Managing and Compiling Data Dependencies for Reproducible Workflows

Marvin Hofer, Johannes Frey, Fabian Götz, Sebastian Hellmann

Reproducibility Aspects of Data Workflows

Repeatability
same team, same setup

Reproducibility
different team, same setup

Replicability
different team, different setup

?

Consume Data
access in/out data

Re-create Data
reproduce existing data

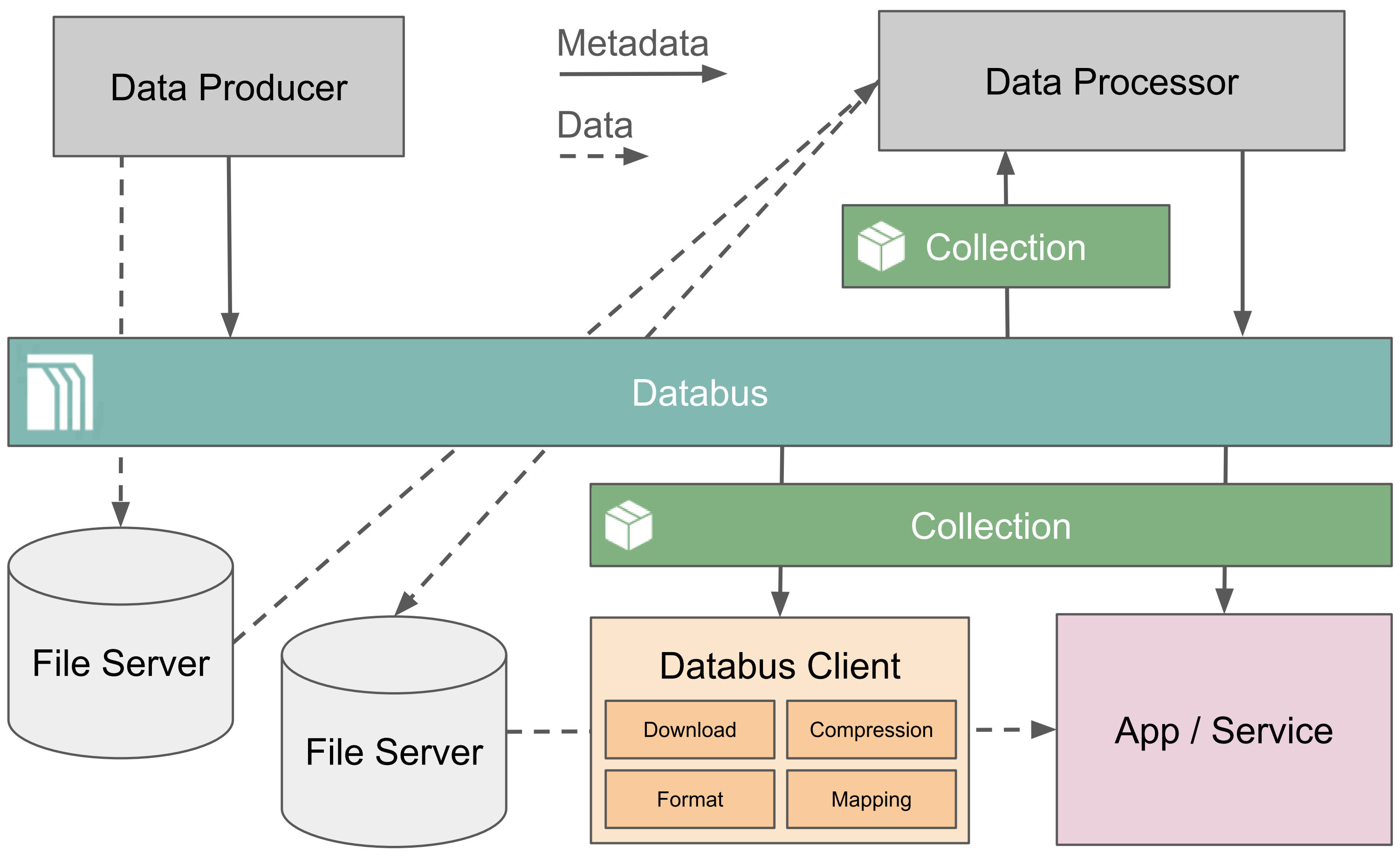
Apply New Data
same workflow but new data

Workflow Challenges w.r.t. Data

- **Size:** number of involved files per agent
- **Complexity** of data flow and life cycle
  - Versions: release frequency and forks (co-evolution)
  - Dependencies between different dataset lifecycles and multi-user
  - Phases: parallel and consecutive data processing steps
  - Debugging of data and incremental iterations.

