21 September 2022

Data Delivery System

Technical Overview

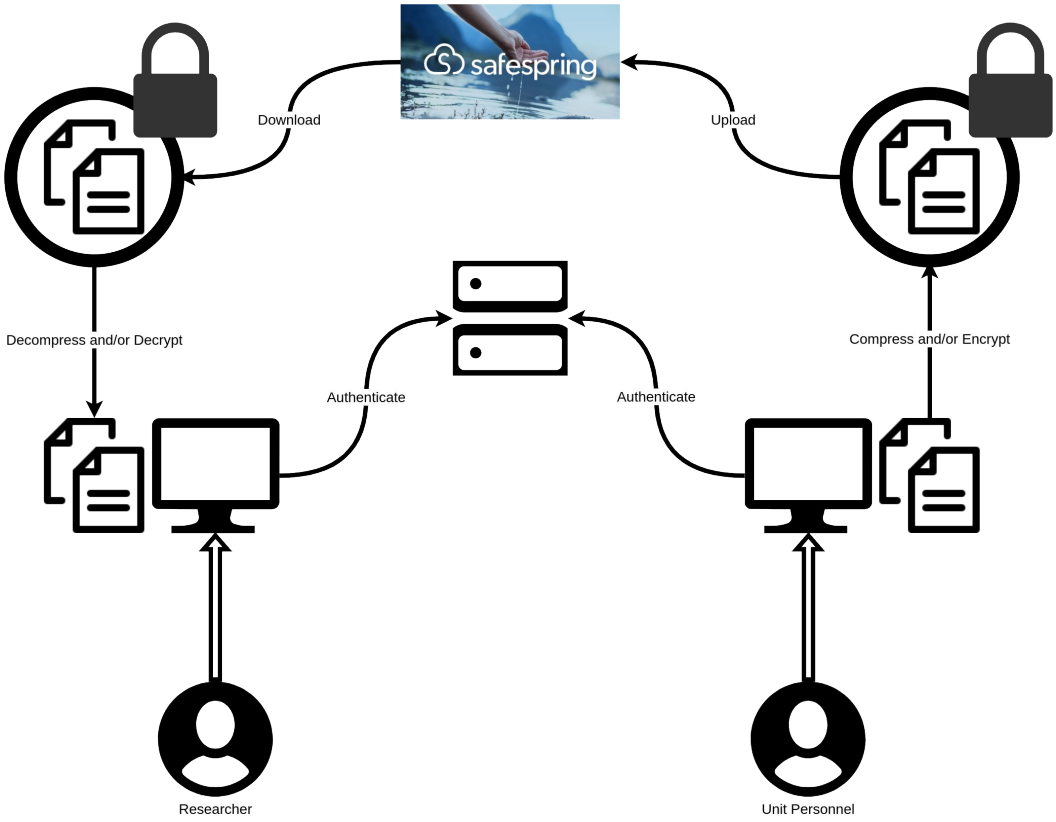
Abbreviations and terminology

2FA	Two Factor Authentication
API	Application Programming Interface
Archiving	Changing the project status to Archive
Authentication token	In the context of DDS, a file created upon successful user authentication and used for secure API calls during limited period of time
CLI	Command Line Interface
DDS	Data Delivery System
DiA	Days in Available
DiE	Days in Expired
Kubernetes cluster	A set of node machines for running containerized applications
NGI	SciLifeLab National Genomics Infrastructure
PI	Principal Investigator
PO	DDS Product Owner. Member of the SciLifeLab Data Centre responsible for the DDS project, including development and external contact such as with SciLifeLab units.
Releasing	Changing the project status from *In Progress* to *Available*
REST	REpresentational State Transfer[1]
Retracting	Changing the project status from *Available* to *In Progress*
Safespring	Swedish provider of locally based Cloud services
SUNET	Swedish University Computer Network
Sysadmin	System administrator
Unit	Within DDS, the representation of a SciLifeLab platform or one of the units within a platform
Web (part)	Interface for accessing the DDS through the Internet using a web browser

[1] https://aws.amazon.com/what-is/restful-api/

1. What is the Data Delivery System?

The Data Delivery System (DDS) is a tool built for the simple and secure delivery of data from SciLifeLab platforms to their users. The system uses Safespring's[2] object storage service as the delivery medium, thereby keeping the data within the Swedish borders. The DDS has built-in encryption and key management, enabling protection of the data without requiring the users to perform these steps themselves prior to delivery.



The DDS has been developed by the SciLifeLab Data Centre together with NGI Stockholm. The web interface can be found at https://delivery.scilifelab.se and the documentation at GitHub pages.

What is it not?

The system is not a storage solution; The data will only be stored for a short period of time[3] and the recipients are themselves responsible for the future handling of their data. Backup of the data is not provided during the data delivery. The system is also not built for data sharing; it is purely for delivery of data from one data producer to a data owner.

---

[2] For more information on Safespring and their storage service, visit https://www.safespring.com/en/services/storage/

[3] The storage time is described below: Automatic expiry of data access, Automatic archiving of project and in the Appendix - Project Statuses.

2. How is it used?

The DDS is hosted in a Kubernetes cluster[4] owned and maintained by the SciLifeLab Data Centre. The system consists of two main components: a web part ("web") and a REST API ("API"). The vast majority of the DDS functionality resides in the API, which can be utilized via the DDS command line interface (CLI). At this time, the web can be used for user registration (Registration), changing and resetting passwords and listing the projects. In addition, the web links to the code base, documentation and the Python Package Index (PyPi). In order to meet the needs of more users (e.g. those not as familiar with the command line) we plan to include more functionality currently present in the CLI into the web.

This section describes the standard DDS use-case and the flow of data within the system. The steps do not include actions taken prior to data delivery, such as sample collection, the production of the data or the communication between customer (user) and SciLifeLab platform.

Accounts within the system will be referred to by the role of the account in question.

- A *Super Admin* refers to the person behind a DDS account with the role "Super Admin"
- A *Unit Admin* refers to the person behind a DDS account with the role "Unit Admin". Unit Admins are associated with a SciLifeLab unit.
- A *Unit Personnel* refers to the person behind a DDS account with the role "Unit Personnel". Unit Personnel are associated with a SciLifeLab unit.
- A *Researcher* refers to the person behind a DDS account with the role "Researcher". A Researcher is any person who uses the services of a SciLifeLab unit.
- A *Project Owner* refers to the person behind a DDS account with the role "Researcher", but who is marked as the *owner* of a specific project. For more information on the account roles and their permissions within the DDS, please see the appendix (User Roles).

2.1. Delivery Overview

Sections 2.2 to 2.11 provide a more detailed description of the following steps.

1. A *Super Admin* creates a *unit* in the DDS
2. A *Super Admin* invites *Unit Admins*
3. The *Unit Admins* register their accounts in the DDS
4. The *Unit Admins* invite *Unit Personnel*
5. The *Unit Personnel* register their accounts in the DDS

[4] The cluster nodes can only be accessed by admins.

6. A *Unit Admin / Personnel* create a project
7. A *Unit Admin* / *Personnel* uploads data
8. A *Unit Admin / Personnel* releases the data to the recipient
9. A *Researcher* / *Project Owner* downloads data
10. The project status is automatically changed to *expired* after a number of days
11. If the project status has not been changed to *available* again within a number of days, the project is automatically *archived*

2.2. Creating a Unit in the DDS

In order to begin using the DDS, a SciLifeLab platform or unit must contact the SciLifeLab Data Centre at datacentre@scilifelab.se. When the necessary agreements are in place, the unit sends the following information to the DDS Product Owner ("PO") (* indicates required information, none indicates optional):

Unit Name * The name of the unit or platform

External Unit Name The unit name to display to users e.g. in emails

Contact Email * The email address on which the unit would like to be contacted regarding the DDS. This address will be displayed in some emails sent by the DDS and on the web page, e.g. when renewal of project access is required.

Public ID An identifier for the SciLifeLab unit (e.g. abbreviation of the SciLifeLab unit name) which may be displayed publicly. If not chosen by the unit, this will be identical to the Internal reference ID. If the public ID is chosen and the Internal reference ID is not, this field must adhere to the rules listed under the Internal reference ID below.

Internal reference ID At this time, this specifies the prefix of the public ID of the future projects within the unit. Preferably, this ID should be a short abbreviation of the unit name, and should adhere to the following rules.

- Only contain letters, digits, dots (.) and hyphens (-).
- Begin with a letter or digit.
- Contain a maximum of two dots.
- Not start with "xn--".

Days in Available The number of days during which the data will be available for download by the *Researchers.* The countdown starts when the project is released. There is no time limit when the project is *In Progress* and the project has not been released. For more information on the project

statuses and what actions can be performed during them, see the appendix (Project Statuses). After *Days in Available* (DiA) number of days has passed, the project is automatically set as *Expired*.

