



Enterprise Platform and Integration Concepts

Fachgebiet | Hasso-Plattner-Institut
Universität Potsdam



HART

Produktpräsentation

Isabel Amon, Björn Daase

08.02.2019

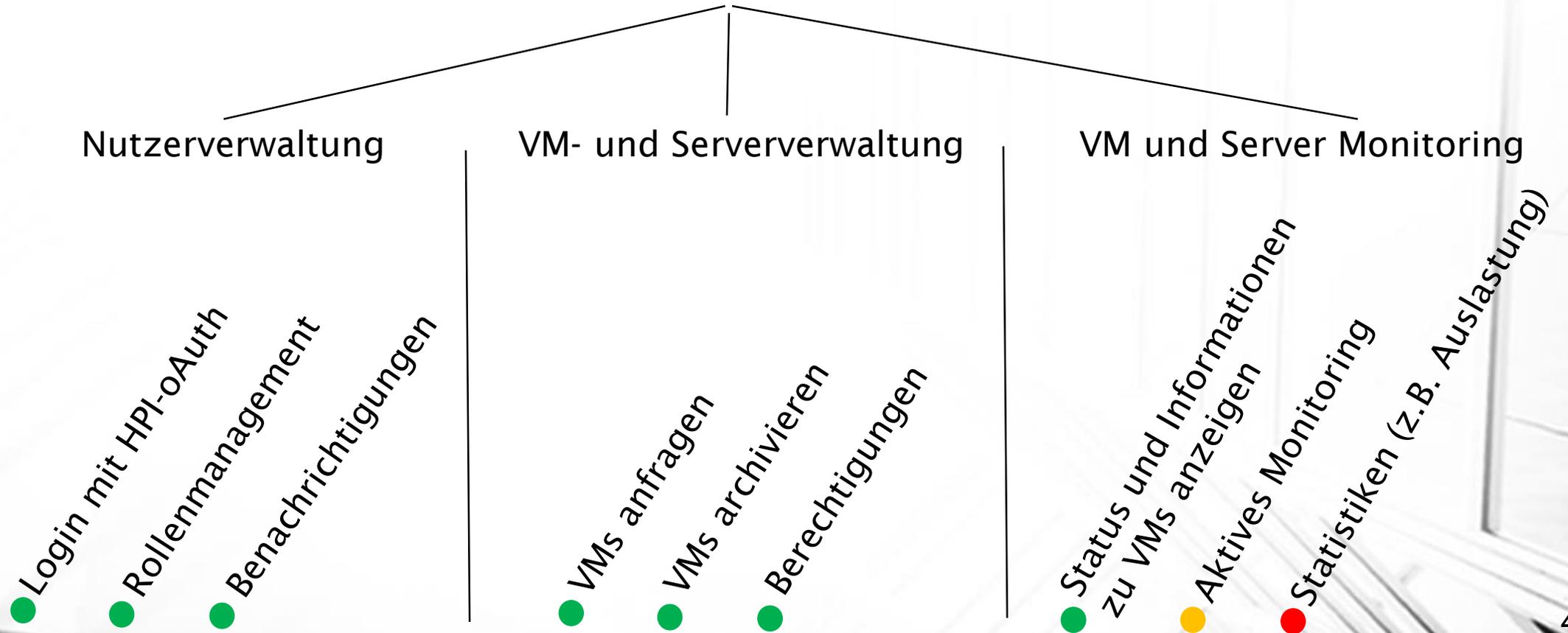
Live - Demo

Agenda

1. Vision
2. Architektur
3. Externe Abhängigkeiten
4. Deployment & Hosting
5. Ausblick
6. Prozess
7. Teams
8. Meetings und Schedule
9. Methoden

