

# Symmetric Searchable Encryption component

Deployment manual version 0.6.0

Component name: SSE – Symmetric Searchable Encryption

Component deployment name: sse

## Changelog

Table 1 Document Change History

Version	Date	Pages	Author	Modification
0.1	5/3/2020	9	Hai-Van Dang	Initial release: deployment with docker in cloud or locally
0.2	30/3/2020	9	James Bowden	Deployment with docker compose
0.3	11/08/2020	12	Hai-Van Dang	Minio deployment Deployment with MySQL Deployment with database-as-a-service
0.4	15/12/2020	12	Hai-Van Dang	Minio deployment with SSL support Providing technical notes on re-compiling SJCL javascript, and modifying sjcl python package
0.5	2/3/2021	13	Hai-Van Dang	Refactor the document Support enabling SGX at SSE TA Provide technical notes about implementation of decryption using mbedtls library in a SGX enclave Explain and instruct on the configuration settings
0.6	9/3/2021	12	Hai-Van Dang	Update the configuration explanations, and deployment with docker-compose Remove technical notes

## 1. Table of Contents

1. Terminology .....	2
2. Introduction .....	2
3. Deployment with docker-compose.....	3
4. Deployment using MiCADO .....	4
5. Deployment in development mode .....	4
5.1. SSE server/ SSE TA/ SSE client without docker containers .....	4
5.2. MinIO server deployment.....	6
6. Deployment SSEServer/ TA with MySQL/ Postgres .....	7
7. Deployment SSEServer/ TA with external database-as-a-service .....	8

<b>8. Configurations explanation and example .....</b>	<b>8</b>
<b>References .....</b>	<b>11</b>

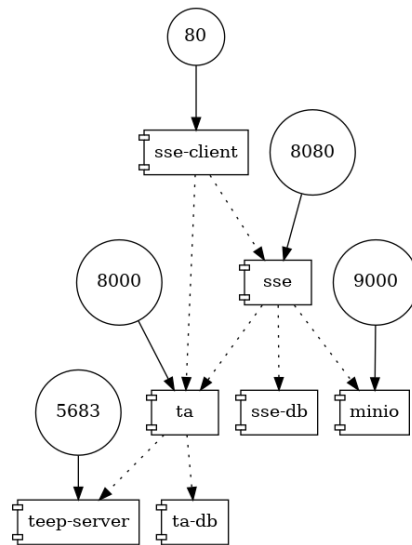
## 1. Terminology

Terminology/ Abbreviation	Explanation
SSE server	Server which stores encrypted data
SSE Trusted Authority (SSE TA)	Server which stores metadata necessary for upload/ search encrypted data
SSE client	An application which utilizes SSE javascript APIs, i.e. sse.js, to upload/ search over encrypted data
SSE database	Database of SSE server
TA database	Database of TA server

## 2. Introduction

This manual instructs how to deploy the three main Symmetric Searchable Encryption components (SSE server, SSE Trusted Authority, and a demo SSE client) and the two additional components (MinIO server [3] and teep-server [4]). SSE client [7] is a web application which uses SSE services such as uploading encrypted data, search/ update/ delete over the stored encrypted data. SSE server [5] stores the encrypted data which is uploaded by the SSE client. It also provides the services such as search/ update/ delete over the stored data. SSE Trusted Authority (SSE TA) [6] stores the metadata which is used for generating the search/ update/ delete token (at the SSE client), and verifying the search/update/delete tokens on behalf of the SSE server. Apart from that, MinIO server stores encrypted binary large objects (blob), and teep-server provides SGX [2] related services such as creating, installing SGX enclaves. These two components are optional. MinIO is only necessary when a client application wish to support encrypting blob data and search/ update/ delete over its encrypted metadata. Meanwhile, teep-server is needed when we wish to enable SGX at SSE TA to increase the security level.

Figure 1 illustrates these components and their connections, where sse-client meaning SSE client, ta meaning SSE TA, sse meaning SSE server, minio meaning MinIO server, ta-db meaning the database of SSE TA, and sse-db meaning the database of SSE server.



**Figure 1 SSE main components and two additional components**

For further explanation on the components SSE Server, SSE TA, and SSE client, please refer to the paper [8].

### 3. Deployment with docker-compose

It is possible to deploy all the SSE components on a single host using docker-compose [12]. The following section describes how to do that.

