KOGNIC>

Story time!

How We Revolutionized Developer Experience with 3.5 Platform Engineers





the data platform accelerating machine learning for performance-critical applications

Software company, SaaS

Founded 2018 as Annotell
Rebranded 2022 to **Kognic**

HQ in Gothenburg, Sweden

2023 numbers

~100 employees
~50 engineers

$ whoami

Jessica Andersson



Product Area Lead Engineering Enablement

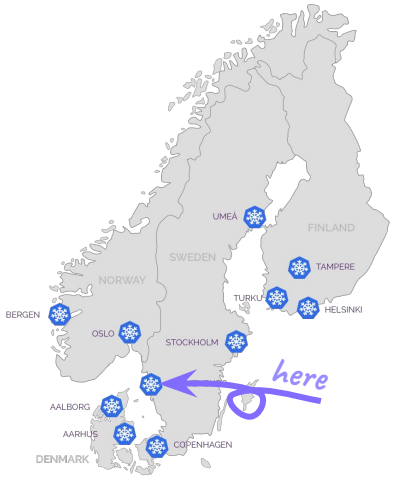
CNCF Ambassador

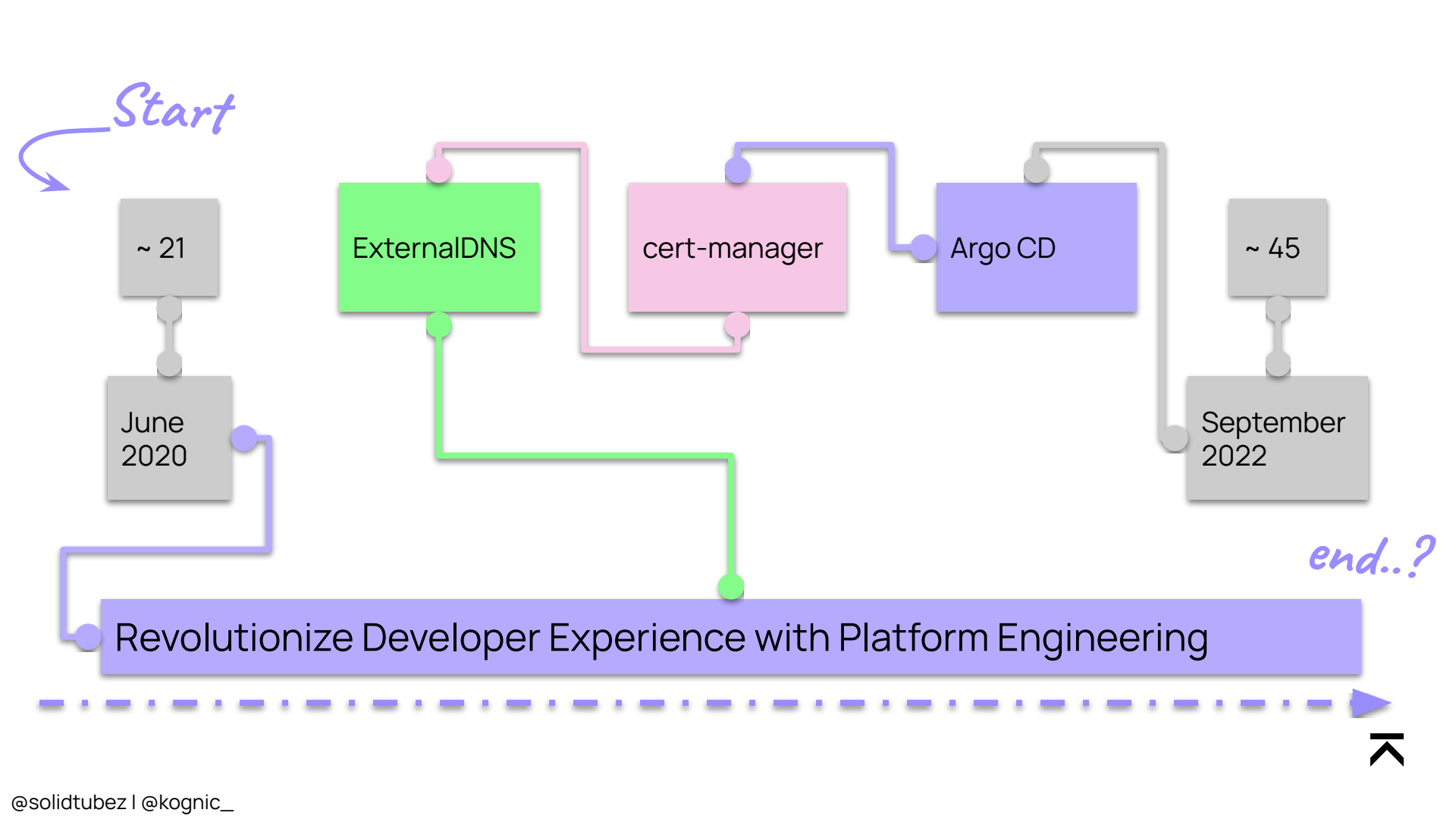
Cloud Native Nordics

Meetup Organizer in Gothenburg

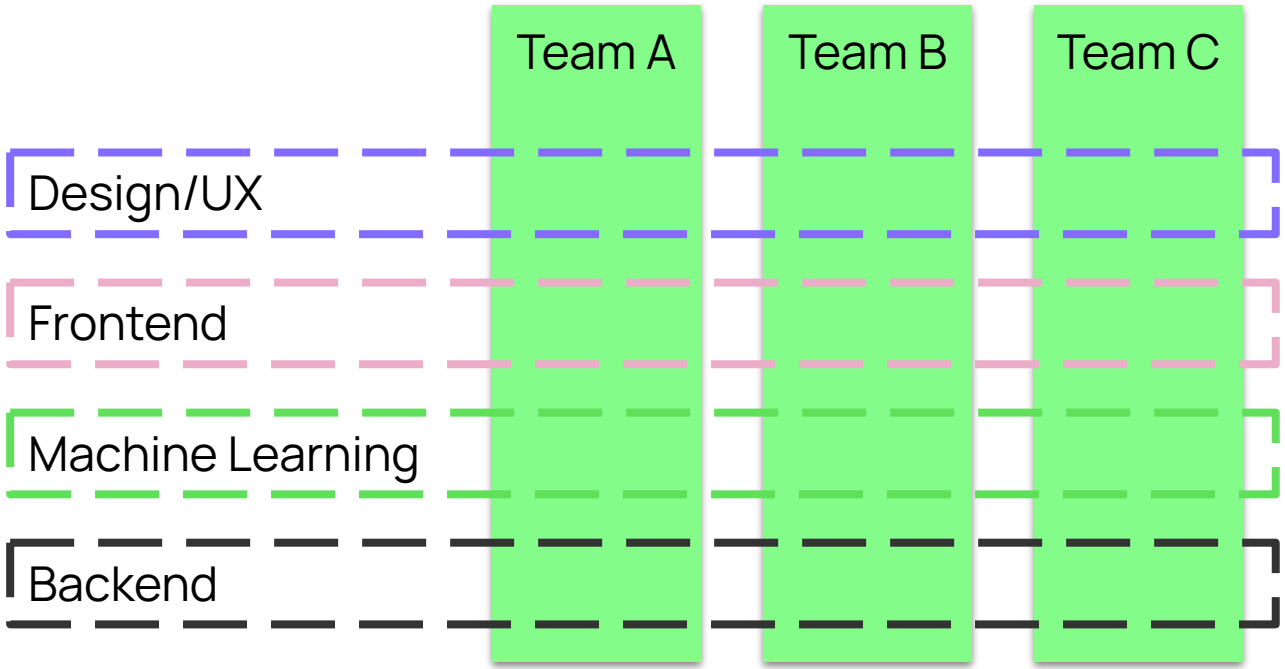


CLOUD NATIVE
NORDICS




Start
~ 21
June 2020
ExternalDNS
cert-manager
Argo CD
~ 45
September 2022
end..?
Revolutionize Developer Experience with Platform Engineering

