

Shifting Spotify Engineering from spreadsheets to Backstage

Agenda

- Spotify's Journey
- Introduction to Backstage
- Not just an Inventory
- Demo
- Adoption Strategies



Johan Haals



Patrik Oldsberg

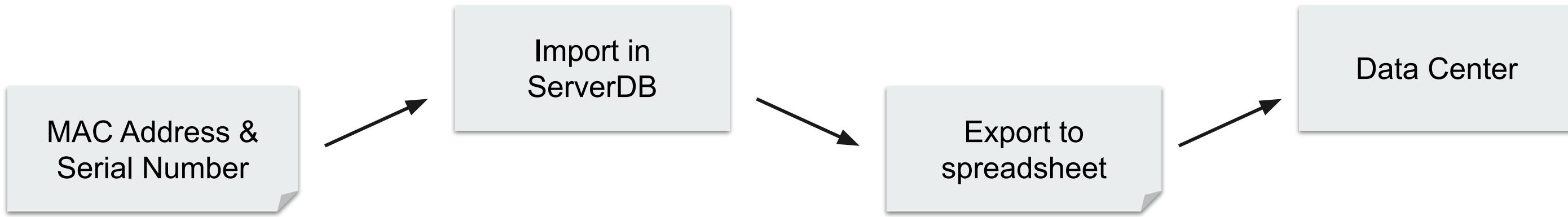


The Stone Age

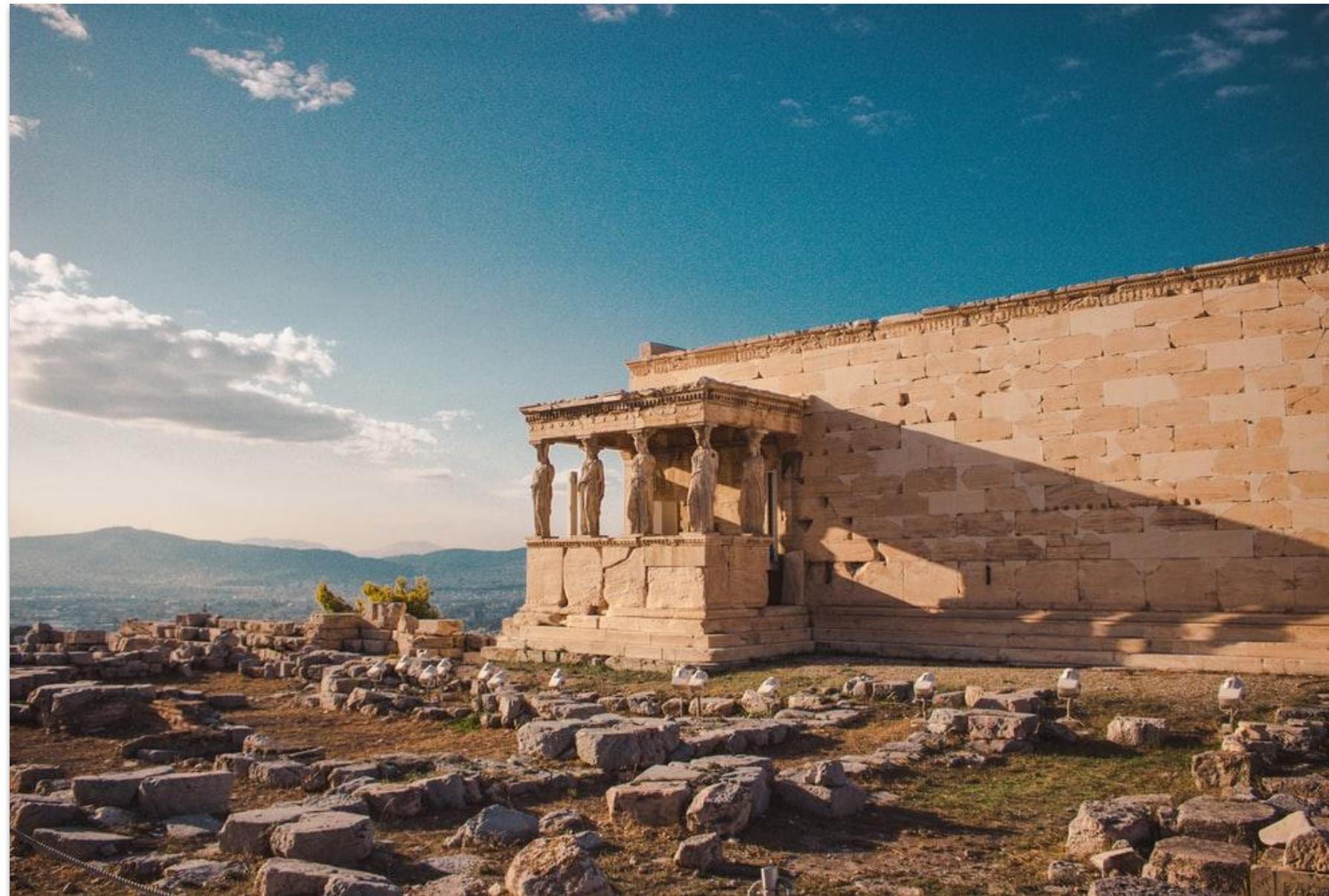


2010

6 people running all operations

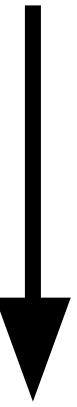


The Bronze Age

