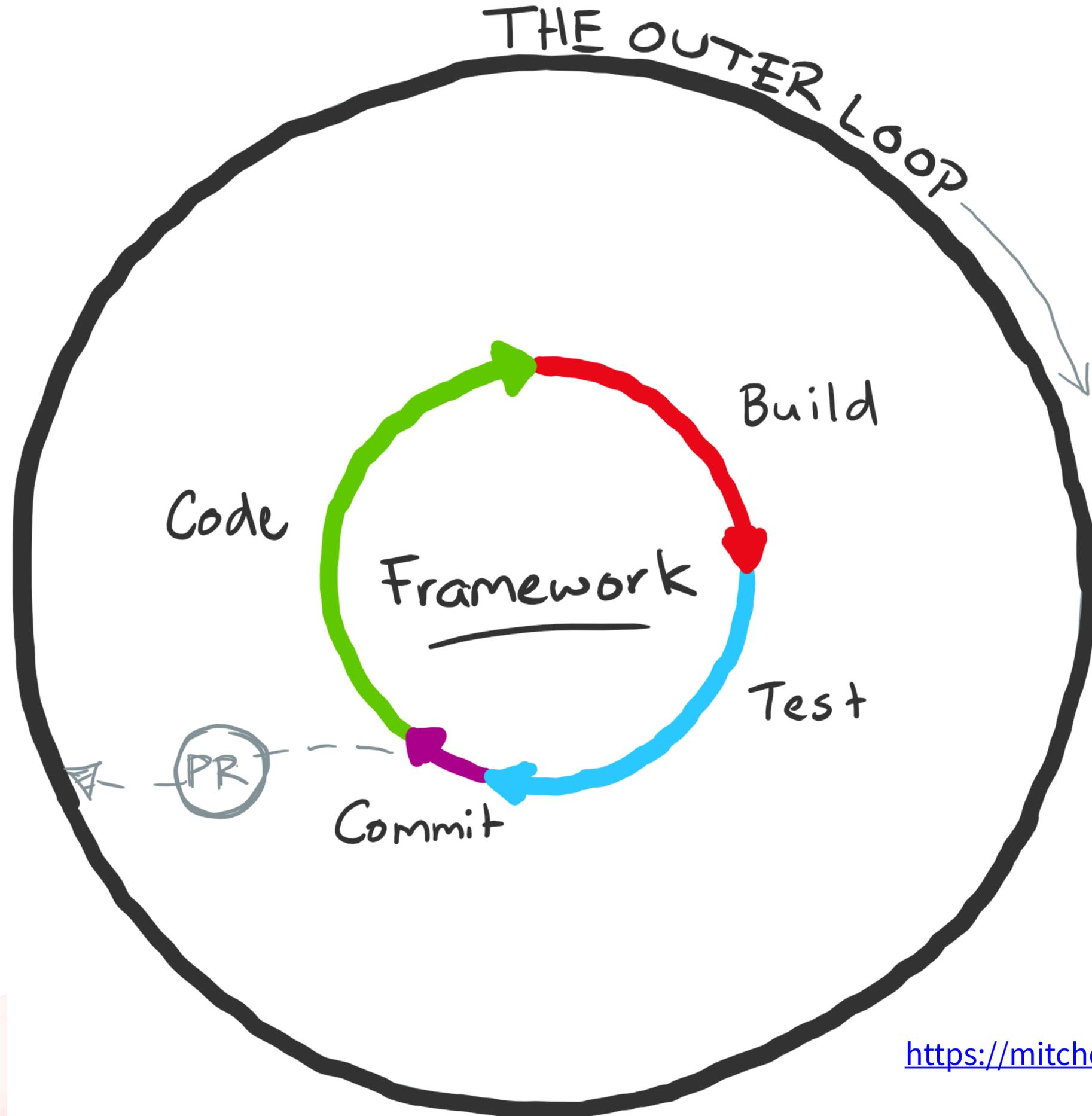


Intro to Telepresence: Fast Development Workflows for Kubernetes

tl;dr

- The inner development loop can be painful with microservices + Kubernetes
- Telepresence proxies your dev machine into the cluster
- Telepresence supports several workflows, from small to large systems
- Ongoing work for splitting client and cluster components
- Please support this CNCF project — get involved!

Setting the Scene



The Kubernetes inner & outer loop can be the same...



- Write code
- Build container
- Push to registry
- Deploy to cluster
- Test

And can be SLOW.

Automation helps speed it up ...



DRAFT



SKAFFOLD



garden



GITKUBE

But you want a *really* fast inner loop (no docker build/push) ...





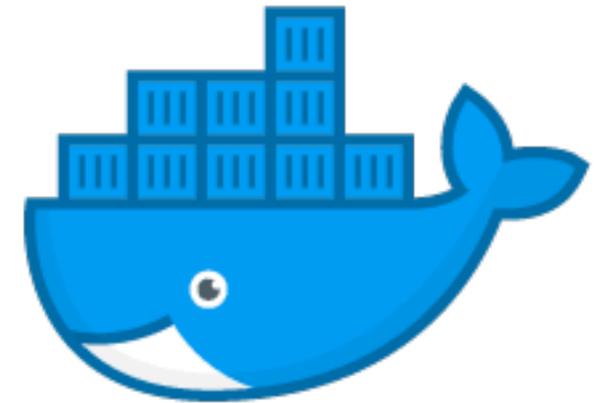
And you want to use your own tools.

Let's do everything locally...

