

Tech Talks

Platform Engineering With Backstage

Getting Started



Disclaimer

The content contained herein is for informational purposes only, may not be referenced or added to any contract, and should not be relied upon to make purchasing decisions. It is not a commitment, promise, or legal obligation to provide any features, functionality, capabilities, code, etc. or to provide anything within any schedule, date, time, etc. All Mirantis product and service decisions remain at Mirantis sole and exclusive discretion.

\$ whoami

Martin Nirlt

Solution Architect



I am an IT engineer with strong backgrounds in software, DevOps/platform and electronic engineering working for [Mirantis](#) as a pre-sales solution architect. Next to my job, my main side-hustles are all around Kubernetes , IaC and automating things. From time to time, I even build little apps in Go or other languages.



[martinnirlt](#)



[martinnirlt](#)



[martinnirlt](#)



Slides, Code and more!

Agenda

Platform Engineering

Backstage

Architecture & More

Getting started

Mirantis Lens ApplQ



Platform Engineering

Let's feel the buzz! 🚖

Software Engineering Challenges

Increased complexity due to modern software architectures

High cognitive load and developer toil

Jack of all trades - Code, AWS, Terraform, Helm, Argo and many more

Lack of talent

“Throw over the fence” workflow ineffective

No standards or best practices across teams

Software Engineering Utopia

Self-sufficient teams

Go fast and independent

Great overall developer experience

No friction and wait times

Clear boundaries and responsibilities

Right level of flexibility

Standardization and best practices

The Way of Platform Engineering

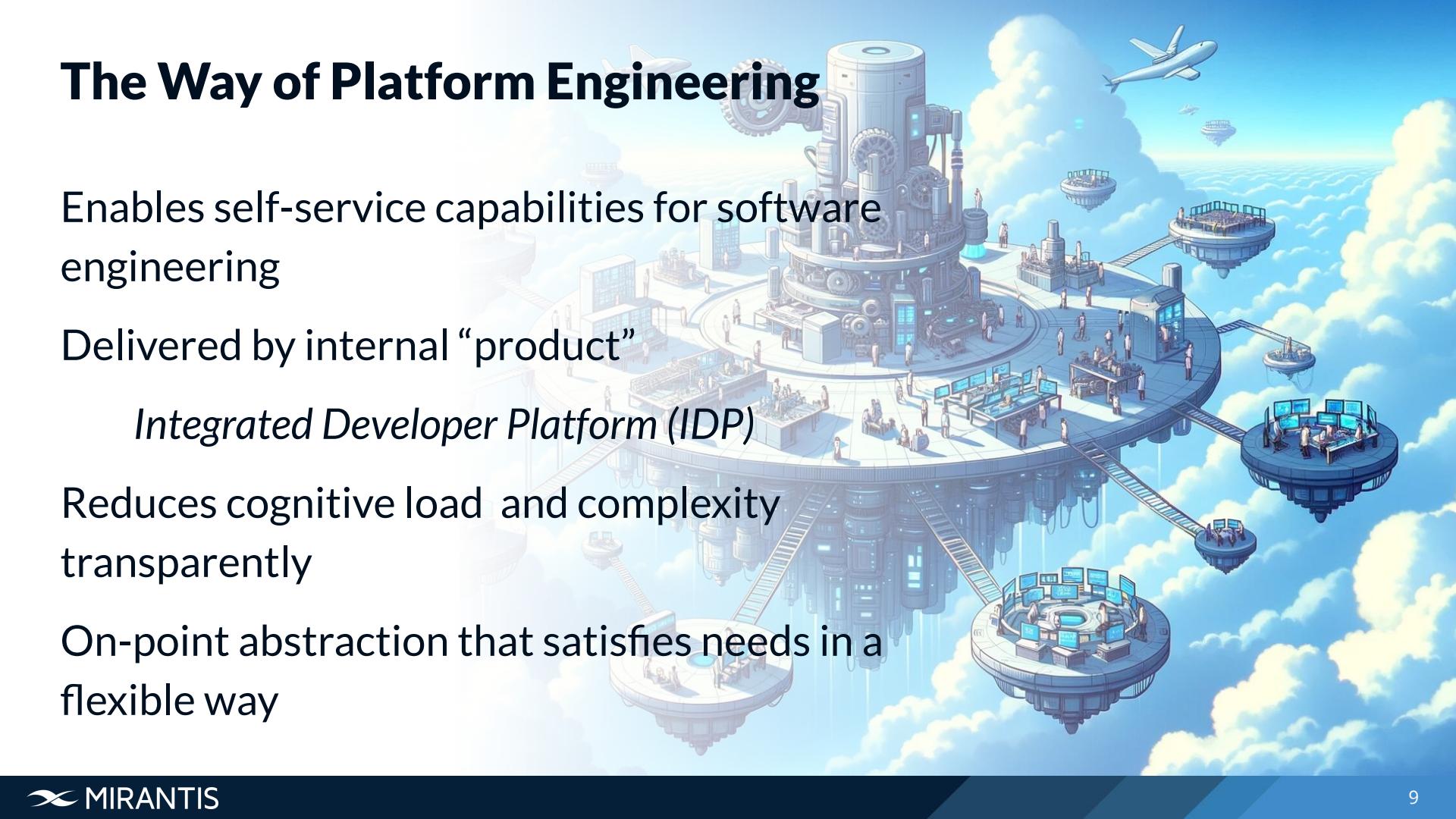
Enables self-service capabilities for software engineering

