

ATLAS ITk Production Database use and tools



CHEP 2024, Oct 19-25

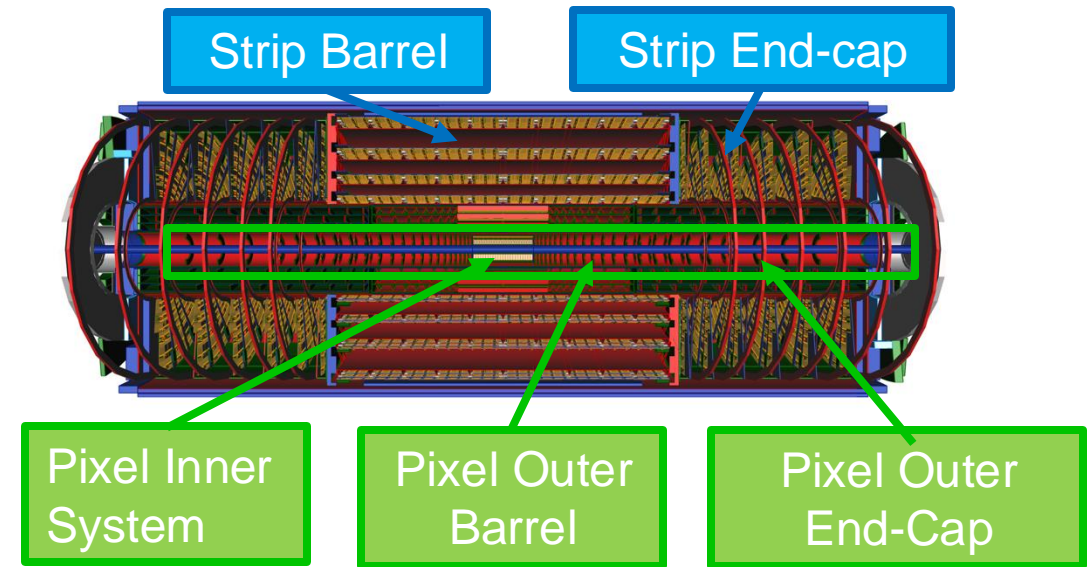
Monika Wielers (STFC – RAL)
on behalf of the ATLAS ITk collaboration



Overview of the ATLAS ITk Detector

ATL-PHYS-PUB-2021-024

- 🐾 ITk is the new ATLAS all-silicon inner tracking detector being build for running at the high-luminosity LHC
 - 🐾 New detector $\sim 3 \times$ larger than the current one
- 🐾 All parts used in detector (front-end chips, sensors, modules, supports, services) to be tracked and QC/QA test results recorded (including prototypes and pre-production items)
- 🐾 Total production period is ~ 5 years.
 - 🐾 ~ 100 institutes from 22 countries
 - 🐾 This also includes industry partners
- 🐾 In total $\sim 10^6$ parts to be produced
 - 🐾 Need for a **database** to store this information, monitor production and allow data-mining in case issues occur during build process or operation



ITk (ID)	Area (m ²)	# Modules	#channels (M)
Pixels	13 (1.6)	~ 9000 (~ 2000)	5100 (92)
Strips	165 (61)	~ 18000 (~ 4000)	60 (6.3)

