



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich



Bernhard Tellenbach

# Vom Fahrgastinformations- zum Infotainmentsystem

Master Thesis MA-2005-20  
November 2004 to April 2005

Tutor: Ulrich Fiedler  
Co-Tutor (Ascom): Norbert Graipel  
Supervisor: Prof. Bernhard Plattner

### **Abstract**

Die Aufgaben, die ein Fahrgastinformationssystem(FIS) in Zügen bewältigen soll, werden immer umfangreicher. Neben Streckeninformationen soll auch Unterhaltung und Werbung über diese Systeme verbreitet werden. In der Masterarbeit wird aufgezeigt, wie mit dem FIS der Ascom als Basis, Unterhaltung und Werbung in Zügen realisiert werden kann. Es werden Lösungen zur Multimedia Datenverteilung, zur Steuerung und Planung der Inhalte, sowie zur Anzeige dieser Inhalte präsentiert und beurteilt. Dabei werden sowohl technische als auch wirtschaftliche Aspekte berücksichtigt. Weiter wird der Bau und die Konfiguration eines Demonstrators zur Umsetzung und Evaluation der erfolgsversprechendsten Lösungen beschrieben. Dazu gehören detaillierte Erläuterungen zur Funktionalität und Struktur der für die einzelnen Teilprobleme verwendeten oder selbst entwickelten Software. Die auf dem Demonstrator durchgeführten Evaluationen bestätigen schliesslich, dass mit den vorgeschlagenen Lösungen ein Infotainmentsystem realisiert werden kann.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Motivation . . . . .	8
1.2	Beschreibung der Aufgaben . . . . .	8
1.3	Ähnliche Arbeiten und Produkte . . . . .	9
1.4	Überblick . . . . .	9
<b>2</b>	<b>Das Fahrgastinformationssystem</b>	<b>11</b>
2.1	Kommunikation im Zug . . . . .	11
2.1.1	Der Train Communication Network(TCN) Standard . . . . .	12
2.2	Das mobile Fahrgastinformationssystem der Ascom . . . . .	13
2.2.1	Das RAILVOX . . . . .	15
<b>3</b>	<b>Übertragen und anzeigen der Infotainment-Daten</b>	<b>17</b>
3.1	Systemstrukturen und Datenformate . . . . .	17
3.1.1	Kennzahlen zu den unterstützten Datenformaten . . . . .	18
3.1.2	Fehlende Unterstützung für Power Point Präsentationen . . . . .	18
3.2	Der dezentrale Ansatz . . . . .	18
3.2.1	Konzepte basierend auf Unicast . . . . .	20
3.2.2	Konzepte basierend auf Unicast und Multicast . . . . .	23
3.2.3	Konzepte basierend auf Multicast . . . . .	25
3.2.4	Probleme und Lösungen beim realen Netzwerk . . . . .	27
3.3	Der zentrale Ansatz . . . . .	31
3.4	Design Entscheid: Variante Zentral oder Dezentral . . . . .	33
3.4.1	Evaluation . . . . .	33
3.4.2	Entscheid für die Variante Dezentral . . . . .	34
3.5	Auswahl der zu evaluierenden Konzepte . . . . .	35
3.5.1	Multicast: Multicast . . . . .	35
3.5.2	Multicast: Screen Export . . . . .	35
<b>4</b>	<b>Der Demonstrator</b>	<b>37</b>
4.1	Verwendete Hardware . . . . .	38
4.1.1	Speichererweiterung des Master Panel PCs . . . . .	39
4.2	Verwendete Software . . . . .	41
4.2.1	Betriebssystem . . . . .	41
4.2.2	Grafische Oberfläche/Grafikausgabe . . . . .	42
4.2.3	Applikationen . . . . .	43
4.2.4	VNC Server und Client . . . . .	43
4.2.5	Multicast Dateitransfer . . . . .	43
4.2.6	Streamen und Anzeige von Videos . . . . .	43
4.3	Performance Evaluation . . . . .	45
4.3.1	Multicast:Screen Export . . . . .	47
4.3.2	Multicast:Multicast . . . . .	47
4.4	Fazit . . . . .	48

<b>5</b>	<b>Kommunikations- und Kontrollebene</b>	<b>51</b>
5.1	Software Architektur der Kommunikations- und Kontrollebene . . . . .	51
5.2	Kommunikation über das Ethernet . . . . .	52
5.2.1	Massnahmen zur Erkennung von fehlgeleiteten Paketen . . . . .	54
5.2.2	Absichern der Kommunikation gegen Angreifer . . . . .	54
5.3	Dienste auf dem Master . . . . .	55
5.3.1	Der MasterMsgModule Dienst . . . . .	55
5.3.2	Der MasterFileTF Dienst . . . . .	55
5.3.3	Der MasterPlayer Dienst . . . . .	56
5.3.4	Der UnicastReceiver Dienst . . . . .	56
5.4	Dienste auf den Slaves . . . . .	60
5.4.1	Der ClientMsgModule Dienst . . . . .	60
5.4.2	Der ClientFileTF Dienst . . . . .	60
5.4.3	Der ClientPlayer Dienst . . . . .	60
5.4.4	Der MulticastReceiver Dienst . . . . .	60
5.5	Vom Master und Slave benötigte Dienste . . . . .	65
5.5.1	WatchDog . . . . .	65
<b>6</b>	<b>Darstellung von Fahrgastinformation und Unterhaltung</b>	<b>67</b>
6.1	Das Ebenenmodell . . . . .	67
6.1.1	Lösungen für verschiedene Anordnungen der Ebenen . . . . .	68
6.2	Das Ebenenmodell auf dem Demonstrator . . . . .	69
6.2.1	Realisierung des Ebenenmodells . . . . .	69
<b>7</b>	<b>Ablaufplanung der Multimedia Inhalte</b>	<b>71</b>
7.1	Algorithmus zur Sicherstellung der Planbarkeit der Inhalte zu jedem Planungszeitpunkt . . . . .	71
7.1.1	Bedingungen für die Zulassung eines Inhalts . . . . .	72
7.2	Algorithmus zur Auswahl des zu planenden Inhalts . . . . .	74
7.2.1	Auswahlkriterien . . . . .	74
7.2.2	Algorithmus . . . . .	74
7.2.3	Algorithmus zur Auswahl des zu planenden Inhalts auf dem Demonstrator . . . . .	75
<b>8</b>	<b>Fazit und Ausblick</b>	<b>79</b>
8.1	Fazit . . . . .	79
8.2	Ausblick . . . . .	79
<b>A</b>	<b>RCM Erweiterungen</b>	<b>81</b>
A.1	RCM Erweiterung: Multicast . . . . .	81
A.1.1	Spezifikation der benötigten RCM Funktionalität . . . . .	81
A.2	RCM Erweiterung: Priorisierung . . . . .	83
A.2.1	Priorisierung anhand der Zieladresse . . . . .	83
A.2.2	Priorisierung gemäss IEEE 802.3ac . . . . .	83
A.3	RCM Erweiterung: ID basiertes VLAN . . . . .	83
A.3.1	Spezifikation der benötigten RCM Funktionalität . . . . .	83
<b>B</b>	<b>Tool zur Editierung der Playlist</b>	<b>87</b>
<b>C</b>	<b>Installation des Demonstrator Systems</b>	<b>89</b>
C.1	Installation von ELinOS v3.1 . . . . .	89
C.2	Konfiguration und Anpassung des ELinOS Linux Betriebssystem . . . . .	89
C.2.1	Hinzufügen eines als Source Code vorliegendes Programms . . . . .	89
C.3	Installieren eines ELinOS Projekts auf die Harddisk des Panel PCs . . . . .	90
C.3.1	Konfiguration des Entwicklungsrechners und des Panel PCs . . . . .	90
C.3.2	Installation des von ELinOS erzeugten Systems auf die Harddisk des Panel PCs . . . . .	92

<b>D</b>	<b>Software für den ECM 3611 SBC</b>	<b>93</b>
D.1	Kernelmodule für den CLE266 . . . . .	93
D.2	Installation und Konfiguration von XFree86 v4.4.0 . . . . .	93
D.3	Installation des Mozilla Browsers und des Macromedia Flash Plug-Ins . . . . .	95
D.4	Konfiguration und Installation der Software zum Abspielen von Videos . . . . .	95
D.4.1	Installation der Wrapper Bibliothek . . . . .	96
D.4.2	Installation der VideoLan Software . . . . .	96
D.4.3	Installation der MPlayer Software . . . . .	97
D.4.4	Installation der mp2player Software . . . . .	98
D.5	Installation und Bedienung der Kommunikations- und Kontrollebene . . . . .	99
D.5.1	Starten der Kommunikations- und Kontrollebene . . . . .	99
<b>E</b>	<b>Konfiguration der Dienste des Infotainmentsystems</b>	<b>101</b>
E.1	MasterMsgModule . . . . .	101
E.2	MasterFileTF . . . . .	102
E.3	MasterPlayer . . . . .	103
E.4	UnicastReceiver . . . . .	103
E.5	ClientMsgModule . . . . .	103
E.6	ClientFileTF . . . . .	104
E.7	ClientPlayer . . . . .	104
E.8	MulticastReceiver . . . . .	105
E.9	PlanerApp . . . . .	105
<b>F</b>	<b>Konfiguration des Kernels für den ECM 3611 SBC</b>	<b>107</b>



# Kapitel 1

## Einleitung

Ein Fahrgastinformationssystem zu entwickeln, erfordert ein breites elektrotechnisches Know-how. Die Komplexität dieser Systeme, sowie auch die Systeme selbst werden meist kaum bewusst wahrgenommen. Erst das Fehlen der automatischen Ansage oder ein ausbleibender Wechsel des Anzeigeninhalts bei Systemausfällen rückt das Fahrgastinformationssystem ins Bewusstsein. Die Vielseitigkeit der Problemstellungen bei der Entwicklung eines solchen Systems verlangt den Ingenieuren einerseits viel ab, verspricht aber andererseits auch eine herausfordernde und interessante Arbeit. Unter anderem muss der Ingenieur bahnkonforme Hardware entwickeln oder integrieren, er muss für die Verarbeitung von Sensordaten, die z.B. vom Wegimpulsgeber kommen, sorgen und er muss die gesamte Software zur Steuerung und Verteilung der Fahrgastinformationen entwickeln. Die Integration neuer Technologien, welche den Betrieb der Fahrgastinformationssysteme vereinfachen oder welche das wachsende Bedürfnis der Fahrgäste nach Information befriedigen, werden auch in Zukunft für eine abwechslungsreiche und interessante Arbeit sorgen. Das Angebot der Gruppe für Fahrgastinformationssysteme des Bereichs Communication der Security Solutions Division der Firma Ascom in Bern Bümpliz, im Rahmen einer Masterarbeit ihr Fahrgastinformationssystem so zu erweitern, dass mit diesem neben Fahrgastinformationen auch Unterhaltung in Form von Videoclips und Animationen gezeigt werden können, war deshalb höchst willkommen.

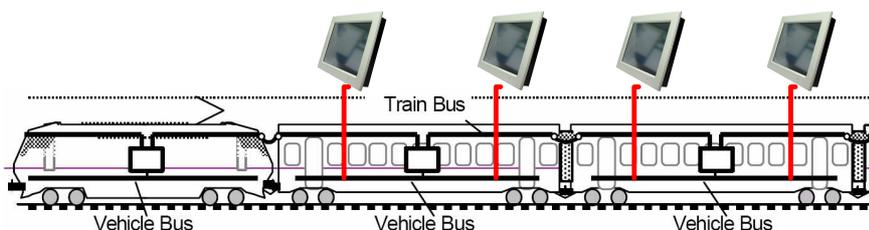


Abbildung 1.1: Zug mit Anzeigen für die Darstellung von Fahrgastinformationen

Speziell interessant war, dass neben der reinen Konzipierung der Erweiterung auch der Bau eines Demonstrators zur Implementation und Evaluation der vorgeschlagenen Konzepte zu den Aufgaben gehörte. Es stellte sich heraus, dass diese Erweiterung in der Tat die Lösung unterschiedlichster Probleme erforderte. Nach der Einarbeitung in das Fahrgastinformationssystem der Ascom und den im Bahnumfeld geltenden Anforderungen an Hardware und Software galt es, als erstes zu untersuchen, wo und wie die zu zeigenden Inhalte gespeichert werden und wie diese im Zug zur Anzeige gelangen. Anschliessend konnte die zum Konzept passende Hardware und Software bestimmt und der Demonstrator gebaut und konfiguriert werden. Überraschen mag dabei die Erkenntnis, dass das Abspielen von Videos auf für den industriellen Einsatz geeigneter und auf Platzbedarf optimierter Hardware wegen mangelnder Performance kaum ohne Tricks möglich ist. Als letztes wurden noch die Konzepte für die Steuerung und Planung der zu zeigenden Inhalte entwickelt und auf dem Demonstrator umgesetzt.

## 1.1 Motivation

Information, Unterhaltung und Werbung werden heute nicht nur zu Hause vor dem Fernseher konsumiert, sondern per Mobiltelefon auch unterwegs. Auch im Bereich des öffentlichen Verkehrs(öV) ist ein klarer Trend hin zu immer besseren Fahrgastinformationssystemen(FIS) zu erkennen. Vermehrt werden auch bereits Infotainmentsysteme(IS) eingesetzt, die neben der reinen Information(INFO) auch noch Unterhaltung(enterTAINMENT) in Form von Filmen oder Werbung anbieten. Mit der Erweiterung des FIS der Ascom<sup>1</sup> zum Infotainmentsystem kann einerseits dem Kunden des öV ein Mehrwert in Form von Unterhaltung und andererseits einem öV Unternehmen ein Mehrwert in Form einer zusätzlichen Einnahmequelle geboten werden. Die zusätzliche Einnahmequelle ist die mit dem IS mögliche Verbreitung von Werbung. Die Ascom ist ein Schweizer IT Unternehmen, das vor allem in den Bereichen Telekommunikation, Netzwerke und öffentlicher Verkehr tätig ist. In den Bereich öffentlicher Verkehr gehört auch das sehr flexible und vielseitige FIS der Ascom. Es bietet viele Anschluss- und Erweiterungsmöglichkeiten. Neben Matrix- und Textanzeigen können z.B. auch sogenannte Panel PCs zur Anzeige von Fahrgastinformationen(FI) angeschlossen werden. Unter einem Panel PC versteht man die Kombination von Rechner und Anzeige in einem möglichst kleinen und flachen Gehäuse. Meist wird für den Rechner ein Single Board Computer(SBC) verwendet. Details zum Panel PC der Ascom sind im Abschnitt 4.1 zu finden. Das Potential der Panel PCs wird momentan aber kaum ausgenutzt. Es wird nur wenig mehr Information als vorher auf den ein- oder mehrzeiligen Matrixanzeigen angeboten. So wird z.B. während der Fahrt eine sogenannte Perlschnur angezeigt. Damit ist eine Linie mit jeweils einem Kreis für bereits passierte und noch anzufahrende Stationen gemeint. Der nächste Halt wird darauf speziell hervorgehoben (z.B. farblich). Ein Blick auf die Verwendung der Bildschirme in Flugzeugen macht aber klar, welches Potential neben dem Vermitteln von FI noch in den Anzeigen steckt. Dort werden neben den Informationen zum Flug auch Kinofilme, Werbung und Informationssendungen gezeigt. Der Bedarf nach zusätzlichen Diensten wie dem Anzeigen von Werbung und kurzen Infofilmen scheint auch im Zugverkehr vorhanden zu sein. Einige Konkurrenten bieten jedenfalls bereits solche Systeme für den Einbau in Busse oder fixe Zugkompositionen (Tram, ICN, u.s.w.) an.<sup>2</sup> Mit dem RAILVOX Konzept könnte Infotainment aber auch auf frei konfigurierbaren Zügen zur Realität werden. Die Erweiterung des FIS der Ascom zum Infotainmentsystem hat das Potential, dessen Attraktivität entscheidend zu steigern. Diese Masterarbeit entwickelt deshalb Konzepte und Lösungen, welche den Weg des FIS der Ascom hin zum Infotainmentsystem ebnen sollen.

## 1.2 Beschreibung der Aufgaben

Ein FIS besteht aus mehreren statischen und mobilen Komponenten. Im Rahmen dieser Masterarbeit wird nur der mobile Teil des FIS betrachtet. Das Ziel ist die Machbarkeit einer Erweiterung des FIS zum IS durch Entwicklung und Evaluation von verschiedenen Lösungsansätzen zu prüfen, sowie den vielversprechendsten Ansatz auf einem kleinen Testsystem umzusetzen. Das erste Teilziel ist die für das Entertainment nötigen Daten von einem Panel PC an weitere Panel PCs zu verteilen und auf diesen anzuzeigen. Hierzu müssen folgende Aufgaben gelöst werden:

- Konzept zur Verteilung der Daten
- Hardware für den Panel PC organisieren und zusammenbauen.
- Betriebssystem und Programme für den Panel PC konfigurieren und installieren
- Abspielen von Videos im MPEG-(1,2, ev. 4) und von Animationen im Flash Format auf einem Panel PC
- Abspielen von Videos und Animationen auf mind. zwei Panel PCs gleichzeitig

Das zweite Teilziel ist die Realisierung der Steuerung für den Wechsel von FI zu Entertainment und umgekehrt. Damit verbunden sind folgende Aufgaben:

<sup>1</sup> Informationen zur Ascom sind unter <http://www.ascom.ch> zu finden

<sup>2</sup>s. Abschnitt 1.3 für Details

- Konzept für die Steuerung
- Software zur Steuerung der Panel PCs entwickeln

Das dritte Teilziel ist die Definition eines Regelsystems zur Bestimmung der Abspielreihenfolge der Entertainment Inhalte. Damit verbunden sind folgende Aufgaben:

- Regelset definieren
- Algorithmus für die Bestimmung der Abspielreihenfolge
- Steuerung aufgrund des Regelsets implementieren

## 1.3 Ähnliche Arbeiten und Produkte

Es gibt bereits viele FIS auf dem Markt, die auch Infotainment ermöglichen. All diese Systeme sind vom Konzept her relativ ähnlich. Der mobile Teil besteht aus einem zentralen Rechner, der für die Verteilung der vom Infotainmentsystem benötigten Daten zuständig ist. Die Art, wie die Daten zu den Anzeigen gelangen, sowie der Grad an Intelligenz in den Anzeigen variiert. Als Beispiel seien hier das System von Innova Semiconductors [3] genannt, welches in der Berliner Metro im Einsatz ist. Dieses spielt die zentral gespeicherten Daten auch zentral ab und überträgt das Grafiksignal über ein proprietäres Kommunikationsnetz, das Gigastar genannt wird, an die Anzeigen. Die Anzeigen besitzen neben dem Empfänger mit einer (optionalen) Konfigurationschnittstelle/Rückkanal über RS485 keine Intelligenz. Andere, vor allem in Bussen eingesetzte Systeme, verteilen das Grafiksignal der zentral gespeicherten und abgespielten Inhalte analog über VGA Kabel oder digital über DVI Kabel. Diese Varianten werden im Abschnitt 3.3 noch näher vorgestellt. Das primäre Einsatzgebiet des FIS der Ascom ist der Zug. Dort ist die Versorgung der Anzeigen mit Infotainment Inhalten durch einen zentralen Rechner pro Zug wegen den fehlenden schnellen Kommunikationskanälen über die Kupplungen der Waggons hinweg nicht möglich. Die verfügbaren Systeme könnten aber Lösungsansätze für das System innerhalb eines Waggons liefern. Dies vor allem bezüglich der verwendeten Softwarelösungen für das Verteilen der Multimedia Daten. Leider wurde bei der Suche nach ähnlichen Arbeiten kein System gefunden, das wie die von der Ascom verwendeten Panel PCs, auf Linux basiert. Es existiert allerdings eine umfangreiche Doktorarbeit zu einem System 'mobil info' mit dem Titel: Digitales multimediales Fahrgastinformationssystem für Infotainmentanwendungen in öffentlichen Verkehrsmitteln [1], welche auf einer Windows Plattform basiert. Wo Gemeinsamkeiten vorhanden sind, werden diese hier konkretisiert und erweitert. Dies bezieht sich vor allem auf die Kapitel 6 und 7 dieser Arbeit. Die der 'mobil info' Arbeit fehlenden konkreten Konzepte zur Verteilung und Anzeige der Infotainment Inhalte ist ein Kernpunkt dieser Arbeit. Weitere Arbeiten gibt es vor allem in den Bereichen Kommunikation. Es sei hier das EU Projekt TrainCom [2] genannt, das eine neue und schnellere Kommunikationsinfrastruktur für die Kommunikation mit dem Zug und innerhalb des Zuges definiert.

## 1.4 Überblick

Abbildung 1.2 stellt die in der Masterarbeit betrachteten Abstraktionsebenen im Zug dar und listet die Beiträge dieser Arbeit zur jeweiligen Ebene inklusive zugehöriger Kapitelnummer(n) auf. Dieser Bericht ist wie folgt gegliedert: Kapitel 1, zu dem auch dieser Überblick hier gehört, nennt die ausschlaggebenden Gründe für das Schreiben dieser Arbeit, liefert die Aufgabenstellung und verweist auf ähnliche Arbeiten und Produkte. Kapitel 2 stellt den Train Communication Network Standard zur Kommunikation innerhalb des Zuges vor und leitet aus diesem Konsequenzen für FIS in Zügen ab. Es folgt die Einführung des zum Infotainmentsystem zu erweiternde FIS der Ascom. Kapitel 3 beschreibt Konzepte zur Übertragung und Anzeige der Infotainment Daten. Nach der Auswertung der Konzepte anhand von Kriterien wie Skalierbarkeit, Implementationsaufwand und Kosten wird schliesslich ein Konzept zur Realisierung ausgewählt. Kapitel 4 macht Angaben zum Aufbau und zur Konfiguration des Demonstrators. Zudem wird die zur Realisierung des Konzeptes aus Kapitel 3 benötigte Software und die damit auf dem Demonstrator erreichten Performance Messungen präsentiert. Kapitel 5 befasst sich mit

dem Konzept und der Implementation der für die Steuerung der Abläufe im Infotainmentsystem notwendigen Kommunikations- und Kontrollebene. In Kapitel 6 wird das Konzept zur Darstellung der Infotainment Inhalte und der Fahrgastinformationen vorgestellt und dessen etwas abgeänderte Umsetzung auf dem Demonstrator erklärt. In Kapitel 7 ist schliesslich das Konzept für den Algorithmus zur Ablaufplanung der Infotainment Inhalte beschrieben. Inwiefern dieses Konzept auf dem Demonstrator umgesetzt wurde, kann ebenfalls dort nachgelesen werden. Die Arbeit schliesst mit einer Zusammenfassung des Erreichten und einem kurzen Ausblick in Kapitel 8. Der Anhang enthält Anleitungen zum Aufbau und zur Konfiguration des Demonstrators und des Entwicklungsrechners. Auch Details zu einigen in der Arbeit nur kurz beschriebenen Erweiterungen können dort gefunden werden.

2 Zug				
2 FIS der Ascom				
	IS	Konzept	Implementiert	Ausgewählt & Konfiguriert
7	Ablaufplanung	X		
5 +	Kommunikation und Applikationskontrolle	X	X	
6				
3 +	Applikationen	X		X
4				
4	Hardware			X

x Zugehöriges Kapitel ist Kapitel x

Abbildung 1.2: Das Infotainmentsystem als Erweiterung des FIS der Ascom im Zug: Beiträge der Arbeit auf den verschiedenen Systemebenen

## Kapitel 2

# Das Fahrgastinformationssystem

Die Umgebung in die das IS eingebettet werden wird, soll hier vorgestellt werden. Zur weiteren Umgebung gehört der Zug, zur direkten Umgebung das bestehende FIS der Ascom.

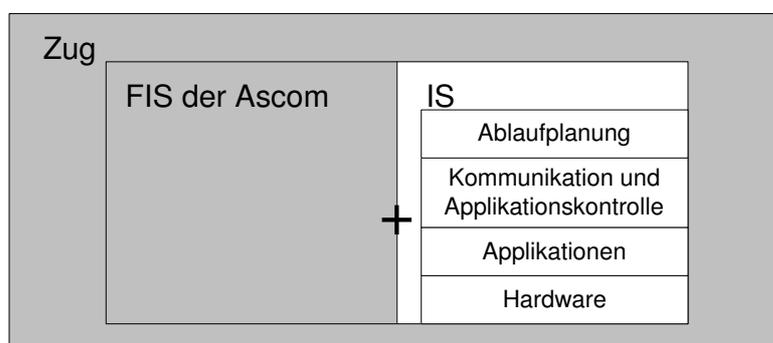


Abbildung 2.1: Betrachtete Systemebene: Zug und FIS der Ascom

Seit es öffentliche Verkehrsmittel wie Bahn, Bus und Flugzeug gibt, existieren auch Fahrgastinformationssysteme (FIS). Das FIS erfüllt seit jeher denselben Zweck: Jedem die für seine Reise relevanten Informationen zu liefern. Da aber ein Reisender, der seine Fahrt bereits angetreten hat, meist andere Informationen benötigt als einer, der seine Reise noch plant, wird zwischen mobilem und stationärem FIS unterschieden. Wenn als Beispiel der Zugverkehr genommen wird, bestand das mobile FIS aus einem oder mehreren Zugbegleitern und das Stationäre aus dem Bahnhofsvorsteher. Der Bahnhofsvorsteher musste zumindest über alle Züge, die an seinem Bahnhof hielten, Auskunft geben können, der Zugbegleiter zumindest über alle Stationen an denen der Zug hielt. Mit neuen technischen Errungenschaften wurde auch das FIS modernisiert und technisiert. Die Verbreitung von Fahrgastinformationen(FI) via Lautsprecher war der erste Schritt in Richtung Zentralisierung und Technisierung. Heute fährt in vielen Fahrzeugen kein Personal mehr mit. Das mobile FIS übernimmt auf Knopfdruck des Fahrzeugführers oder, mit Hilfe von Global Positioning System(GPS)Daten, auch automatisch das Ansagen und/oder das Anzeigen der nächsten Haltestelle. Das stationäre FIS besteht meist aus grossflächigen Anzeigetafeln, die über Ankunfts- und Abfahrtszeiten der Züge für einen Bahnhof informieren. Auch In vielen Fahrzeugen wurden bereits Anzeigen verschiedenster Art installiert. Die heute am häufigsten anzutreffende ist wohl die einzeilige Textanzeige. Der Trend geht heute aber in Richtung Thin Film Transistor(TFT) Anzeigen. Jeder führende Anbieter von FIS hat zumindest eine TFT Variante im Programm.

### 2.1 Kommunikation im Zug

Die Ascom ist ein führender Hersteller von mobilen FIS. Die von der Ascom verkauften Systeme werden vor allem in Zügen eingesetzt, weshalb hier auch der Zug als Einsatzgebiet des mobilen FIS näher betrachtet wird. Das System Zug, in welches das mobile FIS integriert werden

muss, ist sowohl technisch wie auch geometrisch sehr uneinheitlich. Begründet ist dies einerseits dadurch, dass die Züge aus Waggons unterschiedlichen Alters und von unterschiedlichen Herstellern zusammengestellt werden, aber andererseits auch dadurch, dass der Zugverkehr international geworden ist und Zugkompositionen mit Waggons aus anderen Ländern keine Seltenheit sind. Um ein mobiles FIS zum Einbau in Zügen erfolgreich verkaufen zu können, braucht es also nicht nur ein qualitativ hochwertiges, sondern auch ein anpassungsfähiges Produkt. Am Beispiel des Kommunikationsnetzes in Zügen soll diese Problematik hier verdeutlicht werden. Das Kommunikationsnetz ist für ein IS eine kritische Komponente, da über dieses Multimedia Daten, die Bandbreiten von bis zu 10Mbit/s benötigen, verteilt werden. Es hat deshalb einen starken Einfluss auf das Design des Infotainmentsystems.

Ein Kommunikationsnetz ist aber bereits für ein FIS eine wichtige Komponente, da die Passagiere zur rechten Zeit und am rechten Ort mit Informationen versorgt werden wollen. Das System muss also wissen, wann was angezeigt werden soll. Dazu benötigt es Signale aus dem System Zug. Diese Signale könnten z.B. durch das Drücken eines Knopfes im Führerstand ausgelöst werden oder auch von einem GPS stammen. Um mit diesen Signalen in allen Wagen die entsprechenden Ansagen und/oder Anzeigen auszulösen, muss ein Kommunikationsnetz innerhalb des Zuges zur Verfügung stehen. Die Nutzung des standardisierten und fast überall vorhandenen Train Communication Network(TCN) [6] bietet sich hierzu an. Um den Standard und dessen Einfluss auf das Design des mobilen FIS zu verstehen, ist ein historischer Rückblick auf dessen Entwicklung, sowie ein Blick auf die technischen Spezifikationen notwendig.

### 2.1.1 Der Train Communication Network(TCN) Standard

Vor 1980 kam es selten vor, dass einzelne Waggons der nationalen Bahnunternehmen über die Landesgrenze hinaus verkehrten. Es war daher für die Fahrzeugbauer legitim, proprietäre Kommunikationsschnittstellen einzusetzen, um die Kundenbindung zu erhöhen. Mit der Zunahme des internationalen Güter- und Personenverkehrs war aber das Bedürfnis nach einer flexiblen Zusammenstellung der Zugkompositionen auch aus Waggons aus anderen Ländern grösser geworden. Standardisierungsbestrebungen waren deshalb die logische Konsequenz. Reichte es früher noch, ein gemeinsames Kabel mit mehreren Leitungspaaren, eines für jeden Service (Licht, Türen u.s.w.), zu definieren, wurde um 1988 der Ruf nach einem effizienten, erweiterbaren und rekonfigurierbaren digitalen Kommunikationsnetz laut. Für dieses sollte ein Set von Spezifikationen entwickelt werden, welches zukünftig die Kompatibilität von Subsystemen und Fahrzeugen sicherstellt. Die International Electrotechnical Commission (IEC) in Zusammenarbeit mit der Union International de Chemin de fer (UIC) begann deshalb 1988 mit der Entwicklung des TCN Standards. Der Standard wurde 1999 in den Status FDIS(First Draft International Standard) mit dem Namen IEC 61375-1 [6] erhoben. Er definiert die interne Struktur von Kommunikationsnetzen in Zügen, stellt den Datentransfer zwischen programmierbarem Equipment in verschiedenen Fahrzeugen sicher und definiert die zu verwendenden Schnittstellen zwischen den Netzwerken und dem verwendeten Equipment.

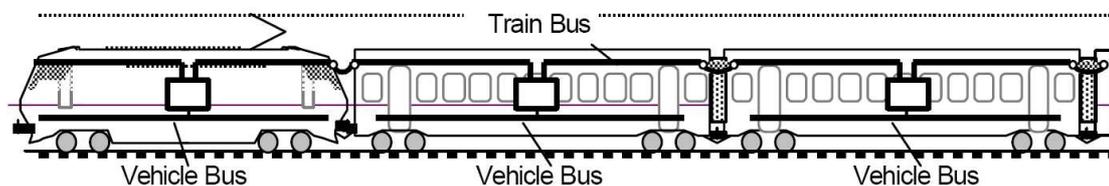


Abbildung 2.2: Train Communication Network (TCN)

Abbildung 2.2 zeigt die zweistufige hierarchische Struktur des TCN. Sie besteht aus dem Zugbus (Train Bus), welcher auch als Wired Train Bus (WTB) bezeichnet wird, sowie dem Fahrzeug- oder Wagenbus (Vehicle Bus), welcher auch als Multifunctional Vehicle Bus (MVB) bezeichnet wird. Die physikalische Schnittstelle wird im UIC 558 Standard [5] durch die Definition des zu verwendenden Kabels für den WTB festgelegt. Über den Zugbus ist die Kommunikation über mehrere Wagen möglich, über den Wagenbus die wageninterne Kommunikation. In jedem Wagen stellt ein Gateway die Verbindung zwischen Wagenbus und Zugbus sicher. Der Zugbus ist

optimiert für die Kommunikation über lange Distanzen (bis zu 860m) und liefert bei bis zu 32 anschließbaren Knoten eine Datenrate von max. 1Mbit/s zwischen zwei Gateways. Auf dem Zugbus gibt es einen Master, der den Zugriff steuert. Der Zugbus kann sich selbst konfigurieren, d.h. er kann z.B. Adressen an neu hinzugefügte Geräte verteilen. Der Wagenbus MVB ist optimiert für den Zugriff von vielen Geräten (bis mehrere hundert Geräte) und ist für eine maximale Länge von 200m bei max. 1.5 Mbit/s ausgelegt. Aber was wird eigentlich über das TCN versendet? Es gibt zwei Kategorien von Daten. Prozessdaten sind kurze, periodische Meldungen innerhalb vorgegebener zeitlicher Rahmen. Die Wiederholrate liegt zwischen 100ms und mehreren Sekunden. Nachrichten Daten sind ereignisbasierte Daten ohne zeitliche Rahmenbedingungen. Grosse Datenpakete werden in mehrere kleine unterteilt und jeweils einzeln gesendet wenn der Master den Bus dafür freigibt. Das TCN transportiert die Daten von der Quelle zum Ziel unabhängig vom Inhalt. Das Format der transportierten Daten wird für gebräuchliche Zugapplikationen im UIC Leaflet 556 [5] beschrieben. Die Nutzung des TCN zur Kommunikation im Zug beeinflusst massgeblich das Design der im Zug verwendeten Geräte. Die kleine Bandbreite des TCN ist unter anderem auf die Schwierigkeit zurückzuführen, günstige und störungssichere Steckverbindungen für high speed Datenverbindungen über die Wagenkupplungen hinweg zu finden. Wegen der geringen Datenrate kann das TCN nicht zum Transport von Inhalten für das IS genutzt werden. Da es im Moment und in naher Zukunft keinen neuen Standard zur Kommunikation im Zug geben wird, müssen die Daten für das FIS oder IS in jedem Waggon vorhanden sein. Diese Einschränkung wiegt weniger schwer, wenn die Entwicklung im Fahrzeugbau betrachtet wird. Diese geht hin zur fixen Zugkomposition, wie z.B. dem ICE oder ICN, deren Waggons nicht mehr mit beliebigen anderen Waggons kombiniert werden können. Sie besitzen meist spezielle oder gar keine Kupplungen. Der ganze Zug könnte dann als Waggon im Sinne des FIS der Ascom betrachtet werden.

## 2.2 Das mobile Fahrgastinformationssystem der Ascom

Da das IS auf dem mobilen FIS der Ascom basieren soll, wird es hier kurz vorgestellt. Das mobile FIS der Ascom wird in verschiedenen Ausführungen geliefert. So können die Passagiere z.B. über Lautsprecher oder über die unterschiedlichsten Anzeigentypen im und ausserhalb des Zuges informiert werden oder an verschiedene zusätzliche Schnittstellen angebunden werden. Diese Arbeit befasst sich nur mit der Variante, die Panel PCs als Anzeigen verwendet.

Abbildung 2.3 zeigt eine Übersicht über das mobile FIS mit Panel PCs, wie es z.B. von den Schweizerischen Bundesbahnen (SBB) eingesetzt werden könnte. Es ist daraus ersichtlich, dass der mobile Teil des FIS aus mehreren Subsystemen besteht. Der statische Teil des FIS ist stark vereinfacht dargestellt. Er besteht aus einer Systemeinheit Backoffice und aus einer Systemeinheit Communication. Die Backoffice Einheit liefert Steuerdaten, macht Statusabfragen oder generiert und verschickt dynamische Fahrgastinformationen an das mobile FIS unter Verwendung der Communication Einheit. Die Communication Einheit stellt die Verbindung zu den Fahrzeugen via Wireless Local Area Network (WLAN) oder GSM/GPRS (Global System for Mobile communication/General Packet Radio Service) her. Es ist natürlich möglich, dass das Backoffice System von einem anderen Hersteller als der Ascom geliefert wird und eine Schnittstelle vereinbart werden muss.

Auch im mobilen Teil des FIS werden nur die grün unterlegten Komponenten RAILVOX, Panel PC, GSM, GPS(Global Positioning System) und WLAN Module von der Ascom geliefert. Gemäss internationalem Standard muss z.B. das in 2.1.1 beschriebene und in jedem Wagen vorhandene TCN bestehend aus Gateways, WTB und MVB zur Kommunikation innerhalb des Zuges genutzt werden. Dieser Standard hat dann auch in grossem Masse die Struktur des mobilen FIS der Ascom bestimmt. So ist z.B. mit einer Datenrate von max. 1 Mbit/s auf dem WTB<sup>1</sup> keine Struktur mit nur einem zentralen Rechner pro Zug denkbar, da zu viele Daten für die Ansteuerung der Anzeigen und der Lautsprecheranlagen über diesen Bus laufen müssten. Das System hat deshalb in jeder über eine Kupplung getrennten Wageneinheit mindestens eine RAILVOX.<sup>2</sup> Das RAILVOX ist für die Fahrgastinformation und die Anbindung weiterer Dienste, wie z.B. Videoüberwachung, in einer Wageneinheit zuständig. Es bindet Komponenten über

<sup>1</sup> Einzige Möglichkeit zur standardisierten Kommunikation mit allen Teilen des Zugs, siehe 2.1.1 für Details.

<sup>2</sup> Diese Einheit wird RAILVOX genannt, weil das System die Fahrgastinformation ursprünglich nur über Lautsprecher (Vox=lat. für Stimme, Sprache) in der Eisenbahn (rail= engl. für Eisenbahn, Schiene) ausgegeben hat.

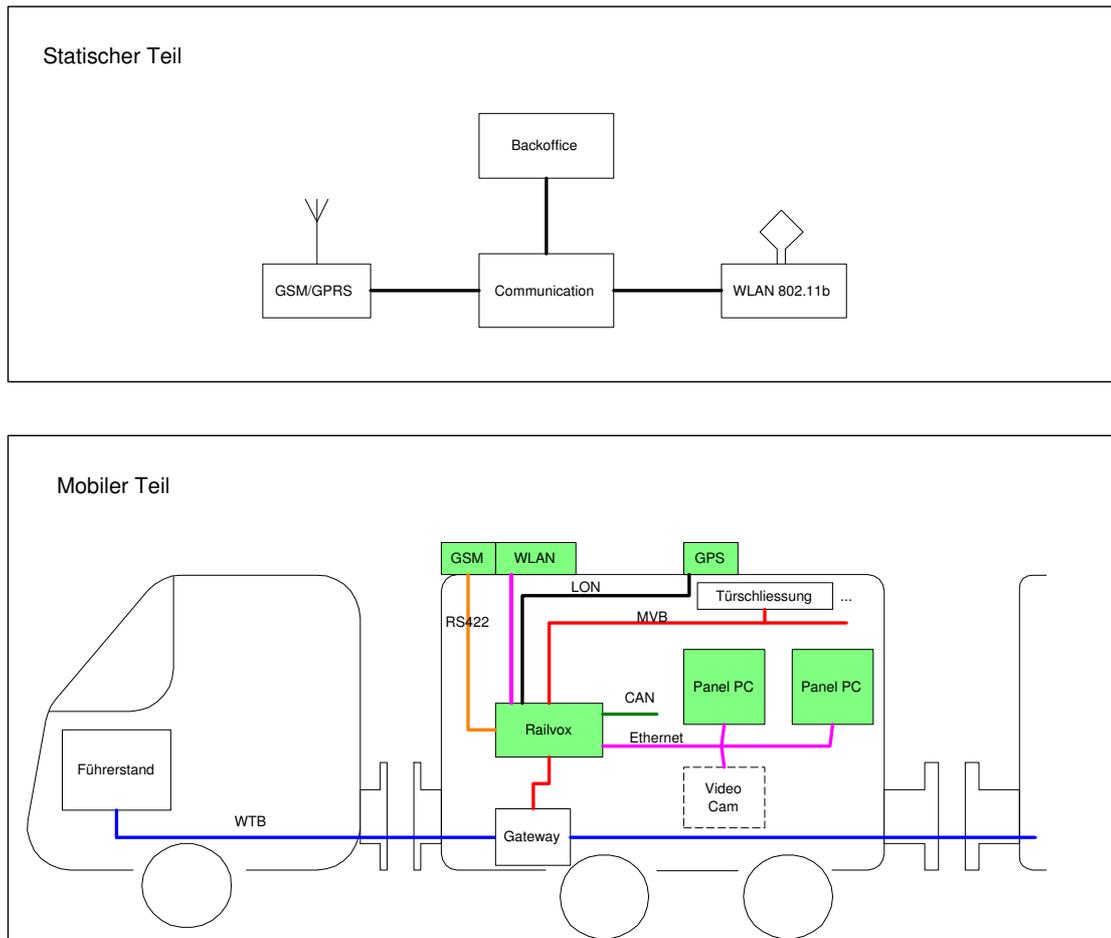


Abbildung 2.3: Systemübersicht mobiles FIS

verschiedene Schnittstellen an und kann via Gateway auch auf Daten vom WTB zugreifen. Dieser Lösungsansatz erlaubt eine vom WTB unabhängige und je nach verwendeter Schnittstelle vielfach schnellere Vernetzung des RAILVOX mit seiner Peripherie.

### 2.2.1 Das RAILVOX

Die heute aktuelle Variante des RAILVOX ist das RAILVOXplus. Das System besteht aus den in Abbildung 2.4 gezeigten Komponenten.

**RCM - RAILVOX Communication Module** Das RCM ist ein via Local Operating Network(LON) konfigurierbarer<sup>3</sup> 100/10 Mbit/s 5-Port Ethernet Switch. Er dient zum Anbinden von Geräten wie z.B. Videoüberwachungsanlagen. Die Verbindung REC-RCM erfolgt über 10Mbit/s Ethernet. Das RCM kann, um den Anschluss von mehr als fünf Geräten zu ermöglichen, kaskadiert werden. Ausserdem unterstützt es Layer 3 Operationen. Dies bedeutet, der Inhalt der Header der Pakete der Data Link Layer (Ethernet Paketformat) und der Network Layer (Internet Protokoll Paketformat) kann interpretiert und in bestimmten Grenzen auch beeinflusst werden. Eine Übersicht zum TCP/IP Netzwerk ist in Abbildung 3.3 im Abschnitt 3.2 zu finden.

**REC - RAILVOX Embedded Controller** Der REC ist der zentrale Steuerknoten des RAILVOXplus-Systems. Er besitzt einen PowerPC Prozessor und einen digitalen Signalprozessor. Zusammen mit den Modulen RIU und RKS resultiert bereits ein komplettes Fahrgastbeschallungssystem. Über LON oder die serielle Schnittstelle RS485 können diverse Anzeigen angesteuert werden. Die aktuelle Position wird mittels GPS oder Radumdrehungszähler ermittelt. Zur Kommunikation mit weiteren Komponenten oder RAILVOX Systemen in anderen Wagen stehen diverse Kommunikationsmodule wie das RVB, das RIO oder das RCM zur Verfügung. Mit entsprechenden Systemteilen für WLAN oder GSM/GPRS kann auch drahtlos kommuniziert werden. Das Softwaremodul Fahrgastinformationsserver ermöglicht es, bei einer bestimmten Position, digitale Ansagen automatisch abzuspielen, sowie Innenanzeigen und Zuglaufschilder anzusteuern. Das Softwaremodul läuft auf dem REC und basiert auf einer relationalen Datenbank mit Daten auf einer PC-Card. Eine Kartenkapazität von 1Gigabyte reicht für bis zu 3 Tagen Musik. Über die serielle Serviceschnittstelle RS232 oder Ethernet kann das System konfiguriert und überwacht werden. Über TCP-IP bieten die Anwendungen Web, FTP und Telnet ihre Dienste an. Mit dem GSM Modul kann ausserdem eine Ferndiagnose via Webbrowser, oder Telnet Client durchgeführt werden.

**RKS - RAILVOX Knotenspeisung** Ansagen die von einer Applikation auf dem REC ausgelöst werden, gelangen via RKS und Lautsprecher zu den Passagieren. Das RKS stellt die benötigte Leistung zur Ansteuerung der Lautsprecheranlage zur Verfügung und bietet Anschlüsse für Messmikrophone. Es kann die Lautstärke der Durchsage, basierend auf dem Geräuschpegel im Wagen, anpassen.

**Weggeber Interface** Dieses Modul liest die Daten vom Weggeber und stellt die Daten den anderen Modulen via MVB zur Verfügung. Der Weggeber ist ein Sensor, der bei jeder Radumdrehung ein Signal generiert.

**RAN - RAILVOX Analogknoten** An dieses Modul lassen sich eine Sprechstelle, mehrere Notrufsäulen, sowie die UIC Telefonleitung anschliessen. Die UIC Telefonleitung ist Teil des WTB und dient zur Kommunikation mit angeschlossenen Sprechstellen oder der Leitstelle(via Funkanbindung). Das Modul steuert die Kommunikation zwischen dem UIC Telefon und einer anderen Sprechstelle oder der Leitstelle.

<sup>3</sup>Die Software zur Konfiguration des Switches hat momentan nur einen eingeschränkten Funktionsumfang und muss für das IS erweitert werden. Mehr hierzu in Kapitel 3 und in Anhang A.1

**RIU - RAILVOX Interface für den UIC Bus** Es dient als Schnittstelle zwischen dem RAILVOX und dem Zugbus gemäss UIC Standard. Die Ansteuerung der 16 vom UIC 568 Standard definierten Leitungen erfolgt durch unterschiedliche Module des RAILVOXplus. Leitung 3 und 4 werden z.B. vom RAN Modul angesteuert.

**RVB - RAILVOX MVB Interface** Diese Schnittstelle erlaubt den Datenaustausch zwischen MVB Wagenrechner und RAILVOXplus. Der Multifunctional Vehicle Bus MVB ist, wie in 2.1.1 erläutert, Teil des TCN. Der MVB Wagenrechner stellt Informationen wie Zeit, Geschwindigkeit und Reservationsnummern zur Verfügung. Über den MVB kann das WTB Gateway erreicht werden (s. Abbildung 2.2) und somit über diese Schnittstelle auch mit RAILVOX Systemen in anderen Wagen kommuniziert werden.

