Allkirjastamise teenused SiGa ja SiVa - Info täiendamisel

SiGa ja SiVa teenus

TEHIK'us on konteinerite allkirjastamise ja allkirjavalideerimise teenusena kasutusel riiklik SiGA ja SiVa teenus. Teenuse kasutamiseks on TEHIK'us loodud vahendus teenus.

Teenuste kasutamiseks peab kasutama TEHIK'u vahendus teenust. See võimaldab tulevikus lihtsamini hallata muudatusi riiklikus teenuses.

Teenuse keskkonnad

Arendus

SiGa: https://siga.arendus.tehik.ee/siga

SiVa: https://siga.arendus.tehik.ee/siva

Test

SiGa: https://siga.test.tehik.ee/siga

SiVa: https://siga.test.tehik.ee/siva

Live

SiGa:

SiVa:

Viited

SiGa riikliku teenuse dokumentatsioon: https://github.com/open-eid/SiGa/wiki

SiGa riikliku teenuse lähtekood: https://github.com/open-eid/SiGa

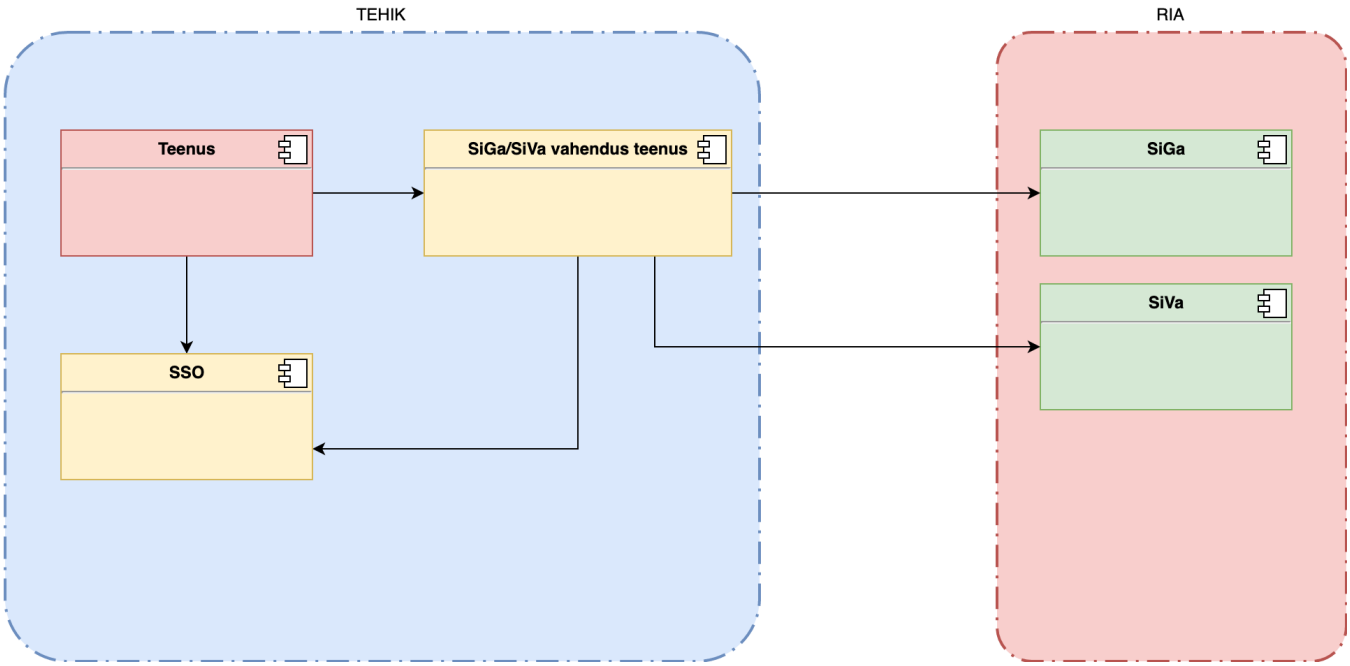
SiVa riikliku teenuse dokumentatsioon: http://open-eid.github.io/SiVa/

SiVa riikliku teenuse lähtekood: https://github.com/open-eid/SiVa

TEHIK SiGa SiVa vahendusteenuse lähtekood: https://gitlab.sotsiaalministeerium.ee/siga-siva-teenus

KeyCloak teegid: https://www.keycloak.org/docs/latest/securing_apps/

Põhimõtteline joonis



Teenusega liidestamine

Autentimine

SiGa vahendus teenuse kasutamiseks on vajalik luua liidestavale teenusele SSO teenusesse klient teenuste realmi. Teenuse kliendile tuleb määratleda õige skoop. Teenuse autentimine käib JWT identsustõendi abil.

SiVa teenusega liidestamiseks puudub vajadus autentimise järgi.

SSO kasutamine

Iga päring mis teostatakse SiGa teenuste poole, peab sisaldama JWT tõendit!

Kõige mõistlikum on lahendada JWT tokeni kasutamine SSO platvormi KeyCloak'i poolt loodud teekidega.

Kuna JWT identsustõendil on eluiga, siis peab liidestatav süsteem jälgima identsustõendi kehtivust. Kui kehtivus aeg on läbi, tuleb uuendada tõendit.

Võimalus on selleks kasutada *refresh* (värskendus) tõendit, millega endale uus *access* (ligipääsu) tõend hankida. Kuid see samm ei ole kohustuslik ja on lihtsuse mõttes võib olla mõistlikum kohe teostada uus autentimine vaastu SSO teenust.

Sellisel juhul puudub vajadus hallata *refresh* tõendeid.

