



Marc-Philipp Knechtle

## Going public

Tips for publishing your own code

# Table of contents

- 1 Hide your secrets!
- 2 Creating a healthy community
- 3 Consistency with Continuous Integration
- 4 Documentation
- 5 Promote your code

# Sensitive data in git

- Problem: (Accidental) commits of secrets
- e.g. passwords, application secret keys, OAuth secret keys
- The file is still accessible in history, even if the original file was deleted
- Security risk, even in private repositories

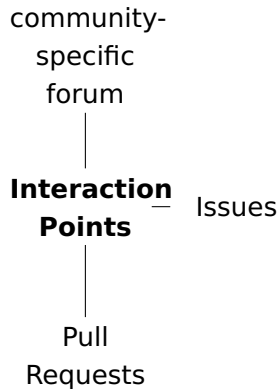
## Removing sensitive data

This slide can show the process of removing the private data if a practical exercise is desired

# Encrypt sensitive data

- github encrypted secrets
- git-secret
- technology specific encryption tool (e.g. ansible-vault)

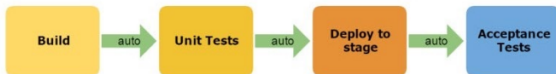
# Points of interaction with the community



# Code of Conduct

- Goal: healthy, constructive community behaviour
- CODE\_OF\_CONDUCT file
- e.g. Contributor Covenant <sup>1</sup>, Django Code of Conduct <sup>2</sup>

### Continuous Integration



### Continuous Delivery



### Continuous Deployment





# Documentation

This was included in my task description, but is also topic of the work of @Nico.  
Should I include this in my presentation?

# Ideas for promoting your code

■ asdf

# References