Storage Events:
distributed users, federation and beyond

Paul Millar
(on behalf of the dCache team)

Nordic e-Infrastructure
Collaboration

CHEP 2018 at Sophia, Bulgaria; 2018-07-09
https://indico.cern.ch/event/587955/

How storage is used (currently)

Upload a file

OK

Download a file

OK

Delete a file

Permission denied

Stage file from tape

`Request queued`

Is the file on disk?

NO

Is the file on disk?

NO

Is the file on disk?

YES

The problems…

Upload a file

OK

Delete a file

OK

Rucio/LFC/…

Stage files from tape

`Request queued`

Are these files on disk?

`NO, NO, NO, …`

Are these files on disk?

`NO, NO, NO, …`

Are these files on disk?

`NO, YES, NO, …`
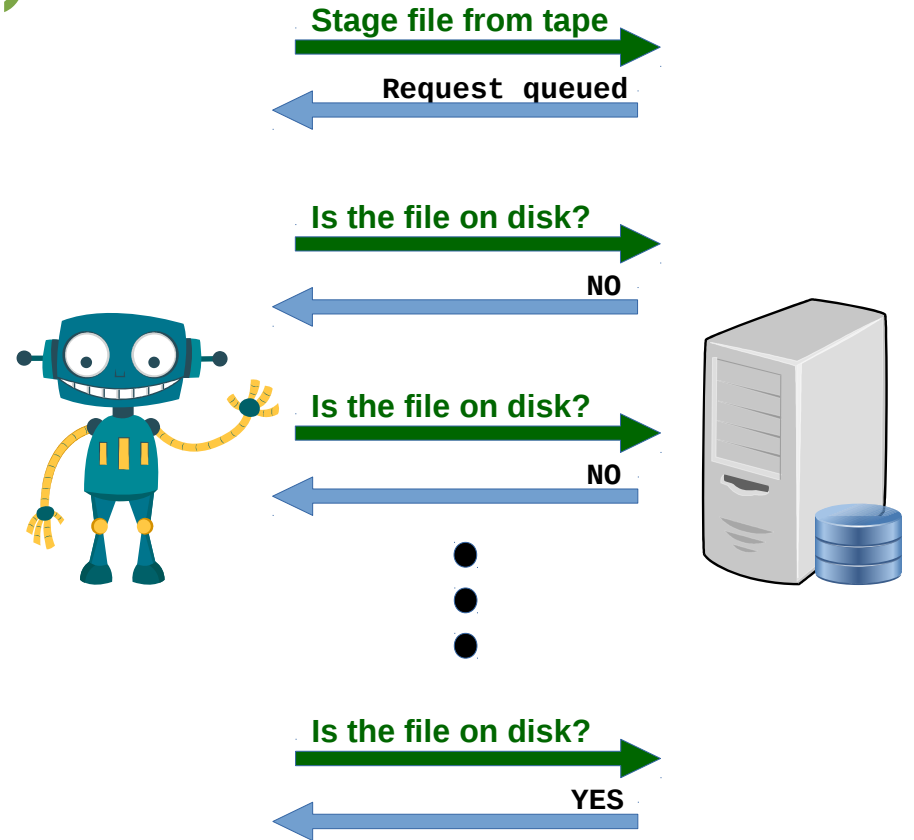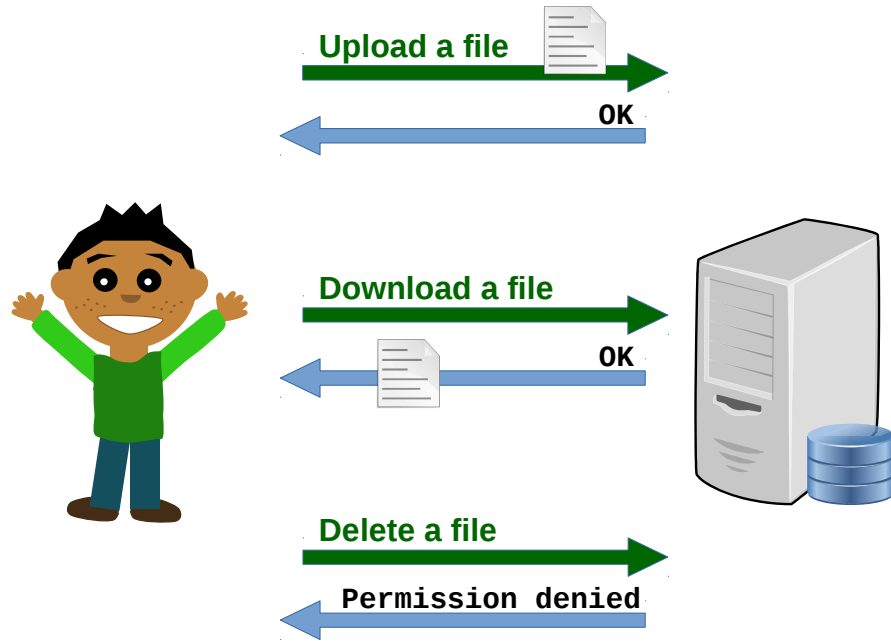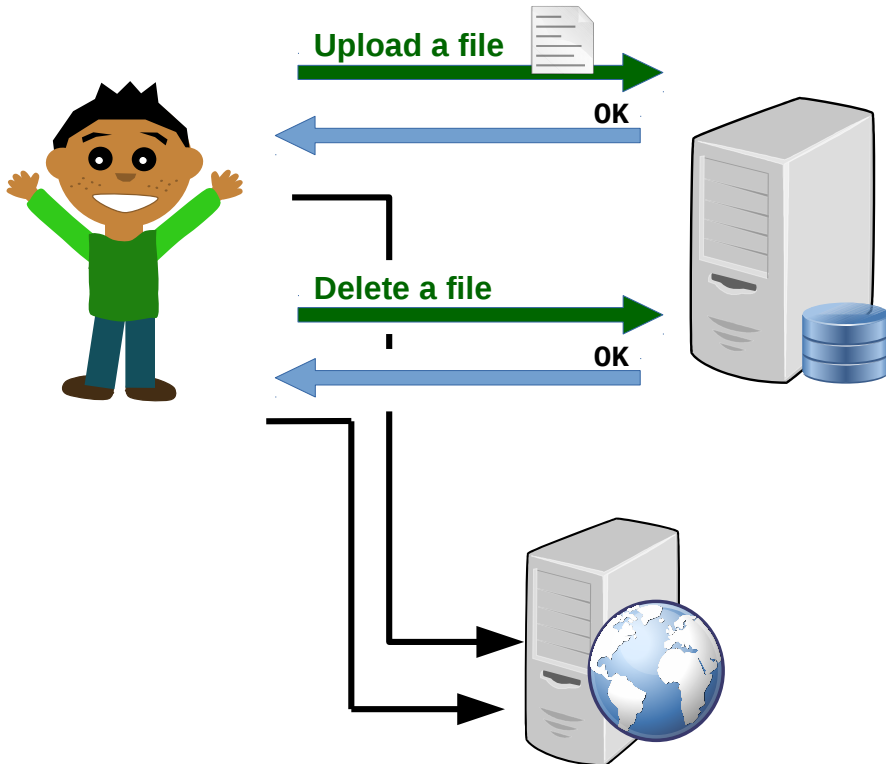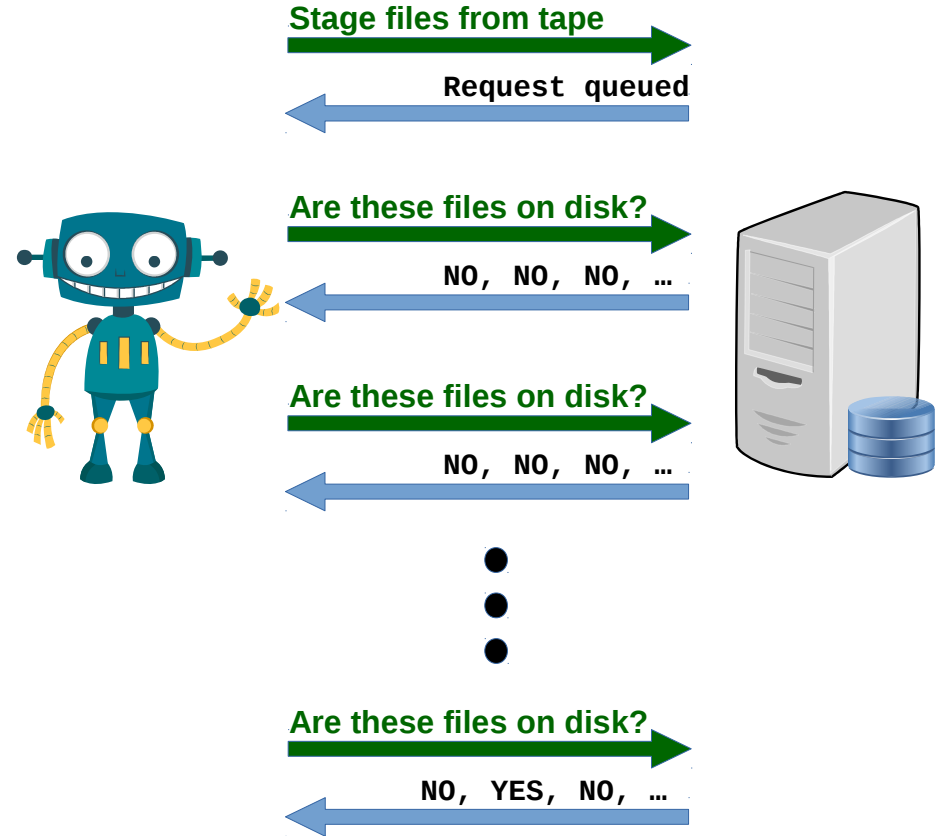
New way of interacting with storage: storage events

