

# Bachelorarbeit: Yoda / Luke – Konzeption und Realisierung eines Programmsystems für die Echtzeitdokumentation in verteilten Rechenzentren

## Motivation

- Ein Rechenzentrum ist ein Ort stetigen Wandels durch Änderungen in der Infrastruktur.
- Die XCOM AG arbeitet an selbstregulierendem verteiltem Rechenzentrum.
- Vorstufe: Die Schaffung einer zentralen Datenbasis durch Echtzeitdokumentation der Rechenzentrumskomponenten.



## Methodik

### ISO 27001 und IT-Grundschutz

- Sicherheitsziele in Anlehnung an ISO 27001.
- Logbuch über Aktivitäten im Programmsystem.
- Kryptokonzept gemäß IT-Grundschutz

### Unterstützung von ITIL-Prozessen

- Rechenzentrumsbetrieb in Service Transition und Service Operation
- Primär tangiert das Programmsystem Service Asset and Configuration Management (SACM).
- SACM beeinflusst das Change Management, das Supplier Management und das Event-Management.
- Auch der Service Desk greift auf SACM zurück.

## Konzeption

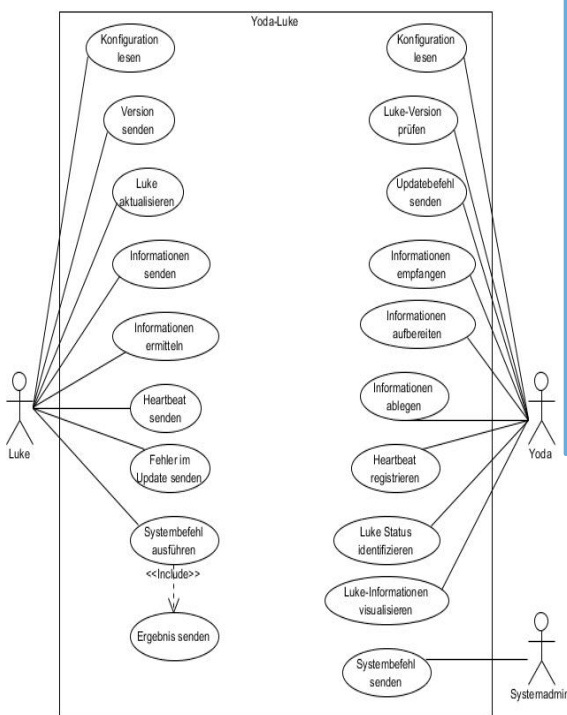
- Yoda besitzt Benutzerschnittstelle, Anwendungslogik und Datenbankschicht
- Luke verfügt über Anwendungslogik und fungiert als „Sonde“ auf Servern im Rechenzentrum
- Intelligente Informationsübertragung

## Realisierung Yoda

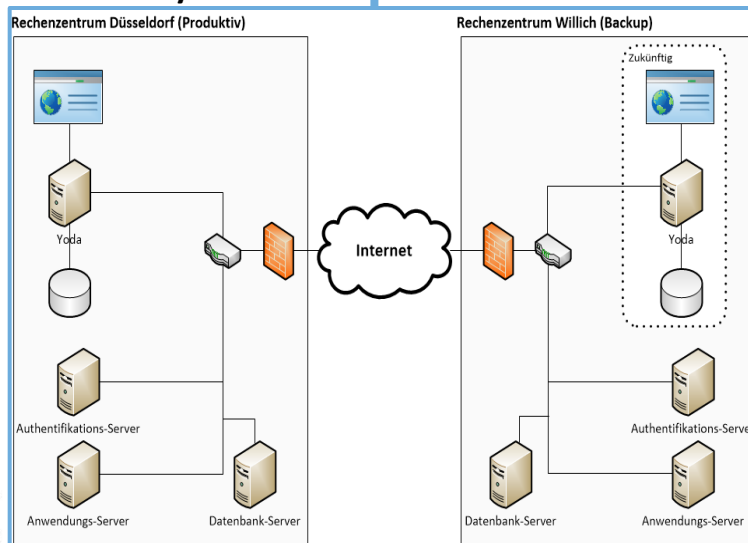
- JRuby (on Rails), ZeroMQ
- Entwicklungsumgebung: RubyMine
- Tools: MiniTest, simplecov



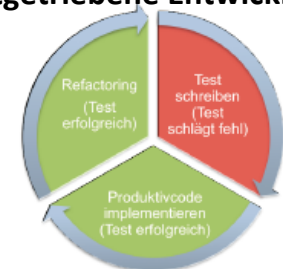
## Use-Cases in der Übersicht



## Systemskizze



## Testgetriebene Entwicklung



## Messung der Testabdeckung

```

23 @Override
24 public void provideComponents() {
25     Collection<RzObject> objects = new ArrayList<RzObject>();
26     RzSystemCommand cmd = new RzSystemCommand(command);
27     try {
28         Process process;
29         if(System.getProperty("os.name").toLowerCase().contains("win")){
30             process = Runtime.getRuntime().exec(command);
31         }
32         else{
33             String[] commandArray = new String[]{"./bin/sh", "-c", command};
34             process = Runtime.getRuntime().exec(commandArray);
35         }
36         cmd.setStdout(getStreamAsString(process.getInputStream()));
37         cmd.setStderr(getStreamAsString(process.getErrorStream()));
38         process.waitFor();
39         cmd.setExitCode(process.exitValue());
40     } catch (Exception e) {
41         logger.error("Komte Befehl nicht ausführen.", e);
42         cmd.setStderr().add(e.toString());
43     }
44     objects.add(cmd);
45     actualResults = objects;
46 }
    
```

## Realisierung Luke

- Java, ZeroMQ, Bash, Windows Batch
- Entwicklungsumgebung: Eclipse Luna
- Tools: JUnit, EclEmma

## Resultat

Ein Programmsystem, dass die Echtzeitdokumentation in verteilten Rechenzentren übernimmt. Die Echtzeitdokumentation erfasst Hardware- und Softwarekomponenten. Weitere Systeminformationen können per Web-Interface „on Demand“ angefordert werden. Das Programmsystem ist durch die Verwendung von Java-Technologien auch für andere Betriebssysteme erweiterbar.