Hvordan bygge sikker og vanntett plattform “Everything by Design”

Frode I. Bjerk,
ITSMF konferansen 2024, Gardermoen

Versjon 0.9

Introduksjon og agenda

- Hvor er vi – bakgrunn
- Hva er problemet?
- Hvorfor skal vi løse problemet?
- To nivåer
  - Utviklings prosessen
  - Informasjon og dataprodukter
- Fordeler av å jobbe med hele verdikjeden utviklingssammenheng

Bakgrunn

Sikkerhet i et tidsperspektiv

Autonome team, et Proof of Concept

Metode og prinsipper for endring

IT sikkerhetsarbeid, i et tidsperspektiv

Forebyggende / Omgående

- "Shift left"
- Automatisk
- Attraktivt

Operasjonelt / dag til dag

- Sikkerhets-prinsipper
- Manuelt

Kontinuerlig utvikling / tilbake-melding

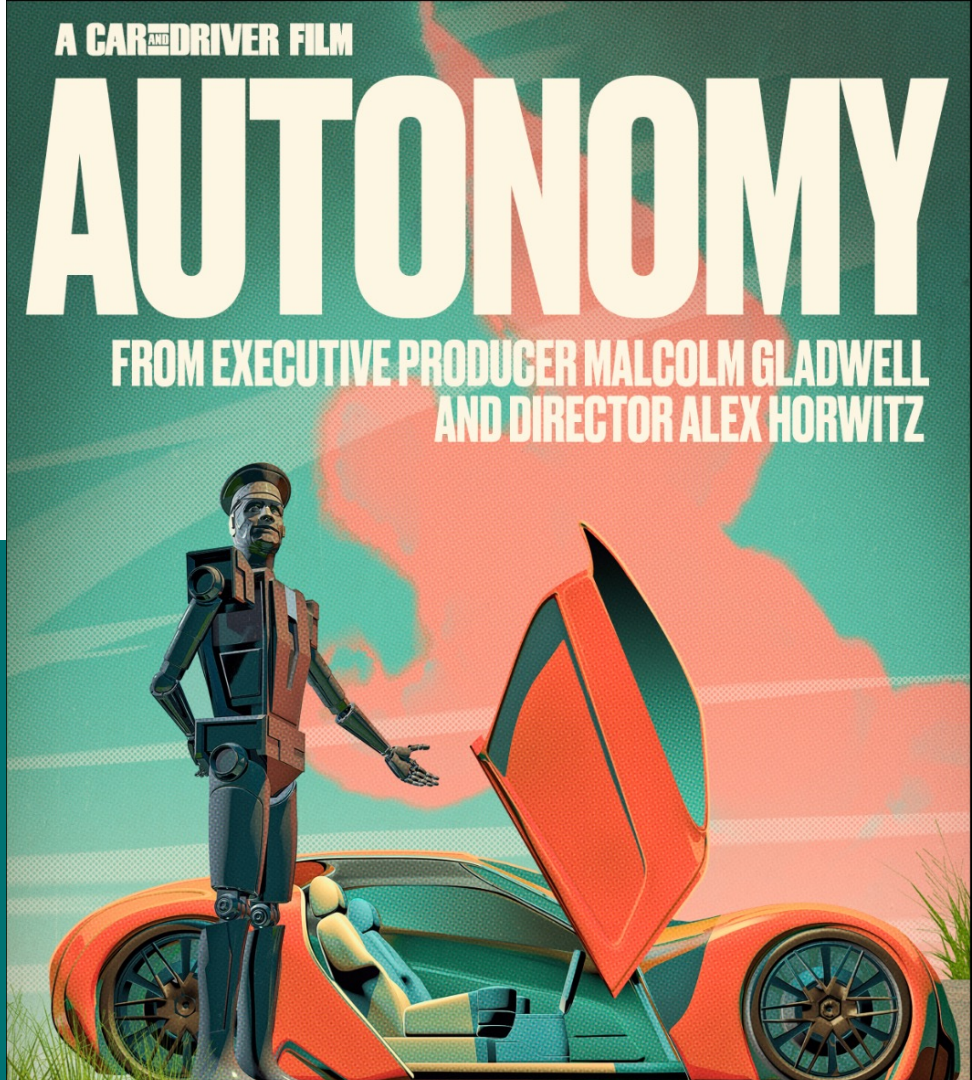
- Vurdering av kontroller
- Operasjonell testing