Subscribe to events

OK

Something happened #1

Something happened #2

Something happened #3

Something happened #5

Something happened #4
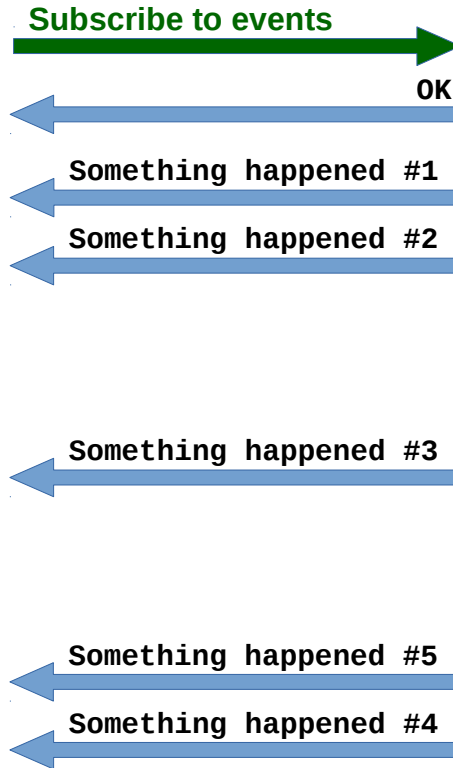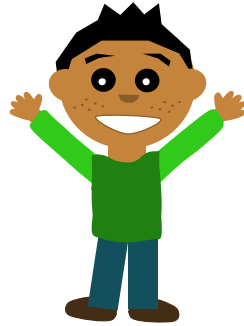
New solutions to old problems:

New solutions to old problems:

Stage files from tape

`Request queued`

Subscribe ...

