2.0 Lokal IdP

Revisionshistorik

Version	Datum	Aktör	Kommentar
0.1	30 Oct 2018	Unknown User (staffanssonm)	Upprättad
0.2	30 Oct 2018	Grim Skarsgård	Information om konfiguration vid första uppstart
0.3	12 Dec 2018	Unknown User (staffanssonm)	Information om hur man skapar Mongodatabas och användare
0.4	13 Dec 2018	Grim Skarsgård	Information om mTLS konfiguration och första uppstart
1.0	21 Jan 2019	Grim Skarsgård	Fastställd för v.1.0.0
1.1	24 Jan 2019	Grim Skarsgård	Exempel på routes i lastbalanserare
1.2	31 Jan 2019	Grim Skarsgård	Information om HSA-anslutning
1.3	04 Feb 2019	Grim Skarsgård	Omstrukturering. Första version av en checklista.
1.4	04 Apr 2019	Grim Skarsgård	Uppdaterad för v.1.0.4.
1.5	04 Oct 2019	Unknown User (staffanssonm)	Uppdaterad för v.1.1.0.
1.6	07 Apr 2020	Unknown User (staffanssonm)	Uppdaterad för v.1.2.0.
1.7	05 May 2020	Unknown User (lexhagenm)	Ändrat HSA-adresser från direktintegration till NTjP-adresser
1.8	29 May 2020	Skarsgård, Grim	Förtydligat kring MongoDB och kravet på replica set. Uppdaterat MongoDB-version.
1.9	03 Mar 2021	Skarsgård, Grim	Initial uppdatering för v.2.0
2.0	30 Mar 2021	Skarsgård, Grim	Uppdaterad för v.2.0

1. Nytt i denna versionen

Ändringar sedan senaste lokala versionen (1.2):

Se 2.0 Release notes - IdP.

IdP 2.0 har stöd för anslutning till Ineras Autentiseringstjänst. Se Att ansluta e-tjänster för information om huruvida anslutning av lokala IdP:er till Ineras Autentiseringstjänst går att beställa.

Ytterligare ändringar under ytan:

1. Möjlighet att aktivera TLS-kryptering mot DB (både MongoDB och Redis [dock är inte redis-TLS-funktionaliteten testad i produktionslik miljö]).
2. Möjlighet att använda Thales HSM för nyckelhantering.
3. Omstrukturering av databaskollektioner för anslutna klienter (existerande kollektioner migreras automatiskt vid första uppstart).
4. Rättade en bugg som gjorde att det inte gick att ändra på federerade SAML-klienter i admin-GUI.
5. Rättade en bugg som gjorde att det inte gick att exportera (ladda ner) klienter ifrån admin-GUI.

1.1. Konfig-förändringar

1.1.1. Transportkryptering mot DB

Se nedan - Transportkryptering mot MongoDB samt Transportkryptering mot Redis - för konfiguration för att sätta upp transportkryptering mot DB.

1.1.2. Aktivera/Deaktivera autentiseringsmetoderna för SITHS eID

I och med att IdP har stöd för nya autentiseringsmetoder så behöver konfiguration sättas för vilka metoder som är aktiva i IdP och valbara vid registrering av klienter.

Använd följande konfiguration för att som tidigare enbart ha mTLS aktiv.

```
authentication-method.methods.MTLS.enabled=true
authentication-method.methods.SITHS_EID_OTHER_DEVICE.enabled=false
authentication-method.methods.SITHS_EID_SAME_DEVICE.enabled=false
```

1.1.3. Använd Thales HSM för nyckelhantering

IdP har stöd för att använda Thales Luna HSM för nyckelhantering. Thales HSM-klient (SafeNet Luna HSM Client) och Java-API (SafeNet JSP) behöver finnas installerat och tillgängligt för IdP. Åtkomst till HSM-klient och JSP kräver separat avtal med Thales.

Följande konfiguration behövs i IdP för att ansluta till och använda Luna HSM.

```
inera.hsm.enabled=true
inera.hsm.slots=1,3
inera.hsm.user.role=CRYPTOOFFICER
inera.hsm.user.pwd=replaceme
inera.hsm.signer.enabled=${inera.hsm.enabled}
inera.hsm.signer.key-aliases=idp.mobiltsiths.ineratest.org-key
```

2. Checklista inför driftsättning av lokal IdP

En delmängd av de saker som behöver göras inför driftsättning av lokal IdP:

