

itSMF-konferansen 2022

WAKE ME UP!

IT for en bærekraftig fremtid

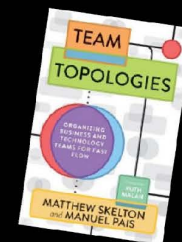
Clarion Hotel Oslo Airport, 10.-11.nov. 2022



Fart og flyt med produktteam og Team Topologies i NAV

Tommy Jocusen og Anders Skifte
NAV

Meld deg på
og få med
deltagergave!



itSMF.no

Påmelding: itsmfkonferansen.no

Denne PDF-versjonen av presentasjonen inneholder snakkebobler, slik som denne. De var ikke en del av den opprinnelige presentasjonen, da den ble presentert, men er satt inn i ettertid. Snakkeboblene inneholder omtrent det samme som ble sagt, da vi presenterte. Vi har prøvd å plassere de slik at de ikke er i veien for annet innhold, og vi håper de gjør det enklere for leseren å få med seg budskapet. God fornøyelse!

- Tommy & Anders, NAV IT

Tommy & Anders jobber i et team som har i oppdrag å hjelpe andre team med å oppnå fart og flyt, og da er teorien fra *Team Topologies* relevant, fordi boka handler om nettopp det. Altså: Hvordan kan vi organisere folk og teknologi på en måte som tilrettelegger for god fart og flyt, og høy verdiskapning? Det er en sosioteknisk problemstilling.

Sosioteknisk

Sosioteknisk handler kort forklart om samspillet mellom mennesker og teknologi. Mer spesifikt, og i kontekst av digital produktutvikling, handler det om hvordan vi best kan organisere oss rundt teknologi for å skape høy verdi.

Sosio-delen av ordet: mennesker og våre sosiale relasjoner og interaksjon med hverandre. Teknisk: datamaskiner, teknologi, kode, bygninger, kontorlandskap, prosesser, rutiner m.m – som påvirker hvem vi kommuniserer med og hvordan vi jobber. Det tekniske påvirker det sosiale. Det sosiale påvirker også det tekniske. Det betyr at måten vi organiserer teamene våre på, og hvordan de samhandler og kommuniserer med hverandre, vil påvirke systemarkitekturen (Conways law).

«Team-organiseringen er teknologi-arkitekturens førsteutkast».

Hvis vi ønsker en arkitektur hvor ulike deler av et system skal kunne endres hyppig og uavhengig av hverandre, så må vi også organisere oss for det. Da kan vi ikke designe arkitektur og organisasjon hver for seg. Vi er nødt til å se det i sammenheng.

Sosialt

Teknisk



Topologikameratene

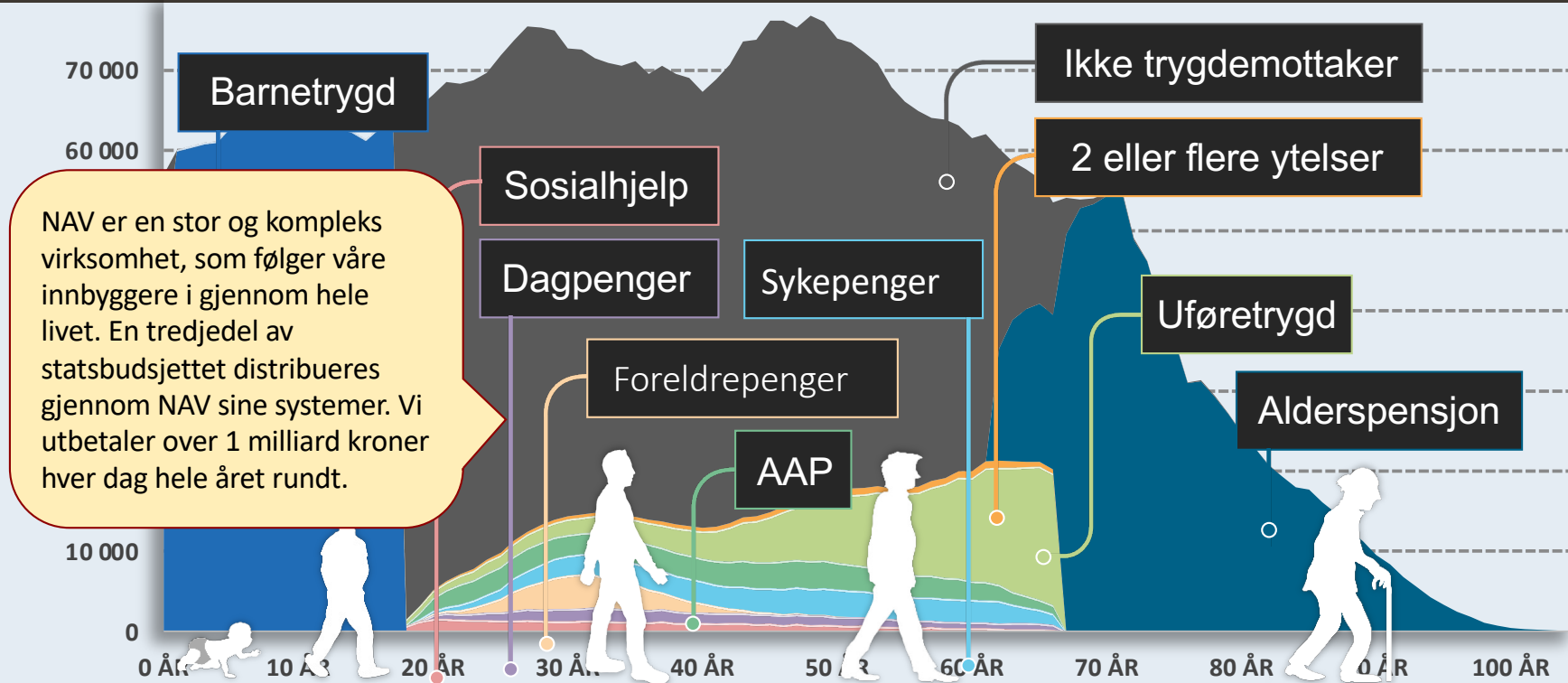
Typisk Fast Flow, aligned, complicated
enabling sub-platform stream team



Teamet vårt heter Team
Sosioteknisk, men noen
mener vi burde bytte
navn..

MURAL

NAV fra vugge til grav



Den norske befolkningen per desember 2017, og koblingen hvert årskull har til noen av NAVs trygdeordninger. Tall for 2020 viser at på grunn av pandemien mottok langt flere dagpenger enn det som kommer fram av denne illustrasjonen.

Dagens agenda er de tre elementene som må til for å lykkes
sosioteknisk: Tankesett, folk/organisering og verktøy/teknologi.

Folk og
tankesett



Organisering



Plattformer




Den gamle modellen

Fram til omtrent 2016 foregikk utvikling etter plan-build-run-modellen. Klassisk vannfall med prosjekter og overleveringer mellom siloer. NAV utredet og spesifiserte før utviklingen ble satt ut som konsulentoppdrag. Deretter skulle NAV kvalitetssikre. NAV hadde ikke kompetanse på utvikling selv. Det er flere problemer med denne måten å jobbe på, men noe av det viktigste er: Det tar lang tid fra idé til produksjon (ledetid). Det involverer mange mennesker og krever mye koordinering. Eierskap blir uklart, og det er lite eller ingen læring underveis om hvorvidt det man gjør skaper verdi. Dette er ikke synonymt med fart, flyt og høy endringsevne. Det var i tillegg store og komplekse systemer, som gjorde leveranser både krevende og risikofylte.



Plan-build-run



4 x hoved-
leveranse
per år

..så da valgte vi å gjøre leveranser sjeldent og ved å koordinere så mye som mulig på en gang. Det var fire hovedleveranser per år, som hver kunne inneholde mange tusen utviklingstimer. Det gjorde at funksjonalitet som var ferdig utviklet (i henhold til spesifikasjon) ble liggende i påvente av release, uten at noen fikk glede av det. Da skapes ingen verdi. Denne måten å jobbe på var en effektiv brems for læring, innovasjon, og verdiskapning. I tillegg ble all risiko skjøvet frem til release. For man visste jo lite om hvorvidt det som ble utviklet faktisk løste problemet det var ment å løse, før det ble levert og tatt i bruk.

Foto: [kees torn](#) - [EVER GIVEN](#)



... Og når noe går galt, så går det gjerne veldig galt. Store konsekvenser. Det er i tillegg dyrt og krevende å endre i ettertid, etter at alt er levert.

Enda dyrere ble det dersom leveransen viste seg å ikke løse problemet det var ment å løse.

Foto: [NASA](#)



Man skal aldri sette bort noe man ikke skjønner

NAV ble årets digitale suksesshistorie »
08. jun 2017
NAV vant i dag digitaliseringsprisen for årets digitale suksesshistorie på Direktoratet for forvaltning og IKTs (Difi) digitaliseringskonferanse i Oslo.

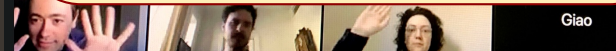
For å sikre at det er vi som sitter i førersetet, tar vi nå grep om IT-utviklingen i NAV.

Slik koda NAV ny dagpenge-løsning på tre dager

- Vi er dødsslitne, men stolte, forteller NAV-utviklerne som løste krisa med React, Kotlin og PostgreSQL.



I 2015 kom Torbjørn Larsen inn som ny og visjonær IT-direktør med klare meninger om hvordan NAV måtte endre seg, tenke nytt, jobbe smidig, ikke sette bort utvikling – men snarere ansette egne folk og ta kontroll på egen teknologiutvikling. Teknologi ble endelig ansett som kjernevirksomhet, også i NAV. Etter at man så positive effekter av den nye måten å jobbe på, så økte endringstakten ytterligere.



Vi er dødsslitne, men stolte – vi besto testen. De første søkerne fikk utbetalt penger på onsdag, og da feiret vi skriver Stefan Andreas Engelsen i NAV, og legger ved dette bildet av utviklerne. 🙌 Privat



AKTUELT

NAV har gjort et byks på Norsk innovasjonsindeks

Universums IT-liste 2020
+ Nav inn som en komet på IT-folkets drømmearbeidsgiver
– Det er ikke noe mysterium for oss, sier Navs IT-sjef Nav Jonas Slørdaahl Skjærpe.
06. nov. 2020

Digital evolusjon

NAV har i dag omtrent 400 internt ansatte innen utvikling, design og data. Vi har hendene på rattet selv, og beriker teamene med konsulenter der det er behov for det. Vi har gått fra en prosjektmodell til en produktmodell, hvor teamene eier og har ende-til-ende-ansvar for produktene sine, fra vugge til grav. Vi jobber for at teamene skal kunne bevege seg så fritt og uavhengig av hverandre som mulig.