Cross functional and autonomous teams



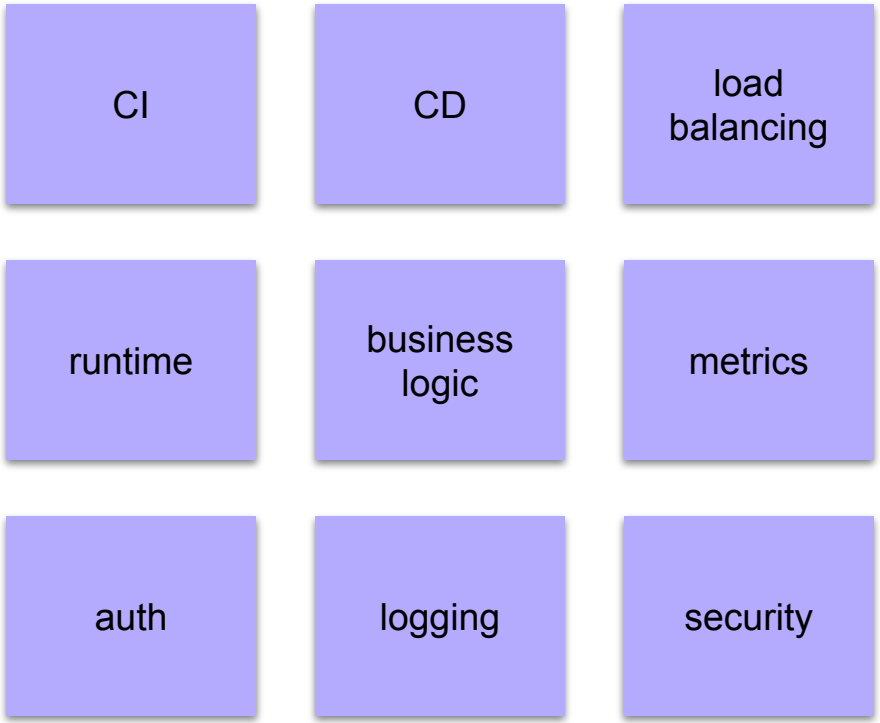
Developer Experience

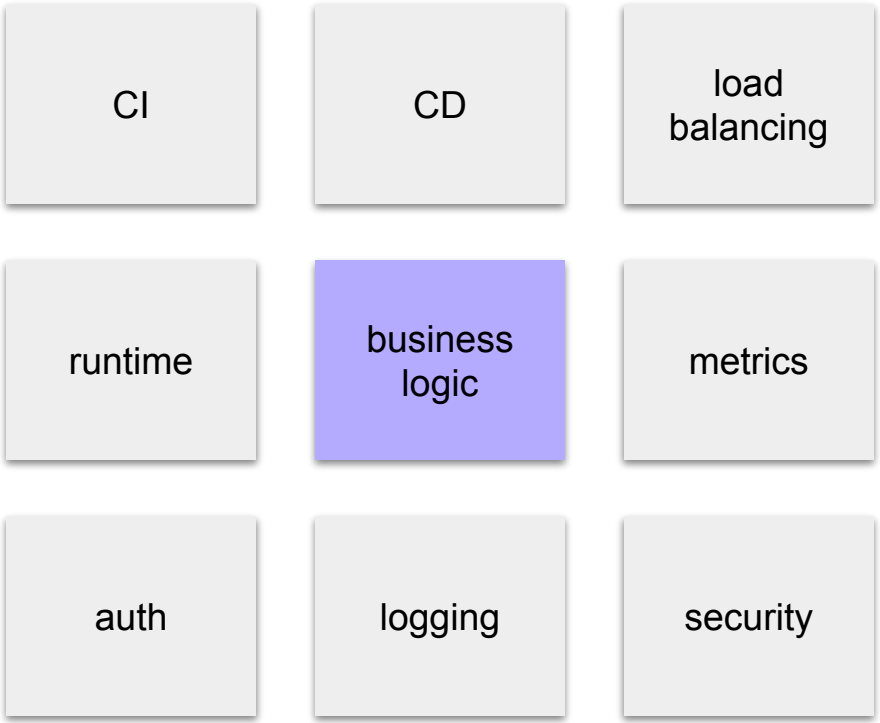


Why do I care about

Developer Experience *?*




CI
CD
load balancing
runtime
business logic
metrics
auth
logging
security


CI
CD
load balancing
runtime
business logic
metrics
auth
logging
security

Golden Path / Paved Road

A platform that solves **common needs** allows developers to **focus** on what is **important**



Photo by Christian puta on Unsplash

80 %



Platform as a Product



Platform Engineering

Engineering Enablement



Why do we have developers?



Business value



How does being a small team impact?



Uncomfortable insights

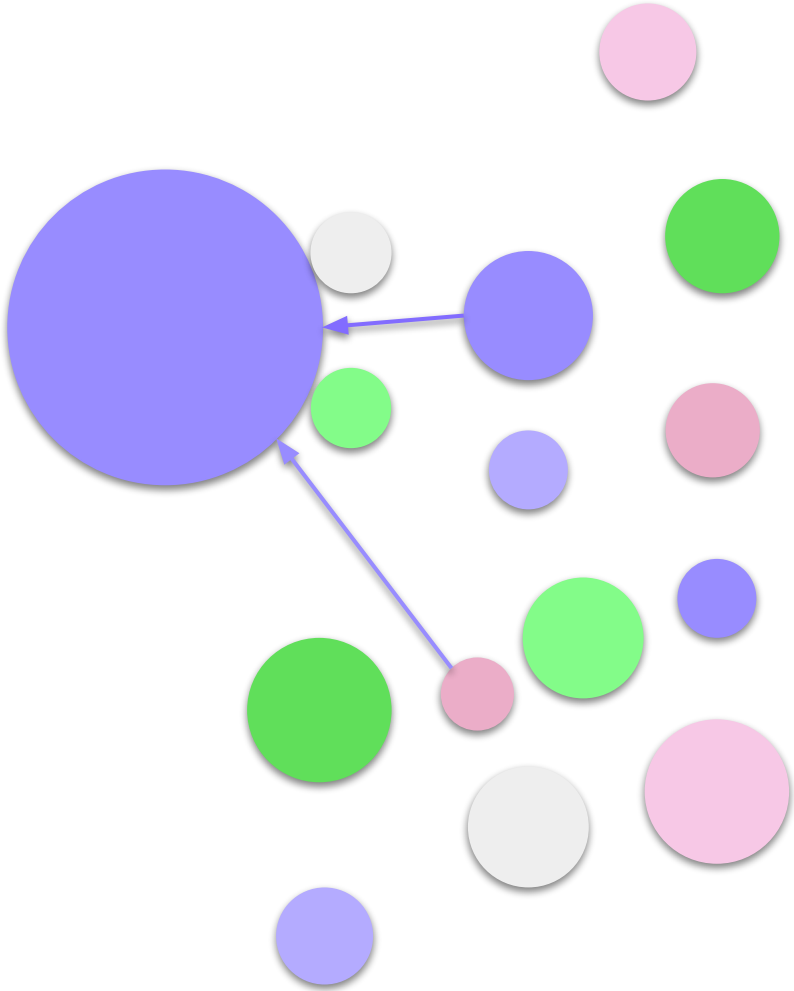
1. You can't do **all the things**
2. Valuable to **not** do something
3. Upgrades are **frequent** and **take time**
4. You don't need a Service Mesh (probably…)



Photo by Yaniv Knobel on Unsplash

Assess → Act

Slow effect magnet



So what’s our strategy?



