

Universität zu Köln

Regionales  
Rechenzentrum  
RRZK  
Universitätsweiter Service



## Streaming

## Inhalt

<b>Einführung</b>	<b>..2</b>
<b>1. Begriffserklärungen</b>	<b>..2</b>
1.1. Allgemeine Begriffe	..3
1.2. Allgemeine Protokolle	..3
1.3. Streaming Protokolle	..3
1.4. Kommunikationskardinalitäten	..4
1.5. Unicast	..4
1.6. Broadcast	..4
1.7. Multicast	..5
<b>2. Vorüberlegungen zum Streaming</b>	<b>..6</b>
2.1. Encoder	..6
2.2. Streamingserver	..8
2.3. Client	..8
<b>3. Streamingplattformen</b>	<b>..10</b>
3.1. Einführung	..10
3.2. Realnetworks	..10
3.3. Apple	..11
3.4. Microsoft	..12
3.5. Andere Angebote	..12
3.6. Überblick	..13
<b>4. Streaming</b>	<b>..15</b>
4.1. Einführung	..15
4.2. Encodieren	..15
4.3. Senden	..20
<b>5. Player einbinden in eine Webseite</b>	<b>..21</b>
5.1. Einbinden per Link	..22
5.2. Einbinden per <OBJECT> Tag	..22
5.3. Einbinden per <EMBED> Tag	..25
5.4. Verknüpfung von <OBJECT> Tags und <EMBED> Tag	..26
<b>6. Metadateien ASX, RAM und SMIL</b>	<b>..28</b>
6.1. Einführung	..28
6.2. RAM und RPM	..28
6.3. ASX	..29
6.4. SMIL	..31
<b>7. Literatur</b>	<b>..34</b>

## Einführung

Wer das Internet nicht nur mit Lynx oder einem anderen Textbrowser durchforstet, wird unweigerlich auf ein großes multimediales Angebot stoßen, angefangen bei der kleinen musikalischen Einlage auf einer privaten Homepage, über Internet Radio und Konferenzen, bis hin zu Live Konzerten und ausgewachsenen Spielfilmen in (fast) Fernsehqualität. Der Begriff Multimedia wird mit unterschiedlichen Bedeutungen gebraucht. Häufig wird damit jede mediale Anwendung beschrieben, zum Beispiel ein Musikstück, das in einer Internetseite eingebunden ist. Multimedia im eigentlichen Wortsinn ist die Verbindung mehrerer medialer Objekte, z.B. Ton mit Bild oder Video, aber auch Text mit Video usw. Im Rahmen dieses Kurses wird der Schwerpunkt auf Audio und Video liegen. Um multimediale Objekte über Netzwerke wie das Internet zu verbreiten, werden häufig sogenannte Streams verwendet, die dem Anwender ein möglichst komfortables Betrachten und Benutzen zu ermöglichen. Streaming bedeutet 'strömend' und bezeichnet das Abspielen von Multimediadateien schon während des Downloads, d.h. sie 'strömen' konstant zum Betrachter. Im Unterschied zu einem gewöhnlichen Download werden die Daten jedoch nicht lokal auf dem Computer des Betrachters gespeichert. Es handelt sich beim Streaming nicht um eine festgelegte Methode, sondern um einen Sammelbegriff verschiedener Techniken. Es wird generell zwischen zwei Arten des Streaming unterschieden, dem On-Demand Streaming, bei dem eine auf einem Streaming Server liegende Datei abgespielt wird, und dem Live Streaming, wobei eine Live Aufnahme direkt in ein Netzwerk gespeist wird. Da die Video- und Audiodaten aufgrund ihrer Größe in der Regel nicht unkomprimiert in ein Netzwerk eingespeist werden können, müssen sie zunächst mit einer Encoder Software in ein passendes Format umgewandelt und komprimiert werden. Nach der Komprimierung werden die Daten auf einem Streaming Server hinterlegt, der sie an die Empfänger weiterleitet. Im Falle des Live Streams müssen die Daten nicht lokal auf dem Server hinterlegt werden, sondern die Daten werden direkt vom Encoder an den Empfänger weitergeleitet, wobei hier ein Server als zwischengeschalteter Verteiler dienen kann. Die zugrundeliegende Technik, die Erstellung solcher Streams und die Einschränkungen werden Thema dieses Kurses sein. Die Möglichkeiten für multimediale Übertragungen im Internet sind sehr groß. Ein paar Beispiele: einfache Werbung, als Möglichkeit zur Entspannung und Zerstreuung, internationalen Live Konferenzen in Bild und Ton, Übertragung von (aufgezeichneten oder live) Seminaren und Veranstaltungen, Lehrmaterial, Informationsvermittlung, aber auch Überwachung usw. sind möglich. Die Möglichkeiten werden also nur von der Vorstellungskraft der Menschen eingeschränkt.

Zu Beginn werden häufig benutzte Begriffe vorgestellt, die im Rahmen dieses Kurses verwendet werden.

## 1. Begriffserklärungen und –bestimmungen

### 1.1 Allgemeine Begriffe

LAN Lokal Area Network: Räumlich eng begrenztes Netzwerk. Z.B. das UKLAN der Universität zu Köln

WAN Wide Area Network: Ein Zusammenschluss mehrerer LANs Z.B. das Internet

Internet	Größtes weltweites Netzwerk
Protokoll	Definierte Vereinbarungen über die Art und Weise des Informationsaustausches zwischen zwei Systemen. Damit sind alle Regeln, Formate, Parameter und Eigenschaften gemeint, die zu einer vollständigen, fehlerfreien und effektiven Datenübertragung beitragen.
HTML	Hypertext Markup Language: Hypertext Dokumentformat, das im Internet und LAN benutzt wird zur Darstellung von Internetseiten.
RFC	Request For Comments: Das Internet und TCP/IP betreffende Standarddokumente.
Routing	Prozess der optimalen Wegwahl von Daten vom Sender zum Empfänger. Die Wegwahl wird mit bestimmten Routing Algorithmen getroffen.

## 1.2 Allgemeine Protokolle

TCP/IP	Transmission Control Protocol/ Internet Protocol: Protokollfamilie, die verschiedene Rechner miteinander verbindet und mittlerweile Grundlage für das Internet ist. TCP ist für die erfolgreiche Versendung von Daten zuständig und kontrolliert, ob Verluste bei der Übertragung aufgetreten sind, in welchem Fall das Datenpaket erneut angefordert wird (RFC 791). IP ist für das Routing (Wegwahl) der Pakete zuständig. (RFC 793)
HTTP	Hyper Text Transfer Protocol: Client-Server-TCP/IP-Protokoll, das zum Austausch von HTML Dokumenten benutzt wird.
FTP	File Transfer Protocol: FTP sorgt für die Übertragung von Daten, gewährt aber keinen Zugriff auf diese. (RFC 959)
SMTP	Simple Mail Transfer Protocol: Protokoll für das Versenden und Empfangen von E-Mails. (RFC 821)
POP	Post Office Protocol: Protokoll zum herunterladen von E-Mails von einem Mail Host. (RFC 1081)

## 1.3 Streaming Protokolle

UDP	User Datagram Protocol: Transportprotokoll, das keine Kontrolle über die gesendeten Daten bietet, ist daher aber schneller als TCP. Es setzt, wie TCP auf IP auf. (RFC 768)
RTP	Real-Time Protocol: HTTP kompatibles Protokoll, d.h. der Streaming Server verhält sich wie ein gewöhnlicher Webserver. RTP ist ein offener Standard.(RFC 1889)

**Real-Time Streaming Protocol:** Streaming Protokoll das (im Gegensatz zu HTTP) auf RTSP die Zeitlinie von Streams achtet, so dass z.B. mehrere Streams zeitgleich betrachtet und gesteuert werden können.

**MMS** Microsoft Media Streaming: Proprietäres Format von Microsoft, mit ähnlicher Technologie wie RTP.

**SDP** Session Description Protocol: Protokoll, das zur Beschreibung einer Multicast Session dient.

**HTTP** Da viele Netzwerke hinter Firewalls liegen, werden nicht alle Streaming Protokolle 'durchgelassen'. Daher weicht man auf HTTP aus.

## 1.4 Kommunikationskardinalitäten

Kommunikationskardinalitäten sind abstrakte Begriffe, die zur Beschreibung eines Kommunikationszustandes zwischen Sender und Empfänger (Server - Client) verwendet werden. Es gibt folgende Kommunikationskardinalitäten:

- **1:1 / Unicast:**  
Der Regelfall: Genau ein Client und genau ein Server kommunizieren untereinander. (Aufrufen einer Webseite, FTP Up- oder Download; TCP und UDP)
- **1:n / Broadcast:**  
Ein Sender schickt Daten an alle Empfänger innerhalb eines Netzes
- **1:(n-x) / Multicast:**  
Ein Sender schickt Daten an eine Gruppe von Empfänger innerhalb eines Netzes
- **m:1 / Concast (Concentration):**  
Viele Sender schicken Daten an einen Empfänger. (Beispiel:??? ;UDP)
- **m:n / Multipeer:**  
Viele Sender schicken an viele Empfänger, z.B. bei einer Konferenz.

Im Bereich des Streamings werden die Begriffe Uni-, Multi- und Broadcast aber auch wie folgt verwendet:

## 1.5 Unicast

1:1 (Transportprotokolle TCP, UDP),  
m:1 (Transportprotokoll UDP).

**Unicast** stellt zur Zeit die häufigst verwendete Art der Kommunikation dar. Ein oder mehrere Sender schicken an genau einen Empfänger Datenpakete.

## 1.6 Broadcast

1:n Kommunikation,  
m:n Kommunikation.

**Broadcast** stellt generell eine Kommunikation an mehrere Clients dar, kann aber sowohl von mehreren Sendern, als auch von genau einem Sender ausgehen. Es 'flutet' ein IP Teilnetz, d.h.

der oder die Sender kommunizieren mit allen Rechnern in diesem Teilnetz. **Broadcast** benutzt das UDP und ist daher nur innerhalb eines Lokalen Netzwerkes anwendbar, da Firewalls in der Regel die UDP Ports abblocken. Die Adressierung folgt über einen IP + UDP Port. Der englische Begriff *broadcast* ist eine Rundfunkübertragung im allgemeinen Sinn, während *live broadcast* eine Direktübertragung meint. Beim Lesen englischsprachiger Dokumente kann es zu einem Missverständnis zwischen diesen Begriffen und den hier verwendeten Begriff **Broadcast** kommen. Im Rahmen dieses Textes ist aber die Kommunikationsart **Broadcast** gemeint.

## 1.7 Multicast

1:n Kommunikation,  
m:n Kommunikation.

Die Kommunikationskardinalitäten des **Multicast** entsprechen denen des **Broadcast**. Der Unterschied liegt jedoch darin, dass der oder die Sender nicht ein ganzes Teilnetz 'flutet', sondern an einen logischen Verband von Empfängern sendet.

### 1.7.1 Gruppenadressen

Um den Verband zu erreichen ist der IP Adressbereich von 224.0.0.0 bis 239.255.255.255 für Multicasttechniken reserviert. Eine Adresse ist nicht auf einen speziellen Rechner festgelegt, sondern auf eine Gruppe von Empfängern, eine so genannte Gruppenadresse. Sie kann also nicht als Quelladresse verwendet werden, sondern nur als Zieladresse. Die Adressen 244.0.0.1 und 244.0.0.2 haben jeweils eine besondere Aufgabe: 244.0.0.1 umfasst sämtliche **Multicast** fähigen Knoten im lokalen Netzwerk, während 244.0.0.2 die **Multicast Router (MRouter)** umfasst.

