Progress on the IRF3 Data Format

T. Hassan

IRF Basics

- The Instrument Response Function relates the array reconstructed quantities with the parameters of the source emitted photons

$$R_\gamma(\theta', \phi', E'|\theta, \phi, E) = A_\gamma(\theta, \phi, E) \times PSF(\theta', \phi'|\theta, \phi, E) \times D(E'|\theta, \phi, E)$$

J. Knödlseder

- The IRF elements are:
  - Effective area
  - Energy dispersion
  - Direction dispersion (PSF)
  - Background “acceptance”

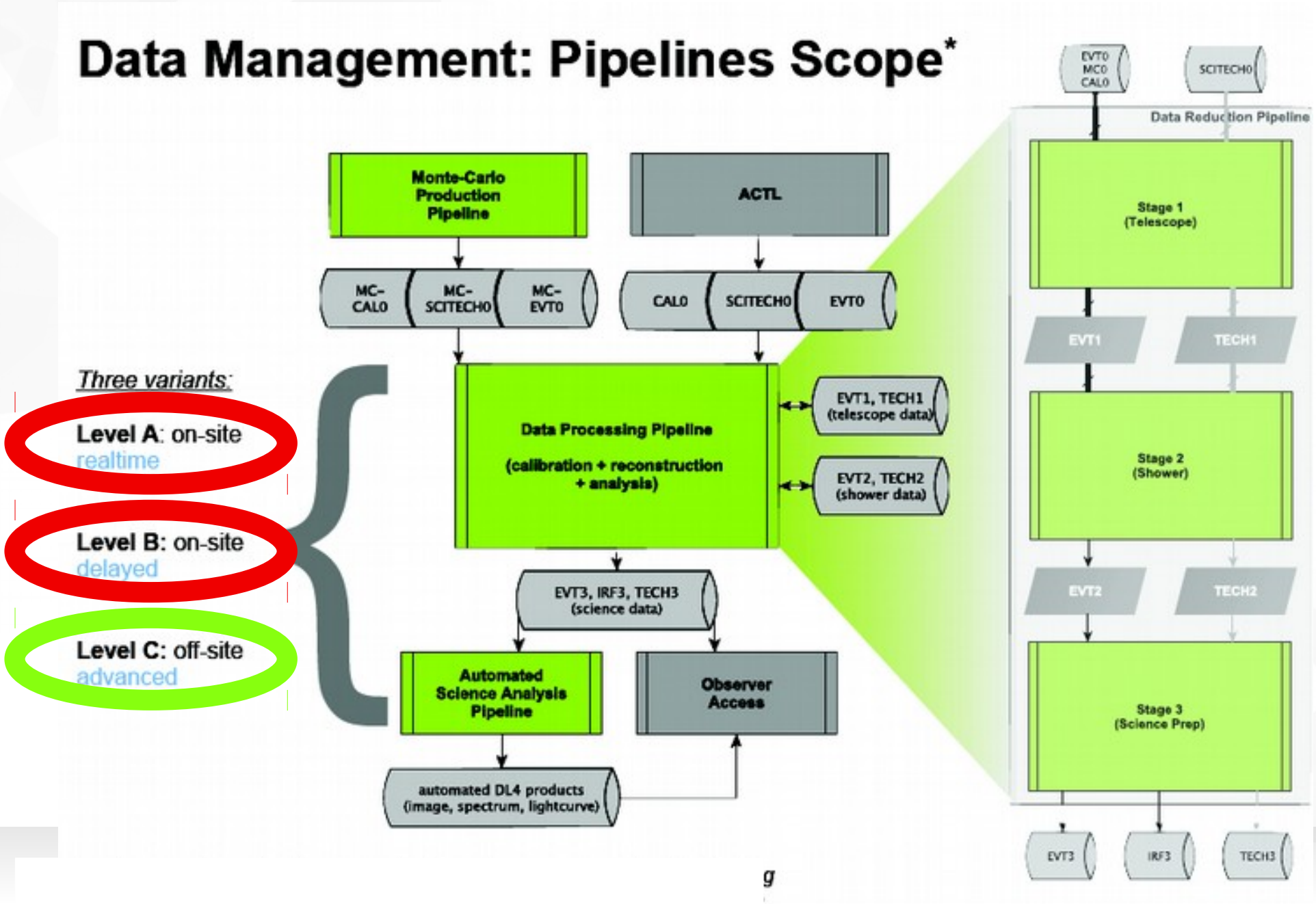
IRF Basics

- DL3 = EVT3 + IRF3 + TECH3

- Unlike spaceborne instruments, IACT IRFs are **highly** influenced by external parameters

