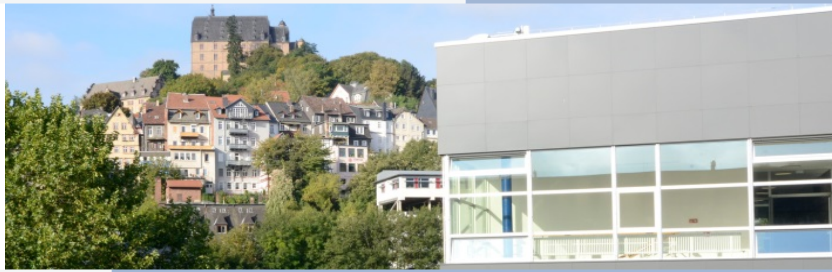
Philipps Universität Marburg

dspyce

Jörg Bieszczak, Eike Löhden



Overview

1. Who are we and what is our scenario?

2. What is *dspyce*?

3. Further developments

We are ...

- **Eike Löhden** DevOps for DSpace at the Library of Marburg University, since 2022. Angular and Java development, support for multiple DSpace repositories.

- **Jörg Bieszczak**, DevOps for RDM at IT-Department. Working on main topic „Digitization Workflow and Storage of archival material for HLA (State Archives of Hesse)“

- Philips University of Marburg is located in german federal state of Hesse. One of the oldest universities in Germany (Est. 1527).
- Close connected to several institutions in research or GLAM contexts...
- We currently host/develop 10 DSpace productive repository instances (67 to -~10 million items)

We are working on ... DSpace ;-)

landesarchiv. hessen

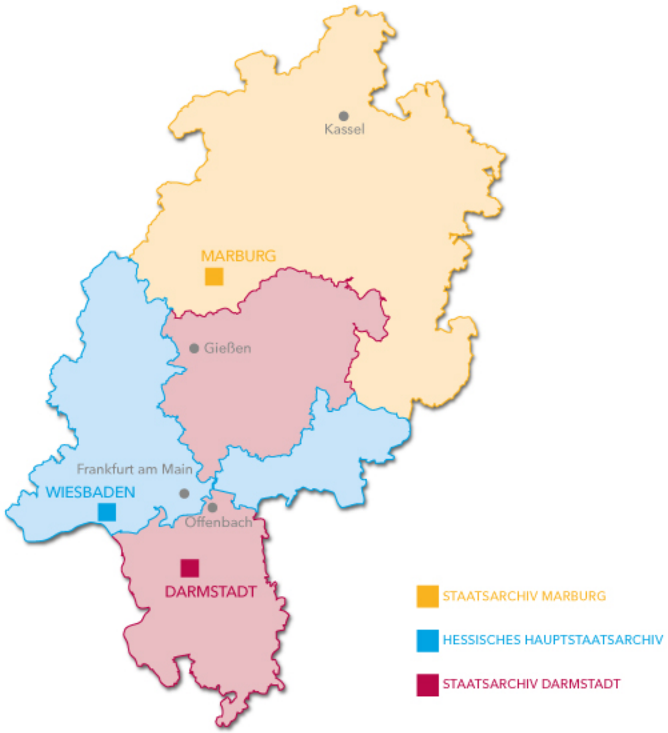
Our Scenario

Building a new Repo for the State Archives of Hesse (HLA)

- Initial import of ca. 26 Millions of master items, representing 2% of total material (175 Kilometers “on shelf”)
- 782 TB of Imagery (Dec. 2023) on CEPH-Storage
- Estimating ca. 40 Million Bitstreams for import
- A „human readable“ data structure to transform into DSpace bitstreams.
- A little bit like building Harry Seldons Foundation on Terminus ;-)