### 1.7.2 Routing und MRouter

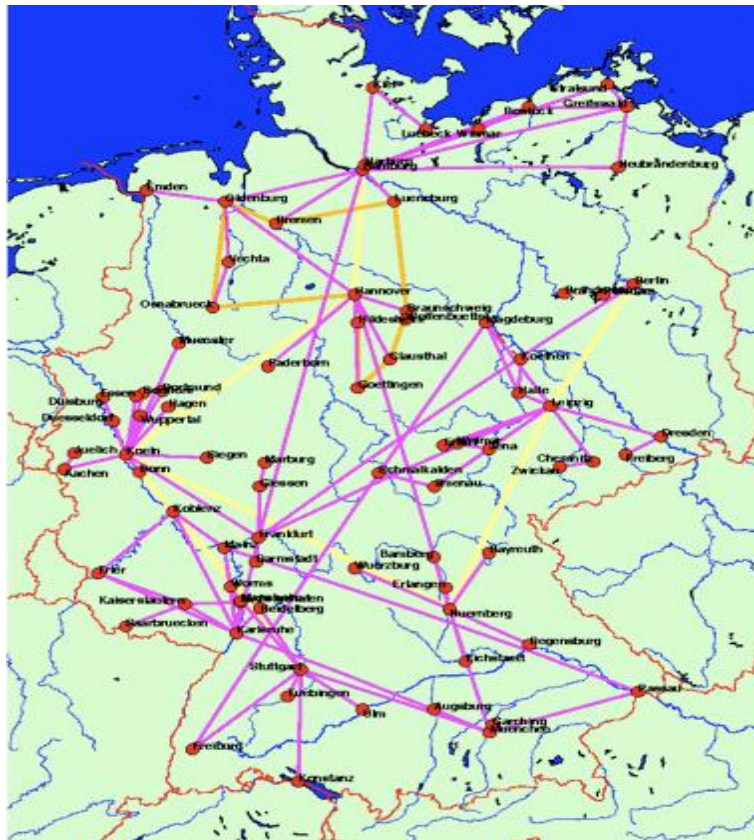
**Multicast** ist technisch für ein LAN vorgesehen, um aus dem LAN in ein WAN wie das Internet zu gelangen ist ein spezielles Routing nötig, das von **MRoutern** übernommen wird. Sie sind dafür verantwortlich, dass die Daten richtig versendet werden. Das Routing in einer Gruppe muss so effizient wie möglich sein, d.h. folgende Anforderungen werden gestellt:

- a. Die Auslastung des Netzwerkes muss so gering wie möglich gehalten werden, um eine Überlastung zu vermeiden.
- b. Schleifen müssen vermieden werden.
- c. Vermeiden von Flaschenhälsen durch die Überlastung eines Knotenpunktes.

Die Router bauen anhand eines Multicast-Algorithmus einen Multicast-Baum auf, Der Multicast-Baum ist ein Verteilerbaum, der unter Berücksichtigung der Gruppenmitgliedschaft den Pfad von Sender zum Empfänger optimiert. Der Baum unterteilt sich in Teilbäume bzw. Teilnetze, die jeweils wieder von eigenen Routern verwaltet werden, er hat also eine hierarchische rekursive Struktur. Daneben haben die Router die Aufgabe der Verwaltung ihrer Gruppen, die sich auf das An-, Abmelden und die Kontrolle von Mitgliedern bzw. Untergruppen beläuft. Diese Verwaltung erfolgt mittels des IGMP (Internet Group Management Protocol), das in das IP integriert ist. Es verlässt das lokale Teilnetz nicht (TTL von 1), sondern kommt nur bis zum nächsten Router. Für die weitere Verbreitung sind Multicastprotokolle zuständig.

### 1.7.3 Multicast BackBONE (MBONE) und DFN Multicast

Der **MBone** ist ein weltweites Overlay-Netzwerk, welches auf der existierenden Internet Infrastruktur aufsetzt und eine Kopplung von entfernten multicastfähigen Netzwerken darstellt. Es bedient sich hierbei dem **MRouting**, soweit das Netzwerk dazu fähig ist. Wenn ein MRouter an ein nicht multicastfähiges Netzwerk stößt werden die Daten in ein TCP verpackt und dann an das nächste LAN gesendet. Diese Technik wird Tunnel genannt. Seit April 2000 hat der **DFN** (Deutsches Forschungsnetz) die IP Multicastfunktionalität in allen Netzknoten des Deutschen Wissenschaftsnetzes integriert. Es besteht aus ca. 3000 angeschlossenen Routern. Das **DFN Multicast** Netzwerk:



## 2. Vorüberlegungen zum Streaming

Es ist eine Reihe von Voraussetzungen und Vorüberlegungen nötig, um Streaming verfügbar zu machen. Diese lassen sich generell in drei große Kategorien unterteilen: Dem Encoder, der die Video- und Audio Daten komprimiert, die Hard- und Software des Servers, der das Streaming bereithält und andererseits die Hard- und Software des Clients, der das Streaming empfangen möchte.

### 2.1 Encoder

#### 2.1.1 Einführung

Die Aufgabe des Encoders ist es, die rohen Daten in ein über das Netzwerk versendbares Format zu codieren. Im Falle des Live Streams muss dies in (fast) Echtzeit geschehen, um eine möglichst verzögerungsfreie Präsentation zu gewährleisten. Die an einen Encoder

gestellten Anforderungen sind hoch. Je nachdem ob er für Live oder On-Demand Streaming oder für beides verwendet wird, muss er in der Lage sein große Datenmengen zu bewältigen und/oder eine gute Prozessorleistung erbringen.

### *2.1.2 Live Encoder*

Der Live Encoder nimmt die Daten, die er über die Video- bzw. Audiokanäle eingespeist bekommt, auf und encodiert sie sofort in mehrere Bitraten, welche er dann als Stream in das Netzwerk einspeist. Da das Encodieren sehr rechenaufwendig ist, benötigt er einen leistungsstarken Prozessor. Kann er die Codierung nicht mehr in Echtzeit bewältigen, wirkt sich das auf die Qualität des Streams aus, welcher zu stocken anfängt. Für Live Streaming reicht, nach Microsoft Angaben, ein Pentium II Prozessor mit 450 MHz aus.

Ein weiterer Punkt ist die Anbindung an das lokale Netzwerk mit einer Netzwerkkarte. Eine 100 Mbit Karte sollte schon Standard sein. Die eigentliche Überlegung liegt in der Installation einer aktiven oder passiven Karte, da eine aktive Netzwerkkarte die Prozessorlast um 50% reduzieren kann und somit die Qualität des Streams sichert.

Als Alternative bieten sich noch Encoderboxen an, die den Prozessor nur gering belasten und die Daten per Hardware encodieren.

### *2.1.3 On-Demand Encoder*

Eine wichtige Überlegung beim On-Demand Stream betrifft die Festplatte, da Audio und Videodaten schnell auch komprimiert hohe Datenmengen annehmen. Ein unkomprimiertes Video von 15 Minuten Dauer mit einer Bildgröße von 320x240, das zur Weiterbearbeitung auf einer Festplatte liegt, nimmt 4,8 GB in Anspruch, komprimiert mit einem Indeo 5.04 Codec sind es 100 bis 200 MB, je nach Bildrate und Material. Aber auch stärker encodierte Daten sind je nach Länge und Format platzraubend. Neben der reinen Festplattengröße spielt auch der Cache der Festplatte eine Rolle, um einen kontinuierlichen Datenstrom zu garantieren. Er sollte mindestens eine Größe von 2 MB aufweisen. Auch sollte überlegt werden, ob eine SCSI oder Firewire Festplatte sinnvoll ist, da diese eine höhere Datendurchsatzrate als herkömmliche IDE Platten bieten, aber gerade in höheren GB Bereich unverhältnismäßig teuer werden. Für Live-Streaming ist die Größe der Festplatte und deren Cache eher nebensächlich, da die Daten direkt in das Netzwerk eingespeist und nicht auf der Platte zwischen gespeichert werden. Der Arbeitsspeicher ist für beide Arten des Streamings mit einer Grundausstattung von 128 MB ausreichend bestückt. Lediglich, wenn der Server neben dem Streaming noch mit anderen Aufgaben betreut werden soll, ist wahrscheinlich eine Aufrüstung des RAM notwendig. Es existieren also unterschiedliche Hardwarevoraussetzungen für den Server, je nachdem, ob er für Live- oder On-Demand Streaming vorgesehen ist. Für On-Demand Produktionen ist besonders auf einen ausreichenden Festplattenspeicher und -cache, sowie der Busbreite der Festplatte zu achten, während für Server, die auf Live-Streaming ausgelegt sind, eine schnelle Prozessorleistung wichtig ist. Um dem Prozessor Arbeit abzunehmen, empfiehlt sich eine aktive Netzwerkkarte beim Live-Streaming.

### *2.1.4 Software*

Ein Encoder ist in der Regel Software (Ausnahme: Encoderboxen), welche auf einem Rechner installiert wird. Die genauere Beschreibung und Funktionsweise der Encoder wird in Streamingplattformen erläutert.

## 2.2 Streamingserver

### 2.2.1 Einführung

Der Server sorgt für die Weiterleitung der Daten an den Client. Bei On-Demand Streams werden die Daten lokal auf diesem Server gespeichert, bei Live Streams dient er nur als Verteiler an die Clients. Im Gegensatz zu einem Webserver ist er für die Verwaltung und Weiterleitung von Streams konzipiert. Er schickt die Daten mittels der Streamingprotokolle chronologisch an die Empfänger und übernimmt auch statistische Aufgaben.

### 2.2.2 Betriebssystem

Das Betriebssystem des Servers spielt eine wichtige Rolle, da nicht alle Multimediaserver auf allen Betriebssystemen laufen. So kann der Microsoft Multimedia Service nur auf dem eigenem System installiert werden. Mehr hierzu erfahren Sie unter dem Kapitel Streamingplattformen, generell lässt sich jedoch sagen, dass im Zweifelsfall ein Server unter UNIX verwendet werden sollte, da diese die bessere Stabilität bieten. Leider sind nicht alle Multimediaserver unter UNIX lauffähig.

### 2.2.3 Hardware

Die Hardware des Servers spielt nur dann eine wichtige Rolle, wenn auf dem Server gleichzeitig auch die Encoder Software installiert ist. Neben genügend Festplattenplatz für On-Demand Streams ist für den Server ein hoher Datentransfer und eine gute Verbindung in ein Netzwerk wichtig, damit er neben den Streams auch den sonstigen Datenverkehr bewältigen kann.

