
KubeCon
CloudNativeCon
North America 2024

KubeCon

CloudNativeCon

North America 2024

Navigating the Future: Exploring the Latest in Kubernetes Dashboard Development

Marcin Maciaszczyk, Plural
Sebastian Florek, Plural

Introduction

Marcin Maciaszczyk
Fullstack Engineer at Plural

Working on the Kubernetes Dashboard project since the beginning. One of the key contributors and a SIG-UI co-leader.

Sebastian Florek
Fullstack Engineer at Plural

Working on the Kubernetes Dashboard project since the beginning. One of the key contributors and a SIG-UI co-leader.

Agenda

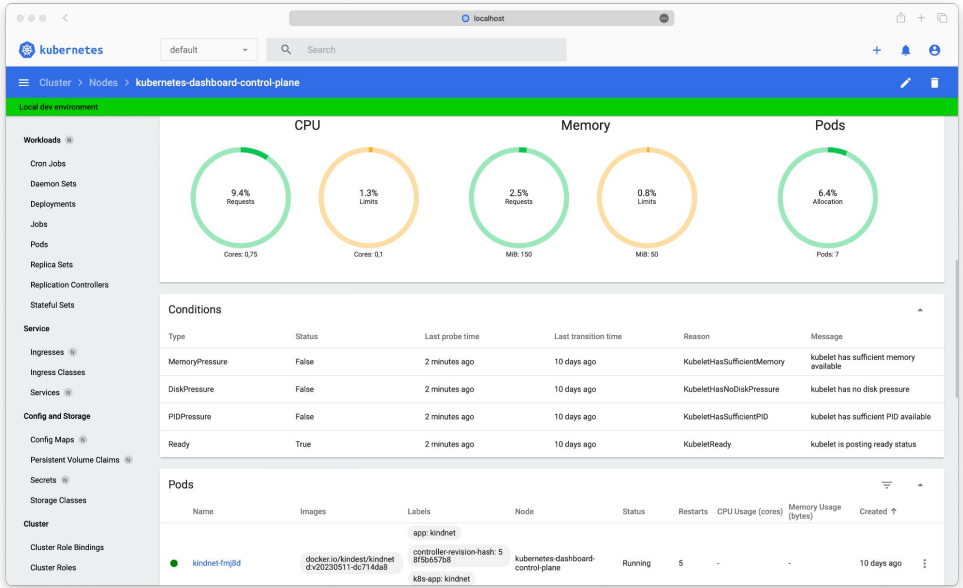
- Project Background
- Key Changes
  - New Architecture
  - Standalone API
  - Resource List Cache
- User Experience Enhancements
- Roadmap
- Q&A