- IRF3 will be required by:
    - Any High-level analysis, using DL3 data
    - Real-time/on-site analysis (no available tailored MC)
    - Scientific forecast (proposals by guest observers)

IRF Basics


Data Management: Pipelines Scope*
Monte-Carlo Production Pipeline
ACTL
MC-CAL0
MC-SCITECH0
MC-EVT0
CAL0
SCITECH0
EVT0
Three variants:
Level A: on-site realtime
Level B: on-site delayed
Level C: off-site advanced
Data Processing Pipeline
(calibration + reconstruction + analysis)
EVT1, TECH1 (telescope data)
EVT2, TECH2 (shower data)
EVT3, IRF3, TECH3 (science data)
Automated Science Analysis Pipeline
Observer Access
automated DL4 products (image, spectrum, lightcurve)
EVT0 MC0 CAL0
SCITECH0
Data Reduction Pipeline
Stage 1 (Telescope)
EVT1
TECH1
Stage 2 (Shower)
EVT2
TECH2
Stage 3 (Science Prep)
EVT3
IRF3
TECH3

IRF format – not only CTA

- There is an actual need of an open IRF format for IACTs
  - Long term archive of current IACTs data products
  - Data sharing between collaborations
    - Cross-callibration
    - Combined analysis
    - ...
- Some effort has been already done:

`http://gamma-astro-data-formats.readthedocs.org/en/latest/irfs/index.html`

Ideal IRF format for IACTs

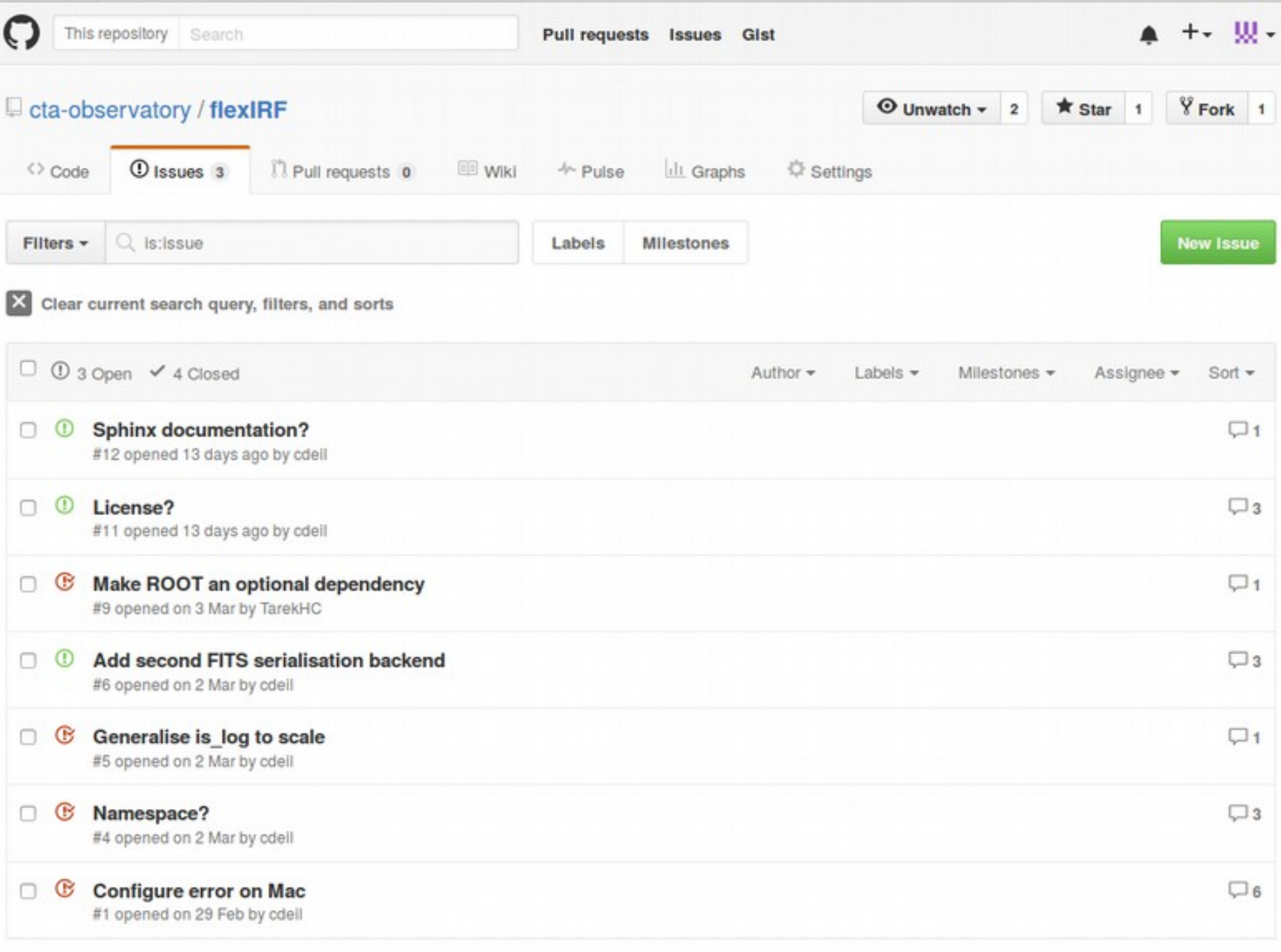
- It must be stored as a FITS file
- Should store all required IRF elements into a single (FITS) file
- Should store enough metadata to be reproducible (provenance)
- Should be able to handle different parameterizations
- Should be able to handle additional dependencies
- Should not change with time