### 2.2.4 Software

Generell ist es durchaus möglich eine multimediale Datei auf einem gewöhnlichen Web Server zu hinterlegen. Der überwiegende Teil an Videoformaten kann auch schon wiedergegeben werden während die Datei im Hintergrund noch geladen wird. Allerdings ist das Video in diesem Fall auf dem Rechner des Nutzers abgelegt. Dies ist allerdings nicht immer im Sinne des Inhabers. Soll dem Zuschauer das Speichern des Videomaterials nicht erlaubt (oder zumindest erschwert) werden, bietet sich es an das Material als On-Demand Stream zu Verfügung zu stellen. Zudem kann ein kontinuierlicher Datenstrom, wie er für größere Multimediateien erforderlich wäre, der außerdem noch an viele hundert Clients gleichzeitig gesendet werden muss, einen Web Server überlasten, da dieser für das Verteilen von kleinen bis mittelgroßen Dateien konzipiert ist. Zur vollen Nutzung der Möglichkeiten eines multimedialen Angebotes ist allerdings ein Streamingserver notwendig. So lassen sich Live Übertragungen nur mit einem solchen realisieren, ebenso lassen sich über dynamische Wiedergabelisten wesentlich flexiblere Inhalte ausstrahlen. Nicht zuletzt lassen sich über einen Streamingserver nicht nur On-Demand Inhalte verbreiten, sondern es lässt sich auch ein mit dem Fernsehen vergleichbares Szenario schaffen, bei dem sich der Zuschauer in einen kontinuierlichen Stream einklinken kann.

## 2.3 Client

### 2.3.1. Einführung

Der Client ist der Endnutzer, der Empfänger des Streams. Für ihn wurde der ganze Aufwand, von der Aufnahme des Videos, über das Encodieren, bis hin zum Aufstellen des Servers und

der Gestaltung einer Webseite betrieben. Daher lohnt es sich durchaus ein paar Gedanken dessen Hard- und Softwarevoraussetzungen zu widmen.

### *2.3.2 Hardware*

Die wichtigste Voraussetzung auf der Hardwareseite, als Empfänger von multimedialen Daten, ist die Anbindung an ein Netzwerk bzw. an das Internet. Dabei ist die Qualität der Daten direkt von der zur Verfügung stehenden Bandbreite abhängig. Ein 56k Modem kann maximal einen Datenstrom von 56 kbps empfangen. In der Regel liegt die tatsächliche Durchgangsrate deutlich darunter. Bei Streaming rechnet man als Faustregel 25% für analoge Geräte (Modem) und 10% für digitale Geräte (ISDN, DSL etc.) von der maximal möglichen Rate ab. Das wären bei einem 56k Modem also 42 kbps, was nicht sehr viel ist, wenn man sowohl Video als auch Audio in diesem Stream unterbringen möchte. Die 25% bzw. 10% Abzug sind durchaus berechtigt, denn neben dem Stream selber werden auch Kontrollfunktionen, Anwendungsinformationen und sonstige Daten über den Stream mitgeschickt. So wird bei jedem versendeten Byte (8 Bit) je ein Start- und Stoppsbit versendet, also pro Byte 10 Bit.

Achtung: Die so ermittelte Bandbreite ist die Obergrenze! Sollen neben dem Stream weitere Daten (Bilder, ein 2. Stream etc.) übermittelt werden, müssen diese mit eingerechnet werden!

Sollte eine ausreichende Netzwerkanbindung beim Empfänger vorhanden sein (z.B. DSL, LAN Anbindung), so kann es trotzdem zu Verzögerungen oder Qualitätseinbußen kommen, denn das Decodieren des Streams muss zeitgleich mit dem Empfangen und der Wiedergabe geschehen, so dass ein schwacher Prozessor unter Umständen schnell an seine Grenzen stößt. Hier helfen neben einem stärkeren Prozessor auch leistungsstarke Grafik- und Soundkarten, die den Hauptprozessor entlasten. Microsoft schlägt folgende Hardware als minimale Konfiguration für eine Datenrate bis zu 2 Mbps vor: Pentium II mit 233 MHz, 64 MB Arbeitsspeicher, High Performance VGA Grafikkarte (ab Baujahr 1998 oder später, insbesondere mit 3D Grafikunterstützung und eigenem RAM)

### *2.3.3 Software*

Während der Stream durchaus den Hardwarevoraussetzungen des Empfängers angepasst werden kann, stellt die Software hier das größere Problem dar. Denn abhängig vom Anbieter des Multimediaservers, wird auch auf Grund von Patenten dessen Encoder Software verwendet, so dass der Empfänger gezwungen ist auch das Player Plug-In des Herstellers zu verwenden. Leider ist aber nicht jeder Player auf jedem Betriebssystem verfügbar. So existiert für UNIX/Linux zur Zeit keine offizielle Version des Microsoft Media Players und für Apple Quicktime nur eine mangelhafte Betaversion. Alle streamingfähigen Player können auch in Browser eingebettet werden, und dem Anwender ein höchstmögliches Maß an Komfort zu bieten. Dieser Komfort wird aber wieder durch verschiedene Standards im Browser- und Playerkampf der Anbieter zunichte gemacht.

### *2.3.4 Zusammenfassung*

Der Anbieter muss sich also überlegen, welche Zielgruppe er ansprechen möchte und welche Bandbreite, Hardware und Software dieser Zielgruppe zur Verfügung steht. (Z.B. verwenden Mediziner häufig Macintosh Computer.) Davon abhängig kann er sich dann auf eine bestimmte Server- und Encodersoftware festlegen.

## 3. Streamingplattformen

### 3.1 Einführung

Wer sich mit dem Erstellen, Lagern und Abspielen von multimedialen Dateien im Internet beschäftigt, wird unweigerlich mit den Produkten von Real Networks, Apple und Microsoft konfrontiert werden, da diese Firmen sich die Marktführung in diesem Bereich streitig machen. Alle bieten mittlerweile jeweils ihr eigenes Konzept für Server, Encoder und Player an. Die Beschreibung des Funktionsumfanges und der Vor- und Nachteile wird Thema dieses Kapitels sein.

### 3.2 Real Networks

#### 3.2.1 Server

Real Networks bietet ihren Streamingserver in der Basic Version zum kostenlosen Download an, der aber maximal 25 Clients gleichzeitig versorgen kann. Der Realsystem Server Plus 8.0 und Realsystems Server Professional 8.0 sind kostenpflichtig und bedienen bis zu 65 Clients (Plus) bzw. ab 100 bis 2000 Clients (Professional). Vorteile: Real Networks ist für jedes gängige Betriebssystem zu haben und unterstützt alle gängigen Protokolle, wie unter anderem: RTP, RTSP, UDP und HTTP. Ein eigenes Proxy Kit für Firewalls wird neben anderen Besonderheiten ebenfalls angeboten.

#### 3.2.2 Encoder und Codec

Als Encoder bietet Real den Real Producer an, der sowohl in der Basic Version (kostenlos), als auch in der Plus Version (kostenpflichtig) existiert. Er konvertiert die Daten in die von Real patentierten Codecs RealAudio und RealVideo. Mit dem Producer Plus ist es auch möglich, multiple Streamraten in einer Datei zu platzieren (SureStream), und er unterstützt sowohl Live- als auch On-Demand Streaming in allen Versionen. Der Producer Basic unterstützt diese Möglichkeit nur eingeschränkt. Der Producer codiert die Streams in Echtzeit, was für Live Stream notwendig, für On-Demand Produktionen jedoch häufig Qualitätsverlust bedeutet. Real Producer Plus bietet auch die Möglichkeit einer Bandbreiten Simulation, um so das Endergebnis beim Empfänger zu kontrollieren. Real verwendet das \*.rm Format (auch: \*.ra für Audio und \*.rv für Video) für seine RealVideo - und RealAudio Codecs, darüber hinaus existieren die Formate RealPix für Bilder (Jpeg und Gif), RealText für Text und RealFlash für Shockwave Flash, die aber nur Container für gängige Formate darstellen. RealAudio und RealVideo Codecs sind von Real Networks lizenziert und nur mit dem Real Player abspielbar. Die Qualität der Real Codecs ist eher mittelmäßig. Besonders bei geringen Bitraten ist die Video und Audio Qualität schlechter als bei Windows Media.

#### 3.2.3 Player

Der Real Player steht in der Basic Version als kostenloser Download zur Verfügung, der Real Player Plus ist kostenpflichtig. Obwohl er keine anderen Möglichkeiten bietet als die anderen Player, so ist er doch auf allen gängigen Betriebssystemen problemlos einsetzbar. Die neueste Real Player Generation ist der RealOne Player, der aber sehr kontrovers diskutiert wird. Der Player versteht SMIL, mit dem dann die Präsentation gesteuert werden kann. Der große Vorteil von Real, gegenüber anderen Formaten, ist seine Kompatibilität mit verschiedenen Betriebssystemen: Er ist für alle gängigen Betriebssysteme (Unix/Linux, Macintosh und Windows) erhältlich.

## 3.3 Apple

### 3.3.1 Server

Der Quicktime Streaming Server (QTSS) ist Teil des Apple Mac OS X Server Betriebssystems. Er bietet den vollen Umfang eines Multimedia Servers an. Zur Übertragung wird RTP, RTSP oder HTTP verwendet. Durch Relayschaltung mehrerer Server kann eine (un)endliche Menge an Zuschauern, sowohl über Uni- als auch Multicast, bedient werden. Der QTSS Publisher ist ein Dienstprogramm zur einfachen Verwaltung der Wiedergabelisten auf dem Streaming Server. Für Live Übertragungen kann Quicktime Broadcaster verwendet werden, welcher auch mit dem gewöhnliche Mac OS X nutzbar ist. Der Unterbau des QTSS ist unter dem Namen Darwin Streaming Server (DSS) als Open Source Software unter der Apple Public Source Licence für verschiedene andere Systeme verfügbar. Der DSS kommt allerdings nur mit einer rudimentären Weboberfläche daher und kann nicht so komfortabel bedient werden wie der QTSS.

### 3.3.2 Encoder und Codec

Im Gegensatz zu Real und Microsoft bietet Apple kein spezielles Tool zum Encodieren von Multimedia Daten an, da dies mit dem Quicktime Pro Player möglich ist. Zum Encodieren von Videos verwendet Quicktime (unter anderem) den Sorenson Video 3 Codec und den QDesign Music Codec v1/v2, die in die Quicktime Architektur eingebettet sind. Quicktime Pro kann keine multiplen Bitraten erzeugen. Dafür muss eine gesonderte Software von Sorenson gekauft werden. Apple verwendet für Streaming kein eigenes Format, sondern benutzt einfach eine zusätzliche Spur für die Streaming Informationen in ihrem \*.mov Format. Standardgemäß produziert Quicktime Pro eine Auflösung von 160x120 und 6 Bilder pro Sekunde (fps). Mit dem Sorenson Codec entsteht so gute Qualität für ruhige bzw. Standbilder. Sobald aber Bewegung zu sehen ist wird das Bild schnell ruckelig und unscharf. Die Tonqualität reicht für verständliche Sprache, mehr leistet es jedoch nicht. Wenn die Verbindung schlecht ist, leidet auch die Qualität darunter, z.B. durch Paketverlust. Weiterhin ist Live Streaming nur mit dem kostenpflichtigen 'Media Broadcaster' von Sorenson realisierbar.

### 3.3.3 Player

Um sich Audio und Video Streams aus dem Netz anschauen zu können, bietet Apple den Quicktime Player an. Er ist die kostenlose Standardversion für Quicktime Formate (\*.mov) und unterstützt sowohl Live, als auch On-Demand Streaming mit RTP und RTSP Format. Daneben unterstützt er alle gängigen Video- und Audioformate, sowie die hauseigenen Produkte Quicktime VR (Virtual Reality Tool) und Apple Skript (vergleichbar mit VB für Windows), weiterhin Flash 4 und SMIL.

