Refactoring monolith to microservices

... but Kubernetes

Pavol Hronsky

Head of Container Automation
Nordea

- Background in applied mathematics
- Java developer
- Working with containers since 2017
- Likes memes

Configuration drift

Version conflicts

Inconsistent environments

Slow deployment

Inability to scale

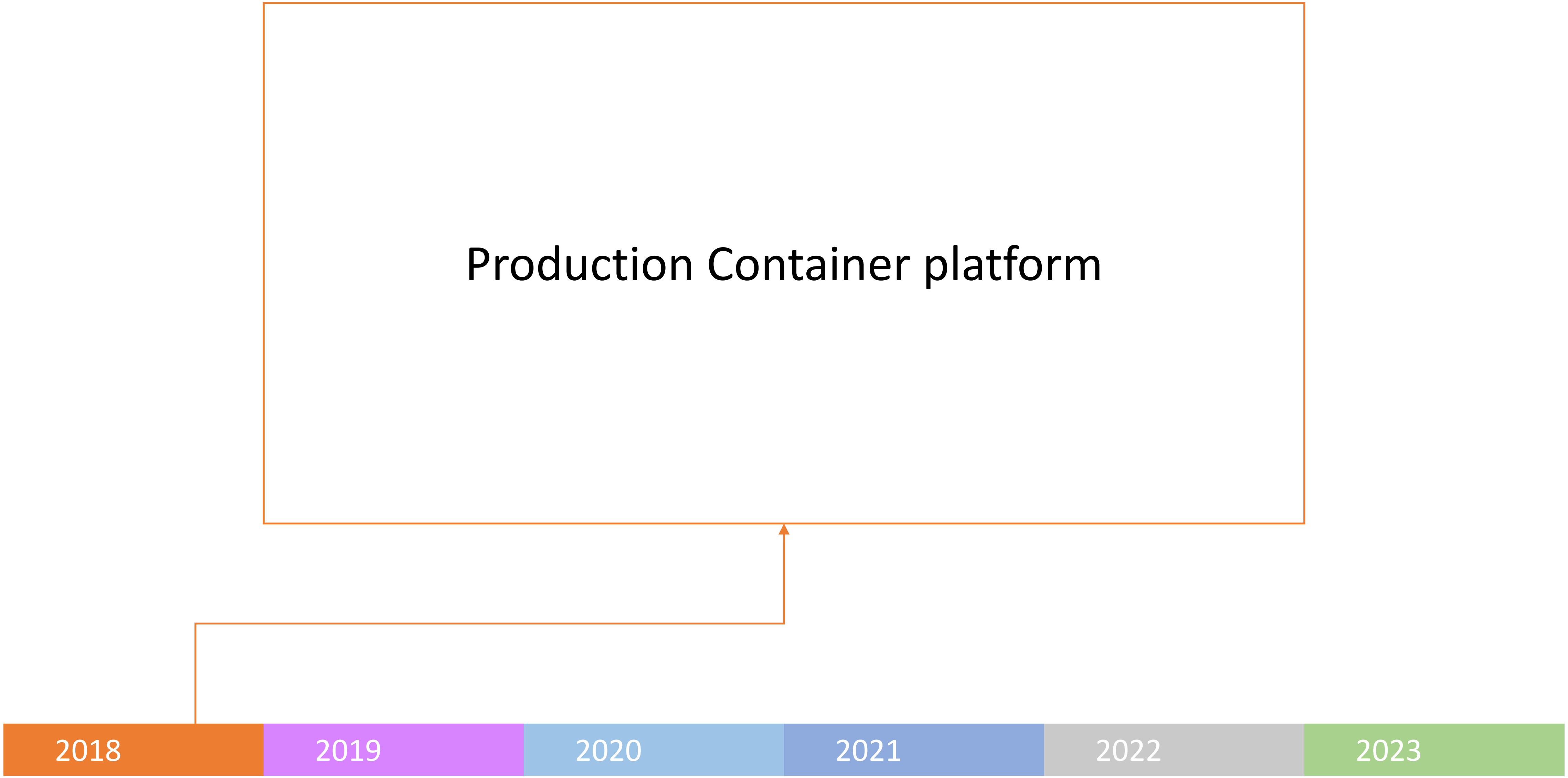
Limited visibility

Lack of flexibility

CONTAINERS
CONTAINER EVERYWHERE
imgflip.com

Building a monolith (on-prem)

