Driftsdokumentasjon for Falk

Falk består av følgende tjenester:

* **Falk** – Endepunkt for OpenId connect (OIDC) og OAuth2
* **Bar** – intern webtjeneste som håndterer alle roller og rettigheter i Falk
* **Falkoner** – grensesnitt for håndtering av applikasjoner, brukere og tilganger.
* **Dodo** – Endepunkt for WS-Federation

I tillegg bruker Falk en database og et verktøy for oppdatering av databasen.

# Teknologier

Falk bygger på .**Net Core 2.1.4** og består av fire ASP.Net MVC Core-applikasjoner. For å støtte WS-Federation må Dodo kjøre på **.Net 4.6.2**, men denne vil også bli migrert til .Net Core i framtiden. Framtidige versjoner av Bar vil bruke nyere versjoner av rammeverkene.

Falk bruker **SQL Server 2016** som databasemotor. Falk kan ikke kjøre på tidligere versjoner av SQL Server da den bruker **Temporal Tables** som loggmekanisme.

Figuren under viser en overordnet oversikt over Falk. I figuren ser vi at alle de tre offentlige tjenestene bruker Bar som datakilde, men at de også skriver logger direkte til databasen.



# Logging

Alle komponentene i Falk skriver tekniske logger ved hjelp av rammeverket Serilog. Dette konfigureres for hver komponent separat. Dokumentasjon av konfigurasjonsformatet finnes her: <https://github.com/serilog/serilog-settings-configuration/blob/dev/README.md>

Normalt vil man sette **MinimumLevel** til et riktig nivå og **WriteTo** så den skriver til en fil.

# Konfigurasjonsvariabler

Hver komponent i Falk konfigureres separat via sin egen appsettings.json-fil. Mye av konfigurasjonen er felles for alle komponentene og kan gjenbrukes på tvers.

## Felles konfigurasjon

* **Serilog** – oppsett av logging
  + **MinimumLevel** – Styrer hvilket nivå av detaljer som skal logges. Følgende verdier er tillatt: Verbose, Debug, Information, Warning, Error, Fatal
  + **WriteTo** – Styrer hvor loggen skal skrives. Her velger man normalt File og angir en sti.
* **Hemit** – oppsett av Bar-tjenesten
  + **SigningCertificateThumbprint** – Thumbprint for sertifikatet som sikrer kommunikasjonen mellom den enkelte tjenesten og Bar-api
  + **Issuer** – Issuer for sertifikatet
  + **Audience** – Audience for sertifikatet
  + **BaseUri** – Url hvor Bar api kjører
* **HelseId** – konfigurasjon av tillitsforholdet mellom Falk og HelseId
  + **Authority** – Url for HelseId
  + **ClientId** – Klient id for Falk i HelseId. Hentes fra konfigurasjonsverktøyet til HelseId.
  + **ClientSecret** – Delt hemmelighet mellom Falk og HelseId. Opprettes i konfigurasjonsverktøyet til HelseId.
* **ActivityLog** – konfigurasjon av aktivitetsloggen til Falk og tilhørende tjenester
  + **ConnectionString** – Connection string for koblingen til Loggdatabasen til Falk

## Konfigurasjon av Falk (OpenId-Connect-endepunktet)

* **LocalAuthenticationCookie** – Levetid i sekunder for en brukersesjon i Falk
* **Falkoner** – Url for Falkoner-tjenesten.

## Konfigurasjon av Falkoner (brukergrensesnittet til Falk)

* **Falk** – konfigurasjon av tillitsforholdet mellom Falkoner og Falk
  + **Authority** – Url for Falk
  + **ClientId** – Klient id for Falkoner i Falk. Opprettes ved første gangs oppsett av Falk
  + **ClientSecret** – Delt hemmelighet mellom Falkoner og Falk. Opprettes ved første gangs oppsett av Falk.
* **Sms** – konfigurasjon av SMS-tjenesten til NHN
  + **AuthToken** – Delt hemmelighet mellom Falkoner og SMS-tjenesten
  + **ServiceUrl** – Url for tjenesten
* **Resh** – konfigurasjon av Resh-tjenesten til NHN
  + **Endpoint** – Url for tjenesten
  + **UserName** – brukernavn for Falkoner i tjenesten
  + **Password** – passord for Falkoner i tjenesten
* **Helseregister** – konfigurasjon av integrasjonen mellom Falk og Helseregister.no
  + **Url** – Url for tjenesten
  + **RegistryKeys** – Liste over registre som tillater konvertering av data fra Helseregister til Falk
    - **ClientId** – ClientId som brukes for oppsettet av registeret i Falk
    - **ServiceId** – ServiceId for registeret i Helseregister.no

