



Konfigurations- und Changemanagement mit Bcfg2

Marko Jung
mjung@markojung.net

01.03.2007

Frühjahrsfachgespräch der GUUG
Freie Universität Berlin



Sagen & Mythen

- „Jede meiner Maschinen ist einzigartig.“
- „Ich brauche nie mehr ein solches System wie dieses!“
- „Die Konfiguration ändert sich nicht oft.“
- „Dieser Server ist nur eine temporäre Zwischenlösung und wird bald ersetzt.“
- „Systemkonfigurationstools werten meine Fertigkeiten und Fähigkeiten ab!“

Wirklich?



Motivation

- Systeme und Netzwerke werden stetig
 - größer
 - komplexer
 - heterogener
- Steigende Anforderungen an Systeme und deren Verfügbarkeit
 - Updates / Patches
 - Konfigurationsanpassungen
- Konstante bzw. stagnierende administrative Manpower



Konfigurationsmanagement

- Vereinfachung des Systemkonfigurationsprozesses
 - Einheitlichkeit ausnutzen
 - Automatisierung wiederholender Tasks
 - Automatische Konfigurationsgenerierung (aus deklarativer Spezifikation)
 - Erfassung des aktuellen Konfigurationsstatus des gesamten Netzwerks
 - Identifikation von Abweichungen von der zentralen Spezifikation
- ➔ Proaktive statt reaktiver Administration



Weit verbreitete Tools

- cfengine
- SystemImager
- KickStart
- JumpStart
- AutoYAST
- Redhat Network
- GNU dd
- \$YOUR_FAVORITE_TOOL_HERE



Nachteile vieler dieser Tools

- Nur Systeminstallation,
keine langfristige Konfiguration
- Imaging-Tools unterstützen meistens
keine Konfigurationsabweichungen
- Architektur- / Distributionsspezifisch
- Imperativ
- Idempotente Skripte
- Keine höherstufige Logik
- Freaky



Bcfg2

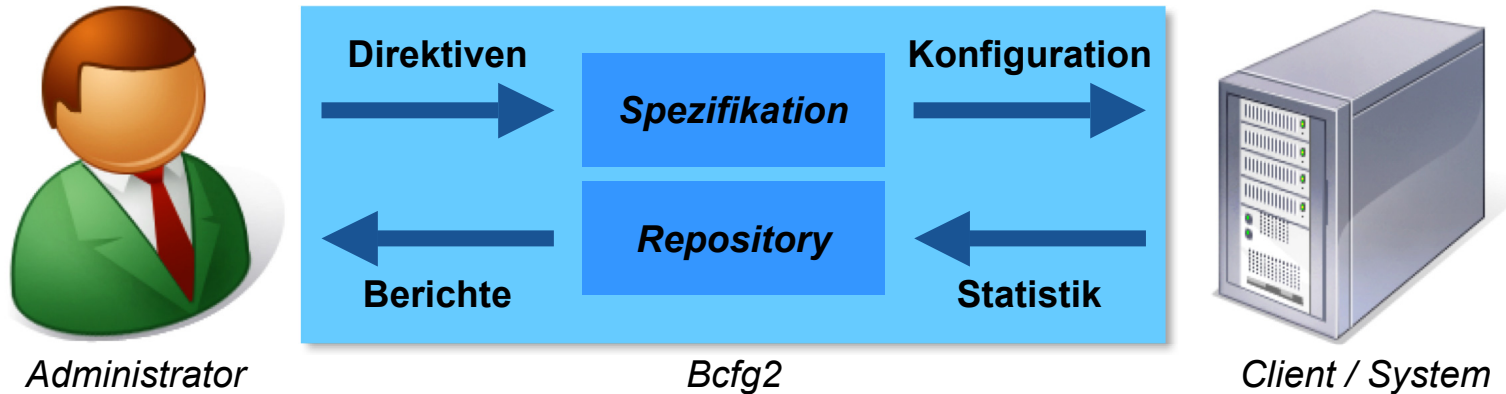
- Client/Server-Applikation
- Open Source unter BSD-Lizenz
- Python-Implementierung
- Abstraktion der verwendeten Systemarchitekturen und Distributionen
- Unterstützte Plattformen
 - LINUX (Debian, Ubuntu, Fedora/Redhat, SUSE, Gentoo,...)
 - Solaris
 - FreeBSD
 - MacOS X (in Entwicklung)