…unintentionally



Production Container platform

2018

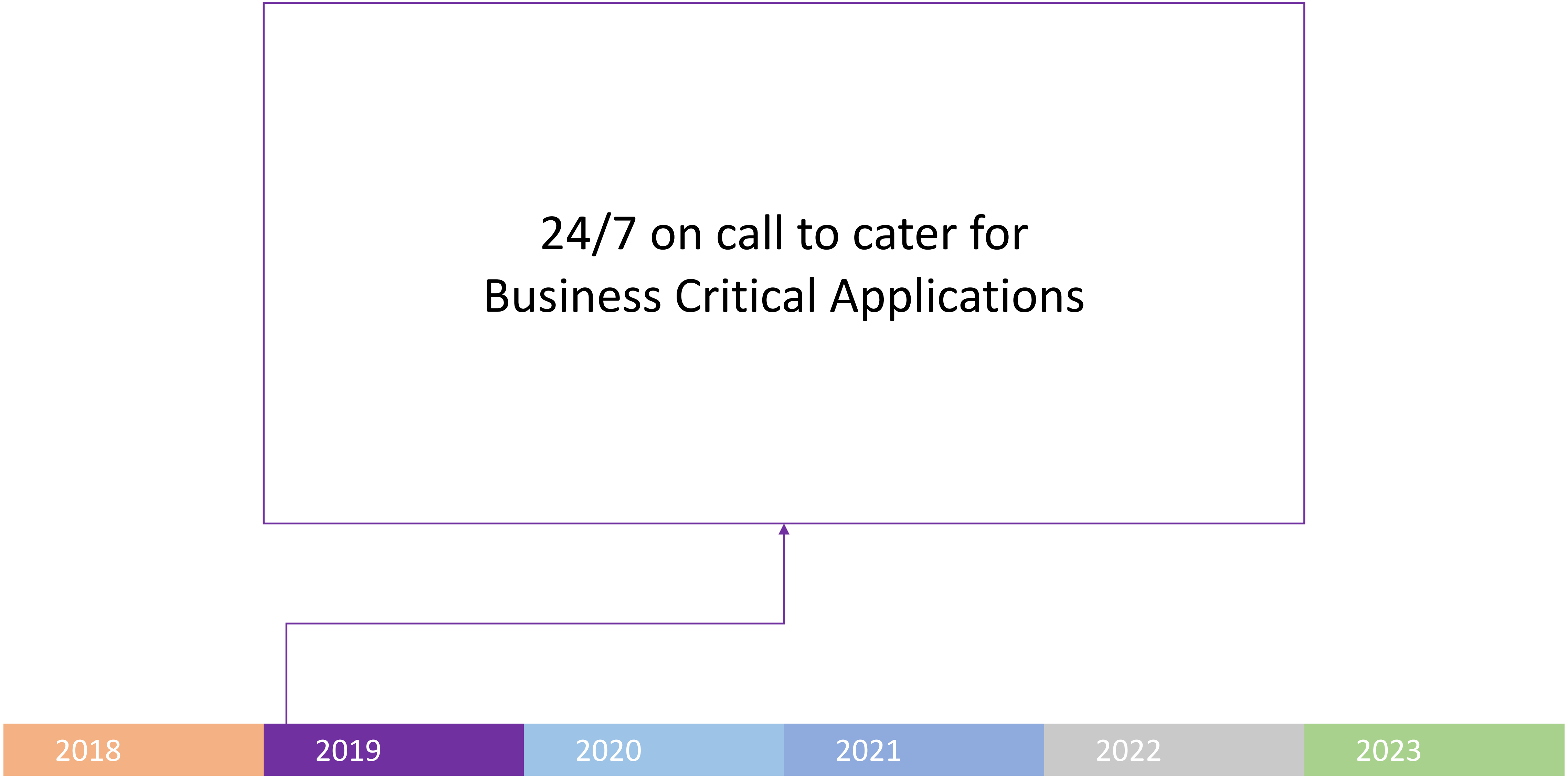
2019

2020

2021

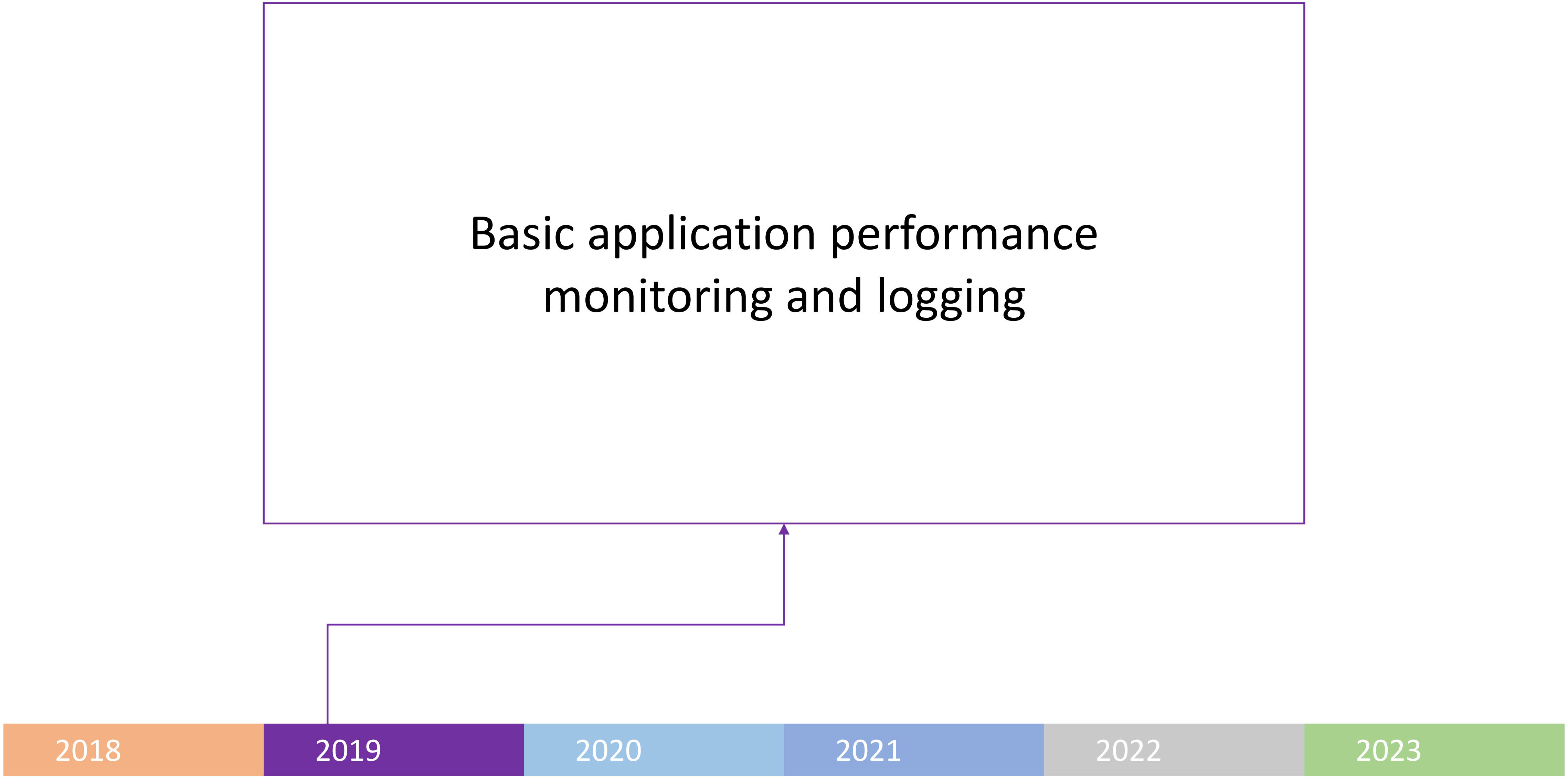
2022

2023

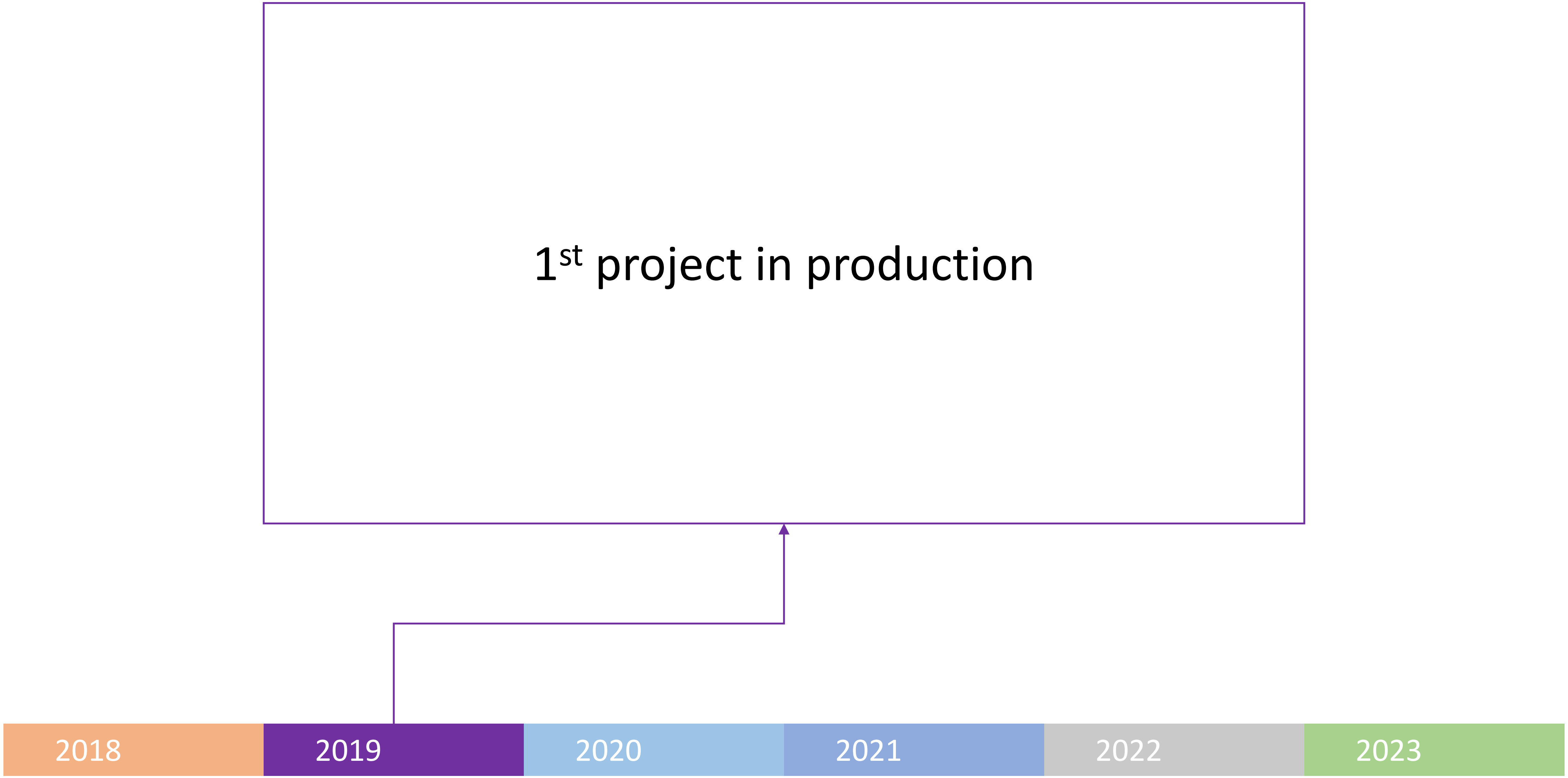


24/7 on call to cater for
Business Critical Applications

2018 2019 2020 2021 2022 2023

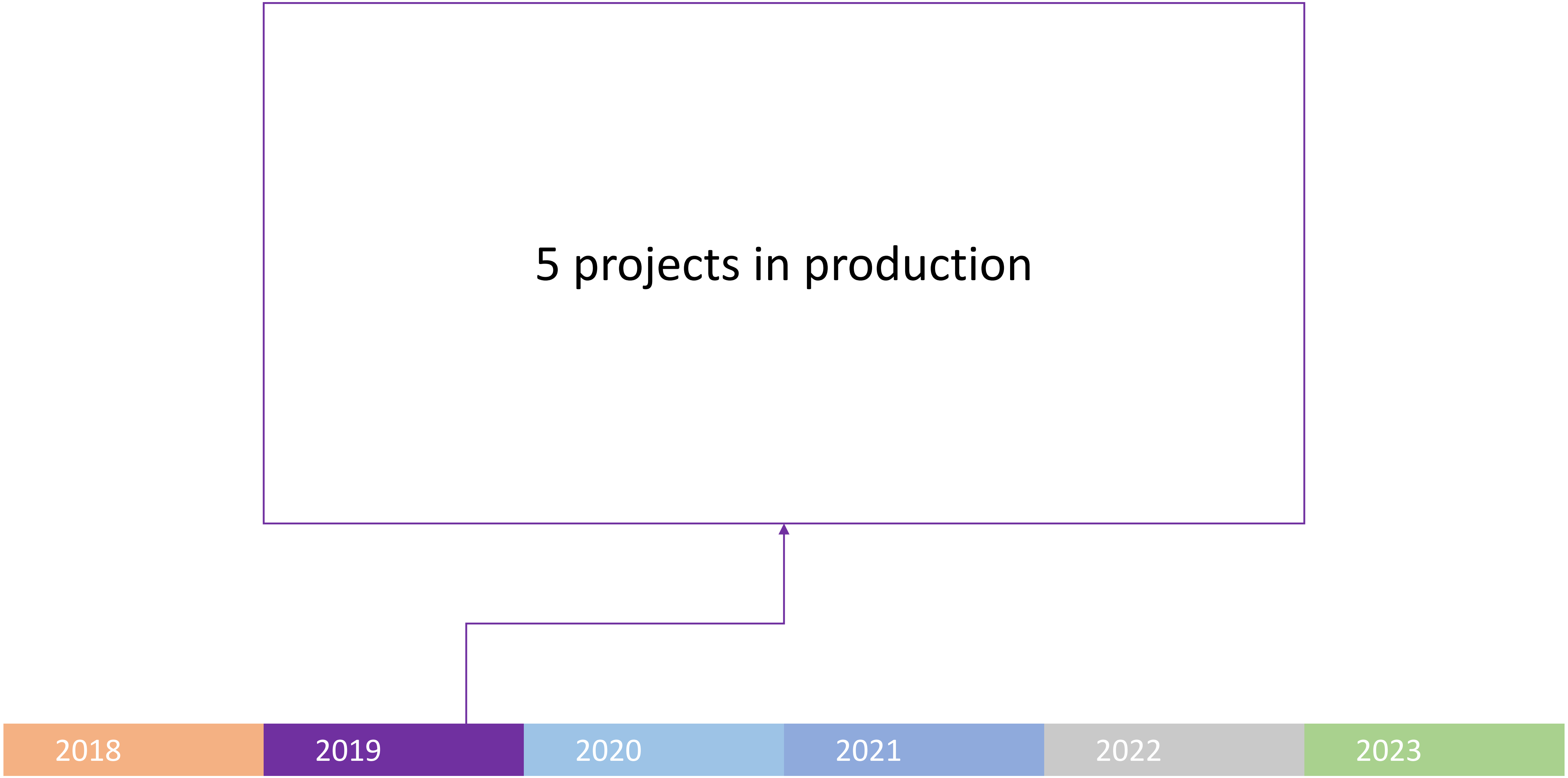


Basic application performance monitoring and logging