Um den vollen Leistungsumfang von Quicktime zu genießen, benötigt man die Pro Version, die allerdings kostenpflichtig ist. Sie bietet neben erweiterter Bedienung, wie z.B. Vollbild Modus, auch die Möglichkeit zum Erstellen und Schneiden von Video und Audio. Auch Encodieren ist nur mit der Pro Version möglich. Quicktime ist offiziell für Macintosh und Windows zu haben, unter UNIX/Linux existiert eine Beta Version des Players, daher bietet sich WINE oder das Crossover Plugin (kostenpflichtig).

## 3.4 Microsoft

### 3.4.1 Server

Der Microsoft Streaming Service baut auf dem Internet Information Server Version 4 (Windows NT Server), Version 5 (Windows 2000 Server) oder Version 6 (Windows Server 2003) auf und ist daher auf Windows 9x/Me/XP/Vista nicht lauffähig. Auch andere Betriebssysteme unterstützt die Software nicht. Im Gegensatz zu Real Networks und Apple nutzt Microsoft Media Streaming nicht das unabhängige RTP, sondern ein proprietäres Protokoll: MMS, das aber ähnliche Technologie verwendet wie RTP. MMS ist sowohl Uni- als auch Multicast fähig.

### 3.4.2 Encoder und Codec

Zum Aufbereiten der Videodaten bietet Microsoft den Windows Media Encoder als kostenlosen Download an. Der Encoder konvertiert die Videodaten in das Windows Media Video Format (WMV). Ebenso sind ein paar kleine Tools enthalten um die Metadaten vorhandener Videos zu bearbeiten und die Encodierungsprofile zu verändern. WMV und WMA sind eine proprietäre Weiterentwicklung des MPEG-4 Videoformates und stellen eine feste Größe in Sachen Kompression und Bildqualität dar. Die verwendeten Codecs sind unter anderem die hauseigenen Windows Media Video v7/v8/v9 für Video, für Audio Windows Media Audio und der Sipro ACELP.net. Mit dem Windows Media Encoder ist es auch möglich ein Videosignal zu encodieren und an einen Windows Media Streaming Server weiterzuleiten (Encoder Pull) um es von dort als Live-Stream zu senden.

### 3.4.3 Player

Der Microsoft Media Player ist aktuell in der Version 11 für Windows zu haben und ist zum Abspielen aller gängigen Formate fähig. Der Windows Media Player for Mac (Version 9) wurde Anfang 2006 von Microsoft eingestellt. Über das Anschauen und Anhören von Video- und Audiodaten hinaus bietet der Media Player die üblichen Spielereien, wie Skins und Audiovisualisierung. Darüber hinaus bietet er sich als Mediadatenverwalter an etc. Im Gegensatz zu den Playern von Real und Quicktime, versteht der Windows Media Player kein SMIL, was aber durch den Internet Explorer ab Version 5.5 (SP 2) unterstützt wird. Um Windows Media Dateien auf dem Mac abspielen zu können benötigt man nun das filp4Mac Decoder PlugIn für Quicktime welches kostenlos beim Hersteller oder den Mac-Webseiten von Microsoft heruntergeladen werden kann. UNIXe haben die Möglichkeit über WINE oder mit dem Crossover Plugin (kostenpflichtig) den Media Player zu starten. Als weitere Alternative für Linux Benutzer bietet sich der MPlayer an, welcher auch Unterstützung für Windows Media bieten soll.

## 3.5 Andere Angebote

Neben den soeben aufgeführten Anbietern gibt es natürlich auch andere Streaming Angebote. Als Beispiel sei kurz SHOUTcast erwähnt, ein MP3 Streaming Server von Nullsoft, die auch den bekannten Winamp Player anbieten. Das kostenlose SHOUTcast ist sowohl für On-Demand als auch Live Streaming einsetzbar und für die meisten Betriebssysteme zu haben. Als Encoder wird Lame verwendet. Je nach Betriebssystem empfiehlt Nullsoft einen anderen Player, für Windows natürlich den hauseigenen Winamp, für Linux den XMMS und für Macintosh iTunes. Im Rechenzentrum wird außerdem die Encoder Software 'Cleaner 5' verwendet und soll daher hier kurz vorgestellt werden:

### 3.5.1 Cleaner 5

Der Cleaner 5 ist eine kostenpflichtige On-Demand Encoder Software von Discreet (mittlerweile Autodesk). Sie bietet die Möglichkeit eine Präsentation (Video, Audio, aber auch Bilder und Text) in alle gängigen Formate, auch die proprietären von Real Networks, Apple und Microsoft, zu codieren. Im Gegensatz zum Real Producer und Windows Media Encoder wird aber nicht in Echtzeit codiert, was eine bessere Qualität der produzierten Streams zur Folge hat. Der Cleaner ist für den Batchbetrieb ausgelegt, so dass nicht nur Streams in verschiedenen Bandbreiten, sondern auch in verschiedenen Formaten 'am Stück' produziert werden können. Für jedes Format können die jeweils möglichen Einstellungen und Konfigurationen vorgenommen werden. Viele nicht lineare Videoschnitt Programme, zum Beispiel Premiere, liefern eine Light Version des Cleaners mit, die nicht den vollen Funktionsumfang bieten, aber für normale Stream Generierung ausreichend ist. Nachteilig fällt am Cleaner auf, daß er teilweise, abhängig von bestimmten Einstellungen, instabil läuft, was besonders bei einem größeren Batchjob sehr störend sein kann. Außerdem kann er keine Live Streams produzieren.

## 3.6 Übersicht

Im Folgenden werden die Player, Encoder und Streamingserver der einzelnen Anbieter nebeneinander dargestellt.

### 3.6.1 Player im Überblick

Name	Version	Unix	Mac	Win	SMIL	Bemerkung
Real Player Basic	8	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	Kostenlos, diverse Funktionen
Real Player Plus	8	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	Kostenpflichtig, diverse Funktionen
RealOne	1	<b>?</b>	<b>?</b>	<b>X</b>	<b>X</b>	Kostenlos oder kostenpflichtig, diverse Funktionen inkl. Jukebox
QuickTime	5	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	Kostenlos, diverse Funktionen
QuickTime Pro	5	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	Kostenpflichtig, einfache Clipbearbeitung, Encodierung, diverse Funktionen
Windows Media Player	7	<b>---</b>	<b>X</b>	<b>X</b>	<b>---</b>	Kostenlos, diverse Funktionen

### 3.6.2 Encoder im Überblick

Name	Codec	Format	Multiple Bitraten	Live Streaming	On-Demand Streaming	Bemerkungen
<b>Real Producer Basic</b>	RealAudio RealVideo	RealAudio RealVideo (*ra; *.rv; *.rm)	<b>X</b> (eingeschränkt)	<b>X</b>	<b>X</b>	Kostenlos, codiert nur mit aktuellem Codec
<b>Real Producer Plus</b>	RealAudio RealVideo	RealAudio RealVideo (*ra; *.rv; *.rm)	<b>X</b>	<b>X</b>	<b>X</b>	Kostenpflichtig, einfache Clipbearbeitung, Bitratensimulation
<b>Apple Quicktime Pro</b>	Sorenson Video 3 QDesign Music	Quicktime (*mov)	--- (extra Software)	--- (extra Software)	<b>X</b>	Kostenpflichtig, einfache Clipbearbeitung
<b>Windows Media Encoder</b>	WM Video 8 WM Audio	Windows Media (*wmv; *.wma; *.asf)	<b>X</b>	<b>X</b>	<b>X</b>	Kostenlos, einfache Clipbearbeitung, Bitratensimulation
<b>Discreet Cleaner5</b>	Alle und ein paar mehr	Alle und ein paar mehr	<b>X</b>	---	<b>X</b>	Kostenpflichtig, einfache Clipbearbeitung

### 3.6.3 Streamingserver im Überblick

Name	Protokolle						Maximale Clients	Bemerkungen
	UDP	RTP	RTSP	HTTP	SDP	MMS		
<b>Apple Darwin</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>?</b>	<b>X</b>	---	3000	Kostenlos
<b>Apple Quicktime</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>?</b>	---	3000	Kostenlos; nur für Mac OS X Server
<b>Real Server Basic</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	---	---	25	Kostenlos
<b>Real Server Plus</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	---	---	65	Kostenpflichtig
<b>Real Server Professional</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	---	---	2000	Kostenpflichtig
<b>Microsoft Streaming Service</b>	<b>X</b>	---	---	<b>X</b>	---	<b>X</b>	3000	Kostenlos; nur für Windows Server

## 4. Streaming

### 4.1 Einführung

Streaming bezeichnet das Abspielen von Multimediadateien schon während des Downloads. Im Unterschied zu diesem werden die Daten jedoch nicht lokal auf dem Computer gespeichert. Von der Aufnahme der Daten bis zum Betrachten selbiger über das Internet, sind 3 Schritte notwendig, welche in diesem Kapitel behandelt werden:

1. Aufnehmen: die nötigen Kenntnisse können unter Digitale Audio- und Videobearbeitung erworben werden.
2. Encodieren: die Daten werden mit Encodersoftware komprimiert, damit sie auch über ein Netzwerk gesendet werden können.
3. Senden: nachdem die Daten encodiert wurden, können sie von einem Server an die Clients verschickt werden.

### 4.2 Encodieren

Beim Streaming wird ein sog. Stream vom Server zum Empfänger gesendet. Dieser setzt sich sowohl aus den Video und Audio Daten, sowie verschiedenen Kontrollinformationen zusammen. Damit der Empfänger der Daten den Stream ohne Probleme (Ruckeln, Verzögerung beim Abspielen etc) anhören oder sehen kann, müssen die Audio und Video Daten der verfügbaren Bandbreite angepasst werden, indem sie komprimiert werden. Die Komprimierung der Daten (Audio und Video) erfolgt mit Encoder Programmen. Dabei kann das Material live codiert und direkt gesendet werden (Live Streaming) oder eine Mediendatei erzeugt werden, die hinterher auf dem Server hinterlegt wird (On-Demand Streaming). Diese kann dann zum Download oder als Stream auf dem Server abgelegt werden. Die Encodierung der Audio und Video Daten ist je nach Bandbreite mehr oder weniger verlustbehaftet. Daher muss sich der Produzent überlegen, welche Daten er mehr gewichten kann und wieviel Datenverlust akzeptabel für eine gute Präsentation ist. Dies ist je nach Art der Präsentation unterschiedlich. Bei einem On-Demand Lehrgang z.B. ist eine deutliche Sprache in der Regel wichtiger als ein klares Bild, die entsprechenden Folien etc. könnten hinterher als Grafiken auf der Seite eingebunden werden (Hierzu ein [Beispiel](#)). Bei der Übertragung eines Konzertes ist auf eine gute Mischung zwischen Ton und Bild zu achten, während ein Landschaftsfilm ein gutes Bild benötigt. Als Faustregel sollte darauf geachtet werden, dass max. 75% der zur Verfügung stehenden Bandbreite für analoge Übertragung und 90% der Bandbreite für digitale Übertragung des Clients genutzt werden, was bei geringen Bandbreiten zu einem Problem werden kann.

#### 4.2.1. Bandbreite

Hier die üblichen Verbindungen und ihre maximale Streaming Raten, nach einer Empfehlung von Real.