Vision

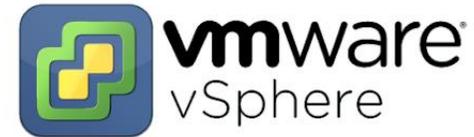
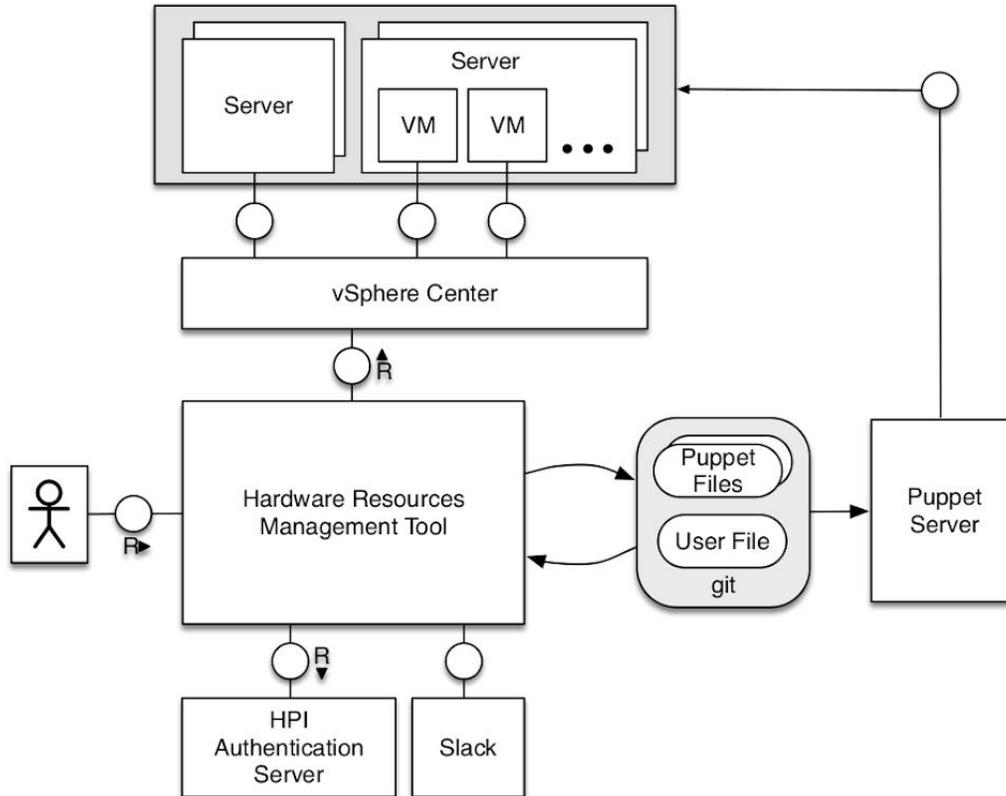
Wir wollen das Management von VMs und Servern des EPIC-Lehrstuhls zentralisiert und einfach möglich machen.



