



CLOUD
NATIVE
CON
Europe 2017



KubeCon
A CNCF EVENT



Fission: Serverless Functions For Kubernetes

Soam Vasani, Platform9 Systems

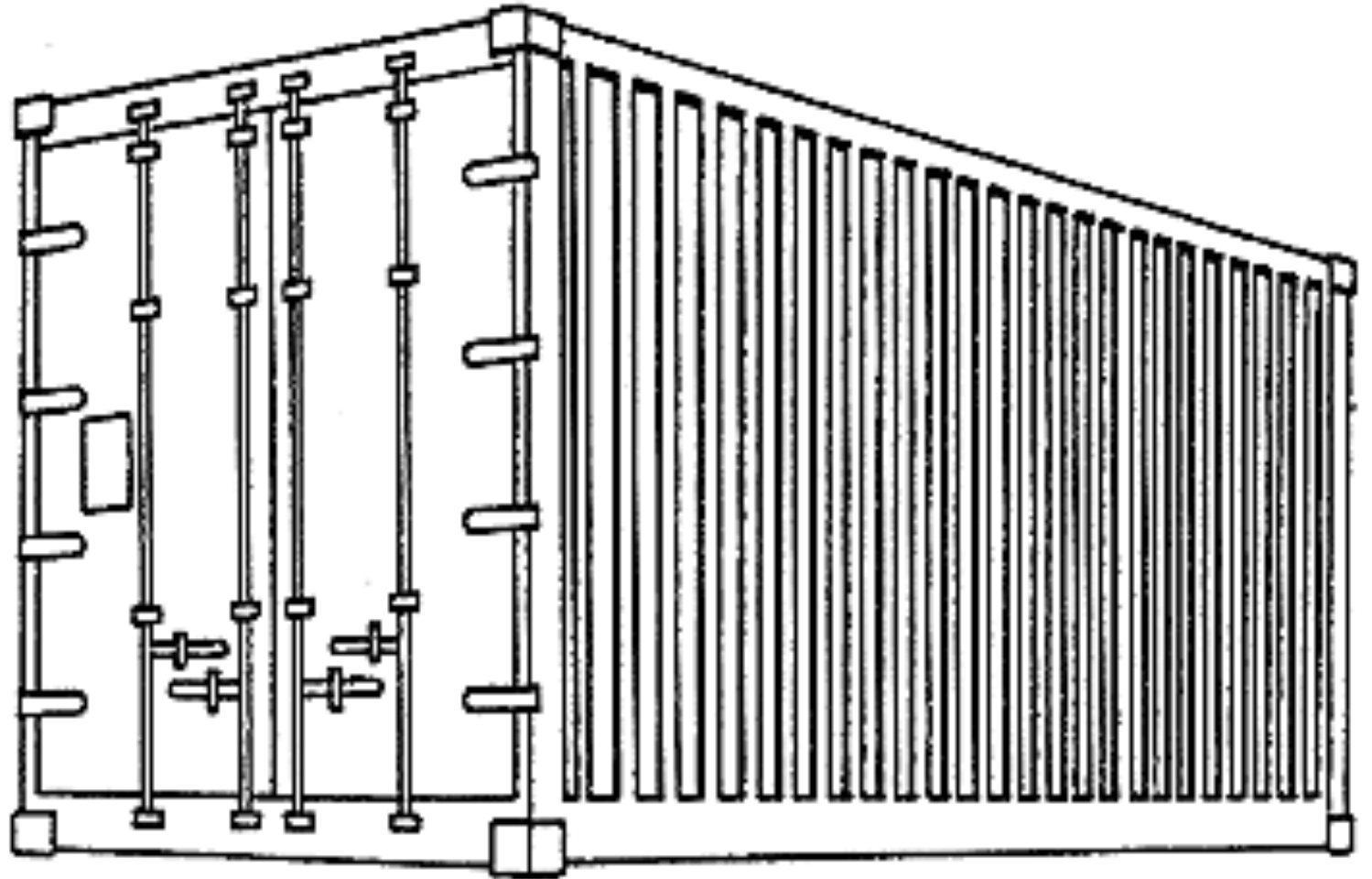
***Serverless*, adj.:**
Using someone else's Kubernetes cluster.

