# Making the right abstractions

when building a platform

LUNAR®

# About the speaker

Name : **Henrik Høegh**

**Platform engineer** at **Lunar**

Part of **Squad Odyssey**

Interests :

- **Cloud Native**, **Dungeon & Dragons**

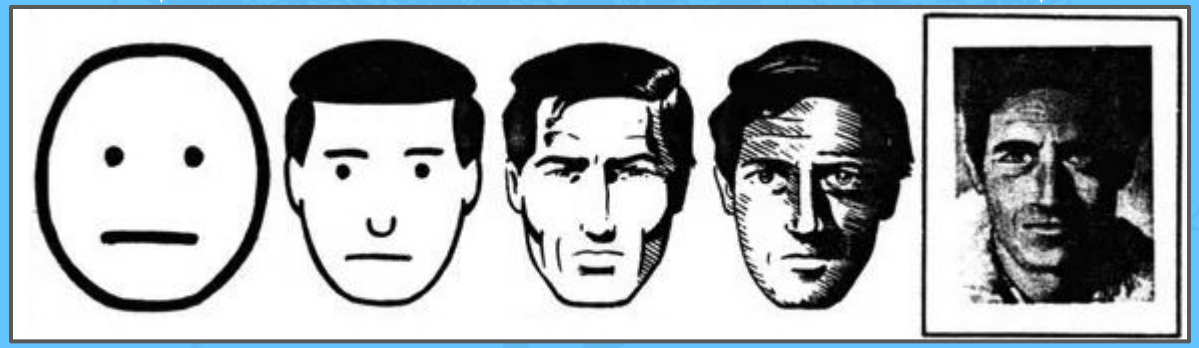
# So... what is an abstraction

LUNAR®

# Definition of *abstract*

Great for pictionary

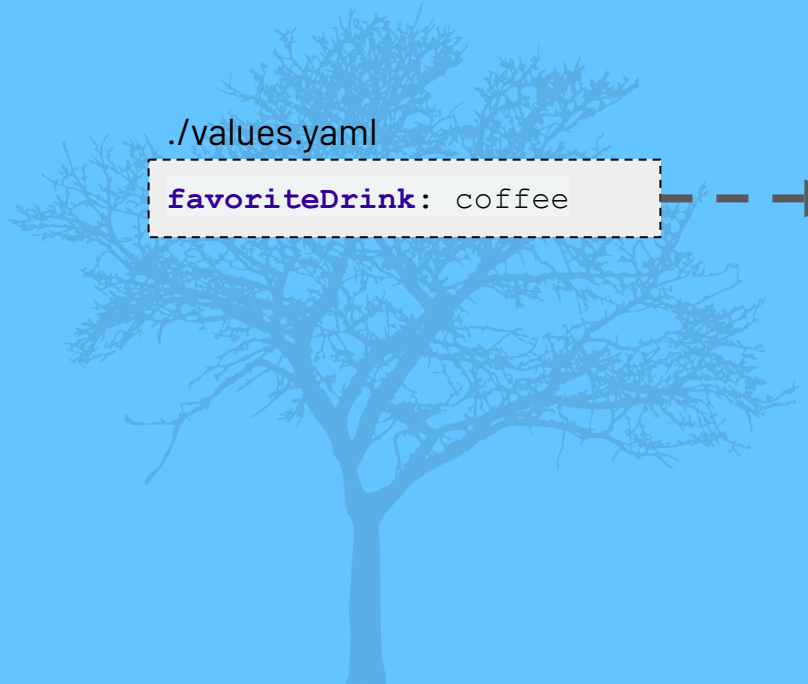
Great for police sketching



A face

Also a face

# Helm is abstracting away details



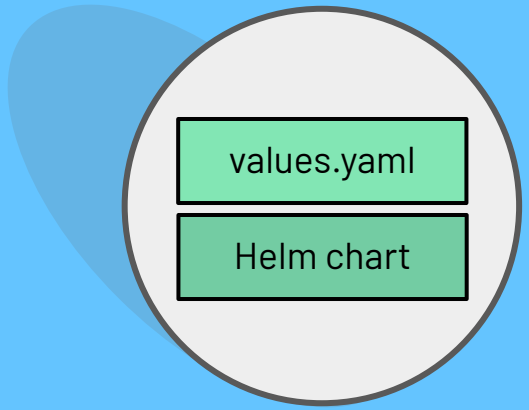
./values.yaml

```
favoriteDrink: coffee
```

