SELF-SOVEREIGN IDENTITY

DigdirCamp 2021

Juni Bugge
Martin Immanuel Agnalt Burgos
Ingunn Langtangen Furuberg
Gunvor Huso
Vilde Taklo Kenworthy
Eirik Schøien
Jonas Hjelle van Weert
Inger Helen Yri

6. august 2021

Digdir

Innhold

1 Introduksjon

1.1 Bakgrunn for SSI

Self-sovereign identity (SSI) eller egen-kontrollert identitet er et tankesett der brukeren skal ha full kontroll på egen identitet. Dette tankesettet er en motreaksjon til informasjonssamlingen de store plattformene, som Google og Facebook, har gjennomført de siste årene (Forbrukerrådet, 2021). For at en identitet skal være egen-kontrollert er det viktig at brukeren skal eie sine data og ikke trenger å stole på en sentral entitet for å bevise en påstand om seg selv. I tillegg er det viktig at brukeren har full kontroll på den personlige informasjonen som brukeren deler og hvem som har tilgang til den. Til slutt må identiteten kunne verifiseres på tvers av plattformer og lokasjoner ved å ta i bruk samme standarder (Tykn, 2021).

Den 3.juni 2021 la EU-kommisjonen frem et forslag om en europeisk digital identitet som skal være tilgjengelig for alle innbyggere i EU. Gjennom en europeisk digital identitetslommebok skal innbyggerne ha mulighet til å bevise sin identitet og dele elektroniske dokument (EU, Commission proposes a trusted and secure Digital Identity for all Europeans, 2021). Hvert medlemsland i EU skal tilby en lommebok til sine innbyggere på et nasjonalt nivå. Den europeiske digitale identiteten skal være gyldig på tvers av landegrenser innad i EU. Målet er at en verktøykasse skal være klar innen september 2022, og ett år etter forskriften er lovbestemt skal lommebøkene være implementert og mulig å ta i bruk (EU, Questions and Answers, 2021).

I denne rapporten vil vi derfor presentere våre funn i forbindelse med vår Proof of Concept. Videre utforsker vi SSI-økosystemet, der vi ser på alternativer til implementasjon av SSI. Før vi til slutt ser på rollen Digdir kan ha i dette økosystemet.

1.2 SSI-økosystem

Et SSI-økosystem består hovedsakelig av fire aktører: utstedere av bevis, brukere, tjenester som mottar bevis og et tillitsrammeverk (Duffy, 2020).

En utsteder (issuer) oppretter bevis tilknyttet en identitet og sender denne til brukerens digitale lommebok. I utgangspunktet kan alle utstede et bevis. For at beviset imidlertid skal ha noe verdi må utstederen være anerkjent og ha myndighet til å utstede de typene bevis den tilbyr. Bevis inneholder en signatur fra utsteder, slik at andre i økosystemet kan verifisere utstederen og legitimiteten til beviset.

En bruker (holder) kontrollerer sin identitet gjennom en form for digital lommebok, der man oppbevarer opplysninger i form av bevis, og styrer hvem man ønsker at denne informasjonen deles med. Det er brukeren selv som skal dele eller gi tillatelse til at opplysninger deles, og brukeren skal kunne trekke de tilbake.

En bruker kan være et individ, men også en organisasjon, instans eller lignende. De har til felles at de er holdere av en analog eller digital identitet og opplysninger tilknyttet denne, som de har enerett over og bestemmer hvordan skal brukes og behandles. Dette kan sees i sammenheng med EUs personvernforordning General Data Protection Regulation (GDPR), vedtatt i 2016, som ga individer innenfor EU og EØS-området utstrakte rettigheter over sine personopplysninger og data (Gisle, 2018).

En tjeneste (verifier) er de som mottar og verifiserer bevis fra brukere. Opplysninger en tjeneste ønsker å motta kan være navn, alder, personnummer og annen personlig informasjon, om de har en viss type tillatelse / sertifikat eller lignende. Det er ønskelig at tjenesten kun skal kreve de opplysningene den strengt tatt behøver. Ønsker tjenesten for eksempel å vite at brukeren er over 18 år, skal den kun forespørre et bevis på at brukeren oppfyller dette, uten at brukeren må dele sin faktiske alder.

Tillitsrammeverket (verifiable data registry) er et oppslagsverk der informasjon for å kommunisere og verifisere bevis er tilgjengelig. Formålet er at utsteder, bruker og tjeneste skal kunne ha tillit til hverandre i økosystemet. Det kan inneholde identifikatorer og nøkler, samt standarder for dem. Det kan også inneholde informasjon om hvilke typer bevis de ulike utstederne tilbyr og standarder for bevis.

Gjennom tillitsrammeverket vil man ved bruk av kryptografiske metoder sikre at:

1. Utsteder og bevis er legitime
2. Utsteder er tilknyttet riktig identifikator
3. Brukeren som ønsker å verifisere seg mot tjenesten er den de påstår å være
4. Ha muligheten til å sjekke om tjenesten er legitim

Det finnes ulike løsninger for hvordan et SSI-økosystem kan se ut. Dette dreier seg i stor grad om hvordan tillitsrammeverket realiseres, og hvor desentralisert det skal være. Tradisjonelt har man vært avhengig av sentraliserte autoriteter, der parter stoler på hverandre fordi de begge stoler på denne aktøren. En løsning der slike sentraliserte autoriteter styrer hva som er tilgjengelig i tillitsrammeverket kalles «permissioned». En ulempe med en slik løsning er at man har fått sin identitet fra noen andre, derav «tillatt», og er avhengig av at denne ikke blir trukket tilbake.

Ved å bruke blant annet ny blokkjedeteknologi er det mulig å få et såkalt «permissionless», desentralisert økosystem uten noen sentraliserte autoriteter. Oppslagsverket vil være distribuert på blokkjeden, den vil også sørge for at entitetene kan ha tillit til hverandre.

SSI-tankesettet bygger på en rekke prinsipp eller retningslinjer som må følges for å oppnå et fullverdig og velfungerende SSI-økosystem. Christopher Allen, medforfatter av TLS-standarden for sikker kommunikasjon og pioner innenfor SSI-bevegelsen, definerer dem slik:

1. **Existence.** *Users must have an independent existence.*
2. **Control.** *Users must control their identities.*
3. **Access.** *Users must have access to their own data.*
4. **Transparency**. *Systems and algorithms must be transparent.*
5. **Persistence.** *Identities must be long-lived.*
6. **Portability.** *Information and services about identity must be transportable.*
7. **Interoperability.** *Identities should be as widely usable as possible.*
8. **Consent.** *Users must agree to the use of their identity.*
9. **Minimalization.** *Disclosure of claims must be minimized.*
10. **Protection.** *The rights of users must be protected.*

(Allen, 2016)

Det følger av SSI-tankesettet at det ikke bare vil være én tilbyder av digitale lommebøker, men en rekke alternative digitale lommebøker, på samme måte som det finnes flere banker. Disse må kunne brukes om hverandre, og støtte de ulike utstederne og tjenestene. Det vil derfor være behov for standardisering av bevis og identifikatorer. Identifikatorer er adresser entitetene bruker for å kommunisere med hverandre. Standardene som oppgis i DID-spesifikasjonen (W3C, Decentralized Identifiers (DIDs) v1.0, 2021) og VC Data Model av W3C (W3C, Verifiable Credentials Data Model - 3.2 Credentails, 2021) er aktuelle for å oppnå dette.

2 Begreper

2.1 Verifiable Credential

En Verifiable Credential (VC) er et verifiserbart legitimasjonsbevis med informasjon relatert til en type legitimasjon, hvem beviset tilhører, hvem som har utstedt beviset og hvor lenge beviset er gyldig. En VC kan være legitimasjon som erstatter fysiske legitimasjonsbevis som førerkort, vitnemål og aldersbevis, eller ikke-fysiske bevis som eierskap i et selskap. VC'en skal inneholde samme spesifikasjoner som fysiske legitimasjonsbevis, slik som for eksempel en førerkortklasse (W3C, Verifiable Credentials Data Model - 3.2 Credentails, 2021). En person kan ikke få en VC om hen ikke oppfyller kravene for denne VC'en.

VC kan være alle former for informasjon om et subjekt, vi ser for oss at informasjonen alltid delt opp i enkeltpåstander. Eksempler er en VC for fødselsnavn, og en VC for folkeregistrert adresse. Informasjon som vanligvis pleier å bli hentet sammen fra folkeregistret.

2.2 Verifiable Presentation

En Verifiable Presentation (VP) er en presentasjon av en utvalgt mengde VC'er. En VP er dermed satt sammen av en eller flere VC'er, og utvalget er spesialtilpasset hva en tjeneste ønsker å se legitimasjonsbevis for (W3C, Verifiable Credentials Data Model - 3.3 Presentations, 2021). Et eksempel på en VP kan være en samling av VC'ene for gyldig sykepleierutdanning, politiattest og at personen er over 18 år.

En VP er satt opp på en måte slik at utsteder av bevis og eier av bevis kan stoles på etter en prosess med kryptografisk verifisering. Noen typer VP kan inneholde data som er fremstilt fra, men som ikke inneholder, de originale VC'ene, for eksempel gjennom nullkunnskapsbevis.

En VC og VP kan produseres av hvem som helst. Derfor benyttes alltid signatur av utsteder for å sikre eierskap til hvem som produserte beviset.

2.3 Digital signatur

Digital signering er en metode som sikrer integritet og autentisering. Signeringen er kun gyldig så lenge meldingen forblir uendret, ettersom signaturen baserer seg på tilhørende melding. Autentiseringen er sikret ettersom det benyttes private og offentlige nøkkelpar (CISA, 2021).

Kort hvordan digital signatur fungerer

Meldingen som skal bli signert blir først hashet og deretter kryptert med signererens private nøkkel. Den krypterte hashverdien kalles digital signatur. Dersom meldingen endres, vil det føre til en helt annen hashverdi. Derfor er det lett å kontrollere at meldingen er lik den opprinnelige, eller ikke. Man verifiserer en digital signatur ved å dekryptere signaturen med

den offentlige nøkkelen til signereren. Da får man hashverdien. Dersom denne stemmer overens med den originale hashverdien til meldingen, er signaturen gyldig og meldingen uendret.

2.4 ZKP – Nullkunnskapsbevis

Nullkunnskapsbevis (Zero knowledge proofs) er en metode for å bevise at informasjon er sann uten å avsløre selve informasjonen.

"Zero knowledge proof (ZKP) allows a powerful prover to convince a weak verifier that a statement is true, without leaking any extra information about the statement beyond its validity" (Zhang, Xie, Zhang, & Song, 2020). Denne definisjonen forklarer essensen i det som egentlig er et veldig abstrakt begrep med røtter fra matematikken. I matematikken har ZKP vært et begrep siden 1985 (Wikipedia, 2021). I kryptografi og i den digitale verden derimot har ZKP blitt et aktuelt tema først etter 2020 (MATTR, 2021), da nye fremskritt innen teknologi har modnet til et punkt der bruken av ZKP er sett på som mulig. Årsaken til denne muligheten er at genereringen av ZKP med nye metoder ikke tar flere minutter, men kanskje bare et sekund for små ZKP bevis, og flere sekunder for store ZKP bevis (Zhang, Xie, Zhang, & Song, 2020). I forbindelse med et stort SSI økosystem, der tillitt til alle aktører ikke alltid er like god, kan ZKP være ikke bare en gode, men en nødvendighet.

ZKP baserer seg på tre ting:

- Kompletthet: Det er mulig å overbevise verifikator, gitt et utsagn.
- Soundness: En ondsinnet bruker kan ikke overbevise en verifikator.
- Null-kunnskap: Nullkunnskapsbeviset avslører ikke mer informasjon annet enn om et utsagn er sant.

Eksempel på et zero knowledge proof:

Dersom man ønsker å gå inn på en bar, må man bevise alderen sin. I dag skjer dette ved å vise frem et gyldig bevis, slik som pass eller førerkort. Ved bruk av SSI kan dette heller skje gjennom digitale bevis. Men i dette tilfellet ønsker man ikke å dele fødselsdatoen sin med vekteren i tilfelle vedkommende vil misbruke det. Dette kan man løse ved å ta i bruk nullkunnskapsbevis. Man ønsker altså å bevise at man er over en viss alder uten å vise fødselsdatoen sin.

2.5 DID - Desentralisert Identifikator

For å skape et SSI-økosystem stilles det et fundamentalt krav om globale unike identifikatorer. Problemet var tidligere at ingen eksisterende standarder sørget for både en global unik identifikator og muligheten til å ha dette desentralisert.

Desentralisert Identifikator (DID) er en ny type identifikator og standarden legger til rette for at en digital identitet kan være både desentralisert og verifiserbar. En DID refererer til et

subjekt i form av et DID-dokument. Et DID-dokument inneholder metoden for hvordan en bruker kryptografisk kan bevise sitt eierskap av DID'en. Eierskapet bevises gjerne gjennom å demonstrere kontroll over en privat nøkkel. DID-dokumentet kan også inneholde forskjellige offentlig nøkler som er brukt under denne identiteten. Eksempler på nøkler i et DID-dokument er identitetens offentlige RSA og DSA nøkler.

Et av problemene med DIDs er at det per i dag finnes over 100 standarder, noe som gjør at det tar lang tid å støtte alle typene. En DID-metode er kode som konverterer en spørring av en DID til et faktisk DID-dokument. Siden det ikke er standardisert, gjør ulike DID-metoder dette forskjellig i dag. Dette problemet vil kanskje bli mindre med tiden, da konkurranse mellom metodene sannsynligvis fører til at de beste metodene overlever. Eller så kan mengden DID-metoder løses ved å ha en «universal resolver», som samler alle metodene ett sted og konverterer standardiserte input til standardisert output uavhengig av DID-metode (DWH, 2021).

