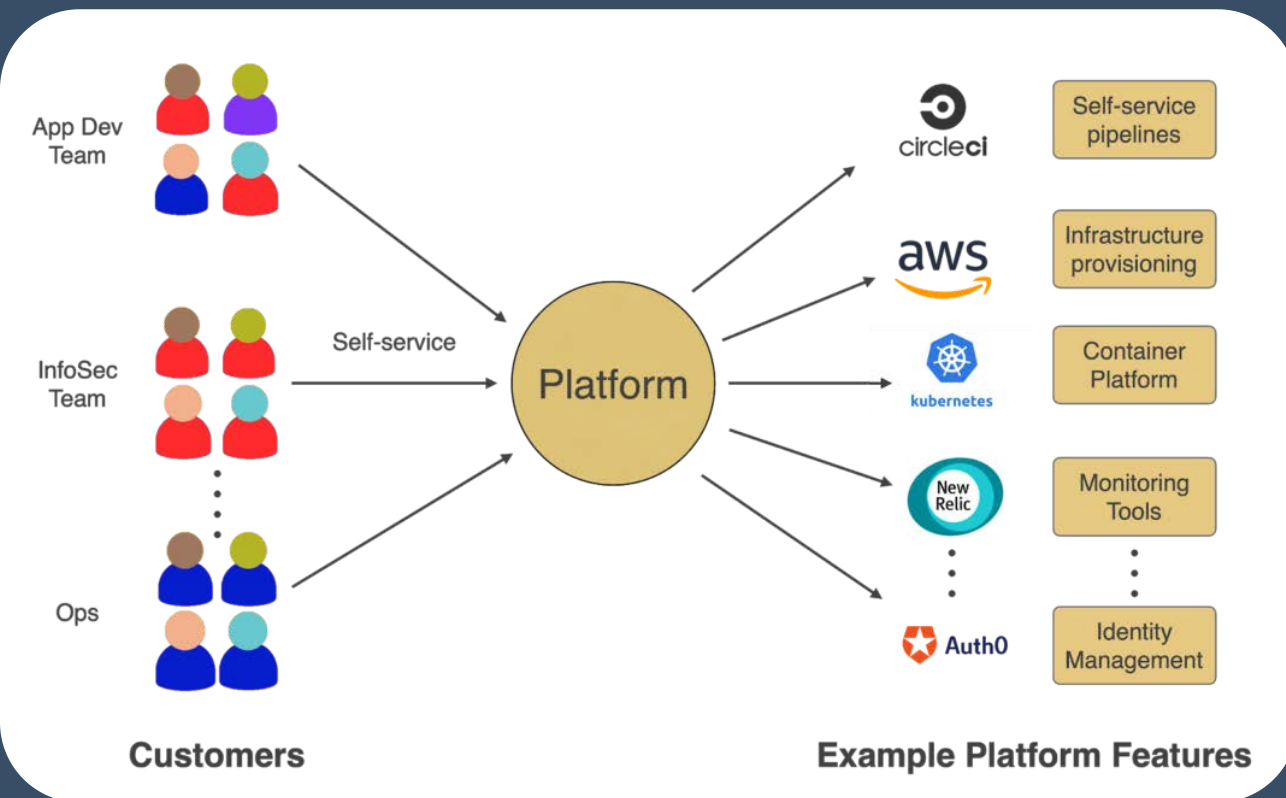


Platform Engineering: Why and How to start



Serg Hospodarets

My path

Title	Managed team size
Tech Lead, Architect	~10
Engineering Manager	~20
Director of Engineering	50+
SDirE, Global Head of Eng.	100+

Platforms Creator



cTrader Platform

The platform allows our customers to create, develop and deliver their mobile and desktop applications for traders.

Demo applications:



 GUIDEWIRE | CLOUD

JUTRO Digital Platform:
Infrastructure, Design
System, and UI Framework

DevOps and Web contributor



Serg Hospodarets
shospodarets

Technology leader, speaker, Web and Cloud infrastructure enthusiast.

116 followers · 74 following

shospodarets@gmail.com
https://hospodarets.com/
@shospodarets

jhipster/generator-jhipster Public

JHipster is a development platform to quickly generate, develop, & deploy modern web applications & microservice architectures.

TypeScript 20.3k 3.9k

css-vars Public

Sass mixin to use CSS Custom Properties with Sass

JavaScript 178 8



Types // as Comments in JS ECMAScript Proposal

Goal

```
const message: string = "Hello, types"
console.log(message)
```

Runs without tools, runs in a browser ✓

Why?

What problem do we try to solve
and Why Platform Engineering?

Problem with the current state of DevOps



A screenshot of a chat interface with a dark grey background. At the top, a title bar reads "DevOps is a Software Development Methodology" with a hamburger menu icon on the left and a plus sign on the right. The chat history shows four messages:

- A user profile picture (a man in a blue jacket) asks: "Hey ChatGPT. What is DevOps?" with a pencil icon on the right.
- The ChatGPT logo (a green square with a white knot) responds: "DevOps is a software development methodology and culture that emphasizes collaboration and communication between development and operations teams in order to improve the speed and quality of software releases." with thumbs up and thumbs down icons on the right.
- The same user asks: "Great, how do I start?" with a pencil icon on the right.
- The ChatGPT logo responds: "🤖 go figure..."



