dCache.org

Macaroons and dCache

… or delegating in a cloudy world

Patrick Fuhrmann
Paul Millar

On behave of the project team

INDIGO DataCloud

Fermilab NDGF DESY HELMHOLTZ ASSOCIATION Federal Ministry of Education and Research LSDMA

AAI … but

This talk is about the second 'A': **Authorisation**.

Quick recap: which is which?

Credential

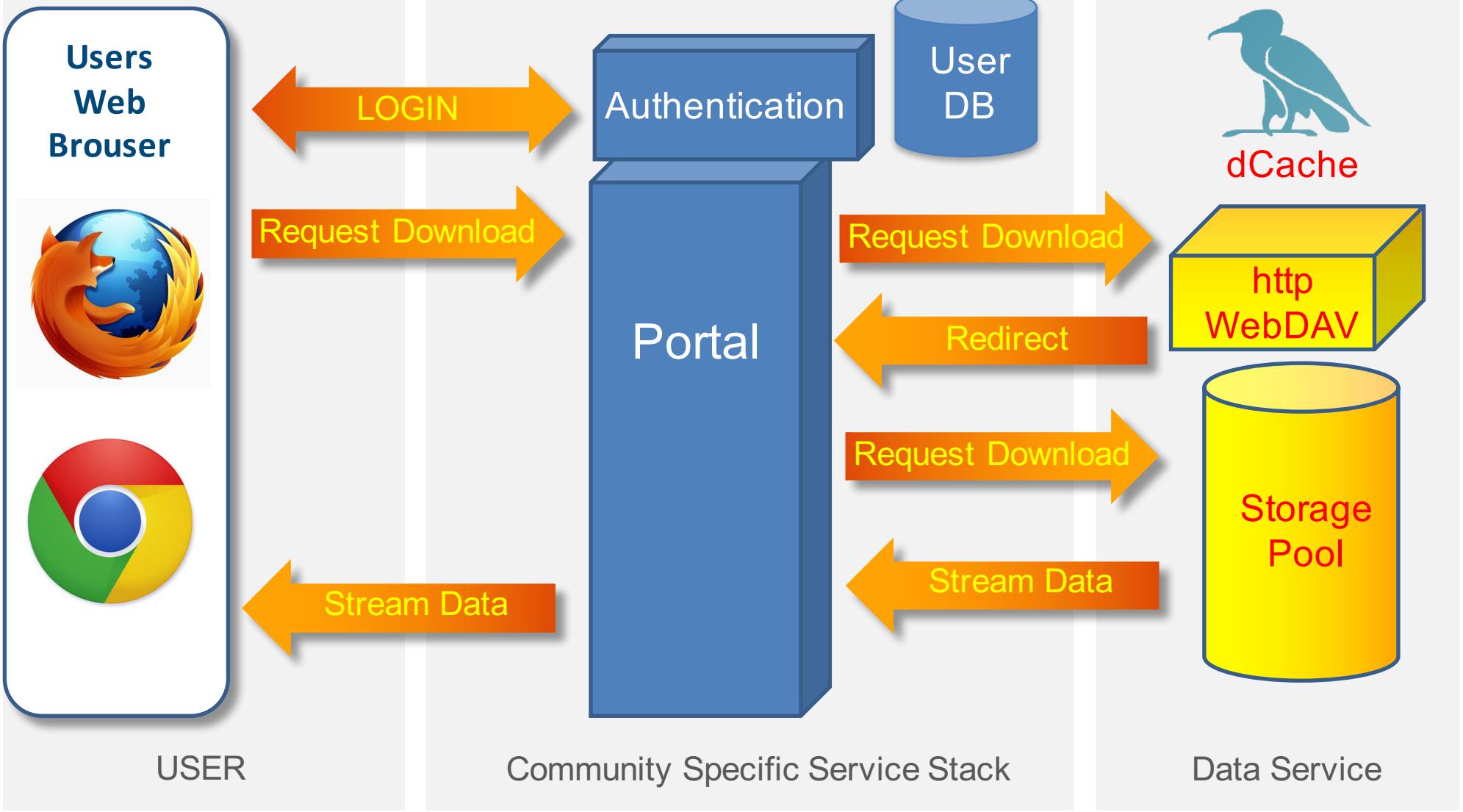
Authentication

Authorization

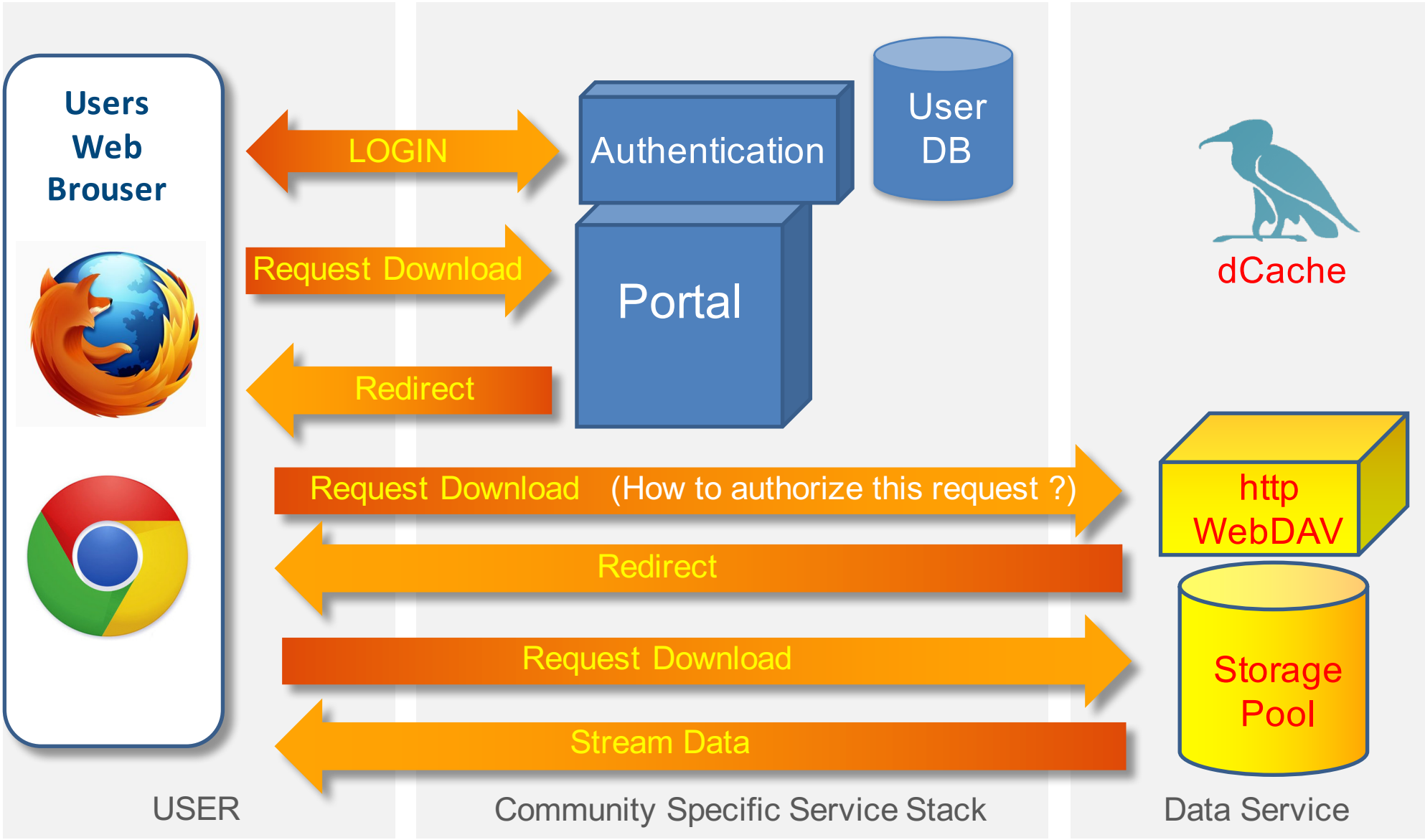
Authorisation without authentication?

