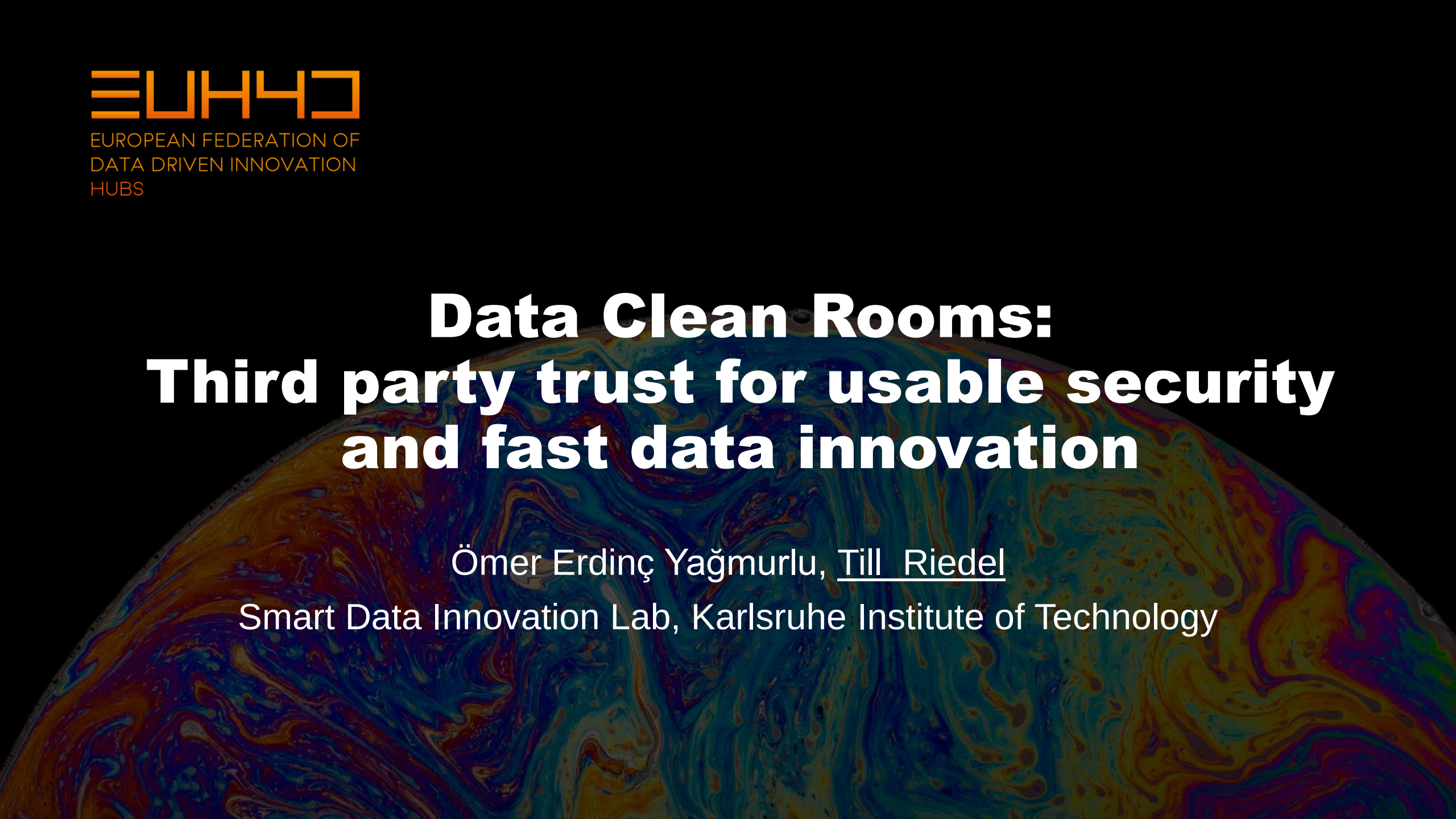
EUROPEAN FEDERATION OF DATA DRIVEN INNOVATION HUBS

Data Clean Rooms: Third party trust for usable security and fast data innovation



Ömer Erdinç Yağmurlu, Till Riedel

Smart Data Innovation Lab, Karlsruhe Institute of Technology

Can IDS technology be adapted so that usability of existing data science workflows is not significantly decreased?

