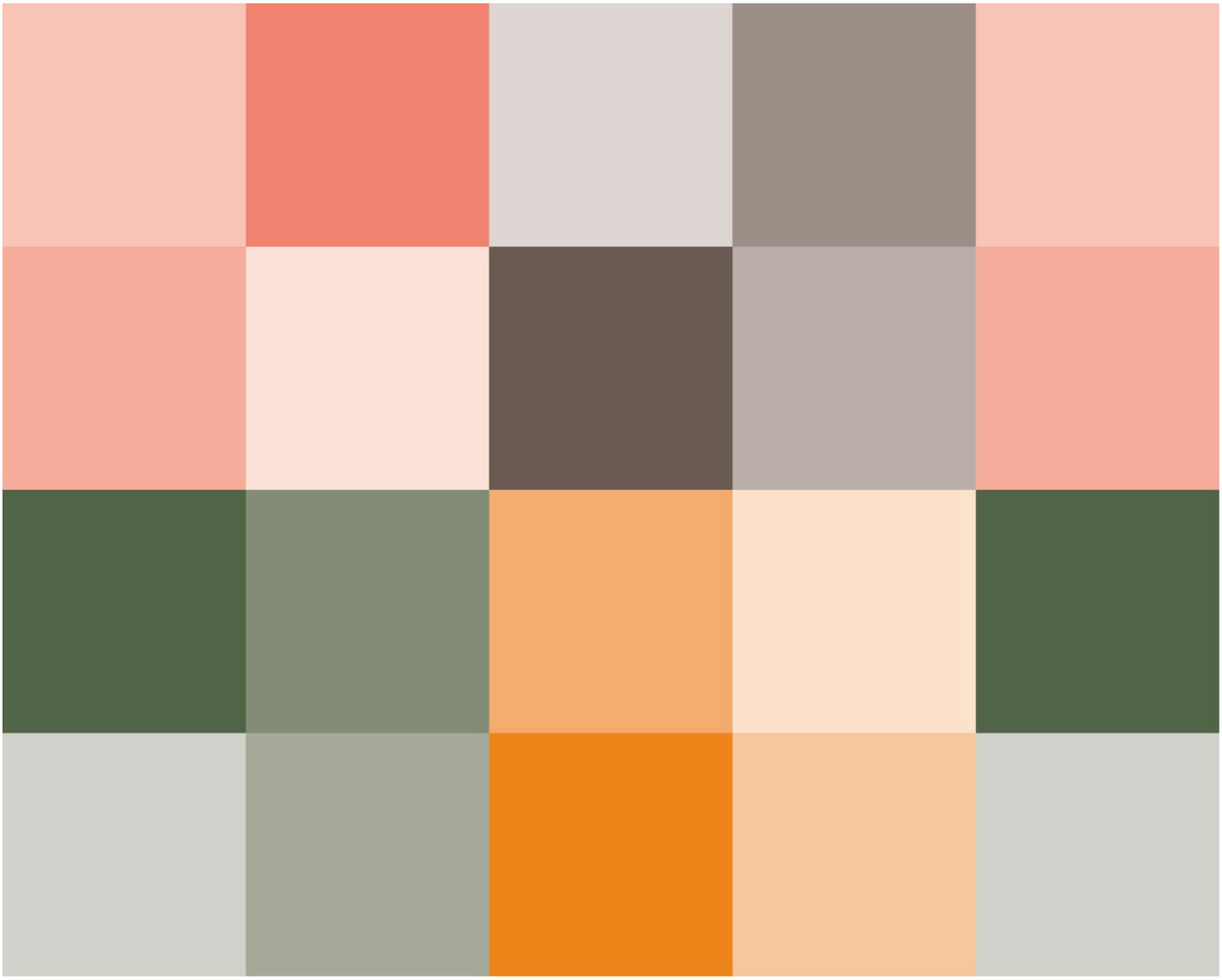


Referensarkitektur för biljettintyg

MYNDIGHETEN FÖR DIGITAL FÖRVALTNING
Agency for Digital Government

Innehållsförteckning

1 Dokumentinformation

1.1 Bakgrund

Covidbeviset togs fram för att öppna upp för resande mellan EU länder på ett säkert sätt under pandemin. Intygen kom senare även att användas nationellt i Sverige för att få gå på event. Covidbeviset fungerar som en biljett som man bär med sig, där kontroll måste fungera även när verifieraren är offline.

Lösningen bygger på ett biljettsystem för kollektivtrafik och möjliggör för snabb och säker verifiering i sammanhang som behöver hantera ett stort flöde av människor. Intygen är kryptografiskt verifierbara genom en digital signatur som finns i QR koden. QR koden innehåller en biljett i dataformatet CBOR och är digitalt signerad (COSE) vilket tillsammans kallas för en CBOR Web token (CWT). COSE signaturen skapas med en asymmetrisk krypteringsnyckel och verifieras med hjälp av den publika nyckeln, vilket säkerställer intygets äkthet och innehållets integritet. Intygslösningen är beprövad och förhållandevis enkel att vidareutveckla och förvalta. Därför bedöms den kunna göra nytta för andra intygsområden inom offentlig sektor.

1.2 Syfte

Referensarkitekturen utgör en abstrakt beskrivning av en lösning för verifierbara intyg baserat på Hcert specifikationen och Sveriges nationella Covidbevislösning. Den utgör en mall som kan fungera som utgångspunkt för att konstruera en konkret och anpassad lösning. Referensarkitekturen för biljettintyg syftar till att stödja den offentliga förvaltningen genom att tillhandahålla en standardiserad modell som inkluderar styrande principer, beskrivningar, definitioner och en konceptuell systembeskrivning av ett biljettsystem. Genom detta kan den underlätta en ökad ömsesidig förståelse mellan tjänsteleverantörer och myndigheter. Referensarkitekturen kan även användas som grund för kravställning i samband med upphandling eller vid vidareutveckling av intygsystem.

1.3 Målgrupp

Detta dokument riktar sig i första hand till lösningsarkitekter, IT-strateger, IT-beslutsfattare samt utvecklare av verifierbara intyg.

1.4 Disposition

Referensarkitekturen är uppdelad på följande avsnitt:

Definitioner och begrepp – Definitioner och begrepp som används i dokumentet och är kopplat till hantering av biljettintyg.

Styrande principer – Krav, principer och beskrivningar som utgör mallen för biljettintyg.

Användningsfall – En beskrivning av de tydligaste användningsfallen för biljettintyg.

Verksamhetsvy – Beskriver behov och drivkrafter som finns hos utfärdare och verifierare av biljettintyg. Samt de verksamhetsförmågor som kan behövas för att utfärda, distribuera och verifiera biljettintyg.

Logisk arkitektur – Beskriver de tekniska förmågor och funktioner som behövs för att realisera verksamhetsförmågorna.

Referenser – Länkar till referensprojekt för återanvändning av kod och EU dokumentation.

1.5 Versionshistorik

Version	Datum	Författare	Kommentarer
1.0	2023-11-01	Martin Bertrandsson	Första utkastet

2 Definitioner och begrepp

Bevis
Dokument eller data, inbegripen text eller ljud, bildinspelningar eller audiovisuella inspelningar, oavsett vilket medium som använts för att bevisa fakta.

Intyg
Ett intyg är ett dokument som bekräftar att en person, organisation eller entitet har uppfyllt vissa krav eller kvalifikationer. Intyg kan utfärdas av en myndighet, ett företag eller en organisation.

Verifierbart intyg
Digitalt eller fysiskt dokument (exempelvis med en QR kod) som kan innehålla olika typer av information, till exempel en persons identitet, utbildningsnivå, förarbevis eller yrkeskvalifikationer. Intygen är verifierbara genom att de är signerade med en digital signatur som kan användas för att bekräfta att intyget är äkta och inte har manipulerats.

Biljettintyg
Verifierbara intyg som utfärdas i CBOR Web Token format. CBOR används för att optimera datamängden för generering av en QR kod. QR koden möjliggör för utskrift och användning i pappersformat.

CBOR
CBOR, eller Concise Binary Object Representation, är ett binärt dataformat som används för att representera data. Det är ett kompakt och effektivt format som är lämpligt för överföring av data från begränsade enheter.

COSE
En COSE-signatur är en digital signatur som använder CBOR (Concise Binary Object Representation) för att representera data. COSE-signaturen använder ett assymetriskt signaturschema.

CWT
En CBOR web token (CWT) är en typ av JSON web token (JWT) som använder CBOR (Concise Binary Object Representation) för att representera data och COSE för att signera och säkerställa äktheten.

Digital signatur
En digital signatur kan användas för att säkerställa ett dokuments äkthet och integritet. Den elektroniska signaturen skapas med en kryptografisk nyckel, ett DSC certifikat och ett assymetriskt signaturschema.

2D kod
En 2D-kod är en tvådimensionell streckkod som kan innehålla mer information än en traditionell streckkod och avläses optiskt, exempelvis i QR eller Aztek kod.

Entitet
Person, organisation eller digital hårdvara som innehar ett unikt id.

Utfärdare
Organisation som utställer, signerar och distribuerar intyget till en innehavare. Kan även återkalla ett intygs status genom att lägga till det i en spärrlista. Distribuerar metadata till ett Verifierbart dataregister (VDR) för Verifierare.

Verifierare
Person eller organisation som kontrollerar bevis i form av biljettintyg med hjälp av en verifieringsapp/tjänst. Verifieringsappen synkar kontinuerligt metadata från en distribueringsnod (VDRen).

Innehavare
Individ, organisation eller entitet som intyget är utställt till.

Decentraliserad identitet
Individer, organisationer och entiteter hanterar sin digitala identitet och attribut kopplat till identitet i en digital identitetsplånbok utan beroende av en central tjänsteleverantör. Self Sovereign Identity (SSI) är ett ramverk som bygger på konceptet runt decentraliserad identitet.

Self Sovereign Identity SSI
Självstyrande identitet eller Self Sovereign identity är ett ramverk för decentraliserad identitet som bygger på de tre pelare.

- W3C Verifiable credentials data format
- Blockkedja
- Identitetsplånbok

W3C Verifiable Credentials
W3C Verifiable Credentials 2.0 är ett data format för digitala verifierbara intyg som utvecklats av World Wide Web Consortium (W3C). Verifiable Credentials (VCs) är

utformade för att vara en säker och portabel metod för att dela personlig information, till exempel namn, födelsedatum och utbildningsnivå.
VC använder kryptografi och digitala signaturer för att säkerställa att informationen i intygen är korrekt och inte har manipulerats. Formatet används för att beskriva och ta fram ekosystem för verifierbara intyg och är en av de tre pelarna inom SSI.

eIDAS
EIDAS är en förordning från EU som syftar till att främja den digitala identitetsprocessen inom EU. Förordningen innehåller regler för elektronisk identifiering, autentisering och betrodda tjänster.

EU-plånboken
EU-plånboken är en digital identitetsplånbok som ska göra det möjligt för EU-medborgare och invånare att lagra och använda sina digitala identitetsuppgifter och attribut. EU-plånboken är baserad på eIDAS-förordningen och kommer att göra det möjligt för EU-medborgare och invånare att autentisera sig och identifiera sig på ett enkelt och säkert sätt när de använder digitala tjänster. Det tekniska förmågorna som krävs för ekosystemet är snarlik de som finns för Självsuvurän identitet (SSI).

DSC – Document Signer Certificate
Digitalt certifikat som används för att signera och verifiera den digitala signaturen på ett dokument och bekräfta dess äkthet och ursprung. Dessa certifikat används i elektroniska och digitala sammanhang för att säkerställa att ett dokument har signerats av rätt person eller entitet och att dokumentet inte har ändrats sedan det signerades.

CA - Certificate Authority
En certifikatmyndighet är en organisation eller tjänst som utfärdar och revokerar digitala certifikat baserat på kryptografiska metoder.

PKI - Public Key Infrastracture
PKI är processer runt nyckelceremoni, certifikathantering och tekniska tjänster för att utfärda, revokera och distribuera certifikat som bygger på assymetrisk kryptering.

HSM – Hardware Security Module
En hårdvarumodul (HSM) kan användas för att skydda privata nycklar som används för att skapa certifikat eller digitala signaturer.

Revokera
Lägga till ett intyg eller certifikat i en spärrlista (återkalla ett intyg).

VDR - Verifierbart Dataregister
Distribueringsnod för metadata exempelvis publika nycklar, spärrlistor, regler som

används av verifierare. Begreppet kommer ifrån W3C Verifiable Credential 2.0 standarden. Kallas för ”Trustpoint” inom Covidbevis. Inom decentraliserad identitet (SSI) används distribuerad databasteknik (DLT) för att distribuera metadata.

3 Styrande principer

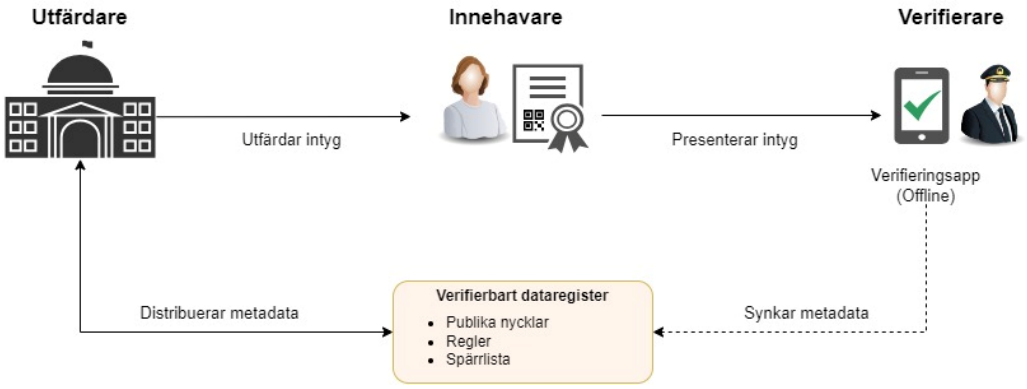
Styrande principer är grundläggande riktlinjer som styr utformningen och implementeringen av biljettintyg. Principerna baseras på Hcert specifikationen för EU DCC (Covidbevis).

3.1 Grundprinciper för alla biljettintyg

- Ska bäras av innehavaren och måste kunna säkert verifieras offline, genom beprövade kryptografiska primitiver.
- Ska gå att använda både digitalt och i pappersformat.
- Dataobjektet ska göras så kompakt som möjligt för optisk transport.

