

A View on the CS3 API

Klaas Freitag, CTO ownCloud GmbH

Overview

- 1 CS3 APIs and Infinite Scale
- 2 On the edge...
- 3 Infinte Scale Spaces
- 4 LibreGraph API
- 5 HTTP APIs and CS3



Klaas Freitag
Chief Technical Officer

The CS3 APIs and Infinite Scale:

Infinite Scale uses Reva as one of its essential building blocks.

Both systems have influenced each other since the beginning

Infinite Scale uses the Reva components and starts them as part of its services.

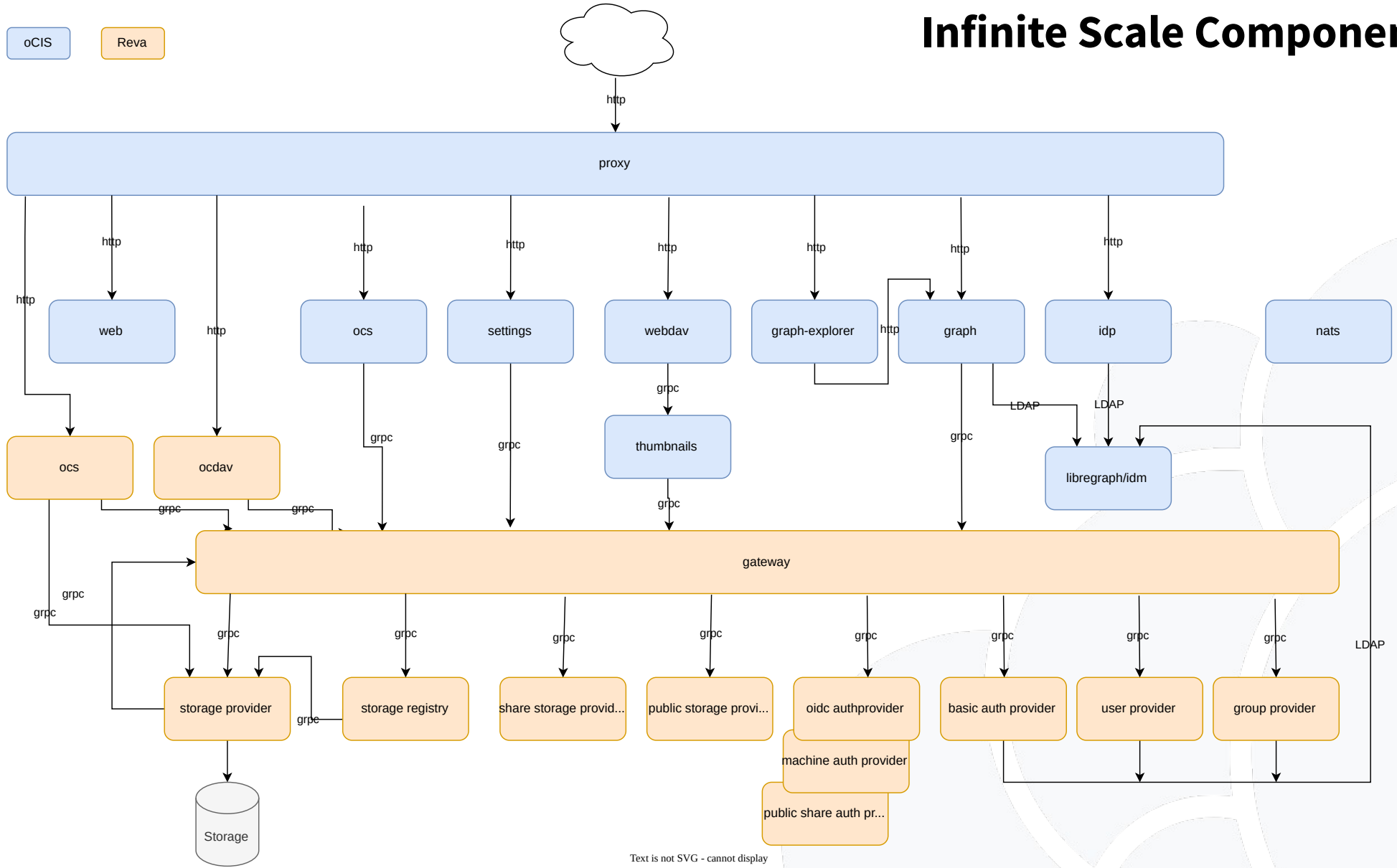
Infinite Scale adds more functionality that is not (yet) in Reva on top.

