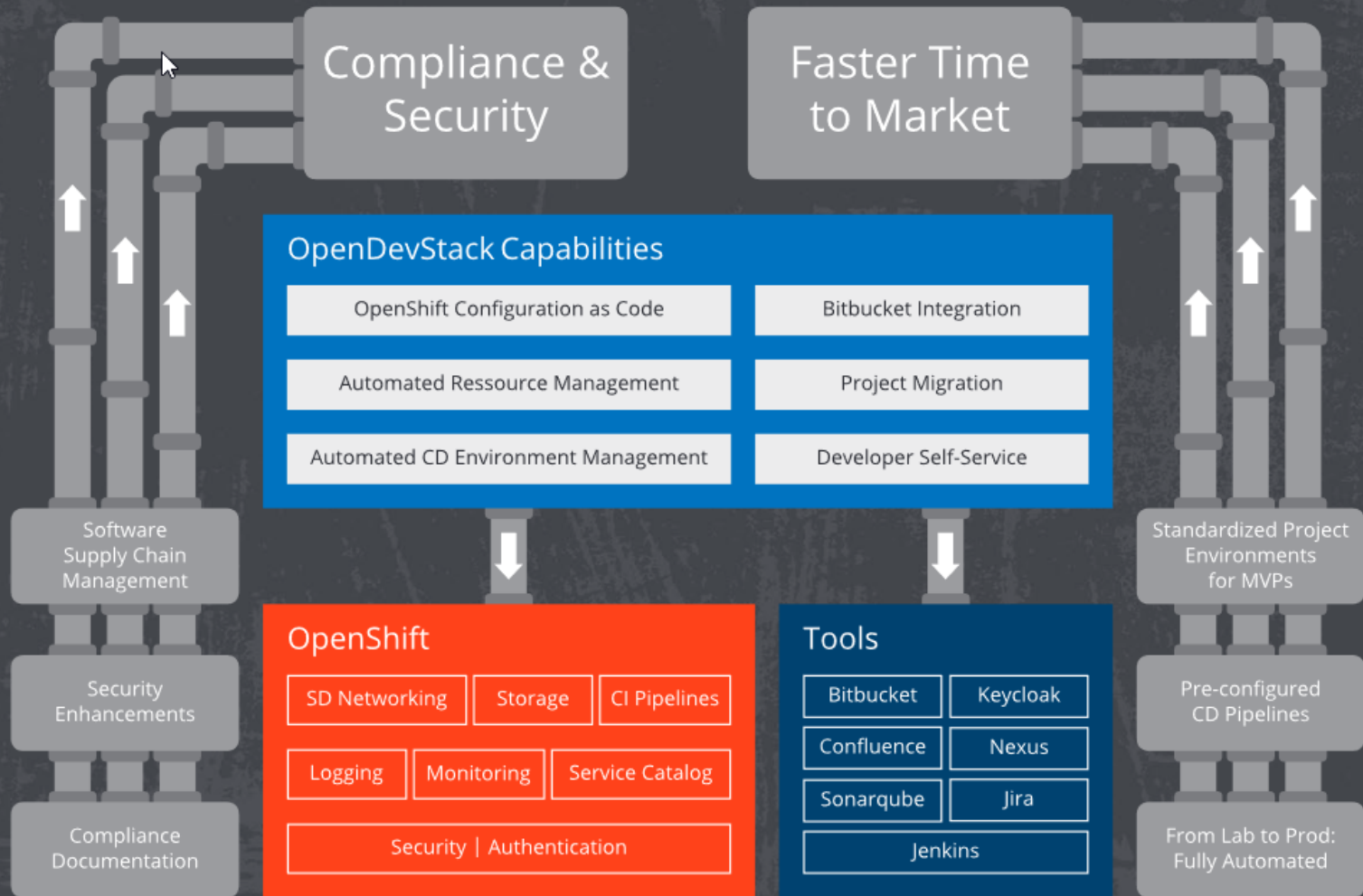




OpenDevStack

Scaling DevOps in the Enterprise
based on OpenShift

Richard Attermeyer, OPITZ CONSULTING



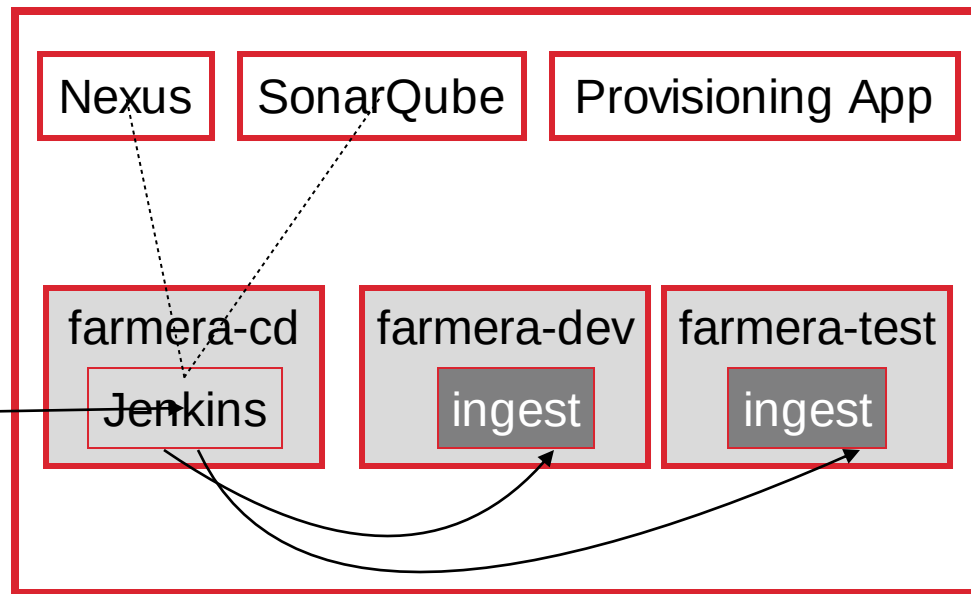
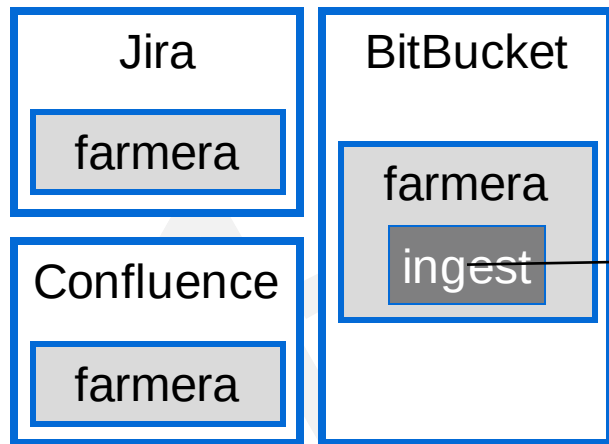
Key Concepts of OpenDevStack

- One common toolset, uniformly configured and integrated right from the start
- Freedom + control within the team (agile!)
- Quickstarters to aid MVP development
- Build application against a platform
- Built-in CI/CD delivery pipeline

■ See: <https://www.opendevstack.org/ods-documentation/common/latest/getting-started/introduction.html>

OpenDevStack Overview

1. Provision project
2. Provision component



Roadmap

- More & better integration of tools
- Extend quickstarters
- Software defined everything
- Documentation generation and multi-repo orchestration





Demo

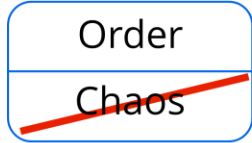
<https://pixabay.com/photos/adult-artisan-tools-workshop-1866533/>

Demo

<https://www.youtube.com/watch?v=ijgMajbD5d0>

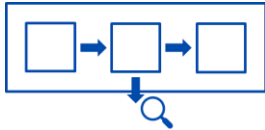


Adressing DevOps scaling obstacles

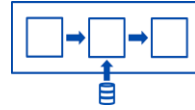


Standardization

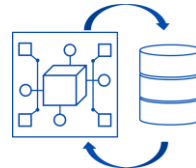
OpenDevStack: Scaling DevOps in the Enterprise



Code Analysis



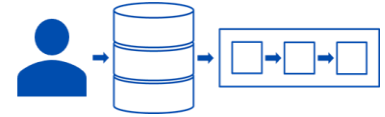
Centrally Defined
Pipeline Functions



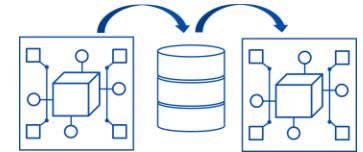
GitOps



Self-service



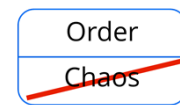
Bitbucket Integration



Cluster Migration

Quickstarters

Currently supported project types



Standardization



Flask

web development,
