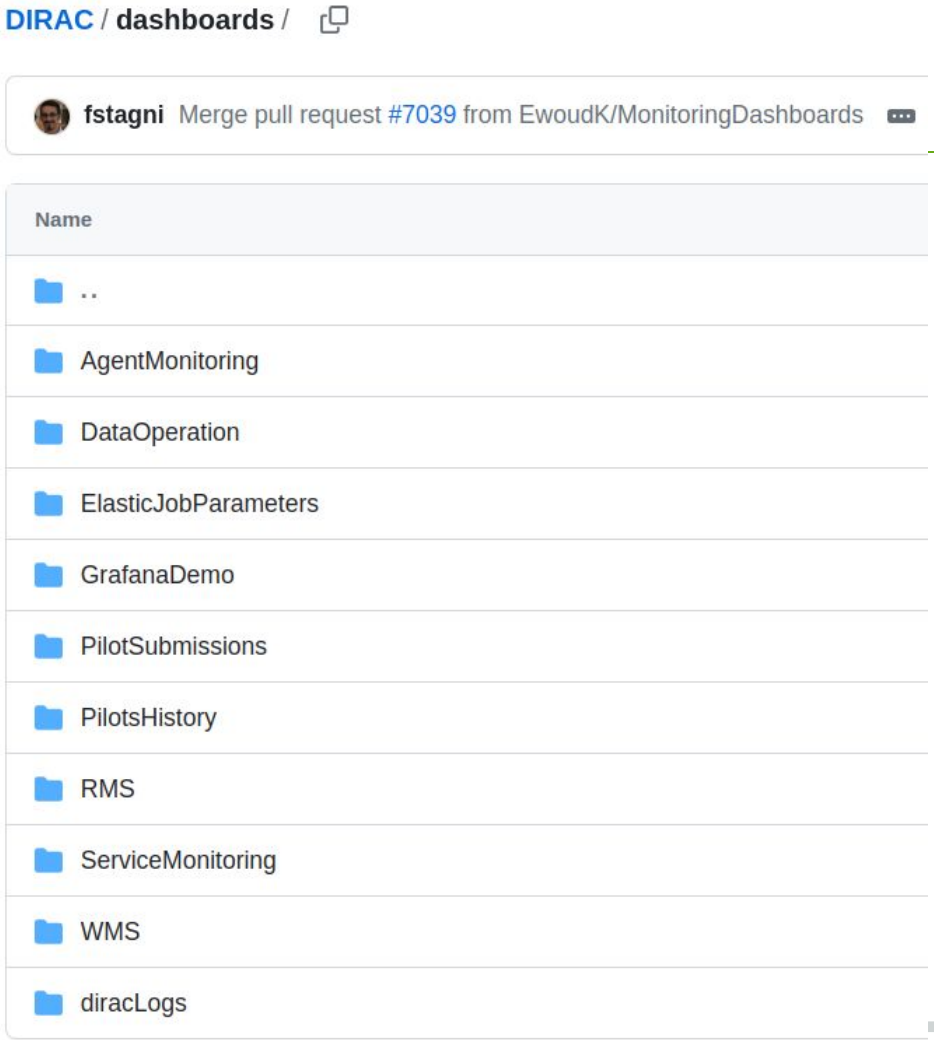
Tokens support

Basically: trying to respect the WLCG timeline

Interfacing with IAM and EGI Check-IN IdP

DIRAC v8 adds `client_credentials` flow for submitting pilots

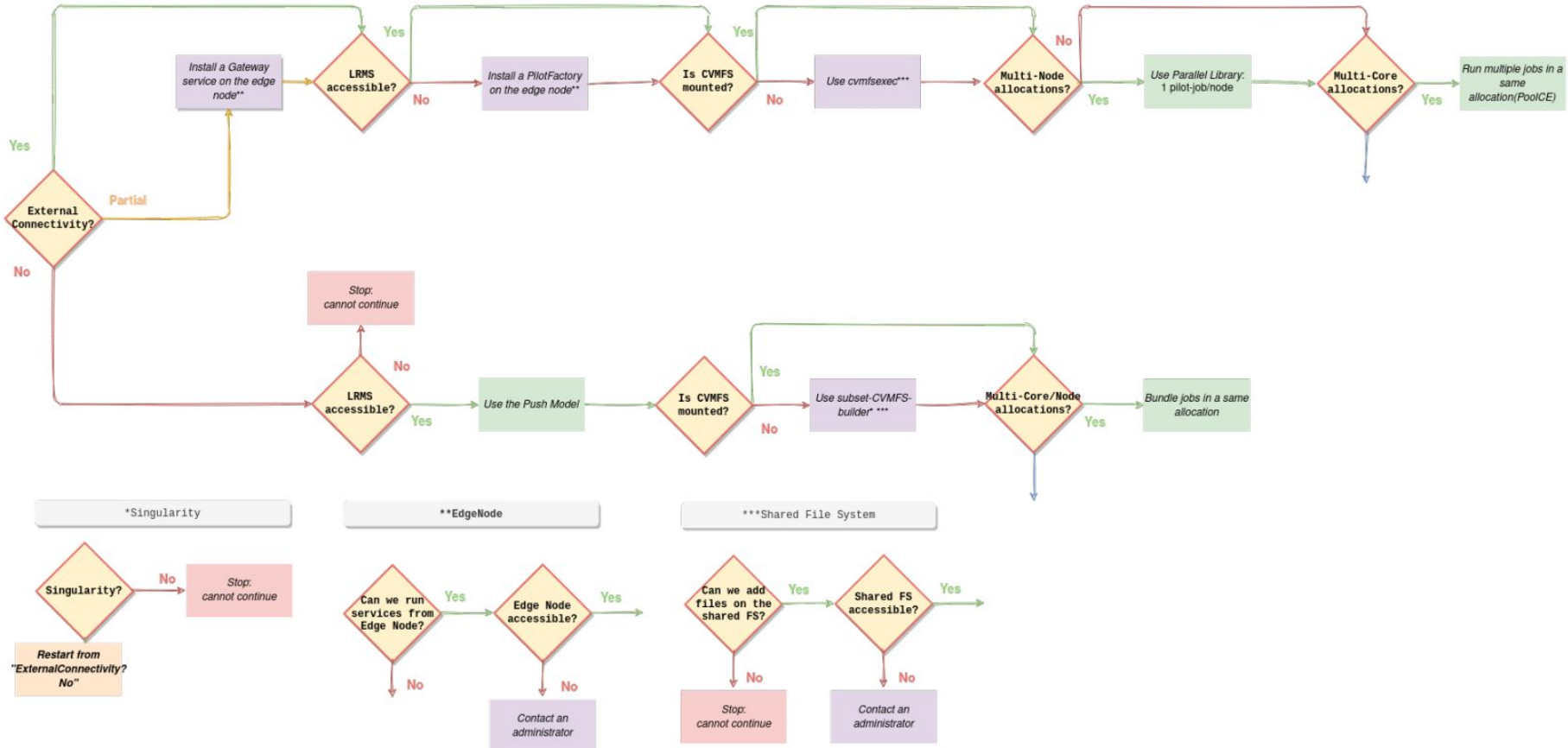
FTS only?



Monitoring

- Added support for OpenSearch (ElasticSearch support was already there), which also becomes the favourite option
  - dropped ES6 support
- Added several OpenSearch indexes that can be filled in
- Added dashboard definitions for Kibana and grafana
- removed `gMonitor` and the `Framework/Monitoring` service (“ActivityMonitoring”)