Current IRF format status

- Two “formats” were proposed:
  - Essentially, no differences in **which** data to store, but **where** to store it
  - Different FITS serializations used: IMAGE and BINTABLE
- Regardless of the serialization, we need a decent IRF “builder”
- A (very preliminary) prototype was made public:
  - https://github.com/cta-observatory/flexIRF

flexIRF – What's new?

- Github provides an excellent forum:
    - Anyone can contribute!
    - Excellent forum to discuss the implementation and propose new features (thanks Christoph!)


This repository
Search
Pull requests
Issues
Gist
cta-observatory / flexIRF
Unwatch 2
Star 1
Fork 1
Code
Issues 3
Pull requests 0
Wiki
Pulse
Graphs
Settings
Filters
is:issue
Labels
Milestones
New issue
Clear current search query, filters, and sorts
3 Open
4 Closed
Author
Labels
Milestones
Assignee
Sort
Sphinx documentation?
#12 opened 13 days ago by cdeil
1
License?
#11 opened 13 days ago by cdeil
3
Make ROOT an optional dependency
#9 opened on 3 Mar by TarekHC
1
Add second FITS serialisation backend
#6 opened on 2 Mar by cdeil
3
Generalise is_log to scale
#5 opened on 2 Mar by cdeil
1
Namespace?
#4 opened on 2 Mar by cdeil
3
Configure error on Mac
#1 opened on 29 Feb by cdeil
6

flexIRF – What's new?

- Github provides an excellent forum:
    - Anyone can contribute!
    - Excellent forum to discuss the implementation and propose new features (thanks Christoph!)
- The code is currently able to:
    - read/write IRFs with any number of dependencies
    - Use any parameterization (just a proof-of-concept)
    - BINTABLE serialization has been recently implemented (not fully operational yet)
- Very simple example macros to read/write IRFs (~50 lines!)

`https://github.com/cta-observatory/flexIRF/tree/master/src`

IRF3 – Next steps

- Fully implement both serializations and test any limitation
- Define use cases where IRF I/O is involved
    - Does current standards suffice?
- Develop an open tool for IRF generation from event lists:
    - CTA pipelines
    - Would allow to test the format on real data
- Focus the effort of everyone involved into the same direction

Proposed IRF3 format - Conclusions

Thank you!

