TOAR Data Technical Guide #1

TOAR Data Infrastructure

toar-data.fz-juelich.de

Version 2.0.0 | September 11, 2025

The TOAR Data Team

Forschungszentrum Jülich GmbH
ESDE | JSC - FSD
52425 Jülich

CONTENTS:

LIST OF FIGURES

LIST OF TABLES

CHAPTER

ONE

INTRODUCTION

The **Tropospheric Ozone Assessment Report** (TOAR) is an activity of the International Global Atmospheric Chemistry (IGAC) organisation (see https://igacproject.org/activities/TOAR). It currently runs in its second phase, TOAR-II 2020-2024, https://igacproject.org/activities/TOAR/TOAR-II (in our context denoted as TOAR V2[1]). It is collecting surface ozone measurements and related data from all over the world in a central database at Forschungszentrum Jülich, Germany. The purpose of collecting this data is to provide globally consistent metrics for analyses of health, vegetation, and climate impacts from ozone air pollution. The database is exposed via a REST API and graphical web services[2] which allow users to visualise data and compare them with other data sets and model observations. We collect data from cooperating data centres, harmonise it, and check its quality before adding it to the TOAR database.

The TOAR Data Centre governs a multi-faceted e-infrastructure with a repository for surface ozone observations and related data. This infrastructure supports the Tropospheric Ozone Assessment Report - phase II and other independent scientific studies. The TOAR Data Centre is developed and operated by the Jülich Supercomputing Centre (JSC) at Forschungszentrum Jülich (FZJ), Germany. Fig. 1.1 shows the environment of the TOAR Data Centre which includes the TOAR database infrastructure as main component (detailed in Fig. 2.1).

[1] TOAR phase I ran 2014-2019 - in our context it is TOAR V1

[2] At the time of writing the GUI is available for the TOAR V1 database only

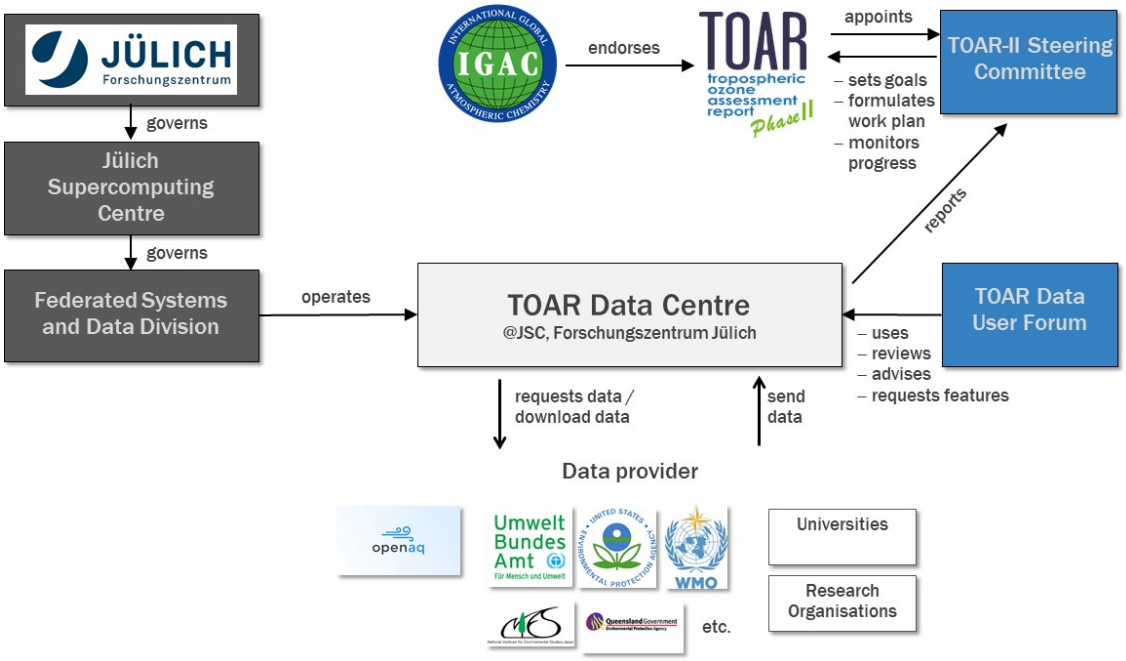


Fig. 1.1: Context of the TOAR Data Centre

This technical guide illustrates the infrastructure set-up for housing the TOAR Data Centre infrastructure and its services. It describes the service architecture and technical implementations of the TOAR data services for system administrators who need to set-up and maintain the infrastructure. The document can be shared freely as it does not any sensitive information.

CHAPTER

TWO

TOAR DATA CENTRE INFRASTRUCTURE COMPONENTS

This section details the infrastructure's systems and services as well as software, backup and other implemented housekeeping functions.

2.1 Description of System and Service Components

The TOAR Data Centre infrastructure consists of four main products and services (Fig. 2.1). These are:

- the **TOAR data portal** — a one-stop-shop to locate and access tropospheric ozone data from a large variety of measurement platforms (https://toar-data.org),
- the **TOAR web services** — an interactive GUI (graphical user interface) for the online analysis of station-based surface ozone measurements and related variables (https://toar-data.fz-juelich.de/gui/v1 | v2)[7] together with a REST API (representational state transfer, application programming interface) (https://toar-data.fz-juelich.de/api/v2 resp. https://toar-data.fz-juelich.de/api/v1) which allows for machine access to ozone data and ozone analyses,
- the **TOAR database** of station-based ground-level measurements of ozone, ozone precursors and meteorological variables. This database also contains meteorological variables from weather models. Access to the data is provided via the TOAR web services,
- the **TOAR data publication service** enables the TOAR data curators to publish data sets to the external service B2SHARE[3] at FZJ (https://b2share.fzjuelich.de/communities/TOAR). B2SHARE offers trusted long-term publication of ozone data sets as well as TOAR analysis products and it includes DOI (digital object identifier) assignment with Datacite.