Delivered by internal “product”

Integrated Developer Platform (IDP)

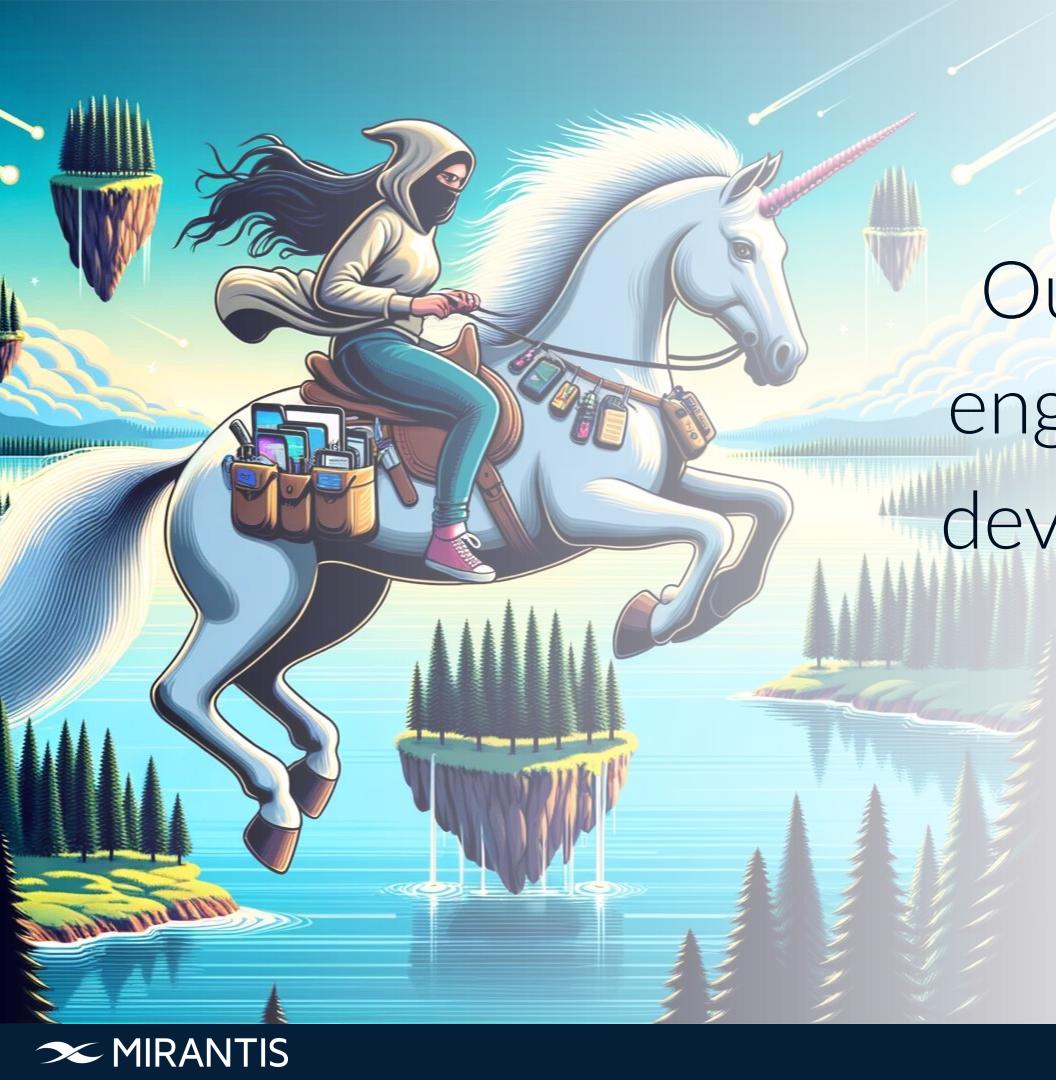
Reduces cognitive load and complexity transparently

