



JBug OWL

JBoss Operations Network

Kamil Swierkot
kswierkot@s-und-n.de

Was ist JBoss Operations Network?

- ◆ JON ist ein zentrales Tool um Applikationen und Server zu managen, Monitoren und deployen
 - ◆ Inventar: Automatische Ressourcen Erkennung
 - ◆ Konfiguration: Ressourcen konfigurieren
 - ◆ Monitoring:
 - ◆ Graphen von Metriken einzelner Ressourcen
 - ◆ Alerting
 - ◆ Operations
 - ◆ Deployments, Patching von Servern
 - ◆ Start, Stop, Restart Servers

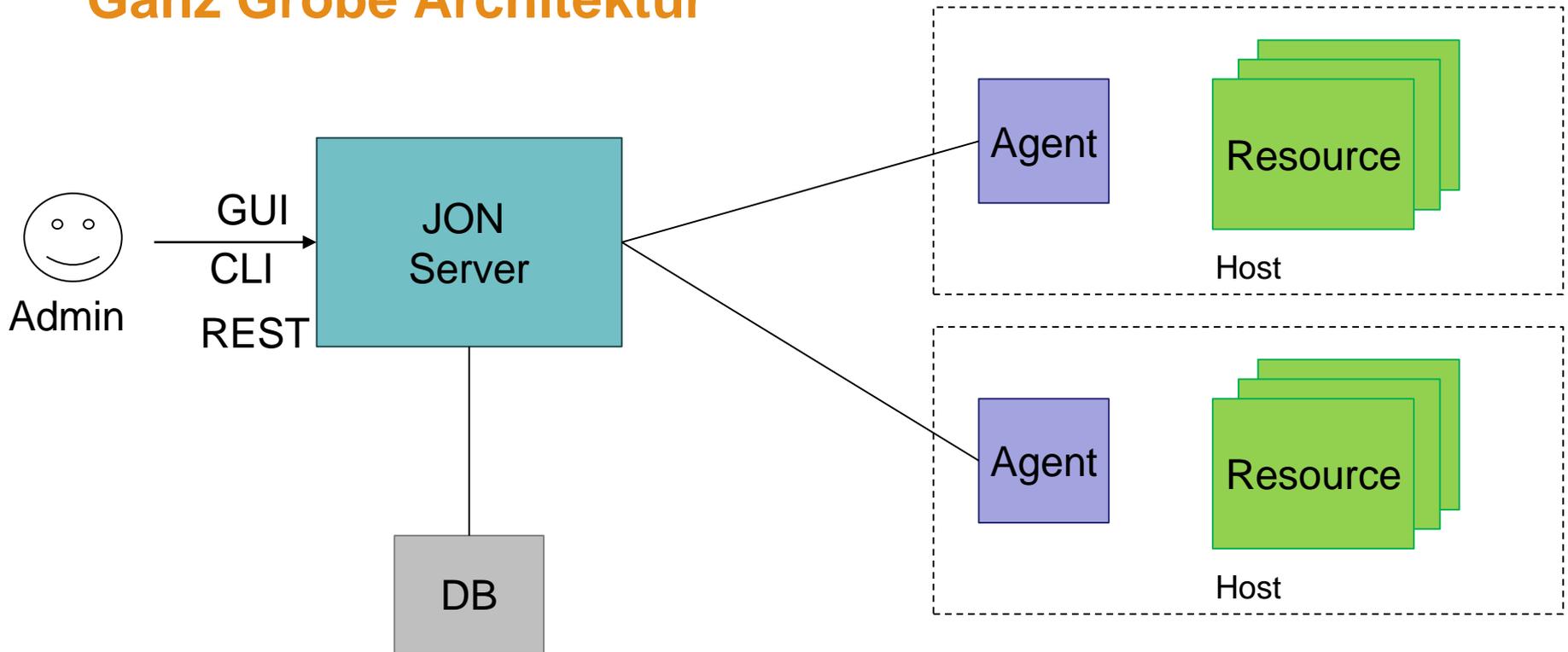
Upstream Project

- ◆ JON hat ein Open Source Upstream Project

<http://rhq-project.github.io/rhq/>

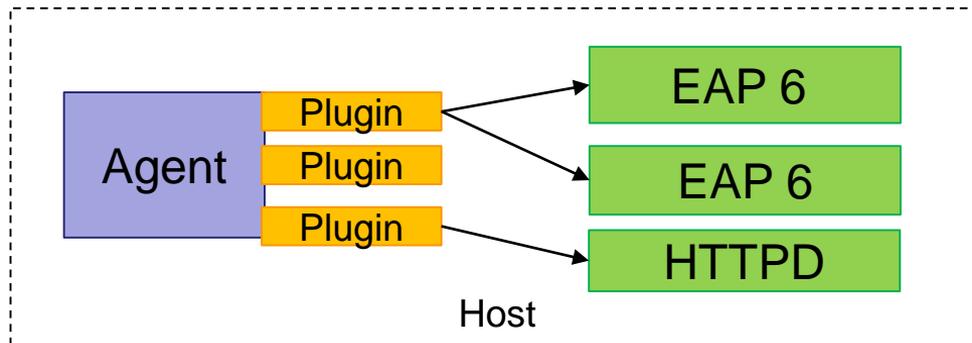
- ◆ JON 3.3 basiert auf RHQ 4.12

Ganz Grobe Architektur



Agenten

- Agenten sammeln mit Hilfe von Plugins Daten über laufende Prozesse
=> Agenten sind Plugin Container



- Agenten schicken Daten zum JON Server
- Server speichert Daten ab

Agenten – Nicht nur EAP

- ◆ Agenten Plugins (= JARs) werden in Packs vertrieben
 - ◆ EAP (EAP, RHQ Server, JBoss AS 7, HornetQ)
 - ◆ A-MQ
 - ◆ BPM Suite / BRMS
 - ◆ ...
- ◆ Man kann auch selber Agenten entwickeln
- ◆ Liste aller Open Source Plugins:
<https://docs.jboss.org/author/display/RHQ/Plugin+Community>

Datenpunkte - Aggregates

- ❖ Alle Metriken werden als Datenpunkte mit Zeit, Resource und Wert abgespeichert