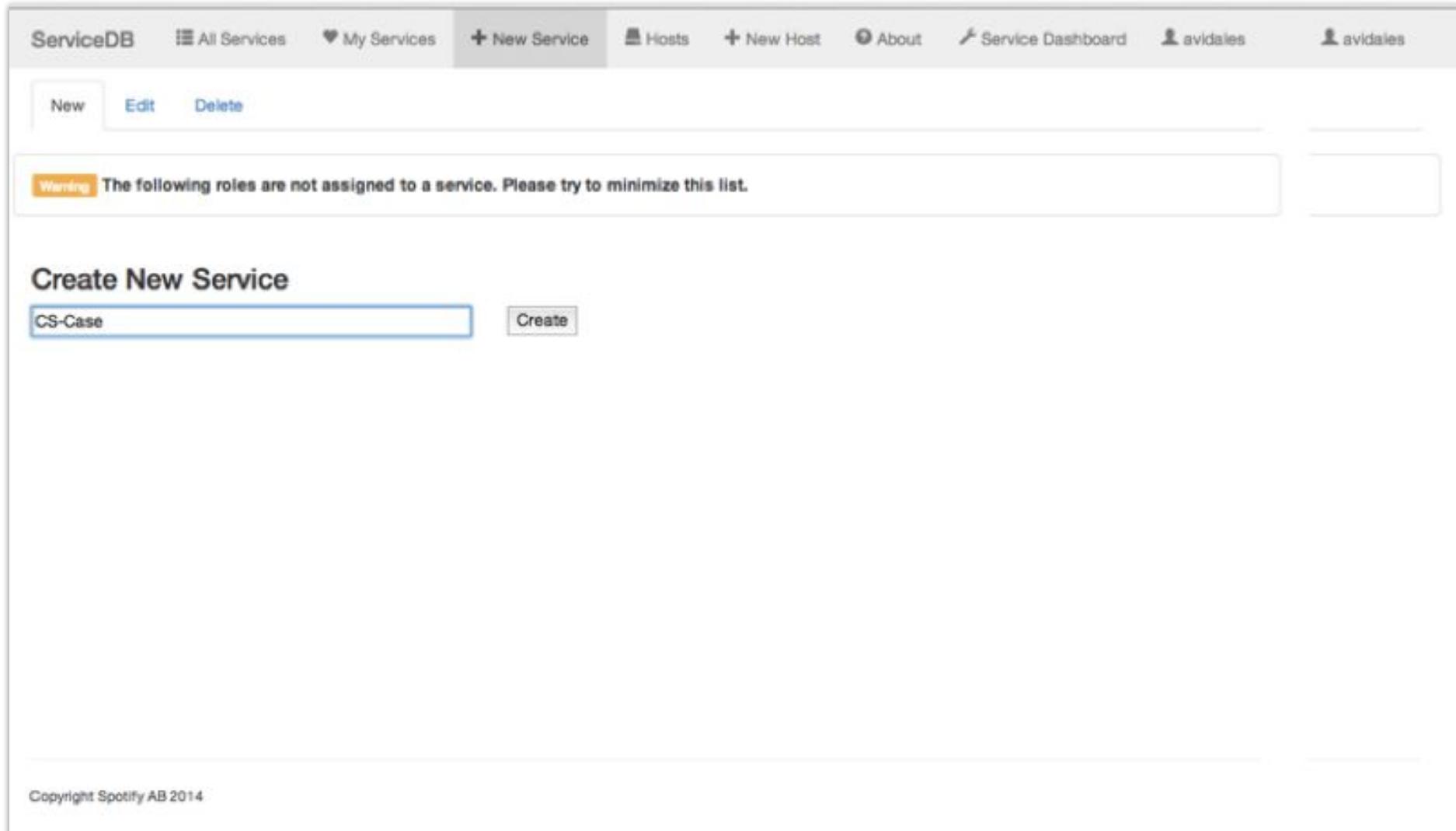


Mid 2010's

ServerDB



ServiceDB



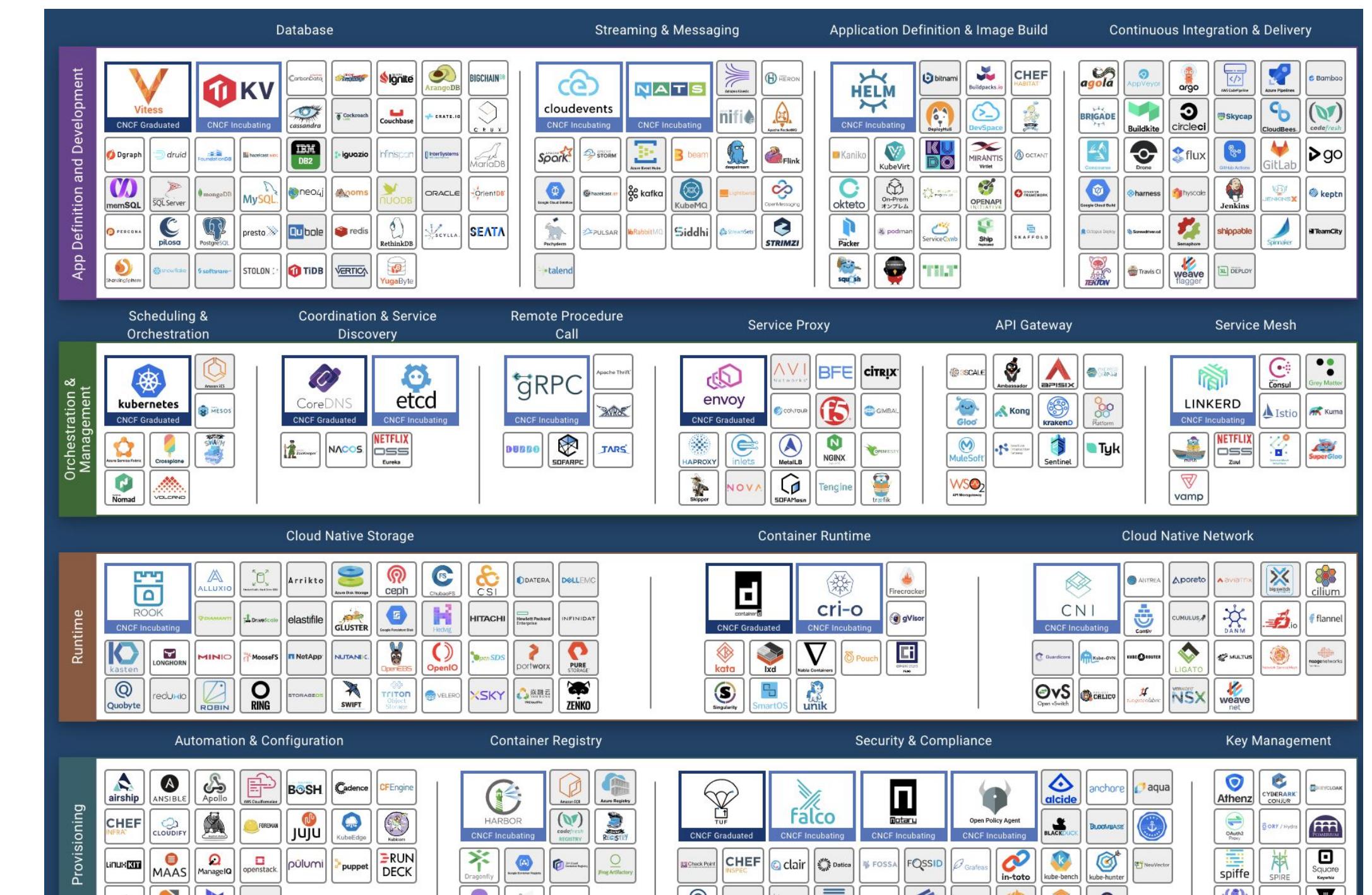
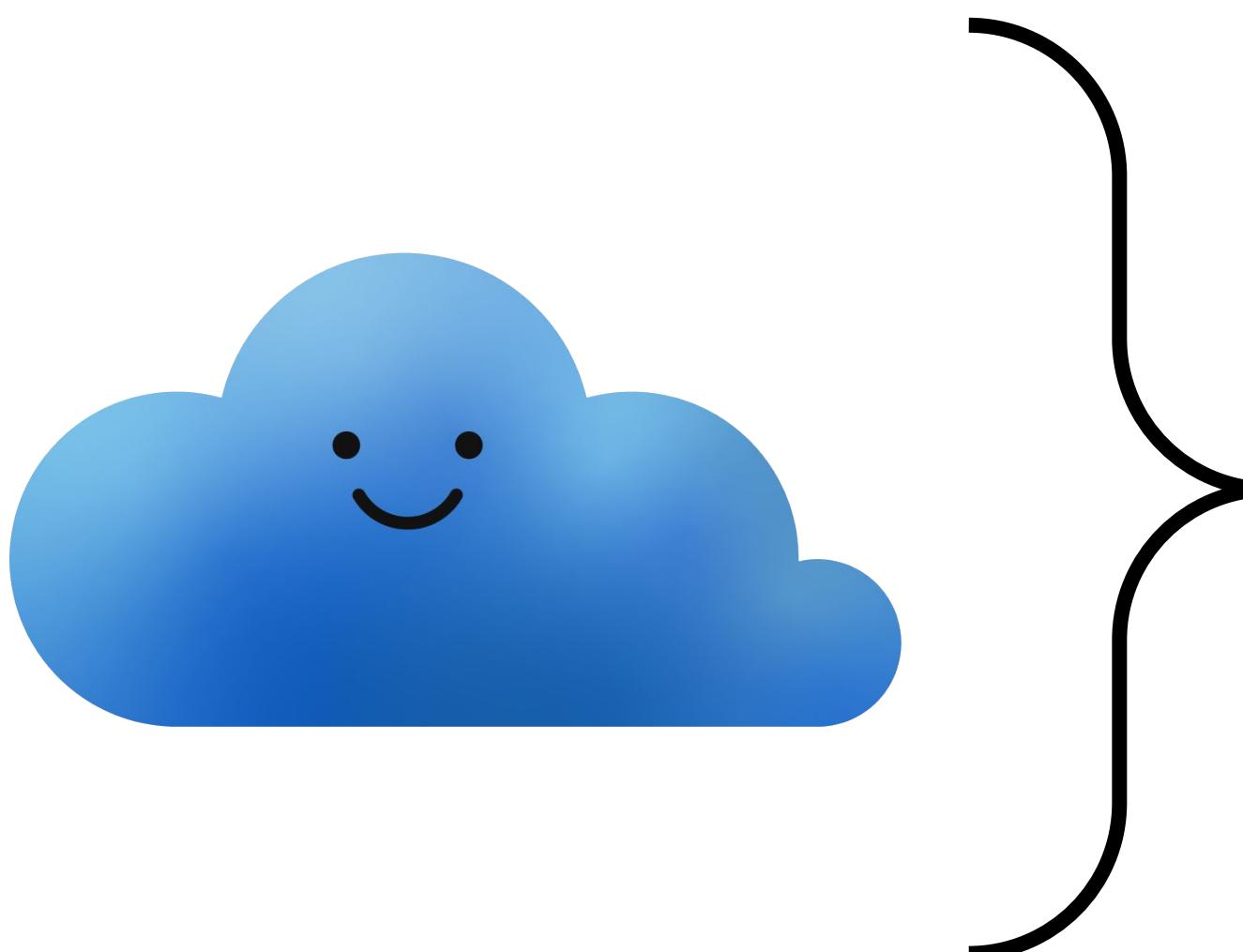
The screenshot shows the ServiceDB application's interface. At the top, there is a navigation bar with links: ServiceDB, All Services, My Services, + New Service, Hosts, + New Host, About, Service Dashboard, and two user profiles. Below the navigation bar, there are three buttons: New (highlighted), Edit, and Delete. A warning message box is displayed, stating: "Warning: The following roles are not assigned to a service. Please try to minimize this list." Below the message, there is a section titled "Create New Service" with a text input field containing "CS-Case" and a "Create" button. At the bottom of the page, there is a copyright notice: "Copyright Spotify AB 2014".