Current state of DevOps and problems

1. COMPLEXITY GREW EXPONENTIALLY 📌

Common Problem: Cognitive overload, support complexity

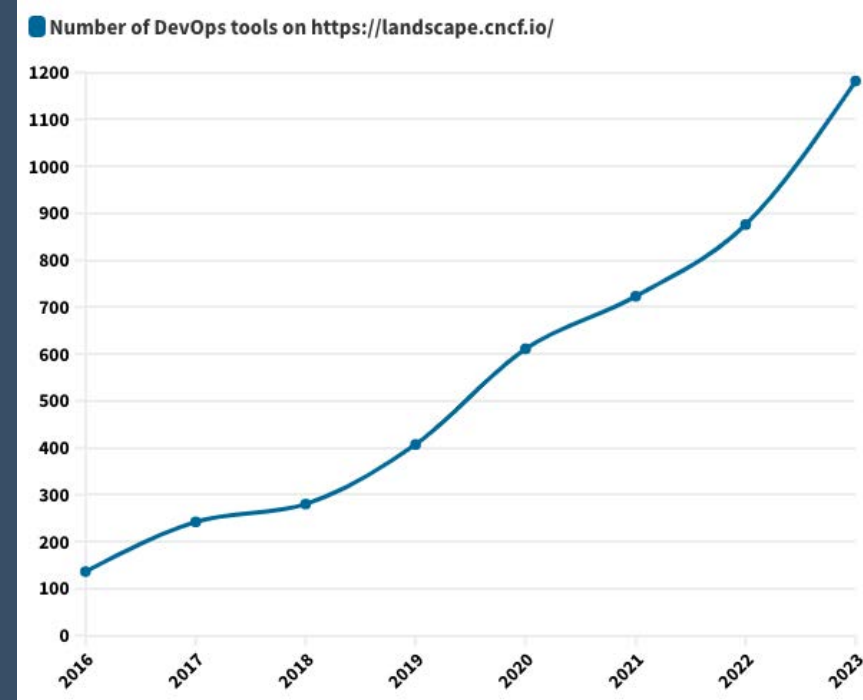
2. Independent teams topology adopted

Architecture/DevOps problem: No way to align the technology and architecture

Ops, Security problem: No way to introduce security scans, patches, centralised monitoring etc.







Business problem: \$ cost of solutions entropy. It also doesn't allow to quickly pivot at scale

Eng. leadership platform: How to follow the "do more with less" mandate from business



Independent teams example	Team A	Team B	Team C
Tech Stack	Java	Python	React
CI/CD	Gitlab	Gitlab/Github	Jenkins/Github
Hosting	AWS Serverless	Bare metal	Vercel
Security	Checkmarx	Manual	SemGrep
Monitoring	DataDog	Prometheus	Sentry

Usual problems and Use cases to enable

- Common: Provide more common tech stack to simplify the hiring and onboarding
-  Dev: What is recommended stack and tools do I use for Front-End/Back-End/Data, CI/CD, Cloud?
-  Eng. leader: Minimize time to onboarding
-  Security: Security fix like Log4j
-  Ops: Automate the company-wide way of security check
-  Business: We acquired a company, how we integrate them
-  Business: Can you operate twice faster and more cost effectively?

How to apply DevOps in practice at enterprise level?



To apply DevOps at an enterprise level, organizations can start from Platform engineering.

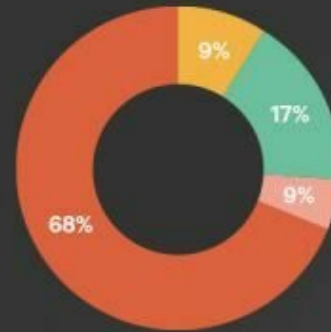
Platform engineering is the practice of building, maintaining, and scaling a software platform that can be used by multiple teams to build and run their applications.

So why Platform Engineering?

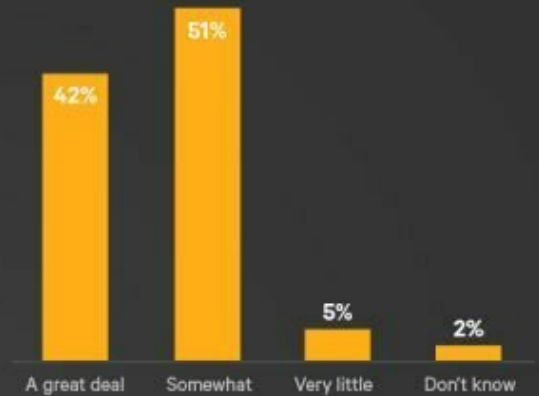
Contributes to "Do more with less" enablement

Did the platform team have a direct impact on development velocity?

Increased Decreased Stayed the same Don't know



By approximately how much did development speed increase after the inception of the platform team?



