

ALICE Event Display

from the legacy ROOT-based visualization
to the web-based application

Julian Myrcha

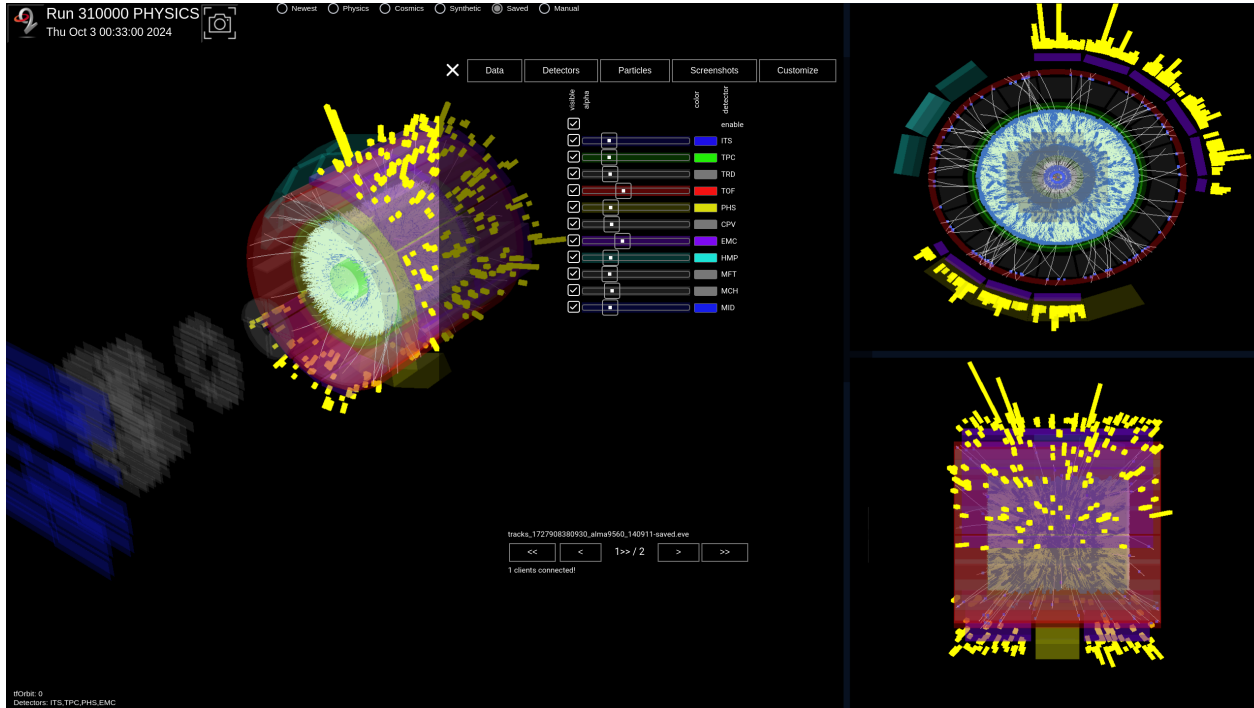
Institute of Computer Science

22.10.2024

on behalf of the ALICE Collaboration



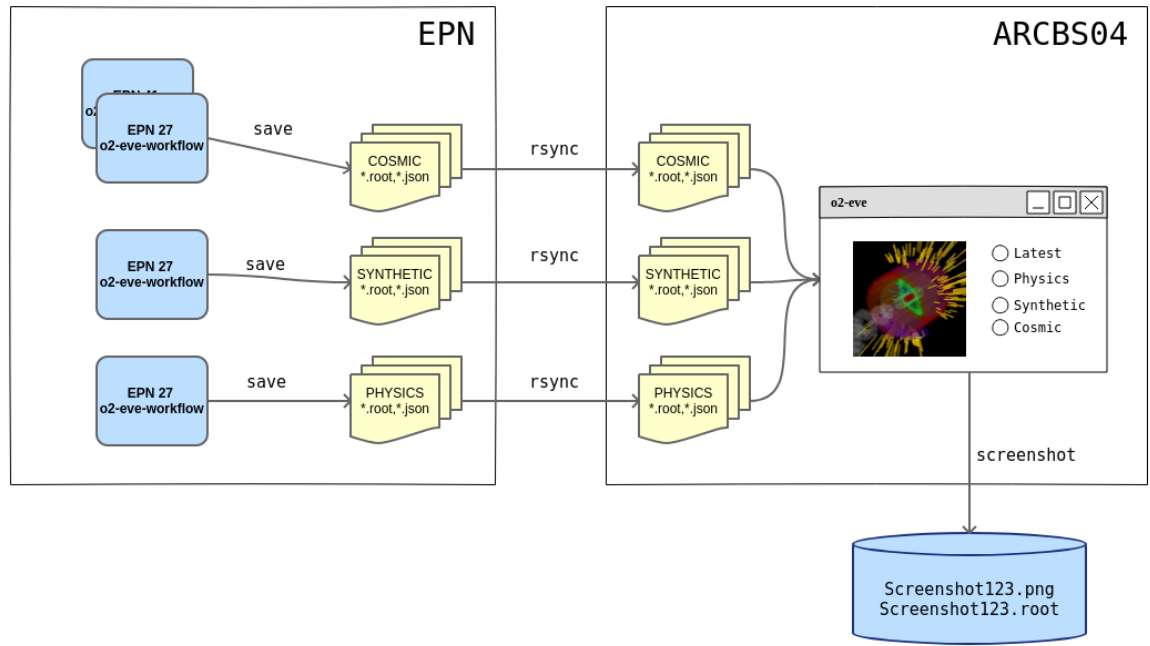
- o2-eve architecture presentation
- transition process between previous (beginning of run 3) and current visualisation
- reasons why changes was done





LEGACY ARCHITECTURE

