BNL Streaming Orchestration Testbed

Wen Guan, Dmitry Kalinkin, Sakib Rahman, Michel Villanueva, Torre Wenaus, Zhaoyu Yang, Xin Zhao

Nuclear and Particle Physics Software (NPPS) Group, BNL Physics Dept

Wouter Deconinck

University of Manitoba

Takahashi Tomonori

Osaka University

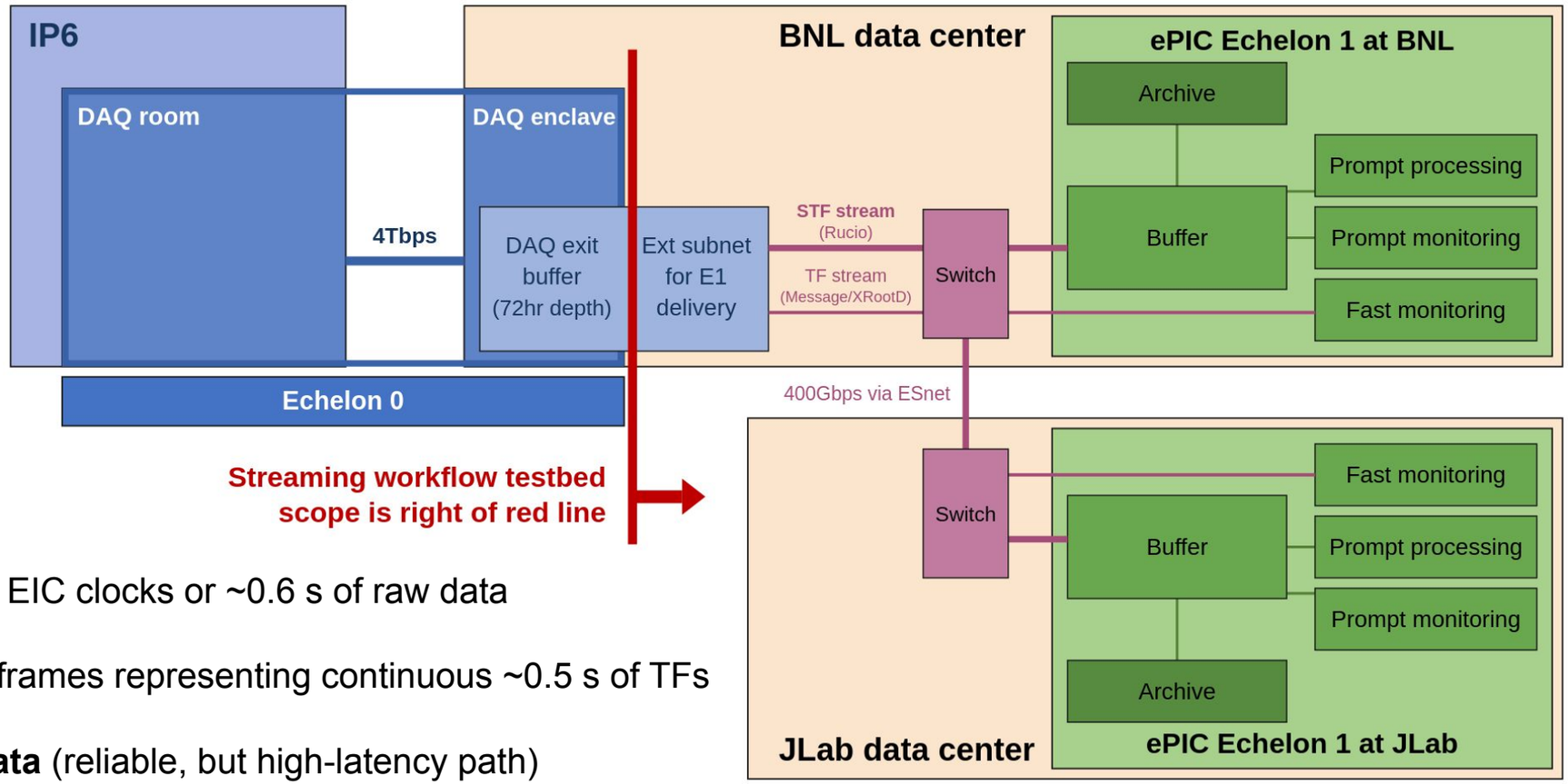
EICUG-ePIC Joint Meeting
July 15 2026

Streaming Workflow Testbed

Designed around goals of developing and prototyping

- Running within constraints: realistic compute and networking
- Define workflows for prompt STF processing, fast STF slice processing and analysis
- Prototyping of interfaces
  - E0-E1: **DAQ interface**, online databases
  - BNL-JLab: **butterfly model**
  - **E1-E2 interfaces**: enabling monitoring, autonomous alignment and calibration
  - First class **AI integration** with awareness of the system state and complete feedback loop for control over it
  - Prototype **STF formats**, realistic **reconstruction** computation in payloads
  - System monitoring: how **system state is exposed** to shifters/analyzers/experts
- Specific choice of core technologies: ActiveMQ, Rucio, PanDA, Django
- Large design space to navigate and lots to prove for ePIC Streaming and Computing Model

Data orchestration: interface to DAQ and Echelons 1 and beyond



TF - continuous recording of 2^{16} EIC clocks or ~0.6 s of raw data

STF - ~2 GB files of ~1000 timeframes representing continuous ~0.5 s of TFs

Rucio for **prompt processed data** (reliable, but high-latency path)

XRootD or message based access for **fast monitoring/processing**
(low-latency, subsampling-capable, but not implementing retention)

Testbed components

Links are to the GitHub repositories

- swf-testbed
  - Testbed umbrella repo: doc, packaging, venv, example agents, service management, test driver, **extensions for complex workflows: orchestration of persistent agents, extensible workflow definitions driven by common runner**
- swf-common-lib
  - Common infrastructure, logging utils, packaging infrastructure
- swf-monitor
  - Django service with full-system browser UI, REST API, ActiveMQ and logging
  - Postgres DB serving the full system via REST
  - Comprehensive **Model Context Protocol (MCP)** service for info and control via LLM
- swf-transform
  - **Interfaces to EICrecon processes**
- swf-panda-workers
  - **Provisions and scales the worker pool through iDDS and PanDA**
- ActiveMQ based agents
  - Operate as **persistent agents ready for work**, controlled by CLI or MCP
  - DAQ simulator **expressing the E0-E1 interface**
  - **STF processing agent:** workflow for end to end PanDA/Rucio processing of STFs
  - **Fast processing agent:** implementing workflow I will describe

