

# Installations- og Driftsvejledning

## Indholdsfortegnelse

Installations- og Driftsvejledning .....	1
Indholdsfortegnelse .....	1
Versionshistorik .....	2
Baggrund .....	3
Krav til Driftsmiljø .....	3
Installation og Drift .....	4
Generelt .....	4
BIVWS – Bivirkningsindberetnings Web Service .....	4
Kort Beskrivelse .....	4
Netværk .....	4
Ressourcer .....	4
Environment Variable .....	4
Konfiguration .....	5
Whitelisting .....	6
Porte og Prober .....	6
Logging og metrikker .....	6
Junosender .....	7
Kort Beskrivelse .....	7
Ressourcer .....	7
Netværk .....	7
Konfiguration .....	7
Porte og Prober .....	8
Logging og metrikker .....	8
Database .....	8
Meldenbivirkning .....	10
Kort Beskrivelse .....	10
Netværk .....	10
Ressourcer .....	10
Konfiguration .....	10
Porte .....	10
Logging og metrikker .....	11
Client Service .....	11
Kort beskrivelse .....	11

Netværk .....	11
Ressourcer .....	11
Konfiguration .....	11
Porte og Prober.....	13
Logging og metrikker .....	13
R2R3 Converter.....	13
Kort beskrivelse .....	13
Netværk .....	13
Ressourcer .....	13
Konfiguration .....	13
Porte og Prober.....	14
Logging og metrikker .....	14
Documentation Viewer og Documentation Static.....	14
Kort beskrivelse .....	14
Netværk .....	14
Ressourcer .....	14
Konfiguration .....	14
Porte og Propes .....	17
Logging og metrikker .....	17
IP Begrænsning .....	18
Smoketest .....	18
Kort beskrivelse .....	18
Netværk .....	18
Ressourcer .....	18
Konfiguration .....	18

## Versionshistorik

Versi on	Dato	Ændring	Forfatter
0.0.1	19.04.2022	Initiel version Røde mærkninger ved udestående afklaringer	Jeppe Gravgaard
0.0.2	09.05.2022	Rettet fejl af store bogstaver i konfiguration	Jeppe Gravgaard

		Omskrevet mount beskrivelse	
		Tilføjet generel beskrivelse omkring brugere i docker-container	
0.0.3	10.05.2022	Rettelser på baggrund af Review fra Driftsleverandør	Jeppe Gravgaard
0.0.4	12.05.2022	Rettelser på baggrund af Review fra Driftsleverandør	Jeppe Gravgaard
0.0.5	16.05.2022	Tilføjelse af whitelistning konfiguration af BIVWS	Jeppe Gravgaard
0.0.6	16.05.2022	Tilføjelse af konfiguration af docviewer og docstatic	Jeppe Gravgaard
0.0.7	24.05.2022	Tilføjelse af paths for nginx container Tilføjelse af probe sti for nginx container	Jeppe Gravgaard

## Baggrund

Dette dokument giver en vejledning til installation samt drift af de enkelte services som udgør det samlede system "PVInddata". Formålet med dokumentet er, at en driftsleverandør med dokumentet i hånden kan lave en komplet installation af systemet samt have tilstrækkelig viden til at drifte systemet. Dette indebærer overordnet forståelse af systemet samt mere dybdegående viden om håndtering af konfiguration, logs, metrikker og certifikater.

Dokumentet er en del af en samlet dokumentationspakke, hvorfor der undervejs henvises til andre ressourcer for dybere systemforståelse. Dokumentet skal dog i sin funktion være selvindholdende.

## Krav til Driftsmiljø

Systemet er bygget ud fra 12FA- og cloud-native principperne og testet på en Kubernetes platform (Trifork Cloud Stack Dev-cluster), men da samtlige workloads er selvindeholdte stateless containere, kan systemet installeres på en hvilken som helst host med Docker installeret.