*Requirement:* If you wish to enable SGX at SSE TA, the machine needs to be able to run SGX applications. In cloud environment, at the time of this writing document, Microsoft Azure supports Intel SGX enabled virtual machines [1].

1. Get the docker-compose definition from the repository [12]

```
git clone https://gitlab.com/asclepios-project/sse-deployment
```

The docker-compose file involves the definitions of the SSE Server, SSE TA, SSE client, MinIO Server, and TEEP deployer server. If you do not need to use SGX for SSE TA, and/or use MinIO Server, you can customize the docker-compose file by removing teep-server and/or minio definitions.

2. Configure the environmental variables

```
cd sse-deployment
```

If using SGX at SSE TA:

```
cp template_sgx.env .env
```

If not using SGX at SSE TA:

```
cp template_nonsgx.env .env
```

Edit the `.env` file to modify the configurations by following the meaning and examples of configuration values in section 8. Please note that a few configurations need synchronization between components. You can have a look at the example of `.env` [13] to see how to modify the configurations.

### 3. Build the docker images

```
docker-compose build
```

### 4. Run the SSE service

```
docker-compose up
```

Once it has successfully deployed, each component is accessible on the host as follows:

- SSE Client - [http://IP\\_ADDRESS:80](http://IP_ADDRESS:80)
- SSE Server - [http://IP\\_ADDRESS:8080](http://IP_ADDRESS:8080)
- SSE TA - [http://IP\\_ADDRESS:8000](http://IP_ADDRESS:8000)
- MinIO server – [http://IP\\_ADDRESS:9000](http://IP_ADDRESS:9000)

### 5. Create the data bucket at MinIO server

Go to [http://IP\\_ADDRESS:9000](http://IP_ADDRESS:9000)

Login using the `MINIO_ACCESS_KEY` and `MINIO_SECRET_KEY` defined in the `.env` file

Select the following icon to create a bucket named as defined in `MINIO_BUCKET_NAME` in the `.env` file.



## 4. Deployment using MiCADO

Please find examples of ADT which can be used to deploy the application with MiCADO at [12] and/or [14].

## 5. Deployment in development mode

The following paragraphs instruct how to run the three applications, i.e. SSE server, TA, SSE client in development mode with SQLite instead of Postgres/ MySQL, and without docker containers. It also describes how to deploy MinIO server with/ without SSL, and as a storage server or a gateway to Amazon S3 storage.

### 5.1. SSE server/ SSE TA/ SSE client without docker containers

1. Re-configure database to use SQLite

In case of SSE TA:

```
cp template_settings_sqlite.py TA/settings.py
```

In case of SSE Server:

```
cp template_settings_sqlite.py SSEServer/settings.py
```

In case of SSE client:

```
cp template_settings_sqlite.py SSEclient/settings.py
```

## 2. Initialize database in SQLite

This needs to run once when database needs to be initialized (e.g. for the 1<sup>st</sup> run of the application, or after deleting the previous database, i.e. db.sqlite3 file).

```
python3 manage.py migrate
```

3. (In case of SSE TA and SSE Server) Define values of the environment variables, which are defined in [10] (for SSE TA) and [9] (for SSE Server).

**Table 2 Environmental variables at SSE Server**

```
export DJANGO_LOGLEVEL=DEBUG \  
DJANGO_DEBUG=true \  
TA_SERVER=http://127.0.0.1:8000 \  
ALLOWED_HOSTS=* \  
MINIO_ACCESS_KEY=MINIO_ACCESS_KEY_HERE \  
MINIO_SECRET_KEY=MINIO_SECRET_KEY_HERE \  
MINIO_BUCKET_NAME=asclepios \  
MINIO_URL=127.0.0.1:9000 \  
MINIO_SSL_SECURE=false \  
MINIO_EXPIRE_GET=1 \  
MINIO_EXPIRE_PUT=1
```

**Table 3 Environmental variables at SSE TA**

```
export DJANGO_LOGLEVEL=DEBUG \  
DJANGO_DEBUG=True \  
ALLOWED_HOSTS=* \  
HASH_LENGTH=256 \  
IV=TA_IV_VALUE_IN_UTF8_HERE
```

4. (In case of SSE client) Fill constant values of [sseConfig](#) variable in [sse.js](#) [11].
5. Run the servers in development mode