`OK`

`File #1653 on disk`

`File #331 on disk`
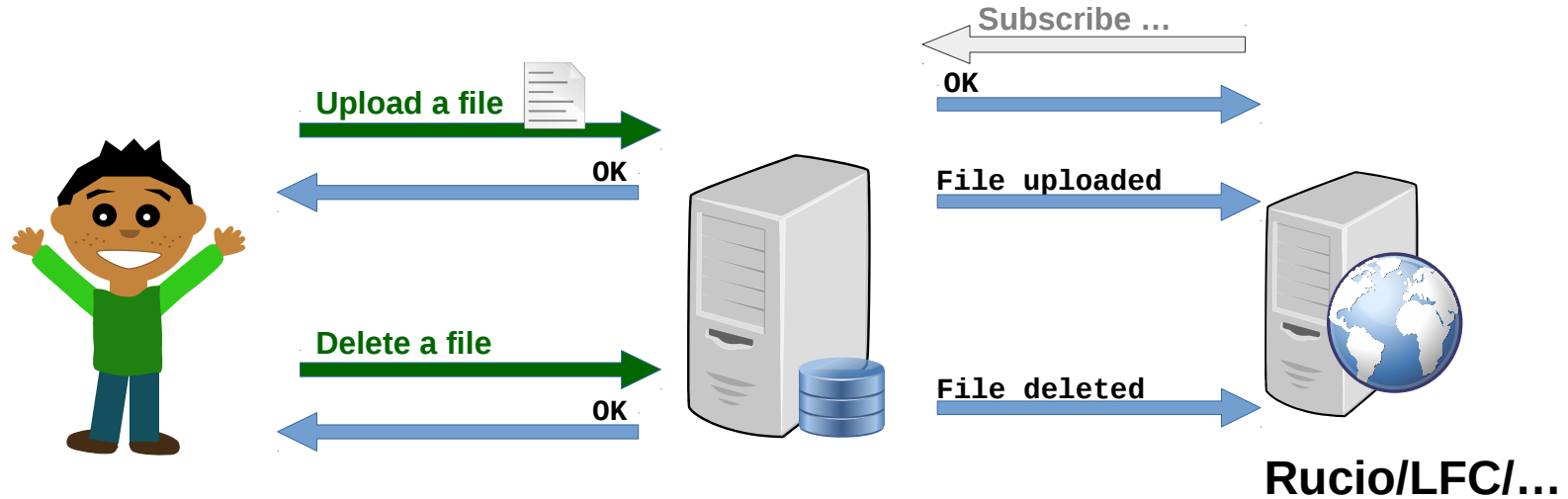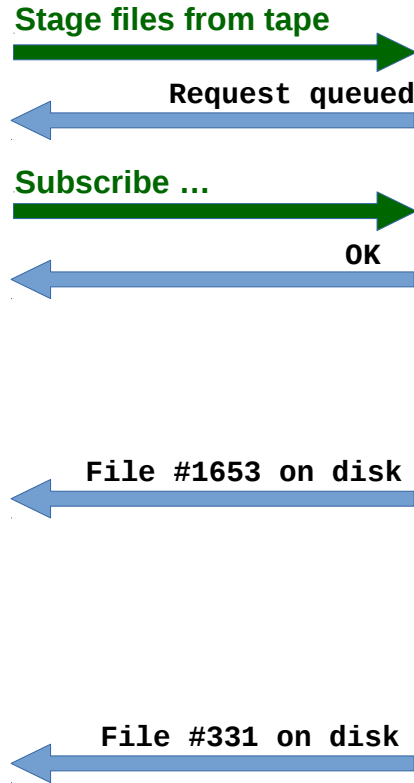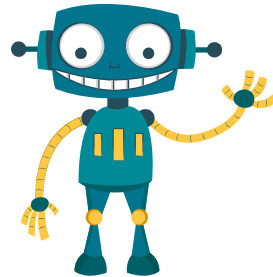
Comparison: it's what industry is doing...

