Dogfooding Openshift with our CI infrastructure

Michalis Kargakis
@kargakis
Software Engineer, Red Hat

Agenda

- Problems (of the past)
- Goals
- Benefits (of the present)
- CI architecture
- Problems (of the present)
- Future work

Problems (of the past)

- “CI is slow”
- Tribal knowledge of our CI tools
- Nobody wants to maintain CI infra
- Non-extensible, no multirepo support

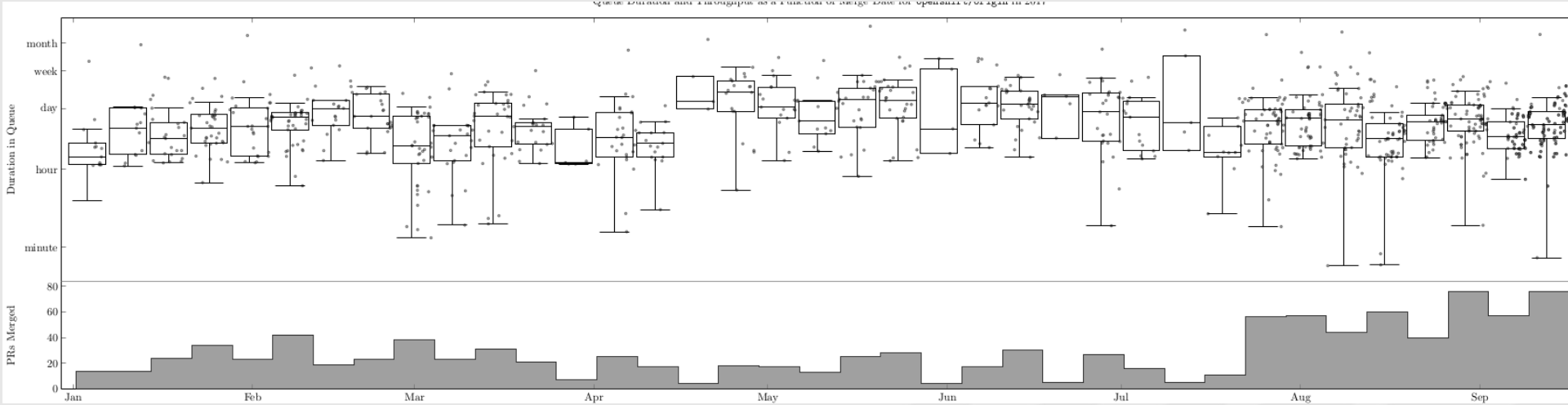
Goals

- Test and merge *efficiently*
- Support multiple repositories
- Make CI infra maintenance fun

Benefits (of the present)

- “CI is fast”
- Providing feedback to our development teams (dogfooding)
- Co-maintenance with the Kubernetes community
- CI infra shares code with core Kubernetes!
- Multirepo support, extensible, “quantum” CI

Benefits (of the present)



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

CI architecture

- Replaced the old bot with goodies from k8s/test-infra
- **prow** is responsible for testing and merging PRs and all the user interactions in between
- **Jenkins** is still in the picture (for now)
- Test results/artifacts are pushed by Jenkins/prow into GCS buckets
- **gubernator** exposes test results/artifacts

