

# Using DSpace as backend service

## Workflow-centric repository development in practice

**Ari Häyrinen**

information system expert

University of Jyväskylä

Open Science Centre

# Workflow-centric design?

**CLAIM 1: Good workflows save time and keep data solid**

**CLAIM 2: Workflows in modern repositories are cross system workflows**

```
If (claim 1 == true and claim 2 == true)
```

.then we need a **flexible** way to build workflows

**PROBLEM: Workflows are system specific**

**SOLUTION: Let's build our own workflows in top of REST APIs**

Converis

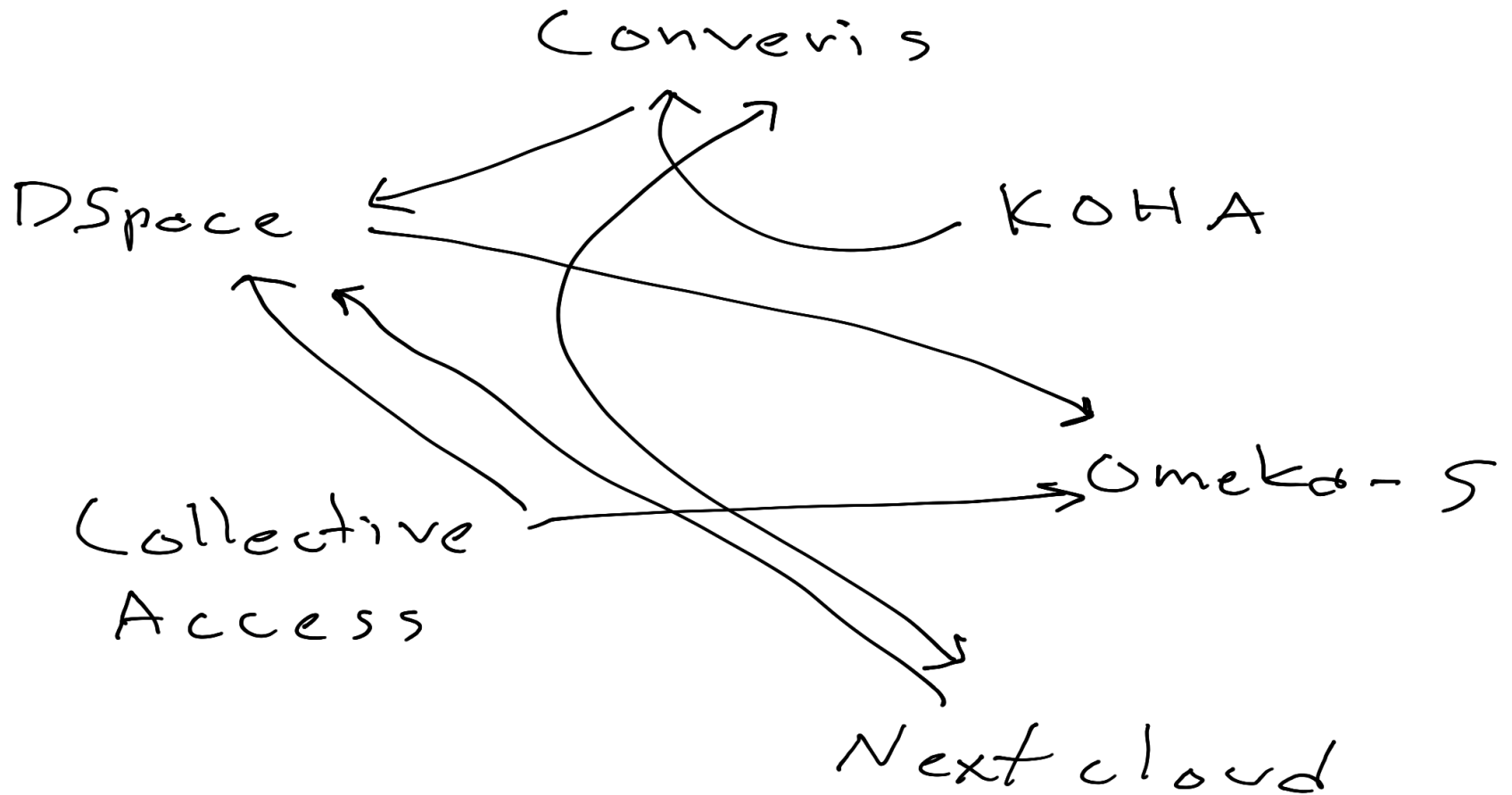
DSpace

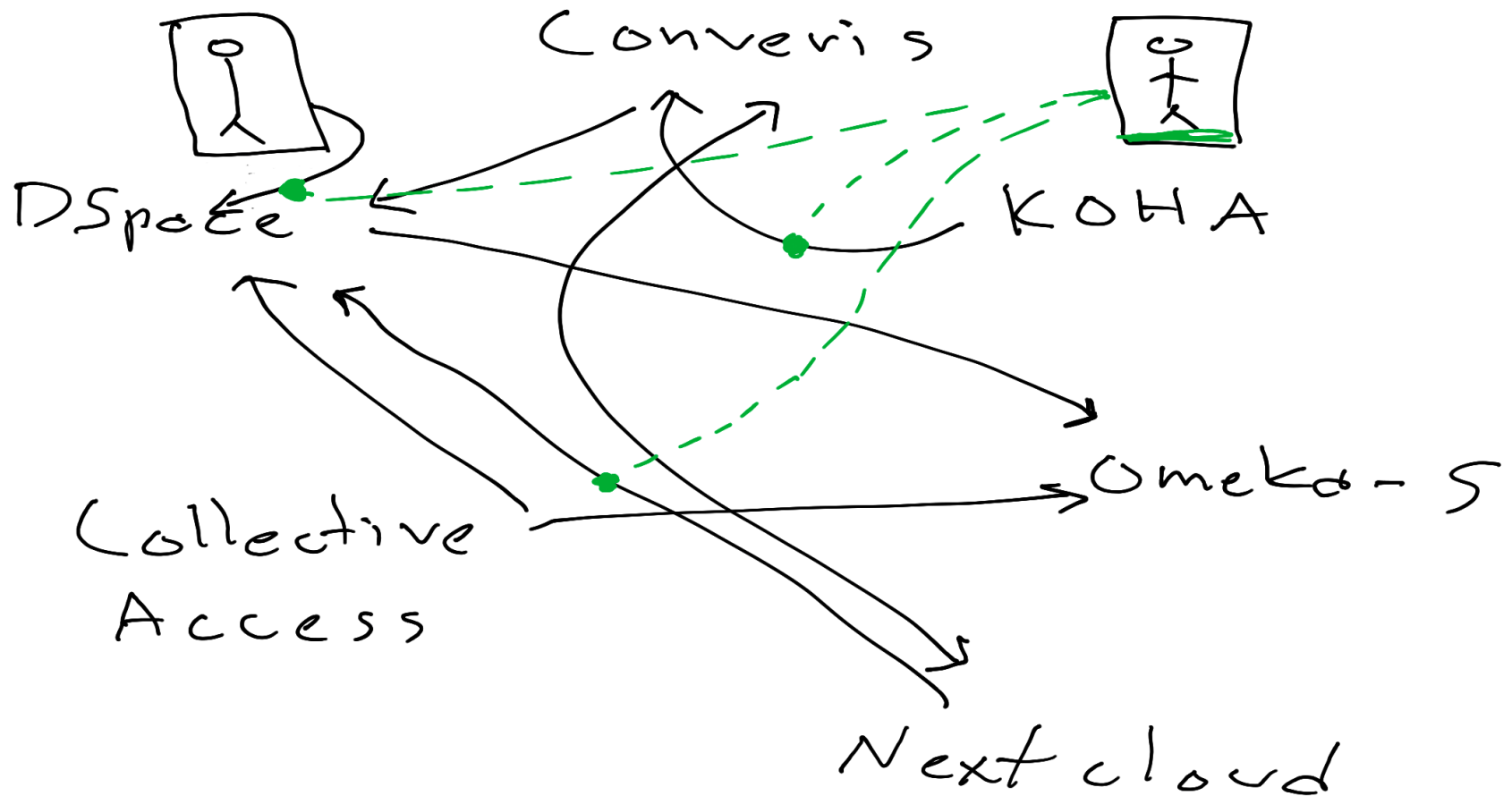
KOHA

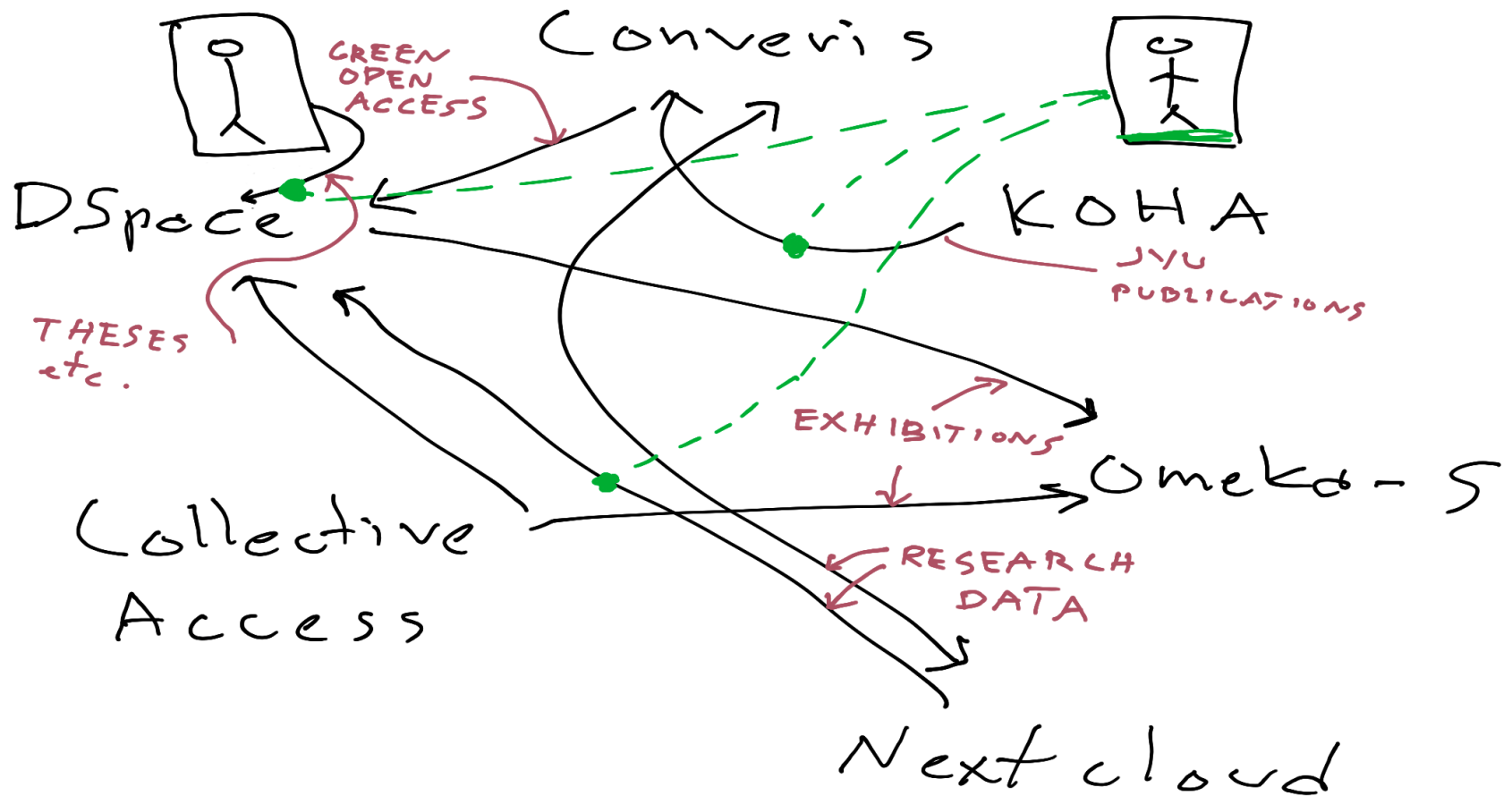
Collective  
Access

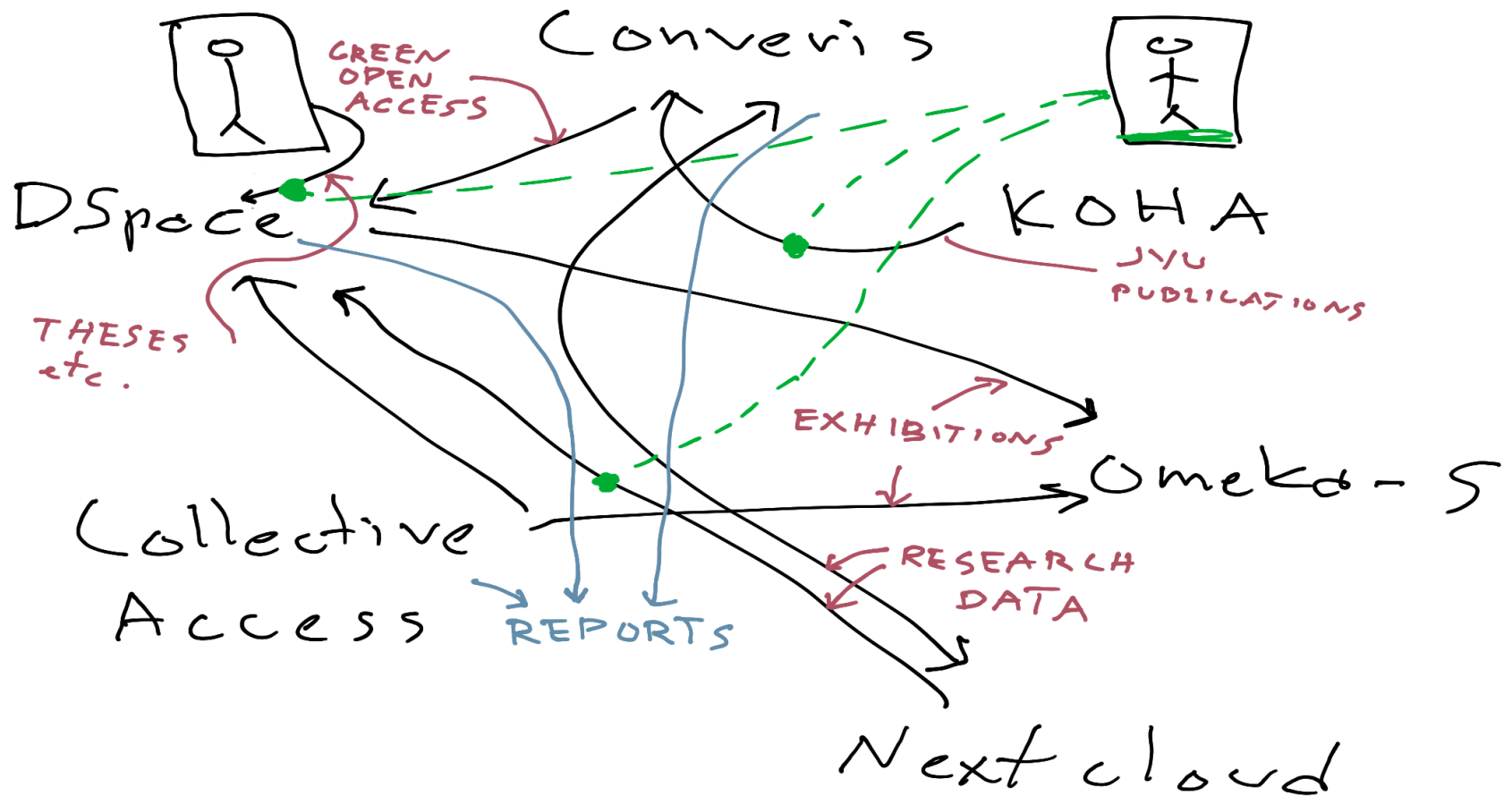
Omeka-S

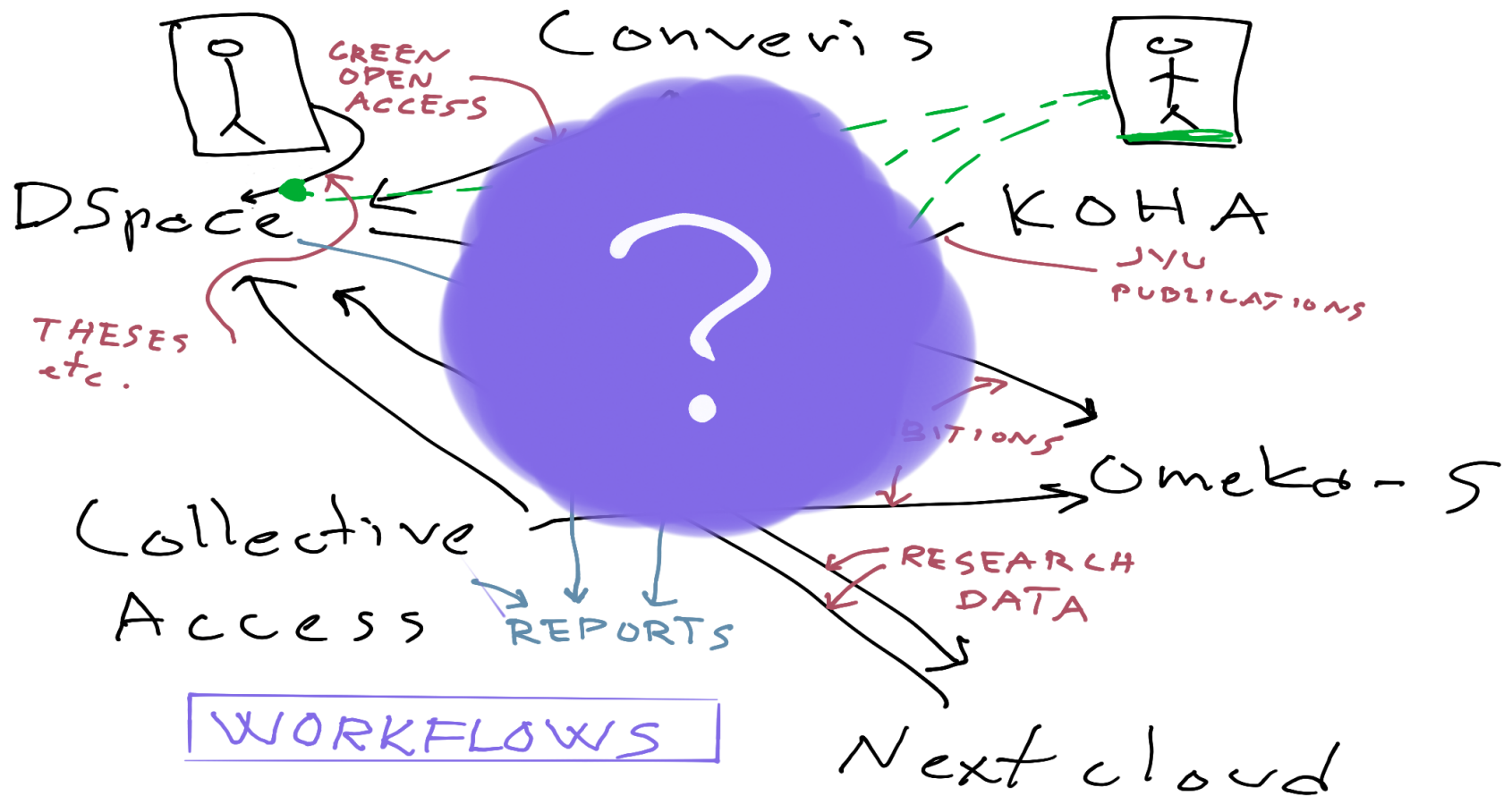
Nextcloud



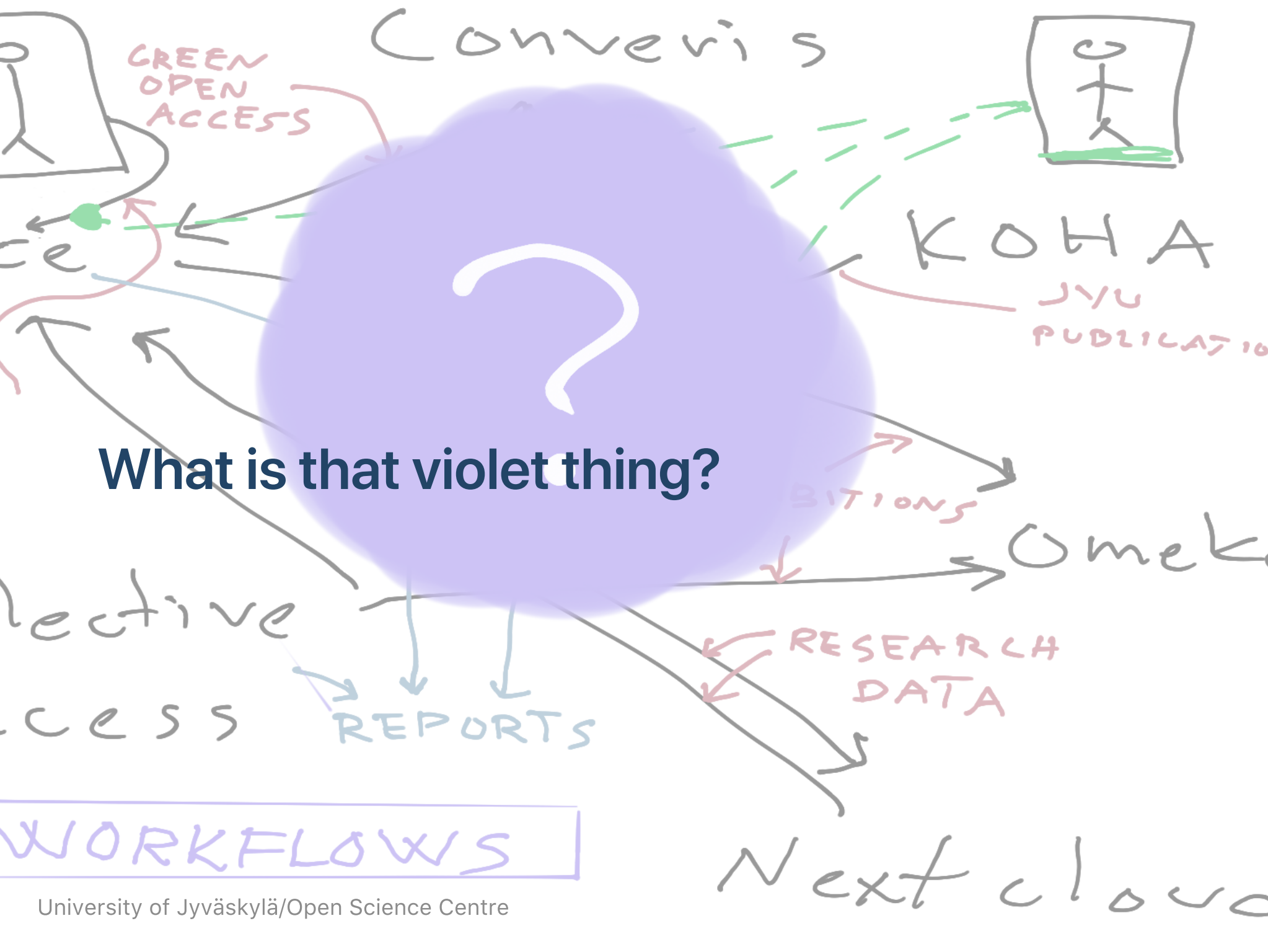












**What is that violet thing?**

# Implementation

## User interfaces are web apps (html + js)

- If possible, the webapp uses system APIs directly
