

INNOQ Technology Day / 13.11.2023



Backstage



TAMMO VAN LESSEN
Principal Consultant



SASCHA SELZER
Senior Consultant

Agenda

- What is Backstage?
- Core Features
 - Catalog
 - Software Templates
 - Tech Docs
- Authentication
- Plugins



What is Backstage?

Born out of necessity

- "Where's the API for that service we're all supposed to be using?"
- "What version of that framework is everyone on?"
- "This service isn't responding, who owns it?"
- "I can't find documentation for anything!"



March 2022



CLOUD NATIVE
COMPUTING FOUNDATION

WTH is Backstage?

- Open platform for building developer portals
- Centralizes project/service information by crawling and collecting service definitions from different locations
- Aggregates them in a service catalog
- Integrate 3rd party services/content via plugins
- Everything is a plugin
- No hooks, just code

OUBE

- It's not a product, no docker file, no binary.
- It's a platform to run your very own developer portal from the beginning
- Still easy to start:



```
$ npx @backstage/create-app@latest
```

Backstage Catalog

Kind

Component

Type

all

PERSONAL

Owned

3

Starred

0

BACKSTAGE

All

15

OWNER

LIFECYCLE

Owned (3)

NAME	SYSTEM	OWNER	TYPE	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
playback-order	audio-playback	user:guest	service	production	Playback Order	java playback	  
searcher		user:guest	service	production	Searcher	go	  
shuffle-api	audio-playback	user:guest	service	production	Shuffle API	go	  

Core Features

Five Core Pillars of Backstage

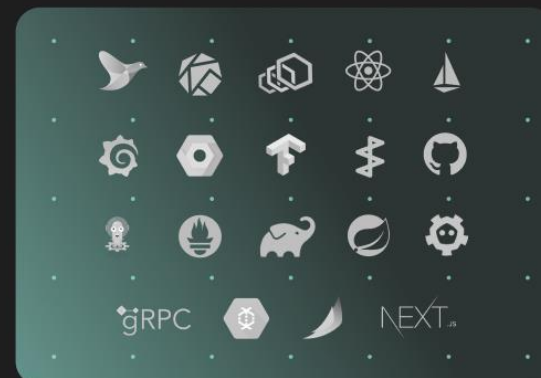
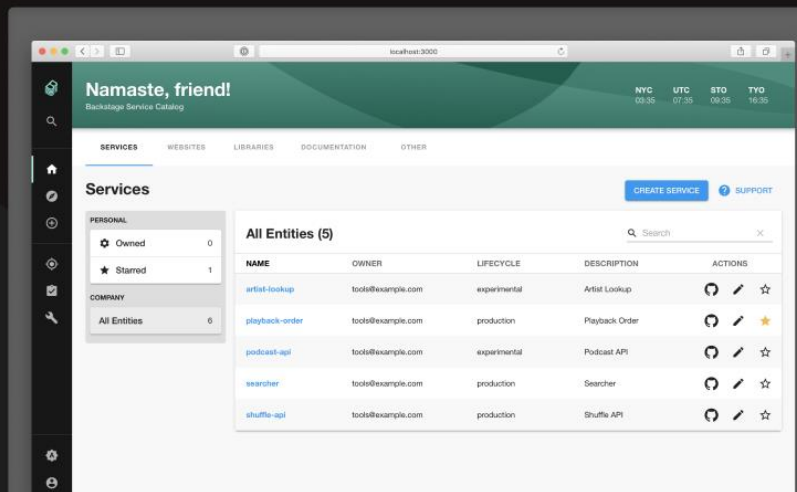
- **Software Catalog**
- Kubernetes
- **Software Templates**
- Backstage Search
- **Tech Docs**

Software Catalog

Backstage

9 lines (9 sloc) 182 Bytes

```
1 apiVersion: backstage.io/v1alpha1
2 kind: Component
3 metadata:
4   name: podcast-api
5   description: Podcast API
6 spec:
7   type: service
8   lifecycle: experimental
9   owner: tools@example
```