...so it's really fast and I can use my tools!



minikube

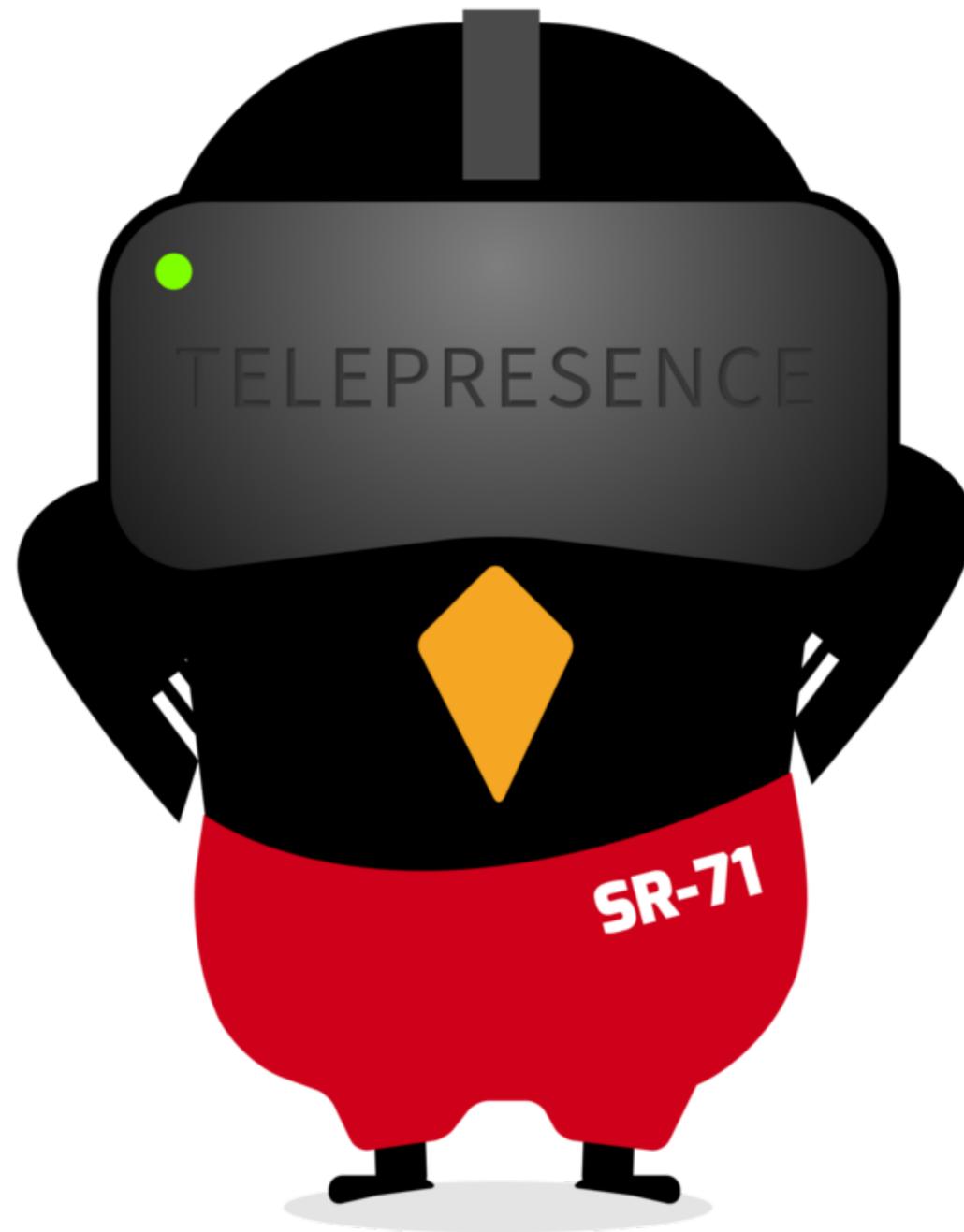


docker

MicroK8s

**“My laptop only
has 16GB RAM and
you’re asking it to
run Kubernetes
and a JVM and a
database???”**





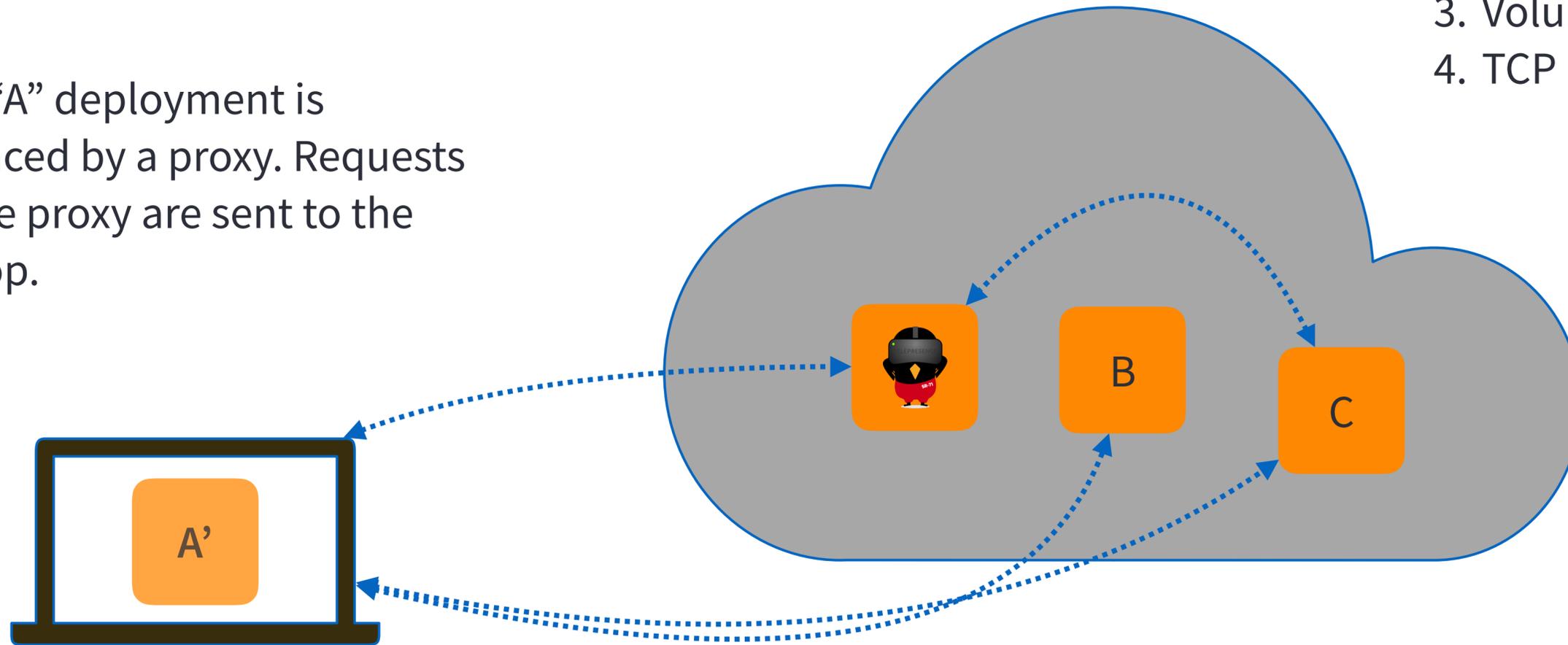
TELEPRESENCE

What is Telepresence?

- “Fancy Kubernetes VPN for development”
- “kubectl port-forward on steroids”
- A network bridge between your laptop and the Kubernetes cluster