That is this all about,
Starting with a use-case

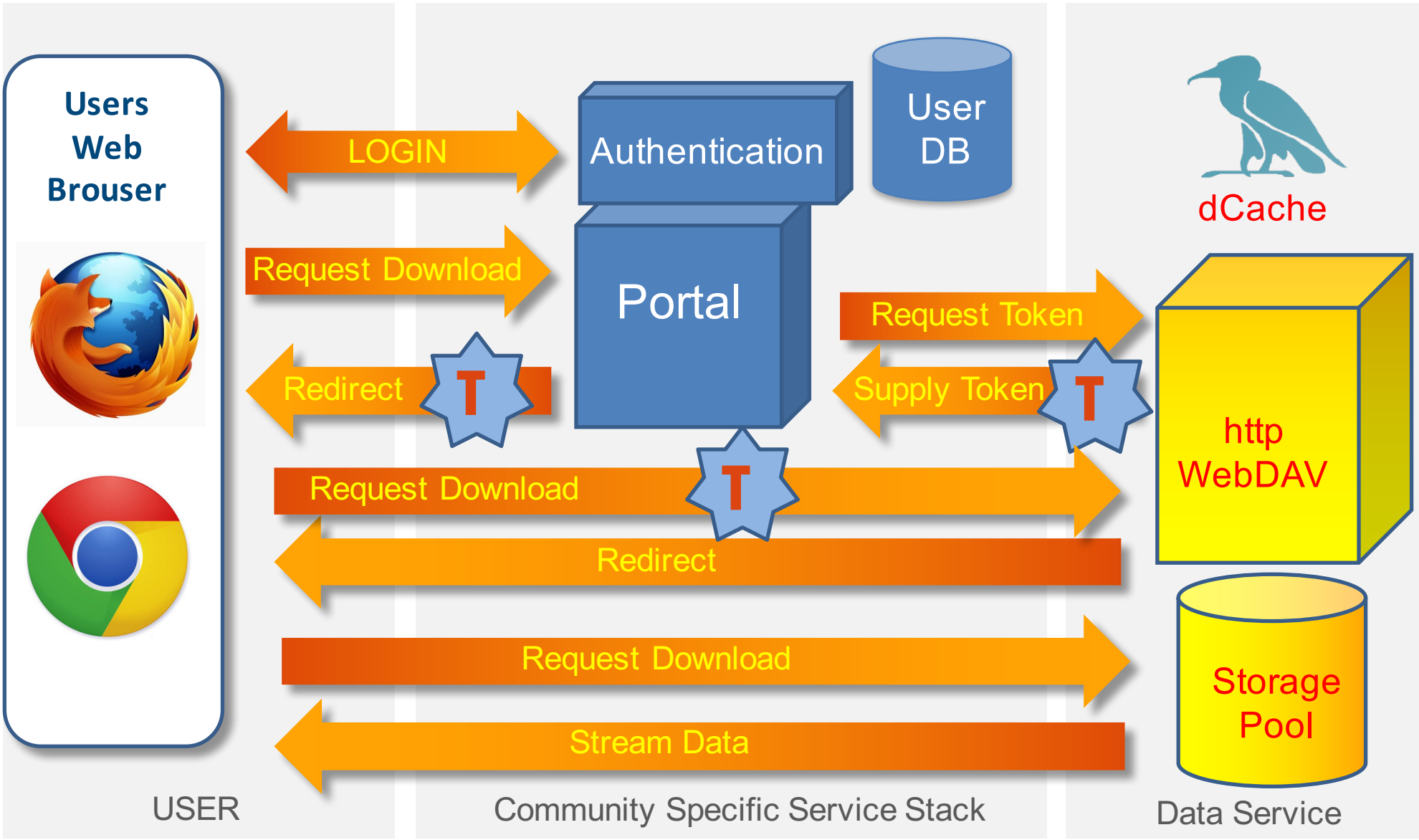
Photon Science portal use-case



Desired: client downloads directly



Desired: client downloads directly



What are bearer tokens?

