

Kuha2

A New Open Source DDI Aware Metadata Server

Toni Sissala and Matti Heinonen
Finnish Social Science Data Archive FSD



TIETOARKISTO
FINLANDS SAMHÄLLSVETENSKAPLIGA DATAARKIV
FINNISH SOCIAL SCIENCE DATA ARCHIVE

License: [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)



What is Kuha2?

How is Kuha2 built?

What next?



TIETOARKISTO
FINNISH SOCIAL SCIENCE
DATA ARCHIVE

What is Kuha2?



Metadata server

- > Software to serve social science metadata for harvesting
 - Ingest DDI
 - Output OSMH-JSON, DC & DDI-C
- > Support multiple harvesting protocols
- > Targeted at data archives, who
 - use DDI,
 - wish to enable metadata harvesting



Software components

- > 3 servers, client, library and database
 - Document Store –server
 - OAI-PMH Repo Handler –server
 - OSMH Repo Handler –server
 - Client – Synchronize metadata to Document Store
 - Kuha Common –library
 - MongoDB –Database



Document Store

- > Core server application - access database
- > REST API for full CRUD support
 - JSON over HTTP
- > Query API to selectively fetch records
 - JSON over HTTP
- > Import API to initially import DDI-XML
 - XML over HTTP



OAI-PMH Repo Handler

- > Server application - construct records on-the-fly
- > XML over HTTP
- > OAI-identifiers & selective harvesting
- > Metadataformats:
 - OAI-DC
 - Study subset of DDI-C (for CDC)
 - Full DDI-C



OSMH Repo Handler

- > Server application - construct records on-the-fly
- > JSON over HTTP
- > Study, Variable, Question & StudyGroup
- > Option to stream response to reduce memory consumption



Client

- > CLI application - control records in Document Store
- > Synchronize metadata from filesystem to Document Store
 - Using Document Store REST API
 - Delete absent records
- > Customize to support arbitrary metadata files



Kuha Common & MongoDB

- > Kuha Common is a Python library
- > MongoDB is a database
- > More info in "How is Kuha2 built?"

