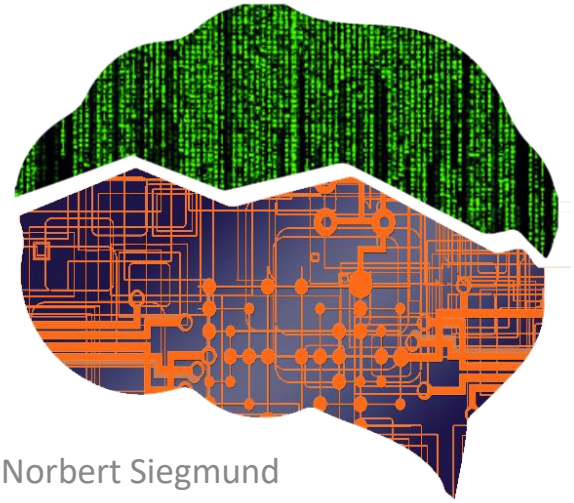
Modern Software Technologies

Introduction and Motivation



Prof. Dr.-Ing. Norbert Siegmund
Intelligent Software Systems

Bauhaus-Universität Weimar

What this Course is About

- You will learn state of the art software technologies, such as
  - Docker, continuous integration, MicroServices, etc.
- You will learn how to present the acquired skills
- You will apply these techniques by writing a tutorial that can be used by your fellow students

Topics: Full Stack Java Developer

Essential skills that are needed in industry!

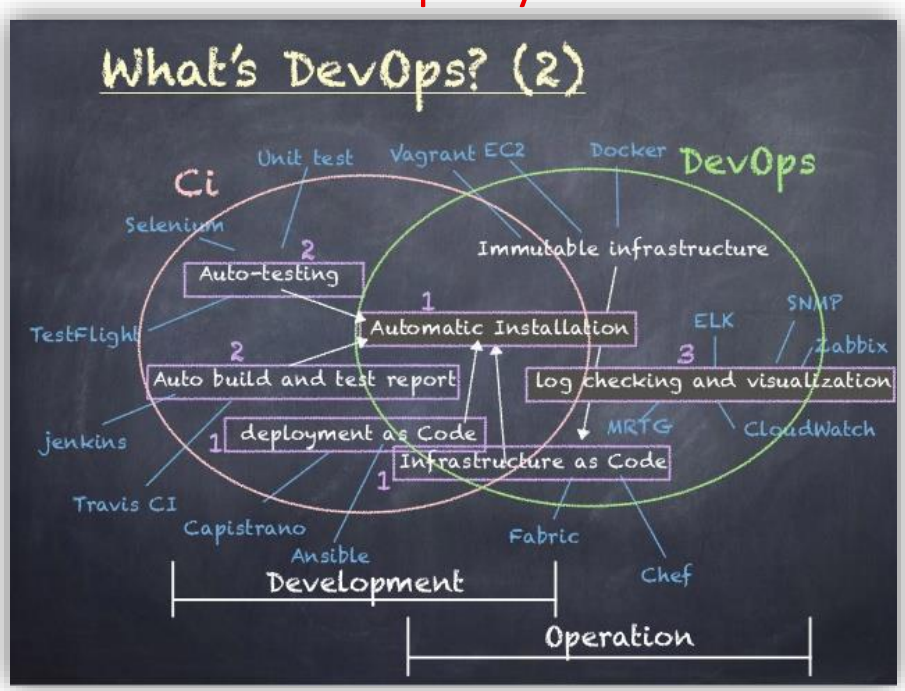
Wanted Skills

Wenn Du darüber hinaus noch intensive und fundierte Erfahrungen mit einigen der folgenden Tools, Technologien und Konzepten mitbringst:

- Spring (Boot/Cloud)
- JAX-WS und JAX-RS
- ReactJS, TypeScript, Webpack
- Gradle, Git, SonarQube, Jenkins
- JUnit, Cucumber, Selenium, testcontainers
- MariaDB, Oracle, PostgreSQL, MongoDB
- Docker, Puppet, Vagrant, Ansible
- Amazon Web Services
- Microservices, Micro Frontends

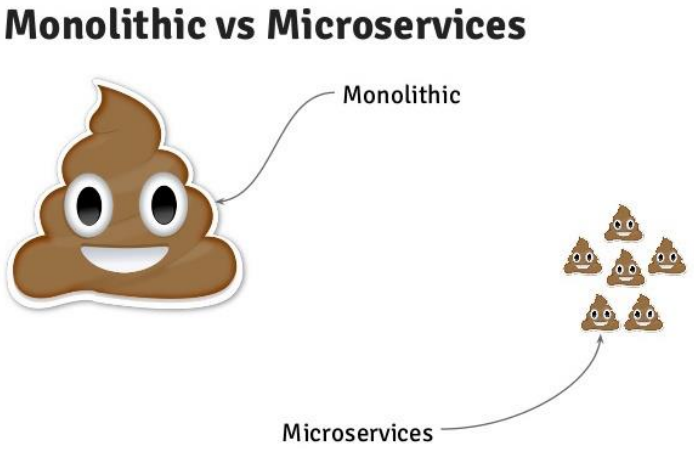
… dann wollen wir Dich sogar unbedingt kennenlernen!

