2023

Gaia-X and Future Mobility Alliance

ONBOARDING PROCESS

Contents

1 Onboarding Process with Participant agent and CLI

This document explains how you can onboard to a test Gaia-X environment as well as the Future Mobility Alliance.

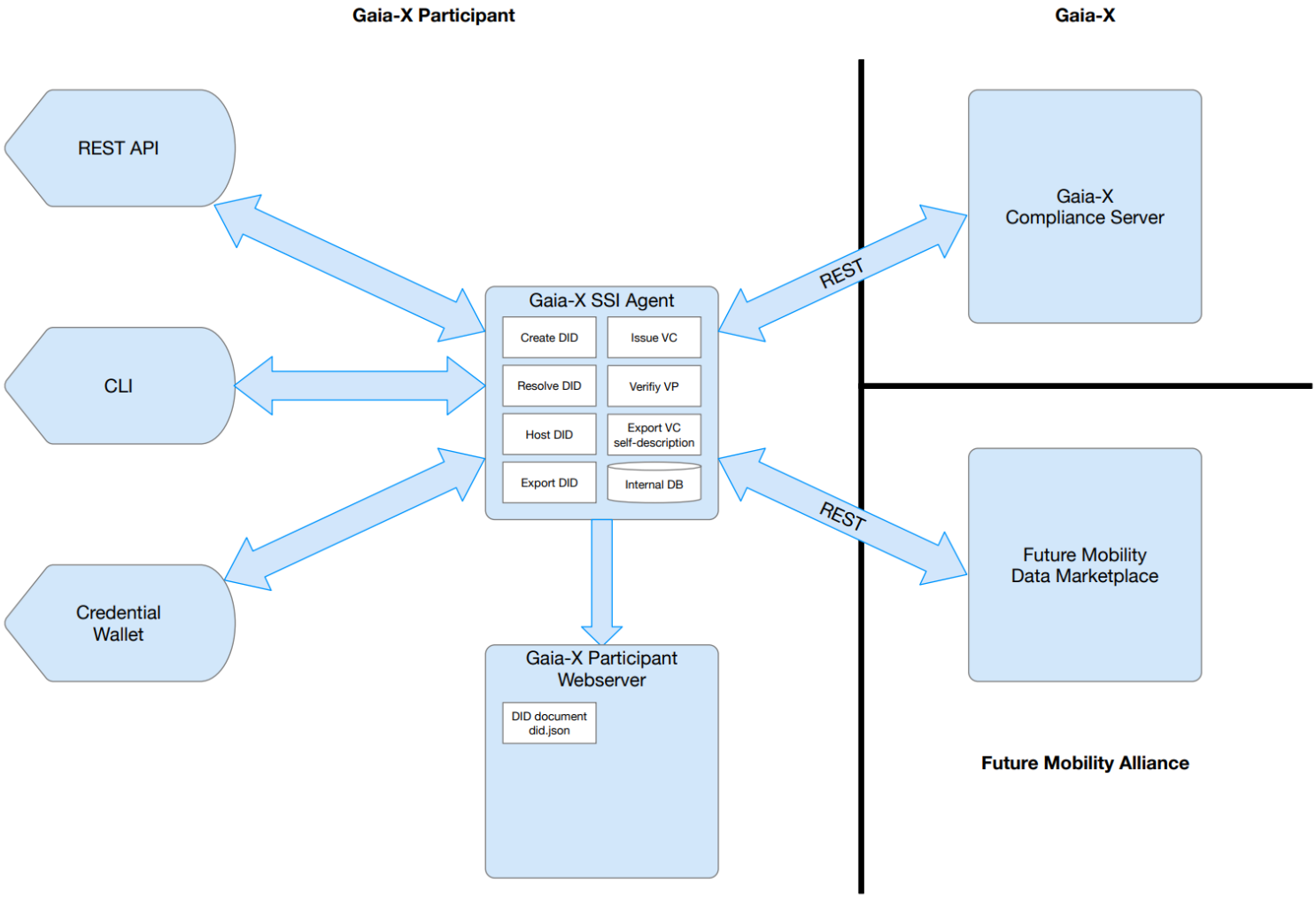
1.1 Participant Agent and CLI

The Gaia-X agent is an agent that can create Verifiable Credentials and Presentations out of Gaia-X self-descriptions. It can submit these to the Gaia-X Compliance Server, to get back Compliance Credentials. The agent can also issue and verify generic Verifiable Credentials and Presentations. Lastly the agent can export well-known resources, like DID:web and X.509 Certificate chains needed in a Gaia-X context.