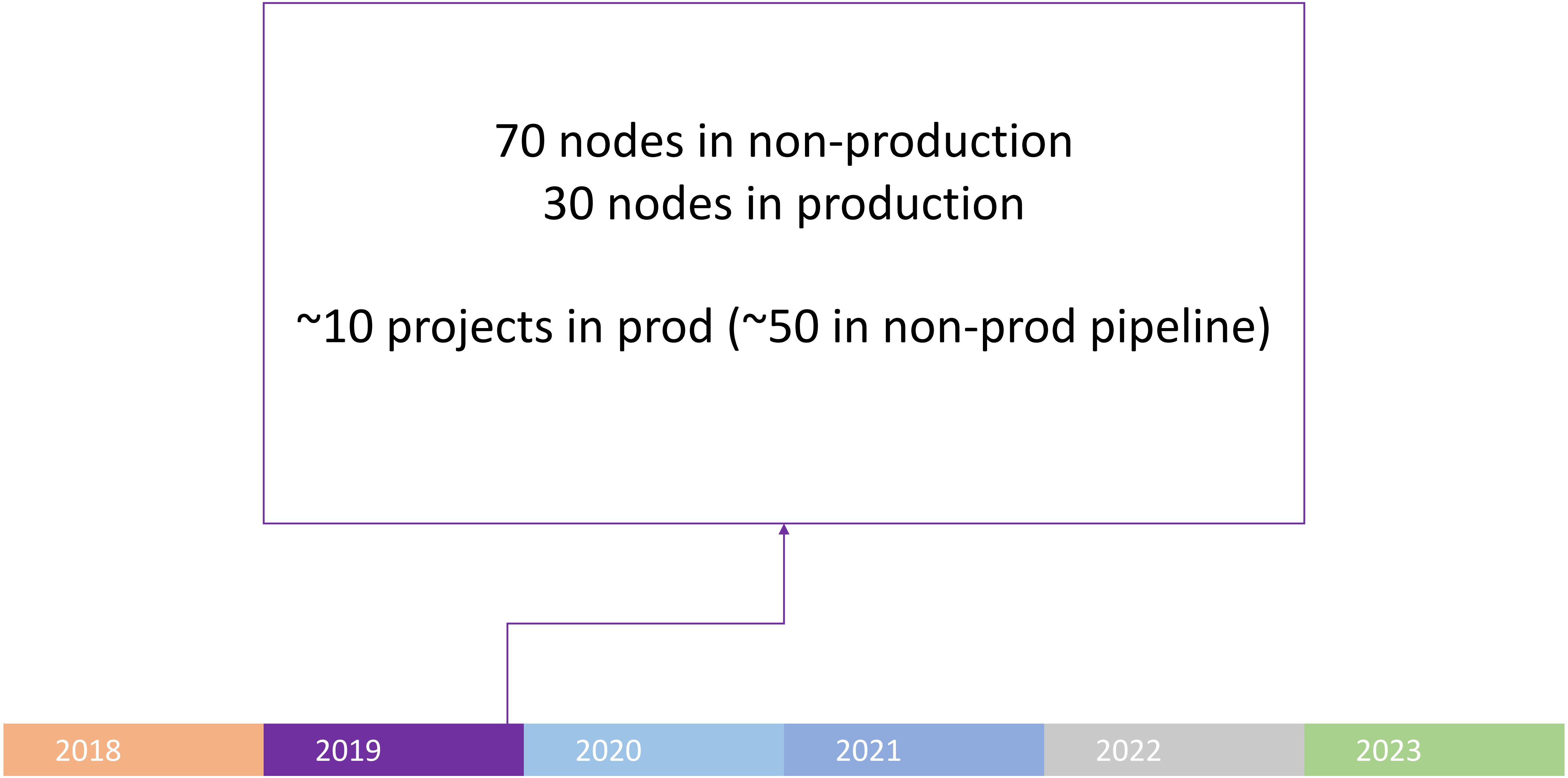


1st project in production

2018 2019 2020 2021 2022 2023

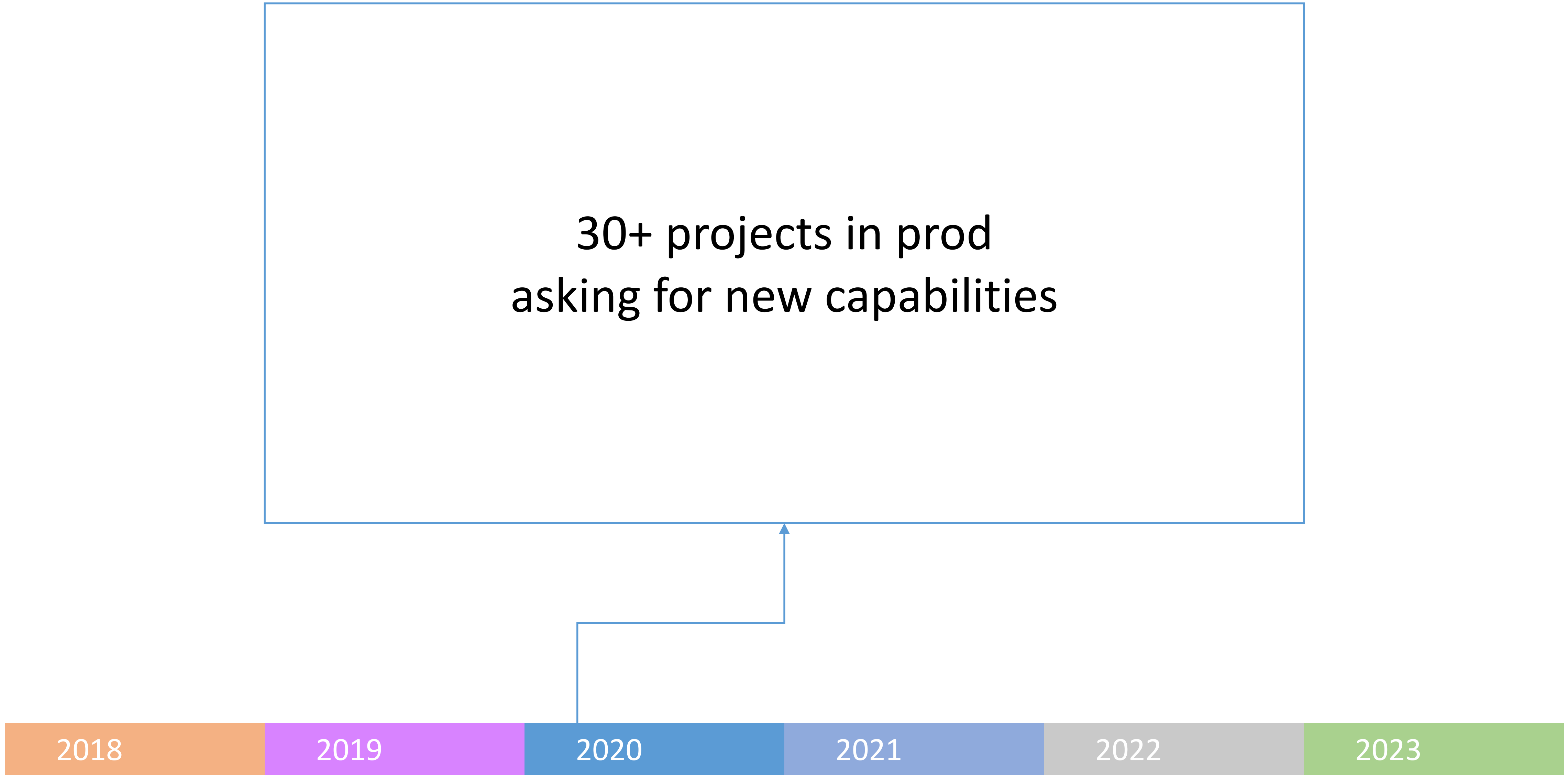


5 projects in production

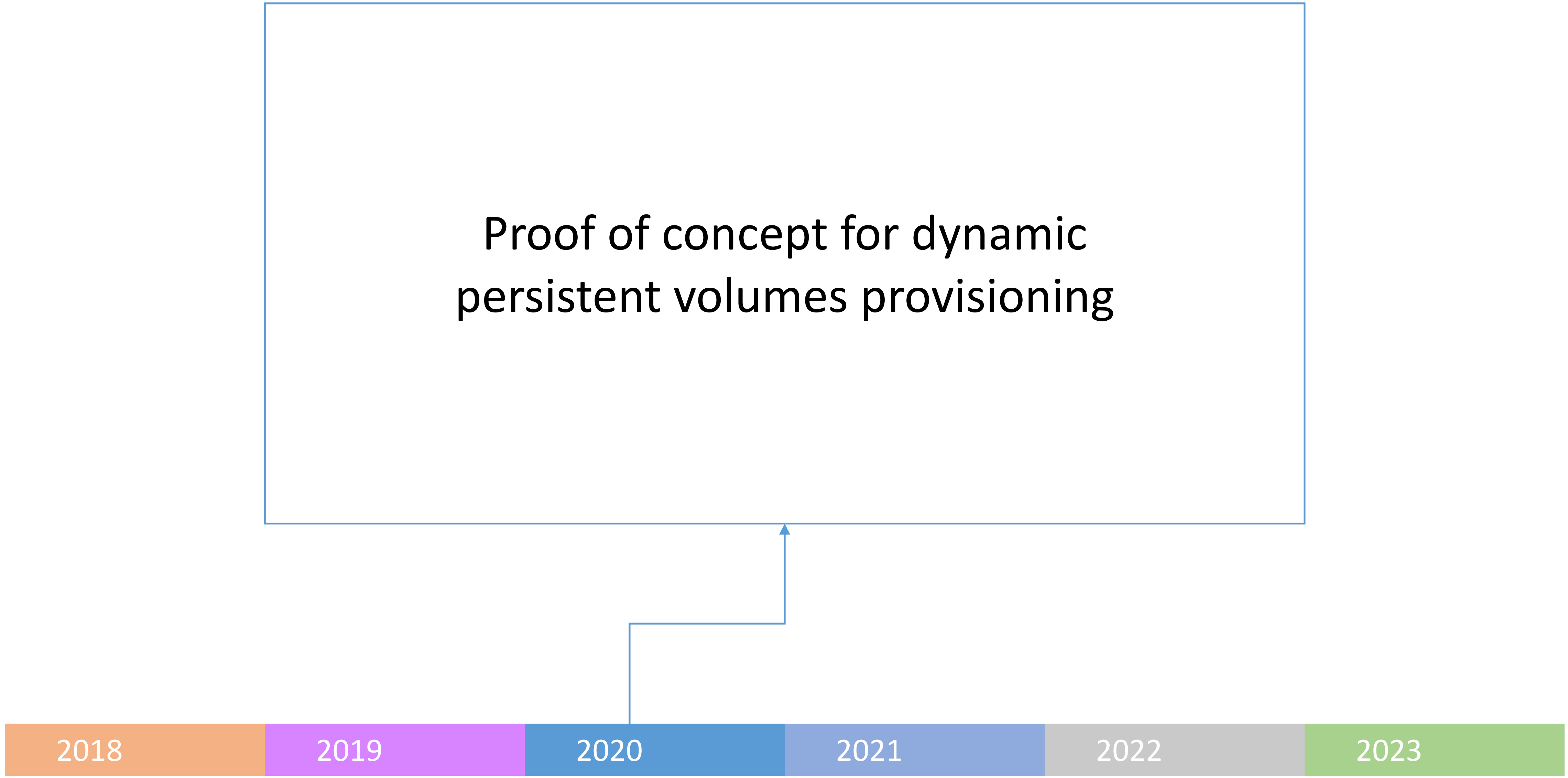


70 nodes in non-production
30 nodes in production

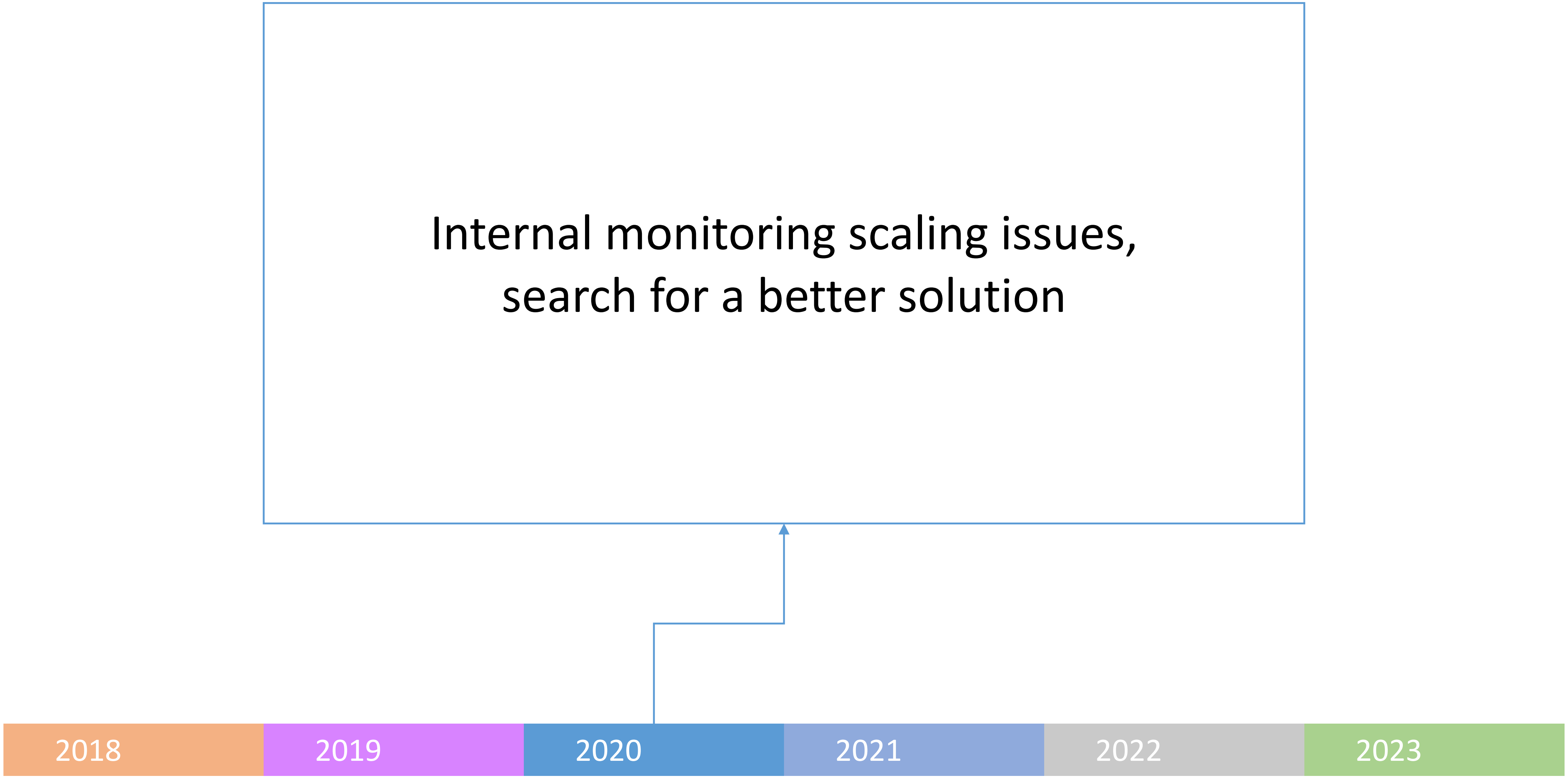
~10 projects in prod (~50 in non-prod pipeline)