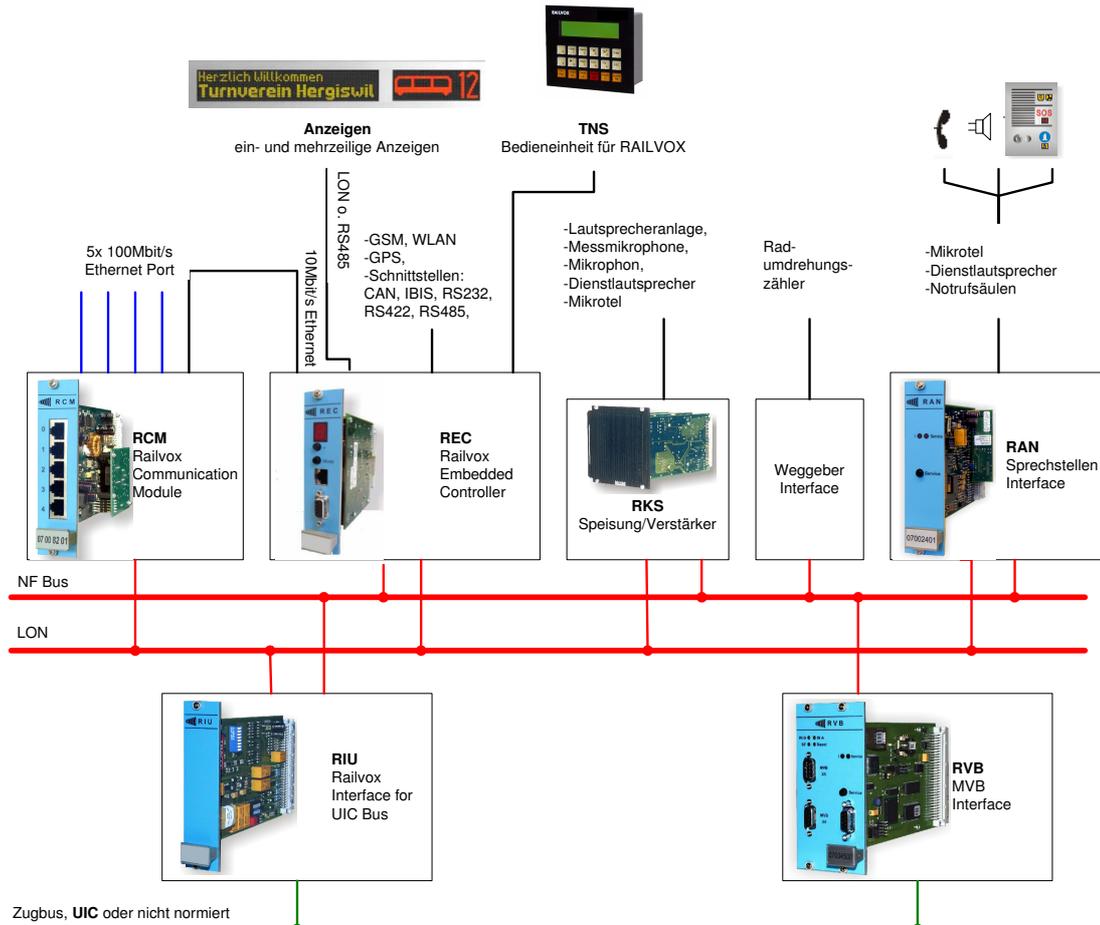


Abbildung 2.4: RAILVOXplus Systemübersicht

## Kapitel 3

# Übertragen und anzeigen der Infotainment-Daten

Hier sollen einige Konzepte für die Übertragung und Anzeige der Infotainment Inhalte vorgestellt, sowie die dazu benötigten Applikationen genannt werden. Es sei darauf hingewiesen, dass alle hier genannten Preise auf Listenpreisen vom Dezember 2004 basieren. Die Preise gelten für Stückzahlen kleiner als 10 Stück.

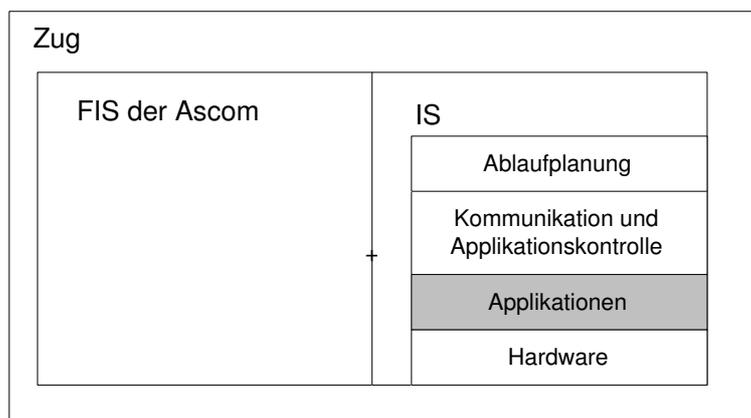


Abbildung 3.1: Betrachtete Systemebene: Applikationen

### 3.1 Systemstrukturen und Datenformate

Es existieren generell zwei unterschiedliche Ansätze für die Übertragung und Anzeige der Infotainment Inhalte. Der eine Ansatz geht von einer zentralen Generierung des Grafiksignals für die Anzeigen aus. Das Grafiksignal wird anschliessend über ein Kommunikationsnetz an TFT(Thin-Film Transistor) Bildschirmen mit Video Graphics Array(VGA), Digital Video Interface(DVI) oder Low Voltage Differential Signaling(LVDS) Eingang verteilt. Die Anzeigen selbst besitzen höchstens einen Konverter, welcher das empfangene Signal für den verwendeten Anzeigenanschluss<sup>1</sup> umsetzt. Die andere Variante ist, in alle Anzeigen einen Rechner zu integrieren und von einer ausgezeichneten Anzeige oder einem speziellen Server aus die Infotainment Inhalte z.B. als MPEG oder HTML(Hypertext Markup Language) Dateien zu verteilen. Das Grafiksignal für die Anzeigen wird dann lokal generiert. Dies entspricht der für das FIS der Ascom bereits verwendeten dezentralen Struktur.

Im Abschnitt 3.2 werden verschiedene Konzepte zum dezentralen Ansatz vorgestellt. Der Abschnitt 3.3 tut dies für den Ansatz mit zentraler Generierung des Grafiksignals. Schliesslich

<sup>1</sup>VGA, DVI oder LVDS

werden im Abschnitt 3.4 die beiden Ansätze verglichen, und es wird aufgezeigt, dass der bereits vom FIS der Ascom verwendete dezentrale Ansatz einige entscheidende Vorteile besitzt. Allen Konzepten ist gemeinsam, dass die Infotainment Inhalte aus Kostengründen auf einem Linux System abgespielt werden sollen. Um später Abschätzungen über die Eignung von diversen Konzepten vornehmen zu können, ist es notwendig, einige Kennzahlen für die vom IS unterstützten Datenformate zu nennen.

### 3.1.1 Kennzahlen zu den unterstützten Datenformaten

Gemäss der Aufgabenstellung sollen folgende Datenformate angezeigt werden können:

- Videos im Format MPEG-1 und MPEG-2
- Macromedia Flash<sup>2</sup> Animationen
- JPEG und GIF Bilder

Eine Abschätzung der Dateigrösse oder Datenrate der verschiedenen Formate führt zu den in Tabelle 3.1 aufgelisteten Werten.

Format	Auflösung	Grösse / Datenrate
MPEG-1	352x288 (ca. VHS Qualität)	1150 kbit/s (Nur Video. Audio: 224 kbit/s)
MPEG-2	480x576 (ca. TV Qualität)	2600 kbit/s (Nur Video)
MPEG-2	720x576 (DVD Format)	bis zu 9.8 Mbit/s (Nur Video. Audio: 32-1536 kbit/s)
MM Flash	1024x768	xxx-xxx MByte <sup>a</sup>
JPEG	1024x768, maximale Qualität	ca. 530 kByte
JPEG	1024x768, minimale Qualität	ca. 14 kByte
GIF	1024x768, 256 Farben	ca. 388 kByte

<sup>a</sup>Sehr stark vom Animationsgrad abhängig

Tabelle 3.1: Abschätzung der Dateigrösse oder Datenrate verschiedener Datenformate

### 3.1.2 Fehlende Unterstützung für Power Point Präsentationen

Ursprünglich war vorgesehen, dass auf dem IS auch Power Point Präsentationen gezeigt werden können. Schliesslich wurde aber davon abgesehen, obwohl das Anzeigen von Formaten, die aus der Windows Welt kommen, mit Software wie CrossOver Office [36] von Codeweavers oder OOImpress [35] von OpenOffice.org unter Linux möglich ist. OOImpress ist Bestandteil von OpenOffice und verlangt zumindest eine Teilinstallation dieses Pakets. Dem Abschnitt 4.1 über die eingesetzte Hardware kann entnommen werden, dass die zur Speicherung der Software eingesetzten CompactFlash Karte eine Grösse von 256 oder 512 MBytes hat. Eine Installation dieses Paketes würde also bereits einen grösseren Teil des verfügbaren Speichers belegen. Auch das CrossOver Office, mit dem Plug-Ins für Windows auf Linux installiert werden können, benötigt ca. 50 MBytes an Speicherplatz. Hinzu kommt noch die Installation des jeweiligen Windows Plug-Ins sowie aller benötigten Bibliotheken. Ein weiteres Argument gegen den Einsatz von CrossOver Office ist, dass 100 Lizenzen 1950 US\$ kosten.<sup>3</sup> Da ausserdem Power Point Präsentationen kaum von Werbeagenturen zur Werbung verwendet werden, wurde von einer Unterstützung dieses Formates abgesehen.

## 3.2 Der dezentrale Ansatz

Die Konzepte für den dezentralen Ansatz basieren auf der Struktur des FIS der Ascom. Wie in Abschnitt 2.2 vorgestellt, verwendet dieses zur Anzeige von FI mehrere mit Ethernet verbundene Panel PCs. Damit das FIS der Ascom zum Infotainmentsystem wird, müssen die Panel PCs

<sup>2</sup>Verbreitetes, relativ kompaktes Format für Animationen oder Webseiten [29]

<sup>3</sup>Stand 29.11.2004

neben FI auch Bilder, Animationen und Videos wiedergeben können. Um dies zu erreichen, erfordert es Lösungen für die folgenden beiden Problembereiche:

- Die Platzierung der abzuspielenden Daten auf den Panel PCs
- Die Wiedergabe von Videos und Animationen auf den Panel PCs

Die beiden Problembereiche sind leider gekoppelt. Will man z.B. Videos im Streamingverfahren an die Panel PCs übertragen, muss eine entsprechende Software zur Wiedergabe dieser Streams vorhanden sein. Zuerst müssten also Lösungsvorschläge für die Verteilung der Daten an die Panel PCs entwickelt werden, um anschliessend zu prüfen, ob es dazu auch passende Lösungen zum Abspielen der Daten gibt. Beim Abspielen ist zu beachten, dass alle Anzeigen möglichst synchron sein sollten.

Für eine erste Analyse der Konzepte wird ein idealisiertes Netzwerk angenommen, d.h. der Datenverkehr besteht nur aus den zu verteilenden Infotainment Inhalten. Diese werden zudem immer mit der maximal möglichen Geschwindigkeit übertragen. Es steht also die volle Bandbreite des jeweiligen Links zur Datenübertragung zur Verfügung. Mit dieser Annahme und den Zahlen zu den Datenformaten aus Tabelle 3.1 kann für das jeweilige Konzept abgeschätzt werden, ob es überhaupt brauchbar ist. Abbildung 3.2 zeigt eine Übersicht der vorgestellten Konzepte. Nach der Betrachtung für ein idealisiertes Netzwerk wird in Abschnitt 3.2.4 die Problematik im realen Netzwerk<sup>4</sup> im Zug erläutert. Ausserdem werden Vorschläge zur Entschärfung einiger Probleme gemacht.

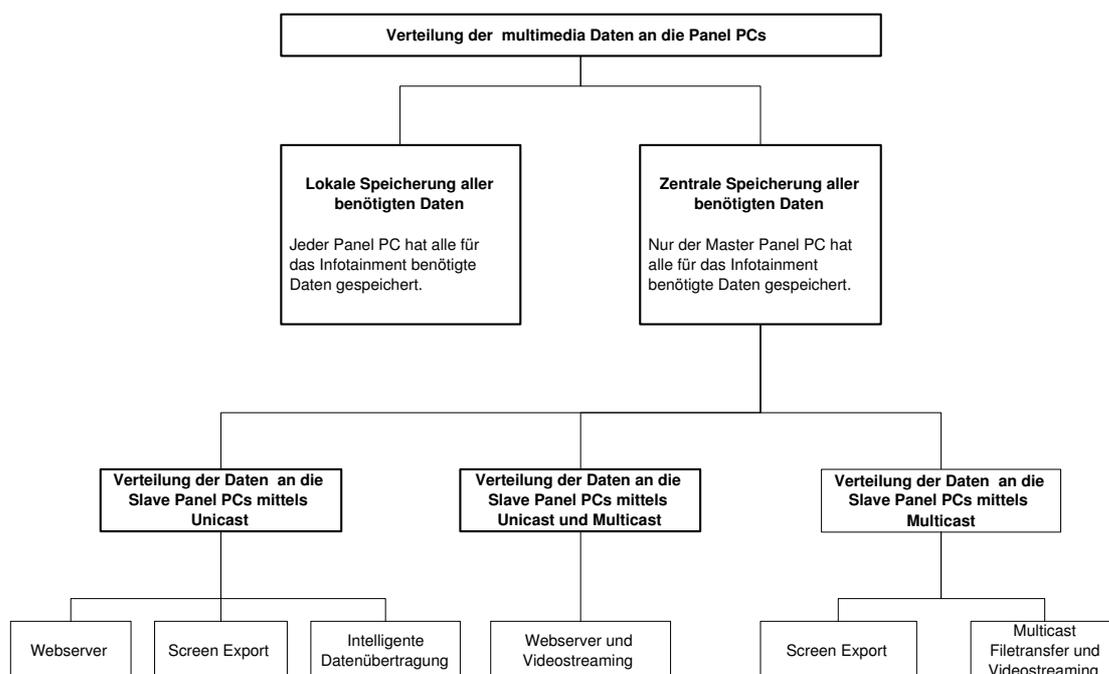


Abbildung 3.2: Übersicht über die vorgestellten Konzepte zur Datenverteilung

Abbildung 3.2 zeigt eine Übersicht der angedachten Konzepte. Die lokale Speicherung aller benötigten Daten auf jedem Panel PC wird von Vorneherein ausgeschlossen. Der Grund für den Ausschluss dieser Variante ist der Mangel an Speicherplatz<sup>5</sup> auf den Panel PCs. Ein Speicher- ausbau, der den Anforderungen für den Einsatz in Zügen genügt, kommt nicht in Frage, da dies das System zu stark verteuern würde. Abschnitt 4.1.1 nennt Beispiele solcher Speichererweiterungen im Zusammenhang mit der Speichererweiterung des Master Panel PCs. Die restlichen Konzepte gehen von einem als Master ausgezeichneten Panel PC aus. Dieser verteilt die Daten

<sup>4</sup>Im realen Netz werden noch andere Daten transportiert als die vom IS Generierten, und der Datentransport erfolgt nicht mit der theoretischen maximalen Bandbreite von 100MBit/s.

<sup>5</sup>Abschnitt 4.1 liefert Angaben zum verwendeten Speicher.

an die mit Slave bezeichneten Panel PCs. Wie ebenfalls aus Abbildung 3.2 hervorgeht, wird bei diesem Ansatz noch die Art der Datenverteilung unterschieden. Die betrachteten Möglichkeiten sind Unicast<sup>6</sup>, Unicast und Multicast<sup>7</sup> oder reines Multicast. Für alle vorgestellten Konzepte wird eine kurze Beschreibung gegeben. Anschliessend werden sie in einer Vergleichstabelle einander gegenübergestellt. Die Vergleichstabelle beinhaltet einige Angaben zur benötigten Softwareausstattung, eine Latenzschätzung und einige Vorteile und Nachteile des jeweiligen Konzeptes. Die Latenzschätzung betrifft die Zeit, die es braucht, bis alle  $N$  Slave Panel PCs die Datenmenge  $Data$  oder  $FrameData$  bei einer Bandbreite von  $LinkBW$  pro Link bei Einsatz von  $M$  Switches empfangen haben. Die verwendeten Parameter sind in Tabelle 3.2 zusammengefasst. Abbildung 3.4 gibt einen Überblick über die verwendete Systemstruktur und zeigt die Datenpfade für die jeweiligen Konzepte.

Anzahl Switches	$M$
Anzahl Slave Panel PCs	$N$
Bandbreite pro Link in Bytes/s	$LinkBW$
Menge der zu verteilenden Daten in Bytes	$Data$
Benötigte Datenmenge zur Darstellung eines Einzelbildes	$FrameData$

Bevor nun die einzelnen Konzepte für das Verteilen und Abspielen der Infotainment Inhalte im Detail erläutert werden, wird im Folgenden ein Überblick über das für die Datenverteilung zur Verfügung stehende Kommunikationsnetz gegeben.

**Die Ethernet und TCP/IP Architektur des RAILVOXplus** Das Kommunikationsnetz, das für die Vernetzung der Panel PCs und anderen Geräten wie z.B. Überwachungskameras oder WLAN Knoten verwendet wird, basiert auf der Ethernet Technologie. Eine übliche Netzwerkstruktur zeigt Abbildung 3.3. Es ist zu sehen, dass in diesem Beispiel der REC das WLAN Modul, sowie die automatische Fahrgastzählung(AFZ), an einem RCM hängen. Wie bereits erwähnt, erfüllt der Ethernet Port am REC nur den 10 Mbit/s Ethernet Standard. Der RCM, an dem das REC hängt, ist mit einem weiteren RCM verbunden, an dem Panel PCs und ein Videosystem zur Überwachung des Zuges hängen. Es können bei Bedarf weitere RCMs kaskadiert eingesetzt werden.

### 3.2.1 Konzepte basierend auf Unicast

Das Hauptargument für eine Verteilung der Infotainment Inhalte per Unicast ist, dass damit die Blockierung des RECs bei Einsatz von nicht konfigurierbaren Switches verhindert werden kann. Da nicht konfigurierbare Switches normalerweise alle an eine Multicast Adresse versendeten Pakete broadcasten, würde die 10MBit/s Verbindung zum REC bereits bei geringer Auslastung des restlichen Netzes(100MBit/s) stark belastet oder gar blockiert. Sind konfigurierbare Switches vorhanden, kann durch die Verwendung von Unicast eine ansonsten notwendige Konfiguration<sup>8</sup> vermieden werden.

Wie der Netzstruktur in Abbildung 3.3 zu entnehmen ist, hängen die Panel PCs jeweils am RCM. Will nun der Master die Daten an die Slaves verteilen, so muss man bei Verwendung von Unicast, die Daten bei  $N$  vorhandenen Slaves auch  $N$  mal versenden.<sup>9</sup> Die folgenden drei auf Unicast basierenden Varianten wurden untersucht: Verwendung eines Webservers, Übertragung des Bildschirminhaltes des Masters(Screen Export) und die Verteilung der Infotainment Inhalte mittels eines speziellen Algorithmus(intelligente Datenverteilung).

<sup>6</sup>Unicast: Übermittlung eines IP-Pakets an genau einen Empfänger

<sup>7</sup>Multicast: Versenden eines Datenpakets zum Empfang durch viele Empfänger. Dafür wird das Paket aber nur einmal und an eine einzige IP Adresse geschickt. Es ist also eine one-to-many - Kommunikation, bei der Router oder Switches die Vervielfältigung und Verteilung des gesendeten Pakets an bestimmte Empfänger, den Teilnehmer einer sogenannten Multicastgruppe, übernehmen. Die IPv4 Adresszuweisung für Multicast Pakete ist: Klasse D: 224.0.0.0 bis 239.255.255.255, für IPv6 sind dies die Adressen mit dem Präfix: 1111 1111(Binär)=FF (Hexadezimal).

<sup>8</sup>In jedem Switch müssten die zu einer Multicastgruppe gehörenden Teilnehmer definiert werden. Nur an diese würde dann Multicast Verkehr weitergeleitet und damit eine Überflutung des REC vermieden.

<sup>9</sup>Nur falls keine intelligente Verteilmethode benutzt wird. Siehe hierzu 3.2.1

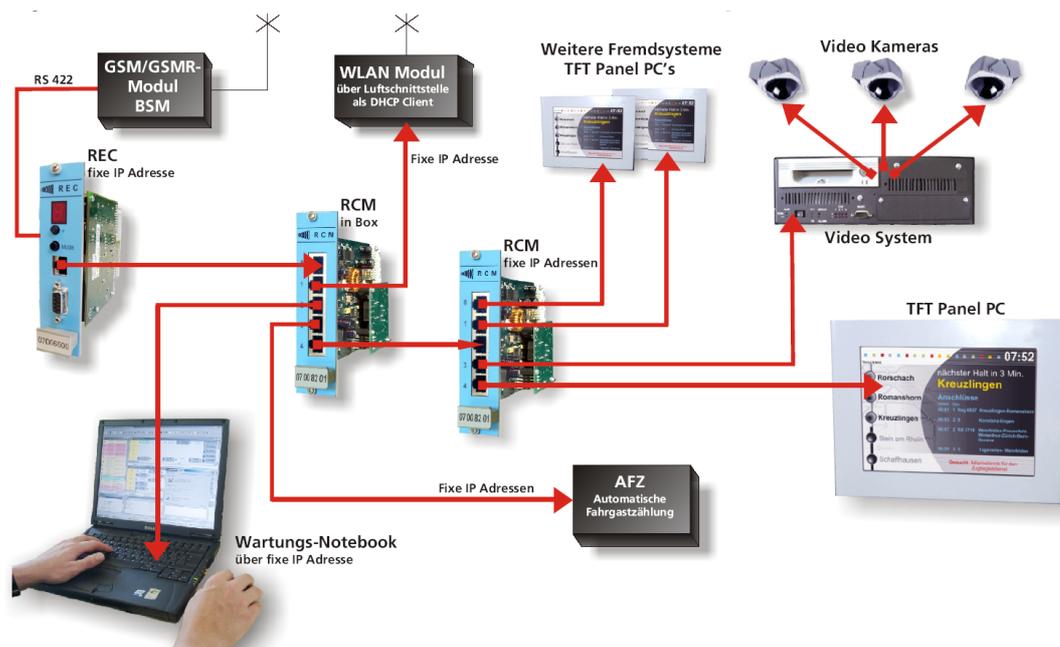


Abbildung 3.3: Ethernet und TCP/IP Architektur von RAILVOXplus

**Webserver** Auf dem Master Panel PC wird ein Webserver installiert. Die anzuzeigenden Inhalte werden in Webseiten integriert. Auf den Slave Panel PCs werden zur Anzeige ein Browser, sowie die entsprechenden Plug-Ins installiert. Die Slaves können auf diese Weise via Universal Resource Locator(URL) auf die auf dem Master gespeicherten Daten zugreifen. Abbildung 3.4 a) veranschaulicht dieses Verfahren.

**Screen Export** Auf dem Master Panel PC wird ein Virtual Network Computing(VNC) Server installiert. Dieser kann den Bildschirminhalt des Masters an VNC Clients senden. Auf dem Slave empfängt ein VNC Client den Bildschirminhalt und stellt ihn dar. Der VNC Server komprimiert den zu übertragenden Bildschirminhalt, um die zu übertragende Datenmenge zu reduzieren. Es gibt VNC Server, die hierzu verlustlose Kompressionsverfahren einsetzen, es gibt aber auch solche, die Verlustbehaftete verwenden. Eine Kompression ist dringend notwendig, wie folgende Beispielrechnung zeigt: Bei einer Auflösung von 1024x768 und 24 Bit Farbtiefe, sowie 25 Bilder pro Sekunde wäre ein Datenstrom von 59 MBytes/s zu übertragen. Wohlgedenkt, dies wäre die benötigte Bandbreite beim Anschluss von nur einem einzigen Slave Panel PC. Es ist ausserdem eher unwahrscheinlich, dass ein Panel PC genügend performant ist, 25 Bilder pro Sekunde zu komprimieren und zu verschicken.

**Intelligente Datenverteilung** Die abzuspielenden Daten werden an die Panel PCs z.B. unter Verwendung eines FTP Servers auf dem Master und Clients auf den Slaves übertragen. Dies setzt voraus, dass mindestens genügend Speicherplatz für die Daten auf den Slave Panel PCs vorhanden ist, um den gerade angezeigten Inhalt und den währenddessen übertragenen Inhalt zu speichern. Zusätzlich wird auf allen Panel PCs eine Software installiert, welche ein festgelegtes Verteilschema für die Daten umsetzt. Das Ziel dieses Konzepts ist es, die Latenz trotz Verwendung von Unicast möglichst gering zu halten. Bei den beiden vorderen Varianten wurde vom Master aus an jeden Slave Panel PC die Daten verteilt. Der dabei entstehende Engpass vom Master zum Switch kann vermieden werden, wenn der Master die Daten nicht an alle Slave Panel PCs verteilt, sondern nur an einige, und diese dann die Weiterverteilung übernehmen. Bei geeigneter Netzstruktur kann die Latenz bedeutend reduziert werden. Abbildung 3.5 illustriert den folgenden Verteilvorgang:

1. Master: Sendet die Daten an jeweils einen Panel PC pro Switch.

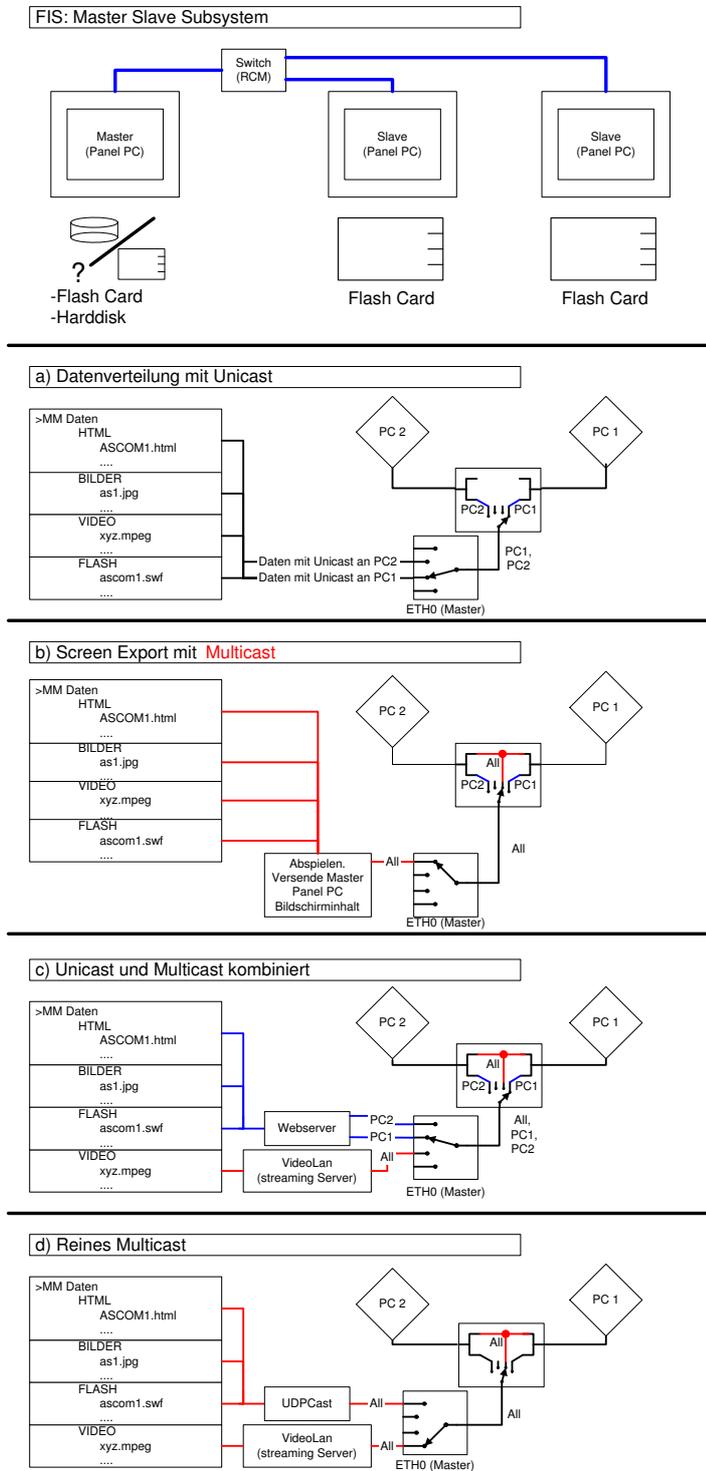


Abbildung 3.4: Systemstruktur und Datenpfade für verschiedene Konzepte

2. Slaves: Senden die vom Master erhaltenen Daten an X weitere Slaves am selben Switch wie sie selbst.
3. Slaves: Jeder mit Daten versorgte Slave, senden diese an einen Slave am selben Switch weiter.
4. Wiederhole Schritt 3 bis alle Slaves die Daten erhalten haben.

Abbildung 3.6 zeigt die benötigte Anzahl Iterationen, bis alle Slaves die zu verteilenden Daten erhalten haben, in Abhängigkeit der verwendeten Anzahl Slaves pro Switch und X. Dies für ein Beispiel mit 20 Slave Panel PCs. Abbildung 3.6 zeigt, dass bei optimaler Netzwerkstruktur die relative Latenz nur 1/4 der Variante mit dem Webserver beträgt.

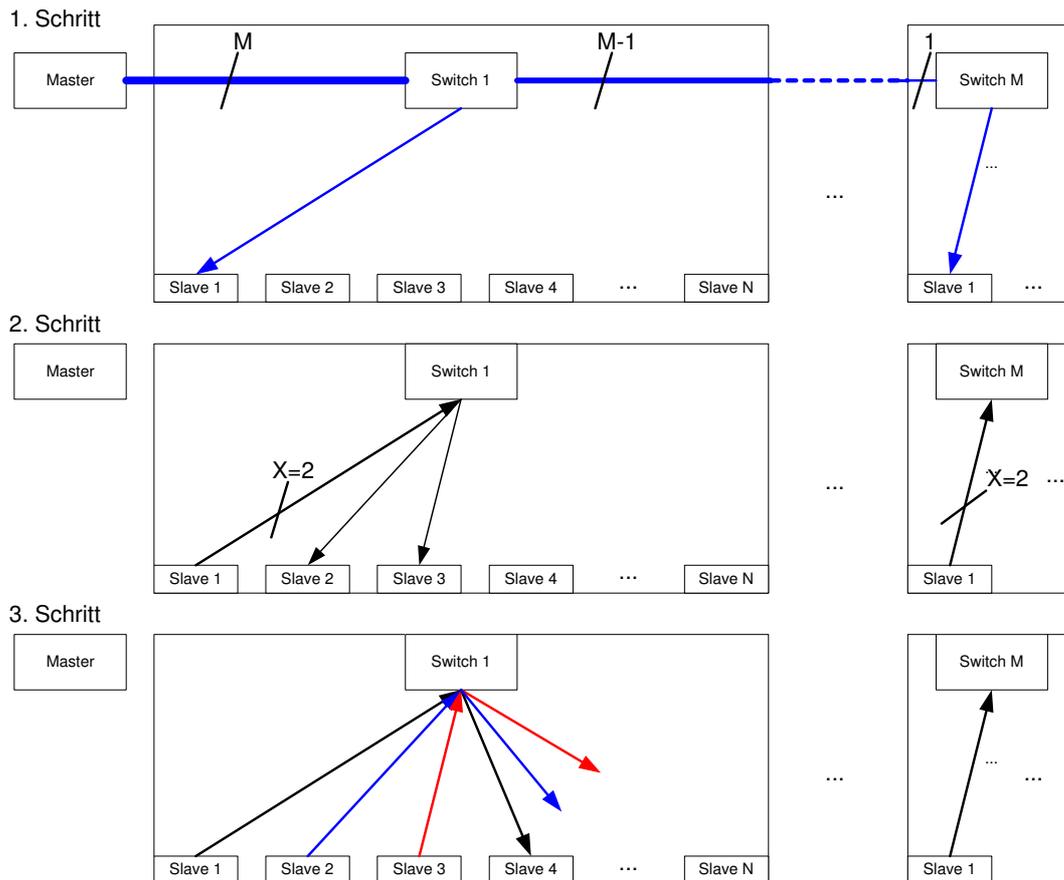


Abbildung 3.5: Beispiel zur intelligenten Datenverteilung

### 3.2.2 Konzepte basierend auf Unicast und Multicast

Falls in einem Zug Switches eingesetzt werden, die geeignet für den Einsatz mit Multicast konfiguriert werden können, steht der Verwendung von Multicast nichts im Wege. Eine Konfiguration der Switches ist aus dem in Abschnitt 3.2.1 genannten Grund notwendig. Ausgehend von der Beschreibung des RCM konnte von einem konfigurierbaren Switch ausgegangen werden. Eine genauere Untersuchung hat gezeigt, dass eine Konfiguration mit der aktuellen Konfigurationssoftware zwar möglich ist, aber das Eintragen eines Portes als Mitglied einer Multicastgruppe von dieser noch nicht unterstützt wird. Im Anhang A.1 wird Multicast mit dem RCM der Ascom beschrieben. Erläuterungen zu den vorzunehmenden Erweiterungen der Konfigurationssoftware sind ebenfalls dort zu finden.

Die Kombination von Unicast und Multicast ermöglicht es oft, eine einfachere Systemstruktur zu erreichen als mit einem rein auf Multicast basierendem System. Um das Kommunikationsnetz

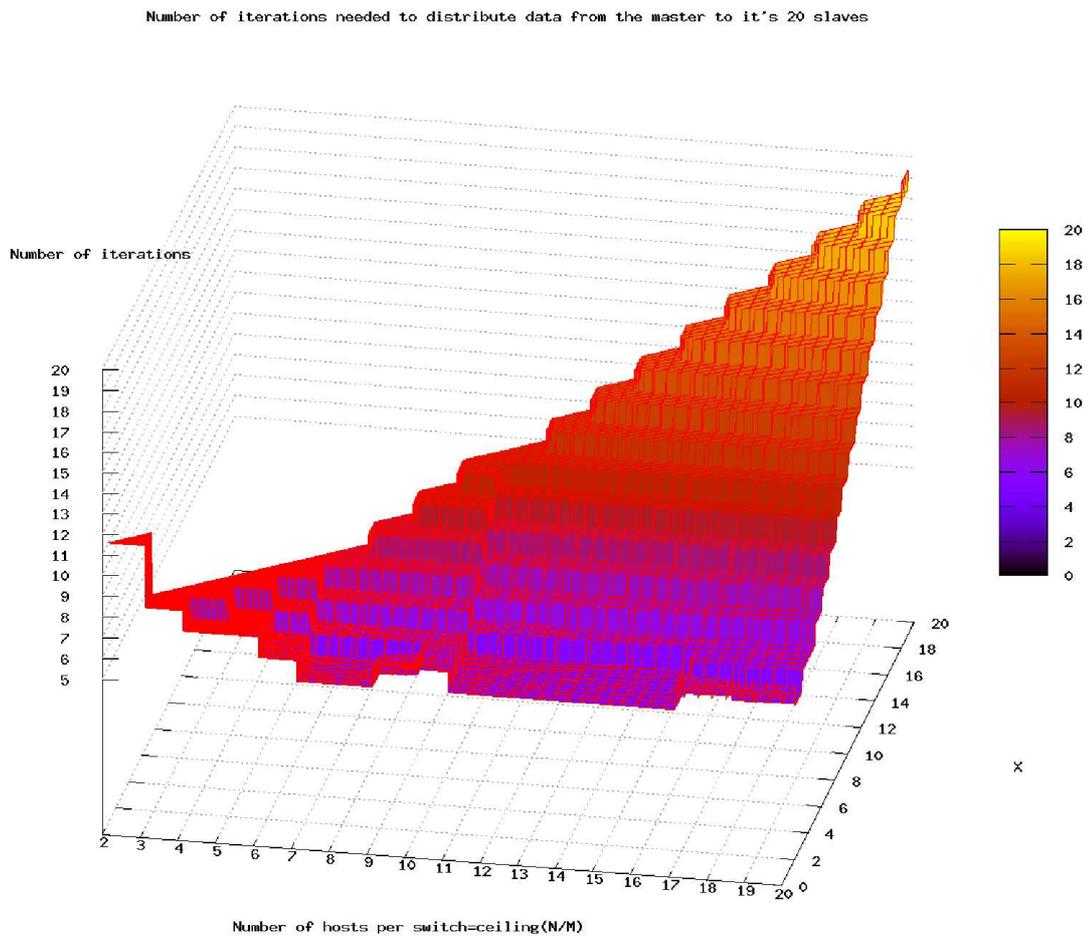


Abbildung 3.6: Relative Latenz für 20 Hosts

möglichst optimal zu nutzen, sollte der grösste Teil der Daten per Multicast übertragen werden. Dies ist z.B. der Fall, falls Multicast zum Streamen von Videodaten benutzt wird und einen Webserver die Webseiten zum Anzeigen dieser Streams bereitstellt.

**Videostreaming und Webserver** Auf dem Master Panel PC wird ein Webserver und ein Streamingserver für die Videodaten installiert. Die anzuzeigenden Inhalte (Bilder, Animationen oder Videos) werden im HTML Code einer Webseite referenziert. Die Referenzierung eines vom Master per Multicast versendeten Videostreams erfolgt z.B. durch das für das Streaming verwendete IP/Port Paar. Zur Anzeige der Webseiten muss ein Browser, sowie entweder ein eigener Player oder ein Plug-In zum Abspielen der in der Webseite referenzierten Streams vorhanden sein. Abbildung 3.4 c) zeigt den Datenpfad für diese Variante.

Die Schätzung der Latenz bezieht sich auf die Zeit, die benötigt wird, bis alle Panel PCs die HTML Daten vom Webserver erhalten haben. Hierbei bezieht sich *HTMLData* die Grösse der verwendeten HTML Datei auf dem Webserver. Ist eine Animation im Macromedia Flash Format oder ein Bild referenziert, beinhaltet *HTMLData* zusätzlich die entsprechenden Daten. Dies, da Macromedia Flash Animationen nicht per Multicast vom Webserver gestreamt werden können.

### 3.2.3 Konzepte basierend auf Multicast

Bei den vorangehenden Konzepten wurde zumindest teilweise immer auch Unicast zur Verteilung der Daten verwendet. Die hier vorgestellten Konzepte 'Screen Export' und 'Multicast' verwenden ausschliesslich Multicast. 'Screen Export' wurde bereits in einer Unicast Variante in Abschnitt 3.2.1 vorgestellt. 'Multicast' beruht auf dem Versenden von Dateien und Videostreams mittels Multicast.

**Screen Export** Auf dem Master Panel PC wird ein Multicast fähiger VNC Server und auf den Slaves werden die VNC Clients installiert. Es existiert hierzu eine Java basierte Software mit dem Namen MulticastVNC [39], sowie das daraus hervorgegangene TeleTeachingTool [41]. Beide wurden an der Universität von Trier entwickelt. Das Prinzip der Software ist, dass auf dem Master ein herkömmlicher VNC Server (z.B. RealVNC, TightVNC) die zu versendenden Daten generiert. Die Daten werden aber nun nicht per Unicast an die VNC Clients auf den Slaves versendet, sondern an den Multicast VNC Server. Dieser übernimmt dann die Verteilung der Daten per Multicast an die Multicast VNC Clients auf den Slaves. Wie beim Unicast Screen Export Konzept braucht der Master Panel PC nur Software zum Abspielen der Multimedia Daten. Abbildung 3.4 b) illustriert den Datenpfad dieses Konzepts. Es gelten hier dieselben Bedenken bezüglich der Performance, wie in Abschnitt 3.2.1. Die Erzeugung der Daten für die VNC Clients erfolgt ja durch denselben VNC Server wie bei der Unicast Screen Export Variante. Der Unterschied liegt nur in der effizienteren Nutzung der Bandbreite des Netzwerkes bei mehr als einem Slave Panel PC.

**Multicast** Alle Daten, ausser den Videodaten, werden mit Hilfe der Software UDPCast [40] per Multicast auf die Panel PCs verteilt. Auf den Panel PCs ist zur Speicherung der Daten ein Buffer notwendig. Dieser muss mindestens die gerade abgespielten Inhalte, sowie den nächsten anzuzeigenden Inhalt speichern können. Die Art und Grösse des Bufferspeichers, sowie der Bufferungsalgorithmus wird Thema in späteren Abschnitten sein.

Die Videodaten werden vom Master Panel PC unter Benützung der VideoLan [33] Software gestreamt. Wie bereits im Abschnitt 3.2.2 erwähnt, ist VideoLan Multicast fähig. Abbildung 3.4 d) illustriert den Datenpfad für dieses Konzept. Bei der Schätzung der Latenz werden nur Macromedia Flash Animationen, HTML Daten und Bilder, welche vor dem Anzeigen komplett auf die Slave Panel PCs übertragen werden müssen, berücksichtigt. Dies, da die Videodaten gestreamt werden, und es nur eine kleine Verzögerung vom Absenden des ersten Bildes durch den Master bis zur Anzeige dieses Bildes durch die Slaves gibt.

Verfahren	Benötigte Ausstattung	Latenzschätzung	Vorteile	Nachteile
Unicast: Webserver	Master Panel PC: Webserver (z.B. AppWeb [27])  Alle Panel PC: Browser (z.B. Mozilla [28]), Macromedia Flash Plug-In und MPlayer [30] zum Abspielen von Flash Animationen und Videos.	$Latenz = \frac{N \times Data}{LinkBW}$	Keine Konfiguration der Switches notwendig.	Skaliert nicht.  Bei voller Bandbreite (100MBit/s) sind bei einem MPEG-2 Video in DVD Qualität nur gerade 10 Panel PCs anschliessbar.
Unicast: Screen Export	Master Panel PC: VNC Server Software wie z.B. RealVNC [38] oder TightVNC [37]. MPlayer zum Abspielen von Videos, Macromedia Flash Player zum Abspielen von Macromedia Flash Animationen.  Slave Panel PC: VNC Client.	$Latenz = \frac{N \times VNCdata(FrameData)}{LinkBW}$ $VNCdata(x) = \text{Vom VNC Server generierte Datenmenge aus dem Inhalt } x \text{ des Bildschirms.}$	Keine Konfiguration der Switches notwendig.  Nur VNC Client Software nötig auf Slaves.  Master könnte auch ein Windows OS haben. Abspielen von Linux fremden Formaten wie Power Point ist kein Problem.	Skaliert nicht.  Performance Problem. Kompression und Versenden des Bildschirminhalts in Realtime notwendig (mind. 25 Bilder pro Sekunde)
Unicast: Intelligente Datenverteilung	Alle Panel PCs: FTP Server/Client zum Verteilen der Daten.  Software zur Realisierung des Verteilungsmusters. Browser (Mozilla) und Plug-Ins zum Abspielen von MPEG Videos (MPlayer) sowie Macromedia Flash Animationen.	$Latenz = (M + \lceil \log_{x+1} \frac{N}{M} \rceil + x) \frac{Data}{LinkBW}$	Keine Konfiguration der Switches notwendig.  Optimierung der Latenz für die Datenverteilung per Unicast	Skaliert schlecht.  Software zur Realisierung des Verteilungsmusters notwendig.  Temporäre Speicherung von Infotainment Inhalten notwendig. Benötigt viel Speicher auf den Slave Panel PCs
Multicast und Unicast: Videostreaming & Webserver	Master Panel PC: Webserver (z.B. AppWeb) sowie VideoLan [33] Software für das Multicast Streaming von Videos.  Slave Panel PC: Browser mit Macromedia Flash und VideoLan Plug-In.	$Latenz_{HTML} = \frac{N \times Data}{LinkBW}$ $Latenz_{Video} = \frac{FrameData}{LinkBW}$	Einfache Softwarestruktur.  Skaliert teilweise.  Kaum Speicher zum Zwischenspeichern von Infotainment Inhalten auf den Slave Panel PCs notwendig.	Andere Inhalte als Videos oder Audio nur via Unicast verteilbar (Webserver)  Konfiguration des Switches notwendig.
Multicast: Screen Export	Master Panel PC: VNC Server (z.B. RealVNC [38]) sowie MulticastVNC Erweiterung. MPlayer zum Abspielen von Videos, Macromedia Flash Player zum Abspielen von Macromedia Flash Animationen  Slave Panel PC: VNC Client	$Latenz = \frac{VNCdata(FrameData)}{LinkBW}$ $VNCdata(FrameData) = \text{Vom VNC Server generierte Datenmenge aus dem Inhalt } FrameData \text{ des Bildschirms.}$	Skaliert.  Master könnte auch ein Windows OS haben. Abspielen von Linux fremden Formaten wie Power Point ist kein Problem	Performance Problem. Kompression und Versenden des Bildschirminhalts in Realtime notwendig (mind. 25 Bilder pro Sekunde).
Multicast: Multicast	Auf allen Panel PC: UDPcast, Software zum Übertragen von Dateien mit Multicast. Browser und Plug-Ins zur Wiedergabe der Multimedia Daten.	$Latenz = \frac{Data}{BW\_Link}$	Skaliert.	Konfiguration der Switches notwendig.  Pufferspeicher bei Slave Panel PCs notwendig.

Tabelle 3.2: Vergleichstabelle für die verschiedenen Konzepte

### 3.2.4 Probleme und Lösungen beim realen Netzwerk

Für die Betrachtung des nicht idealisierten Netzwerkes wird die in Tabelle 3.3 angegebene Verkehrszusammensetzung angenommen. Diese Zusammensetzung beinhaltet bereits auch die erst in Zukunft über das Ethernet des IS kommunizierenden Dienste. Die Sprachverbindungen werden z.B. momentan noch über eigens hierfür vorgesehene Leitungen hergestellt. In Zukunft soll dies aber, unter Verwendung von Voice over IP (VoIP), über das Ethernet erledigt werden. Die Verkehrsarten aus Tabelle 3.3 haben folgende Quellen:

- Signalisierung: Master Panel PC, REC. Steuersignale an die Master/Slave Panel PCs.
- Sprachverbindungen: Sprechstellen im Fahrzeug. Insbesondere die Notfallsprechsäulen zur Kontaktaufnahme mit dem Zugpersonal oder der Zentrale.
- Videoüberwachung: Standbildabruf durch die Zentrale (via GSM/GPRS oder WLAN).
- SW/Daten Upgrade: Durch die Zentrale oder auch Datenabgleich zwischen Komponenten innerhalb des Zuges.
- Infotainment Daten: Master Panel PC sendet diese an die Slave Panel PCs.
- Zukünftige Dienste: z.B. interaktive Infoterminals.