Example: *Image Thumbnail Creation*

Photograph is taken — *Lambda is triggered*

S3

Photo is uploaded to S3 Bucket — *Lambda runs image resizing code to generate web, mobile, and tablet sizes*
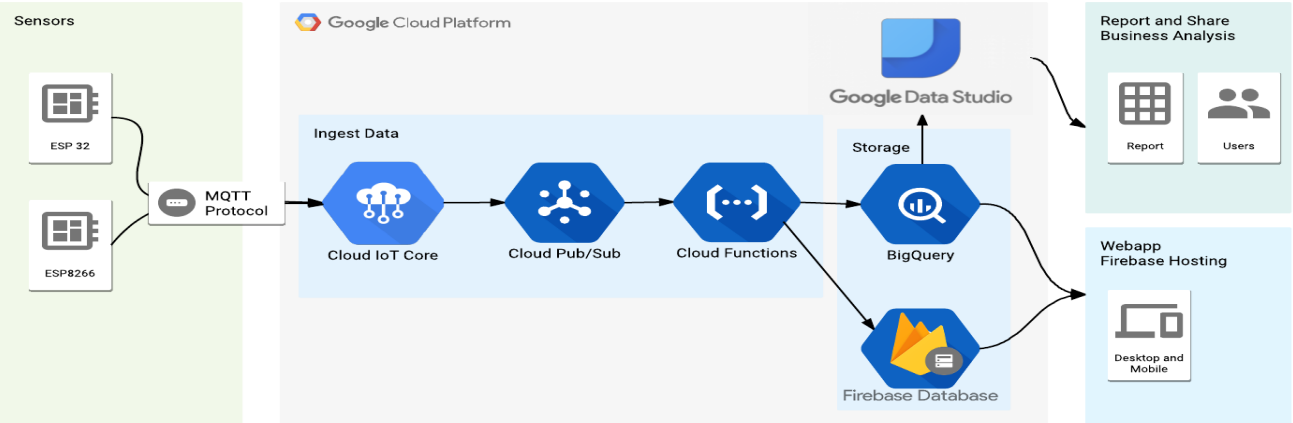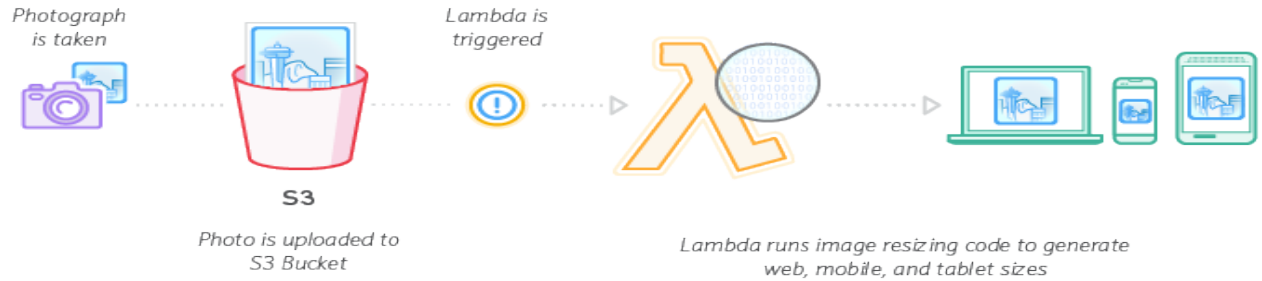
Comparison: it's what Open-Source is doing…