With Infinite Scale, you get Reva (the CS3 API) as a product.



oCIS Reva

Infinite Scale Components



Text is not SVG - cannot display

Reva master and edge

Edge Branch

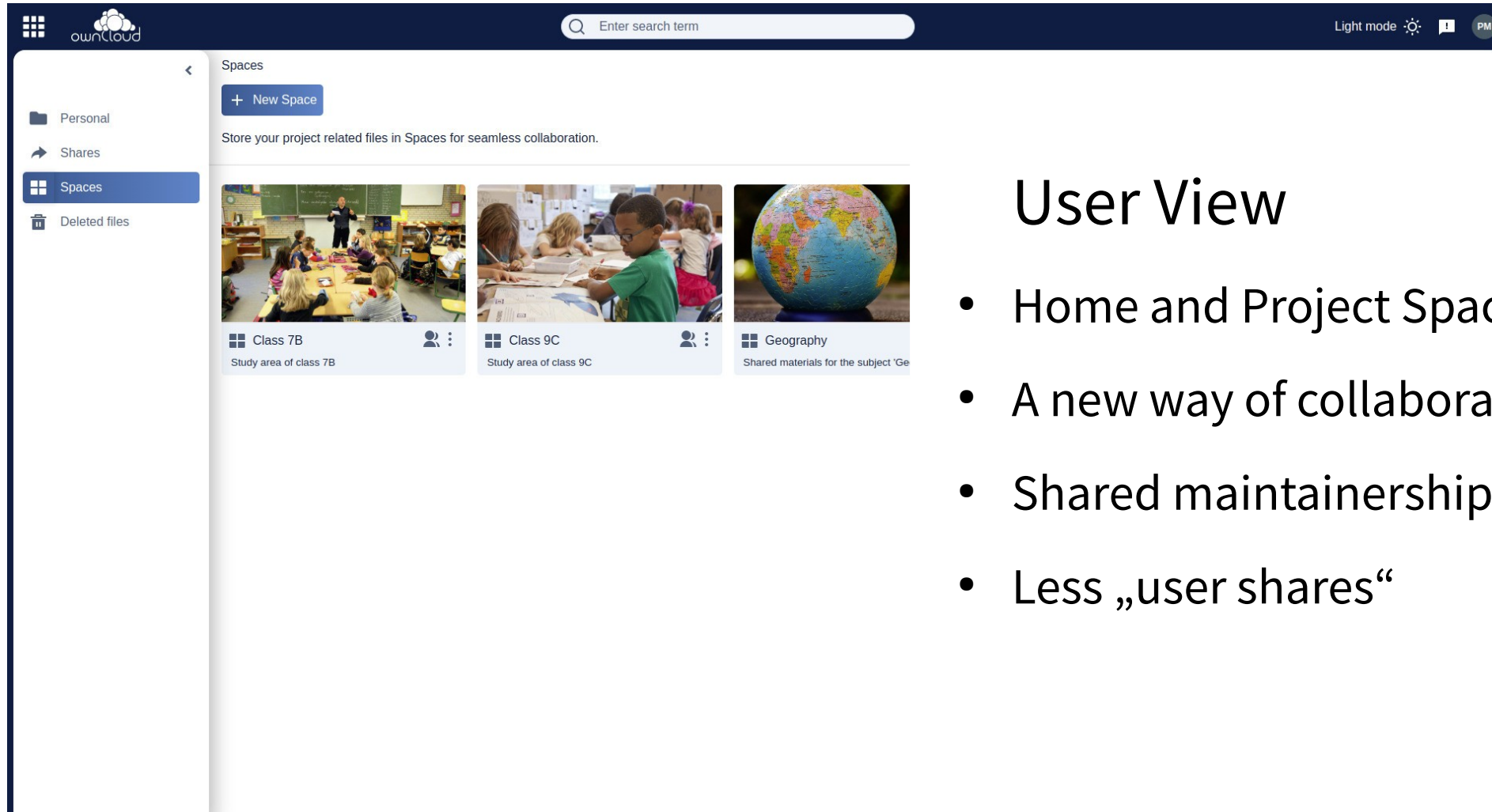
- Spaces: Not a single file tree, but independent file spaces.
- LibreGraph API
- More features such as asynchronous upload etc.