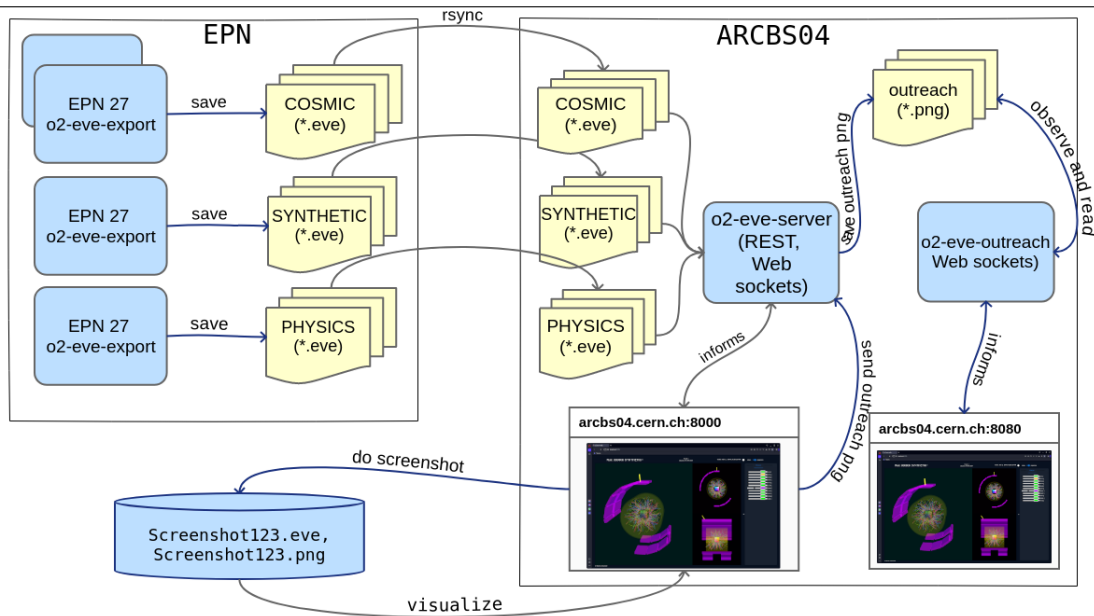
- root based visualisation
 - separation between visualisation and data acquisition
 - 3D visualisation routines in ROOT not commonly used - no knowledge base in case of troubles
 - slow development as any change require recompile steps:
-
- very last node in workflow produce files with visualisation data
 - files are transferred to visualisation machine
 - newest file is visualized