Driftsmiljøet skal opfylde projektets krav til testunivers indeholdende 1 test-miljø, 1 valideringstest-miljø og 1 produktions-miljø, som hver for sig er fuldstændigt afgrænsede både i form af Rolle / rettigheder, konfiguration, data, certifikater mm. (*eksempelvis namespaces kubernetes*). For yderligere beskrivelse af testuniverset henvises til dokumentet [PV Inddata Testunivers.docx](#)

Systemet har en række integrationer til eksterne services som omfatter **CRL service for certifikater** anvendt af Security Token Servicen på Den Nationale Service Platform (NSP STS), Postgres database, DCC på sundhedsdatanettet (SDN) og Juno (**UVIST omkring det bliver SDN / TLS 1.2**), hvorfor der skal være adgang til dette fra miljøet. Sikkerhedsarkitekturen i disse integrationer er forskellig i et test-

og produktionssetup. Hvorfor De enkelte services' konfigurationer er beskrevet mere detaljeret nedenfor. For dybere indblik i det samlede applikationslandskab henvises til filen [Pharmacovigilance Inddata Arkitektur.docx](#).

## Installation og Drift

I følgende beskrives, hvordan de enkelte services installeres og konfigureres individuelt.

### Generelt

Images fra GitHubs Container Registry [ghcr.io/trifork/](#) kræver en bruger med *read:packages* adgang for pull, da dette er et private repository.

Java – containere kører som brugeren 1000 og nginx kører som brugeren 101 som er nonprivileged i filsystemet.

### BIVWS – Bivirkningsindberetnings Web Service

Image: [ghcr.io/trifork/pvinddata-pvsubmission](#):{TAG}

### Kort Beskrivelse

Spring boot application, som udstiller en SOAP-XML service til bivirkningsindberetning for Lægepraksis systemer og Meldenbivirkning. Anvender NSPs Access library for at validere DGWS security-tickets. Requests valideres og transformeres til HL7 E2B og lagres i databasen, der fungerer som kø-mekanisme.

### Netværk

1. Deler Postgres database med Junosender. Se afsnit om database.
2. Accesshandler kræver adgang til <http://crl.systemtest34.trust2408.com> på test miljøer (Usikkerhed omkring PROD setup. Kræver hosting af NSP CRA service!)
3. Tilgængelig fra NSP via Decouplings komponent (DCC) i Prod
4. Tilgængelig fra internettet i testmiljøerne (LPS integratorer)

### Ressourcer

Ved almindelig belastning bruges 200m CPU og 250 Mi ram. Spring boot bruger 7-10 gange så meget under opstart. Hvis containeren bliver throttlet for meget kan det give class load problemer.

### Environment Variable

Variable	Beskrivelse
NSP_TEST_FEDERATION	Beskrivelse af sikkerhedsmodellen for DGWS. Anvendes af NSP-Accesshandler biblioteket. Skal sættes til "TEST" for test miljøer

### Konfiguration

Nedenstående konfigurationer kan læses ind ved at sætte environment variabelen `SPRING_CONFIG_ADDITIONALLOCATION` (eksempelvis `"file:/etc/config/"`) og mounte en `application.properties` fil derind. Applikationen har classpath i `/BOOT-INF/classes/` så ressourcer kan mountes hertil for at blive en del af classpath (eksempelvis `/BOOT-INF/classes/regionsopdelt-postnummer-2022-06-22.csv` eller `/BOOT-INF/classes/whiteListConfig.yml`).

Property	Beskrivelse	Eksempel
<code>management.server.port</code>	Port til metrikker og prober.  <i>Default: 8080</i>	8083
<code>spring.profiles.active</code>	Der er opsat specifik logging til profilen <code>test-env</code> som logger personfølsomme data.	testenv
<code>dcc.endpoint</code>	Endpoint hvorpå DCC kan nå servicen vi SDN.  Denne property er kun relevant i prod.  <i>Default:</i> <i><code>http://localhost:8080/pvs</code></i>	
<code>juno.skip-validation</code>	Beskriver om R3 xml skal valideres før det lægges i køen.  Skal være <code>True</code> for prod  <i>Default: false</i>	
<code>juno.mapregion</code>	Toggle til om postnummer skal mappe til regioner  <i>Default: true</i>	
<code>juno.largestpostcode</code>	Beskrivelse af største postnummer der skal mappes <i>Default: 10000</i>	
<code>juno.postCodeToRegionMapFileName</code>	Filnavn på region-mapnings-csv. Filen skal ligge fladt på classpath.  <i>Default: regionsopdelt-postnummer-2022-06-22.csv</i> <i>Hvilket peger på en præ-pakket fil.</i>	
<code>juno.ucumClassificationFileName</code>	Filnavn på ucum-klassifikations-mapning-csv. Filen skal ligge fladt på classpath	

