

Sammenstilling af OS2-løsninger - en open source analyse

maj 2025

Indhold

Baggrund	3
Del 1 - Open source er licenser, praksis og metoder	6
1. Strukturerede releases og adgang til kildekoden	7
2. Åbne kodebiblioteker og uafhængighed af proprietære licenser	16
3. Anvendelig dokumentation og README-filer	18
4. Changelogs og dokumentation af fejlrettelser	20
Opsamling	21
 Del 2 - Gennemgang af de enkelte OS2-produkter	 24
OS2autopoces	24
OS2borgerPC	25
OS2compliance	27
OS2display	29
OS2faktor	31
OS2flytjord	33
OS2forms	34
OS2iot	36
OS2kitos	38
OS2korrespondance	40
OS2kravmotor	41
OS2mo	43
OS2nectar	45
OS2rollekatalog	46
OS2-SkadesØkonomi	48
OS2skoledata	51
OS2sofd	53
OS2sync	55
OS2udoglær	57
OS2valghalla	59

Baggrund

Vi har udarbejdet denne analyse, fordi der er en helt åbenlys uoverensstemmelse mellem OS2's idealer og så den praksis, bestyrelsen tillader for en stor del af OS2-løsningerne.

Vi har ad flere omgange gjort OS2's bestyrelse opmærksom på problemet med closed source-løsninger, der blåstemples som OS2-produkter, senest i notatet "Oplæg til dagsordenspunkt til OS2-bestyrelsesmøde 24. juni 2024"¹, som vi sendte til bestyrelsen primo juni 2024. Men vi får ingen feedback fra OS2, der undlader at reagere på vores henvendelser.

OS2 startede ud med de rigtige intentioner i 2012, men OS2 er i dag en indkøbsforening, der har glemt sit formål. Det meste af det, OS2 står for i dag, har ikke meget med open source at gøre.

Det er en reel bekymring, at open source-miljøet i Danmark bliver ødelagt af OS2. Foreningens accept af løsninger, der intet har med open source at gøre, ødelægger vækstlaget og økosystemet. Og det ødelægger innovationskraften blandt de startups og mindre virksomheder, der vælger open source som deres forretningsmodel. For open source koster tid, ressourcer og dermed også penge at udvikle. Alt dette underminerer OS2 med sin accept af løsninger, der reelt er closed source.

Hvorfor er det kritisk, at OS2 blåstempler løsninger, der ikke er open source? Det er det af to overordnede grunde:

1) Økosystemet bliver skadet og troværdigheden forsvinder

Det er konkurrenceforvridende for de virksomheder, der vitterligt leverer open source. Kvalitet koster penge, og det samme gør open source af høj kvalitet. Når OS2 sætter sit stempel på løsninger, der ikke er open source, og som leverandøren ikke behøver anstrenge sig for at levere, så skævvrider det markedet.

Frivillige og professionelle open source-udviklere bliver demotiverede, når de oplever, at deres kvalitetsarbejde sidestilles med lukkede løsninger.

Det skader i sidste ende vækstlaget og økosystemet for de ægte open source-virksomheder og startups, der investerer tid og ressourcer i at udvikle åbne løsninger

Vi oplever det allerede i dag. Vi har svært ved at konkurrere med lukkede løsninger, der 'sælges' som open source af OS2.

OS2 medvirker dermed til "open source washing", når foreningen bruger sit open source-brand til at fremme produkter, der ikke lever op til standarderne.

¹https://www.magenta.dk/wp-content/uploads/2025/02/1-Indsigelse_-OS2-bestyrelsesmoede-24.-Juni-2024.pdf

Dermed mister OS2 sin troværdighed i open source-fællesskabet - i Danmark og internationalt. Medlemmerne, kommuner såvel som virksomheder, oplever det som et svigt af de principper, OS2 er grundlagt på. Og det samme gør udviklere og andre open source-organisationer i ind- og udland.

2) Falsk tryghed og øget risiko

Anvenderne af de OS2-løsninger, der ikke er open source, tror med rette, de får fordelene ved open source - blandt andet sikkerhed gennem transparens og ingen vendor lock-in. Men i virkeligheden får de lukkede systemer. Det øger risikoen for sikkerhedshændelser.

De lukkede løsninger kan have proprietære licenser - noget vi dokumenterer i denne analyse - og det kan medføre omkostninger og begrænsninger i anvendelsen af løsningerne.

Desuden er det ganske enkelt forbundet med stor risiko, fordi OS2's løsninger potentielt er usikre. OS2 argumenterer helt korrekt for, at open source er mere sikkert end closed source. Det er samme pointe som bl.a. Europa-Kommissionen og mange efterretningstjenester fremfører. Pointen blev gentaget af OS2's repræsentant på en høring, som Folketingets Digitaliseringsudvalg afholdt 8. maj 2025.²

Derfor er det kritisk, at OS2 mod bedre vidende blåstempler løsninger, der intet har med open source at gøre.

Vi, der arbejder professionelt med open source, frygter et scenarie, hvor et OS2-produkt, der ikke er open source, bliver hacket. Eftersom ingen ved, hvad der ligger under motorhjelm, og kun leverandøren kan rette eller fejlfinde, så er de kommuner, der benytter løsningen, prisgivet og uden mulighed for at beskytte forretningen for den it-løsning, de troede var sikker. Fordi løsningen havde fået open source-stemplet.

Det kommer til at ramme open source generelt - også de rigtige open source-løsninger, som Magenta og en række andre virksomheder i Danmark og udlandet leverer. Dette scenarie kan vi ikke sidde overhørig. Vi har som nævnt forsøgt at trænge igennem med budskabet på de interne linjer igennem flere år, men eftersom vi er blevet ignoreret, har vi valgt at publicere resultaterne i denne rapport.

I Magenta stiller vi krav til os selv og til de løsninger, vi udvikler i samarbejde med OS2. De skal være open source. Både fordi det er open source, Magenta er grundlagt på, og fordi open source er en af de to grundpiller i OS2-samarbejdet.

² <https://www.ft.dk/da/udvalg/udvalgene/diu/tv?s=20241&m=tv.9670&as=1#player>

Det skader OS2 internt

Når OS2 mod bedre vidende accepterer closed source-løsninger og giver dem OS2-stemplet, vil det på sigt føre til interne uenigheder og splittelse blandt medlemmerne. Mange af OS2's medlemmer er engagerede i open source-idealet, og de vil ikke have forståelse for svigtet af open source-principperne.

Fokus kommer til at flytte sig fra samarbejde og vidensdeling omkring åbne løsninger til et traditionelt leverandør-kunde-forhold omkring lukkede produkter. Dermed risikerer OS2 at miste sin klare identitet som en open source-forening. Reelt er OS2 i dag en almindelig it-indkøbsforening.

Accepten af closed source-løsninger i en open source-forening har vidtrækkende negative konsekvenser for open source-fællesskabet og OS2's troværdighed. Det underminerer principperne om åbenhed og transparens, som open source står for.

Notatets opbygning

Første del af notatet gennemgår de overordnede krav til open source-løsninger, og vi præsenterer vores konklusioner her.

Anden del indeholder en mere detaljeret gennemgang af de enkelte løsninger.

Vi har naturligvis gravet dybt i OS2's GitHub repositorier for de enkelte produkter, men for at yde alle løsninger maksimal retfærdighed har vi også søgt bredt efter, om der mon skulle ligge kildekode, information om releases, container images eller andet på andre platforme.

Når vi har fundet oplysninger om de enkelte løsninger på andre platforme end GitHub, er det kommet den enkelte løsning til gode i analysen. Vi har således ikke anlagt en rigid fortolkning af open source i den forstand, at kildekoden skal være placeret i OS2's repositorier. Havde vi gjort det, ville endnu flere af OS2's løsninger være stemplet ikke-open source på endnu flere områder.

Hvis man måtte opdage unøjagtigheder, mangler eller fejl, så kontakt Magenta på info@magenta.dk. Så medtages rettelserne i en revideret version.

København, maj 2025

Del 1 - Open source er licenser, praksis og metoder

Open source er mere end licenser og software, hvor kildekoden er frit tilgængelig, så alle kan bruge, ændre og dele den. Det er naturligvis en betingelse, at kildekoden er frigivet på en anerkendt licens³, men det er langt fra tilstrækkeligt til at gøre kildekode til reel open source.

Open source er en praksis, der bidrager til, at idealerne om åbenhed og frihed kan blive til virkelighed. Kun gennem klare og strukturerede arbejdsmetoder kan open source realiseres. Det er metoderne, der sikrer, at softwaren bliver anvendelig og i praksis lever op til de idealer, open source bygger på. Metoder og praksis gør udviklingsprocessen gennemskuelig for tredjepart og sandsynliggør, at koden er forståelig og genanvendelig.

Lad os se på de arbejdsgange, der er væsentlige for, at kode forvandles til open source.

Adgang til kildekoden er afgørende. Og små, hyppige commits og veldokumenterede releases åbner op for, at tredjepart kan gennemskue koden og bidrage til den. Uden denne mulighed forbliver softwaren reelt lukket.

Åbne kodebiblioteker. Når der gøres brug af tredjeparts kodebiblioteker, er det afgørende, at de er åbne. Det sikrer gennemsigtighed og fri adgang til at modificere og vedligeholde koden. Afhængighed af lukkede kodebiblioteker skaber begrænsninger og afhængigheder, der strider mod open source-principperne om åbenhed. Det kan betyde, at man ikke frit kan bruge og videregive løsningen, fordi brugerne pålægges at skaffe sig adgang og måske købe licenser til proprietære biblioteker. Derfor gør afhængighed af lukkede kodebiblioteker koden til closed source.

Anvendelig dokumentation er en uundværlig del af open source. Dokumentationen gør koden tilgængelig, forståelig og brugbar for både udviklere og slutbrugere. **README-filer** er en væsentlig del af dokumentationen og fungerer som den første indgang til et produkt. Her får man en præcis beskrivelse af produktets formål, installationskrav, og hvordan man kommer i gang med udvikling eller brug.

Changelogs og dokumentation af fejlrettelser medvirker til kodekvaliteten og -stabiliteten af open source-løsningen. Blotlægning af fejl og fejlrettelser giver transparens.

Opsummeret er rigtig open source, udover en godkendt licens:

1. Strukturerede releases og adgang til kildekoden
2. Åbne kode-biblioteker
3. God dokumentation, herunder særligt README-filer
4. Changelogs, herunder dokumentation af fejlrettelser

³ <https://opensource.org/>

Open source er altså langt mere end en softwarelicens. Open source er en praksis og en række strukturerede metoder, der sikrer kodekvalitet, samarbejde og åbenhed.

Nedenfor trækker vi nogle 'highlights' fra analysen i del 2 frem i en opsamling, der belyser nogle overordnede konklusioner på de fire metodiske punkter.

1. Strukturerede releases og adgang til kildekoden

Det siger sig selv, at et open source-projekt skal offentliggøre sin kildekode, men det er ikke nok, at koden offentliggøres. Et kig i OS2-produkternes repositorier viser, at mange af dem ikke afspejler den faktiske udviklingshistorik og versionering af produktet.

To af løsningerne lever ikke op til det helt fundamentale krav om at offentliggøre kildekoden. Det drejer sig om OS2flytjord, som har et offentligt GitHub-repo, men det indeholder ingen kildekode. Repoet består udelukkende af en skabelon, der viser, hvilke filer et projekt bør indeholde, og indeholder derfor ingen specifik information om selve løsningen.

Den anden løsning uden kildekode er OS2nectar. Det har ikke været muligt at finde et repo for denne løsning, hverken på GitHub eller andre platforme, hvor kildekoden kunne være offentliggjort. Ved at følge henvisningen til kildekoden på OS2's egen hjemmeside ledes man i stedet til GitHub-repoet for OS2borgerPC-løsningen.

Da kildekoden for disse to projekter ikke er offentligt tilgængelig, er det dermed ikke muligt for tredjepart at bruge, undersøge, ændre eller videregive løsningen. Derfor kan disse projekter heller ikke betragtes som open source.

Andre projekter, herunder OS2rollekatalog, OS2skoledata og OS2sync, udviser tydelige tegn på, at udviklingen foregår privat, og at kildekoden først skubbes til GitHub i form af større kodedumps. Det hæmmer produkternes transparens, da man ikke kan se de individuelle commits for et bugfix eller en feature.

Når kode offentliggøres på denne måde - i store 'bunker' - mister man muligheden for at have et overblik over specifikke ændringer i kodebasen. Når der er få commits, som dækker store ændringer i kodebasen, bliver det vanskeligt - eller reelt umuligt - at følge udviklingshistorikken og få et indblik i de enkelte ændringer, som er tilføjet til koden.

Derudover er der ofte manglende sammenhæng mellem releases og de versioner, der er tilgængelige på fx Docker Hub eller produkternes hjemmesider. Eksempelvis findes der på Docker Hub en release 4.6.4 af OS2sync, mens der i GitHub-repoet kun kan findes en release 4.4.1.

Det er en vigtig del af principperne om open source og sikkerhed, at brugerne har adgang til koden for den version, som faktisk er i drift.

En klar release-struktur gør det lettere for tredjepart at sætte sig ind i ændringer foretaget mellem releases, og det hjælper til forståelsen af kildekoden. Nogle få løsninger i analysen - OS2borgerPC, OS2forms, OS2kitos og OS2mo - præsenterer deres releases i form af release-tags, og skaber derfor et overskueligt overblik af releases samt commits lavet op til et release.

Overensstemmelse mellem releases og commits er afgørende årsager af flere årsager, der alle bidrager til transparens, tillid, sikkerhed og samarbejde, som er kerneprincipper i open source.

Det handler om gennemsigtighed i udviklingshistorikken. Når releases og commits stemmer overens, giver det et klart og troværdigt billede af, hvordan softwaren er udviklet over tid. Hver release bør afspejle en specifik samling af commits, der tilsammen udgør den pågældende version. Uoverensstemmelser skaber forvirring og gør det svært at spore, hvilke ændringer der er inkluderet i en given release.

Det handler om verifikation og reproducerbarhed. Overensstemmelse gør det muligt for tredjepart at verificere, at den kode, der er erklæret som en bestemt release, faktisk er den kode, der ligger til grund for de tilsvarende commits i kilderepositoriet. Dette er afgørende for reproducerbarhed af builds og for at sikre, at der ikke er blevet tilføjet uannonceret kode.

Det handler om sikkerhedsaudit og -analyse. Når releases og commits er synkroniserede, bliver det lettere at foretage sikkerhedsaudits. Tredjepart kan undersøge de specifikke kodeændringer, der er introduceret i hver release, og relatere dem til eventuelle sikkerhedsrettelser eller nye sårbarheder. Uoverensstemmelser gør dette arbejde betydeligt sværere og mere usikkert.

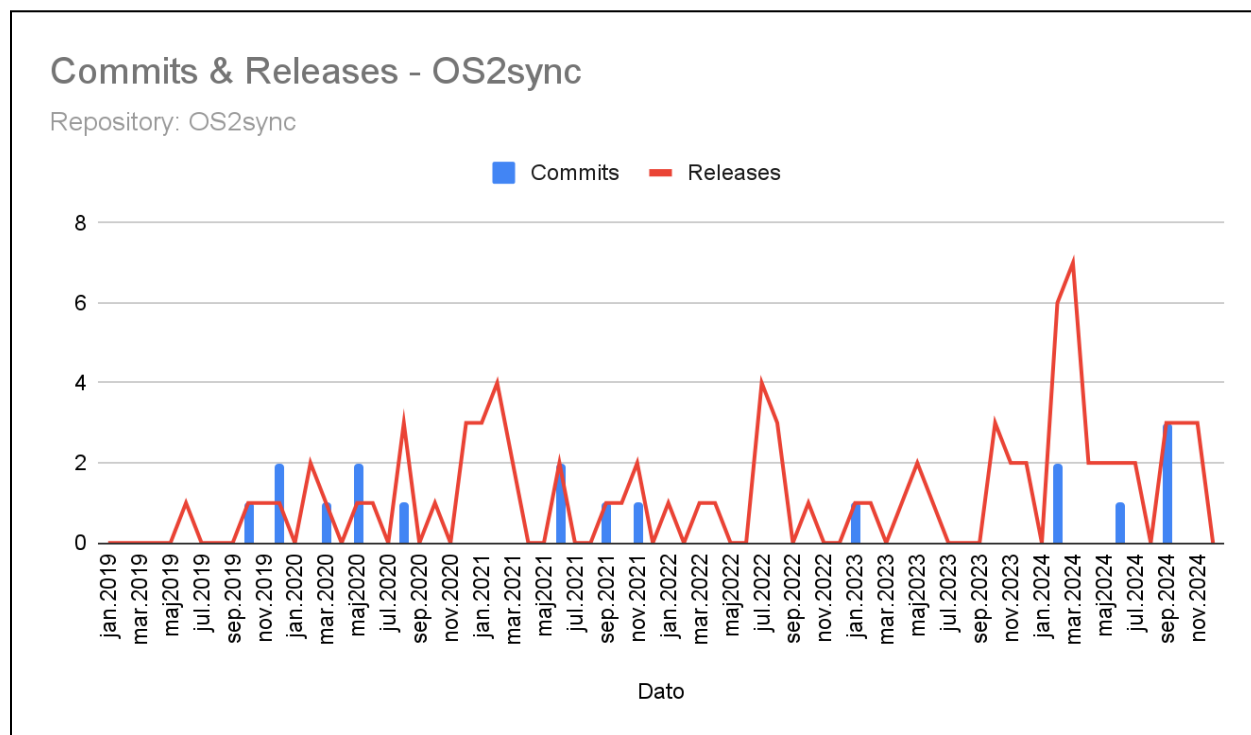
Langt de fleste OS2-løsninger viser imidlertid ikke releases i form af release-tags eller på anden vis. Tredjepart er i stedet nødsaget til at gennemgå de enkelte commits for at få indblik i løsningens release-struktur.

En klar release-struktur er ikke et formelt krav for, at noget kan kaldes open source; men besværliggøres muligheden for at sætte sig ind i løsningens kildekode, bliver det også sværere at bruge, undersøge og ændre løsningen. Desuden er changelogs for de fleste produkter enten mangelfulde, fragmenterede eller helt fraværende. OS2sync viser for eksempel store spring i versionsnumre mellem commits, hvilket tyder på, at releases sker uden tilhørende commits. Et eksempel er springet fra version 2.2.0 til 2.5.0.

Et eksempel på en løsning med en transparent udviklingsproces er OS2mo. Projektet har siden opstarten i 2017 opereret med små, hyppige commits og releases. Og hver gang med mindre og overskuelige kodeændringer per commit. Der findes 967 release-tags i os2mo-repoet på GitHub, hvilket giver en klar release-historik. En gennemgang af OS2mo på Docker Hub viser,

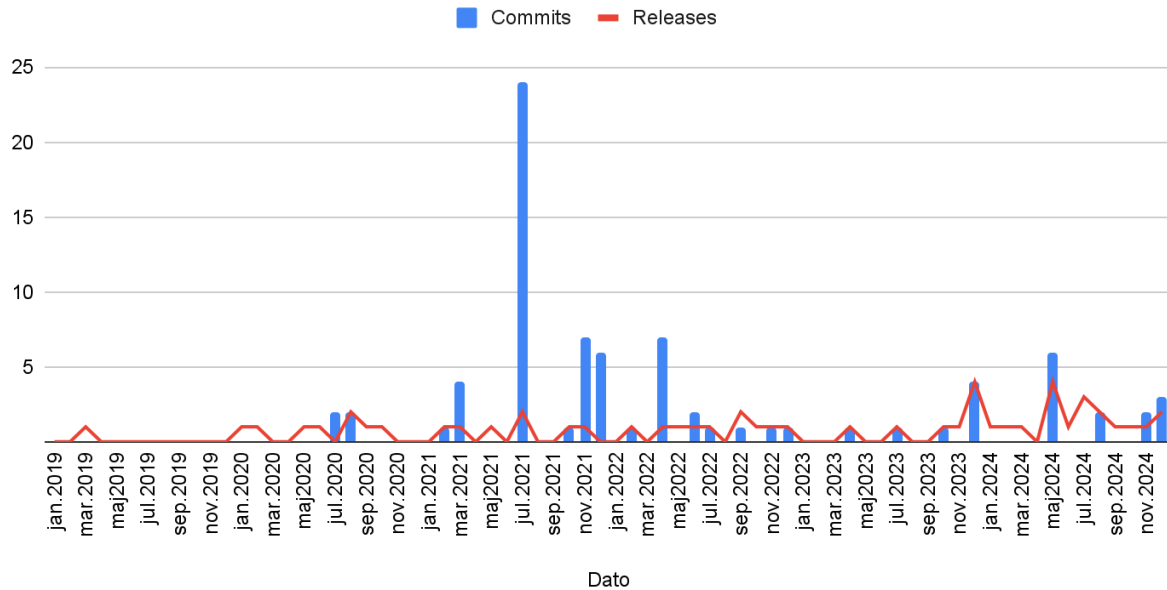
at releases er i overensstemmelse med commits på GitHub. Dermed er det også sikret for tredjepart, at man har et overblik, og at man ikke på noget tidspunkt er afhængig af lukket kode.