<b>Maximale Streamingraten (Angaben nach Real)</b>	
<b>Verbindung</b>	<b>Streamingraten</b>
14.4 Kbps Modem	10 Kbps
28.8 Kbps Modem	20 Kbps
56 Kbps Modem	34 Kbps
64 Kbps single ISDN	45 Kbps
112 Kbps dual ISDN	80 Kbps
LAN	150 Kbps
256 Kbps DSL/cable Modem	225 Kbps
384 Kbps DSL/cable Modem	350 Kbps
512 Kbps DSL/cable Modem	450 Kbps

Dies sind keine festen oder notwendigen Vorgaben, sondern Erfahrungswerte, die speziell auf Real Codecs abgestimmt sind. Sie können sowohl von Anwendung zu Anwendung, als auch von Anbieter zu Anbieter unterschiedlich sein, und sollten daher nur als Richtwert genutzt werden. Die optimalen Raten für die persönlichen Bedürfnisse können nur durch Experimentieren gefunden werden.

#### 4.2.1.1 Audio

Die folgende Tabelle zeigt Standard Datenraten für reine Audiostreams:

<b>Standard Datenraten für (Real) Audio ohne Video</b>				
<b>Zielgruppe</b>	<b>Sprache</b>	<b>Sprache und Musik</b>	<b>Mono Musik</b>	<b>Stereo Musik</b>
<b>28.8 Kbps Modem</b>	16 Kbps	16 Kbps	20 Kbps	20 Kbps
<b>56 Kbps Modem</b>	16 Kbps	32 Kbps	32 Kbps	32 Kbps
<b>64 Kbps single ISDN</b>	32 Kbps	32 Kbps	44Kbps	44Kbps
<b>112 Kbps dual ISDN</b>	64 Kbps	64 Kbps	64 Kbps	64 Kbps
<b>LAN</b>	64 Kbps	64 Kbps	64 Kbps	96 Kbps
<b>256 Kbps DSL/cable Modem</b>	64 Kbps	64 Kbps	64 Kbps	96 Kbps
<b>384 Kbps DSL/cable Modem</b>	64 Kbps	64 Kbps	64 Kbps	96 Kbps
<b>512 Kbps DSL/cable Modem</b>	64 Kbps	64 Kbps	64 Kbps	96 Kbps

Und hier die Audiodatenraten, wenn auch ein Video im Stream eingebettet ist.

<b>Standard Datenraten für (Real) Audio mit Video</b>					
Zielgruppe	Clip	Sprache	Sprache und Musik	Mono Musik	Stereo Musik
<b>28.8 Kbps Modem</b>	20 Kbps	5 Kbps	6,5 Kbps	8 Kbps	8 Kbps
<b>56 Kbps Modem</b>	34 Kbps	5 Kbps	6,5 Kbps	8 Kbps	8 Kbps
<b>64 Kbps single ISDN</b>	45 Kbps	8,5 Kbps	8,5 Kbps	11 Kbps	11 Kbps
<b>112 Kbps dual ISDN</b>	80 Kbps	8,5 Kbps	8,5 Kbps	16 Kbps	20 Kbps
<b>LAN</b>	150 Kbps	32 Kbps	32 Kbps	32 Kbps	32 Kbps
<b>256 Kbps DSL/cable Modem</b>	225 Kbps	32 Kbps	32 Kbps	44 Kbps	44 Kbps
<b>384 Kbps DSL/cable Modem</b>	384 Kbps	32 Kbps	32 Kbps	44 Kbps	64 Kbps
<b>512 Kbps DSL/cable Modem</b>	450 Kbps	32 Kbps	32 Kbps	64 Kbps	96 Kbps

Auch hier gilt, dass es sich nicht um feste Vorgaben handelt, sondern um Empfehlungen.

#### 4.2.1.2 Video

Die Datenraten für Video sind stark abhängig von der verwendeten Bildgröße. Wir geben hier nur die Video Qualität in Abhängigkeit von der Bildgröße an, um einen Überblick zu verschaffen. Je nach Videomaterial und verwendetem Codec ist eine andere Videogröße empfehlenswerter.

<b>Video Qualität in Abhängigkeit von der Bildgröße</b>					
Verbindung	Video Daten	176 x 132	240 x 180	320 x 240	640 x 480
<b>28.8 Kbps Modem</b>	20 Kbps	hervorragend	gut	brauchbar	schlecht
<b>56 Kbps Modem</b>	34 Kbps	hervorragend	gut	brauchbar	schlecht
<b>64 Kbps single ISDN</b>	45 Kbps	hervorragend	gut	gut	schlecht
<b>112 Kbps dual ISDN</b>	80 Kbps	hervorragend	hervorragend	hervorragend	brauchbar
<b>LAN</b>	150 Kbps	hervorragend	hervorragend	hervorragend	gut
<b>256 Kbps DSL</b>	225 Kbps	hervorragend	hervorragend	hervorragend	hervorragend
<b>384 Kbps DSL</b>	350 Kbps	hervorragend	hervorragend	hervorragend	hervorragend
<b>512 Kbps DSL</b>	450 Kbps	hervorragend	hervorragend	hervorragend	hervorragend

Auch hier gilt, dass es sich nicht um feste Vorgaben handelt, sondern um Empfehlungen.

### 4.2.2 Sonstige Einstellungen

Neben der Beachtung der Datenraten, können im Stream noch viele weitere Einstellungen vorgenommen werden. Da diese sich aber von Anbieter zu Anbieter unterscheiden und auch teilweise voneinander abhängig sind, gibt es hierzu keine vorgegebenen 'Kochrezepte'. Die Einstellungen müssen von Fall zu Fall erneut getestet werden. Die wichtigsten Möglichkeiten sollen aber kurz am Beispiel von Real und Windows Media vorgestellt werden. Grundlagenwissen zu vielen Einstellungen erfahren Sie unter Digitale Audio- und Videobearbeitung.

#### 4.2.2.1 Audio

- **Codec:** der verwendete Audiocodec kann die Qualität der Audiodaten stark beeinflussen. Z.B. ist der ACLEP.net hervorragend für niedrige Bandbreiten und Stimme geeignet.
- **Sampling Rate:** Anzahl der Messpunkte zur Digitalisierung. Je höher die Abtastrate, desto höher die Qualität, aber auch die Größe der Datei. Mehr finden Sie unter Digitale Audiobearbeitung.
- **Datenrate:** abhängig von der Bandbreite des Endnutzers kann die Datenrate eingestellt werden. Je niedriger die Datenrate, desto höher ist der Qualitätsverlust der Audiodaten. Beachten Sie, dass evt. auch Videodaten mit im Stream übertragen werden.
- **Kanäle:**  
Die verwendeten Kanäle. Stereo verwendet 2 Kanäle und benötigt daher die doppelte Datenmenge als Mono.

Weitere Einstellungen für Audio (nach RRZK)								
Datenrate	Codec		Samplingrate		Datenrate		Kanäle	
	Real	Windows Media	Real	Windows Media	Real	Windows Media	Real	Windows Media
<b>28.8 Kbps Modem</b>	Real Audio Voice	ACLEP.net	?	8.000 kHz	6,5 kbps	6,5 kbps	Mono (?)	Mono
<b>56 Kbps Modem</b>	Real Audio Voice	ACLEP.net	?	8.000 kHz	8,5 kbps	6,5 kbps	Mono (?)	Mono
<b>64 Kbps single ISDN</b>	Real Audio Voice	ACLEP.net	?	16.000 kHz	16 kbps	16 kbps	Mono (?)	Mono
<b>112 Kbps dual ISDN</b>	Real Audio Voice	Windows Media Audio V8	?	22.050 kHz	16 kbps	22 kbps	Mono (?)	Stereo
<b>LAN</b>	Real Audio Voice	Windows Media Audio V8	?	22.050 kHz	32 kbps	32 kbps	Mono (?)	Stereo
<b>320 kbps</b>	Real Audio Voice	Windows Media Audio V8	?	44.100 kHz	64 kbps	64 kbps	Mono (?)	Stereo

#### 4.2.2.2 Video

- **Codec:** Der verwendete Videocodec kann die Qualität der Videodaten stark beeinflussen.
- **Bildgröße:** Die Größe des Bildes beeinflusst auch die Datenmenge.
- **Keyframe:** Gibt die Anzahl der Frames zwischen den einzelnen Keyframes an. Es handelt sich um die Voreinstellungen der Hersteller, welche aber der Theorie widersprechen. Der Cleaner 5 gibt als Faustregel an, daß bei hohen Datenrate (ungefähr ab DualISDN) ein Keyframe jede Sekunde (entspricht Mindestraten für Videoschnitt) und bei niedrige Datenraten ein Keyframe alle 5 Sekunden gesendet werden kann.
- **Frames pro Sekunde:** Je mehr Einzelbilder in einer Sekunde gezeigt werden, desto fließender ist das Bild. Dies erhöht aber auch die Datenmenge.
- **Datenrate:** Abhängig von der Bandbreite des Endnutzers kann die Datenrate eingestellt werden. Je niedriger die Datenrate, desto höher ist der Qualitätsverlust der Videodaten. Beachten Sie, dass eventuell auch Audiodaten mit im Stream übertragen werden.

Weitere Einstellungen für Video (nach RRZK)										
Datenrate	Codec		Bildgröße		Keyframe		Frames pro Sekunde		Datenrate	
	Real	Windows Media	Real	Windows Media	Real	Windows Media	Real	Windows Media	Real	Windows Media
<b>28.8 Kbps Modem</b>	Real Video 8	Windows Media Video V8	176x140	160x120	100	60	6	7,5	12	12
<b>56 Kbps Modem</b>	Real Video 8	Windows Media Video V8	176x140	176x140	80	80	10	10	28	28
<b>64 Kbps single ISDN</b>	Real Video 8	Windows Media Video V8	240x180	176x140	80	80	10	10	34	35
<b>112 Kbps dual ISDN</b>	Real Video 8	Windows Media Video V8	240x180	240x180	80	80	15	15	58 kbps	64 kbps
<b>LAN</b>	Real Video 8	Windows Media Video V8	240x180	320x240	120	120	15	15	200 kbps	200 kbps
<b>320 kbps</b>	Real Video 8	Windows Media Video V8	320x240	320x240	120	120	15	15	320 kbps	320 kbps

#### 4.2.2.3 Andere Einstellungen

Viele Einstellungen sind optional, die wichtigsten sind:

- Metafile: erzeugt \*.ram und \*.asx Dateien
- Seitenverhältnisse: Gibt Größe des Bildes an. (4:3; 16:9; wie Quelle; benutzerdefiniert)
- Wasserzeichen: Eine Möglichkeit um Videos zu markieren und Logos im Video anzuzeigen.

- multiple Streamraten: Mehrere Streams werden in eine Datei verpackt. Häufig wird nur eine Audiospur für alle Datenraten verwendet.
- In- und Outpoint: Setzt den Beginn und das Ende der Encodierung im Clip fest. Praktisch, wenn der Clip nicht geschnitten werden soll.
- Metadata: Angaben über Clip, Darsteller, Author etc.
- Diverse Video- und Audio Filter
- Etc. pp.