1. Teckna användaravtal med Inera för åtkomst att ladda ner applikationen
2. Påbörja anslutningsförfarandet mot HSA i god tid innan planerad driftsättning av IdP
3. Se över klusteruppsättning (egna burkar eller virtuella miljöer)
4. Installera/paketera Java 11
5. Sätt upp MongoDB med säkerhetskopiering
6. Sätt upp Redis
7. Sätt upp lastbalanserare
8. Se över portöppningar
9. Certifikat för åtkomst till HSA, förmodligen ett SITHS-utfärdat funktionscertifikat
10. Certifikat för TLS-terminering
11. Certifikat för SAML- och OIDC-meddelandesignering och -kryptering
12. Fastställ behörighetsregler för administrationsgränssnittet
13. Fastställ tolkning av Tillitsnivå (LoA) för olika typer av användarcertifikat

3. Plattform och tredjepartsprodukter

3.1. Plattform

Lokal IdP levereras som en zip-fil med en filstruktur innehållandes konfigurationsfiler tillsammans med en så kallad "fat jar", d.v.s. en .jar-fil som innehåller applikationen samt webserver och alla applikationens kodberoenden.

Jar-filen kan köras rakt upp och ner på egna servrar, köras i virtuella maskiner eller paketeras i t.ex. en docker-container och hanteras via en container-orkestreringsplattform. Den nationella instansen av Inera IdP paketeras t.ex. i docker-containers baserade på en enkel RHEL-image med Java 11 installerat och driftsätts sedan m.h.a. OpenShift.

3.1.1. Java

Java 11 krävs för att starta applikationen. OpenJDK rekommenderas, men även Oracle JDK/JRE bör fungera.

3.2. Databaser

IdP:n använder sig av MongoDB och Redis.

Redis-databasen håller enbart temporär lagring (cache, sessioner, et.c.) och behöver således inte säkerhetskopieras.

I MongoDB lagras persistent data (certifikat, klientmetadata, et.c.) och den bör därför säkerhetskopieras regelbundet.

Installation och konfiguration av databaserna ligger utanför scopet för detta dokument.

Följande versioner av databaserna har testats med IdP:

Databas	Version
MongoDB	4.0.17
Redis	4.0.11

3.2.1. MongoDB

IdP:n **kräver** att MongoDB är uppsatt som ett replica set (för att transaktioner ska fungera). Se MongoDB's dokumention för hur man skapar ett replica set. Huruvida det ligger en eller flera noder bakom replica set:et spelar ingen roll för IdP:ns del.

Applikationen kräver även att det finns en databas och en användare skapad i MongoDB som den kan använda. För att skapa upp detta, anslut till MongoDB med klienten (*mongo*/*mongo.exe*) och ange följande kommandon:

mongo

```
idpdb = db.getSiblingDB("idp")
idpdb.createUser({ user: "idpuser", pwd: "idppassword", roles: [ "readWrite" ]})
quit()
```

Namnet på databasen (**idp** i exemplet ovan) samt användarnamnet och lösenordet (**idpuser** och **idppassword**) kan väljas valfritt, men måste stämma överrens med konfigurationen i *application-custom.properties*.

IdP:n kommer sedan att vid anslutning automatiskt skapa upp de kollektioner som den behöver.

3.2.1.1. Transportkryptering mot MongoDB

Ifall trafiken mellan IdP och MongoDB skall krypteras behöver följande inställning konfigureras:

```
#Lägg till ssl=true som query-parameter i mongodb.uri. T.ex:
spring.data.mongodb.uri=mongodb://user:password@mongodb-node1:27017,mongodb-node2:27017,mongodb-node3:27017,
mongodb-node4:27017/database?replicaSet=mongo-replica-set-name&ssl=true

mongo-ssl-ca-file=<sökväg till truststore innehållandes utfärdare av databasens certifikat>
```

3.2.2. Redis

Redis används av IdP som en gemensam cache. Alla IdP-noder behöver alltså anslutas till samma uppsättning av Redis.

IdP:n kan ansluta till sentinel (kluster) eller singelnod av Redis. Redis saknar användare, men kan konfigureras för att kräva lösenord för att ansluta. IdP:n har stöd för båda alternativ. Använder man lösenord måste detta konfigureras i *application-custom.properties*.

3.2.2.1. Transportkryptering mot Redis

> TLS-funktionaliteten mot Redis är inte fullständigt testad och används inte (än) av Nationell IdP.

Redis stödjer TLS från och med version 6. Ifall trafiken mellan IdP och Redis skall krypteras behöver följande inställningar konfigureras:

```
spring.redis.ssl=true
lettuce.client.customizer.trust-store-file=<sökväg till truststore  innehållandes utfärdare av databasens
certifikat>
lettuce.client.customizer.trust-store-pwd=<lösenord för truststore ovan>
lettuce.client.customizer.disable-peer-verification=true
```

3.3. Lastbalanserare

IdP:n är tänkt att köras med mer än en instans (klustrad). Det innebär att det behövs en extern lastbalanserare som fördelar lasten mellan noderna.