The Agent can be deployed and used in multiple scenarios:

- As a Command Line tool (CLI)
- As a REST API
- Directly integrated into a typescript and/or React-Native application

Below you can see the possible interactions with the agent, as well as the connection to other components. This document is about using the Agent CLI, as that currently is the only documented way to setup the Agent.



2 Prerequisites and Installing the Gaia-X Participant Agent and CLI

The Agent Command Line (CLI) interface is currently the only documented way to work the Gaia-X agent. Soon a web based application and mobile wallet will be available, making these tasks simpler. In order to use the CLI you have to make sure all the prerequisites and tasks defined in this chapter are met.

2.1 Create X.509 keys and get SSL certificate

You will first need to have an existing X.509 EV SSL certificate or create a new one. Appendix A explains how to setup a new X.509 certificate. Without following the steps in the appendix you cannot be onboarded as Gaia-X participant!`

2.2 Install the CLI

We are working on create a binary release of the CLI tool, meaning all you have to do is download a single file and you are good to go. As long as this release is not available yet, you will have to follow the process outlined in this chapter.

2.2.1 NodeJS version 16

Please download NodeJS version 16. You can find NodeJS for your computer on the following page: https://nodejs.org/en/blog/release/v16.16.0/ Follow the installation instructions on the NodeJS website

2.2.2 Install the Gaia-X agent CLI tool

After installing NodeJS open a terminal window (linux, Mac OS X) or command prompt (Windows) on your computer. Ideally with elevated permissions. Type in the following command:

```
npm install -g @sphereon/gx-agent-cli --no-audit
```

If you are familiar with NodeJS and are using yarn instead of npm (required you to install yarn first separately from NodeJS), you could run the following command instead of the above command:

```
yarn global add @sphereon/gx-agent-cli
```

Check that the CLI is available to you. Open a new terminal or command prompt on your computer and type **command (documentation)**:

```
gx-agent --help
```

2.2.3 Create the Gaia-X agent configuration file

Creates the agent.yml agent configuration file. It has one option, which is not mandatory. The -l/--location option can have a value of cwd, meaning the agent.yml file will be written to current working directory, or home, meaning the agent.yml file will be written to the .gx-agent directory in your user home-directory. If no option is provided, home will be assumed. Please be aware that if you choose “cwd”, you will have to run the CLI commands in the future from this same directory. Otherwise the agent.yml file will not be found.

Command (documentation):

```
gx-agent config create
```

You can now open the configuration file in a text-editor in the location that is mentioned in the output of the above command.

3 Creating an identity

You first will have to create an identity using the certificate mentioned in the previous chapter. These identities are created using so called W3C conformant Decentralized Identifiers (DID). A DID has the form of: did:method:method-specific-id

For Gaia-X, the so called did:web method is being used, meaning DIDs associated with your domain name like did:web:example.com are hosting a so called DID document at a well known location (https://example.com/.well-known/did.json). The DID Document will list at least the X.509 Certificate public key generated from the public certificate you received in the previous chapter.

3.1 Creating a DID document

The DID document is responsible for listing public keys associated with the DID and your organizational domain. This DID is used to sign Gaia-X self-descriptions and so called Verifiable Credentials. This allows others to determine that data is authentic and not manipulated, originating from your organization and associated with your domain.

Command (documentation):

```
gx-agent did create --private-key-file=privkey.pem  --cert-file=cert.pem --ca-chain-file=cacerts.pem --domain=example.com
```

The above command creates a new DID Document, with the privkey.pem file as private Key input, the cert.pem file as public certificate input and cacerts.pem as the Certificate Chain input. Lastly you need to pass in the domain name that will host the DID document. This domain name needs to be exactly the same as the CN input parameter of the X.509 Certificate!