FINAL ARCHITECTURE

- uses industry standard libraries ([Three.js](#) and [React.js](#))
 - huge knowledge base
 - fast development as hot replace allows changes in running application
- smoothly replaces previous solution
- no installation for end user
 - only a browser and web access to the server needed

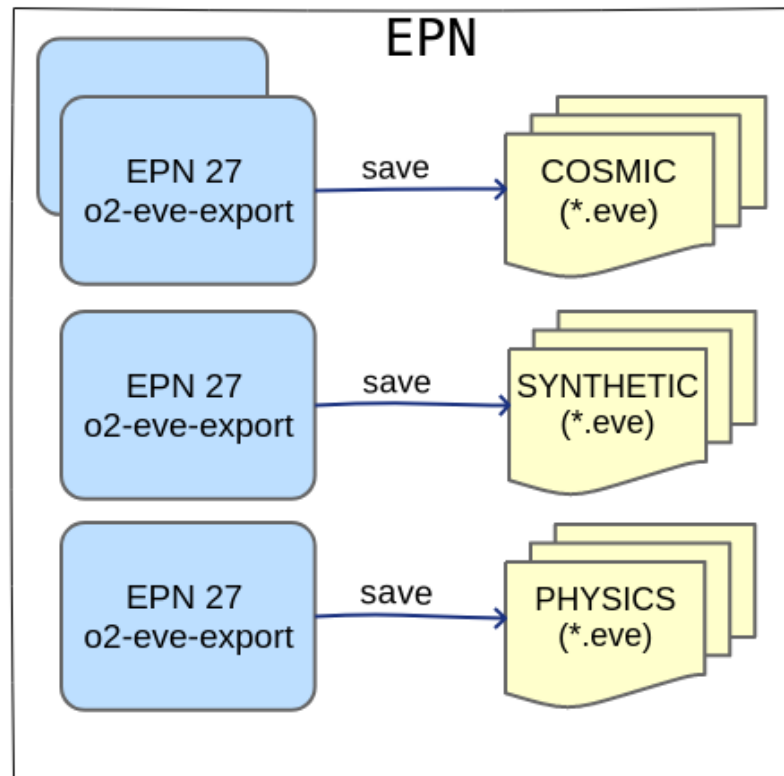


possible alternatives: [JSROOT](https://root.cern/manual/jsroot/) <https://root.cern/manual/jsroot/>