DIDs for å bevise eierskap av offentlige nøkler

For at en verifiserende tjeneste skal kunne stole på autentiseringsprosessen og integritetssjekk gjennom Public Key Infrastructure (PKI), må tjenesten vite at den offentlige nøkkelen korresponderer til riktig eier. For at desentralisert kommunikasjon mellom digitale tjenester og lommeboken skal være sikker, kreves en sikker og skalerbar måte for at identitetseiere kan bevise eierskap ovenfor deres offentlige nøkler. DIDs gir oss mulighet til dette (Reed & Preukschat, 2021).

2.6 JWT

JWT (JSON Web Token) er et rammeverk for sikker informasjonsoverføring av JSON objekter mellom ulike parter (Harris, 2021). En JWT kan signeres og innholdet kan krypteres, noe som er hensiktsmessig når en VC skal overføres sikkert mellom utsteder, lommebok og tjeneste.

2.7 Distributed Ledger

Distributed Legder (DL) er en database som er delte med konsensus og synkroniserer digitale data som er spredt over flere nettsteder, land eller institusjoner (Majaski, 2021). Blokkjeder er et eksempel på Distributed Ledger.

2.8 Grunnidentitet

For at utsteder skal kunne gi ut en VC, må brukeren sende med en form for grunnidentitet til utsteder. Dette kan gjøres på ulike måter, men grunnidentiteten skal bekrefte identiteten til brukeren. Det er viktig at utstederen av grunnidentiteten er stolt på av de andre aktørene i SSI-økosystemet.

3 Analyse

For å kunne tilby et dypere dykk i hvordan et SSI-økosystem kunne fungert for offentlig sektor, og for å forstå teorien bedre i praksis, har vi laget en prototype (POC). Denne prototypen består av alle de fire aktørene i et SSI-nettverk, men er skalert betydelig ned slik at det tillot seg å utvikle fire noder på et nettverk samtidig og innen tidsfristen.

Dette gjør at vi ikke har fått testet alle teknologivalg, spesielt er ikke teknologier som baserer seg på Distributed Ledger (DL) testet i praksis. Derfor er denne rapportens innhold om DL basert på tidligere erfaringer og artikler på nett.

3.1 Teknologivalg

I prototypen har vi tatt i bruk en del ulike teknologier. I utsteder har vi kun backend og har tatt i bruk Java og SpringBoot. I lommeboken har vi kun frontend og har valgt å bruke React Native til dette. Tjenesten har både frontend og backend, og her bruker vi React i frontend med Java og SpringBoot i backend. Vi velger å ikke gå dypere inn i teknologivalgene våre da dette ikke vil være særlig relevant for videre utvikling.

3.2 Utsteder

En utsteder har ansvar for å produsere, signere og distribuere gyldige bevis. Dette kan være en organisasjon, et statlig organ, en person eller en ting, for eksempel en måler i et kjernekraftverk (Sovrin, 2021). Det viktigste med en utsteder er at de kan utgi gyldige bevis som ikke kan forfalskes. I den digitale verden kan dette oppnås ved bruk av digitale signaturer. Disse gir høy kredibilitet til hvem som har sendt et bevis og sørger for at det er tydelig dersom det er manipulert (tamper evident).

I et SSI-økosystem må alle tjenester selv evaluere om de stoler på utstederen, eller så må et øvrig organ bestemme at bare tillittsverdige aktører får utstede bevis. Etter tillitten er etablert, vil ethvert bevis med utsteders signatur bli godtatt.

Utstedere skal ta imot en grunnidentitet for å utstede VC'en som brukeren ønsker. Det er en spesiell type utsteder som utsteder denne grunnidentiteten. Et eksempel på dette kan sees i vedlegget 8.4 JWT-Standard – Eksempel på Grunn-ID. Grunnidentiteten blir brukt til å verifisere identiteten til brukeren. Implementasjonen som er gjort i forbindelse med vår PoC utstedes en grunnidentitet etter en innlogging i ID-porten.

En VC signeres av en utsteder, der utstederen benytter sin private nøkkel. Det er viktig å sørge for har tilgang til utstederens korresponderende offentlige nøkkel, uten å måtte sende en forespørsel til utstederen. Det er her VDR kommer inn. En utsteder kan ha en eller flere unike identifikatorer i VDR, der hver identifikator korresponderer til én offentlig nøkkel. Dette vil gjøre tjenesten sikker på at ingen kan utgi seg for å være en etablert utsteder, og at utsteder faktisk er den han utgir seg for å være. Hvordan VDR'en ser ut, kommer vi tilbake til

i et senere avsnitt. Vi fikk ikke tid til å implementere permanente nøkler til usteder, og har derfor opprettet nøkkelpar til hvert bevis.

3.3 Lommeboken

En digital lommebok skal kunne lagre bevis, nøkler og annen sensitiv data (Johnson, 2021). Ved hjelp av lommeboken kan en bruker blant annet forespørre og motta en VC fra en utsteder og dele VP med en tjeneste. En lommebok skal være lett å ta i bruk, være sikker og det er brukeren selv som skal velge hvem hen vil dele informasjon med.

3.3.1 Lignende apper og funksjonalitet

Vi startet med å undersøke om det fantes noen rammeverk og ferdige lommebøker som vi kunne benytte oss av. Vi fant et par ferdige lommebøker, som COV-ID Wallet (TrustNetPK, 2021) og Trinsic (Trinsic, 2021), der den siste både er et rammeverk og en ferdig lommebok. Vi så også på andre rammeverk som Evernym Mobile SDK (Evernym, 2021). Til slutt så vi på Microsoft Authenticator. Microsoft har iverksatt en beta-versjon av VC'er og hvordan disse kan brukes med en form for lommebok-applikasjon (Neira, 2021). Etter å ha sett på en del rammeverk og ferdige lommebøker, kom vi fram til at vi ønsket å lage lommeboken selv. Vi valgte dermed å ikke benytte oss av et rammeverk eller en ferdig lommebok. Dette gjorde vi fordi vi så for oss at det kom til å gi oss bedre innsikt i hva som er viktig å ha med i lommeboken, og generelt hva som burde være med i SSI-økosystemet. På grunn av dette ble vi nødt til å gjøre vurderinger underveis. I tillegg så vi for oss at dersom vi hadde startet med et rammeverk eller en ferdig lommebok, ville vi bli låst til den løsningen.

Vi har også sett på andre apper med funksjonalitet som kan være relevant for prosjektet vårt. Blant annet har Vipps og Førerkort-appen en «onboarding»-prosess som bruker ID-porten. Vipps har funksjonalitet for autentisering og identifikasjon. Dersom man bruker Vipps til å identifisere seg mot tredjeparts-tjenester vil man få en liste over disse, med mer detaljert informasjon om hvilke personopplysninger man har delt med disse. Dette ligner deler av funksjonaliteter vi ønsker i en digital lommebok. Det kan tenkes at Vipps i årene som kommer vil kunne tilby løsninger tilsvarende en digital lommebok slik SSI-tankegangen beskriver.

Førerkort-appen har førerkortet inne i appen som et bevis, som en kontrollør kan verifisere med en tilhørende QR-kode. Et interessant moment ved førerkort-appen er at det er bilde av personen, for å sjekke at den som blir kontrollert faktisk er den personen førerkortet er ment for. Avhengig av implementasjon og hvordan bevis utveksles og verifiseres, kan et slikt bilde i lommebok-appen være aktuelt. Dette kan forhindre mange tilfeller der en person har lånt eller urettmessig fått tilgang til en annens lommebok, og forsøker å late som de har bevis de ikke har.

3.3.2 Hva brukeren kan gjøre i lommeboken

I prototypen vår kan brukeren:

1. Onboarde (registrere seg) i lommeboken via ID-porten, brukeren vil da motta en grunnidentitet. Dette trenger man kun å gjøre én gang.
2. Opprette en personlig kode for å kunne logge inn i lommeboken, etter å ha mottatt grunnidentitet.
3. Logge ut av lommeboken.
4. Forespørre og motta VC fra en utsteder.
5. Sende VP til en tjeneste.
6. Brukeren har oversikt over alle bevisene sine og hvem disse er delt med.
7. Slette et bevis fra lommeboken.
8. Dele et bevis med en fysisk tjeneste, for eksempel Vinmonopolet, ved å bruke en QR-kode som ligger i lommeboken.
9. Svare på forespørsler fra en tjeneste om å få tilgang til et bevis.
10. Slette brukeren sin – da vil også alle bevis bli slettet.

Videre burde en bruker også kunne gjøre dette i en lommebok:

11. Konstruere en VP ut fra en eller flere VC'e
12. Mulighet til å fjerne tilgangen tjenester har til bevis.
13. Logge inn på en tjeneste med lommeboken via ID-porten, lommeboken blir da en ny påloggingsmekanisme i ID-porten.
14. Brukeren skal kunne hente ut et bilde av seg selv fra en utsteder

3.3.3 Sikkerhet i lommeboken

I prototypen vår har vi valgt å ikke implementere en backend. Dette er fordi vi ønsker at det er brukeren som skal ha kontroll over lommeboken og hva som deles. Altså ønsker vi ikke at det skal ligge data noe annet sted enn i selve lommeboken, med mindre brukeren selv velger det. I tillegg ønsker vi ikke at brukerens private nøkler skal sendes mellom en frontend og en backend, da dette kan utsette applikasjonen for en rekke cyberangrep. Eksempel på angrep kan være Cross Site Request Forgery(CSRF), Man-in-the-middle angrep og Denial-of-service angrep. Vi så det derfor som tryggere å kun ha frontend. En annen grunn til at vi valgte å ikke ha backend er at etter å ha planlagt hvilke funksjoner vi ønsket at lommeboken skulle ha, kom vi fram til at det ikke var mye en backend kunne gjøre. Dette fordi alle funksjonalitetene kunne lages kun ved bruk av frontend.

Derimot kan større prosesser, slik som generering av offentlige og private nøkler, gjøre appen treg med tanke på at en mobil har begrenset med prosessorkraft. En eventuell backend-side av applikasjonen kan avlaste frontend-siden for slike operasjoner, og dermed gjøre brukeropplevelsen bedre.

Når det kommer til sikkerheten i lommeboken til nå, er ikke lommeboken sikker. Dette fordi vi først ønsket å implementere den viktigste funksjonaliteten til lommeboken. Vi tenkte at det kom til å ta tid å gjøre lommeboken sikker, noe som kunne gått på bekostning av andre deler av økosystemet. Når det kommer til lagring, bruker vi AsyncStorage. AsyncStorage er en ukryptert, asynkron lagringsmetode som lagrer i form av <nøkkel, verdi>. Ettersom lagringen i AsyncStorage er ukryptert, fører dette til at lagringen er ubeskyttet. Dataen som lagres burde derfor krypteres, noe den ikke blir nå. For å gjøre denne lagringen sikrere kan man benytte seg av for eksempel react-native-keychain, som vil lage en sikker nøkkel, og denne nøkkelen vil kryptere dataen (Kalveram, 2021)

3.3.4 Lagring av bevis

Videre kan det diskuteres om bevisene burde lagres lokalt eller ikke. Fra kravene til EU burde data lagres lokalt slik at alt kan brukes uten internett (Europakommisjonen, 2021). En bruker skal kunne være hvor som helst og fortsatt ha muligheten til å bevise noe. En annen fordel med at bevisene lagres lokalt er at brukeren selv har kontroll på hvor bevisene sine ligger lagret, og oversikt over hvilke tjenester som har tilgang til dem. Dersom man benytter seg av skybasert lagring vil man ikke ha tilgang på filene som er lagret dersom nettet er nede på enten brukerens eller leverandørens side (CAS, 2021).

På den andre siden kan det i praksis være mer hensiktsmessig å lagre bevisene på en sky, noe som vil fungere som en backend. Det kan argumenteres at man ønsker en backup av egne bevis dersom man for eksempel mister mobilen. For å kunne benytte seg av bevis offline, kan det være mulig å cache dataen slik at man kan bruke det uten dekning (Edwin, 2021).

Encrypted Data Vaults (EDV) kan være et godt alternativ for sikker lagring. EDV'er tillater brukere å lagre data på en skytjeneste slik at lagringsleverandøren ikke har tilgang på dataen. Med EDV lagres data kryptert, hvor klienten selv krypterer og dekrypterer ved bruk av nøkler tilknyttet deres desentraliserte identifikatorer. Dette sørger for at de får kontroll på egen data som er lagret i skyen (Transmute, 2021). EDV'en sørger for at både dataen alltid er tilgjengelig og at den er beskyttet mot databrudd av lagringsleverandøren (W3C, Encrypted Data Vaults 0.1, 2021).

3.3.5 Sending av bevis

Det finnes hovedsakelig to ulike måter en bruker kan sende bevis fra lommeboken på, pull og push. Push handler om at brukeren initierer samhandlingen med tjenesten fra lommeboken, og velger her hvilke bevis som skal sendes. Pull går ut på at en tjeneste ber om bevis av en bestemt type, som brukeren må akseptere at blir utlevert. Det finnes både fordeler og ulemper med begge disse typene, se tabellen under.

	Pull	Push
Fordeler	Mer intuitivt for brukeren. Brukeren trenger bare å godta. Brukeren kan godta å sende bare nødvendige bevis. Bruker risikerer da ikke å sende for mye, eller feil informasjon.	Brukeren har full kontroll over det som sendes. Brukeren velger selv hvilken tjeneste han vil sende til, og det er brukeren som starter kommunikasjonen. Mindre sjanse for å bli lurt.
Ulemper	Tjenesten starter kommunikasjon, og kan dermed utgi seg for å være noen de ikke er, eller etterspør informasjon uten gyldig grunn. Krever mer tillit til tjenesten.	Brukeren kan sende feil bevis. Brukeren kan sende mer data en nødvendig.