The DID Document is now created. You will export it later. If you do want to have a quick peek, how the DID document looks like you could use the following command to resolve the DID Document directly from the agent

Command (documentation)

```
gx-agent did resolve did:web:example.com -l
```

Please note that example.com has to correspond with the –domain value you supplied during DID creation. As long as you only create 1 DID method in the agent, you can also omit the DID for the resolve command, as the agent will automatically select the DID. The same is true for any future commands. If you however create more DIDs from the agent, you will have to supply a -d/--did option to most commands, to indicate the DID that should be used.

Note: If you just created the DID document in the agent, you will have to provide the -l/--local-only option to ask the agent to resolve the DID document locally in the agent. This has to do with the fact that you will have to export the DID document and host it on the webserver associated with your

provided domain (example.com in this case). See the next section about exporting and hosting the DID document.

3.2 Export the DID document and host it on your domain

The agent can issue and verify credentials where you are the issuer of the credential, even without the DID Document being hosted on your domain. However any other party expects the DID document to be found on your webserver in a well-known location. Meaning https://example.com/.well-known/did.json

That DID document also links towards the X.509 Certificate Authority Chain. So you will need to host both files on your domain. The agent supports an export method and in the future could also host the files, allowing you to run the agent on your webserver. For now the export is the only option.

Command (documentation):

```
gx-agent did export
```

By default the DID document and the CA certificate chain will be exported to the folder ***exported/example.com*** in the ***.well-known*** hidden directory. You need to copy that .well-known folder to your webserver at example.com. If your server already has a .well-known directory, you should copy only the contents of the exported/example.com/.well-known directory into the webserver directory. Do not rename the file-names, nor any location in the well-known directory!

How you upload or host these files on your domain's webserver, is highly webserver and/or Internet Provider specific. Please consult a technical person if you do not know how to perform this task.

Note: The DID document should be hosted on a TLS secured website, meaning only https:// is supported. This is a requirement of the did:web specification!

After the files have been copied to the webserver, you should be able to resolve the DID into a DID document, as should others. You can directly access the URL https://example.com/.well-known/did.json with a browser, or you can use the agent to resolve this DID. Please note that this time you will not be using the -l/--local-only option, as you want to resolve the DID using your internet connection this time.

Command (documentation)

```
gx-agent did resolve did:web:example.com
```

Note: The output should be a DID document. Do not continue the process, if you get errors, as external parties like the Compliance Service need to be able to resolve this DID!

4 Onboarding as Gaia-X participant

You first need to become a Gaia-X compliant participant. In order to do so, you first need to create as self-description. This is a so called Credential in a specific form. You will need to sign this self-description, using your DID, making it a Verifiable Credential. The compliance service will issue an attestation, in the form of a Participant Credential, signed by the Compliance Server DID. This allows you to prove to others that you are a Gaia-X participant.

4.1 Export an example participant-input-description.json to disk

The command below exports the example participant input self-description to disk in the current directory. The file will be called participant-input-description.json. Please note that if you run this command again, it will overwrite the file on disk. So if you make modifications to the file, we suggest to rename the file.

Command (documentation)

```
gx-agent participant sd export-example
```

4.2 Update the participant-input-description.json input file with your information

Update the participant-input-description.json input file. Please make sure to only update the values. Leave all the keys and structure intact!. In the example below all the bold parts have been updated to reflect the organizational values.

Note: In the future this would be a wizard to fill out during the onboarding process