Master Branch

- OCM
- Integrations with EOSC
- EOS support
- Other, CERN specific features

Convergence of the branches is ongoing

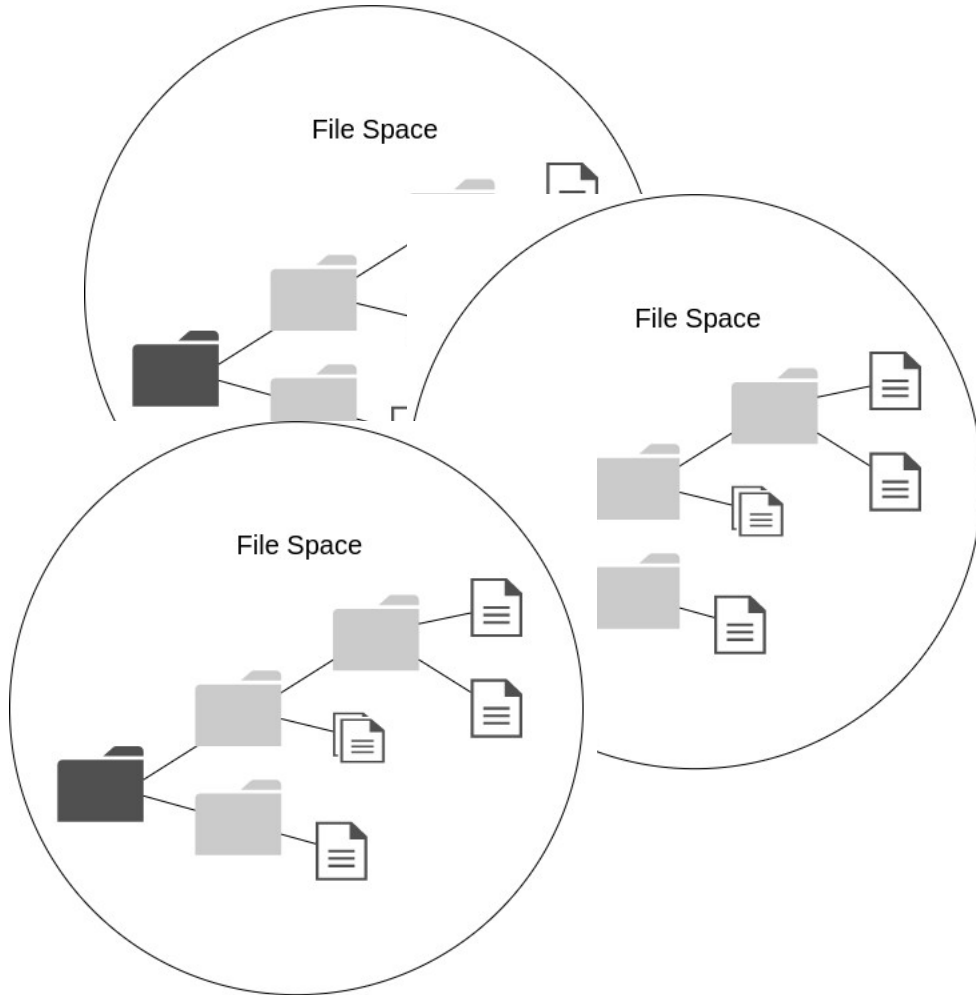
Spaces Concept



User View

- Home and Project Spaces
- A new way of collaboration
- Shared maintainership of Spaces
- Less „user shares“