templates/configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
 name: {{ .Release.Name }}-configmap
data:
 drink: {{ .Values.favoriteDrink }}
```

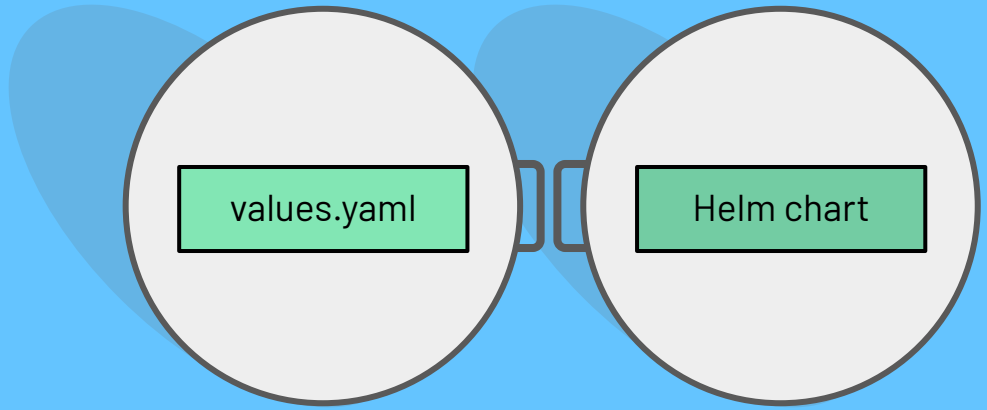
# But only if....




**Dev**

Nothing is abstracted **away**, you've only added complexity

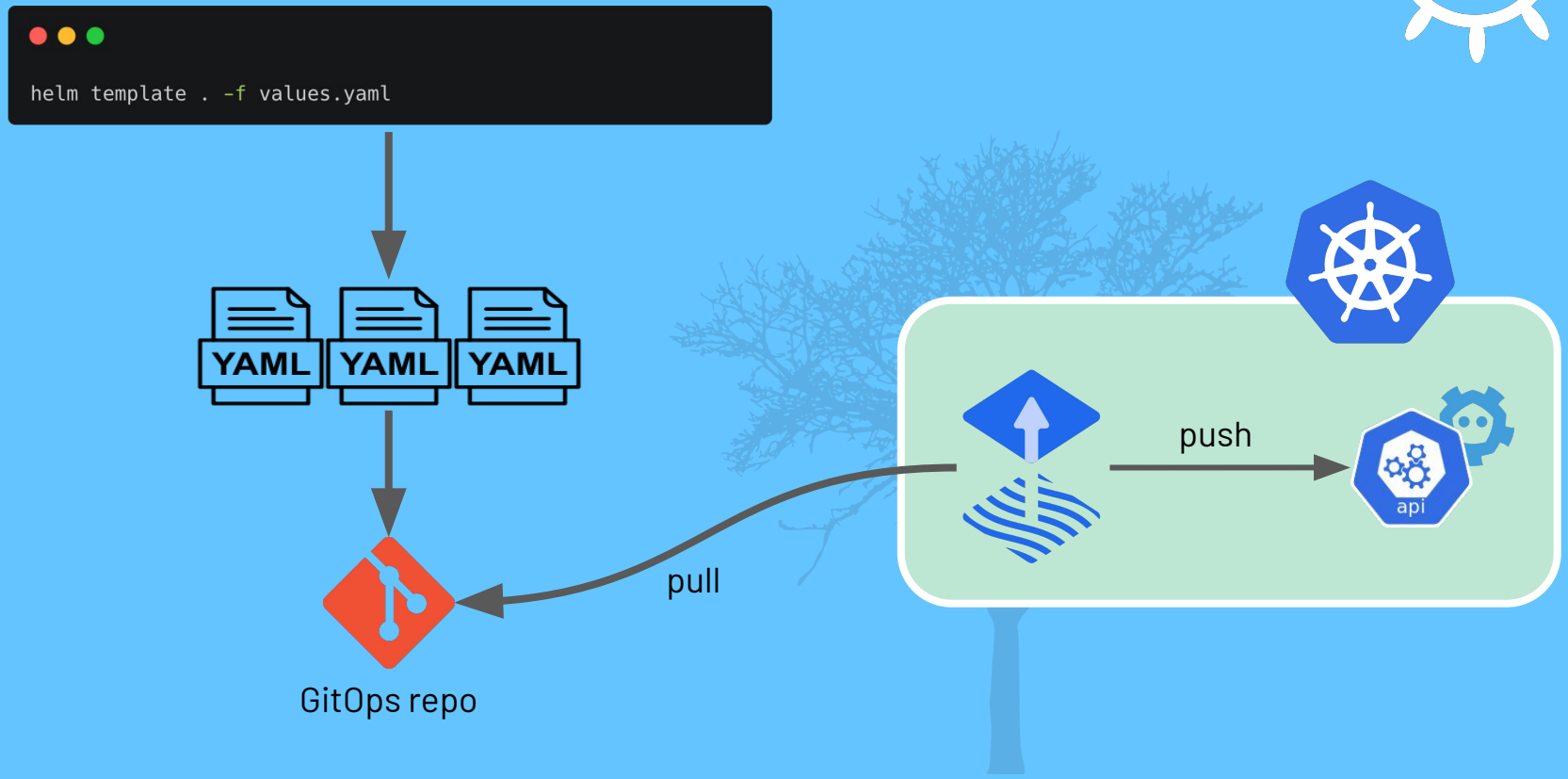
**VS**



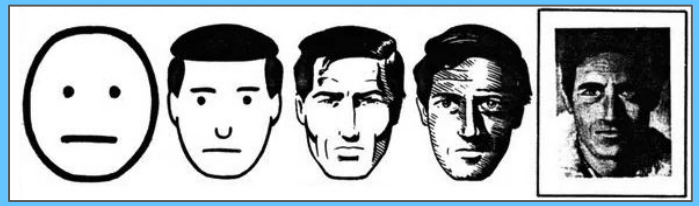

**Dev** **Platform**

Details not important to Dev is taken care of by the Platform team

# Using Helm with GitOps



# Desired state != Actual state



High level