one drop at a time



Aktuell: 15, siehe <https://www.opendevstack.org/ods-documentation/ods-quickstarters/latest/index.html>

One prototype Jenkins Pipeline

General Pipeline



Centrally Defined
Pipeline Functions

Customizable Build Instructions

```
odsPipeline(  
  image: "${dockerRegistry}/cd/jenkins-slave-maven",  
  projectId: projectId,  
  componentId: componentId,  
  branchToEnvironmentMapping: [  
    'master': 'test',  
    '*': 'dev'  
  ]  
) { context ->  
  stageBuild(context)  
  stageScanForSonarqube(context)  
  stageStartOpenshiftBuild(context)  
  stageDeployToOpenshift(context)  
}
```



```
def stageBuild(def context) {  
  // depending on the technology  
}
```

Jenkins Pipeline Build

Deployment in Openshift - `branchToEnvironmentMapping`

- Jenkinsfile: `branchToEnvironmentMapping` defines how branches are mapped to OpenShift environments
- In `be-spring-boot`:

```
branchToEnvironmentMapping: [  
  'master': 'test',  
  '*': 'dev'  
]
```

- `,master'` builds are deployed into `,test'`
- All other branches will be deployed into `,dev'`



Richard Attermeyer

Richard Attermeyer works as Senior Solution Architect for OPITZ CONSULTING in the area of technology management and innovation and, among other things, heads the OC architecture board.



WWW.OPITZ-CONSULTING.COM



[@OC_WIRE](https://twitter.com/OC_WIRE)



[OPITZCONSULTING](https://www.youtube.com/OPITZCONSULTING)



[opitzconsulting](https://www.linkedin.com/company/opitzconsulting)



[opitz-consulting-bcb8-1009116](https://twitter.com/opitz-consulting-bcb8-1009116)