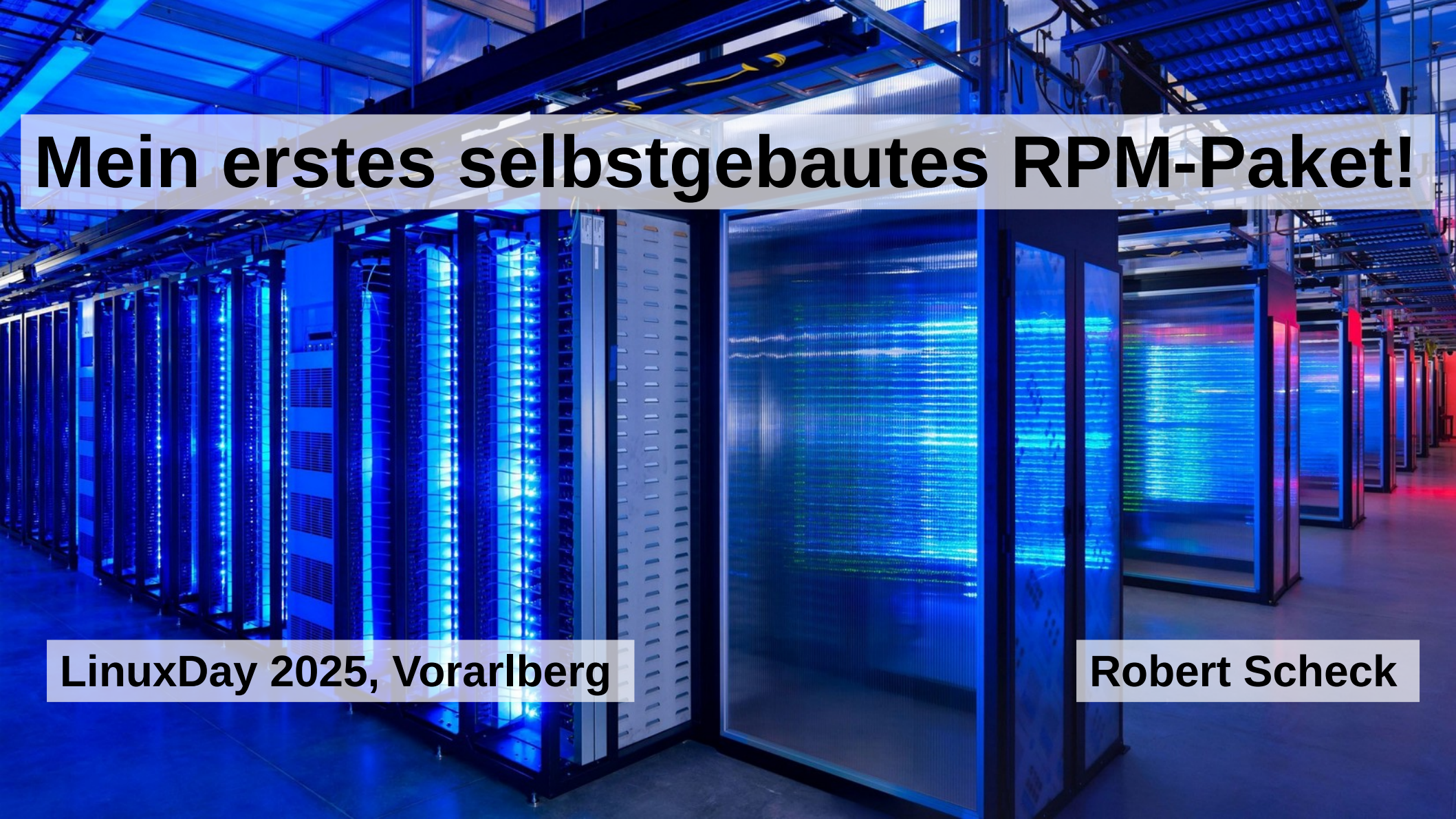


Mein erstes selbstgebautes RPM-Paket!