    Results of a later phase are used to improve earlier phases

Databus is the right tool -- for many distributed users & files, complex & interrelated data life cycles, automated consumers



Databus

- RDF-based metadata registry
- Holds metadata about files:
  - Format, Compression, Size, Checksums, Access URLs, Content-variants
- Data-retrieval can be done via SPARQL-queries
- Federated SPARQL over multiple triple-stores for inter-Databus aggregation
- High focus on automatization, interoperability and extensibility

Structure is inspired by Maven (Software Dependencies)

Artifact
A logical dataset (e.g. "Wikipedia Labels", "Data About Water Turbines"). May have multiple versions and files in different formats or languages

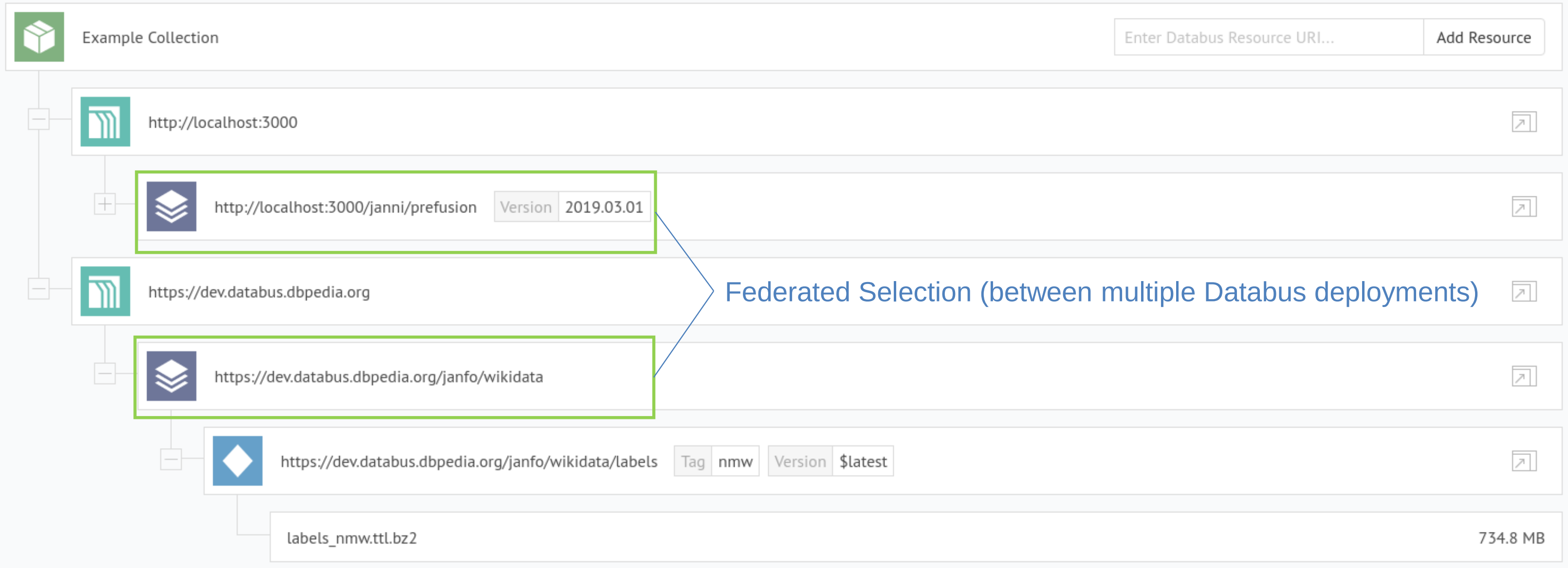
Group
Multiple Artifacts grouped together (e.g. "Mapping-based Extraction")

Version
Version of an Artifact. (e.g. "2016-10 release of All Wikipedia Labels")

DataId
Metadata document associated with exactly one **Group, Artifact and Version**

Dependency Definitions (Collections)

- The core aggregation and retrieval mechanism of a Databus deployment
- Shopping cart for data (selection over distributed data artifacts)
- Graphical editor provided with web interface