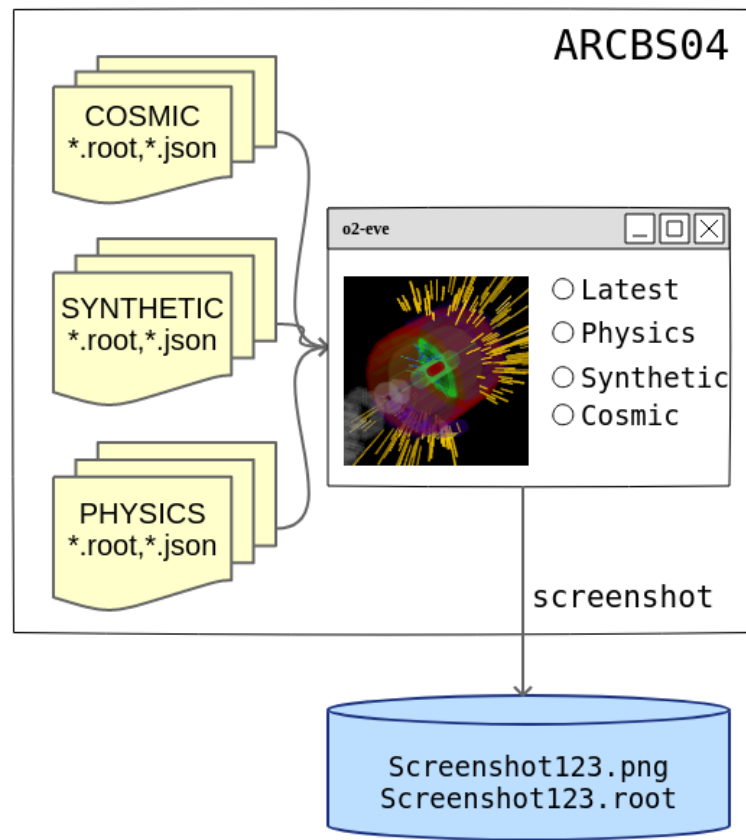
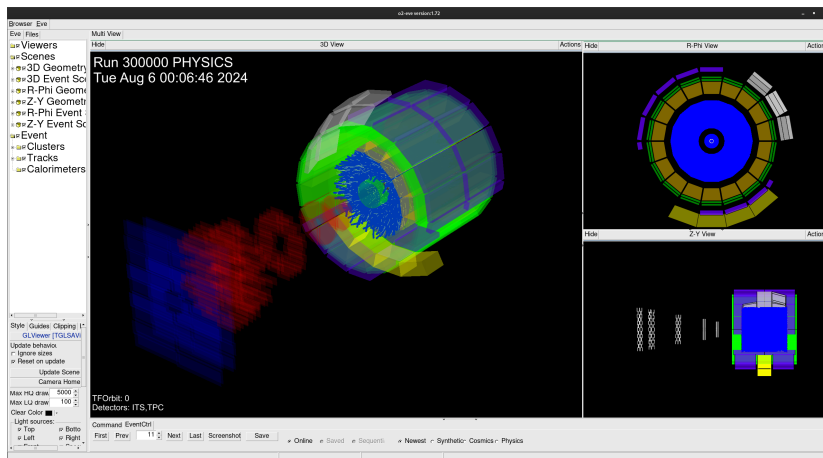
PREPARED TOOLS

- `o2-eve-export` - (workflow) prepare data, save as files
- `o2-eve` - (desktop) legacy visualisation tool
- `o2-eve` - (web) web based visualisation tool



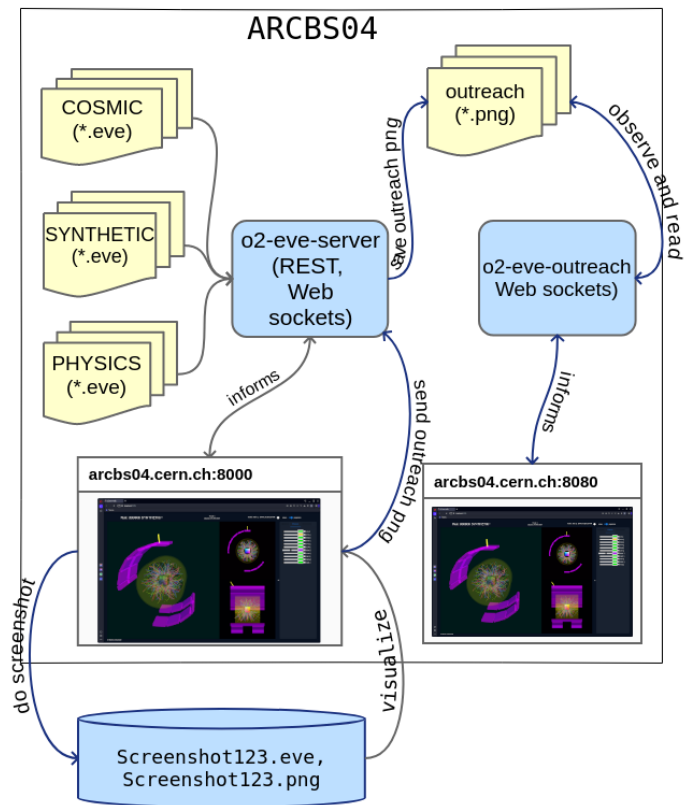
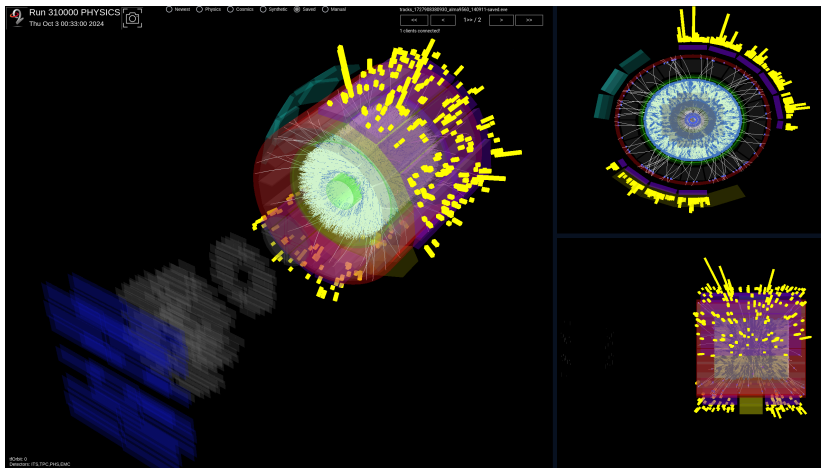
PREPARED TOOLS