The Iron Age



Late 2010's

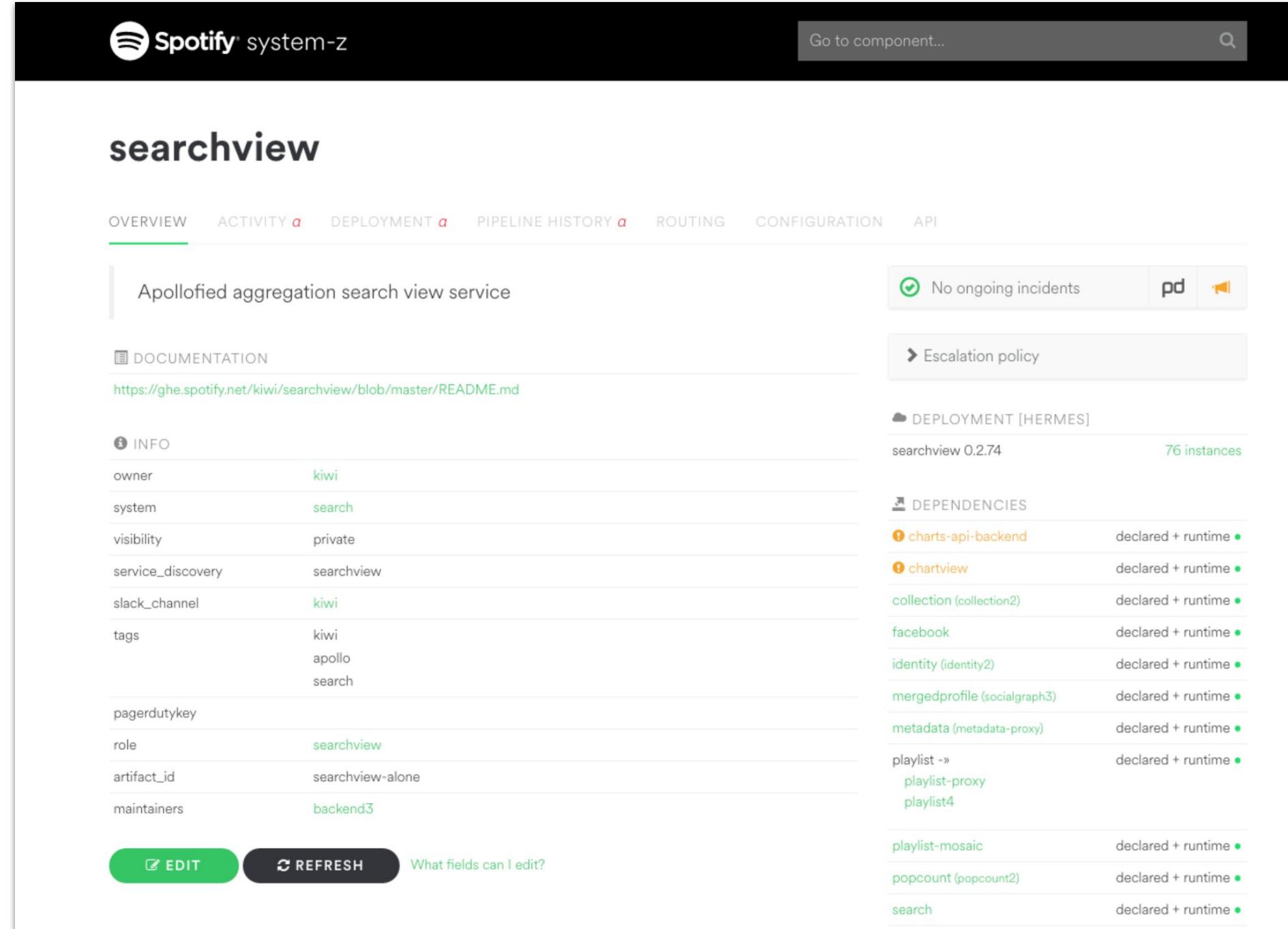
From the cloud to a world of new tools



ServiceDB



system-z



The screenshot shows the Spotify Backstage interface for the component **searchview**. The top navigation bar includes the Spotify logo and the text "Spotify system-z". A search bar with the placeholder "Go to component..." and a magnifying glass icon is also present.

The main content area displays the following information:

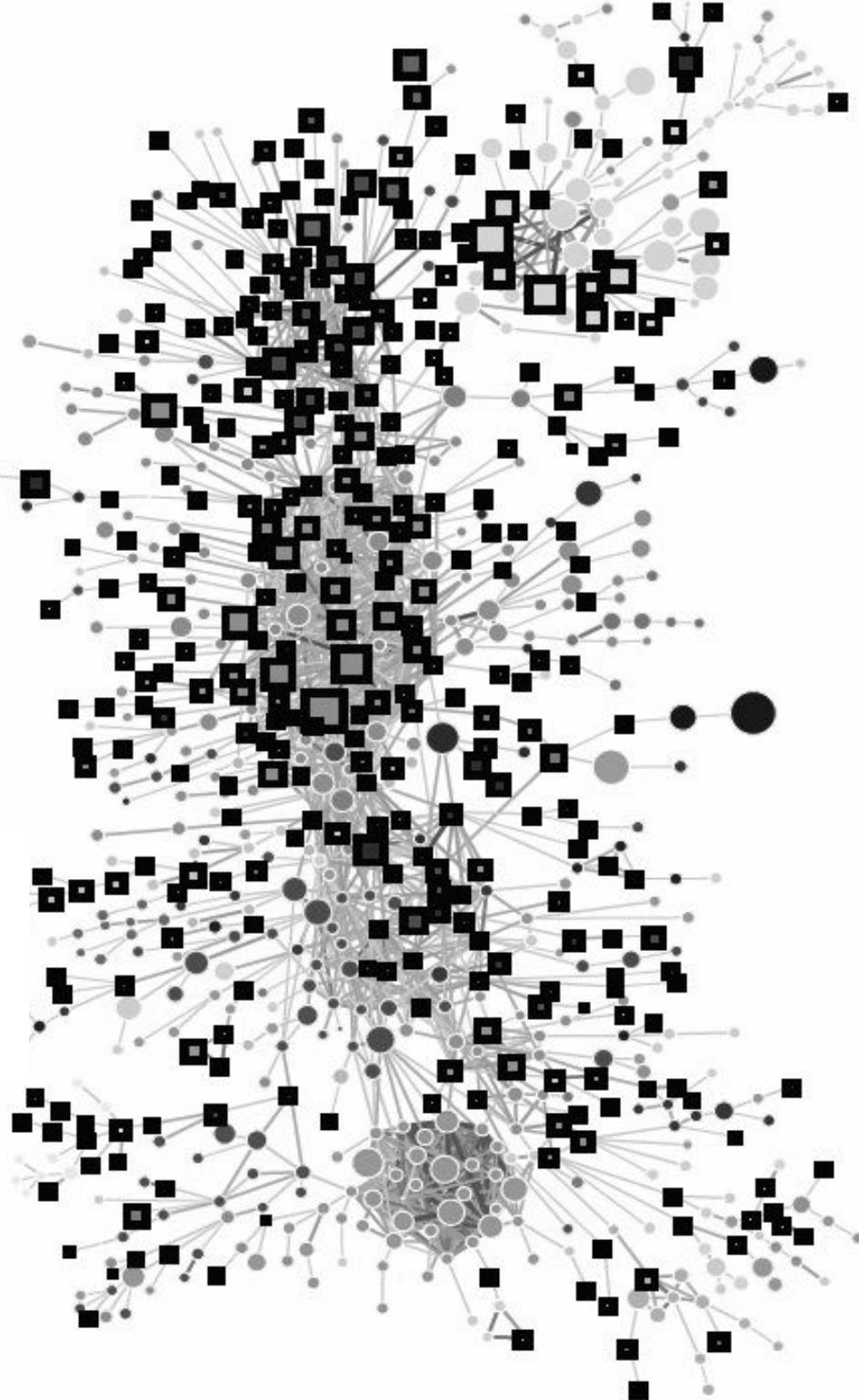
- OVERVIEW**:
 - Apollofied aggregation search view service
 - DOCUMENTATION**: <https://ghe.spotify.net/kiwi/searchview/blob/master/README.md>
 - INFO**:

owner	kiwi
system	search
visibility	private
service_discovery	searchview
slack_channel	kiwi
tags	kiwi apollo search
pagerdutykey	
role	searchview
artifact_id	searchview-alone
maintainers	backend3
 - EDITION**: Buttons for **EDIT** and **REFRESH**, and a link "What fields can I edit?"
- ROUTING**: Shows "No ongoing incidents" with a green checkmark, a "pd" button, and an orange bell icon.
- CONFIGURATION**: A link to "Escalation policy".
- API**: A link to "Deployment [HERMES]" showing "searchview 0.2.74" with "76 instances".
- DEPENDENCIES**: A list of dependencies:
 - charts-api-backend: declared + runtime •
 - chartview: declared + runtime •
 - collection (collection2): declared + runtime •
 - facebook: declared + runtime •
 - identity (identity2): declared + runtime •
 - mergedprofile (socialgraph3): declared + runtime •
 - metadata (metadata-proxy): declared + runtime •
 - playlist -»
 - playlist-proxy: declared + runtime •
 - playlist4: declared + runtime •
 - playlist-mosaic: declared + runtime •
 - popcount (popcount2): declared + runtime •
 - search: declared + runtime •