[7] At the time of writing the GUI is available for the TOAR V1 database only

[3] https://www.eudat.eu/services/b2share/

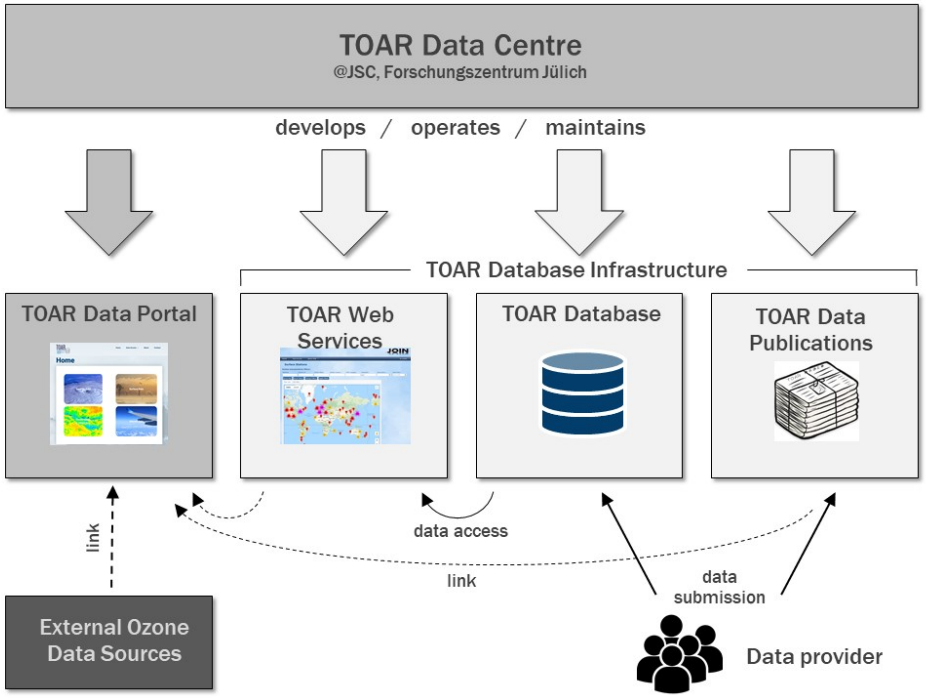


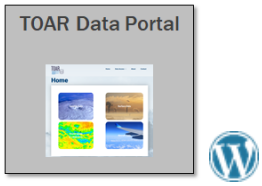
Fig. 2.1: The TOAR Data Centre infrastructure and its main service components

In addition to these four core services several other services are running in the background. These are local instances of EUDAT's B2SHARE, OpenStreetMap, and different geolocation services. In addition, software has been developed for the data ingestion workflow which is at the heart of the TOAR database.

The servers housing the services are Virtual Machines (VM) in OpenStack and VMware clusters at JSC or systems at third party providers.

The following sections describe the individual service components, starting from the four core services followed by some additional or external services which are integrated into the TOAR infrastructure.

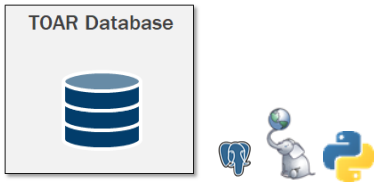
2.1.1 TOAR Data Portal



The TOAR data portal is a WordPress website hosted by Xerb[4], the domain toar-data.org is owned by Forschungszentrum Jülich and linked to the instance at xerb.de. Xerb, as host, is taking care of backing up the WordPress instance with its database that is critical for the recovery of the website.

[4] https://xerb.de/

2.1.2 TOAR Database



The TOAR database is a PostgreSQL database with PostGIS extensions. The database server runs on a VM in the Helmholtz Data Federation (HDF) cloud (see Section 3 for details). The database structure (data model) is described in Metadata Reference. The database model is also available as schema dump on gitlab and the installation instructions are given in the README there.

Software for data ingestion consists of various Python programs which are available from the gitlab repository on request. The details of the ingestion workflow are given in The TOAR Data Processing Workflow. Data ingestion also makes use of geospatial information (station metadata). The management of geospatial data is described in Section 2.1.6 - Section 2.1.8.

For administrators: the TOAR database code is available from gitlab repository and the documentation from pages.

2.1.3 TOAR Web Services

The user-accessible web services of the TOAR Database Infrastructure consist of a REST API (https://toar-data.fz-juelich.de/api/v1 resp. https://toar-data.fz-juelich.de/api/v2) for machine access to the TOAR database and a GUI (https://toar-data.fz-juelich.de/gui/v1)[Page 3, 7] for interactive data analysis.

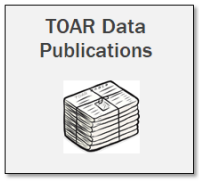
The REST API is written using the Python package fastapi (https://fastapi.tiangolo.com/). The source code can be found at the gitlab repository.