Discover more in the

2023 State of DevOps Report [▶](#)
Platform Engineering Edition





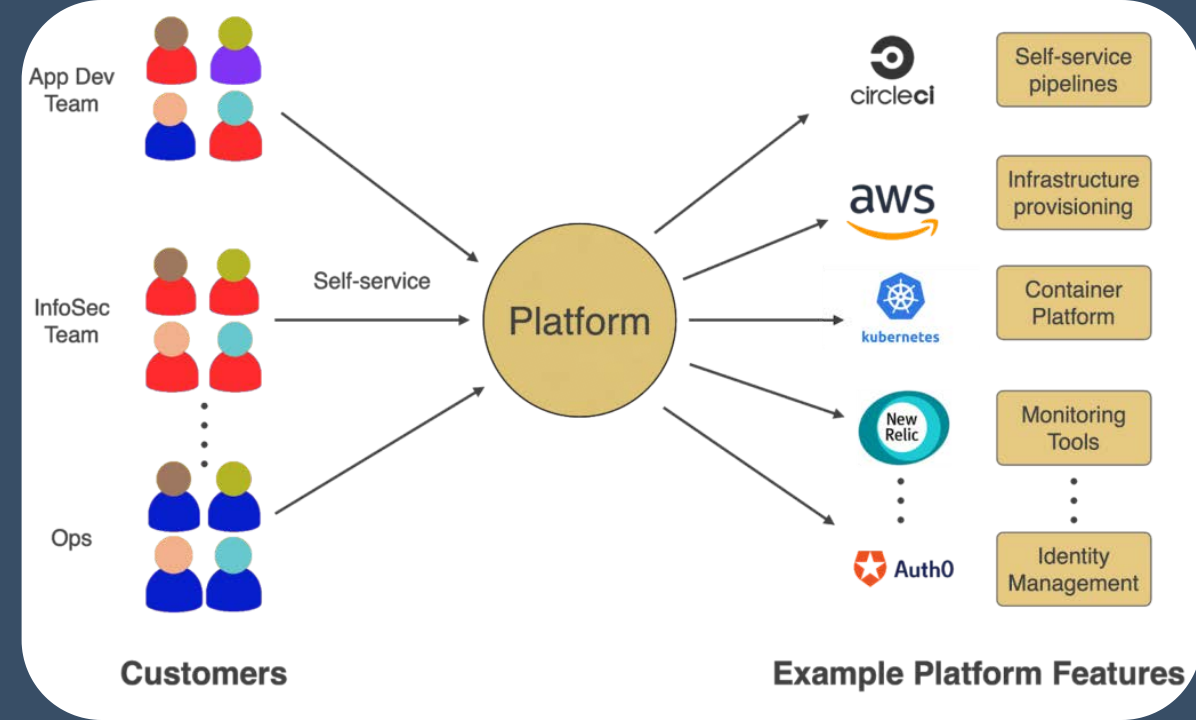
WHAT?

What are the objectives and goals?

Platform as a Product

Main Customers- internal developers first

Main metrics- customers NPS and ADOPTION



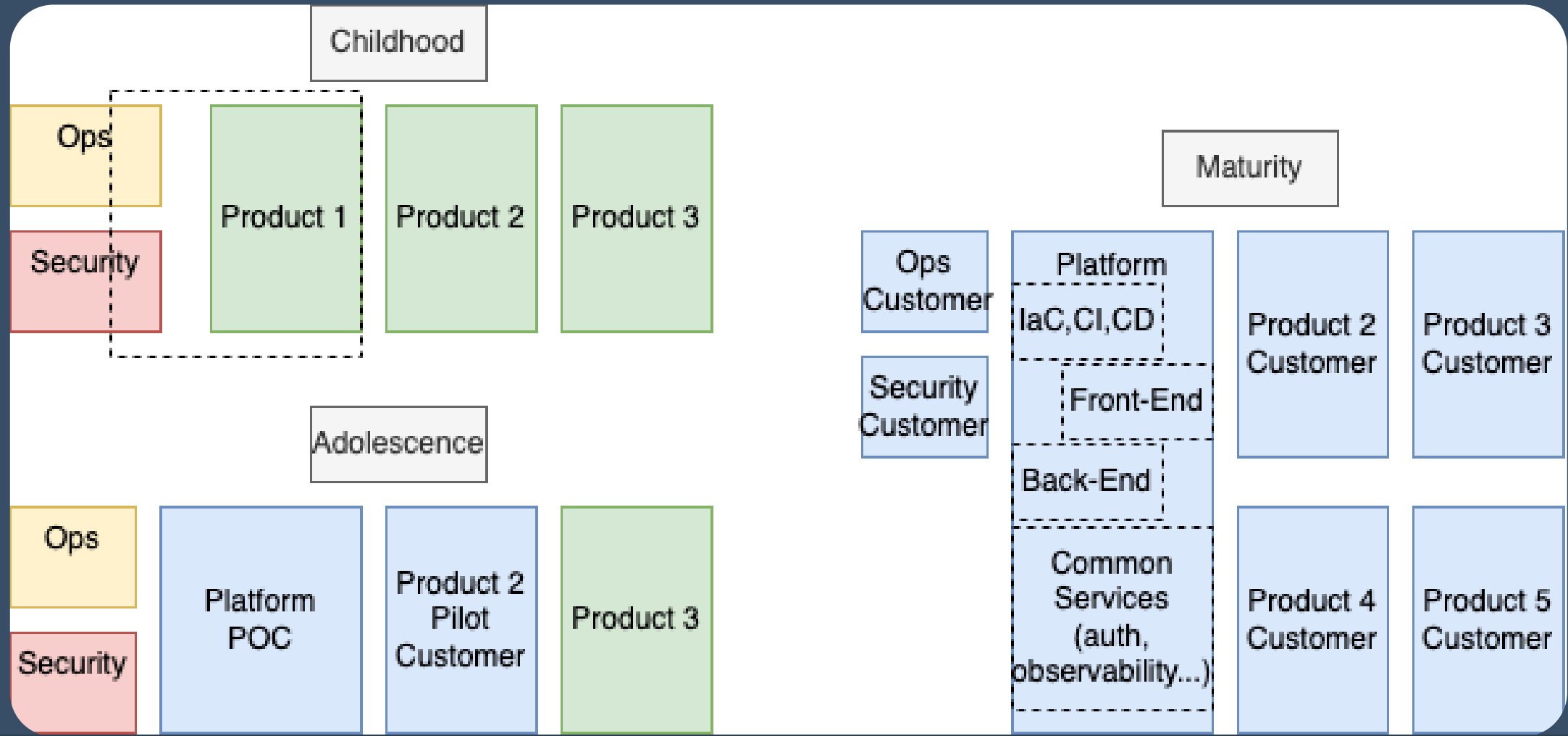
1. Define the **scope and objectives** of your platform
2. **Assess the current state** of your infrastructure and tools
3. Create a **plan** for building, maintaining, and scaling the platform
4. Establish a **cross-functional team** to work on the platform
5. **Implement IDP**, automation and monitoring tools
6. Establish **best practices and guidelines** for the teams using the platform
7. **Continuously improve** and adapt the platform to meet the evolving needs of the organization.