dCache.org

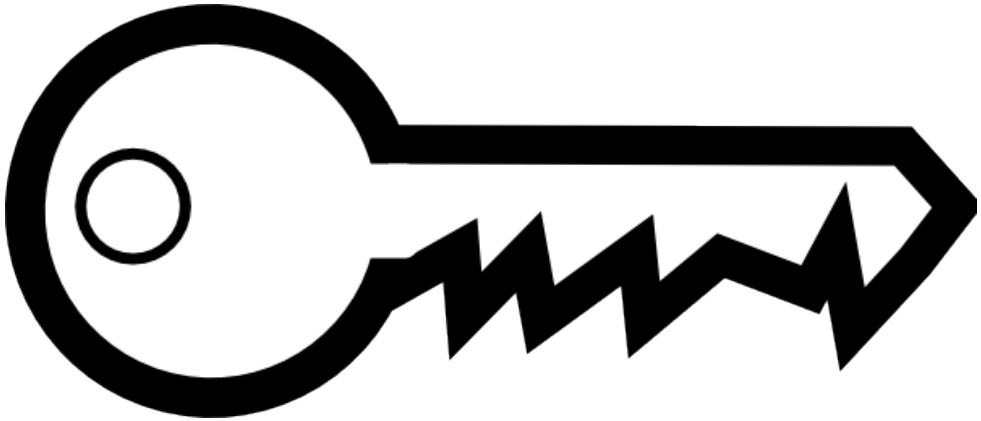
Bearer token is something the user presents with a request so the server will authorise it. There's no interaction between client and server.

Examples of bearer tokens:

- HTTP BASIC authn, anything stored as a cookies.

Counter-examples:

- X.509 credential,
- SAML,
- Kerberos.



Bearer tokens for download authz

dCache.org

- Redirection should work **without JavaScript**,
- Simple: **embed token** in redirection URL.

http://webdav.example.org/path/to/file?authz=<TOKEN>

(There are nicer ways of embedding the token, but the URL is the only thing we can control)

- **Complete token** always sent with the request.
- What can we do to stop someone **stealing** this token?
- … or make the token useless if they steal it.

