

A green Muppet character, resembling a scientist or developer, is the central focus. He has a friendly expression and is wearing black-rimmed glasses. He is dressed in a light blue lab coat over a green sweater and a red tie. In his left hand, he holds a blue book with the words "MUPPET LABS" printed on the cover. His right hand is raised, with his index finger pointing upwards. The background is a white surface with a pattern of blue polka dots.

# Puppet für Entwickler

Sebastian Hempel

# Sebastian Hempel

IT-Consulting Hempel

 <https://it-hempel.de>

 <https://github.com/ithempel>

 @ithempel

Puppet-Master seit 2010



# Build Manager



# Continuous Integration



One pimping continuous integration light (CC BY) | [Tim Lucas](#)

# Test-Umgebung



# Deployment



# Was ist Puppet?



# Wie arbeitet Puppet?

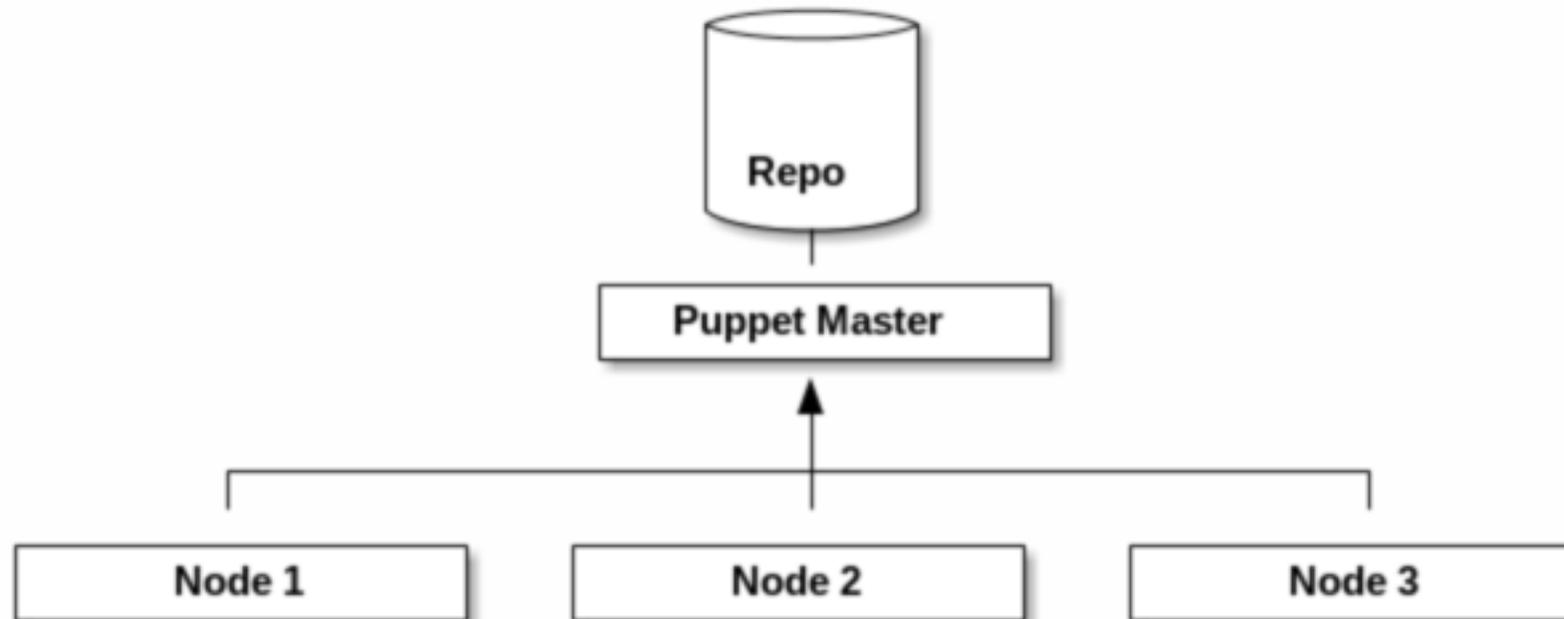
deklarativ

Wie soll es aussehen?