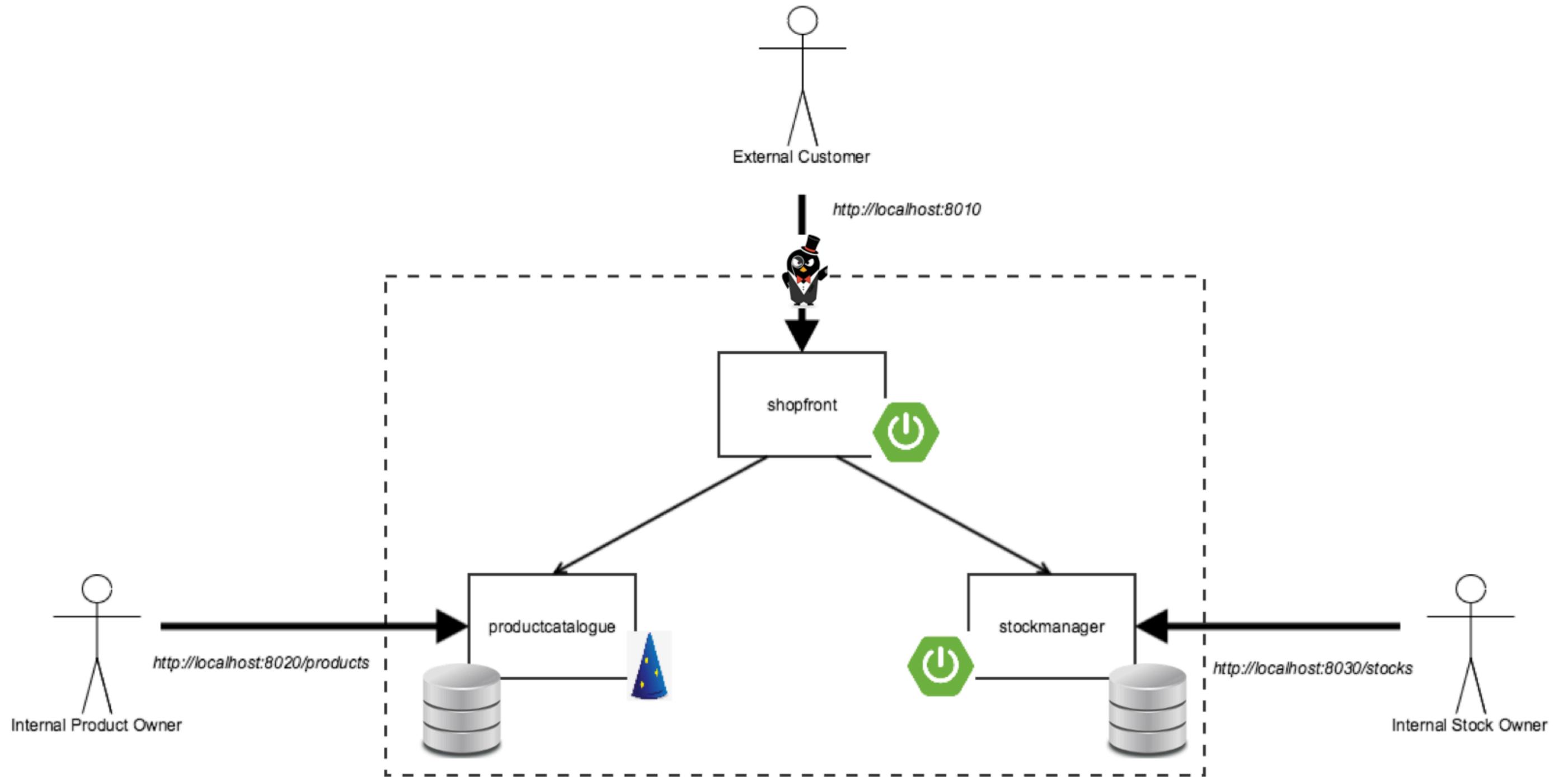
How it works

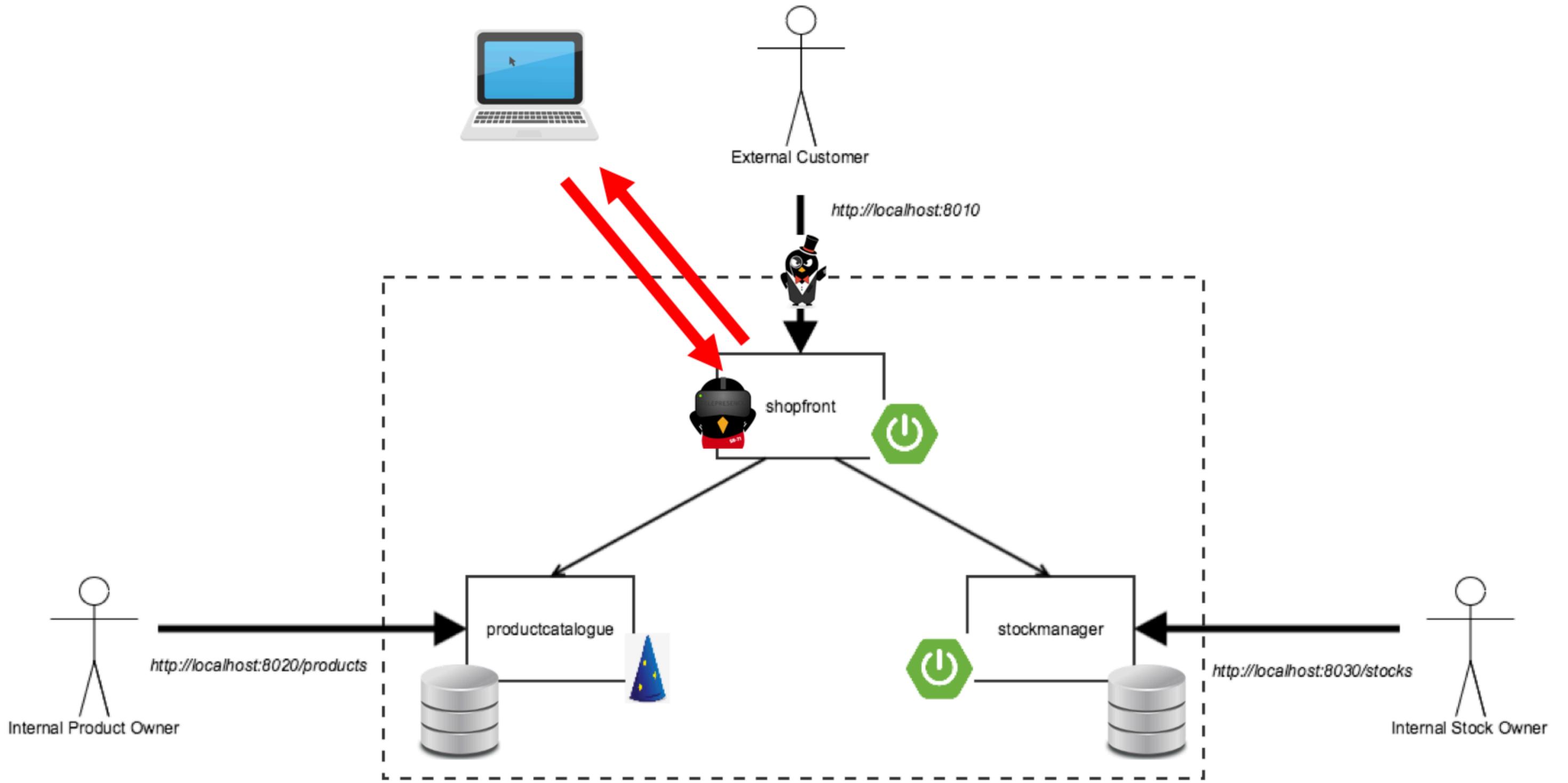
The “A” deployment is replaced by a proxy. Requests to the proxy are sent to the laptop.



Telepresence:

1. Intercepts DNS
2. Environment variables & secrets
3. Volumes
4. TCP





DEMO



Benefits

- Use any tool that runs on your laptop: IDE, profiler, debugger, ...
- Connect to cloud-based resources
- Same network namespace (e.g., nslookup works)
- Very fast inner loop!
- Requirements
 - Network connection
 - kubectl access to cluster
 - Mac or Linux

Telepresence is a CNCF project, with a wide variety of users.



