Norges Bank

3 | 2026

Norges Bank Memo

Digitale sentralbankpenger –
eksperimentell testing i prosjektfase 5

Sammendrag

I henhold til Strategi 25 skulle Norges Bank «*... gjøre oss klare til eventuelt å kunne innføre digitale sentralbankpenger. Digitale sentralbankpenger kan bli nødvendig for at det også i fremtiden skal oppleves som sikkert, effektivt og attraktivt å betale med norske kroner.*» (Norges Bank, 2022, s.10)

Som en oppfølging av dette strategipunktet etablerte Norges Bank et prosjekt som skulle utrede behovet for innføring av en digital sentralbankpenge (DSP[1]) og i tilfellet hvordan en DSP-løsning burde utformes.[2] Prosjektet tok utgangspunkt i at løsningen skulle baseres på blokkjedeteknologi[3] slik at gevinstene ved denne teknologien kunne utnyttes. Som en del av utredningen gjennomførte prosjektet teknisk og funksjonell testing i to teknologiske sandkasser. Målet med testingen var å få innsikt i hvordan blokkjedeteknologien kan brukes til å fremme et sikkert, effektivt og attraktivt betalingssystem, og å belyse hvilke gevinster, utfordringer og ulemper som følger av ulike tekniske alternativer.

De to sandkassene er basert på to ulike ledger[4]-plattformer og de er koblet sammen gjennom en bro som muliggjør overføring av tokens[5] (DSP, tokeniserte bankinnskudd (TBI) og tokeniserte verdipapirer) mellom dem. Testprogrammet i denne utredningsfasen omfatter ulike former for betalinger med kunderettet DSP mellom privatpersoner, bruk av DSP for oppgjør av betalinger mellom banker, betalinger med tokeniserte bankinnskudd med sentralbankoppgjør i tokeniserte reserver, og anvendelse av ledgerteknologien for å automatisere og effektivisere statsgjeldsforvaltningen.

Testcasene ble utformet med utgangspunkt i sentrale prosesser i betalings- og finanssystemet, men gjennomført på en forenklet og tilpasset måte for å teste egenskaper, gevinster og utfordringer knyttet til ulike alternative teknologiske design. Hverken kapasitetstester, stresstester, eller teknologiens evne til å håndtere unormale hendelser og angrep fra ondsinnede aktører inngikk i testprogrammet.

Hovedfunnene kan oppsummeres i tre punkter. For det første gir en single-ledger[6]-arkitektur, der DSP, TBI og tokeniserte verdipapirer er registrert på samme ledger, fordeler.

---

[1] Digitale sentralbankpenger (DSP) *er en digital form for sentralbankpenger utstedt av sentralbanken, og representerer et direkte krav på sentralbanken*.

[2] Se Norges Bank (2026).

[3] Blokkjedeteknologi: En type distribuert hovedbok (DLT) der transaksjoner lagres i «blokker» som kryptografisk lenkes sammen i en kjede, og oppdateres/synkroniseres mellom flere deltakere uten én felles sentral database. «En blokkjede er en desentralisert og distribuert digital «regnskapsbok» som gjør det mulig å registrere, spore og synliggjøre alle digitale transaksjoner» (Store norske leksikon, 2026).

[4] Ledger brukes i rapporten som et samlebegrep for et delt, distribuert og kryptografisk sikret register (DLT), og omfatter både blokkjeder og andre varianter av distribuert hovedbok/register.

[5] Token/tokenisering: Tokenisering innebærer å representere penger, verdipapirer eller andre rettigheter som digitale tokens på en *ledger*, slik at eierskap og overføringer kan håndteres programmessig.

[6] Single-ledger: En arkitektur der alle digitale aktiva er tokenisert og håndteres på én og samme DLT/ledger. Overføringer og oppgjør skjer innenfor samme ledger, uten behov for flere DLT-er eller broer mellom dem.

De kommer i form av at arkitekturen muliggjør atomisk[7] oppgjør på tvers av aktiva, eliminering av motpartsrisiko, høy grad av sporbarhet og enklere teknisk etterlevelse av enkelte regulatoriske krav. For det andre gir multi-ledger[8]-arkitekturer (der ulike aktiva som DSP, TBI og verdipapirer er registrert på ulike ledgere og flyttes mellom dem ved bruk av broer[9]) økt teknisk og operasjonell kompleksitet og større angrepsoverflate. Delivery versus Payment (DvP)[10] kan oppnås, men når oppgjøret av verdier gjennomføres på ulike ledgere, skjer det ikke samtidig. Det innebærer at man ikke oppnår atomiske oppgjør ved et slikt design. Dette øker behovet for styring, overvåking og standardisering. For det tredje fremgår det at de viktigste gevinstene ved DSP og ledger-teknologi først realiseres dersom andre aktører enn sentralbanken selv kan registrere egne aktiva på ledgeren og utvikle løsninger som bruker DSP som oppgjørsmiddel.

Denne rapporten dokumenterer det tekniske designet, gjennomføringen og funnene fra testingen i sandkassene. Rapporten er en teknisk utdyping og underlagsrapport til prosjektets sluttrapport[11], og inneholder ikke anbefalinger om innføring av DSP.

Kildekoden som er grunnlaget for testsystemene i sandkassene blir lagt ut som *open source* og kan finnes her i vårt github repo [12].

[7] Atomisitet innebærer at en sammensatt transaksjon enten gjennomføres i sin helhet eller ikke gjennomføres i det hele tatt. Alle relevante tilstandsendringer, for eksempel overføring av penger og verdipapirer, behandles som én udelelig operasjon. Det oppstår ingen delvise eller mellomliggende tilstander der bare deler av transaksjonen er gjennomført. Atomisitet eliminerer dermed risiko for at én part leverer uten at motytelsen samtidig fullføres.

[8] Multi-ledger: En multi-ledger-arkitektur innebærer at ulike aktiva eller aktører benytter separate ledgere, og at oppgjør mellom dem krever koordinering eller broer.

[9] Broer: Teknisk mekanisme som kobler to (eller flere) separate DLT-er slik at tokens/verdier og relevante meldinger kan overføres mellom dem, typisk ved låsing/brenning på én ledger og utstedelse/utbetaling på en annen.

[10] Delivery versus Payment (DvP) er et oppgjørsprinsipp der levering av et aktivum og betaling skjer samtidig og betinget av hverandre, slik at ingen av partene eksponeres for motpartsrisiko i oppgjøret. I tradisjonelle systemer oppnås DvP gjennom koordinering mellom separate verdipapir- og betalingssystemer. I DLT-baserte løsninger kan DvP enten implementeres som én teknisk atomisk transaksjon når penger og aktiva er representert på samme ledger (single-ledger), eller som en orkestrert prosess på tvers av flere ledgere, der samtidighet og endelighet ikke oppnås gjennom én enkelt teknisk operasjon.

[11] Se Norges Bank (2026).

[12] Se Github (2026).

Innhold

Innledning 5

Testcase 1 – Etablering av en ledger for kunderettede DSP og utforskning av egenskapene ved en slik ledger 8

Formålet med testcaset 8

Teknisk design 8

Resultater og funn 11

Testcase 2 – Etablering av en ledger for DSP for oppgjør og egenskapene ved en slik ledger 12

Formålet med testcaset 12

Teknisk design 12

Resultater og funn 15

Testcase 3 – Etablering av en bro for overføring av tokens/verdier mellom ledgere og egenskaper ved en slik bro 16

Formålet med testcaset 16

Teknisk design 17

Resultater og funn 20

Testcase 4 – Egenskaper ved tokeniserte bankinnskudd i en multi-ledger arkitektur 21

Formålet med testcaset 21

Teknisk design 22

Resultater og funn 28

Testcase 5 – Oppgjør av tokeniserte verdipapirer på en felles ledger 29

Formålet med testcaset 29

Teknisk design 30

Resultater og funn 34

Testcase 6 – Utstedelse av statsobligasjoner på en felles ledger 35

Formålet med testcaset 35

Teknisk design 36

Resultater og funn 41

Oppsummering av funn 42

Publisering av kildekode 43

Referanser 44

Innledning

I henhold til Strategi 25 skulle Norges Bank «... *gjøre oss klare til eventuelt å kunne innføre digitale sentralbankpenger. Digitale sentralbankpenger kan bli nødvendig for at det også i fremtiden skal oppleves som sikkert, effektivt og attraktivt å betale med norske kroner.*» (Norges Bank, 2022, s.10)

Som en oppfølging av dette strategipunktet etablerte Norges Bank et prosjekt som skulle utrede behovet for innføring av en digital sentralbankpenge (DSP) og i tilfellet hvordan en løsning for en slik DSP burde utformes.[13] Prosjektet tok utgangspunkt i at en DSP-løsning skulle baseres på blokkjedeteknologi slik at gevinstene ved denne teknologien kunne utnyttes. Som en del av utredningen gjennomførte prosjektet teknisk og funksjonell testing av anvendelse av teknologien til betalings- og andre finansielle tjenester i to teknologiske sandkasser.[14]

Denne rapporten beskriver de teknologiske valgene som ligger til grunn for etablering av sandkassene. Videre redegjøres for hvordan sandkassene har blitt etablert og hvordan den tekniske og funksjonelle testingen har vært gjennomført. Til slutt oppsummeres hovedfunnene fra testingen.

Det overordnede målet med den tekniske testingen har vært å få informasjon som er relevant for vurderingen av om innføring av DSP basert på blokkjedeteknologi vil bidra til et effektivt og sikkert betalingssystem. Testingen skulle gi innsikt i hvilke gevinster, utfordringer og ulemper som følger av ulike tekniske valg, herunder hvordan penger, verdipapirer og andre fordringer kan representeres på tokenisert form på en ledger, metoder for overføring av verdier mellom ledgere, hvordan oppgjør kan automatiseres ved hjelp av smartkontrakter,[15] og hvordan roller og ansvar kan håndheves teknisk på en felles teknologisk plattform.

DLT, blokkjede, og ledger

I denne rapporten brukes begrepet «ledger» om en blokkjede for registrering av eierskap og transaksjoner basert på teknologi som faller inn under DLT (Distributed Ledger Technology). DLT er et delt, distribuert og kryptografisk sikret register, og omfatter også andre varianter av distribuert hovedbok enn blokkjeder. De ledger-løsningene som testes i dette prosjektet har ikke nødvendigvis vært delt eller distribuert mellom uavhengige aktører. Blokkjeden kan like godt være sentralisert eller delvis distribuert innenfor et kontrollert og tillatelsesbasert oppsett.

[13] Se Norges Bank (2026)

[14] En teknologisk sandkasse er et avgrenset testmiljø der man kan prøve ut teknologi og funksjonalitet trygt, uten å påvirke reelle systemer eller data.

[15] Smartkontrakt (smart contract): *er programkode som kjøres på en ledger og automatisk håndhever regler, rettigheter og transaksjoner når forhåndsdefinerte vilkår er oppfylt.*

De teknologiske sandkasser er etablert som isolerte testmiljøer uten kobling til produksjonssystemer. Sandkassene er utviklet i samarbeid med to eksterne IT-leverandører. De er basert på to ulike ledger-plattformer og er koblet sammen med en teknologisk bro som gjør det mulig å «flytte» tokens mellom ledgerne ved å låse verdi på én side og utstede en representasjon av verdien på den andre ledgeren.[16]

Formålet med å bruke sandkasser har vært å kunne eksperimentere med ulike arkitekturer og mekanismer uten å ta hensyn til all kompleksiteten i dagens infrastruktur. Betalinger, oppgjør, og forvaltning av verdipapirer og statspapirer er modellert slik at de ligner prosessene i dagens systemer, men er bevisst forenklet og generalisert. Testene omfatter følgelig ikke alle detaljer i eksisterende prosesser, regelverk eller grensesnitt mot eksterne systemer.

Testingen omfattet hverken last- og stresstester, teknologiens evne til å håndtere feilhendelser eller angrep fra ondsinnede aktører. Fokuset har vært på forventet bruk og virkemåte under normale forhold («happy path»), arkitekturvalg, rettighets- og rollemodeller, samt samspillet mellom ledger, smarte kontrakter og tilgrensende systemkomponenter, som broer.

Tilnærmingen til testingen har vært todelt. For det første ville vi utforske teknologiens anvendelighet ved realistiske bruksscenarier basert på praktiske behov og eksisterende prosesser i betalings- og finanssystemet. For det andre ønsket vi å benytte ulike løsninger og arkitekturer i hvert scenario, for å identifisere styrker og svakheter ved ulike tilnærminger og ulik anvendelse av teknologien.[17]

Testingen har hatt som formål å utforske egenskapene ved to typer DSP, som benyttes til ulike formål[18]:

Kunderettede DSP er digitale sentralbankpenger tilgjengelig for publikum, til bruk ved betalinger på utsalgssteder, mellom privatpersoner, ved netthandel og for inn- og utbetalinger til offentlig sektor. Innføring av kunderettede DSP vil innebære en betydelig omlegging av betalingssystemet og kan, ved omfattende utbredelse, påvirke bankenes og andre private aktørers roller og forretningsmodeller i betalingssystemet.

DSP for oppgjør kan sammenlignes med tradisjonelle sentralbankreserver, men på en ny teknologisk plattform basert på ledger-teknologi. DSP for oppgjør er kun tilgjengelig for banker og andre foretak i finansiell sektor med konto i sentralbanken, og benyttes til

[16] I praksis beveger pengene seg aldri, men låses på en ledger for å kunne representeres og benyttes til betalinger på en annen ledger. Derfor er beskrivelser som «flytting» og «broer» kun metaforer for det som skjer i realiteten i en DLT-kontekst.
[17] Testingen var avgrenset til et utvalg ledger-teknologier, herunder Hyperledger Besu og Hyperledger Fabric, og er ment å belyse arkitektur- og designvalg snarere enn å gi en fullstendig teknologisammenligning.
[18] For nærmere beskrivelse av formål med eventuell innføring av DSP, se Norges Bank (2026).

oppgjør mellom banker, blant annet i forbindelse med handel med tokeniserte verdipapirer og tokeniserte bankinnskudd.

Hovedtyngden av testingen har vært rettet mot DSP for oppgjør mellom banker. Med unntak av testcase 1, som fokuserer på kunderettede DSP, omhandler testene bruk av DSP som oppgjørsmiddel i interbankoppgjør og i samspill med tokeniserte verdipapirer og tokeniserte bankinnskudd.

Totalt er det gjennomført seks testcaser i prosjektperioden:

- Testcase 1: Etablering av en ledger for kunderettede DSP og utforskning av egenskapene ved en slik ledger.
- Testcase 2: Etablering en ledger for DSP for oppgjør og utforskning av egenskapene ved en slik ledger.
- Testcase 3: Etablering av en bro for overføring av tokens/verdier mellom ledgere og egenskaper ved en slik bro.
- Testcase 4: Egenskaper ved tokeniserte bankinnskudd (TBI) i en multi-ledger-arkitektur.
- Testcase 5: Handel og oppgjør av tokeniserte verdipapirer på én felles ledger.
- Testcase 6: Utstedelse og livssyklus for statsobligasjoner på én felles ledger.

Deler av testene er gjennomført i samarbeid med DNB som deltakende bank i sandkassene. I utformingen av enkelte testscenarier med verdipapirhandel har også DNB Carnegie bidratt. Dette har gitt mulighet til å teste tekniske løsninger mot relevante behov og transaksjoner i aktørene i finanssektoren.

Hvert testcase beskrives i egne avsnitt med følgende struktur: først beskrives formålet med testcaset, deretter teknisk design. Til slutt oppsummeres resultater, innsikt og funn i hvert testcase. Helt til slutt i rapporten oppsummeres læringspunkter fra alle testcasene.

Testcase 1 – Etablering av en ledger for kunderettede DSP og utforskning av egenskapene ved en slik ledger

Formålet med testcaset

Formålet med dette testcaset var å utforske hvordan kunderettede DSP kan utformes i en to-lags distribusjonsmodell, der sentralbanken har teknisk håndhevbar kontroll[19] over rammer, regler og pengemengde, mens banker og regulerte tredjeparter har ansvar for kundeforhold og kundedata. Testen tok utgangspunkt i at sentralbanken må sikre at ingen andre aktører kan utstede og destruere DSP, samtidig som sentralbanken ikke har innsyn i enkeltkunders betalinger med eller beholdninger av DSP.

Et sentralt mål var å undersøke hvordan roller, rettigheter og krav til forvaltning kan håndheves teknisk i en DLT-basert løsning. Ledgerteknologi inneholder ikke innebygd tilgangsstyring, og testcaset skulle derfor belyse hvordan nødvendige mekanismer for tilgangsstyring og ansvarsdeling kan defineres eksplisitt, både på ledgeren og i tilgrensende systemer. Dette inkluderte å undersøke hvordan løsningen kan støtte prinsippet om at bankene har den direkte kundekontakten, i tråd med dagens ansvarsdeling i betalingssystemet.

