DATA EXCHANGE DEMO

Share data while retaining control and confidentiality of your data
Gains and difficulties of sharing confidential data

+ Access to non-public data.
+ Potential new research and collaborations.
- More work to manage confidential data.

Possible to gain new insights.
- Risks on privacy and security.
- Additional work without direct return on investments (ROI).

Gain is usually with the data requester, burden is with the data provider.
Willingness to share data

Return on Investment (ROI) is determined by the balance between effort it takes to share data, and the gain received by sharing data.

Trust is determined by the balance between the risks (due to privacy or competition), and the control (due to verification and security) of sharing and usage of data.
Type of Data Owners

Data aggregators
- Health care (Palga, NZa)
- Social-economic (CBS, municipalities)

Hospitals + medical institutions
- Hospital (AMC, vuMC, St. Antonius)
- Insurance companies (Zilveren Kruis)

Onderzoekers + universiteiten
- Universities (Twente, Wageningen, Groningen)
- Researchers

Bedrijven
- Friesland-Campina, Elsevier

Privacy sensitive

Competitive data
Example: Find the average income

Run #1
- 21 people
- Algorithm verified
- Outcome guaranteed not to be traceable to individual people

Run #2
- 22 people (same 21 and 1 other)
- Algorithm verified
- Outcome guaranteed not to be traceable to individual people

Even if individual runs are fine, combining two runs may reveal confidential data
## Different Methods to Ease Data Sharing

### Agreements
- Stipulation of what can/cannot be done
- Signing of contract or NDA
- Dispute resolution process

### Registration
- Authentication
- Verification of credential
- Reputation score
- Policy framework
- Audit trails

### Pseudonymization
- Filtering (on records)
- Pruning (on properties)
- Aggregation (combine records)
- Make coarse grained buckets
- Slight alteration of data
- One-way hashing
- One-time identifiers
- Synthetic data (mix records)

### Data Vault
- Data source retains control
- Delegate permissions
- No central data lake
- Data marketplace

### Secure Containers
- Bring algorithm to data
- At Trusted third party or at data provider
- Share output instead of data

### Secure Computing
- Secure multi-party computation
- Homomorphic encryption
- Garbled Circuits
- Zero-knowledge proof
# Data Exchange

## Vision

Realize a platform where data can easily be shared, while retaining control and confidentiality of the data.

<table>
<thead>
<tr>
<th>Target Group</th>
<th>Needs</th>
<th>Product</th>
<th>Business Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data providers with confidential data. E.g. • Companies; • Academic hospitals.</td>
<td>Data providers like to share data, while • retain control who can use the data for what purpose; • adhere to legal limitations of processing data.</td>
<td>Proof of concept (demonstration). Secure environment at trusted third party. Performs calculations on data on behalf of a researcher, with explicit consent from the data provider.</td>
<td>Facilitate open science Researchers make more use of data sources. Provide an easy-to-use and trusted solution for both parties, data providers and researchers.</td>
</tr>
</tbody>
</table>

Researchers who like to use data from other organizations for a specific purpose.

Data consumers (researchers) don’t want to be limited to public datasets.

---

Text © SURF. Licensed under Create Commons Attribution 4.0 International License

Concept of product vision board © Roman Pichler, used under Creative Commons Attribution-ShareAlike 3.0 Unported License
Collaborating without direct Sharing Data

Data Provider

Trusted Third Party

Result

Curation of result

Secure container

Data

Researchers (Algorithm Provider)

Code +Data

Result
Workflow

Share data
- Data provider shares data with trusted third party;
- Researcher shares algorithm with trusted third party;

Request
- Researcher makes request to data provider;

Verify algorithm
- Data provider verifies requester and algorithm;
- ... and selects data set(s);

Run
- Trusted third party creates secure container;
- ... mounts algorithm and data set;
- ... executes algorithm;

Curate output
- Data provider verifies output and algorithm behaviour;

Release output
- Once released, the researcher receives the output.
## Permission Models

<table>
<thead>
<tr>
<th>One-off permission</th>
<th>Trust a researcher</th>
<th>Run on a data stream</th>
</tr>
</thead>
</table>
| The data provider permits a researcher to **run** a specific algorithm **once** on a specific dataset. | The data provider permits a researcher to **run any algorithm on a specific dataset**. The permission can be revoked at any time. Example use cases:  
• the data provider trust the researcher to always write benevolent code  
• the researchers wants to tweak the algorithm, and run it on a sample dataset every time. | The data provider permits a researcher to **run a specific algorithm on any data set in a selected folder**. Every time a new dataset is added to the folder, the algorithm is automatically run. The permission can be revoked at any time, but is also automatically revoked as soon as a change to the shared algorithm is detected. |

Currently supported permission models
Implementation (Proof of Concept)

- Working prototype
- Non-production (not scalable nor fast, not rigorously tested)
- Data stored at ResearchDrive (OwnCloud implementation at SURF for researchers)
- Data sharing: [https://dataexchange.surfsara.nl/](https://dataexchange.surfsara.nl/)
  (simple password to emphasis it is a demonstration only: demo / dex)
- **Goal is to understand user requirements**
Technical Implementation of the prototype

Data & Algorithm Storage

Data provider account → Data Exchange account → Researcher account

Secure container

Data Exchange

WebDAV file copy

External integrations

Internal Components

Frontend (Sapper)

Backend (Django)

Database (PostgreSQL)

Message Queue (RabbitMQ)

Backend Listener (Django)

Tasker (Scala)

File Manager (Scala)