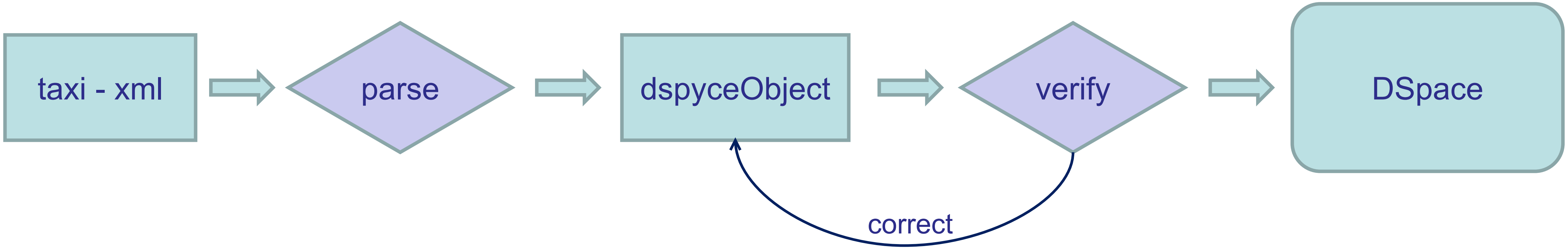
The Challenge

- Nearly 0,8 PB images of digitized material
- 24 TB of derivates (rendered JPGs, Thumbnails etc.)
- Transformation of „human readable“ data structure into DSpace bitstreams and metadata
- Using of intermediate XML-Format TAXI with BaseX
- House / Fond / Item (Archiv-Standort / Bestand / Stück bzw. Verzeichnungseinheit)
- Structure mapping exactly to the Community/Collection-Concept
- No downtime for item-Access during importing process and necessary symlinking procedure until full representation with DSpace
- Proper interaction with KITODO and ARCINSYS



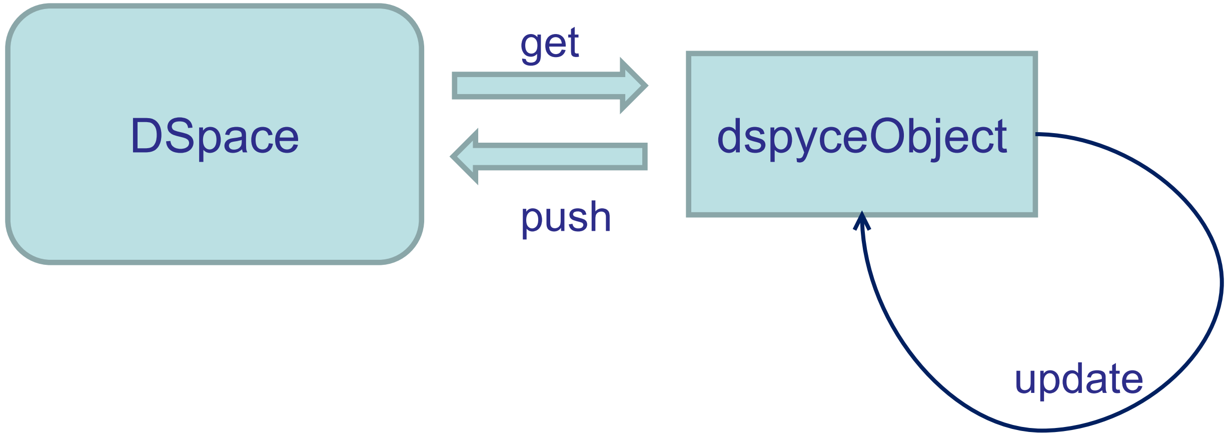
dspyce – technical overview

- We needed a consistent workflow to create and modify the data:
  - Important was not only to create the dspace objects via RestAPI, but to have a representation of DSpace-objects to verify the data before adding them to the repository