Videre var formålet å utforske hvordan sentralbanken kan få sanntids, aggregerte oversikter over sirkulasjon og beholdninger i systemet, uten tilgang til detaljerte kundedata. Testcaset ga dermed innsikt i hvordan tilgangsstyring på ledgernivå kan brukes til å balansere sentralbankens behov for styring og kontroll over utstedelse og destruksjon av sentralbankpenger med hensyn til personvern, ansvarsdeling og bankenes rolle i en kunderettet DSP-løsning.

Teknisk design

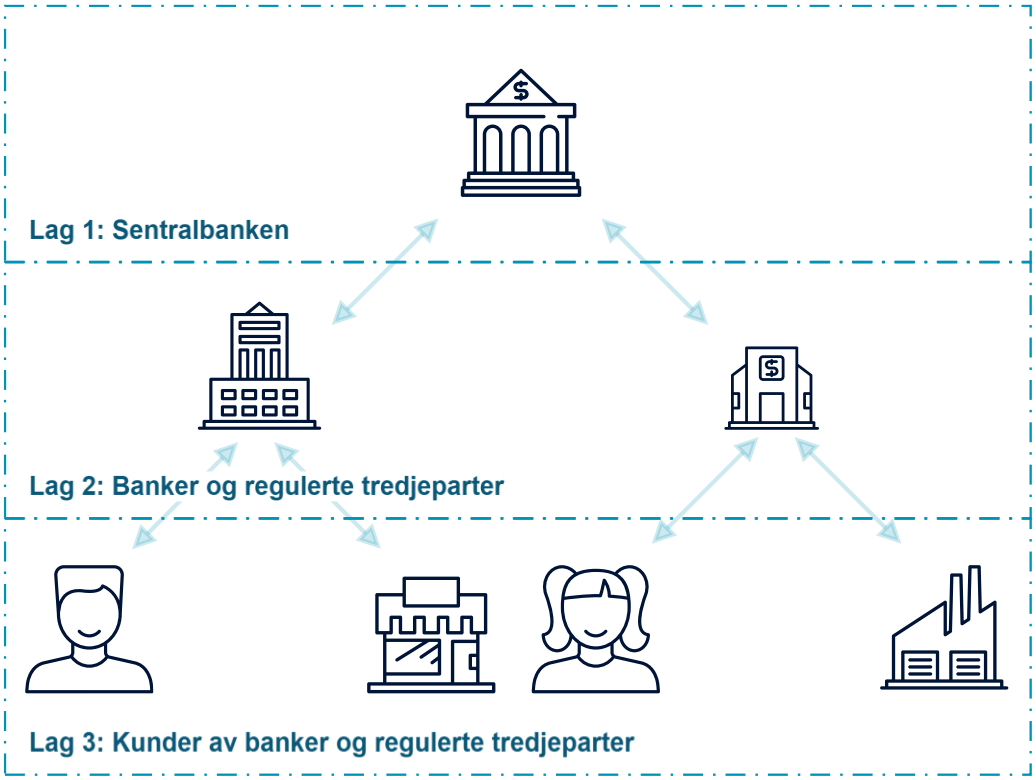
Sandkasseløsningen i dette testcaset er basert på Hyperledger Besu, som er en tillatelsesbasert ledger kompatibel med Ethereum Virtual Machine (EVM).[20]

Løsningen følger en tolags distribusjonsmodell som illustrert i Figur 1. Sentralbanken utgjør øverste lag og er eneste aktør med rettighet til å utstede og destruere kunderettet DSP. Banker og andre godkjente, regulerte tredjeparter utgjør distribusjonslaget og fungerer som

[19] I denne rapporten brukes begrepet «sentralbankens kontroll» for å beskrive sentralbankens teknisk håndhevbare rettigheter og ansvar i den testede arkitekturen, særlig knyttet til utstedelse og destruksjon av DSP, fastsettelse av overordnede regler, og tilgang til aggregerte oversiktsdata.
Begrepet innebærer ikke full operasjonell kontroll over alle komponenter, løpende drift av alle ledgere, eller innsyn i detaljerte transaksjoner eller kundeforhold hos banker og andre aktører. Operasjonelt ansvar, datatilgang og teknisk drift kan være fordelt mellom flere aktører i tråd med arkitektur og rollemodell.

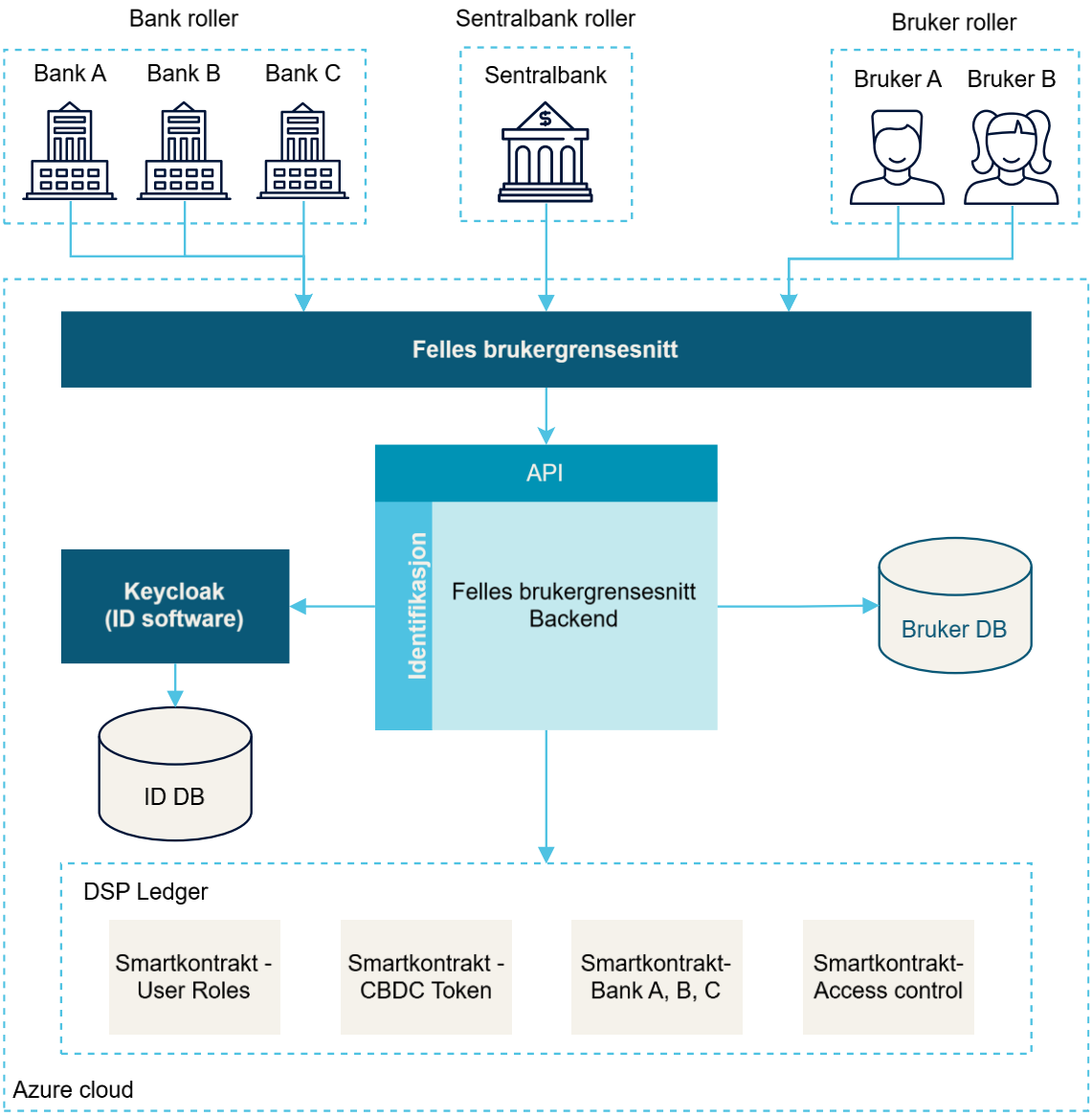
[20] Ethereum Virtual Machine (EVM) er en standardisert kjøremodell for smarte kontrakter som benyttes av Ethereum og en rekke kompatible ledgerplattformer, herunder Hyperledger Besu. EVM definerer hvordan kontraktkode kjøres deterministisk på tvers av noder, hvordan kontoer og kontrakter lagrer tilstand, og hvordan transaksjoner valideres og føres i en delt hovedbok.

mellomledd mellom sentralbanken og sluttkundene. Sluttkunder inngår ikke direkte i ledgerens styringsstruktur, men benytter tjenester levert av banker og andre tredjepartsaktører.



Figur 1: To-lags distribusjonsmodell

Ansvars- og tilgangsfordelingen er håndhevet teknisk gjennom rollebaserte smarte kontrakter på ledgeren. Banker har ansvar for kundeetablering og for å knytte kunder til -ledgeradresser- som er registrert i bankenes egne kontoregistre. Kun adresser som er godkjent av banker kan holde og bruke kunderettet DSP. Sluttkunder kan disponere egen beholdning og gjennomføre betalinger, men kan ikke selv opprette adresser eller endre rettigheter. Sentralbanken har ikke tilgang til kundedata og har ikke ansvar for AML/KYC/CTF, men har gjennom egne roller tilgang til aggregerte oversikter over sirkulasjon, beholdninger og transaksjonsaktivitet på ledgeren.



Figur 2: Overordnet arkitekturskisse av sandkassen til kunderettet DSP

Figur 2 viser løsningsdesignet på et overordnet nivå. Et felles applikasjonslag med brukergrensesnitt og API-er[21] håndterer autentisering, autorisasjon og samspill med ledgeren.

Ulike innloggingsmekanismer er utprøvd, inkludert tradisjonell pålogging via eksterne ID-løsninger og bruk av digitale lommebøker. Selve ledgeren inneholder kun det som er nødvendig for å representere eierskap og overføringer av kunderettet DSP, mens

[21] Application Programming Interface (API) er et sett med regler og protokoller som gjør det mulig for ulike programvareløsninger å kommunisere og utveksle data med hverandre.

kundeinformasjon og identitetsdata forblir hos bankene og utenfor ledgeren. Designet er bevisst valgt for å teste hvordan tekniske mekanismer kan sikre hensiktsmessig ansvarsdeling og kontroll over utstedelse og destruksjon av sentralbankpenger, samtidig som innsyn i kundedata begrenses.

Resultater og funn

Testingen i «retailsandkassen» viser at rollebasert tilgangsstyring kan håndheves teknisk i en DLT-basert løsning, slik at ansvars- og rettighetsfordelingen mellom sentralbank, banker og sluttkunder kan implementeres direkte på ledgeren ved hjelp av smarte kontrakter. Ulike innloggingsmekanismer ble testet, der digitale lommebøker gir høy grad av brukerkontroll, men innebærer risiko ved tap av nøkler. Pålogging via eksterne ID-løsninger gir bedre brukervennlighet og mulighet for gjenoppretting, men medfører et ekstra mellomledd.

Testingen har ikke hatt som mål å gi grunnlag for en helhetlig vurdering av personvern i juridisk forstand, men å belyse tekniske egenskaper som har betydning for personvernforhold. Løsningen er utformet slik at kundeopplysninger forblir hos bankene, mens ledgeren kun inneholder det som er nødvendig for å gjennomføre overføringer av kunderettet DSP. Løsningen baserer seg på pseudonymitet[22], noe som reduserer eksponering av personopplysninger i normal drift. Samtidig innebærer dette en strukturell personvernrisiko dersom ledger-adresser kan knyttes til identitet fordi full transaksjonshistorikk da blir synlig. Valg av ledgermodell og adressestruktur har dermed stor betydning for hvordan krav til personvern kan ivaretas. Tilgangsstyring på applikasjonsnivå er ikke tilstrekkelig for å håndtere sporbarhet på ledger-nivå.

Testen viste videre at oppdatering og videreutvikling av smarte kontrakter er krevende, ettersom kontrakter ikke kan endres etter publisering. Ved behov for justeringer må det etableres nye smartkontrakter og oppfølging av avhengigheter.

[22] Pseudonymitet er en egenskap ved et system der handlinger og data knyttes til en identifikator (et pseudonym) i stedet for direkte til en virkelig identitet. Pseudonymet fungerer som en stedfortreder for personen eller aktøren, og gjør det mulig å følge aktivitet over tid uten nødvendigvis å vite hvem som står bak. Dersom koblingen mellom pseudonym og identitet etableres, kan imidlertid tidligere og fremtidige handlinger knyttes til den samme identiteten.

Testcase 2 – Etablering av en ledger for DSP for oppgjør og egenskapene ved en slik ledger

Formålet med testcaset

Formålet med dette testcaset var å etablere en teknisk sandkasse for DSP som brukes til oppgjør mellom banker (DSP for oppgjør), og å undersøke om Hyperledger Fabric[23] kan støtte en slik løsning på en trygg og kontrollerbar måte. Testcaset skulle gi praktisk erfaring med de grunnleggende funksjonene i et slikt interbank oppgjørssystem, der DSP benyttes som oppgjørsmiddel mellom banker på en felles ledger.

Et sentralt mål var å utforske hvordan rettigheter, innsyn og konfidensialitet kan håndheves teknisk i et slikt system. Dette innebærer at det må være entydig definert hvilke handlinger ulike aktører har rett til å utføre på ledgeren, og at disse rettighetene kan håndheves teknisk. Samtidig må løsningen sikre konfidensialitet mellom banker, slik at banker kan gjøre opp betalinger seg imellom uten at den enkelte bank får tilgang til konfidensiell informasjon om andre bankers betalinger mv.

Videre var formålet å undersøke hvordan sentralbanken kan ha tilgang til tilstrekkelig informasjon for å sikre at oppgjørssystemet fungerer som forutsatt. Dette omfatter blant annet oversikt over samlet mengde og bruk av DSP i omløp, uten at sentralbanken har tilgang til detaljerte transaksjonsdata mellom banker.

Teknisk design

I testcase 2 benyttes Hyperledger Fabric som DLT-plattform for å utforske egenskaper ved *DSP for oppgjør* i interbankoppgjør. Løsningen er etablert som en felles oppgjørsplattform der sentralbanken og alle deltakende banker inngår i samme nettverk, men med desentralisert drift og differensiert innsyn og informasjonsdeling.

Desentralisert drift vil si at hver bank drifter og er teknisk ansvarlig for sin egen node på ledgeren, og er dermed ansvarlig for egen teknisk infrastruktur. Figur 3 visualiserer hvordan «bank peer node» håndteres av hver bank, og kobler seg til ledgeren.

I den testede løsningen er styring av innsyn og informasjonsdeling realisert ved bruk av *private data collections* (PDC). PDC gjør det mulig å gjennomføre oppgjør på én felles ledger, samtidig som utvalgte deler av transaksjonsdata kun deles mellom autoriserte parter.

[23] Hyperledger Fabric er en åpen kildekode-plattform for delt «ledger» (DLT) utviklet under Linux Foundation, laget for virksomheter som trenger kontrollert tilgang og mulighet til å skjerme data mellom deltakere.

Testscenarioet er videre utvidet for å utforske en UTXO-basert modell for representasjon av DSP for oppgjør, i kontrast med den kunderettede sandkassen som er utformet som en kontobasert modell.