@ggreer on twitter

<https://twitter.com/ggreer/status/839171195920498688>

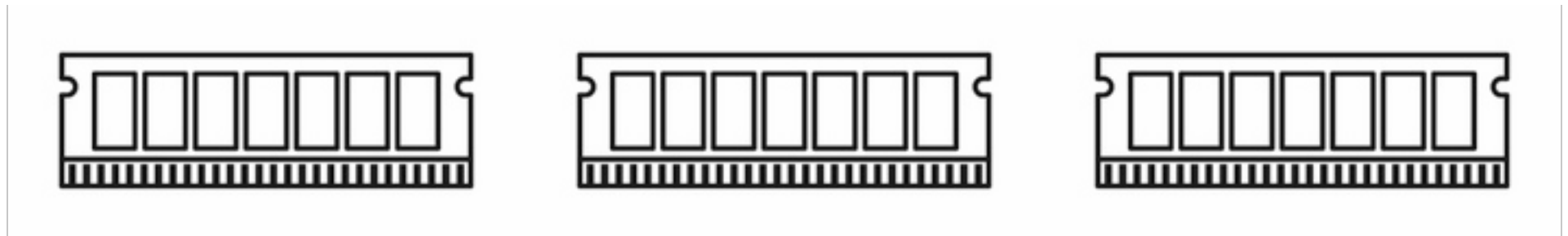
Containers

- Uniform
- But also: heavy!



Resource Allocation

- Rarely used services still need minimum resource alloc
- Cluster capacity as function of deployment size vs. actual usage



What if?

- What if we had the power of containers but very light dev workflows?
- What if we could have cluster capacity as a proportion of actual service usage?

Functions as a Service

- Short-lived stateless “functions”
- Source / function / module level
- Associated with an Event / HTTP / other trigger; activated on request only
- Fast on-demand start (“*cold start*”)

