HZB Helmholtz Zentrum Berlin

UPDATE ON BESSY-II BEAMLINES

EPICS COLLABORATION MEETING

POHANG 2024

12.04.2024

Marcel Bajdel, Luca Porzio, Will Smith, Simone Vadilonga and the WI-AOS

bluesky EPICS

INTRODUCTION

Helmholtz-Zentrum Berlin emerged from a large center (HMI) and a small light source provider (BESSY)

HMI: Wannsee site, nuclear reactor, ion implantation – now eye tumor therapy

BESSY: Adlershof site, accelerator based light sources

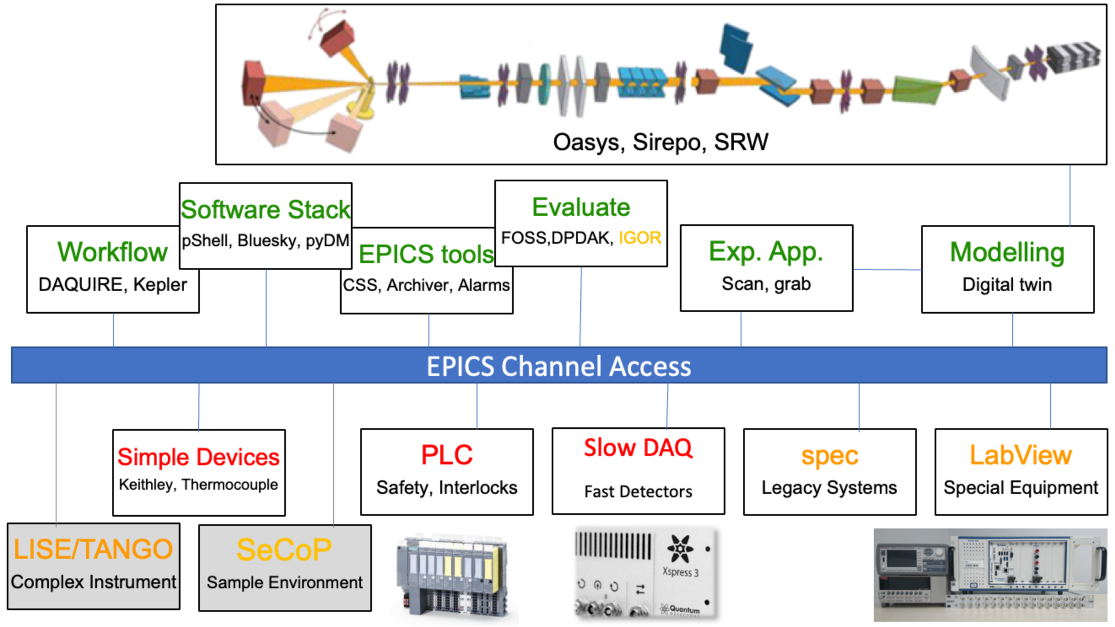
1992: With the help of DESY - BESSY first EPICS facility in Europe

Since 2015 strategic shift with main focus on renewable energy, materials and technology.

2019: Modernization concept for DAQ and Automation at BESSY II Beamlines and Instruments worked out, presented at ICALEPCS
https://accelconf.web.cern.ch/icalepcs2019/doi/JACoW-ICALEPCS2019-MOCPL02.html

Need for a new control system approach:

- require automated build and deployment procedures,
- standardize on an experiment orchestration tools,
- Enable machine learning for beamlines and experiments, develop digital twins for beamlines,
- require extensive testing.



CURRENT STATUS

Project based: upgrade, commissioning, construction, operational, legacy status.

Mixture of staff, external and temporary positions.

Beamlines are staff operated, less coherent than accelerator.

Footprints of external partners span beamlines, instruments, labs.