New (current) inner detector

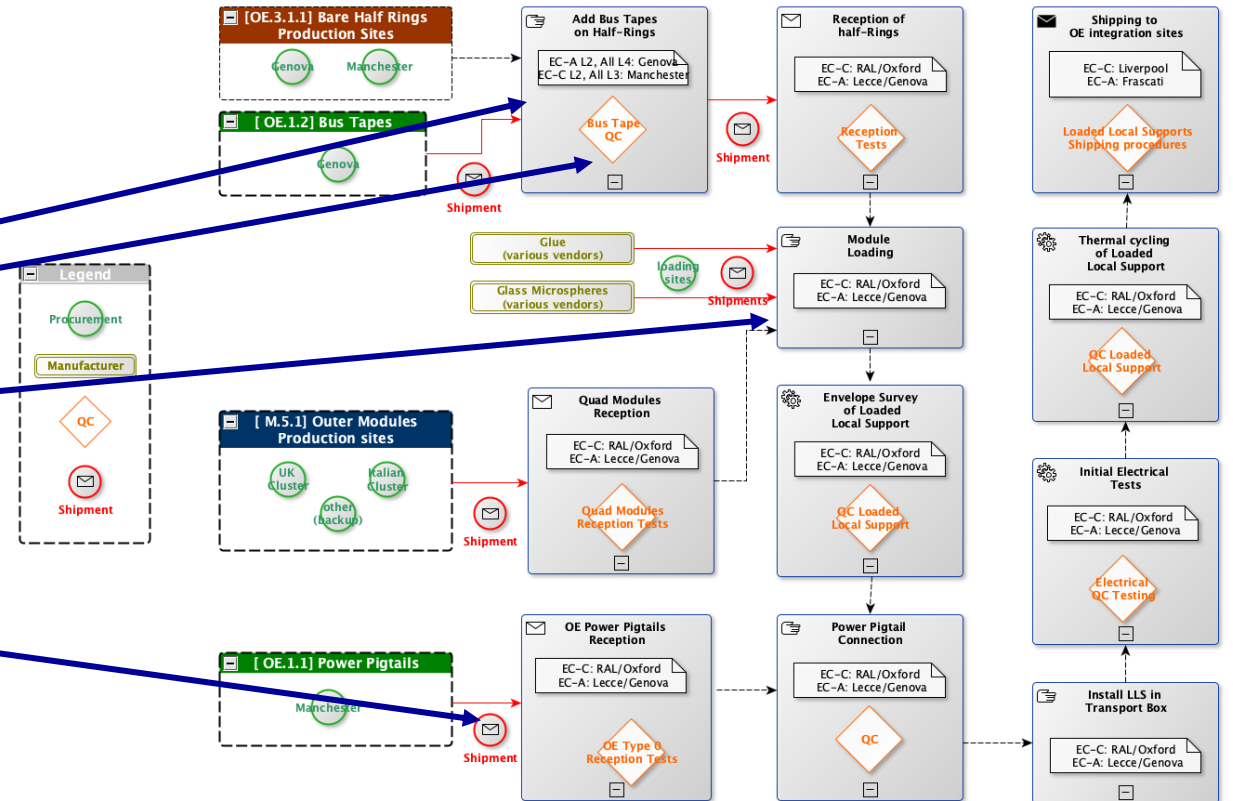
Project	Staves	Modules	Sensors	FE Chips
pixels	354	9464	9464	34292
strips	776	17888	17888	233856

Production Database: Introduction

Production database was developed by vendor for ITk

Main production database (PDB) functionality

- Registering parts (components)
- Dealing with batches
- Serial number scheme
- Tracking build status (stage)
- Adding test data
- Handling assemblies
- Tracking of parent/child relationships
- Tracking shipments
- Querying data
- Assignments of users, institutes and their privileges



Designed to allow up to 50 users can use the database simultaneously

Component information to be retained during construction and 10 years of data-taking

Production Database Technical Setup

- The ITk Production Database based on mongoDB 

- Two parts

 - User interface (front-end)

 - Server side (back-end)

- Based on the uuApp Framework  developed by vendor

 - Framework is based primary on open-source technologies

 - Fully designed as a cloud-based application

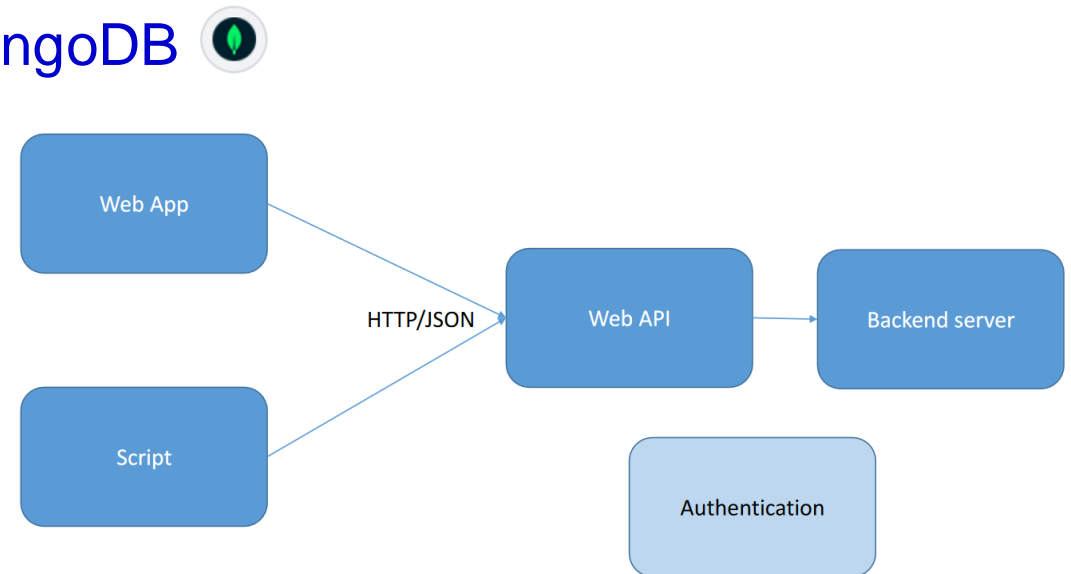
 - Operated in uuCloud, which is powered by MS Azure

