



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

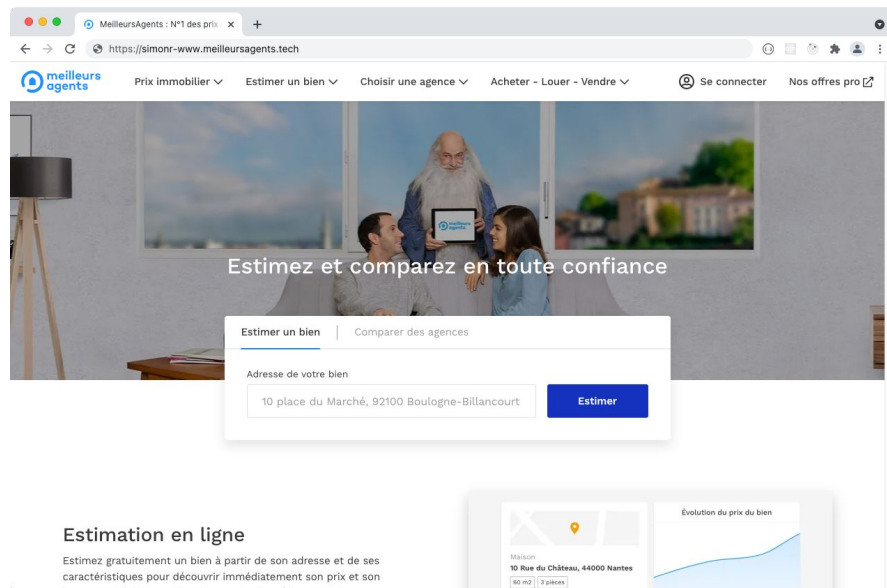
- Cloud migration, security and run
- 50 people, 30 SREs
- We are hiring!