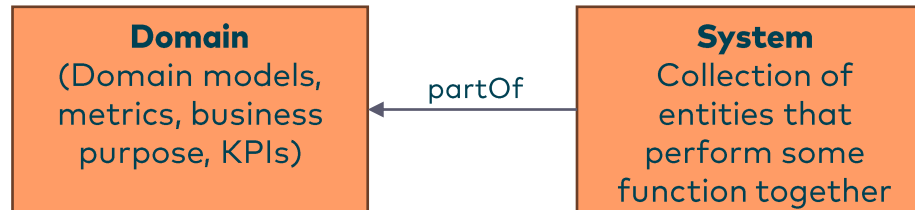


**SERVICE
DEFINITIONS**

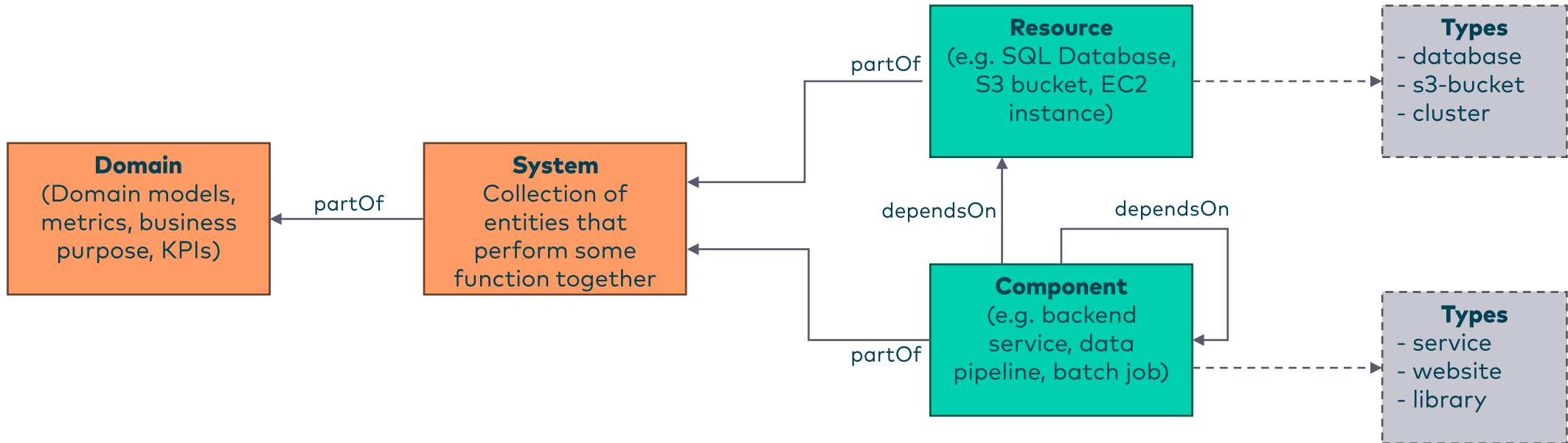
**BACKSTAGE
SERVICE CATALOG**

**INTEGRATED TOOLING
VIA PLUGINS**

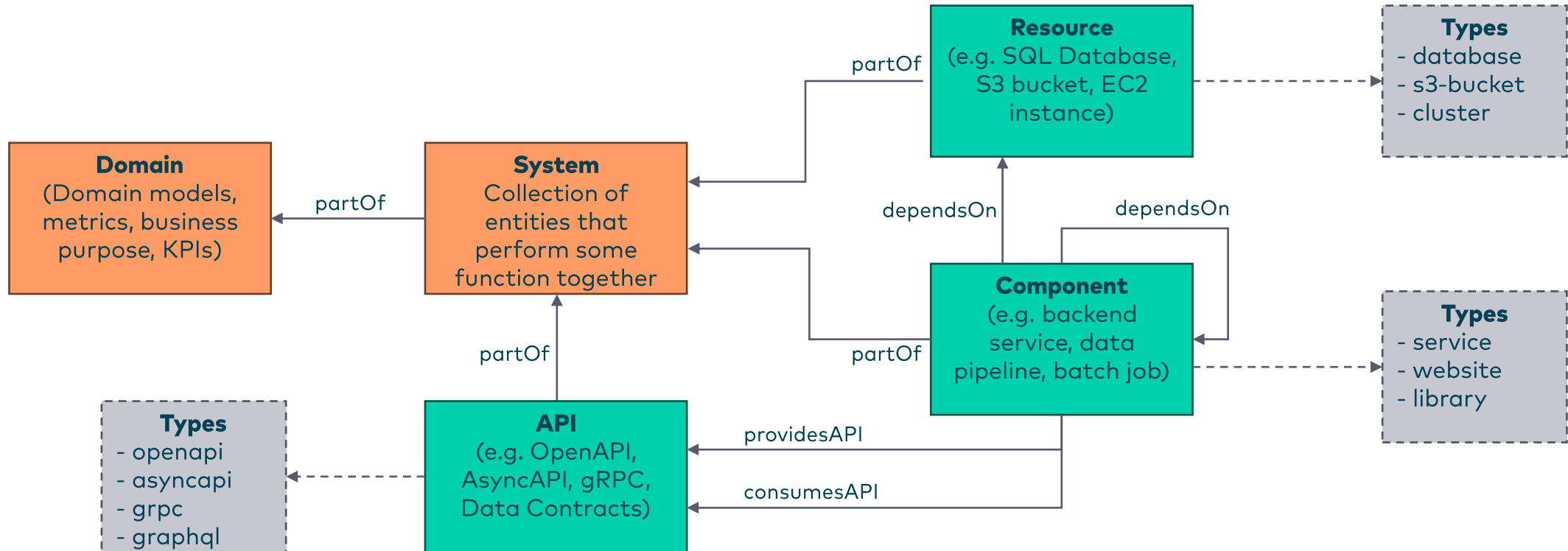
Catalog Entities



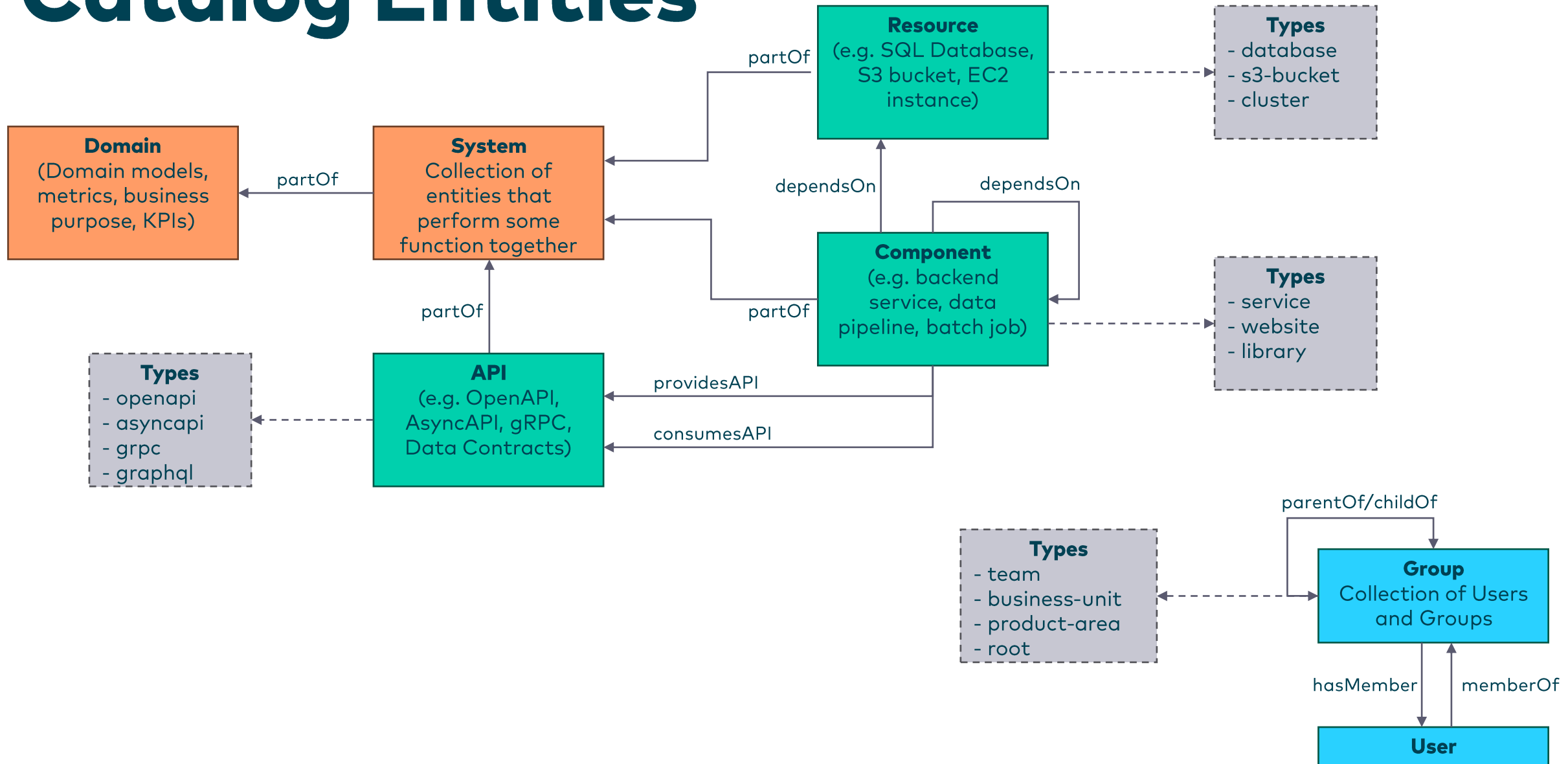
Catalog Entities



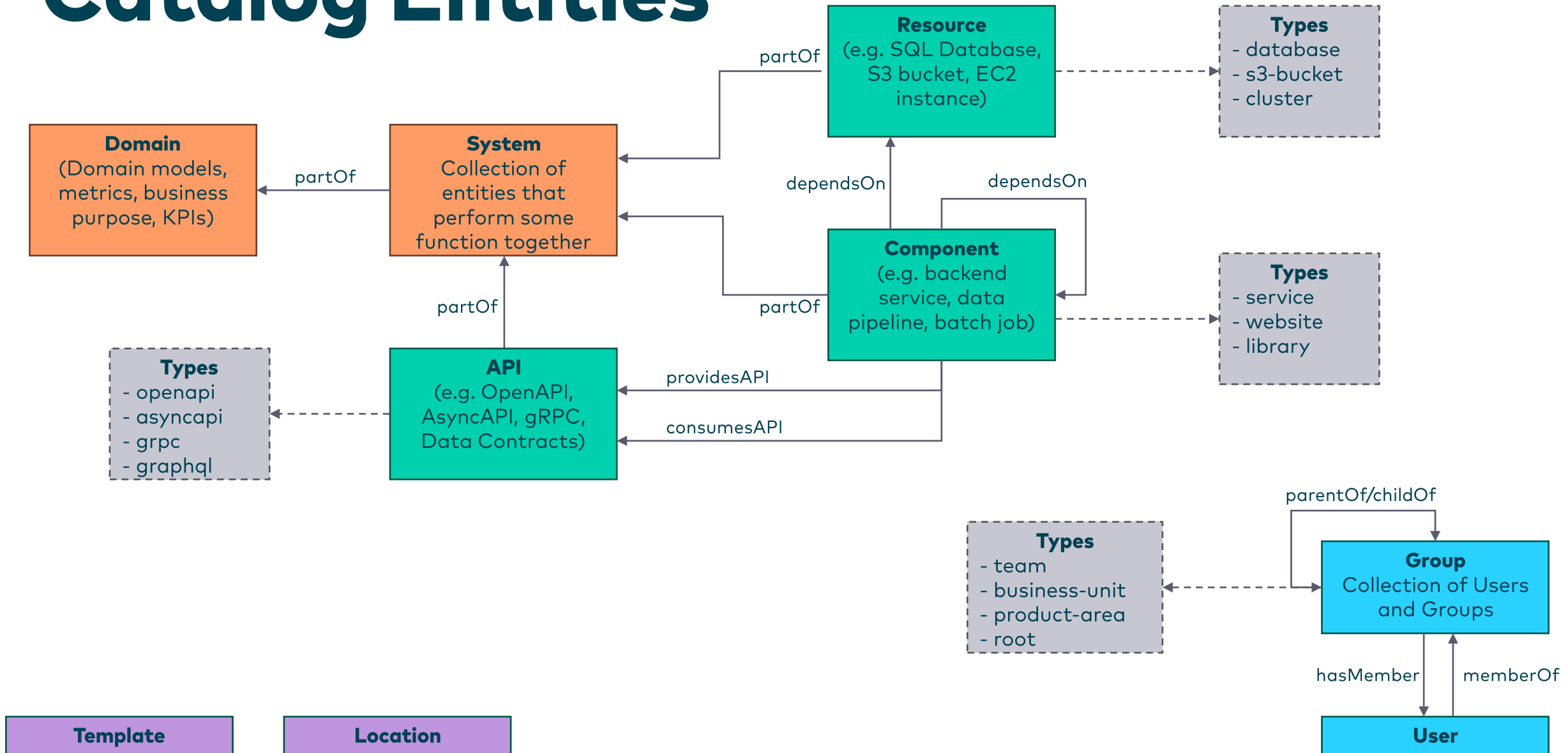
Catalog Entities



Catalog Entities



Catalog Entities



... are described in YAML

```
apiVersion: backstage.io/v1alpha1
kind: Component
metadata:
  name: artist-web
  description: The place to be, for great artists
  labels:
    example.com/custom: custom_label_value
  annotations:
    example.com/service-discovery: artistweb
    circleci.com/project-slug: github/example-org/artist-website
  tags:
    - java
  links:
    - url: https://admin.example-org.com
      title: Admin Dashboard
      icon: dashboard
      type: admin-dashboard
spec:
  type: website
  lifecycle: production
  owner: artist-relations-team
  system: public-websites
```



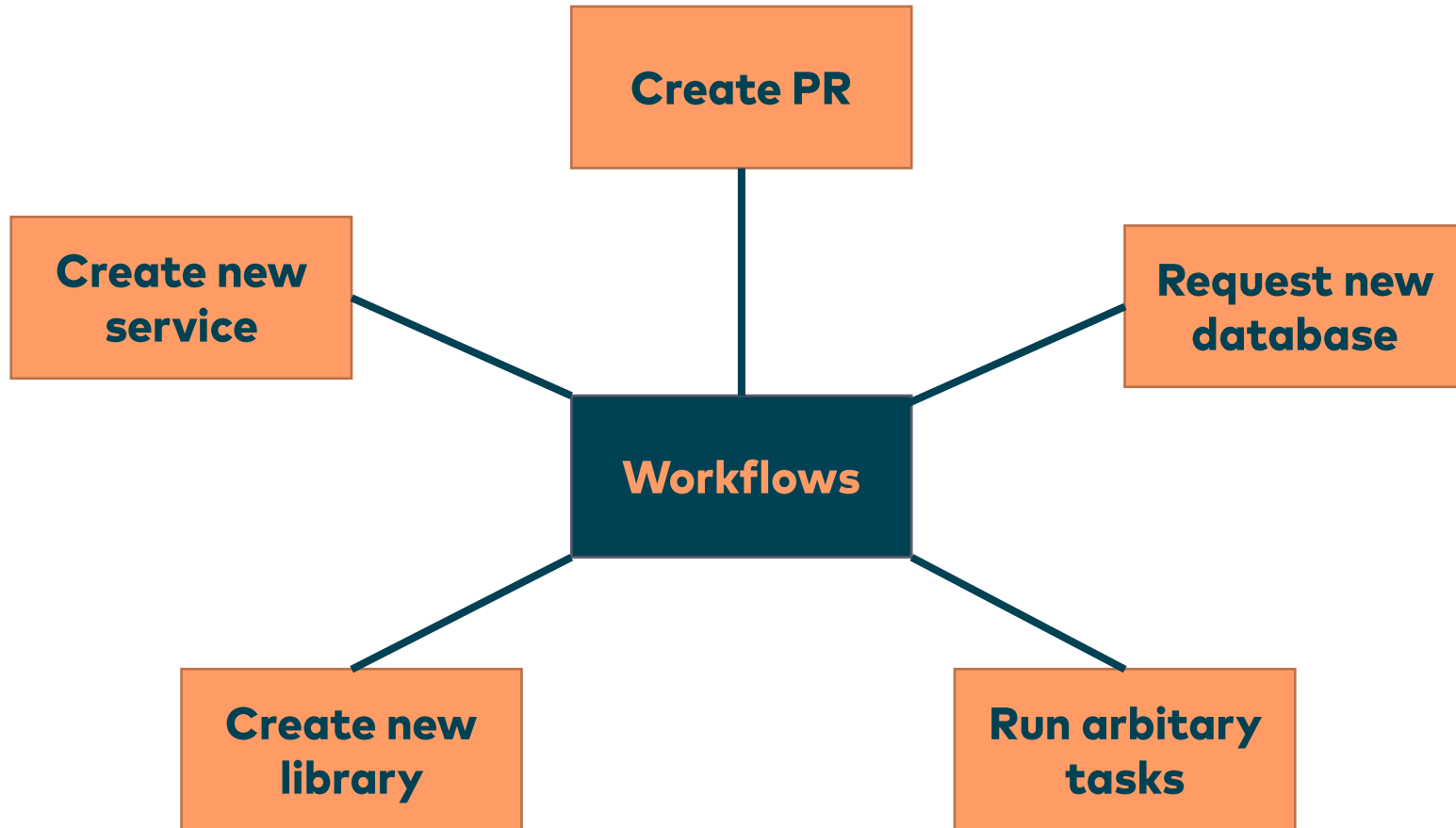
...and put into the repository as catalog-info.yaml next to the source code.
External entity providers are possible, too.