Nedenfor er der udarbejdet diagrammer, der krydser commits og releases på produkterne OS2sync, OS2rollekatalog og OS2mo i perioden januar 2019-december 2024, altså samlet 6 år. Søjlerne dækker over et kvartal. Formålet er at sætte fokus på produkter, der laver kodedumps og produkter, der laver kontinuerlige commits og releases.



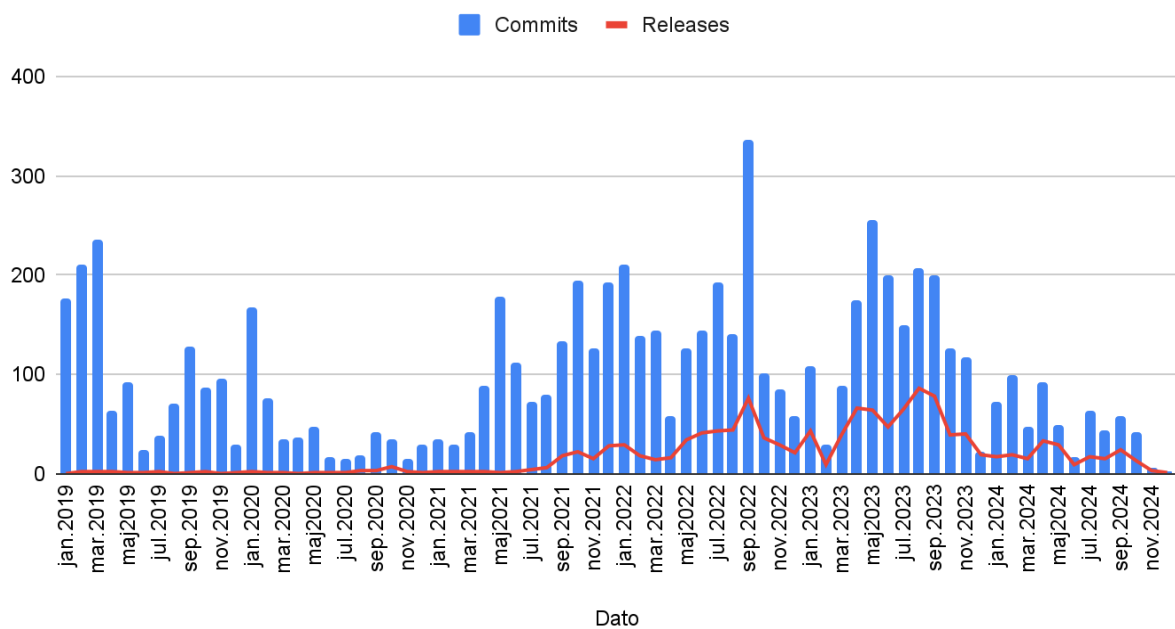
Commits & Releases - OS2rollekatalog

Repository: OS2rollekatalog



Commits & Releases - OS2mo

Repository: os2mo



Bemærk y-aksens tal. OS2sync kommer kun i ét kvartal over 4 commits eller releases. OS2rollekatalog kommer kun over 7 commits i ét kvartal. OS2mo ligger i gennemsnit på 100 commits eller releases hvert kvartal.

Hvis commits og releases ikke sker kontinuerligt, betyder det i praksis, at der ikke er adgang til kildekoden. Det betyder, at udviklingen sker i lukkethed, og at kildekoden kun i begrænset omfang er transparent og tilgængelig. I visse situationer kan man være tvunget til at bruge lukket software, hvis man anvender den nyeste version af OS2sync, da det ikke er sikkert, at kildekoden til denne version på Docker Hub er offentligt tilgængelig på GitHub. Vi ser eksempler på dette i diagrammerne, hvoraf det fremgår, at OS2sync har 67 tilfælde, hvor releases på Docker Hub ikke har tilsvarende commits på GitHub.

På OS2syncs Docker Hub findes der fem versioner af løsningen, som er nyere end den version, der er tilgængelig på GitHub. Det betyder, at man skal fem versioner bagud for at finde og anvende en version, der ikke er bag lukket kode.

På OS2mo-løsningen, derimod, er der ikke foretaget et eneste kodedump. Releases på Docker Hub har tilsvarende commits på GitHub, samtidig med at nyeste version af Docker images stemmer overens med versionen af kildekoden på GitHub. På den måde sikres der transparens på OS2mo-projektet over for tredjepart, der måtte være interesseret i at arbejde videre med løsningen.

Hyppige og veldokumenterede releases er således en arbejdsgang, der spiller en vigtig rolle for, om et produkt reelt er open source. Det sikrer gennemsigtighed, gør det lettere for andre at følge projektets udvikling og øger muligheden for fejlfinding. Analysen af OS2-produkterne viser, at kun få af produkterne følger dette princip.

Et eksempel på manglende sammenhæng i denne forbindelse er OS2sync, hvis repository kun har haft 18 commits siden oktober 2019. De 18 commits repræsenterer alle releases med større ændringer til kodebasen. Det er derfor ikke muligt at gennemskue den reelle release-historik ved kun at kigge på produktets GitHub.

OS2sync er kun én af mange OS2-løsninger, hvor det har været nødvendigt at finde yderligere information om releases ved at søge på andre platforme end GitHub, som for eksempel Docker Hub eller produktets hjemmeside.

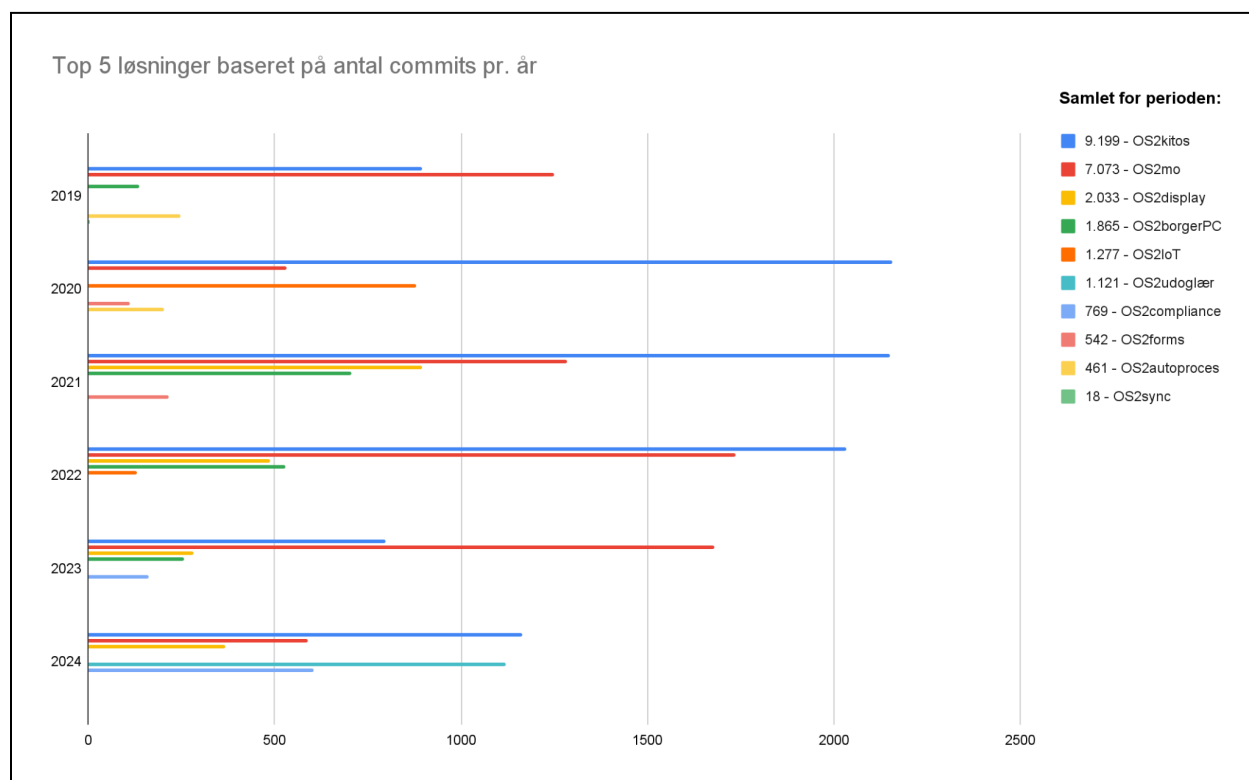
Modsat er OS2mo et eksempel på et produkt med en aktiv og transparent udviklingshistorik. OS2mo-repoet har som tidligere nævnt 967 releases, der alle er identificeret via release-tags, hvilket er med til at sikre en gennemsigtig release-historik. Med næsten 10.000 commits siden 2017 udviser OS2mo en udviklingsstruktur, hvor commits sker ofte og repræsenterer mindre ændringer til kodebasen frem for store, sjældne releases, sådan som det er tilfældet med de fleste OS2-løsninger.

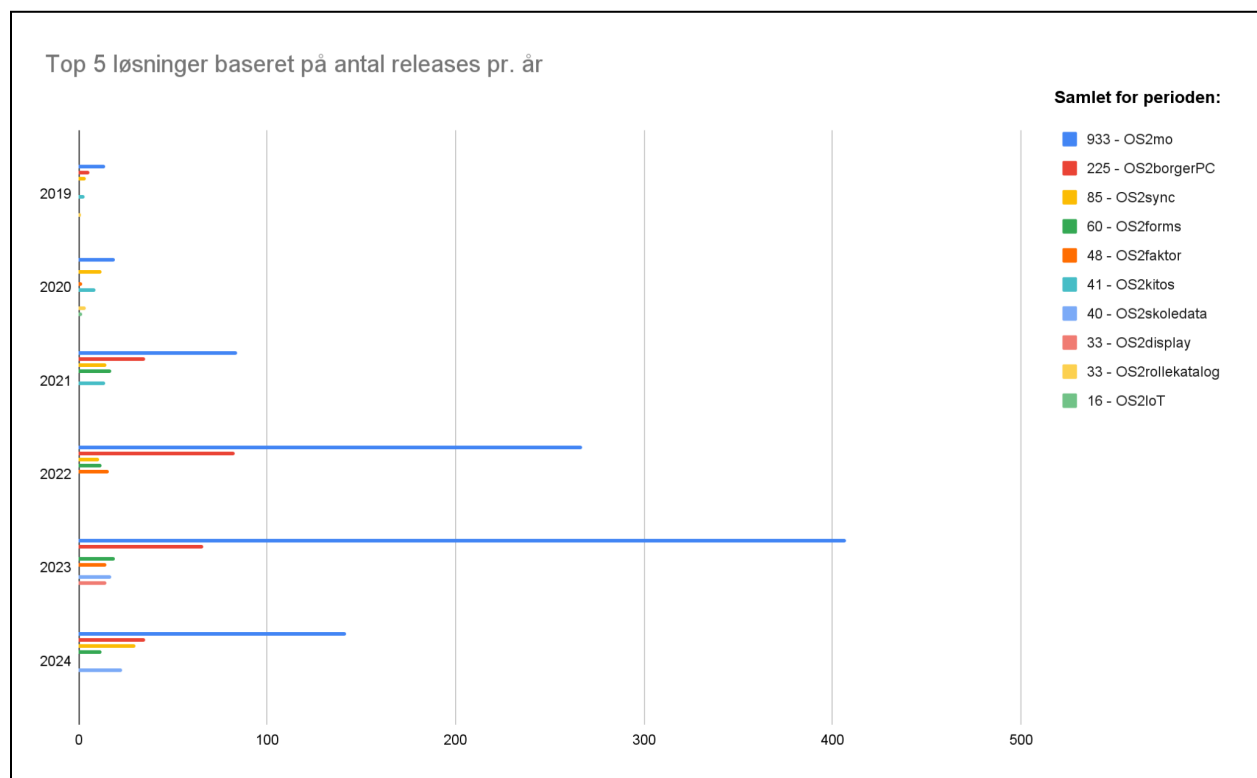
Overblik - commits og releases

Nedenfor ses to diagrammer, der illustrerer den markante forskel på OS2-løsningerne, når det kommer til hyppigheden af commits og releases.

For hvert år har vi anført de fem løsninger, der har flest commits henholdsvis releases. Eftersom det ikke er de samme løsninger, der hvert år er i top fem, så fremgår der flere end fem løsninger af diagrammet.

Løsningerne er ikke listet alfabetisk, men efter antal commits og releases i alt over de seks år i faldende orden. Til venstre for hver løsning er summen af commits henholdsvis releases anført.





Tallene i perspektiv

Top tre løsninger med flest releases

Antal releases på OS2mo: 961

Antal releases på OS2borgerPC: 236

Antal releases på OS2sync: 85

Opgjort i procent

Magentas to løsninger, OS2mo og OS2borgerPC står alene for 74% af samtlige releases på tværs af de 20 OS2-løsninger. De resterende 18 løsninger udgør således kun 26% af alle releases.

Det er vigtigt at pointere, at der er forskel på produkterne. De er forskellige i størrelse, og de kræver forskellige grader af kontinuerlig udvikling.

Endelig er nogle af produkterne nyere end de seks år, som analysen dækker over. Dette kan dog ikke begrunde den massive forskel, der ses mellem produkterne på diagrammerne. Når OS2mo og OS2borgerPC udgør 74% af alle releases på tværs af produkterne, og de resterende produkter tilsammen kun udgør 26%, så fortæller det en historie om, at det koster tid, kræfter og omhyggelighed at udvikle open source-løsninger.

Top tre produkter med flest commits

Antal commits lavet af OS2kitos: 14.488

Antal commits lavet af OS2mo: 9.923

Antal commits lavet af OS2borgerPC: 3.155

Opgjort i procent

Tages der udgangspunkt i OS2kitos, udgør dette projekt alene 41% af alle commits på tværs af løsningerne. Tages OS2mo og OS2borgerPC med i disse beregninger, udgør de tre løsninger tilsammen 78% af commits i tidsperioden.

En transparent udviklingshistorik betyder, at tredjepart let kan få et overblik over ændringer lavet til en løsning. Dette overblik understøttes, hvis løsningen har hyppige releases, som indeholder et moderat antal commits. Ses der på commit/release ratioen på løsningerne ovenfor er ratio på:

OS2kitos: 198,4

OS2mo: 10,3

OS2borgerPC: 13,3

OS2sync: 0,2

Det betyder tallene

En lav ratio viser få commits per release. Eksempelvis betyder en ratio på 1, at der er et commit pr. release, og commitet indeholder derfor alle ændringer lavet til et release. OS2syncs ratio indikerer derfor, at releases ikke har haft tilsvarende commits, og at der derfor ikke har været adgang til kildekoden for nogle af de releases, der er lavet til løsningen.

En høj ratio betyder mange commits per release. OS2kitos' ratio indikerer derfor også, at der foretages mange commits mellem releases, hvilket kan gøre udviklingshistorikken mindre overskuelig for tredjepart, som vil sætte sig ind i ændringerne foretaget mellem releases.

Derudover er der generelt en mangel på dokumentation af releases. Mange OS2-produkter, såsom OS2kravmotor, beskriver for eksempel ikke, hvilke ændringer der er foretaget mellem releases. Andre produkter indeholder kun én sætning pr. release. Det gør det umuligt for tredjepart at gennemskue, hvilke specifikke ændringer der er foretaget mellem versionerne. I nogle tilfælde er releases markeret med commits, der blot hedder "release" uden yderligere beskrivelse, hvilket gør det helt umuligt at forstå, hvad der er ændret.

Et mønster, der går igen på tværs af flere produkter, er, at commits til GitHub indeholder store kodeændringer, der ikke er dokumenteret i en changelog.

Denne adfærd er ikke befordrende for et åbent udviklingsmiljø, sådan som man kan forvente i et open source-projekt. Ændringer og opdateringer er ikke offentligt tilgængelige, før store releases bliver offentliggjort. Det er tydeligt i projekter som OS2skoledata, OS2sofd og OS2sync, hvor der er meget få commits. De refererer typisk til releases frem for mindre features og bugfixes.

I projektet OS2sync beskrives samtlige commits som "release <versionsnummer>" uden yderligere information om indhold eller ændringer. Det er derfor svært at få et klart billede af, hvordan systemet udvikler sig, og hvad den seneste version af systemet faktisk indeholder, ligesom det er umuligt at få indsigt i den faktiske udviklingshistorik.

Desuden kan man langtfra være sikker på, at kildekoden på GitHub faktisk er den seneste version af systemet. På leverandørens hjemmeside eller Docker-miljø - hvis det overhovedet er tilgængeligt - er versionen af systemet, der er tilgængeligt på GitHub, op til flere versioner bagud i forhold til det faktiske produkt. Som nævnt ovenfor i afsnittet *Strukturerede releases og adgang til kildekoden* er der på Docker Hub udgivet en version 4.6.4 af OS2sync, mens GitHub kun viser version 4.4.1.

Selvom kildekoden for alle de OS2-løsninger, der er gennemgået i dette notat, er frigivet på en MPL-2.0 licens, mangler helt centrale elementer for en stor del af løsningerne, hvis de reelt skal kunne gøre sig fortjent til betegnelsen 'open source'.

Desuden er der et tydeligt mønster i commit-strukturen på mange af de analyserede løsninger, såsom OS2kravmotor, OS2skoledata eller OS2faktor, som viser, at langt de fleste produkter udvikles i private repositorer. De commits, der skubbes til GitHub, er store, hvilket peger på en intern udviklingsproces, hvor små opdateringer og fejlrettelser ikke bliver offentliggjort, før de er blevet samlet i en større release. Dette betyder, at udviklingshistorikken ikke er gennemsigtig, og at eksterne udviklere kun har adgang til de store opdateringer - ikke de løbende ændringer. Det svækker i stor grad åbenheden.

2. Åbne kodebiblioteker og uafhængighed af proprietære licenser

Flere af de løsninger, der er gennemgået i dette notat, benytter sig af licensbaserede, lukkede biblioteker. Det gælder blandt andet OS2sofd, OS2kravmotor og OS2rollekatalog, der alle anvender et licens- og betalingsbaseret Twitter Bootstrap-tema kaldet "Angle." Da bibliotekets filer er binære, betyder det, at man ikke ved, hvad filerne indeholder, og dermed er man tvunget til at anvende lukket kode. Et andet produkt, som lider under samme afhængighed, er OS2korrespondance, der benytter et andet licensbaseret tema, "Inspinia", som også kræver betaling.

Selvom disse temaer ikke nødvendigvis kompromitterer løsningernes funktionalitet, vil det stadig kræve, at man enten køber en licens eller bruger et licensbaseret bootstrap-tema for at kunne anvende systemet. Hvis der bruges et nyt tema, er der risiko for, at koden skal tilpasses, da nogle bootstrap-temaer kan afvige fra standard bootstrap-strukturen. Ved at bruge disse biblioteker skabes der et niveau af lukkethed og tvunget brug af lukket software.

For at løsninger som OS2sofd, OS2kravmotor, OS2rollekatalog og OS2korrespondance reelt kan have et open source-stempel, bør der som standard anvendes et open source-tema. Brugen af licensbaserede temaer betyder, at den samlede løsning ikke kan benævnes open source, fordi man ikke frit kan bruge, undersøge, ændre og videregive den samlede løsning. Ved at vælge en open source-løsning som standard-tema undgår man afhængighed af proprietære temaer som "Angle" eller "Inspinia". Disse kunne i stedet tilbydes som et valgfrit tilvalg og eventuelt anbefales i dokumentationen for dem, der ønsker at benytte dem.