Architektur



Recap 15.11.18

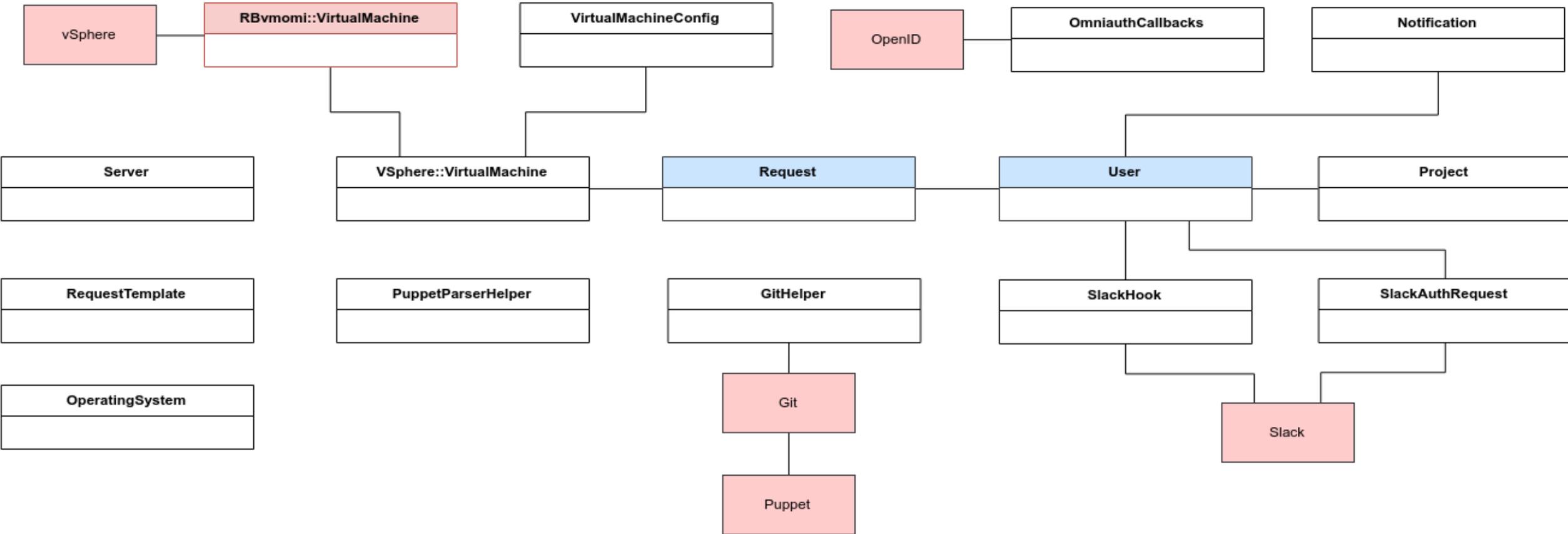


Abhängigkeiten/Kernsystem

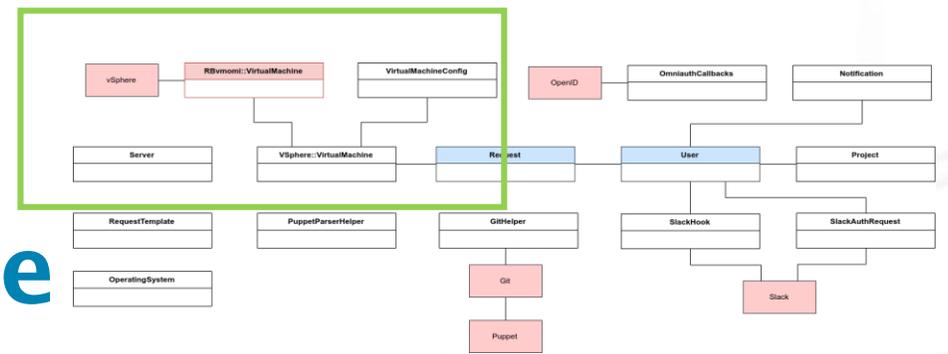
- **Motivation:**
 - Bewährte Konzepte nutzen
 - Nicht alle Teile selber schreiben
 - Integration in bestehende Systeme sowie Workflow
- VSphere, Git/Puppet, OpenID, Slack



Architektur



Abhängigkeiten - vSphere

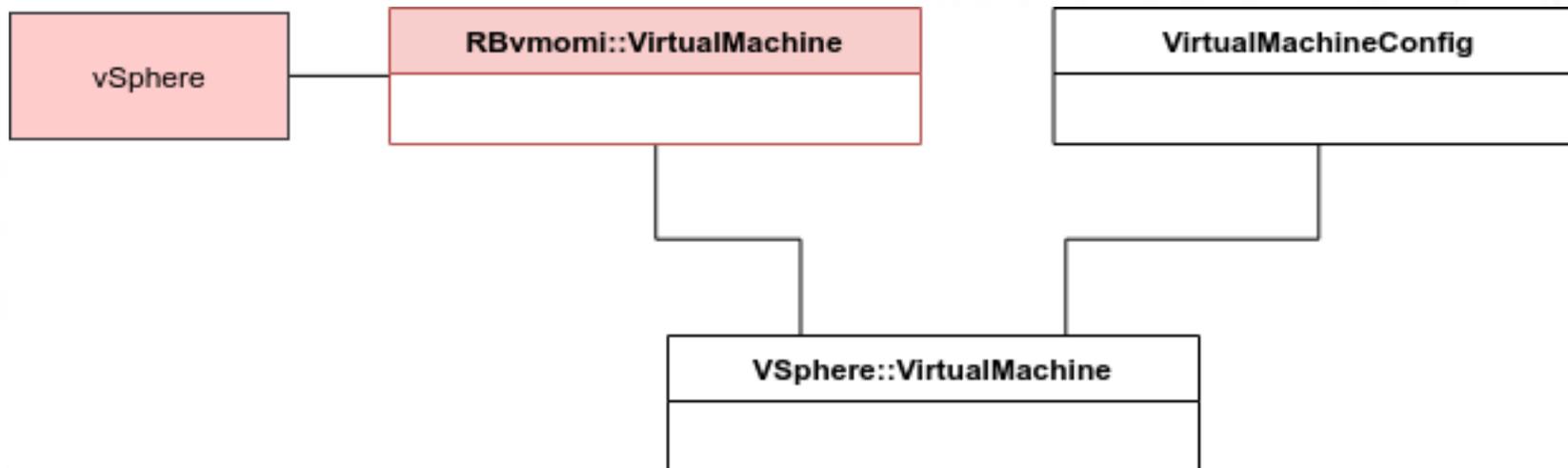


- Stellt eine grafische, administrative Verwaltung der gesamten Umgebung (VMs, Cluster, Hosts) zur Verfügung
- vSphere übernimmt die hardwareseitige Verwaltung der VMs
 - stellt VMs zur Verfügung
 - stelle Betriebssysteme für VMs zur Verfügung
- **Kommunikation HART - vSphere:**
 - Erstellen von VMs
 - Verwaltung der VMs in einer Ordnerstruktur
 - Anzeigen von Statistiken über VMs und Hosts

Synchronisation HART - vSphere



- **Challenge:** Daten über VMs mit vSphere synchron halten und trotzdem zu VMs eigene Daten halten



Puppet-Scripts in HART

- Bieten eine Möglichkeit, zentral die ACLs der einzelnen VMs zu verwalten
- Werden automatisch aus den Informationen der VM generiert
- Existierende Puppet Scripts werden geparst um die Datenbank zu populieren

Puppet Script



```
--- |
class node_test-bjoern-6 {
    $admins = ["Ad.Min", "Sarah-Isabel.Amon"]
    $users = ["Florian.Schmidt", "Sarah-Isabel.Amon", "Bjoerr

    realize(Accounts::Virtual[$admins], Accounts::Sudoroot[$
    realize(Accounts::Virtual[$users])
}

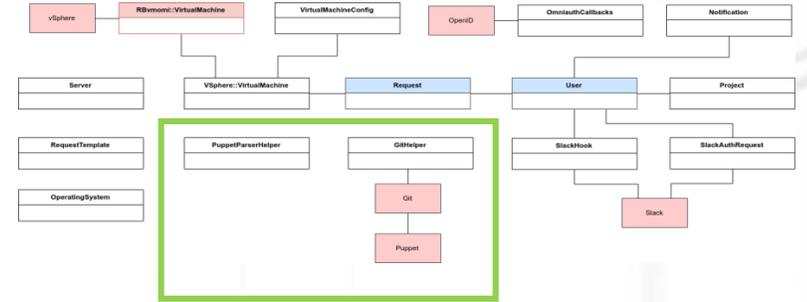
--- |
node 'test-bjoern-6'{

    if defined( node_test-bjoern-6) {
        class { node_test-bjoern-6: }
    }
}
```