Days in Expired The number of days (after being available) during which the data is still stored but not available for download. During this time, the project can be *renewed*, leading to the project being available again and therefore allowing for downloads again. When the *Days in Expired* (DiE) number of days has passed, the project is automatically archived by the system.

The unit also must provide the email addresses of at least two (but preferably three or more) individuals which should have the DDS account role *Unit Admin.* For more information on why this is required, see the section Creating a Project.

When this information has been received, the Data Centre contacts Safespring in order to create a new Safespring project (not to be confused with a delivery project). The Safespring project is the location within the cloud where the uploaded data will be stored. When a Safespring project is created, the PO receives the following information from Safespring:

Keys *Access key* and *secret key*. These are sent within an encrypted file, only decryptable for a certain amount of time by a certain person - the PO.

Endpoint The storage site. This is currently the same for all, but may change in the future, either for all or some of the Safespring projects.

Name A unique name for the Safespring project. This is currently not used for anything aside from keeping track but will be useful information once the invoicing system has been implemented.

A system administrator (sysadmin) in the Data Centre accesses the DDS deployment and runs a command using the information described above. This inserts the unit information into the database. The database is encrypted with the Advanced Encryption Standard (AES) mode Counter (CTR), with a 512 bit key.

2.3. Registration

In order to obtain an account, another user already having an account in the system needs to send an invite[5]. When a unit is first created in the DDS, the *Super Admin* invites *Unit Admins* to the specific unit. When invited, the *Unit Admin* receives an email from the DDS with the subject

[5] An invite is valid for 7 days. At this time, there is no clean up functionality to remove unanswered invites, however the invitee can follow the expired invite link to deactivate it completely or the inviter can remove the invite. Both of these alternatives can then be followed by a new invite to the same user if needed.

“[USER] invites you to the SciLifeLab Data Delivery System“. The email contains a link to the registration page. When following the link, the user is prompted to enter the following information. All fields are required.

Field	Requirement(s)	Note
Name	At least two characters long	The full name of the person registering an account.
Username	Not taken by another user	It is not possible to change the username at this time. Please save the username in a secure way, e.g. a password management software.
	Between three and 30 characters long	
	Can only contain the following characters: ● Letters ● Digits ● Underscore (_) ● Dot (.) ● Dash (-)	
Password	Between ten and 64 characters long	Please remember your password. We highly recommend storing it using a password management software such as BitWarden or LastPass. Forgetting your password means potential loss of access to data. You can reset your password, however this means that another user needs to renew your project access if there are any active and ongoing projects. If your access is not renewed, or all users that are able to renew your access also have forgotten their passwords, any data that has been uploaded will not be possible to download or decrypt and therefore a new delivery project will be needed. *Super Admins* do not have access to any of the data or your passwords and therefore cannot help you retrieve any data if you lose project access.
	Must contain at least… ● One upper case letter ● One lower case letter ● One digit **or** special character	
Email Address	Not registered to another account.	This cannot be changed as the invite is specific to one email. In addition one email address can only be connected to one account and by extension one account role.

Once a *Unit Admin* has registered an account, they can invite *Unit Personnel*. *Unit Personnel* can only be invited to the unit to which the *Unit Admin* is connected. *Super Admins*, *Unit Admins* and *Unit Personnel* can all invite *Researchers*. All invited users get identical emails as described above, and register their account in the DDS by following the link and filling in the same required information.

2.4. Authentication

Before delivering data with the CLI, it needs to be installed (see the Installation Guide located in GitHub pages). all users are required to authenticate themselves. The authentication process via CLI is as follows.

1. The user runs[6] the authentication command and the CLI asks for the username and password.
2. Assuming the username and password are correct, the user is asked to authenticate with two factor authentication (2FA). The first authentication requires 2FA via email: An email containing an eight digit code is sent to the user's registered email and the user is prompted to enter the code. After this, the user can set up 2FA via authenticator app[7] if they wish. If not, the default method of 2FA via email will be used for all authentication within the DDS.
3. Assuming that the 2FA succeeds, an encrypted token is saved on the local machine or server, depending on where the command is executed. This token acts as an authenticated session and will be used in the succeeding commands. The default destination of the encrypted authenticated token is the home directory. If a specific destination is specified in the authentication command, the token will be saved in that location[8]. The token is valid for seven days. We highly recommend you to re-authenticate before uploading or downloading any data[9].

The same procedure is followed when logging into the web; The username and password is entered, followed by 2FA with the chosen 2FA method. The web session however, in contrast to the CLI session, only lasts for one hour. If a user is inactive for more than one hour, they will be asked to log in again. As a final note on the authentication process, a user can attempt the authentication via CLI and login via the web a maximum of ten times in one hour.

---

[6] https://scilifelabdatacentre.github.io/dds_cli/auth/#dds-auth-login

[7] https://scilifelabdatacentre.github.io/dds_cli/auth/#dds-auth-twofactor

[8] If a custom destination is chosen during authentication you also must specify the path to the token in all following commands.

[9] We're looking into the possibility of authenticating during an ongoing delivery to reduce the risk of a failed delivery due to an expired token. This is not implemented at this time. Therefore, if the delivery is not finished before the token expires, the ongoing upload or download will fail.

2.5. Inviting users

The CLI is used to invite any type of user[10]. The following list describes which roles can invite which roles, if any, and if there are any exceptions or additional important information to note.

<table>
<tr><th rowspan="2">Inviter role</th><th colspan="4">Invitee role</th></tr>
<tr><th>Super Admin</th><th>Unit Admin</th><th>Unit Personnel</th><th>Researcher</th></tr>
<tr><td>Super Admin</td><td colspan="4">A Super Admin can invite any type of user to the DDS in general. They cannot, however, invite users to specific projects, since they do not have access to any projects themselves or any of the data uploaded within them.</td></tr>
<tr><td>Unit Admin</td><td>×</td><td colspan="2">Unit Admins can invite Unit Admins / Personnel to the unit which their account is associated with. [11]</td><td rowspan="2">Unit Admins / Personnel can invite Researchers, both within the system in general, and to specific projects associated with the unit the Unit Admin / Personnel is associated with.</td></tr>
<tr><td>Unit Personnel</td><td>×</td><td>×</td><td>Unit Personnel can invite other Unit Personnel to the unit which their account is associated with. [14]</td></tr>
<tr><td>Researcher</td><td>×</td><td>×</td><td>×</td><td>Researchers cannot in general invite any other users. However, if the Researcher is marked as the owner (Project Owner) of a specific project, they can invite both other Project Owners and Researchers into that specific project.</td></tr>
</table>

[10] https://scilifelabdatacentre.github.io/dds_cli/user/#dds-user-add

[11] Unit Admins / Personnel cannot be invited to specific projects - all accounts with these roles have access to all projects within the specific unit.

2.6. Creating a Project

Unless otherwise stated, a *project* refers to a *delivery project* - a container in the DDS to which a unit uploads one or more data files to be delivered to a researcher.

Both *Unit Admins* and *Unit Personnel* can create delivery projects for their unit. However, a unit will not be able to create any delivery projects until there are at least two accounts with the role "Unit Admin" for the specific unit. The reasoning behind this is explained below:

When a user registers an account in the DDS (Registration), a key pair is generated ("user key pair") for that specific user. The public part of this pair ("user public key") is stored in the database without any protection apart from the encryption affecting the entire database. The private part of the key pair ("user private key"), however, is encrypted by a key derived from the password chosen during registration and thereafter saved to the database.

A key pair is also generated when a project is created ("project key pair"). In the same way as the user public key, the public part of the project key pair ("project public key") is saved to the database without an extra layer of encryption. The private part ("project private key") is encrypted with the user public key.

The above means that the project private key can only be decrypted with the user private key, which can only be decrypted with a key derived from the user's password. Since the project private key is required for decrypting all data within a specific project, forgetting the DDS password and resetting it also results in the loss of data access. Note that this is not the case for *changing* the password, where the user logs in to the web and provides the old password in order to set a new one. This does not affect the data access in any form.

The reasoning behind requiring at least two *Unit Admins* for a specific unit is to reduce the risk that all *Unit Admins* associated with a specific unit lose access to the project data at the same time. If this would occur, they would not be able to fix the others access[12] and *Unit Personnel* cannot fix the project access for *Unit Admins.*

Note: Users only choose and keep track of their username and password. All other keys are generated by the system and stored in the database.

If a unit has two *Unit Admins*, any user attempting to create a project will be prompted with a warning, informing them of the risk of data access loss[13]. Once a unit has at least three *Unit Admins* this warning will disappear.