TELEPRESENCE

▲ Why's that company so big? I could do that in a weekend (danluu.com)
712 points | Oct 3, 2016 | hide | past | web | favorite | 423 comments | add to buffer



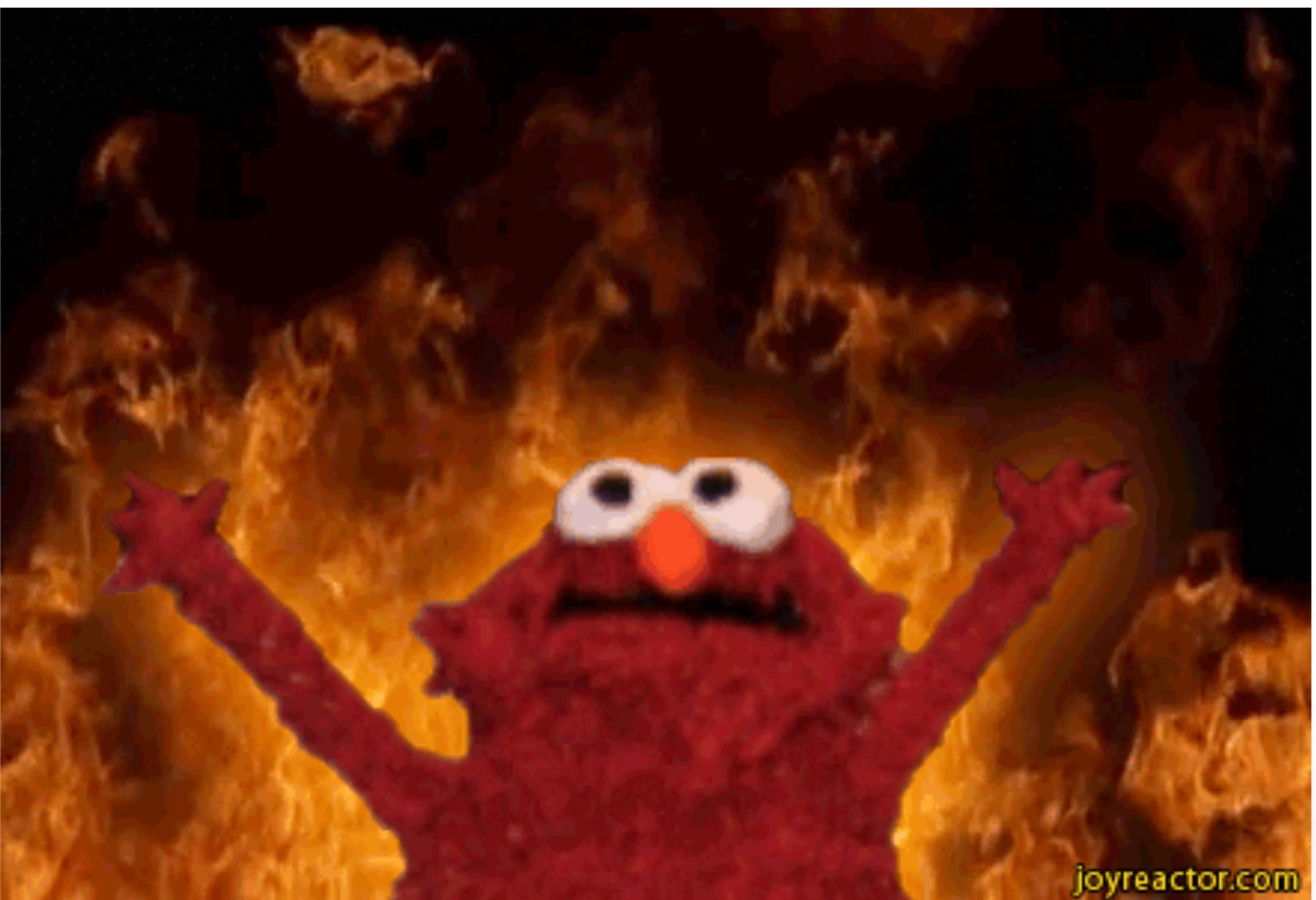
- Capturing DNS and routing only part of it to the cluster is tricky
- No two laptops are alike
- Swapping out your (real) deployment for a (proxy) deployment is tricky

USING THIS IN YOUR WORKFLOW

Simple Workflow (YOLO)

- One (production) cluster for single-source-of-truth microservices
- Developer checks out code and works on microservices
- Swapping services on demand using Telepresence + IDE
- Telepresence is used in default (vpn-tcp) mode
- Once code is done, microservices are built and deployed via pipeline

<https://www.youtube.com/watch?v=75soljoqhjY>



Typical Workflow (SME)

- One (dev) cluster for single-source-of-truth microservices
- Developers check out one of the microservices (and coordinate with other devs)
- Developers Telepresence + IDE
- Telepresence is used in default (vpn-tcp) mode
- Once code is done, CircleCI is used to run tests, build, and deploy

<https://www.youtube.com/watch?v=75soljoqhjY>

@danielbryantuk | @datawireio



The image shows a YouTube video player. The main content is a presentation slide titled "INTRODUCING TELEPRESENCE" from DEVOXX United Kingdom. The slide features a diagram of a microservices architecture. A "LOCAL PROCESS" (represented by a laptop icon) is connected to a "LOAD BALANCER" which routes traffic to "SERVICE A", "SERVICE B", and "SERVICE C". Each service is connected to a "PROXY" which in turn connects to "CONFIG" and "MQ" (Message Queue) components. The entire system is labeled "K8S CLOUD". A "Config map" is also shown. The video player interface includes a progress bar at 4:44 / 15:34, a play button, and social media icons for YouTube, Twitter, and LinkedIn. The video title is "Fast Kubernetes development with Telepresence.io by Cesar Tron-Lozai" and it has 286 views as of May 16, 2019.