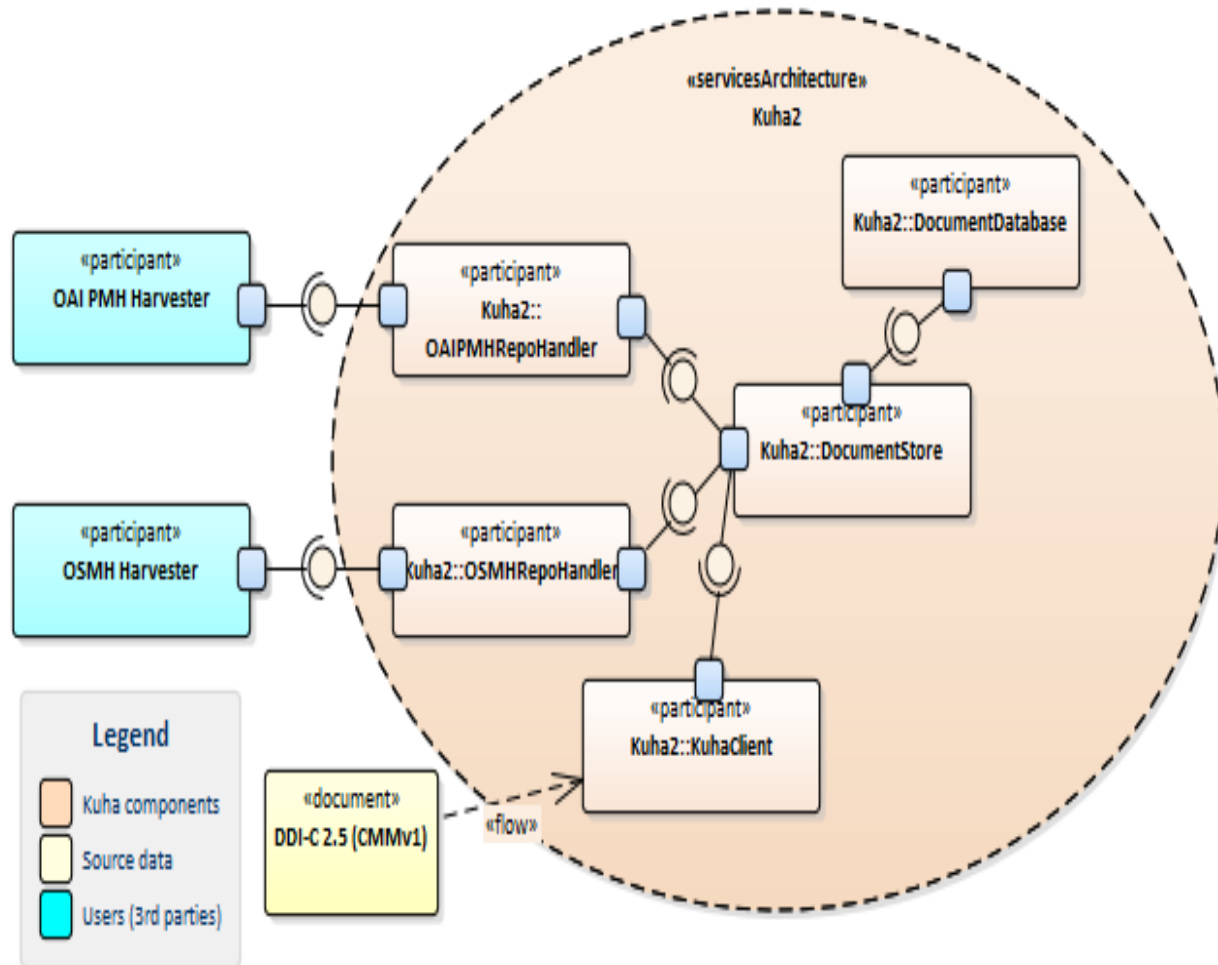


Data

- > Uses a single source for data. Serves it via multiple endpoints.
- > Data model based on CMMv1, but not limited to it.
 - Collections contain records: Study, Question, Variable, StudyGroup
 - Few mandatory fields per record – low barrier for deployment
 - Supports all fields required by CDC.