Proposed IRF3 format - Conclusions

Some IACT IRF dependencies

Name	expressed by	Valid Range	Bins	Comments
Energy (Rec.)	E'	10 GeV - 300 TeV	$\sim$ 200?	Current E_{rec} MC format
γ-ray direction (Rec.)	θ'	0° - 5°	10	0.5° per bin (depends on the obs. Mode)
	ϕ'	0° - 360°	6	60° per bin
		param	2-3	specific to each IRF
Energy (True)	E	10 GeV - 300 TeV	$\sim$ 200?	Current E_{true} MC format
γ-ray direction (True)	θ	0° - 5°	10	0.5° per bin (depends on the obs. Mode)
	ϕ	0° - 360°	6	60° per bin
		param	2-3	specific to each IRF
Pointing direction	Θ	0° - 90°	9	10° per bin
	Φ	0° - 360°	6	60° per bin
		Param.	2-3	specific to each IRF
Event Selection cuts	Bg. suppression	0-1	2	Different efficiencies regarding background suppression.
	Angular Cut	Point-like, no cut	2	narrow θ^2 cut influence certain Pdfs
Atmospheric conditions	Transmittance	0-1	3/4?	Interpolate between different sim. levels: M. Gaug
	Mol. prof.	Summer - Winter	2	2 enough? Interpolate between few sim. cases. M. Gaug
NSB		extragal, gal, n×NSB	4?	Interpolate between few sim. cases
Obs. Mode		Point, Divergent	2	Pre-defined observation modes
Array		Full, LST, MST+SST...	-	Normal modes + partial arrays
Hardware status		reflectivity, trig. thresholds	-	?

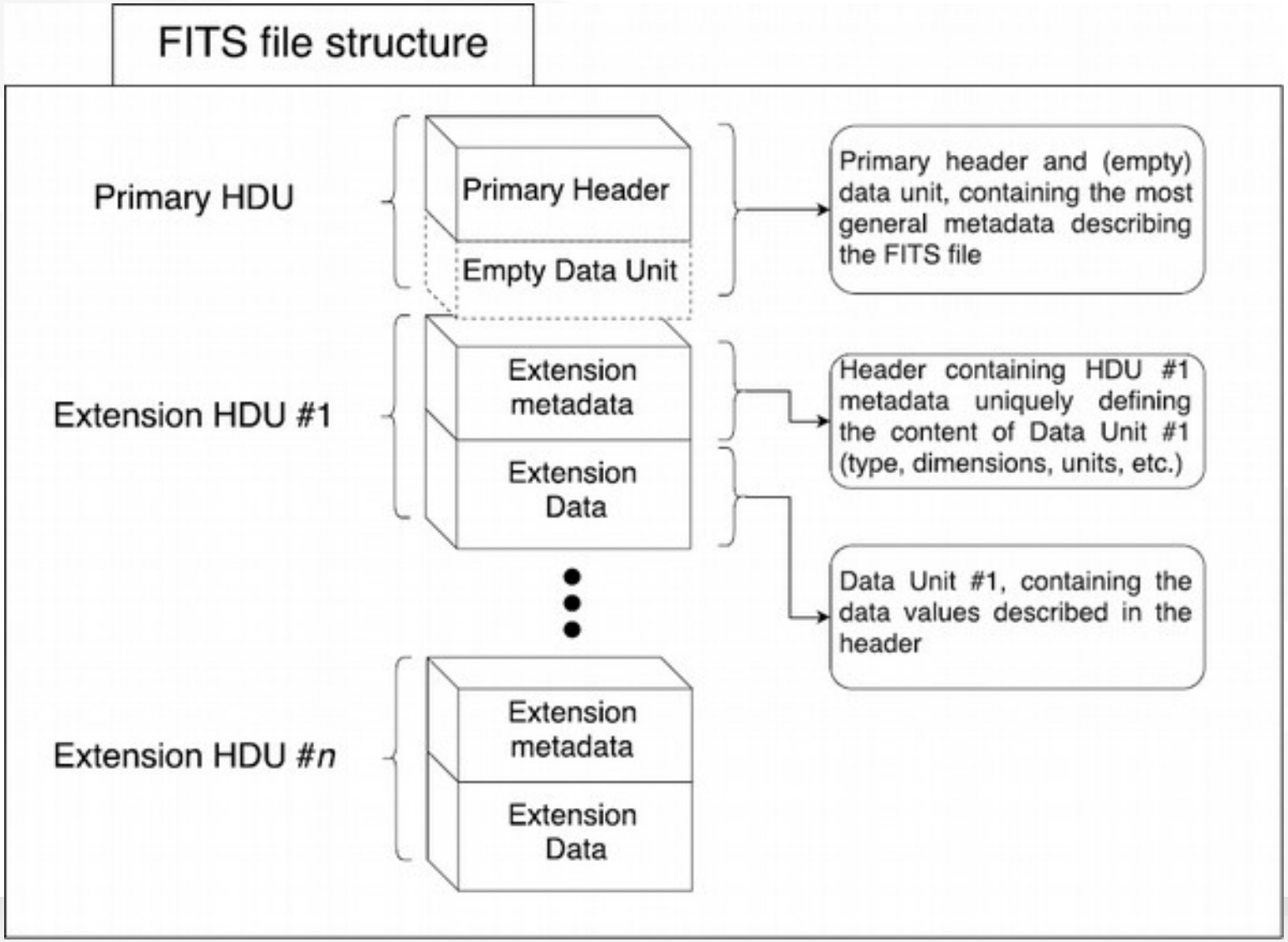
Mig. Matrix ~ 60 Gb per Obs. Mode/array/hw. Status...

