

Systemdokumentasjon

Elektronisk valgadministrasjon - EVA Resultat



Innhold

- Introduksjon
 - Strategi for systemdokumentasjon
 - Effektiv systemdokumentasjonsforvaltning
 - Overordnet systembeskrivelse - EVA
 - EVA - Bruksområder
 - Kontekstdiagram
 - Verdikjedekontekst
- Overordnet beskrivelse - EVA Resultat
 - EVA Resultat
 - Bruksområde
 - Kontekstdiagram
 - Verdikjedekonteksts
 - Sikkerhetsstrategi
 - Kontekst for sikkerhetsstrategi
 - Sikkerhet gjennom lagdeling
 - Policies og prosedyrer
 - Policy
 - Prosedyre
 - Valgdirektoratets policies
 - Kommunikasjon over internett
 - Kryptering av kommunikasjon over internett
 - Sertifikater / PKI
 - Klientsertifikater
 - Funksjonelle moduler
 - Mandatberegning
 - Videreformidling av valgresultat
 - Valgresultatdomenet
- Arkitektur
 - Innledning
 - EVA Resultats utvikling
 - Applikasjonskonfigurasjon
 - Applikasjonsarkitektur
 - Sikring av infrastruktur
 - Tilgangsbegrensning
 - Redundans
 - Moduler i EVA Resultat
 - Mottak
 - Backend
 - Datamodell
 - Database
 - Frontend
 - Integrasjoner
 - Vedlegg

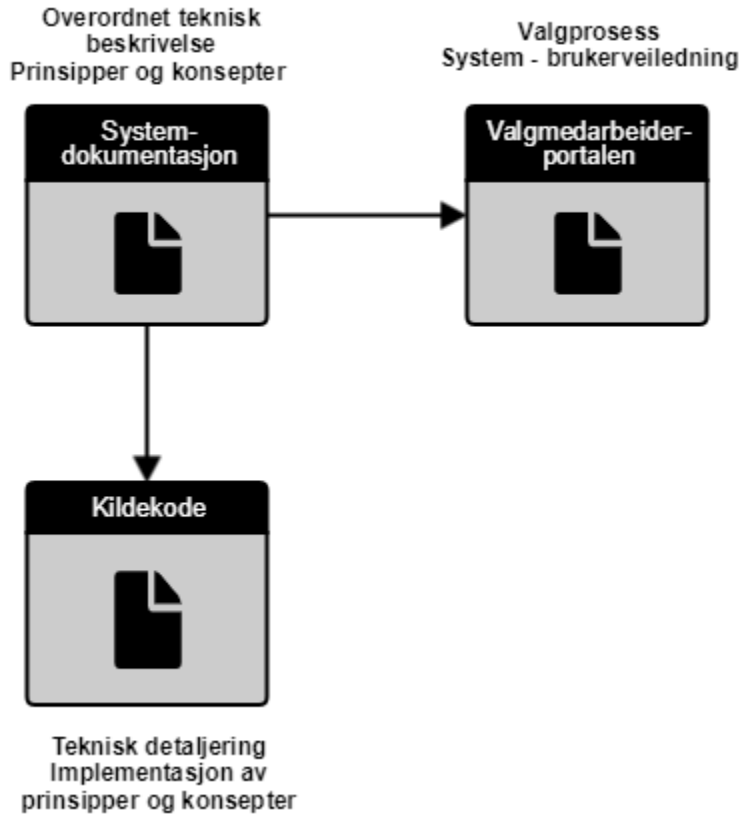
Introduksjon

Systemdokumentasjonen gir en overordnet beskrivelse av EVA -systemet fra et funksjonelt og teknisk perspektiv.

Dokumentet gir beskrivelser på et konseptuelt nivå, og er ikke en uttømmende teknisk beskrivelse av EVA. Konsepter og prinsipper gir i sin tur bakgrunn til kildekode og konfigurasjon som er hoveddokumentasjonen for systemet på et detaljert teknisk nivå.

Inngående beskrivelser av selve valgprosessen er gitt på valgmedarbeiderportalen, med brukerveiledning, prosess- og rutinebeskrivelser, samt skjematiske framstillinger av prosess.

Skisse som illustrerer hvordan dokumentasjonen er organisert:



Strategi for systemdokumentasjon

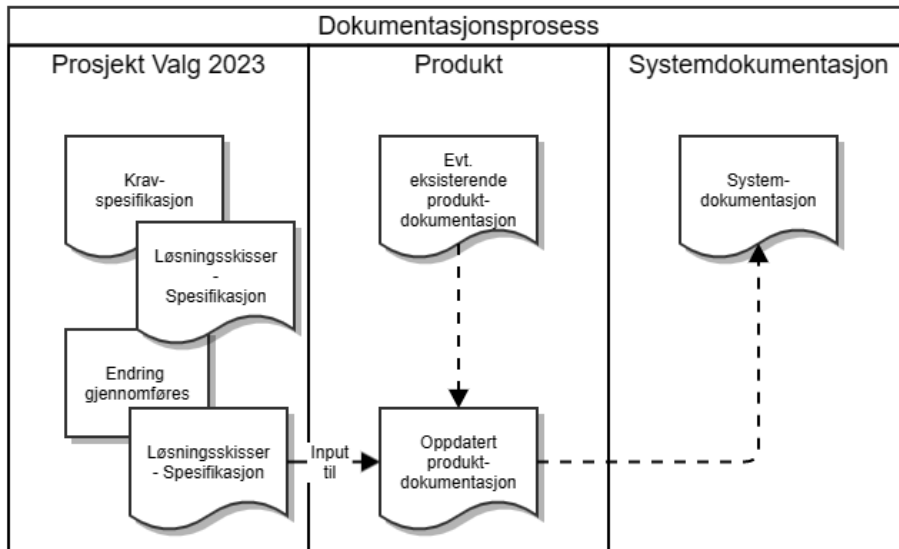
Valgdirektoratet bruker Atlassian Confluence (wiki) for å dokumentere IT-løsninger og systemer som utvikles og forvaltes av Valgdirektoratet. Systemdokumentasjon er derfor generert ut i fra det dette systemet.

Dokumentasjonen forankrer en enhetlig forståelse av hvordan systemet virker for å oppfylle de krav som stilles, dokumentasjonen brukes også som hjelpemiddel i utviklingsprosessen. Dokumentasjonen er ikke detaljert på et kodenært nivå, for å sikre en bestandig dokumentasjon som gir nødvendig overblikk og forståelse av prosesser og konsepter. Konkrete implementasjonsdetaljer må leses i kildekoden.