3.2 Roller

I ekosystemet för biljettintyg är aktören som utfärdar biljettintyg en Utfärdare. Personen eller organisationen som erhåller ett intyg är en Innehavare och presenterar sitt biljettintyg för en Verifierare. Detta illustreras i nedan informationsflödesskiss och bygger på samma principer som i W3C Verifiable Credentials standarden. Verifieringsappen hämtar metadata från ett Verifierbart dataregister (VDR), såsom publika nycklar, regler och spärrlista. Utfärdaren är ansvarig för att distribuera nödvändiga metadata till VDRen.



Figur 1 - Beskrivning av informationsflöde för biljettintyg

3.3 Dataformat

Biljettintygets dataobjekt ska göras så kompakt som möjligt för optisk transport, exempelvis i en QR kod. CBOR (Concise Binary Object Representation) är ett binärt dataformat som används för att representera data på ett kompakt och effektivt sätt och är baserat på JSON.

Biljettintyg ska struktureras och kodas som en CBOR med en digital COSE-signatur med hjälp av ett asymmetriskt signaturschema enligt COSE-specifikationen (RFC 8152) för att en Verifierare ska kunna kontrollera biljettens äkthet och integritet. Detta kallas tillsammans för CBOR Web token (CWT) och definieras i RFC 8392. CWTs är en profil av JSON Web tokens (JWTs) som är optimerade för begränsade enheter.

3.3.1 CWT struktur

Biljettintygets innehåll transporteras i CWT ”claims” som defineras efter behov. Engelskans ”claim” kan översättas till fält, men båda definitioner används löpande i denna text. Intygets payload claims, header claims och signaturen utgör CWT biljetten. I Figur 2 presenteras strukturen för Hcert specifikationens CWT biljett och i efterföljande avsnitt beskrivs fälten mer i detalj.

- Protected Header
  - Signature Algorithm (`alg`, label 1)
  - Key Identifier (`kid`, label 4)
- Payload
  - Issuer (`iss`, claim key 1)
  - Issued At (`iat`, claim key 6)
  - Expiration Time (`exp`, claim key 4)
  - Health certificate (`hcert`, claim key -260)
- Signature

Figure 2 - CWT struktur för Covidbevis

3.3.2 Header claims

3.3.2.1 Signature Algorithm

Signature algorithm (alg) anger vilken algoritm som använts för att skapa signaturen. I Hcert specifikationen finns en primär och en sekundär algoritm definierad. Den sekundära algoritmen bör endast användas om den primära algoritmen inte är acceptabel inom de regler och föreskrifter som gäller för utfärdaren.

Primär algoritm: Den primära algoritmen är Elliptic Curve Digital Signature Algorithm (ECDSA) enligt definitionen i (ISO/IEC 14888-3:2006) avsnitt 2.3, med hjälp av P-256-parametrarna enligt definitionen i bilaga D (D.1.2.3) i (FIPS PUB 186-4) i kombination med SHA-256-hashalgoritmen enligt definitionen i (ISO/IEC 10118-3:2004) funktion 4.

Det motsvarar COSE-algoritm-parametern: ES256.

Sekundär algoritm: Den sekundära algoritmen är RSASSA-PSS enligt definitionen i (RFC 8230) med en modul på 2048 bitar i kombination med SHA-256-hashalgoritmen enligt definitionen i (ISO/IEC 10118-3:2004) funktion 4.

Det motsvarar COSE-algoritm-parametern: PS256.

3.3.2.2 Key Identifier

Fältet Key identifier (kid) kan användas av en Verifierare för att välja rätt offentliga nyckel från en lista med nycklar. Flera nycklar kan användas parallellt av en Utfärdare av administrativa skäl och vid utförande av nyckelrullningar. Fältet är valbart och ska endast användas om det finns ett behov av att särskilja utfärdares nycklar på ett effektivt sätt.

I Hcert specifikationen beräknas den fram vid konstruktion av listan över betrodda offentliga nycklar från DSC-certifikatet och består av ett trunkerat (första 8 bytes) SHA-256-fingeravtryck av DSC:n kodad i DER (rå) format. Förkortningen är gjord av utrymmesbesparande själ. På grund av förkortningen av identifieraren finns det en liten men icke-noll chans att den övergripande listan med DSC:er som accepteras av en validerare kan innehålla DSC:er med dubbla kids. Av denna anledning måste en verifierare kontrollera alla DSC:er med det kid.

3.3.3 Payload claims

Payload claims är valbara och väljs med omsorg utifrån de krav som finns på biljettintyget. I hcert specifikationen har man valt att lägga vissa fält i intygets hcert claim i stället för att använda redan registrerade claims. Det beror främst på hur man avser att organisera informationen i ”biljetten”. En biljett med ett litet innehåll kan klara sig med redan registrerade CWT claims.

3.3.3.1 Utfärdare

Fältet för utfärdare ”iss” (Issuer, claim key 1) är ett strängvärde som för Covidbevis innehöll ISO 3166-1 alfa-2-landskoden för den enhet som utfärdar biljettintyget. Detta fält kan användas av en Verifierare för att identifiera vilken uppsättning DSC:er som ska användas för validering.

3.3.3.2 Giltighetstid

Fältet för giltighetstid ”exp” (Expired, claim key 4) skall innehålla en tidsstämpel i integer NumericDate-format (enligt RFC 8392 avsnitt 2) som anger hur länge intyget skall anses vara giltigt. Syftet med giltighetstidsparametern är att tvinga fram en begränsning av giltighetstiden. Giltighetstiden får inte överstiga giltighetstiden för DSC:n.

3.3.3.3 Utfärdat datum

Fältet för utfärdat datum ”iat” (Issued at, claim key 6) skall innehålla en tidsstämpel i integer NumericDate-format (enligt RFC 8392 avsnitt 2) som anger den tidpunkt då intyget skapades. Issued At (iat) får inte föregå giltighetstiden för DSC:n. Verifierare kan tillämpa ytterligare regler i syfte att begränsa giltigheten för biljettintyget baserat på utfärdandedatumet.

3.3.3.4 Unik identifierare

Fältet ”cti” (CWT Id, claim key 7) kan användas för att definiera en unik identifierare för CWT:n. Identifieraren måste tilldelas på ett sätt som säkerställer

att det är en försumbar sannolikhet att samma värde av misstag tilldelas ett annat dataobjekt. Fältet är valbart och används ej i Hcert specifikationen. Organiseras istället tillsammans med intygets objekt med nyckeln ”ci”.

3.3.3.5 ”Not before”

Fältet ”nbi” (Not before, claim key 5) identifierar tiden före vilken CWT:n inte får accepteras för bearbetning. Bearbetningen kräver att det aktuella datumet/tiden måste vara efter eller lika med not-before-datum/tid som anges i claimen. Verifierare kan tillåta en liten marginal, vanligtvis inte mer än några minuter, för att ta hänsyn till klocksynkroniseringsproblem. Värdet måste vara ett tal som innehåller ett NumericDate-värde. Fältet är valbart och används ej i Hcert specifikationen.

3.3.3.6 Subjekt

Fältet ”sub” (Subject, claim key 2) identifierar personen som är föremål för CWT:n. Subjekt-värdet måste antingen vara begränsat till att vara lokalt unikt i utfärdarens kontext eller vara globalt unikt. Fältet är valbart och används ej i Hcert specifikationen.

3.3.3.7 Mottagare

Fältet för mottagare ”aud” (Audience, claim key 3) identifierar de mottagare/verifierare som CWT:n är avsedd för. Varje Verifierare som avser att bearbeta CWT:n måste då identifiera sig själv med ett värde i mottagarkravet. Fältet är valbart och används ej i Hcert specifikationen.

3.3.3.8 Intygets payload

Om payloaden för intyget innehåller unik information eller mer information än de registrerade claims som finns så kan detta behöva transporteras i en unik claim för intyget. Man kan välja att registrera en publik claim för att minska risken för kollisioner av andra oregistrerade claims eller använda en privat claim. Vilka claim keys som är lediga går att hitta på *CBOR Web Token (CWT) Claims (iana.org)* .

För Covidbevis registrerades en publik claim ”hcert” med claim key -260, som kan återanvändas för andra hälsointyg. Hcert är ett JSON-objekt (RFC 7159) som

innehåller information om intyget. Strängar i JSON-objektet bör vara NFC-normaliserade enligt Unicode-standarden.

3.3.4 Signatur

COSE signaturen skapas genom att generera en hashsumma av CWT objektet, som sedan signeras med hjälp av privat nyckel, DSC certifikat och vald signeringsalgoritm (primärt ECDSA).

3.4 Intygets payload

Intygets payload kan kodas i ett JSON objekt (RFC 7159) om det innehåller mer information än de registrerade claims som CWT standarden erbjuder. Även om varje intyg har olika krav på innehåll, så finns det ett antal principer från EU DCC schemat som kan återanvändas.

Generella principer

Eftersom alla personer som vistas i landet inte har ett person/samordningsnummer, så kunde man inte använda det som personidentifierare i biljettintyget för Covidbevis. I stället användes namn och födelsedatum som matchades mot en giltig ID handling för att säkerställa riktighet. Detta kan ses som en bra princip om man vill inkludera så många som möjligt, men gör det samtidigt svårare för en verifierare som måste manuellt granska en ID handling och matcha mot intygets personuppgifter.

Om intyget ska användas internationellt, behöver namn även translittereras. enligt ICAO 9303 MRTD transliteration rules

JSON fält bör hållas nere för att optimera storleken till QR koden, exempelvis givenName = gn.

Exempel JSON dokument från EU DCC på vaccinerad person.


```
{
        "ver": "1.3.0",
        "nam": {
                "fn": "Lövström",
                "fnt": "LOEVSTROEM",
                "gn": "Oscar",
                "gnt": "OSCAR"
        },
        "dob": "1958-11-11",
        "v": [
                {
                "tg": "840539006",
                "vp": "1119349007",
                "mp": "EU/1/20/1528",
                "ma": "ORG-100031184",
                "dn": 2,
                "sd": 2,
                "dt": "2021-06-11",
                "co": "SE",
                "is": "Swedish eHealth Agency",
```

```
                    "ci": "URN:UVCI:01:NL:DADFCC47C7334E45A906DB12FD859FB7#1"
                    }
            ]
}
```

Versionsnumret lagras i ”ver” och representerar versionen av schemakodningen.

```
"ver": "1.3.0"
```

Innehavarens namn lagras i elementet ”nam”. ”fn” innehåller efternamnet och ”gn” innehåller förnamnet på innehavaren kodat i valfri UTF-8-alfabetisk karaktär. Fälten ”fnt” och ”gnt” innehåller efternamnet respektive förnamnet translittererat.

```
"nam": {
            "fn": "Smith-Jones",
            "fnt": "SMITH<JONES",
            "gn": "Charles Edward",
            "gnt": "CHARLES<EDWARD"
}
```

Innehavarens födelsedatum ingår i elementet ”dob”. Det är formaterat enligt ISO-8601 och kan innehålla antingen: YYYY, YYYY-MM eller YYYY-MM-DD. Detta datum måste matcha datumet på innehavarens id-handling.

```
"dob": "1964-01-01"
```

Det ska finnas en unik identifierare så det går att identifiera ett biljettintyg. Identifieraren ska inte baseras på personidentifierare eller verksamhetsspecifika data. I Covidbevisets JSON schema kallas detta för Certificate id (ci) och genereras enligt följande:

UVCI:{version}:{country}:{random}:{checksum}

```
"ci": "URN:UVCI:01:NL:DADFCC47C7334E45A906DB12FD859FB7#1"
```

3.5 Flöde för att generera en 2D kod

I Figur 3 nedan beskrivs flödet för att generera en 2D kod med en digital signatur som en Verifierare kan kontrollera med en mobilapp som innehåller den publika nyckeln.

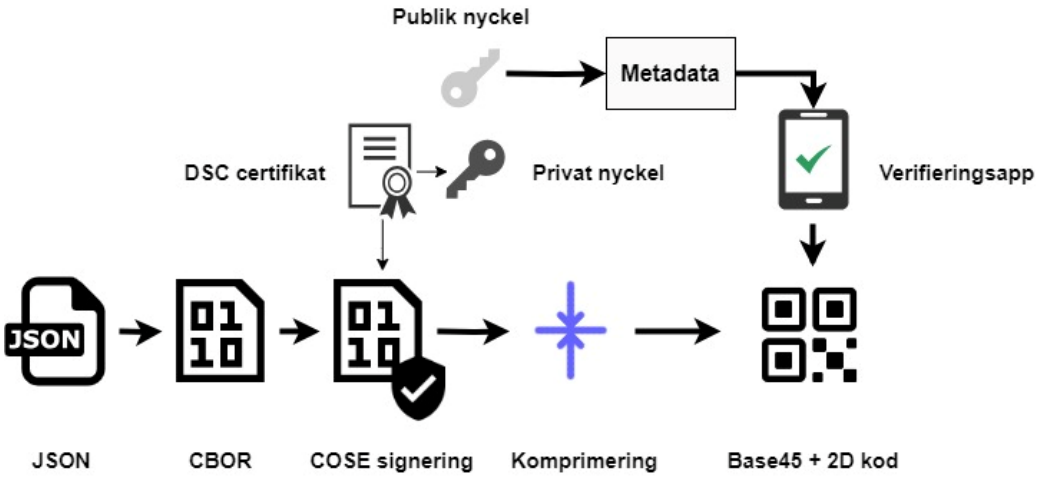


Figure 3 - Flöde för att skapa en 2D kod

Steg 1 – Skapa JSON

Intygets grunddata hämtas in, aggregeras och skapas i JSON format.

Steg 2 – CBOR

Intygets JSON dokument och övriga metadata sätts samman i ett set av CBOR Webtoken claims och kodas om till binär form.

Steg 3 - COSE signering

COSE (CBOR Object Signing and Encryption) är en understandard till CBOR som tillhandahåller stöd för digital signering av CBOR-objekt. Signering utförs för att en Verifierare ska kunna säkerställa att CWT token är autentisk och inte har manipulerats. För att skapa en digital signatur behövs ett DSC certifikat och motsvarande privata nyckel.

Steg 4 – Komprimering

För att minska storleken och förbättra hastigheten och tillförlitligheten vid läsning av 2D koden skall CWT:en komprimeras.

Steg 5 – BASE45 + 2D kod

Base45 encoding används för att generera 2D koder eftersom den är en effektiv och robust metod för att representera binära data i en ASCII-baserad sträng. Den använder en 45-teckens alfabet som består av siffror, bokstäver och ett par symboler. Detta gör att den kan representera data i en mindre mängd utrymme än andra kodningsmetoder.

Man kan lägga till ett prefix för att identifiera den alfanumeriska koden, exempelvis HC1 (Health Certificate 1). Detta gör att verifierare kan kunna upptäcka vilken typ av data som kodats och välja rätt avkodnings- och bearbetningsschema, när det finns behov av att verifiera olika typer av intyg.