3.3.1. Routes och TLS-terminering

IdP går upp med två connectorer, en för TLS-trafik (som skall termineras i lastbalanseraren) och en för mTLS-trafik (som skall släppas igenom av lastbalanseraren och termineras i applikationen).

3.3.1.1. Huvuddomän

Trafik mot IdP:s huvuddomän SSL-termineras i lastbalanseraren.

Certifikat för denna domän installeras alltså i lastbalanseraren.

3.3.1.2. Subdomän för mTLS

Trafik mot subdomänen *secure* (typ secure.idp.inera.se, om idp.inera.se är huvuddomänen) skall släppas igenom till applikationen som själv sköter mTLS-termineringen. Nycklar för hantering av mTLS-termineringen läses in i applikationen via admin-gui.

3.3.1.3. Exempelkonfiguration av routes i LB

Givet följande konfiguration i application-custom.properties:

```
...
idp.server.protocol=https
idp.server.host=idp.domain.test
idp.server.port=443
...
inera.common.server.mtls.port=8443
...
```

så kommer applikationen att innanför lastbalanseraren serva två portar: 8080 (default) samt 8443. Samtidigt är adresserna utåt https://idp.domain.test:443 och https://secure.idp.domain.test:443.

Följande konfiguration skulle då användas i lastbalanseraren:

Inkommande adress	målport hos applikationen	SSL-terminering i LB
https://idp.domain.test:443	8080	Ja
https://secure.idp.domain.test:443	8443	Nej (Passthrough)

Förslagsvis så redirectas också http-trafik (port 80) till https (port 443).

3.3.2. Headers

Lastbalanseraren måste skicka med följande headers till applikationen:

- X-Forwarded-Proto
- X-Forwarded-Host
- X-Forwarded-Port
- X-Forwarded-For

3.4. Certifikat

3.4.1. TLS-trafik

IdP går upp med två connectorer, en för okrypterad trafik (som skall termineras i lastbalanseraren) och en för mTLS-trafik (som skall släppas igenom orört av lastbalanseraren och termineras i applikationen).

- Certifikat och nyckel för IdP:s huvuddomän (ex. idp.domain.test) läses in i lastbalanseraren och används för TLS terminering på all trafik mot huvuddomänen.
- Certifikat och nyckel för subdomänen *secure* (ex. secure.idp.domain.test om idp.domain.test är huvuddomänen) läses in i applikationen via admin-gui.

Det kan antingen vara två separata certifikat, eller ett wildcard- eller multi-domain-certifikat, t.ex. ett SAN-cert med både huvuddomänen och secure-subdomänen bland sina Subject Alternative Names.

3.4.2. HSA-kommunikation

För kommunikation med HSA-katalogen krävs i regel (och definitivt vid anslutning till den nationella HSA-katalogen) ett SITHS-utfärdat funktionscertifikat vars HSA-id är registrerat i HSA-katalogen som behörigt att anropa aktuella tjänstekontrakt.

3.4.3. Övriga certifikat

Övriga certifikat är de som används för signering av SAML- och OIDC-meddelanden. Vanligtvis är detta också ett SITHS-utfärdat certifikat, och möjligen samma som används för kommunikation med HSA.

Se användarhandboken samt avsnittet om förstagångskonfiguration nedan för mer information kring installation av certifikat och nycklar.

3.5. Portöppningar

Applikationen behöver åtkomst till

IP/System
Mongo databas (samtliga noder)
Redis databas (samtliga noder)
HSA
OCSP/CRL
SAMBI, ifall federerat metadata skall hämtas
Autentiseringstjänsten, ifall autentisering med SITHS eID-klienterna skall användas

4. Beroenden till externa system

4.1. HSA

IdP nyttjar HSA som attributkälla, specifikt genom de tjänstekontrakt som finns specificerade i SAD:en.

4.1.1. Anslutning till den nationella HSA-katalogen

Anslutning av en tjänst till den nationella HSA-katalogen föregås av en utförlig anslutningsprocess. Läs mer på https://www.inera.se/tjanster/katalogtjanst-hsa/katalogtjanst-hsa/bestall--andra/ och kontakta Inera för att påbörja ett anslutningsförfarande.

4.1.2. Anslutning till regional HSA-katalog

Anslutning till en lokal/regional HSA-katalog (eller annan tjänst som implementerar de aktuella tjänstekontrakten) hanteras av den lokala/regionala förvaltningen.