dCache Storage Events: Kafka
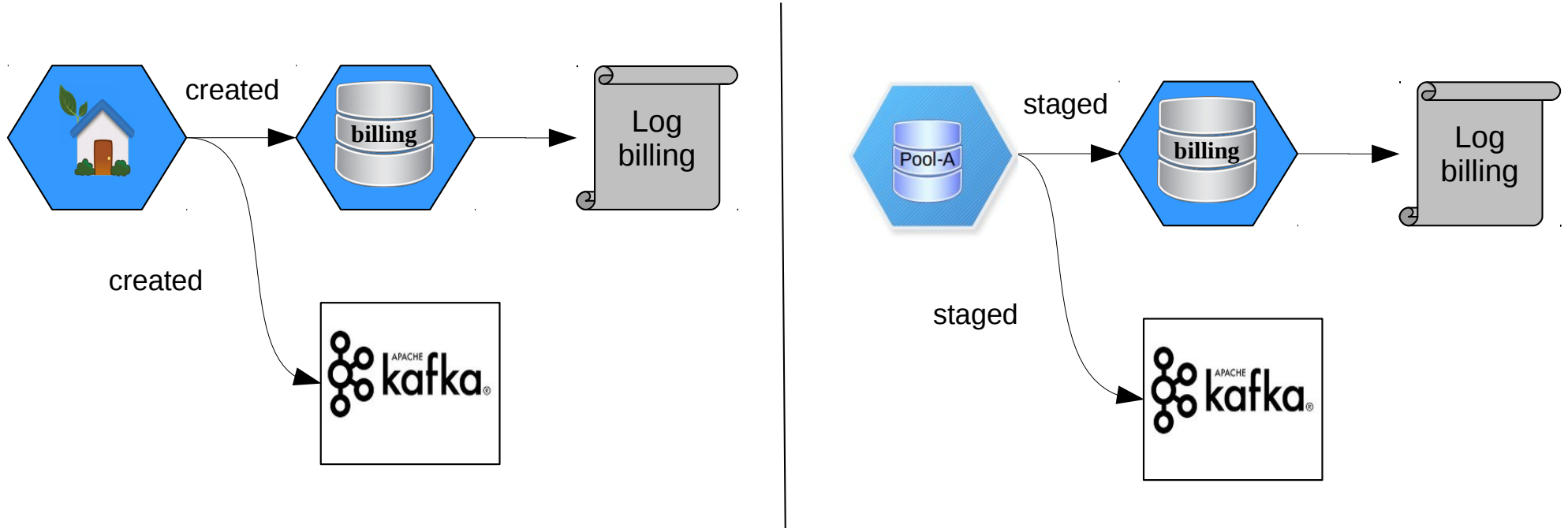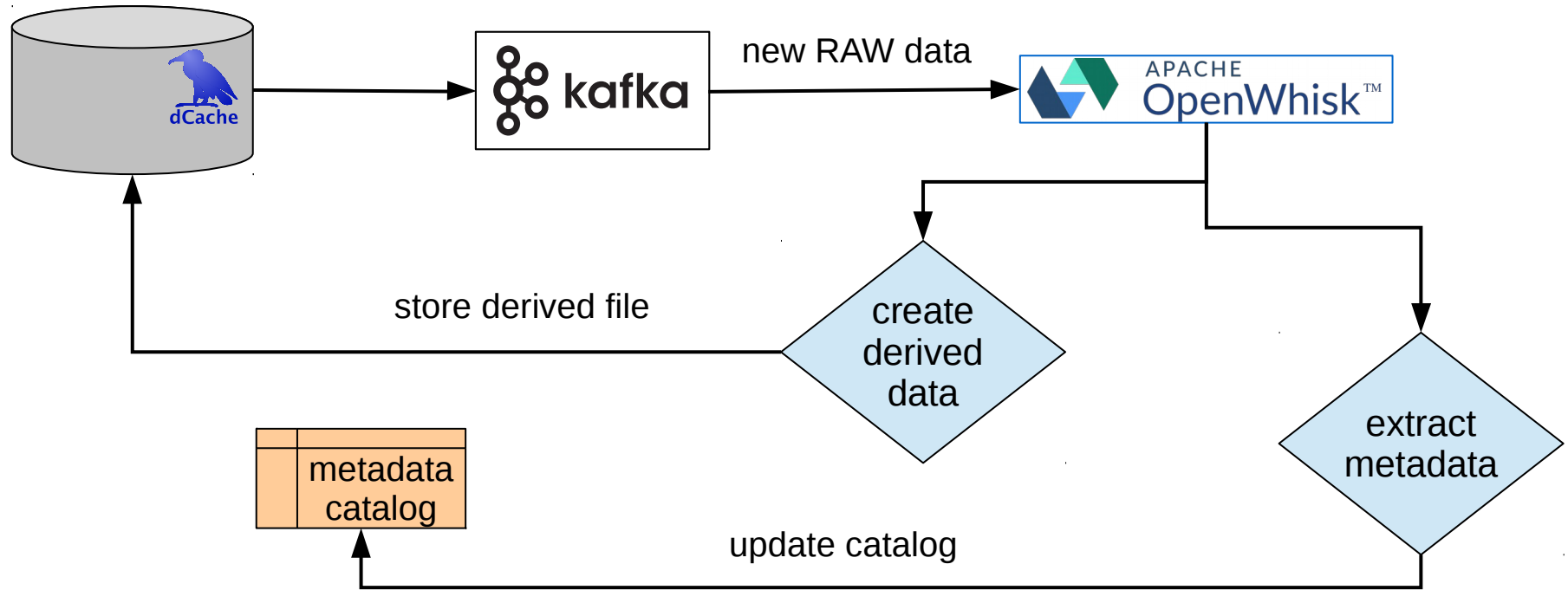
dCache Server-Sent Events (SSE)

- Based on HTTP v1.1

- HTML 5 **standard**

    Support for many languages and web-browsers

- Initially adding support for **inotify** events

    (it's how Linux does namespace notification)

- Plan to add:

    - Locality change notification: flush, stage, …

    - Transfer-related events

    - QoS changes

Cheat sheet: Kafka vs SSE

	Kafka	W3C® SSE
Standard …	Component	Protocol
What events does it see?	dCache internal events	Controlled
Main benefit	Easy integration	Built-in security
"Catch-up" storage	Memory & disk	Memory-only (currently)
Target audience	Site-level integration	Events for users

