

Events & dCache: an introduction

Paul Millar
on behalf of the dCache team

12th international dCache Workshop

Hamburg, Germany; 2018-05-28

<https://indico.desy.de/indico/event/19920/overview>



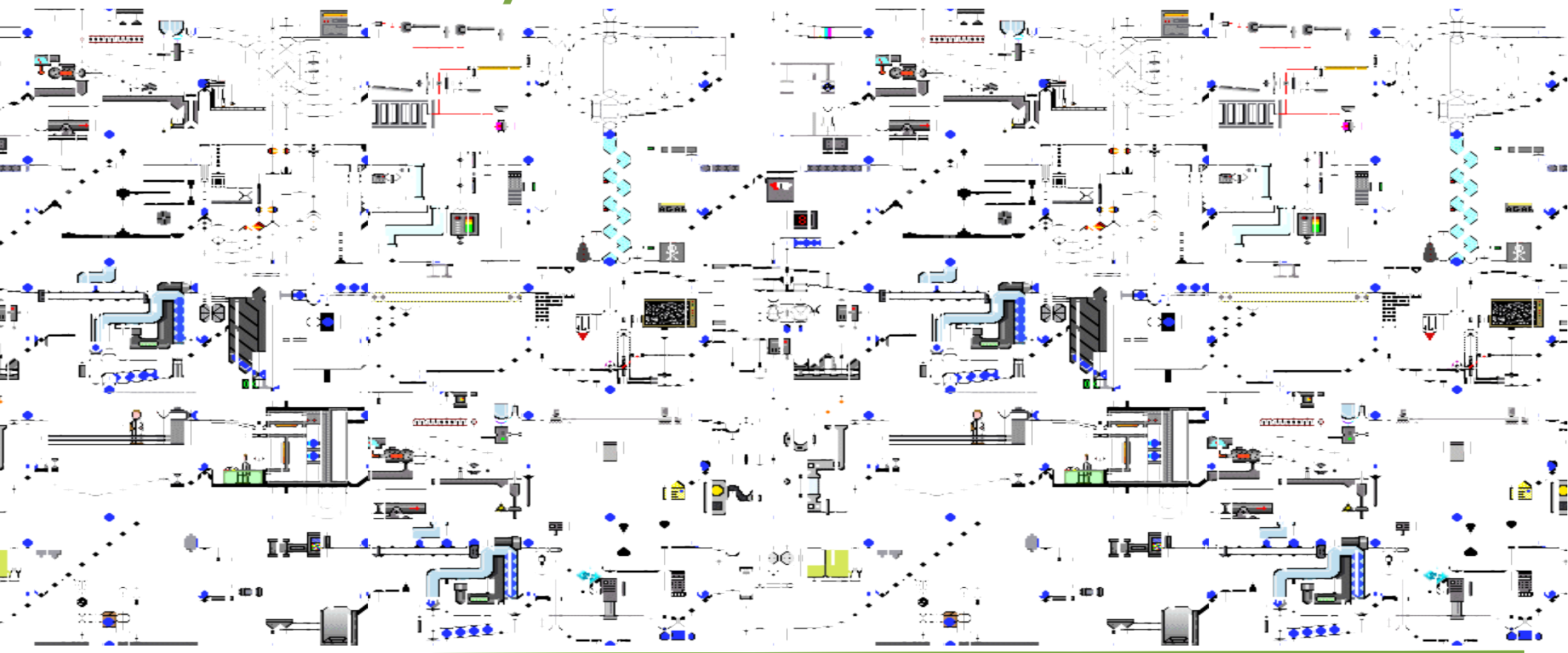
Nordic e-Infrastructure
Collaboration



eXtreme DataCloud



dCache is a busy machine ...



Admins might want to know ...

... all transfers

- Update monitoring
- Generate statistics
- Look for unusual behaviour

... any alerts

- Corrupted data,
- Crashed pool,
- Database connection problems

... staging activity

- Monitoring
 - Looking for problems
-

Users might want to know...

... a file was just upload

(I need to extract its metadata)

... a file was just staged from tape

(good! I can start the analysis job)

... a file was just deleted

(OK, I must remove it from the catalogue)

...

Polling: watching for side-effects

- Internally, dCache generates events, but these events do not propagate outside dCache.
 - The only way of detecting these changes is to look for ***the effect*** of that change:
 - Uploaded file → directory's mtime or listing changes,
 - A staged file → file locality changes to ONLINE.
 - So, you can query the current status:
 - ... and again 5 seconds later,
 - ... and again 5 seconds later,
 - ... and again 5 seconds later,
 - ...
-

Polling is bad



Polling is bad

- Each time you query the current status, dCache must fetch that information and report back.
 - Many queries with the same response
 - Each time you do this, dCache generates a report that is (almost always) the same:
 - Wasting dCache resources, network bandwidth, ...
 - Doesn't scale: need to query each "thing" you're interested in:
 - If you have 500 pools, that's 500 queries.
-

An alternative – add a back-channel

- If a catalogue needs to be kept up-to-date, use a special client that also makes changes in the catalogue.

