Tools-to-data:
report of a proof-of-concept

November 2022

Introduction

This report describes a project which explored the idea of ‘tools-to-data’. This concept allows researchers to work with data in a safe environment that prevents data leaks. Currently data providers often send a copy of the data to researchers to work with. With tools-to-data this is turned around: the researchers send their tool to the environment and receive the results. In the most strict case the researcher is not even allowed to see the data.

Tools-to-data is of great interest to the CLARIAH community. It allows researchers to work with data that are currently often unavailable to them, e.g. publications that are protected by copyright law, or datasets with privacy issues. With tools-to-data data providers can offer more service to researchers, because they can make more of their valuable collections available. **In short: tools-to-data can help improve the results of humanities research by expanding the available source material.**

The project examined the viability of the tools-to-data concept by developing a proof-of-concept. Important questions were: 1) Can researchers really work in a meaningful way with data without receiving a copy and without even seeing the data? 2) Can data providers safely entrust sensitive data to this environment, guaranteeing the safety of the data to their stakeholders? 3) What are the requirements for a useful tools-to-data environment?

The project was part of the CLARIAH program[1], running from november 2020 till november 2022. It was a collaboration of KB National Library of the Netherlands, two humanities researchers and SURF. This report describes the approach and results of the project. But first we describe the concept of tools-to-data in more detail.

[1] More about CLARIAH: https://www.clariah.nl/. The proof-of-concept was one of the ‘use cases’ of Work Package 6, Text.

Tools-to-data: what is it about?

Humanities researchers often work with data by using text and data mining techniques. They use tools (i.e. algorithms) to discover patterns in the data. This type of research typically uses the data in bulk and applies computational analysis tools (as opposed to viewing the data files one by one in a user interface like Delpher).

Usually researchers obtain a copy of the data from a data provider, e.g. the KB. They store the copy on their own laptop or server to work with. However, this has two disadvantages.

Firstly, the data providers sometimes are not allowed to provide a copy of the data due to copyright law, contracts, privacy issues etc. Researchers therefore often pragmatically choose to restrict their research to data that may be copied to them. This means that there may be large gaps in their sources. For example: linguistic research that excludes recent publications because of copyright issues.

Secondly, copying the data is cumbersome, impractical and error prone. It burdens the researcher with storing and managing the data. Moreover, it can lead to many copies of data that are hard to keep track of.

Tools-to-data turns things around by bringing the tool to the data. The data remain in a safe environment, where the tool runs as well. The researcher (and the tool) cannot copy the data and the researcher only receives the results of their tool. In the most strict variant, the researcher cannot see the data at all (‘non-consumptive use’).

Of course the safety of the data needs sufficient guarantees. These can be technical and organisational. Technically the environment needs to prevent copying of the data, e.g. by having no internet connection while the tool runs on the data. Organisationally the data provider needs to be in charge of the environment. The most important aspect is that the data provider can check the results of the tools for data leakage before making them available to the researcher.

Approach

Aims of the proof-of-concept

The proof-of-concept was built in order to demonstrate the viability of a tools-to-data-environment. This question has three perspectives:

1. researchers: can they conduct their research without receiving a copy of the data and without even seeing the data?
2. the data providers: is the safety of the data sufficiently guaranteed? Which technical and organisational measures are needed?
3. CLARIAH: can the tools-to-data-environment comply with the requirements of the CLARIAH infrastructure?

The proof-of-concept was built to explore the most strict variant of tools-to-data, where the researchers cannot see the data at all and only receive the results of their tools.

Development by SURF

The proof-of-concept was developed by SURF. They were very well positioned for this project, because they had already developed a Data Exchange prototype which had many similarities to the tools-to-data concept.

In a first effort SURF tried to further develop their prototype to meet the requirements of tools-to-data. It soon turned out that this had too many restrictions, because the prototype did not fit well in the SURF production environment. So SURF decided to combine and adapt existing components of their production environment, i.e. the SURF Research Cloud.

No effort was spent on building a dedicated user interface, because this was not necessary for the proof-of-concept. The existing interface was good enough to answer the questions of the proof-of-concept.

Two research cases

At the very beginning of the project a brainstorm meeting was held to gather high level requirements. Attendees were seven humanities researchers, several contributors from SURF, several legal and infrastructural experts from KB and a CLARIAH infrastructure expert. Based on their input the project was started.