Effektiv systemdokumentasjonsforvaltning

Systemdokumentasjonen for Valgdirektoratets IT-systemer er bygget opp som et aggregat av de enkelte produktdokumentene, man unngår dermed duplisering og forvaltning av duplisert dokumentasjon og informasjon. Dette betyr også at informasjonen i systemdokumentasjonen er den samme informasjonen som brukes internt i Valgdirektoratet knyttet til systemforvaltningen, og således er *levende* og *benyttet* dokumentasjon.

Skissen illustrerer hvordan systemdokumentasjon bygges opp og vedlikeholdes



- Et krav stilles og spesifikasjon / løsningsskisser lages i Confluence.
- Endringen gjennomføres, systemet er nå endret ihht. spesifikasjon / løsningsskisse
- Eventuell eksisterende produktokumentasjon for det gitte området / den gitte funksjonen oppdateres, eller ny dokumentasjon legges til
- Systemdokumentasjonen oppdateres siden den er et aggregat av produktokumentasjonen

Overordnet systembeskrivelse - EVA

EVA - (Elektronisk valgadministrasjon) er Valgdirektoratets IT-støttesystemportefølje som kommuner og fylkeskommuner kan benytte seg av for å forenkle valg gjennomføringen. EVA er ikke et saksbehandlingssystem og kan ikke erstatte valgstyrene sitt ansvar for å påse at valghendelsene blir gjennomført etter gjeldende regelverk, men fungerer som et støtteverktøy for kommuner og fylkeskommuner i de ulike fasene av valg gjennomføring. EVA har funksjonalitet som støtter opp om gjennomføringen av

- Stortingsvalg
- Fylkestingsvalg
- Kommunestyrevalg
- Direkte valg til kommunedelsutvalg
- Sametingsvalg
- Lokalvalg til Longyearbyen

Systemporteføljen brukes også av Valgdirektoratet til oppfølging av valg gjennomføring, samt til rapportering til 3. parter.

EVA - Bruksområder

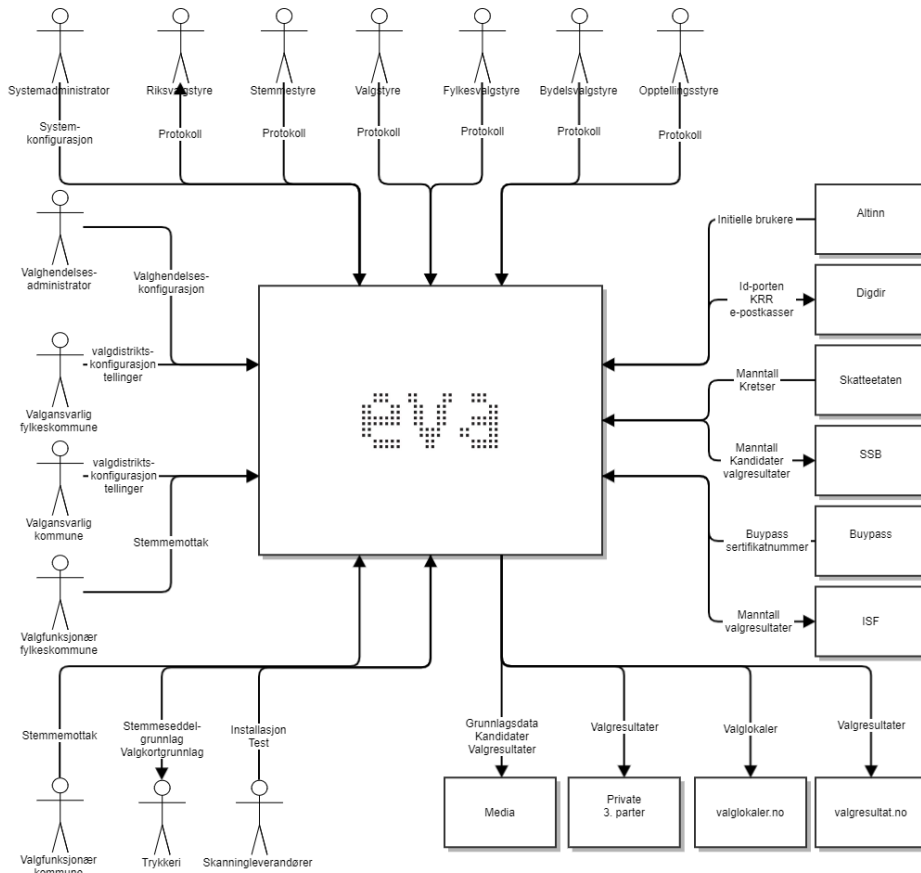
Valgavviklingen foregår over 4 faser og EVA porteføljen støtter alle disse fasene:

- Forberedelsesfasen
 - EVA Admin
- Stemmegivningsfasen
 - EVA Admin
- Opptellingsfasen
 - EVA Admin
 - EVA Skanning
 - EVA Resultat
- Valgoppgjøringsfasen
 - EVA Admin
 - EVA Resultat

Som det framgår av oversikten over brukes EVA Admin i alle faser, mens andre deler av porteføljen kun brukes for utvalgte faser.

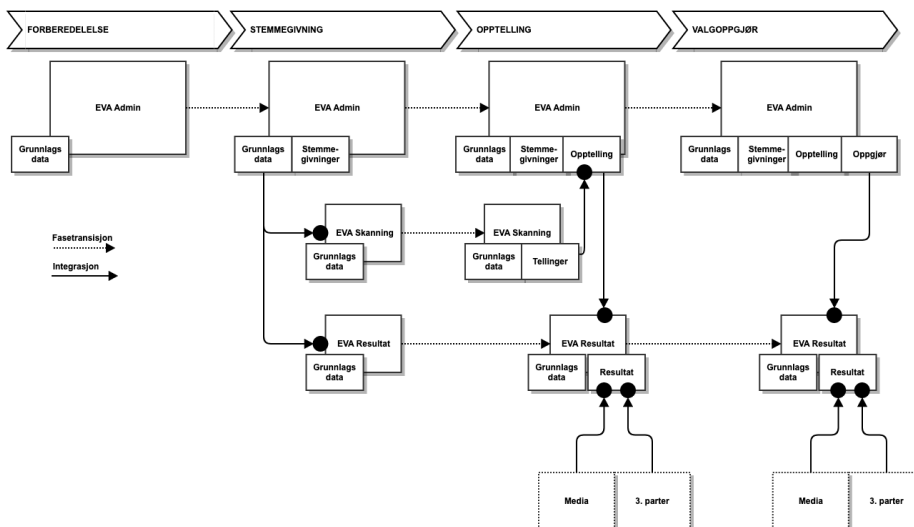
Kontekstdiagram

Diagrammet under beskriver de aktører og systemer / etater som har interaksjon og/eller interesser i/for EVA porteføljen.