- Terminology

 - A server-side functionality is called uuCmd and is represented by an endpoint

 - The API can be called via HTTP requests (GET or POST method)

- Front End (scripts/web apps) is developed by the wider ITk community



Database Component/Test Set-up

- Database store production history per component including QC/QA tests
 - A hierarchical component structure reflects complex assembly
- Component information:
 - Essential: e.g. type, manufacturer, creation date & origin
 - Contingent: e.g. location, stage, defects
 - Related objects: parents, children, tests
- Test information:
 - Essential: e.g. pass/fail, date, location, user
 - Contingent: e.g. test parameters
 - Related objects: components
 - Note: tests done in different stages only needs to be set-up once
- Shipment information:
 - Sender, recipient, courier, tracking number, parts list
- Currently >400 component and >2000 test types defined

Sub-project	component types	test types
Strips	~200	~1000
Pixels	~200	~1100

Tools to interact with Production Database

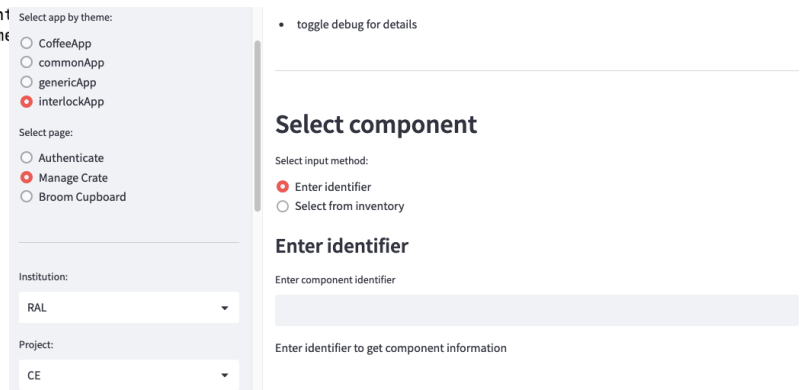
- 🐾 General purpose tools and tools for dedicated tasks developed by ITk collaboration
 - 🐾 These use modern interface tools for GUIs, webapps, notebooks and command-line scripts
- 🐾 Usage depends on subgroup/institutes
 - 🐾 Important to tailor tools to needs in diverse community (technicians, engineers, physicists)
- 🐾 Tools distributed via CERN's IT platforms & licences

 Python scripts distributed via git

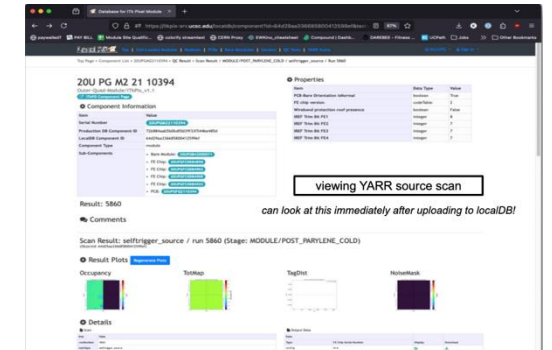
```
monika.wielers@HEPDOCK213 production_database_scripts % python3 registerComponent.py
```


```
[INFO]$ Running ITk Production Database component registration interface.
dbAccess: Getting token.
dbAccess: Token already exists in shell environment.
[INFO]$ Updating list of institutions.
[PROMPT]$ To always print the available input options for codes, please type 'y/Y' or t
ype 'n/N' to suppress this output:
n
[INFO]$ Use escape codes &PRINT to print the available options, &JSON to print
ent JSON for your component, or &CANCEL to cancel the registration at any time
[PROMPT]$ Enter your institution code:
RAL
[INFO]$ Using code: RAL (STFC Rutherford Appleton Laboratory)
```