Spaces Concept



Benefits from technical POV:

- Different storage types for spaces
- Clearer separation of concerns
- Better scalability
- Operations more fine granular
- Simpler Federation

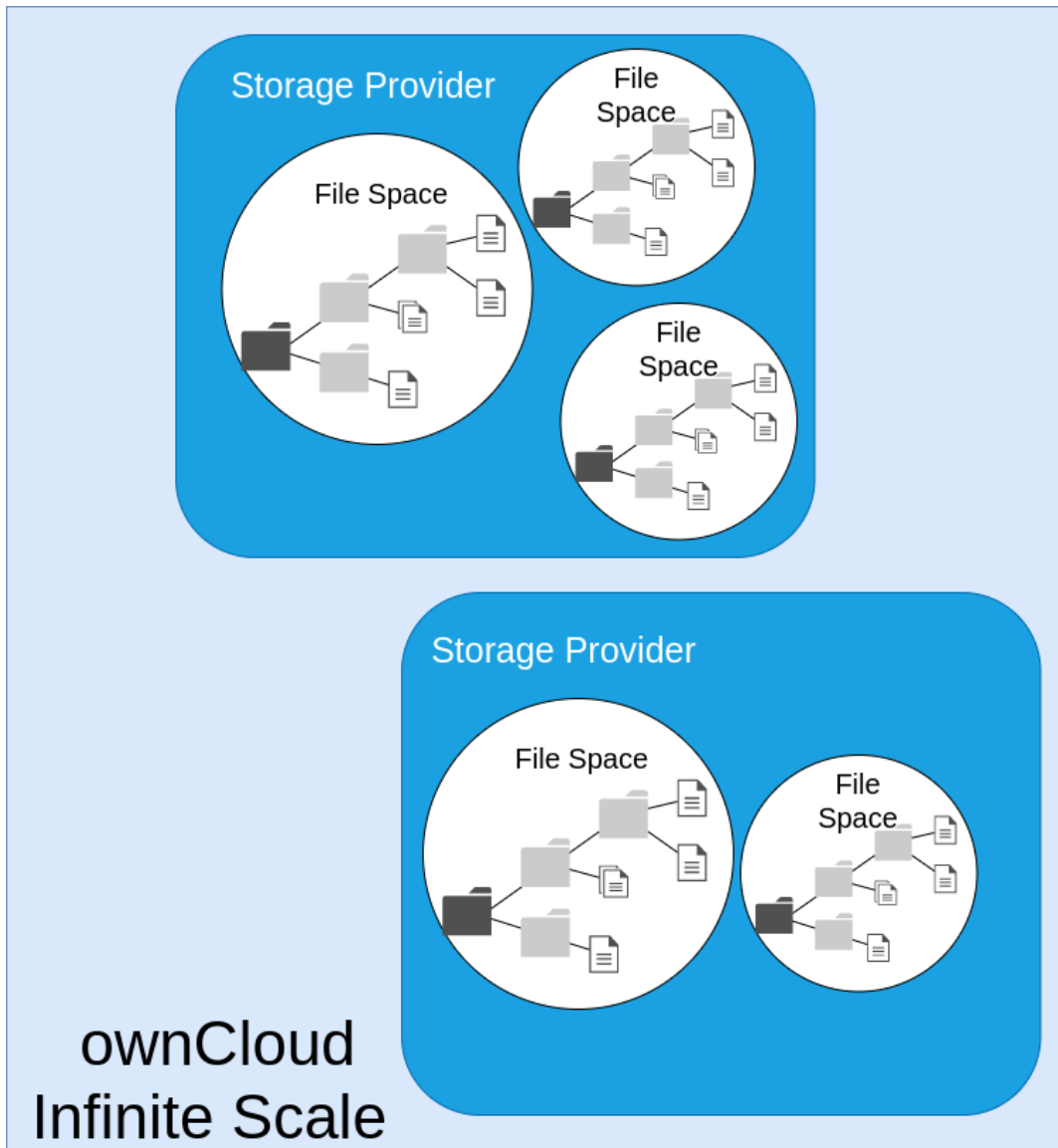
Spaces Concept

If you try to have everything in one tree...





... but this is what we try to connect, especially in federation!



For the APIs that means:

- A resource is identified by a tuple of UUIDs for
 - storage provider ID
 - space ID
 - file ID or relative path
- Legacy APIs still work but are inefficient
- Shares are also „Spaces“
- Path alias are supported for URLs

