March 2019

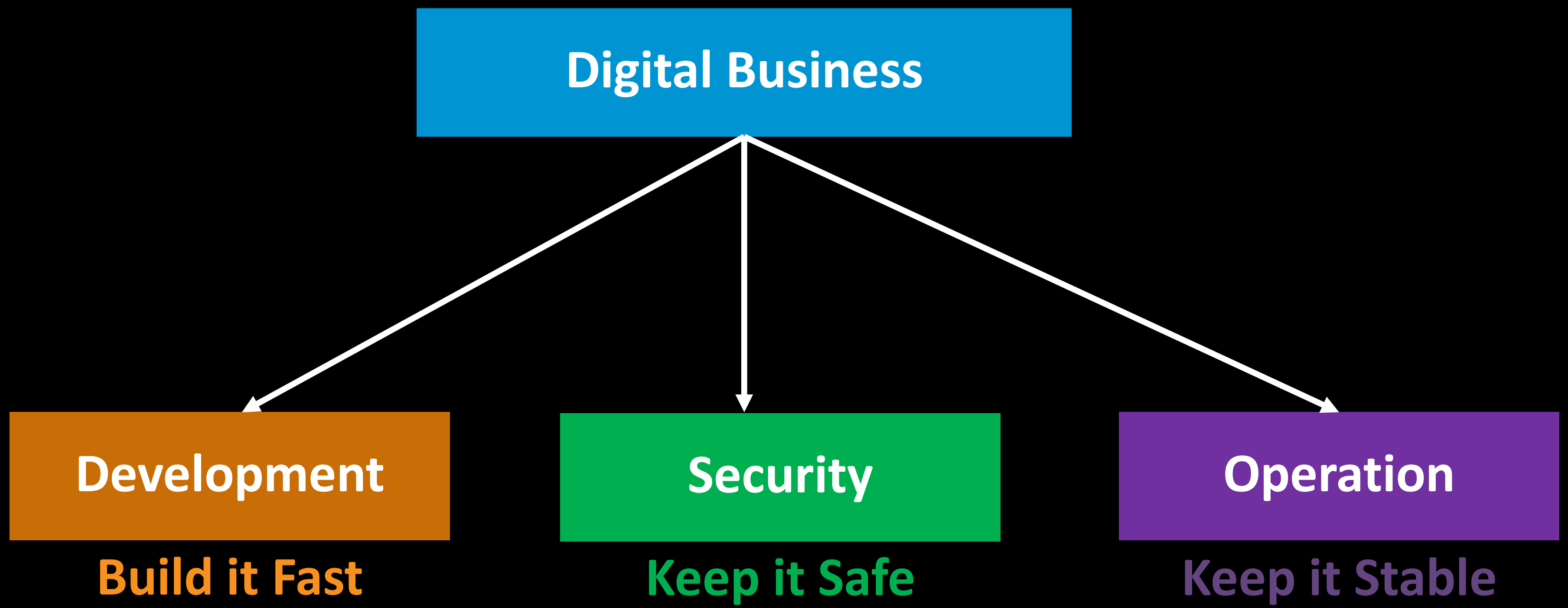
# Security in an Agile/DevOps environment

**PRESENTED BY:**

Shain Singh, Security Architect [APCJ]
ss@f5.com

WE MAKE APPS GO FASTER. SMARTER. SAFER.

# The Collaborating Teams



Digital Business

Development

Build it Fast

Security

Keep it Safe

Operation

Keep it Stable

# The Challenge

SPEED [✓ YES] NO

QUALITY [✓ YES] NO

SECURE YES [✓ NO]

## The Challenge

**SPEED** YES [✓ NO]

**QUALITY** [✓ YES] NO

**SECURE** [✓ YES] NO

# The Challenge

|  |  |  |
|---|---|---|
| SPEED | [✓ YES] | NO |
| QUALITY | YES | [✓ NO] |
| SECURE | [✓ YES] | NO |

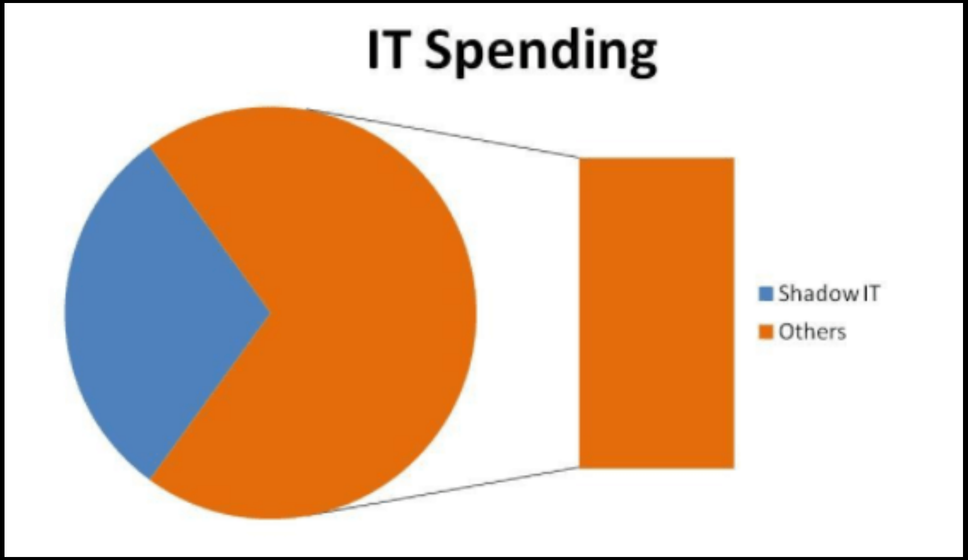
# The Challenge

CONFLICT

# Everyone Brings Best to The Table

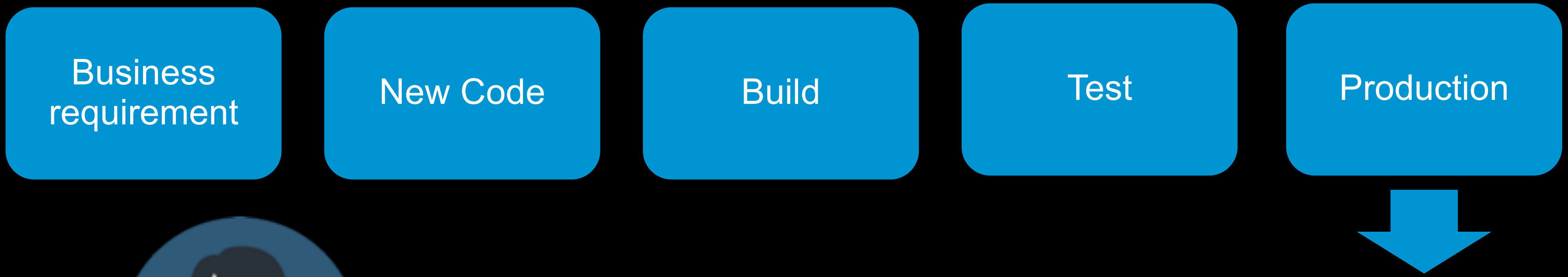
# App teams are running (away from IT) ..

