ATLAS ITk Production Database use & Tools for ITk Pixels community

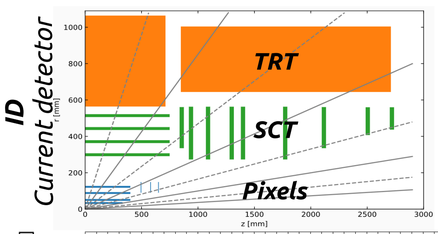
The ATLAS ITk upgrade

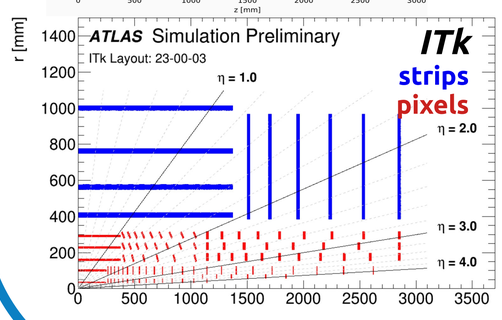
Inner Tracker upgrade for HL-LHC

	∫L	<μ>	|η|
ID	400 fb^{-1}	~58*	2.5
ITk	**3 000 fb^{-1}**	**200**	**4**

**2024 average*

# Modules	Area	# Channels
4000	63m^2	100M
9 500 (pixels) + **18 000** (strips)	**180m^2**	**5000M**





HL-LHC conditions

- High pile-up
- High level of radiations
  10 MGy Total Ionising Dose closest to beam pipe

New Inner Tracker

- Maintain track reconstruction performances
- Extend eta coverage

Challenge

- **Bigger scale → production logistics**
- **Stricter specifications**
  (radiation hardness, mechanical envelop,…)
- Same volume!

Monitoring the production

Large scale production: ~**150** institutes, ~**1300** users of the database

➔ Ensure the detector is built **on-time** and **to specifications**:

- *Track all ~10^6 components*
- *Monitor production rate & yields*
- *Control components quality*
- *Keep track of components assemblies*
- *CERN oversight of dual-use* item exports*

**dual-use item can have both civilian and military applications (e.g. FE chips) and are subject to tight control.*

Three different (interconnected) **objects to store**:

Components
`Type, location, building stage, attached tests, assembled to, ...`
Sensor, modules, services, adhesive, transport box, ...

Quality Assurance & Control tests
`pass/fail, date, operator, location, results, ...`
Electrical tests, visual inspections, mechanical tests, ...

Shipments across institutes
`sender, recipient, content, courier, ...`

Very **heterogeneous** needs: need **flexibility**

➔ **Custom** database developed on a **MongoDB** backend

➔ To be **maintained** until the end of ITk operations, circa 2040!

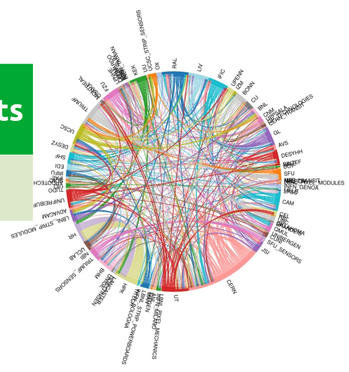
PLUS4U UNICORN *Product of the Unicorn University*

The scale of the ITk Production Database (PDB)

Pre production and production starting

Already managing a **large amount of data**

Component Types	Components Registered	Test Types	Test Runs registered	Shipments
> 450	**> 900 000**	**> 2 000**	**> 9 500 000**	**> 8000**



All registered shipments

Complex production flows of each component

- **Building stages** per components
- **Tests** attached to building stages
- **Parent/Children** relations in assemblies

	Stages	average / component	max / component	Test Types	average / component	max / component
Strips	~600	~3	11	~1250	~13	**260** (AMAC Chip)
Pixels	~770	~5	58	~850	~9	**161** (FE Chip)