4.1.3. Konfiguration av HSA-anslutning

1. Certifikat för kommunikation med HSA läses in enligt förstagångs-konfigurationen nedan.
2. Vilken HSA-katalog som IdP skall ansluta till konfigureras med följande parametrar i application-custom.properties (se avsnittet om systemkonfiguration nedan):

```
# HSA TK URL (exempel Prod Internet)
inera.common.hsa.host=https://esb.ntjp.se
# Paths
inera.common.hsa.authorization.getadmincredentialsforpersonincludingprotectedperson=/vp/infrastructure/directory
/authorizationmanagement/GetAdminCredentialsForPersonIncludingProtectedPerson
inera.common.hsa.authorization.getcredentialsforpersonincludingprotectedperson=/vp/infrastructure/directory
/authorizationmanagement/GetCredentialsForPersonIncludingProtectedPerson
inera.common.hsa.employee.getemployeeincludingprotectedperson=/vp/infrastructure/directory/employee
/GetEmployeeIncludingProtectedPerson
```

https://esb.ntjp.se/vp/infrastructure/directory/employee/GetEmployeeIncludingProtectedPerson/2/rivtabp21?wsdl
https://esb.ntjp.se/vp/infrastructure/directory/authorizationmanagement/GetCredentialsForPersonIncludingProtectedPerson/2/rivtabp21?wsdl
https://esb.ntjp.se/vp/infrastructure/directory/authorizationmanagement/GetAdminCredentialsForPersonIncludingProtectedPerson/2/rivtabp21?wsdl

4.2. Autentiseringstjänsten

Om SITHS eID-autentiseringsmetoderna skall användas behöver IdP anslutas till Autentiseringstjänsten. Se Anslutningsguide till Autentiseringstjänsten för information om anslutningsförfarande, samt Nätverksinställningar för IAM-tjänster för adresser.

IdP behöver ett SITHS Funktionscertifikat för kommunikation med Autentiseringstjänsten. IdP:ns HSA-id skickas in för registrering i Autentiseringstjänsten efter att anslutningen är godkänd. Detta HSA-id måste sedan matcha subject SERIALNUMBER i det certifikat som IdP använder för kommunikation mot Autentiseringstjänsten.

4.2.1. Konfiguration

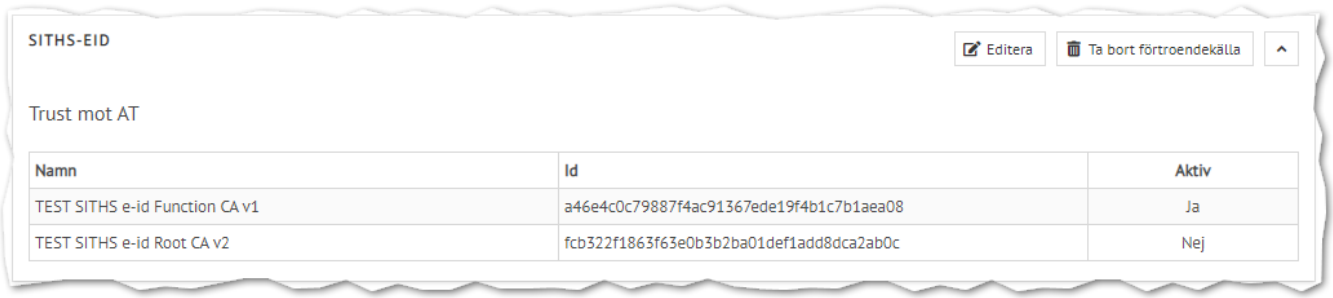
Konfigurera url:en till Autentiseringstjänstens api, och aktivera SITHS eID-metoderna.

```
# Define which authentication methods should be available at authentication as well as client-registration.
authentication-method.methods.MTLS.enabled=true
authentication-method.methods.SITHS_EID_OTHER_DEVICE.enabled=true
authentication-method.methods.SITHS_EID_SAME_DEVICE.enabled=true

# URL to the RP-API used for SITHS eID.
siths-eid.host=https://secure-authservice.mobiltsiths.ineratest.org/api/rp/v1
```

4.2.2. Trust för server-kommunikation

Lägg till utfärdandekedjan för Autentiseringstjänstens certifikat i förtroendekällan "siths-eid".