Im nicht idealen Netz sei hier ausserdem eine effektiv nutzbare Bandbreite pro Link von 80MBit/s angenommen. Dies entspricht 80% der idealen Bandbreite von 100MBit/s .

Art der Daten	Datenmenge pro Übertragung /Bandbreitenbedarf	Häufigkeit
Signalisierung	<10kByte	periodisch
Sprachverbindungen	~64kbit/s pro Verbindung	unvorhersehbar
Videoüberwachung(Bildversand)	<1MByte	unvorhersehbar
SW/Daten-Upgrade	0 - X GBytes	unregelmässig, vorwiegend während Ruhezeit des Fahrzeuges
Infotainment Daten	0 - max. nutzbare Bandbreite	während Einsatzzeit des Fahrzeuges
Zukünftige Dienste	unbekannt	unbekannt

Tabelle 3.3: Art, Menge und Häufigkeit von Daten im Ethernet des IS

Jede der aufgelisteten Verkehrsarten stellt gewisse Anforderungen bezüglich minimaler durchschnittlicher Bandbreite, Verzögerungen und Reaktionszeit. Diese Kriterien verbindet man gemeinhin mit dem Begriff Quality of Service(QoS). Tabelle 3.4 hält die Anforderungen, sowie die Gründe dafür fest.

Um diese Anforderungen erfüllen zu können, ist die Priorisierung bestimmter Verkehrsarten unumgänglich. Folgendes Szenario soll dies verdeutlichen:

Zentrales initiiertes Datenbankupgrade: Die Daten werden zum im Bahnhof stehenden Zug via WLAN 802.11g (max. 54MBit/s) übertragen. Zur gleichen Zeit streamt der Master Panel PC ein Video in DVD Qualität. Hierfür werden bereits maximal 64MBit/s an Bandbreite belegt. Verschickt der Master Panel PC nun z.B. noch eine Macromedia Flash Datei für den nächsten Werbeclip an die Slaves, kann eine Beeinträchtigung des Videostreams oder von in dieser Zeit initiierten Notrufen nicht ausgeschlossen werden.

Die Anforderungen können auf verschiedene Weise erfüllt werden. Der für den Aufbau des Ethernets verwendete Switch(RCM) ist mit einem Marvell 88E6063 Chip versehen und bietet im Prinzip folgende Möglichkeiten:

1. Priorisierung anhand der Zieladresse.

Art der Daten	Anforderung	Begründung
Signalisierung	Auslieferung darf nur von Sprachverbindungen verzögert werden	Eine zur Steuerung der Anzeigen versendete Meldung sollte möglichst bei allen Anzeigen gleichzeitig ankommen, unabhängig vom Verkehr auf den einzelnen Netzwerkpfaden.
Sprachverbindungen	Muss immer ohne Qualitätsverlust oder Unterbruch zu Stande kommen.	Notrufe müssen in jedem Fall und ohne Unterbrechung übermittelt werden.
Videoüberwachung, Bildversand	Hohe Priorität. Darf nicht blockiert werden.	Bildübertragung während der Fahrt via GSM/GPRS oder ev. WLAN darf im Alarmfall nicht blockiert sein.
SW/Daten Upgrade	Niedrige Priorität, Verzögerung tolerierbar	Meist grössere Datenmengen. Abgleich oft während Ruhezeit des Fahrzeuges. Zeitkritische Daten Upgrades, wie z.B. Verspätungsmeldungen, erfolgen mittels Signalisierung.
Infotainment Daten	Unterbrechungs- und störungsfreie Wiedergabe. Ausnahmen: Notfallsituationen.	Die Passagiere und Anbieter erwarten eine störungs- und unterbrechungsfreie Präsentation des Infotainment.
Zukünftige Dienste	undefiniert	Unbekannter Einsatzzweck

Tabelle 3.4: Anforderungen der verschiedenen Verkehrsarten an das Kommunikationsnetz

2. Definition eines Virtual Local Area Networks(VLAN). Aufteilung des physikalischen Netzes in virtuelle Netze.
3. IEEE 802.3ac tagging. Anfügen/Entfernen oder unverändertes Durchreichen einer Prioritätsinformation durch den Switch abhängig vom Eingangsport.
4. Interpretation des ToS/DiffServ Feldes des IP Headers. Der Switch interpretiert diese Information standardmässig. In diesem Fall muss aber das korrekte Setzen dieses Feldes von den angeschlossenen Devices unterstützt, d.h. gegebenenfalls implementiert werden.

Die ersten drei Punkte erfordern jeweils eine Konfiguration der Switches. Bei Punkt 4 ist dies nicht notwendig. Die Punkte 1, 2 und 3 sind ohne Modifikation der Software zur Konfiguration des RCMs nicht verwendbar. Anhang A.2 spezifiziert die notwendigen Erweiterungen der Software für Punkt 1 und 3, Anhang A.3 spezifiziert sie für Punkt 2. Im Folgenden werden vier Varianten vorgestellt, welche die Anforderungen aus Tabelle 3.4 zumindest teilweise erfüllen.

**Priorisierung anhand der Zieladresse** Abbildung 3.7 b) illustriert, wie mit der Priorisierung anhand der Zieladresse, die Anforderungen aus Tabelle 3.4 erfüllt werden können. Der Switch führt bei dieser Art der Priorisierung eine Tabelle, die pro Zieladresse genau eine Prioritätsangabe enthält. Damit die Anforderungen aus Tabelle 3.4 erfüllt werden, müssen die IP für den REC, sowie die IPs der Sprechstellen die höchste Prioritätsstufe bekommen. Leider wird aber auf diese Weise der gesamte Verkehr zum REC mit Stufe 1 priorisiert und nicht nur der Verkehr, der von den Sprechstellen kommt. Das ist solange kein Problem, als nur sehr geringe Datenmengen zum REC fliessen. Bereits im Falle eines grösseren Updates ist das aber nicht mehr der Fall. Eine Lösung wäre hier, dem REC mehrere IPs mit Hilfe von virtuellen Interfaces zuzuweisen. Allerdings müssten dann die jeweiligen Geräte auch wissen, mit welcher IP des REC sie zu kommunizieren haben. IPs von Panel PCs werden in allen Switches mit der Prioritätsstufe 2 geführt. IPs von anderen Komponenten werden in der Tabelle mit der Prioritätsstufe 3 geführt.

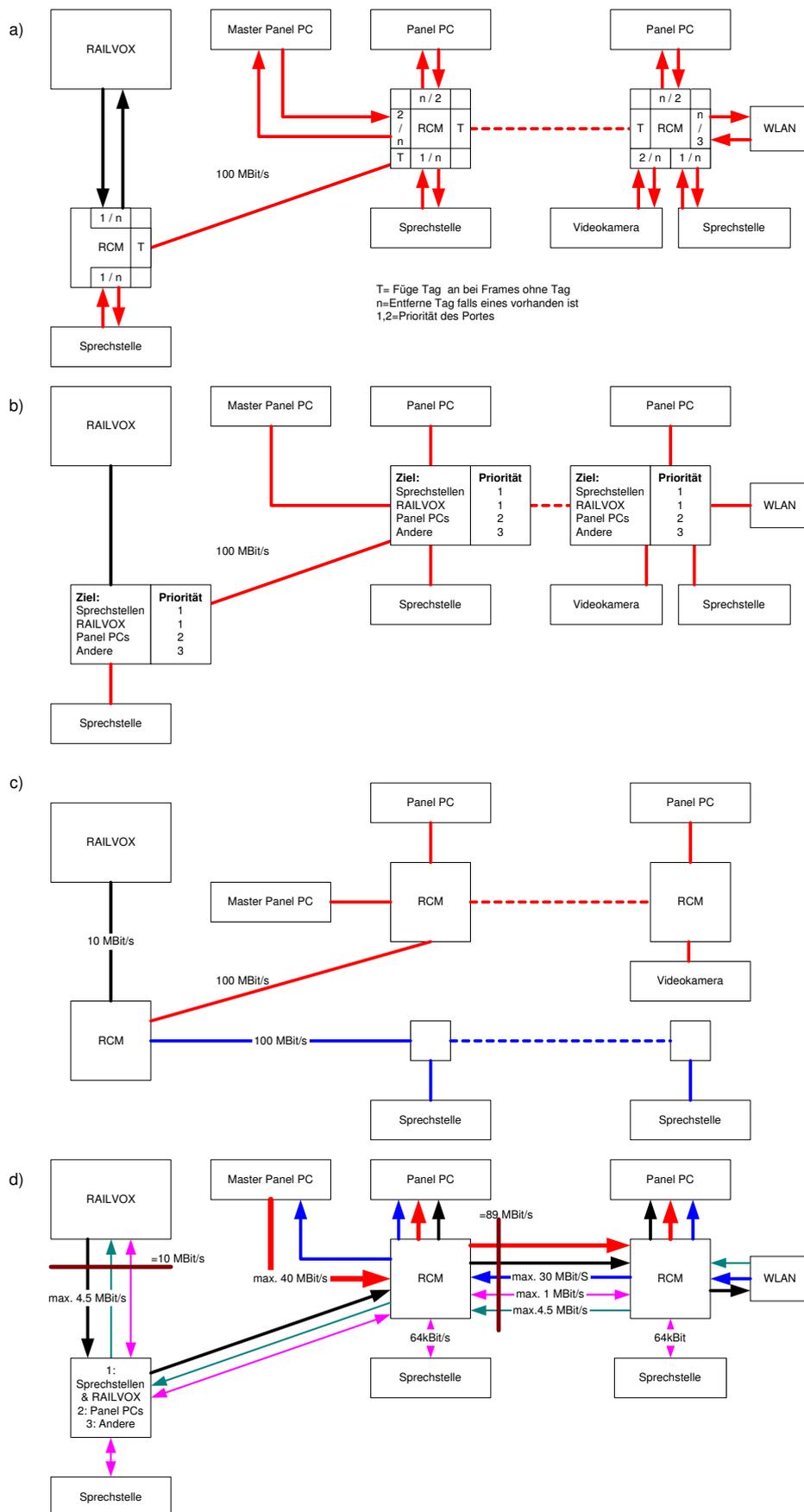


Abbildung 3.7: Lösungsmöglichkeiten zum Erfüllen der Anforderungen gemäss Tabelle 3.4

**Separate Netze durch VLAN** Die Aufspaltung des Netzwerkes in mehrere VLANs kann ebenfalls dazu beitragen, die Anforderungen aus Tabelle 3.4 zu erfüllen. Abbildung 3.7 c) zeigt ein Beispiel eines solchen Netzes. Dieser Ansatz beruht auf der bis auf einen gemeinsamen Punkt vollkommenen physikalischen Trennung des Netzes für die Sprache, sowie für die restlichen Dienste. Die physikalische Trennung ist vertretbar, da bis anhin sowieso für jede Sprechstelle eine eigene Leitung eingezogen werden musste. Eine Konfiguration des RCM ist somit nur beim Verbindungspunkt notwendig. Dort kann mit Hilfe der VLAN Einstellungen festgelegt werden, dass nur vom Port, an dem der REC hängt, Daten ins Sprachnetz gesendet werden dürfen. Daten vom Sprachnetz an den REC müssen zudem Priorität gegenüber Daten aus dem anderen Netz haben. Ob eine Priorisierung effektiv notwendig ist, kann aber nur mit Versuchen am System festgestellt werden. Das Erfüllen der Anforderungen für die restlichen Verkehrsarten erfolgt auf ähnliche Weise, wie bei den nicht physikalisch getrennten Netzen, aber ohne die Problematik der Priorisierung der Sprachdaten. Der Vorteil dieser Lösung ist, dass hier im Gegensatz zur Priorisierung anhand der Zieladresse keine virtuellen Interfaces benötigt werden.

**Priorisierung unter Verwendung von IEEE 802.3ac tagging** Der IEEE 802.3ac Standard definiert ein spezielles, gegenüber dem IEEE 802.3 Ethernet Standard, um 4 Oktetts<sup>10</sup> erweitertes Ethernetframe. Diese 4 Oktetts enthalten sowohl die Priorität, als auch die VLAN-Zugehörigkeit eines Frames. Ein konfigurierbarer Switch kann diese Information je nach Konfiguration den eintreffenden Frames selbständig hinzufügen. Ob überhaupt und welche Priorität und VLAN-Zugehörigkeit eingefügt werden soll, kann für jeden Port des Switches separat definiert werden. Das 802.3ac Tag kann vom Switch beim Weiterleiten auch wieder entfernt werden. Abbildung 3.7 a) illustriert die Verwendung von IEEE 802.3ac tagging für das gegebene Netzwerk. Alle Geräte tauschen Daten mit dem REC mit der für sie spezifizierten Priorität aus. Das REC selbst sendet immer mit der höchsten Priorität. Dies stört nicht, da der Ethernet Anschluss am REC mit max. 10 MBit/s senden kann und somit nur ein Zehntel der Bandbreite des restlichen Netzes beträgt. Der Verkehr, der von den Sprechstellen ausgeht, ist mit der höchstmöglichen Priorität versehen. Da der Master Panel PC Infotainment Daten versendet oder mit dem REC kommuniziert und dabei die Sprachverbindungen nicht behindern soll, werden sie mit der zweithöchsten Priorität(2) versehen. Alle anderen angehängten Geräte ausser der Videoüberwachung versenden Daten der Prioritätsstufe 3. Die Verwendung der Stufe 2 für die Videoüberwachung verhindert das Blockieren durch Infotainment Daten.

**Bandbreitenbegrenzung** Es ist auch möglich, ganz ohne Prioritäten auszukommen. Eine Begrenzung der Bandbreite der Datenströme macht dies bei entsprechender Ausrüstung aller angeschlossenen Komponenten möglich. Für ein System bestehend aus einem Multicast basierten Infotainmentsystem, einer Überwachungskamera, sowie vier Sprechstellen, könnte z.B. die in Abbildung 3.7 d) dargestellte Bandbreitenaufteilung verwendet werden. Diese, auf der statischen Vergabe von Bandbreiten pro Komponente basierte Variante, setzt voraus, dass die Vorgaben durch die Geräte selbst erfüllt werden. Der WLAN Knoten müsste dann im Beispiel den Datenstrom auf max. 30MBit/s begrenzen können. Eine Konsequenz dieses Ansatzes ist die ineffiziente Nutzung der Bandbreite des Kommunikationsnetzes. Durch die dynamische Vergabe von Bandbreiten könnte dies vermieden werden. Der hierfür zu betreibende Aufwand wäre aber im Gegensatz zu einer einmaligen Konfiguration der RCMs zur Realisierung der prioritätsbasierten Lösung unverhältnismässig.

## Fazit

Es kann folgendes festgestellt werden: Die erste Variante funktioniert nur solange, bis keine Quelle Pakete mit unterschiedlicher Priorität versenden muss. Die zweite Variante verursacht

<sup>10</sup>Aus: Wikipedia, der freien Enzyklopädie ([www.de.wikipedia.org](http://www.de.wikipedia.org)): *Ein Oktett (engl. octet) ist die Bezeichnung für ein Tupel (eine zusammengehörende Informationsmenge) von 8 Bit in der Informatik und Digital-Technik. Im Gegensatz zum Byte besteht ein Oktett stets aus 8 Bit und kann demnach  $2^8 = 256$  verschiedene Zustände darstellen. Da moderne Computer nahezu ausschließlich Bytegrößen von 8 Bit verwenden, werden die beiden Begriffe oftmals als Synonym verwendet. Als Maßeinheit zur Angabe von Speichergrößen dient aber die von Bit abgeleitete Einheit "Byte", wobei gilt: 1 Byte = 8 Bit.*

zusätzlichen Aufwand bei der Verkabelung und die Vierte eine Modifikation bei jedem angeschlossenen Gerät. Die Dritte Variante ist die flexibelste und verursacht, abgesehen von der Konfiguration des RCM, keinen zusätzlichen Aufwand. Sie sollte daher am ehesten in Betracht gezogen werden. Die Priorisierung wird für den Testaufbau nicht eingesetzt. Dies, da für den Testaufbau kein RCM als Switch verwendet wurde, und da das Testsystem nicht in ein Netzwerk mit REC und Sprechstellen integriert ist. Wie bereits erwähnt, müsste hierfür zuerst die Konfigurationssoftware für den RCM um die in Anhang A.2 und A.3 beschriebenen Funktionalität erweitert werden.

### 3.3 Der zentrale Ansatz

Die hier vorgestellten Konzepte verwenden alle einen zentralen Server, der die Infotainment Inhalte abspielt und das Grafiksignal für die Anzeigen generiert. Dieser Server könnte z.B. ein den Anforderungen für den industriellen Einsatz genügender SBC, in Form einer PCI oder CompactPCI Karte, für dein Einbau in ein 19" Rack sein. Die Kosten hierfür liegen bei ca. 1000 US\$. Die Daten am Grafikausgang des Servers werden dann über ein Netzwerk an alle TFT Bildschirme verteilt. Die erste Variante benutzt hierfür die Gigastar Digital Display Link Technologie, die Zweite benutzt Konverter und CAT5e Kabel. Die dritte Variante verwendet einen VGA Signal Vervielfacher, um die Bildschirme anzusteuern. Tabelle 3.5 vergleicht sowohl technische, wie auch wirtschaftliche Aspekte der drei Varianten bezüglich des in Abbildung 3.8 gezeigten Systemaufbaus. Die Kosten der für das Netzwerk benötigten Hardwarekomponenten, sowie die Gesamtkosten für das Beispiel in Abbildung 3.8 sind als Richtwert zu verstehen. Grosse Abweichungen bei der Wirtschaftlichkeit der Varianten sind bei grösseren Stückzahlen oder bei anderer Anzahl Anzeigen durchaus zu erwarten.

	Gigastar Digital Display Link	VGA über CAT5 Kabel	VGA Splitter
Hersteller	Innova-Technology	z.B. Minicom [17]	z.B. ByteRunner Technologies [16]
Max. Überbrückbare Distanz ohne Repeaterinsatz	bis 100 Meter	bis 110 Meter	ca. 65 Meter mit VGA-Koaxialkabel (zweifach geschirmt)
Max. Auflösung	1600x1200Hz	1600x1200@75Hz	2048X1536@60Hz
Temperaturbereich für vorgestellte Ausrüstung	-40-85 Grad Celsius	0-40 Grad Celsius	0-40 Grad Celsius
Verwendete Kabel	Kupfer oder Multimode Glasfaser	CAT5e Kabel <sup>11</sup>	VGA, Koaxial, zweifach geschirmt
Ungefähre Kosten für Kommunikationsnetz	1900US\$	1384 US\$	547 US\$
Verwendete Hardware Komponenten: Anschlussmöglichkeiten und Kosten			
Sender	2 Kanäle: 350US\$	1-Port: 345 US\$, 8-Port: 500 US\$	–
Empfänger Splitter	Für 2 TFT's: 400US\$ –	Dongle: 110 US\$ 8-Port: 316 US\$	– 8-Port: 100 US\$, 4-Port: 70 US\$
Kabel	Kupfer(Koaxial)30m: 75US\$	30m: 26 US\$ 7.5m: 8US\$	30m: 75 US\$, 7.5m: 21 US\$

Tabelle 3.5: Varianten für die Verteilung der Inhalte an die TFT Bildschirme

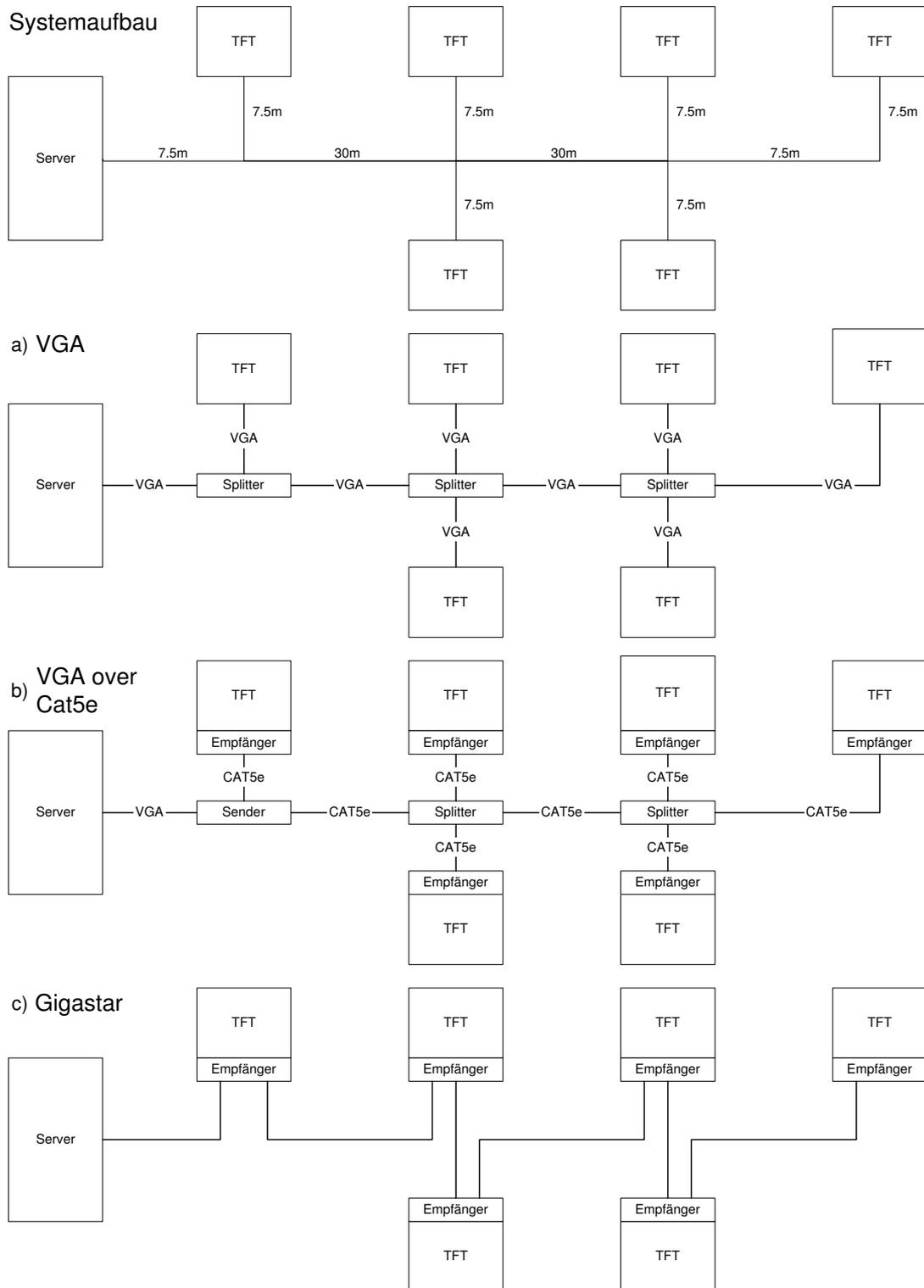


Abbildung 3.8: Beispielaufbau und drei Varianten zur Vernetzung

## 3.4 Design Entscheid: Variante Zentral oder Dezentral

Hier soll nun gezeigt werden, dass der vom FIS der Ascom bereits verwendete dezentrale Ansatz durchaus auch für das IS die bessere Wahl ist. Folgende Kriterien sollen für den Entscheid massgebend sein:

- Bildqualität
- Synchronisation der Anzeigen
- Rückkanal/Interaktivität
- Zur Datenübertragung verwendetes Netzwerk
- Fernsteuerung von Helligkeit/Kontrast der Anzeigen

### 3.4.1 Evaluation

Vor- und Nachteile der Variante Zentral und Dezentral sind im Folgenden aufgelistet.

#### Bildqualität

Zentral:

- - Signal vom Server generiert. Fix für alle am selben Netz angeschlossenen Anzeigen.
- - Anzeigen mit anderer nativen Auflösung müssen das Signal interpolieren.
- + GigaStar Digital Display Link besitzt 2 Kanäle. Zwei unterschiedlich grosse Anzeigen mit nativer Auflösung ansteuerbar.

Dezentral:

- + Lokale Generierung des Grafiksignals -> Auflösung an angeschlossene Anzeige anpassbar.
- + Skalierung von Inhalten erfolgt durch die CPU/GPU. Korrektur/Filterung möglich.
- + Optimal für ohne Qualitätsverlust skalierbare Grafiken wie Vektorgrafiken.<sup>12</sup>

#### Synchronisation der Anzeigen

Ein angezeigtes Bild eines Videos muss auf allen anderen Anzeigen innerhalb von maximal 40ms ebenfalls gezeigt werden. Bei grösseren Abweichungen könnte der Unterschied bereits wahrgenommen werden. Zentral:

- + Anzeigen garantiert synchron (Abweichung <40ms). Signallaufzeit nur von Verzögerung in Repeater, Splitter oder Empfänger abhängig. Diese liegt bei der betrachteten Hardware jeweils unter 1ms.

Dezentral:

- - IP Stack und anderer Verkehr auf dem Netzwerk kann Verzögerungen grösser als 40ms verursachen.
- - Einsatz von Priorisierung oder gar spezielle Massnahmen zur Synchronisation sind eventuell notwendig.

---

<sup>12</sup>Aus <http://www.galileocomputing.de/glossar/gp/anzeige-9022/FirstLetter-V>: Eine Vektorgrafik ist in einer Beschreibungssprache abgefasst, die eine Programmiersprache (wie Encapsulated PostScript(EPS) Format) oder eine Markup-Sprache (wie eXtensible Markup Language(XML) beim Scalable Vector Graphics(SVG)-Format) sein kann. Anders als eine Pixelgrafik besitzt eine Vektorgrafik keine feste Anzahl Bildpunkte(Auflösung) mit zugehöriger Farbinformation; sie besteht aus verschiedenen Elementen, wie Kurven, Geraden, gefüllten Flächen oder Text in skalierbarer Schrift, die durch eine Art Interpreter 'gerastert' werden müssen. Dieses Interpretieren der Vektorgrafik in ein Pixelbild kann bei jeder Auflösung und jeder Bildgröße geschehen; Vektorgrafiken lassen sich also verlustfrei skalieren. Was hierbei zur erheblichen Verwirrung vieler beiträgt, ist, dass eine Vektorgrafik auch Pixelbilder enthalten kann, beispielsweise als Textur ausgefüllter Flächen. Bei diesen Elementen können unter Umständen dann doch Skalierungsprobleme auftreten.

**Rückkanal/Interaktivität**

Zentral:

- - Kein Rückkanal vorhanden oder Interaktivität nur mit spezieller Software zu realisieren.
- - Aber: Alle Anzeigen zeigen den gleichen Inhalt.<sup>13</sup>
- + Bei GigaStar ist ein einziges, abgesetztes interaktives Terminal denkbar (2 Kanäle).

Dezentral:

- + Einfache Realisierung. Z.B. Webserver auf dem Master Panel PC. Browsen auf den Slave Panel PCs.
- + Jeder Panel PC kann Inhalte unabhängig von den übers Netz verteilten Inhalt anzeigen.
- + Jede Anzeige via IP gezielt adressierbar.

**Zur Datenübertragung verwendetes Netzwerk**

Zentral:

- + Netzwerk nur von Anzeigen benutzt. Volle Bandbreite steht zur Verfügung.
- - Separates, ansonsten nicht benötigtes Netzwerk muss in die Waggons eingebaut werden.

Dezentral:

- + Nutzt das sowieso zur Vernetzung von Geräten wie Videoüberwachung oder WLAN Access Point installierte Ethernet.
- - Muss Bandbreite mit anderen Geräten teilen.

**Fernsteuerung von Helligkeit/Kontrast der Anzeigen**

Zentral:

- - Falls überhaupt vorgesehen, nur mit zusätzlicher Verkabelung zu erreichen.
- + Bei Verwendung des GigaStar Digital Display Link über ein CAN Bus Interface machbar.

Dezentral:

- + Über die LVDS Schnittstelle können die an die Panel PC angeschlossenen Anzeigen gesteuert werden.

**3.4.2 Entscheid für die Variante Dezentral**

Wenn die Ausbaufähigkeit des Systems betrachtet wird, muss der Entscheid auf die dezentrale Variante fallen. Nur diese erlaubt es, in grösserem Umfang Touchscreens, z.B. für Fahrplanauskünfte<sup>14</sup>, anzubinden. Nur dort kann jede Anzeige problemlos ein, vom gerade über das Netzwerk vorgegebenen Infotainment Inhalt, unabhängiges Bild anzeigen. Leider ist bei solchen Anwendungen die vom Terminal benötigte Bandbreite zum Herunterladen der Informationen oder Webseiten nicht unproblematisch. Falls viele solche interaktive Terminals installiert werden sollten, ist eine Beeinträchtigung der angebotenen Dienste möglich. In diesem Falle wäre der Umstieg auf eine andere, ebenfalls weit verbreitete Netzwerktechnologie in Betracht zu ziehen. Gigabit Ethernet eignet sich hierzu hervorragend, da durch die vielen Anbieter von Geräten,

<sup>13</sup>Denkbar wäre auch die Übertragung von mehreren Inhalten auf derselben Leitung, sowie die Vergabe von IDs an die Anzeigen. So könnte dann je nach ID der Anzeige ein anderer Inhalt wiedergegeben werden. Dabei würde aber die zur Verfügung stehende Bandbreite pro Inhalt massiv reduziert.

<sup>14</sup>Bei Berührung des Bildschirms könnte auf eine Eingabemaske zur Fahrplanabfrage umgeschaltet werden. Die Stationseingabe könnte z.B. ähnlich wie beim Billetautomaten der SBB erfolgen. Die Anfrage könnte dann via WLAN oder GSM/GPRS an den Onlinefahrplan der SBB geschickt werden.

Chips und Kabel von einem tiefen Preisniveau ausgegangen werden kann. Bei der dezentralen Variante kann ausserdem eine Abhängigkeit von nur einem einzelnen Anbieter vermieden werden. Dies ist bei der Verwendung von proprietären Technologien wie GigaStar leider nicht möglich. Der Entscheid der Ascom, Panel PCs und Ethernet für ihr FIS zu verwenden, gründete ursprünglich auch auf einem Vergleich der Kosten. Dieser Vergleich zeigte, dass eine Lösung basierend auf einem Server und Gigastar auch noch beim Anschluss von 8 Bildschirmen teurer wäre, als die Variante mit Panel PCs. Wie dies heute genau aussehen würde, kann und soll an dieser Stelle nicht geklärt werden. Anhaltspunkte können aber die in Tabelle 3.5 genannten Hardwarepreise liefern. Falls der Einsatz von unterschiedlich grossen Anzeigen, sowie von Erweiterungsmöglichkeiten bezüglich interaktiven Terminals nicht vorgesehen ist, wird empfohlen, den Einkauf und die Integration einer Komplettlösung<sup>15</sup> in Betracht zu ziehen.

## 3.5 Auswahl der zu evaluierenden Konzepte

Nach dem Entscheid für die dezentrale Variante gilt es nun, aus den im Abschnitt 3.2 vorgestellten Konzepten die am Erfolgsversprechendsten auszuwählen. Die Wahl fiel auf folgende Konzepte:

- Multicast: Multicast
- Multicast: Screen Export

### 3.5.1 Multicast: Multicast

Das im Abschnitt 3.2.3 vorgestellte Konzept, welches Multicast sowohl für die Verbreitung der Videodaten per Streaming Server als auch für die Verteilung der anderen Inhalte per Multicast-Dateiübertragung verwendet, ist für eine Realisierung die erste Wahl. Der wichtigste Grund hierfür ist die Skalierbarkeit. Es steht unabhängig von der Anzahl angeschlossener Panel PCs immer gleich viel Bandbreite pro Panel PC zur Verfügung. Ebenfalls ein wichtiges Argument ist, dass beim Versenden der Videostreams per Multicast, die Anzeigen bestmöglichst synchronisiert werden. Bei entsprechender Priorisierung kommt nur die unterschiedliche Signallaufzeit, sowie das Durchlaufen des IP Stacks und der darstellenden Videosoftware zum Tragen. Von den im Konzept bereits genannten Nachteilen wiegt wohl der benötigte Bufferspeicher am schwersten. Damit alle Panel PCs gleichzeitig mit dem Abspielen des nächsten Inhaltes beginnen können, muss dieser, ausser im Falle des gestreamten Videos, lokal auf dem Panel PC vorhanden sein. Zur Sicherstellung, dass ein benötigter Inhalt zur rechten Zeit auf den Slaves verfügbar ist, muss ein spezieller Algorithmus definiert werden. Abschnitt 7.1 erläutert den für den Demonstrator verwendeten Algorithmus. Wegen dem sehr beschränkten Speicherplatz auf der von den Panel PCs verwendeten CompactFlash Karte und wegen dem Umstand, dass die Lebensdauer des Flash Speichers durch jeden Schreibzugriff abnimmt, bietet sich die Verwendung einer RAM Disk an.

### 3.5.2 Multicast: Screen Export

Für dieses Konzept spricht vor allem die einfache Softwarestruktur. Ausser einem VCN Client ist keine zusätzliche Software zur Steuerung der Dateiübertragung oder des Players auf den Slaves notwendig. Auch wird kein lokaler Speicher zur Bufferung von Infotainment Daten benötigt, da der Master Panel PC lediglich den gerade angezeigten Bildschirminhalt verschickt. Ein Nachteil dieses Konzepts ist, dass der Master und die Slaves nicht synchron sind. Das kommt daher, dass ein vom Master gerade angezeigtes Bild zuerst komprimiert und verschickt werden muss. Denkbare Lösungen für dieses Problem sind die Verwendung eines Masters ohne Anzeige oder dessen gleichzeitige Verwendung als Master und Slave. Der Master kann dann ebenfalls die von ihm versendeten Daten empfangen und auf der Anzeige darstellen. Der zu versendende Bildschirminhalt wird hierzu für eine virtuelle Anzeige generiert. Ob der Panel PC und die verfügbare Multicast Screen Export Software genügend schnell sind, um bewegte Bilder in ausreichender Qualität darzustellen, wird die Evaluation in Abschnitt 4.3 zeigen.

---

<sup>15</sup>z.B. von Innova Computers



# Kapitel 4

## Der Demonstrator

Die Bezeichnung Demonstrator wird für den während dieser Masterarbeit zusammengebaute und für Tests benützte Laboraufbau verwendet. Die Abbildung 4.2 zeigt den Laboraufbau von vorne, Abbildung 4.3 einen einzelnen Panel von oben und hinten. Damit daraus schliesslich ein Panel PC wird, reicht ein Einbau in ein entsprechendes Gehäuse. Im Folgenden wird die für den Demonstrator verwendete Hardware und Software vorgestellt. Die Resultate der Performancemessungen erlauben schliesslich, einen Entscheid für eine bestimmte Hardware- und Softwarekonfiguration zu fällen.

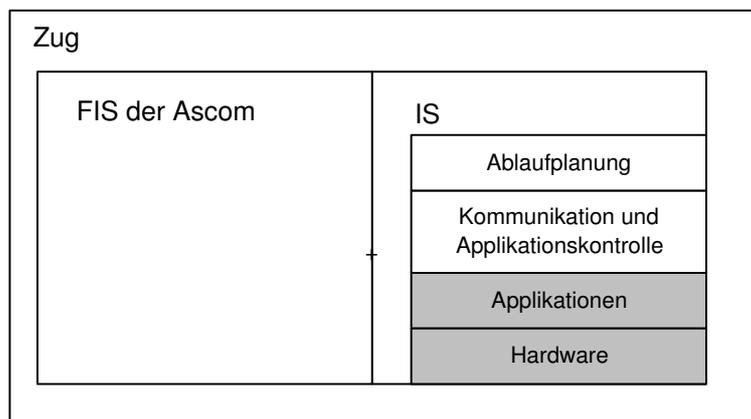


Abbildung 4.1: Betrachtete Systemebene: Hardware und Applikationen



Abbildung 4.2: Laboraufbau von vorne

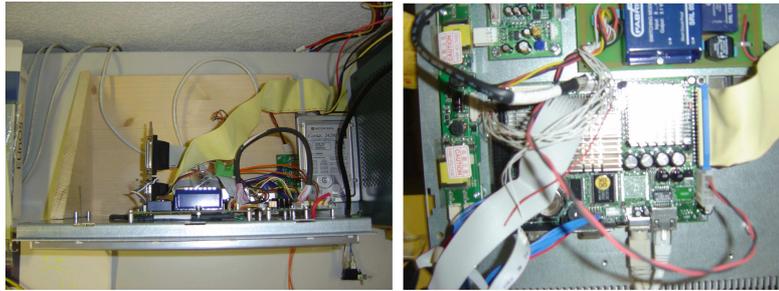


Abbildung 4.3: Einzelner Panel von oben und hinten

## 4.1 Verwendete Hardware

Die Hardware eines bereits für das FIS der Ascom eingesetzten Panel PCs sollte als Basis für den Demonstrator dienen. Die Herstellungskosten für diesen Panel PC betragen ungefähr 3330US\$.<sup>1</sup> Der Panel PC besteht aus einem TFT Bildschirm, einem SBC und einem Gehäuse. Alles soll den industriellen Anforderungen genügen. Das Gehäuse z.B. muss hierfür verschiedene Sicherheitsnormen erfüllen.<sup>2</sup> Für den Laboraufbau wurde wegen der besseren Zugänglichkeit der Hardware kein Gehäuse verwendet. Zur Speicherung der Software ist bei den Slaves der Einsatz einer CompactFlash Karte mit einer Speicherkapazität von 256 oder 512 MByte vorgesehen. Weil diese Kapazität zur Speicherung der Infotainment Inhalte auf dem Master aber kaum ausreicht, stellt Abschnitt 4.1.1 Speicherlösungen für den Master vor. Der vollständig passiv gekühlte SBC ECM 3610 von Evalue Technology Inc. ist mit einem Via Eden ESP6000 Prozessor, sowie dem Twister-T Grafikchip von S3 bestückt. Der Grafikchip verwendet einen Teil des Hauptspeichers als Grafikspeicher und hat einen integrierten MPEG Decoder. Der Via Eden Prozessor basiert auf dem Ezra Kern und läuft mit einer Taktfrequenz von 667MHz. Nachforschungen zeigten, dass die Floating Point Unit(FPU) dieses Prozessors nur mit halber Taktfrequenz läuft. Zudem ist die für die Wiedergabe von Multimedia Daten wichtige Erweiterung Streaming SIMD Extensions(SSE)<sup>3</sup> nicht vorhanden. Nur die MultiMedia eXtension(MMX)<sup>4</sup> ist vorhanden. Die Performance Messungen am fertigen System aus Abschnitt 4.3 zeigen, dass zusätzlich ein leistungsfähigerer SBC in Betracht gezogen werden sollte. Die Anforderungen an einen solchen SBC sind:

- Vollständig passiv kühlbar.
- LVDS Grafikausgang (ev. durch Zusatzmodul)
- Einschub für eine CompactFlash Karte
- Board Format: 3,5"
- Schneller als das ECM 3610

Die Auswahl an Produkten, welche all diese Anforderungen erfüllen, ist relativ klein. Eigentlich kommt nur das ECM 3611, ebenfalls von Evalue Technology Inc., in Frage. Dieser SBC basiert auf einem VIA C3 Prozessor mit Nehemiah Kern und 1GHz Taktfrequenz. Dieser Prozessor ist Intel 686 kompatibel, besitzt also auch die SSE Erweiterung. Im Gegensatz zum Via Eden Prozessor läuft hier die FPU mit vollem Prozessortakt. Ein Minuspunkt dieses SBC ist, dass er in der Standardausführung aktiv gekühlt wird. Der kleine Kühlkörper und der darauf montierte Ventilator sind im Bild in der Tabelle 4.1 zu sehen. Inovis, der Lieferant des ECM 3610 für die Panel PCs des FIS der Ascom, versicherte aber, dass eine passive Kühlung möglich ist. Der Grafikchip ist ein VIA CLE266 mit integriertem MPEG Decoder. Als Grafikspeicher verwendet er, genau wie der Twister-T des ECM 6310, einen Teil des Hauptspeichers. Im Gegensatz zu

<sup>1</sup>Stand November 2004

<sup>2</sup>Vibrationsresistenz gemäss Europäischer Norm EN50155, Glasteile aus Einscheiben-Sicherheitsglas (ESG) oder Verbundsicherheitsglas(VSG), Gehäuse der Schutzklasse IP56 [8] u.s.w.

<sup>3</sup>Die Webseite <http://www.tommeseani.com/SSE.html> liefert einige Details zu SSE.

<sup>4</sup>Die Webseite <http://www.tommeseani.com/MMXPrimer.html> liefert einige Details zu MMX.

jenem ist hier aber die Verwendung von DDR SDRAM möglich. Dadurch ist eine Entschärfung des Engpasses Speicherinterface zu erwarten. Tabelle 4.1 fasst die technischen Spezifikationen der beiden SBCs, sowie einige Kennzahlen zum verwendeten TFT zusammen. Die Angaben für das ECM 3610 stammen aus [13], diejenigen für das ECM 3611 aus [14]. Spezielle Daten über die verwendeten Prozessoren wurden dem Dokument [15] entnommen. Weitere Details zum verwendeten TFT sind in [12] zu finden.

	ECM 3610 Evalue Technology Inc.	ECM 3611 Evalue Technology Inc.
		
Prozessor	VIA Eden ESP6000, 667MHz, Ezra Kern	VIA C3, 1GHz, Nehemiah Kern
SSE Erweiterung	Nein	Ja
FPU	Ja, mit halbem Prozessortakt	Ja
Grafik Chip	S3 Twister-T, AGP4x	Via CLE266, AGP8x
MPEG Decoder	Ja <sup>a</sup>	Ja
Hauptspeicher	SDRAM, 133MHz	DDR SDRAM, 266MHz
CompactFlash Karten	bis 512 MByte	bis 1 GByte
Harddisk Anschlüsse	Enhanced IDE	Enhanced IDE, Serial ATA
Display Anschluss	Digitalausgang, externer Konverter zu LVDS	LVDS Ausgang
Samsung LTM150XH-L05 TFT		
Auflösung	1024x768 bei 75 Hz	
Farben	16,2 Millionen	
Anschluss	LVDS	
Versorgungsspannung	3.3V	

<sup>a</sup>Nutzung unter Linux kaum möglich. Siehe Abschnitt 4.2.2 für Details.

Tabelle 4.1: Angaben zu den verwendeten SBCs und dem eingesetzten TFT Display.

#### 4.1.1 Speichererweiterung des Master Panel PCs

Die im Panel PC installierte 256 MByte Flash Karte reicht nicht aus, um Multimedia Daten für das Infotainment abzulegen. Es könnte einerseits eine Flashkarte mit höherer Kapazität verwendet oder der IDE Anschluss des SBC zur Erweiterung des Speichers genutzt werden. Da der Slot für die Flashkarte auf Karten mit maximal 512 MByte Kapazität beschränkt ist, muss die Erweiterung über den IDE Anschluss erfolgen. Es kann entweder eine vibrationsresistente Harddisk oder ein als Harddisk anschliessbares Flash Drive verwendet werden. Tabelle 4.2 enthält Kennzahlen zu momentan<sup>5</sup> verfügbaren Produkten. Die Kosten für die Integration des jeweiligen Produktes in den Panel PC, sind hier nicht berücksichtigt. Durch die ähnliche Bauform dürften sie aber für alle etwa gleich hoch sein.

Die effektiv benötigte Menge an zusätzlichem Speicherplatz hängt sehr vom Einsatzprofil des Infotainmentsystems ab. Die Art der gezeigten Inhalte, sowie deren Häufigkeit sind die bestimmenden Faktoren. Die drei in Tabelle 4.3 vorgestellten Einsatzprofile verwenden die in Tabelle 3.1 spezifizierten Infotainment Inhalte.

<sup>5</sup>Stand 02.12.2004

	Harddisk	Flash Drive 1	Flash Drive 2
Hersteller	Fujitsu[20]	Memtech[18]	BiTMICRO Networks, Inc. [19]
Modell	MHT AH Series 2.5"	AT2550 Wolverine (2.5")	E-Disk, IDE/ATA 3.5"
Kapazität	40 - 80 GBytes	1 - 60 GByte	1 - 155.6 GByte
Anhaltende Übertragungsrates	100 MBytes/s	20 MBytes/s	14 - 28 MBytes/s
Mittlere Zugriffszeit	18 ms	0.1 ms	68 - 250 $\mu$ s
Vibrationsresistenz (zufällige Vibrationen)	1G (in Betrieb)	Bis 15G	Bis 16.7G
Shock Resiszenz	225 G für 2 ms (in Betrieb)	1000 G	2000 G
Lebenszeit	3 Jahre (300'000 mal Start / Stop des Spindelmotors)	Lesevorgänge: unlimitiert,  Schreibvorgänge: 8 Millionen	Lesevorgänge: unlimitiert,  Schreibvorgänge: 926 TByte geschriebene Daten (=27 Jahre mit 100 GByte geschriebene Daten pro Tag)
Temperaturbereich	5 - 55 Grad Celsius	max. -55 bis +125 Grad Celsius	max. -60 bis +125 Grad Celsius
Kosten <sup>a</sup>	60 GByte: 136 US\$	27 GByte: 12'000 US\$ 16 GByte: 7400 US\$	30 GByte: 12'336 US\$

<sup>a</sup>Stand 02.12.2004, Quellen:

Fujitsu: <http://www.softhardag.ch>,

Memtech: <http://www.storesys.com/publichp/produkt/storesys/massensp/altec.pdf>,

BiTMICRO: Angaben von Noel M. Vinluan, Verkäufer bei BiTMICRO

Tabelle 4.2: Speichererweiterung via IDE Anschluss

- Werbeclip/Film (Video): max. 9.8MBit/s
- Werbeclip Animation oder Standbilder: < 800kBit/s
- Fahrgastinformation: < 100kBit/s

Einsatzprofil	Intercity	Regionalzug	Sonderfahrt
Einsatzdauer pro Tag	20 Stunden	20 Stunden	8 Stunden
Werbeclip/Film (Video)	60%	20%	98%
Werbeclip Animationen	35%	60%	0%
Fahrgastinformation	5%	20%	2%
Anz. Wiederholungen	1	3	0
Benötigter Speicherplatz:	26 GByte	5.2 GByte	32 GByte

Tabelle 4.3: Benötigter Speicherplatz bei verschiedenen Einsatzprofilen

Der benötigte Speicher fällt je nach Einsatzprofil sehr unterschiedlich aus. Die Speichererweiterung sollte dem Einsatzprofil des Kunden angepasst werden. Zu Beachten gilt es ausserdem noch, dass während der Ruhezeit des Fahrzeuges ein vollständiges Update möglich sein sollte. Wird das Update für das Einsatzprofil Intercity z.B. über WLAN 802.11b vorgenommen, muss die Ruhezeit ca. 6 Stunden betragen. Bei 802.11g sind es nur noch ca. 1.5 Stunden. Für die im Rahmen dieser Arbeit durchgeführten Tests wird eine herkömmliche IDE Festplatte verwendet. Dies sowohl für den Master, wie auch für die Slaves.

