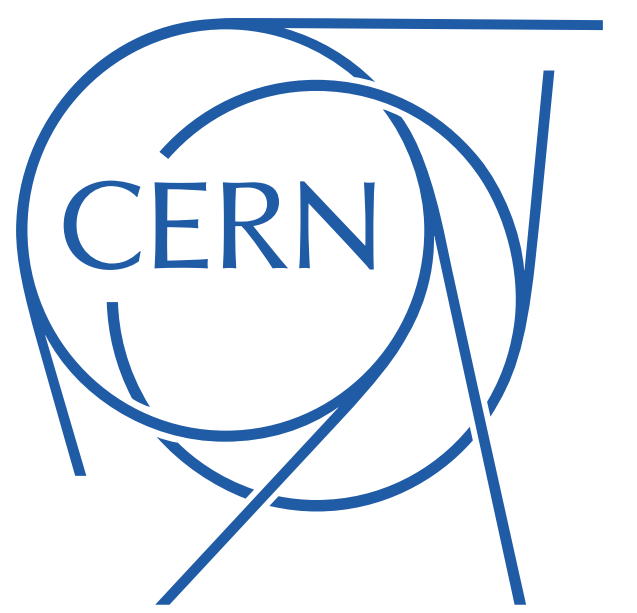


TOWARDS A RESPONSIVE CVMFS ARCHITECTURE

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



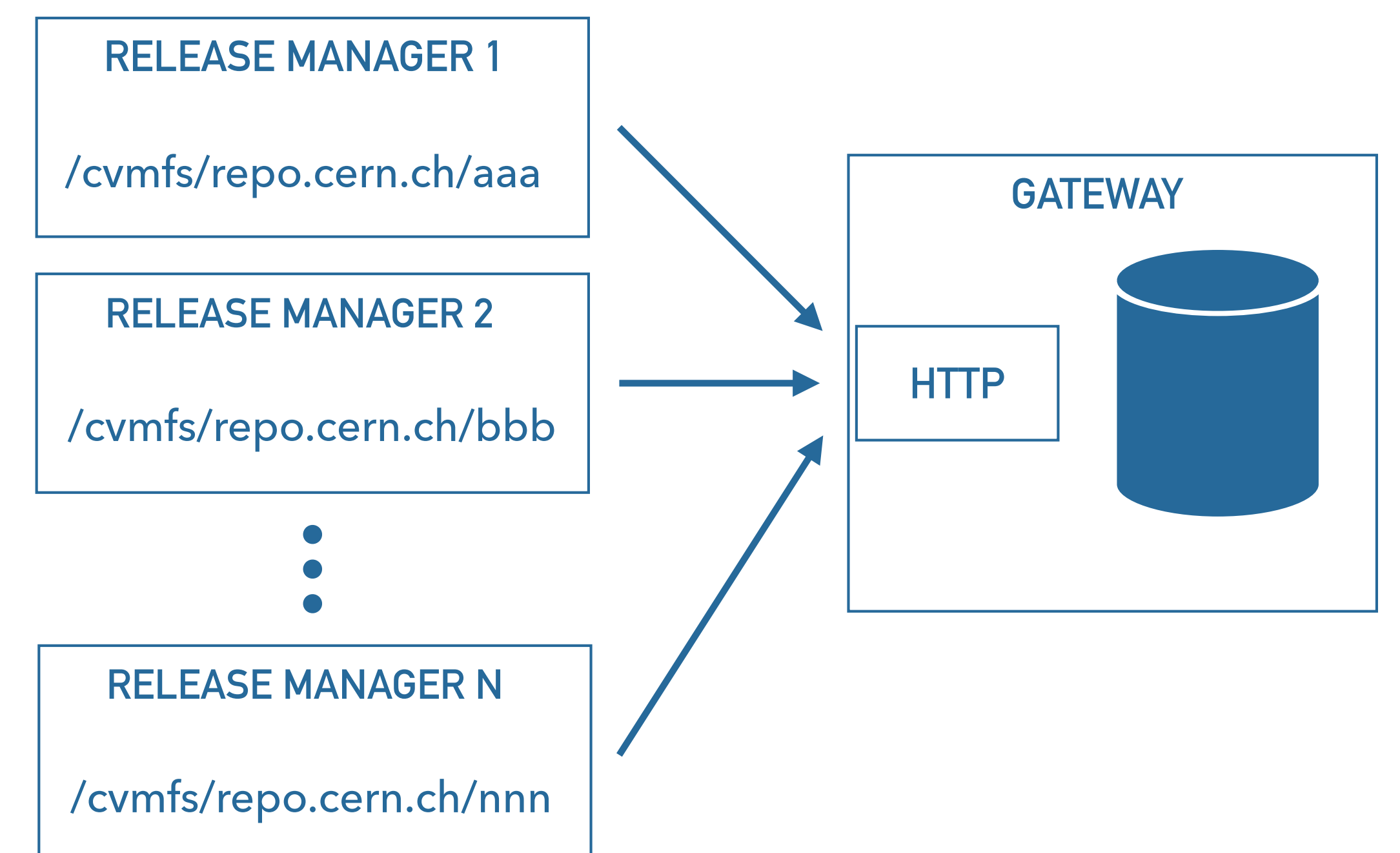
The classic CVMFS architecture involves a single publisher per repository. Clients initiate all communication and request updated versions of repository manifests based on time-to-live (TTL) values. New changesets are seen by clients around five minutes after publication time. This architecture is efficient 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 (CVMFS 2.5)

