

FS Overordnet ekstern målarkitektur

FØRST

1	Sammendrag	3
2	Målarkitektur	4
2.1	Personvern og sikkerhet	5
2.2	Identitet og tilgangsstyring	5
2.3	Utviklerportal	5
2.4	De programmerbare grensesnittene (APIene)	6
2.4.1	Eksisterende APIer	6
2.4.2	Fremtidige APIer	7
2.4.3	Avvikling av APIer.....	7
2.5	Applikasjonene	7
2.5.1	Eksisterende applikasjoner	7
2.5.2	Fremtidige applikasjoner	8
2.5.3	Avvikling av applikasjoner.....	8
2.6	Integrasjoner og automatisering	8
3	Verdiskaping og fordeler med målarkitekturen	8
3.1	Verdiskaping	9
3.2	Interne fordeler	9
4	Konsekvenser og omfang.....	9
4.1	Konsekvenser for sektoren	9
4.2	Interne konsekvenser	10
5	Åpen kildekode, utviklingsprosess og samarbeidsform	10
5.1	Åpen kildekode	10
6	Oppsummering	11
7	Referanser.....	11

1 Sammendrag

«Vi må både etablere standardiserte fellestjenester og samtidig dra nytte av den brukernære innovasjonskraften i institusjonene. Fellestjenester må levere plattformer med programmerbare grensesnitt som tilrettelegger for digital transformasjon gjennom enkle, lettbenyttede og effektive tjenester. Dette vil også åpne for et bredt samarbeid med næringslivet om nyskaping, slik flere av institusjonene allerede har i dag.»

— Handlingsplan for digitalisering i høyere utdanning og forskning

Når FS skal moderniseres er det et viktig premiss at arbeidet understøttes av en målarkitektur som er i tråd med Produktstrategien for FS, sektorens strategiske retning og rammene som pålegges digitalisering av UH-sektoren.

Dette innebærer at FS må etableres som *en plattform som tilrettelegger for digital transformasjon*. Plattformen skal ¹for utviklere ²å lage applikasjoner og integrasjoner som utnytter FS-data og FS-prosesser for å løse behovene til sektoren og næringslivet. Den må bygges på en måte som gjør at kontinuerlig utvikling kan utføres på en økonomisk bærekraftig og forutsigbar måte.

FS-plattformen må bygges med tanke på langsiktighet og stabilitet. Langsiktighet følger av at FS allerede har levert verdi til UH-sektoren i mer enn 25 år, og behovene innen det studieadministrative området er voksende heller enn avtagende. Stabilitet kreves fordi økosystemet rundt FS er stort og voksende. Det er med andre ord mange viktige prosesser som er avhengig av FS og som vil slutte å fungere eller få følgefeil dersom FS feiler.

Når man bygger en plattform som skal levere verdi over så lang tid, så er det vesentlig å ta høyde for at både behovene og teknologien endrer seg med tiden. Vi legger derfor til grunn at FS må være en responsiv teknologisk plattform med en tydelig livssyklusmodell, slik at vi kan innføre og utvikle funksjonalitet og teknologi etter behov, på en forutsigbar og forsvarlig måte.

Videre legger vi opp til at plattformen bygges på en åpen, inkluderende og tillitsvekkende måte². Vi legger derfor til rette for at sektoren og andre skal ha innsyn, og anledning til å bidra, i utviklingen av plattformen. Dette sier vi mer om i kapittel 5.

Dette dokumentet går ikke i detaljer rundt teknologivalg og implementasjon av målarkitekturen. I stedet fokuserer vi på den stabile flaten av plattformen som UH-sektoren, Unit selv og tredjeparter kan bruke for å bygge løsninger og skape verdi. For detaljer om det som ligger under overflaten viser vi til utviklerportalen som vi vil publisere i løpet av høsten 2021 og deretter holde løpende oppdatert.