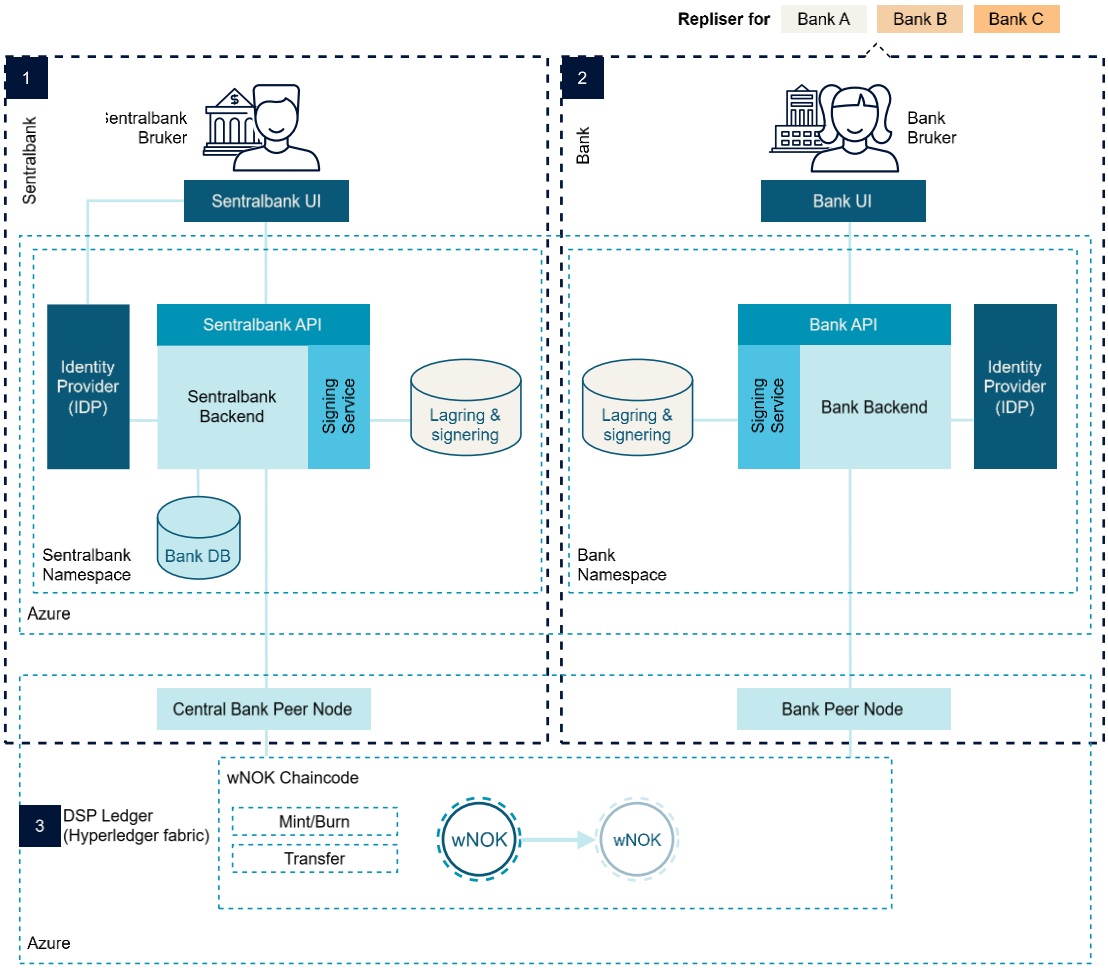
Token modeller

UTXO-modellen kan forstås som digitale «sedler» eller «verdibiter» med en bestemt verdi. Når en betaling gjennomføres, må én eller flere verdibiter brukes slik at de samlet dekker betalingsbeløpet. Dersom det for eksempel betales 100 kroner ved å bruke en digital «200-kronerseddel», forbrukes den opprinnelige verdibiten, og systemet oppretter to nye: én som overføres til mottaker (100 kroner) og én som returneres til betaler som «vekslepenger» (100 kroner). Den gjenværende verdien er et eksempel på en *unspent transaction output* (UTXO).

I en UTXO-modell er eierskap ikke knyttet til en konto med saldo, men til hver enkelt verdibit. Eierskap defineres av hvem som er autorisert til å bruke verdibiten som input i en ny transaksjon. Hver verdibit har sin egen historikk fra utstedelse til den destrueres, noe som gjør det teknisk enklere å verifisere at samme verdibit ikke kan brukes to ganger, og å følge verdier gjennom etterfølgende transaksjoner. Dette gir høy sporbarhet på token-nivå og kan være godt egnet for parallell behandling av transaksjoner.

Den kontobaserte modellen ligner på hvordan banker fører bankinnskudd i dag. Hver deltaker har en konto som til enhver tid viser en samlet saldo. Når det gjennomføres en betaling på 100 kroner, reduseres betalers saldo med 100 kroner og mottakers saldo øker tilsvarende. I denne modellen er fokuset på saldo per konto, ikke på hvilke konkrete «verdibiter» saldoen består av. Sporbarheten er i større grad knyttet til transaksjonsloggen og endringer i kontosaldoer over tid: man kan se hvem som har sendt og mottatt verdier, og hvordan saldoer endres, men verdien har ikke en egen «seddel-historikk». Den kontobaserte modellen gir ofte enklere saldohåndtering og mer intuitiv tilgangs- og innsynsstyring på kontonivå.

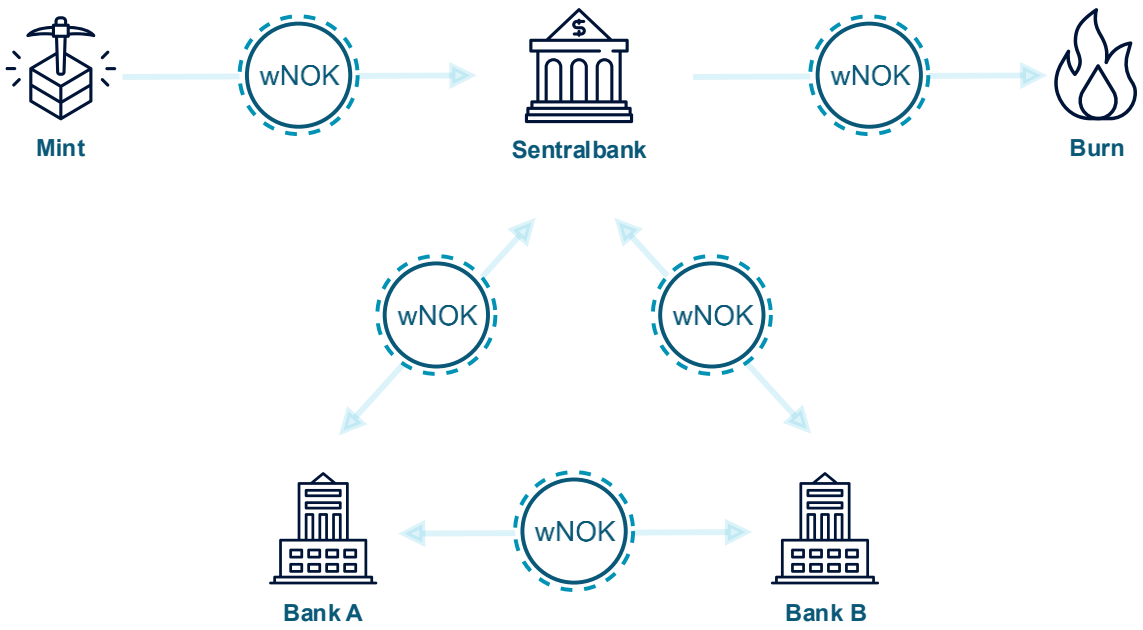
Begge modellene kan brukes til å støtte oppgjør digitalt, men de representerer ulike tekniske tilnærminger med forskjellige styrker og svakheter knyttet til sporbarhet, kompleksitet og håndtering av tilganger.



*Figur 3: Arkitekturskisse av sandkassen til DSP for oppgjør.**1)** viser komponentene som sentralbanken benytter og som ligger under sentralbankens infrastruktur og driftsansvar. **2)** viser komponentene som hver deltakende bank benytter og som ligger under hver bank sin infrastruktur og driftsansvar. **3)** viser Hyperledger Fabric nettverket, altså selve ledgeren. Merk at hver «peer node» ligger inne «på ledgeren». Det er også her wNOK (DSP for oppgjør) tokenet ligger.*

Løsningen har to separate brukergrensesnitt, ett for sentralbanken og ett annet for banker, begge er tilpasset deres roller. Sentralbankens visning gir en samlet oversikt over DSP i omløp, sentralbankens totalbeholdning av DSP, transaksjonsaktivitet over valgte tidsperioder og totalbeløp per bank. Bankenes brukergrensesnitt kan benyttes til betalingsoppdrag og oversending av DSP til andre banker eller tilbake til sentralbanken, samt å få informasjon om bankens egen saldo og egne transaksjoner.

I testcaset inngår tre deltakende banker (Her kalt Bank A, Bank B og Bank C i Figur 3). På denne måten kunne vi teste egenskaper som tilgang til ledgeren og innsyn i og skjerming av informasjon. Med flere deltakende banker blir det lettere å vurdere hvordan løsningen oppfører seg når flere deltakere inngår i samme infrastruktur.



Figur 4: Oversikt over hvordan DSP for oppgjør opprettes (mint), destrueres (burn), og flyttes mellom deltagere i systemet. I figuren er wNOK = DSP for oppgjør.

Sentralbanken utsteder DSP for oppgjør og de nyskapte tokenene registreres på sentralbankens adresse på ledgeren. Derfra distribueres DSP for oppgjør til deltakende banker (her illustrert som Bank A og Bank B). Bankene kan deretter benytte DSP for oppgjør fritt til oppgjør for betalinger seg imellom gjennom overføringer av tokens på ledgeren. Når en bank ønsker å konvertere hele eller deler av sin beholdning av DSP for oppgjør til tradisjonelle reserver, sendes DSP for oppgjør tilbake til sentralbanken, som destruerer tokenene og tar dem ut av sirkulasjon. Som en integrert del av prosessen vil beholdningen av tradisjonelle reserver økes tilsvarende. Denne delen av transaksjonen har imidlertid ikke inngått i testen.

Resultater og funn

Dette testcaset viser at Hyperledger Fabric kan være et aktuelt teknologivalg for DSP brukt til oppgjør mellom banker, særlig i situasjoner der det er ulike behov for innsyn og hvor informasjon må kunne skjermes. Testene viser at denne fleksibiliteten medfører kostnader og utfordringer i form av mer krevende oppsett, drift og forvaltning enn løsninger basert på én fullt delt ledger med likt innsyn for alle.

Testen viser videre at Hyperledger Fabric gjør det teknisk mulig å håndtere DSP for oppgjør både i en kontobasert modell og i en UTXO-basert modell. Begge modellene støtter de grunnleggende funksjonene som kreves i et interbanksystem, herunder utstedelse, overføring mellom banker og destruering av tokens. UTXO-modellen gir i tillegg sporbarhet på tokennivå, noe som kan være relevant ved etterkontroll, feilsøking og rekonstruksjon av hendelser.

Testen viser at bruk av PDC gjør det mulig å styre hvilke transaksjonsdata ulike aktører har teknisk lesetilgang til, selv om oppgjørene gjennomføres på én felles ledger. I den testede løsningen kan sentralbanken overvåke at interbankoppgjør finner sted, herunder beløp, motparter og tidspunkt, samt tilgang til aggregerte oversikter over sirkulasjon. Detaljerte bilaterale transaksjonsdata og informasjon om underliggende kundeforhold er skjermet og følgelig ikke tilgjengelig for sentralbanken. PDC benyttes til å sikre at banker kun har innsyn i oppgjør de selv er part i, og ikke i andre bankers transaksjoner.

Testene indikerer at Hyperledger Fabric kan benyttes som teknologi for en løsning for DSP for oppgjør med teknisk gjennomførbarhet for flere tokenmodeller, presis innsynsstyring og tilgang til aggregerte oversiktsdata. Det største problemet testen avdekket ved denne løsningen er at strukturen blir relativt kompleks. Etter hvert som flere banker kobles på og flere informasjonsgrenser skal håndheves, øker kravene til standardisering, styring og operasjonell robusthet sammenlignet med løsninger basert på én fullt delt ledger.

Testcase 3 – Etablering av en bro for overføring av tokens/verdier mellom ledgere og egenskaper ved en slik bro

Formålet med testcaset

Formålet med testcaset var å undersøke konsekvenser tilknyttet DSP for oppgjør i en infrastruktur der hhv. DSP og andre verdier er registrert på ulike ledgere.

Testcaset tok utgangspunkt i at sentralbanken har behov for å styre bruken av DSP på en egen, lukket ledger, samtidig som andre aktører ønsket å utvikle betalingsløsninger ved hjelp av smartkontrakter ledgere/på egne eksterne ledgere. Gjennom testen ønsket vi å avklare hvordan DSP for oppgjør kunne brukes som oppgjørsmiddel på tvers av systemer, hvordan styring og rollebasert ansvar kan ivaretas, og hvilke egenskaper en slik struktur hadde.

I testcaset var DSP for oppgjør registrert på en sentralbankstyrt ledger, mens verdipapirer og applikasjonslogikk lå på en ekstern ledger utenfor sentralbankens kontroll. En bro ble

benyttet som teknisk mekanisme for å muliggjøre samhandling mellom de to ledgerne i en slik multiledgerstruktur.

Teknisk design

Broen i testcase 3 er bygget som en sentralisert kobling mellom to separate blokkjede-miljøer: en kjerneledger for DSP for oppgjør basert på Hyperledger Fabric, og en ekstern ledger basert på Hyperledger Besu. Hensikten er å gjøre det mulig å bruke DSP i applikasjoner på den eksterne ledgeren, uten at DSP faktisk flyttes ut av kjerneledgeren, og uten å gi fra seg kontrollen over pengeutstedelse og pengemengde.

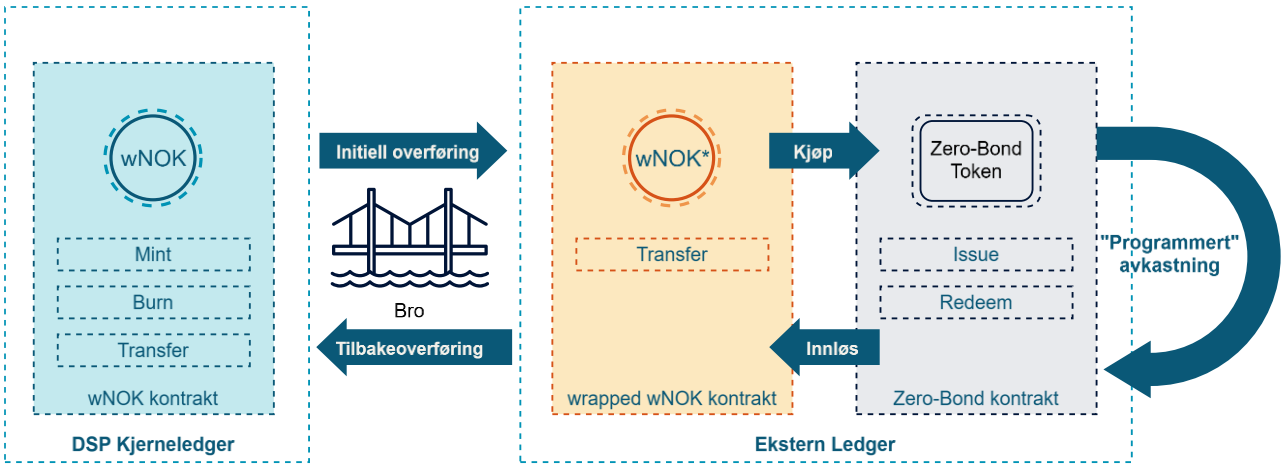
Figur 5 viser hovedkomponentene og pengestrømmen mellom dem. På venstre side ligger kjerneledgeren med DSP for oppgjør-kontrakten. På høyre side ligger den eksterne Besu-ledgeren, som blant annet inneholder en wrapped[24] DSP for oppgjør -kontrakt (wNOK*) og en enkel nullkupongs obligasjonskontrakt. Mellom disse ligger brokomponenten, som har systemtilgang til begge ledgerne.

DSP for oppgjør kan kun utstedes og destrueres av sentralbanken. I brooppsettet benyttes en forvaltningskonto på kjerneledgeren for midlertidig låsing av DSP når midlene gjøres tilgjengelig på en ekstern ledger. Dette sikrer at samme DSP-verdi ikke kan disponeres parallelt på flere plattformer /i flere systemer.

På den eksterne ledgeren representeres DSP for oppgjør gjennom en smartkontrakt der kun broen har rettigheter til å utstede og destruere tokens. Broen er implementert som en ekstern komponent med tilkobling til begge ledgerne og ansvar for koordinering av overføringer av DSP.

Den konkrete overføringsflyten er illustrert i figuren under.

[24] Et wrapped token er en representasjon av en verdi som er låst på en annen ledger, og som kan brukes som betalingsmiddel eller oppgjørsmiddel på mål-ledgeren.



Figur 5: Oversikt over hvordan penger flyttes mellom ledgere. På kjerneledgeren er DSP for oppgjør notert som «wNOK». På den eksterne ledgeren er wrapped DSP for oppgjør notert som «wNOK»*

Når en bank ønsker å bruke DSP i en applikasjon på den eksterne ledgeren, initieres en brooperasjon. I et typisk forløp vil en bank logge inn i et webgrensesnitt koblet til kjerneledgeren, velge beløp og måladresse på den eksterne ledgeren, og sende en forespørsel til broen om å «flytte» DSP til den oppgitte adressen.