Post-cyber attack status

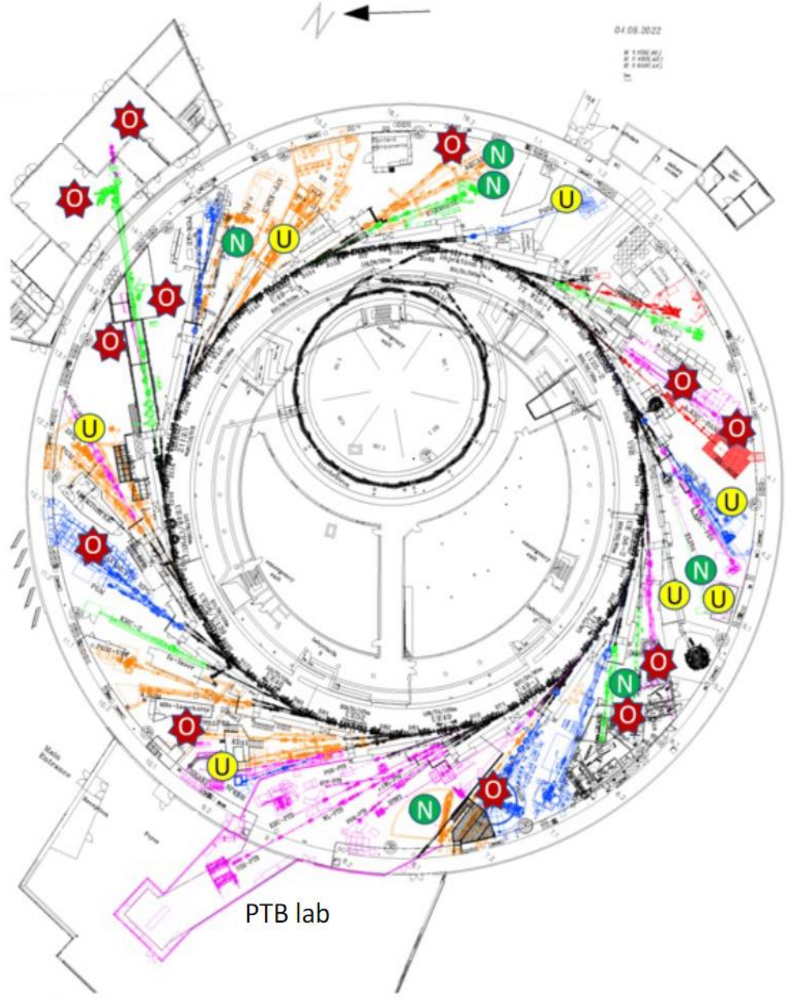
Experimental hall network segmented into production zones after cyber attack.

Zero trust network works for accelerator but not for the beamlines.

Upgrade BESSY II+ and BESSY III already after CDR.

Develop workflow that will potentially span over all beamlines to deploy microservices and maintain them

BESSY II

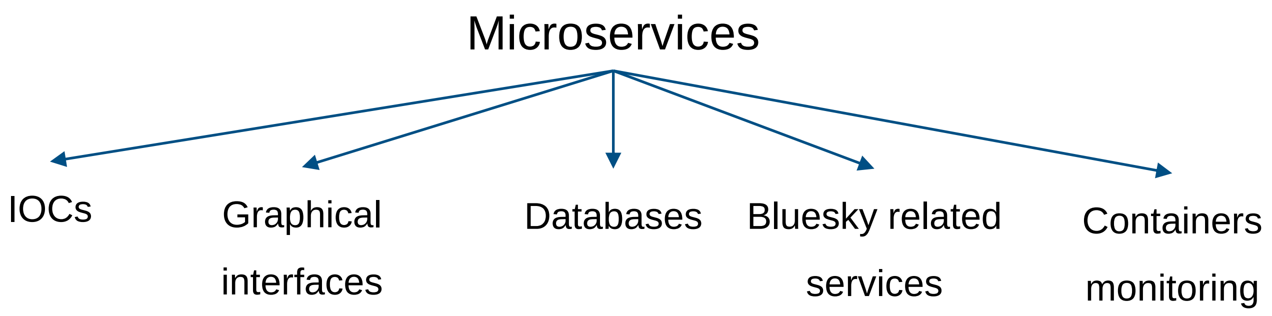


O Operando experiments already possible
N New infrastructure, started of finalised before Upgrade Proposal
U New infrastructure in the Upgrade Proposal