Ettersom vi kun har frontend kom vi fram til at det var lettest å implementere push først. Her vil brukeren velge hvilke bevis som skal sendes til en tjeneste, og sende disse bevisene til tjenesten. Vi har senere implementert en variant av pull der brukeren skanner en QR-kode på nettsiden til en tjeneste, og vil da få opp hvilke bevis tjenesten ønsker i lommeboken. Denne forespørselen kan brukeren velge å godta eller ikke, og dersom brukeren godtar vil bevisene sendes til tjenesten. I tillegg til dette har hvert bevis en QR-kode som en eventuell fysisk tjeneste kan skanne for å få opp beviset.

Når det kommer til push og pull, ser vi på det som best å hovedsakelig ha pull, da dette vil være mer intuitivt for brukeren. Dette vil også være positivt med tanke på at brukeren kun sender det som blir etterspurt, verken for mye informasjon eller feil informasjon.

3.4 Tjenesten

En tjeneste i SSI-økosystemet har som rolle å kreve et eller flere bevis av visse typer fra en bruker, for at brukeren skal få lov til å benytte seg av tjenesten. Bevis som blir mottatt skal være knyttet til brukerens digitale lommebok. En tjeneste kan være en nett-tjeneste eller en fysisk tjeneste, for eksempel Vinmonopolet som krever et bevis for at kunden er over 18 år gammel. Ved fysiske tjenester kan for eksempel tjenesten selv skanne en QR-kode fra lommeboken, der QR-koden inneholder selve beviset.

3.4.1 Hva skal en tjeneste gjøre?

Når en tjeneste mottar et bevis må den kunne verifisere at det ble utstedt av en troverdig utsteder, at beviset ikke har blitt endret på siden det ble utstedt, og at den ikke er utløpt

eller blitt trukket tilbake av utstederen. I tillegg er et spørsmål om det finnes kryptografisk bevis om at innehaveren av beviset korresponderer til brukeren som beviset omhandler (Affinidi, 2021).

Det er viktig at beviset følger standardiseringen fra tillitsrammeverket slik at tjenesten kan lese det. Utformingen av beviset kan brukes til å sjekke at beviset omhandler riktig bruker, og at beviset er av typen(e) som ble forespurt, ved at tjenesten sjekker ulike felt i bevisutformingen.

Dersom beviset blir verifisert av tjenesten, skal brukeren få lov til å benytte seg av tjenesten som har krevd beviset.

3.4.2 Tjeneste i POC

Tjenesten mottar en Verifiable Presentation (VP) i form av JWT, som inneholder en eller flere Verifiable Credentials (VC'er) basert på hva tjenesten krever, samt en identifikator for lommeboken, og metadata for VP'en.

For å sørge for autentisering og integritet av VP'en, kan tjenesten verifisere den digitale signaturen til VP'en ved hjelp av offentlig nøkkelkryptering, se krypterings-delen. Tjenesten slår opp den offentlige nøkkelen til brukerens lommebok i tillitsrammeverket ved å bruke lommebokidentifikatoren. For å sjekke at den autentiserte brukeren korresponderer til brukeren som VC'ene omhandler, skal issuer-feltet i VP'en sammenlignes med subject-feltet i VC'ene (W3C, Verifiable Credentials Data Model 1.0 - 6.2 JSON-LD, 2021)

På samme måte kan tjenesten verifisere autentisering og integritet av VC'ene. Signaturen til hver VC verifiseres ved å hente ut den offentlige nøkkelen til utstederen på tillitsrammeverket, med utstederidentifikatoren i VC'en. Tjenesten må her stole på at tillitsrammeverket kun lagrer informasjon om troverdige utstedere. Hvis signaturen blir verifisert vet tjenesten at beviset er uforandret etter å ha blitt utstedt.

For å verifisere at VP'en inneholder VC'er av forespurte typer, sjekkes hvert VC mot tillitsrammeverket. Ulike type-felt i VC-utformingen må samsvare med typestandardiseringer som ligger på VDR'en. Samtidig må tjenesten sjekke at alle forespurte typer korresponderer med VC'ene i VP'en.

3.4.3 Hvilke tjenester kan verifisere?

Alle VP'er inneholder et aud-felt, som lar bruker angi hvilken tjeneste hen tillater å verifisere beviset. Alle JWT-verifikatorer/tjenester som ikke er identifisert i dette feltet vil være pålagt til å avvise JWT'en (W3C, Verifiable Credentials Implementation Guidelines 1.0 - 8 Using the JWT aud claim, 2021). Dette oppfyller SSI sitt ønske om at brukeren selv velger hvem som skal ha tilgang til egen informasjon.

3.5 Tillitsrammeverket

Tillitsrammeverket (verifiable data registry) er en omdiskutert aktør i økosystemet. Enkelte mener at dette burde være en obligatorisk del av økosystemet, mens andre mener det er kostbart og unødvendig. VDR kan være databaser man har tillit til, desentraliserte databaser, myndighetene sin ID-database eller distributed ledger (DL). I de neste seksjonene skal vi diskutere fordeler og ulemper med blokkjedeløsninger, skytjenester og X.509 sertifisering (W3C, Verifiable Credentials Data Model - 3.2 Credentails, 2021).

3.5.1 Blokkjede

Det er lite tvil om at desentralisering står sterkt i SSI tankesettet. Når man ser på omfanget av kilder det finnes om blokkjeder og SSI, samt EUs arbeid med EBSI (European Blockchain Services Infrastructure) (EU, EBSI, 2021) kan det tyde på at det er mange som har bestemt seg for at dette er løsningen. NIST definerer blokkjeder som en type distributed ledger som er motstandsdyktig og sikre mot manipulering. Disse blir implementert på en distribuert måte og vanligvis uten styring fra sentrale myndigheter (NIST, Blockchain Overview, 2021). Dette passer godt inn i definisjonen til SSI, men det er fortsatt mange spørsmål å ta stilling til. Har teknologien kommet langt nok? Hva skal legges til blokkjeden? Skalerer det godt nok og hvordan opprettholder man personvern?

Blokkjeden er ikke bare desentralisert og uforanderlig, den er også et nettverk av maskiner kjørt av så mange mennesker overalt i verden at nedetid er så å si umulig (Bhushan, 2021). Det gir også mulighet til å bli uavhengig av tredjeparter med tillit. Dette på grunn av at tjenester når som helst skal kunne verifisere påstander gjennom å hente utsteders nøkkel fra blokkjeden og kan dermed verifisere det uten å kontakte utsteder direkte (EU, Blockchain and Digital Identity, 2021).

Det er viktig at ingen sensitive opplysninger blir lagret i klartekst på blokkjeden, da dette vil føre til at alle kan lese denne informasjonen. Sensitiv informasjon burde heller håndteres offchain eller ved å bruke hash. Dersom data blir lagret offchain kan de blant annet lagres i lommeboken, men kan også kombineres med onchain funksjonalitet slik at bare deler av informasjonen blir lagret på blokkjeden (NIST, A Taxonomic Approach to Understanding , 2021). Brønnøysundregisteret sin aksjeeierbok bruker både onchain og offchain data. Alt som er persondata holdes offchain grunnet GDPR. Eksempelvis står det da onchain at en ethereumadresse **eier** 10000 aksjer i Symfoni AS, men hvem som er skjult bak adressen må hentes ut fra Brønnøysundregisteret sitt offchain "orakel" som er et vanlig API (Ramvi, 2021).

EU sitt blokkjede-forum har kommet med forslag til ulike formål man kan bruke teknologien til i denne sammenhengen. Under forklarer vi kort om de vi anser som mest relevante (EU, Blockchain and Digital Identity, 2021).

- Blokkjedeadresser fungerer godt sammen med DID. Dette er fordi blokkjedeadressene unikt blir generert av brukeren selv og allerede utnytter offentlig/privat-nøkkelkryptografi.
- Blokkjeden kan også brukes til å lagre et DID-register for å knytte DID opp mot eieren. Dette er spesielt aktuelt for utstedere, slik at tjenester kan vite at den gitte utstederen er den samme som har utstedt VC'en den ønsker å sjekke gyldigheten til.
- Et annet nyttig felt for blokkjeder er å putte en hash knyttet til ulike VC'er sammen med tidspunktet VC'en ble utstedt. Dette kan være nyttig for å sørge for gyldighet av VC. En tjeneste kan da sjekke at det er samme VC som blir sendt til dem ved å 'hashe' en mottatt VP og sammenligne den med hashen som ligger på blokkjeden. Dermed er brukerens personvern også ivaretatt, da det bare er brukeren og tjenester brukeren har godtatt tilgang til som kan se informasjonen som ligger i VC'en.
- Rettighet til tilgang av informasjon kan bli sendt som en transaksjon til blokkjeden sammen med dens utløpsdato. Dette kan være med på at brukeren selv kan velge hvor lenge en gitt tjeneste skal kunne ha tilgang til informasjonen og er da med på å ivareta prinsipp fra GDPR (EU, Blockchain and Digital Identity, 2021, s. 16).

Etter kommunikasjon med Snorre Lothar von Gohren Edwin fra DIN kommer det frem at bare de to første punktene er nødvendig for å bruke blokkjeder som VDR.

Ethereum og smart kontrakter

Ethereum gjør det mulig å opprette DID gjennom dens adresser og kan gjennom en smart kontrakt lagre DIDs i for eksempel et register for godkjente utstedere. En "smart kontrakt" er bare en liten bit kode som kjører på Ethereum. Det kalles en "kontrakt" fordi kode som kjører på Ethereum kan kontrollere verdifulle ting som ETH eller andre digitale resurser (Ethereum, Ethereum, 2021). Snorre fra DIN forklarer at Ethereum har en did:ethr metode som er en smart kontrakt som holder på en did:ethr state. Har man et gyldig nøkkelpar kan man gjøre endringer på dette DID dokument. Har man en adresse kan man resolve hvilken som helst adresse til et DID dokument, did:eth gjør som vist i fotnote[1].

Som nevnt tidligere ser man for seg at det kan være nyttig å lagre et DID register med gyldige utstedere sin DID. En tanke som vi ikke har fått tid til å teste ut i praksis er at Digdir kan lage et UI som gjør det lettere for statlige utstedere å legge sin DID på blokkjeden. Symfoni har laget en React plugin med navn hardhat[2] som gjør det enklere å opprette en smart kontrakt i en React-applikasjon. Dette kan trolig være et godt verktøy for å gjøre det mulig å opprette et standard UI til bruk for gyldige utstedere.

[1] https://github.com/uport-project/ethr-did-registry/blob/develop/contracts/EthereumDIDRegistry.sol
[2] https://hardhat.org/tutorial/

Tilbaketrekking

Utstedere kan produsere et signert tilbaketrekkingstoken, i forbindelse med utstedelse av VC, som blir lagret av både utsteder og lommebok. Tokenet inneholder en hash av VC'en, utsteders DID og lommebokens DID. Ved tilbaketrekking må dette tokenet publiseres i en tilbaketrekkingsliste (revocation list) på DL. Denne listen vil sjekkes av en tjeneste ved verifikasjon av VC'er i VP. For å legge et token på listen må signaturen verifiseres. I tillegg må DID'en til den som ønsker å publisere samsvare med en DID i tokenet. Dette er en kryptografisk sikker måte å sørge for at bare autoriserte aktører kan trekke tilbake et bevis (Abraham, More, Rabensteiner, & Hörandner, 2021).

Offline verifisering

Dersom både bruker og verifikator er uten internett, kan ikke validiteten til et bevis sjekkes, for eksempel ved å sjekke tilbaketrekkingslisten. Brukeren kan da på forhånd be om et token fra DL, og denne burde inneholde et tidsstempel for når gyldigheten ble attestert. Dette tokenet vil ha kortere levetid i forhold til VC'en. Det er utstederen som bestemmer levetiden, og dette er bestemt før brukeren ber om tokenet. Dersom en VC er tilbaketrukket, vil ikke dette tokenet bli utstedt fra DL. På denne måten vil man sikre at beviset fortsatt er gyldig med jevne mellomrom, selv uten nettforbindelse (Abraham, More, Rabensteiner, & Hörandner, 2021).

Skalerer det godt nok?

Skalering er i dag et problem i forbindelse med blokkjede. Alle noder i en blokkjede må ha all data på DL, som gjør at det blir umulig å slette data som er lagret på en blokkjede. Dette gjør også at blokkjede bare kan vokse til en viss størrelse uten å måtte kreve for mye plass og prosessorkraft. Det blir derfor et spørsmål om å skalere og håndtere belastningen til at for eksempel hele Europa bruker samme kjede. En løsning kunne vært å bruke flere ulike blokkjeder som hver er laget for sitt land eller område. Totalt vil det ved en god standardisering være mulig å kommunisere mellom kjedene dersom DIDs blir brukt. Dette siden DIDs sier noe om hvor brukeren er registrert og på hvilken kjede det finnes mer informasjon om brukeren. De ulike blokkjedene jobber også med å løse dette problemet. Ethereums Layer 2 løsning begynner å skalere ganske bra og om rundt to år kommer Ethereum 2.0 som ytterligere vil skalere bedre.

Miljø

I dag er det Proof of Work som er den vanligste konsensusalgoritmen som blir brukt i blokkjeder. Algoritmen blir brukt til å bekrefte en transaksjon og legge en ny blokk til kjeden. I algoritmen konkurrerer en gruppe mennesker, også kalt miners, om å legge til denne blokken. De får en matematisk utfordring (challenge) som må løses, og den første som løser denne får legge til blokken på kjeden og får da også en belønning (Javatpoint, 2021). Det kreves enorme menger prosessorkraft for å løse denne utfordringen og dermed også utrolig mye energi. Siden strømmen i Europa ikke bare er grønn, kan dette ha en negativ innvirkning på miljøet. De ulike blokkjedene jobber nå med en overgang til Proof of Stake

som også er en konsensusalgoritme (Ethereum, Proof-of-stake (POS), 2021). Dette gjør at det blant annet kreves langt mindre energi for å legge en ny blokk til kjeden. Det kan derfor hende at et skifte til denne algoritmen kan være en av løsningene på dette problemet. Vi anbefaler derfor å bruke en blokkjede som bruker Proof of Stake eller planlegger å gå over til dette i nær fremtid.