Broen starter da en prosess som i hovedtrekk følger disse stegene:

1. Broen oppretter først en *asset reference* på kjerneledgeren og låser det aktuelle DSP for oppgjør-beløpet i sin forvaltningskonto. Etter denne operasjonen er beløpet fortsatt på kjerneledgeren, men ikke lenger disponibelt for vanlige transaksjoner.
2. Når låsingen er bekreftet på kjerneledgeren, sender broen en transaksjon til *wrapped DSP for oppgjør*-kontrakten på den eksterne ledgeren, som utsteder et tilsvarende *wrapped DSP for oppgjør* beløp til mottakerens adresse på den eksterne ledgeren.
3. Mottakeren kan nå bruke *wrapped DSP for oppgjør* i applikasjoner på den eksterne ledgeren, for eksempel til å kjøpe nullkupongs obligasjonstoken. Nullkupongskontrakten mottar *wrapped DSP for oppgjør* fra kjøper, utsteder obligasjonstoken, og ved forfall betaler kontrakten tilbake pålydende beløp til obligasjonseierens adresse i *wrapped DSP for oppgjør*.

Når aktøren på den eksterne ledgeren ønsker å føre midlene tilbake til kjerneledgeren, gjennomføres en omvendt transaksjonsflyt. Brukeren (eller en applikasjon på dens vegne) sender *wrapped DSP for oppgjør* tilbake til broens adresse på Besu (såkalt «bridge back»-operasjon). Broen validerer at mottatt beløp stemmer og utfører deretter to steg:

1. Broen kaller *wrapped DSP*-kontrakten og destruerer det mottatte *wrapped DSP*-beløpet. Etter dette finnes ikke lenger *wrapped DSP for oppgjør* på den eksterne ledgeren.

2. Broen sender en transaksjon til kjerneledgeren som låser opp det tilsvarende DSP for oppgjør-beløpet i forvaltningskontoen og overfører det tilbake til mottakerens konto på kjerneledgeren.

Prosessen er bygd som en koordinert og sekvensiell flyt, der låsing av DSP på kjerneledgeren må være bekreftet før *wrapped DSP for oppgjør* kan utstedes på den eksterne ledgeren. Dersom låsing på kjerneledgeren ikke lykkes, avbrytes prosessen og det utstedes ingen *wrapped DSP*. Ved tilbakeføring gjennomføres destruksjon av *wrapped DSP* på den eksterne ledgeren før midlene låses opp på kjerneledgeren. Dersom destruksjon lykkes, men unlock-operasjonen på kjerneledgeren feiler, forblir verdien låst og utilgjengelig inntil broen enten lykkes med nytt forsøk eller rekonstruerer en konsistent tilstand basert på tidligere bekreftelser. Det åpnes ikke for at *wrapped DSP* blir stående i omløp uten en bekreftet lås på kjerneledgeren.

Så lenge DSP for oppgjør-beløpet står låst på kjerneledgeren, forblir den samlede mengden DSP under sentralbankens kontroll, selv om midlene midlertidig gjøres tilgjengelig for bruk på en ekstern ledgeren i form av *wrapped DSP for oppgjør*. Låse- og utstedelsesmekanismen sikrer at det til enhver tid finnes en entydig disponering av beløpet: enten som tilgjengelig DSP på kjerneledgeren eller som låst DSP på kjerneledgeren med en tilsvarende *wrapped DSP* på ekstern ledger. Det oppstår dermed verken dobbeltbruk eller avvik i den samlede mengden DSP.

Siden broen er en sentral komponent med rettigheter til å låse midler på kjerneledgeren og utstede/brenne tokens på den eksterne ledgeren, er det tekniske designet lagt opp med tydelig ansvarsdeling og begrensede rettigheter. På kjerneledgeren har broen kun de rettighetene som trengs for å flytte midler inn og ut av egen forvaltningskonto, men ikke til å utstede nye DSP. På den eksterne ledgeren er broens konto satt som eier/administrator av *wrapped DSP*-kontrakten, med eksklusiv tilgang til utstedelse- og destruksjon-funksjonene. Det er ikke mulig for at andre enn broen å endre total mengde *wrapped DSP for oppgjør*.

Broens funksjon medfører at sentralbanken ikke behøver å drifte eller være ansvarlig for infrastrukturen på den eksterne ledgeren. Det er tilstrekkelig at broen har en konto på den eksterne ledgeren og tilgang til et node-endepunkt for å sende og lese transaksjoner. Dette gir fleksibilitet til å benytte DSP til betalinger på eksterne ledgere uten å ta driftsansvar for hele nettverket av ledgere. Samtidig forutsetter denne arkitekturen at den eksterne ledgeren har et styrings- og kontrollregime som hindrer at kontrakten som representerer *wrapped DSP for oppgjør* kan endres, erstattes eller omgås på en måte som svekker broens funksjon. Det innebærer at den tekniske og operasjonelle kontrollen med den eksterne infrastrukturen må vurderes før DSP kan tas i bruk som oppgjørsmiddel på en slik plattform.

Det tekniske designet innebærer også enkelte begrensninger. I denne testen kan broen kun overføre DSP for oppgjør mellom ledgere, ikke andre typer tokens eller data. Den gjør det heller ikke mulig å kalle smarte kontrakter på tvers av ledgerne direkte.

Nullkupongsobligasjonskontrakten er et eksempel på en ren applikasjon som forekommer på den eksterne ledgeren og kun bruker *wrapped DSP for oppgjør* som betalingsmiddel; logikken i denne kontrakten påvirker ikke kjerneledgeren direkte. Dette er et viktig skille mellom en bro som kun benyttes til likviditetsoverføring og mer avanserte interoperabilitetsløsninger hvor kontrakter på én ledger kan trigge kontrakter på en annen.

Designet som er brukt i testcase 3 tar utgangspunkt i at broen skal være forståelig og kontrollerbar også fra et forretnings- og tilsynsperspektiv. Hvert steg i flyten mellom ledgerne kan forklares som enkle operasjoner: *lock, mint (utstede), transfer, burn (destruere), unlock*. Figur 5 visualiserer hvordan disse stegene henger sammen, slik at det er mulig å følge tokenene gjennom hele forløpet, både for tekniske og ikke-tekniske lesere. Dette gir et rammeverk der bruken av DSP i eksterne applikasjoner kan utvides gradvis, uten at grunnprinsippene for kontroll over pengemengden endres.

Resultater og funn

Testene viser at det er mulig å holde entydig kontroll over én og samme DSP-verdi selv om midlene gjøres tilgjengelig på en ekstern ledger via en bro. Løsningen sikrer at de samme pengene aldri er disponible på to steder samtidig, ved at DSP-verdien låses på kjerneledgeren samtidig som det utstedes en representasjon av verdien på den eksterne ledgeren, forutsatt at broen fungerer korrekt.

Oppgjørsprosessen i det bro-baserte oppsettet fungerte etter hensikten innenfor testens forutsetninger, men hadde ikke samme grad av samtidighet og endelighet som oppgjør innenfor én felles ledger. Ved oppgjør over bro består prosessen av flere koordinerte trinn på tvers av ledgere, noe som kan gi midlertidige mellomtilstander ved forsinkelser eller tekniske avbrudd. I testen ble dette håndtert i et kontrollert testmiljø, men i et reelt driftsmiljø vil slike oppgjør prosesser kreve kontinuerlig operasjonell overvåking og aktiv feilhåndtering.

Bro-løsningen forutsetter at én aktør har privilegert tilgang til begge systemene for å kunne låse opp, utstede og destruere DSP. I denne testen var denne rollen lagt til sentralbanken. Selv om den tekniske driften av broen i teorien kan utføres av en tredjepart, forutsetter løsningen at sentralbanken til enhver tid har kontroll over og ansvar for de funksjonene som påvirker samlet pengemengde. Utstedelse og destruksjon av DSP må derfor være teknisk og organisatorisk underlagt sentralbanken, uavhengig av hvem som drifter den underliggende infrastrukturen. En slik sentralisering gir klar og teknisk håndhevbar kontroll

over pengemengden, men innebærer samtidig økt ansvar og strengere krav til drift, sikkerhet, og oppfølging av den aktøren som eventuelt forvalter broen.

I arkitekturer med flere uavhengige ledgere vil interoperabilitet kreve en form for bro eller tilsvarende interoperabilitetsprotokoll mellom ledgerne. Når DSP i hovedsak benyttes som oppgjørsmiddel for prosesser som gjennomføres på eksterne ledgere, realiseres få av de sentrale gevinstene ved DLT, som tett integrasjon mellom penger og forretningslogikk, atomisk samhandling og redusert behov for ekstern orkestrering. Testene indikerer at de fulle gevinstene ved bokkjedebasert DSP-løsning forutsetter at eksterne aktører kan bruke DSP direkte i egne løsninger, slik at penger, logikk og prosesser kan samhandle på samme teknologiske plattform.

Testcase 4 – Egenskaper ved tokeniserte bankinnskudd i en multi-ledger arkitektur

Formålet med testcaset

Formålet med dette testcaset var å utforske egenskaper, fordeler og utfordringer ved en blokkjedebasert infrastruktur der tokeniserte bankinnskudd (TBI)[25] benyttes som betalingsmiddel, og der DSP for oppgjør ble brukt som endelig oppgjørsmiddel mellom de involverte bankene. Testen skulle vise hvordan banker kunne utstede og håndtere TBI på egen ledgerteknologi, undersøke om TBI kan konverteres sikkert og sømløst til kunderettet DSP utstedt av sentralbanken, og samtidig belyse fordeler og ulemper ved en struktur der aktørene opererer på flere ledgere. Testcaset bygger videre på erfaringer fra testcase 3, men med et noe annet fokus: her er TBI betalingsmiddelet, mens DSP for oppgjør benyttes utelukkende til interbankoppgjør.

En sentral motivasjon var å teste ut egenskaper i et scenario der betalingssystemet ikke er bygget på én felles ledger for alle banker og aktører, men der hver bank i større grad kunne velge teknologiplattform og innovasjonstempo. Dette samsvarer både med internasjonale beskrivelser av «*unified ledgers*»[26] og «*regulated liability networks*»[27], og med et marked der

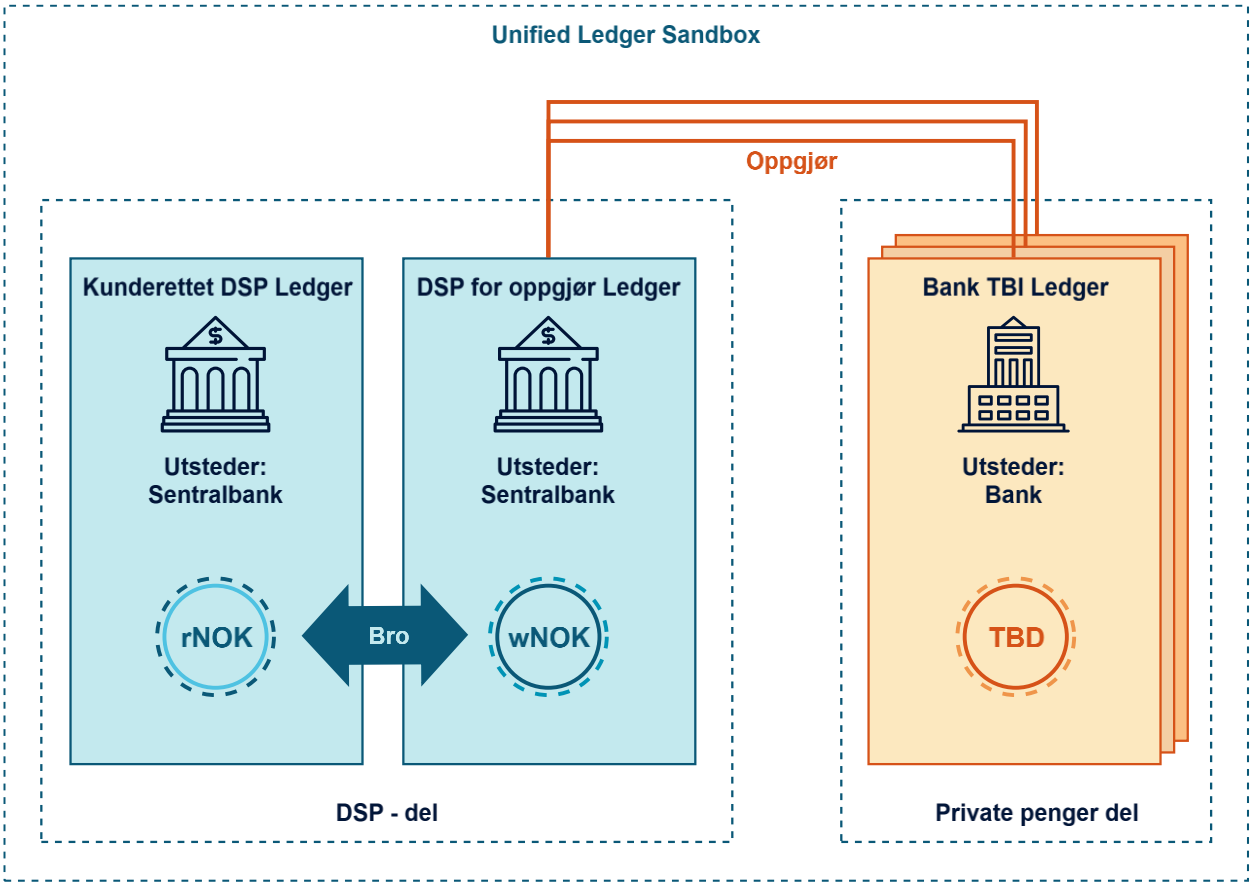
[25] TBI (tokenisert bankinnskudd): Et digitalt bankinnskudd utstedt av en kommersiell bank som en token på en DLT/ledger, der tokenet representerer et krav på banken med samme verdi som innskuddet (typisk 1:1).
[26] BIS´ "Unified Ledger": Konsept for en felles, programmerbar plattform der tokeniserte penger (sentralbankpenger og bankinnskudd) og tokeniserte aktiva kan samhandle, slik at transaksjoner kan automatiseres og avvikles mer sømløst – gjerne med endelig oppgjør i sentralbankpenger. (BIS, 2023 og BIS, 2025).
[27] Regulated Liability Network (RLN): Konsept for en regulert (ofte tillatelsesbasert) finansiell infrastruktur med delt ledger som registrerer, overfører og gjør opp tokeniserte forpliktelser ("liabilities") fra regulerte

nasjonale og internasjonale aktører må samhandle på tvers av system- og landegrenser. Testen hadde også et innovasjonsmål: å undersøke om bankinnskudd i tokenisert form muliggjorde nye prosesser eller brukstilfeller, herunder om bankene kunne tilby kundene rask realtidsutveksling mellom TBI og kunderettet DSP på tvers av teknologier på en trygg og kontrollert måte, og hvordan interbankoppgjør kan håndteres når både sentralbankens DSP og bankenes TBI-systemer inngikk i samme testmiljø.

Teknisk design

Testoppsettet omfatter mekanismer for utstedelse og håndtering av TBI hos den enkelte bank, konvertering mellom TBI og kunderettet DSP, samt oppgjør mellom banker i DSP for oppgjør. Løsningen bygger videre på arkitekturen fra testcase 3, der en kjerneledger for DSP for oppgjør og en ledger for kunderettet DSP er koblet sammen via en bro styrt av sentralbanken. I testcase 4 er det i tillegg etablert separate ledgere for TBI hos hver bank. Den samlede arkitekturen omtales som en «*Unified Ledger Sandbox*» og er illustrert i Figur 6.

utstedere – typisk sentralbankpenger, tokeniserte bankinnskudd og e-penger – med mål om oppgjør til pari og bedre samhandling mellom institusjoner.



Figur 6: Oversikt på høyt nivå av den implementerte løsningen. Kunderettet DSP og DSP for oppgjør på to forskjellige ledgere er koblet sammen via en bro som styres av sentralbanken. TBI-er finnes på ledgere hos forretningsbankene. Transaksjoner med TBI gjøres opp i DSP for oppgjør.

I Figur 6 vises kjerneelementene i løsningen: DSP for oppgjør og kunderettet DSP ligger på hver sin ledger, forbundet av en bro som kan låse DSP for oppgjør og utstede kunderettet DSP, og omvendt. I tillegg har hver bank en egen TBI-ledger der bankens kunders innskudd er representert som TBI-tokens. Transaksjoner i TBI mellom banker gjøres opp i DSP på DSP for oppgjør-ledgeren. TBI-ledgerne er implementert som egne kanaler i det samme distribuerte nettverket som brukes for DSP for oppgjør, men der hver kanal er logisk adskilt og styrt av den enkelte bank. Disse kanalene kan anses som hver sin ledger, se tekstboks under.

Kanaler (channels) i Hyperledger Fabric

I Hyperledger Fabric er en kanal et logisk avgrenset delnettverk som fungerer som en egen ledger, med egne data, egne smarte kontrakter og egne tilgangsregler. Kun aktører som er medlemmer av en kanal har teknisk innsyn i transaksjoner og beholdninger på kanalen.

