

Experience with ITk production database in ATLAS



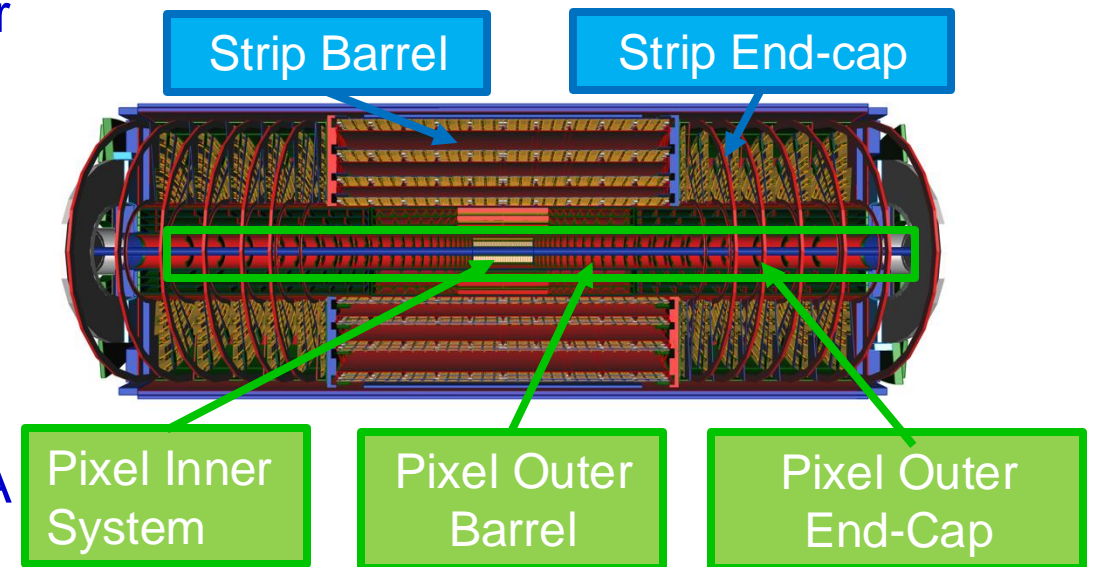
UK EIC meeting, March 12

Monika Wielers (STFC – RAL)

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 HL-LHC
 - 165 m² of Si detector coverage
 - Strips: 4 layers in barrel, 6 disks in fwd/bkd direction
 - Pixels: 5 layers in barrel, up to 18 rings in forward/backward direction
- All parts used in detector (readout 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 (incl industry partners)
- In total ~10⁶ parts to be produced
 - Need for a **database** to store part 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