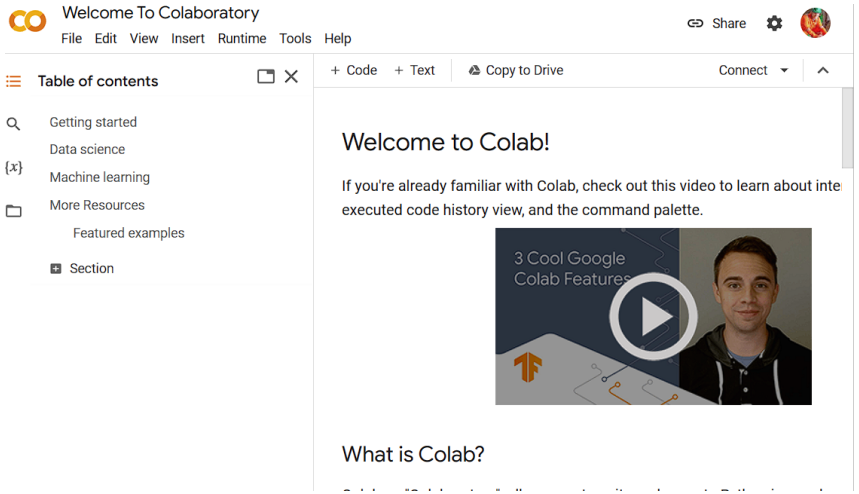
Research Question…

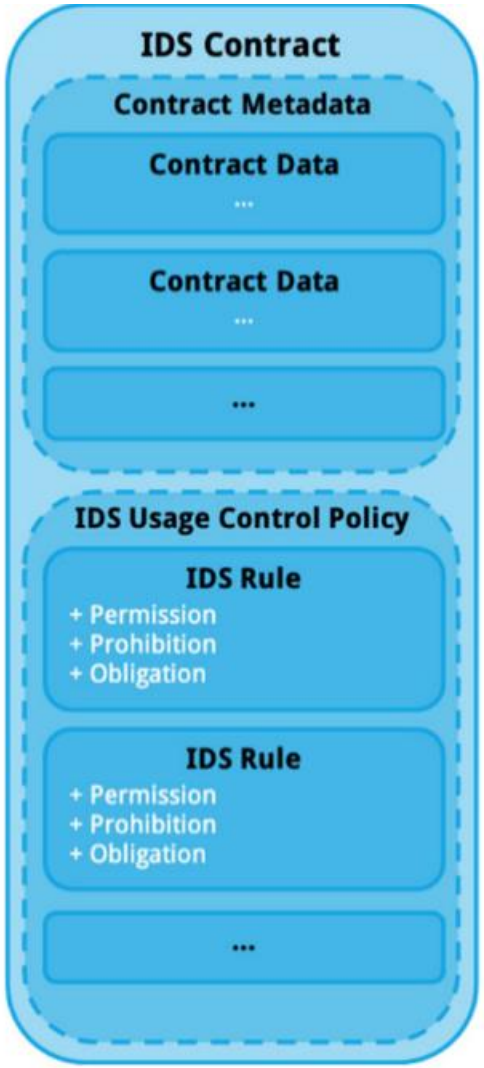
My personal Opinion: IDS lacks consumers/demand

- Infrastructure over people ??
- Learning curve to steep to just use:
  Lack of client documentation
- Focus on providers over users

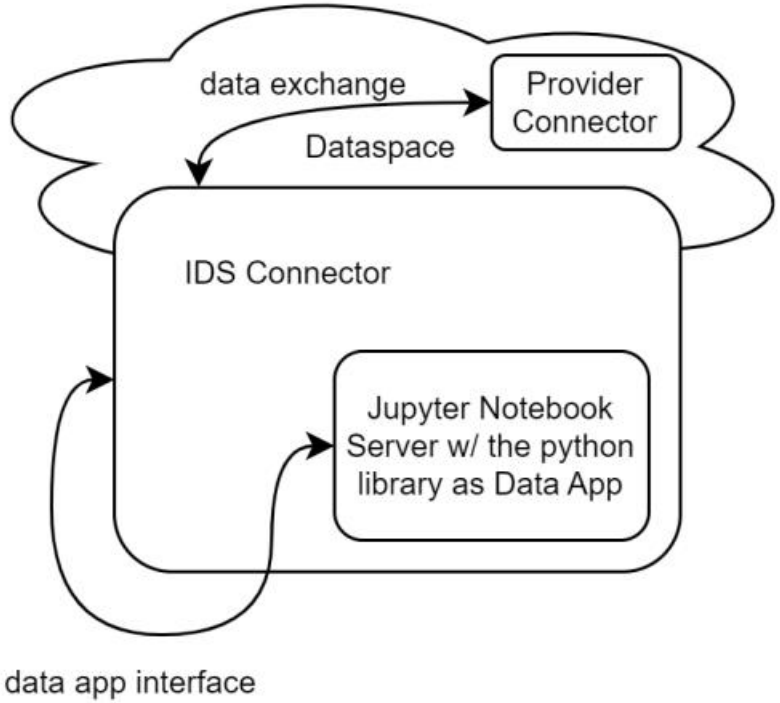
Idea: Can we do something like JupyterHub/Google Colab in

