Path Transparency Observatory (PTO) Application Programming Interface (PAPI) Version 3: *Once More, With Feeling*

Brian Trammell (ETH)

MAMI Plenary Oslo, 4-5 July 2017

mami

measurement and architecture for a middleboxed internet

measurement architecture experimentation

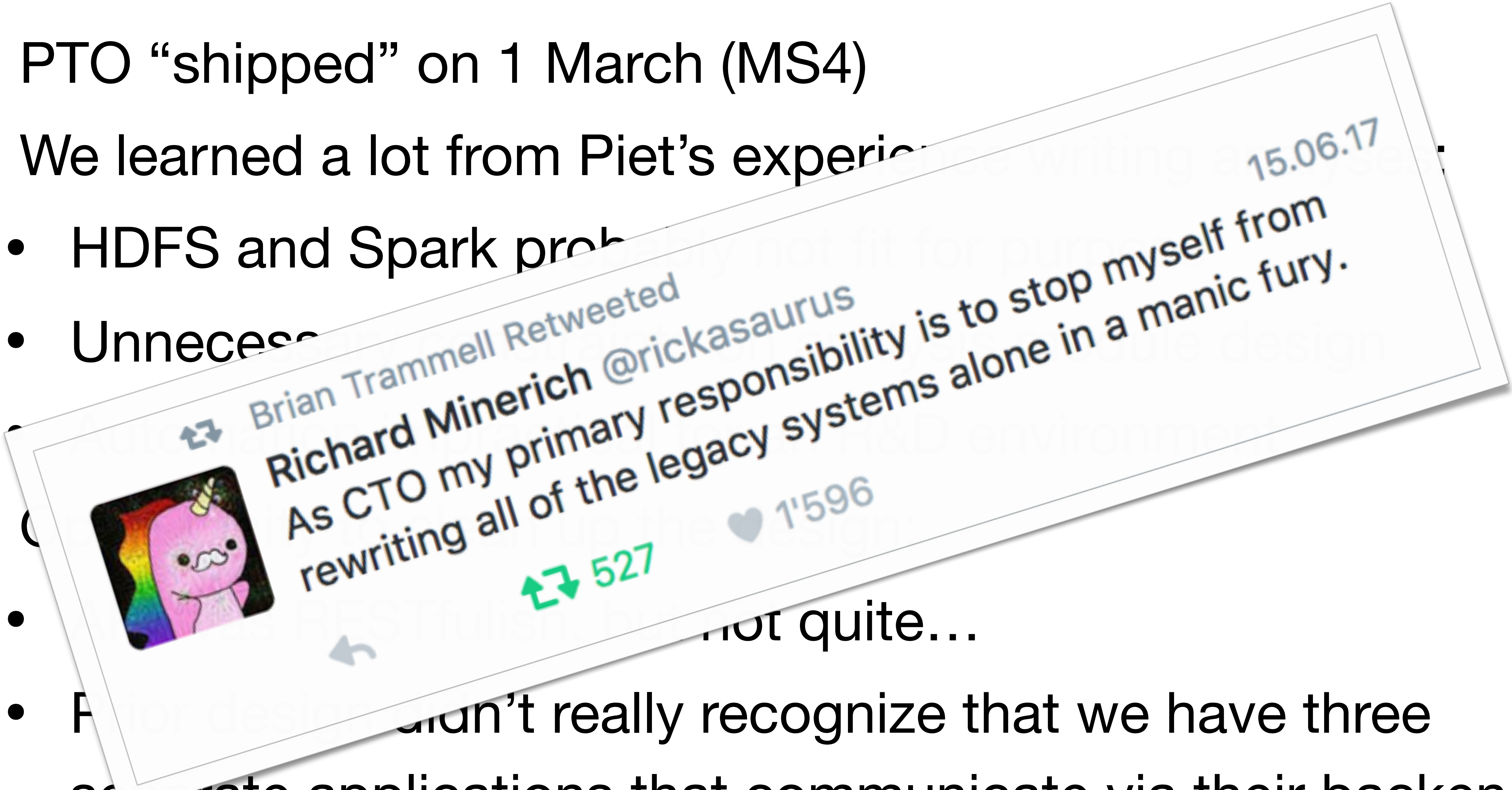
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421.The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

What and why?