Tillit blant befolkningen

I tillegg til at blokkjeder er en relativt ny teknologi, er også begrepet blokkjede nært knyttet til sin bruk som valuta. Kryptomarkedet er veldig ustabilt, og mange er skeptiske til teknologien. Det er da et spørsmål om den generelle befolkningen vil ha tillitt til denne teknologien.

3.5.2 Skytjeneste

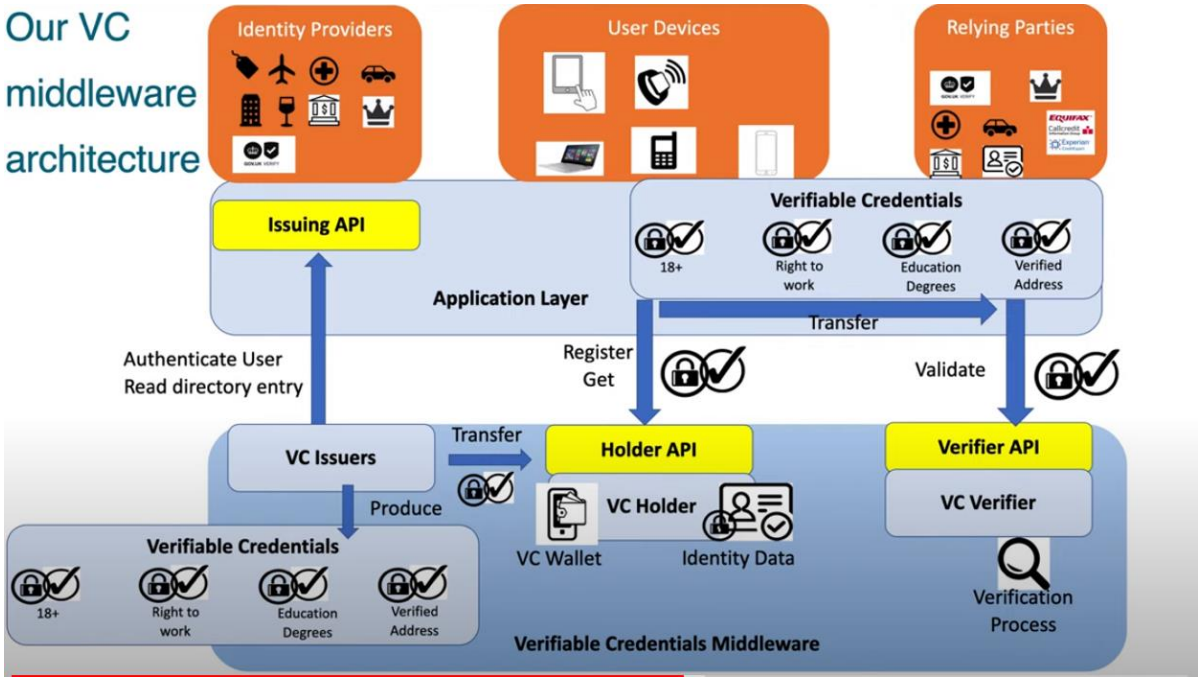
Et annet alternativ til VDR er å lagre de offentlige nøklene i en database på en skytjeneste. Fordelen med dette er at det er en mye enklere implementasjon enn for eksempel blokkjeder. En veletablert skytjeneste lover minimal nedetid gjennom en tjenestenivåavtale (service-level agreement). Skytjenester er gjerne tryggere enn lokale servere da man selv ikke har ansvar for vedlikehold. Likevel er det alltid en risiko ved å lagre sensitive personopplysninger i skyen. Et annet problem er at en skytjeneste konsentrerer mye data på ett sted, dette kan være risikabelt ved et angrep mot tjenesten. Dette går også litt imot SSI-tankegangen, da man gjerne ønsker en desentralisert lagring (Senel, 2021).

For tilbaketrekking i skytjeneste er det mulig å for eksempel slette offentlige nøkler eller å lage en tilbaketrekkingsliste. Siden skytjenester tillater sletting og oppdatering av informasjon, er det ikke så mange begrensninger for tilbaketrekking av bevis.

3.5.3 X.509 sertifikater

X.509 er et standardformat for offentlige nøkkelbevis og digitale dokumenter som sikkert knytter kryptografiske nøkkelpar til identiteter som nettsteder, enkeltpersoner eller organisasjoner (Russell, 2021). En stor fordel med X.509 sertifikater er at det er en foreløpig sikker og veletablert standard. Dermed vil løsninger bygget på denne standarden være kompatibel med andre eksisterende tjenester. Det vil bli enklere å etablere og krever ikke at tjenester som skal tilrettelegge seg etter lommeboken må sette seg inn i noe helt fremmed.

Et eksempel på en applikasjon som har benyttet seg av X.509 sertifikater er FIDO2 (Fast Identity Online) (Chadwick, 2021). Her argumenteres det for at blokkjeder eller annen lagringsform ikke er nødvendig når X.509 sin tillitsinfrastruktur blir brukt. Chadwick mener heller at det både er unødvendig og kostbart med bruk av blokkjeder. I FIDO2 er det en egen FIDO server som lagrer den offentlige nøkkelen, brukernavn og annen unik informasjon.



Figur 1: FIDO2 sin VC-arkitektur (Chadwick, 2021)

Figur 1 illustrerer hvordan FIDO2 har løst arkitekturen for å sende og verifisere en VC. Autentiseringen er det FIDO som tar seg av, da den har kontroll over at brukeren bruker den samme offentlige nøkkelen hver gang. Applikasjonene har også her behov for en utsteder som sender ut en form for grunnidentitet for å identifisere brukeren. Arkitekturen baserer seg på kortlevde VC'er etter behov, og har dermed ikke behov for avanserte tilbaketrekkingsmekanismer.

Selv om Chadwick argumenterer sterkt for at X.509 er den beste løsningen på sending og mottak av VC, er det vanskelig å finne andre legetime kilder som mener det samme. Det gjør at man må spørre seg selv om det er på tide med et skifte av standard. Det er ikke til å legge skjul på at de største tilhengerne av SSI-tankesettet har skylapper når det kommer til blokkjede som VDR. Dette skyldes blant annet desentraliseringen en blokkjede tillater. Når man bruker x.509 sertifikater er man avhengig av at en sertifikatmyndighet verifiserer enhetens identitet (Russell, 2021). Siden SSI prøver å gå vekk i fra avhengighet til tredjeparter, er det en mulighet at standarden gradvis vil bli mer utdatert.

For å konkludere har FIDO2 vist at det er fullt mulig å gjøre X.509 sertifisering modent nok til bruk i SSI-økosystemet. Vi tror likevel det er viktig å følge med på de nye trendene med DID og desentraliserte løsninger til VDR, da mye tyder på at det er mange som har bestemt seg for at det er det som er fremtiden.

3.5.4 VDR i POC

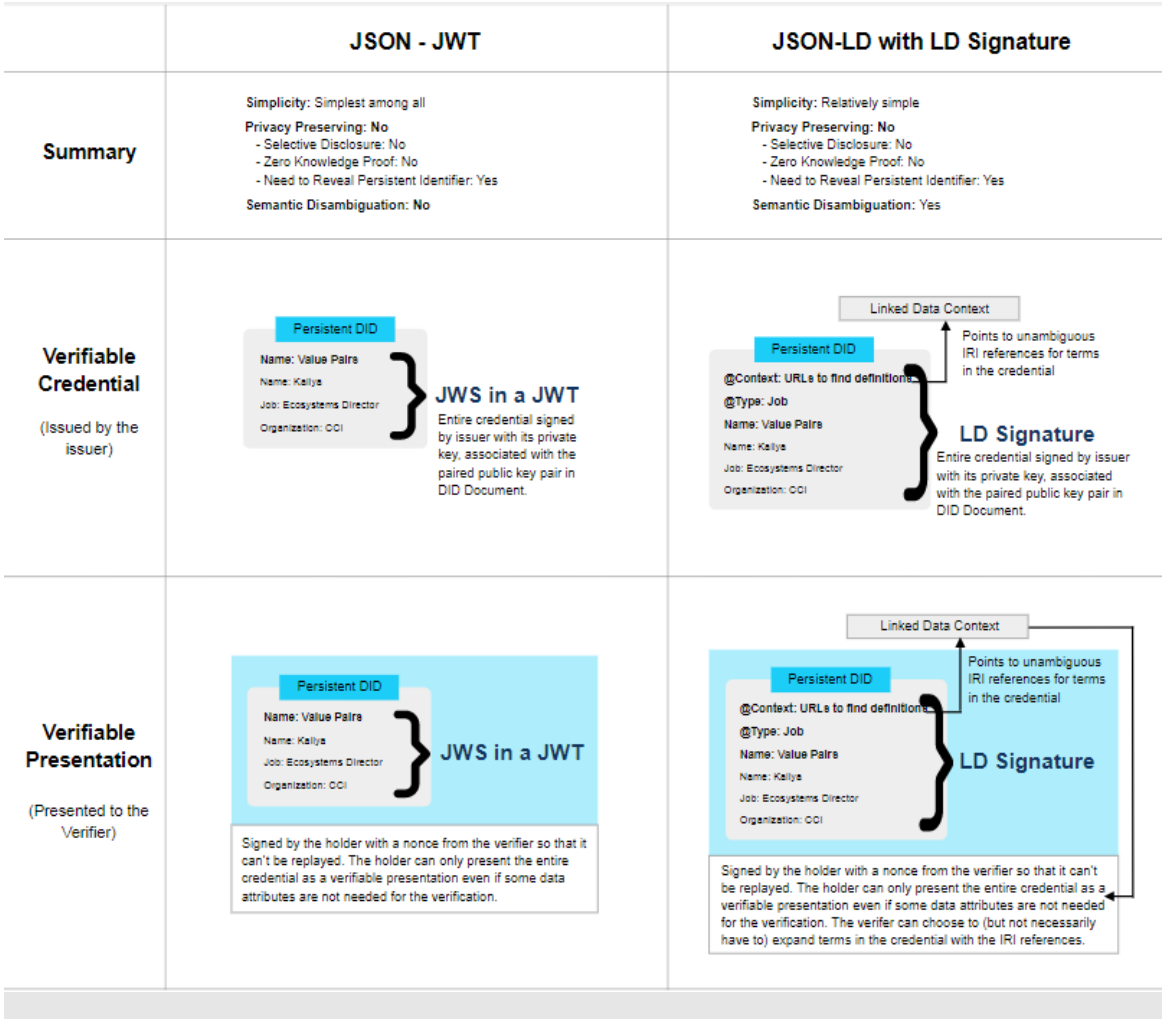
I POC'en har vi valgt å implementere VDR som en tekstfil da dette var en enkel løsning som fortsatt ga oss en forståelse av aktøren. I denne filen lagrer vi en lommebok-ID for hver bruker og tilhørende offentlig nøkkel, samt offentlige nøkler tilknyttet utsteder-ID for de ulike utstederne. Ut fra de tre alternativene vi har presentert ligner dette mest på en skytjeneste. Filen, og dermed VDR, er inntil videre lagt hos utsteder.

4 Kommunikasjon

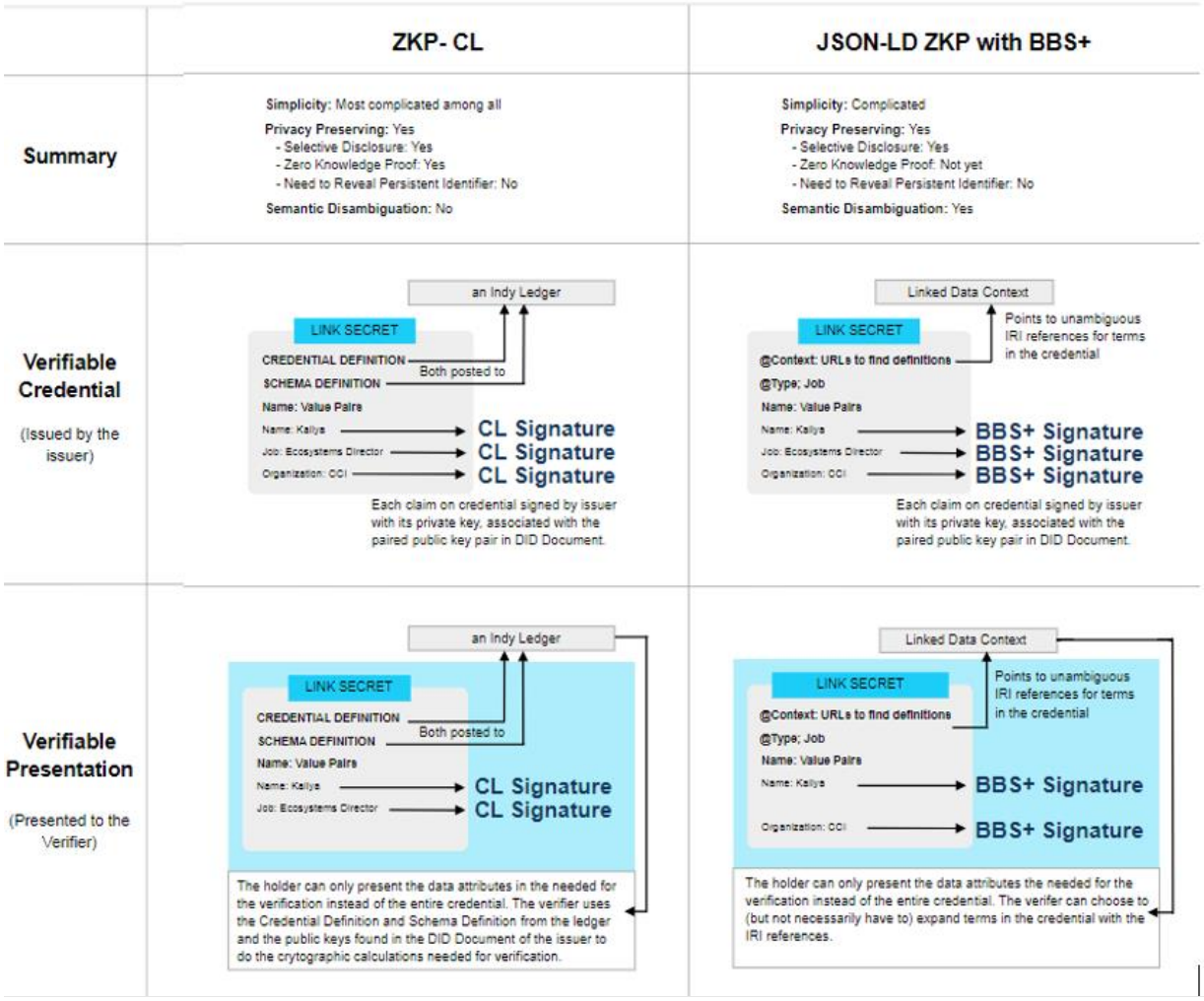
I dette kapittelet drøfter vi forskjellige metoder en kan bruke til å sende bevis (VC og VP) over nettet på en sikker måte. Hver metode har en tilhørende signeringsmetode. I Figur 2 og 3 under ligger først en oversikt over alle metodene, før vi går gjennom JWT og JSON-LD + LD-Signature, som vi har mest erfaring med. Senere har vi også lagt til JSON-LD + BBS+. Her er det viktig å finne en balanse mellom en type som er enkel å bruke for bedrifter, og i tillegg passer bra inn i SSI-økosystemet. Hvor enkelt det vil være å ta i bruk for organisasjoner, spørs hvilken type man går for.

