



Backstage

Nick McKenzie

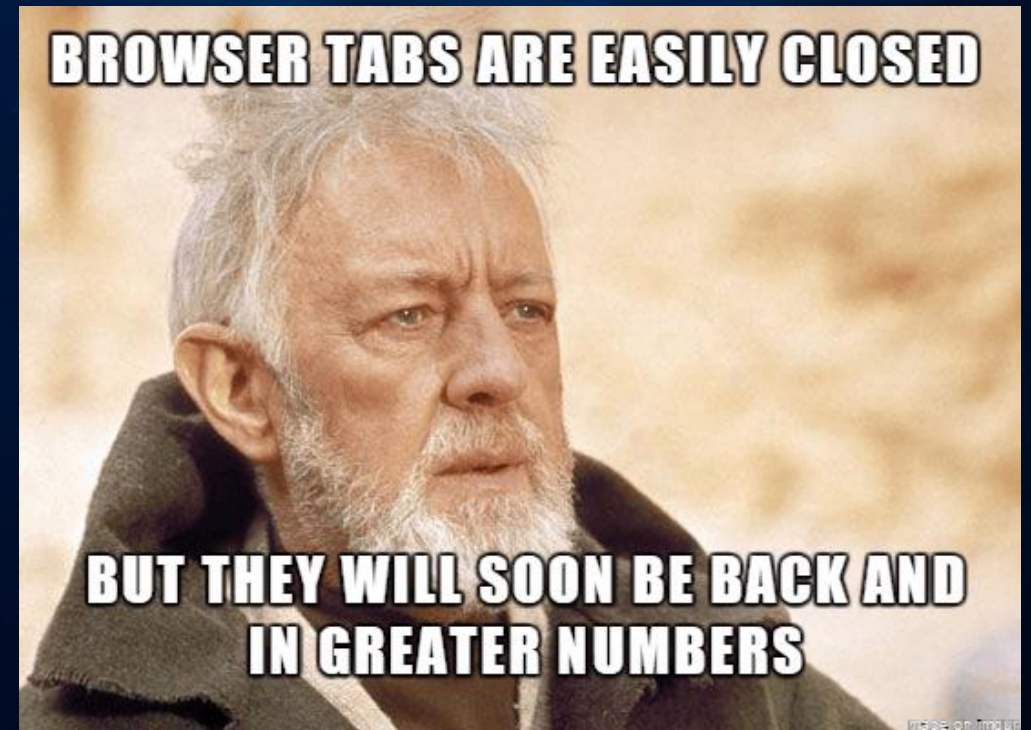
MCB Engineering Chapter Lead

nickmck.net



Developer Experience 2024

- How many tabs do you typically have open?
- How many apps do you interact with?
- Typical day at MCB:
 - Sonar, ADO, Openshift, OpsGenie, Confluence, Jaeger, Prometheus, Grafana, Sharepoint, Azure Portal, Jira...
- Then there's the challenge to find the right info...



What is Backstage?

- An open-source developer portal platform created by Spotify and donated to CNCF.
 - Access to Information from multiple sources
 - Library of templates
 - Discover services and their owners
- Single 'Pane of Glass' to Development and Ops Tools
- Centralizes and standardizes software development infrastructure
- Improves developer experience and productivity



Backstage at MCB

- Consistent access to information
- Improve Discoverability
- Standardise Ops Information and Actions
- Home for OpenApi Docs
- Extend Architecture Documentation
 - Stores the implementation detail for our Enterprise Architecture.



Where are we?

- 2 POC's Completed with feedback from the Engineers
- Trained our Tech Leads
- Modelled parts of our environment
- Evaluating 'Docs as Code'
- Implementing our first upgrade
- Implemented our first plugin
- Branding is MCB Vault

Core Features

A row of three black metal rolling cases with their lids open, containing audio equipment, in a stage setting. The cases are on wheels and have silver handles. The background shows stage lighting and trussing.