	<i>Default:</i> <i>ucum-units-e2b-r3-version-12_en.csv</i> <i>Hvilket peger på en præ-pakket fil.</i>	
spring.config.import	Filnavn på whitelistning fil.  Se nedenfor for dybere forklaring.  <i>Default:</i> mountedWhiteListConfig.yml	whiteListConfig.yml

### Whitelisting

Adgang til servicen baseres på et whitelisting system, hvor systemnavn og cvr fra det STS signeret ID kort, som indgår i DGWS headeren, skal matche en fra listen af whitelistede enheder (whiteListConfig.yml). Derfor skal certifikatet fra PV Client service whitelistedes (cvr: 37052485, systemName: "PVSubmission-Client-Service") samt kommende integratorer indskrives her.

whiteListConfig.yml	Whitelistede entiteter  <i>Default:</i>  whitelist: entities: - cvr: 20921897 systemName: "Trifork" - cvr: 20921897 systemName: "PVSubmission-Client-Service"
---------------------	--

Under opstart Udskriver applikationen, hvor mange enheder, der er whitelisted  
 "d.d.p.submission.config.WhiteListConfig : Loaded 4 whitelisted entities"

### Porte og Prober

Servicen udstilles på port 8080

Prober udstilles på porten der er konfigureret via management.server.port ovenfor

*Liveness sti: /actuator/health/liveness*

*Readiness sti: /actuator/health/readiness*

### Logging og metrikker

Applikationsloggen skrives til Standard out.

Metrikker udstilles på porten der er konfigureret via management.server.port ovenfor på stien  
 /actuator/prometheus

## Junosender

Image: ghcr.io/trifork/pvinddata-junosender:{TAG}

### Kort Beskrivelse

Spring boot application, som fungerer som consumer af de indberetninger, der er lagret i databasen. Via et job fremsøges indberetningsdokumenter fra databasen og afsendes til juno hvorefter status opdateres. Via et job opryddes gamle dokumenter fra databasen. Da der ikke er implementeret håndtering af deadlocks skal der kun være 1 kørende junosender af gangen pr miljø / database.

### Ressourcer

Ved almindelig belastning bruges 100m CPU og 250 Mi ram. Spring boot bruger 7-10 gange så meget under opstart.

### Netværk

1. Deler Postgres database med BIVWS. Se afsnit om database.
2. **Adgang til Juno (som ovenfor usikkerhed på forbindelse)**

### Konfiguration

Nedenstående konfigurationer kan læses ind ved at sætte environment *variablen* `SPRING_CONFIG_ADDITIONALLOCATION` (eksempelvis `""file:/etc/config/""`) og mounte en `application.properties` fil derind.

Bemærk nedenstående cron schedules er specificeret med Spring syntax (se dokumentation [her](#))

Property	Beskrivelse	Eksempel
management.server.port	Port til metrikker og prober. <i>Default: 8080</i>	8083
junosender.enabled	Toogle for afsendelse af beskeder til Juno (junosender-job)  <i>Default:</i> true	
junosender.cron	Spring Cron schedule for afsendelse af beskeder til Juno  <i>Default: 0 0/1 * * * ?</i>	
junosender.batchsize	Batch size for Junosender-job  <i>Default: 100</i>	
junosender.maxretries	Antal gange et dokument skal forsøges afsendt:  <i>Default: 10</i>	
junosender.retrydelay	Minutters delay før afsendes forsøges igen	

	<i>Default: 1</i>	
cleanupjob.enabled	Toogle for oprydning af beskeder (cleanupjob.cron)  <i>Default true</i>	
cleanupjob.cron	Spring Cron schedule for oprydning af gamle data beskeder  <i>Default: 30 0/10 * * * ?</i>	
cleanupjob.batchsize	Batch size for cleanupjob  <i>Default: 100</i>	
cleanupjob.retaindays	Antal dage beskeder skal bevares før de slettes  <i>Default: 30</i>	

#### Porte og Prober

Servicen udstilles på port 8080.

Prober udstilles på porten der er konfigureret via management.server.port

*Liveness sti: /actuator/health/liveness*

*Readiness sti: /actuator/health/readiness*

#### Logging og metrikker

Applikationsloggen skrives til Standard out.