For the data processing and graphical analysis a standalone software package has been developed (toarstats) which is integrated with the TOAR V2 REST API (https://toar-data.fz-juelich.de/api/v2).

A special REST API service for flux-based vegetation damage assessment due to ozone based on the DO3SE model is currently under development. The beta version of its API can be accessed at https://toar-data.fz-juelich.de/do3se/api/v1/.

The TOAR V2 GUI is currently developed as a dashboard in Python with the help of plotly's dash library. Leaflet and our local instance of the OpenStreetMap tile service described below are used for map displays.

2.1.4 TOAR Data Publications



The TOAR data publication service is realised as a python-tool for the TOAR data curators to prepare the data for publication, specifically to map the metadata to the schema used by B2SHARE. A specific community within the EUDAT B2SHARE instance has been created for TOAR publications (https://b2share.fz-juelich.de/communities/TOAR) and only the TOAR data curators have the right to publish in this B2Share community.

2.1.5 B2SHARE

B2SHARE is external to the TOAR Data Centre infrastructure and used by the data publication service. It is a trustworthy publication archive co-developed by leading European science institutions. The instance used by the TOAR Data Centre infrastructure is running on a VM also maintained at JSC. For the publications from the TOAR community a special community metadata profile has been generated and uploaded to the B2SHARE service. The metadata profile is available from https://b2share.fz-juelich.de/communities/TOAR.

For administrators: Information about B2SHARE can be found at https://www.eudat.eu/services/b2share; the server source code can be obtained from https://github.com/EUDAT-B2SHARE. Note, however, that registration as DOI registry is necessary with Datacite (https://datacite.org/dois.html) to deliver the full functionality of TOAR data publications.

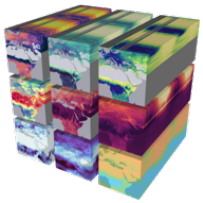
2.1.6 GEO PEAS

GEO PEAS (GEOspatial Point Extraction and Aggregation Service) provides harmonised access to information from various geospatial datasets in aggregated form so that it can be included as metadata in the TOAR database or used in special analysis procedures. The GeoLocation service consists of a REST API which is written in Python with the Django web framework. The source code of the geolocation service can be found at the repository https://gitlab.version.fz-juelich.de/esde/toar-data/geolocationservices.

Note: This service runs behind a firewall and cannot be publicly exposed, because many earth observation (EO) datasets don't allow open re-distribution.

The geospatial datasets which are processed by the GeoLocation service include a variety of EO datasets and vector data from a local instance of OpenStreetMap's Overpass service (see below).

2.1.7 Rasdaman Array Database



The EO data which is analysed by the GeoLocation service is stored in a **rasdaman array database** (geoCube) which is for internal use only. For the complete list of data sources, version information and processing instructions refer to https://gitlab.version.fz-juelich.de/esde/toar-data/geolocationservices/-/wikis/Rasdaman-Data (access with JSC account).

2.1.8 OpenStreetMap Service

A local instance of **OpenStreetMap**'s (OSM) tile service is set up on the HIFIS (Helmholtz Federated IT Services) OSM Service VM. It is used to provide the map tiles on the GUI of the TOAR phase I web service (JOIN).

Overpass API is a read-only API that serves up custom selected parts of the OSM map data. It is implemented as part of the local OSM instance.

Nominatim geocoder uses OSM data to find locations on earth by name and address (from lat/long to location, revers lookup). The TOAR database infrastructure uses it for identifying the country and state where the station is located. It is installed together with the OSM tile service and uses PostgreSQL and apache.

2.1.9 Analysis Service

The user-accessible web services for analysis of the TOAR Database Infrastructure consist of a REST API (https://toar-data.fz-juelich.de/api/v2/analysis/) for machine access to the TOAR database.

The REST API is written using the Python package fastapi (https://fastapi.tiangolo.com/). The source code is currently not publicly available.

For the data processing a standalone software package has been developed (toarstats) which is integrated with the TOAR V2 REST API. The source code for the standalone software package can be found at the gitlab repository (https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarstats).

The analysis services provide access to bulk time series downloads of time series as well as bulk aggregated time series downloads.

2.2 Server Environment

The TOAR database infrastructure is operated on different virtual machines hosted at JSC (Fig. 2.2). In addition to these VMs, the TOAR data infrastructure uses about >2 Terabytes disk space on the GPFS parallel file system of JSC and a similar amount of archive space on tapes. Backup copies are maintained at JSC and RWTH Aachen (see Section 2.3). Furthermore, the TOAR database infrastructure makes use of the B2SHARE service (administered outside of the TOAR Data Centre infrastructure, but also hosted at JSC).

