LINUTRONIX
LINUX FOR INDUSTRY

Updates sicher und flexibel gestalten mit Linux

Embedded Computing Conference 2018

Jan Altenberg

Linutronix GmbH

June 05, 2018

LINUTRONIX
LINUX FOR INDUSTRY

Überblick

1. **Update Konzepte**
   - Einführung / Motivation
   - Rescue System
   - Redundantes System
2. **Implementierung von Updates**
   - swupdate
   - Streaming
   - Security
3. **Update Deployment**
   - Lokal
   - Netzwerk
   - Server-basiert
4. **Zusammenfassung**

LINUTRONIX
LINUX FOR INDUSTRY

1. Update Konzepte
   - Einführung / Motivation
   - Rescue System
   - Redundantes System
2. Implementierung von Updates
   - swupdate
   - Streaming
   - Security
3. Update Deployment
   - Lokal
   - Netzwerk
   - Server-basiert
4. Zusammenfassung

LINUTRONIX
LINUX FOR INDUSTRY

Allgemein

- Warum sind Updates notwendig?
  - SECURITY
  - Funktionalität (neue Features, ...)
  - Up-to-date (Fehlerbehebungen, Konfiguration)
  - Root-of-trust (Certificates et.al.)
- Bekannte Herausforderungen mit Updates:
  - Eingeschränkte Ressourcen
  - Unzuverlässige Kommunikation
  - Fehlerhafte oder unvollständige Updates
  - Integrität
- In der heutigen Zeit sind Updates auch für Embedded Systeme zwingend erforderlich!

LINUTRONIX
LINUX FOR INDUSTRY

Allgemeine Anforderungen

- Security
  - Sicherstellen der Updateintegrität
  - Schutz gegen Schadsoftware
  - Nutzung bekannter Verschlüsselungsstandards
- Zuverlässigkeit
  - Umgang mit fehlerhaften oder unvollständigen Updates
- Flexibilität
  - Anpassbarkeit für wechselnde Anforderungen
  - Anpassbarkeit auf neue Umgebungen
  - Portierbarkeit für neue Zielsysteme
- Skalierbarkeit

