

Professionelle Software-Entwicklung mit Linux

Ein Erfahrungsbericht zum Einsatz von Linux in der Produktentwicklung

Inhaltsverzeichnis

1. Vorstellung
2. Ausgangssituation
3. Entwicklungsserver
4. Beispiele
5. Fazit

Gliederung

1. Vorstellung
 - 1.1. Vortragende
 - 1.2. Software

Vortragende

Markus Eckert

- Fachinformatiker Anwendungsentwicklung
- Software-Entwickler bei blackned GmbH seit Oktober 2011
- Teamleiter Core-Entwicklung
- Neben Entwicklung für Linux-Konfiguration zuständig
- Verwendete Linux-Distribution: Arch Linux

Vortragende

Markus Flingelli

- Diplom-Informatiker
- Leiter Software-Entwicklung bei blackned GmbH seit Oktober 2011
- Verwendete Linux-Distribution: Linux Mint
- Administration Entwicklungsserver

Software

Mobile Unified Platform (MUP)

- Hochmobiles System zur Kommunikation auf WLAN-Basis
- Integrierte Dienste (Auswahl):
 - Authentifizierung über RADIUS
 - Statisches und dynamisches Routing (OSPF)
 - Verzeichnisdienst
 - Public-Key-Infrastruktur
 - Telefonie (SIP, SCCP)

MUP – Rucksackvariante

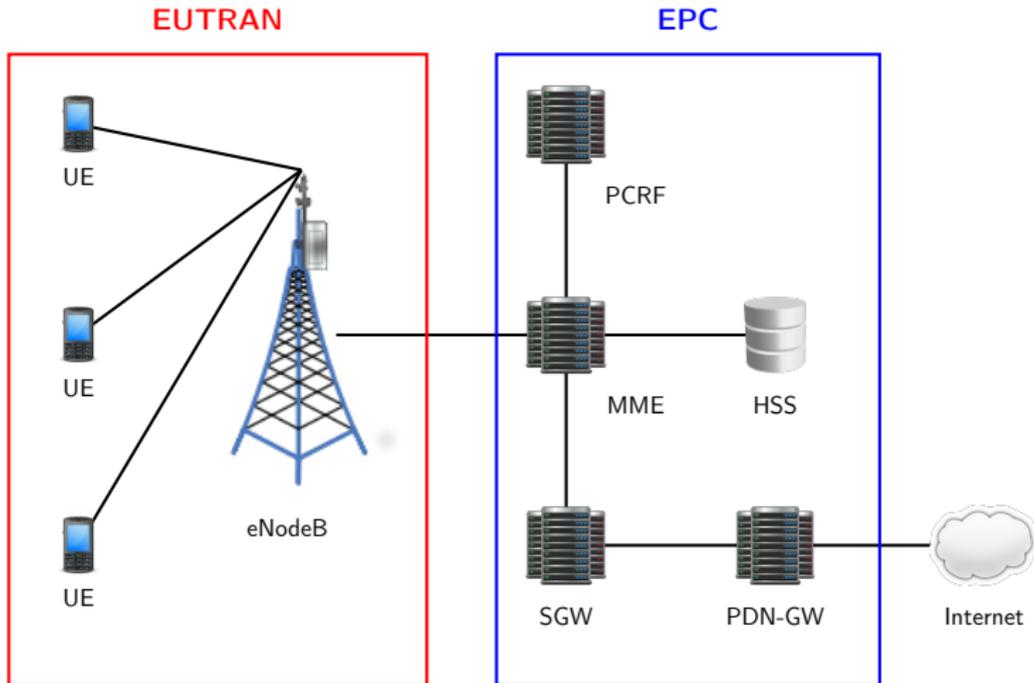


Software

RIDUX

- Erweiterung der MUP
- Schwerpunkt auf LTE-Erweiterung
 - Integration mehrerer Basisstationen (eNodeB)
 - Erweiterung HSS (Home Subscriber Server)
 - Integration Serving- und PDN-Gateway
 - Handover der Teilnehmer

LTE-Netzarchitektur



RIDUX – Fahrzeugintegration



Gliederung

- 2. Ausgangssituation
 - 2.1. Vorbedingungen
 - 2.2. Vorkenntnisse
 - 2.3. Erwartungen
 - 2.4. Herangehensweise
 - 2.5. Umsetzung
 - 2.6. Auswirkungen auf die Entwickler

