



# Migrating the Openshift CI infrastructure into a microservice architecture

Michalis Kargakis  
Software Engineer, Red Hat

# Agenda

- Intro to Kubernetes and OpenShift
- The dark age
- State of the world
- Future work

# Intro to Kubernetes and OpenShift

- Kubernetes is a framework for running containers
  - Native support for orchestrating various workloads
  - Controller loop concept is fundamental
  - Extensible API
- and more ...

# Intro to Kubernetes and OpenShift

- OpenShift is an enterprise-ready distribution of Kubernetes
  - Developer-centric tools
  - Strict security profiles by default
  - Awesome UI
- and more (again) ...

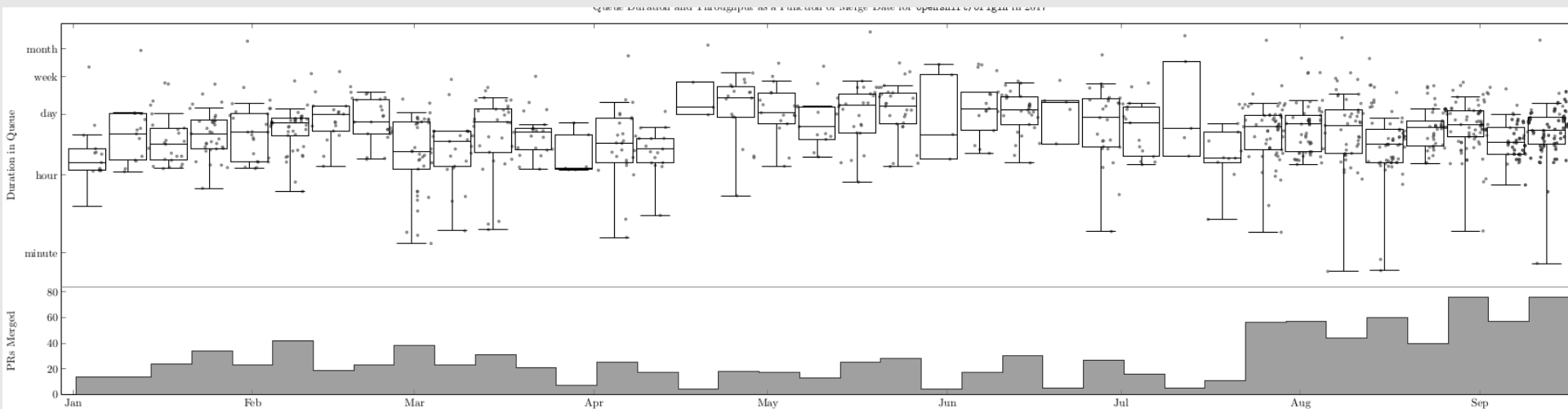
# The dark age

- Single bot written in Ruby responsible for testing and merging pull requests
- More pull requests and test flakes make the bot reach its limits
- A slow merge queue stomps developer productivity
- The project moves forward but the infra that empowers developing the project is not

# State of the world

- Replaced the old bot with goodies from kubernetes/test-infra: prow, mungegithub, gubernator
- prow is responsible for driving all tests to completion, soon it will also handle merging
- mungegithub does some stuff, including merging. Deprecated in favor of moving functionality in prow
- gubernator exposes test results and integrates with Github

# State of the world



Duration of a PR in the queue stays the same, merge throughput quadruples

# State of the world

- prow needs to run on top of a Kubernetes cluster
- Extends the Kubernetes API with ProwJobs and brings declarative management to your tests  
(one ProwJob == one test run)
- Comprised by a set of microservices that act as controller loops for ProwJobs
- Each service is responsible for a specific task



# State of the world

Two entrypoint services for creating tests in the cluster:

- **hook**, based on Github events
- **horologium**, based on configured periodic jobs

# State of the world



*hook creates PJs in  
Kubernetes in  
response to specific  
Github events, eg.  
creation of a PR*



**hook**

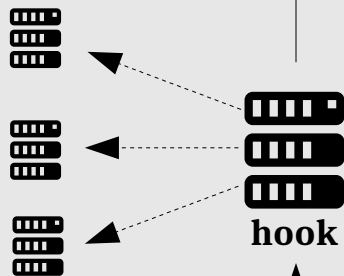
*hook responds to Github  
events based on a set of  
configured plugins*