3.6 Optisk verifering

För att bättre hantera äldre utrustning som är utformad att fungera med ASCII-payloads kodas den komprimerade CWT:n som ASCII med Base45 innan den kodas in i en 2D-kod.

Valet mellan Aztek och QR-kod bör göras med hänsyn till de specifika behov och förutsättningarna för intyget. Om intygen ska verifieras med en mobilkamera så är det lättare för en mobilkamera att skanna av en QR kod. Om man istället använder sig av en streckkodsavläsare, så anses Aztek vara det bättre valet. QR-koder har förmågan att hantera något mer data, inklusive bilder. Under Covidbevisprojektet testades båda varianter, där QR koden i slutändan var det bästa och snabbaste alternativet vid olika tester då verifierarna i störst utsträckning använde mobilkamera.

Figur 4 - *Exempel Aztec kod*

Figur 5 - *Exempel QR kod*

QR-formatet enligt definitionen i (ISO/IEC 18004:2015) kan användas för generering av 2D-streckkoder. En felkorrigeringshastighet på "Q" (cirka 25 %) rekommenderas. Alfanumerisk (läge 2/QR-kodssymboler 0010) måste användas tillsammans med Base45.

För att verifierare ska kunna upptäcka vilken typ av data som kodats och välja rätt avkodnings- och bearbetningsschema ska de Base45-kodade data (enligt Hcert specifikation) föregås av en kontextidentifierare "HC1:". Nya versioner av biljettintyg kan behöva definiera en ny kontextidentifierare, medan tecknet efter "HC" SKALL tas från teckenuppsättningen [1-9A-Z]. Ordningen av inkrementet är definierad att vara i den ordningen, d.v.s. först [1-9] och sedan [A-Z].

Det rekommenderas att den optiska koden återges på presentationsmediet med en diagonal storlek mellan 35 mm och 60 mm för att rymma läsare med fast optik där presentationsmediet måste placeras på läsarens yta.
Om den optiska koden trycks på papper med skrivare med låg upplösning (< 300 dpi) måste man se till att varje symbol (punkt) i QR-koden representeras exakt fyrkantigt. Icke-proportionell skalning kommer att resultera i att vissa rader eller kolumner i QR:en har rektangulära symboler, vilket kommer att försvåra läsbarheten i många fall.

3.7 Tillitsmodel

För nya implementationer av biljettintyg bör en tillitsmodell tas fram för att tydliggöra de grundprinciper som varje aktör i ekosystemet måste förhålla sig till. Tillitsmodellen kan beskriva tekniska funktioner och principer för skapande och distribuering av certifikaten som används för att signera och verifiera intygen. Det kan till exempel handla om vilket signaturschema som ska användas, giltighetstider på DSC certifikaten och huruvida intygen ska gå att återkalla.

För Covidbevis sattes en PKI lösning upp med en dedikerad CA för utfärdande av DSC certifikat och en nationell VDR för att distribuera publika nycklar och metadata via en EU nätsluss som byggde på konceptet ICAO Master list (elektroniska pass). För nationella applikationer av biljettintyg, kan det räcka med att publicera en lista med giltiga DSC certifikat som kan nås av verifieringsapparna. Huruvida man använder sig av en dedikerad, delad eller kommersiell CA tjänst är något som bör motiveras och framgå i tillitsmodellen. Detta är bara några exempel på krav och principer som bör finnas i en tillitsmodell.

3.8 Säkerhetsprinciper

Intyget giltighetstid

För att minska behovet av att behöva återkalla intyg, är det rekommenderat att sätta en begränsning på intygets giltighetstid. Den acceptabla giltighetstiden kan bestämmas av praktiska begränsningar. Till exempel kanske en resenär inte har möjlighet att förnya ett intyg under en resa utomlands. Giltighetstiden på ett intyg får aldrig överskrida DSC certets giltighetstid.

Validering av inhämtade data

För att minimera riskerna förknippade med denna attackvektor bör alla inmatningsfält valideras ordentligt med avseende på datatyper, längder och innehåll.

Nyckelhantering

Utfärdandetjänsten bör använda en hårdvarumodul (HSM) för att hantera privata nyckeln som används tillsammans med certifikatet för att signera datamängden i QR-koden. Detta innebär att inga nycklar finns som kan spridas utan de skyddas i en hårdvara.

Stöld, Dataläckor

Skyddat mot säkerhetsrelaterade händelser som exempelvis stöld, dataläckor. Intyget bör kombineras med giltig ID handling för att säkerställa ägarskap och riktighet.

4 Användningsfall

Biljettintyget kan användas för alla typer av verifierbara intyg, men är främst en lösning då det finns krav på pappersformat. Det är ett intyg som innehavaren bär med sig och där verifieraren har behov av att kunna verifiera även offline. Krav på pappersformat används oftast i relation till att inkludera fler människor i annars digitala lösningar, exempelvis dom som saknar e-legitimation, svenskt personnummer eller har någon slags kognitiv funktionsnedsättning. Ibland finns det behov av att skriva ut sin biljett, exempelvis vid långa resor då man inte kan garantera tillgång på digital infrastruktur eller ström. Eftersom intygen är kryptografiskt verifierbara även offline, passar dom för scenarios där man inte kan lita på uppkoppling till internet eller den digitala infrastrukturen.

Exempel:

- Intyg för kriser och katastrofer
- Intyg som man bär med sig på platser med opålitlig digital infrastruktur, exempelvis resor och event.

Biljettintyget är en förhållandevis enkel lösning att vidareutveckla och förvalta, framför allt om intyget endast ska användas i en nationell kontext då det förenklar distribueringen av metadata för verifiering.

Begränsningar

Lösningen har idag inte stöd för krav som lyfts i nya eIDAS förordningen kopplat till selektivt avslöjande, eftersom intyget i sin helhet måste finnas i QR koden för att kunna skrivas ut i pappersformat.

Intygets storlek begränsas av QR koden. QR koder kan maximalt innehålla 2953 bytes (~3kbyte) data. Dataobjektet kan visserligen delas upp på flera QR koder.

Intygen bör kontrolleras mot ett giltig ID handling för att säkerställa riktighet. Detta slår på effektivitet i verifieringsprocessen i jämförelse med en identitetsplånbok.

5 Verksamhetsvy

Verksamhetsvyn beskriver vanliga behov och drivkrafter samt förmågor som behöver finnas hos organisationer kopplat till hantering av biljettintyg. Verksamhetsförmågorna driver de tekniska förmågor och tjänster som presenteras i kapitel 5 (Logisk arkitektur). Avsnittet är uppdelat på *Utfärdare* och *Verifierare* för att tydliggöra de drivkrafter som finns hos respektive aktör.

5.1 Utfärdare av biljettintyg

5.1.1 Verksamhetsbehov och drivkrafter

Kostnadsbesparingar: Minskning av kostnader relaterade till hantering av intyg och bevis inom offentlig sektor. Detta kan ske genom att öka graden av automatiserade flöden, genom självprovisionering via e-tjänster, distribution via brevlådeoperatör eller via en utskriftstjänst.

Säkerhet: Försvåra möjligheten att förfalska eller förvanska intyg.

Förvaltningsbarhet: Ett system som är kostnadseffektivt, enkelt att administrera, vidareutveckla och förvalta.

Regulatoriska krav och dataskydd: Uppfylla lagkrav och skydda känslig information.

Användarvänlighet: Det ska vara lätt att beställa, använda och återskapa ett biljettintyg om den förloras.

Tillgänglighet och inkludering: Säkerställa att lösningen är tillgänglig för alla och inkluderande. Även för personer utan e-legitimation eller med kognitiva funktionsnedsättningar.

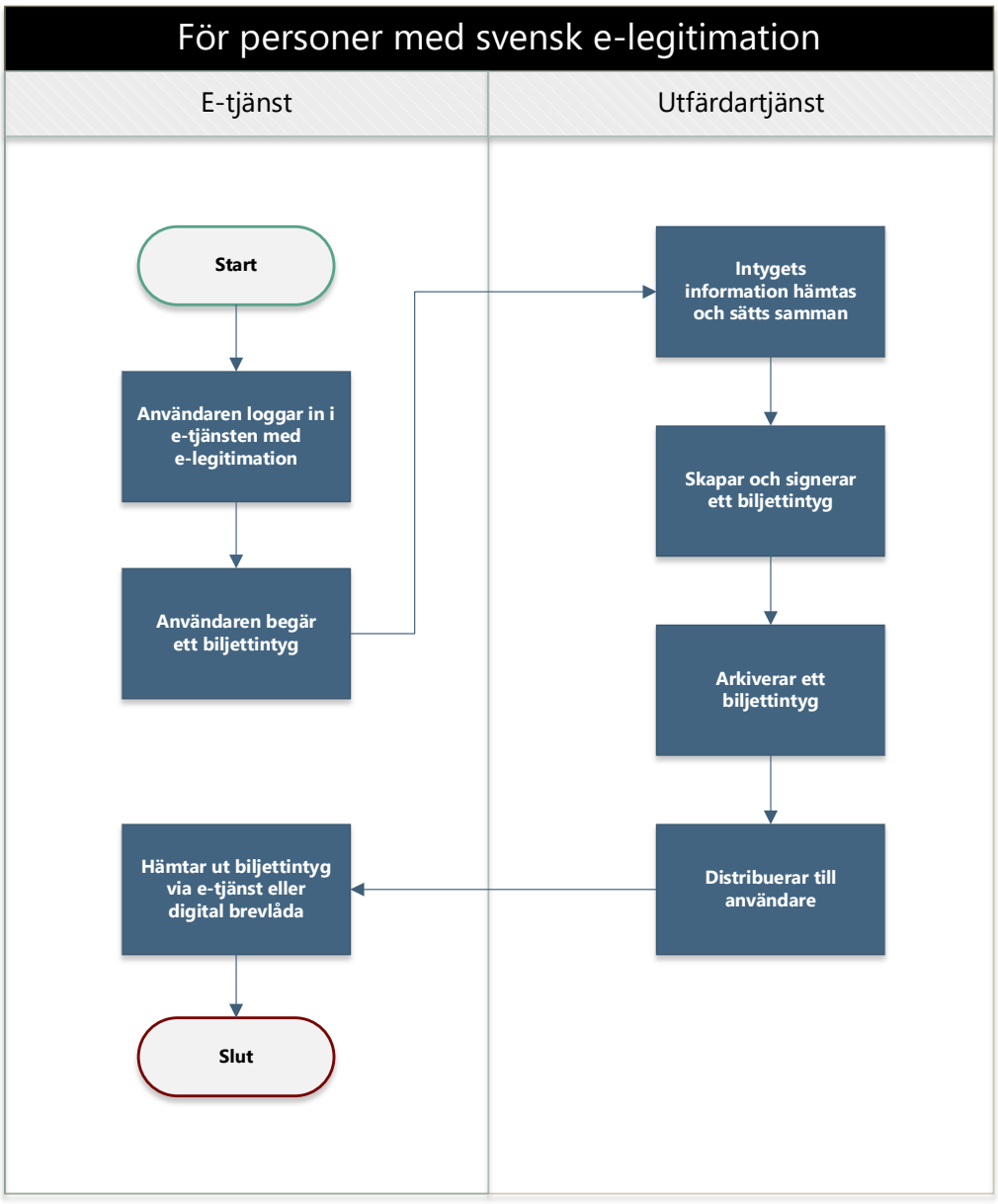
5.1.2 Förmågor och flöden

I följande avsnitt beskrivs vanliga förmågor som *Utfärdare* behöver och de informationsflöden och scenarios som detta resulterar i.

5.1.2.1 Skapa och distribuera biljettintyg

Förmågan beskrivs i flödesdiagrammen nedan utifrån följande scenarios:

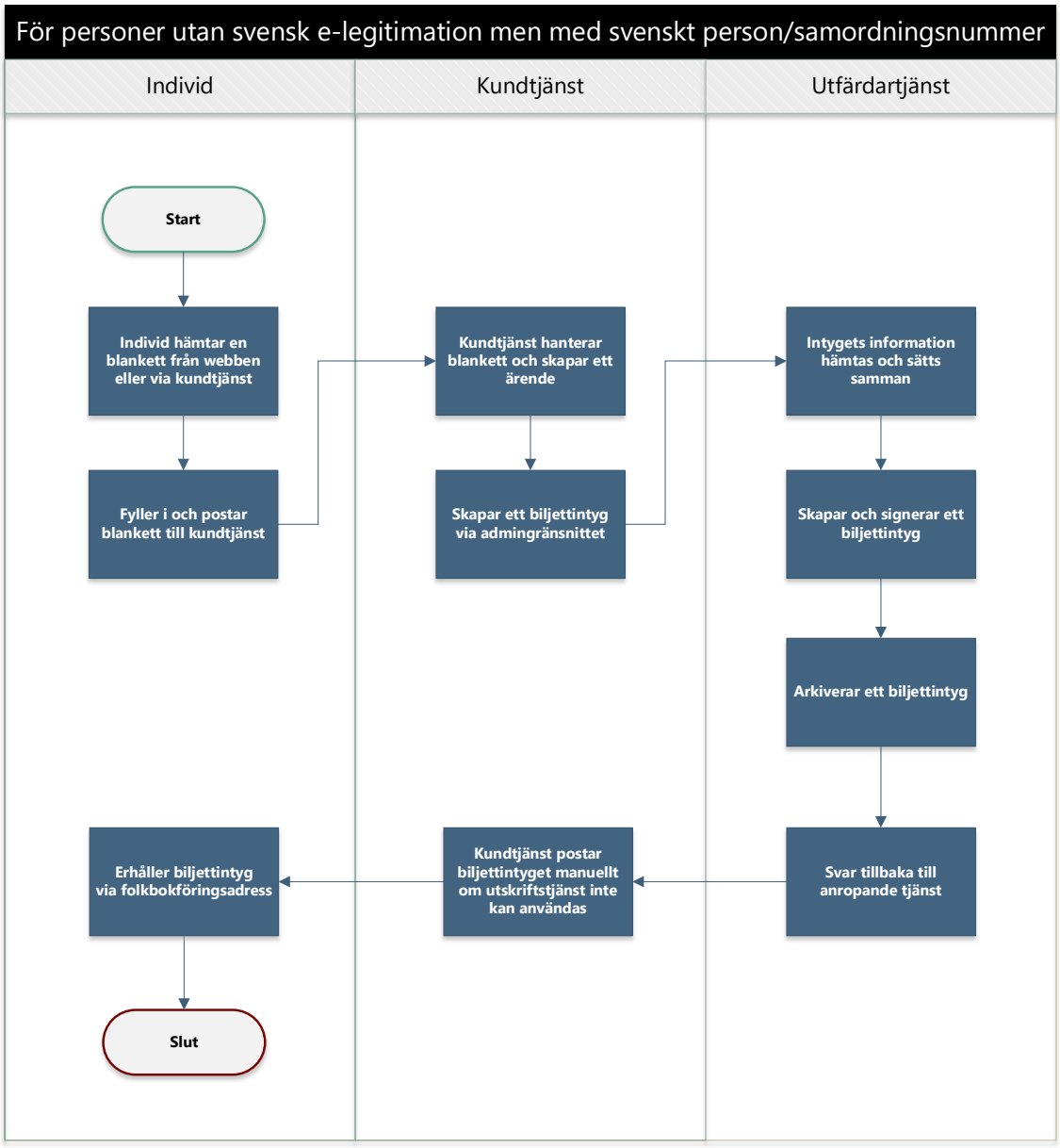
1. Normalflöde för personer med svensk e-legitimation