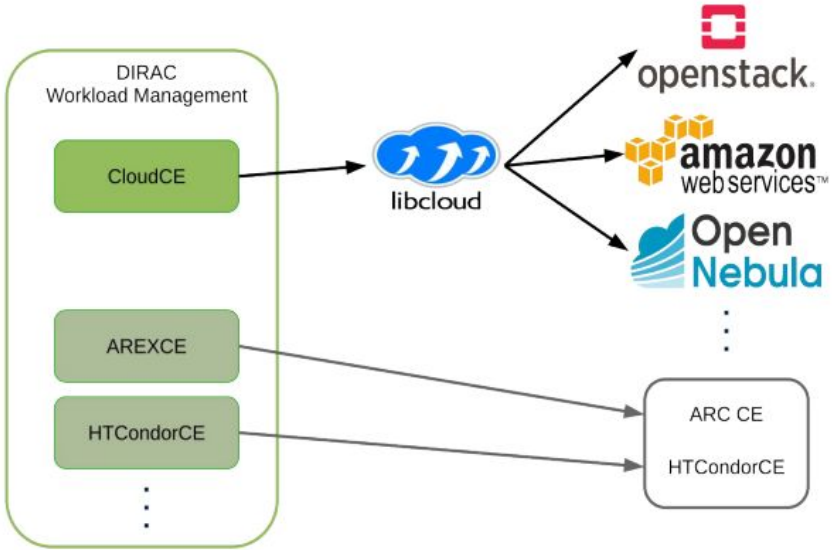
HPCs: choosing the right approach



Cloud CE

CloudCE: Not so special anymore.

- Inherits from DIRAC ComputingElement
- Instead of communication with a grid compute element, the code calls the respective libcloud interface with the correct parameters/credentials
- The pilot payload script and data are added as instance metadata in cloud-init format; this allows any image containing cloud-init to decode and start the DIRAC pilot bootstrap scripts.
- **We pride ourselves in LOC removed :-)**



DIRAC v9.0

- Postponed to Jan 2024
- Abandoning the concept of “Setup”
  - several changes/simplifications at CS and DB level
- The last of DIRAC releases!

DIRAC issues

- complex, with high entrance bar
  - got better dropping python2 compatibility
- somewhat cumbersome deployment
  - got better dropping python2 compatibility
- late on “standards”
  - http services
  - tokens
  - monitoring
- “old”-ish design (RPC, “cron” agents…)
- not very developer-friendly
  - rather un-appealing/confusing, especially for new (and young) developers
- multi-VO, but was not designed to do so since the beginning
- no clear interface to a running DIRAC instance

The list can go on

- the WebApp is highly custom, and somewhat un-maintainable
  - with an intermediate python layer
- runsv is “dead”, we create the RPMs…
- DIRAC’s plotting is “old-ish”
- Moving to JSON serialization quite painful
- Upgrades are not always easy, and sometime scary
- …

- we have been accumulating problems for years
- out there the world evolved in different directions (e.g. REST APIs)

Some DIRAC developments

- Done: Python 3
  - py3 clients supported since version 7.2 (pip installable)
  - py3 server supported since version 7.3
  - py2 support ended with 8.0 (released last week)
    - with some obvious exceptions of part of pilots code
- Done: ES/kibana/grafana dashboards
- Ongoing/advanced: `dips://` → `https://`
  - `dips`: DIRAC proprietary protocol for RPC calls
  - `http`: based on tornado
  - most DIRAC services already available using HTTP
    - we said that `http` would be the default for all the DIRAC services from version 9.0
- Ongoing: token support, and IdP (IaM, Check-in)
- Ongoing: running on kubernetes (goal: define a *helm* chart)
- Started: using celery and RabbitMQ (retiring executors)

Keeping the project successful

- ➔ It felt like we were at the end of a technology cycle.
- ➔ in order to keep the project successful we are creating the neXt dirac incarnation in what we dubbed project “DiracX”[*]

[*] incidentally “X” == 10 (in Roman numbers)

DiracX in just one slide

- A cloud native app
- Multi-VO from the get-go
- Standards-based
- Not a framework

More stable releases

- DIRAC itself is very stable
  - Has a lot of failover mechanisms
  - Gracefully degrades when things fail
- Changing the code is a different story (but improving)
  - Design the entire package to be testable
  - More robustness to

