Powering Tests analysis using SWAN notebooks:

Towards a (semi)-automatic coherent analysis of all tests on all circuits

1st Part: The current state

M.Bednarek, Z. Charifoulline, A. Mnich

01/12/2022, MP3 Day

SIGMON (Signal Monitoring) - current state

- What we have:
  - ~50 notebooks for the analysis of the Powering Test
  - FPA notebooks for the manual analysis of quench
  - Most of the code is in the *lhcsmapi* library although can't be considered as a proper api yet - many analysis code still coupled with the Jupyter environment
  - New 'single-analysis' packages (*lhcsmapi-qh*, *lhcsmapi-ee*, *lhcsmapi-pic*, …)
  - Library with the tooling around the notebooks - `lhcsmnb`
  - CI/CD (Continuous Integration/Continuous Delivery) pipeline on GitLab for the quality assurance and deployments (to EOS and AccPy repository)
  - Jenkins pipeline for the notebooks' remote execution
- How it is used by MP3 (and Proof of Concepts):
  - Manual execution of the notebooks on SWAN with the 'copy&paste' from AccTesting
  - Script execution in the AccPy virtual environment on VM
  - Manual execution of the notebook on VM (within AccPy venv)
  - PyCharm on Windows (with Spark in the local mode)

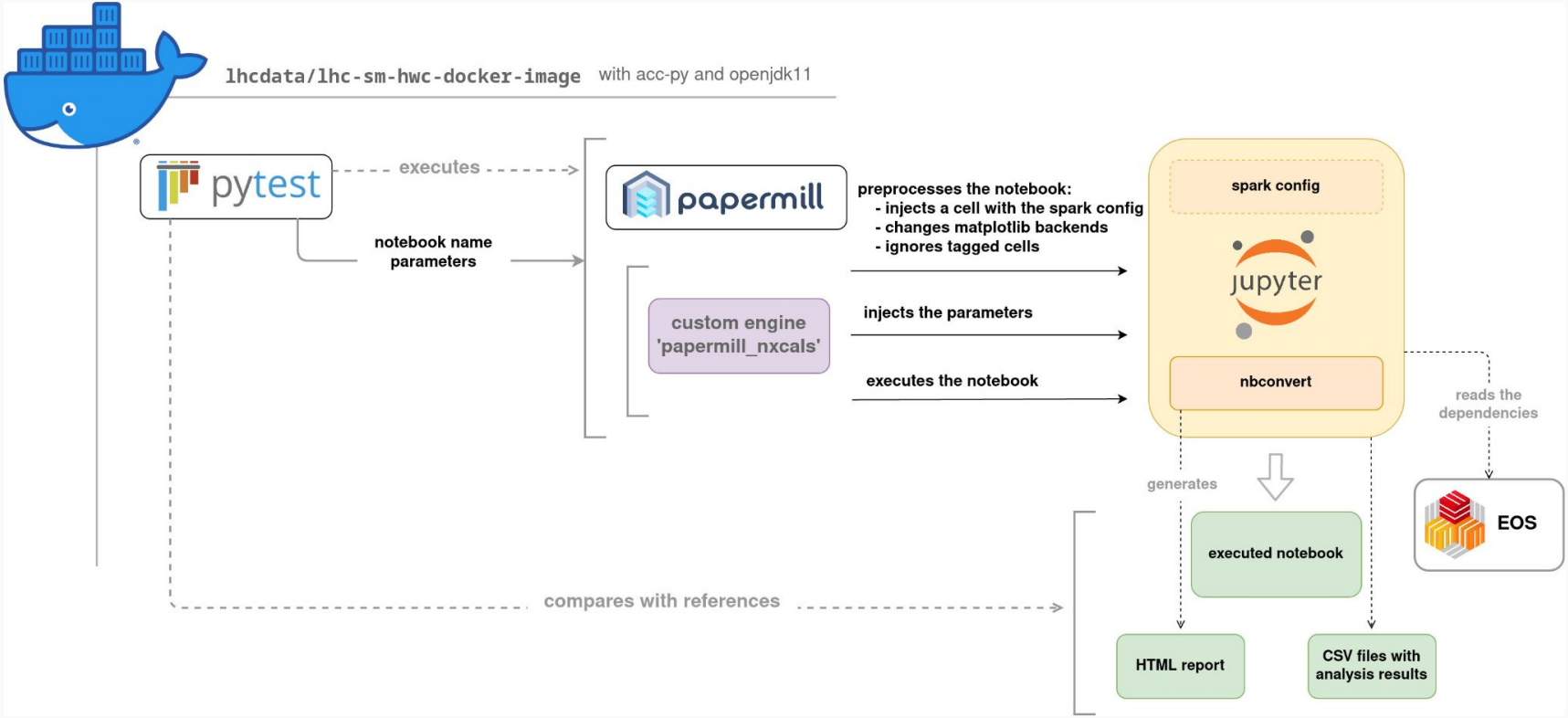
Current state - Sigmon core and API

- Code coupling reduced - easier to debug, easier to use, resolving metadata, querying and processing can be used independently
- Query API:
  - signals from NxCals
  - signals and scalars from PM dumps
  - timestamps and sources of the dump events
  - tutorial notebook
- Processing API:
  - common transformations
  - tutorial notebook
- Analysis API:
  - generic interface of the *Analysis* and *AnalysisManager*
  - common approach to store and access signals and partial results ()
  - possibility to easily intercept all the warnings occurred
  - evaluation ongoing (new analysis code)
- QPS parameters - tutorial notebook

Current state - Notebooks' Quality Assurance

- Dedicated job that executes all the notebooks on the top of the current PRO and current DEV and checks the results
- possibility to manually trigger a job that would test all the notebooks on the top of any branch
- documentation
- lessons learned:
  - sometimes certain lines are not flushed to the output (even though everything was executed)
  - sometimes empty line are inserted randomly to the output (parsing may fail)
  - sometimes the Python kernel dies (race conditions in the kernel)