¹ Her inkluderer vi også utviklere som ikke har tilknytning til Unit eller sektoren

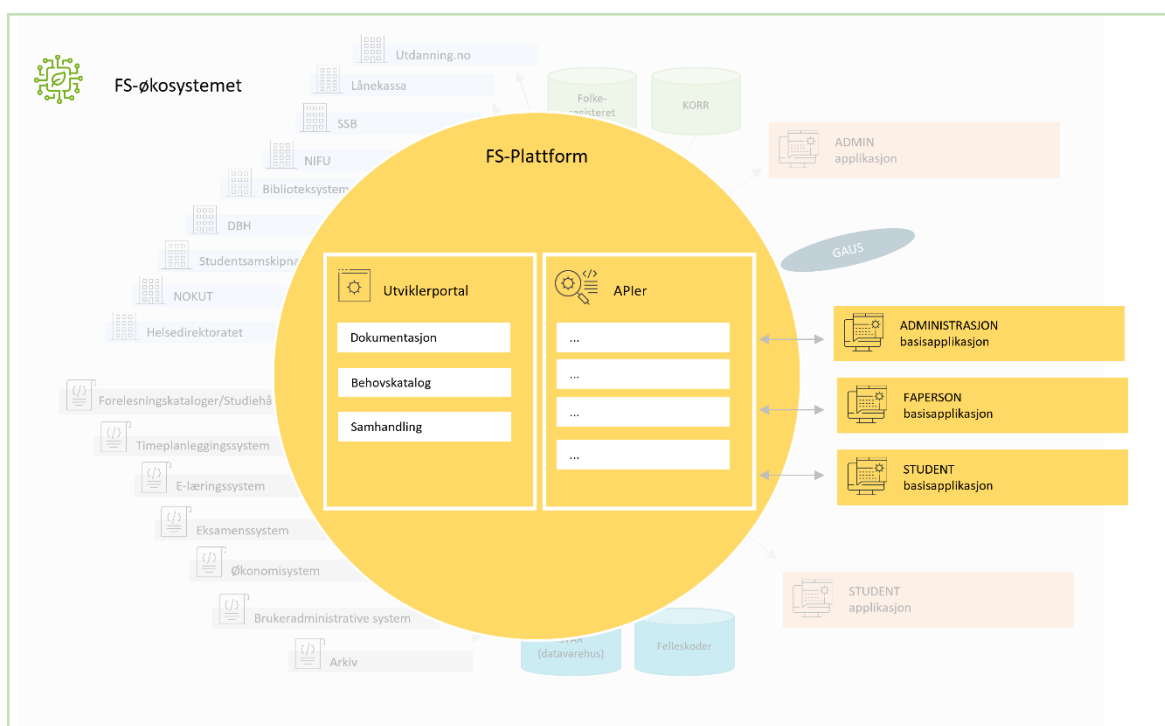
² Følger av regjeringens digitaliseringsstrategi for offentlig sektor

[<https://www.regjeringen.no/no/dokumenter/en-digital-offentlig-sektor/id2653874/>](https://www.regjeringen.no/no/dokumenter/en-digital-offentlig-sektor/id2653874/)

2 Målarkitektur

En plattform er et grunnleggende system man bygger programvare på toppen av. Plattformen tilgjengeliggjør funksjonalitet og verktøy som gjør det mulig å levere forretningsverdi raskere og billigere, samtidig som den avgrenser tilgang til det underliggende systemene plattformen selv bygger på.

— FS-arkitektur og modernisering



Figur 1: Overordnet arkitekturskisse

Som illustrert i Figur 1, skal de programmerbare grensesnittene (APIene) i FS-plattformen være den eneste måten³ man kan aksessere data og forretningslogikk for FS.