Metrikker udstilles på porten der er konfigureret via management.server.port ovenfor og på stien /actuator/prometheus

#### Database

BIVWS og Junosender integrerer til en fælles database som fungerer som en kømekanime. Konfigurationen for databaseforbindelse i de to komponenter er derfor fælles beskrevet i dette afsnit. BIVWS er producer og Junosender er consumer. BIVWS vil under opstart forsøge at migrere databasen via flyway, hvorefter både BIVWS og Junosender via proprietien (spring.jpa.hibernate.ddl-auto), kan validere/opdatere/slette skemaet ud fra JPA entiteter i kode. Der er pt support for både h2 og postgres database, da migrerings scripts er skrevet i begge dialects. Da både BIVWS og Junosender er skrevet med Spring boot frameworket kan konfiguration foretages enten via environment variable eller properties således:

Property	Beskrivelse	Eksempel
Spring.datasource.url	Database URL. Det eneste krav er at URL	<i>Eksempelvis</i> <i>jdbc:postgresql://\$(DB_HOST):\$(DB_PORT)/\$(DB_NAME)?socketTimeout=1500</i>



Som environment variable: SPRING_DATASOURCE_URL	skal være jdbc:h2 eller jdbc:postgresql  <i>Default:</i> <i>jdbc:h2:mem:pvs;DB_CLOSE_DELAY=-1</i>	<i>Hvor DB_HOST er host navnet på database forbindelsen, DB_PORT er porten på database forbindelse og DB_NAME er navnet på den dedikeret database.</i>
Spring.datasource.username  Som environment variabel: SPRING_DATASOURCE_USERNAME	Database username <i>Default: sa</i>	
Spring.datasource.password  Som environment variabel: SPRING_DATASOURCE_PASSWORD	Database password <i>Default: ""</i>	
spring.jpa.hibernate.ddl-auto	Spring Data JPA specifik konfiguration af DB schema værktøjet.  <u>SKAL</u> være validate i produktion, da  <i>Default: none</i>	validate
spring.datasource.driver-class-name	JDBC Driver configuration  <i>Default: org.h2.Driver</i>	org.postgresql.Driver
spring.jpa.database-platform	Spring Data JPA specifik konfiguration af hibernate dialect  <i>Default:</i> <i>org.hibernate.dialect.H2Dialect</i>	org.hibernate.dialect.PostgreSQLDialect

## Meldenbivirkning

Image: ghcr.io/trifork/pvinddata-meldenbivirkning:{TAG}

### Kort Beskrivelse

Webapplikation til bivirkningsindberetning for både sundhedsprofessionelle og borgere.

### Netværk

1. Skal nåes fra internettet. Filer ligger på stien "/"

### Ressourcer

Ved almindelig belastning bruges 100m CPU og 20 Mi ram.

### Konfiguration

Konfiguration er JSON-baseret.

Nedenstående konfiguration skal mountes til /usr/share/nginx/html/assets/configuration.json

Property	Beskrivelse
backendBasePath	Scheme, host og port for pv client service  <i>Default:</i> https://meldenbivirkning.lms.tcs.trifork.dev/
endpoint	Endpoint for servlet  <i>Skal ALTID være</i> /api/SubmitAdverseDrugReaction
developerSettings	Rodstruktur for konfigurationer som kan anvendes i test miljøer som fx debugging.
DeveloperSettings.headerMessage	Besked stående i headeren i html.  <i>Default: Null</i>
DeveloperSettings.draft	Toogle for om det skal være muligt at autogenere  <i>Default: false</i>

Eksempel:

```
{
  "backendBasePath": https://meldenbivirkning.lms.tcs.trifork.dev/,
  "endpoint": "/api/SubmitAdverseDrugReaction",
  "developerSettings": {
    "headerMessage": "TESTMILJØ",
    "draft": true
  }
}
```

### Porte

Servicen og prober udstilles på port 8080.

Metrikker udstilles på port 9113

Liveness sti: /

Readiness sti: /

#### Logging og metrikker

Applikationsloggen skrives til Standard out.