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

Publishers can **write files concurrently** with each other, to separate subpaths in the repository. This allows dealing with an **increased publication volume** through **horizontal scaling**.

The publication rights of individual release manager machines can be restricted to specific subpaths. **In this way, multi-tenant repositories** can be implemented (i.e. user project areas, container image hubs).

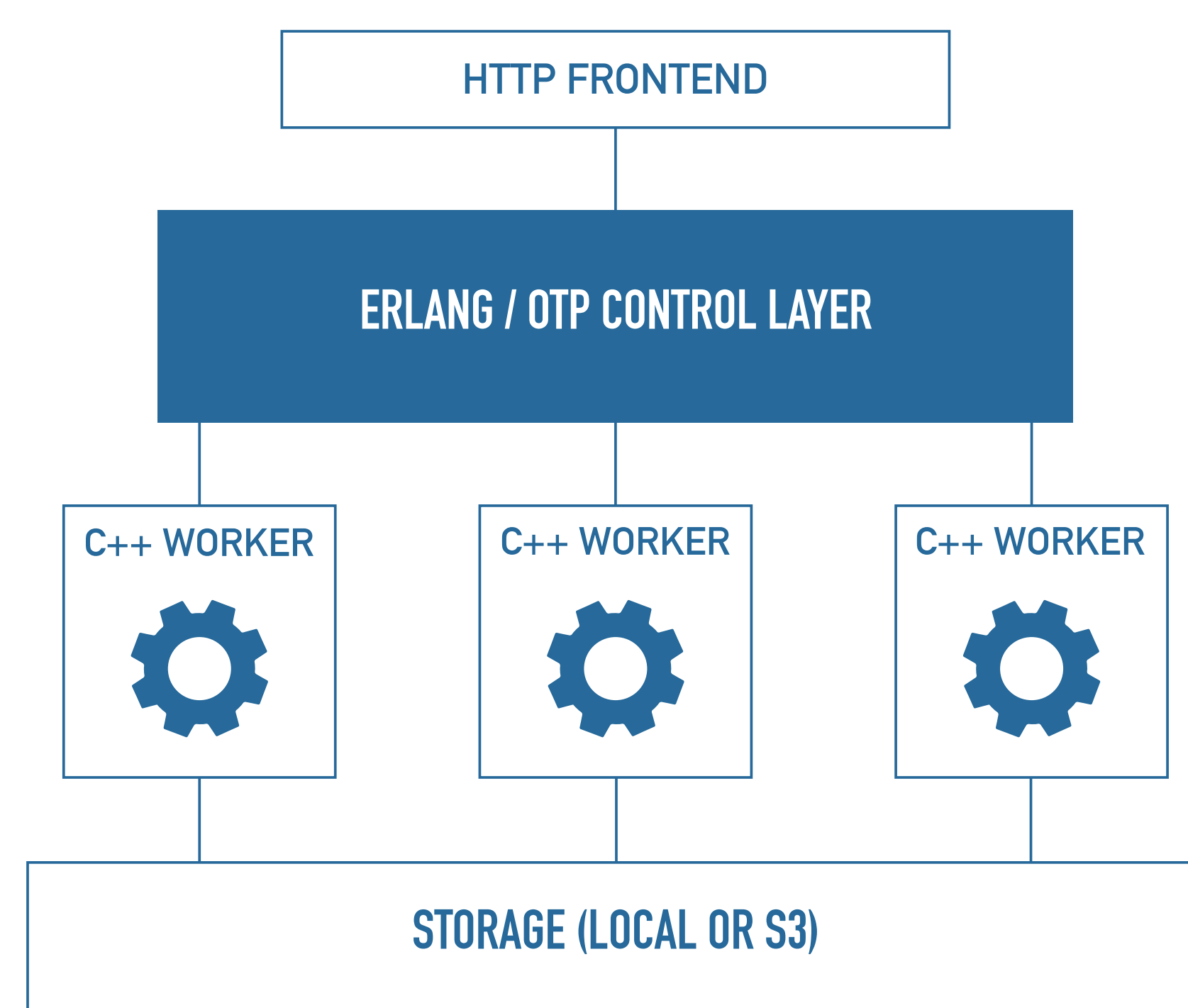


REPOSITORY GATEWAY (CVMFS 2.5)

A new component which receives the payloads from the multiple publishers and commits the changesets into the repository.