## 4.2 Verwendete Software

Eine der wichtigsten Aufgaben war das Finden der für die Realisierung der zwei ausgewählten Konzepte benötigten Softwarekomponenten. Die Softwarekomponenten können in folgende Gruppen eingeteilt werden:

- Betriebssystem
- Grafische Oberfläche/Grafikausgabe
- Applikationen

### 4.2.1 Betriebssystem

Als Betriebssystem ist Linux durch die Ascom vorgegeben worden. Die Vorteile von Linux liegen klar bei den wegfallenden Lizenzkosten, sowie der Möglichkeit, ein sehr kompaktes, dem Anwendungszweck entsprechendes System konfigurieren zu können. Die Nachteile liegen in der oft fehlenden oder nicht vollständigen Unterstützung von spezieller Hardware, sowie dem enormen Konfigurationsaufwand. Viele Softwarepakete müssen erst auf das System angepasst und kompiliert werden. Dies fällt oft besonders schwer, falls ein Softwarepaket nicht auf dem System installiert werden soll, auf dem es kompiliert wird. Dies trifft hier zu. Das auf die Panel PCs aufzuspielende System wird auf einem Entwicklungsrechner zusammengestellt und konfiguriert. Erst danach wird es auf eine CompactFlash Karte oder eine Harddisk überspielt. Zur Konfiguration und Kompilierung wird die Cross-Development Toolchain ELinOS(Embedded Linux Operating System) in der Version 3.1 [21] eingesetzt. Diese erlaubt es, eine Grundkonfiguration des Linux Systems über eine graphische Oberfläche vorzunehmen. Dabei reichen die Optionen vom kompakten, für Embedded Anwendungen geeigneten Linux bis zu einem auch für den täglichen Gebrauch im Büro geeigneten Grundsystem. ELinOS kümmert sich ausschliesslich um das Basissystem. Webbrowser, eine graphische Oberfläche, wie z.B. KDE(Kommon Desktop Environment), oder Player zum Abspielen von Multimedia Daten müssen selbst in das System eingebracht werden. Auch einige weiterführende Konfigurationen sind nur über die manuelle Editierung der entsprechenden Dateien zu erreichen. Einige wichtige Hinweise, sowie eine Installationsanleitung für die im Demonstrator verwendete Konfiguration sind im Anhang C und D zu finden. Die Tabelle 4.5 enthält einige Kennzahlen zum verwendeten Betriebssystem.

## 4.2.2 Grafische Oberfläche/Grafikausgabe

Zur Ansteuerung der Grafikhardware gibt es in Linux mehrere Ansätze:

**Verwendung eines XServers:** Auf dem XServer laufen sogenannte X Window Anwendungen. Die Anwendungen kommunizieren über die Netzwerkschicht mit dem XServer und benutzen dessen Interface zur Grafikausgabe. Da Sie über die Netzwerkschicht mit dem XServer kommunizieren, ist es egal, ob der XServer sich auf demselben Rechner befindet oder nicht. Heute gibt es allerdings auch Erweiterungen, welche die Umgehung dieser Netzwerkschicht bei einem lokalen XServer erlauben. Dazu gehören z.B. das Direct Rendering Interface(DRI) oder der Direct Graphics Access(DGA). Der XServer selbst setzt zur Grafikausgabe meist entweder auf dem SVGA oder dem Framebuffer Device auf. Er bringt hierzu für die meisten gebräuchlichen Grafikkarten sowohl einen Treiber, der auf dem SVGA Device aufsetzt, als auch einen Treiber, der auf dem Framebuffer Device aufsetzt. Die Devices werden durch spezielle Kernelmodule generiert und verwaltet. Sie ermöglichen den Zugriff auf den Speicher und die Register der Grafikhardware.

**Grafikausgabe in der Konsole:** Die Grafikausgabe in der Konsole erfolgt durch die direkte Verwendung des SVGA oder des Framebuffer Devices durch die Anwendungen. Es gibt allerdings nur wenige grafische Anwendungen für die Konsole. Einerseits existieren die für das IS relevanten Applikationen zum Abspielen von Videos, aber andererseits gibt es keinen grafikfähigen Webbrowser und keine VNC Software hierfür. Da das Konzept Multicast:Multicast einen Webbrowser und das Konzept Multicast:Screen Export VNC Software voraussetzt, fällt diese Möglichkeit zur Grafikausgabe weg.

Es galt nun, denjenigen XServer zu finden, der beim Abspielen von Videos die geringste Prozessorlast verursacht. Die Prozessorlast hängt vor allem von den durch den Grafiktreiber und den Grafikchip unterstützten Methoden zum Abspielen von Videos ab. Folgende Fragen müssen zur Beurteilung eines XServers geklärt werden: Welche Methoden zur Grafikausgabe werden unterstützt?

- X11: Der einfache X11-Bildausgang. Ein laufender XServer genügt hierfür. Dekodieren, Skalieren u.s.w in Software.
- Xvideo(auch Xv genannt): Dieser Ausgang benötigt eine Xvideo-kompatible Grafikkarte. Hardwarebeschleunigung für YUV-Transformationen<sup>6</sup> und Skalierungen.

Ist es möglich, zum Abspielen den MPEG Decoder des Grafikchips zu benutzen?

- XvMC: Xvideo Motion Compensation. Generisches XServer Interface für das Benutzen eines MPEG Decoders zum Abspielen von Videos. Der verwendete Grafiktreiber muss dieses Interface implementieren. Ausserdem muss ein spezielles Direct Rendering Manager(DRM) Kernel Modul, das den direkten Zugriff auf den Grafikchip ermöglicht, vorhanden sein.

Kann die 3D Beschleunigung des Grafikchips aktiviert werden?

- DRI: Direct Rendering Interface. Generisches XServer Interface für den Zugriff auf den 3D Beschleuniger. Der verwendete Grafiktreiber muss dieses Interface implementieren. Auch hier wird das DRM Kernel Modul benötigt.

Verbreitete XServer sind die von X.Org [23] oder XFree86 [24]. Es gibt jedoch auch noch weitere, weniger verbreitete XServer. Ein Beispiel hierfür ist der ebenfalls getestete kommerzielle XServer Accelerated-X Summit v2.2 von Xi Graphics [22]. Von beiden XServern wurden bereits viele verschiedene Versionen veröffentlicht. Eine neuere Version des XServers zu benutzen,

<sup>6</sup>Eine Methode, Farbwerte in Rastergrafiken oder Videodaten darzustellen, ähnlich wie bei RGB. Bei RGB wird eine Farbe aus einer Mischung der Grundfarben Rot, Grün und Blau erzeugt. Das RGB Signal gibt für jede der drei Grundfarben an, wie viel hinzu gemischt werden soll(Intensität). Auch YUV stellt eine Farbe mit drei Werten dar, nämlich die Helligkeit Y und die Farbwerte U und V. MPEG-2(DVD) kodierte Daten enthalten Farbinformationen im YUV Format. Zur Darstellung auf dem Monitor müssen diese normalerweise auf RGB umgerechnet werden.

bedeutet aber nicht unbedingt, dass die in einer älteren Variante unterstützten Features immer noch verwendet werden können. Für die Variante XFree86 R4.2.0 gab es zum Beispiel einen Closed Source Treiber von S3, welcher den MPEG Decoder des Twister-T Chips auf dem ECM 3610 benutzen konnte. 2003 wurde diese Source veröffentlicht, eine Portierung auf spätere Versionen von XFree86 gibt es aber nicht. Die Open Source Treiber, die in den XServern von X.Org und XFree86 enthalten sind, beinhalten nur den Code für die Server Seite, weil die Bibliothek für die Client Seite anscheinend nie funktioniert hat. Es bleibt abzuwarten, ob sich daran in Zukunft noch etwas ändern wird.<sup>7</sup> Bezüglich X.Org oder XFree86 soll hier offengelassen werden, welcher die bessere Wahl ist.<sup>8</sup> Basierend auf den Performance Messungen, sowie dem subjektiven Eindruck bezüglich der Qualität der angezeigten Videos wurden schliesslich die in Tabelle 4.2.2 hervorgehobenen und die in Tabelle 4.5 spezifizierten Konfigurationen verwendet. Tabelle 4.2.2 enthält Daten zu einer Auswahl an getesteten Konfigurationen. Eine detaillierte Installationsanleitung für das beim ECM 3611 eingesetzte System enthält Anhang D.

### 4.2.3 Applikationen

Hier werden die für den Demonstrator verwendeten Applikationen vorgestellt, sowie auf allfällige Probleme damit hingewiesen. Einige Applikationen wurden erst im Verlauf dieser Arbeit durch Patches oder Updates für das IS verwendbar.

### 4.2.4 VNC Server und Client

Um das Konzept Multicast:Screen Export realisieren zu können, wird ein Multicast fähiger VNC Server und VNC Client benötigt. Die Software MulticastVNC [39], welche an der Universität von Trier entwickelt wurde, bietet diese Funktionalität. Das Prinzip der Software ist, dass ein herkömmlicher VNC Server gestartet wird (z.B. RealVNC [38], TightVNC [37]), der die Daten an die als Client fungierende Multicast VNC Server Software sendet. Diese übernimmt dann die Verteilung der Daten per Multicast an die Multicast VNC Clients. Die Wahl für den VNC Server fiel auf TightVNC. Dieser Server erlaubt es, unterschiedliche Kompressionsraten für die versendeten Daten zu verwenden. Somit kann die Belastung des Netzwerkes durch höhere Kompressionsraten reduziert oder die Prozessorlast durch geringere Kompressionsraten vermindert werden.

### 4.2.5 Multicast Dateitransfer

Um das Konzept Multicast:Multicast realisieren zu können, wird eine Software zum Übertragen von Dateien per Multicast benötigt. Die Software UDPCast bietet die benötigte Funktionalität. Das ursprünglich zur Installation von Software auf mehreren PCs, z.B. in Computerräumen von Schulen, gedachte Tool wird auch heute noch ständig weiter verbessert. Die Anwendungsmöglichkeiten und die Optionsvielfalt des Tools sind gross. Ein Beispiel: Zur Übertragung von Dateien stehen sowohl eine Methode mit einem Rückkanal per Unicast als auch eine Methode mit Forward Error Correction(FEC) ohne Rückkanal zur Wahl. Die FEC kann zudem über Parameter den eigenen Bedürfnissen angepasst werden. Für detailliertere Informationen sei auf [40] verwiesen.

### 4.2.6 Streamen und Anzeige von Videos

Für das Streamen von Videos wurden die weit verbreiteten Programme MPlayer [30] und VideoLan Client(VLC) [33] in Betracht gezogen. Das Empfangen und Anzeigen der Videostreams kann ebenfalls von diesen Programmen übernommen werden. Weil aber die Videos schliesslich in einem Webbrowser angezeigt werden sollen, muss dieser den MPlayer und/oder den VLC ansteuern können. Momentan kommt hierfür nur der Mozilla Webbrowser mit den entsprechenden Plug-Ins in Frage.

<sup>7</sup>Das Interesse daran scheint, wie das Mail auf <http://lists.freedesktop.org/pipermail/xorg/2005-February/006018.html> zeigt, noch vorhanden zu sein.

<sup>8</sup>Diskussionsthema im Internet. Ein Beitrag hierzu: <http://www.linuxforums.org/forum/topic-25071.html>

Konfigurationen für den ECM 3610				
Verwendeter XServer	XFree86 R4.2.0	XFree86 R4.4.99	X.Org R6.8.1	Accelerated-X Summit v2.2
Grafiktreiber	S3(Closed Source) / XFree86	XFree86	X.Org	Accelerated-X (Closed Source)
Xvideo(Xv)	Ja / Ja	Ja	Ja	Ja <sup>a</sup>
Xvideo Motion Compensation (XvMC) <sup>b</sup>	Ja <sup>c</sup>	Nein	Nein	Nein
3D Beschleunigung	Ja	Ja	Nein <sup>d</sup>	Ja
Subjektive Bildqualität: <sup>e</sup>				
X11	ruckelt stark	ruckelt stark	ruckelt stark	ruckelt stark
Xvideo	ruckelt	ruckelt	ruckelt wenig	ruckelt selten
XvMC	nur grünes Rechteck	–	–	–
Konfigurationen für den ECM 3611				
Verwendeter XServer	X.Org Release 6.8.2	XFree86 Release 4.4.0		
Grafiktreiber	X.Org / Unichrome Project <sup>f</sup>	XFree86 / Unichrome Project		
Xvideo(Xv)	Ja / Ja	Ja / Ja		
Xvideo Motion Compensation (XvMC)	Nein / Ja	Nein / Ja		
3D Beschleunigung	Nein <sup>g</sup> / Ja	Nein / Ja		
Subjektive Bildqualität:				
X11	ruckelt / ruckelt flüssig / flüssig	ruckelt / ruckelt flüssig / flüssig		
Xvideo	– / nicht getestet	– / flüssig / flüssig		
XvMC <sup>h</sup>				

<sup>a</sup>Proprietäre Implementation des Xvideo Interfaces für die Videowiedergabe.

<sup>b</sup>In Zukunft eventuell unterstützt. Siehe hierzu <http://lists.freedesktop.org/pipermail/xorg/2005-February/006018.html>

<sup>c</sup>Treiber existiert, lief aber nie richtig. Nur grünes Rechteck bei Wiedergabe mit dem MPlayer.

<sup>d</sup>Aktivierung möglich, 2D Leistung des Treibers scheint aber geringer zu sein

<sup>e</sup>MPEG-2 Video mit durchschnittlicher Bitrate von 7 MBit/s über das Netzwerk gestreamt. X11 und Xvideo Wiedergabe mit dem VLC und dem MPlayer, XvMC Wiedergabe nur mit dem MPlayer.

<sup>f</sup>Kaum ausführliche Anleitungen. Howtos für XFree86 4.4.0 Installation mit einigen Anpassungen verwendbar.

<sup>g</sup>Eventuell über X.Org vom CVS, zusätzlicher Mesa Bibliothek, sowie dem DRM Kernel Modul vom CVS möglich. Howto unter: <http://dri.freedesktop.org/wiki/Building>

<sup>h</sup>Nur mit gepatchtem, selbst kompiliertem MPlayer. Anleitung im Anhang D.4.3.

**Streaming:** Für das Streaming per Multicast existieren zwei gebräuchliche Protokolle. Das Eine ist das Real Time Protocol(RTP), welches auf dem User Datagram Protocol(UDP) aufsetzt, das Andere ist UDP alleine. RTP fügt im Gegensatz zu UDP zu jedem Paket einen Zeitstempel hinzu und kann mehrere Streams, z.B. ein Audio und ein Videostream, synchronisieren. Durch die erweiterte Funktionalität erhöht sich aber auch die Anzahl der für die Headerdaten pro Paket benötigten Bytes um 12. Je nach durchschnittlicher Paketgrösse verursacht dies ca. 1-9%<sup>9</sup> mehr Netzwerkverkehr. Die weitere Funktionalität des RTP Protokolls hängt sehr vom transportierten Datenformat ab. Einige Fragen und Antworten zu diesem Protokoll sind in [11] zu finden. Der MPlayer unterstützt im aktuellen Release 1.0pre6a nur das Streamen per RTP. Dies aber auch nur, wenn er mit Unterstützung für die LIVE.COM Streaming Media<sup>10</sup> Bibliothek kompiliert wird. UDP Streaming ist allerdings seit kurzem im CVS<sup>11</sup> und wird in zukünftigen Versionen wohl enthalten sein.

**Grafikausgabe:** Für die Grafikausgabe sollten die Programme mindestens Xvideo unterstützen. Laut den Spezifikationen tun sie dies auch. Allerdings gab es mit dem offiziellen VLC Release 0.8.1 Probleme bei der Xvideo Ausgabe. Der VLC stürzte bei Verwendung von Xvideo und dem Unichrome Grafiktreiber(ECM 3611) reproduzierbar ab. Seit dem 18. November gibt es einen Patch<sup>12</sup> für den Quellcode, der aber nicht die Ursache des Fehlers behebt. Seit dem 19. Dezember ist die Ursache des Fehlers gefunden und die nötigen Änderungen am Code im CVS gemacht. Der offizielle Release 0.8.1 enthält diese Korrekturen aber noch nicht. Für die Tests mit der Xvideo Ausgabe war die Anpassung des Sourcecode des offiziellen Releases nötig, da diese z.T. bereits Ende November 2004 gemacht wurden. Zwecks Vergleichbarkeit wurde diese Version auch nach Verfügbarkeit der CVS Version beibehalten.

**Plug-Ins:** Das vlc-plugin [34] kann alle Funktionen des VLC nutzen. Bei den Plug-Ins für den MPlayer scheint dies leider nicht der Fall zu sein. Es wurde bisher kein Weg gefunden, mit den Plug-Ins 'mozplugger' [32] und 'mplayer-plugin' [31] Multicast Streams zu empfangen und abzuspielen. Dies ist schade, da der MPlayer, im Gegensatz zum VLC, die Videoausgabe über XvMC unterstützt. Allenfalls könnte eine gegen Ende dieser Masterarbeit gefundene und nicht näher geprüfte Methode zum Empfangen von Real Time Session Protocol(RTSP) Streams zum Erfolg führen. Diese beruht auf der Verwendung einer sogenannten Session Description Protocol(SDP) Datei zum Auffinden und Abspielen eines Multicast Streams. Genauere Angaben über Konfiguration und Nutzung dieses Features sind aber schwer zu finden.

Tabelle 4.4 fasst den Vergleich des MPlayer in der Version 1.0pre6a und des VLC in der Version 0.8.1 zusammen.

## 4.3 Performance Evaluation

Aufgrund der Performance Messungen für die Konzepte Multicast:Screen Export und Multicast:Multicast wird entschieden, welches von diesen weiterverfolgt wird. Für die Messungen auf dem Demonstrator werden die in Tabelle 4.5 genannten Konfigurationen verwendet.

Vor der eigentlichen Messung wurde eine Abschätzung auf dem Entwicklungsrechner gemacht. Nur falls diese Abschätzung zufriedenstellende Ergebnisse lieferte, wurden Messungen auf dem Demonstrator vorgenommen. Kennzahlen des Entwicklungsrechners sind:

- Prozessor: Pentium M 1,5 GHz
- RAM: 1 GByte
- Grafikkarte: ATI 9600Pro, 128 MB Grafikspeicher

<sup>9</sup>Bei angenommener minimaler Grösse der UDP Pakete von 120 Bytes und einer MTU von 1500 Bytes

<sup>10</sup><http://www.live.com/mplayer>

<sup>11</sup>gemäss:

<http://www1.mplayerhq.hu/cgi-bin/cvsweb.cgi/main/ChangeLog?rev=1.191&content-type=text/x-cvsweb-markup>

<sup>12</sup>Beitrag in der Mailingliste unter: <http://www.via.ecp.fr/via/ml/vlc/2004-11/msg00160.html>

	MPlayer v1.0pre6a	VLC v0.8.1
Videoausgabe: x11 Xvideo(Xv) XvMC	Ja Ja Ja <sup>a</sup>	Ja Ja <sup>b</sup> Nein
Stream Input	HTTP, RTP, RTSP, MMS, MMST, SDP	HTTP, UDP, RTP, MMS, FTP, SDP
Stream Output	Nein	HTTP, UDP, RT(S)P, MMSH, FTP
Plug-Ins	mozplugger [32], mplayer- plugin [31]	vlc-plugin [34]

<sup>a</sup>Nur durch patchen des Source Codes mit dem Unichrome Project Patch sowie einem XvMC aktivierten XServer.

<sup>b</sup>Für das ECM 3611 nur in der CVS Versionen ab dem 19.12.2004 oder durch den 'Patch' auf <http://www.via.ecp.fr/via/ml/vlc/2004-11/msg00160.html> verfügbar

Tabelle 4.4: Gegenüberstellung von MPlayer v1.0pre6a und VLC v0.8.1

	ECM 3611	ECM 3610
<b>Betriebssystem</b>		
Plattform:	Linux	
Kernel:	2.6.9	
Kompiliert für:	Via C3 (Nehemiah Kern)	Via Eden(Ezra Kern)
Grafikchip unterstützt:	Nein	Nein
<b>Grafische Oberfläche/Grafikausgabe</b>		
XServer:	XFree86 R4.4.0	ECM 3610: XFree86 R4.4.99
Kompiliert für:	Via C3 (Nehemiah Kern)	Via Eden(Ezra Kern)
Grafiktreiber:	Unichrome Treiber	Treiber von X.Org oder von [25]
DRM Kernelmodule:	Vom dri.freedesktop.org CVS	Vom X.Org XServer Paket oder [25]
X11 Video:	Ja	Ja
Xvideo:	Ja	Ja
Xvideo Motion Compensation(XvMC):	Ja	Nein
<b>Applikationen (kompiliert für den jeweiligen Prozessor)</b>		
Videostreaming:	VideoLan Client(VLC) v0.8.1	
Video Player:	VideoLan Client(VLC) v0.8.1 oder MPlayer v1.0pre6a	
Browser:	Mozilla v 1.7.5	
Browser Plug-Ins:	vlc-plugin, mozzplugger, mplayer-plugin	
Vom Linux System auf dem Entwicklungsrechner kopierte Software (kompiliert für i386/i586)		
Zusätzliche Tools:	Kommandos wie: killall, ipcs u.s.w. Tools zur Messung der Performance: top, iftop, u.A.	
Bibliotheken:	Von obiger Software referenzierte Bibliotheken <sup>a</sup>	

<sup>a</sup>Werden von ELinOS bei der Generierung des Dateisystems automatisch vom Entwicklungsrechner kopiert (falls vorhanden).

Tabelle 4.5: Für den Demonstrator verwendete Systemkonfiguration

### 4.3.1 Multicast:Screen Export

Die Abschätzung mit TightVNC auf dem Entwicklungsrechner zeigt bereits deutlich die Grenzen dieses Konzepts. Die Kompression des Bildschirminhalts in Echtzeit ist selbst für den Entwicklungsrechner nicht machbar. Abbildung 4.4 zeigt die Ausgabe des Linux Tools 'top' während dem Abspielen einer DVD und dem Anzeigen der Ausgabe auf dem virtuellen Desktop von TightVNC. Neben der 100%igen Auslastung fällt auch die durch die Kompression verminderte Bildqualität auf. Schriften werden, wie in Abbildung 4.4 zu sehen ist, kantig und unscharf.

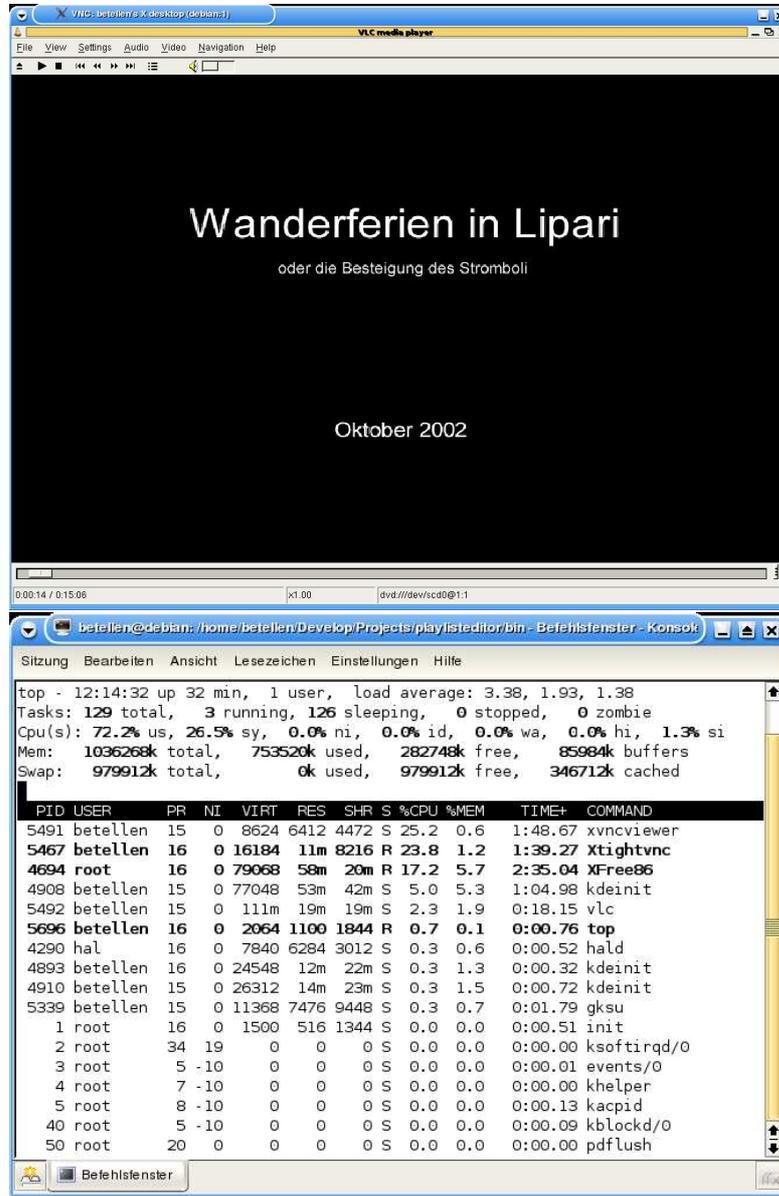


Abbildung 4.4: TightVNC auf dem Entwicklungsrechner

### 4.3.2 Multicast:Multicast

Die Abschätzungen auf dem Entwicklungsrechner zeigten eine Auslastung von durchschnittlich 35% beim Abspielen einer DVD. Die DVD wurde vom VLC und MPlayer über Xvideo ausgegeben. Die daraufhin gemachten Performance Messungen auf dem Demonstrator lieferten die in

Abbildung 4.5 gezeigten Ergebnisse. Für die Messung wurde ein Videostream mit durchschnittlicher Datenrate von 7 MBit/s an die Slaves gesendet. Die Ausgabe erfolgte, wo vorhanden, über X11, Xv oder XvMC. Da für das ECM 3610 bei Verwendung von X11 eine ständig 100%ige Auslastung resultierte und die Ausgabe kaum mehr einem Video entsprach, wurde die Messung der besseren Übersichtlichkeit wegen nicht eingetragen.

## 4.4 Fazit

Nur das Multicast:Multicast Konzept erlaubt das Anzeigen von Videostreams ohne Ruckeln oder Qualitätseinbußen. Das Multicast:Screen Export Konzept scheiterte bereits an der Performance Abschätzung auf dem Entwicklungsrechner. Im Moment kann allerdings wegen der fehlenden Unterstützung von XvMC durch den VLC, sowie dem Fehlen eines Plug-Ins, mit dem auch Multicast Videostreams angezeigt werden können, nicht die Ideallösung für das Multicast:Multicast Konzept realisiert werden. Die Ideallösung: VLC zum Streamen, MPlayer(mit XvMC Ausgabe) zum Wiedergeben. Durch bereits erwähnte Neuerungen, wie dem UDP Stream Empfang durch den MPlayer, könnte dies aber bald möglich werden. Bis dahin muss die Variante VLC zum Streamen, VLC zum Empfangen und vlc-plugin zum Anzeigen im Browser realisiert werden. Bei Streams mit Bitraten um 9 MBit/s und DVD Qualität muss daher eine durchschnittliche Prozessorauslastung von bis zu 80% in Kauf genommen werden.

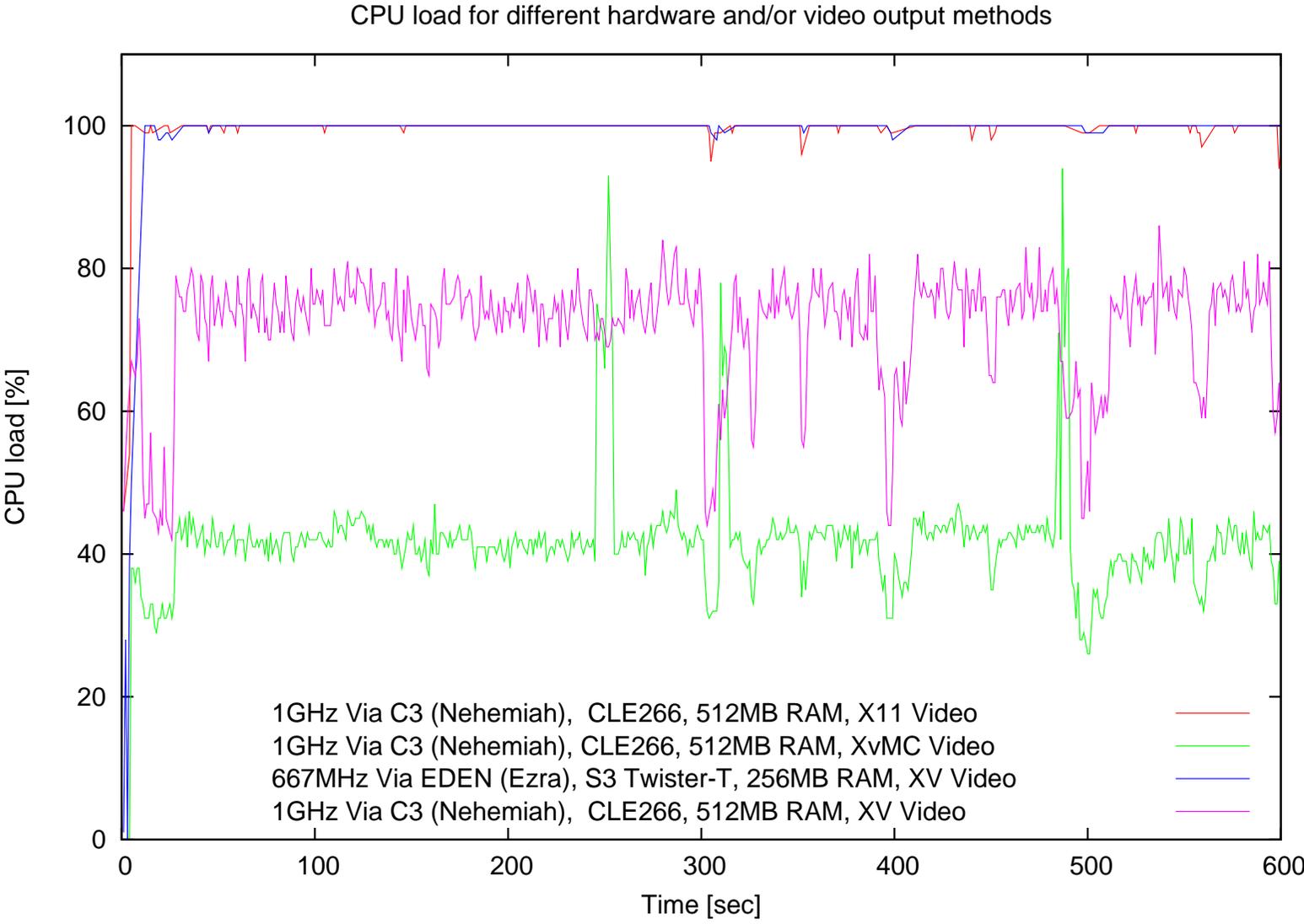


Abbildung 4.5: Performance Messungen für die beiden Demonstrator Konfigurationen



## Kapitel 5

# Kommunikations- und Kontrollebene

Gemäss dem vorangehenden Kapitel ist das dort vorgestellte Multicast:Multicast Konzept die beste Art und Weise für die Übertragung der Infotainment Inhalte an die Slave Panel PCs. Bei diesem Konzept sendet die VLC Software die Videodaten per Multicast an die Slaves. Ebenfalls per Multicast verteilt die Software UDPCast alle anderen Inhalte als Datei an die Panel PCs. Die Anzeige der Videostreams und der Inhalte in Dateiform<sup>1</sup> erfolgt im Webbrowser Mozilla. Dieses Kapitel stellt nun die zur Realisierung dieses Konzepts notwendige Kommunikations- und Kontrollebene vor. Die Kommunikations- und Kontrollebene stellt Komponenten zur Master-Slave Kommunikation, sowie zur Kontrolle der Applikationen VLC, UDPCast und Mozilla bereit.

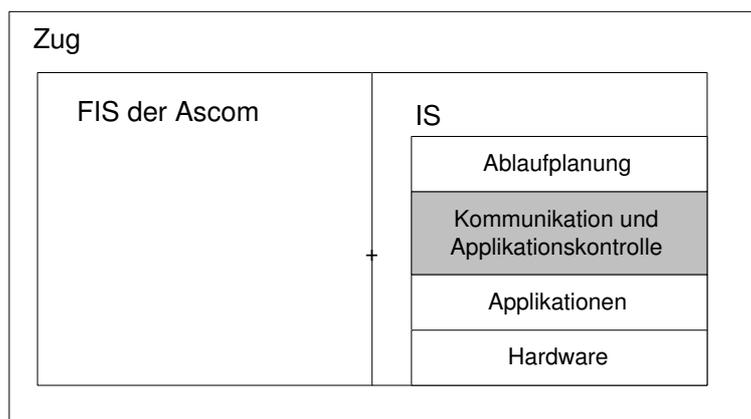


Abbildung 5.1: Betrachtete Systemebene: Kommunikation und Applikationskontrolle

### 5.1 Software Architektur der Kommunikations- und Kontrollebene

Abbildung 5.2 zeigt die Software Architektur für die Kommunikations- und Kontrollebene. Die Architektur wurde so gewählt, dass weitere Funktionalität möglichst einfach und weitgehend unabhängig von der bereits bestehenden hinzugefügt werden kann. Dies wird durch die Auftrennung der einzelnen Aufgaben in Threads erreicht. Jeder Thread stellt also einen bestimmten Dienst bereit. Zu diesen Diensten gehört die Initiierung und Kontrolle des Dateitransfers, des Videostreamings, der Anzeige der Inhalte und der Kommunikation. Ein Watchdog übernimmt das Starten, Überwachen und Beenden der Dienste. Um auf dem Master und Slave nicht unterschiedliche Software installieren zu müssen, wurde die gesamte Funktionalität in ein

<sup>1</sup>Wie z.B. Macromedia Flash Animationen

Voraussetzung	Gegenmassnahme	Für Demonstrator implementiert <sup>a</sup>
Kaum Paketverluste	Pakete mehrmals senden	Ja
Wenig Übertragungsfehler	Forward Error Correction	Nein
Variation der Verzögerung von Paketen auf den verschiedenen Pfaden möglichst klein	QoS im Netzwerk, Netzauslastung begrenzen <sup>b</sup>	Nein

<sup>a</sup>Durch Schalter im Code aktivierbar/anpassbar.

<sup>b</sup>Siehe auch Abschnitt 3.2.4

Tabelle 5.1: Voraussetzung für möglichst synchronen Meldungsempfang

einziges Programm integriert. Mittels Parameter kann dann gewählt werden, ob der Panel PC ein Master, ein Slave oder beides ist. Zur Kommunikation zwischen den einzelnen Threads werden POSIX Message Queues verwendet. Diese erlauben bei geeigneter Nutzung des selektiven Meldungsempfangs eine Priorisierung der Meldungen anhand ihrer Identifikationsnummer(ID). Zur Kommunikation mit dem Master/Slave dienen die MsgModule und die Receiver Threads. Die MsgModule fungieren als Sammel- und Sendeeinheit für Meldungen an den Master / an die Slaves via Ethernet. Die Receiver warten auf für sie bestimmte Meldungen aus dem Ethernet und leiten diese an die MessageModule weiter. Denkbar wäre auch, dass der Receiver bei sehr zeitkritischer Kommunikation die Meldungen direkt an die Message Queue des dafür zuständigen Threads schickt. Beim Einfügen eines neuen Dienstes müssten dann allerdings jeweils sowohl der Receiver, wie auch das MsgModule angepasst werden. Informationen zur Kommunikation über das Ethernet können im Abschnitt 5.2 gefunden werden.

Die Konfiguration der Dienste erfolgt über eine beim Starten eingelesene XML Datei. Die XML Datei kann nach der Erstellung mit Hilfe der zum Projekt gehörenden Document Type Definition(DTD) überprüft werden. Die Konfiguration der Slaves kann auch über den Austausch von Meldungen erfolgen<sup>2</sup>, ist aber momentan nicht so implementiert. Die Funktionsweise der in Abbildung 5.2 gezeigten Threads/Dienste wird nach einigen Erläuterungen zur Kommunikation über das Ethernet in den Abschnitten 5.3 und 5.4 primär durch eine Art Kontrollflussgraph beschrieben. Informationen zu den Konfigurationsoptionen für die jeweiligen Dienste sind im Anhang E zu finden. An einigen Stellen wird der erst im Abschnitt 7 vorgestellte PlanerApp Dienst erwähnt. Dieser Dienst übernimmt die Planung der abzuspielenden Inhalte und benutzt die Kommunikations- und Kontrollebene zur Umsetzung des Plans.

## 5.2 Kommunikation über das Ethernet

Die Kommunikation über das Ethernet erfolgt durch in UDP Paketen verschickte Meldungen. Der Master versendet die UDP Pakete per Multicast an die Slaves. Die Slaves hingegen verwenden Unicast für den Versand von Meldungen an den Master. UDP erlaubt das Verschicken von Paketen an Multicast Adressen, hat aber den Nachteil, dass es kein Protokoll für verlässlichen Datentransport ist. Die Detektion von Paketverlusten und deren Behandlung bleibt dem Benutzer überlassen. Die Verwendung von TCP kommt aber nicht in Frage, da der Empfangszeitpunkt einer Meldung durch den Aufbau der TCP Verbindungen mit anschliessendem Meldungsaustausch mit jedem Slave einer grösseren Variabilität unterzogen ist als bei UDP. Der möglichst gleichzeitige Empfang der Meldung durch alle Slaves ist aber für die Synchronität der Anzeigen wichtig. Allerdings kann auch mit UDP und Multicast nur unter gewissen Voraussetzungen ein möglichst synchroner Empfang der Meldungen erreicht werden. Tabelle 5.1 nennt drei dieser Voraussetzungen mit entsprechenden Gegenmassnahmen bei deren Nichterfüllung. Der Tabelle 5.1 ist zu entnehmen, dass ein mehrmaliges Senden der Pakete für den Demonstrator implementiert wurde. Dieses Feature ist aber standardmässig deaktiviert. Aktivieren kann

<sup>2</sup>Für diese Art der Konfiguration müsste fest einprogrammiert sein, wie diese Meldungen zu empfangen sind. Das Versenden und Empfangen über die Multicast-Broadcast Adresse 224.0.0.1 wäre ein gangbarer Weg.

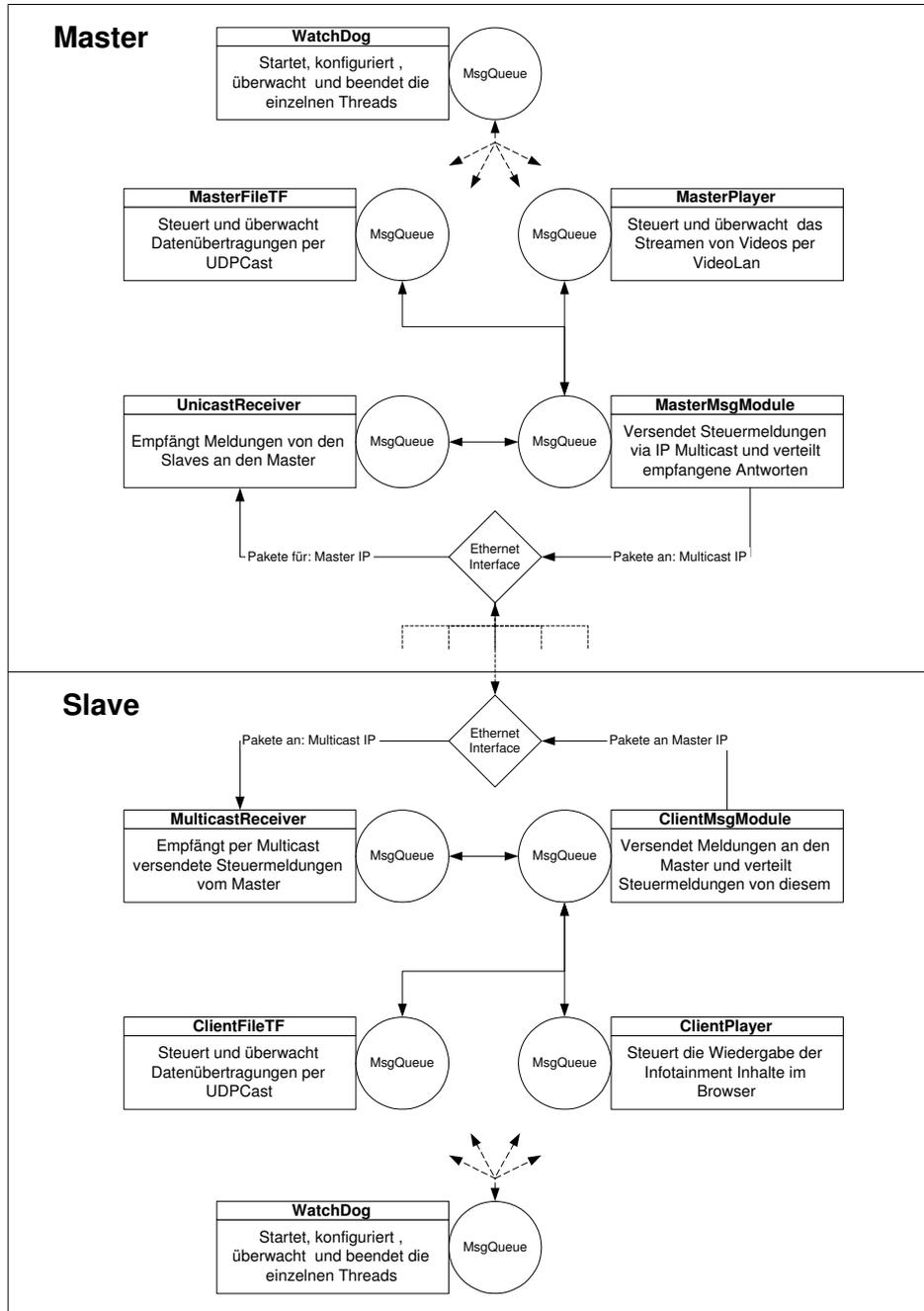


Abbildung 5.2: Software Architektur der Kommunikations- und Kontrollebene

Bits: 0..31	32..63	64..95	96..96+8× <i>MsgLength</i>
SeqNum, Sequenznummer	MsgLength, Länge der Meldung in Bytes	ID der Meldung	Meldung

Tabelle 5.2: Verwendetes Format für die im UDP Paket transportierten Daten

man es durch die Definition<sup>3</sup> der Anzahl zu versendenden Pakete pro Meldung grösser als eins. Die Empfängerseite muss also erkennen können, ob eine bestimmte Meldung bereits empfangen wurde oder nicht. Aus diesem Grund wird jeder Meldung eine Sequenznummer hinzugefügt. Der Empfänger speichert dann in einer für jede Sender IP Adresse nach dem First-In-First-Out(FIFO) Prinzip geführten Liste maximal MAX\_SEQ\_NUM\_LISTSIZE unterschiedliche Sequenznummern ab. Ist die Sequenznummer in der Liste für den Absender des gerade empfangenen Pakets bereits enthalten, wird das Paket verworfen.

### 5.2.1 Massnahmen zur Erkennung von fehlgeleiteten Paketen

Es besteht die Möglichkeit, dass UDP Pakete von anderen Komponenten im Ethernet irrtümlich an die vom IS zur Kommunikation verwendeten IP/Port Kombinationen gesendet werden. Diese Pakete enthalten dann irgendwelche Daten, welche fälschlicherweise als Meldung interpretiert würden. Die Wahrscheinlichkeit einer Fehlinterpretation kann aber durch das Anfügen von zusätzlichen Informationen zur Meldung massiv verkleinert werden. Die Strukturierung der im UDP Paket transportierten Daten zeigt Tabelle 5.2. Die Sequenznummer im SeqNum Feld dient nicht der Erkennung von Paketen mit fremdem Inhalt, sondern, wie bereits erläutert, der Elimination von Duplikaten. Das MsgLength Feld gibt die Länge des Feldes Meldung in Bytes an. Mit Hilfe des MsgLength Feldes kann bereits mit ziemlicher Sicherheit gesagt werden, ob dies ein fremdes oder ein eigenes UDP Paket ist. Wird nämlich zur Längenangabe im MsgLength Feld die Länge der drei der eigentlichen Meldung vorangehenden Feldern hinzugezählt, muss das Resultat der Anzahl vom Socket gemeldeten Datenbytes für das UDP Paket entsprechen. Bei einem fremden UDP Paket ist eine Übereinstimmung dieser beiden Werte sehr unwahrscheinlich. Als weitere Sicherheit wird auch noch die ID der Meldung geprüft. Entspricht diese nicht einer gültigen ID, wird das Paket ebenfalls verworfen. Diese Massnahmen schützen natürlich nicht vor einer Störung durch mutwillig im selben Format zugesandten UDP Paketen.

