TechniSat

Herausforderungen der Software-Entwicklung für Embedded-Devices

Kai Stuhlemmer

Agenda

1. Vorstellung
2. Was ist eine DVB-Set-Top-Box (STB)?
3. Entwicklungsprozess einer DVB-STB
4. Software-Organisation und Architekturanforderungen
5. Ablauf und Herausforderungen der Software-Entwicklung
6. Ausgewählte Bugs
7. Fragen

Über mich

- 1998 Abschluss als Diplom-Informatiker an der TU Dresden
- Seit 20 Jahren bei TechniSat tätig:
  - Treiber-Entwicklung
  - Build-System
  - Portierung auf neue Hardware (OS, Tool-Chain)
  - Hardware-Inbetriebnahme
  - Wartung des Build-Systems
  - Software-QS, Code Review, Schulung

TechniSat Dresden GmbH

- Seit 1990
- Aktuell ca. 80 Mitarbeiter
- Hard- und Software-Entwicklung
- DVB-Settop-Boxen, Fernseher, DAB-Radios, DVB-Multischalter, SmartHome-Zentrale, SmartHome-Geräte
- Auftragsentwicklung

TechniSat Firmengruppe

- Seit 1987
- Unternehmenshauptsitz in Daun/Eifel
- Produktionsstandorte in Schöneck/Vogtland, Staßfurt und Oborniki/PL
- Forschung und Entwicklung in Dresden

Agenda

1. Vorstellung
2. Was ist eine DVB-Set-Top-Box (STB)?
3. Entwicklungsprozess einer DVB-STB
4. Software-Organisation und Architekturanforderungen
5. Ablauf und Herausforderungen der Software-Entwicklung
6. Ausgewählte Bugs
7. Fragen

Begriffe

- **Digital Video Broadcasting (DVB)** – Sammlung von Standards zur digitalen Rundfunkübertragung von Video- und Audiosignalen
- **MPEG** – Sammlung von Standards zur Komprimierung von Audio- und Video-Daten sowie Meta-Daten und deren Multiplex
- **DVB-S2, DVB-T2, DVB-C** – spezifizieren Übertragung über Satellit, Terrestrisch und Kabel
- **Transportstrom (TS)** – Daten-Multiplex nach MPEG-Systems-Standard