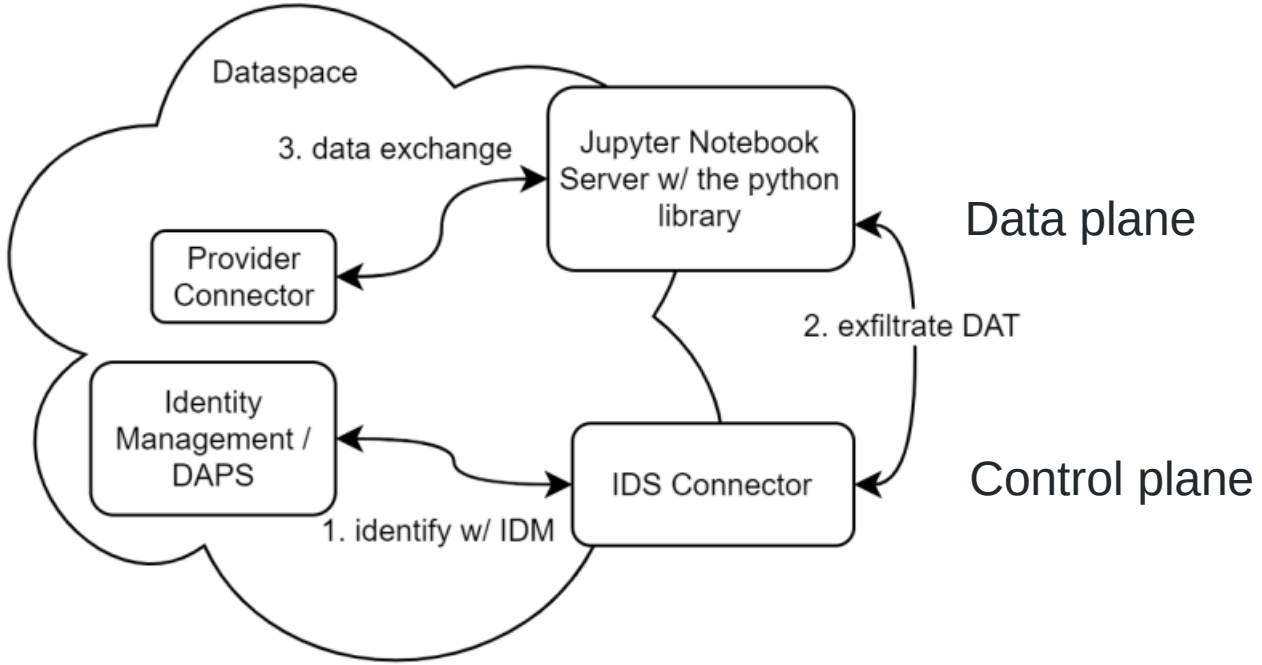
- Provide:
  - a “data cleanroom” for experimentation on private data
  - easy to use interfaces and known tools to build prototypes
  - enforce contracts IDS Usage control





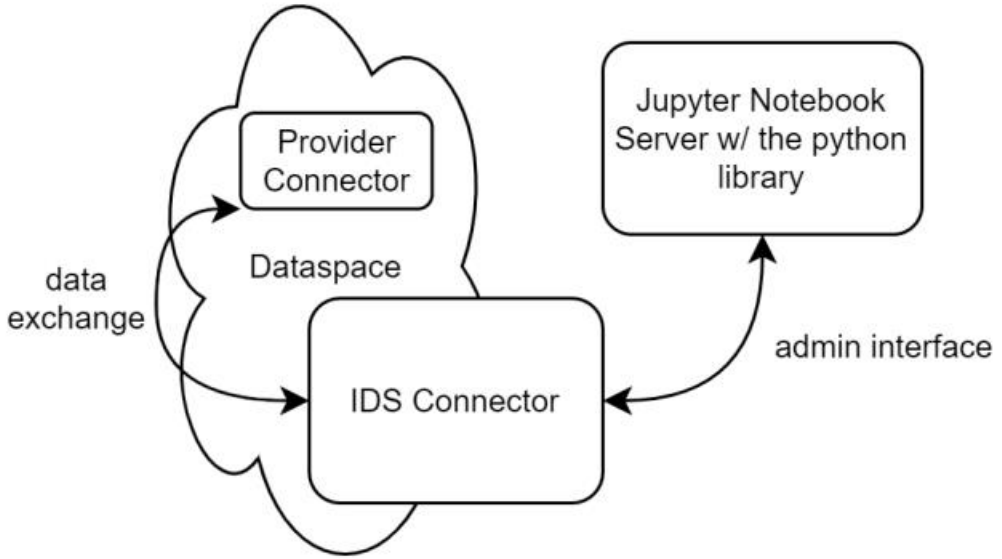
Design options we considered...