Stop!
Demo Time

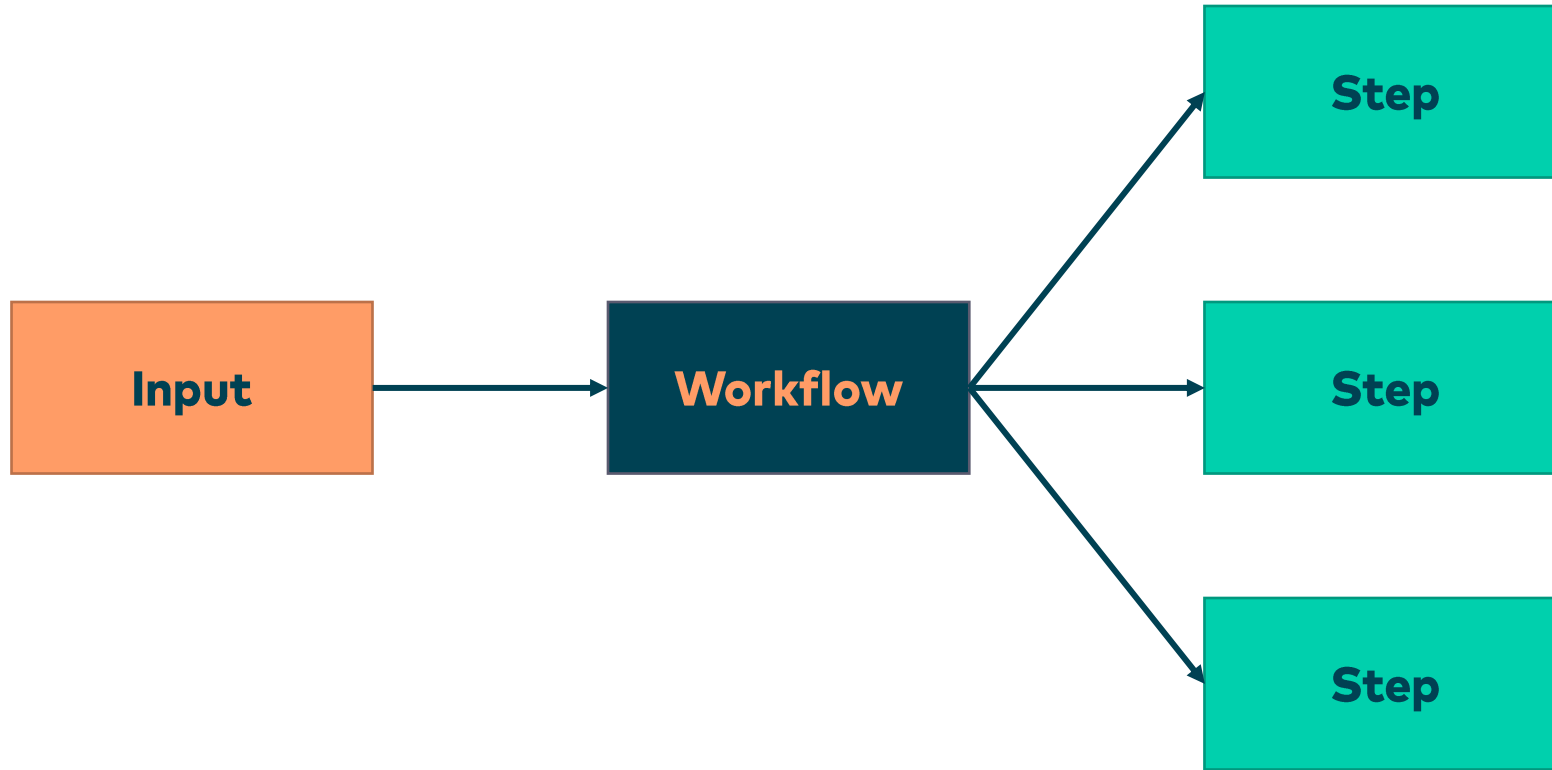
Software Templates

Automate common developer workflows

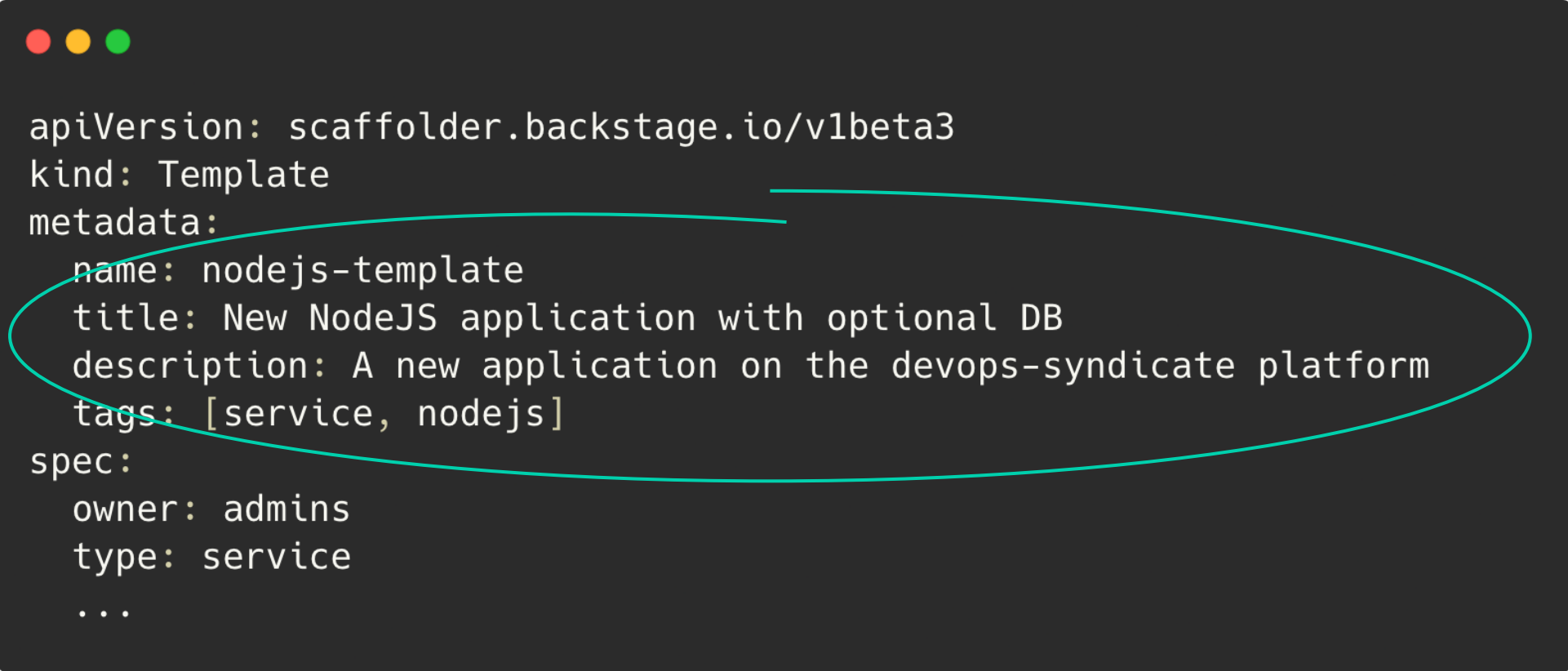
Software Templates



Software Templates



Software Template Resource



```
apiVersion: scaffolder.backstage.io/v1beta3
kind: Template
metadata:
  name: nodejs-template
  title: New NodeJS application with optional DB
  description: A new application on the devops-syndicate platform
  tags: [service, nodejs]
spec:
  owner: admins
  type: service
  ...
```