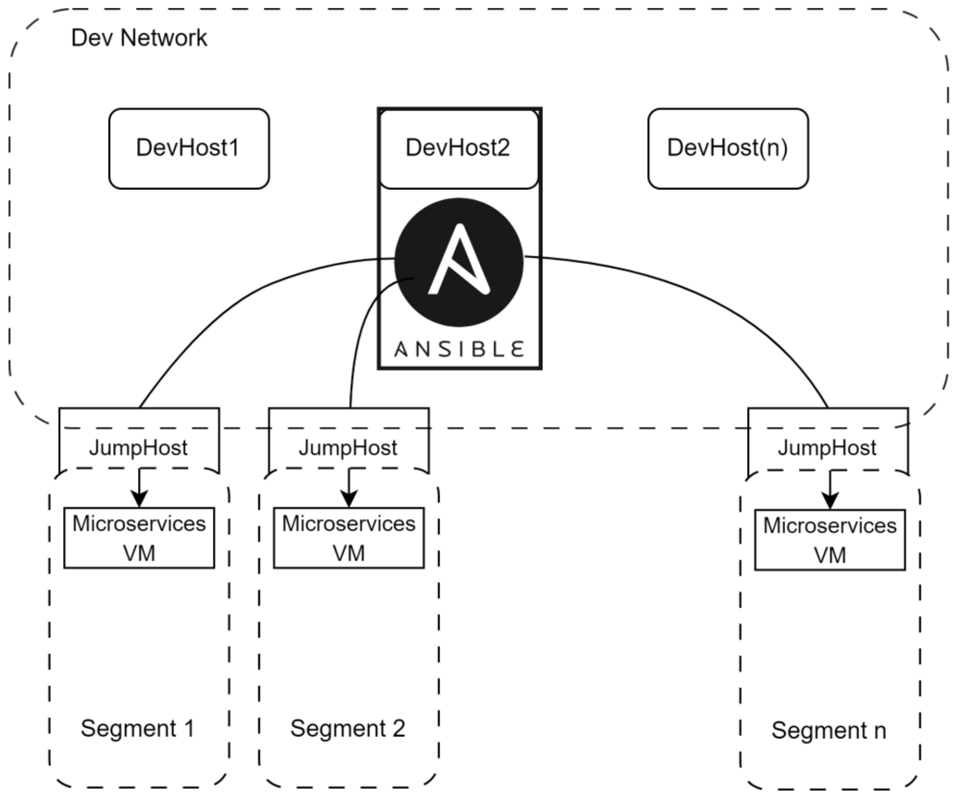
Beamline Colours (responsible division):

Green:	Solar Energy and Chemical Energy
Blue:	Quantum and Functional Materials
Orange:	Photon Science
Rot:	Scientific Instrumentation and Support
Magenta:	CRG Beamlines and PTB
Black:	Accelerator/Storage Ring

DEPLOYMENT WORKFLOW

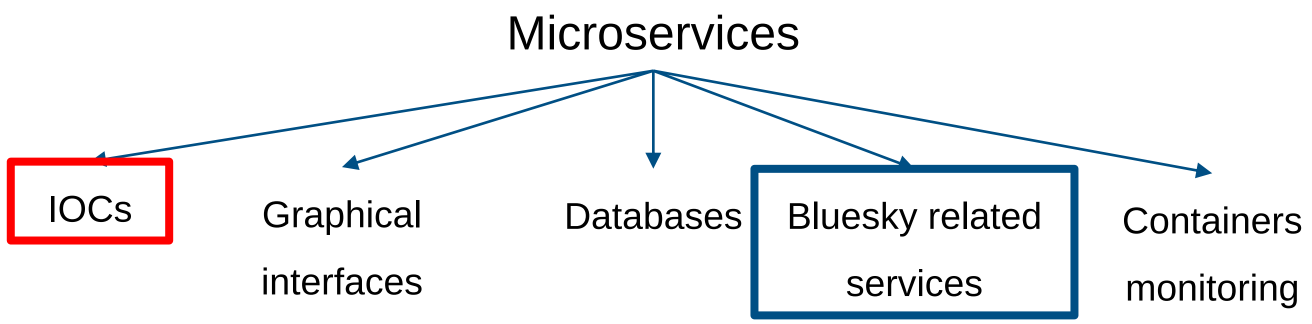


- Every microservice, perceived as a separate software agent, which can be containerized.
- Environments to run applications are also held in container images (epics base, python based image to run ipython shell).
- We try to use community standard tools only and keep it simple.
- We aim to enhance newcomers' ability to quickly become productive in their work.