Teams Topology evolution example

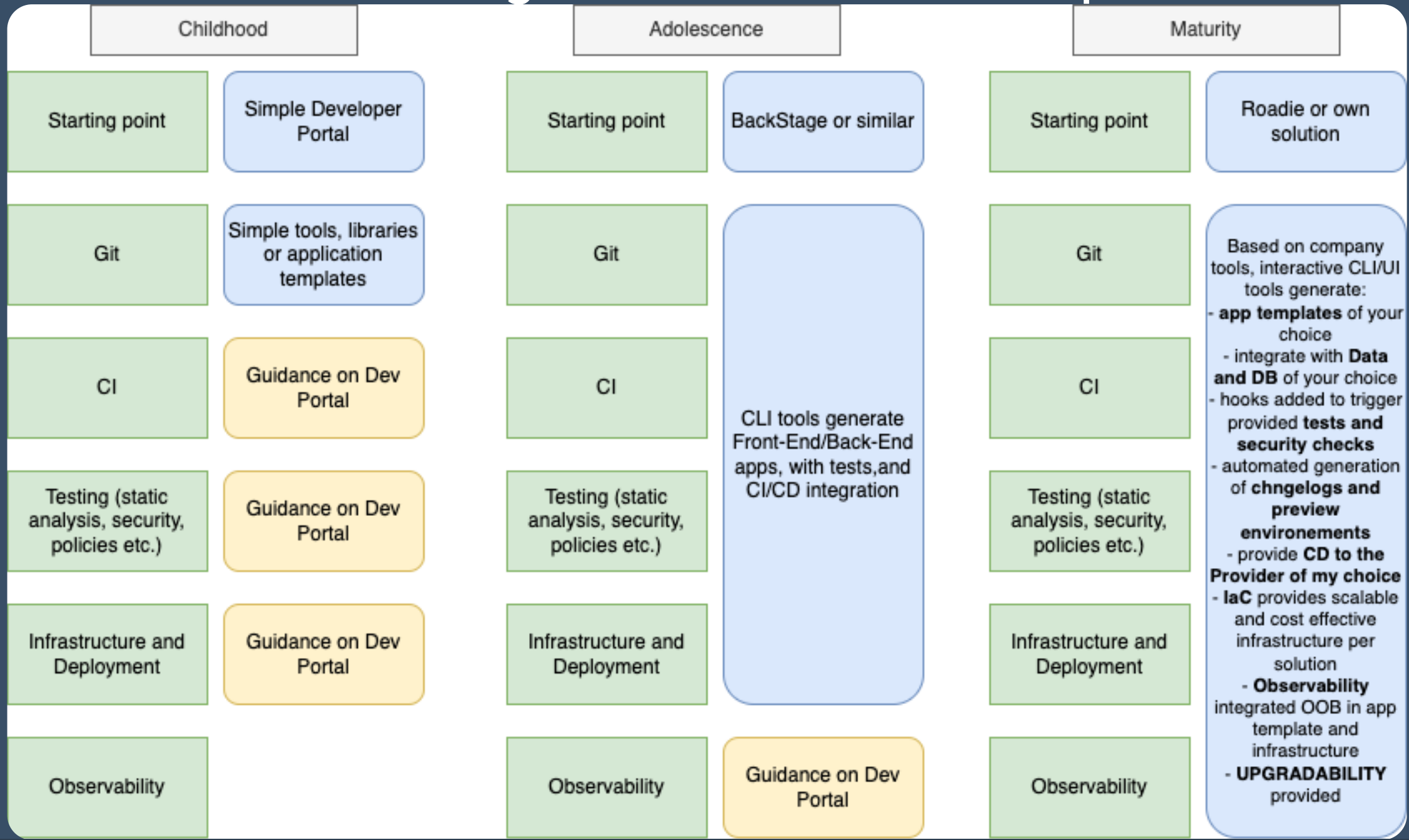


* Prerequisite: Define company common tools, like CI/CD (Gitlab, Github...), Cloud provider (AWS, Azure...), current target tech-stack (Java, React...) etc.
If it's not done, check if you need a Platform at this point.

- pick a pilot team carefully, to get a customer and evangelist
- target some of Product teams become "X as a service"



IDP* target and evolution example





How to create Platform Engineering org
and Platform itself?

