

En *Nais* app!

Hvordan ser *egentlig* en
nais app ut?



Notabene

Alle kodeeksempler i sliden er tatt fra <https://docs.nais.io>

- Når noe blir oppdatert(utdatert?), så vil "latest & greatest" være å finne der!

En *gyllen sti*

Naisplattformen er *opinionated*

Mål: å lette *kognitiv last* & *akselerere* utvikleropplevelsen

- -> fra `git commit` til produksjon

Typiske *nais* apps vi ser

1. APler, á la REST/gRPC/etc, for eksempel foran en DB

Typiske *nais* apps vi ser

1. APler, á la REST/gRPC/etc, for eksempel foran en DB
2. Frontend apps, serverer kun html/js/css/lignende
 - i. Backends-For-Frontend

Typiske *nais* apps vi ser

1. APler, á la REST/gRPC/etc, for eksempel foran en DB
2. Frontend apps, serverer kun html/js/css/lignende
 - i. Backends-For-Frontend
3. Queue-worker, leser og agerer på kø/DB

Typiske *nais* apps vi ser

1. APIer, á la REST/gRPC/etc, for eksempel foran en DB
2. Frontend apps, serverer kun html/js/css/lignende
 - i. Backends-For-Frontend
3. Queue-worker, leser og agerer på kø/DB
4. Proxies, eksempelvis for noe on-prem

Zero-Trust / tjenestesegmentering

Koblinger utføres direkte & eksplisitt

1. [Workload identity](#)
2. [AccessPolicies](#)
 - [Default outbound allowList](#)
 - [Ingresser == "åpen port inn via URL på nettverksnivå!"](#)

```
spec:
  accessPolicy:
    inbound:
      rules:
        - application: otherApp
          permissions:
            roles:
              - custom-role
            scopes:
              - custom-scope
    outbound:
      external:
        - host: ekstern-app.eksempel.no
        - host: 1.2.3.4
        ports:
          - port: 9200
      rules:
        - application: newApp
          cluster: tenant-dev
          namespace: fancyTeam
```

[Frode](#) skrev en forklarende bloggpost om dette tilbake i 2020 🙄!