Dette gjør vi for å muliggjøre Produktstrategien for FS, som krever at Unit på sikt skal kunne endre på hvor og hvordan data og forretningslogikk lagres og vedlikeholdes, uten at det påvirker økosystemet rundt FS. Endringen er også i tråd med Gartner sitt anbefalte arkitekturprinsipp om at nåværende «informasjonseierskap» ikke skal være førende for systemet⁴.

Kjernebrukergruppen av FS-plattformen er utviklere som skal kunne bygge applikasjoner og integrasjoner som benytter seg av FS-data og forretningslogikk.

I tillegg til plattformen så vil Unit fortsatt være ansvarlig for å utvikle og forvalte et sett med basisapplikasjoner til studenter, fagpersoner og administratorer, som er de tre andre

³ Et mulig unntak for dette er behov knyttet til rapportering og analyser. Her arbeides det med å etablere en løsning for Felles beslutningsstøtte, og det er naturlig at denne løsningen også tilgjengeliggjør data fra FS.

⁴ Side 24 av Gartner sin evaluering av FS <<https://www.unit.no/aktuelt/evaluering-av-felles-studentsystem-fs>>

kjernebrugergruppene for FS. Som illustrert er basisapplikasjonene ikke en del av plattformen, og skal kun ha tilgang til FS-funksjonalitet ved hjelp av APIene i plattformen. Se kapittel 2.5 for en overordnet beskrivelse av hvordan dette er tenkt.

Utviklerportalen og verktøyene i denne skal fasilitere samarbeid mellom utviklerne og brukerne av plattformen. Samarbeidsformen beskrives ytterligere i kapittel 5. Til høyre i figuren ovenfor viser vi felleskomponenter som plattformen benytter seg av og samspiller med. Hvordan dette henger sammen beskriver vi i det følgende.

Målarkitekturen vi beskriver i denne rapporten er begrenset til den delen av plattformen som er synlig utenfra. For detaljer om hvordan plattformen skal bygges og håndtere data og forretningslogikk i de kommende 5-10 årene så henviser vi til utviklerportalen.

2.1 Personvern og sikkerhet

Det følger av [Nasjonal strategi for digital sikkerhet](#) og [personopplysningsloven](#) at datasikkerhet og personvern er førende prinsipper for utviklingen av FS-plattformen. Disse temaene er tett knyttet til hvor og hvordan data og forretningslogikk lagres. Derfor viser vi til utviklerportalen for detaljer om hvordan plattformen bidrar til å ivareta disse prinsippene.

Helt overordnet vil vi likevel poengtere at det å håndtere personvern og sikkerhet på en god og forsvarlig måte krever en kultur hvor viktigheten av dette blir forstått og verdsatt. Denne kulturen må innarbeides og utøves av både ledelsen og av de som bygger plattformen. Vi kommer derfor til å jobbe bredt og systematisk med dette på tvers av tjenester og systemer.

2.2 Identitet og tilgangsstyring

All bruk av plattformen må selvsagt være autentisert slik at vi vet hvor forespørselen kommer fra og hvem som står bak. Autentisering skjer i første omgang ved hjelp av Feide. Vi vil eventuelt legge til støtte for flere autentiseringsløsninger underveis ved behov.

Alle forespørsler til plattformen vil deretter autoriseres i lys av autentiseringen. Dette betyr at forespørslene blir avvist dersom det ikke er mulig å bevise at nødvendig tilgang foreligger. Vi viser igjen til *utviklerportalen* for detaljer om hvordan dette skjer i praksis.

Sektoren har besluttet⁵ å etablere et felles system for tilgangsstyring (IAM). Plattformen legger derfor opp til å støtte dette tilgangsstyringssystemet. Detaljene rundt hvordan dette skal fungere må vi etablere i samarbeid med IAM-prosjektet tidlig i moderniseringen. Vi kommer også til å koordinere oss med Datadelingsprosjektet som jobber med å etablere en referansearkitektur som løser disse samme problemstillingene.