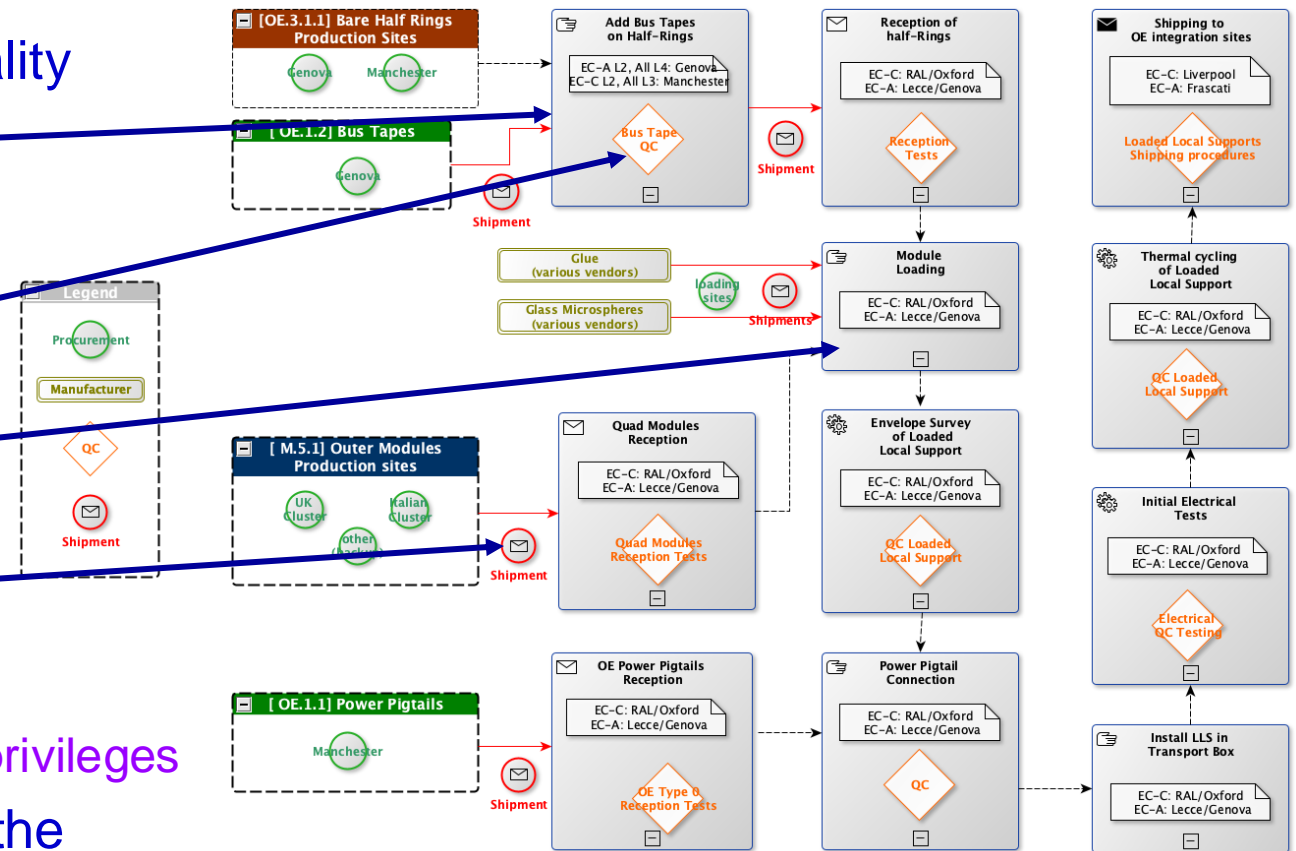
Database Development

- 🐾 ATLAS opted to have the database developed by commercial company
 - 🐾 Not enough manpower was available to develop one on our own
 - 🐾 DB not a delivery for the experiment
- 🐾 Pros
 - 🐾 No need to find the manpower to develop one
 - 🐾 Developed according to standards
- 🐾 Cons
 - 🐾 Costs (also includes costs to run the database during detector operation)
 - 🐾 Cannot use infrastructure in computing centre at laboratory hosting experiment
 - 🐾 Need to oversee development to ensure product well tailored to your community
 - 🐾 Our working environment very different from standard companies
- 🐾 General
 - 🐾 Need well written requirement document before project starts
 - 🐾 Includes how uploads are expected to be done and handling the pass/fail test decision
 - 🐾 Decide what you need being developed by vendor and what by your community, e.g. ask for API commands and expect scripts/web apps for data upload / reporting is done in your community

Production Database: Requirements

❖ Main production database (PDB) functionality

- ❖ Registering parts (components)
 - ❖ Dealing with batches
 - ❖ Serial number scheme
 - ❖ Tracking build status (stage)
 - ❖ Adding test results
 - ❖ 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
- ❖ We now have a database based on MongoDB running in cloud in place



