We were in KEK, Tsukuba, Japan for the “DIRAC & Rucio Workshop 2023”
  - the first of these workshop types

This presentation builds on the workshop’s content
Overview

- 44 Participants
  - In-person, no remote participation (except for one presenter)
- 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
- Recordings of all talks available soon!
- Thanks a lot to Ikuo Ueda, Takanori Hara, and team!
Workshop agenda

- One combined workshop, not two workshops
- Sessions
  - 3x Community & User talks
  - 8x Technology talks
  - 6x Q&A, Hackathons, Tutorials
- Welcome drink on Monday
- Workshop Dinner on Thursday
- Fantastic coffee & tea breaks
Why a combined workshop

- Several experiments are interested in both DIRAC and Rucio. A few are already using both of them in production.
  - Belle2 is one of them (we were in KEK)
- Check for possible further collaborations.
- Mutual developers’ interests.
Community & User talks

- Belle 2
- LHCb
- ATLAS
- EGI
- CMS
- CTAO
- GridPP
- Juno/BES3
- Fermilab/DUNE
- JINR
- Rubin/LSST
Technology talks 1/2

● **Tokens!**
  ○ Tokens in Rucio
  ○ Security profile and tokens in DiracX
  ○ Interacting with CEs using tokens in DIRAC
  ○ Rucio ↔ DiracX tokens discussion

● **Deployment**
  ○ Rucio deployment
  ○ DIRAC & DiracX deployment
  ○ Monitoring DIRAC
  ○ Monitoring Rucio
Technology talks 2/2

● HPCs & Clouds
  ○ Running jobs on HPCs with DIRAC
  ○ Running jobs on clouds with DIRAC
  ○ Cloud storage handling with Rucio
  ○ Running productions in DIRAC

● WebUIs
  ○ Rucio WebUI
  ○ Dirac(X) WebUI

● Transfers
  ○ FTS
  ○ Rucio and Torrents R&D
DIRAC
What’s DIRAC?

- 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](https://github.com) 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

---

**Slide that I (Federico) have been presenting for years, with minimal variations**

Are things about to change? Yes, but not fully
Today’s DIRAC (py3) stack

- DIRACOS2
- DIRAC
- WebAppDIRAC
- Pilot

Conda-based package “manager”
Client and Server code
What holds the business logic

ExtJS6 + python layer

“The Pilot that flies in all the skies”

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, atm, 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
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

**Timeline**

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

**DOI:** https://doi.org/10.5281/zenodo.7014668
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
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
  ○ this is, effectively, the main difference with DIRAC
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
  - Trivial 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)

Easier to maintain extensions on

“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

DiracX-charts
- Helm charts for running DiracX services
  - Already in use

DiracX
- The neXt DIRAC incarnation
  - What holds the business logic
  - A demo was shown

DiracX Web UI
- A modern WebApp: “just another client”
  - Was part of the demo

Pilot
- The same DIRAC Pilot (adapted, of course)
Timeline

**DIRAC stack**
- v8.0: May 2022
  - Stop support v7.3: NOW, Oct 2023
- v9.0: Jan 2024
  - Stop support v8.0
  - (next WS?)
  - ...at some point

**DiracX stack**
- v0.1.0: Using DiracX services
  - DIRAC+X certifications
- demo: on v9.0.0aX
Next workshop

- 10th DIRAC Users’ workshop
- Lyon, France
- May or June 2024 (date will be confirmed ASAP)
- 3 days
Rucio
Organisation Update

● Advisory board established last year
  ○ Long-term priorities and plans of communities
  ○ Advise on resource and person-power situation within the Rucio project
  ○ Advice on collaboration on funded projects

● Very successful and useful kick-off of the RAB

● Three SIGs
  ○ Metadata
    ■ → Dec-2023, Rob Barnsley
  ○ Quality of Service
    ■ → Dec-2023, Doug Benjamin
  ○ Tokens
    ■ → Mar-2026, Dimitrios Christidis
Personpower situation