Etterlevelse / rapportering

- Overvåking
- Tilbakemelding
- Manuelt

Hensikt / målsetting

- Begrense merarbeid for de autonome teamene
- Minimere nødvendig informasjon sendt til GRC system (Service Now)
- Fokusere på informasjon som er viktig for etterlevelse
- Prioritere en fungerende informasjonsstrøm igjennom hele verdikjeden
- Gjør det attraktivt og enkelt å utvide til flere strømmer og team



Metode / Arkitekturprinsipper

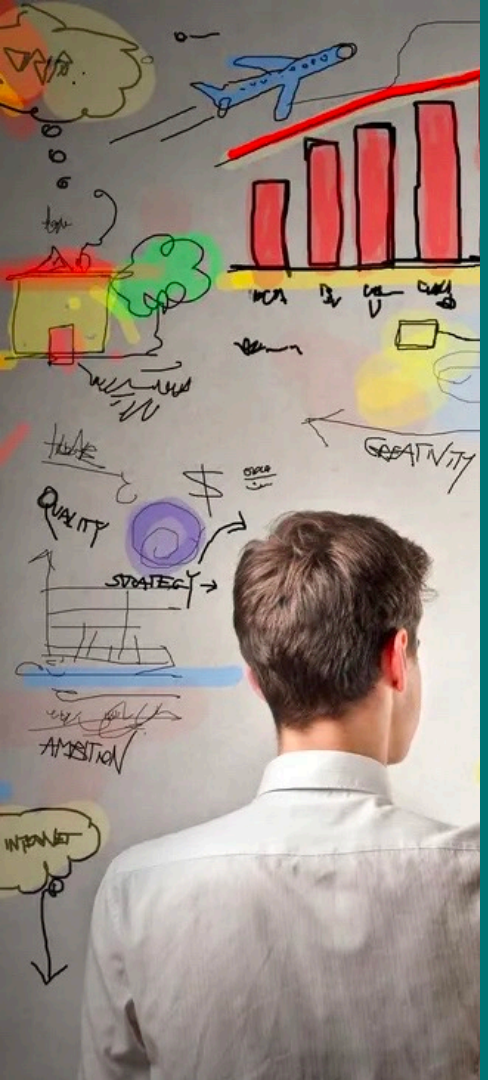
Hent inn data fra «én sann kilde»

Kontroller en gang, etterlev mange

«Control once, comply many»

Koble indikatorer og relevante kontroller til riktige forretnings applikasjoner eller tjenester.

Rapporter i henhold til forventninger og krav fra regulerings-myndigheter



Hvorfor skal vi løse denne floken?

(Start with why
– Simon Sinek)

Hvorfor er dette viktig?

- Programvareutvikling må være sikker
- Forretning må bevise at utviklingen er sikker

Synlig sikkerhet

- Endringstakten øker stadig
- Rammebetingelser endres i økende takt
- Forventningene for økt endringstakt stiger

Alt er i endring

- Evnen til å endre seg er en forretningsfordel
- Etterlevelse er en forretningsfordel

Forretnings-verdi

Hva kan gjøres?

- Finne relevant informasjon om prosesser og utvikling
- Bringe informasjonen videre i stegene til verdikjedene
- Omsette informasjonen slik at sammenheng og relevans blir konkretisert og målbar
- Fore informasjonen tilbake via risikoer, slik at styringen av prosessene blir aktiv og riktig.
- Sette scenen for utvikling med riktig og viktig informasjon

Hvordan / hypotese

Prinsipper som bør brukes;

- Bruke den kilden til informasjon som er mest riktig. “Single source of truth”
- Stol på kilden til informasjonen (f.eks. utviklere, produkteiere etc)
- Bruke sentraliserte kontrollmekanismer
- Gjør kontrollpunktene desentraliserte, både i teknisk implementasjon, og ansvar
- Overvåk først og fremst, for så å informere om svakheter, og kontroller bare ved kritiske nivåer
- Lag prosessene og prosedyrene automatiske og transparente – vi trenger tillit, legitimitet og forståelse i alle nivåer