Metoder:

- JSON + JSON Web Signature (JWT)
- JSON Linked Data + Linked Data Signature (JSON-LD + LD-Signature)
- ZKP + Camenisch-Lysyanskaya Signatures (ZKP-CL)
- JSON-LD + Zero knowledge proof with BBS+ (JSON-LD ZKP with BBS+)



Figur 2: JSON-JWT / JSON-LD with LD Signature. https://www.lfph.io/wp-content/uploads/2021/04/Verifiable-Credentials-Flavors-Explained-Infographic.pdf



Figur 3: ZKP-CL / JSON-LD with BBS+. https://www.lfph.io/wp-content/uploads/2021/04/Verifiable-Credentials-Flavors-Explained-Infographic.pdf

4.1 JSON + JSON Web Signature (JWT)

I POC'en valgte vi å ta i bruk JSON Web Tokens (JWT), som VP og VC, da dette var enkelt å implementere og fordi W3C nevner det som et alternativ til måter å sende VP og VC. En JWT koder et sett av et eller flere bevis som et JSON-objekt og blir lagt inn i en JSON Web Signature (JWS) (W3C, Verifiable Credentials Data Model 1.0 - 6.3.1 JSON Web Token, 2021). JWT blir mye brukt i dag til å sende informasjon over nettet, og gjør det derfor enklere for eksisterende system og delta i SSI-økosystemet (W3C, Verifiable Credentials Data Model 1.0 - 6.3.1 JSON Web Token, 2021). Til å bruke i prototypen vår har vi laget vår egen JWT-standard ut fra anbefalingene til W3C Se vedlegget JWT-Standard.

Nettside med JWT-dekoder og forskjellige JWT-bibliotek: https://jwt.io/

Fordeler med JWT:

- Enkel implementasjon
- En standard som er mye brukt. Er derfor enklere for eksisterende eller nye system å bli med.

4.2 JSON Linked Data + Linked Data-signature(JSON-LD + LD Signature)

JSON-LD er et dokument basert på JSON og linked data. Linked data er data som inneholder lenker til annen data, gir en graf av informasjon og er en måte å skape et nettverk av maskinlesbar data på tvers av nettsider. Dette gir oss muligheten til å følge integrerte lenker til andre deler av linked data på andre nettsider. Siden JSON-LD er kompatibel med JSON medfører dette til en enkel overgang for eksisterende system (W3C, Verifiable Credentials Data Model 1.0 - 6.2 JSON-LD, 2021). JSON-LD tilfører noen felt i forhold til vanlig JSON, blant annet @context, @id og @type. Disse brukes henholdsvis til å beskrive sammenhengen man kommuniserer i, dokumentets universelle identifikator på nettet og hvilken type dokument det er (W3C Community Group, 2020).

Linked Data Signature er en type kryptografisk bevis for å signere JSON-LD-dokument. En signatur inneholder blant annet feltene type og creator. Type lenker til algoritmen som er brukt for å signere dokumentet, og creator lenker til den offentlige nøkkelen til eieren av signaturen. Den offentlige nøkkelen lenker videre til eieren og eieren linker tilbake til nøkkelen. Dette gjør at man kan stole på systemet da alle dokumenter er signert og linket til hverandre (Sporny, 2021).

Et bibliotek for implementasjon av LD-Signature i Java:
https://github.com/WebOfTrustInfo/ld-signatures-java

LD-signature:

Inkluderer:

type (required)
A URI that identifies the digital cryptographic suite that was used to create the signature. For example: `Ed25519Signature2018`.

created (required)
The string value of an [ISO8601] combined date and time string generated by the Proof Algorithm.

domain (optional)
A string value specifying the restricted domain of the signature.

***nonce* (optional, but strongly recommended)**
A string value that is included in the digital signature and *MUST* only be used once for a particular domain and window of time. This value is used to mitigate replay attacks.

***signature value* (required)**
One of any number of valid representations of *signature value* generated by the Proof Algorithm. Example: `jws` for detached JSON Web Signatures.

Eksempel:

A signature can be added to a Linked Data document like the following:

EXAMPLE 5: A simple Linked Data document

```
{
  "@context": {"title": "https://schema.org#title"},
  "title": "Hello world!",
}
```

by adding the parameters outlined in this section:

EXAMPLE 6: A simple signed Linked Data document

```
{
  "@context": [
    {"title": "https://schema.org#title"},
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ],
  "title": "Hello world!",
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2020-11-05T19:23:24Z",
    "verificationMethod": "https://ldi.example/issuer#z6MkjLrk3gKS2nnkeWcmcxi
      ZPGskmesDpuwRBorgHxUXfxnG",
    "proofPurpose": "assertionMethod",
    "proofValue": "z4oey5q2M3XKaxup3tmzN4DRFTLVqpLMweBrSxMY2xHX5XTYVQeVbY8nQA
      VHMrXFkXJpmEcqdoDwLWxaqA3Q1geV6"
  }
}
```

Fordeler med JSON-LD + LD-Signature

- Kan bruke terminologi som er universelt maskinlesbar.
- Kan uttrykke informasjon som er mer detaljert.
- Trenger ikke pre- eller postprosessering i motsetning til JWT
- Et system der all informasjon er linka og signert som gjør at man kan stole på det.
- Er relativt enkel å implementere.

4.3 JSON Linked Data + BBS+ Signature (JSON-LD BBS+)

Et av hovedprinsippene med SSI er minimalisering av data som deles. Dette betyr at man bare skal dele akkurat nødvendig informasjon om seg selv. For eksempel om en skal bevise at en er over en minstealder, for eksempel 18 år, skal ikke brukeren trenge å vise alderen sin. For å bevise at en bruker er over 18 uten å vise alderen, har MATTR utviklet en plattform som nytter BBS+ signaturer for å oppnå «selective disclosure». De kan da lage et bevis som støtter «Zero knowledge proof» uten å dele unødvendig informasjon om sluttbrukeren eller legge mer arbeid til utsteder (MATTR, 2021).

Dette skriver MATTR om BBS+ signaturer:

«*One of the benefits of using the BBS+ cryptographic scheme to sign credentials is the ability to derive a zero knowledge proof from the signature, where the party generating the proof can choose to partially disclose statements from the original message. When enabled, this feature allows issuers to create a credential that effectively enforces minimal data disclosure using the MATTR Platform and a compliant digital wallet*.» (MATTR, 2021)

Et bibliotek for bruk av JSON-LD + BBS+ Signature fra MATTR:
https://github.com/mattrglobal/jsonld-signatures-bbs

Fordeler med JSON-LD + BBS+:

- Støtter Zero knowledge proofs
- Støtter Selective disclosure

4.4 Lommebok via OIDC

For at brukere skal kunne logge på offentlige tjenester i dag, må de bruke ID-porten med blant annet BankID eller MinID. Et alternativ vi har sett på er å bruke lommeboken som et innloggingsalternativ i ID-porten. Dette er dog ikke forsøkt implementert i vår prototype, da det ikke strakk til med tid.

I dag loggføres alle innlogginger med ID-porten, noe som strider imot SSI-prinsippet. Dersom man bruker appen som et innloggingsalternativ til ID-porten vil kun én innlogging med ID-porten loggføres, nemlig første gang man logger inn i lommeboken. Senere ser vi for oss at man kun trenger for eksempel en pinkode for å komme inn i lommeboken, slik som i en rekke mobilbankapplikasjoner. Deretter kan man logge inn i offentlige tjenester via lommeboken. Dette er også en tjeneste som forenkler innloggingsprosessen til offentlig tjenester for brukeren.

Bruk av VC og VP er en ny teknologi og kan derfor være vanskelig å implementere. Å få lommeboken som et alternativt valg i ID-porten kan gjøre overgangen til bruk av VC og VP til verifisering enklere for offentlige tjenester.

4.4.1 ID-porten sender id-token til tjeneste

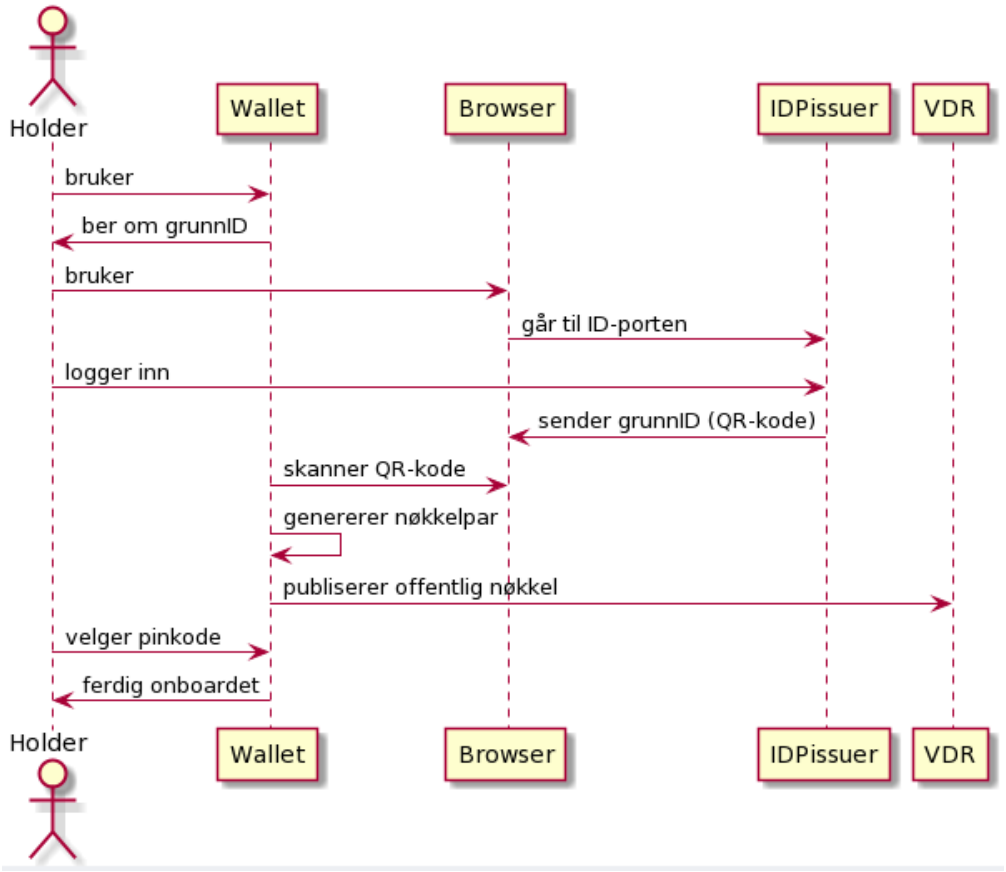
En annen løsning kan være at man bruker ID-porten på vanlig måte, bare at lommeboken er et alternativ. Her sender man ikke en VP, men heller at ID-porten sender id-token på vanlig måte til tjenesten. Man kan da velge lommeboken som alternativ og få en push-notifikasjon i appen der man må godta. Dette blir ganske likt BankID på mobil.

4.4.2 ID-porten som videreformidler av VP til tjeneste

I dette alternativet kunne det vært en løsning at ID-porten egentlig bare fungerer som en videreformidler av en VP for innlogging/verifisering. Dette vil foregå ved at en bruker velger lommeboken som valg i ID-porten og logger inn. ID-porten vil deretter videreformidle en VP som bruker ønsker å sende til tjenesten.