Metrikker kan nås på stien /nginx\_status

#### Client Service

Image: ghcr.io/trifork/pvinddata-clientservice:{TAG}

#### Kort beskrivelse

Spring boot application fungerende som backend service til Meldenbivirkning. Request mappes og sendes til BIVWS

#### Netværk

1. Adgang til NSP STS
  - a. på testmiljøer (https på port 8443)
  - b. **(Usikkerhed omkring hvad url STS er på I prod)**
2. Adgang til BIVWS på Testmiljøer
3. Adgang til BIVWS via DCC på SDN I prod
4. Skal udstilles på internettet med samme host og port som Meldenbivirkning for at undgå at browser for problemer med CORS. Routing til denne komponent kan ske via pathen /api

#### Ressourcer

Ved almindelig belastning bruges 150m CPU og 250 Mi ram. Spring boot bruger 7-10 gange så meget under opstart. Hvis containeren bliver throttlet for meget kan det give class load problemer.

#### Konfiguration

Nedenstående konfigurationer kan læses ind ved at sætte environment variabelen `SPRING_CONFIG_ADDITIONALLOCATION` (eksempelvis `""file:/etc/config/""`) og mounte en `application.properties` fil derind. Applikationen har classpath `/BOOT-INF/classes/` så ressourcer kan mountes hertil (eksempelvis `/BOOT-INF/classes/cert/prod-cert.p12`). Jeppe Gravgaard har certifikater, der skal anvendes.

Property	Beskrivelse
management.server.port	Port til metrikker og prober.  <i>Default: 8080</i>
app.pvsubmission.base-url	FQDN til BIVWS  Forbindelsen skal gå via SDN I produktion  <i>Default:</i> <i>http://localhost:8887/pvs/BivirkningsindberetningWebService</i>

app.pvsubmission.connect-timeout	Connect timeout i sekunder til BIVWS  <i>Default: 10s</i>
app.pvsubmission.read-timeout	Read timeout i sekunder til BIVWS  <i>Default: 10s</i>
sosi-ws.sts.base-url	Host og port til STS  <i>Default: https://test2-cnsp.ekstern-test.nspop.dk:8443</i>
sosi-ws.sts.connect-timeout	Connect timeout i sekunder til STS:  <i>Default: 10s</i>
sosi-ws.sts.read-timeout	Read timeout i sekunder til STS:  <i>Default: 10 s</i>
sosi-ws.system-information.organisation-cvr	Cvrnummer på kalder-system til DGWS security.  <i>Default: 20921897</i>  Skal sættes til 37052485 i prod
sosi-ws.system-information.organisation-name	Organisations navn på kalder system til DGWS security.  Skal sættes til i Prod Lægemiddelstyrelsen
sosi-ws.system-information.system-name	Systemnavn til DGWS security.  <i>Default:</i> PVSubmission-Client-Service
sosi-ws.keystore-filename	Sti til FOCES certifikat som anvendes for DGWS level 3 kald  <i>Default:</i> /cert/TRIFORK AS - Apotekerforeningen TEST - ASP.p12
sosi-ws.keystore-password	Password til FOCES certifikat  <i>Default: Test1234</i>
sosi-ws.certificate-alias	Keystore Alias for FOCES certifikat  <i>Default: apotekerforeningen test - asp</i> (funktionscertifikat)

### Porte og Prober

Servicen udstilles på port 8080.

Prober udstilles på porten der er konfigureret via `management.server.port`

*Liveness sti: `/actuator/health/liveness`*

*Readiness sti: `/actuator/health/readiness`*

### Logging og metrikker

Applikationsloggen skrives til Standard out.

Metrikker udstilles på `/actuator/prometheus` på den port er konfigureret via `management.server.port` ovenfor

### R2R3 Converter

Image: `ghcr.io/trifork/pvinddata-r2r3converter:{TAG}`

#### Kort beskrivelse

Spring boot application som udstiller en SOAP/XML snitflade. Servicen er midlertidig og API er identisk med deprecated ældre version BIVWS. Servicen opkonvertere kaldet og kalder nye BIVWS.

#### Netværk

1. Adgang til BIVWS
2. Tilgængelig fra NSP Decouplings komponent (DCC)

#### Ressourcer

Ved almindelig belastning bruges 200m CPU og 250 Mi ram. Spring boot bruger 7-10 gange så meget under opstart. Hvis containeren bliver throttlet for meget kan det give class load problemer.

### Konfiguration

Nedenstående properties kan læses ind ved at sætte `SPRING_CONFIG_ADDITIONALLOCATION` (eksempelvis `"file:/etc/config/"`) og mounte en `application.properties` fil ind.