CI architecture

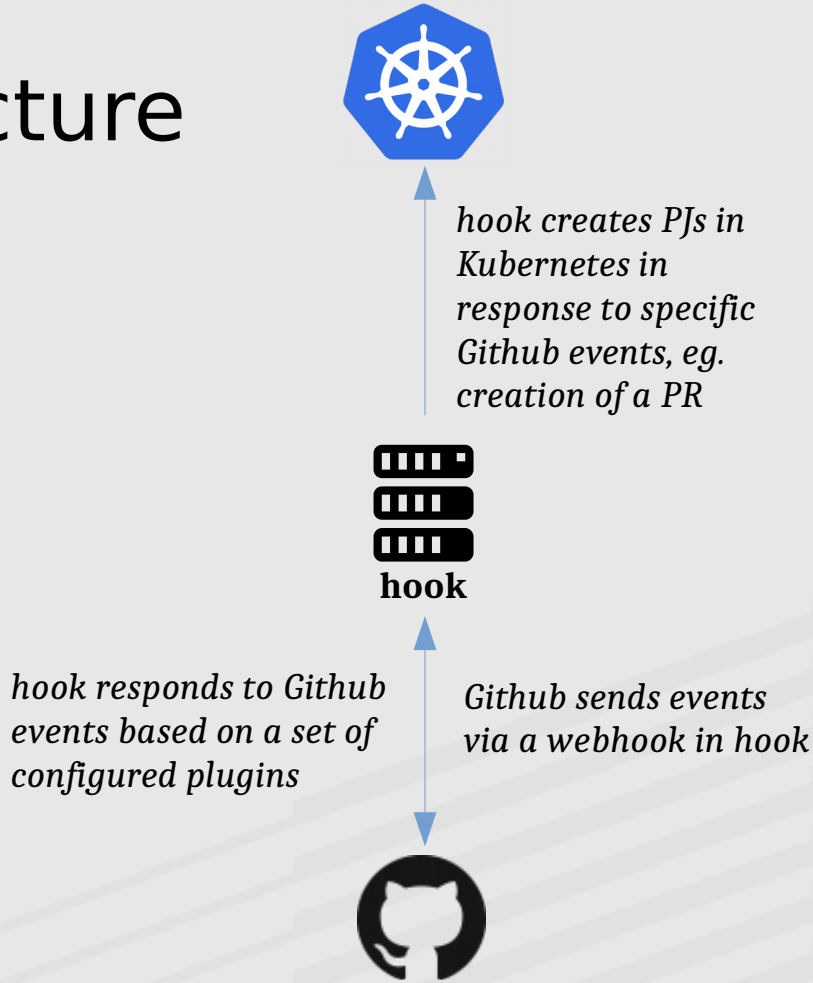
- prow needs to run on top of a Kubernetes cluster
- Extends the Kubernetes API with ProwJobs
- Comprised by a set of microservices that act as controller loops for ProwJobs
- Each service is responsible for a specific task

CI architecture

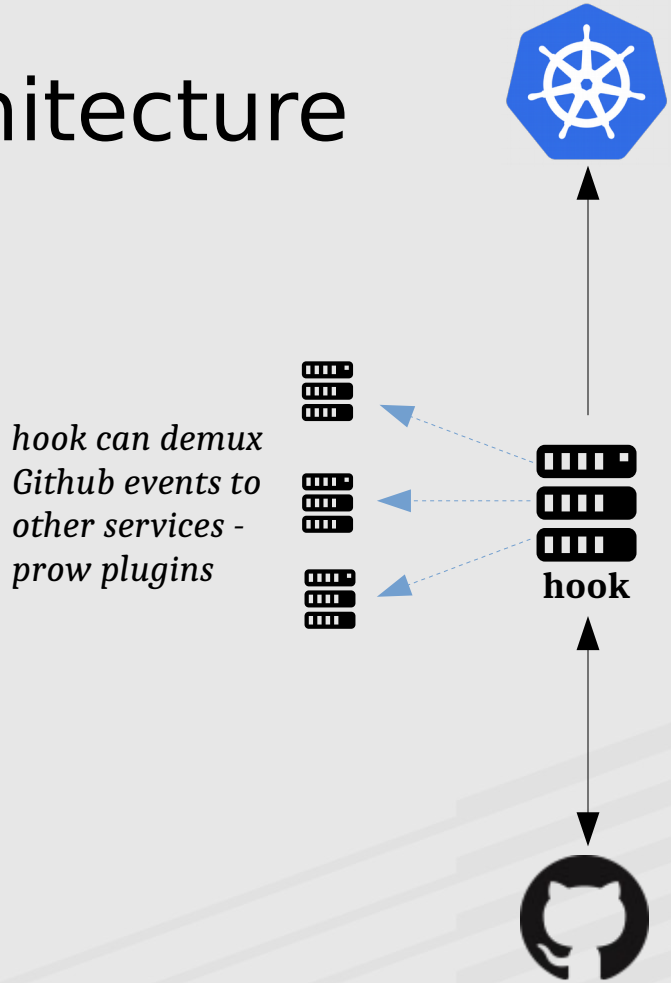
Entrypoint services for creating tests in the cluster:

- **hook**, based on Github events
- **horologium**, based on configured periodic jobs
- **tide** handles merging and retesting pull requests