Selvom alle produkter er distribueret under MPL-2.0-licensen, er det tydeligt, at mange af projekterne ikke overholder licensens krav. For eksempel mangler projekter som OS2kravmotor, OS2autoproc og OS2rollekatalog den påkrævede Exhibit A som boilerplate i kildekodefilerne.

«««««

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

«««««

Ifølge punkt 1.4 i MPL-2.0:

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

Dette betyder, at kildekode-filer eksplicit skal indeholde "Exhibit A" for at være dækket af licensen. Ifølge Q4 i Mozillas officielle FAQ⁴ skal dette enten indsættes direkte i hver fil eller alternativt håndteres via REUSE/SPDX-headers.

Et eksempel er OS2mo, som anvender REUSE/SPDX-headers i kildekodefilerne. Fraværet af sådanne licensmeddelelser i andre OS2-projekter kan skabe juridisk uklarhed om, hvorvidt kildekoden formelt set er dækket af MPL-2.0.

Ved at anvende en anden licens kan man komme uden om dette krav. Men i den nuværende situation har OS2 lagt sig fast på MPL-2.0, og det er den licens, man som leverandør må følge, uanset om man måtte være uenig i kravet - eller licensen.

⁴ <https://www.mozilla.org/en-US/MPL/2.0/FAQ/>

3. Anvendelig dokumentation og README-filer

God dokumentation gør koden tilgængelig, forståelig og brugbar for både udviklere og slutbrugere, hvilket sikrer, at projektet er transparent og kan vedligeholdes og videreudvikles.

Nogle nedslagspunkter: OS2korrespondance indeholder ingen brugbar dokumentation. Der er en README.md-fil og en CONTRIBUTING.md-fil. Men de er alene skabeloner til, hvordan disse filer burde bygges op, og de indeholder således ikke nogen produktspecifik information.

OS2faktor har fragmenteret dokumentation, hvor vigtig information omkring kørsel og opsætning af OS2faktor-MFA kun findes i "OS2faktor-Login"-repoet frem for at være tilgængelig under det "OS2faktor-MFA"-repo, som det omhandler. Dette betyder, at tredjepart skal lede efter nødvendig information på tværs af flere repositorier, hvilket forringer brugernes mulighed for at sætte sig ind i projektet.

Flere af de henvisninger, der oplyses i governance-rapporten for OS2kravmotor, linker til sider, som ikke længere eksisterer. Det gælder eksempelvis links, der skulle vise løsningens organisationsstruktur eller dokumentere, hvorvidt udbudsregler og lovgivning er overholdt.

Derudover henviser flere løsninger fortsat til deres tidligere Bitbucket-repositorier, hvilket indikerer, at governance-rapporten ikke har været gennemgået eller opdateret, siden projektet blev flyttet til GitHub i 2022.

Når det ikke er muligt at følge de oplyste henvisninger i governance-rapporten og det samtidig fremgår, at rapporten ikke er blevet opdateret i minimum to år – hvordan kan man så have tillid til, at det angivne modenhedsniveau på OS2's hjemmeside er retvisende?

På OS2mo er den samlede dokumentation offentligt tilgængelig via en dedikeret dokumentationsside⁵. Denne tilgang sikrer, at løsningen er tilgængelig for både interne og eksterne udviklere.

Gennemgangen viser, at der er stor variation i kvaliteten af dokumentationen på OS2-løsningerne. Nogle få løsninger har en god dokumentationspraksis, mens de fleste har en fragmenteret og svært tilgængelig eller helt utilstrækkelig dokumentation.

README-filer

En velformuleret og struktureret README-fil giver en præcis beskrivelse af løsningens formål, installationskrav og vejledning, sådan at anvendere kan komme i gang med udvikling eller brug. Analysen af OS2-produkterne viser, at README-filer ofte er mangelfulde og i nogle tilfælde direkte fraværende.

⁵ <https://rammearkitektur.docs.magenta.dk/os2mo/>

Et eksempel på en README-fil med minimal information findes i OS2faktor-Login, hvor filen kun indeholder en enkelt kommandolinje til sikkerhedsscanning. Der er ingen yderligere vejledning om projektets formål, eller hvordan tredjepart kan anvende løsningen.

Lignende ses i OS2kravmotor, hvor README-filen kun indeholder en kort introduktion og licensbetingelserne. OS2rollekatalog har en README-fil, men den giver alene en kort beskrivelse af produktet og henviser til, hvor en del af dokumentationen kan findes.

Der findes også eksempler på projekter, hvor README-filen eksisterer, men er tom eller uden relevant indhold. OS2skoledata har eksempelvis en README.md-fil, men denne indeholder udelukkende projektets navn. OS2sofd har slet ingen README.md-fil i roden af projektet. En README.txt-fil findes længere nede i projektets mapper, men den indeholder kun ganske få detaljer om et enkelt API-endpoint. Ellers er der ingen informationer.

Omvendt findes der også produkter med overskuelige og anvendelige README-filer. På OS2mo indeholder README-filen en detaljeret beskrivelse af projektet samt en hurtig guide til opsætning af et udviklingsmiljø. Dette gør det lettere for eksterne bidragsydere at komme i gang med at anvende løsningen. Eller bidrage, hvis man måtte ønske det.

Samlet set viser analysen, at der er en meget stor variation i kvaliteten af README-filer i OS2-produkterne. Nogle projekter mangler README-filer helt, mens andre har minimal information, der ikke giver en fyldestgørende forståelse af løsningen.

4. Changelogs og dokumentation af fejlrettelser

En velstruktureret changelog er vigtig for at sikre transparens og en forståelse af projektets udviklingshistorik. Der er stor forskel på, hvordan changelogs bruges på tværs af OS2-produkterne.

Et eksempel på projekter, hvor changelogs er en integreret del af udviklingsprocessen, er OS2mo og OS2forms. OS2mo viser en detaljeret, automatisk opdateret changelog på løsningens hjemmeside.⁶ Changeloggen inkluderer også ændringer, der endnu ikke er blevet released, og det er derfor muligt at følge løsningens udvikling mellem releases. Changelogs kan således bruges til at skabe overblik over udviklingen af et produkt.

I modsætning hertil er de fleste andre OS2-løsninger fragmenterede eller helt uden changelogs. Et eksempel er OS2autoproses, som har en changelog-fil, der kun indeholder én enkelt opdatering fra 2021. Ligeledes findes der ingen changelogs overhovedet i OS2kravmotor eller OS2sync. Det gør det umuligt at få overblik over ændringer mellem versioner uden manuelt at gennemgå commits.

Et andet problem ses i projekter som OS2faktor, hvor changelogs er spredt over flere mapper og kun indeholder summariske beskrivelser. For eksempel:

"2023-05-02 - version 2.0.0 - Major performance improvements and improved reconnect logic."

Denne type changelogs giver i bedste fald en overfladisk forståelse af de ændringer, der er foretaget mellem releases. De kræver, at udviklere søger yderligere information ved at skulle lede igennem de mange kodeændringer, som er tilføjet i det tilsvarende commit. Det samme gælder OS2skoledata.

Tilsvarende findes der i produktet OS2korrespondance formelt changelogs, men de fungerer mest som placeholders, og de har intet reelt indhold.

Et udbredt problem er manglende referencer til changelogs i projektets rod. I mange af løsningerne skal der graves ned i projektets mapper for at finde frem til changelogs - hvis de altså eksisterer. Der er sjældent en samlet changelog-fil, men flere filer er opdelt i separate mapper for de enkelte komponenter. Det gør det stort set umuligt at opnå et reel billede af projektets udviklingshistorik. Denne fragmentering af changelogs svækker muligheden for at få indsigt i systemets udviklings-historik.

⁶ <https://rammearkitektur.docs.magenta.dk/os2mo/changelog.html>

Opsamling

Samlet set viser analysen, at de fleste OS2-produkter ikke består ét eller flere af de krav, der kendetegner et åbent og transparent open source-projekt. Der er store barrierer for både at forstå, anvende og bidrage til løsningerne, og det er tydeligt, at udviklingen foregår privat, og at ændringer bliver offentliggjort i store kode dumps, efter at de er gået i produktion.

Manglen på åbenhed, gennemsigtighed og afhængighed af lukket kode bevirker, at disse produkter ikke er open source.

I skemaet nedenfor samler vi op på en evaluering af de fire kategorier for samtlige løsninger. Løsningerne er listet alfabetisk, således at de leverandører, der kun har én løsning, listes først. Sorteringsordenen er som sagt alfabetisk efter løsningen og ikke leverandøren.

Fem leverandører har én og kun én løsning tilknyttet. OS2flytjord og OS2nectar er som fremhævet inaktive projekter uden kildekode, så NIRAS og Novataris tæller ikke med i aktive leverandører.

To leverandører har hver to løsninger - ITK-Development og Magenta.

Endelig har Digital Identity samlet ni OS2-løsninger.

Otte leverandører til OS2 - ikke 64

Der er altså samlet otte leverandører til OS2. OS2's oversigt over leverandører er dermed langt fra retvisende. På den ene side viser vores data, at AestasGIS og Omnia fungerer som driftsleverandører på hver deres OS2-løsning. Alligevel er ingen af disse to nævnt i OS2's officielle leverandøroversigt.

På den anden side fremgår det, at kun seks af de leverandører, der er angivet i oversigten, faktisk har et aktivt produkt, som de leverer på. Dertil kommer den nye leverandør, Droids Agency, der har lanceret produktet OS2FleetOptimiser. Det er ikke taget med i analysen her, eftersom første commit er i 2025, og analysen stopper med udgangen af 2024.

Endelig kan det diskuteres, om OS2kravmotor er en aktiv løsning, eftersom der er ikke blevet lavet commits til projektet i GitHub i 3 år.

Der er en markant uoverensstemmelse mellem det, OS2 kommunikerer på foreningens website, og de faktiske leverandører til OS2's projekter. Det gør det derfor også vanskeligt at få et reelt overblik over, hvem der reelt står bag udvikling og drift af OS2's løsninger.

Samlet er der syv leverandørvirksomheder plus Aarhus Kommunes digitaliseringsenhed, ITK-Development, på de 18 løsninger, der er aktive (19 med OS2FleetOptimiser). OS2's liste med 64 leverandører er således ikke retvisende. OS2 accepterer leverandører, der ikke har - og aldrig har haft - en open source-løsning i OS2-regi. At forholdet er tæt på 1:10 mellem faktiske

og potentielle leverandører er måske lige i overkanten, hvis OS2 skal fastholde en eller anden form for troværdighed omkring leverandørfeltet.

OS2-løsninger i overblik

GRØN = Ingen væsentlige anmærkninger GUL = Mangler vigtige elementer RØD = Kritisk				
OS2-løsning	Strukturerede releases og adgang til kildekoden	Åbne kodebiblioteker og uafhængighed af proprietære licenser	Anvendelig dokumentation og README-filer	Changelogs og dokumentation af fejlrettelser
NIRAS				
OS2flytjord (inaktivt)	RØD	RØD	RØD	RØD
Netcompany				
OS2iot	GRØN	GRØN	GRØN	GUL
Strongminds				
OS2kitos	GRØN	GRØN	GUL	GUL
Novataris				
OS2nectar (inaktivt)	RØD	RØD	RØD	RØD
AestasGIS				
OS2-SkadesØkonomi	GUL	GRØN	GUL	RØD
Novicell				
OS2udoglær	GRØN	GRØN	GRØN	GUL
Omnia				
OS2valghalla	GRØN	GRØN	GUL	GUL
ITK-Development				
OS2display	GRØN	GRØN	GUL	GRØN
OS2forms	GRØN	GRØN	GRØN	GRØN

GRØN = Ingen væsentlige anmærkninger GUL = Mangler vigtige elementer RØD = Kritisk				
OS2-løsning	Strukturerede releases og adgang til kildekoden	Åbne kodebiblioteker og uafhængighed af proprietære licenser	Anvendelig dokumentation og README-filer	Changelogs og dokumentation af fejlrettelser
Magenta				
OS2borgerPC				
OS2mo				
Digital Identity				
OS2autopoces				
OS2compliance				
OS2faktor				
OS2korrespondance				
OS2kravmotor				
OS2rollekatalog				
OS2skoledata				
OS2sofd				
OS2sync				

Del 2 - Gennemgang af de enkelte OS2-produkter

OS2autoprocess

Test og Deployment

OS2autoprocess' frontend bygges ved brug af docker. Det fremgår af den Dockerfile, der er tilgængelig i roden af repoet. Derudover beskrives dette også i README filen. Der findes ikke en guide til deployment. Der bliver alene beskrevet, at dette håndteres af kunden, og at projektet bygges som en del af et docker image. Backend-repoet kompiles ved brug af Maven (<https://maven.apache.org/>).

Det beskrives kort, hvordan et lokalt udviklingsmiljø kan køres op ved at bruge "npm run serve". Dette forudsætter imidlertid, at backenden først kører på <https://dev.os2autoprocess.eu/>

Struktur og Dokumentation

Projektets dokumentation er fragmenteret. Den nødvendige dokumentation om bidrag og opsætning er kun tilgængelig i frontend-repositoriet. En centralisering af dokumentationen og bedre forklaring af opsætningsprocessen er nødvendig for at gøre projektet lettere tilgængeligt for eksterne udviklere.

Afhængigheder

Visse dokumenter relateret til projektet opbevares på et lukket drev i OneDrive og kræver, at der anmodes om adgang. Dette begrænser tilgængeligheden af dokumentation og skaber unødvendige barrierer for eksterne udviklere og interessenter.

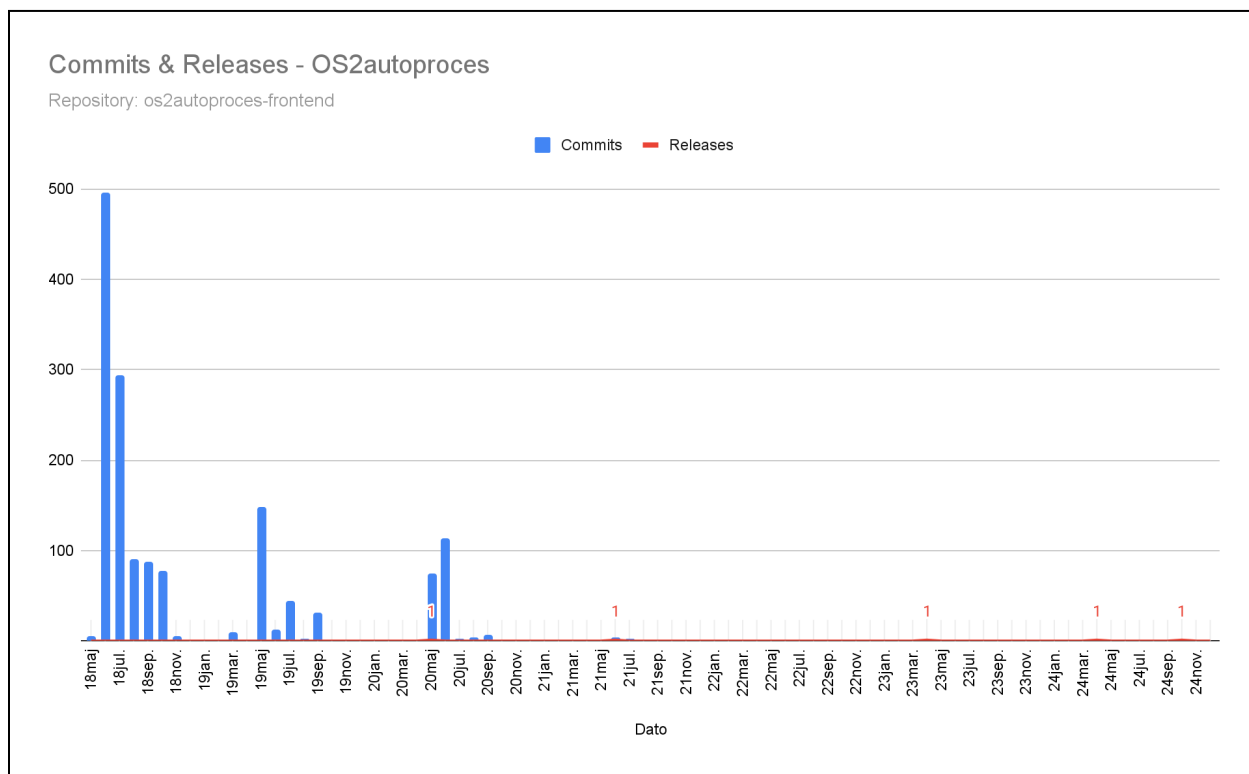
GitHub Commits, Releases og Versionering

Projektets GitHub aktivitet er begrænset. Backend-repositoriet havde kun 12 commits i løbet af 2024, mens frontend-repositoriet har haft 9 commits siden version 1.0.0 blev udgivet i 2021 - altså 9 commits på 4 år. I alt har dette repository haft 1.516 commits i løbet af dets levetid. Sidste commit blev skubbet i oktober 2024. Antallet af commits til repositoryet falder imidlertid markant i forbindelse med produktets overgang fra Bitbucket til GitHub i april 2023. Repoet har kun haft 2 commits/releases siden denne overgang.

Gennemsnittet af kodeændringer, et commit har indeholdt, lå lavt før 2022, men i de seneste år er der skubbet større ændringer til kodebasen på meget få commits. Det lader til, at der først bliver skubbet commits til repoerne, når der er foretaget større ændringer i projektet frem for små hyppige commits, der gør det overskueligt at følge med i projektets udvikling.

Der findes 1 release tag i repoet. Alle andre releases er udgivet som et commit med titlen "release", men mangler beskrivelser af ændringerne foretaget mellem releases. Der er ingen hjælp at hente i changelog filen, da den kun refererer til dokumentationen, som heller ikke indeholder en beskrivelse af ændringer.

For at finde frem til, hvor mange releases projektet har haft, har vi forsøgt at finde information om releases og versioner på andre platforme end GitHub. Det har ikke været muligt at finde supplerende information, hverken på løsningens, leverandørens eller OS2's hjemmesider. Vi har heller ikke kunnet finde container-images på Docker Hub eller i GitHubs eget container registry.



Af diagrammet fremgår det, at alle projektets releases har tilsvarende commits.

OS2borgerPC

Test og Deployment

I roden af os2borgerpc-admin-site repoet findes en docker-mappe, som indeholder en Dockerfile og relaterede filer til det Docker-image, der anvendes. Dermed er det forholdsvis nemt for tredjepart at deploye adminsitet ved brug af Docker. Derudover er det gjort enkelt for brugere at bygge projektet ved at kunne bruge Docker Compose.

Struktur og Dokumentation

OS2borgerPC's dokumentation dækker alle aspekter af projektet – både tekniske og brugervendte. Der findes vejledninger, der guider brugere gennem hele opsætningsprocessen. Derudover indeholder dokumentationen en detaljeret guide til, hvordan både kiosk og borgerPC installeres lokalt samt i et produktionsmiljø.