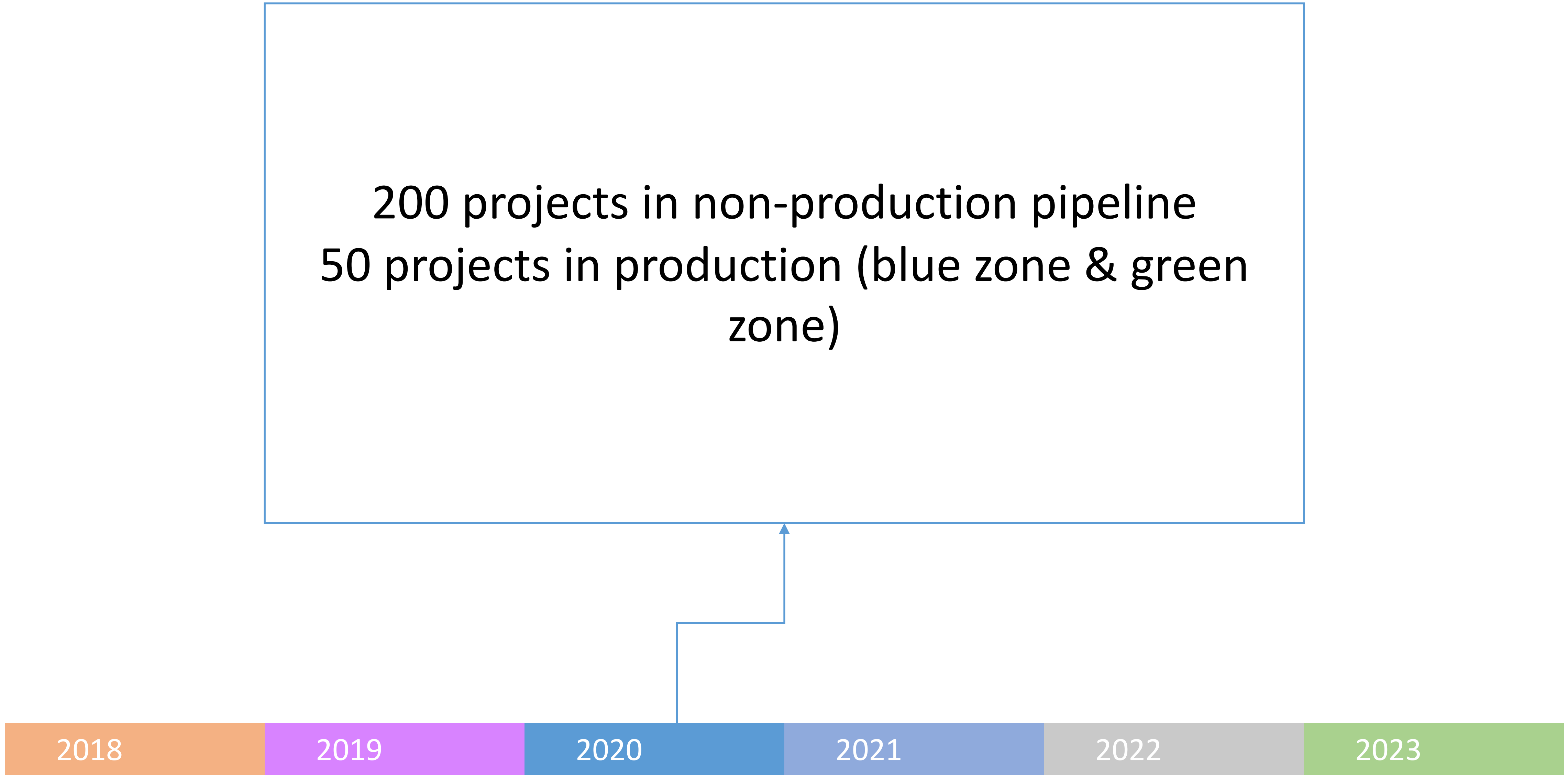
30+ projects in prod
asking for new capabilities



Proof of concept for dynamic persistent volumes provisioning

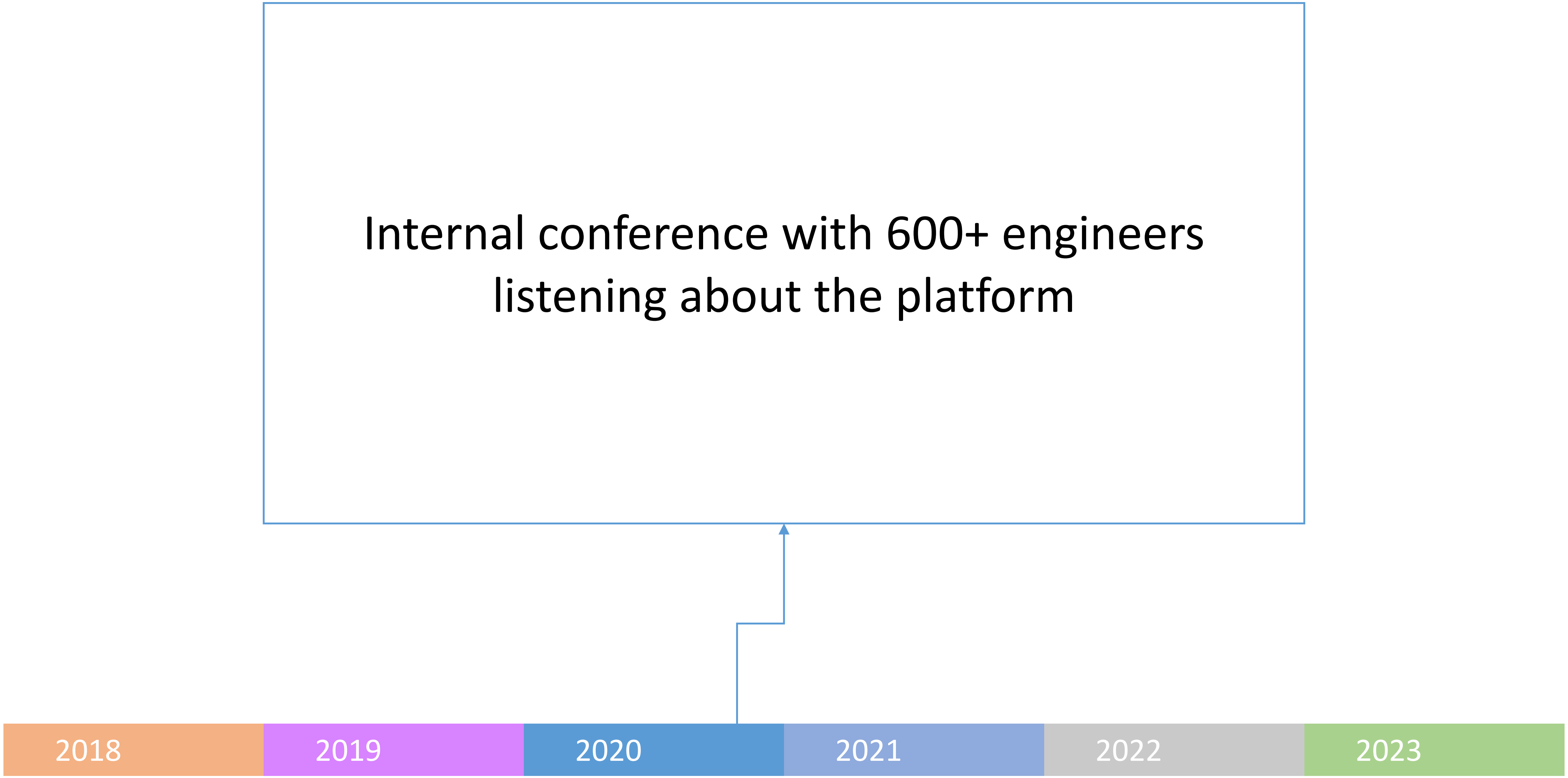


Internal monitoring scaling issues, search for a better solution



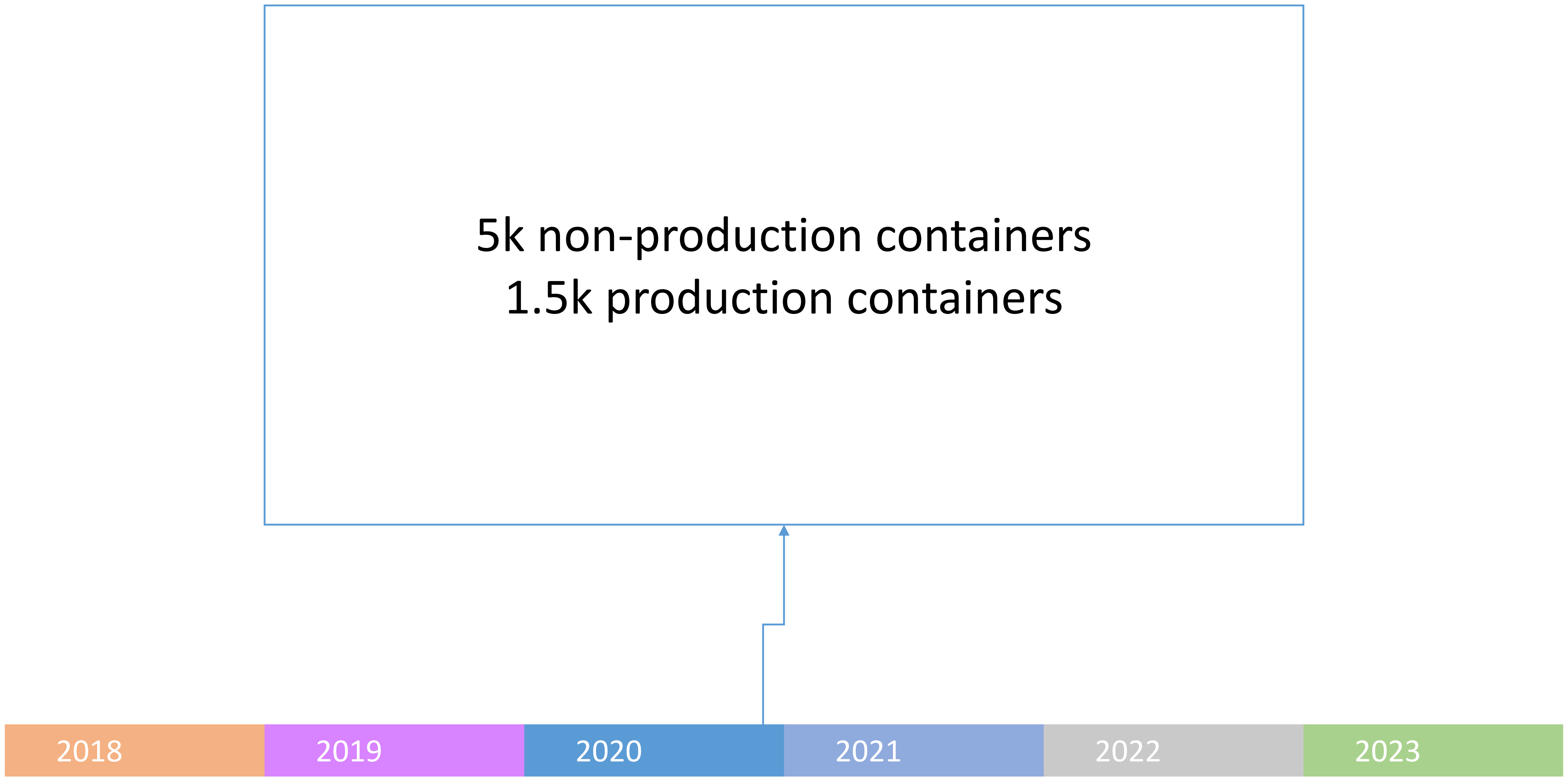
200 projects in non-production pipeline
50 projects in production (blue zone & green zone)