Tooling: Engineering Portal

First step- simple static docs:

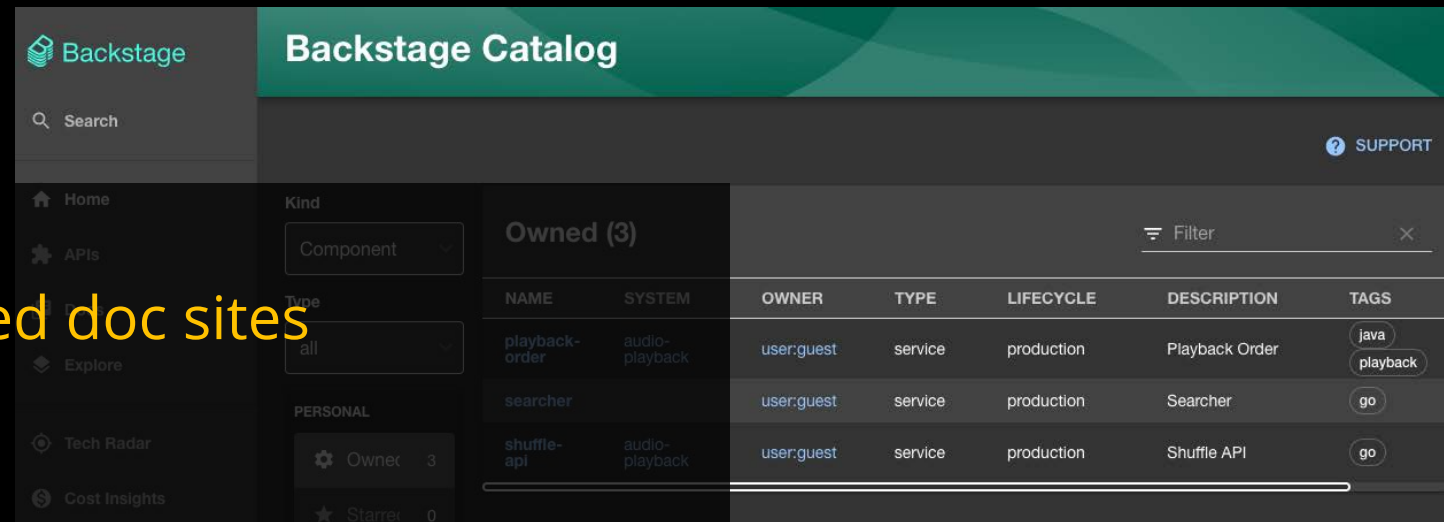
- Docusaurus- create simple versioned doc sites
- Slate- simple static docs for API

Next step:

- Backstage- open-source Internal Development Portal

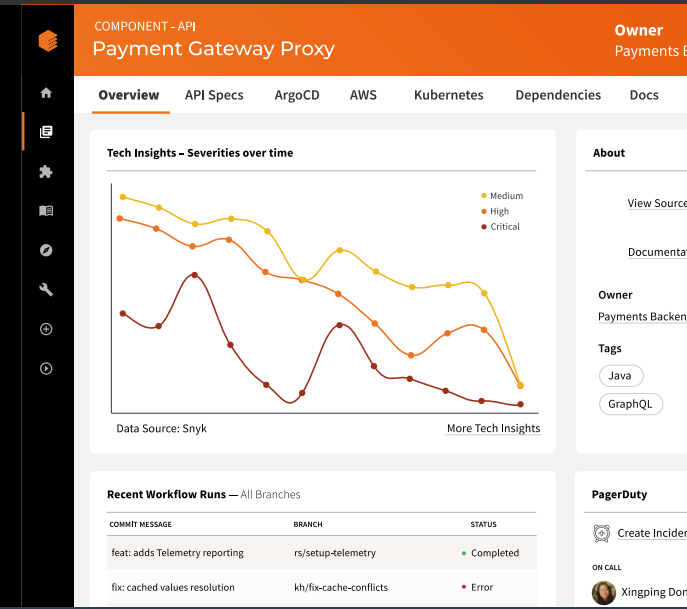
For scale:

- Roadie- paid Backstage SaaS



The screenshot shows the Backstage Catalog interface. The header includes the Backstage logo and 'Backstage Catalog'. A search bar is visible. The main content area displays a table of 'Owned (3)' components. The table has columns for NAME, SYSTEM, OWNER, TYPE, LIFECYCLE, DESCRIPTION, and TAGS. The components listed are 'playback-order', 'searcher', and 'shuffle-api'.