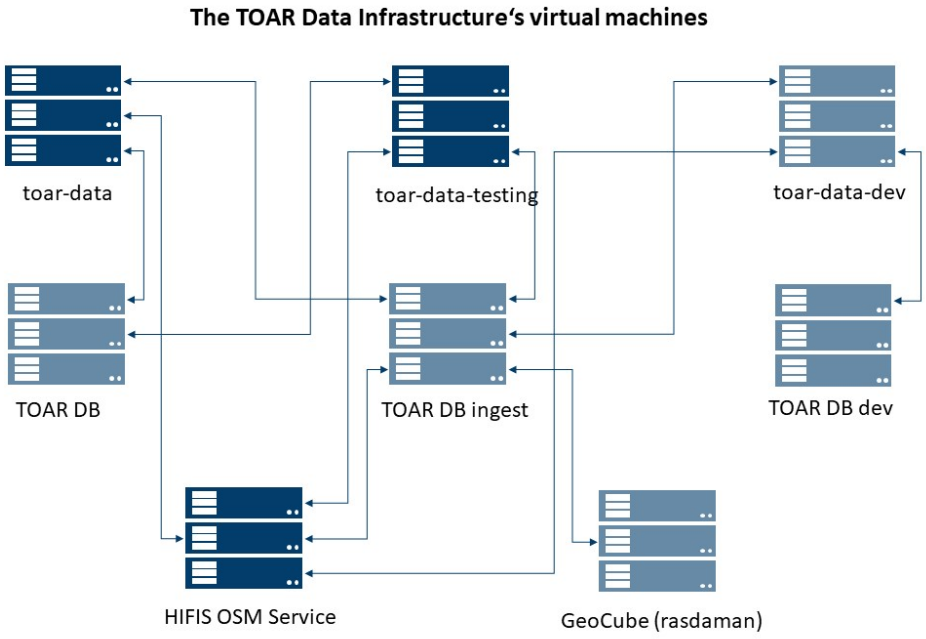


Fig. 2.2: Summary of TOAR-related VMs operated at JSC and the service relations between them; VMs accessible from outside JSC are in dark blue, the others are internal VMs.

Each VM is administered by a small number of JSC staff who are the only ones with ssh access to the system. The Openstack and VMware cloud infrastructures are also maintained by a small group of administrators, some of which also act as administrators for the TOAR VMs. The only systems which are accessible from outside JSC are the toar-data web server, the toar-data-testing instance, and the HIFIS OSM service.

The following table summarises the tasks of the virtual machines in the TOAR data infrastructure:

Table 2.1: Tasks of the VMs in the TOAR Database Infrastructure

toar-data	Interface (REST API) to the TOAR V2 database including the DO3SE web services[8] together with the interfaces (REST API and GUI) to the TOAR V1 database.

continues on next page

Table 2.1 – continued from previous page

toar-data-testing	Open access instance to new versions (alpha and beta releases) of the GUI and REST API to allow for testing by the user community.\The configuration of this VM is kept as close to the operational toar-data VM to eliminate potential problems during the rollout of new software versions. toar-data-testing is also used to test new versions of the operating system and installed software packages with the operational version of toar-data, before these system-upgrades are rolled out on toar-data.
toar-data-dev	Internal use only; development instance for GUI, Rest API, statistics, do3se and GEO PEAS
TOAR DB	VM containing full functional, complete and operational database containing all TOAR V2 data.\The operational database contains a second schema for staging data where provider data can be stored for testing purposes, which are checked by the provider and released after the provider's OK.
TOAR DB ingest	VM to particularly preprocess the data and stage it in a database which is set up similar to the operational DB. In addition, it contains a separate database for staging OpenAQ data.\All scripts for inserting data (individual submissions, harvesting, NearRealTime) into the production database should run from this machine. Apache Airflow[9] will be set up so that the UBA-NRT data can be processed via Airflow.\In addition, the GeoLocation service is running on this VM for verifying and adding location specific metadata.
HIFIS OSM Service	Lookup of locations; OSM Nominatim, OSM Overpass, OSM tileservice, postgresql, and apache are installed on this VM.\This service is primarily used by the HIFIS Community but is also used in the TOAR data preprocessing steps and by the GUI.
TOAR DB dev	This VM is set up "as equal as possible" (same operating system, same PostgreSQL version) to TOAR DB.\It can be used to test operating system/software upgrades to avoid surprises on the production system. On this VM there are two database instances "toardb-testing" and "toardb-dev":\"toardb-testing" is set up to test database schema, operating system, posgresql version. It should not be too small to allow for performance testing. It must be clear to test users that toardb-testing is not an exact copy of toardb.\"toardb-dev" is, as the name suggests, a pure development database.
GeoCube	Internal use only; RASDAMAN array database with geolocation data used in the data ingestion process.

A list of the installed software stack on each of the VMs listed above can be found in the Annex Section 5.

2.3 Data Locations and Backup Facilities

Each of the VMs is set up such that daily incremental backups are automatically created from all data belonging to the respective VM. JSC is using IBM Spectrum Prospect (has been known as Tivoli Storage Manager) for backup. The backup of TOAR DB is replicated to RWTH Aachen University's computing centre. In addition, database dumps of the operational TOAR database (on VM TOAR DB) are taken annually and at certain events, for example when the database state is frozen to provide a consistent analysis base for the TOAR phase II assessment report. All database dumps will be accessible via B2SHARE. The code for setting up a new database instance and loading database dumps into this instance are provided publicly in the git repository[5] including a (step-by-step documentation in the README.md file[6], see also Section 3

[8] Note that the integration of DO3SE is still work in progress at the time of writing of this documentation
[9] https://airflow.apache.org/
[5] https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi
[6] https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi/-/blob/master/README.md

below).