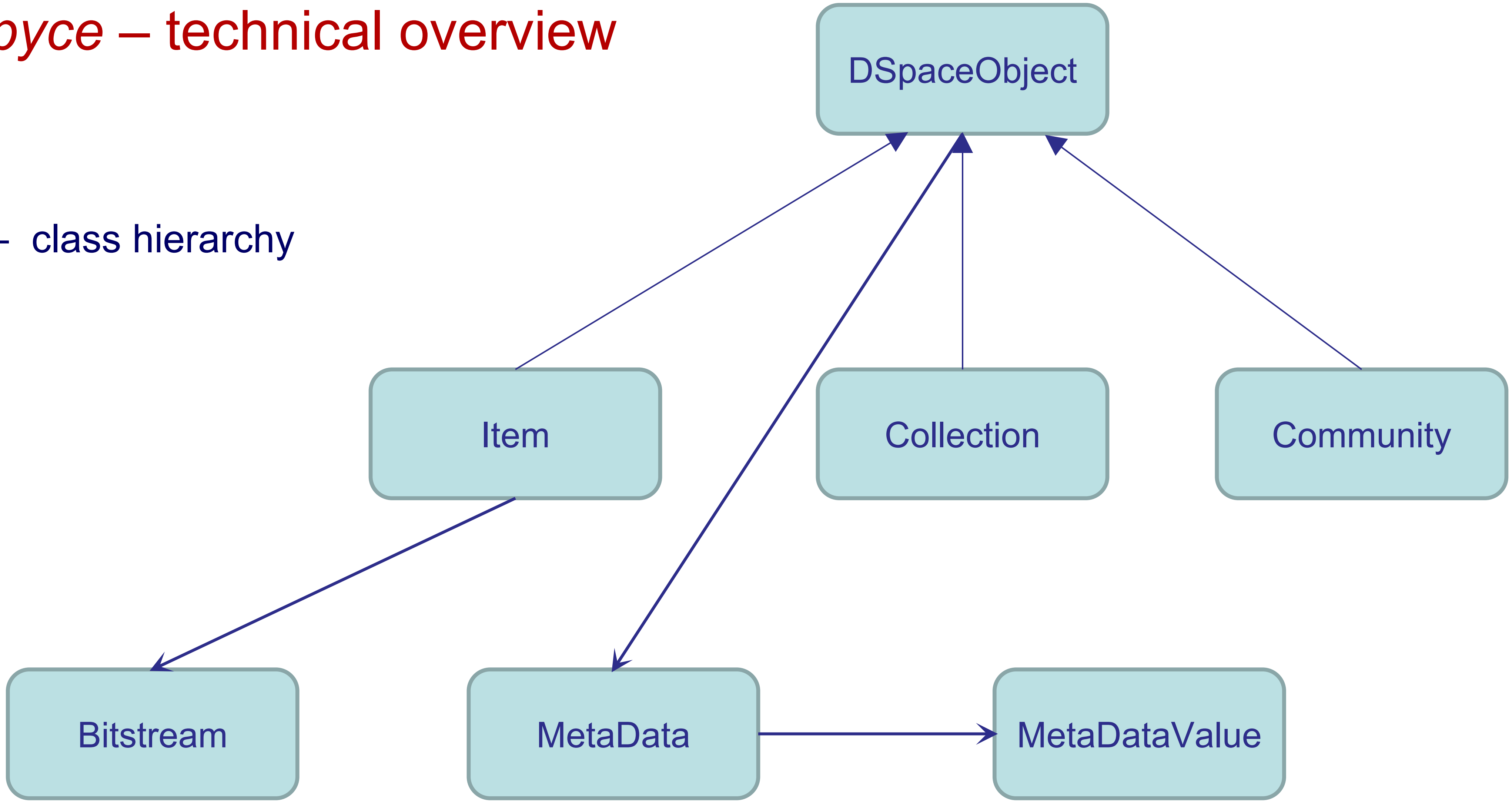
dspyce – technical overview

- We needed a consitent workflow to create and modify the data:
  - The dspyce representation helps us to modify the data without risking to break the compatibility with our repository



dspyce – technical overview

- class hierarchy



Planned developments

- Better integration for the RestAPI
- Authorization management
- Parallelisation
- Better metadata validation
- Unifying with *dspace-rest-python*

Thank you