Property	Beskrivelse
<code>management.server.port</code>	Port til metrikker og prober.  Bør være forskellig fra <code>server.port</code> som default er 8080.  <i>Default: 8080</i>
<code>dcc.endpoint</code>	Endpoint hvor på DCC en kan nå servicen (kun relevant i prod)  <i>Default:</i> <a href="http://localhost:8080/r2">http://localhost:8080/r2</a>  Kun relevant i Prod
<code>r3.endpoint</code>	Sti til BIVWS på det interne netværk

	<i>Default:</i> <a href="http://pvsubmission:8080/pvs/BivirkningsindberetningWebService">http://pvsubmission:8080/pvs/BivirkningsindberetningWebService</a>
--	--

#### Porte og Prober

Servicen udstilles på port 8080.

Prober udstilles på porten der er konfigureret via `management.server.port`

*Liveness sti:* `/actuator/health/liveness`

*Readiness sti:* `/actuator/health/readiness`

#### Logging og metrikker

Applikationsloggen skrives til Standard out.

Metrikker udstilles på porten der er konfigureret via `management.server.port` på stien `/actuator/prometheus`

#### Documentation Viewer og Documentation Static

Image: `ghcr.io/trifork/pvinddata-docviewer:{TAG}`

Image: `ghcr.io/trifork/pvinddata-docstatic:{TAG}`

#### Kort beskrivelse

Disse services udgør tilsammen et dokumentationssystem. Documentation Viewer er en nginx-container indeholdende udelukkende én single page application til visning af dokumentation. Dokumentation der skal vises er konfigurerbar for at kunne vise flere versioner på. Denne dokumentation, omfatter autogeneratede filer fra pipelinen, der ligger som statiske filer i nginx containeren doc-static, samt eksterne på eksterne ressourcer så som på særskilt offentlig projekt repository på [her](#).

#### Netværk

1. Doc-viewer skal kunne nåes fra internettet. Filer ligger på stien `/`
2. Doc-static skal kunne nåes fra internettet. Filer ligger på stien `/docs`

#### Ressourcer

Ved almindelig belastning bruges 100m CPU og 20 Mi ram pr container.

#### Konfiguration

Konfiguration er JSON-baseret og består af en række dokumentationssektioner. Hver sektion har en title, en række versioner indeholdende en række items.

Nedenstående konfigurationer skal mountes til `/usr/share/nginx/html/assets/ documentation-items.json`

Property	Beskrivelse
Title	Titel på en sektion

	<i>Default: Er der to sektioner</i> <ul style="list-style-type: none"> <li>- "Genereret fra kode"</li> <li>- "Various external files from Microsoft teams"</li> </ul>
versions[]	En række versioner for denne sektion  <i>Default:</i> <i>Er der en version i hver sektion.</i>
versions[].version	Versionsnummer for denne versionsgruppe  <i>Default: Der findes 1 version af hver sektion</i> <ul style="list-style-type: none"> <li>- 1.0.0 for "Genereret fra kode"</li> <li>- 0.0.1 for "Various external files from Microsoft teams"</li> </ul>
versions[].items[]	En række versionerede dokumentation
versions[].items[].type	Fil typen på denne dokumentation. Skal have typen html   pdf   pdf_in_new_tab   mp4
versions[].items[].url	Url til at hente denne dokumentation. Links til dokumentation under "genereret fra kode" vil altid pege på en instans af docstatic, og links til eksterne vil pege på et tag på github.
versions[].items[].description	Beskrivelse af denne dokumentation