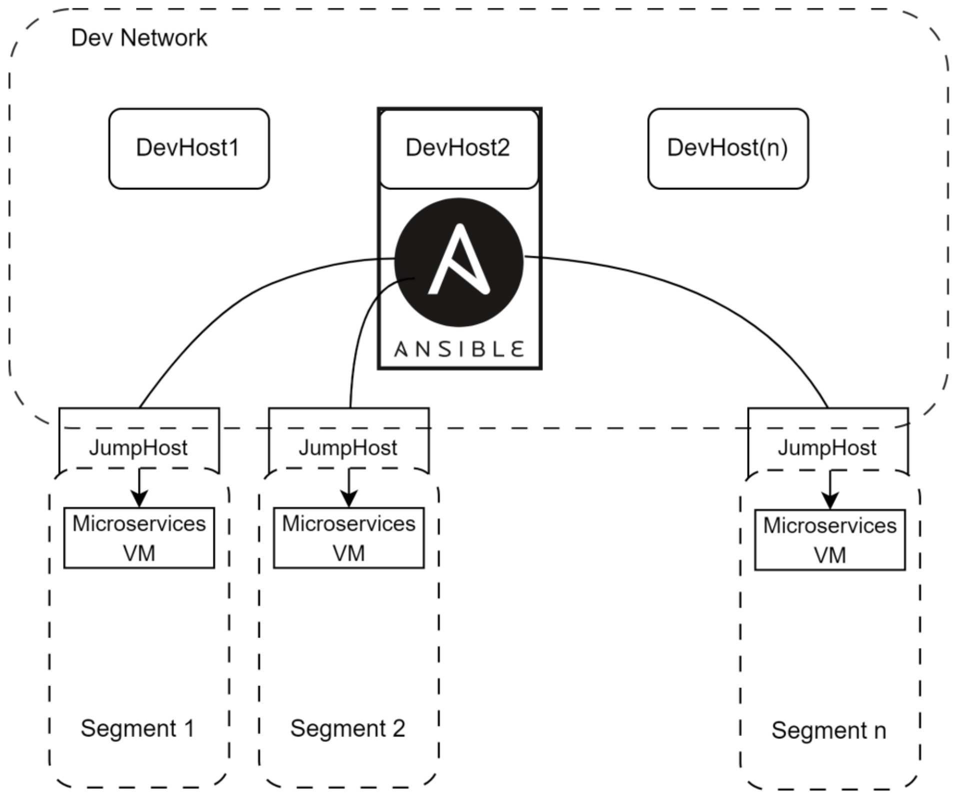


JumpHost (a gateway) that is used for a secure access to segmented networks

DEPLOYMENT WORKFLOW



- Every microservice, perceived as a separate software agent, which can be containerized
- Environments to run applications are also held in container images (epics base, python based image to run ipython shell)
- We try to use community standard tools only and keep it simple
- We aim to enhance newcomers' ability to quickly become productive in their work.



JumpHost (a gateway) that is used for a secure access to segmented networks

IOC TEMPLATING

Requirements for the IOC automation:

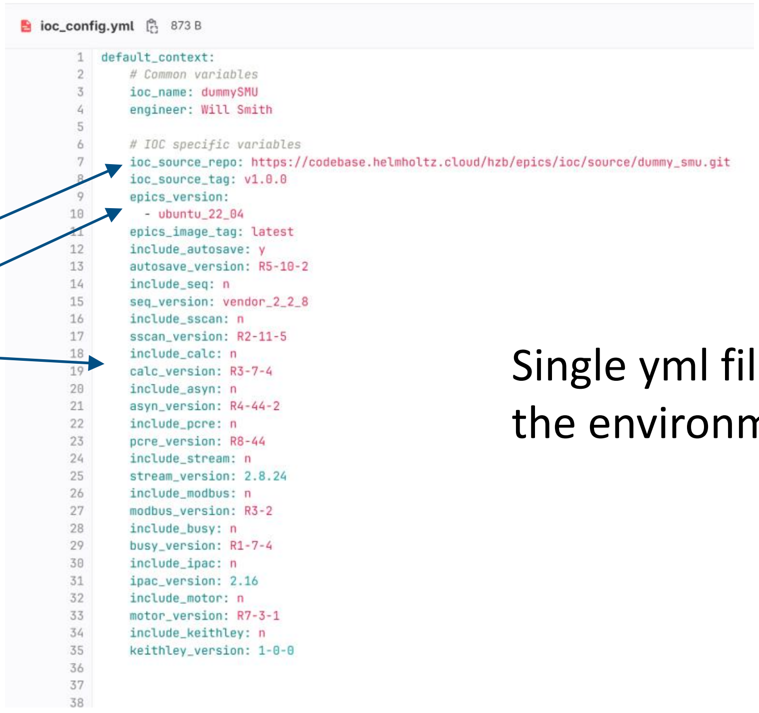
- IOC source files templating
- IOC support modules templating

