

# Puppet und SVN für das Configuration- und Changemanagement

LinuxTag 09.06.2010 - Berlin

*Referent: Michael Streb*

## Agenda

DESTINATION	TIME	REMARK
KURZVORSTELLUNG	3 SLIDES	ON TIME
WAS IST PUPPET?	10 SLIDES	ON TIME
WAS IST SUBVERSION?	2 SLIDES	ON TIME
WARUM DIESE TOOLS?	1 SLIDE	ON TIME
WELCHE VORTEILE?	1 SLIDE	ON TIME
UMSETZUNGSMÖGLICHKEIT	2 SLIDES	ON TIME

BOARDING

# KURZVORSTELLUNG



## Kurzvorstellung MICHAEL STREB

- **28 Jahre**
- **seit 2006 bei der NETWAYS GmbH**
- **Managing Consultant / Trainer**
- **zuvor 7 Jahre im Bereich IT-Infrastruktur und Datacenter**
  - Betrieb von Netzwerken
  - Aufbau von Rechenzentren
  - Interner IT Betrieb eines Softwareunternehmens



## Kurzvorstellung NETWAYS

- **Firmengründung 1995**
- **GmbH seit 2001**
- **Open Source seit 1997**
- **Nagios / Netsaint seit 1999**
  
- **20 festangestellte Mitarbeiter**
  
- **Spezialisierung in den Bereichen Open Source Systems Management und Open Source Datacenter Infrastructure**



## Unsere Leistungen

### Open Source DataCenter Solutions

#### OS Systems Management

- Monitoring
- Performance Management
- Configuration Management
- Service Management
- Knowledge Management
- Asset Management
- Identity Management
- Backup & Recovery

#### OS Datacenter Infrastructure

- High Availability Lösungen
- Cluster Lösungen
- Loadbalancing
- Virtualisierung
- Speicherlösungen
- Firewalls
- Datenbanken
- Voice over IP

Managed Services

Monitoring HW

Veranstaltungen



BOARDING

WAS IST PUPPET?

