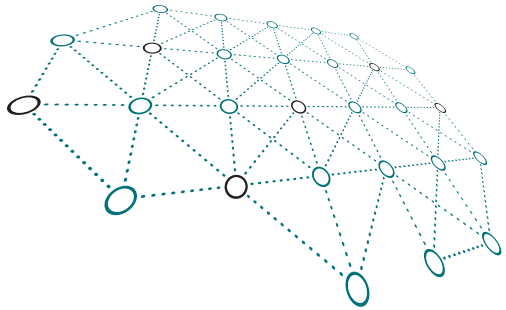


4S seminar

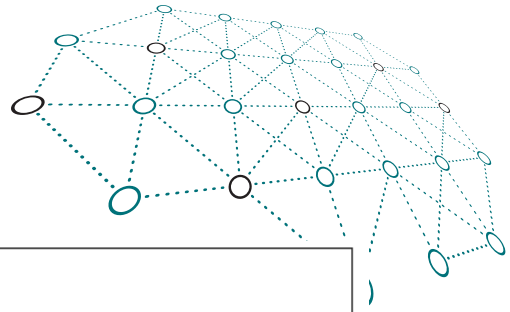
Om open source-microservices, HL7 FHIR mv. til telemedicin og telesundhed

Michael Christensen
Koordinator for Softwaregruppen i 4S

Chef Softwarearkitekt ved Health IT, Alexandra Instituttet

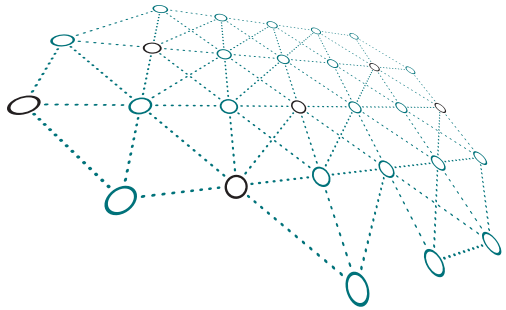
STIFTELSEN FOR SOFTWAREBASEREDE SUNDHEDSSERVICES

Dagens program



9.30 - 10.00	Kaffe og rundstykker
10.00 - 10.15	Velkomst og siden sidst v. Morten Kyng, 4S koordinator, Aarhus Universitet, Alexandra Instituttet
10.15 - 11.00	4S CDA buildere, v. Bjarne Hansen, Lakeside Bjarne Hansen kommer og fortæller om design og udvikling af nye open source-CDA buildere, som med finansiering fra MedCom og med Systematic som udviklere har udviklet et værktøj til produktion af HL7 PHMR, QRD og QFDD dokumenter efter de danske profileringer af den internationale standard.
11.00 - 11.45	OpenTele3 proof of concept-projektet, v. Michael Christensen, Alexandra Instituttet og 4S Michael Christensen fortæller om proof of concept-projektet omkring udviklingen af en ny OpenTele-generation, som er microservicebaseret. Projektet, som løber fra november 2016 til ultimo januar 2017 konkretiserer omfang af og krav til en OpenTele3 microservicebaseret infrastruktur for telemedicin og søger at afdække og beskrive, hvordan vi på den tekniske front effektivt kan understøtte open source-flerleverandørstrategi, forbedrede muligheder ift. CE-mærkning og kvalitetssikring, referencearkitekturer mv. Projektet er finansieret af Region Midtjylland.
11.45 - 12.30	Frokost
12.30 - 13.00	Samtykkehåndtering i microservicearkitekturer, v. Mads Schaarup Andersen og Jonas Lindstrøm, Alexandra Instituttet Mads og Jonas fra Alexandra Instituttets Security Lab kommer og fortæller om deres seneste arbejde med samtykkehåndtering i microservicearkiteturer. Et arbejde som er udført i tæt samarbejde med OpenTele3 proof of concept-projektet.
13.00 - 13.30	MedCom om FHIR, v. Michael Johansen, MedCom Michael Johansen fortæller om indtryk fra FHIR DevDays 2016 og giver et bud på, hvordan MedComs erfaringer fra de tidlige EDI-dage potentielt kan inddrages i kommende arbejde med FHIR. Endelig giver Michael en status på det interne MedCom-projekt, hvor mapning fra XRPT01 laboratoriesvar til FHIR afprøves.
13.30 - 14.30	Diskussion og opsamling

4S STIFTE



OpenTele3 forprojekt

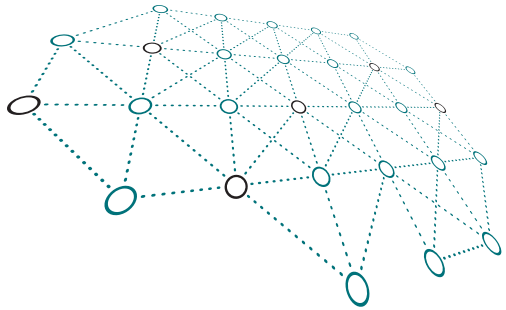
Proof of Concept og konkretisering

Michael Christensen
Koordinator for Softwaregruppen i 4S

Chef Softwarearkitekt ved Health IT, Alexandra Instituttet

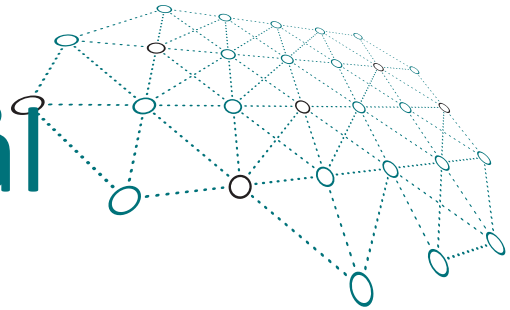
STIFTELSEN FOR SOFTWAREBASEREDE SUNDHEDSSERVICES

Projektrammer



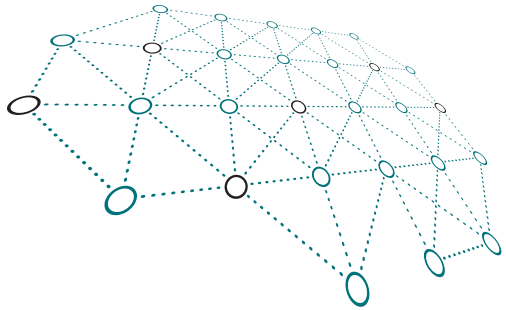
- *Periode: November 2016 – januar 2017*
- *Finansieret af Region Midtjylland*
- *Suppleret med bidrag fra Alexandras resultatkontrakter*
  - Nem adgang til datadeling på tværs af sundhed og velfærd
  - *Sikkerheds- og privacyværktøjer*

Forprojekt: Overordnede mål



- Input til udbud
- Vurdering af det samlede omfang af opgaven
- Beslutningsgrundlag mht. OT1, OT2, OT3
- Kørende prototyper, som leverandører kan anvende til eksperimenter

OpenTele3 strategiske mål



Effektiv understøtte:

- Open source flerleverandørstrategi
- Sammenhængen med projekt Modning af Telemedicinsk Infrastruktur (MaTIS)
- Det tværsektorielle samspil mellem regioner, kommuner og øvrige sundhedsaktører
- Forbedrede muligheder ift. CE-mærkning og kvalitetssikring

Forprojekt: Konkrete leverancer

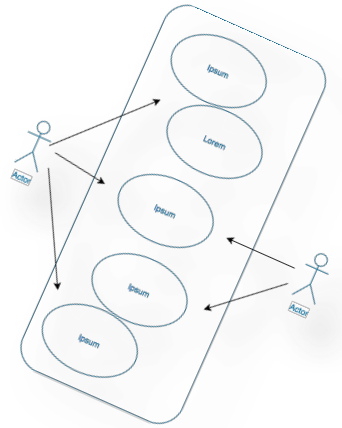
Prototyper:

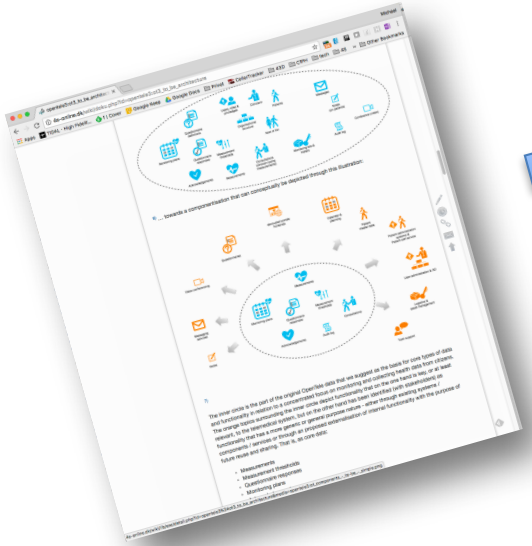
- Prototype af serviceplatform for microservice infrastruktur for telemedicin
- Prototyper af udvalgte microservices, som understøtter en specifik telemedicinsk case