On-point abstraction that satisfies needs in a flexible way



Backstage

Backstage on Frontstage 😜

A vibrant, fantastical illustration of a woman with long dark hair, wearing a light-colored hoodie and jeans, riding a white unicorn. The unicorn is rearing slightly, carrying a saddle bag filled with various electronic devices like phones and tablets. They are flying over a series of small, green, tree-covered islands that float on a calm blue sea. The background features a bright sky with a few shooting stars and distant, hazy landmasses.

Our goal is to provide
engineers with the best
developer experience in
the world.

Backstage Vision



What's Backstage?

A platform for building developer portals

Incubating CNCF project

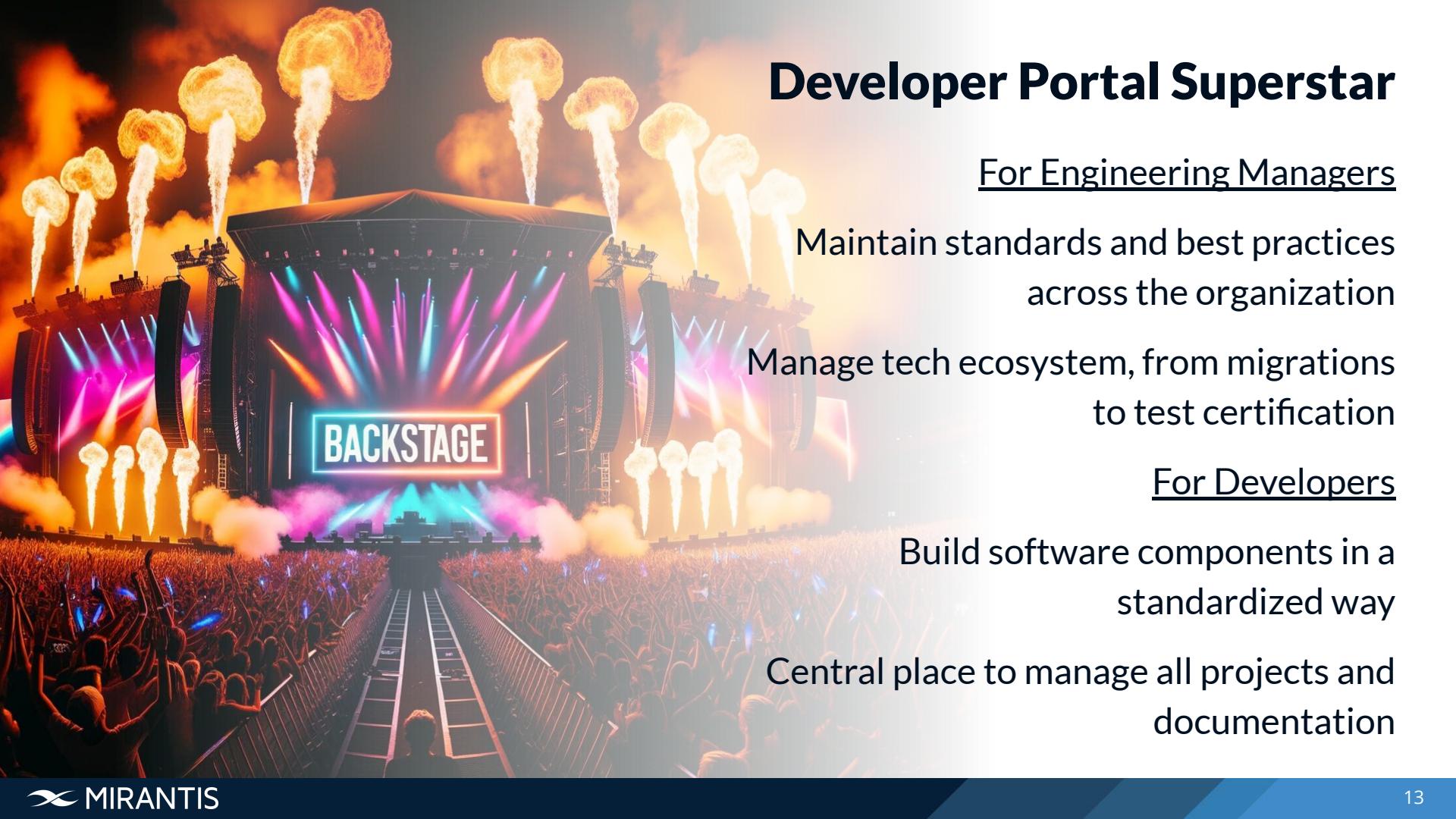
Transparency through centralized software catalog

