Bolagsverket

Mina ombud

Nationell infrastruktur för digitala fullmakter

API-dokumentation

Version 2.1

mina ombud

Innehållsförteckning

Ändringshistorik

Version	Datum	Beskrivning
1.0	2023-03-06	Första version
1.1	2023-04-21	Uppdaterad endpoint i testmiljö för att hämta tokens
1.2	2023-08-23	Beskrivning över versionshantering av API och dokumentation, samt kommunikation gällande ny major-version av API, samt mindre förtydliganden i texter.
2.0	2023-10-19	Ny major-version av API (2.0) där behörighetstyp har tagits bort i informations- och datamodell. Se API-specifikation för mer detaljer. Förtydliganden i kapitel 3.2 gällande att access token begärs.
2.1	2024-02-22	Uppdatering av OIDC specifikation i avsnitt 3.3.1.1. Mindre förtydliganden i text och struktur inför release av privatperson som fullmaktsgivare.

1 Inledning

Dokumentet avser att ge utvecklare och arkitekter inblick i några av de vanligaste användningsfallen som finns för Mina ombud från ett konsumentperspektiv för en anropande tjänst. Antingen så hämtas behörighet från fullmakt eller all fullmaktsinformation. API-operationer och de datamodeller som gäller för API:et beskrivs i API-specifikationen som finns publicerad i sidfoten på webbportalen för Mina ombuds administrationsverktyg, https://admin.minaombud.se.

Detta dokument beskriver också hur API-anrop genomförs och vilka förutsättningar som gäller för respektive typ av anrop – om det finns en identifierad slutanvändare eller inte.

2 Användningsfall

Nedan beskrivs några av de vanligaste användningsfallen som involverar anslutens parts initiativ. Detta för att ge förståelse om vilka delar som ingår i de processer som etablerats inom Mina ombud.