1

Figur 2 - Skapa och distribuera biljettintyg, Scenario 1

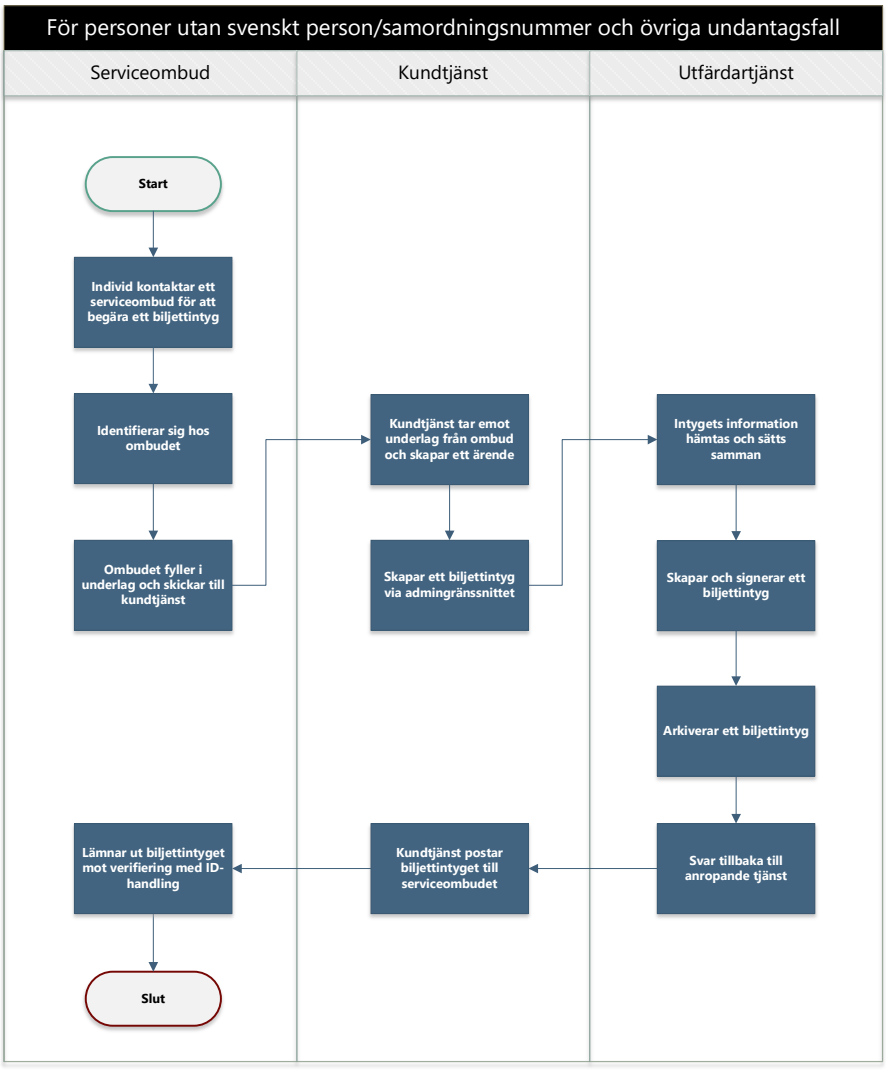
2. För personer utan svensk e-legitimation men med svenskt person/samordningsnummer



Figur 3 - Skapa och distribuera biljettintyg, Scenario 2

3. För personer utan svenskt person/samordningsnummer och övriga undantagsfall.

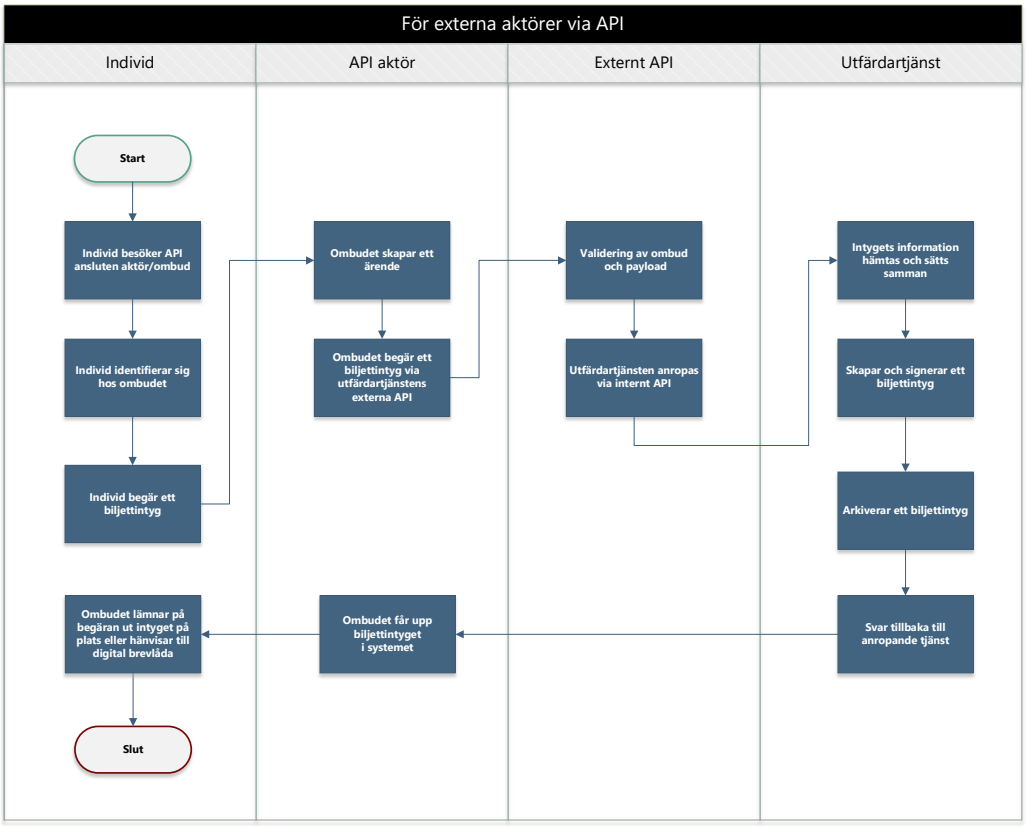
Detta scenario baseras främst på de individer som vistades i Sverige under pandemin men som saknades svenskt person eller samordningsnummer, vilket begränsande deras möjligheter att använda digitala lösningar. Det gäller även hantering av de fall där grunddatakällan saknar korrekt information om individen och kräver manuell hantering.



Figur 4 - Skapa och distribuera biljettintyg, Scenario 3

4. För externa aktörer via API

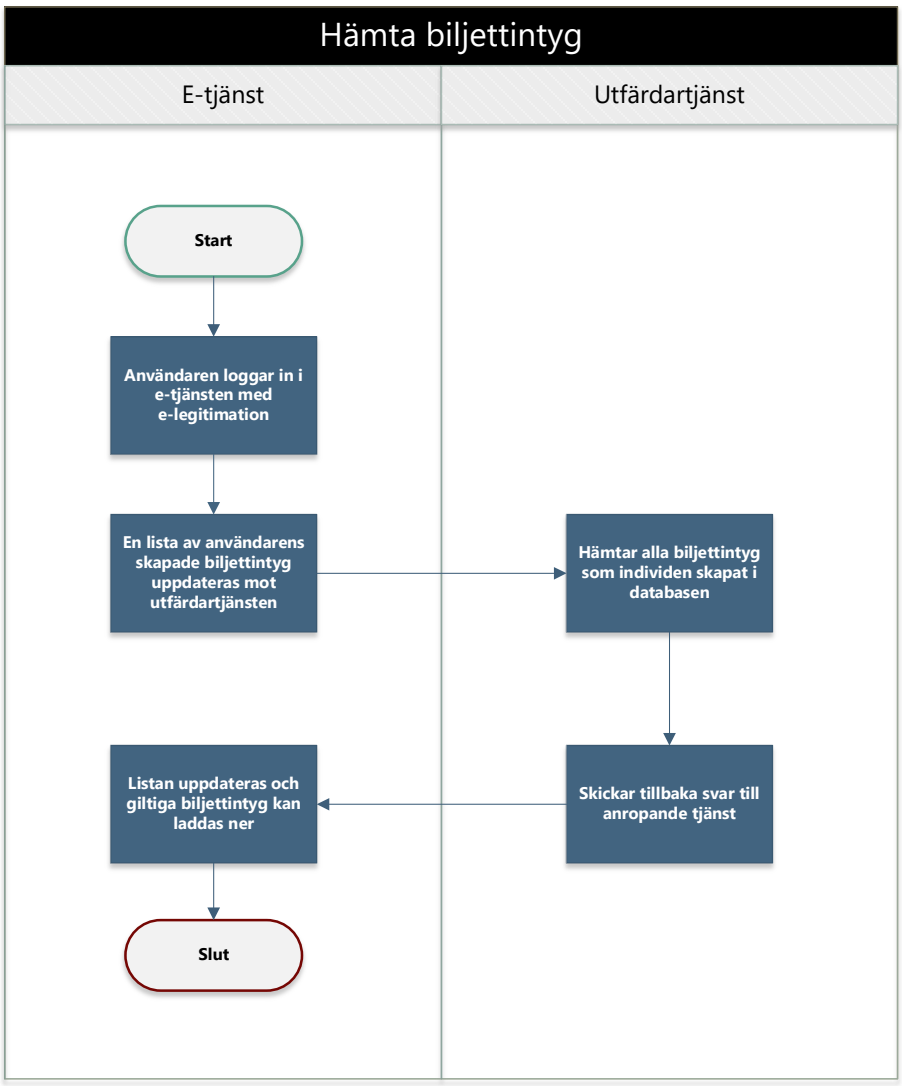
Detta scenario bygger på de tester som vårdgivare utförde under Covidbevis och de efterföljande testbevisen som aktörerna kunde generera på begäran av den testade. Genom ett externt API kunde testaktören skapa ett Covidbevis på begäran av den testade, vid negativt resultat.



Figur 5 - Skapa och distribuera biljettintyg, Scenario 4

5.1.2.2 Hämta biljettintyg

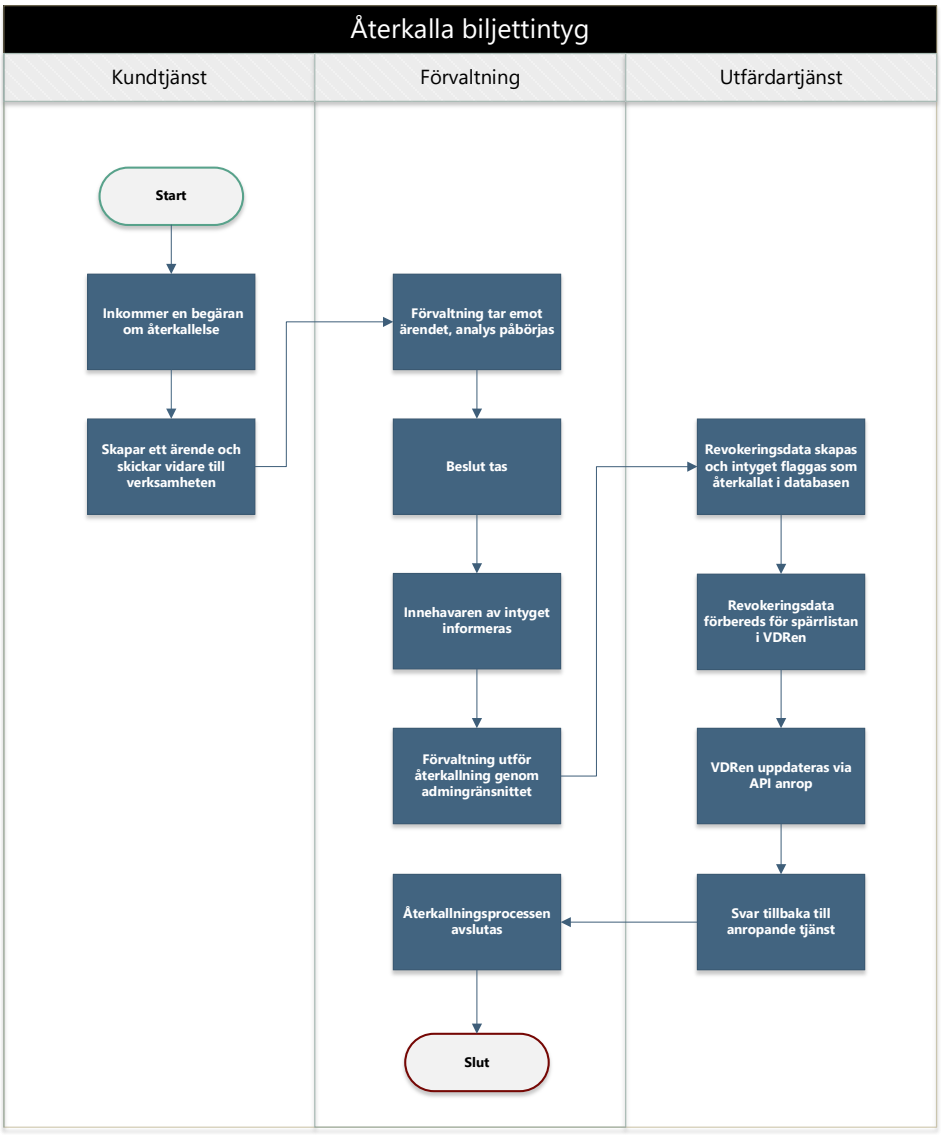
Om man vill begränsa hur många intyg som får skapas av användare, så behövs det en funktion som kan hämta och lista giltiga intyg, samt en funktion där enskilda intyg kan laddas ner från listan. Då kan användaren enkelt hämta intyget på nytt via e-tjänsten ifall det tappats bort. Hur lång tid det måste gå innan användaren tillåts skapa ett nytt intyg behöver beslutas. En rimlig frekvens kan exempelvis vara 1 biljettintyg/dag. För att denna funktion ska vara möjlig måste det finnas rättsligt stöd att spara/arkivera biljettintygen.