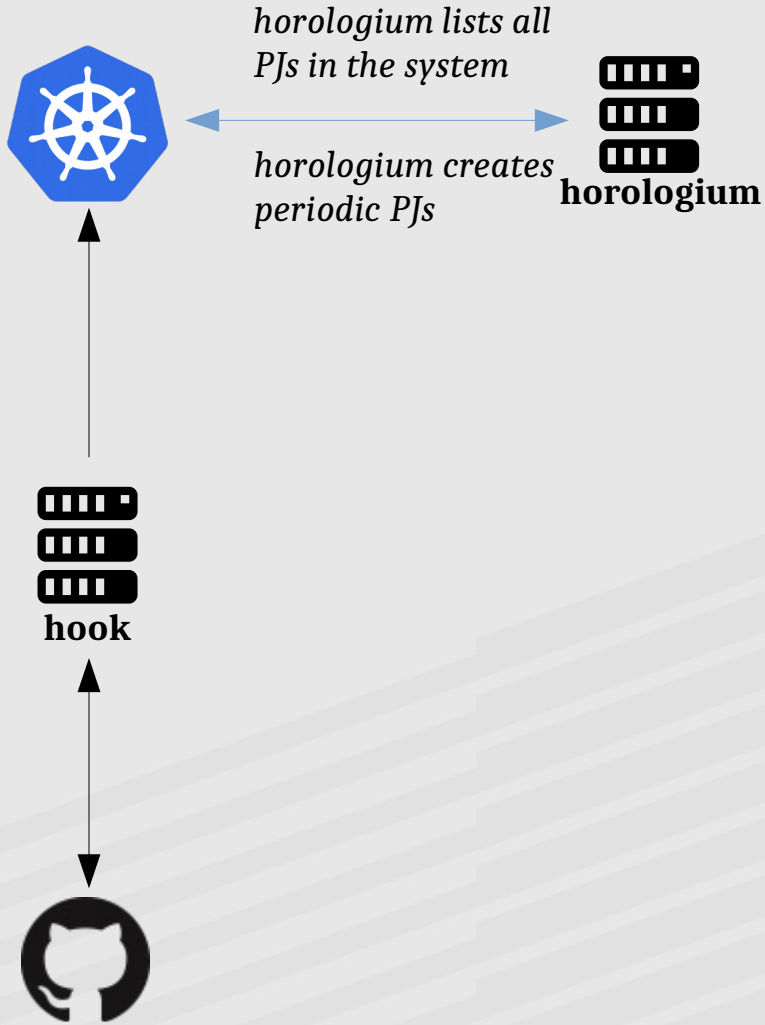
CI architecture



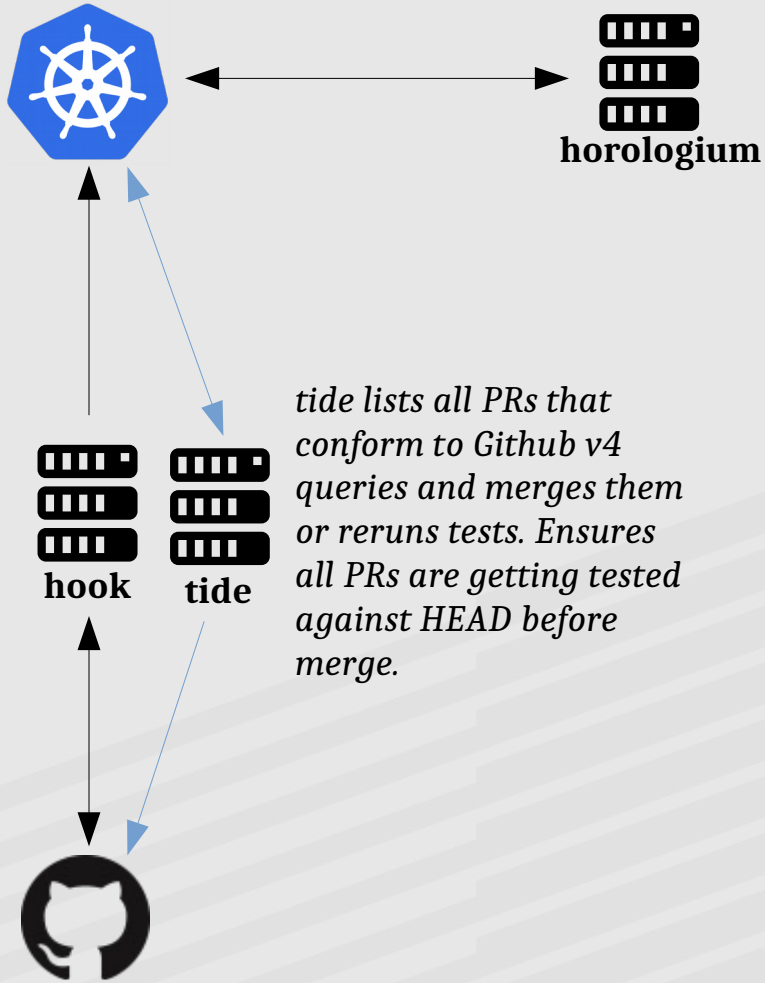