Standardization via software templates

Easy access to documentation

Great ecosystem (plugins)

Highly extensible and customizable



Developer Portal Superstar

For Engineering Managers

Maintain standards and best practices
across the organization

Manage tech ecosystem, from migrations
to test certification

For Developers

Build software components in a
standardized way

Central place to manage all projects and
documentation



Developer Portal Superstar

For Platform Engineers

Extensibility and scalability through easy integration of tools and services (via plugins), as well as extending the functionality of existing ones

For Everyone

Single, consistent experience that ties all tooling, resources, standards, owners, contributors, and administrators together in one place

Backstage Round Trip

Let's check out the [demo portal!](#)



At this point, we switch over to the Backstage Demo App.

If you want to see the demo,
you can find the video on [Youtube](#)

Architecture & More

How does Backstage work? 

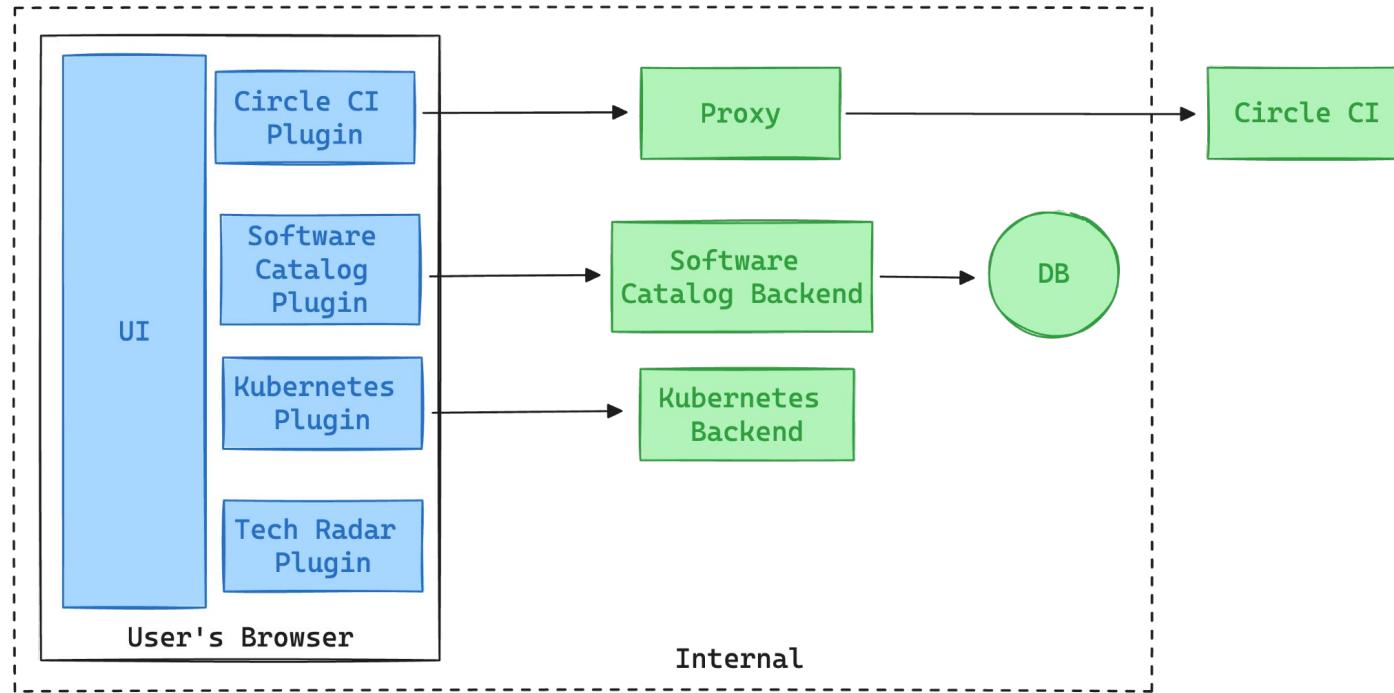
Backstage Architecture - User Interface

The screenshot shows the Backstage UI Core Component interface. On the left is a sidebar with navigation links: Home, APIs, Docs, Plugins, Backstage, and Search. The 'Backstage' link is highlighted. A large orange box contains the title 'UI Core Component'. Below it is the 'Backstage Catalog' header. The main area displays a table titled 'All components (15)'. The columns are: NAME, SYSTEM, OWNER, TYPE, LIFECYCLE, DESCRIPTION, TAGS, and ACTIONS. The table lists various components like 'artist-lookup', 'backstage', 'backstage demo', etc., with details such as 'experimental' or 'production' status and 'java' or 'go' tags. A blue watermark 'Software Catalog / Plugins' is overlaid across the center of the table. The sidebar also includes sections for 'OWNER' and 'LIFECYCLE' filters.

NAME	SYSTEM	OWNER	TYPE	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
artist-lookup	artist-engagement-portal	team-a	service	experimental	Artist Lookups	java, data	
backstage		cncf	library	experimental	Backstage is an...		
backstage demo		backstage/mainainers	website	experimental	An example...		
petstore		team-c	service	experimental	(Title...)		
playback-order	audio-playback	userguest*	service	production	Playback Order	java, playback	
playback-sdk	audio-playback	team-c	library	experimental	Audio and video...		
podcast-api	podcast	team-f	service	experimental	Podcast API	java	
queue-proxy	podcast	team-d	website	production	Queue Proxy	go, website	
searcher		userguest	service	production	Searcher	go	
shuffle-api	audio-playback	userguest	service	production	Shuffle API	go	
wayback-archive		team-a	service	production	Archive of the...		
wayback-archive-ingestion		team-d	service	production	Ingestion...		
wayback-archive-storage		team-a	service	production	Storage...		
wayback-search		team-a	service	production	Search of the...		
www-artist	artist-engagement-portal	team-a	website	production	Artist main website		