Example:

```
{
  "@context": ["https://www.w3.org/2018/credentials/v1",
"https://registry.gaia-x.eu/v2206/api/shape"],
  "type": ["VerifiableCredential", "LegalPerson"],
  "id": "https://example.com/.well-known/participant.json",
  "issuer": "did:web:example.com",
  "issuanceDate": "2022-09-23T23:23:23.235Z",
  "credentialSubject": {
    "id": "did:web:example.com",
    "gx-participant:name": "Example Org",
    "gx-participant:legalName": "Example Org B.V.",
    "gx-participant:registrationNumber": {
      "gx-participant:registrationNumberType": "local",
      "gx-participant:registrationNumberNumber": "93056589"
    },
    {
        "gx-participant:registrationNumberType": "leiCode",
        "gx-participant:registrationNumberNumber": "9695007586GCAKPYJ703"
    },
    "gx-participant:headquarterAddress": {
      "gx-participant:addressCountryCode": "NL",
      "gx-participant:addressCode": "NL-NLD",
      "gx-participant:streetAddress": "Kerkstraat 10",
      "gx-participant:postalCode": "4007 AB"
    },
```

```
    "gx-participant:legalAddress": {
      "gx-participant:addressCountryCode": "NL",
      "gx-participant:addressCode": "NL-NLD",
      "gx-participant:streetAddress": "Kerkstraat 10",
      "gx-participant:postalCode": "4007 AB"
    },
    "gx-participant:termsAndConditions":
"70c1d713215f95191a11d38fe2341faed27d19e083917bc8732ca4fea4976700"
  }
}
```

4.3 Submit the self-description and obtain a Gaia-X compliance credential

This next command verifies the self-description, then internally creates a signed Verifiable Credential for the Self-Description and stores it in the agent. It then creates a so called Verifiable Presentation and submits it to the Gaia-X compliance server. The compliance server then returns the compliance Credential, which again is stored in the agent.

The below command submits the input file using the -sif/--sd-input-file option. If for whatever reason the Compliance Service is not available or there is a problem with your internet connection, you can list the participant self-description, to get the ID of the self-description from the agent. Then you can submit the compliance credential by id. In this case it looks up the already created credential in the agent, and submits it to the compliance server.

Command (documentation):

```
gx-agent participant compliance submit -sif=participant-input-description.json
```

5 Onboarding with Future Mobility Alliance

5.1 Add ecosystem

In order to prove membership and have service offering self-description compliance credentials, you will need to configure the Future Mobility Ecosystem first in the Gaia-X agent.

Depending on your agent.yml configuration file, you either already have the Future Mobility Alliance ecosystem available or not. In order to check it you can run the following command.

```
gx-agent ecosystem list
```

If the output does not list an entry called “FMA”, you will need to create the FMA ecosystem configuration. The below command will create an entry for the test FMA ecosystem

```
gx-agent ecosystem add FMA https://test.fma.sphereon.com -d "Future Mobility Alliance TEST"
```

5.2 Onboard as Future Mobility Alliance Member

Now let’s onboard with the new ecosystem.

gx-agent ecosystem submit FMA --sd-id=<abcd> --compliance-id=<efgh>

6 Issue Labels

An organization issuing labels, would ideally receive a Verifiable Presentation from the subject organization, which is also a member of the FMA. However the organization can and typically will have it's own process to determine whether an FMA participant is entitled to a certain label.

6.1 Example ISO certificate

The below is an example Credential, which can be issued as a Verifiable Credential to a FMA member organization.

```
{
"@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://sphereon-opensource.github.io/vc-contexts/fma/iso/iso-v1.jsonld"
  ],
  "type": ["VerifiableCredential", "ISOCertificate"],
  "id": "1674487178632",
  "issuanceDate": "2023-01-23T15:19:38.632Z",
  "credentialSubject": {
    "id": "did:web:nk-gx-agent.eu.ngrok.io",
    "initialApproval": "2022-05-30T09:30:10Z",
    "issuanceDate": "2023-01-01T08:00:10Z",
    "validUntil": "2025-12-31T23:59:59Z",
    "certificateNo": "1233456789",
    "standardId": "ISO/IEC 27001:2013",
    "organizationName": "Example BV"
  }
}
```

Make sure to store the input credential containing the appropriate information on the file system. For example: example-iso-certificate.json

6.2 Issue ISO Certificate Label

Issuing labels can be accomplished by using the generic Verifiable Credential CLI commands. For instance to issue a signed Verifiable Credential you can use the above example input from a file.