2018 2019 2020 2021 2022 2023



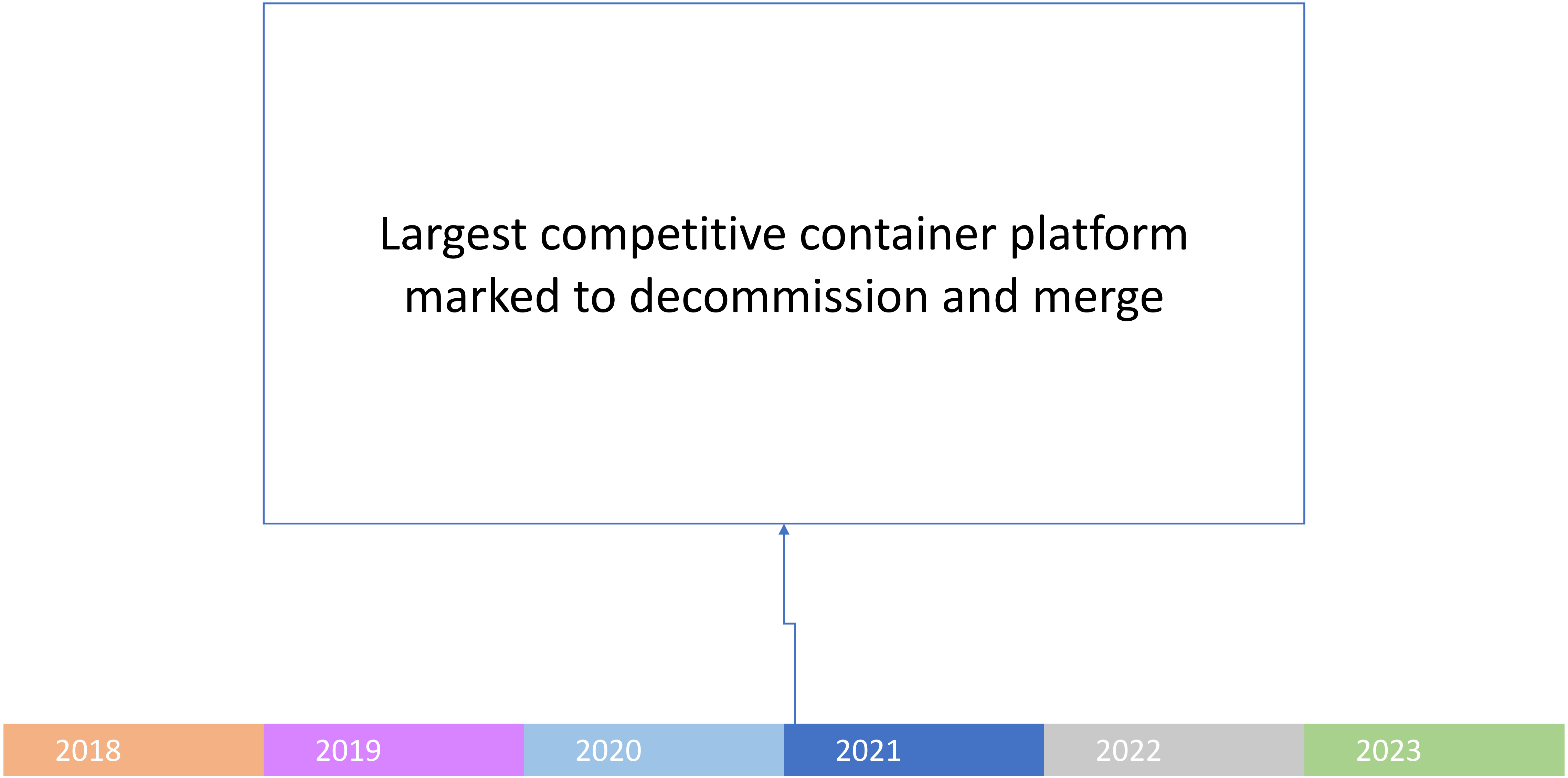
Internal conference with 600+ engineers listening about the platform

2018 2019 2020 2021 2022 2023



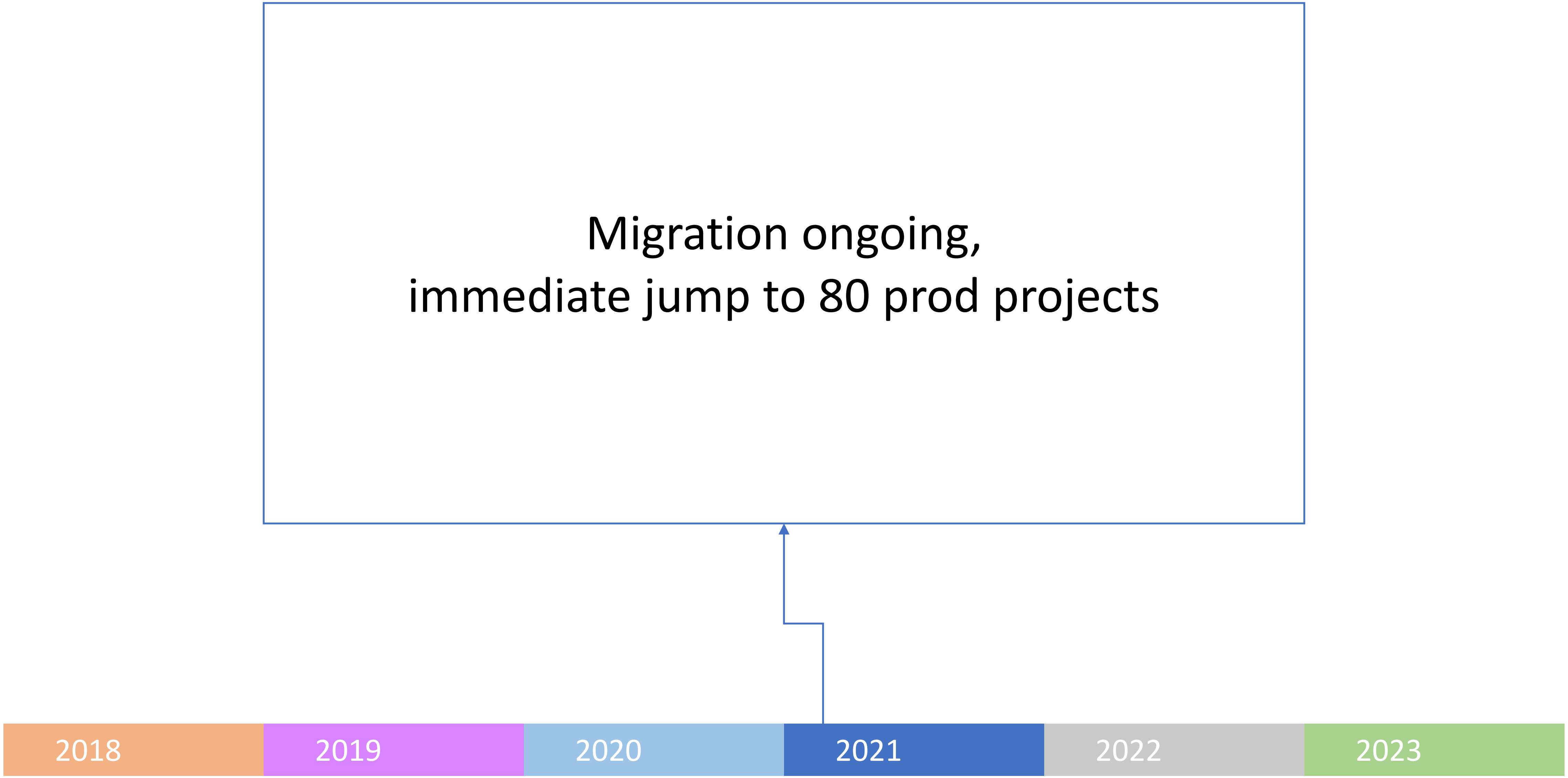
5k non-production containers
1.5k production containers

2018 | 2019 | 2020 | 2021 | 2022 | 2023



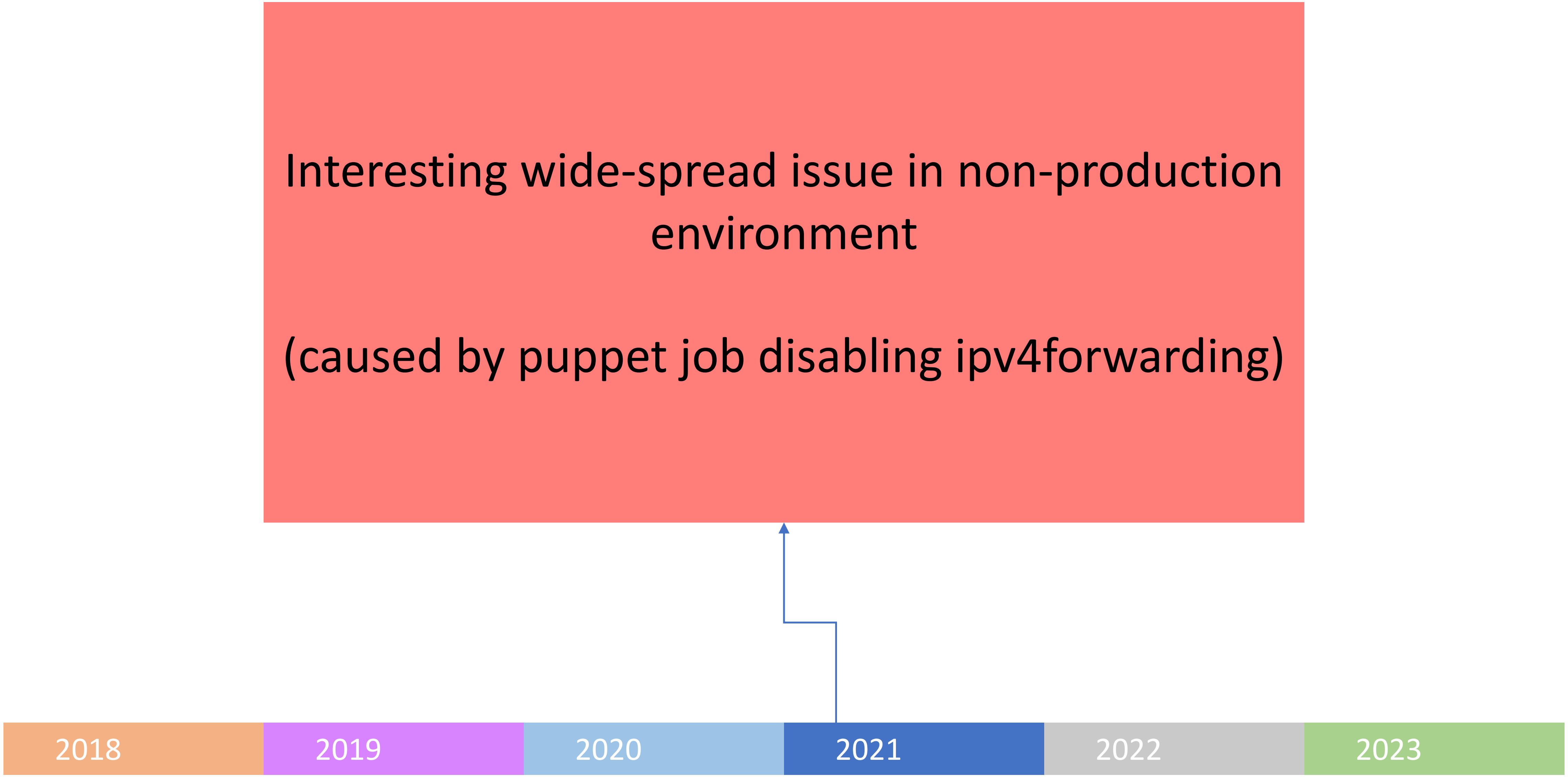
Largest competitive container platform marked to decommission and merge