Project Background

Kubernetes Dashboard

General purpose UI for Kubernetes clusters

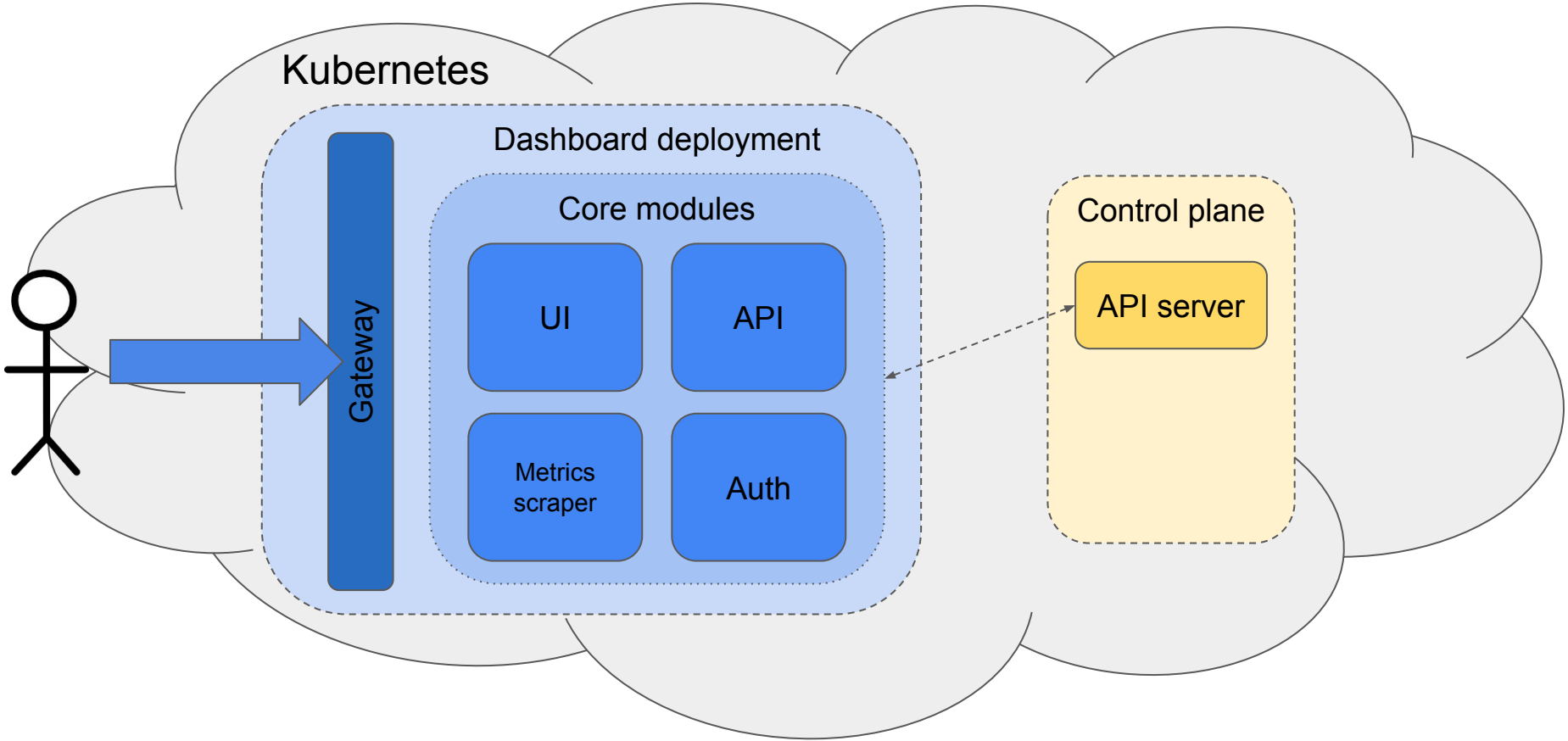
- First commit in 2015
- v7.9.0 released 22th October with support for Kubernetes v1.31
- ~8 millions image pulls monthly
- >1,1 billion total image pulls



