

DEVELOPER EXPERIENCE

Crafting High-Efficiency developer workflows @ Lunar

Kasper Juul Hermansen

@ME

Kasper Juul Hermansen

Platform Engineer @ Lunar

- github.com/kjuulh
- git.front.kjuulh.io/kjuulh
- blog.kasperhermansen.com

DEVELOPER EXPERIENCE

- What is Developer Experience
- Developer Experience at Lunar
- Questions

DEVELOPER EXPERIENCE, DEVEX, DX - WHAT?

- Focusing on Developer Happiness
 - Better results
 - More consistently
 - Sustainability

DEVELOPER EXPERIENCE

- What is the problem we want to solve
 - Surveys
 - Research
- How should it be solved
 - Communicate the problem
 - Implement the solution

RECENT HYPE

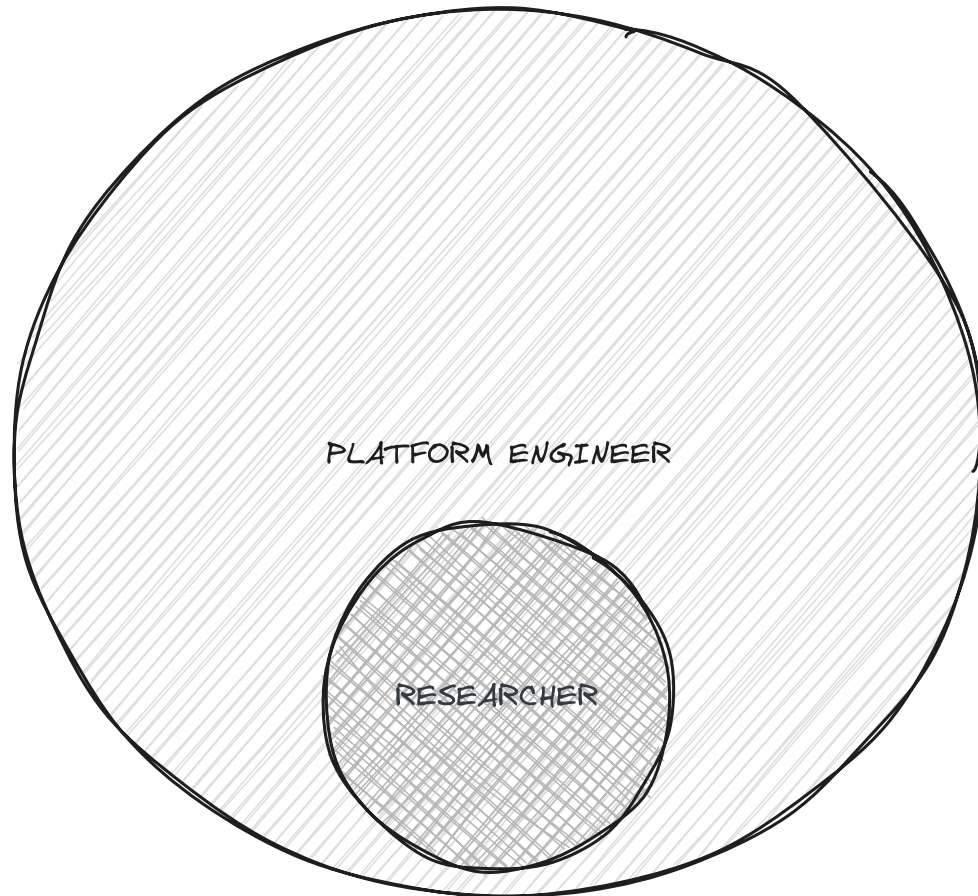
- Marketing: DX
- But an old term:
<https://wiki.mozilla.org/DeveloperExperience> ~
2012