Production Database Component and Test Set-up

- Already >400 component and >2000 test types defined

- Properties and parameters set-up for everything which needs being recorded

Sub-project	component types	test types
Strips	226	976
Pixels	237	1184

- Possibility to add attachments to components or tests

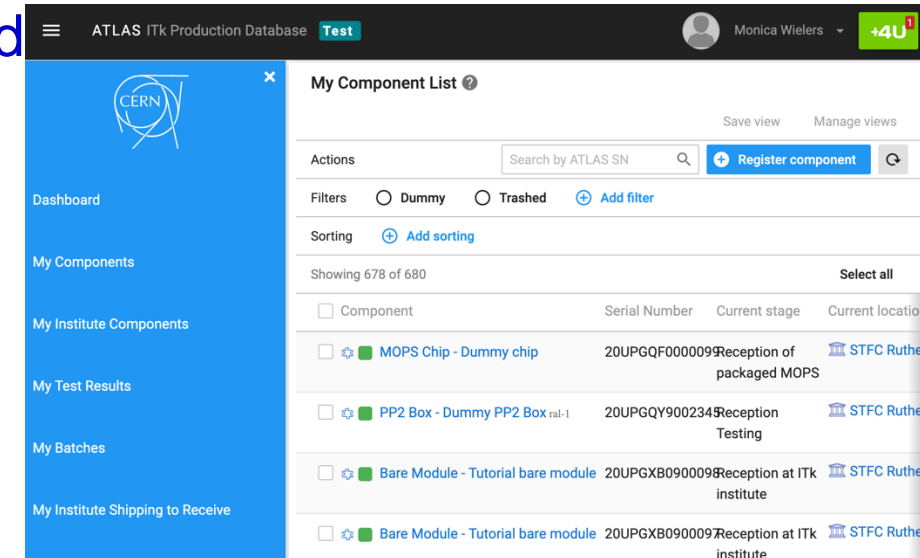
- Needed for visual inspection images, raw data for some electrical tests to redo test decisions, configuration files
- Found out this is more commonly used than expected

- Handled in 2 ways to keep costs for low

- Usage of binary storage attached to database if file < 64kB
- Use CERN EOS cloud space for files > 64kB (not integrated as integral part in database)

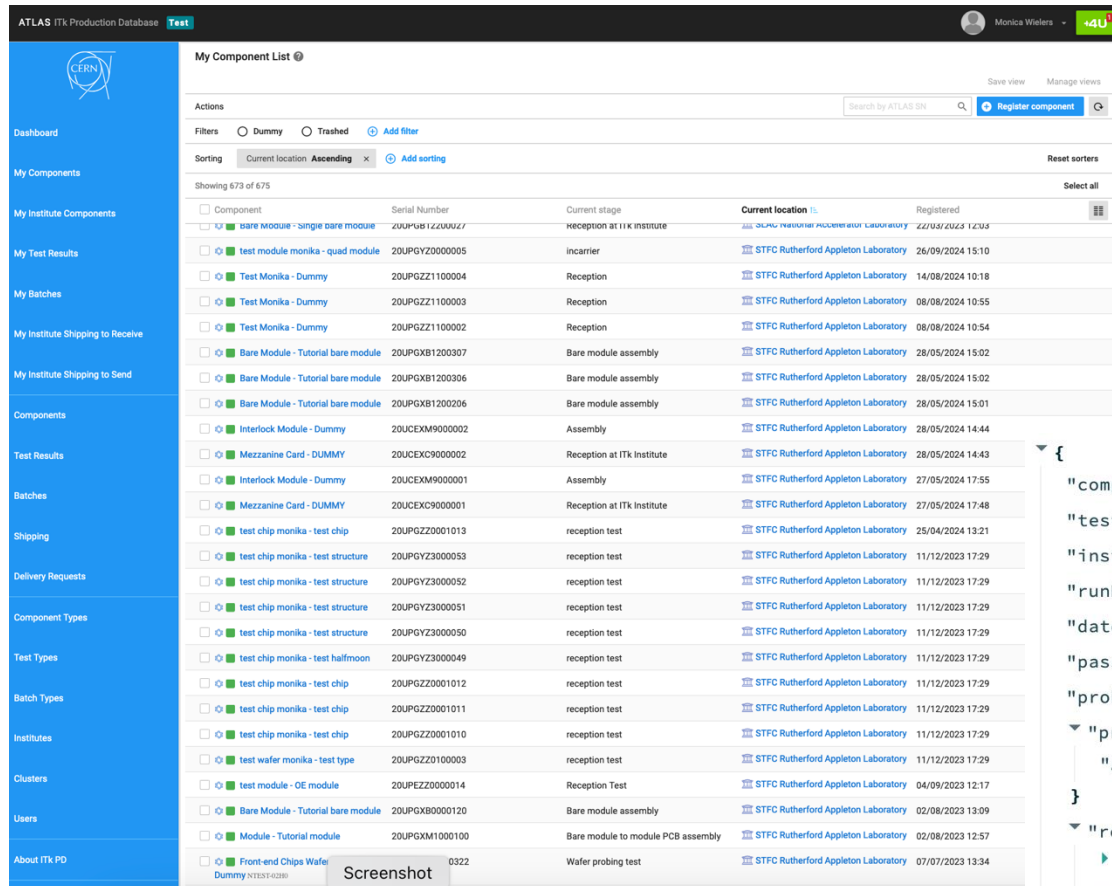
- Needs careful monitoring and clear rules to ensure appropriate use