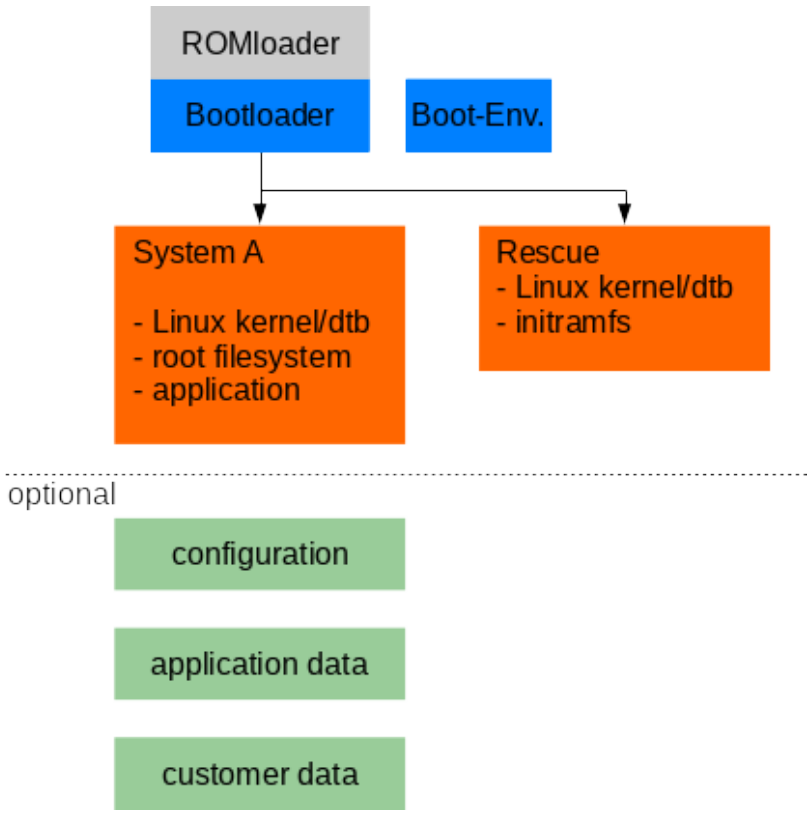
LINUTRONIX
LINUX FOR INDUSTRY

Grundlegende Konzepte

- Updates werden nicht mehr (wie traditionell oft üblich) aus dem Bootloader durchgeführt, sondern aus Linux
- Dies garantiert eine hohe Wiederverwendbarkeit von Treibern und eine hohe Portierbarkeit von Updatkonzepten!
- Grundsätzlich unterscheidet man folgende Updatekonzepte:
  - Rescue System: Immer updatefähig
  - Redundantes System: Immer voll funktionsfähig
- Designentscheidung für das Produkt!