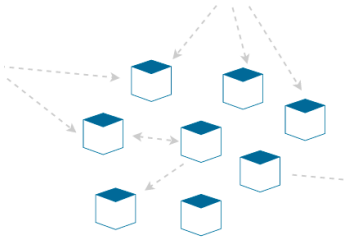
Notat:

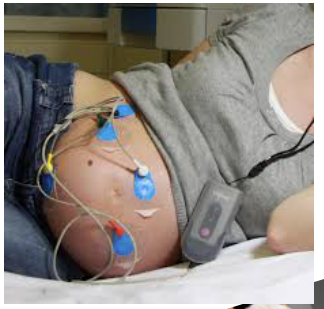
- Oplæg til krav til serviceplatform for microservice infrastruktur
- Minimumssæt af microservices, som er nødvendige for at understøtte et basalt telemedicin setup.
- Oplæg til roadmap for udvikling af platform og services

Udgangspunkterne



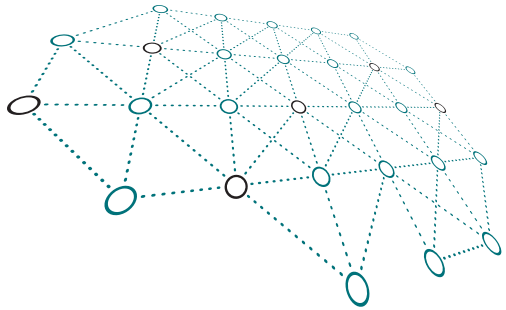






- microservices
- IoT
- sundhed/telemedicin

Principper



General principles

1. Work towards a microservice architectural style
2. Use international standards for communication and content formats
3. Aim for cross-platform support and low platform dependence

Microservices

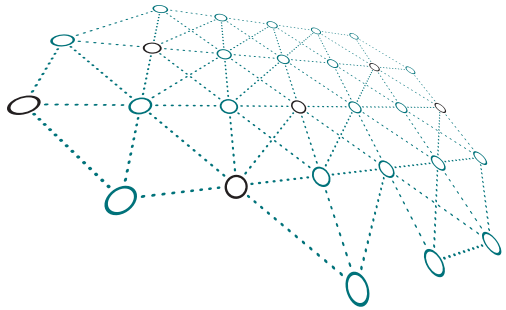
- Small, and doing one thing well
- Autonomous

- Loose coupling and high cohesion
- Separate frontends from backend data services.
- Decentralized data management
- Small number of ways for components to interact - well described and standardised.
- Componentisation via services

Specific principles

1. Does the national eHealth reference architectures say anything?
2. Does the Continua Design Guidelines say anything?
3. Does HL7 FHIR say anything?
4. If your service handles any kind of resource, support a RESTful interface based on ~~hl~~ FHIR

1. skridt:
Prioritering af kvalitetsattributter



Category 5:

Modifiability: other groups or individuals can easily add new functionality to the platform and/or the microservices.

Monitorability: it must be possible to monitor resource usage, errors, dependencies and other important aspects of the system. As a part of a microservice system, it is crucial to be able to monitor the system at all times.

Developer distribution: in an open source world this must be prioritized as high as possible.

Usability (of architecture): from the developers' point view, it must be easy to add new components and/or use existing components as easy as possible (i.e. usability is not towards the end users - in this project, all though this is how Bass et al meant it).

Category 4:

Interoperability: since there are many different suppliers to the real system, the various microservices in the system must be able to communicate easily, and the microservice-platform must be able to communicate easily with the microservices.

Security: must have high priority but not a cat. 5 since this would mean security dominates over everything below.

Testability: it must be easy to test the microservices and the integration of the microservices with each other and the platform.

Category 3:

Deployability (single service): it should be easy to deploy single services.

Scalability: both horizontal scaling and load balancing must be a part of the architectural solution

Availability: it is not mission critical or life critical that the system is running 24/7. If the system is down for 2 minutes, it is okay, and this is why it is prioritized this as cat 3 and not higher

Variability: the system is able to handle different kinds of tasks and in different domains. But in this project the design needs only to include tasks in the telemedicine domain (regional and municipal use)

Category 2:

Mobility (like elasticity): it must not be a requirement that the components/microservices scale automatically, but the services must be able to be replaced or moved as needed.

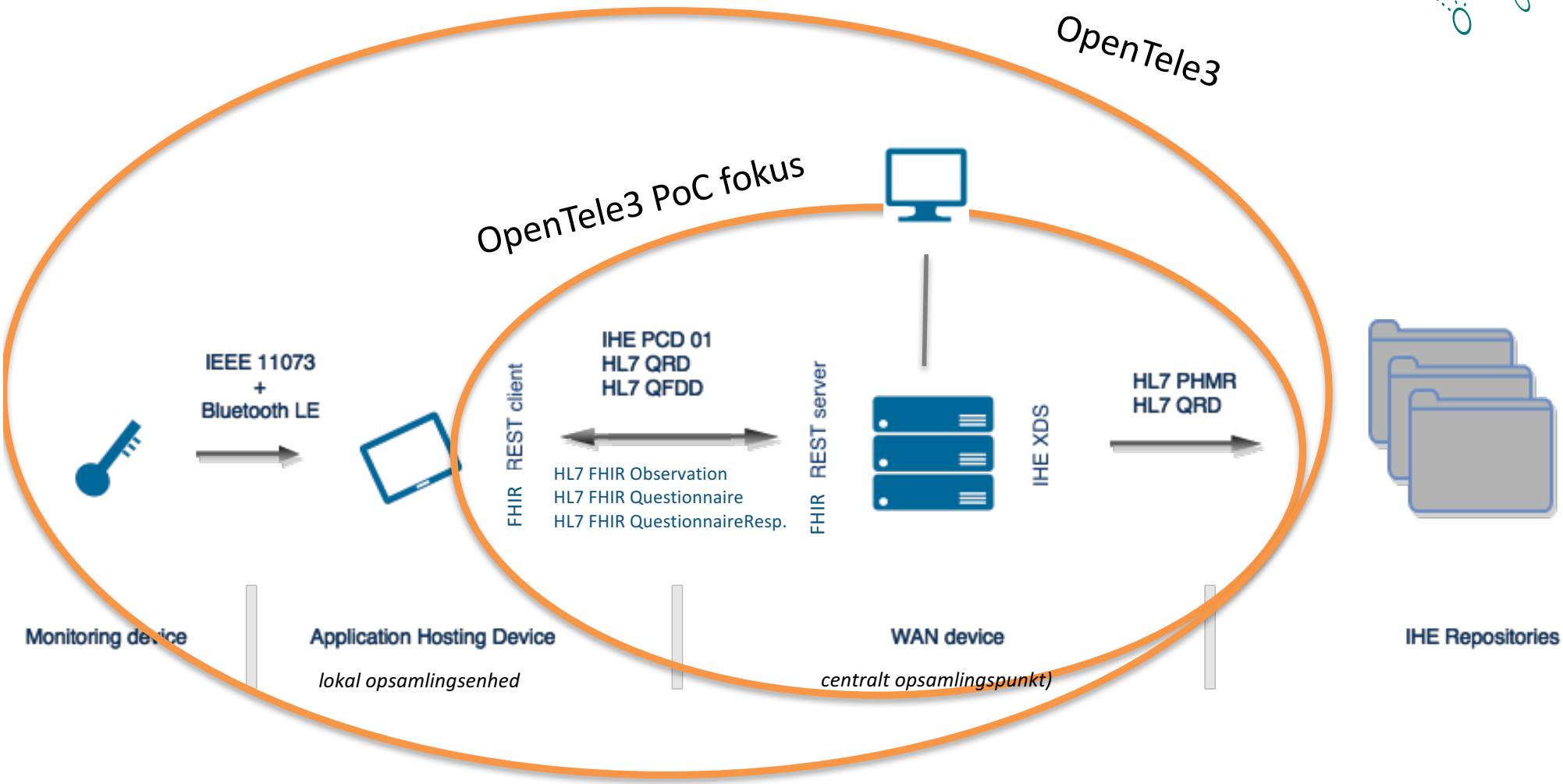
Performance: since this is not a high performance system with low latency requirements, and since there are not many concurrent users of the system, this attribute is prioritized low.

Category 1:

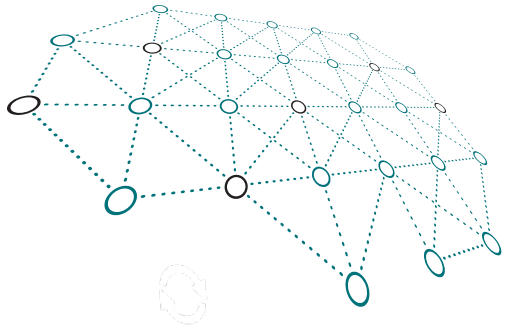
Deployability (whole platform): the production system is not intended to be moved around from time to time. Rather, we know that the system will be deployed at specific hosts.