The streaming workflow testbed

The first streaming testbed workflows on the common WFMS platform

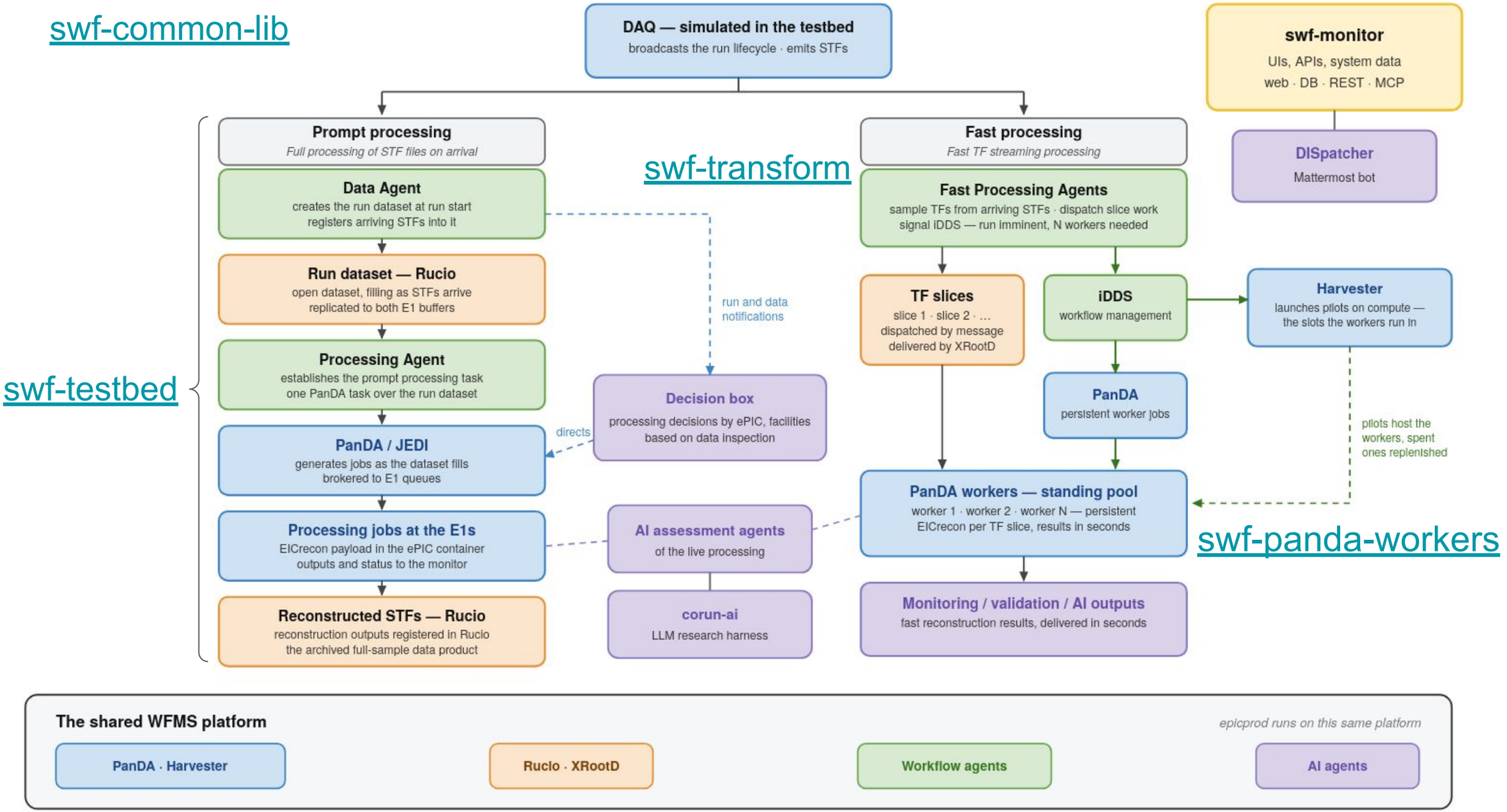
swf-common-lib

swf-monitor

swf-testbed

swf-transform

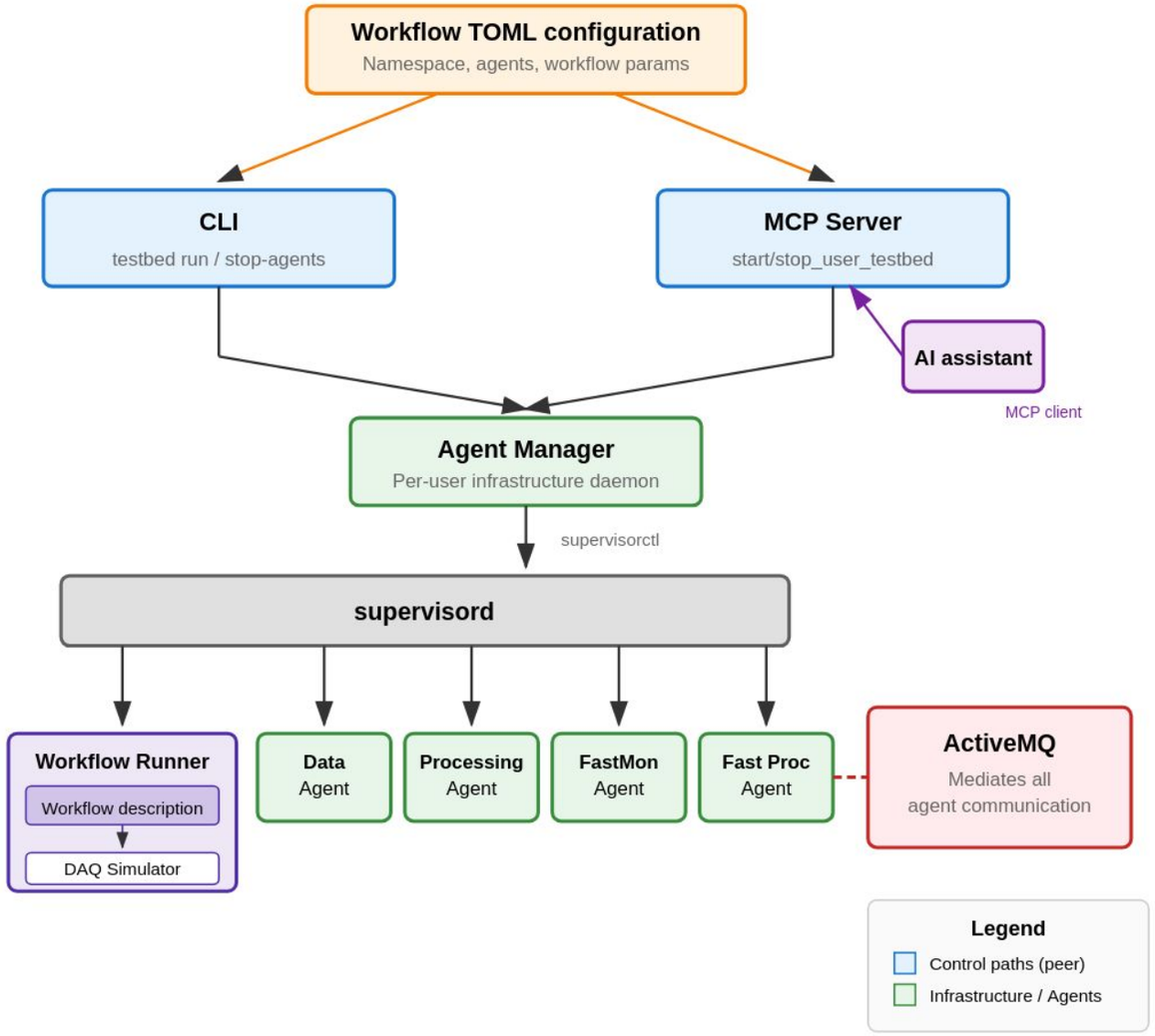
swf-panda-workers



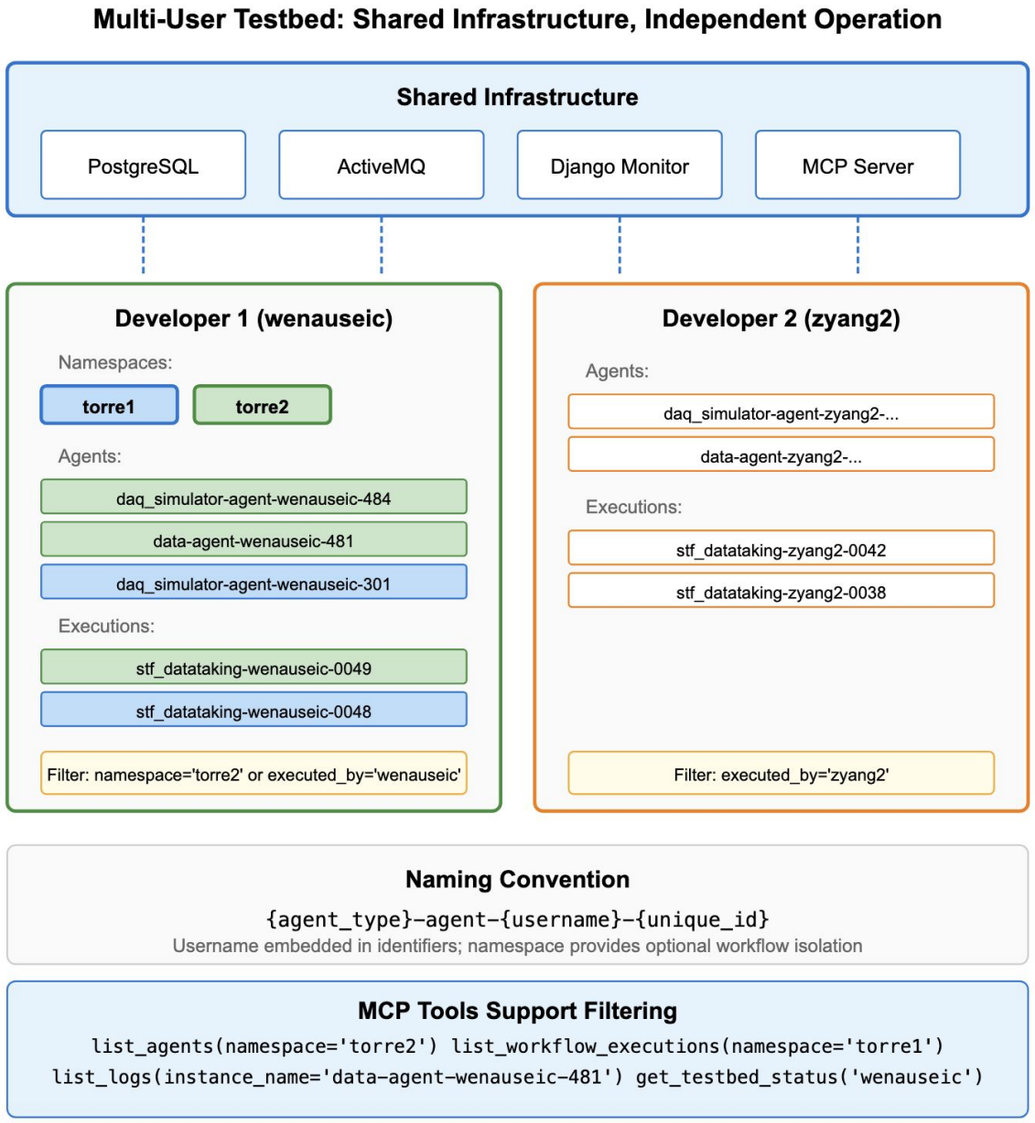
Agent orchestration

Agents are processes with defined responsibilities.

- **suprevisord** manager: instantiates/tears down agents
  - Complete configurations provided as TOML files
  - Can run a subset of the full system
  - Replace individual agents with other implementations
- First-class **MCP integration** allows to control and inspect testbed state with AI agents (interactively or autonomously)