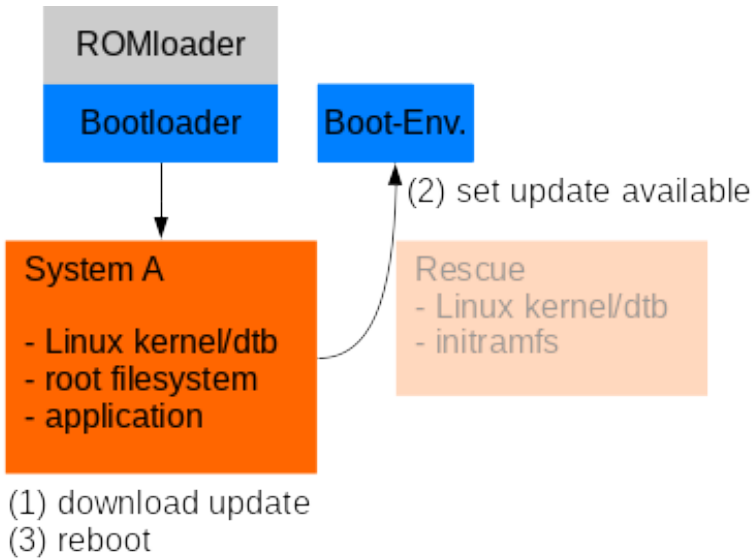
LINUTRONIX
LINUX FOR INDUSTRY

Überblick



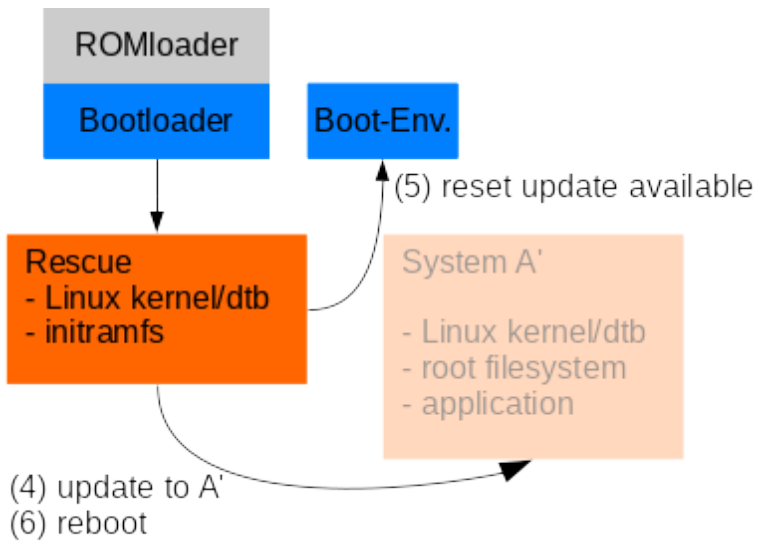
LINUTRONIX
LINUX FOR INDUSTRY

Vorbereiten des Updates



LINUTRONIX
LINUX FOR INDUSTRY

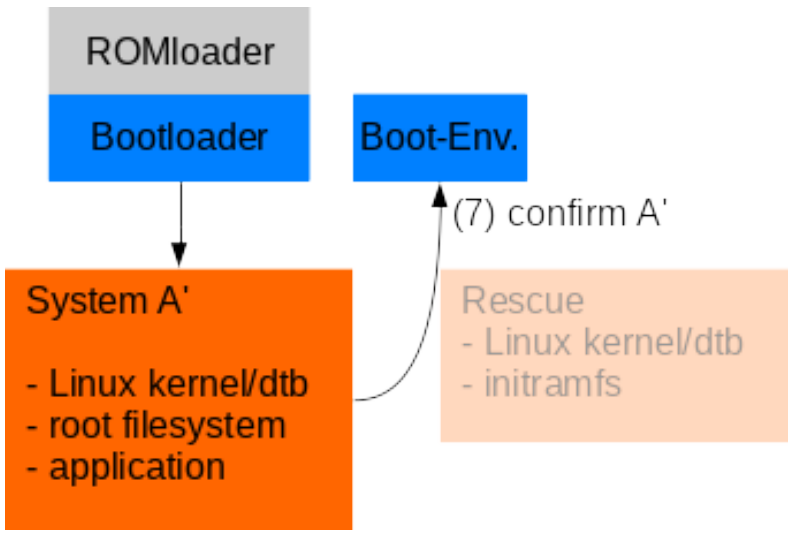
Durchführung des Updates



Reboot (Schritt 6) wird per Watchdog abgesichert!

LINUTRONIX
LINUX FOR INDUSTRY

Bestätigung des Updates



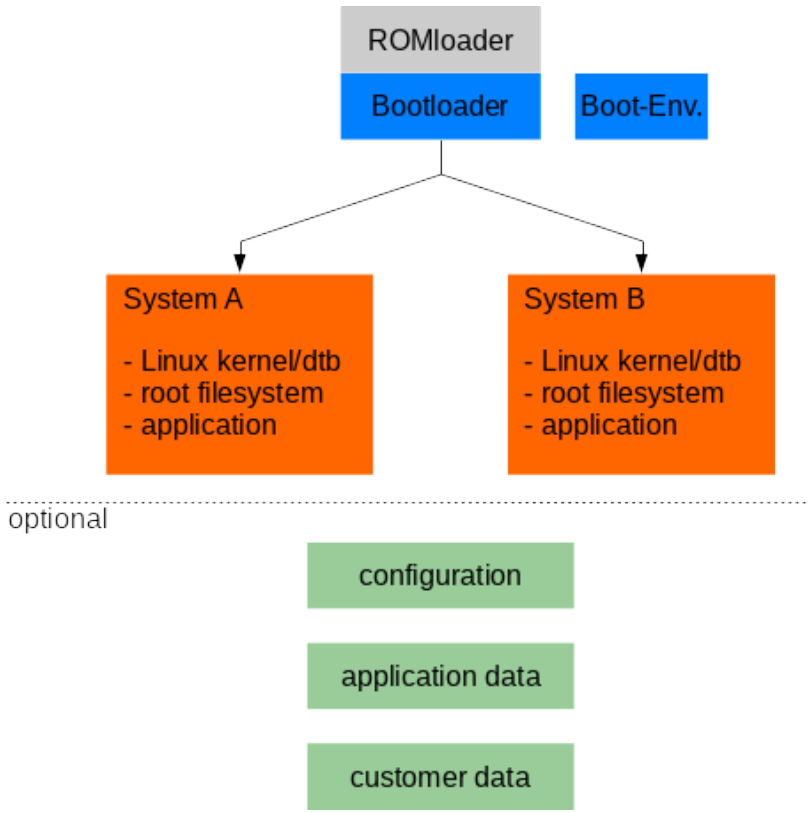
LINUTRONIX
LINUX FOR INDUSTRY

Funktionalität

- Immer updatebares System
- Separates Root-Filesystem Layout
- Im Rescuesystem volle Linux Funktionalität (NAND, USB, WiFi, etc)
- Pros:
  - Wenig Speicherbedarf
  - Hardwarezugriffe können im Updatefall eingeschränkt werden
- Cons.:
  - Zwei Reboots pro Update
  - Produktivsystem während Updateprozess nicht nutzbar
  - Kein Fallback für Produktivsystem