- `o2-eve-export` - (workflow) prepare data, save as files
- `o2-eve` - (desktop) legacy visualisation tool
- `o2-eve` - (web) web based visualisation tool



PREPARED TOOLS

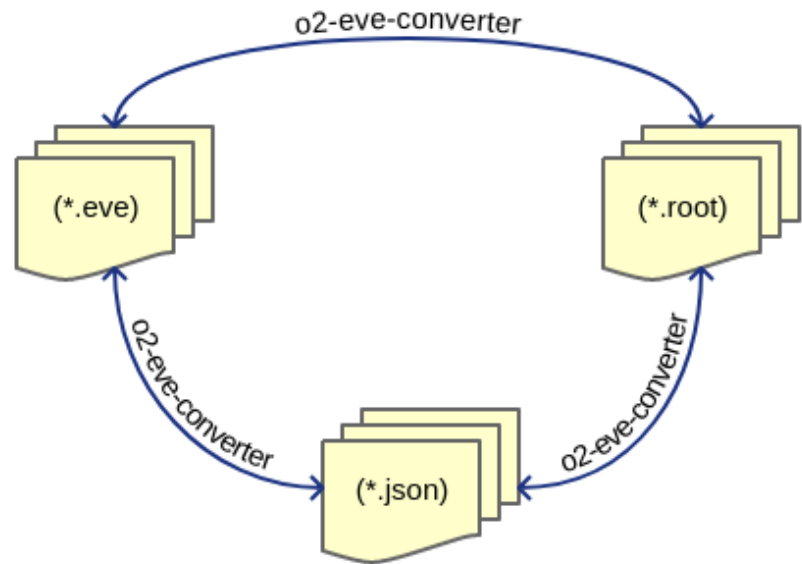
- `o2-eve-export` - (workflow) prepare data, save as files
- `o2-eve` - (desktop) legacy visualisation tool
- `o2-eve` - (web) web based visualisation tool





PREPARED TOOLS

- `o2-eve-export` - (workflow) prepare data, save as files
- `o2-eve` - (desktop) legacy visualisation tool
- `o2-eve` - (web) web based visualisation tool
- `o2-eve-convert` - (command line) conversion tool
 - single file
 - folder
 - folder (live)





FILE FORMATS

- *.json - easy to create test data, easy to view/modify, very slow for large files (like 800MB)