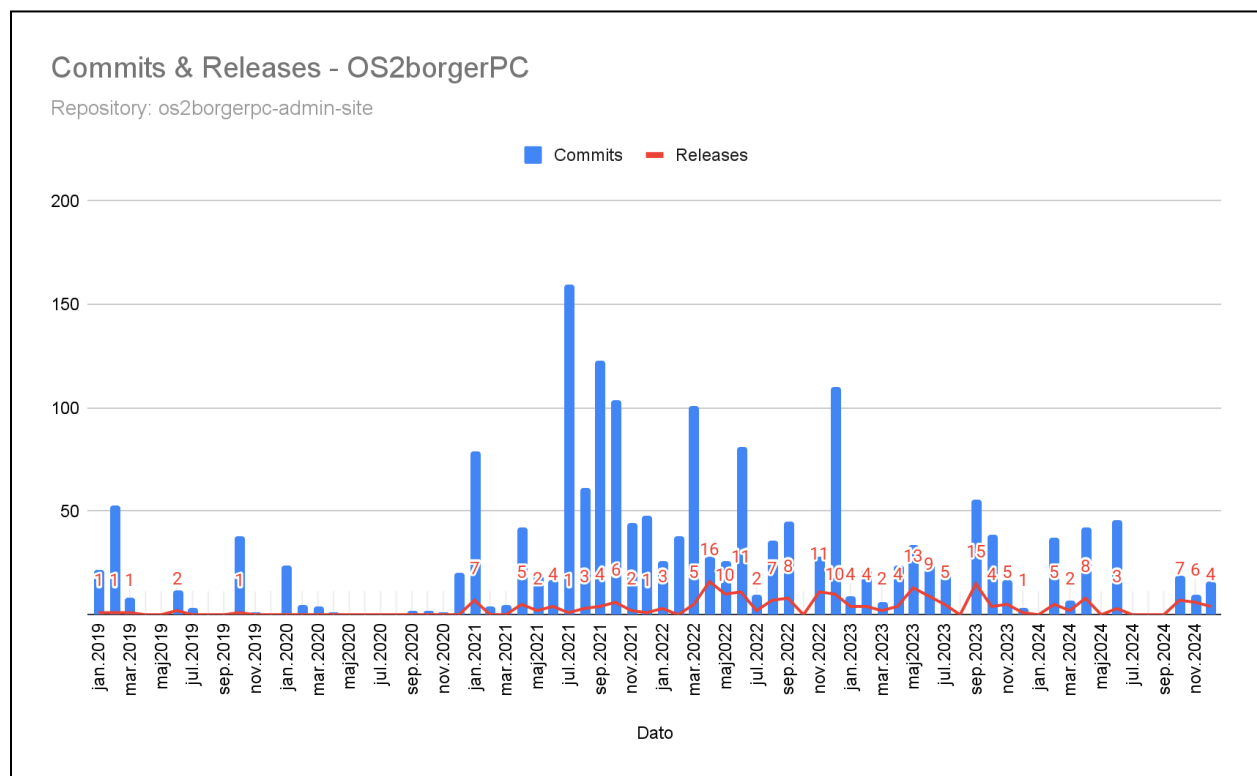
Adminsitet er veldokumenteret med omfattende guides, der giver tekniske og ikke-tekniske brugere mulighed for at navigere og udnytte de tilgængelige funktioner.

README-filen giver en kort beskrivelse af systemet og henviser til yderligere dokumentation, hvor brugere kan finde mere dybdegående information om blandt andet, hvordan man bidrager til projektet.

På ReadTheDocs findes changelogs med detaljerede oplysninger om de ændringer, som er lavet mellem releases. Disse kan også findes i GitHub repoet under "NEWS.md" filen. Det skaber en transparent releasehistorik, der giver tredjepart mulighed for at følge udviklingen af systemet tæt.

Afhængigheder

Det har ikke været muligt at finde afhængigheder til lukkede biblioteker eller software.



GitHub Commits, Releases og Versionering

os2borgerpc-admin-site-repoet har haft i alt 3.232 commits siden 6. maj 2013. Disse commits har omfattet mindre kodeændringer, der gør det overskueligt at se ændringerne, som er foretaget mellem releases. Repoet gør brug af release-tags, som giver et klart overblik over, hvilke dele af koden der er blevet opdateret og hvornår. I alt er der 230 release-tags på os2borgerpc-admin-site-repoet.

Ved at standardisere versioneringen er det muligt at følge udviklingen af koden og se sammenhængen mellem release-tags og changelogs. Dette sikrer den gennemsigtighed, som forventes af et open source-projekt, og det giver både udviklere og tredjepart et præcist billede af systemets opdateringer og forbedringer.

OS2compliance

Test og Deployment

Vejledningen til deployment er relativt sparsom og indeholder primært information om systemkrav, om at projektet deployes via Docker-images med indbygget JDK samt en liste over konfigurationsparametre. Man guides ikke til, hvordan projektet rent faktisk deployes, men snarere i hvad der skal være på plads før deployment.

Derudover anvendes GitHub Actions som CI-pipeline til at teste og bygge projektet.

Struktur og Dokumentation

OS2compliance-projektets README-fil er fyldestgørende og indeholder en let forståelig oversigt med henvisninger til relevant dokumentation.

Dokumentationen er opdelt mellem en GitHub Pages-hjemmeside og filer direkte i GitHub-repoet. Dokumentationen på GitHub Pages er primært rettet mod vejledning og information om, hvordan projektet bygges og køres samt information om deployment.

Det er muligt at få indsigt i projektets udviklingshistorik ved at tilgå Changelog-filen i projektets rod. Denne fil giver tredjepart mulighed for hurtigt at få overblik over ændringer mellem releases og forstå, hvordan projektet har udviklet sig over tid.

Derudover findes en Contributing-fil, som giver retningslinjer for, hvordan eksterne udviklere kan bidrage til projektet via GitHub.

Der er imidlertid ingen dokumentation, der beskriver, hvordan løsningens API fungerer, hvilke data det modtager eller sender eller dets endpoints. Uden denne information er det vanskeligt for tredjepart at forstå, hvordan man kommunikerer med API'et.

Afhængigheder

Der ses ikke nogen afhængigheder til leverandøren eller lukket software i projektet.

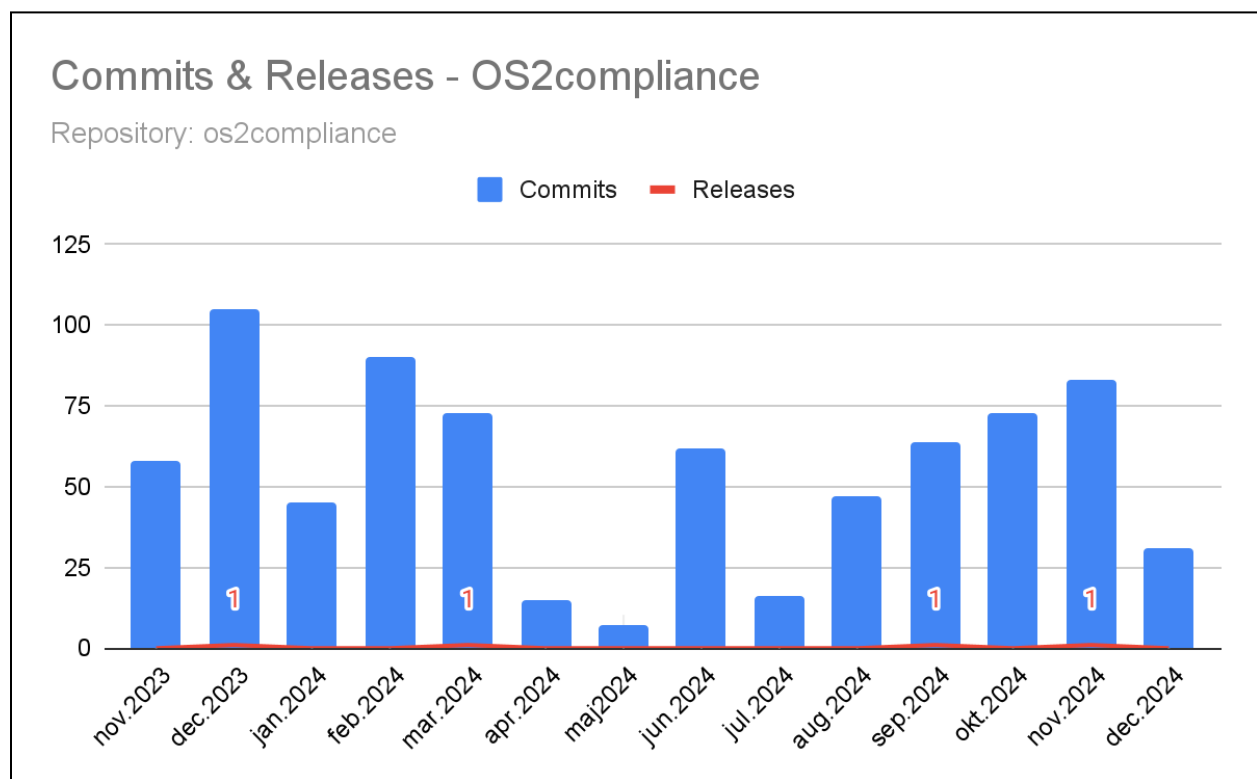
GitHub Commits, Releases og Versionering

Siden november 2023 har OS2compliance-projektet på GitHub haft i alt 769 commits. Disse commits varierer i størrelse, hvor leverandøren i nogle måneder tilføjer 1.200 linjer kode per commit, mens leverandøren i andre måneder kun tilføjer omkring 6 linjer kode. I gennemsnit, fra november 2023 til december 2024, indeholder hvert commit omkring 230 linjer kode.

Det er optimalt at der tilføjes så få linjers kode som muligt per commit, da mindre commits gør det nemmere at identificere de præcise ændringer, der er blevet foretaget, og hvilken funktionalitet der er blevet tilføjet eller rettet – såsom bugfixes eller nye features.

Der er oprettet 4 releases i projektet, som kan ses via release tags på GitHub. De 2 første releases, der kan ses på GitHub, er oplyst til at være udgivet i juli 2024. Ser man på tags på repoet samt på de container images, der kan findes på GitHub, lader det imidlertid til, at disse releases er udgivet i december 2023 og marts 2024.

Indholdet af releases stemmer overens med opdateringerne i projektets Changelog-fil, hvilket gør det nemt at få indsigt i de vigtigste ændringer mellem de forskellige versioner af systemet. Desværre har det ikke været muligt at finde versionoplysninger andre steder end på GitHub, og det er derfor ikke muligt at sammenligne, om versionen, der findes på GitHub, også er den nyeste version af systemet, der er tilgængelig.



OS2display

Test og Deployment

Projektet på GitHub giver brugbare og letforståelige vejledninger i at bygge og køre development-, test- og demo-miljøer. Nogle af disse vejledninger er imidlertid spredt ud på forskellige repo'er. Vejledningen til at køre test- og demo miljøer findes i et "os2display-docker-compose"-repo, og vejledningen til development-miljøer findes i "display-docs"-repoet. Development-guiden informerer om, at OS2display er bygget op af 4 dele (API, Admin, Client og Templates), som også findes som individuelle repo'er på GitHub. Der skal derfor også bruges docker-compose.yml filer fra de respektive repo'er for at køre projektet.

Af "Hosting" i dokumentationen fremgår det, at der findes docker images for de enkelte apps (API, Admin, Client, Templates), og at disse bygges ved brug af GitHub Actions. Denne dokumentation informerer og vejleder også om, hvordan projektet kan bruge Docker til deployment.

Struktur og Dokumentation

OS2display's opdelte repo-struktur gør det svært at få en overordnet forståelse af systemet. Men dykker man ned i dokumentations-repoet "display-docs", findes der et link til dokumentationens hjemmeside, hvor man får en kort oversigt over interessepunkter. Det er dog ikke den samlede dokumentation, der er nævnt på siden.

For eksempel henvises der til en ny vejledning i et "display-docker-server" repo i GitHub under vejledningen for Docker deployment. Denne information er således ikke tilgængelig i dokumentationen. Derudover er vejledningen til opsætning og kørsel af demo- og test-miljøerne ikke nævnt i dokumentationen, og tredjepart skal derfor selv finde frem til vejledningen ved at gennemgå repo'erne under projektet. Vigtig information for opsætning, bygning og kørsel af løsningen er således ikke tilgængelig i dokumentations-repo'et, og det kan derfor være nødvendigt at grave ned i alle af projektets repoer for at finde frem til nyttig information og få det fulde overblik af projektet.

Projektet indeholder changelogs for hver af de 4 dele af projektet. Derudover findes der en samlet changelog på dokumentations-hjemmesiden, der også viser, hvornår releases fandt sted for api, admin, client og templates repoerne. Releases er imidlertid ikke konsekvent afspejlet i changeloggen, da nogle releases ikke er opgivet, og nogle datoer ikke stemmer overens med, hvornår releases faktisk fandt sted.

Projektet indeholder ingen guides til, hvordan man bidrager til projektet.

Afhængigheder

Der lader ikke til at være afhængigheder til leverandøren eller lukket software i projektet.

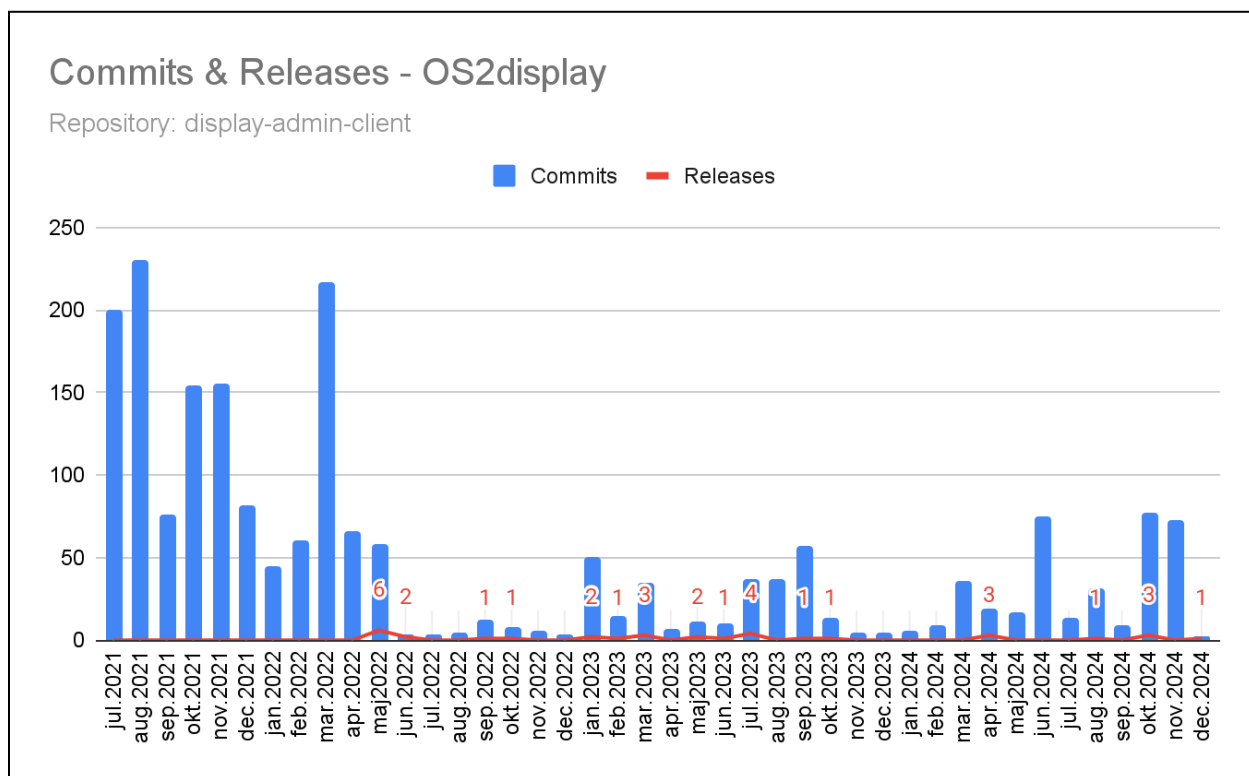
GitHub Commits, Releases og Versionering

OS2display-projektet er opdelt i flere repositories. Nedenfor er der taget udgangspunkt i det, der lader til at være det centrale repo - "display-admin-client" - da det er her, der er mest aktivitet.

Fra juli 2021 til december 2024 har der været 2.033 commits til dette repo. Disse commits er foretaget med korte mellemrum og indeholder mindre kodeændringer, hvilket giver et godt overblik over udviklingshistorikken i denne del af projektet. Der er lavet 32 releases, som kan ses via release-tags. 11 af disse releases - fra version 1.3.1 til 2.1.1 - kan også findes som releases på GitHub repoet. Versioner også udgivet som releases giver en yderligere changelog, hvor commits forårsaget som en del af releasen, er præsenteret. Der kan dog stadig findes changelogs fra versionerne 1.3.0 til 1.0 i repoets changelog-fil. Hyppige releases bidrager til en overskuelig udviklingshistorik.

For at få det fulde billede af antallet af projektets releases er der søgt efter releases af container images på andre platforme end kun GitHub. Det oplyses i dokumentations-repoet, at images til admin client repoet findes på Docker Hub. På Docker Hub kan der fremfindes 27 releases.

Releases i diagrammet nedenfor er en samling af unikke releases på tværs af de to platforme.



OS2faktor

Test og Deployment

Deployment-processen kræver mange manuelle tekniske indgreb i flere forskellige dele af projektet for både Login- og MFA repositorerne, som er komplekse og vanskelige at følge.

Struktur og Dokumentation

Den læsbare dokumentation i projektet er minimal, og den resterende dokumentation skal findes frem ved at lede i underliggende mapper. Det er kun OS2faktor-Login repositoret, der har en "README.md"-fil i roden. Denne fil indeholder dog udelukkende:

```
"""
```

```
    Perform security scanning with
```

```
$ mvn dependency-check:check
```

```
"""
```

Det meste af den dokumentation, der findes i projektet, er binære filer i .docx-format, der kræver, at de downloades, før de er læsbare. Login-repositoriet indeholder en "HowToSetup.md"-fil, som vejleder evt. bidragere i at opsætte et udviklingsmiljø. Denne information kunne med fordel integreres i en samlet README.md for bedre overblik og for at følge best practice. Derudover mangler projektet en CONTRIBUTING.md-fil, som kan guide eksterne udviklere i, hvordan de kan bidrage.

Afhængigheder

Der kan ikke ses at være afhængigheder til leverandøren eller lukket software i projektet.

GitHub Commits, Releases og Versionering

Der findes ikke en "CHANGELOG.md"-fil i roden af repositoriernene. I stedet er de spredt på tværs af mapper. Derudover er indholdet af Changelog filerne summarisk og indholder typisk en enkelt sætnings beskrivelse per version. Eksempel:

```
"""
```

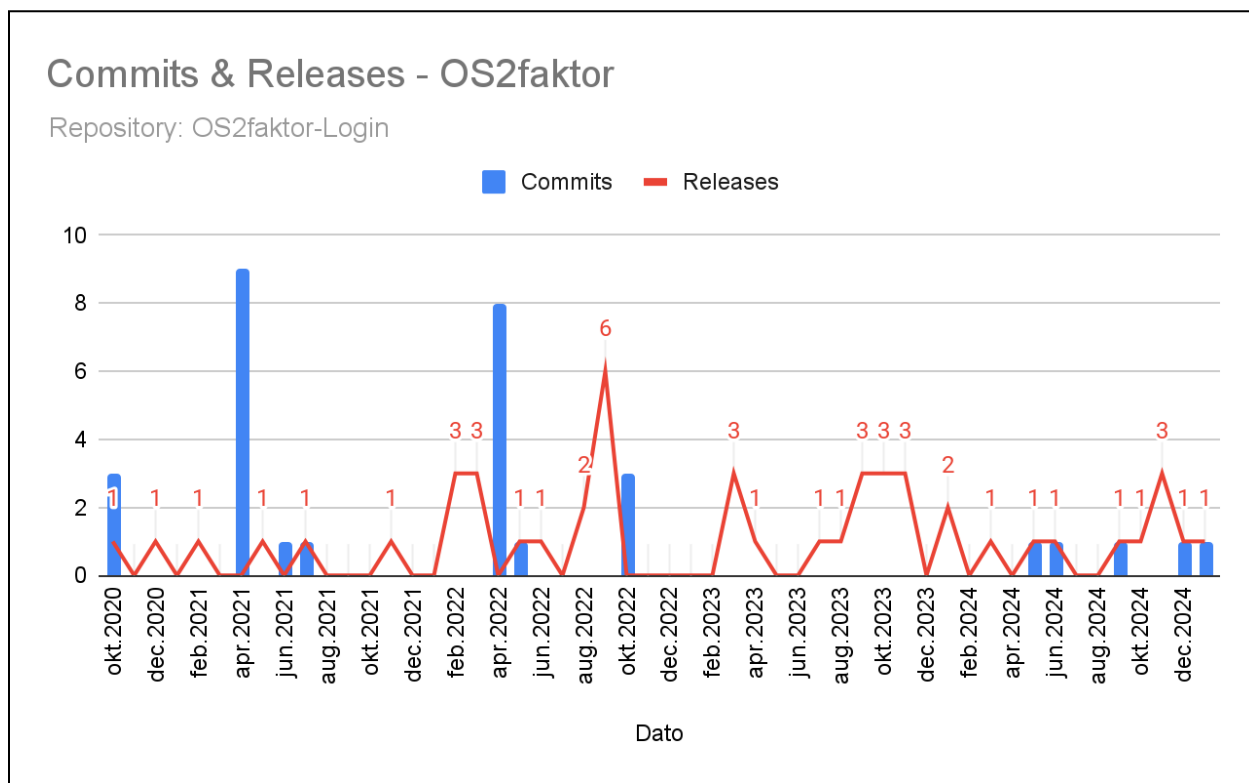
```
2023-05-02 - version 2.0.0 - Major performance improvements and improved reconnect logic
```

```
2023-06-17 - version 2.1.0 - Support executing a custom powershell script on password expiry events
```

```
2024-03-11 - version 2.2.0 - Include servername in logs to make debugging easier
```

```
"""
```

For at kunne afgøre, om den version, der var tilgængelig på GitHub, også var den seneste version i produktion, var det nødvendigt at søge på andre platforme, da det ikke var tydeligt via GitHub. Ved at søge igennem OS2faktors hjemmeside er det åbenbart, at versionerne i GitHub repoernes changelogs ikke stemmer overens med det reelle system. Et eksempel⁷ er, hvor version 2.14.0 ses som den sidste nye version. I det offentlige repo er samme changelog-fil imidlertid kun på version 2.11.0.