- ❖ Datenpunkte können nicht für immer aufbewahrt werden
- ❖ Tradeoff zwischen Speichergröße und Präzision
 - ❖ Raw Data -> 1 Stunden Aggregate
 - ❖ 1 Stunden -> 6 Stunden Aggregate
 - ❖ 6 Stunden -> 24 Stunden Aggregate

Datenpunkte - Probleme

- ◆ Oft mehrere Metriken pro Ressourcen
- ◆ Oft Mehrere Ressourcen pro Host / Agent
- ◆ Oft Mehrere Server
- ◆ => Sehr viele Datenpunkte pro Intervall müssen abgespeichert werden
- ◆ Es gibt 2 Möglichkeiten
 - ◆ Scale-up: Bessere Hardware, Bessere Datenbanken
 - ◆ Scale-out: (Mehrere) Einfache Server, Andere Datenbanken

Storage

- ◆ JON (≥ 3.2) hält Daten in einem Hybriden Ansatz:
 - ◆ Konfigurationen etc. werden weiterhin in einer RDMS gespeichert
 - ◆ Datenpunkte werden ab JON 3.2 nicht mehr in einer Relationalen Datenbank gespeichert.
 - ◆ Datenpunkte werden in einer Cassandra Datenbank gehalten

=> neue Komponente **Storage**

Storage – Warum Cassandra?

- ❖ Optimiert für hohen Schreibdurchsatz
- ❖ Mit einer oder mehreren (gleichartigen) Nodes nutzbar
- ❖ Nodes können später ergänzt werden
- ❖ Daten haben Ablaufdaten
- ❖ Gut für Zeitreihen analyse
- ❖ Map Reduce Jobs (Datenaggregate)
- ❖ In Java Implementiert

Quelle: http://www.jboss.org/dms/judcon/2013unitedstates/presentations/judcon2013_day2track2session2.pdf

Demo

Vielen Dank !

Kamil Swierkot
kswierkot@s-und-n.de