Internal tools and compliance processes

3rd party tools and services

Best practices and technical documentation

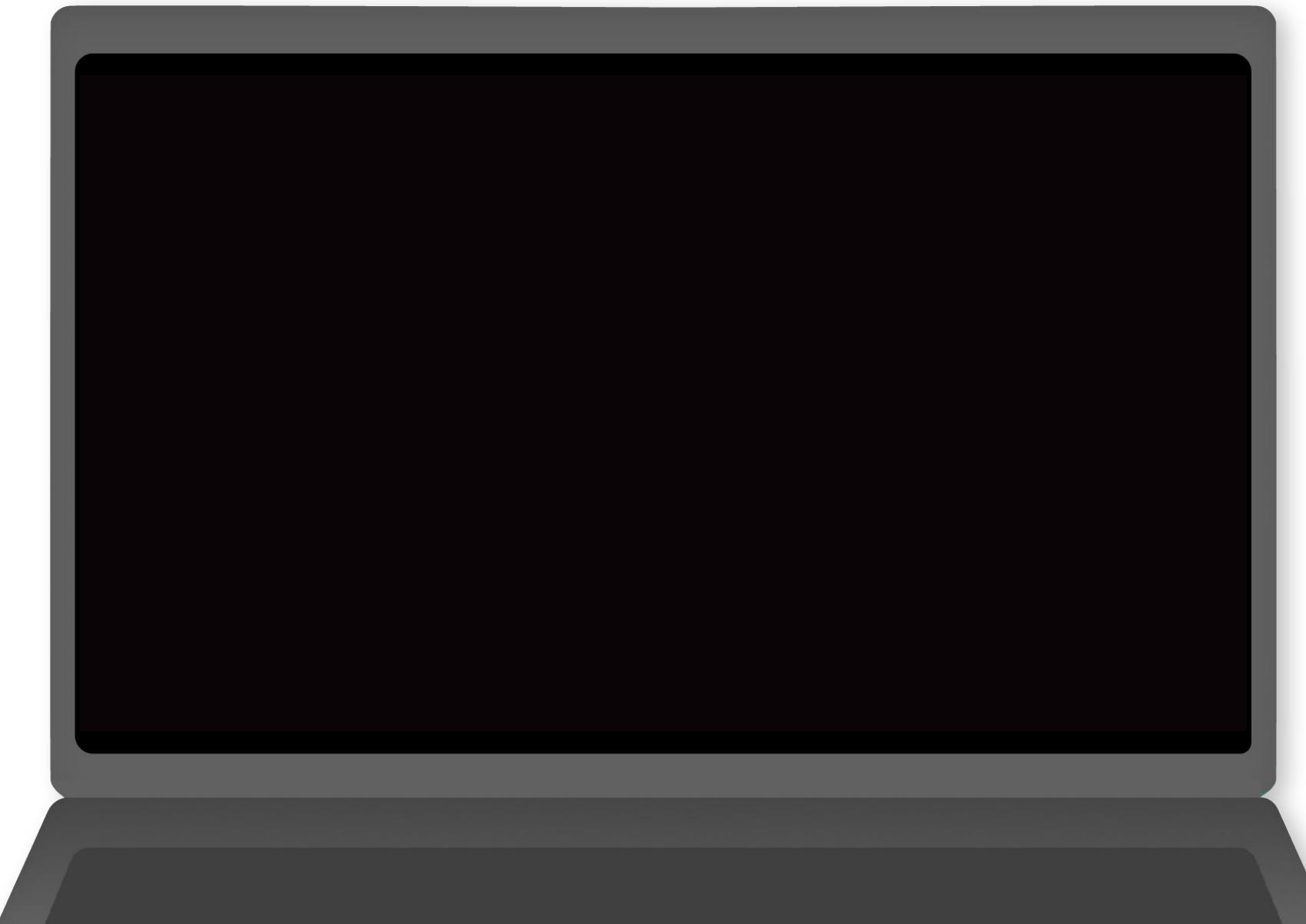


Backstage

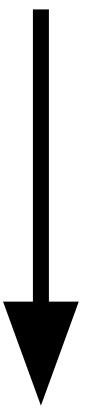
Engineers, etc.