Designziele

- Generierung der gesamte Konfiguration basierend auf zentraler Spezifikation
 - Deklarative Konfigurationsspezifikation
 - Sukzessive Entwicklung der Konfigurationsspezifikation
 - Keine Diskriminierung lokaler Änderungen
- Konfigurationsstatus aller Clients
- Einfache Erweiterbarkeit des Systems
 - Plugin-Architektur (*Generatoren*)
- Einfachheit vs. Flexibilität

Architektur



- Interaktion mit Bcfg2, nicht den Clients
 - Administrator erstellt die Spezifikation, welche in clientspezifische Konfiguration transformiert wird
 - Client sendet Statistiken an Server, welcher diese in Berichte für den Administrator aufbereitet

Konfigurationsprozess

Kommunikation immer
seitens des Client
mittels XML-RPC

1. Anforderung von *Proben*
2. Senden der Ergebnisse
3. Abrufen der Konfiguration
4. Konfigurationsanpassung
 - a. Lokale Inventur
 - b. Zwei-Wege-Verifikation
 - c. Korrektur der Fehlkonfigurationen
5. Senden der Statistiken



Bcfg2-Client

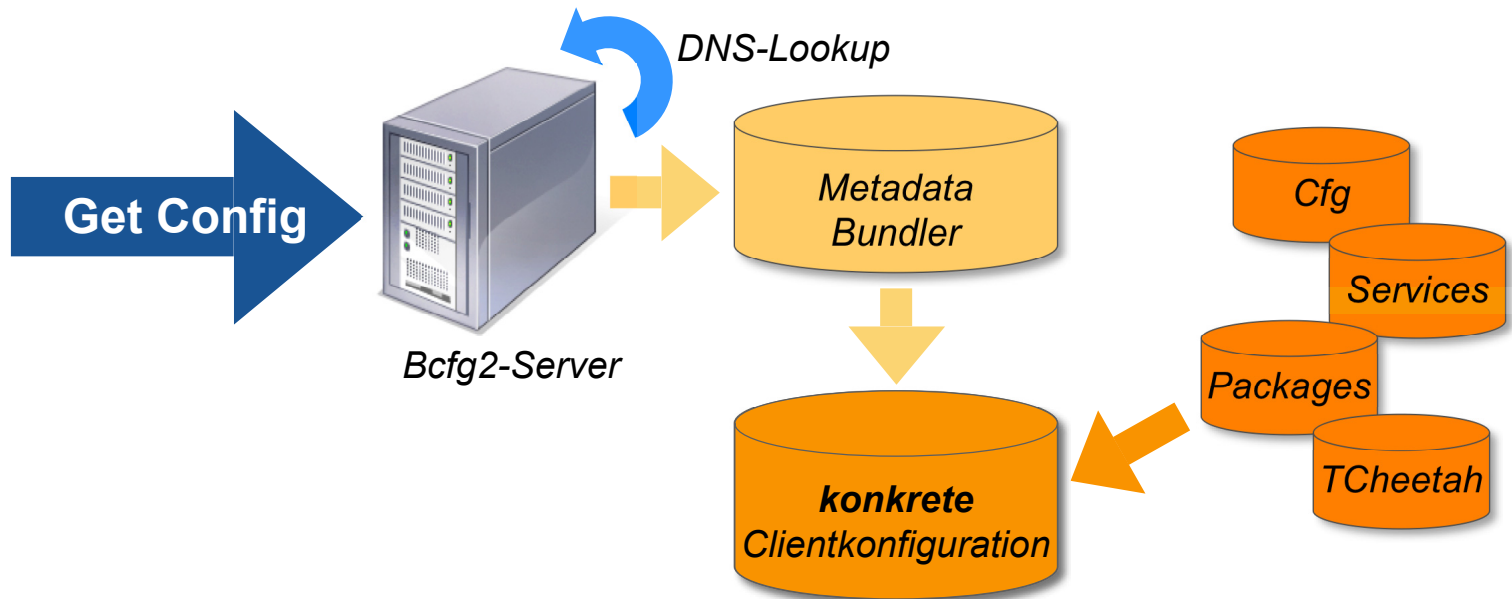


Get Probes
Send Results
Get Config
Send Statistics



Bcfg2-Server

Konfigurationsgenerierung



- Identifikation des Clients (DNS-Name)
- Sammeln der abstrakten Konfiguration
 - Metadata
 - Bundles
- Generierung der konkreten Konfiguration
 - Generatoren/Plugins



Metadata

- Grundlage der Spezifikation
- Zuweisung je eines *Profils* (Basisgruppe) an einen Client

```
<Clients>
  <Client profile="desktop" name="foo.example.com" />
  <Client profile="desktop" name="bar.example.com" />
  <Client profile="desktop-testing"
    name="test.example.com" pingable="N"/>
  <Client profile="webserver" name="www.example.com" />
  <!-- .. --!>
</Clients>
```



Metadata (2)