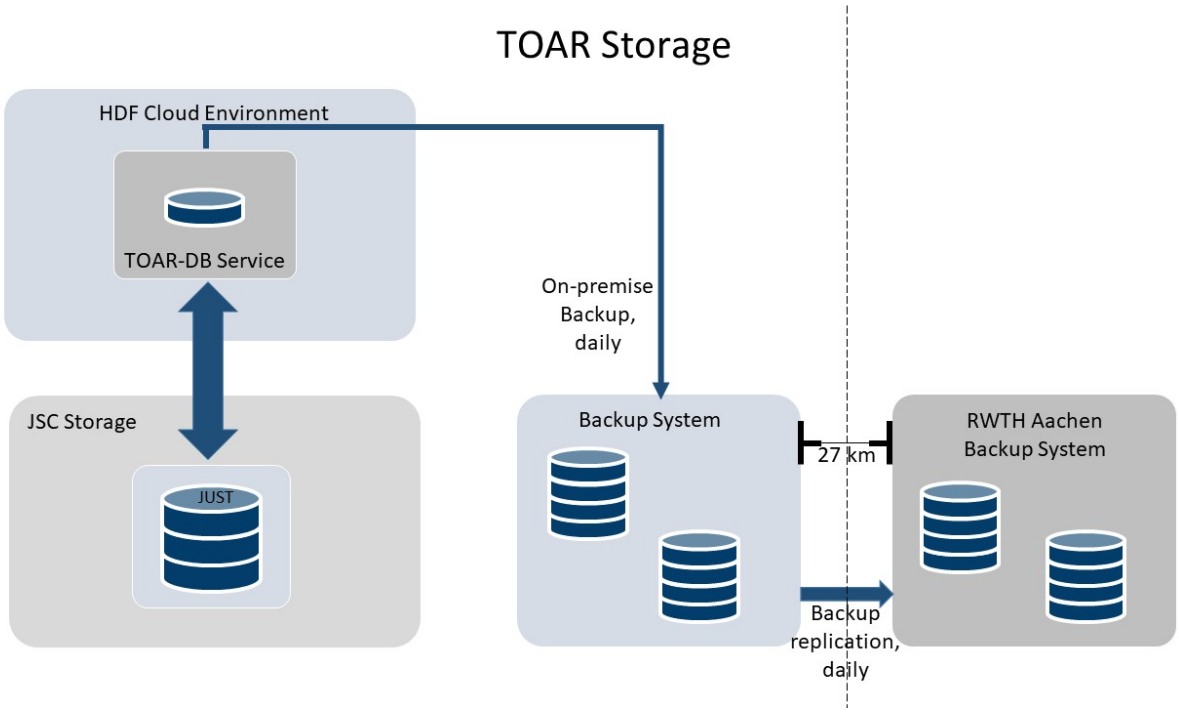


Fig. 2.3: Storage and Backup System

Raw data files downloaded from other air quality data archives or sent to the TOAR data centre by email are archived on tapes in the JSC archival storage. At least one additional copy is always available on a local PC or workstation.

CHAPTER

THREE

TOAR DATABASE SETUP AND OPERATION

This section is intended for administrators of a TOAR data infrastructure installation.

It describes two use cases:

- setup of a database copy for local use (“clone”)
- setup of a database mirror site

The technical setup and the procedures for operating the TOAR database system are described in the following using the system setup at the Jülich Supercomputing Centre (JSC) as an example. The description is intended to be as generic as possible.

The TOAR database is a PostgreSQL database with the PostGIS and toar_controlled_vocabulary[10] extensions installed. A suitable server configuration for hosting a copy of the TOAR database should include at least 2 cores, 4 GBytes of memory and 3 TBytes of storage space. The system will benefit from additional cores and memory under high load. Sufficient IO Bandwidth should be provided as even the indices of some tables can grow quite large and cannot not always be cached in memory. For example, the indices of the data_hourly table are at around 100GB at the time of writing of this documentation. The current system configuration at JSC has 2 cores, 4GB memory, 4TB disk space. Running in a virtual environment, these parameters can easily be scaled up if needed.

The Postgres data directory is located under /postgres/data. Other directories under /postgres are

- /postgres/archive - (compressed) WAL files written by the DB server process
- /postgres/backup - (compressed) base backups

3.1 Setup of a Local Database Copy (Clone)

If you don’t want to mirror the TOAR database, but only install a local copy for your own use, you can clone a published snapshot of the database. Therefore, this section only focuses on setting up a database clone and does not describe how to perform backup and recovery; this can be adapted to the local conditions of your operating system. Suggestions can be found in Section 3.3.

[10] https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi/-/tree/master/extension/toar_controlled_vocabulary

3.1.1 Installation

The necessary steps for the installation depend on your actual set-up, e. g. whether PostgreSQL/PostGIS is already available on your system, or whether you already defined roles in your database system.

An installation from scratch contains the following steps:

1. the configuration files for a fresh PostgreSQL/PostGIS installation (on an Ubuntu system) have to be adapted and the installation needs to be done. Second, a new database named “toardb_v2” has to be created with the following roles:
    - *toaradministrator*: Superuser who owns the database and all its contents
    - *toarcurator*: A user with read and write access to TOAR metadata and data
    - *toaruser*: A user with read only access to TOAR metadata and data
2. the PostGIS and toar_controlled_vocabulary extensions have to be installed within this new database.
3. the necessary database tables (see: https://esde.pages.jsc.fz-juelich.de/toar-data/toardb_fastapi/docs/toardb_fastapi.html#models and checkout the database schema at https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi/-/blob/master/toardb_v2_schema.sql) are set up. The repository also contains a small set of test data to fill the newly created database toardb_v2 for testing purpose.
4. install the REST API from https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi/ to then test your new database with https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi/-/blob/master/production_tests.sh.

3.1.2 Loading Data

In order to populate a fresh database, it is necessary to load a dump (see installation guide at https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi#installation-guide). Database dumps are done on a regular basis and made publicly available on b2share (https://b2share.fz-juelich.de/communities/TOAR).