Zero-Trust / tjenestesegmentering

Koblinger utføres direkte & eksplisitt

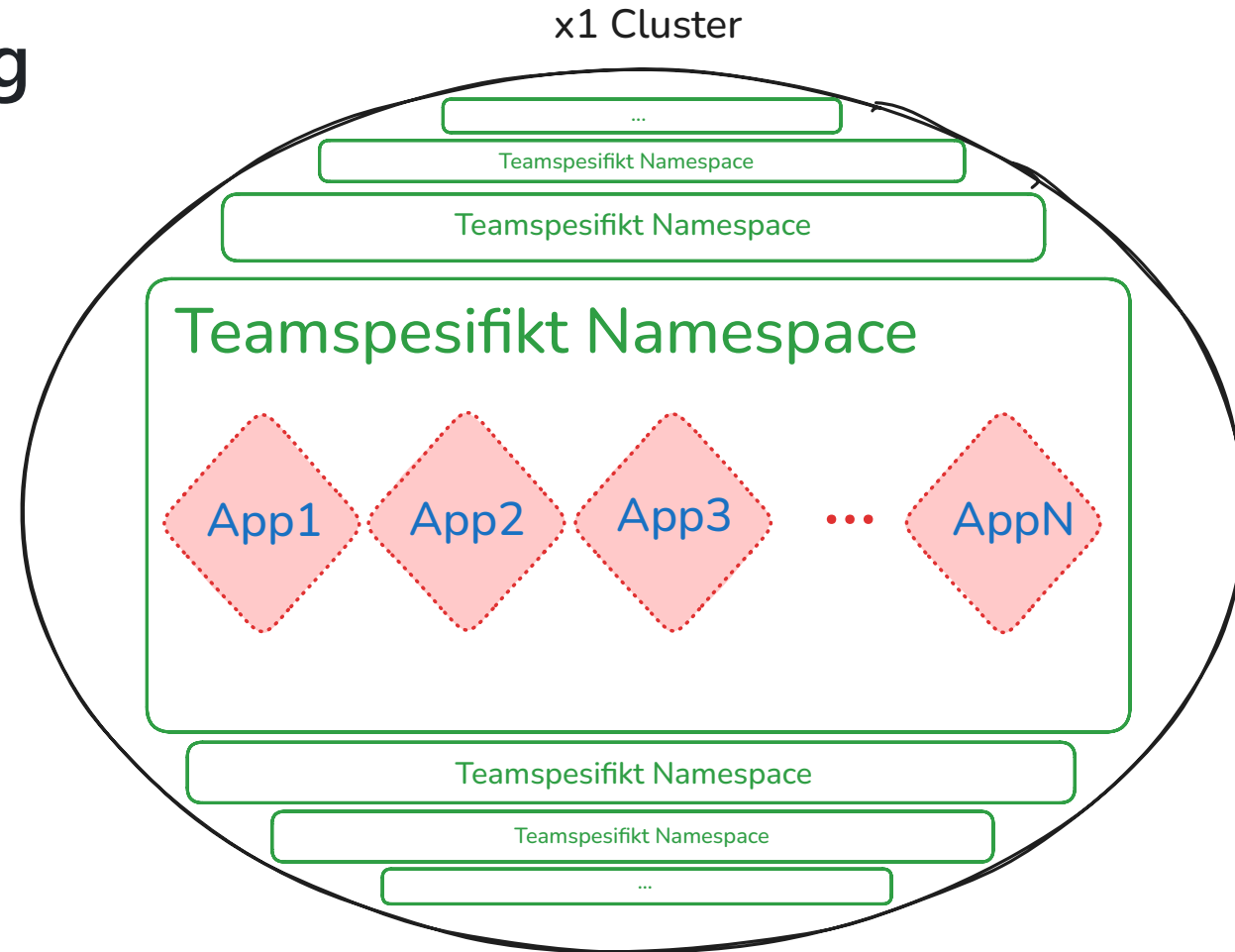
3. Apps deploys *segregert**

$\forall \text{app} \exists \text{team},$

$\forall \text{team} \exists \text{namespace} \in \text{cluster},$

$\forall \text{app} \exists \text{team} \text{Namespace} \in \text{cluster}$

- 1x namespace for *alle* teamets apper per cluster



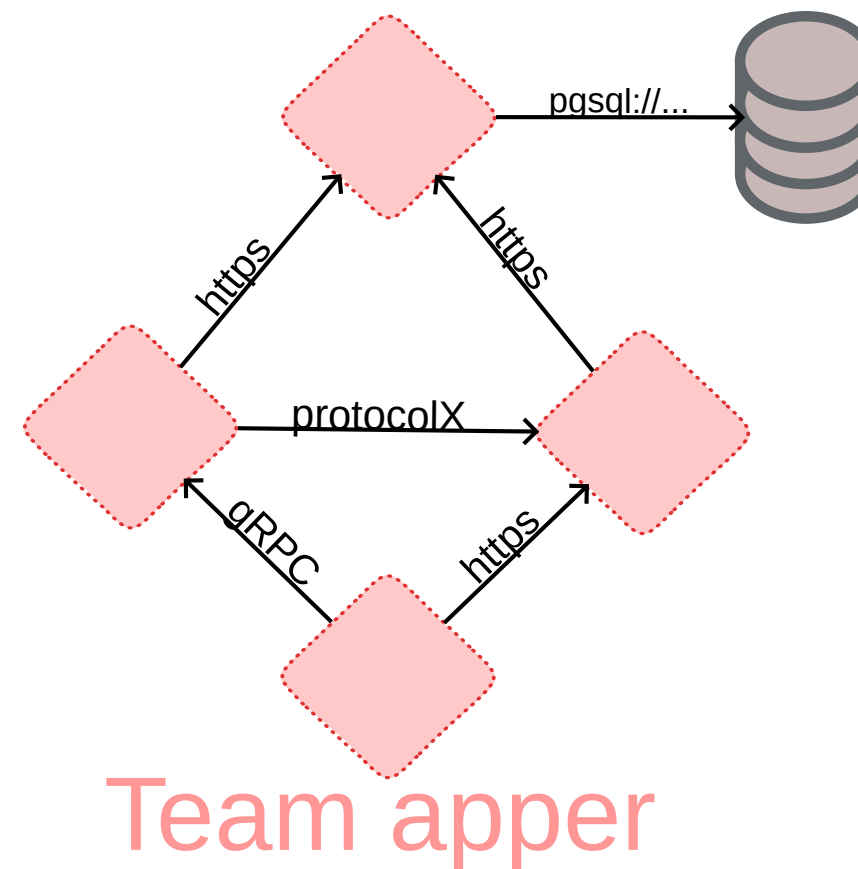
Autentiseringsmekanismer *Nais* tilbyr

- LoginProxy
 - Kan også sikre OIDC compliant Autentisering på vegne av appen!

Og på sikt enda fler 😊, følg med i [#nais-announcements](#)

Hva forventer plattformen av en *nais* app?

1. Ingen *delte* databasetilkoblinger på tvers av *nais* apps ❌
2. Eksplisitte koblinger mellom *nais* apps/tjenester ✅



Hva forventer plattformen av en *nais* app?

1. At man følger *cloud native* ledestjerner[1]

Se også [Good practices](#) hos nais doc'en! 😊

- En fin test: takler tjenesten at appen din
 - a. én pod (gjernes blant flere) blir drept
 - b. hvert kvarter?

[1]: Det ble inspirert en gang av (nå litt utdatert?) [12-factor apps](#)!

Hva forventer plattformen av en *nais* app?

1. Containeren din sitt innhold kan du få styre *helt selv*!
2. Containeren er utviklerens grensesnitt å forholde seg til
 - i. Styrt av `nais.yml` og **Nais Console**

```
apiVersion: nais.io/v1alpha1