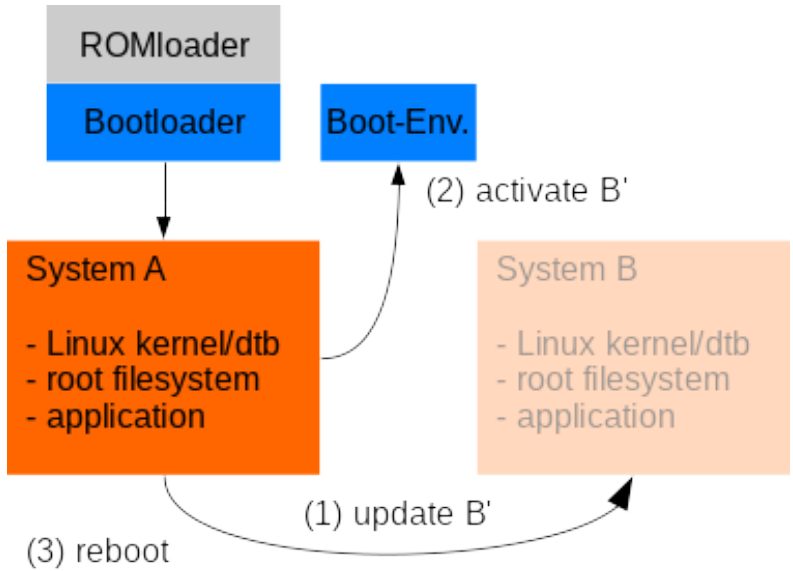
LINUTRONIX
LINUX FOR INDUSTRY

Überblick



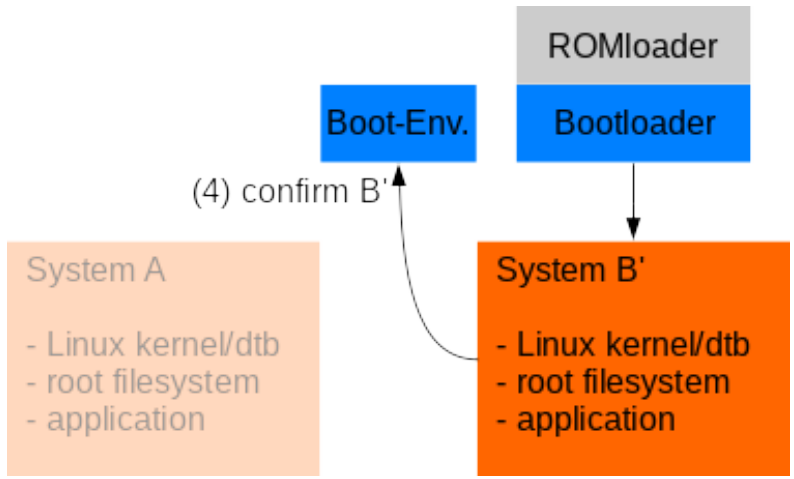
LINUTRONIX
LINUX FOR INDUSTRY

Durchführung des Updates



Reboot (Schritt 3) wird per Watchdog abgesichert!

Bestätigung des Updates



LINUTRONIX
LINUX FOR INDUSTRY

Funktionalität

- System ist immer voll funktionsfähig
- Single Root-Filesystem Layout
- Pros:
  - Fallback für Produktivsystem
  - Nur ein Reboot pro Update notwendig
- Cons:
  - Speicherbedarf

LINUTRONIX
LINUX FOR INDUSTRY

1 Update Konzepte
- Einführung / Motivation
- Rescue System
- Redundantes System

2 Implementierung von Updates
- swupdate
- Streaming
- Security

3 Update Deployment
- Lokal
- Netzwerk
- Server-basiert

4 Zusammenfassung

LINUTRONIX
LINUX FOR INDUSTRY

Das Rad nicht neu erfinden!