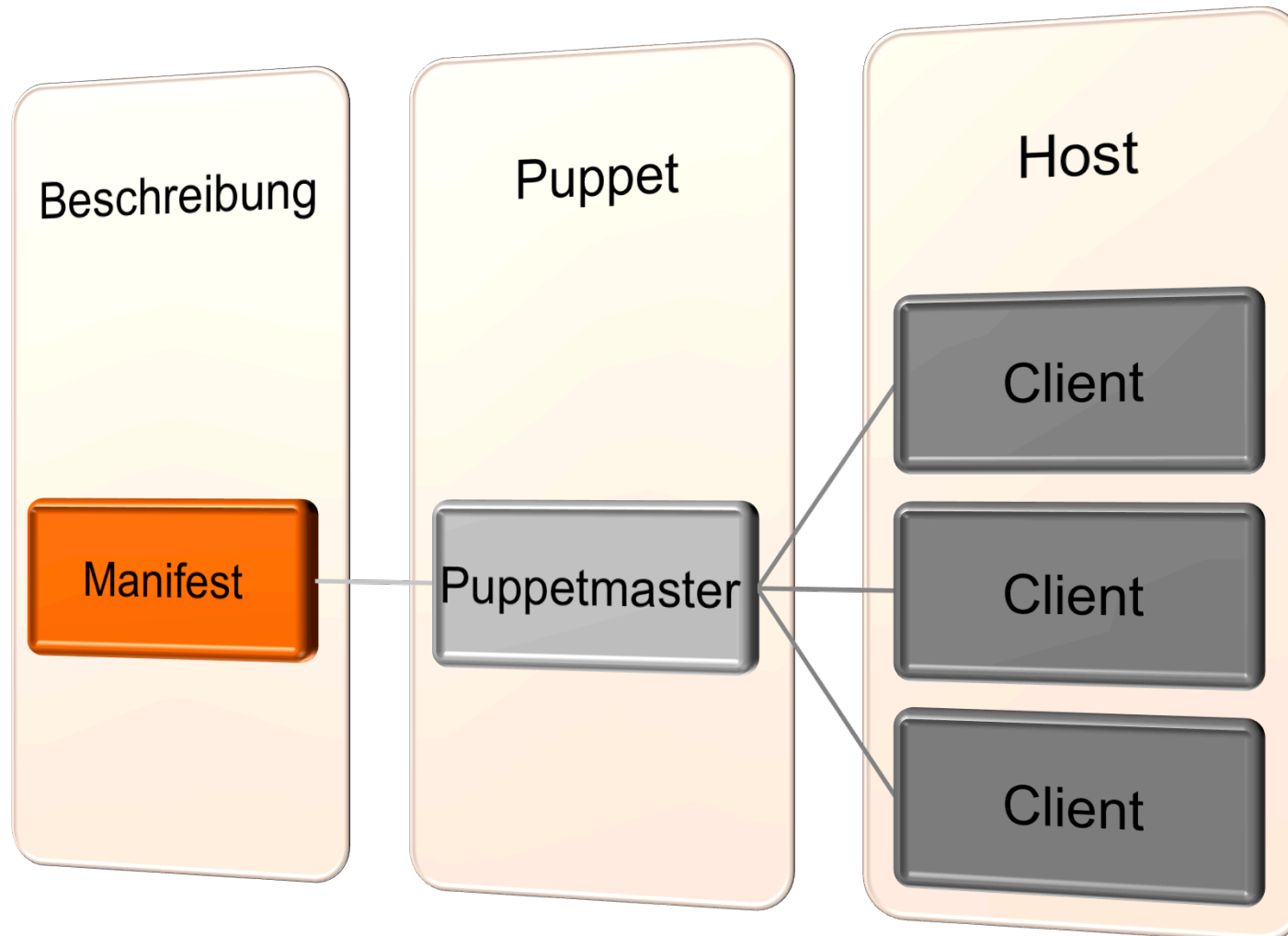


## WAS IST PUPPET?

- **Konfigurationsengine**
  - basiert auf Ruby
  - eigene Sprache für Manifest (ralsh)
  - Manifest ist Plaintext
- **Automatisierung von Tasks**
  - Paketinstallation
  - Anlegen von Benutzer/Gruppen
  - Verteilen von Konfigurationsdateien
- **Factor für die Erkennung**
  - Puppet kennt das System
  - befüllt Variablen über das System



## WIE FUNKTIONIERT PUPPET?



## WIE SIEHT EIN MANIFEST AUS?

- **Hauptkonfigurationsdatei**

```
import "classes/*"
```

```
$server = "puppethost.example.com"
```

```
# global filebucket on server
```

```
filebucket { main:
```

```
    server => $server
```

```
}
```

```
node "host.example.com" {
```

```
    include kunden
```

```
}
```



## WIE SIEHT EIN MANIFEST AUS?

- Erzeugen von Klassen

```
class kunden {  
  group { nagios:  
    ensure => "present", gid => 1000;  
  }  
  schedule { puppet:  
    period => hourly, repeat => 6  
  }  
  file {  
    "/etc/passwd":  
      owner => "root",  
      group => "root",  
      mode => 644  
    }  
}
```



## WIE SIEHT EIN MANIFEST AUS?

- **Zentrale Verteilung einer Datei**

```
file {  
  "ntp.conf":  
    mode => 644,  
    owner => root,  
    group => root,  
    path => "/etc/ntp.conf",  
    source => "puppet:///files/etc/ntp.conf"  
}
```