[12] https://scilifelabdatacentre.github.io/dds_cli/project/#dds-project-access-fix

[13] "Data access loss" refers to the occasion that no user connected to a specific unit has access to download and decrypt data within a specific project. *Super Admins* can not help with this since they do not have access to any of the data.

The following information is required (and saved to the database) when creating a project:

Title The title of the project. There is no upper limit to the length of this, however it must be at least one character long and it can only contain letters, digits and spaces - special characters are not allowed. It is also not unique; there can be multiple projects with identical titles.

Description A description of the project. It needs to be at least one character long, but there is no upper limit. In addition to letters, digits and spaces, special characters are allowed in the description.

Principal Investigator An email address to the Principal Investigator (PI). Note that this email address is not used for anything other than information at this time. It is not connected to any DDS account, and they receive no emails. This is due to the fact that some PIs are not involved in the data collection, and may be listed as the PI as a formality. Thus, the PI does not necessarily have to have a DDS account. If the PI should receive emails and have a DDS account with access to your project, we suggest that you set the same person as the *Project Owner.* This can be done in two ways:

1. Creating a project and then adding users to the project[14].
2. Creating a project and *at the same time* specifying which *Researchers* should have access and which user(s) should be set as an *owner*[15].

In addition to this, the following information is generated for each project:

Public ID The public ID is generated from the Internal Reference ID (Creating a Unit in the DDS) and a counter. Thus, each project public ID associated with a specific unit will have the same prefix. For example, if the Internal Reference ID for the associated unit is **intref**, the first project created for that unit would be **intref00001**, the second **intref00002**, and so on.

Bucket The name of the project-specific bucket with the Safespring project. This is generated from the public ID, the current time (Date Created below) and a randomly generated string of characters.

Date Created A timestamp representing the date and time when the project was created.

---

[14] https://scilifelabdatacentre.github.io/dds_cli/user/#dds-user-add

[15] https://scilifelabdatacentre.github.io/dds_cli/project/#dds-project-create

Date Updated A timestamp representing the date and time when the project was last updated, e.g. a folder or file was deleted.

Key Pair As described in the beginning of this section, a key pair is generated for each project. This key pair consists of a public and a private part. The public part ("project public key") is saved together with the project information and there is a single copy. The private part ("project private key") is encrypted with the user public key of each user with access to the project in question. Thus, there is one copy of the project private key per user.

Sensitive Information on whether the project is regarded as sensitive or not. If any sensitive data, such as (for example) human related data, will be delivered at any stage in a delivery project, the project should be set as sensitive. This is the default. There is an option to specify that the project can be regarded as non-sensitive, however at this time the handling of sensitive and non-sensitive project data are identical: all data is encrypted and decrypted. This may change in a future version.

Released Information on whether or not the project data has been released (project status set to "Available") at some point. For more information on the project statuses, please see the appendix (Project Statuses).

Active Whether or not the project is currently active. The default is active. The following statuses are regarded as active:

- In Progress
- Available
- Expired

The following statuses are regarded as non-active:

- Deleted
- Archived (incl. aborted)

2.7. Uploading data

Once a project has been created, the project status is automatically set to *In Progress*. This status (and this status only) allows for upload of data within the project[16]. Only *Unit Admins* and *Unit Personnel* have the permissions to upload data.

When starting an upload, a directory ("staging directory") is created by the executing command. The default location of the staging directory is the current working directory, however the user can specify an existing directory in which the staging directory should be placed. Independent of the location (specified or default), the staging directory is named *DataDelivery_<timestamp>_<project_id>*, where <timestamp> is the date and time when the upload was started, and <project_id> is the ID of the project the user is attempting to upload data to. If there is no data to upload, this directory is deleted immediately. If not, the staging directory will contain three subdirectories:

- **logs** The *logs* directory will be empty if the upload is successful. If an error occurs

[16] https://scilifelabdatacentre.github.io/dds_cli/data/#dds-data-put

during the upload, a log file called `"dds_failed_delivery.json"` will be created and placed in the *logs* directory. If the actual upload to the cloud is successful but the file information (e.g. name, checksum, size etc.) fails to be saved to the database, the log file lets the PO insert the missing file data into the database. Without this information, the uploaded file(s) will be located in the cloud but will not be accessible by any of the user roles since there is no record of it (/them) in the database. Thus, in the case of this log file being generated, contact the Data Centre immediately and attach the log file to the message. If you do not contact us with this information, the DDS cleanup actions[17] will:

- Remove the files found in the cloud which are missing database metadata
- Remove rows in the database which are missing the corresponding file in the cloud

- **files** This is the subdirectory where the actual staging occurs. Prior to the upload of a file, the DDS checks whether or not the file appears to be compressed by examining the file signature contained in the first few bytes of the file. If the file appears to be compressed, no compression is performed on the file. If the file appears to be in raw format, it is compressed by the DDS[18]. As mentioned previously, all files are also encrypted. When encrypting (and possibly compressing) the files, the new files ("staged files") are placed in the *files* directory. As soon as the file has been uploaded, the staged file is deleted. The original file is not affected.

- **meta** The *meta* directory will always be empty. This is a remnant of a previous version and has not yet been removed.

The staging process and upload is executed with several concurrent threads. This means that the DDS does not handle one file at a time. Instead, the default is for the DDS to stage and upload four files at a given time. However, this may vary (e.g. may be less) depending on the computer or server capabilities, and can also be changed by using one of the CLI options. Assuming that the default is used and that more than four files are included in the ongoing upload, there will be a maximum of four progress bars displayed in the terminal. The file names are displayed to the left of the bars, and to the left of the file names is the current ongoing process for each file. A lock indicates that a file is being compressed and encrypted, and an arrow pointing up indicates that the file is being uploaded to the bucket.

Uploading the same data (files or directories with the same names / paths) again will by default not succeed. However, the user can explicitly tell the DDS to upload the same data again - to overwrite the current file information in the database and to upload a new version of the files. This will generate a new file version in the database, which is kept track of for invoicing purposes.

---

[17] The cleanup actions will be executed once a week, manually by one of the sysadmins.

[18] Compression algorithm: ZStandard. See https://github.com/ScilifelabDataCentre/dds_web/blob/dev/doc/architecture/decisions/0006-use-zstandard-as-the-compression-algorithm.md for more information on why this was chosen as the compression algorithm.

Uploading data to the cloud requires collection[19] of the Safespring keys which provides the connection between the CLI and Safesprings cloud storage. The plan is to change this to use a similar method as what the download (Downloading data) uses - pre-signed urls. The plan is also to allow for the continuation of a failed upload. Currently, a failed upload requires the restart of the process. For a detailed flowchart describing the upload process, see the appendix Flowcharts.

2.8. Releasing the data

The project status *In Progress* does not allow for any data download by *Researchers*. In order to give the *Researchers* access to the data, the project needs to have the status *Available*[20]. The process of changing the project status from *In Progress* to *Available* is hereby referred to as "releasing" the data (/project). When the data is released, the countdown of the DiA[21] (Creating a Unit in the DDS) starts, and upload is no longer possible. In the case that some file has been missed during the upload, the *Unit Admin / Personnel* can change the project status from *Available* to *In Progress* ("retracting"). This lets the *Unit Admin / Personnel* continue uploading files and prevents any downloads by the *Researchers*. However, retracting the project does not pause the countdown of DiA. When DiA number of days have passed, the project status is automatically set to *Expired*. More information on this can be found in section Automatic expiry of data access.

Releasing the data also, by default, notifies the *Researchers* that there is data available for download. If the unit does not want these emails to be sent, it can be disabled when performing the status change.

2.9. Downloading data

Once the project has been released, the *Researchers* with access to the project can begin downloading the data. *Available* is the only status that allows for data download by *Researchers*, and upload and deletion of any project contents by *Unit Admin* and *Unit Personnel* is not possible. When downloading data, the *Researchers* can either choose to download specific file(s), specific folder(s), or the entire project contents.

As with the upload (Uploading data), a staging directory is created when downloading the data. This directory is placed by default in the current working directory, and is named *DataDelivery_<timestamp>_<project_id>.* However unlike the upload command, downloading allows the user to choose the name of the directory - specifying a *destination*. The destination cannot be an existing directory[22] - it must be a new directory. Since a new destination is required

[19] Collection of the Safespring keys is automatically handled by the CLI.

[20] https://scilifelabdatacentre.github.io/dds_cli/project/#dds-project-status-release

[21] When releasing a project, the default DiA is the value set during unit creation in the DDS. It is possible, however, to specify a custom DiA for a specific project by using the deadline option in the release command.

[22] This was originally to avoid any overwriting of existing files, however it has been brought to our attention

with every download, downloading the same file(s) multiple times is possible and is only limited by the amount of available storage space on the client (computer or server where the command is run).