Remove time consuming tasks and bottlenecks



1: Unblocking the developers



2: Removing time consuming tasks from the team



3: Removing time consuming tasks from the developers



1: Unblocking the developers

2: Removing time consuming tasks from the team

3: Removing time consuming tasks from the developers

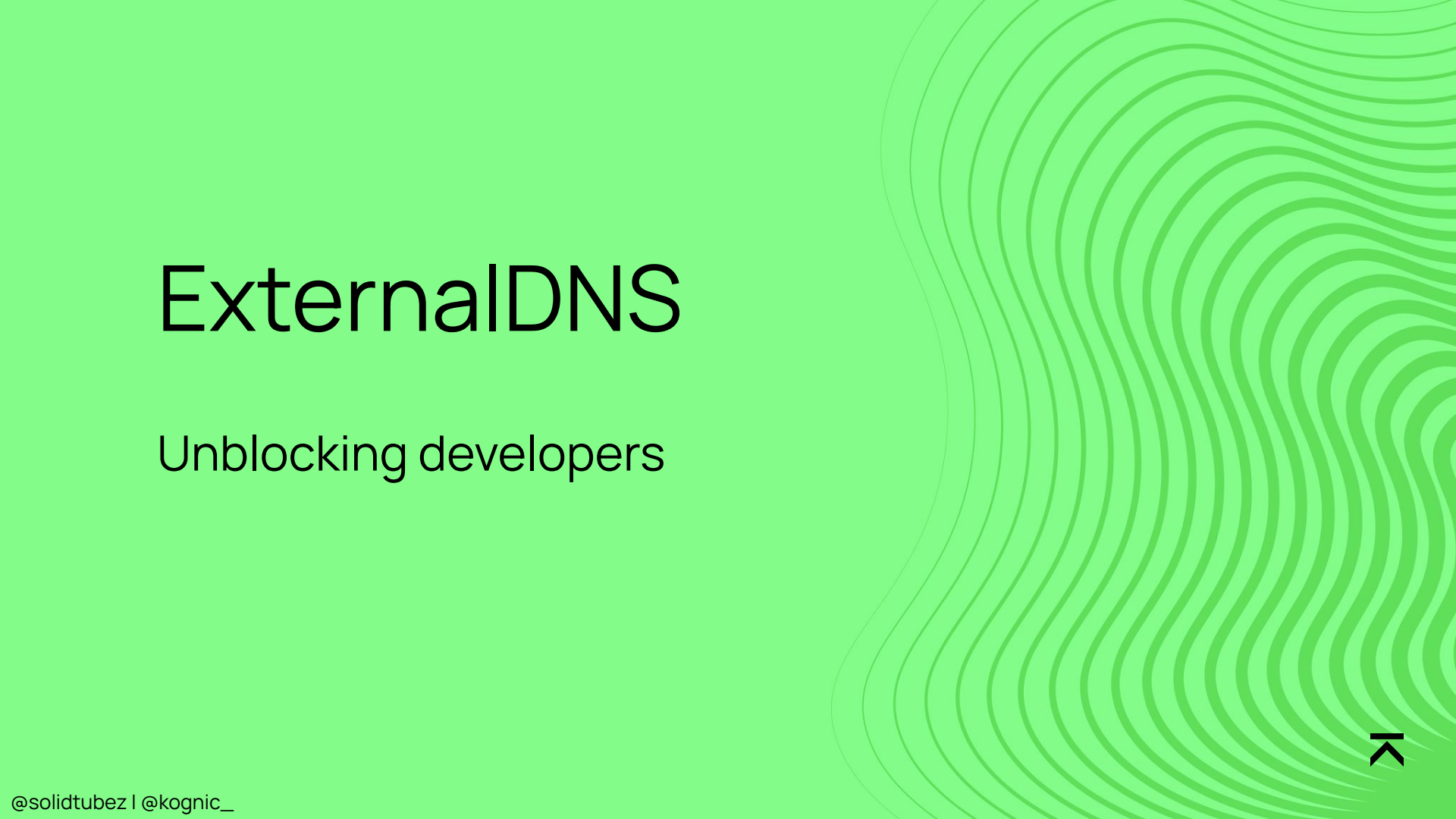


Some real examples!