En organisasjon kan være medlem av flere kanaler samtidig. Dette gjør det mulig å kjøre flere logisk adskilte ledgere på én felles teknologisk plattform, og å skjerme sensitiv informasjon mellom aktører.

Den aktøren som oppretter og eier en kanal, har også ansvar for den operasjonelle driften av kanalen, herunder tilgangsstyring, medlemskap, smartkontrakter og drift av tilhørende infrastruktur.

I denne løsningen er hver banks TBI-ledger implementert som en egen kanal, styrt og operert av den enkelte bank. Kanalene behandles derfor arkitektonisk som separate ledgere, selv om de teknisk sett er del av samme Fabric-nettverk.

På hver TBI-ledger kjører en egen smartkontrakt som definerer hvordan TBI kan utstedes, destrueres og overføres mellom kundekontoer. Ledgerne er kontobaserte: hver kunde har en identifiserbar konto med en saldo i TBI. Kun banken har rettigheter til å utstede og destruere TBI på egen ledger, mens overføringer mellom kundekonti skjer etter regler definert i smartkontrakten. Sentralbanken er medlem på kanalnivå i sandkassen for å kunne lese aggregerte data (for eksempel total utestående TBI), men har ingen styringsrettigheter over TBI-kontraktene.

Målet med testcase 4 har vært å anvende dette oppsettet til å utforske egenskapene ved DLT-baserte versjoner av to finansielle prosesser:

- *Konvertering fra TBI til kunderettet DSP* (kan sammenlignes med et kontant uttak fra en bankkonto), og
- *Interbankoppgjør for TBI* (oppgjør i *DSP for oppgjør* mellom banker, som følge av overføringer mellom kunder i ulike banker).

Begge prosessene er implementert slik at de teknisk kan fungere uavhengig av hvilken ledger den enkelte bank har valgt for sin TBI-ledger. I sandkassen er TBI bygget på Hyperledger Fabric-kanaler, men logikken for oppgjør og konvertering er lagt på DSP for oppgjør-ledgeren og i egen orkestreringskomponent, slik at det i prinsippet kunne vært brukt andre typer TBI-ledgere så lenge de kan kobles opp mot disse komponentene.

Bankene har dermed full frihet til å velge ledger for sine TBI. Med dette har vi bevisst valgt en av de mest komplekse måtene å utstede TBI.

Orkestrering av prosesser i et multi-ledger oppsett

I det testede multi-ledger-oppsettet kan sammensatte prosesser som konvertering av TBI og interbankoppgjør ikke gjennomføres som én atomisk transaksjon på én ledger. Sammenhengen mellom nødvendige operasjoner håndheves derfor gjennom eksplisitt orkestrering i applikasjonslaget, utenfor ledgerne. Orkestreringen er implementert som tilstandsbasert prosesslogikk (*finite state machines*), og styrer rekkefølge på operasjoner, låsing av midler for å forhindre dobbeltbruk, kall til broen der dette inngår, samt håndtering av feil og avbrudd.

Det er viktig å skille orkestreringen fra selve broen.

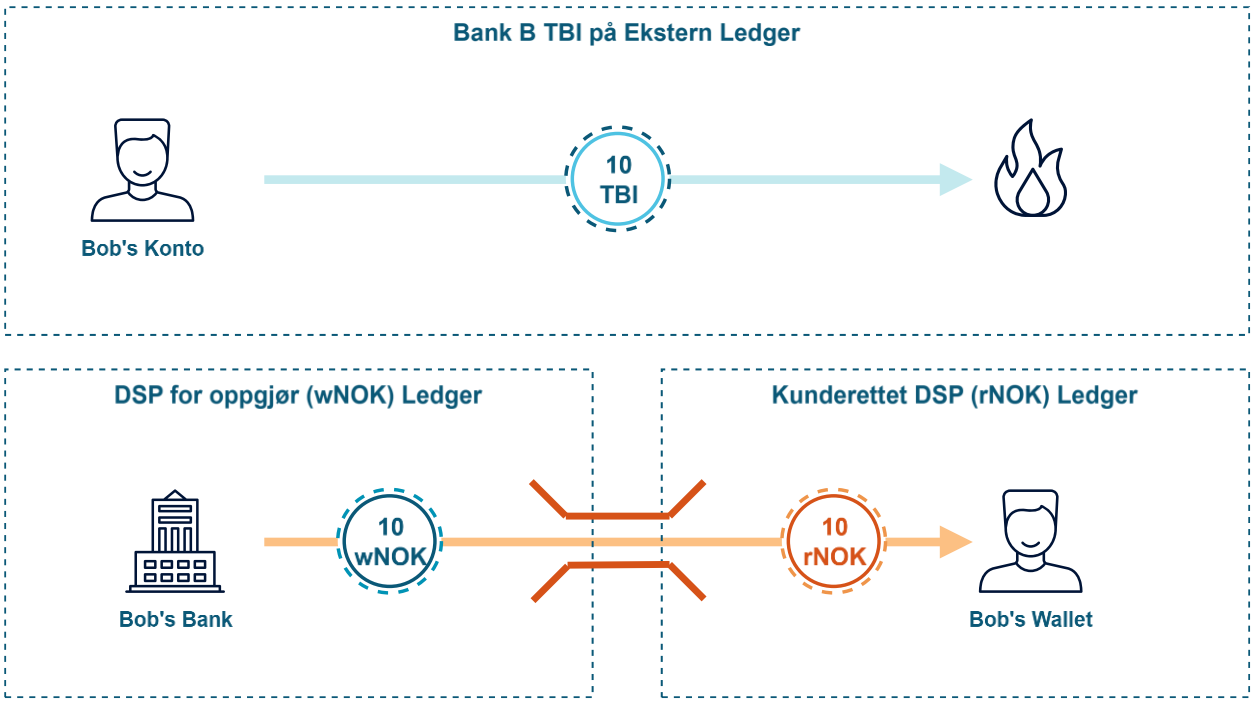
En **bro** er en avgrenset teknisk komponent som muliggjør overføring eller representasjon av verdier mellom ledgere, typisk ved låsing av verdi på én ledger og utstedelse av en tilsvarende representasjon på en annen. Broen har ingen forretningslogikk og ingen helhetlig forståelse av prosessen den inngår i.

Orkestreringen er derimot ansvarlig for den samlede prosessen og avgjør *når* og *om* irreversible steg skal utføres. Oppgjør i et multi-ledger-oppsett oppnås derfor ikke gjennom broen alene, men gjennom en kontrollert orkestrert prosess der irreversible operasjoner først gjennomføres når nødvendige forutgående bekreftelser foreligger. Dersom feil oppstår før et definert punkt i prosessen, kan hele forløpet avbrytes og reverseres.

I prinsippet kan broer brukes som en del av en orkestreringslogikk. *I den testede løsningen er broen imidlertid kun brukt som en verdioverføringsmekanisme*, mens selve orkestreringen – inkludert styring av rekkefølge, låsing og feilhåndtering – ligger eksplisitt i applikasjonslaget utenfor broen.

Konvertering fra TBI til kunderettet DSP

Figur 7 illustrerer konverteringen fra TBI til kunderettet DSP. Prosessen kan ses som den DLT-baserte parallellen til at en kunde gjennomfører et kontantuttak fra sin bankkonto.



Figur 7: Konvertering fra TBI til kunderettet DSP

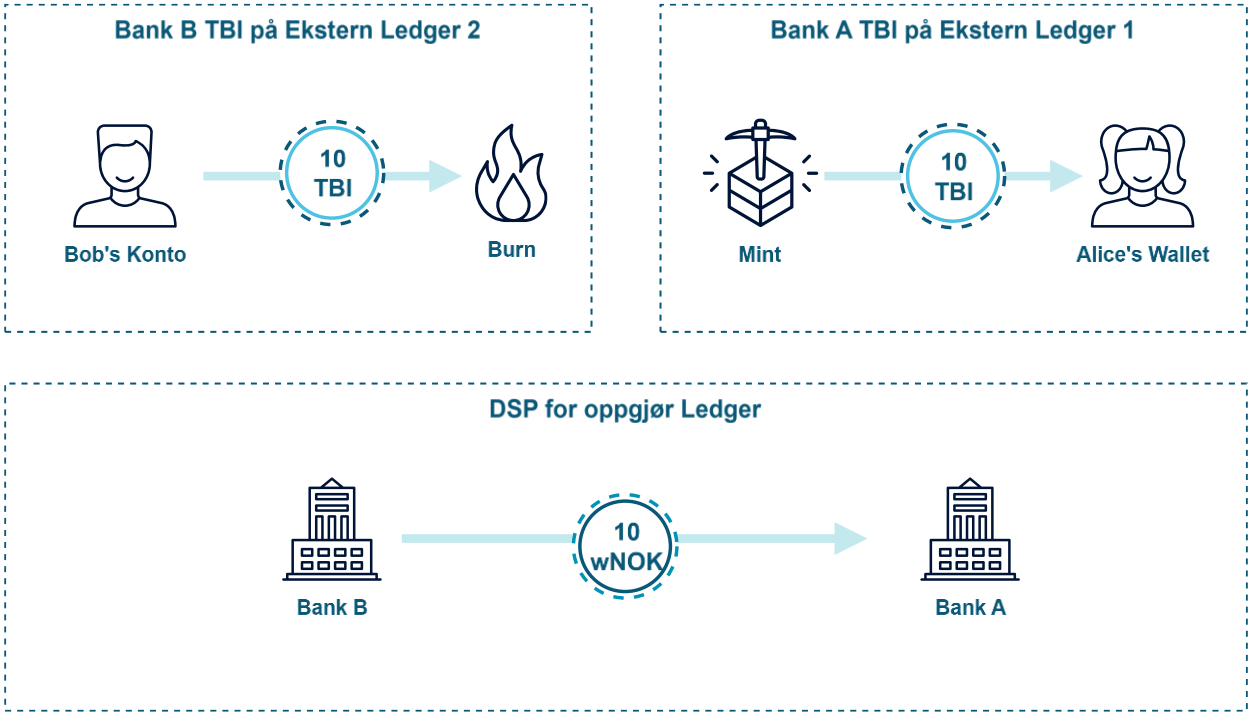
Teknisk skjer følgende:

1. Kunden initierer et uttak ved å be banken om å konvertere et gitt beløp TBI til kunderettet DSP.
2. Banken kontrollerer at kunden har tilstrekkelig TBI-saldo, og låser deretter det aktuelle TBI-beløpet på bankens TBI-ledger. Dette gjøres ved hjelp av kjedekoden på TBI-kanalen, som endrer kundens saldo slik at beløpet ikke kan brukes i andre transaksjoner mens uttaksprosessen pågår.
3. Banken initierer deretter en oppgjørstransaksjon på DSP for oppgjør-ledgeren. Banken reduserer sine reserver i DSP for oppgjør med samme beløp som kunden skal motta i kunderettet DSP. Denne reduksjonen skjer gjennom en transaksjon fra bankens konto til en brostyrt konto, som inngår i bromekanismen til kunderettet DSP-ledgeren.
4. Broen mellom DSP for oppgjør og kunderettet DSP-ledgeren låser DSP for oppgjør (wNOK) og utsteder DSP på kunderettet DSP-ledgeren. Kunderettet DSP krediteres først bankens konto på den kunderettede ledgeren.
5. Banken overfører til slutt det aktuelle beløpet i kunderettet DSP videre til kundens adresse på kunderettet ledger. Etter dette sitter kunden med en direkte fordring på sentralbanken i form av kunderettet DSP, og tilsvarende TBI-beløp er permanent redusert på bankens TBI-ledger.

På balansenivå innebærer dette, som angitt i Figur 7, at bankens innskudd (TBI) reduseres, og bankens reserver (DSP for oppgjør) reduseres tilsvarende. På sentralbankens balanse reduseres bankens reserver, mens beholdningen av kunderettet DSP øker. Løsningen sikrer at det alltid er en én til- én--sammenheng mellom reduksjonen i TBI og økningen i kunderettet DSP, og at ingen nye penger skapes i prosessen.

Interbankoppgjør for TBI

Figur 8 viser hvordan interbankoppgjør for TBI er modellert i sandkassen. Scenariet er at kunder overfører TBI mellom hverandre på tvers av banker, for eksempel fra en kunde i Bank B til en kunde i Bank A. I dag skjer et tilsvarende oppgjør ved at bankenes innskudd (kundekonter) oppdateres i hver sin interne ledger, mens bankenes innbyrdes posisjon gjøres opp i sentralbankreserver. I testcase 4 gjøres de samme overføringene som en integrert prosess.



Figur 8: Interbankoppgjøret for TBI

Teknisk består en slik kredittoverføring av tre hovedoperasjoner på tre ulike ledgere:

1. **På Ledger 2 (TBI-ledgeren til Bank B)**: Bank B reduserer kundens TBI-saldo med beløpet som skal sendes. I sandkasseimplementasjonen skjer dette ved at TBI-tokens destrueres eller låses fra kundens konto, med en referanse til en unik oppgjørs-ID.

2. **På Ledger 1 (TBI-ledgeren til Bank A)**: Bank A øker mottakerens TBI-saldo med tilsvarende beløp. Dette kan implementeres ved at nye TBI-tokens utstedes på mottakerens konto, knyttet til samme oppgjørs-ID.
3. **På DSP for oppgjør-ledgeren**: Det gjennomføres et oppgjør i DSP for oppgjør mellom Bank A og Bank B. Bank B får redusert sine reserver i DSP for oppgjør, mens Bank A får økt sine reserver med samme beløp i en integrert operasjon.

Disse tre operasjonene kobles sammen gjennom en oppgjørsprosess, eller i form av en orkestrert protokoll på DSP for oppgjør-ledgeren. Protokollen sørger for at det finnes en enkel sannhetskilde[28] for selve oppgjøret i DSP for oppgjør, ved at den lagrer en oppgjørstransaksjon med referanser til de to bankenes TBI-transaksjoner (for eksempel via unike transaksjons-ID-er). Dersom noe feiler før selve DSP-transaksjonen på DSP for oppgjør-ledgeren er fullført, kan prosessen ikke fullføres. Dersom DSP-transaksjonen er gjennomført, regnes oppgjøret som endelig, og bankene må sørge for at deres TBI-ledger gjenspeiler dette.

I Figur 8 uttrykkes dette også på balanseform: Bank B får redusert «innskudd» (TBI) og tilhørende reduksjon i reserver i DSP for oppgjør. Bank A får økt innskudd (TBI) og en tilsvarende økning i reserver i DSP for oppgjør. På sentralbankens balanse flyttes reserver fra Bank B til Bank A, mens den totale mengden DSP for oppgjør og TBI i systemet som helhet er uendret.

Løsningen er bevisst utformet slik at logikken for interbankoppgjør ligger på DSP for oppgjør-ledgeren, ikke på hver enkelt TBI-ledger. Dette gjør at selve oppgjørsprosessen kan standardiseres og kontrolleres sentralt, samtidig som banker kan ha stor grad av frihet til å velge teknologi og struktur for egne TBI-ledgere, så lenge de kan referere til en unik oppgjørs-ID og utføre utstedelse/destruksjon eller tilsvarende operasjoner i samsvar med oppgjørsresultatet.

Resultater og funn

I et multi-ledger oppsett skjer oppgjør på tvers av flere uavhengige systemer. Dette innebærer at transaksjoner må koordineres utenfor selve ledgerne, noe som kan gi mellomtilstander og lavere grad av samtidighet enn atomisk oppgjør i én felles ledger. Slike løsninger krever derfor ekstra mekanismer for koordinering og feilhåndtering. Når involverte aktører benytter separate ledgere, kan ikke smartkontrakter styre operasjoner på tvers direkte. Koordinering må skje gjennom eksterne mekanismer, som orkestrering eller broer,

[28] Sannhetskilde: I IT--sammenheng kjent som *single source of truth (SSOT)*, og viser til én autoritativ kilde som avgjør endelige informasjonssannhet. I dette oppsettet er oppgjøret i DSP for oppgjør på DSP for oppgjør ledgeren den avgjørende sannhetskilden.

noe som øker teknisk og operasjonell kompleksitet sammenlignet med en single-ledger arkitektur.

Testen viser at når hver aktør har sin egen ledger, reduseres flere av de sentrale gevinstene ved blokkjedeteknologien. Dette gjelder særlig mulighetene for sammenhengende automatisering, atomisk oppgjør, og tett integrasjon mellom penger og forretningslogikk, ettersom slike egenskaper i stor grad forutsetter at aktiva og logikk håndteres på samme ledger. I multi-ledger oppsett må disse sammenhengene etableres gjennom ekstern orkestrering og meldingsutveksling. Dette kan sammenlignes med måten det håndteres på i dagens betalingssystemer, noe som reiser spørsmål om forbedringer i dagens systemer er et bedre alternative enn å implementere en DSP basert på ledger-teknologi.