The staging directory contains the same subdirectories as the upload creates, however the usage of them differs slightly.

- **logs** The *logs* directory will be empty if the download is successful. If an error occurs during the download, a log file called `"dds_failed_delivery.json"` will be created and placed in the *logs* directory. In the case of this log file being generated, contact the Data Centre and attach the log file to the message.
- **files** This is the subdirectory where the actual staging occurs. This is also the final destination of the downloaded and decrypted files. All downloaded files are saved to this location. Succeeding the download, all files are decrypted and those that were initially compressed by the DDS prior to upload are decompressed. As soon as a file has been decrypted and decompressed, the staged (downloaded) file is deleted from the *files* subdirectory. Downloading data does not affect any of the files in the cloud.
- **meta** As during upload, the *meta* directory will always be empty.

The download and decryption/decompression process is also executed with several concurrent threads. As for the upload, the download also has four as the default number of threads and this is adjustable in the same way. Assuming that the default is used and that more than four files are included in the ongoing download, there will be a maximum of four progress bars displayed in the terminal. The file names are displayed to the left of the bars, and to the left of the file names is the current ongoing process for each file. An arrow pointing up indicates that the file is being downloaded from the bucket, and a lock indicates that a file is being decrypted and decompressed.

In contrast to the upload, the download of data from the cloud does not require the collection of the Safespring keys. Instead, the download is handled with the use of pre-signed urls. These urls are generated and cryptographically signed within the DDS when a user is attempting to download the data. If the download of a file fails, the process must start from scratch again. For a detailed flowchart describing the download process, see the appendix Flowcharts.

2.10. Automatic expiry of data access and archiving of project

After DiA number of days, the DDS automatically changes the project status to *Expired*. This starts the countdown of the DiE number of days. In this status, all data and metadata is kept in the system, but all uploads, downloads and deletions of data by all roles are blocked. After DiE

---

that the ability to specify the same staging directory for several downloads would be a welcome feature. There is a plan to look into this.

days, the project status is automatically changed to *Archived*. This is the end of a project life cycle[23] and all files are deleted.

If *Researchers* associated with the project need access to download the data, they must contact the responsible SciLifeLab unit. A *Unit Admin* or *Unit Personnel* within the DDS can then change the project status to *Available* again, thereby allowing for download and resetting the countdown of the DiA. Similarly, if additional data should be delivered in the project, a *Unit Admin* or *Unit Personnel* must first change the project status to *Available* and from there to *In Progress*[24].

The renewal of data access can be performed a maximum of two times. Once a project expires, it can be released again, then expire again and then released one last time. After this, the project expires, and can then only be archived.

[23] The project is set to non-active.

[24] Note that it may be a good idea to use the –no-mail flag in this case to avoid *Researchers* receiving email notifications regarding available data.

3. What's the invoicing model?

The SciLifeLab Data Centre will cover the costs for the DDS if the cost of a SciLifeLab unit is below 5000 SEK per year. If the cost is above 5000 SEK per year, the SciLifeLab unit will be invoiced. This invoice model will be in effect from 2022-10-01. The system will periodically collect usage information per project for each unit and the SciLifeLab will invoice quarterly.

The storage usage is counted in gigabyte-hours (GBHours). This is the number of hours that one GB has been stored (GB x hours). The current cost is 9 öre per GB per month, thus one GB stored for one month would cost 9 öre.

Example

Data usage: 10 GB
Storage time: One year (24 hours x 365 days)
GBHours: 10 GB x 24 hours x 365 days = 87600 GBHours.

Cost per GB per month: 9 öre = 0.09 kr
Cost per GBHour: 9 öre / (24 hours x 30 days)

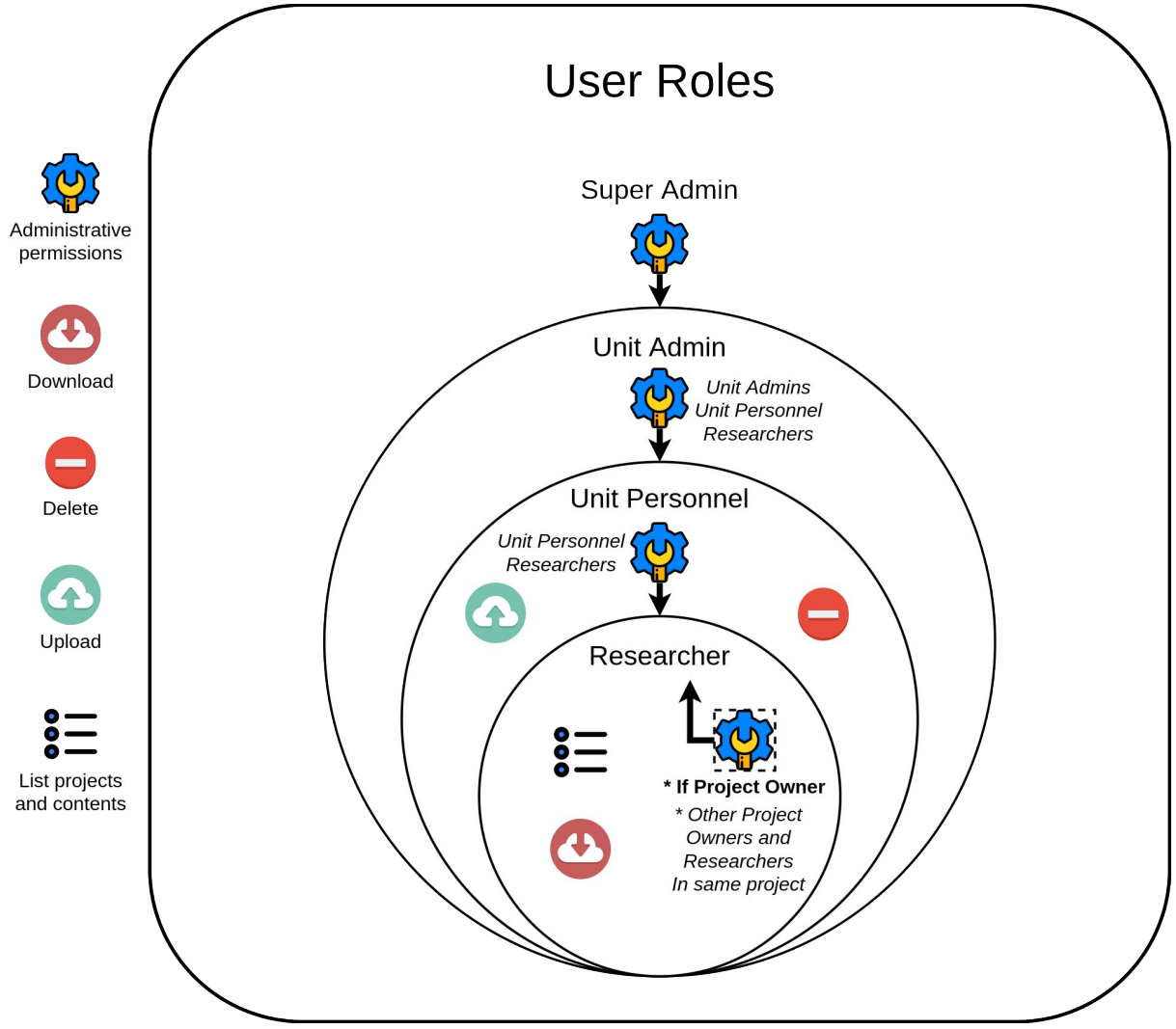
Cost for 10 GB stored for one year straight: $\frac{87600\ GBHours \times 0.09\ kr}{24\ hours\ x\ 30\ days} =$

Appendix

A. User Roles

There are four different roles in the DDS: “Super Admin”, “Unit Admin”, “Unit Personnel” and “Researcher”. The figure below shows a *simplification* of the permission hierarchy of these roles:

- *Researchers* can list projects and their contents and download data, but only have administrative permissions if they are set as an owner of a specific project.
- *Unit Personnel* can list projects and their contents, download data, upload data and delete it. They also have administrative permissions regarding other *Unit Personnel* and *Researchers*.
- *Unit Admins* can list projects, their contents, download data, upload data and delete data. They have administrative permissions regarding other *Unit Admins* and also *Unit Personnel* and *Researchers*.
- *Super Admins* do not have access to any data but have administrative permissions regarding all other roles, including other *Super Admins*.



21 September 2022

Super Admin

The *Super Admin* role is reserved for certain employees of the SciLifeLab Data Centre.

The table below shows which permissions a *Super Admin* has (“Yes”) and does not have (“No”). * indicates that the permissions relate to all roles. If a role has a permission relating to one or more specific roles, these roles are listed below the permission.