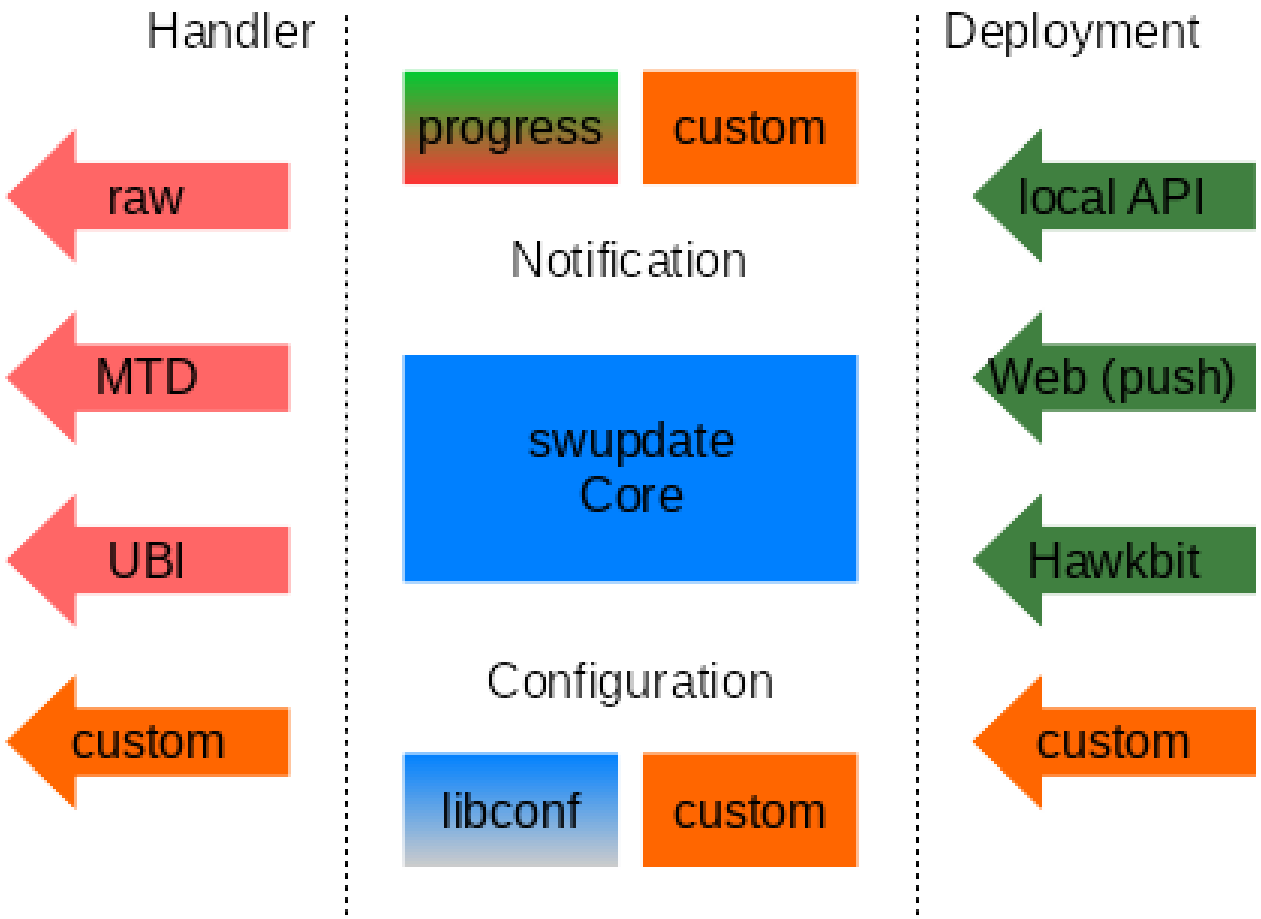
Sowohl für das Einspielen, als auch für das Verteilen von Updates gibt es bereits bewährte Tools!

Features I

- https://github.com/sbabic/swupdate
- Open-Source (GPLv2)
- Bootloader, OS, FPGA Images
- Streamfähige Updates
- Multi-Target Updates
- Powerfail Safe
- Krypto Support
  - Image Signaturen (Integrität)
  - Image Verschlüsselung (Schutz)
- Voll anpassbar
- Erweiterbar

LINUTRONIX
LINUX FOR INDUSTRY

Überblick



LINUTRONIX
LINUX FOR INDUSTRY

Features II

- Integrierte Handler
  - Raw Flash
  - Datei basiert
  - MTD (Linux Subsystem)
  - UBI (Linux Subsystem)
- Integrierte Deployment Methoden
  - Lokale Dateien
  - Netzwerk (pull)
  - Web Interface (push)
  - Hawkbit Deployment Server (push)