2.1 Fullmaktshavare agerar i e-tjänst med hjälp av fullmakt

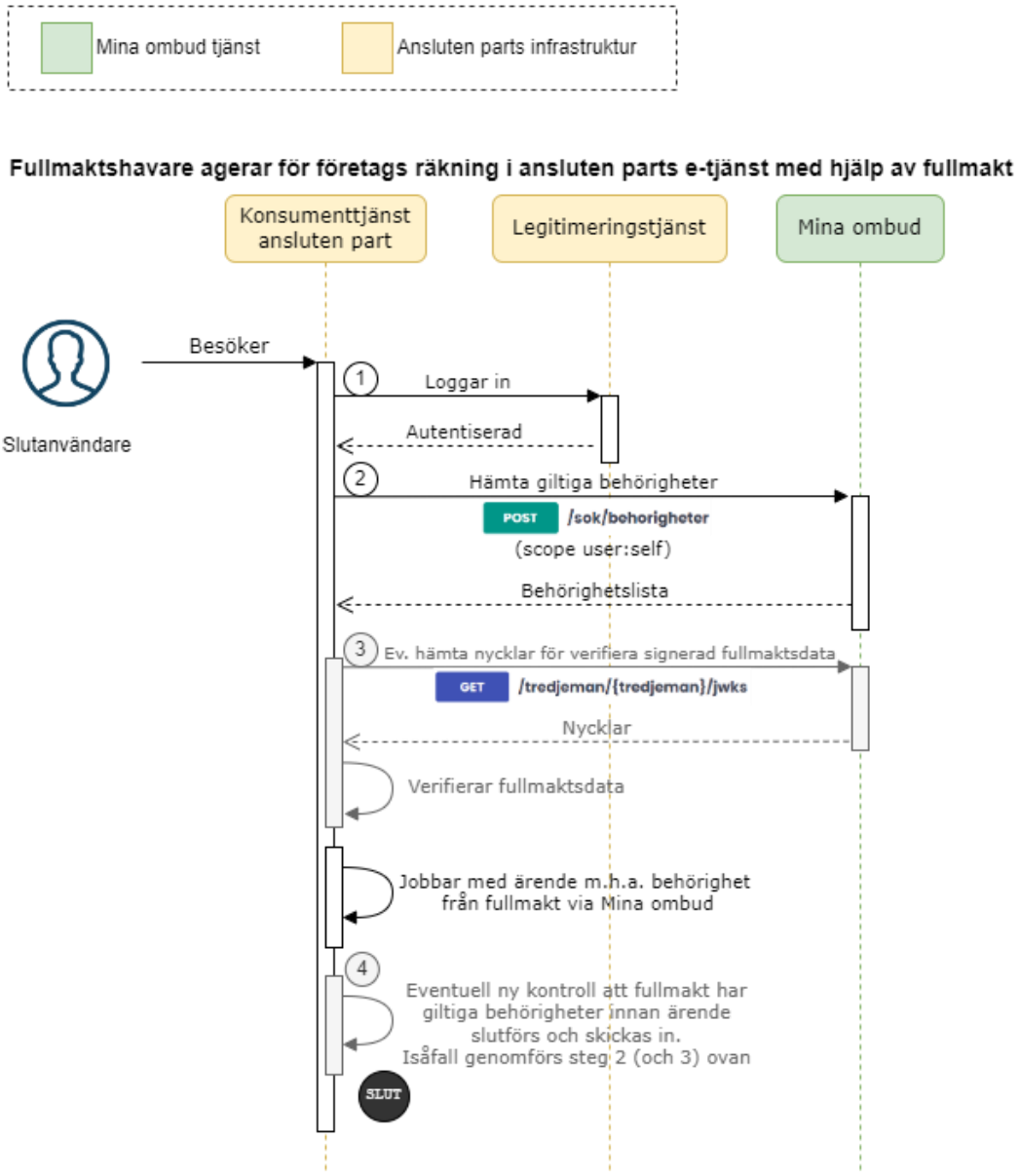

Bild 1: Fullmakt nyttjas i digital tjänst hos konsumenttjänst

Nedan beskrivs de steg som syns i bild 1.

1. Ansluten part identifierar slutanvändaren genom förslagsvis någon typ av eID.
2. E-tjänst eller annat verksamhetssystem anropar Mina ombud API genom API-operationen */sok/behorigheter*, och skickar med användaruppgifter (identifierad slutanvändares personnummer).
3. När behörigheter erhålls från Mina ombud API kan ansluten part verifiera riktigheten genom att efterfråga den publika nyckeln i API-operationen */tredjeman/{tredjeman}/jwks*.
   Denna används tillsammans med den signering som kom i svaret på API-operationen */sok/behorigheter*.
   Framför allt är det viktigt att verifiera informationen om den förmedlas vidare till ett annat system. Då är det systemet som utför verifieringen så att det inte går att anropa det systemet med förfalskade uppgifter.
4. I de fall det förekommer utkasthantering av ärende (slutanvändare sparar sitt ärende och återkommer senare) bör behörigheten på fullmakten verifiera sin giltighet en gång till innan den nyttjas.