Ausgangssituation

- Verwaltungsoberfläche auf PHP-Basis war begonnen
- Vorheriges Systemhaus mittlerweile insolvent (dennoch Unterstützung durch einen ehemaligen Entwickler möglich)
- Hohe Lizenzkosten für PHP-Code
- Lieferzeitpunkt (einschließlich Hardware) innerhalb von 18 Monaten
- Redesign der Hardware durch externen Zulieferer
- Zertifizierung der Hardware

Vorbedingungen

- Software-Entwicklung musste von Grund auf aufgebaut werden
- Begrenzte finanzielle Mittel
- Verwaltungsoberfläche über Web (angelehnt an Smart Phones)
- Verwendetes Betriebssystem auf Linux-Basis
- Für die Software-Entwicklung benötigte Server mussten definiert und aufgesetzt werden

Vorkenntnisse

- Geringe Kenntnisse im Bereich von Linux-Systemen
- Bisher nur als Entwickler eingesetzt
- Programmierkenntnisse im Bereich Java und C/C++
- Geringe praktische Erfahrungen im Projektmanagement
- Keine Erfahrungen in der Hardware-Integration
- Entwicklung eher im Bereich Prototypenentwicklung

Erwartungen

- Zeitnahe Entwicklung, um erste Version auf Messe vorstellen zu können (nach ca. sechs Monaten)
- Hardware-Integration mit geforderten Zertifizierungen
 - EMV
 - Umwelteinflüsse (Feuchtigkeit, Staub, Luftdruck)
 - Schwingungen („Rüttel- und Schütteltest“)
- Lieferung von 72 Systemen in einer Rucksackvariante nach 18 Monaten

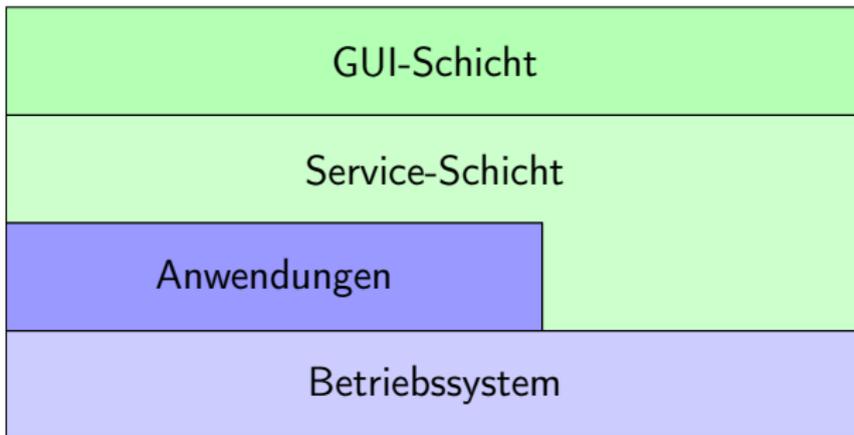
Herangehensweise

- Einarbeiten in Linux
- Schulungen
- Spezialisieren auf bestimmte Anwendungen
- Arbeiten mit realer Hardware soweit möglich

Umsetzung

- Service-Layer in Java erstellt
- Weboberfläche mit [Vaadin](#) in Java umgesetzt
- Service- und GUI-Layer bestehen aus [OSGi](#)-Bundles
- Verwenden von fertiger und im Einsatz befindlicher Open Source Software
- Verwenden von Standards (Protokolle)

Umsetzung



Auswirkungen auf die Entwickler

- Einarbeiten in neues Betriebssystem bzw. Vertiefen der Linux-Kenntnisse
- Arbeiten mit Bash-Script-Dateien
- Beschäftigen mit unterschiedlichen Dateisystemen
- Verwenden von Konsolenfenstern
- Übernahme von Aufgaben in der Serveradministration

Gliederung

- 3. Entwicklungsserver
 - 3.1. Verwendete Server
 - 3.2. Client-Systeme
 - 3.3. Integration in Windows-Landschaft
 - 3.4. Zusammenwirken der Server

Entwicklungsserver

- Die meisten Server laufen unter Debian 7 und Debian 8
- Server laufen meist als virtuelle Maschinen in externem Rechenzentrum

Entwicklungsserver

Versionsverwaltung

- Begonnen mit Subversion unter [FusionForge](#)
- Umstellung auf Git unter [GitLab](#)

Entwicklungsserver

Build-Server

- Continuous Integration mit [Jenkins](#)
- Mittlerweile Jenkins-Cluster
 - Vier Debian-Systeme (darunter Master)
 - Linux-Mint-System für GUI-Tests
 - Windows-7-System zur Erstellung von Windows-Installern