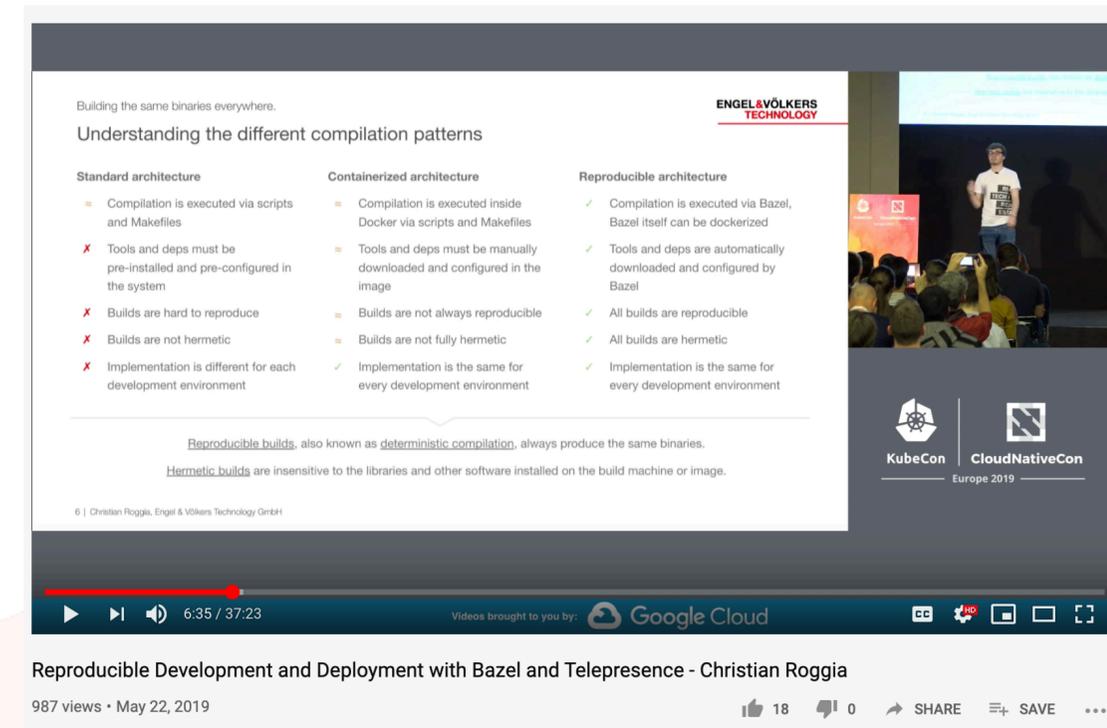
Advanced Workflow

- One cluster with multiple namespaces for all single-source-of-truth microservices
- Developers check out one of the microservices
- Developers use Bazel + Telepresence + IDE
- Telepresence is used in container mode with Bazel images
- Once code is done, Jenkins is used to run tests, build, and deploy



ENGEL & VÖLKERS®

<https://www.youtube.com/watch?v=tD0FIlxO1AQ>



Building the same binaries everywhere.

Understanding the different compilation patterns

Standard architecture	Containerized architecture	Reproducible architecture
<ul style="list-style-type: none">Compilation is executed via scripts and MakefilesTools and deps must be pre-installed and pre-configured in the systemBuilds are hard to reproduceBuilds are not hermeticImplementation is different for each development environment	<ul style="list-style-type: none">Compilation is executed inside Docker via scripts and MakefilesTools and deps must be manually downloaded and configured in the imageBuilds are not always reproducibleBuilds are not fully hermeticImplementation is the same for every development environment	<ul style="list-style-type: none">Compilation is executed via Bazel, Bazel itself can be dockerizedTools and deps are automatically downloaded and configured by BazelAll builds are reproducibleAll builds are hermeticImplementation is the same for every development environment

Reproducible builds, also known as deterministic compilation, always produce the same binaries.

Hermetic builds are insensitive to the libraries and other software installed on the build machine or image.

© | Christian Roggia, Engel & Völkers Technology GmbH

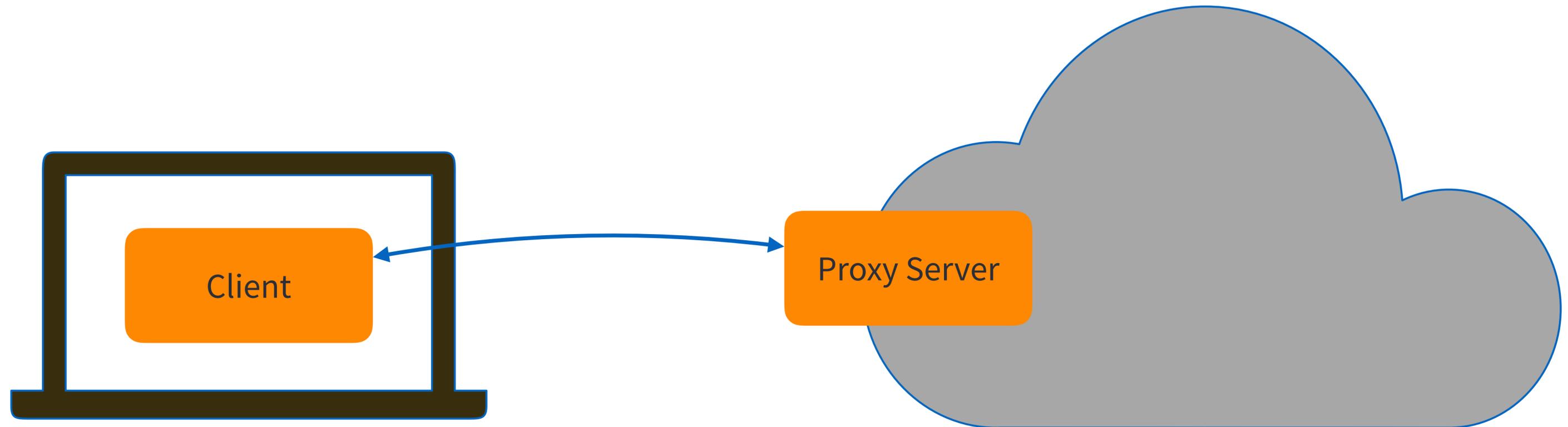
Reproducible Development and Deployment with Bazel and Telepresence - Christian Roggia

