HELMHOLTZ

RESEARCH FOR
GRAND CHALLENGES

HIFIS | HELMHOLTZ FEDERATED IT SERVICES

HIFIS backbone transfer service: FTS for everyone

Tim Wetzel[1], Paul Millar[1], Uwe Jandt[1], Patrick Fuhrmann[1],
Stefan Helmert[2], Tobias Huste[2], Guido Juckeland[2]

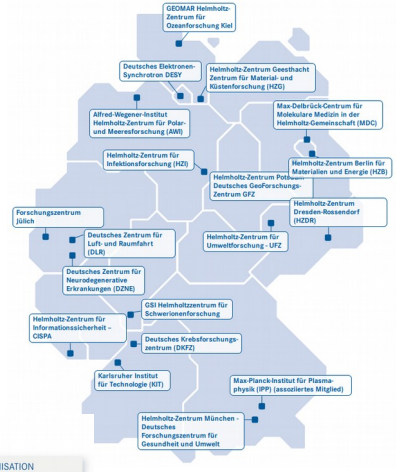
[1] Deutsches Elektronen-Synchrotron DESY,
[2] Helmholtz-Zentrum Dresden-Rossendorf HZDR

HIFIS-Meeting, 22 October 2020

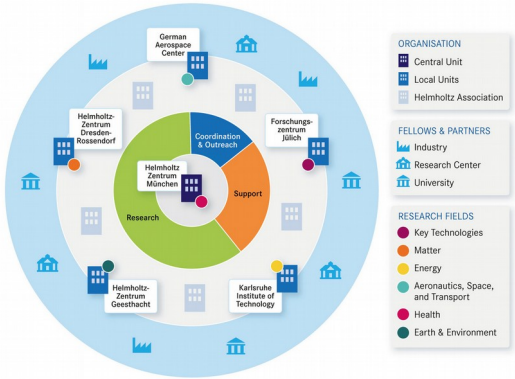
Why data transfers?

- Helmholtz centres **distributed** all over Germany
- Large data sets in **collaborative** research projects
- **Policy-driven** data transfers required
- Data analysis often sensitive to **latency**
- **Data locality** is important!
- Part of **HIFIS** backbone contract