[https://ocis/files/spaces/personal/einstein?fileId=<storage-prov-id\\$space-id!fileid>](https://ocis/files/spaces/personal/einstein?fileId=<storage-prov-id$space-id!fileid>)

LibreGraph API

A Microsoft Graph [1] inspired API für different functions in Infinite Scale.

Microsoft developed the Restful Graph API to access Microsoft Cloud service resources.

Examples for usage in Infinite Scale:

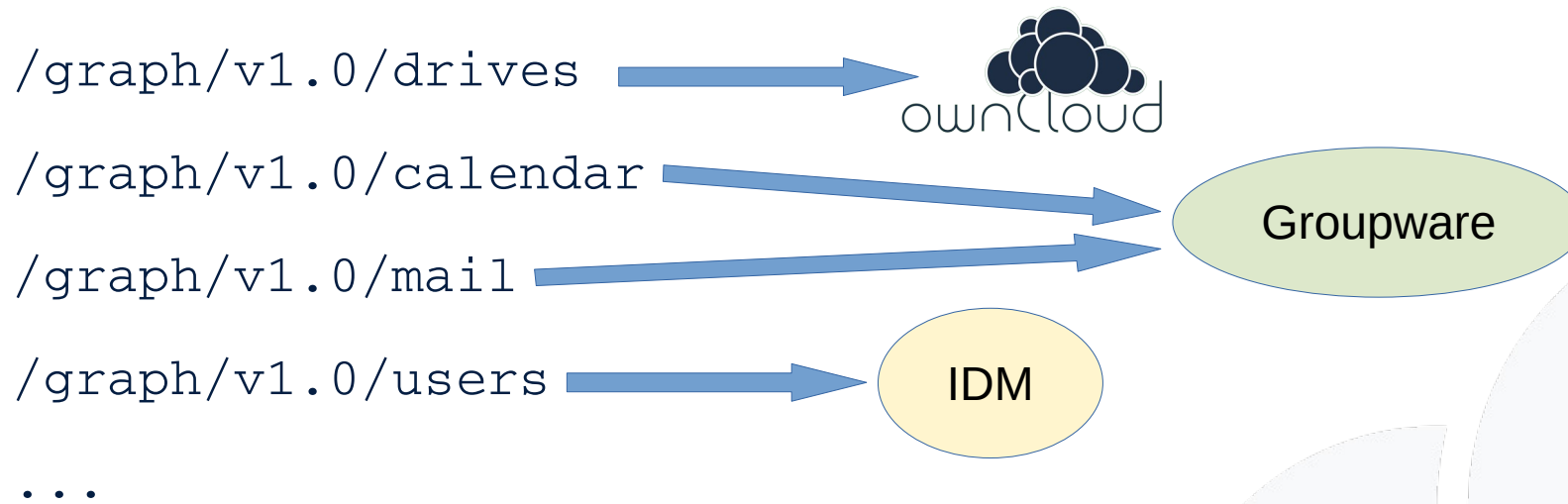
- Spaces listing and -management
- User and Roles Provisioning
- Tags and Meta management

[1] <https://learn.microsoft.com/en-us/graph/>
<https://owncloud.dev/libre-graph-api>
<https://libregraph.github.io>

LibreGraph API Idea

The MS Graph API allows access to Microsoft Cloud service resources.