IRF3 format requirements

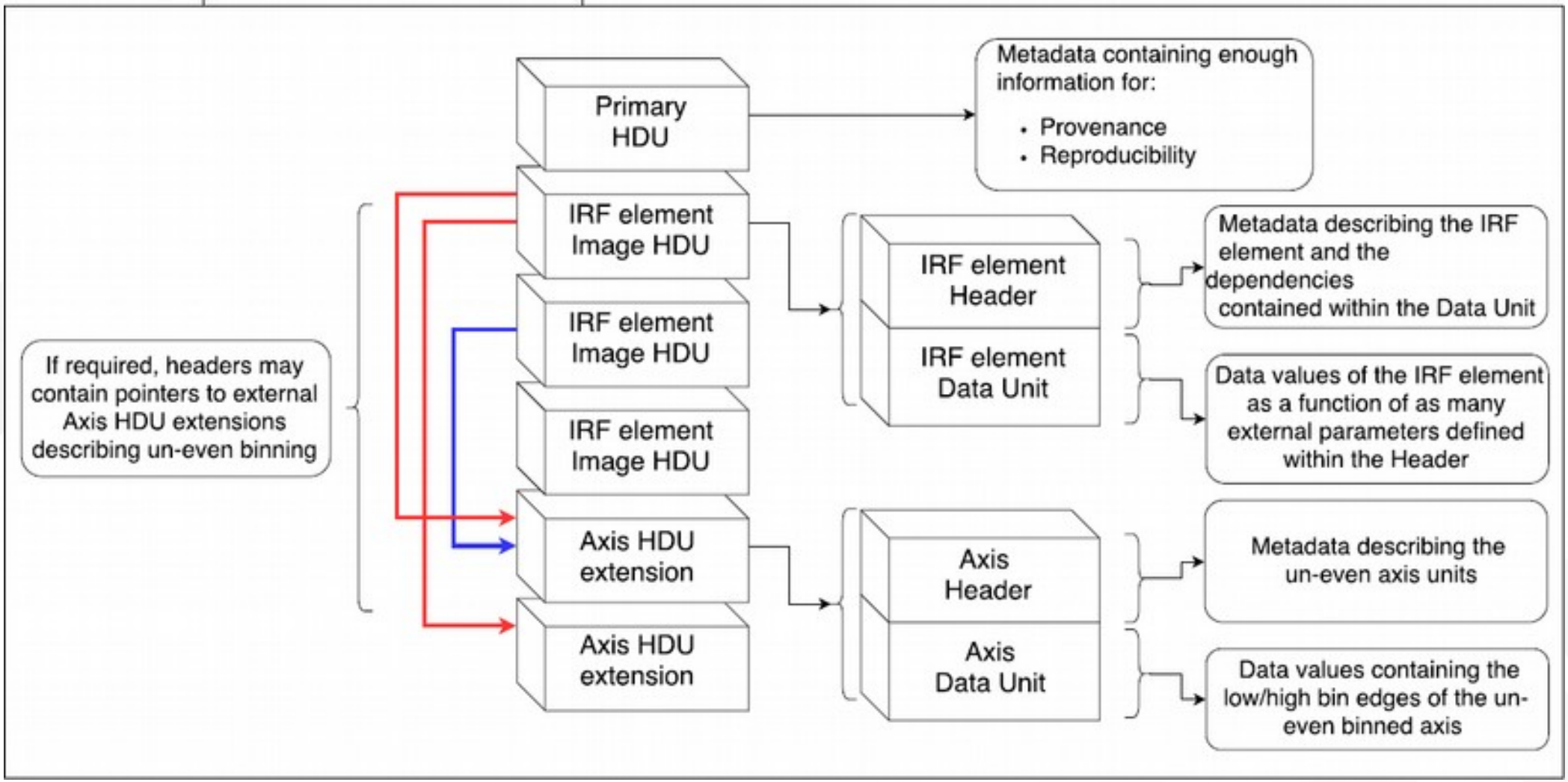
- A list of requirements were defined within the TDR:
  - contain enough information of how they were produced (provenance)
  - be provided in a single FITS file for all relevant observational conditions, (including the information of the valid range in which they are applicable)
  - allow estimation of the Flux (units of cm^{-2} s^{-1}) as a function of the energy, incident direction and time with uncertainties in accordance with CTA requirements