2.2 Fullmaktsdata hämtas av ansluten part

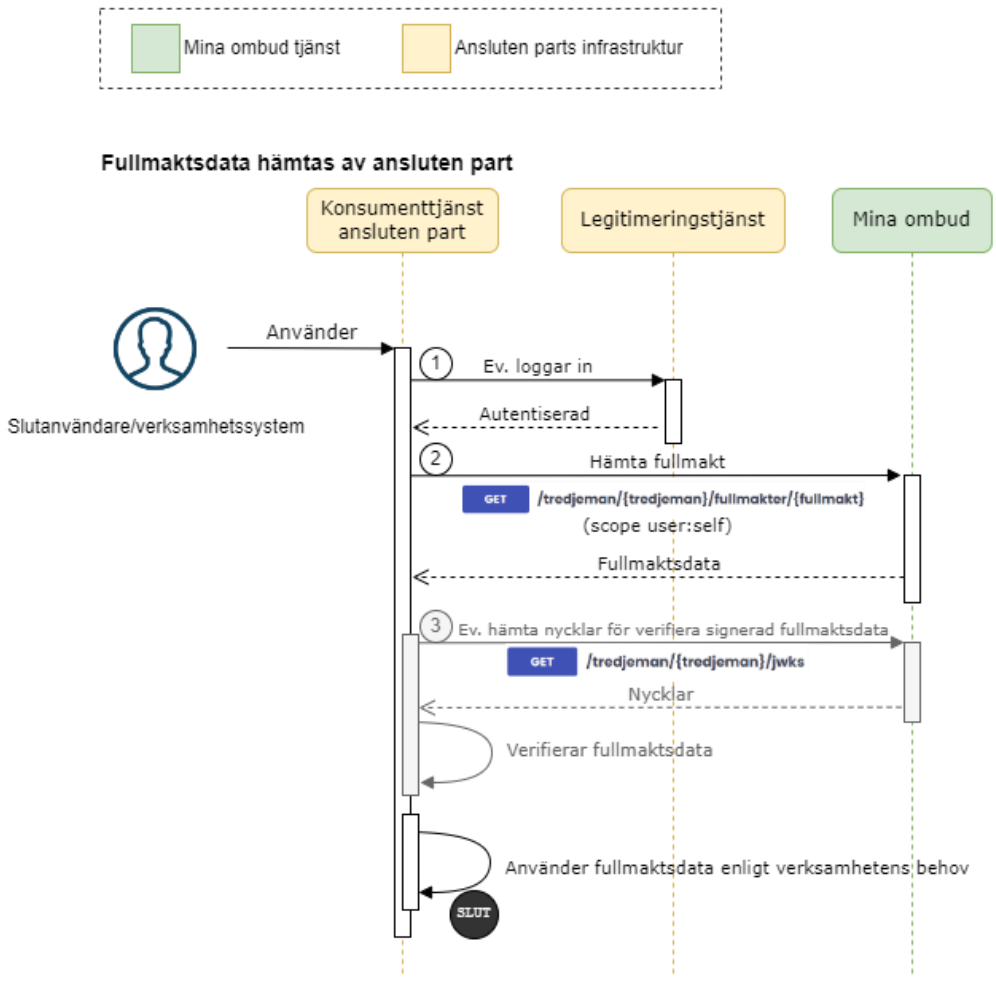

Bild 2: Fullmakt hämtas av anslutens parts konsumenttjänst

Nedan beskrivs de steg som syns i bild 2.

1. Ansluten part identifierar slutanvändaren genom förslagsvis någon typ av eID.
2. E-tjänst eller annat verksamhetssystem anropar Mina ombud API genom API-operationen */tredjeman/{tredjeman}/fullmakter/{fullmakt}*, och skickar med användaruppgifter (identifierad slutanvändares personnummer).
3. När fullmaktsdata erhålls från Mina ombud API kan ansluten part verifiera riktigheten genom att efterfråga den publika nyckeln i API-operationen */tredjeman/{tredjeman}/jwks.*
   Denna används tillsammans med den signering som kom i svaret på API-operationen */tredjeman/{tredjeman}/fullmakter/{fullmakt}.*
   Framför allt är det viktigt att verifiera informationen om den förmedlas vidare till ett annat system. Då är det detta system som utför verifieringen så att det inte går att anropa med förfalskade uppgifter.

3 Anrop till API

3.1 Semantisk versionshantering

Mina ombud API använder sig av semantisk versionshantering.

API:ets version kopplad till denna dokumentation hanteras på följande vis:

- Ny major-version av API genererar också i ny major-version av denna dokumentation.
- Ny minor-version av API kan ge ny minor-version av denna dokumentation, men är inget krav. Vidare kan denna dokumentation genomgå flera ändringar som inte återspeglas i att ny minor-version av API skapas.
- API:ets specifikation förhåller sig alltid till senast publicerad version av dokumentation eller tidigare majorversion. Ex så finns v1.x av API dokumenterad i version 1 av denna dokumentation.

API-konsumenter **SKALL** kunna hantera att nya attribut tillkommer i API-svar.

Innan release av ny major-version av Mina ombud API, ska de anslutna parterna informeras om de förändringar som kommer. Detta sker minst två (2) månader innan release via olika kanaler.

3.2 Access tokens

Vid anrop till Mina ombud API måste en giltig access token användas. Exemplen nedan beskriver hur man använder sig av JSON Web Token (JWT).