Use Case – Harvest Metadata to CDC

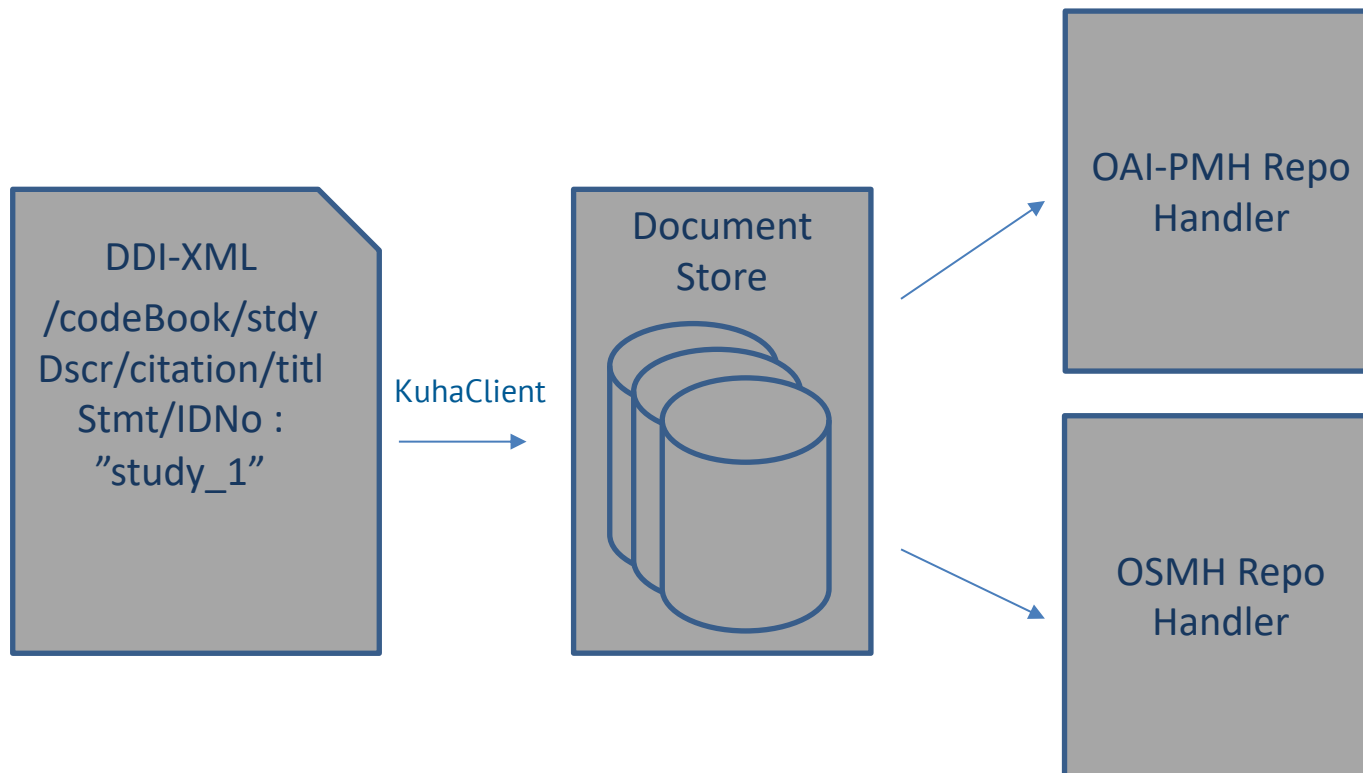
1. Synchronize a directory of DDI-files to Document Store
2. CDC Harvester requests metadata from OAI-PMH Repo Handler
3. OAI-PMH Repo Handler queries the Document Store
4. Document Store queries the Database
5. Metadata flows from Database to Document Store to OAI-PMH Repo Handler to CDC Harvester





Metadata in - metadata out

https://domain.net/oai?verb=GetRecord&metadataPrefix=ddi_c&identifier=study_1



https://domain.net/osmh/GetRecord/Study/study_1



Requirements & support

- > Python 3.5+
- > MongoDB 3.4+
- > DDI (1.2.2., 2.5., (3.1. WIP)) metadata
- > Recommended: Python3 virtualenv
- > Recommended: Ubuntu 16.04
- > <https://kuha2.readthedocs.io>



How We Got Here?

2014

2016

2017

2018

Finna

CESSDA
Harvester

SaW

CESSDA
Catalogue

- Kuha
Sept '14

- Omicrops
Jun '16

- Kuha2
Oct '17
Completely new
codebase

- FSD
Apr '18
- SASD
Oct '18



TIETOARKISTO
FINNISH SOCIAL SCIENCE
DATA ARCHIVE

How is Kuha2 built?



Architecture

- > Written entirely in Python3
- > Follows CESSDA Technical Architecture guidelines
 - Code commented throughout
 - Continuously tested in a CI platform
- > 12 Factor app methodology
- > REST Principles



Frameworks & libraries

- > Tornado web application framework
 - Minimal & efficient – fast under heavy load
- > MongoDB persistent storage
 - Schemaless NoSQL Database
 - Horizontally scalable
 - Fast queries, slow writes
 - Integration with Tornado
- > Kuha Common
 - Common functionality for Kuha applications



Extensibility

- > Document Store REST API
 - JSON via HTTP
 - CRUD operations
 - Kuha Client uses the REST API
- > Kuha Common for python development
 - Interface for data models and queries
 - Develop repo handlers
 - Develop clients