We are lame ;)

- Simple design used for user testing
- Eclipse data space components should solve the problem



Task: User Testing

- Users had only access to original documentation
- Given connector and Jupyter Remote instance for Google Colab
- “Connected” Dataset from transparency portal of the city of Karlsruhe: weather station on the roof of ”Heinrich-Hubsch-Schule” vocational school

Task 1:

- Browse the catalog for a weather fitting dataset

Task 2:

- select a contract offer
- navigate available artifact
- negotiate agreements
- download timeseries
- calculate the mean value

Usability study for an intake based IDS Connector client

Background

International Data Spaces standards enable technical, legal and organizational interoperability when sharing private data sets with across companies. Dataspace Connectors for IDS have been developed, but require considerable knowledge for set-up and integration. On the other hand most data-science research revolves around the use of lightweight technology. Data innovation labs like SDIL offer rapid turnover and facilitates collaborative research and innovation on data and offer "clean room" environments that are safe by design.

Problem statement

Precompetitive environments like data innovation labs and competitive operational environments differ fundamentally in their workflows. Porting concepts like IDS Connectors to precompetitive enviroments, as well as operating precompetitive environments with IDS technology is difficult.

Thesis idea

This thesis aims to build up a prototype platform (iSpace) that brings both concepts together and acts as a trusted infrastructure third party, that can particularly enforce securing and deleting data sets. The Data Provider thus allows the external Data Consumer to access data only on the premises of a Infrastructure Service Provider like an iSpace.

Usability study

Setting and assumptions

You are a data scientist (Data Consumer) and planning on doing an experiment using meteorological data from the City of Karlsruhe (Data Provider). City of Karlsruhe (Data Provider) publishes this data on their IDS Connector (provider connector on the right panel) as a series of csv files along with some rules that dictate it's usage policy. Data Provider also legally limits your access to the data only from within iSpace infra (Infrastructure Service Provider / e.g. SDIL / what the form on the right represents).

You are given access to iSpace infrastructure using an hosted Jupyter instance (Google Colab with a local runtime for the study). iSpace offers you (and everyone else) your own individual virtual machine that comes configured with a local IDS Connector (consumer connector on the right panel), an hosted Jupyter instance (instructions below on how to access), and SSH credentials to access the aforementioned two services via port forwarding (see the right panel.) In order to obtain the SSH credentials, you need an access token, which you obtain by registering (please don't register multiple times, store your access token and use it instead). From within their Jupyter instance, you can reach the IDS Connector of the Data Provider (right panel)

Note: For the purposes of this study, you don't need to concern yourselves with the auth mechanisms of IDS. Consumer connector is using the default admin/password combo and auth with provider is taken care of.

Methodology

In the form on the right side, register, or if you already have an access token, enter it and click 'Save'. In order to create a new VM with SSH/Jupyter/local IDS connector, press the 'New Instance' button. Then use the following command to forward the hosted Jupyter instance in VM to your localhost:

Your personal data will be removed 2 hours after the completion of the test. Your participation in the study is voluntary, you can withdraw from the study at any point during the study. By clicking the 'Register' button, you consent to the terms of this study.

Registration

If you already have an access token, skip to the next step.

Name

Register

Access Token

Access Token

Save / Refresh

User Info

Name:
Token:

New Instance

This may take ~20 seconds

IDS REST solution?