## WIE SIEHT EIN MANIFEST AUS?

- **Differenziertes Manifest**

```
case $virtual {
  "xenu": {
    exec {"Change clocksource":
      path => "/usr/bin:/usr/sbin:/bin",
      command => "echo jiffies > /sys/devices/system/
        clocksource/clocksource0/current_clocksource",
      unless => "test `cat /sys/devices/system/clocksource/
        clocksource0/current_clocksource` = jiffies";
    }
  }
  "xen0": {
    exec {"Change clocksource":
      path => "/usr/bin:/usr/sbin:/bin",
      command => "echo jiffies > /sys/devices/system/clocksource/
        clocksource0/current_clocksource",
      unless => "test `cat /sys/devices/system/clocksource/
        clocksource0/current_clocksource` = jiffies";
    }
  }
}
```



## WIE KOMMUNIZIERT PUPPET?

- **XMLRPC over HTTP(S)**
  - Verschlüsselt
  - Zertifikatsbasiert
  - Verwaltung durch Puppet

```
# puppetca --sign client1.netways.de
```



## WO HILFT PUPPET?

- **Verwaltung großer Umgebungen**
  - Beschreibung der Konfiguration
  - Regelbasiert
- **Initial höherer Aufwand, aber:**
  - einfaches Massendeployment
  - einfaches Scaleout der Plattform
  - definierter Konfigurationsstand der Clients



## WO FUNKTIONIERT PUPPET?

- **CentOS**
- **Debian 3.1 and later**
- **Fedora Core 2-6**
- **Fedora 7 and later**
- **Gentoo Linux**
- **Mandriva Corporate Server 4**
- **RHEL 3 and later**
- **Oracle Linux**
- **SuSE Linux 8 and later**
- **Ubuntu 7.04 and later**
- **BSD**
- **FreeBSD 4.7 and later**
- **OpenBSD 4.1 and later**
- **Macintosh OS X**
- **Sun Solaris 2.6, 7 and later**
- **!!! Windows support is coming in 2010 !!!**



## MIT WEM KANN PUPPET?

- Apache
- Asterisk
- AutoFS
- Bacula
- BIND
- DNS
- Courier IMAP/  
POP
- denyhosts
- exim
- heartbeat
- iptables
- git
- Linux Vserver
- Lighttpd
- MediaWiki
- munin
- MySQL
- Nagios
- NTP
- OpenVPN
- OpenLDAP
- postfix
- PostfixAdmin
- Shorewall
- snmpd
- SpamAssassin
- SSH Webproxy
- Subversion
- sudo
- Tomcat
- Trac
- Varnish
- Xen 3
- Zope





WAS IST SVN?



## WAS IST SVN?

- **Versionsverwaltung**
- **Element hat eindeutige ID**
  - ID kann dynamisch ersetzt werden
- **mehrere Revisionen von Elementen**
  - einfacher Vergleich von Änderungen
- **branches**
  - verschiedene Teilbereiche
  - Kunden
  - gleiche Basis



## WAS BIETET SVN?

- **Benutzerverwaltung**
  - wer darf was?
- **Changetracking**
  - wer hat welches Manifest geändert?
- **ermöglicht einfachen Rollback**
  - checkout einer anderen Revision
- **Visualisierung (WebSVN)**
- **Netzwerkkommunikation**
  - eigenes Protokoll
  - WebDAV über Apache
- **Hooks**



BOARDING

WARUM DIESE TOOLS?



## WARUM DIESE BEIDEN?

- **Puppet**
  - vergleichsweise einfach!!
  - hoher Anfangserfolg
  - sehr flexibel
  - einfach erweiterbar
- **SVN**
  - HOOKS!!
  - vorhandene Skripte
  - ebenso mit GIT möglich



**BOARDING**

**WELCHE VORTEILE**



## WELCHE VORTEILE?

- **Configurationmanagement**
  - Pakete sind auf allen Systemen gleich
  - Konfigurationen sind durchgängig
  - Zentrale Pflege der Konfiguration
  - Testing möglich
  - schnelle Incidentbehandlung
- **Changetracking**
  - wer hat was geändert?
  - wann wurde die Änderung vorgenommen?
  - Genehmigungsprozess möglich