- The typical answer 'just in case' is not good enough



Production Database Interactions

🐾 User interface



The screenshot shows the ATLAS ITK Production Database web interface. The left sidebar contains navigation links: Dashboard, My Components, My Institute Components, My Test Results, My Batches, My Institute Shipping to Receive, My Institute Shipping to Send, Components, Test Results, Batches, Shipping, Delivery Requests, Component Types, Test Types, Batch Types, Institutes, Clusters, Users, and About ITK PD. The main content area is titled 'My Component List' and shows a table of components. The table has columns for Component, Serial Number, Current stage, Current location, and Registered. The table is sorted by Current location in ascending order. The table shows 673 components. The first few rows are:

Component	Serial Number	Current stage	Current location	Registered
Component	Serial Number	Current stage	Current location	Registered
test module monika - quad module	20UPGY23000005	incarrier	STFC Rutherford Appleton Laboratory	26/09/2024 15:10
test module monika - quad module	20UPGZZ1100004	Reception	STFC Rutherford Appleton Laboratory	14/08/2024 10:18
test module monika - quad module	20UPGZZ1100003	Reception	STFC Rutherford Appleton Laboratory	08/08/2024 10:55
test module monika - quad module	20UPGZZ1100002	Reception	STFC Rutherford Appleton Laboratory	08/08/2024 10:54
Bare Module - Tutorial bare module	20UPGXB1200307	Bare module assembly	STFC Rutherford Appleton Laboratory	28/05/2024 15:02
Bare Module - Tutorial bare module	20UPGXB1200306	Bare module assembly	STFC Rutherford Appleton Laboratory	28/05/2024 15:02
Bare Module - Tutorial bare module	20UPGXB1200206	Bare module assembly	STFC Rutherford Appleton Laboratory	28/05/2024 15:01
Interlock Module - Dummy	20UCEXM9000002	Assembly	STFC Rutherford Appleton Laboratory	28/05/2024 14:44
Interlock Module - Dummy	20UCEXM9000001	Assembly	STFC Rutherford Appleton Laboratory	27/05/2024 17:55
Mezzanine Card - DUMMY	20UCEXC9000001	Reception at ITK Institute	STFC Rutherford Appleton Laboratory	27/05/2024 17:48
test chip monika - test chip	20UPGZZ0001013	reception test	STFC Rutherford Appleton Laboratory	25/04/2024 13:21
test chip monika - test structure	20UPGY23000053	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test chip monika - test structure	20UPGY23000052	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test chip monika - test structure	20UPGY23000051	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test chip monika - test structure	20UPGY23000050	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test chip monika - test halfmoon	20UPGY23000049	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test chip monika - test chip	20UPGZZ0001012	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test chip monika - test chip	20UPGZZ0001011	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test chip monika - test chip	20UPGZZ0001010	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test wafer monika - test type	20UPGZZ0100003	reception test	STFC Rutherford Appleton Laboratory	11/12/2023 17:29
test module - OE module	20UPEZZ0000014	Reception Test	STFC Rutherford Appleton Laboratory	04/09/2023 12:17
Bare Module - Tutorial bare module	20UPGXB0000120	Bare module assembly	STFC Rutherford Appleton Laboratory	02/08/2023 13:09
Module - Tutorial module	20UPGXM1000100	Bare module to module PCB assembly	STFC Rutherford Appleton Laboratory	02/08/2023 12:57
Front-end Chips Wafer	0322	Wafer probing test	STFC Rutherford Appleton Laboratory	07/07/2023 13:34