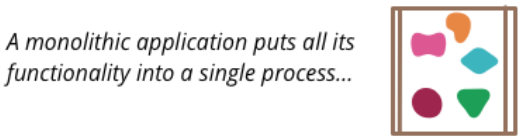
These are the topics you will learn!

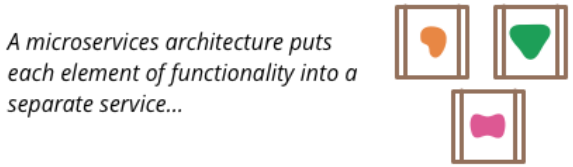


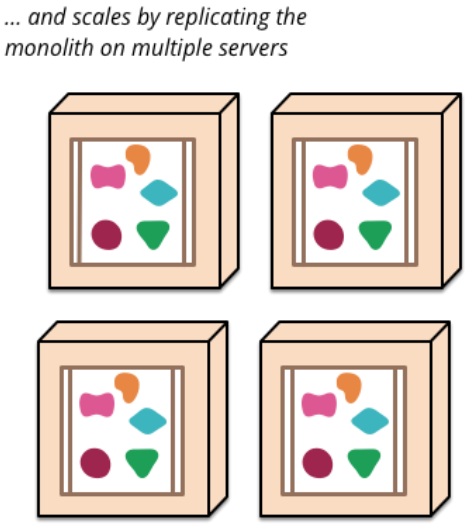
MicroServices

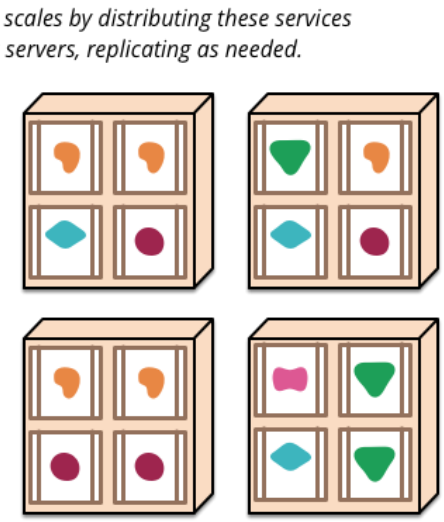
- Functional scalable applications







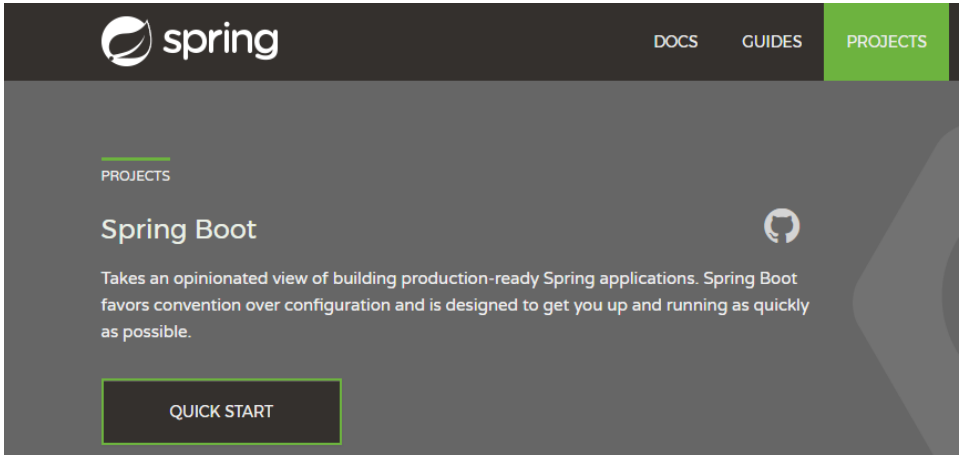




Frontend: Vue.js
Service-Layer: REST
Backend: Spring Boot

Spring Boot & Flask

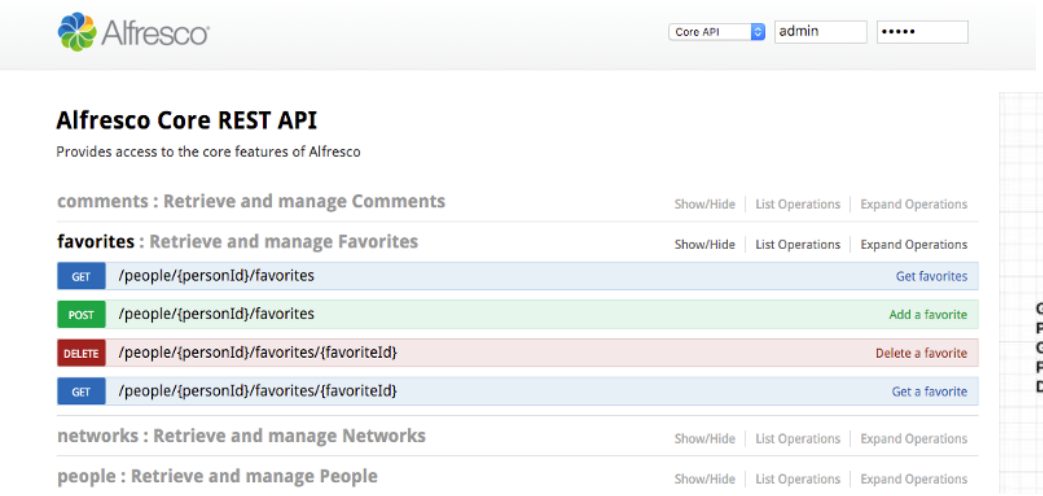
- Rapidly build Web-based Java applications with minimal overhead

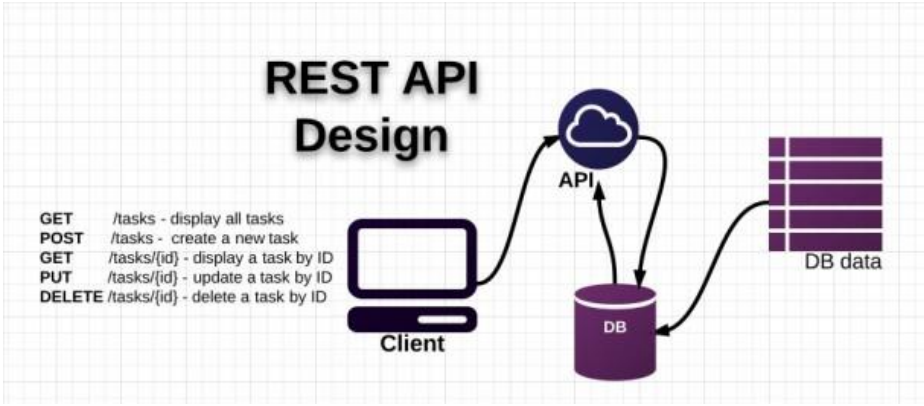