Simpler installation and configuration

- Turn key solution
  - Trival to run a development instance locally
  - Easy for a sysadmin to get a production instance up and running
- Guided upgrade path between versions, should tell you
  - DB changes
  - Configuration changes
  - Deployment changes
- Ideally changes are automated (or wizard-like)

and <u>much more</u>

Easier to maintain extensions (

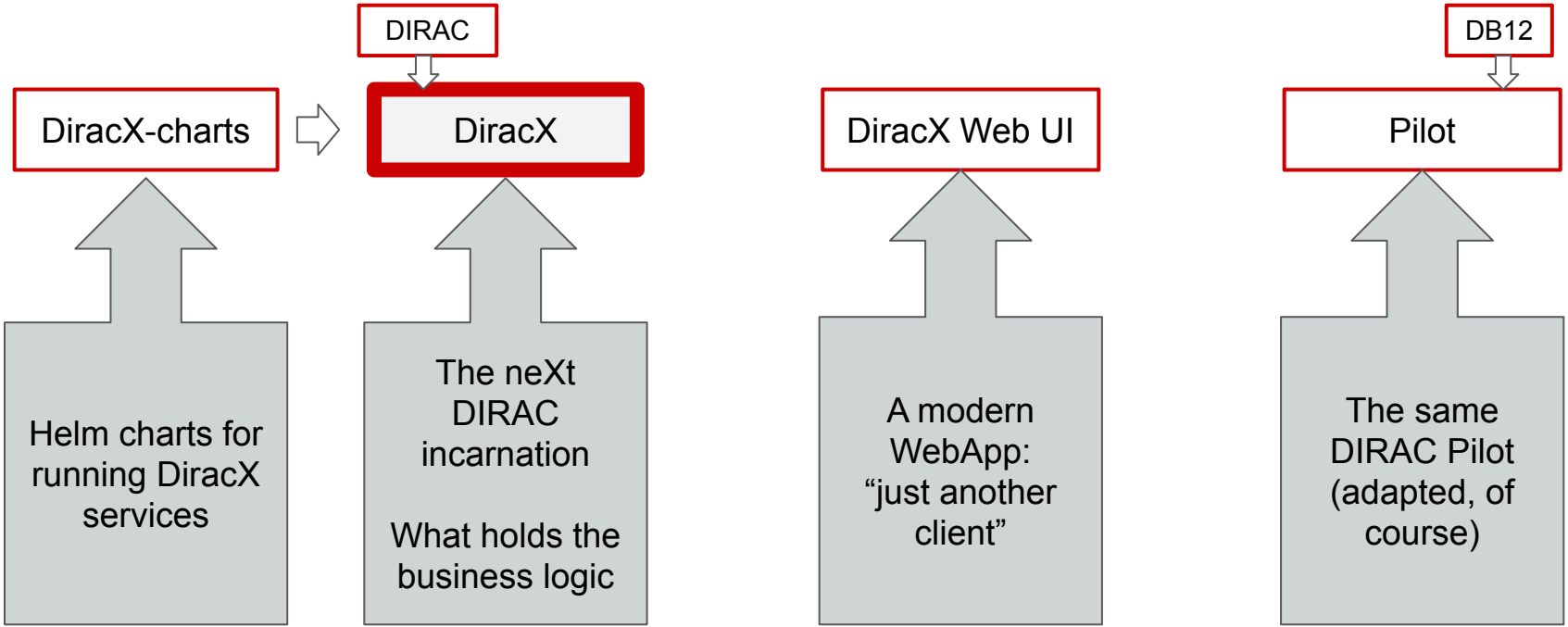
“We worry about catching all the changes”

- Extensions are currently tightly coupled to DIRAC
  - Can modify just about anything
  - Sometimes even overriding private methods!
  - Not sustainable
- Need a clearer interface of what is extendable
- Make a smoother path to maintain extensions by design