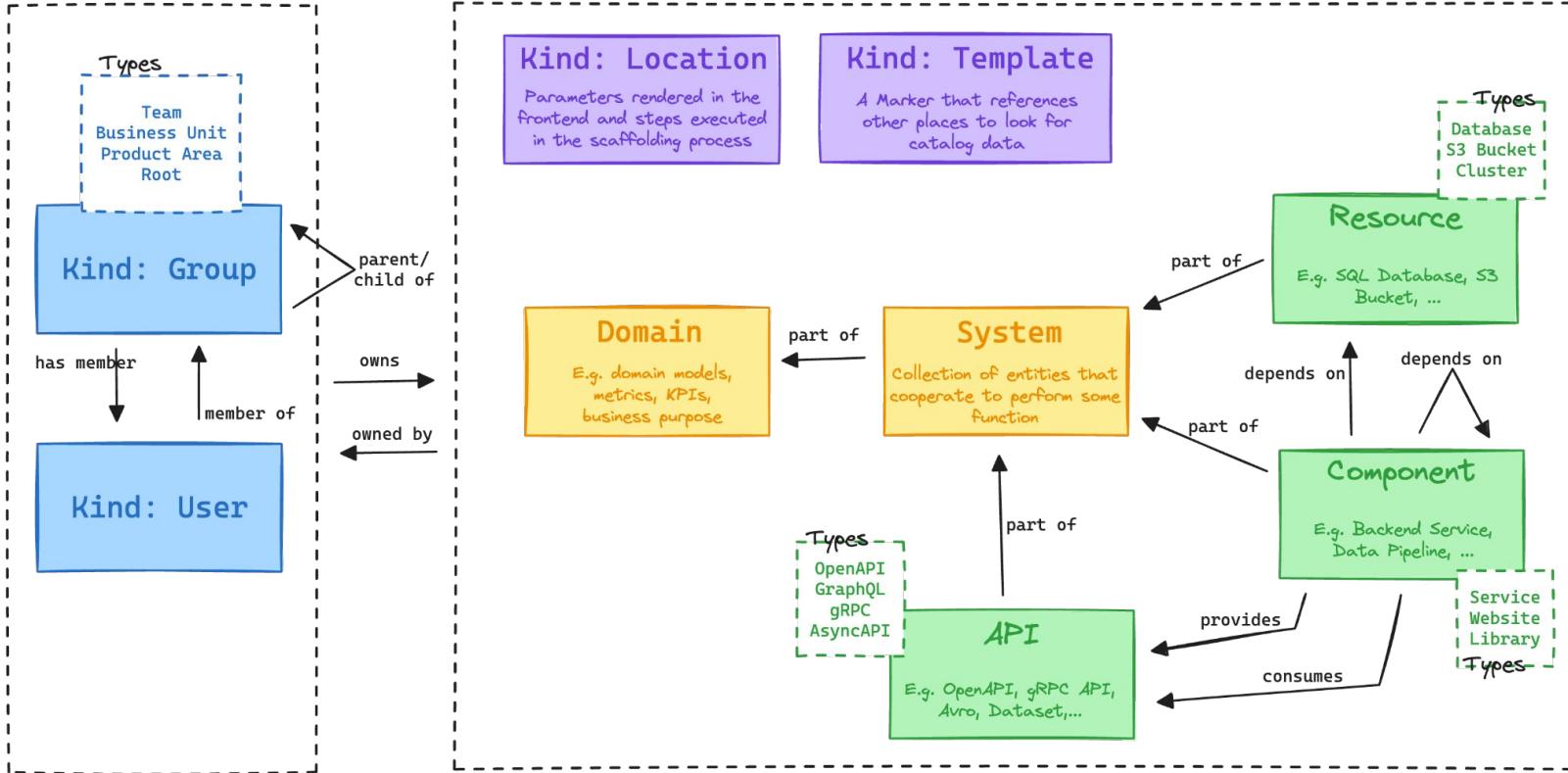
Source: backstage.io

Backstage Architecture



Source: backstage.io

Software Catalog - System Model and Entities



Source: backstage.io

Plugins by Example - Kubernetes (Core Plugin)



The screenshot shows the Backstage.io interface for the **dice-roller** service. The top navigation bar includes links for Overview, CI/CD, Sentry, API, Docs, **Kubernetes** (which is selected), Pull Requests, and Code Insights. The service name **dice-roller** is displayed with a yellow star icon. On the right, the Owner is listed as guest and the Lifecycle as production. A vertical sidebar on the left contains icons for Home, Services, Catalog, +, Observability, Checks, Metrics, and Notifications.

Error Reporting

CLUSTER	KIND	NAME	MESSAGES
eu-1	Pod	dice-roller-canary-6bc4fcbe44-hphr9	back-off 2m40s restarting failed container=other-side-car pod=dice-roller-canary-6bc4fcbe44-hphr9 default(3a301761-49c8-424e-a10e-5a2fb050dfb1) back-off 2m40s restarting failed container=side-car pod=dice-roller-canary-6bc4fcbe44-hphr9 default(3a301761-49c8-424e-a10e-5a2fb050dfb1)
eu-1	Pod	dice-roller-canary-6bc4fcbe44-sjpt	back-off 2m40s restarting failed container=other-side-car pod=dice-roller-canary-6bc4fcbe44-sjpt default(62291bd4-3f63-42a9-bbc9-26c1fad5976) back-off 2m40s restarting failed container=side-car pod=dice-roller-canary-6bc4fcbe44-sjpt default(62291bd4-3f63-42a9-bbc9-26c1fad5976)
eu-1	Deployment	dice-roller-canary	Deployment does not have minimum availability.
eu-1	Pod	dice-roller-canary-6bc4fcbe44-hphr9 <small>+ 1 other</small>	containers with unready status: [side-car other-side-car] containers with unready status: [side-car other-side-car]
eu-1	Pod	dice-roller-canary-6bc4fcbe44-hphr9 <small>+ 1 other</small>	container=other-side-car restarted 5 times container=side-car restarted 5 times