Namn	Id	Aktiv
TEST SITHS e-id Function CA v1	a46e4c0c79887f4ac91367ede19f4b1c7b1aea08	Ja
TEST SITHS e-id Root CA v2	fcb322f1863f63e0b3b2ba01def1add8dca2ab0c	Nej

4.2.3. Trust för användarcertifikat

Lägg till utfärdarkedjan för användarcertifikat som skall accepteras vid autentisering via SITHS eID-metoderna i förtroendekällan "user-siths-eid". (Alltså separat hantering från förtroendekällan "user" som endast styr mTLS-inloggningen).

Namn	Id	Aktiv
TEST SITHS e-id Person HSA-id 3 CA v1	fc6fb340a84d5b3643d13ab38b6360634ca3d67f	Ja
TEST SITHS e-id Person ID 3 CA v1	cb6a39dc642290661694acaa1290ed044e90b272	Ja
TEST SITHS e-id Person HSA-id 2 CA v1	dd7b7d394be740ac6a518246fcbd83a4afa4320f	Ja
TEST SITHS e-id Function CA v1	a46e4c0c79887f4ac91367ede19f4b1c7b1aea08	Ja
TEST SITHS e-id Person ID 2 CA v1	a8fad6eaeba3fd331d1604a35e378cf029b6a2af	Ja
TEST SITHS e-id Person ID Mobile CA v1	b5638117e42ccae071a4b7d770686844bb5964f7	Ja
TEST SITHS e-id Root CA v2	fcb322f1863f63e0b3b2ba01def1add8dca2ab0c	Nej

I bilden ovan så är Function CA v1 inläst för att tillåta automatiserade tester (bilden är från en testmiljö).

5. Applikationskonfiguration

5.1. Application properties

Installationsspecifik konfigurering görs i filen **config/application-custom.properties**. En exempelfil medföljer, men viss konfigurering i denna måste göras innan uppstart.

Framförallt måste *idp.server.host*, dvs den externa URL som man ansluter till denna instans av IdP:n sättas, samt konfiguration för att ansluta till databaserna (*spring.redis.** och *spring.data.mongodb.uri*) innan uppstart.