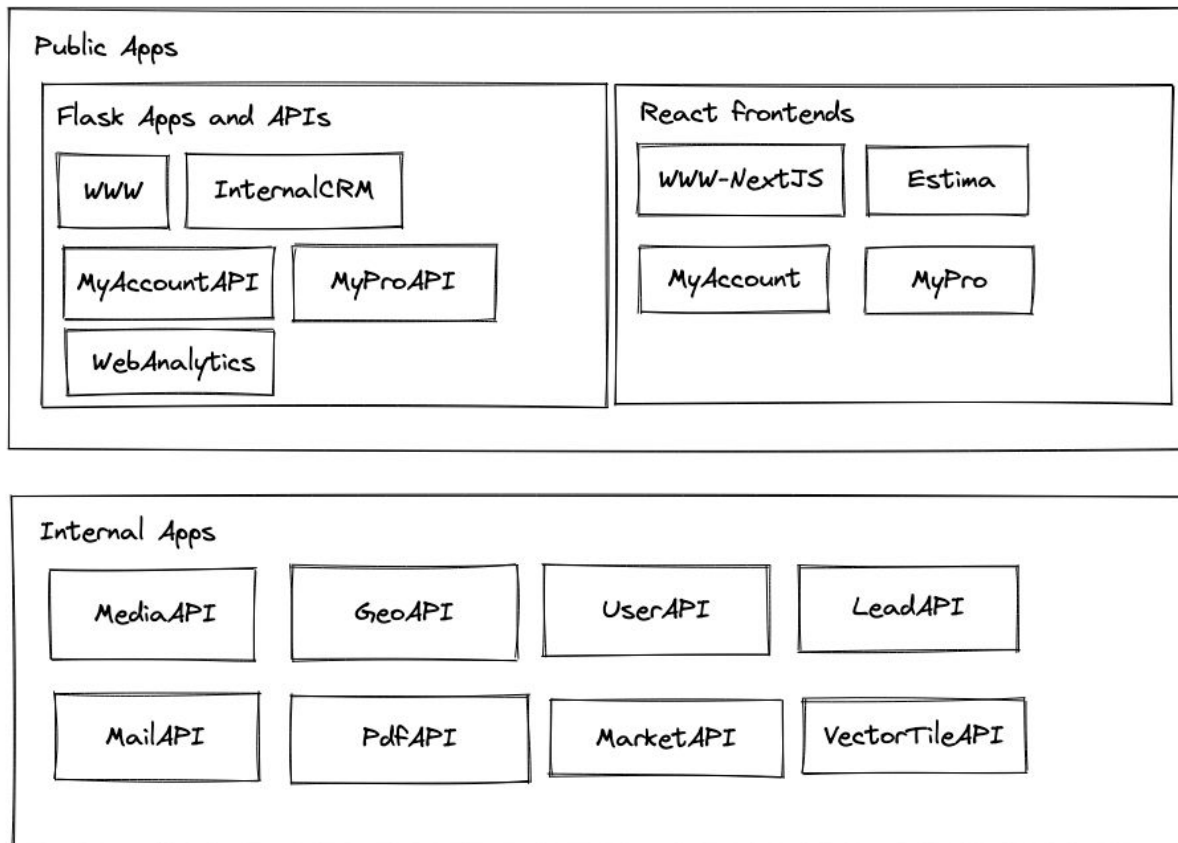
The environment before the project

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

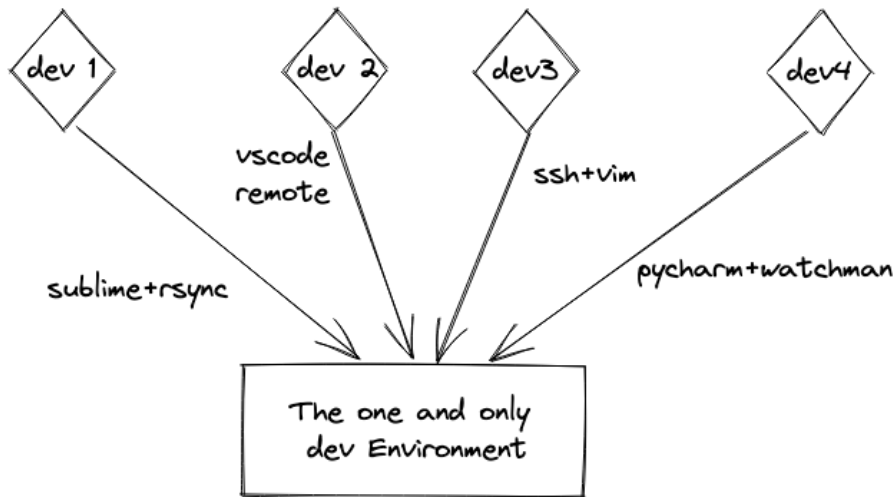


Meilleurs Agents web platform (simplified)