Utvikling satt ut i markedet



Eierskap med egne utviklere og designere



Prosjekter og overleveringer



Smidig utvikling i produktteam med eierskap til domenet



Koordinerte hovedleveranser



Kontinuerlig leveranse



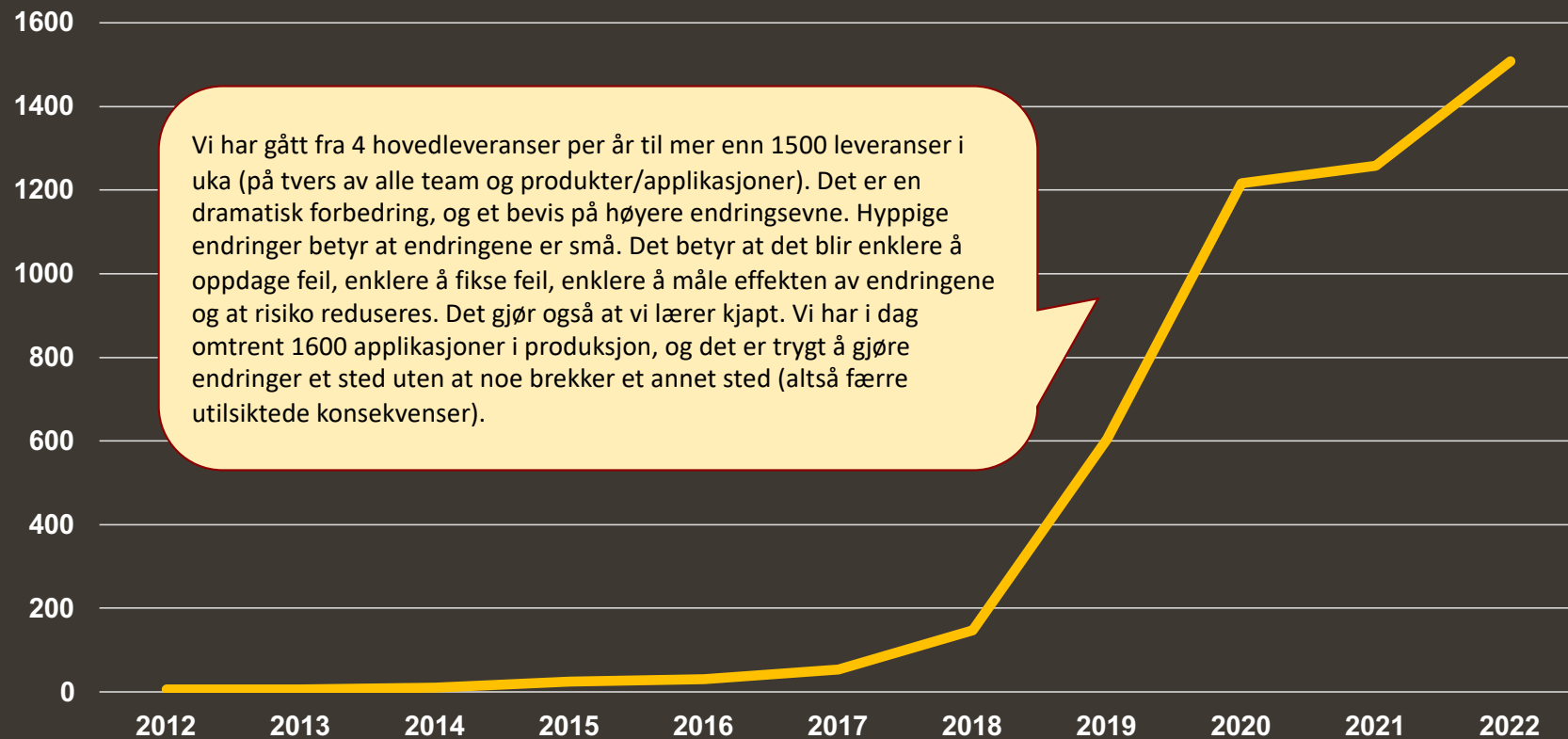
Noen få store, tunge og lite endringsdyktige systemer



Mange små løst koblede og endringsdyktige tjenester



Gjennomsnittlig antall prodsettinger per uke per år



Vi har gått fra 4 hovedleveranser per år til mer enn 1500 leveranser i uka (på tvers av alle team og produkter/applikasjoner). Det er en dramatisk forbedring, og et bevis på høyere endringsevne. Hyppige endringer betyr at endringene er små. Det betyr at det blir enklere å oppdage feil, enklere å fikse feil, enklere å måle effekten av endringene og at risiko reduseres. Det gjør også at vi lærer kjapt. Vi har i dag omtrent 1600 applikasjoner i produksjon, og det er trygt å gjøre endringer et sted uten at noe bryr seg et annet sted (altså færre utilsiktede konsekvenser).

.. Men det hjelper ikke å løpe fort om
man løper for lenge i feil retning

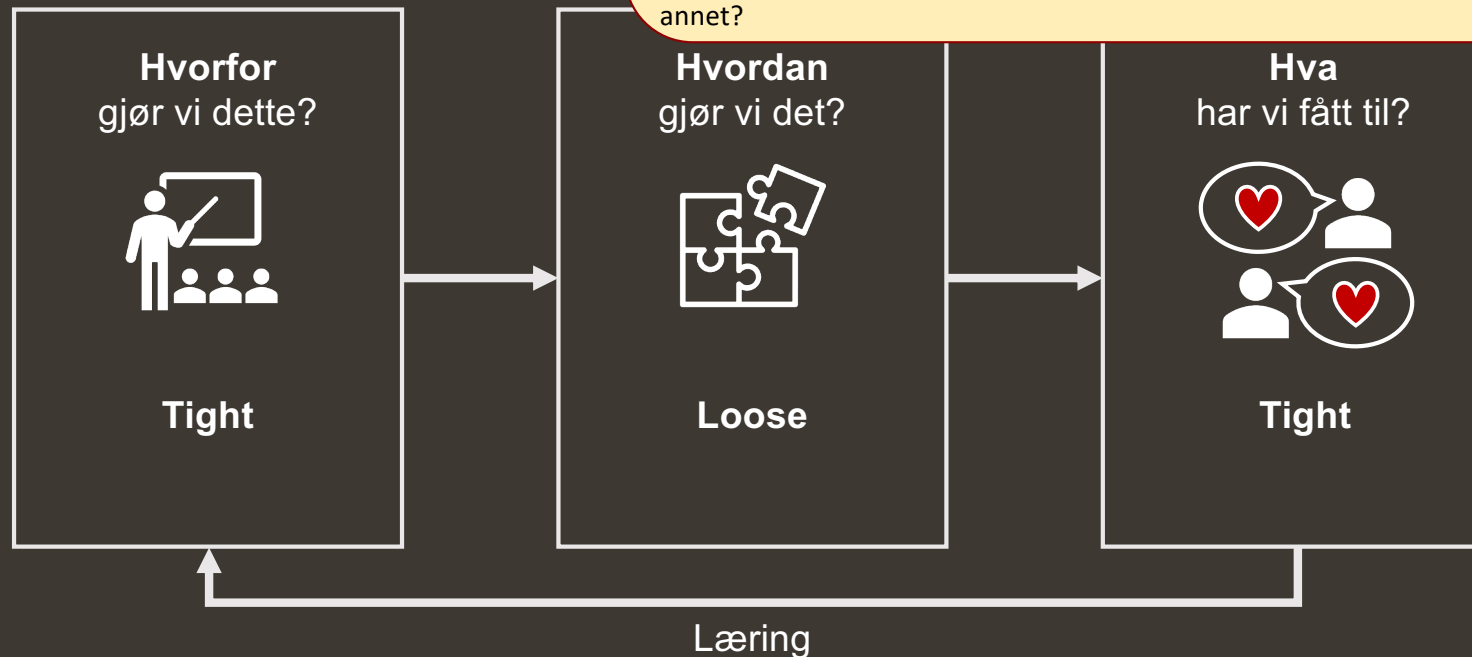
Fart er ingen erstatning for retning

Tight Loose Tight

Vi praktiserer Tight Loose Tight (TLT)-ledelse i NAV. Med TLT får vi tydelig retning og tydelige mål, vi får myndiggjorte team som tar egne valg, og vi måler og verifiserer effektene (fortløpende). **Tight nr 1:** Det er et lederansvar å sette retning, så her er ledere tett på (tight). Mål og effekter som ønskes oppnådd hører hjemme her, og de defineres gjerne i samarbeid mellom ledelse og team.

Loose: Ledelsen slipper opp og lar teamet selv finne ut hvordan målene skal nås og problemer skal løses.

Tight nr 2: Ledelsen følger opp hvilke effekter som er oppnådd og hva teamet har fått til. Så lærer man og gjør eventuelle justeringer. Skal teamet fortsette, eller legge om kursen? Bør målet justeres? Er målet uoppnåelig, slik at teamet bør stoppe og heller fokusere på noe annet?



Sosioteknisk teknologiambisjon 2022-2025



**Vi optimaliserer for
endringsevne over
tid**

Vi bygger bærekraftige team med eierskap til problem og løsning

Vi bygger treffsikre løsninger ved å lære raskt gjennom hypoteser og eksperimentering

Vi gir team rammer som peker retning, tar ned kognitiv last og understøtter autonomien

Endringsevne er viktig på grunn av uforutsigbare og stadig endrede muligheter og omstendigheter. Det være seg nye teknologiske muligheter, høyere brukerforventninger, pandemi, krig, endrede lover og regler, andre politiske prioriteringer eller annet som gjør at vi må endre prioritering.

Sosiotekniske mål

Vi tror teamtyper hjelper oss å:

- Gjøre det tydeligere for teamene hva de skal levere og for hvem.
- Gjøre hensikten med det enkelte team tydelig for alle andre. Det gir økt forutsigbarhet i hva man kan forvente. Det bidrar til å bygge tillit på tvers.
- Tilrettelegge bedre for effektiv samhandling mellom team.

1

Alle team vet hvilken teamtype de er

2

Alle team har en plan for samhandlingsmønstre med andre team

3

Alle team er bevisst sitt ansvarsområde og avgrensninger mot andre team og domener

Hvordan og hvorfor team samhandler er en nøkkelindikator for hvor godt en organisasjon presterer. Vi ønsker at teamene skal være bevisste på dette. Tydelige og fornuftige grenser mellom team er også nødvendig for god fart og flyt.

La oss se litt på hvordan vi har
valgt å organisere teamene våre.