This is what currently happens currently in WLCG

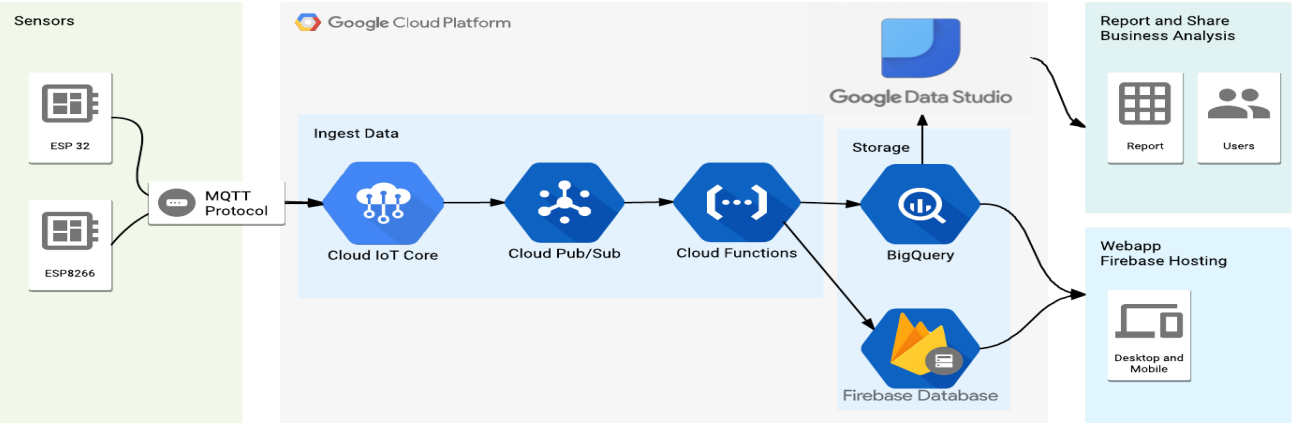
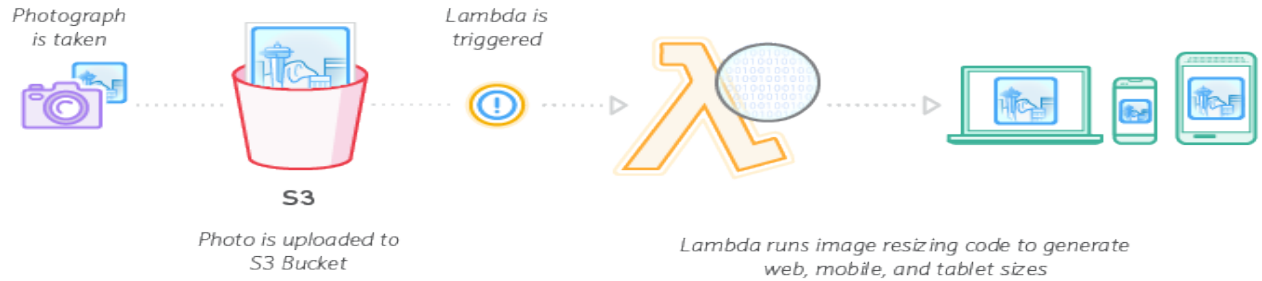
- This also has limitations:
 - Requires specialist client (can't “just copy” data)
 - Failures can result in dangling links or dark data.
 - Client needs to know which service to contact
-

A better alternative... events

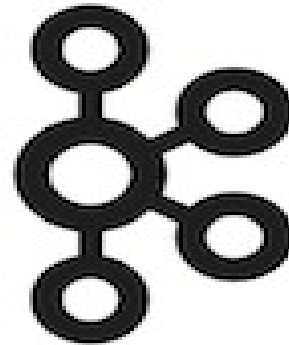
- What if... dCache simply told you when something you're interested in happened?
 - when new data is available
 - when files are staged from tape
 - This is much more efficient (i.e., faster)
 - It has other benefits ... integration with standard tools
-

Events is what industry is doing...

Example: Image Thumbnail Creation



Events is what Open-Source is doing



Two approaches: KAFKA and SSE

- **KAFKA** – industry standard **component**
 - Eventually see all dCache internal events,
 - Easy integration,
 - Currently all clients can see all internal events,
 - Configurable event “catch-up” storage,
 - Suitable for site-level (maybe VO-level) integration.
 - **SSE** – industry standard **protocol**
 - Clients in all major languages and major web-browsers,
 - Built-in dCache-user focused security,
 - Limited event “catch-up” storage,
 - Suitable for user-driven integration.
-

Complementary, not competing

- Kafka and SSE have different target audiences
(more details in a bit...)
 - In future, SSE/frontend could consume Kafka events
(SSE becomes a translation service)
-

Now on to the details...