**According to Gartner Shadow IT spending accounts for 30% of IT spending**



https://en.wikipedia.org/wiki/Shadow_IT

# It’s all about speed



IT Security (Traditional)

- Something got blocked - Oops Human Error - forgot to update the policy on the second data center
- Are you sure webscraping is blocking google? I’ll check it tomorrow I’m super busy..
- Build WAF policy – I’m going to learn all URLs and protect from web scraping

Business Perception: Security is preventing business/breaking the app

# Objectives

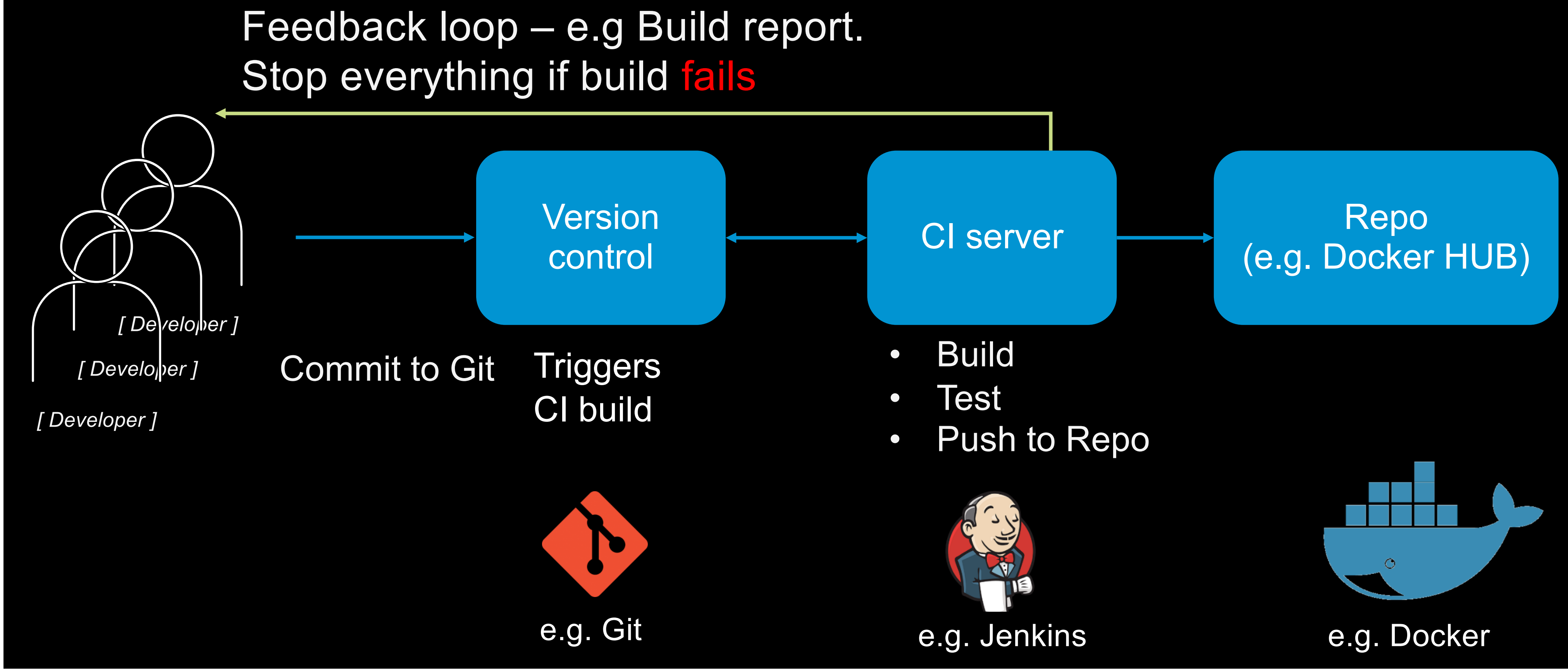
By the end of this class, you will be able to:

- Get an intro to the DevSecOps world terminology
- Handle a conversation on how to integrate F5 security services in an agile application dev workflow
- Describe how F5 security services can be integrated into the CI/D workflow

# An Intro to CI/CD

# Application pipeline

## Continuous integration - CI



Feedback loop – e.g Build report.
Stop everything if build fails

[ Developer ]

[ Developer ]

[ Developer ]

Commit to Git

Version control

Triggers
CI build

CI server

- Build
- Test
- Push to Repo

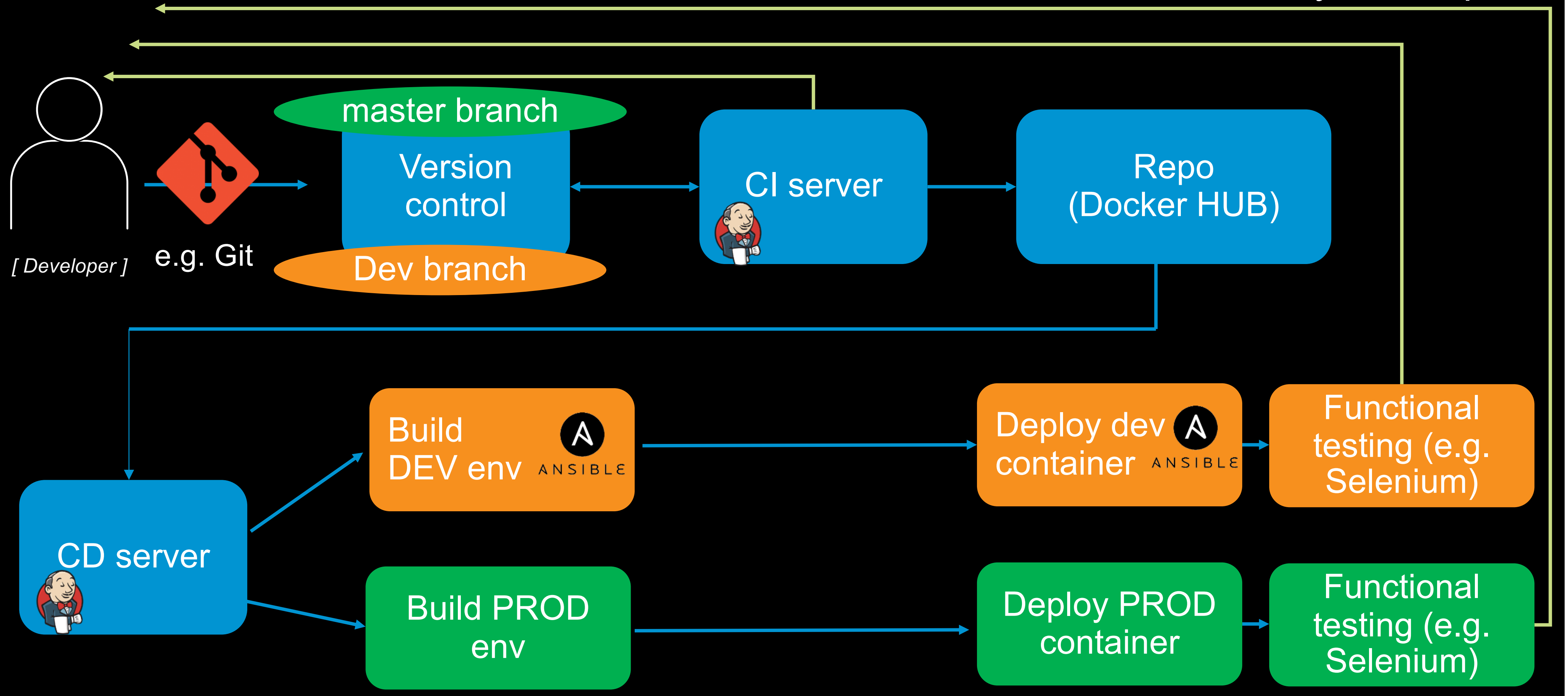
Repo
(e.g. Docker HUB)