➔ Reliable, comfortable and robust transfer methods needed



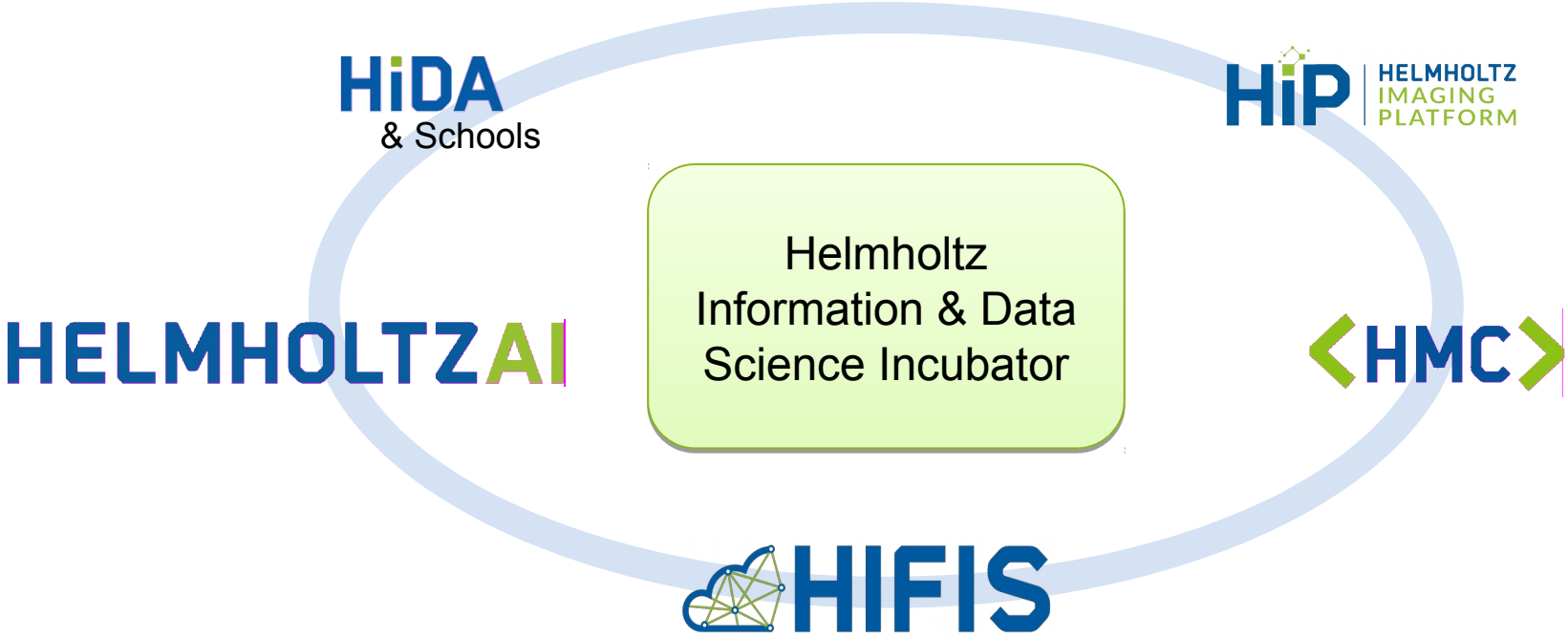
Helmholtz centres in Germany



helmholtz.ai structure

Introduction

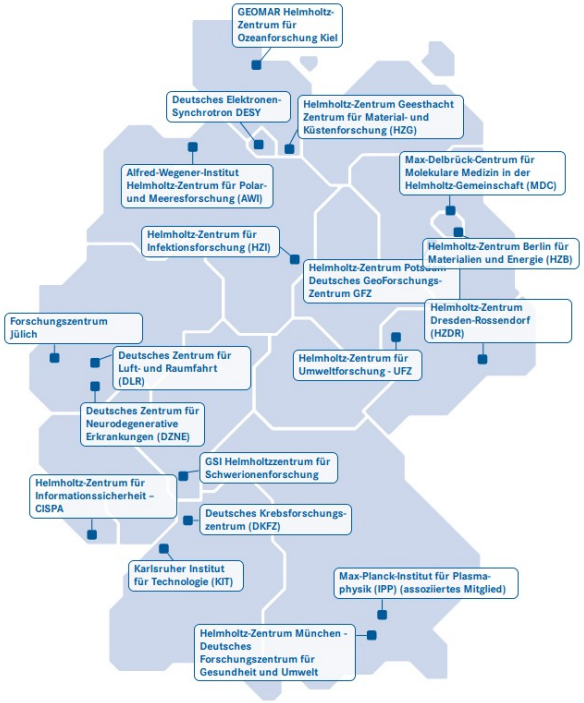
- Platforms for interdisciplinary science projects.



Use cases

- Data transfers of large data sets
- For collaborative projects in e.g. HelmholtzAI, HIP
- Asynchronous one-time transfers
- Policy implementation for synchronization of perpetually renewed data sets

- Introduction of easy to deploy Apache endpoint

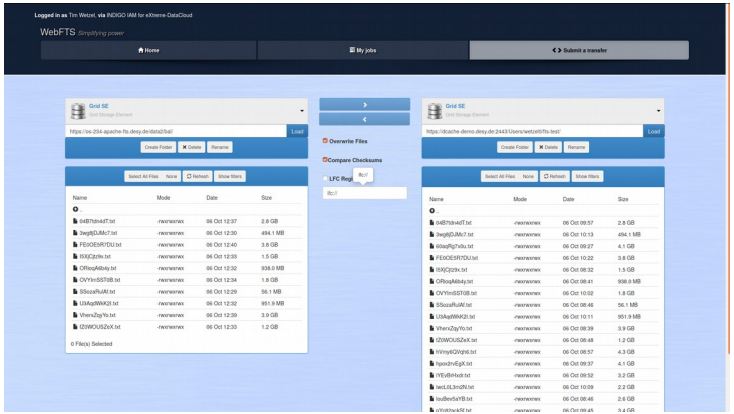


Helmholtz centres in Germany

Transfer services

- As **HIFIS** backbone core service
- CERN‘s **FTS3** as backend
- **webFTS** as comfortable WebUI
- **FTS3-REST-API** as CLI for automated transfers
- Later: **Rucio** for policy driven transfers

➔ Client applications for all needs and purposes



WLCG storage Endpoints

- storage solutions in WLCG:
  - ✓ dCache
  - ✓ EOS
  - ✓ DPM
  - ✓ StoRM
- Developed for constant high load and huge data volumes
- Enclosed view on data
- ➔ More open endpoint solution needed for **HIFIS**

