TOWARDS A RESPONSIVE CVMFS ARCHITECTURE

Radu Popescu (radu.popescu@cern.ch), Jakob Blomer, Gerardo Ganis
CERN, EP SFT

The classic CVMFS architecture involves a single publisher per repository. Clients request updated versions of metadata catalogs based on time-to-live values. Clients initiate all the communication. This architecture is efficient and robust, and well suited for the most common use cases.
New features and subsystems are being added to CVMFS to allow the implementation of new workflows, whether through scaling to larger publication workloads, ensuring lower changeset propagation delays, or constructing complex pipelines around CVMFS repositories.

MULTIPLE RELEASE MANAGERS

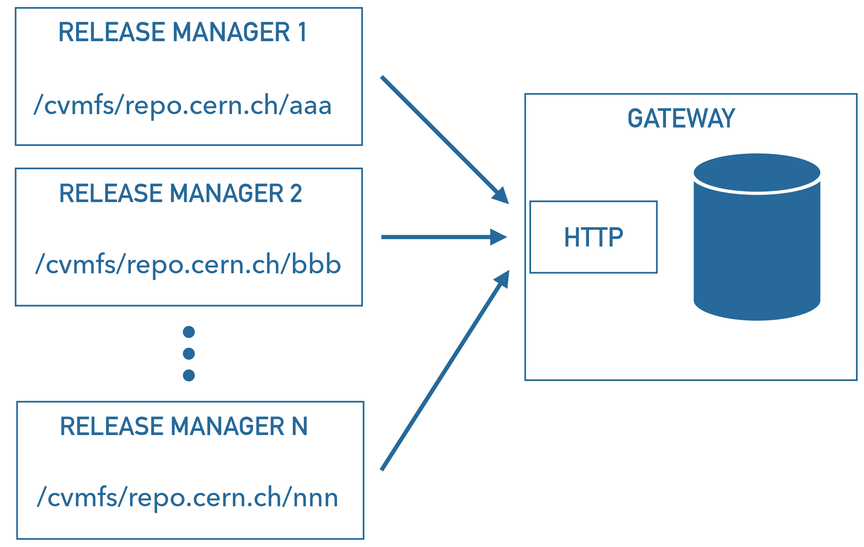
It is now possible to operate multiple publishers (release managers) per repository.

Each publisher can write files concurrently with the others to separate subpaths in the repository.

This allows dealing with an increased publication volume through horizontal scaling.

The release manager machines can be restricted to only be able to publish to specific subpaths.

Multi-tenant repositories can be implemented (i.e. user project areas, container image hubs).



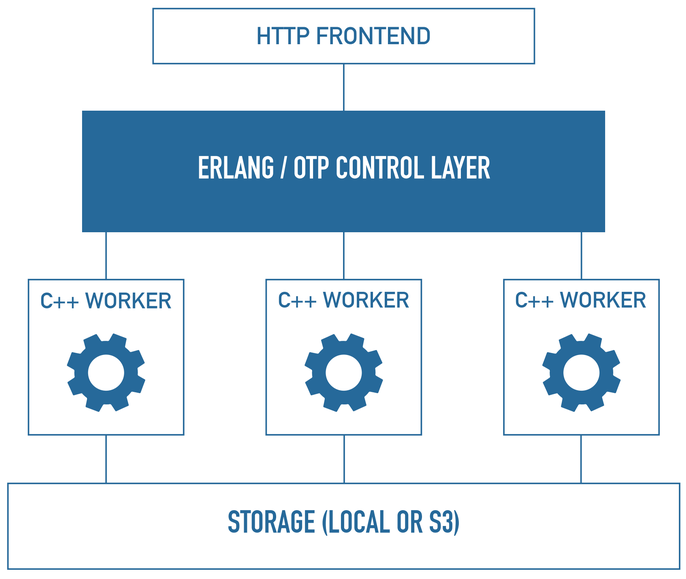
REPOSITORY GATEWAY

A new component which receives the payloads from the multiple publishers and does the actual writing into the repository.

It enforces access control policies and hands out exclusive leases to different repository subpaths.