Proposed IRF3 format

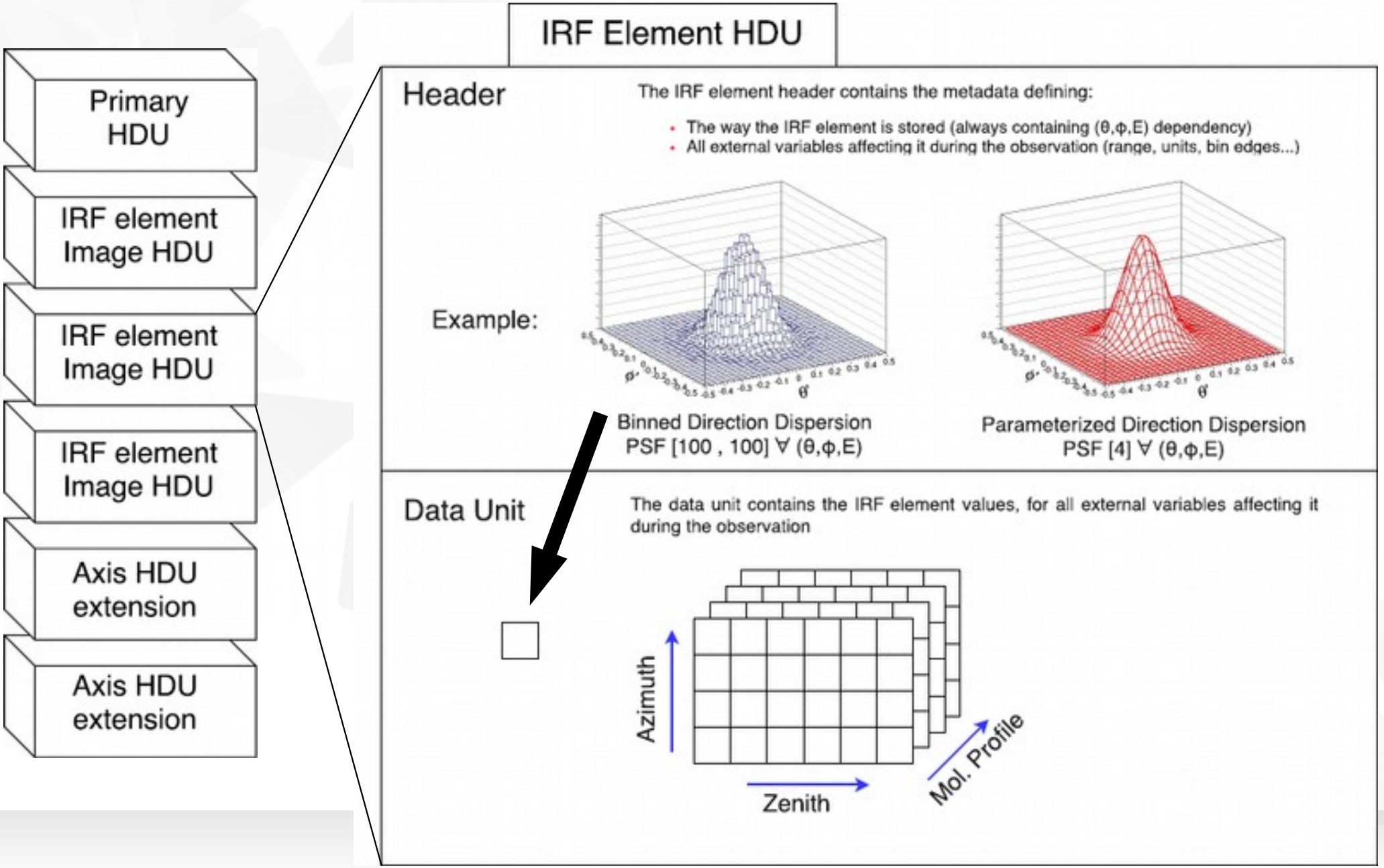
- FITS files are a collection of Header Data Units (HDUs)