In case of SSE TA:

```
python3 manage.py runserver 0.0.0.0:8000
```

In case of SSE Server:

```
python3 manage.py runserver 0.0.0.0:8080
```

In case of SSE client:

```
python3 manage.py runserver 0.0.0.0:80
```

## 5.2. MinIO server deployment

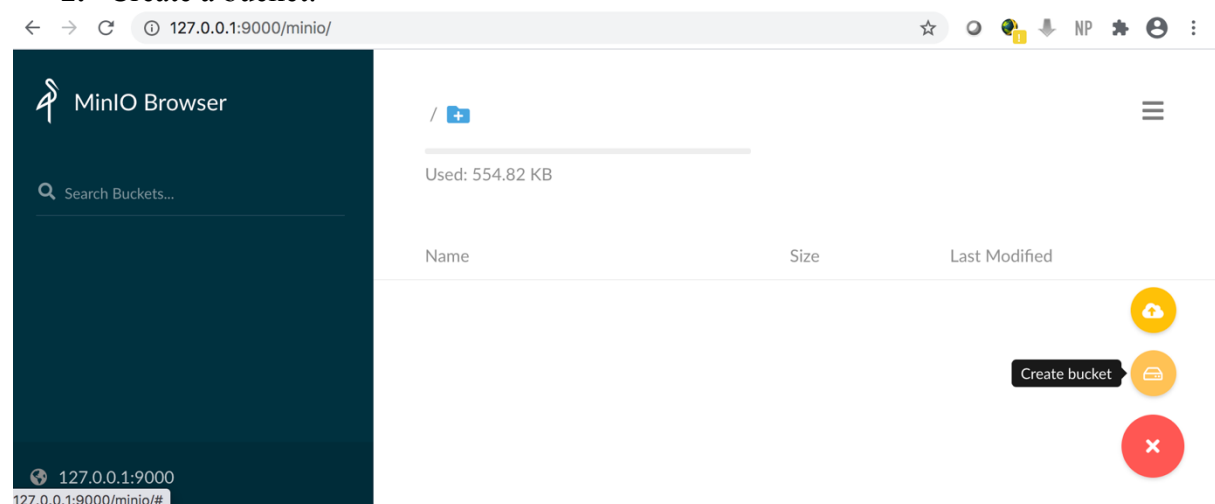
### A. As docker container without SSL

#### 1. Deploy as a docker container:

```
docker run -p 9000:9000 --name minio \
  -e 'MINIO_ACCESS_KEY=choose_access_key_here' \
  -e 'MINIO_SECRET_KEY=choose_secret_key_here' \
  -v folder_in_host_machine:/data \
  minio/minio server /data
```

If deployed successfully, you can access Minio server at <http://127.0.0.1:9000> with the chosen access key and secret key.

#### 2. Create a bucket:



### B. As a gateway to Amazon S3 without SSL

Assuming that you've created an Amazon S3 bucket. You can run Minio as a gateway to the created Amazon S3 with the following command.

```
docker run -p 9000:9000 --name minio-s3 \
    -e 'MINIO_ACCESS_KEY=access_key_from_aws' \
    -e 'MINIO_SECRET_KEY=access_secret_from_aws' \
    minio/minio gateway s3
```

### C. As docker container with SSL using self-signed certificate

#### 1. Generate self-signed certificate using OpenSSL

Example:

```
openssl genrsa -out private.key 2048
openssl req -new -x509 -days 3650 -key private.key -out public.crt -subj
"/C=US/ST=state/L=location/O=organization/CN=<domain.com>"
```

#### 2. Create a folder to store data in the host machine

```
mkdir data
```

#### 3. Create a folder to store private.key and public.crt in host machine

```
mkdir config
mkdir config/certs
```

#### 4. Copy private.key and public.crt to config/certs

Run minio as a docker container with SSL enabled

```
docker run -p 9000:9000 --name minio \
    -e 'MINIO_ACCESS_KEY=choose_access_key_here ' \
    -e 'MINIO_SECRET_KEY=choose_secret_key_here ' \
    -v data:/data \
    -v config:/root/.minio \
    minio/minio server /data
```

If deployed successfully, you can access Minio server at <https://127.0.0.1:9000> with the chosen access key and secret key.

## 6. Deployment SSEServer/ TA with MySQL/ Postgres

Database engine can be changed easily with the existing template files.

