



adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE
21st – 23rd OCTOBER 2024

Scaling up - Implement large-scale projects on AEM Edge Delivery Services

Anian Weber (TechDivision), Markus Haack (Adobe)



Anian Weber

Software Architect
TechDivision GmbH

 /anianweber



Markus Haack

Sr. Software Engineer
Adobe AEM Sites

 /markushaack

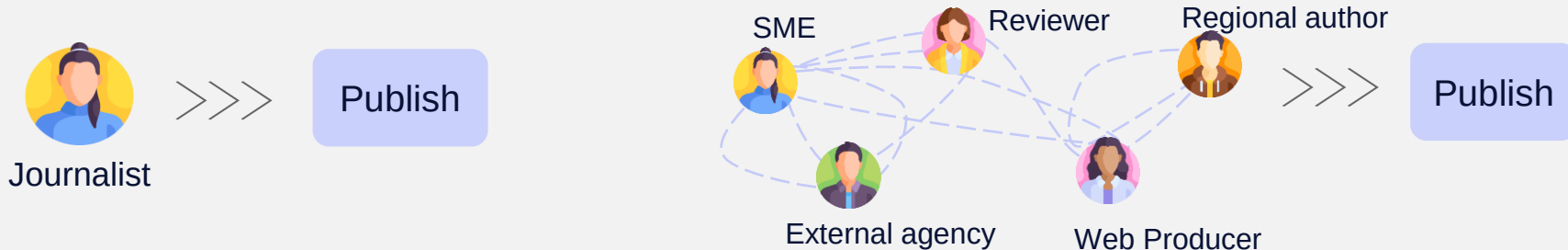
Agenda

- Large projects on Edge Delivery Services
- Project Types:
 - one for all
 - customizations and overrides

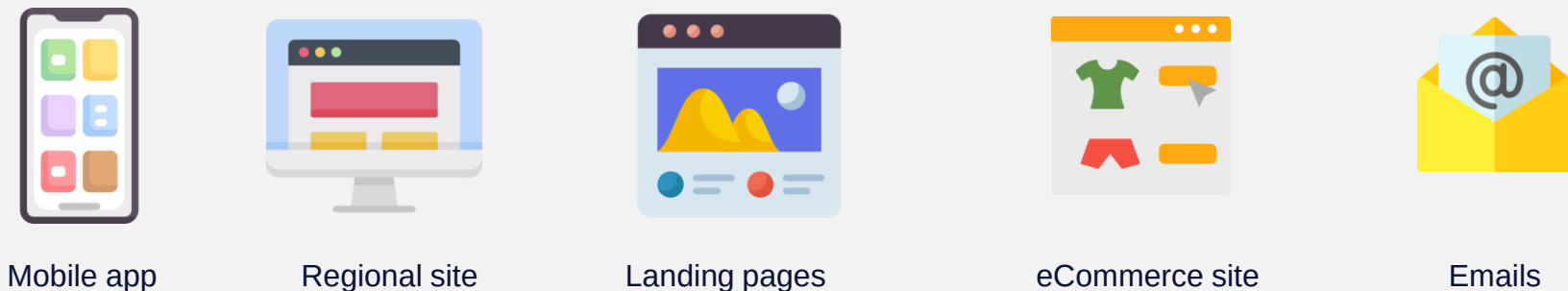
Overview

One size doesn't fit all

Different customers have different workflows & goals



Different customers have different use cases



What are “large scale” projects?

- Global multi brand companies
- Multi region & multi language sites
- Many domains connected
- Different content sources
- Complex sites due to laws and regulations

Challenges



Developer

- Code reuse and sharing across multiple sites with flexibility and control
- Maintain different site themes, branding
- Maintain different localizations
- Avoid CDN stitching mess



Business

- Content reuse across regions and brands
- Automated translation workflows
- Integration of corporate DAM
- Content migration