HTTP Third-Party Copy (TPC)

- WLCG development for data transfers
  - ✓ Extension of the **HTTP** protocol
  - ✓ **Third party** can commission transfers between source and destination
  - ✓ Data is transferred **directly** between endpoints w/o third party
  - ✓ One endpoint needs to understand TPC-COPY extension (**active party**, WLCG)
  - ✓ The other endpoint needs to enable PUT or GET requests for files (**passive party**)
- **Asynchronous** data transfers possible

➔ Not implemented in standard Apache httpd

➔ dCache needed as active party

Endpoint components

- Apache **httpd** webserver modules used:
  - ✓ mod_ssl (SSL/TLS capabilities)
  - ✓ mod_dav (webDAV capabilities)
  - ✓ mod_auth_openidc (OpenIDConnect/OAuth2)
  - ✓ modified mpm_itk (Multiprocessing module, user mapping)
  - ✓ self-written lua script (local user mapping)
  - ✓ self-written mod_want_digest (instance digests following RFC 3230)
- → Compatible with FTS3 and accessible via OAuth2 secured webDAV
- → Transfers are possible between WLCG storage and Apache EPs
- → Direct transfers between two Apaches are WIP

Endpoint components

- mod_want_digest (github.com/wetzel-desy/mod_want_digest):
  - Developed by Tim Wetzel and Paul Millar, fragments taken from httpd‘s mod_negotiation
  - Implements instance digests in accordance with RFC 3230 (HTTP headers „**Want-Digest**“ and „**Digest**“)
  - Supports ADLER32, MD5 and SHA digests
  - Alpha version until now
    - ✘ No digest caching mechanism or on-the-fly calculation
    - Has to read file from disk for digest calculation

➔ Good first version but needs to be optimized (WIP)

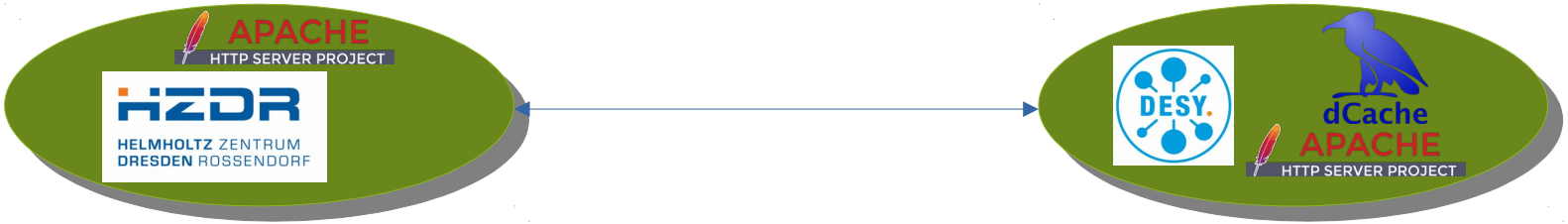
Transfer tests

- Endpoints:
  - DCache @ DESY as active party
  - Apache @ **HZDR** Dresden (OAuth2-secured webDAV EP) & @ **DESY** (+ local user mapping and instance digests) as passive parties