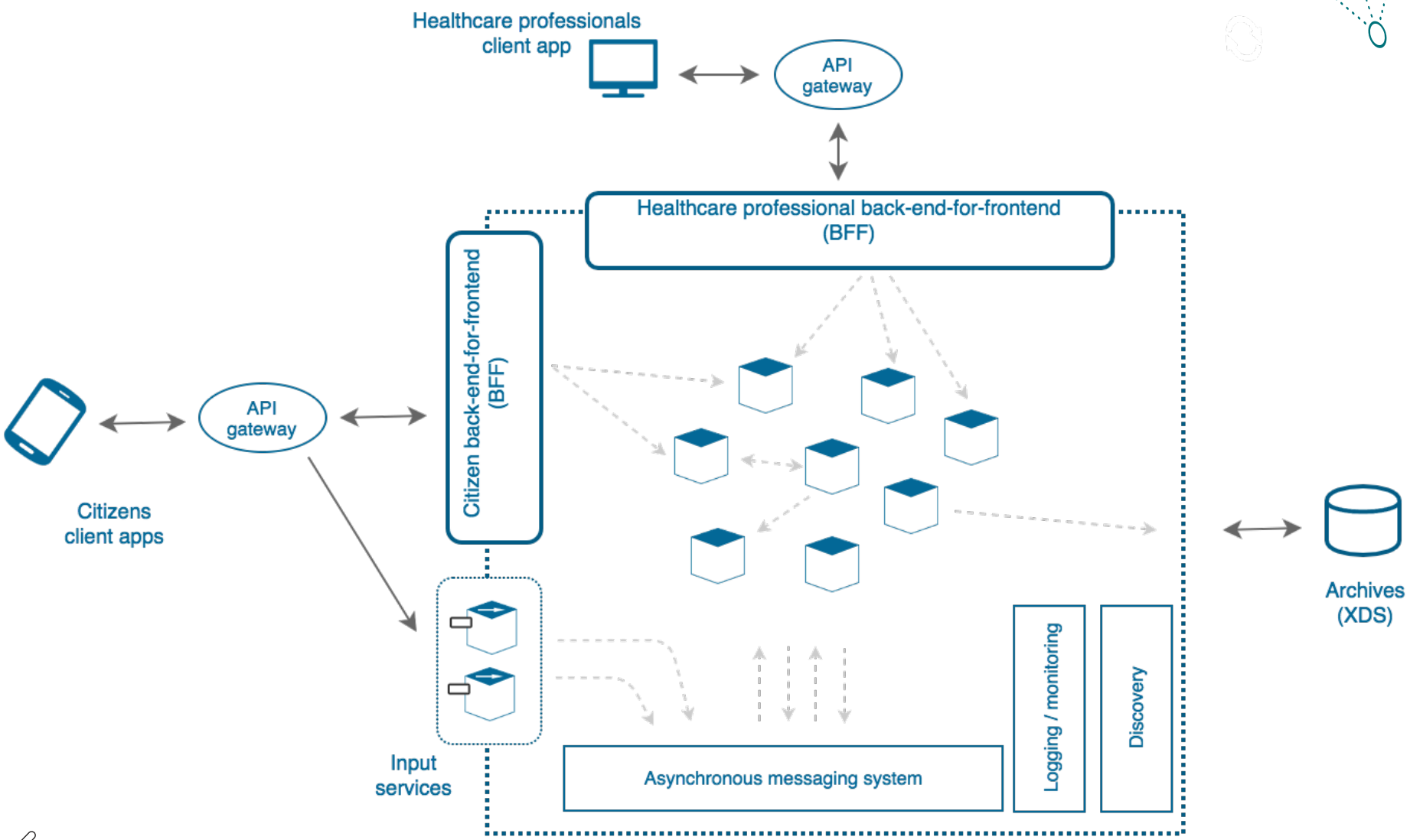
Portability: it is not necessary that the production system is able to run on a bunch of different platforms.