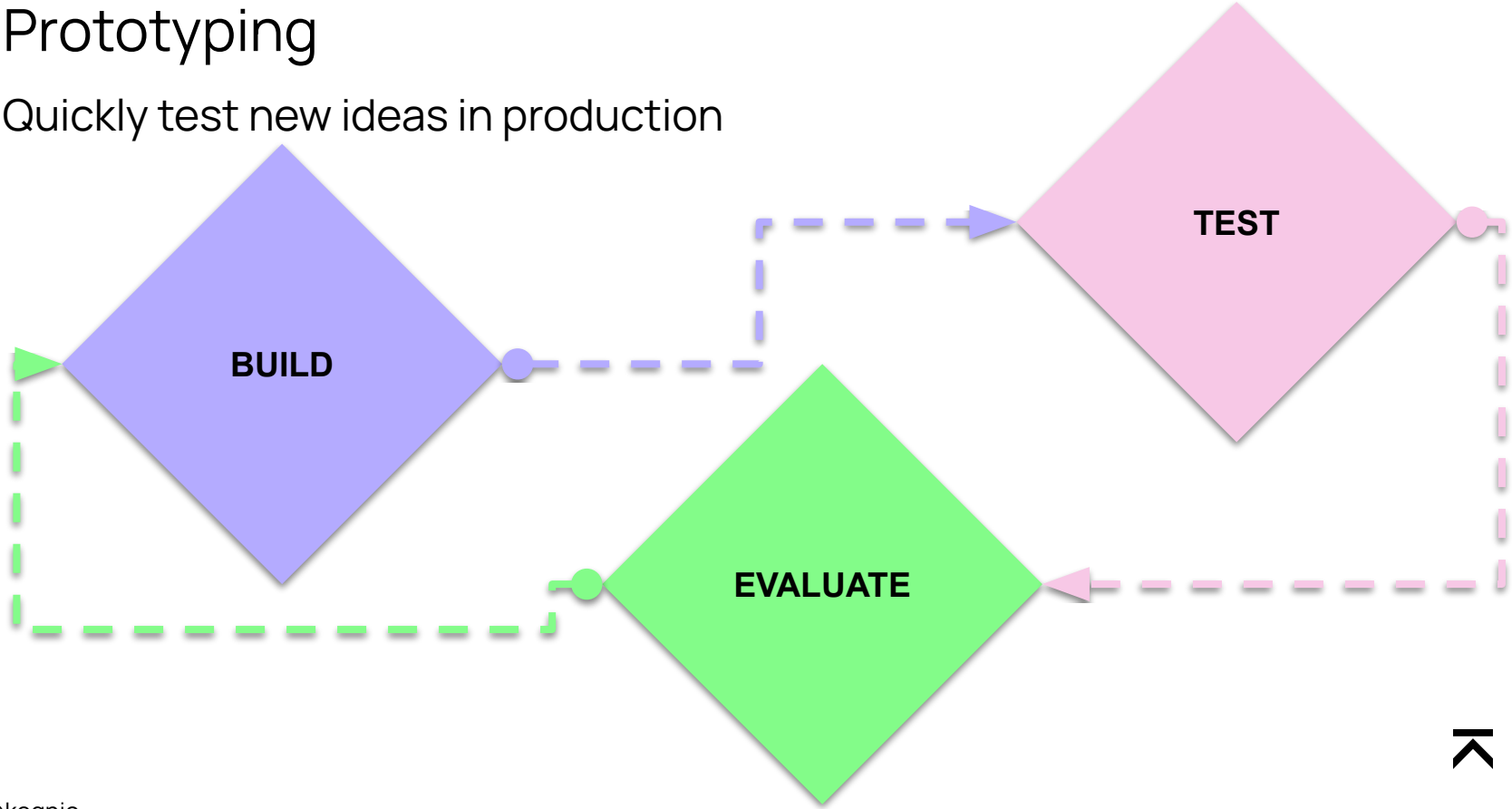
ExternalDNS

Unblocking developers



Prototyping

Quickly test new ideas in production



DNS allocation process before

1. Build app
2. Deploy app → get IP
3. Ask for DNS
4. Wait ←––– BLOCKER
5. Get DNS record
6. Keep developing



github.com/kubernetes-sigs/external-dns



Easy to install

- Cloudflare API token → running

Tracks ownership with the help of TXT record

Type ▲	Name	Content
A	hello-argo	[redacted]
TXT	hello-argo	"heritage=external-dns,extern...

One time investment - migration

- **Figure out what we have**



One time investment - migration

- **Figure out what we have**
- **Annotate services with the right DNS**

```
apiVersion: v1
kind: Service
metadata:
 annotations:
   external-dns.alpha.kubernetes.io/hostname: hello-argo.domain.com
 name: hello-argo-01
```



One time investment - migration

- Figure out what we have
- Annotate services with the right DNS

```
apiVersion: v1
kind: Service
metadata:
 annotations:
   external-dns.alpha.kubernetes.io/hostname: hello-argo.domain.com
 name: hello-argo-01
```

- Fake TXT records in Cloudflare

```
"heritage=external-dns,external-dns/owner=prod,external-dns/resourc
e=service/namespace/hello-argo-01"
```



One time investment - migration

- Figure out what we have
- Annotate services with the right DNS

```
apiVersion: v1
kind: Service
metadata:
 annotations:
   external-dns.alpha.kubernetes.io/hostname: hello-argo.domain.com
 name: hello-argo-01
```

- Fake TXT records in Cloudflare

```
"heritage=external-dns,external-dns/owner=prod,external-dns/resourc
e=service/namespace/hello-argo-01"
```

- Deploy ExternalDNS in cluster

One time investment - migration

- Figure out what we have
- Annotate services with the right DNS

```
apiVersion: v1
kind: Service
metadata:
 annotations:
   external-dns.alpha.kubernetes.io/hostname: hello-argo.domain.com
 name: hello-argo-01
```

- Fake TXT records in Cloudflare

```
"heritage=external-dns,external-dns/owner=prod,external-dns/resourc
e=service/namespace/hello-argo-01"
```

- Deploy ExternalDNS in cluster
- Success!