Type of permission	Yes	No
Delivery related	● List all projects *Note, however, that a Super Admin does not have access to the projects and can therefore not access any data uploaded within it. Also, when the functionality to display detailed information about individual projects has been implemented, Super Admins will not be able to use that functionality. The Super Admins can only see the information displayed with the project listing functionality. At this time, this is for the purpose of assisting with user queries. It will be investigated and possibly changed.*	● Create projects ● List project contents ● Download data ● Upload data ● Delete data ● Change project statuses
Administrative	● Create units (Creating a Unit in the DDS) ● List units ● List users ● Invite users * ● Activate and deactivate users * ● Delete users *	

21 September 2022

Unit Admin

The table below shows which permissions a *Unit Admin* has (“Yes”) and does not have (“No”). * indicates that the permissions relate to all roles. If a role has a permission relating to one or more specific roles, these roles are listed below the permission.

Type of permission	Yes	No
Delivery related	● Create projects ● List projects ● List project contents ● Upload data *Only when project status is In Progress.* ● Download data ● Delete data *Only when project status is In Progress.*	
Administrative	● Invite users: ○ *Unit Admins* ○ *Unit Personnel* ○ *Researchers* ● Add *Researchers* to specific projects, including *Project Owners* ● Grant, revoke and fix users access to projects when access has been lost ○ *Unit Admins* ○ *Unit Personnel* ○ *Researchers* ● Activate and deactivate users: ○ *Unit Admins* ○ *Unit Personnel* ● Delete users: ○ *Unit Admins* ○ *Unit Personnel*	● Any actions pertaining to *Super Admins* ● Activate and deactivate *Researchers* *This is due to the fact that Researchers can be involved in the projects of multiple different units. If a Unit Admin would like to deactivate a Researcher, please contact the PO (also Super Admin).* ● Delete *Researchers* *Same reasoning as for activating and deactivating Researchers, as described above.*

21 September 2022

Unit Personnel

The table below shows which permissions a *Unit Personnel* has (“Yes”) and does not have (“No”). * indicates that the permissions relate to all roles. If a role has a permission relating to one or more specific roles, these roles are listed below the permission.

Type of permission	Yes	No
Delivery related	● Create projects ● List projects ● List project contents ● Upload data *Only when project status is In Progress.* ● Download data ● Delete data *Only when project status is In Progress.*	
Administrative	● Invite users: ○ *Unit Personnel* ○ *Researchers* ● Add *Researchers* to specific projects, including *Project Owners* ● Grant, revoke and fix users access to projects when access has been lost ○ *Unit Personnel* ○ *Researchers*	● Any actions pertaining to *Super Admins* and *Unit Admins* ● Activate and deactivate users ● Delete users

Researcher

The *Researcher* role represents the data recipient. *Project Owner* is a subcategory of the *Researcher* role, and is intended for those *Researchers* that should have some administrative permissions within specific projects. Thus, the *Project Owner* is not a DDS user role, but a "role" within a certain project. The first table below reflects the permissions of *Researchers* in general. The second table reflects the permissions of *Researches* which have been marked as *Project Owners* within a specific project.

The table below shows which permissions a ***Researcher*** has ("Yes") and does not have ("No"). * indicates that the permissions relate to all roles. If a role has a permission relating to one or more specific roles, these roles are listed below the permission.

Type of permission	Yes	No
Delivery related	• List projects *Not all projects, only those the user has been associated with by a Unit Admin, Unit Personnel or Project Owner.* • List project contents (uploaded data) *Only when project status is Available* • Download data *Only when project status is Available.*	• Create projects • Upload data *Only when project status is In Progress.* • Delete data *Only when project status is In Progress.*
Administrative	*Researchers have **no** administrative permissions.*	

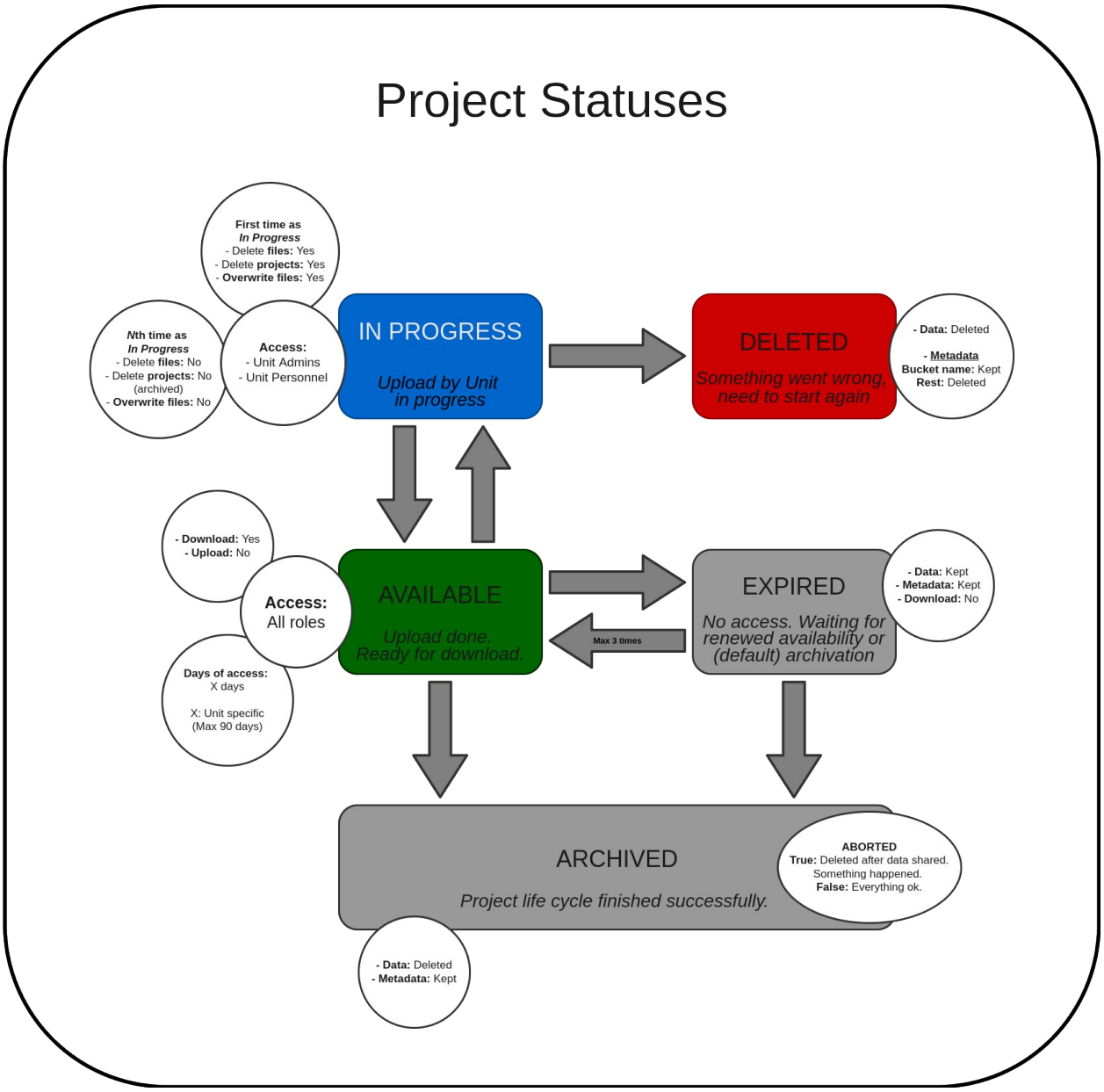
The table below shows which permissions a ***Project Owner*** has ("Yes") and does not have ("No") ***in a specific project***. * indicates that the permissions relate to all roles. If a role has a permission relating to one or more specific roles, these roles are listed below the permission.

Type of permission	Yes	No
Delivery related	• List project contents (uploaded data) *Only when project status is Available* • Download data *Only when project status is Available.*	• Create projects • Upload data *Only when project status is In Progress.* • Delete data *Only when project status is In Progress.*
Administrative	• Invite and add *Researchers* (including other *Project Owners*) to project • Grant, revoke and fix	• Any actions pertaining to *Super Admins, Unit Admins* and *Unit Personnel* • Delete users *

Type of permission	Yes	No
	Researchers (including other *Project Owners*) access to projects when access has been lost	• Activate and deactivate users * • Add or invite users to other projects

B. Project Statuses

There are five different possible project statuses available within the DDS. The section "How is it used?" describes the standard project life cycle, from creating a project with the automatic status *In Progress*, to project archivation. The figure below shows a summary of the different statuses and what they mean. The rest of this section describes them in more detail.



