On-demand development environments

Meilleurs Agents

- Web platform which connects seller and real estate professionals
- Production environment on GCP (60% apps in kubernetes, 40% legacy apps in VMs)
- 46 developers (frontend and backend)
- 8 SREs

- and 2 opened SRE positions (confirmed and senior)c

Padok

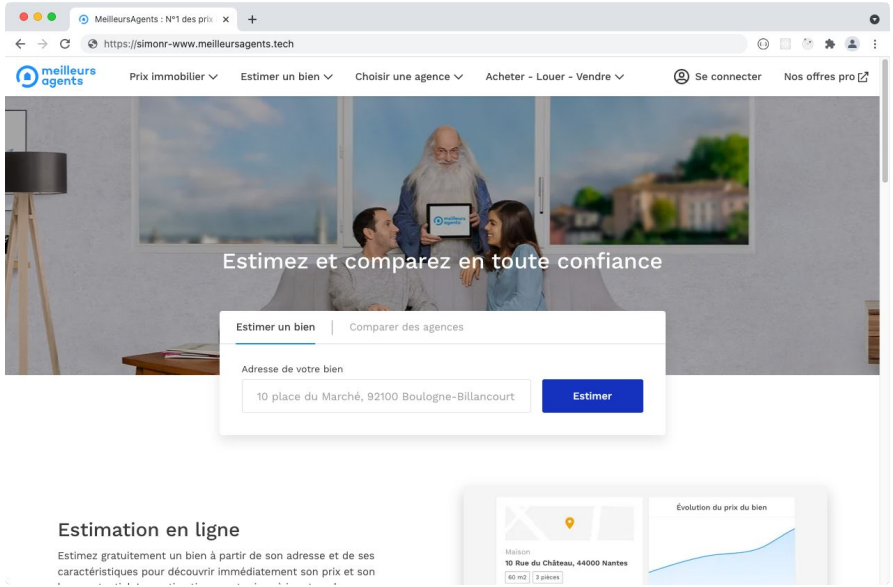
- Cloud migration, security and run
- 50 people, 30 SREs

- We are hiring!

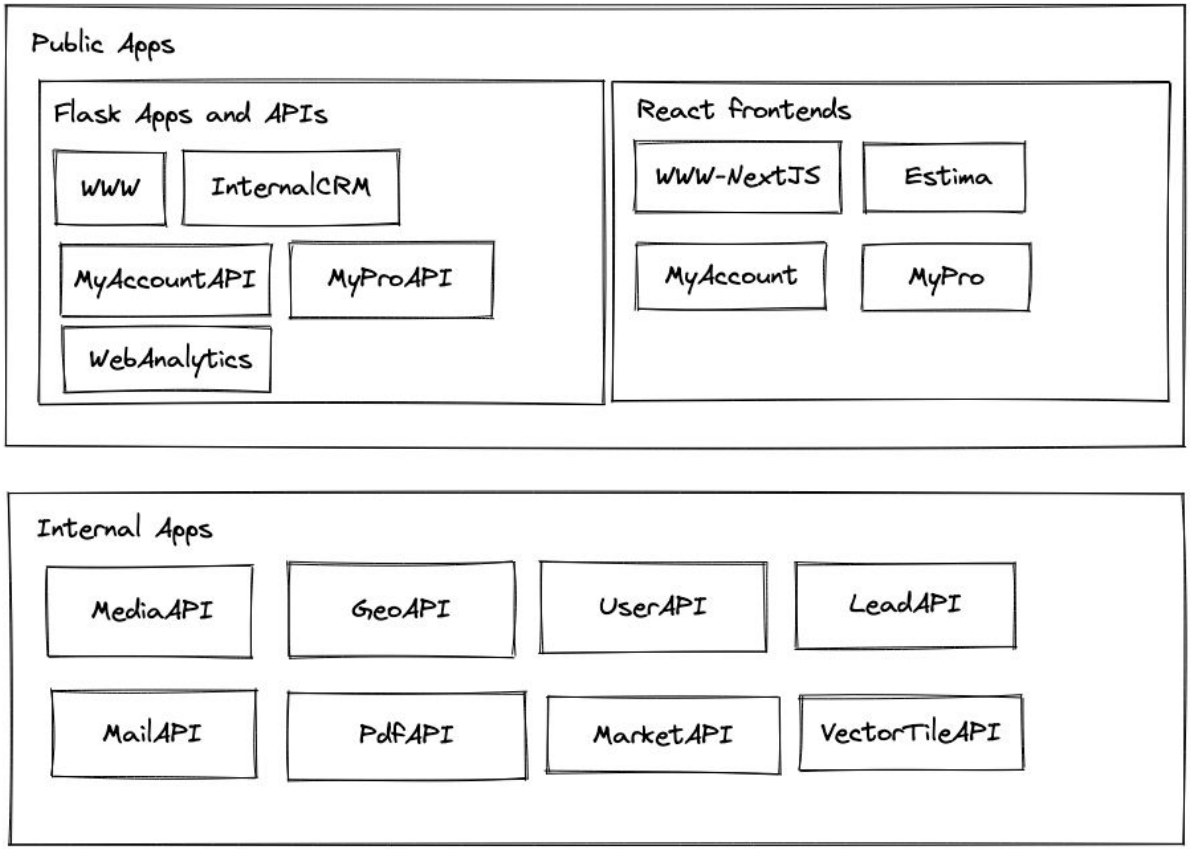
The environment before the project

The original need

- Full easy to update platform which every web dev can update
- Product team members have access to this environment for testing purposes
- Easy to update when someone merge a Pull Request
- No access for the outside world



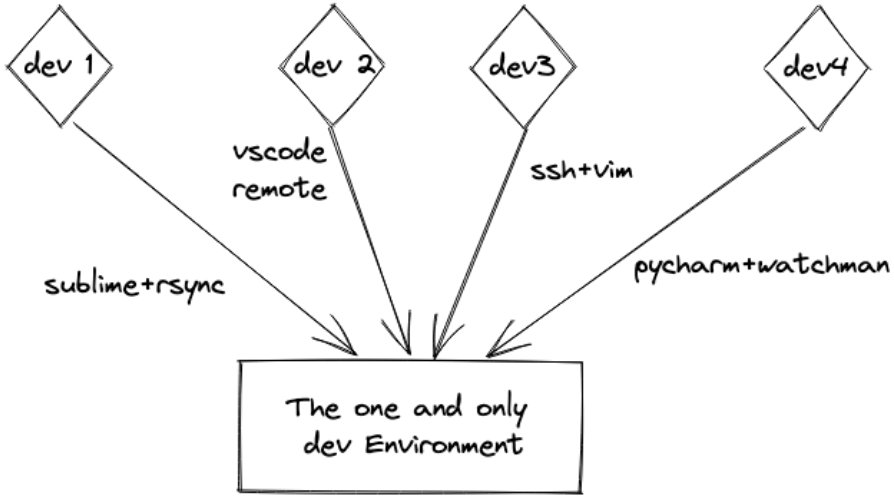
Meilleurs Agents web platform (simplified)



