



redhat.

**RHQ**

Heiko W. Rupp <[heiko.rupp@redhat.com](mailto:heiko.rupp@redhat.com)>

SIG-JBoss 26.6.2008

# Inhalt

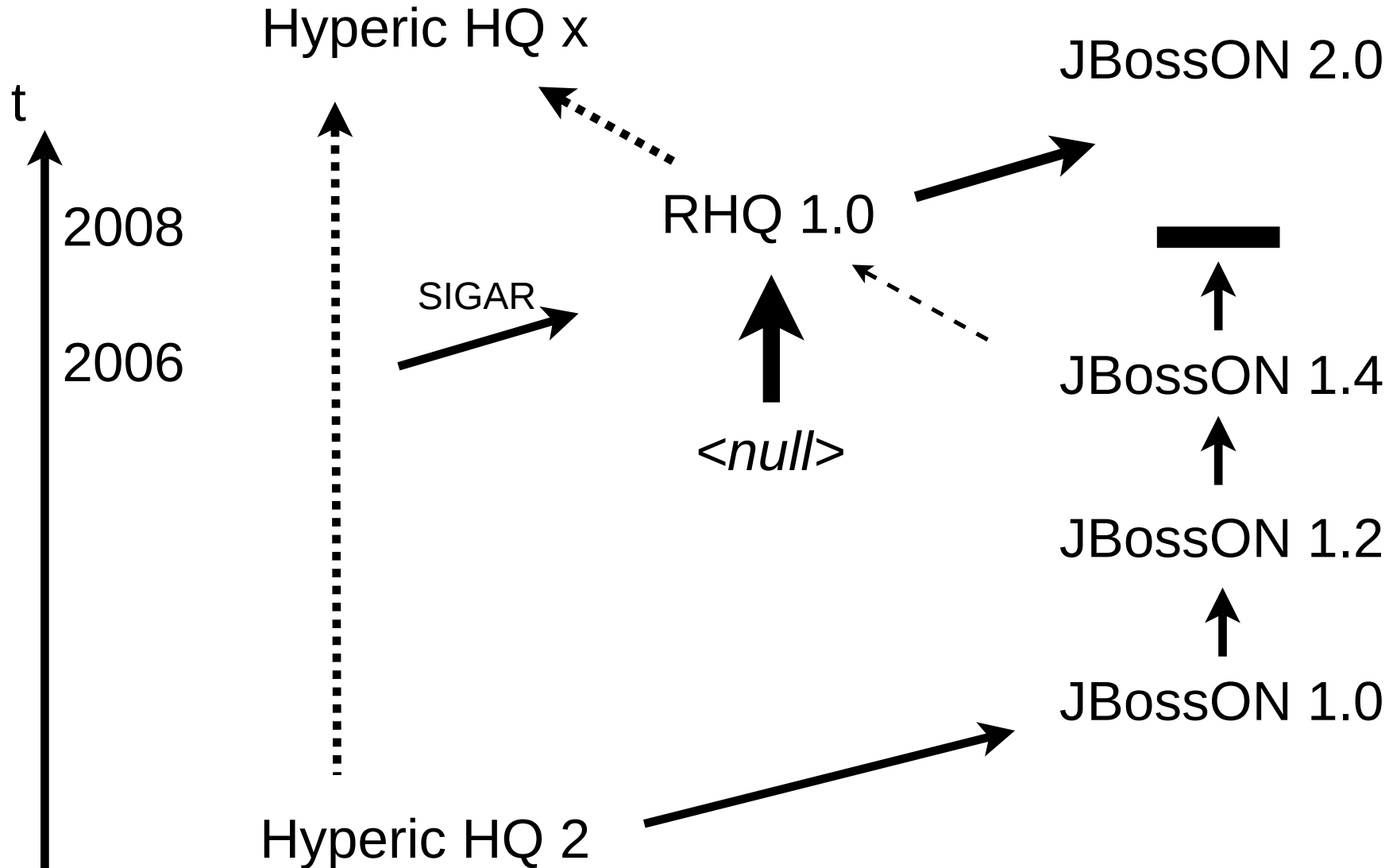
- Einführung
- Historie
- Architektur
- Subsysteme
- Demo
  