Fission: FaaS on Kubernetes

- Functions
- Environments: “Generic” containers
- Triggers

Fission: how it works

- Pool of “generic” containers for each environment
- Functions loaded on demand

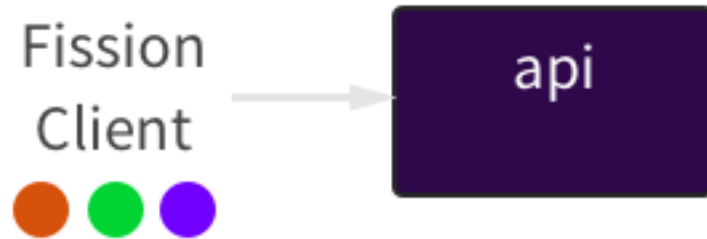
Fission
Client



api



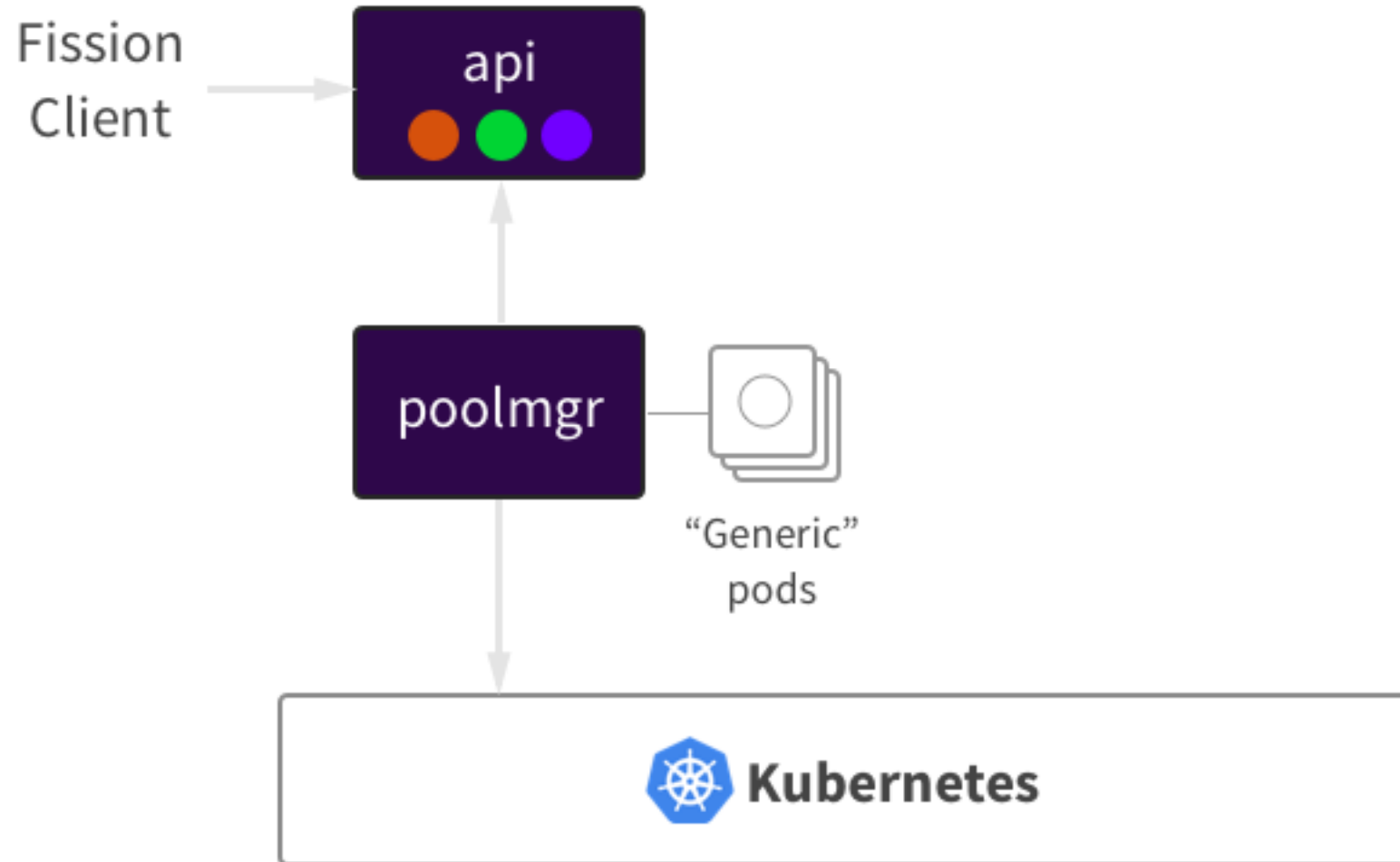
Kubernetes

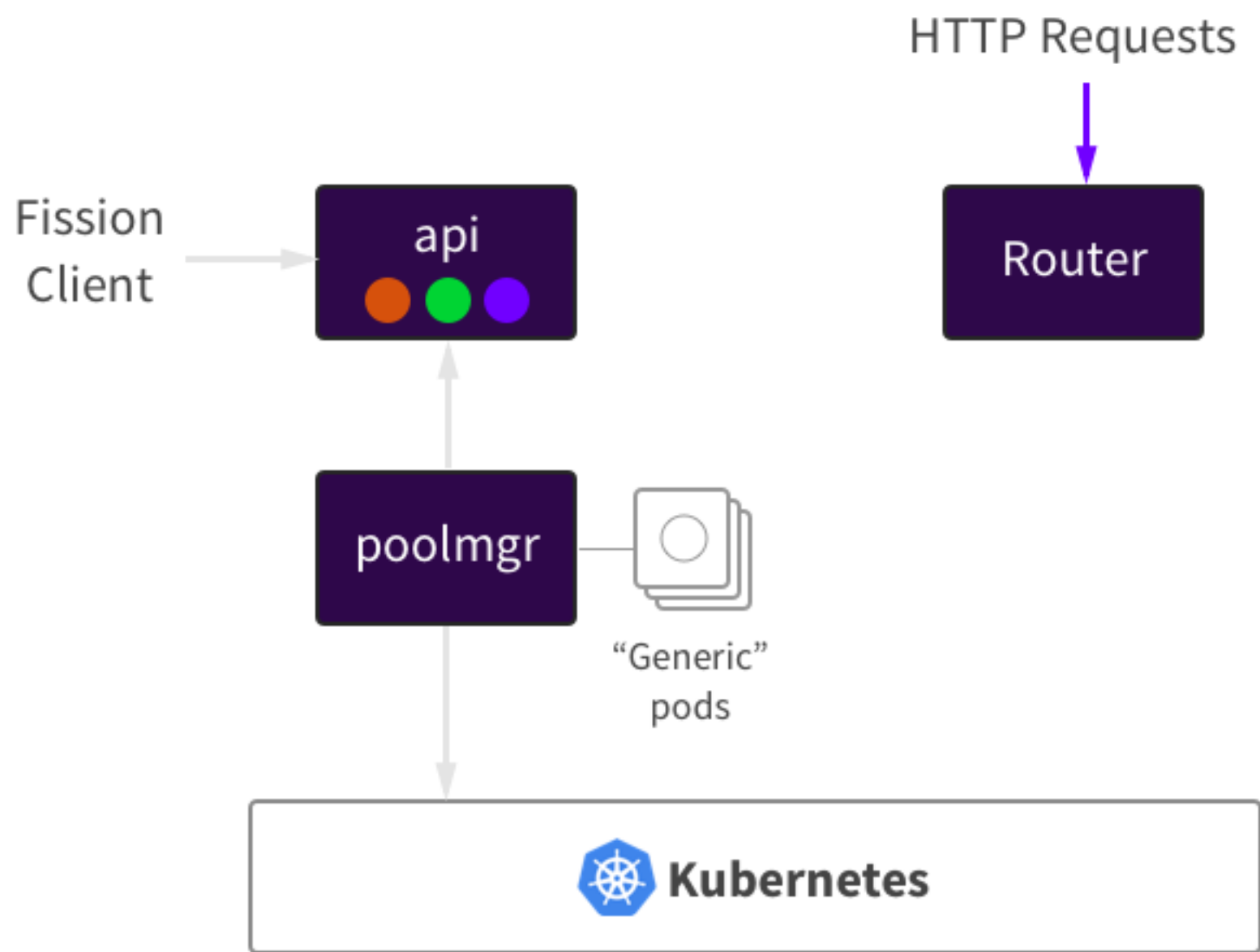