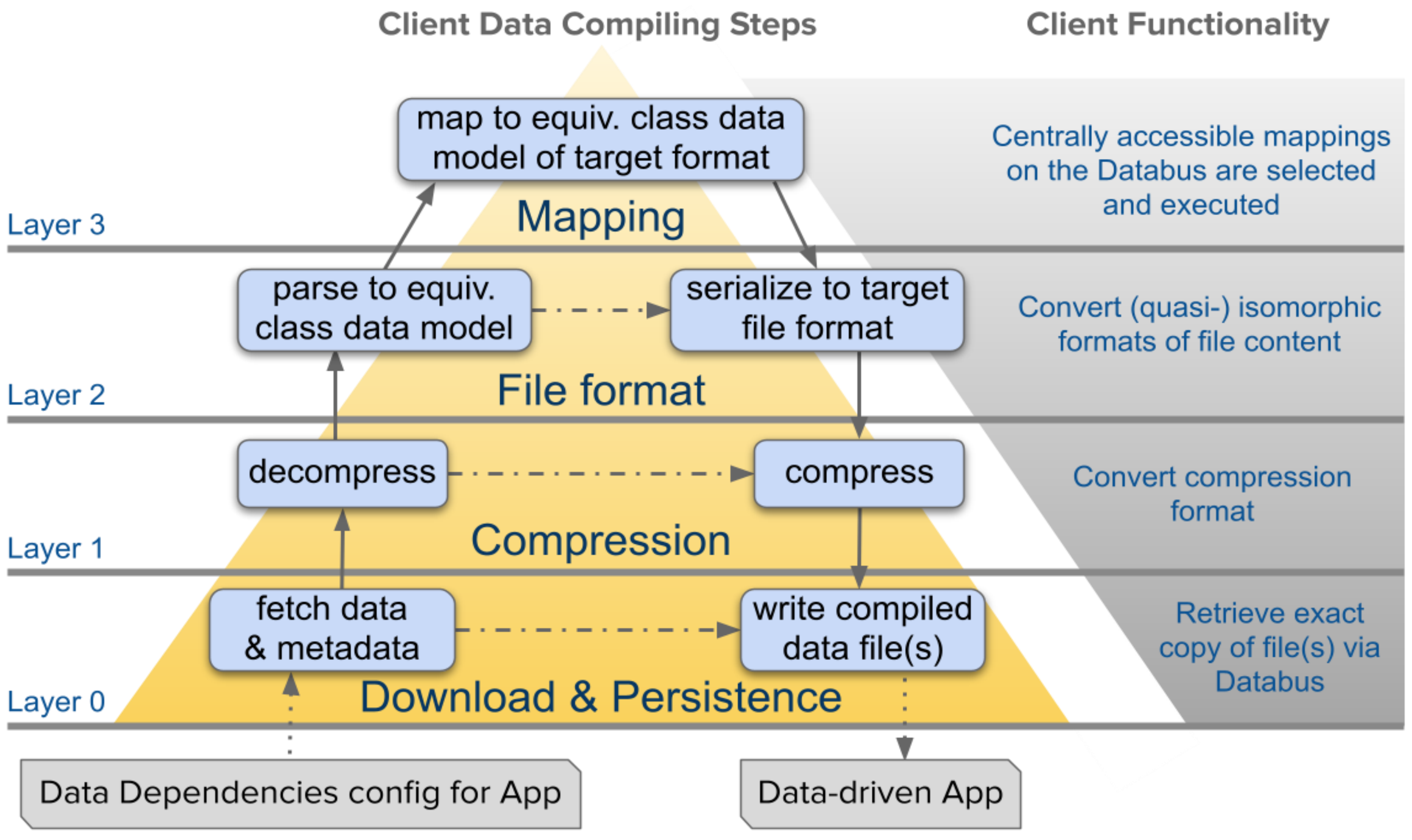

Example Collection
Enter Databus Resource URI...
Add Resource
http://localhost:3000
http://localhost:3000/janni/prefusion Version 2019.03.01
https://dev.databus.dbpedia.org
https://dev.databus.dbpedia.org/janfo/wikidata
https://dev.databus.dbpedia.org/janfo/wikidata/labels Tag nmw Version $latest
labels_nmw.ttl.bz2 734.8 MB