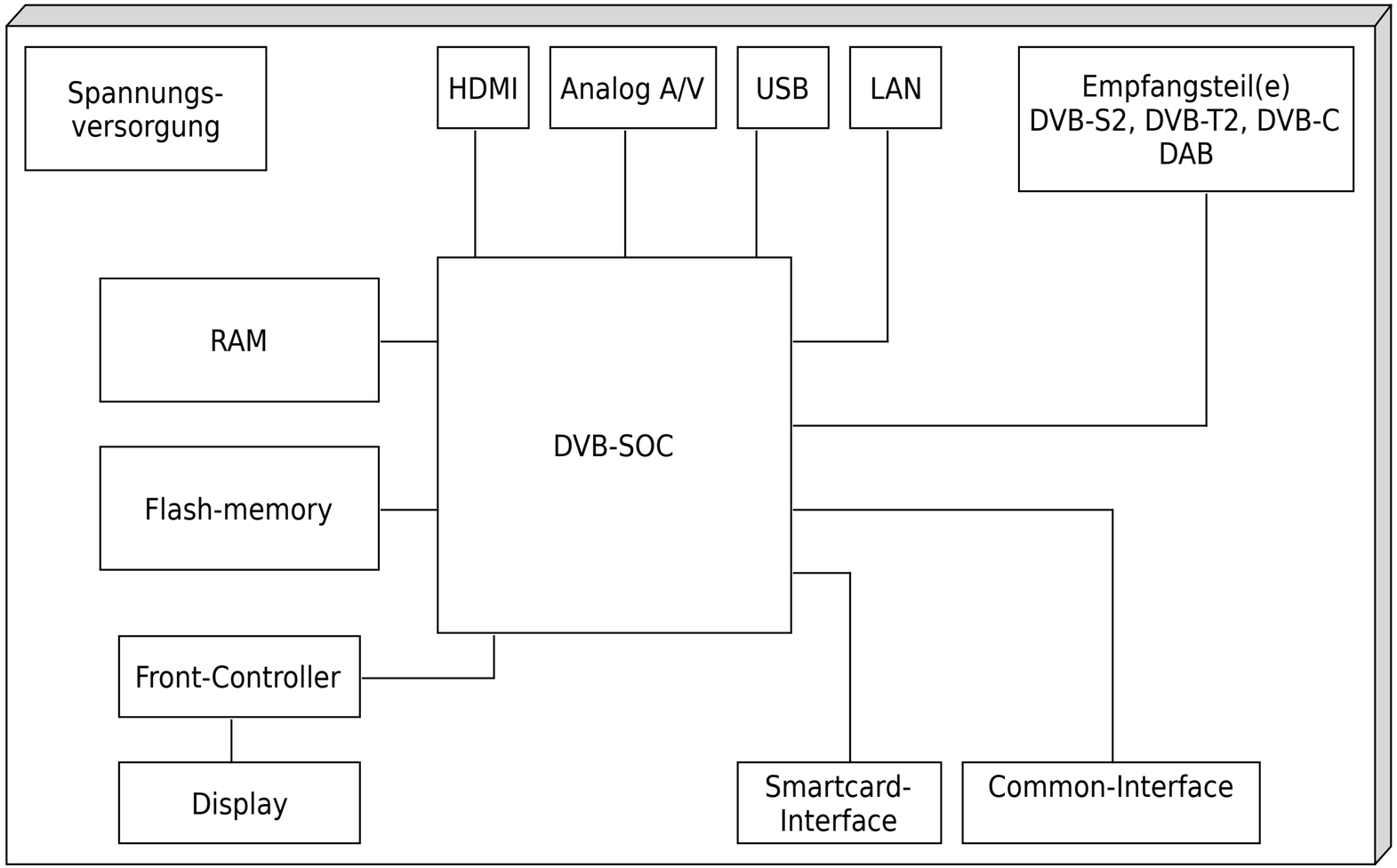
Begriffe #2

- **SD, HD, UHD** – Gruppen von Videoauflösungen
  - **SD, Single Definition**: 720 x 576 (entspricht Analog-TV)
  - **HD, High Definition**: bis 1920 x 1080
  - **UHD, Ultra High Definition**: bis 3840 x 2160

Begriffe #3

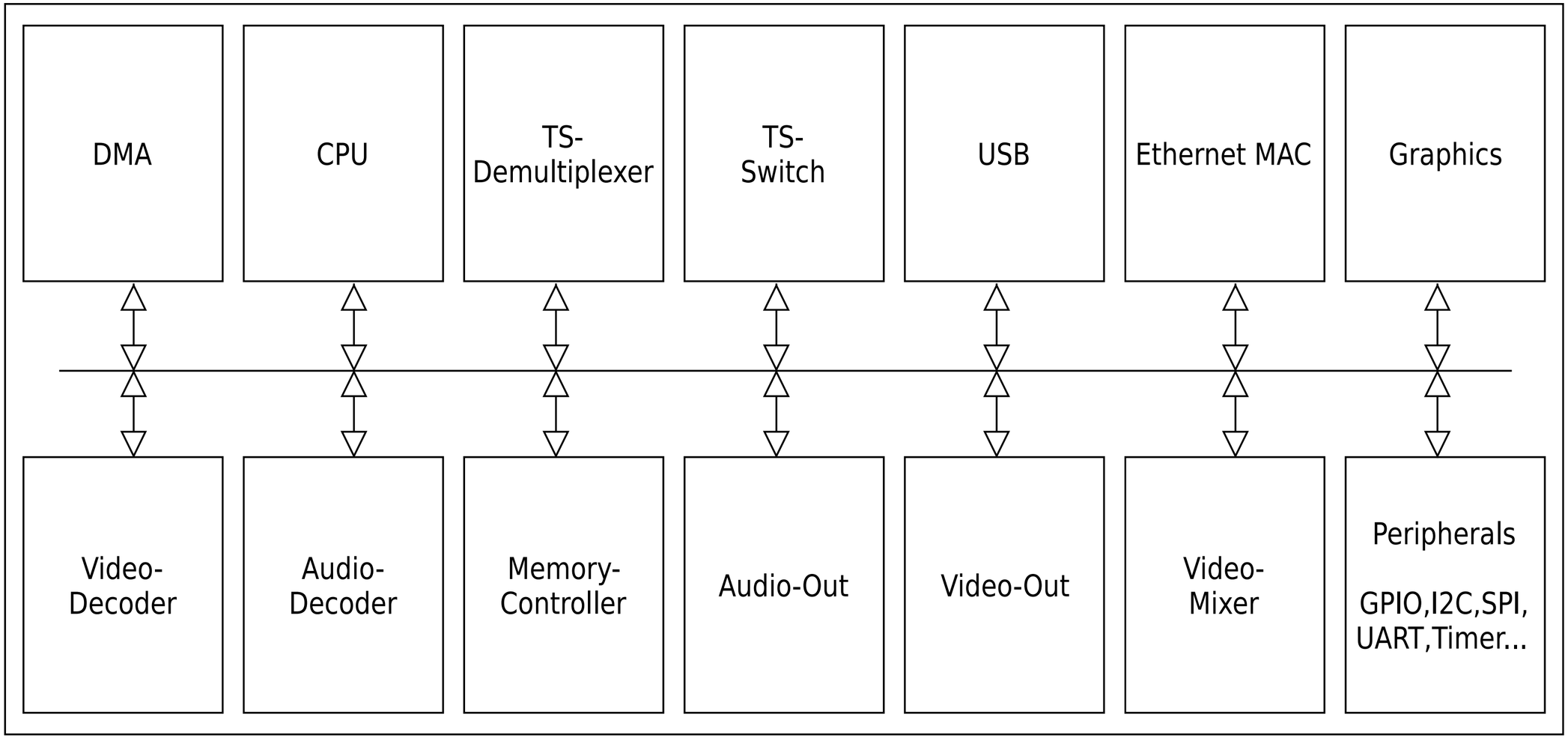
- **Conditional Access System (CAS)** - DRM-System
- **Common Interface (CI, CI+)** – Einschubsystem für das Nachrüsten eines CAS
- **Common Scrambling Algorithm (CSA)** – Verschlüsselungsalgorithmus
- **Descrambler** – Hardware-Einheit zur Entschlüsselung von CSA-verschlüsselten Daten
- **Demultiplexer** – Hardware-Einheit zum Demultiplexen von Transportströmen