Current state - Notebooks' Quality Assurance



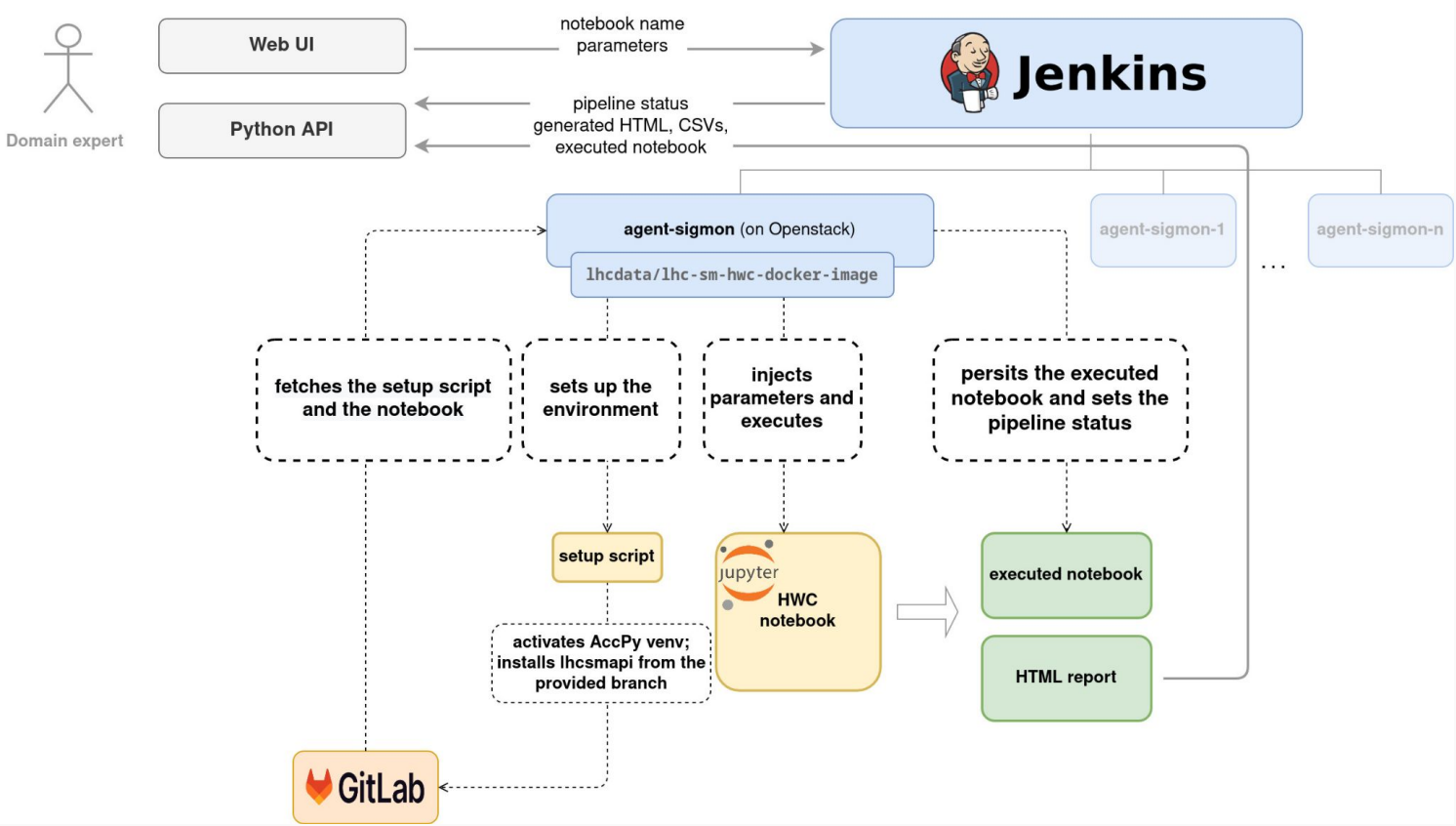
Current state - Qualification, notebooks triggering

- A dedicated pipeline on the MPE SW team Jenkins instance
- Can be triggered manually from the UI
- Can be triggered using the Python API - documentation
- All the Sigmon notebooks that are currently maintained by the SW team can be parametrized and executed without any change

```
import logging
from lhcsmnb import jenkins
logging.basicConfig(level=logging.INFO)
auth = (USERNAME, USER_TOKEN)

parameters = {
    'notebook': 'ipq/AN_IPQ_FPA.ipynb',
    'nb_type': 'FGC',
    'circuit_name': 'RQ10.L4',
    'timestamp_fgc': 1614319551500000000}
jenkins.execute_notebook(parameters, auth, './')
```