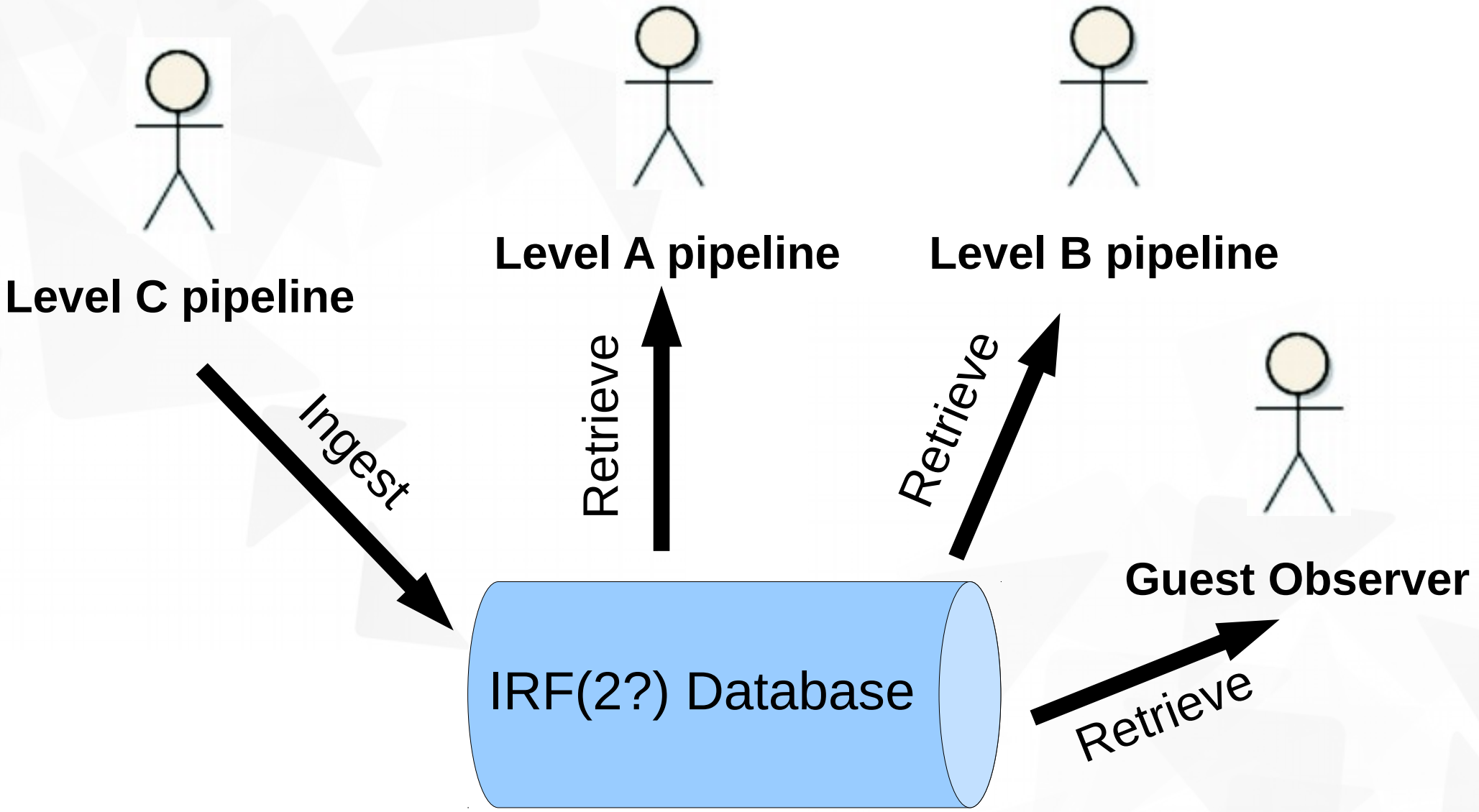
IRF3 file structure



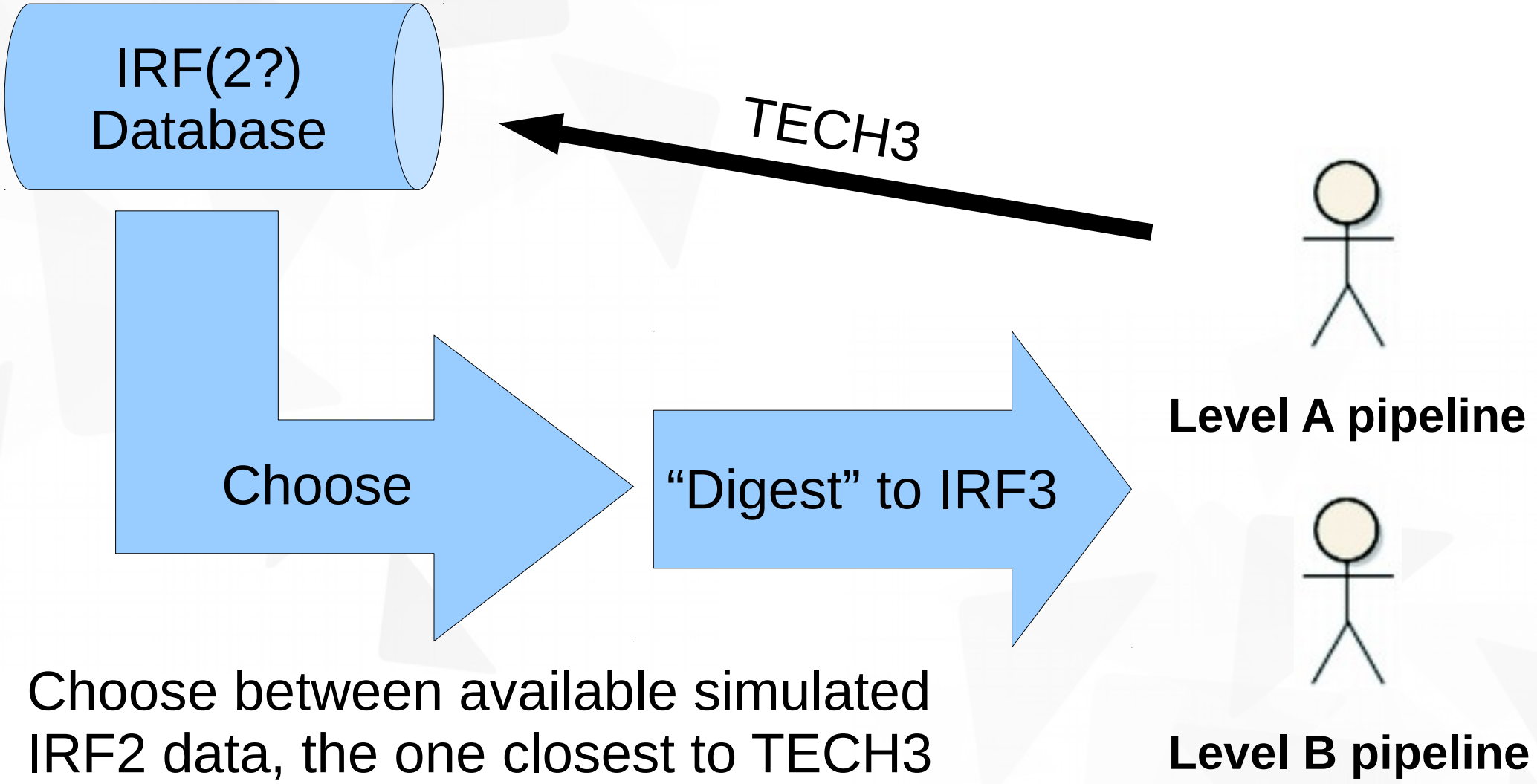
Proposed IRF3 format - Progress



IRF3 – Next steps



Proposed IRF3 format - Archive



Choose between available simulated IRF2 data, the one closest to TECH3