Testcase 5 – Oppgjør av tokeniserte verdipapirer på en felles ledger

Formålet med testcaset

Formålet med dette testcaset var å undersøke om handel og oppgjør av tokeniserte verdipapirer ga vesentlige gevinster når alle relevante aktiva og aktører samles på én felles ledger, sammenlignet med løsningene som var testet i en multi-ledger-arkitektur. Testen tok utgangspunkt i erfaringene fra tidligere testcases, som viste at transaksjoner på tvers av ledgere reduserer muligheten til å dra nytte av alle fordelene ved ledger-teknologi og DSP, særlig knyttet til atomisitet, automatisering og effektiv risikoreduksjon.

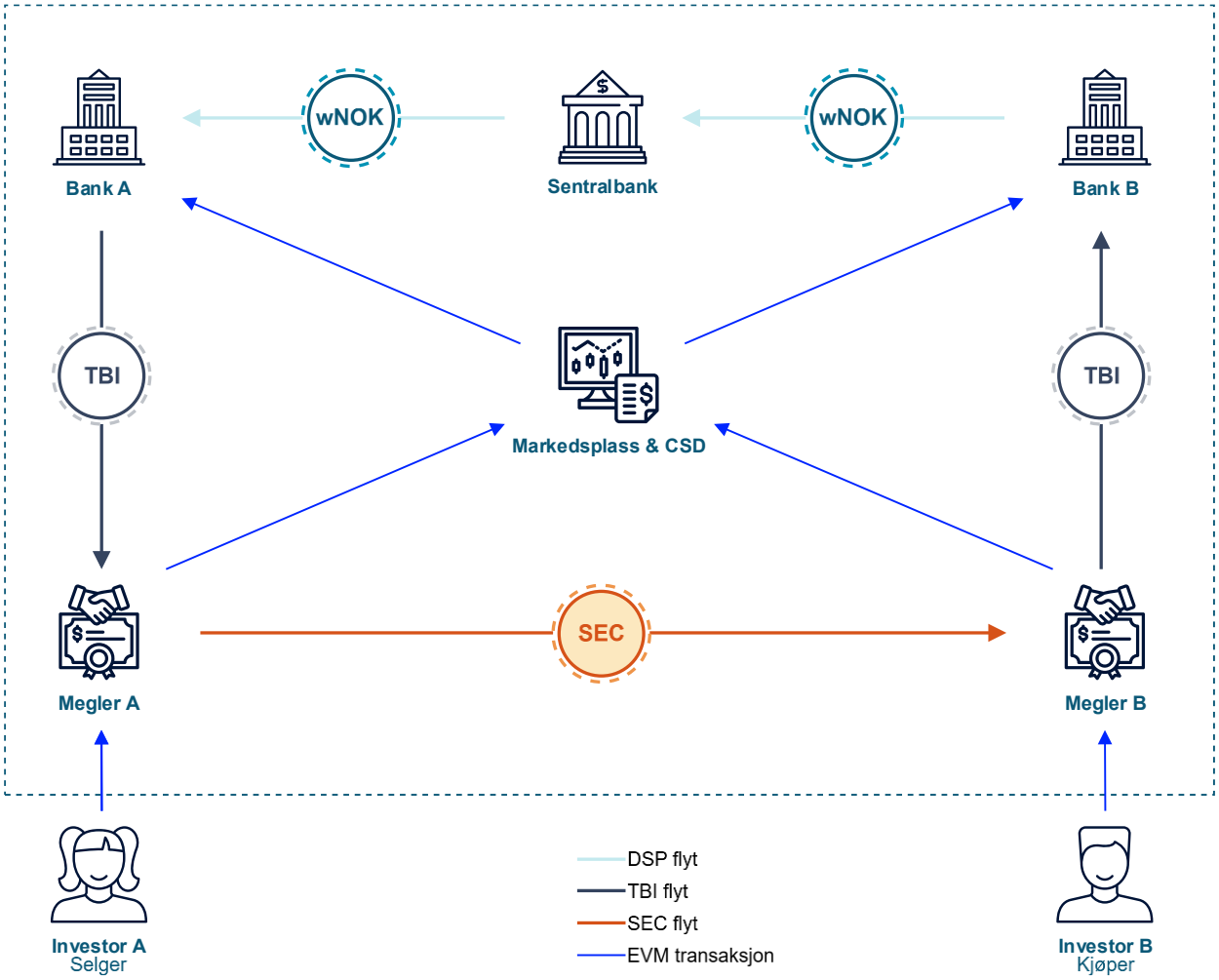
Testcaset er relevant fordi samtidig overføring av eierskap til flere verdier på samme ledger både gir atomisitet og reduserer oppgjørs- og motpartsrisiko. I motsetning til bro- og multi-ledger-løsninger, der oppgjør skjer som koordinerte sekvensielle prosesser, gir én felles ledger mulighet for at alle endringer skjer i én og samme transaksjon. Dette gir sanntidsoppgjør, høy sporbarhet og en enkelt sannhetskilde for både penger og verdipapirer.

Testen hadde også som formål å belyse hvilke nye muligheter for automatisering og effektivitet som oppstår når finansielle prosesser, inkludert ordrebehandling, oppgjør, rollevalidering og regeletterlevelse, implementeres direkte som smarte kontrakter på én ledger. Dette inkluderer muligheten for å etterligne netting-lignende effekter i et bruttosystem, redusere likviditetsbehov og fjerne behovet for broer og ekstern orkestrering mellom separate systemer.

Samtidig skulle testcaset synliggjøre hvilke nye utfordringer en single-ledger-modell medførte. Når alle aktører måtte forholde seg til én felles ledger, kunne dette oppleves som mer inngripende enn løsninger der hver aktør hadde sin egen ledger. Spørsmål knyttet til

styring, endringshåndtering, og ansvarsdeling var derfor en viktig del av vurderingsgrunnlaget.

Teknisk design



Figur 9: Oversikt av aktører og transaksjonsflyt på single-ledgeren. DSP for oppgjør er notert som wNOK og verdipapir er notert som SEC.

I denne testen opererer sentralbanken, bankene, meglerne, markedsplassen og investorene på én felles digital ledger. Sentralbanken utsteder DSP for oppgjør som benyttes som oppgjørsmiddel mellom bankene, mens hver bank utsteder TBI til egne kunder. Når to investorer handler et verdipapir via sine meglere, registreres ordren på en felles ordrebok på ledgeren.

Når en kjøps- og salgsordre matches i den felles ordreboken, initieres oppgjøret automatisk. Kjøpers bank belaster kundens TBI, og gjør samtidig opp handelen i DSP for oppgjør ovenfor selgers bank som utsteder nye TBI til selger. Parallelt overføres eierskapet til verdipapiret fra selger til kjøper. Kunden har ikke direkte tilgang til DSP for oppgjør;

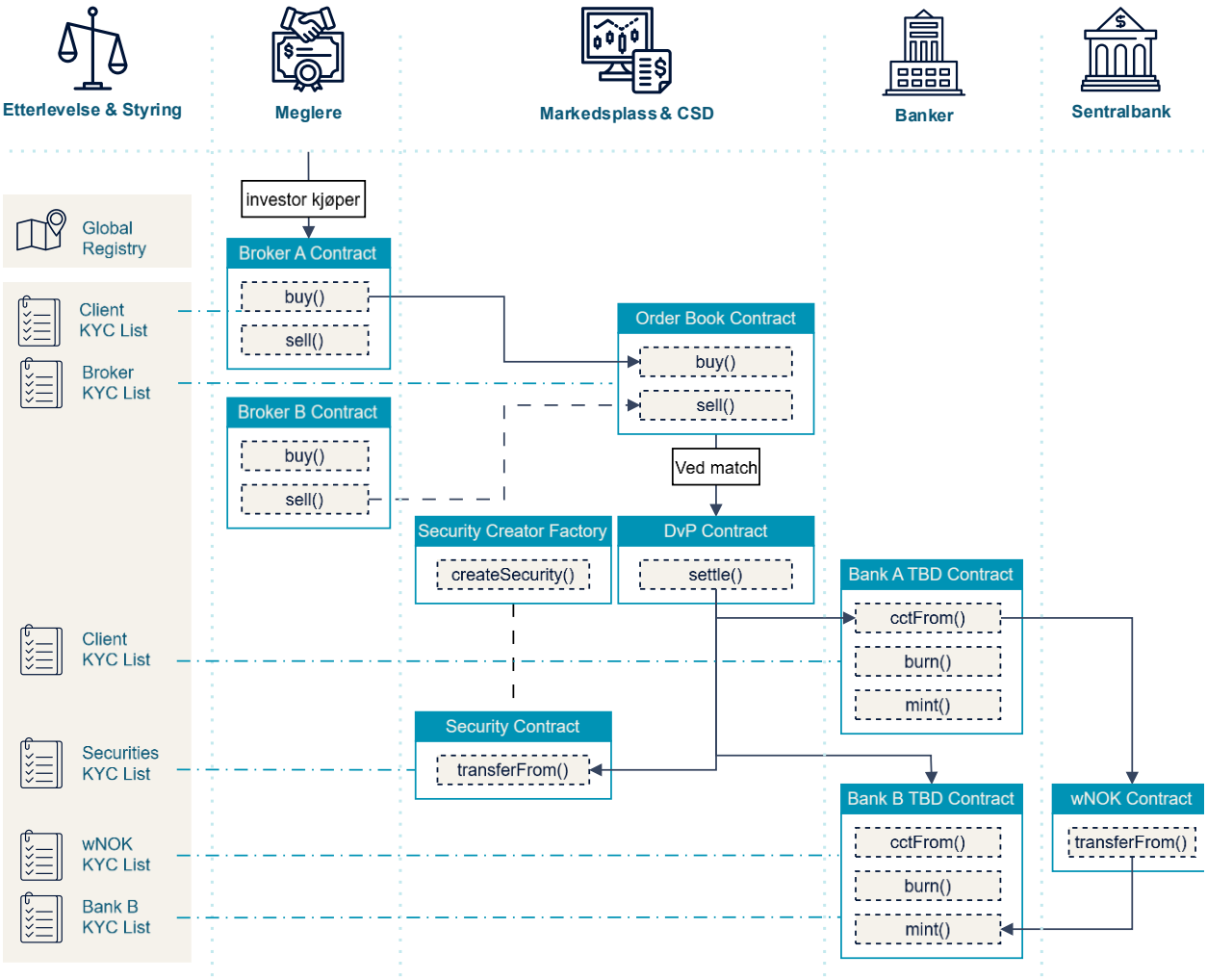
oppgjøret i sentralbankpenger skjer utelukkende mellom bankene. Oppgjøret er implementert som en atomisk DvP-prosess, der penge- og verdipapirbenet gjennomføres enten samlet med endelig virkning, eller det gjennomføres ikke i det hele tatt dersom ett steg i prosessen feiler. Verdipapiret skifter dermed kun eier dersom betalingen samtidig gjennomføres.

Om atomisitet

Atomisitet foreligger når alle relevante tilstandsendringer i et oppgjør, for eksempel overføring av penger og verdipapirer, gjennomføres som én udelelig teknisk simultan operasjon, typisk på én ledger eller innenfor samme system. Transaksjonen får kun én endelig tilstand: fullført eller ikke-gjennomført. Det oppstår ingen mellomtilstander der bare deler av oppgjøret er gjennomført.

I koordinerte oppgjør, som i multi-ledger-arkitekturer eller ved bruk av broer, gjennomføres oppgjøret som en sekvens av avhengige operasjoner på tvers av systemer. Slike løsninger kan være funksjonelt sammenkoblede, men oppnår ikke samme tekniske samtidighet som ved atomisitet, og krever tilleggsorkestrering og -feilhåndtering.

Tilgang til systemet er begrenset til godkjente aktører, og adresser og rettigheter valideres mot et felles register. Arkitekturen legger dermed til rette for at handel, betaling og registerføring kan gjennomføres innenfor én og samme tekniske plattform, med konsistent sporbarhet og uten behov for ekstern avstemming i selve oppgjørsøyeblikket.



Figur 10: Oversikt av hvordan smartkontrakene som konstituerer systemet for handel av tokeniserte verdipapirer samhandler.

I testoppsettet ligger en *Global Registry* som én kilde til sannhet for kontraktadresser og «whitelister» (godkjente banker, meglere og kunder). En investor legger ordre inn via Broker-kontrakten, som først sjekker at kunden er godkjent og at megleren er autorisert. Ordren sendes videre til «*Order Book*»-kontrakten, som matcher kjøp og salg og, ved treff, initierer oppgjør ved å kalle *DvP-kontrakten*.

DvP-kontrakten sørger for et atomisk oppgjør der begge bena gjennomføres samtidig:

- **Verdipapirbenet:** Verdipapiret (utstedt fra *Security Creator Factory* og representert i *Security-kontrakten*) låses hos selger og klargjøres for overføring.
- **Pengebenet:** Kjøpers bank gjør opp handelen i DSP for oppgjør overfor selgers bank, kjøpers banks utlegg dekkes ved å redusere kundens TBI-saldo-, slik at kundens

fordring på banken slettes og tilsvarende TBI utstedes hos mottakende bank. Kunden mottar aldri DSP for oppgjør; oppgjøret i sentralbankpenger skjer utelukkende mellom bankene.

Oppgjøret i begge ben fullføres atomisk i samme transaksjon; feiler en kontroll (på grunn av f.eks. manglende autorisasjon, utilstrekkelig saldo, ugyldig motpart eller feil versjon i *Registry mv.*), stoppes alt, slik at hverken overføringen av eierskap til verdipapirer, eller betalingen gjennomføres Dersom alle kontrollene lykkes, kaller *DvP-kontrakten transferFrom()* i *Security*-kontrakten slik at eierskapet registreres på kjøper, og oppgjøret fullføres.

«*Factory*»-mønsteret benyttes for å utstede nye verdipapirer på en kontrollert måte (deterministisk adressering via *Registry* og innebygde whitelister), og speiler hvordan en produksjonsprosess kan se ut. Selv om full AML/KYC/CTF er utenfor scope, håndhever testen en enkel variant: ukjente kunder eller uautoriserte meglere stoppes on-chain før ordre aksepteres.

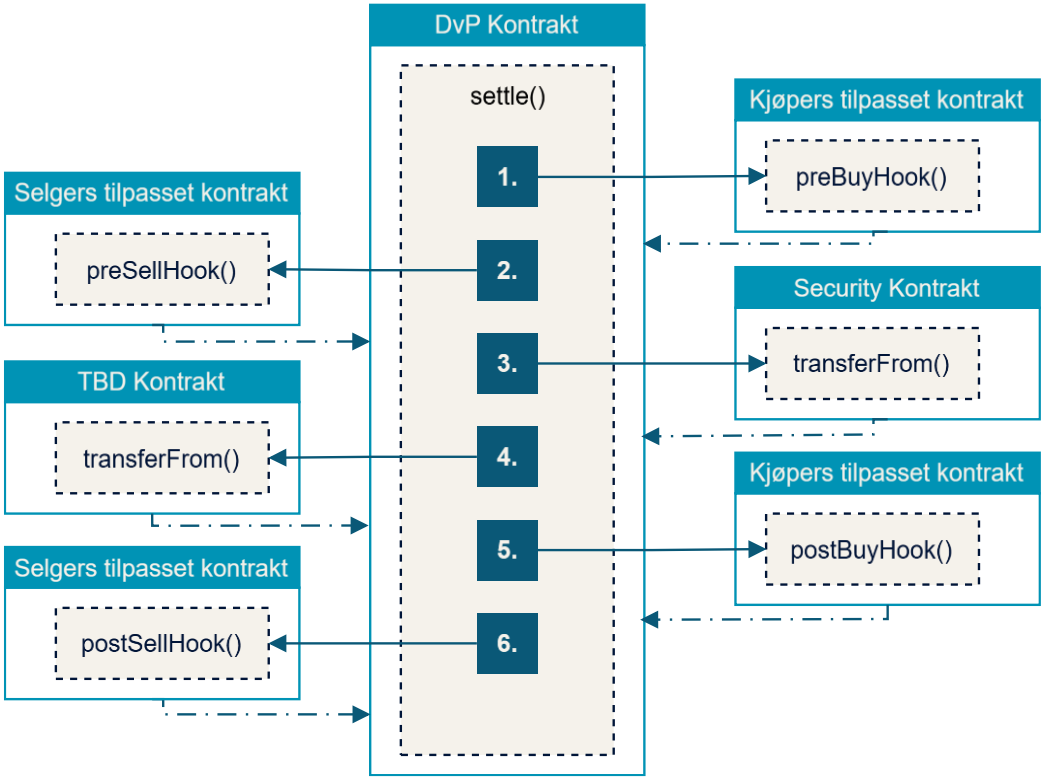
Nettinglignende effekter i en smartkontraktbasert oppgjørsmodell

I tradisjonelle verdipapir- og betalingssystemer oppnås netting ved å samle flere handler gjennom en tidsperiode i satsvise oppgjør (batcher) og det gjøres kun opp nettoposisjoner mellom deltakerne på fastsatte tidspunkter. Dette reduserer likviditetsbehovet, men kan samtidig innebære mellomliggende motpartsrisiko, operasjonell kompleksitet og behov for clearing og garantimekanismer.