In Progress

A project is automatically set to *In Progress* when it's created. This status is to allow upload by *Unit Admins* and *Unit Personnel* and indicate that the data delivery is in progress. If invited to a project during this stage, *Researchers* can also see the project in their list of projects, however they cannot see if anything has been uploaded.

The permissions of *Unit Admins* and *Unit Personnel* vary depending on whether or not the project has just been created (First time as In Progress), or if it has been released and later retracted one or more times (Nth time as In Progress).

First time as In Progress

The first time as *In Progress*, data can be uploaded, downloaded, deleted and overwritten. The project can also be *Deleted*, indicating that something has gone wrong before the data was released to the recipients.

*N*th time as In Progress

If the project has been made *Available* and then retracted to *In Progress* again, deleting the project is no longer an option. In this case, the *Unit Admin* or *Unit Personnel* will need to archive the project and indicate that the project is aborted[25]. Deleting and overwriting files is also not an option at this stage. This is a design choice and reduces the risk of *Researchers* downloading incomplete or changed data.

Deleted

A project can only be deleted if it has the status *In Progress* and it has not been previously released to the recipients. The *Deleted* status is to indicate that something has gone wrong, e.g. project information is incorrect.

When a project is deleted, all data is deleted from the cloud. In addition, all project metadata aside from the bucket name is deleted from the database. The name of the bucket is required for invoicing purposes and keeping track of that there are no residual files left after deletion. The deletion of a project is not reversible - once a project is deleted, it cannot be used for delivery again.

Available

When a project is released, all roles (aside from *Super Admins*) have access to the data. *Researchers* can now download the project data. Upload is no longer possible. If additional data

[25] https://scilifelabdatacentre.github.io/dds_cli/project/#dds-project-status-archive

must be uploaded within the same project, a *Unit Admin* or *Unit Personnel* can retract the project, thereby setting the status to *Available* again.

An *Available* project cannot be deleted. If the project should be closed, a *Unit Admin* or *Unit Personnel* can do this by archiving the project. To indicate that something has gone wrong and that the project life cycle has deviated from the standard, use the abort flag. If nothing has gone wrong and the project should simply be closed due to that the data has been downloaded by the users, the archivation command can be used as is.

When a project is released (from *In Progress*), the countdown of DiA begins. After this number of days have passed, the project is automatically expired.

Expired

The project status *Expired* can only occur if DiA days pass - a project cannot be expired via the CLI. When the status is changed to *Expired*, the countdown of the DiE number of days starts. In this status, all data and metadata is kept in the system, but all uploads, downloads and deletions of data by all roles are blocked. After DiE days, the project status is automatically changed to *Archived*. This is the end of a project life cycle and all files are deleted. Metadata, however, is kept.

In the case that the data recipients do not download the data in time, they can contact one of the *Unit Admins* or *Unit Personnel* connected to the unit responsible for the project. These roles then have the necessary permissions to release the project data again. Once released again, this starts the countdown of DiA again.

The renewal of data access can be performed a maximum of two times. Once a project expires, it can be released again, then expire again and then released one last time. After this, the project expires, and can then only be archived.

Archived

An archived project marks the end of a project life cycle. This is not reversible. All data is deleted and all metadata is kept.

Aborted

An archived project can also be marked as aborted. This indicates that data has been (either intentionally or accidentally) shared with the recipients and that something has gone wrong during the data delivery. It can also indicate that the *Unit Admin* or *Unit Personnel* wishes for the project metadata to be deleted, which it is not if the project is simply archived.

21 September 2022

C. Keys

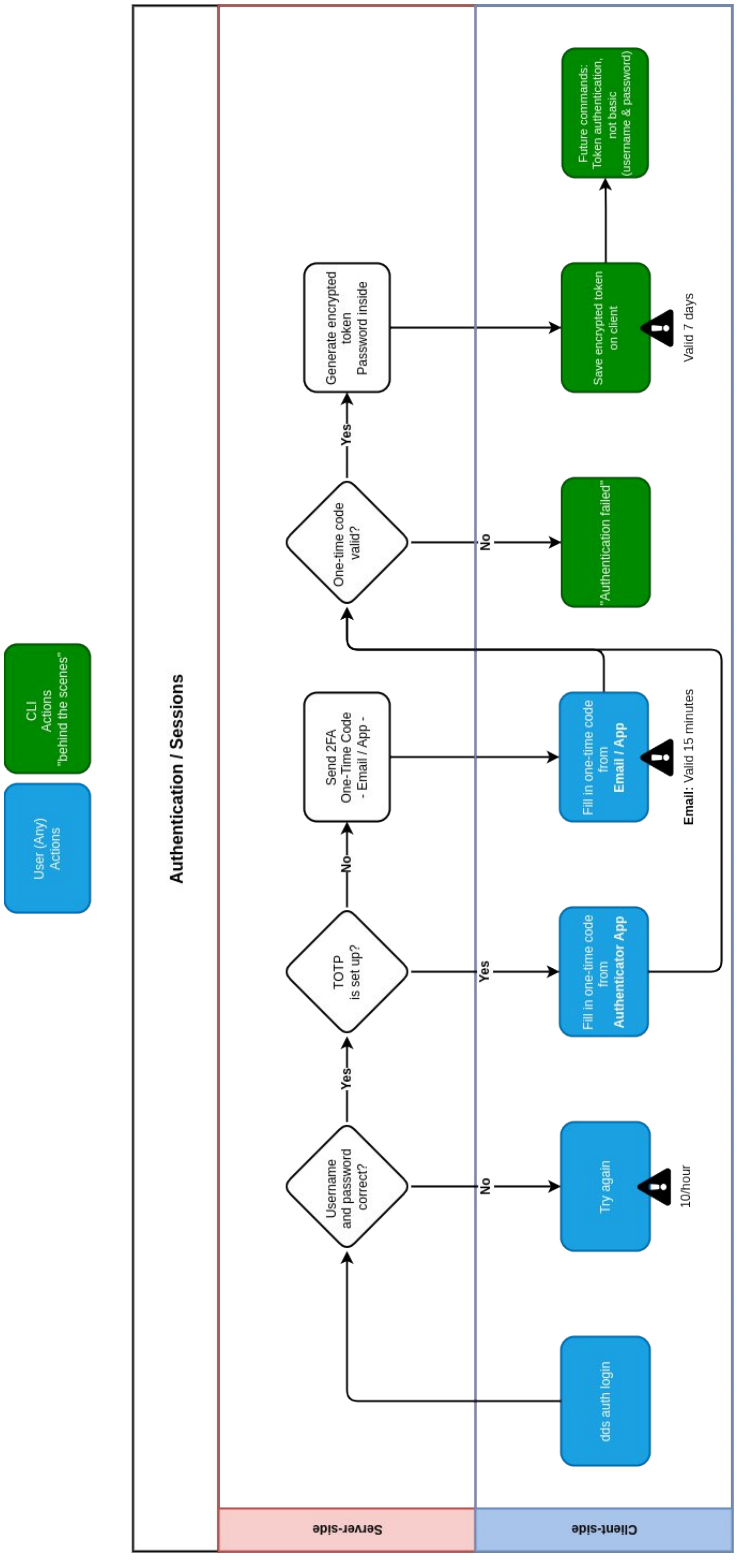
Information on the user- and project keys are coming.