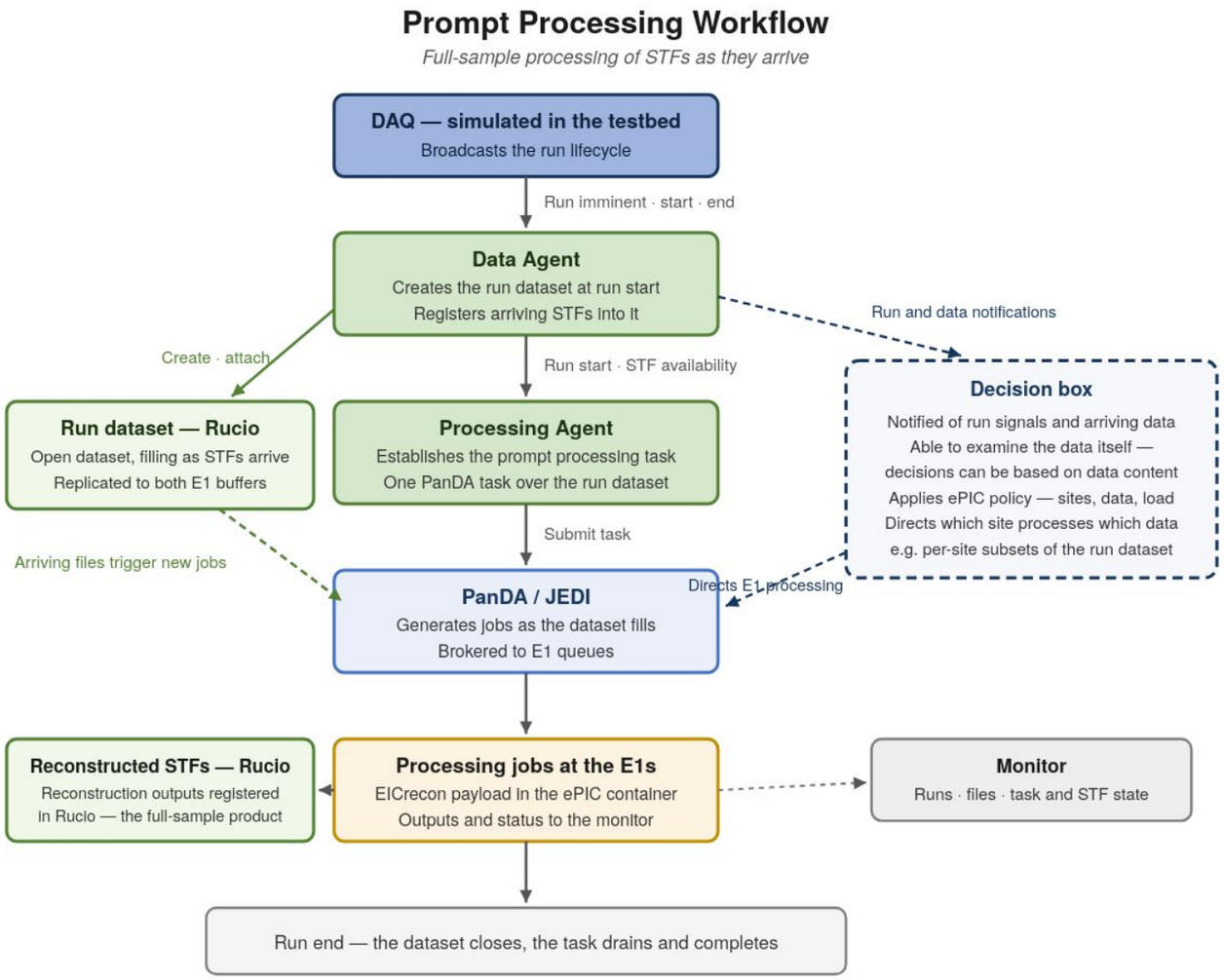


Agent communication



- Agents are connected to **ActiveMQ messaging bus**
- **Multi-user support**: per-user configuration overrides (for running different instances on the same system)
  - Separation by agent (user) name
  - Support for collaborative work with (optionally) shared namespaces
- Interface for **SSE message streaming** allowing messages
  - Successfully used to demonstrate agent-to-agent communication in a Echeloned environment (BNL ↔ Osaka)

Prompt STF file processing workflow



- Workflow runner simulates DAQ activity issuing run begin/end, STF files
- Data Agent implemented with proper Rucio capability
- Run/file registration is carefully accounted and reported in monitoring
- Testing with mock STF data files, but operation with real EICrecon payload was demonstrated
  - Official ePIC DIS 10x100 / 2us timeframes
  - No zero suppression
  - 200 MB/file / 40 events – a bit large
  - ~5s / event for processing – reasonable in terms of work

Prompt STF file processing workflow: Implementation

- Daily workflow execution enables timely detection of issues across the testbed, PanDA, and Rucio
- Full chain submission supported via both CLI and MCP
- Current step: adding the ‘policy box’ as PanDA plugin that will allow filtering in each E1’s processing chain to apply ePIC and facility policy in what’s processed where

```
> Start a prompt processing workflow

✦ The user wants to start a prompt processing workflow, and I have a specific skill available for this task - start-prompt-processing - which I should invoke immediately.

✦ I'll start a prompt processing workflow for you.

 ✓  Skill Use skill: "start-prompt-processing"

    Starts a prompt processing workflow in the swf-testbed system (namespace test-zy, workflow "prompt_processing", config "prompt_processing.toml") and then retrieves and presents an execution summary. Use
    this skill whenever the user asks to start, run, launch, or kick off a prompt processing workflow, even if they don't specify the namespace or config.

✦ Following the skill instructions, I need to start the workflow with the exact specified parameters before proceeding to find the execution ID.

✦ I'll start the prompt processing workflow now.

 ✓  swf_start_workflow (swf-testbed MCP Server) {"workflow_name":"prompt_process

    {
      "success": true,
      "message": "Workflow 'prompt_processing' start command sent to DAQ Simulat
      "workflow_name": "prompt_processing",
      "namespace": "test-zy",
      "config": "prompt_processing.toml",
      "realtime": true,
      "params": {
        "namespace": "test-zy"
      },
      "note": "Workflow runs asynchronously. Use swf_list_workflow_executions to

✦ The workflow has completed successfully. Here's the summary:

                                                                  d the execution ID by listing w
```

Field	Value
Execution ID	prompt_processing-zyang2-0434
Workflow	prompt_processing
Namespace	test-zy
Status	completed
Phase	ended
Run ID	102417
Started	2026-03-26T16:08:58 UTC
Ended	2026-03-26T16:09:23 UTC
Duration	25.0 seconds
STF Count	3
Errors	None

Timestamp	App Name	Instance Name	Level	Message	Module	Function
20260326 12:09:23	base_agent	daq_simulator-agent-zyang2-1266	INFO	Workflow completed: prompt_processing (execution: prompt_processing-zyang2-0434)	workflow_runner	_run_workflow_thread:808
20260326 12:09:17	base_agent	data-agent-zyang2-1264	INFO	Run 102417 status updated successfully	data_agent	update_run_status:451
20260326 12:09:17	base_agent	data-agent-zyang2-1264	INFO	Updating run 102417 status to completed in monitor...	data_agent	update_run_status:443
20260326 12:09:17	base_agent	data-agent-zyang2-1264	INFO	Processing end_run message	data_agent	handle_end_run:288
20260326 12:09:17	base_agent	data-agent-zyang2-1264	INFO	STF file swf.102417.000003.stf registered with ID 4654fc3e-045f-4fad-8c7b-1ef7725e0ee0	data_agent	register_stf_file:498
20260326 12:09:17	base_agent	data-agent-zyang2-1264	INFO	Registering STF file swf.102417.000003.stf in monitor...	data_agent	register_stf_file:471
20260326 12:09:17	base_agent	data-agent-zyang2-1264	INFO	Processing STF file	data_agent	handle_stf_gen:384
20260326 12:09:16	base_agent	daq_simulator-agent-zyang2-1266	INFO	Broadcasted run_end message	workflow_runner	broadcast_run_end:253
20260326 12:09:16	base_agent	daq_simulator-agent-zyang2-1266	INFO	Broadcasted stf_gen message	workflow_runner	broadcast_stf_gen:291

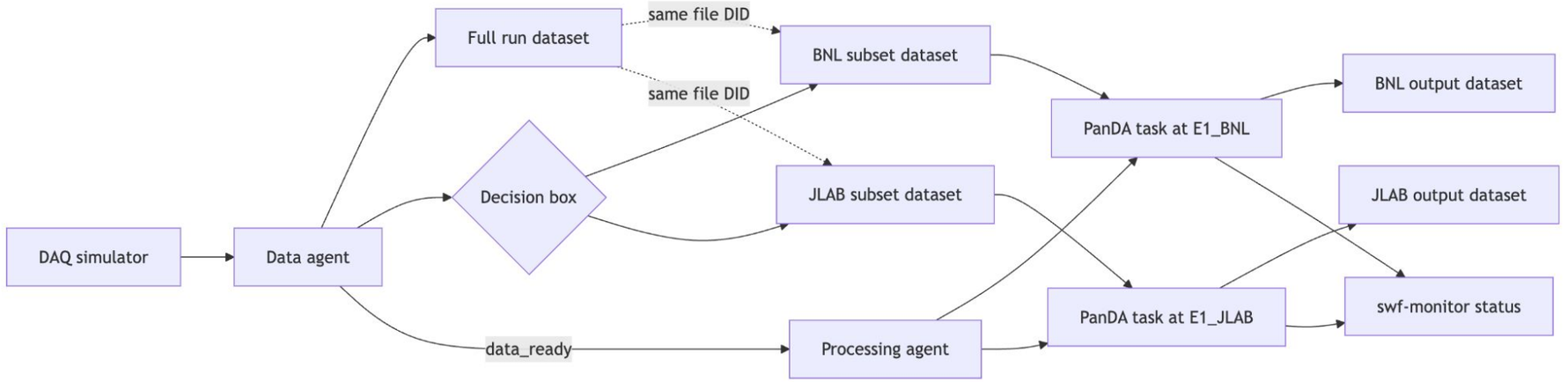
Decision Box for Site-Aware Prompt Processing

Decision box imposes ePIC and facility policies onto PanDA engine

- STF files start in “on-hold” state in PanDA
- Policy box acts to make PanDA generate an active job at a specific E1 site