TEAM TOPOLOGIES

- @Lunar
- At a large company
- At a really large company

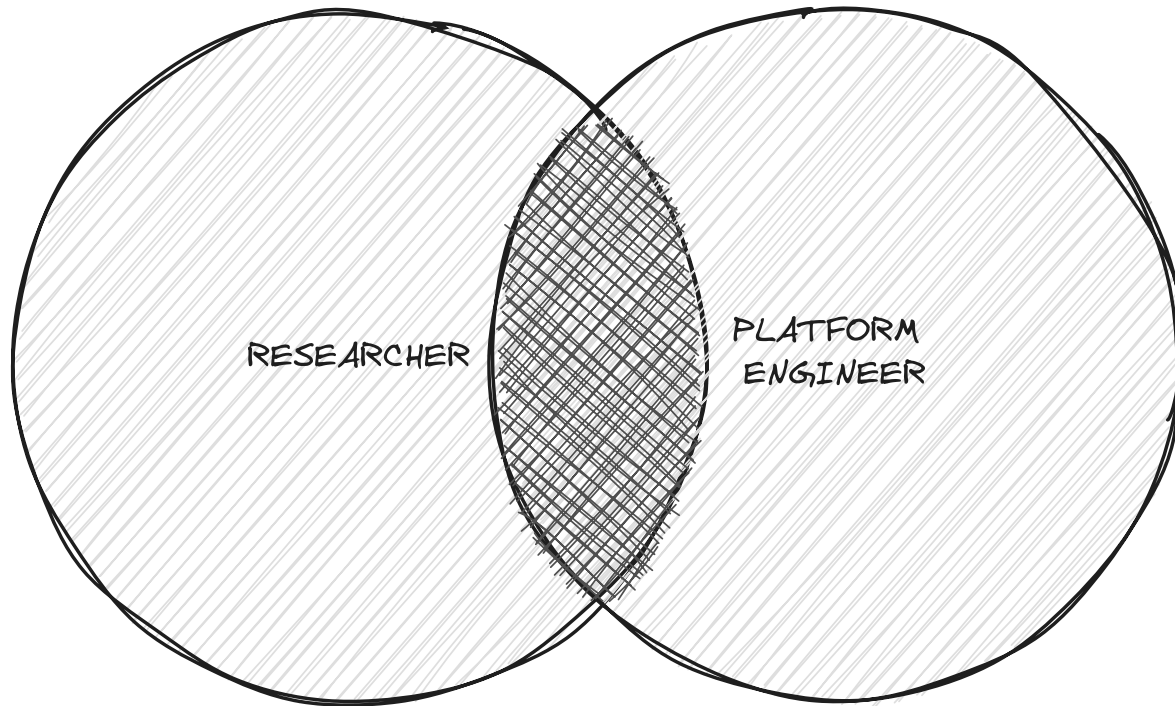
TEAM TOPOLOGIES: SMALL SCALE

- Exists as part of a team, or a subset of



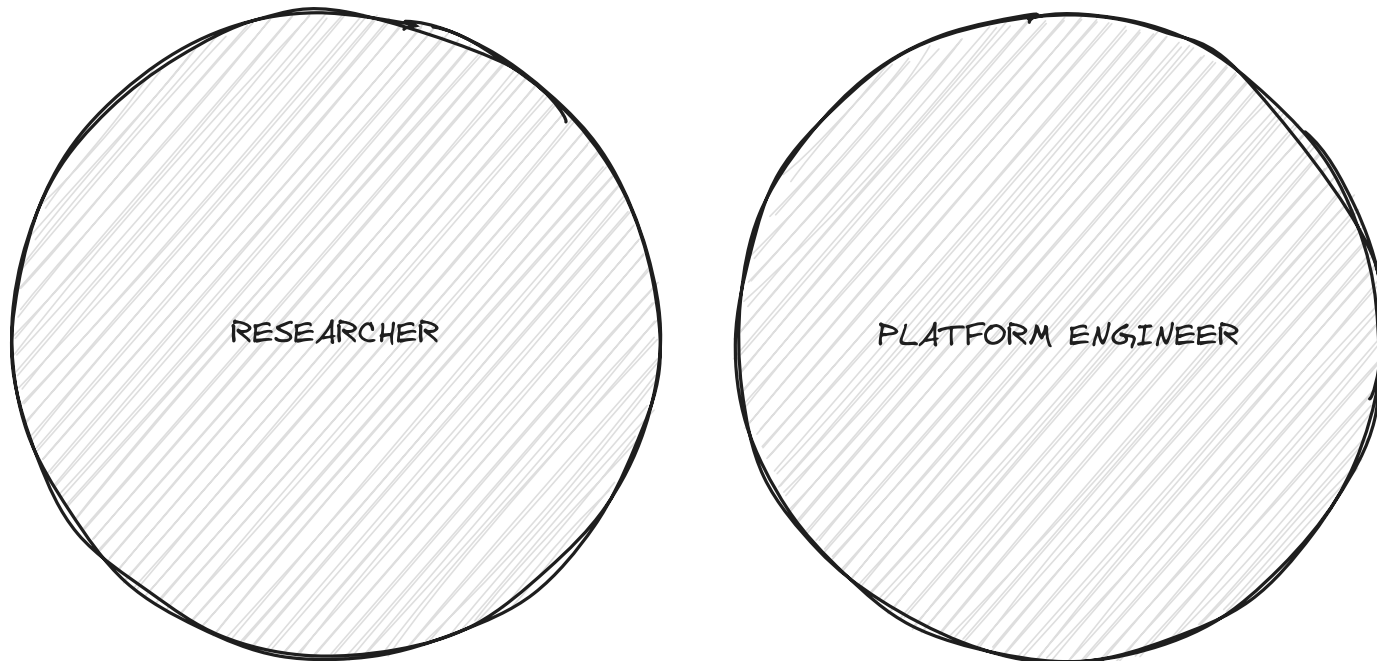
TEAM TOPOLOGIES: MEDIUM SCALE

- Exists as a team with



TEAM TOPOLOGIES: LARGE SCALE

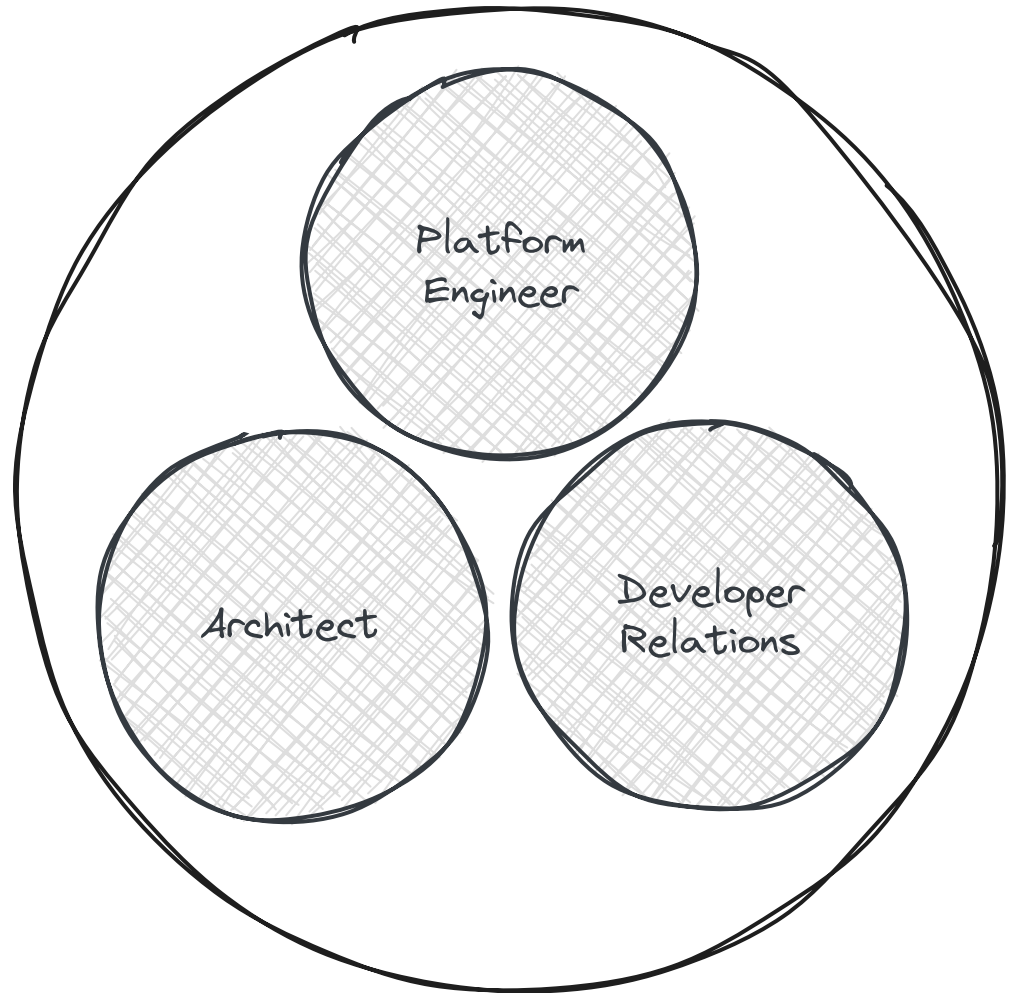
- Exists as a separate team



DEVELOPER EXPERIENCE @ LUNAR

TOPOLOGY

- Developer Relations
- Platform Engineer
- Architect
- No Academics?



DEVELOPER EXPERIENCE IN PRACTICE

- Finding problems to solve
 - Actually talking to people
 - Some times they come to us on their own
 - Looking at data, DORA metrics, our own observability

WHAT PROBLEMS TO SOLVE

- 80/20 rule
- Solve classes of problems
- Couple data with user feedback
- Be conservative
 - Don't feel bad for your ideas, they're not puppies

EXECUTING

- Quickly build and test tools
- Rolling out solutions
 - Gathering feedback
 - Remember to reflect on feedback before acting
- Communication is key

ASPIRATIONS

- Focusing on the golden path
- We take on platform complexities and abstractions
- Letting Developers be in control what is necessary for them to succeed

EXAMPLE: CREATING A NEW SERVICE

- 1 form in backstage
- 1 commit to a github repository to go to production
- around 10 minutes for a service with a database, message queue, grpc, http, logging, metrics, etc.