Software Template Resource

Templates

service ☆

New Golang application with optional DB

DESCRIPTION

A new application on the devops-syndicate platform

OWNER

[admins](#)

TAGS

service golang

 CHOOSE

service ☆

New NodeJS application with optional DB

DESCRIPTION

A new application on the devops-syndicate platform

OWNER

[admins](#)

TAGS

service nodejs

 CHOOSE

service ☆

New Simple Data product

DESCRIPTION

A new simple data product reading from Kafka and transform with simple sql statement

OWNER

[data-platform-team](#)

TAGS

data

 CHOOSE



```
apiVersion: scaffolder.backstage.io/v1beta3
```

```
kind: Template
```

```
...
```

```
spec:
```

```
...
```

```
parameters:
```

```
- title: Basic information
```

```
  required:
```

```
    - name
```

```
    - owner
```

```
  properties:
```

```
    name:
```

```
      title: Name
```

```
      type: string
```

```
      description: Unique name of the component
```

```
      ui:autofocus: true
```

```
      ui:options:
```

```
        rows: 5
```

```
    owner:
```

```
      title: Owner
```

```
      type: string
```

```
      description: Owner of the component
```

```
      ui:field: OwnerPicker
```

```
      ui:options:
```

```
        allowedKinds:
```

```
          - Group
```

New NodeJS application with optional DB

1 Basic information

Name*

Unique name of the component

Owner*

Owner of the component

BACK

NEXT STEP

2 Location

3 Ingress

4 Database

```
apiVersion: scaffolder.backstage.io/v1beta3
kind: Template
```

```
...
```

```
spec:
```

```
...
```

```
steps:
```

```
- id: fetch-base
  name: Fetch Base
  action: fetch:template
```

```
  input:
```

```
    url: https://github.com/devops-syndicate/backstage-templates/blob/main/skeletons/nodejs
```

```
    targetPath: ./app
```

```
    values:
```

```
      name: ${ parameters.name }
```

```
      owner: ${ parameters.owner }
```

```
      createIngress: ${ parameters.createIngress }
```

```
      createDB: ${ parameters.createDB }
```

location of template files

values for template files

Template files

```
apiVersion: backstage.io/v1alpha1
kind: Component
metadata:
  name: "${{ values.name }}"
  annotations:
    github.com/project-slug: "devops-syndicate/${{ values.name }}"
    argocd/app-name: "${{ values.name }}"
    backstage.io/techdocs-ref: dir:.
  links:
    - title: Link to ArgoCD
      url: http://argo-cd.127.0.0.1.nip.io/applications/${{ values.name }}
spec:
  type: service
  owner: "${{ values.owner }}"
  lifecycle: production
  providesApis:
    - "${{ values.name }}"-api
```

usage of values in template files

-  Fetch RDS Database Template Skipped
-  Open PR for new database Skipped
-  Fetch Base 2 seconds
-  Publish 5 seconds
-  Register Service 1 second
-  Register API 0 seconds

 [Open Github Repository](#)

 [Open Catalog](#)

START OVER

Search



```
1 2023-04-16T17:02:38.289Z Beginning step Register API
2 2023-04-16T17:02:38.292Z info: Registering https://github.com/devops-syndicate/app-d/tree/main/api/api.yaml in the cata
3 2023-04-16T17:02:38.726Z Finished step Register API
```

OVERVIEW

CI/CD

API

DEPENDENCIES

DOCS

About



VIEW
SOURCE



VIEW
TECHDOCS

DESCRIPTION

No description

OWNER

team-a

SYSTEM

No System

TYPE

service

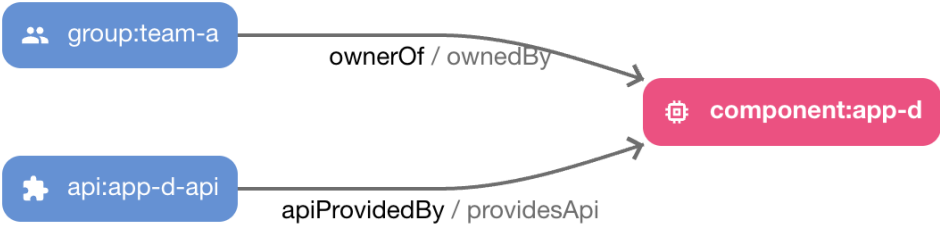
LIFECYCLE

production

TAGS

No Tags

Relations



View graph →

Default actions

- Load skeletons of code
- Template in some variables
- Publish template to some location
- Delete/Change files in workspace
- Register new component in backstage
- Add label to pull requests

Custom action

```
import { createTemplateAction } from '@backstage/plugin-scaffolder-node';
import fs from 'fs-extra';
import { z } from 'zod';

export const createNewFileAction = () => {
  return createTemplateAction({
    id: 'acme:file:create',
    schema: {
      input: z.object({
        contents: z.string().describe('The contents of the file'),
        filename: z
          .string()
          .describe('The filename of the file that will be created'),
      }),
    },

    async handler(ctx) {
      await fs.outputFile(
        `${ctx.workspacePath}/${ctx.input.filename}`,
        ctx.input.contents,
      );
    },
  });
};
```

Tech Docs

Docs as Code

- MkDocs based Static Site Generator integrated into Backstage
- Docker-based Builds, easy setup
- Ingests docs into integrated search framework
- Layout is customizable

The screenshot displays the Backstage documentation website. On the left is a dark sidebar with the Backstage logo and navigation links: Home, APIs, Docs, Explore, Tech Radar, and Cost Insights. The main header is pink with the text 'Documentation / Backstage' and 'Backstage' in large font, followed by the subtitle 'Main documentation for Backstage features and platform APIs'. On the right side of the header are filters for 'Component' (backstage), 'Owner' (cnf), 'Lifecycle' (experimental), and '<>' (Source). Below the header is a search bar labeled 'Search backstage docs'. The main content area is titled 'Overview' and 'What is a Service Catalog?'. It describes the Backstage Service Catalog as a centralized system for tracking ownership and metadata for software in an ecosystem. A 'Table of contents' is visible on the right side of the main content area.

Backstage

Documentation / Backstage

Backstage

Main documentation for Backstage features and platform APIs

Component backstage Owner cnf Lifecycle experimental <> Source

Search backstage docs

Overview

What is a Service Catalog?

The Backstage Service Catalog — actually, a software catalog, since it includes more than just services — is a centralized system that keeps track of ownership and metadata for all the software in your ecosystem (services, websites, libraries, data pipelines, etc). The catalog is built around the concept of [metadata YAML files](#) stored together with the code, which are then harvested and visualized in Backstage.