- Zusammenfassung einzelner Elemente und *Bundles* zu Gruppen

```
<Groups>
  <Group name='desktop' profile='true'>
    <Bundle name='motd' />
    <Bundle name='networking' />
    <Group name='gnome-desktop' />
    <Group name='ubuntu-stable' />
  </Group>

  <Group name='desktop-testing' profile='true'>
    <Group name='gnome-desktop-testing' />
    <!-- ... --!>
    <Group name='ubuntu-testing' />
  </Group>

  <Group name='ubuntu-stable' toolset='debian' />

  <!-- ... --!>
```



Bundler

- Zusammenfassung von Einträgen zu logischen Einheiten (*Bundles*)

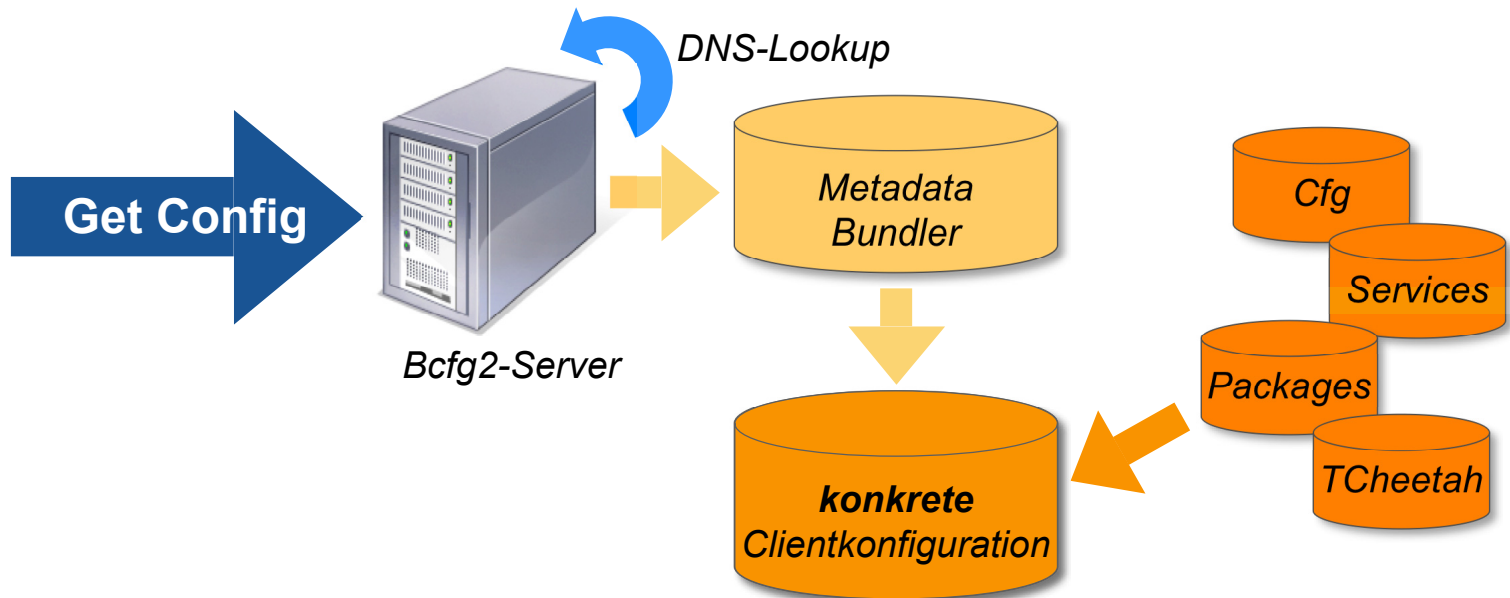
```
<Bundle name='motd' version='2.0'>  
  <Package name='login' />  
  <ConfigFile name='/etc/motd' />  
</Bundle>
```

```
<Bundle name='gdm' version='2.0'>  
  <Package name='gdm' />  
  <Package name='gdm-theme-example.com' />  
  <ConfigFile name='/etc/gdm/gdm.conf' />  
  <Service name='gdm' />  
</Bundle>
```



```
<Bundle name='ssh' version='2.0' revision='$Revision: 2668
  <ConfigFile name='/etc/ssh/ssh_host_dsa_key' />
  <ConfigFile name='/etc/ssh/ssh_host_dsa_key.pub' />
  <ConfigFile name='/etc/ssh/ssh_host_key' />
  <ConfigFile name='/etc/ssh/ssh_host_key.pub' />
  <ConfigFile name='/etc/ssh/sshd_config' />
  <ConfigFile name='/etc/ssh/ssh_config' />
  <Group name='fedora'>
    <Package name='openssh' />
    <Package name='openssh-askpass' />
    <Service name='sshd' />
    <Group name='fc5' >
      <Group name='desktop' negate='true'>
        <Package name='openssh-clients' />
      </Group>
      <Package name='openssh-server' />
    </Group>
  </Group>
  <Group name='debian'>
    <Package name='ssh' />
    <Service name='ssh' />
  </Group>
</Bundle>
```