*Github sends events  
via a webhook in hook*



# State of the world

*hook can demux  
Github events to  
other services -  
prow plugins*



# State of the world



*horologium lists all  
PJs in the system*

*horologium creates  
periodic PJs*



**horologium**



**hook**



**OPENSIFT™**

by Red Hat®

# State of the world

Two services for managing the lifecycle of tests:

- **plank** runs tests in Kubernetes pods
- **jenkins-operator** runs tests in Jenkins

# State of the world



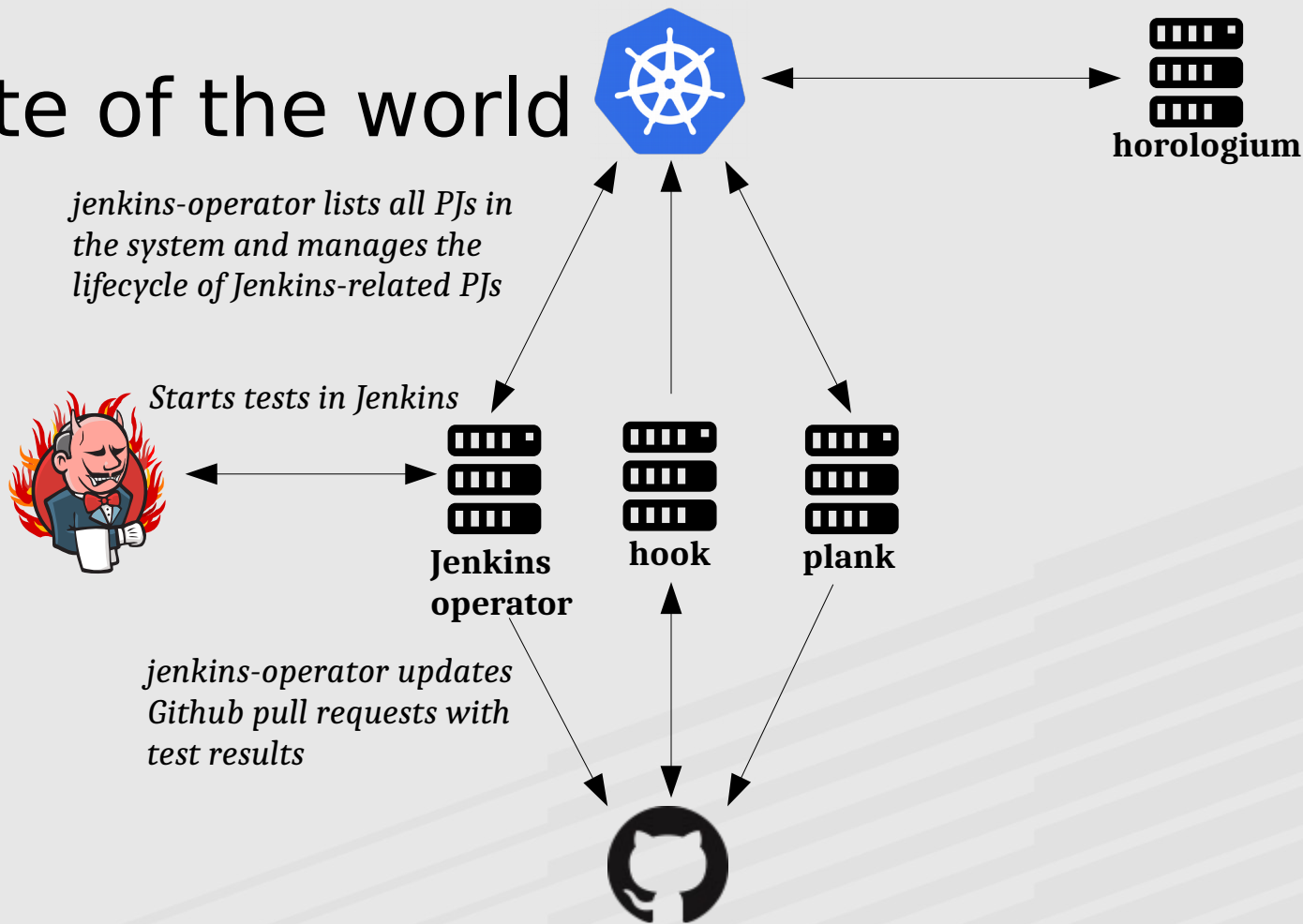
*plank lists all PJs in the system  
and manages the lifecycle  
of Kubernetes-related Pjs.  
It creates pods to run tests*