The beginning

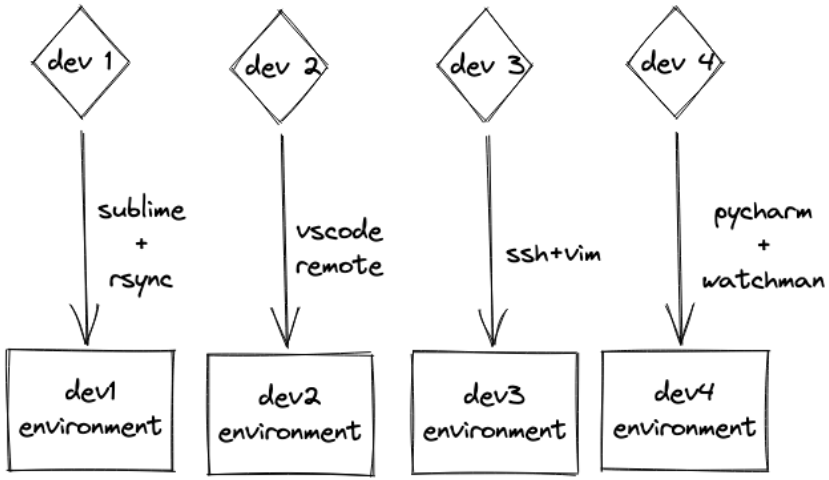
- **One** distant virtual machine for all developers
- A very complicated nginx configuration
- Everyone work in their ~ ($HOME) directory
- Laptop is just a terminal for ssh+tmux or rsync

- A problem?
  - Did you check `df -h`?
  - Did someone played with nginx?
  - Broken for everyone in the open space



How it evolved (2018)

- Every developers have their own virtual machine (20 at the time)
- Terraform and Ansible for provisioning (~15 minutes)
- Interactive shell script for:
  - git clone
  - tools installation
  - systemctl configuration
  - approximately 1 hour
- Update with `git pull` on app repositories



Pros and cons

Pros

- 1 machine per developer, no conflicts on the filesystem
- URL available with VPN or in the workplace
- Really easy and fast to update an application (if this one is working)

Pain points

- Lack of debugging knowledge for web developers (shell, systemctl, docker for frontends...)
- Disk usage with docker images (50go at first, 100+go in the end)
- Costs (can be moderated with a good lifecycle management)
- Really different architecture from **production** environment
- Does not scale with growing team

New integration environment : the objectives

Each team must have its own independent environment

The time between production and development deployment must be inferior to 5 min

Deploy a custom version of an application in an environment must be inferior to 5 min

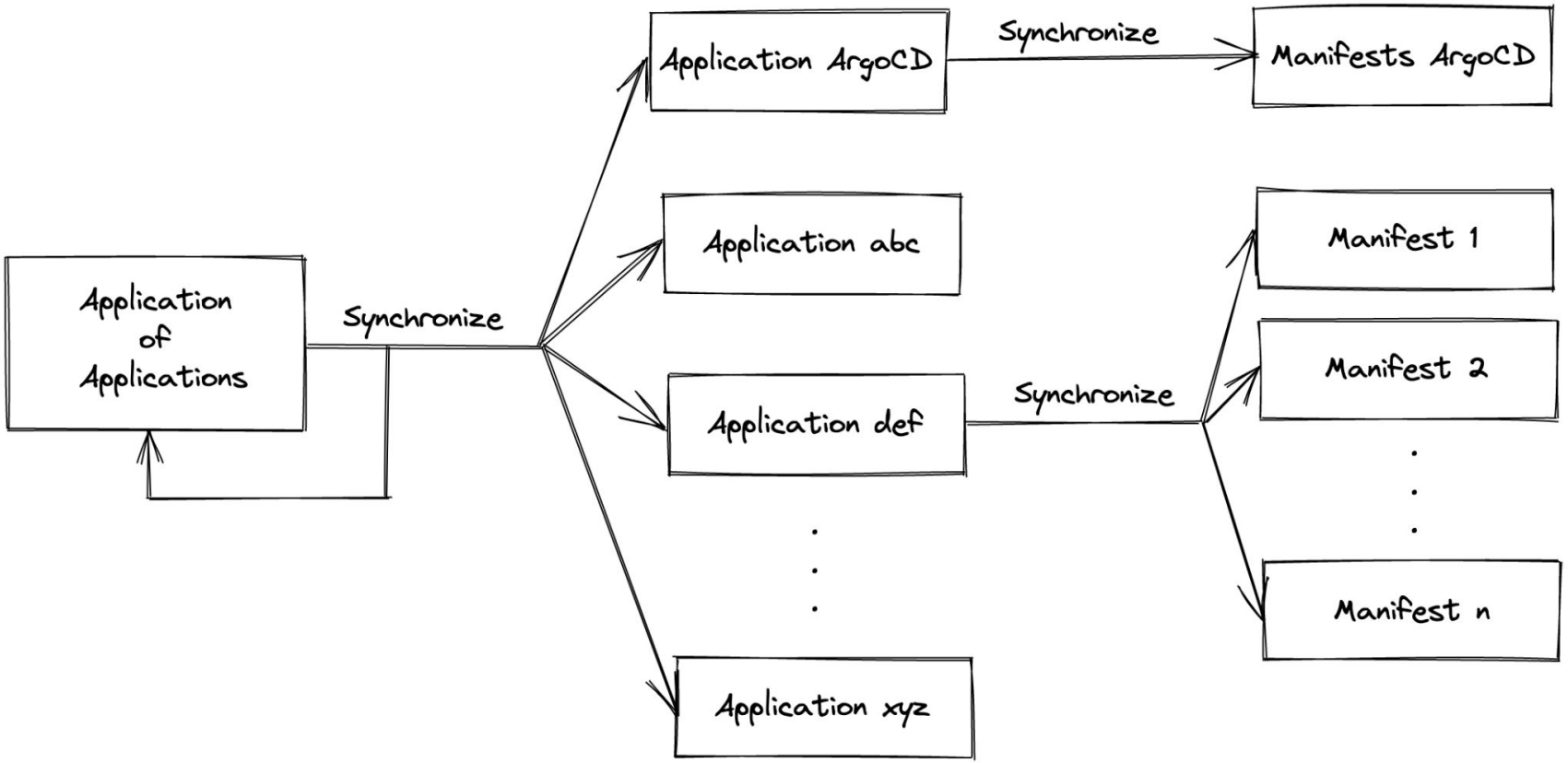
A full environment must be easily reproducible

