OS2faktor Login

Exit Strategi

**Version**: 1.1.0

**Date**: 23.04.2022

**Author**: BSG

# Formål

Dette dokument har til formål at beskrive de aktiviteter der er nødvendige for at overdrage driften af OS2faktor Login til en anden driftsleverandør end den nuværende.

Dokument forholder sig ikke til kontraktuelle forhold, men alene til aktiviteter, processer og opgaver som skal udføres i forbindelse med overdragelsen af driften fra én driftsleverandør til en anden.

# Omfang

Dette dokument omfatter alene OS2faktor Login løsningen, og dermed ikke andre omliggende komponenter, som OS2faktor MFA eller andre 3.part integrationer (fx 3.parts MFA løsninger) udarbejdet af leverandører udenfor OS2 samarbejdet.

I praksis er følgende komponenter indbefattet af dokumentet

* Håndetering af overdragelse af data, domæner og certifikater
* OS2faktor Login selvbetjening og administratorportal
* OS2faktor Login Identity Provider
* OS2faktor Login Active Directory integration
* OS2faktor Azure Active Directory integration

Kildekode til OS2faktor Login løsningen forefindes på Github her

<https://github.com/OS2faktor/OS2faktor-Login>

# Overdragelse af data, domæner & certifikater

Bemærk at OS2faktor Login er en single-tenant løsning, og der er derfor en konkret opsætning per kommune.

I forbindelse med overdragelse af driften af løsningen fra én driftsleverandør til en anden, skal der ske en overdragelse af data, DNS-registreringer og certifikater som beskrevet nedenfor

## Overdragelse af data

OS2faktor Login holder data om brugernes brugerkonti og deres erhvervsidentiteter, og den samlede log for alt hvad disse er blevet brugt til og hvordan de er blevet registreret.

Disse data skal overdrages i forbindelse med etablering af drift hos anden driftsleverandør. Hvis dette ikke sikres, skal alle brugerne have registreret nye erhvervsidentiteter.

Alle data opbevares i ét SQL skema i en MySQL database, og en SQL backup af dette skal overdrages til den nye driftsleverandør

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der er adgang til en SQL backup af data fra den kørende løsning
* At denne SQL backup kan re-etableres i en MySQL eller MariaDB database

## Registrering af domæne/DNS oplysninger

OS2faktor Login løsningen afvikles under et kundespecifikt DNS navn, og den enkelte kommune har registreret 2 DNS records der peger på hhv selvbetjeningen og Identity Provideren. Disse DNS records skal omregistreres så de peger på den nye driftleverandørs driftsmiljø.

De konkrete DNS navne er bestemt af den enkelte kommune.

Man bør som udgangspunkt forsøge at undgå at disse DNS adresser ændres, da både slutbrugere og evt lokal kommunal dokumentation formodentligt gør brug af disse. Lokale integrationer i den enkelte kommune vil ligeledes pege på disse DNS adresser.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* De to DNS records som kommunen har opsat identificeres og ændres til at pege på den nye driftleverandørs driftsmiljø

## Udstedelse af nye certifikater og udveksling af nye metadata oplysninger

Identity Provider komponenten er udstyret med et certifikat der anvendes til at udstede login tokens (SAML Assertions). Der er en række NSIS relaterede krav til administration af dette certifikat, og det anbefales på det kraftigste at der i forbindelsen med overdragelsen af løsningen til en ny driftsleverandør, bestilles et nyt certifikat, og der foretages en opdatering af certifikatet i Identity Provider løsningen, så den gamle driftsleverandør ikke har adgang til et gyldigt certifikat.