Folk og
tankesett



Organisering

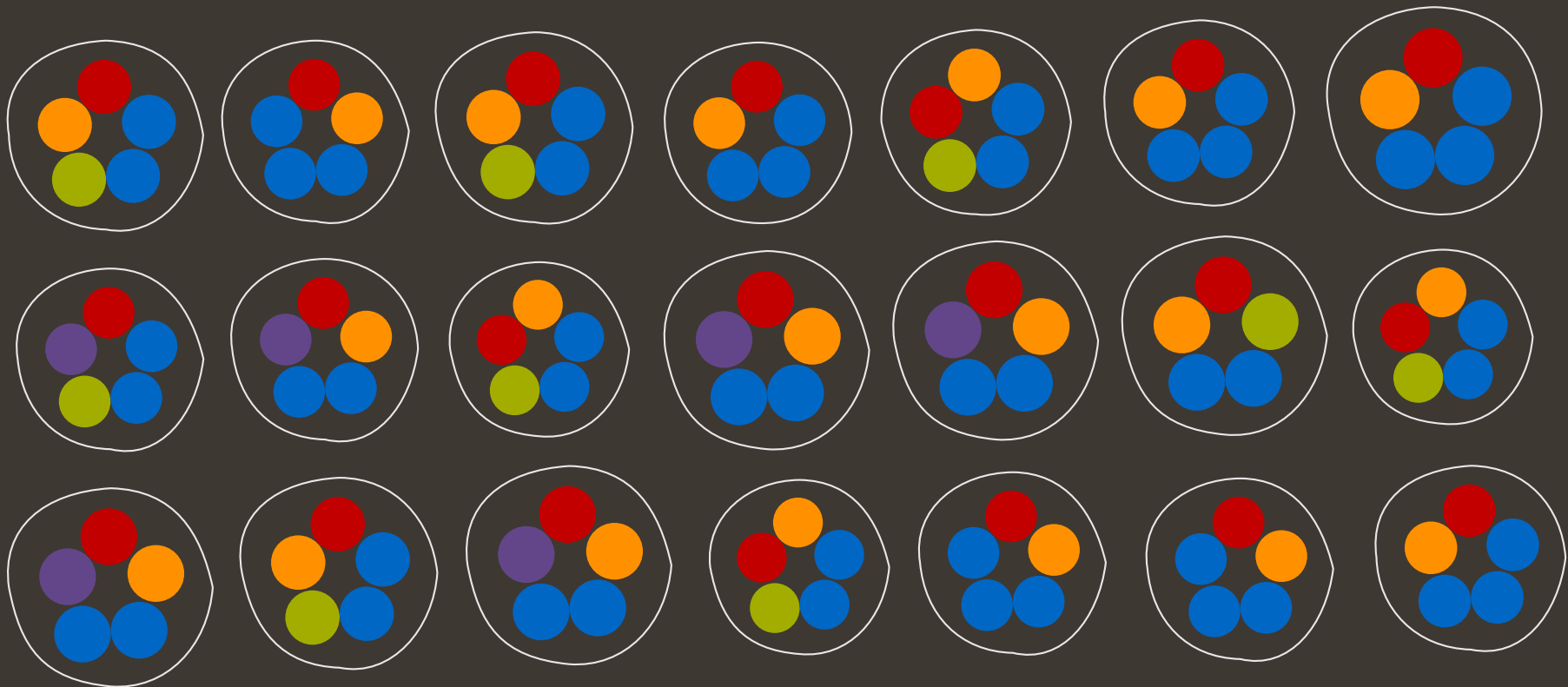


Plattformer

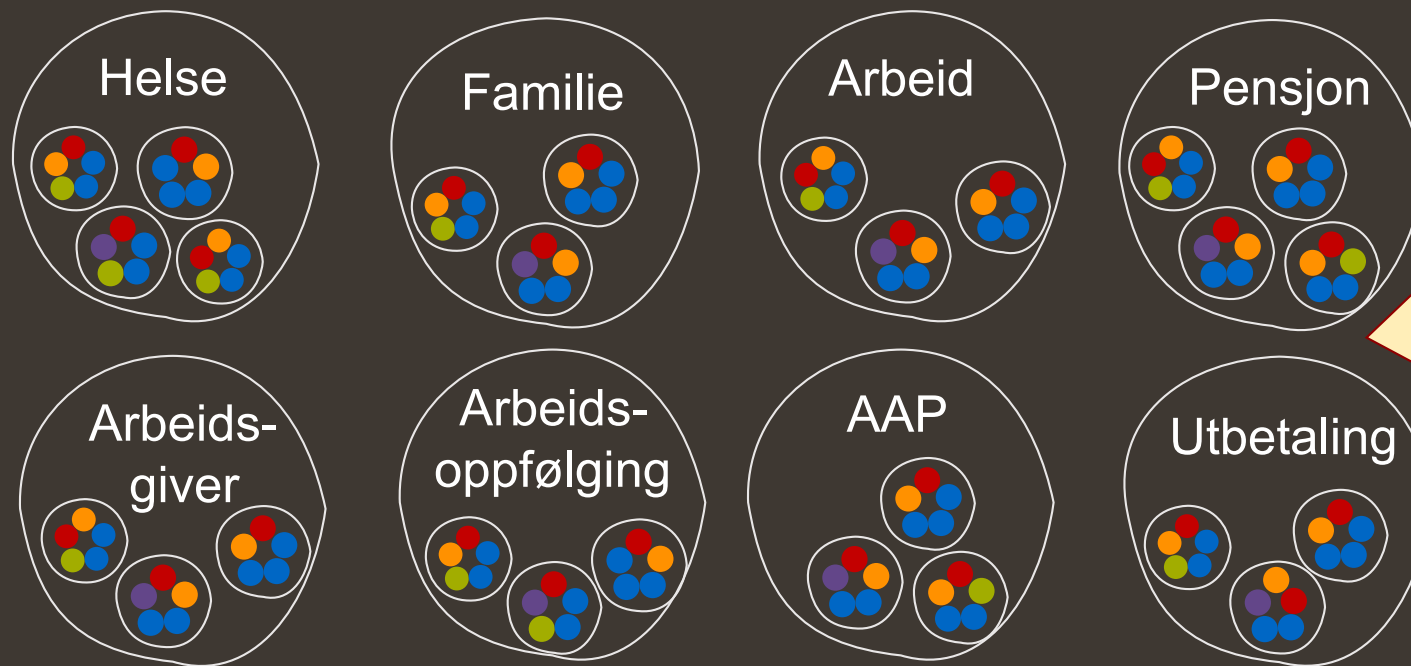


~120 tverrfaglige produktteam

I NAV har vi mange team, og i tillegg til å bygge gode team, så må vi rigge team-grupperinger på en fornuftig måte også.

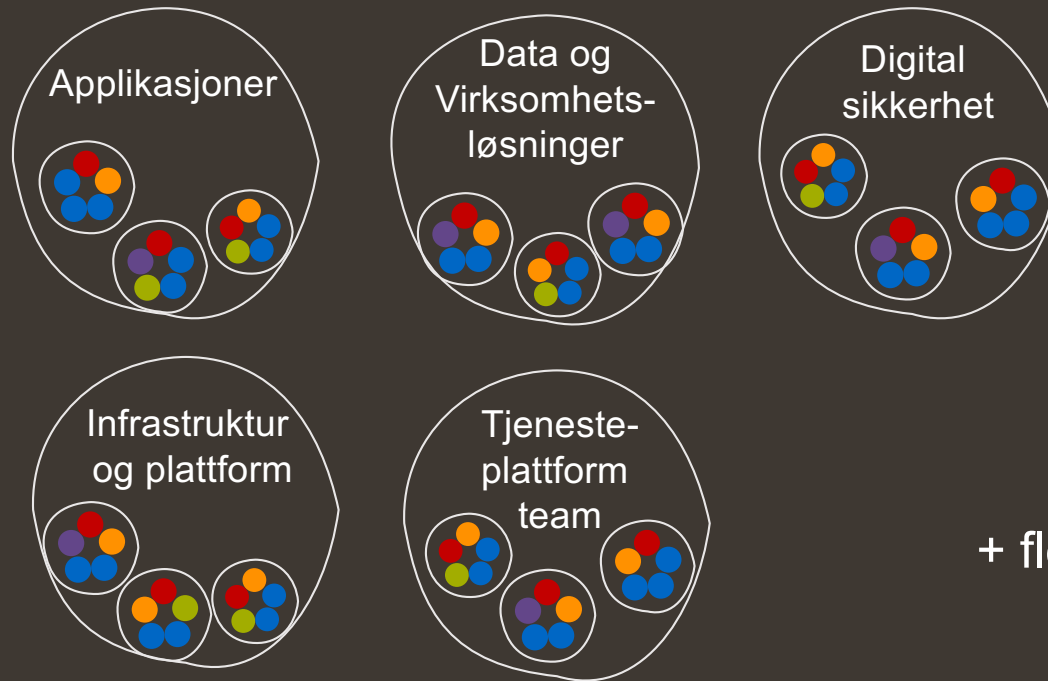


Domenedrevet organisering i produktområder



Produktområdene gjenspeiler NAV sin kjernevirksomhet. Et produktområde består av en mindre gruppe med team som alle jobber innenfor et avgrenset domene.

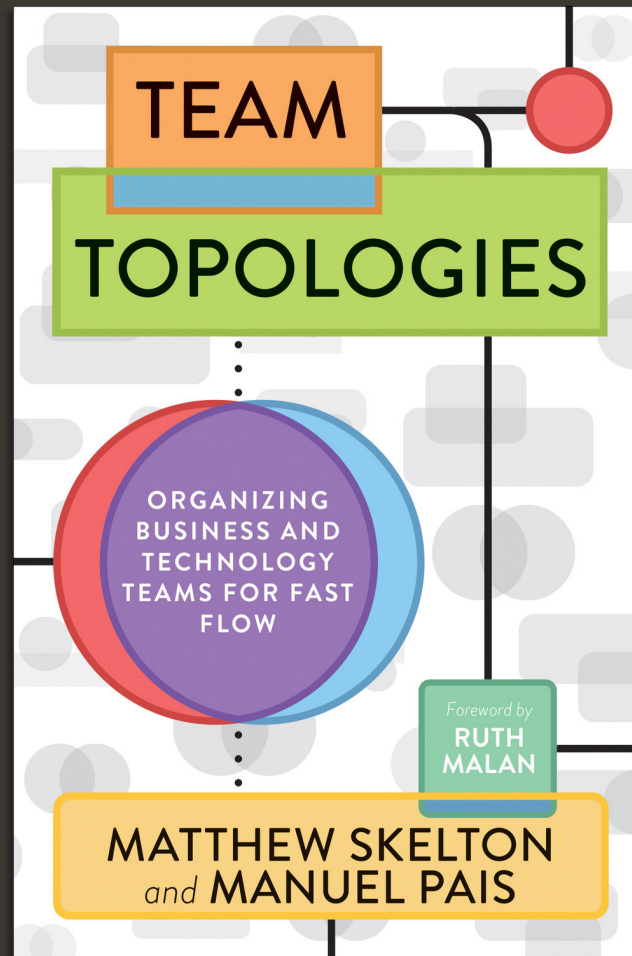
...og IT-områder



Ikke alt hører naturlig hjemme i ett produktområde (PO). Derfor har vi også IT-områder. Her finner man blant annet plattformene, som brukes på tvers av PO-er.

Verdt å merke seg er at selve arbeidsformen skal være det samme i både produktområder og IT-områder: produkt-tankesett, brukerens behov først, kontinuerlig leveranse, tverrfaglige og myndiggjorte team osv.

+ flere team utenfor områder



Vi har mange team. Noen karakteristika går igjen på tvers av de. Og da finnes det en bok som setter det i system og som gjør det enklere å forholde seg til: Team Topologies.

Deler av tankesettet fra denne boka var allerede på plass i NAV da boka kom (2019), men den har likevel gitt oss en bedre felles forståelse for tematikken og et vokabular som gjør det enklere å snakke om teamorganisering og hvordan teamene bør forholde seg til hverandre, hvordan ansvar fordeles osv.

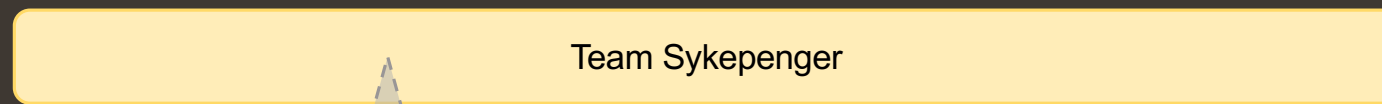
Topologi endres evolusjonært

Team Sykepenger

Et eksempel på hvordan vi anvender teorien:
Vi har et stream-aligned team som driver med sykepenger. Men dette domenet er ganske omfattende med mange detaljer i regelverket, så den totale kognitive kompleksitet er ganske stor.

Team Topologies gir oss nyttige mønstre for hvordan vi kan forenkle den kognitive lasten for slike team, ved å avlaste deler av ansvaret til andre team.