### 5.2.2 Absichern der Kommunikation gegen Angreifer

Eine Absicherung der Kommunikation zwischen dem Master und den Slaves hat niedrige Priorität, da das Ethernet im Zug für nicht autorisierte Personen kaum zugänglich ist. Falls eine Absicherung nötig werden sollte, wird vorgeschlagen, jedes UDP Paket mit einem Zeitstempel, sowie einer Sequenznummer zu versehen und dieses anschliessend digital zu signieren. Der Empfänger überprüft beim Empfangen die Signatur, ob der Zeitstempel innerhalb eines definierten Zeitrahmens liegt und ob die Sequenznummer korrekt ist. Die Signatur schützt gegen gefälschte Meldungen, der Zeitstempel und die Sequenznummer gegen Replay Attacks. Die Sequenznummer ist notwendig, da sonst innerhalb des Zeitrahmens, in dem der Zeitstempel gültig ist, Replay Attacks möglich wären. Der Zeitrahmen ist also maximal so gross zu wählen, dass in dieser Zeit nicht X Pakete empfangen werden können. X bezeichnet hierbei die Anzahl möglicher Sequenznummern.

<sup>3</sup>Via #define NUM\_OF\_PKG\_COPIES Anweisung im Source Code der UDPSocket Klasse. Erneutes Kompilieren notwendig.

## 5.3 Dienste auf dem Master

### 5.3.1 Der MasterMsgModule Dienst

Dieser Dienst prüft die an seine Message Queue gesendeten Meldungen und leitet sie entweder per Multicast<sup>4</sup> via Ethernet an die Slaves oder schickt sie an die Message Queue des dafür zuständigen Dienstes. Die Meldungen für die Slaves werden an eine in der Konfigurationsdatei definierte Multicast IP geschickt. Der Master versendet ausserdem in bestimmten Abständen eine Meldung, in der er seine IP mitteilt. Neu hinzugekommene Slaves können sich dann beim Master per Unicast anmelden und eventuell benötigte Parameter erfragen. Abbildung 5.3 zeigt die Funktionsweise des MasterMsgModule Dienstes.

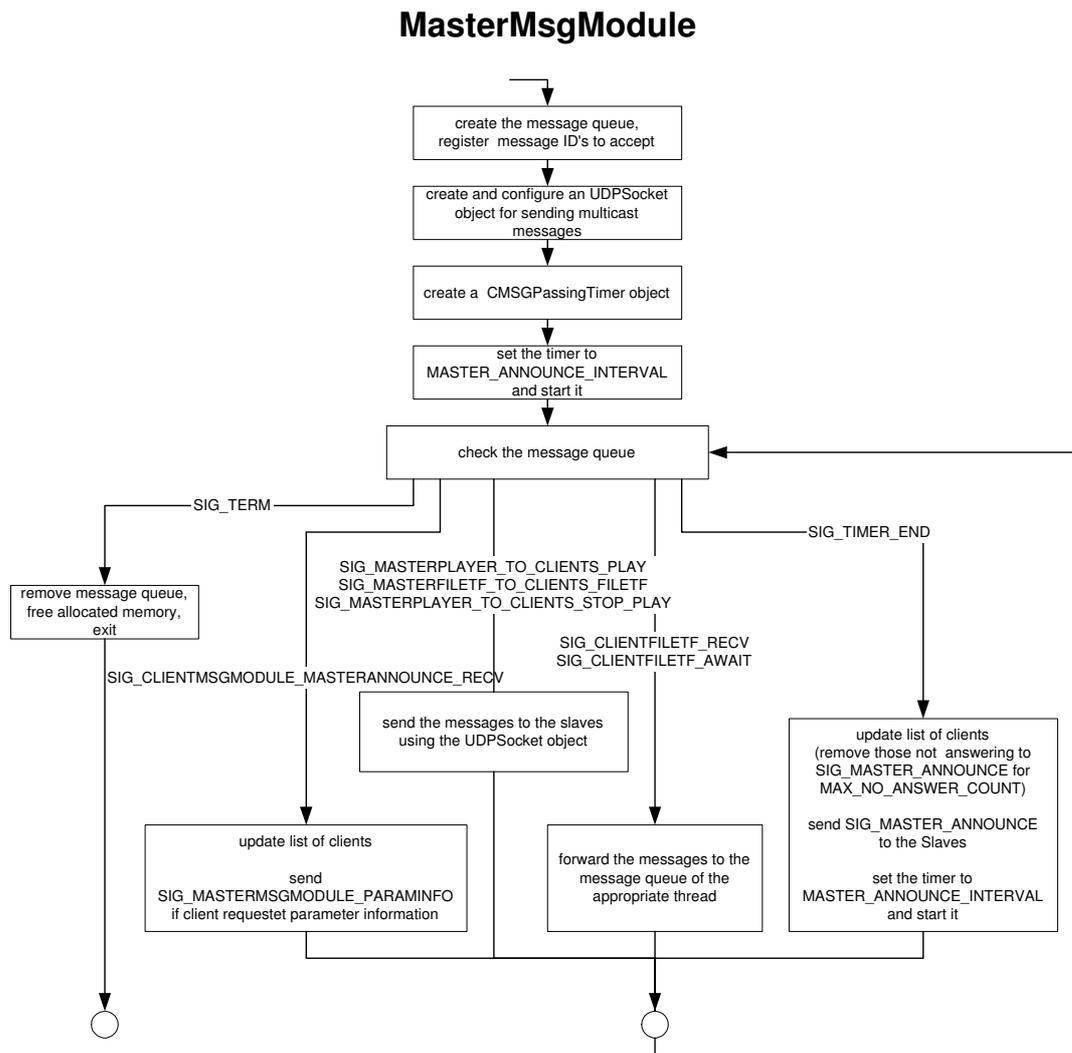


Abbildung 5.3: Funktionsweise des MasterMsgModule Dienstes

### 5.3.2 Der MasterFileTF Dienst

Dieser Dienst steuert die Übertragung von Dateien vom Master zu den Slaves unter Benützung der Software UDPCast. Dazu wartet der Dienst auf den Befehl zur Initiierung eines Transfers von der PlanerApp. Die PlanerApp muss sicherstellen, dass die zu transferierende Datei auch tatsächlich vorhanden ist. Die PlanerApp teilt dem Dienst auch mit, wie lange dieser Transfer

<sup>4</sup>Oder auch per Unicast an einzelne Slaves

maximal dauern darf. Dauert er länger, meldet der Dienst den Fehlschlag des Transfers an die PlanerApp zurück. Nach der Anforderung eines Transfers durch die PlanerApp informiert der MasterFileTF Dienst die Slaves via MasterMsgModule über den anstehenden Transfer. Erst wenn diese ihre Bereitschaft zum Empfangen der Datei zurückmelden, wird mit dem Transfer begonnen. Momentan ist dies bereits bei Rückmeldung eines einzigen Slaves der Fall. Die Einführung einer Mindestanzahl von z.B.  $n-2$  Rückmeldungen bei  $n$  vorhandenen Slaves würde nur eine kleine Änderung des Source Codes erfordern. Von einer Implementierung dieses Feature wurde abgesehen, weil der Master keine Information über die tatsächliche Anzahl vorhandener Slaves hat, sondern nur die Anzahl beim MasterMsgModule angemeldeter Slaves kennt. Die tatsächliche Anzahl Slaves kennt nur das noch nicht an den Demonstrator angebundene RAILVOX. Abbildung 5.4 zeigt die Funktionsweise des MasterFileTF Dienstes.

### 5.3.3 Der MasterPlayer Dienst

Dieser Dienst steuert alle für das Abspielen von Infotainment Inhalten notwendigen Vorgänge. Der Dienst erhält den Befehl zum Abspielen eines Inhaltes von der PlanerApp. Er prüft, ob der Inhalt vom Typ Stream ist. Falls ja, sendet er die zum Abspielen notwendigen Kommandos an die VLC Software. Unabhängig vom Inhalt informiert er anschliessend via MasterMsgModule die Slaves über den abzuspielenden Inhalt. Weitere Details, wie das Führen einer internen Playlist für die zur VLC Playlist hinzugefügten Inhalte, gehen aus dem Source Code zum MasterPlayer Dienst hervor. Das zur Steuerung der VLC Software via HTTP Interface programmierte VLC-Controller Objekt ist ebenfalls nur durch den Source Code dokumentiert. Abbildung 5.5 zeigt die Funktionsweise des MasterPlayer Dienstes.

### 5.3.4 Der UnicastReceiver Dienst

Dieser Dienst ist für das Empfangen von Meldungen von den Slaves an den Master, sowie für das Weiterleiten dieser Meldungen an den MasterMsgModule Dienst zuständig. Die Kommunikation mit den Slaves erfolgt über das Ethernet. Abschnitt 5.2 lieferte bereits die relevanten Details dazu. Abbildung 5.6 zeigt die Funktionsweise des UnicastReceiver Dienstes.

### MasterFileTF

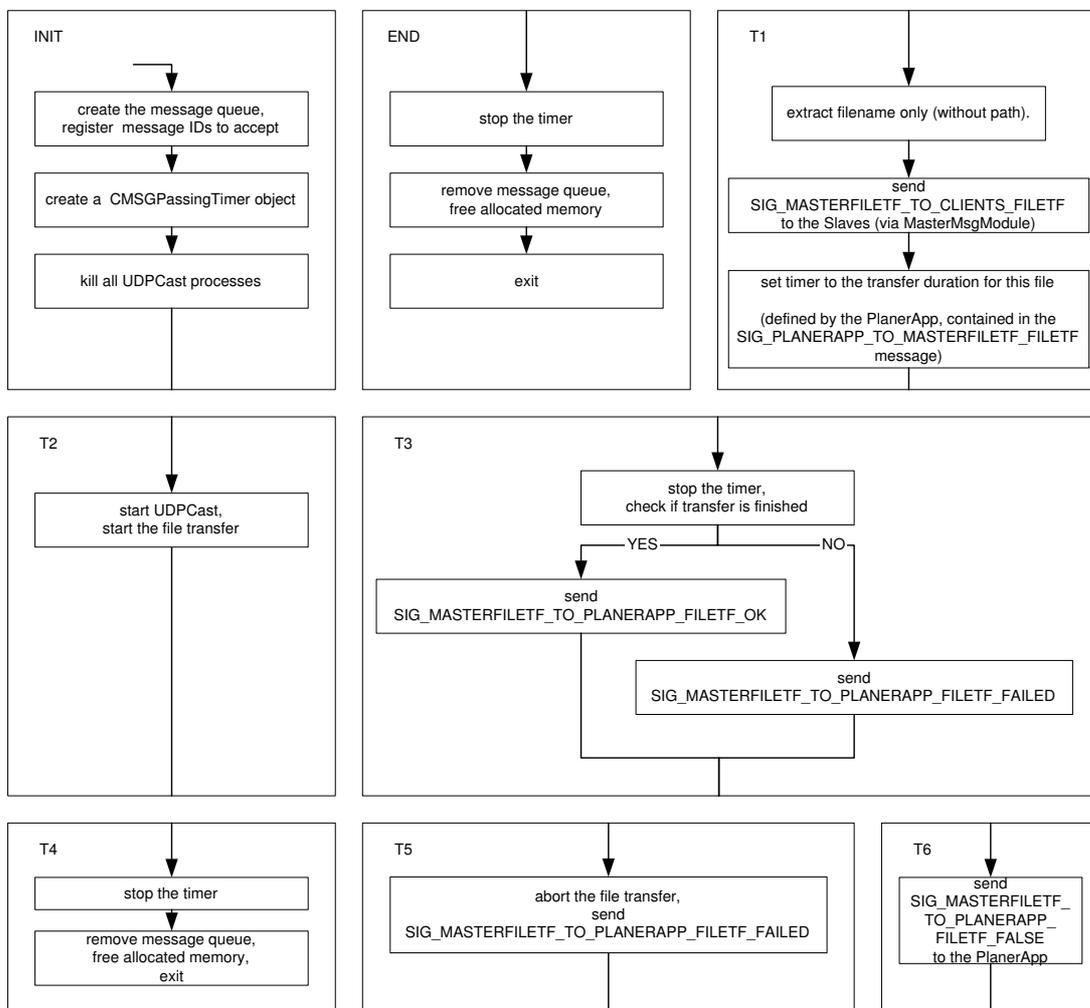
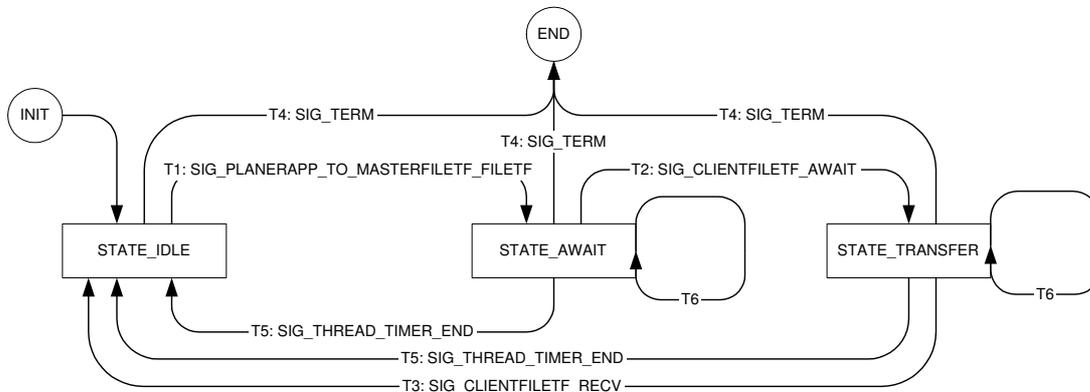


Abbildung 5.4: Funktionsweise des MasterFileTF Dienstes

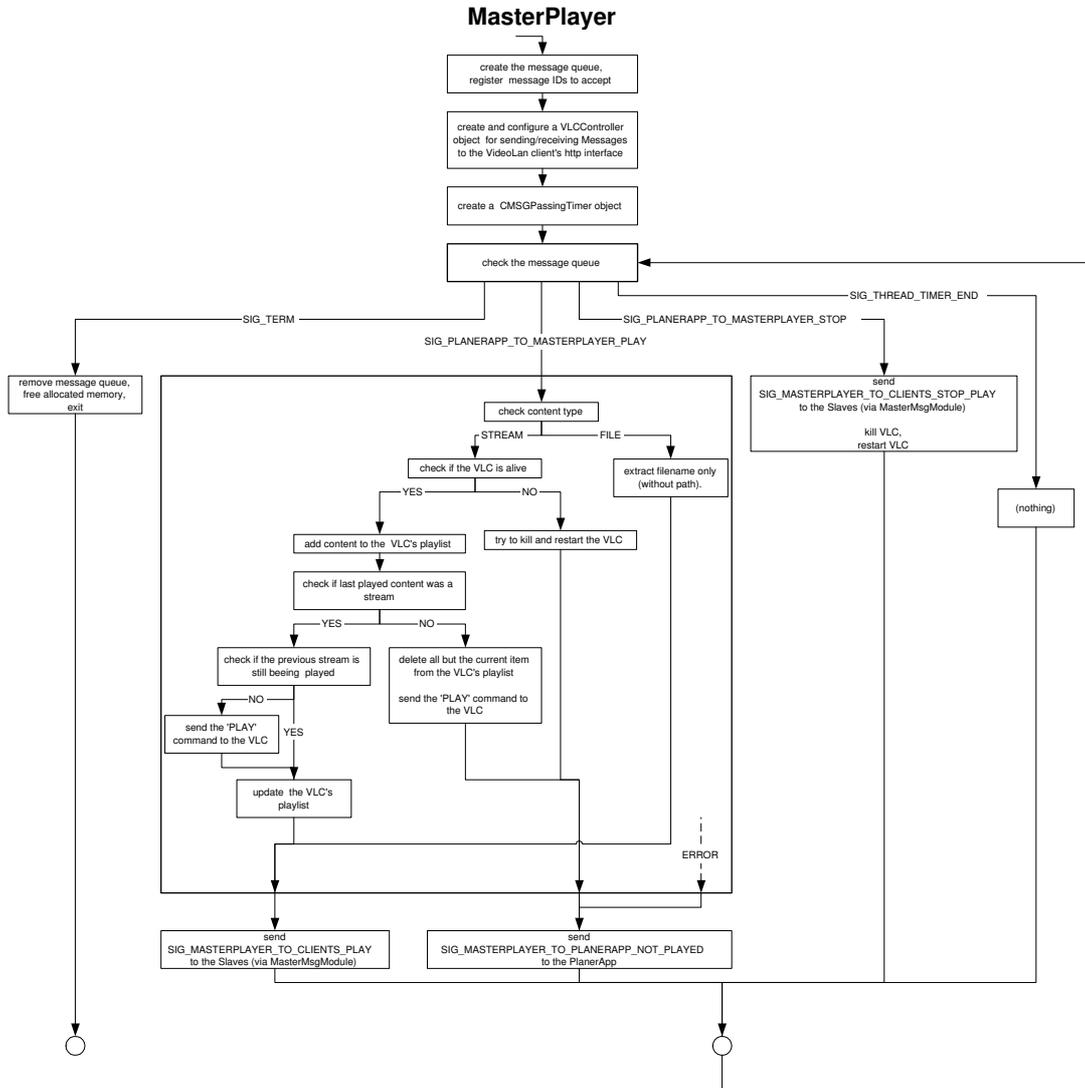


Abbildung 5.5: Funktionsweise des MasterPlayer Dienstes

## MulticastReceiver

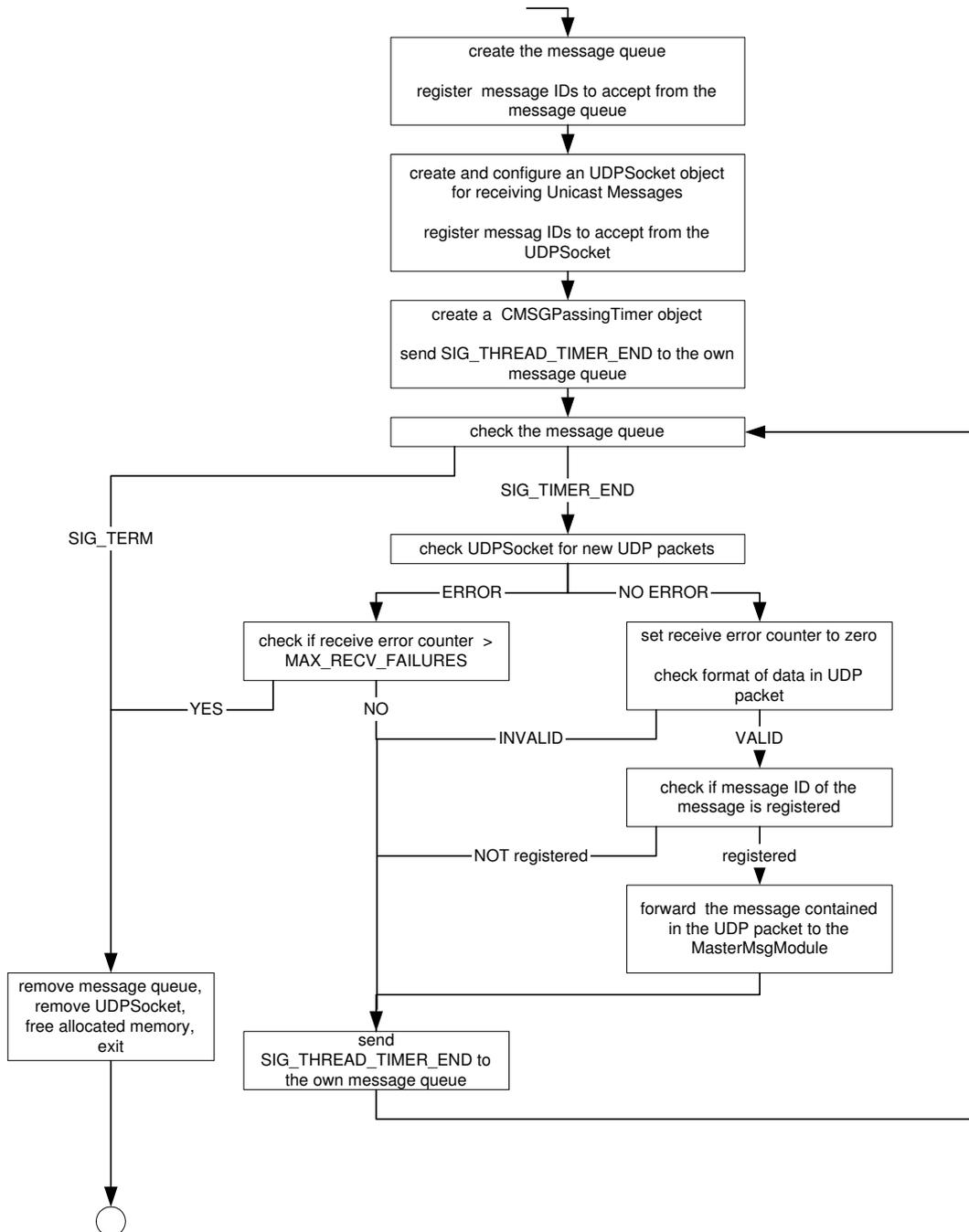


Abbildung 5.6: Funktionsweise des UnicastReceiver Dienstes

## 5.4 Dienste auf den Slaves

### 5.4.1 Der ClientMsgModule Dienst

Dieser Dienst prüft die an seine Message Queue gesendeten Meldungen und leitet sie entweder per Unicast via Ethernet an den Master oder schickt sie an die Message Queue des dafür zuständigen Dienstes. Die IP des Masters erfährt der Dienst durch die vom Master in bestimmten Abständen per Multicast verschickten und vom MulticastReceiver Dienst aufgefangenen SIG\_MASTERMSGMODULE\_MASTERANNOUNCE Meldungen. Nachdem die IP des Masters bekannt ist, kann der Dienst einige Konfigurationsdaten anfordern und nach deren Erhalt in den regulären Betrieb übergehen. Wie bereits erwähnt, werden bei der gegenwärtigen Implementation der Kommunikations- und Kontrollebene die meisten Konfigurationsdaten nicht übertragen, sondern aus einer Datei eingelesen. Falls erforderlich, ist der Verzicht auf die Konfigurationsdatei auf den Slaves dank dem eben beschriebenen Mechanismus zum Austausch von Konfigurationsdaten aber ohne weiteres möglich. Der in Abschnitt 5.4.3 beschriebene ClientPlayer Dienst dient als Beispiel für diese Art der Konfiguration. Seine Sektion in der Konfigurationsdatei ist leer. Abbildung 5.6 zeigt die Funktionsweise des ClientPlayer Dienstes.

### 5.4.2 Der ClientFileTF Dienst

Dieser Dienst steuert den Empfang der vom Master per UDPCast versendeten Dateien. Die Dateien werden in einem über die Konfigurationsdatei spezifizierbaren Ordner abgelegt. Der Ordner befindet sich beim Demonstrator auf einer RAMdisk,<sup>5</sup> um Lese- und Schreibvorgänge zu beschleunigen, sowie um in Zukunft die zur Speicherung der Software verwendete Compact-Flash Karte nicht zu belasten. Ebenfalls über die Konfigurationsdatei wird der für die Speicherung der Dateien zur Verfügung stehende Speicherplatz definiert. Vor jeder Übertragung muss dann der Dienst prüfen, ob genügend Speicherplatz vorhanden ist. Falls nicht, löscht er ohne Rücksicht darauf, ob die Dateien bereits abgespielt wurden, nach dem FIFO Prinzip die zuvor übertragenen Dateien. Um garantieren zu können, dass innerhalb der Spieldauer einer Datei immer mindestens Dateien für wieder die gleiche Spieldauer übertragen werden können, darf keine Datei, die grösser ist als die Hälfte des insgesamt zur Verfügung stehenden Speichers, angenommen werden. Zusätzlich darf die Übertragungszeit für eine Datei sicher nicht grösser sein als ihre Spieldauer. Die Angabe der für eine Datei maximal zulässigen Transferzeit erfolgt durch die PlanerApp.

### 5.4.3 Der ClientPlayer Dienst

Dieser Dienst ist für das Anzeigen der durch den Master angekündigten Inhalte zuständig. Da die Inhalte gemäss dem Multicast:Multicast Konzept im Mozilla Browser anzuzeigen sind, muss der ClientPlayer den Browser fernsteuern können. Wie diese Fernsteuerung genau aussieht, kann im Abschnitt 6.2 nachgelesen werden. Abbildung 5.6 zeigt die Funktionsweise des ClientPlayer Dienstes.

### 5.4.4 Der MulticastReceiver Dienst

Dieser Dienst ist für das Empfangen von Meldungen vom Master, sowie für das Weiterleiten dieser Meldungen an den ClientMsgModule Dienst zuständig. Die Kommunikation mit dem Master erfolgt über Unix Sockets. Abschnitt 5.2 lieferte bereits die relevanten Details dazu. Abbildung 5.10 zeigt die Funktionsweise des MulticastReceiver Dienstes. Sie ist bis auf die Initialisierung des Sockets identisch zu Abbildung 5.6 für den UnicastReceiver Dienst.

<sup>5</sup>Wird eine RAMdisk im RAM angelegt, kann der dafür reservierte Speicher wie eine Festplatte verwendet werden.

## ClientMsgModule

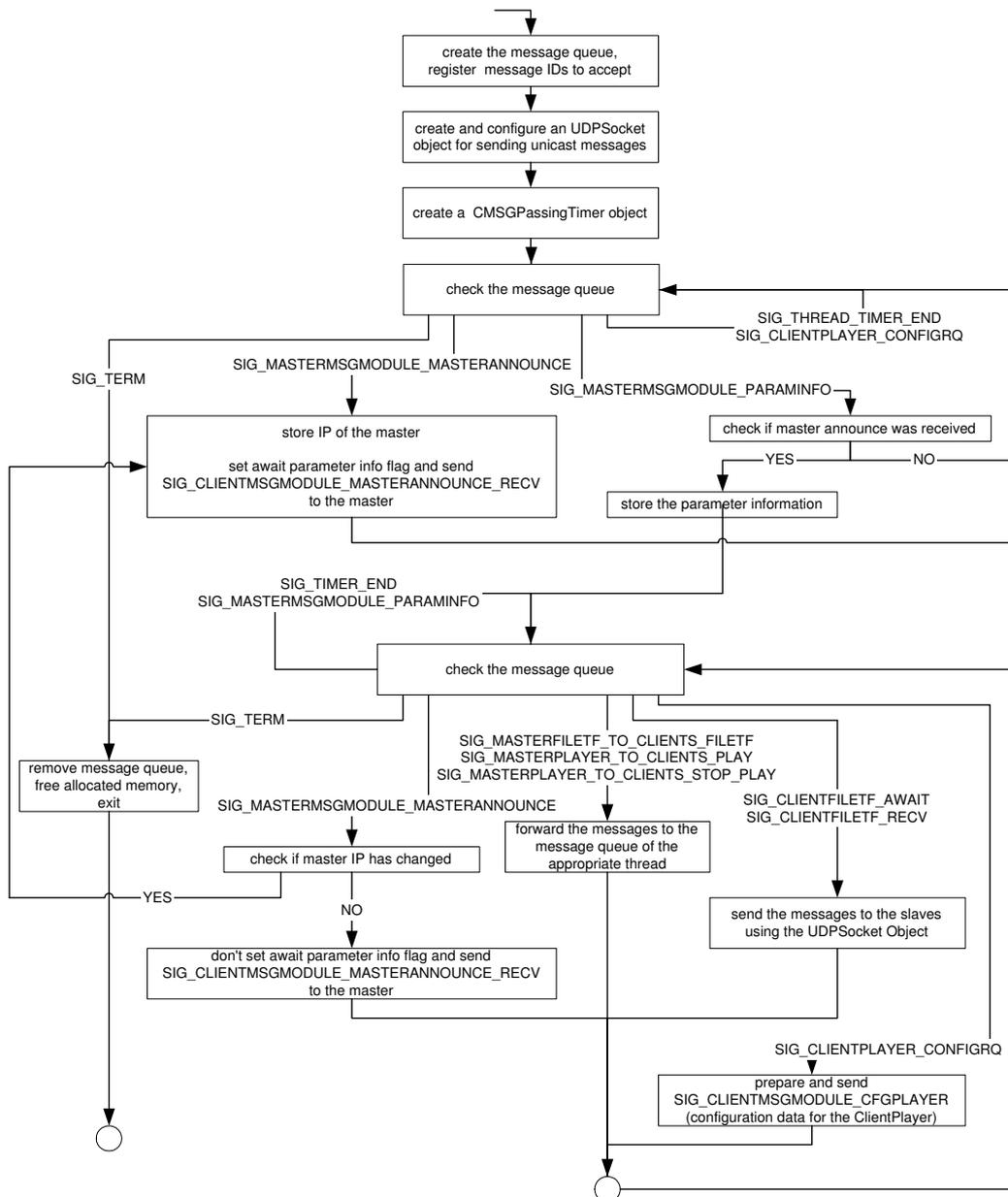


Abbildung 5.7: Funktionsweise des ClientMsgModule Dienstes

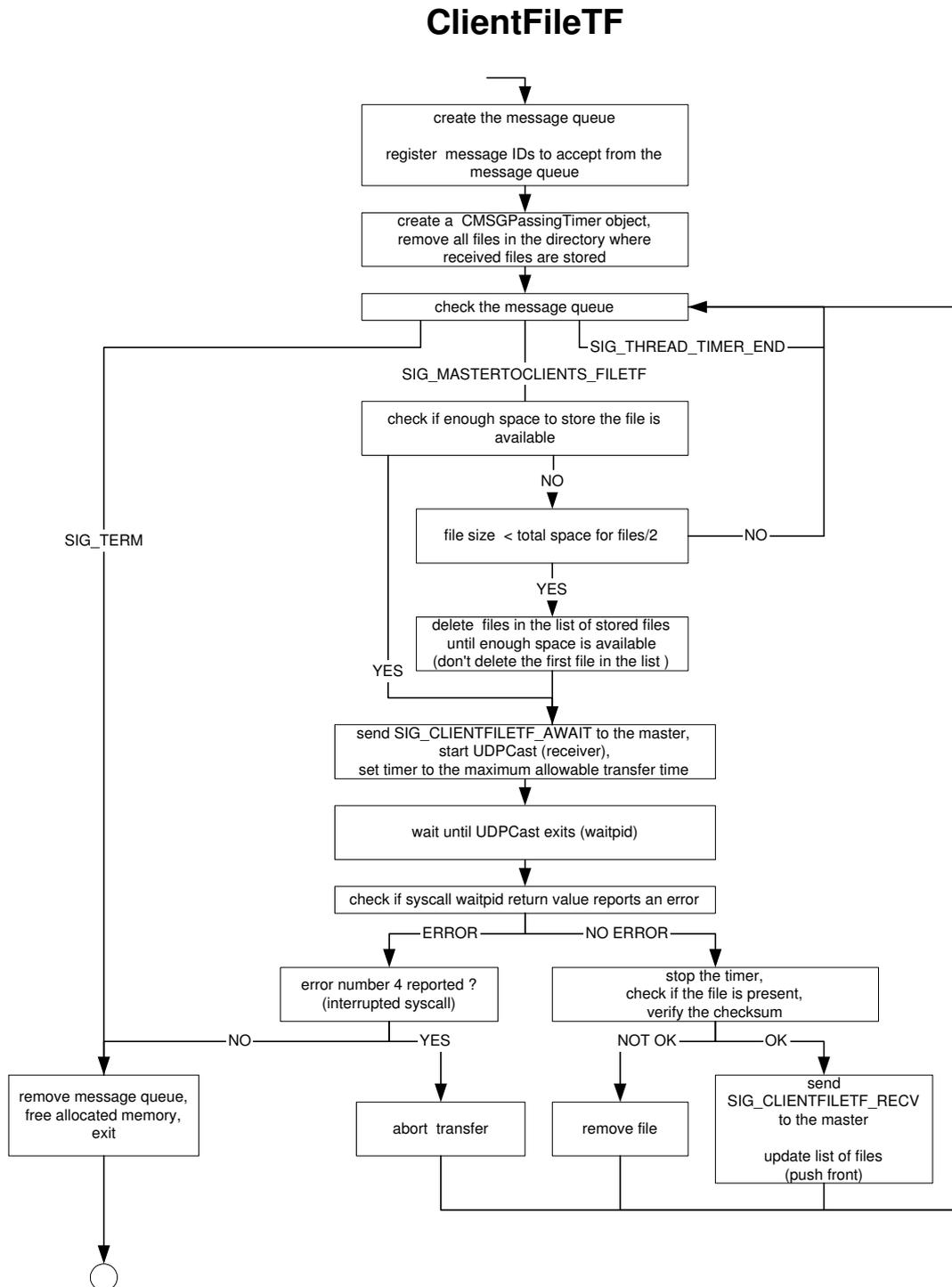


Abbildung 5.8: Funktionsweise des ClientFileTF Dienstes

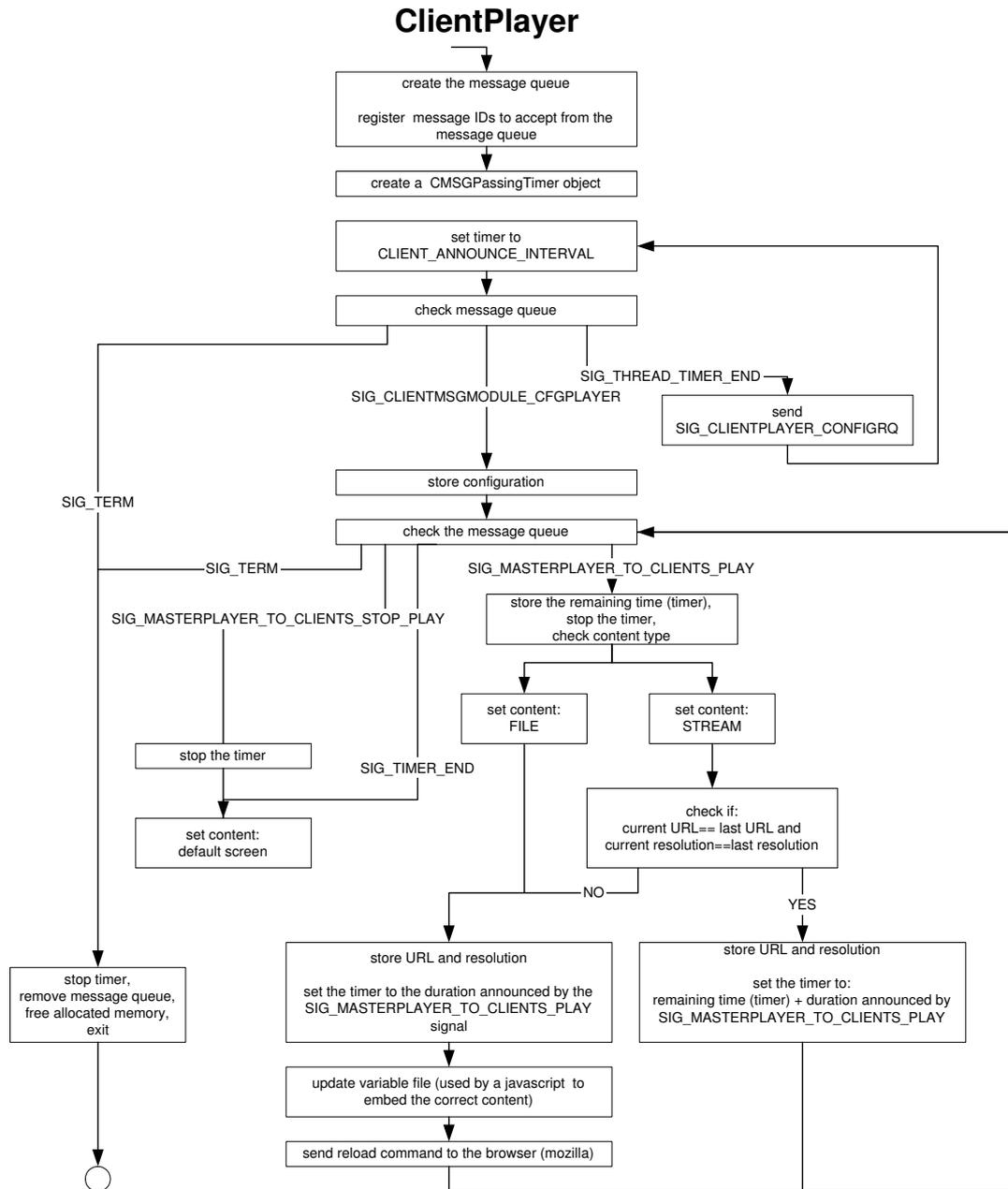


Abbildung 5.9: Funktionsweise des ClientPlayer Dienstes

## MulticastReceiver

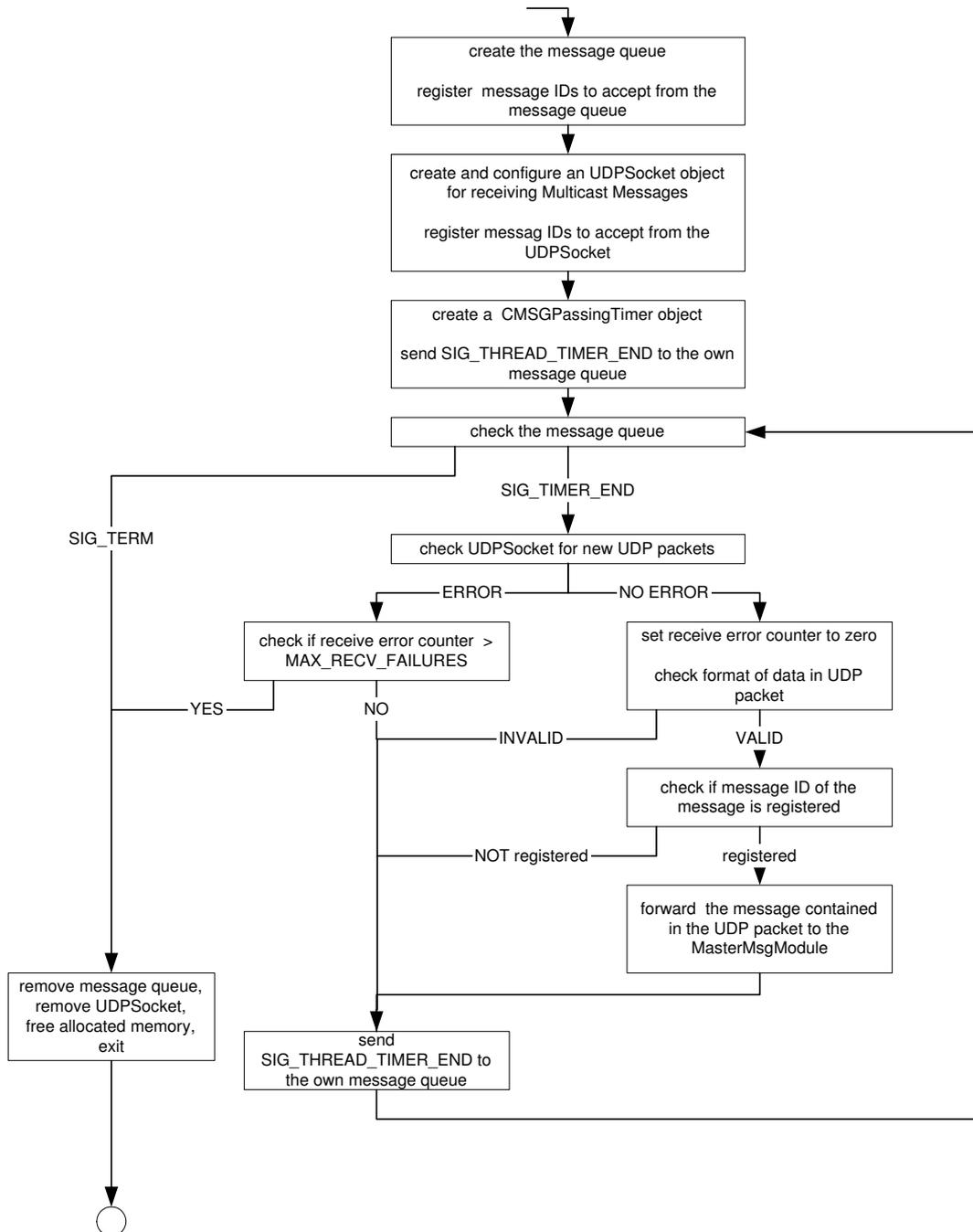


Abbildung 5.10: Funktionsweise des MulticastReceiver Dienstes

## 5.5 Vom Master und Slave benötigte Dienste

### 5.5.1 WatchDog

Dieser Dienst liest die Konfigurationsdatei ein, konfiguriert und startet abhängig vom Modus (MASTER, SLAVE oder BOTH) die entsprechenden Dienste(Threads) und überwacht diese anschliessend. Der WatchDog geht davon aus, dass ein Dienst im Fehlerfall korrekt terminiert und somit seine Message Queue per Systemaufruf löscht. Hängt sich ein Dienst auf, wird dies vom Watchdog nicht bemerkt. Die Einführung einer regelmässig durch den WatchDog an die Dienste versendete und durch diese zu beantwortende Meldung könnte solche Hänger aber aufdecken. Sowohl der WatchDog wie auch alle anderen Dienste sind dafür bereits vorbereitet.

Wenn sich ein Dienst aufhängt, gibt es mehrere Probleme. Erstens kann seine Message Queue nun nicht mehr so ohne Weiteres gelöscht werden,<sup>6</sup> und zweitens wird der WatchDog den Dienst nicht mehr korrekt beenden können. Daher muss der WatchDog sich selbst und damit auch den IS Prozess beenden. Ein Skript sorgt anschliessend dafür, dass alle noch existierenden Message Queues gelöscht und der IS Prozess neu gestartet wird. Ein Neustart kann auch durch das Senden des Signals SIG\_ABORT<sup>7</sup> an den IS Prozess erzwungen werden. Abbildung 5.11 zeigt die Funktionsweise des WatchDog Dienstes.

---

<sup>6</sup>Message Queues werden durch Systemaufrufe erzeugt/gelöscht. Eine Art Schlüssel ist zur Identifikation der richtigen Message Queue notwendig. Sinnvollerweise kennt nur der Erzeuger diesen Schlüssel.

<sup>7</sup>z.B. durch Ctrl+C ausgelöst

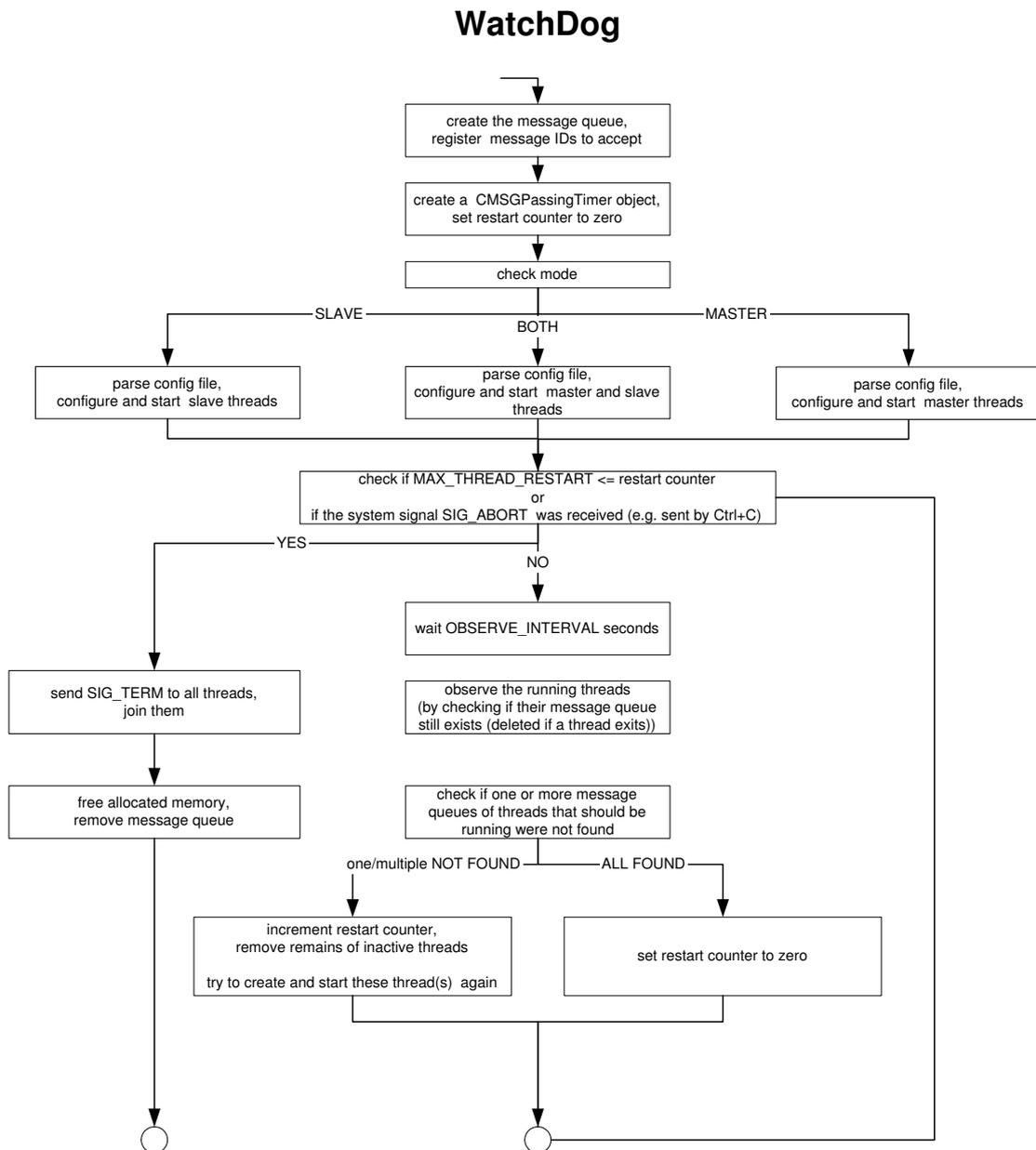


Abbildung 5.11: Funktionsweise des WatchDog Dienstes

## Kapitel 6

# Darstellung von Fahrgastinformation und Unterhaltung

Nach der Realisierung der Verteilung der Inhalte, der Kommunikation zwischen dem Master und den Slaves und der Kontrolle der Programme zur Anzeige der Inhalte muss definiert werden, wann FI und wann Multimedia Inhalte(MI) gezeigt werden und wie dies geschieht. Das im nächsten Abschnitt vorgestellte Ebenenmodell dient zusammen mit einigen konkreten Anwendungen der Beantwortung der Frage nach dem Wann. Die Frage nach dem Wie wird in der anschließenden Beschreibung der Umsetzung für den Demonstrator beantwortet.