4.5 Utveksling

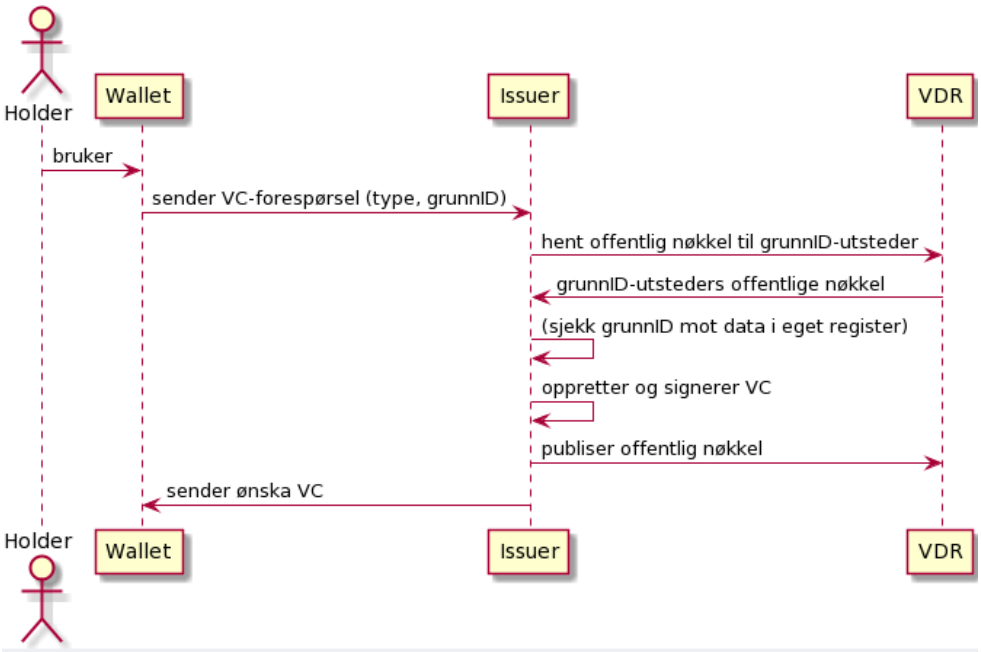
For å opprette forbindelse mellom de ulike aktørene i økosystemet vårt, er det viktig med tilgjengelige HTTP-kall. Vi har valgt å kode noen av utvekslingene i form av QR-koder for at brukeren skal skrive minst mulig, og dermed bidrar til en bedre brukeropplevelse.



Figur 4: Onboarding i applikasjonen

Ved onboarding må bruker hente en grunnidentitet fra en utsteder, som kan skje gjennom en innlogging i ID-porten, se Figur 4. Dette blir gjort som et HTTP-kall til utsteder sin server, i vårt tilfelle endepunktet «/protectedpage». Server og bruker gjør også et kall til VDR for nøkkellevering.

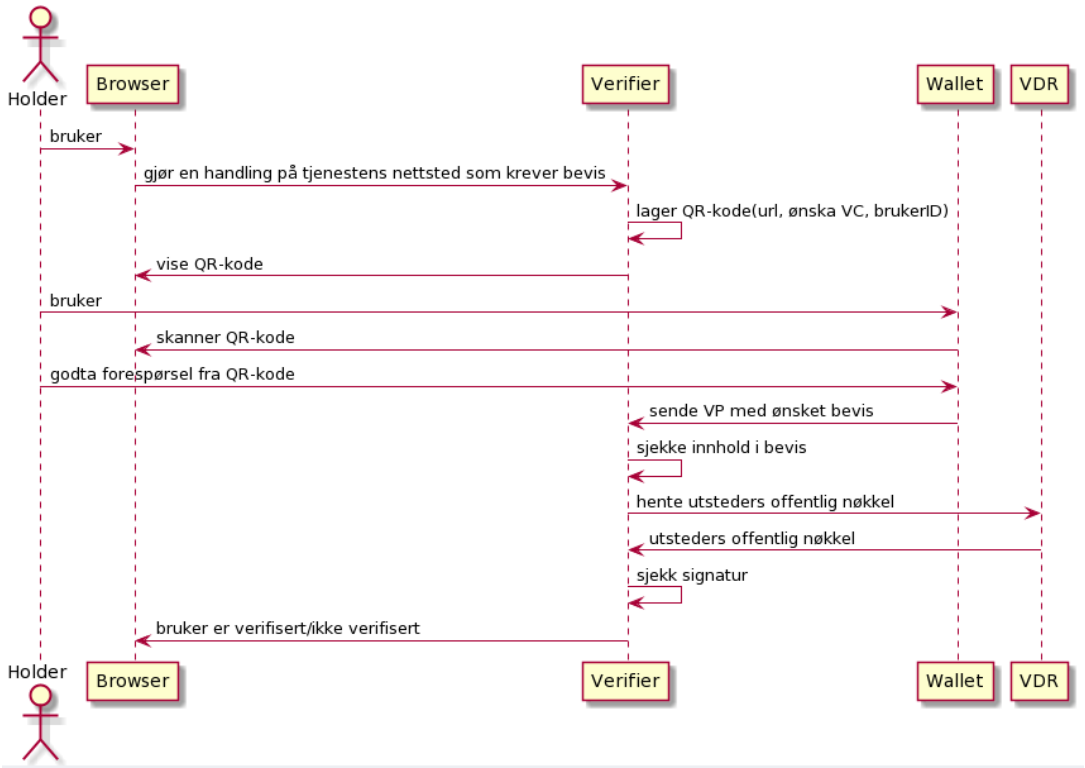
Ved henting av VC'er fra en utsteder vil bruker igjen gjøre et HTTP-kall til utsteders server, se Figur 5. Denne gangen sendes grunnidentitet, ønsket bevistype og ønsket utsteder i HTTP-URL'en til serveren. Serveren håndterer dette og dersom utstederen kan utstede det forespurte beviset, vil utsteder utstede dette beviset. Vi antar det er relativt lett å sjekke at brukeren faktisk har tilgang til ønsket VC, så vi har ikke implementert denne sjekken.



Figur 5: Hent bevis fra utsteder

Neste steg er muligheten for bruker å bevise noe hos en bestemt tjeneste. Dette kan gjøres gjennom en HTTP-forbindelse til tjenesten, eller en QR-kode som diskutert tidligere. Tjenesten kan, etter mottatt bevis fra bruker sitt HTTP-kall, finne nøkkelidentifikatorene brukt i forbindelse med beviset og spørre VDR om korresponderende offentlige nøkler.

I POC'en genererer tjenesten en QR-kode som inneholder hvilke bevis den krever og en URL som lommeboken skal sende disse bevisene til, se Figur 6. En annen mulighet kan også være at QR-koden inneholder en DID istedenfor en URL. I dette tilfelle må lommeboken «resolve» tjenesten sin URL via DID-metoden.



Figur 6: Verifisere bevis mot tjeneste

Som anbefaling til hva som kan brukes offline, ser vi muligheten til å bruke Bluetooth forbindelse mellom enheter som opprettes etter en skanning av en QR-kode. Altså en hybrid mellom HTTP og QR-kode.

4.6 Standard for VC

I forbindelse med vår Proof Of Concept har vi, med anbefalinger fra W3C, laget en standard for en VC (W3C, Verifiable Credentials Data Model 1.0, 2021). Se vedlegg 9.3 for et eksempel.

I header har vi:

kid – ID til nøkkel dersom en utsteder innehaver flere nøkler identifikatorer.

typ – Type token

alg – Signatur/krypteringsalgoritme

I payload har vi:

sub - Subject, hvem tokenet referer til. Her har vi foreløpig brukt fødselsnummer som en ID, da DID's ikke har blitt implementert i POC'en. I tillegg har vi tenkt at fødselsnummer er noe offentlige etater bruker mye, og kan derfor passe som ID til en bruker.

nbf – Not Before, ikke gyldig før dette tidspunktet.

iss – Issuer, hvem som lagde og signerte dette tokenet.

exp – Expiration time, når tokenets gyldighet utgår.

iat – Issued at, når tokenet ble utestedt.

vc – Verifiable credential, denne inneholder **credentialSubject**, **type** og **@context**.

> **credentialSubject** – Inneholder en eller flere påstander om Subject. Her kan det være en god idé å ha en standard på hvilke typer bevis en utsteder kan utstede. Disse kan for eksempel ligge i VDR'en.
>
> **type** – Hvilken type credential dette er. Om det er et studentbevis, førerkortinformasjon, helseforsikring osv. Dette feltet må være med for at beviset skal være verifiserbart.
>
> **@context** – Utdyper informasjon som gir kontekst til tokenet. @context linker til et dokument som forteller hvordan VC'en skal se ut på maskinlesbar standard. @context gjør det kompatibelt å bruke JSON-LD med JWT.

nonce – Tilfeldig unik verdi for tokenet.

jti – En hash av resten av «payload», slik sørges det for at jti alltid er unik, men rekalkulerbar.

Se Figur 7 under for et eksempel på en VC.

Figur 7: JWT-eksempel på VC

VC for grunnidentitet følger samme standard som bevisene ellers. Å ha en felles standard for alle bevis kan være en fordel, da en slipper forskjellige metoder for å lese VC'er.

5 SSI i praksis

5.1 Hva kan Digdirs og offentlig sektors rolle være i SSI?

Digdir kan ha flere roller, blant annet kan de ta ansvaret for å utforme et rammeverk for appen og/eller en bevisstandard som andre offentlige og private aktører kan ta i bruk. For å etterkomme forskriften til EU kan man også se for seg at Digdir kan lage en egen digital lommebok, som et offentlig alternativ. Videre vil offentlig sektor i stor grad ha ansvar for å utstede ulike bevis til brukerens lommebok, som ID, førerkort og utdanningsbevis.

5.1.1 Utstede grunnidentitet

Hver bruker må ha en grunnidentitet i lommeboken som vi ser for oss at benyttes for å forespørre nye bevis. Hvordan man utsteder grunnidentiteten kan gjøres på flere måter.

Digdir/ID-porten kan opprette grunnidentitet via onboarding, der innlogging med ID-porten skjer. Dette er en nyttig grunnidentitet til for eksempel offentlige tjenester. Vi har basert oss på denne typen grunnidentitet i vår prototype. I dette tilfelle må man unngå å sende inn grunnidentitet til private aktører som ikke trenger et like sikkert identitetsbevis. Dette kan løses ved å ikke bruke sensitiv data i grunnidentitetsbeviset. Det hadde vært mulig å heller bruke en lommebok-ID eller DID, fordi grunnidentiteten fortsatt hentes og signeres via ID-porten.

En annen løsning kan være at grunnidentiteten kun er basert på enheten og lommeboken, for så at bruker henter et bevis via ID-porten som for eksempel personnummer. Det kan da brukes mot andre offentlige tjenester som krever dette.

Et annet alternativ vil være å bruke DID som grunnidentitet, per enhet for eksempel. Det kan også opprettes en ny DID for hver tjeneste, slik at man unngår sporing.

5.1.2 Digdirs rolle som informasjonstilbyder

Digdir sin rolle kan også bare være overordnet. Da er tanken at de er en informasjonstilbyder. De kan sette krav for hva de mener en lommebok bør tilby, og deretter lage en oversikt over ulike tilbydere fra det private markedet som oppfyller disse kravene. Det er et stort marked for å lage lommebøker og dette kan føre til konkurranse i markedet og bedre tilbud for brukerne.

5.1.3 Digdir kan lage en referansearkitektur/implementasjon

En annen ide kan være at EU og/eller Digdir starter forarbeidet med å utvikle et forslag på arkitekturvalg eller andre tips som gjør det lettere for mindre aktører på det private markedet å starte prosessen med å lage lommebøker. Da utnytter man at Digdir har ressurser til å begynne prosessen, og det vil spare mange aktører for arbeid, som gjør at flere vil prøve å utvikle en lommebokløsning.

5.2 Hvorfor vil noen bruke dette?

En digital lommebok i et SSI-økosystem tilbyr brukeren sikker lagring og deling av personlig informasjon og gir mulighet for å trekke tilbake informasjon om seg selv. Ved at en slik applikasjon tillater gjenbruk av legitimasjoner, blir prosessen av å bevise noe om seg selv mye raskere, spesielt hvis en ønsker å gjenta en slik prosess hos flere tjenester (Northern Block , 2021).

Det vil være viktig at lommeboken er lett å ta i bruk. Dersom funksjonene i lommeboken blir gjort enkel og brukervennlig vil dette gjøre det attraktivt for brukere. Det at bevisene vil være digitale gjør også at man unngår å miste fysiske bevis, og man slipper å holde styr på flere ulike dokumenter ettersom alt er samlet på ett sted. Samtidig er alle bevis lett tilgjengelig for brukeren ettersom alt ligger på brukerens telefon. Siden alle bevis er standardisert, trenger ikke brukeren tenke på om applikasjonen de velger kan samhandle med en tjeneste eller ikke.

Lommeboken kan også benyttes som en innloggingsmekanisme hos ulike tjenester. Brukeren unngår da å måtte huske ulike brukernavn og passord til spesifikke tjenester. Det kan være en idé at brukere kan ha flere typer lommebøker, for eksempel en til offentlige tjenester og en til sosiale medier. Dette er blant annet grunnet befolkningens splittede tillitssyn på staten og private tjenester, og dermed er det ikke ønskelig at sikkerhet rundt disse tjenestene blandes.

Det er viktig å understreke at ikke hele den norske befolkningen ønsker å digitalisere livet sitt. Vi tror derfor det er viktig at de nåværende løsningene for disse brukergruppene fortsatt blir ivaretatt, og at den digitale lommeboken heller blir sett på som et nyttig verktøy.

5.3 Hvem tjener penger og hvordan?

Det finnes flere måter å tjene penger på i SSI-økosystemet. Det første alternativet kan være at man må betale for lommeboken enten som en årlig avgift eller en engangssum. En lommebok kan tilby spesifikke funksjoner som en bruker kan ønske å betale for. Dette kan for eksempel være en spesiell form for kryptering eller lagring. Et offentlig alternativ til lommeboken vil også kunne betales av staten. Man kan også se for seg at det kan være kostnadsbesparende for både tjenesten og utstederen at brukeren har en lommebok, og at disse aktørene derfor vil ha et ønske om å finansiere lommebøkene.