The Golden Age

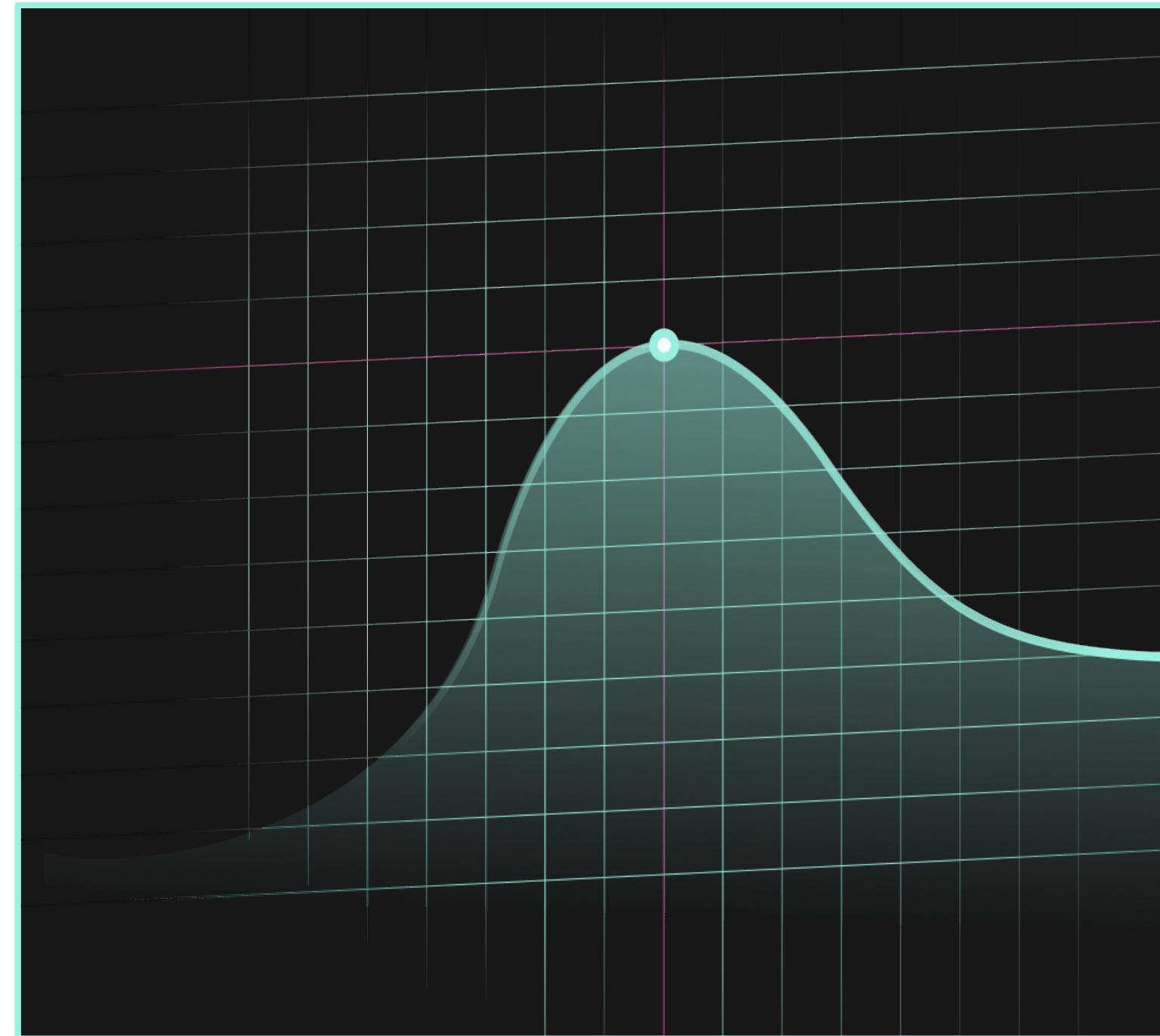
Present Day



system-z



 Backstage



DATA ENDPOINT

backstage-e2e-test-data.TestData

backstage-e2e-test-data ▾

Owner tools Lifecycle EXPERIMENTAL

OVERVIEW DOWNSTREAMS UPSTREAMS LINEAGE COUNTERS VALIDATION PROFILING ANOMALY DETECTION AUTOTUNE

Notice: No Monitoring & Alerting Set-up

Overview

backstage-e2e-test-data scio pipeline

Details

OPEN IN GCS ACCESS GRANTED VIEW TECHDOCS VIEW SOURCE CONTACT OWNER GET NOTIFIED

DESCRIPTION
backstage-e2e-test-data scio pipeline

Owned by tools

STORAGE GCS LIFECYCLE EXPERIMENTAL ACCESS TYPE BROAD

Access Granted

You already have non-expiring access to this endpoint by belonging to the backend LDAP group. Only click request access to this endpoint if you would like access for additional emails such as a service account.

REQUEST FOR ANOTHER ACCOUNT

Delivery

DAILY DELIVERY 2021-08-22 is OK SLO CONFIGURED not set

Status Legend TRY SLO CALCULATOR

Last 28 days compared to the previous 28 days.

DELIVERY TIME - P90	SLO COMPLIANCE	SLO BREACHES	BREACH DURATION
13 min +1.4%	100.0% 0.0%	0	0 min

Data Lineage

UPSTREAMS 0 endpoints > 0 owners to root endpoint

DOWNSTREAMS 0 endpoints > 0 total downstream endpoints

NOTIFY DOWNSTREAM CONSUMERS

Image above is for illustrative purposes only.



COMPONENT – SERVICE

greeter ☆

Owner: mathy-mammals | Lifecycle: production | ...

[OVERVIEW](#) [CI/CD](#) [API](#) [DEPENDENCIES](#) [DOCS](#)

About

[VIEW SOURCE](#) [VIEW TECHDOCS](#) [VIEW API](#)

DESCRIPTION: Serves greetings at planetary scale.

OWNER: mathy-mammals SYSTEM: solar TYPE: service

LIFECYCLE: production TAGS: go, demo, hello

Read me

Greeter

Provides greetings to calm every visitor.

```
go run main.go
```

API Endpoints

- / provides greetings
- /_health provides health check.

[Read me →](#)

Links

[Website](#) [Documentation](#)
[Grafana Dashboard](#) [Logging](#)

Languages



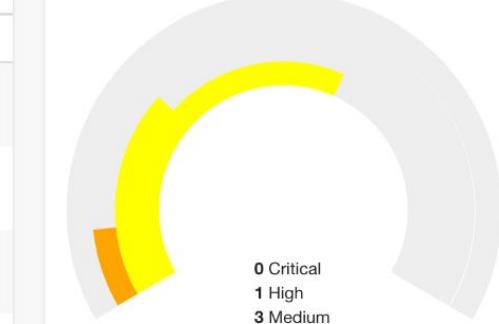
TypeScript - 92.18%	HTML - 4.53%
Dockerfile - 2.02%	Go - 0.70%
JavaScript - 0.57%	

Recent Workflow Runs

All Branches

COMMIT MESSAGE	BRANCH	STATUS
Adjusted the gizmo	main	● Completed
Modified the widget	main	● Completed
Adding the feature without a bug	main	● Completed
Adding a feature with a bug	main	● Error
Getting the hang of this	main	● Completed
First workflow run!	main	● Completed

Vulnerabilities



0 Critical
1 High
3 Medium
6 Low

Across 1 projects →

PagerDuty

[SERVICE DIRECTORY](#) [CREATE INCIDENT](#)

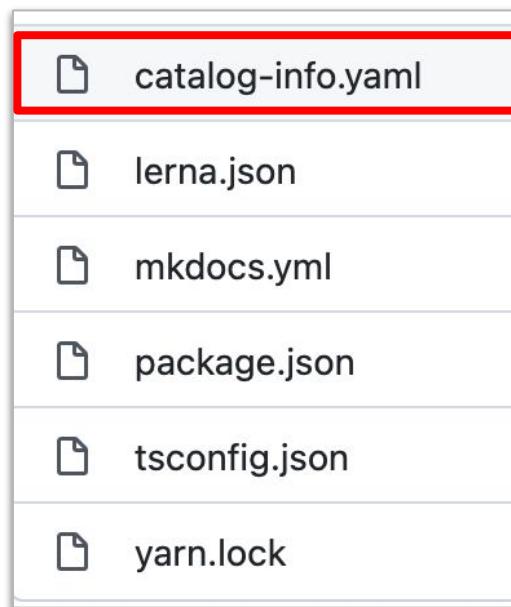
Nice! No incidents found!



