

Her følger som avtalt mer informasjon om back-up løsningen benyttet av både Flyt PPT og de fleste av Visma sin øvrige skyløsninger for offentlig sektor.

Visma har som nevnt tidligere rutiner som gjør at det tas backup hver andre time av Visma Flyt PPT. Formålet med denne er å kunne ta en restore av hele løsningen om det blir en katastrofe. Vi har sikret løsningen godt med master/slave-oppsett på to ulike fysiske lokasjoner slik at selv om masterdatabasen går ned vil vi fortsatt kunne bruke slaven som master.

Selve løsningen er en multi-tenant som betyr at alle kundene ligger i en og samme database. Dette gir mange fordeler i forhold til drift og sikkerhet, men gjør det mye mer komplisert å gjøre restore av en kunde i forhold til historiske On Prem / lokalt installerte løsninger med sine egne databaser.

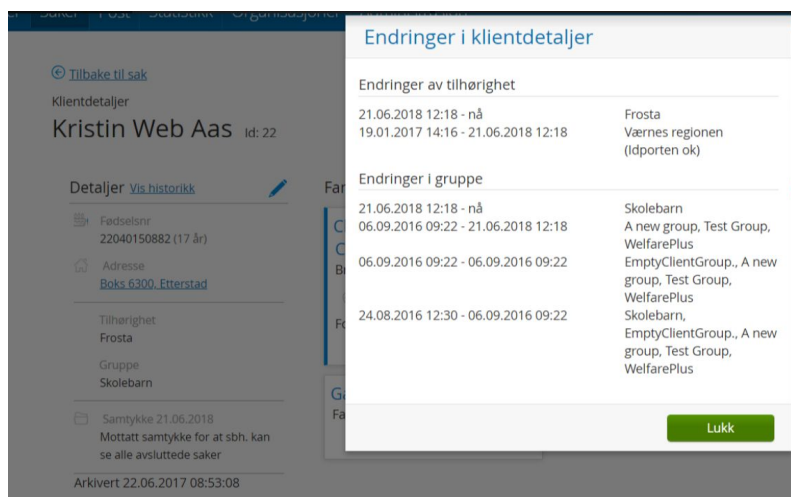
Løsningen benytter microservices som en del av en moderne arkitektur. Dette gjør at tradisjonell backup/restore er mindre aktuelt siden hver microservice tar hånd om 'sine' data i egen database.

En annen utfordring er knyttet til restore er håndtering av integrasjoner (SvarUt, SvarInn, Arkivering, Checkware etc). Når det gjøres en restore vil alle referanser til eksterne løsninger få seg en alvorlig knekk og data være borte (f.eks SvarInn-dokumenter). Dette er en betydelig jobb i å rydde opp i. Det gjelder f.eks. dokumenter som ligger i filestorage som vi ikke har id'en eller passordet til, eller arkiv som vil inneholde dokumenter som ikke finnes lenger i hovedløsningen etc. Derfor vil restore være siste utvei i forhold til problemer i løsningen, og ikke en mekanisme vi ser at det er hensiktsmessig å benytte for å rette opp evt "feil" en kunde eller bruker har gjort. Derfor er heller ikke en slik strategi valgt.

Vi har derimot basert oss på å bruke existence time på data slik at data aldri slettes fra databasen og dermed kan gjenopprettes via forskjellige mekanismer om det skulle være nødvendig. På denne måten kan vi håndtere "brukerfeil" via funksjonalitet i systemet med mulighet til å se audit-logger, tidligere verdier på felter, etc. Dette da i stedet for å gjøre en tradisjonell database restore. Vi har allerede gjort oss nytte av slik funksjonalitet i noen sammenhenger der det er naturlig at bruker kan gjøre feil. For eksempel kan en ved redigering av dokumenter gå tilbake og se på tidligere lagrede versjoner (og hente frem disse). Det er videre mulig å benytte seg av disse lagrede dataene til å angre visse funksjoner i systemet.

Et konkret eksempel fra Flyt PPT er 'bulk-endring' der vi i bakgrunnen tar vare på hva som blir gjort av endring, og kan lage til kode for å angre utføringen av bulkendringen.

Det er også mulig å f.eks. lage til en 'historievisning' av tidligere registrerte data på feltnivå, noe vi gjør med tilhørighet og adresseinformasjon på klienten ('vis historikk'-link på klientkortet).



En mer generell funksjon for dette med historiske data finnes også i administrasjon / aktivitetslogg der en kan se gammel og ny verdi. På sikt kan vi også se for oss andre funksjoner som f.eks å åpne en person, sak eller annet og 'bla seg bakover' i tid for å se alle endringer. Vi har ikke prioritert den type funksjonalitet foreløpig og det ligger heller ikke på roadmap for den nærmeste perioden, men mulighetene og dataene finnes.

Felt	Gammel verdi	Ny verdi
Sak tittel	Klientmappe	Klientmappe
Sak nummer	2018-163	2018-163
Sak klient id	38754	38754
Sak organisasjon		
Sak type	Client	Client
Sak opprettet dato	18. september 2018	18. september 2018
Sak eiet av	806	806
Sak eiet av team id	38384	38384
Sak event element sist endret	18. september 2018	18. september 2018
Sak henvist av klient ja/nei	False	False
Sak tillegskategorier id'er	[]	[]
Sak medsaksbehandlere	[]	[]
Sak hovedvanske	352	352
Sak oppdrags id		
Sak status	New	New
Sak status tidsstempel	18. september 2018	14. mars 2018
Sak oppfølgingsdato		
Sak contactpersonid		
Sak prereferralcalldate		
Sak receiveddate	18. september 2018	14. mars 2018
Sak assignmentordererid		
Sak closingreasonregistryid		
Sak usedefaultdeadline	True	True
Sak closingreasonassignmentregistryid		

Oppsummert kan vi si at strategien med å benytte databaserestore for å rette opp i ting henger igjen fra den gangen systemer var mye mer isolerte enn de er i dag. Denne strategien mener vi ikke fungerer i dagens systemer med mengder av integrasjoner og avhengigheter mellom både interne og eksterne tjenester.

Restore-muligheten har vi for å kunne håndtere helt katastrofale situasjoner der alternativet hadde vært å miste alle data om vi ikke hadde backup.

Vi kan uansett bekrefte og garantere at alle dataene ligger i løsningen, men at måten å restore data på er annerledes enn hva dere er vant med i tradisjonell forstand.

Håper dette svarte dere ytterligere på spørsmål dere har hatt.