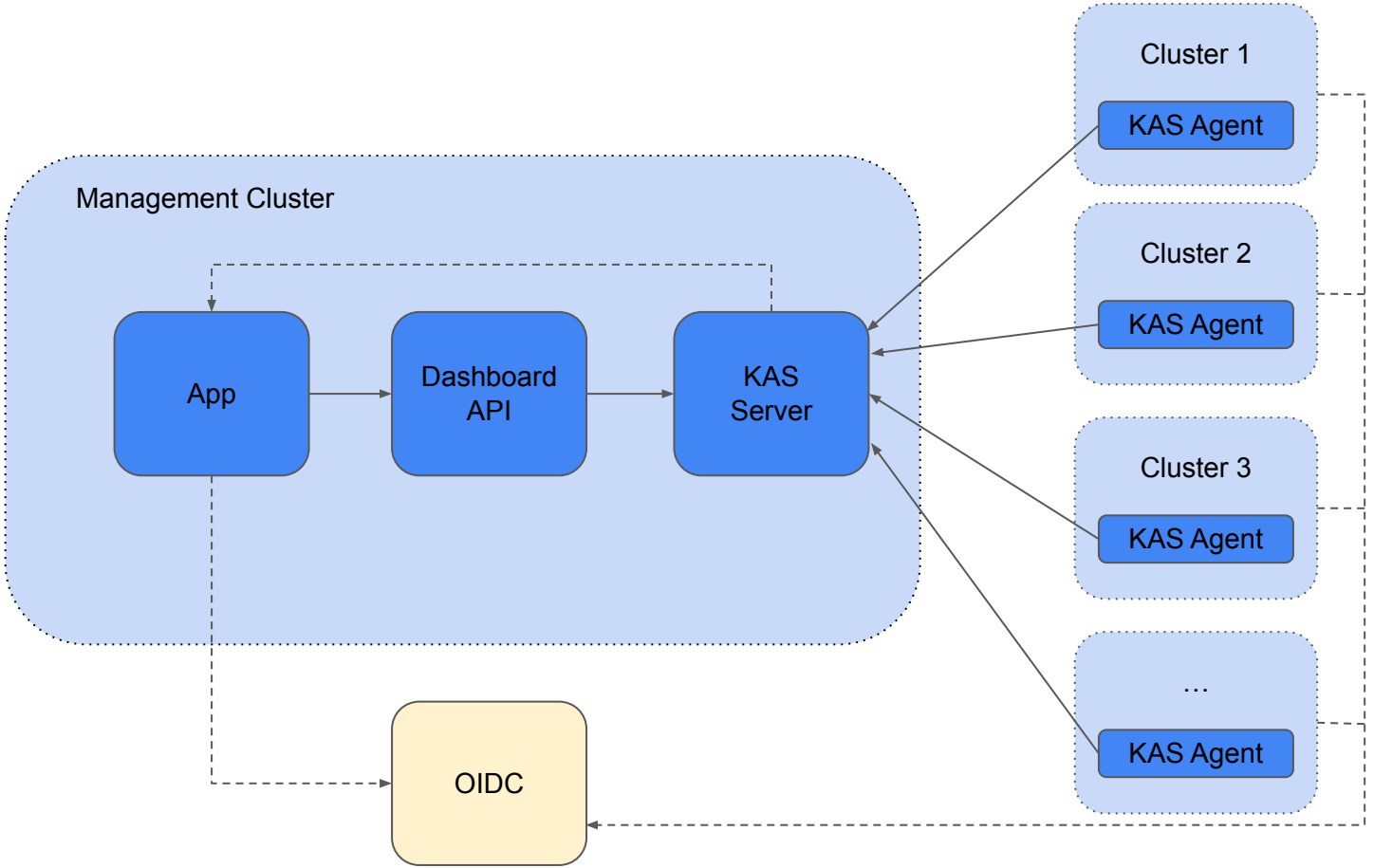
kubernetes/dashboard

#sig-ui

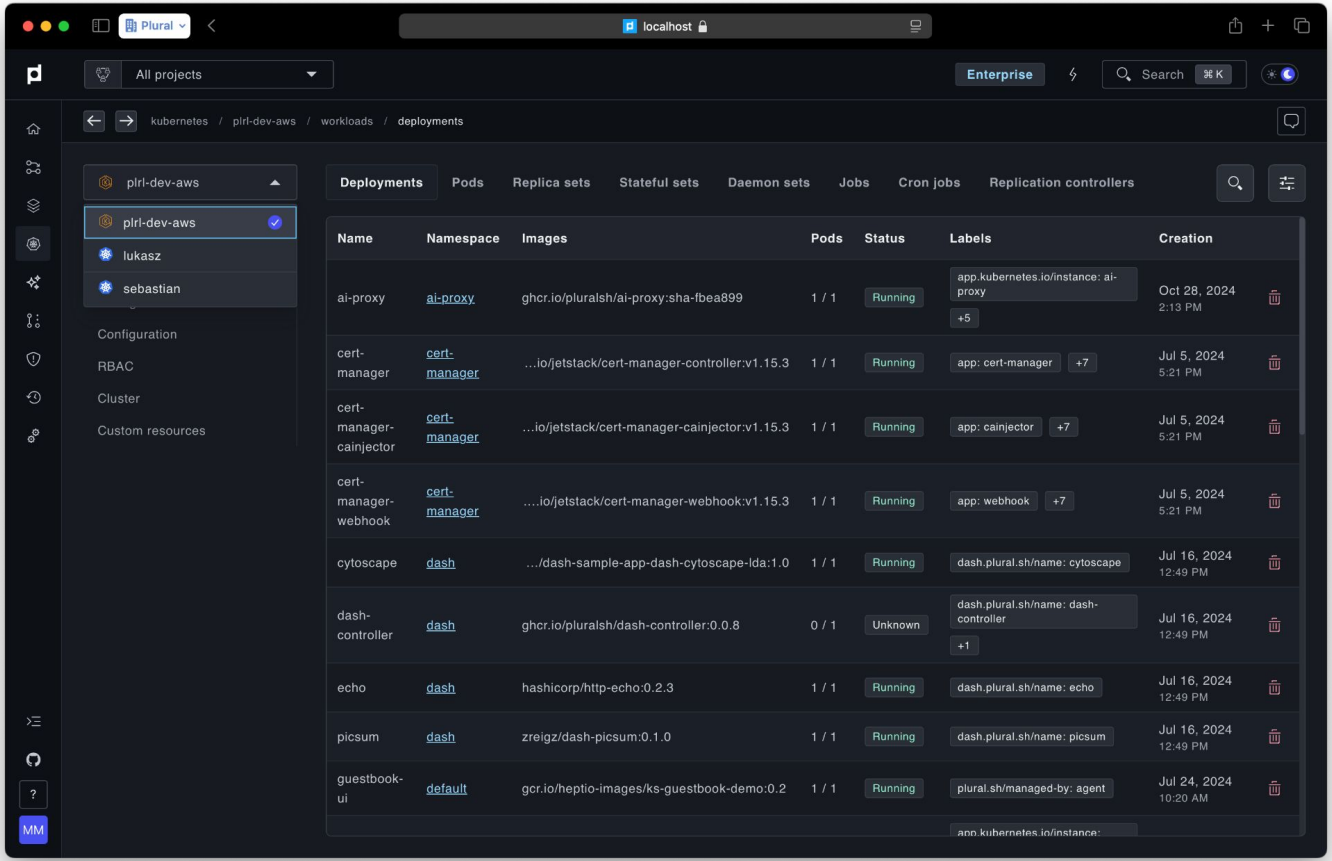
Key Changes - New Architecture