LINUTRONIX
LINUX FOR INDUSTRY

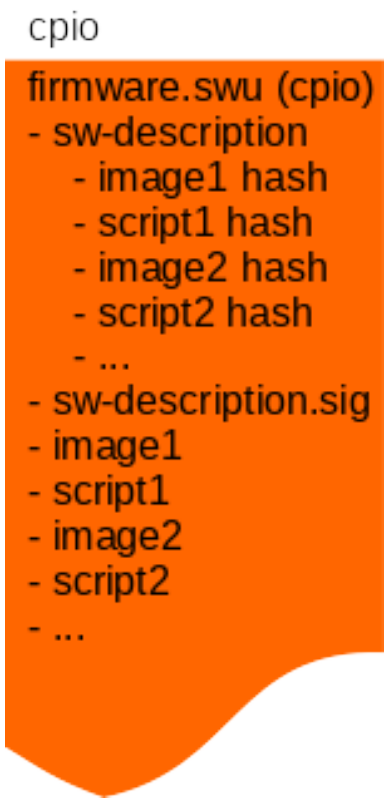
Update Container

- Update Container Format:
  - Serialisiert
  - Streamfähig
  - 1st Element: Beschreibung
  - 2nd Element: Signatur der Beschreibung
- Beschreibungselemente:
  - Image Format
  - Image Size
  - Image Hash
  - Install Target
  - Install Handler

LINUTRONIX
LINUX FOR INDUSTRY

Update Container Format

- File Format: cpio
  - Offenes Dateiformat
  - Weit verbreitet
  - Indiziert
  - Sehr guter Tooling Support
  - Seek wird unterstützt (z.B. für Streaming)



LINUTRONIX
LINUX FOR INDUSTRY

Update Verifikation

- Erzwingen der Signaturprüfung
- Signatur sichert Integrität der Updatebeschreibung
  - Integritätsprüfung zu frühem Zeitpunkt im Update Prozess
  - Hash-basierte Image und Script Validierung
  - Kein Image Download, wenn Integritätsprüfung fehlschlägt

LINUTRONIX
LINUX FOR INDUSTRY

Signed Update Layout

sw-description

```
[...]
stable =
{
  sys_a =
  {
    images =
    (
      {
        filename = "rfs.ext3.gz";
        sha256 = "32df9a4057c858abd2ce64b8bffd2722bd2502d257f221b15295bd9fb85c325b";
        [...]
      }
    );
    scripts =
    (
      {
        filename = "10_markB.sh";
        sha256 = "cbe23f93b00fb758690fc09867c9758b74a15d10522bcc5a4a51f881d2dde0a6";
        type = "postinstall";
      }
    );
  };
[...]
```

LINUTRONIX
LINUX FOR INDUSTRY

1. Update Konzepte
   - Einführung / Motivation
   - Rescue System
   - Redundantes System
2. Implementierung von Updates
   - swupdate
   - Streaming
   - Security
3. Update Deployment
   - Lokal
   - Netzwerk
   - Server-basiert
4. Zusammenfassung

LINUTRONIX
LINUX FOR INDUSTRY

swupdate local Deployment Interfaces

- Direktes Update Processing
  - Einfach und schnell (Kommandozeile)
  - Skriptbar
  - Für Entwicklung und Test
- Socket-basiertes Interface
  - flexibel
  - client API
  - direkte Integration in Applikation möglich