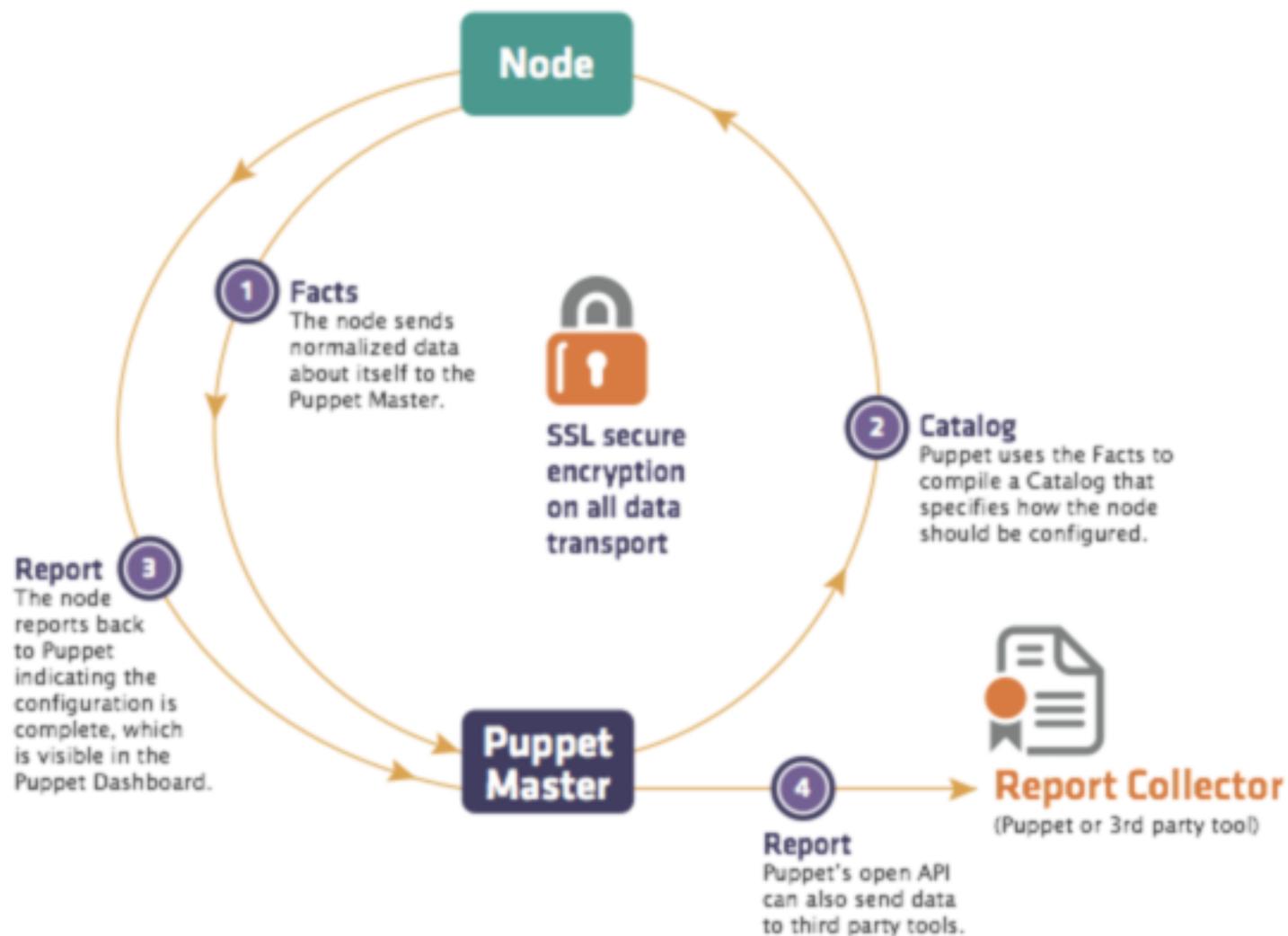
idempotent

mehrfach anwendbar

# Architektur von Puppet



# Ein Puppet-Lauf



# Puppet Language

## Resource

---

```
file { '/etc/default/jenkins':  
  ensure => present,  
  source => 'puppet:///modules/jenkins/etc/default/jenkins',  
  owner  => 'root',  
  group  => 'root',  
  mode   => '0644',  
}
```

---

# Puppet Language

## Conditional

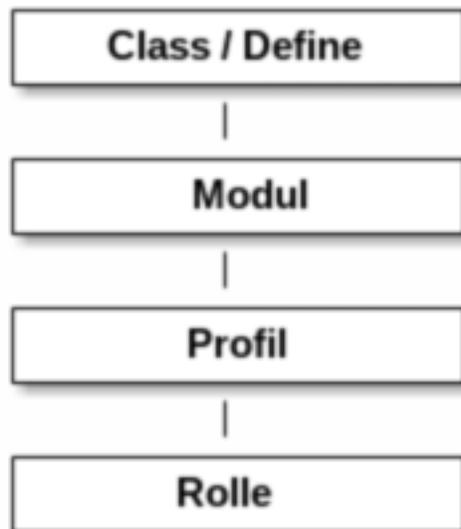
---

```
if $::os['name'] == 'Debian' {  
  ...  
}
```

---

# Puppet Language

## Strukturierung



# Trennung von Code und Daten

## Code

Aus welchen Elementen besteht die Konfiguration?