NAME	SYSTEM	OWNER	TYPE	LIFECYCLE	DESCRIPTION	TAGS
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

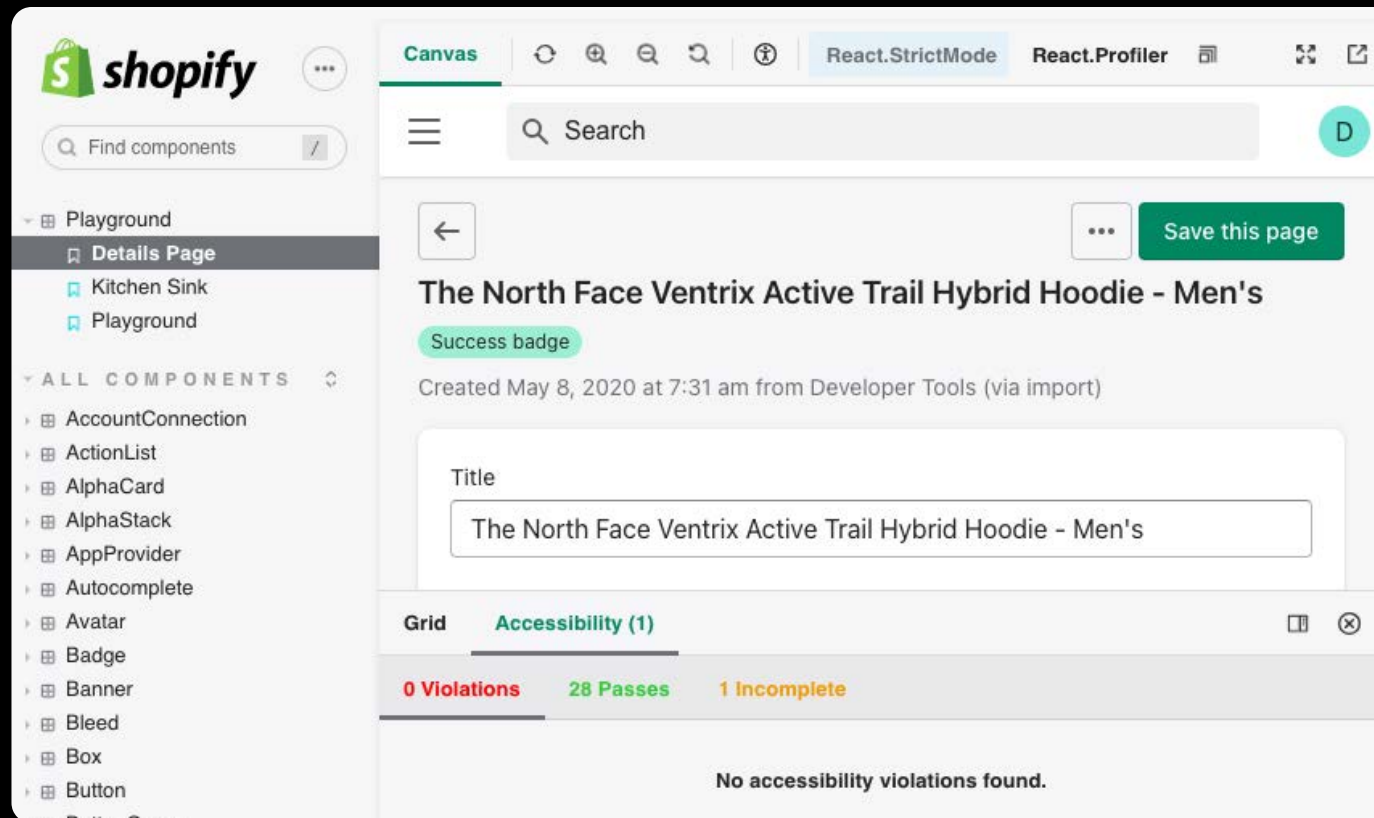


The screenshot shows the Backstage component page for 'Payment Gateway Proxy'. The page includes a navigation menu with options like Overview, API Specs, ArgoCD, AWS, Kubernetes, Dependencies, and Docs. A 'Tech Insights - Severities over time' chart is displayed, showing trends for Medium, High, and Critical severities. Below the chart, there is a 'Recent Workflow Runs' table and a 'PagerDuty' section with a 'Create Incident' button.

COMMIT MESSAGE	BRANCH	STATUS
feat: adds Telemetry reporting	rs/setup-telemetry	Completed
fix: cached values resolution	kh/fix-cache-conflicts	Error

Tooling: Front-End

- **Storybook** - Frontend workshop for UI development (React, Vue, Angular, JS)
- **Create React apps self-service** - Provide own app templates
- **Customize app config** - own app configs



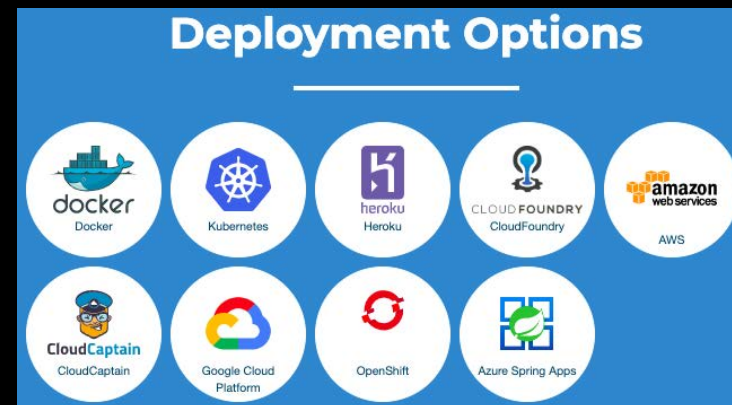
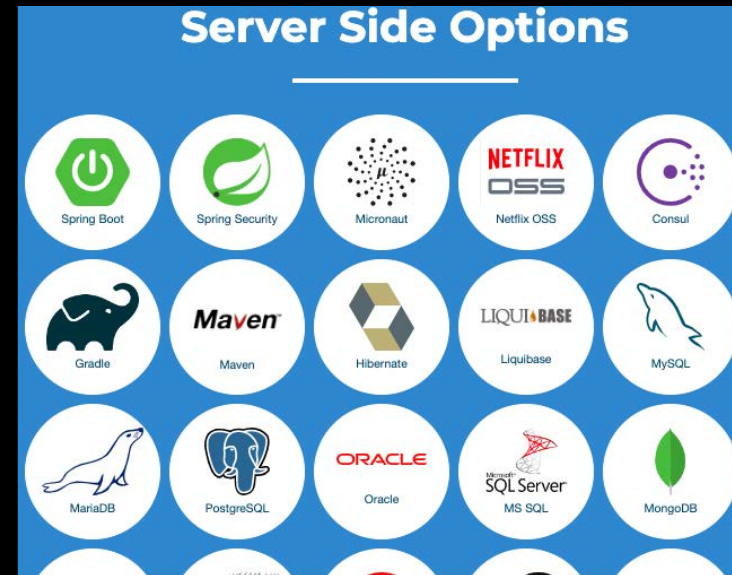
Tooling: Configurable Microservices/Apps self-service