In order to change to MySQL, copy the template files then re-build the docker image for TA and/ or SSEServer.

```
cp template_Dockerfile_mysql Dockerfile
cp template_requirements_mysql.txt requirements.txt
cp template_entrypoint_mysql.sh entrypoint.sh
cp template_settings_mysql.py TA/settings.py
```

In order to change to Postgres, copy the template files then re-build the docker image for TA and/ or SSEServer.

```
cp template_Dockerfile_postgres Dockerfile
cp template_requirements_postgres.txt requirements.txt
cp template_entrypoint_postgres.sh entrypoint.sh
cp template_settings_postgres.py TA/settings.py
```

## 7. Deployment SSEServer/ TA with external database-as-a-service

In order to use database-as-a-service, for instance Amazon Relational Database Service, at first you need to create a database at the CSP, for i.e. AWS. Then you change the environment variables DB\_HOST and DB\_PORT correspondingly.

Example:  
DB\_HOST=....rds.amazonaws.com  
DB\_PORT=3306

## 8. Configurations explanation and example

**Table 4 Configurations at SSE Client**

Parameters	Value	Meaning	Example	Configurable	Need compatibility with
SSE_CLIENT_ALLOWED_HOSTS	string	List of the allowed hosts which are separated by whitespace [16]	"127.0.0.1"	Y	
SSE_CLIENT_DJANGO_DEBUG	boolean	Debug level of Django [15]	True/ False	Y	
SSE_CLIENT_DJANGO_SECRET_KEY	String, 50 characters	Django secret key [16]		Y	
SSE_CLIENT_TA_URL	URL	URL of SSE TA	http://127.0.0.1:8000	Y	
SSE_CLIENT_SSE_URL	URL	URL of SSE Server	http://127.0.0.1:8080	Y	
SSE_CLIENT_SALT	base64 string, 8 bytes	Salt value which is used for key generation from a passphrase, if needed	ZRVja4LFrFY =	Y	
SSE_CLIENT_IV	base64, 8 bytes	Initial vector value which is used for encrypting data	ZRVja4LFrFY =	Y	
SSE_CLIENT_ITER	Number	Number of iteration for generating key from passphrase, if needed	10000	Y	
SSE_CLIENT_KS	Number	Key size	128 or 256	Y	
SSE_CLIENT_TS	Number	Tag size	64	Y	
SSE_CLIENT_MODE	string	Cipher mode	ccm	Possibly, but not tested for other mode like "gcm" yet	



SSE_CLIENT_ADATA	String	Authenticated data	"	Y	
SSE_CLIENT_ADATA_LEN	Number	Length of authenticated data	0	Y	
SSE_CLIENT_HASH_LENGTH	Number	Output length of hash function	256	Y	
SSE_CLIENT_CHUNK_SIZE	Number	Size of 1 slice/ chunk for encryption (in uint8 items) when encrypting blob data	32768	Y, but needs experimenting to avoid memory crash in browser	
SSE_CLIENT_NO_CHUNKS_PER_UPLOAD	Number	Number of chunks to be packed in 1 upload when encrypting and uploading blob data	30	Y, but needs experimenting to avoid memory crash in browser	
SSE_CLIENT_SALT_TA	base64 string, 8 bytes	Salt value which is used for key generation from a passphrase, if needed	ZRVja4LFrFY=	Y	
SSE_CLIENT_IV_TA	base64 string, 8 bytes	Initial vector value which is used for encryption	YXp5bWJscWU=	Y	TA_IV <sup>1</sup> (in case SGX is not enabled at SSE TA) or iv <sup>2</sup> (in case SGX is enabled) at SSE TA
SSE_CLIENT_ITER_TA	number	Number of iteration for generating key from passphrase, if needed	10000	Y	
SSE_CLIENT_KS_TA	number	Key size	128, 256	Y if SGX is not enabled. N if SGX is enabled (its value is 128 then)	TA_KS at SSE TA
SSE_CLIENT_TS_TA	number	Tag size	64	N	
SSE_CLIENT_MODE_TA	string	Cipher mode	ccm	Possibly if SGX is not enabled ("ccm" or "gcm"), but not tested yet. N if SGX is enabled.	TA_MODE at SSE TA
SSE_CLIENT_ADATA_TA	String	Authenticated data	"	N	
SSE_CLIENT_ADATA_LEN_TA	Number	Length of authenticated data	0	N	
SSE_CLIENT_SGX_ENABLED	boolean	true if SGX is enabled at SSE TA, false otherwise	true/ false	Y	TA_SGX at SSE TA
SSE_CLIENT_CP_ABE_URL	URL	URL of CP-ABE server		Y	
SSE_CLIENT_DEBUG	Boolean	true if debug (logs will be printed), false otherwise	true/false	Y	