## Daten

Konfigurationswerte / Attribute der Elemente

# Reuse - Puppet Forge

<https://forge.puppetlabs.com/>

über 3000 Module  
Puppet Supported  
Puppet Approved

# Build-Systeme



Grace Murray Hopper at the UNIVAC keyboard, c. 1960 (CC BY) | [Carl Malamud](#)

# Java

## Modul: puppetlabs-java

---

```
include 'java'
```

---

build-server.yaml

---

```
---
```

```
java::distribution: 'jdk'
```

---

# Build-System

## Modul: maestrodev-maven

---

```
include 'maven'  
maven::settings { 'global-settings':  
  mirrors => hiera('maven::mirrors'),  
  servers => hiera('maven::servers'),  
}
```

---

### build-server.yaml

---

```
---  
maven::maven::version: '3.2.5'  
maven::mirrors:  
  - 'central':  
    id: 'central'  
    url: 'https://local.nexus.example.com/nexus/content/groups/public/'  
    mirrorOf: 'central'  
maven::servers:  
  - 'snapshot':  
    active: 'true'  
    username: 'deployment'
```

---

# Git

## Konfiguration über Template

---

```
package { 'git':  
  ensure => installed,  
}  
file { '/var/lib/jenkins/.gitconfig':  
  ensure => file,  
  content => template(git/gitconfig.erb),  
  owner   => 'jenkins',  
  group   => 'jenkins',  
  mode    => '0644',  
}
```

---

gitconfig.erb

---

```
[user]  
  name = Build Server  
  email = jenkins@company.org  
[push]  
  default = current
```

---

# Jenkins

## Modul: rtyler-jenkins (Puppet Approved)

---

```
include 'jenkins'
```

---

build-server.yaml

---

```
---
jenkins::install_java: false
jenkins::lts: true
jenkins::config_hash:
  JENKINS_HOME:
    value: '/srv/jenkins'
jenkins::plugin_hash:
  git:
    version: '2.3.4'
  credentials:
    version: '1.20'
  ssh-credentials:
    version: '1.10'
  git-client:
    version: '1.15.0'
```

---

# Test- und Abnahme-Systeme



# Application-Server

Konfiguration von  
DataSource  
JNDI Parameter  
Subsystem (z.B. JMS)  
User Accounts

# Konfigurationsdateien

---

```
file { '/opt/wildfly/standalone/configuration/standalone.xml':  
  ensure => file,  
  content => template('wildfly/standalone.xml.erb'),  
  owner   => 'wildfly',  
  group   => 'wildfly',  
  mode    => '0640',  
}
```

---

standalone.xml.erb

---

```
<datasources>  
  <datasource jndi-name="java:jboss/datasources/<%= @database_name %>"  
pool-name="Applicaton" enabled="true" use-java-context="true">  
  <connection-url>jdbc:mysql://localhost/<%= @database_name %>  
ction-url>  
  <driver>mysql</driver>
```

---

# Command Line Tools

## Modul: cpitman/puppet-jboss\_admin (GitHub)

---

```
jboss_admin::server {'main':  
  base_path => '/opt/jboss'  
}
```

```
jboss_admin::resource::datasource{"/subsystem=datasources/data-  
source=${database_name}":  
  ensure      => present,  
  connection_url => "jdbc:mysql://localhost/${database_name}",  
  driver_name  => mysql,  
  jndi_name    => "java:jboss/datasources/${database_name}",  
  jta         => true,  
  user_name   => sa,  
  password    => sa,  
  server      => main  
}
```

---

# weitere Server (z.B. Datenbank)

## Modul: puppetlabs-mysql

---

```
include 'mysql::server'  
mysql::db { 'application_database':  
  user      => 'db_user',  
  password => 'db_password',  
  host      => 'localhost',  
  grant     => [ 'INSERT', 'UPDATE', 'SELECT', 'DELETE', ],  
}
```

---

### build-server.yaml

---

```
mysql::server::override_options:  
  mysqld:  
    bind-address: '127.0.0.1'  
    datadir: '/srv/mysql'
```

---

# Development → Operating



Gene Kim and Patrick Debios (CC BY-NC-SA) | [Matt Moor](#)

# Development

Entwicklung = Produktion  
Test der Konfiguration  
Feedback vom Operating

# Operating

Konfiguration ist getestet  
direkte Übernahme  
kein Spezialist notwendig  
dokumentiert

# DevOp Kultur

Shared Tool

Austausch

Vertrauen

**Zusammen**arbeit

# Konfiguration ist Code



# Versionsverwaltung

“ | *Code gehört unter  
Versionsverwaltung!*

# Qualitätssicherung

(Unit-)Tests  
Codeanalyse  
Refactoring



Fragen?