I den forbindelse skal det sikres at alle Relying Parties (Service Providers, Tjenestudbydere, kært barn mange navne) får opdateret i deres ende, så de modtager oplysninger om det nye certifikat. Denne proces er identisk med den normale proces ved certifikatskifte.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der bestilles et nyt certifikat til Identity Provideren
* Det identificeres hvilke tjenesteudbydere som der skal leveres nye metadata til

# Overdragelse af OS2faktor Login kodebasen

Kildekoden til OS2faktor Login befinder sig i Github repositoriet her

<https://github.com/OS2faktor/OS2faktor-Login>

Backenden består af 4 komponenter. Disse komponenter er alle skrevet i Java (på nær PasswordAgent modulet der er skrevet i C#), og kræver Java 11.

Man bør starte med at kompilere alle modulerne fra roden af backend kildekoden, for at sikre at alle afhængigheder er kompileret. Til dette formål anvendes Maven, og man kan kompilere ved at skrive

$ mvn clean install

Bemærk at applikationen forudsætter eksistensen af MySQL (eller MariaDB) databaseserver, og at der er oprettet et database skema samt et tilhørende brugernavn/kodeord som applikationen kan anvende.

Konfiguration af det forventede skema, brugernavn og kodeord kan opsættes i config/application.properties filerne i de enkelte undermoduler.

## OS2faktor Login selvbetjening og administratorportal

Komponten der udstiller den fælles brugergrænseflade for selvbetjeningen og adminstrationsmodulet ligger i folderen ”ui”. Når alle modulerne er kompileret, er brugergrænsefladen det også, og man kan starte det op fra kommandolinjen ved at skrive

$ mvn spring-boot:run

Dette afvikler kodebasen “in-place”, hvilket er fint til afprøvning af kildekoden.

Bemærk at Brugergrænsefladen er en såkaldt SAML tjenesteudbyder og skal udstyres med et certifikat, som anvendes til at signere login-forespørgsler til Identity Provider komponenten. Der ligger et eksempel keystore i config folderen, men i produktion skal der bruges et egentligt produktions-certifikat.

Når man kompilerer projektet, dannes en selvindeholdt JAR fil under ”target” folderen, som indeholder den fulde applikation, eksklusiv konfiguration. Man kan kopiere denne applikation over på en server, og så placere en konfigurationsfil i en konfigurationsfolder ved siden af, fx på denne måde

.

├── config

│   └── application.properties

│   └── keystore.pfx

└── os2faktor-nsis-ui.jar

Herefter kan man starte applikationen op ved at skrive

$ java -Dloader.path=”config/” -jar os2faktor-nsis-ui.jar

Man kan tage udgangspunkt i den konfigurationsfil der ligger i kildekodens ”config” folder, og tilpasse til ens konkrete deployment behov.

Bemærk at brugergrænsefladen har en integration til Fjernprint snitfladen på Serviceplatformen, og der er indgået en Serviceaftale om brugen af denne. Man skal oprette et nyt it-system på Serviceplatformen, knytte det nye certifikat til dette, og indgå en serviceaftale om brugen af Fjernprint snitfladen, så der kan sendes e-boks beskeder.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der er adgang til kildekoden for OS2faktor Login brugergrænsefladen
* Kildekoden kan kompileres og startes
* Der er bestilt et nyt certifikat til brug i brugergrænsefladen
* Der er indgået en serviceaftale om brugen af e-boks

## OS2faktor Login Identity Provider

Komponten der udstiller Identity Provider funktionaliteten ligger i folderen ”idp”. Når alle modulerne er kompileret, er Identity Provideren det også, og man kan starte det op fra kommandolinjen ved at skrive

$ mvn spring-boot:run

Dette afvikler kodebasen “in-place”, hvilket er fint til afprøvning af kildekoden.

Som nævnt tidligere, er Identity Provideren udstyret med et certifikat der anvendes til at udstede tokens. Det er kritisk i henhold til NSIS kravene, at der her bestilles et nyt certifikat (og at denne proces udføres i henhold til NSIS kravene).

Når man kompilerer projektet, dannes en selvindeholdt JAR fil under ”target” folderen, som indeholder den fulde applikation, eksklusiv konfiguration. Man kan kopiere denne applikation over på en server, og så placere en konfigurationsfil i en konfigurationsfolder ved siden af, fx på denne måde

.