So far...



- Identifikation des Clients (DNS-Name)
- Sammeln der abstrakten Konfiguration
 - Metadata
 - Bundles
- Generierung der konkreten Konfiguration
 - Generatoren/Plugins



Generatoren

- Plugins zur Generierung der konkreten Konfigurationsinformation
 - Python-Modul +
 - Datenquelle
- Auswahl eines individuellen Generators je Konfigurationseintrag
 - Beliebige Datenquellen
 - Vielzahl von Spezifikations Sprachen möglich
 - Komplexe Logik
 - Mehrere Frontends



Pkgmgr - Paketverwaltung

- Plattformunabhängige Softwareverteilung
- Auflösen der abstrakten *Package*-Einträge
 - Version
 - Quelle
 - Up- / Downgrade notwendig?
 - Verifikation des Pakets gewünscht?

```
<Bundle name='webserver' version='2.0'>  
  <Package name='apache2' />  
  <Package name='libapache2-php5' />  
  <Service name='libapache2-modxsel' />  
  <Group name='ultraspecialprojectserver' >  
    <Package name='apache2' version='2.2.3-3.3' />  
  </Group>  
</Bundle>
```



Pkgmgr - Paketverwaltung (2)

- XML-Daten zu verfügbaren Paketen
 - Regelmäßige Scans der Paketrepositories
 - Prioritäten für Repositories
 - Zuweisung der Pakete zu Gruppen/Clients

```
<PackageList priority="50" type="deb">
  <Group name="ubuntu-stable">
    <Package name="libc6-i686" version="2.3.6-0ubuntu20.4"/>
    <Package name="base-files" version="3.1.9ubuntu7.1"/>
    <Package name="dpkg" version="1.13.11ubuntu7"/>
    <Package name="cpio" version="2.6-10ubuntu0.2"/>
    <Package name="ubuntu-minimal" version="0.120"/>
    <Package name="locales" version="2.3.18.1"/>
    <Package name="libc6" version="2.3.6-0ubuntu20.4"/>
  </Group>
</PackageList>
```