config/application-custom.properties

```
############################## IDP CONFIGURATION ##############################

################################################################################
############################ SERVER CONFIGURATION ##############################
################################################################################
# Outward facing address, should match public address in LB
idp.server.protocol=https
idp.server.host=idp.domain.test
idp.server.port=443


################################################################################
############################### MTLS CONNECTOR #################################
################################################################################
inera.common.server.mtls.port=8443

# Disable before first start, until the identity-group (below) has been configured with certificates
inera.common.server.enable=true

# Certificates and keys, configured in the admin GUI
inera.common.server.mtls.identity-group=idp-secure


################################################################################
################################### SECURITY ###################################
################################################################################
# Security level for admin GUI
# oidc: Secured with OIDC, default
# password: Secured with formlogin using user/password
# none: Unsecured
inera.common.security.web.level=oidc

# Username and password for admin GUI when security level is set to password
inera.common.security.web.admin-user.user-name=qwerty
inera.common.security.web.admin-user.password=asdfgh

# IP ranges allowed to access actuator endpoints
inera.common.security.web.internal-ip-range=127.0.0.1,10.0.0.0/8

# Allow or disallow access with certificates for which OSCP status cannot be verified due to network issues
inera.common.trust.allow-undetermined=true


################################################################################
############################## DB CONFIGURATION ################################
################################################################################
# Collection prefix
idp.db.prefix=idp


################################################################################
############################ REDIS CONFIGURATION ###############################
################################################################################
# Password, if any
#spring.redis.password=password

# Connection timeout, ISO8601 Duration format
spring.redis.timeout=PT1M
```

```
# Redis single node configuration
#spring.redis.password=
spring.redis.host=redis-master
spring.redis.port=6379

## Redis sentinel configuration
#spring.redis.sentinel.master=redis-cluster-name
#spring.redis.sentinel.nodes=redis-sentinel-1:26379,redis-sentinel-2:26379,redis-sentinel-3:26379


################################################################################
############################ MONGODB CONFIGURATION ############################
################################################################################
## MongoDB replica set configuration
spring.data.mongodb.uri=mongodb://user:password@mongodb-node1:27017,mongodb-node2:27017,mongodb-node3:27017,
mongodb-node4:27017/database?replicaSet=mongo-replica-set-name&ssl=true
mongo-ssl-ca-file=<file path to truststore containing the CA issuing the certificate used by MongoDB>

# QUARTZ (using MongoDB)
spring.quartz.properties.additionalconfig.uri=${spring.data.mongodb.uri}
spring.quartz.properties.additionalconfig.collection-prefix=${idp.db.prefix}_quartz


################################################################################
########################### AUTHENTICATION METHODS ############################
################################################################################
# Define which authentication methods should be available at authentication as well as client-registration.
authentication-method.methods.MTLS.enabled=true
authentication-method.methods.SITHS_EID_OTHER_DEVICE.enabled=false
authentication-method.methods.SITHS_EID_SAME_DEVICE.enabled=false

# URL to the RP-API used for SITHS eID.
#siths-eid.host=https://secure-authservice.mobiltsiths.ineratest.org/api/rp/v1


################################################################################
################################### HSA ########################################
################################################################################
# HSA TK URL (exempel, direktanslutning HSA, test, sjunet)
#inera.common.hsa.host=https://wstest.hsa.sjunet.org
# Paths
#inera.common.hsa.authorization.getadmincredentialsforpersonincludingprotectedperson=/getadmincredentials_2
/hsaws/getadmincredentialsforpersonincludingprotectedperson
#inera.common.hsa.authorization.getcredentialsforpersonincludingprotectedperson=/tk2/hsaws
/getcredentialsforpersonincludingprotectedperson
#inera.common.hsa.employee.getemployeeincludingprotectedperson=/tk2/hsaws/getemployeeincludingprotectedperson

# HSA TK URL (exempel, via nationella tjänsteplattformen, Prod, Internet)
inera.common.hsa.host=https://esb.ntjp.se
# Paths
inera.common.hsa.authorization.getadmincredentialsforpersonincludingprotectedperson=/vp/infrastructure/directory
/authorizationmanagement/GetAdminCredentialsForPersonIncludingProtectedPerson
inera.common.hsa.authorization.getcredentialsforpersonincludingprotectedperson=/vp/infrastructure/directory
/authorizationmanagement/GetCredentialsForPersonIncludingProtectedPerson
inera.common.hsa.employee.getemployeeincludingprotectedperson=/vp/infrastructure/directory/employee
/GetEmployeeIncludingProtectedPerson

# Whether or not to include a connectivity check towards HSA in /actuator/health
inera.common.hsa.healthcheck=false
# Personal identity number used in connectivity test
#inera.common.hsa.connectivity-test-person-identity-number = 19121212121212

# Searchbase
#inera.common.hsa.default-search-base = c=SE
# Logical adress
#inera.common.hsa.logical-adress = SE165565594230-1000


################################################################################
############################ HSM CONFIGURATION ################################
################################################################################
```

```
inera.hsm.enabled=false

# Prioritized list. Application will try and connect to the first one in the list,
# if that fails it continues with the next in the list until it manages to establish a connection.
# If none of the specified slots work the application will crash.
inera.hsm.slots=1,3

inera.hsm.user.role=CRYPTOOFFICER
inera.hsm.user.pwd=replaceme

inera.hsm.signer.enabled=${inera.hsm.enabled}

# The aliases in the HSM that the signer service should use to fetch credentials from. The first alias in the
list is the one that will be used.
inera.hsm.signer.key-aliases=idp.domain.test-key


################################################################################
################################## LOG CONFIG ##################################
################################################################################
# External log config, enables updating of log settings in runtime
logging.config=file:/deployments/logging/logback-spring.xml


################################################################################
################################## SAMBI ##################################
################################################################################
# Automated job to fetch federated SAML metadata from SAMBI
saml.sambi-job-enabled=false
saml.sambi-job-cron-expression=0 0 0/2 * * ?
#saml.sambi-job-cron-expression=0 * * ? * *
#saml.sambi-job-cron-expression=0 */5 * ? * *

# URI to SAMBI federated metadata
saml.federated-metadata-url=https://fed.sambi.se/trial/md/metadata.xml


################################################################################
################################## MISC ##################################
################################################################################
# Link to the user manual in GUI
idp.usermanual=https://confluence.cgiostersund.se/x/T6yhCg

################################################################################
```

5.2. Loggning

Inställningar för loggning kan göras i filen **logging/logback-spring.xml**.

Per default skrivs loggarna till fil (logs/auth-application.log), detta går att ändra till att skrivas till standard out (konsoll) genom att ändra raden *<appender-ref ref="FILE" />* till *<appender-ref ref="CONSOLE" />*.

logging/logback-spring.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration scan="true" scanPeriod="60 seconds">

  <property name="LOG_FILE" value="logs/auth-application.log" />
  <property name="LOG_FILE_MAX_SIZE" value="10MB" />
  <property name="LOG_FILE_MAX_HISTORY" value="7" />

  <include resource="org/springframework/boot/logging/logback/defaults.xml" />
  <include resource="org/springframework/boot/logging/logback/console-appender.xml" />
  <include resource="org/springframework/boot/logging/logback/file-appender.xml" />