🐾 API commands

🐾 Mostly often used in our community

- 🐾 Easy to develop scripts
- 🐾 Allows development of custom scripts/web apps for data uploads

🐾 Allows

- 🐾 Batch upload via csv/excel files or google sheets
- 🐾 Machine output can be converted to format needed for DB uploads
- 🐾 Sanity checks help avoiding human mistakes

🐾 Uploads done using json format

🐾 ITk community also developed api wrappers (itkdb, dbAccess)

- 🐾 Eases script developments

```
{
  "component": "...",
  "testType": "ADC_TEST",
  "institution": "...",
  "runNumber": "...",
  "date": "2024-10-02T09:32:49.343Z",
  "passed": true,
  "problems": false,
  "properties": {
    "ANALYSIS_VERSION": null
  },
  "results": {
    "FINE_GRAINED_CALIBRATION": {},
    "ADC_CHANNEL_STATUS": true,
    "ADC_OFFSET": 0,
    "ADC_GAIN": 0,
    "TEMPERATURE": 0
  }
}
```

Tools to interact with Production Database

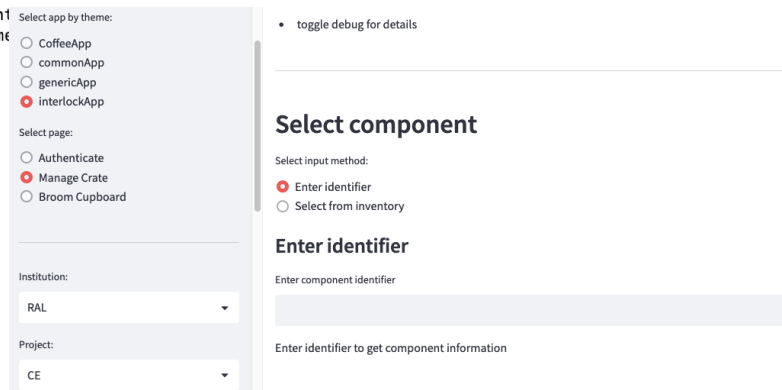
- 🦊 General purpose tools and tools for dedicated tasks developed by ITk collaboration
 - 🦊 GUIs, webapps, notebooks and command-line scripts
- 🦊 Usage depends on subgroup/institutes
 - 🦊 Important to tailor tools to needs in diverse community (technicians, engineers, physicists)
- 🦊 Tools distributed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