2.3 Utviklerportal

For å gjøre det enkelt for utviklere å bruke plattformen, må vi ha en utviklerportal med både overordnet og detaljert informasjon om plattformen, gode veiledende artikler og eksempelprosjekt man kan lære av. Her vil man finne en oversikt over APIene som inngår i plattformen og hvilke behov de dekker, samt detaljert dokumentasjon av hvert enkelt API. Datakatalog/metadata?

Portalen skal i tillegg ivareta forutsigbarhet og åpenhet, noe som er viktige suksessfaktorer for plattformen. Diskusjonsforum, issue tracker og kildekode vil være tilgjengelig for de som deltar i utviklingen. Oppdaterte veikart og endringslogger gir interesserte anledning til å holde seg orientert

⁵ Beslutningen om å etablere felles tilgangsstyringssystem (IAM) ble fattet av digitaliseringsstyret. <https://www.uninett.no/tilgangsstyring#om-iam-prosjektet>

om retning og fremdrift, mens periodiske nyhetsartikler vil gi en oppsummering av fremdriften for de som ikke har tid til å følge med fra dag til dag.

APIer som er under utvikling vil selvsagt være tydelig markert og det vil foreligge en avviklingsplan. Se kapittel 2.4.3 for detaljer.

2.4 De programmerbare grensesnittene (APIene)

Et API er et programmerbart grensesnitt som lar utviklere løse brukerbehov ved å utføre operasjoner mot et eller flere underliggende systemer.

— *FS-arkitektur og modernisering*

Hva som er det beste teknologivalget for bygging av APIer har endret seg mye i løpet av de siste 30 årene. CORBA dominerte på 1990-tallet, SOAP på begynnelsen av 2000-tallet før REST etter hvert tok over og nå er fullstendig dominerende. Sannsynligvis vil det skje nye omveltninger i fremtiden.

Målarkitekturen for plattformen sier derfor lite om konkrete teknologier. I stedet konsentrerer vi oss om behovene en slik plattform må tilfredsstillere for at den skal være et godt verktøy for sektoren, samtidig som den kan utvikles og forvaltes på en effektiv måte i møte med stadig endrede behov og muligheter.

Det er likevel trygt å si at plattformen vil støtte brukere som ønsker å bli varslet om hendelser, i tillegg til den mer vanlige modellen hvor brukeren sender forespørsler og mottar svar. Se digitaliseringsdirektoratets referansearkitektur for datautveksling for mer informasjon om datautvekslingsmønstre⁶.

Et viktig krav til APIene i plattformen, uavhengig av teknologi, er at det må foreligge detaljerte spesifikasjoner i et maskinlesbart format. Dette utgjør kontraktene mellom plattformen og brukerne og er dermed helt nødvendige dokument når man skal designe, bygge og forvalte plattformen.

2.4.1 Eksisterende APIer

Selv om FS-plattformen ikke eksisterer enda, så finnes det allerede mange APIer som tilgjengeliggjør data og funksjonalitet fra FS. Som beskrevet i rapporten *FS-arkitektur og modernisering* så er flere av disse under utvikling og har vært det i lang tid. Likevel er de mye brukt, og Unit har derfor ikke lyktes med å fullføre utviklingen.

Tidlig i moderniseringen vil vi gjennomføre en grundig gjennomgang av alle de eksisterende APIene for å avdekke hvilke som skal inkluderes i plattformen og hvilke som skal utvikles. Merk at de APIene som inkluderes vil flyttes og ta i bruk ny felles autentiseringsløsning, dermed vil alle applikasjoner som bruker FS måtte endres i tiden som kommer.

For å gjøre denne prosessen så smidig som mulig, vil Unit utarbeide migreringsplaner for samtlige APIer med tydelig informasjon om hva som må gjøres og en tidsplan for når dette må gjennomføres for at applikasjonene skal fortsette å fungere.