The optional -p/--persist option, allows you to store the Verifiable Credential in the agent.

```
gx-agent vc issue -f ./example-iso-certificate.json
```

6.3 Verify ISO Certificate Label

Verifying labels can be accomplished by using the generic Verifiable Credential CLI commands. For instance to verify an ISO label you can use a file containing a VC as input, or you can supply an ID of a Verifiable Credential stored in the agent.

```
gx-agent vc verify -f ./iso-certificate-vc.json
```

```
gx-agent vc verify -id 123456789
```

7 Appendix A - X.509 - SSL Certificate Creation

In case the organization that wants to participate in FMA does not have an SSL certificate with access to the private key yet, then the first thing to do would be to either purchase a certificate or using for instance let's encrypt to get one for free. Do not request or create a new certificate if an existing certificate is already present for a particular domain. If this is the case there are 2 options:

- Acquire the private and public keys/certificates from the existing certificate
- Create a subdomain eg. 'subdomain.example.com', that you will use for the certificate. Creating (sub)domains is out of scope for this guide.

If you do have an existing certificate you can skip this whole chapter.

When creating a new Certificate there are 2 options currently.

- Create a Certificate Signing Request yourself and purchase a certificate with any party providing SSL certificate.
- Generate a certificate using Letsencrypt

7.1 Create a CSR and purchase a certificate

The first step is creating the private key together with a Certificate Signing Request (CSR). This CSR is sent to the Certificate Authority, allowing them to create the public Certificate part belonging to the private key you are generating.

7.1.1 Install OpenSSL on your computer

<Install instructions here>

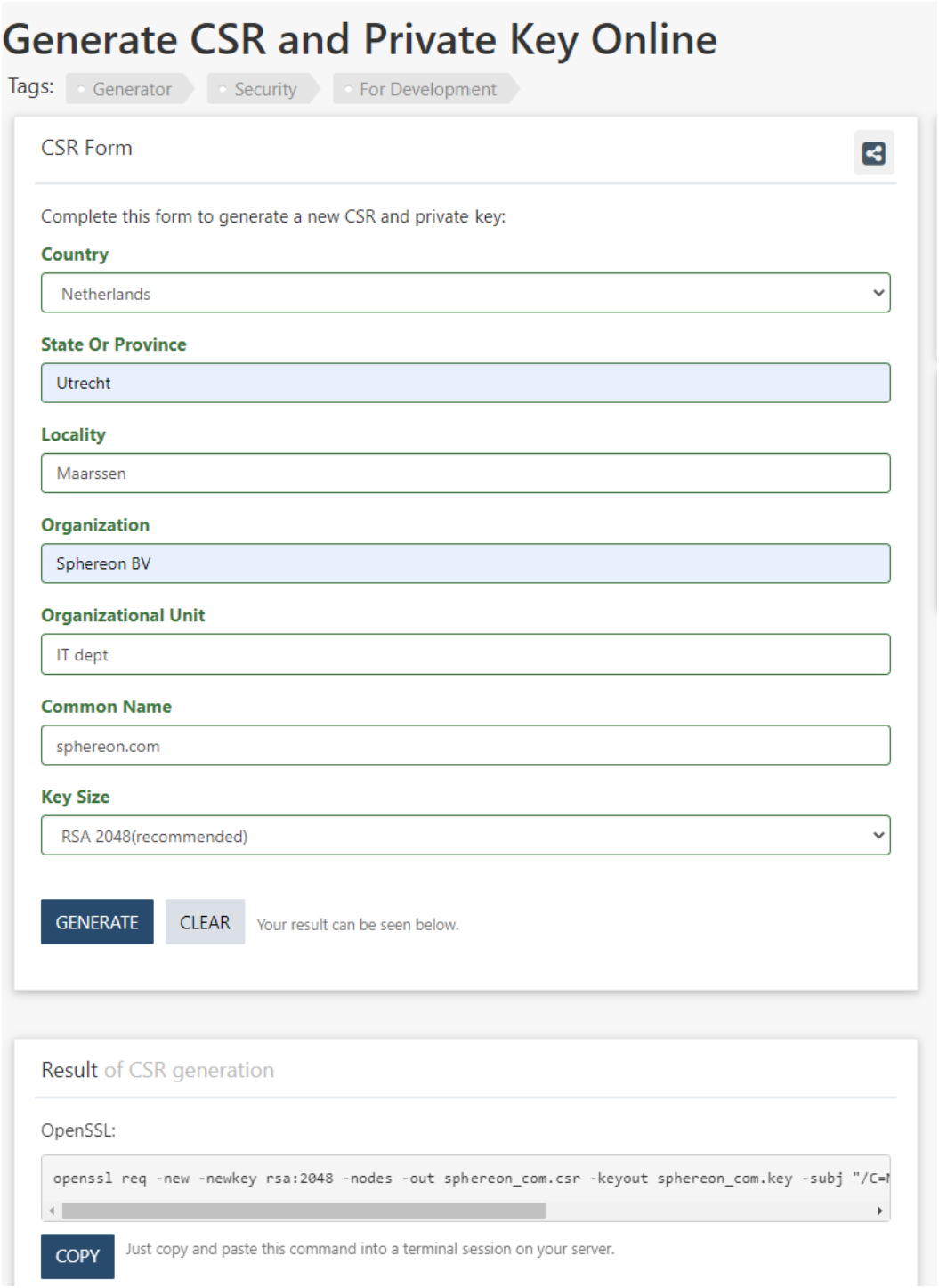
7.1.2 Certificate Signing Request (CSR)