  <!-- Global logging level of application -->
  <logger name="com.cgi.se.inera" level="INFO" />

  <!-- Supress verbose loggers -->
  <logger name="com.novemberain.quartz.mongodb" level="WARN" />

  <!-- INFO level needed to log the SOAP messages -->
  <logger name="org.apache.cxf" level="WARN" />
  <logger name="org.apache.cxf.services" level="WARN" />

  <!-- WARNING! -->
  <!-- Enabling message logging can and will expose personal information about end users in the logs! -->
  <!-- Only enable message logging if it is needed for debugging purposes, and only for limited times. -->

  <!-- OpenSaml logger for SAML request/response. DEBUG for SAML messages -->
  <logger name="PROTOCOL_MESSAGE" level="DEBUG" />

  <!-- fine tune individual service logging -->
  <logger name="org.apache.cxf.services.GetEmployeeIncludingProtectedPersonResponderInterface.REQ_OUT" level="
WARN" />
  <logger name="org.apache.cxf.services.GetEmployeeIncludingProtectedPersonResponderInterface.RESP_IN" level="
WARN" />
  <logger name="org.apache.cxf.services.GetEmployeeIncludingProtectedPersonResponderInterface.FAULT_IN" level="
INFO" />

  <logger name="org.apache.cxf.services.GetCredentialsForPersonIncludingProtectedPersonResponderInterface.
REQ_OUT" level="WARN" />
  <logger name="org.apache.cxf.services.GetCredentialsForPersonIncludingProtectedPersonResponderInterface.
RESP_IN" level="WARN" />
  <logger name="org.apache.cxf.services.GetCredentialsForPersonIncludingProtectedPersonResponderInterface.
FAULT_IN" level="INFO" />

  <logger name="org.apache.cxf.services.GetAdminCredentialsForPersonIncludingProtectedPersonResponderInterface.
REQ_OUT" level="WARN" />
  <logger name="org.apache.cxf.services.GetAdminCredentialsForPersonIncludingProtectedPersonResponderInterface.
RESP_IN" level="WARN" />
  <logger name="org.apache.cxf.services.GetAdminCredentialsForPersonIncludingProtectedPersonResponderInterface.
FAULT_IN" level="INFO" />

  <logger name="org.apache.cxf.services.NetiDAccessServerSoap.REQ_OUT" level="WARN" />
  <logger name="org.apache.cxf.services.NetiDAccessServerSoap.RESP_IN" level="WARN" />
  <logger name="org.apache.cxf.services.NetiDAccessServerSoap.FAULT_IN" level="INFO" />

  <logger name="com.cgi.se.inera.common.pkix.server.X509HeaderFilter" level="WARN" />
  <logger name="com.cgi.se.inera.common.pkix.TrustServiceImpl" level="WARN" />
  <logger name="com.cgi.se.inera.auth.oidc.endpoint.advice.OIDCExceptionHandler" level="DEBUG" />
  <logger name="com.cgi.se.inera.common.job.NonSystemExitMongoDBJobStore" level="WARN" />

  <logger name="com.cgi.se.inera.auth.core.logging.ResponseLoggingFilter" level="WARN" />
  <logger name="org.springframework.web.filter.CommonsRequestLoggingFilter" level="WARN" />

  <root level="INFO">
    <!-- Log to file or to console -->
    <appender-ref ref="FILE" />
    <!-- <appender-ref ref="CONSOLE" /> -->
  </root>

</configuration>
```

6. Inför första uppstart: Konfiguration av nycklar, cert och behörighet

6.1. Ställ ner säkerheten på admin-gui och stäng av mTLS-connectorn

När applikationen skall startas första gången så måste säkerheten på administrationsgränssnittet sättas ner för att kunna komma åt admin-gui för att konfigurera nycklar, certifikat och behörigheter.

```
inera.common.security.web.level=password
inera.common.security.web.admin-user.user-name=qwerty
inera.common.security.web.admin-user.password=asdfgh
```

Samtidigt måste mTLS-connectorn vara avstängd tills det finns en nyckelkollektion den kan använda.