**Table 5 Configurations at SSE TA**

Parameters	Value	Meaning	Example	Configurable	Need compatibility with
TA_DJANGO_LOGLEVEL	string	Log level of Django	DEBUG, INFO, WARNING, ERROR, CRITICAL [15]	Y	
TA_DJANGO_DEBUG	boolean	Debug level of Django [16]	True/ False	Y	
TA_DJANGO_SECRET_KEY	String, 50 characters	Django secret key [16]		Y	
TA_ALLOWED_HOSTS	String	List of the allowed hosts which are separated by	127.0.0.1	Y	

<sup>1</sup> The TA\_IV value is in UTF-8 string format. Meanwhile, SSE\_CLIENT\_IV\_TA is in base64 format. You can use this page to <https://onlineutf8tools.com/convert-utf8-to-base64> to convert between them

<sup>2</sup> In case using SGX, if you want to change value SSE\_CLIENT\_IV\_TA which is different from 'YXp5bWJscWU=', you need to change the value of iv variable at [https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/0.4/teeclient/common/aes\\_ccm.cpp#L10](https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/0.4/teeclient/common/aes_ccm.cpp#L10) at SSE TA, and re-compile the code in <https://github.com/UoW-CPC/Asclepios-TrustedAuthority/tree/0.4/teeclient> to update the application image [https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/develop-sgx/teeclient/enclave\\_a/enclave\\_a\\_signed](https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/develop-sgx/teeclient/enclave_a/enclave_a_signed). Please follow instructions at [4] to re-compile the code.

		whitespace [16]			
TA_DB_NAME	string	Name of database	tadb	Y	
TA_DB_USER	string	Database user name	tauser	Y	
TA_DB_PASSWORD	string	Database user's password	tapwd	Y	
TA_DB_HOST	string	Database hostname	ta-db	Y	
TA_DB_PORT	Number	Database port	5432 if using postgres, 3306 if using mysql	Y	
TA_HASH_LENGTH	Number	Output length of hash function	256	Y	
TA_IV (in case SGX is not enabled)	Utf-8 string	Initial vector value which is used for encryption	abcdefgh <sup>3</sup>	Y	SSE_CLIENT_IV_TA at SSE client
iv [17] (in case SGX is enabled)	Array of bytes	Initial vector value which is used for encryption	{0x61,0x7a,0x79,0x6d,0x62,0x6c,0x71,0x65} <sup>4</sup>	Y	SSE_CLIENT_IV_TA at SSE client
TA_MODE	string	Cipher mode	ccm	Y if SGX is not enabled (ex. "gcm", "ccm"). N if SGX is enabled	SSE_CLIENT_MODE_TA at SSE client
TA_KS	number	Key size	128, 256	Y if SGX is not enabled. N if SGX is enabled (in such case, TA_KS=128)	SSE_CLIENT_KS_TA at SSE client
TA_TEEP_SERVER	URL	URL of TEEP deployer server	coap://127.0.0.1:5683/teep	Y	
TA_SGX	number	Indicate if SGX is enable	1 if SGX is enabled, 0 otherwise	Y	SSE_CLIENT_SGX_ENABLED at SSE client

**Table 6 Configurations at SSE Server**

Parameters	Value	Meaning	Example	Configurable	Need compatibility with
SSE_SERVER_DJANGO_LOG_LEVEL	string	Log level of Django [15]	DEBUG, INFO, WARNING, ERROR, CRITICAL [15]	Y	
SSE_SERVER_DJANGO_DEBUG	boolean	Debug level of Django [16]	True/ False	Y	
SSE_SERVER_ALLOWED_HOSTS	String	List of the allowed hosts which are separated by whitespace [16]	127.0.0.1	Y	
SSE_SERVER_DJANGO_SECRET_KEY	String, 50 characters	Django secret key [16]	String, 50 characters	Y	
SSE_SERVER_TA_SERVER	URL	URL of SSE TA	http://127.0.0.1:8000	Y	
SSE_SERVER_DB_NAME	String	Name of database	ssedb	Y	
SSE_SERVER_DB_USER	String	Database user name	sseuser	Y	
SSE_SERVER_DB_PASSWORD	String	Database user's password	ssepwd	Y	