3.2 Setup of a Database Mirror Site

3.2.1 Installing a Mirror Site

At JSC, the TOAR database is located on a virtual machine (VM) in the HDF Cloud[11] currently reachable as zam10131.zam.kfa-juelich.de. To install a database on your mirror site, follow the instructions given in Section 3.1.1.

3.2.2 Loading Data to Mirror Site

In order to populate a fresh database with the current database of TOAR-II, it is necessary to load a current dump (see installation guide at https://gitlab.jsc.fz-juelich.de/esde/toar-data/toardb_fastapi#installation-guide). Database dumps are done on a regular basis and made publicly available, but this will never mirror the actual database system. To get a current dump of toardb_v2, the TOAR Data Centre has to be contacted (support@toar-data.org).

[11] Jülich Supercomputing Centre. (2019). HDF Cloud – Helmholtz Data Federation Cloud Resources at the Jülich Supercomputing Centre. Journal of large-scale research facilities, 5, A137. http://dx.doi.org/10.17815/jlsrf-5-173

3.2.3 Data Updates

Updates of data in the database happen automatically multiple times per day via the real-time scripts. Manual additions to the database for individually submitted data are done with various python scripts. For details see The TOAR Data Processing Workflow.

3.3 Operating a Mirror Site

When operating a mirror site, special precautions should be taken to avoid data loss or a mismatch between the original site and the mirror site. Therefore, the following two sections describe the measurements of backup and recovery that are taken by the original site at the TOAR Data Centre.

3.3.1 Backup

Due to the size of the DB (>2TB), regular backups of the entire DB at short intervals, e.g. 1 day, seem unrealistic. Therefore, we use the continuous archiving scheme, for which one uses a base backup of the entire DB server contents (can be done w/o any downtime) combined with so called write-ahead-log files (WAL).

This approach is described in the PostgreSQL manual[12]. WAL files are written after they reach a size of 16MB, which is the default for PostgreSQL but at the latest after 6 hours.

Both the monthly base backup as well as all WALs that were written in the meantime are backed up to JSC's central backup facility. A mirror of this data is also sent to RWTH Aachen University for a geo redundant backup. We currently keep the backup data of the past two months as active copies on the system. The oldest of these copies is deleted on the first day of every month, when the monthly base backup has been created. We expect the backup server to discard old backups of the data more than 30 days after their inactivation.

3.3.2 Recovery

The DB can be recovered using the most recent base backup in /postgres/backup and a recovery.conf file in the data directory (/postgres/data). The single relevant entry for recovery is this (cf. the PostgreSQL documentation[13]):

```
restore_command = 'zcat /postgres/archive/%f > %p'
```

As you can see, the WAL archive files are expected under /postgres/archive, which is the location where they are written. The current setup is such that WAL archives are compressed using gzip, therefore we need to uncompress them using zcat during recovery.

Should the entire VM have been lost, the data must be restored from the backup server using TSM Spectrum Protect software[14]. The backup node for the VM is registered as TOAR-DB.

In case of a loss of the entire data centre, including the site-local backup, the backup is available at the mirror site RWTH Aachen University employing the same backup software as is used for the local setup. It is possible to retrieve the backup of the data from there. When building an entirely new instance outside JSC, support of the administrators at RWTH will be required.

[12] https://www.postgresql.org/docs/10/continuous-archiving.html
[13] https://www.postgresql.org/docs/10/index.html
[14] https://www.ibm.com/products/data-protection-and-recovery

3.4 System Configuration

We employ configuration management through Puppet[15], which is responsible for the entire configuration of the system and database server. The Puppet classes assigned to the host comprise a number of profiles to configure the following aspects:

Table 3.1: System configuration

Module	Function
fsd::nrpeprobes::pgsql	Nagios NRPE probe to automatically monitor PostgreSql server availability
fsd::profile::big_postgresql	PostgreSQL server setup for a “big” database including WAL based continuous backup, including a monthly base backup
fsd::profile::defaults	general defaults employed within the group Data Services and Infrastructure at JSC
fsd::profile::firewall::defaults	disallow all inbound traffic on ports other than the ones explicitly specified
fsd::profile::firewall::tsm	allow inbound traffic from backup server
fsd::profile::toar_db	create PostgreSql DB administrators

While this scheme can be employed to quickly bring up another TOAR DB server, manual configuration may be desirable when upgrading between different versions of PostgreSQL, as this is not well supported by the external Puppet classes we use for this purpose. In order to install multiple versions of PostgreSQL at the same time, the packages provided by the PostgreSQL developers are well suited for this[16].

3.5 File System Backup

For file system backup the local TSM backup client is used. As almost everything on the machine is configured automatically, we merely backup the payload data. This comprises /home and /postgres with the exception of /postgres/data. As a consequence, the include/exclude list on the machine is defined as:

```
exclude /.../*
include /postgres/.../*
exclude.dir /postgres/data
include /home/.../*
```

3.6 Additional Procedures

After importing large amounts of data, there is an increased number and volume of write-ahead-log (WAL) files until the next base backup will be created. This number and volume are cumbersome to handle. It is therefore advisable to do another base backup after such imports. This can be as easy as running the base_backup script manually, ideally in a terminal multiplexer (tmux) session or using nohup. The script and its parameters can be derived by:

```
$ sudo crontab -u postgres -l
  @monthly python3 /postgres/base_backup.py -b /postgres/backup -a \ /postgres/archive
```

[15] https://puppet.com/docs/puppet/6/puppet_index.html
[16] https://www.postgresql.org/download/linux/ubuntu/

Generally, we keep the last two monthly base backups and the corresponding WAL files, such that a point in time recovery would be possible for the past two months. Besides the local backup of database dumps and the WALs, once a month a complete database dump is transferred to RWTH Aachen University. In addition, every day the changes to the latest database dump are also transferred to RWTH Aachen University. Therefore, the database can always be restored from the backups in Aachen in the unlikely event of a complete data loss at JSC.