Figur 6 - Hämta biljettintyg

5.1.2.3 Återkalla biljettintyg

Verksamhetsförmågan omfattar både administrativa och tekniska åtgärder. På administrativ nivå behöver verksamheten införa en ärendehantering enligt förvaltningslagen. Flödet kan initieras av enskilda personer, andra myndigheter eller om myndigheten upptäcker att ett bevis har utfärdats felaktigt eller på olaglig väg. På teknisk nivå behöver man markera arkiverade intyg som återkallade, samt lägga till intygen i en spärrlista som kan distribueras till verifieringsappar.



Figur 7 - Återkallelse av biljettintyg

Schematisk beskrivning

1. **Inkommer en begäran om återkallelse**
   Kundtjänst får en begäran som innehåller underlag för att återkalla beviset med en anledning/orsak.

2. **Skapar ett ärende**
   Kundtjänst skapar ett ärende och dokumenterar ovanstående i ett ärendehanteringssystem. Det ska även fungera för spårbarhet vid eventuella felhanteringar. Skickar till förvaltning.

3. **Förvaltning tar emot ärendet, analys påbörjas**
   Förvaltning ska fungera som en kontrollinstans för att analysera och säkerställa att återkallelsen sker på rätt grunder.

4. **Beslut tas**
   Förvaltning behöver besluta ifall anledningen/orsaken är tillräckligt för att kunna revokera samt efter att analysen är genomförd.

5. **Innehavaren av beviset informeras**
   Myndigheten kontaktar innehavaren av beviset och informerar om att beviset kommer att återkallas (ifall det är möjligt). Ärendehanteringen måste följa förvaltningslagen, vilket innebär att den enskilde behöver få tillfälle att yttra sig innan beslut fattas (som huvudregel). När beslut fattats måste denne få information om hur beslutet kan överklagas samt information om personuppgiftsbehandlingen.

6. **Förvaltning utför återkallning genom admingränsnittet**
   Admingränsnittet har en funktion för att återkalla biljettintyg.

7. **Återkallningstjänsten skapar revokeringsdata**
   a) Intyget flaggas som återkallat i databasen
   b) Revokeringsdata förbereds för spärrlista i VDRen
   c) VDR uppdateras via API
   d) Svar tillbaka till anropande system

8. **Återkallningsprocessen avslutas**

5.1.2.4 Hantera DSC certifikat för signering av biljettintyg

Det behöver finnas en verksamhetsprocess runt skapande och återkallande av DSC certifikat som används för signering av biljettintyg. Det är vanligt att man växlar in nya privata nycklar i samband med detta (nyckelrotation), vilket inkluderar en nyckelceremoni. Det är starkt rekommenderat att skydda och hantera privata nycklar för signering i en HSM.

Exempelflöde för nyckelceremoni med HSM:

1. Deltagare som ska bevittna nyckelceremonin bjuds in enligt den process som finns för nyckelceremonin.
2. En HSM-klient konfigureras för kommunikation med HSM, klientcertifikat för MTLS och HSM-konfiguration läggs in.
3. Verifiering görs så att HSM klient kan kommunicera med HSM.
4. Nyckeladministratör skapar ny nyckel i HSM och genererar en CSR (Certificate Signing Request).
5. Hämtar ut CSR fil och konfiguration.
6. Loggar in i CA tjänstens webbgränssnitt och genererar ett Document Signer certifikat (DSC) av CSR filen. Alternativt skickas CSR filen till en CA administratör för utförande.
7. Uppdaterar utfärdartjänst med nytt DSC certifikat

5.1.2.5 Hantera och distribuera regler

Denna verksamhetsförmåga är kopplad till behov som kanske främst finns för hälsointyg där forskning kan förändra regelverket över tid. Under Covidbevis projektet uppstod en mosaik av olika regelverk kopplat till inreseregler samt eventbesök i takt med att forskningen och folkhälsomyndigheternas (inom EU) rekommendationer ändrades. De nya reglerna behövde distribueras ut snabbt till verifieringsapparna. Apparna kunde då verifiera att attributen i intygen uppfyllde gällande regelverk i landet.

5.1.2.6 Hämta och presentera statistik

Erfarenheterna runt Covidbevis visade hur viktigt det är att kunna ta ut statistik. Det kan behövas som underlag för uppföljning, samt för att kunna analysera användandet. För att använda data i rent statistiskt syfte (utan risk för gallring)

kan det vara nödvändigt ur ett rättsligt perspektiv att avpersonifiera den och i stället använda sig av demografiska termer som åldersgrupper, kön och postort.

5.1.2.7 Arkivera och gallra

Respektive myndighet som ska införa en intygsinfrastruktur måste självständigt beakta de rättsliga förutsättningarna för att säkerställa att informationshanteringen sker i förenlighet med gällande rätt. I fallet med Covidbevis så hade ansvarig myndighet initialt gjort bedömningen att man inte fick spara intygen alls. Sedan bedömdes det i enlighet med Tryckfrihetsförordningen andra kapitel gällande att man som myndighet expedierat en handling, så ska varje bevis/intyg arkiveras som en allmän handling. Arkivlagen specificerar mer i detalj hur detta ska gå till. Det ska tilläggas att det finns olika definitioner av allmänna handlingar och i detta fall alltså inte handlar om offentliga handlingar.

I en sista runda gjorde man en ny bedömning och ansåg att det fanns tillräckligt med stöd i EU förordningen för att spara uppgifter under angiven tid för att bibehålla en spårbarhet på bevisen som utfärdats i syfte att kunna kontrollera bevisets äkthet och giltighet. För E-hälsomyndigheten var detta en förutsättning för att kunna utreda och identifiera felaktiga uppsåt för att vara behjälpliga i brottsutredningar. Vidare, behöver myndigheten spara loggar under angiven tid för att kunna felsöka, övervaka och utreda händelser och resultat.

5.1.2.8 Rätten att bli glömd

Huvudregeln enligt artikel 17.1 i dataskyddsförordningen är att den enskilde efter begäran har rätt att få samtliga personuppgifter raderade utan onödigt dröjsmål. En rättslig bedömning behöver göras vid nya tillämpningar av biljettintyg av hur detta ska tillämpas.

5.2 Verifierare av biljettintyg

Verifieraren ska säkerställa att Innehavaren har den behörighet/rättighet som krävs för att få tillträde, enligt gällande regelverk.

5.2.1 Verksamhetsbehov och drivkrafter

Användarvänlighet
Det ska vara lätt för en Verifierare att använda och förstå verifieringsappen.

Effektivitet och snabbhet
Det ska gå snabbt att verifiera en Innehavares intyg.

Tillförlitlighet
En Verifierare ska kunna lita på att resultatet i verifieringstjänsten följer gällande regelverk.

Tydlighet i regelverk
Det ska framgå vilka regler som gäller för godkänt resultat, samt vem som är ansvarig för regelverket. Ansvarig myndighet för regelverket kan exempelvis definieras i tillitsmodellen. Ett intyg kan innehålla flera olika attribut. Vid användning av lång giltighetstid, kan en utfärdare behöva revidera reglerna för vad som anses godkänt eller också så finns det olika användningsfall som har olika regler. För Covidbevis fanns det olika regler för inresa och event. Man fick exempelvis resa till andra länder med testbevis, men man fick inte gå på event med något annat än vaccinationsbevis. För att tillgodose detta behov infördes en regelmotor i veriferingsapparna som kunde synka aktuella regler från VDRen.

Integritet
Hantering av personuppgifter sker i enlighet med gällande rätt. Personuppgiftslagen, GDPR, reglerar behandling av personuppgifter. Den syftar till att skydda människors personliga integritet när deras personuppgifter behandlas, det vill säga samlas in, registreras, lagras och används. Om personuppgifter behöver lagras måste det finnas rättsligt stöd samt det måste framgå varför man lagrar och det måste ges information om detta i både appen och i integritetspolicyn. En anledning till att spara eller skicka vidare verifieringsdata

skulle kunna vara för att granska efterlevnad hos Verifierare eller av statistiska anledningar.

Uppgiftsminimering
Rent juridiskt så är det verifieraren som är ansvarig för den personuppgiftbehandlingen som medföljer då appen används. Informationen som presenteras vid godkänt resultat kan reduceras för att ytterligare minimera personuppgifter som presenteras.
För negativt resultat kan man visa endast Ej Godkänt och en felkod.
För att ytterligare säkra att inte obehörig ska kunna ta del av informationen, så kan resultatsidan automatiskt stängas ner efter en viss tid.

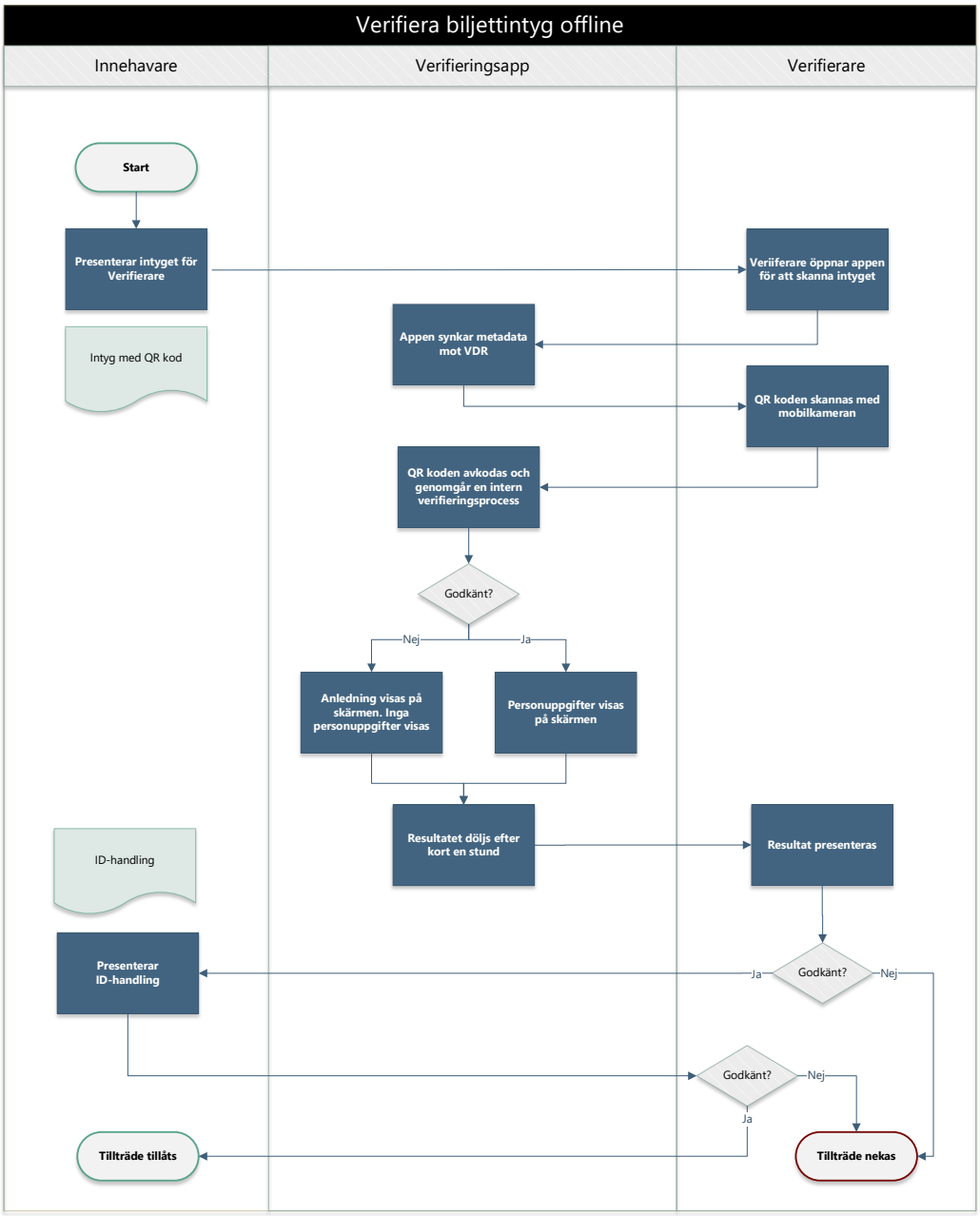
5.2.2 Förmågor och flöden

I följande avsnitt beskrivs de förmågor som en *Verifierare* behöver och de informationsflöden och scenarios som detta resulterar i.

5.2.2.1 Verifiera biljettintyg offline

Biljettintyg måste kunna verifieras då verifieraren är offline. Exempelvis under event, på flygplatser, bussar, i marina miljöer eller på geografiskt fjärran platser. När intygen används för någon form av resande/transport så är det viktigt att man tar höjd för tillfälliga störningar i digital infrastruktur. Detta är huvudscenariot för biljettintygen, där all verifiering sker mot nedladdade metadata såsom publika nycklar, spärrlista och regler. En veriferingsapp bör automatiskt försöka hämta och uppdatera metadata från när den används, men ska fortsatt fungera när kontakten ligger nere. Om den digitala verifieringen blir godkänd måste en giltig

ID handling kontrolleras mot namn och födelsedatum i intyget.



Figur 9 - Verifiera biljettintyg offline

Schematisk beskrivning av ett verifieringsflöde (offline)

1. Innehavare presenterar intyget för Verifierare.
2. Verifierare öppnar appen eller startar mobilkamera för att skanna intyget.