BOARDING

UMSETZUNGSMÖGLICHKEIT

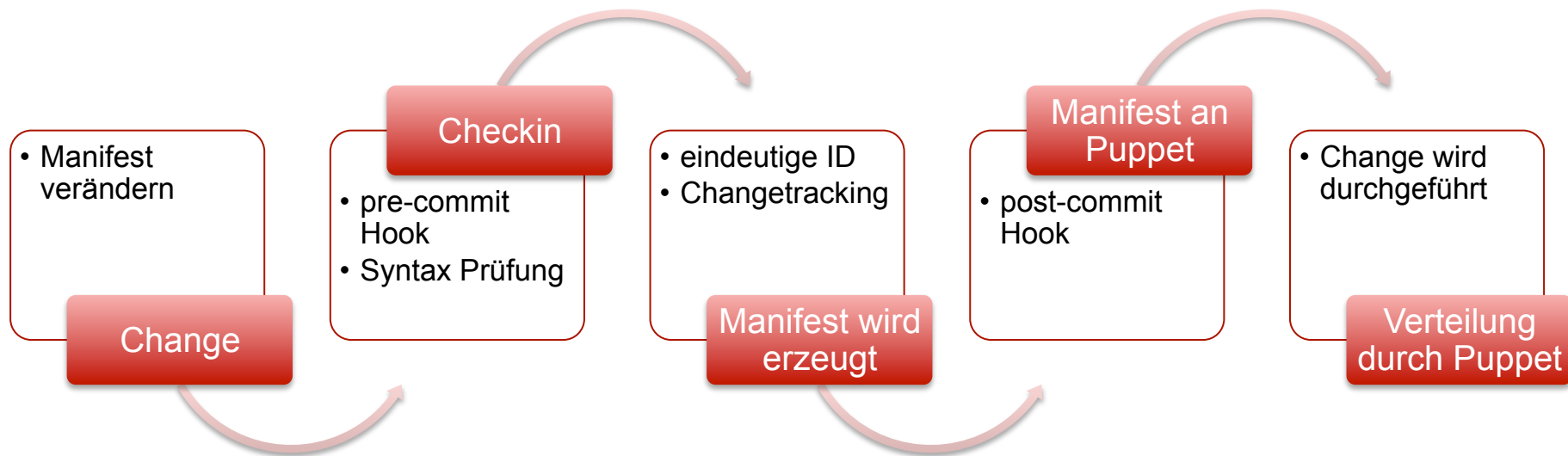


## UMSETZUNGSMÖGLICHKEITEN

- **Puppet Manifeste in SVN**
- **Syntaxcheck vor commit (pre-commit)**
  - vollständiger compile
  - nur syntaxcheck
- **Export bei valider Syntax (post-commit)**
- **Verteilung über Puppetmaster**
  
- **bereits fertig:**
  - [http://projects.puppetlabs.com/projects/puppet/wiki/Subversion\\_Commit\\_Hooks\\_Patterns](http://projects.puppetlabs.com/projects/puppet/wiki/Subversion_Commit_Hooks_Patterns)
  - [http://projects.puppetlabs.com/projects/puppet/wiki/Puppet\\_Version\\_Control](http://projects.puppetlabs.com/projects/puppet/wiki/Puppet_Version_Control)



# UMSETZUNGSMÖGLICHKEIT



BOARDING

FRAGEN UND ANTWORTEN!?



## Fragen und Antworten

**Jetzt und Hier**

---

**NETWAYS GmbH  
Deutschherrnstrasse 15-19  
90429 Nürnberg**

---

 **michael.streb@netways.de**

 **Twitter: twitter.com/netways**

---

**www.netways.de**

 **blog.netways.de**

---

**www.google.de/search?q=netways**

**www.google.de/search?q=michael+streb**