├── config

│   └── application.properties

│   └── keystore.pfx

└── os2faktor-nsis-idp.jar

Herefter kan man starte applikationen op ved at skrive

$ java -Dloader.path=”config/” -jar os2faktor-nsis-idp.jar

Man kan tage udgangspunkt i den konfigurationsfil der ligger i kildekodens ”config” folder, og tilpasse til ens konkrete deployment behov.

Bemærk at Identity Provideren er udstyret med en aftale om brug af NemID til identifikation af brugere. Denne aftale (og tilhørende certifikat) bør enten overdrages til den nye driftsleverandør, eller der skal indgås en ny aftale om brug af NemID til dette formål.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der er adgang til kildekoden for OS2faktor Login Identity Provideren
* Kildekoden kan kompileres og startes
* Der er bestilt et nyt certifikat til brug i Identity Provideren
* Der er sikret adgang til brug af NemID til identifikation af brugere

## OS2faktor Login Active Directory Integration

Der er i OS2faktor Login løsningen udstillet en snitflade til overførsel af stamdata om personer der må bruge løsningen. Denne snitflade findes der flere integrationer til, og nogle kommuner har udarbejdet deres egne integrationer. Hvis en kommune anvender standard integrationen til overførsel af data fra Active Directory, er det nødvendigt for den nye driftsleverandør at håndtere denne integration.

Der er ligeledes lavet en integration den anden vej, der synkroniserer kodeordsskifte tilbage til Active Directory. Dette er en del af nedenstående integration.

Koden til Active Directory integrationen er fordelt på flere kode-moduler, som findes her

<https://github.com/OS2faktor/OS2faktor-Login/tree/master/ADSync>

<https://github.com/OS2faktor/OS2faktor-Login/tree/master/websockets>

<https://github.com/OS2faktor/OS2faktor-Login/tree/master/PasswordAgent>

### ADSync

Det første modul (ADSync) er en C# baseret windows service, som installeres lokalt i kommunens it-miljø, og som løbende overfører data om personer der må anvende løsningen. Koden kan indlæses i Visual Studio og kompileres derfra. Der findes et InnoSetup script i Installer folderen, som kan bruges til at danne en EXE installer. Man kan også bruge andre værktøjer som AdvancedInstaller eller lignende.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der er adgang til kildekoden til ADSync
* Kildekoden kan kompileres og bygges til en EXE installer

### Websockets

Det andet modul (websockets) er en Java baseret kodebase, der kræver Java 11. Den består af en middleware komponent der anvendes af OS2faktor Login backenden til at kommunikere med kommunens Active Directory for at udføre password operationer.

Komponenten kommunikerer dermed med det næste modul (PasswordAgent) der er installeret i kommunens on-premise miljø.

Man kan starte komponenten vha Maven, ved at skrive

$ mvn spring-boot:run

Dette afvikler kodebasen “in-place”, hvilket er fint til afprøvning af kildekoden.

Når man kompilerer projektet, dannes en selvindeholdt JAR fil under ”target” folderen, som indeholder den fulde applikation, eksklusiv konfiguration. Man kan kopiere denne applikation over på en server, og så placere en konfigurationsfil i en konfigurationsfolder ved siden af, fx på denne måde

.

├── config

│   └── application.properties

└── os2faktor-nsis-websockets.jar

Herefter kan man starte applikationen op ved at skrive

$ java -Dloader.path=”config/” -jar os2faktor-nsis-websockets.jar

Man kan tage udgangspunkt i den konfigurationsfil der ligger i kildekodens ”config” folder, og tilpasse til ens konkrete deployment behov.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der er adgang til kildekoden til Websockets
* Kildekoden kan kompileres og afvikles