CI architecture



CI architecture



CI architecture

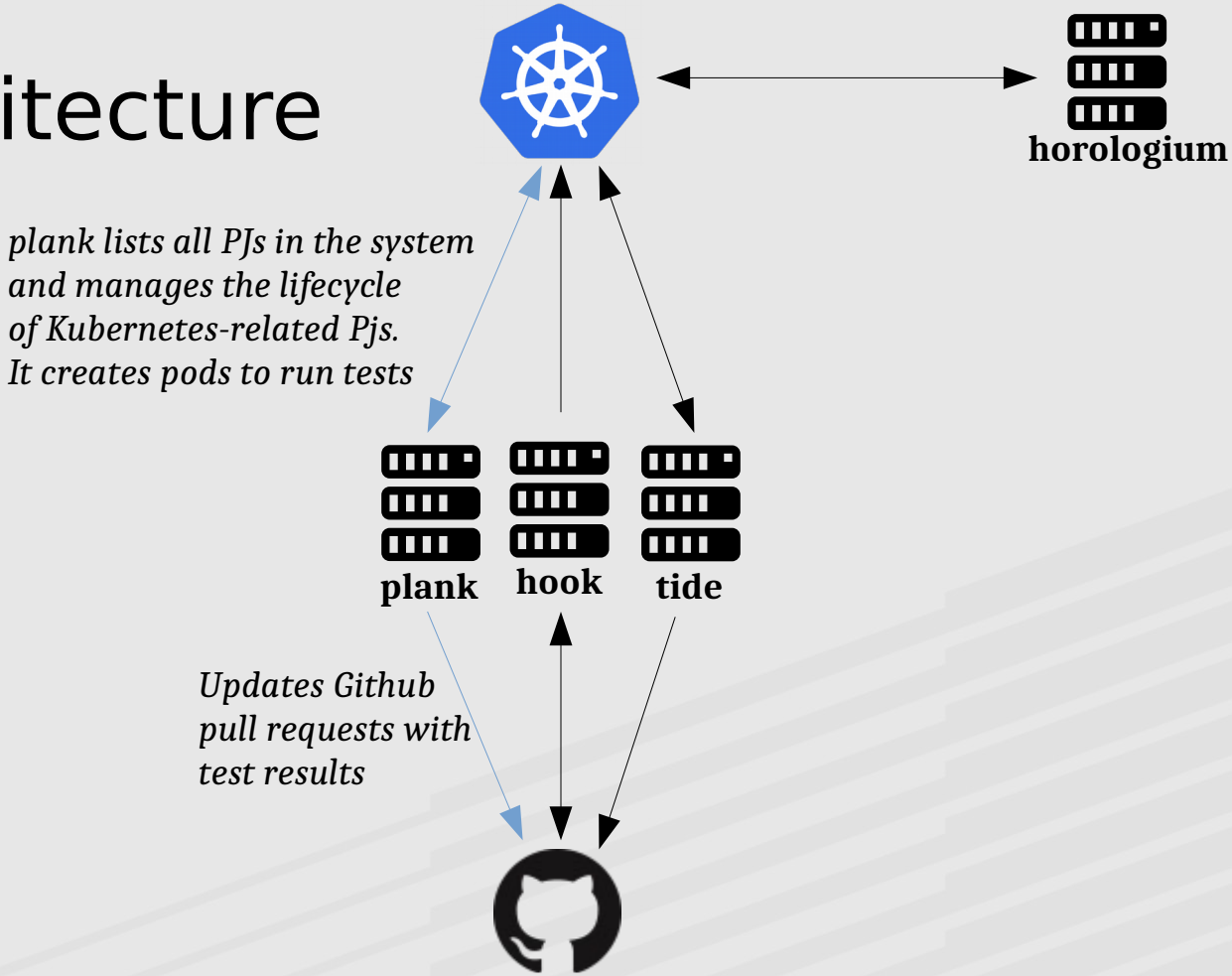