Current state - Qualification, notebooks triggering



Current state - Deployments

- AccPy Repository
  - possibility to install lhcsmapi with *pip*

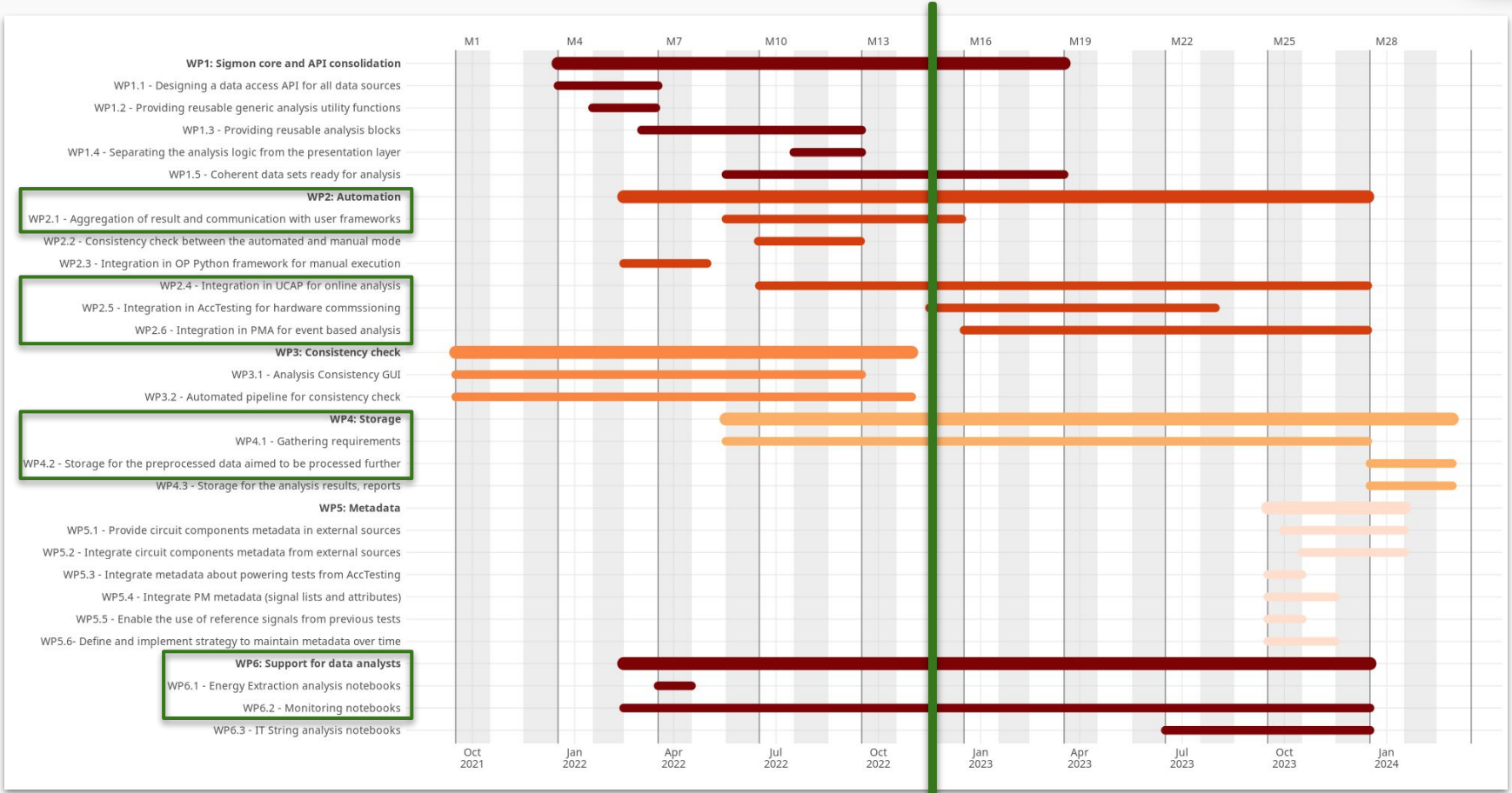
    (passing additional arguments: *"--extra-index-url http://acc-py-repo.cern.ch:8081/repository/vr-py-releases/simple --trusted-host acc-py-repo.cern.ch"*)
- EOS
  - DEV deployed nightly
  - PRO deployed during the release; all the previous versions available -> quick roll out possible
  - Possibility to deploy the venv linked to a specific branch

```
bash-4.2$ echo $PYTHONPATH
/eos/project/l/lhcsm/venv_SIGMON-176_13kA:/cvmfs/projects.cern.ch/cryogenics/hepak/100_nxcals:/cvmfs/sft.cern.ch/lcg/views/
LCG_100_nxcals/x86_64-centos7-gcc9-opt/python:/cvmfs/sft.cern.ch/lcg/views/LCG_100_nxcals/x86_64-centos7-gcc9-opt/lib:/cvmf
s/sft.cern.ch/lcg/views/LCG_100_nxcals/x86_64-centos7-gcc9-opt/lib/python3.8/site-packages
bash-4.2$
bash-4.2$ cd /eos/project/l/lhcsm
bash-4.2$ ls
dump.txt            venv_copy                                          venv_SIGMON-329_analysismanager
hwc                 venv_dev                                           venv_SIGMON-339_resurrect_monitoring_notebooks
hwc_dev             venv_gitlab_ci_notebooks_change                    venv_SIGMON-347_create_exemplary_notebook
monitoring          venv_json_schema                                   venv_SIGMON-354_qh_tests
public              venv_master                                        venv_SIGMON-397_replace_notebooks
venv                venv_pro_backup                                    venv_SIGMON-409_quench_detection_ipq
venv_0.0.1-test.3   venv_SIGMON-119_slope_calculation                  venv_SIGMON-418_check_deployment
venv_1.5.19         venv_SIGMON-176_13kA                               venv_SIGMON-420_pno_d3
venv_1.5.20         venv_SIGMON-241_document_differences_between_apis  venv_test
venv_1.5.21         venv_sigmon_291_replace_signaltransformationbuilder venvtesttest
venv_1.6.0          venv_SIGMON-295_replace_feature_query              www
venv_1.6.1          venv_SIGMON-316_use-parameters-resolver
venv_1.6.2          venv_SIGMON-320_add-tutorial-notebook
bash-4.2$
bash-4.2$ ls venv_SIGMON-176_13kA/lhcsmapi
analysis  api  gui  __init__.py  metadata  __pycache__  pyedsl  reference.py  resources  signal  Time.py  Timer.py  utils
bash-4.2$
```