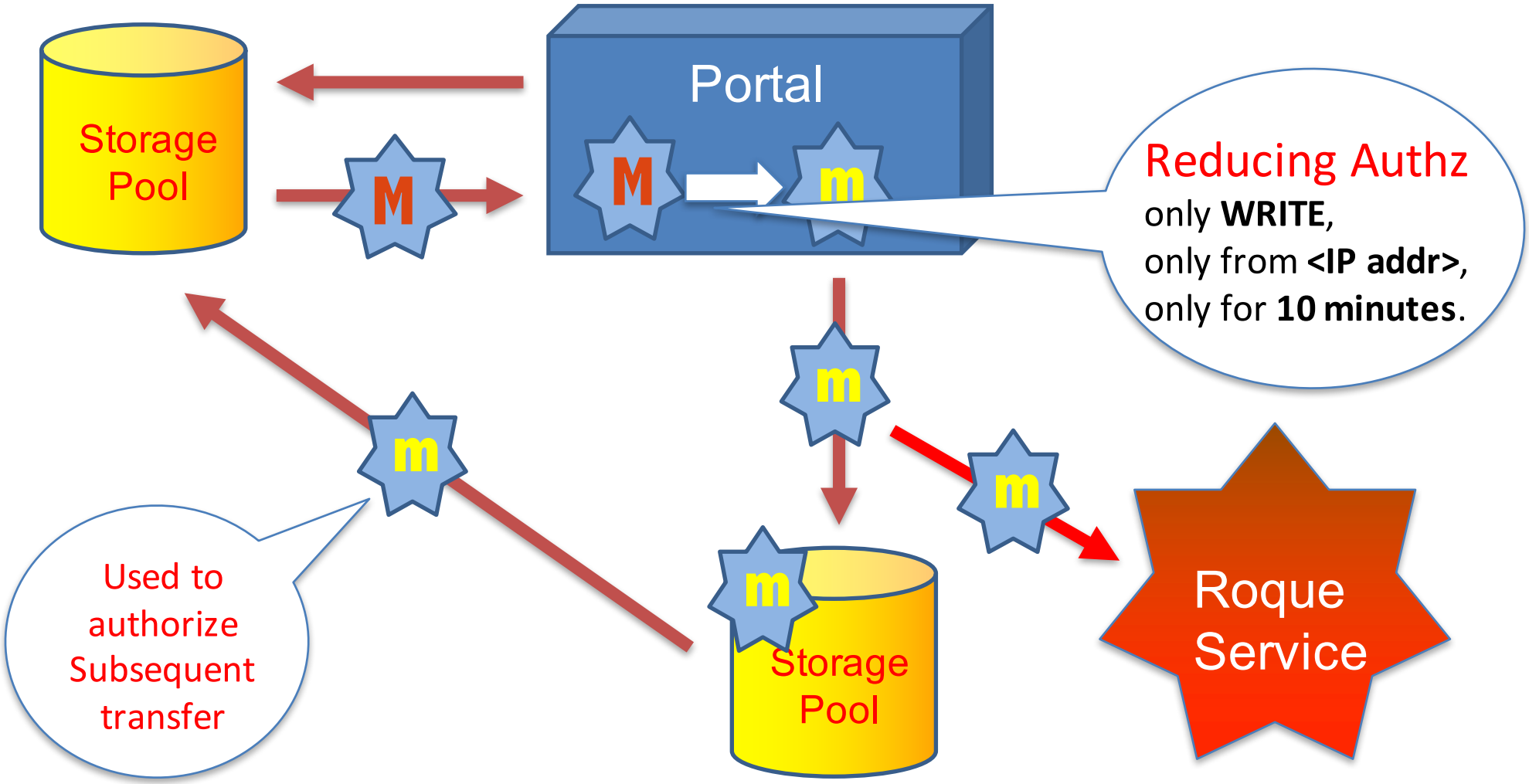
Introducing Macaroons

Macaroons 101

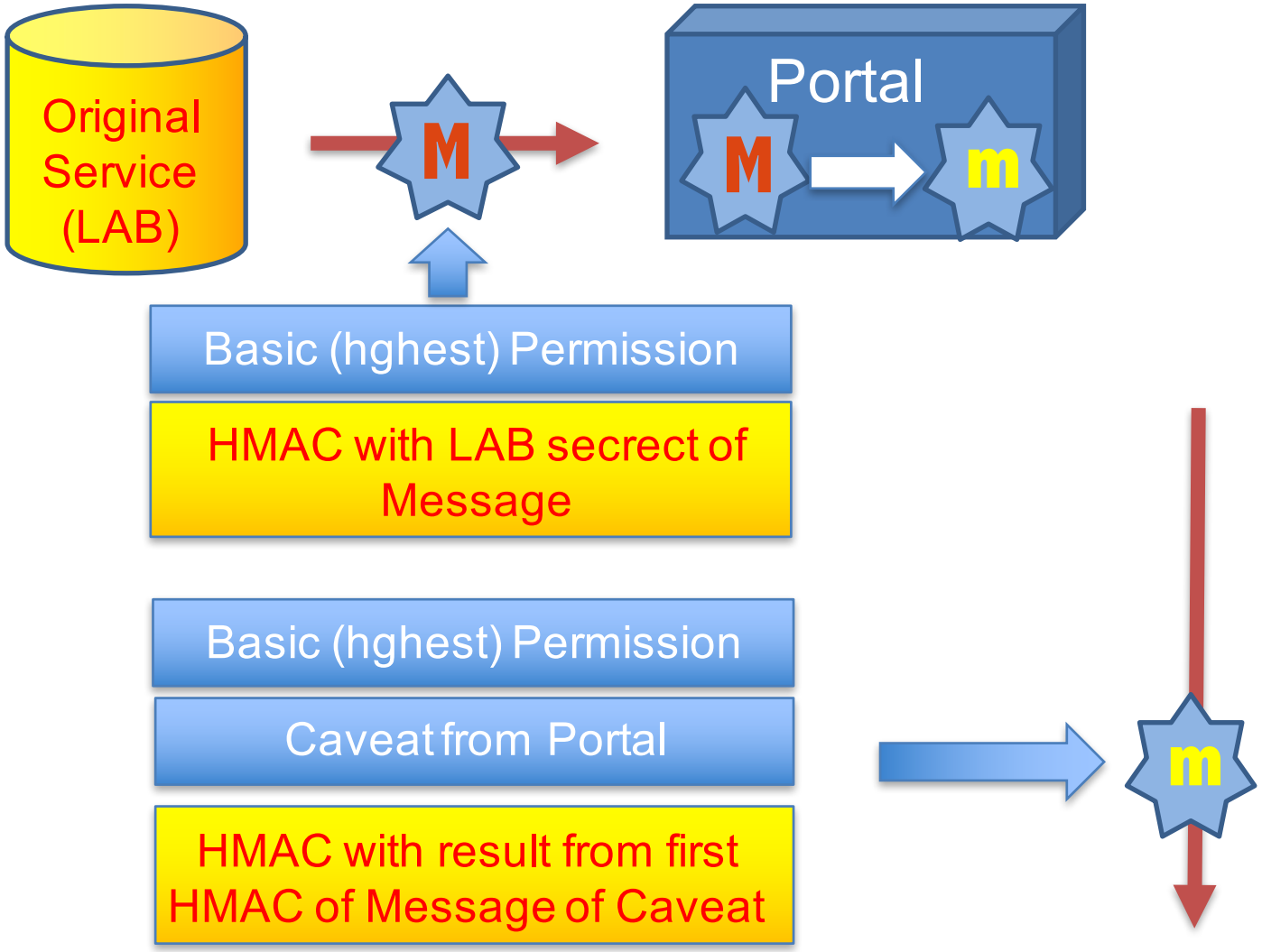
dCache.org

- Macaroon is a **bearer token**.
- Macaroon contains zero or more **caveats**.
- Each caveat **limits** something:
  - **who** can use it, or
  - **what** they do with it.
- Anyone can **add** a caveat to a macaroon:
  - Create a new macaroon that is more limited.
- Nobody can **remove** a caveat from a macaroon.