 Python scripts distributed via git

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

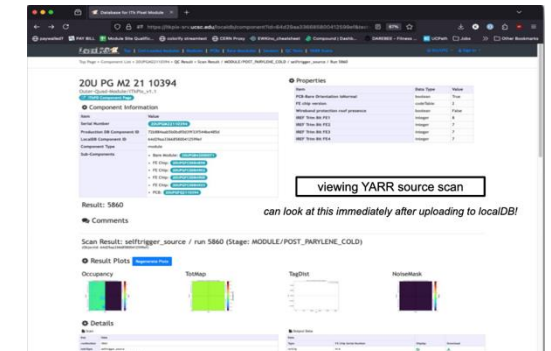
```
[INFO]$ Running ITk Production Database component registration interface.
dbAccess: Getting token.
dbAccess: Token already exists in shell environment.
[INFO]$ Updating list of institutions.
[PROMPT]$ To always print the available input options for codes, please type 'y/Y' or t
ype 'n/N' to suppress 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 

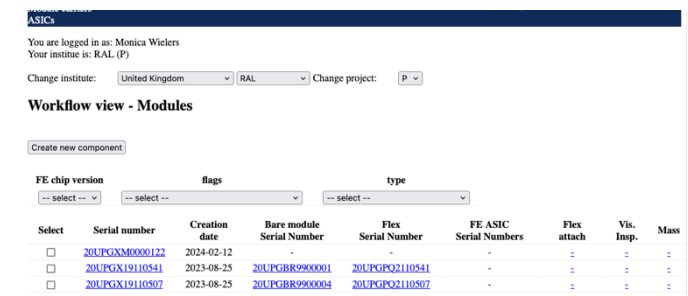


The image shows a Streamlit web application for registering components. It has a sidebar with 'Select app by theme' (CoffeeApp, commonApp, genericApp, interlockApp) and 'Select page' (Authenticate, Manage Crate, Broom Cupboard). The main area has 'Select component' with 'Enter identifier' and 'Select from inventory' options. Below is a text input for 'Enter component identifier' and a button 'Enter identifier to get component information'.

LocalDB for pixel modules distributed
via git



Flask for pixel modules hosted on
CERN OpenShift in docker image 



The image shows a Flask web application for pixel modules. It has a sidebar with 'Workflow view - Modules' and 'Create new component'. The main area shows a table of modules with columns: Select, Serial number, Creation date, Bare module Serial Number, Flex Serial Number, FE ASIC, Flex attach, Vis. Insp., and Mass.

Select	Serial number	Creation date	Bare module Serial Number	Flex Serial Number	FE ASIC	Flex attach	Vis. Insp.	Mass
<input type="checkbox"/>	20UPGX0000122	2024-02-12						
<input type="checkbox"/>	20UPGX19110541	2023-08-25	20UPGBR9900001	20UPGQ2110541				
<input type="checkbox"/>	20UPGX19110507	2023-08-25	20UPGBR9900004	20UPGQ2110507				

Database Usage

- Current information in database
 - Note: production for many parts has not yet started

registered components

Project	# components	# institute	# user
Pixel	~170k	~70	~300
Strips	~860k	~70	~450

uploaded tests

~9.8·10⁶

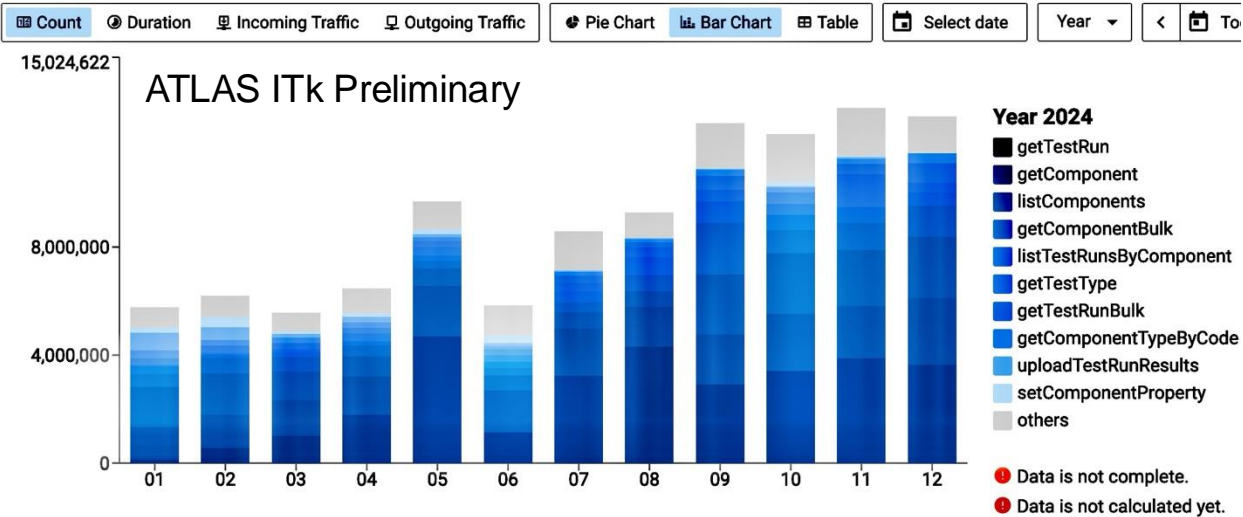
shipments

~9000

- Number of requests sent to database
 - Reading out data far outstrips inputting

Top 3 commands

getTestRun	39%
getComponent	21%
listComponents	12%



Shipments

🐾 Integral part of database

🐾 Important feature for production as it allows to track location of parts