Streamlit (multi-purpose tool) hosted on CERN OpenShift in docker 



LocalDB for pixel modules distributed via git



Flask for pixel modules hosted on CERN OpenShift in docker image 

Select	Serial number	Creation date	Bare module Serial Number	Flex Serial Number	FE ASIC Serial Numbers	Flex attach	Vis. Insp.	Mass
<input type="checkbox"/>	20UPGX06000122	2024-03-12						
<input type="checkbox"/>	20UPGX19110541	2023-08-25	20UPGBR9900001	20UPGPQ2110541				
<input type="checkbox"/>	20UPGX19110507	2023-08-25	20UPGBR9900004	20UPGPQ2110507				

Database Usage

🐾 Current information in database

🐾 Note: production for many parts has not yet started

registered components

Project	# components	# institute	# user
Pixel	~130k	~60	~300
Strips	~820k	~60	~450

uploaded tests

~9.5 · 10⁶

shipments

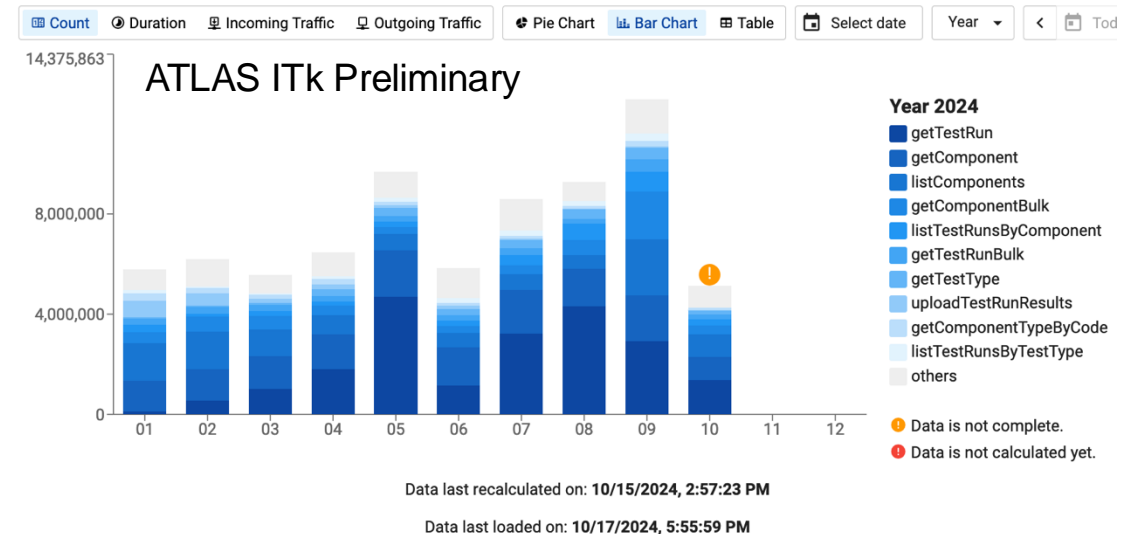
~8000

🐾 Number of requests sent to database

🐾 Reading out data far outstrips inputting

Top 3 commands

getTestRun	39%
GetComponent	21%
listComponents	12%



Reporting (done outside database)

🐾 Common tools to create reports

- 🐾 Flattening
- 🐾 Visualisation (tables, reports)
- 🐾 Distribution (creation of data panels)

🐾 Type of reports

- 🐾 On-demand reports
- 🐾 Scheduled reports
- 🐾 Alerts in case parts do not fulfill specifications

🐾 Reports make use of CERN licenses and platforms

- 🐾 Reporting (python 🐍) code stored on gitLab 🍌
- 🐾 Automated reports run in containers on Openshift 🔄
- 🐾 Generated reports uploaded to EOS 📦
- 🐾 Website generated via GitLab 🍌 and collected in central repository