Creating a new environment must take less than 5 min

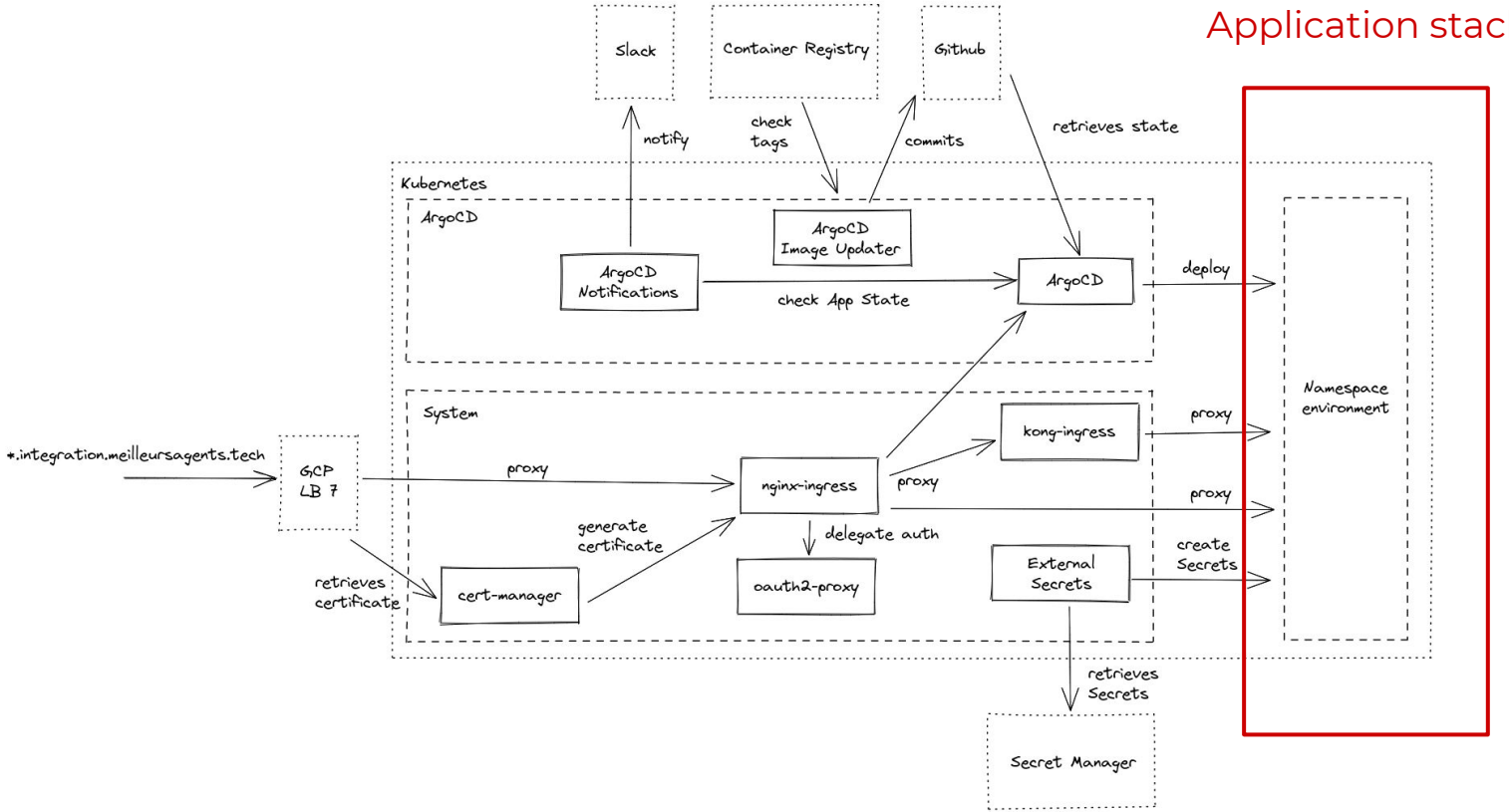
Integrate local development with the new infrastructure