Verdikjedekontekst

Diagrammet beskriver EVA systeminteraksjon i valg gjennomførings verdikjeden, kun hovedmoduler og aktører er inkludert i diagrammet.



Dette dokumentet omhandler kun EVA Resultat, de andre modulene beskrives i separate dokumenter spesifikke for den enkelte modul.

Overordnet beskrivelse - EVA Resultat

EVA Resultat

EVA Resultat er IT-systemet som brukes for å beregne og eksponere valgnattsprognose samt foreløpig og endelig valgresultat.

Når en opptelling er godkjent i EVA Admin, rapporterer kommunene og fylkeskommunene dette videre til EVA Resultat. I EVA Resultat ligger prognosemodellen som direktoratet tilbyr. Mediehus kan skrive en avtale med direktoratet for å benytte prognosemodellen.

Valgdirektoratet formidler også prognose og valgresultater via nettstedet valgresultat.no. Her presenteres resultatene i tall på en nøytral måte. Tallene hentes direkte fra EVA Resultat, og er de samme tallene som mediehusene får tilgang til. Valgdirektoratet publiserer sin prognose klokken 21.00 på valgresultat.no.

Kommunene og fylkeskommunene benytter kun EVA Admin og EVA Skanning. EVA Resultat er en intern applikasjon i Valgdirektoratet som kun sender tall mellom applikasjon og ut til andre interessenter.

Bruksområde

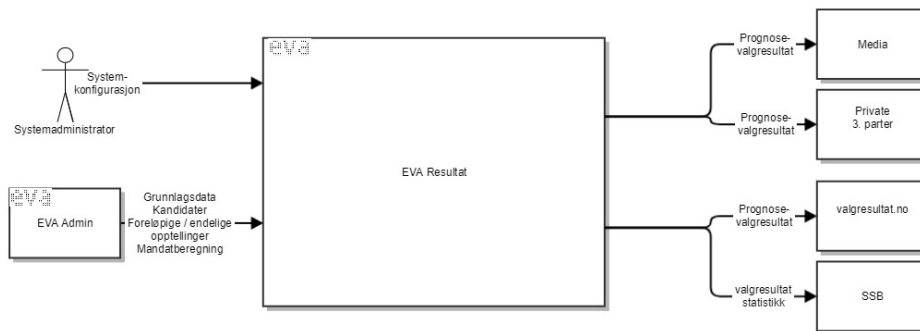
EVA Resultat brukes i opptellings- og valgoppgjørstfasen for beregning og eksponering og prognose samt valgresultater:

- Opptellingsfasen
 - Prognose beregnes fra valglokalene stenger til ca 01:00 - 02:00 påfølgende natt
 - Prognosen baserer seg på forhåndsstemmer og innkommende resultater underveis
 - Prognosen gir antatt partifordeling og mandatberegning
 - Når grunnlaget er komplett presenteres det faktiske valgresultatet og den antatte mandatfordelingen
- Valgoppgjørstfasen
 - Når valgoppgjør er gjennomført og endelig valgresultat foreligger med mandatkåringer presenteres det endelige valgresultatet med mandatfordeling

EVA Resultat eksponerer historiske valgresultater og statistikk og er således det eneste systemet i EVA porteføljen som også er aktivt utover valg gjennomføringen.

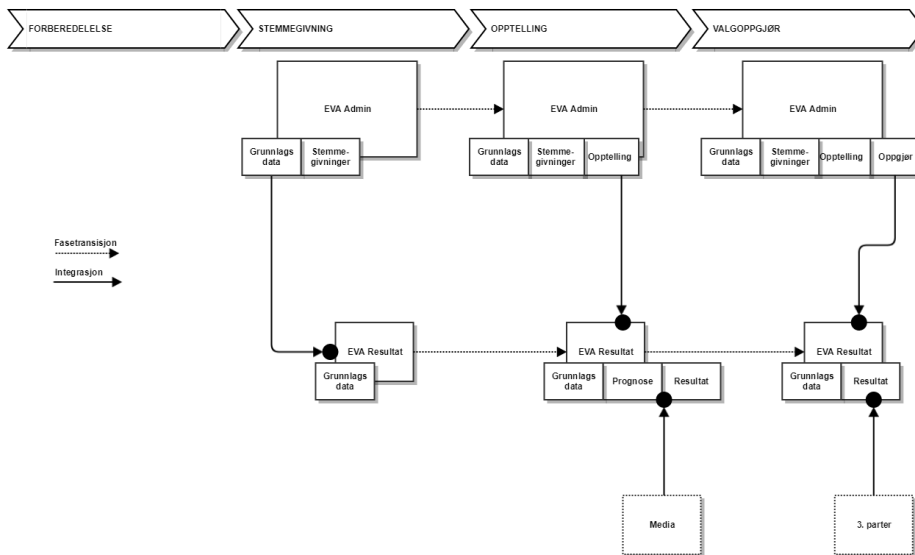
Kontekstdiagram

Kontekstdiagrammet beskriver de aktører og systemer som bruker EVA Resultat



Verdikjedekontekts

Diagrammet beskriver EVA systeminteraksjon i valg gjennomførings verdikjeden med fokus på EVA Resultat, kun sanntidsintegre 3. parter er inkludert i diagrammet



Sikkerhetsstrategi

Kontekst for sikkerhetsstrategi

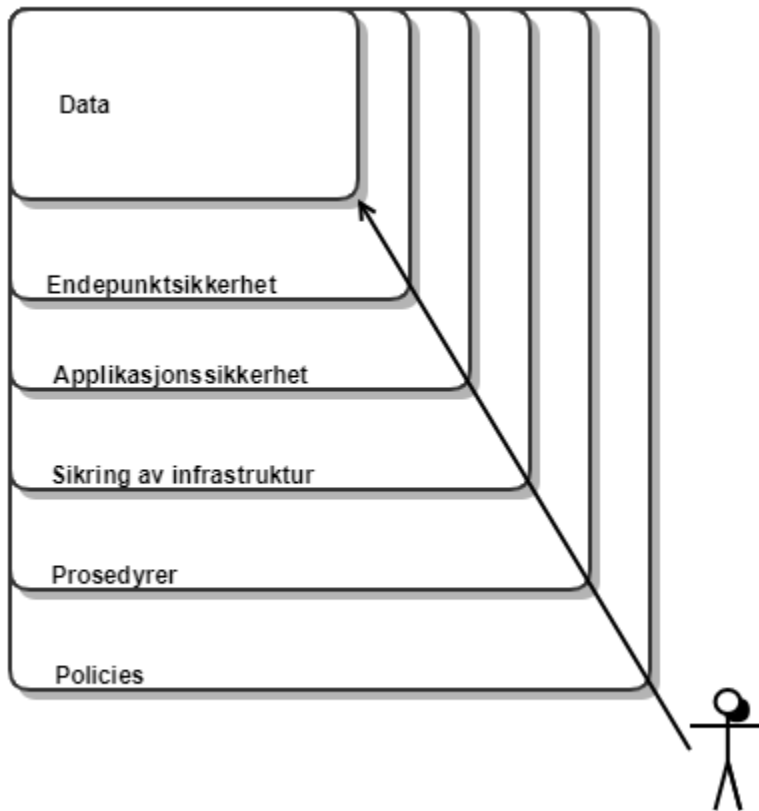
Sikkerhetsstrategien for EVA porteføljen inngår som en del av Valgdirektoratets policies for informasjonssikkerhet, sikkerhet og styring og kontroll.