Enligt OAuth2 finns det ett antal olika sätt att generera sina access tokens. Mina ombud stöder *Client Credentials*, enligt RFC 6749 (https://www.rfc-editor.org/rfc/rfc6749.html#section-4.4). Därför **SKALL** denna typ användas för dina åtkomstnycklar. Man **SKALL** begära en access token dynamiskt från applikation och skicka den som ett header-attribut (*Authorization: Bearer*).

Dessutom **SKALL** en header vid namn *X-Service-Name* ([a-zA-Z0-9-]) skickas med i alla anrop där värdet identifierar namnet på den tjänst som utför anropet.

Endpoint adress som används för att begära ny access token finns i avsnitt 3.5.

Mina ombud API är skyddad med scopes, som är en extra behörighet för varje resurs. För att begära en access token med rätt scope specificerar du detta i ditt anrop, se avsnitt 3.3.1 för mer information.

3.2.1 Begära access token

Eftersom man kan använda i stort sett vilket programmeringsspråk som helst för sin applikation är det svårt att ge konkreta tips på hur man exakt ska göra. För att få ett dynamiskt flöde kan man exempelvis följa dessa konceptuella steg:

1. Begär en access token genom att skicka en POST-request till endpoint med åtkomstinformation i headern.
2. Håll koll på när denna access tokens giltighetsperiod tar slut, se attributet *expires_in* som anger tiden i antal sekunder.
3. Gör de anrop som behövs mot Mina ombud API.
4. Återupprepa från steg 1 när token gått ut.

3.3 Identifiering av användare

För att enkelt kunna användas med anslutande parters existerande autentisering så skickas information om slutanvändaren (resursägaren i OAuth2) separat från autentisering av API-anropet.

Systemet behöver verifierar de användaruppgifter som anges i API-anropen och därför behöver ansluten part tillhandahålla en endpoint där en *JSON Web Key Set* (JWKS) enligt RFC 7517 (https://www.rfc-editor.org/rfc/rfc7517) publiceras med den publika nyckeln som användes för att signera användaruppgifterna. Denna endpoint anges sedan i *JWKS URI* i anslutningsuppgifterna, se avsnitt 3.6 nedan.

Nyckelhantering

Användaruppgifter **SKALL INTE** signeras med samma nycklar i produktionsmiljö och testmiljö. Om samma nycklar används så finns risk att uppgifter från testmiljö godtas i produktionsmiljö vid en konfigurationsmiss.

Samma URL **KAN** användas men samma nycklar **SKALL INTE** användas och de **SKALL INTE** heller ha samma nyckel-id (*kid*-attributet). Säkrast är att anslutande part har separata miljöer för att hantera publika nycklar i JWKS anrop.

Användarinformationen som skickas i headern *X-Id-Token* **SKALL** vara en signerad JWT enligt RFC 7515 (https://www.rfc-editor.org/rfc/rfc7515.html).

Krav på nycklar för verifiering av signaturer finns presenterade på Github, https://github.com/bolagsverket/mina-ombud-samples#krav-på-signering-av-id-token-för-slutanvändare

3.3.1 Anrop

Respektive ansluten parts konto konfigureras med en behörighetskontroll i form av scope.

Anropande tjänst **SKALL** skicka med användaruppgifter om slutanvändare, när så är möjligt. Det finns därför två sätt att anropa Mina ombud API:

1. **Inloggad användare (slutanvändare) agerar i anropande tjänst**

   När en användare agerar med fullmakt måste den identifieras i tjänsten och dess användaruppgifter bifogas i anropet. Detta är standardsättet att anropa Mina ombud API.

2. **Inloggad användare (slutanvändare) saknas i anropande tjänst**

   Inträffar t.ex. när ett system hämtar information om fullmakter för digital bearbetning. Detta sätt kräver en diskussion med Mina ombud för att kunna genomföras och konfiguration krävs i Mina ombud.

Dessa båda beskrivs ytterligare nedan.

3.3.1.1 Inloggad användare (slutanvändare) agerar i anropande tjänst

Observera

Scope user:self är obligatoriskt mot produktionsmiljön för Mina ombud när det finns en identifierad slutanvändare.