DNS allocation process after

1. Build app
2. Deploy app → get IP
3. Ask for DNS
4. Wait ←––– BLOCKER
5. Get DNS record
6. Keep developing



DNS allocation process after

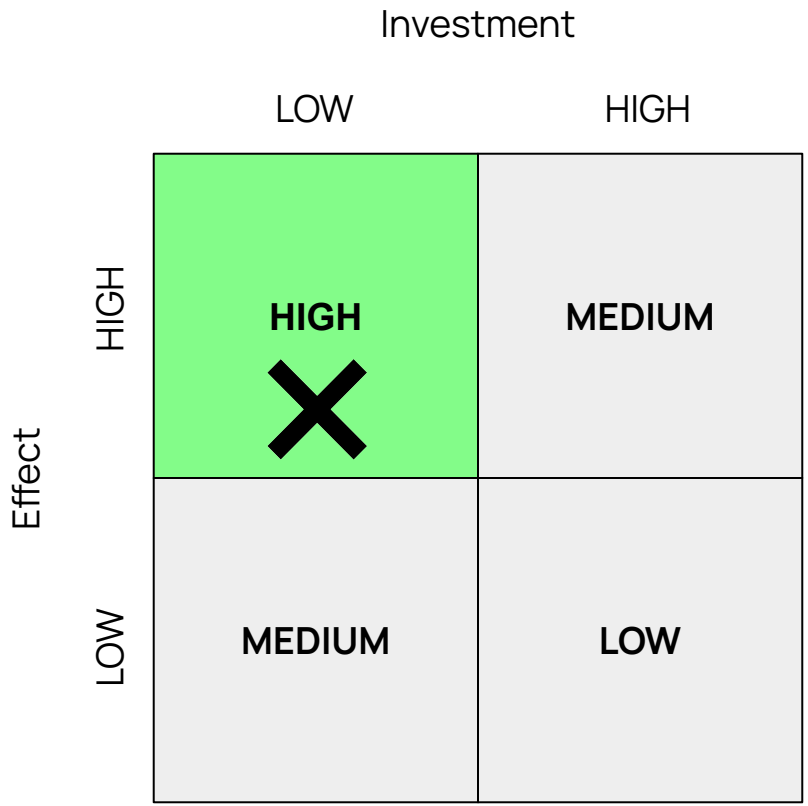
1. Build app
2. Deploy app ~~→ get IP~~
3. ~~Ask for DNS~~
4. ~~Wait ← BLOCKER~~
5. Get DNS record
6. Keep developing

1. Build app
2. Deploy app with annotation
3. Get DNS record
4. Keep developing



ROI

- **Self serve**
- **Easy to use**



cert-manager

Removing time consuming tasks from the team

Certificate renewal process before

1. Use custom shell scripts in local terminal to create new certificates
2. Manually edit secrets in kubernetes to contain the new certificates
3. Rolling restart all deployments

https://cert-manager.io/docs/




```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- https://github.com/cert-manager/cert-manager/releases/download/v1.10.0/cert-manager.yaml
- clusterissuer.yaml
```

- Set up cluster issuer to connect to letsencrypt + Cloudflare
- Can track multiple DNS zones
- Plays nice with Ingress resources

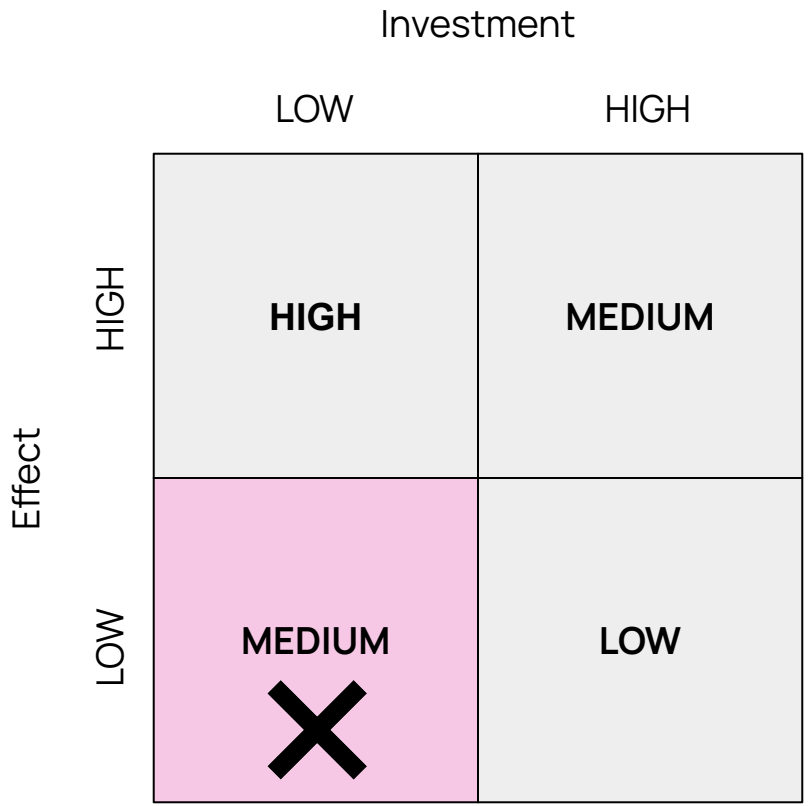
Certificate renewal process after

1.