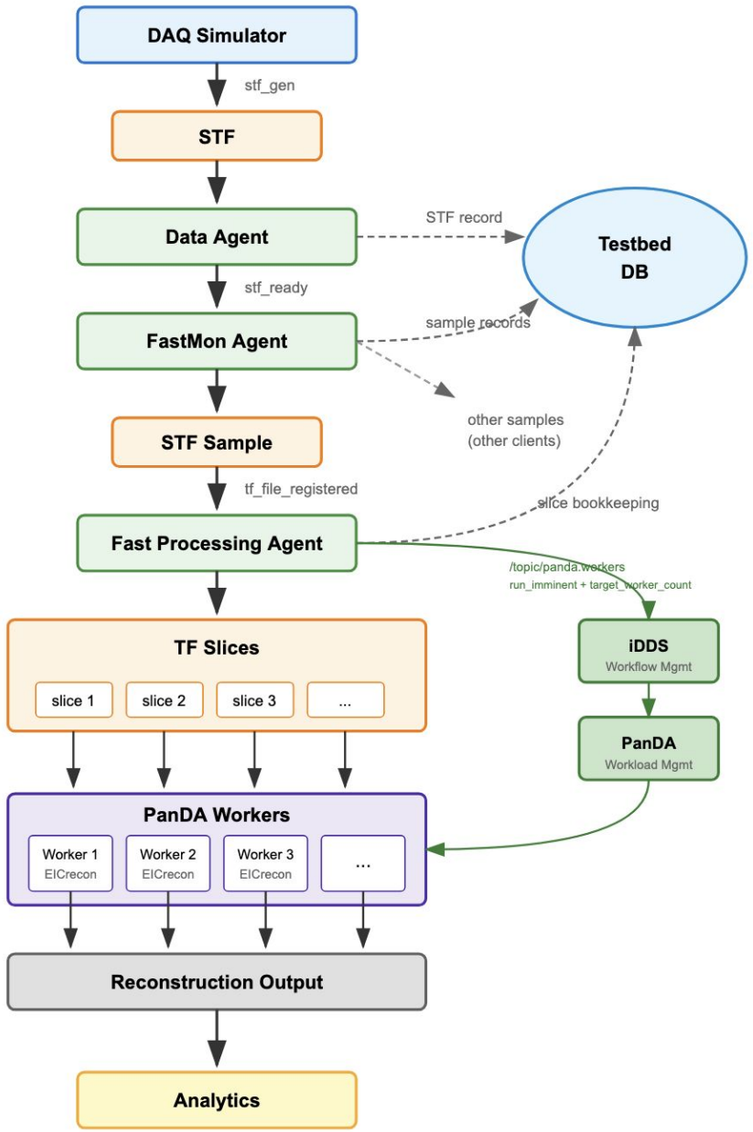
Current implementation (WIP):

- Data agent owns Rucio input membership: full run dataset plus E1-specific subset datasets
- Decision box selects where each STF DID is attached: E1_BNL, E1_JLAB, both, or neither
- Processing agent submits one runUntilClosed=True PanDA task per E1 subset dataset
- PanDA processes only the STF DIDs present in that site’s subset dataset



- The decision is encoded as Rucio dataset membership, so PanDA does not need cross-task decision logic

Fast processing



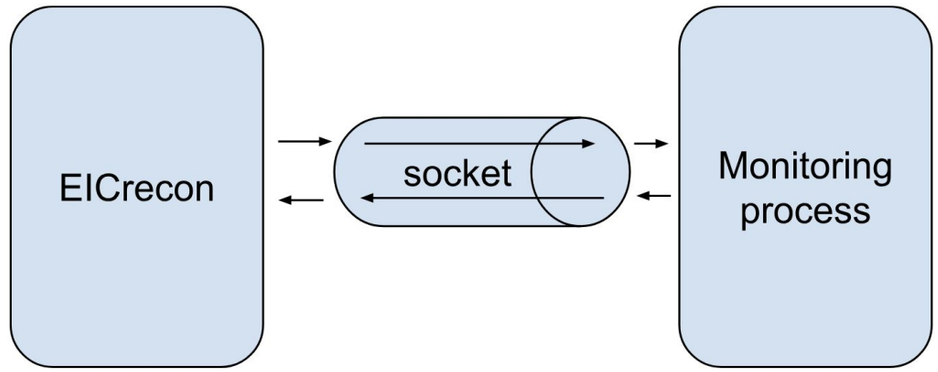
- Message-based fast workflow processing integrated with PanDA
- swf-panda-workers maintains persistent PanDA jobs with iDDS
- swf-transformers payload receives TF slices for reconstruction
- Currently has integrated realistic EICrecon payload
  - Streaming input over XRootD

```
2026-05-27 14:35:00,929 Thread-3       PayloadProcessor        INFO    EPIC version: 26.03.0 (source: filename)
2026-05-27 14:35:00,930 Thread-3       PayloadProcessor        INFO    eicrecon plan: file=root://dtn-
eic.jlab.org:1094//volatile/eic/EPIC//FULL/26.03.0/epic_craterlake/Bkg_Exact1S_2us/GoldCt/10um/DIS/NC/10x100/minQ2=1/pythia8NCDIS_10x100_minQ2=1_beamEffects_xAngle=-0.025_hiDiv_
1.5156.edm4hep.root, nskip=93, nevents=2, output=/tmp/eic_kUDUTNY5/PanDA_Pilot-882911/stf_datataking-wguan2-0649_slice_004.edm4eic.root
2026-05-27 14:35:00,930 Thread-3       PayloadProcessor        INFO    Processing script written to: /tmp/eic_kUDUTNY5/PanDA_Pilot-882911/eicrecon_zm_67kvw.sh
2026-05-27 14:35:00,930 Thread-3       PayloadProcessor        DEBUG   Script content:
#!/bin/bash
#
# eicrecon_process.sh — process a single slice from an edm4hep input file
#                       inside the EIC Singularity container.
#
# Placeholders (replaced at runtime by process_payload):
#   26.03.0   — EPIC campaign version, e.g. 26.03.0
#   root://dtn-
eic.jlab.org:1094//volatile/eic/EPIC//FULL/26.03.0/epic_craterlake/Bkg_Exact1S_2us/GoldCt/10um/DIS/NC/10x100/minQ2=1/pythia8NCDIS_10x100_minQ2=1_beamEffects_xAngle=-0.025_hiDiv_
1.5156.edm4hep.root     — XRootD or local path of the input edm4hep ROOT file
#   /tmp/eic_kUDUTNY5/PanDA_Pilot-882911/stf_datataking-wguan2-0649_slice_004.edm4eic.root    — path for the eicrecon output edm4eic ROOT file
#   93          — number of events to skip (= start index)
#   2         — number of events t
#
set -e

SINGULARITY_IMAGE="/cvmfs/singula

singularity exec "${SINGULARITY_I
set -e
10:35:52.555  [info] Created event pool with level=PhysicsEvent and size=1
10:35:52.555  [info] Arrow topology is:
10:35:52.555  [info]   ------------------------------------------------------
10:35:52.555  [info]          Arrow          Parallel  Direction  Place   ID
10:35:52.555  [info]   ------------------  --------  ---------  ------  --
10:35:52.555  [info]   PhysicsEventSource      0       Input      Pool    1
10:35:52.555  [info]                                  Output     Queue   0
10:35:52.555  [info]   PhysicsEventMap2        1       Input      Queue   0
10:35:52.555  [info]                                  Output     Pool    1
10:35:52.555  [info]   ------------------------------------------------------
10:35:52.555  [info]
10:35:52.555  [warn] To pause processing and inspect, press Ctrl-C.
10:35:52.555  [warn] For a clean shutdown, press Ctrl-C twice.
10:35:52.555  [warn] For a hard shutdown, press Ctrl-C three times.
10:35:52.555  [warn] For worker status information, press Ctrl-Z, or run `jana-status 2948619`
10:35:52.555  [info] Initialized JEventSource 'JEventSourcePODIO' ('root://dtn-
eic.jlab.org:1094//volatile/eic/EPIC//FULL/26.03.0/epic_craterlake/Bkg_Exact1S_2us/GoldCt/10um/DIS/NC/10x100/minQ2=1/pythia8NCDIS_10x100_minQ2=1_beamEffects_xAngle=-0.025_hiDiv_
1.5156.edm4hep.root')
[] [JEventProcessorPODIO] [info] Using 'root' backend for output file: /tmp/eic_kUDUTNY5/PanDA_Pilot-882911/stf_datataking-wguan2-0649_slice_004.edm4eic.root
10:35:52.557  [info] Initialized JEventProcessor 'JEventProcessorPODIO'
10:35:52.557  [info] Initialized JEventProcessor 'JEventProcessorJANATOP'
10:35:52.557  [info] Initializing JWiringService
10:35:52.557  [info] Configuration Parameters
10:35:52.557  [info]
```

Fast processing EICrecon payload

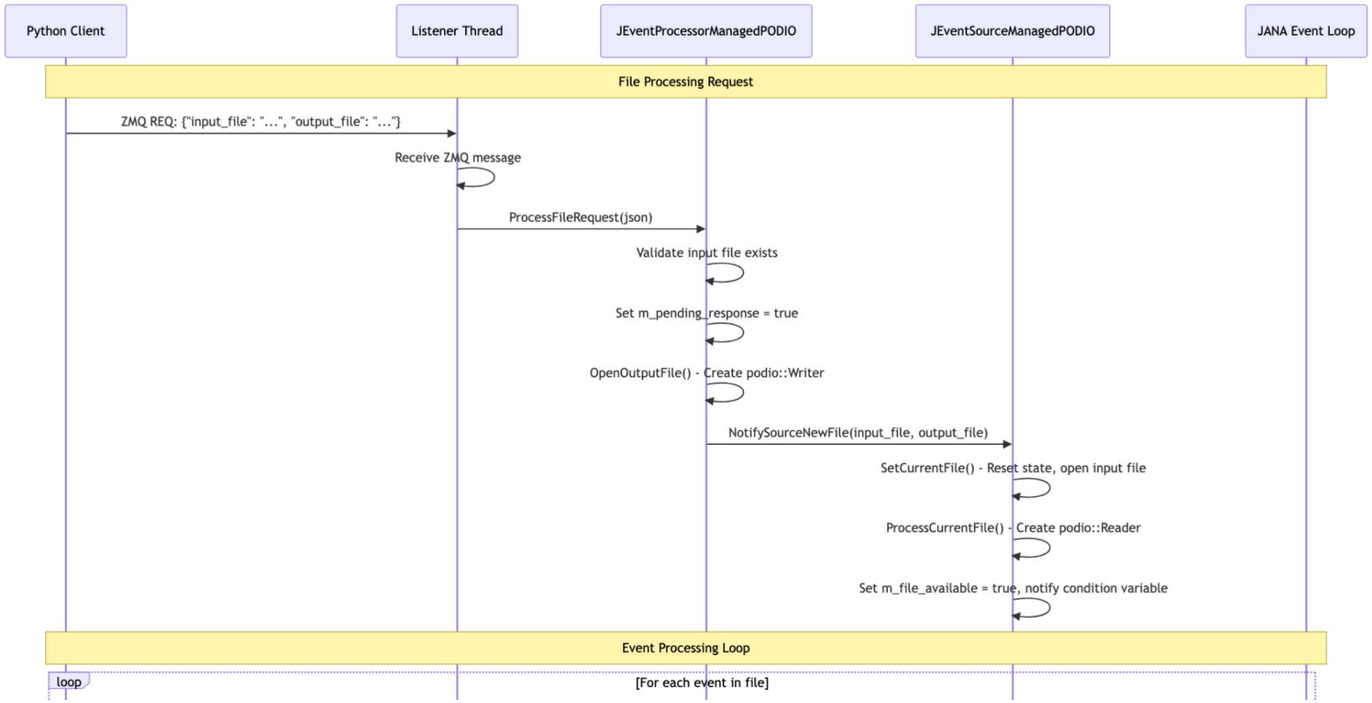
Persistent reconstruction process is needed to provide **low-latency** reconstruction.