More accessible to new developers

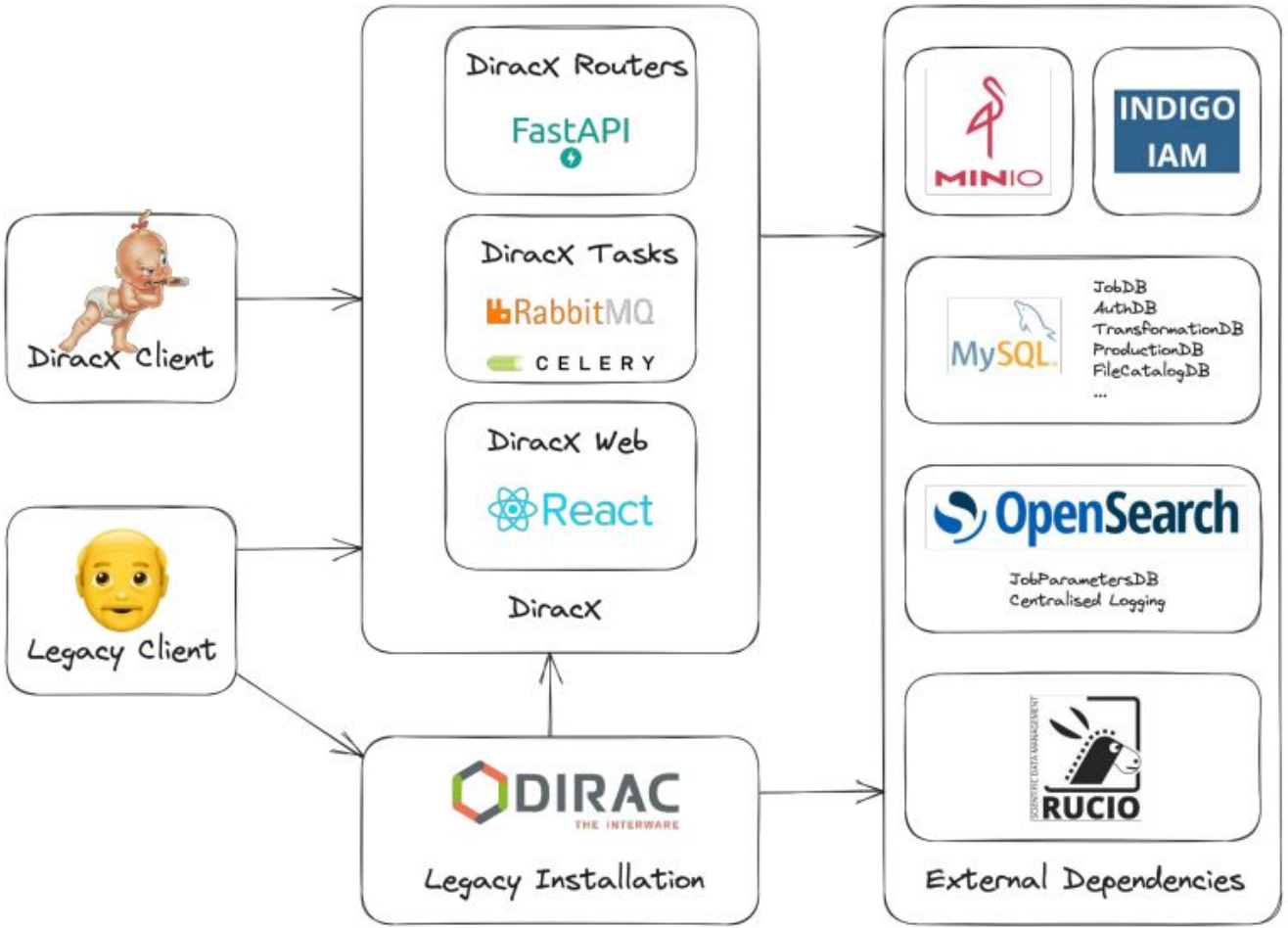
- Our fields have a strong bias towards
  - inexperienced developers
  - short contracts
- Needs to be as accessible as possible