987 views · May 22, 2019

18 0 SHARE SAVE

WHAT'S NEXT?

Telepresence components



- Capture DNS requests for Kubernetes resources
- DNS resolution for Kubernetes resources
- Proxy TCP connections to cluster

- Proxy TCP connections to laptop
- Canonical DNS for Kubernetes resources

Telepresence Client

- Use kubectl port-forward to bootstrap sshuttle (ssh-based VPN)
- sshuttle has a lot of functionality that is not used by Telepresence (it's a VPN, after all)
- Replace sshuttle with something else — designed for Kubernetes
 - Capture DNS locally
 - Do DNS resolution intelligently
 - Redirect TCP connections to the right destination
- Already in progress, will replace VPN-TCP method
- Also support multiple simultaneous swaps

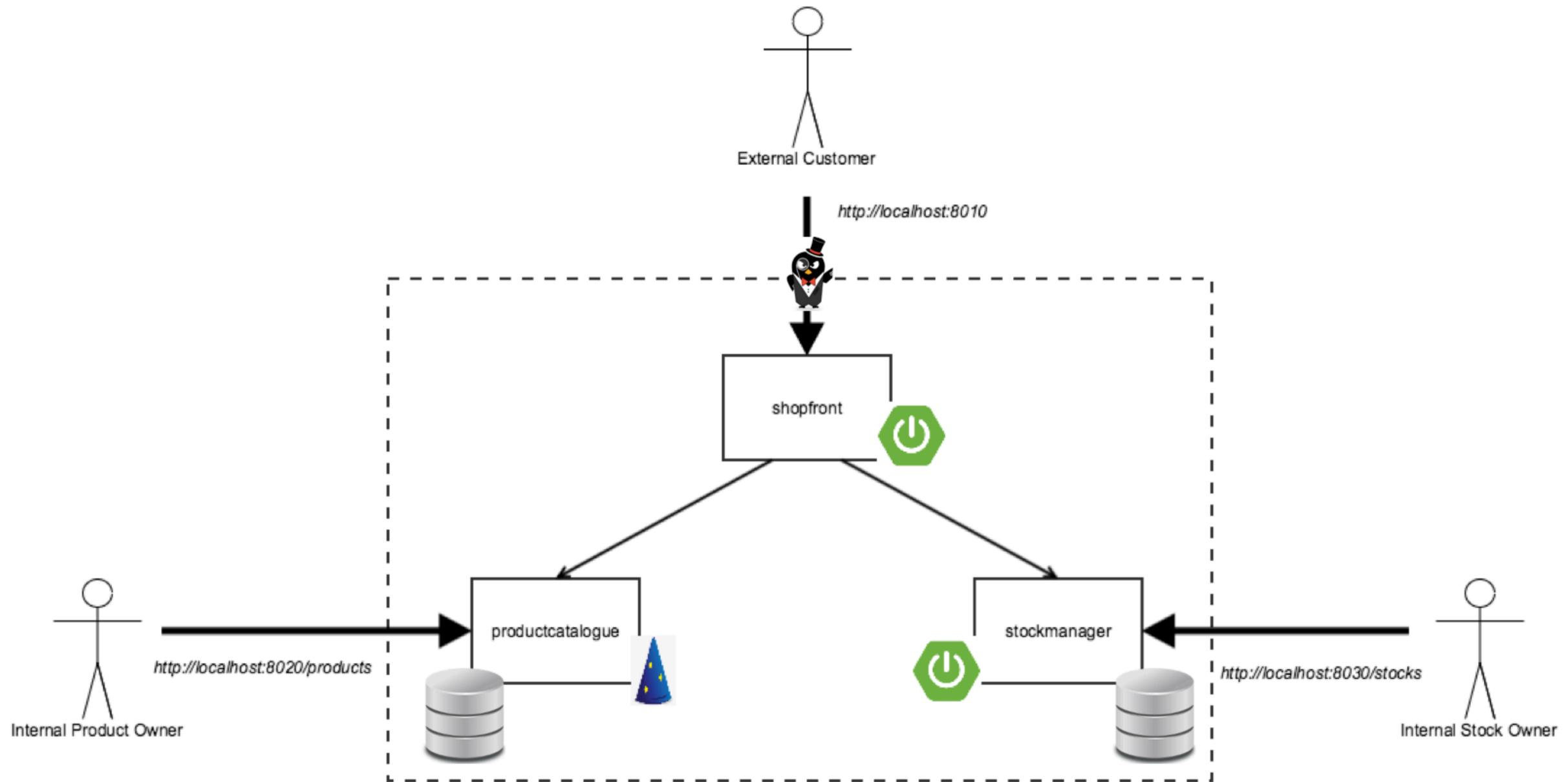
Telepresence Proxy Server

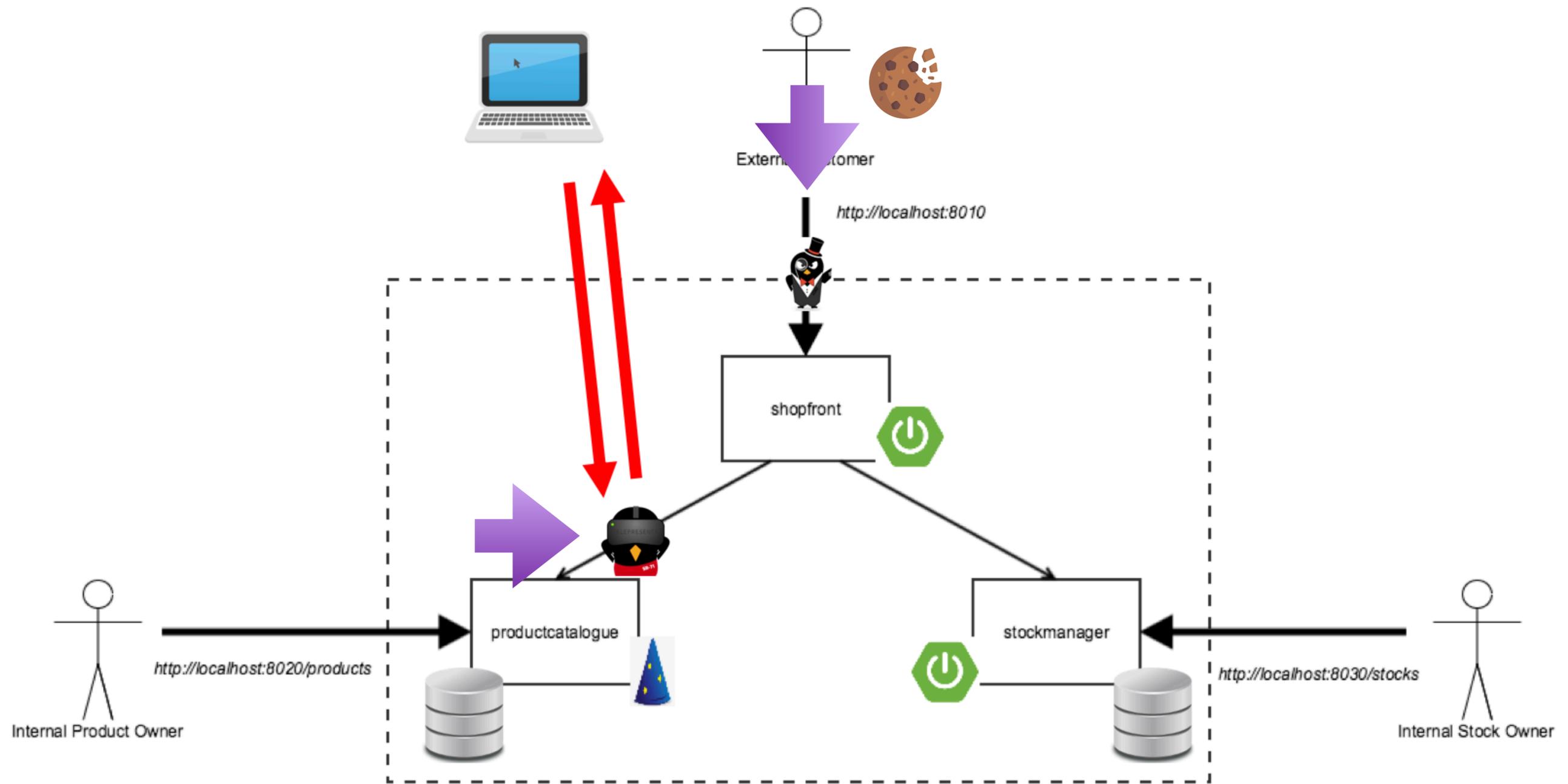
- Currently deployed every time you do a swap-deployment (and then deleted)
- Switch mode to be persistent (this is what Kubernetes is good for)
 - Speeds up startup
 - Support reconnect

Multi-User Development

- Swap deployment doesn't work well for multi-user development
- Solution today is to use namespaces and/or individual clusters
 - Expensive (more hardware & setup)
 - Delays integration testing until after commit







Wrapping Up

Contributing to Telepresence

→ Documentation and workflows

- * *how* you integrate Telepresence into your development workflow

→ Let us know if you're a user!

→ Triage

- * Lots of folks use GitHub for support, which doesn't scale well

→ Join Slack to help other users <https://d6e.co/slack> #telepresence

Thank you! Questions?

- db@datawire / @danielbryantuk
- www.telepresence.io
- <https://github.com/telepresenceio>
- More KubeCon sessions — 17:20 Today!!
 - * “Use Your Favorite Developer Tools in Kubernetes With Telepresence” by Abhay Saxena in Room 30ABCDE
- Visit us at Booth #S46
- Sign up for the Ambassador Edge Stack preview
 - * <https://www.getambassador.io/early-access>