Fokus og standarder

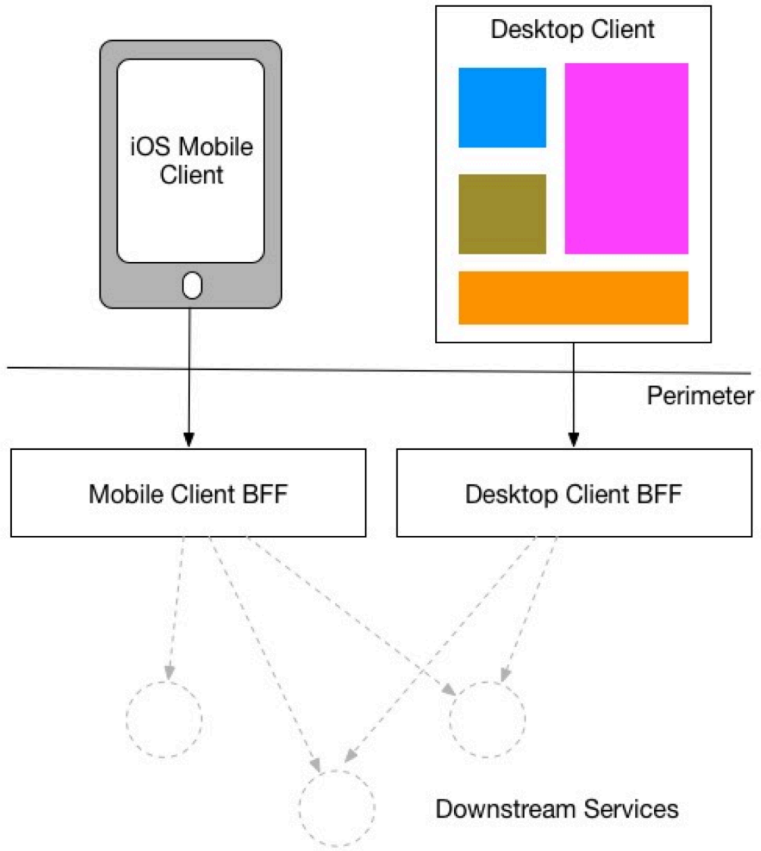


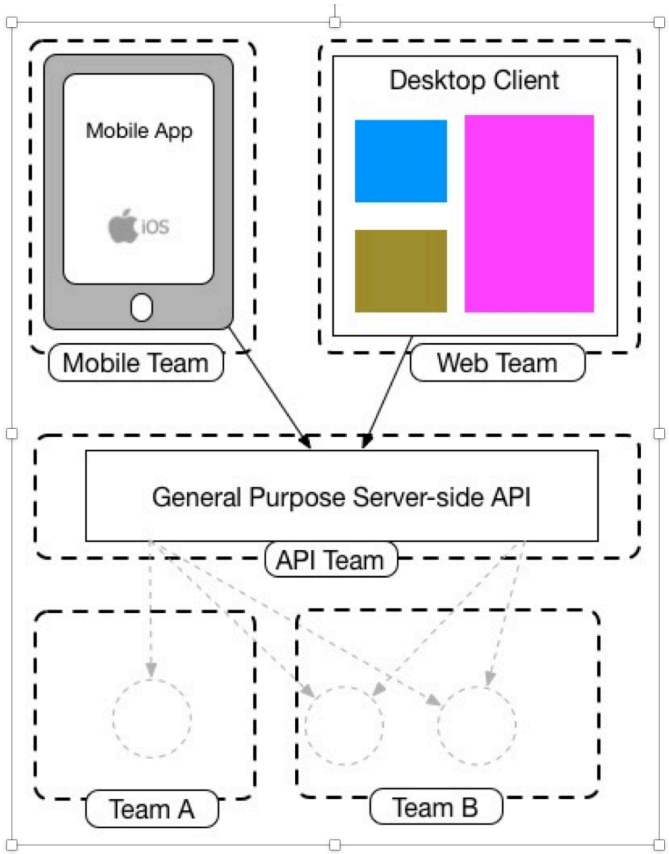
Overordnet arkitektur





Backends-for-frontends

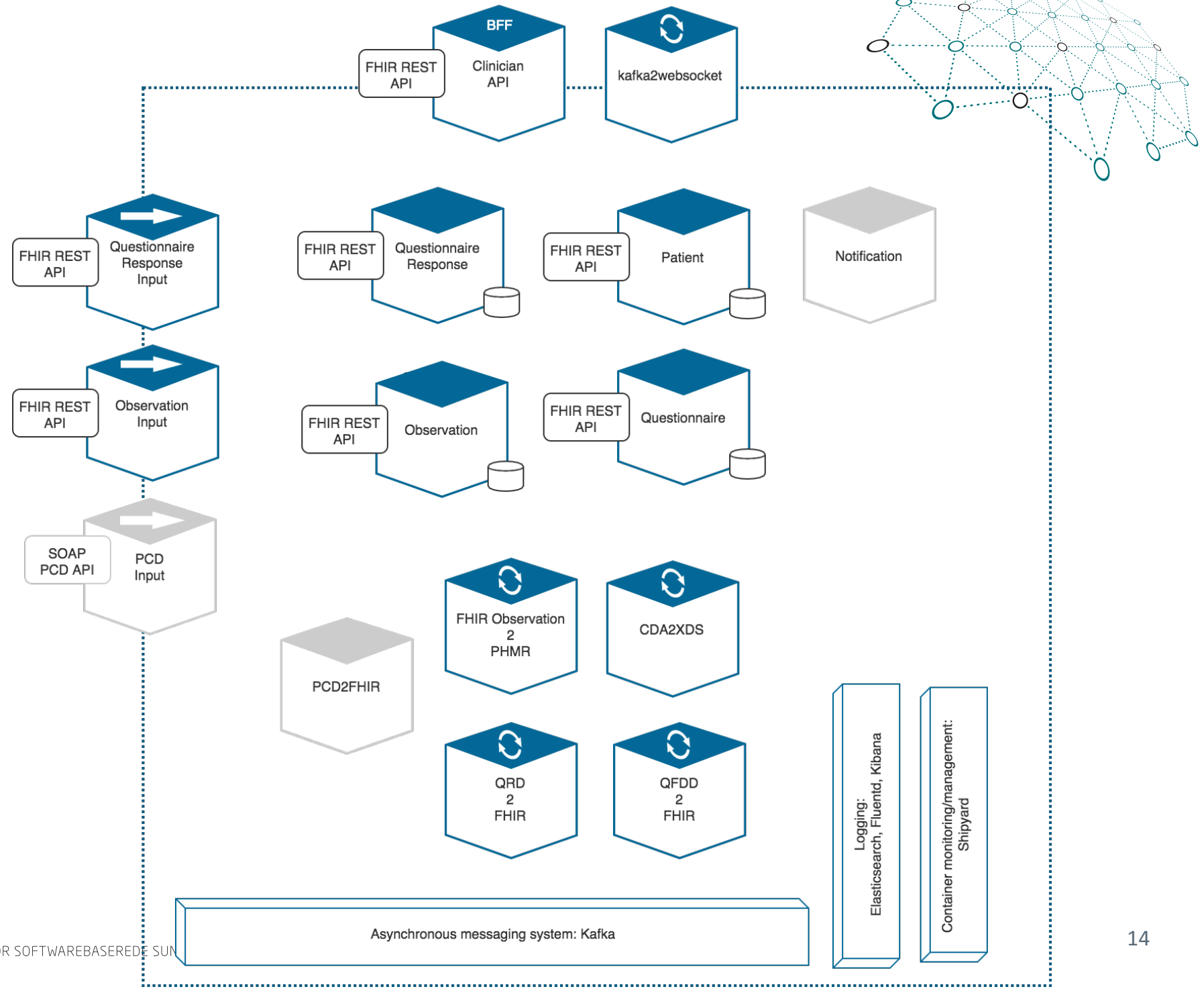




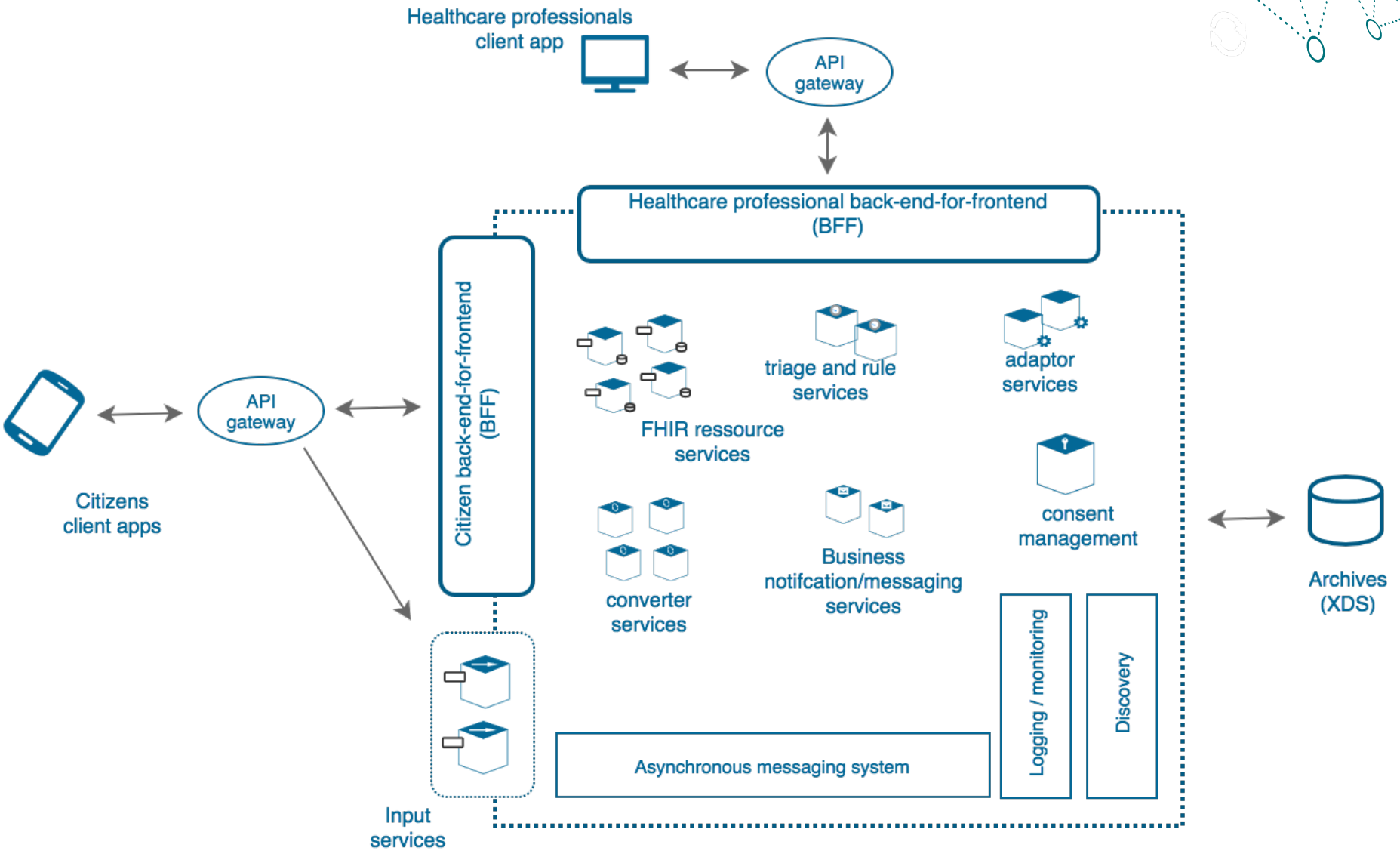
Tabel 2: BFF versus general purpose server-side API iflg. [7]