```
inera.common.server.enable=false
```

Starta sedan applikationen enligt uppstarts-instruktionerna.

6.2. Konfigurera systemet via admin-gui

Åtkomst till administrationsgränssnittet sker genom att gå mot /admin-endpointen (t.ex. https://idp.domain.test/admin).

Se användarhandboken för information om hur gränssnittet används.

6.2.1. Konfigurera applikationens certifikat och nycklar

Lägg upp alla nyckelgrupper som behövs och läs in certifikat och nycklar.

Grupp-ID	Beskrivning
idp	Anger de certifikat och nycklar som används av IdP för SAML och OIDC. Det aktiva certifikatet används för signering och övriga certifikat ingår som en del av IdP metadata (inom både SAML och OIDC).
idp-authentication	Anger de certifikat och nycklar som används av administrationsgränssnittets klient för anslutning mot IdP.
idp-secure	Anger de certifikat och nycklar som används av mTLS-connectorn på *secure*-subdomänen för användarautentisering via mTLS.
hsa	Anger de certifikat och nycklar som används för anslutning till HSA. Är typiskt sett ett SITHS funktionscertifikat vars HSA-id är registrerat i HSA-katalogen som betrott att anropa aktuella tjänstekontrakt.

6.2.2. Registrera en OIDC-klient för admin-gui

Skapa en OIDC-klient för admin-gui. Kopiera värden från fliken "RP Information" i admin-gui. Dubbelkolla att nyckelgruppen som anges under "RP Information" är skapad enligt ovan.

6.2.3. Konfigurera behörighet för admin-gui

Sätt upp behörighetsregler för vilka HSA-attribut som krävs för att komma åt admin-gui.

1. Gå in på "Behörighet"
2. Klicka "Ny resurs"
3. Fyll i "ADMIN"
4. Lägg till en "READ" eller "WRITE"-Action
5. Klicka på respektive action under ADMIN-noden som dyker upp i behörighetsvyn i mitten
6. Lägg till önskade Conditions
    a. Namnsättningen är enligt OIDC-attributen på Attributlistan (t.ex. "employeeHsaId" om ni vill lägga till administratörer en och en, eller "systemRole" och "healthCareProviderHsaId" om alla med en viss roll i en organisation skall ha åtkomst)
    b. Tillgängliga OIDC-attribut är [name, employeeHsaId, commissionHsaId, commissionName, healthCareProviderHsaId, organizationName, mail, mobileTelephoneNumber, systemRole]
7. Klicka på respektive Condition och lägg till önskade värden

6.2.4. Läs in betrodda certifikat

Läs in betrodda certifikatsutfärdare för server-2-server kommunikation, användarcertifikat, sambi-federationen och eventuellt övriga metadatautfärdare. Certifikatsutfärdare för IdP:s egna certifikat måste finnas inlästa för att admin-klienten och IdP skall kunna kommunicera med varandra. Se Användarhandbok för IdP-administration för information om vilka förtroendekällor som behövs.

6.2.5. Lägg in organisations- och kontaktuppgifter

Under "Konfiguration" i admin-gui: Lägg till organisationsuppgifter samt minst två kontaktpersoner (en av Typ: technical och en av Typ: support). Denna information kommer med i IdP:s SAML-metadata.

6.3. Ställ upp säkerheten på admin-gui och aktivera mTLS-connectorn

6.3.1. Säkra admin-gui med OIDC-inloggning

När sedan trust och identiteter satts upp så ställs säkerheten på administrationsgränssnittet upp till att skyddas genom normal inloggning.

```
inera.common.security.web.level=oidc
```

6.3.2. Aktivera mTLS-connectorn

Aktivera mTLS-connectorn nu när det finns en nyckelgrupp för den att använda.

```
inera.common.server.enable=true
```

6.3.3. Starta om applikationen

7. Uppstart

Följande är ett exempel på hur applikationen kan startas med nödvändiga JVM-parametrar och environment-variabler.

Starta IdP med nödvändiga JVM-parametrar och environment-variabler

```
java -jar \
-Dfile.encoding=UTF-8 \
-Duser.country=SE \
-Duser.language=sv \
-Dspring.profiles.active=custom \
-Xms256m \
-Xmx1024m \
auth-application-*.jar
```

Lägg dessutom till följande konfig för att peka ut var Thales LunaProvider-jar (LunaProvider.jar) samt bibliotek (libLuna.so or LunaAPI.dll) ligger ifall IdP skall använda HSM för nyckelhantering.

HSM-konfig

```
-Djava.library.path=/usr/local/luna/jsp/64
-Dloader.path=/usr/local/luna/jsp/LunaProvider.jar
```

8. IdP-metadata

IdP tillhandahåller SAML- och OIDC-metadata på följande endpoints:

SAML-metadata	OIDC-metadata
<idp url>/saml	<idp url>/oidc/.well-known/openid-configuration

9. Administration (GUI)

Inloggning i administrationsgränssnittet sker genom att gå mot /admin (t.ex https://idp.domain.test/admin).

Se användarhandboken för instruktioner kring hur admin-gui används.