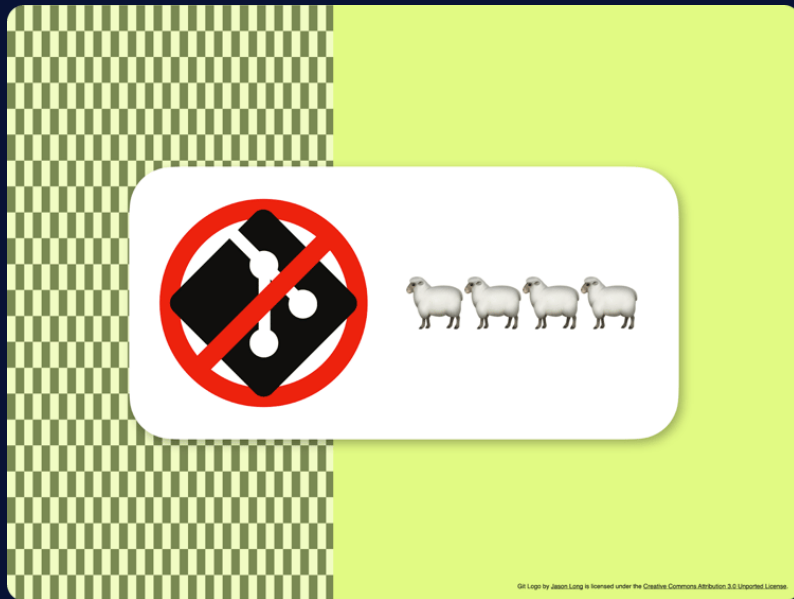
Project Type: one for all

One codebase, many sites

- Traditionally 1:1 mapping
 - GitHub org and repo map to site URL
- aem.live
 - Org & Site are independent from GitHub
 - Share and re-use code (GitHub repo) for multiple sites
 - Set up via the Config Service

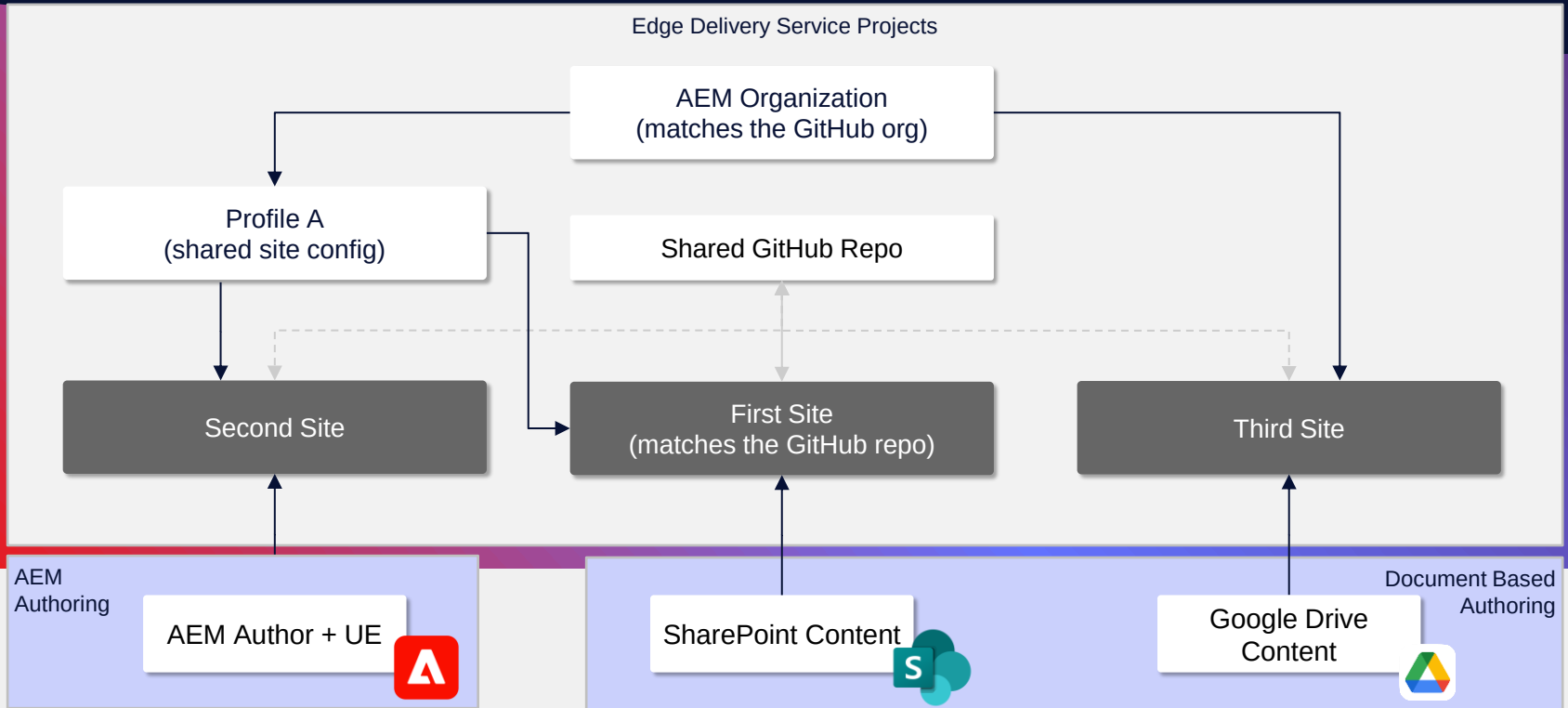
Docs: <https://www.aem.live/docs/repoless>