The LibreGraph API can be a **drop in replacement** implemented by different FOSS services providing different endpoints of the API

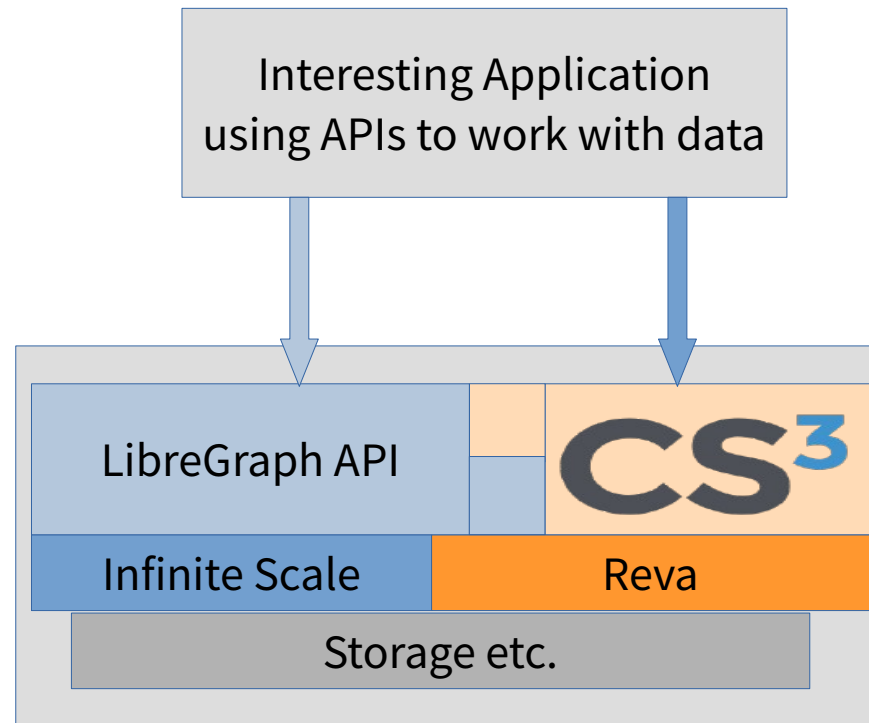


This could be an important building block for a free implementation of a well known API. It integrates free services and avoids vendor lock-in.

LibreGraph and CS3?