- Entwicklung von Plugins



# Einführung

- RHQ - Gemeinsames Projekt von Red Hat und Hyperic
- RHQ ist die Basis für JBoss Operations Network 2.0
- RHQ ist Open Source seit Februar 2008
  - Einstiegsseite: <http://www.rhq-project.org/>
  - Forums: <http://forums.rhq-project.org/>
  - JIRA: <http://jira.rhq-project.org/secure/Dashboard.jspa>
  - Wiki:
  - SVN: <http://www.rhq-project.org/svn.php>
  - Plugin-Community: <http://support.rhq-project.org/display/RHQ/RHQ+Plugin+Community>
- Commit-Zugang benötigt Contributor-Agreement

# Historie

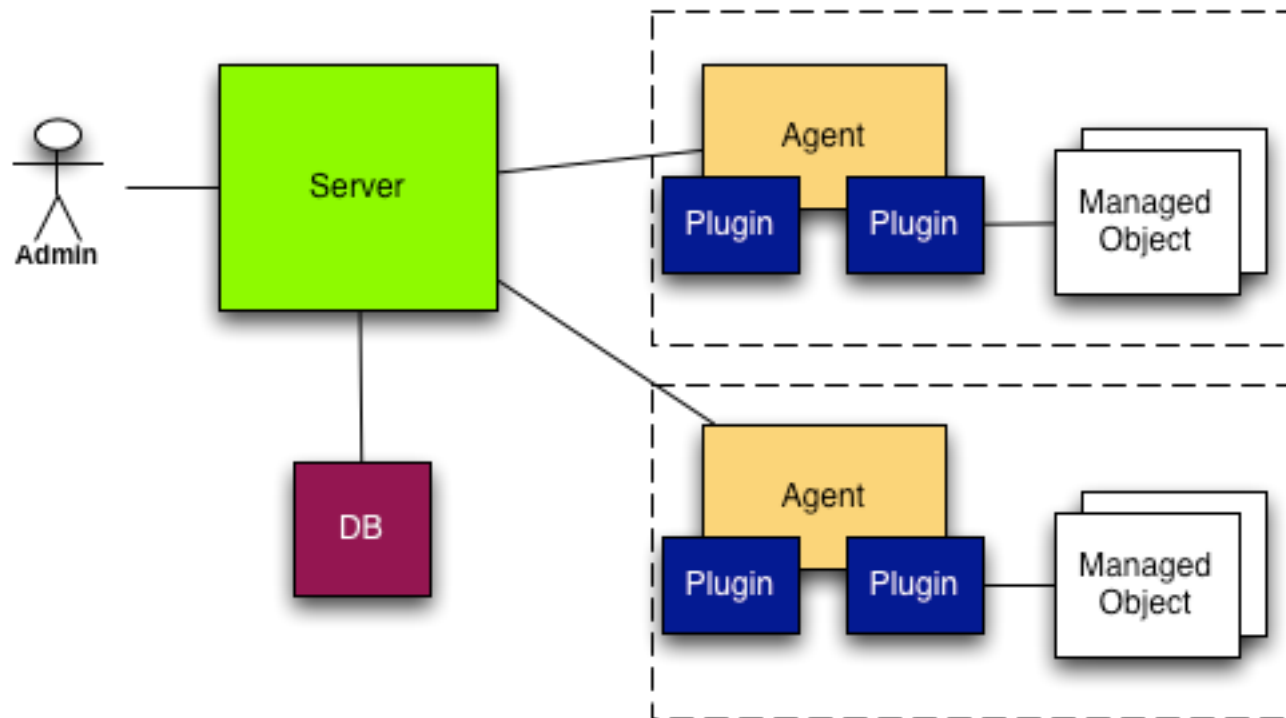


## RHQ 1.0

- Basiert zu sehr großen Teilen auf neugeschriebenem Code
  - JBoss AS 4.2.1
  - JBoss Remoting
  - EJB 3
  - JPA / Hibernate
  - JSF / Richfaces
- SIGAR Library von Hyperic
- Teile der GUI noch aus JBoss ON 1.x

# Architektur

- Zentraler Server
- Agent pro Maschine („platform“)