I den smartkontraktbaserte modellen som er implementert her, finnes det ikke et slikt utsatt oppgjørstrinn. Alle handler gjøres opp umiddelbart og atomisk gjennom *DvP-kontrakten*. Samtidig kan modellen gi ***nettinglignende effekter***, særlig i form av redusert likviditetsbehov, gjennom måten flere transaksjoner og operasjoner kan kobles sammen og gjennomføres samlet i én og samme atomiske prosess.

Dette muliggjøres ved at DvP-oppgjøret- kan omkranses av forhånds og etterfølgende smartkontraktkall («pre- og posthooks»), som utføres som del av den samme transaksjonen. Disse omkransende kallene er vist under i Figur 11. Slike kall kan for eksempel brukes til å skaffe nødvendig likviditet, gjennomføre tilknyttede handler, eller rydde opp og konsolidere posisjoner etter oppgjøret. Flere handler, betalinger og eventuelle likviditetstrinn kan dermed kjedes sammen slik at kun den samlede nettoeffekten materialiseres på deltakerens saldo, uten at midler må stilles brutto på forhånd. Resultatet er at økonomiske effekter som tradisjonelt forbindes med netting, særlig lavere likviditetsbinding, kan oppnås uten avregninger med utsatt oppgjør, batchbehandling eller mulighet for motpartsrisiko. Det er derfor mer presist å- omtale dette som nettinglignende

effekter, snarere enn netting i klassisk forstand. Oppgjøret forblir umiddelbart og endelig, men selve oppgjørsprosessen kan internt være sammensatt av flere koordinerte trinn som enten lykkes samlet eller ikke gjennomføres i det hele tatt.



Figur 11: Forbedret DvP kontrakt, der fire eksterne smartkontraktfunksjoner – preBuyHook(), preSellHook(), postBuyHook() og postSellHook() – kalles som del av samme transaksjon. Funksjonskallene omkranser den faktiske DvP prosessen, inkludert overføring av verdipapirer og tokeniserte bankinnskudd (inkludert DSP)

Resultater og funn

Testen viser at når tokeniserte penger og verdipapirer håndteres på én felles ledger, kan oppgjøret gjennomføres atomisk: enten gjennomføres både penge og verdipapirbenet simultant og med endelig virkning, eller så fullføres ikke transaksjonen dersom én del feiler. Den testede single-ledger-arkitekturen oppfyller dermed sentrale krav til sammenhengende og sikkert oppgjør, teknisk håndheving av roller og tilgang, samt sporbarhet og etterlevelse. Sammenlignet med multi-ledger-løsninger reduseres behovet for mellomledd som broer og meldingsmekanismer, og den tekniske kompleksiteten blir lavere. Samtidig viser erfaringene fra testen at flere forhold må klargjøres. Før en slik løsning kan tas i bruk, kreves videre testing av robusthet under reelle lastforhold, cybersikkerhet, beredskap og gjenoppretting ved feil, samt interoperabilitet med utenlandske og multinasjonale systemer. Kravet om at alle aktører må benytte én felles ledger reiser også spørsmål om fleksibilitet, insentiver og vilje til adopsjon, særlig for aktører med særskilte regulatoriske eller internasjonale behov.

Testen viser videre at bruk av smarte kontrakter på en felles ledger gjør det mulig å håndheve oppgjørsregler, roller og tilgang direkte i infrastrukturen, uten behov for ekstern orkestrering eller manuelle kontroll ledd. Ansvar for infrastrukturen peker mot en delt modell, der sentralbanken har ansvar for DSP og prinsipper for sikkert oppgjør, mens banker og andre aktører forvalter tokens som representerer bankinnskudd, verdipapirer og andre verdier. Selve driften av ledgeren og grunninfrastrukturen kan muligens egne seg for felles organisering, for eksempel gjennom et felles eid selskap.

Testcase 5 har ikke hatt som mål å undersøke hvordan hensyn til personvern er ivaretatt. Derimot har testen belyst strukturelle begrensninger ved single-ledger arkitekturer basert på EVM-kompatible plattformer, som Hyperledger Besu. Personvernet baserer seg i hovedsak på pseudonymitet, noe som ikke gir tilstrekkelig beskyttelse dersom uautoriserte likevel klarer å knytte adresser til identiteter. Rolle og tilgangsstyring i smarte kontrakter begrenser hvilke handlinger aktører kan utføre, men hindrer ikke analyse av transaksjonsmønstre på ledger nivå. Erfaringer fra andre testcaser indikerer at alternative DLT-plattformer, som Hyperledger Fabric, har et bedre teknisk utgangspunkt for konfidensialitet gjennom mer finmasket innsynsstyring. Ytterligere personverntiltak, som anonymiseringslag eller lagdelte arkitekturer, er ikke undersøkt i denne testen.

Testcase 6 – Utstedelse av statsobligasjoner på en felles ledger

Formålet med testcaset

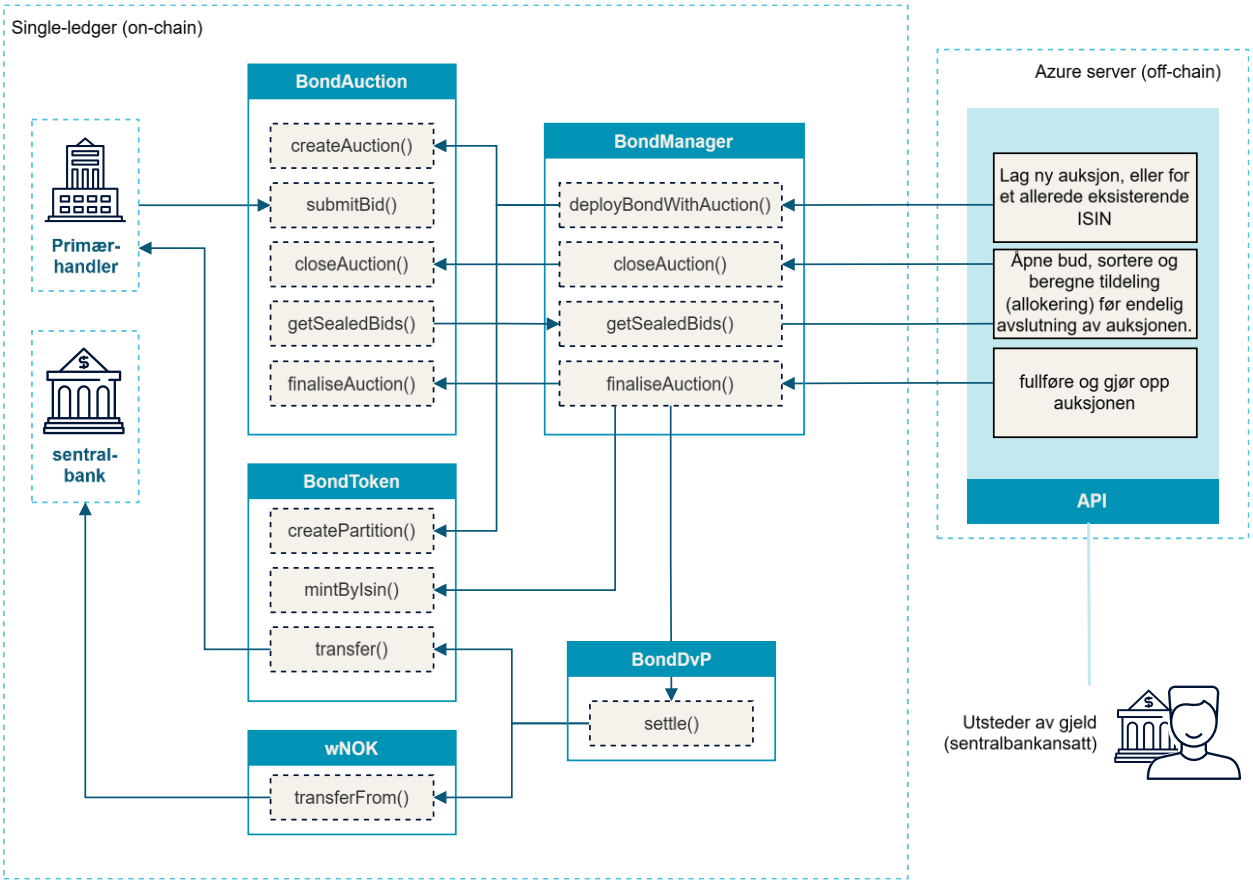
Formålet med denne testen var å finne ut om hele livssyklusen for statsobligasjoner kunne tokeniseres og registreres på én felles ledger og utføres ved hjelp av smartkontrakter. Hele transaksjonskjeden fra utstedelse via en auksjon, handel i sekundærmarkedet, kupongutbetalinger, tilbakekjøp og innfrielse ved forfall ble håndtert gjennom smartkontrakter. Statsobligasjoner, tokeniserte bankinnskudd og DSP for oppgjør ble representert som egne tokens. Oppgjøret skulle gjennomføres atomisk – slik at enten både oppgjøret av penger og verdipapirer ble gjennomført samtidig, eller at ingen av eiendomsoverføringene ble gjennomført.

Tokeniserte verdipapirer pekes ofte ut som et usecase med stort potensial. Formålet med denne testen var å utforske:

1. Om man enkelt kan utvide ledgeren til å håndtere utstedelse og livssyklus til statsobligasjoner.
2. Hva slags fordeler og ulemper ga tokeniserte verdipapir til utstedere og handlere.

Teknisk design

Løsningen er utformet som en hybrid modell der deler av prosessen håndteres på ledgere (on-chain) ved hjelp av smartkontrakter, mens enkelte operasjoner gjennomføres utenfor ledgeren (off-chain) av praktiske og regulatoriske hensyn. Dette gjelder særlig bud åpning og tildeling av verdipapirer i auksjonsprosessen, som ikke er fullt automatisert på ledgeren.



Figur 12: Oversiktsbilde av smartkontrakts-arkitektur for tokeniserte statsobligasjoner

Figur 12 viser et oversiktsbilde av hvordan smartkontraktsystemet for tokeniserte statsobligasjoner er implementert i sandkassen. Hele systemet styres av en sentral komponent, «*BondManager*», som håndterer initiering av nye auksjoner, stenging for nye bud ved budfristen, tildeling av obligasjoner, og til slutt oppgjør av penge- og verdipapirsiden. Denne kontrakten kaller på andre, mer spesialiserte kontrakter, som håndterer f.eks oppgjør.

Oversiktsbildet i Figur 12 viser fire hovedprosesser i auksjonsløpet for statsobligasjoner. Dette omfatter etablering av auksjonen, mottak og registrering av bud, stenging og avklaring av auksjonen (herunder beslutning om hvilke bud som aksepteres), samt

gjennomføring av tildeling, utstedelse av obligasjonstokens og tilhørende oppgjør. I tillegg finnes det en mekanisme for utvidelse av et eksisterende lån (ISIN), men denne er ikke vist i figuren.

Auksjonsprosessen starter med at sentralbanken oppretter en ny auksjon gjennom et API-grensesnitt. Denne delen starter utenfor ledgeren. API-grensesnittet aktiverer så de relevante smartkontraktene, som oppretter både nye obligasjonstokens og selve auksjonen på ledgeren. Når dette er på plass, er auksjonen åpnet for bud fra primærhandlerne.

I budgivningsfasen forsegler primærhandlerne budene sine ved hjelp av både sentralbankens og egne krypteringsnøkler. Dette skjer utenfor ledgeren for å sikre konfidensialitet, samtidig som budene senere kan verifiseres. De forseglede budene sendes så inn til auksjonen på ledgeren.

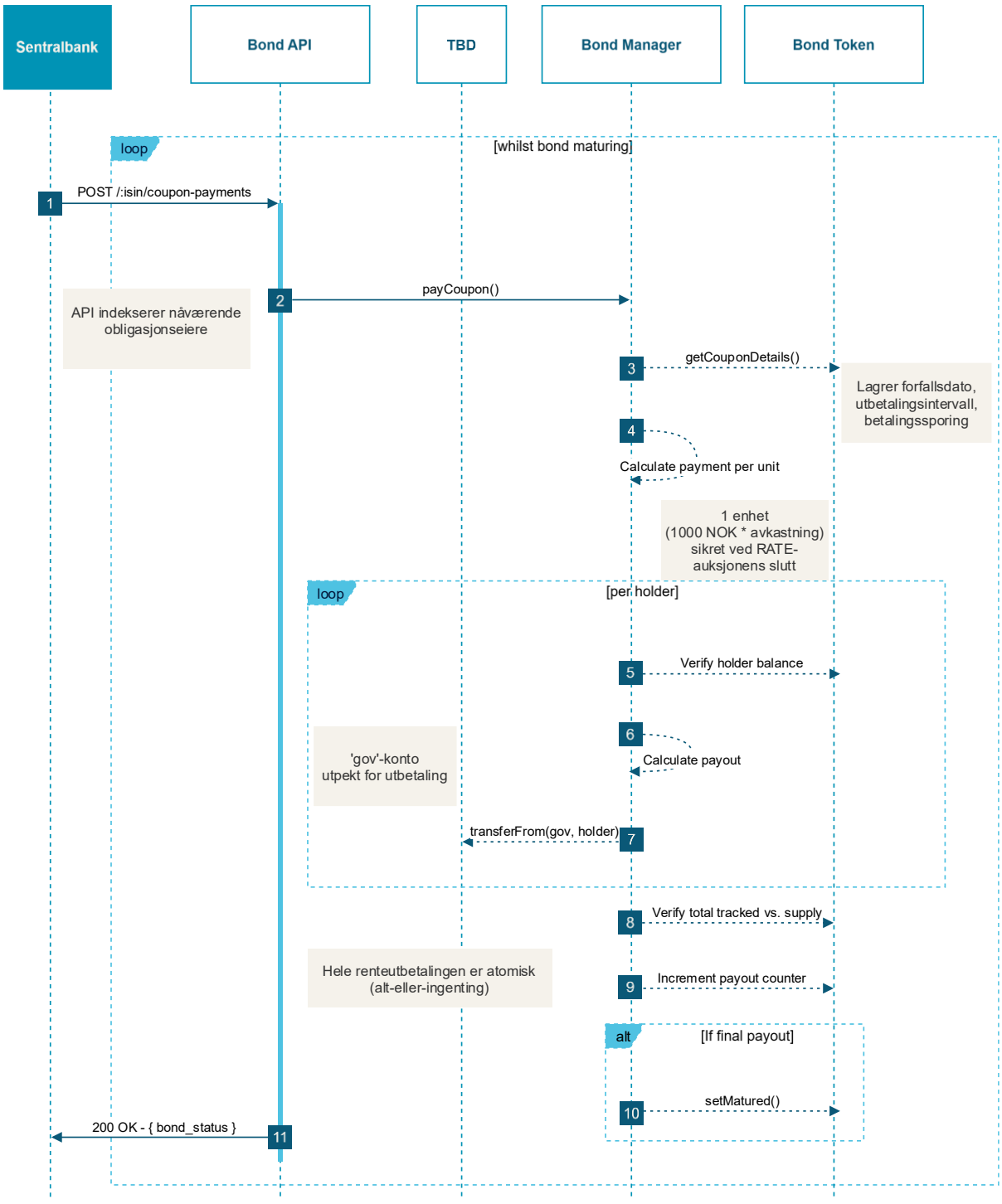
Når budfristen passeres, lukker sentralbanken auksjonen gjennom API-grensesnittet. APIet utløser da smartkontrakten «*BondAuction*», som stenger auksjonen på ledgeren og hindrer at nye bud kan mottas. Etter at auksjonen er lukket, åpner sentralbanken de forseglede budene gjennom API-grensesnittet og verifiserer innholdet. Denne delen av prosessen foregår også utenfor ledgeren.

I den avsluttende fasen vurderer sentralbanken om budene skal aksepteres. Via API-grensesnittet kan auksjonen enten fullføres eller ikke dersom budene ikke skal godtas. Resultatet av auksjonen publiseres som del av denne prosessen. Når auksjonen sluttføres, aktiverer API-grensesnittet smartkontrakten «*BondManager*», som setter i gang oppgjøret og tildelingen av obligasjonstokenene gjennom de underliggende smartkontraktene på ledgeren.

Til slutt oppretter API-grensesnittet en tidsstyrt arbeidsflyt utenfor ledgeren. Denne arbeidsflyten aktiveres automatisk når det er tid for kupongutbetaling på obligasjonen.