Entwicklungsserver

Repository-Server

- Begonnen mit Artifactory
- Mittlerweile [Sonar Nexus](#)

Entwicklungsserver

Statische Code-Analyse

- [Nexus SonarQube](#)
- Integration in CI

Entwicklungsserver

Dokumentation

- [Weblate](#)
 - Übersetzungen der Java-Properties-Dateien
 - Übersetzen der Android-XML-Dateien
 - Übersetzen der Release-Notes der Updates
- Single Source Publishing mit [Documenta](#)
 - Erstellen der Handbücher
 - Übersetzen der Handbücher

Entwicklungsserver

Aufgaben- und Bug-Verwaltung

- Begonnen mit Eigenentwicklung [JAST](#) und [JAWS](#)
- Mittlerweile [JIRA](#)
- Nachweis der manuellen Tests als Plugin in JIRA integriert

Entwicklungsserver

Produktionssystem

- Debian-System mit MUP (virtuelle Maschine)
- Erweitert um Verwaltung der Systeme, Artefakte und Release-Konfigurationen
- Steuert Continuous Deployment
 - Automatische Produktion der Systeme auf vSphere-Center-Cluster
 - Zeitgesteuerte System- und GUI-Tests auf CI-Server

Entwicklungsserver

QS-System

- Debian-System mit MUP (19-Zoll-Server)
- Erweitert um automatisierte Test-Script-Dateien
- Zusätzliche Hardware zum Testen der Telefonie

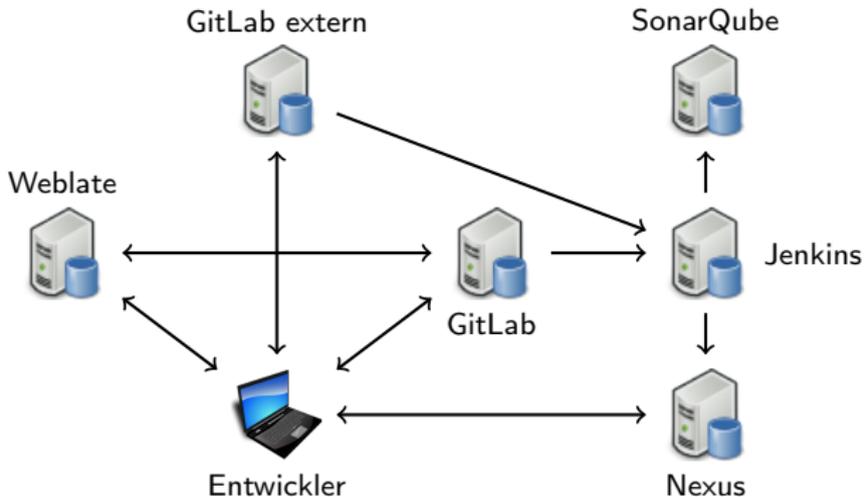
Client-Systeme

- Die meisten Entwickler verwenden Linux (Arch, Ubuntu, Mint)
- Entwicklungsumgebungen [IntelliJ](#) und [Eclipse](#) werden verwendet
- Virtualisierung mit [VirtualBox](#)

Integration in Windows-Landschaft

- Domänenstruktur mit Active Directory (AD)
- Anbindung der Linux-Server über LDAP an AD
- Entwickler greifen über Webseite auf E-Mails zu (Outlook Web Access)
- Server können meist über Weboberfläche bedient werden