(in dev) DiracX stack

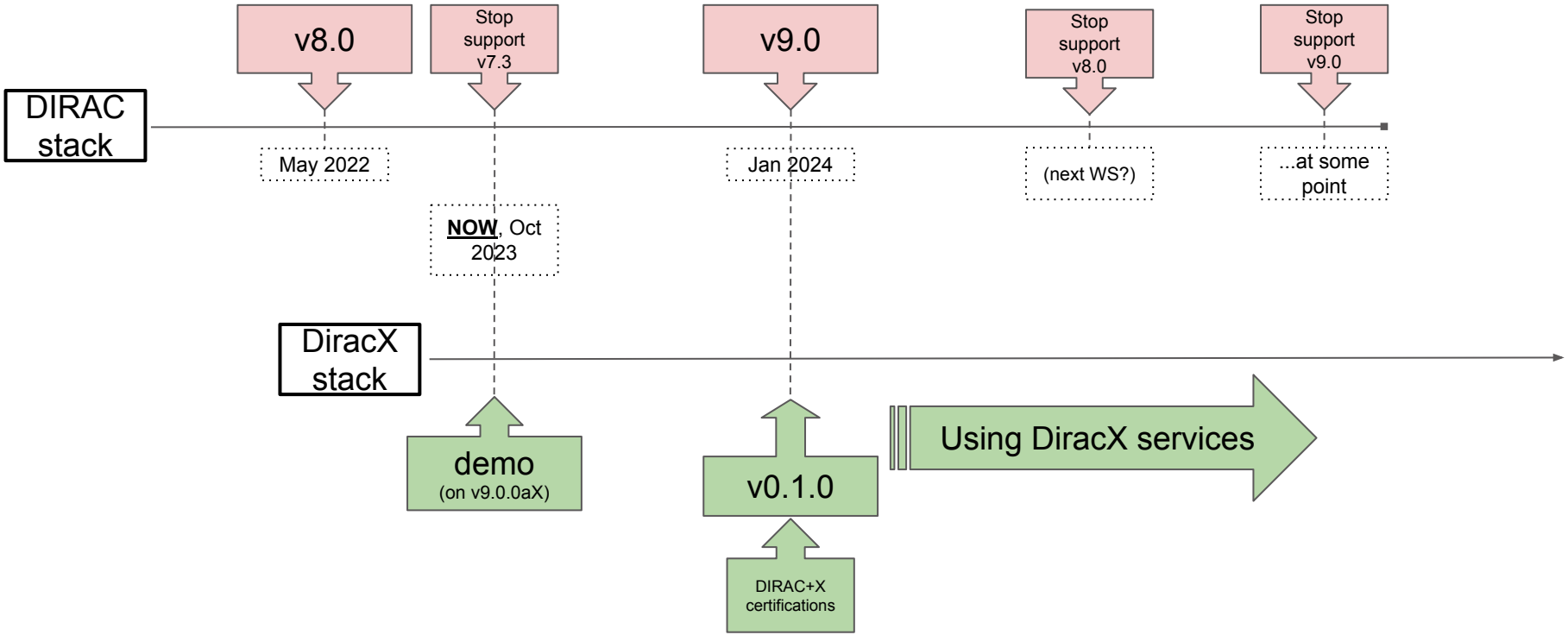


Architecture

A transitioning plan is laid out (see backup slides)



Timeline


v8.0
Stop support v7.3
v9.0
Stop support v8.0
Stop support v9.0
DIRAC stack
May 2022
Jan 2024
(next WS?)
...at some point
NOW, Oct 2023
DiracX stack
demo
(on v9.0.0aX)
v0.1.0
Using DiracX services
DIRAC+X certifications

Rucio?

Rucio + DiracX

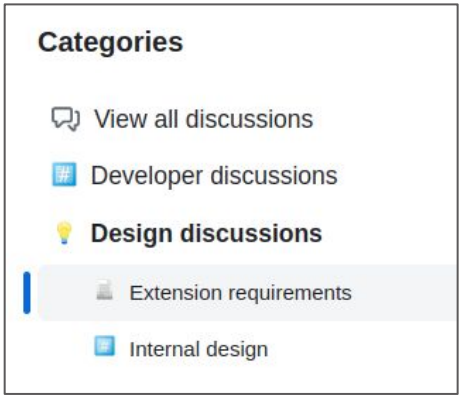
- DiracX is a good opportunity to push forward better Dirac/Rucio integration. Here is a list of topics that can be addressed :
  - Integration of Rucio subscriptions into DIRAC (e.g. possibility to define a subscription in DIRAC CS)
  - Dataset lifetime management
  - Common monitoring (based on ELK stack)
  - Tokens : How Rucio and DiracX can play together ?
  - Introduce some unit-tests
- docker-compose/helmcharts to start a DIRAC-Rucio instance

