

# Installations- og Driftsvejledning

## Indholdsfortegnelse

Installations- og Driftsvejledning .....	1
Indholdsfortegnelse .....	1
Versionshistorik .....	2
Baggrund .....	3
Krav til Driftsmiljø .....	3
Installation og Drift .....	3
Generelt .....	3
BIVWS – Bivirkningsindberetnings Web Service .....	4
Kort Beskrivelse .....	4
Netværk .....	4
Ressourcer .....	4
Environment Variable .....	4
Konfiguration .....	4
Porte og Proper .....	5
Logging og metrikker .....	5
Junosender .....	6
Kort Beskrivelse .....	6
Ressourcer .....	6
Netværk .....	6
Konfiguration .....	6
Porte og Proper .....	7
Logging og metrikker .....	7
Database .....	7
Meldenbivirkning .....	9
Kort Beskrivelse .....	9
Netværk .....	9
Ressourcer .....	9
Konfiguration .....	9
Porte .....	10
Logging og metrikker .....	10
Client Service .....	10
Kort beskrivelse .....	10
Netværk .....	10

Ressourcer .....	10
Konfiguration .....	10
Porte og Proper.....	12
Logging og metrikker .....	12
R2R3 Converter.....	12
Kort beskrivelse .....	12
Netværk .....	12
Ressourcer .....	12
Konfiguration .....	12
Porte og Proper.....	13
Logging og metrikker .....	13
Documentation Viewer og Documentation Static.....	13
Kort beskrivelse .....	13
Netværk .....	13
Ressourcer .....	13
Konfiguration .....	14
Porte og Propes .....	14
Logging og metrikker .....	14
IP Begrænsning .....	14
Smoketest .....	14
Kort beskrivelse .....	15
Netværk .....	15
Ressourcer .....	15
Konfiguration .....	15

## 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  Omskrevet mount beskrivelse	Jeppe Gravgaard

		Tilføjet generel beskrivelse omkring brugere i docker-container	
0.0.3	10.05.2022	Rettelser på baggrund af Review fra Driftsleverandør	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)

*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 variablen `SPRING_CONFIG_ADDITIONALLOCATION` (eksempelvis `"file:/etc/config/"`) og mounte en `.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`).

Property	Beskrivelse	Eksempel
management.server.port	Port til metrikker og proper.  <i>Default: 8080</i>	8083
spring.profiles.active	Der er opsat specifik logging til profilen test-env som logger personfølsomme data.	Test-env
dcc.endpoint	Endpoint hvorpå DCC kan nå servicen vi SDN.	

	<p>Denne property er kun relevant i prod.</p> <p><i>Default:</i> <i>http://localhost:8080/pvs</i></p>	
juno.skip-validation	<p>Beskriver om R3 xml skal valideres før det lægges i køen.</p> <p>Skal være True for prod</p> <p><i>Default: false</i></p>	
juno.mapregion	<p>Toggle til om postnummer skal mappe til regioner</p> <p><i>Default: true</i></p>	
juno.largestpostcode	<p>Beskrivelse af største postnummer der skal mappes</p> <p><i>Default: 10000</i></p>	
juno.postCodeToRegionMapFileName	<p>Filnavn på region-mapnings-csv. Filen skal ligge fladt på classpath.</p> <p><i>Default: regionsopdelt-postnummer-2022-06-22.csv</i> <i>Hvilket peger på en præ-pakket fil.</i></p>	
juno.ucumClassificationFileName	<p>Filnavn på ucum-klassifikations-mapning-csv. Filen skal ligge fladt på classpath</p> <p><i>Default:</i> <i>ucum-units-e2b-r3-version-12_en.csv</i> <i>Hvilket peger på en præ-pakket fil.</i></p>	

#### Porte og Proper

Servicen udstilles på port 8080

Proper 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 `.properties` fil derind.

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

Property	Beskrivelse	Eksempel
<code>management.server.port</code>	Port til metrikker og proper. <i>Default: 8080</i>	8083
<code>junosender.enabled</code>	Toogle for afsendelse af beskeder til Juno (junosender-job)  <i>Default:</i> true	
<code>junosender.cron</code>	Spring Cron schedule for afsendelse af beskeder til Juno  <i>Default: 0 0/1 * * * ?</i>	
<code>junosender.batchsize</code>	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 Proper

Servicen udstilles på port 8080.

Proper 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 propertyen (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  Som environment variable: SPRING_DATASOURCE_URL	Database URL. Det eneste krav er at URL skal være jdbc:h2 eller jdbc:postgresql  <i>Default:</i> <i>jdbc:h2:mem:pvs;DB_CLOSE_DELAY=-1</i>	<i>Eksempelvis</i> <i>jdbc:postgresql://\$(DB_HOST):\$(DB_PORT)/\$(DB_NAME)?socketTimeout=1500</i>  <i>Hvor DB_HOST er host navnet på database forbindelsen,</i> <i>DB_PORT er porten på database forbindelse og</i> <i>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 konfiguration  <i>Default: org.h2.Driver</i>	org.postgresql.Driver
spring.jpa.database-platform	Spring Data JPA specifik konfiguration af hibernate dialect	org.hibernate.dialect.PostgreSQLDialect



	<i>Default:</i> <i>org.hibernate.dialect.H2Dialect</i>	
--	---	--

## Meldenbivirkning

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

### Kort Beskrivelse

Webapplikation til bivirkningsindberetning for både sundhedsprofessionelle og borgere.