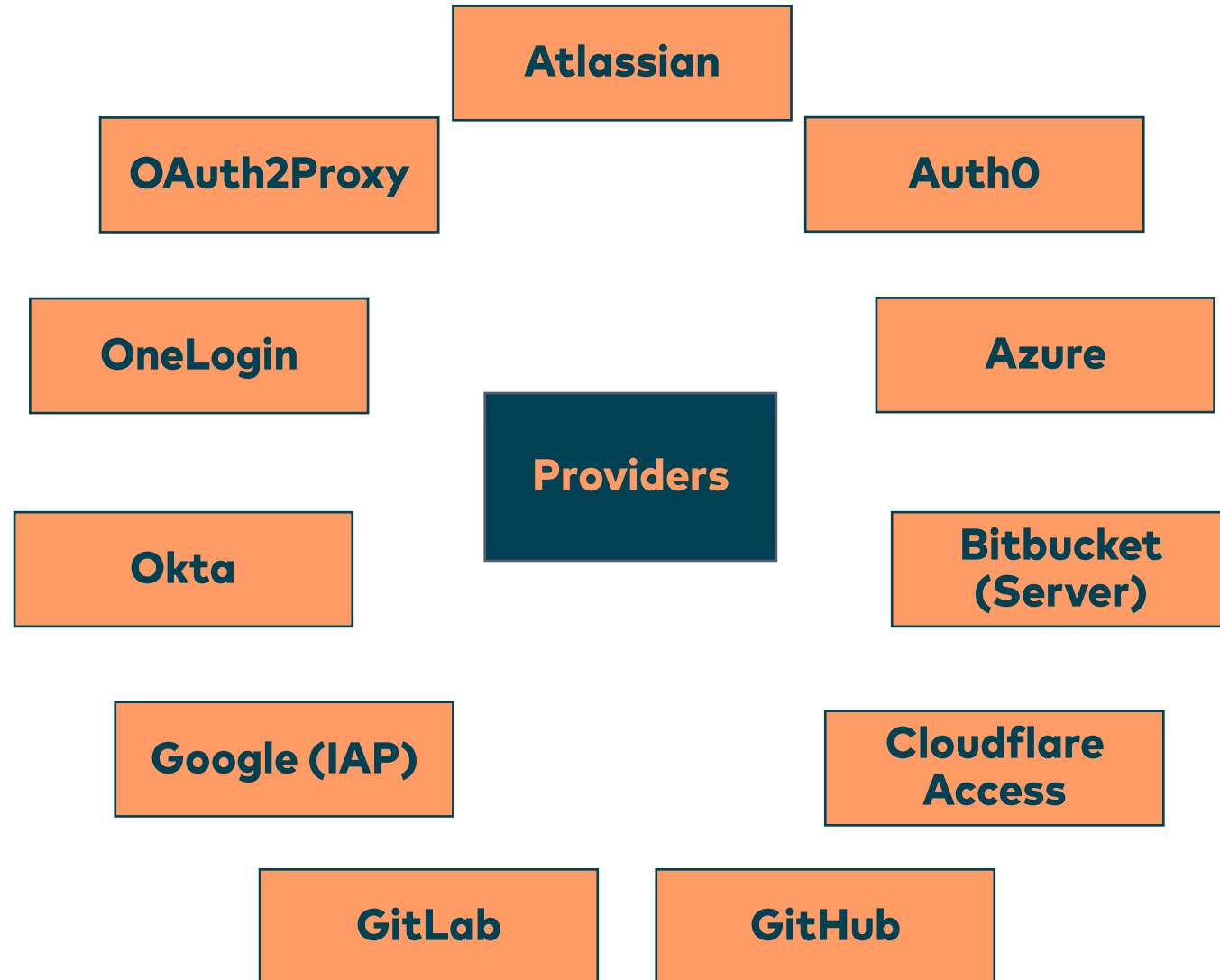
Table of contents

- What is a Service Catalog?
- How it works
- Getting Started
- Adding components to the catalog
- Manually register components

Stop!
Demo Time

Authentication

Authentication Providers



Login Page

```
const app = createApp({
  components: {
    SignInPage: props => (
      <SignInPage
        {...props}
        auto
        provider={{
          id: 'github-auth-provider',
          title: 'GitHub',
          message: 'Sign in using GitHub',
          apiRef: githubAuthApiRef,
        }}
      />
    ),
  },
  ...
});
```

Login Page

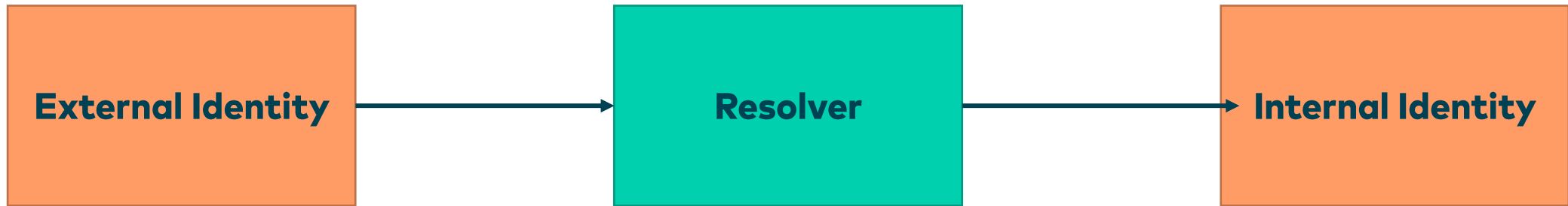
DevOps Syndicate - Center

GitHub

Sign in using GitHub

SIGN IN

Identity Resolver



Simple Identity Resolver

```
signIn: {  
  resolver(_, ctx) {  
    const userRef = 'user:default/guest'  
    return ctx.issueToken({  
      claims: {  
        sub: userRef,  
        ent: [userRef],  
      },  
    }),  
  },  
},
```

Github Identity Resolver



```
signIn: {  
  resolver: providers.github.resolvers.usernameMatchingUserEntityName(),  
},
```

Github Identity Resolver

GENERAL

AUTHENTICATION PROVIDERS

FEATURE FLAGS

Profile



tommy1199

⋮

Backstage Identity

User Entity:

user:default/tommy1199

Ownership Entities:

user:default/tommy1199

group:default/admins

group:default/data-platform-team

group:default/team-a

Appearance

Theme

Change the theme mode

LIGHT THEME 

DARK THEME 

AUTO 

Pin Sidebar

Prevent the sidebar from collapsing

☒

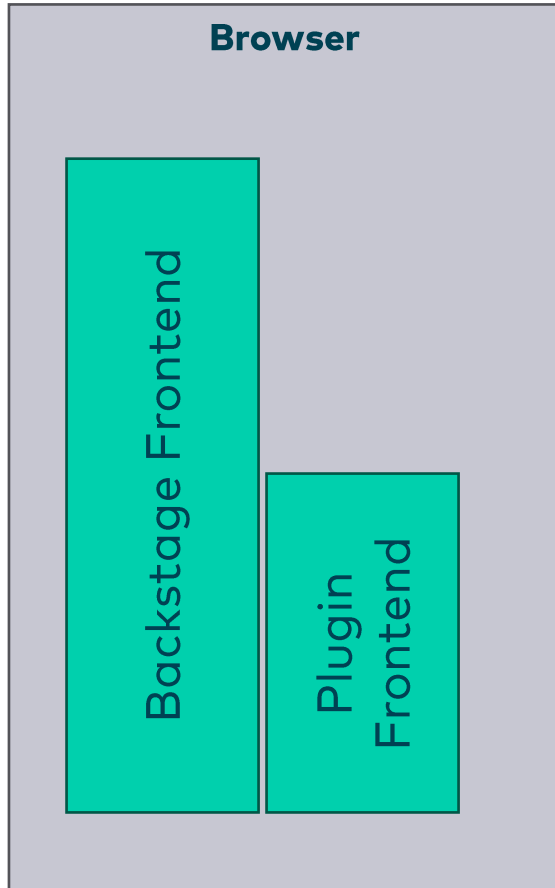
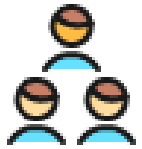
Stop!
Demo Time