One codebase, many sites

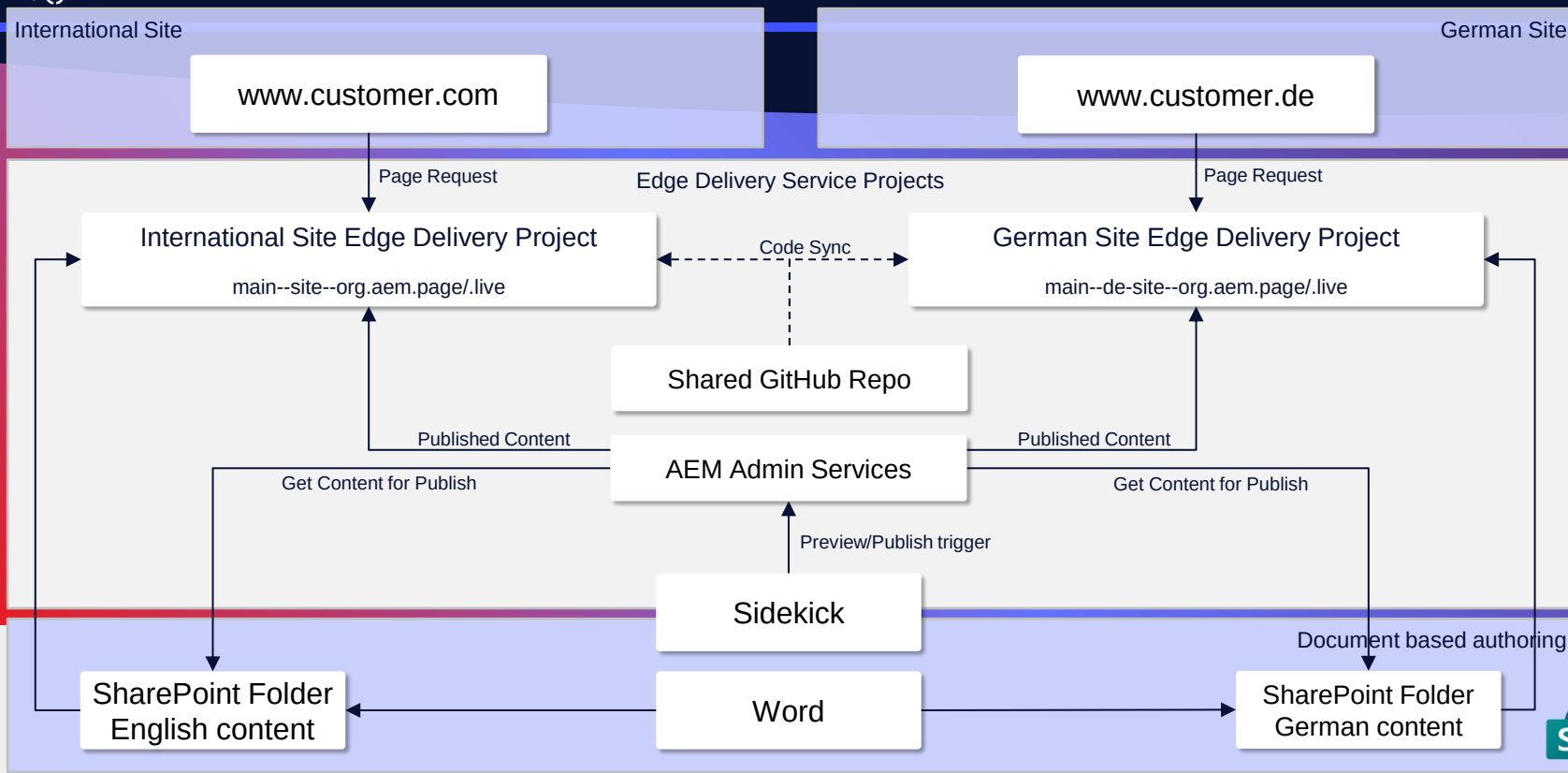