- Python-based microframework for Microservice creation

REST: Representational State Transfer

- Programming paradigm for distributed systems (e.g., Web services)
- Every resource or entity addressed by an URI will elicit a response (e.g., as XML, HTML, JSON, etc.)





Vue.js (or …)

The Progressive JavaScript Framework

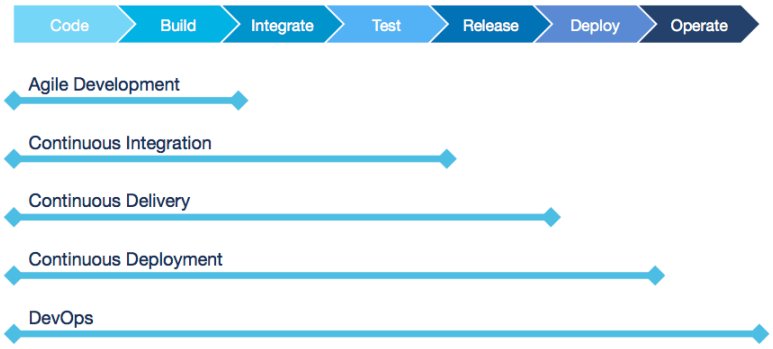
GET STARTED GITHUB

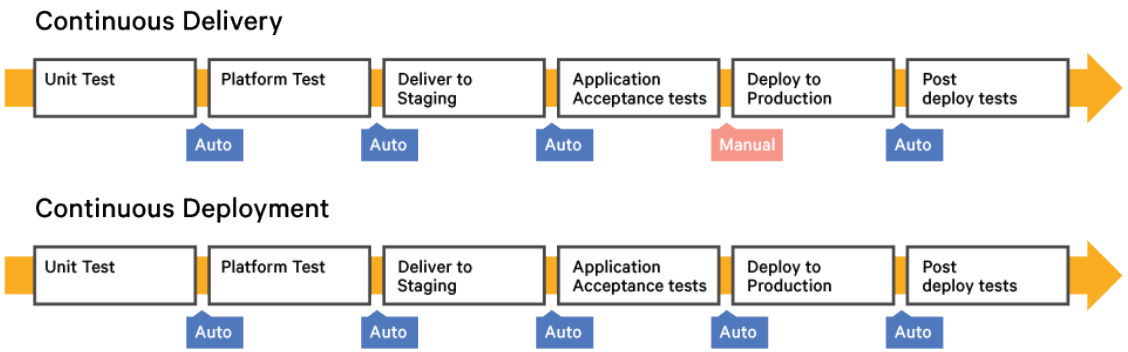
- Easy to learn JavaScript-Framework for rapidly building user interfaces