Create a microservice [optionally, with microfrontend]

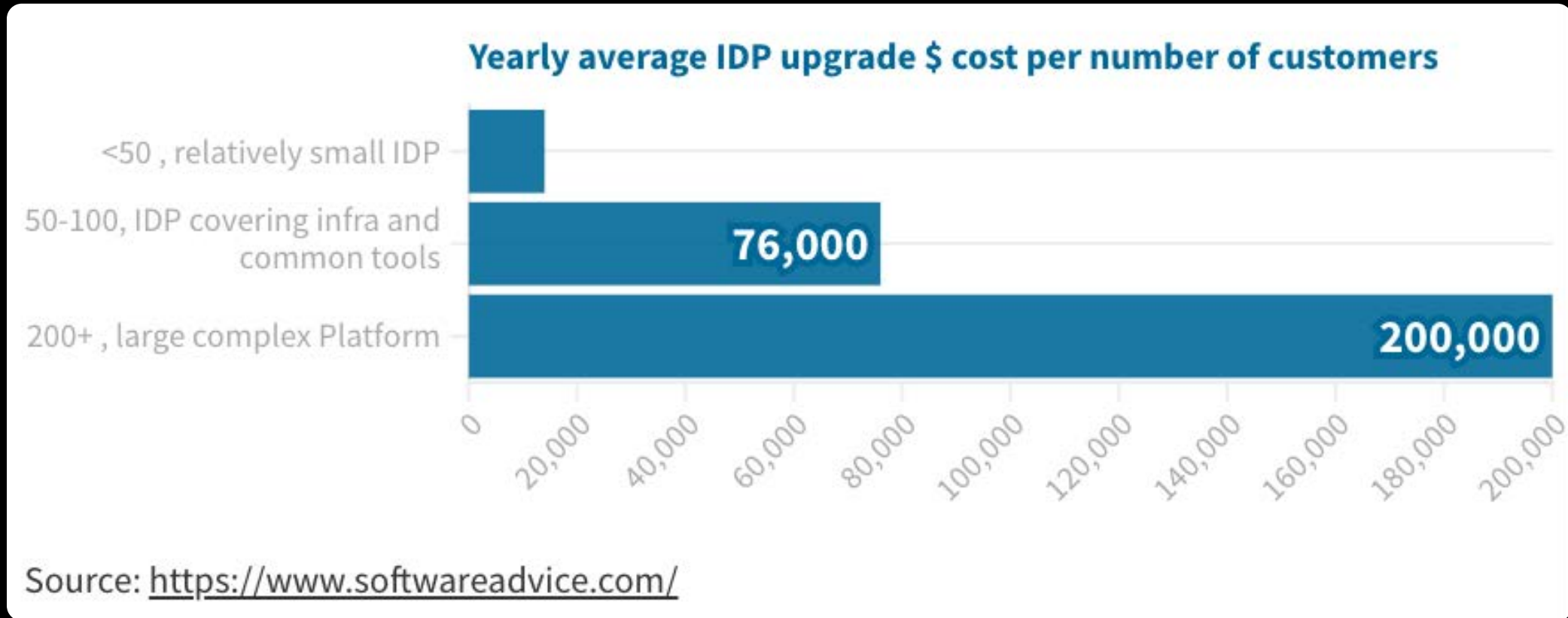
- <https://www.jhipster.tech/creating-an-app/> (CLI)
- <https://start.jhipster.tech/generate-application> (Web-based)

Other:

- Provide own application templates
- Integrate you CI/CD
- Define DB, Testing tooling, Monitoring etc.