Example: 3rd party copy



A bit on security



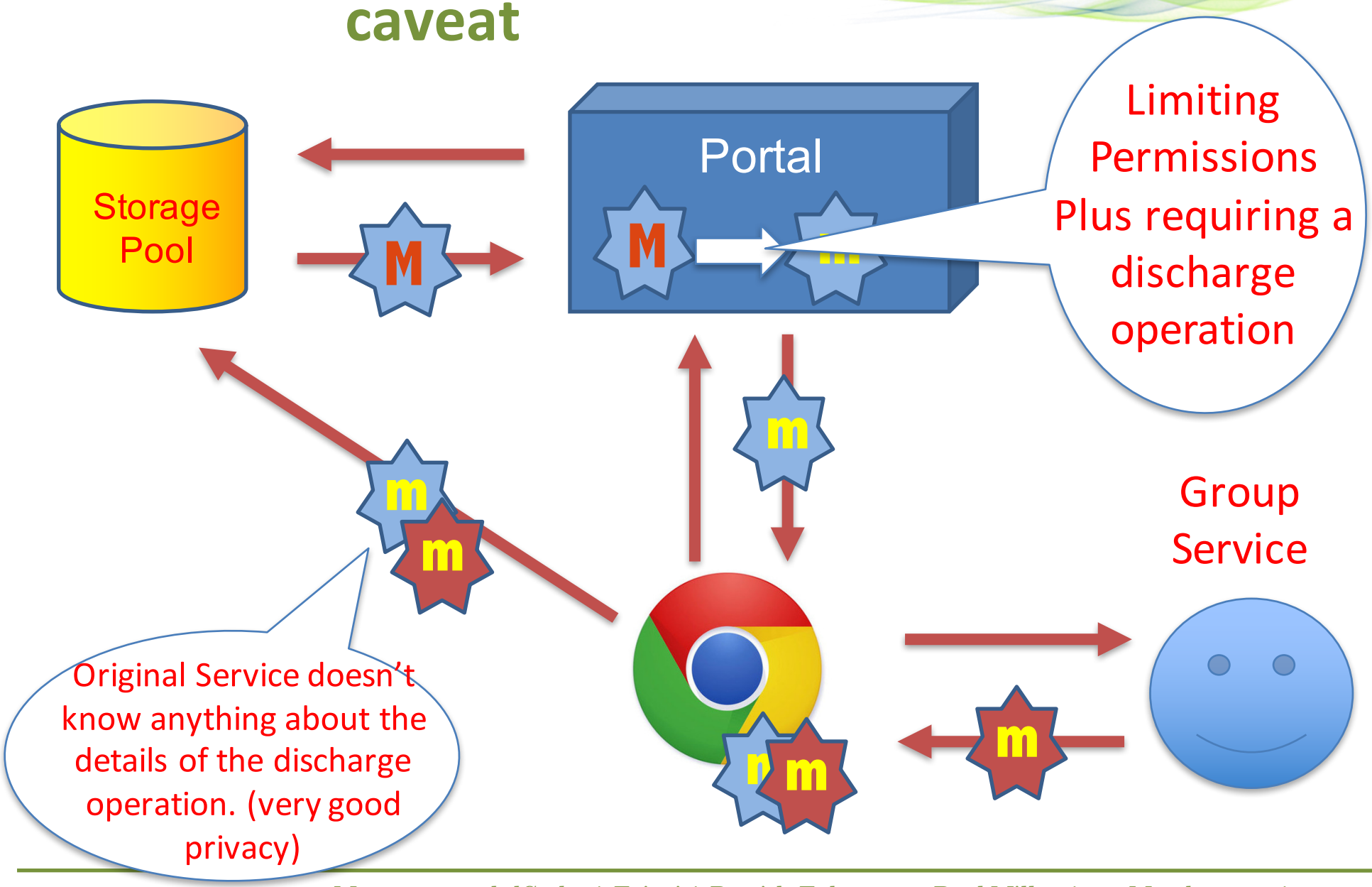
3rd party caveats – extra cool!

dCache.org

- 1st party caveat can be satisfied by the client.
- 3rd party caveat requires proof from some other service; e.g.
  - only **fred@facebook**,
  - only members of **VO ATLAS**,
  - only if not part of a **denial-of-service attack**.
- The proof is another macaroon: a **discharge macaroon**.