2018 2019 2020 2021 2022 2023



Migration ongoing,
immediate jump to 80 prod projects

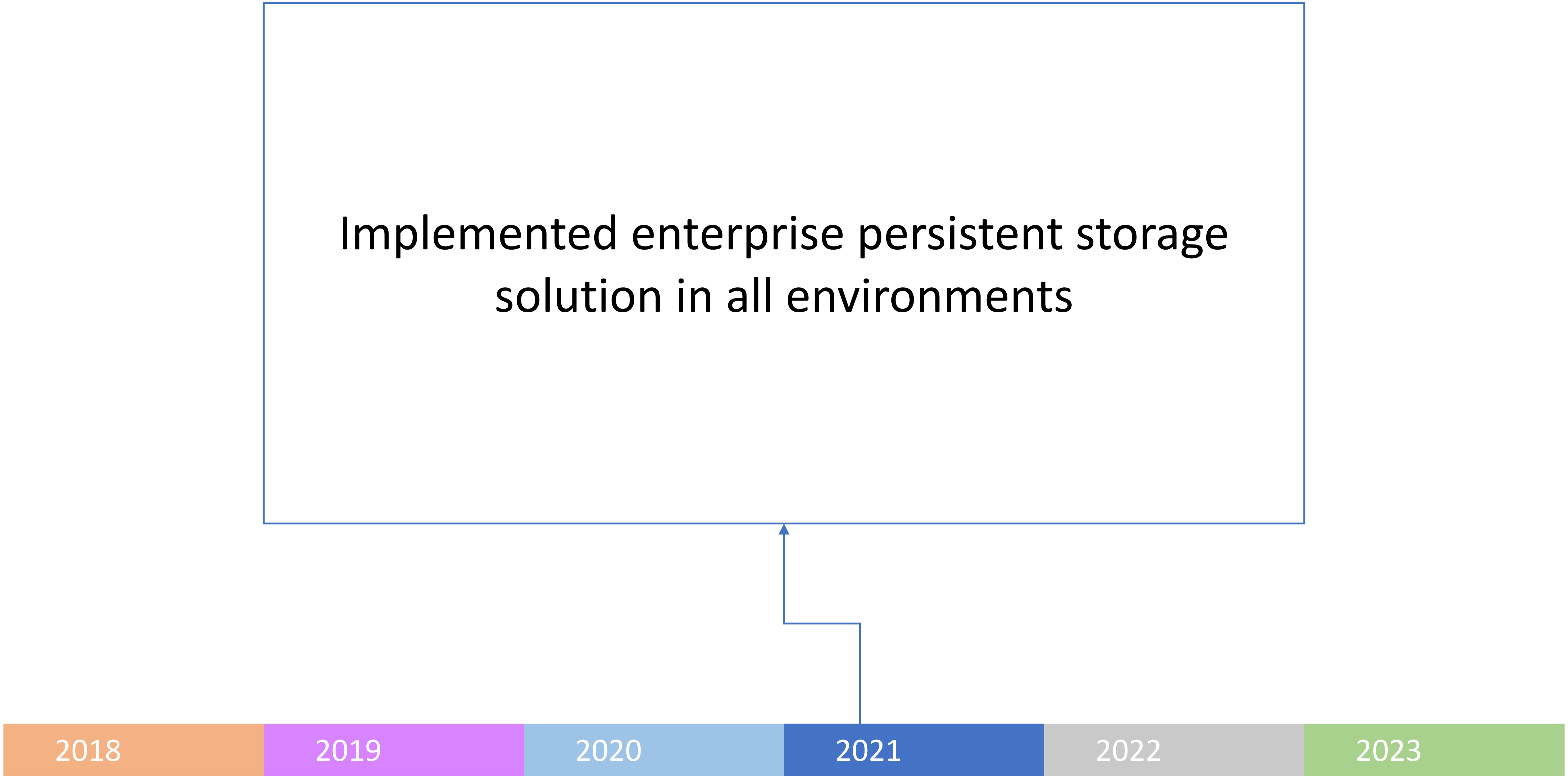
2018 2019 2020 2021 2022 2023



Interesting wide-spread issue in non-production environment

(caused by puppet job disabling ipv4forwarding)