A photograph of a server room with blue ambient lighting. Several server racks are visible, some with glass doors showing internal components. The room has a high ceiling with exposed pipes and cables.

LinuxDay 2025, Vorarlberg

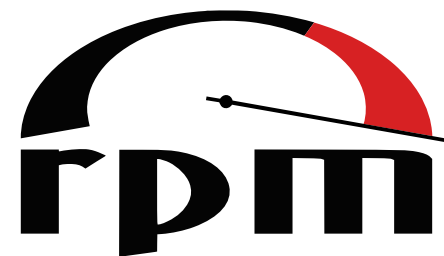
Robert Scheck

Robert Scheck

- Mitwirkender beim Fedora Project (seit 2005)
 - Package Maintainer (ca. 250 Pakete) mit Fokus auf EPEL
 - Provenpackager und Packager Sponsor
- Mitwirkender bei diversen anderen Open-Source-Projekten
u.a. aggregate6, Baresip, bgpdump, bgpq4, Coturn, CUPS, iputils, jwhois, Linux-Kernel, MIMEDefang, Nextcloud, OpenSSL, ownCloud, Partclone, PlayOnLinux, phpMyAdmin, popt, rpki-client, systemd
- Mail: robert@fedoraproject.org
- Web: <https://fedoraproject.org/wiki/RobertScheck>

Grundwissen

- RPM = RPM (früher: Red Hat) Package Manager
- Spec-Datei (* .spec) ist für RPM wie ein Makefile für make
- Format: <Name>-<Version>-<Release>.<Arch>.rpm
- Einige Beispiele für Binärpakete:
 - bash-5.3.0-2.fc43.x86_64.rpm
 - fedora-release-43-0.22.noarch.rpm
 - systemd-258-1.fc43.aarch64.rpm
- Source-RPM: bash-5.3.0-2.fc43.src.rpm



System-Vorbereitungen

RPM benötigt einige bestimmte Verzeichnisse:

```
drwxr-xr-x. 2 root root 4096 Sep 22 20:36 BUILD
drwxr-xr-x. 2 root root 4096 Sep 22 20:36 RPMS
drwxr-xr-x. 2 root root 4096 Sep 22 20:36 SOURCES
drwxr-xr-x. 2 root root 4096 Sep 22 20:36 SPECS
drwxr-xr-x. 2 root root 4096 Sep 22 20:36 SRPMS
```

Werkzeug installieren und ausführen:

- `dnf install rpmdevtools`
- `rpmdev-setuptree`

RPM-Build-Verzeichnisse

Von rpmdev-setuptree erstellte Verzeichnisse in `~/rpmbuild/`:

<code>BUILD /.../</code>	Quelltextdateien werden hierin kompiliert
<code>BUILD/.../BUILDROOT/</code>	Installationsverzeichnis beim Paketbau
<code>RPMS/.../</code>	Enthält alle gebauten binären RPM-Pakete
<code>SOURCES/</code>	Quelldateien (Archivdateien, Patches, ...)
<code>SPECS/</code>	Enthält „make-Dateien für RPM“
<code>SRPMS/</code>	Beinhaltet gebaute Source-RPM-Pakete

RPMS-Verzeichnis enthält Unterverzeichnisse je CPU-Architektur, wie z.B. `i686`, `x86_64`, `aarch64`, `noarch`

Wichtig: Verzeichnisse haben nichts mit Cross-Compiling zu tun!

Sicherheitshinweise

Pakete niemals als root-Benutzer bauen!