Code example: print distinct keywords

```
import sys

from tornado import ioloop

from kuha_common.document_store import Study
from kuha_common.query import (
    QueryController,
    Query
)

async def print_distinct_keywords():
    keywords = await QueryController().query_distinct(
        Study, fieldname=Study.keywords)
    print(keywords)

def main():
    Query.set_base_url('http://10.100.102.60:6001/v0')
    ioloop.IOLoop.current().run_sync(lambda: print_distinct_keywords())

if __name__ == '__main__':
    sys.exit(main())
```

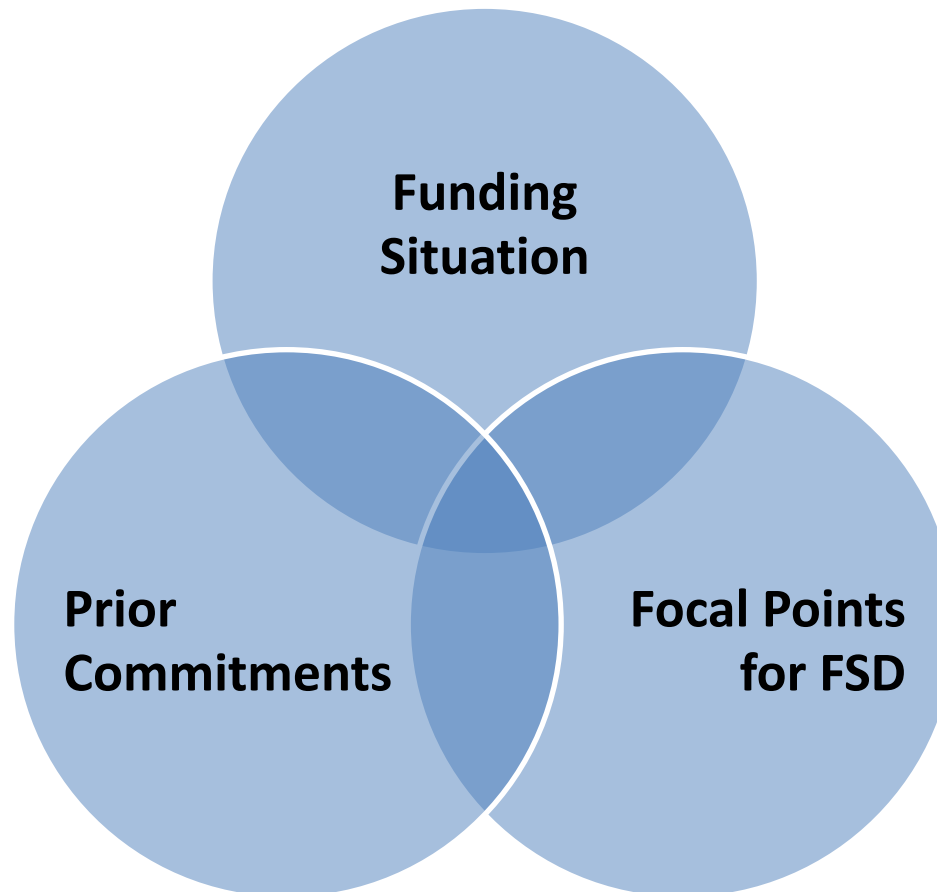


TIETOARKISTO
FINNISH SOCIAL SCIENCE
DATA ARCHIVE

What Next?



There Are Constraints





Prior Commitments

Project with DNA

- > Basic input support for DDI 3.1
 - As used by DNA
 - To the extent required by the CESSDA Catalogue
- > Target date Feb '19