2018 2019 2020 2021 2022 2023



Implemented enterprise persistent storage solution in all environments

2018

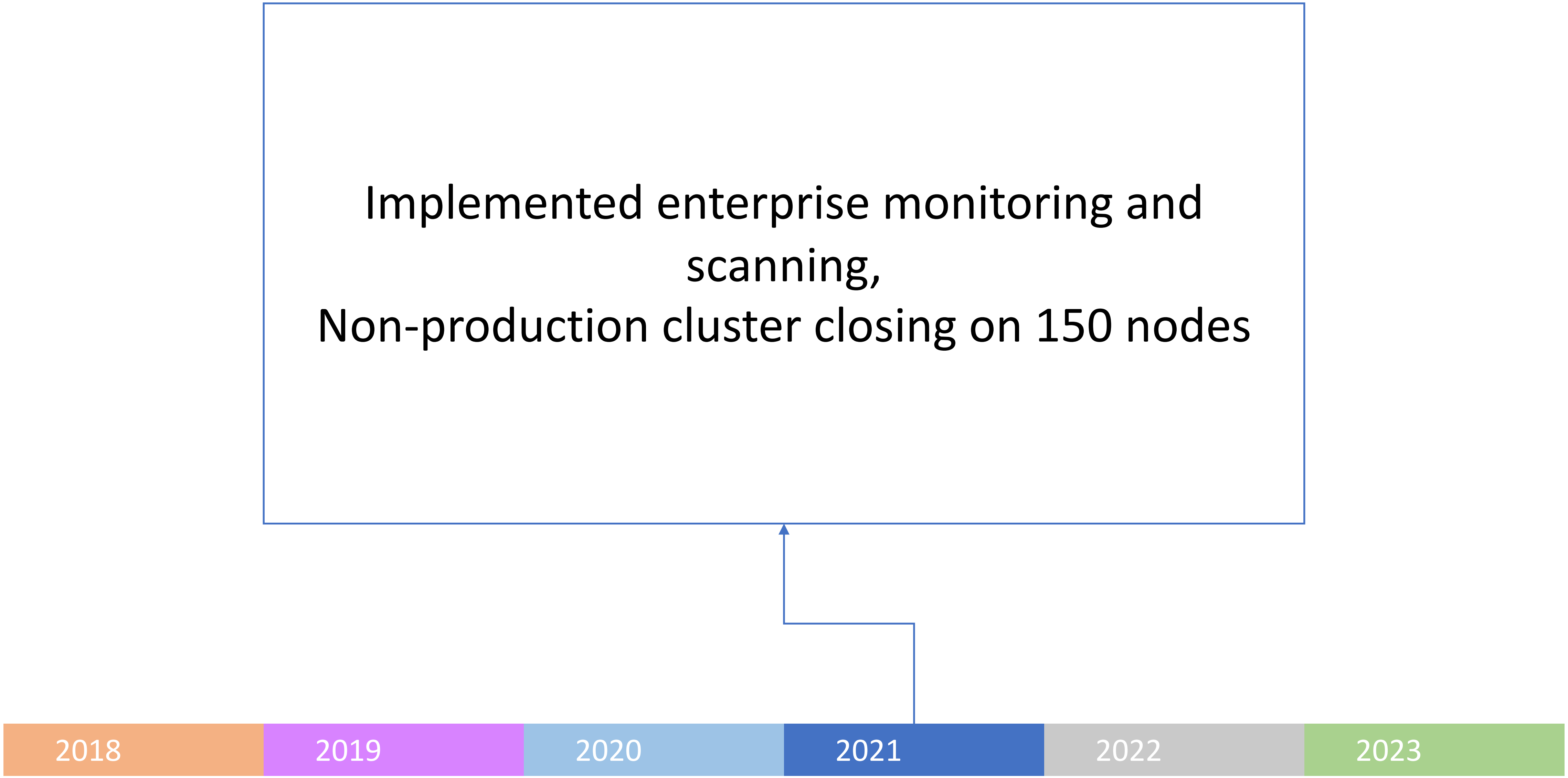
2019

2020

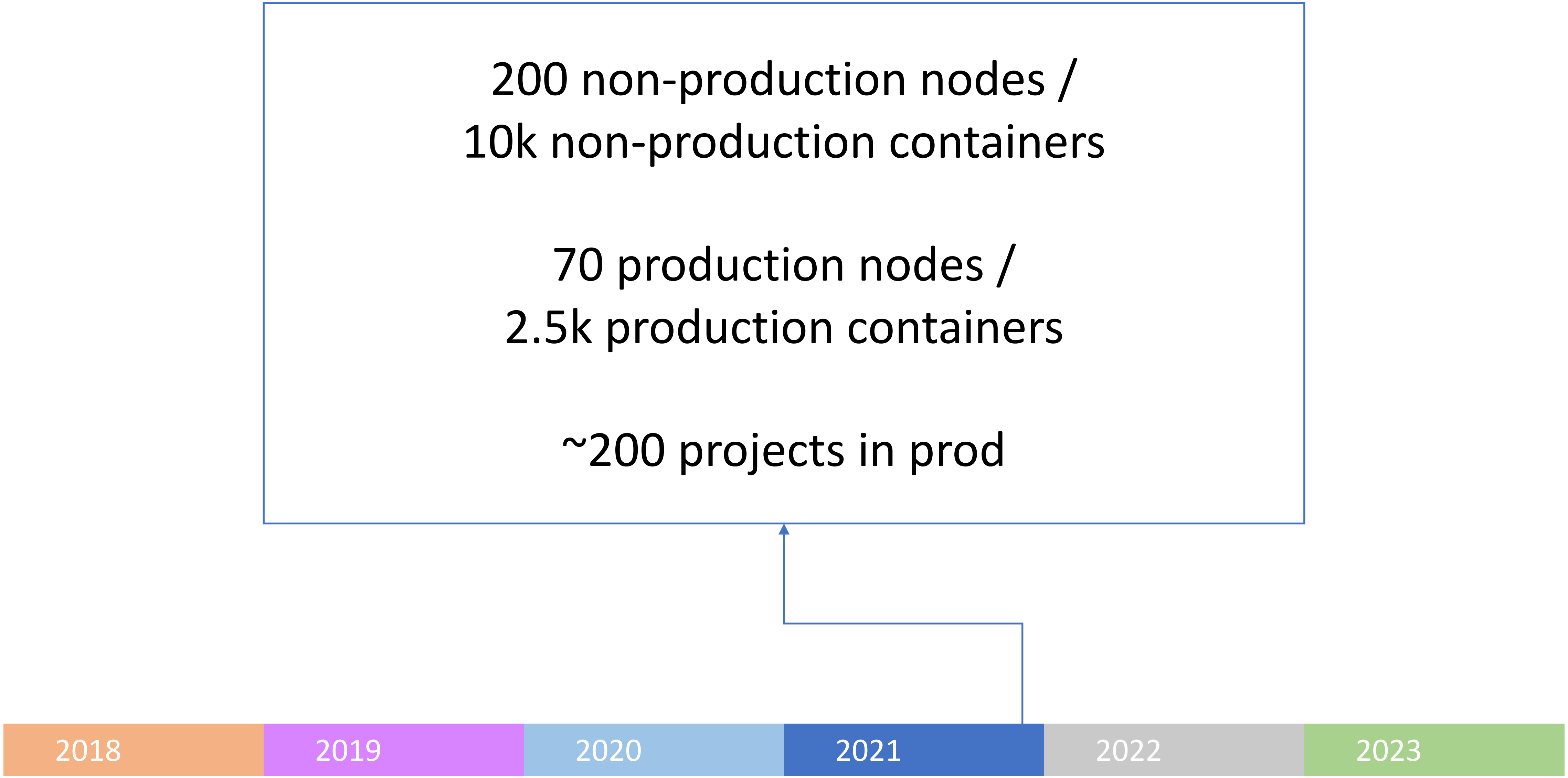
2021

2022

2023



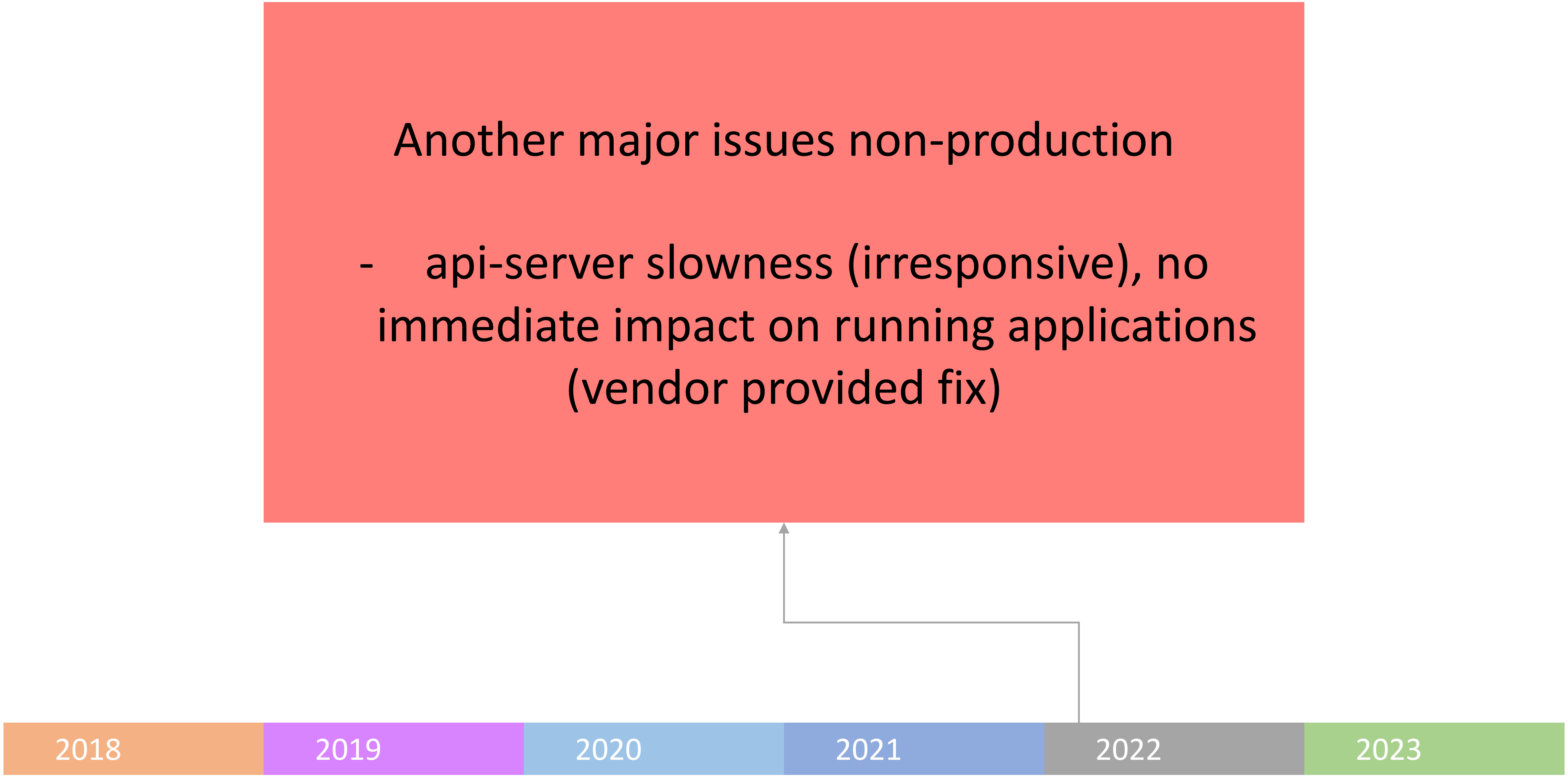
Implemented enterprise monitoring and scanning,
Non-production cluster closing on 150 nodes



200 non-production nodes /
10k non-production containers

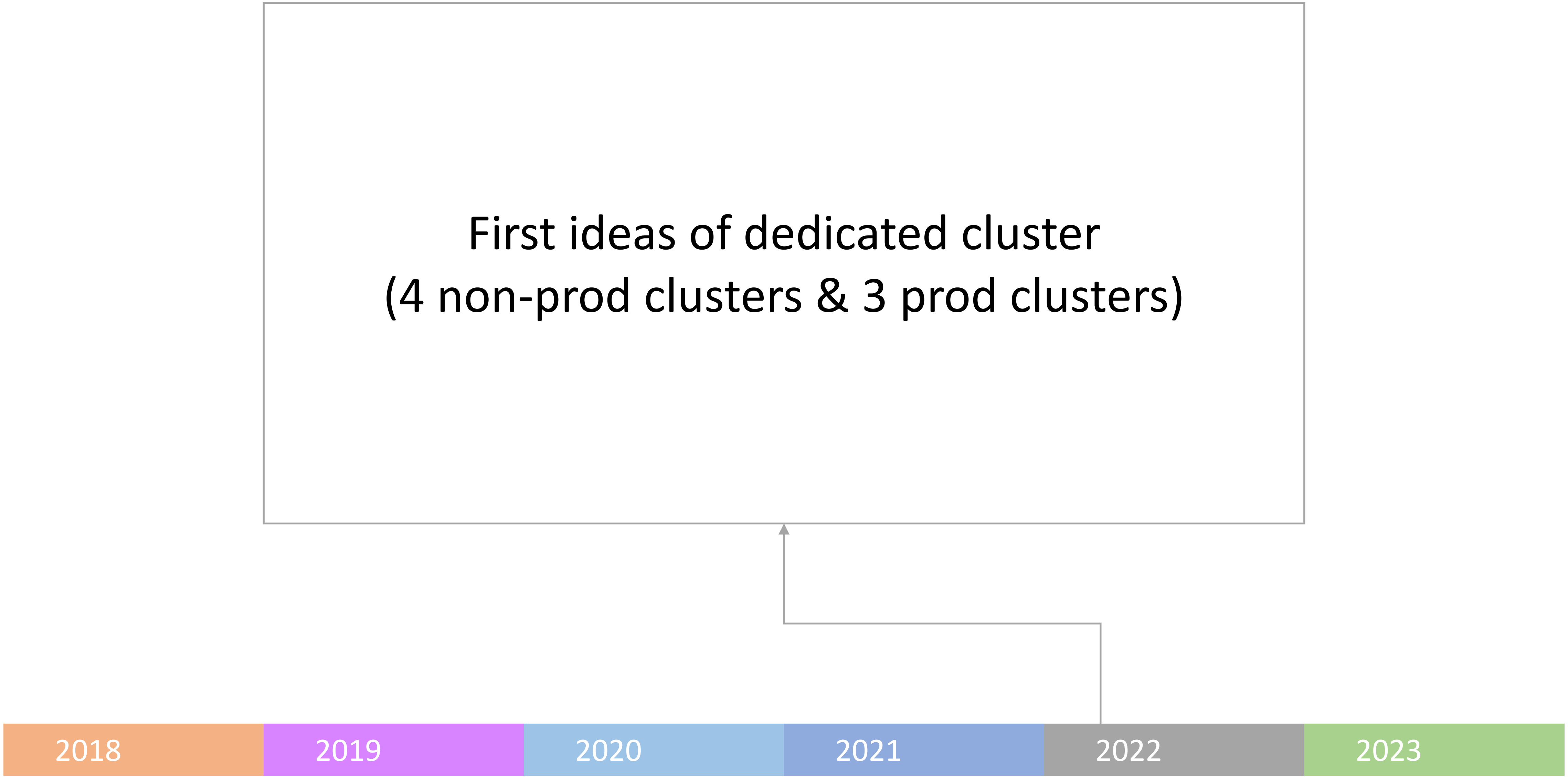
70 production nodes /
2.5k production containers