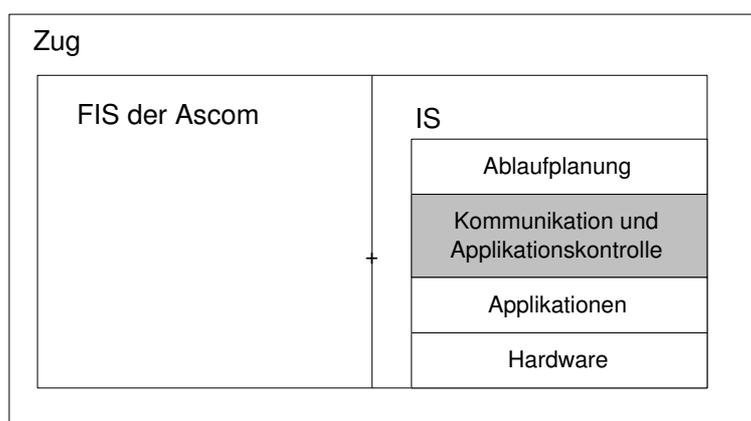


Abbildung 6.1: Betrachtete Systemebene: Kommunikation und Applikationskontrolle

### 6.1 Das Ebenenmodell

Das zur Anzeige von Informationen verwendete Modell basiert auf mehreren koexistierenden Ebenen denen unterschiedliche Prioritäten zugewiesen werden können. Es werden immer die Ebenen angezeigt, welche erstens Informationen enthalten und zweitens die höchste Priorität besitzen. Falls es Ebenen gibt, welche ebenfalls Informationen enthalten und die die bereits gezeigten Ebenen nicht überschneiden, wird von diesen wiederum diejenige mit der höchsten Priorität gewählt. Das Einfügen der FI in eine Ebene mit genügend hoher Priorität kurz vor der Einfahrt in den nächsten Bahnhof bewirkt somit, dass der Bahnkunde bei der Einfahrt nicht den neusten Trailer für einen Kinofilm, sondern Informationen zum nächsten Halt zu sehen bekommt. Die Zuweisung der FI zur entsprechenden Ebene erfolgt z.B. anhand von Ereignissen wie 'Einfahrt in Bahnhof X' oder 'Ausfahrt aus Bahnhof X'. Um bei einer Überlappung der FI Ebe-

ne(n) mit der MI Ebene den Wechsel von MI zu FI inmitten eines Videoclips zu vermeiden, muss bereits zum Startzeitpunkt des Videoclips bekannt sein, wann die MI Ebene voraussichtlich von FI überdeckt werden wird. So kann dann ein Videoclip gezeigt werden, der in diese Zeitspanne hineinpasst. Im Folgenden werden verschiedene, auf dem Ebenenmodell basierende Darstellungsmöglichkeiten der Infotainment Inhalte vorgestellt. Allenfalls notwendige Erweiterungen des RAILVOX werden ebenfalls beschrieben.

### 6.1.1 Lösungen für verschiedene Anordnungen der Ebenen

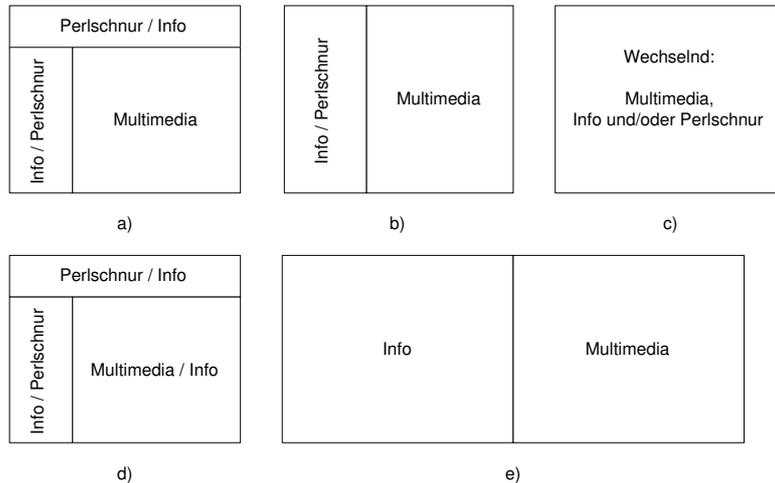


Abbildung 6.2: Präsentationsarten für das Infotainment

Abbildung 6.2 zeigt mögliche Anordnungen der Ebenen. Die Varianten a) und b) erlauben das gleichzeitige Darstellen von FI und Multimedia, Variante c) zeigt nur jeweils entweder FI oder Multimedia Inhalte an. Variante d) kann im Gegensatz zu a) und b) durch Überlagerung der MI Ebene mit einer FI Ebene FI auch im grossen Fenster anzeigen. Variante e) zeigt eine Zweischirmlösung, wie sie z.B. in der Berliner U-Bahn eingesetzt wird. Wenn der Kunde bereit ist, die Mehrkosten für die Zweischirmlösung zu übernehmen, ist dies bezüglich Übersichtlichkeit und Implementationsaufwand wohl die optimale Variante. Sie soll hier aber nicht weiter berücksichtigt werden, da von einem Panel PC mit nur einem Bildschirm ausgegangen wird.

Falls sich die FI und MI Ebenen nicht überschneiden, ist eine unabhängige Steuerung der FI und der MI möglich. Dies ist bei den Varianten a), b) und e) der Fall. Weil für das FIS der Ascom bereits eine Steuerung zur Darstellung von FI in unterschiedlichen Browserfenstern existiert, muss für das IS also nur noch die Regelung der Abspielreihenfolge der MI definiert werden. Auf die Varianten c) und d) trifft dies leider nicht zu, da sich die FI und MI Ebenen überschneiden. Abhängig von den Anforderungen müssen hierfür spezielle Lösungen entwickelt werden.

Für folgende Anforderungen werden Lösungen vorgestellt:

- Fahrgastinformation spätestens kurz vor dem nächsten Halt anzeigen. Abbrechen von Videoclips oder Animationen erlaubt. Anzeige von MI bei erneuter Aufnahme der Fahrt.
- Fahrgastinformation spätestens kurz vor dem nächsten Halt anzeigen. Abbrechen von Videoclips oder Animationen nur in Ausnahmefällen. Anzeige von MI bei erneuter Aufnahme der Fahrt.

**Abbrechen von Videoclips und Animationen erlaubt:** Die FI hat in jedem Fall Priorität gegenüber den MI. Der Entscheid, ob FI oder MI angezeigt werden, muss also von derjenigen Stelle kommen, welche auch die FI steuert. Da diese Stelle der REC ist, muss die Information, ob nun FI oder MI höhere Priorität hat, über das Ethernet an den Master Panel PC geschickt werden. Welcher Inhalt höhere Priorität besitzt, wird anhand der vom REC generierten Signalen 'Einfahrt in den Bahnhof' und 'Ausfahrt aus dem Bahnhof' bestimmt. Da das Abbrechen

von Videoclips und Animationen erlaubt ist, reicht es im Prinzip aus, die Software des REC so abzuändern, dass diese Signale an den Master Panel PC weitergeleitet werden. Sofort nach Erhalt des Signals 'Einfahrt in den Bahnhof' könnte dann das Abspielen von MI gestoppt und FI angezeigt werden. Bei Erhalt des Signals 'Ausfahrt aus dem Bahnhof' würden dann wieder MI gezeigt.

**Abbrechen von Videoclips und Animationen nur im Ausnahmefall:** Damit ein Videoclip nicht wie im vorderen Lösungsvorschlag bei der Einfahrt in den Bahnhof einfach unterbrochen wird, muss vor dem Abspielen eines Videoclips bekannt sein, ob die Zeit hierfür noch ausreicht. Falls nicht, wird ein kürzerer oder gar kein Videoclip gezeigt. Die voraussichtliche Zeit bis zum Eintreffen im nächsten Bahnhof wird momentan vom RAILVOX nicht berechnet. Verfügbar sind aber die Daten zur Distanz zwischen zwei Bahnhöfen, sowie der aktuellen Position des Fahrzeuges. Basierend auf diesen Daten könnte sehr wohl eine Schätzung der Zeit bis zum Eintreffen im nächsten Bahnhof gemacht werden. Ohne einige tiefer gehenden Änderungen an der Software für den REC ist dies allerdings nicht möglich.

## 6.2 Das Ebenenmodell auf dem Demonstrator

Aus dem vorangehenden Abschnitt geht hervor, dass für Lösungen mit sich überschneidenden Darstellungsebenen für FI und MI Änderungen an der Software für den REC notwendig sind. Von Seiten der Ascom ist noch nicht näher geprüft worden, ob und welche der beiden vorgeschlagenen Varianten allenfalls auf dem REC implementiert werden sollten. Eine oder beide Varianten selbst zu implementieren, kam aufgrund der zur Einarbeitung in das Echtzeitbetriebssystem OSE<sup>1</sup>, sowie das Softwareframework für den REC benötigten Zeit nicht in Frage. Eine Programmierung durch Entwickler der Ascom wird erst erfolgen, wenn der Entscheid für die Weiterentwicklung, sowie ein definitives Design des Systems gefällt worden ist. Es machte aus diesem Grund keinen Sinn, für den Demonstrator eine Variante mit sich überschneidenden FI und MI Ebenen zu implementieren. Deshalb wurde die Variante a) aus Abbildung 6.2 mit sich nicht überschneidenden FI und MI Ebenen zur Realisierung auf dem Demonstrator ausgewählt. Wie aus Abschnitt 6.1.1 hervorgeht, existiert bereits ein Interface zur Steuerung der FI durch den REC, und es muss nur noch die Software zur Ablaufplanung der MI auf den Panel PCs entwickelt werden. Kapitel 7 schlägt Algorithmen zur Realisierung der beiden im Abschnitt 6.1.1 genannten Varianten mit und ohne Unterbrechung von Videoclips vor und beschreibt den auf dem Demonstrator eingesetzten Algorithmus. Nun zur Umsetzung der Variante a) auf dem Demonstrator.

### 6.2.1 Realisierung des Ebenenmodells

Da laut Konzept die Anzeige der Inhalte innerhalb eines Webbrowsers erfolgen soll und der ClientPlayer für das Abspielen der MI zuständig ist, muss dieser den Webbrowser auf irgendeine Weise fernsteuern können. Eine Möglichkeit ist das Schreiben einer Erweiterung für den Browser, welche z.B. über ein Unix Socket Kommandos vom Client Player entgegennimmt. Für den Demonstrator ist die Entwicklung einer Browsererweiterung aber nicht notwendig, da die in Mozilla eingebaute Funktionalität zur Browserfernsteuerung ausreicht.

**Fernsteuerung des Browsers** Die für diese Arbeit entwickelte Software funktioniert nur, falls kein anderer Mozilla Prozess existiert als der vom ClientPlayer Verwendete. Der Grund hierfür liegt in der Benützung des 'mozilla -remote "openURL(url)'" Kommandos zum Laden einer Seite in das Browserfenster. Das Kommando lädt die bezeichnete Seite mithilfe des zuerst gefundenen Mozilla Prozesses. Da bisher für die Darstellung der FI nur ein Browser Prozess verwendet wurde, stellt dies allerdings keine relevante Einschränkung dar. Die Aufteilung in unterschiedliche Bereiche(Ebenen) kann wie im bisherigen FIS mit der XML User Interface Language(XUL) [42] erreicht werden. Die Aktualisierung der einzelnen Bereiche ist mit Javascript

<sup>1</sup>Mehr Informationen zu diesem Betriebssystem unter: <http://www.ose.com/>

und XUL ohne ein erneutes Laden der anderen Bereiche möglich.<sup>2</sup> Da für die Aktualisierung der MI aber das 'mozilla -remote "openURL(url)" Kommando die zugrundeliegende XUL Seite neu lädt, ist ein Neuaufbau aller Bereiche nach jedem Wechsel der URL oder der Auflösung für die MI nicht zu vermeiden. Bei Änderungen in der FI können nur die betroffenen Bereiche neu geladen werden. Mangels Anbindung des Demonstrators an den die FI liefernden REC öffnet die XUL Seite des Demonstrators jedoch keine Bereiche zur Anzeige von FI. Abbildung 6.3 illustriert die beschriebenen Vorgänge zur Steuerung und Aktualisierung der Inhalte im Browser bei angebundenem REC.

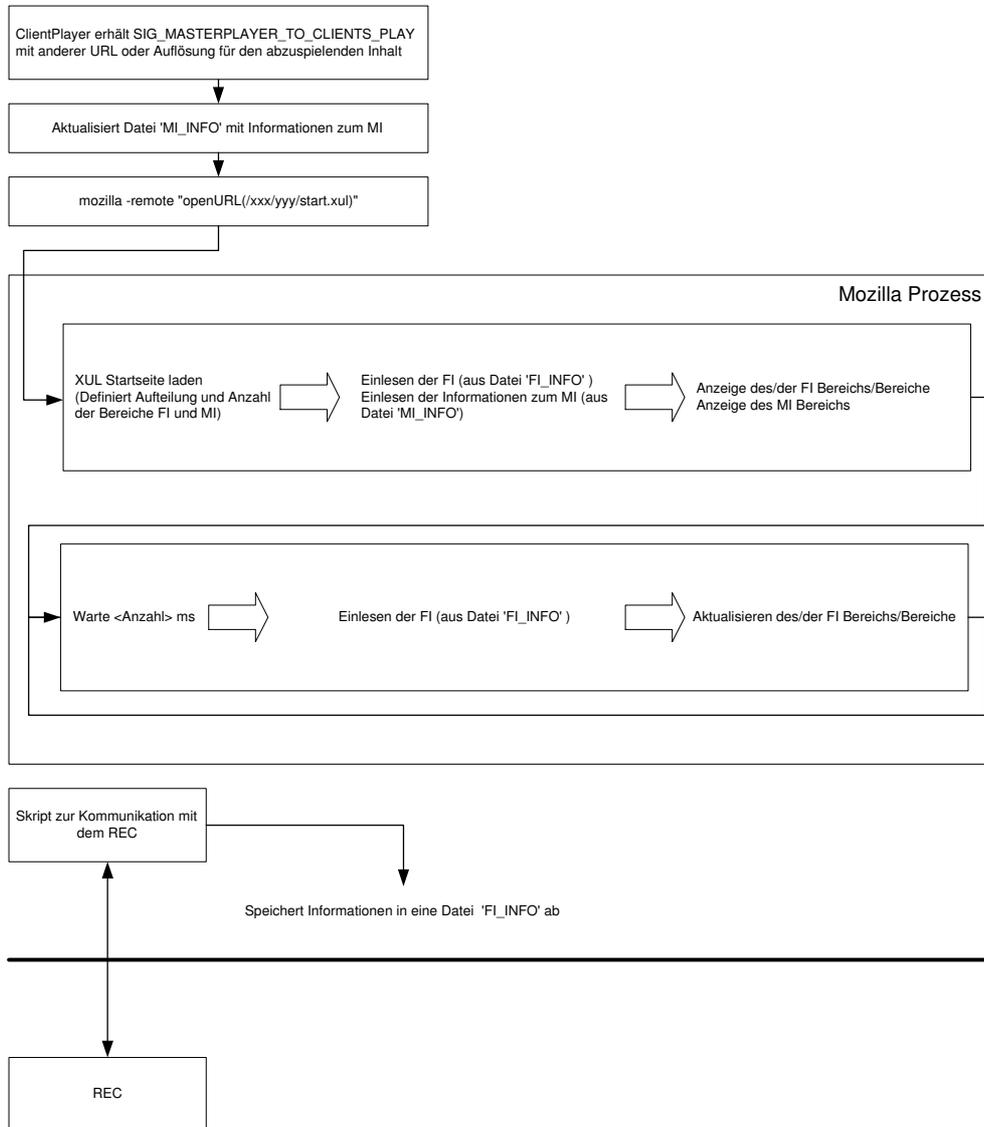


Abbildung 6.3: Steuerung und Aktualisierung der Inhalte im Browser

<sup>2</sup>Es sei hier auf die Dokumentation und Software zum Projekt THURBO der Ascom verwiesen. Diese ist nur Mitarbeitern der Ascom zugänglich.

## Kapitel 7

# Ablaufplanung der Multimedia Inhalte

Nach der Klärung der Fragen, wann FI und wann MI gezeigt werden und wie dies auf dem Demonstrator gemacht wird, muss nur noch definiert werden, wann welche MI präsentiert wird. Eigentlich würde folgender Algorithmus zur Definition, wann welche MI präsentiert werden, ausreichen: Wähle zufällig einen MI aus und spiele diesen ab. Da aber erstens Werbung abhängig von Ort, Ereignissen oder Zeit möglich sein soll und zweitens Inhalte vom Typ FILE, bevor sie angezeigt werden können, zu den Slaves übertragen werden müssen, ist die Lösung nicht ganz so einfach. Im Folgenden wird zuerst ein Algorithmus entwickelt, der sicherstellt, dass jeder Inhalt zu jedem Planungszeitpunkt geplant werden kann, ohne dass dabei erzwungene Totzeiten entstehen. Solche Totzeiten können bei Inhalten vom Typ FILE entstehen, wenn vom Planungszeitpunkt bis zum geplanten Abspielzeitpunkt nicht genügend Zeit für deren Übertragung zu den Slaves zur Verfügung steht. Anschliessend wird noch ein Algorithmus zur Auswahl des nächsten zu planenden Inhalts gemäss Kriterien wie Ort, Ereignisse und Zeit angegeben. Vorschläge zur Implementierung dieses Algorithmus, sowie Vorschläge, wie die Datenbank mit den Informationen zu den Inhalten aussehen könnte, runden dieses Kapitel ab.

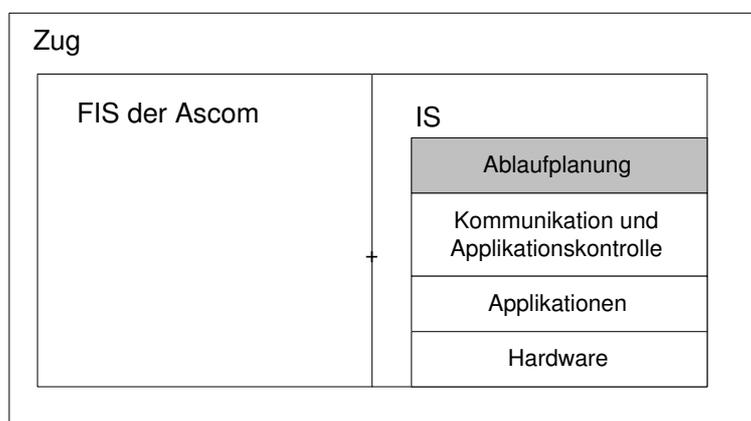


Abbildung 7.1: Betrachtete Systemebene: Ablaufplanung

### 7.1 Algorithmus zur Sicherstellung der Planbarkeit der Inhalte zu jedem Planungszeitpunkt

Für ein System, das nur Inhalte vom Typ STREAM verwendet, ist die Planung der Abspielreihenfolge sehr einfach. Hierzu wird jeweils kurz vor Ende des gerade gespielten Inhalts der nächste Inhalt geplant. Beim hier entwickelten IS ist dieses Vorgehen nicht möglich, da Inhalte

	Bezeichner	Wert
Maximale Bandbreite (theoretisch)	$B_{theo}$	100 MBit/s
Maximale Bandbreite für die Dateiübertragung (künstlich limitiert durch UDPCast)	$B_{UDPCast}$	10 MBit/s
Durchschnittliche Bandbreite für die Dateiübertragung (pessimistische Schätzung)	$B_{avg}$	$0.6 \times B_{UDPCast}$
Zeit für die Initiierung eines Transfers (Starten von UDPCast, Meldungsaustausch)	$t_{init}$	5 s
Dateigrösse	$FileSize$	xxx Bytes
Spieldauer	$t_{play}$	xxx ms

vom Typ FILE vor dem Abspielen zuerst an die Slaves übertragen werden müssen. Damit die Inhalte vom Typ FILE bis zum Zeitpunkt, zu dem sie abgespielt werden sollen, zu den Slaves übertragen werden können, muss bei der Planung genügend weit in die Zukunft geschaut werden. Plant man z.B. beim Starten eines Inhalts bereits den Nächsten und stellt fest, dass dieser vom Typ FILE ist, kann versucht werden, die Datei während dessen Spieldauer zu übertragen. Reicht die Spieldauer zur Übertragung nicht aus, wird entweder die Übertragung abgebrochen und der Inhalt nicht angezeigt, oder es entsteht bis zu deren Beendigung eine Totzeit. Totzeiten sind unerwünscht, da währenddessen keine Einnahmen generierenden Inhalte gezeigt werden können. Eine brauchbare Planung ist erst dann gefunden, wenn der zur Planung verwendete Algorithmus garantiert, dass ausser im Fehlerfall alle Inhalte abgespielt werden können, ohne dass erzwungene Totzeit entsteht. Um diese Garantie geben zu können müssen gewisse Bedingungen für die Zulassung von Inhalten festgelegt werden, wie die Betrachtung des ungünstigsten Falls (worst case) einer Abspielreihenfolge zeigt.

### 7.1.1 Bedingungen für die Zulassung eines Inhalts

Der ungünstigste Fall tritt ein, wenn alle Inhalte vom Typ FILE sind und somit jeder Inhalt vor dem Abspielen zuerst zu den Slaves transferiert werden muss. Für die genauere Untersuchung dieses Falls sei folgendes angenommen:

Die Annahmen sind sehr konservativ, da damit auch beim Abspielen eines Videostreams in DVD Qualität mit gleichzeitiger Dateiübertragung nie mehr als 20% der maximalen theoretischen Bandbreite des Netzwerkes belegt würden. Ausserdem wurde die Zeit von 5 Sekunden für die Initiierung eines Transfers in den Tests nie überschritten. Mit diesen Annahmen kann für jeden Inhalt die benötigte Übertragungszeit wie folgt abgeschätzt werden:

$$t_{transfer} \leq \frac{FileSize \times 8}{B_{avg}} + t_{init}$$

Falls die Übertragung einer Datei mehr als die abgeschätzte Zeit benötigt, wird die Übertragung abgebrochen.

**Der ungünstigste Fall:** Um garantieren zu können, dass alle Inhalte ohne erzwungene Pausen und mit möglichst kurzer Verzögerung bis zur Anzeige des ersten Inhalts abgespielt werden, müssen einige Bedingungen für die Zulassung eines Inhaltes aufgestellt werden. Nimmt man an, dass jeder Inhalt zu jedem Planungszeitpunkt planbar ist, lauten diese wie folgt:

- Die benötigte Zeit für die Übertragung ALLER Inhalte ausser dem Ersten darf nicht grösser sein, als die Summe der Spieldauer aller Inhalte ohne den Letzten.
- Zu jedem Zeitpunkt muss die Spielzeit der bereits geplanten Inhalte mindestens der längsten gemeldeten Übertragungszeit eines Inhaltes entsprechen.

Wie die Beispiele a) und b) in Abbildung 7.2 illustrieren, garantieren diese Bedingungen das Abspielen ohne Zwangspause nur bei entsprechender Wahl der Abspielreihenfolge. Wenn, wie in Beispiel b) gezeigt, zum ersten Planungszeitpunkt nach der Initialisierungsphase Inhalt1 geplant werden soll, gelingt dies nicht ohne das Einfügen einer Zwangspause. Also müssen die

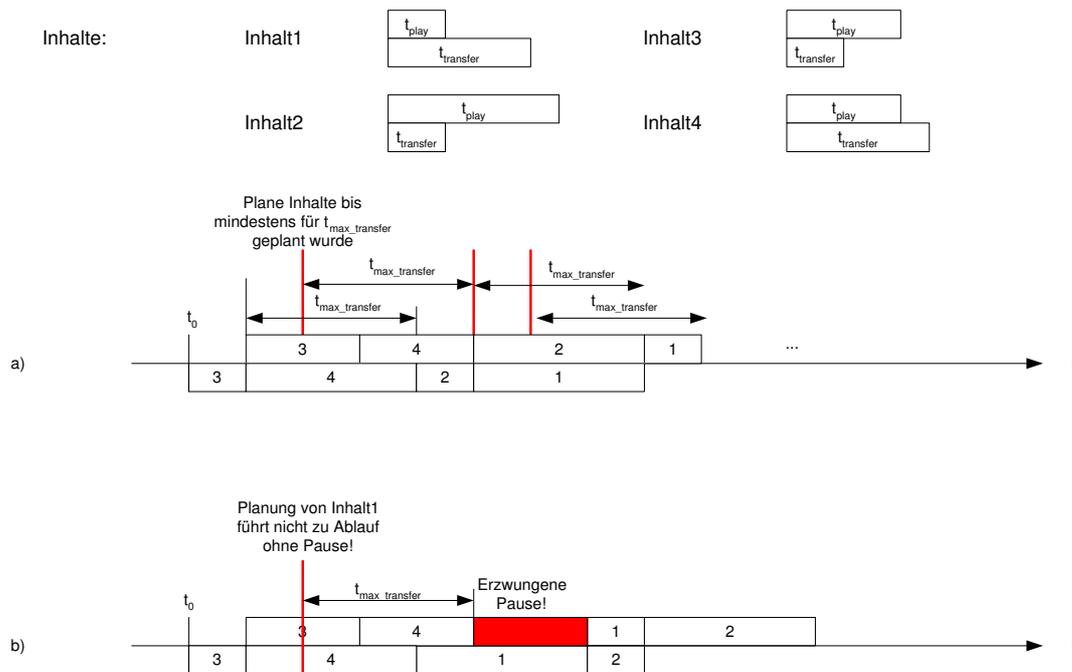


Abbildung 7.2: Ungünstigster Fall: Nur Inhalte vom Typ FILE (I)

Bedingungen noch stärker formuliert werden. Der PlanerApp Dienst, der auf dem Demonstrator sicherstellt, dass jeder Inhalt zu jedem Planungszeitpunkt geplant werden kann, basiert auf folgenden Formulierungen:

1. Die benötigte Zeit für die Übertragung EINES Inhaltes darf nicht länger sein, als die Spieldauer dieses Inhaltes.
2. Zum Startzeitpunkt eines Inhaltes muss eine bestimmte Mindestzeit  $t_{plan}$  über dessen Endzeitpunkt hinaus geplant werden.
3. Jeder Inhalt vom Typ FILE muss in der Zeit  $t_{plan}$  übertragen werden können.
4. Beim Starten des Planungsprozesses muss als erster abzuspielender Inhalt eine Pause der Länge  $t_{plan}$  definiert werden.

Fasst man diese Bedingungen zusammen, liefern sie eine Vorgabe der maximal zulässigen Dateigrösse von Inhalten des Typs FILE. Punkt 1 impliziert:

$$FileSize_{max1} = \frac{(t_{play} - t_{init}) \times B_{avg}}{8}$$

Die Punkte 2 und 3 implizieren:

$$FileSize_{max2} = \frac{(t_{plan} - t_{init}) \times B_{avg}}{8}$$

Zusammengefasst ist die maximal zulässige Grösse einer Datei also:

$$\min(FileSize_{max2}, FileSize_{max1})$$

Abbildung 7.3 illustriert den ungünstigsten Fall bei diesen Einschränkungen und zeigt damit, dass ausser im Fehlerfall garantiert ist, dass jeder Inhalt zu jedem Planungszeitpunkt geplant werden kann.

Die Funktionsweise des im PlanerApp Dienst eingesetzten und diese Einschränkungen verwendenden Algorithmus zeigt Abbildung 7.4. Bei Versendung der Signale 'Start' und 'Stop' durch den REC kann bereits die Variante mit sich überschneidenden FI und MI Ebenen mit erlaubtem

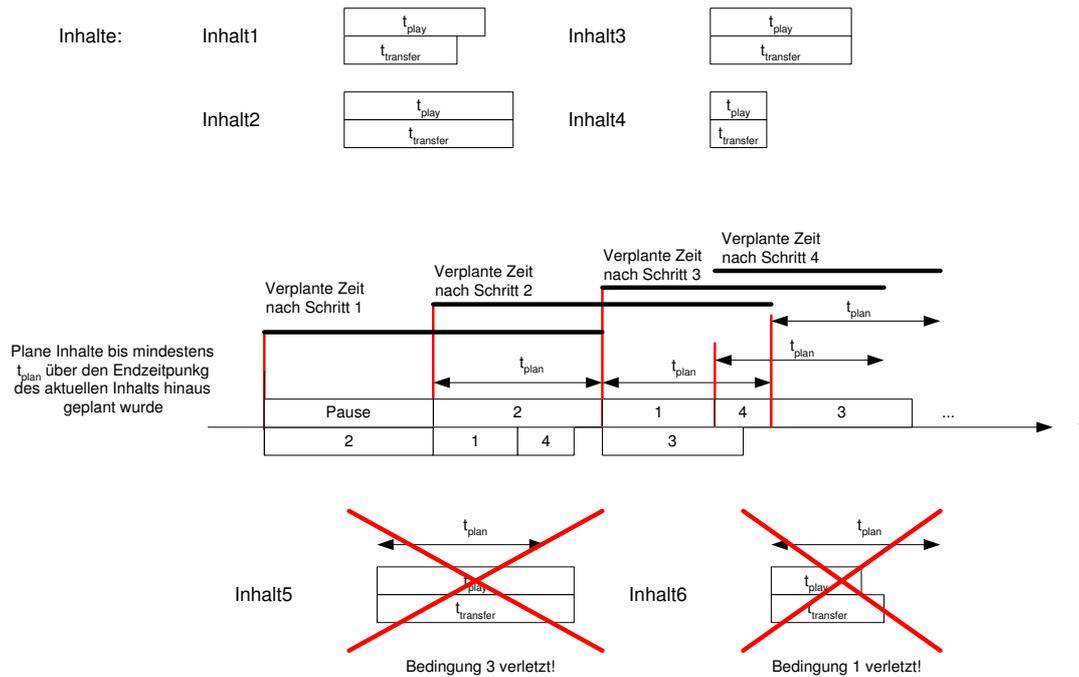


Abbildung 7.3: Ungünstigster Fall: Nur Inhalte vom Typ FILE (II)

Abbruch von Videoclips realisiert werden. Momentan werden diese Signale wegen der fehlenden Anbindung an den REC noch vom PlanerApp Dienst selbst generiert. Das Ziel ist, die Variante mit sich überschneidenden FI und MI Ebenen ohne Abbruch von Videoclips zu realisieren. Dafür muss der Schritt 'Plane Inhalte, bis mindestens für  $t_{plan}$  geplant wurde' anhand der noch verbleibenden Zeit bis zur nächsten Bahnhofseinfahrt die nächsten Inhalte planen.

## 7.2 Algorithmus zur Auswahl des zu planenden Inhalts

Im letzten Abschnitt ist ein Algorithmus präsentiert worden, der garantiert, dass jeder Inhalt zu jedem Planungszeitpunkt geplant werden kann. Wie nun aus einer Menge von Inhalten der nächste zu planende Inhalt ausgewählt wird, soll hier definiert werden. In Abbildung 7.4 entspricht dies dem grau hinterlegten Schritt: 'Plane solange Inhalte, bis mindestens für  $t_{plan}$  geplant wurde'.

### 7.2.1 Auswahlkriterien

In einem ersten Schritt werden die Kriterien anhand deren die Planung vorgenommen wird definiert. Die Kriterien sollen mit dem Ziel, orts-, zeit- und ereignissabhängige Werbung platzieren zu können, ausgewählt werden. Dabei muss natürlich sichergestellt werden, ob die Information überhaupt auf dem Panel PC oder REC verfügbar ist:

- Ort: Angabe des nächsten Bahnhofs durch den REC
- Zeit: Aktuelle Zeit vom REC
- Ereignisse: Noch zu definierende Ereignisse. Mitgeteilt durch den REC (z.B. Werbung für Anti-Stress Pillen bei Ereignis 'Verspätung').

### 7.2.2 Algorithmus

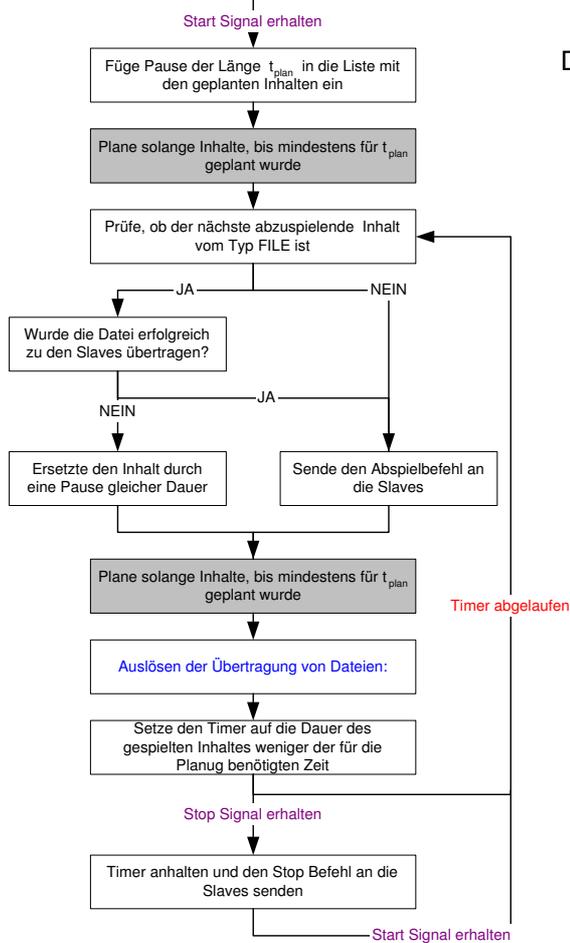
Um die Planung der Inhalte gemäss diesen Kriterien vornehmen zu können, muss für jeden Inhalt definiert werden, bei welchen Werten er abgespielt werden soll. Dies kann z.B. durch

einen Eintrag in eine Datenbank mit der in Abbildung 7.5 gezeigten Struktur gemacht werden. Die in der Abbildung gezeigte Datenbank ist eigentlich nur eine XML Datei. Der Grund für den Vorschlag eine XML Datei als Datenbank zu verwenden, liegt in der einfachen Implementation des ebenfalls in Abbildung 7.5 gezeigten Algorithmus zur Planung des nächsten Inhalts. Mit Hilfe einer Extensible Stylesheet Language Transformations(XSLT) Datei, welche die Auswahl der Inhalte gemäss dem genannten Algorithmus vornimmt und die entsprechend den gerade gültigen Werten von Ort, Zeit und Ereignis angepasst wird, können alle in Frage kommenden Inhalte ausgewählt werden. Der Algorithmus vereinfacht sich, wenn das Abbrechen von Videoclips bei Bahnhofseinfahrten erlaubt ist. Es kann dann der zweite Schritt des Algorithmus übersprungen werden.

### **7.2.3 Algorithmus zur Auswahl des zu planenden Inhalts auf dem Demonstrator**

Auf dem Demonstrator wird zur Auswahl des nächsten abzuspielenden Inhalts kein Algorithmus verwendet, sondern einfach der nächste Eintrag in der XML Datenbank genommen. Wegen der fehlenden Anbindung an den REC erschien es nicht sinnvoll, den Algorithmus zu implementieren. Nur die in der XML Datenbank in Abbildung 7.5 grau hinterlegten Informationen werden für den Demonstrator verwendet. Zur Editierung der auf dem Demonstrator als Datenbank verwendeten XML Datei kann das ebenfalls im Rahmen dieser Arbeit entwickelte plattformunabhängige Tool 'playlisteditor' verwendet werden. Das Tool wird im Anhang B kurz vorgestellt.

Algorithmus zum Abspielen der Inhalte:



Dateiübertragung:

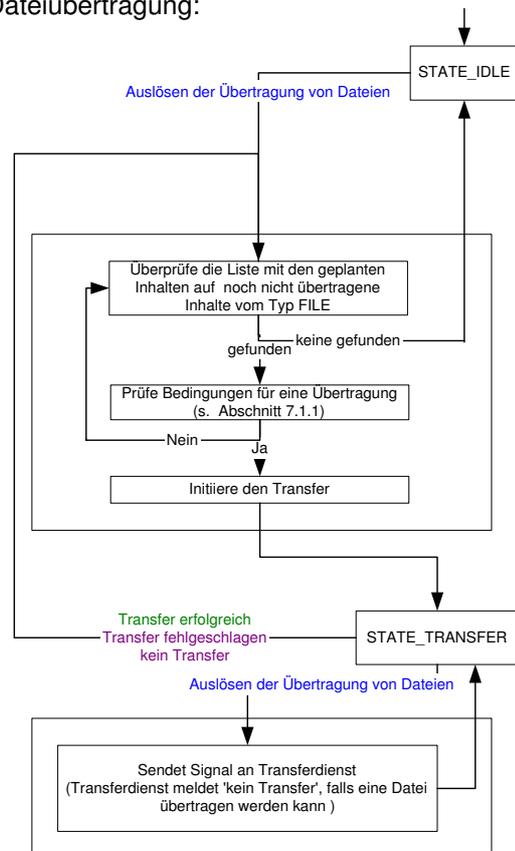
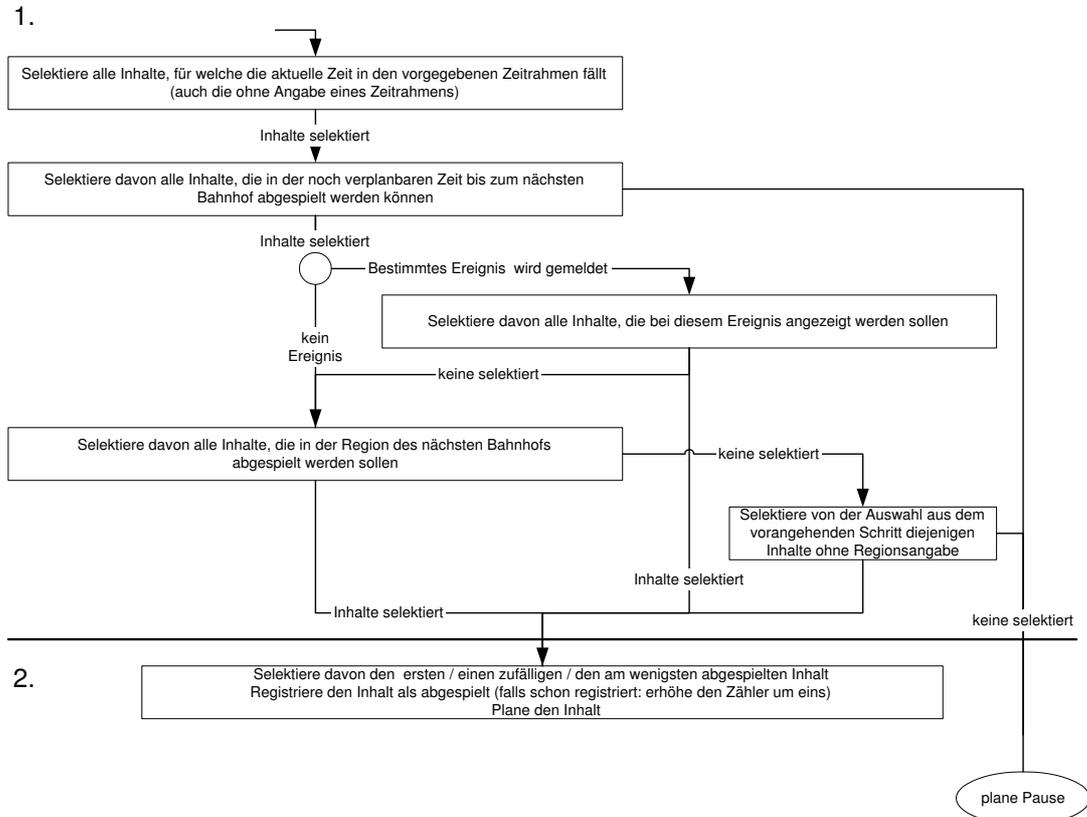


Abbildung 7.4: Funktionsweise des im PlanerApp Dienst eingesetzten Algorithmus zur Ablaufplanung



## Datenbank und Selektionsprozess

### XML Datenbank:

```

Beispiel für einen Eintrag:
<Eintrag>
<Datei wert="URL"/>
<Typ wert="STREAM | FILE | ..."/>
<Dauer wert="Dauer in ms"/>
<Zeitrahmen>
  <Von wert="Zeit"/>
  <Bis wert="Zeit"/>
</Zeitrahmen>
<Geografisch>
  <Ort wert="Bahnhofsname">
  ...
</Geografisch>
<Ereignisse>
  <Ereignis wert="Kennung">
  ...
</Ereignisse>
  
```

### Selektionsprozess:

1. Generiere eine Extensible Stylesheet Language Transformations(XSLT) Datei, welche angewandt auf die Datenbank die gewünschten Inhalte auswählt und speichert.
2. Wähle aus diesen den ersten / einen zufälligen / den am wenigsten abgespielten Inhalt

Abbildung 7.5: Algorithmus zur Auswahl des zu planenden Inhalts



# Kapitel 8

## Fazit und Ausblick

### 8.1 Fazit

Diese Arbeit liefert das Gesamtkonzept für die Erweiterung des FIS der Ascom zum IS. Werden alle im Gesamtkonzept enthaltenen und in Tabelle 8.1 aufgelisteten Konzepte umgesetzt, wird die der Spalte IS der Tabelle 8.2 genannte Funktionalität erreicht.

Neben der konzeptionellen Arbeit wurde auch ein Demonstrator entwickelt, auf dem die meisten Konzepte implementiert und überprüft werden konnten. Performance Messungen am Demonstrator haben gezeigt, dass dieser mit der beschriebenen Hardware- und Systemkonfiguration auch ohne die Dekodierung von MPEG Videos in Hardware<sup>1</sup> über genügend Performance zur Verwendung in einem späteren Produkt verfügt. Es besteht ausserdem die Möglichkeit, dass in Zukunft Plug-Ins existieren, welche Multicast Videostreams empfangen und via XvMC im Browserfenster darstellen können. Dadurch würde eine durchschnittliche Prozessorauslastung von unter 45% erreicht<sup>2</sup>. Welche Konzepte auf dem Demonstrator implementiert und überprüft werden konnten, ist der Tabelle 8.1 zu entnehmen. Diese Konzepte benötigen Daten aus dem FIS der Ascom, genauer vom REC. Der REC kann die benötigten Daten aber ohne Softwareänderung und Definition eines Interfaces für den Master Panel PC nicht bereitstellen. Da der Aufwand hierfür aber beträchtlich ist und von den Entwicklern der Ascom vorgenommen werden muss, werden die Änderungen und das Interface erst im Falle einer Weiterentwicklung des Demonstrators zum Produkt definiert. Trotzdem ist der Funktionsumfang des Demonstrators nur wenig eingeschränkt gegenüber dem Gesamtkonzept IS, wie der Vergleich in Tabelle 8.2 zeigt.

### 8.2 Ausblick

Die Ascom muss nun entscheiden, ob aus dem Demonstrator und den Konzepten aus dieser Masterarbeit einmal ein marktreifes Produkt entwickelt werden soll. Obwohl die Masterarbeit bereits einen grossen Schritt in diese Richtung gemacht hat, erfordern die erwähnten Änderungen an der REC Software, sowie die daraus folgenden Anpassungen an der Demonstrator Software noch einiges an Entwicklungs- und Testzeit. Zur komfortablen Verwendung des mobilen Teils des IS ist ausserdem ein statisches Gegenstück wünschenswert. Ohne dieses Gegenstück, das die Infotainment Daten erfasst, die Abspielbedingungen definiert und schliesslich alles an das mobile IS übergibt, muss alles von Hand erledigt werden. Gerade im Bereich der Übertragung der Daten vom stationären zum mobilen IS auch während der Fahrt mit hoher Geschwindigkeit gibt es noch ein grosses Entwicklungspotential. Ein Versuch der Französischen Staatsbahnen vom 21. April 2004<sup>3</sup> zeigte z.B., dass mit einer Satellitenverbindung bei 300 km/h gerade mal Übertragungsraten von 2Mbit/s im Upstream und 4MBit/s im Downstream erreichbar sind.

---

<sup>1</sup>Obwohl die Dekodierung in Hardware durch die Xvideo with Motion Compensation(XvMC) funktioniert, kann sie nicht für den Demonstrator verwendet werden. Wieso dies so ist, kann in Abschnitt 4.3 nachgelesen werden.