Sikkerhet gjennom lagdeling

Applikasjonene sikres gjennom lagdeling som konseptuelt kan deles inn som følger:

- Policies
- Prosedyrer
- Sikring av infrastruktur
- Applikasjonssikkerhet
- Endepunktsikkerhet

Skisse som illustrerer lagdelingen som er beskrevet over



Policies og prosedyrer

Valgdirektoratet har utarbeidet policies og prosedyrer for sikkerhet.

Policy

En policy beskriver overordnede føringer og prinsipper for etablering, oppfølging og forbedring av et funksjonsområde.

Prosedyre

En prosedyre beskriver konkret framgangsmåter for å etterkomme en eller flere policies.

Valgdirektoratets policies

Valgdirektoratet har utarbeidet følgende policies med tilhørende prosedyrer:

- "Policy for informasjonssikkerhet"
- "Policy for sikkerhet"
- "Policy for styring og kontroll"

i tillegg til et sett med prosedyrer og retningslinjer, samt implementasjoner som oppfyller prinsipper og retningslinjer gitt i policies.

Kommunikasjon over internett

Kryptering av kommunikasjon over internett

Samtlige av Valgdirektoratets EVA produkter kommuniserer over internett, enten som avsender av informasjon eller som mottaker.

EVA produktene kommuniserer utelukkende på HTTPS-protokollen (Hypertekst Transfer Protocol Secure) der kommunikasjonen skjer over åpne nettverk (internett).

For en nærmere beskrivelse av HTTPS, se [Store norske leksikon](#) sin beskrivelse av protokollen, det gis ingen utdypende forklaring av protokollen i systemdokumentasjonen - annet enn at protokollen brukes.

Sertifikater / PKI

Valgdirektoratet har anskaffet nødvendige sertifikater fra *anerkjente sertifikatutstedere* (CAs) for å understøtte sikker kommunikasjon over internett og ivareta behovet for:

- Autentisering
- Kryptering (asymmetrisk)

For en nærmere beskrivelse av sertifikater/PKI se [Store norske leksikon](#) sin beskrivelse av PKI.

Klientsertifikater

Valgdirektoratet utsteder klientsertifikater som alle som skal kommunisere med EVA produktene må installere, dette er en ekstra sikkerhetsmekanisme for å begrense tilgangen til EVA produktene samt tilleggs-autentisere alle brukere av systemene. Dette påvirker ikke mekanismene beskrevet over.

Funksjonelle moduler

EVA Resultat har ikke funksjonelle moduler da det ikke skjer noen brukerinteraksjon mot EVA Resultat (utover Valgdirektoratets brukere for konfigurasjon og monitorering), EVA resultat tilbyr kun APIer for valgresultat og valgresultatrapporter som bl.a konsumeres av [valgresultat.no](#).

EVA Resultat utfører imidlertid to funksjoner:

- Mandatberegning
- Videreformidling av valgresultat

Mandatberegning

Modulen har ansvaret for å regne ut mandatfordeling i et valgdistrikt og lagre dette i MANDATBEREGNING tabellen. Mandatberegning gjøres i MainMandatBeregning.beregn()-metoden og stegene her er forholdsvis godt dokumentert (i kode og kommentarer). Kort oppsummert er prosessen slik - for et gitt område (land, fylke, kommune):

- 1) Finn stemmetall (summen av FHS og VTS) pr parti
- 2) Distriktsmandatfordeling for fylker/kommuner med stemmetallene funnet i 1)
- 3) Utjevningsmandatfordeling for landet med stemmetallene funnet i 1)

Stemmetallsberegning

Stemmetall kan som oftest beregnes svært enkelt - simpelthen ved å telle opp de stemmetall som finnes i området. Dette kalles da "opptaltberegning".

Distriktsmandatfordeling

Distriktsmandatene er enkle å fordele - de forholder seg til Sainte Lagües modifiserte metode. Det eneste man trenger å vite er hvor mange mandater som er til fordeling og stemmetallene for hvert parti. For å gjøre det litt mer informativt har man besluttet å vise mandatrang og kvotient for det siste mandatet partiet har vunnet, i tillegg til kvotient og mandatrang for det hypotetiske neste mandatet partiet ville vunnet dersom det var så mange mandater til fordeling. Disse tallene kan gi en indikasjon på hvilket parti som er nærmest å vinne et mandat til, og hvilket parti som er nærmest å tape et. Dette er bare tilleggsberegninger som ikke påvirker selve mandatfordelingen. I tillegg gjøres det ekstra beregning dersom man er i FY-valg, for da ønsker media at det også blir beregnet hvor mange distriktsmandater et parti ville fått dersom det var ST-valg. Slike ting kompliserer koden litt ekstra. Likevel, i det store og det hele så er denne utregningen uproblematisk.

Utjevningsmandatfordeling

Utjevningsmandatene er noe som kun deles ut i ST-valg på landsbasis (men av samme grunn i som for distriktsmandatene så må beregningen også gjøres i FY-valg på landsbasis). Utjevningsmandatberegningen er litt innfløkt, men forholdsvis godt kommentert. Det finnes også [gode forklaringer på nett](#). Igjen blir det gjort beregninger på "kampen om sistemandatet". Mye av koden går også med til å lagre informasjonen til databasen og oppdaterer data /geografi-modellen.

Videreformidling av valgresultat

Valgresultater tilbys i form av et API og et sett med rapporter.

valgresultat.no er Valgdirektoratets presentasjon av valgresultater, denne konsumerer API for valgresultater og presenterer disse.