Eksempel som er default konfiguration:

```
[
  {
    "title": "Genereret fra kode",
    "versions": [
      {
        "version": "1.0.1",
        "items": [
          {
            "type": "html",
            "url": "http://localhost:4310/docs/validationerrors.html",
            "description": "Validation errors"
          },
          {
            "type": "html",
            "url": "http://localhost:4310/docs/testreports/pvclientservice/surefire-report.html",
            "description": "Test Report PV Client Service"
          }
        ]
      }
    ]
  }
]
```

```

    },
    {
      "type": "html",
      "url": "http://localhost:4310/docs/testreports/bivws/surefire-report.html",
      "description": "Test Report For BIVWS"
    },
    {
      "type": "html",
      "url": "http://localhost:4310/docs/testreports/junosender/surefire-report.html",
      "description": "Test Report For Junosender"
    },
    {
      "type": "html",
      "url": "http://localhost:4310/docs/testreports/queueolib/surefire-report.html",
      "description": "Test Report For Queueolib"
    },
    {
      "type": "html",
      "url": "http://localhost:4310/docs/testreports/r2r3converter/surefire-report.html",
      "description": "Test Report For R2R3converter"
    },
    {
      "type": "html",
      "url": "http://localhost:4310/docs/testreports/meld-en-bivirkning-
dk/mochareports/report.html",
      "description": "Cypress reports"
    },
    {
      "type": "html",
      "url": "http://localhost:4310/docs/testreports/meld-en-bivirkning-
dk/screenshots/report.html",
      "description": "Cypress screenshots"
    },
    {
      "type": "mp4",
      "url": "http://localhost:4310/docs/testreports/meld-en-bivirkning-
dk/videos/accessibility/accessibility_test.spec.ts.mp4",
      "description": "Cypress Accesibility video (mp4) - more will come"
    }
  ]
}
],
{
  "title": "Various external files from Microsoft teams",
  "versions": [
    {
      "version": "0.0.1",
      "items": [
        {
          "type": "pdf_in_new_tab",

```



```

    "url": "https://github.com/trifork/pvinddata-ekstern-
dokumentation/raw/v0.0.1/BivWS%20Implementeringsguide_reviewTRIFORK.pdf",
    "description": "BivWS Implementeringsguide"
  },
  {
    "type": "pdf_in_new_tab",
    "url": "https://github.com/trifork/pvinddata-ekstern-
dokumentation/raw/v0.0.1/BivWS%20Teknisk%20Dokumentation.pdf",
    "description": "BivWS Teknisk Dokumentation"
  },
  {
    "type": "pdf_in_new_tab",
    "url": "https://github.com/trifork/pvinddata-ekstern-
dokumentation/raw/v0.0.1/Bivirkningsindberetning%20Web%20Service%20(BivWS)%20Snitflade
dokumentation.pdf",
    "description": "Bivirkningsindberetning Web Service (BivWS) Snitfladedokumentation"
  },
  {
    "type": "pdf_in_new_tab",
    "url": "https://github.com/trifork/pvinddata-ekstern-
dokumentation/raw/v0.0.1/Pharmacovigilance%20Inddata%20Arkitektur.pdf",
    "description": "Pharmacovigilance Inddata Arkitektur"
  },
  {
    "type": "pdf_in_new_tab",
    "url": "https://github.com/trifork/pvinddata-ekstern-
dokumentation/raw/v0.0.1/installations-%20og%20driftsvejledning.pdf",
    "description": "Installations- og driftsvejledning"
  }
]
}
]
}
]

```

Default sektion og items skal i princippet altid være tilgængelig, men url skal overskrives.

#### *Porte og Prober*

Servicen og prober udstilles på port 8080.

Liveness sti (docviewer): /

Readiness sti (docviewer): /

Liveness sti (docstatic): /docs/validationerrors.html

Readiness sti (docstatic): /docs/validationerrors.html

#### *Logging og metrikker*

Applikationsloggen skrives til Standard out.

### IP Begrænsning

For test-miljøet og valideringstest-miljøet skal adgang til systemet ip-begrænses til leverandør og kunde. Pr dags dato omfatter dette Trifork, Duckwise og Lægemiddelstyrelsen, hvis CIDR er anført i nedenstående tabel.

CIDR	Ejer
80.160.70.132/32	Lægemiddelstyrelsen
80.160.70.133/32	LægemiddelstyrelsenH
62.199.211.144/28	Trifork (Europaplads)
85.235.239.35/32	vpn11.trifork.com
172.24.141.57/32	Duckwise

### Smoketest

Image: ghcr.io/trifork/pvinddata-sitecar:{TAG}

#### Kort beskrivelse

En container der kan udføre en smoketest ved at integrere til Client service. Via K6-frameworket eksekveres tests og containeren vil give en fejlkode, hvis smoketesten ikke overholder tærskler, som er prædefineret.

#### Netværk

1. Skal have adgang til internettet for at kalde Client Service

#### Ressourcer

For en smoketest 100m CPU og 20 Mi ram.

#### Konfiguration

Miljøet der skal testes kan konfigureres via environment variablen HOST. Eksempelvis

```
` docker run -e URL=https://meldenbivirkning.lms.tcs.trifork.dev ghcr.io/trifork/pvinddata-sitecar:{TAG} `
```