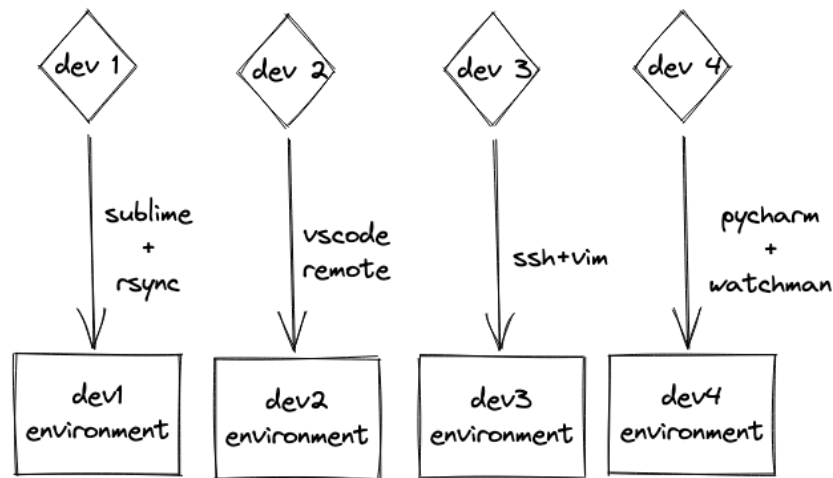
The beginning

- **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

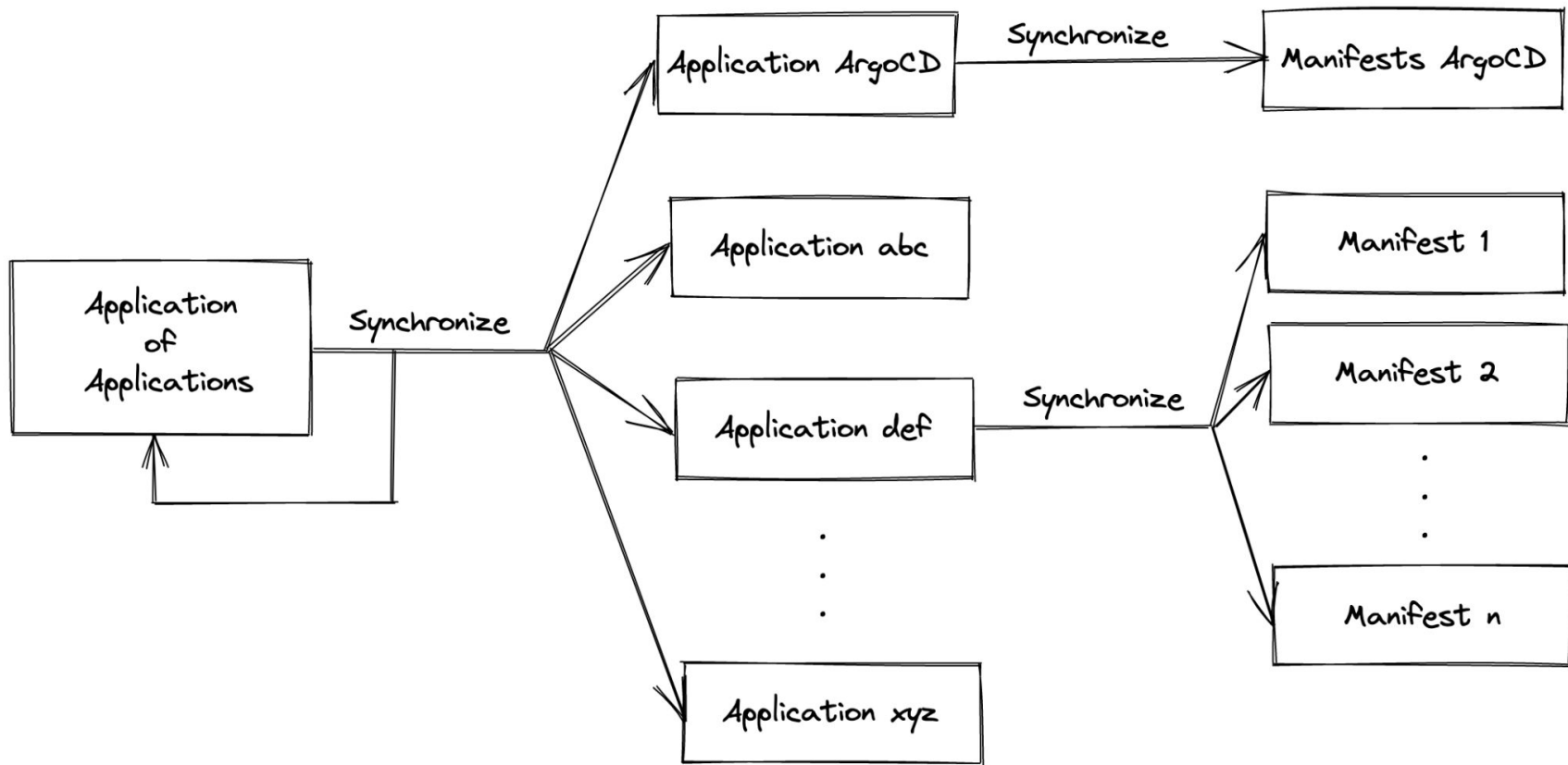
A full environment must be easily reproducible