Identsustõendi küsimise näide

Enne proxy teenuse väljakutsumist, peab kleint küsima SSO teenusest ligipääsutõendi (access token), kasutades selleks registreerumisel saadud clientId-d ja salasõna (credentails Secret).

Ligipääsutõendi küsimiseks tuleb teostada Keycloak *token* päring.

Näiteks:

```
curl \
-d "client_id={SSO_CLIENT_ID}" \
-d "client_secret={SSO_CLIENT_SECRET}" \
-d "grant_type=client_credentials" \
"{SSO_URL}/auth/realms/services/protocol/openid-connect/token"
```

Token meetodi vastuseks saadav JSON-i access_token andmeväljas on esitatud JWT formaadis ligipääsutõend. Ligipääsutõend sisaldab järgmisi olulisi andmevälju:

- exp - *token*-i eluiga
- sub - kasutaja süsteemne ID

- realm_access.roles või resource_access.account.roles - peab sisaldama rolli "siga_siva_proxy_client"
- clientId - kliendi ID

Proxy teenuse päringu teostamine

Saadud ligipääsutõendi väärtus tuleb iga järgneva proxy päringuga edastada HTTP Authorization päise atribuudina. Näiteks:

```
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA....
```

Ligipääsutõendi aegumine

Enne ligipääsutõendi kasutamist tuleb veenduda, et see ei ole aegunud. Tõendi aegumise aeg on määratu JWT exp andmeväljas.

Kui tõend on aegunud, siis tuleb SSO teenuselt küsida uus tõend, kasutades selleks uuesti /token päringut.

REST teenuse kasutamine

SiGa

SiGa API detailne kirjeldus: https://github.com/open-eid/SiGa/wiki/Hashcode-API-description

Arvestama peab, et erisus on autentimises ja SiGa vahendusteenuse poole pöördudes peab andma kaasa valiidse TEHIK'u SSO identsustõendi.

Proxy teenus võimaldab ligipääsu järgmistele meetoditele:

SiGa Hashcode API

```
POST https://[server url]/siga/hashcodecontainers
POST https://[server url]/siga/upload/hashcodecontainers
POST https://[server url]/siga/hashcodecontainers/{containerId}
GET https://[server url]/siga/hashcodecontainers/{containerId}/datafiles
POST https://[server url]/siga/hashcodecontainers/{containerId}/datafiles
DELETE https://[server url]/siga/hashcodecontainers/{containerId}/datafiles/{datafileName}
POST https://[server url]/siga/hashcodecontainers/{containerId}/remotesigning<br>
PUT https://[server url]/siga/hashcodecontainers/{containerId}/remotesigning/{signatureId}
POST https://[server url]/siga/hashcodecontainers/{containerId}/mobileidsigning
GET https://[server url]/siga/hashcodecontainers/{containerId}/mobileidsigning/{signatureId}/status
POST https://[server url]/siga/hashcodecontainers/{containerId}/smartidsigning
GET https://[server url]/siga/hashcodecontainers/{containerId}/smartidsigning/{signatureId}/status
GET https://[server url]/siga/hashcodecontainers/{containerId}/signatures
GET https://[server url]/siga/hashcodecontainers/{containerId}/signatures/{signatureId}
GET https://[server url]/siga/hashcodecontainers/{containerId}/validationreport
POST https://[server url]/siga/hashcodecontainers/validationreport
GET https://[server url]/siga/hashcodecontainers/{containerId}
DELETE https://[server url]/siga/hashcodecontainers/{containerId}
```

SiGa Container API

```
POST https://[server url]/siga/containers
POST https://[server url]/siga/containers/{containerId}/datafiles
GET https://[server url]/siga/containers/{containerId}/datafiles
DELETE https://[server url]/siga/containers/{containerId}/datafiles/{datafileName}
POST https://[server url]/siga/containers/{containerId}/remotesigning
PUT https://[server url]/siga/containers/{containerId}/remotesigning/{signatureId}
POST https://[server url]/siga/containers/{containerId}/mobileidsigning
GET https://[server url]/siga/containers/{containerId}/mobileidsigning/{signatureId}/status
POST https://[server url]/siga/containers/{containerId}/smartidsigning
GET https://[server url]/siga/containers/{containerId}/smartidsigning/{signatureId}/status
GET https://[server url]/siga/containers/{containerId}/signatures
GET https://[server url]/siga/containers/{containerId}/signatures/{signatureId}
GET https://[server url]/siga/containers/{containerId}/validationreport
POST https://[server url]/siga/containers/validationreport
GET https://[server url]/siga/containers/{containerId}
DELETE https://[server url]/siga/containers/{containerId}
```

SiVa

SiVa teenuse API detailne kirjeldus: http://open-eid.github.io/SiVa/siva3/interfaces/

Proxy teenus võimaldab ligipääsu järgmistele teenustele:

SiVa REST API

```
POST https://[server url]/siva/validate
POST https://[server url]/siva/validateHashcode
POST https://[server url]/siva/getDataFiles
```

SiVa SOAP API

```
POST https://[server url]/siva/soap/validationWebService
POST https://[server url]/siva/soap/hashcodeValidationWebService
POST https://[server url]/siva/soap/dataFilesWebService/getDocumentDataFiles
```

SOAP/XML

NB: Teame et üks kindel muudatus tuleb. MID allkirjastamisel tuleb anda teenusesse kaasa lisaks mobiilinumbrile ka isikukood. Isikukood oli eelnevalt vabatahtlik!

SOAP teenus on planeerimisel. Kui õnnestub, siis loome DigiDocService'le 1:1 teenuse, et võimaldada sujuvamat üleminekut.

1:1'le loodavas SOAP teenuses kaetakse ainult allkirjastamisega seotud teenused. MID autentimiseks tuleb kasutada TEHIK'u SSO teenust.