Topologi endres evolusjonært



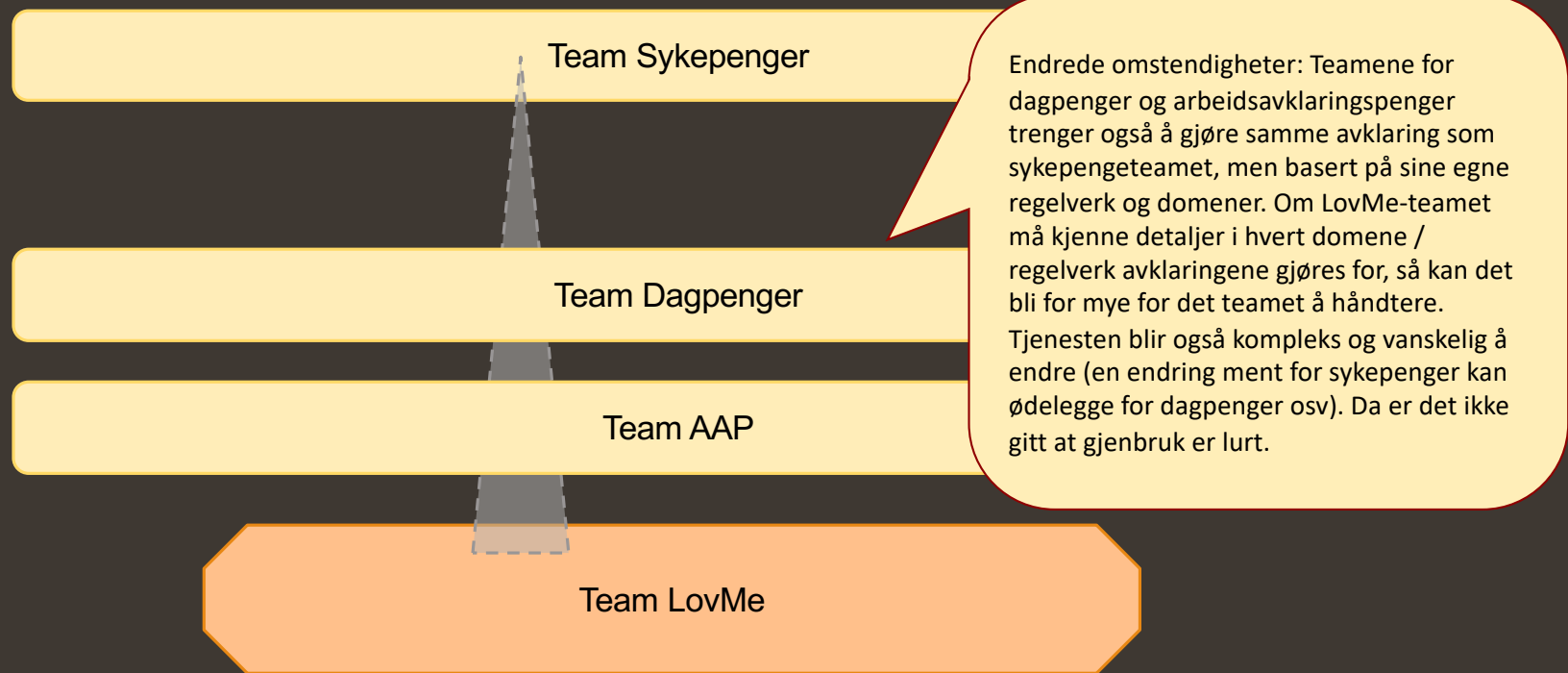
Team Sykepenger

Team LovMe

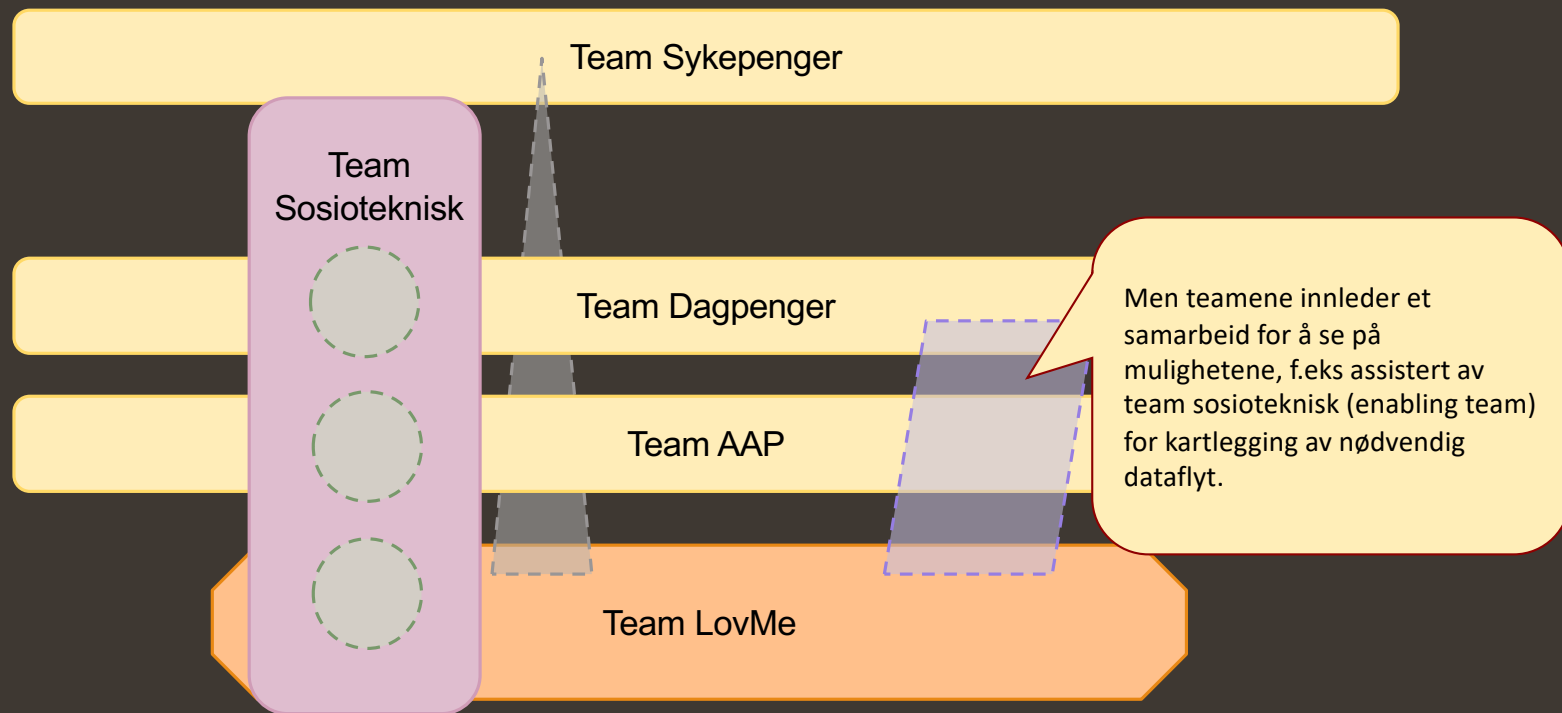
Sykepenger støtter seg på et subsystemteam som leverer en tjeneste som gjør en avklaring på hvorvidt den som søker om sykepenger faktisk har rett på ytelsen. Om man har rett til en ytelse kan for eksempel avhenge av hvor lenge man har vært medlem av folketrygden og hvorvidt man har jobbet i utlandet eller ei. Hvis man har eller har hatt arbeid i utlandet kan det hende ansvaret også ligger der og at et annets land lovverk skal gjelde. LovMe står for lovvalg og medlemskap.

Denne splitten gjør hverdagen til begge teamene enklere.

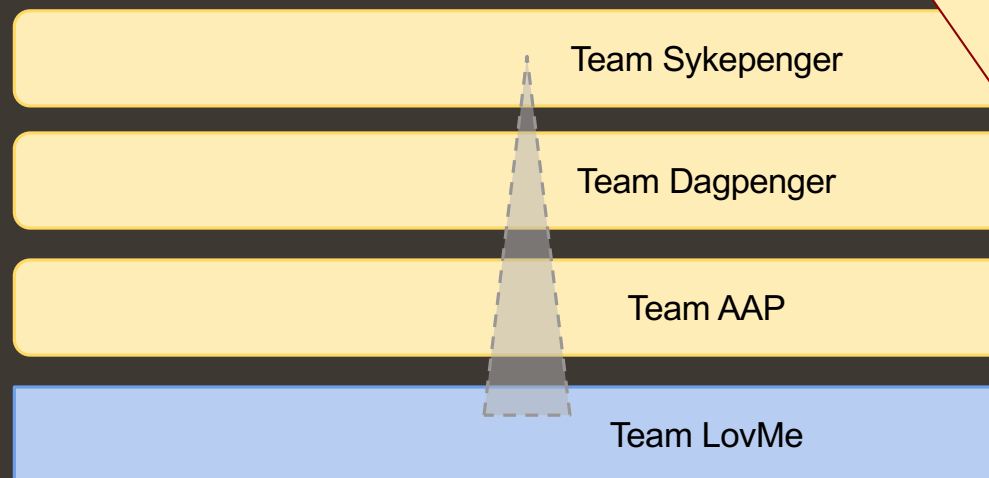
Topologi endres evolusjonært



Topologi endres evolusjonært



Topologi endres evolusjonært



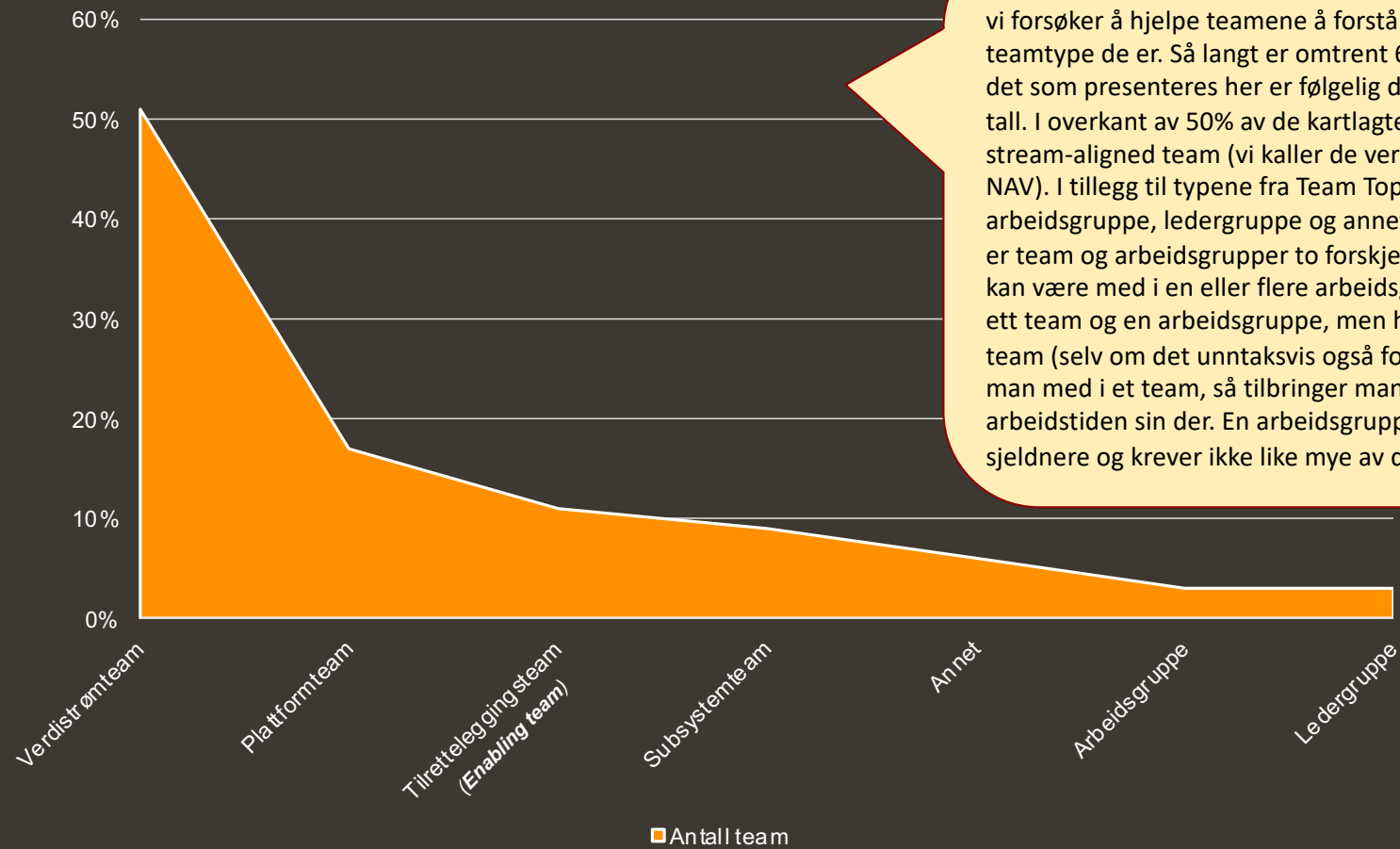
Dersom LovMe ikke trenger å vite noe de forskjellige regelverkene, men får instruks (input-parameter e.l.) fra den som ber om avklaring om hvor lang periode bakover i tid som skal sjekkes, så kan det være aktuelt at LovMe-tjenesten går fra å være et subsystem kun for sykepenger til å bli en plattformtjeneste som kan brukes på tvers av domener.

En plattform bør være domeneagnostisk. Det betyr at den ikke inneholder forretningslogikk eller trenger å vite noe om regelverk og annet som gjelder for de ulike appene / tjenestene som bruker den.

Om LovMe må kunne skille på regelverk e.l, slik at kompleksitet (i tjenesten) og kognitiv last (i teamet) øker, kan det være bedre at tjenesten fortsatt bare er myntet på sykepenger og at øvrige team må løse sine avklaringer på andre måter.