- Wenn ein Makefile oder die Software sich nicht wie erwartet verhält, kann das gesamte System beschädigt werden!
- Sandbox-System „Mock“ in Erwägung ziehen:
 - „Einfache“ Verwaltung von Chroot-Build-Umgebungen für Paketbau
 - Repository-Konfigurationen für verschiedene Linux-Distributionen
 - <https://rpm-software-management.github.io/mock/>
 - In Kürze: `dnf install mock, usermod -aG mock <Benutzer>`

Tags in Spec-Dateien

- `Name`: Upstream-Name der paketierte Software
- `Version`: Upstream-Versionsnummer der Software
- `Release`: Build-Nummer des Paket-Maintainers
- `Summary`: Kurze Beschreibung, max. 80 Zeichen
- `URL`: Upstream-Webseite der paketierte Software
- `Source`: URI zur Archivdatei mit dem Quelltext bzw. Dateiname

Tags in Spec-Dateien

- `License`: Lizenz(en) der Software, z.B. SPDX-Ausdruck
 - Beispiel: `(GPL-2.0-or-later OR LGPL-3.0-or-later) AND GPL-3.0-or-later`
 - <https://spdx.org/licenses/>
 - <https://docs.fedoraproject.org/en-US/legal/allowed-licenses/>
- `BuildRequires`: Benötigte andere Pakete zum erfolgreichen Bauen des eigenen Pakets, typischerweise `-devel`-Pakete
- `BuildArch`: Build-Einschränkungen (z.B. `noarch` für CPU-unabhängige RPM-Pakete oder `Specials` für Buildsysteme)

„Knigge für Spec-Dateien“

- Englische Sprache, Zeichensatz: ASCII oder UTF-8
- Spec-Datei sollte normal und leicht lesbar sein
- Nicht offensichtliche Dinge kurz kommentieren
- Jeder Paket-Maintainer sollte die Spec-Datei lesen können ohne die Software kennen zu müssen
- Nicht zu viele Platzhalter in %files nutzen: Verwirrt leicht und könnte auch zu viele Dateien erfassen (für Unterpakete, neue Dateien, geänderte SOVERSION etc.)

Hinweise und Tipps

Welche Lizenz(en) hat die Software?

- COPYING-, LICENSE- oder README-Datei, Quelltext-Kopfzeilen, gegebenenfalls Entwickler/Upstream fragen

Wie findet man die Build-Abhängigkeiten heraus?

- Einfach ausprobieren und Fehler analysieren
- `dnf provides '*/openssl.h'`
- `dnf group install development-tools`
- `dnf group list | grep development`

Und los gehts...

Ein neues leeres Spec-Skelett erhält man mit:

- `rpmdev-newspec <Name>`

RPM-Paketbau wird normalerweise gestartet mit:

- `rpmbuild -ba <Name>.spec`

Empfehlenswerter Weg mit Mock nach den ersten Erfahrungen:

- `rpmbuild -bs <Name>.spec`
- `mock -r fedora-43-x86_64 rebuild \`
`~/rpmbuild/SRPMS/<Name>-<Version>-<Release>.src.rpm`

Paket nach Fedora bringen

Fedora-Webseite besuchen und Account erstellen:

- <https://accounts.fedoraproject.org/>

Anleitungen, Regeln und Abläufen folgen:

- https://docs.fedoraproject.org/en-US/package-maintainers/Joining_the_Package_Maintainers/

Wichtig: RPM-Paket aktiv pflegen und aktualisieren!



Links

- <https://fedoraproject.org/wiki/RobertScheck>
- <https://ftp.osuosl.org/pub/rpm/max-rpm/>
- <https://rpm-packaging-guide.github.io/>
- <https://rpm-software-management.github.io/mock/>
- <https://docs.fedoraproject.org/en-US/packaging-guidelines/>
- <https://docs.fedoraproject.org/en-US/legal/allowed-licenses/>
- <https://accounts.fedoraproject.org/>
- https://docs.fedoraproject.org/en-US/package-maintainers/Joining_the_Package_Maintainers/

Fragen?



A photograph of a server room with rows of server racks. The room is dimly lit with a strong blue glow emanating from the server units. The racks are filled with various electronic components, and some have glass doors that show internal hardware. The perspective is from a low angle, looking down a long aisle of the server racks. A white rectangular box with the text 'Vielen Dank!' is overlaid on the right side of the image.

Vielen Dank!