⁶ <<https://nasjonal-arkitektur.github.io/architecture-repository/data-exchange-patterns/about-data-exchange-patterns.html>>

2.4.2 Fremtidige APIer

Vi jobber med å fullføre en evaluering av en PoC av «spesifikasjon først»-metodikken for API-utvikling som vi gikk i gang med våren 2021. Metodikken innebærer at man produserer en maskinlesbar spesifisering av APIet før man begynner å bygge. Resultatene viser tegn til at dette gir APIer som:

1. er billigere å produsere
2. er bedre tilpasset behovene
3. har høyere kodekvalitet/færre feil

Vi kommer derfor til å bygge fremtidige APIer ved hjelp av denne metodikken. Hvilke APIer som skal bygges og i hvilken rekkefølge er for øvrig noe vi må kartlegge tidlig i moderniseringen.

2.4.3 Avvikling av APIer

For å holde forvaltningskostnadene for plattformen under kontroll, så er det nødvendig å kunne avvikle APIer. Problemet er at dette påfører brukerne kostnader. Dermed må det gjøres på en måte som minimerer den totale kostnaden for sektoren, ikke bare for plattformen.

Vi, som plattform leverandører, har ansvaret for å levere en plattform som er så stabil og forutsigbar som mulig. Dette innebærer at avvikling og utfasing skjer sjeldent og varsles i god tid. Dette varselet skal vise til en avviklingsplan som forklarer hvorfor APIet avvikles, tidsplanen for avviklingen og hvordan plattformen ivaretar behovene APIet har tjent. Det bør også publiseres migreringsveiledere som gjør det enklere for brukere å oppdatere applikasjonene sine.

Brukerne av plattformen er ansvarlige for å lese avviklingsplanen og oppdatere applikasjonene sine i tide. Dersom dette ikke lar seg gjøre, må man ta kontakt innenfor gitte frister, slik at man kan samarbeide om å finne en forenlig løsning.

2.5 Applikasjonene

Unit har ansvaret for å vedlikeholde en rekke applikasjoner for studenter, ansatte og administrasjon. Målkitekturens krav om at disse applikasjoner ikke lenger kan koble seg til FS-databasene direkte er en vesentlig endring som innebærer et betydelig transformasjonsarbeid.

Mange av applikasjonene bærer i tillegg preg av tidens tann. Dette er et problem fordi det medfører en del forvaltningsutfordringer som følge av at de bygger på gammel teknologi, samtidig som det påvirker brukeropplevelsen negativt.

Sett i sammenheng, er dette derfor en gylden anledning til å gjøre en grundig gjennomgang av applikasjonsporteføljen i sin helhet. Målkitekturen legger derfor opp til å redusere antall brukerflater. For å lykkes med dette vil vi samarbeide med sektoren og jobbe brukersentrert og ut ifra brukerreisene til kjernebrukergruppene.

Denne prosessen vil resultere i et sett med basisapplikasjoner som vil være tilgjengelig for alle brukerne av plattformen. Applikasjonene vil i tillegg bli utviklet på en måte som gjør det mulig for sektoren og andre å bidra i utviklingen, dersom man skulle ønske dette.

2.5.1 Eksisterende applikasjoner

Som nevnt kommer vi til å gjøre en grundig gjennomgang av samtlige eksisterende applikasjoner for å bestemme hvilke som skal bli en del av plattformen og hvilke som skal avvikles.

Det er sannsynlig at applikasjonene som inkluderes i plattformen vil gjennomgå en så stor transformasjon, at de i praksis vil fremstå som nye applikasjoner; både for sluttbrukeren og utviklerne.

Vi viser derfor til kapittel 2.5.3 for litt mer om prosessen rundt avviklingen av applikasjoner.

2.5.2 Fremtidige applikasjoner

Målarkitekturen setter som krav at all data og funksjonalitet skal være tilgjengelig ved bruk av APIer, og at dette skal være den eneste måten applikasjoner skal kunne bruke FS på. Basisapplikasjonene er ikke et unntak for denne regelen, dermed følger det at plattformen vil bli utviklet ved hjelp av «API-først»-metodikken.