Templating engine based on jinja2

Works for existing EPICS IOCs
Works for new EPICS IOCs

EPICS 7.0.7

List of support modules together with versions


ioc_config.yml 873 B

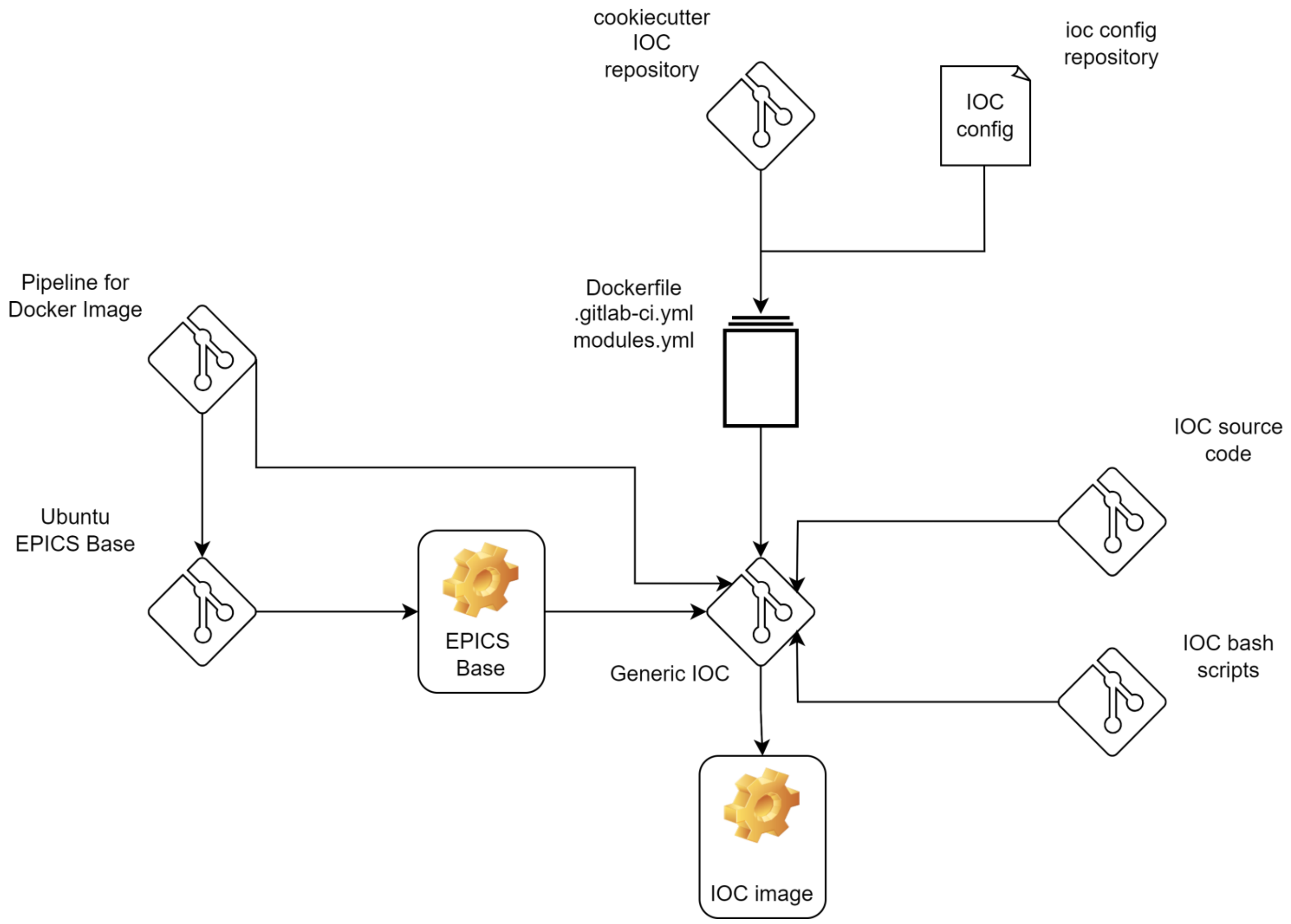
```yaml
default_context:
    # Common variables
    ioc_name: dummySMU
    engineer: Will Smith

    # IOC specific variables
    ioc_source_repo: https://codebase.helmholtz.cloud/hzb/epics/ioc/source/dummy_smu.git
    ioc_source_tag: v1.0.0
    epics_version:
      - ubuntu_22_04
    epics_image_tag: latest
    include_autosave: y
    autosave_version: R5-10-2
    include_seq: n
    seq_version: vendor_2_2_8
    include_sscan: n
    sscan_version: R2-11-5
    include_calc: n
    calc_version: R3-7-4
    include_asyn: n
    asyn_version: R4-44-2
    include_pcre: n
    pcre_version: R8-44
    include_stream: n
    stream_version: 2.8.24
    include_modbus: n
    modbus_version: R3-2
    include_busy: n
    busy_version: R1-7-4
    include_ipac: n
    ipac_version: 2.16
    include_motor: n
    motor_version: R7-3-1
    include_keithley: n
    keithley_version: 1-0-0