### PasswordAgent

Det tredje modul (PasswordAgent) er en C# baseret windows service, som installeres lokalt i kommunens it-miljø, og som kommunikerer med OS2faktor backenden via websockets. Den håndterer password operationer mod Active Directory på baggrund af forespørgsler fra OS2faktor Login backenden. Der findes et InnoSetup script i Installer folderen, som kan bruges til at danne en EXE installer. Man kan også bruge andre værktøjer som AdvancedInstaller eller lignende.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der er adgang til kildekoden til PasswordAgent
* Kildekoden kan kompileres og bygges til en EXE installer

## OS2faktor Login Azure Active Directory Integration

Der er i OS2faktor Login løsningen udstillet en snitflade til overførsel af stamdata om personer der må bruge løsningen. Denne snitflade findes der flere integrationer til, og nogle kommuner har udarbejdet deres egne integrationer. Hvis en kommune anvender standard integrationen til overførsel af data fra Azure Active Directory, er det nødvendigt for den nye driftsleverandør at håndtere denne integration.

Der er ligeledes lavet en integration den anden vej, der synkroniserer kodeordsskifte tilbage til Azure Active Directory. Dette er en del af nedenstående integration.

Koden til Azure Active Directory integrationen er indeholdt i én komponent, som findes her

<https://github.com/OS2faktor/OS2faktor-Login/tree/master/azure>

Modulet er en Java baseret kodebase, der kræver Java 11. Den består af en middleware komponent der anvendes af OS2faktor Login backenden til at kommunikere med kommunens Azure Active Directory for at udføre stamdata overførsel og password operationer.

Man kan starte komponenten vha Maven, ved at skrive

$ mvn spring-boot:run

Dette afvikler kodebasen “in-place”, hvilket er fint til afprøvning af kildekoden.

Når man kompilerer projektet, dannes en selvindeholdt JAR fil under ”target” folderen, som indeholder den fulde applikation, eksklusiv konfiguration. Man kan kopiere denne applikation over på en server, og så placere en konfigurationsfil i en konfigurationsfolder ved siden af, fx på denne måde

.

├── config

│   └── application.properties

└── os2faktor-nsis-azure-ad-sync-1.0.0.jar

Herefter kan man starte applikationen op ved at skrive

$ java -Dloader.path=”config/” -jar os2faktor-nsis-azure-ad-sync-1.0.0.jar

Man kan tage udgangspunkt i den konfigurationsfil der ligger i kildekodens ”config” folder, og tilpasse til ens konkrete deployment behov.

I forbindelse med overdragelsen af driften til en ny driftsleverandør, skal denne sikre at

* Der er adgang til kildekoden til Azure integrationen
* Kildekoden kan kompileres og afvikles

# Opsigelse af aftale og sikre alle data slettes

Den eksisterende driftsaftale på OS2faktor MFA opsiges hos nuværende leverandør, og det sikres at alle data, nøgler m.m., der befinder sig under den nuværende leverandørs kontrol, slettes jf den indgående databehandleraftale.

# Revision

Driften af OS2faktor produkterne (MFA og Login) er under revision for at kunne anvende denne til NSIS anmeldelse. Det bør sikres at den nye driftsleverandør etablerer et driftsmiljø og en tilhørende driftsorganisation som kan overholde kravene til en NSIS revision, og at en sådan revision bliver udført inden overdragelsen af driften, da dette er en forudsætning for den enkelte kommunes egen NSIS revision og anmeldelse af deres løsning til Digitaliseringsstyrelsen.

I forbindelse med en overdragelse af OS2faktor MFA til en anden driftsleverandør, bør den nye driftsleverandør sikre at

* De er bekendt med alle kravene til NSIS revision
* De har etableret alle de fornødne organisatoriske, tekniske og driftsmæssige foranstaltninger, der gør det muligt at gennemføre en NSIS revision
* Der udføres en NSIS revision, og at revisionsrapporten udleveres til kommunerne, så de kan indeholde denne i deres anmeldelse til Digitaliseringsstyrelsen

Ovenstående bør ske før de nye driftsmiljø tages i brug.