D. Flowcharts

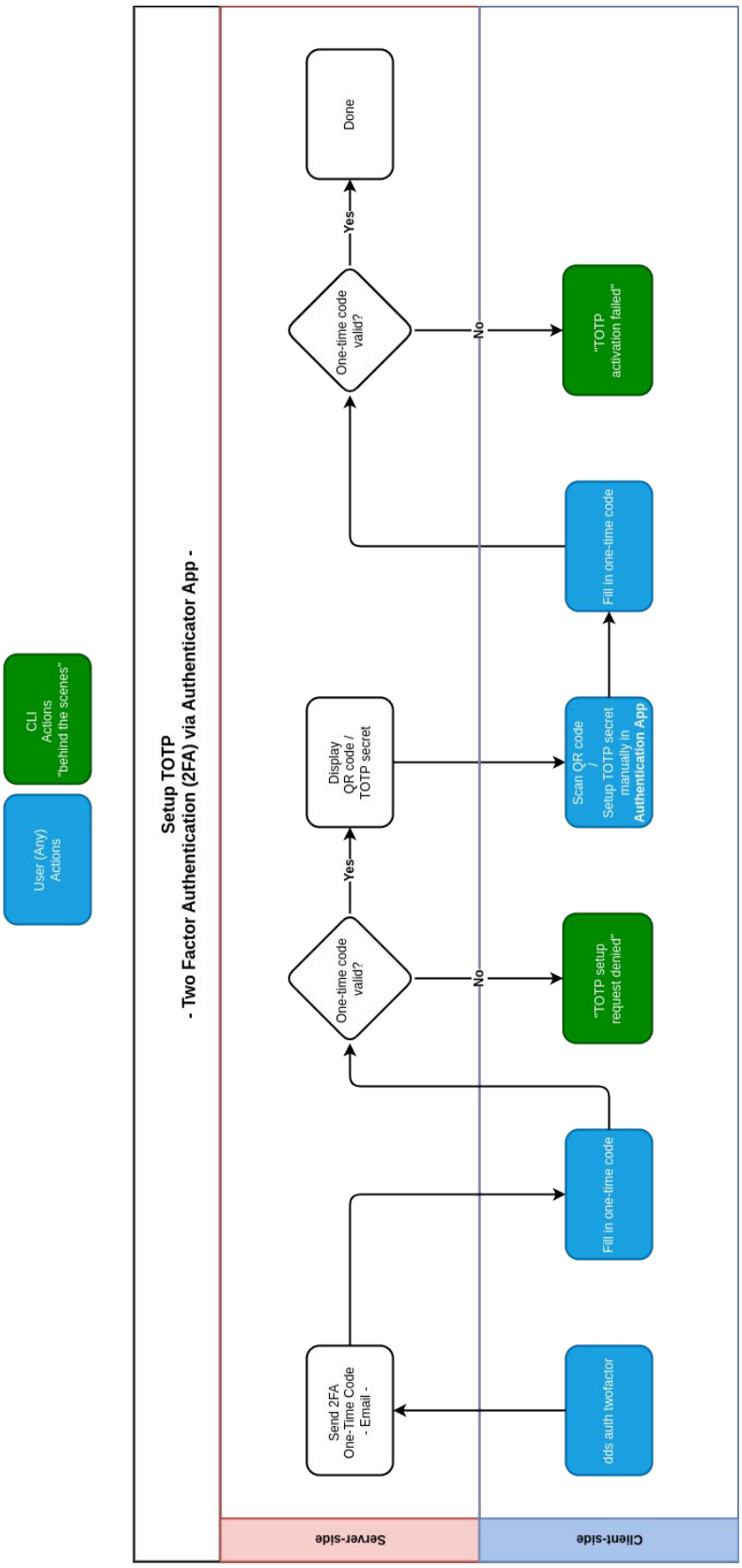
This section shows flowcharts describing the DDS. The flowcharts are of varying sizes and some can therefore be difficult to follow and read. To see these in a better resolution, please go to the page linked below. Note that after clicking the link, you need to click the “open with”, “diagrams.net” and then authorize with google to access it. The flowcharts are located in the tab “Flowcharts”, found at the bottom of the board. These will most likely be available in a different format later, e.g. not requiring access to the diagrams.net page.

https://drive.google.com/file/d/1ophR0vtGByHxPG90mzjAPXgMTCjVcN_Z/view?usp=sharing