Valgresultatdomenet

EVA Resultat håndterer et begrenset domene - valgresultat, domenet er således begrenset til 2 områder:

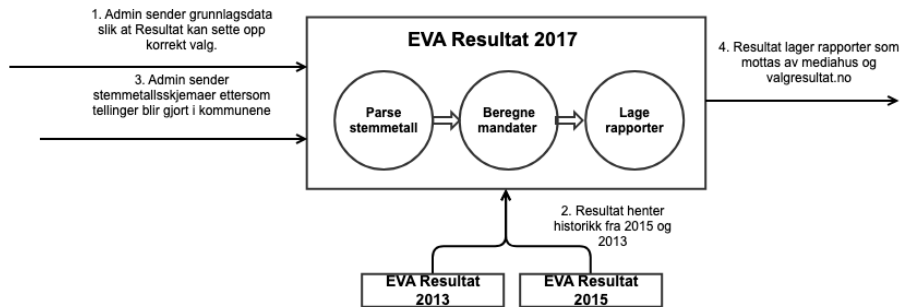
- Valgresultat
- Mandatberegning

Domenene er like som i EVA Admin, se systemdokumentasjon for EVA Admin for detaljer, For implementasjon se beskrivelse se "Datamodell" i kapitlet "Backend".

Arkitektur

Innledning

EVA Resultat har som navnet indikerer ansvaret for å produsere resultater. En måte å se systemet på er som en diger kalkulator som har stemmetall som input og produserer rapporter som output. Følgende diagram viser en forholdsvis abstrakt fremstilling av systemet:



Tallene viser rekkefølgen ting skjer. Først settes systemet opp ved hjelp av grunnlag fra EVA Admin og historikk fra tidligere valg (1 og 2). Deretter starter valget og vi får inn stemmetall og produserer rapporter (3 og 4). Diagrammet viser også hvordan systemet har en avhengighet til tidligere valg. Dette skiller systemet fra EVA Admin, som ikke har noe forhold til historikken. Historikken i Resultat-systemet er viktig av 4 grunner:

- Rapportene inneholder **endringen** i resultat fra siste og neste valg. For å finne denne endringen så må Resultat vite resultatene fra de 2 foregående valg.
- Historiske valgresultater er av offentlig interesse, og selv om SSB publiserer historiske valgresultater, så er det tenkelig at Valgdirektoratet kan tilby en langt bedre historisk tjeneste.

Grunnen til at dette nevnes allerede i den første introduksjonen er at Resultat ikke bare er et system for et bestemt valgår, men et system som skal fungere for alle valgår! Det betyr at vi må være ytterst varsom med historiske data og hele tiden ha kjørbare versjoner av alle valg.

EVA Resultats utvikling

EVA Resultat oppsto da oppgaven med å beregne prognoser ble flyttet fra SSB til Kommunal- og moderniseringsdepartementet og er så videre overtatt av Valgdirektoratet. Prognosemodellen ble i stor grad videreført fra SSBs system, men er ikke lenger i bruk. Nå er det en ren formidling av den fortløpende optellingene og det endelige valgresultatet.

Applikasjonskonfigurasjon

EVA Resultat konfigureres basert på grunnlagsdata fra EVA Admin, se beskrivelse "Statistiske data - Grunnlagsdata" i kapittel om database. I tillegg konfigureres EVA Resultat med historiske valg, se beskrivelse "Statistiske data - Historiske valg" i kapittel om database.

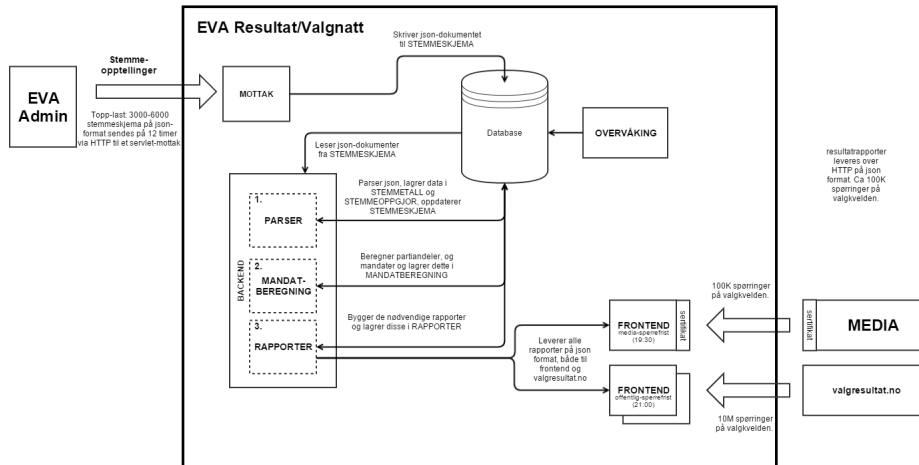
Applikasjonsarkitektur

Diagrammet viser de mest sentrale komponentene i EVA Resultat (bokser med heltrukken strek). En komponent i denne sammenheng en selvstendig prosess, enten som en web applikasjon som kjører i en applikasjonsserver, database eller en prosess som er utenfor Valgnatt. De mest sentrale tabellene er også nevnt i tekstene, slik at man ser den viktigste flyten hva angår database les/skriv. Av ting man kan merke seg er at systemflyten går fra venstre mot høyre, og at ingen ytre systemer kan påvirke Backend-prosessen direkte. Det samme prinsippet gjelder overfor EVA, som vi heller ikke kan påvirke, vi bare mottar data derfra. Pilenes retning angir hvem som er klient og server i et forhold, evt. at det både leses og skrives.

Med denne systemarkitekturen oppnår vi følgende ting:

- Skjermer den sentrale prosesseringsloopen i Backend fra direkte ytre påvirkninger
- Lagrer all nødvendig tilstand i databasen mellom hvert steg i prosesseringen
- Skalerbarhet i uthenting av rapporter
- Synkron utføring av stegene parsing og rapporter, da disse må ses i sammenheng

De sentrale komponentene i denne skissen vil bli beskrevet på Komponenter-siden.