The new infrastructure

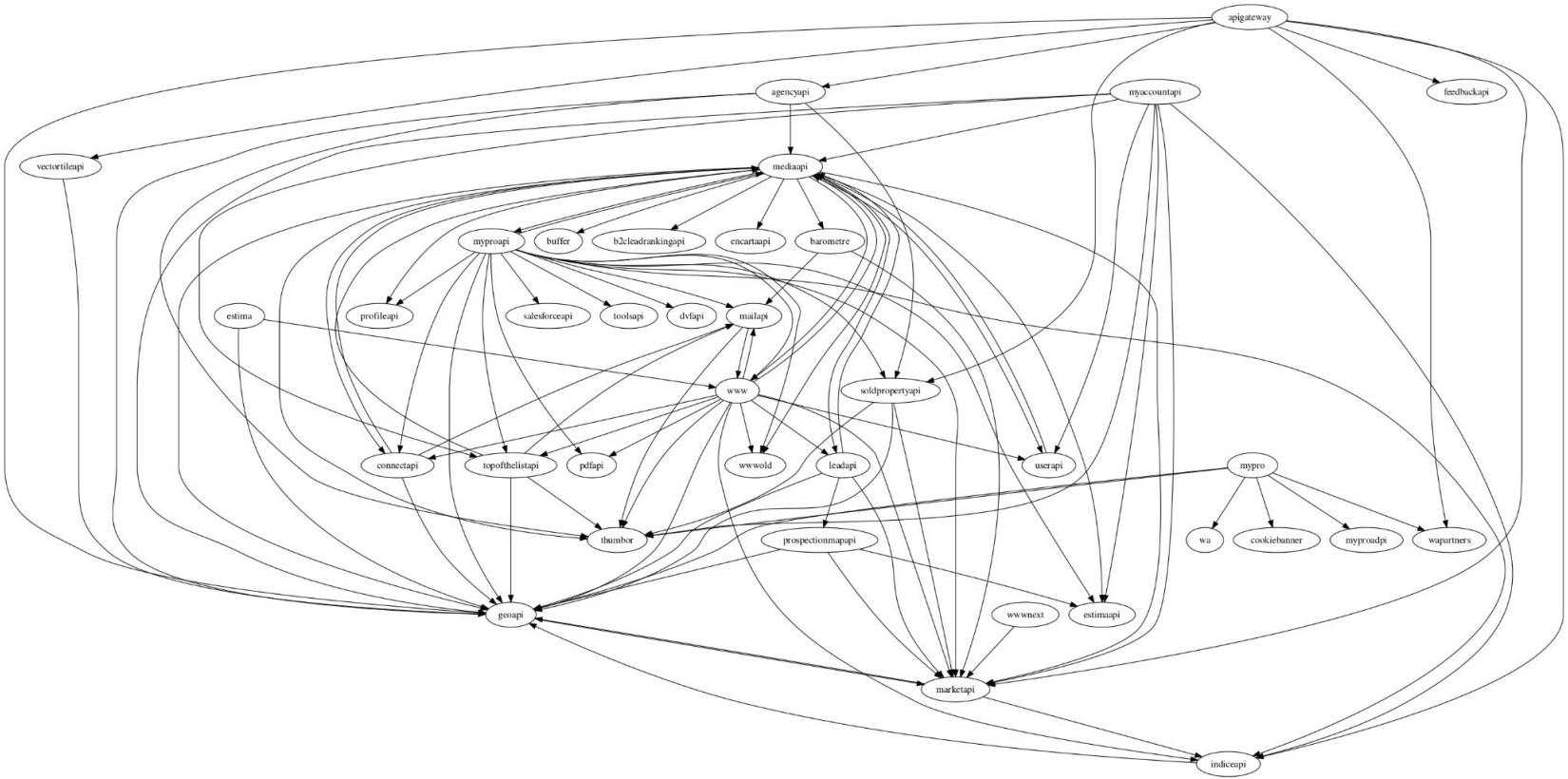
ARGOCD: APP OF APPS PATTERN



GLOBAL ARCHITECTURE



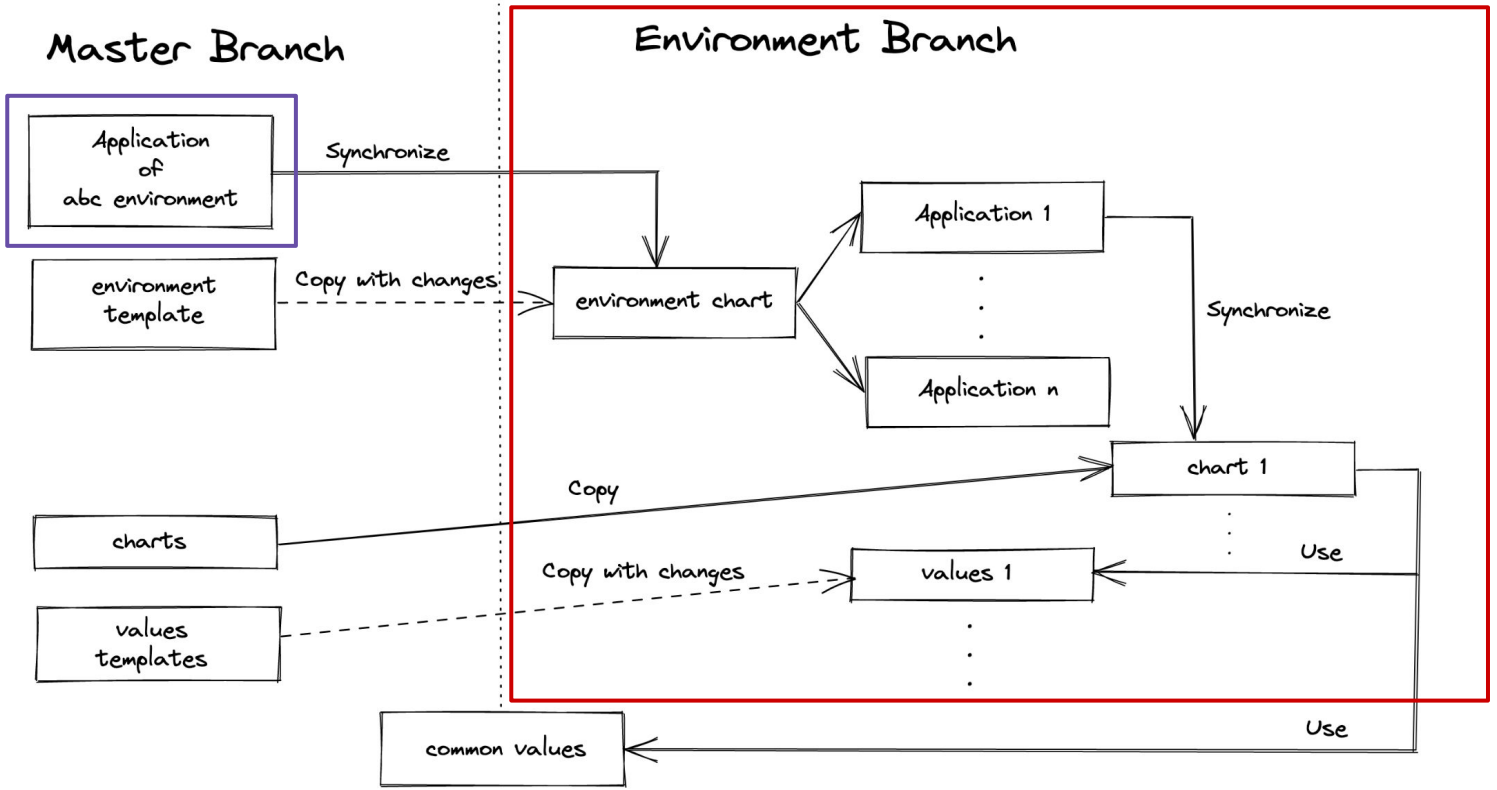
A quick overview of the application dependencies