Schematischer Aufbau einer STB



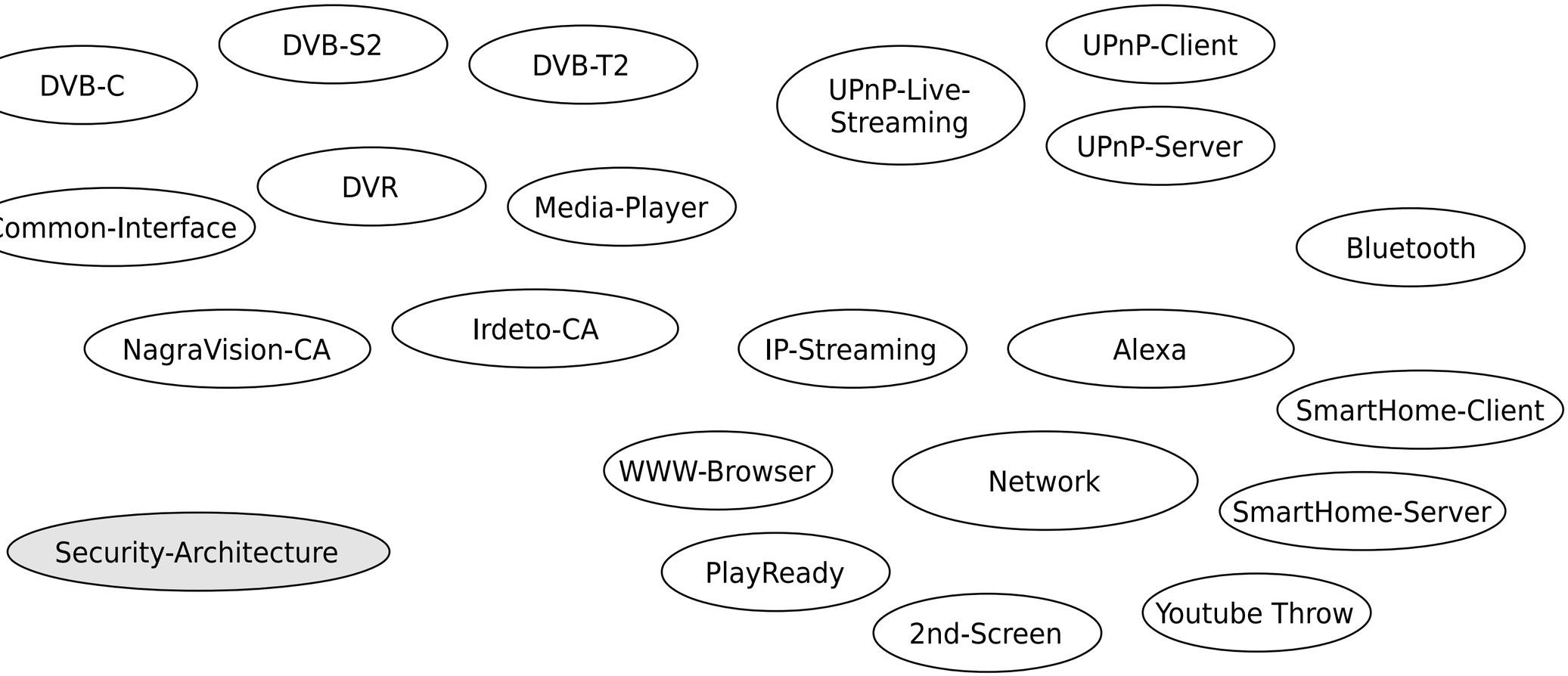
TechniSat

Schematischer Aufbau eines DVB-SOC



TechniSat

Mögliche Features (Auswahl)



TechniSat

Agenda

1. Vorstellung
2. Was ist eine DVB-Set-Top-Box (STB)?
3. Entwicklungsprozess einer DVB-STB
4. Software-Organisation und Architekturanforderungen
5. Ablauf und Herausforderungen der Software-Entwicklung
6. Ausgewählte Bugs
7. Fragen

Hardware-Komponentenauswahl

- Ausgangspunkt: Anforderungen des Auftraggebers (Spezifikation)
- Auswahl eines passenden SOC (System-On-Chip)
- Dimensionierung von RAM und Flash-Speicher
- Auswahl zusätzlicher Komponenten, z.B. WLAN-Modul