🐾 This is in particular important for parts falling under dual-use export licences

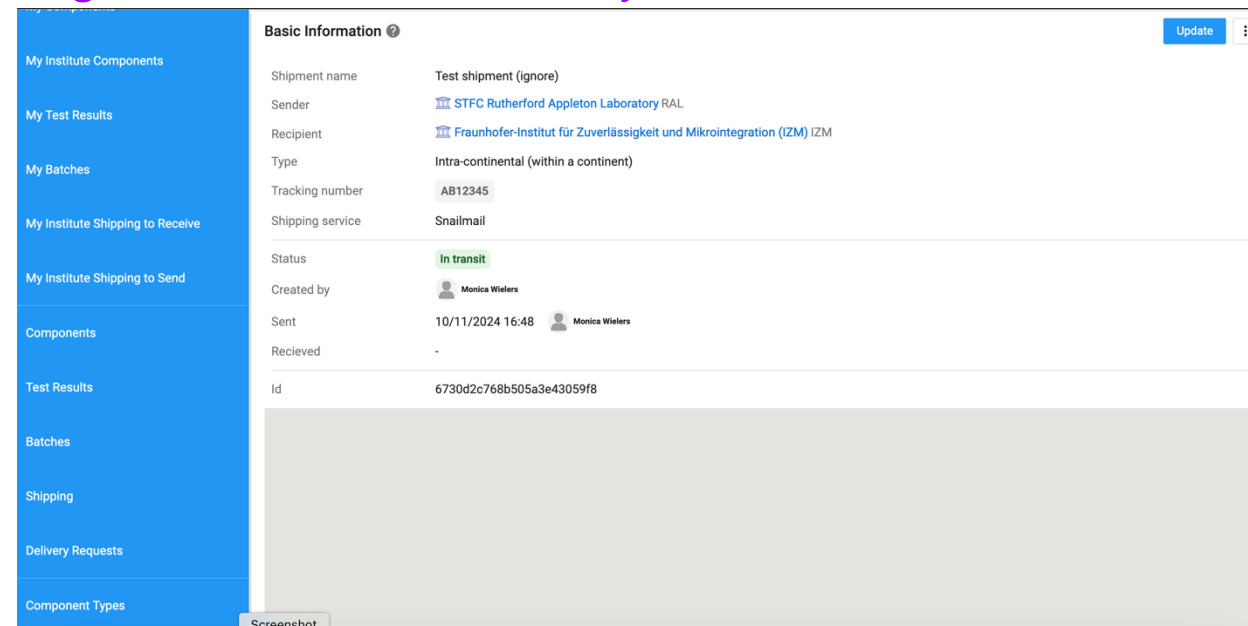
- Implies all parts need to be shipped back to institute who bought these parts (CERN in our case), can be individual items or larger structures on which these parts were mounted
- Make sure all parts falling under this have a unique vendor number to ensure traceability

🐾 When sending a shipment, people at receiving institute are notified by mail

🐾 This includes the information on shipping service and tracking number

🐾 When accepting a delivery, user fills out checklist to track if any damage occurred during shipment

🐾 Upon delivery, sending institute is notified by mail



The screenshot displays a web application interface for managing shipments. On the left is a blue sidebar with navigation links: 'My Institute Components', 'My Test Results', 'My Batches', 'My Institute Shipping to Receive', 'My Institute Shipping to Send', 'Components', 'Test Results', 'Batches', 'Shipping', 'Delivery Requests', and 'Component Types'. The main content area is titled 'Basic Information' and contains the following details:

Shipment name	Test shipment (ignore)
Sender	STFC Rutherford Appleton Laboratory RAL
Recipient	Fraunhofer-Institut für Zuverlässigkeit und Mikrointegration (IZM) IZM
Type	Intra-continental (within a continent)
Tracking number	AB12345
Shipping service	Snailmail
Status	In transit
Created by	Monica Wielers
Sent	10/11/2024 16:48 Monica Wielers
Received	-
Id	6730d2c768b505a3e43059f8

At the top right of the main area, there is an 'Update' button and a menu icon. Below the main information table, there is a large grey rectangular area, likely for a checklist or additional notes. A 'Screenshot' label is visible at the bottom left of the main content area.

Delivery Requests

- This is a very useful additional feature
 - Way to ensure all institutes have enough parts in hand
 - Dashboards can show how many parts are available and ready to be shipped to building sites
- Institutes can request parts from other institutes, or are requested by part flow managers
 - These requests are typically discussed in the strips and pixel production group meetings. Ensures parts are shipped in a fair way to all institutes involved