e.g. Git

e.g. Jenkins

e.g. Docker

# Application Delivery Pipeline

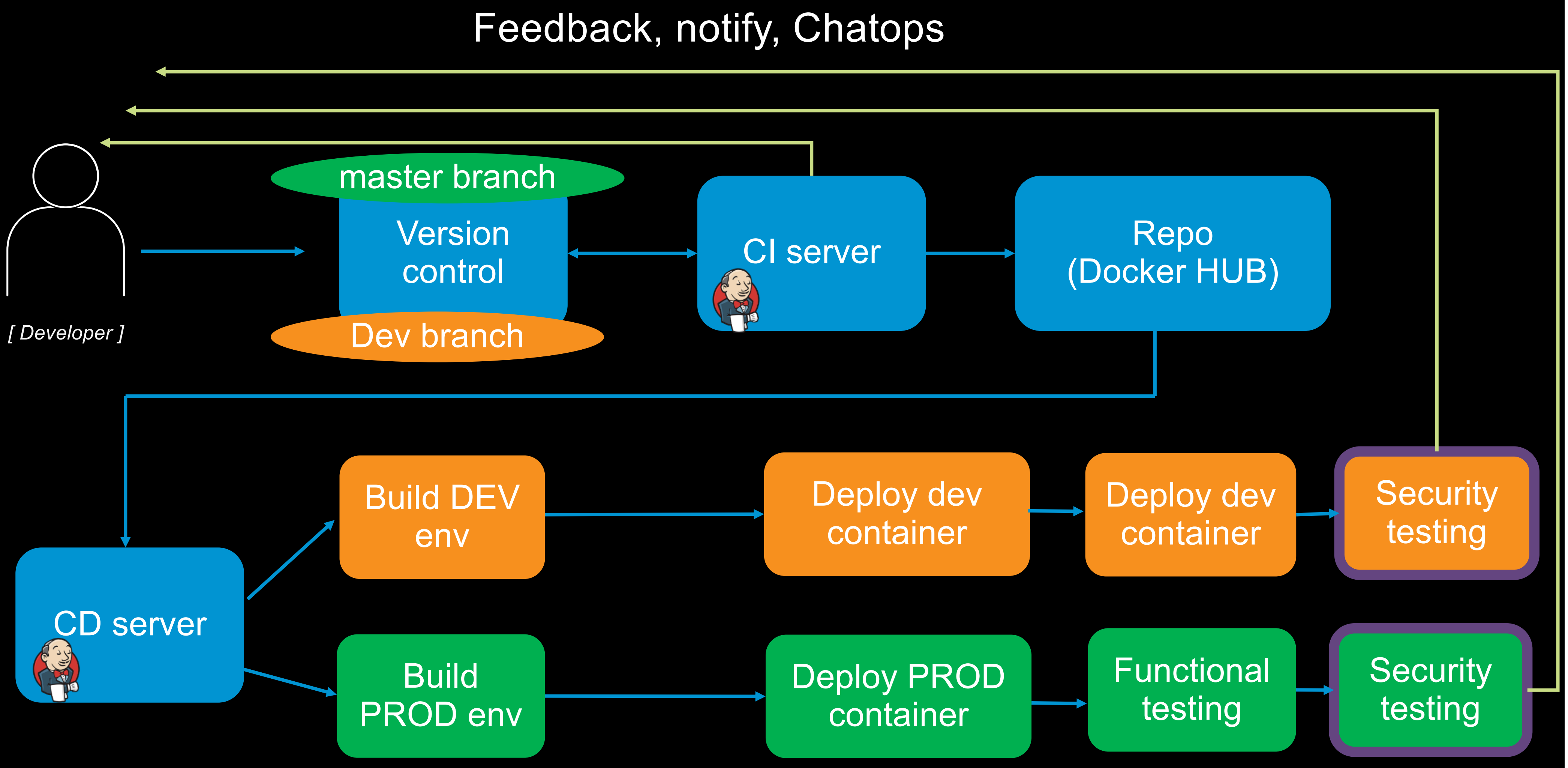


# Continuous delivery - CD

# DevSecOps Principles

- **Culture change** (annoying but still true)
  - Security is a shared responsibility
  - Operating as a team, OWNING security for the product
  - Visible changes
- **Team structure**
  - Developers are in the security group
  - End to end teams. You Build it - you own it
- **Introduce security** as early as possible in the dev life cycle

https://www.safaribooksonline.com/library/view/agile-application-security/9781491938836/



Feedback, notify, Chatops

[ Developer ]

master branch

Version control

Dev branch

CI server

Repo (Docker HUB)