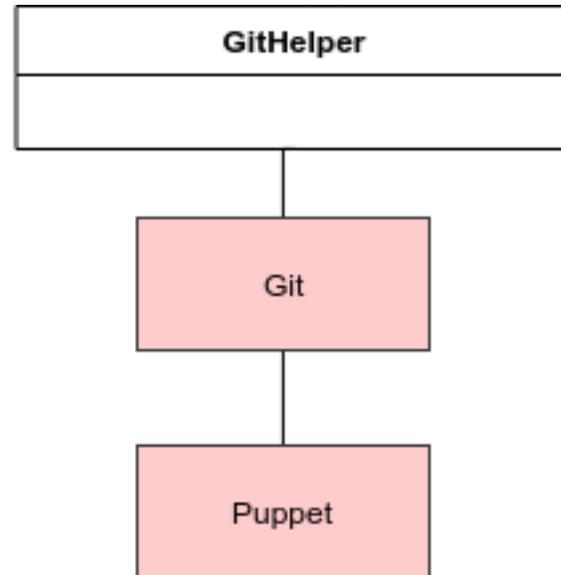
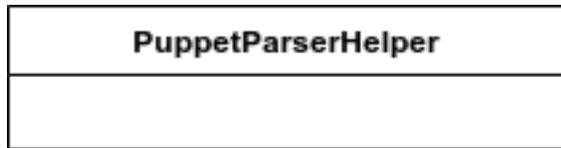
Push to git

Close

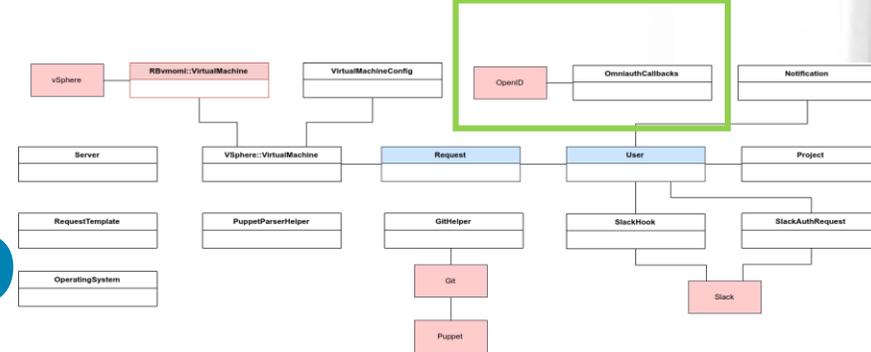
Abhängigkeiten - Git/Puppet



- Ein Github Repo wird verwendet, um die Puppet Scripts auf dem Puppet Server und auf dem Hart Server synchron zu halten.



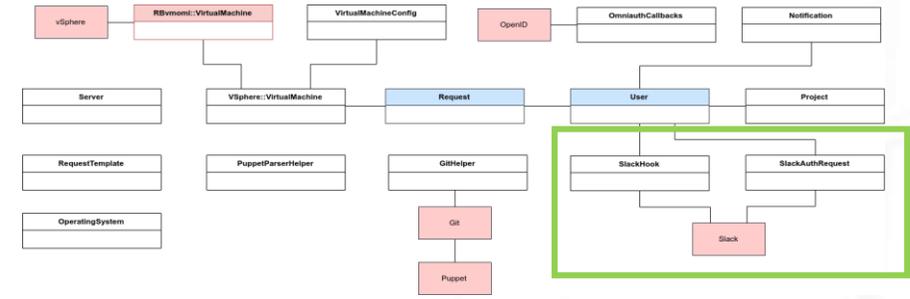
Abhängigkeiten - HPI OpenID



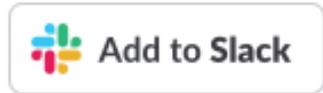
- Hintergrund:
 - Den Zugang zu HART auf HPI'ler beschränken
 - SSH-Zugang zu den VMs über HPI OpenID automatisieren
- Integration in sonstige HPI-Tools



Abhängigkeiten - Slack



- Weitere Notifikationsmöglichkeit neben internen Notifikationen und Mailbenachrichtigungen
- HART ist eine *vollwertige* Slack-App



- Slack bietet noch wesentlich mächtigere Möglichkeiten
 - z.B.: Akzeptieren von VM-Requests über Slack denkbar

HART interne Daten

- Zusatzinformationen zu:
 - VMs
 - User
- Server
- Projekte
- Requests
- Konfiguration von Requests