Decision point	Angular 2	React	Vue.js
Stable	Yes	Yes	Yes
Backed by a strong community or some big players	Yes, huge community and Google is behind it	Yes, huge community and Facebook backs it	Not as huge but is big enough and is backed by Laravel and Alibaba
Good documentation	Yes	Yes	Yes
Easy to learn	Not with Typescript	Kind of	Yes
Integration with Bootstrap	Yes	Yes	Yes
Small	566K	139K	58.8 K
Allow us to reuse code	Yes	No, only CSS	Yes, HTML and CSS
Coding speed	Slow	Normal	Fast
Reactivity	Kind of	Yes	Yes
Component based	Yes	Yes	Yes

Continuous Integration (CI)

- SE principle for trying to merge each change made to the code base into productive code
  - Continuous testing (and deployment)
  - Change rejected when test fails or code quality is low





Maven™ (or Gradle)

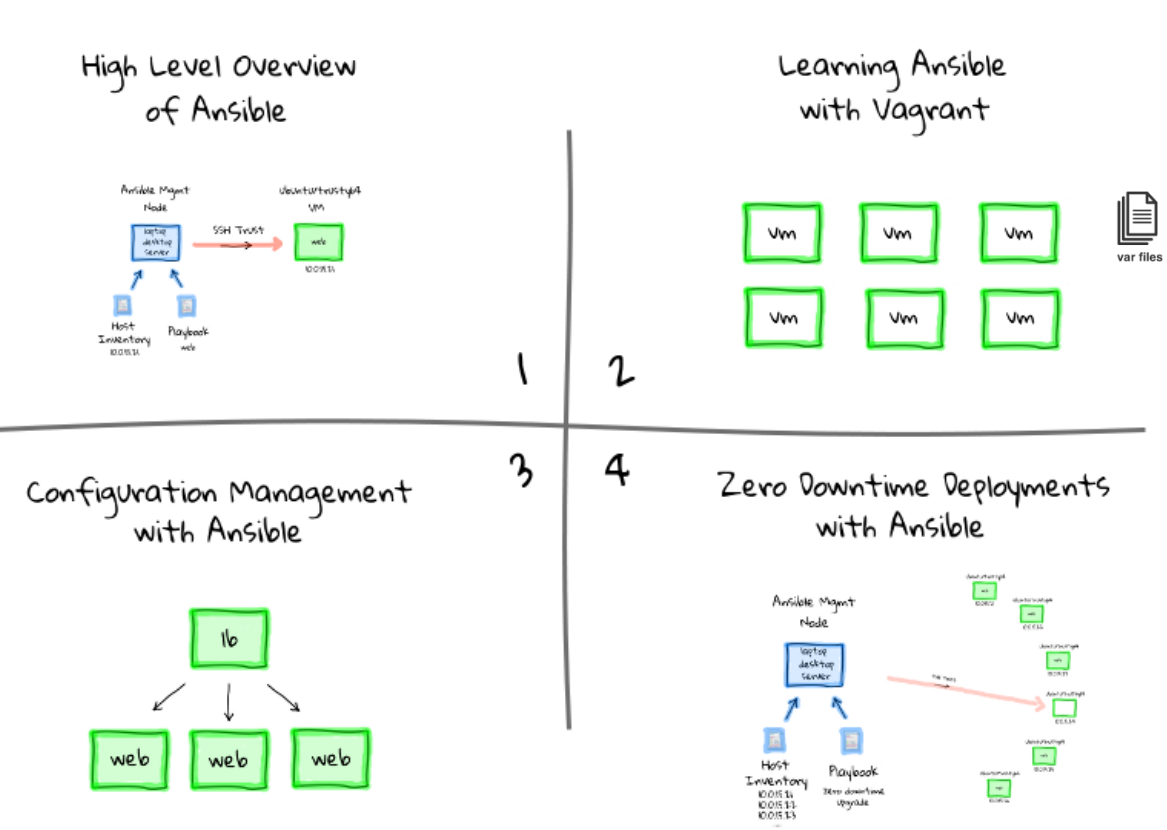
- Manages project **builds**, **dependencies**, and **reporting**