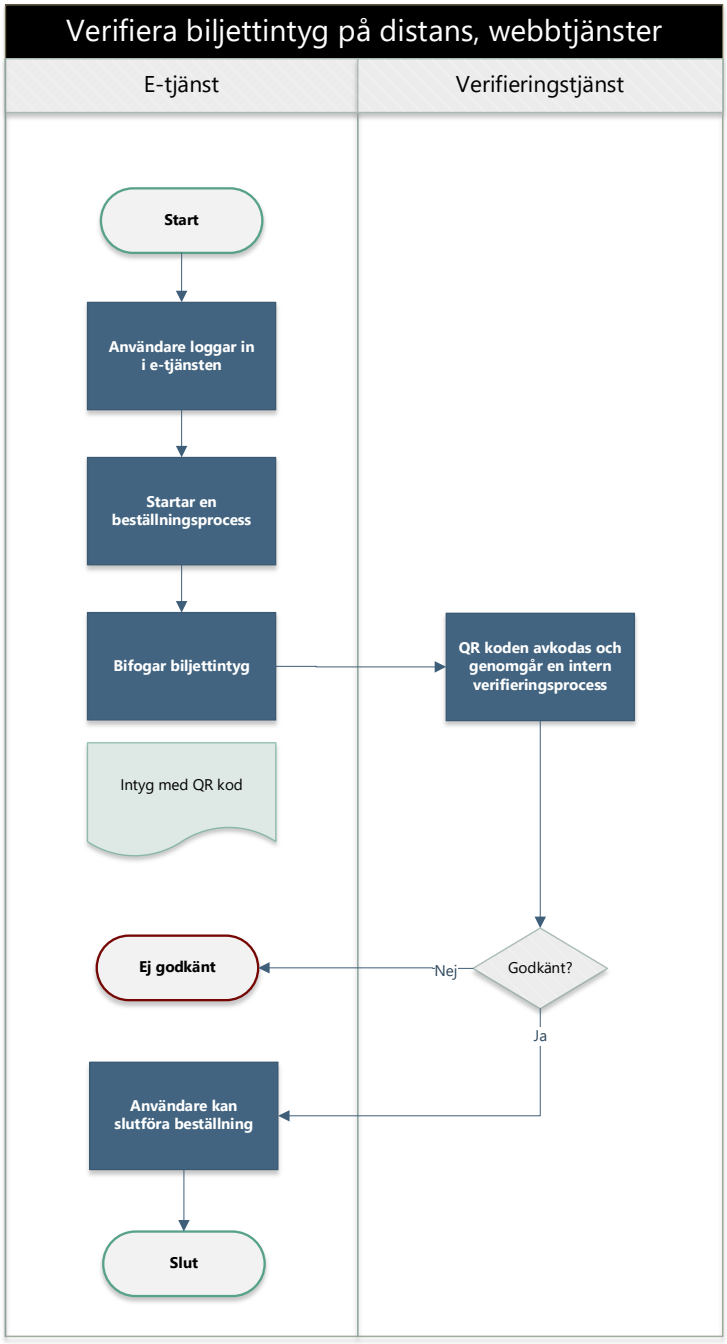
3. Appen synkar metadata från Verifierbara dataregistret (VDRen).
4. QR koden skannas med mobilkameran.
5. QR koden avkodas och genomgår en verifieringsprocess där
   a) Signaturens äkthet kontrolleras
   b) Innehållets integritet verifieras
   c) Kontroll av giltig utfärdare
   d) Giltighetstid ok
   e) Spärrlista kontrolleras (om behov finns)
   f) Nationella regler kontrolleras mot innehållet (om behov finns)
6. Negativt resultat visar inga personuppgifter, positivt resultat visar minimalt med information för att kunna verifiera individ mot ID handling. Resultatet bör av integritetsskäl döljas automatiskt efter en viss tid.
7. Verifieraren ser resultatet på skärmen och verifierar mot ID handlingen vid positivt utfall.
8. Om ID handling matchar uppgifterna i det positiva resultatet i verifieringen, så ges individen tillträde.

5.2.2.2 Verifiera biljettintyg online

Det finns ett antal olika scenarios som bör tydliggöras när man syftar till online verifiering.

Scenario 1 - Verifering på distans, webbtjänster

Verifiering på distans innebär att en innehavare kan ladda upp sitt digitala biljettintyg (pdf), som sedan avläses maskinellt och bekräftas i tjänsten. Detta kan tillämpas i olika sammanhang, såsom i en incheckningsprocess till en flygresa under pandemin med ett digitalt covidbevis eller att ansöka till universitet med ett verifierbart examensbevis. Verifieringsprocessen ser likadan ut som för offline scenariot, bortsett från att QR koden avläses maskinellt i e-tjänsten i stället för optiskt med mobilkameran.



Figur 9 - Verifering på distans, webbtjänster

Schematisk beskrivning

1. Användaren loggar in i e-tjänsten.
2. Användaren startar en beställningsprocess och begärs lägga till sitt biljettintyg.

3. Användaren lägger till biljettintyget genom att bifoga och ladda upp PDFen.
4. Verifieringstjänsten avkodar QR koden och genomgår en verifieringsprocess där:
   a) Signaturens äkthet kontrolleras
   b) Innehållets integritet verifieras
   c) Kontroll av giltig utfärdare
   d) Giltighetstid ok
   e) Spärrlista kontrolleras (om behov finns)
   f) Nationella regler kontrolleras mot innehållet (om behov finns)
5. Vid godkänt utfall kommer användaren vidare i e-tjänsten och kan fullfölja sin beställning.

Scenario 2 - Uppslag mot register

Uppslag mot register är en online-metod där en Verifierare söker information från ett online register med hjälp av en unik identifierare. Detta används när det behövs bekräftelse av en individs identitet eller behörighet, som exempelvis vid poliskontroller av körkort eller när parkeringsvakter kontrollerar fordon.
Uppslaget mot register är en okomplicerad metod för att bekräfta identitetsuppgifter hos en individ, till exempel för att verifiera ett fysiskt intyg.
Nackdelen med denna metod är att den inte alltid är tillgänglig eller att processen kan ta tid, vilket gör den mindre lämplig i situationer som kräver snabb identifiering för att effektivt kunna hantera en större mängd människor.
När man exponerar register ökar dessutom risken för obehörig spridning av personuppgifter samt att personuppgifter kan hamna i fel händer, till exempel genom dataintrång eller missbruk.

Detta är inte ett mönster som ska användas för verifiering av biljettintyg.

Scenario 3 - Verifering på distans (med en identitetsplånbok)

I likhet med mönstret för federerad identitet, såsom inloggning med mobilt bankid, så går det att i e-tjänster sätta upp en säker kommunikationskanal med en identitetsplånbok för att utbyta och verifiera presentationer (attributintyg).

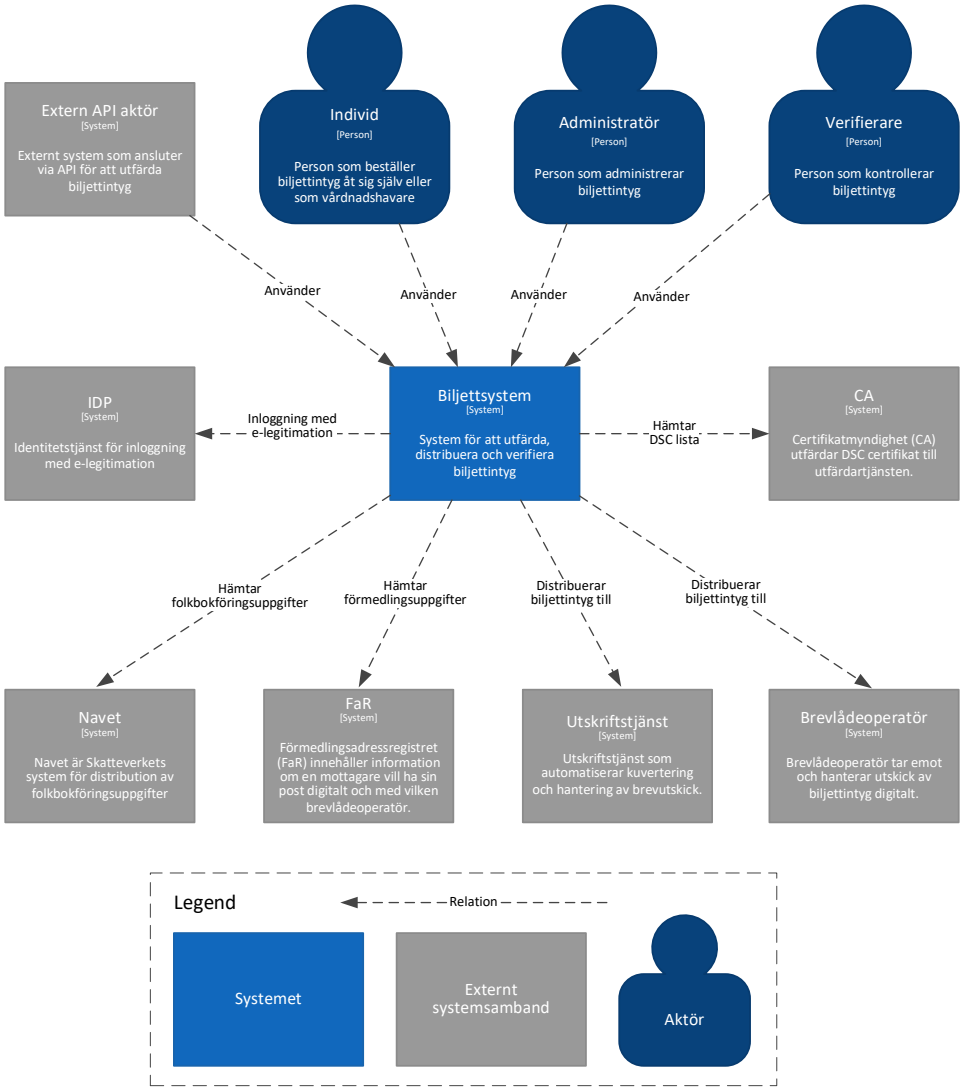
Detta är inte ett mönster som kan användas för verifiering av biljettintyg.

6 Logisk arkitektur

Den logiska arkitekturen beskriver systemet för biljettintyg och hur de olika delarna samverkar. Den beskriver de tekniska förmågor och tjänster som behövs för att stödja verksamhetens behov och förmågor som beskrivs i verksamhetsvyn. För att visualisera arkitekturen använder referensarkitekturen C4 notationens två första nivåer:

- Nivå 1 – Systemkontext vy
- Nivå 2 – Applikations vy

6.1 Systemkontext – Nivå 1



Figur 8

Figur 2 – Övergripande systemskiss med externa integrationer och PKI tjänster.

Systemkontext-nivån används för att beskriva det övergripande syftet med systemet, inklusive dess användare/aktörer, gränssnitt och andra system som det interagerar med.

I figur 2 visualiseras den systemkontext som biljettintyg verkar i inom offentlig förvaltning. Det finns ett antal användare/aktörer samt externa systemsamband, som beskrivs i efterföljande avsnitt.

6.1.1 Aktörer

Individ – Aktör med svensk e-legitimation som kan beställa biljettintyg via e-tjänsten åt sig själv eller åt andra i egenskap av vårdnadshavare.

Administratör – Aktör som på uppdrag av utfärdarmyndigheten kan skapa/hämta biljettintyg via ett admingränsnitt. Exempelvis åt dom som saknar e-legitimation.

Verifierare – Aktör som behöver hämta/uppdatera metadata från biljettsystemet för att kunna verifiera biljettintyg offline.

API aktör – Aktör som är ansluten till biljettsystemet med ett eget admingränsnitt och kan skapa biljettintyg.

6.1.2 Externa systemsamband

6.1.2.1 IDP - Identity provider

En IDP, eller Identity Provider, används för att identifiera en användare och autentisera deras identitet med hjälp av en godkänd e-legitimation. Det är idag ett vanligt sätt att identifiera en användares identitet i webapplikationer då det uppfyller svensk tillitsnivå 3 (LoA3). Det finns flera olika typer av e-legitimationer, både för privatpersoner och för organisationer/myndigheter.

6.1.2.2 Navet - Folkbokföringsuppgifter

Skatteverkets Navet är en tjänst som används för att dela folkbokföringsuppgifter mellan Skatteverket och andra myndigheter, kommuner och företag. Används även för att hämta vårdgivareförhållande för en angiven personidentitet. Det är

framför allt fullständigt namn, ålder, adressuppgifter och relationer som används för att skapa biljettintyg.

Att hämta och använda folkbokföringsuppgifter är ofta en central funktion i flera av myndighetens tjänster. För att minska beroendet till Navet använder vissa myndigheter en lokal cache som källa i stället, dvs en spegling av Navet. Detta minskar belastningen mot Skatteverkets API och ökar tillgängligheten i tjänsten för alla lokala tjänster i myndigheten.

Ett annat alternativ är att använda uppgifter som följer med inloggningen med e-legitimation. Folkbokföringsuppgifter om en individ framkommer i samband med att en invånare legitimerar sig med bank-id. Denna information skulle kunna användas istället för folkbokföringsuppgifter via Navet. Individens bank-id information uppdateras dock inte löpande utan är kopplat till folkbokföringsuppgifterna som var aktuella när certifikatet skapades. Detta innebär att IdPn inte uppdaterar förändringar som namnbyte, exempelvis till följd av giftermål. Vårdnadshavarförhållande är inte heller tillgängligt i IdP-informationen.

6.1.2.3 FAR - Förmedlingsadressregistret

Förmedlingsadressregistret (FaR) är en nationell databas som innehåller information om en individ vill ha sin post digitalt och med vilken brevlådeoperatör. Anrop måste först göras med FaR för att få reda på vilken brevlådeoperatör som intyget ska distribueras till.

6.1.2.4 Brevlådeoperatör

I skrivande stund finns det följande anslutna brevlådeoperatörer i Sverige:

- Kivra
- Min myndighetspost
- Fortnox
- Billo

6.1.2.5 Utskriftstjänst

Försäkringskassan har en utskriftscentral som automatiserar hanteringen av postutskick genom att skriva ut, kuvertera och skicka post till den adress som finns i försättsbladet. PDF inkl. försättsblad med adress paketeras i ZIP-filer och laddas upp till Försäkringskassans SFTP-yta. Det finns exempel på andra utskriftscentraler som används inom offentlig sektor, exempelvis Strålfors (Postnord).

Utskriftstjänster fungerar dock inte för:

- Personer med skyddad folkbokföring
- Personer som saknar en adress i Navet
- Personer med sekretessmarkering
- Övriga scenarios som kräver manuell hantering, t.ex. personer som är antagen i anstalt/häkte.