Hvordan – CI/CD

CI/CD flow to be

Business Owner

Security & governance policies

Jira
BA#, SO#
Application profile

Infrastructure as code testing

Code testing

Application security testing
SAST/DAST

GitLab tools

GitLab pipes

GitLab + SN

Service Now

Incident MGMT

SGF
Security Governance Framework

CMDB
Configuration Management Data Base

Change Velocity

Risk register
BA#, SO#

DevOps

SN Control/ indicators

Reporting / Dashboards

Image

Library / function reg

Vulnerability Reporting
Grafana / SN / AWS Inspector 2

Hvordan – CI/CD

CI/CD flow with Policy

Business owner will state the requirements according to regulations for the specific application type

Business Owner

IT Owner

Governance

IT Owner will choose the control policy

Policy register

Policy Log

Policy engine

Policy enforcement point

Security & governance policies

Jira
BA#, SO#
Application profile

Infrastructure as code testing

Code testing

Application security testing
SAST/DAST

Service Now

Incident MGMT

SGF
Security Governance Framework

CMDB
Configuration Management Data Base

Change Velocity

Risk register
BA#, SO#

DevOps

SN Control/ indicators

Reporting / Dashboards

GitLab tools

GitLab pipes

GtiLab + SN

Policy is loaded into GitLab as templates and patterns, and prompts correct security checks.

Image

Bugs

Library / function reg

Vulnerability Reporting
Grafana / SN / AWS Inspector 2

Hvordan – CI/CD

CI/CD flow with Policy

Business owner will state the requirements according to regulations for the specific application type

Policy is loaded into GitLab as templates and patterns, and prompts correct security checks.

Business Owner / IT Responsible (RISK)