*plank updates Github  
pull requests with  
test results*



# State of the world



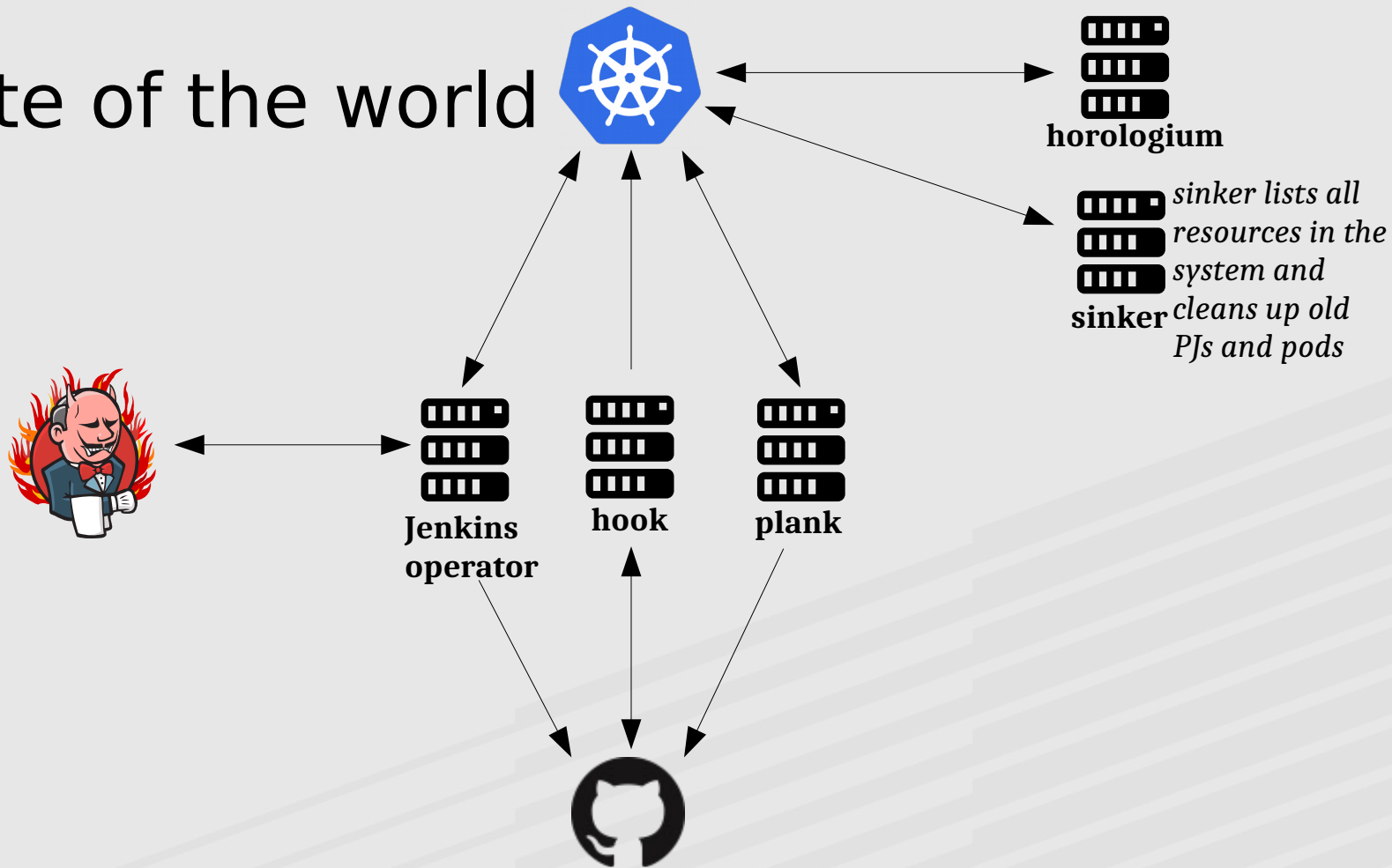
# State of the world

Others:

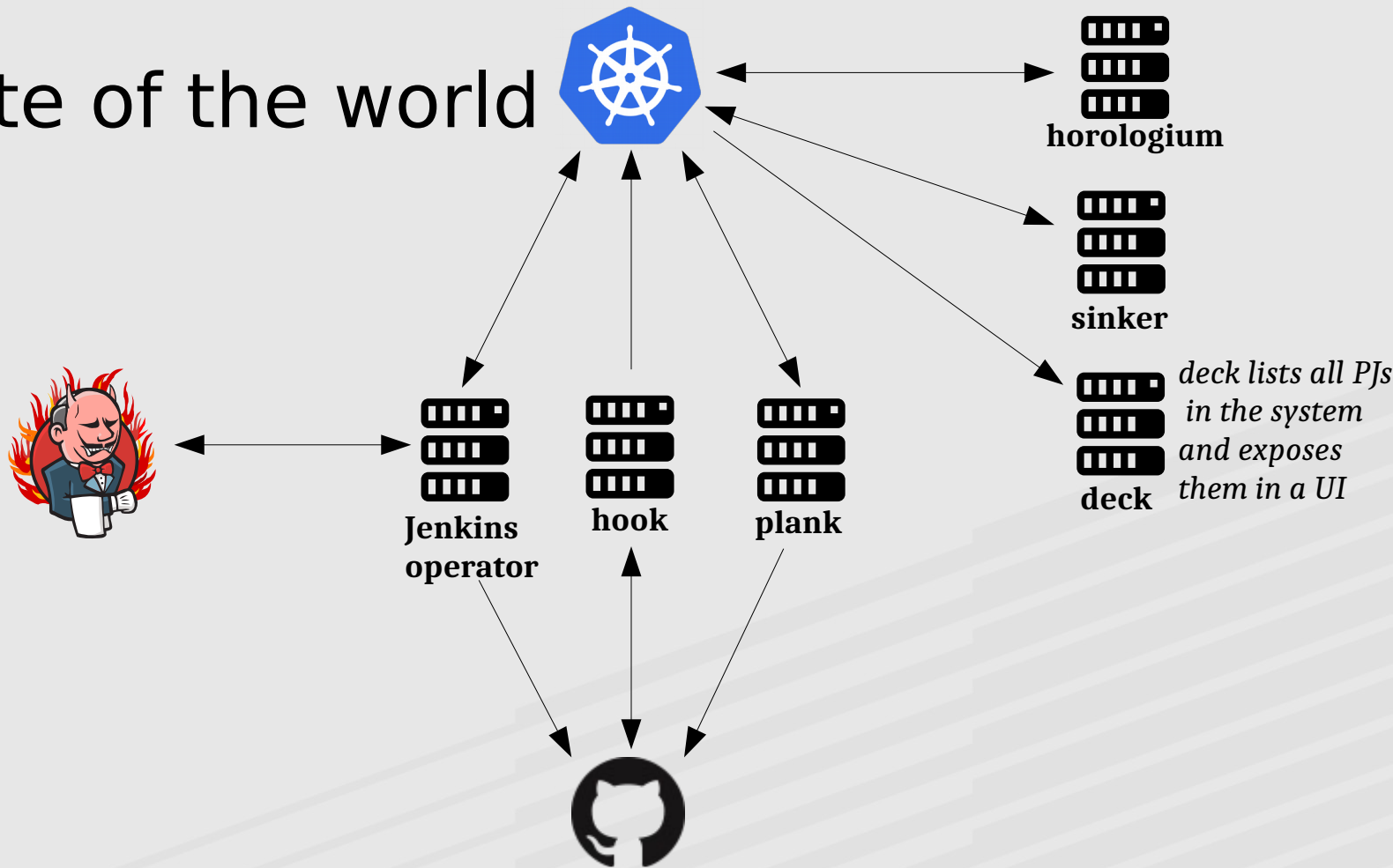
- **sinker** handles garbage collection of old PJs and pods
- **deck** is the front-end of prow
- **tide** handles merging and retesting pull requests