~200 projects in prod

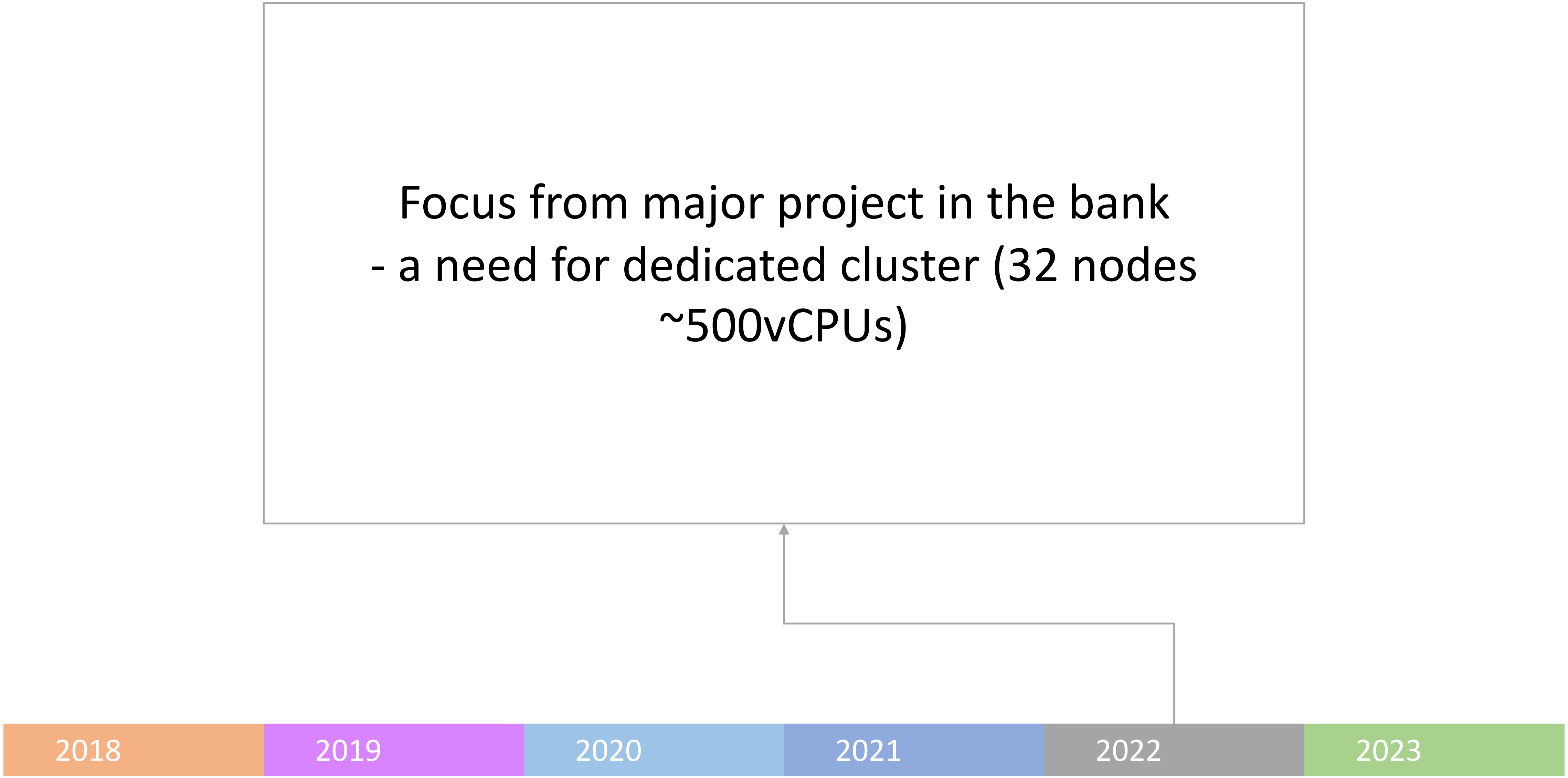


Another major issues non-production

- api-server slowness (irresponsive), no immediate impact on running applications (vendor provided fix)



First ideas of dedicated cluster
(4 non-prod clusters & 3 prod clusters)

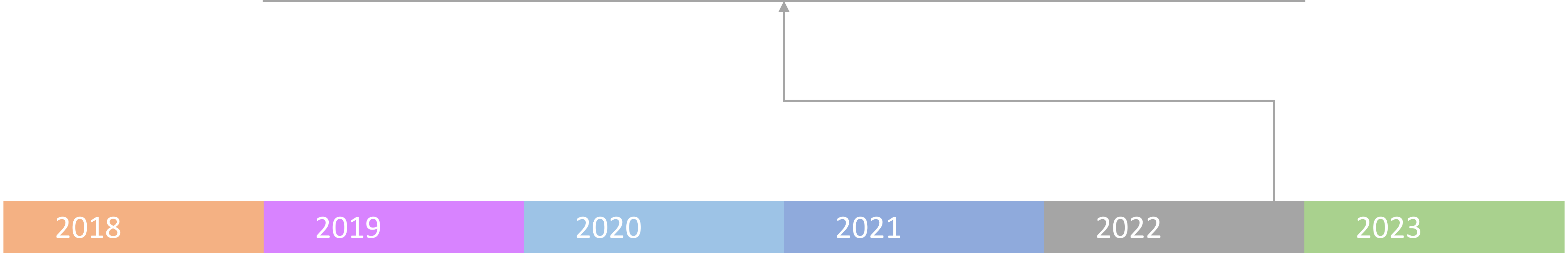


Focus from major project in the bank
- a need for dedicated cluster (32 nodes ~500vCPUs)

<500 non-production nodes /
18k non-production containers

180+ production nodes /
5k production containers

~500 projects (in pipeline)/18 clusters



2018

2019

2020

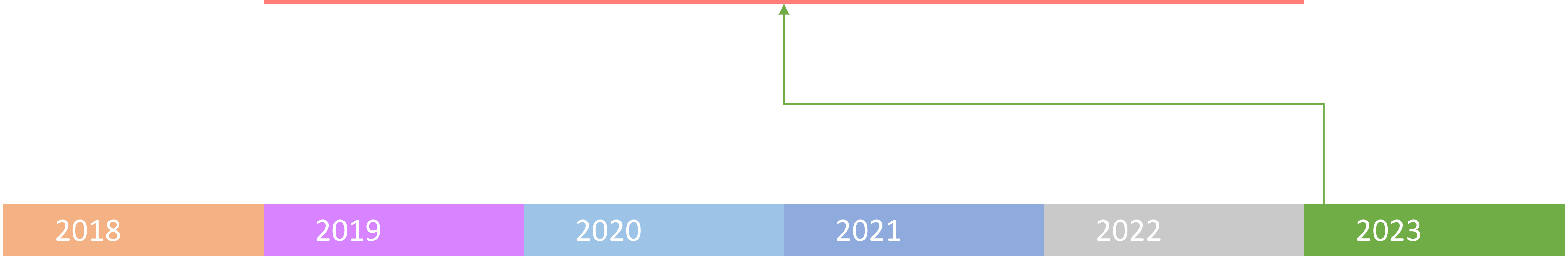
2021

2022

2023

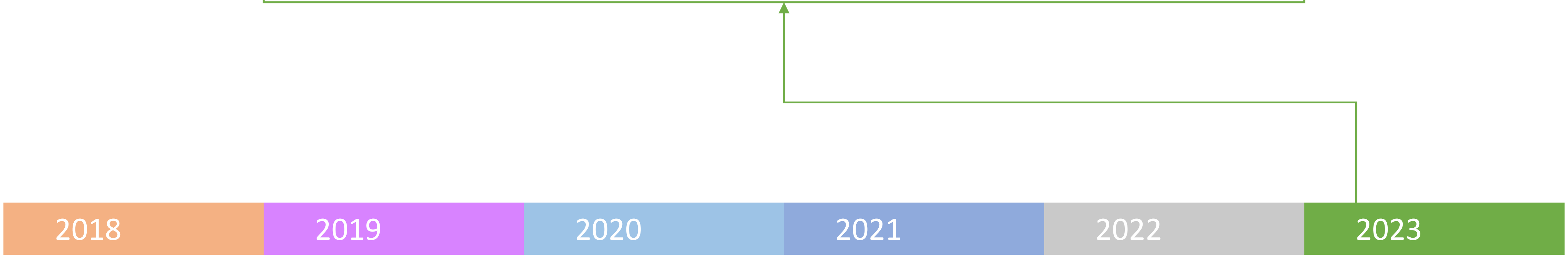
Yearly major issues saga continues

- An upgrade of the platform failed, backup taken just before was corrupted, …



2018 2019 2020 2021 2022 2023

And that is getting us into the present…



2018

2019

2020

2021

2022

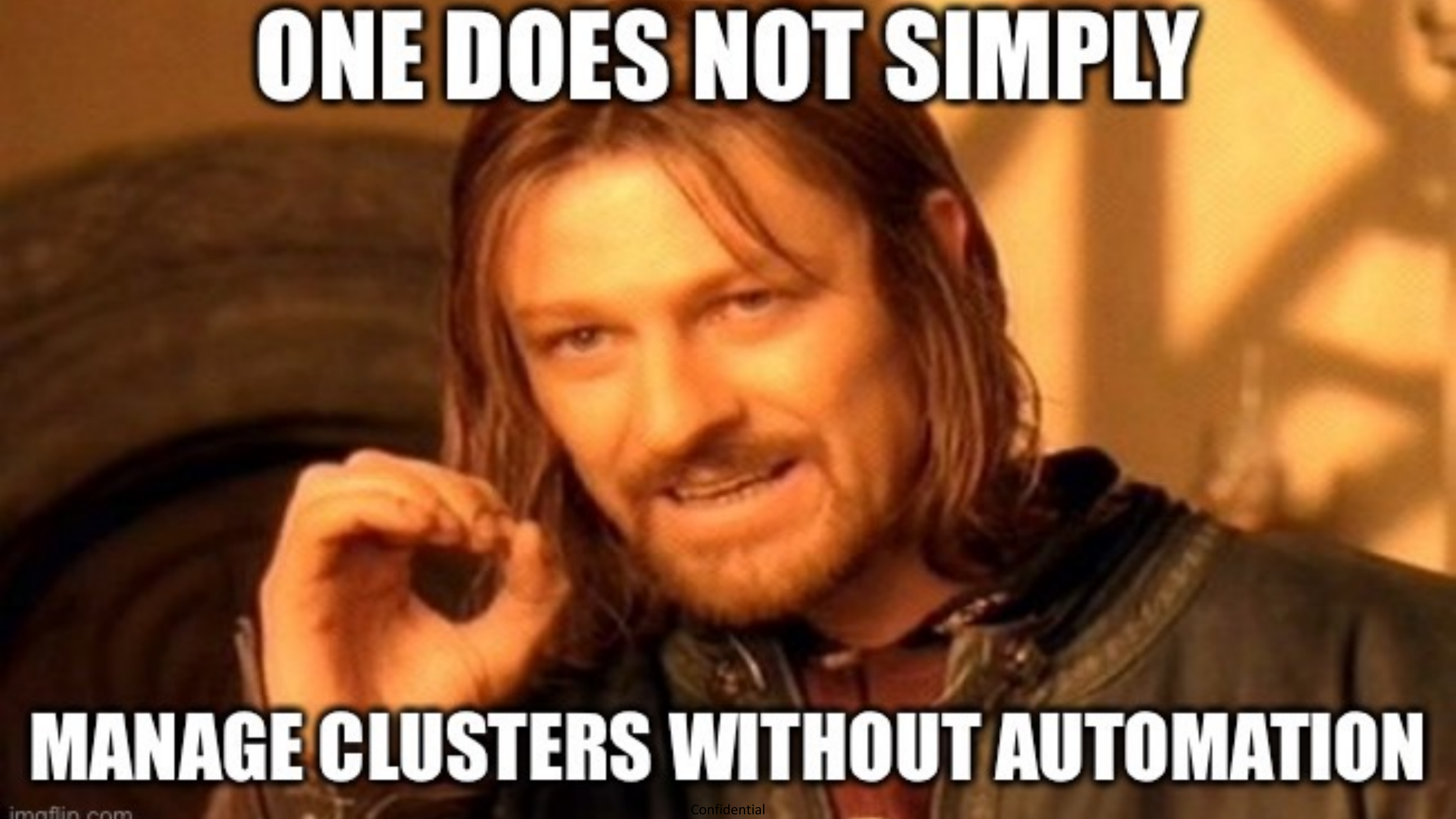
2023

Our motivation

As a platform engineering team responsible for the container platform, our primary goal is to ensure an efficient and reliable run of applications.

Our motivation

- Establishing a standard Container platform
- Automating the deployment process
- Implementing monitoring and alerting
- Ensuring security and compliance
- Providing training and support



ONE DOES NOT SIMPLY
MANAGE CLUSTERS WITHOUT AUTOMATION
imgflip.com

Next steps

- Splitting the largest non-production cluster into dedicated environment clusters (dev, test and pre-production).
- Clusters designed for purpose
- Utilize GitOps principles (we also like built-in feature in kubectl called Kustomize) to bootstrap our clusters
- Multi-cluster management, clusters’ lifecycle, infrastructure lifecycle

Thank you!

Any questions?