- PTO “shipped” on 1 March (MS4)
- We learned a lot from Piet’s experience writing analyses:
  - HDFS and Spark probably not fit for purpose
  - Unnecessary constraints on analysis module design
  - Automation impractical for an R&D environment
- Opportunity to clean up the design:
  - API was RESTfulish, but not quite…
  - Prior design didn’t really recognize that we have three separate applications that communicate via their backends.

What and why?

- PTO “shipped” on 1 March (MS4)
- We learned a lot from Piet’s experience writing analyses:
  - HDFS and Spark probably not fit for purpose
  - Unnecessary [illegible] module design
  - Automation [illegible] R&D environment
- Opportunity to clean up the design:
  - API was RESTfulish, but not quite…
  - Prior design didn’t really recognize that we have three separate applications that communicate via their backends.



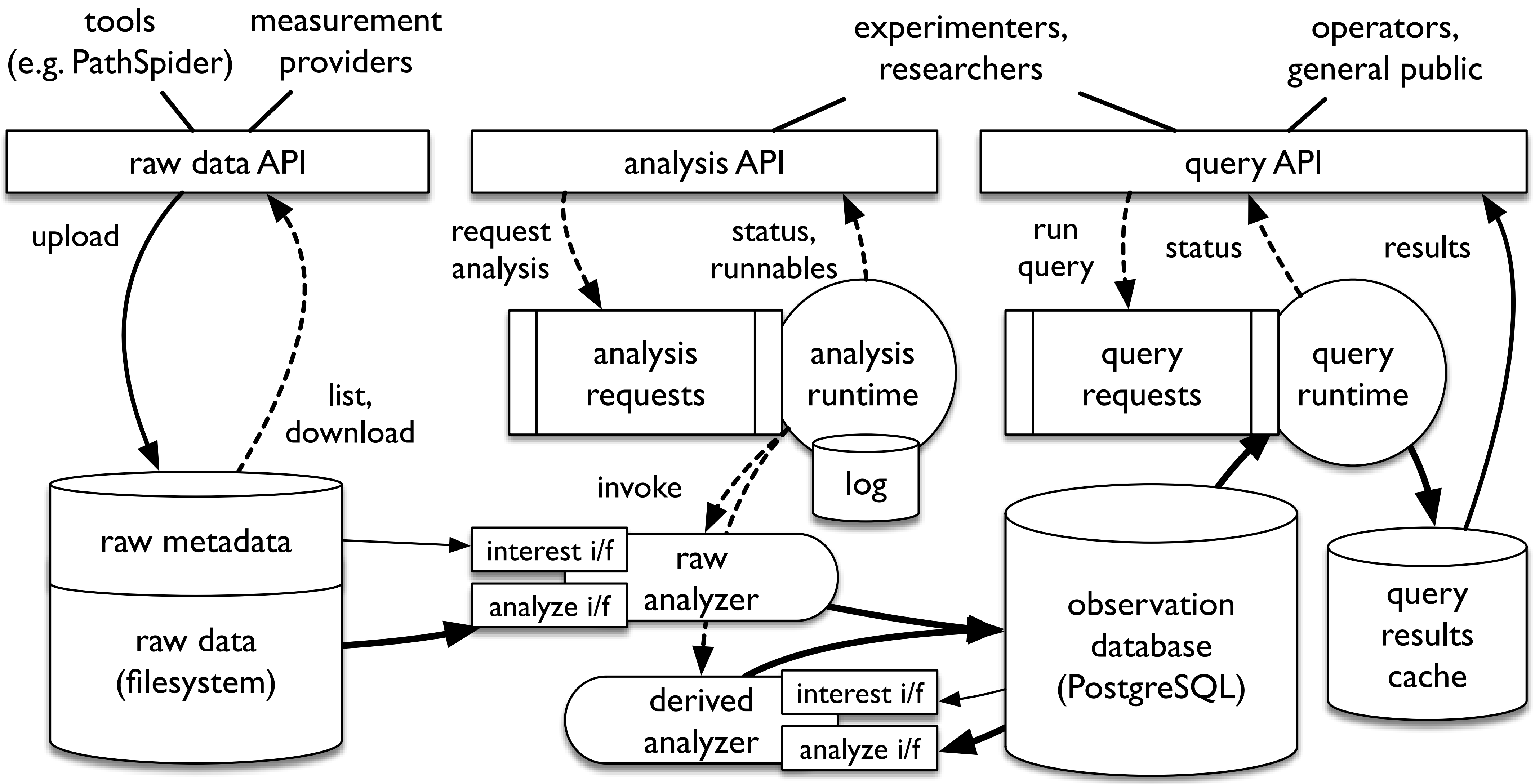
Brian Trammell Retweeted

Richard Minerich @rickasaurus 15.06.17

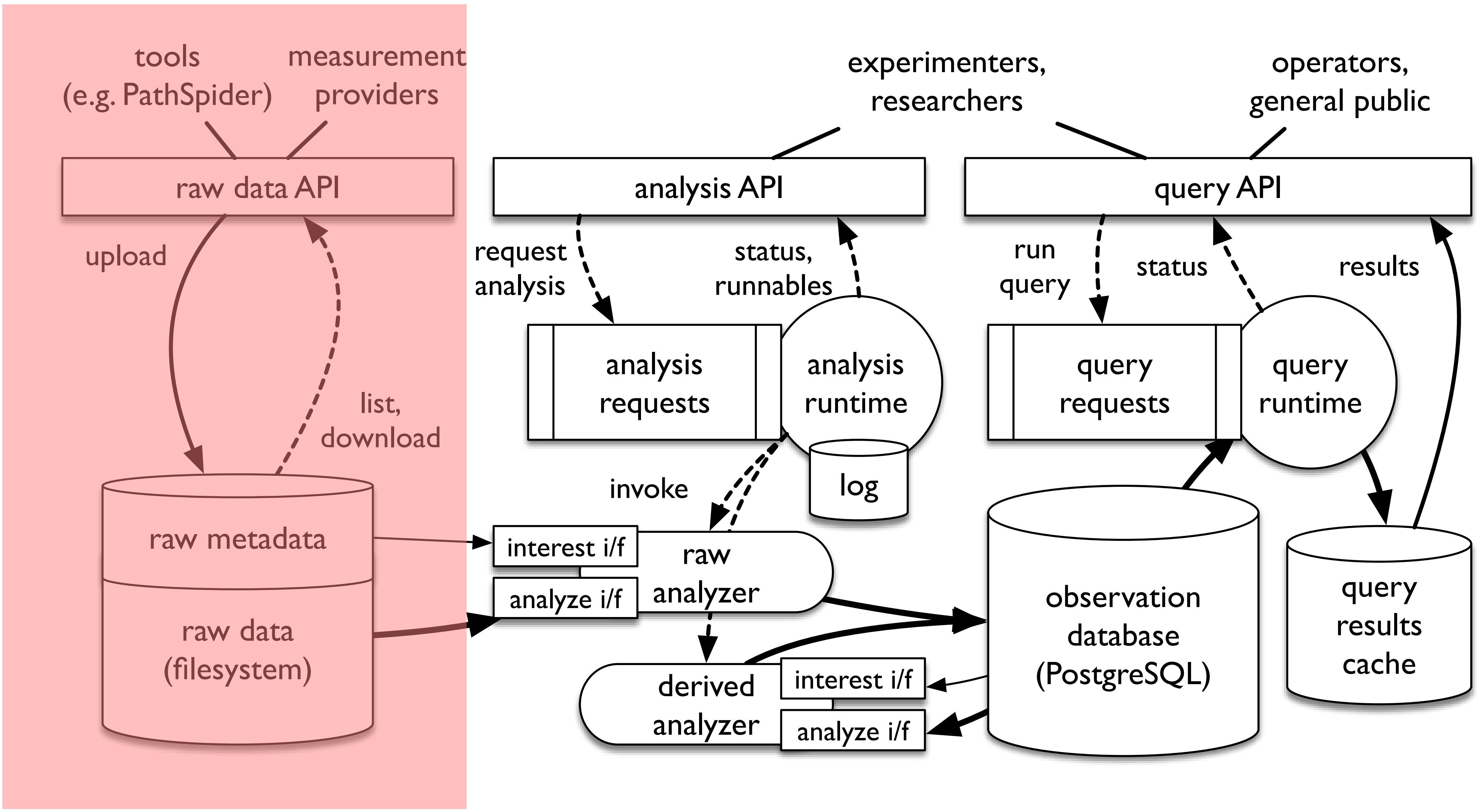
As CTO my primary responsibility is to stop myself from rewriting all of the legacy systems alone in a manic fury.