Creating a new environment must take less than 5 min

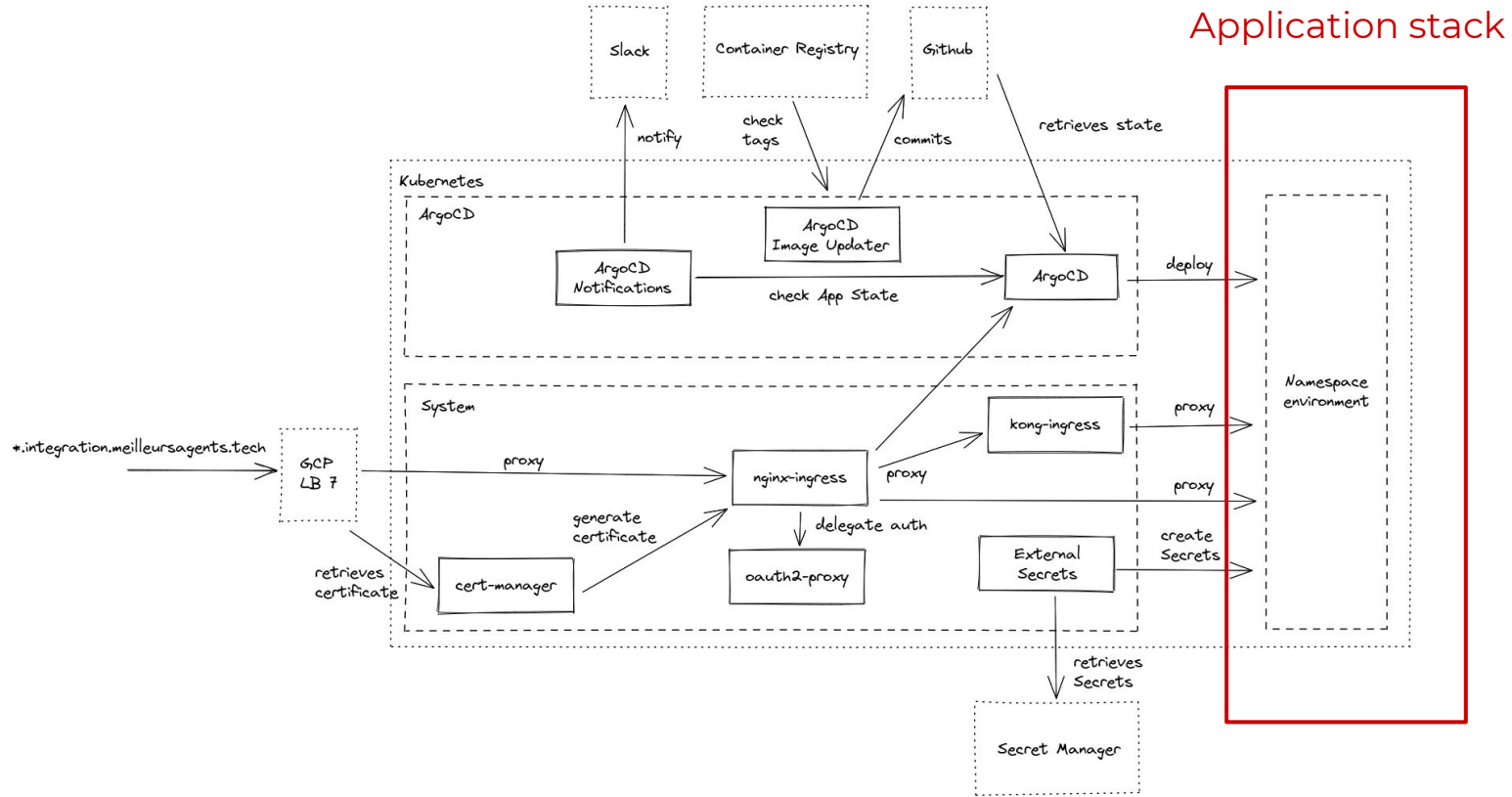
Integrate local development with the new infrastructure