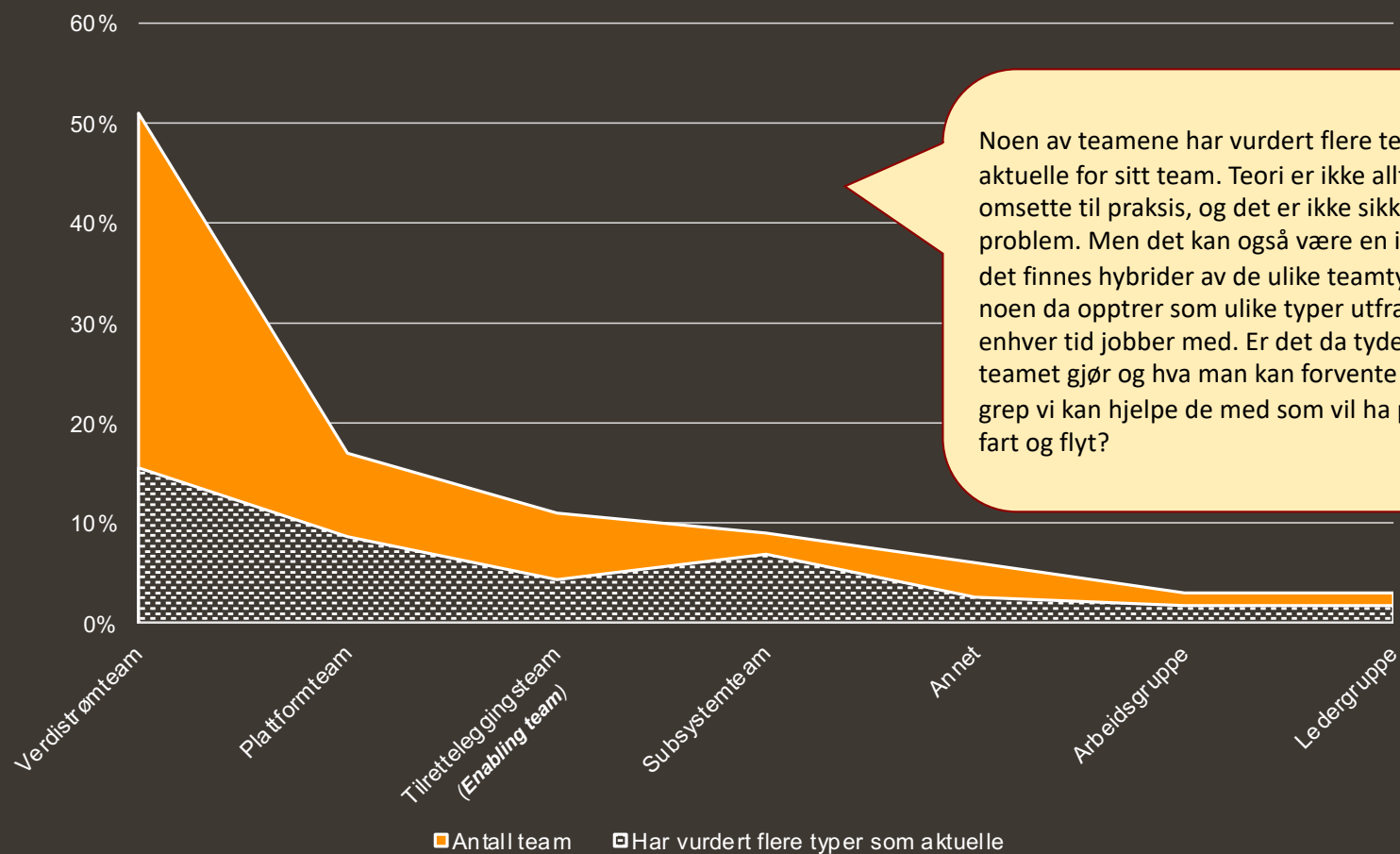
Kartlegging av teamtopologi i NAV

Teamtyper i NAV



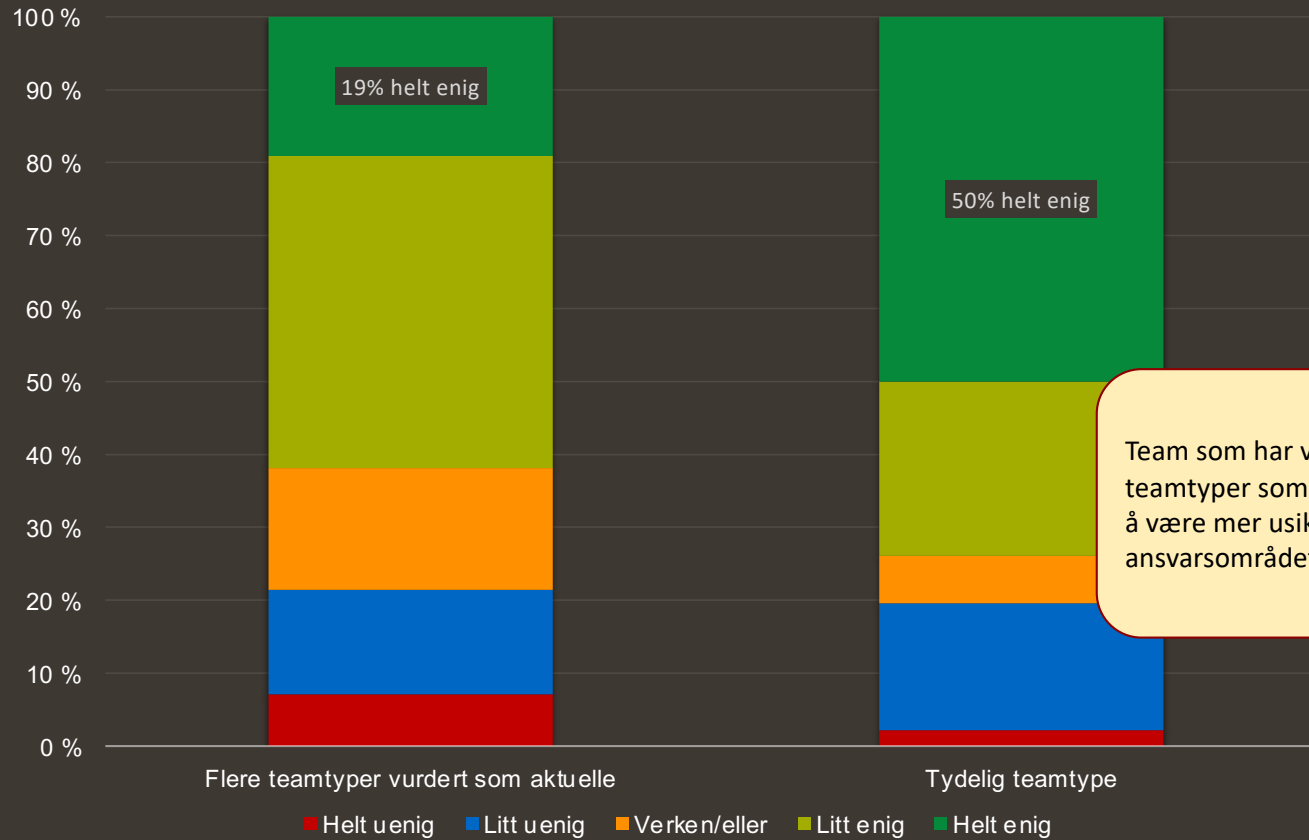
Akkurat nå kartlegger vi teamtopologi, samtidig som vi forsøker å hjelpe teamene å forstå hvilken teamtype de er. Så langt er omtrent 60% kartlagt, og det som presenteres her er følgelig da kun foreløpige tall. I overkant av 50% av de kartlagte teamene er stream-aligned team (vi kaller de verdistrømteam i NAV). I tillegg til typene fra Team Topologies har vi arbeidsgruppe, ledergruppe og annet. Slik vi ser det er team og arbeidsgrupper to forskjellige ting. Man kan være med i en eller flere arbeidsgrupper, eller i ett team og en arbeidsgruppe, men helst ikke i flere team (selv om det unntaksvis også forekommer). Er man med i et team, så tilbringer man majoriteten av arbeidstiden sin der. En arbeidsgruppe møtes typisk sjeldnere og krever ikke like mye av den enkelte.

40% har vurdert flere teamtyper som aktuelle



Noen av teamene har vurdert flere teamtyper som aktuelle for sitt team. Teori er ikke alltid like lett å omsette til praksis, og det er ikke sikkert det er noe problem. Men det kan også være en indikasjon på at det finnes hybrider av de ulike teamtypene, og at noen da opptrer som ulike typer utfra hva de til enhver tid jobber med. Er det da tydelig for andre hva teamet gjør og hva man kan forvente av de? Er det grep vi kan hjelpe de med som vil ha positiv effekt på fart og flyt?

Fornuftig avgrenset ansvarsområde?



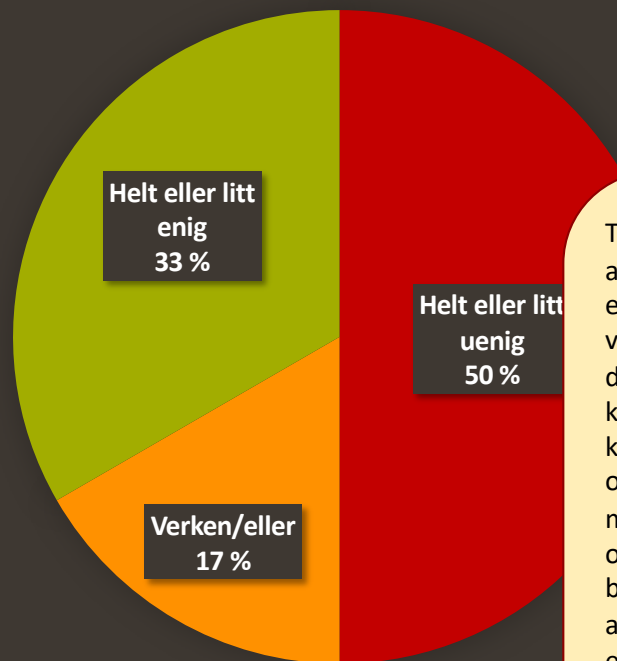
Team som har vurdert flere teamtyper som aktuelle, virker også å være mer usikker på ansvarsområdet sitt enn andre.

Domenekartlegging (Event storming)

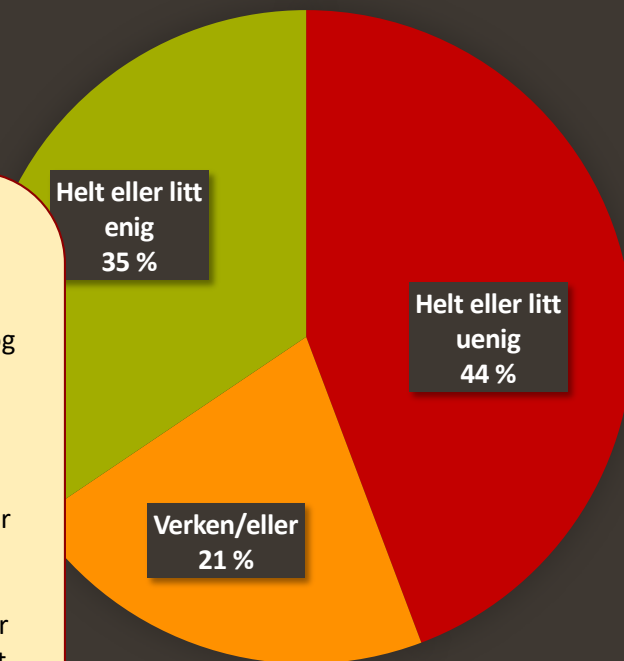
For å hjelpe team å bringe klarhet i domenet / ansvarsområdet sitt, bistår vi for eksempel med fasilitering av Event Storming-workshops. Det er en teknikk hvor man ser på hendelser i et system i kronologisk rekkefølge, fra venstre til høyre. Man kan da også få klarhet i hvordan systemet henger sammen med andre systemer. Det kan gjøre det enklere å se om ansvarsgrensene mellom teamene som håndterer de ulike tingene er fornuftig. Er det noe team A har ansvar for i dag som egentlig hører hjemme i B? Og er det noe i C som egentlig hører hjemme i A? I tillegg kan man gjennom Event Storming få klarhet i hvor et domene eller system kan splittes i to, dersom det er for stort for bare ett team.



Andre team sine API-er så velfungerende og godt dokumentert at vi kan være selvbetjent

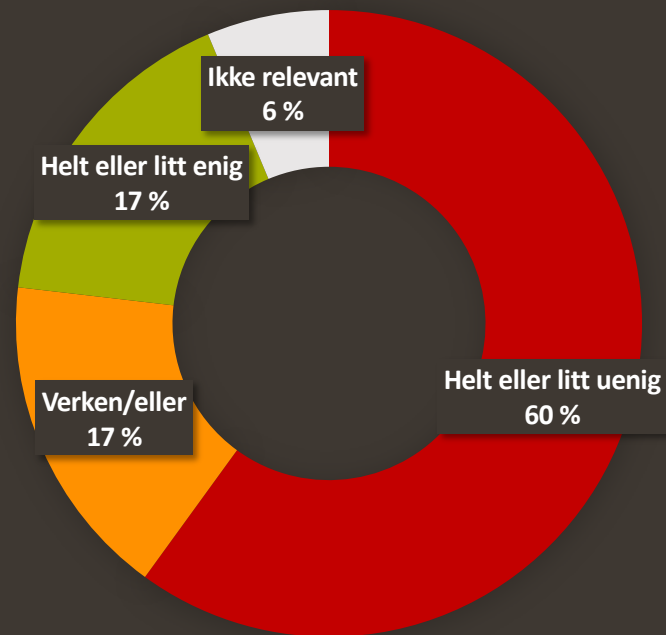


Våre API-er er så velfungerende og godt dokumentert at andre kan være selvbetjente



Team som tilbyr en tjeneste som andre konsumerer bør normalt etterstrebe at konsumentene kan være selvbetjente. Mindre friksjon, og dermed mindre behov for kommunikasjon mellom tilbyder og konsument, vil gi lavere kognitiv last og økt fart og flyt i begge leire. Hvor mye og hvor nøye man dokumenterer og hvor mye man investerer i god brukervennlighet, vil alltid være en avveining av flere ting. Men vi ser her en indikasjon på potensial for økt fart og flyt også blant teamene i NAV.