Sikring av infrastruktur

EVA Resultat tilbyr API for valgresultater i to kanaler:

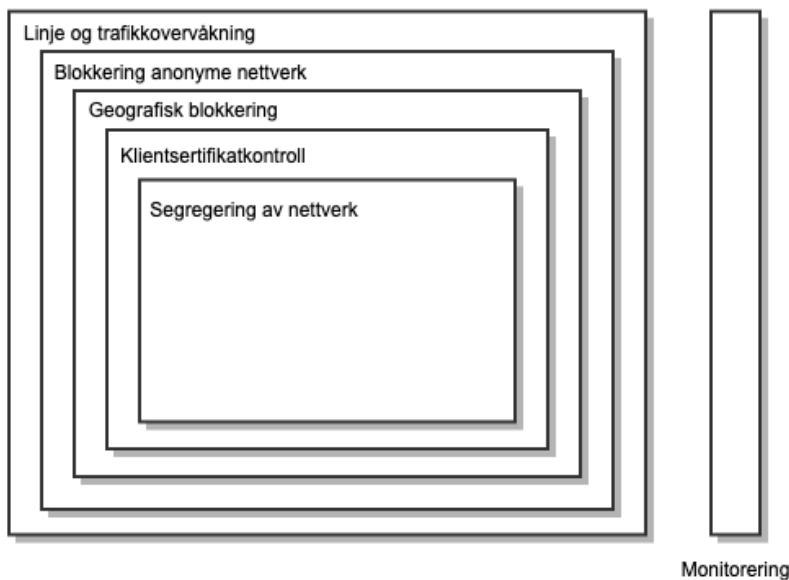
- Åpent - APIet er åpent for almenheten med gitt bruksvilkår
- Lukket - APIet tilbys til medieaktører med særskilt avtale og utvidet support

Kapitlet "tilgangsbegrensning" beskriver den lukkede kanalen til mediaktører.

Tilgangsbegrensning

Tilgang til Valgdirektoratets sentral driftede IT-applikasjoner er regulert gjennom forskjellige mekanismer for å snevre inn hvem som gis tilgang.

Skisse som forenklet illustrerer tilgangsbegrensning gjennom infrastruktur:



Tilgangskontroll

- Geografisk:
 - Tilgang fra anonyme nettverk forsøkes blokkert
 - Tilgang fra andre nasjoner er blokkert

- Personlig
 - Tilgang der man ikke har klientsertifikat utstedt fra valgdirektoratet er blokkert
- Logisk:
 - Tilgang til frontend servere er beskyttet av brannmur / i VLAN
 - Tilgang fra frontend til backend server er beskyttet av brannmur / i VLAN
 - Tilgang fra backend til database er beskyttet av brannmur i/VLAN
 - Frontend/backend er koblet i nodepar og isolert i VLAN

Monitorering

Alle ledd monitoreres med varslinger ved avvik

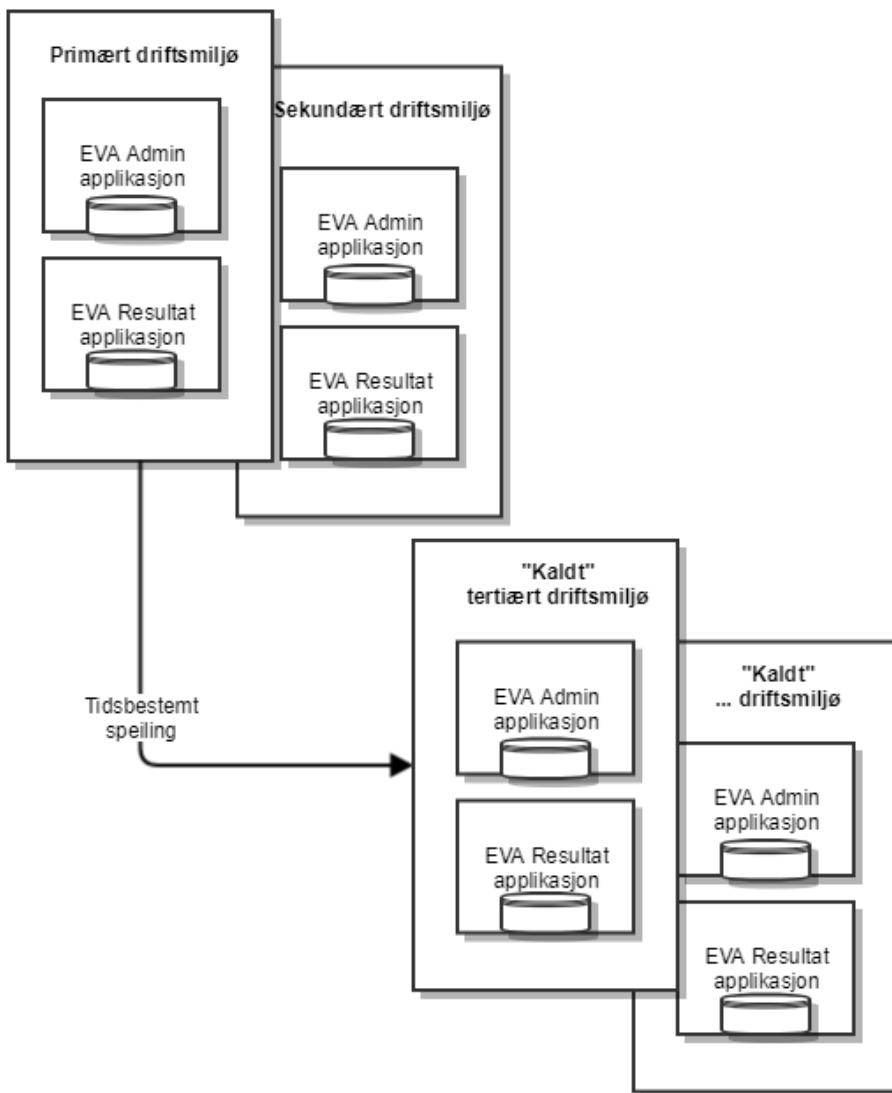
Merknad til "Tilgangskontroll":

EVA Resultat avviker fra denne beskrivelsen ved at frontend *ikke* har tilgang til backend - backend pusher, frontend forespør aldri. Se for øvrig beskrivelse i kapittelet

Redundans

Valgdirektoratets sentralt driftede IT-applikasjoner er sikret mot brudd i tjenestene med redundans for driftsmiljøene for disse applikasjonen.

Skissen illustrerer prinsippet for redundans av driftsmiljø



I en krise eller beredskapssituasjon hvor det primære driftsmiljøet ikke lenger fungerer vil man kunne ta i bruk det sekundære driftsmiljøet i løpet av svært kort tid. Miljøet speiler det primære driftsmiljøet i sanntid og således umiddelbart er klart til bruk uten datatap.

Om det sekundære driftsmiljøet ikke lenger fungerer vil man kunne ta i bruk det tertiære driftsmiljøet etter noe lenger tid. At miljøet er "kaldt" betyr at det ikke speiler produksjonsmiljøet i sanntid og ikke umiddelbart er klart til bruk.