Key Changes - Standalone API



Key Changes - Standalone API



Key Changes - Resource List Cache

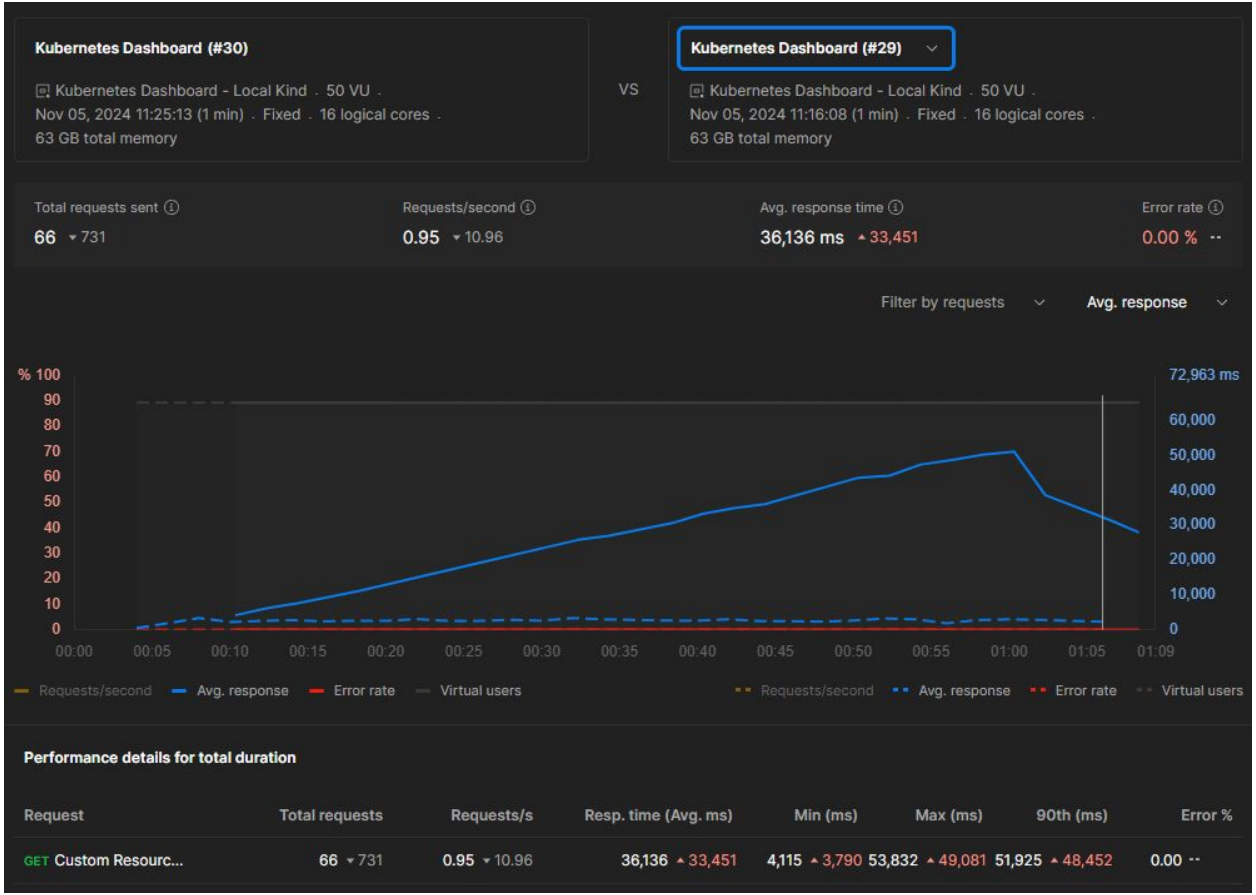
Benefits:

- Reduced latency - minimized time required to retrieve data from the Dashboard API during consecutive requests
- Enhanced user experience - smoother and more responsive experience
- Optimized resource utilization - less pressure on Kubernetes API Server, improved cluster performance
- Configurability and flexibility - cache can accommodate clusters of various sizes and complexities, it can run in multi-cluster mode as well. Users can opt-out and have same experience as before

Trade-offs:

- Using “cache-and-network” caching method can result in slightly stale data sometimes (affects only core resource lists)
- Increased average memory consumption but decreased peak memory usage

Key Changes - Resource List Cache



Test scenario:

- CRD List
- 50 virtual users
- Fixed request rate
- ~250 CRDs

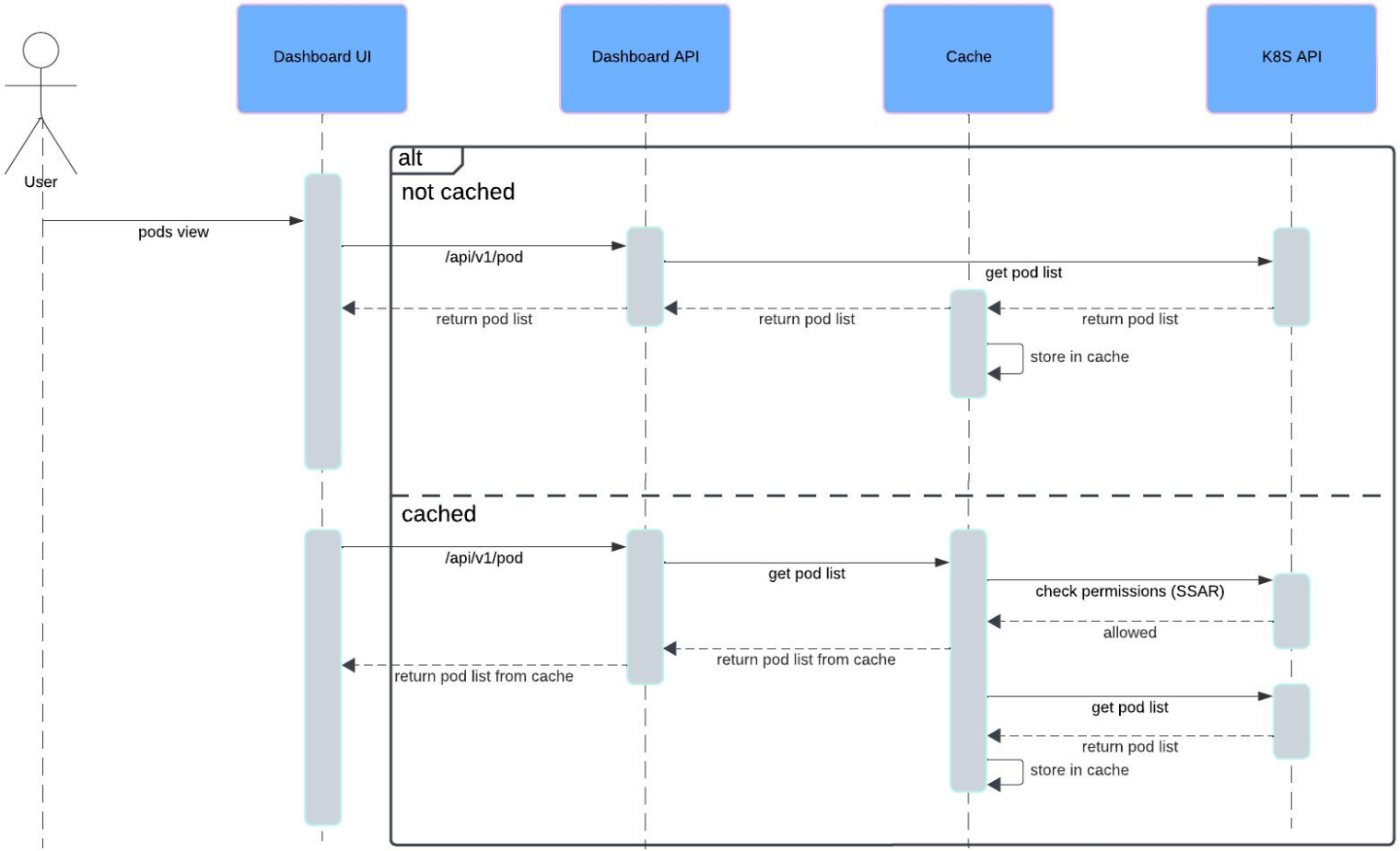
Without cache:

- Avg. response time: 36s
- Avg. requests/second: <1
- Total requests: 66

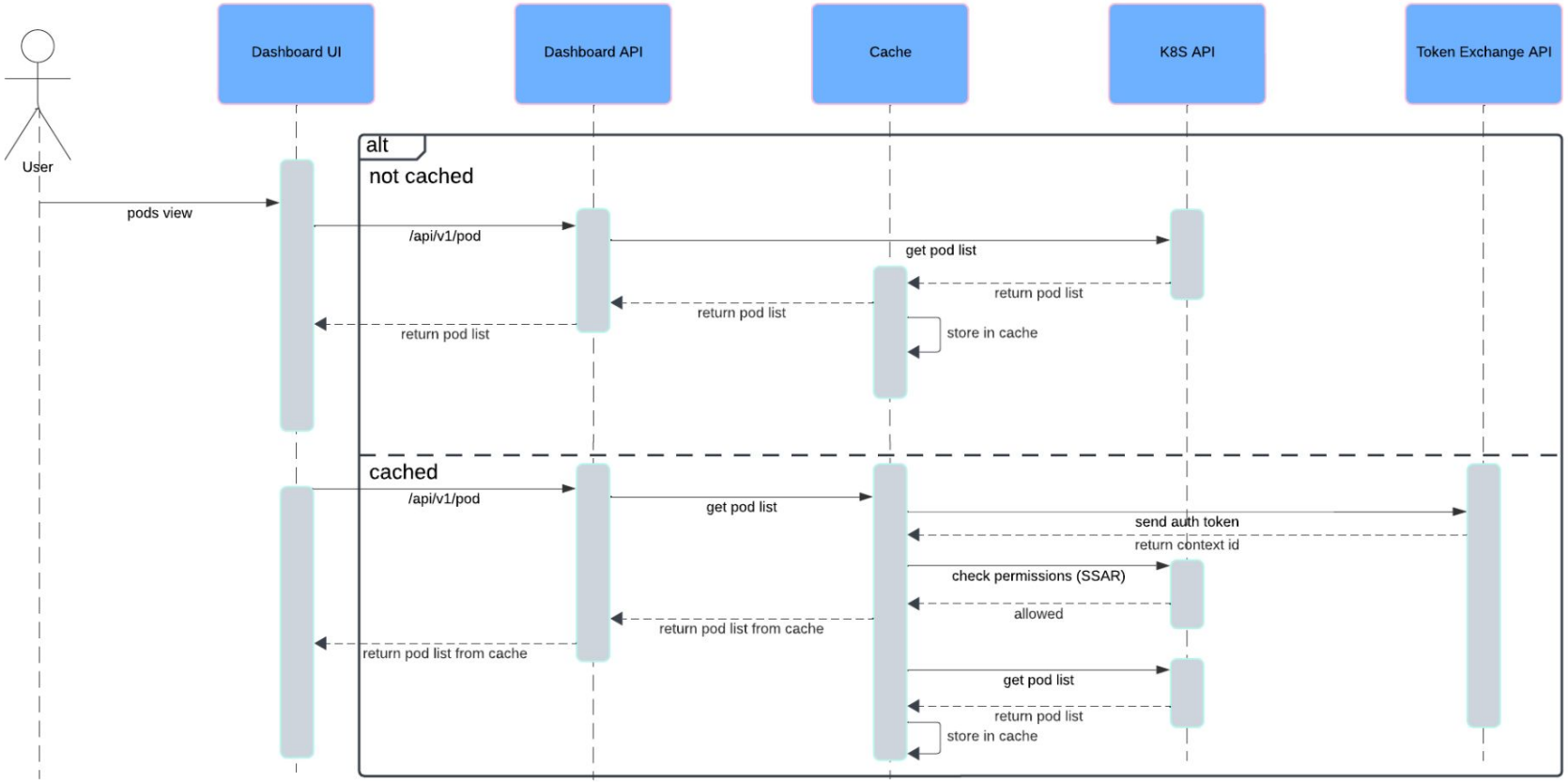
With cache:

- Avg. response time: 2.5s
- Avg. requests/second: >10
- Total requests: 797

Key Changes - Resource List Cache



Key Changes - Resource List Cache



User Experience Enhancements

Usability:

- Ingress paths are rendered as links
- About page is replaced by footer visible across the app
- Default resource auto refresh time interval was increased to 10 seconds
- Added username to user info panel
- Added support for delete propagation policy
- Added owner ref information to object meta
- Allowed hiding "All namespaces" in namespaces dropdown list

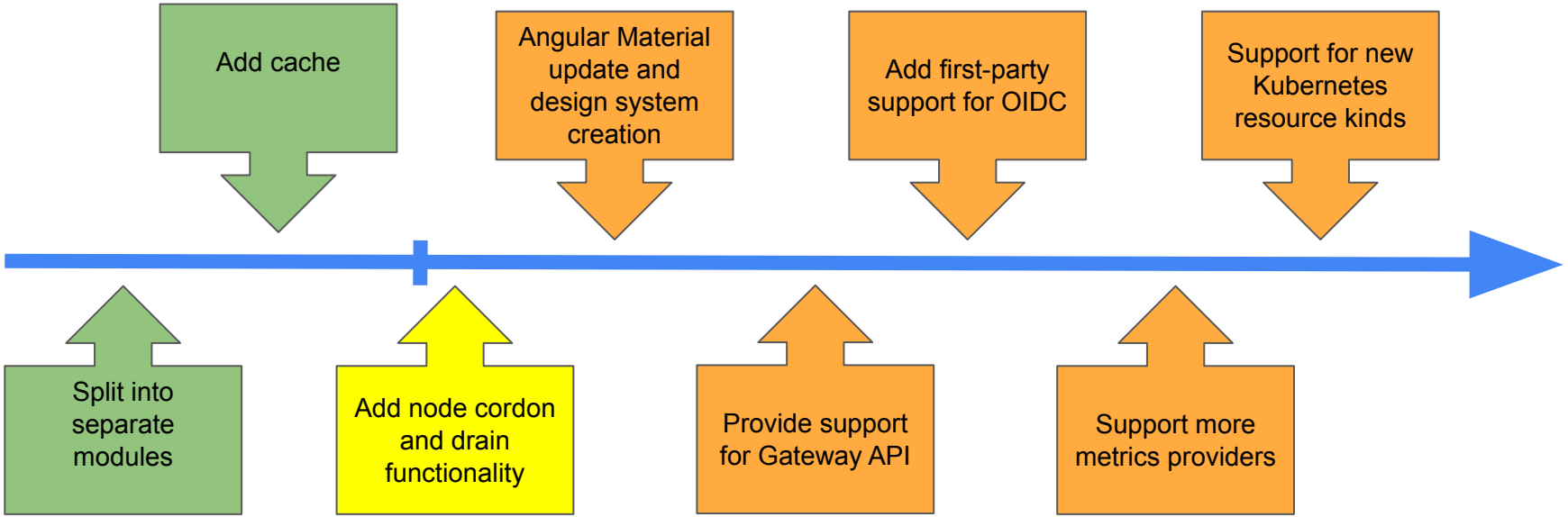
Security:

- Settings save relies on user permissions instead of Dashboard

Other:

- Improved pod container status logic to keep it in sync with kubectl
- Rewrite settings logic
- Added a small script to index.html to dynamically generate tag

Roadmap


Add cache
Angular Material update and design system creation
Add first-party support for OIDC
Support for new Kubernetes resource kinds
Split into separate modules
Add node cordon and drain functionality
Provide support for Gateway API
Support more metrics providers

Any questions?

The End

Thank you!