<sup>2</sup>Bezieht sich auf die Messergebnisse in Abschnitt 4.3

<sup>3</sup>Siehe hierzu die Meldung auf <http://www.tt-hardware.com/modules.php?name=News&file=article&sid=8088>

Konzept	Auf Demonstrator implementiert
Datenverteilung mit Multicast	Ja
Master-Slave Kommunikation	Ja
Planung der Inhalte unter Berücksichtigung der für die Übertragung von Dateien benötigten Zeit	Ja
Planung der Inhalte unter Berücksichtigung von Kriterien wie Ort, Zeit und Ereignisse	Nein
Anzeige der Inhalte im Browser	Ja
Berücksichtigung mehrerer konkurrierender Inhalte im Browser	Nein

Tabelle 8.1: Im Gesamtkonzept enthaltene Konzepte

	IS	Demonstrator
Multicast Dateitransfer	Ja	Ja
Multicast Master-Slave Kommunikation	Ja	Ja
Unicast Slave-Master Kommunikation	Ja	Ja
Inhalte und Dateiübertragungen gemäss einer Playlist planen	Nein	Ja
Inhalte und Dateiübertragungen anhand von Kriterien wie Zeit, Ort und Ereignissen planen	Ja	Nein
Anzeige der Inhalte in einem Browserfenster	Ja	Ja
Mehrere Anzeigebereiche im Browser	Ja	Ja
Mehrere Ebenen unterschiedlicher Priorität (pro Anzeigebereich im Browser)	Ja	Nein

Tabelle 8.2: Funktionalität des konzipierten IS im Vergleich mit dem Demonstrator

# Anhang A

## RCM Erweiterungen

### A.1 RCM Erweiterung: Multicast

IP Adressen der Klasse D (224.0.0.0 und 239.255.255.255, Binär: 1110 xxxx xxxx xxxx) sind Multicast Adressen. Die Verwendung der folgenden Adressen ist spezifiziert:

Local scoped Adressen (werden nicht geroutet):

224.0.0.1 All Hosts

224.0.0.2 All Multicast Routers

224.0.0.3-224.0.0.255

Global scoped Adressen (werden geroutet):

224.0.1.0-238.255.255.255

Limited scope Adressen (geroutet innerhalb eines autonomen Systems):

239.0.0.0-239.255.255.255

Um Multicast auf der Data Link Layer (Layer 2) zu ermöglichen, muss auch hier ein Multicast Paket erkannt werden können. Dies wird erreicht, indem ein von der IANA für diesen Zweck reservierter Block von MAC Adressen verwendet wird. Beim Verschicken eines Pakets mit einer IP der Klasse D übersetzt ein OS mit Multicast Unterstützung diese in eine entsprechende MAC Adresse. Die IEEE LAN Spezifikation definiert, dass für Multicast und Broadcast Adressen das Bit 40 der MAC auf eins gesetzt ist. Zur Identifikation der Gruppe stehen auf Layer 3 28 Bit zur Verfügung. Auf Layer 2 nur 23 Bit, nämlich der MAC Bereich von 0100.5e00.0000 - 0100.5e7f.ffff. Es werden die letzten 23 Bit der Layer 3 Adresse auf die 23 Bit der MAC Adresse gemapped. Zu beachten ist, dass dadurch unterschiedliche Layer 3 Gruppenadressen auf die gleiche Layer 2 Gruppenadresse gemappt werden können. Abbildung A.1 zeigt ein Netzwerk wie es mit den spezifizierten Erweiterungen konfiguriert werden kann.

#### A.1.1 Spezifikation der benötigten RCM Funktionalität

Neben der benötigten Funktionalität sind auch die beteiligten Register des Marvell 88E6063 sowie einige Details zu deren Inhalt angegeben um die Implementation mit Hilfe der Chip Dokumentation zu erleichtern.

##### 1. Eintrag in die ATU:

Eintragen einer oder mehrerer MACs (Multicast MACs) in die ATU(Address Translation Unit) mit Angabe an welche Ports diese weitergeleitet werden sollen. Benötigte Parameter: MAC Adresse, Portvektor, DBNum=0 (Database Number=0, da nur eine Datenbank notwendig). Beteiligte Register:

MAC Adresse: ATU MAC Adressregister

Portvektor: ATU DATA Register

DBNum: ATU Operation Register

Operationsdefinition:

Laden= Entry State Feld im ATU DATA Register nicht Null, Bit 14:12 von ATU Operation Register sind 011

2. Löschen eines Eintrages in der ATU:

Wie beim Eintragen aber mit Operationsdefinition:

Löschen= Entry State Feld im ATU DATA Register ist Null, Bit 14:12 von ATU Operation Register sind 011

3. Löschen aller ATU Einträge:

Nur das ATU Operation Register ist relevant. Lösche alle: Bit 14:12 des ATU Operation Register sind 001

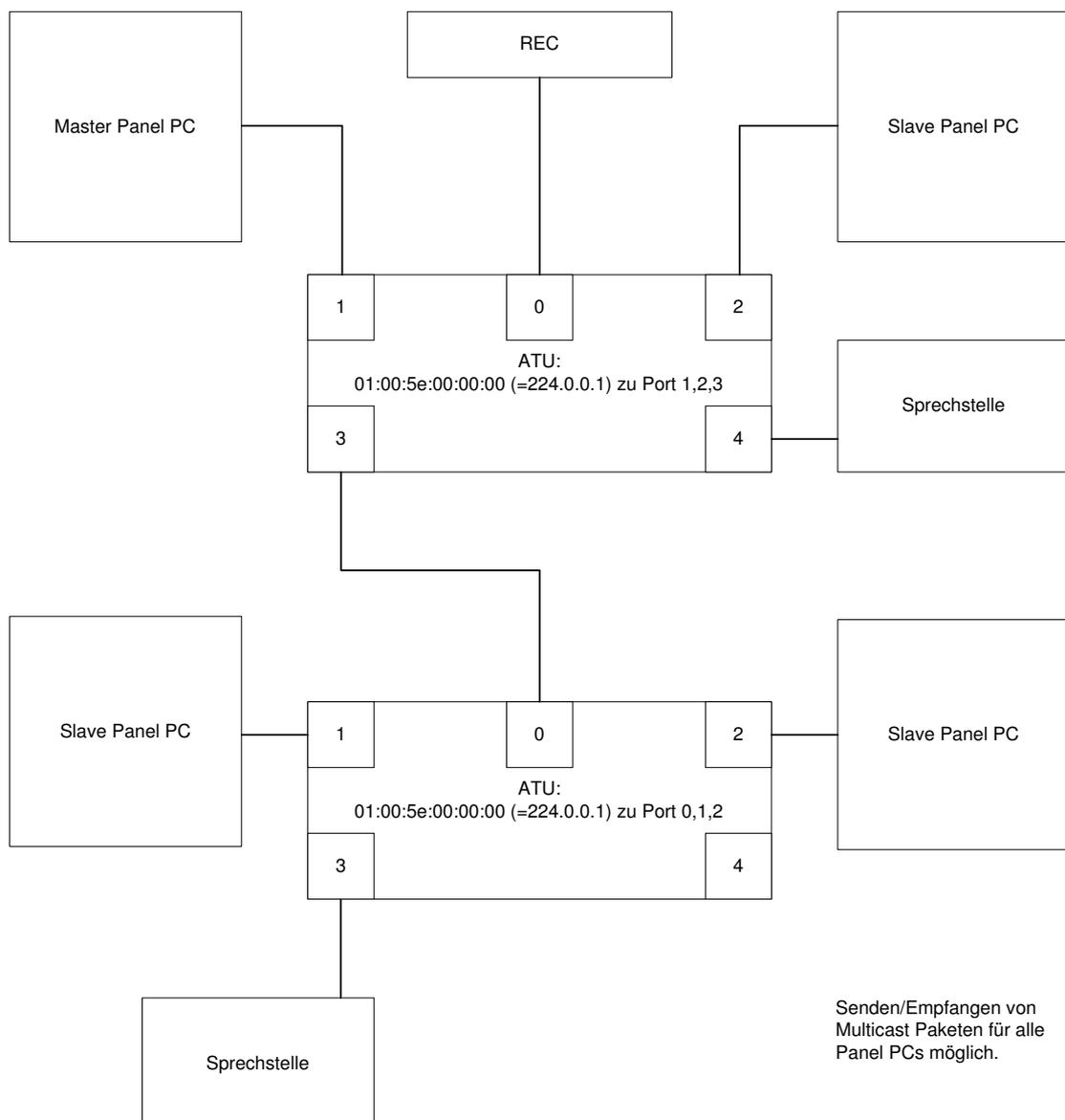


Abbildung A.1: Beispiel eines Multicast Netzwerkes mit 2 RCMs

## A.2 RCM Erweiterung: Priorisierung

Die Konfigurationssoftware für den RCM muss zur Realisierung der 3.2.4 genannten Punkte

1. Priorisierung anhand der Zieladresse
3. Priorisierung gemäss IEEE 802.3ac

angepasst werden. Die Standardeinstellungen erlauben nur eine Priorisierung gemäss dem ToS/DiffServ Feld des IP Paketes.

### A.2.1 Priorisierung anhand der Zieladresse

Um eine Priorisierung anhand der Zieladresse machen zu können, muss für jede Zieladresse ein statischer Eintrag in die ATU (Address Translation Unit) gemacht werden, welcher die Prioritätsinformation enthält. Um Einträge in die ATU machen zu können, muss dieselbe Funktionalität wie in A.1 gewährleistet sein. Zusätzlich muss für einen Eintrag in die ATU der Parameter Prioritätsinformation existieren.

### A.2.2 Priorisierung gemäss IEEE 802.3ac

Bei einer portbasierten Priorisierung ist die Information über die Prioritätsstufe eines Portes nur innerhalb dieses Switches verfügbar. Will man diese Informationen an einen anderen Switch übergeben, muss diese explizit übergeben werden. Genau dies kann durch die Verwendung von IEEE 802.3ac erreicht werden. Wenn nun ein Frame, das noch keine Prioritätsinformationen enthält, an einen anderen Switch gesendet werden soll, wird einfach ein Tag gemäss IEEE 802.3ac eingefügt. Dieses enthält dann die Prioritätsinformation des Quellports. Wird ein Frame aber nur zwischen Geräten die am selben Switch hängen ausgetauscht, ist dies nicht nötig. Es kann direkt die Standard Prioritätsstufe des Quellports verwendet werden. Die bereits vorhandene Software zur Konfiguration des RCMs muss wie folgt erweitert werden:

1. Für jeden Port muss definiert werden können, wie die Frames verschickt werden (tagged, untagged, unmodified).
2. Definition der Standard Prioritätsstufe des Ports (DefPri)
3. Definition der Standard Prioritätsstufe des Ports welche beim Einfügen eines IEEE 802.3ac Tags verwendet wird (TagPriLSB).

## A.3 RCM Erweiterung: ID basiertes VLAN

Ein Port basiertes VLAN kann relativ einfach konfiguriert werden und wird bereits vom RCM unterstützt (via LON konfigurierbar). Port basiertes VLAN meint die Möglichkeit für jeden Port zu definieren, an welche(n) anderen Port(s) er Pakete senden darf. So können z.B. LAN A und LAN B voneinander isoliert werden. Bei der Kaskadierung von RCMs wo sowohl am einen, wie auch am andern Switch Hosts aus LAN A und LAN B hängen ist dieses Konzept aber nicht mehr anwendbar. Erst die Nutzung von IEEE 802.3ac<sup>1</sup> ermöglicht die Isolierung auch bei kaskadierten RCMs. Abbildung A.2 zeigt, eine Beispielkonfiguration eines Netzwerks mit 2 VLANs und zwei RCMs.

### A.3.1 Spezifikation der benötigten RCM Funktionalität

Der Marvell 88E6063 kann IEEE 802.3ac Tags selbständig anfügen, wegnehmen oder unmodifiziert passieren lassen. Dies kann durch Beschreiben der Global Control Register für jeden Port definiert werden. Ob ein Tag eingefügt werden soll, wird durch die Egress Einstellungen des Ausgangsports bestimmt. Soll eines angefügt werden, wird kann als VID (VLAN ID) die

<sup>1</sup>IEEE 802.3ac [9] definiert das Format eines gegenüber IEEE 802.3 um 4 Oktetts erweiterten Ethernetframes.

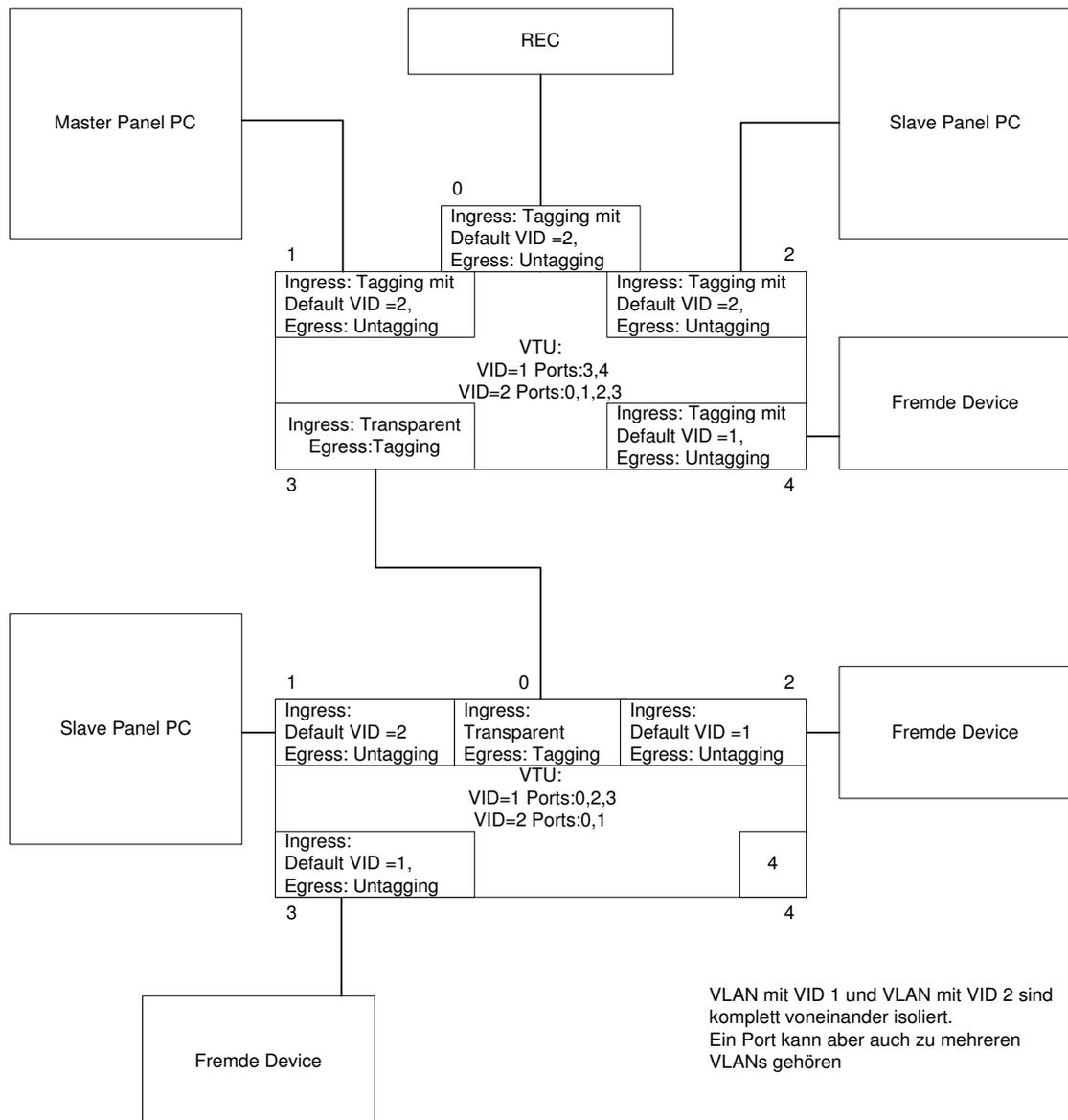


Abbildung A.2: Beispiel eines Netzwerkes mit 2 RCMs und ID basierten VLANs

Standard VID des Ports, über den das Frame hereinkam, verwendet werden. Tags werden sinnvollerweise nur für Ports, die zwei RCMs verbinden, angefügt. Die Berücksichtigung der VLANs bei der Kommunikation von Geräten die am selben RCM angeschlossen sind, erfolgt ebenfalls über die Standard VID der Ports mit Hilfe des 802.1Q [10] Features des RCMs. Ist 802.1Q für einen Port aktiviert, wird für jedes Frame ohne Tag die Standard VID als VLAN ID verwendet und das Switching nur zu den Ports in der VTU mit derselben VID erlaubt. Die Information welcher Port zu welcher VID gehört gehört kann also der VTU (VLAN Translation Unit) entnommen werden. Ein Port kann mehreren VIDs zugewiesen sein. Einträge in die VTU können über das Beschreiben der VTU Register vorgenommen werden. Die bereits vorhandene Software zur Definition von portbasierten VLANs muss wie folgt erweitert werden:

1. Beschreiben der VTU Register mit welchen die VTU Einträge definiert werden können.
2. Für jede VID muss angegeben werden können, welche Ports dazu gehören.
3. Für jeden Port muss definiert werden können, wie die Frames verschickt werden (tagged, untagged, unmodified).
4. Für jeden Port muss die Standard VID festgelegt werden können.
5. Für jeden Port muss der 802.1Q Mode definiert werden können.



# Anhang B

## Tool zur Editierung der Playlist

Dieses Tool wurde mit Hilfe des Qt Kits entwickelt und ist plattformunabhängig. Es ist ein GUI für die Editierung der vom Demonstrator zur Bestimmung der Abspielreihenfolge verwendeten Playlist. Abbildung B.1 zeigt einen Screenshot des Tools. Zusätzlich zur Editierung der Playlist, kann mit dem Tool auch noch ein Start- und ein Stop Kommando an den durch die IP und den verwendeten Port spezifizierten Master Panel PC geschickt werden. Das Start Kommando startet das Abspielen der Inhalte der Playlist. Das Stop Kommando unterbricht das Abspielen. Beim erneuten Senden des Start Kommandos wird die Abarbeitung der Playlist fortgesetzt.

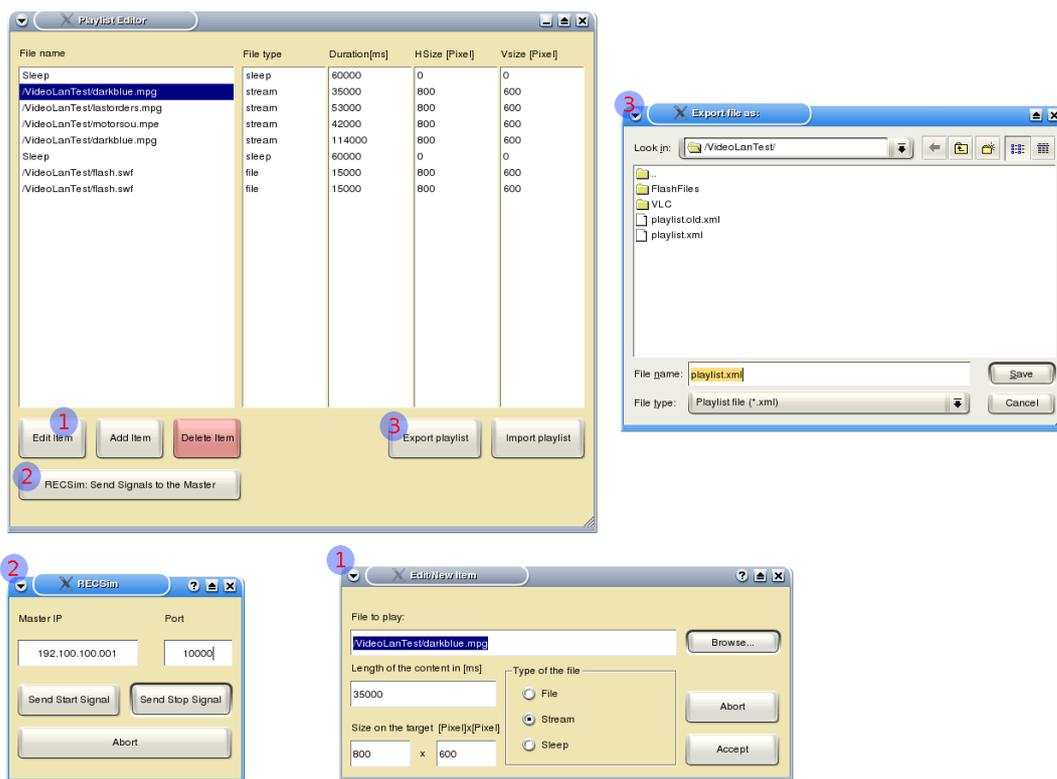


Abbildung B.1: Screenshots vom 'playlisteditor' Tool



## Anhang C

# Installation des Demonstrator Systems

Die Konfiguration und Installation des Demonstrator Systems erfordert folgende Schritte:

1. Installation der ELinOS Linux Distribution und Entwicklungsumgebung.
2. Konfiguration des Linux Betriebssystem für den Panel PC
3. Konfiguration und Installation<sup>1</sup> der benötigten Software für den Panel PC.
4. Erzeugen des Linux Systems für den Panel PC auf dem Entwicklungsrechner.
5. Übertragen des Systems vom Entwicklungsrechner auf den Panel PC.

Die folgenden Abschnitte sollen als Hilfestellung bei der Ausführung der Schritte 1, 2, 4 und 5 dienen. Eine relativ umfangreiche Anleitung zum Schritt 3 bietet Abschnitt D.

### C.1 Installation von ELinOS v3.1

Das Handbuch zu ELinOS führt durch die Installation von ELinOS und sollte zuerst durchgelesen werden. Nach der Installation, kann man die beiden Projekte InstSys und X11 von der DVD zur Masterarbeit in ein beliebiges Verzeichnis verschieben und mit ELinOS öffnen. Es gilt darauf zu achten, dass für den Zugriff auf das Projektverzeichnis keine Rootrechte notwendig sind. So kann bei der Arbeit mit dem Projekt nicht versehentlich das System des Entwicklungsrechner kompromittiert werden.

### C.2 Konfiguration und Anpassung des ELinOS Linux Betriebssystem

ELinOS bietet mit dem Tool Embedded Linux Konfigurator(ELK) die Möglichkeit das Systems über eine grafische Benutzeroberfläche zu konfigurieren. Eine Anleitung dazu liegt ELinOS bei. Die für den Demonstrator notwendigen Kernelkonfigurationen können dem Abschnitt F entnommen werden.

#### C.2.1 Hinzufügen eines als Source Code vorliegendes Programms

Das Programm muss kompiliert und in das Verzeichnis `$ELINOS_PROJECT/app.rootfs` installiert werden. Wie dies konkret gemacht wird, zeigt Abschnitt D anhand von mehreren Beispielen.

---

<sup>1</sup>Installation in das ELinOS Projekt auf dem Entwicklungsrechner. Mit Hilfe des ELinOS Konfigurators kann dann das Linux System für den Panel PC erzeugt werden.

## C.3 Installieren eines ELinOS Projekts auf die Harddisk des Panel PCs

Um dies zu tun, kann das InstSys Projekt von der DVD zur Masterarbeit verwendet werden. Dieses ermöglicht das Booten des Panel PCs über das Netz. Das root-Dateisystems befindet sich hierzu auf dem Entwicklungsrechner (bspl. /tftpboot/192.0.0.1). Nachdem der Panel PC mit diesem System gebootet wurde, kann ein ELinOS Projekt, das auf der Bootmethode lilo basiert, übertragen und installiert werden. Dieses Vorgehen funktioniert momentan nur mit dem ECM 3610 SBC, da der ECM 3611 SBC das Booten über Netzwerk unter Verwendung von PXE nicht unterstützt. Mit einem entsprechenden BIOS Upgrade, sollte dies aber laut Herrn Müller<sup>2</sup> von Inovis auch klappen.

### C.3.1 Konfiguration des Entwicklungsrechners und des Panel PCs

Auf den Entwicklungsrechner muss folgendes Zutreffen:

- NFS Support ist im Linux Kernel integriert oder als Modul verfügbar.
- Installiertes ELinOS v3.1 oder höher
- Benötigte Pakete für den Entwicklungsrechner(Debian Linux):
  - bootp
  - netkit-inetd
  - tftpd-hpa
  - nfs-common (NFS Server)
  - nfs-kernel-server

Das InstSys Projekt ist auf einen Panel PC mit folgender Hardware angepasst:

- Prozessor: Via EDEN C3 (NICHT NEHEMIAH Kern!)
- Netzwerk: RealTek RTL-8139 Chip
- Chipset: VIA VT8606 (Video), TwisterT mit S3 Savage4 Core
- Audio,I/O: VIA VT82C686B

Bei Änderungen in der HW müssen die entsprechenden Änderungen in der Konfiguration des Kernels für das InstSys Projekt vorgenommen werden.

Um sein ELinOS Projekt PXE bootfähig zu machen, muss PXE NFSRoot im ELinOS Projekt als Bootstrategie gewählt werden. Danach sollten alle hierfür notwendigen Kerneleinstellungen automatisch angewählt sein. Aber Achtung: Bei Änderungen im Feature Editor von ELinOS oder bei Neudefinition des Project Environments werden manche im Kernel Konfigurator vorgenommene Anpassungen, rückgängig gemacht/zurückgesetzt. Wichtig ist auch, dass die zum System passende Netzwerkkarte aus [KERNEL]->[Device Drivers]->[Networking Support]-> [Network device support]->[Ethernet (10 or 100MBit/s)->[EISA,VLB,PCI and on board controllers] ausgewählt werden. Nach dem erstellen des Systems mit ELinOS, kann dieses mit dem Skript InstallNFS.sh nach /tftpboot auf dem Entwicklungsrechner installiert werden. Zur Installation in ein anderes Verzeichnis, passe man das Skript entsprechend an. Nun muss der Entwicklungrechner so konfiguriert werden, dass der Panel PC das /tftpboot Verzeichnis als sein root-Dateisystem nutzen kann. Dieses Verzeichnis muss also über das Netzwerk exportiert werden können. Dies übernimmt unter Linux ein Network Filesystem(NFS) Server. Nun muss dem Panel PC beim Booten nur noch mitgeteilt werden, wie und wo er den Kernel findet. Diese Information wird bei Benutzung von PXE an den Panel PC (Booten über LAN mit PXE im BIOS des Panel PC aktivieren) per BOOTP übertragen. Der Kernel kann dann via TFTP übertragen werden. Die TFTP Funktionalität ist im PXE BIOS enthalten. Nach dem

<sup>2</sup>Kontakt: mueller@inovis.ch

Laden des Kernels kann dann via NFS das vom Entwicklungsrechner exportierte Verzeichnis als Root Filesystem gemountet werden. Also muss auf dem Entwicklungsrechner neben einem TFTP und BOOTP Server, auch noch ein NFS Server installiert und konfiguriert werden.

Bios Einstellungen des Panel PCs für das Booten über das LAN:

- Advanced BIOS Features->First Boot Device->[LAN]
- Integrated Peripherals->Onboard Lan Boot ROM->[Enabled]

Dann beim Booten SHIFT F10 drücken um das Boot ROM zu konfigurieren.

- Network Boot Protocol->PXE
- Boot Order->Int 18h

**Konfiguration des Entwicklungsrechners:** Es müssen Informationen via BOOTP an den Panel PC geschickt werden. Da der Entwicklungsrechner aber meist in einem Firmennetz hängt, kann die Installation eines BOOTP/DHCP Servers im Konflikt mit einem bereits vorhandenen DHCP Server stehen. Daher sollte der auf dem Entwicklungsrechner aufgesetzte BOOTP Server und der Panel PC auf einer anderen IP Basis als das Firmennetz operieren (Vorzugsweise auf einem nicht gerouteten Bereich, z.B. 192.168.0.xxx). Da oft nur ein Ethernet Interface zur Verfügung steht, muss dieses demzufolge für den Empfang von Paketen aus beiden Netzen konfiguriert werden. Dies kann mit einem virtuellen Interface bewerkstelligt werden. Der folgende Eintrag im File /etc/network/interfaces:

```
iface eth0:0 inet static
    address 192.0.0.99
    netmask 255.255.255.0
    network 192.0.0.0
    broadcast 192.0.0.255
```

weist dem Interface eth0 nach dem Ausführen von 'ifup eth0:0' zusätzlich die IP 192.0.0.99 zu. Zur Konfiguration des BOOTP Daemon muss das File /etc/bootptab angepasst werden.

```
//Name der Konfiguration, Zielverzeichnis
elinos01:td=/tftpboot
    //Das per NFS exportierte Verzeichnis
    :hd=/tftpboot
    //Subnetzmaske
    :sm=255.255.255.0
    //IP des Entwicklungsrechners
    :sa=192.0.0.99
    //Vom Panel PC als root-Dateisystem zu mountendes Verzeichnis
    :rp=/tftpboot/192.0.0.1
    //Zu bootender Kernel. Relativ zum Verzeichnis für :hd
    :bf=/pxelinux.0
    //Netzwerktyp
    :ht=ethernet
    //Adresse, an die der BOOTP Daemon die Antworten sendet
    :ra=255.255.255.255
    //MAC Adresse des Panel PC. Antwortet nur Paketen mit dieser MAC
    :ha=00045F00D2FE
    //Der MAC Adresse zuzuteilende IP
    :ip=192.0.0.1:
```

BOOTP kann entweder als Daemon ('bootpd' ausführen) oder via dem Internet Super Server(ISS) Dienst (inet) bei einer eintreffenden Anfrage gestartet werden. Hierzu ist der Eintrag:

```
bootps dgram udp wait root /usr/sbin/bootpd bootpd -i -t 120}
```

in der Datei /etc/inetd.conf vorzunehmen. Ausserdem muss der inetd service gestartet sein. Prüfen mit: 'ps aux | grep inetd'. Falls nicht gestartet, starten mit: '/etc/init.d/inetd start'

Der TFTP Daemon zum Übertragen des Kernels kann ebenfalls vom inetd bei Bedarf gestartet werden. Folgender Eintrag in die Datei /etc/inetd.conf ist hierzu notwendig:

```
tftp dgram udp wait root /usr/sbin/tcpd /usr/sbin/in.tftpd /tftpboot
```

Um Zugriffe von anderen PCs als dem Panel PC zu verhindern, kann mit Hilfe des Portmappers und der Dateien /etc/host.deny, /etc/host.allow festgelegt werden, welche Hosts welche Services benutzen dürfen. Mehr Informationen hierzu im Internet.

Nun muss nur noch der NFS Server konfiguriert und gestartet werden. Die Datei /etc/exports definiert, welche Verzeichnisse wie freigegeben werden. Mit dem untenstehenden Eintrag, erhält z.B. der Host mit dem Namen elinos01 Lese/Schreibzugriff mit den Rechten des Benutzers Root auf das Verzeichnis /tftpboot/192.0.0.1.

```
/tftpboot/192.0.0.1 elinos01(sync,rw,no_root_squash)
```

Falls kein Domain Name Server(DNS) installiert ist, der den Namen elinos01 in die entsprechende IP umsetzen kann, muss in der Datei /etc/hosts der Eintrag: '192.0.0.1 elinos01' stehen.<sup>3</sup> Wenn alle Änderungen vorgenommen wurden, sollten alle Dienste neu gestartet werden (Neustart des Entwicklungsrechners oder die konfigurierten Dienste einzeln beenden und neu starten).

### C.3.2 Installation des von ELinOS erzeugten Systems auf die Harddisk des Panel PCs

Ausführen des InstallHD.sh Shellscripts, das im X11 Projekt enthalten ist. Es werden alle für die Installation benötigten Dateien in das NFSRoot Verzeichnis (/tftpboot/192.0.0.1) verschoben. Danach wird der Panel PC gebootet. Durch ausführen des InstallLILO.sh Shellscripts im Root Verzeichnis des Panel PCs, wird das X11 System auf die Harddisk des Panel PCs installiert. Es kann sein, dass das InstallLILO.sh Skript und das lilo.conf File neuen Hardware Gegebenheiten<sup>4</sup> angepasst werden muss.

<sup>3</sup>Dies, da der BOOTP Server so konfiguriert wurde, dass er dem Panel PC die IP 192.0.0.1 zuweist

<sup>4</sup>Zum Beispiel wenn zwei Harddisks angeschlossen sind und das System auf der zweiten Harddisk (gemäss Device Nummer) installiert werden soll.

## Anhang D

# Software für den ECM 3611 SBC

Voraussetzung für die Konfiguration und Erstellung des Systems für den ECM 3611 SBC ist die Installation der ELinOS Linux Distribution. Danach kann entweder das mit X11 benannte ELinOS Projekt von der dieser Masterarbeit beiliegenden DVD kopiert und editiert werden oder es kann ein neues ELinOS Projekt erstellt und konfiguriert werden. Hier wird davon ausgegangen, dass ein ELinOS System kreierte wurde und die Kernelkonfiguration derjenigen aus Anhang F entspricht. Als erstes wird nun der für die Grafikausgabe zuständige XServer installiert. Dieser wird so konfiguriert, dass er zum Abspielen von Videos mit CLE266 MPEG Decoder geeignet ist. Danach folgen Anleitungen zur Installation der Software zum Abspielen und Anzeigen der Multimedia Inhalte. Schliesslich wird noch die Installation und Bedienung der selbst entwickelten Kommunikations- und Kontrollebene für das Infotainmentsystem beschrieben.

### D.1 Kernelmodule für den CLE266

Damit die Beschleunigungsfunktionen des CLE266 Grafikchip von Via genutzt werden können, muss zuerst das Direct Rendering Manager(DRM) Kernel Modul sowie das 'via' Kernel Modul kompiliert und installiert werden:

1. Wechseln in das Verzeichnis, wo die Kernel Source für das ELinOS Projekt abgelegt ist.

```
>cd $ELINOS_PROJECT/linux/drivers/char/drm
```

2. Herunterladen der aktuellen Source für das Kernel Modul:

```
>cvcs -z3 -d:pserver:anonymous@dri.freedesktop.org:/cvs/dri co drm
```

3. Kompilieren und Installieren der Kernelmodule:

```
>cd drm/linux-core
>make LINUXDIR=$ELINOS_PROJECT/linux DRM_MODULES=via
>cp -p *.ko $ELINOS_PROJECT/app.rootfs/lib/modules/{KERNEL-
VERSION}/kernel/drivers/char/drm/
```

### D.2 Installation und Konfiguration von XFree86 v4.4.0

Wechseln in das ELinOS Projektverzeichnis, setzen der Umgebungsvariablen für ELinOS (> ELinOS.sh). Wechseln in das '\$ELINOS\_PROJECT/src' Verzeichnis.

1. Herunterladen des Quellcodes für den XFree86 v4.4.0 Xserver von <http://prdownloads.sourceforge.net/unichrome> . Entpacken des Quellcodes:

```
>tar jxf XFree86-4.4.0.tar.bz2
```

Dies erzeugt ein Verzeichnis 'xc' im \$ELINOS\_PROJECT/src Verzeichnis.

2. Herunterladen der benötigten Grafiktreiber und Bibliotheken sowie den Unichrome Patch für den XFree86 v4.4.0:

```
>wget -nd -P . http://prdownloads.sourceforge.net/unichrome/
XFree86-4.4-libviaXvMC-0.13.3-patch.bz2
>wget -nd -P . http://prdownloads.sourceforge.net/unichrome/
unichrome-X-r30.tar.gz
```

3. Patchen des Xserver Quellcodes:

```
>cd xc
>bzcat ../XFree86-4.4-libviaXvMC-0.13.3-patch.bz2 | patch -p1
>cd programs/Xserver/hw/xfree86/drivers
>mv via via.orig
>tar zxvf ../../../../../../unichrome-X-r30.tar.gz
>ln -s unichrome-X-r30 via
>cd ../../../../../../
```

4. Konfiguriere XFree86 für die Kompillierung. Hierzu muss eine Datei 'host.def' erstellt werden:

```
>cat >config/cf/host.def <<-"END"
#define DefaultGcc2i386Opt -march=i686 -O2 -fomit-frame-pointer
-ffast-math -finline-functions
#define GccWarningOptions -pipe
#define HasMTRRSupport YES
#define HasMMXSupport YES
#define Has3DNowSupport NO
#define HasSSESupport YES
#define NothingOutsideProjectRoot YES
#define EtcX11Directory /etc/X11
END
```

Natürlich kann dies auch mit einem Editor gemacht werden. Es ist darauf zu achten, dass die letzte Zeile der Datei leer ist.

5. Kompiliere XFree86 und installiere ihn danach in das Wurzelverzeichnis für eigene Applikationen im ELinOS Projekt:

```
>make World &> World.log
>make install DESTDIR=$ELINOS_PROJECT/app.rootfs
```

6. Aktivieren der 3D Beschleunigung (nicht notwendig für die Nutzung des MPEG Decoders)

- (a) Herunterladen der Mesa Source (ENTER drücken, wenn nach dem Passwort gefragt wird):

```
>cd $ELINOS_PROJECT/src
>cvs -d:pserver:anonymous@cvs.freedesktop.org:/cvs/ mesa login
>cvs -z3 -d:pserver:anonymous@cvs.freedesktop.org:/cvs/ mesa co Mesa
```

- (b) Konfigurieren von Mesa. Konfigurationsdatei alternativ mit einem Editor erstellen.

```
>cat >>configs/linux-dri-x86 <<-"END"
DRM_SOURCE_PATH = $(ELINOS_PROJECT)/linux/drivers/char/drm/drm
OPT_FLAGS = -march=i686 -O2 -fomit-frame-pointer -ffast-math
-finline-functions
DRI_DIRS = dri_client unichrome
X11_INCLUDES= -I$(ELINOS_PROJECT)/app.rootfs/usr/X11R6/include
-I$(ELINOS_PROJECT)/app.rootfs/usr/X11R6/include/extensions
END
```

- (c) Kompilieren des Treibers und der benötigten Bibliotheken:

```
>make linux-dri-x86
```

- (d) Installieren der Treiber und Bibliotheken ins ELinOS Projekt:

```
>mkdir -p $ELINOS_PROJECT/app.rootfs/usr/local/lib/modules/dri
>find lib/ -maxdepth 1 -iname "libgl*.so*" -print |
  xargs tar cf - \> |
  (cd $ELINOS_PROJECT/app.rootfs/usr/local && tar xpf -)
>cp lib/unichrome_dri.so
  $ELINOS_PROJECT/app.rootfs/usr/local/lib/modules/dri
>cd $ELINOS_PROJECT/app.rootfs/usr/local/lib/modules/dri
>ln -s unichrome_dri.so via_dri.so
```

7. Die Installation ist abgeschlossen. Jetzt muss nur noch die Konfigurationsdatei für den XServer ins entsprechende Verzeichnis kopiert werden:

```
>cp $ELINOS_PROJECT/startupconfig/XF86Config $ELINOS_PROJECT/
  app.rootfs/usr/X11R6/lib/X11
```

## D.3 Installation des Mozilla Browsers und des Macromedia Flash Plug-Ins

Da die Slave Panel PCs die Inhalte in einem Browser anzeigen sollen, muss auch ein solcher installiert werden. Folgendes Vorgehen wird empfohlen:

1. Herunterladen des Releases v1.7.5.1<sup>1</sup> oder neuer von <http://www.mozilla.org>. Type: Linux (x86), Full Installer.
2. Den Installer auf dem Entwicklungsrechner ausführen und 'navigator only' auswählen. Danach als Installationsverzeichnis \$ELINOS\_PROJECT/app.rootfs/usr/local/mozilla angeben.
3. Konfigurieren des Browsers. Es können entweder die Konfigurationen aus dem ELinOS Projekt der Masterarbeit übernommen werden oder beim ersten Starten des Browsers auf dem Panel PC eine entsprechende Konfiguration erstellt und auf den Entwicklungsrechner zurück übertragen werden. Die Konfigurationen befinden sich im ELinOS Projekt im Verzeichnis: \$ELINOS\_PROJECT/app.rootfs/.mozilla/
4. Herunterladen des Macromedia Flash Plug-Ins von [http://www.macromedia.com/shockwave/download/download.cgi?P1\\_Prod\\_Version=ShockwaveFlash](http://www.macromedia.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash). Version 7.0 kann im ELinOS Projekt der Masterarbeit unter \$ELINOS\_PROJECT/src/flash\_player\_7\_linux gefunden werden.
5. Entpacken des Plug-Ins in das Verzeichnis \$ELINOS\_PROJECT/src/flash\_player\_7\_linux. Installation in das ELinOS Projekt mittels:

```
>cd $ELINOS_PROJECT/src/flash_player_7_linux
>cp flashplayer.so $ELINOS_PROJECT/app.rootfs/usr/local/mozilla/plugins
>cp flashplayer.xpt
  $ELINOS_PROJECT/app.rootfs/usr/local/mozilla/components
```

## D.4 Konfiguration und Installation der Software zum Abspielen von Videos

Anleitung zur Konfiguration und Installation der VideoLan und MPlayer Software zum Streamen und Abspielen von Videos. Ausserdem wird noch ein kleiner Player(mp2player) zum testen des MPEG Decoders, sowie die zur Nutzung des MPEG Decoders notwendige XvMC Wrapper Bibliothek installiert.

<sup>1</sup>Dieser Release wurde für die Masterarbeit verwendet.

### D.4.1 Installation der Wrapper Bibliothek

1. Ins Source Verzeichnis des Projekts wechseln, die Source von [http://sourceforge.net/project/showfiles.php?group\\_id=102048](http://sourceforge.net/project/showfiles.php?group_id=102048) herunterladen und entpacken:

```
>cd $ELINOS_PROJECT/src
>tar -zxvf libXvMCW-0.9.3.tar.gz
>cd libXvMCW-0.9.3
```

2. Konfigurieren und installieren der Bibliothek:

```
>echo libviaXvMC.so.1 > XvMCConfig
>./configure --x-includes=$ELINOS_PROJECT/app.rootfs/usr/X11R6/include
--x-libraries=$ELINOS_PROJECT/app.rootfs/usr/X11R6/lib
--includedir=/usr/X11R6/include --libdir=/usr/X11R6/lib
--prefix=/home/betellen/ELinOS31/X11/app.rootfs
--exec-prefix=/home/betellen/ELinOS31/X11/app.rootfs
>make
>make install
>cp XvMCConfig $ELINOS_PROJECT/app.rootfs/etc/X11/
```

### D.4.2 Installation der VideoLan Software

Konfiguration und Installation der VideoLan Client(VLC) Software. Dies beinhaltet ebenfalls die Installation des Mozilla Plug-Ins. Damit das Plug-In bei fehlendem Videostream nicht der Schriftzug '(no picture)' anzeigt, müssen Anpassungen am Code vorgenommen werden. Falls für den VLC die Source der Version 0.8.1 verwendet wird, müssen noch weitere Anpassungen vorgenommen werden. Ansonsten kann die Xvideo Ausgabemethode nicht verwendet werden! Wird eine spätere Version verwendet, ist dieses Problem bereits behoben. Es sei darauf hingewiesen, dass diese Anleitung für die Version 0.8.1 geschrieben wurde und sie für andere Versionen möglicherweise angepasst werden muss.