Core Features



Software
Catalogue



Kubernetes



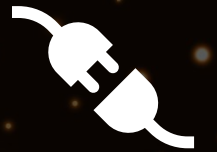
Search



Templates



TechDocs



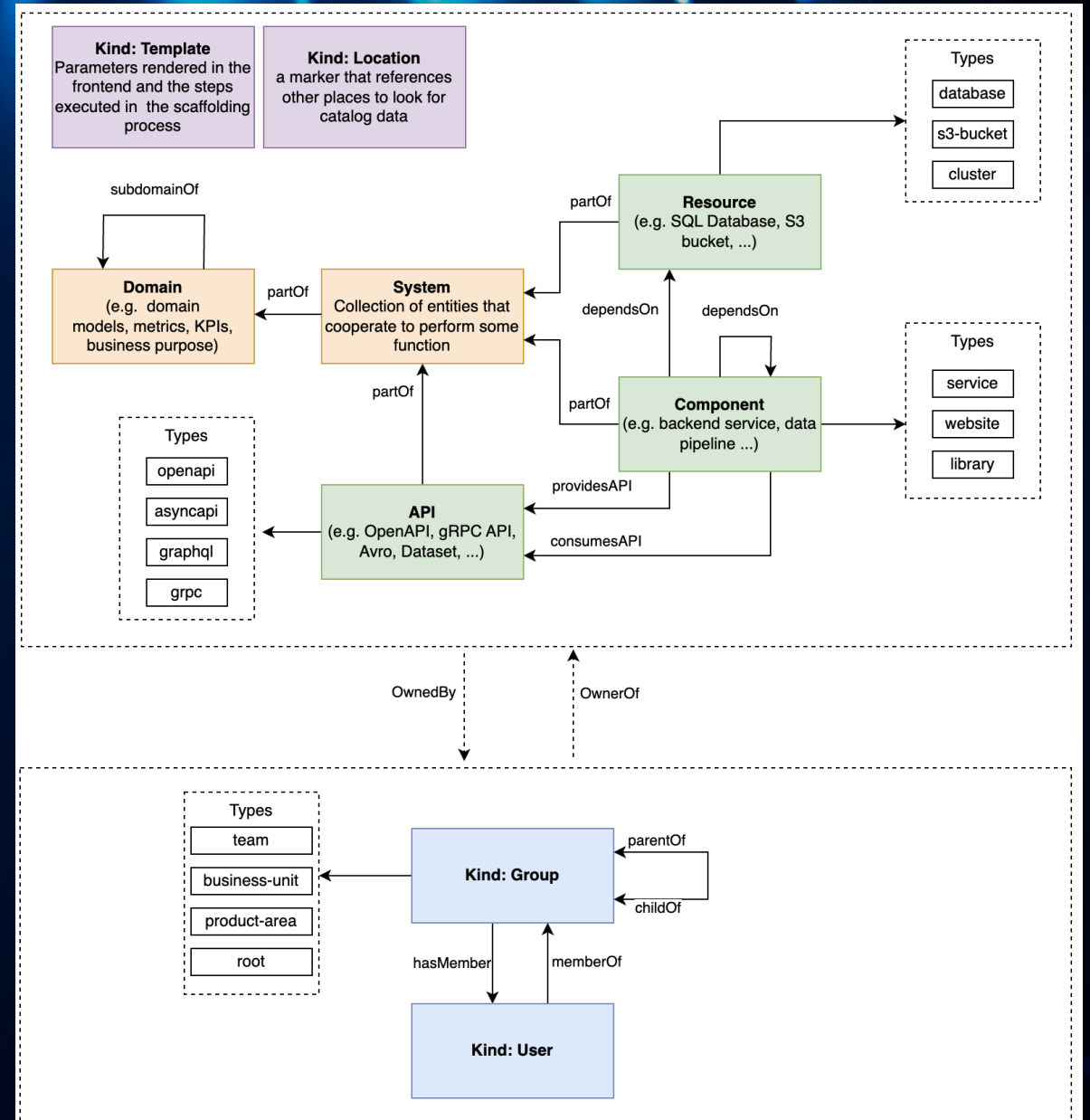
Plugins

Software Catalogue

- Central repository for all software components
- Provides visibility into ownership, lifecycle, and dependencies
- Uses
 - Service discovery
 - Dependency management
 - Ownership tracking
 - Impact Analysis

Entity Model

Standardized way to describe software components and their relationships.