Tages der udgangspunkt i samme changelog, kan der også ses en uoverensstemmelse mellem commits og releases/versioner. Udviklingsprocessen foregår tilsyneladende privat, og kode pushes på et noget senere tidspunkt til de offentlige repositorier. Commits sker sjældent, og de indeholder store ændringer til kodebasen. Et andet tegn på fravær af gennemsækelighed er fx et commit med titlen "latest Jira issues merged into **public master**". Ingen uden internt kendskab til løsningen ved, hvad de seneste Jira issues består af.

Nogle commits er blot navngivet "release", men har hverken et release tag eller yderligere information om releasesets indhold. Versionerne på disse releases følger deres eget versioneringsformat, der ikke stemmer overens med formatet i diverse changelog. Dette gør det også svært at tyde, hvilke versioner der stemmer overens med releases.

⁷ <https://www.os2faktor.dk/download/OS2faktorCoreDataSyncChangelog.txt>

OS2flytjord

Der er ikke udgivet kildekode på løsningens projekt på GitHub. Projektets repoer indeholder tomme filer og skabeloner med information om, hvad filen burde indeholde.

Test og Deployment

OS2flytjord giver ingen information om, hvorvidt man bygger, kører eller deployer løsningen.

Struktur og Dokumentation

Der findes et [GitHub projekt](#) til løsningen, som der også henvises til fra OS2's hjemmeside, og som indeholder to repoer. Det ene repo, "os2flytjord-docs", er et repo ment til at indeholde dokumentationen for projektet. Der findes dog ingen informationer om projektet i dette repo.

Derudover findes der heller ikke nogen dokumentation i det andet repo, "os2flytjord", som ville kunne give tredjepart information om projektet.

os2flytjord-repoet indeholder alle de relevante filer, såsom changelog, contributing, readme mm. Men disse er kun skabeloner uden indhold, hvilket betyder, at filerne ikke giver nogen information om løsningen.

Afhængigheder

Da der ikke er noget kildekode overhovedet på projektets offentlige GitHub, er det heller ikke muligt at bygge, køre eller hoste løsningen - endsige gennemgå koden.

Manglen på adgang til kildekoden betyder, at man gør sig 100% afhængig af leverandøren for at få adgang til systemet.

GitHub Commits, Releases og Versionering

OS2flytjord har kun ét enkelt commit i løbet af de 7 måneder, projektet har været på GitHub (maj-dec 2024). Dette commit blev foretaget i maj 2024, og det indeholdt udelukkende de fil-skabeloner, som er nævnt ovenfor, hvilket vil sige, at intet af projektets kildekode er offentligt tilgængelig.

Der lader ikke til at være releases eller andet, der informerer om projektets udvikling.

OS2forms

Test og Deployment

Projektet kan startes op på forskellige måder, for eksempel med Docksal eller Drupal. Da der ikke findes en overordnet guide, er det nødvendigt at gennemgå forskellige repoer for at finde vejledning i opsætningen.

Der findes en vejledning til at starte en kopi af projektet op til udviklingsformål på to forskellige måder. Denne vejledning er beskrevet i README-filen i et af projektets tre kernerepoer. Guiden forklarer, hvordan projektet kan køres enten med Docksal eller "Traditional", hvor 'drush' (Drupal) anvendes. README-filen indeholder også instruktioner til deployment. Dog henviser guiden til en mere detaljeret beskrivelse af deploymentprocessen. Denne henvisning peger dog på en fil, der ikke længere eksisterer. Det er derfor uklart, om den detaljerede vejledning stadig er tilgængelig, eller om der kun findes en overordnet beskrivelse.

Projektet anvender GitHub Actions som CI-pipeline. I "os2forms8"-repoet er alle workflow-runs fejlet i "Build"-fasen.

Struktur og Dokumentation

Løsningens dokumentation er udelukkende beskrevet i de enkelte repoers README-filer. Det er derfor nødvendigt at læse på tværs mellem repoerne løbende for at få den fulde dokumentation samt den overordnede forståelse af løsningens tekniske opsætning.

Der findes et dokumentationsrepo, "os2forms_docs," som indeholder overordnet information om projektet, herunder licens, test og CI, git samt forløbet af et codereview. Denne information præsenteres i repoets README-fil. Selve dokumentationsmappen indeholder kun en skabelon til et overblik over OS2forms-moduler.

I README-filen under "os2forms8"-repoet beskrives, hvordan løsningen køres, samt hvordan deployment processen udføres. I "os2forms"-repoet beskrives hvordan løsningen installeres og opdateres.

README-filen i "os2forms_docs"-repoet oplyser, at alle OS2forms-projekter anvender EUPL v1.2-licensen. Dog findes der ikke nogen licensfil med disse licensoplysninger i nogen af projektets repositories. Derudover ser det ud til, at der ikke er tilknyttet en licens til "os2forms_forloeb_profile" og "os2forms"-repoerne, som ellers beskrives som to af de centrale dele af projektet.

Afhængigheder

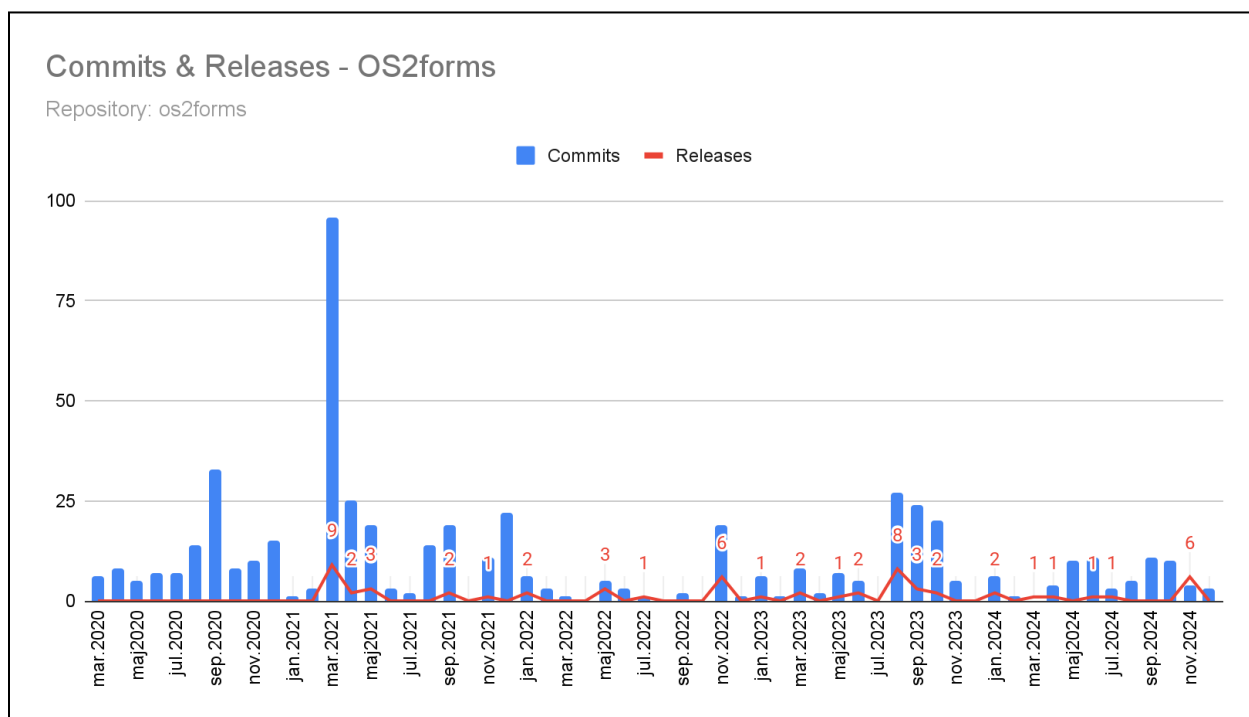
Der lader ikke til at være afhængigheder til leverandøren eller lukket software.

GitHub Commits, Releases og Versionering

I det følgende tages der udgangspunkt i det kerne-repo, der har mest aktivitet gennem tiden på

GitHub. "os2forms"-repoet har haft 690 commits siden november 2019 og frem til december 2024. Disse commits forekommer hyppigt og indeholder små kodeændringer, der repræsenterer bugfixes eller feature-tilføjelser.

Releases er markeret med release-tags, hvilket gør det nemt at følge repoets udviklingshistorik. Der er udgivet 114 tags på repoet, men kun 71 releases. Man skal derfor være opmærksom på, at alle release-tags ikke fremgår som releases i GitHub repoet, og det er derfor nødvendigt at finde information om nogle releases ved at kigge direkte i changelogs. Beskrivelserne af de ændringer, der er foretaget mellem releases, fremgår af changelogs, som yderligere bidrager til gennemsigtigheden af udviklingshistorikken.



På OS2's hjemmeside fremgår det, at OS2forms-løsningen udvikles af tre leverandører: Bellcom, ITK og Novicell. Ved at gennemgå de brugere, der har bidraget til repoet i løbet af 2024, ses det dog, at kun fire brugere har lavet commits til projektet. Tre af disse brugere er fra ITK og har tilsammen stået for 109 commits i 2024, mens den fjerde bidragsyder, fra Bellcom, har lavet 57 commits i samme periode.

OS2 angiver, at Bellcom er den primære udviklingsleverandør af projektet. Imidlertid viser commit-historikken, at ITK har stået for størstedelen af udviklingsarbejdet i 2024, så ITK må være den primære leverandør og spiller dermed en mere central rolle i den løbende udvikling, end hvad der kommunikeres på OS2's hjemmeside.

OS2iot

Test og Deployment

OS2iot's dokumentation har en detaljeret vejledning, der guider tredjepart i at sætte projektet op. Denne vejledning beskriver, hvordan projektet kan køres op lokalt ved hjælp af docker-compose, samt hvordan det kan køres i et Kubernetes-miljø. Dokumentationen dækker også deployment via Jenkins, hvilket gør det nemt for udviklere at implementere og vedligeholde systemet i forskellige miljøer.

Projektet anvender GitHub Actions og Azure pipeline som CI-pipeline, hvilket automatiserer test- og build-processer. Alle workflow runs foretaget i "OS2iot-backend"-repoet siden maj 2024 er imidlertid fejlet.

Struktur og Dokumentation

I dokumentationen fremgår det, at man for at kunne køre projektet skal anvende repoerne "OS2iot-frontend", "OS2iot-backend" og "OS2iot-docker". Disse repositories betragtes derfor som projektets kerne-repoer.

Derudover findes der et separat repo, "OS2iot-docs", som indeholder den samlede dokumentation for projektet. Ved at samle dokumentationen ét sted bliver det lettere for tredjepart at få et overblik over projektet, uden at man behøver gennemgå alle repoer individuelt.

Selvom der ikke findes en changelog i repoerne, er det stadig muligt at få indsigt i ændringer mellem releases ved at gennemgå release-tags. Dette giver mulighed for at følge projektets udviklingshistorik, men uden en egentlig changelog kræver det en manuel gennemgang.

Afhængigheder

Der lader ikke til at være afhængigheder til leverandøren eller lukket software i projektet.

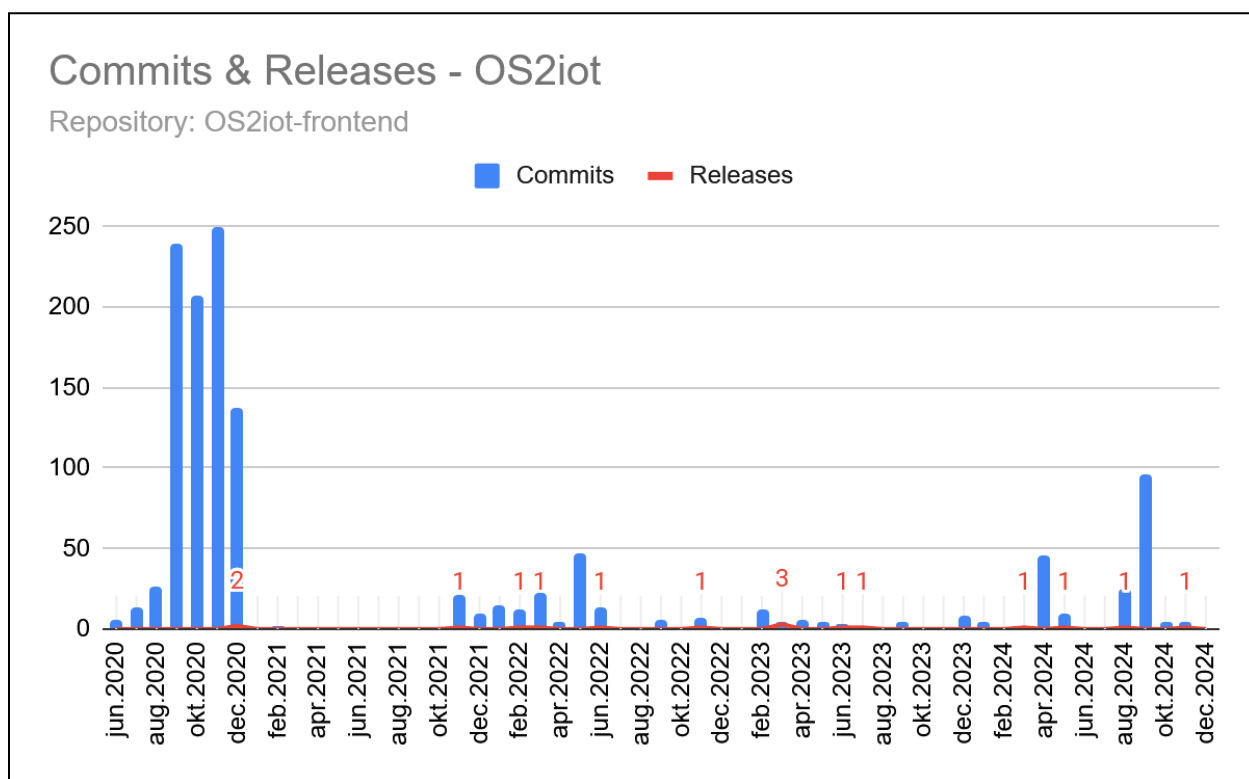
GitHub Commits, Releases og Versionering

Da projektet er opdelt i flere repoer, tages der udgangspunkt i det kerne-repo, der har set mest aktivitet på GitHub: "OS2iot-frontend".

Fra juni 2020 og frem til november 2024 er der foretaget 1.279 commits til "OS2iot-frontend"-repoet. Disse commits forekommer som regel hyppigt, og de indeholder primært mindre ændringer til kildekoden. Der er imidlertid perioder, hvor der foretages få commits med 1 til 2 måneders mellemrum. Disse perioder tilføjer oftest ikke store kodeændringer. Men i december 2023 er der tilføjelser af næsten 2.000 kodelinjer per commit i gennemsnit. I april 2024 på omkring 1.500 og august 2024 på over 3.500 kodelinjer. Små og hyppige commits og releases er klart at foretrække, da de holder udviklingshistorikken overskuelig og transparent for eventuelle tredjeparter.

Repoet har derudover haft 16 releases, som er identificeret gennem release-tags på GitHub. Anvendelsen af release-tags gør det nemt for tredjepart at få et overblik over projektets udvikling. For at sikre, at alle releases tælles med, er der søgt oplysninger om releases/versioner af projektet på andre platforme end kun GitHub.

Projektet bygges blandt andet ved at bruge Docker. Derfor er der søgt efter container images på Docker Hub. Desuden nævner OS2iot's dokumentation, at der under udvikling er brugt Azure Container Registry (ACR) til at opbevare container images. Det har ikke været muligt at finde andre informationer om releases eller versioner af projektet på andre platforme end GitHub, og det er derfor ikke muligt at sammenligne, om versionen tilgængelig på GitHub er samme version, der bruges i produktion.



OS2kitos

Test og Deployment

Tre ud af de fire repoer i OS2kitos GitHub projektet indeholder en beskrivelse af, hvordan den respektive del af projektet køres. Den mest omfattende vejledning findes i "kitos_frontend"-repoet, som indeholder den nye frontend til OS2kitos.

I "kitos"-repoet beskrives den nye frontend som værende under udvikling og er derfor ikke den aktuelle version af frontenden for OS2kitos. Dog nævnes dette ikke i "kitos_frontend"-repoet, hvilket skaber usikkerhed om systemets faktiske tilstand.

"kitos_frontend" kan bygges og startes ved hjælp af yarn. Ved at køre "start"-kommandoen startes en development server, som forbinder til projektets API. Derudover beskrives det, hvordan man kører en development server med en lokal backend, men det kræver, at en backend allerede er oppe og køre lokalt. Der findes imidlertid ingen vejledning til, hvordan backenden køres.

Tredjepart vejledes også i, hvordan frontend-projektet kan bygges til både udviklings- og produktionsmiljøer.

"kitos"-repoet, som indeholder både backenden og den nuværende frontend af projektet, mangler en vejledning til, hvordan projektet bygges eller køres. Der findes hverken vejledning til demo, udvikling eller produktion. Dette gør det nærmest umuligt for nye udviklere at få projektet op at køre uden hjælp fra leverandøren.

Struktur og Dokumentation

OS2kitos-projektets tekniske dokumentation mangler flere dele. Den eneste dokumentation findes i de enkelte repoers README-fil, hvor der som regel kun gives en overordnet beskrivelse af de enkelte teknologier. Dog gives tredjepart en vejledning i at køre dele af projektet

Der findes heller ikke dokumentation om brug af systemet, hverken på OS2's hjemmeside eller på projektets GitHub. Nogle vejledninger kan dog findes indirekte ved først at navigere til projektets egen hjemmeside og derefter opsøge et filarkiv. Ifølge tidsstemplerne i filarkivet er disse vejledninger ikke blevet opdateret i flere måneder. Da dokumentationen opdateres separat fra kildekoden og ikke er tilgængelig i projektets repoer, er det usikkert, om de stadig er relevant for den nuværende version af projektet.

Desuden mangler der vejledning i, hvordan backend og den nuværende frontend bygges eller køres. Dette gør det svært for tredjepart at bidrage, undersøge eller anvende løsningen

Selvom der er angivet en udviklingshistorik, er der stadig mangel på dokumentation af de ændringer, der er foretaget. Der eksisterer ikke en changelog eller lignende i projektets repo, som kunne give en beskrivelse af, hvad de enkelte ændringer er, eller hvad de bevirker.