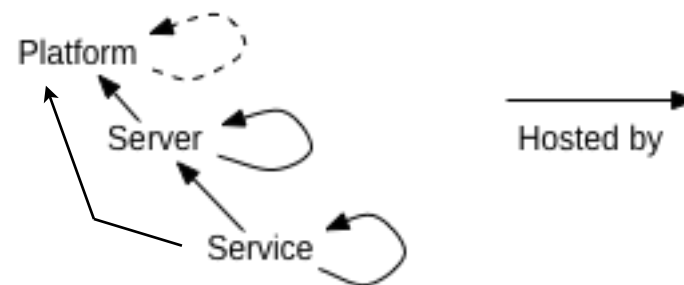


# Subsysteme

- Monitoring
  - Messungen von Zuständen
  - Grafiken auf dem Server
- Inventory
  - Zentrale Komponente, die alle zu managenden Ressourcen kennt
- Configuration
  - Konfiguration von Ressourcen
- Alerts
  - Benachrichtigung von Menschen
- Operations
  - Operationen auf Ressourcen ausführen
- Packages
  - Softwaredownload auf Ressourcen

# Ressourcen

- Ressourcen
  - Alles was gemanaged werden kann
- Ressource Type
  - Z.B. Linux, Max OS X, JBoss AS, Hibernate Stats, DB-Tabelle
- Ressource Category
  - Platz in der Hierarchie
  - Vater - Kind - Beziehung
  - Platform = Maschine



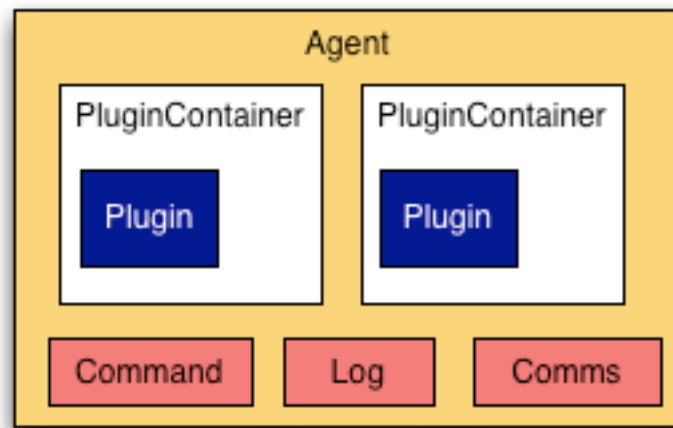




# Demo

# Plugins entwickeln

- Plugins leben im Plugin Container

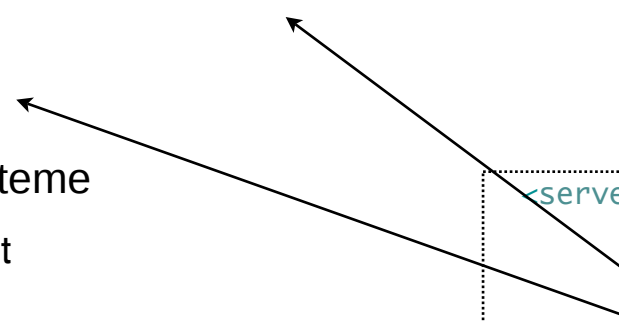


- Hot-Deploy über Server
  - Agent lädt herunter und restartet den Plugin-Container