Kubernetes Integration

- Visualize K8s resources
- Monitor deployments and health
 - Clusters
 - Namespaces
 - Pods
 - Health
 - Logs

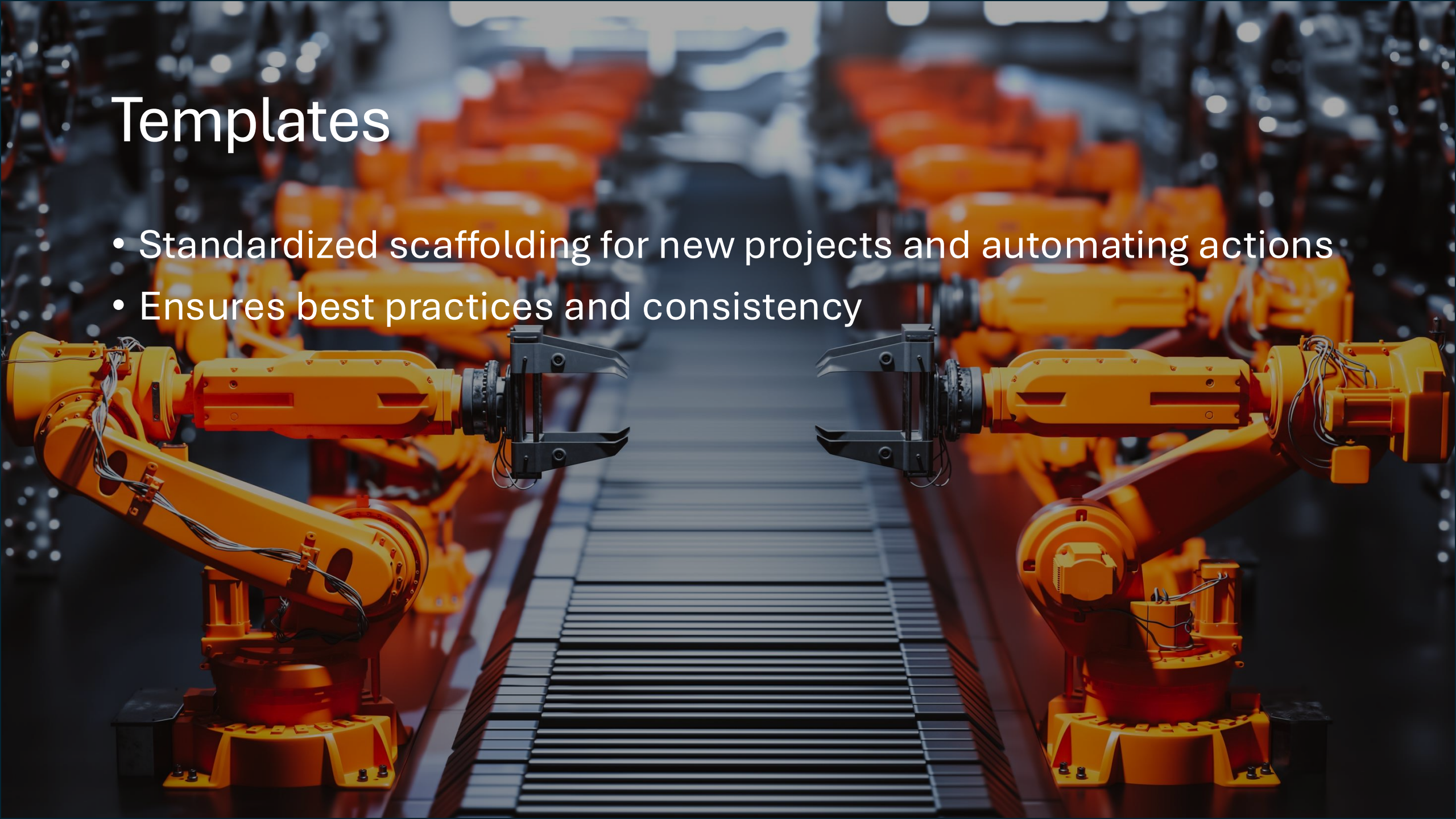


Search

- Unified search across all Backstage data
- Quick access to services, documentation, and resources
- Out of the box support for Lunr, Postgres and Elasticsearch.

Templates

- Standardized scaffolding for new projects and automating actions
- Ensures best practices and consistency



TechDocs

- Markdown-based documentation
- Versioned and stored with code – [Docs as Code](#)
- Entities can also link to documentation stored in Sharepoint, Confluence etc.

Plugins

- Extend the capabilities of Backstage

- Front-end
- Back-end
- Scaffolder

- Over 200 Plugins available at <https://backstage.io/plugins/>

- ...or create your own!