Nicht-softwaretechnische Abhängigkeiten

- Development



COVERALLS



Travis CI

- Hosting



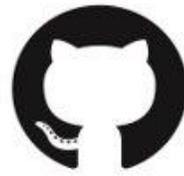
NGINX



Deployment

- **Vollständig automatisiert**

- Continuous Integration für Testing
- Für Development und Production



Travis CI

- **Branchstruktur:**

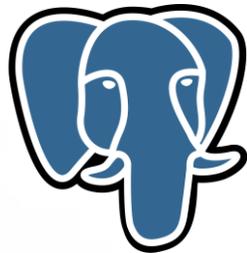
- Feature Branches → lokal, nicht deployed
- *Dev* - Branch → **deployed** nach jeder erfolgreichen Integration (hart-dev.epic-hpi.de)
- *Master* - Branch → **deployed** im 1 - 2 Wochen Rhythmus (hart.epic-hpi.de)

Deployment & Hosting

- Deployment: **Mina**
- Webserver: **Puma**
- Reverse Proxy Server: **NGINX**
- Datenbank: **PostgreSQL**



NGINX



Hosting



- **Deployte Versionen laufen in VMs auf Ubuntu 18.04.1 LTS**
- **VMs liegen auf EPIC-eigenen Servern im HPI-Netz**
 - Ermöglicht Erreichbarkeit der HPI-internen vSphere Daten
 - Vollständige Kontrolle über Sicherheit, Daten und Netzwerk

Monitoring

- Gesamtübersicht: <https://stats.uptimerobot.com/j8DADFQnv>
- Deployte Branches
 - Master: hart.epic-hpi.de
 - Dev: hart-dev.epic-hpi.de
- Deployment Details: hart-deploy.epic-hpi.de/deployed
- Deployment Error Collection: <https://errbit-vm-portal.herokuapp.com>

Monitors

Status	Last 7 Days	Type								
			01 Feb	31 Jan	30 Jan	29 Jan	28 Jan	27 Jan	26 Jan	
	100.00%	Deployer App	http	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	100.00%	Errbit VM-Portal	http	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	100.00%	Production deployment	http	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	100.00%	Staging deployment	http	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Overall Uptime

100.00% (last 24 hours)

(no downtime)

100.00% (last 7 days)

(no downtime)

99.733% (last 30 days)

(453 minutes of downtime)

Latest Downtime

It was recorded (for the monitor Staging deployment) on 2019-01-24 13:56:32 and the downtime lasted for 2 hrs, 29 mins.

Quick stats

 **All Clear**
Refreshing in 26 secs

 **Up** 4

 **Down** 0

 **Paused** 0

Qualität

- **Verlass auf ausgereifte APIs und externe Dienste**
- **Modulare Architektur**
- **Sehr hohe Testabdeckung (> 85%)**
- **Vollständig in englischer Sprache**
- **Usability auf verschiedenen Plattformen durch Bootstrap**

Ausblick

- **Experimente auf Servern ermöglichen**
 - Reservierung von Zeiträumen
 - Historie über durchgeführte Experimente
- **Erweitertes Monitoring von VMs und Servern**
 - z.B. Erreichbarkeit
 - Automatisches Warnen
- **Auswertungen und Statistiken**
 - Ressourcenverteilung (auf Nutzer)
 - Gesamtauslastung
 - Oft/Wenig genutzte Systeme, Hardware...

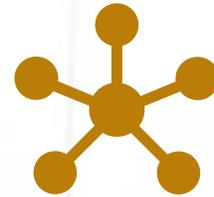
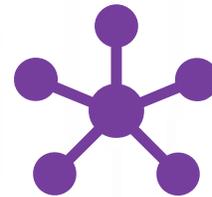
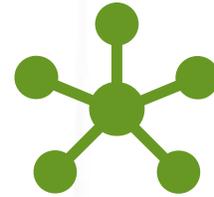
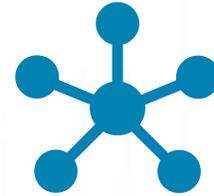
Prozess