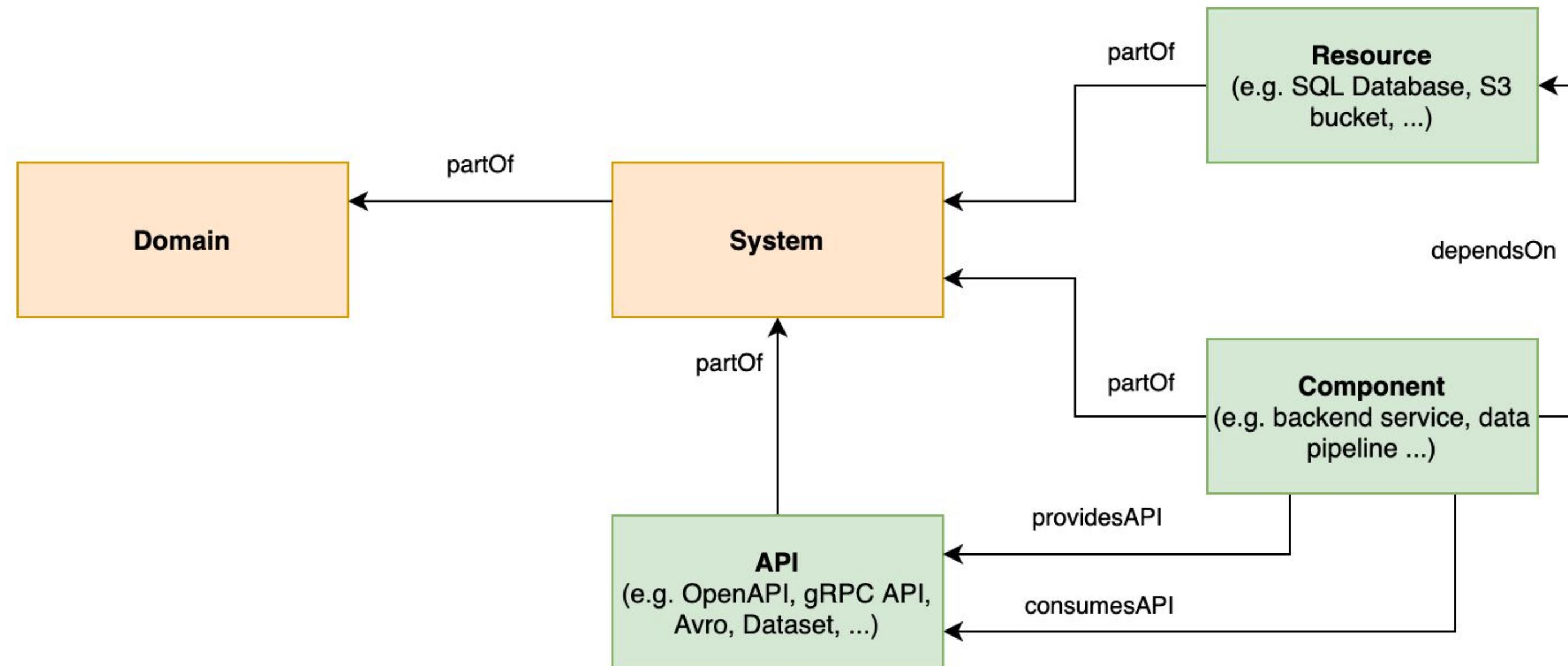
ON CALL

Rugvip	✉ ↑
jhaals	✉ ↑

Images above are for illustrative purposes only.



```
apiVersion: backstage.io/beta1
kind: Component
metadata:
  name: greeter
  links:
    - icon: graph
      title: Metrics
      href: https://grafana.corp/...
  annotations:
    pagerduty.com/key: abc123
spec:
  type: service
```



```
apiVersion: backstage.io/beta1
kind: Component
metadata:
  name: greeter
  links:
    - icon: graph
      title: Metrics
      href: https://grafana.corp/...
  annotations:
    pagerduty.com/key: abc123
spec:
  type: service
```

The screenshot shows the Backstage interface for a component named 'greeter'. The top navigation bar includes links for 'OVERVIEW', 'CI/CD', 'API', 'DEPENDENCIES', and 'DOCS'. The 'OVERVIEW' tab is selected. On the left sidebar, there are icons for 'VIEW SOURCE', 'TECHDOCS' (which is highlighted in blue), and 'VIEW API'. The main content area displays the 'About' section, which includes a 'Read me' file containing the description: 'Provides greetings to calm every visitor.' and the command 'go run main.go'. It also lists 'API Endpoints' such as '/ provides greetings' and '/_health provides health check.'. Below this is a 'Recent Workflow Runs' table showing seven completed runs across different branches. To the right of the 'About' section are three cards: 'Vulnerabilities' (0 Critical, 1 High, 3 Medium, 6 Low), 'PagerDuty' (Nice! No incidents found!), and 'Languages' (TypeScript 92.18%, HTML 4.53%, Dockerfile 2.02%, Go 0.70%, JavaScript 0.57%).

```
apiVersion: backstage.io/beta1
kind: Component
metadata:
  name: greeter
  links:
    - icon: graph
      title: Metrics
      href: https://grafana.corp/...
  annotations:
    pagerduty.com/key: abc123
  spec:
    type: service
```

Image above is for illustrative purposes only.

The screenshot shows the Backstage UI with a dark theme. On the left is a vertical sidebar with icons for navigation, search, home, settings, and more. The main area has a teal header bar with the title "Register an existing component". Below the header, there's a sub-header "Start tracking your component in Backstage Example App".
Left Panel (Wizard Steps):

- Step 1:** Select URL. A text input field contains the URL `https://github.com/backstage/backstage/blob/master/catalog-info.yaml`. Below it is a placeholder: "Enter the full path to your entity file to start tracking your component". A blue "ANALYZE" button is present.
- Step 2:** Import Actions (Optional). This step is marked as optional.
- Step 3:** Review.
- Step 4:** Finish.