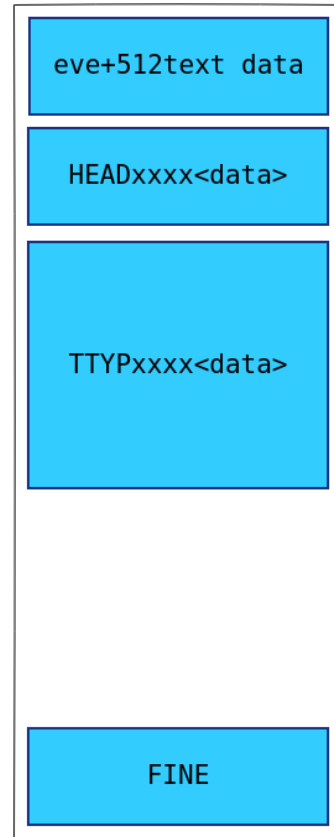


FILE FORMATS

- `*.json` - easy to create test data, easy to view/modify, very slow for large files (like 800MB)
- `*.root` - efficient reading, tools (ROOT) to view contents, require ROOT software, compression, data should be reorganised for visualisation

FILE FORMATS

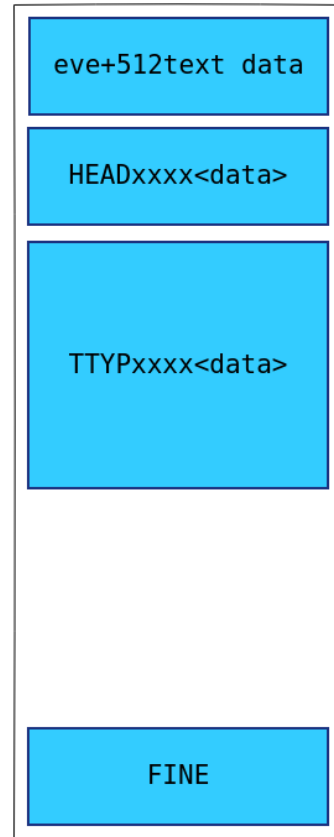
- *.json - easy to create test data, easy to view/modify, very slow for large files (like 800MB)
- *.root - efficient reading, tools (ROOT) to view contents, require ROOT software, compression, data should be reorganised for visualisation
- o2-eve - proprietary format following *.png chunks approach, data ready for visualisation (bytes are being copied into OpenGL structures)
 - it is possible to create private chunks - as not recognized they will be ignored in visualisation
 - it is possible to add new type of chunks - server sending data for visualisation may compute value from other chunks or provide defaults if source do not contain them (like old data)



FILE FORMATS

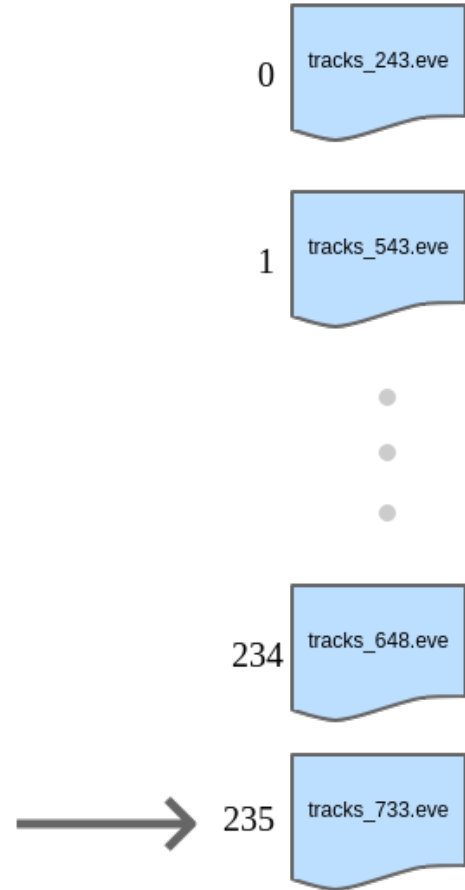
- *.json - easy to create test data, easy to view/modify, very slow for large files (like 800MB)
- *.root - efficient reading, tools (ROOT) to view contents, require ROOT software, compression, data should be reorganised for visualisation
- o2-eve - proprietary format following *.png chunks approach, data ready for visualisation (bytes are being copied into OpenGL structures)
 - it is possible to create private chunks - as not recognized they will be ignored in visualisation
 - it is possible to add new type of chunks - server sending data for visualisation may compute value from other chunks or provide defaults if source do not contain them (like old data)
 - first 512 bytes are treated as text (comment) which can be retrieved using linux head tool -> **no software needed to see information**

```
1 head -c 512 tracks_1726470156940_epn023.internal_2784411.eve
```