👤 Squad Aura

CHOOSE

👤 Squad Aura

CHOOSE

service



Empty RPA REFramework Process Repository

Creates an near empty RPA REFramework
process repository with default Lunar files

👤 Guild RPA

CHOOSE

cli



Go CLI

Create a new Go cli

👤 Squad Aura

CHOOSE

service



Go Service with fx-baseservice

Create a new Go service using fx-baseservice
for components

👤 Squad Aura

CHOOSE

service



Go Service with vostok

Create a new Go service using Vostok for
components

👤 Squad Aura

CHOOSE

Create a new component

Create new software components using standard templates in your organization

Go Service with vostok

Create a new Go service using Vostok for components



Fill in the needed information to create the service



Review

Name *

some-service

Unique name of the service.

Description *

some-explanation

What is the responsibility of the service?

Owner *

squad-aura

The squad that will own the service.

Domain *

developer-enablement

The business domain the service belongs to, eg. `card-payments`. If your domain is not in the dropdown reach out to your friendly neighbourhood architect.

Kubernetes namespace

The kubernetes namespace, eg. bank. Leave empty to default to workload namespaces, ie. dev and prod.

Setup auto release policy for




dev



prod




platform

lunarway / some-service

Q Type to search


>+⌵

deIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

some-servicePrivate

Edit PinsWatch 2Fork 0Star 0

master1 Branch0 TagsGo to fileAdd fileCode

kjuulhInitial commit from Backstage scaffolder439256a · now1 Commits

.github/workflows	Initial commit from Backstage scaffolder	now
cmd	Initial commit from Backstage scaffolder	now
internal	Initial commit from Backstage scaffolder	now
.gitignore	Initial commit from Backstage scaffolder	now
CODEOWNERS	Initial commit from Backstage scaffolder	now
README.md	Initial commit from Backstage scaffolder	now
go.mod	Initial commit from Backstage scaffolder	now
go.sum	Initial commit from Backstage scaffolder	now
renovate.json	Initial commit from Backstage scaffolder	now
shuttle.yaml	Initial commit from Backstage scaffolder	now

README

some-service

some-explanation

Scaffolded from [go-service-template-vostok](#)

Aboutsome-explanation

squad-aura

ReadmeActivityCustom properties0 stars2 watching0 forks

ReleasesNo releases publishedCreate a new release

PackagesNo packages publishedPublish your first package

LanguagesGo 100.0%



master ▾

some-service / shuttle.yaml



kjuulh Initial commit from Backstage scaffolder ●



43 lines (34 loc) · 883 Bytes

Code

Blame

```
1  plan: git://git@github.com:lunarway/lw-shuttle-go-plan.git
2
3  vars:
4    service: some
5    squad: aura
6    domain: developer-enablement
7    docker:
8      builder-image: golang:1.22.0
9
10   db: false
11   rabbitmq: false
12   ingress: false
13
```

WHAT YOU GET

- Signals:
 - Dashboards in grafana
 - Shared logs
 - Alerts
- Dependency updates:
 - Renovate

WHAT YOU GET...

- Security:
 - Security notices and updates
- Continuous integration
- Continuous deployment
- Alerting on keystone metrics

TIME TO PROD

Zero to prod

~4 minutes

APPLICATION AS A SERVICE

- No kubernetes yaml files
- No CI pipeline
- No bash scripts
- Only code and a bit of configuration

BUILDING TOOLS FROM FIRST PRINCIPLES

- If nothing better isn't out there
- Focus entirely on the problem at hand
 - Use any means, open source, proprietary, bought
- Only expose necessary parts that our developers needs to use

EXAMPLE: CI

- New CI system
 - Focusing on solving reliability problems
 - Lots of feedback from developers
 - Solving their specific needs



```
# file: .github/workflows/ci.yaml
name: ci
on:
  push:
    branches:
      - "*"
  pull_request:
  workflow_dispatch:
jobs:
  ci:
    uses: lunarway/lw-shuttle-go-plan/.github/workflows/plan-c
    secrets: inherit
    with:
      lifecycle: stable
```

WHAT YOU GET

- Build times from seconds to a few minutes
 - 99th percentile ~4 minutes
 - 95th percentile ~ 1 minute

CENTRALIZED CI PIPELINE

- Built from entirely in Golang
- Packaged like regular software libraries
- Tested like normal software
- Teams can own their own parts of the pipeline, if they desire

FORMING CONNECTIONS

- Let people become ambassadors for you
- Small victories over big bangs

QUESTIONS?



Slides