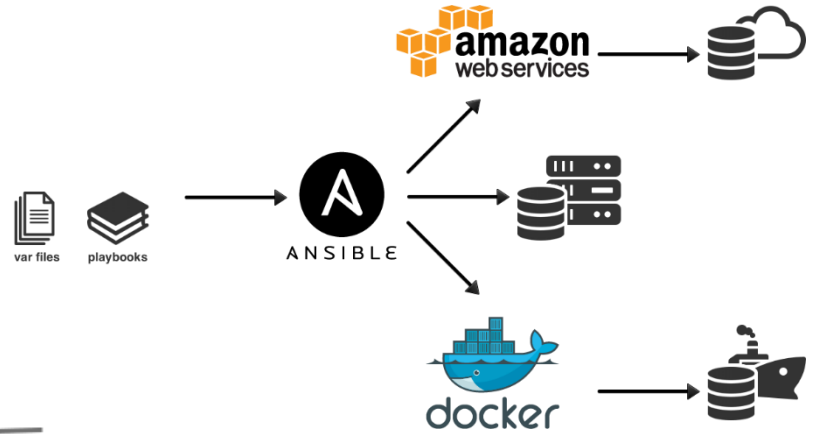
Benefits of Maven over Ant

1. It makes project build process easy.
2. It provides easy and uniform build system.
3. It provides quality project document Information.
4. Managing project dependencies.
5. Provides guild lines for better project management practices.
6. It allows to build project using project object model (POM).
7. It downloads required dependency's jar files automatically from Maven central repositories.

Ansible

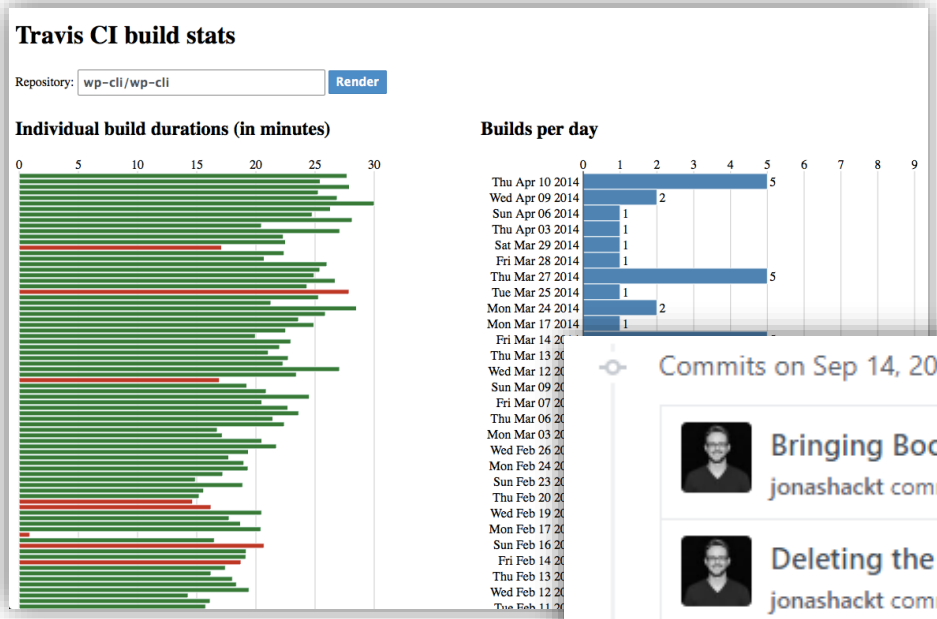
- Automation engine for software deployment and administration as well as configuration management

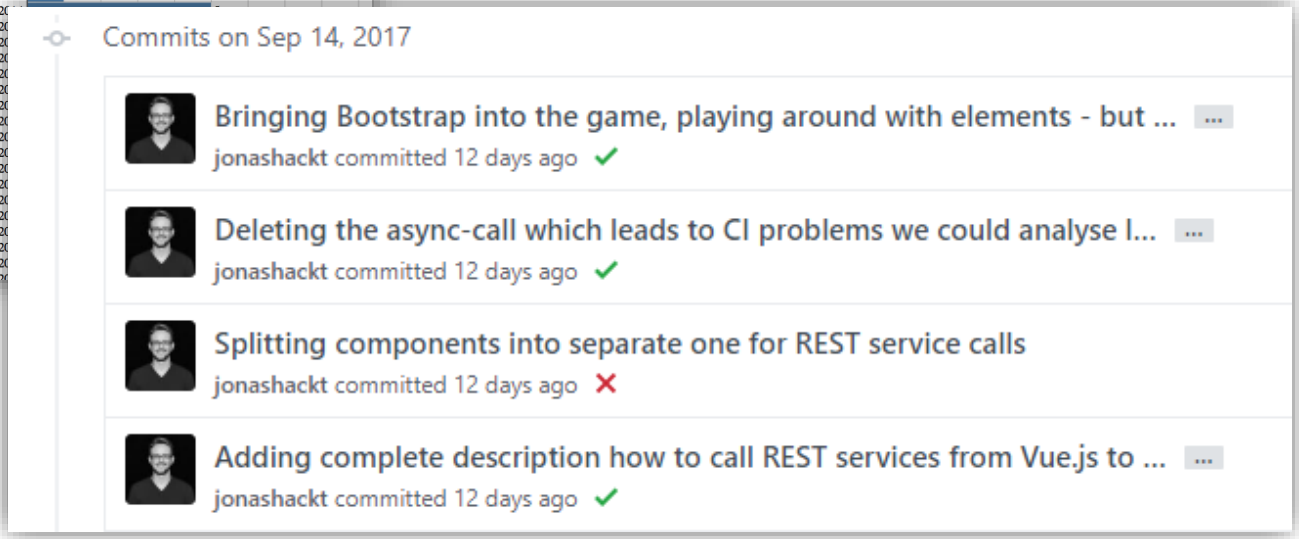




Travis CI (or Jenkins)