Konsequenzen der Komponentenauswahl

- Einfluss auf BOM (Bill-Of-Material) und damit auf Endverkaufspreis
- Einfluss auf mögliche Software-Feature:
  - Hohe Ansprüche an CPU- und Speicher-Ressourcen durch Web-Browser
  - Erhöhte Anforderung an CPU-Ressource durch IP-Streaming
  - Limitierung der möglichen Software-Feature durch Speichergrenzen (RAM und Flash)

Elektrische und mechanische Konstruktion

- Anzahl der Leiterplattenlagen haben Einfluss auf die BOM
- Spannungsversorgungsleitungen müssen richtig dimensioniert sein
- Mögliches Übersprechen zwischen Leitungen muss beachtet werden
- Verbindung zu RAM-ICs ist kritisch
- Neue Gehäuse erfordern neue Werkzeuge in der Produktion
- Schaltplan-Review

Software-Portierung

- Verwendung der Referenz-Hardware des SOC-Herstellers
- Support einer neuen Tool-Chain im Build-System
- Evtl. Implementierung der OS-Abstraction-Layer für weiteres RTOS
- Implementierung der Driver-Abstraction-Layer (i.d.R. Adaption an Board-Support-Package des SOC-Herstellers)
- Oft nur noch kleinere Anpassungen an endgültige Hardware nötig

Erste Hardware-Prototypen

- Materialorderung, kleine Menge Leiterplatten evtl. im Schnellservice
- Bestückung in begrenzter Menge im Produktionswerk (i.d.R. keine weitere Funktionskontrolle)
- Hardware-Inbetriebnahme
  - Prüfung der Spannungsversorgungen

Software-Inbetriebnahme

- Vorprogrammierter Flash-Speicher mit 1st-stage-loader
- Evtl. mit Hilfe eines Hardware-Debuggers via JTAG-Interface
- Schrittweise Inbetriebnahme einzelner Funktionskreise, z.B. Video-Output, OSD, Empfangsteile, Schnittstellen (LAN, USB)
- Portierung der vollständigen Applikation

Konformitätsprüfungen

- Elektromagnetische Verträglichkeit (EMV)
- ESD-Prüfungen (ESD – ElectroStatic Discharge)
- Analoge Audio- und Videopegel
- Ökodesign-Richtlinie (Standby-Verbrauch)
- USB, LAN, Common Interface
- Kennwerte der Empfangsteile (z.B. Empfindlichkeit über spezifizierten Frequenzbereich)
- Dolby, CI+, Embedded CAS

Produktionsvorbereitung

- Portierung und Anpassung der Produktionstest-Software
- Flash-Image zur Vorprogrammierung erstellen
- Unterstützung des Produktionsbetriebes bei der Einrichtung der Fertigungslinie

Produktionstest-Software

- Ermöglicht Testung in der Produktionslinie
- Finalisiert das Gerät (endgültige Software, Seriennummer, CA-Schlüssel)
- Aktiviert Schreibschutz
- Deaktiviert Hardware-Debug-Schnittstellen
- Aktiviert Secure-Boot

Produkt-Qualitätssicherung

- Testabteilung in Dresden
- Testung beim Auftraggeber
- Freigabeprozeduren für Hard- und Software
- Produktionstests
- Support-Center
- Lange Versorgung mit Software-Updates

Agenda

1. Vorstellung
2. Was ist eine DVB-Set-Top-Box (STB)?
3. Entwicklungsprozess einer DVB-STB
4. Software-Organisation und Architekturanforderungen
5. Ablauf und Herausforderungen der Software-Entwicklung
6. Ausgewählte Bugs
7. Fragen

DVB Software-Stack

- Lange Historie (seit etwa 11/2000)
- Anfangs reiner C-Code, später Migration zu C++/C++11
- Viele Jahre monolithische Applikation mit RTOS-Kern (Nucleus, uITRON, OS20, OS21), später auch Linux-System
- Unterstützung verschiedener Tool-Chains
- Unterstützung verschiedener Prozessorarchitekturen (MIPS, ARM, ST20, ST40)
- GNU make basierter Build-Prozess

Software-Organisation

- Verwaltung mit Versionsverwaltungssystem Subversion
- Branches mit verschiedenen Rollen:
  - Software-Platform-Branches, stabile Software-Basis für eine Menge von Hardware-Plattformen
  - Entwickler-Branches für die Feature-Entwicklung und Portierung auf neue Hardware-Plattformen (abgeleitet von und zurück integriert nach Software-Platform-Branch)
  - Release-Branches, abgeleitet von Software-Platform-Branches
- Tags für Release-Kandidaten und Releases

Qualitätssicherung in der Software-Entwicklung

- Compiler-Warnungen sind Fehler
- Continuous-Build-System
- Continuous-Test-System
- Statische Code-Analyse (Klocwork)
- Code-Reviews
- Test-getriebene Entwicklung auf dem PC
- Schulungen