HARTe Fakten

HARTe Fakten

- 4 Teams



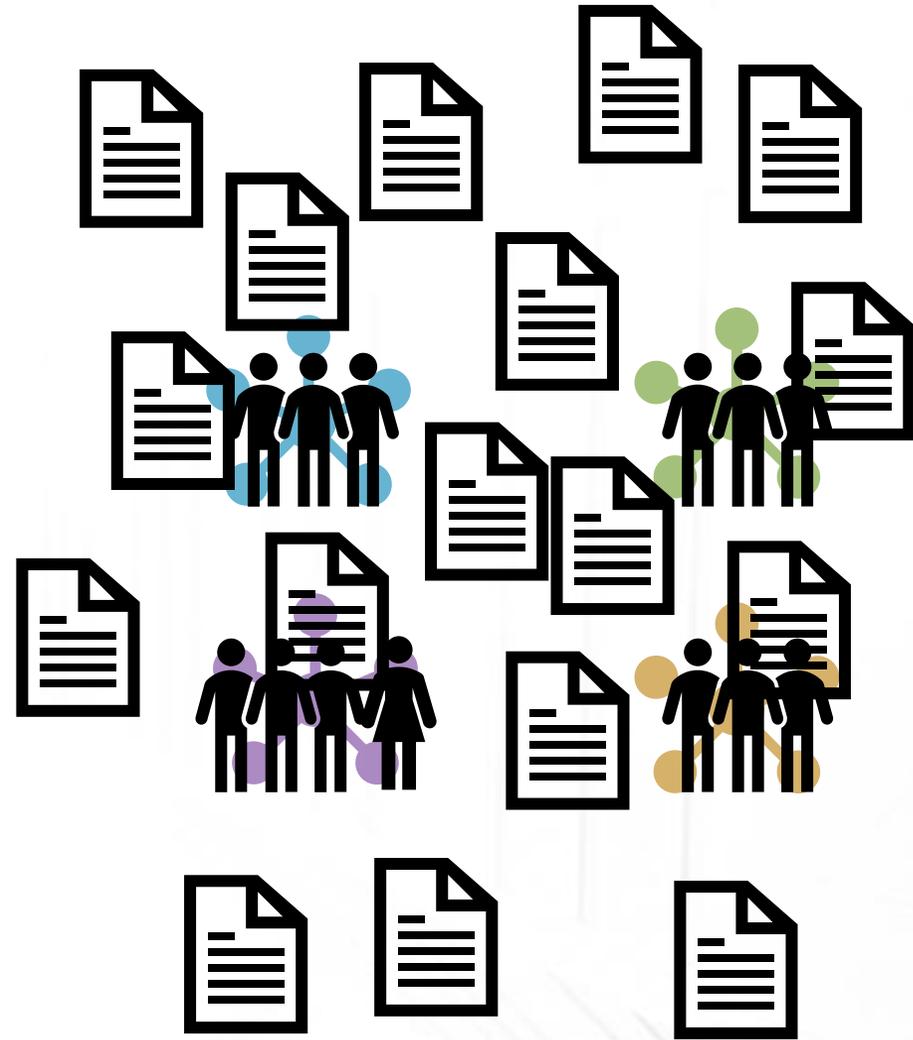
HARTe Fakten

- 4 Teams
 - 26 Developer



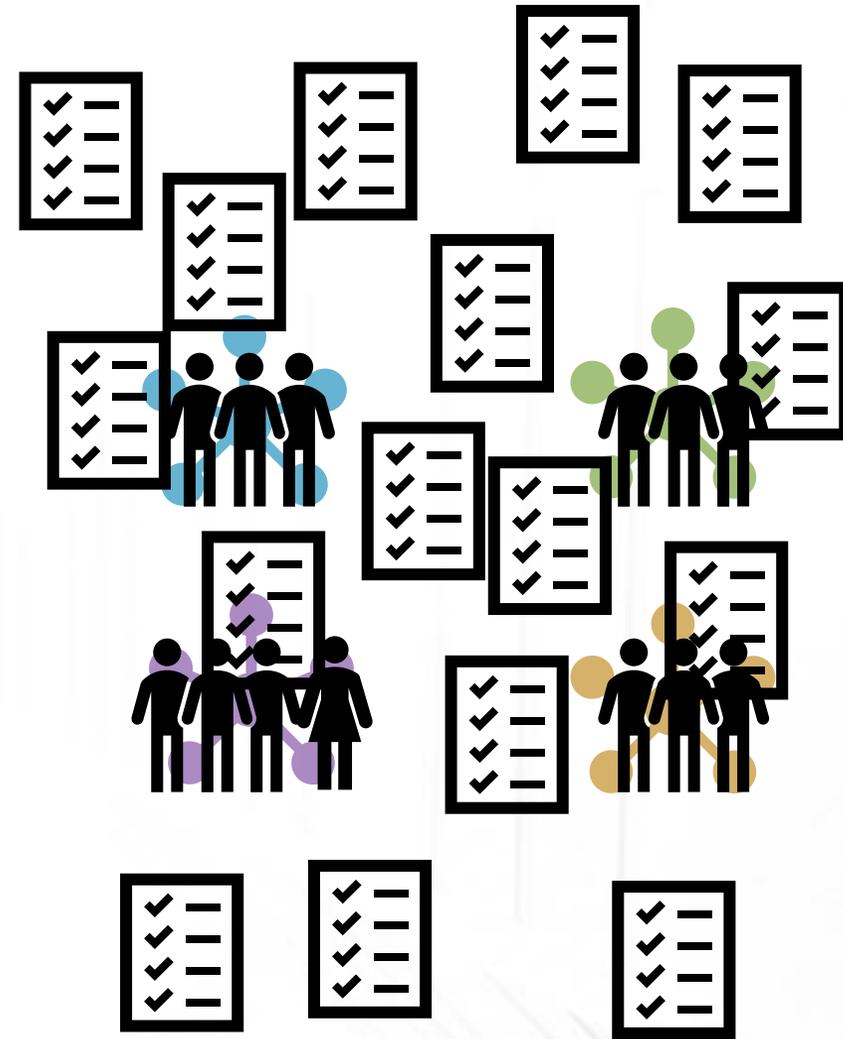
HARTe Fakten

- 4 Teams
 - 26 Developer
- 157 Issues



HARTe Fakten

- 4 Teams
 - 26 Developer
- 157 Issues
- 637 Commits



Scrum

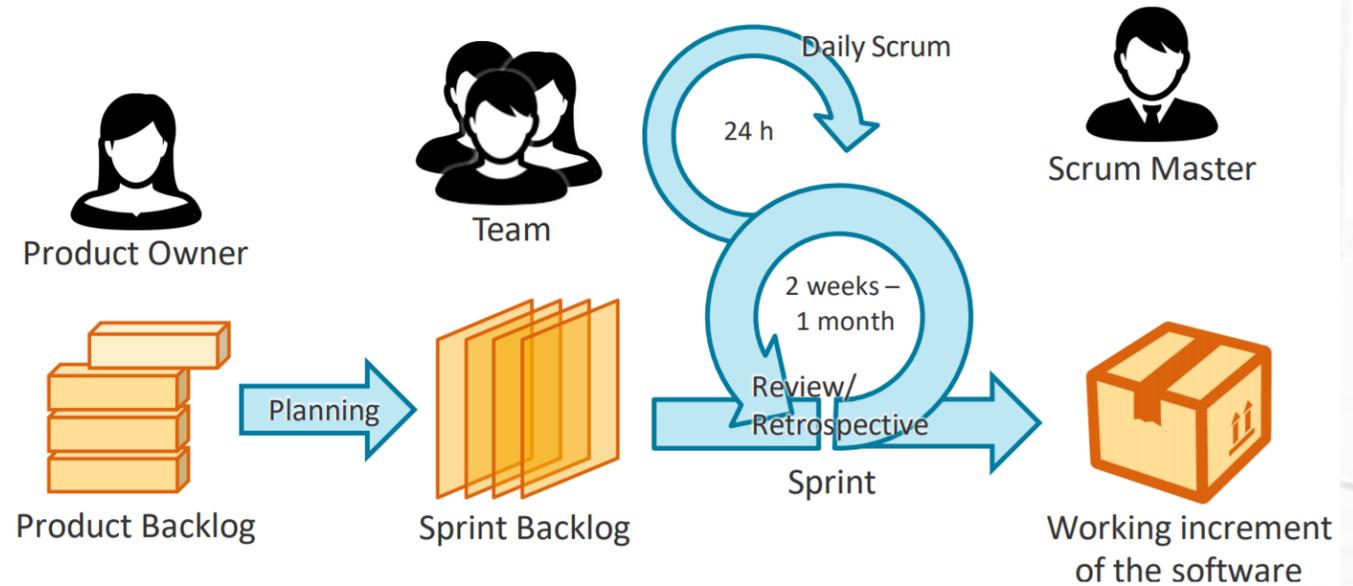
- Prozess zur agilen Softwareentwicklung

- 4 Teams

- 1 Product Owner
- 5-6 Entwickler
- 1 Scrum Master

- Sprints

- Planning
- 2-3 Wochen
- Weekly Scrum
- Review, Retrospektive



Team Oldies

- **Aufgabenbereiche:**
 - VM-/Host-/Server-Übersichtsseiten
 - VM-/Host-/Server-Detailseiten
 - Server anlegen
 - VMs über HART starten/stoppen lassen
- **Heldentaten: User-Interface für fast alle Seiten erneuert**

Team Scaffold

- **Aufgabenbereiche:**
 - Requestprozess
 - Konfiguration für Requests
 - Templates
 - Operating Systems
 - Archivieren/Wiederbeleben von VMs
 - Notifications – Slack und Mail
 - vSphere API
- **Heldentaten: Travis-Laufzeit um 75% reduziert**

Authentoguys

- **Aufgabenbereiche:**
 - Nutzerverwaltung
 - Rollenmanagement
 - Nutzerübersicht
 - Projekte
- **Heldentaten: HPI openID-Login**

Team Vertical

- **Aufgabenbereiche:**
 - Puppet Scripts
 - User Interface
 - Logo
 - Name
 - Pagedesign
 - Interne Notifications
- **Heldentaten: Automatische Git-Synchronisierung der Puppet Scripts**

Schedule - Gesamtübersicht

	November	Dezember	Januar	Februar	
	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Kanban
Vertical	Grundlegendes UI	Notifications	Puppet Scripts Dashboard	Puppet Scripts	Puppet Scripts
Authento	Manueller Login Logout	Zugriffs- beschränkungen	Rollen- management	HPI openID Login	WebKit-Browser reparieren
Scaffold	VM-Erstellung	Request-Prozess	Archivierung/ Wiederbeleben	Request-Prozess	Notifications - Mail
Oldies	VM- und Hostlisten	Start und Stopp von VMs	Server anlegen	Listen überarbeiten	Requestliste