kind: Application
metadata:
  labels:
    firstLabel: value1
    secondLabel: value2
  name: myapplication
  namespace: myteam
spec:
  ...
```

Datatjenester *Nais* tilbyr

Datatjenester *Nais* tilbyr

1. Trenger du en relasjonsdatabase (OnLine Transactional Processing)?
 - PostgreSQL

```
...
spec:
  postgres:
    cluster:
      majorVersion: "17"
    resources:
      cpu: 100m
      diskSize: 1Gi
      memory: 2G
```

Datatjenester *Nais* tilbyr

1. Trenger du en relasjonsdatabase (OnLine Transactional Processing)?
 - PostgreSQL
2. Dataanalyse (OnLine Analytical Processing)?
 - Google BigQuery

```
...  
spec:  
  gcp:  
    bigQueryDatasets:  
      - name: <MY-DATASET>  
        permission: READ  
        # permission: READWRITE
```


Datatjenester *Nais* tilbyr

1. Trenger du en relasjonsdatabase (OnLine Transactional Processing)?
 - PostgreSQL
2. Dataanalyse (OnLine Analytical Processing)?
 - Google BigQuery
3. Statistiske webressurser?
 - CDN

```
name: Push to CDN
on:
  push:
jobs:
  upload:
    runs-on: ubuntu-latest
    permissions:
      contents: 'read'
      id-token: 'write'
    steps:
      - uses: 'actions/checkout@v4'
      - name: Upload static files to CDN
        uses: nais/deploy/actions/cdn-upload/v2
        with:
          team: <team name>
          source: path/to/assets/
          destination: <contextPath -> /my-app/dist>
```