Two of the researchers remained closely involved during the entire project. They are both used to develop their own tools by applying and adapting language models to large datasets. As “programming researchers” they represent a small but important type of users of the KB collections. We decided to focus on this type of research for the project, because it is the most challenging for the concept of tools-to-data. Once the proof-of-concept is viable for these cases, it can be extended for other researchers, e.g. by offering Jupyter notebooks or by offering some standard algorithms.

During the project two real-life research cases were tested in the proof-of-concept-environment, using a KB dataset.

- Joris van Zundert (senior researcher and developer in humanities computing, Huygens ING) and Roel Smeets (Assistant Professor of Modern Literature and Digital Culture, Radboud Universiteit). Their research focuses on characters in Dutch fiction. Their algorithm first tries to find characters in the text and then analyses their development.
- Melvin Wevers (Assistant Professor in Digital History, Universiteit van Amsterdam). His research focuses on sentiment analysis in Dutch fiction, using time series analysis.

The two research cases were of crucial importance to the project, because the proof-of-concept could be built and tested realistically.

The researchers described their requirements in much detail and fine tuned them while the project was making progress. The conversations with the researchers helped to get a better understanding of their requirements. The researchers also tested several iterations of the proof-of-concept. Their involvement made the team more flexible by jointly finding solutions to issues.

Dataset

The researchers used a dataset from the digital collections of the KB. The dataset consisted of more than 1600 digital books (in xml TEI format) from the DBNL collection[2]. Some of the works are copyright protected, but KB did have permission of the rights holders to make them available for research. This made it a realistic dataset for the proof-of-concept.

The dataset helped the project team to discover the requirements from the perspective of a data provider. The KB tested the proof-of-concept as well, focusing on the checks that are necessary in order to guarantee the safety of the data.

During the project we worked with a copy of the dataset on a SURF Research Drive. This sufficed for the proof-of-concept. In future tools-to-data may have to be integrated to the data infrastructure of data providers (like KB). Depending on the legal and contractual obligations of the data provider, it may be required to keep the data within the environment of the data provider.

CLARIAH infrastructure

CLARIAH has specified the requirements for software to fit into the CLARIAH infrastructure (CLARIAH development requirements).

During the project a CLARIAH Solutions Architect was involved to keep an eye on these requirements. He also contributed to the discussions with SURF on implementation issues.

The proof-of-concept does not yet comply with all the CLARIAH requirements, but can easily be made to do so.

[2] DBNL (Digitale Bibliotheek der Nederlandse Letteren, Digital Library of Dutch Literature) is a large digital collection of Dutch publications in the domain of literature, linguistics and cultural history, https://www.dbnl.org/

Project results

The proof-of-concept demonstrated what we hoped for. It proves that tools-to-data is a viable concept, which deserves to be developed toward a full-fledged environment.

The project delivered several results:

1. Software, implementing the proof-of-concept
2. Requirements, describing what a fully-fledged tools-to-data-environment entails
3. Workflow, describing the process of working with a tools-to-data-environment
4. Input for a new project (SANE, see below)
5. Start of a pilot project, where KB will test the environment in collaboration with Dutch publishers
6. Better understanding of the legal issues
7. Results of the two research cases. These were not part of this project and will be described by the researchers in separate publications.

1. Software

As mentioned above, SURF used several components of their existing infrastructure. The proof-of-concept is fully functioning, offering a safe environment where researchers can apply their tools to the data without having access to the data themselves. The proof-of-concept is very basic, without any dedicated user interface. This was suitable to answer the main questions of this project.

The setup of components in more detail:

- The dataset is stored on a SURF Research Drive in a directory that is only accessible to a KB account.
- Each researcher has a dedicated directory on this SURF Research Drive to upload a tool. Alternatively a researcher can specify a URL to a repository (e.g. Git) or they can use a Docker-file.
- The researcher then starts a workspace on the SURF Research Cloud environment. The dataset is made available to this workspace and the specified tool is started.
- The results of the tool are stored in a separate directory on the SURF Research Drive. The results consist of the output of the tool, supplemented by the algorithm code, log files and error messages.
- Initially only the KB has access to these results. They can check the result files for unwanted data leakage. If the results are acceptable, the KB can share the results directory with the researcher.
- The researcher can download the results and use them for further analysis.