CI architecture

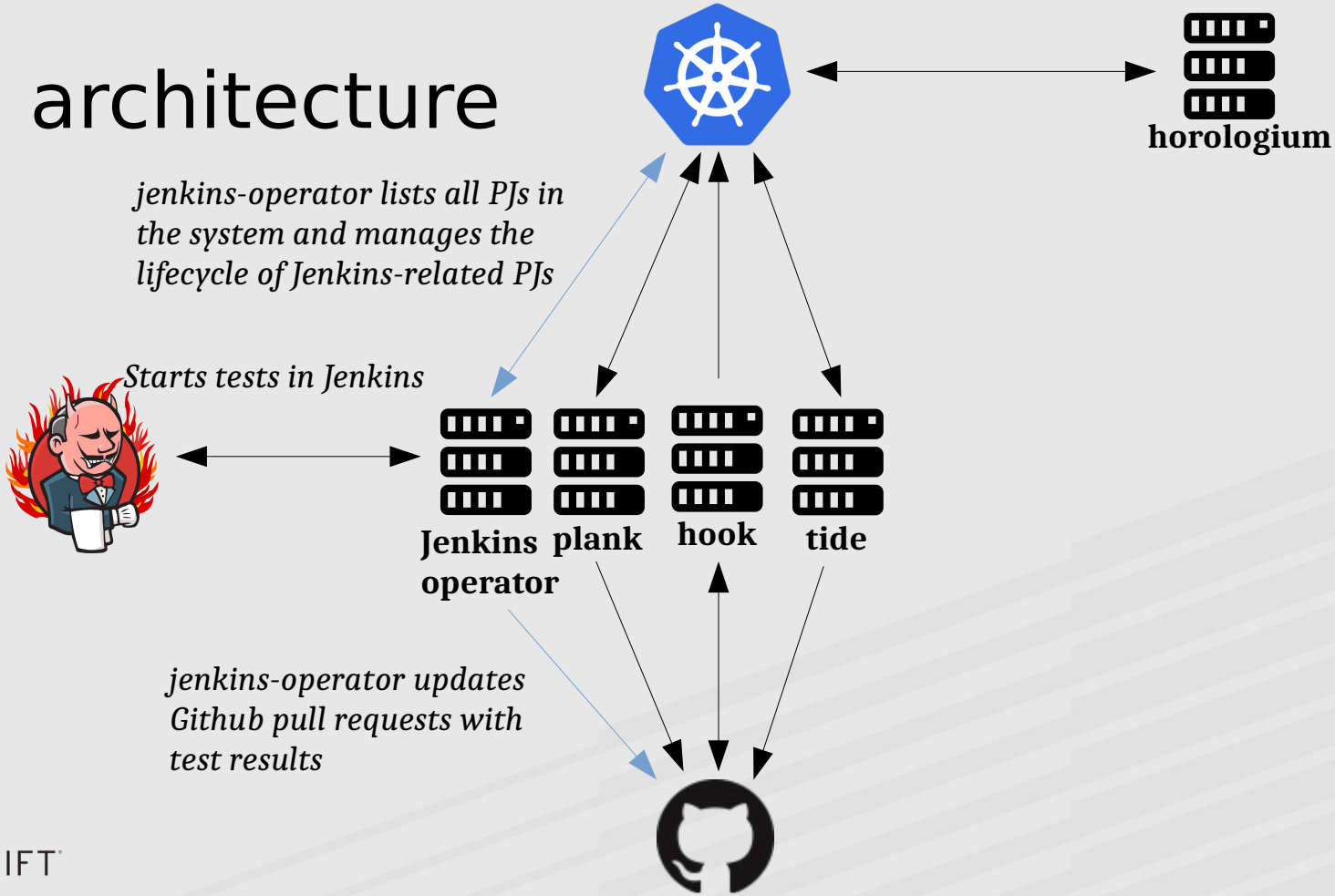
Two services for managing the lifecycle of tests:

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

CI architecture



CI architecture

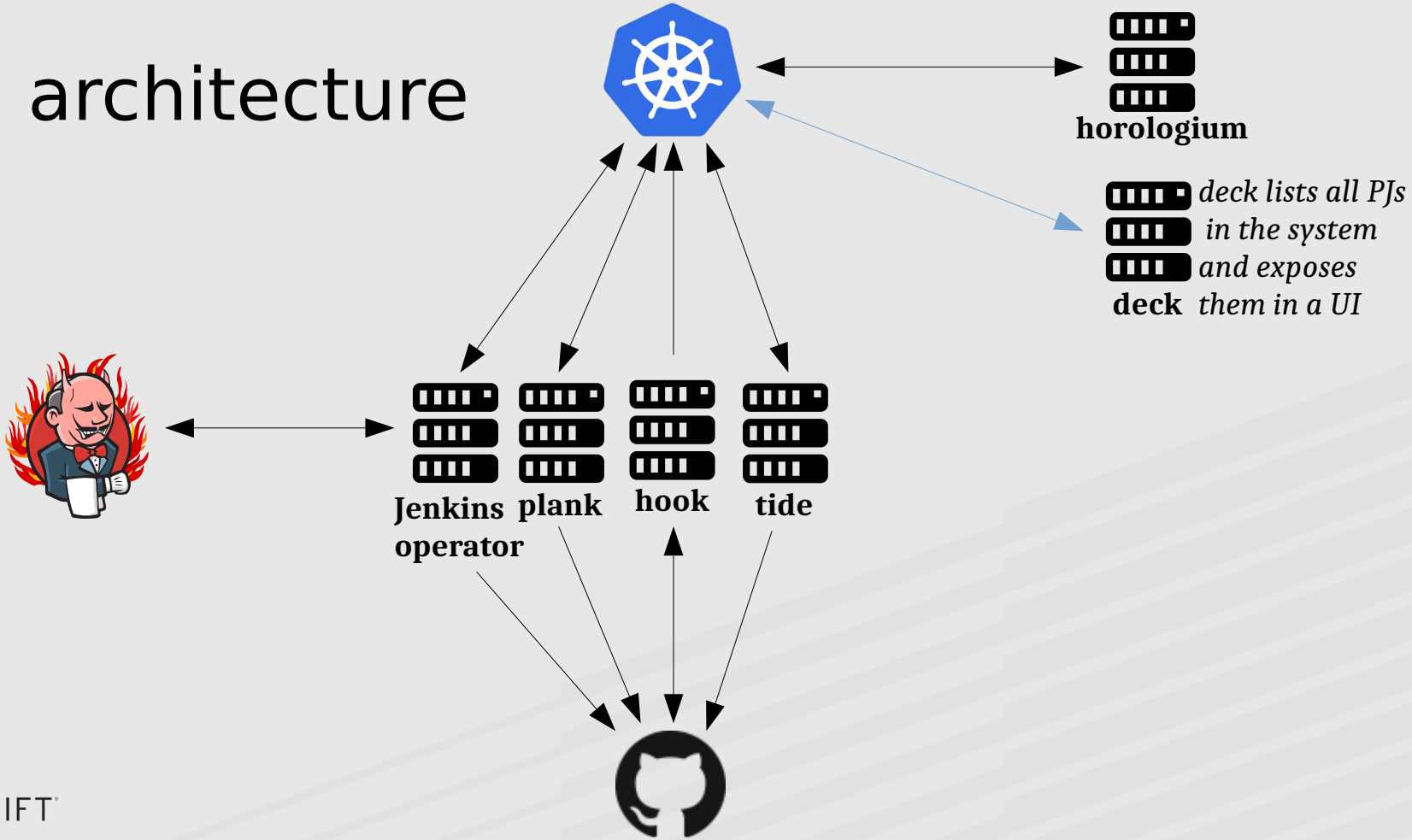


CI architecture