CHAPTER

FOUR

RISK ASSESSMENT

The TOAR database contains data from various networks of measuring stations, which are made available through an ecosystem of web services (Fig. 2.1) for evaluation purposes. In some cases, the data is sent from individual institutions directly to the TOAR data team and is processed according to a semi-automated workflow (see description in Automated Data Preparation). All raw data files, harmonised metadata and data are stored at Forschungszentrum Jülich. They are long-term archived and backed up. In order to keep the risk for this data and the database as low as possible, we have carried out an internal risk assessment. The data risks are evaluated on the basis of the risk assessment matrix developed by Matthew S. Mayernik et.al.[17].

In general, data in the TOAR database is available under CC-BY 4.0 licence[18]. Personal data is not stored in the database. We take action against each individual risk factor and carry out a new risk assessment every two years.

In the following the main risk categories for the TOAR repository are given with the relevant risks factors including the estimated degree of control[19], the estimated impact on users, and the countermeasures taken proactively:

4.1 Data Risks

- Lack of metadata & documentation (**risk: high, impact: low**)

  During data ingestion into the TOAR database the available metadata is checked for plausibility and missing metadata is generated where possible. Direct communication with data providers is used to clarify incomplete or ambiguous data submissions. Documentation of the database is thoroughly maintained by the TOAR database infrastructure team.

- Data mislabelling (**high, low**)

  To avoid data misidentification, we conduct thorough testing of new software and new data submissions. The workflow design includes inspection of ingested data by the provider as well as use of the database in assessment reports (prompts many scientists to scrutinize the data when analysing it). Extensive documentation of errors and regular training of database administrators to sensitise them to these issues are performed.

- Poor data governance (**high, low**)

  New staff members get an intensive training and a tight control of access rights and responsibilities is in place (only experienced personnel is allowed to curate data). An automation of curation processes

[17] Mayernik, M.S., Breseman, K., Downs, R.R., Duerr, R., Garretson, A., Hou, C.-Y. (Sophie) . and (EDGI) and Earth Science Information Partners (ESIP) Data Stewardship Committee, E.D.G.I., 2020. Risk Assessment for Scientific Data. Data Science Journal, 19(1), p.10. DOI: http://doi.org/10.5334/dsj-2020-010

[18] https://creativecommons.org/licenses/by/4.0/

[19] Degree of control: How much control an organization or individual has over whether a risk factor is present or will occur

is adopted where possible. The possible deterioration of data governance is minimized by retraining or replacement of personnel.

- Accidental deletion (**high, medium**)

  Only very few skilled people have root privilege on the TOAR database infrastructure. System updates and major database updates are planned by at least two people. The infrastructure is set-up so that it rarely requires manual intervention. Restoring data from backup copies is possible.

- Lack of planning (**high, high**)

  In order to avoid overloading a leader, measures for the transfer of responsibilities are established; written documentation and planning are important to us. Also regular team meetings to discuss issues, progress and plans are held.

- File format obsolescence (**medium, low**)

  Regular user forums and user interaction will allow to foresee requests and code development can be planned early and will prevent file formats from becoming obsolete. Data are not stored as files but in a database, i.e. a conversion tool can easily be made available. Durable file formats have been chosen for the database.

- Lack of provenance information (**low, low**)

  Clear rules for data submission including specific metadata attributes for provenance and direct contact with data providers are established as precaution.

- Over-abundance (**low, low**)

  In this case direct communication with data providers is sought and careful planning of calls for data submission is done. A prioritisation of processing and curation is used to limit the impact of the delays. We expect such a situation to occur very rarely as the total expected data volume is quite well known.

4.2 Physical Risks

- Media deterioration (**very high, medium**)

  To prevent the main danger from ageing of hard disks or tapes a comprehensive backup strategy including copies at a remote location (RWTH Aachen). All file systems are located on RAID (Redundant Array of Independent Disks) mode disks including those for the operational database and web services. The hardware is closely monitored and in case of hardware degradation the hardware is replaced. Should such failures lead to loss of data, it can be restored from backup.

- Storage hardware breakdown (**medium, medium**)

  In order to counteract this risk, the following measures are taken: continuously monitoring of hardware components and hot-swap strategy as well as regular acquisition of new hardware and re-installation of database and services.

- Cybersecurity breach (**medium, medium**)

  Elevated security measures are implemented at JSC and regularly reviewed and updated. As long-term HPC service provider there is strong awareness of cybersecurity issues. In case of a breach the TOAR data services are cordoned off. The TOAR database can be reinstalled from a trusted backup version in case of malicious attacks.

- Bit rot and data corruption (**medium, low**)

  Our database technology contains safeguard measures for the TOAR database. Depending on severity of the problem, either individual values are fixed manually or long time series and sets of time series are re-inserted from backup copy or from raw data submissions.

- Malicious attacks (**low, high**)

  Access to the hardware hosting the TOAR database infrastructure is regulated and only possible for designated personnel. Security measures include locked doors with entry system. Furthermore, the hardware is placed on a secured campus with manned gates.