Kilde: http://samnewman.io/patterns/architectural/bff/

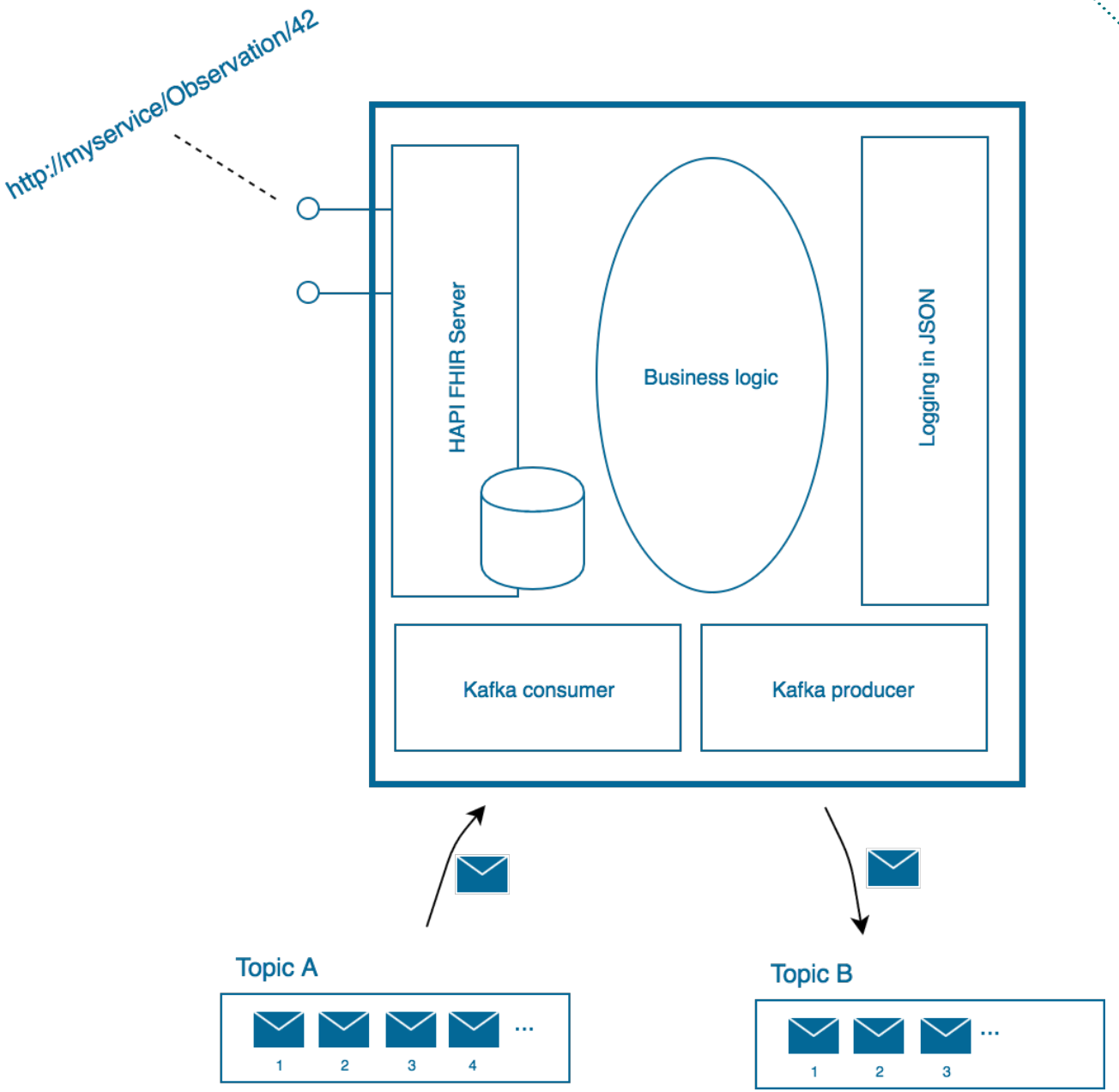
Implementerede services



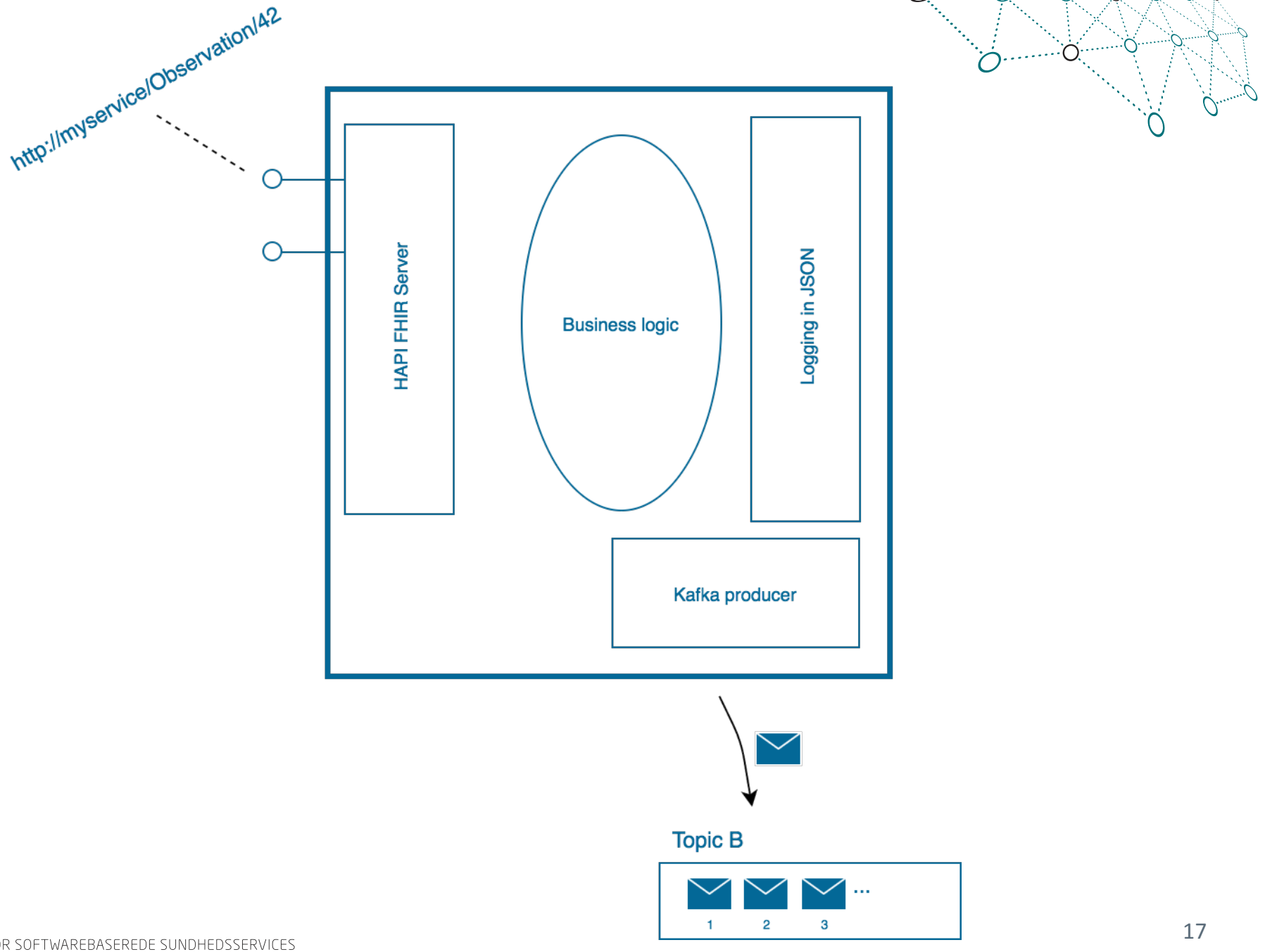
Servicetyper



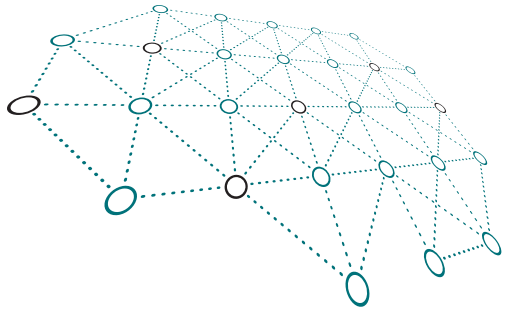
Service struktur – FHIR resource service