Vi kan enkelt finne informasjonen vi trenger fra andre team uten å måtte ta direkte kontakt med dem



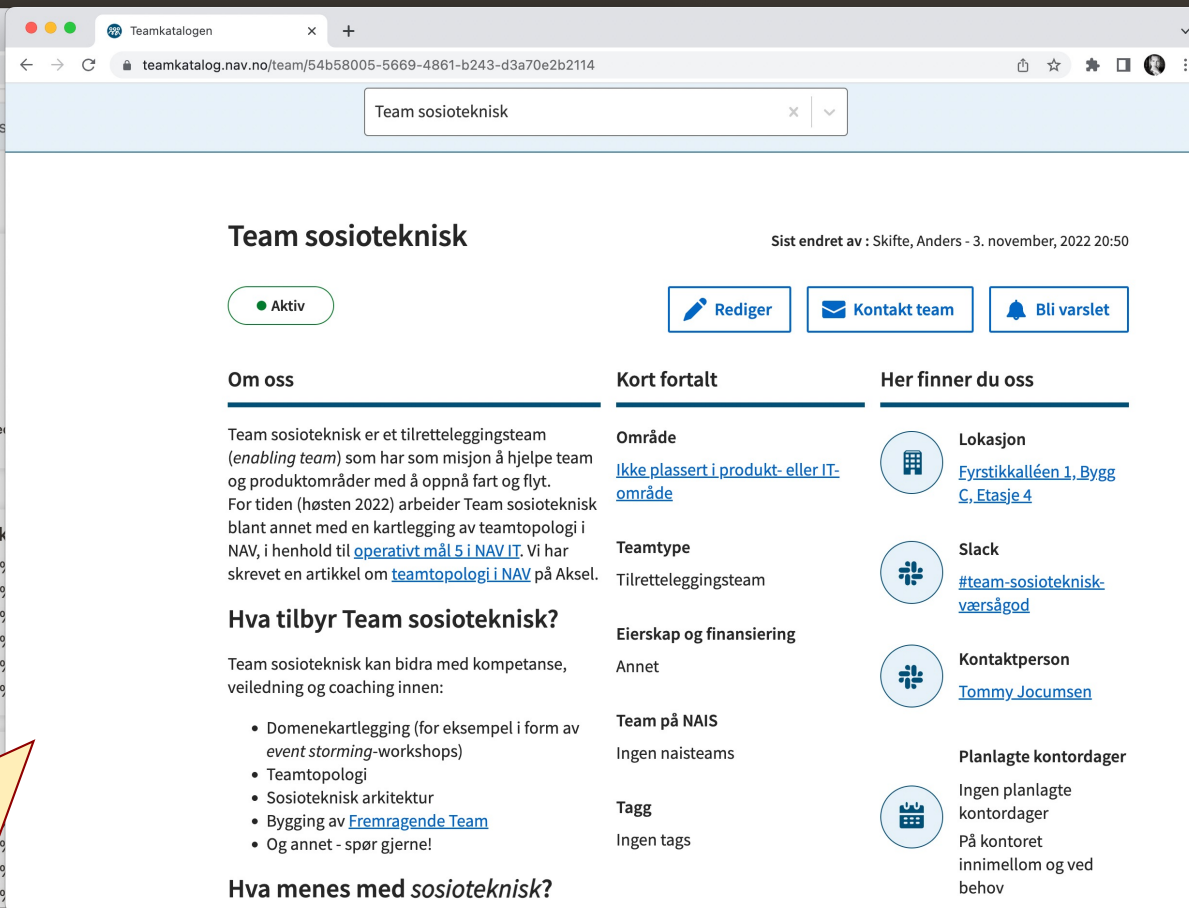
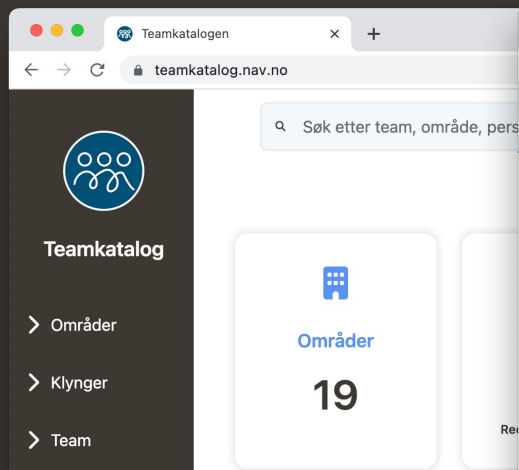
Denne må ses i sammenheng med forrige slide. En stor andel av teamene sier det ofte ikke oppleves enkelt å finne informasjonen de trenger fra andre, uten å ta kontakt med de. Det er ikke nødvendigvis alltid et problem, men likevel en indikasjon på at det er noe å hente mtp fart og flyt også her.

Om man bruker Slack e.l, så kan et grep være å ha mest mulig av kommunikasjonen i åpne kanaler slik at flere får tilgang til samme informasjon. Kanskje det blir enklere for team å se hvilke spørsmål som går igjen og hva som ofte oppleves uklart – enn om spørsmål og svar går i direktemeldinger? Hva mer kan man gjøre? Kanskje noe å ha i et Team-API?

Et Team «API» er ikke et API i teknisk forstand, men noe som gir omverden nyttig informasjon om teamet. Det kan være, og er ikke begrenset til: Hva teamet gjør, hva de har ansvar for, hvordan man kan komme i kontakt med de, hvem som er med i teamet, hvilke produkter de har ansvar for, hva målene deres er, hva status er nå, hvor dokumentasjon og kode ligger, hvilke andre team teamet samhandler med og på hvilken måte (vise avhengigheter) og lignende.

Team-API kan også fungere som springbrett til informasjon, og kan i så måte gjerne inneholde lenker. Det kan være noe så enkelt som en Wiki-side, men selve team-API-et **må** være lett å finne. Et Team-API som ingen finner er utgjør ingen forskjell.

Team «API»



Teamkatalogen i NAV er i praksis en samling av Team-API-er. Hvert team har sin egen side, bestående av både strukturerte og ustrukturerte data. Teamkatalogen er søkbar. Teamene er selv ansvarlige for å holde egen informasjon oppdatert. Vi tror potensialet i teamkatalogen kan utnyttes enda bedre (mer innhold og pekere til annet innhold), og at det kan gjøre det enklere å finne informasjonen man søker. Teamkatalogen er et internt utviklet produkt som forbedres og videreutvikles kontinuerlig.

Bærekraftige team



*Et bærekraftig team opprettholder en høy grad av **motivasjon, psykologisk trygghet og tilfredshet** blant medlemmene, samtidig som det holder **god fart over tid**. Teamet er **fleksibelt** og tåler fravær eller utskifting av enkeltpersoner.*

NAV ønsker bærekraftige team med eierskap til problem og løsning. Men hva er et *bærekraftig* team? Her er vår egen definisjon.

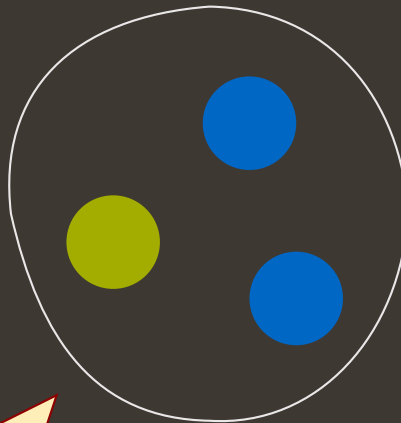
Det vanskelige men viktige spørsmålet er jo: Hvordan bygger vi slike team? La oss se litt på det.

Tverrfaglig, myndiggjort og selvstendig



Beslutning

For å opprettholde motivasjon og fart må teamet oppleve mestring, mening og myndiggjøring. Med myndiggjøring mener vi at teamene har myndighet til å ta egne valg, innenfor sine rammer, for å nå sine mål. Teamene må være tverrfaglige og besitte all nødvendig kompetanse / kapabiliteter som kreves. Da bør for eksempel ikke beslutningsmyndighet eller kompetanse på hvordan tolke et regelverk ligge utenfor teamet. Det vil føre til at teamet må spørre om lov og vente på avklaringer, slik at fart og flyt bremses.



Regelverkstolkning

Tverrfaglig, myndiggjort og selvstendig

Vi må i stedet forsøke å dytte dette ansvaret (f.eks beslutning, tolkning av regelverk e.l) inn i teamene, slik at teamene kan opptre mest mulig selvstendig og uavhengig av andre (som sitter utenfor teamet). Folk utenfor teamet har som regel sin egen timeplan som ikke nødvendigvis er i synk med teamet. Da oppstår kø og venting.



Stabilt, men likevel fleksibelt

Psykologisk trygghet og tillit blant teammedlemmer er viktig for at et team skal fungere bra. Det er ikke noe man oppnår over natta. Det tar tid å bli kjent og bygge relasjoner, og er noe som kontinuerlig må pleies. Derfor er det en fordel om et team er stabilt over tid (blant annet).

Samtidig må teamet være fleksibelt nok til å tåle fravær eller utskifting av enkeltpersoner. Hva om noen blir sykemeldt, slutter, eller ønsker å bytte team? Hva om organisasjonen må gjøre omprioriteringer og flytte på folk?

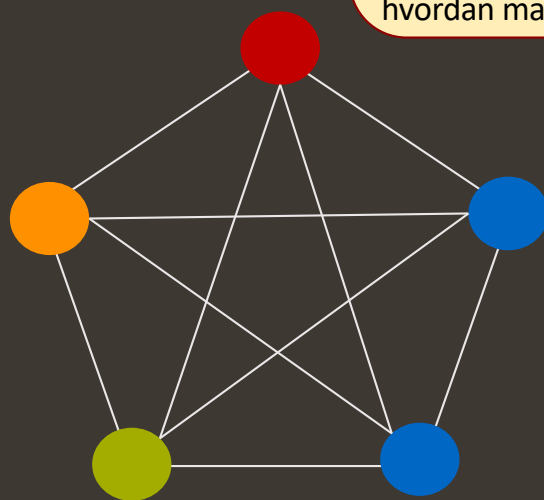


Tiltak for å gjøre seg mindre sårbar ved endring kan f.eks være par/mobprogrammering eller andre former for kunnskaps- og kompetansedeling i teamet.

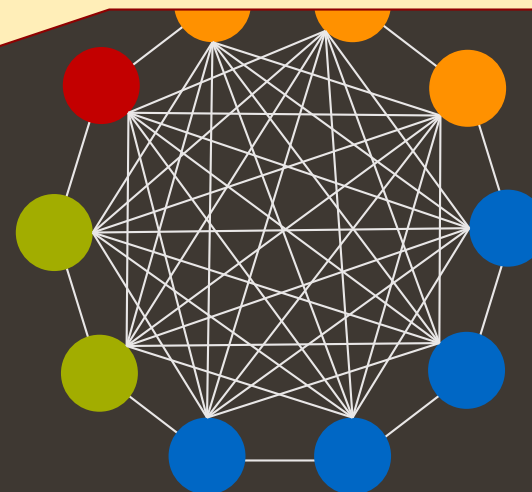
Om vi i tillegg skriver kode på en måte som gjør at flere kan forstå det (obvious code over clever code), får vi økt bærekraft i både team og systemer.

Teamstørrelse

Jo større team, jo flere relasjoner og potensielle kommunikasjonslinjer. En dobling av antall medlemmer gir en mangedobling av antall linjer på kryss og tvers i teamet. Team Topologies antyder at ideell teamstørrelse er 5-9 personer. Amazon snakker om sine «two pizza teams» (to pizza er nok for teamet). Store team gir mer overhead, og det blir mer krevende å bygge psykologisk trygghet og tillit. Om ansvarsområdet øker i omfang kan det være fristende å øke teamstørrelsen, men det har altså også en potensiell slagside (om teamet blir for stort). Når omfang øker, så bør man heller se på hvordan man kan splitte i flere mindre team på en fornuftig måte.