Utvidelse av eksisterende obligasjon:

Dersom sentralbanken ønsker å øke volumet i et eksisterende lån (ISIN), følges nesten samme rutine som i opprettelse av auksjon. Utvidelsen legges ut som auksjon og følger auksjonsprosessen nevnt ovenfor. En viktig forskjell er at ISIN beholdes og kupongrenten er uendret.



Figur 13 Prosessdiagram for utstedelse og auksjon av tokeniserte obligasjoner

Kupongutbetalinger

For å gjennomføre kupongutbetalinger og utbetaling ved forfall må smartkontraktene få tilgang til eksternt genererte oversikter over obligasjonseierne. Årsaken er at kontraktene

ikke selv identifiserer alle adressene som til enhver tid har krav på utbetaling. Disse eierlistene produseres derfor utenfor ledgeren og brukes som grunnlag for de operasjonene som deretter utføres på ledgeren.

Modellen i Figur 14 er en forenklet fremstilling av dagens obligasjonssyklus. Kupongutbetalingene gjentas med faste intervaller frem til obligasjonen når forfallsdato.

Når tidspunktet for en kupongutbetaling er kommet, starter prosessen med at sentralbanken kaller *BondAPI* for den aktuelle obligasjonen, identifisert ved ISIN. APIet aktiverer deretter smartkontraktfunksjonen *payCoupon()* i *BondManager*. Samtidig sørger APIet for at det foreligger en oppdatert oversikt over hvem som eier obligasjonen på dette tidspunktet, slik at utbetalingen går til riktige mottakere.

BondToken inneholder sentrale opplysninger om obligasjonen, blant annet forfallsdato, intervall mellom kupongutbetalingene og en teller som viser hvor mange kuponger som allerede er utbetalt. *BondManager* henter disse opplysningene og beregner deretter hvor stor kupongen skal være per enhet. En enhet er her knyttet til et nominelt beløp, for eksempel 1 000 kroner, og kupongrenten er den som ble fastsatt ved den opprinnelige renteauksjonen.

Deretter går *BondManager* gjennom alle registrerte obligasjonseiere. For hver eier kontrolleres beholdningen i *BondToken*, og den individuelle kupongutbetalingen beregnes ut fra hvor mange enheter vedkommende eier. Når beløpet er fastsatt, instrueres en overføring fra en dedikert «gov»-konto (representert ved TBD i figuren) til den aktuelle obligasjonseieren. På denne måten mottar hver eier kupongutbetalingen som svarer til egen beholdning.

Hele kupongutbetalingen er definert som atomisk. Det betyr at enten gjennomføres alle utbetalingene samlet, eller så blir hele operasjonen tilbakeført dersom noe feiler. Dette reduserer risikoen for delvis eller inkonsistente utbetalinger.

Når alle eiere har mottatt kupongutbetaling, kontrollerer *BondManager* mot *BondToken* at samlet utbetalt kupong samsvarer med den totale obligasjonsbeholdningen i systemet. Deretter oppdateres telleren i *BondToken*, slik at systemet registrerer at den aktuelle kupongperioden er fullført. Dersom dette var siste kupongutbetaling i henhold til obligasjonens struktur, kaller *BondManager* en funksjon i *BondToken* som markerer obligasjonen som forfalt («*matured*»).

Til slutt sender APIet en statusmelding tilbake til sentralbanken. Denne bekrefter hvor i løpet obligasjonen befinner seg, og om den nå har nådd forfall.

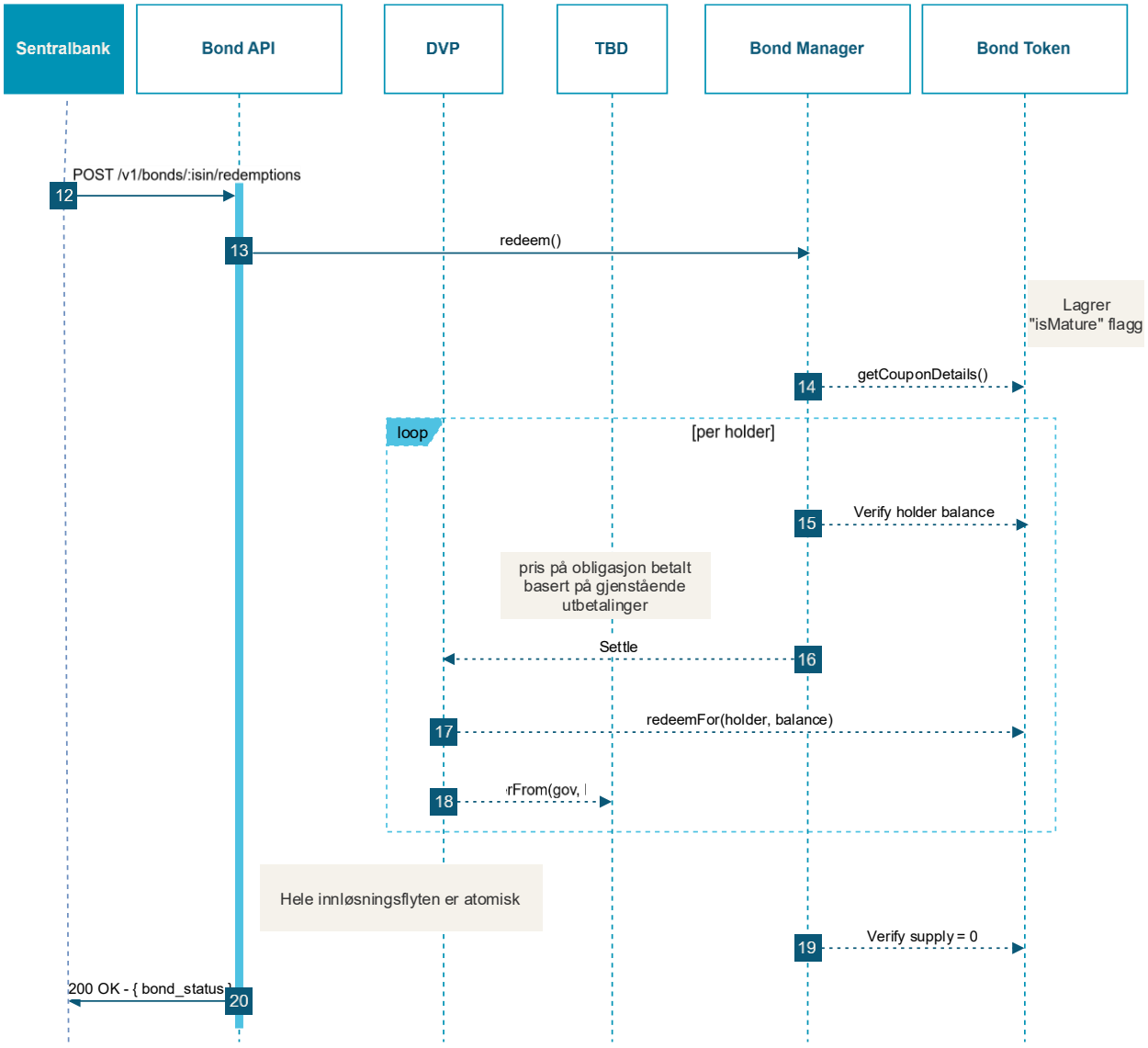
Innløsning

Når obligasjonen har nådd forfall og alle kuponger er utbetalt, starter sentralbanken innløsningsfasen, jf. Figur 14. Sentralbanken kaller *BondAPI* mot et eget endepunkt for innløsning, som igjen aktiverer *BondManager*-funksjonen *redeem()*. *BondToken* opprettholder en status (*isMature*) som viser at obligasjonen har nådd forfall, og denne statusen benyttes av *BondManager* som grunnlag for innløsningen.

BondManager henter deretter relevante opplysninger fra *BondToken*, blant annet det som trengs for å fastsette hva obligasjonseierne skal få utbetalt ved innløsning. I teorien kan beløpet beregnes ut fra hvilke betalinger som gjenstår på obligasjonen. Når obligasjonen har nådd forfall, skal hovedstolen utbetales, eventuelt sammen med utbetaling av den siste kupongbetalingen.

Videre går *BondManager* gjennom alle gjenværende obligasjonseiere og verifiserer hver enkelt beholdning i *BondToken*. Selve innløsningsoppgjøret gjennomføres via *BondDvP*-komponenten, slik at betaling og sletting av obligasjonstokens skjer samtidig. *BondManager* instruerer *BondDvP* om å gjennomføre oppgjøret, hvorpå *BondDvP* kaller *BondToken* for å innløse eierens beholdning gjennom *redeemFor(holder, balance)*. I praksis innebærer dette at obligasjonstokenene fjernes fra eierens adresse. Samtidig overføres oppgjørsbeløpet fra «gov»-kontoen (TBD) til den samme eieren.

Innløsningsflyten er atomisk. Når utbetalingsprosessen er gjennomført, verifiserer *BondManager* at alle obligasjonstokens er destruerte. Obligasjonen er da fullt innløst og avsluttet i systemet. *BondAPI* sender deretter en statusrapport til sentralbanken som bekrefter at innløsningen er gjennomført.



Figur 14: Prosessdiagram for kupongutbetaling til eiere av tokeniserte statsobligasjoner

Resultater og funn

Testen viser at en ledger kan benyttes til å håndtere store deler av livsløpet til statsobligasjoner, fra utstedelse og handel til kupongutbetalinger, tilbakekjøp og innløsning ved forfall. Viktige tilstandsendringer og oppgjør kan automatiseres gjennom smartkontrakter, noe som gir god sporbarhet, kontroll og effektivitet. Samtidig viser testen at enkelte sentrale prosesser fortsatt må håndteres utenfor ledgeren, særlig åpning og fordeling av bud i auksjoner, samt identifisering av gjeldende obligasjonseiere før kupongutbetalinger og innløsning.

Atomisk DvP er implementert i tråd med formålet. Det innebærer at eierskap til verdipapirer og betalinger gjøres opp samtidig, og at hele transaksjonen ikke blir fullført dersom én del feiler. Dette reduserer motpartsrisikoen og fremstår som en styrke ved løsningen. Auksjonsprosessen er gjennomført med krypterte bud («*sealed bids*»), der budene bare kan leses utenfor ledgeren, mens tildelingsbeslutningen registreres på ledgeren med et signert kryptografisk bevis. Dette gir god etterprøvbarhet, men ikke full sikkerhet for at tildelingen automatisk er korrekt i forhold til alle bud.

Løsningen kan styrke integriteten, kontrollen og rettighetsstyringen ved gjennomføring av auksjoner, men enkelte prosesser i testen er ikke fullt automatisert og må støttes av eksterne rutiner (*off-chain*). Konfidensialitet og personvern på ledgeren er forbedret gjennom krypterte bud og streng tilgangsstyring, men enkelte opplysninger, som aktivitetsmønstre, tidspunkt og antall bud, kan fortsatt observeres på ledgeren. Funnene illustrerer dermed behovet for en avveiing mellom hva som bør håndteres på ledgeren, og hva som fortsatt bør løses utenfor.

Oppsummering av funn

Den tekniske testingen har gitt praktisk innsikt i hvordan DSP basert på ledger-teknologi kan brukes i betalings- og oppgjørsprosesser, og hvordan arkitekturvalg i stor grad bestemmer både gevinster, begrensninger og rollen til DSP. På tvers av testcasene fremstår arkitektur som den viktigste faktoren av betydning for hvilke egenskaper som kan realiseres ved bruken av denne teknologien.

Et hovedfunn er de ulike egenskapene single- og multi-ledger arkitekturer har. Når penger, aktiva og forretningslogikk håndteres på én felles ledger, kan oppgjør gjennomføres samtidig og atomisk. Dette muliggjør integrert DvP, reduserer motpartsrisiko og gjør det mulig å håndheve regler, roller og rettigheter direkte i infrastrukturen ved hjelp av smarte kontrakter. I slike oppsett inngår DSP som en funksjonelt integrert del av prosessene.

I multi-ledger arkitekturer, der aktiva og applikasjoner er fordelt på separate ledgere, må derimot sammenhengen mellom transaksjoner etableres gjennom broer, meldingsutveksling eller ekstern orkestrering. Testene viser at dette gir økt teknisk og operasjonell kompleksitet, lavere grad av samtidighet og økt oppgjørsrisiko. Samtidig viser testene at flere av de sentrale gevinstene ved ledger teknologi, særlig knyttet til tett integrasjon mellom penger og forretningslogikk, ikke realiseres i samme grad som i single-ledger-arkitekturer.

Testingen viser videre at når oppgjør og tilhørende logikk samles på én ledger, kan flere operasjoner inngå i én samlet og atomisk prosess. Dette kan for eksempel gi effektivitets- og likviditetsgevinster, blant annet i form av nettinglignende effekter, uten utsatt oppgjør

eller mellomliggende motpartsrisiko. Testene indikerer samtidig at enkelte funksjoner og prosesser ikke egner seg for å utføres direkte på ledgeren, og at en hensiktsmessig arkitektur derfor vil innebære en bevisst kombinasjon av onchain- og offchainkomponenter.

Testene har i hovedsak vært rettet mot teknisk gjennomførbarhet og utforskning av konseptuelle egenskaper. Forhold som ytelse under last, håndtering av feil og avvik utenfor «happy path», operasjonell drift over tid, sikkerhet, beredskap og samhandling mellom et større antall aktører er i liten grad belyst. Videre arbeid bør også adressere hvordan krav til anonymitet og personvern kan ivaretas bedre, for eksempel gjennom lagdelte arkitekturer eller bruk av sekundærlag («*layer 2*») over en kjerneledger. Disse forholdene bør derfor være et fokus ved videre testing, sammen med konkrete anvendelser der DSP og ledger-teknologi kan gi merverdi utover å replisere dagens strukturer. Dette kan inkludere testing av samhandling i bredere økosystemer, vurdering av insentiver og styringsmodeller, og utforsking av arkitekturer som muliggjør reell integrasjon mellom penger, aktiva og prosesser.

Publisering av kildekode

Tilgjengeliggjøring av kildekode og bruk av sandkasseløsningene

Kildekoden som ligger til grunn for sandkasseløsningene og testcasene beskrevet i denne rapporten blir gjort offentlig tilgjengelig som *open source*. Formålet er å bidra til åpenhet, etterprøvbarhet og kunnskapsdeling, samt å gi andre aktører mulighet til å benytte kildekoden til egen testing og utforsking.

Koden er utviklet for å støtte de konkrete testcasene og eksperimentene som er gjennomført i prosjektet, og er tilpasset forenklede prosesser og kontrollerte sandkassemiljøer. De kan *forkes* og kjøres lokalt i egne utviklingsmiljøer, slik at interesserte kan sette opp sandkasseløsninger og teste funksjonalitet og arkitekturvalg på egen maskin. Koden kan også benyttes som utgangspunkt for videre eksperimentering eller sammenligning med alternative tekniske løsninger. Kildekoden blir tilgjengelig i Norges Banks Github-repository[29].

Avgrensninger og forbehold

Koden er **ikke utviklet for bruk i produksjonssettinger**, og skal ikke benyttes som grunnlag for reelle betalings-, oppgjørs- eller verdipapirsystemer. Det er ikke gjennomført

[29] Se Github (2026).

fullstendige vurderinger av sikkerhet, robusthet, ytelse, personvern eller regulatorisk etterlevelse, og det er gjort bevisste forenklinger i arkitektur og implementasjon for å muliggjøre effektiv eksperimentell testing.

All bruk av koden skjer på eget ansvar. Eventuell videre bruk eller videreutvikling forutsetter selvstendige vurderinger av tekniske, sikkerhetsmessige, juridiske og regulatoriske forhold.

Referanser

BIS (2025) Annual Economic Report 2025, Chapter III. The next-generation monetary and financial system. (lenke)

BIS (2023) Annual Economic Report 2023, Chapter III. Blueprint for the future monetary system: improving the old, enabling the new. (lenke)

Github (2026) Official CBDC lab repo of Norges Bank. (lenke)

Linux Foundation (2026) Linux Foundation Projects (lenke)

Norges Bank (2026) Digitale sentralbankpenger – sluttrapport fra prosjektfase 5, *Norges Bank Memo, 1/2026*. (lenke)

Norges Bank (2022) Sentralbanken. *Strategi 25*. (lenke)

Store norske leksikon (2026) Definisjon av «blokkjede». (lenke)

Norges Bank
Norges Bank Memo

Oslo 2026

Adresse: Bankplassen 2
Post: Postboks 1179 Sentrum, 0107
Oslo Telefon: 22 31 60 00
E-post: post@norges-bank.no
www.norges-bank.no

ISSN 1894-0277 (online)
ISBN 978-82-8379-398-7 (online)