Adoption Story:
DFT Model in Fedora Commons
Thomas Zastrow, Peter Wittenburg
■ Goal & DFT Model
■ Overview: Fedora Commons
■ Containers & Binaries
■ PID Assignment
■ Example: Dataset of Maps
■ Resolving
The Data Foundation and Terminology WG task is to describe a basic, abstract data organization model which can be used to derive a reference data terminology that can be used across communities and stakeholders to better synchronize conceptualization, to enable better understanding within and between communities and finally to stimulate tool building, such as for data services, supportive of the basic model’s use.

Definition:
A digital object (DO) is represented by a bitstream, is referenced and identified by a persistent identifier and has properties that are described by metadata.

RDA Data Foundation and Terminology - DFT: Results RFC dx.doi.org/10.15497/06825049-8CA4-40BD-BCAF-DE9F0EA2FADF
• Is DFT model just on paper?
• Can we implement it?
• Can it be used out of the box?
• Can we help people who need to setup a proper repository?

So let’s try to implement it in Fedora Commons.
Fedora Commons (FC) is a generic data repository software
  - Useful for middlesized datasets, but probably not for big data

FC comes along with basic functionality common for modern data repositories:
  - User management
  - Metadata handling
  - APIs for automatic access
  - ...

FC can be used as backend for individual and discipline specific data repository implementations
A FC based data repository can consist of two elements:
- Containers
- Binaries

Containers are used to group binary elements or other containers. With nested containers, a folder-like hierarchy can be build up.

Binaries are representing uploaded files / bitstreams.

Both containers and binaries can be assigned with metadata.
In FC version 4.x, **PIDs** are part of the metadata of an FC object

Assigning PIDs to FC objects has to be done by the user

PIDs can be assigned to containers and / or binaries
Assigning a PID to a **binary**

**Pros:**
- Checksums can be calculated and can be used for validity checks etc.

**Cons:**
- As a representation of a single file, the binary itself is often not a semantic meaningful unit – it has to be seen in its context, „standalone“ usage makes no sense
- Scalability: large datasets can contain a lot of binaries which can lead to performance issues when assigning, maintaining and resolving PIDs
Assigning a PID to a **container**

**Pros:**
- Discipline specific, semantic meaningful unit

**Cons:**
- Changes inside a container cannot be tracked by PIDs
Example: Dataset of Maps

- maps
  - world
    - africa
      - africa_globus
        - africa_globus.png
    - europe
    - americas
  - asia
  - antarctica

→ 11226/HandleTest_RDA1
→ 11226/HandleTest_RDA2
→ 11226/HandleTest_RDA3
In the example, 3 PIDs were assigned (two to containers, one to a binary)

It would be possible to assign PIDs also to any other FC object: for example, the „continent“ containers could be also addressable via PIDs
The assigned PIDs can be resolved from any Handle resolver:

- hdl.handle.net
- www.doi.org
1. The DFT model can be implemented easily in Fedora Commons

2. Fedora Commons can be downloaded in several forms from the Fedora Commons webpage or activated as Docker container:

   `docker run -p 1111:8080 -t -i thomaszastrow/fedora-commons /bin/bash`

   It should download the image automatically from docker hub and run the image - after a few seconds, point your local browser to:

   `http://localhost:1111`
Thank you.

Thomas Zastrow
thomas.zastrow@mpcdf.mpg.de

Peter Wittenburg
Peter.Wittenburg@mpi.nl