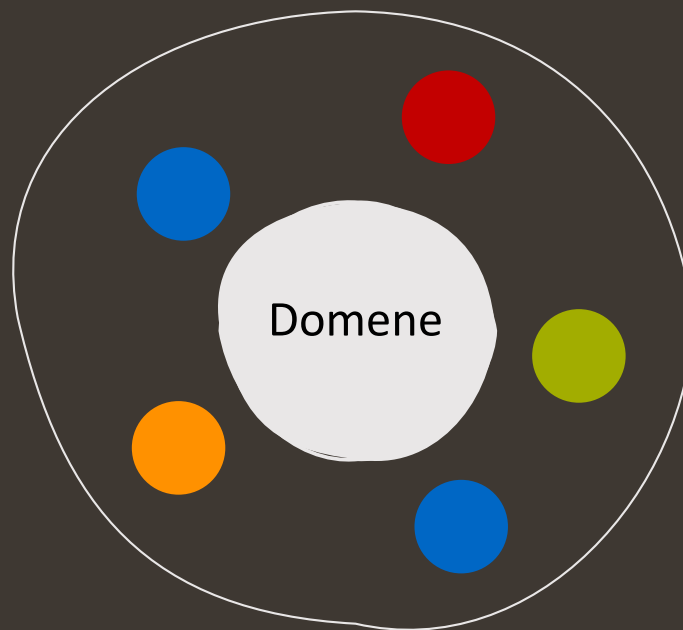


5 personer
10 relasjoner



10 personer
45 relasjoner

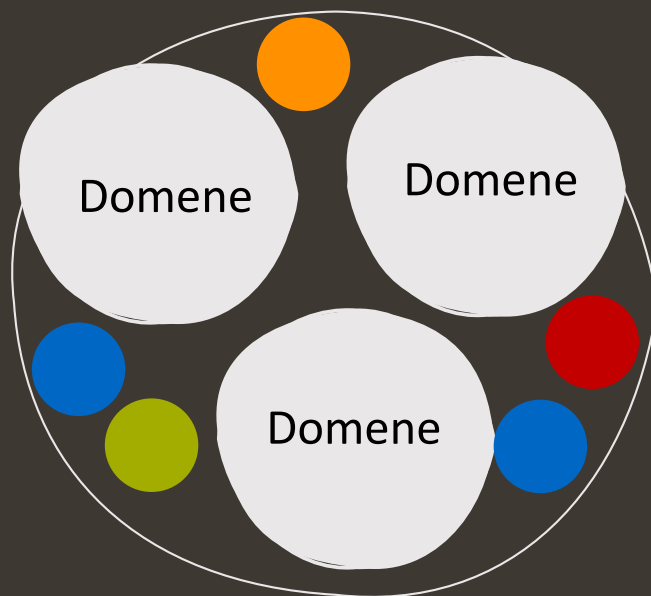
Ansvarsområde tilpasset kognitiv kapasitet



Bonus: Utelatt fra opprinnelig presentasjon pga lengde.

Det er i tillegg viktig at ansvarsområdet får «plass i hodene» til teamet. Det må ikke være større enn hva teamet har kognitiv kapasitet til å håndtere.

Kognitiv overbelastning

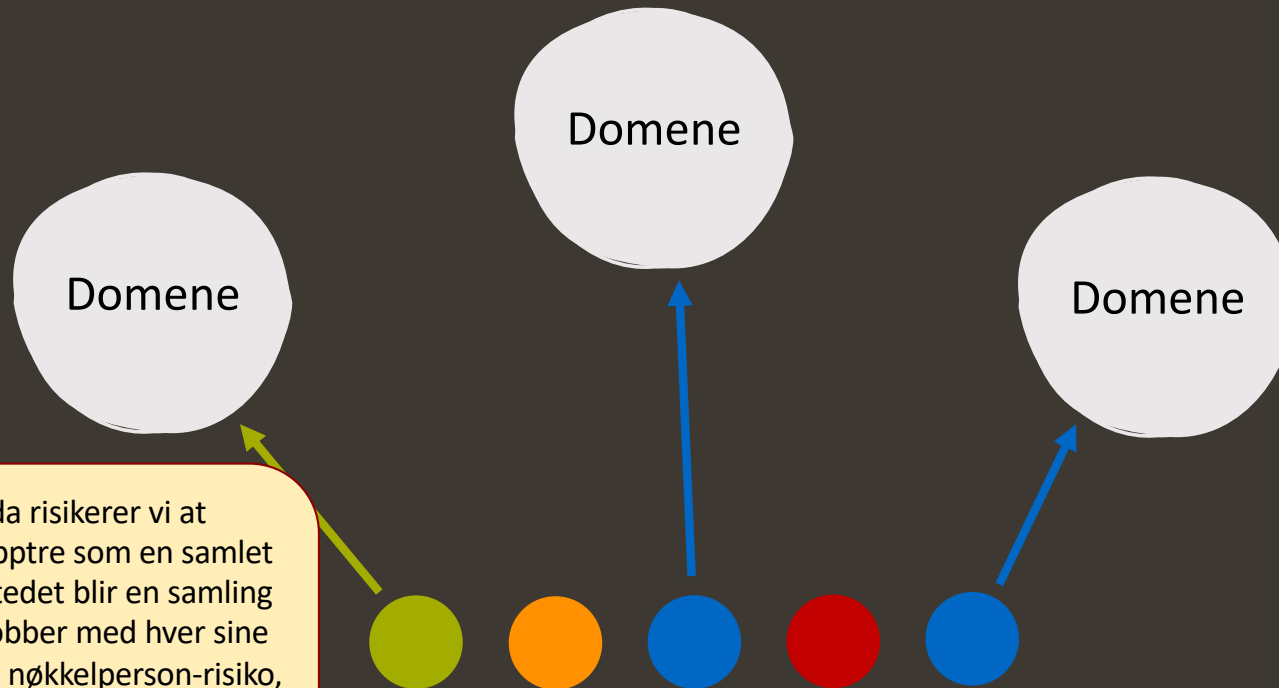


Bonus: Utelatt fra opprinnelig presentasjon pga lengde.

For stort, spredt eller komplisert ansvarsområde gir kognitiv overbelastning. Teamet mister oversikt og havner bakpå. Det bremser fart og går ut over kvalitet.

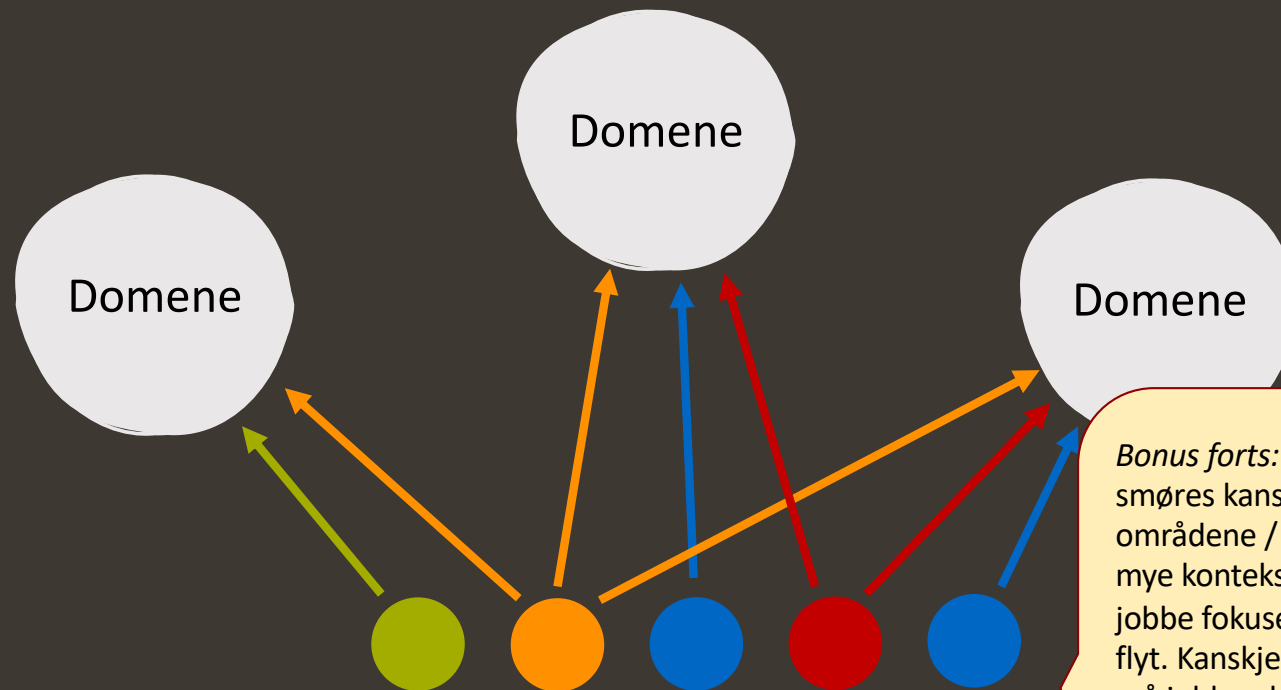
Et spredt ansvarsområde kan i tillegg føre til at teamet dras i ulike retninger og at prioritering følgelig også blir vanskelig.

Teamet slutter å opptre som en enhet



Bonus forts: .. Og da risikerer vi at teamet slutter å opptre som en samlet enhet og at det i stedet blir en samling av individer som jobber med hver sine ting. Vi bygger høy nøkkelperson-risiko, ettersom den enkelte blir mer spesialisert innen «sitt» område. Da blir vi sårbare for fravær eller utskifting av enkeltpersoner.

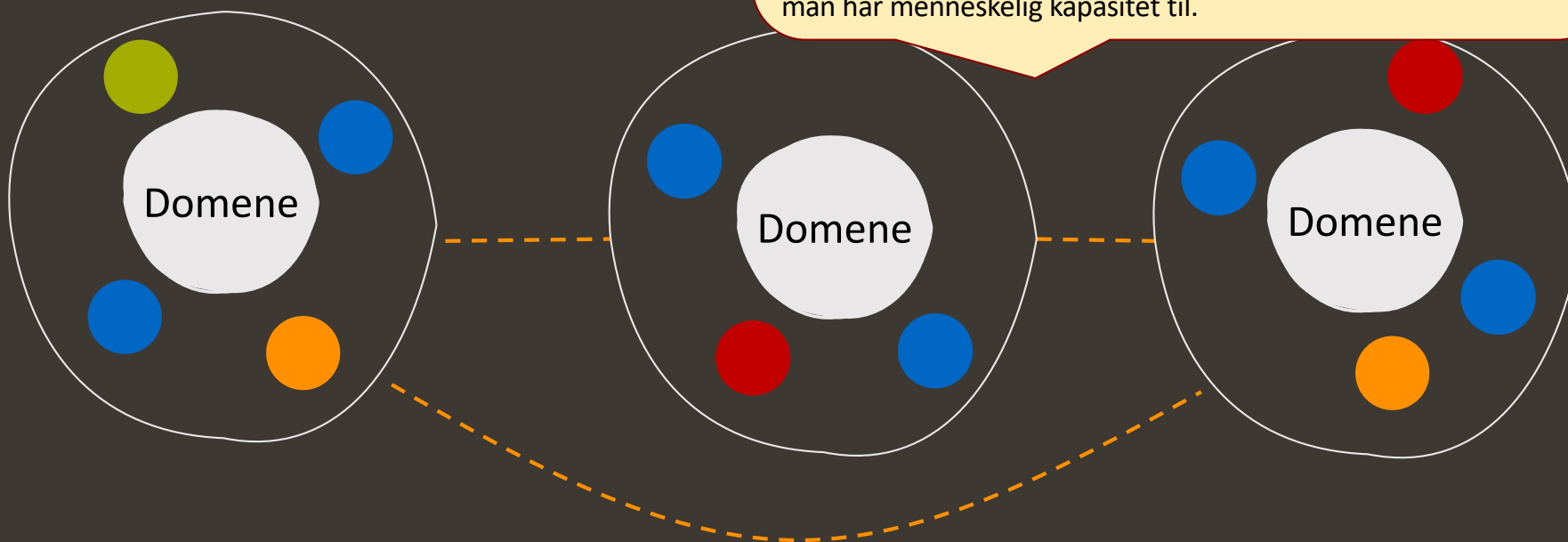
Teamet slutter å opptre som en enhet



Bonus forts: Noen teammedlemmer smøres kanskje tynt utover de ulike områdene / domenene og opplever mye kontekst-bytting. De kan ikke jobbe fokusert. Det hemmer fart og flyt. Kanskje føler man da at man må jobbe ekstra hardt og lenge, for å henge med? Det er veien til utbrenthet, og ikke bærekraftig.

En bedre situasjon

Bonus forts: Det kan være bedre å splitte team og ansvarsområde i mindre, separate deler. Splitten bør gjøres på en måte som ikke krever mye kommunikasjon i mellom teamene (etter splitten). Mye kommunikasjon er et tegn på at splitten ikke er gjort riktig. Om man ikke har tilstrekkelig antall folk blir det heller et spørsmål om prioritering og hvordan begrense antall ting som går i parallell. De fleste organisasjoner har større ambisjoner enn hva man har menneskelig kapasitet til.