Data Exchange
## Risks and Mitigations

<table>
<thead>
<tr>
<th>Risk</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data is leaked to outside world</td>
<td>Researcher can never view the raw data, only the result</td>
</tr>
<tr>
<td>Data is used in other ways than intended</td>
<td>Data provider can review algorithm</td>
</tr>
<tr>
<td>Algorithm is leaked to outside world</td>
<td>Algorithm is not reviewed by data provider, researcher is trusted to write benevolent code only *</td>
</tr>
<tr>
<td>Output contains confidential information</td>
<td>Data provider curates output before releasing it to researcher</td>
</tr>
<tr>
<td>Malicious algorithm tries to copy data to remote server</td>
<td>No network access is allowed in secure container</td>
</tr>
<tr>
<td>Malicious algorithm tries to embed data in output</td>
<td>Data provider can review algorithm</td>
</tr>
<tr>
<td>Algorithm is altered after it is shared</td>
<td>Permissions involving this algorithm are automatic revoked</td>
</tr>
<tr>
<td>Researcher can no longer be trusted</td>
<td>Permission can be revoked by data provider at any time</td>
</tr>
<tr>
<td>Trusted third party can no longer be trusted</td>
<td>Sharing of data to trusted third party can be revoked at any time</td>
</tr>
<tr>
<td>Data is corrupt or data provider can no longer be trusted</td>
<td>Researcher should look for other data sources</td>
</tr>
<tr>
<td>Data can’t leave premises, not even to a trusted third party</td>
<td>Secure container can be run at premises of data provider *</td>
</tr>
</tbody>
</table>

* Not yet implemented in the prototype
14 Data is shared with the Data exchange
Algorithm is shared with the Data exchange by researcher
Researcher makes a request to the data provider
Data provider reviews request and selects dataset
18 Trusted Third Party runs algorithm on dataset
|-------------------------|-----------------------|-----------------------|

**Algorithm Owner**
freek.dijkstra@surfsara.nl

**Algorithm Name**
calculate_sum.py

**Output**
30717

**Algorithm Dependencies**
sys

**Permission Type**
one time permission

**Permission Information**
The selected algorithm will be ran on your selected dataset once.

**Algorithm Length**
Lines: 22, Words: 44, Characters: 522

**Used Dataset**
random_numbers.txt

---

**Data provider reviews output**
Researcher can see released output
Data provider can at any time withdraw permissions
Figure 7

- The Portal, Data Node, Secure Supercomputer, together with the Microdata Facilities, form the ODISSEI Data Facility.

(3) The Portal also facilitates automatic and semi-automatic data access policy management (subtask 1.3c).

Unclear data licensing or access policies are currently an obstacle in open science and the application of the FAIR principles, even for research datasets that are available as open data. ODISSEI will enrich its research data catalogue with explicit, and as detailed as possible information on licensing and access policies, preferably in a machine-readable format. The owners of each dataset will be able to provide the Portal with metadata describing what the policy for obtaining access entails. The access process varies between data providers: Statistics Netherlands requests that the user is affiliated with an authorized research institute and using their data involves formalities and costs, whereas other research data are often freely available for download to anyone around the globe. For datasets with machine-readable access policy metadata, the ODISSEI Data Node, an automated system that is closely connected to the Portal, will be able to facilitate the researcher, for example by sending data access request to the data owner, by initiating a federated authentication session, or by redirecting researchers to the landing page of the open dataset. In case a dataset does not yet have fully machine-readable access policy metadata, the ODISSEI Data Steward based at EUR will help the data owner and researcher with the access process.

Once the data owner reaches an agreement with the researchers, the owner allows the ODISSEI Data Node to transfer the data to the designated analysis environment, typically the ODISSEI Secure Supercomputer (in case of large, complex or sensitive data) or the computer of the researcher (in case of small and/or open data). The Data Node will be designed and prototyped by SURFsara (secure authentication and link to the Secure Supercomputer), DANS (owner of NARCIS), and VU Amsterdam (linked data expertise). Statistics Netherlands will make its metadata available and provide expertise on the secure data transfer connection. Design and development will happen within the first two years of the project by information scientists at VU Amsterdam and data stewards at EUR, DANS and Statistics Netherlands. DANS will then operate the Portal/Data Node.

Researchers who have created or altered data, will be encouraged to properly store them according to the ODISSEI user agreement, with the help of the FAIR support team (see the Hub).

Across this task, the team at VU Amsterdam will consist of a full-time senior scientist and a PhD student in information sciences (€ 525,000). They will be supported by a team of four at DANS and SURFsara including software developers and data stewards (€ 660,000). There also are licensing costs (€ 16,000) [Total € 1,201,000].

Related Projects

ODISSEI Secure Supercomputer (OSSC)
- In production
- Processes CBS micro-data on Cartesius
- Does pseudonymization as well

AMdEX
- Collaboration of interested parties
- Initiated by Amsterdam Economic Board
- Goal is to build an infrastructure for multiple Data Marketplaces
Partnership Questions

- Who may benefit from a data exchange?
  - Are there researchers that want to use confidential data?
  - Who are the data providers in this case?
  - Under what conditions would these data providers release their data?
- What should the role of SURF?
  - Service provider; software developer; community manager; ...
- Should SURF turn this prototype into a pilot?
- Are there other projects we should collaborate with?
Technical Questions

- Is a trusted third party the right approach?
- What is the trust relation?
  - Does the data provider trust the researcher?
  - Does the data provider trust the algorithm?
- More advances user scenarios (e.g. with 3 parties):
  - Patient trust a hospital with their data
  - Hospital trust a researcher with the patient data
  - What are the implications for the current demo with 2-part user-scenario?
  - Who gives what permissions, and is that a continuous permission? How to withdraw permissions?
COLLABORATION WITHOUT SHARING DATA

Driving innovation together

This presentation is available under the creative commons attribution 4.0 license