Svcmgr - Dienstverwaltung

- Verwaltung von Systemdiensten
 - Aktivierung und Deaktivierung
 - Monitoring

```
<Services priority='0'>                               Svcmgr/services.xml  
  <Service name='ntpd' status='on' />  
</Services>
```

```
<Base>                                                Base/base.xml  
  <Group name='BeLeniX'>  
    <Package name='ntp' />  
    <Service name='ntpd' />  
  </Group>  
</Base>
```



Cfg - Konfigurationsdateien

- Dateisystem-Repository
 - je Datei ein Verzeichnis `./etc/gdm/gdm.conf/`
 - `gdm.conf` Standard
 - `gdm.conf.H_foo.ex.com` Host
 - `gdm.conf.G50_foobar` Gruppe
 - `gdm.conf.G90_bar`
 - `:info` Metadaten
- Möglichkeit zur Mischung von Konfigurationen (diff-basierend)



TCheetah

- Integration der Cheetah-Template-Sprache in Konfigurationsdaten
- Verzeichnisstruktur wie bei *Cfg*-Generator
 - `info` Metadaten
 - `template` Cheetah-Template mit Bcfg2-Erweiterungen



Tcheetah (2 - Beispiel)

```
Hostname is $self.metadata.hostname
```

```
Groups:
```

```
#for $group in $self.metadata.groups:
```

```
  * $group
```

```
#end for
```

```
Categories:
```

```
#for $category in $self.metadata.categories:
```

```
  * $category -- $self.metadata.categories[$category]
```

```
#end for
```

```
<ConfigFile name="/foo" owner="root" perms="0624"  
  group="root">
```

```
Hostname is topaz.mcs.anl.gov
```

```
Groups:
```

```
  * desktop
```

```
  * ypbound
```

```
  * workstation
```

```
  * debian-sarge
```

```
  * debian
```

```
</ConfigFile>
```