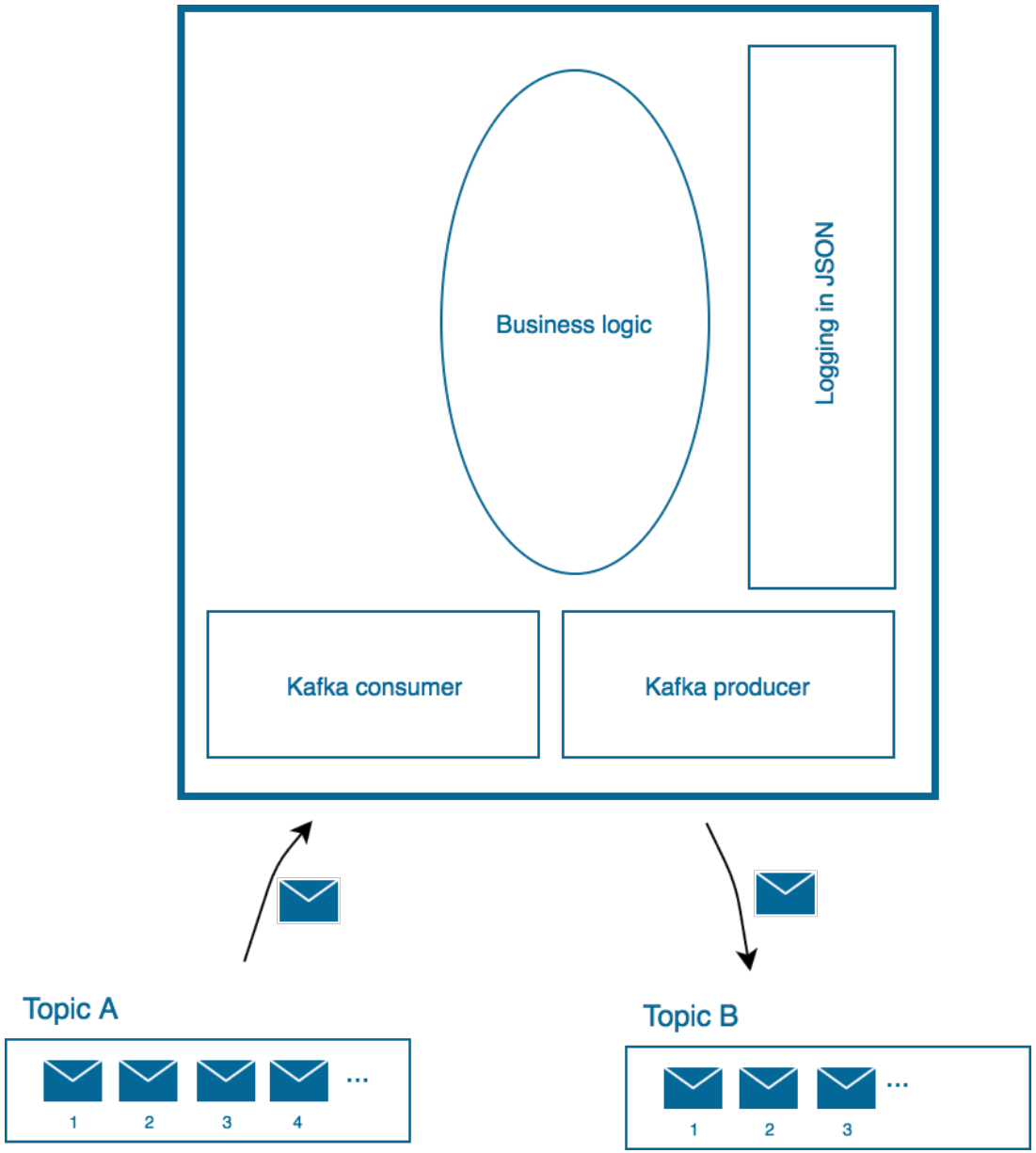


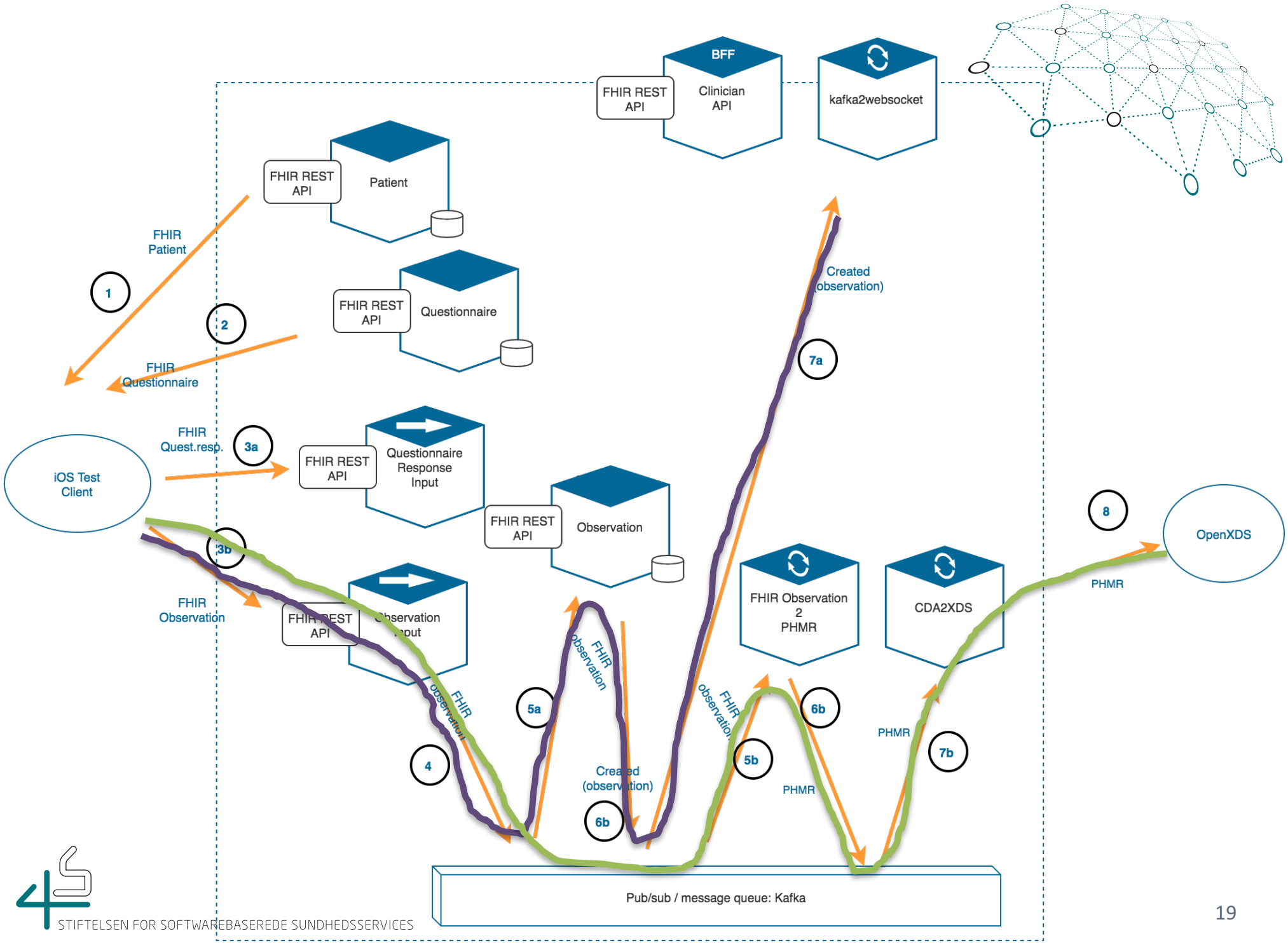
Service struktur – input service



Service struktur – converter service



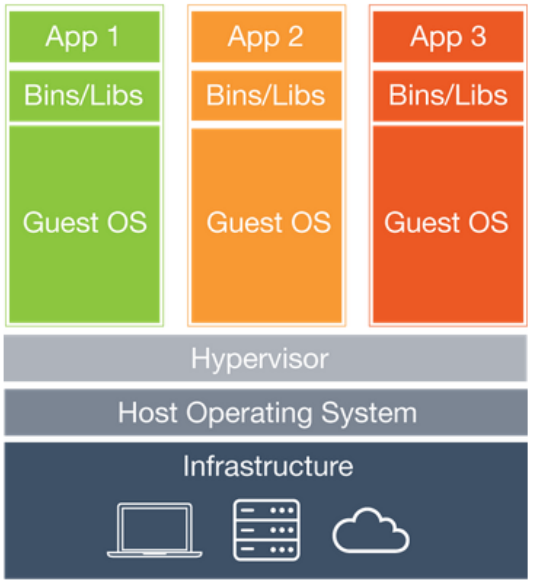



BFF
FHIR REST API
Clinician API
kafka2websocket
FHIR REST API
Patient
FHIR Patient
1
2
FHIR REST API
Questionnaire
FHIR Questionnaire
iOS Test Client
FHIR Quest.resp.
3a
FHIR REST API
Questionnaire Response Input
FHIR REST API
Observation
3b
FHIR Observation
FHIR REST API
Observation Input
FHIR observation
4
5a
FHIR observation
Created (observation)
6b
Created (observation)
7a
FHIR observation
5b
FHIR Observation 2 PHMR
6b
PHMR
CDA2XDS
7b
8
OpenXDS
PHMR
Pub/sub / message queue: Kafka

Deploymentplaform: Docker

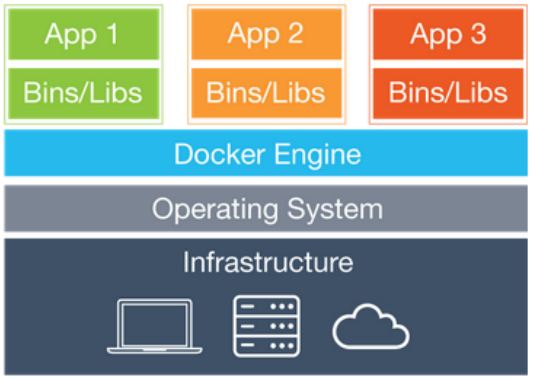
How is this different from virtual machines?

Containers have similar resource isolation and allocation benefits as virtual machines but a different architectural approach allows them to be much more portable and efficient.



Virtual Machines

Each virtual machines includes the application, the necessary binaries and libraries and an entire guest operating system - all of which may be tens of GBs in size.