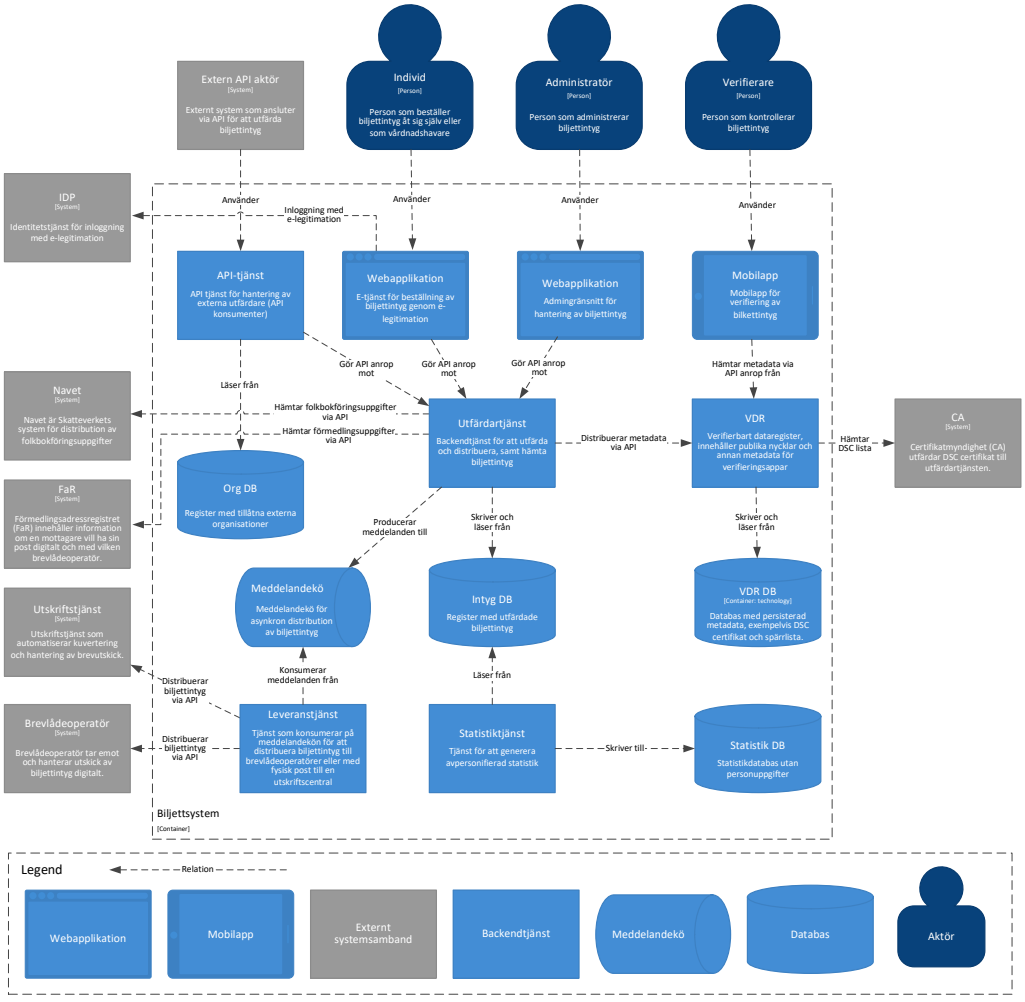
6.1.2.6 CA - Certifikatmyndighet

För att en utfärdartjänst ska kunna signera biljettintyg med något annat än självsignerade certifikat så behövs en CA tjänst som kan skapa DSC certifikat. En utfärdare kan använda en dedikerad CA för biljettintyget eller en befintlig CA som används till andra tjänster. Under Covidbevisprojektet användes en dedikerad Country Signing Certificate Authority (CSCA) med ett självsignerat root certifikat. Det fanns EU specifikationer på hur certifikaten skulle utformas med giltighetstider, utökade fält och ansvarig utgivare som gjorde att det i detta fall behövdes en dedikerad CA. Men med andra förutsättningar och krav kan befintliga DSC-certifikat användas eller så kan en utfärdare till och med välja att anskaffa dessa från en kommersiell certifikatmyndighet.

CA tjänsten som refereras till i avsnitt 7 har ett admingränssnitt som används för att skapa eller återkalla DSC certifikat för utfärdandetjänsten. Det är rekommenderat att man skyddar CA tjänstens privata nycklar som används för att skapa certifikat i en HSM. För att distribuera dessa certifikat till Verifierare används en distribueringsnod för metadata i systemet som kallas för Verifierbart Dataregister (VDR).

6.2 Applikationsvy – Nivå 2

Syftet med applikationsvyn är att visa den logiska grupperingen av komponenter i systemet, såsom webapplikationer, tjänster och databaser. Applikationsvyn är ett sätt att visualisera arkitekturen för biljettsystemet och att förstå hur de olika komponenterna interagerar med varandra. Den kan också användas för att identifiera potentiella arkitektoniska problem, såsom tät koppling mellan komponenter och onödiga beroenden.



Figur 9 – Applikationsvy för biljettsystem

6.2.1 Webapplikationer

Webapplikationerna innehåller e-tjänster för att skapa, hämta och distribuera intyg till individ. I applikationsvyn finns det två webapplikationer med olika

ändamål och användare. Den första webapplikationen innehåller en e-tjänst för självprovisionering genom inloggning med e-legitimation. Den används av individen för att beställa åt sig själv eller som vårdnadshavare åt andra. Den andra webapplikationen är ett admingränsnitt för handläggare inom utfärdarmyndigheten, som möjliggör administrering och hantering av biljettintyg åt andra. Webapplikationerna kan anslutas till en IDP genom en intern service provider (SP) för inloggning med e-legitimation och har i fallet med Covidbevis innehållit en Single Page Application (SPA) som laddats ner till användarens browser efter lyckad inloggning. Backend for frontend (BFF) mönstret kan användas för att separera ansvar från SPA applikationen och skydda känsliga data som behövs för exempelvis auktorisation med utfärdartjänstens APIer.

6.2.1.1 Webapplikation för självprovisionering

Webapplikationen innehåller en e-tjänst för att beställa biljettintyg efter godkänd inloggning med e-legitimation. Kan även användas för att hämta sina giltiga biljettintyg, om lösningen konstrueras för att arkivera dessa i databasen. Konfigureras med en service provider som ansluter till en identitetstjänst (IDP) för hantering av flera olika e-legitimationer som uppfyller svensk tillitsnivå 3 för e-legitimering. Tjänsten bör vara publikt åtkomlig från internet.

6.2.1.2 Webapplikation för att administrera intyg

För de individer som saknar e-legitimation eller har någon form av kognitiv funktionsnedsättning behöver utfärdarmyndigheten ett admingränsnitt för handläggare att hantera supportärenden. Det kan även uppstå situationer där grunddata är felaktig, där en utfärdare behöver möjligheten att mata in intygsuppgifter manuellt som ett undantagsfall. Dessa användningsfall måste kunna hanteras av en utfärdare så att enskilda individer inte hamnar i utanförskap. Administrering av biljettintyg startas genom en verksamhetsprocess, som samlar in underlag från individen genom exempelvis ett blankettförfarande (se verksamhetsvy avsnitt 5.1.2.1).
Inloggning och behörighetsstyrning rekommenderas att göras med en identitetstjänst som uppfyller ett minimum av svensk tillitsnivå 3 för e-legitimering, exempelvis EFOS eller SITHS.

6.2.2 Utfärdartjänst

Central backendtjänst för att skapa och distribuera, samt hämta biljettintyg. Utfärdartjänsten kan även användas för att återkalla ett biljettintyg. Utfärdartjänsten är tillståndslös och ska innehålla ett eller flera APIr med metoder för att:

- Skapa och distribuera ett biljettintyg
- Lista biljettintyg från databasen
- Hämta ett biljettintyg från databasen
- Revokera ett biljettintyg

I följande avsnitt redogörs för de funktioner som behöver finnas i utfärdartjänsten, samt hur den samverkar med andra logiska komponenter i systemet. Referensprojekt för en utfärdartjänst finns länkat i avsnitt 8.4.

6.2.2.1 Skapa 2D kod

Tjänsten ska skapa en 2D kod som innehåller en digitalt signerad token (CWT) biljett. Det elektroniska signaturschemat för att skapa COSE signaturen behöver ett giltigt DSC certifikat och den kryptografiska privata nyckeln för att signera en hashsumma (sha-256) av CWT biljettens innehåll. Det är starkt rekommenderat att hantera nyckeln i en HSM. Flödet för att skapa 2D koden beskrivs i avsnittet runt Styrande principer (kapitel 3.5).

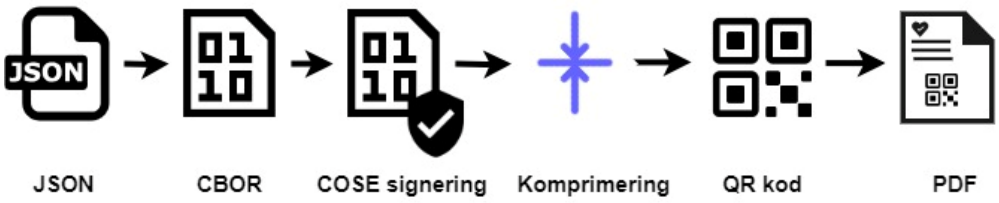
6.2.2.2 CI/CTI beräkning

Det behöver skapas en unik identifierare i CWTn så det går att identifiera ett biljettintyg. Detta behövs exempelvis för att kunna återkalla ett intyg eller för spårbarhet och felsökning. Identifieraren rekommenderas att inte baseras på personidentifierare, eftersom dom behöver användas vid återkallande av intyg. I Hcert specifikationens JSON schema kallas detta för Certificate id (ci) och genereras enligt följande:

CI:{version}:{country}:{random}:{checksum}

6.2.2.3 PDF mall

Det behövs en design på hur intyget ska se ut utöver QR koden. I Covidbevisprojektet har man i Sverige använt verktyget Adobe Indesign för att skapa PDF mallar som sedan använts i utfärdandetjänsten tillsammans med Apache PDFBox biblioteket för att generera upp texter och QR kod i platshållare. En fördel med denna hantering är att PDF mallen och dess design kan tas fram och hanteras separat i verksamheten.



Figur 10 - Flöde för PDF generering

6.2.2.4 Databas

Databas för arkivering/lagring av skapade intyg. Arkivering ska ske enligt gällande rätt och måste konsulteras och beslutas i varje enskilt fall genom en juridisk analys. Det är rimligt att biljettintygets aggregerade JSON dokument och den genererade PDFen, samt identiteten på innehavaren och skaparen sparas för att möjliggöra felsökning och spårbarhet vid behov.

Databasen kan även användas för köhantering vid distribution/leverans av intyg.

6.2.2.5 Distribuering

Biljettintyg kan distribueras till individen på flera olika sätt och utfärdartjänsten kan möjliggöra denna leverans på olika sätt. Det rekommenderas att alltid utgå från en ”digitalt först” strategi.

Direkt

Detta syftar till bevis som levereras direkt i svaret till anropande tjänst (gränssnitt eller anropande systemet. Beviset levereras som PDF, men i vissa fall även med bild och innehåll för att kunna sättas samman till något annat presentationsformat. E-tjänsten kan även erbjuda möjlighet att transformera biljettintyget för nedladdning och hantering i en ”Wallet”.

Digital brevlåda

Digital leverans sker genom de brevlådeoperatörer som finns på den svenska marknaden och där FaR används för att först avgöra huruvida en individ har digital brevlåda. Kravställningen för Covidbevis byggde på att distribution alltid skulle ske mot ansluten digital brevlåda när en sådan fanns. Samt att presentera för användaren i e-tjänsten vilken brevlåda beviset skickats till, förutsatt att en digital brevlåda finns att tillgå.

Dessa krav gjorde att FaR behövde vara en del i det synkrona flödet för utfärdandet av ett Covidbevis, vilket resulterade i en risk att det inte skulle gå att skapa ett Covidbevis om förmedlingsadressregistret blev otillgängligt. Ett alternativ till detta skulle vara att separera ansvar och göra ett uppslag mot FaR redan efter inloggning i webapplikationen. Då kan utskicket göras valbart, eftersom användaren istället kan ladda ner intyget i PDF.

Fysiskt brev

Med fysisk distribution avses bevis som skickas via vanlig post/brev. Detta kan vara till individens folkbokföringsadress eller annan angiven adress i Sverige eller utlandet. Utskicken kan ske på olika sätt beroende på förutsättningarna runt individen och dess adress. Men att ansluta biljettsystemet till en utskriftscentral för att automatisera flödet och minska belastningen på utfärdarens administratörer bör ses som en föredragen lösning om förutsättning finns.

6.2.2.6 Meddelandekö

För att leveransen av ett biljettintyg via digital eller fysisk post inte skall påverka själva skapandet av biljettintyget, förbereds och placeras ett skapat biljettintyg på en meddelandekö för att senare skickas med en dedikerad leveranstjänst. Denna

design följer ”separation of concerns” principen i syfte att fördela ansvar och gör systemet mer robust då leveransen kan hanteras asynkront.

För Covidbevis användes intygsdatabasen som meddelandekö, med en status flagga som indikerade om meddelandet skickats. Att denna lösning för kö valdes beror på tidspress under utvecklingen och där det snabba behovet av en asynkron förmåga att sända digitalpost var såpass akut att ändra.

6.2.2.7 Återkalla biljettintyg

En återkallelse av ett biljettintyg innehåller både administrativa och tekniska åtgärder. Den tekniska tjänsten ska (om intyget är arkiverat) göra följande:

1. Flagga intyget som återkallat i databasen.
2. Förbereda revokeringsdata för spärrlista i VDRen.
3. Uppdateras VDR via API.
4. Svar tillbaka till anropande system (admingränsnitt).

För covidbevis togs en återkallningsmekanism baserat på hashlistor och blomfilter fram utifrån krav på att inte sprida personidentifierare (direkta eller indirekta). En del länder använde olika typer av personidentifierare i uppbyggnaden av bevisets unika ID, vilket gjorde att även ”ci” identifieraren betraktades som en personidentifierare och kunde därför inte användas i spärrlistan utan att ”avpersonifieras”.

Lösningen som togs fram av EU hade också för avsikt att hålla nere på storleken på spärrlistan som måste laddas ner i mobiltelefoner för offline verifiering. Konceptet för denna lösning presenteras här: ehealth-eudcc-revocation_en.pdf (europa.eu).

6.2.2.8 Throttling mönster (begränsa anrop)

För att hantera begränsningar i antal samtidiga anrop mot externa källor kan throttling mönstret tillämpas genom en asynkron mekanism med trådhantering, även i tillståndslösa tjänster. Varje integrationspunkt har en egen trådpool som begränsas utifrån antal tillåtna aktiva trådar och anrop som får ligga i kö. När dessa parametrar överskrids tillåts inga fler anrop till underliggande tjänst och

utfärdartjänsten returnerar HTTP-status 429 (Too many requests) till anropande system.

6.2.3 Leveranstjänst

Leveranstjänsten konsumerar på meddelandekön för att hantera och distribuera biljettintyg till brevlådeoperatörer eller med fysisk post till en utskriftscentral. Genom att separera ansvar från utfärdandetjänsten och använda en dedikerad leveranstjänst blir både utfärdande och leveransflödet mer robust.
Leveranstjänsten behöver kunna hantera omsändningar, både automatiskt enligt ett bestämt mönster och manuellt vid behov i händelse av tillgänglighetsproblem i utskriftstjänst eller brevlådeoperatörers tjänster.