När detta scope begärs agerar användaren som sig själv, vilket betyder att användaren måste vara antigen fullmaktshavare eller representera fullmaktsgivaren.

Regler, inloggad användare

- Behörigheter kontrolleras enligt följande regler
  - fullmaktshavare har full behörighet till tilldelade fullmakter
  - när fullmaktsgivaren är ett företag eller organisation så har dess företrädare läsbehörighet och firmatecknarna full behörighet
  - om fullmaktsgivaren är en enskild näringsverksamhet så har föreståndare och prokurist läsbehörighet och innehavaren full behörighet
- Access token **SKALL** begäras med scope user:self
- Information om slutanvändaren **SKALL** vara en signerad JWT enligt RFC 7515 (https://www.rfc-editor.org/rfc/rfc7515.html) som följer OIDC Swedens attribut-specifikation (https://github.com/oidc-sweden/specifications/blob/main/swedish-oidc-claims-specification.md) och skickas i headern *X-Id-Token* i alla API-anrop.
- Följande attribut **SKALL** finnas i signerad JWT
  - https://id.oidc.se/claim/personalIdentityNumber
    eller
    https://id.oidc.se/claim/coordinationNumber
  - family_name
  - given_name
  - name
- Ansluten part förser Mina ombud med en JWKS-endpoint som innehåller publik nyckel enligt avsnitt 3.3.

Exempel - söka behörighet som fullmaktshavare

1. Begär access token

Begär access token

```
S> $token = Invoke-RestMethod https://auth-
accept.minaombud.se/auth/realms/dfm-accept2/protocol/openid-connect/token
`
  -Method POST `
  -Body
@{grant_type="client_credentials";client_id="...";client_secret="...";
scope="user:self"} `
  | Select -ExpandProperty access_token
```

Angivet scope används för att verifiera behörigheterna i Mina ombuds system för användarhantering. Token erhålls vid verifierade uppgifter.

client_id och *client_secret* är unika och tillgängliga för respektive ansluten part.

2. Skapa/signera användaruppgifter (för slutanvändaren) - signerad JWT

Gå till http://jwt.io/ och skriv in följande uppgifter eller generera en signerad JWT (RFC 7515) på annat lämpligt sätt.

OBS! För att detta ska fungerar behöver Mina ombud verifiera de användaruppgifter som anges i API-anropet och därför behöver ansluten part tillhandahålla en endpoint där ett JSON Web Key Set (JWKS) publiceras med den publika nyckeln som användes för att signera användaruppgifterna (*Verify signature* nedan).

```
Header:
{
  "alg": "RS256",
  "typ": "JWT"
}
Payload:
{
  "sub": "95c72b50-ae52-4000-868f-521ec6a75b42",
  "iss": "...från anslutningsuppgifterna",
  "aud": "...från anslutningsuppgifterna",
  "name": "Nina Greger",
  "given_name": "Nina",
  "family_name": "Greger",
  "https://id.oidc.se/claim/personalIdentityNumber": "195206142597"
}
Verify signature:
{
  *Klistra in din privata nyckel, .pem eller .jwk format
}
```

Spara den genererade token (från encoded fältet på vänster sida på http://jwt.io/) från ovan och använd den sedan i *X-Id-Token* headern för API-anropet.

3. Anropa API

Anrop API

```
PS> $idtoken = "...token från ovan"
> Invoke-RestMethod https://fullmakt-
test.minaombud.se/dfm/formedlare/v2/sok/behorigheter `
  -Method POST -ContentType "application/json" `
  -Headers @{'Authorization'="Bearer $token"; 'X-Service-
Name'='yourLocalServiceName'; 'X-Id-Token'=$idtoken} `
  -Body
(@{fullmaktshavare=@{id="198101052382";typ="pnr"};tredjeman="2120000829"}
| ConvertTo-Json)
```

Arbetsflöde

Följande sekvensdiagram förklarar ovan exempel - hur anropen sker och vad som förväntas ske hos respektive komponent i flödet.