Put into context and RISK

Record and aggregate

Collect data and state

Secure - dev-biz-ops

Give instructions and requirements

Business Owner

Governance

IT Owner will choose the

Policy register

Security & governance policies

Jira BA#, SO# Application profile

Infrastructure as code testing

Code testing

Application security testing

GitLab pipes

GtiLab + SN

Incident MGMT

SGF Security Governance Framework

Reporting / Dashboards

Image

Bugs

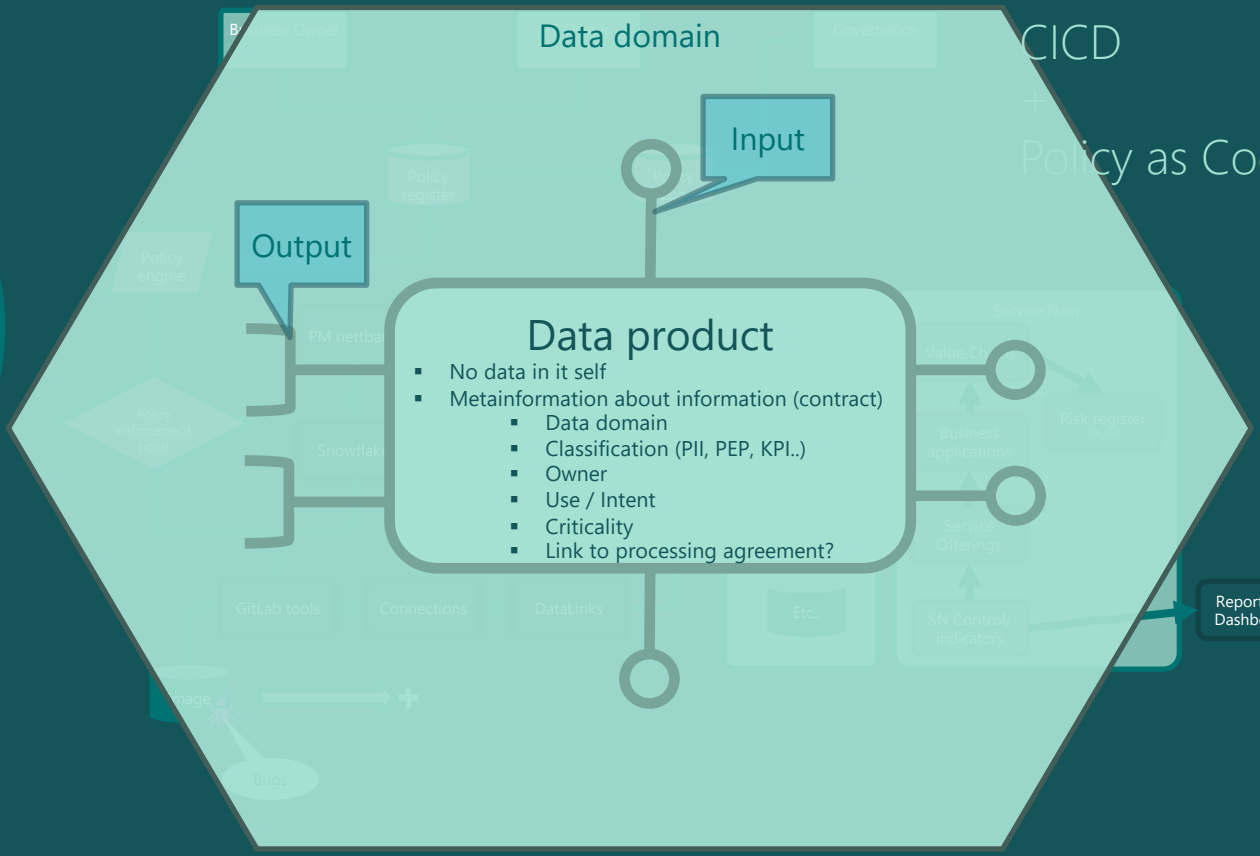
Inspector 2

Hva med informasjonen vår?

Vi må vite hva vil skal sikre.
Tenk om vi kunne vite hvor hvilken informasjon
var til en hver tid?