Man kan tenke seg at en utsteder kan kreve betaling for å utstede et bevis. Dette kan skje ved at utsteder krever en avgift fra lommebokselskapet per utstedelse som så legges til et tidsbestemt oppgjør som brukeren betaler. Dette kan sammenlignes med hvordan pass utstedes i dag, da man også må betale for dette.

Denne finansieringen kan altså ende opp med å være en delt kostnad, slik at flere eller samtlige aktører delfinansierer økosystemet.

5.4 Hvordan stole på at en lommebok er trygg?

For at en bruker skal stole på at en lommebok er trygg finnes det noen alternativer. Blant annet burde koden til lommeboken være Open Source. Det kan også være lurt å ha et organ som verifiserer ulike lommebøker på markedet, og sjekker om de oppfyller visse sikkerhetskrav. Dette organet burde ikke være staten eller EU, men heller være for eksempel en startup eller en annen bedrift som er spesialisert innenfor sikkerhet (Edwin, 2021). Det kunne vært en idé at to separate selskaper undersøker om en lommebok er trygg og oppfyller kravene, før den kan bli godkjent. Dersom lommeboken blir godkjent, kan for eksempel EU eller Digdir legge denne til i en liste over trygge lommebøker. Etter hvert vil også markedet bestemme, og lommebøker som benyttes av mange er de som vil overleve.

5.5 Tillit mellom utsteder, tjeneste og lommebok

Når det kommer til utstedere og tjenester, skal de i utgangspunktet ikke bry seg om hvilken lommebok brukerne benytter seg av. Dermed vil det ikke ha noe å si for dem om lommeboken er sikker eller ikke. Dette er fordi det er brukerens identitet, og dermed også brukerens ansvar å være trygg på løsningen de ønsker å benytte. Det som skal bety noe for utsteder og tjeneste er hvem brukeren er, og deres tilhørende nøkkel(er).

Hvordan ha tillit til utsteders utstedte bevis? Nøkler kan som sagt lagres i tillitsrammeverket og brukes til å verifisere signaturer. Dersom en stoler på at tillitsrammeverket kun tillater legitime utstedere å laste opp nøkkelen deres, er dette en metode for å finne ut hvilke utstedere man kan stole på. På denne måten hindrer man å godta falske bevis fra uautoriserte enkeltpersoner og illegitime bedrifter.

5.6 Hvordan trekke tilbake informasjon som allerede er delt

I noen situasjoner ønsker en tjeneste kun å verifisere informasjon, men ikke lagre opplysningene. I andre situasjoner lagrer tjenesten opplysninger over tid, og da skal brukeren kunne trekke tilbake godkjenningen sin. Appen bør sende et varsel til tjenesten om at brukeren ønsker å trekke tilbake sin informasjon hen tidligere delte. Deretter må tjenesten slette informasjonen, i henhold til Personvernforordningen, også kalt GDPR. Datatilsynet kontrollerer at tjenester overholder lovverket.

5.7 Forhindre misbruk av identitet

For at et digitalt legitimasjonsbevis ikke skal bli misbrukt i den analoge verden bør det følge med et bilde av personen beviset tilhører. Det gjør det vanskeligere å stjele identiteten til noen andre. Ta eksempelet der en person bruker appen til å bevise at hen er over 18 år i en butikk. En ansatt kan da verifisere at bildet, og QR-koden med beviset, tilhører en person på lik linje som et legitimasjonskort i dag. Fremover kan det være aktuelt å utforske

ansiktsgjenkjenning mellom bildet i appen og personen som ønsker å bruke beviset i f.eks. en selvbetjent butikk. Bilde kan for eksempel bli hentet fra passregisteret i løpet av registreringsprosessen.

I tillegg bør man hindre at samme bevis er lagret i flere apper eller på flere enheter. Her er det hovedsakelig to løsninger. Den første løsningen går ut på at dersom det samme beviset blir forespurt i en annen lommebok vil beviset bli slettet i den lommeboken det allerede ligger i. Den andre løsningen er å kun tillate å være registrert på én enhet samtidig. Når en person allerede er registrert i en app på en enhet, og prøver å registrere seg på en annen enhet, får ikke brukeren opprette den nye brukeren. På denne måten finnes det alltid kun ett gyldig eksemplar av et bevis.

5.8 Tillit

I SSI-økosystemet er tillit et gjennomgående og sentralt begrep.

Tillittsflyten er som følgende:

- Alle aktører stoler på at det skal være åpenbart dersom VDR manipuleres.
- Tjenesten stoler på at utstedere utstedte beviset det mottar. Dette oppnås gjennom en signatur.
- Bruker og tjeneste stoler på at utsteder skal utstede korrekte bevis.
- Bruker stoler på at lommeboken lagrer informasjonen sin sikkert.

Dette skiller seg fra vanlige tillittsmodeller:

- Utsteder og tjeneste trenger ikke å stole på lommeboken.
- Utsteder trenger ikke å stole på, og heller ikke kjenne tjenesten.

(W3C, Verifiable Credentials Data Model - 5.2 Trust Model, 2021)

5.9 Modenhet – er tiden for lommebøker og VC inne?

Vi mener at mye som skal til for å implementere et SSI-økosystem i dag, er tilgjengelig. Etter at EU bestemte seg for at det skal innføres, har temaet fått mer fokus. Det er i tillegg mange organisasjoner, som blant annet Digital Identitet Norden (DIN), som jobber for at dette skal være mulig (DIN, 2021). Verden har blitt såpass digital at det å kunne identifisere og verifisere sin egen identitet også burde kunne digitaliseres.

Det er noen deler som ikke er helt modne enda, som for eksempel tilgjengelige bibliotek for LD-Signature. LD-Signature er derimot ikke et krav for SSI. Her finnes det flere alternativ, som blant annet JWT. Vi forventer at et sikrere og produksjonsklart bibliotek for LD-Signature kommer til å bli opprettet når flere tar det i bruk.

Andre konsept som ikke er helt modent enda er standardiseringene rundt SSI og om for eksempel nullkunnskapsbevis (ZKP) skal benyttes. Det er fordeler med å bruke ZKP, men det kan også kreve mer prosessorkraft fra mobiler. Blokkjeder er også en av standardiseringene som vi ikke vet om er moden nok, fordi det er en relativt ny teknologi. Vi spurte Jon Ramvi, fra Symfoni AS, om modenheten til Ethereum. Han mener Ethereum er i «improving»-fasen. Det gjenstår fortsatt å ferdigstille layer 2 løsninger for skalerbarhet, overgang til Proof of Stake og ytterligere skalerbarhet med sharding (Ethereum, What is sharding?, 2021). Sistnevnte vil komme med Eth 2.0 som trolig ikke vil være klart før 2023.

Analysebyrået Gartner sier følgende om modenheten til blokkjedeteknologi: "Vi er vitne til mye utvikling i blokkjedeteknologi som vil endre dagens mønster. Innen 2023 vil blokkjedeplattformer være skalerbare, interoperabelt og vil kunne støtte smart kontrakt bærbarhet og 'cross chain' funksjonalitet. De vil også støttet tillitbaserte private transaksjoner med datakonfidensialiteten som er nødvendig. Disse teknologiske fremskrittene vil ta oss mye nærmere mainstream blokkjede og et desentralisert nett, også kjent som Web 3.0" (Litan, 2021). Det er likevel flere nasjoner som tidlig har valgt å ta i bruk teknologien i både små og store systemer, blant annet Storbritannia, Chile og Sveits (NEWS, 2021). I løpet av august planlegger også Brønnøysundregisteret å lansere sin aksjeeierbokplattform på enten Oprimism[3] eller Arbitrum[4], som begge er Ethereum layer 2 løsninger.

Dersom det lovlige rammeverket blir laget, slik at tillit er mulig å etableres, er det dog ingen god grunn til å ikke utvikle SSI. Bruk av litt enklere løsninger som JWT i stedet for JSON-LD kan fungere som en bro fra dagens til morgendagens system.

[3]https://optimism.io
[4] https://offchainlabs.com

6 Konklusjon

I denne rapporten har vi presentert våre funn i forbindelse med vår Proof Of Concept. Videre har vi utforsket SSI-økosystemet, der vi har sett på alternativer til implementasjon av SSI. Til slutt har vi sett på rollen Digdir kan ha i dette økosystemet og diskutert noen viktige problemstillinger rundt implementasjonen av SSI.

Vi anbefaler at utstedere utsteder så små VC'er som mulig. Optimalt sett ser vi at en VC kun inneholder nødvendig og tilstrekkelig informasjon som sikrer et gyldig legitimasjonsbevis. Et eksempel er et førerkort som kun viser hva slags type kjøretøy sjåføren har lov til å kjøre, og ikke tilleggsinformasjon slik som navn eller fødselsnummer. Dette vil gjøre det enklere for brukeren å kun dele ønskede informasjonsbiter om seg selv. Dersom tjenesten trenger mer informasjon vil det være mer hensiktsmessig å sette sammen flere små VC'er til en VP.

Digdir kunne ha tatt på seg utvikling av lommebøker, men dette er ikke nødvendigvis noe de må gjøre. For å kunne effektivisere et økosystem, kan det være viktig å ta med private aktører. Vi tenker at der det er mest penger å tjene for en privat aktør ved utvikling av en lommebokapplikasjon. En lommebok kan tilby spesifikke og ulike funksjoner som en bruker kan ønske å betale for. Konkurranse i markedet kan også være med på å øke kvaliteten av lommebøker.

Vi mener at Digdir burde ta på seg en rolle som informasjonstilbyder og å se på mulighetene for å lage en referansearkitektur. Som informasjonstilbyder kan Digdir, sammen med EU, sette standarder for hva som er en god lommebok. De kan da anbefale lommebøker som er innenfor disse standardene. En referansearkitektur kan gjøre det enklere for bedrifter å ta del i dette markedet.

For å oppnå høy bruk av lommebøker, er det viktig at appen er enkel og brukervennlig. En lommebok burde gjøre prosessen for å bevise noe enklere og det burde være en standard for hvordan man beviser noe. Dette for at en bruker skal slippe å måtte laste ned ulike lommebøker for forskjellige tjenester eller utstedere. I tillegg er det viktig at lommeboken er sikker, og at en bruker stoler på lommeboken.

Å desentralisere informasjon vil trolig bli viktigere framover. Her kan blokkjedeteknologi være nyttig. Vi vil spesielt anbefale å se på Ethereum, da de tilbyr en rekke funksjonaliteter for desentralisering av identitet. Det er også dette mange eksperter anbefaler. I første omgang kan man bruke Ethereum-addresser for å opprette DID til utstedere og lagre de på et register på blokkjeden.

7 Refleksjon

SSI er et økosystem som baserer seg på kompliserte konsepter. Noe av det vanskeligste har vært å forstå relevante begreper og oppnå oversikt/skille mellom det som er mulig og det som er umulig å implementere.

Selve utviklingen er både komplisert og enkel på samme tid. Årsaken er at SSI i seg selv ikke er komplisert, men å skalere det opp til et høyt sikkerhetsnivå tar tid og kan være krevende. Serversidene utsteder, tjeneste og VDR har vært det letteste, fordi det finnes god dokumentasjon på implementasjon av disse. Vi mener lommeboken er den viktigste delen og også den viktigste å kode sikkert, da det ikke eksisterer arkitektur på samme måte som det gjør for blant annet utsteder. For utsteder kan det brukes kraftige datamaskiner og databaser som allerede er etablert.

Det er viktig å sørge for at lommeboken er trygg ved å lagre sensitiv data på en sikker måte. I tillegg må lommeboken kunne støtte flere måter å kommunisere på. Ved lokal lagring oppstår det problemer dersom du mister mobilen. Her kan det være en løsning å bruke en tilbaketrekkingsliste, slik at bevis kan trekkes tilbake om noen andre har fått tilgang til lommeboken din.

Det er også viktig å nevne at det er vanskelig å finne gode kilder som forklarer ferske konsepter som blokkjeder i SSI sammenheng, DIDs og nullkunnskapsbevis på en god måte. En rekke av slike begreper, og et ønske om å se hvor vanskelig det ville vært å implementere disse, har gjort arbeidet vårt teoretisk, vanskelig, lærerikt og spennende.

8 Referanser

Abraham, A., More, S., Rabensteiner, C., & Hörandner, F. (2021, Februar 9). *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. doi:https://doi.org/10.1109/TrustCom50675.2020.00136

Affinidi. (2021, Juli 29). *What is a Verifiable Credential and Verifiable Claim?* Hentet fra Affinidi: https://docs.affinidi.com/intro/#what-is-a-verifiable-credential-and-verifiable-claim

Allen, C. (2016, April 25). *The Path to Self-Sovereign Identity*. Hentet fra Life with Altracity: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html

Bhushan, V. A. (2021, Juli 22). *Is Blockchain an important tool for SSI*. Hentet fra Imaginea: https://labs.imaginea.com/is-blockchain-an-important-tool-for-ssi/

CAS. (2021, Juli 26). *Skybasert eller lokal lagring?* Hentet Juli 26, 2021 fra https://www.cas.no/nyheter/skybasert-eller-lokal-lagring

Chadwick, D. (2021, Juli 20). *Integrating W3C Web Authentication (FIDO2) and Verifiable Credentials*. Hentet fra Youtube: https://www.youtube.com/watch?v=62IYP1XtTYU

CISA. (2021, Juli 21). *Security Tip (ST04-018)*. Hentet fra https://us-cert.cisa.gov/ncas/tips/ST04-018

DIN. (2021, August 02). *DIN - Framside*. Hentet fra DIN: https://www.din.foundation/

Duffy, K. H. (2020). *Applying Self-Sovereign Identity Principles to Interoperable Learning Records.* U.S. Chamber of Commerce Foundation.

DWH. (2021, April 11). *Don't Use DIDs*. Hentet fra dwhuseby.medium.com: https://dwhuseby.medium.com/dont-use-dids-58759823378c

Edwin, S. L. (2021, Juli 27).

