Files in Invenio v3

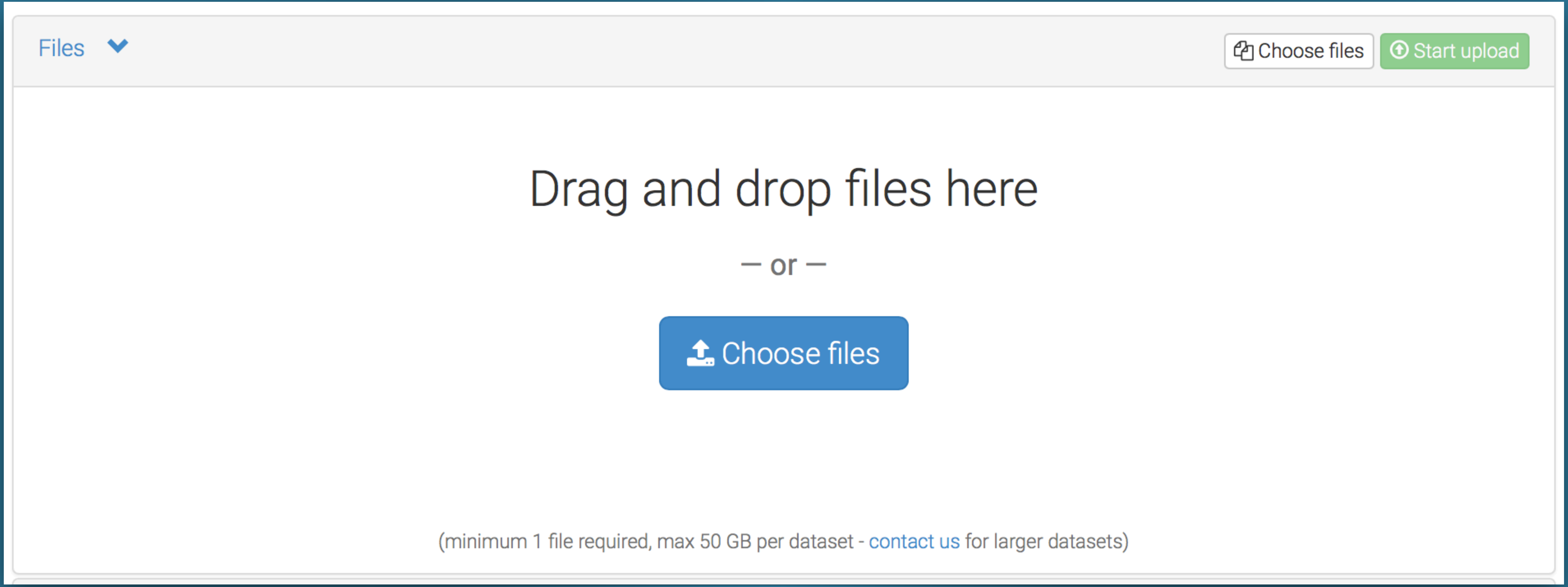
Lars Holm Nielsen
CERN

Invenio User Group Workshop 2017, Garching bei München

INVENIO

MLZ
Heinz Maier-Leibnitz Zentrum

Demo



Files

Choose files

Start upload

Drag and drop files here

— or —

Choose files

(minimum 1 file required, max 50 GB per dataset - contact us for larger datasets)

File upload + quotas + fixity check

Lesson learned from v1:

Do not move files

Do not access files

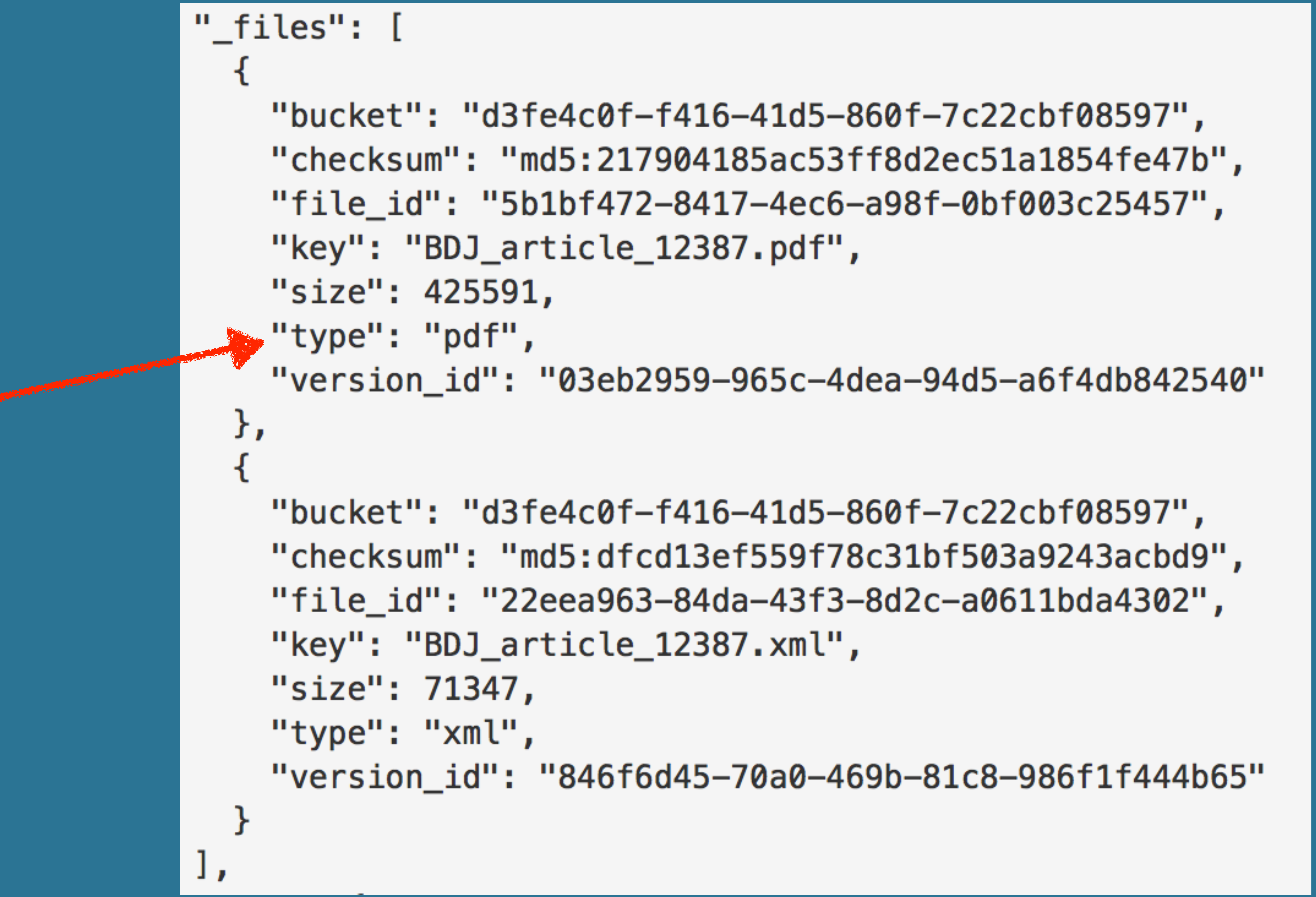
Files

- **Locations (think storage system)**
  - Example: */opt/invenio/var/data/files*
- **Buckets (think directories)**
  - UUID, size, quota, locked, deleted.
- **Objects (think files)**
  - Example: *unicorn.jpg*, but also *path/to/unicorn.jpg* (i.e. you can have “subdirectories” in a bucket)