CD server

Build DEV env

Deploy dev container

Deploy dev container

Security testing

Build PROD env

Deploy PROD container

Functional testing

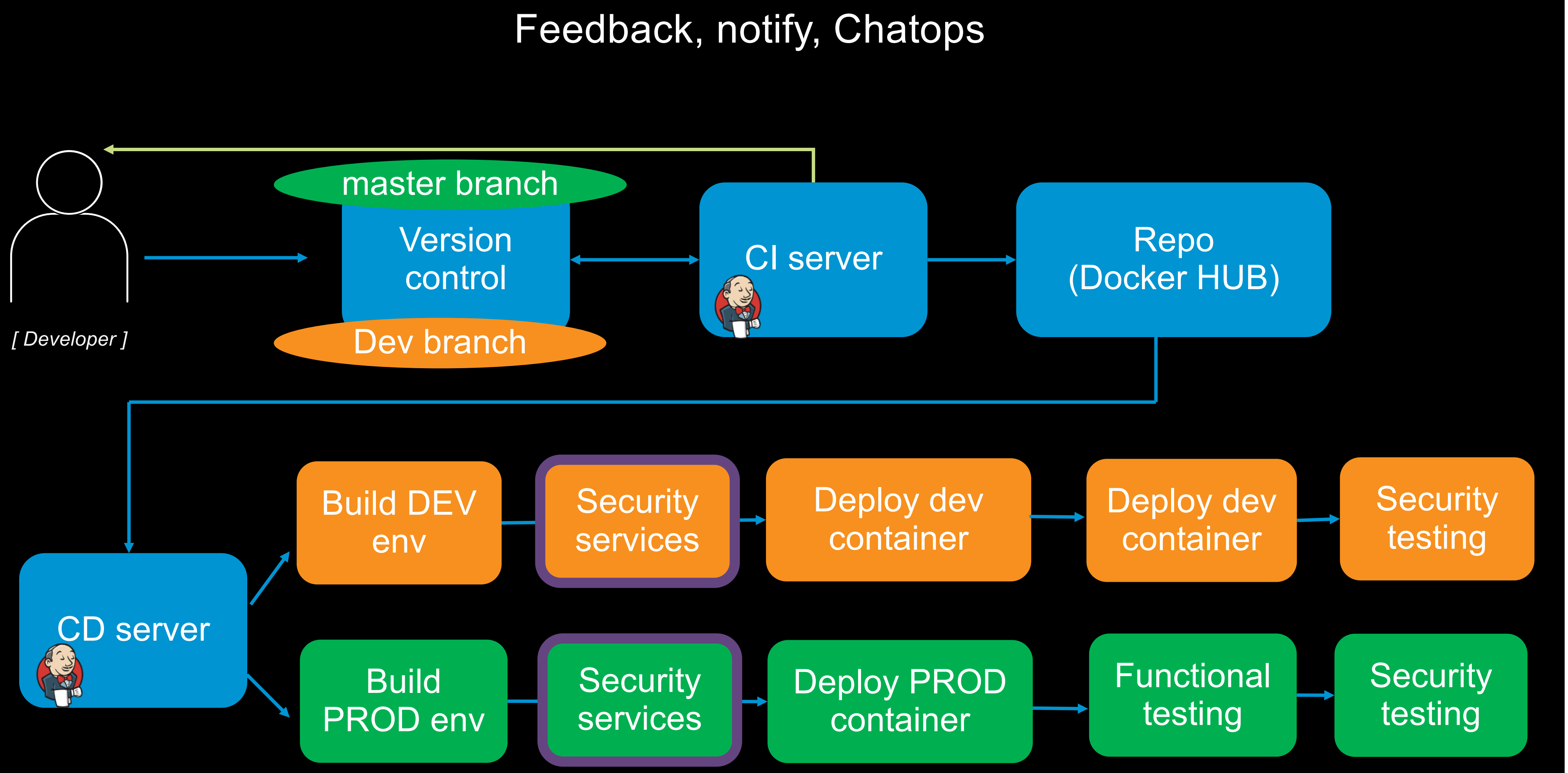
Security testing

# Continuous Delivery - CD

# F5 value proposition to agile dev team

- **Add protection from 0-day**
- **Address hard to solve problems:**
- L7 DOS
- Credential stuffing
- BOTs, Mobile BOTs, TLS , Rate limiting, Account takeover…
- Security plugins (iRules)

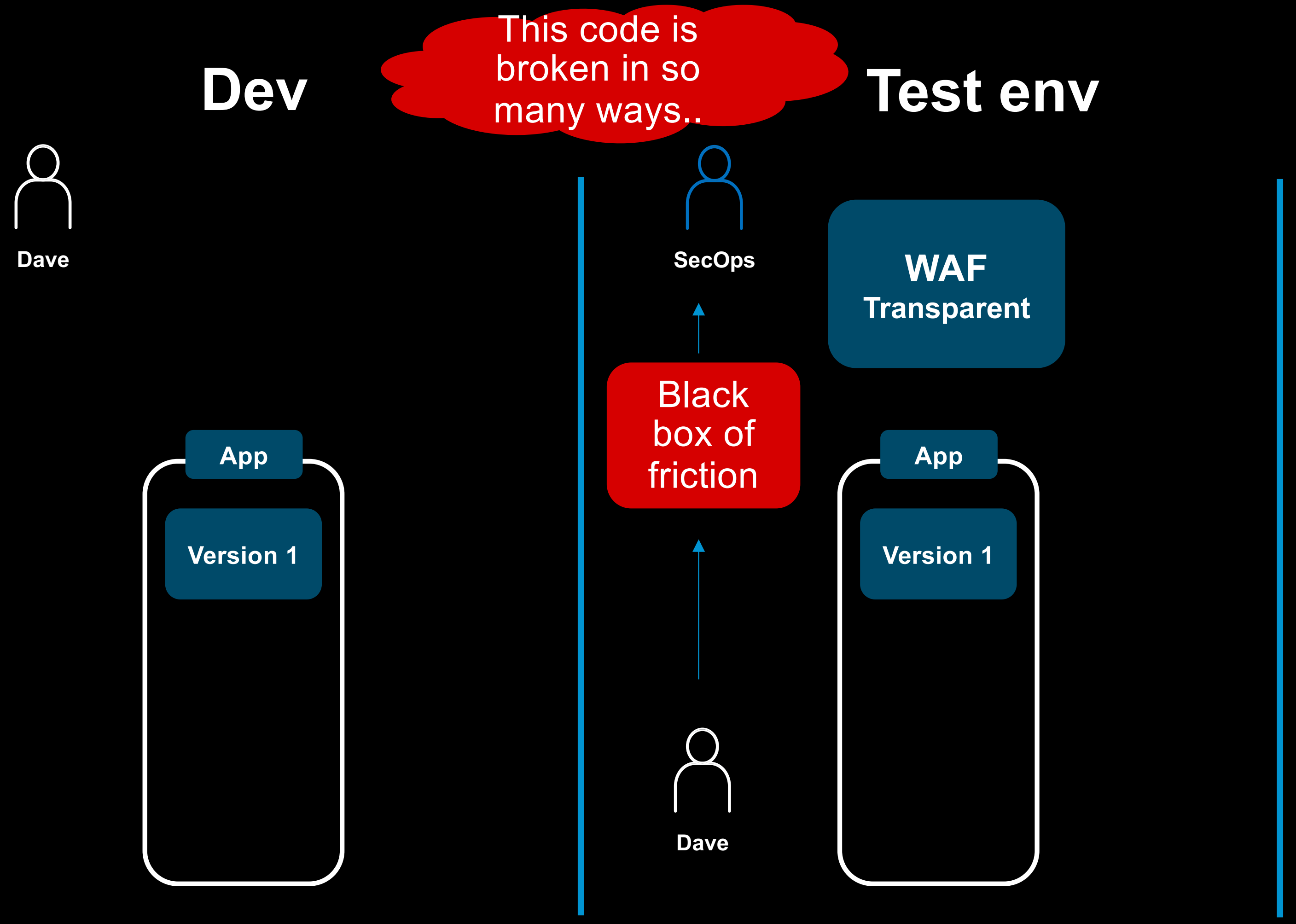
Not so relevant – Positive security controls e.g.: URL learning, parameter learning



# Continuous delivery - CD

# How to plug F5 WAF into the pipeline?

# Traditional Approach to rolling a policy to production:



Dev

Dave

App

Version 1

This code is broken in so many ways..