<sup>3</sup> This value is in UTF-8 string format. Meanwhile, SSE\_CLIENT\_IV\_TA is in base64 format. You can use this page to <https://onlineutf8tools.com/convert-utf8-to-base64> to convert between them

<sup>4</sup> If you want to change this value, you need to re-compile the source code in <https://github.com/UoW-CPC/Asclepios-TrustedAuthority/tree/0.4/teepclient> to update the application image [https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/develop-sgx/teepclient/enclave\\_a/enclave\\_a\\_signed](https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/develop-sgx/teepclient/enclave_a/enclave_a_signed). Please follow instructions at [4] to re-compile the code.

<a href="#">SSE_SERVER_DB_HOST</a>	String	Database hostname	sse-db	Y	
<a href="#">SSE_SERVER_DB_PORT</a>	Number	Database port	5432 if using postgres, 3306 if using mysql	Y	
<a href="#">MINIO_ACCESS_KEY</a>	String	Access key to MinIO server	minio	Y	
<a href="#">MINIO_SECRET_KEY</a>	String	Secret key to access MinIO server	miniopwd	Y	
<a href="#">MINIO_BUCKET_NAME</a>	String	Name of data bucket created at MinIO server	mydata	Y	Bucket has been/ will be created at MinIO server
<a href="#">MINIO_URL</a>	URL	URL of MinIO server	127.0.0.1:9000 <sup>5</sup>	Y	
<a href="#">MINIO_SSL_SECURE</a>	Boolean	Indication if communication to MinIO server is protected SSL or not	True/ False	Y	SSL configuration at MinIO Server [5.2]
<a href="#">MINIO_EXPIRE_GET</a>	Number	Expiry (the number of days) of the presign URL to get data from MinIO server	1	Y	
<a href="#">MINIO_EXPIRE_PUT</a>	Number	Expiry (the number of days) of the presign URL to put/post data to MinIO server	1	Y	

## References

1. Azure confidential computing, <https://docs.microsoft.com/en-gb/azure/confidential-computing/confidential-computing-enclaves>
2. Costan, V., & Devadas, S. (2016). Intel SGX Explained. IACR Cryptol. ePrint Arch., 2016(86), 1-118.
3. MinIO Server, <https://min.io/>
4. Teep-deployer implementation, <https://gitlab.com/asclepios-project/teep-deployer>
5. SSE Server implementation, <https://gitlab.com/asclepios-project/symmetric-searchable-encryption-server>
6. SSE Trusted Authority implementation, <https://gitlab.com/asclepios-project/sseta>
7. SSE Client implementation, <https://gitlab.com/asclepios-project/sseclient>
8. Dang, H. V., Ullah, A., Bakas, A., & Michalas, A. (2020, October). Attribute-Based Symmetric Searchable Encryption. In *International Conference on Applied Cryptography and Network Security* (pp. 318-336). Springer, Cham.
9. Environment variables defined for SSE Server, <https://gitlab.com/asclepios-project/symmetric-searchable-encryption-server/-/blob/develop/api/resources.py#L29>
10. Environment variables defined for SSE TA, <https://gitlab.com/asclepios-project/sseta/-/blob/develop-sgx/parameters.py>
11. Configurations for SSE client, <https://gitlab.com/asclepios-project/sseclient/-/blob/develop-sgx/sse/static/js/sse.js#L25>
12. SSE deployment, <https://gitlab.com/asclepios-project/sse-deployment>
13. Configuration example, <https://gitlab.com/asclepios-project/sse-deployment/-/blob/main/example.env>
14. SSE-Sleep use case deployment, <https://gitlab.com/asclepios-project/micado-adts/-/tree/asclepios/ADT/sleep>.
15. Django log levels, <https://docs.djangoproject.com/en/3.1/topics/logging/>
16. Django deployment configurations, <https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/>
17. Initialization vector value defined at SGX-enabled SSE TA, [https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/0.4/teepclient/common/aes\\_ccm.cpp#L14](https://github.com/UoW-CPC/Asclepios-TrustedAuthority/blob/0.4/teepclient/common/aes_ccm.cpp#L14)

<sup>5</sup> Please do not include http into the value of MINIO\_URL