Datatjenester *Nais* tilbyr

4. Når du vil ha en kø?

- Kafka
 - Topic

```
...
spec:
  kafka:
    pool: <MY-POOL>
---
apiVersion: nais.io/v1alpha1
kind: Topic
metadata:
  name: <MY-TOPIC>
spec:
  pool: <MY-POOL>
  acl:
    - application: <APPLICATION-NAME>
      # Applications named <APPLICATION-NAME>
      # from <SOME-TEAM> team
      team: <SOME-TEAM>
      # has read
      access: write
    - team: <MY-TEAM>
      # All apps belonging to <MY-TEAM>
      # with a name starting with `queue-` prefix
      application: queue-*
      # can write to this topic
      access: readwrite
    - team: <TRUSTED-TEAM>
      # Apps belonging to <TRUSTED-TEAM>
      application: *
      access: read
```

Datatjenester *Nais* tilbyr

4. Når du vil ha en kø?

- Kafka

5. In-memory enkel key/value database?

- Valkey

```
...
spec:
  valkey:
    - instance: $NAME
      access: read
      # access: readwrite
      # access: write
      # access: admin
```

Datatjenester *Nais* tilbyr

4. Når du vil ha en kø?
 - *Kafka*
5. In-memory enkel key/value database?
 - *Valkey*
6. Ustrukturerte data man ønsker å fritekstsøke?
 - *OpenSearch*

```
...
spec:
  openSearch:
    instance: $NAME
    access: read
    # access: readwrite
    # access: write
    # access: admin
```

Friends don't let friends build their own feature flag system[1]

Nais tilbyr **Unleash!**

[1]: <https://www.getunleash.io>

Nais tilbyr full LGTM observability stack

1. Loki

Nais tilbyr full LGTM observability stack

1. Loki
2. Grafana

Nais tilbyr full LGTM observability stack

1. Loki
2. Grafana
3. Tempo

Nais tilbyr full LGTM observability stack

1. Loki
2. Grafana
3. Tempo
4. Prometheus/Mimir

Nais tilbyr alerts

To alternativer listet i [Nais doc'en](#)

- GrafanaAlerts
- PrometheusRule

Her er et PrometheusRule eksempel

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  ...
spec:
  groups:
  - name: nameOfMyAlert
    rules:
    - alert: InstanceDown
      expr: >
        kube_deployment_status_replicas_available{
          namespace="<namespace>",
          deployment="<application name>"
        } == 0
      for: 5m
      annotations:
        consequence: Application is unavailable
        action: >
          `kubectl describe pod <podname>`
          ->
          `kubectl logs <podname>`
        summary: |-
          This is a multi-line summary with
          linebreaks and everything. Here you
          can give a more detailed
          summary of what this alert is about
      labels:
        namespace: <MY-TEAM> # required
        severity: critical
```

Ambisjon: plattformuavhengighet!

- Naisplattformen byr primært på (F)OSS tjenester/integrasjoner
 - i. Kubernetes ➡ OSS ✓
 - ii. LGTM ➡ OSS ✓
 - iii. Alle* datalagringstjenester ➡ OSS ✓*
 - iv. LoginProxy, baserer seg på OIDC ➡ OSS protokoll ✓
 - v. Unleash ➡ OSS ✓

* Utenom Google BigQuery! ⚠

Ambisjon: plattformuavhengighet!

- Naisplattformen byr primært på (F)OSS tjenester/integrasjoner

- i. Kubernetes ➡ OSS ✓

- ii. LGTM 100% OSS ✓

- iii. Alle* datalagringstjenester OSS ✓

- iv. LoginProxy, baserer seg på OIDC ➡ OSS protokoll ✓

- v. Unleash ➡ OSS ✓

➡ Lettere (enn hos cloud-vendor XYZ) å koble seg om til en annen sky!

Spørsmål?

- Husk at [#nais](#) kanalen er alltid tilgjengelig, og du er *ikke* alene om å lure på det du undrer om 😊!

Mange hyggelige folk der som både kan lurere på det samme (?), og gjerne dele svaret de sitter med 💡!