We call it “repoless”

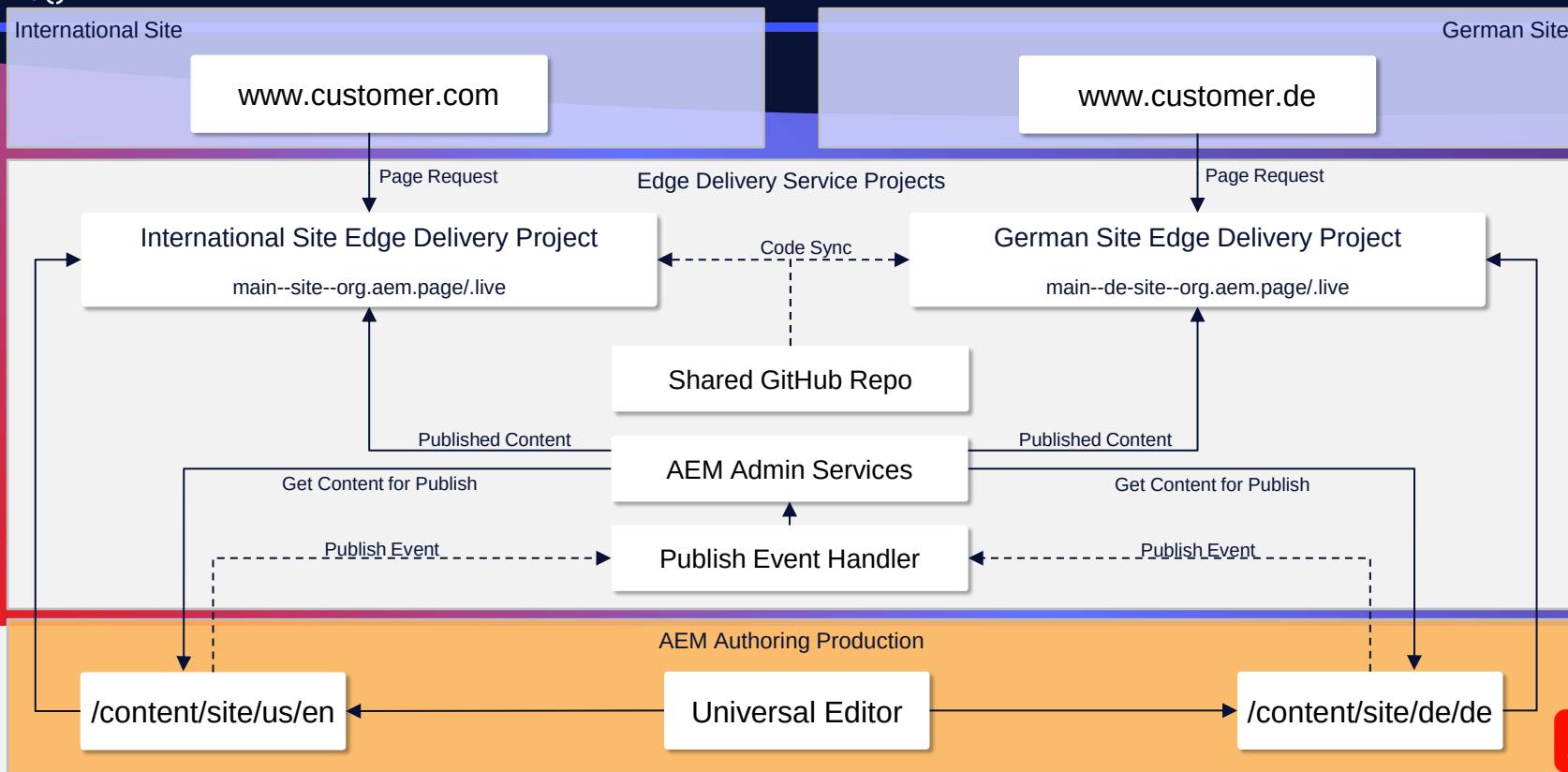
How does it work?