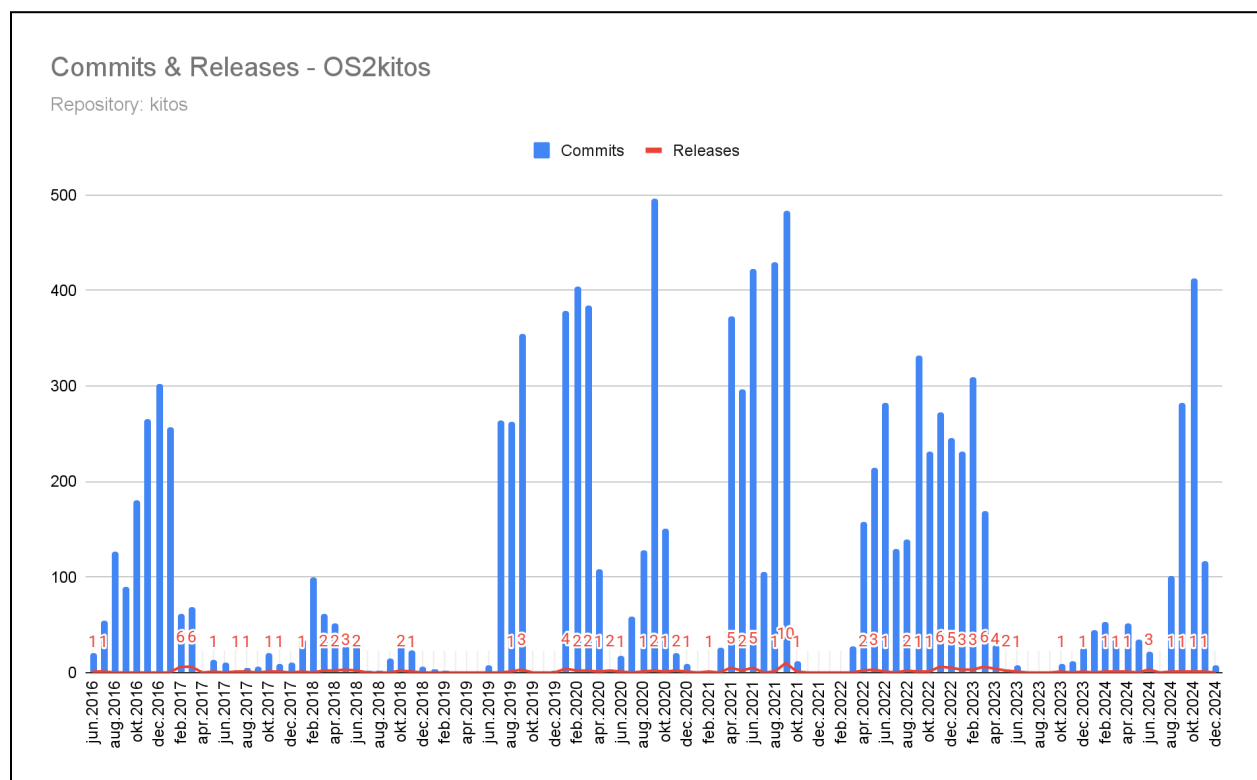
Afhængigheder

Der lader ikke til at være afhængigheder til leverandøren eller lukket software i projektet.

GitHub Commits, Releases og Versionering

Siden februar 2014 er der foretaget 14.482 commits til OS2kitos' "kitos"-repo. Disse commits er lavet med korte mellemrum og indeholder mindre kodeændringer, hvor hver ændring repræsenterer enten en feature eller et bugfix. Denne struktur sikrer en overskuelig udviklingshistorik og gør det nemt at få indblik i de specifikke ændringer, der er blevet foretaget over tid.

Derudover er der 98 releases på "kitos"-repoet, hvilket yderligere understøtter en gennemsuelig udviklingshistorik. Løsningen ses dog at have flere releases end dem, der er angivet som GitHub releases. Ser man på projektets tags, er der oplyst 132 releases. Dette giver dog stadig ikke det fulde billede af løsningens releases, da det tidligste release tag, der kan findes, er fra version 1.3.0.



Note: For diagrammets overskuelighed vises data fra første release og fremefter. Commits foretaget fra februar 2014 til maj 2016 kan derfor ikke ses på diagrammet.

OS2korrespondance

Det lader til, at OS2korrespondance er i det tidlige stadie. Det kan ses på GitHub, at initial commit blev udgivet i september 2024.

Test og deployment

Projektet indeholder en docker-compose.yml fil, der udelukkende opretter en MariaDB database. Ingen yderligere information om kørsel eller deployment af projektet er til stede.

Struktur og Dokumentation

Projektet indeholder ingen dokumentation, der giver indblik i at forstå, opsætte eller deploye systemet.

Projektet i GitHub indeholder filer som README, CONTRIBUTING, CHANGELOG mm. Disse filer har imidlertid ikke nogen relevant information om OS2korrespondance, da de kun er skabeloner for den struktur, som repoet er genereret ud fra, og de viser derfor kun, hvad filerne *burde* indeholde.

Afhængigheder

OS2korrespondance projektet gør brug af en Inspinia-webjar, som er et licens-baseret bootstrap tema. Der skal derfor betales licens for at kunne køre projektet. Brugen af dette tema er ikke oplyst, hverken i README filen eller andet dokumentation. Der er derudover ikke en licens for temaet tilgængelig, og det er derfor ikke muligt at finde information om licensbetingelserne.

GitHub Commits, Releases og Versionering

OS2korrespondance repoet på GitHub har kun 3 commits siden september 2024, hvor 1 commit har titlen "First public release". Der er ingen tags på repoet. Det første commit - release commitet - indeholder store ændringer til kodebasen. Dette commit blev skubbet i december 2024, og repoet er ikke opdateret siden.

OS2kravmotor

Test og Deployment

Der er ingen lettilgængelig vejledning til at køre systemet, da dokumentationen er gemt som binære filer, og ingen information gives i README. Der gives derudover ingen information, der vejleder i opsætning af miljøer, hverken for udvikling, demo eller produktion.

Downloader man løsningsbeskrivelsen fra GitHub, kan man læse, at Kravmotoren designes til at blive afviklet fra et Docker image til produktion. Dette er dog ikke nævnt i anden dokumentation. Der er ingen tegn på, at Docker er inkorporeret i, hvad der er tilgængeligt i kodebasen, hverken for deployment eller bygning af projektet. Det har ikke været muligt at finde noget Docker image tilgængeligt på Docker Hub. Der er kun en vejledning i at køre projektet gennem udviklerens IDE (Hvis der bruges IntelliJ eller Eclipse), og hvordan de to sub-moduler (ui og webjar) compiles.

Struktur og Dokumentation

Den eksisterende README-fil befinder sig i projektets rod, men er minimal og tilbyder kun en kort introduktion til projektet samt oplysninger om licensbetingelser. Hvis løsningen anvender licensbaseret software, forventes det også, at der oplyses, om denne software er nødvendig for projektet, hvordan den kan erstattes med ikke-licensbaseret software, og om det overhovedet er muligt at udskifte den. Sådanne oplysninger fremgår ikke af projektets README-fil. Uden denne information begrænses tredjeparters mulighed for at bruge, undersøge, ændre eller videregive løsningen.

Projektet mangler også en CONTRIBUTING.md-fil eller anden form for dokumentation, som kunne vejlede potentielle eksterne udviklere i, hvordan de kan bidrage til projektet.

En governance-rapport i repoet linker til en ekstern side, som oprindeligt skulle indeholde historik om lovoverholdelse og udbudsregler. Linket fører imidlertid til en 404-side, hvilket gør disse oplysninger utilgængelige.

Changelogs er ikke tilgængelige i repoet eller på projektets hjemmeside.

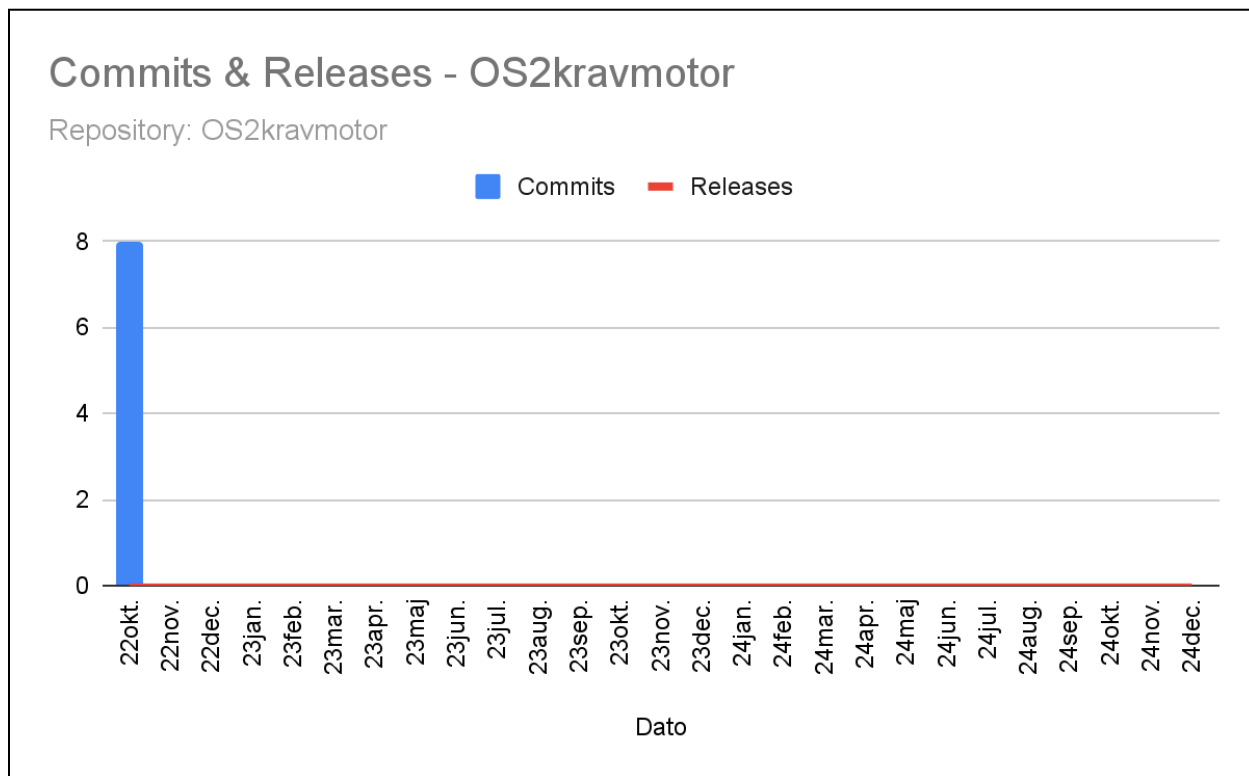
Afhængigheder

OS2kravmotor er afhængig af lukket software. Projektet bruger et licensbaseret Twitter Bootstrap-tema (Angle), som kræver en betalt licens. Der er købt en "Extended License" til OS2kravmotorens produktionsmiljø, hvilket vil sige, at temaet kun kan bruges i forbindelse med OS2kravmotor, og at der er en grænse for hvor mange gange produktet må distribueres ifølge WrapBootstraps hjemmeside.⁸ Denne afhængighed begrænser fleksibiliteten og gør det vanskeligt for eksterne udviklere at arbejde med projektet.

⁸ <https://support.wrapbootstrap.com/help/usage-licenses>

GitHub Commits, Releases og Versionering

Udviklingsaktiviteten i OS2kravmotor-repositoriet er begrænset. Alle 8 commits, som projektet har set, er foretaget i november 2022. Det har heller ikke været muligt at identificere nogle releases i repoet, hvilket gør det vanskeligt at få overblik over projektets versioner og generelle historik.



Der er uoverensstemmelser mellem de offentlige commits på GitHub og de senest registrerede opdateringer på OS2kravmotors issue tracker. For eksempel er det sidste commit dateret november 2022, mens et issue i Jira blev løst i oktober 2023. Denne forskel indikerer, at der kan gå meget lang tid, før ændringer til kildekoden bliver udgivet offentligt, hvilket skaber usikkerhed omkring projektets faktiske status.

OS2mo

Test og Deployment

OS2mo-projektet gør brug af Docker til deployment, hvilket fremgår af en Dockerfil placeret i projektets rod. Derudover findes en docker-mappe, som indeholder filer relateret til projektets Docker-image. For at sikre en nem opstart af projektet er der en docker-compose.yml-fil tilgængelig, hvilket gør det enkelt for tredjepart at køre projektet op.

Der findes desuden en guide til, hvordan projektet startes via docker-compose. Projektet indeholder tests og benytter GitLab CI til at opbygge sin CI/CD-pipeline, hvilket kan ses via en .gitlab-ci.yml-fil i projektets rod.

Struktur og Dokumentation

OS2mo projektet i github har 43 underliggende repositorier. Tages der udgangspunkt i os2mo repoet, er det veldokumenteret, og dokumentationen er let tilgængelig i projektets README fil. Der gives derudover en quick guide i at køre projektet op til et dev-miljø, så eventuelle bidragsydere har nemt ved at bygge systemet.

I OS2mo-projektet linkes der til en hjemmeside i README filen samt NEWS.md filen, hvor opdaterede changelogs er beskrevet.

Afhængigheder

Der lader ikke til at være afhængigheder til leverandøren eller lukket software i projektet.

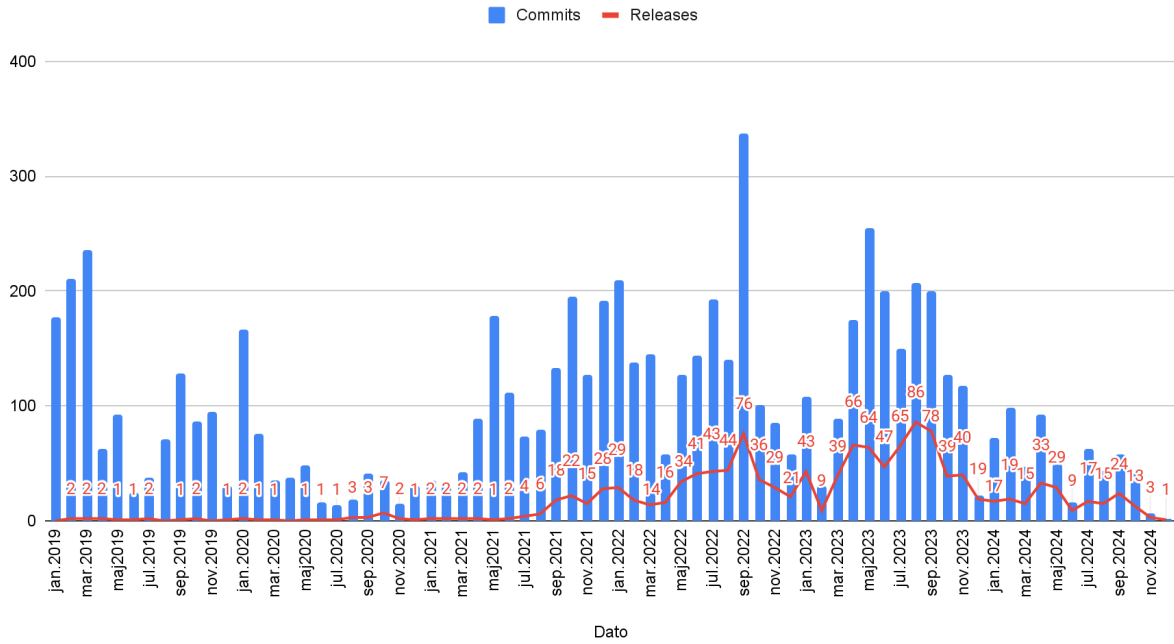
GitHub Commits, Releases og Versionering

os2mo repoet har i alt 9.997 commits siden maj 2017. Commits skubbet til GitHub hyppigt, og de indeholder hver især mindre ændringer til kodebasen.

Der findes 967 release tags på repoet, og det er derfor muligt at følge releasehistorikken gennem projektets levetid. For at sikre, at der ikke er skubbet releases uden tilsvarende commits i GitHub repoet, er repoet os2mo på Docker Hub også blevet gennemgået for releases. Resultatet kan ses i diagrammet nedenfor.

Commits & Releases - OS2mo

Repository: os2mo



OS2nectar

Test og Deployment

Det har ikke været muligt at finde information om, hvordan projektet bygges, køres, deployes eller downloades.

Struktur og Dokumentation

Der er ingen dokumentation tilgængelig for løsningen, hverken på et GitHub projekt, på hjemmesider eller på andre platforme.

Afhængigheder

Da der ikke er muligt at finde noget information eller kildekode om projektet, er det ikke muligt at undersøge projektets relation til lukket software.

Der er dog en stærk afhængighed af leverandøren, da ingen information eller kildekode er offentlig tilgængelig. Det er derfor kun muligt at bruge løsningen ved at købe sig ind gennem leverandøren.

GitHub Commits, Releases og Versionering

Det har ikke været muligt at finde et GitHub projekt eller lignende for løsningen.

Øvrige noter

Det har ikke været muligt at undersøge oplysninger om commits, releases eller kildekoden generelt, da OS2nectar tilsyneladende ikke er offentligt tilgængeligt. Henvisningen til OS2nectars kildekode på OS2's hjemmeside fører i stedet til GitHub-projektet for OS2borgerPC, som er en helt anden løsning.

Forsøg på at fremsøge projektet har ikke givet nogen resultater, hvilket tyder på, at der ikke findes en offentlig version af kildekoden. Hvis kildekoden ikke er offentligt tilgængelig, har tredjepart ingen mulighed for at ændre, undersøge, teste eller videregive projektet, hvilket ellers forventes af open source software.

Derudover findes der ingen information om produktet ud over en generel beskrivelse af dets funktionalitet på OS2's hjemmeside. Manglen på adgang til kildekode, licens og dokumentation betyder, at OS2nectar ikke kan betragtes som en open source-løsning, da det fundamentale princip om fri adgang til koden ikke er opfyldt.

OS2rollekatalog

Test og deployment

Der står i dokumentationen, at projektet kan køres ved brug af Docker Compose. Der er imidlertid ikke en docker-compose.yml fil for projektet tilgængelig, og der findes ingen dokumentation, der anviser, hvordan man konkret skal gøre som udvikler. Der er kun en skabelon på, hvilke miljøvariabler og mounted volumes docker-compose.yml filen kan indeholde under "Docker" dokumentationen i "Online Docs" mappen, men dette betyder stadig, at man selv skal oprette denne fil, før projektet kan køres op med docker. Dette er med til at besværliggøre kørslen af projektet.

Struktur og Dokumentation

Projektet mangler en detaljeret README-fil, som kan introducere nye udviklere til projektets funktionalitet, opsætning og brug. Den eksisterende README-fil tilbyder kun en kort beskrivelse af OS2rollekatalog og henviser kun til dele af dokumentationen gemt som binære .docx-filer i en "doc"-mappe.

Mange af henvisningerne i governance-rapporten ("OS2rollekatalog_GOVERNANCE_REPORT.md") fører til enten 404-fejl eller de tidligere nævnte binære filer. Dette skader gennemsigtigheden og forhindrer adgang til vigtig information.

Afhængigheder

OS2rollekatalog er afhængig af closed source software. For det første kræver projektet, ifølge dokumentationen, at der bruges Microsoft ADFS-servere, hvilket begrænser integrationen med open source-alternativer. Desuden benytter projektet et licensbaseret Twitter Bootstrap-tema kaldet "Angle." Temaet inkluderes via en .jar-fil ("angle-webjar-3.7.5.jar"), som ikke kan læses grundet dets binære format. I en af projektets mapper - "ui" mappen - ser man, at der bruges endnu et licensbaseret Bootstrap tema "Inspinia". "inspinia-webjar-lib" mappen indeholder ingen licens-fil, og det er derfor ikke muligt at finde information om, hvorvidt en licens er frikøbt til produktet - eller omfanget af denne licens.

Brugen af Bootstrap-temaerne er ikke nævnt i README-filen, hvilket gør det svært for tredjepart at kende til projektets afhængigheder.