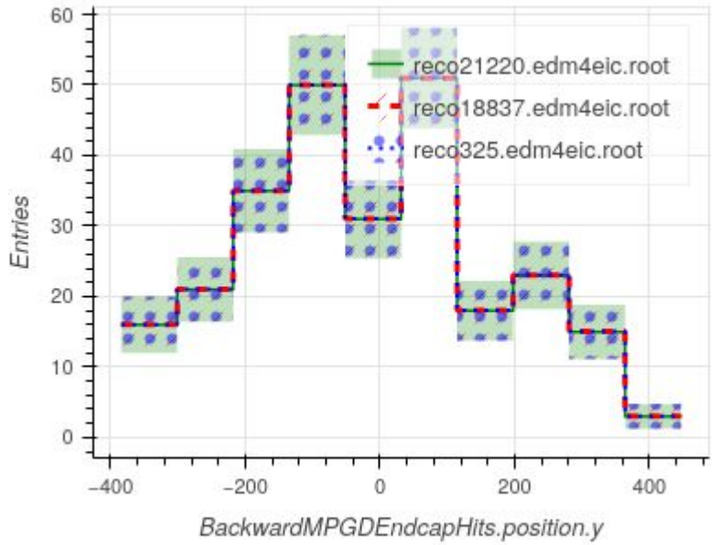
[Design document] for payload with key decisions



Internals of our persistent job

Implementation available [EICrecon#2604]. See more in [Reco WG mtg. presentation].

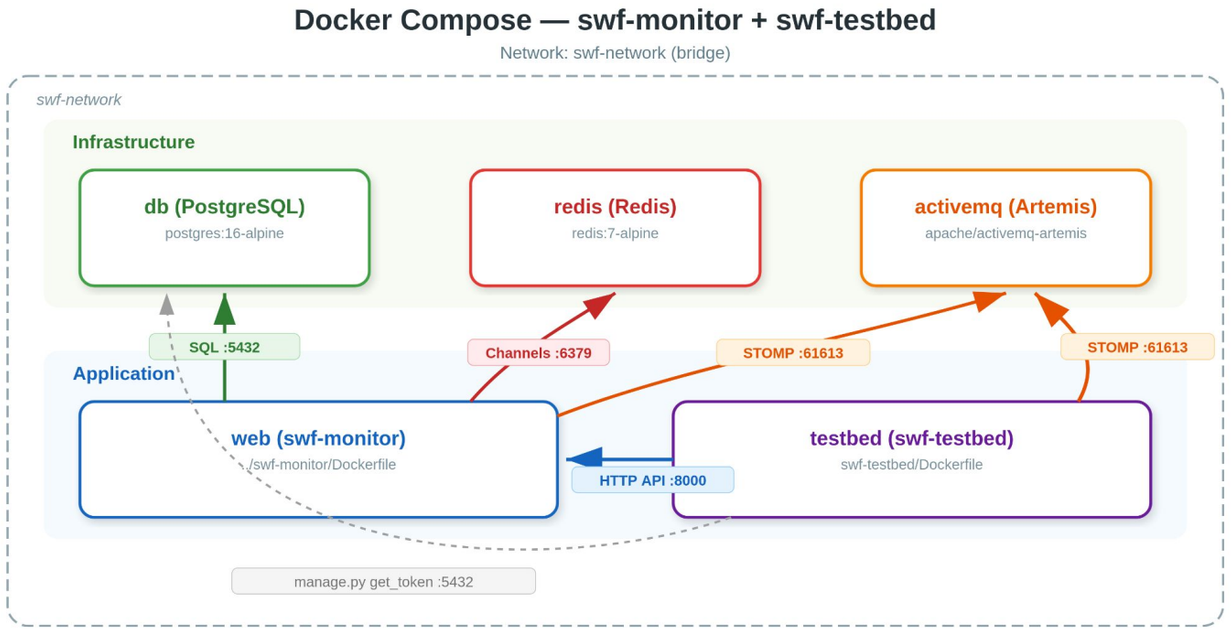




A test on the single input queued into EICrecon multiple times showing identical outputs

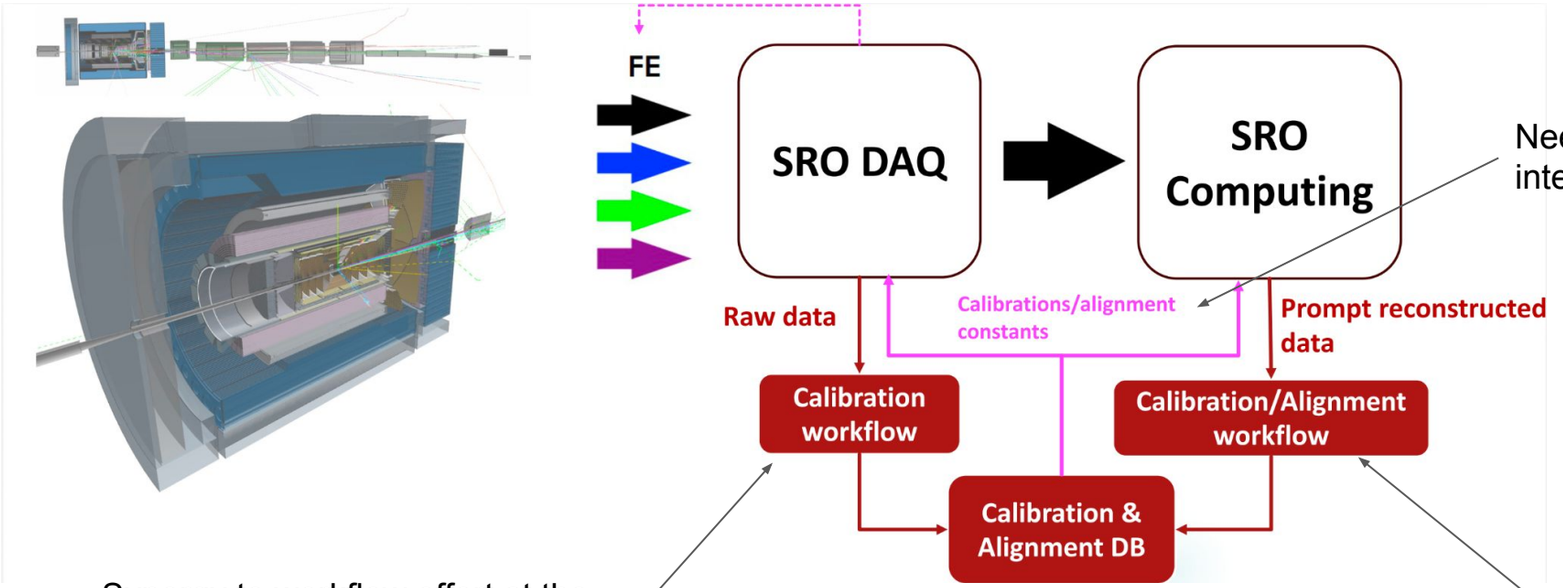
Testbed testing infrastructure

- **Panda in a box** [panda-compose]
  - Easy new way to get a local PanDA
  - Not yet a real production-grade deployment
- **Docker-compose of the testbed** revamped
  - Brings testbed (with the monitor) to any machine
  - Next: could use a similarly containerized Rucio and panda-compose
- Initial end-to-end workflow **CI testing**
  - Docker compose running on GitHub Actions
  - Runs prompt processing workflow
  - Next: fast processing



Calibration workflows at the testbed

- Will build calibration workflow prototypes in the testbed (EEEMCal detector is the first target) towards the Streaming Computing challenges
- Extend testbed to fast calibration, alignment and monitoring workflows
  - Start with file-based processing payload (Snakemake workflow) with a proto-interface to calibration DB
  - Extend to fast processing



Need to define this interface

Synergy to workflow effort at the Computing side?