The new infrastructure

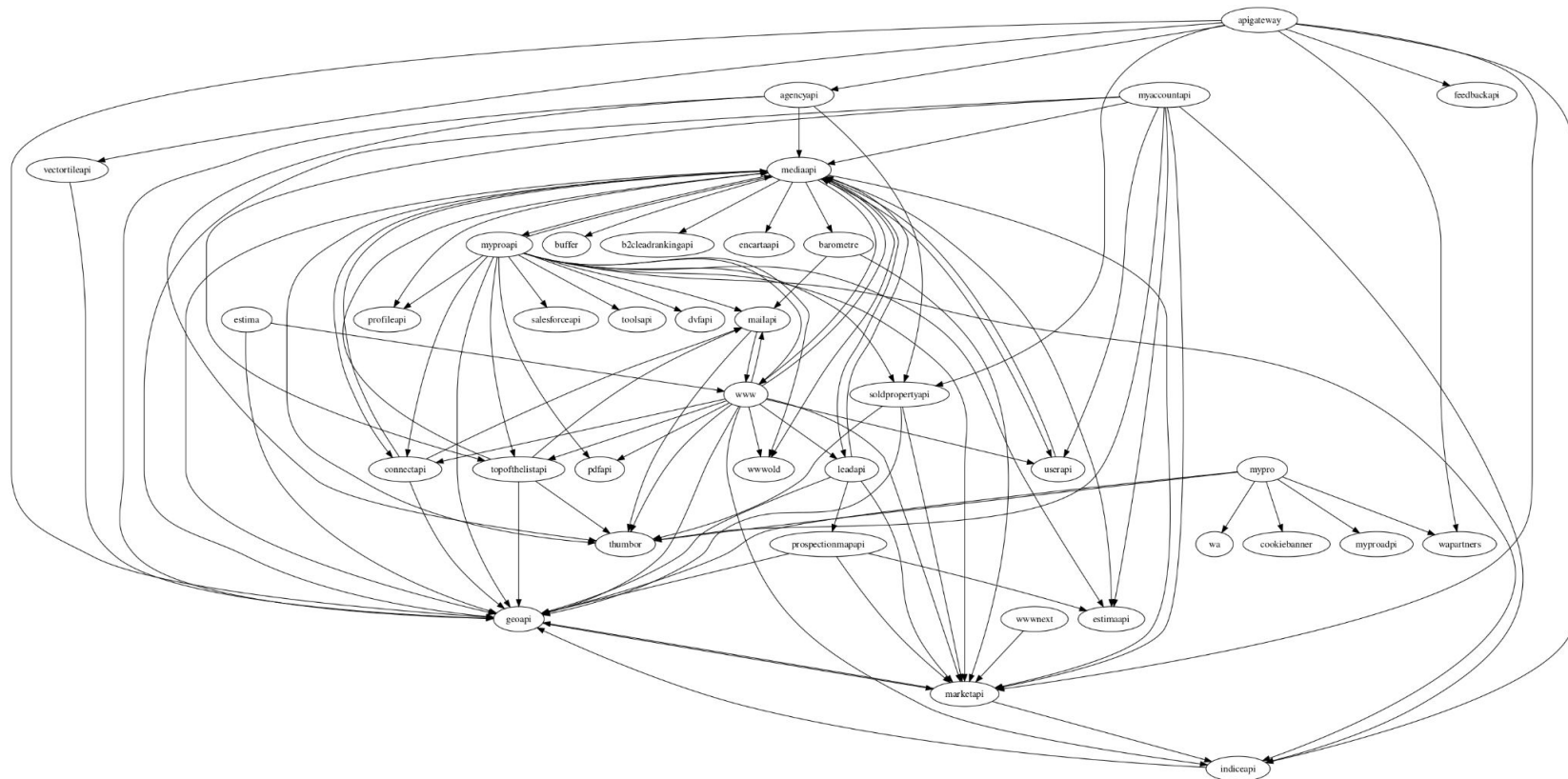
ARGOCD: APP OF APPS PATTERN



GLOBAL ARCHITECTURE



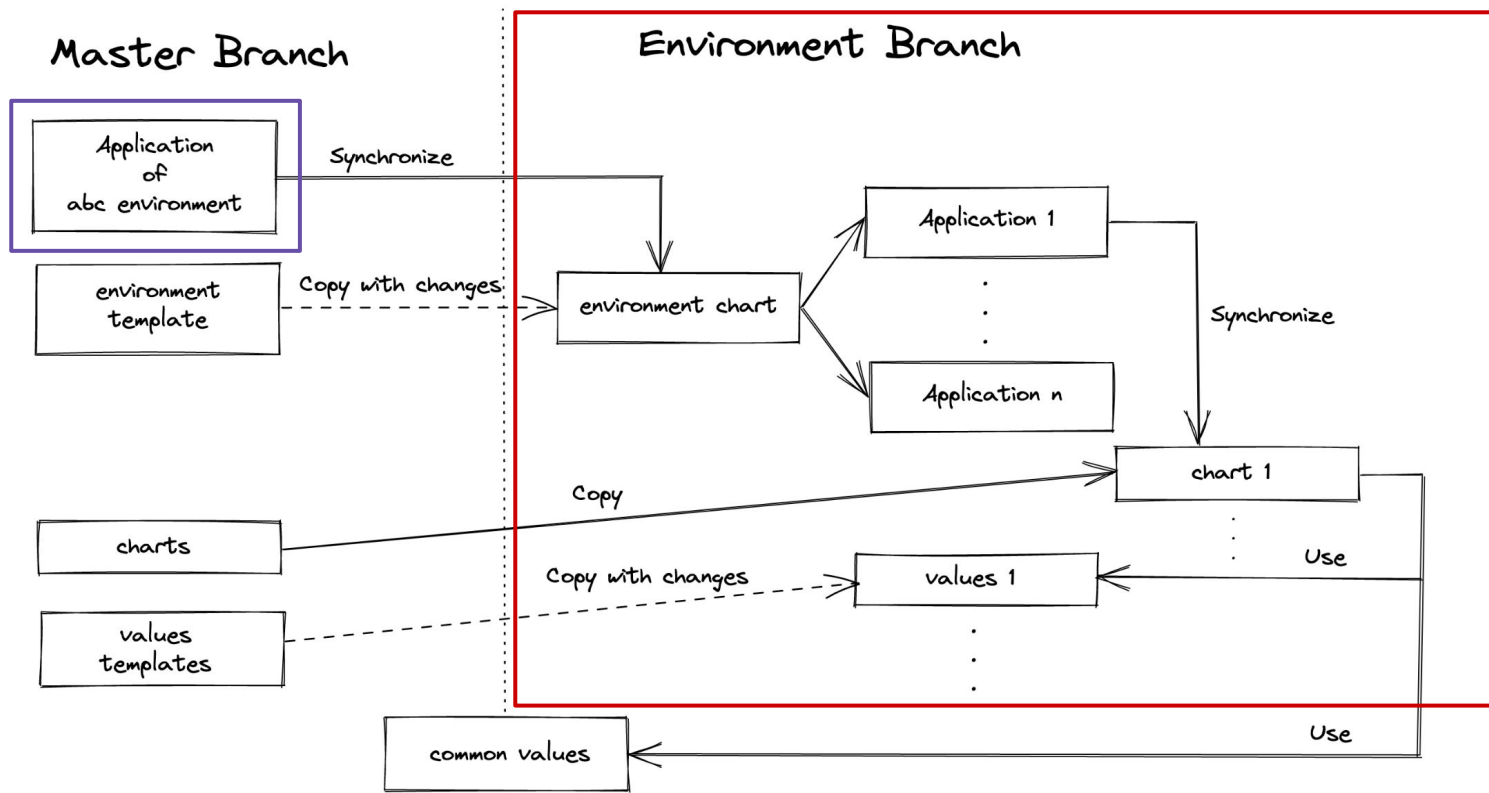
A quick overview of the application dependencies