To conclude

Exciting and busy time

- Rewriting DIRAC
  - WMS functionalities will come first
  - you are very welcome to come onboard
  - your input is needed:

    https://github.com/DIRACGrid/diracx/discussions



- DIRAC v9 will be the bridge for getting there
  - We'll try to ensure stability as much as possible
- We hope CEPC computing will follow in the steps of BES3 and Juno and use Dirac(X)

Questions?

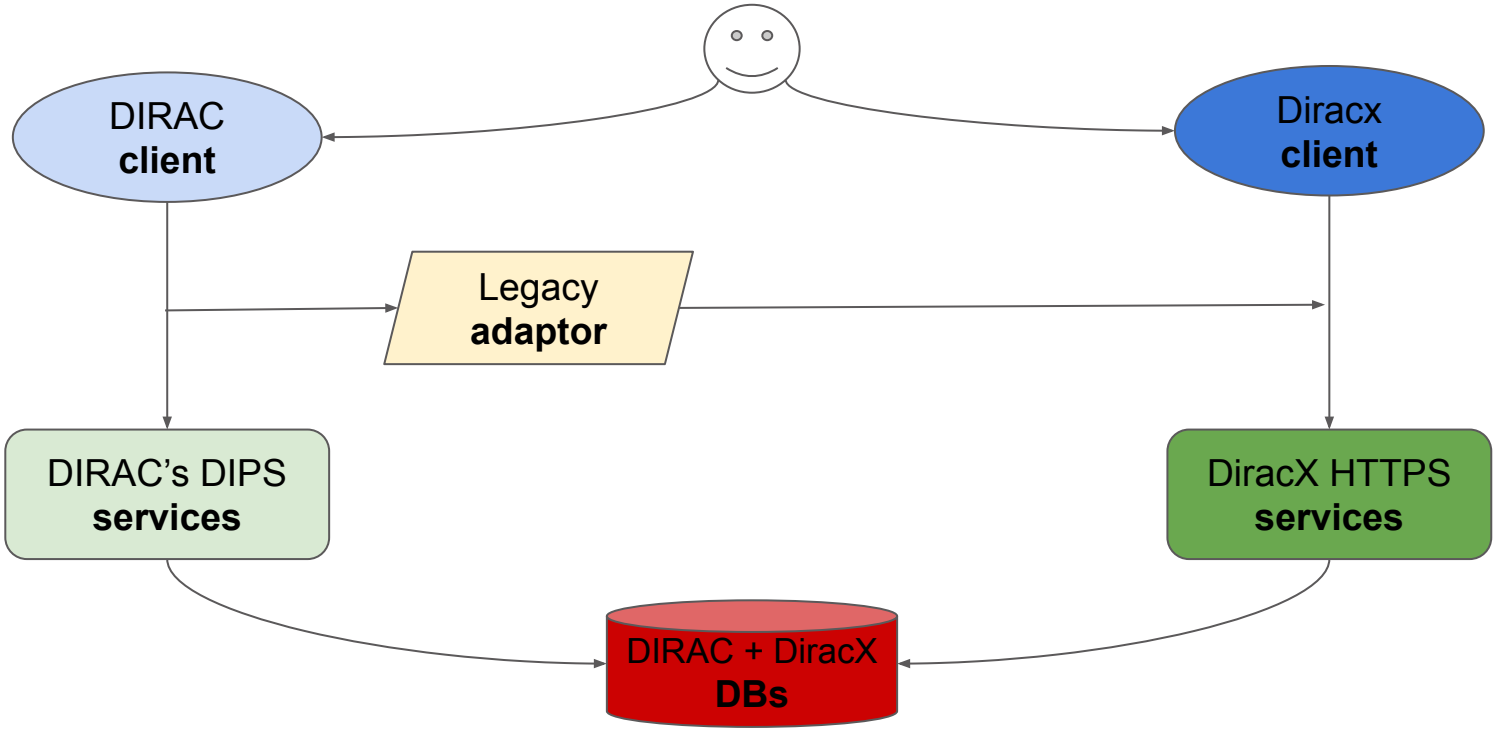
https://github.com/DIRACGrid

- DIRAC's doc: dirac.readthedocs.io
  - including code documentation
- DiracX's doc https://github.com/DIRACGrid/diracx/tree/main/docs
  - We might use RTD also for DiracX
- Dev+Ops+general questions:
  - DIRAC github discussions