It enforces access control policies and hands out exclusive leases to different repository subpaths, to ensure that the repository stays consistent.

It is implemented in **Erlang/OTP**, for **scalability and robustness**.

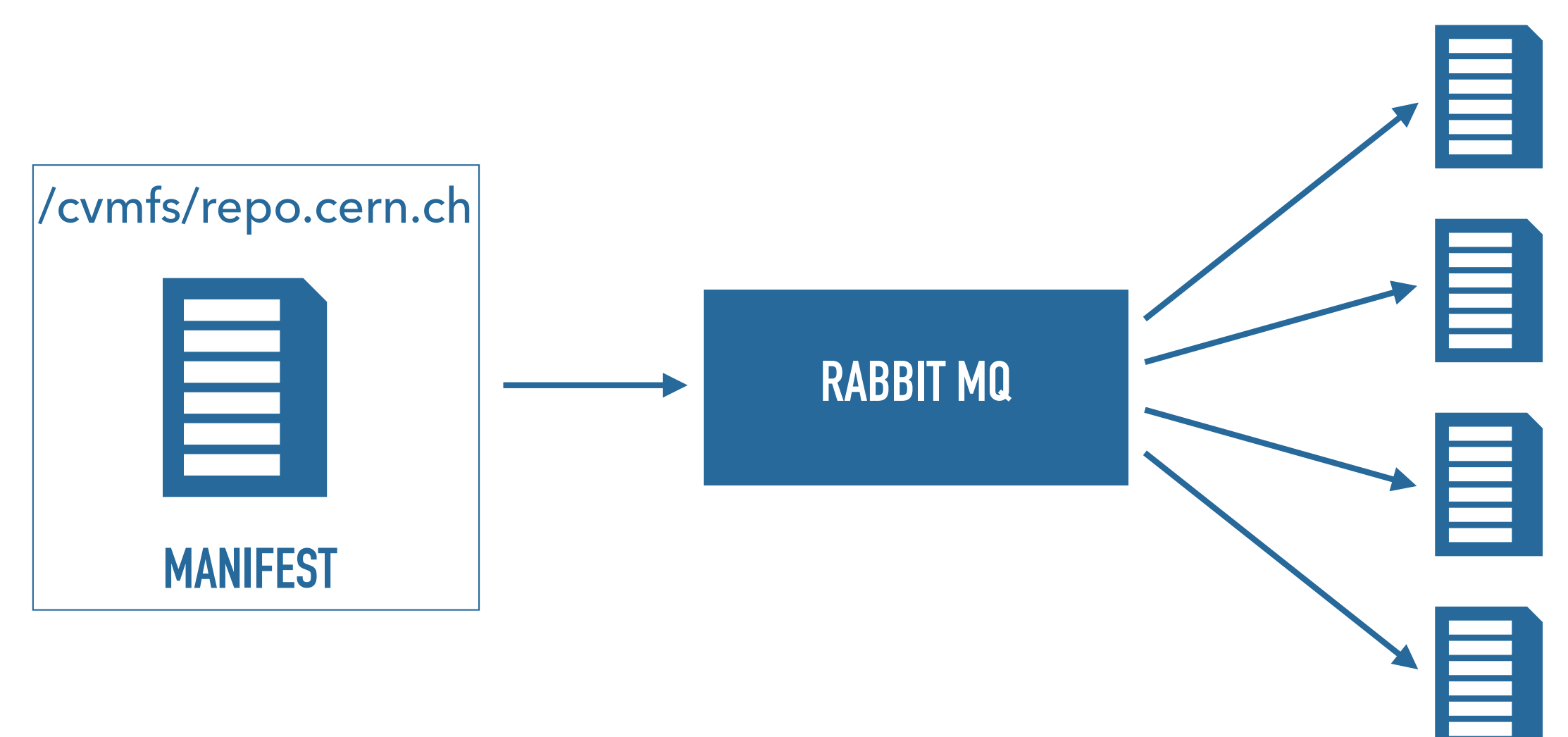


THE NOTIFICATION SYSTEM (COMING SOON IN CVMFS 2.6)

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 **reduces propagation delay**, manifests are dispatched to all the subscribers in a matter of **seconds**.



A PROXY LAYER BASED ON XCACHE (EXPERIMENTAL)

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

Exposes an HTTP server interface, CVMFS clients 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).

Uses RabbitMQ, an efficient, 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.

Collaboration with Andrew Hanushevsky and Wei Yang, SLAC.

THE APPLICATIONS OF THE RESPONSIVE ARCHITECTURE

LOW PROPAGATION DELAY 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