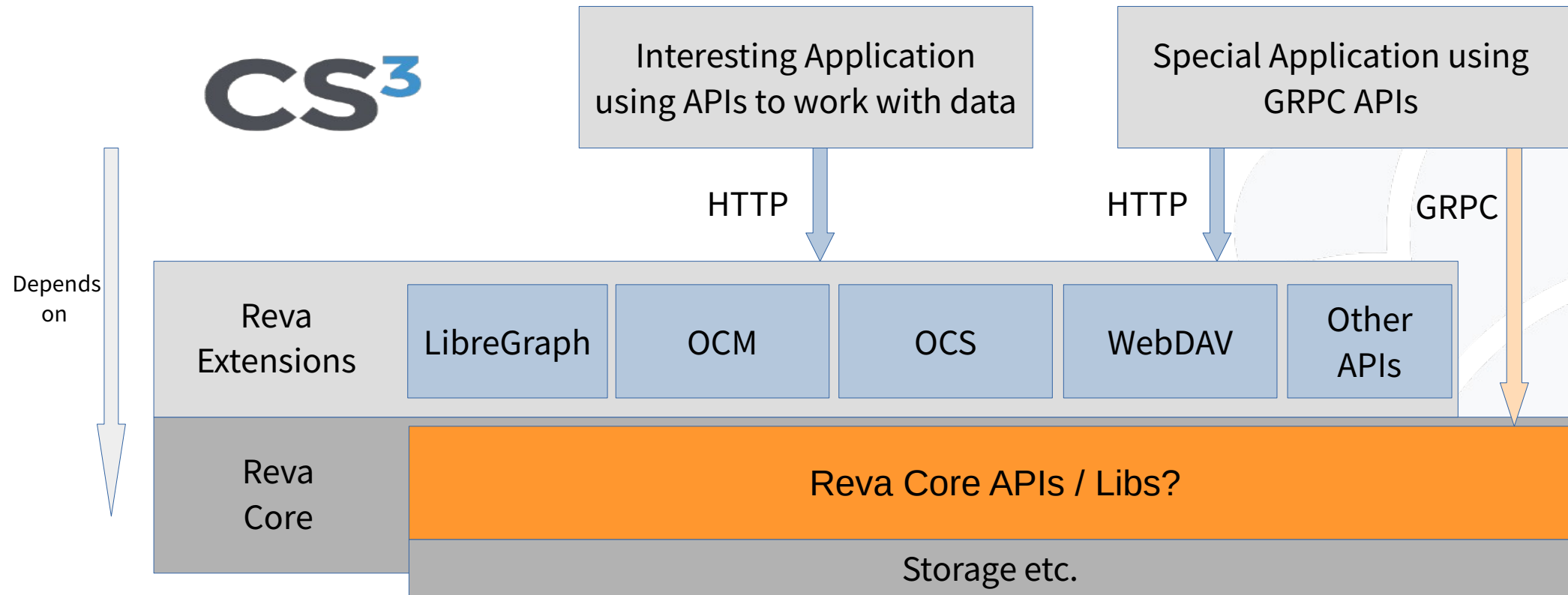
The LibreGraph API overlaps and integrates with APIs that are already part of the CS3 APIs.

Current Situation:



HTTP APIs and CS3?

The LibreGraph API overlaps and integrates with APIs that are already part of the CS3 APIs.



Thank you very much

CS3 Conference 2023

WebDAV API Response example

```
<d:response>
  <d:href>/dav/files/einstein/my%20Folder/eule.jpg</d:href>
  <d:propstat>
    <d:prop>
      <oc:id>03ec6a92-5988-4016-ad90-de9291dd74dd$4c510ada-c86b-4...</oc:id>
      <oc:fileid>03ec6a92-5988-4016-ad90-de9291dd74dd$4c510ada-c86b-4...</oc:fileid>
      <oc:spaceid>4c510ada-c86b-4815-8820-42cdf82c3d51</oc:spaceid>
      <oc:file-parent>03ec6a92-5988-4016-ad90-de9291dd74dd$4c510ada-c86b-4815-8820-4...</oc:file-parent>
      <oc:name>eule.jpg</oc:name>
      <d:getetag>&quot;1f2b7628db83de0e88d56bff5cbb9570&quot;</d:getetag>
      <oc:permissions>RDNVWZ</oc:permissions>
      <d:resourcetype/>
      <d:getcontentlength>2937595</d:getcontentlength>
      <d:getcontenttype>image/jpeg</d:getcontenttype>
      <d:getlastmodified>Sun, 15 Jan 2023 14:16:35 GMT</d:getlastmodified>
      <oc:checksums>
        <oc:checksum>SHA1:a183fd4cf96e18adb7207a34ba41524276409712 ADLER32:cf0f035c</oc:checksum>
      </oc:checksums>
      <oc:favorite>0</oc:favorite>
    </d:prop>
    <d:status>HTTP/1.1 200 OK</d:status>
  </d:propstat>
</d:response>
```