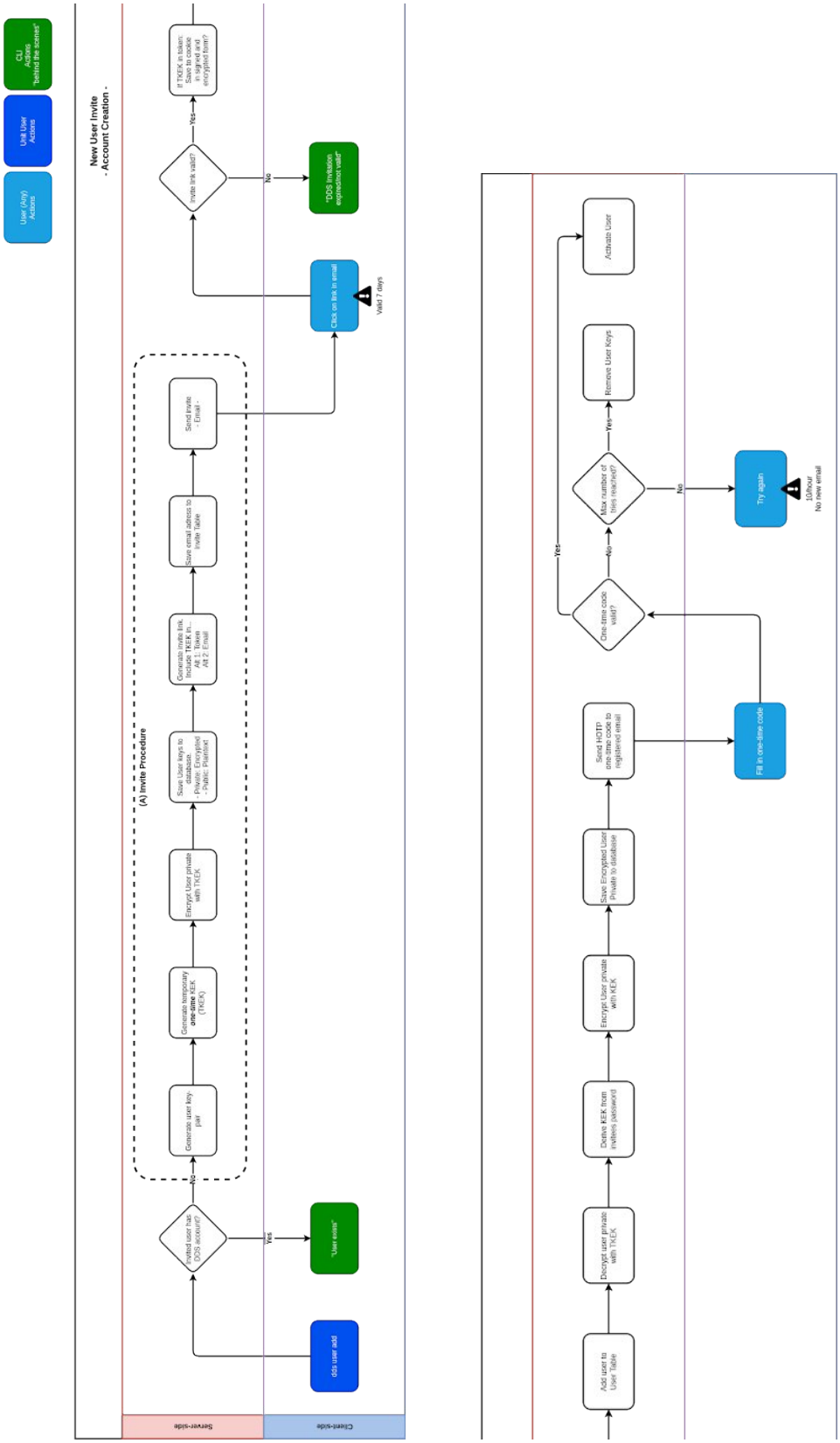
Authentication



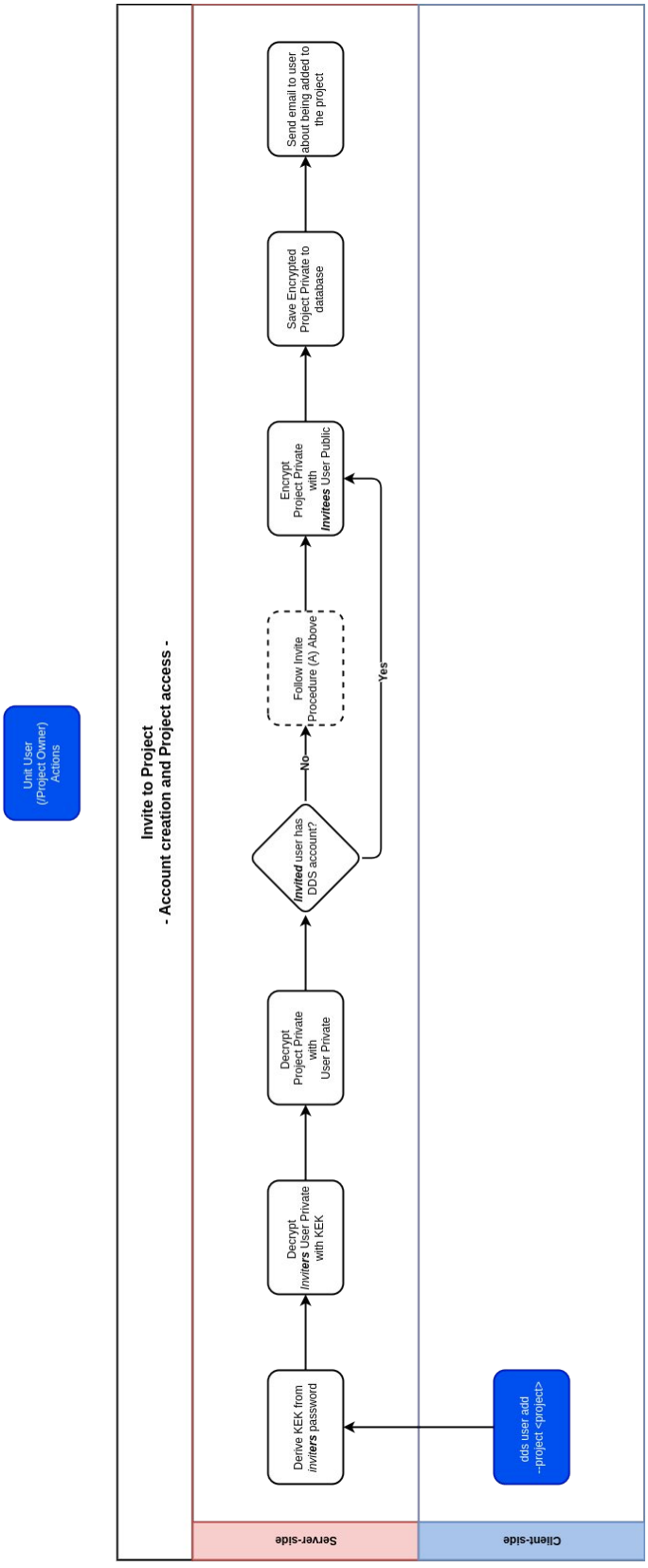
Setup of 2FA via Authentication App



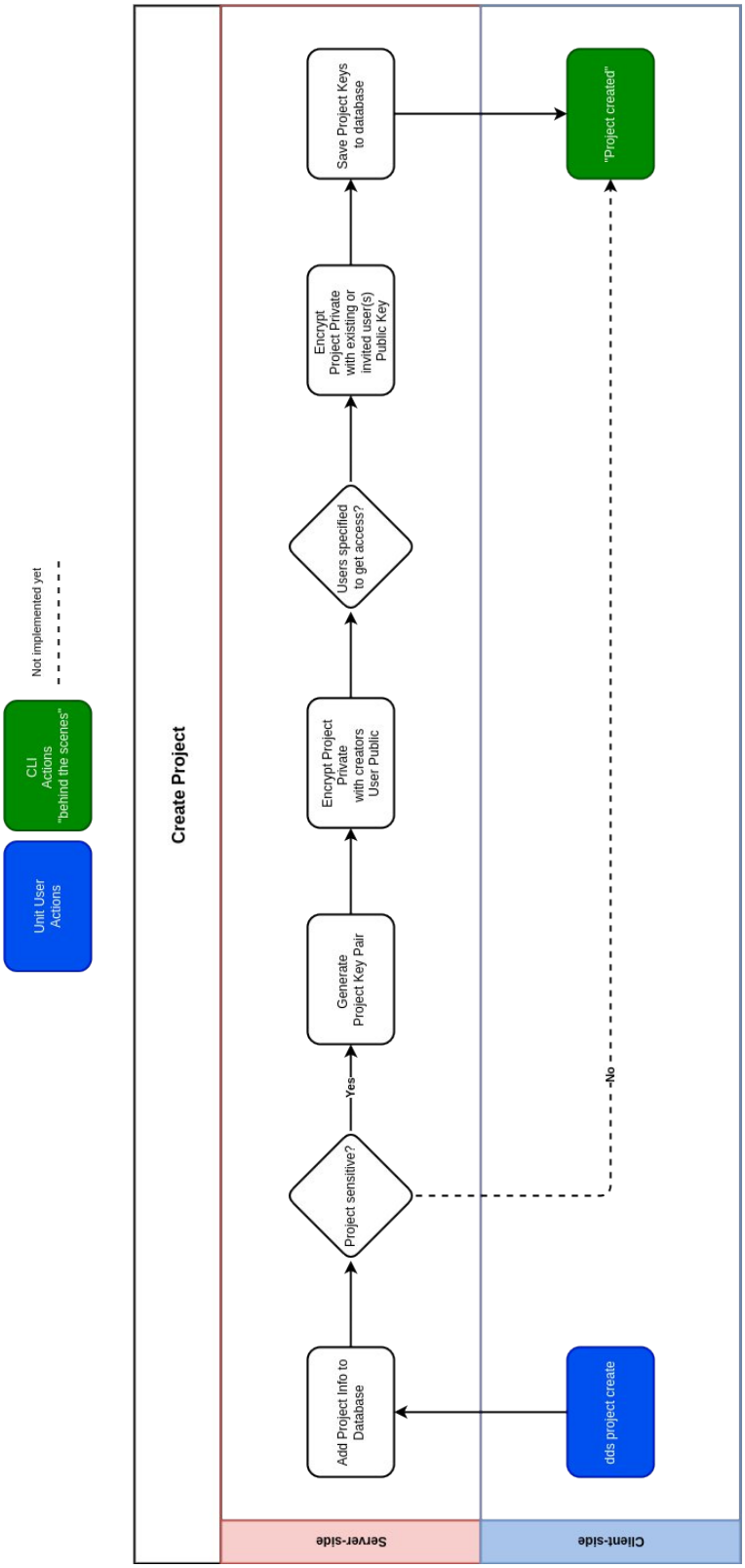
Inviting users



Inviting users to Project

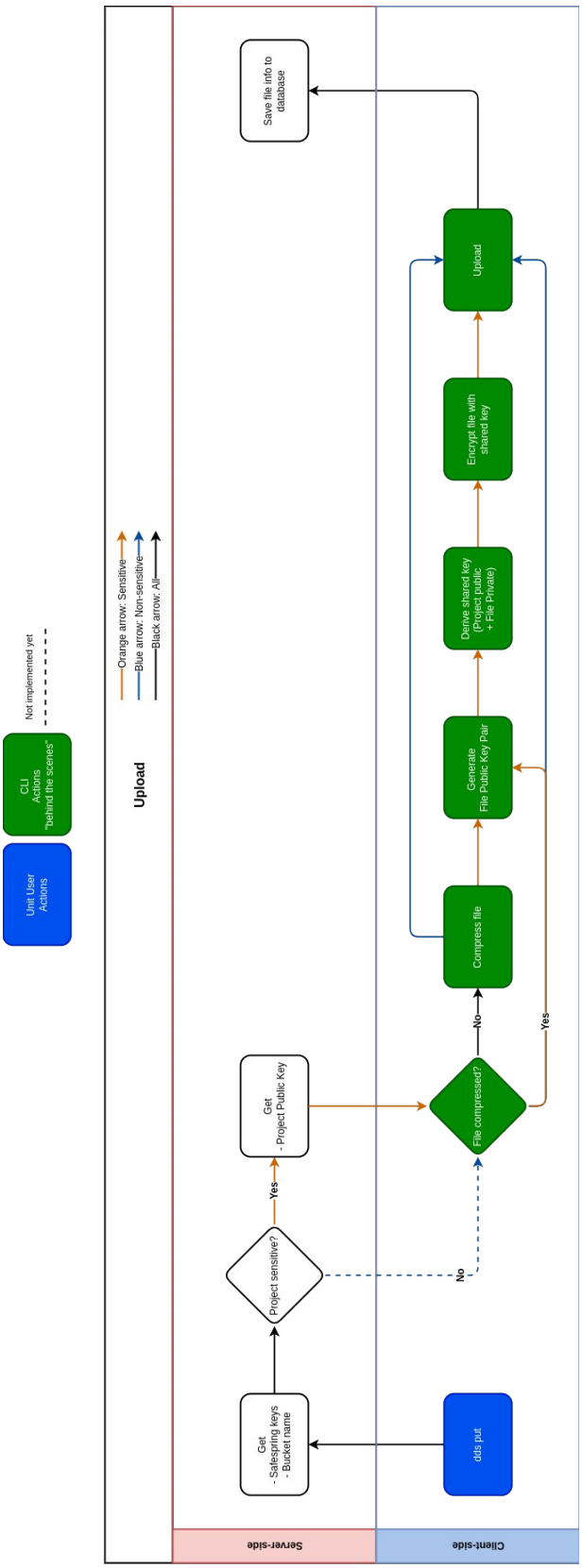


Creating a Project


Create Project
Client-side
Server-side
dds project create
Add Project Info to Database
Project sensitive?
Yes
No
Generate Project Key Pair
Encrypt Project Private with creators User Public
Users specified to get access?
Encrypt Project Private with existing or invited user(s) Public Key
Save Project Keys to database
"Project created"
Unit User Actions
CLI Actions "behind the scenes"
Not implemented yet

21 September 2022

Uploading data



21 September 2022

Downloading data