SecOps

Black box of friction

Dave

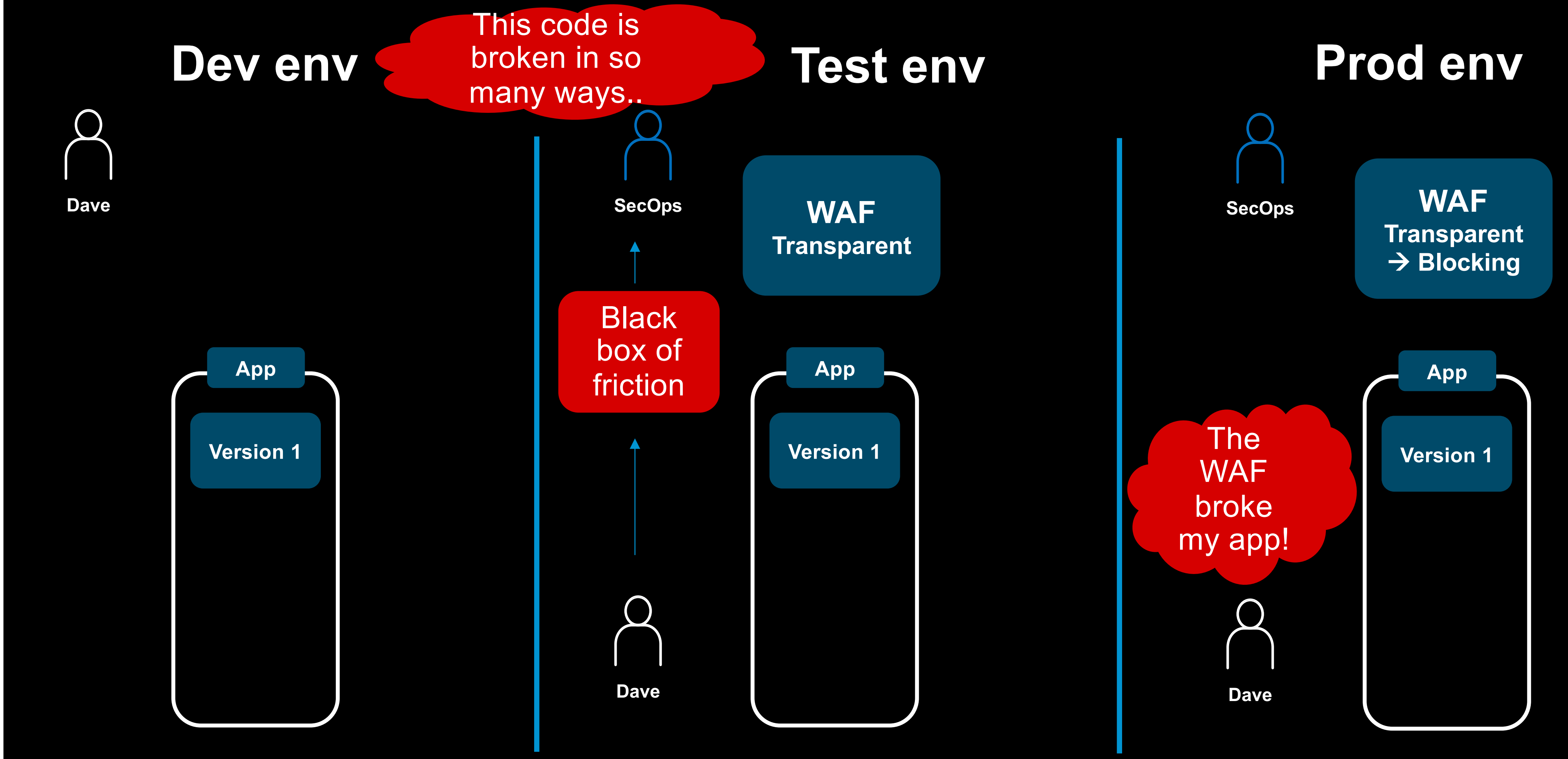
Test env

WAF Transparent

App

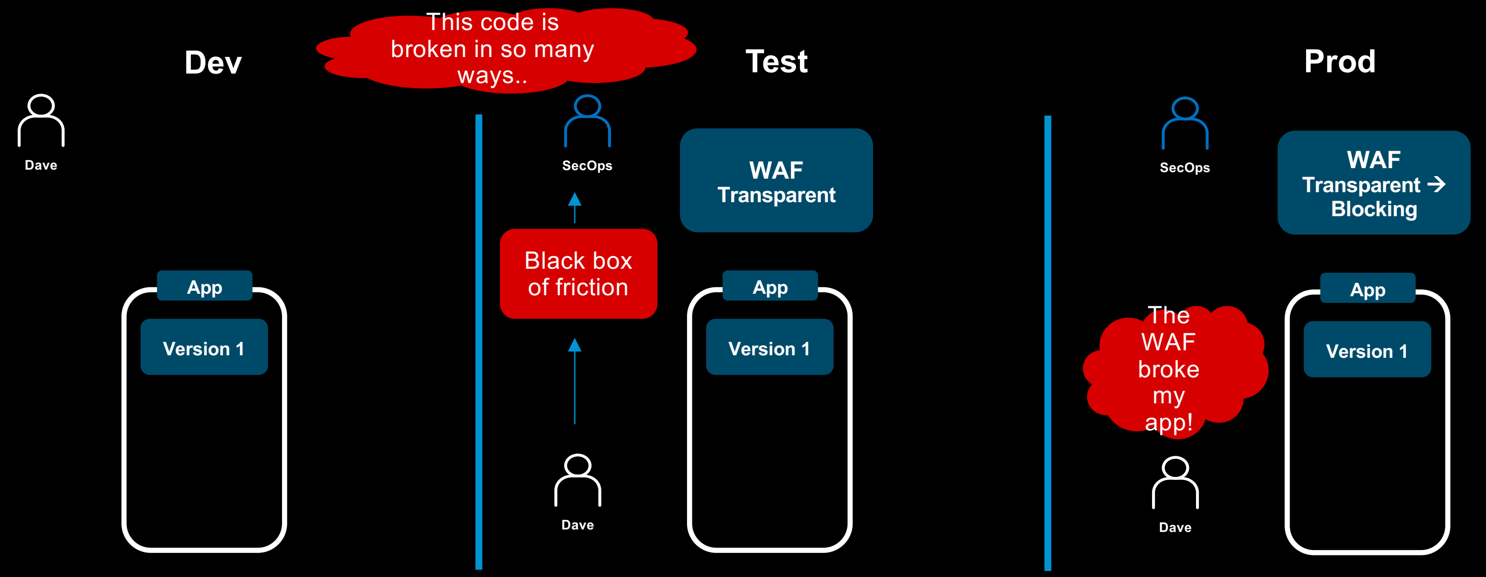
Version 1

# Traditional approach of rolling a policy to production



# Traditional approach of rolling a policy to production

- Error prone
- Security involved in each deployment
- Not properly documented
- Adds friction and slow the pace of innovation



# DevSecOps: What to Build (principles)

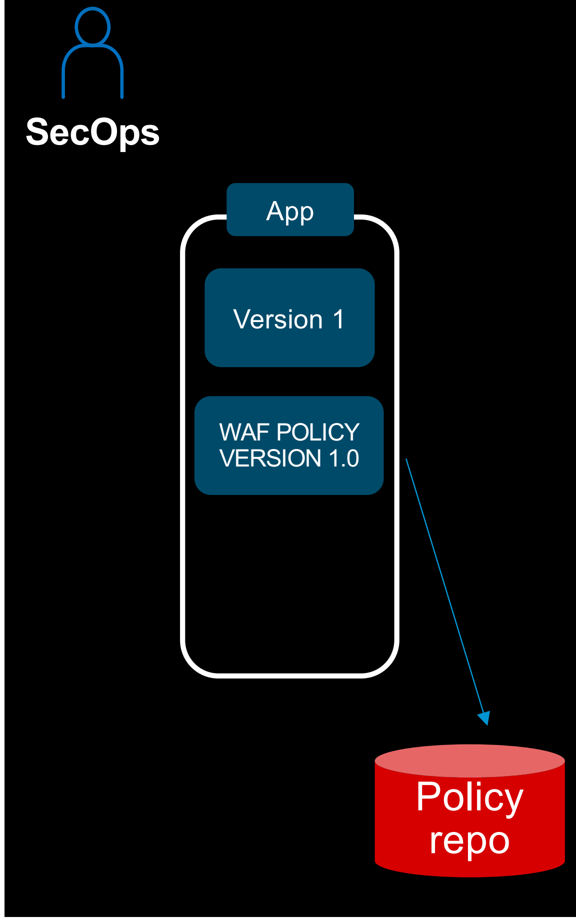
- Stay away from the deployments! Focus on policy
- Build reusable, security services
- CVE Vulnerabilities / Bot Protection / Web Exploits
- DOS protection
- Rate limiting
- Provide visibility to the app owner  - feedback loop
- Use app/security metrics to test negative/positive impact and continuously improve