ROI

- **Removed manual process**
- **Less cognitive load**
- **No rolling restart of all the things!**



Argo CD

Removing time consuming tasks from developers



Deployment process before

- Build pipeline posted to slack channel
  - Pick up new image tag from slack
- Run script locally with new image tag and application name
  - kubectl set image
  - kubectl rollout status
  - Post to slack



Problems

- Black box
- Hard to debug
- Lacking version control of kubernetes resources
  - What is deployed and why?
  - Changing configuration scary
  - No way to restore

Needs

- Transparency
- Less time spent debugging failures
- Resilience



GitOps

1. Declarative
2. Versioned and Immutable
3. Pulled Automatically
4. Continuously Reconciled

https://opengitops.dev/



GitOps



1. Declarative ← .yaml files
2. Versioned and Immutable ← In GitHub
3. Pulled Automatically ← Argo CD
4. Continuously Reconciled ← Argo CD

https://opengitops.dev/

https://argoproj.github.io/cd/

Migration

1. Figure out what is running …
2. … and add to GitHub repo
3. Set up Argo CD with sync disabled
4. Announce migration window → enable sync
5. Success! → ?



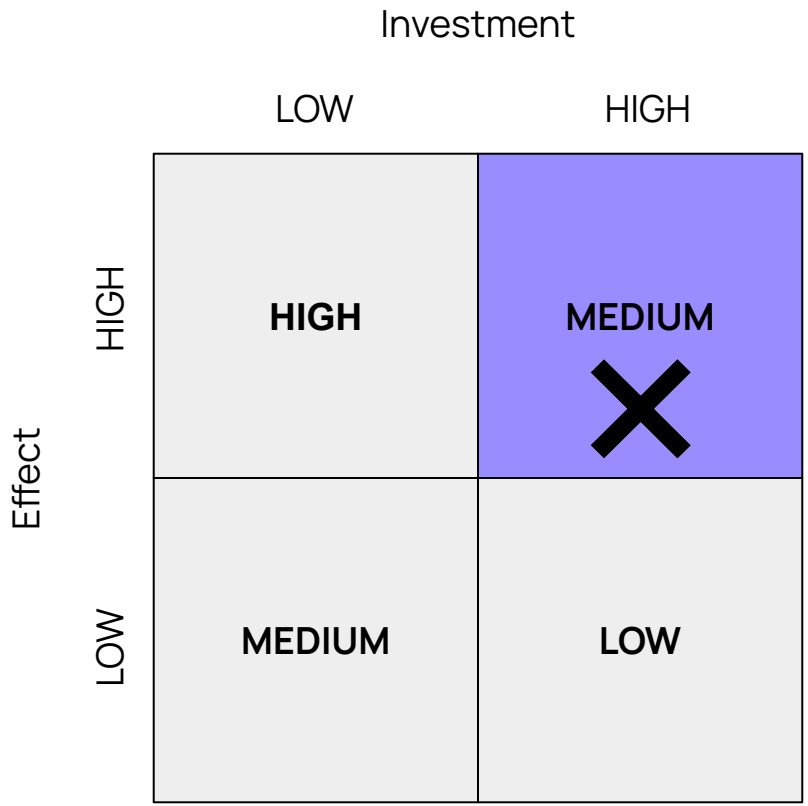
Deployment process after

- Get latest master image tag
- Add to git repository