1. Herunterladen des Source Codes für den VLC v0.8.1 von <http://www.videolan.org/vlc/download-sources.html> und abspeichern im Verzeichnis \$ELINOS\_PROJECT/src/VLC/. In dieses Verzeichnis wechseln und die Datei entpacken:

```
>cd $ELINOS_PROJECT/src/VLC
>tar -zxvf vlc-0.8.1.tar.gz
```

2. Herunterladen der ffmpeg Bibliothek von <http://www.videolan.org/vlc/download-sources.html> und abspeichern im Verzeichnis \$ELINOS\_PROJECT/src/VLC/vlc-0.8.1. In dieses Verzeichnis wechseln und die Datei entpacken:

```
>cd $ELINOS_PROJECT/src/VLC/vlc-0.8.1
>tar -jxf ffmpeg-20041113.tar.bz2
```

3. Konfiguration und Kompilation der Bibliothek:

```
>cd $ELINOS_PROJECT/src/VLC/vlc-0.8.1/ffmpeg
>./configure --cpu=i686 --enable-pp --enable-gpl
>make
```

4. Herunterladen der libmpeg2(mpeg2dec) Bibliothek von <http://www.videolan.org/vlc/download-sources.html> und abspeichern im Verzeichnis \$ELINOS\_PROJECT/src/VLC/vlc-0.8.1. In dieses Verzeichnis wechseln und die Datei entpacken:

```
>cd $ELINOS_PROJECT/src/VLC/vlc-0.8.1
>tar -zxvf mpeg2dec-0.4.0.tar.gz
```

5. Herunterladen der libdvbpsi Bibliothek von <http://www.videolan.org/vlc/download-sources.html> und abspeichern im Verzeichnis \$ELINOS\_PROJECT/src/VLC/vlc-0.8.1. In dieses Verzeichnis wechseln und die Datei entpacken:

```
>cd \${ELINOS_PROJECT}/src/VLC/vlc-0.8.1
>tar -jxf libdvbpsi-20041028.tar.bz2
```

6. Konfigurieren und Kompilieren der libdvbpsi:

```
>cd \${ELINOS_PROJECT}/src/VLC/vlc-0.8.1/libdvbpsi
>./configure
>make
```

7. Jetzt sind alle benötigten Bibliotheken vorhanden. Falls die Version 0.8.1 verwendet wird<sup>2</sup>, muss nun folgende Änderung am Code vorgenommen werden (bei neueren Versionen nicht notwendig):

```
+++ vlc-0.8.1/modules/video_output/x11/xcommon.c
--   while( I_OUTPUTPICTURES < MAX_DIRECTBUFFERS )
++   while( I_OUTPUTPICTURES < 1 /*MAX_DIRECTBUFFERS*/ )
```

8. Nun wird der 'no picture' Schriftzug entfernt, der vom Mozilla Plug-In bei nicht vorhandenem Stream angezeigt wird:

```
+++ vlc-0.8.1/mozilla/vlcshell.c
-- #define WINDOW_TEXT "(no picture)"
++ #define WINDOW_TEXT " "
```

9. Um das Mozilla Plug-In kompilieren zu können, sind auf einem Debian System folgende Pakete notwendig: mozilla-dev und libnspr-dev. Sind diese installiert, kann die VideoLan Software konfiguriert werden:

```
>cd $ELINOS_PROJECT/src/VLC/vlc-0.8.1
>./configure --x-includes=$ELINOS_PROJECT/app.rootfs/usr/X11R6/include
--x-libraries=$ELINOS_PROJECT/app.rootfs/usr/X11R6/lib --disable-mad
--with-ffmpeg-tree=$ELINOS_PROJECT/src/VLC/vlc-0.8.1/ffmpeg
--with-libmpeg2-tree=$ELINOS_PROJECT/src/VLC/vlc-0.8.1/mpeg2dec
--disable-wxwindows --disable-skins2
--with-dvbpsi-tree=/home/betellen/ELINOS31/X11/src/VLC/vlc-0.8.1/libdvbpsi
--disable-rpath --enable-mozilla
```

10. Nun muss man die erzeugte vlc-config Datei editieren und die Einträge -lplds4, -lplc4, -lnspr4 und -lxpcom entfernen. Diese Einträge befinden sich in der Mozilla Sektion. Dies ist notwendig um das Mozilla Plug-In kompilieren zu können<sup>3</sup>. Nun kann die Software kompiliert und installiert werden:

```
>make
>make install DESTDIR=$ELINOS_PROJECT/app.rootfs
>cd mozilla
>cp libvlcplugin.so $ELINOS_PROJECT/app.rootfs/usr/local/mozilla/plugins
>cp vlcintf.xpt $ELINOS_PROJECT/app.rootfs/usr/local/mozilla/components
>rm -r $ELINOS_PROJECT/app.rootfs/usr/local/lib/mozilla
```

### D.4.3 Installation der MPlayer Software

Zur Nutzung des MPEG Decoders muss der MPlayer speziell konfiguriert werden. Der dazu verwendete Video Ausgang XvMC wird durch den bereits entsprechend konfigurierten XServer und der XvMC Wrapper Bibliothek zur Verfügung gestellt. Folgendes Vorgehen führt zu einem XvMC fähigen MPlayer:

<sup>2</sup>Es sei hierfür auf folgende Meldung verwiesen: <http://www.via.ecp.fr/via/ml/vlc/2004-11/msg00160.html>

<sup>3</sup>Siehe hierzu den Hinweis von <http://trac.annodex.net/changeset/1113>

1. Herunterladen der MPlayer Source 'MPlayer-1.0pre6a.tar.gz' von <http://www.mplayerhq.hu/homepage/design7/news.html> und Ablage im Verzeichnis \$ELINOS\_PROJECTS/src/MPlayer. Entpacken der Source:

```
>cd \${ELINOS_PROJECTS}/src/MPlayer
>tar -zxvf MPlayer-1.0pre6a.tar.gz
```

2. Herunterladen des Patches vom Unichrome Projekt unter [http://sourceforge.net/project/showfiles.php?group\\_id=102048](http://sourceforge.net/project/showfiles.php?group_id=102048). Ablage im Verzeichnis \$ELINOS\_PROJECTS/src/MPlayer/MPlayer-1.0pre6a/. Patchen der Source durch:

```
>bzcat MPlayer-1.0pre6a-unichrome-0.5.bz2 | patch -p1
```

3. Herunterladen der LIVE.MEDIA Bibliothek von <http://www.live.com/liveMedia> um RTP/RTSP Streams empfangen zu können. Ablage im Verzeichnis \$ELINOS\_PROJECTS/src/RTPliveLib. Das Entpacken der Source generiert darin das Verzeichnis live/. Verschiebe dieses Verzeichnis mit:

```
>cd \${ELINOS_PROJECTS}/src/RTPliveLib
>cp live $ELINOS_PROJECTS/app.rootfs/usr/local -R
```

4. Konfigurieren des MPlayers. Es muss geprüft werden ob nach Eingabe des folgenden Befehls unter anderem 'XvMC (Via VLD) ....yes' ausgegeben wird!

```
>cd \${ELINOS_PROJECTS}/src/MPlayer/MPlayer-1.0pre6a
>./configure --with-x11incdir=${ELINOS_PROJECT}/app.rootfs/usr/X11R6/include
--with-x11libdir=${ELINOS_PROJECT}/app.rootfs/usr/X11R6/lib/
--prefix=${ELINOS_PROJECT}/app.rootfs/usr
--confdir=${ELINOS_PROJECT}/app.rootfs/etc/mplayer
--enable-runtime-cpudetection --disable-gui --disable-lirc
--enable-xxmc --with-xxmclib=viaXvMC
--with-extraincdir=${ELINOS_PROJECT}/app.rootfs/usr/include/X11/extensions
--enable-live --with-livelibdir=${ELINOS_PROJECT}/app.rootfs/usr/local/live
```

5. Kompilation und Installation des MPlayers:

```
>make
>make install
```

#### D.4.4 Installation der mp2player Software

Diese Software kann nach der Installation des XServers und der XvMC Wrapper Bibliothek in das ELinOS Projekt eingebunden werden um auf dem Panel PC den MPEG Decoder zu testen. Die Software kann allerdings nur MPEG Dateien mit Inhalt vom Typ Elementary Videostream(ES)<sup>4</sup> abspielen. MPEG Dateien vom Type Transport Stream(TS) oder Program Stream(PS) können nicht verwendet werden.

1. Herunterladen der Source von [http://sourceforge.net/project/showfiles.php?group\\_id=102048](http://sourceforge.net/project/showfiles.php?group_id=102048). Entpacken, konfigurieren und installieren mit:

```
>cd \${ELINOS_PROJECTS}/src/MP2Player
>tar -zxvf mp2player-1.2.3.tar.gz
>cd \${ELINOS_PROJECTS}/src/MP2Player/mp2player-1.2.3
>./configure --x-includes=${ELINOS_PROJECT}/app.rootfs/usr/X11R6/include
--x-libraries=${ELINOS_PROJECT}/app.rootfs/usr/X11R6/lib
--includedir=${ELINOS_PROJECT}/app.rootfs/usr/X11R6/include
--libdir=${ELINOS_PROJECT}/usr/X11R6/lib
--prefix=${ELINOS_PROJECT}/app.rootfs
--exec-prefix=${ELINOS_PROJECT}/app.rootfs
>make
>make install
```

<sup>4</sup>Das sind MPEG Dateien, die nur Bilddaten und keine Toninformation enthalten.

## D.5 Installation und Bedienung der Kommunikations- und Kontrollebene

Die Software für die Kommunikations- und Kontrollebene wird aus dem KDevelop3 Projekt 'InfotainmentSystem' generiert welches sich auf der DVD zur Masterarbeit befindet. Das Laden und Kompilieren des Projektes mit KDevelop3 erzeugt die ausführbare Datei start\_is. Ausser dieser Programmdatei wird noch die Konfigurationsdatei InfotainmentSysCfg.xml aus dem /src Verzeichnis des Projektes benötigt. Zur Installation in das ELinOS Projekt müssen diese Dateien lediglich in das Verzeichnis \$ELINOS\_PROJECT/app.rootfs/InfoSys kopiert werden. Zur Anzeige Der Inhalte im Browser muss noch das Verzeichnis /VideoLanTest von der DVD zur Masterarbeit in das Verzeichnis \$ELINOS\_PROJECT/app.rootfs verschoben werden. Diese Verzeichnis enthält alle benötigten XUL und HTML Dateien zur Anzeige der Inhalte im Browser. Ausserdem enthält es ein Beispiel für eine Playlist (playlist.xml).

### D.5.1 Starten der Kommunikations- und Kontrollebene

Die Kommunikations- und Kontrolleben bietet folgende Optionen:

```
****Infotainment System Controll Software, (c) by Ascom (Schweiz) AG ****
Usage: start_is mode [CFG file] \n");

mode: -master  Starts only control software for the master panel PC
      -slave   Starts only control software for the slave panel PC
      -both    Starts control software for the master + slave panel PC
CFG file: Optional. Path and name of the configuration file
          If not specified: \"InfotainmentSysCfg.xml\" assumed
```

Wurde die Software in einer Konsole gestartet, kann sie durch drücken von CTRL+C beendet werden. Das Beenden kann einige Sekunden dauern. Falls die Software im Hintergrund läuft, kann sie durch das Senden von SIG\_ABORT an den start\_is Prozess beendet werden. Bei Abstürzen oder nicht korrektem Beenden, bleiben unter Umständen die im System registrierten Message Queues für den start\_is Prozess erhalte. In diesem Falle entfernt folgendes Skript alle noch vorhandenen Message Queues des Benutzers 'OWNER':

```
#!/bin/bash
if [ $# -eq 1 ]
then
OWNER=$1
for i in `ipcs -q |grep $OWNER |nawk '{print $2}'`
do
echo "killing ipcs process" $i
ipcrm -q $i
done
else
echo "Usage: ${0} OWNER"
fi
```



# Anhang E

## Konfiguration der Dienste des Infotainmentsystems

Die Konfiguration der Dienste des IS erfolgt über eine XML Datei. Die Struktur dieser Datei wird durch die im Projekt enthaltene und InfotainmentSysCfg.dtd benannte Document Type Definition(DTD) vorgegeben. Alle Dienste ausser dem WatchDog haben eine eigene Sektion in der Konfigurationsdatei. Benötigt der Dienst Informationen aus Sektionen von anderen Diensten, werden diese in seiner Sektion referenziert. Für jeden Parameter ist ein Element definiert. Das zugehörige Attribut 'value' enthält den Wert diesen Parameter. Falls der Parameter von einem anderen Dienst referenziert wird, besitzt er ausserdem noch ein Attribut 'id'.

### E.1 MasterMsgModule

Die Sektion für den MasterMsgModule Dienst:

```
<MasterMsgModule>
  <MCConnectionInfo>
    <MCGroupIP id="MMMGIP" value="224.0.0.100" />
    <MCDestPort id="MMMDPORT" value="9000" />
    <MCSrcPort value="8999" />
    <MCNetInterface value="eth0" />
    <UCNetInterface value="eth0:0" />
  </MCConnectionInfo>
  <ParameterInfo>
    <DefaultHTMLPage value="welcome.html" />
    <MozillaStartCommand value="mozilla -remote openURL(" />
    <XULMainPage value="/VideoLanTest/start.xul" />
    <VariableFile value="/VideoLanTest/vars.js" />
  </ParameterInfo>
  <UCSrcPortRef ref="UCSRCPORT" />
</MasterMsgModule>
```

Bedeutung der einzelnen Parameter:

- MCGroupIP: Meldungen für die Slaves werden an diese IP gesendet. Muss eine Multicast IP sein.
- MCDestPort: Meldungen an die Slaves werden an diesen Port geschickt.
- MCSrcPort: Source Port der Meldungen an die Slaves.
- MCNetInterface: Zum Senden der Meldungen an die Slaves zu benutzendes Netzwerkinterface.
- UCNetInterface: Die diesem Netzwerkinterface zugewiesene IP wird den Slaves als Master IP angekündigt.

- DefaultHTMLPage: Per Default anzuzeigende HTML Seite. Angabe relativ zum Verzeichnis im Parameter 'VariableFile'.
- MozillaStartCommand: Kommando zum anzeigen eines neuen Infotainment Inhalt. Entweder muss das Verzeichnis, wo sich Mozilla befindet, in der PATH Umgebungsvariable enthalten sein, oder der gesamte Pfad muss hier angegeben werden.
- XULMainPage: Pfad und Name der für das MozillaStartCommand zu verwendenden HTML oder XUL Datei.
- VariableFile: Pfad und Name der Datei, in welche die Informationen zum nächsten anzuzeigenden Inhalt geschrieben werden sollen. Muss im selben oder in einem Unterverzeichnis des durch den Parameter 'XULMainPage' spezifizierten Verzeichnis sein.<sup>1</sup> Wird von der HTML Seite benötigt, welche die Infotainment Inhalte anzeigt. Muss deshalb dem dort verwendeten Namen für dieses Datei entsprechen.

## E.2 MasterFileTF

Die Sektion für den MasterFileTF Dienst enthält die der Software UDPCast übergebenen Parameter:

```
<MasterFileTF>
  <MasterUDPCastStart>
    <Mode value="--async" />
    <MaxBitRate value="10m" />
    <FEC value="8x8" />
    <MCInterface value="eth0" />
    <MCAllAddr id="MCALLADDR" value="224.0.0.1" />
    <PortBase id="PORTBASESEND" value="11000" />
    <Autostart value="1" />
    <FullDuplex value="yes" />
  </MasterUDPCastStart>
</MasterFileTF>
```

Bedeutung der einzelnen Parameter für UDPCast:

- Mode: Der `--async` Modus verwendet keinen Rückkanal. Der Modus `--sync` hingegen schon.
- MaxBitRate: Maximale Bitrate mit der Dateien versendet werden. Angabe mit `<Anzahl>m` in Megabits.
- FEC: Art und Anzahl der eingefügten Redundanz zur Forward Error Correction(FEC).
- MCInterface: Für den Transfer zu benutzendes Netzwerkinterface.
- MCAllAddr: Für die Ankündigung des Transfers zu verwendende IP Adresse. Sollte eine Multicast IP sein.
- PortBase: Für den Transfer zu verwendender Port. Dieser Port +1 wird ebenfalls belegt.
- Autostart: Anzahl Pakete zur Ankündigung des Transfers. Danach wird mit dem Transfer begonnen. Nur sinnvoll beim Asynchronen Transfer. Beim Synchronen Transfer kann z.B. ab einer bestimmten Anzahl für den Transfer angemeldeter Clients mit diesem begonnen werden.
- FullDuplex: Falls vom Netzwerk unterstützt, kann der FullDuplex Modus aktiviert werden.

Detailliertere Informationen zu den Parameter von UDPCast können bei [40] gefunden werden.

<sup>1</sup>Ansonsten ist dessen Referenzierung aus einer HTML Seite nur umständlich via `file://` möglich.

## E.3 MasterPlayer

Die Sektion für den MasterPlayer Dienst:

```
<MasterPlayer>
  <SOUTRef ref="VLCSOUT" />
</MasterPlayer>
```

Da der MasterPlayer den VLCController steuert, wird auch die Sektion für den VLCController benötigt:

```
<VLCController>
  <VLCPATH value="/usr/bin/vlc" />
  <HTTPsrc value="/VideoLanTest/VLC/http.simple" />
  <VLCInterface value="http" />
  <VLCIP value="127.0.0.1" />
  <VLCPort value="9000" />
  <SOUT id="VLCSOUT" value="udp://224.0.0.50:1234" />
</VLCController>
```

Bedeutung der einzelnen Parameter:

- SOUTRef: Referenziert den SOUT Parameter des VLCControllers.
- VLCPATH: Pfad und Name des zu startenden VideoLan Clients.
- HTTPsrc: Der Name der vom HTTP Interface des VLC zu verwendenden HTML Seite.
- VLCInterface: Name des zur Fernsteuerung des VLC verwendenden Interfaces. Momentan nur http.
- VLCIP: Für die Fernsteuerung des VLC zu verwendende IP.
- VLCPort: Für die Fernsteuerung des VLC zu verwendender Port.
- SOUT: Vom VLC für das Streamen von Videos zu verwendende <Protokoll>://<IP>:<Port>. Multicast fähige Protokolle sind ausser UDP noch RTP/RTSP.

## E.4 UnicastReceiver

Die Sektion für den UnicastReceiver Dienst:

```
<UnicastReceiver>
  <UCConnectionInfo>
    <UCSrcPort id="UCSRCPORT" value="10000" />
    <UCNetInterface value="eth0:0" />
  </UCConnectionInfo>
</UnicastReceiver>
```

Bedeutung der einzelnen Parameter:

- UCSrcPort: Für den Empfang von Unicast Meldungen zu verwendender Port.
- UCNetInterface: Für den Empfang von Unicast Meldungen zu benutzendes Netzwerkin-terface.

## E.5 ClientMsgModule

Die Sektion für den ClientMsgModule Dienst:

```
<ClientMsgModule>
  <UCSrcPortRef ref="UCSRCPORT" />
  <UCNetInterface value="eth0" />
</ClientMsgModule>
```

Bedeutung der einzelnen Parameter:

- **UCSrcPortRef:** Referenziert Parameter aus anderer Sektion. Schickt Meldungen an den dort spezifizierten Port.
- **UCNetInterface:** Das für das Versenden von Meldungen an den Master zu verwendende Netzwerkkinterfece.

## E.6 ClientFileTF

Die Sektion für den ClientFileTF Dienst enthält unter anderem die der Software UDPCast übergebenen Parameter:

```
<ClientFileTF>
  <ClientUDPCastStart>
    <MCInterface value="eth0" />
    <MCAIAddrRef ref="MCAIADDR" />
    <PortBase ref="PORTBASESEND"/>
  </ClientUDPCastStart>
  <FileDirectory value="/VideoLanTest/FlashFiles/" />
  <MaxSpace value="20000000" />
  <AutoClean value="yes">
</ClientFileTF>
```

Bedeutung der einzelnen Parameter:

- **MCInterface:** Für den Transfer zu benutzendes Netzwerkkinterfece.
- **MCAIAddrRef:** Referenziert Parameter aus anderer Sektion. Wartet auf die Ankündigung von Transfers über diese IP.
- **PortBase:** Referenziert Parameter aus anderer Sektion. Hört auf dem dort spezifizierten Port auf angekündigte Transfers.
- **FileDirectory:** Das für die Ablage der empfangenen Dateien zu verwendende Verzeichnis.
- **MaxSpace:** Nicht mehr als diese Anzahl Bytes dürfen im Verzeichnis 'FileDirectory' abgelegt werden.<sup>2</sup>
- **AutoClean:** Wenn 'yes', dann wird beim Starten der Software jeweils der Inhalt des Verzeichnis 'FileDirectory' gelöscht.

## E.7 ClientPlayer

Die Sektion für den ClientPlayer Dienst:

```
<ClientPlayer />
```

Keine Parameter.

<sup>2</sup>Damit wird auch die maximal erlaubte Grösse einer einzelnen Datei festgelegt. Siehe hierzu die Erläuterungen im Abschnitt 5.4.2 und 5.3.2

## E.8 MulticastReceiver

Die Sektion für den MulticastReceiver Dienst:

```
<MulticastReceiver>
  <MCGroupIPRef ref="MMMGIP" />
  <MCDestPortRef ref="MMMDPORT" />
  <MCNetInterface value="eth0" />
</MulticastReceiver>
```

Bedeutung der einzelnen Parameter:

- MCGroupIPRef: Referenziert Parameter aus anderer Sektion. Empfängt Meldungen von der dort spezifizierten IP.
- MCDestPortRef: Referenziert Parameter aus anderer Sektion. Empfängt Meldungen vom dort spezifizierten Port.
- MCNetInterface: Das für das Versenden der Multicast Meldungen zu verwendende Netzwerkinterface.

## E.9 PlanerApp

Die Sektion für den PlanerApp Dienst:

```
<PlanerApp>
  <PlaylistFile value="/VideoLanTest/playlist.xml" />
  <AvgDatarate value="6000000" />
  <PlanAhead value="60000" />
  <MaxTFInitTime value="5000" />
</PlanerApp>
```

Bedeutung der einzelnen Parameter:

- PlaylistFile: Pfad und Name der Datei, welche die Liste mit den abzuspielenden Daten enthält.
- AvgDatarate: Zu erwartende Datenrate für die UDPCast Transfers. Richtwert:  $0.6 \times MaxBitRate$ . Der Parameter MaxBitRate ist in der Sektion des Dienstes MasterFileTF spezifiziert.
- PlanAhead: Zeit in Millisekunden, die über das Ende des nächsten abzuspielenden Inhaltes hinaus geplant wird. Relevant für die Transfers mit UDPCast. Nähere Angaben zu diesem Parameter können im Abschnitt 7.1 gefunden werden.
- MaxTFInitTime: Zeit, die für die Initialisierung eines Transfers mit UDPCast eingerechnet wird. Beeinflusst, ob Dateien noch übertragen werden können oder nicht. Nähere Angaben zu diesem Parameter können im Abschnitt 7.1 gefunden werden.



# Anhang F

## Konfiguration des Kernels für den ECM 3611 SBC

Auflistung der für den Demonstrator gesetzten Kernelkonfigurationen. Alle nicht gesetzten Konfigurationen wurden weggelassen.

```
CONFIG_X86=y
CONFIG_MMU=y
CONFIG_UID16=y
CONFIG_GENERIC_ISA_DMA=y
CONFIG_GENERIC_IOMAP=y

\#
\# Code maturity level options
\#
CONFIG_EXPERIMENTAL=y
CONFIG_BROKEN=y
CONFIG_BROKEN_ON_SMP=y

\#
\# General setup
\#
CONFIG_LOCALVERSION=""
CONFIG_SWAP=y
CONFIG_SYSVIPC=y
CONFIG_POSIX_MQUEUE=y
CONFIG_BSD_PROCESS_ACCT=y
CONFIG_SYSCTL=y
CONFIG_LOG_BUF_SHIFT=14
CONFIG_HOTPLUG=y
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
CONFIG_EMBEDDED=y
CONFIG_KALLSYMS=y
CONFIG_FUTEX=y
CONFIG_EPOLL=y
CONFIG_IOSCHED_AS=y
CONFIG_IOSCHED_CFQ=y
CONFIG_SHMEM=y
CONFIG_ELINOS_CMDLINE="vga=0x305 ip=bootp root=/dev/hda1 rw "
```

```
\#
\# Loadable module support
\#
CONFIG_MODULES=y
CONFIG_MODULE_UNLOAD=y
CONFIG_MODULE_FORCE_UNLOAD=y
CONFIG_OBSOLETE_MODPARM=y
CONFIG_MODVERSIONS=y

\#
\# Processor type and features
\#
CONFIG_X86_PC=y
CONFIG_MVIAC3_2=y
CONFIG_X86_CMPXCHG=y
CONFIG_X86_XADD=y
CONFIG_X86_L1_CACHE_SHIFT=5
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_ALIGNMENT_16=y
CONFIG_X86_USE_PPRO_CHECKSUM=y
```

```
CONFIG_HPET_TIMER=y
CONFIG_PREEMPT=y
CONFIG_X86_UP_APIC=y
CONFIG_X86_UP_IOAPIC=y
CONFIG_X86_LOCAL_APIC=y
CONFIG_X86_IO_APIC=y
CONFIG_X86_TSC=y
CONFIG_X86_MCE=y
CONFIG_X86_MSR=y
CONFIG_X86_CPUID=y

\#
\# Firmware Drivers
\#
CONFIG_NOHIGHMEM=y
CONFIG_MTRR=y
CONFIG_HAVE_DEC_LOCK=y

\#
\# ACPI (Advanced Configuration and Power Interface) Support
\#
CONFIG_ACPI=y
CONFIG_ACPI_BOOT=y
CONFIG_ACPI_INTERPRETER=y
CONFIG_ACPI_BUTTON=y
CONFIG_ACPI_FAN=y
CONFIG_ACPI_PROCESSOR=y
CONFIG_ACPI_THERMAL=y
CONFIG_ACPI_BLACKLIST_YEAR=0
CONFIG_ACPI_BUS=y
CONFIG_ACPI_EC=y
CONFIG_ACPI_POWER=y
CONFIG_ACPI_PCI=y
CONFIG_ACPI_SYSTEM=y

\#
\# CPU Frequency scaling
\#
CONFIG_CPU_FREQ=y
CONFIG_CPU_FREQ_DEFAULT_GOV_USERSPACE=y
CONFIG_CPU_FREQ_GOV_PERFORMANCE=y
CONFIG_CPU_FREQ_GOV_USERSPACE=y
CONFIG_CPU_FREQ_TABLE=y
CONFIG_X86_LONGHAUL=y

\#
\# Bus options (PCI, PCMCIA, EISA, MCA, ISA)
\#
CONFIG_PCI=y
CONFIG_PCI_GOANY=y
CONFIG_PCI_BIOS=y
CONFIG_PCI_DIRECT=y
CONFIG_PCI_MMCONFIG=y
CONFIG_PCI_NAMES=y

\#
\# Executable file formats
\#
CONFIG_BINFORMAT_ELF=y
CONFIG_BINFORMAT_MISC=y

\#
\# Generic Driver Options
\#
CONFIG_STANDALONE=y
CONFIG_PREVENT_FIRMWARE_BUILD=y

\#
\# Block devices
\#
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_RAM_SIZE=131072
CONFIG_BLK_DEV_INITRD=y

\#
\# ATA/ATAPI/MFM/RLL support
\#
CONFIG_IDE=y
CONFIG_BLK_DEV_IDE=y
CONFIG_BLK_DEV_IDEDISK=y
CONFIG_IDEDISK_MULTI_MODE=y
CONFIG_BLK_DEV_IDECD=y
CONFIG_IDE_GENERIC=y
CONFIG_BLK_DEV_IDEPCI=y
CONFIG_BLK_DEV_GENERIC=y
CONFIG_BLK_DEV_IDEDMA_PCI=y
CONFIG_BLK_DEV_VIA82CXXX=y
```

```
CONFIG_BLK_DEV_IDEDMA=y
CONFIG_IDEDMA_IVB=y

\#
\# SCSI device support
\#
CONFIG_SCSI=y
CONFIG_SCSI_PROC_FS=y

\#
\# SCSI low-level drivers
\#
CONFIG_SCSI_QLA2XXX=y

\#
\# Networking support
\#
CONFIG_NET=y

\#
\# Networking options
\#
CONFIG_PACKET=y
CONFIG_PACKET_MMAP=y
CONFIG_NETLINK_DEV=y
CONFIG_UNIX=y
CONFIG_NET_KEY=y
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_PNP=y
CONFIG_IP_PNP_BOOTP=y

\#
\# Bridge: Netfilter Configuration
\#
CONFIG_XFRM=y

\#
\# SCTP Configuration (EXPERIMENTAL)
\#
CONFIG_VLAN_8021Q=y

\#
\# QoS and/or fair queueing
\#
CONFIG_NET_SCHED=y
CONFIG_NET_SCH_CLK_JIFFIES=y
CONFIG_BT is not set

\#
\# Bluetooth device drivers
\#
CONFIG_NETDEVICES=y
CONFIG_DUMMY=y

\#
\# Ethernet (10 or 100Mbit)
\#
CONFIG_NET_ETHERNET=y
CONFIG_MII=y

\#
\# Tulip family network device support
\#
CONFIG_NET_PCI=y
CONFIG_8139TOO=y
CONFIG_VIA_RHINE=y
CONFIG_VIA_RHINE_MMIO=y

\#
\# Input device support
\#
CONFIG_INPUT=y
CONFIG_INPUT_MOUSEDEV=y
CONFIG_INPUT_MOUSEDEV_PSAUX=y
CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768
CONFIG_SOUND_GAMEPORT=y
CONFIG_SERIO=y
CONFIG_SERIO_I8042=y
CONFIG_SERIO_SERPORT=y
CONFIG_INPUT_KEYBOARD=y
CONFIG_KEYBOARD_ATKBD=y
CONFIG_INPUT_MOUSE=y
CONFIG_MOUSE_PS2=y

\#
```

```
\# Character devices
\#
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_DEF_KEYMAP="de"
CONFIG_HW_CONSOLE=y

\#
\# Serial drivers
\#
CONFIG_UNIX98_PTYS=y

\#
\# Watchdog Cards
\#
CONFIG_RTC=y

\#
\# Ftape, the floppy tape device driver
\#
CONFIG_AGP=m
CONFIG_AGP_VIA=m

\#
\# I2C support
\#
CONFIG_I2C=y
CONFIG_I2C_CHARDEV=y

\#
\# I2C Algorithms
\#
CONFIG_I2C_ALGOBIT=y

\#
\# I2C Hardware Bus support
\#
CONFIG_I2C_ISA=y
CONFIG_I2C_PROSAVAGE=y
CONFIG_I2C_VIAPRO=y

\#
\# Hardware Sensors Chip support
\#
CONFIG_I2C_SENSOR=y
CONFIG_SENSORS_VIA686A=y

\#
\# Multimedia devices
\#
CONFIG_VIDEO_DEV=y

\#
\# Video For Linux
\#
CONFIG_VIDEO_DPC=y

\#
\# Digital Video Broadcasting Devices
\#
CONFIG_VIDEO_SAA7146=y
CONFIG_VIDEO_SAA7146_VV=y
CONFIG_VIDEO_VIDEObUF=y
CONFIG_VIDEO_BUF=y

\#
\# Graphics support
\#
CONFIG_FB=y
CONFIG_FB_MODE_HELPERS=y
CONFIG_FB_VESA=y
CONFIG_VIDEO_SELECT=y

\#
\# Console display driver support
\#
CONFIG_VGA_CONSOLE=y
CONFIG_DUMMY_CONSOLE=y
CONFIG_FRAMEBUFFER_CONSOLE=y
CONFIG_FONT_8x8=y
CONFIG_FONT_8x16=y

\#
\# Logo configuration
\#
CONFIG_LOGO=y
CONFIG_LOGO_LINUX_MONO=y
```

```
CONFIG_LOGO_LINUX_VGA16=y
CONFIG_LOGO_LINUX_CLUT224=y

\#
\# Sound
\#
CONFIG_SOUND=y

\#
\# Advanced Linux Sound Architecture
\#
CONFIG_SND=y
CONFIG_SND_TIMER=y
CONFIG_SND_PCM=y
CONFIG_SND_SEQUENCER=y
CONFIG_SND_OSSEMUL=y
CONFIG_SND_MIXER_OSS=y
CONFIG_SND_PCM_OSS=y
CONFIG_SND_SEQUENCER_OSS=y
CONFIG_SND_VERBOSE_PRINTK=y

\#
\# USB support
\#
CONFIG_USB=y
CONFIG_USB_UHCI_HCD=y
CONFIG_USB_PRINTER=y
CONFIG_USB_STORAGE=y
CONFIG_USB_HID=y
CONFIG_USB_HIDINPUT=y

\#
\# File systems
\#
CONFIG_EXT2_FS=y
CONFIG_EXT2_FS_XATTR=y
CONFIG_EXT2_FS_POSIX_ACL=y
CONFIG_FS_MBCACHE=y
CONFIG_FS_POSIX_ACL=y
CONFIG_AUTOFS_FS=y
CONFIG_AUTOFS4_FS=y

\#
\# CD-ROM/DVD Filesystems
\#
CONFIG_ISO9660_FS=y
CONFIG_JOLIET=y
CONFIG_ZISOFS=y
CONFIG_ZISOFS_FS=y
CONFIG_UDF_FS=y
CONFIG_UDF_NLS=y

\#
\# Pseudo filesystems
\#
CONFIG_PROC_FS=y
CONFIG_PROC_KCORE=y
CONFIG_SYSFS=y
CONFIG_TMPFS=y
CONFIG_RAMFS=y

\#
\# Network File Systems
\#
CONFIG_NFS_FS=y
CONFIG_NFS_V3=y
CONFIG_LOCKD=y
CONFIG_LOCKD_V4=y
CONFIG_SUNRPC=y

\#
\# Partition Types
\#
CONFIG_PARTITION_ADVANCED=y
CONFIG_MSDOS_PARTITION=y

\#
\# Native Language Support
\#
CONFIG_NLS=y
CONFIG_NLS_DEFAULT="iso8859-1"
CONFIG_NLS_CODEPAGE_850=y
CONFIG_NLS_ISO8859_1=y
CONFIG_NLS_ISO8859_15=y
CONFIG_NLS_UTF8=y

\#
```

```
\# Kernel hacking
\#
CONFIG_X86_FIND_SMP_CONFIG=y
CONFIG_X86_MPPARSE=y

\#
\# Cryptographic options
\#
CONFIG_CRYPT=y
CONFIG_CRYPT_HMAC=y
CONFIG_CRYPT_NULL=y
CONFIG_CRYPT_MD5=y
CONFIG_CRYPT_DES=y
CONFIG_CRYPT_AES_586=y
CONFIG_CRYPT_CRC32C=y

\#
\# Library routines
\#
CONFIG_CRC32=y
CONFIG_LIBCRC32C=y
CONFIG_ZLIB_INFLATE=y
CONFIG_X86_BIOS_REBOOT=y
```

# Abkürzungsverzeichnis

AFZ	Automatische Fahrgastzählung
AGP	Accelerated Graphics Port
API	Application Programming Interface
ATU	Address Translation Unit
CAN	Controller Area Network. Bussystem mit 10 kbit/s für Buslängen von 6,7km bis 1 Mbit/s für Buslängen bis 40m.
CAT5e	CATegory 5enhanced Kabel
CEN	Comité Européen de Normalisation
CVS	Concurrent Versions System
DDR	Double Data Rate
DGA	Direct Graphics Access
DRI	Direct Rendering Interface
DRM	Direct Rendering Manager
DTD	Document Type Definition
DVI	Digital Video Interface
EN	Europäische Norm. Comité Européen de Normalisation(CEN) herausgegebene Normen.
EN50155	Europäische Norm 50155. Vom CEN definierte Norm für Elektronische Einrichtungen in Schienenfahrzeugen
FDIS	First Draft International Standard
FI	Fahrgastinformation
FIS	Fahrgastinformationssystem
FTP	File Transfer Protocol
FZPF	Fahrzeug Plattform
FZPF	Schnittstelle
GPRS	Global Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communication
HTML	Hypertext Markup Language
IBIS	Integriertes Bordinformationssystem. Spezifikation nach VDV 300 [7] mit maximaler Übertragungsrate von 1200 Baud.
ID	Identifikationsnummer
IDE	Integrated Device Electronics
IEC	International Electrotechnical Commission
IEC 61375	IEC Standard zur Definition der Schnittstellen zwischen Geräten die (a) im selben oder (b) in einem anderen Wagen lokalisiert. Die Schnittstellen sind als Verbindungen zum TCN definiert.
IEC60529	Definiert Schutzklassen für Gehäuse. Angabe durch: IP<Nummer>
IP56	Ingress Protection 56. Schutzklasse für Gehäuse gemäss IEC60529.
KIS	Kundeninformationssystem
LON	Local Operating Network
LVDS	Low Voltage Differential Signaling
MAC	Media Access Control
MI	Multimedia Inhalte
MMSH	Microsoft Media Server Protokoll eingebunden in HTTP
MMSH	Microsoft Media Server Protokoll
MMST	Microsoft Media Server Protokoll mit TCP-basiertem Transport

MMX .....	<u>M</u> ulti <u>M</u> edia <u>e</u> Xtension
MPEG .....	<u>M</u> oving <u>P</u> icture <u>E</u> xpert <u>G</u> roup
MPEG 1,2 u.4 ....	Durch die MPEG definierte Kompressionsverfahren zur verlustbehafteten Kompression von bewegten Bilder.
MVB .....	<u>M</u> ultifunctional <u>V</u> ehicle <u>B</u> us. Bus zur Kommunikation innerhalb eines Wagens. Datenrate von bis zu 1.5 Mbit/s und bis zu 4096 angeschlossene E/A Einheiten [4].
RAM .....	<u>R</u> andom <u>A</u> ccess <u>M</u> emory
RCM .....	<u>R</u> AILVOX <u>C</u> ommunication <u>M</u> odule
REC .....	<u>R</u> AILVOX <u>E</u> Embedded <u>C</u> ontroller
RIO .....	<u>R</u> AILVOX <u>I</u> nterface <u>O</u> utput Module
RIU .....	<u>R</u> AILVOX <u>I</u> nterface für den <u>U</u> IC Bus
RKS .....	<u>R</u> AILVOX <u>K</u> noten <u>S</u> peisung
RLI .....	<u>R</u> AILVOX <u>L</u> ight <u>I</u> nterface
RS .....	<u>R</u> ecommended <u>S</u> tandard
RS232,485,422 ...	Standards für die Kommunikation über eine serielle Schnittstelle
RTP .....	<u>R</u> eal <u>T</u> ime <u>P</u> rotocol
RTSP .....	<u>R</u> eal <u>T</u> ime <u>S</u> ession <u>P</u> rotocol
RVB .....	<u>R</u> AILVOX <u>M</u> VB Interface
SATA .....	<u>S</u> erial <u>A</u> dvanced <u>T</u> echnology <u>A</u> ttachment
SBB .....	<u>S</u> chweizerische <u>B</u> undes <u>B</u> ahnen
SBC .....	<u>S</u> ingle <u>B</u> oard <u>C</u> omputer
SDP .....	<u>S</u> ession <u>D</u> escription <u>P</u> rotocol
SDRAM .....	<u>S</u> ynchronous <u>D</u> ynamic <u>R</u> andom <u>A</u> ccess <u>M</u> emory
SQL .....	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage
SSE .....	<u>S</u> treaming <u>S</u> IMD <u>E</u> xtension
SSL .....	<u>S</u> ecure <u>S</u> ocket <u>L</u> ayer
SVG .....	<u>S</u> calable <u>V</u> ector <u>G</u> raphics
TCN .....	<u>T</u> rain <u>C</u> ommunication <u>N</u> etwork. Netzwerk zur Kommunikation in Zügen bestehend aus dem MVB und dem WTB. Spezifiziert in IEC 61375 [6].
TFT .....	<u>T</u> hin- <u>F</u> ilm <u>T</u> ransistor
UDP .....	<u>U</u> ser <u>D</u> atagram <u>P</u> rotocol
UIC .....	<u>U</u> nion <u>I</u> nternational de <u>C</u> hemins de fer. Vereinigung mit dem Ziel internationale Standards zur veresserung der Interoperabilität der Fahrzeuge und Schienennetze zu schaffen.
UIC 558 .....	UIC standardisierter Kommunikationsbus in Zügen. Verwendet Twisted Pair Kabel. Bis 860 Meter lang. Kann als WTB verwendet werden.
VDV .....	<u>V</u> erband <u>D</u> eutscher <u>V</u> erkehrsunternehmen
VDV 300 .....	Standard der VDV zur Definition der physikalischen Charakteristiken von Computern, Peripherie, Verbindungskabel und Stecker, sowie Protokoll- und Meldungsstruktur für Busse und Züge.
VGA .....	<u>V</u> ideo <u>G</u> raphics <u>A</u> rray
VLC .....	<u>V</u> ideo <u>L</u> an <u>C</u> lient
VNC .....	<u>V</u> irtual <u>N</u> etwork <u>C</u> omputing
WTB .....	<u>W</u> ired <u>T</u> rain <u>B</u> us. Bus zur Kommunikation zwischen allen Wagen eines Zuges mit maximal 1 Mbit/s und 32 Einheiten [4]. Teil des TCN.
WTLS .....	<u>W</u> ireless <u>T</u> ransport <u>L</u> ayer <u>S</u> ecurity. Basiert auf SSL.
XML .....	<u>e</u> Xtensible <u>M</u> arkup <u>L</u> anguage
XUL .....	<u>X</u> ML <u>U</u> ser <u>I</u> nterface <u>L</u> anguage
YUV .....	Eine Methode, Farbwerte in Rastergrafiken oder Videodaten darzustellen. Helligkeit <u>Y</u> und die Farbwerte <u>U</u> und <u>V</u> .
ZCom .....	<u>Z</u> ug <u>B</u> us. Kommunikationsbus für kurze Züge (<10 Wagen).

# Literaturverzeichnis

- [1] Mobil Info: Digitales multimediales Fahrgastinformationssystem für Infotainmentanwendungen in öffentlichen Verkehrsmittel  
Christian M.J. Schäffer, Trend Network AG, Februar 2001,  
<http://mitglied.lycos.de/tecos/moin.pdf>,  
20.11.2004
  
- [2] The TrainCom project  
Part of the Information Society Technologies Programme of the European Commission, 2002-2004,  
[www.traincom.org/documents/presentation.pdf](http://www.traincom.org/documents/presentation.pdf) ,  
20.11.2004
  
- [3] CompactPCI multimedia system for the trams in Athens  
Innova Computers, September 2004,  
[http://www.inova-computers.de/press/Boards\\_and\\_Solutions/pdf/BoardsSolutions\\_09\\_MM\\_04.pdf](http://www.inova-computers.de/press/Boards_and_Solutions/pdf/BoardsSolutions_09_MM_04.pdf),  
06.04.2005
  
- [4] InduCom9 Kommunikationsstrukturen in der Bahntechnik, robuste Schnittstellen mit zusätzlicher Funktionalität  
Simon Seereiner, HARTING Electronics GmbH & Co. KG, 2004,  
<http://www.harting.com/en/en/de/techinfo/tecnews/data/artikel/03232/index.de.html>,  
19.11.2004
  
- [5] Union international de chemin de fer Standards.  
Nicht öffentlich zugänglich. Nur für UIC Mitglieder,  
<http://www.uic.asso.fr>,  
19.11.2004
  
- [6] Train Communication Network, IEC 61375,  
International Electrotechnical Committee, Geneva 1999,  
<http://www.iec.ch>,  
19.11.2004
  
- [7] RBL / LSA / IBIS - Stand und Trends (5): Datentelegramme IBIS Wagen-/ Zugbus (aktualisierte Dokumentation der VDV-Schrift 300, Ergänzung 2, Anlage 11),  
Verband Deutscher Verkehr, 1999,  
<http://www.vdv.de>, *VDV Nummer 3305*,  
23.11.2004
  
- [8] Ingress Protection Standards,  
Beschrieben in IEC/EN 60529,  
<http://www.icode.co.uk/icatcher/support/ip-info.html>,  
25.11.2004
  
- [9] TechFest Ethernet Technical Summary  
TechFest.com, 1999,  
<http://www.techfest.com/networking/lan/ethernet2.htm>,  
15.12.2004

- [10] 802.1Q-IEEE Standards for Local and metropolitan area networks. IEEE Computer Society, 2003,  
<http://standards.ieee.org/getieee802/download/802.1Q-2003.pdf>,  
15.12.2004
- [11] RTP FAQ  
H. Schulzrinne, März 2003  
<http://www.cs.columbia.edu/hgs/rtp/faq.html>,  
13.04.2005
- [12] Samsung TFT-LCD. MODEL NO. LTM150HX-L06  
Samsung Electronics, Juni 2002,  
<http://www.inovis.ch/dokumente/348389980200342141112.pdf>,  
11.04.2005
- [13] ECM 3610 User's Manual, 6th Edition  
Evalue Technology Inc., November 2004,  
<http://www.evalue-tech.com/products/ECM-3610.cfm>,  
11.04.2005
- [14] ECM 3611 User's Manual 1st Edition  
Evalue Technology Inc., October 2004,  
<http://www.evalue-tech.com/products/ECM-3611.cfm>,  
11.04.2005
- [15] VIA Eden, Embedded System Platform Processor Datasheet v. 1.79  
VIA TECHNOLOGIES, INC., September 2004,  
[http://www.via.com.tw/en/downloads/datasheets/processors/eden\\_esp.zip](http://www.via.com.tw/en/downloads/datasheets/processors/eden_esp.zip),  
11.04.2005
- [16] Connectivity and Communication Devices,  
ByteRunner Technologies,  
<http://www.byterunner.com>,  
01.12.2004
- [17] Minicom advanced Systems,  
<http://www.minicom.com>,  
01.12.2004
- [18] Solid State Flash Drives,  
Memtech, 2005  
<http://www.memtech.com>,  
02.12.2004
- [19] Solid State Flash Drives,  
BiTMICRO Networks Inc.,  
<http://www.bitmicro.com>,  
02.12.2004
- [20] MHT2060AH Spezifikationen,  
Fujitsu, 2004,  
[http://www.fdg.fujitsu.com/home/v3\\_\\_product.asp?pid=366&inf=fsp&wg=30](http://www.fdg.fujitsu.com/home/v3__product.asp?pid=366&inf=fsp&wg=30),  
02.12.2004
- [21] Embedded Linux Operating System ELinOS  
Sysgo, April 2005,  
<http://www.sysgo.de/index.php?id=21>,  
13.04.2005
- [22] Accelerated-X Summit v2.2  
XiG, September 2003,  
<http://www.xig.com/Pages/Summit/CardsChips/VIA-S3/VIA-EdenTwisterT.html>,  
12.04.2005

- [23] X.Org X Window System Version 11,  
X.Org Foundation, 2005  
<http://www.x.org>,  
13.04.2005
- [24] XFree86 X Window System Version 11,  
The XFree86 Project, Inc., 1994-2005,  
<http://www.xfree86.org/#whatis>,  
13.04.2005
- [25] Treiber für den S3 Savage (Twister-T)  
Tim Roberts, Dezember 2003,  
<http://www.probo.com/timr/savage40.html>,  
11.04.2005
- [26] Unichrome Grafiktreiber  
The Unichrome Project, März 2005,  
<http://unichrome.sourceforge.net/>,  
11.04.2005
- [27] AppWeb, Embedded Web Server  
mbedthis,  
<http://www.mbedthis.com/>,  
29.11.2004
- [28] Mozilla Webbrowser  
The Mozilla Organization, 2000-2005  
<http://www.mozilla.org>,  
30.11.2004
- [29] Macromedia Flash,  
Macromedia,  
<http://www.macromedia.com/de/software/flash/>,  
08.12.2004
- [30] MPlayer, Media Player for Linux  
The MPlayer Project, 2000-2005,  
<http://www.mplayerhq.hu/homepage/design7/projects.html>,  
06.04.2005
- [31] mplayer plug-in, embedded video player for mozilla  
mplayer plug-in for netscape/mozilla project, Februar 2004,  
<http://mplayerplug-in.sourceforge.net/>,  
06.04.2005
- [32] mozplugger  
Mozplugger Project, 2000-2005,  
<http://mozplugger.mozdev.org>,  
06.04.2005
- [33] VideoLan, Free Software and Open Source video streaming solution for every OS  
VideoLan Project,  
<http://www.videolan.org/>,  
29.11.2004
- [34] The Mozilla plugin  
VideoLan Project,  
<http://www.videolan.org/doc/vlc-user-guide/en/ch07.html>,  
13.04.2005
- [35] OpenOffice Impress, Free to impress your audience,  
Sun Microsystems,  
<http://www.openoffice.org/product/impress.html>,  
29.11.2004

- [36] CrossOver Office,  
CodeWeavers,  
<http://www.codeweavers.com/site/products/cxoffice>,  
29.11.2004
- [37] TightVNC,  
TightVNC project, Constantin Kaplinsky  
<http://www.tightvnc.com>,  
30.11.2004
- [38] RealVNC,  
RealVNC,  
<http://www.realvnc.com>,  
30.11.2004
- [39] MulticastVNC,  
Peter Ziewer,  
<http://www2.in.tum.de/ziewer/multicastvnc/>,  
30.11.2004
- [40] UDPCast,  
LiLux - GNU/Linux User Group Luxembourg ,  
<http://udpcast.linux.lu/>,  
30.11.2004
- [41] TeleTeachingTool,  
Lehrstuhl für Programmiersprachen und Übersetzer Universität Trier,  
<http://teleteaching.uni-trier.de/ttt.html>,  
30.11.2004
- [42] Mozilla XUL and Script Reference,  
XUL Planet, 1999-2005  
<http://www.xulplanet.com/>,  
19.04.2005