Demo

A Backstage Tour



A row of black equipment cases on wheels, open, in a backstage area. The cases are filled with various pieces of audio equipment, including microphones and stands. The scene is lit with blue stage lights, and the background shows a complex metal truss structure.

Implementing Backstage

Skills Required

- Familiarity with modern web development
- JavaScript/TypeScript
- React
- Node.js
- Docker



Getting Started

A row of three black equipment cases with silver metal frames and wheels. The cases are open, revealing blue-lined interiors. The first case contains various pieces of audio equipment, including a mixer and several microphones. The cases are arranged in a line on a wooden stage floor. In the background, there is a complex metal truss structure with blue and white lights, suggesting a stage or event space.

// Scaffolding a new Backstage instance...

```
npx @backstage
```

```
// Start your Backstage instance  
cd my-backstage  
yarn dev
```

```
⊗ nicmck  
yarn r  
$ conc  
$ yarn  
$ yarn  
$ back  
$ back  
[0] Lo  
[0] <i  
[0] <i  
[0] <i  
[0] <i  
[0] <i  
[1] Lo  
app-co
```

Backstage

Search

Home

APIs

Docs

Create...

Tech Radar

Settings

My Company Catalog

Components

CREATE SUPPORT

Type: All

PERSONAL

- Owned 0
- Starred 0

MY COMPANY

- All 1

OWNER

LIFECYCLE

PROCESSING STATUS

NAME	SYSTEM	OWNER	TYPE	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
example-website	examples	guests	website	experimental			📄 ✎ ★

EXPLORER

MCB-VAULT

- types
- mples
- e_modules
- kages
- o
- ckend
- ADME.md
- ins
- kerignore
- ntignore
- ntrc.js
- gnore
- ttierignore
- config.local.yaml
- config.production.yaml
- config.yaml
- ckstage.json
- log-info.yaml
- a.json
- age.json
- wright.config.ts
- README.md
- tsconfig.json
- yarn.lock


```
// Adding database support...
```

```
//Start Postgress
```

```
docker run --name some-postgres -e POSTGRES_PASSWORD=xxxxxxxxxxx -d postgres
```

```
//Update app-config.yaml
```

```
database:
```

```
  client: pg
```

```
  connection:
```

```
    host: localhost
```

```
    port: 5432
```

```
    user: postgres
```

```
    password: ${DB_PASSWORD}
```

Populating the Catalogue



Automated Discovery

- Catalogue files added to root of project.
- This is ultimately the preferred route
- You may not want to start here.



Manual Registration

- Catalogue files manually registered – and then automatically updated.
- Good option for POC
- Useful when testing.



API Integration

- External system registers entities via API.
- Useful to import Systems or information in other stores.

Configuring Plugins

- There are multiple plugins
<https://backstage.io/plugins/>
- Most are configured by adding annotations to your catalogue entries.
- Check the repo of the plugin for details.

annotations:

backstage.io/source-location: url:<path to repo>

sonarqube.org/project-key: "Ops_Risk_PFM_"

dev.azure.com/project-repo: Ops-Risk/PFM

dev.azure.com/host-org: dev.azure.com/mcb-itapplication

opsgenie.com/component-selector: "responders:'Deployment Monitor'"

prometheus.io/rule: jvm_memory_used_bytes, jvm_threads_live_threads

prometheus.io/alert: Identity-availability

backstage.io/kubernetes-label-selector: 'app=bpm-application'