Tooling: Upgradability



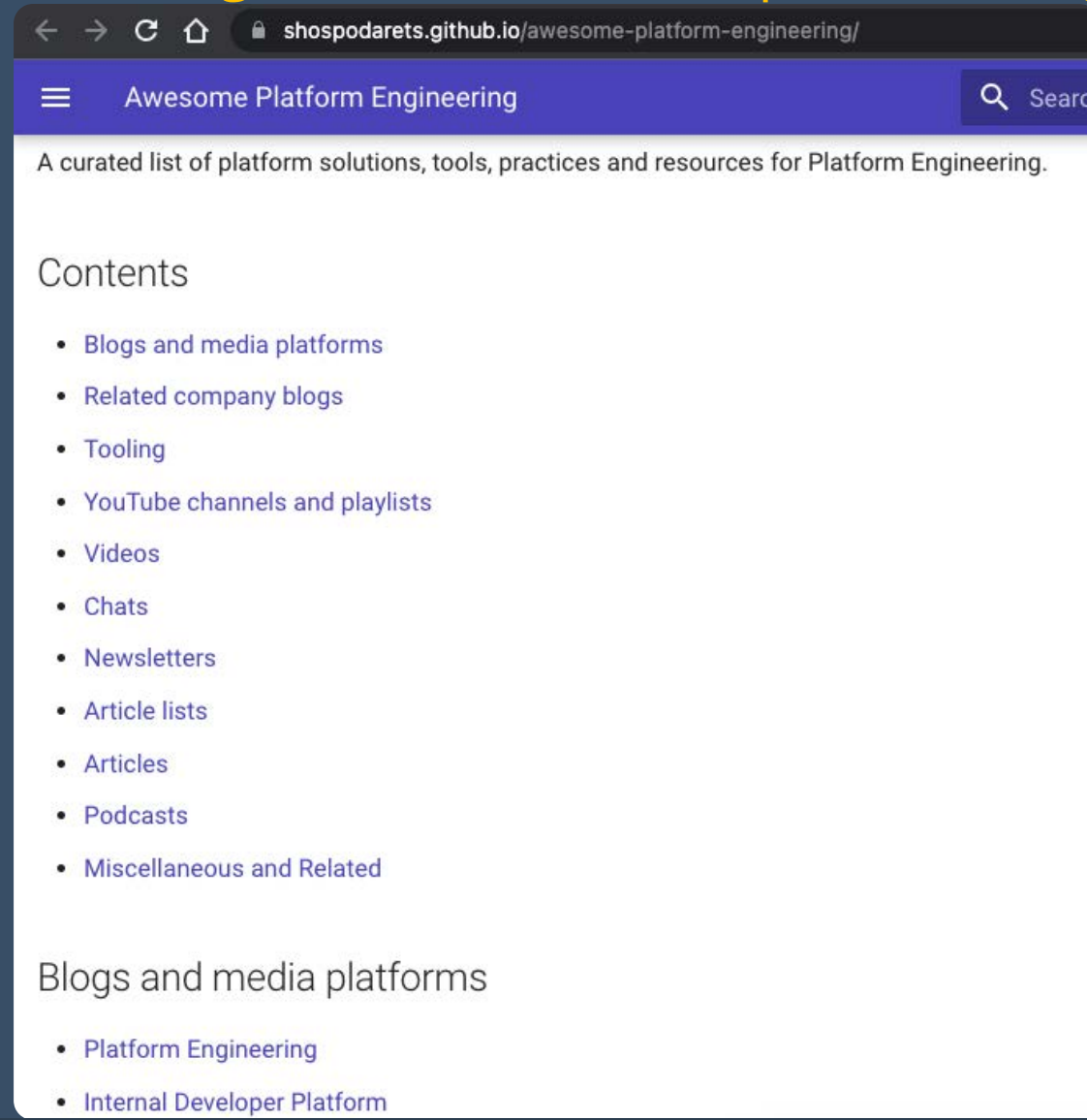
Front-End: **Codeshift** / **Codemods**

Microservice and infra: **JHipster Upgrading application**

Dependencies: **Renovate**- multi-platform and multi-language.

More details and resources on Platform Engineering

<https://shospodarets.github.io/awesome-platform-engineering/>



The screenshot shows a web browser displaying the 'Awesome Platform Engineering' website. The browser's address bar shows the URL 'shospodarets.github.io/awesome-platform-engineering/'. The website has a purple header with a search bar and a navigation menu. Below the header, there is a description: 'A curated list of platform solutions, tools, practices and resources for Platform Engineering.' The main content area is titled 'Contents' and lists various categories: 'Blogs and media platforms', 'Related company blogs', 'Tooling', 'YouTube channels and playlists', 'Videos', 'Chats', 'Newsletters', 'Article lists', 'Articles', 'Podcasts', and 'Miscellaneous and Related'. Below this list, there is a section titled 'Blogs and media platforms' which includes 'Platform Engineering' and 'Internal Developer Platform'.

← → ↻ 🏠 shospodarets.github.io/awesome-platform-engineering/

☰ Awesome Platform Engineering 🔍 Search

A curated list of platform solutions, tools, practices and resources for Platform Engineering.

Contents

- [Blogs and media platforms](#)
- [Related company blogs](#)
- [Tooling](#)
- [YouTube channels and playlists](#)
- [Videos](#)
- [Chats](#)
- [Newsletters](#)
- [Article lists](#)
- [Articles](#)
- [Podcasts](#)
- [Miscellaneous and Related](#)

Blogs and media platforms

- [Platform Engineering](#)
- [Internal Developer Platform](#)

Tips from experience

- Pilot team(s) is super important to become evangelists

- Treat as product as soon as possible- enable business problem, and use technology to do it

1. Treat internal teams as customers, provide support, Slack etc. channels

2. Invest in Platform PM and DevX culture

3. Evangelise- regular releases, demos, newsletters

- For UI adoption put heads together with the UX

- Jackpot Platform strategy- find a way and deliver your platform to your company customers



**Experience is simply the name
we give our mistakes.**

Oscar Wilde

Inspirational 

- Prefer golden path over golden cage

- Always think on Thinnest Viable Platform

- Target the right level of abstraction

- Establish the proper "X-as-Service" or similar definitions for each team

- Foster the Pull vs Push culture

- Work with leadership for a top-down support for bottom-up enablement

THANK YOU!

A JOURNEY OF A
THOUSAND MILES
BEGINS WITH A
SINGLE STEP

LAO TZU

Slides:

[https://slides.com/shospodarets/
platform-engineering](https://slides.com/shospodarets/platform-engineering)

Platform Engineering resources:

[https://github.com/shospodarets/
awesome-platform-engineering](https://github.com/shospodarets/awesome-platform-engineering) ★

Serg Hospodarets