Dermed må alle behovene applikasjonene skal løse håndteres ved hjelp av APIene i plattformen. Dersom dette ikke er mulig, må APIene utvides før applikasjonen kan skrives. De resulterende APIene blir naturligvis en del av plattformen og er dermed underlagt reglene for APIer som vi etablerte i kapittel 2.4.

Vi går grundigere inn på applikasjonsarkitekturen og prosessene vi vil bruke internt for å sørge for at dette skjer på en effektiv måte i utviklerportalen.

2.5.3 Avvikling av applikasjoner

For å kunne etablere plattformen etter målarkitekturen må vi som nevnt gjøre større endringer på alle dagens applikasjoner. I forbindelse med dette skal vi vurdere hver enkelt applikasjon for å se om en slik investering er formålstjenlig, eller om behovene heller skal løses et annet sted og at applikasjonen avvikles.

Dette må skje på en måte som minimerer kostnaden for sektoren og frustrasjon for sluttbrukerne våre. Vi vil derfor utarbeide en detaljert prosess for avvikling i samarbeid med sektoren. Her forventer vi å mellom annet måtte etablere avviklingsplaner som tydelig kommuniserer tidsperspektivet og omfanget for de respektive avviklingen, samt veiledninger for de berørte brukergruppene.

2.6 Integrasjoner og automatisering

De fleste integrasjonene mot FS skjer i dag ved hjelp av de eksisterende APIene og vil dermed håndteres av prosessen vi beskriver i kapittel 2.4.1.

I tillegg finnes det en del direkte integrasjoner og automatiseringer i form av mellom annet:

- Robotic process automation (RPA) mot FS-klienten
- Direkte tilgang til FS-databaser
 - Applikasjoner som kobler til FS-databaser
 - Brukere som kobler til FS-databaser
- Prosedyrer i FS-databaser som kjøres med jevne mellomrom
- Prosedyrer i FS-databaser som kjøres manuelt fra FS-klienten

Alt dette gjøres for å ivareta behov som ikke kan løses ved hjelp av dagens APIer, eller fordi det fungerer og man ikke har hatt anledning eller motivasjon til å endre prosessene.

Målarkitekturen gjør at alle disse løsningene må avvikles. Vi kommer derfor til å samarbeide med sektoren om å utarbeide en rekke avviklingsplaner i tråd med beskrivelsen i 2.4.3. Som del av dette vil vi identifisere hvilke behov som ikke allerede dekkes av APIer, slik at disse kan utarbeides.

3 Verdiskaping og fordeler med målarkitekturen

Den nye arkitekturen åpner opp for en rekke muligheter som tidligere har vært vanskelig å realisere.

3.1 Verdiskaping

Målarkitekturen for FS åpner opp for et samfunn/community der deltakerne kan bidra til å bygge løsninger som er til fellesskapets beste. Noen av de viktigste verdimålene arkitekturen gir er:

- Forutsigbarhet og åpenhet rundt utvikling av plattformen slik at flere brukergrupper enn tidligere kan påvirke prioriteringene og gi verdifull innsikt i de faktiske behovene som foreligger
- Muliggjør større grad av gjenbruk i sektoren
- Stabilitet og forutsigbarhet for brukerne
- Vesentlig forbedret mulighet for prosessautomatisering som følge av at all funksjonalitet blir tilgjengelig som APIer
- Enklere å komme i gang med å bruke FS-data og funksjonalitet som følge av godt dokumenterte APIer og bruksveiledninger
- Åpner opp for å lage langt mer brukersentrerte sluttbrukerapplikasjoner
- Gjør det mulig å bygge inn funksjonalitet fra FS i eksisterende brukerflater
- Institusjoner og andre som ønsker å bidra til utviklingen av plattformen kan gjøre dette⁷