### 4.3 Senden

Um die encodierten Streamingdaten auch empfangen zu können, müssen sie von einem Server an die Clients verschickt werden. Prinzipiell ist es auch möglich, eine beliebige Mediendatei auf einem Webserver zu hinterlegen, so dass eine Übertragung der Daten per HTTP erfolgen würde. Dies hat jedoch auch Nachteile. So wird die Datei komplett auf die Festplatte des Empfängers heruntergeladen, was für manche Daten, z.B. aus urheberrechtlichen Gründen, nicht immer wünschenswert ist. Das Vor- bzw. Zurückspringen auf einen bestimmten Punkt (z.B. an einen Abschnittsanfang), 'Marker' genannt, ist nicht möglich ohne die ganze Datei zu übertragen und das Vor- und Rückspulen ist zwar machbar, aber umständlich. Auch benötigt man eine geschlossene Datei, auf die der Server zurückgreifen kann. Deshalb ist Live Streaming über einen Webserver nicht möglich, da der Stream ein kontinuierlicher Datenstrom und keine geschlossene Datei im eigentlichen Sinne ist. Ein weiterer Nachteil ist die Beschaffenheit des Webservers und die Art der Transportprotokolle. Der Webserver ist zum Herunterladen kompakter Inhalte (html, gif oder jpg) konzipiert. TCP ist so konzipiert, dass jedes Datenpaket quittiert und bei Fehlerhaftigkeit erneut verlangt und gesendet wird, so dass die Geschwindigkeit stark gebremst wird und die Belastung des Netzes sehr hoch ist, wenn die Verbindung schlecht ist. Als Alternative bietet sich ein Streaming Server an, da dieser für die Verteilung großer Datenmengen am Stück konzipiert ist und die Protokolle bei fehlerhaften Daten keine erneute Abfrage durchführen. Dadurch wird die Netzlast verringert. Die auftretenden Übertragungsfehler werden vom Player korrigiert, zum Beispiel durch Interpolation bei fehlenden Bildinformationen. Der Server ist darüber hinaus auch zu Live Streaming und Multicast fähig. Auf dem Server werden die Daten mit einem 'Mountpoint' bzw 'Alias', in Form eines oder mehrerer Unterverzeichnisse(s), verknüpft, über das sie abgerufen werden können.

protokoll://servername.domain.de/mountpoint

Probleme können auftauchen, wenn Client oder Server hinter einer Firewall stehen. Eventuell muss die Portnummer und das Übertragungsprotokoll konfiguriert werden, da die meisten Streamingprotokolle und Portnummern von Firewalls geblockt werden. Der Server kann in der Regel so konfiguriert werden, dass er die Daten in HTTP Paketen verschickt. Der Stream genießt aber trotzdem die Vorteile eines Streamingsservers. Manche Server bieten ein Protokoll Rollover an. Das heißt, dass der Server nacheinander verschiedene Protokolle ausprobiert, bis ein Protokoll funktioniert.

#### 4.3.1 On-Demand Produktionen

Beim On-Demand Streaming wurden die Daten vom Encoder vorgeneriert und liegen als Datei auf dem Encoderrechner. Die Streamingdatei wird auf dem Streamingserver in ein Verzeichnis kopiert und kann dort per Link aufgerufen werden.

protokoll://servername.domain.de/pfad/da.tei

Ein **Beispiel**:

rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/kleinesvideo.rm

- rtsp - das verwendete Übertragungsprotokoll;
- zaik-rm - der Real Media Streamingserver des ZAIK;
- rrz - die Subdomain des [Kölner Rechenzentrums](#)
- uni-koeln.de - Domain der [Universität zu Köln](#)
- rrzk/streaming/dokumentation/ - der Pfad, in welcher die Datei hinterlegt wurde;
- kleinesvideo.rm - die Streamingdatei, hier eine RealMedia Datei

#### 4.3.2 Live Produktionen

Da ein Live Stream keine geschlossene Datei im eigentlichen Sinne ist, kann er nicht auf einem Webserver gespeichert werden, sondern muß über einen Streamingserver vom Encoder weitergeleitet werden. Dieser sollte, um die Leistung zu optimieren, auf einem separaten Rechner laufen. Der Server holt sich die Daten über die IP und Port des Encoderrechners und streamt sie dann in das angeschlossene Netzwerk. Da der Live Stream keine geschlossene Datei ist, wird ein sogenannter Mountpoint, in Form eines Unterverzeichnisses eingerichtet, über den der Live-Stream abrufbar ist. Der typische Aufbau eines Mountpoints:

protokoll://servername.domain.de/mountpoint

Als Beispiel für einen Live Stream soll hier der Radiosender [Kölncampus](#) dienen, der auch [online](#) 'sendet'. Die ASX Datei, die hier aufgerufen wird, ist eine sogenannte Metadatei ( siehe Kapitel [6. Metadateien ASX, RAM und SMIL](#)), die schließlich auf den Mountpoint des Videoservers im Rechenzentrum verweist:

http://zaik-wm.rrz.uni-koeln.de/koelncampus-live

- http - das verwendete Übertragungsprotokoll
- zaik-wm.rrz.uni-koeln.de - Der Windows Media Streamingserver des Rechenzentrums
- /koelncampus-live - der Mountpoint des Live Streams

## 5. Player einbinden in eine Webseite

Vorweg: Die hier aufgeführten Beispiele sind für den Real Player geschrieben worden, da dieser für die meisten Betriebssysteme erhältlich ist. Bei Verwendung mit anderen Playern, z.B. mit dem Windows Media Player, kann der Quellcode sich ändern, da sie unterschiedliche Anforderungen an das <OBJECT> bzw. <EMBED> Tag stellen. Eine einführende Dokumentation bieten die SDK's der Encoder bzw. Player des jeweiligen Anbieters oder dessen Supportdatenbank bzw. Hilfe Seiten. Links hierzu finden Sie im Literaturverzeichnis. **Achtung:** Die meisten Hersteller beziehen sich bei der Beschreibung des <OBJECT> Tags nur auf die ActiveX Kontrollen (siehe Kapitel [5.2.2 ActiveX](#) ) Über die Einbindung des

Players mit dem MIME-Type wird nur in Bezug auf das <EMBED> hingewiesen. Dies hat sich leider durch die Einführung eigener Standards, wie z.B. das <EMBED> Tag, welches von Netscape stammt, oder ActiveX von Microsoft, im Browserkampf so eingebürgert.

Sie benötigen also den Real One Player, welchen Sie [hier](#) herunterladen können, um sich die Beispiele anschauen zu können.

Es gibt 3 Möglichkeiten einen Stream über eine Webseite aufzurufen: Das einfache Einbinden per Link ( siehe Kapitel [5.1 Einbinden per Link](#) ), das Einbetten per <OBJECT> Tag, welches am häufigsten verwendet wird, oder das Einbetten per <EMBED> Tag, wovon Netscape Navigator und Opera Gebrauch machen. Am Ende dieses Kapitels ist eine Möglichkeit aufgeführt <Object> und <EMBED> Tag miteinander zu verknüpfen.

## 5.1 Einbinden per Link

Die einfachste Methode eine Multimedia Datei in eine Webseite einzubinden, die bereits im vorangegangenen Kapitel gezeigt wurde, ist ein gewöhnlicher Link. Obwohl einfach, ist dies nicht sehr elegant, da sich der Player separat in einem Fenster öffnet.

Anmerkung: Laut Konvention sollten Tags klein geschrieben werden, zur Verdeutlichung sind sie hier aber in Großbuchstaben gehalten.

Syntax:

```
<A HREF="URL">Link</A>
```

Erläuterung:

- HREF="URL" - Die URL über die der Clip angesprochen werden kann.

Beispiele mit einer [WindowsMedia Datei](#)

```
<A HREF="http://ZAIK-WM.RRZ.UNI-KOELN.DE/rrzk/streaming/dokumentation/kleinesvideo.wmv">
Mit einer WindowsMedia Datei</A>
```

und einer [Realmedia Datei](#)

```
<A HREF="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/kleinesvideo.rm">
Realmedia Datei</A>
```

## 5.2 Einbinden per <OBJECT> Tag

Die Alternative zum Link ist das <OBJECT> Tag, mit dem jede Datei oder Datenquelle, die sich außerhalb der HTML Datei befindet, in eine Seite eingebunden werden kann. Dazu gehört auch ein Media Player. Das <OBJECT> Tag bettet den Player und seine Steuerelemente direkt in die Seite ein und kann von dort aus gesteuert werden. Das Tag kann in 2 Varianten eingesetzt werden: Mit Angabe des MIME-Types oder der ClassID. Die Angabe des MIME-Types ist W3C Standard wird aber nicht vom Internet Explorer interpretiert.

Je nach Player sind unterschiedliche Angaben erforderlich, die genauen Anforderungen können Sie beim Hersteller finden ( siehe ).

### 5.2.1 MIME-Type

Die <PARAM> Tags geben Möglichkeiten zur Parameterübergabe, wobei der Aufbau immer das Muster NAME="Parameter Name" VALUE="Parameter Inhalt" hat.

Die MIME-Types der wichtigsten Player (Plug-Ins):

Real Player                    audio/x-pn-realaudio-plugin

Windows Media Player video/x-ms-wmv

Quicktime                    video/quicktime

OBJECT Syntax I:

```
<OBJECT DATA="URL" TYPE="MIME-Type" WIDTH="Breite" HEIGHT="Höhe">
```

```
Alternativer Inhalt
```

```
</OBJECT>
```

Erläuterung:

- TYPE="mime-type" - Die genaue Angabe des MIME-Types
- DATA="URL" - Die URL über die der Clip angesprochen werden kann.
- WIDTH="Breite" - Die Breite des gesamten eingebundenen Objektes inklusive Steuerelementen.
- HEIGHT="Höhe" - Die Höhe des gesamten eingebundenen Objektes inklusive Steuerelementen.
  - Alternativer Inhalt - Hier können weitere Tags stehen, falls das <OBJECT> Tag vom Browser nicht interpretiert wird.

Hier ein Beispiel für die Verwendung des MIME-Types mit dem Real Player:

Syntax des Beispiels:

```
<OBJECT TYPE="audio/x-pn-realaudio-plugin" DATA="playerinfo/kleinesvideo.rpm"
CONTROLS="ImageWindow,ControlPanel,StatusBar" HEIGHT="228" WIDTH="236">
<b>Einbinden des Real Players per MIME-Type. Wenn Sie diesen Text sehen,
kann Ihr Browser das <OBJECT> Tag mit MIME-Type nicht richtig darstellen
oder versteht das <OBJECT> Tag nicht.</b>
</OBJECT>
```

Erläuterung:

- TYPE="audio/x-pn-realaudio-plugin" - Der MIME Type für das Real Media Player Plug-In.
- DATA="playerinfo/kleinesvideo.rpm" - Die URL über die der Clip angesprochen werden kann.
- CONTROLS="ImageWindow,ControlPanel,StatusBar" - Angabe von Steuerelementen für den Realplayer. Dieses Attribut ist nur für den Real Player möglich. Andere Player nutzen andere Möglichkeiten.
- WIDTH="Breite" - Die Breite des gesamten Real Players inklusive Steuerelementen.
- HEIGHT="Höhe" - Die Höhe des gesamten Real Players inklusive Steuerelementen.
  - <b>Einbinden des Real Players ...</b> - Alternativer Text, der angezeigt werden soll, wenn das <OBJECT> Tag nicht dargestellt werden kann.