```

Single yml file defining the environment for an IOC

- Dockerfile
- Support modules config
- Gitlab pipeline to scan for security issues, image manifest and build/push to container registry with tag

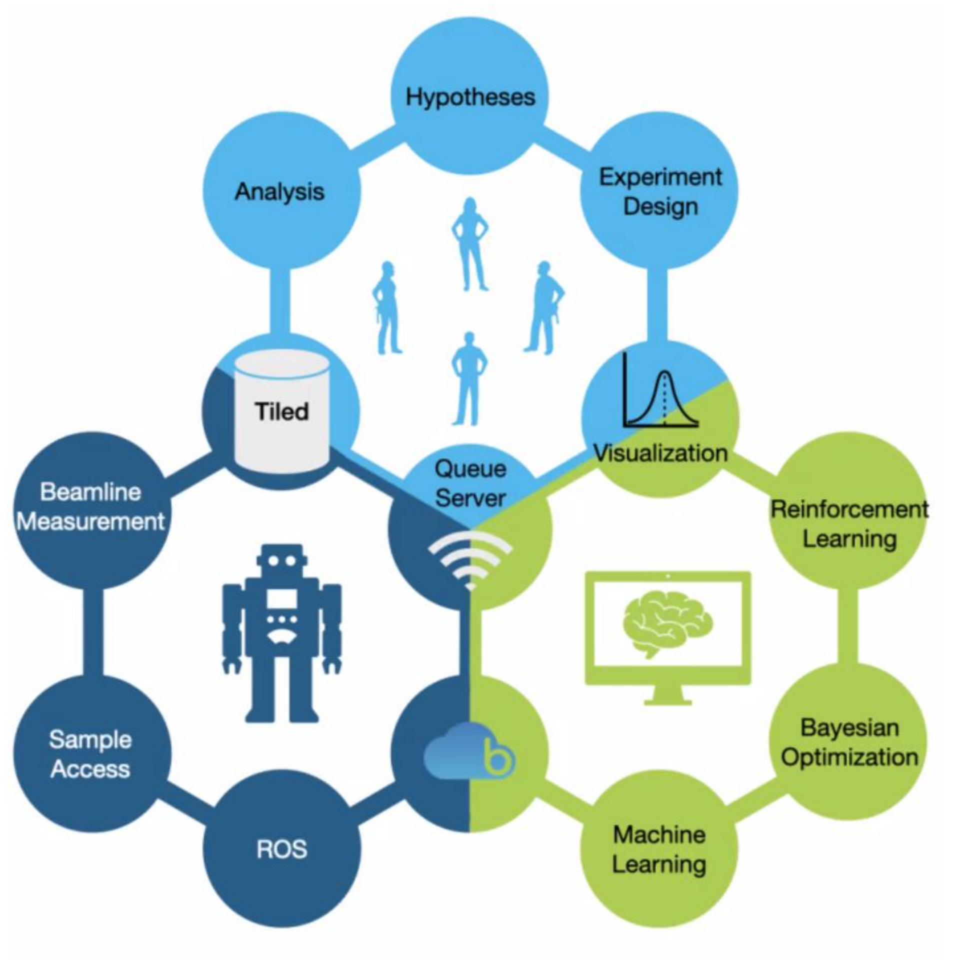
IOC AUTOMATION WORKFLOW

- IOC config for every device.
- Ubuntu as base for EPICS 7.0.x.