A typical document-based project with aem.live



A typical Crosswalk project with aem.live



How to set up? (1/2)

- Pre-requisites
 - Github Repository
 - Content Repository
- Create your first base site
 - Create organization
 - Create site within organization
- URL: main--site--org.aem.live

Docs: <https://www.aem.live/docs/admin.html>

```
curl -X PUT https://admin.hlx.page/config/{org}.json
```

```
curl -X POST
https://admin.hlx.page/config/{org}/sites/{site}.json \
-H 'content-type: application/json' \
--data '{
  "code": {
    "owner": "{org}",
    "repo": "{site}"
  },
  "content": {
    "source": { "url":
"https://{org}.sharepoint.com/sites/aem/Shared%20Documents/{site}" }
  }
}'
```

How to set up? (2/2)

- Create your second “repoless” site
 - With same Git Repository
 - With different Content Source
- URL: main--site2--org.aem.live

```
curl -X POST
https://admin.hlx.page/config/{org}/sites/{site2}.json \
-H 'content-type: application/json' \
--data '{
  "code": {
    "owner": "{org}",
    "repo": "{site}"
  },
  "content": {
    "source": { "url":
"https://{org}.sharepoint.com/sites/aem/Shared%20Documents/{site2}" }
  }
}'
```

Advantages of a repless setup

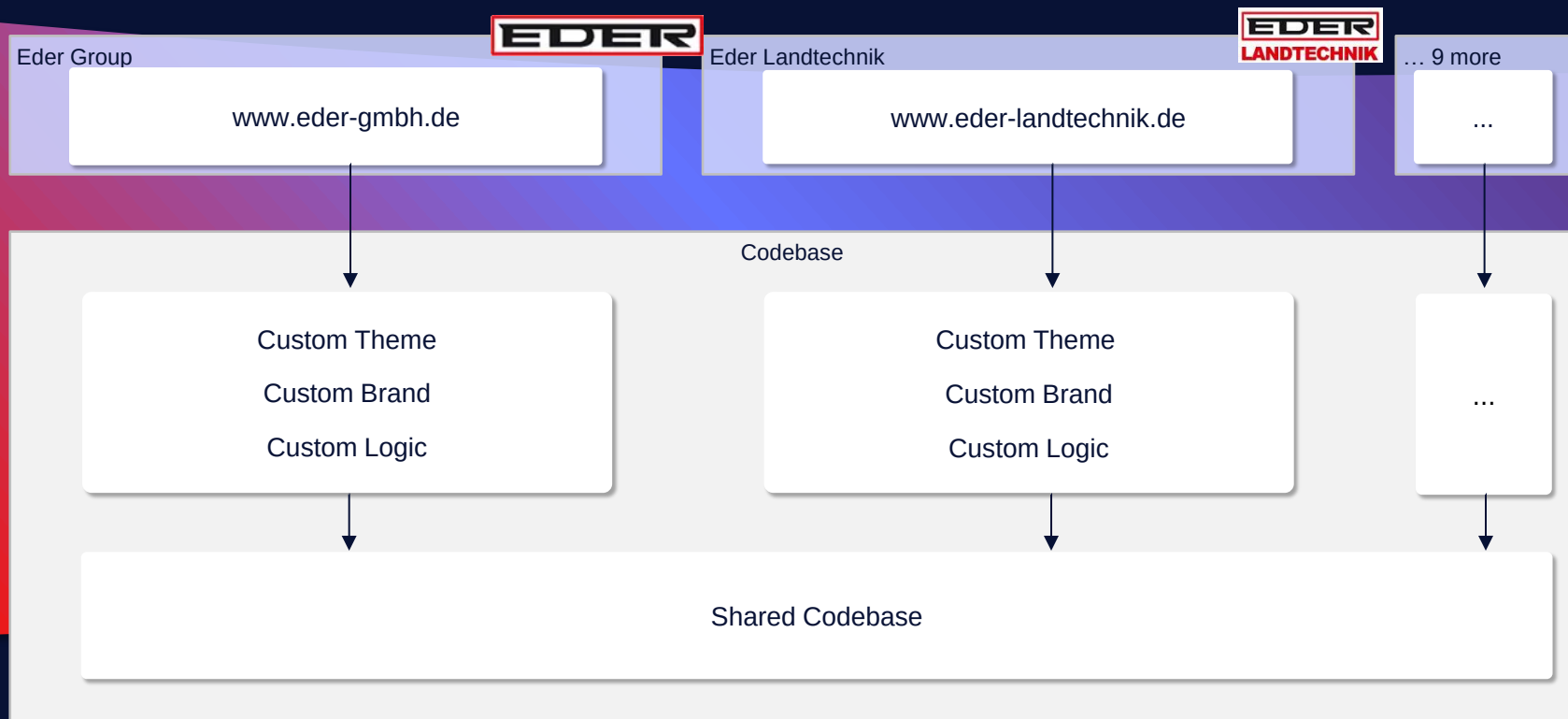
- Sharing project code base on multiple sites
- Use different content sources
- For Crosswalk
 - align with the AEM CS deployment patterns (DEV, STAGE, PROD)
 - MSM site setup