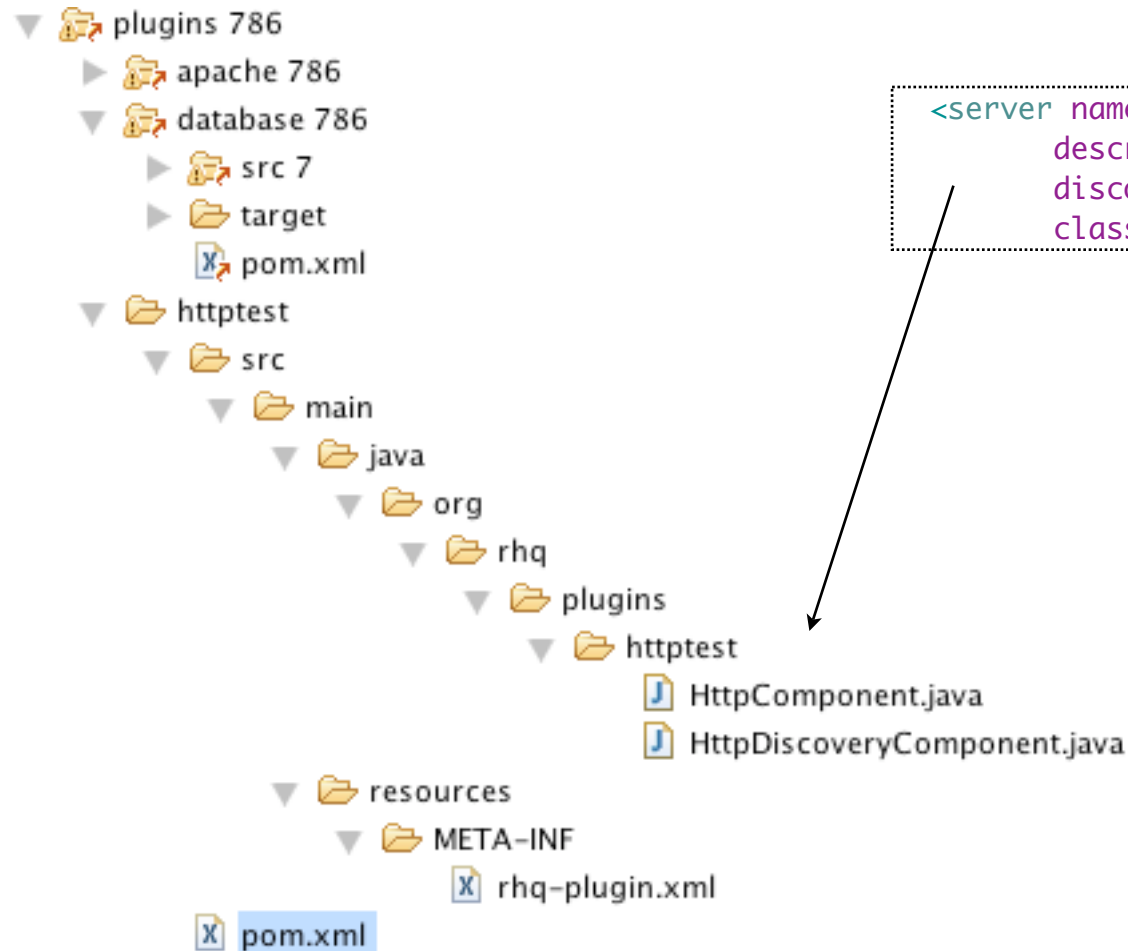
## 3 Komponenten eines Plugins

- Discovery Component
  - Zuständig für das Entdecken / Hinzufügen neuer Ressourcen
  - Ressource-Key muss eindeutig sein und bleiben !
- Plugin Component
  - Implementiert *Facets*
  - Prinzipiell wie Subsysteme
    - MeasurementFacet
    - OperationsFacet
    - ...
- Plugin-Deskriptor
  - Definiert Fähigkeiten des Plugins als Metadaten
  - Verdrahtet die Java-Klassen

```
<server name="HttpCheck"  
description="Httpserver pinging"  
discovery="HttpDiscoveryComponent"  
class="HttpComponent">
```



# Paketstruktur



```
<server name="HttpCheck"
  description="Httpserver pinging"
  discovery="HttpDiscoveryComponent"
  class="HttpComponent">
```

# Plugin-Generator

- Siehe <http://pilhuhn.blogspot.com/2008/06/rhq-tip-of-day-use-skeleton-generator.html>
- Legt pom + Deskriptor an
- Rest muss aktuell von Hand gemacht werden
- (Kommt aber sicher noch :-)

## Weiteres ...

- JBoss ON 2 Doku: <https://docs.jbosson.redhat.com/confluence/display/JON2/Home>
- JBoss ON Homepage: <http://www.jboss.com/products/jbosson>
- JBoss ON Forum: <http://www.jboss.com/index.html?module=bb&op=viewforum&f=285>
- Blog-Postings über Plugin-Entwicklung: <http://pilhuhn.blogspot.com/2008/05/writing-rhq-plugin-part-1.html> + verlinkte Posts