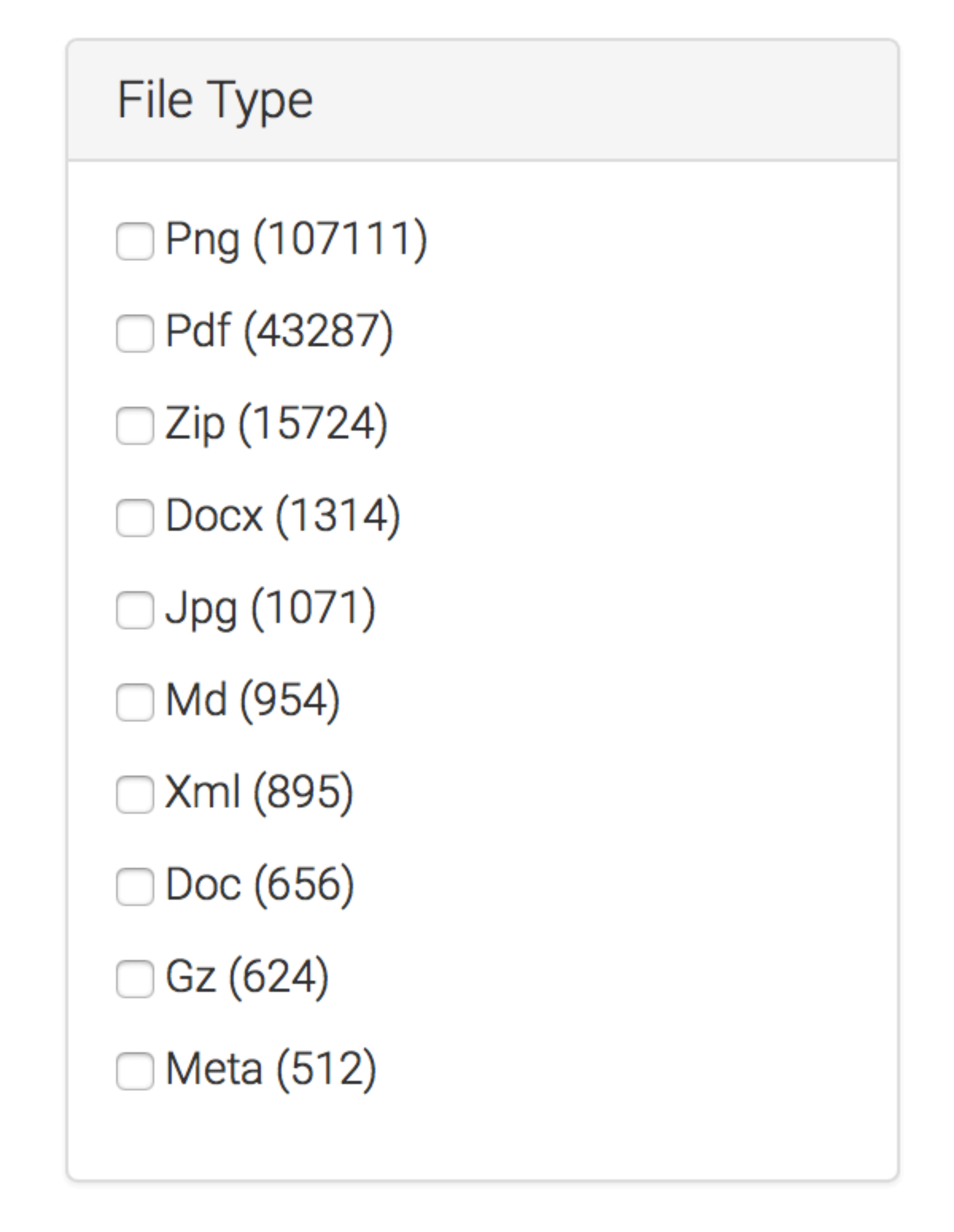
Versioning

- **Object Versions**
  - Normal version: has a file
  - Delete marker: has no file
- **File instance (phyiscal file on disk)**
  - **URI**, size, checksum, writeable, readable, last fixity check

Files for records



```
"_files": [
  {
    "bucket": "d3fe4c0f-f416-41d5-860f-7c22cbf08597",
    "checksum": "md5:217904185ac53ff8d2ec51a1854fe47b",
    "file_id": "5b1bf472-8417-4ec6-a98f-0bf003c25457",
    "key": "BDJ_article_12387.pdf",
    "size": 425591,
    "type": "pdf",
    "version_id": "03eb2959-965c-4dea-94d5-a6f4db842540"
  },
  {
    "bucket": "d3fe4c0f-f416-41d5-860f-7c22cbf08597",
    "checksum": "md5:dfcd13ef559f78c31bf503a9243acbd9",
    "file_id": "22eea963-84da-43f3-8d2c-a0611bda4302",
    "key": "BDJ_article_12387.xml",
    "size": 71347,
    "type": "xml",
    "version_id": "846f6d45-70a0-469b-81c8-986f1f444b65"
  }
],
```



File Type

- □ Png (107111)
- □ Pdf (43287)
- □ Zip (15724)
- □ Docx (1314)
- □ Jpg (1071)
- □ Md (954)
- □ Xml (895)
- □ Doc (656)
- □ Gz (624)
- □ Meta (512)

File metadata
vs
File operations

File storage abstraction

- FileStorage
    - open
    - send_file
    - save
    - copy
    - initialize
    - update
    - checksum
    - delete

File storage: Open

- PyFilesystem:
  - /opt/invenio/var/instance
  - http://…
  - ftp://
  - root://
  - s3://

File storage: Send file

- HTTP Redirect: Let another system serve your file
- Apache/Nginx: Send file or proxy request
- Python: Stream file through Python.

File storage: Save

- File upload: binary data directly in HTTP request.
- Server receives a binary data stream
- Stream is written once to storage disk

File storage: Checksum

- Option 1: Let Invenio compute checksum
- Option 2: Let storage system compute checksum

Server configuration

- Option 1: Let another system send the file to the client.
- Option 2: Let Invenio send the file:
  - Use UWSGI
  - Watch the timeouts

All the other stuff

- Permissions
- Dynamic quotas (e.g. per user, per community, per ?)
- File migration
- Multipart file upload
- Upload by URL
- Previewers