3.2 Interne fordeler

For Unit vil ny målarkitektur være med å bidra til å strømlinjeforme større deler av tjenesteutviklingen og teknologien vi vil benytte oss av i fremtiden. De største gevinstene internt vil være følgende:

- API-først muliggjør gjenbruk av APIer og forenkler innovasjon ettersom det er mulig å bygge på eksisterende APIer.
- Felles studentregister / IAM + Feide vil gjøre autentisering og autorisasjon enklere for applikasjoner, og samtidig sikre etterlevelse av lovverk og retningslinjer
- Økt kvalitetssikring på leveranser
- Økt samarbeid gjennom et fellesskap hvor det er mulig å samarbeide på tvers av institusjonene
- Økt grad av samarbeid og transparens i arbeidet med utvikling av tjenestene vil legge til rette for en mer effektiv innhenting av tilbakemeldinger fra brukerne. Dette gir økt innsikt om hva brukerne mener vil gi mest verdi.

4 Konsekvenser og omfang

Målarkitekturen for FS er en vesentlig endring på hvordan man har tenkt på og drevet utviklingen av FS frem til i dag. Dermed er konsekvensene omfattende og endringene må innføres gradvis.

4.1 Konsekvenser for sektoren

Moderniseringen av FS etter ny målarkitektur vil medføre endringer i mange prosesser. Unit vil komme med et konkret forslag og retningslinjer for hvordan dette kan håndteres best mulig, men det er tydelig at det vil komme endringer som kan bli krevende for sektoren.

De overordnede konsekvensene er som følger:

- All bruk av eksisterende APIer vil måtte oppdateres til å bruke ny felles autentiseringsmetode samt ny API-adresse

⁷ Dette vil skje etter hvert, da det krever at prinsippene og verktøyene for åpen kildekode og en åpen utviklingsmodell er på plass.

- All bruk av data/funksjonalitet fra FS som ikke skjer via Unit sine applikasjoner eller APler må endres så snart det er mulig å løse behovene ved hjelp av plattformen
- Migrering av eksisterende applikasjonene vil medføre at sektoren må oppdatere interne prosesser og dokumentasjon

4.2 Interne konsekvenser

For Unit vil etablering av ny målarkitektur få konsekvenser for vår eksisterende portefølje og hvordan denne skal vedlikeholdes videre, samt at det skal pågå en prosessendring i mange ledd for hvordan forvaltning og utvikling skal foregå fremover. Noen av de viktigste konsekvensene moderniseringen har for Unit er:

- Setter større krav til Unit som skal forvalte plattformen: Dokumentasjon, ytelse, kommunikasjon, samarbeidsevne, smidig i større deler av prosessen enn kun utvikling.
- Nye prosesser vil ta tid å innføre før alt er operativt (viktig å ha kort- og langsiktig plan mtp. forventningsstyring).
- Økt testing
- Eksisterende portefølje må gradvis oppdateres og/eller skrives om til å passe ny arkitektur eller evt. avvikles.

5 Åpen kildekode, utviklingsprosess og samarbeidsform

I tråd med Units virksomhetsstrategi for videreutvikling og samhandling, og på digitaliseringsdirektoratets oppfordring⁸, bør FS-plattformen utvikles som åpen kildekode, og sektoren oppfordres til å bidra.

Institusjonene skal kunne bruke FS-plattformen for å innovere og løse egne behov, samtidig som felles behov løses i samarbeid for å få best mulig fellestjenester raskest mulig. Vi anbefaler derfor at Unit og sektoren går sammen for å finne ut hvordan dette kan gjøres på en best mulig måte, og utvikler prinsipper og arbeidsmetoder for dette i tiden framover.