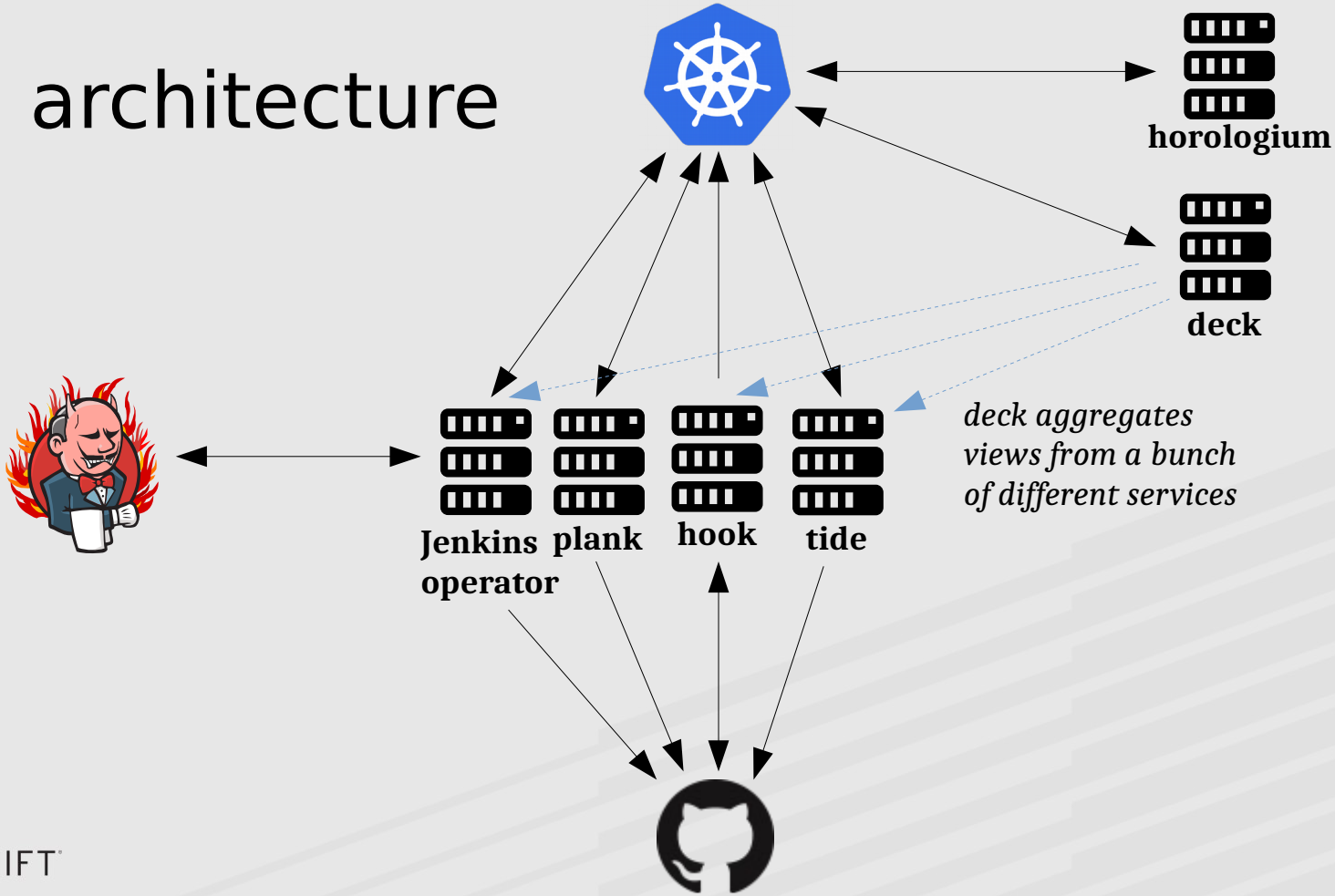
Others:

- **deck** is the front-end of prow
- **sinker** handles garbage collection of old PJs and pods