## Konfigurasjon av Dodo (Endepunktet for WS-Federation)

* **Falk** – konfigurasjon av tillitsforholdet mellom Dodo og Falk
  + **Authority** – Url for Falk
  + **ClientId** – Klient id for Dodo i Falk. Opprettes ved første gangs oppsett av Falk
  + **ClientSecret** – Delt hemmelighet mellom Dodo og Falk. Opprettes ved første gangs oppsett av Falk.
* **Falkoner** – Url for Falkoner-tjenesten.

## Konfigurasjon av Bar-api

* **Database** – oppsett av databasen for Bar api
  + **ConnectionString** – Connection string som peker til database

# Grunnlegende konfigurasjon av Bar-database

For at tjenestene i Falk skal fungere tilfredsstillende må Bar-databasen ha et sett med grunnleggende data på plass. Hemit leverer et SQL-script som setter opp de fleste av disse verdiene, men noen verdier må settes i samarbeid med driftsleverandøren.

Scriptet fra Hemit vil sette opp tilgjengelige claimstyper, og registrere de aktuelle applikasjonene som Falk består av. Scriptet skal også registrere brukerne som er administrator av Falk. For at scriptet skal fungere som forventet må driftsleverandøren stille med riktig url for hver tjeneste, pseudonym for administratorene av løsningen og delte hemmeligheter for hver tjeneste.

# Import av tilganger fra Helseregister.no

Fram til Falk blir tatt i bruk så lagrer Helseregister.no alle brukertilganger i de nasjonale kvalitetsregistrene. Falk skal tilby en overføring av disse tilgangene, men bare for aktive brukere av registrene. For å håndtere dette har Falk en integrasjon med Helseregister.no der vi lar brukeren logge på med sitt gamle brukernavn og passord for å importere tilgangene til det aktuelle registeret.

I Helseregister.no er verdien ServiceId nøkkelen som peker til det enkelte registeret, Falk bruker verdien ClientId. For at konverteringen skal fungere må Falk vite hvilken ServiceId i Helseregister.no som passer til hvilken ClientId i Falk. Dette gjøres via konfigurasjonen til Falkoner under seksjonen **Helseregister.RegistryKeys**.

# Vedlikehold av Bar-databasen

Sammen med Falk følger det et verktøy som vedlikeholder databasen til Bar-tjenesten. Dette verktøyet, **BarDatabaseUpgrader**, brukes på følgende måte:

**BarDatabaseUpgrader updateDb "connection string til databasen"**

Programmet vil da sikre at alle tabeller og objekter i databasen har den struktur som den aktuelle versjonen av Bar-api krever.

Programmet tilbyr to opsjoner, **-d** og **-n**. Opsjonen **-d** vil slette databasen dersom den eksisterer og **-n** vil prøve å opprette databasen dersom den ikke eksisterer.

# Nettverksoppsett og tilgang til webtjenester

Alle komponentene i Bar skal brukes via https, ikke http. Dette er et designvalg som er gjort i Falk og bygger på designvalgene som er gjort i OpenId Connect og OAuth2 standardene. Det skal heller ikke brukes http i interne kall mellom tjenestene (server til server).

Tjenestene Falk, Falkoner og Dodo må være tilgjengelige for publikum, men bare internt i Helsenettet.

Tjenesten Bar skal kun være tilgjengelig internt i Falk.

Alle de fire tjenestene må kunne kommunisere med databasen direkte.

Falk må kunne nå riktig instans av **HelseId** via https.

Falkoner må kunne nå riktig instans av **Helseregister.no** via https, **SMS gateway** via https og **Resh-tjenesten** via https.