Anforderungen an die SW-Architektur

- Konfigurierbarkeit (CI, CA, DVR, WWW...)
- Sparsamer Umgang mit Ressourcen (CPU, Speicher)
- Ressourcenbedarf (CPU, Speicher) skaliert mit Konfiguration
- Portierbarkeit auf neue Hardware

Sicherstellung der Anforderungen

- Klare Abstraktionsschnittstellen (z.B. OS, Treiber)
- Schlanke UI-Implementierung
- C-Präprozessor-Schalter (nicht schön, aber effektiv)
- Messung und Optimierung
- Build-System ermöglicht umfangreiche Konfiguration des gewünschten Zielsystems

Beispiel Security-Architecture (SecArch)

- CAS-Provider fordern hohe Isolation einzelner Funktionskreise (DVB, DVR, Mediaplayer, WWW...)
- Zerlegung in verschiedene Linux-Prozesse
- Isolation in Linux-Container (LXC)
- Maximale Beschränkung der Privilegien

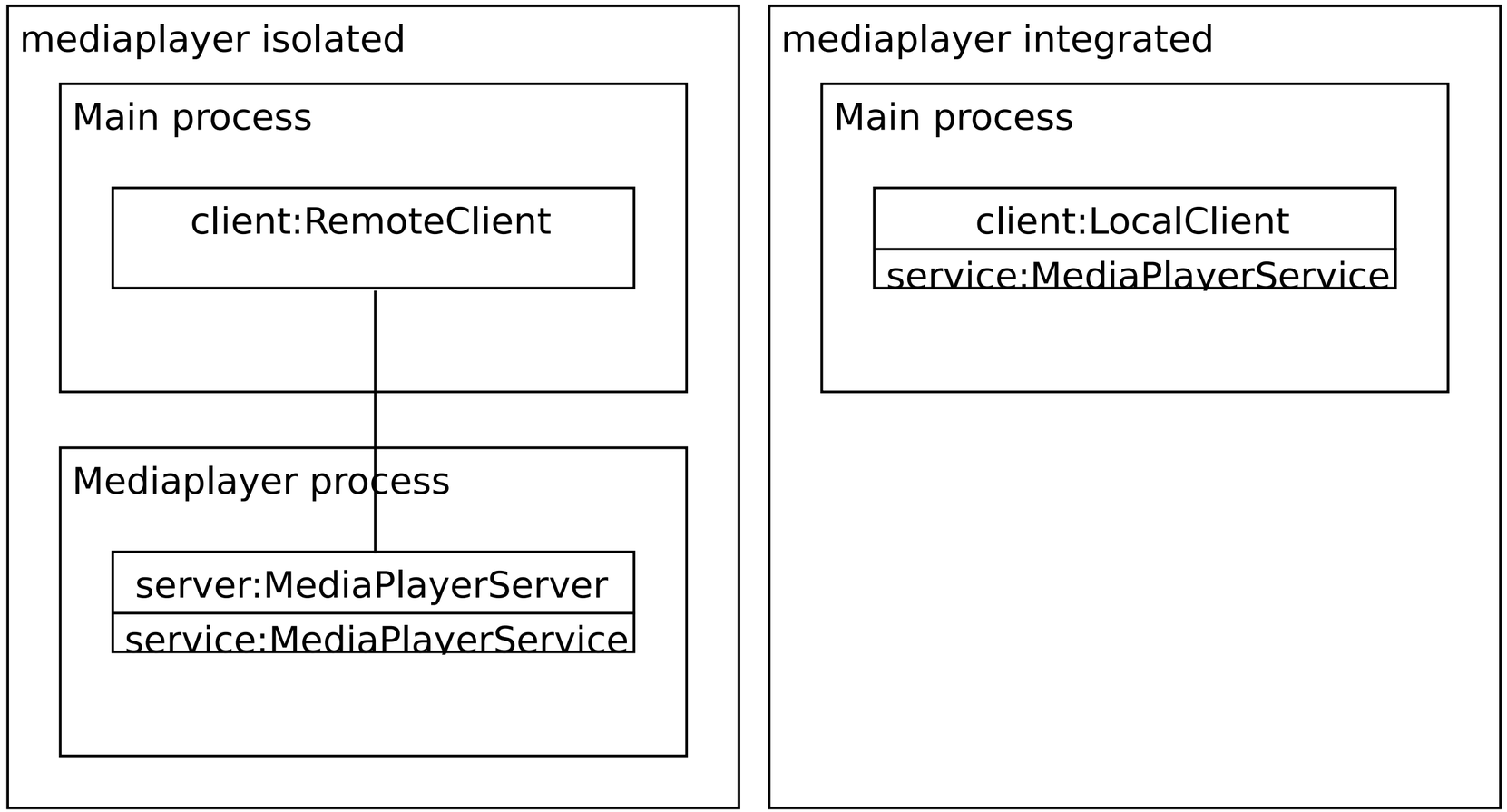
SecArch: Konsequenzen

- Mehr Prozesskontextwechsel
- Erhöhter Speicherbedarf
- Zusätzlicher Ressourcenbedarf durch IPC