2. Requirements

One of the main activities of the project was the analysis and specification of the requirements. We decided to describe the requirements for a fully-fledged environment for tools-to-data, because we wanted to have a good understanding of all the needs. Then we distinguished between short term and long term requirements. The proof-of-concept implemented only the short term requirements.

We deliberately took ample time to discuss the requirements. This was important for the mutual understanding of the needs from all perspectives: the researchers, the data providers and the CLARIAH infrastructure.

The requirements were described in a common work document, using the format of 'user stories'[3]. This format was really helpful in creating a common understanding. It was a living document, because the requirements kept being fine-tuned during the project while our understanding of tools-to-data grew.

The requirements are available in a separate document, which is published as an appendix to this report (in Dutch). The main themes are:

- *Safety.* The environment must guarantee the safety of the data and must prevent data leaks. The proof-of-concept implemented this by using a 'workspace' where the tool runs on the data without internet connection. The researcher cannot see or download the data in any way.
- *Selection of the data*. The researchers must be able to select which data they want to use for their research. (This was not implemented in the proof-of-concept).
- *Development and application of tools*. The researchers must be able to apply existing tools to the data, or adapt an existing tool, or develop their own tool. The development of tools is done outside the tools-to-data-environment. It requires that researchers can get information on the structure of the data (e.g. an xml format) to configure the tool. During the project it turned out that researchers always need to be able to view a few data files in order to develop and test their tool. In other words: researchers cannot be entirely 'blind' to the data, they need a few files in a 'sandbox' where they can work with the data. (This was not implemented in the proof-of-concept).
- *Providing the tool.* The researchers specify which tool they want to apply to the data. They can upload the tool, or they can specify a URL to the code in a repository (like Git) or they can use a Docker image. (Currently the proof-of-concept supports Python scripts, but this can easily be extended to other programming environments).
- *Viewing and downloading the results.* The researcher can see and download the results of the tool, after these have been checked by the data provider. This also entails log files and error messages. During the project it turned out that researchers need a work space to store intermediate results of their tool. This is important for algorithms that take a long time to run, because you do not want to restart from the beginning each time you run into an error along the way.
- *Control*. The data provider must be in control of the environment. The data provider decides who may use it, which data they may use and whether they may receive the results of their tools. We borrowed from the HathiTrust 'threat model' by assuming that researchers act in good faith and will not deliberately steal data. However, they can unwittingly create data leaks in their results, so the tools cannot be trusted[4]. In the proof-of-concept we focused on this latter approach, letting the data provider check all the result files for data leaks before releasing them to the researcher. (Other forms of control could be optional in a fully fledged tools-to-data-environment).

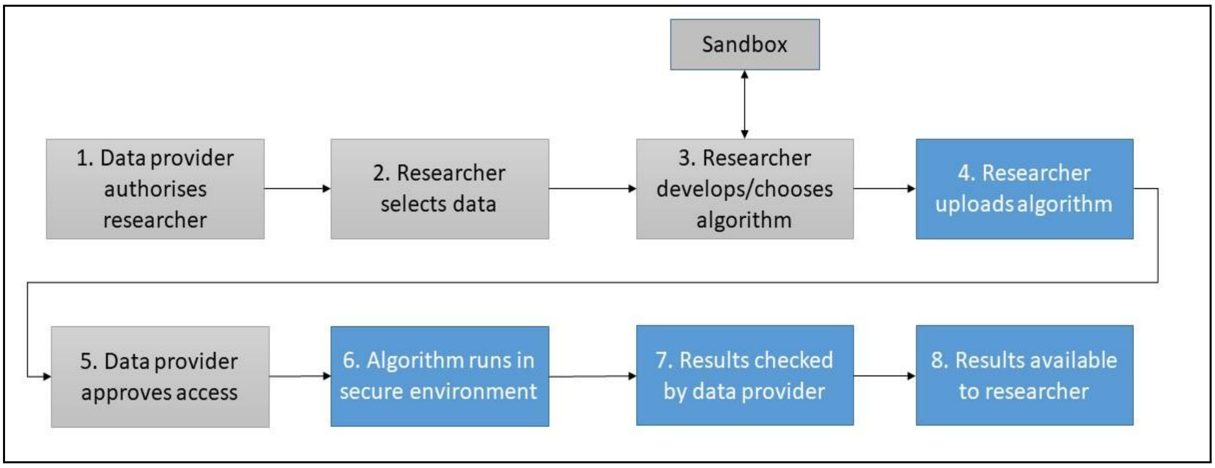
[3] A user story is a semi structured format with a short description of a required functionality, the perspective/role for which it is needed and the reason why it is needed.