Need a way to define detector-specific workflows and provide robust mechanisms for running those, incl. at E2+

Snakemake integration

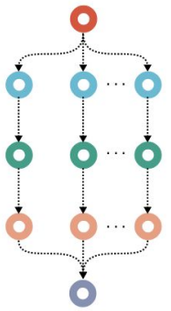
Snakemake Workflow Management System embodies properties needed for a data analysis to become reproducible, adaptable and transparent.

a

```
configfile: "config.yaml"

rule all:
    input:
        expand(
            "results/plots/{country}.hist.pdf",
            country=config["countries"]
        )

rule download_data:
    output:
        "data/worldcitiespop.csv"
    log:
        "logs/download.log"
    conda:
        "envs/curl.yaml"
    shell:
        "curl -L https://burntsushi.net/stuff/worldcitiespop.csv > {output} 2> {log}"

rule select_by_country:
    input:
        "data/worldcitiespop.csv"
    output:
        "results/by-country/{country}.csv"
    log:
        "logs/select-by-country/{country}.log"
    conda:
        "envs/xsv.yaml"
    shell:
        "xsv search -s Country '{wildcards.country}' "
        " {input} > {output} 2> {log}"

rule plot_histogram:
    input:
        "results/by-country/{country}.csv"
    output:
        "results/plots/{country}.hist.svg"
    container:
        "docker://faizanbashir/python-datascience:3.6"
    log:
        "logs/plot-hist/{country}.log"
    script:
        "scripts/plot-hist.py"

rule convert_to_pdf:
    input:
        "{prefix}.svg"
    output:
        "{prefix}.pdf"
    log:
        "logs/convert-to-pdf/{prefix}.log"
    wrapper:
        "0.47.0/utils/cairosvg"
```

Legend
- domain knowledge
- technical knowledge
- Snakemake knowledge
- trivial

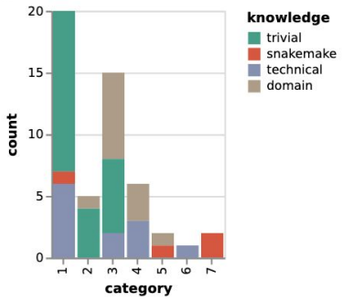
b



c

```
import sys
sys.stderr = open(snakemake.log[0], "w")

import matplotlib.pyplot as plt
import pandas as pd

cities = pd.read_csv(snakemake.input[0])

plt.hist(cities["Population"], bins=50)

plt.savefig(snakemake.output[0])
```

d



- Snakemake allows to define data flow as **Direct Acyclic Graph of jobs** (b), composing tool/script (c) calls with effortless scalability
- **Rules similar to Makefiles** (a) specify inputs in addition to outputs
- Uses a language that is a very **simple extension of Python**; leaves minimal syntax footprint (d)
- Has been in use at ePIC to run physics and detector benchmarks since 2023
  - allowed to run benchmarks in two different systems: automated isolated CI as well as user and developer personal machines
  - existing integration with PanDA is investigated
- We are looking to explore **integration into testbed** for monitoring and calibration workflows

Brookhaven National Laboratory

ePIC benchmarks on PanDA – Distributed CI

Problem: current gitlab server hosted at ANL is overloaded by CI and validation jobs
Solution: offload to other compute resources, ideally with minimal changes or impact

Approaches:

- jacamar-ci (LLNL Exascale Compute Project)
  - Runs each GitLab job on a computing farm
  - Demonstration for OSG via htcondor: detector_benchmarks#279
- snakemake-executor-plugin-panda (Developed as part of BNL testbed)
  - We are also extending snakemake-storage-plugin-rucio to work for ePIC benchmarks
  - Demonstrated running of select detector_benchmarks spawned from eicweb job to PanDA+Rucio on OSG queue – singular GitLab job steering multiple jobs on grid computing

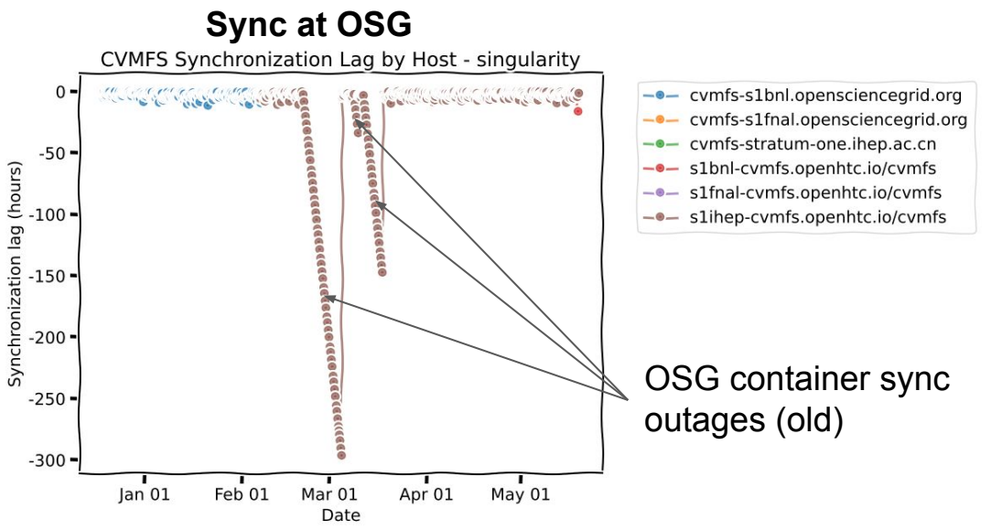
More information in backup slides

Summary

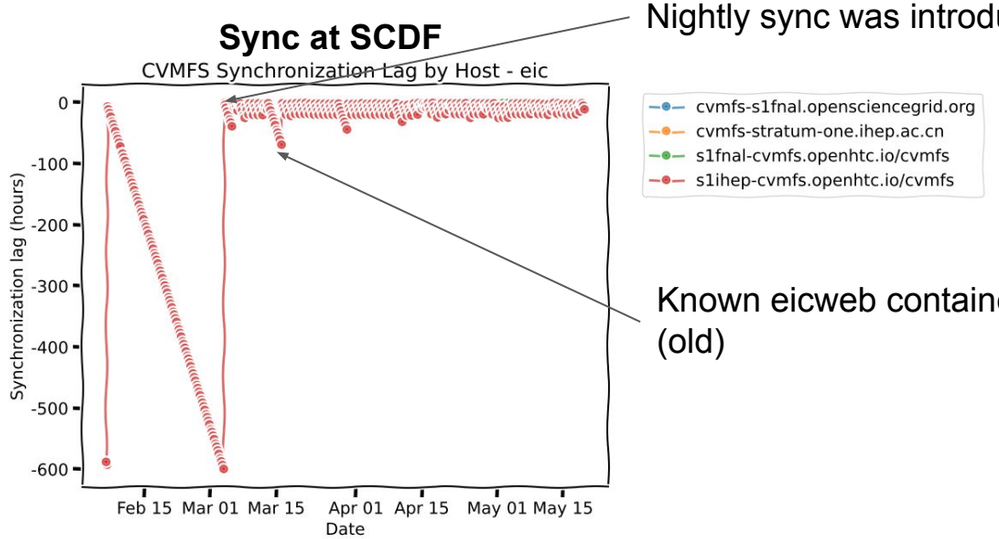
- Building this on a common infrastructure with the testbed, each benefits from the other
- **Prompt STF processing** is our testbed heartbeat, exercising system and new elements every day
  - Prime ground for enhancing testbed's AI integration
  - Decision box will extend current model to work on two (for now, virtual) E1's
- **Fast TF processing** runs the full chain from STF to PanDA workers
  - Is now working with a real EICrecon payload in persistent ZMQ server mode
- Building on the workflows in new directions
  - Containerization/packaging including the long wished for miniPanDA!
  - **ePIC distributed CI**, calibration including Snakemake-PanDA integration that can support it

Backup

CVMFS operations at BNL for ePIC



OSG container sync outages (old)



Nightly sync was introduced at BNL

Known eicweb container builder outage (old)

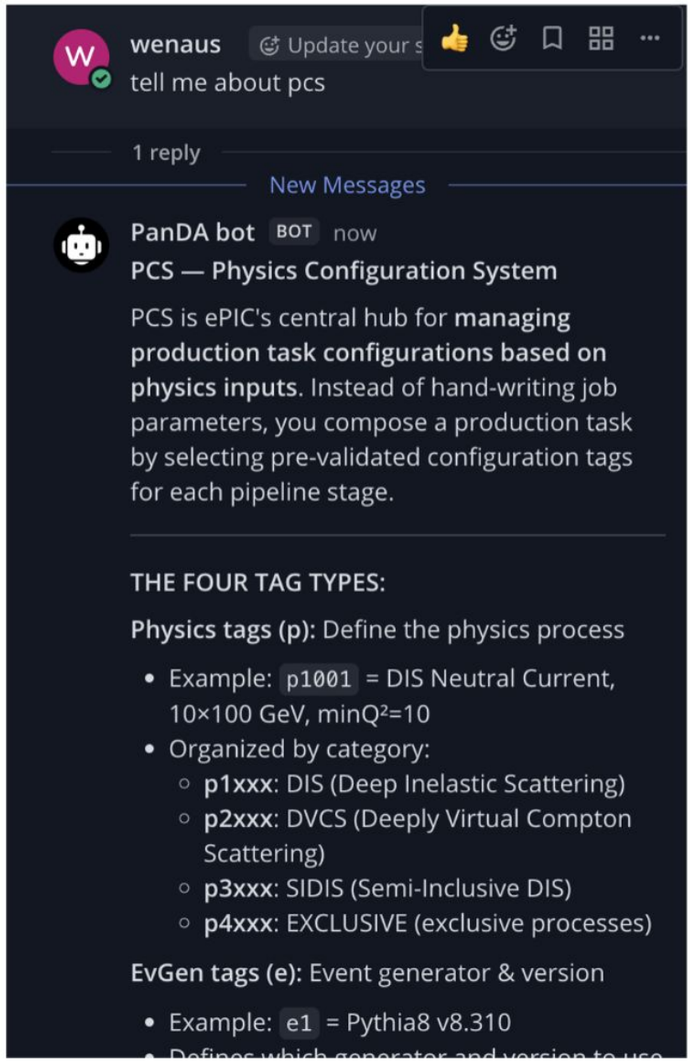
<u>No new outages since last PD update.</u>