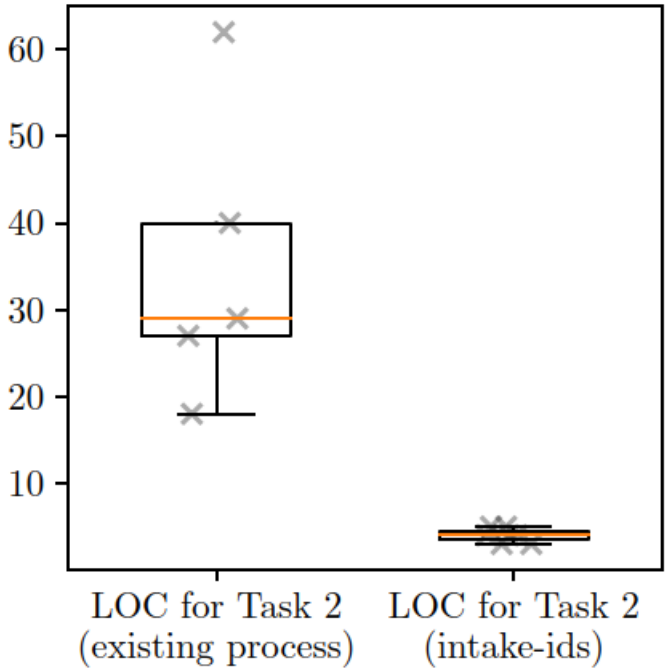
(only the downloading task from user testing: can be improved a bit using OpenAPI...)

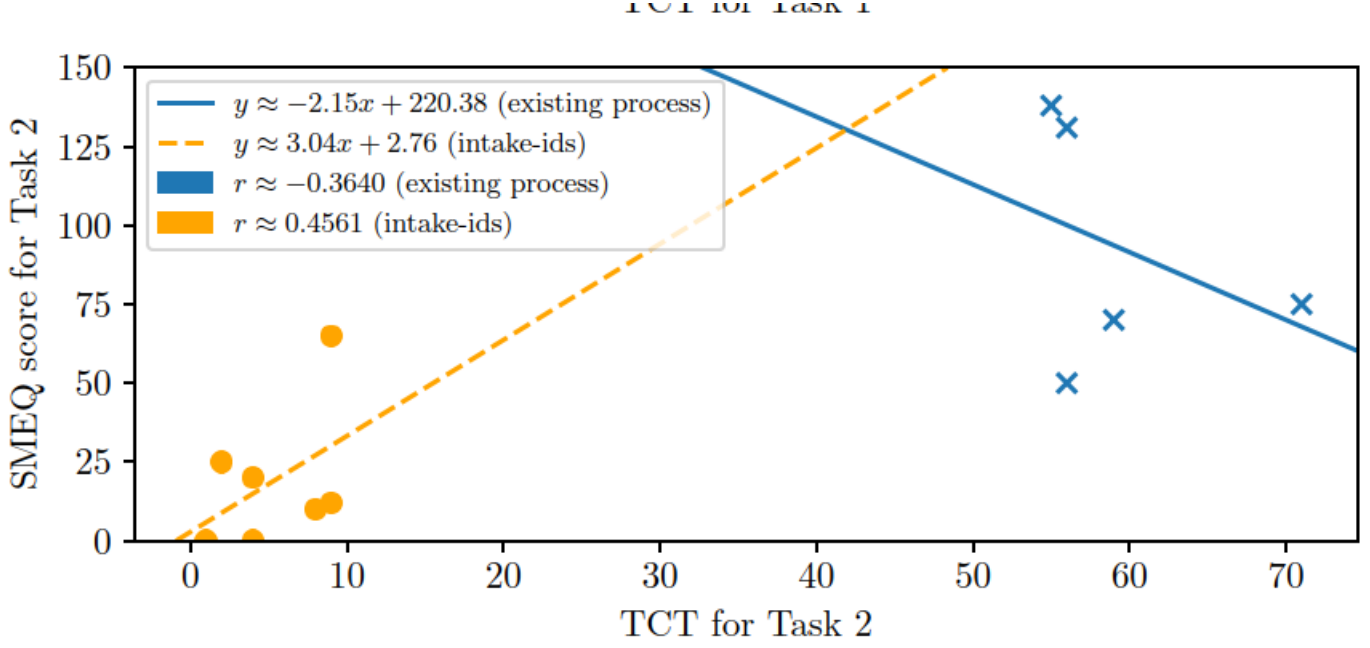
```
import pandas as pd
from io import StringIO

params = {"recipient": "https://provider0:8080/api/ids/data", 'elementId': '
    ↪ https://provider0:8080/api/catalogs/a3057f95-6245-444f-b9cd-4e4e271fbc0c'
    ↪ }
re = requests.post("https://consumer0:8080/api/ids/description", params=params,
    ↪ auth=auth, verify=False)

artifactIds = [x['@id'] for x in re.json()['ids:offeredResource'][0]['ids:
    ↪ representation'][0]['ids:instance']]
display(artifactIds)

params = {"recipient": "https://provider0:8080/api/ids/data", 'resourceIds': '
    ↪ https://provider0:8080/api/offers/ae131dba-20f6-4ff7-9ade-8f6476c83a67',
        'artifactIds': 'https://provider0:8080/api/artifacts/cdfc484c-b0fa-4e98
            ↪ -a9ed-af21e37e081a', 'download': True}
body = [ {
       "@type" : "ids:Permission",
       "@id" : "https://provider0:8080/api/rules/9f662e9d-1180-49f0-a1b5-3
           ↪ dacae93d258",
       "ids:description" : [ {
         "@value" : "Provide␣Access␣Usage␣Policy␣for␣Weather␣Data",
         "@type" : "http://www.w3.org/2001/XMLSchema#string"
       } ],
       "ids:title" : [ {
         "@value" : "Usage␣Policy␣for␣Weather␣Data",
         "@type" : "http://www.w3.org/2001/XMLSchema#string"
       } ],
       "ids:action" : [ {
         "@id" : "https://w3id.org/idsa/code/USE"
       } ],
       "ids:target": 'https://provider0:8080/api/artifacts/cdfc484c-b0fa-4e98-
           ↪ a9ed-af21e37e081a'
     } ]

links = []
for id in artifactIds:
    params['artifactIds'] = id
    body[0]['ids:target'] = id
    re2 = requests.post("https://consumer0:8080/api/ids/contract", params=params
        ↪ , json=body, auth=auth, verify=False)
    links.append(re2.json()['_links']['self']['href'] + '/artifacts')

datalinks = []
for link in links:
    re3 = requests.get(link, auth=auth, verify=False)
    datalinks.append(re3.json()['_embedded']['artifacts'][0]['_links']['data']['
        ↪ href'])

dfs = []
for datalink in datalinks:
    re4 = requests.get(datalink, auth=auth, verify=False)
    dfs.append(pd.read_csv(StringIO(re4.text)))
dataframe = pd.concat(dfs)
display("mean", dataframe["Humidity␣(%)"].mean())
```