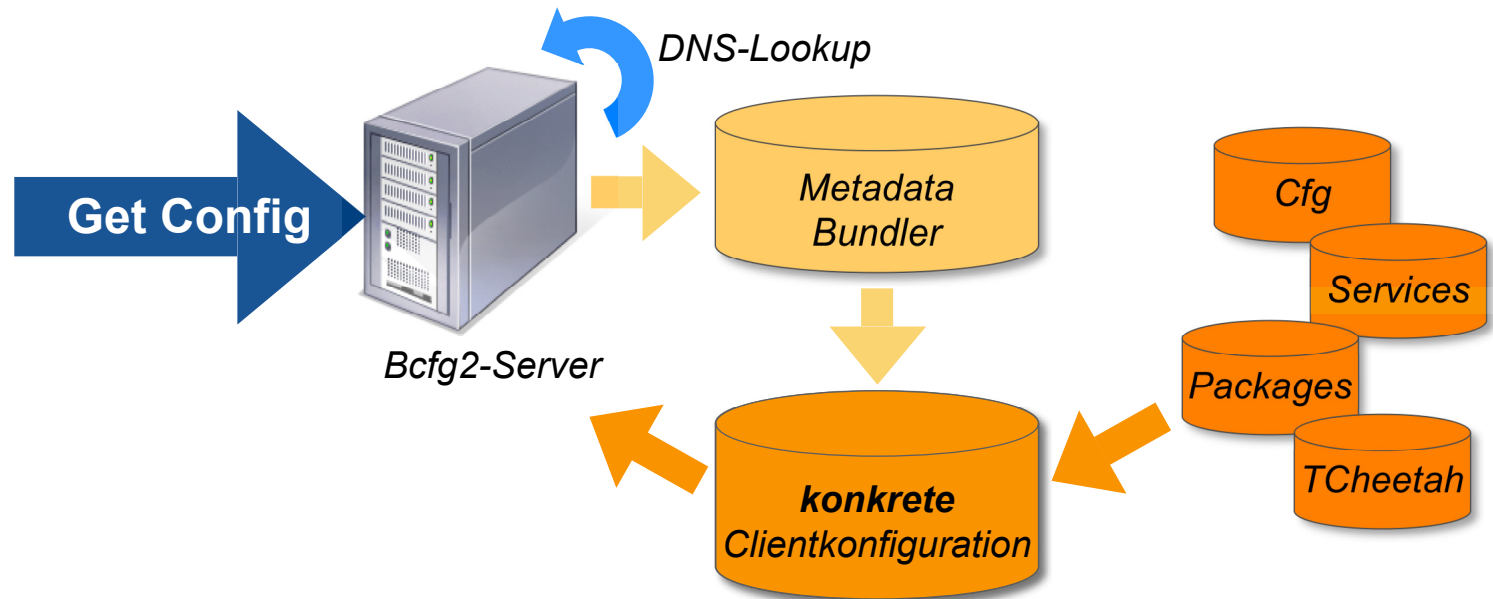


Tcheetah (3)

- Datenbankbindung mit Tcheetah

```
#from Bcfg2.Server.dbconnection import DBPgConnection
#silent result = DBPgConnection().execute(" \
    SELECT ip, netmask, broadcast, gateway \
    FROM hosts WHERE hostname = '%s'" % \
    $self.metadata.hostname)
auto eth0
iface eth0 inet static
    address $result[0]
    netmask $result[1]
    broadcast $result[2]
    gateway $result[3]
```


Done!



- Identifikation des Clients (DNS-Name)
- Sammeln der abstrakten Konfiguration
 - Metadata
 - Bundles
- Generierung der konkreten Konfiguration
 - Generatoren/Plugins

Reports

Info for: topaz.mcs.anl.gov

http://127.0.0.1:8000/clients/topaz.mcs.anl.gov

Google

Bcfg2 Reporting System

...Change Is Coming...

Home
Clients
Displays
System Summary
Timing

Select time: 2006-06-12 10:14:16
✓ 2006-07-17 14:05:53

Time Ran: 2006-07-17 14:05:53

Node: **topaz.mcs.anl.gov** Revision: -1

This node did not run within the last 24 hours-- it may be out of date.

1344 items did not verify and are considered Dirty.

- ConfigFile: /var/cache/debconf/templates.dat
- ConfigFile: /etc/csh.cshrc
- Package: libxrandr2
- Package: libxpm4
- Package: cpio
- Package: xlibs
- Package: xfree86-common
- Package: libxext6
- Package: libxp6
- ConfigFile: /var/cache/debconf/config.dat
- Package: python2.3
- ConfigFile: /etc/motd
- Package: lynx
- ConfigFile: /etc/hosts.allow
- Package: sudo
- ConfigFile: /etc/ssh/sshd_config
- Package: bsdutils
- Package: libxft6



Zusammenfassung

- Architektur- und distributionsunabhängige Konfigurationsverwaltung
- Softwareverteilung und -management
- Sukzessive Migration zu Bcfg2 möglich
- Ständige Verfügbarkeit der Systemzustände
- Modulare Architektur
- Vitale Entwicklergemeinschaft



Vielen Dank!

Fragen?



Referenzen

- Bcfg2 - Projekt,
<http://www.bcfg2.org>
- Argonne National Laboratory,
<http://www.anl.gov/>
- Cheetah - Python powered Template Engine,
<http://www.cheetahtemplate.org/>

Dank an

- S. Pfeifer
- R. Gogolok
- W. Drotschmann
- N. Desai



Picture Credits

Images are taken from **flickr**^{GAMMA} using **© creative commons** licenses

- Daedalicious
- Edwardaggie98
- Cindy47452
- -Fabio-
- Bluman
- Winky!
- Kyle Jones
- Zen
- Designwallah
- El Ramon
- Dvs
- Ashwion
- Ben Cumming
- Henrik
- Stgermh
- Hongkongwillie
- Cgines
- Uirariamumu
- Claudecf
- Brothergrimm
- Wetwater
- LordKhan
- PaveM
- Dulcie
- Sodeste
- TP
- Hinke
- Goatopolis
- MrJoro
- Schnurrbart
- The Jamoker
- R-round the world
- FreaksAnon