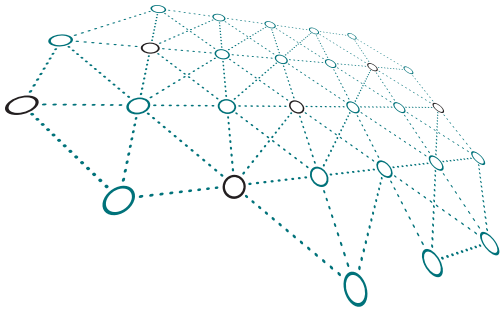


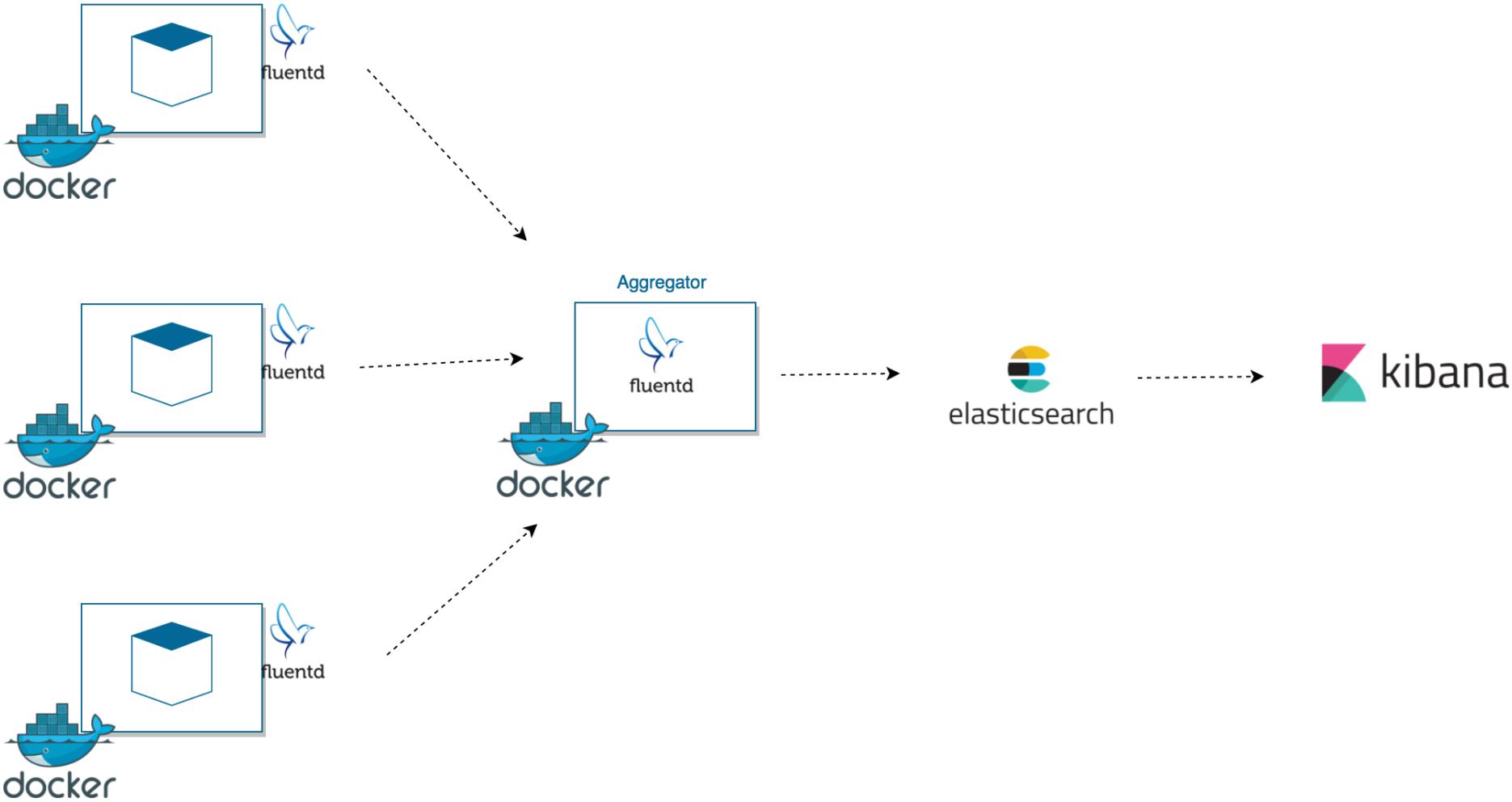
Containers

Containers include the application and all of its dependencies, but share the kernel with other containers. They run as an isolated process in userspace on the host operating system. They're also not tied to any specific infrastructure – Docker containers run on any computer, on any infrastructure and in any cloud.

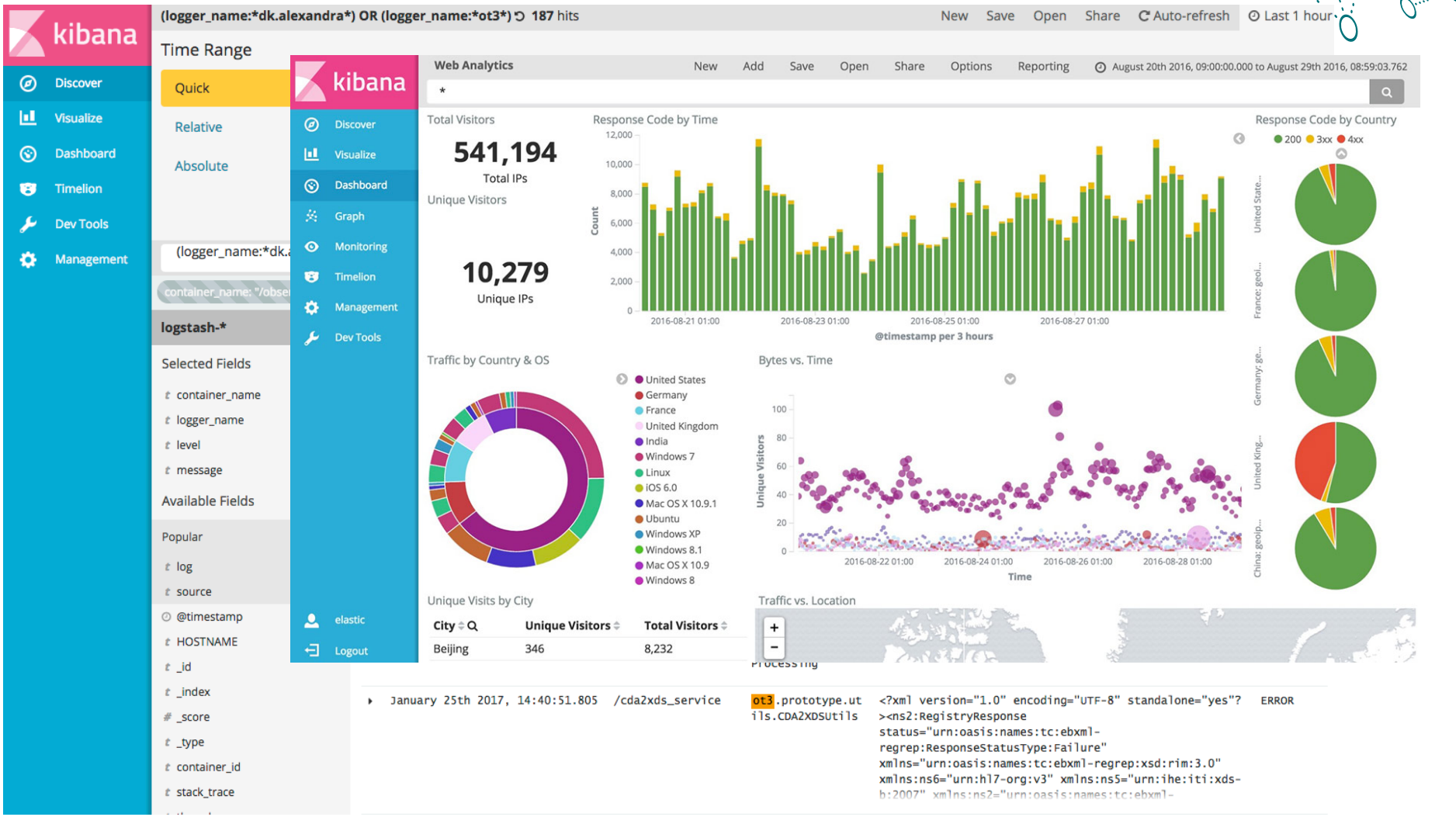
Kilde: http://www.docker.com

Monitoring




fluentd
docker
fluentd
docker
Aggregator
fluentd
docker
elasticsearch
kibana
fluentd
docker

Kibana


kibana
(logger_name:*dk.alexandra*) OR (logger_name:*ot3*) 187 hits
New Save Open Share Auto-refresh Last 1 hour
Time Range
Quick
Relative
Absolute
Discover
Visualize
Dashboard
Timelion
Dev Tools
Management
logstash-*
Selected Fields
container_name
logger_name
level
message
Available Fields
Popular
log
source
@timestamp
HOSTNAME
_id
_index
_score
_type
container_id
stack_trace
Web Analytics
New Add Save Open Share Options Reporting
August 20th 2016, 09:00:00.000 to August 29th 2016, 08:59:03.762
Graph
Monitoring
elastic
Logout
Total Visitors
541,194
Total IPs
Unique Visitors
10,279
Unique IPs
Response Code by Time
Count
@timestamp per 3 hours
Response Code by Country
200 3xx 4xx
Traffic by Country & OS
United States
Germany
France
United Kingdom
India
Windows 7
Linux
iOS 6.0
Mac OS X 10.9.1
Ubuntu
Windows XP
Windows 8.1
Mac OS X 10.9
Windows 8
Bytes vs. Time
Unique Visitors
Time
Unique Visits by City
City
Unique Visitors
Total Visitors
Beijing
346
8,232
Traffic vs. Location
January 25th 2017, 14:40:51.805
/cda2xds_service
ot3.prototype.utils.CDA2XDSUtils
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns2:RegistryResponse status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure" xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" xmlns:ns6="urn:hl7-org:v3" xmlns:ns5="urn:ihe:iti:xds-b:2007"
ERROR

Docker monitoring/management

shipyard | CONTAINERS | IMAGES | NODES | REGISTRIES | ACCOUNTS | EVENTS | ADMIN

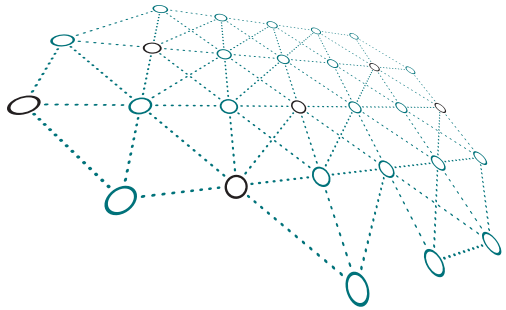
Refresh | Deploy Container | Search containers...