Zusammenwirken der Server



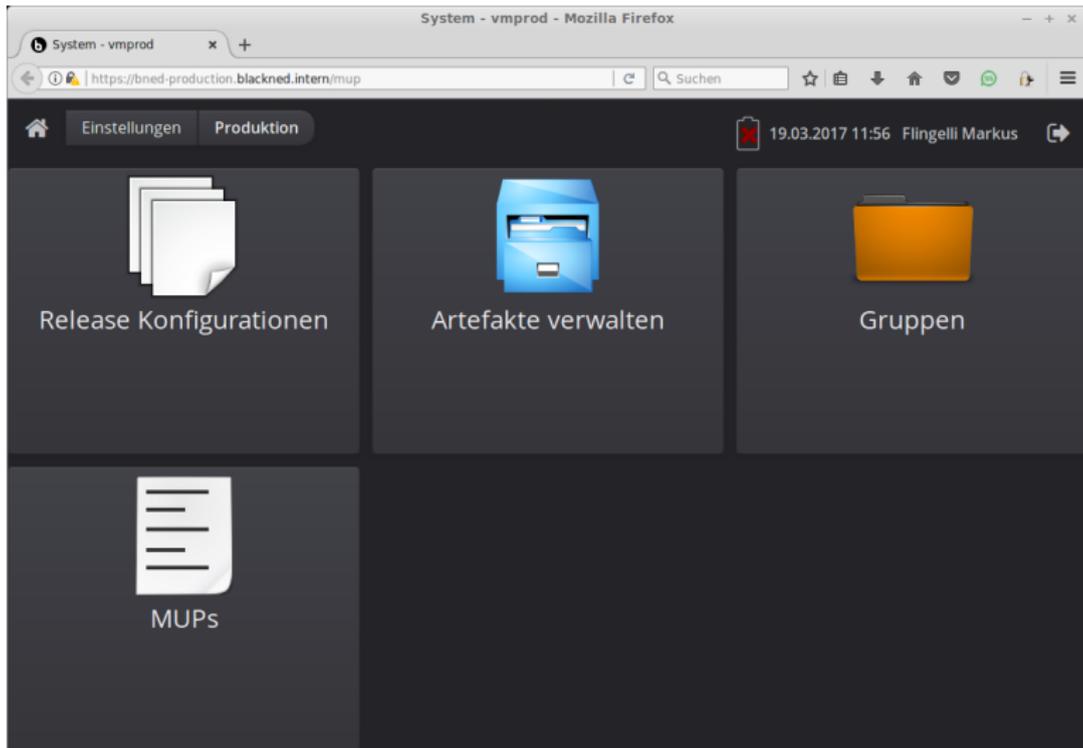
Gliederung

- 4. Beispiele
 - 4.1. Produktionssystem
 - 4.2. Updates erstellen

Produktionssystem

- Clonezilla zum Bespielen mit Basissystem
- Basissystems wird mit gewünschter Konfiguration provisioniert
- Integriert in Weboberfläche

Produktionssystem



Updates erstellen

Allgemeines

- Aktualisierung des Systems über Update-Dateien
 - Debian-Pakete
 - Web-Anwendungen
 - Dokumentation
- Updates sind Archive, die über eine Script-Datei installiert werden

Updates erstellen

Aktualisieren der Debian-Pakete

- Erstellen einer Repository-Datei mit `dpgk-scanpackages`
- Installation der Pakete mit `apt-get`

Updates erstellen

Prinzipieller Aufbau des Scripts

```
#!/bin/bash

deleteOldMupFiles() {

}

update() {

}

deleteOldMupFiles
update

exit 0
```

Updates erstellen

Aktualisieren der Debian-Pakete

```
packageInstaller() {  
    tar xfz /tmp/mup/packages.tar.gz -C /tmp/mup  
    cp -r /tmp/mup/ /fsprotect/system/tmp/  
    echo 'deb file:/tmp/mup/packages ./' > /etc/apt/  
        sources.list  
    apt-get update  
    source /tmp/mup/install.sh  
    apt-get clean  
}
```

Updates erstellen

XML-Tags ändern

```
function fixLdapConfigurationFile {
  local file=/mup/karaf/config/mup-ldap.xml
  if [ -e "$file" ]; then
    sed 's/<loglevel>.*</loglevel>/<loglevel>None
        </loglevel>/g' $file > $file.result
    xmllint --noout $file.result
    if [ $? -eq 0 ]; then
      cp $file.result $file
    fi
    rm $file.result
  fi
}
```

Gliederung

5. Fazit

5.1. Allgemein

5.2. Software-Entwicklung

5.3. Server-Administration

Fazit

Allgemein

- Evaluation verschiedener Anwendungen lohnt sich
- Teilweise muss viel experimentiert werden, da keine oder schlechte Dokumentation vorhanden ist
- Erworbene Kenntnisse kommunizieren
- Feedback an Entwickler oder in Foren geben

Fazit

Software-Entwicklung

- Windows-Entwickler benötigen Einarbeitungszeit
- Entwicklerwerkzeuge vorhanden
- Entwicklung findet nahe am Zielbetriebssystem statt
- Benutzen der Server-Dienste über Weboberflächen
- Arbeiten mit Konsole

Fazit

Server-Administration

- Integration der Linux-Server in Windows-Landschaft meist über LDAP-Anbindung möglich
- Im Wesentlichen Server-Administration über Konsole
- Stabile Server-Dienste (nach Updates keine Neustarts notwendig)
- Windows-Admins lassen meist Finger von Linux-Servern (eigene Administration möglich)

Fragen