Low level

Helm values.yml

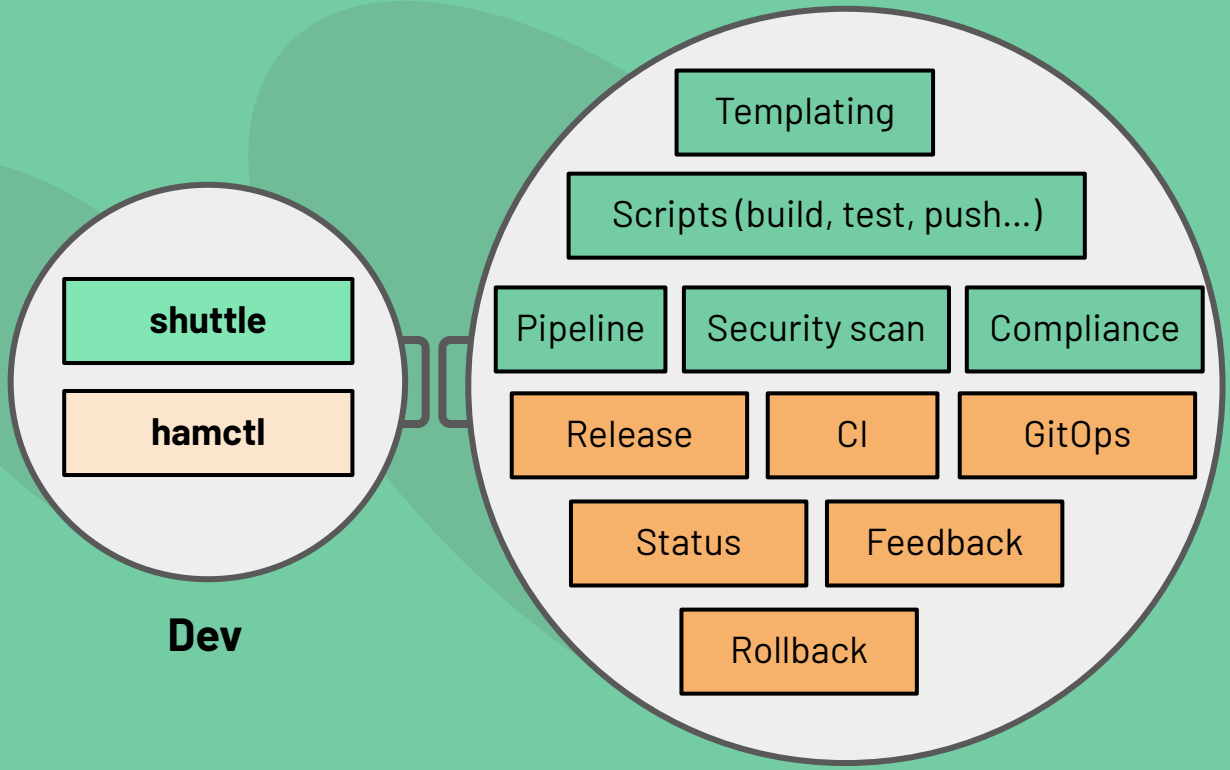
Desired state

Actual state

Thanks to

kosli.com

# At Lunar we abstract the complex parts away




shuttle

Release Manager

# Developer feedback from Ham

slack

# And we have a lot of other developer tools

humio

Grafana

# Paved path vs opt-out

www.linkedin.com/in/hoeghh/
www.twitter.com/HenrikHoegh/

# THANK YOU

**Shuttle**: github.com/lunarway/shuttle
**Release-manager**: github.com/lunarway/release-manager
**Lunar Tech insights :** https://tech.lunar.app/