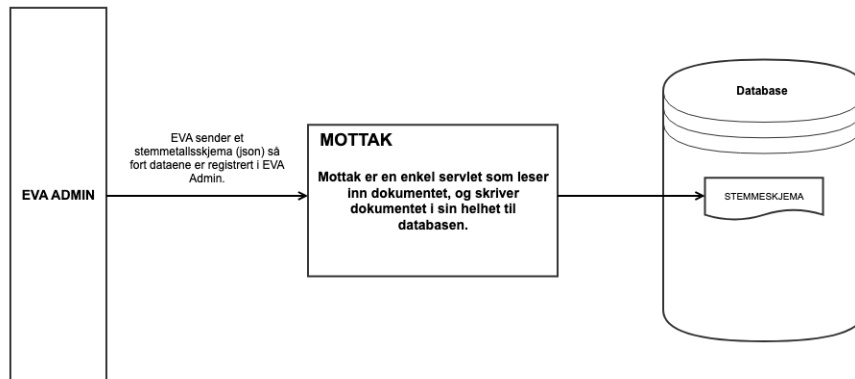
Moduler i EVA Resultat

Mottak

Mottak er en enkel servlet som får tilsendt stemmetalldokumentene fra EVA Admin og skriver dette til STEMMESSKJEMA tabellen i databasen. Et naturlig spørsmål er hvorvidt dette rettferdiggjør en egen web applikasjon og hvorfor akkurat en web applikasjon? I forrige versjon skrev EVA Admin rett til en tabell i databasen, så hvorfor ikke slik nå? Det korte svaret er fordi vi ønsker et robust mottak, med færrest mulige feilkilder som kan påvirke EVA Admin. Et litt lengre svar kan oppsummeres slik:

- EVA Admin kan sende en enkel HTTP-request istedet for å benytte en databasedriver (enklere).
- EVA Admin kan forvente høy opetid på mottaket, fordi mottak kun er avhengig av at applikasjonsserveren og databasen er oppe - ingen avhengigheter til annen prosessering i Valgnatt
- Mottaket virker ikke inn på annen prosessering i Valgnatt - eller visa versa.

Det ble vurdert som sikkerhetsmessig gunstig at EVA Admin sender til EVA Resultat, i stedet for at EVA Resultat henter fra EVA Admin - på den måten er EVA Admin fullstendig skjermet fra inngående trafikk.



Backend

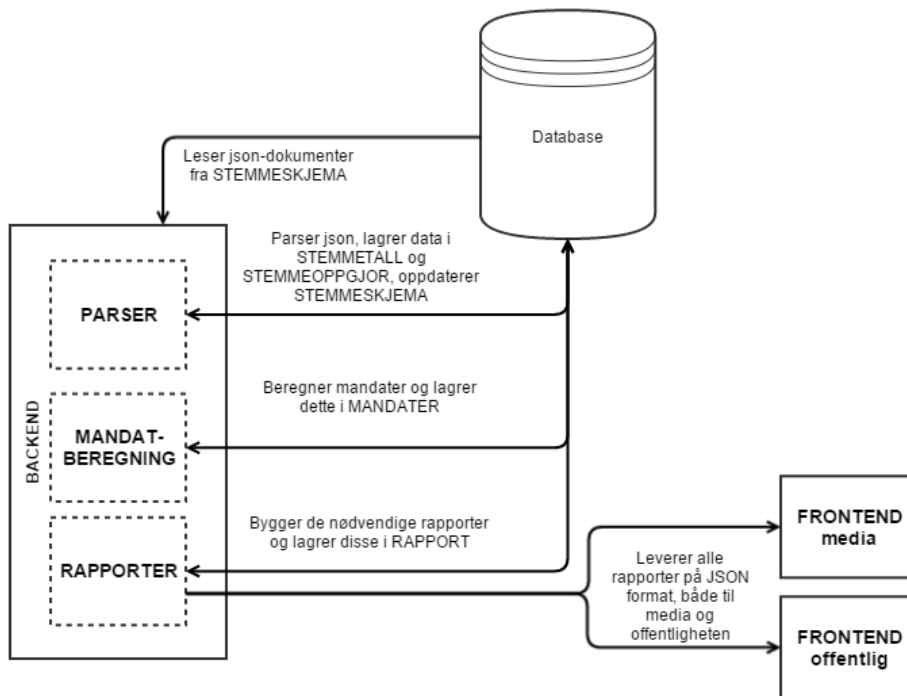
Backend er selve motoren i Valgnatt. Denne prosessen er skjermet fra utenforliggende systemer, og forholder seg hovedsaklig til databasen. Når rapportene er ferdigprodusert så sendes de ut over en HTTPS-kobling til evt. "lytteposter" (Frontend).

Et viktig grep i prosesseringen i Backend er synkron håndtering, dvs. at først henter man ut alle stemmetalldokumenter og parser alle data, deretter beregner man mandater og til slutt lages alle nødvendige rapporter. Tiden det tar for å regne ut mandater og lage rapporter er ikke veldig påvirket av antall stemmetalldokumenter som kommer inn, mandatberegningen tar omtrent like lang tid uansett, mens det kan bli **litt** flere rapporter å beregne dersom det kommer mange stemmetalldokumenter samtidig. Dette betyr at det bare er selve parsingen av dokumenter som påvirkes av antall dokumenter. Dette gjør igjen at uansett hvor mange dokumenter som blir prosessert, så vil alltid Backend-prosessen holde tritt med innkommende antall dokumenter, så lenge det kommer litt færre pr sekund enn det parseren kan parse pr sekund. Slik situasjonen er i dag forventer vi det kommer maks 10000 stemmetalldokumenter i løpet av 24 timer. Parse-prosessen kan håndtere minimum 1 million i samme periode, hvilket skulle gi en ytelse som er 100 ganger mer enn nødvendig.

Hovedloopen finner man i MainBackend klassen som gjør følgende steg:

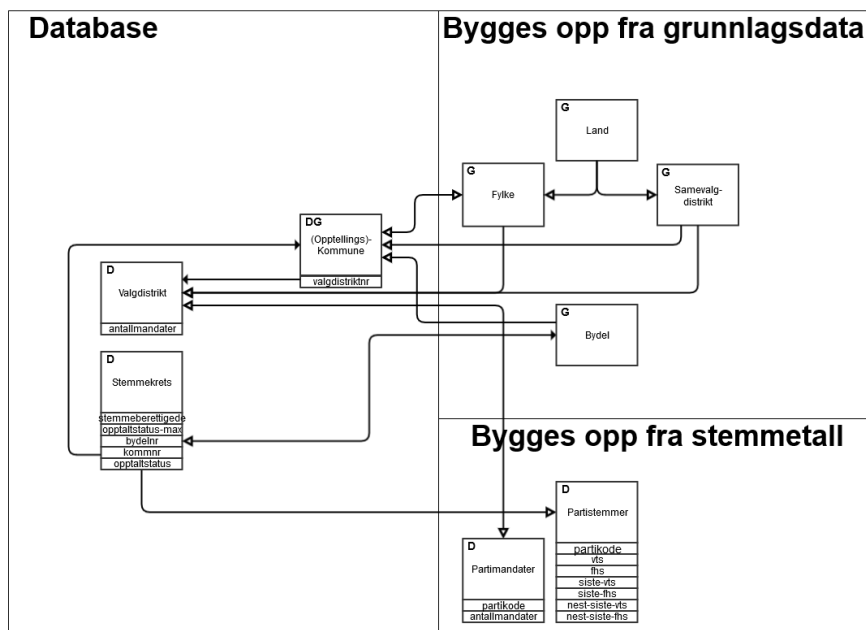
1. Kjør parsing for hvert dokument som ikke er parset (MainParser)
2. Kjør mandatberegning (MainMandatBeregning)
3. Kjør rapportering basert på hvilke dokumenter som har kommet inn (MainRapport)