	Id	Node	Name	Image	Status	Created	Actions
☐	●	792088d130b5	ot3dev	observation_service	docker.alexandra.dk/ot3/observation-service:0.6	Up 5 days	2017-01-25 11:11:09 +0100
☐	●	acc48efa079c	ot3dev	questionnaire_service	docker.alexandra.dk/ot3/questionnaire-service:0.6	Up 5 days	2017-01-25 11:06:29 +0100
☐	●	07811ef18b94	ot3dev	questionnaire_response_service	docker.alexandra.dk/ot3/qr-service:0.6	Up 5 days	2017-01-25 11:03:32 +0100
☐	●	acc481bd78f1	ot3dev	obs_input_service	docker.alexandra.dk/ot3/obs-input-service:0.6	Up 5 days	2017-01-25 10:58:33 +0100
☐	●	57d118841029	ot3dev	qrd_input_service	docker.alexandra.dk/ot3/qrd-input-service:0.5	Up 5 days	2017-01-25 10:50:56 +0100
☐	●	ea7d2d4c86d4	ot3dev	clinician_ui	docker.alexandra.dk/ot3/clinician-ui:1.0.8	Up 5 days	2017-01-24 15:51:04 +0100
☐	●	7239d6a44c2b	ot3dev	cda2xds_service	docker.alexandra.dk/ot3/cda2xds-service:0.4	Up 6 days	2017-01-23 22:27:21 +0100
☐	●	6ec626ac0f95	ot3dev	patient_service	docker.alexandra.dk/ot3/patient-service:0.6	Up 6 days	2017-01-23 19:21:27 +0100
☐	●	7944ad498fd1	ot3dev	qrd_to_fhir_service	docker.alexandra.dk/ot3/qrd-to-fhir-service:0.3	Up 6 days	2017-01-23 19:10:44 +0100

Øvrige overordnede elementer

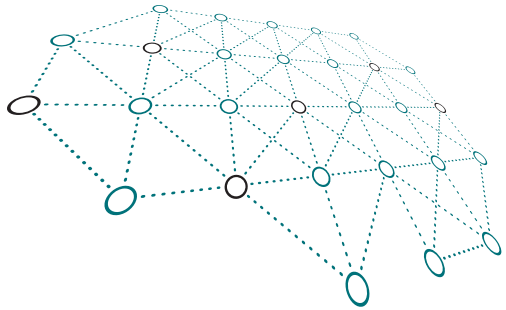
- Discovery
  - DNS-baseret (f.eks. vha. Consul)
  - http://myservice.host.dk:8084/Observation vs. http://myservice/Observation
- Orkestrering
  - F.eks. Docker Swarm eller Kubernetes, men
  - afhænger af konkret deploymentmiljø
- Sikkerhed
  - Mulighed for ekstern/separat authentication
    - I MS-infra mappes der fra identitet til rolle / "intern identitet"
  - Mere om sikkerhed efter frokost...

Anbefalinger



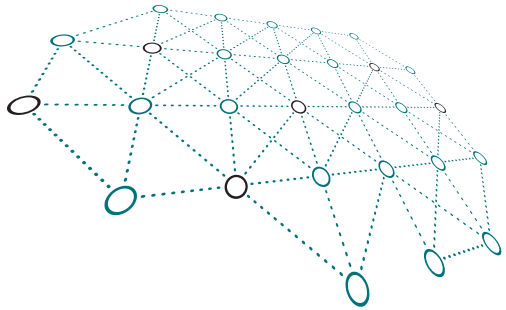
- Asynkront beskedsystem:
  - Udveksl så vidt muligt data via asynk. beskedsystem
  - Giver løs kobling, løbende processering, læsning af data flere steder med forskelligt formål
- Central/global logaggregering
  - log til JSON => bedrer filtreringsmuligheder
- Brug en API gateway til routing af trafik fra klienter
  - *Single point of entry* for borgerrettede og sundhedsprofessionelles apps.
  - Indkapsler den interne microserviceinfrastruktur
  - Typisk også være involveret i andre opgaver, såsom load balancing, caching og autentificering.
- Anvend backends-for-frontends
  - Samler og udstiller forretningsfunktionalitet
  - Rettet mod specifikke typer af klienter
- Fokus på automatisering
  - Test og continuous deployment
  - Dokumentation af snitflader
- Governance through code [Newman]
  - Ikke et perfekt eksempel, men en faktisk service, som gør tingene rigtigt
  - Skræddersyede service templates

FHIR og microservices

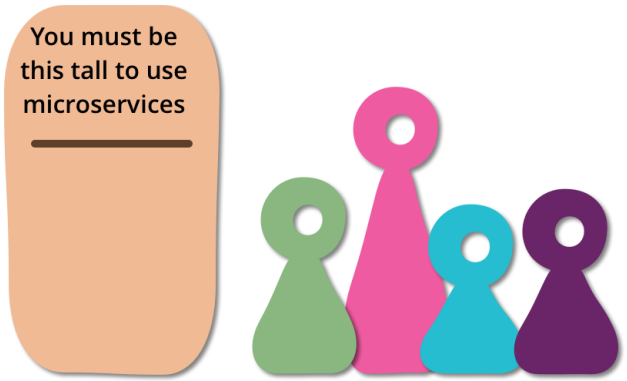


- FHIR publish/subscribe / messaging vs OT3 Kafka fremgangsmåden
  - IBM Watson Health
- Er det godt til både eksterne og interne snitflader?
  - Relativt store objekter man sender rundt
  - Med STU3 kommer der patch-funktionalitet
- Conformance statement
  - http://myservice/metadata
  - Udnytte i microservicearkitektur: Jævnligt tjekke conformance statement hos services, som man afhænger af eller bygge det ind i service discovery
- CDA vs FHIR gapanalyse
  - PHMR og FHIR Observation, Device, Patient mv.
  - HL7 QFDD/QRD og FHIR Questionnaire/QuestionnaireResponse
- FHIR og url-referencer til ressourcer
- Transactions eller ej

Kompetencer

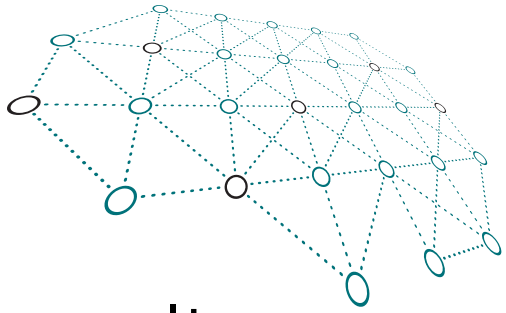


- Agil udvikling
- Automatisering / Continuous …:
  - Byg, test, integration, deployment
- Erfaringer med containerteknologier
- Fowler:
  - Rapid provisioning
  - Basic Monitoring
  - Rapid application deployment



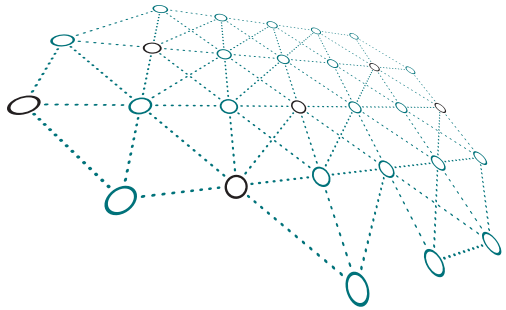
https://martinfowler.com/bliki/MicroservicePrerequisites.html

Drift



- I starten kan du godt gøre det manuelt, men...
- Continuous delivery/deployment (CD) pipeline
- Alternativ til CD til drift: CD til staging
  - evt. med blå/grøn deployment
  - Men husk krav til monitorering

Næste skridt



- Release til open source
- Slutnotat og demo til Region Midtjylland
- 4S softwaregruppen: OpenTele og microservices – hvornår, hvordan og hvorfor?

- Udbud