Folk og
tankesett



Organisering



Plattformer



Plattform som produkt



Tiltalende og helhetlig



Kontinuerlig læring og forbedring



Selvbetjent og friksjonsfritt



Lett å gjøre rett



Domeneagnostisk

Vi ser også på plattformene våre som digitale produkter for øvrig. Forskjellen er at plattformsteam ser på andre team som sine potensielle kunder. Plattformen må være så attraktiv at teamene *velger* å bruke den. Det er ingen tvang. Det betyr at plattformteamet må ha innsikt i teamenes behov og aktivt søke feedback. Det har høy prioritet å sørge for at brukerne (de andre teamene) kan være selvbetjente.



NAIS

Applikasjonsplattform

NAIS er NAVs applikasjonsplattform, bygget på toppen av Kubernetes. Ca 1600 applikasjoner kjører på den i produksjon. Plattformen gjør det enkelt for teamene å utvikle, drifte og forvalte applikasjonene sine.

Hva er NAIS?



Byggeklosser og verktøy for utvikling som gjør det lett å gjøre rett



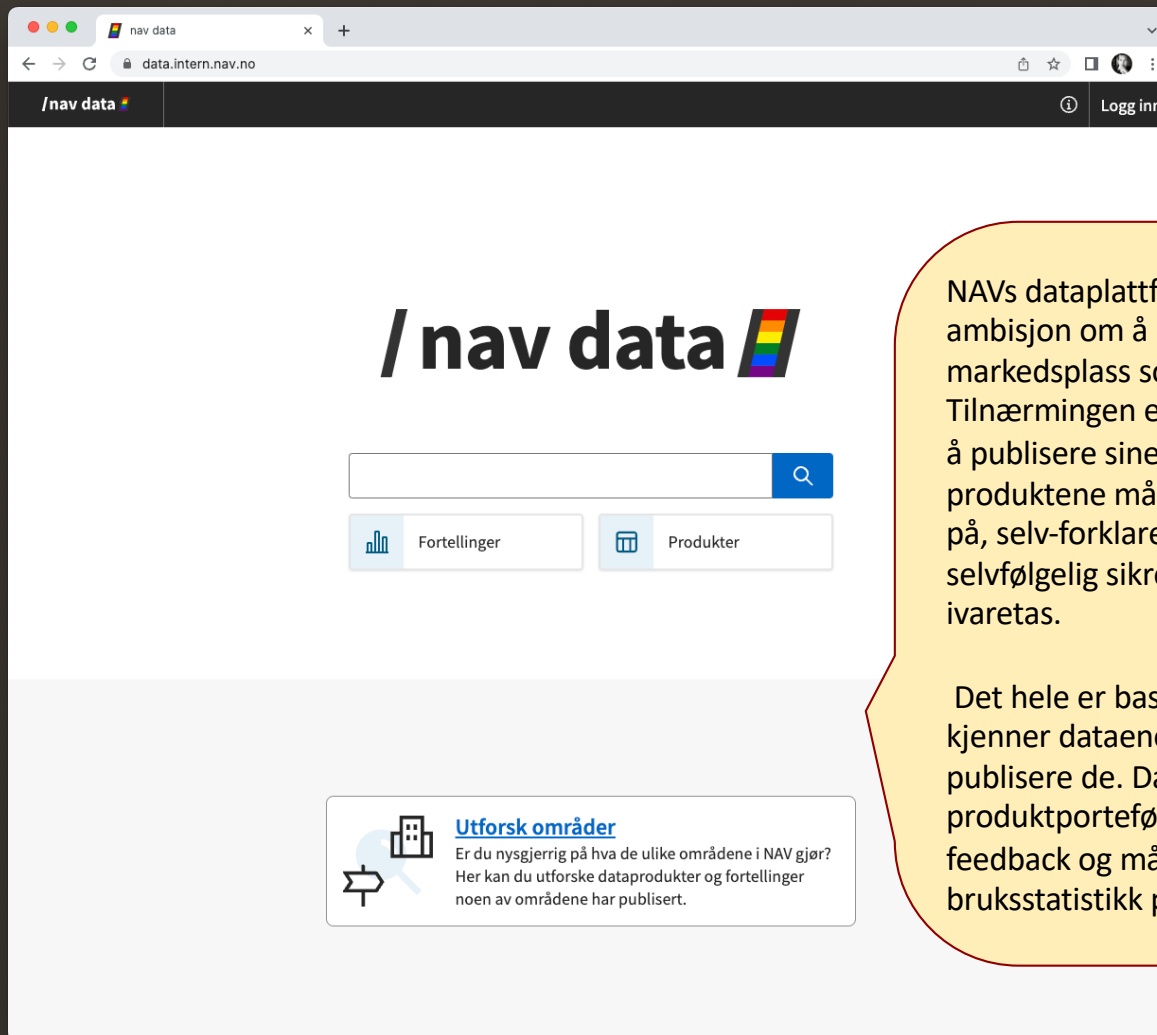
Robust og sikkert kjøretidsmiljø som forenkler driftsoppgavene



Verktøy for monitorering, varslinger og feilsøking

Mer info på nais.io

NAIS er en *opinionated* plattform. Teamet har en sterk mening om hvordan ting burde gjøres og hva som burde tilbys. Om team velger å gjøre ting å NAIS-måten, så får de mye gratis. Det blir lett å gjøre rett, og vi får en slags normering av arkitektur uten å ha et sentralisert teknologiråd. Det er mulig å bruke annen teknologi enn hva som tilbys ut-av-boksen i NAIS (som nevnt ingen tvang), men da må teamet fikse det selv. Det blir mer jobb.



nav data

data.intern.nav.no

/ nav data

Logg inn

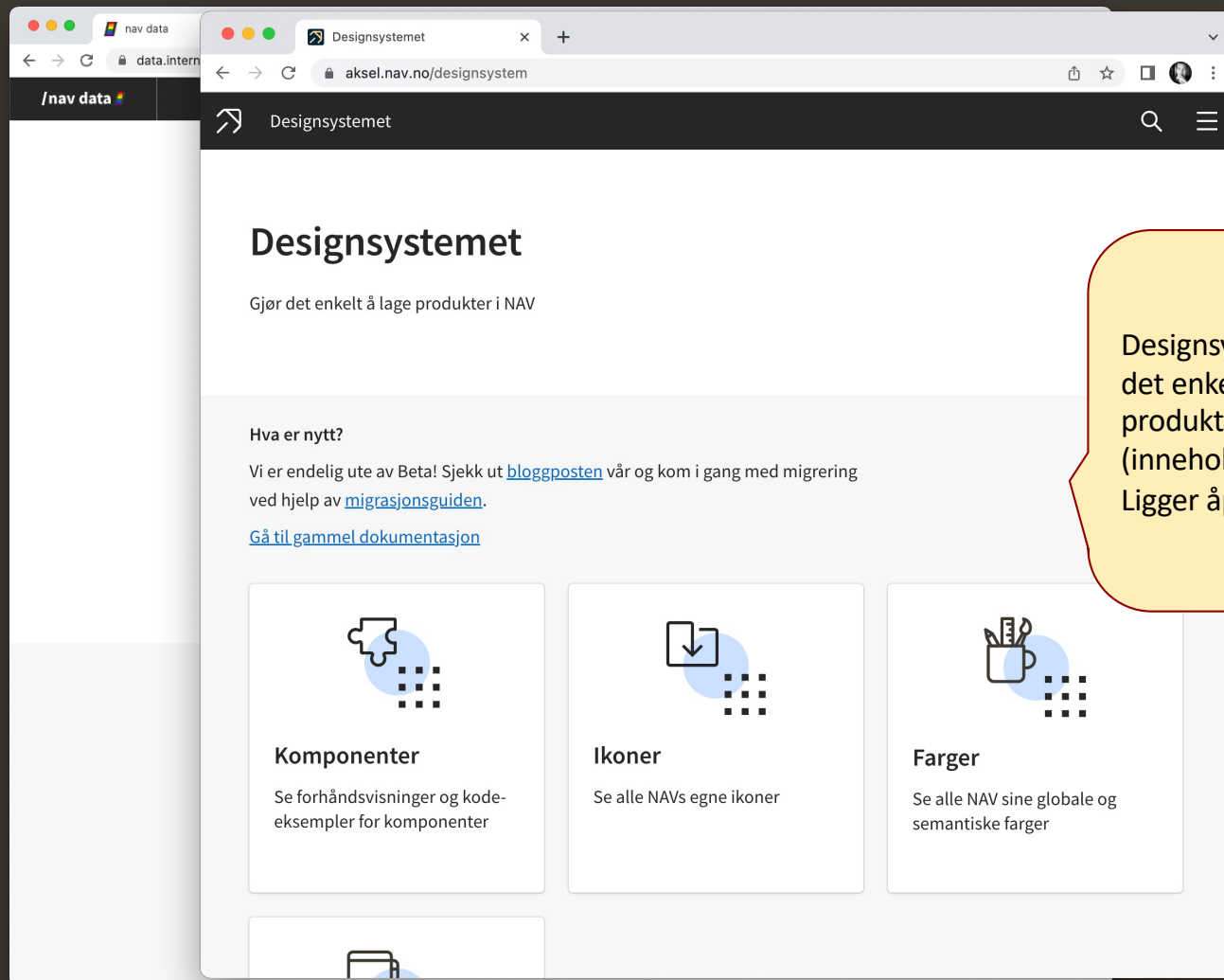
Fortellinger

Produkter

Utforsk områder
Er du nysgjerrig på hva de ulike områdene i NAV gjør?
Her kan du utforske dataprodukter og fortellinger
noen av områdene har publisert.

NAV's dataplattform NADA (NAV Data) er en viktig del av NAV's ambisjon om å bli mer datadrevet. NADA tilbyr en data-markeds plass som gjør det lett å finne, bruke og dele data. Tilnærmingen er desentralisert. Teamene har selv ansvar for å publisere sine data, i form av dataprodukter. Data-produktene må være mulig å finne, adresserbare, til å stole på, selv-forklarende, mulig å koble med andre data og selvfølgelig sikre. Tilgangskontroll og personvern med mer må ivaretas.

Det hele er basert på data mesh-modellen, hvor de som kjenner dataene best (altså teamene) også har ansvar for å publisere de. Dataproduktene inngår i teamenes produktportefølje. Som ved digitale produkter for øvrig er feedback og målinger viktig, og plattformen tilbyr derfor bruksstatistikk per dataprodukt.



Designsystemet

Gjør det enkelt å lage produkter i NAV

Hva er nytt?

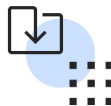
Vi er endelig ute av Beta! Sjekk ut [bloggposten](#) vår og kom i gang med migrering ved hjelp av [migrasjonsguiden](#).

[Gå til gammel dokumentasjon](#)



Komponenter

Se forhåndsvisninger og kode-eksempler for komponenter



Ikoner

Se alle NAVs egne ikoner



Farger

Se alle NAV sine globale og semantiske farger

Designsystemet er en plattform som gjør det enkelt å sørge for helhet på tvers av produkter i NAV. Det er funksjonelt (inneholder kode) så vel som visuelt. Ligger åpent tilgjengelig på aksel.nav.no.

Status.nav.no er under oppbygging og store deler vil bli åpnet for allmenheten. Plattformen viser status på NAVs digitale tjenester, og på sikt også hvilke tjenester som avhenger av hvilke andre tjenester osv. Også her er ansvaret distribuert, og ikke sentralisert. Teamene har selv ansvar for at riktig status vises for deres produkter og tjenester.

The screenshot shows the 'Status digitale tjenester' website for a 'Privatperson' (private individual). The page features a navigation bar with links for 'Internett', 'Privatperson', 'Arbeidsgiver', 'Samarbeidspartner', and 'Vaktor'. The main content area displays a green status message: 'Alle våre systemer fungerer normalt'. Below this, there is a grid of service categories, each with a question mark icon and a dropdown arrow. The categories are: 'Arbeidssøker eller permittert', 'Trenger hjelp til å komme i jobb', 'Trenger hjelp til å fullføre utdanning', 'Søker jobb', 'Har mistet noen i nær familie', 'Pensjon', 'Venter barn', and 'Alene med barn'. A link 'Ekspander områder' is visible to the right of the grid.

Hovedpunkter oppsummering



Hvordan redusere kognitiv last for økt fart og flyt?



Hvordan bygge bærekraftige team?



Hvordan gjør vi det lett å gjøre rett?



Hvordan gir vi teamene tydelig retning?

detsombetyrnoe.no

meetup.com/nav-faglunsj

github.com/navikt

aksel.nav.no



tommy.jocumsen@nav.no

anders.skifte@nav.no