Ensures that the repository stays consistent.

Is implemented with Erlang/OTP, for scalability and robustness.

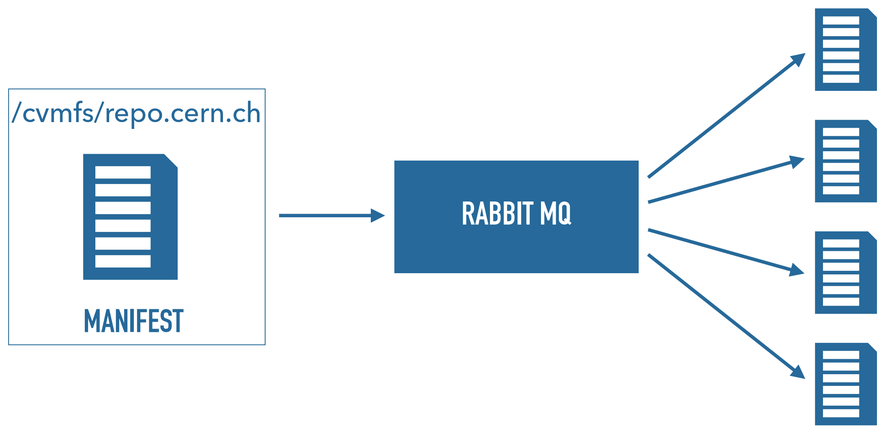


THE NOTIFICATION SYSTEM

A high performance message broker forwards messages from publishers to interested clients.

Clients receive the new manifest from the notification system directly, when it becomes available, no waiting for TTL.

Drastically reduced propagation delay, messages are dispatched to all the subscribers in a matter of seconds.



Uses RabbitMQ, a performant, battle-tested message broker.

Multiple protocols: AMQP, MQTT, MQTT over Websocket.

A single large commodity server could handle 100000 subscribers.

Clustering support, for scalability and high-availability setups.

A PROXY LAYER BASED ON XCACHE

A configuration for XRootD, offering a high-performance proxy layer.

Exposes an HTTP server interface, CVMFS clients to connect directly to it.

Ingests files from an HTTP source (experimental).

Xcache works non-intrusively between a CVMFS repository and clients.

Integration of Xcache with the CVMFS notification system (under development).

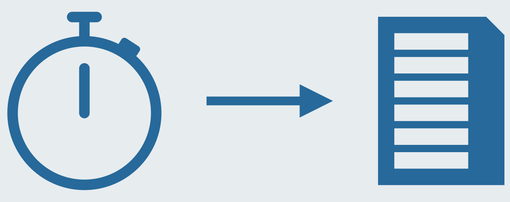
Collaboration with Andrew Hanushevsky and Wei Yang, SLAC.

THE APPLICATIONS OF THE NOTIFICATION SYSTEM

LOW DELAY PROPAGATION FOR CLIENTS

Clients can learn about changes, within seconds, without inefficient polling.

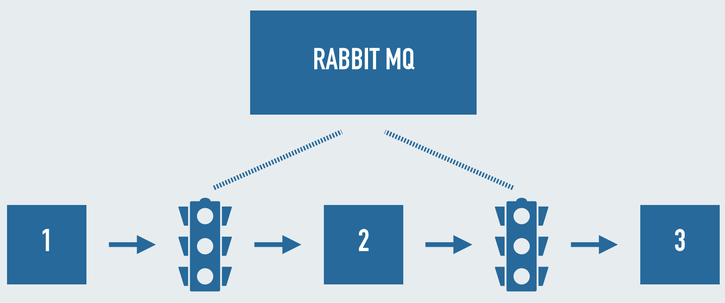
Easier to work with repositories storing rapidly changing data, such as experiment conditions data.



COMPLEX PROCESSING PIPELINES

The notification system allows knowing exactly when a repository has changed.

Makes it easy to implement processing pipelines which publish to repositories and wait for replication, at various points.



SPECIALIZED TOOLS

The subscribers to the notification system can be anything:

- container image builders
- monitoring tools
- dashboards

CHEP, July 2018