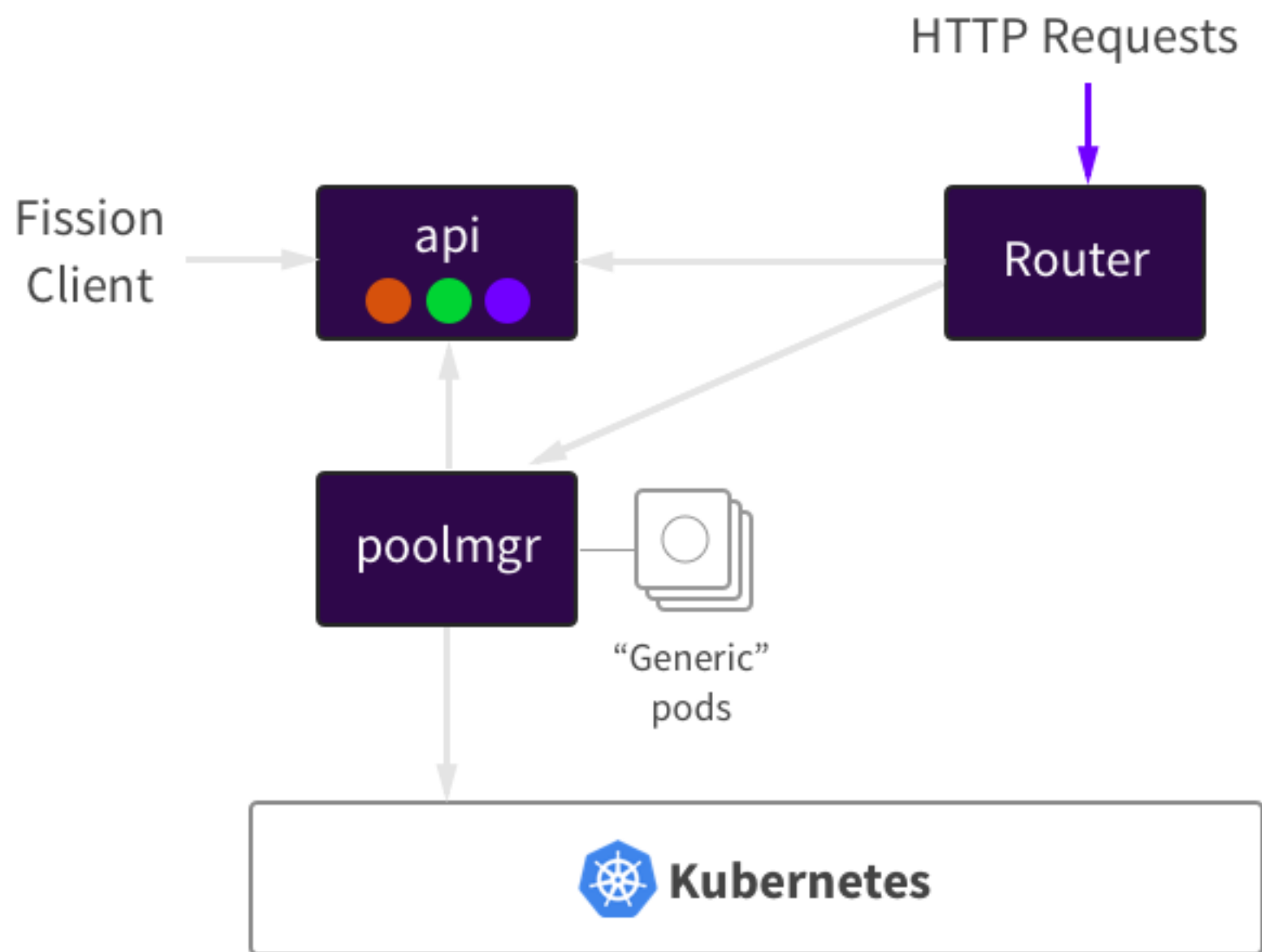
Kubernetes

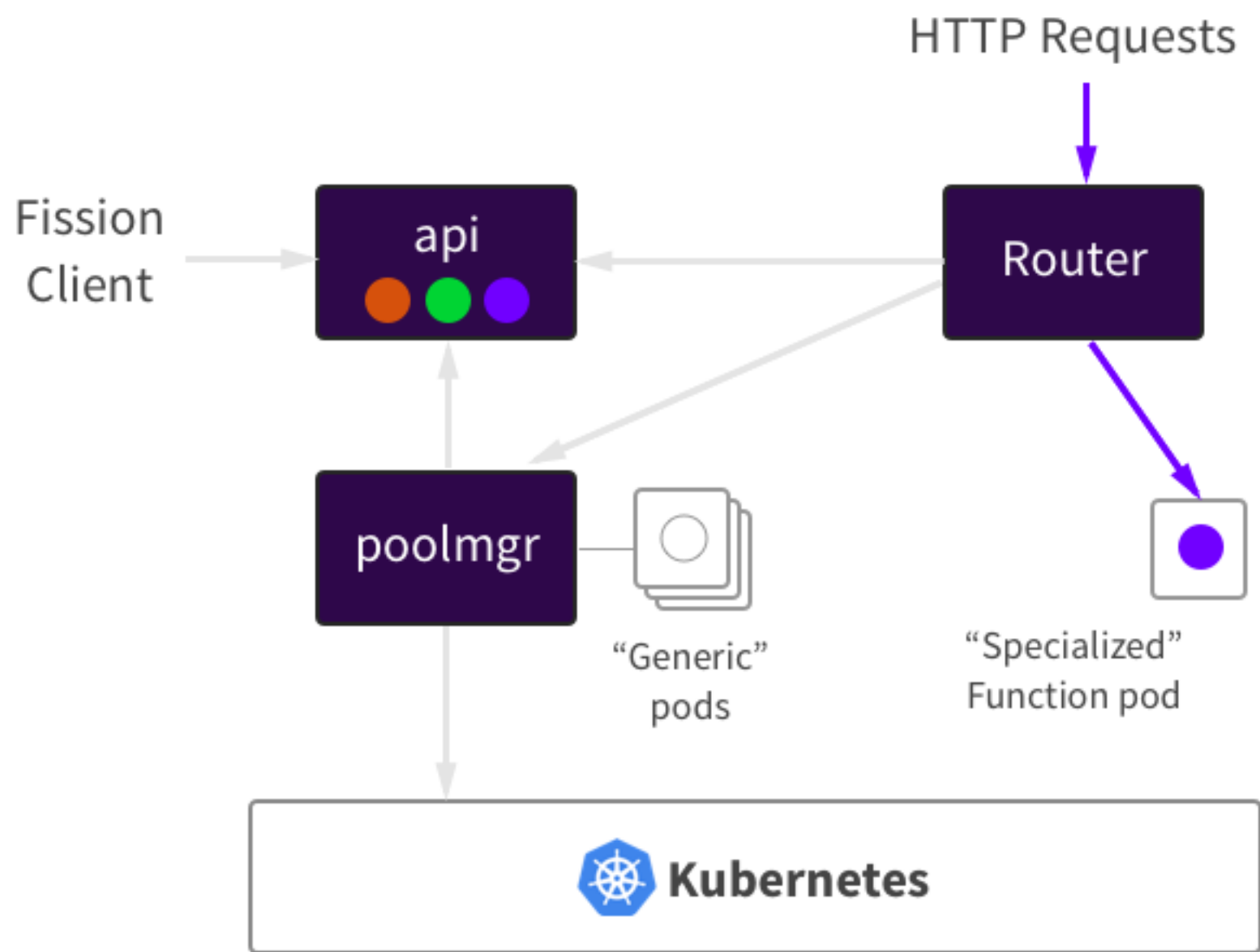


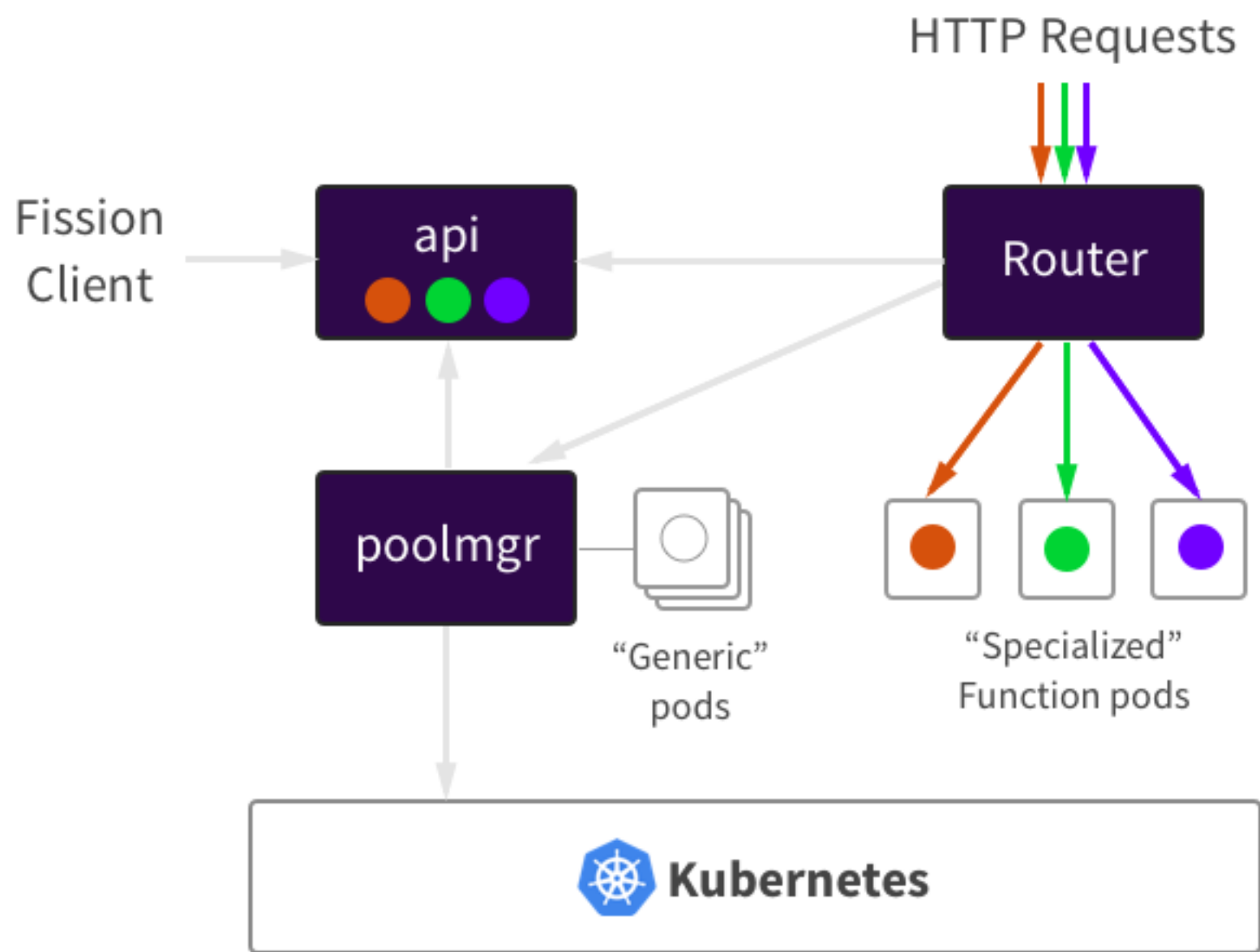
Kubernetes

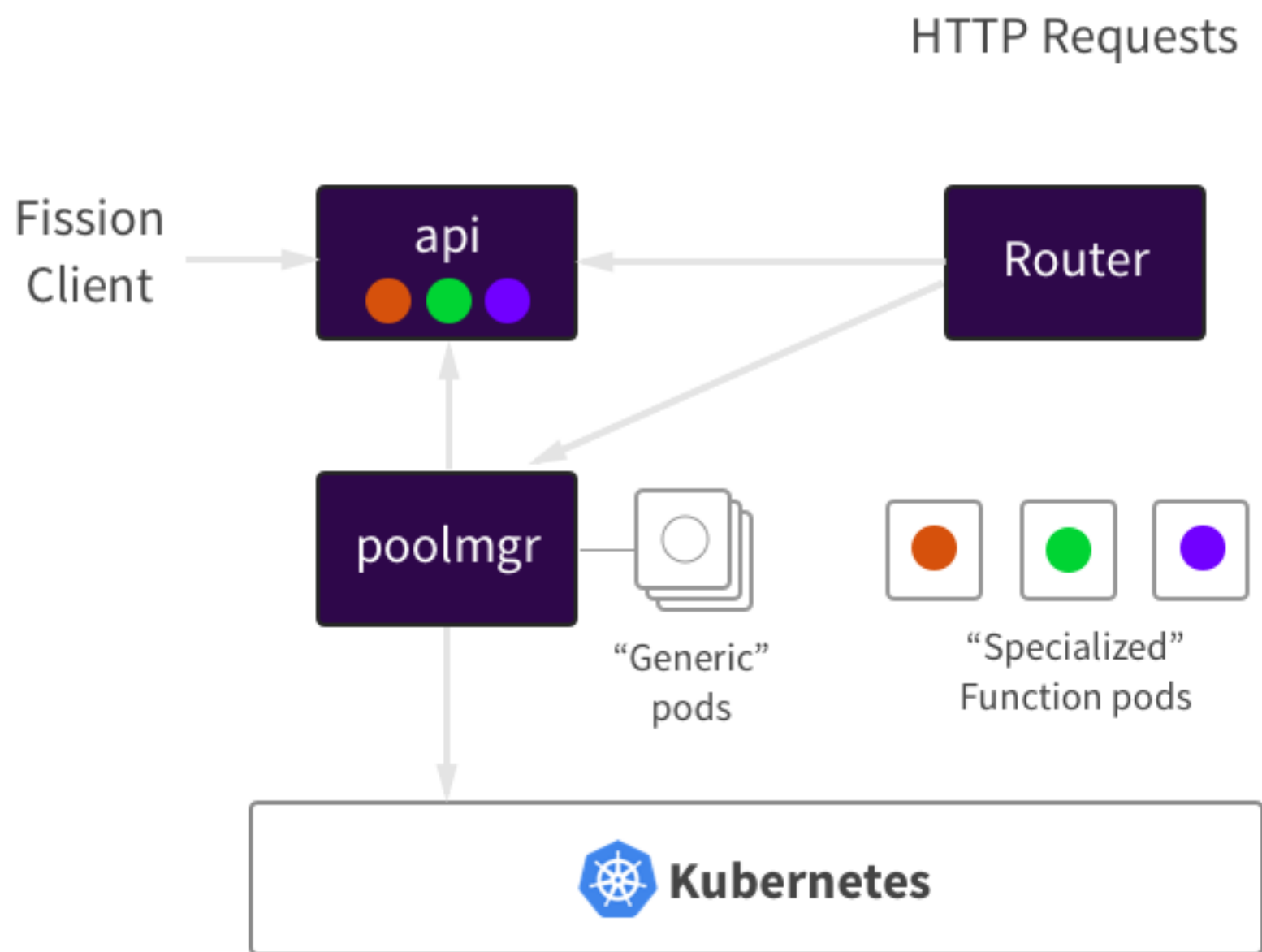


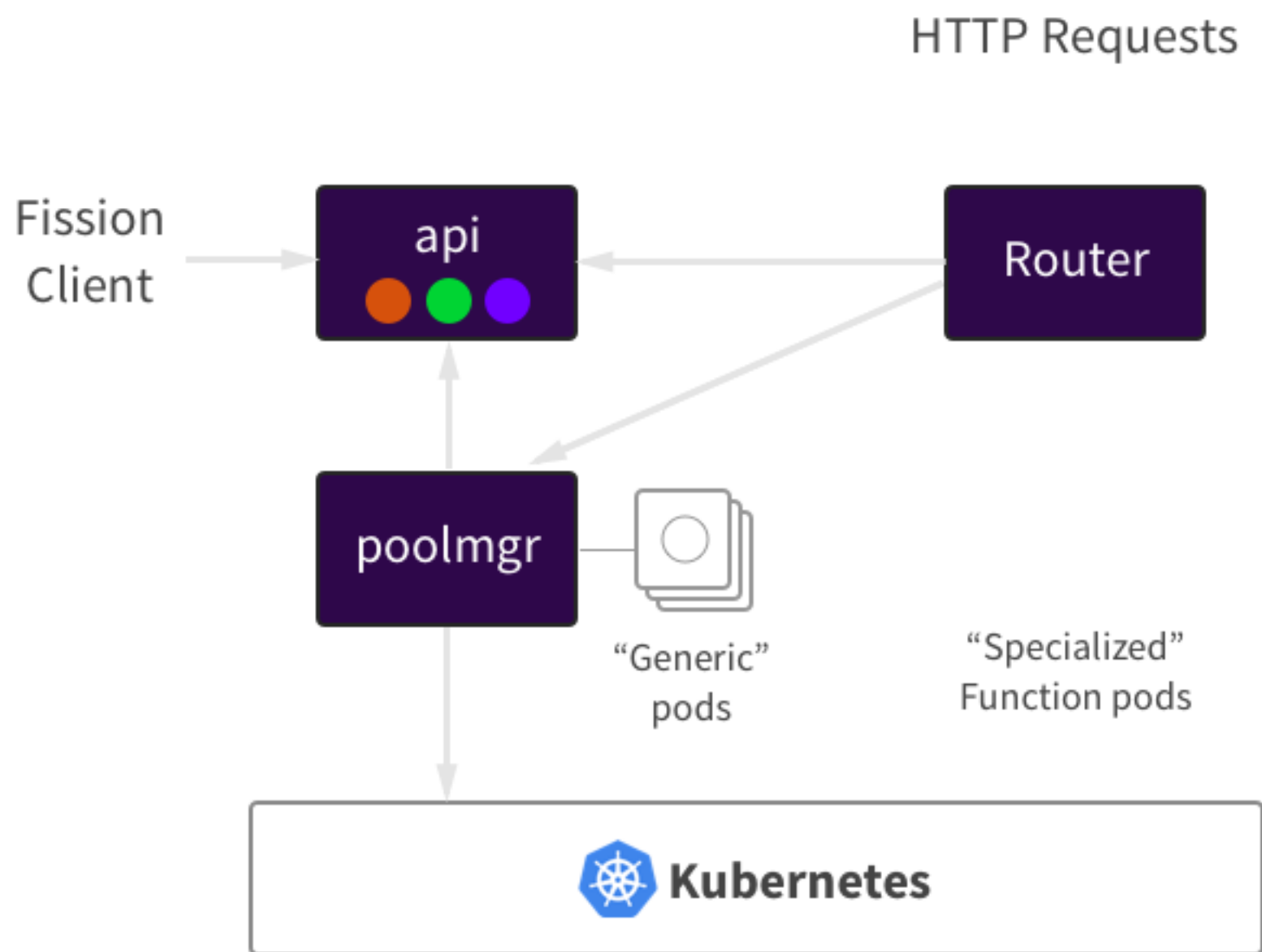












Use cases

- ~Small REST APIs
- Web hooks
- Kubernetes watch event handlers

Demos