Plugins

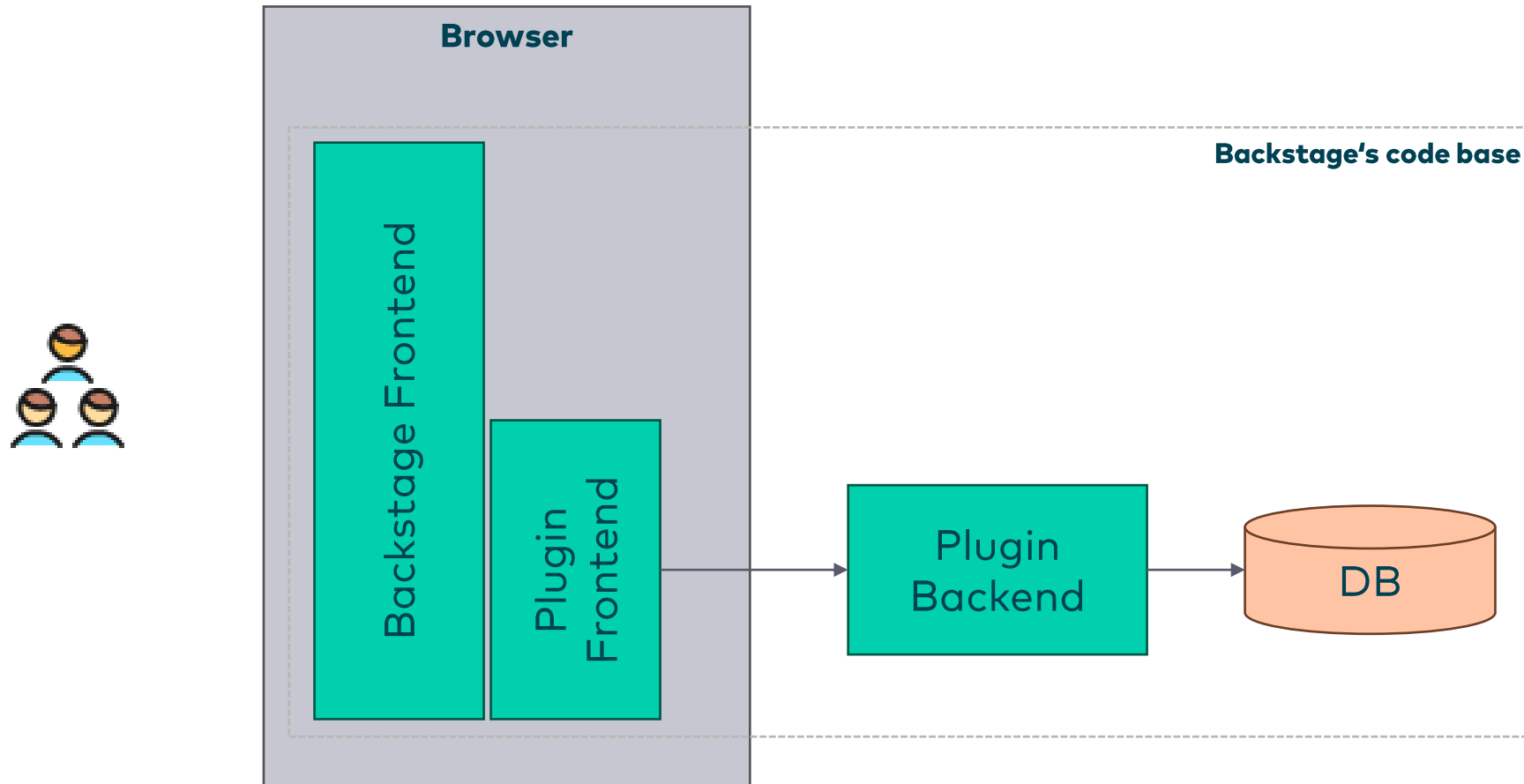
Extensibility API

- Access to Entities
- Access to Logged-In User info
- Access to the catalog
- Access to configuration
- Own UI components
- To communicate with external systems, use the proxy-backend to avoid CORS
- Can be a full page plugin or integrated in the software catalog as tile or tab