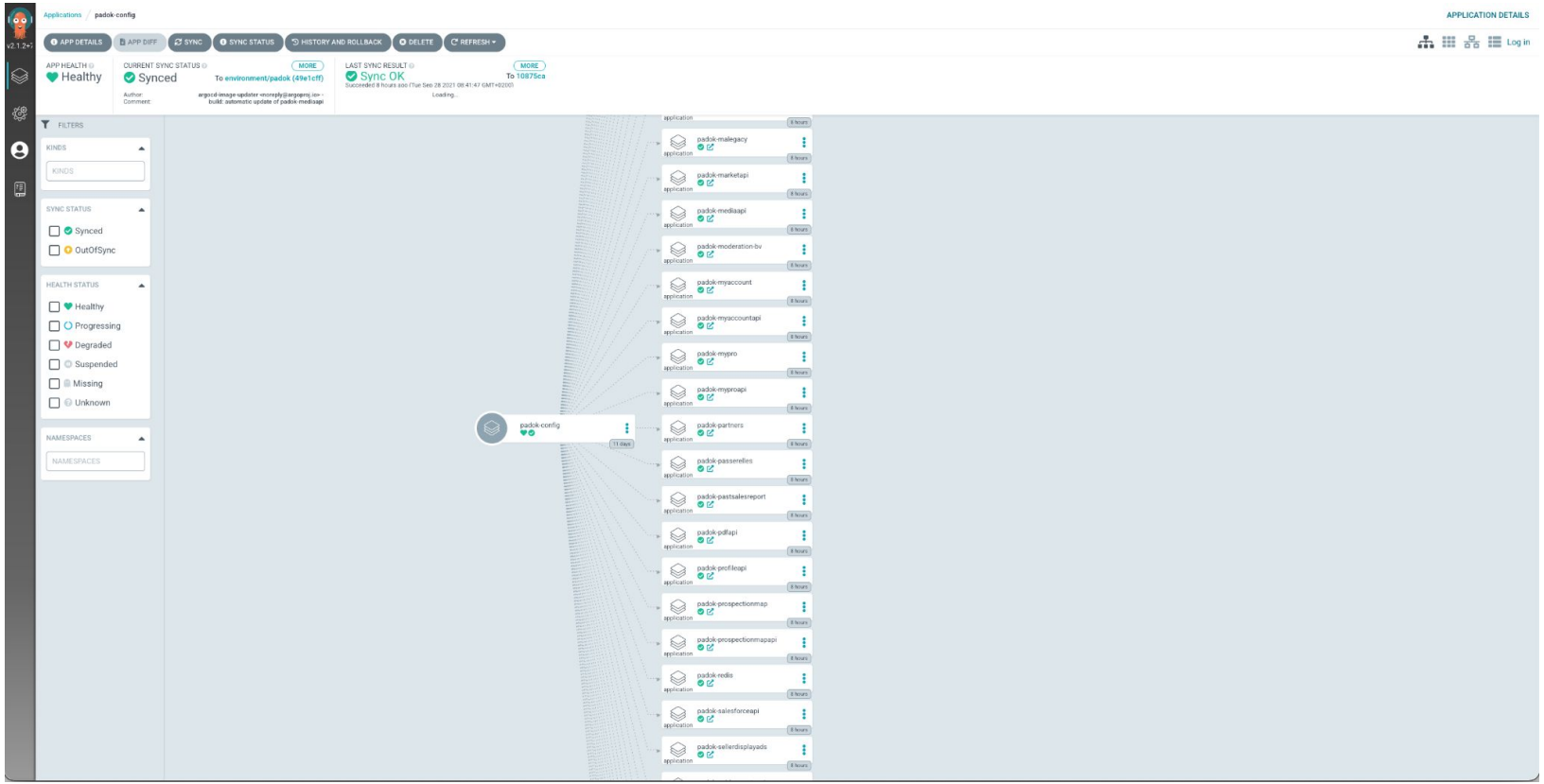
REPRODUCIBILITY: ENVIRONMENT CHART

Generated with Github Action

Generated with script

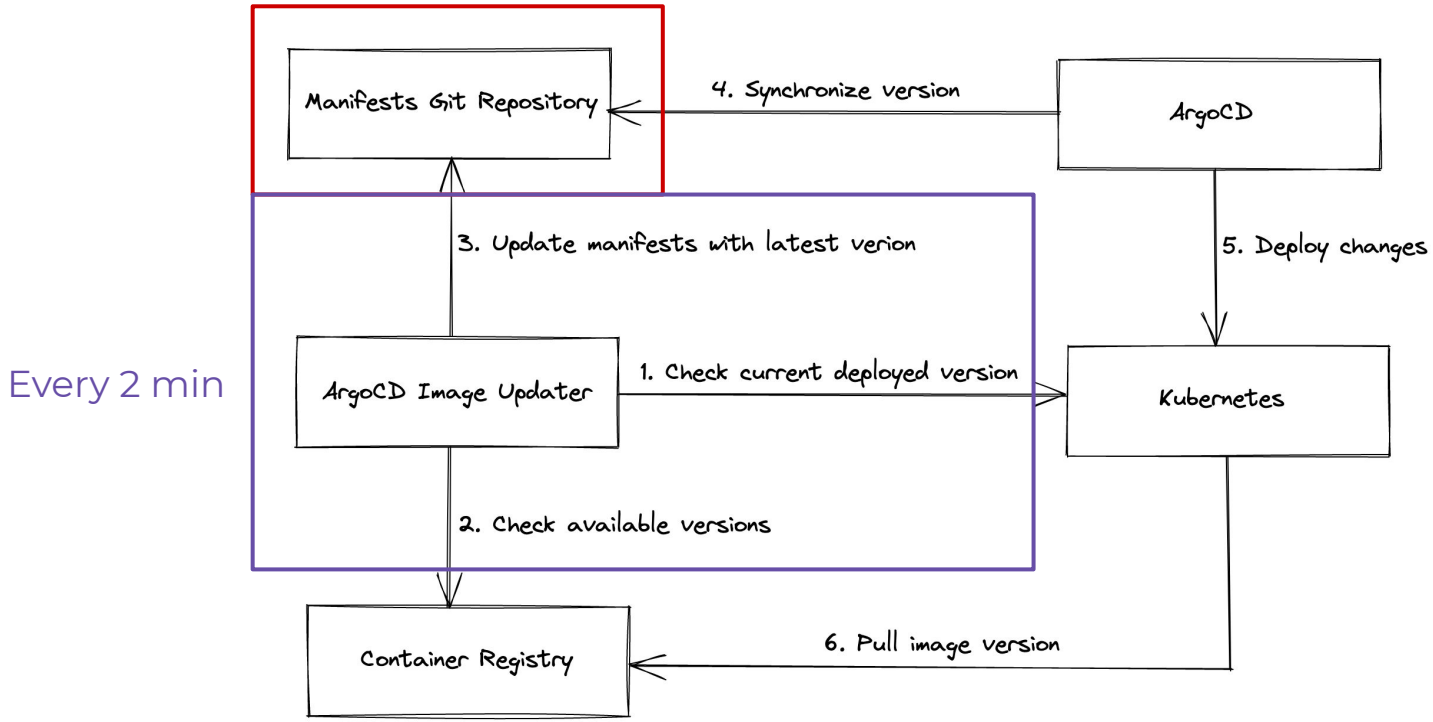


REPRODUCIBILITY: ENVIRONMENT CHART



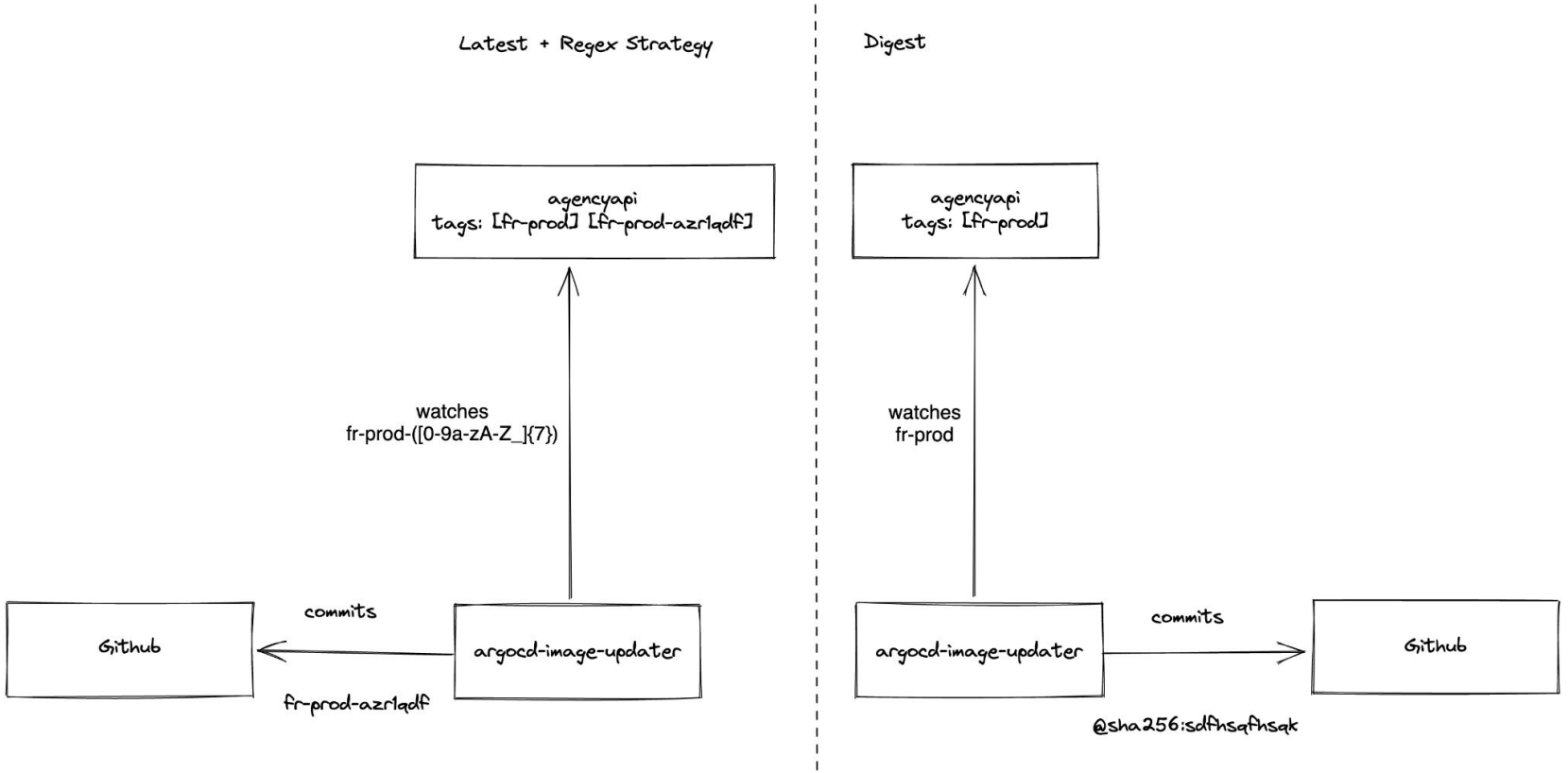
INDEPENDENCE: ARGOCD IMAGE UPDATER

overrideTag principle in every chart




```
image: "{{ .Values.image.repository }}{{- if .Values.image.overrideTag }}:{{ .Values.image.overrideTag }}{{-