### Netværk

1. Skal nåes fra internettet

### 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> <i>https://meldenbivirkning.lms.tcs.trifork.dev/</i>
endpoint	Endpoint for servlet  <i>Skal ALTID være</i> <i>/api/SubmitAdverseDrugReaction</i>
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 autogenerere  <i>Default: false</i>

Eksempel:

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

```
}
}
```

### Porte

Servicen udstilles på port 8080.

Metrikker udstilles på port 9113

### 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 variablen `SPRING_CONFIG_ADDITIONALLOCATION` (eksempelvis `""file:/etc/config/""`) og mounte en .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 proper.  <i>Default: 8080</i>
app.pvsubmission.base-url	FQDN til BIVWS

	<p>Forbindelsen skal gå via SDN I produktion</p> <p><i>Default:</i>  <a href="http://localhost:8887/pvs/BivirkningsindberetningWebService">http://localhost:8887/pvs/BivirkningsindberetningWebService</a></p>
app.pvsubmission.connect-timeout	<p>Connect timeout i sekunder til BIVWS</p> <p><i>Default:</i> 10s</p>
app.pvsubmission.read-timeout	<p>Read timeout i sekunder til BIVWS</p> <p><i>Default:</i> 10s</p>
sosi-ws.sts.base-url	<p>Host og port til STS</p> <p><i>Default:</i> <a href="https://test2-cnsp.ekstern-test.nspop.dk:8443">https://test2-cnsp.ekstern-test.nspop.dk:8443</a></p>
sosi-ws.sts.connect-timeout	<p>Connect timeout i sekunder til STS:</p> <p><i>Default:</i> 10s</p>
sosi-ws.sts.read-timeout	<p>Read timeout i sekunder til STS:</p> <p><i>Default:</i> 10 s</p>
sosi-ws.system-information.organisation-cvr	<p>Cvrnummer på kalder-system til DGWS security.</p> <p><i>Default:</i> 20921897</p> <p>Skal sættes til 37052485 i prod</p>
sosi-ws.system-information.organisation-name	<p>Organisations navn på kalder system til DGWS security.</p> <p>Skal sættes til i Prod Lægemiddelstyrelsen</p>
sosi-ws.system-information.system-name	<p>Systemnavn til DGWS security.</p> <p><i>Default:</i>  PVSsubmission-Client-Service</p>
sosi-ws.keystore-filename	<p>Sti til FOCES certifikat som anvendes for DGWS level 3 kald</p> <p><i>Default:</i>  /cert/TRIFORK AS - Apotekerforeningen TEST - ASP.p12</p>
sosi-ws.keystore-password	<p>Password til FOCES certifikat</p> <p><i>Default:</i> Test1234</p>
sosi-ws.certificate-alias	<p>Keystore Alias for FOCES certifikat</p>

	<i>Default:</i> apotekerforeningen test - asp (funktionscertifikat)
--	--

### Porte og Proper

Servicen udstilles på port 8080.

Proper 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 `.properties` fil ind.

Property	Beskrivelse
<code>management.server.port</code>	Port til metrikker og proper.  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
r3.endpoint	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 Proper*

Servicen udstilles på port 8080.

Proper 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-doc-viewer**:{TAG}

Image: ghcr.io/trifork/**pvinddata-doc-static**:{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 én gang uden behov for nyt release. Denne dokumentation, ligger som statiske filer i nginx containeren doc-static, og omfatter autogenerede filer fra pipelinen. Dog vil yderligere filer indgå i den samlede dokumentationspakke som opbevares i eksterne kilder (Microsoft Teams). Disse vil således skulle mountes ind i doc-static som vist nedenfor.

Disse services skal kun installeres i Prod miljøet.

#### *Netværk*

1. Doc-viewer skal kunne nåes fra internettet
2. Doc-static skal kunne nåes fra internettet

#### *Ressourcer*

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

### Konfiguration

Doc-static	
Fil navn I Teams.	Mount path
BivWS Teknisk Dokumentation.docx	/usr/share/nginx/docs/ext/BivWS_Teknisk_Dokumentation.pdf
Installations- og driftsvejledning.docx	/usr/share/nginx/docs/ext/Installations-_og_driftsvejledning.pdf

Eksterne filer exporteres fra Microsoft teams af leverance-teamet og kan overleveres i docx eller pdf format efter behov. Den samlede liste af ekstern dokumentation kan ses [HER](#).

Doc-viewer
{}

Bemærk de eksterne filer som er mounted til doc-static er indsat med dokument type og sti her i

Ovenstående properties skal mountes til /usr/share/nginx/html/assets/configuration.json

### Porte og Propes

Servicen 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

### 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 variabelen HOST. Eksempelvis

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