- Approval of delivery requests (including the number of parts which are approved) done by part flow managers
- Once approved, the delivery request is added to a shipment and closed once fulfilled

Create a delivery request ?

Recipient*

Fraunhofer-Institut für Zuverlässigkeit und Mikrointegration (IZM) IZM

Name*

Test Delivery request Monika

Sender*

STFC Rutherford Appleton Laboratory RAL

Component type*

test chip monika TESTCHIPMONIKA

Type

test chip ×

Requested quantity

1

−

+

Description

B *I* U ~~ABC~~ \times_2 \times^2 14 \equiv P [Link](#) A

This is a test

Cancel

Create

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

All reports end up in single entry point (reporting hub)

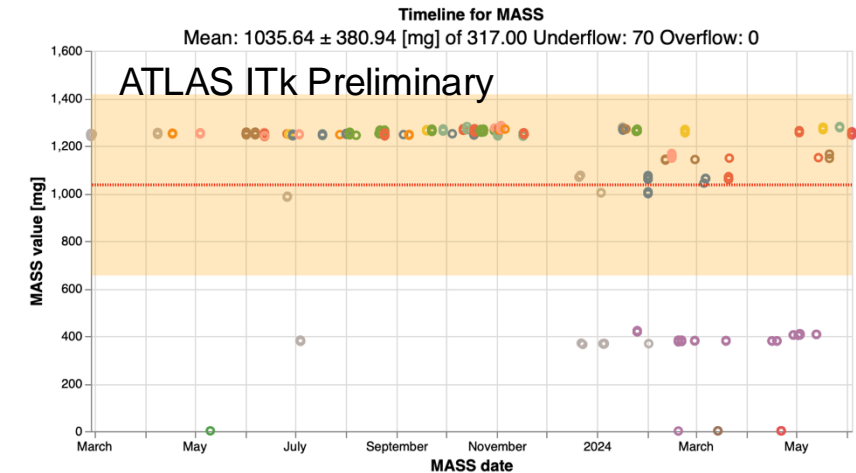
- Different level of details needed for sub-project coordinators, production management group, sub-project analysers, institute level

Reporting being done for

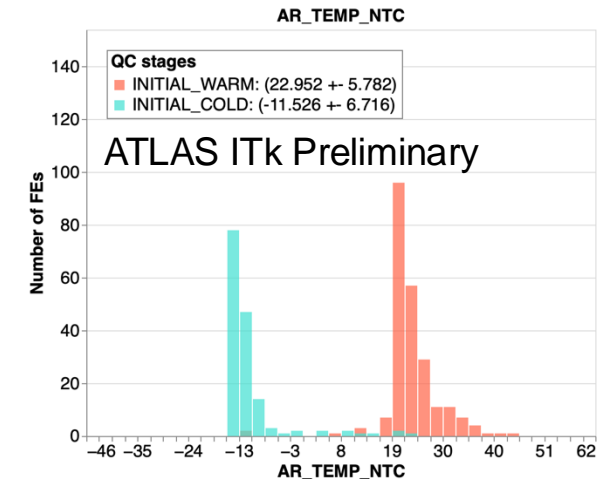
- Checking data consistency
- Monitoring of parts location and production rates
- Monitoring of production quality and yields

Reporting takes up lots of resources! We will use the weekly backup copy hosted at CERN this year

Timeline of mass measurement



Testing temperature in cold/warm



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!

What I like/like less about the database

- DB ensures all the component information and QC/QA test results are in one central place
- Accessible via API commands for script/webapp development
- Front-end and back-end are well separated
 - Back-end maintained by vendor
 - Front-end by ITk community
- Shipment and delivery requests part of the database features
 - Allows tracking of dual-use export licenses
- As DB comes from vendor API commands are their proprietary software and cannot be used on the CERN-based backup
- DB hosted by vendor, so no close integration with eos space for attachments at CERN possible
- We did not provide some recommended/example tools to interact with the database in time
 - Now everyone uses his/her solution, which makes it more complicated that the test pass/fail decisions are done in the same way for all component types with a simple interface to access to the analysis code

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

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