We can do better...

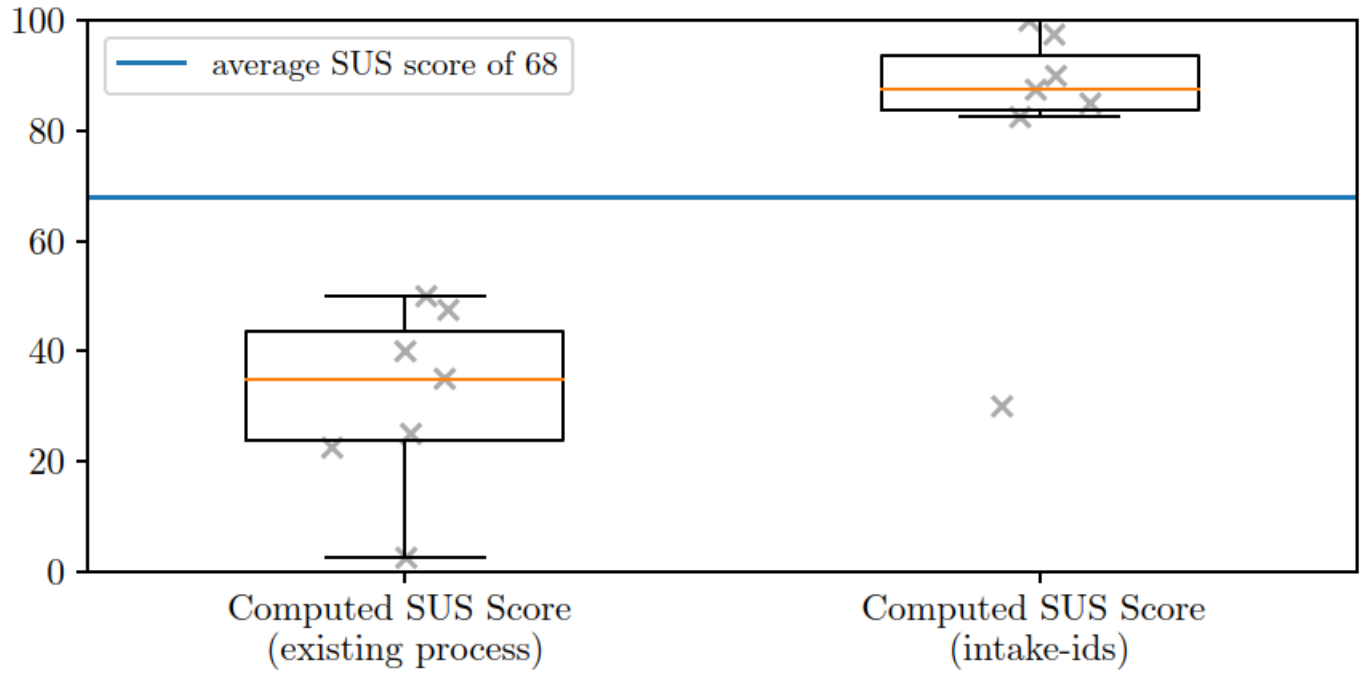
```
import pandas as pd

catalog = ConnectorCatalog(provider_url="https://provider:8080")

dataframe = pd.concat[entry.read() for entry_id, entry in
catalog.search('Wetter').items()]

display("mean", dataframe["Humidity␣(%)"].mean())
```