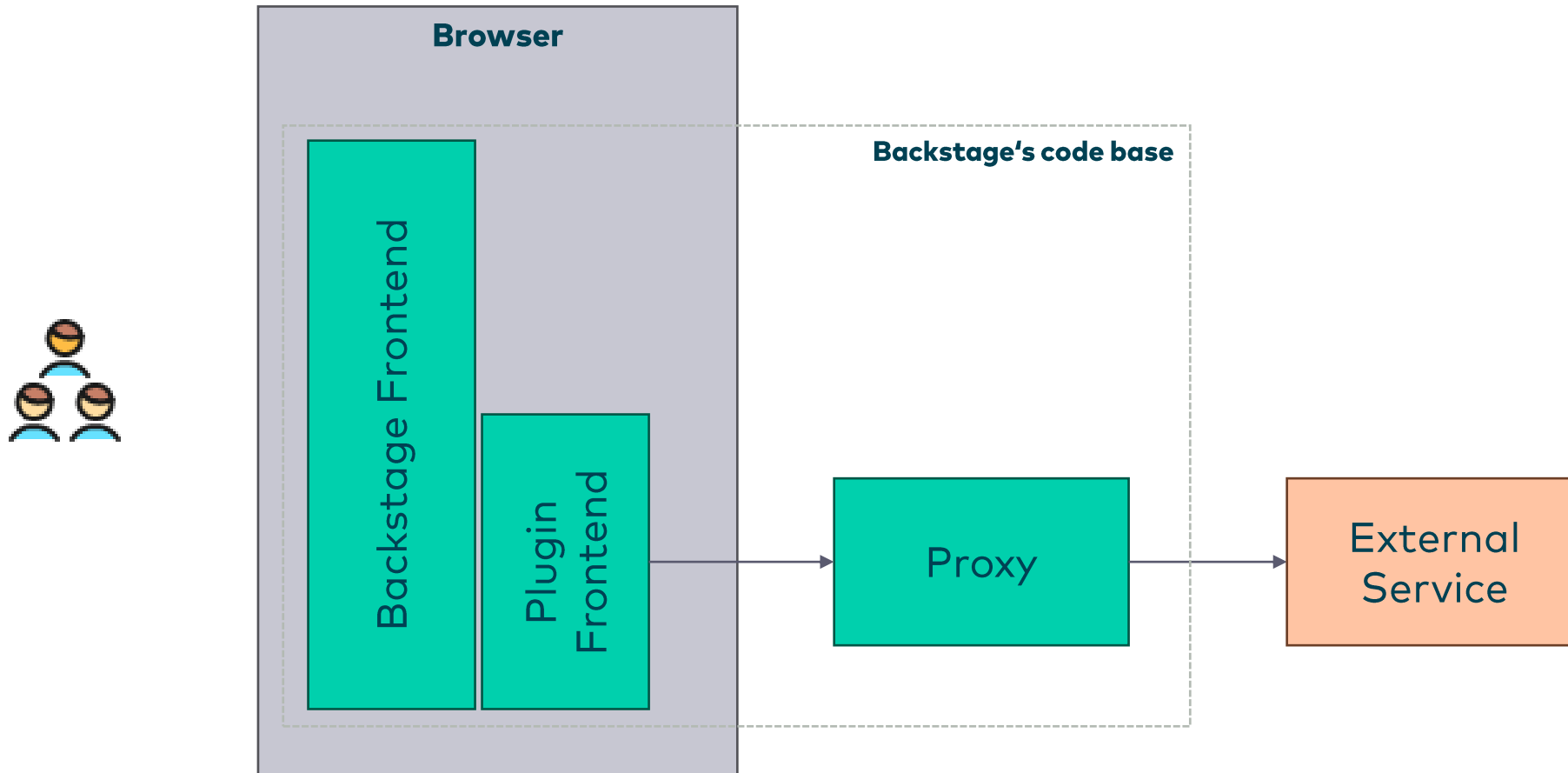
Frontend Integration



Integration with own Backend



Integration via Proxy



Plugin Marketplace

 Architecture Decision Records by Phil Kuang Discovery Browse your project's ADRs. Explore	 Airbrake by Simply Business Monitoring Access Airbrake error monitoring and other integrations from within Backstage. Explore	 Allure Reports by Deepak Bhardwaj Reporting View Allure reports for your components in Backstage. Explore	 Analytics Module: Google Analytics by Spotify Analytics Track usage of your Backstage instance using Google Analytics. Explore
 Announcements by K-Phoen Discovery Write and share announcements within Backstage. Explore	 API Docs by SDA SE Discovery Components to discover and display API entities as an extension to the catalog plugin. Explore	 API Linter by Zalando Linting API Linter is a quality assurance tool that checks the compliance of API's specifications to Zalando's API rules. Explore	 API Spectral Linter by dweber019 Linting API Spectral Linter is a quality assurance tool that checks the compliance of API's specifications Spectral rule sets. Explore
 Apollo Explorer by unredundant Debugging Integrates Apollo Explorer graphs as a tool to browse GraphQL API. Explore	 Argo CD by roadie.io CI/CD View Argo CD status for your projects in Backstage. Explore	 AWS CloudFormation by Purple Technology Infrastructure Load Backstage entities from AWS. Explore	 AWS Code Services by Amazon Web Services CI/CD Explore

<https://backstage.io/plugins>

Gitlab Plugin

OVERVIEW

GITLAB

People

Contributors go to Contributors →



Languages



● Lua - 83.36%

● Makefile - 12.77%

● BibTeX - 3.86%

Merge requests statistics

Avg Time Until Merge 6 days

Merged To Total Ratio 5%

20
Number of MRs

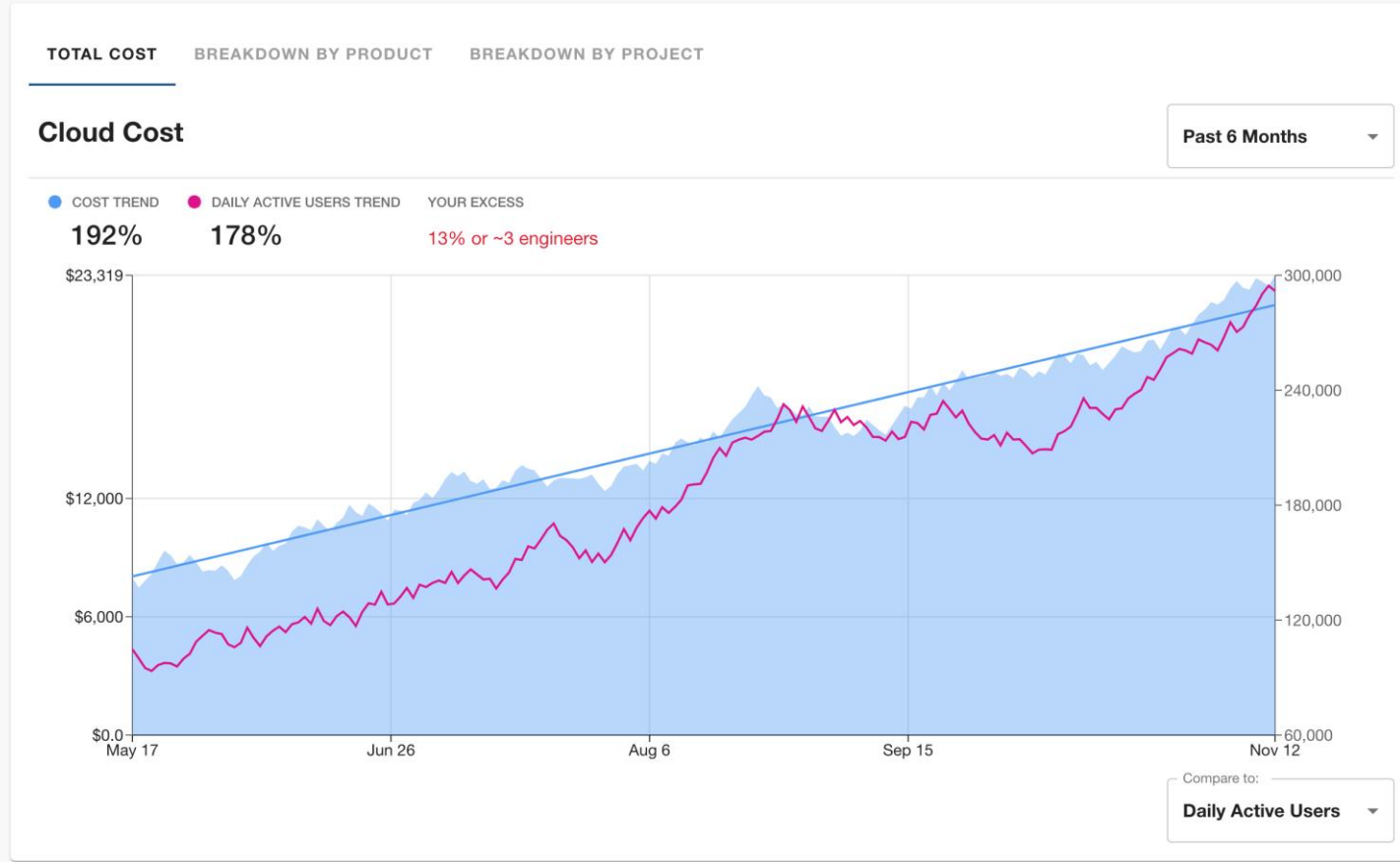
Gitlab Pipelines: migrating-2-cloud-native-primer

Filter

PIPELINE_ID	STATUS	BRANCH	WEB URL	CREATED AT	DURATION
98770	● success	master	https://gitlab.innoq.com/innocent/primers/migrating-2-cloud-native-primer/-/pipelines/98770	a month ago	54s
97433	● success	master	https://gitlab.innoq.com/innocent/primers/migrating-2-cloud-native-primer/-/pipelines/97433	2 months ago	50s

Cost Insights Plugin

- Cost Overview
- Action Items ²
- Compute Engine
- Cloud Storage
- BigQuery



Conclusions

Conclusions

- Backstage provides a decent framework to easily build and run developer portals within your organization
- Extensible catalog meta model
- Extensible tooling via Plugins
- Single entrypoint of knowledge without the downsides of a central catalog repository

Danke! Fragen?



Tammo van Lessen
tammo.van-lessen@innoq.com
@taval



Sascha Selzer
sascha.selzer@innoq.com
@tommy1199

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstrasse 13
20095 Hamburg

Erftstr. 15-17
50672 Köln

Königstorgaben 11
90402 Nürnberg