



# Einführung in die Erstellung von Software-Paketen mit dem RPM Package Manager

Lenz Grimmer <[lenz@grimmer.com](mailto:lenz@grimmer.com)>

2008-03-13, TU Harburg

# Inhalt

- Hintergrund / Historie
- Funktionen / Arbeitsweise
- Installation von RPM-Paketen
- Erstellen von RPM-Paketen
- Weitere Informationen / Links



# Hintergrund/Historie

- Paketmanagementsystem und Dateiformat
- 1997 von Eric Troan und Marc Ewing (Red Hat) entwickelt
- Basis vieler Linux-Distributionen (RHEL, Fedora, SLES, openSUSE, CentOS, Mandriva)
- AIX, OpenSolaris, MacOS X, OpenPKG

# Funktionen / Arbeitsweise

- Erstellung/Installation/Deinstallation von Softwarepaketen
- Konsistenzprüfung (MD5-Checksummen / GPG-Signaturen)
- Überprüfung von Abhängigkeiten / Kollisionen bei Installation
- Meta-Daten (Name, Version, Lizenz, Konfigurationsdateien, Dokumentation)
- Zentrale Datenbank

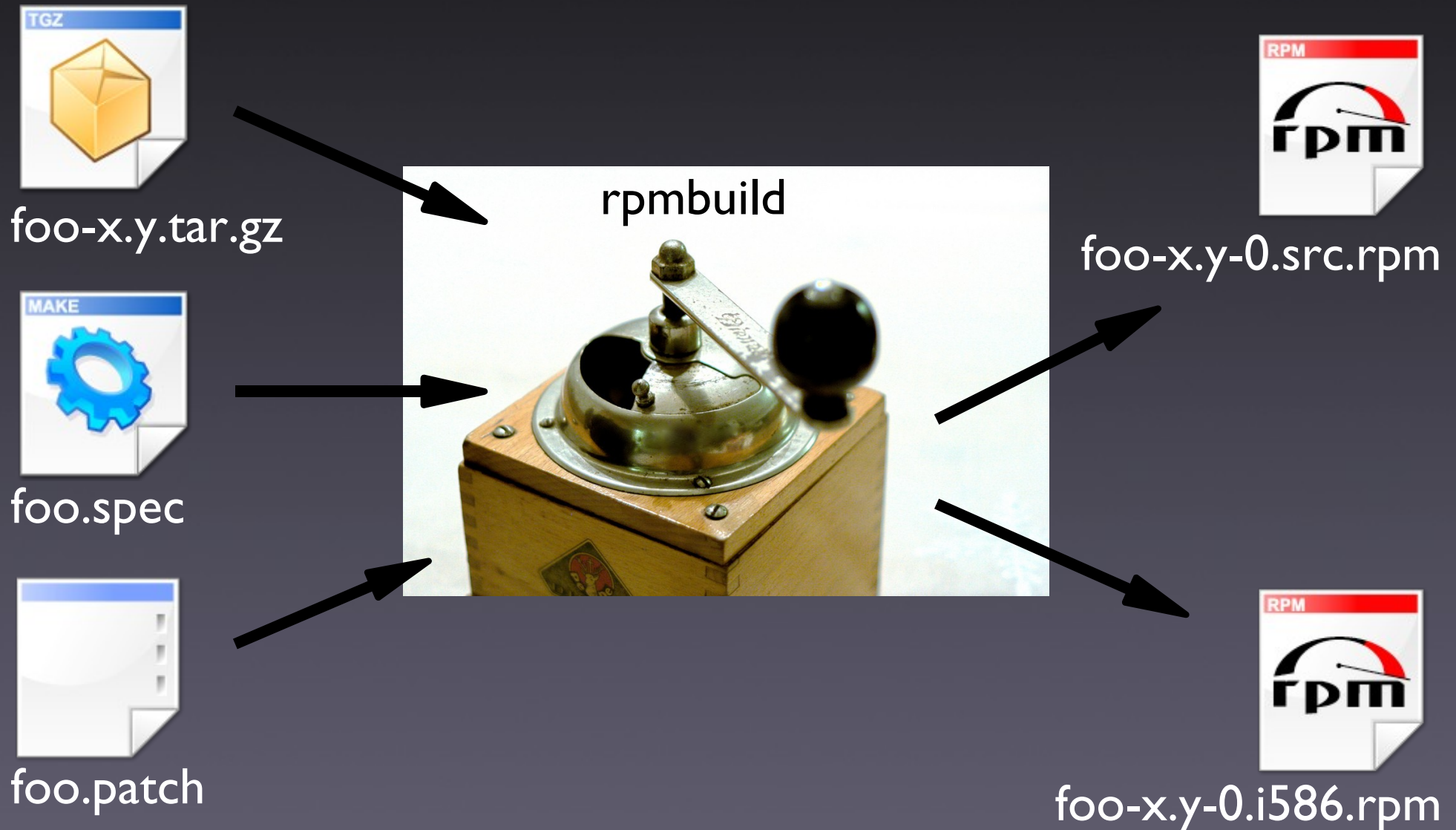
# Vorteile

- Einfache, rückstandsfreie Deinstallation
- Plattform-/Architekturunabhängig
- Automatisierbar (rpm-lib)
- Original-Quellarchive + Patches im Quellpaket enthalten
- Sicherung von Konsistenz & Authentizität
- Reproduzierbare Builds

# Installation von RPM-Paketen

- Paketformat: gzip-komprimierte cpio-Datei mit Header
- Plattformspezifische Pakete (z.B. i586, sparc, ppc)
- Überprüfung der Abhängigkeiten
- Pre-/Postinstallations-Scripts
- Trigger
- RPM-Datenbank `/var/lib/rpm`

# Erstellen von RPM-Paketeten



# Sources / Patches

- „Pristine Sources“ - Original Quellarchiv soll möglichst erhalten bleiben
- Notwendige Modifikationen über separate patch-Dateien (optional)
- Multiple Sourcen/Patches möglich



# .spec-Datei

- Header (Metainformationen)
- Bauanleitung
- Scripts/Trigger
- Dateilisten
- Abhängigkeiten
- Macros: `/usr/lib/rpm`