6.2.4 Statistiktjänst

Det finns vanligtvis ett stort verksamhetsbehov av statistik och uppföljning. Tjänstens syfte är att tillhandahålla en avpersonifierad databas som kan användas av verksamhetens BI system för att följa upp och presentera statistik, utan risk för gallring och helt separat från det aktiva flödet. Statistiktjänsten kan schemaläggas till att ansluta sig till intygsdatabasen på daglig basis och då:

1. Hämta data som inte är markerade som överförda.
2. Transformera enligt specifikation på exempelvis demografisk och geografisk information.
3. Persistera i statistikdatabasen.
4. Markera i intygsdatabasen att intygets data har överförts.

6.2.5 Extern API-tjänst

Om en extern applikation behöver ansluta till utfärdartjänsten så är det rekommenderat att använda en dedikerad API tjänst för att upprätthålla rätt nivå av säkerhet, som öppnas upp mot aktörens ipadress och säkras ytterligare genom

användande av MTLS (dubbelriktad TLS). Detta mönster användes för att ansluta vårdaktörer under covidpandemin. En separat databas kan användas som ett extra lager av säkerhet för att verifiera att den externa aktörens organisationsnummer i TLS certifikatet är godkänd. Det möjliggör också att man kan ”stänga av” en aktör.

6.2.6 VDR – Verifierbart Dataregister

En VDR är en distribueringsnod för publika nycklar (lista på DSC certifikat), men kan även användas för att distribuera spärrlistor, regler och annan metadata till Verifierare. Begreppet ”Verifierbart dataregister” kommer från W3C Verfiable Credentials standarden. VDRen synkar listan med DSC certifikat direkt från CA tjänsten.

Metadatat laddas ner till Verifierarens verieringsapp/tjänst vilket möjliggör säker verifering offline. CA tjänstens spärrlista (CRL) behöver inte användas för att verifiera DSC certifikatets status. I stället kan den primära giltighetsmekanismen (i likhet med Covidbevis) vara närvaron av DSC certifikatet på VDRen. För Covidbevis var APIet som verifieringsapparna kunde göra API anrop mot öppet från internet.

6.2.7 Mobilapp

För att avkoda 2D koden och verifiera signaturen i biljettintyget krävs det en verifieringstjänst. Avsnittet beskriver generella principer, valideringsflödet och tekniska beslut som behöver tas vid utveckling av en verifieringsapp.

6.2.7.1 Generella principer och krav

Användarvänlighet - Det ska vara lätt för en Verifierare att använda och förstå verifieringsappen.

Effektivitet och snabbhet - Det ska gå snabbt att verifiera en Innehavares intyg. Vid dålig belysning kan det vara svårt att fokusera och avläsa en QR kod på papper. Det ska ses som ett krav att telefonens lampa kan användas under skanning.

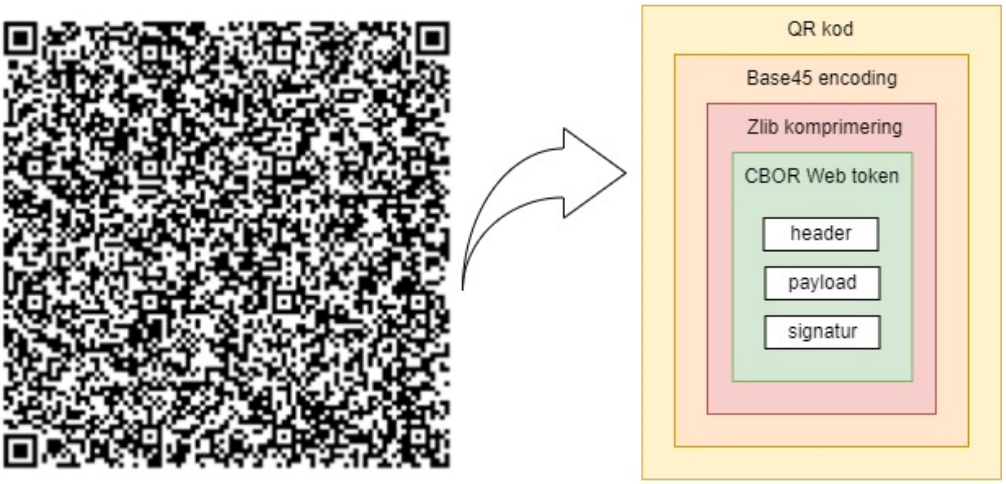
Tillförlitlighet - En Verifierare ska kunna lita på att resultatet i verifieringstjänsten följer gällande regelverk. Om dynamiska regler används måste dessa synkas av verifieringsappen. En veriferingsapp bör automatiskt försöka hämta och uppdatera metadata från när den används, men ska fortsatt fungera när kontakten ligger nere.

Integritet – Hantering av personuppgifter sker i enlighet med gällande rätt och GDPR. Det innebär exempelvis att verifieringsappen måste beskriva hur personuppgifterna används, om dom sparas eller skickas vidare för behandling.

Uppgiftsminimering – Minimera personuppgifter som presenteras i appen, det minskar risken för att personuppgifter hamnar hos obehöriga.

6.2.7.2 Flöde för att verifiera en QR kod

Beskrivning av valideringsflödet för att avkoda och validera en QR kod som innehåller ett biljettintyg.



Figur 11 Avkodning av QR kod baserad på Hcert specifikationen.

1. Avkoda QR koden - Avläs QR-koden för att få QR-alfanumeriska koden. Hcert använder sig av ett prefix (HC1), isåfall börjar den med "HC1:6BF...".

2. Base45 avkodning - Base45-avkoda delen efter "HC1:" för att få det ZLIB-komprimerade innehållet.

3. Packa upp det ZLIB komprimerade innehållet för att få ett CBOR Object Signing and Encryption (COSE) taggat meddelande (CWT).

4. CBOR-avkoda det taggade CWT-meddelandet
   - skyddad header - cbor-kodat
   - payload - cbor-kodat
   - digital signatur (signerad sha256-digest)

5. CBOR-avkoda header för att få signeringsalgoritmen och ”kid” (om det används).

6. CBOR-avkoda payload för att få payload claims (iss, iat, exp och intygets payload).

7. Intern verifieringsprocess
   a) Signaturens äkthet kontrolleras
   b) Innehållets integritet verifieras
   c) Kontroll utfärdad av en giltig utfärdare
   d) Giltighetstid ok
   e) Spärrlista kontrolleras (valbart)
   f) Nationella regler kontrolleras mot innehållet (valbart)

6.2.7.3 Tekniska ramverk

Det finns ett antal tekniska ramverk som bör utvärderas inför utveckling av en verifieringsapp till mobiltelefoner.

Native app

En native app är en app som är utvecklad specifikt för en viss plattform, till exempel Android eller iOS. Native appar är vanligtvis de mest prestandamässigt effektiva och har tillgång till alla funktioner som är specifika för den plattform de är utvecklade för, exempelvis kameran. Nackdelen är att utveckling och tester måste göras på två olika plattformar och programmeringsspråk.

Crossplatform app

En crossplattform app är en app som är utvecklad för att köras på flera plattformar, vanligtvis Android och iOS. Crossplattform appar är ofta utvecklade med hjälp av ett ramverk som gör det möjligt att använda samma kodbas för att skapa appar för olika plattformar. Crossplattform appar är vanligtvis inte lika prestandamässigt effektiva som native appar, men de är oftast enklare och billigare att utveckla. Nackdelen är att när kameran behövs så måste det ändå göras en del plattformsspecifik utveckling. Exempel på ramverk är Xamarin/MAUI, Flutter, React Native.

PWA - Progressive webapp

En PWA, eller Progressive Web App, är en webbapp som har utvecklats för att

fungera som en native mobilapp. En PWA kan installeras på en användares hemskärm och kan användas offline, genom att ladda ner innehåll i förväg. En PWA är vanligtvis snabbare och mer effektiv än vanliga webbappar, men de har inte samma tillgång till funktioner som är specifika för den plattform de är utvecklade för. I skrivande stund så får en PWA inte tillgång till kameran på IOS mobiler, men funkar på Android telefoner.

6.2.7.4 Distribution via appbutiker

Om veriferinsappen ska installeras och användas i vanliga mobiltelefoner, så är en rekommendation att dom distribueras via appbutikerna för respektive plattform. Man kan välja att publicera appar för enstaka personer, en organisation eller publikt för alla. Förutsättningen är att man registrerat sin organisation och skrivit på utvecklaravtal med appbutiken.

Krav som kommer att granskas

Kvalitet och innehåll: Appbutikerna har riktlinjer för appkvalitet och innehåll som måste följas. Dessa riktlinjer kan inkludera krav på att appen inte bryter mot upphovsrätt, inte innehåller skadlig kod, inte innehåller olämpligt innehåll och så vidare.
Tekniska krav: Appen måste uppfylla tekniska krav och vara kompatibel med plattformen och operativsystemet som appbutiken betjänar. Det innebär att appen måste fungera korrekt och ha en användarvänlig upplevelse.
Säkerhet och integritet: Appen måste vara säker och skydda användarnas data och integritet. Appbutiker tar säkerhet och dataskydd på stort allvar. Det behöver finnas en integritetspolicy länkad i appen.
App-ikoner, beskrivning och bilder: Det måste finnas en tilltalande app-ikon, skriva en beskrivning och tillhandahålla skärmbilder som visar appens funktionalitet och utseende.
Testning och godkännande: Innan du kan publicera din app måste den gå igenom en godkännandeprocess som appbutiken utför. Under denna process granskas appen för att säkerställa att den uppfyller alla krav och riktlinjer. Det kan ta några dagar eller veckor beroende på appbutiken.

Uppdateringar och underhåll: Efter att din app har publicerats måste du fortsätta att uppdatera och underhålla den för att hålla den säker och fungerande. Appbutiken kan avlägsna appar som inte uppfyller kraven eller som inte underhålls.

6.2.7.5 Summering

Om det behövs en app med hög prestanda och bästa tillgång till mobilens hårdvara såsom kameran är en native app det bästa valet. Under Covidbevisprojektet märktes stor skillnad i hur mycket snabbare och enklare QR koder avlästes med nativeappar.

I EU var det flera länder som tog fram nativeappar för verifering av Covidbevis med stöd för revokering, certlogic business rules. Dessa kan vid behov användas som referenskod. Länkar finns på eHealth Networks Github sida som är länkat i avsnitt 7.

Om det behövs en verifieringsapp som är enkel och billig att utveckla är en crossplattform app ett bra alternativ. I Sverige togs en Xamarin app fram som referensapp för verifiering av Covidbevis, länk finns i avsnitt 7.

Om det helt saknas mobilutveckling inom verksamheten kan en PWA vara ett alternativ. Fördelen med en PWA är även att du slipper distribuera den via appbutikerna och det blir enklare att förvalta ihop med de andra tekniska tjänsterna om samtliga tjänster kan tas fram av samma team. Kompromissen blir att verifieringsappen inte kommer att fungera på IOS telefoner. Samt att känsligheten i QR avläsning inte är lika bra som på en nativeapp.

7 Referenser

7.1 Om avsnittet

Svenska Covidbevislösningen togs fram genom ett samarbete mellan flera svenska myndigheter. Myndigheten för Digital förvaltning (DIGG) stod initialt för utveckling och förvaltning av utfärdandetjänsten och PKI tjänsterna, medan eHälsomyndigheten tog fram de olika webapplikationerna (frontend) och stod för de rättsliga besluten som formade kraven på tjänsterna. Dokumentation och referenskod som referensarkitekturen baseras på kommer framför allt ifrån den dokumentation som finns tillgänglig på eHealth networks Github sida, eHälsomyndighetens förvaltningsdokumentation för Covidbevis samt de referensprojekt som finns på DIGGSweden.

7.2 Avgränsningar

DIGG har som policy att tillgängliggöra kod som öppen källkod och är de kodprojekt som kommer att refereras i detta avsnitt. Det är också dessa projekt som innehåller de tekniska komponenter som behövs för att skapa och verifiera ett biljettintyg. Projekt för frontend dvs grafiska gränssnitt med unik grafisk profil, javascript ramverk och komponentbibliotek bedöms inte vara relevant att dela och rekommendera i denna referensarkitektur. De som är framtagna för eHälsomyndigheten är idag inte heller tillgängliga som öppen källkod.

7.3 Referenskod

Referensprojekten som refereras till i detta dokument har utvecklats och förvaltats av DIGG i samarbete med eHälsomyndigheten under Covidbevisprojektet.

7.3.1 Teknikstack

Summering av de mest signifikanta tekniska valen av teknik, ramverk och programvaror för utveckling av tjänsterna runt Covidbevis.

Teknik	Typ	Motivation
Java 17	Programmeringsspråk	Ett etablerat programmeringsspråk där flera svenska myndigheter har bred kompetens. Väl anpassat för kritiska system.
Spring Boot 2.x.x	Ramverk	Spring Boot är ett ramverk som är moget, har många användare (lätt att hitta resurser med rätt kompetens) samt oöverträffad dokumentation och är ett bra val för Java-lösningar.
Maven 3.x	Byggverktyg	Etablerat verktyg där flera myndigheter har bred kompetens.
PostgreSQL13.x	Databas	Högpresterande databas där driftleverantör har bra kompetens. Används inom flera kritiska system internationellt.

7.3.2 Länkar

dgc-java - Java bibliotek för att skapa, signera och generera en QR kod med en CWT token.

utfärdartjänst - Källkod för svenska covidbevisets utfärdartjänst (använder dgc-java), Test-API, senderapp för att hantera utskick till digitala brevlådor.

CSCA - Källkod Country Signing Certificate Authority (CSCA)

trustpoint - Källkod Sveriges nationella Trust Point för distribuering av publika nycklar/DSC certifikat och annan metadata.

verifieringsapp – Referensapp för verifiering av covidbevis

7.4 EHealth network

Dokumentation och specifikationer för EU DCC (Covidbeviset), JSON schema och valuesets finns på eHealth Networks Github. Innehåller även implementationer och projekt i andra programmeringsspråk, exempelvis .NET och Python.

eu-dcc-overview Bra ställe att få en överblick av allt som skapats runt Covidbevis, samt hur man kommer igång med att skapa en egen implementation.

eu-dcc-hcert-spec Specificerar kodningsformat som används av DCC. Denna specifikation utgör grunden för styrande principer i kapitel 3.

eu-dcc-schema JSON schemaspecifikationen för den data som lagras i en DCC.

eu-dcc-valuesets Specificerar värdemängderna och ger exempel på värdemängderna (JSON dokument).

eu-dcc-business-rules Ramverket för affärsregler (Certlogic) som användes av vissa veriferingsappar.