Se undersider for mer informasjon om hvert steg i prosessen



Datamodell

Backend har en datamodell (ligger i database.geografi-pakken) som brukes i alle sub-komponenter. Under vises en tegning av datamodellen. Man kan aksessere datamodellen ved å hente ut et valg fra GrunnlagsdataCache. Denne datamodellen er kritisk viktig fordi den inneholder alle data som trengs for å bygge rapportene. Dvs. at datamodellen samler grunnlagsdata og stemmetallsdata. Prosessen for hvordan dette skjer er **forholdsvis** ukomplisert, men det kan nok ta en del tid å få full oversikt. I hovedsak så lastes grunnlagsdata inn først og så lastes stemmetallsdata på nytt hver gang det kommer et nytt stemmeskjema inn.



Hovedpoengene her er følgende:

Domene-konseptene er markert med D under. Med domene-konsept mener vi et element som er helt nødvendig fra et valgteknisk ståsted. I tillegg har vi geografiske strukturer (markert med G). Slike strukturer er nødvendige fordi vi ønsker rapporter for et slikt område. Av modellen nedenfor kommer det fram at vi laster opp domenestrukturane fra databasen, og bruker dette deretter til å bygge resten av objektmodellen. Det går også fram at vår geografiske modell har koblinger til domeneobjektene, og at kommune har en særlig sentral stilling. Det kan også nevnes at det finnes en superklasse som heter "Område" for følgende objekter: Land, Fylke, Samevalgdistrikt, Kommune, Bydel, Stemmekrets. Dette gjør at vi i veldig mange tilfeller (særlig når vi skal generere rapporter) kan behandle ethvert område på en generell måte.

Modellen over er komplett for hver valgtype og hvert valgår, slik at når EVA Resultat starter opp, så lastes alle tidligere grunnlag inn. Dette er i noen grad nødvendig, først og fremst for å kunne hente ut en del opplysninger om gamle valg.

Pilene som er hvite markerer en kobling som gjøres etter at databasen er lest inn gjennom Hibernate, dvs. at disse koblingene bygges opp i gjennom GrunnlagsDataService/Land-klassene.

Database

Det finnes i hovedsak 2 typer informasjon i databasen

- Statistiske data som settes ved oppsett av et valg, som igjen kan deles i 3 kategorier:
 - Grunnlagsdata
 - Historiske data
 - Annet
- Dynamiske data, data som kommer underveis i valget

Skillet mellom grunnlagsdata og historiske data er litt kunstig, fordi vi trenger noe historikk i grunnlaget for å sette opp et nytt valg. Dette skyldes at vi skal beregne endringer i resultatene fra foregående valg.

Statistiske data - Grunnlagsdata

Grunnlagsdata inneholder data som for det meste kommer fra EVA Admin før hvert valg. Valgdirektoratet konfigurerer EVA Resultat med historiske data basert på uttrekk av denne informasjonen fra EVA Admin.

- geografisk informasjon hentes fra følgende tabeller
 - stemmekretser
 - valgdistrikter
 - kommuner
 - bydeler
- mandat/kandidat-informasjon (valgdistrikter, kandidater (brukes bare i ST/SA-valg))
- partilister (partilister - hvilke partier finnes)

Statistiske data - Historiske data

Historiske data ligger lagret i csv-filer som følger med prosjektet år for år. Valgdirektoratet tilfører historiske data til EVA Resultat etter hvert gjennomførte valg. Dermed er systemet igjen klart for et nytt valg om 2 år. De fleste tabellene tar med valgår som en del av nøkkelen, og slik sett legges det bare til nye rader etter hvert som årene går. Historikken bygges opp ved å legge til nye data fra årets valg. Men det finnes 3 tabeller som er et unntak fra dette - dette er tabeller som manipulerer historikken til å passe inn med årets valg (tenk på hva som skjer om kommuner slår seg sammen, og lignende tilfeller). Grunnen til at vi trenger slike tabeller er vi ønsker lett tilgang til sammenlignbare data for forrige/siste og forrige-forrige/nextsiste valg i alle rapporter.

- "ren" historikk:
 - historiske stemmeresultat (historikk_stemmetall)
 - historiske antall stemmeberettigede (stemmekretser)
 - historiske antall mandater (historikk_mandater)
 - historisk geografi (stemmekretser, valgdistrikter, kommuner)
 - historisk valg-informasjon (kandidater)
- manipulert/modifisert historikk for de foregående 2 valg:
 - historisk stemmeresultat (historikk_stemmetall_mod)
 - historisk antall stemmeberettigede (historikk_stemmeberettigede_mod)
 - historisk antall mandater og fordeling av mandater (historikk_mandater_mod)

Statistiske data - Annet

- flyway-info (schema_version)

Dynamiske data

Dynamiske data er kan grovt sette mappes til de 3 delene av Backend-systemet: Mottak håndterer stemmeskjema og leser stemmetall. MandatBeregning gjør mandatberegninger og Rapport lager selvsagt rapporter.

- Stemmeskjema (stemmeskjema)
- Stemmetall (stemmetall, stemmeoppgjør)
- Mandatberegninger (mandater, mandatberegning_logg)
- Rapporter (rapporter)
- Tilstand (tilstand - her settes parametre for systemets tilstand som skal persisteres, slik som sperrefrist)

Frontend

Frontend er en server som gjør et par ting:

- Mottar alle rapporter som Backend lager
- Rapportene fra Backend kommer batchvis (i et JSON-dokument) - Frontend deler opp disse og cacher dem. I tillegg gjør Frontend litt prettifying på rapportene for at de skal se best mulig ut.

Integrasjoner

Se kapittel "Mottak"

Vedlegg

API-beskrivelse valgresultater

Brukervilkår er definert på valgresultat.no i "Om valgresultat.no".