- As with most projects, personpower situation is always fluctuating
  - More recently though, several senior experts took up additional responsibilities in their collaborations and will reduce Rucio development time
  - New people coming in over the next months, but cannot immediately replace senior experts
  - **Need to take extra care about long-term sustainability of project**
- CERN Research & Computing sector technical committee
  - Initiated a [PSO](#) on Rucio
  - Two activities
    - CERN IT to participate in DevOps support for ATLAS & CMS
    - Establishing a Rucio reference data management service at CERN (For SMEs and other sciences)
  - RCS Steering Committee approved the PSO with high priority
Communication

- Migration from Slack → [CERN Mattermost](https://mattermost.com) last year
  - Data privacy & Message retention reasons
  - 475 users on Slack (Feb 2023)
  - 200 users on Mattermost (Nov 2023)

- eMail lists
  - Simplifying channels of communication
  - All google groups closed → All support channeled via Mattermost chat
  - [rucio-news@cern.ch](mailto:rucio-news@cern.ch) news mailing list (<10 mails per year)
    - Replaces [rucio-announce@cern.ch](mailto:rucio-announce@cern.ch)
Technical news

- Switch to semantic versioning
  - 1.32.0 became 32.0.0
- Release plan
  - Continuing with three major (named) releases per year
  - One Long Term Support (LTS) release per year
- From Rucio 32 LTS we require Python 3.9+
- New legacy client compatibility policy
- Switched all official Rucio containers from CC7 to Alma9
Work program 2024 and beyond

- Metadata
  - Will play an increasingly important role in Rucio
- Scalability
  - Recent changes made to the conveyor should be brought to other daemons
- Housekeeping
- Turn-key software / feature conciseness
  - Parts of Rucio still heavily depend on semi-external pieces of code
- Documentation
  - Setup tutorials, newcomer tutorials, etc. - Getting better, but still long way to go
- Tokens!
- DIRAC integration
  - Based on input from this workshop!
Rucio workshop
at SDSC
Sep 30 - Oct 4 2024

Frank Wuerthwein
Director, San Diego Supercomputer Center
Professor, UC San Diego
20min walk to some of the most stunning ocean views of San Diego
Lot’s of hotels rated 4+ on google at all kinds of price points.

Reachable by public transportation:
La Jolla by bus (33min)
Del Mar by bus (26min)
Downtown by tram (40-60min)
(blue line all the way to Mexico)

Though, this is the USA, it’s a car country.

Average high in January ~ average low in August
To conclude,
and what’s next for DIRAC + Rucio
DIRAC+Rucio: next steps

- Documentation to interconnect DIRAC and Rucio
  - Needs to be defined for DiracX
- Include Rucio catalog unit-tests into DiracX
- Better integration of Rucio in DiracX data manager
- Multi-VO between Rucio and DiracX needs special attention
Summary

- DIRAC and Rucio are used by several communities
- Their integration is already a reality for some of them
- Stronger interconnection and common planning are essential for future evolution
- This workshop was a major milestone and setup for future collaboration
  - Stay tuned!
Questions?
Backup
DIRAC: 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
A transitioning plan is laid out (see backup slides)
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

- Install a Gateway service on the edge node**
- Install a PlotFactory on the edge node**

- Is CVWS mounted?

- Use centralized***

- Multi-Node allocations?

- Use Parallel Library: 1 plot/rb/node

- Multi-Core allocations?

- Bundle jobs in a same allocation

- Installs cannot continue

- LEMS accessible?

- Use the Push Model

- Is CVWS mounted?

- Use subset-CVWS-builer***

- Multi-Core/Node allocations?

- Bundle jobs in a same allocation

- *Singularity

- **EdgeNode

- ***Shared File System

- Can we run services from Edge Node?

- Can we add files on the shared FS?

- Edge Node accessible?

- Shared FS accessible?

- Yes

- No

- Contact an administrator

- Stop: cannot continue

- No

- Yes

- Restart from "External Connectivity? No"
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 :-)