  else if .Values.image.shasum }}@{{ .Values.image.shasum }}{{- else }}:{{ .Values.image.tag }}{{- end }}"
```

INDEPENDENCE: ARGOCD IMAGE UPDATER



Limitations and scaling

SHARDING: ARGOCD IMAGE UPDATER

WHY WE HAD TO SHARD IMAGE UPDATER ?

GCR requests limit

Maximum of 10.000 requests per 10 min per IP

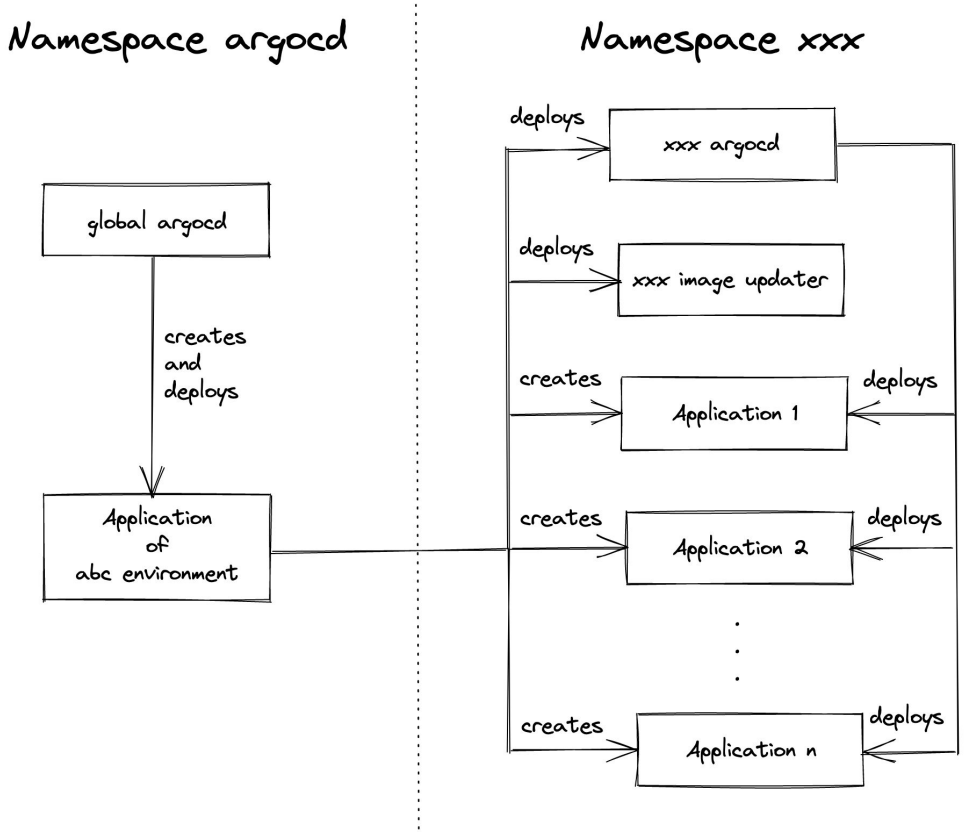
- "digest method uses 3 manifests pulls per tag"
- 11 environments
- More than 50 applications per environment
- Check every 2 min