5 rows | < < 1-5 of 6 > >|

Your Clusters

Source: backstage.io

Plugins by Example - Kubernetes (Core Plugin)



Your Clusters

eu-1 Cluster

dice-roller Deployment

min replicas 10 / max replicas 15
current CPU usage: 98%
target CPU usage: 50%

17 pods
2 pods with errors

15 pods
No pods with errors

NAME	PHASE	CONTAINERS READY	TOTAL RESTARTS	STATUS
dice-roller-7b644f9b77-6dsi6	Running	1/1	0	OK
dice-roller-7b644f9b77-6znj4	Running	1/1	0	OK
dice-roller-7b644f9b77-798qc	Running	1/1	0	OK
dice-roller-7b644f9b77-79tc5	Running	1/1	0	OK
dice-roller-7b644f9b77-8g5t8	Running	1/1	0	OK

5 rows | < | < 1-5 of 15 | > | >|

A screenshot of the Backstage.io interface for a Kubernetes cluster named "eu-1". On the left is a sidebar with various icons. The main area shows the cluster summary with 17 pods (2 with errors) and a deployment named "dice-roller" with 15 pods (0 errors). Below this is a table listing five pod replicas, each in a "Running" state with 1/1 containers ready and 0 total restarts, all marked as "OK". Navigation controls at the bottom allow for viewing more rows and navigating through the results.

Source: backstage.io

Getting Started

What do I need to do? 😊

Getting Started with Backstage

As an Organisation

- Do i need an IDP?
- Dedicated platform team
 - Internal product
- Adoption strategy
 - Lunch & learns, hackathons
- Make it a business case
 - KPIs and metrics

As an Individual

- Platform engineering
- Understand Backstage
 - System model
 - Terminology
- React, NodeJS, YAML, Docker, Monorepo, ...
 - Full list of concepts