527 1'596

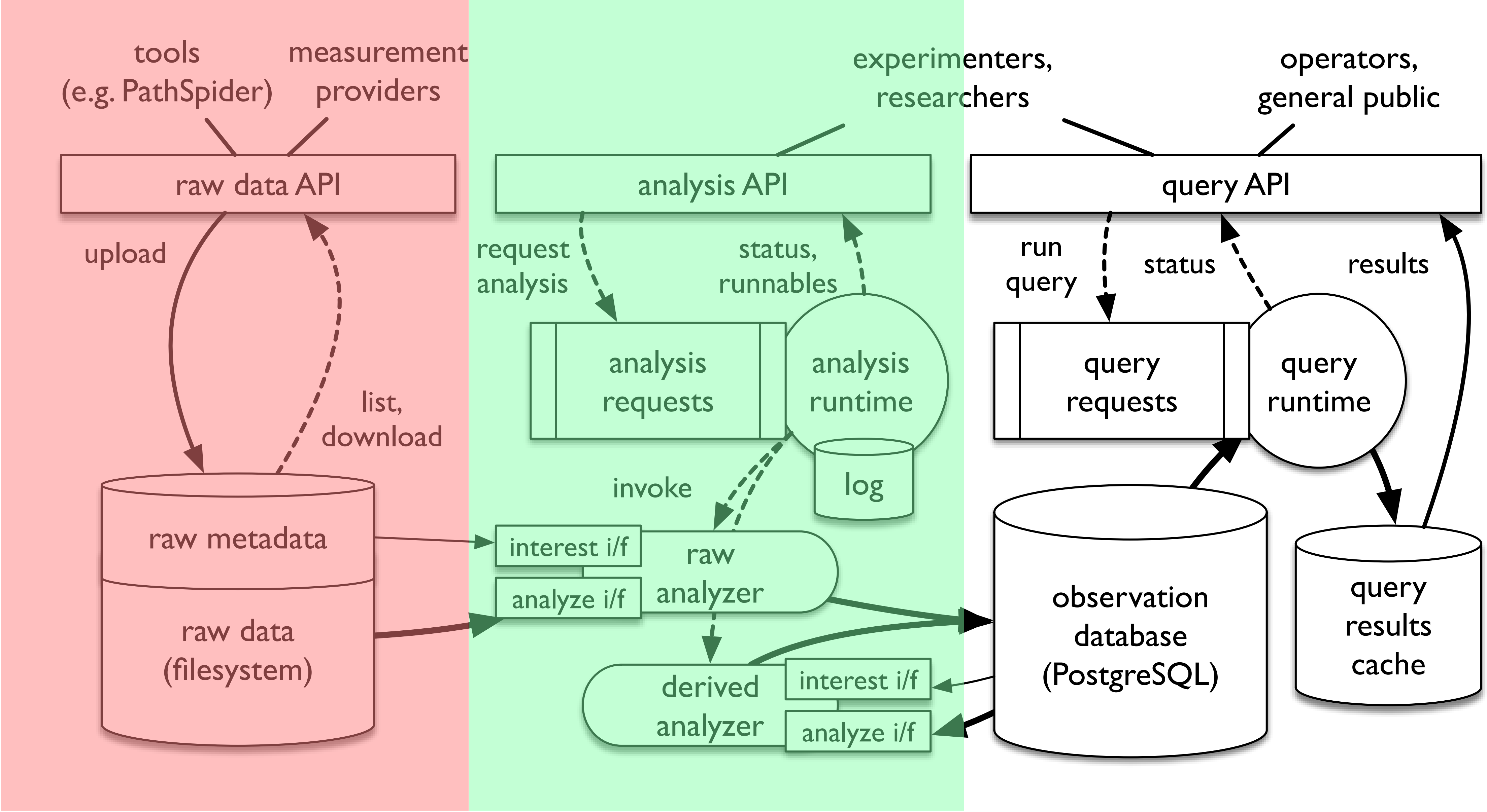
PTO v3 Architecture



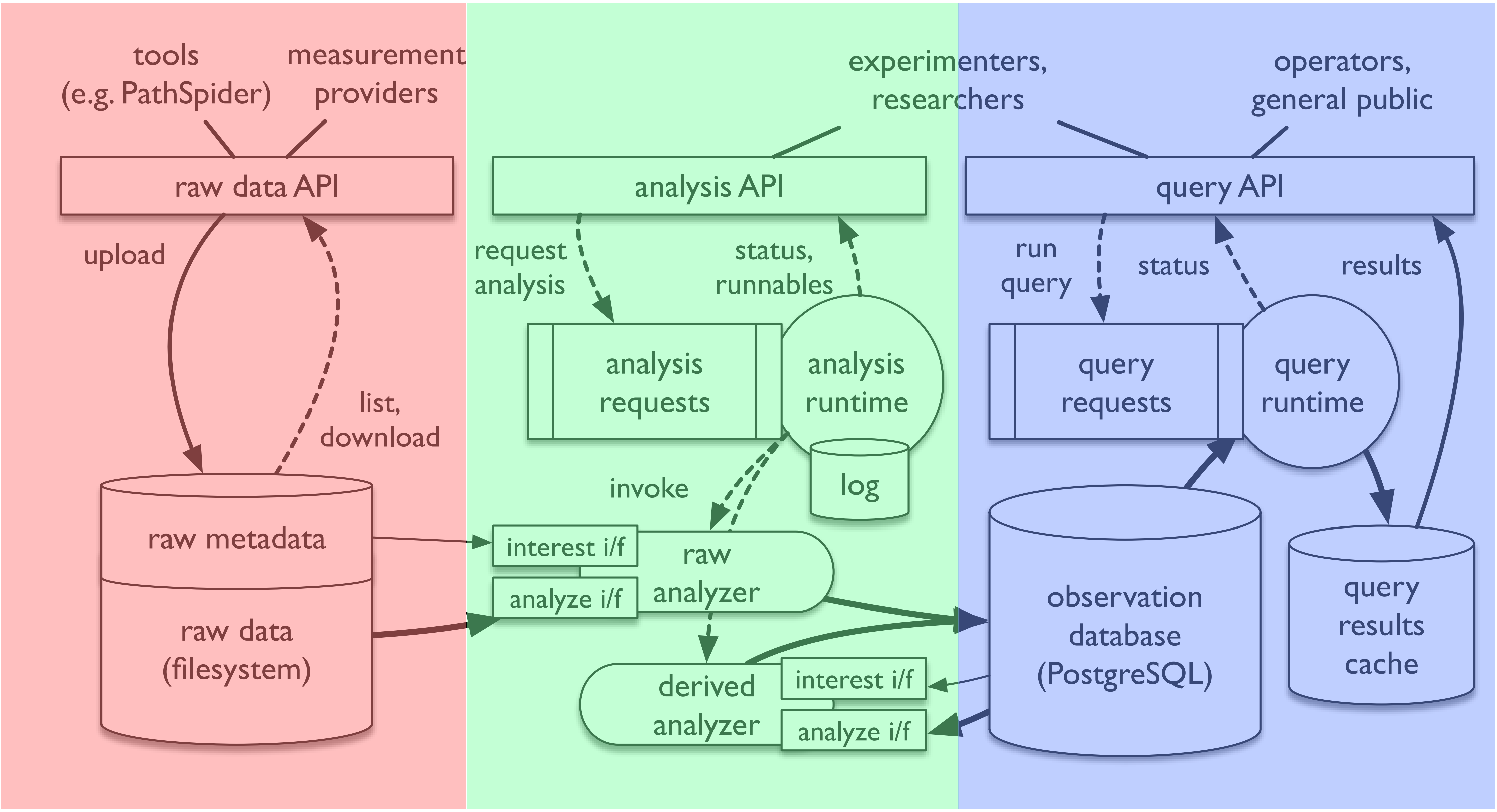
PTO v3 Architecture



PTO v3 Architecture



PTO v3 Architecture



What remains the same?

- Abstract information model for observations unchanged… but now includes the notion of an *observation set.*
  - Set of observations that have the same provenance.
  - Roughly but not necessarily the result of an analysis run.

- No changes planned to IQL

/papi/v3/raw/

- Each campaign is an resource, representing its metadata
  - e.g. `/papi/v3/raw/my_campaign`
- Each file is a resource representing its metadata
  - e.g. `/papi/v3/raw/my_campaign/my_file.ext`
  - Files inherit, override metadata from their campaigns
- File contents are a separate resource under the file
  - e.g. `/papi/v3/raw/my_campaign/my_file.ext/data`
- All metadata and content is immutable once uploaded
  - Deletion is manual.
- No relational database necessary: all information stored in ext4.’
- passes tests, live on muninn at ETH Zürich in August ’17.