### 5.2.2 ActiveX

Eine andere Möglichkeit der Einbettung von Playern funktioniert über die ActiveX Programmierschnittstelle von Microsoft. ActiveX ist kein W3C Standard und wird daher nicht

von allen Browsern und Betriebssystemen unterstützt, trotzdem soll diese Option hier vorgestellt werden, da sie häufig verwendet wird. Im Zweifelsfalle sollten Sie die OBJECT Syntax I nutzen.

Um einen Player mit ActiveX einzubinden gibt man die ClassID an, welche ein bestimmtes ActiveX Steuerelement, z.B. einen Player, referenziert.

Die ClassID der wichtigsten Player:

Real Player                      clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA

Windows Media Player clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95

Quicktime                      clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B

OBJECT Syntax II:

```
<OBJECT WIDTH="Breite" HEIGHT="Höhe" CLASSID="CLSID:XXXXXX-XXXX-XXXXXX">
<PARAM NAME="Name" VALUE="Wert">
</OBJECT>
```

Erläuterung:

- CLASSID="clsid:XXXXXX-XXXX-XXXXXX" - Referenziert ein bestimmtes ActiveX Steuerelement, z.B. einen Player
- WIDTH="Breite" - Die Breite des eingebundenen Objektes.
- HEIGHT="Höhe" - Die Höhe des eingebundenen Objektes.
  - PARAM - Dieses Element gibt verschiedene Parameter für das Objekt an.
  - NAME="Name" - Gibt die Bezeichnung des Parameters an. (Zum Beispiel: autostart; src)
  - VALUE="ImageWindow" - Gibt den Wert des Parameters an. (Zum Beispiel: true|false; URL)

Hier ein Beispiel für die Verwendung der ClassID mit dem Real Player:

Syntax des Beispielles:

```
<OBJECT ID=ImageWindow CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA"
HEIGHT=192 WIDTH=236>
<PARAM NAME="console" VALUE="Clip1">
<PARAM NAME="src" VALUE="playerinfo/kleinesvideo.rpm">
<PARAM NAME="controls" VALUE="ImageWindow">
<b>Einbinden des Real Players per ActiveX. Wenn Sie diesen Text sehen,
kann Ihr Browser das <OBJECT> Tag mit ActiveX nicht richtig darstellen
oder versteht das <OBJECT> Tag nicht.</B>
</OBJECT>
<OBJECT ID=ControlPanel CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA HEIGHT=36
WIDTH=236>
<PARAM NAME="console" VALUE="Clip1">
<PARAM NAME="controls" VALUE="ControlPanel">
</OBJECT>
<OBJECT ID=InfoPanel CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA HEIGHT=55
WIDTH=236>
<PARAM NAME="console" VALUE="Clip1">
<PARAM NAME="controls" VALUE="InfoPanel">
</OBJECT>
```

Eine Besonderheit des Real Players ist, dass sämtliche Kontrollelemente, wie Fenster, Control Panel etc., je ein eigenes <OBJECT> Tag benötigen. Um die einzelnen <OBJECT> Tags aufeinander zu beziehen wird jeweils ein *CONSOLE* Parameter definiert, über dessen Wert in VALUE die <OBJECT> Tags verbunden sind. In unserem Beispiel sind 3 dieser Tags

miteinander verbunden: das ImageWindow, in welchem der Clip abgespielt wird, das ControlPanel, das die wichtigsten Steuerelemente anzeigt, und das InfoPanel, das Informationen über den Clip gibt.

- 1. <OBJECT> Tag
  - ID=ImageWindow - Name des ersten <OBJECT> Tags
  - CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA" - ClassID des Real Players
  - HEIGHT=192 WIDTH=236 - Höhe und Breite des Objektes, die Breite ist bei allen dreien gleich, damit sie optisch zueinanderpassen.
  - PARAM Werte:
    - NAME="console" VALUE="Clip1" - Der Parameter, der die einzelnen <OBJECT> Tags verbindet, er ist daher bei allen gleich.
    - NAME="src" VALUE="playerinfo/kleinesvideo.rpm" - Die URL über die der Clip angesprochen werden kann.
    - NAME="controls" VALUE="ImageWindow" - Gibt an, daß ein Videofenster angezeigt werden soll.
    - <b>Einbinden des Real Players ...</b> - Alternativer Text, der angezeigt werden soll, wenn das <OBJECT> Tag nicht dargestellt werden kann.
- 2. <OBJECT> Tag
  - ID=ControlPanel - Name des zweiten <OBJECT> Tags
  - CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA" - (wieder) die ClassID des Real Players
  - HEIGHT=36 WIDTH=236 - Höhe und Breite des Objects, die Breite ist bei allen dreien gleich, damit sie optisch zueinanderpassen.
  - PARAM Werte:
    - NAME="console" VALUE="Clip1" - Der Parameter, der die einzelnen <OBJECT> Tags verbindet, er ist daher bei allen gleich.
    - NAME="controls" VALUE="ControlPanel" - Gibt an, daß ein Panel mit den wichtigsten Steuerfunktionen angezeigt werden soll.
- 3. <OBJECT> Tag
  - ID=InfoPanel - Name des dritten <OBJECT> Tags
  - CLASSID="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA" - (immer noch) die ClassID des Real Players
  - HEIGHT=55 WIDTH=236 - Höhe und Breite des Objects, die Breite ist bei allen dreien gleich, damit sie optisch zueinanderpassen.
  - PARAM Werte:
    - NAME="console" VALUE="Clip1" - Der Parameter, der die einzelnen <OBJECT> Tags verbindet, er ist daher bei allen gleich.
    - NAME="controls" VALUE="InfoPanel" - Gibt an, daß eine Leiste mit Informationen über den Clip angezeigt werden soll.

### 5.3 Einbinden per <EMBED> Tag

Als dritte Möglichkeit ein Video einzubinden, ist das <EMBED> Tag, welches von Netscape für den Communicator eingeführt wurde und das auch vom Opera Browser anstelle des <OBJECT> Tags genutzt wird. Dies entspricht aber nicht den W3C Richtlinien. Da sowohl der Netscape Communicator als auch der Opera weit verbreitet sind, soll daher auch dieses Tag vorgestellt werden. Netscape 6 versteht dieses Tag zwar noch, richtet sich aber vorrangig nach dem <OBJECT> Tag.

EMBED Syntax:

```
<EMBED SRC="URL" TYPE="MIME-Type" WIDTH="Breite" HEIGHT="Höhe">
```

Erläuterung:

- SRC="URL" - Die URL über die der Clip angesprochen werden kann.
- TYPE="MIME-Type" - Angabe des MIME-Types. Diese Attribut ist nicht obligatorisch, erleichtert aber dem Browser die Erkennung des Plug-Ins.
- WIDTH="Breite" HEIGHT="Höhe" - Legt die Höhe und Breite des <EMBED> Tags fest.

Ein Beispiel für die Verwendung des <EMBED> Tags mit dem Real Player:

Syntax des Beispiels:

```
<EMBED SRC="playerinfo/kleinesvideo.rpm" TYPE="audio/x-pn-realaudio-plugin"
CONTROLS="ImageWindow,ControlPanel,StatusBar" width="236" height="228">
```

Erläuterung:

- SRC="playerinfo/kleinesvideo.rpm" - Die URL über die der Clip angesprochen werden kann.
- TYPE="audio/x-pn-realaudio-plugin" - Angabe des MIME-Types. Diese Attribut ist nicht obligatorisch, erleichtert aber dem Browser die Erkennung des Plug-Ins.
- CONTROLS="ImageWindow,ControlPanel,StatusBar" - Angabe von Steuerelementen für den Realplayer. Dieses Attribut ist nur für den Real Player möglich. Andere Player nutzen andere Möglichkeiten.
- WIDTH="236" - Die Breite des gesamten Real Players inklusive Steuerelementen.
- HEIGHT="228" - Die Höhe des gesamten Real Players inklusive Steuerelementen.

## 5.4 Verknüpfung von <OBJECT> Tags und <EMBED> Tag

Durch die Verwendung dieser drei unterschiedlichen Methoden kann es zu Problemen bei der Darstellung der Seite mit den verschiedenen Browsern kommen. Das <OBJECT> Tag bietet die Möglichkeit, zwischen einleitendem und abschliessendem Tag weiteren HTML Code einzufügen, der nur dann interpretiert wird, wenn das <OBJECT> Tag nicht angezeigt werden kann. Hier kann also auch ein weiteres <OBJECT> Tag oder ein <EMBED> Tag stehen. Eine Ausnahme von dieser Regel ist der Internet Explorer, der auch <OBJECT> Tags in einem <OBJECT> Tag darzustellen versucht.

Damit der Player sowohl in Netscape Communicator, Mozilla bzw. Netscape 6, Internet Explorer, als auch in andere Browser eingebunden werden kann, sollte zuerst das <OBJECT> Tag mit den ActiveX Komponenten definiert werden. Darunter steht das <OBJECT> Tag mit dem MIME-Type, in welchem das <EMBED> Tag eingefügt ist. Der Internet Explorer interpretiert das erste <OBJECT> Tag richtig und versucht auch das Zweite darzustellen, findet aber keine ClassID und gibt daher nur Höhe und Breite des Tags als White Space wieder. Andere Browser ignorieren das erste <OBJECT> Tag und stellen entweder das <OBJECT> Tag mit dem MIME-Type oder das <EMBED> Tag dar:

Syntax:

```
<OBJECT CLASSID="CLSID:XXXXXX-XXXX-XXXXXX" WIDTH="Breite" HEIGHT="Höhe">
<PARAM NAME="Name" VALUE="Wert">
</OBJECT>
<OBJECT DATA="URL" TYPE="MIME-Type" WIDTH="Breite" HEIGHT="Höhe">
<EMBED SRC="URL" TYPE="MIME-Type" WIDTH="Breite" HEIGHT="Höhe">
</OBJECT>
```

Erläuterung:

- OBJECT CLASSID="CLSID:XXXXXX-XXXX-XXXXXX" - Das erste <OBJECT> Tag mit den ActiveX Komponenten. Es wird von Browsern, die kein ActiveX interpretieren, ignoriert.
- OBJECT TYPE="MIME-Type" - Das zweite <OBJECT> Tag mit dem MIME-Type.
  - EMBED SRC="URL" - Das <EMBED> Tag ist eingebettet im zweiten <OBJECT> Tag.