Multiple strategies

Only one strategy per Docker image

- The key is the image name, not its application

SHARDING: ARGOCD



Improvements

Faster syncs

- Only one branch per Argocd
- Lesser applications per ArgoCD

Tradeoffs

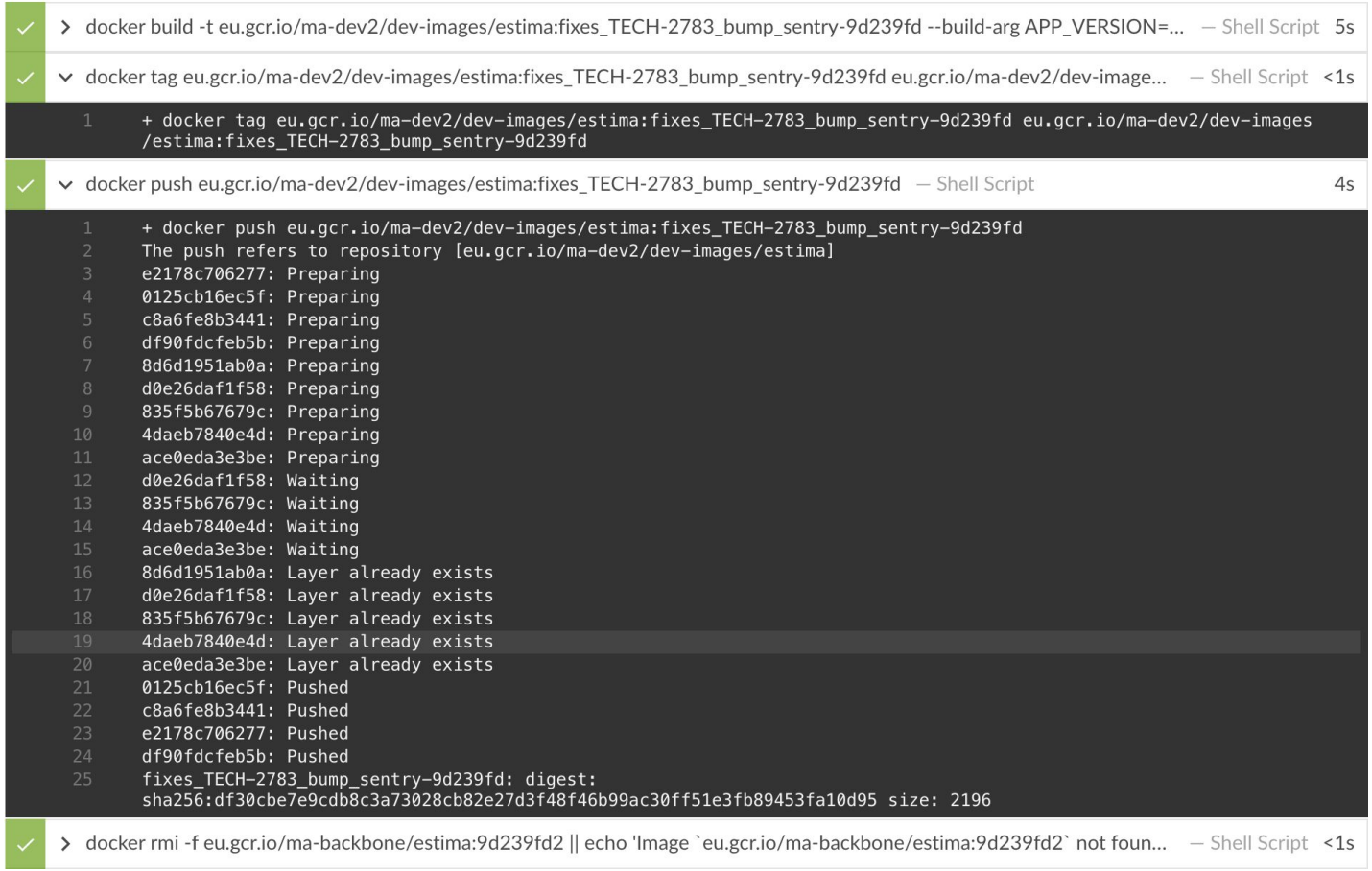
- "xxx" ArgoCD without webhook
- No OAuth2 proxy for "xxx" ArgoCD