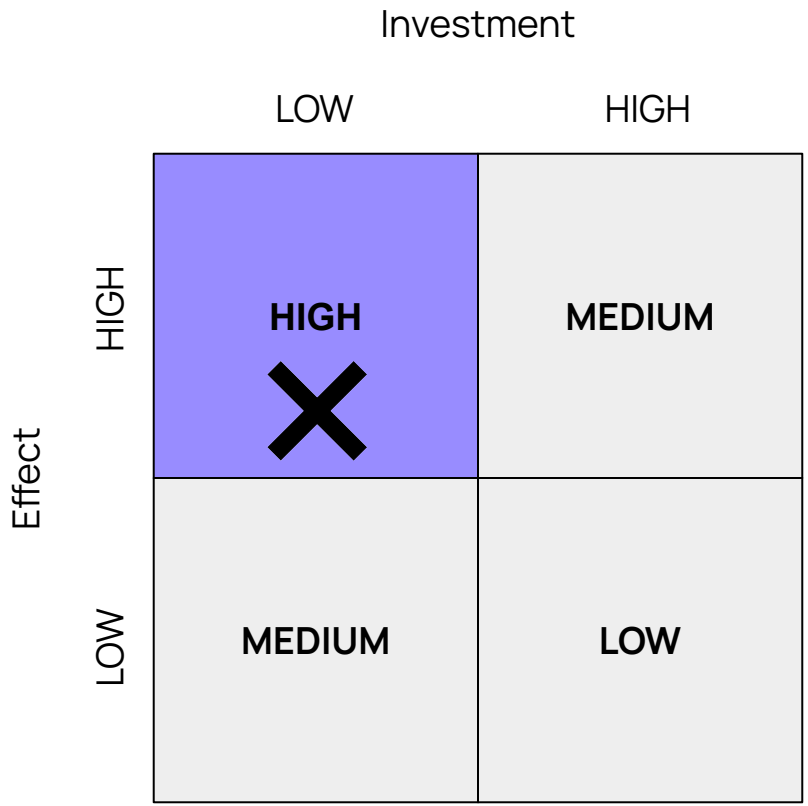
ROI

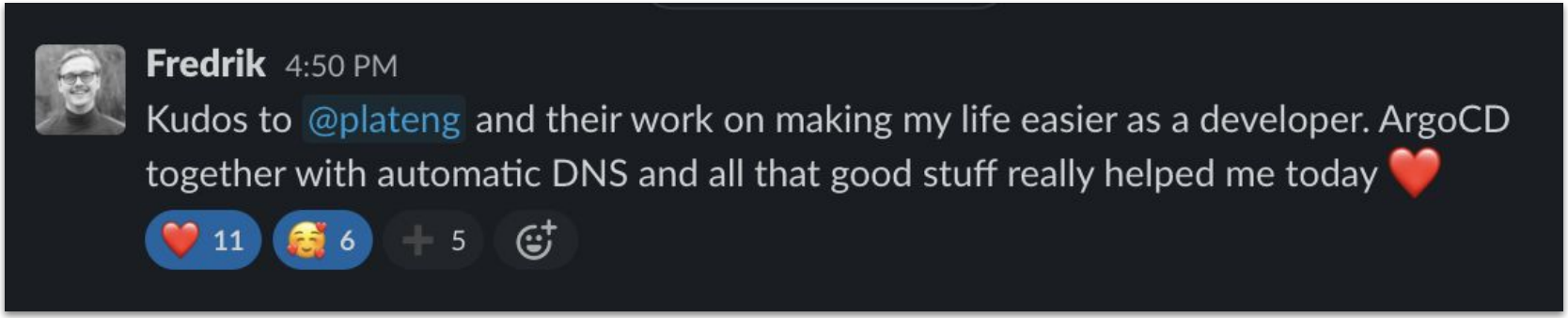
- **Transparent process**
- **Resilience**
- **Easy to use***



> ROI

- **Transparent process**
- **Resilience**
- **Easy to use***




Fredrik 4:50 PM
Kudos to @plateng and their work on making my life easier as a developer. ArgoCD together with automatic DNS and all that good stuff really helped me today ❤️
❤️ 11
🥰 6
5

* see “Rebranding”

Things we more or less* stopped thinking about:

DNS records | Certificates | Deployments



Strategy as a small platform team

1. Unblocking developers



Strategy as a small platform team

1. Unblocking developers
2. Removing time consuming tasks from the team

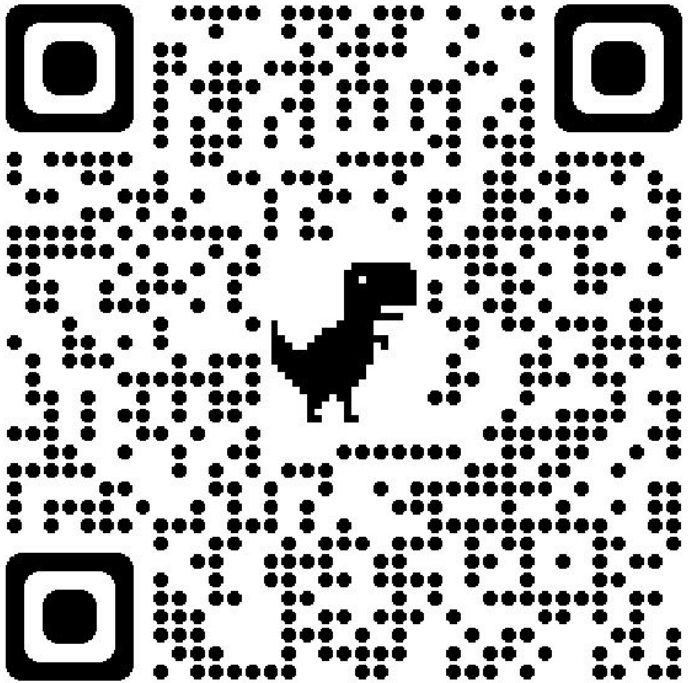


Strategy as a small platform team

1. Unblocking developers
2. Removing time consuming tasks from the team
3. Removing time consuming tasks from the developers



Thank you!



Thank you!

Links

https://teamtopologies.com/key-concepts

GitOps As a Journey - Dan Garfield, Codefresh; Scott Rigby, Weaveworks & Chris Short, AWS

From Kubernetes to PaaS to … Er what's next? - Daniel Bryant, from KubeCon + CloudNativeCon EU 2022

https://github.com/kubernetes-sigs/external-dns

https://cert-manager.io/docs/

https://argoproj.github.io/cd/

https://opengitops.dev/