LINUTRONIX
LINUX FOR INDUSTRY

internes Netzwerk

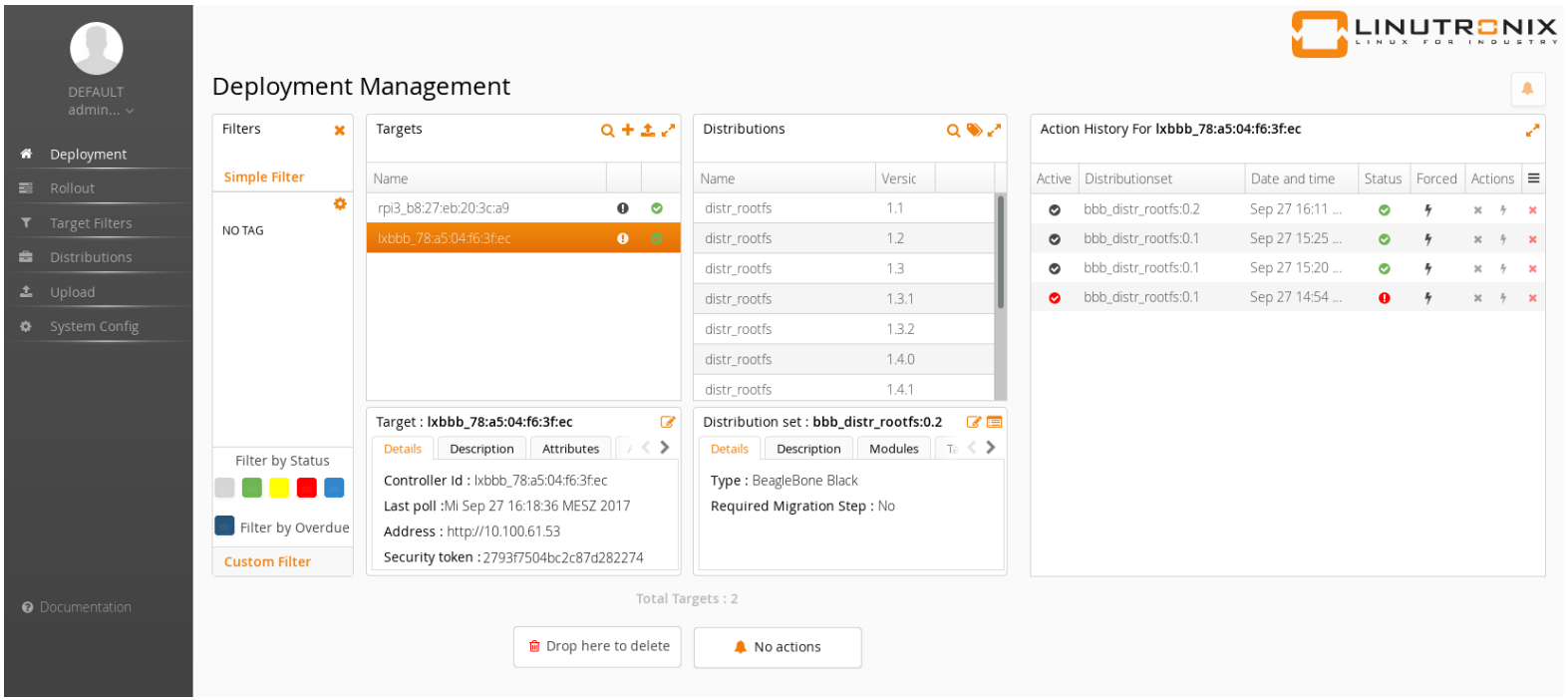
- Direct Update Pull
  - Einfach und schnell (Kommandozeile)
  - "pull Update" von Server URL
  - Für Device-triggered Pull
- interner Web Server
  - Einfaches Web UI
  - Push Updates
  - Neu: besserer WebApp Support

LINUTRONIX
LINUX FOR INDUSTRY

Hawkbit (Server)

- Hawkbit Projekt
  - Open Source (Eclipse Public License 1.0)
  - Ecplipse Projekt
  - Initiiert von Bosch Software Innovations GmbH
- Features
  - Server basiertes grafisches Interface
  - Support für mehrere Protokoll Adapter
  - Flexibles Rollout Management
  - RESTful Management API
- Device Management
  - Datenbank basiert
  - Gerätegruppen
  - Filter
  - "Bulk-Deploy"

Hawkbit Web UI



LINUTRONIX
LINUX FOR INDUSTRY

1. Update Konzepte
   - Einführung / Motivation
   - Rescue System
   - Redundantes System
2. Implementierung von Updates
   - swupdate
   - Streaming
   - Security
3. Update Deployment
   - Lokal
   - Netzwerk
   - Server-basiert
4. Zusammenfassung

Fazit

- Security ist ein wesentlicher Bestandteil des Produktdesigns
- Daher sind Updates auch für Embedded Geräte eine zwingende Anforderung
- Updates aus Linux heraus erhöhen die Wiederverwendbarkeit und Portabilität
- Es gibt fertige Werkzeuge zur Umsetzung von Updatekonzepten
- Diese Werkzeuge sind flexibel und garantieren eine hohe Wiederverwendbarkeit

LINUTRONIX
LINUX FOR INDUSTRY

Vielen Dank für Ihre
Aufmerksamkeit

Linutronix GmbH
Bahnhofstraße 3
88690 Uhldingen-Mühlhofen