- Distributed continuous integration service for building and testing software projects (e.g., GitHub)





Continuous Code Analysis

Web service for tracking code (line) coverage of your tests

Inspect health or quality of your project

C O D A C Y

Alternative to sonarqube

JUnit (and Selenium for GUI Testing)

Unit testing for Java applications

Tool suite to automate Web browsers for testing purposes

Eliminating the problem of “works on my machine”

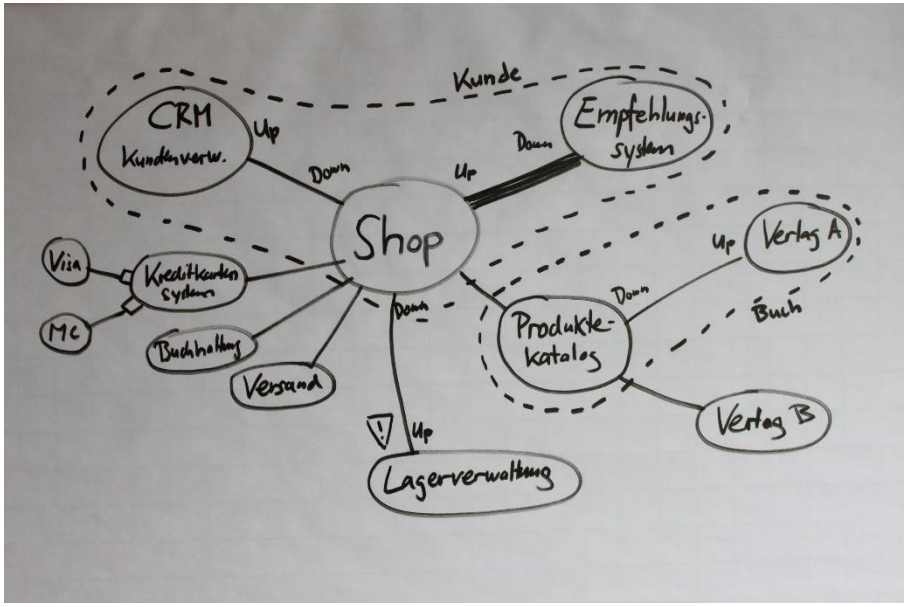
Software runs on isolated containers that are easy to deploy and run without overhead

Testcontainers

“Testcontainers is a Java library that supports JUnit tests, providing lightweight, throwaway instances of common databases, Selenium web browsers, or anything else that can run in a Docker container.”

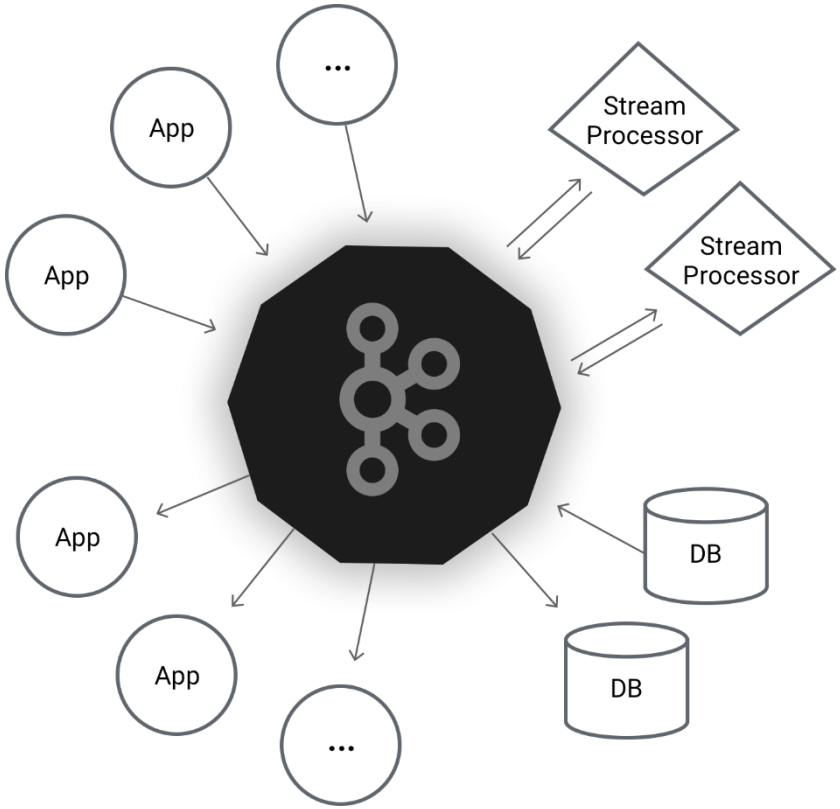
Domain-Driven Design (DDD)

Domain-driven Design is a technique to approach and modell a complex, object-oriented software.