Project Status

- Open sourced Nov 2016
- Currently alpha; beta mid-late this year
- Healthy community!
 - 1600 Github stars, 25 contributors, active Slack channel
 - Go, C#, PHP, Java support; Log aggregation/search, Web UI, many bug fixes

Roadmap

- More powerful environments (packages, compile step)
- Event queues
- Better Kubernetes API & ecosystem integration
- Observability: metrics, tracing, ...
- Kubernetes Volumes support
- Secrets, Config maps
- Unit testing
- Debugging
- Autoscaling

Thanks!

Github: <https://github.com/fission/fission>

Web: <http://fission.io>

Slack: <http://slack.fission.io>

Twitter:

@fissionio

@soamv

Roadmap — Environments v2

- Problems with single-file-loaded-at-runtime:
 - Multiple files, modules etc.
 - Compiled languages
 - Syntax errors in interpreted languages
- Source and deployment *packages*; separate storage service
- “Build” step — check syntax errors / gather deps / compile

Roadmap — API

- Use ThirdPartyResources for state
 - Version-controllable specs
 - No extra DB to manage
- Label-based route->function mappings (idiomatic Kubernetes)
- Ingress
- Hide TPRs and YAML files from users (as much as possible)

Roadmap — Ingress integration

- Create ingress resources for HTTP triggers (optionally)
- Allows fine-grained control over what routes are visible
- Allows integrating with richer API gateways (e.g. traefik)