/papi/v3/analyzer

- Analyzers stored in git repositories at a defined path
    - Repository structure includes metadata
- Available analyzers listed at `/papi/v3/analyzer`
- Analyzer metadata at `/papi/v3/analyzer/analyzer_name`

- Analysis control happens through the **log** and the **queue**

/papi/v3/analysis/log

- List of analyses that *have* run or that *could* run.
  - where an “analysis” is
    - an {analyzer, raw file} or
    - {analyzer, observation set} pair

- `/papi/v3/analysis/log?…`
  list analyses meeting a set of criteria
- `/papi/v3/analysis/log/analysis-id`
  retrieve metadata for a specific analysis

/papi/v3/analysis/queue

- List of analysis runs *pending* or *in progress*
  - where an “analysis run” is
    - an {analyzer, raw file} or
    - an {analyzer, [list of observation sets]} pair

- `/papi/v3/analysis/queue?…`
  list current analysis queue
- `/papi/v3/analysis/queue/analysis-run-id`
  retrieve metadata for a specific analysis run
- POST to `/papi/v3/analysis/queue` to submit an analysis run, analysis run identifier is returned.

Analysis Module Interface

- Analysis modules are simply UNIX executables
  - Raw data input as stdin filehandle
  - Observation data input/output as ndjson array
  - Interest registration and metadata via command-line

- Advantages:
  - Can write analysis modules in any language
  - Can bulk load resulting observation sets into DB

/papi/v3/query/

- List of queries *pending* or *in progress*
  - where a query is an IQL object requesting operation(s) over a set of rows in the observation database
- `/papi/v3/query?…`
  list current query queue
- `/papi/v3/analysis/query/query-id`
  retrieve metadata and result (if available)
- POST to `/papi/v3/analysis/queue` to submit a query, query identifier is returned.

Next steps

- Raw data interface up in August
  - Restart auto-upload from PathSpider over SaltStack
- Next: port and write new analysis modules
- Then: port query interface and front-end

- Goal: slightly-better-than-research-grade, easily-installable, “medium data” PTO as end ’18 outcome of the project.