# Build-Phasen

- `%prep` – Vorbereitung der Quellen
  - Extrahieren
  - Patchen
- `%build` – Übersetzung der Quellen
  - `configure ; make`
- `%install` – Installation
  - `make install`
- `%files` - Dateilisten

# Abhängigkeiten

- Requires:
- PreReq:
- Provides:
- Conflicts:
- Obsoletes:
- BuildRequires:
- BuildPreReq:



# Scripts / Trigger



- %pre
- %post
- %preun
- %postun
- %triggerin
- %triggerun
- %triggerpostun

# minimale .spec-Datei

```
#  
# A very minimal specfile  
#  
Name: demo  
Version: 1.2  
Release: 0  
Group: Applications/Utilities  
License: GPL  
Summary: A demo package  
  
%description  
The most simple package possible
```

# Live-Demo



# Hinweise

- Immer mit BuildRoot arbeiten
- Pakete bauen ohne root-Rechte
- chroot()-Umgebung zur Konsistenzsicherung / Reproduzierbarkeit
- lokal:build (openSUSE) oder mock (Fedora)
- openSUSE Build Service:  
<http://build.opensuse.org/>
- Fedora Koji: <http://koji.fedoraproject.org/koji/>

# Weitere Informationen / Links

- Alternative Paketmanagement-Systeme
  - DEB/dpkg (z.B. Debian, Ubuntu, Fink):  
[http://www.debian.org/doc/FAQ/ch-pkg\\_basics](http://www.debian.org/doc/FAQ/ch-pkg_basics)
  - Conary: <http://wiki.rpath.com/wiki/Conary>
- Auf RPM aufsetzend
  - Yum: <http://linux.duke.edu/projects/yum/>
  - Smart RPM: <http://smartrpm.org>
  - apt-rpm: <http://apt-rpm.org/>



# Links / Dokumentation

- RPM: <http://rpm.org/> / <http://rpm5.org/>
- RPM Guide:  
<http://docs.fedoraproject.org/drafts/rpm-guide-en/>
- Maximum RPM:  
<http://rpm5.org/docs/max-rpm.html>
- RPM HOWTO:  
<http://www.tldp.org/HOWTO/RPM-HOWTO/>
- Creating Quality RPMs (pdf):  
<http://www.gurulabs.com/goodies/guru+guides.php>

Fragen / Diskussion



案内所

Question & Answer

问讯处 안내소

Vielen Dank!

[lenz@grimmer.com](mailto:lenz@grimmer.com)