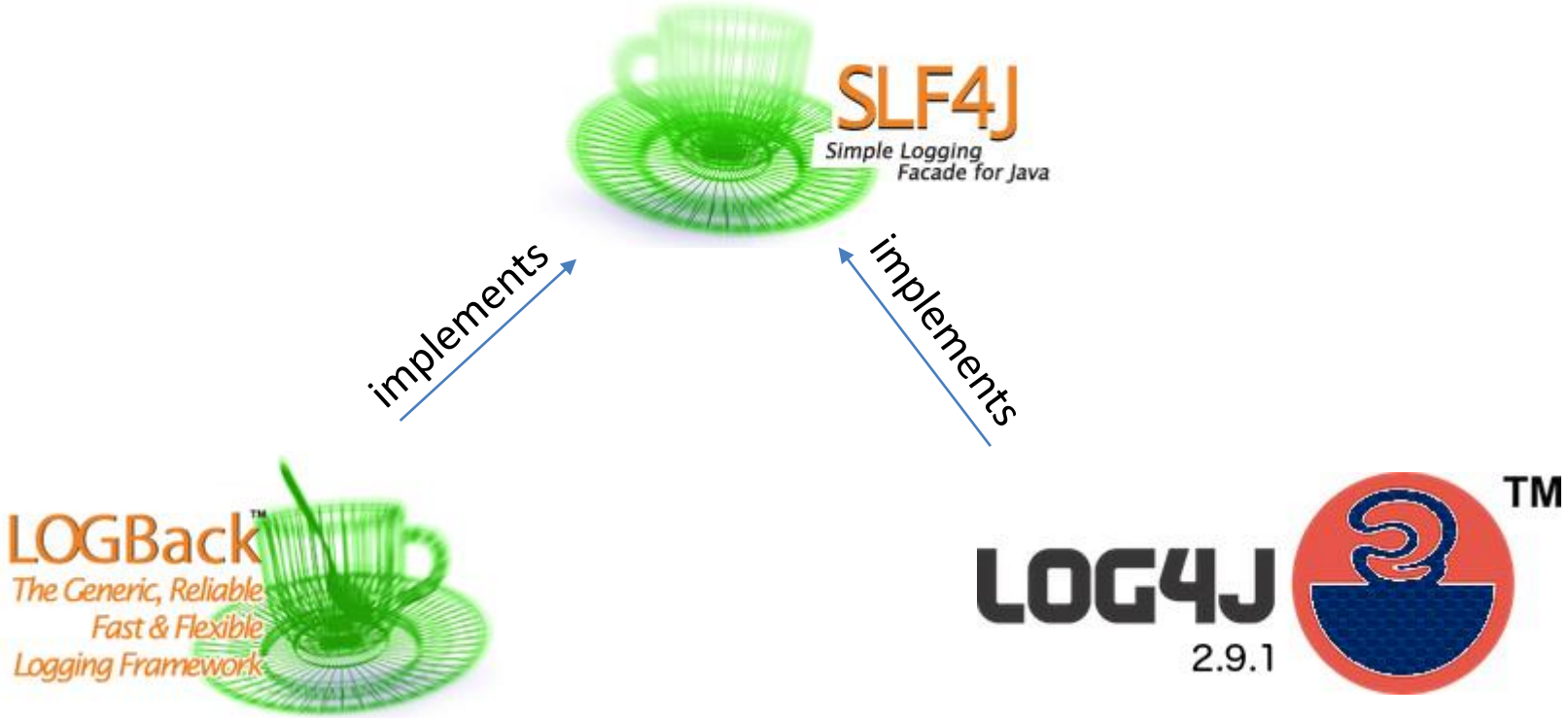
Stream Processing: Apache Kafka

A framework for maintaining and processing data streams.



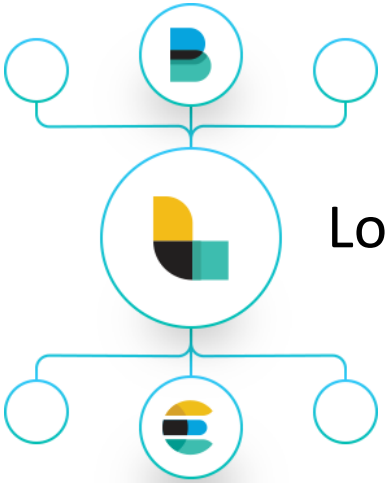
Bonus: Logging in Java (LogBack)