Usage and tradeoffs

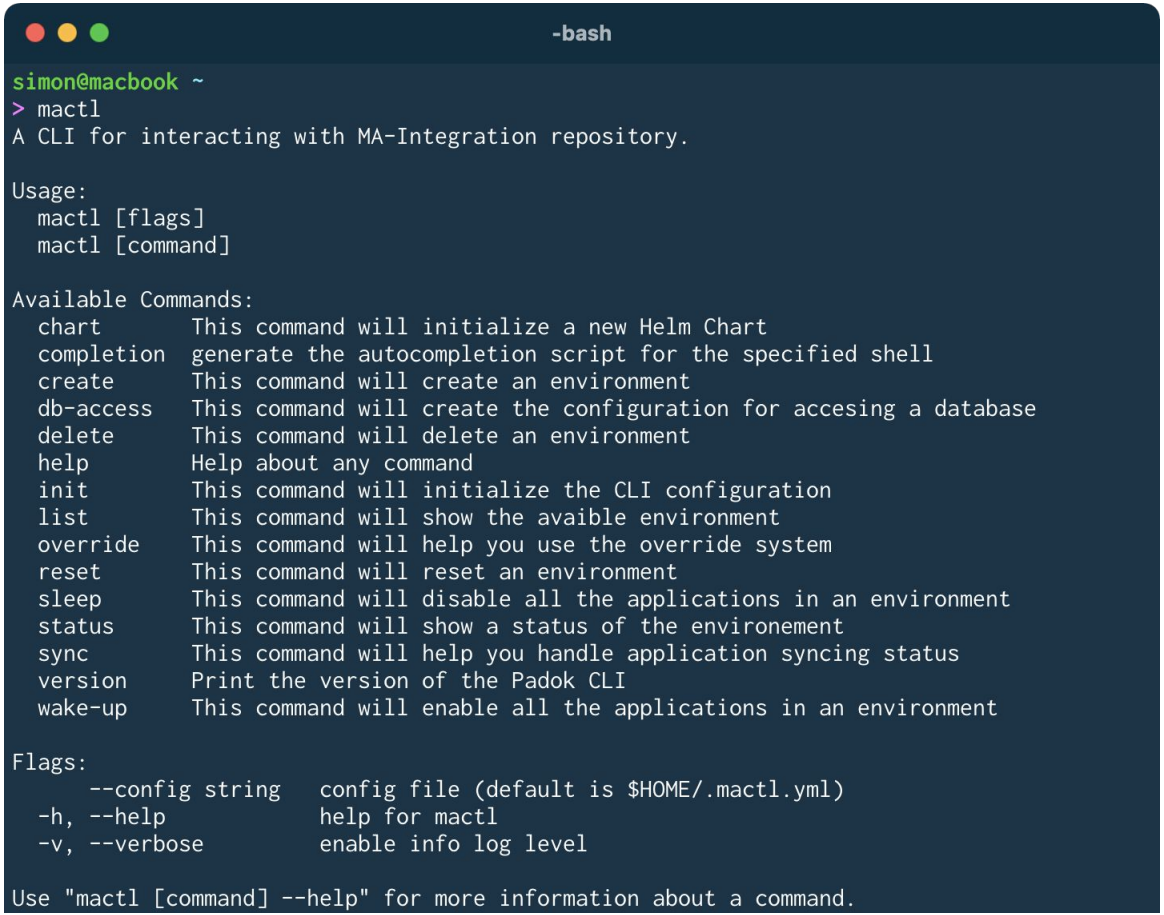
Development workflow

- Web dev launches application locally with docker-compose
- Secrets and configuration retrieved with **berglas** and GCP secret manager
- Consumed APIs are on the cluster requested with team domain name
- Similar make commands in all applications to push a docker tag
- Combination of branch name and commit hash to build the docker tag

Jenkins for docker image tag



CLI for web devs : mactl



CLI for web devs : mactl

Status of the team environment

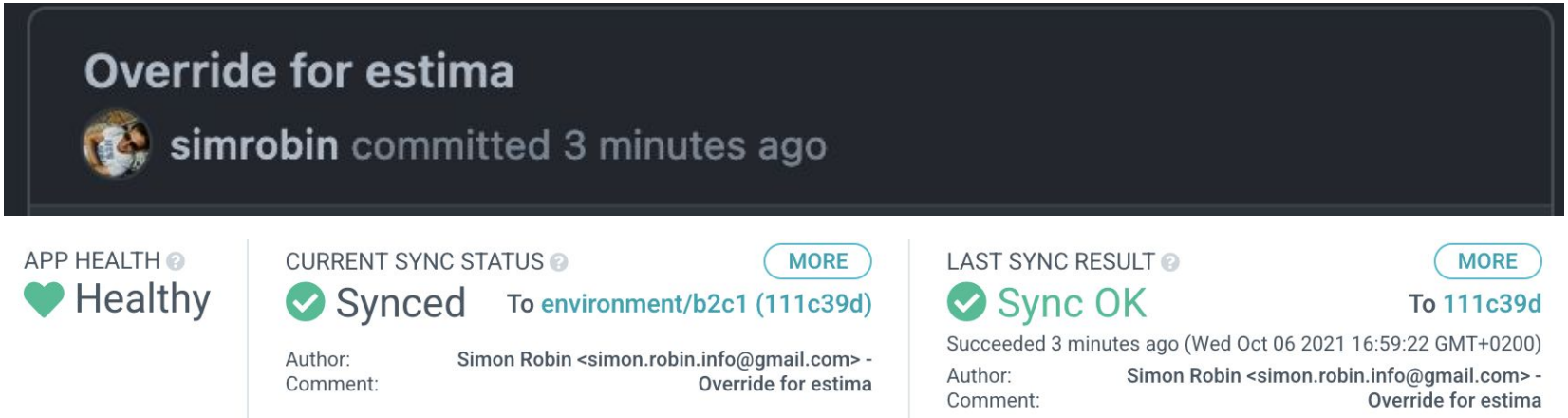
-bash

```
simon@macbook ~
> mactl status
⌛  Checking status of b2c1 environment:

+---------------------+----------------+-----------------------+---------------------------+-------------------+------------------+
|     APPLICATION     | MODIFIED CHART | MODIFIED COMMON VALUES | MODIFIED TEMPLATED VALUES | OVERRIDEN SERVICE | NOT SYNC SERVICE |
+---------------------+----------------+-----------------------+---------------------------+-------------------+------------------+
| partners            |                |                       | X                         |                   |                  |
| cookiebanner        |                |                       |                           | X                 |                  |
| www-next            |                |                       |                           | X                 |                  |
| docs                | X              |                       |                           |                   |                  |
| mediaapi            | X              |                       |                           | X                 |                  |
| barometreapi        |                | X                     |                           |                   |                  |
| topofthelistapi     | X              |                       |                           |                   |                  |
| leadapi             |                | X                     |                           |                   |                  |
| realtorregister     |                | X                     | X                         |                   |                  |
| barometrev2         | X              |                       |                           |                   |                  |
| externaladmanagerapi | X              | X                     |                           |                   |                  |
| malegacy            | X              | X                     |                           |                   |                  |
+---------------------+----------------+-----------------------+---------------------------+-------------------+------------------+
```

CLI for web devs : mactl

Override a tag for an application in the environment

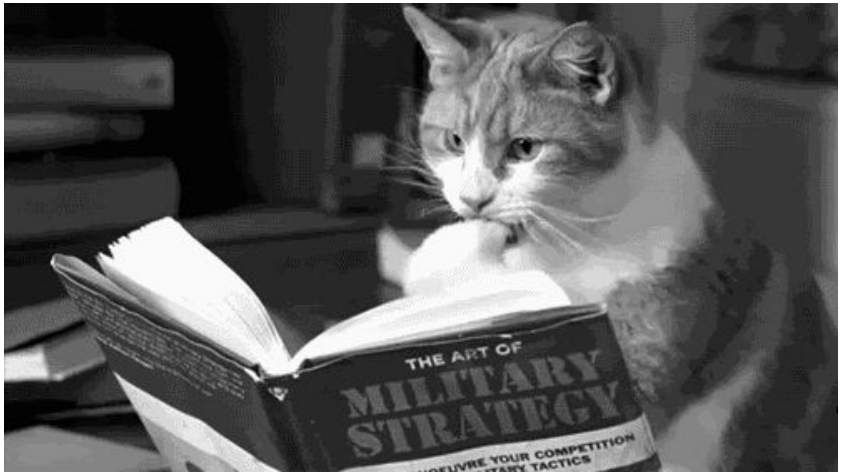


Tradeoffs

- Telepresence to access internal APIs
- An environment per team (vs per developer before)
- Single database for all team environments
- When cluster is down, every team is down

Learnings

- Our legacy apps are not all ready for docker and kubernetes
- Really efficient to have a team (Padok) focused on one subject (integration environment)
- Developers need to be guided with workshops and documentation
- Build a CLI to abstract repetitive and complicated action



What’s next for Meilleurs Agents?

- Migration of our legacy apps to kubernetes in production
- Better understanding of kubernetes compliant app for web developers
- Try to figure out if this architecture is usable in a production environment
- Continue to ask help from Padok for future projects

Thanks