Project Type: customizations and overrides

One codebase, many sites

- Traditionally 1:1 mapping
 - 1 GitHub Repo maps to 1 URL
- This approach is still valid!

Project Type: customizations and overrides

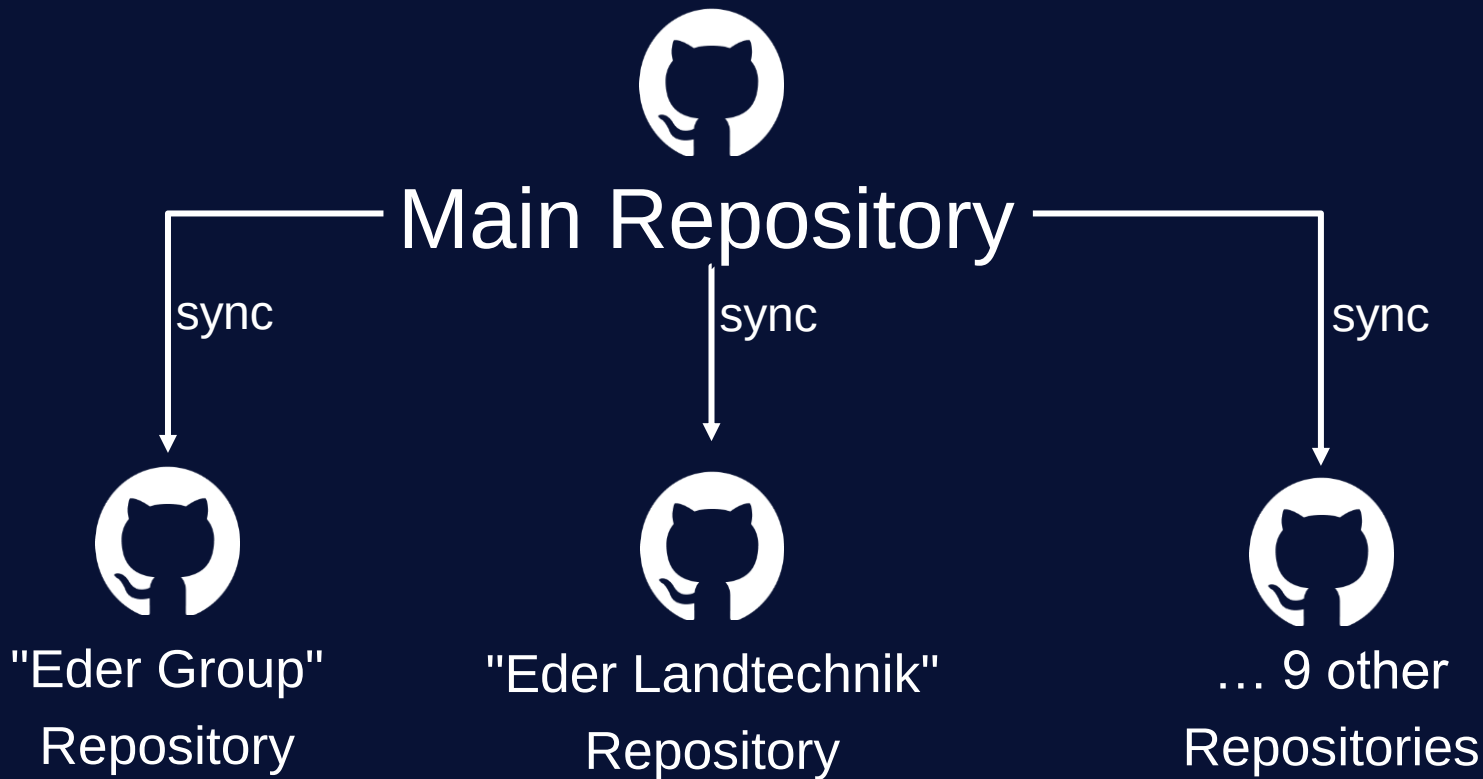


How?

How to?

- Share code
- Be flexible
- Easy and developer friendly setup

How?



How?

Main Repo

```
styles/theme.css  
blocks/teaser/teaser.js  
blocks/teaser/teaser.css  
fstab.yaml
```

Normal Edge Delivery Services project (simplified)

How?

Main Repo

```
.multisite
|
|  eder-group
|  |
|  |  styles/theme.css
|  |  blocks/teaser/teaser.css
|  |  fstab.yaml
|  |
|  |  eder-landtechnik
|  |  |
|  |  |  styles/theme.css
|  |  |  fstab.yaml
|  |
|  |  styles/theme.css
|  |  blocks/teaser/teaser.js
|  |  blocks/teaser/teaser.css
|  |  fstab.yaml
```

For each commit

How?

Temporary Copy

```
.multisite
|
| eder-group
| |
| | styles/theme.css
| | blocks/teaser/teaser.css
| | fstab.yaml
| |
| | eder-landtechnik
| | |
| | | styles/theme.css
| | | fstab.yaml
|
| styles/theme.css
|
| blocks/teaser/teaser.js
|
| blocks/teaser/teaser.css
|
| fstab.yaml
```

Make temporary copy

How?

Temporary Copy

```
.multisite