Rent konkret kan åpen kildekode innebære at utvikling av FS-plattformen og basisapplikasjonene kan planlegges og forvaltes gjennom verktøy som er åpent tilgjengelige for omverdenen. Dette gir institusjoner, samarbeidspartnere og andre interesserte tilgang til å se hva det arbeides med, følge progresjon i arbeidet, samt bidra med kommentarer og innspill til arbeidet. Bidrag til kodeskriving kan også være aktuelt.

Et viktig utgangspunkt for å lykkes med en bred samskaping av FS, er at endringer som skal gjøres begynner med gode behovsbeskrivelser. Dette for å sikre en felles og uttalt forståelse av hva som skal lages. Behovsbeskrivelsene skal kunne diskuteres åpent av interessenter og utviklere, for å videre avklare behovene og komme fram til om de skal løses av FS eller må håndteres andre steder. Når et behov er avklart og verifisert, kan man gå videre til å skissere løsningsforslag, som man igjen diskuterer og itererer åpent om før endringene utføres.

5.1 Åpen kildekode

Åpen kildekode i denne sammenheng innebærer mer enn å åpne opp for tilgang til, og samarbeid om, den tekniske utviklingen av FS. Det betegner også en kultur og et mindset som setter åpenhet og samskaping i sentrum, noe vi tror er viktig for at vi skal kunne transformere FS til en responsiv

⁸ Se arkitekturprinsipp nummer 6 for offentlig sektor <<https://www.digdir.no/digitalisering-og-samordning/prinsipp-6-lag-digitale-losninger-som-stotter-samhandling/1063>>

teknologisk plattform som imøtekommer sektorens behov som stadig er i endring. For å kunne lykkes med en slik transformasjon i hvordan vi arbeider med utviklingen av FS, trenger vi å utarbeide prinsipper og retningslinjer for å sikre at samskapingen skjer på en måte som er til det beste for fellesskapet og ivaretar høy kodekvaliteten. Disse retningslinjene skal også sørge for at samfunnet rundt FS skal være et hyggelig sted å være for alle som skulle ønske å involvere seg, samtidig som de uttrykker hvilke kjøreregler og forventninger som gjelder for å få til et godt samarbeid. Videre må prinsippene og retningslinjene også ta stilling til hvordan denne åpne samskapingen skal skje innenfor de til enhver tid gjeldende strategiske føringer, avklarte prioriteringer og økonomiske rammer. Overdragelse av applikasjoner utviklet av sektoren

Dersom slike løsninger blir mye brukt er det naturlig/mulig at Unit deltar i forvaltning og utvikling. Hvis Unit skal overta forvaltning av en applikasjon så settes det krav til teknisk tilstand/kodekvalitet, samt avklaring av driftskostnader og tilføring av ressurser slik at Unit kan gjøre forvaltning på en tilfredsstillende måte for sektoren.

6 Oppsummering

I tråd med produktstrategien for FS og andre strategiske føringer, skal FS plattformiseres gjennom en teknisk og organisatorisk moderniseringstransformasjon. Ved bruk av åpen kildekode i vid forstand og prinsippet om API-først skal dette gjøre FS mer endringsdyktig i møte med eksisterende og framtidige behov. Dette gir økte muligheter for samhandling i sektoren, åpenhet om utviklingsprosessen, stabilitet i tjenesteleveransen og forutsigbarhet for kunder og sluttbrukere.

7 Referanser

- Virksomhetsstrategi for Unit 2020 – 2023, https://www.unit.no/sites/default/files/media/filer/2020/08/Sak_43_20_Vedlegg_43A_Virksomhetsstrategi_Unit.pdf
- Produktstrategi for FS
- FS-arkitektur og modernisering, Rapport skrevet av Unit høsten 2020
- Utviklerportalen til FS-plattformen, nettsted som blir publisert i løpet av høsten 2021.
- Handlingsplan for digitalisering i høyere utdanning og forskning 2019-21, <https://www.unit.no/handlingsplan-digitalisering-i-hoyere-utdanning-og-forskning> (lest desember 2020).