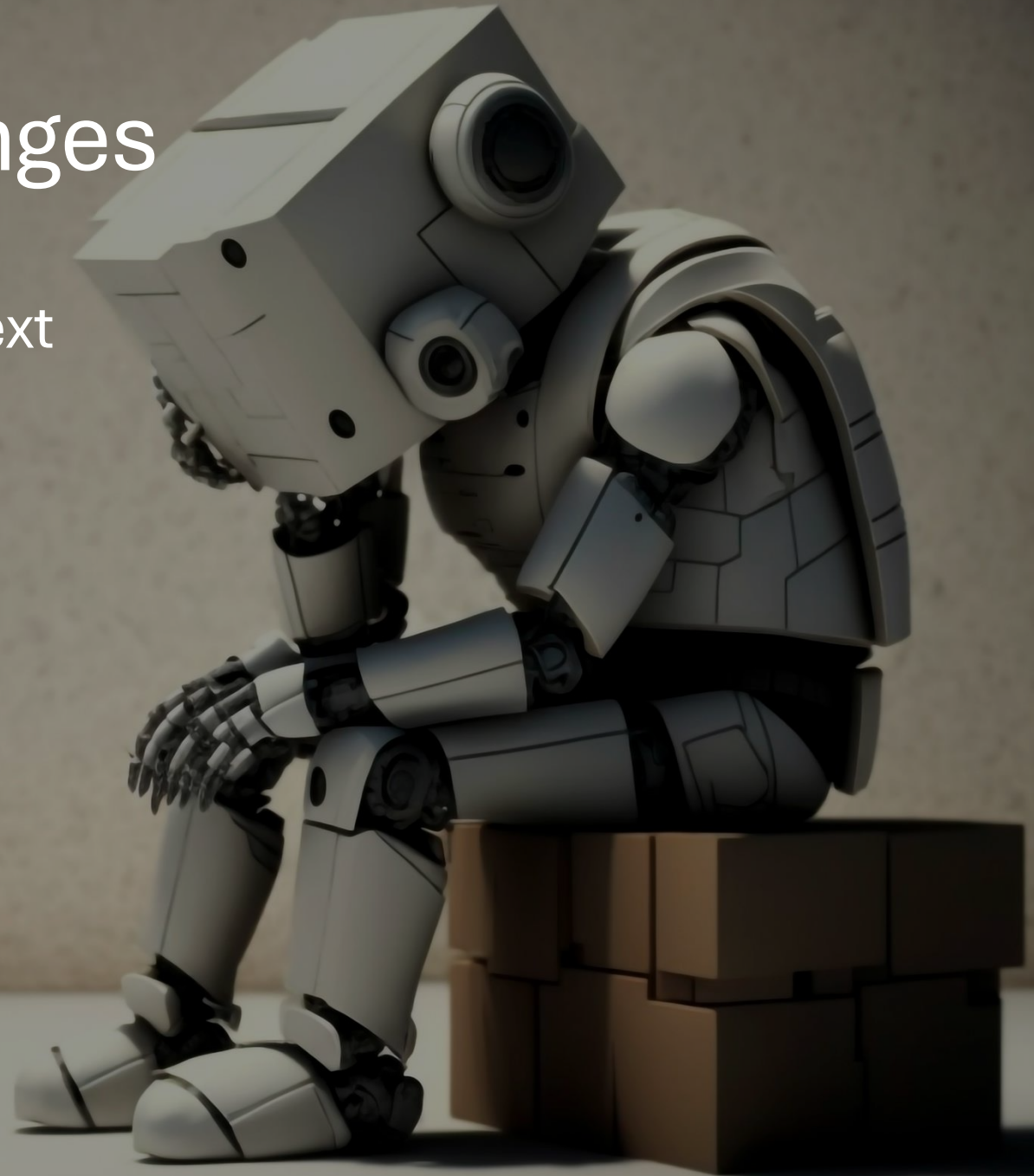
Demo –

Manual Entity Registration & Inspection



Corporate Challenges

- Container Security Context
- Self-signed Certificates
- Proxy
- Firewalls
- Authentication



Living with Backstage

- Automation
- Regular updates
- Ownership assignment
- Automated catalogue generation
- Analytics

Creating Plugins

A row of three black equipment cases with silver metal frames and wheels, arranged in a line on a wooden floor. The cases are open, revealing various pieces of audio equipment inside, including knobs, sliders, and cables. The background is a dimly lit studio or rehearsal space with metal truss structures and blue lighting. The text "Creating Plugins" is overlaid in white on the left side of the image.

// Creating a custom plug-in...

```
$ yarn backstage-cli new
```

```
? What do you want to create?
```

```
plugin-common - A new isomorphic common plugin package
```

```
plugin-node - A new Node.js library plugin package
```

```
plugin-react - A new web library plugin package
```

```
> scaffolder-module - An module exporting custom actions for @backstage/plugin-scaffolder-backend
```

Demo

Azure DevOps Builder



A row of black equipment cases on wheels is shown in a stage setting. The cases are open, revealing various pieces of equipment inside. The background features stage lighting and truss structures. The text "Questions?" is overlaid on the left side of the image.

Questions?