[4] HathiTrust is a large digital library, see https://www.hathitrust.org/ . Their Data Capsule was an important inspiration for our work, see https://www.proquest.com/docview/2240479199 .

- *Replication.* The environment should keep a record of its usage: who used it, which data, which tool, which results? This is important to researchers for the replicability of their research. It is also important to the data provider for monitoring how the environment is used and for investigating in case of unexpected incidents. (This theme was not implemented in the proof-of-concept).

3. Workflow

A simple workflow scheme shows how the researcher and the data provider interact with the environment. The workflow also helped us explain the concept of tools-to-data to others outside the project team.


Sandbox
1. Data provider authorises researcher
2. Researcher selects data
3. Researcher develops/chooses algorithm
4. Researcher uploads algorithm
5. Data provider approves access
6. Algorithm runs in secure environment
7. Results checked by data provider
8. Results available to researcher

The blue parts of the workflow scheme are implemented in the proof-of-concept. The grey ones are not yet implemented, but they are covered in the requirements.

The eight steps are:

1. The data provider authorises the researcher to use the tools-to-data-environment.
2. The researcher selects which dataset(s) to use.
3. The researcher develops a tool or chooses to use an existing tool. This is done outside the environment, but within the safe environment the researcher has access to a few test files in order to develop and test the tool (‘sandbox’).
4. The researcher uploads the tool or specifies a link to it.
5. The data provider approves the usage of the tool (optional)
6. The tool runs on the data in a safe environment (no internet connection).
7. The data provider checks the results of the tool for data leaks (i.e. the results may contain sensitive data or information from which sensitive data may be reconstructed).
8. After this check the data provider releases the results to the researcher, who may view or download the results.

4. Input for the SANE project

An important side product of the project was its input to the SANE project[5]. This project will build a secure data environment for social sciences and humanities. The researcher can analyse sensitive data, but the data provider retains full control. SANE is a collaboration between CLARIAH, ODISSEI and SURF.

SANE will come in 2 varieties, Tinker and Blind. In Tinker SANE the researcher can see and manipulate the data in a secure environment. In Blind SANE, the researcher submits an algorithm without being able to see the data. Our tools-to-data is the precursor of Blind SANE. The requirements for Blind SANE are to a large extent based on those for tools-to-data.

5. Pilot project for publishers

An important part of the digital collections at the KB consists of publications that are copyright protected. Of course the KB cannot publish these collections or provide copies to researchers, unless the rights are cleared with the rights holders or they have given permission.

To the KB this is an important motivation for a tools-to-data-environment. It can be a major enhancement in the services that the KB offers to researchers. It allows researchers to work with data that were previously hardly accessible, while respecting the interests of the copyright holders. The KB embraces this because their mission is to make its digital collections as open as possible.

For this reason the KB works closely together with Dutch publishers. The proof-of-concept offers the opportunity to explore how their publications can be made available to researchers without the risk of leakage. (Nobody wants illegal copies of the newest ebooks to circulate on the internet!).

The concept of tools-to-data was well received by a few Dutch trade publishers in preliminary talks. The KB will start a pilot project with some of their publications, focusing on the procedures (technical and organisational) that are necessary to guarantee the safety of the data.

6. Legal questions

The concept of tools-to-data raises legal questions insofar the data are copyright protected or have privacy issues. What are the conditions for offering data in a tools-to-data-environment?

During the project we discussed this with the legal expert at the KB. There are no clear cut answers, because this is new territory.

Currently it is only possible to give access to some of the data within the ‘walls’ of the KB, so the researchers have to travel to the KB. For this reason the collaboration with the publishers

[5] Secure ANalysis Environment, https://www.surf.nl/en/news/sane-secure-data-environment-for-social-sciences-and-humanities

is very important. With permission of copyright holders the KB can offer their data in a tools-to-data-environment. Their concerns are mainly the safety of the data, rather than the location where the data live (within the KB or elsewhere).