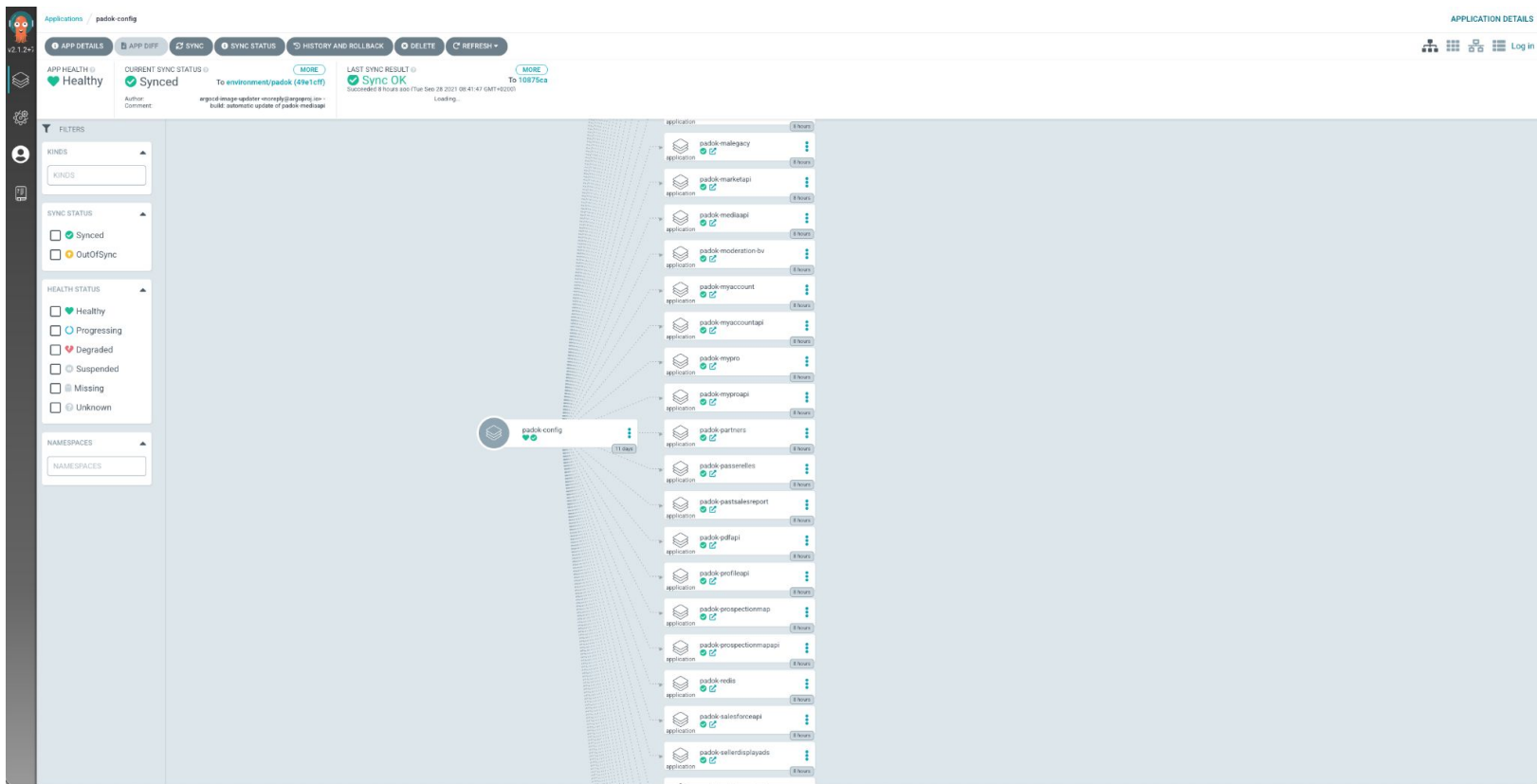
REPRODUCIBILITY: ENVIRONMENT CHART

Generated with Github Action

Generated with script

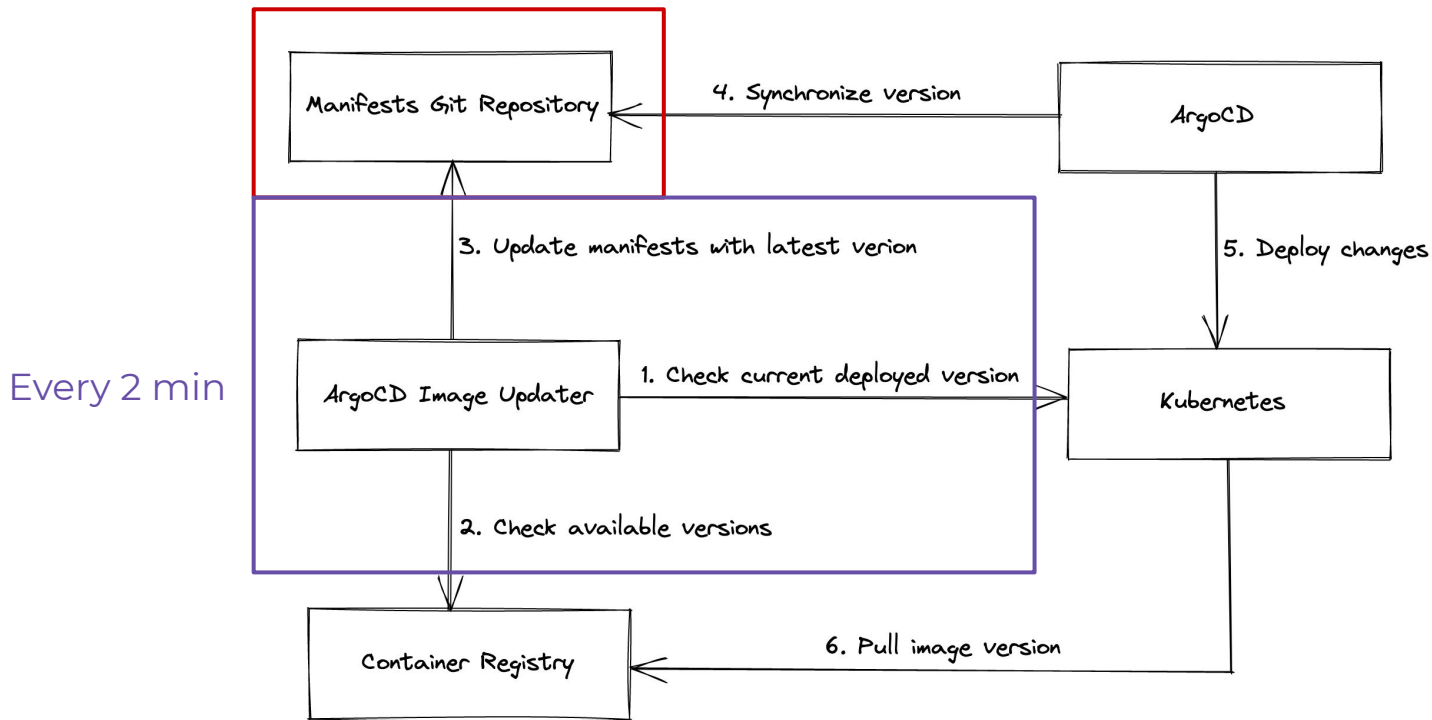


REPRODUCIBILITY: ENVIRONMENT CHART



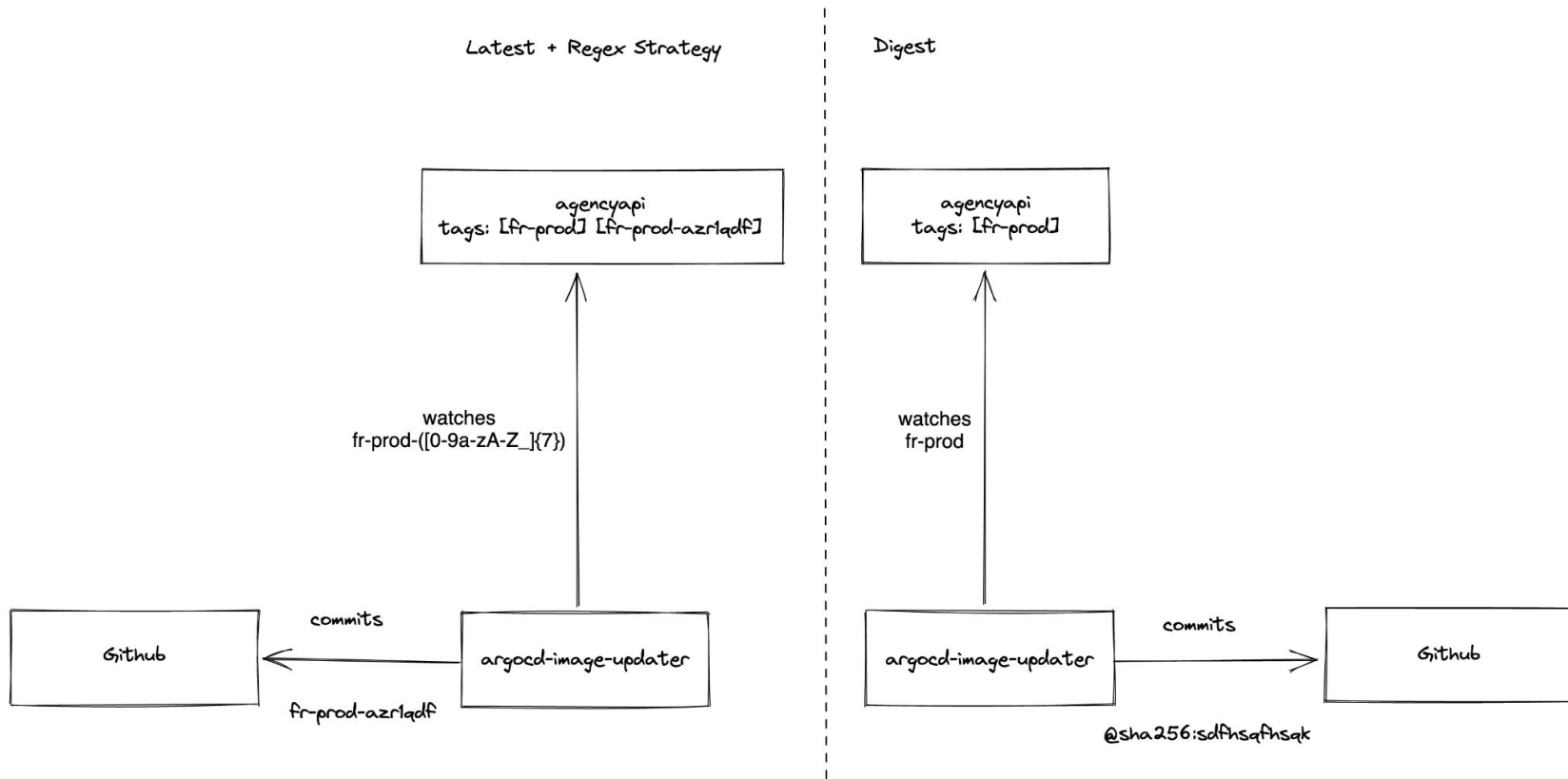
INDEPENDENCE: ARGOCD IMAGE UPDATER

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

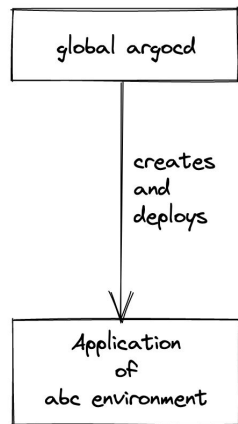
Multiple strategies

Only one strategy per Docker image

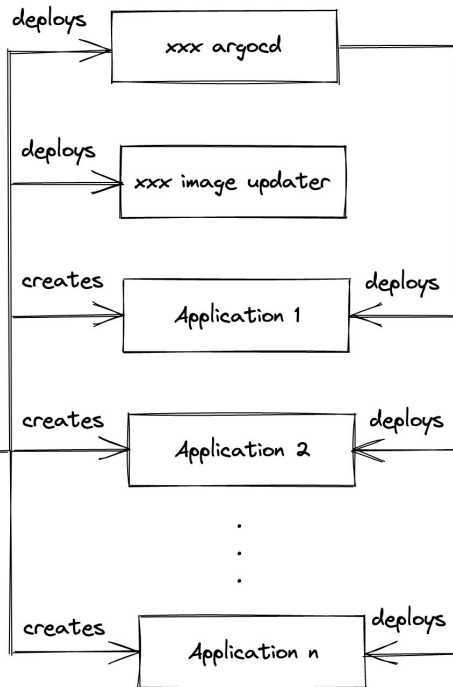
- "digest method uses 3 manifests pulls per tag"
- 11 environments
- More than 50 applications per environment
- Check every 2 min
- The key is the image name, not its application

SHARDING: ARGOCD

Namespace argocd



Namespace xxx



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

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top left corner. The title bar reads "-bash". The terminal shows a user named "simon" on a "macbook" in the directory "~/code/ma/MyAccount" on the "master" branch. The user runs "make generate-config" and then "./config/install.sh". The script prompts for a team name, and the user enters "b2c2".

```
-bash
simon@macbook ~/code/ma/MyAccount (master)
> make generate-config
./config/install.sh

> What is the name of your team? (b2c1,b2b1,webtech1...) (default: b2c1): b2c2
simon@macbook ~/code/ma/MyAccount (master)
> █
```