→ Anforderung ist nicht für alle Geräte nötig

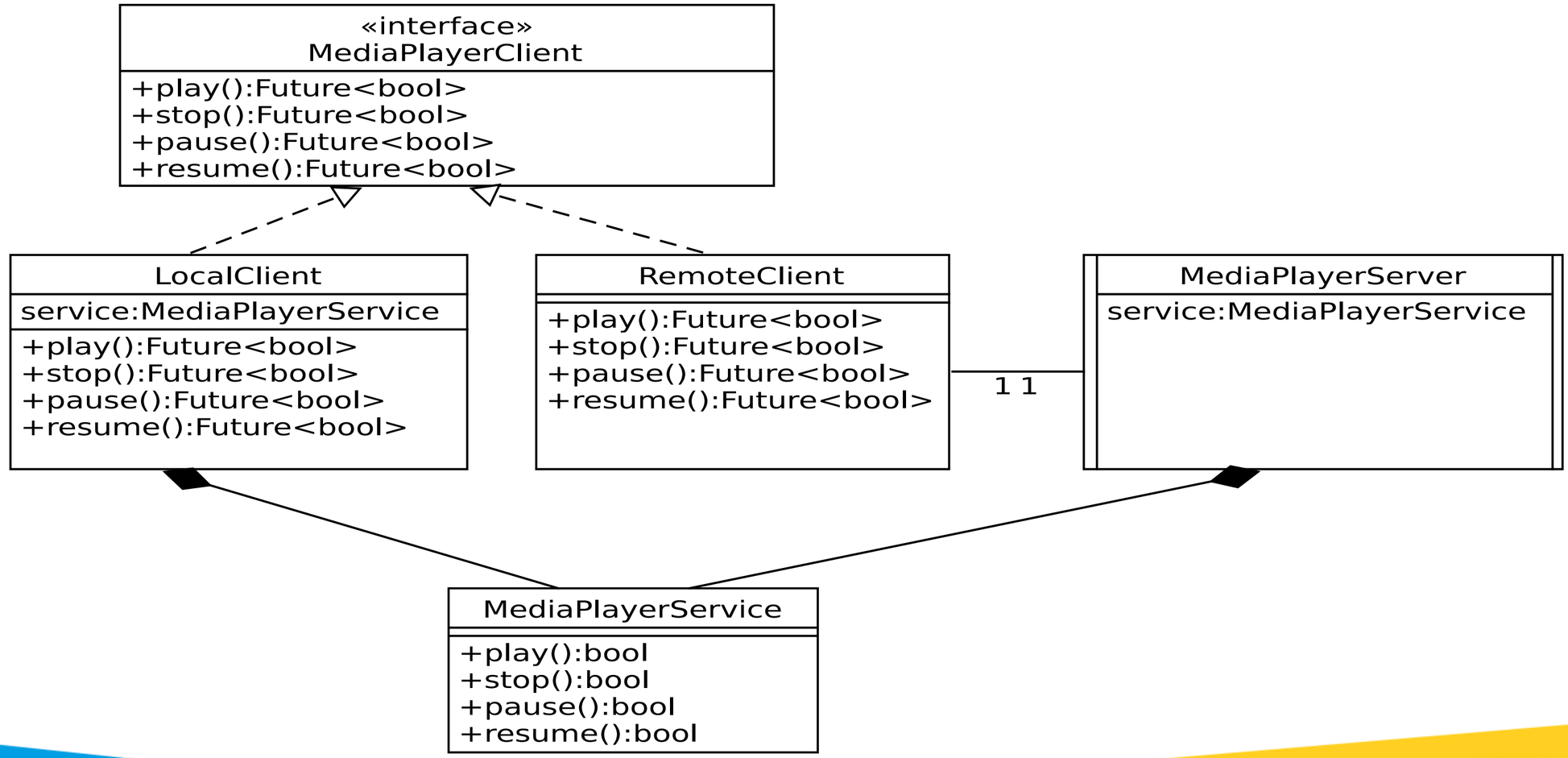
→ Muss Konfigurationsoption des Systems sein