➔ **2344** successful transfers over 3 days, manually initiated via WebFTS

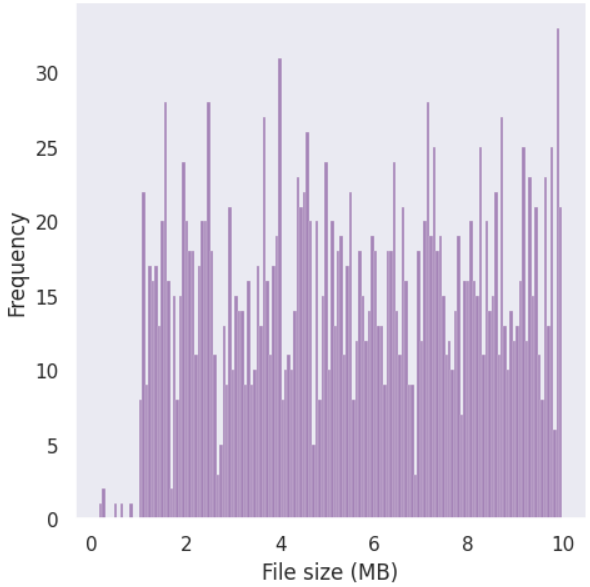
➔ **363.9 GB** of data in total

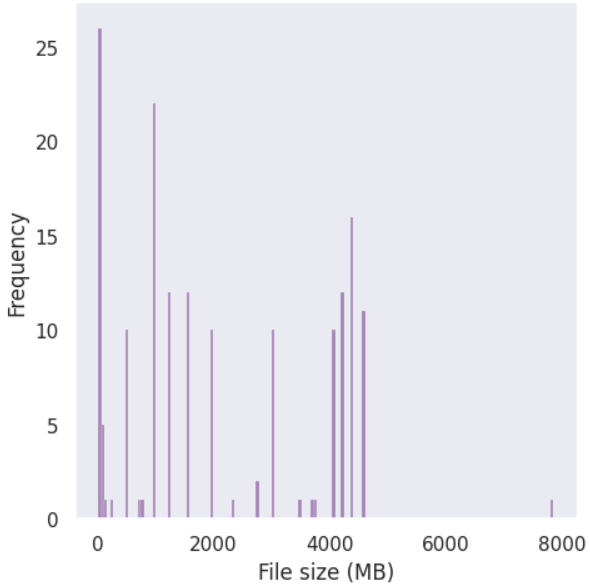
➔ Failed transfers only due to either exceeding disk space or initial misconfigurations of httpd excluded from the results, which could be fixed easily



Transfer tests

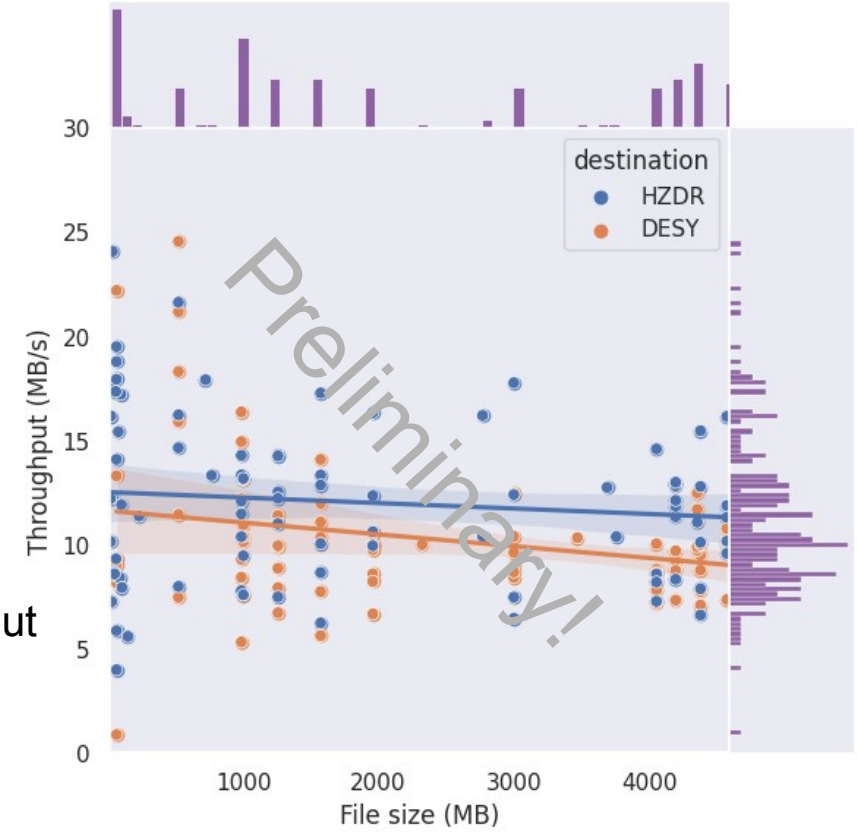
- Total transferred volume: ~363.9 GB in ~2300 files over 3 days
- 2166 < 10 MB, 167 > 10 MB





Transfer tests

- Mean throughput
  - 3.64 MB/s (<10 MB)
  - 11.2 MB/s (>10 MB)
- Standard deviation
  - 2.66 MB/s (<10 MB)
  - 3.83 MB/s (>10 MB)
- DESY instance provides lower throughput because of instance digest calculations