Future plans

- Reflects the roadmap written in March/April 2022
  - 6 work packages: Consolidation of the core and API, Automation, Consistency check, Storage, Metadata, Support for data analysts
- Address the problems identified later on:
  - Missing analysis code, criteria not implemented
  - Limitation of the curve to curve analysis and potential threats
  - Analysis criteria not coherent - lack of the analysis maintenance strategy
  - Lack of the clear separation (in the code and tooling) between the well defined analysis procedures aimed to be automated and the manual, interactive analysis

Future plans



Future plans

- Automation
  - Making all the analysis automation ready (overall boolean analysis result is present, the procedure is followed strictly by the analysis code) - input from the experts is essential
    - PNO.d3 as an example (+ first attempt to have a combined analysis)
  - Automation solution not selected yet - solution might come ATS wide (CTTB task force)
  - Integration with AccTesting - work to be done on both AccTesting and Sigmon sides
  - Integration with PMA - decision not in our hands (but the direct support of Python by the framework more and more unlikely)
- Storage
  - Splice Resistances (Nxcals ?)
- Support for data analysts (actively participating in the development)
  - 13kA Energy Extraction for 600A (digital signals analysis)
  - Splice Resistance Monitoring (automated on GitLab ?)
  - Supporting analysis developers (Zinur, ?)

Summary

- Current state:
  - Rather stable environment (many fallbacks possible)
  - Quite usable APIs
- Current developments:
  - 13kA EE for RB/RQ ongoing
  - PNO.d3 ongoing
  - Code refactoring, bugfixing, documenting
- Future plans:
  - Automation readiness <- absolutely not yet but feasible for Jan 2024
  - Storage, semi-online analysis <- we have strong solution candidates, no development started
  - Analysis maintenance strategies <- discussions and agreements needed

End of part 1

- documentation
  - https://sigmon.docs.cern.ch/
- website
  - https://sigmon.web.cern.ch/
- GitLab
  - https://gitlab.cern.ch/LHCData/lhc-sm-api
  - https://gitlab.cern.ch/LHCData/lhc-sm-hwc
  - https://gitlab.cern.ch/LHCData/lhc-sm-analysis

lhc-signal-monitoring@cern.ch

https://mattermost.web.cern.ch/sigmon