Systemkonfiguration am Beispiel Media-Player mit und ohne SecArch



TechniSat

Klassenhierarchie des Media-Players

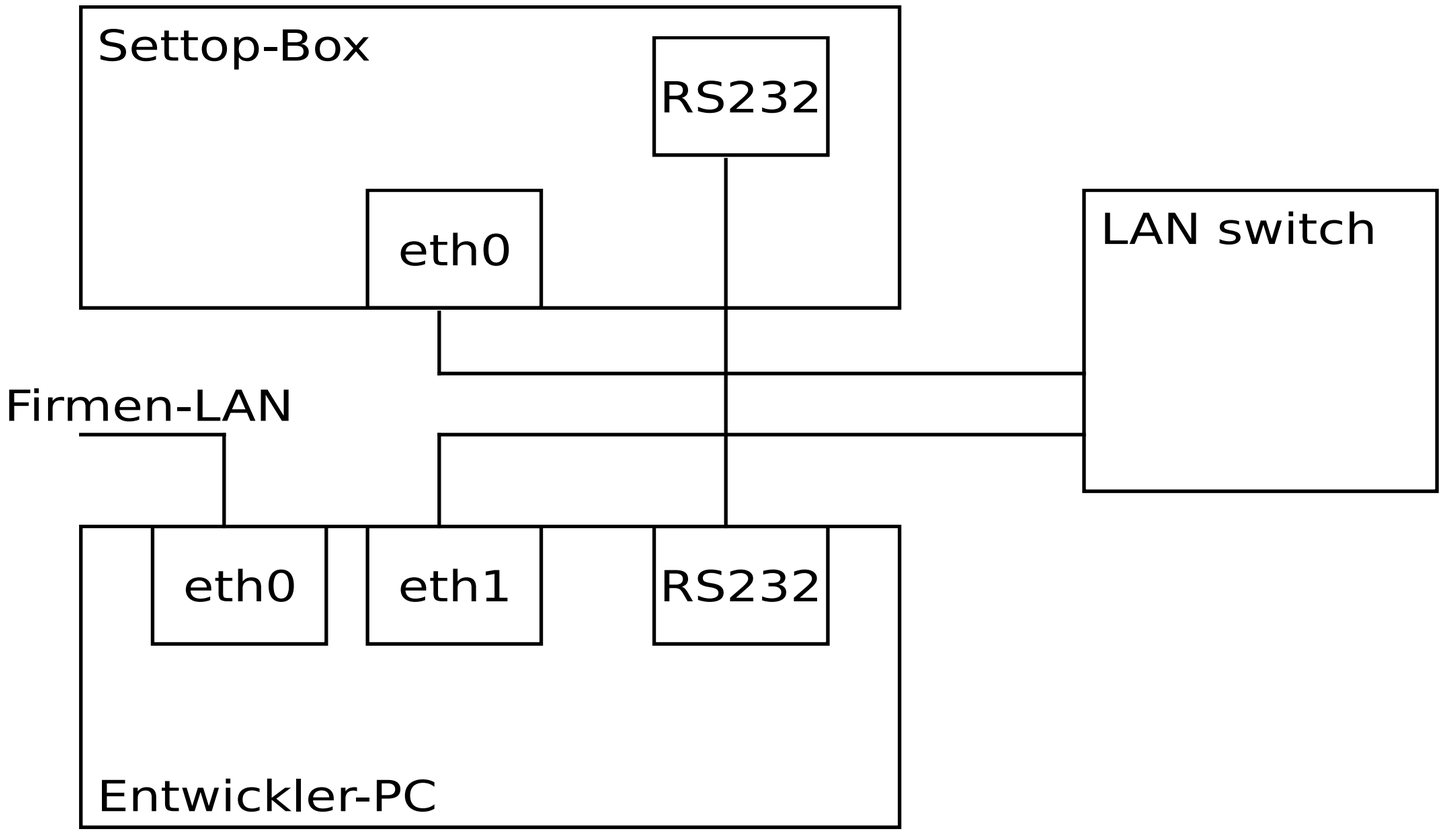


TechniSat

Agenda

1. Vorstellung
2. Was ist eine DVB-Set-Top-Box (STB)?
3. Entwicklungsprozess einer DVB-STB
4. Software-Organisation und Architekturanforderungen
5. Ablauf und Herausforderungen der Software-Entwicklung
6. Ausgewählte Bugs
7. Fragen

Arbeitsplatz eines SW-Entwicklers - schematisch



TechniSat

Arbeitsplatz eines SW-Entwicklers

TechniSat

Ablauf der Software-Entwicklung

1. Checkout/Update vom Versionsverwaltungssystem (Subversion)
2. Code editieren (SlickEdit, Eclipse, Visual Studio Code, ...)
3. SW bauen, Kernel und Root-FS werden in tftp-Ordner kopiert
4. STB starten (Development-Loader lädt SW per tftp und startet diese)
5. Automatischer oder manueller Test
6. Commit ins Versionsverwaltungssystem

Debugging an der STB

- „Print“-Debugging
- Debug-Hooks über serielle Schnittstelle
- Core-Dump-Analyse
- Hardware-Debugger über JTAG-Schnittstelle
- Telnet-Session auf dem Gerät
- Regressionstests

Besondere Herausforderungen

- Probleme in 3rd-Party-Code erfordern:
  - Reduktion auf ein Minimalbeispiel
  - Kommunikation mit dem FAE (Field Application Engineer)
  - Schnelles Verstehen von fremden Code
- Hardware-nahe Fehlersuche erfordert Umgang mit entspr. Messmitteln (z.B. Oszilloskop)
- Fehlerhafte Eingangsdaten (z.B. im Transportstrom) müssen als Ursache erkannt und konserviert werden

Software-Komponenten

- Abgeschlossene Funktionseinheiten mit eigener Versionshistorie
- Bauen aufeinander auf
- Entwicklung ausschließlich auf Entwickler-PC
- Test-getriebene Entwicklung (TDD)
- Bis auf Debug-Code frei von Präprozessor-Schalterung