- Human error (**low, medium**)

  The hardware is operated by trained personnel with long experience in operating complex high performance computer architectures.

- Catastrophes (**low, medium**)

  For the machine halls early warning systems are in place as well as a fire-extinguishing system (Argon) in the main one. A fully equipped fire brigade specialising in all types of fires is located at FZJ. The fire department is located about 650m away from the supercomputing centre.

  The open design of the TOAR database infrastructure and regular publicly posted database dumps allow for anyone to rebuild the TOAR database infrastructure upon loss. JSC as host of TOAR database infrastructure takes the specific precautions of supercomputing centres against loss of infrastructure and data in case of local-scale emergencies (fire, flooding, storm), including a comprehensive backup strategy with copies at a remote location. In case of partial loss, JSC staff will re-install the TOAR database and its web services on new or different hardware; in case of total destruction some other member of TOAR community would have to rebuild the TOAR database infrastructure from the automated backups and archived software stack.

4.3 Human Risks

- Lack of use (**medium, low**)

  Regular user consultation, deep embedding into the TOAR phase 2 activity, proactive technology development, testing, testing, testing, thorough data quality control including user feedback is the strategy to face this risk. Even more communication with users and data providers on issues such as poor documentation, data quality perceived as unreliable, user-unfriendly interface(s), incomplete data, or convenience and speed of data availability compared to other sources.

- Loss of knowledge around context or access (**medium, high**)

  Continuous training of personnel, building redundancy in terms of knowledge to operate the TOAR database infrastructure are in place as well as building up knowledge about the TOAR database infrastructure in the TOAR community. Software and data is archived or published so that the system can be rebuilt by trained personnel.

4.4 Organisational Risks

- Loss of funding for archive (**medium, high**)

  Should there be a risk of termination in sight, the TOAR community will be alerted and a replacement will be sought actively. Furthermore, all software and data backups are made publicly available so that a new data centre can be launched with relatively little effort. Relocation of the TOAR database to another site is possible.

- Legal status for ownership and use (**low, medium**)

  We only accept data under the condition that they are published with an open access license (CC-BY 4.0). There is no duration limit of this license, so it cannot be revoked after data has been published

in the TOAR database. In most cases, the TOAR community will seek disclosure. Changes to rules for the release and use of data is very unlikely.

- Political interference (**low, high**)

  The TOAR data centre and its achievements are openly communicated, so no specific measures are taken. In the extreme case, another TOAR partner is sought from another country who could take over the TOAR database infrastructure.

4.5 Additional Risks Related to Backend Web Services

In addition to the data risks adopted from Matthew S. Mayernik we also identified two additional risks related to the operation of the web services which are an important part of the TOAR data infrastructure. Since there is usually only very little data directly associated with the web services (aside from potential user settings, static images, etc.), it is generally easier to re-install these services from the code repositories.

The additional risks for services are therefore:

- Failure to re-install a service from the repository (**medium, low**)

  This could happen because of incompatibilities in software libraries if the fresh installation uses a different operating system or Python version than the original installation. Another possibility for this risk to materialize is due to coding errors in the ongoing development of the software. The first reason is mitigated through the 3-level deployment strategy, i.e. every new installation (developed on toar-data-dev.fz-juelich.de) is first made on toar-data-testing.fz-juelich.de before it is carried over to the operational system at toar-data.fz-juelich.de. The second aspect of this risk is kept at bay through emphasizing good coding practices including a carefully designed testing strategy. Nevertheless, occasional time pressure and lack of resources will prevent our software from becoming 100% perfect.

- Broken connection between services (**medium, low**)

  If services must be re-installed, possibly under a new domain name, it might happen that essential data connections are broken. This should become apparent while testing a new deployment on toar-data-testing.fz-juelich.de. The risk is controlled by maintaining all configurable parameters in separate configuration files so that it is easy to track down such errors.

CHAPTER

FIVE

ANNEX: TOAR DC SOFTWARE STACK

5.1 A1 Software

TOAR DB, TOAR DB testing/dev:

- Ubuntu 20.04.2 LTS
- PostgreSQL 13

TOAR DB ingest:

- Ubuntu 20.04.2 LTS
- Python 3.8.10
- MongoDB 4.4.6
- geolocation service

toar-data, toar-data-testing, toar-data-dev

- Ubuntu 18.04.5
- Python 3.10.1
- toardb_fastapi (FastAPI 0.54.1)

GeoCube:

- Ubuntu 18.04.5
- Rasdaman: 10.0.0bionic-9
- Tomcat: 9.0.16-3
- PostgreSQL 10.18

HIFIS OSM Service:

- Ubuntu 20.04.2 LTS
- Postgresql 12.8
- Mapnik 3.0
- Apache 2.4.41

5.2 A2 List of Software Repositories

REST API[20]

toarstats[21]

toarqc[22]

geolocation service[23]

DO3se web service[24]

landing page[25]

toar analysis services[26]

[20] https://gitlab.version.fz-juelich.de/esde/toar-data/toardb_fastapi/
[21] https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarstats
[22] https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarqc
[23] https://gitlab.jsc.fz-juelich.de/esde/toar-data/geolocationservices
[24] https://gitlab.jsc.fz-juelich.de/esde/toar-data/do3se-web-service
[25] https://gitlab.jsc.fz-juelich.de/esde/toar-data/toar-data-landing-page
[26] https://gitlab.jsc.fz-juelich.de/esde/workflows/toar-workflow