Team

The project team consisted of:

- Steven Claeyssens (curator digital collections KB). He represented the role of data provider during the project.
- Joris van Zundert and Melvin Wevers, who represented the role of researchers.
- Freek Dijkstra (project lead Data Exchange, SURF, first part of the project), Martin Brandt (Cloud-consultant SURF, last part of the project). Development of the *proof-of-concept*.
- David de Boer (self-employed, solutions architect at CLARIAH). He contributed to the discussions on the technical approach from the perspective of the CLARIAH infrastructure.
- Lotte Wilms (digital scholarship advisor KB, first part of the project) and Marian Hellema (self-employed, advisor CLARIAH KB, second part of the project) as project lead/coordinator.

Conclusions

Tools-to-data is viable

The most important conclusion of this project is: tools-to-data is a viable and worthwhile concept.

- Researchers can do meaningful research without being able to see the data. Tools-to-data offers a way of working with datasets that would otherwise be inaccessible to them (or only accessible with many restrictions).
- Data providers are in control of who may work with the data and which data may be available to researchers. They can make more data available to research, while guaranteeing the safety of the data. Moreover,they can make data available without having to provide copies to the researchers.
- tools-to-data can easily fit in the CLARIAH-infrastructure, although the proof-of-concept does not yet implement all their software-requirements.

The need for a sandbox

Secondly we conclude that researchers need a ‘sandbox’ for the blind variant of tools-to-data. In the sandbox they can view a small testset of the data to test and debug their algorithm. An example: a dataset consists of xml files, which the tool needs to parse according to the xml schema. In the sandbox the researcher can see a few of these files when developing the tool. The sandbox is safe and the test files cannot leak out.

Procedures for checking the results

Thirdly the proof-of-concept shows that the data provider is in control of the tools-to-data-environment. An important step is checking the results of the tool before they are released to the researcher. It is not self-evident how the data provider should perform these checks. What are the criteria when looking for data leaks? How far should these checks go? A good guideline can be the estimate of the risks: assuming that the researchers will not intentionally leak data, but they can use tools that unwittingly create a leak.

We expect that these checks may be time consuming, although this will be mitigated by the small amount of researchers that will use the environment. Data providers will have to work out this task in practice.

Some of these checks might be automated in future, but we expect that it will always require some manual work.

The need for an API to the data

Fourthly we concluded that the tools-to-data-environment should have the option to use an API to connect to the data at the infrastructure of the data provider.

In the current proof-of-concept we used a copy of a dataset that we stored on a SURF Research Drive. However, this is not good enough for future use. Copying the datasets may not be legally allowed in case of copyright protected data. Moreover, it is impractical and error prone because the datasets may be quite large. And it raises synchronisation issues between the ‘master’ data and the copies.

Instead of copying the data, the data provider should have the option to keep the data in their own environment. The tools-to-data-environment may run in the same environment, or the data provider should offer an API to connect to the data. This will be further developed in the SANE project.

Addition to KB data services

Fifthly the KB considers tools-to-data an important addition to its data services. It allows the KB to make more digital collections available for research, opening up its rich collections without harming the interests of copyright holders and other parties.

KB is working on other new data services as well. It intends to develop a corpus selection tool, where researchers can select in detail which data they want to use with an easy to use interface. This may fill in the requirements for Selection that were specified in this project.

Tools-to-data may be expanded to other KB collections, e.g. contemporary ebook publications or web-archives. Some of the collections will not need the most strict protection of the ‘blind’ variant of tools-to-data. This will be further developed In the SANE project (Tinker).

Tools-to-data may also be made usable for researchers who have less programming skills and do not develop their own tool. Jupyter notebooks can be integrated in the tools-to-data-environment to allow researchers to see how the data may be queried and to tinker with it. Besides, the environment could offer some standard tools for processing and analysing the data, e.g. tokens, types, type frequencies, n-grams, embeddings, lemmatisation, context, syntactic function.

Updates

For updates on tools-to-data keep an eye on the CLARIAH project webpage: https://www.clariah.nl/projects/tools-to-data

Appendix: Requirements

The requirements for tools-to-data are published as an appendix to this report (in Dutch).