- We develop the IOCs using development containers (source files repository is cloned at build).
- Bash scripts used to create a RELEASE.local, download and compile support modules, patch Makefiles in the IOCApp.
- We deploy the containers on hosts with Ansible and a .yml file which defines macros.

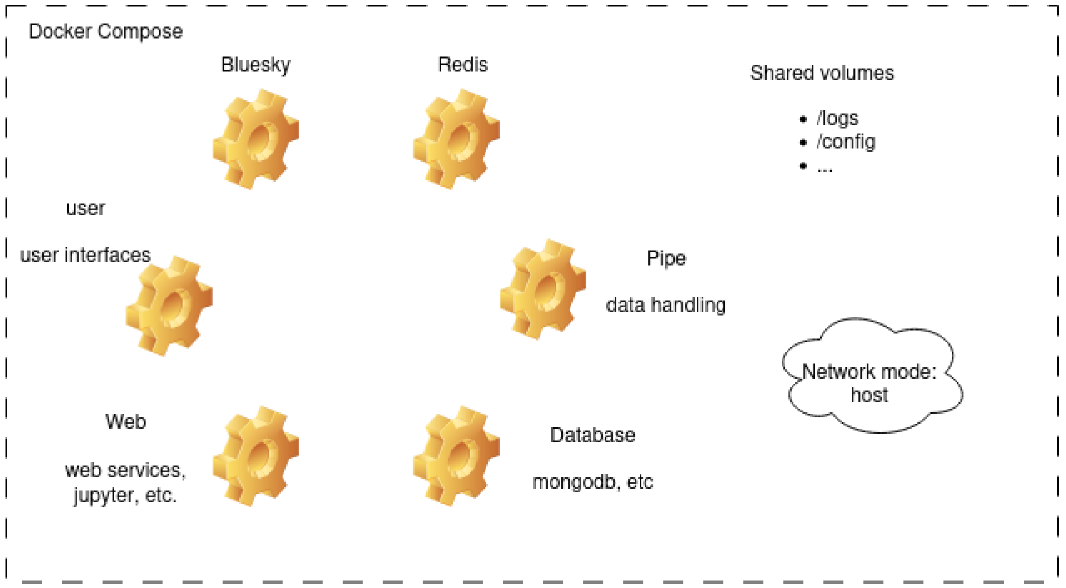


ROCK-IT PROJECT

(Remote, Operando-Controlled, Knowledge-driven, and IT-based)