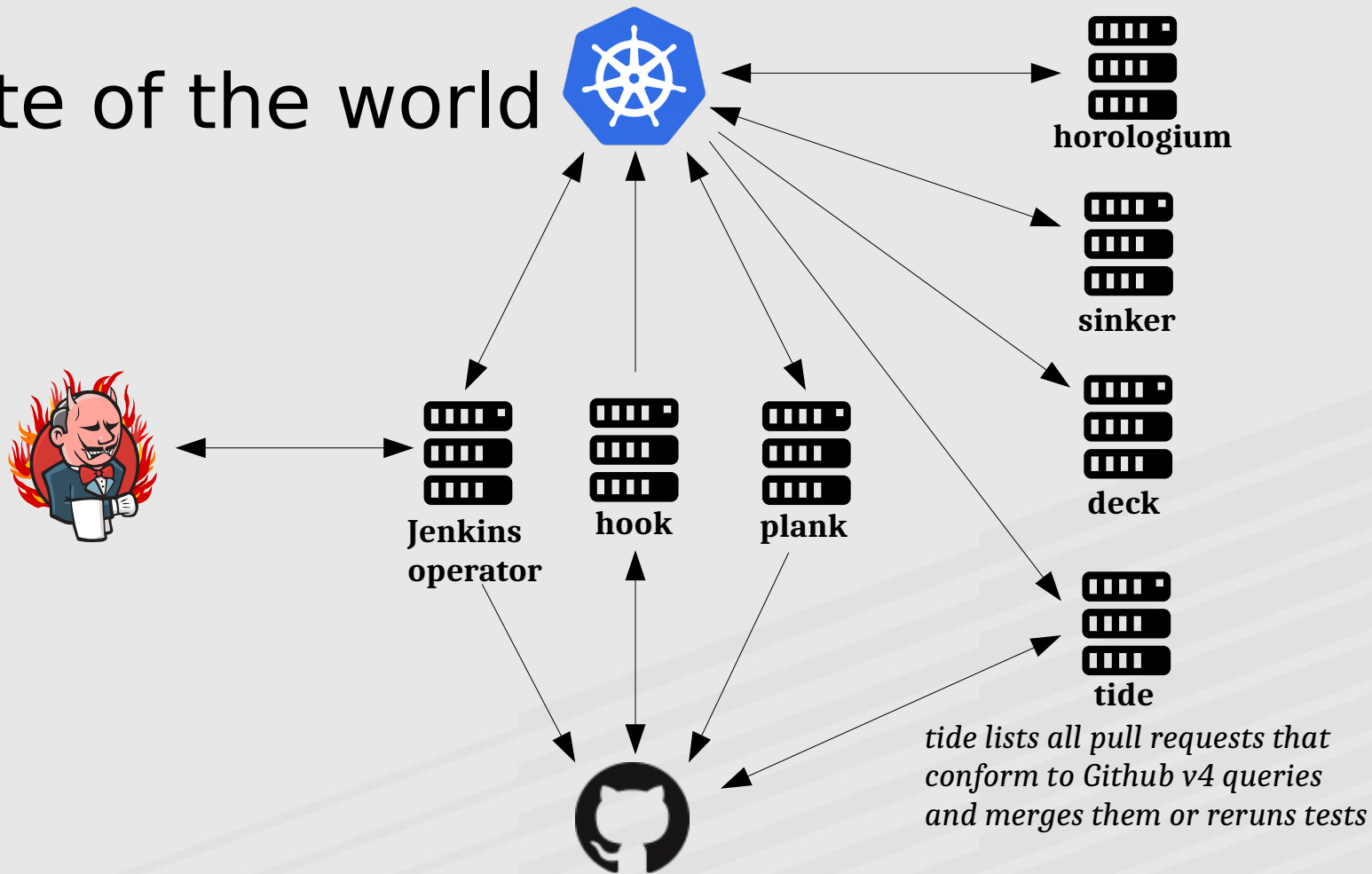
# State of the world



# State of the world



# State of the world



# Future work

- Full-feature parity between mungegithub and prow
- Expose metrics from prow
- Integration testing
- Canary deployments

# Fin

<https://github.com/kubernetes/test-infra/tree/master/prow>