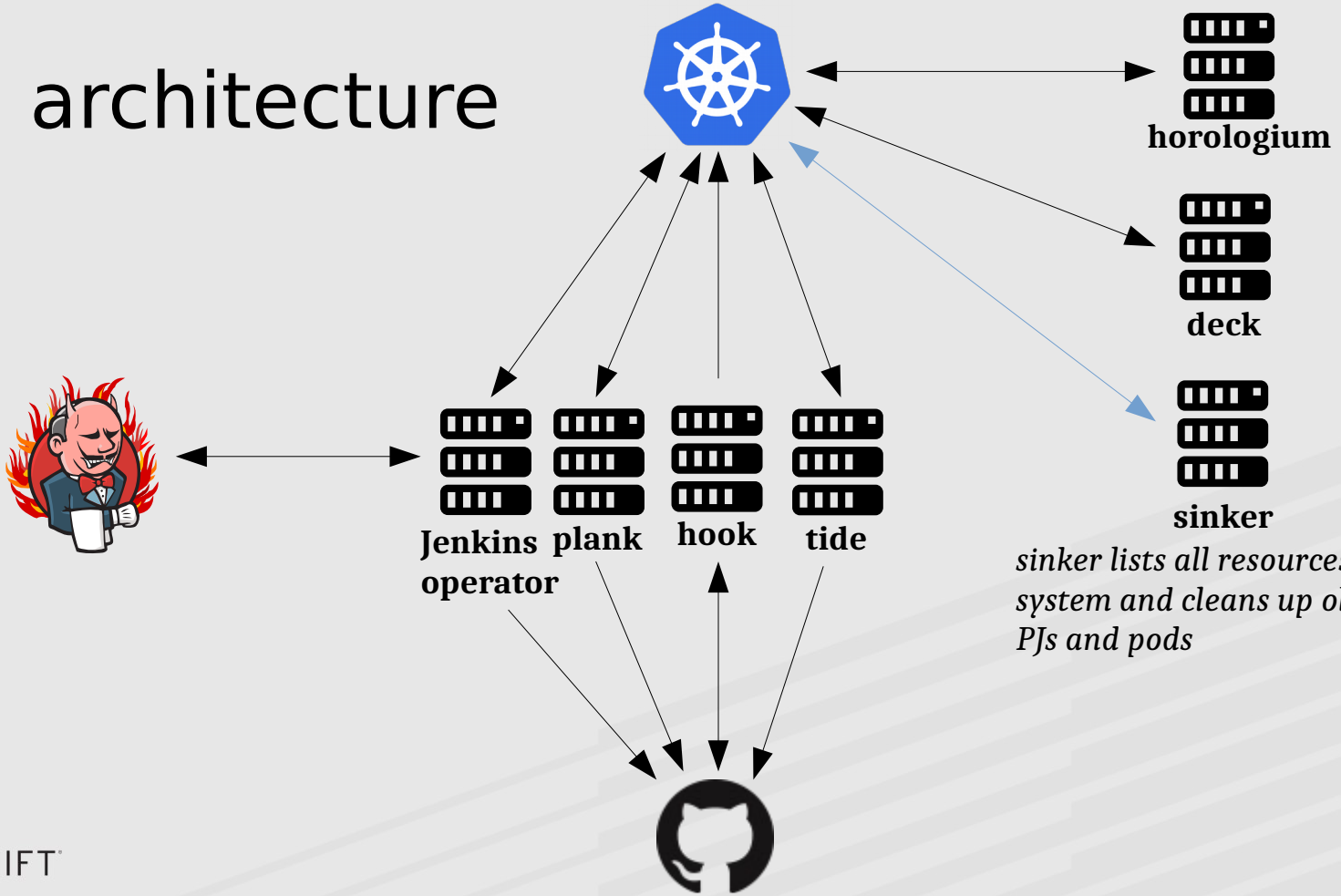
CI architecture



CI architecture

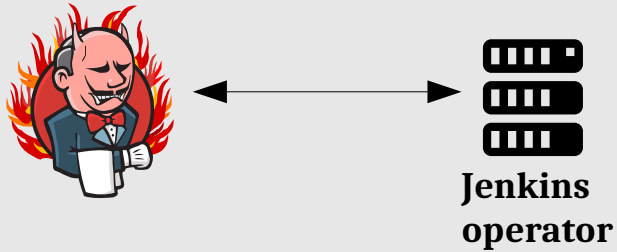


CI architecture



sinker lists all resources in the system and cleans up old PJs and pods

Problems (of the present)



- Jenkins does not scale to our needs
- Impossible to fix itself when it crashes
- Wrong use of Jenkins pipelines

Future work

- Move all Openshift repos to use prow
- Re-architect Jenkins pipelines into Golang pipelines
- Release a declarative API to users that want to setup testing for their projects

Fin