# WAF policy principles

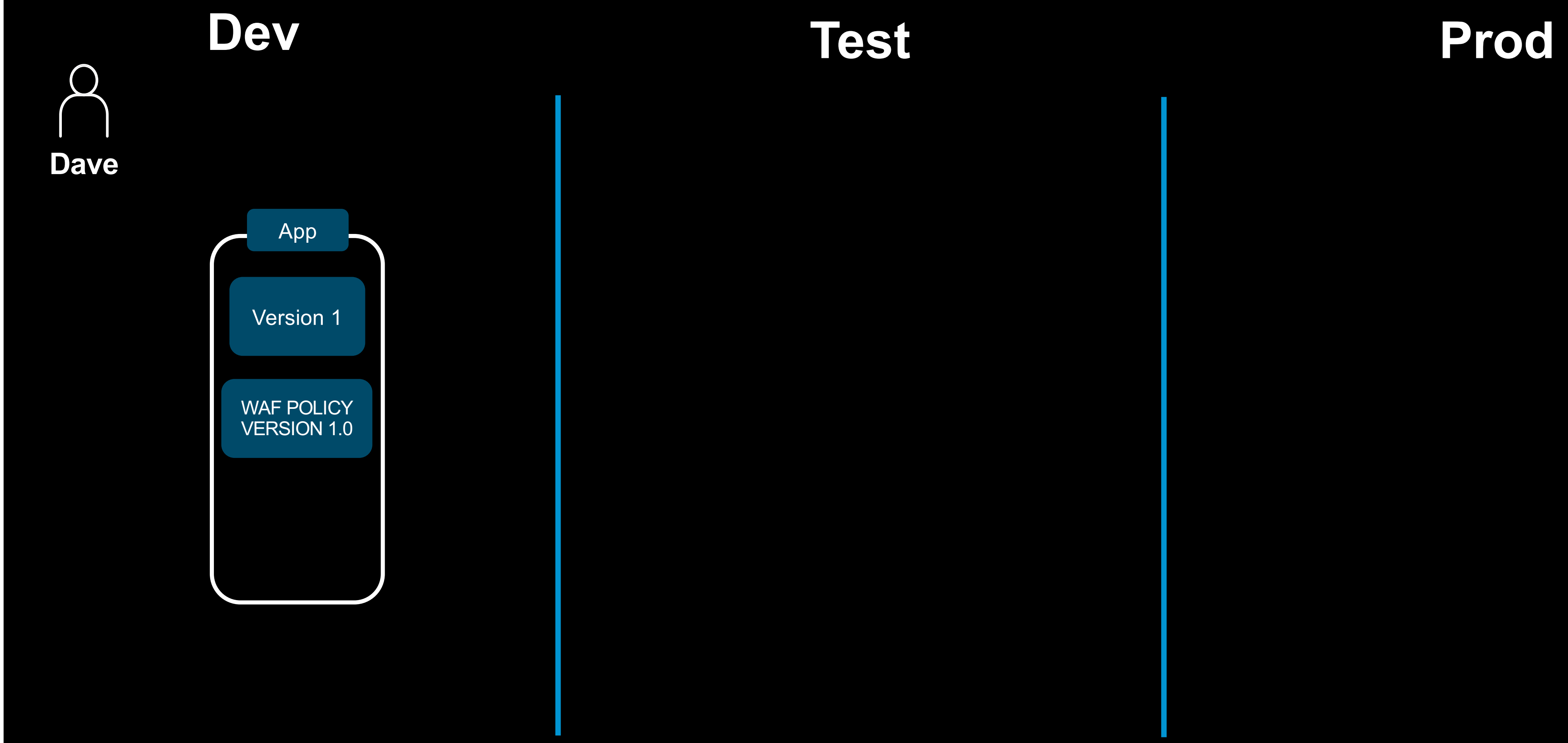
- When / where to use transparent ? **NEVER**
- Fail small/fast in dev
- Policy templates
- Manageable number of templates
- Deviation from templates controlled by the app team
- Can use Policy Builder
- To detect false positives
- Not to tighten the policy

# Creation of a policy template

**Linux-high - CVE Vulnerabilities + Bot Protection**



# Deployment of a policy to DEV



Dev

Test

Prod

Dave

App

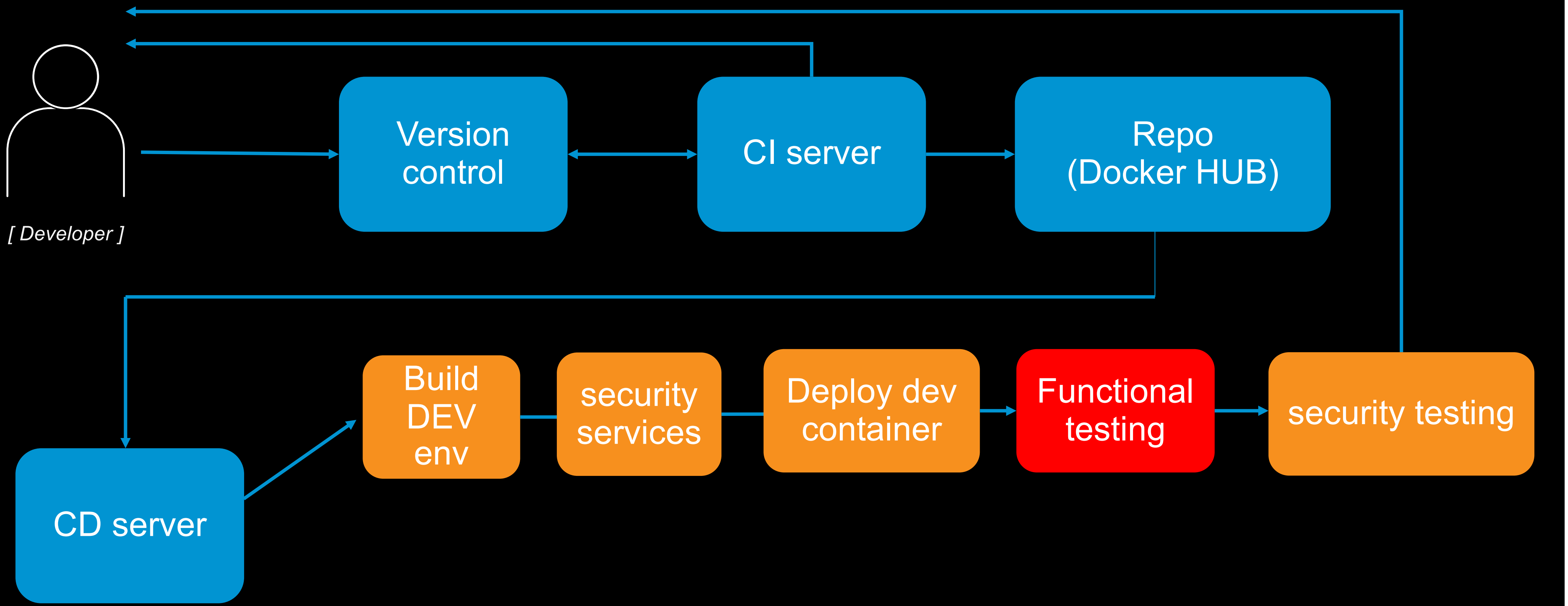
Version 1

WAF POLICY
VERSION 1.0

# CD

Feedback, Notify, Chatops

WAF BLOCKED



[ Developer ]