The first step is creating the private key together with CSR. This CSR is sent to the Certificate Authority, allowing them to create the public Certificate part belonging to the private key you are generating.

You can create the CSR by hand using openssl. A handy website to create the openssl command is https://wtools.io/generate-csr-and-private-key

You can simply fill out the form with your organization information. The important bit to know is that the "Common Name" should equal the Internet Domain Name for your organization, or a subdomain where you will be hosting files related to GAIA-X in the future.

Warning: Do not use the generated CSR and private key from that website. Only copy the openssl command and run that locally to generate a CSR and private key. As the name implies a private key should never be shared, and by simply using the generated version of the CSR/private key the company owning that form tool, would potentially have access to that private key. The openssl command ensures that everything is generated locally on your computer!



Generate the CSR and private key with OpenSSL

The above example resulted in the following command to be ran locally:

```
openssl req -new -newkey rsa:2048 -nodes -out sphereon_com.csr -keyout sphereon_co
m.key -subj "/C=NL/ST=Utrecht/L=Maarssen/O=Sphereon BV/OU=IT dept/CN=sphereon.com"
```

This results in 2 files, sphereon_com.csr and sphereon_com.key respectively. The first file is the Certificate Signing Request (csr) and the second file is the private key. Never send the private key to another party!

Note: In this example the filenames include Sphereon_com, obviously these need to be substituted with the name of your domains, where dots are replaced with an underscore.

7.1.3 Request a certificate with an online Certificate vendor

There are many online certificate vendors, like www.ssls.com, KPN, DigiCert. For Gaia-X so called Extended Validation Certificates (EV SSL) certificates are the right choice for onboarding. These types of certificates typically have a price of roughly 30-60 Euro per year. Be aware that sometimes similar certificates are being sold at prices above 100 Euro/year. Unless you already own such a certificate, there is no need to buy an expensive EV SSL.

The online certificate vendor will create the public certificate for you. You will have to provide information about your organization, payment details, as well as the Certificate Signing Request. Some vendors want to receive the CSR file, but more often they want to receive a textual copy of the CSR. In order to create this textual copy, all you have to do is open the CSR file in a text-editor and copy everything from the file to the appropriate form field of the Certificate vendor.

The Online Certificate vendor now will perform a validation check on your organization. Typically you will have to wait somewhere between minutes and 1 day, depending on the vendor, before you receive your actual public certificate.

Once received, you should copy the CA chain, the received public key as well as the previously generated private key into a separate folder that you will need during onboarding.

7.2 Generate Certificates using LetsEncrypt/Certbot

Letsencrypt allows you to get X509 Certificates for your domain(s) for free. They are recognized by most browsers and can also be used in Gaia-X for now. The process to get these credentials is mostly automated and can be found for your environment in this website: https://certbot.eff.org/. Additional documentation for certbot can be found here: https://eff-certbot.readthedocs.io/en/stable/

After having installed the certificate successfully you will need the following files from the directory /etc/letsencrypt/live/[your-domain]:

- Private key: privkey.pem
- Certificate: cert.pem
- Chain: fullchain.pem