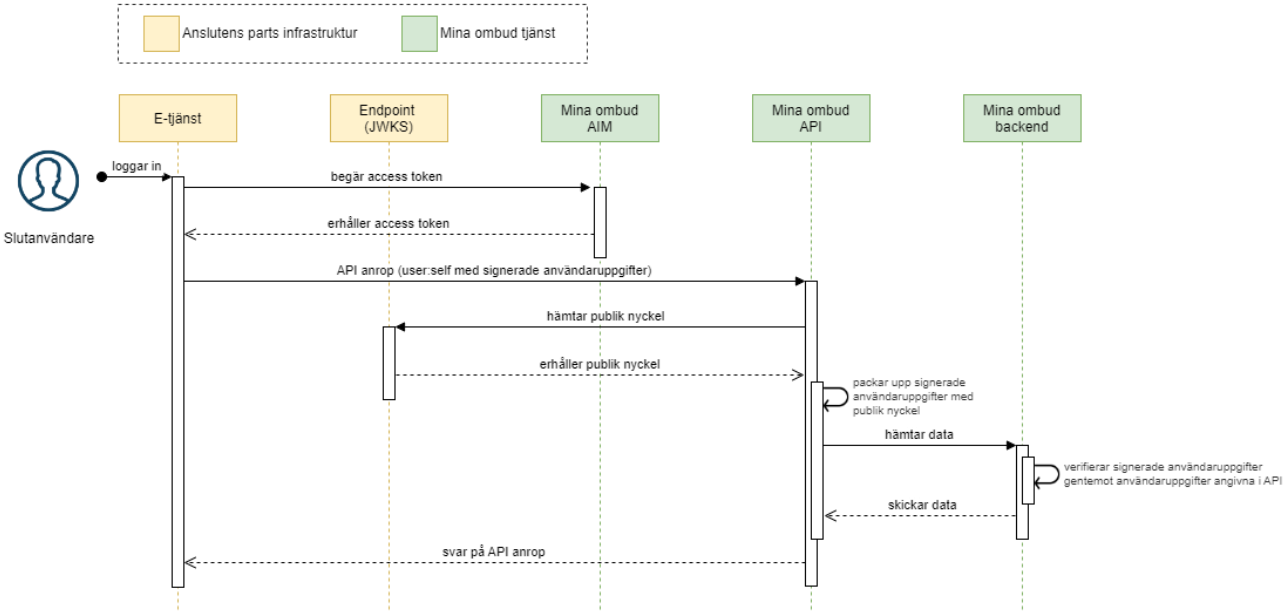

Bild 3: Sekvensdiagram över hantering av scope user:self

3.3.1.2 Inloggad användare (slutanvändare) saknas i anropande tjänst

> **Observera**
>
> Scope user:any används i onboarding av nya anslutna parter mot testmiljöer, för att snabbare kunna börja verifiera API-anrop och dylikt. För produktionsmiljön är scope user:self är standardmetoden och för att nyttja user:any mot produktionsmiljön krävs en dialog och överenskommelse med Mina ombud.

Detta scope används för bakgrundsbearbetning som saknar en slutanvändare och ingen användarinformation ska skickas i API-anropet, dvs maskin-till-maskin.
Anslutande part **SKALL INTE** använda denna typ av autentisering för något annat än ren maskinbearbetning, t.ex. vid batchkörningar. Alla situationer där en användare interagerar med ett system som resulterar i anrop till API:et för fullmakter **SKALL** identifiera denna användare med user:self.

Access token **SKALL** begäras med scope user:any

Exempel - anropa sök behörighet utan slutanvändaren

1. Begär access token

Begär access token

```
PS> $token = Invoke-RestMethod https://auth-
accept.minaombud.se/auth/realms/dfm-accept2/protocol/openid-connect/token
`
  -Method POST `
  -Body
@{grant_type="client_credentials";client_id="...";client_secret="...";
scope="user:any"} `
  | Select -ExpandProperty access_token
```

2. Anropa API

Anropa API

```
> Invoke-RestMethod https://fullmakt-
test.minaombud.se/dfm/formedlare/v2/sok/behorigheter `
  -Method POST -ContentType "application/json" `
  -Headers @{Authorization="Bearer $token"; 'X-Service-
Name'='yourLocalServiceName'} `
  -Body
(@{fullmaktshavare=@{id="198101052382";typ="pnr"};tredjeman="2120000829"}
| ConvertTo-Json)
```

3.4 Signering

3.4.1 Data

Fullmakter och behörigheter signeras för att uppgifternas äkthet ska kunna verifieras.