Schedule - Sprint

Montag	Mittwoch	Freitag	...	Montag	Mittwoch	Freitag
Sprint-Start		Weekly Scrum				Review
PO-Weekly				PO-Weekly	Kundenmeeting	Retrospektive
PO-Planning				Pre-Grooming		Planning

Product Owner

PO & Stakeholders

Team

Methoden

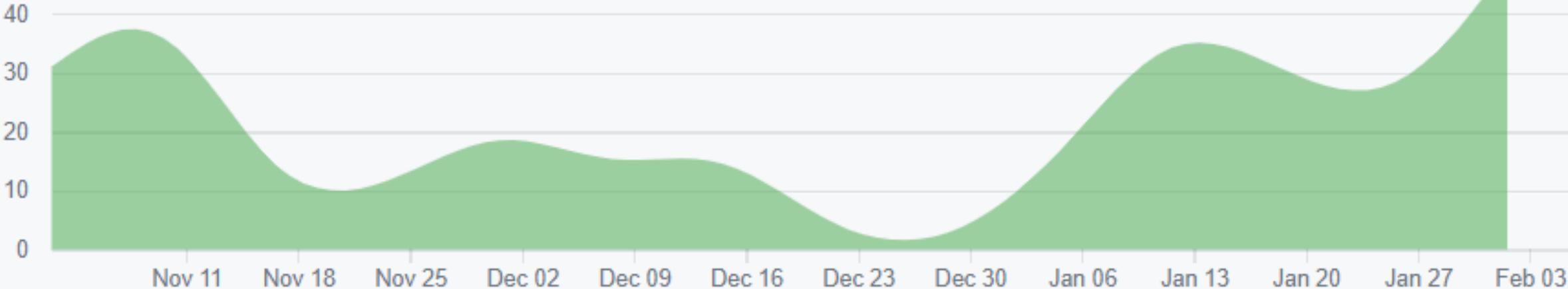
Developer

- Code-Reviews
- Pair-Programming
- Kommunikation zwischen Teams
 - Slack
- Einholen von Profi-Hilfe

Product Owner

- Paper Prototyping
- Mock-Ups
- User Stories gemeinsam schreiben
- Shadowing
- Interteam-Kommunikation

Commits



Sprint 1

Sprint 2

Sprint 3

Sprint 4

Kanban

November

Dezember

Januar

Februar

Kick-Off

Zwischenpräsentation



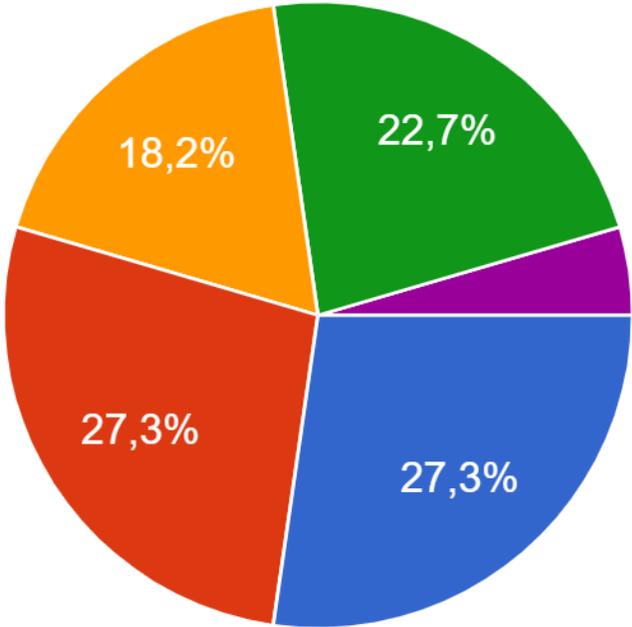
HART

Megaretro

Arne Zerndt, Florian Schmidt
08.02.2019



Umfrage – 22 Antworten



- Team Oldies
- Team Vertical
- Authentoguys
- Team Scaffold
- SWT Teaching team

7 Dinge die wir gelernt haben:

- Scrum (40%)
- Ruby on Rails (36%)
- Kommunikation im Team und mit anderen Teams (31%)
- TDD und Testing im Allgemeinen (31%)
- Arbeit an einem Multi-Team-Projekt (18%)
- Affinity Estimation (9%)
- Deployment (9%)

5 Fragen die noch offen sind:

- Wie funktioniert Rails hinter den Kulissen? (23%)
- Wie kann man TDD konsequent anwenden? (7%)
- Wie sieht eine gute Server-Architektur aus? (7%)
- Wie ändert sich Projektarbeit, wenn man Vollzeit arbeitet? (7%)
- Wie gehe ich mit entfernt arbeitenden Kollegen um? (7%)

Was lief richtig gut?

- Teamwork! (38%)
- Team Atmosphäre (22%)
- Hilfsbereitschaft von Kollegen (27%)
- Kommunikation innerhalb der Teams (22%)
- Arbeit der POs (17%)

Wodurch wurden wir aufgehalten?