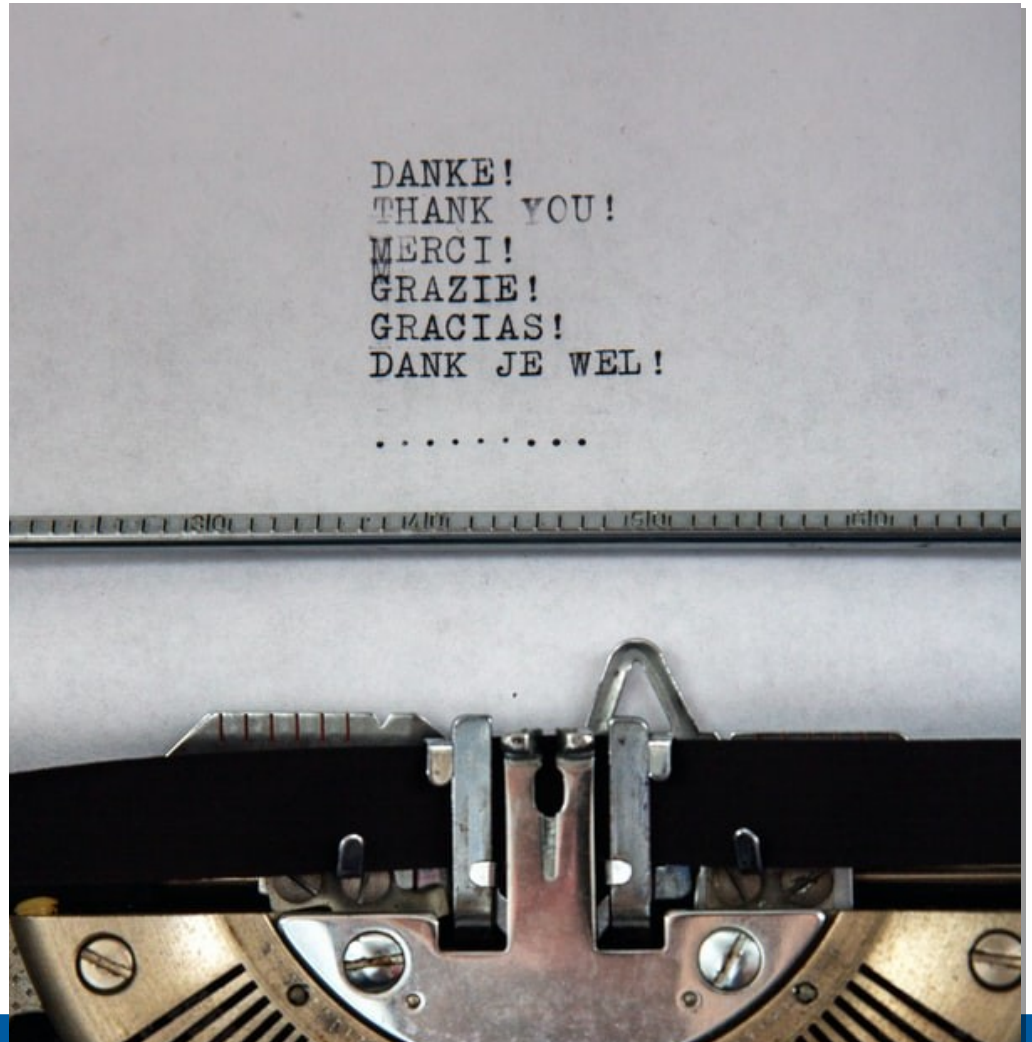
Outlook

- Investigate causes for low throughput (by all means possible)
- Integrate with HelmholtzAAI (currently not possible due to bug in unity, will be solved)
- Investigate implementation of true HTTP-TPC capabilities for Apache httpd (with KIT)
- Optimize overall setup
- Provide different packaging options (Docker, Kubernetes, OpenStack, Ansible…)
- Further testing and assessment of performance and ease of use

Summary

- Enabling data transfers between HGF centres with existing open-source software
- New software configuration for HTTP-TPC that is easy to deploy
- Together with FTS3 and WebFTS provides base for a transfer service
- Transfer tests between Apache endpoints and dCache instance successful
- Instance digest calculation and throughput still present considerable possibilites for optimization
- Currently limited to transfers involving grid storage endpoint (e.g., dCache), but future work will make Apache httpd itself capable of HTTP-TPC

➔ Practical solution for data transfers between Helmholtz centres



tim.wetzel@desy.de
www.hifis.net

Grateful acknowledgements:

Andrea Manzi, EGI
Mihai Patrascoiu, CERN