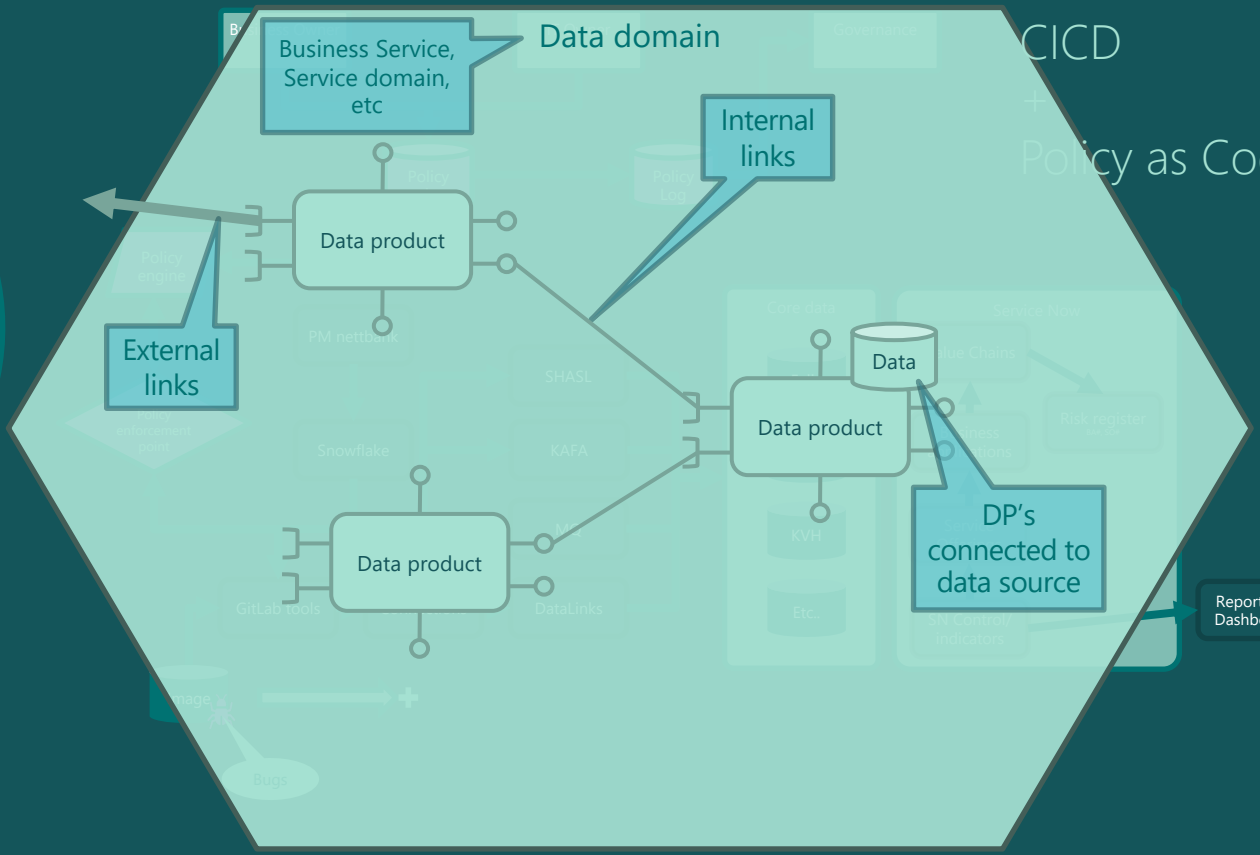
How – Data Products

Data products
+
CICD
+
Policy as Code



How – Data Products

Data products
+
CICD
+
Policy as Code



How – Data Products

Data products
+
CICD
+
Policy as Code

Business Service, Service domain, etc

Business Owner

IT Owner

Governance

Policy register

Policy Log

Probably several domains

Policy engine

Data domain

PM nettbank

Policy enforcement point

Datalake

SHASL

KAFA

MQ

GitLab tools

Connections

DataLinks

Image

Bugs

Core data
Data domain(s)

Core

Kunde

Datalake

Etc..

Service Now

Value Chains

Business applications

Service Offerings

SN Control/ indicators

Risk register
BA#, SD#

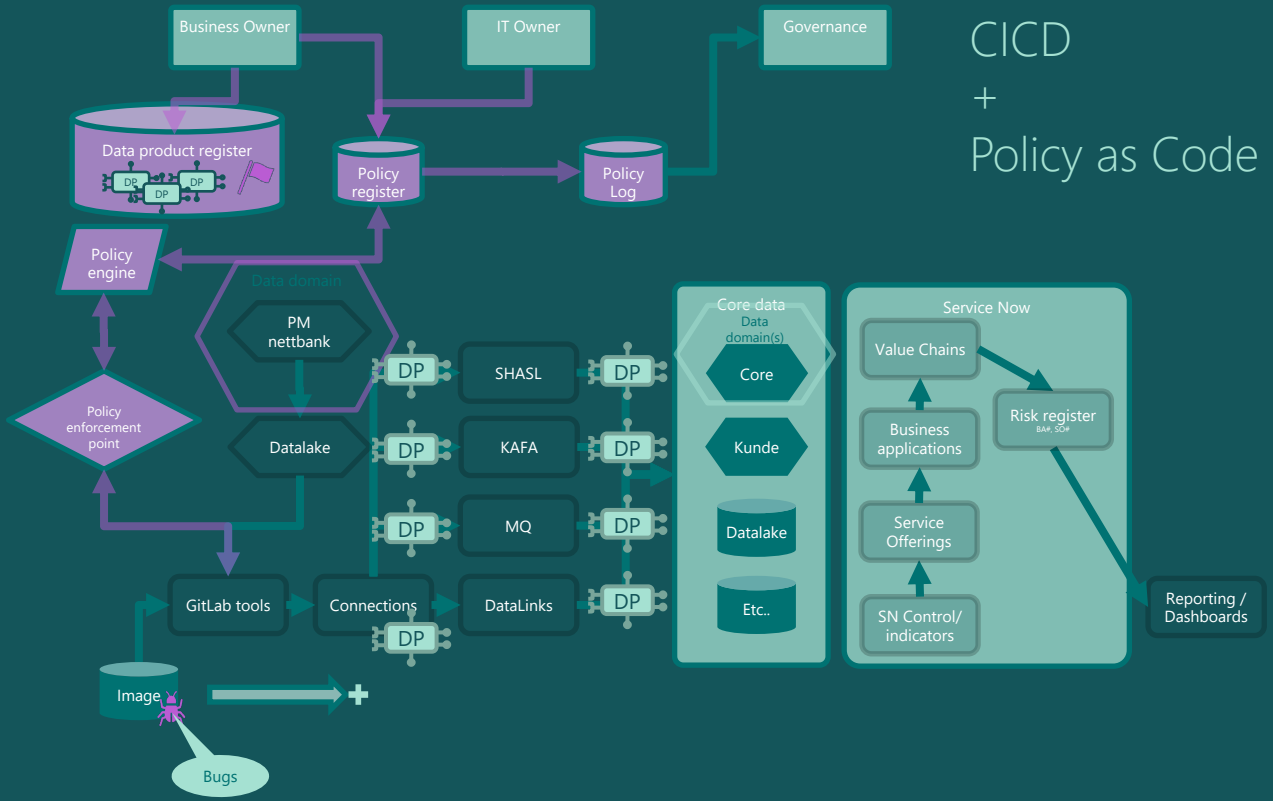
Relationship?
Data domain, Business Service, Service domains, Valuechains,