Software-Komponenten - Vorteile

- Schnellere Entwicklungszyklen
- Kürzere Build-Zeiten
- Besseres Tooling (Valgrind, Sanitizer, Debugger)
- Hohe Testabdeckung
- Wiederverwendung in anderen Projekten

Agenda

1. Vorstellung
2. Was ist eine DVB-Set-Top-Box (STB)?
3. Entwicklungsprozess einer DVB-STB
4. Software-Organisation und Architekturanforderungen
5. Ablauf und Herausforderungen der Software-Entwicklung
6. Ausgewählte Bugs
7. Fragen

„Tatort“-Bug - Problem

- Beschreibung: Gerät stürzt beim „Tatort“ ab.
- Eingrenzung:
  - Aufzeichnung des problematischen Transportstroms
  - Sukzessives Abschalten von Funktionen bringt keine Behebung

➔ Verdacht: Ursache liegt im Video-Elementarstrom (V-ES), der unabhängig von der CPU durch den Video-Dekoder verarbeitet wird.

„Tatort“-Bug - Lösung

- Ursache: Fehlerhafte Werte in den Videodaten führen zu Speicherüberschreibungen durch den Video-Dekoder.
- Fix: Ein Fix des Hardware-Video-Dekoders ist nicht möglich. Eine Verlagerung der ihm zugeordneten Speicherbereiche an das Ende des Speichers vermeidet zumindest die Überschreibung anderer kritischer Speicherbereiche.

Teletext-Bug - Problem

- Beschreibung: Teletext wird auf dem angeschlossenen TV nicht richtig angezeigt.
- Eingrenzung:
  - Teletext-DMA wird öfter nicht im Zeitrahmen beendet
  - Untersuchung des analogen Video-Signal

➔ Verdacht: Fehler im SOC

Teletext-Bug - Lösung

- Ursache: Ungünstige Einstellungen im Memory-Arbiter
- Fix: Fine-Tuning des Memory-Arbiter durch Field-Applikation-Engineer (FAE)

Timer-Bug - Problem

- Beschreibung: Die Uhr läuft zu langsam.
- Eingrenzung:
  - Sekundentakt wird durch periodischen SW-Timer realisiert
  - SW-Timer basiert auf OS-Zeitbasis
  - OS-Zeitbasis wird durch Hardware-Timer realisiert
- Ursache: Der Hardware-Timer ist nicht frei laufend. Er wird immer wieder neu gestartet.

Timer-Bug – Setup Code

```
enum Register
{
    COUNT,
    COMPARE
};

volatile uint32_t* reg(Register); // Get pointer to HW register
void advanceOSTick();

// timer clock is 82.944MHz
static const uint32_t period = 82944; // nr. of ticks for 1ms

void timer_setup()
{
    *reg(COMPARE) = period;
    *reg(COUNT)   = 0;
}
```

Timer-Bug - *ISR*

```
void timer_isr()
{
    uint32_t countValue            = *reg(COUNT);
    uint32_t compareValue          = *reg(COMPARE);
    uint32_t elapsedSinceLastTick = countValue % compareValue;
    uint32_t ticks                 = countValue / compareValue;

    *reg(COMPARE) = compareValue;
    *reg(COUNT)   = elapsedSinceLastTick;

    do
    {
        advanceOSTick();
    } while (--ticks);
}
```

Timer-Bug – *ISR fixed*

```
void timer_isr()
{
    uint32_t countValue            = *reg(COUNT);
    uint32_t compareValue          = *reg(COMPARE);
    uint32_t elapsedSinceLastTick = period + (countValue - compareValue);
    uint32_t ticks                 = elapsedSinceLastTick / period;

    *reg(COMPARE) = period + (ticks * period);

    do
    {
        advanceOSTick();
    } while (--ticks);
}
```

Timer-Bug reloaded

- Beschreibung: Der Ton setzt sporadisch aus und das Gerät ist nicht bedienbar. Diese Blockade löst sich aber nach längerer Zeit eigenständig wieder auf.
- Eingrenzung:
  - Blockade ist immer etwa 51 Sekunden lang

- Verdacht: Fix des ersten Timer-Bugs ist fehlerhaft
- Ursache: Timer-Implementierung enthält eine Timing-Lücke

Timer-Bug reloaded – *ISR fixed*

```
void timer_isr()
{
    uint32_t countValue            = *reg(COUNT) + (period / 16);
    uint32_t compareValue          = *reg(COMPARE);
    uint32_t elapsedSinceLastTick = period + (countValue - compareValue);
    uint32_t ticks                 = elapsedSinceLastTick / period;

    *reg(COMPARE) = period + (ticks * period);

    do
    {
        advanceOSTick();
    } while (--ticks);
}
```

Agenda

1. Vorstellung
2. Was ist eine DVB-Set-Top-Box (STB)?
3. Entwicklungsprozess einer DVB-STB
4. Software-Organisation und Architekturanforderungen
5. Ablauf und Herausforderungen der Software-Entwicklung
6. Ausgewählte Bugs
7. Fragen

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?