Version control

CI server

Repo (Docker HUB)

CD server

Build DEV env

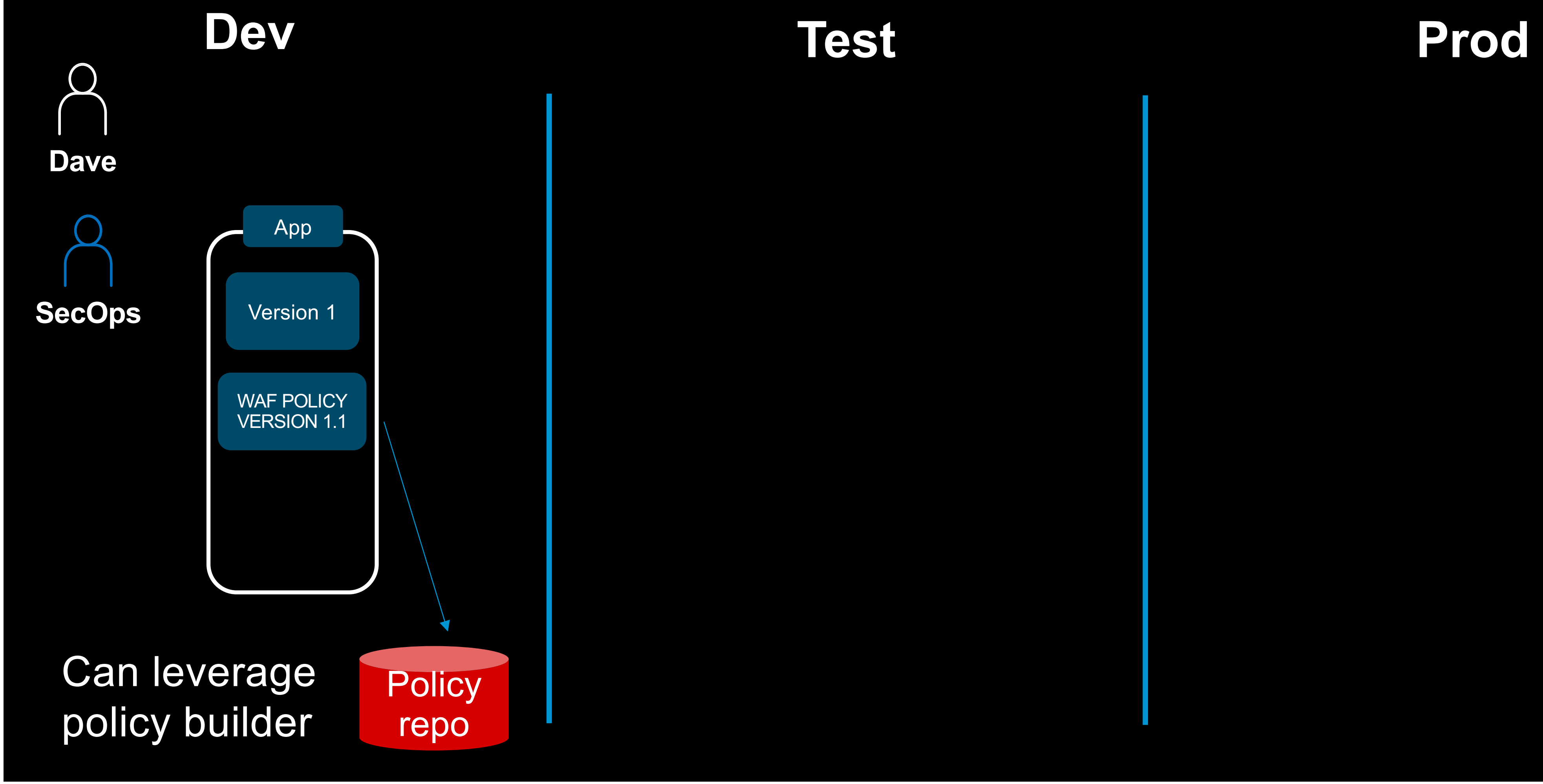
security services

Deploy dev container

Functional testing

security testing

# Fix the WAF policy template



Dev

Test

Prod

Dave

SecOps

App

Version 1

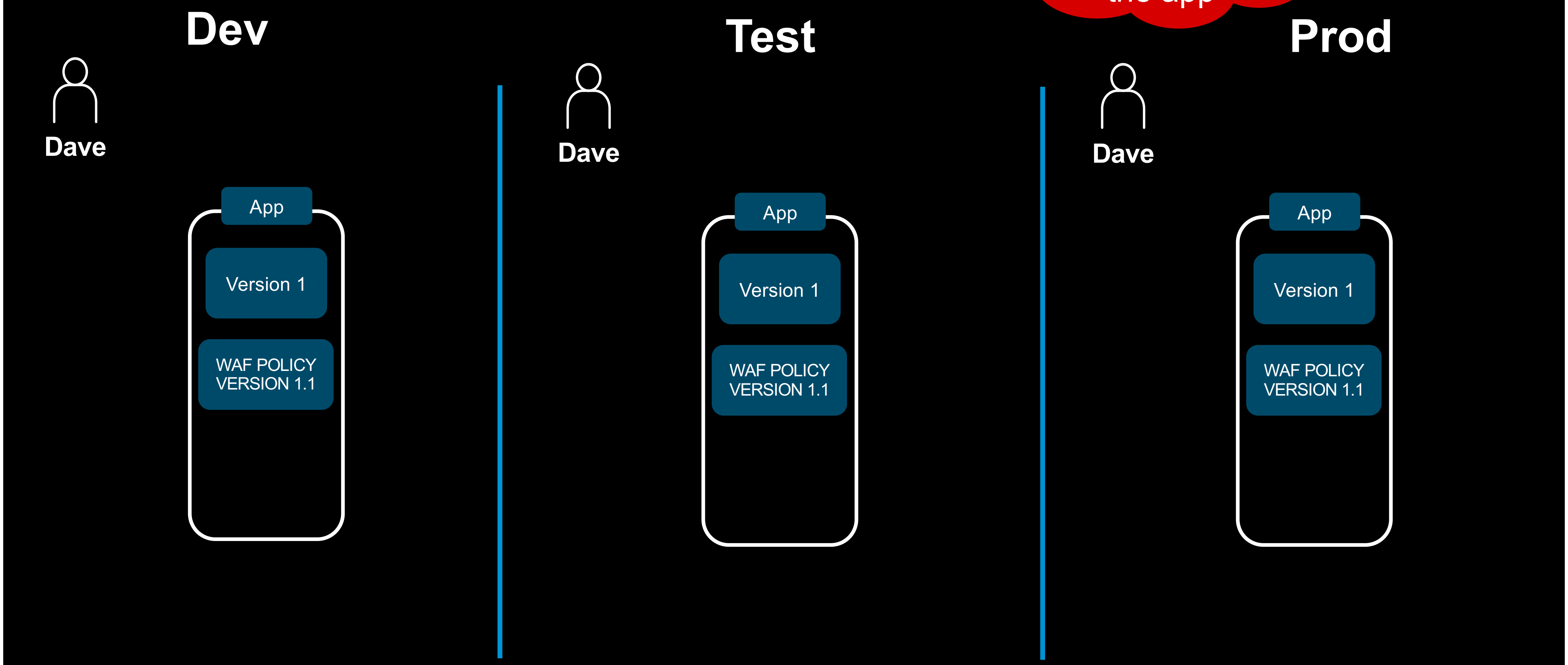
WAF POLICY
VERSION 1.1

Policy repo

Can leverage policy builder

# Deployment of a policy to DEV

App is deployed and WAF protects the app



Dev

Dave

App

Version 1

WAF POLICY VERSION 1.1

Test

Dave

App

Version 1

WAF POLICY VERSION 1.1

Prod

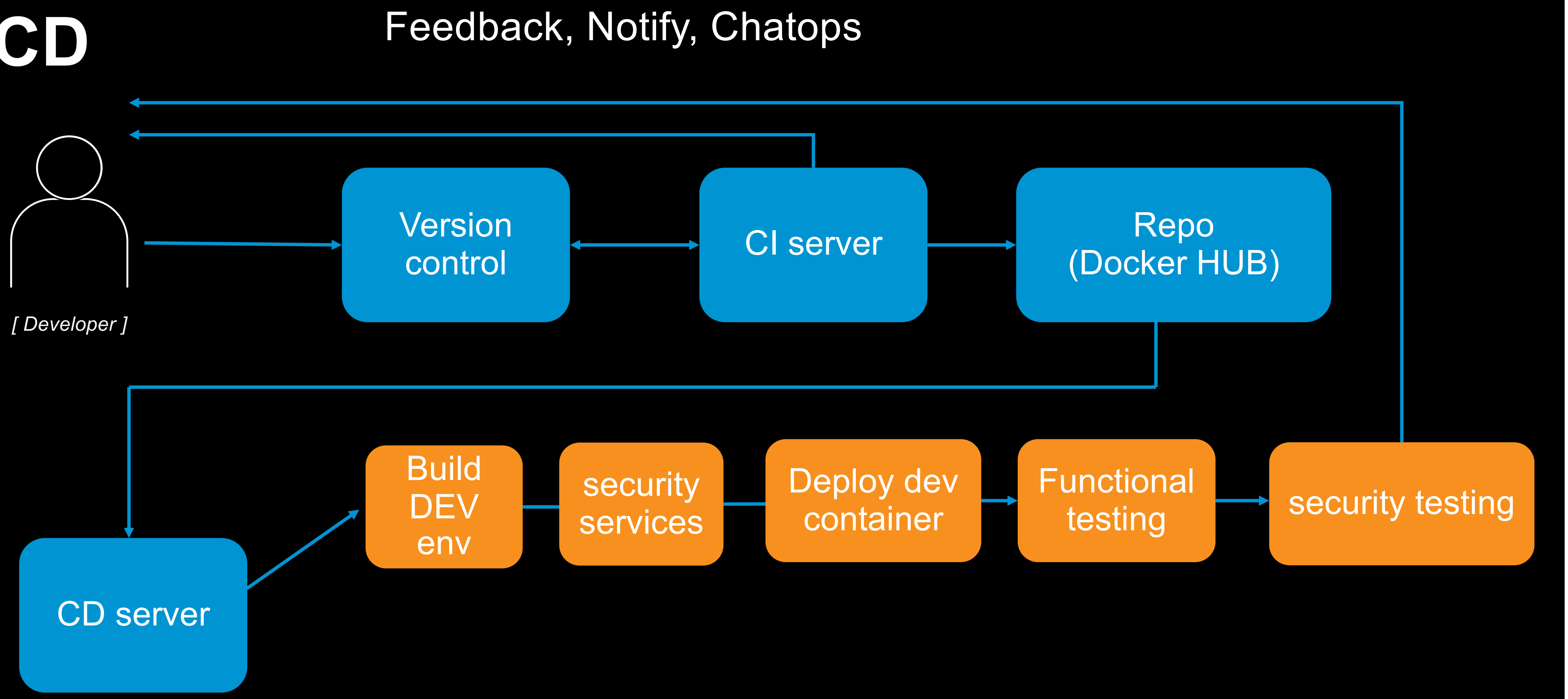
Dave

App

Version 1

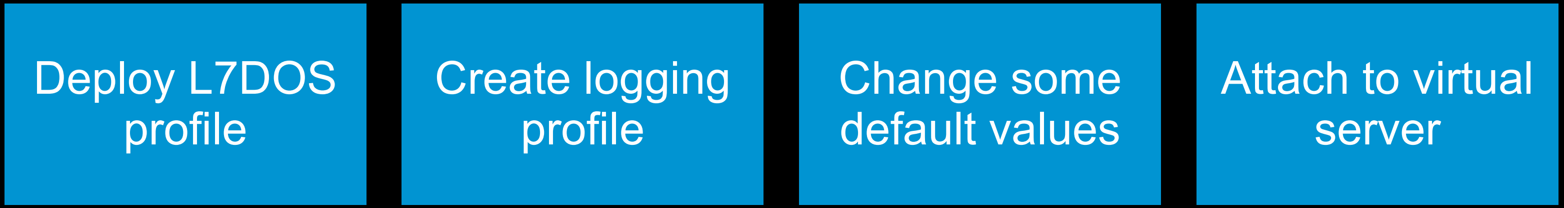
WAF POLICY VERSION 1.1

# CD


Feedback, Notify, Chatops
[ Developer ]
Version control
CI server
Repo (Docker HUB)
CD server
Build DEV env
security services
Deploy dev container
Functional testing
security testing

# Declarative WAF

Problem: Automated attacks
Solution: F5 proactive bot defense
**API** (what should you expose):
botdefense_template: “example”
State: “enable/disable”



Deploy L7DOS profile

Create logging profile

Change some default values

Attach to virtual server

# Benefits / what's important for each team

## SecOps:

1. **Focus on security** not on button pushing
2. Not involved in rollbacks
3. Enables faster adoption of **advanced security features**.

## DevOps (Tools team) :

1. Clear **visible changes**
2. Changes are part of the pipeline – enables **continuous improvement of the deployment**
3. Enables advanced deployments – blue/green , canary - **increases reliability**

## App owner:

1. Security policy is deployed early in development and enables **faster time to market.**
2. Choses what are the features that make sense for him and have '**control over his destiny'**

App
Version 1
WAF POLICY
VERSION 1.1