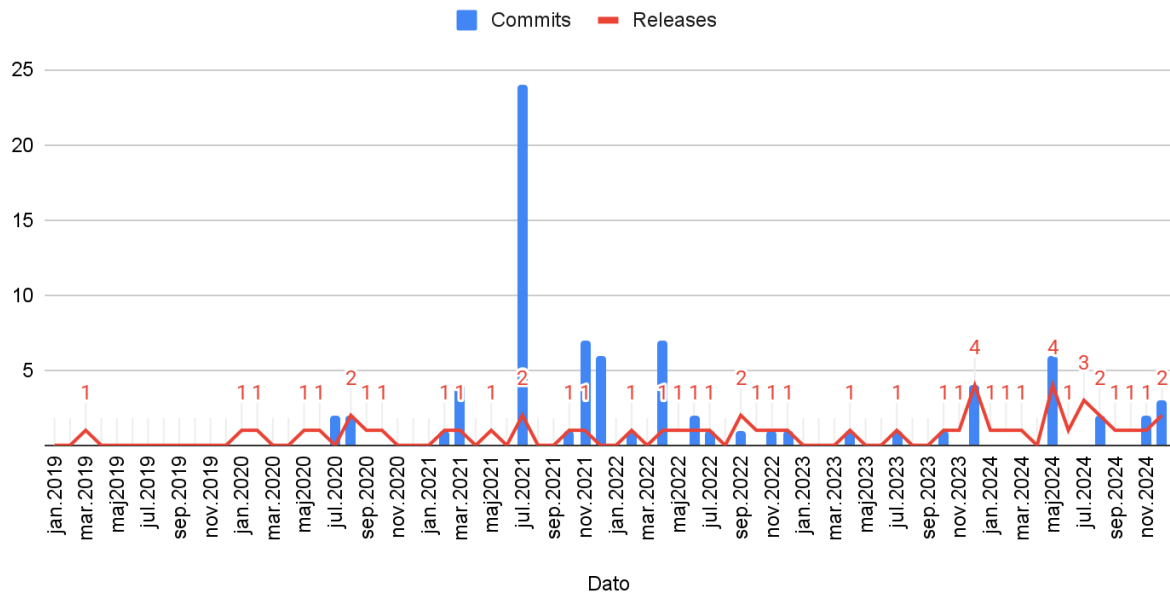
GitHub Commits, Releases og Versionering

Projektets GitHub-repository har kun 82 commits siden juli 2020. Main-branchen opdateres med jævne mellemrum og kun med større kode dumps. Der ses ikke nogen udviklingsbranches eller tilsvarende i repoet, som eller kunne forklare denne mangel på aktivitet. Dette understøtter antagelsen om, at det offentlige repository kun opdateres ved releases.

Der er ingen releases tags - kun commits med titler som "release <år> r<1-4>". Ved en gennemgang af projektets tilsvarende Docker Hub repo er det tydeligt, at der er blevet udgivet nogle releases, som ikke har tilsvarende commits i GitHub repoet.

Commits & Releases - OS2rollekatalog

Repository: OS2rollekatalog



Øvrige Noter

Projektet omtales som en løsning til brugerstyring på tværs af kommunens systemer, men det er ikke nævnt, at løsningen forudsætter, at der bruges closed source software (Microsoft ADFS), og at der derfor ikke kan bruges alternative open source identity providers.

OS2-SkadesØkonomi

Test og Deployment

Der er udarbejdet video-vejledninger til installation af produktet som et plugin til QGIS. Installationen kræver, at tredjepart downloader to ZIP-filer fra "installation"-repoet. Opsætningen er primært rettet mod demo-, test- og produktionsmiljøer, men ikke et udviklingsmiljø. Videovejledningerne dækker derfor ikke, hvordan man opsætter et miljø til udvikling.

Ved at undersøge mapperne i "Udvikling"-repoet kan man under "OS2DamageCost" finde en README-fil, der lykønsker brugeren med at have bygget et plugin til QGIS. Filen indeholder vejledning i, hvordan resource-filen kompileres og deployes ved hjælp af GNU make, samt hvordan plugin'et kan testes.

Filen indeholder imidlertid ikke en vejledning til, hvordan pluginet bygges, hvilket betyder, at tredjepart selv skal finde frem til, hvordan dette gøres for at kunne arbejde med og videreudvikle løsningen.

Struktur og Dokumentation

Dokumentationen i OS2-SkadesØkonomi projektet er vanskelig at få overblik over. I "Dokumentation-og-vejledninger"-repoet findes der skriftlige brugervejledninger om, hvordan løsningen bruges, en præsentation om prissætning af oversvømmelsesskader samt videovejledninger, der guider brugere i at installere og opsætte løsningen som et plugin i QGIS.

Generelt er måden, hvorpå dokumentationen er uploadet til GitHub, vanskelig at få overblik over. Det er både vanskeligt at finde frem til den relevante dokumentation og at gennemskue, hvilken information der faktisk er nødvendig.

Eksempelvis findes der en vejledning i "Installation"-repoets README-fil, som er forældet. Vejledningen angiver selv, at den ikke længere er gældende. Den opdaterede vejledning findes i dokumentations-repoet, men eftersom README-filen i installations-repoet ikke indeholder anden relevant information, bliver den overflødig, og den bidrager ikke til en bedre forståelse af repoets indhold.

Overblikket kompliceres yderligere af, at al relevant dokumentation ikke kan findes i dokumentations-repoet. For eksempel findes vejledning om kompilering og deployment i en README.html-fil i den underliggende "OS2DamageCost"-mappe i det separate "Udvikling"-repo. Denne vejledning er mangelfuld, da den kun beskriver processen, efter at projektet er bygget, men ikke vejleder brugeren i den samlede opsætningsproces.

Derudover findes der en anden README-fil i samme mappe, som giver en kortere, men lidt anderledes vejledning om de samme emner som i den første README-fil. Dette gør det vanskeligt at gennemskue, hvilke filer der faktisk er relevante.

Når det ikke tydeligt fremgår, hvor al relevant dokumentation findes, og tredjepart er nødt til at lede i forskellige repoer og mapper for at samle nødvendig information, skaber det usikkerhed omkring, hvorvidt man har fået det fulde overblik over projektet, eller om der stadig er vigtig dokumentation, som endnu ikke er fundet.

Der er heller ingen logs, der kunne indikere, hvilke ændringer der er foretaget af kildekoden. Det er derfor også vanskeligt at få et overblik over udviklingshistorikken.

Afhængigheder

Der lader ikke umiddelbart til at være afhængigheder til leverandøren eller lukket software i projektet, men pga. projektets tilgang til releases og versionering er det vanskeligt at vurdere, om koden på GitHub er den samme som i produktion.

GitHub Commits, Releases og Versionering

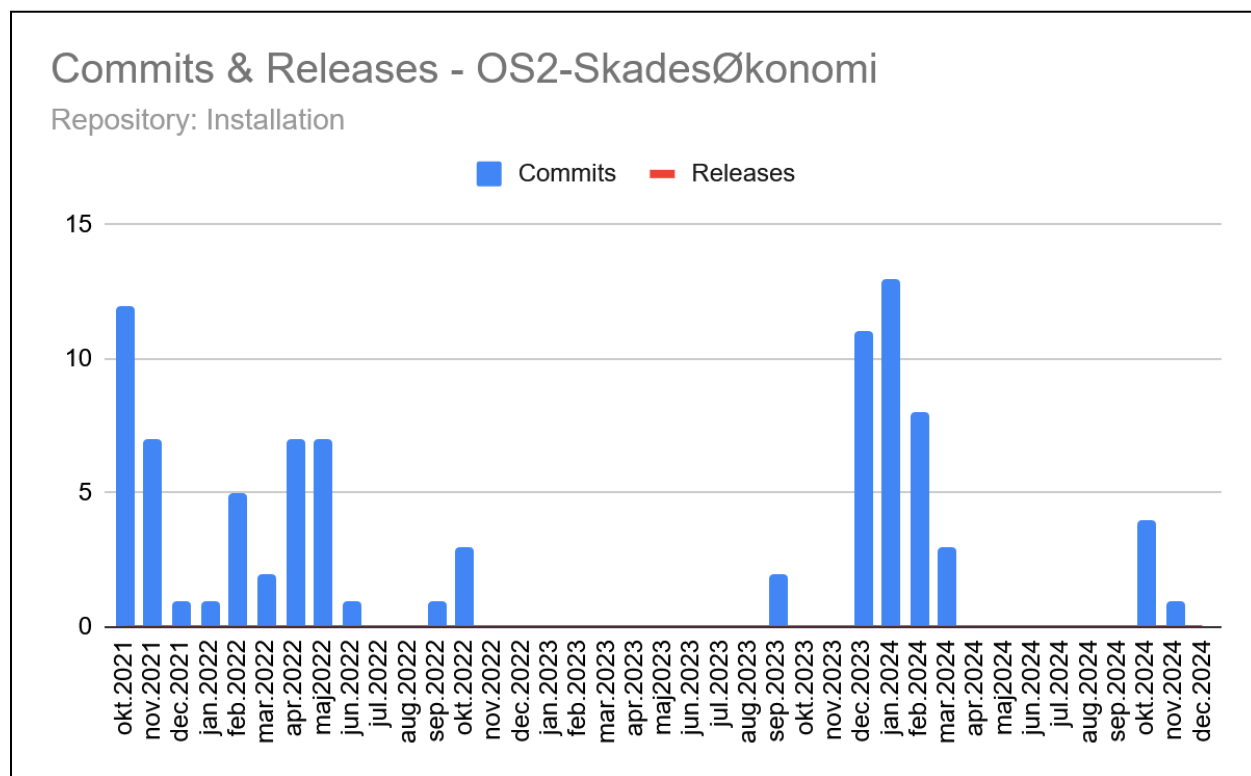
Da projektet køres ved at downloade zip-filer fra løsningens GitHub, er det en udfordring at finde frem til den faktiske version af projektet, der installeres. Zip-filerne er binære, og derfor kan man ikke se, hvilke ændringer der er foretaget i filerne gennem GitHubs brugerflade. I commits lavet til "Installation"-repoet ses primært opdateringer af "OS2DamageCost.zip"-filen, og det er dermed ikke muligt at se, hvilke ændringer der faktisk er foretaget i filen.

For at få indsigt i, hvilke ændringer der er blevet foretaget i "OS2DamageCost.zip", er det nødvendigt at kigge i commits i "Udvikling"-repoet, hvor det er muligt at se ændringerne, som er foretaget i "OS2DamageCost"-mappen.

Den eneste måde at finde ud af, om kildekoden i zip-filen, der downloades fra "Installation"-repoet, stemmer overens med de ændringer, der er lavet i "Udvikling"-repoet, er ved at gennemgå de enkelte filer i zip-filen, efter at den er downloadet. Derefter skal man krydsreferere dem med commits i "Udvikling"-repoet. Denne proces er tidskrævende, og den fordrer en stor indsats - og et stort overblik - hvis man skal være sikker på, at man ikke gør sig afhængig af kode, der endnu ikke er udgivet på GitHub – altså lukket kode.

OS2-SkadesØkonomi-projektets "Installation"-repo har haft 144 commits siden oktober 2021. Den gennemsnitlige størrelse af disse commits er svær at vurdere, da de er foretaget i en zip-fil og derfor ikke kan ses direkte. Når man kigger i commits, der er lavet i "OS2DamageCost"-mappen i "Udvikling"-repoet, kan man dog se, at de indeholder mindre kodeændringer.

Der findes ingen releases, release-tags eller logs, der kan give en oversigt over, hvilke ændringer der er foretaget mellem opdateringerne. Hvis man vil have et overblik over ændringerne, er man nødt til at gennemgå de individuelle commits, som ikke er lavet i en zip-fil, for de respektive repoer.



OS2skoledata

Test og Deployment

OS2skoledata mangler dokumentation for, hvordan udviklings-, demo- og produktionsmiljøerne opsættes. De eneste relevante filer, der kunne give denne vejledning, findes i projektets "doc"-mappe som binære .docx-filer. Disse filer hedder "ADSync opsætning" og "Google Workspace opsætning", men indeholder ingen information om opsætning af udviklingsmiljøer, deployment-processen eller CI-integration. Ingen test-filer eller mapper kunne findes i projektet. Der ses ingen CI-konfiguration i projektet, hvilket tyder på, at der ikke anvendes en automatiseret test- eller deploymentproces.

Struktur og Dokumentation

Projektets README.md er helt tom bortset fra titlen "OS2skoledata". Dokumentationen er begrænset til docx-filer, hvilket gør indholdet utilgængeligt, med mindre filerne downloades. Der er ingen CONTRIBUTING.md-fil, og der nævnes heller ingen vejledning til, hvordan udviklere kan bidrage til projektet.

Der findes en changelog-fil i en undermappe (`os2skoledata-ad-sync/OS2skoledataADSync-changelog.txt`), men dens placering gør den svær at finde. Denne fil er summarisk og indeholder kun en dato, et versionsnummer og én sætning til at beskrive hver version.

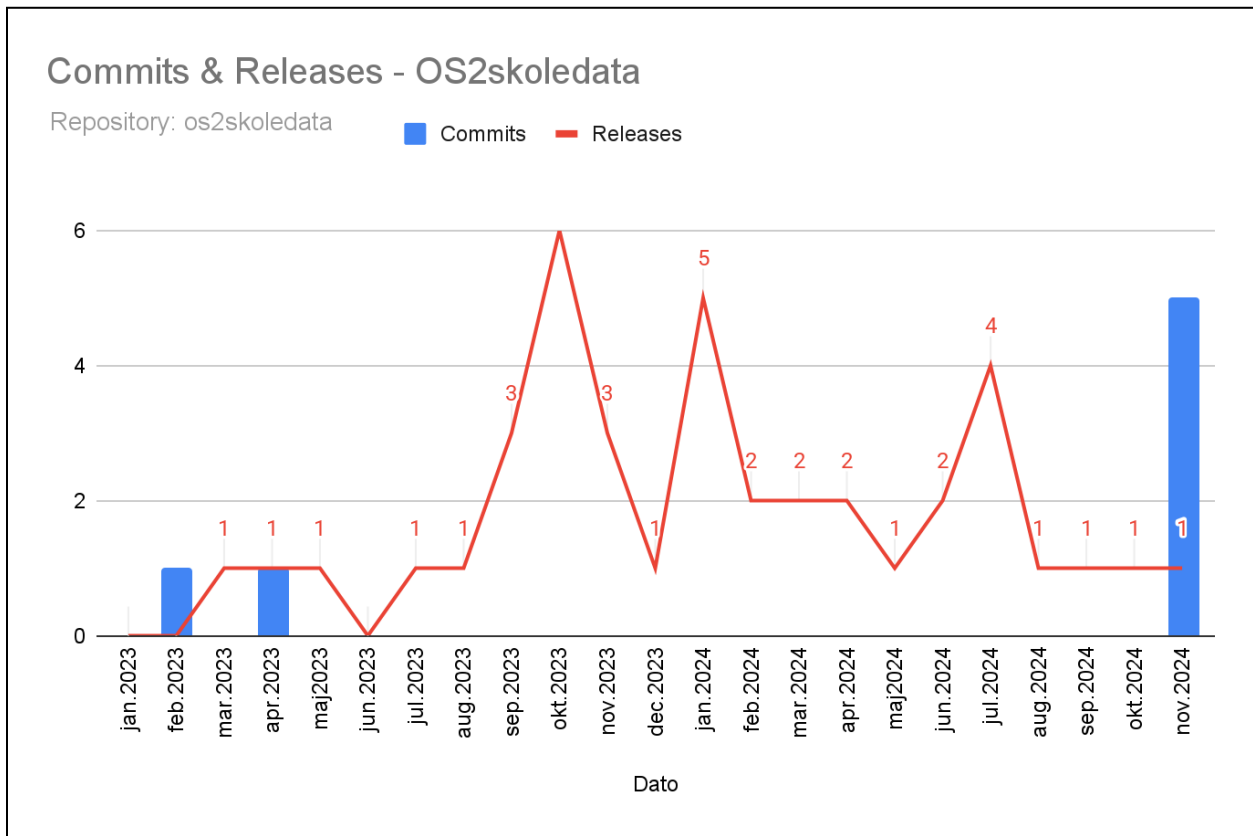
Afhængigheder

Der lader ikke til at være afhængigheder til leverandøren eller lukket software i projektet.

GitHub Commits, Releases og Versionering

Siden projektets første commit den 27. februar 2023 er der kun sket syv commits i alt. Det seneste commit blev skubbet til GitHub den 21. november 2024. De få commits, der er skubbet til GitHub repoet, har indeholdt større ændringer til kodebasen. Men det er uklart, hvad de enkelte commits indeholder af ændringer.

På <https://www.os2skoledata.dk/download> kan man finde frem til en changelog fil til OS2skoledata-ad-sync. Dette kan også findes som mappe i OS2skoledata GitHub repoet. Ved at kigge på de versioner i changelog filen, samt hvornår de er udgivet, kan man se, at projektet har nye versioner oftere, end der committes til GitHub repoet.



Diagrammet ovenfor viser, at der er oplagte uoverensstemmelser mellem versionerne fra changelogen i GitHub og OS2skoledata hjemmeside:

- **Seneste versioner på GitHub** viser kun op til version **1.16.2** med den nyeste ændring fra oktober 2024.
- **OS2skoledata hjemmesiden** viser, at systemet som minimum er oppe på version **1.18.2** - med ændringer foretaget i januar 2025.

Disse uoverensstemmelser viser, at det offentlige repository på GitHub ikke afspejler den faktiske udviklingsstatus for projektet. Dette understøttes yderligere af de kodedumps, der ses per commit på GitHub repoet.

Versioneringsmønsteret er heller ikke konsistent. Eksempelvis er changelog-formatet (fx "2024.10.04 - 1.16.2") anderledes end GitHub-commit-formatet (fx "ad sync 2024r1"), hvilket gør det vanskeligt at forstå, hvordan de forskellige versioner hænger sammen.

OS2sofd

OS2sofd projektet på GitHub er opdelt i 4 underliggende repositorier. I denne gennemgang tages der hovedsageligt udgangspunkt i os2sofd repositoriet, da det anses som projektets kerne repo og har haft mest aktivitet.

Test og Deployment

OS2sofd-projektet mangler vejledning og værktøjer til opsætning af udviklings- og prod-miljøer. Der er ingen synlig deployment-guide eller scripts til at lette opsætningen af en instans. Dette gør det vanskeligt for eksterne udviklere at afprøve eller evaluere systemet.

Derudover findes der ingen Dockerfile eller lignende i projektet, og det er derfor ikke muligt for tredjepart at bygge OS2sofd-projektet.

Struktur og Dokumentation

Projektet lider af en generel mangel på dokumentation. Der er ingen README.md-fil i roden, hvilket betyder, at der ikke findes nogen information om projektets formål, funktionalitet, eller hvordan man bidrager.

Der findes en del dokumentation i en "doc"-mappe, hvor størstedelen er i binært .docx-format. Noget dokumentation er også suppleret som PDF, men dette er ikke standarden. Dokumentationens format gør den mindre tilgængelig og besværlig at navigere i.

En README.txt-fil kan findes i stien `os2sofd/doc/OPUS/README.txt`, men den indeholder kun en sti til ét endpoint og en kort note om, hvor der findes et certifikat. Der gives ingen yderligere information eller kontekst relateret til endpointet.

Der er ingen dokumentation, der beskriver, hvordan eksterne udviklere kan bidrage til projektet. Samlet set svækker dette gennemsigtigheden og forringer mulighederne for samarbejde med eksterne udviklere.

Afhængigheder

Projektet afhænger af et licensbaseret bootstrap-tema, kaldet "Angle". Dette tema kræver køb af en licens, men denne afhængighed nævnes ikke i nogen dokumentation. Den medfølgende licensfil i mappen `webjar` giver kun et link til temaets hjemmeside. Det er derfor ikke muligt at finde information om licensbetingelser, hvilket både svækker projektets transparens og tredjeparts mulighed for at bruge, undersøge og ændre i løsningen.

Derudover skabes der en afhængighed til leverandøren, når løsningen ikke kan bygges eller køres som følge af projektets manglende Dockerfile. Tredjepart gøres derfor afhængig af leverandøren for at kunne bruge løsningen.