Task completion vs subjective mental effort





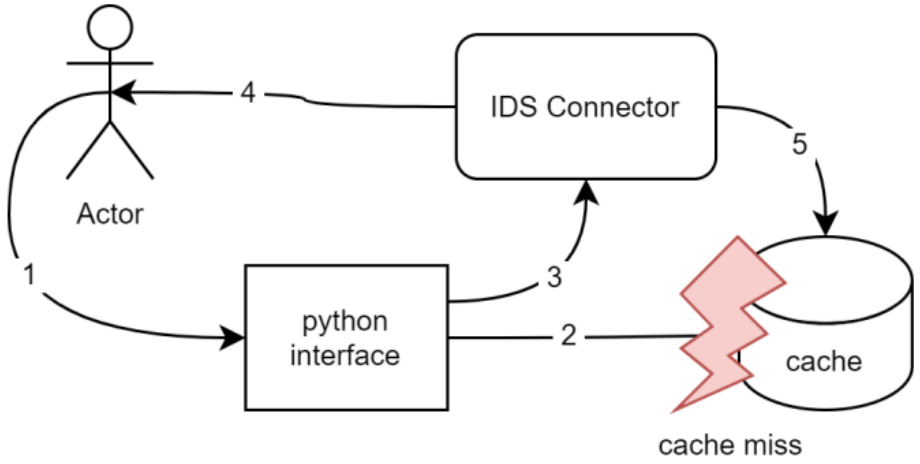
Overall system usability score

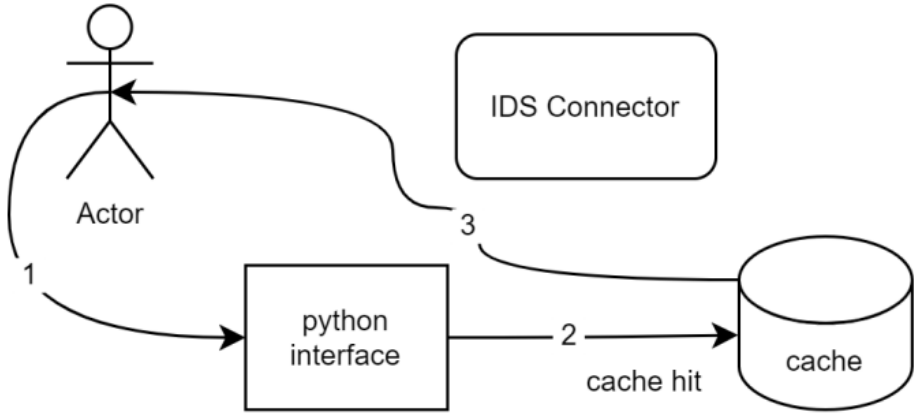


But what about security in our “clean room”

1	Allow the Usage of the Data	yes	provides data usage without any restrictions
2	Connector-restricted Data Usage	yes	allows data usage for a specific connector
3	Application-restricted Data Usage	no	
4	Interval-restricted Data Usage	yes	provides data usage within a specified time interval
5	Duration-restricted Data Usage	yes	allows data usage for a specified time period
6	Location Restricted Policy	no	
7	Perpetual Data Sale (Payment once)	no	
8	Data Rental (Payment frequently)	no	
9	Role-restricted Data Usage	no	
10	Purpose-restricted Data Usage Policy	no	
11	Event-restricted Usage Policy	no	
12	Restricted Number of Usages	yes	allows data usage for n times
13	Security Level Restricted Policy	yes	allows data access only for connectors with a specified security level
14	Use Data and Delete it After	yes	allows data usage within a specified time interval with the restriction to delete it at a specified time stamp
15	Modify Data (in Transit)	no	
16	Modify Data (in Rest)	no	
17	Local Logging	yes	allows data usage and sends logs to a specified Clearing House
18	Remote Notifications	yes	allows data usage and sends notification message
19	Attach Policy when Distribute to a Third party	no	
20	Distribute only if Encrypted	no	
21	State Restricted Policy	no	