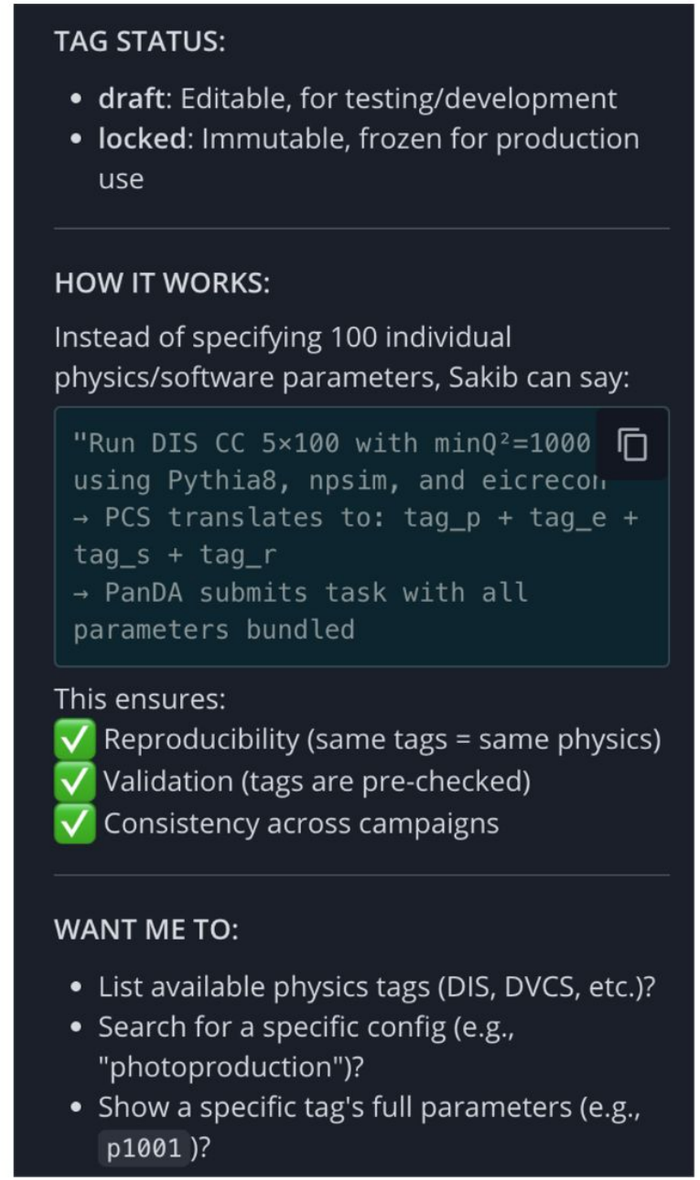
Continued investigation for OSG ticket #82690. PR osg/cvmfs-singularity-sync#548 was submitted and is now online at SCDF writer node. People are likely at HTC26 now.

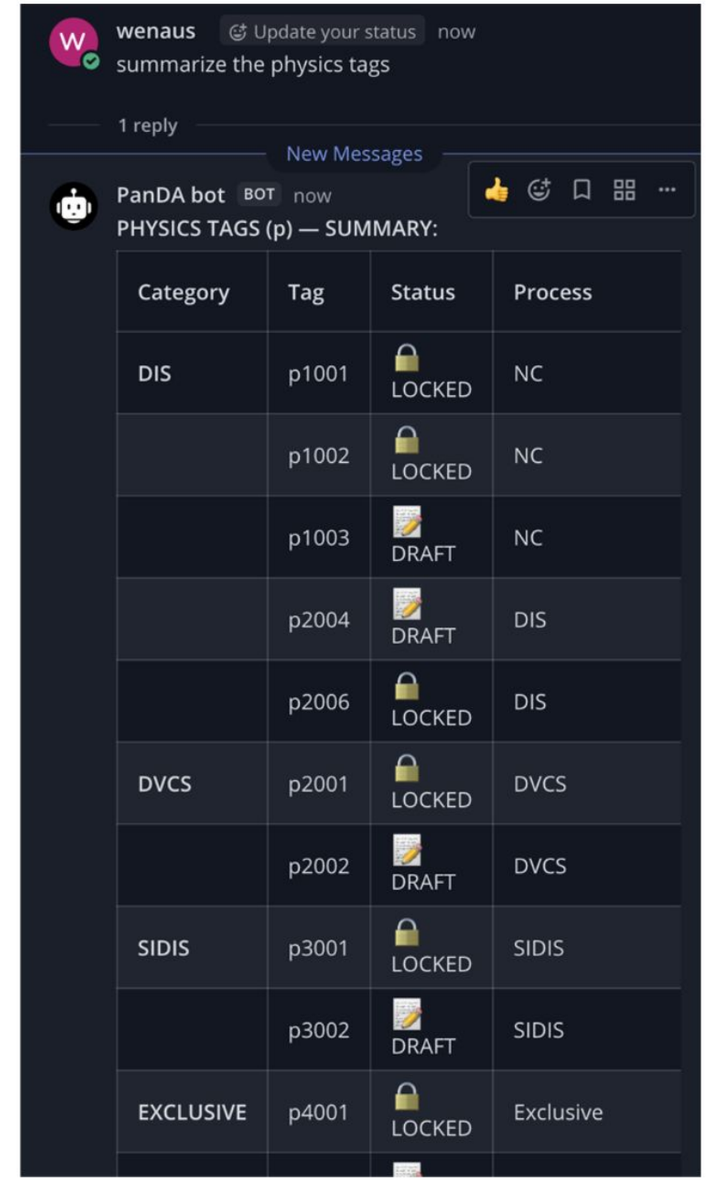
cvmfs-singularity-sync	/cvmfs/eic.opensciencegrid.org (nfs)	
	/tmp (tmpfs)	90% speedup
	--build-dir on tmpfs, then move to nfs	35% speedup
DUCC	Requires cvmfs_server - not on cvmfswrite01	

Mattermost bot in steady development (pandabot => TBD)

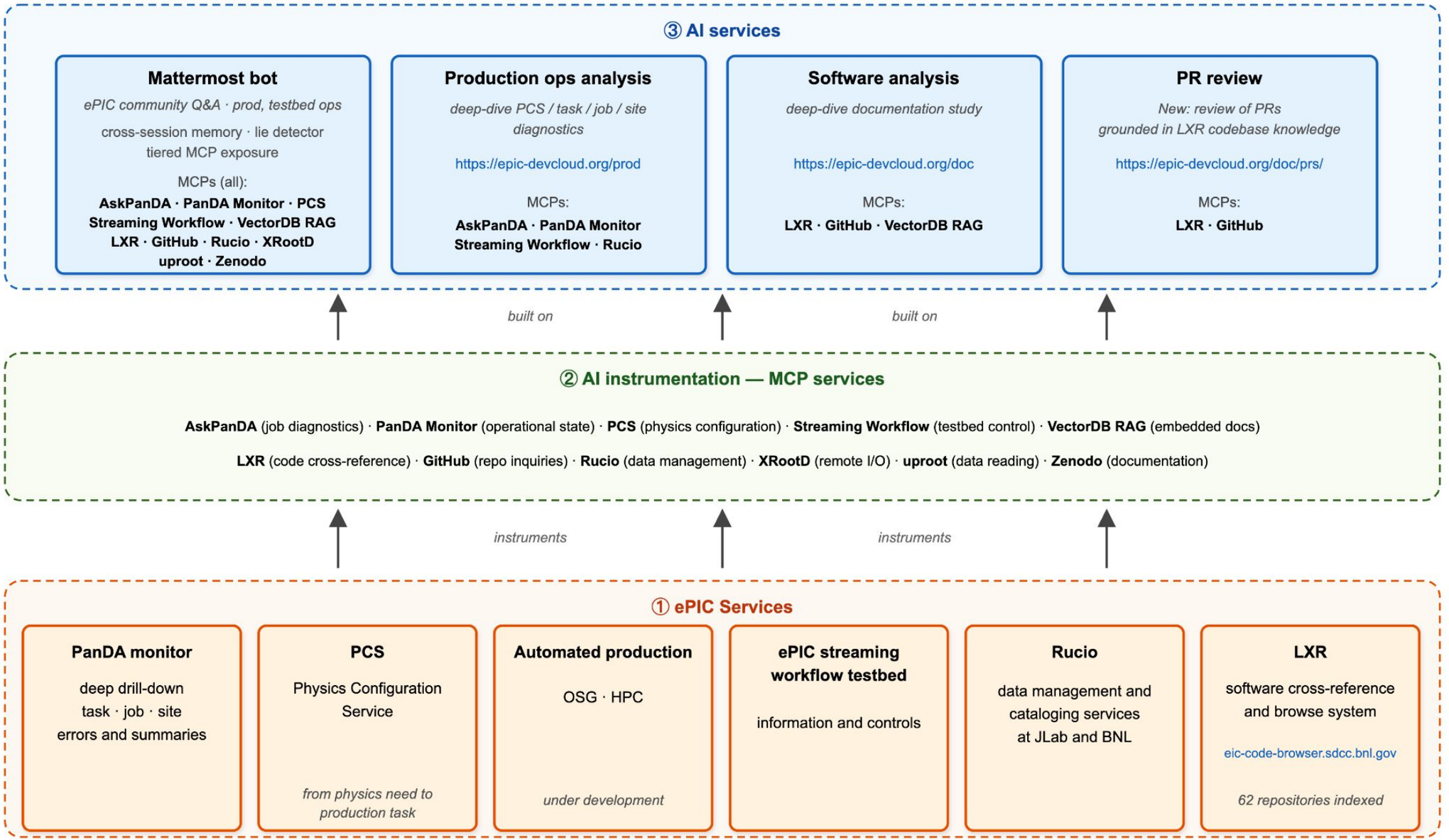
[Bot demo video for AmSC]