Focal Points

CDC

<https://datacatalogue.CESSDA.eu>

Finna

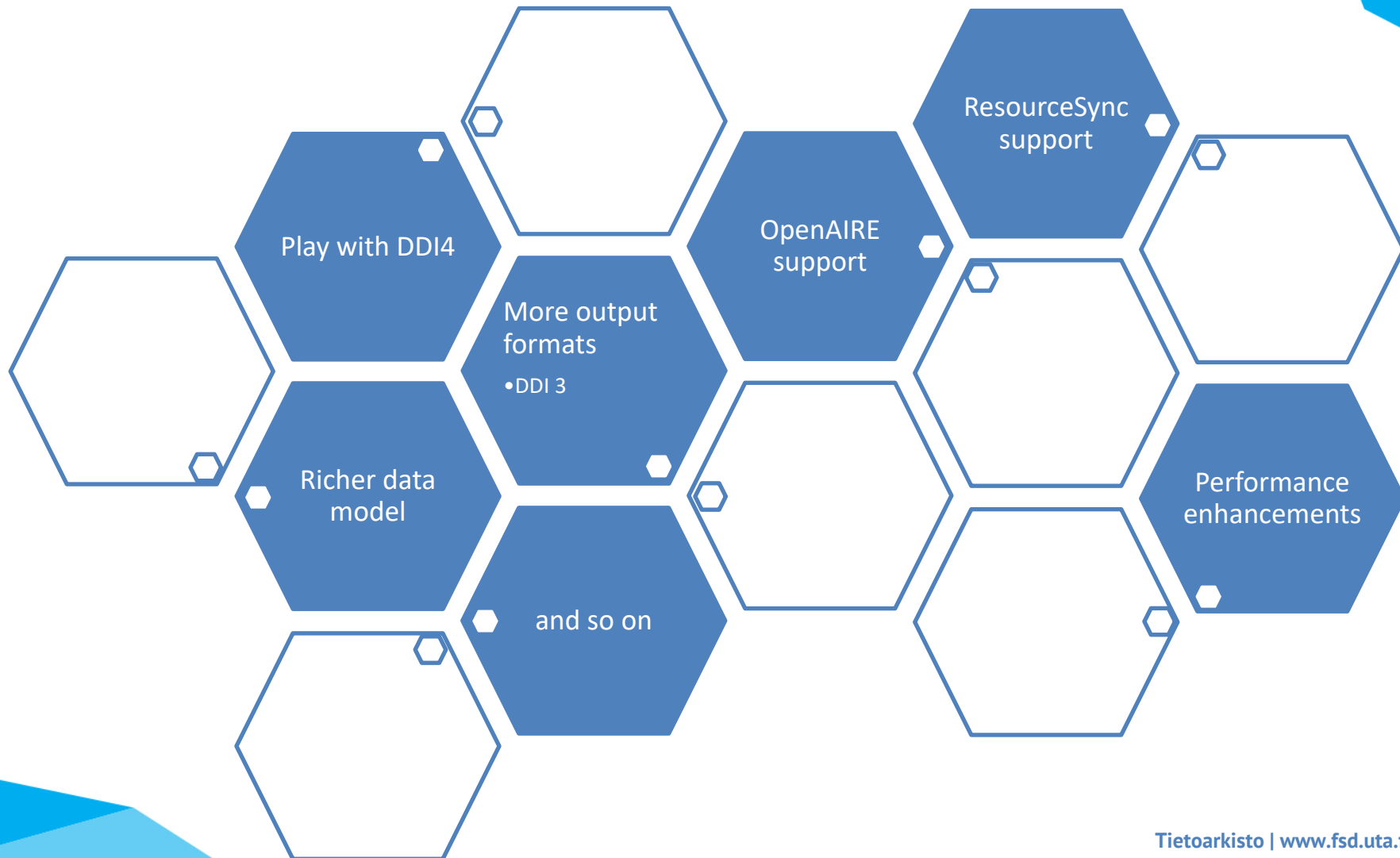
<https://www.finna.fi>

Etsin

Current:
<https://etsin.avointiede.fi/en/>
Forthcoming:
<https://etsin.fairdata.fi>



A lot could be done...





Thank You!

<https://kuha2.readthedocs.io/>

toni.sissala@uta.fi

matti.heinonen@uta.fi

Finnish Social Science Data Archive

~~University of Tampere~~ Tampere University

We are open for collaboration – let's talk!