Example: download w/3rd party caveat

dCache.org

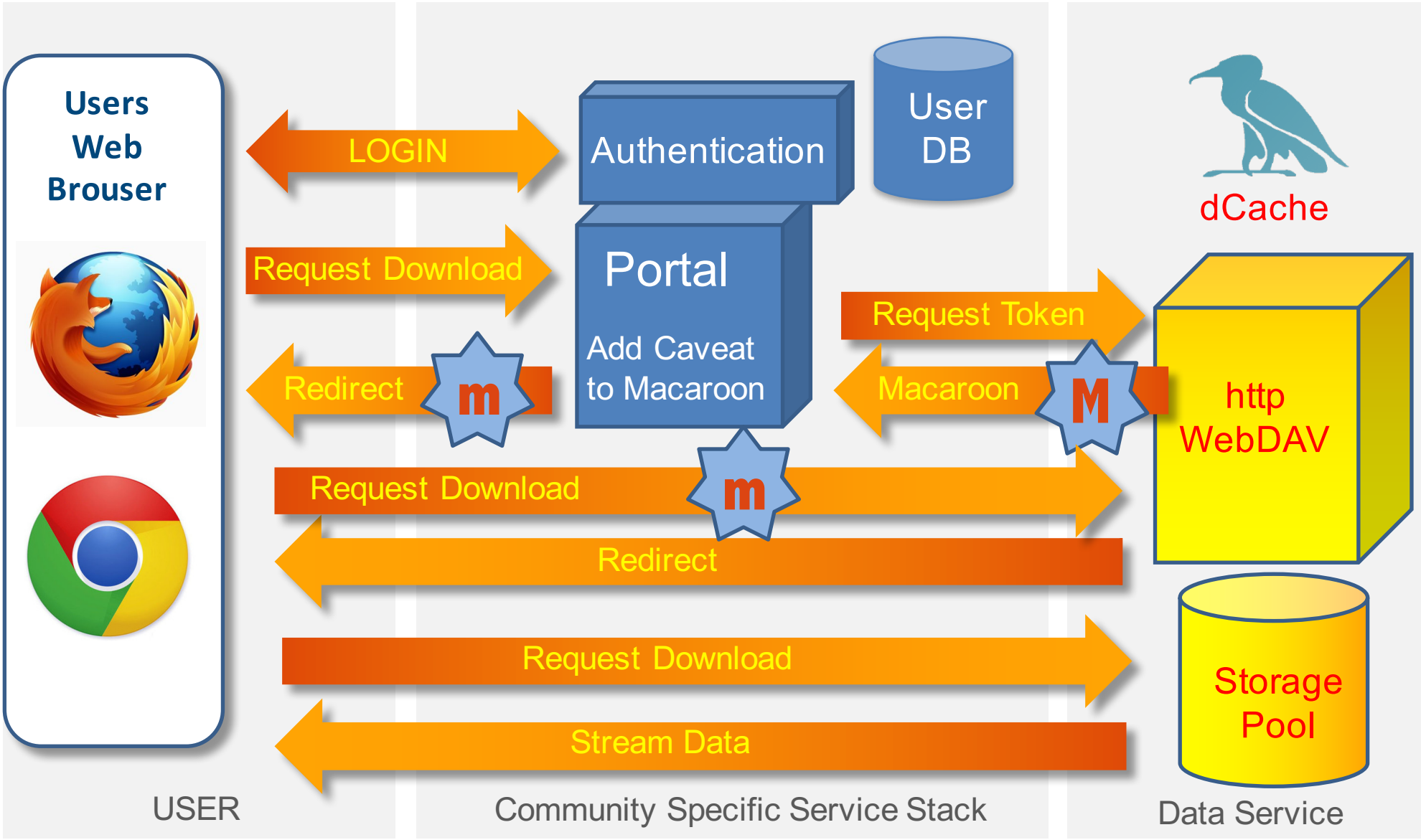


Discharge macaroons

dCache.org

- The client proves it satisfies a 3rd party caveat by having a **discharge macaroon**.
- The original macaroon is only useful with a **valid** discharge macaroon.
- The discharge-macaroon can have **caveats**:
  - Short-lived discharge macaroon can be used to simulate X.509's certificate revocation list.
  - The discharge macaroon can have 3rd-party caveats.

Solution revisited: macaroons



For what else are macaroons good?

Private Sharing!

Enabling sharing: a new interface

dCache.org

- **Create** a macaroon:
  - Need to know the macaroon to access the file.
- **List** macaroons:
  - Facilitate sharing files.
- **Facilitate** adding caveats:
  - Purely in-browser or server-side?
  - Third-party caveats? (e.g., member-of-ATLAS caveat)
- **Destroy** macaroons:
  - Unclear if this really makes sense.

The END

dCache.org

Further reading :

On dCache

www.dCache.org

On macaroons by Google:

Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud.

Presentation

Paper

http://research.google.com/pubs/pub41892.html