System design for the ROCK-IT

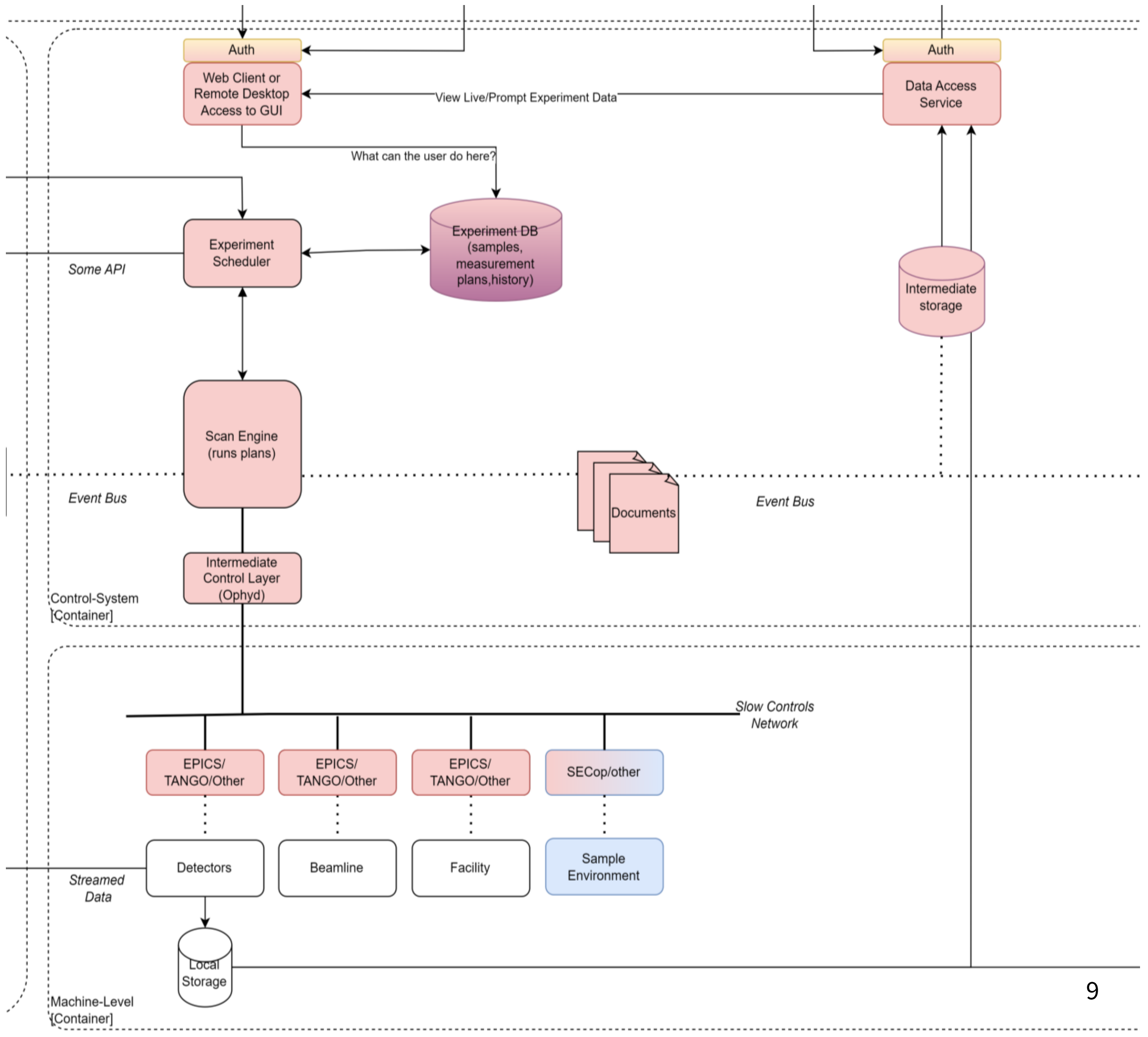


ROCK-IT starter pack

- User-friendly automated experiment environments with similar 'look and feel' at different facilities, aiming to reduce access barriers and accelerate innovation.
- enhanced remote-access protocols, holistic experiment development, and the implementation of machine learning for automation and real-time analysis

ROCK-IT PROJECT

- Implement the full deployment workflow and test it.
- Multiple machine level controls at the machine-level within experiment
- Experiment Control System based on Bluesky unifies the machine-level controls



Summary

- **Nothing is written in stone** – we've been developing, testing and trying out different approaches.
- We aim to use opensource frameworks that a newcomer might already be familiar with(jinja2, python, Gitlab CI/CD).
- HZB is a user facility – we try to find a balance between the best practices, user needs and available solutions.

To do's:

- Develop more unit tests, functional tests, stress tests (Gitlab CI/CD).
- Develop effective way of monitoring the whole control infrastructure (containers, log and data aggregation, remote access).
- Implement the building and deployment workflow and integrate with the central-IT (scientific IT).