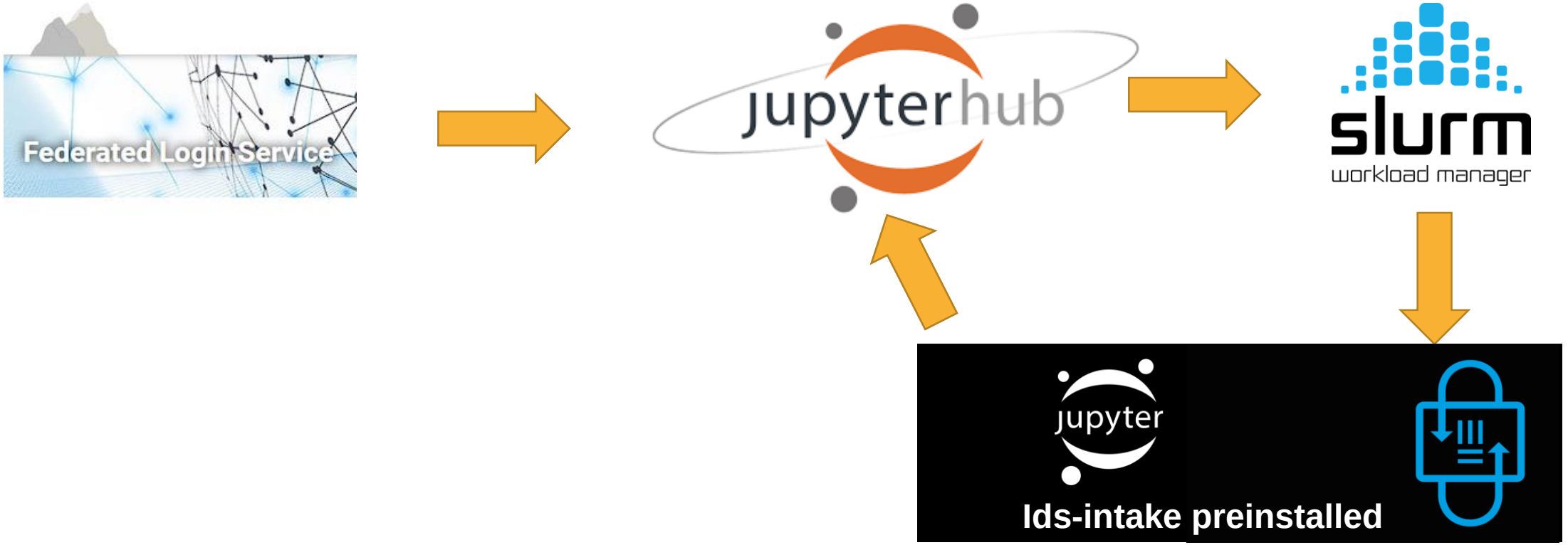
Bonus: Cache acceleration





No.	Title	Caching Strategy
1	Allow the Usage of the Data	Cache First
2	Connector-restricted Data Usage	Network Only
4	Interval-restricted Data Usage	Cache First
5	Duration-restricted Data Usage	Cache First
12	Restricted Number of Usages	Network Only
13	Security Level Restricted Policy	Network Only
14	Use Data and Delete it After	Cache First
17	Local Logging	Network Only
18	Remote Notifications	Network Only

Implementation on SDIL



Trusted Third Parties are uncool!?

- Central trust is uncool but effective as a quick start
- Intermediaries often needed to create quick trust :

- “we have a really secure system in place, but here is a download link for CSV/JSON for testing” is still reality!!

- Problem: Data space components currently provide little ways to quickly share trust and expertise via intermediaries

Data Cleanrooms create demand with usable security

- Support for fast data experimentation!
- User Centric View also on infrastructure much needed!
- Do not only focus on data set quality:
  - who creates perfectly structured or cleaned data just for others?
  - high, but impractical security might have adverse effects

→Need for usable Data Space Solutions for e.g. iSpaces

→Data cleanroom environments inside data space are a perfect fit to test ideas

https://pypi.org/project/intake-ids/

Currently lacking demand for IDS

→ please contact me (riedel@kit.edu) if you are interested:

* in joining Data Space with SDIL
* pick up active development