Jenkins for docker image tag

```
✓ > docker build -t eu.gcr.io/ma-dev2/dev-images/estima:fixes_TECH-2783_bump_sentry-9d239fd --build-arg APP_VERSION=... -- Shell Script 5s

✓ ✓ docker tag eu.gcr.io/ma-dev2/dev-images/estima:fixes_TECH-2783_bump_sentry-9d239fd eu.gcr.io/ma-dev2/dev-image... -- Shell Script <1s

1 + docker tag eu.gcr.io/ma-dev2/dev-images/estima:fixes_TECH-2783_bump_sentry-9d239fd eu.gcr.io/ma-dev2/dev-images
  /estima:fixes_TECH-2783_bump_sentry-9d239fd

✓ ✓ docker push eu.gcr.io/ma-dev2/dev-images/estima:fixes_TECH-2783_bump_sentry-9d239fd -- Shell Script 4s

1 + docker push eu.gcr.io/ma-dev2/dev-images/estima:fixes_TECH-2783_bump_sentry-9d239fd
2 The push refers to repository [eu.gcr.io/ma-dev2/dev-images/estima]
3 e2178c706277: Preparing
4 0125cb16ec5f: Preparing
5 c8a6fe8b3441: Preparing
6 df90fdcfeb5b: Preparing
7 8d6d1951ab0a: Preparing
8 d0e26daf1f58: Preparing
9 835f5b67679c: Preparing
10 4daeb7840e4d: Preparing
11 ace0eda3e3be: Preparing
12 d0e26daf1f58: Waiting
13 835f5b67679c: Waiting
14 4daeb7840e4d: Waiting
15 ace0eda3e3be: Waiting
16 8d6d1951ab0a: Layer already exists
17 d0e26daf1f58: Layer already exists
18 835f5b67679c: Layer already exists
19 4daeb7840e4d: Layer already exists
20 ace0eda3e3be: Layer already exists
21 0125cb16ec5f: Pushed
22 c8a6fe8b3441: Pushed
23 e2178c706277: Pushed
24 df90fdcfeb5b: Pushed
25 fixes_TECH-2783_bump_sentry-9d239fd: digest:
    sha256:df30cbe7e9c8b8c3a73028cb82e27d3f48f46b99ac30ff51e3fb89453fa10d95 size: 2196

✓ > docker rmi -f eu.gcr.io/ma-backbone/estima:9d239fd2 || echo 'Image `eu.gcr.io/ma-backbone/estima:9d239fd2` not foun... -- Shell Script <1s
```


CLI for web devs : mactl

```
-bash

simon@macbook ~
> mactl
A CLI for interacting with MA-Integration repository.

Usage:
  mactl [flags]
  mactl [command]

Available Commands:
  chart      This command will initialize a new Helm Chart
  completion generate the autocompletion script for the specified shell
  create     This command will create an environment
  db-access  This command will create the configuration for accesing a database
  delete     This command will delete an environment
  help       Help about any command
  init       This command will initialize the CLI configuration
  list       This command will show the avaible environment
  override   This command will help you use the override system
  reset      This command will reset an environment
  sleep      This command will disable all the applications in an environment
  status     This command will show a status of the environement
  sync       This command will help you handle application syncing status
  version    Print the version of the Padok CLI
  wake-up    This command will enable all the applications in an environment

Flags:
  --config string  config file (default is $HOME/.mactl.yml)
  -h, --help       help for mactl
  -v, --verbose    enable info log level

Use "mactl [command] --help" for more information about a command.
```

CLI for web devs : mactl

Status of the team environment

```
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

```
-bash

simon@macbook ~
> mactl override add estima --overrideTag fixes_TECH-2783_bump_sentry-9d239fd
✓ Application successfully overridden!simon@macbook ~
> █
```

Override for estima

 **simrobin** committed 3 minutes ago

APP HEALTH ?

♥ Healthy

CURRENT SYNC STATUS ?

MORE

✓ Synced To [environment/b2c1 \(111c39d\)](#)

Author: Simon Robin <simon.robin.info@gmail.com> -
Comment: Override for estima

LAST SYNC RESULT ?

MORE

✓ Sync OK To [111c39d](#)

Succeeded 3 minutes ago (Wed Oct 06 2021 16:59:22 GMT+0200)
Author: Simon Robin <simon.robin.info@gmail.com> -
Comment: Override for estima

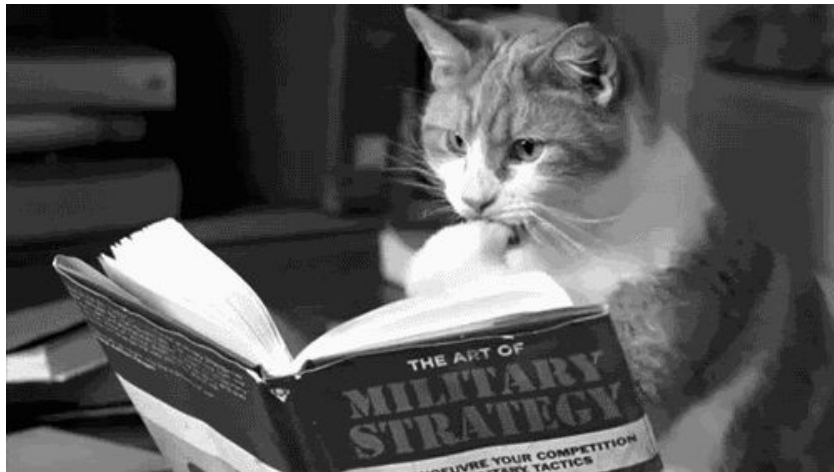
Tradeoffs

- 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