Reporting / Dashboards

DNB

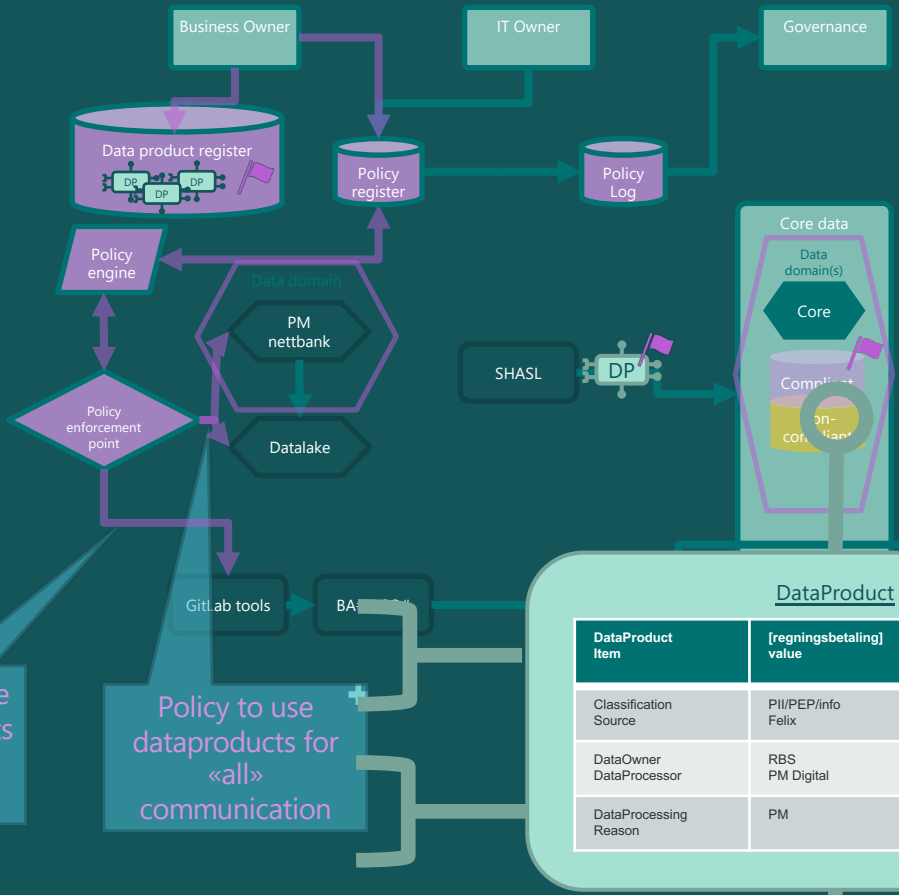
How – Data Products

Data products
+
CICD
+
Policy as Code



How – Data Products

Data products
+
CICD
+
Policy as Code



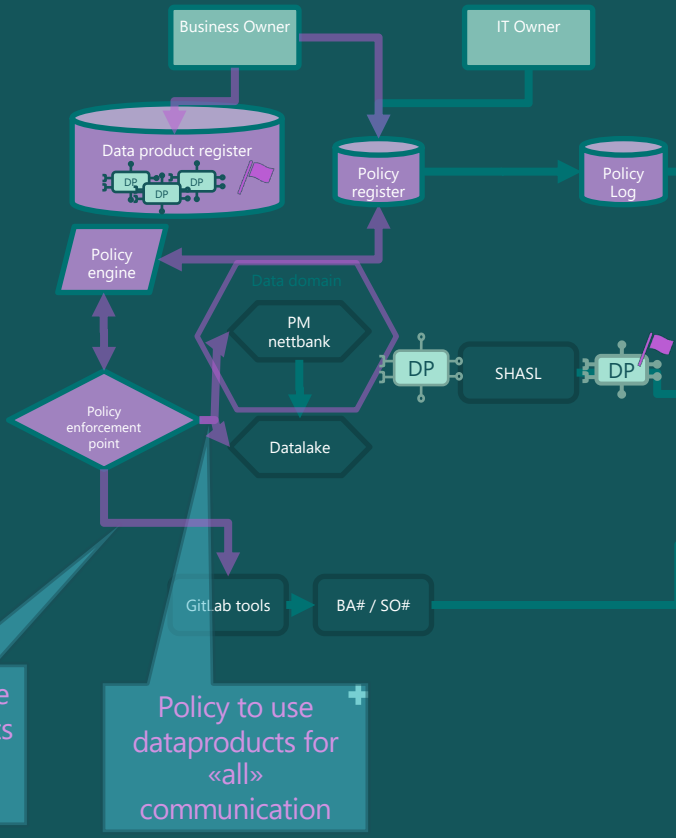
DataProduct

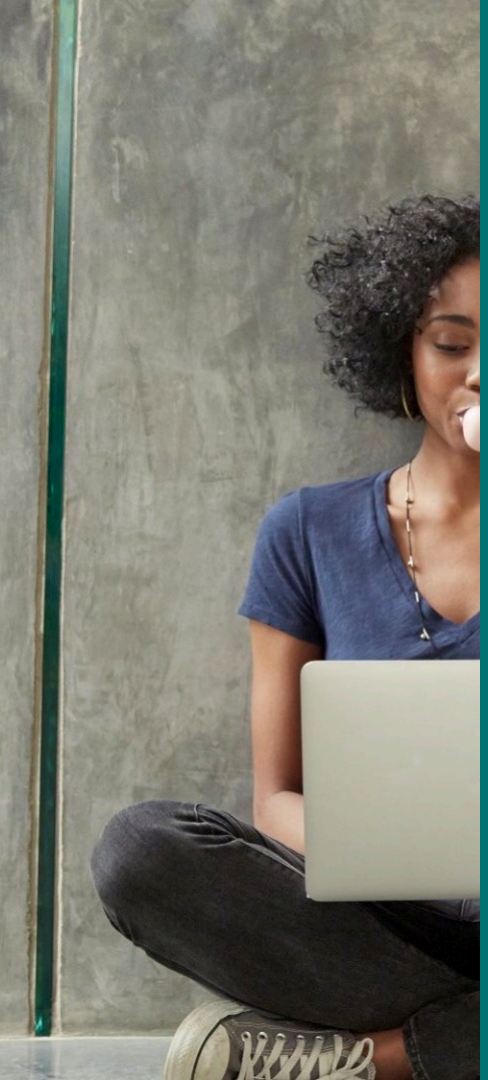
DataProduct Item	[regningsbetaling] value	[dpregid#] Link/ref/DataProduct Register
Classification Source	PII/PEP/info Felix	
DataOwner DataProcessor	RBS PM Digital	
DataProcessing Reason	PM	GDPR

DNB

How – Data Products

Data products
+
CICD
+
Policy as Code





Gevinster

Nesten for mange til å ta for seg..

Forebyggende / Omgående

Operasjonelt / dag til dag

Kontinuerlig utvikling / tilbakemelding

Etterlevelse / rapportering

Forebyggende / Omgående

Operasjonelt / dag til dag

Kontinuerlig utvikling / tilbakemelding

Etterlevelse / rapportering

DevSecOps

Vi vil vite hvilke krav som gjelder for utvikling
Vi vil vite hva vi har I alle våre systemer – til en hver tid

ITSM - CMDB

Avhengigheter og ansvarlig vil bli registrert og oppdatert fortløpende

OPS

Når det brenner på dass, vil vi vite hvilken dass som brenner, og hvor stor brannen er.

Forebyggende / Omgående

Operasjonelt / dag til dag

Security
Krav til sikkerhet kan oppdateres og prøves automatisk

Sårbarheter
Vi vil vite hvilke komponenter som er i produksjon, og slipper å søke etter hvor vi har utfordringer

Change
Konsekvenser av endringer kan modelleres

Kontinuerlig utvikling / tilbakemelding

Etterlevelse / rapportering

Forebyggende / Omgående

Operasjonelt / dag til dag

Kontinuerlig utvikling / tilbakemelding

Generelt

Jo mer vi trykker igjennom vår CI/CD – jo mer oppdatert og compliant blir vi!

Informasjon

Med kontroll på hvilke informasjon som finnes hvor, vil «Lineage» være oppnåelig

Overvåking

Kan trigges i regelmotorer, for å søke spesifikt etter mønster

Etterlevelse / rapportering

Forebyggende / Omgående

Operasjonelt / dag til dag

Kontinuerlig utvikling / tilbakemelding

Etterlevelse / rapportering

Risiko

Risiko kan genereres og aggregeres automatisk

Ansvarsmodellen

Vil være reell, siden informasjonen er til å stole på

Dynamikk

Ny krav og regler kan implementeres når som helst og overalt

Takk for oppmerksomheten

Frode