FOLDERS

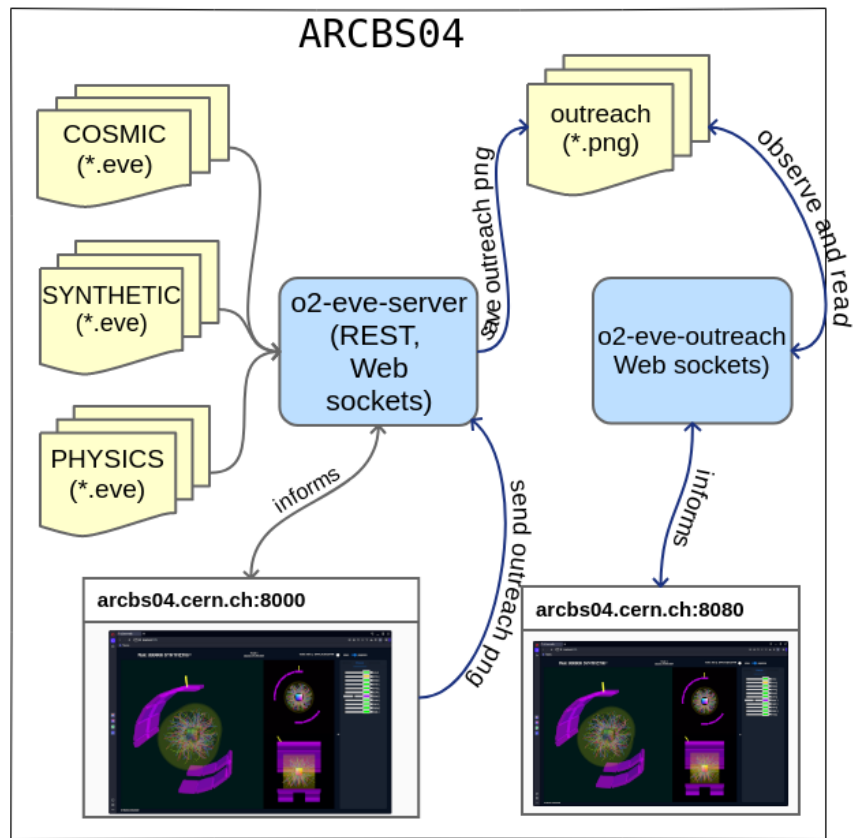
- we need to have access to data from last couple of minutes:
 - to make a screenshot of nice event
 - data file is saved together with screenshot for future recreation of screenshot with different settings
- FIFO is applied to separate folders: **physics**, **cosmics**, **synthetic**
 - we always have access to last visualisation of each type





FINAL ARCHITECTURE

- `o2-eve-server` serve web application:
 - visualisation in Control Room
 - distributed visualisation for CERN users on their own machines
- `o2-eve-outreach` serve images:
 - images created in web application
 - images refreshed simultaneously with the image in Control Room
 - used when user interaction not needed



CONCLUSIONS

- web `o2-eve` is much better solution (for maintenance, testability, adding new features)
- the amount of code was greatly reduced
- component architecture opens possibility to enhance tool with other visual effects
 - using rich infrastructure of `Three.js` it is easily to achieve rich visual effects
 - using `React.js` library it is easy to create user interface elements
 - component based approach used to increase testability