Ethereum. (2021, August 3). *Ethereum*. Hentet fra Lær mer om Ethereum : https://ethereum.org/nb/learn/

Ethereum. (2021, August 04). *Proof-of-stake (POS)*. Hentet fra https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

Ethereum. (2021, August 2). *What is sharding?* Hentet fra Shard chains: https://ethereum.org/en/eth2/shard-chains/

EU. (2021, Juli 22). *Blockchain and Digital Identity.* Hentet fra eublockchainforum: https://www.eublockchainforum.eu/sites/default/files/report_identity_v0.9.4.pdf

EU. (2021, Juli 20). *Commission proposes a trusted and secure Digital Identity for all Europeans*. Hentet fra europa.eu: https://digital-strategy.ec.europa.eu/en/news/commission-proposes-trusted-and-secure-digital-identity-all-europeans

EU. (2021, Juli 22). *EBSI*. Hentet fra What is EBSI: https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/EBSI

EU. (2021, Juli 20). *Questions and Answers*. Hentet fra https://ec.europa.eu/commission/presscorner/detail/en/QANDA_21_2664

Europakommisjonen. (2021, Juli 28). *European Digital Identity*. Hentet fra https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-digital-identity_en

Evernym. (2021, Juli 22). *Gitlab*. Hentet fra Evernym Mobile SDK: https://gitlab.com/evernym/mobile/mobile-sdk/-/tree/main

FN. (2021, Juli 28). *FN*. Hentet fra FN - Bærekraftmål: https://www.fn.no/om-fn/fns-baerekraftsmaal

Forbrukerrådet. (2021, Juli 21). *New analysis shows how Facebook and Google push users into sharing personal data*. Hentet fra https://www.forbrukerradet.no/side/facebook-and-google-manipulate-users-into-sharing-personal-data/

Gisle, J. (2018, November 30). Personvernforordningen. *Store Norske Leksikon*. Hentet fra https://snl.no/Personvernforordningen

Harris, O. (2021, Juli 28). *Security Issues in JWT Authentication*. Hentet fra Software Secured: https://www.softwaresecured.com/security-issues-jwt-authentication/

Javatpoint. (2021, August 4). *Blockchain Tutorial*. Hentet fra Blockchain Proof of work: https://www.javatpoint.com/blockchain-proof-of-work

Johnson, A. (2021, Juli 26). *Trinsic Basics: What Are SSI Digital Wallets?* Hentet fra https://trinsic.id/what-are-ssi-digital-wallets/

Kalveram, M. (2021, Juli 20). *A Bullet-Proof Approach to Storing Sensitive User Data in React Native*. Hentet fra https://medium.com/swlh/a-bullet-proof-approach-to-storing-sensitive-user-data-in-react-native-ab3f7a2779f9

Litan, A. (2021, August 2). *Gartner 2019 Hype Cycle Shows Most Blockchain Technologies Are Still Five to 10 Years Away From Transformational Impact*. Hentet fra Gartner: https://www.gartner.com/en/newsroom/press-releases/2019-10-08-gartner-2019-hype-cycle-shows-most-blockchain-technologies-are-still-five-to-10-years-away-from-transformational-impact

Majaski, C. (2021, Juli 27). *Distributed Ledger*. Hentet fra https://www.investopedia.com/terms/d/distributed-ledgers.asp

MATTR. (2021, Juli 27). *Using privacy-preserving ZKP credentials on the MATTR Platform*. Hentet fra MATTR: https://mattr.global/using-privacy-preserving-zkp-credentials-on-the-mattr-platform/

Neira, B. (2021, Juli 26). *Microsoft Docs - Azure*. Hentet fra Tutorial - Get started with Azure Active Directory Verifiable Credentials using a sample app (preview): https://docs.microsoft.com/en-us/azure/active-directory/verifiable-credentials/get-started-verifiable-credentials

NEWS, I. T. (2021, August 2). *irishtechnews.ie*. Hentet fra GLOBAL BLOCKCHAIN ADOPTION: WHICH COUNTRIES ARE LEADING THE CHARGE?: https://irishtechnews.ie/global-blockchain-adoption-which-countries-are-leading-the-charge/

Nist. (u.d.).

NIST. (2021, August 4). *A Taxonomic Approach to Understanding* . Hentet fra Credential Architectures: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.01142020.pdf

NIST. (2021, Juli 22). *Blockchain Overview*. Hentet fra Blockchain Technology Overview: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

Northern Block . (2021, Juli 27). *Trust, Privacy and Verifiability - Your Way*. Hentet fra https://northernblock.io/products/ssi-digital-wallet/

Ramvi, J. (2021, Juli).

Reed, D., & Preukschat, A. (2021, Juli 27). *The Basic Building Blocks of SSI*. Hentet fra https://freecontent.manning.com/the-basic-building-blocks-of-ssi/

Russell, A. (2021, Juli 19). *Hva er et X.509-sertifikat?* Hentet fra SSL.com: https://www.ssl.com/no/Vanlige-sp%C3%B8rsm%C3%A5l/hva-er-et-x-509-sertifikat/

Senel, A. (2021, Juli 26). *Visolit*. Hentet fra Skytjenester og skybasert lagring - Hva er det?: https://www.visolit.no/artikler/fordeler-og-ulemper-med-skytjenester

Sovrin. (2021, Juli 26). *Sovrin Glossary V2, Appendix G*. Hentet fra https://sovrin.org/wp-content/uploads/Sovrin-Glossary-V2.pdf

Sporny, M. (2021, Juli 21). *Youtube*. Hentet fra Linked Data Signatures: https://www.youtube.com/watch?v=QdUZaYeQblY

Transmute. (2021, Juli 28). *Encrypted Data Vaults for Trusted Data Access*. Hentet fra https://medium.com/transmute-techtalk/encrypted-data-vaults-c794055b170e

Trinsic. (2021, Juli 22). *Trinsic*. Hentet fra Docs: https://docs.trinsic.id/docs

TrustNetPK. (2021, Juli 22). *Github*. Hentet fra TrustNetPK / cov-id-wallet: https://github.com/TrustNetPK/cov-id-wallet

Tykn. (2021, Juli 21). *Self-Sovereign Identity: The Ultimate Beginners Guide!* Hentet fra Characteristics of Self-Sovereign Identity: https://tykn.tech/self-sovereign-identity/

W3C. (2021, Juli 21). *Decentralized Identifiers (DIDs) v1.0*. Hentet fra https://www.w3.org/TR/did-core/

W3C. (2021, Juli 28). *Encrypted Data Vaults 0.1*. Hentet fra https://digitalbazaar.github.io/encrypted-data-vaults/

W3C. (2021, Juli 20). *Verifiable Credentials Data Model - 3.2 Credentails*. Hentet fra https://www.w3.org/TR/vc-data-model/#credentials

W3C. (2021, 7 29). *Verifiable Credentials Data Model - 3.3 Presentations*. Hentet fra https://www.w3.org/TR/vc-data-model/#presentations

W3C. (2021, 7 30). *Verifiable Credentials Data Model - 5.2 Trust Model*. Hentet fra https://www.w3.org/TR/vc-data-model/#trust-model

W3C. (2021, Juli 27). *Verifiable Credentials Data Model 1.0*. Hentet fra https://www.w3.org/TR/vc-data-model/

W3C. (2021, Juli 21). *Verifiable Credentials Data Model 1.0 - 6.2 JSON-LD*. Hentet fra https://www.w3.org/TR/vc-data-model/#json-ld

W3C. (2021, 07 19). *Verifiable Credentials Data Model 1.0 - 6.3.1 JSON Web Token*. Hentet fra JSON Web Token: https://www.w3.org/TR/vc-data-model/#json-web-token

W3C. (2021, Juli 29). *Verifiable Credentials Implementation Guidelines 1.0 - 8 Using the JWT aud claim*. Hentet fra W3C: https://www.w3.org/TR/vc-imp-guide/#using-the-jwt-aud-claim

W3C Community Group. (2020, Juli 21). *JSON-LD 1.1*. Hentet fra JSON-LD 1.1: https://json-ld.org/spec/latest/json-ld/

Wikipedia. (2021, Juli 26). *Wikipedia*. Hentet fra Zero knowledge proof: https://en.wikipedia.org/wiki/Zero-knowledge_proof

Zhang, J., Xie, T., Zhang, Y., & Song, D. (2020, Juli 30). *2020 IEEE Symposium on Security and Privacy.* doi:10.1109/SP40000.2020.00052

9 Vedlegg

9.1 Fordeler med JWT

Feature	JSON + JWTs	JSON-LD + JWTs	JSON-LD + LD-Proofs
PF1a. Proof format supports Zero-Knowledge Proofs.	✓	✓	✓
PF2a. Proof format supports arbitrary proofs such as Proof of Work, Timestamp Proofs, and Proof of Stake.	✓	✓	✓
PF3a. Based on existing official standards.	✓	✕	✕
PF4a. Designed to be small in size.	✓	✕	✕
PF5a. Offline support without further processing.	✓	✕	✕
PF6a. Wide adoption in other existing standards.	✓	✓	✕
PF7a. No type ambiguity.	✓	✕	✕
PF8a. Broad library support.	✓	✕	✕
PF9a. Easy to understand what is signed.	✓	✓	✕
PF10a. Ability to be used as authn/authz token with existing systems.	✓	✓	✕
PF11a. No additional canonicalization required.	✓	✕	✕
PF12a. No Internet PKI required.	✓	✕	✕
PF13a. No resolution of external documents needed.	✓	✕	✕

Figur 8: https://w3c.github.io/vc-imp-guide/#benefits-of-jwts

9.2 Fordeler med JSON-LD + LD-Signature (LD-Proofs)

Feature	JSON + JWTs	JSON-LD + JWTs	JSON-LD + LD-Proofs
PF1b. Support for open world data modelling.	✗	✓	✓
PF2b. Universal identifier mechanism for JSON objects via the use of URIs.	✗	✓	✓
PF3b. A way to disambiguate properties shared among different JSON documents by mapping them to IRIs via a context.	✗	✓	✓
PF4b. A mechanism to refer to data in an external document, where the data may be merged with the local document without a merge conflict in semantics or structure.	✗	✓	✓
PF5b. The ability to annotate strings with their language.	✗	✓	✓
PF6b. A way to associate arbitrary datatypes, such as dates and times, with arbitrary property values.	✗	✓	✓
PF7b. A facility to express one or more directed graphs, such as a social network, in a single document.	✗	✓	✓
PF8b. Supports signature sets.	✗	✗	✓
PF9b. Embeddable in HTML such that search crawlers will index the machine-readable content.	✗	✗	✓
PF10b. Data on the wire is easy to debug and serialize to database systems.	✗	✗	✓
PF11b. Nesting signed data does not cause data size to double for every embedding.	✗	✗	✓
PF12b. Proof format supports Zero-Knowledge Proofs.	✗	✗	✓
PF13b. Proof format supports arbitrary proofs such as Proof of Work, Timestamp Proofs, and Proof of Stake.	✗	✗	✓
PF14b. Proofs can be expressed unmodified in other data syntaxes such as YAML, N-Quads, and CBOR.	✗	✗	✓
PF15b. Changing property-value ordering, or introducing whitespace does not invalidate signature.	✗	✗	✓
PF16b. Designed to easily support experimental signature systems.	✗	✗	✓
PF17b. Supports signature chaining.	✗	✗	✓
PF18b. Does not require pre-processing or post-processing.	✗	✗	✓
PF19b. Canonicalization requires only base-64 encoding.	✗	✗	✓

Figur 9: https://w3c.github.io/vc-imp-guide/#benefits-of-json-ld-and-ld-proofs

9.3 JWT-Standard – Eksempel på bevis (Alder over 18)

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "kid": "keyNumber x",
  "typ": "JWT",
  "alg": "RS256"
}
```


PAYLOAD: DATA


```
{
  "sub": "08089409382",
  "nbf": 1627893505,
  "iss": "folkeregisteret4283afdc-9f54-45b7-a7e2-
5ee965b18f9c",
  "exp": 1629103105,
  "iat": 1627893505,
  "vc": {
    "credentialSubject": {
      "age": {
        "name": "Over 18",
        "type": "over-18"
      }
    },
    "type": [
      "VerifiableCredential",
      "AgeCredential"
    ],
    "@context": [
      "https://www.w3.org/2018/credentials/v1"
    ]
  },
  "nonce": "b433ab30-0977-450",
  "jti": "WMuTzBzx401m24pHWZUsna9Ov531FcArrGpO6o/ZN+Y="
}
```


VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  Public Key or Certificate. Ent
  er it in plain text only if yo
  u want to verify a token
  ,
  Private Key. Enter it in plain
  text only if you want to gener
  ate a new token. The key never
  leaves your browser.
)
```

9.4 JWT-Standard – Eksempel på Grunn-ID

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "kid": "keyNumber x",
  "typ": "JWT",
  "alg": "RS256"
}
```


PAYLOAD: DATA


```
{
  "sub": "08089409382",
  "nbf": 1627893453,
  "iss": "GrunnID-portalen.no62cba64b-0a3c-4538-b138-
5ac7a24e873f",
  "exp": 1629103051,
  "iat": 1627893451,
  "vc": {
    "credentialSubject": {
      "baseid": {
        "name": "BaseID",
        "type": "BaseID"
      }
    },
    "type": [
      "VerifiableCredential",
      "BaseCredential"
    ],
    "@context": [
      "https://www.w3.org/2018/credentials/v1"
    ]
  },
  "nonce": "4c628b16-b3a3-440",
  "jti": "tFsKk9fdcJL0QpDgUa2bcNJc1jX82mCNG3hcxCrS8qw="
}
```


VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  Public Key or Certificate. Ent
  er it in plain text only if yo
  u want to verify a token
  ,
  Private Key. Enter it in plain
  text only if you want to gener
  ate a new token. The key never
  leaves your browser.
)
```