- 🐾 Meta-data includes creator, code, upload timestamp

ITk Production Database Reporting Hub

Links >

Reports ▾

common_electronics >

common_mechanics >

other >

pixels ▾

2.1.10_Data_Trans

2.1.11_DAQ_&_DCS

2.1.1_Sensors

2.1.2_FEs

2.1.3_Hyb_&_A'bly

2.1.4_Services

2.1.5_Loc_Sups

2.1.6_Global_Tools

2.1.7_Integration

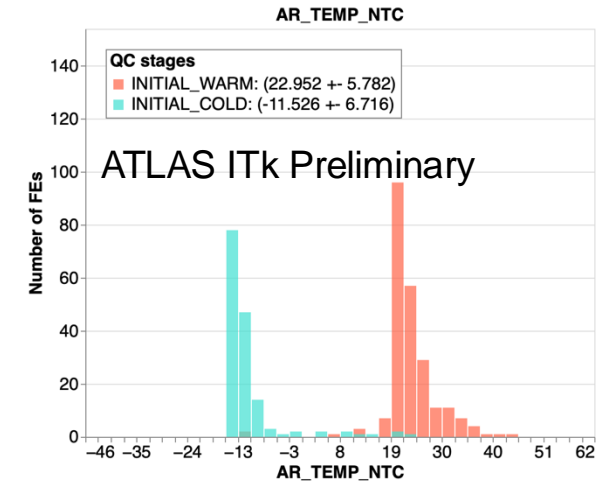
2.1.8_Offline

The screenshot shows a report upload record in the ITk Reporting Hub. At the top, it says 'Uploaded 2024-07-01'. Below that, the report type is 'test'. The upload date and time are '2024-07-01 @ 10:24 by doyeong'. The source code is a GitLab URL: 'https://gitlab.cern.ch/atlas-itk/pixel/module/module-qc-statistical-tools'. An optional manual WBS code is '2.1.3'. Under 'Upload History', there is one entry: 'basic-qc-plots-ANALOG_READBACK.html (previous)'. At the bottom, another 'Uploaded 2024-07-01' entry is partially visible.

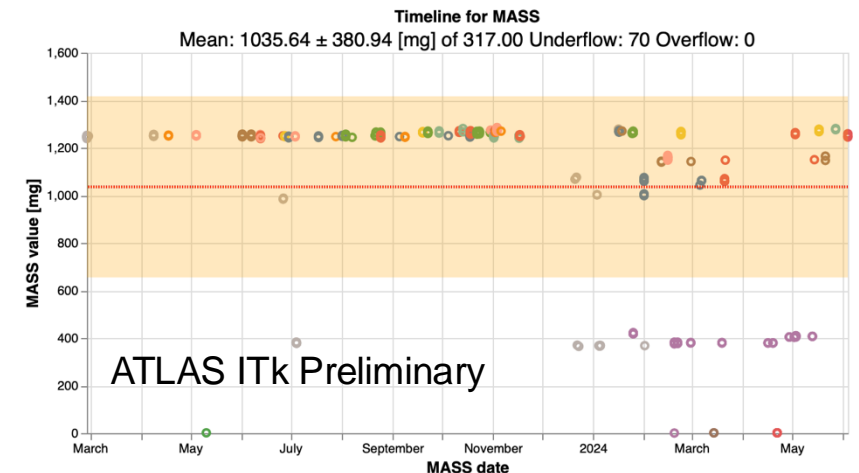
Reporting

- Various generation scripts with common principles
 - Public (to users): Reports visible to ITk members
 - Accessible: common (user) access to reports, without common (developer) format.
 - Credited: who, how, when
- Different types of reports target different audiences
 - Different level of details needed for sub-project coordinators, production management group, sub-project analysers, institute level
- Reporting being done for
 - Checking data consistency
 - Monitoring of parts location and production rates
 - Monitoring of production quality and yields
- Reporting takes up lots of resources and soon will move to backup copy hosted at CERN in OpenStack

Testing temperature in cold/warm



Timeline of mass measurement



Data Mining

- Important to look at data in case issues during production/operations arise
 - Try to find correlations in data
 - In which area of the component did the problem occur
 - Does it affect one batch only
 - Does it affect parts from one vendor only
 - Which test results does it affect
 - Does it affect certain production version
 - Does it happen at the same stage in the production flow
 -
- Successfully used database for these purposes in strips and pixel sub-projects
 - Database data-mining essential to understand issues!

Conclusions

- ATLAS ITk production database is a fully functional distributed data management system in use for the last 4 years by ~100 institutes
- Database heavily used for registration and test uploads
 - $\sim 10^6$ components registered and nearly 10×10^6 test results in database
- Vital tool for monitoring production to ensure detector build on time and to specifications
- Database will be operational for the remaining years of construction and during HL-LHC operation
 - Long-term maintenance of database and the front-end tools used for reporting needed