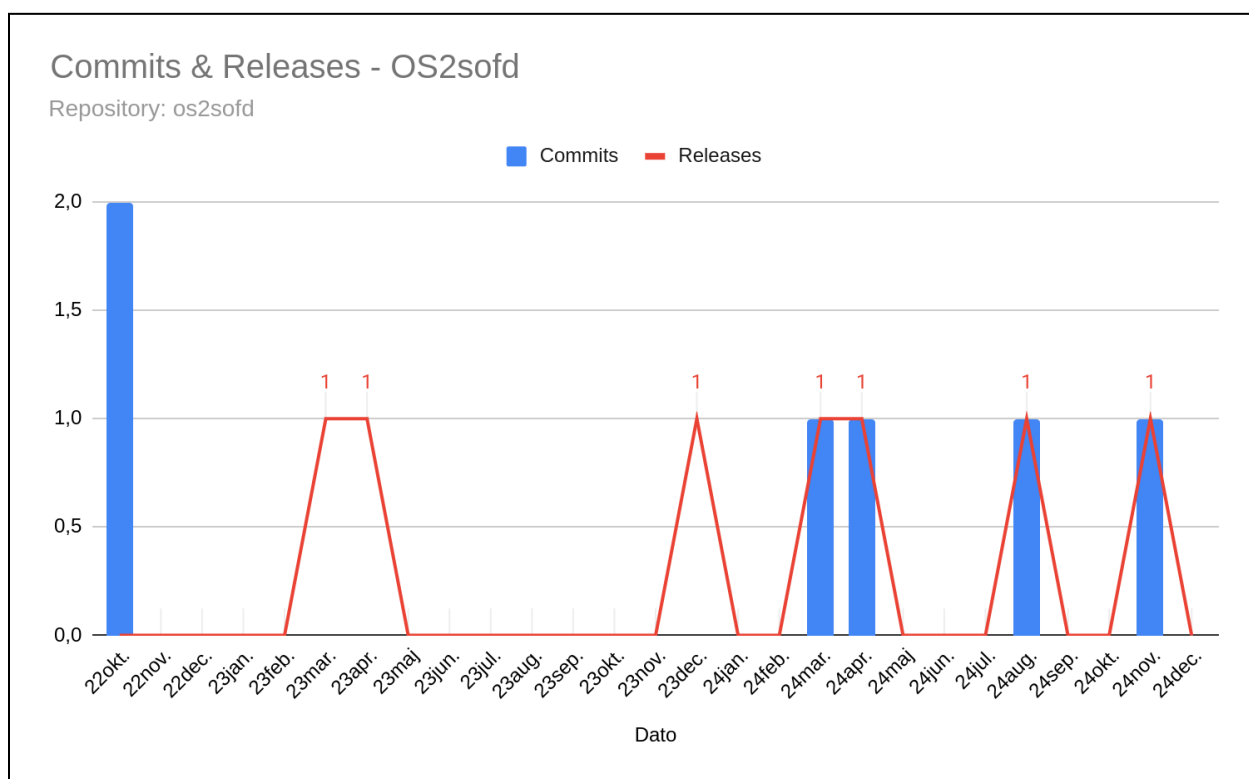
GitHub Commits, Releases og Versionering

Repoet "os2sofd" har kun haft seks commits siden projektets første commit den 14. oktober

2022. Commit-historikken er minimal og består hovedsageligt af commits med titlen “release 2024 r<1-3>”. Disse commits forekommer med mellemrum på flere måneder, og de indeholder større ændringer til kodebasen.

Changelogs findes heller ikke i repositoryet. I stedet skal man navigere til forskellige stier i et andet "agents"-repo for at finde changelog-filer til de individuelle “agents”. Kigger man på produktets hjemmeside, kan der under dokumentation findes en “ændringslog”, som viser de ændringer, der er tilføjet mellem releases. Der er foretaget 6 releases. Disse releases er imidlertid ikke udgivet på samme dato, som de release commits der er i projektets GitHub repo. Ifølge “ændringsloggen” på projektets hjemmeside er der foretaget 3 releases i 2023, som ikke kan findes i GitHub repoet. Derudover er der foretaget et release i marts 2024 på projektets GitHub, som ikke er i oversigten i ændringsloggen på hjemmesiden.

For at få det fulde billede af projektets releases viser diagrammet releases fundet på både GitHub og projektets hjemmeside. Der er også forsøgt at finde oplysninger om releases på andre platforme, men det har ikke været muligt at finde flere informationer.



OS2sync

Test og Deployment

Man kan bruge Docker til deployment, står der i dokumentationen. Projektet indeholder imidlertid ikke en docker fil. Den eneste hjælp, man får, er et eksempel i installationsguiden.

Struktur og Dokumentation

OS2sync-projektets dokumentation mangler væsentlige dele, der gør det vanskeligt for eksterne udviklere at opsætte eller teste miljøer.

Installationsguiden i mappen "Documentation" dækker kun opsætning som Windows Service og Docker Container og angiver eksplicit, at der ikke tilbydes vejledning i at hente, kompilere eller integrere kildekoden. Dette overlades til brugeren selv, hvilket gør opsætningen uoverskuelig, selv hvis man er i besiddelse af en dybere teknisk forståelse.

Guiden forudsætter, at brugeren allerede er bekendt med Docker på sådan et niveau, at der hverken er opstillet Docker eller Docker Compose filer eller er givet en guide til opsætning og kørsel. Der tilbydes kun generelle retningslinjer for konfiguration og placering af certifikater og et eksempel på, hvordan en docker-compose.yml fil kunne se ud, hvilket yderligere begrænser tilgængeligheden for nye brugere og potentielle eksterne udviklere. Denne forudsætning begrundes dog ikke løsningens mangel på en Dockerfile, da opsætningen af denne fil ikke udelukkende kræver kendskab til Docker-syntaks, men også dybdegående viden om projektet selv. Tredjepart er derfor nødsaget til at bruge de docker images, der udgives på løsningens Docker Hub, og har derfor ikke selv mulighed for at ændre og bygge løsningen.

Repoet har en README.md-fil, men indholdet er meget begrænset. Der er kun én sætning om hvad OS2sync er samt et link til projektets hjemmeside. Dokumentationen på hjemmesiden er ikke blevet opdateret i takt med systemet. Vejledninger og guides henviser til henholdsvis versionerne 4.4.0, 4.1.0 og 4.0.0, men undersøger man projektets Docker Hub, kan der ses en version 4.6.4. Tredjeparter kan derfor ikke være sikker på, at dokumentationen afspejler den aktuelle version af systemet.

Den opdaterede installationsguide i mappen "Documentation" lider også af betydelige mangler. Der gives intet overblik over, hvordan man henter, bygger eller vedligeholder kildekoden, og der er ingen beskrivelser af, hvordan eksterne udviklere kan bidrage. En CONTRIBUTING.md-fil er ikke inkluderet, hvilket yderligere svækker gennemsigtigheden og muligheden for samarbejde.

Derudover er der ingen changelog-fil nogen steder i projektet. Ændringer mellem versioner fremgår ikke klart, medmindre man aktivt gennemgår hver ændring i de enkelte commits.

Afhængigheder

I OS2sync's DockerHub står det klart, at den version af løsningen, som er angivet dér, er måneder foran det, man kan se på det offentlige GitHub. Det betyder, at man kører projektet

blindt, hvis man kører systemet på den nyeste version udgivet på Docker, da man ikke kan tilgå kildekoden for denne version.

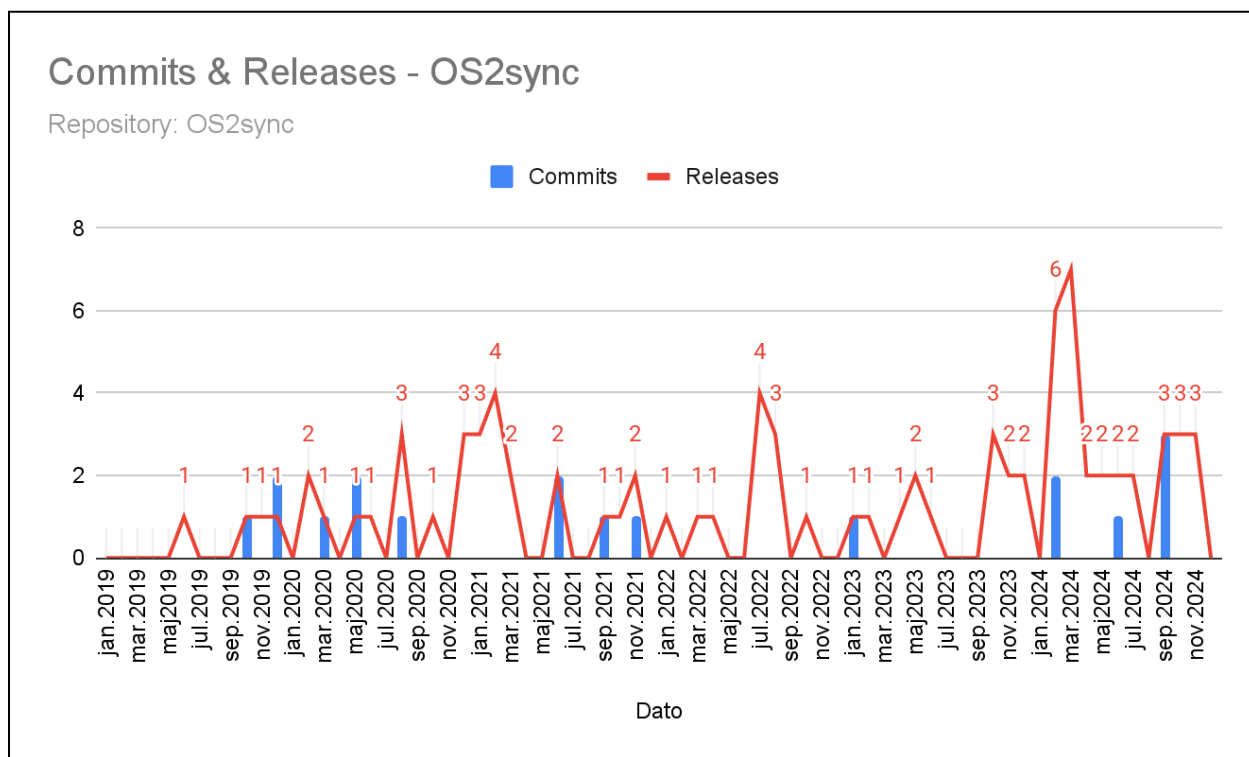
GitHub Commits, Releases og Versionering

OS2sync-repoet har kun haft 18 commits siden det første commit 23. oktober 2019.

Commit-historikken er præget af større spring i versionsnumre (f.eks. fra "release 2.2.0" til "release 2.5.2" og videre til "release 2.7.0"), hvilket tyder på, at ikke alle releases offentliggøres på repoet. Commits repræsenterer kun større milepæle eller releases og giver derfor ikke et overblik over projektets udviklingsproces, da der ikke gives nogen indikation på, hvilke ændringer der er foretaget.

Desuden er projektet ikke i sync med det, der udgives på Docker Hub. Mens det offentlige GitHub-repositories seneste version angiver 4.4.1, offentliggjort september 2024, er der på Docker Hub en version 4.6.4, som udkom januar 2025. Det offentlige GitHub repository afspejler således ikke den nyeste version af systemet og er angiveligt fem versioner bagud.

Ved at gennemgå tags på Docker Hub og krydse dem med de commits og releases, der skubbes til GitHub repoet, bliver den omfattende uoverensstemmelse og de tydelige kode dumps, der foretages på projektet, tydelig.



OS2udoglær

Test og Deployment

OS2udoglær-projektet tilbyder en detaljeret vejledning i, hvordan man opsætter et lokalt udviklingsmiljø ved hjælp af DDEV. Derudover gives der også instruktioner i, hvordan projektet kan installeres lokalt ved brug af en LAMP stack.

Vejledningen dækker de nødvendige trin til at konfigurere miljøet korrekt, hvilket gør det lettere for udviklere at komme i gang med projektet, uanset om de vælger at bruge container-baseret udvikling med DDEV eller en mere traditionel LAMP stack-installation.

Den eneste information, der gives om deployment, er i README-filen, hvor man informeres om, at dette gøres via "Platform.sh CI/CD og Gitops"

Der anvendes GitHub Actions til at bygge og teste projektet.

Projektet indeholder et "os2udoglaer-platform"-repo, der bruges som et centralt repo til projektets deployment pipeline, hvor både backend og frontend deploys.

Struktur og Dokumentation

Projektet indeholder ingen dedikerede dokumentationsmapper eller separate dokumentations-repositories. Al nødvendig dokumentation for opsætning, deployment og bidrag til projektet findes i dets README-filer.

README-filerne indeholder letforståelige og detaljerede vejledninger, der gør det nemt for tredjepart at få et hurtigt overblik over projektet og komme i gang med at arbejde med det. Dokumentationen er struktureret, så både nye og erfarne brugere hurtigt kan finde relevant information uden at skulle navigere gennem flere filer eller eksterne ressourcer.

Der er ingen changelogs eller andre dokumenterede versionslogs, som giver et overblik over ændringer mellem releases. For at få et indblik i, hvilke ændringer der er foretaget, skal man gennemgå projektets commits, hvor der er foretaget merges af release branches. Selvom det er muligt at spore ændringer på denne måde, er det ikke en proces, som giver tredjepart et hurtigt overblik på forskellene mellem versioner. Dog giver det et detaljeret overblik over commits foretaget under et release.

Afhængigheder

Der lader ikke til at være afhængigheder til leverandøren eller lukket software i projektet.

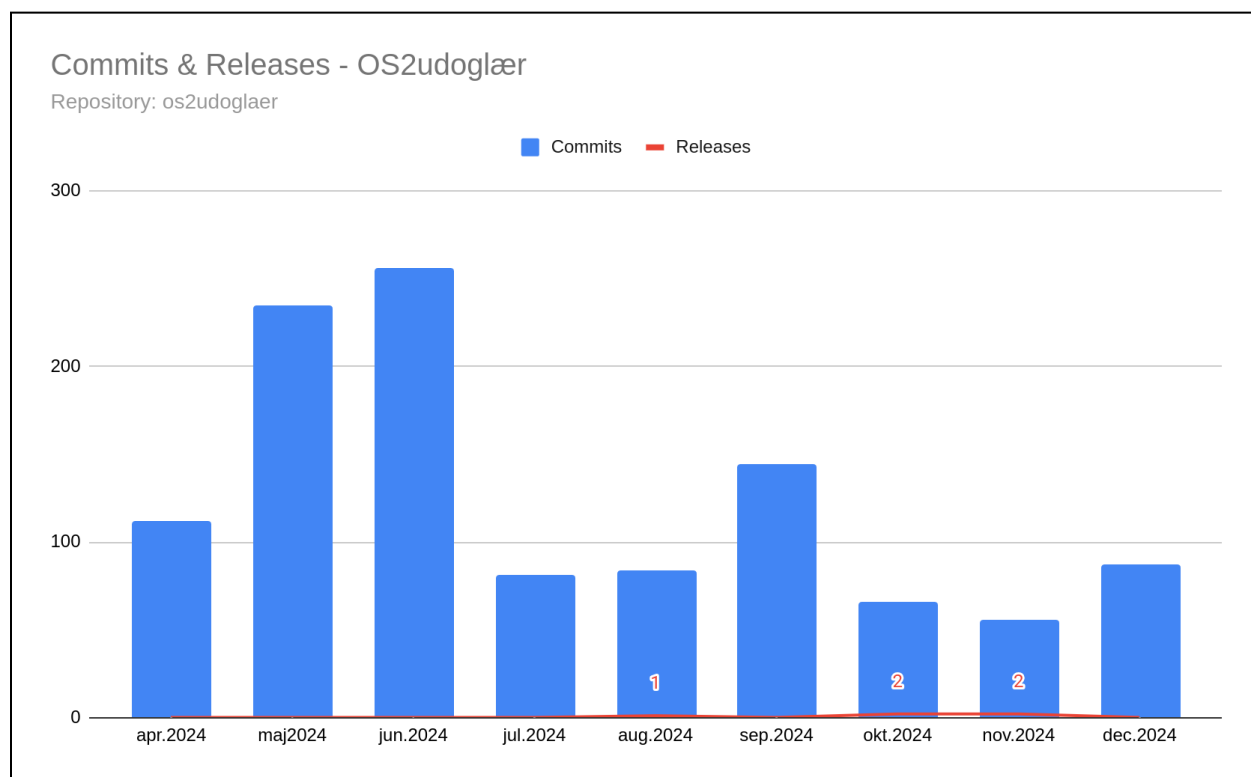
GitHub Commits, Releases og Versionering

Repoet, som har haft mest aktivitet i løbet af projektets levetid på GitHub, er "os2udoglaer". Siden april 2024 er der lavet 1.121 commits til dette repo. Commit-historikken viser, at commits typisk foretages med korte mellemrum og består af mindre kodeændringer.

Projektet anvender en struktureret Git-strategi, hvor features og bugfixes håndteres via separate branches. Branch-navne inkluderer ofte det issue, ændringen vedrører, hvilket gør det nemmere for tredjepart at få et indblik i, hvilke commits der relaterer sig til et issue, og hvornår de er blevet implementeret.

Udviklingshistorikken mangler imidlertid en central oversigt, da der ikke findes nogen changelogs, releases eller release tags. Den eneste måde at spore ændringer mellem versioner er ved at gennemgå release commits, hvor ændringer relateret til et specifikt release kan identificeres. Ved at gennemgå projektets commits har det været muligt at finde frem til 5 unikke releases.

Der er ingen platforme udover GitHub, hvor der kan findes oplysninger om løsningens versioner. Det er derfor ikke muligt at verificere, om de releases, der fremgår på GitHub, faktisk svarer til den version, der er i produktion.



OS2valghalla

Test og Deployment

I "OS2valghalla-3-documentation"-repoet findes en guide beskrevet til at klonе og bygge projektet. Hermed menes der ikke selve OS2valghalla-løsningen, men i stedet dokumentationsdelen af projektet. Løsningens dokumentation kan også tilgås på en ReadTheDocs hjemmeside, som kan findes under "OS2valghalla 3"-repoet.

I "OS2valghalla-3"-repoet findes en README-fil, der beskriver, hvordan systemet køres op. I modsætning til dokumentationsprojektets opsætningsvejledning, som er gjort let at køre op, er opsætningen af selve systemet mere teknisk krævende. Tredjepart skal foretage flere manuelle trin, før det er muligt at få et lokalt udviklingsmiljø op at køre.

Projektets dokumentation forklarer, at OS2valghalla pakkes som en Docker-container, der består af 7 images. 4 af disse images er udviklet specifikt til OS2valghalla. Det har imidlertid ikke været muligt at finde nogle officielle images på Docker Hub.

Der findes en guide til, hvordan udviklere kan tilføje og teste deres kodeændringer på en lokale maskine, men der mangler en guide til at opsætte systemet, der kunne gøre det overskueligt for nye udviklere at komme i gang med løsningen.

Struktur og Dokumentation

Dele af OS2valghallas dokumentation er udførlig og giver nyttige samt detaljerede informationer. Dog er der andre dele af løsningen som ikke beskrives i dokumentationen, og det giver dermed ikke tredjepart de nødvendige oplysninger for at kunne bygge den op.

Der findes en guide til udviklere, der ønsker at bidrage til projektet, hvor der forklares, hvilke trin der skal følges, inden dokumentationsopdateringer eller kodeændringer kan tilføjes som et pull request.

Brugen af projektets API mangler dokumentation. Der er ingen beskrivelse eller henvisning til API-endpoints, og det nævnes kun, at API'et står som modtager af al kommunikation fra frontend. Det betyder, at tredjepart, der ønsker at forstå eller udvide API'et, ikke har noget udgangspunkt for, hvordan det fungerer, eller hvilke endpoints der er tilgængelige.

Der findes heller ikke nogen changelogs eller anden form for versionshistorik, der kan give et overblik over udviklingen af projektet. For at forstå udviklingshistorikken er man nødt til at gennemgå commit-historikken på GitHub. Det er dog ikke altid tydeligt beskrevet i commit-beskederne, hvad omfanget af ændringen er.

Afhængigheder

Der lader ikke til at være afhængigheder af leverandøren eller lukket software i projektet.

GitHub Commits, Releases og Versionering

Der er blevet foretaget 107 commits til "OS2valghalla-3"-repoet siden september 2023. De fleste af disse commits blev imidlertid lavet i marts, april og november 2024, hvor de tre måneder tilsammen står for 77 af de 107 commits. Størrelsen på commits varierer en del, men commits indeholder typisk større kodeændringer. Eksempelvis er det kun i april, oktober og december 2024, at mængden af kodelinjer tilføjet i gennemsnit ligger under 100.

Repoet indeholder også 7 release tags, som giver tredjeparter mulighed for at få indsigt i de specifikke bugfixes og features, der er blevet implementeret i hver af de enkelte releases. Dette giver en forståelse af, hvilke ændringer der er blevet foretaget mellem versionerne, og det gør det lettere for tredjepart at få et indblik i projektets udviklingshistorik.