  - DiracX github discussions
    - DiracX-Web discussions
    - for speedy communications: https://mattermost.web.cern.ch/diracx/

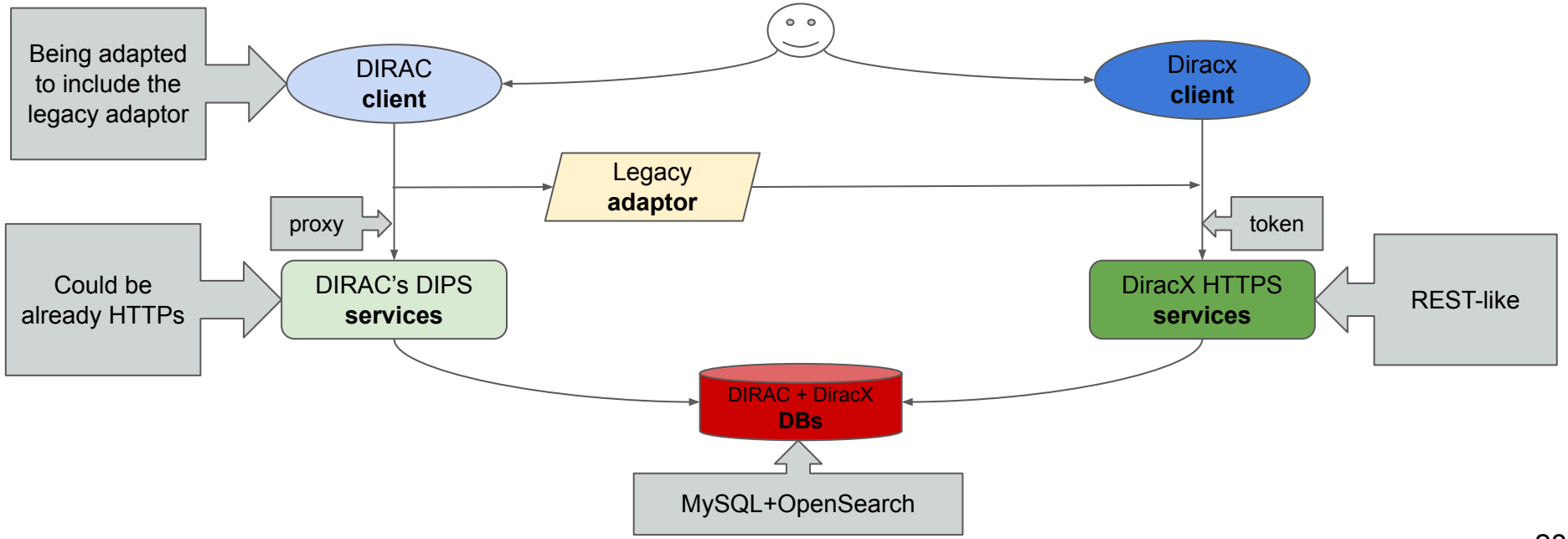
Backup

From DIRAC to DiracX

Transitioning (services)



Transitioning (services)



Transitioning
(agents + executors)

Python **celery** + **RabbitMQ**

https://docs.celeryq.dev/en/stable/getting-started/introduction.html

RabbitMQ™

https://www.rabbitmq.com/

NB: we have not yet started coding for this!

Transitioning from DIRAC agents and executors to DiracX tasks should be easy and straightforward

DIRAC
THE INTERWARE

Q1+Q2
2024

Transitioning stages
(extreme summary)

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