- There are tasks that can not be done in browser
- Therefore there must be a **data hub** or **API for data processing**
  - read files, language detection, PDF text extraction etc.

## **GLAMpipe works as a data hub**

- Provides API for file and data processing
- Holds copy of repository data (or anything else) for fast access and analysing

# EXAMPLE WORKFLOW: Theses

- earlier users submitted directly to DSpace
  - bad metadata, very inflexible, not a very nice user interface
  - modifications REQUIRED DSpace DEVELOPER!
- in next phase students submitted to DSpace via Plone form
  - good metadata, inflexible, limited to Plone elements
  - modifications REQUIRED PLONE AND/OR DSpace DEVELOPER!
- NOW: external form and DSpace workflows are not used
  - flexible, "pretty", anything can be "bolt in"
  - modifications do not need DSpace expertise!

# Student user interface for theses workflow

1. student uploads PDF
2. text is extracted from PDF
3. language is detected (CLD)
4. keyword suggestions are generated (ANNIF)
5. students select keywords and give rest of information
6. student signs publication contract

# Administrator user interface

- list of new thesis is shown
- administrator checks the data
- accepts the work
- submitted to DSpace by GLAMpipe

# Green Open Access Workflow

- List of articles that are not in DSpace is fetched from CRIS
- Submitted to DSpace by browser

# Green Open Access Dunning

We have good Open Access percentage.

- Articles that are in CRIS but that are not self-archived are listed
- Staff tries to get final draft from the authors



# Workflow-centric design

## Pros

- really flexible
- does not require system specific developers
- original UI works as a backup

## Cons

- if there is no proper API, you have to make it yourself
- Users may need to use two separate UIs in some cases

**Thank you!**