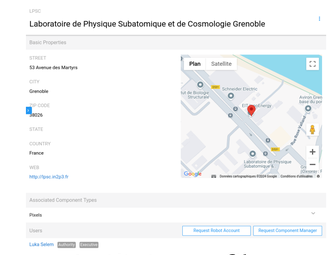
Uploading to the PDB

Component registration and test result uploads

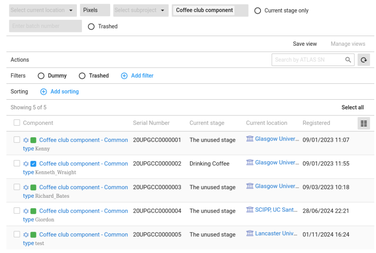
Centralised Unicorn web-based application

- Representation of database structure: **flexible → permissive**

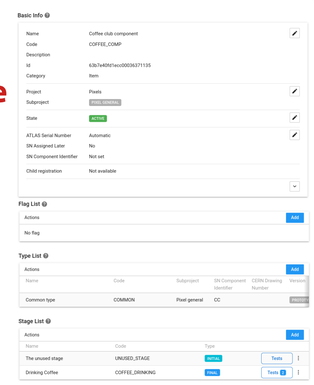
➔ **Visualise, navigate**



Institutes profile



Tunable tables of registered components



Component Type description

Upload of data: must be **constrained**

- Ensure **data integrity**
- User friendly → **avoid mistakes**
- Treatment of (well motivated) **attachments**: size constraints, inode use
  ➔ *Large ones sent to CERN EOS, should be kept at a low level*

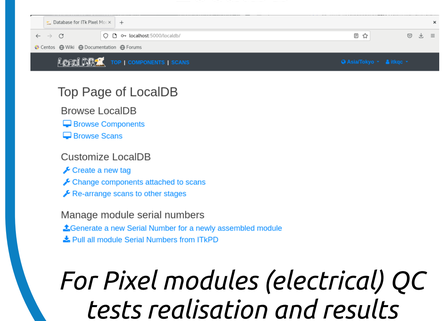
Some **components** or **building stage** = specific DB interaction needs

➔ Unicorn PDB API for **custom PDB interactions**, Python API wrapper

Variety of custom solutions using **modern dev tools**

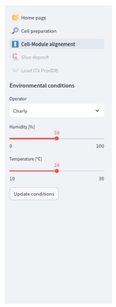
- Code forge **GitLab**, documentation generator **MkDocs**
- **Streamlit**, **Flask** python packages for **web applications**
  ➔ Deployed in **Docker** containers
  ➔ Hosted at CERN in **OpenShift** containers
- **Locally installed** tools (e.g. LocalDB)

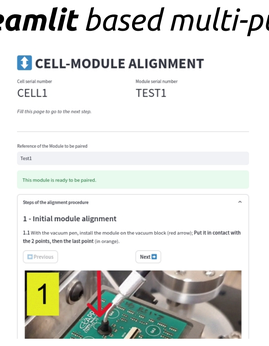
LocalDB



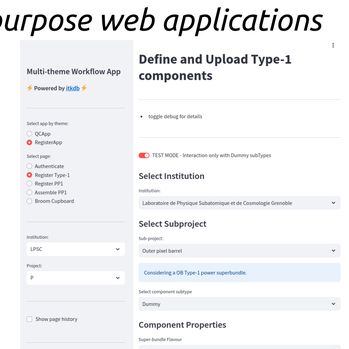
For Pixel modules (electrical) QC tests realisation and results upload

***Streamlit** based multi-purpose web applications*





To control a building stage process



For DB registrations and test results uploads

Reporting

"The Hanseatic League model"

Monitoring of the production evolution using the database:

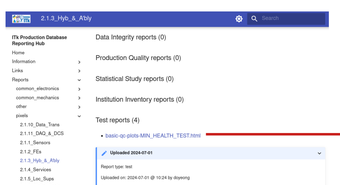
- *Data integrity checks*
- *Production quality monitoring*
- *Yield & cost reporting*
- *Simulation of production flow*
- *Automated reports & alerts*

Reports centrally hosted in a **report hub** built with **mkdocs**

- **Minimum requirements** (allow centralisation in hub, …)
- **Common tools** provided, documented

➔ **Freedom** for actual report design

make report → html → send to eos → script → report hub → link



Reporting Hub
→ *Links to reports html pages*

Example report: *html page*

GRENOBLE | MODANE

Luka SELEM (luka.selem@cern.ch)
On behalf of the ATLAS ITk Collaboration

PIXEL, Strasbourg, November 2024