Right Panel (Information):

Register an existing component

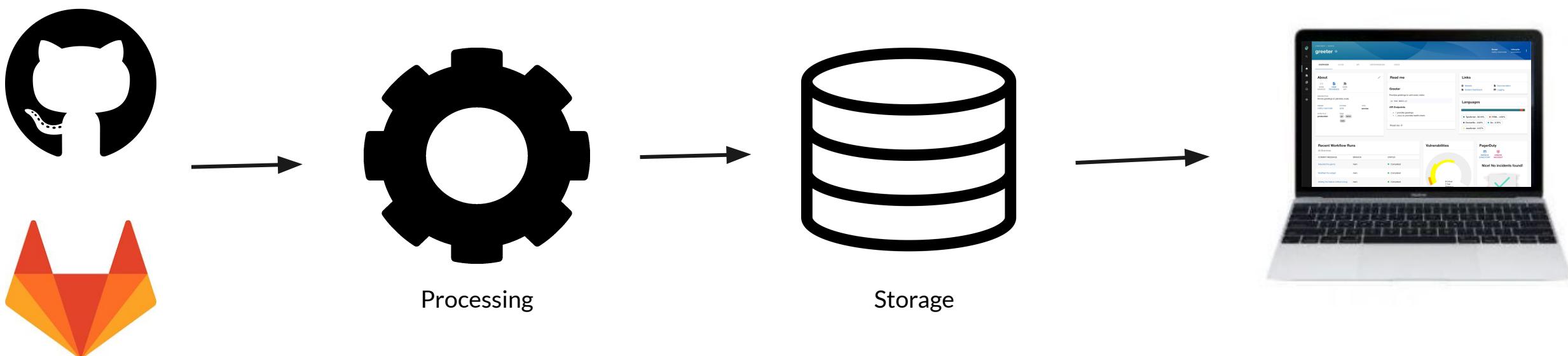
Enter the URL to your source code repository to add it to Backstage Example App.

Link to an existing entity file
Example: `https://github.com/backstage/backstage/blob/master/catalog-info.yaml`
The wizard analyzes the file, previews the entities, and adds them to the Backstage Example App catalog.

Link to a repository GitHub only
Example: `https://github.com/backstage/backstage`
The wizard discovers all `catalog-info.yaml` files in the repository, previews the entities, and adds them to the Backstage Example App catalog.
If no entities are found, the wizard will prepare a Pull Request that adds an example `catalog-info.yaml` and prepares the Backstage Example App catalog to load all entities as soon as the Pull Request is merged.

[Learn more about the Software Catalog →](#)

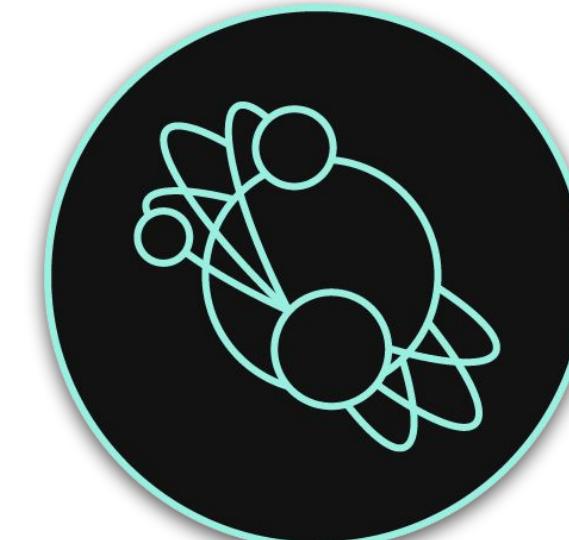
Processing Overview



Populating the Catalog



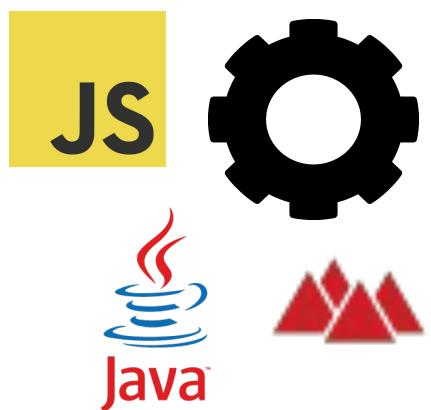
Manual Registration



Software Catalog



Automatic Discovery

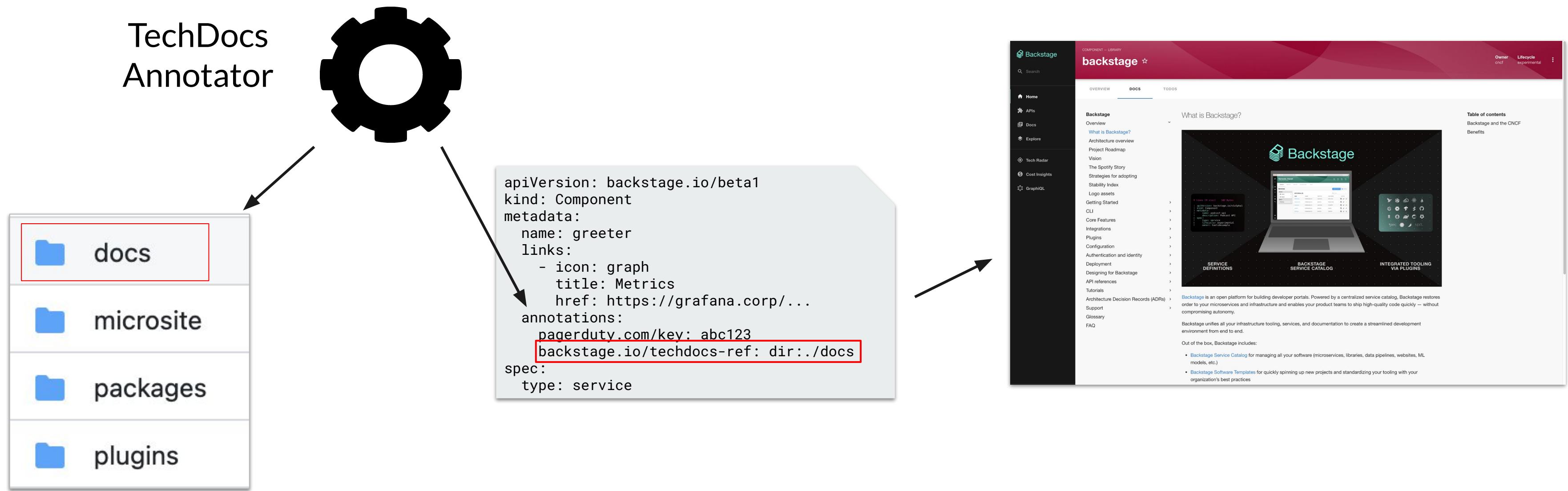


Custom Processors



Configuration

Custom Processor Example



Demo!



Adoption Strategies

Create Incentives



Start Small



Prioritize Efforts





Find out more!

Docs, demos, and updates:

backstage.io

Github:

github.com/backstage/backstage

Community Sessions:

github.com/backstage/community

Discord:

discord.gg/MUpMjP2

Adopting Backstage:

backstage.spotify.com