├── eder-group
│   ├── styles/theme.css
│   ├── blocks/teaser/teaser.css
│   └── fstab.yaml
├── eder-landtechnik
│   ├── styles/theme.css
│   └── fstab.yaml
├── styles/theme.css
├── blocks/teaser/teaser.js
├── blocks/teaser/teaser.css
└── fstab.yaml
```



```
> cp .multisite/eder-group/. ./
```

How?

Temporary Copy

```
.multisite
|
|   eder-group
|   |
|   |   styles/theme.css
|   |   blocks/teaser/teaser.css
|   |   fstab.yaml
|   |
|   |   eder-landtechnik
|   |   |
|   |   |   styles/theme.css
|   |   |   fstab.yaml
|   |
|   |   styles/theme.css
|   |   blocks/teaser/teaser.js
|   |   blocks/teaser/teaser.css
|   |   fstab.yaml
```

```
> git commit -am "apply overrides"
```

How?

Temporary Copy

```
.multisite
├── eder-group
│   ├── styles/theme.css
│   ├── blocks/teaser/teaser.css
│   └── fstab.yaml
├── eder-landtechnik
│   ├── styles/theme.css
│   └── fstab.yaml
└── styles/theme.css
    blocks/teaser/teaser.js
    blocks/teaser/teaser.css
    fstab.yaml
```

eder-group Repo

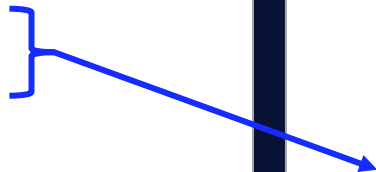
```
styles/theme.css
blocks/teaser/teaser.js
blocks/teaser/teaser.css
fstab.yaml
```

> git push eder-group-repo **--force**

How?