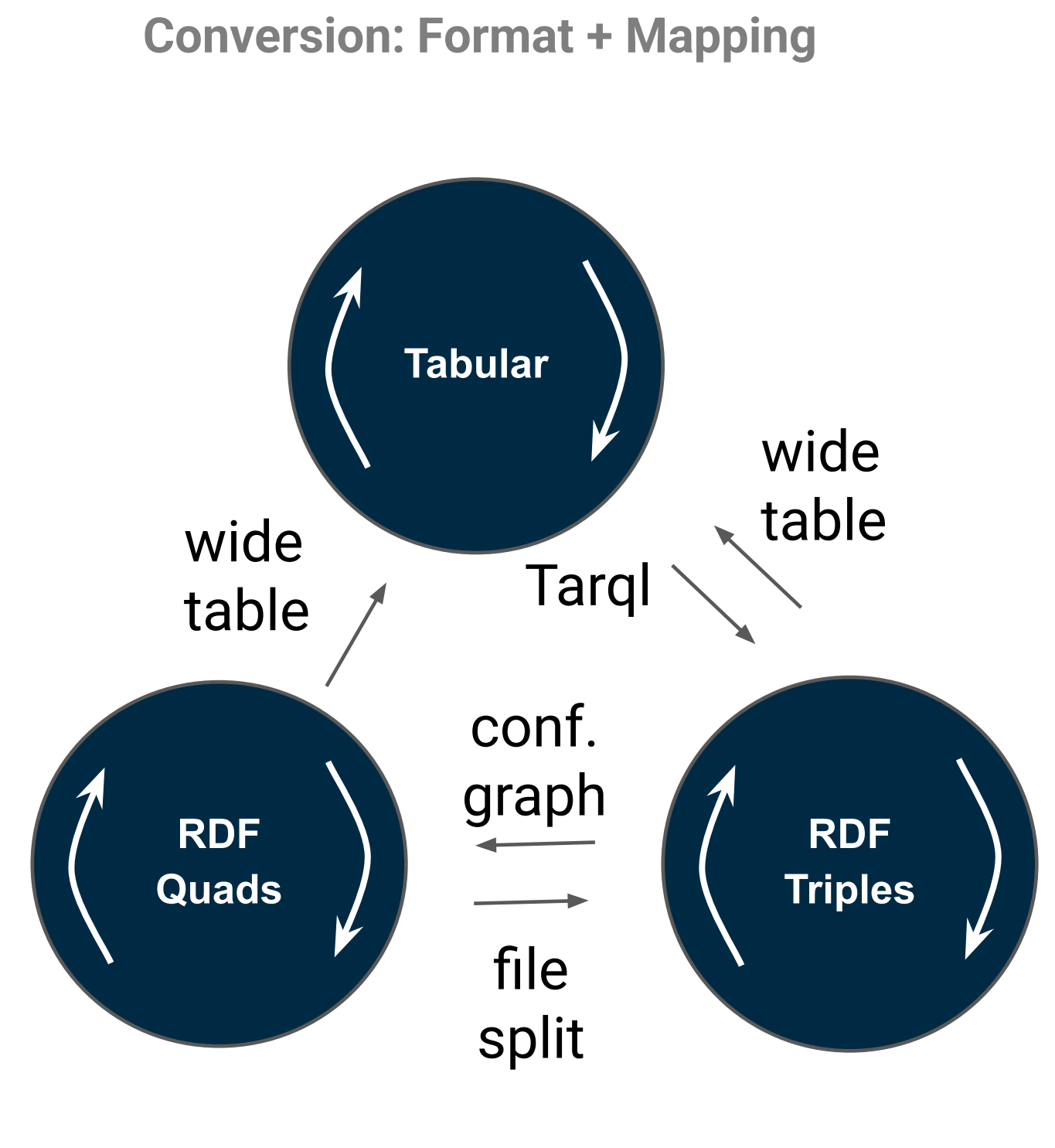
Federated Selection (between multiple Databus deployments)

Databus Client

Frey, J., Götz, F., Hofer, M., & Hellmann, S. (2021). Managing and Compiling Data Dependencies for Semantic Applications Using Databus Client. *MTSR.*

- Modular client to consume data
- Four layers that apply data compiling steps
- Layers can be used stand alone, are interchangeably (well defined interfaces) and apply data compilation steps
- Vertical-sliced implementation for RDF file formats and tabular-structured data


Client Data Compiling Steps
Client Functionality
map to equiv. class data model of target format
Mapping
Layer 3
Centrally accessible mappings on the Databus are selected and executed
parse to equiv. class data model
serialize to target file format
File format
Layer 2
Convert (quasi-) isomorphic formats of file content
decompress
compress
Compression
Layer 1
Convert compression format
fetch data & metadata
write compiled data file(s)
Download & Persistence
Layer 0
Retrieve exact copy of file(s) via Databus
Data Dependencies config for App
Data-driven App



AUTHOR
LEIPZIG UNIVERSITY
Marvin Hofer
hofer@informatik.uni-leipzig.de

SPONSORED BY

Bundesministerium für Bildung und Forschung
FKZ: 01IS18026B

Bundesministerium für Wirtschaft und Klimaschutz
FKZ: 03EI1005E & 01MD19003D

PARTNERS
UNIVERSITÄT LEIPZIG
TECHNISCHE UNIVERSITÄT DRESDEN
InfAI Institute for Applied Informatics
AKSW
LOD-GEOSS
PLASS