Abstraction for various logging frameworks



Bonus: Log Analysis with Elastic Search

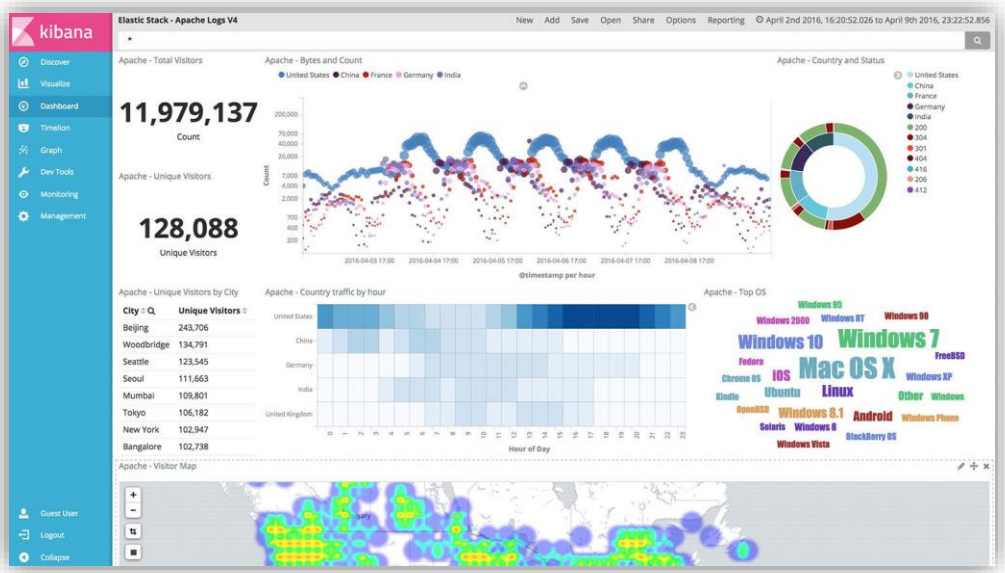
Beats: Data transfer to Logstash



Logstash: Dynamic pipeline for data collection

Elastic Search:
Distributed search and analysis engine

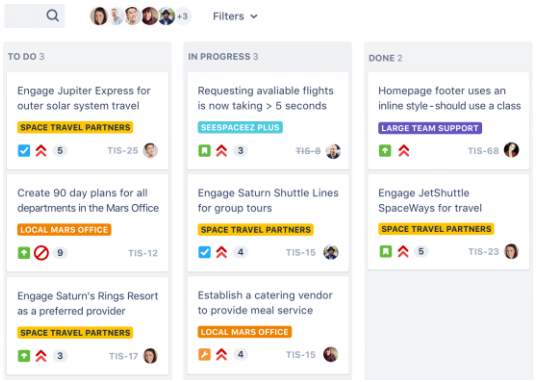
Kibana for visualization and analysis



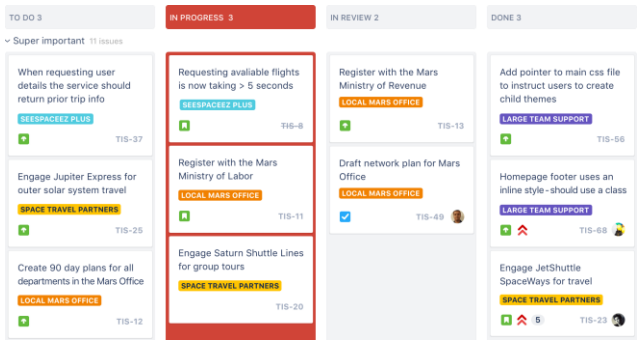
Bonus: Ticketing and Issue Systems (Jira/GitHub Issues)

JIRA

Project management tool for agile software development



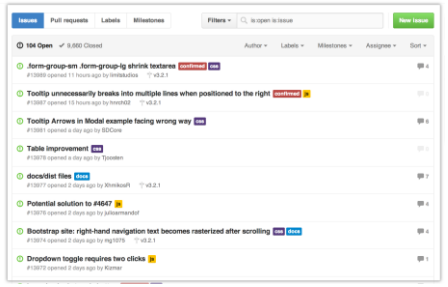
Scrum boards



Kanban boards

Reporting

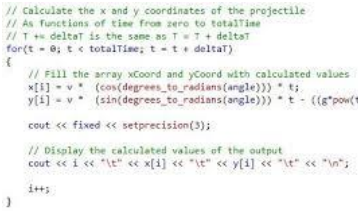
GitHub Issues: Keep track of tasks, bugs, etc.



Grading

- Presentation: 60%+10% present at the dates
  - Is the topic properly motivated?
  - Is the content correct and sufficient?
  - Is the style of the slides appropriate?
  - Is the talk engaging and easy to follow?

- Tutorial: 30%
  - Is the code properly commented / documented?
  - Is the tutorial appropriate?
  - Is everything available online to be reproduced?



Details…

- Topics are selected by lot (Lose); trading is possible within a day
- Presentations can be sent one week before the day for presentation arrives, but must be sent after the presentation

- Schedule will be put on the Web page