Main Repo

```
.multisite
├── eder-group
│   ├── styles/theme.css
│   ├── blocks/teaser/teaser.css
│   └── fstab.yaml
├── eder-landtechnik
│   ├── styles/theme.css
│   └── fstab.yaml
├── styles/theme.css
├── blocks/teaser/teaser.js
├── blocks/teaser/teaser.css
└── fstab.yaml
```



eder-group Repo

```
styles/theme.css
blocks/teaser/teaser.js
blocks/teaser/teaser.css
fstab.yaml
```

eder-landtechnik Repo

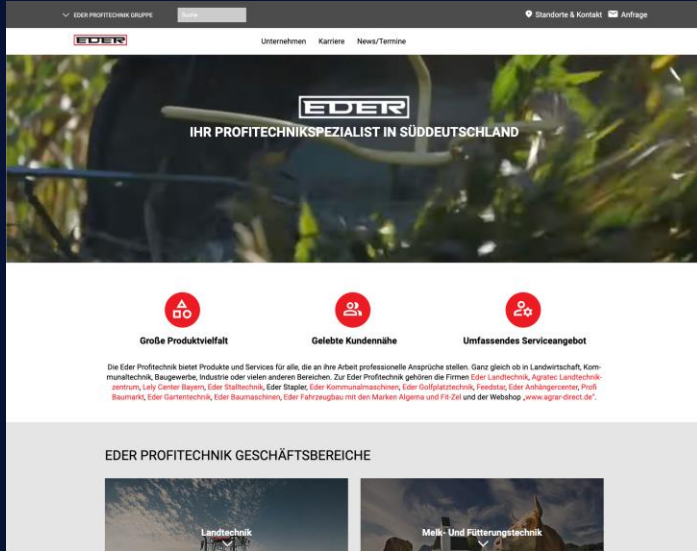
```
styles/theme.css
blocks/teaser/teaser.js
blocks/teaser/teaser.css
fstab.yaml
```

Repeat for all repositories

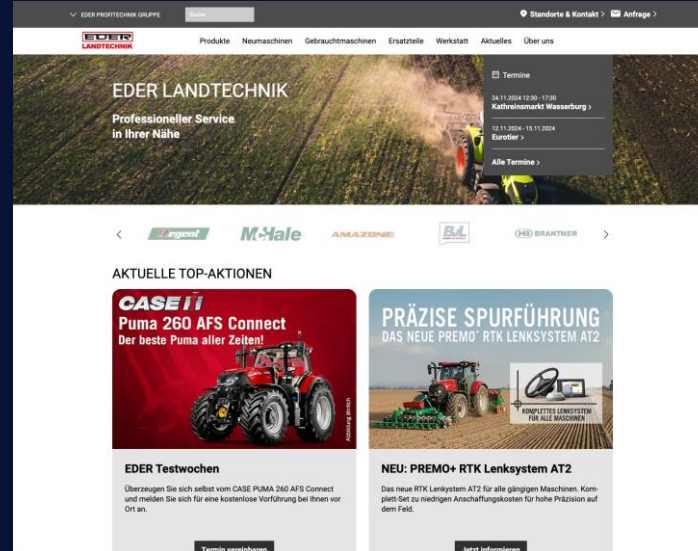
Challenges in this process

- Authentication within GitHub
 - Solved by a private App
- Local development

aem up --site=eder-group



aem up --site=eder-landtechnik



Key Takeaways

Key Takeaways

1. Many Edge Delivery Project come with individual requirements
2. Same key topics – sharing code, localizations, ...
3. Different approaches to address these
 - Repoless with one shared code base
 - Multiple repositories synchronized – with overwrite logic
 - Full controlled releases with dedicated repositories

AEM Edge Delivery Services has proven to
be suitable for large scale projects!

Thank You