EOSC-Pilot demonstrator: EU-XFEL data ingest

dCache → kafka → APACHE OpenWhisk™

new RAW data

store derived file

create derived data

extract metadata

metadata catalog

update catalog

Rucio demonstrator: automated replication with SSE
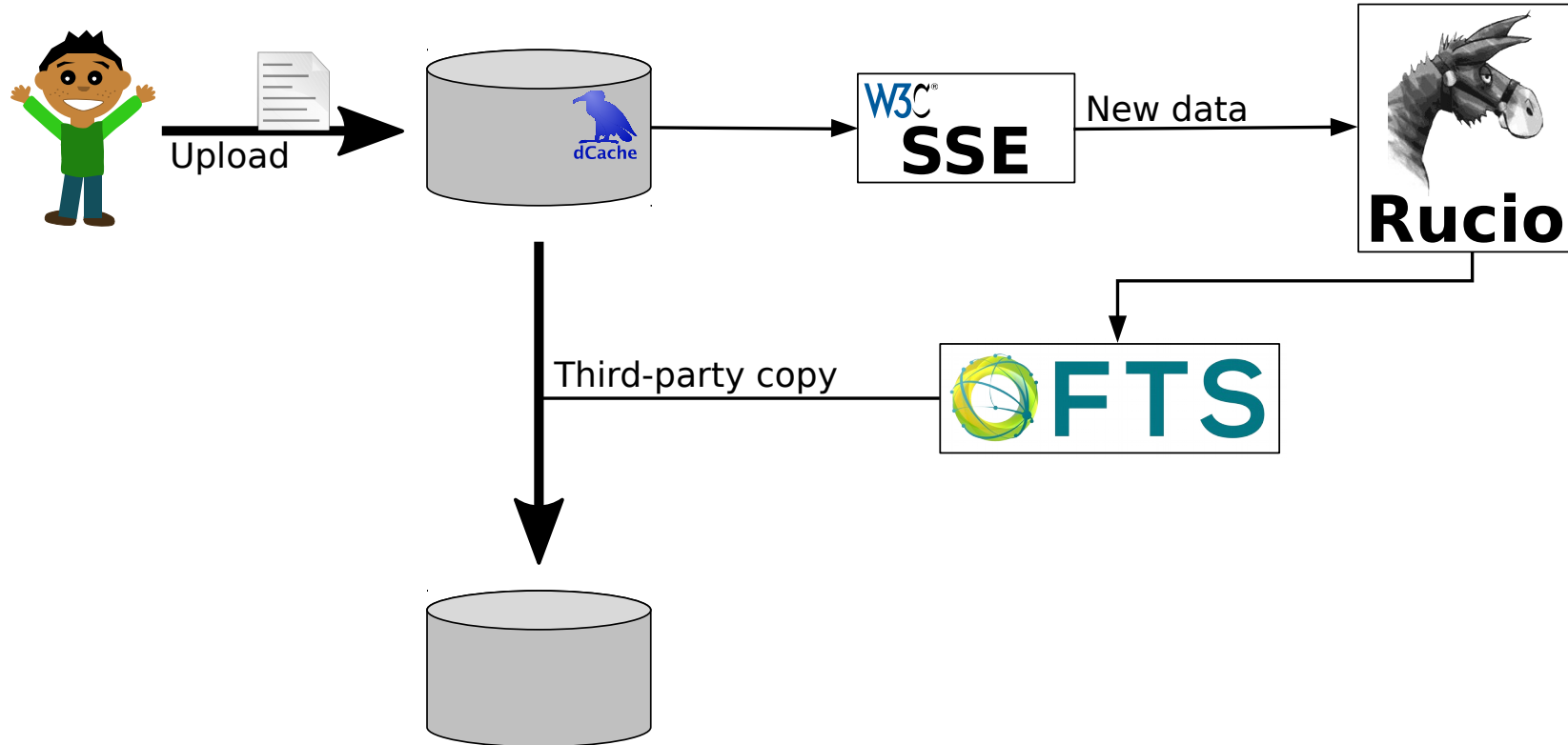
Future directions

- Complete SSE **inotify** support in dCache.

- Add **additional events**, based on initial feedback.

- Further explore **automated data workflow** (EU-XFEL usecase).

- Work with **Rucio team** to explore SSE integration.

- Work with dCache sites to deploy store events in **production**.

Conclusions

- We have presented **storage events**:

    Notification of changes within the storage service

- Several use-cases have been shown that illustrate potential **benefits to HEP**.

- dCache provides two complimentary event delivery mechanisms: **Kafka** and **SSE**.

- Work is on-going:

    - New event types to be added

    - Work with sites to bring this into production.

Thanks for listening!