JSON-data kanonikaliseras enligt RFC 8785 (https://www.rfc-editor.org/rfc/rfc8785.html), attributet *_sig* tas bort och därefter beräknas signaturen på UTF8-representationen av JSON-texten.

Signaturen bäddas in i JSON-informationen i attributet *_sig* som en *detached flattened JWS JSON* enligt RFC 7515 (https://datatracker.ietf.org/doc/html/rfc7515#section-7.2.2).

Signering använder algoritmen *RS256* enligt RFC 7518 (https://www.rfc-editor.org/rfc/rfc7518.html).

Signerat JSON-objekt

```
{
  # payload fields
  # ...
  # signature field
  "_sig": {
    "protected": "<integrity-protected header contents>",
    "signature": "<signature contents>"
  }
}
```

För att verifiera signaturen behövs den publika nyckeln för ansluten part. Den publika nyckeln publiceras i ett *JSON Web Key Set* enligt RFC 7517 (https://datatracker.ietf.org/doc/html/rfc7517#section-5) på */dfm/formedlare/v2/tredjeman/{tredjeman}/jwks* där *{tredjeman}* är organisationsnummer för ansluten part som informationen gäller.

3.5 Endpoints

3.5.1 Produktion

Endpoint	URL
API	https://fullmakt.minaombud.se/dfm/formedlare/v2
Token endpoint	https://auth.minaombud.se/auth/realms/dfm/protocol/openid-connect/token
Mina ombud	https://minaombud.se

Tabell 1: Endpoints för produktionsmiljön

3.5.2 Test

Endpoint	URL
API	https://fullmakt-test.minaombud.se/dfm/formedlare/v2
Token endpoint	https://auth-accept.minaombud.se/auth/realms/dfm-accept2/protocol/openid-connect/token
Mina ombud	https://test.minaombud.se

Tabell 2: Endpoints för testmiljön

3.6 Beroenden

Nedan information beskriver uppgifter som anges vid anslutning till Mina ombud och kan ses som guidande till texter och instruktioner i ovan kapitel.

Uppgifter från ansluten part:

- **JWKS URI**
  Anger adress för att hämta JWKS (nycklar) för att vi ska kunna verifiera användaruppgifterna som skickas med vid API-anrop.
  En HTTP-endpoint som svarar med ett JSON Web Key Set (RFC 7517) för de nycklar som används för att signera användarinformationen (Content-Type skall vara *application/jwk-set+json*).
  Anslutande part kan när som helst rotera nycklar förutsatt att de användaruppgifter som skickas i API-anrop är signerade med nycklar som finns i JWKS.

- **Issuer URI**
  Adress till utfärdaren av användaruppgifterna (JWT), dvs den som säkrat identifiering. Kan sättas till exempelvis motsvarande anslutens parts domän (https://minaombud.se). Matchas i API-anropen.

- **Audience**
  En eller flera sträng-värden som godtas för attributet *aud* i användaruppgifterna (JWT). Vanligen namn på anslutens parts OAuth auktorisationsserver, eller om detta inte finns de applikationer som förväntas nyttja Mina ombud. Matchas i API-anropen.

- **IP-adress range**
  För anslutning till testmiljö, begränsar åtkomst till testmiljön via (IP-whitelisting)

- **Beställningsidentitet**
  Beställningsidentitet som ska används vid anrop till Skatteverket inom ansluten parts eget utrymme. Ansluten part skickar dess beställningsID (erhållet av Skatteverket) tillbaka till Mina ombud. Nytt beställningsID krävs även om ansluten part redan har ett beställningsid, då Mina ombud agerar förmedlare av informationen.

4 Exempelkod

På Bolagsverkets Github konto https://github.com/bolagsverket finns exempelkod publicerad som ämnar att underlätta den initiala testningen och anslutningen mot Mina ombud testmiljö.

Exempelkoden innefattar alla ovanstående delar såsom access token, JWT hantering, API-anrop samt signering. Exemplen finns på flera programmeringsspråk.