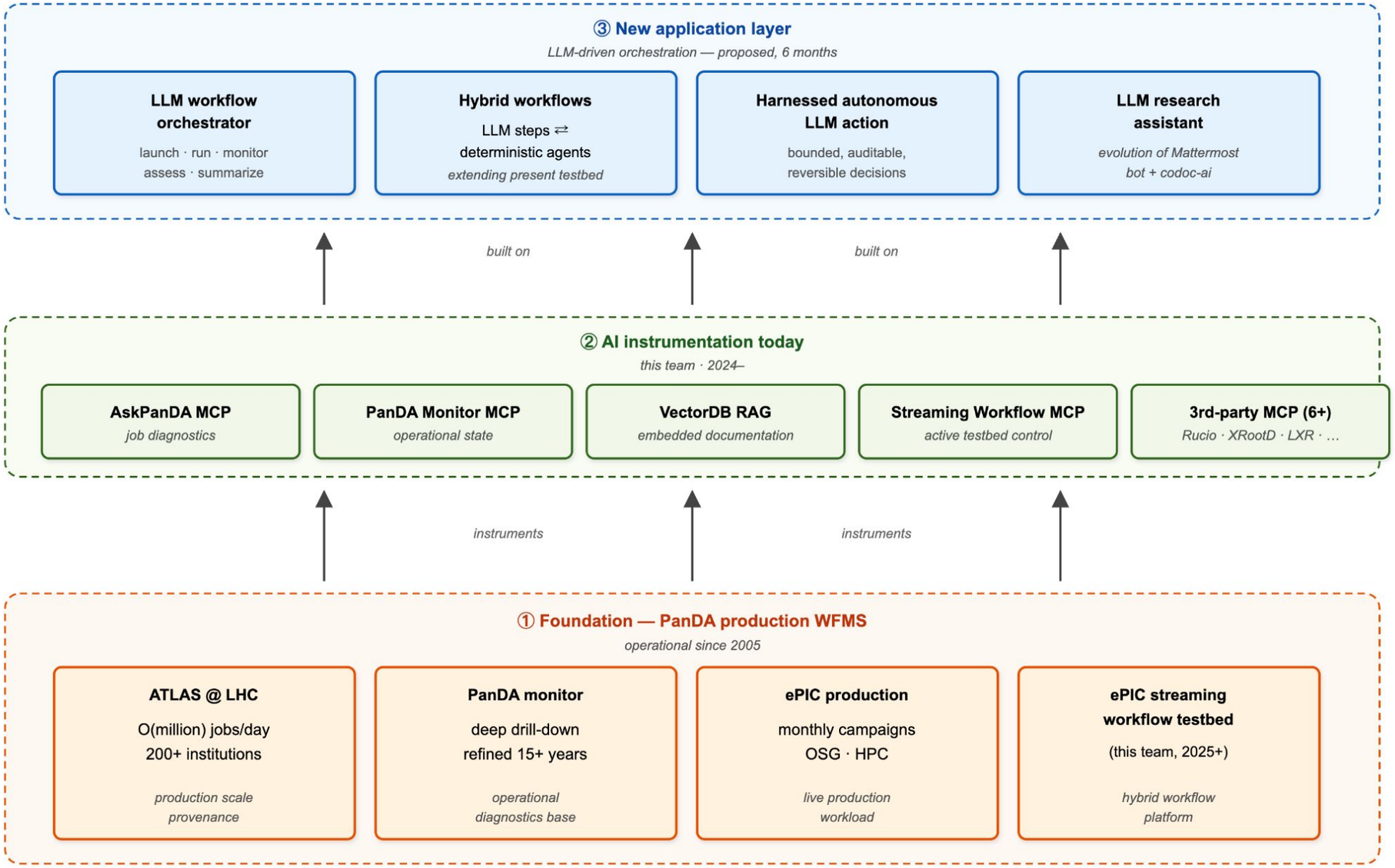


The bot is part of a now large & growing AI stack



Plus a new opt-in read-only postgres MCP service on the testbed/prod machine for LLM (not bot) use

AI in the system - present and future (slide made for AmSC)



Foundation (decades) → **Instrumentation** (this team, recent) → **New application** (6-month delta)

each layer is a bounded, incremental extension of the one below it

Distributed CI for ePIC (for nightlies, ART-like)

Snakemake-PanDA integration opens a pathway for running ePIC benchmarks distributed with PanDA.
ATLAS Release Tester (ART) evaluated for application at ePIC: it is a collection of scripts describing the infrastructure, turns out many comparable components were already built at the ePIC infrastructure:

ART/ATLAS	ePIC
packages, platform	container, Spack platform <- base container (debian vs nvidia)
scripts with ArtHeader	Snakemake workflows
ArtLocal	Snakemake
ArtGrid	TBD
EOS + CVMFS	XRootD /EVGEN/ + eicweb /scratch + Private container registry
ArtDiff (runs `acmd.py diff-root`)	Capybara

The current conclusion is that there is no benefit in re-using ART scripts due to hardcoding. **We will take the inspiration** for the missing distributed execution, namely adopt CVMFS instead of Docker registry.

Snakemake workflows on PanDA (for CI and calib. workflows)

- We work with upstream implementation for Rucio (snakemake-storage-plugin-rucio)
  - Now generally works with BNL instance
  - Path rewriting to be upstreamed #47 - Rucio paths restricted to ^[A-Za-z0-9][A-Za-z0-9\\.\\-_]*$, needs rewriting to a human readable encoding
- New snakemake-executor-plugin-panda is being developed (public release!)
  - Inverted container-payload relation: relies on /cvmfs/eic.opensciencegrid.org/snakemake/9.20 for --use-singularity mode of Snakemake
- PanDA/Rucio access tokens pushed to eicweb
  - OIDC token issued by Indigo IAM instance, packaged to a JSON to be placed at ~/.pathena/.token for panda-client to operate
  - First PanDA tasks submitted from GitLab jobs (submit+wait+rucio access) [Demo]

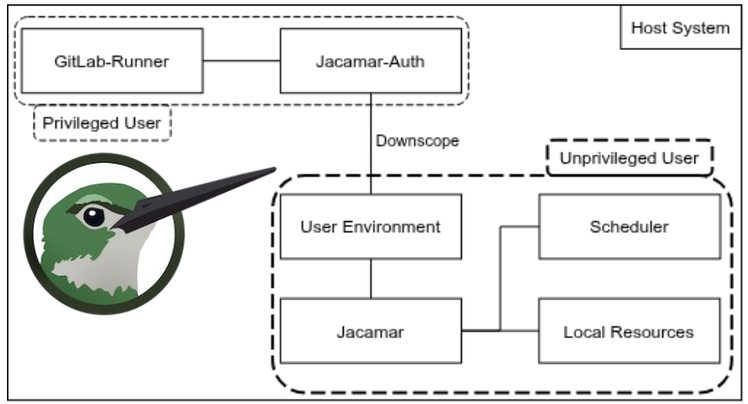
```
68  11:46:19  WARNING : The grid queue names could change due to consolidation, migration, etc. Please check with the command listAnalyPQ to use only
69  11:46:19  INFO : gathering files under /builds/EIC/benchmarks/detector_benchmarks
70  11:46:30  INFO : upload sandbox
71  11:46:32  INFO : submit user.veprbl.eicweb-140703-7829347/
72  11:46:33  INFO : succeeded. new jediTaskID=36285
73  11:46:33  $ PANDA_TASK_ID="$(sed prun.log -ne 's/.*jediTaskID=\([[:digit:]]\+\).*/\1/p')"
74  11:46:33  $ benchmarks/panda/pwait.py "${PANDA_TASK_ID}"
75  11:52:41  Job query error: None
76  11:52:41  None
77  11:52:41  running
78  11:52:41  running
79  11:52:41  running
80  11:52:41  running
81  11:52:41  running
82  11:52:41  done
```

- Corrections to ePIC benchmark workflows towards remote running support
  - Added an init script to automatically initialize the default geometry in the container
  - Corrections to benchmarks to make them compliant – Snakemake workflows running with scheduler/fs plugins are more restrictive (e.g. can't assume relative paths, need explicit file dependencies with "local" files marked)

Distributed CI with jacamar-ci and HTCondor executor to OSG

Problem: current gitlab server hosted at ANL is overloaded by CI and validation jobs
Solution: offload to other compute resources, ideally with minimal changes or impact
Approach: jacamar-ci (LLNL Exascale Compute Project)

- Bridges gitlab-runner with workflow mgmt systems
  - Gitlab runner with custom executors: slurm, pbs
  - EIC developed: htcondor-executor and panda-executor
- Security model: user mapping and permissions



Demonstration: detector_benchmarks#279 ↴

Ongoing work:

- gitlab-runner systemd persistency on ap43
- shared file storage between jobs: pelican
- pipeline containers: push to OSG harbor
- extension to PanDA

```
01:18:20  Running with gitlab-runner 18.11.1 (5265d41d)
01:18:20    on ospool-ap4043 6DnEa3vfS, system ID: s_95c02192ffbf
01:18:20  Preparing the "custom" executor
0 01:46:36      reason: Missing output files: results/campaign; Input files updated by another job: LOG/main/epic/D
0           2=1_beamEffects_xAngle=-0.025_hiDiv_1.npsim.load.png, LOG/main/epic/DIS/NC/10x100/minQ2=1/pythia8NCDIS_
0           v_1.eicrecon.memory.png, LOG/main/epic/DIS/NC/10x100/minQ2=1/pythia8NCDIS_10x100_minQ2=1_beamEffects_xA
0           n/epic/DIS/NC/10x100/minQ2=1/pythia8NCDIS_10x100_minQ2=1_beamEffects_xAngle=-0.025_hiDiv_1.eicrecon.loa
0 01:46:36      resources: tmpdir=/srv/scratch
0 01:46:36  [Sun May 10 21:46:23 2026]
0 01:46:36  Finished jobid: 0 (Rule: campaign_benchmark)
0 01:46:36  6 of 6 steps (100%!)(MISSING) done
0 01:46:36  Complete log(s): /srv/scratch/builds/EIC/benchmarks/detector_benchmarks/.snakemake/log/2026-05-10T21244
01:47:06  Uploading artifacts for successful job
01:47:07  Uploading artifacts...
```