The Mirantis Way

Rocket-start with Lens AppIQ 



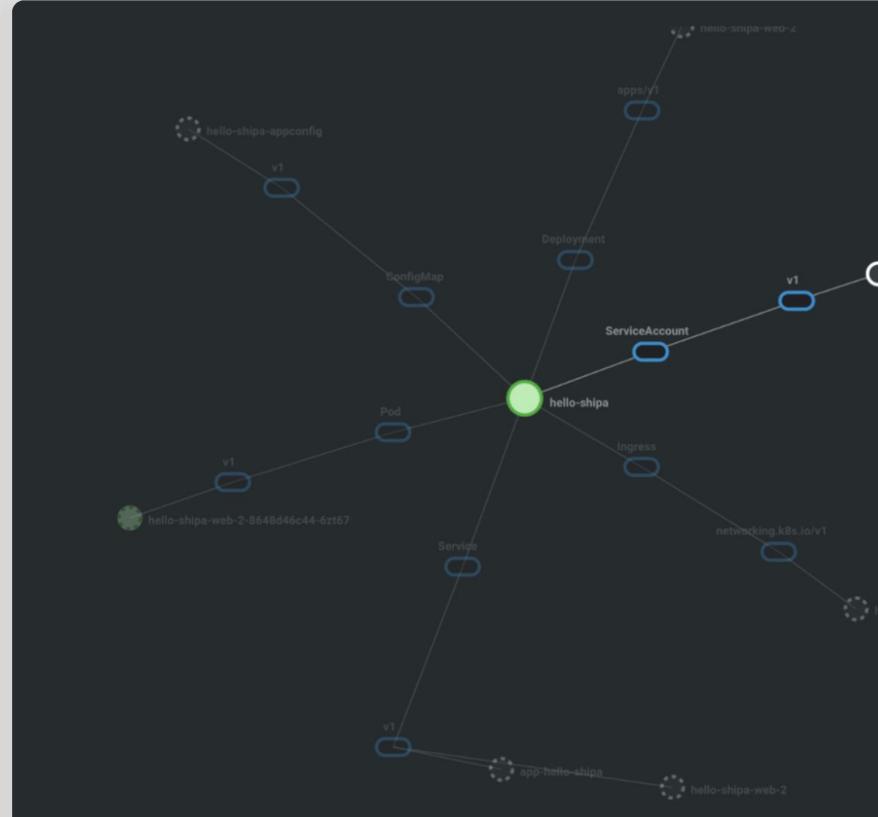
Mirantis Lens AppIQ

A unified platform ready to be consumed by your developers

App-centric intelligence for software engineering

Manage and troubleshoot apps in a self-service way

Ensure and monitor security, governance and compliance





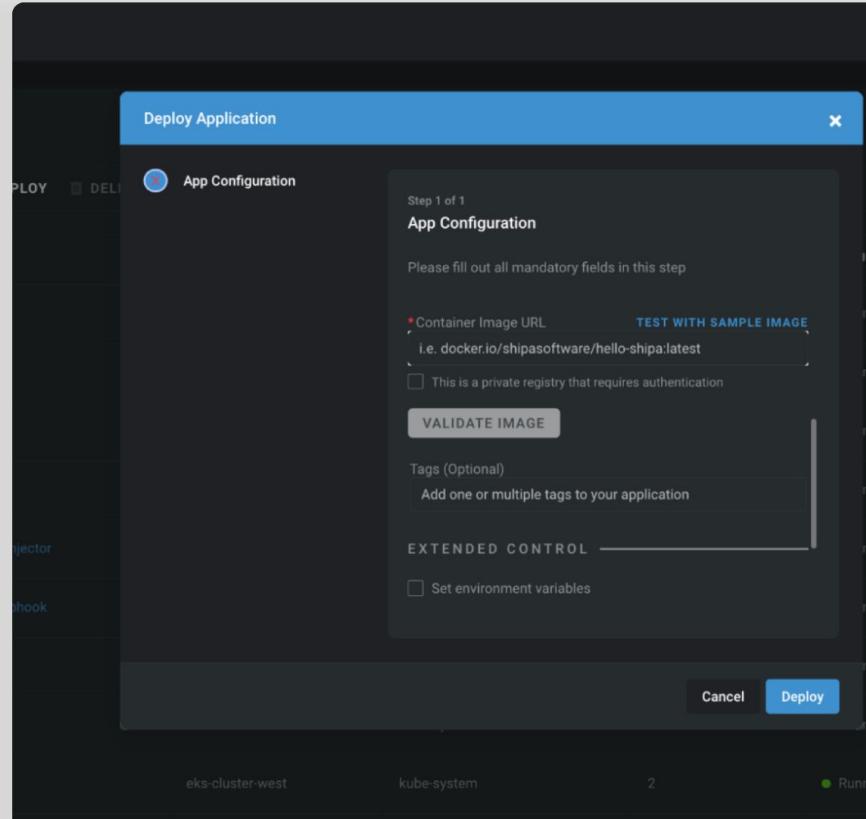
Mirantis Lens AppIQ

Productivity boost for
accelerated releases and reduced
complexity

Increased competitiveness
through faster innovation

Watch the [demo on Youtube](#)

Start using [Lens AppIQ](#) today 



The screenshot shows a dark-themed user interface for deploying an application. At the top, a blue header bar reads "Deploy Application". Below it, a sub-header "App Configuration" is shown with a circular icon containing a gear. The main area is titled "Step 1 of 1 App Configuration" and contains the following fields:

- "Container Image URL": A text input field containing "i.e. docker.io(shipasoftware/hello-shippa:latest)". To its right is a "TEST WITH SAMPLE IMAGE" button.
- A checkbox labeled "This is a private registry that requires authentication".
- A "VALIDATE IMAGE" button.
- "Tags (Optional)": A text input field with the placeholder "Add one or multiple tags to your application".
- "EXTENDED CONTROL": A section with a checkbox for "Set environment variables".

At the bottom of the dialog are "Cancel" and "Deploy" buttons. The background of the slide shows a blurred view of the same application interface with some UI elements like "eks-cluster-west" and "kube-system".