Ein Beispiel zur Einbettung des Real Players, das für alle gängigen Browser funktionieren sollte:

Syntax des Beispiels:

```
<OBJECT ID="ImageWindow" CLASSID="CLSID:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA"
HEIGHT="192" WIDTH="236">
<PARAM NAME="CONSOLE" VALUE="Clip1">
<PARAM NAME="SRC" VALUE="playerinfo/kleinesvideo.rpm">
<PARAM NAME="CONTROLS" VALUE="ImageWindow">
</OBJECT>
<OBJECT ID="ControlPanel" CLASSID="CLSID:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA"
HEIGHT="36" WIDTH="236">
<PARAM NAME="CONSOLE" VALUE="Clip1">
<PARAM NAME="CONTROLS" VALUE="ControlPanel">
</OBJECT>
<OBJECT ID="InfoPanel" CLASSID="CLSID:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA"
HEIGHT="55" WIDTH="236">
<PARAM NAME="CONSOLE" VALUE="Clip1">
<PARAM NAME="CONTROLS" VALUE="InfoPanel">
</OBJECT>
<OBJECT TYPE="audio/x-pn-realaudio-plugin" DATA="playerinfo/kleinesvideo.rpm"
CONTROLS="ImageWindow,ControlPanel,StatusBar" HEIGHT="228" WIDTH="236">
<EMBED TYPE="audio/x-pn-realaudio-plugin" SRC="playerinfo/kleinesvideo.rpm"
CONTROLS="ImageWindow,ControlPanel,StatusBar" HEIGHT="228" WIDTH="236">
</OBJECT>
```

Erläuterung:

- OBJECT ... CLASSID="CLSID:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA" - Das erste von drei <OBJECT> Tags mit den ActiveX Komponenten und der ClassID des Real Players und ist mit den anderen beiden über den Parameter "CONSOLE" verbunden. Das Tag wird von Browsern, die kein ActiveX interpretieren, ignoriert. Dieses Tag stellt das Video Fenster dar.
- OBJECT ... CLASSID="CLSID:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA" - Das zweite ActiveX <OBJECT> Tag. Es stellt das Panel mit den Kontrollen dar.
- OBJECT ... CLASSID="CLSID:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA" - Das letzte ActiveX <OBJECT> Tag, welches ein Informationsfenster einblendet.
- OBJECT TYPE="MIME-Type" - Das <OBJECT> Tag mit dem MIME-Type. Vom Internet Explorer wird dieses Tag als leere Fläche dargestellt.
  - EMBED SRC="URL" - Das <EMBED> Tag ist eingebettet im zweiten <OBJECT> Tag.

## 6. Metadateien ASX, RAM und SMIL

### 6.1 Einführung

Alle drei hier vorgestellten Player haben die Möglichkeit, mit sogenannten Metadateien, auch Playerinformationsdateien genannt, zu arbeiten. Diese Metadateien sind einfache Textdateien und beinhalten Informationen über die Quelle der Multimedia Daten, aber, abhängig von der Art der Metadateien, auch über die Darstellung im Player und Informationen über den Autor, die Präsentation etc. Sie wird anstelle des Streams dem Player übermittelt und kann lokal auf

dem Webserver hinterlegt werden. Die einfachste Art einer Metadatei ist also eine Weiterleitung an den Stream. Womit wir schon bei einem wichtigen Vorteil dieser Dateien sind: Anstelle dem Player einen absoluten Pfad zum Multimediaserver zu geben, wird ein relativer Pfad zur Metadatei gegeben. Dies ist besonders dann nützlich, wenn dieselbe Multimedia Datei häufig von verschiedenen HTML Seiten aus verwendet wird, so dass bei einer Änderung nicht sämtliche HTML Dateien geändert werden müssen sondern nur die Metadatei. Die Erstellung einer solchen Metadatei ist denkbar einfach: Nutzen Sie einen Editor, der Dateien in Plain Text (unformatierter Text) abspeichern kann, z.B. den Windows Editor oder Emacs, und schreiben Sie die entsprechenden Anweisungen nieder. Danach speichern Sie die Datei mit der entsprechenden Metadatei Endung ab und kopieren sie in das gewünschte Verzeichnis auf dem Webserver. Im Gegensatz zum eigentlichen Stream wird die Metadatei, die in der Regel nicht größer als 1 KB ist, in das Temp Verzeichnis des Browsers heruntergeladen, vom Browser mit dem entsprechendem Player assoziiert und dieser gestartet. Der Player interpretiert die Anweisungen der Metadatei und öffnet den angegebenen Stream, wobei auch der Verweis auf einen Live Stream möglich ist.

## 6.2 RAM und RPM

Wird genutzt von:	Real Media
Endungen:	*.ram, *.rpm

Real verwendet, neben SMIL, die Metadateien RAM (\*.ram) und RPM (\*.rpm). Beide sind ansich gleich, jedoch wird die RAM Metadatei bei einem Stand-Alone Player genutzt, während die RPM Metadatei bei eingebetteten Playern eingesetzt wird. Der eigentliche Aufbau ist jedoch derselbe und auch nicht weiter schwierig: Es wird direkt auf den gewünschten Stream verwiesen.

Umfangreiche Präsentationen, wie mit SMIL sind mit RAM/RPM nicht möglich.

Syntax:

```
rtsp://multimediaserver.domain.de/mountpoint/pfad/datei.rm
```

Man hat auch die Möglichkeit mehrere Clips hintereinander abzuspielen, indem man einfach mehrere Adressen angibt.

Syntax:

```
rtsp://multimediaserver.domain.de/mountpoint/pfad/datei1.rm
rtsp://multimediaserver.domain.de/mountpoint/pfad/datei2.rm
rtsp://multimediaserver.domain.de/mountpoint/pfad/datei3.rm
```

Ein Beispiel mit RAM:

[playerinfo/rambeispiel.ram](#)

Und ein Beispiel mit RPM: (playerinfo/rpmbeispiel.rpm)

**„Auf der Webseite kann das Beispiel nicht angezeigt werden“**

Der Aufbau beider Dateien ist gleich, nur die Dateieindung ist anders.

Syntax beider Beispiele:

```
rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/kleinesvideo.rm
rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/piggy56.rm
```

**6.3 ASX**

Wird genutzt von:	Microsoft Media
Endungen:	*.asx (*.wvx, *.wax)

Während die RAM/RPM Metadatei fast nur aus reinen Verweisen auf Streams besteht, nutzt die ASX Metadatei Extensible Markup Language (XML) Tags. XML ist HTML sehr ähnlich, Sie müssen jedoch, neben den unterschiedlichen Tags, noch auf folgende Punkte achten:

Syntax Beispiel ASX I:

```
<ASX VERSION = "3.0"> <!-- Kommentar -->
<ENTRY>
  <REF HREF = "mms://ZAIK-WM.RRZ.UNI-
KOELN.DE/rrzk/streaming/dokumentation/kleinesvideo.wmv"/>
</ENTRY>
</ASX>
```

- Alle XML Dateien beginnen und enden mit einem Tag zwischen denen alle anderen Tags stehen. (<ASX> ... </ASX>)
- Alle XML Tags und Attribute müssen kleingeschrieben werden. (Achtung: Die Beispiele werden der Übersicht halber GROSS geschrieben.)
- Ein Tag, das kein korrespondierendes End Tag besitzt, muss mit einem Forward Slash ("/") enden. (<REF/>)
- Attributwerte stehen in Anführungszeichen. (HREF="mms://.../kleinesvideo.wmv")
- Kommentare werden, wie bei HTML, mit <!-- Kommentar --> dargestellt.

Jede ASX Metadatei beginnt mit <ASX VERSION = "3.0"> und schließt </ASX>. Dazwischen stehen <ENTRY> Tags, die die einzelnen Clips, die abgespielt werden sollen beinhalten. Jedes <ENTRY> Tag sollte zumindest ein <REF/> Tag besitzen, das auf den Stream verweist, kann aber auch weitere Informationen über Autor (<AUTHOR>), Titel (<TITLE>), Copyright (<COPYRIGHT>), Beschreibungen (<ABSTRACT>) usw. enthalten.

Ein ASX Beispiel:

[playerinfo/beispiel2.asx](#)

Syntax Beispiel ASX II:

```

<ASX VERSION = "3.0">
<TITLE>Ein Beispiel für eine ASX Datei</TITLE>
<AUTHOR>Sebastian Schetter</AUTHOR>
<COPYRIGHT>(c) 2002 ZAIK/RRZK</COPYRIGHT>
<ENTRY>
<TITLE>Der erste ENTRY Eintrag </TITLE>
<ABSTRACT>Infos über diesen Entry Tag</ABSTRACT>
<AUTHOR>Sebastian Schetter</AUTHOR>
<COPYRIGHT>(c) 2002 ZAIK/RRZK</COPYRIGHT>
<REF HREF = "mms://ZAIK-WM.RRZ.UNI-
KOELN.DE/rrzk/streaming/dokumentation/kleinesvideo.wmv"/>
</ENTRY>
</ASX>

```

Wenn mehrere <ENTRY> Tags angegeben werden, so werden diese Titel der Reihe nach abgespielt. Auch ist es möglich mehrere <REF/> Tags in einem <ENTRY> Tag anzugeben. In diesem Falle prüft der Windows Media Player die einzelnen Links und spielt **nur** das erste funktionierende File ab.

Ein Beispiel:

[playerinfo/beispiel3.asx](#)

Syntax Beispiel ASX III:

```

<ASX VERSION = "3.0">
<TITLE>Beispiel III für eine ASX Datei</TITLE>
<AUTHOR>Sebastian Schetter</AUTHOR>
<COPYRIGHT>(c) 2002 ZAIK/RRZK</COPYRIGHT>
<ENTRY>
<TITLE>Der erste ENTRY Eintrag </TITLE>
<ABSTRACT>Dieser ENTRY Tag beinhaltet nur einen REF Tag.</ABSTRACT>
<AUTHOR>Sebastian Schetter</AUTHOR>
<COPYRIGHT>(c) 2002 ZAIK/RRZK</COPYRIGHT>
<REF HREF = "mms://ZAIK-WM.RRZ.UNI-
KOELN.DE/rrzk/streaming/dokumentation/kleinesvideo.wmv"/>
</ENTRY>
<ENTRY>
<TITLE>Der zweite ENTRY Eintrag </TITLE>
<ABSTRACT>Dieser ENTRY Tag beinhaltet drei REF Tags, von denen aber nur zwei,
der mittlere und der letzte, funktionieren.</ABSTRACT>
<AUTHOR>Sebastian Schetter</AUTHOR>
<COPYRIGHT>(c) 2002 ZAIK/RRZK</COPYRIGHT>
<REF HREF = "mms://ZAIK-WM.RRZ.UNI-
KOELN.DE/rrzk/streaming/dokumentation/gehtjagarnich.wmv"/>
<REF HREF = "mms://ZAIK-WM.RRZ.UNI-KOELN.DE/rrzk/streaming/dokumentation/piggy56.wma"/>
<REF HREF = "mms://ZAIK-WM.RRZ.UNI-
KOELN.DE/rrzk/streaming/dokumentation/kleinesvideo.wmv"/>
</ENTRY>
</ASX>

```

## 6.4 SMIL

Wird genutzt von:	Real Player, Quicktime (& Internet Explorer)
Endungen:	*.smil, *.smi

SMIL (wird englisch 'smile' ausgesprochen) steht für Synchronized Multimedia Integration Language. Es bietet die Möglichkeit Multimedia Objekte in eine Seite zu integrieren, zu steuern und dient der Synchronisation, Positionierung und Präsentation dieser Objekte. SMIL ist eine eigenständige Sprache, ähnlich wie HTML jedoch nicht zur Darstellung von Text, Tabellen usw. gedacht, sondern dient der Präsentation von Multimedialen Inhalten. SMIL wird, wie andere Metadateien, vom Player interpretiert und die Darstellung der SMIL Anweisungen erfolgen im Player, nicht im Browser. (Ausnahme: Der Internet Explorer ab Version 6 interpretiert SMIL, jedoch nicht der Windows Media Player.)

Hier soll nur eine grundlegende Einführung über die Möglichkeiten mit SMIL gegeben werden unter Nutzung der Möglichkeiten des Real Players. Quicktime nutzt zwar auch SMIL, aber mit anderen Attributen, deren Beschreibung jedoch den Rahmen dieses Dokumentes sprengen würde.

Die vom W3C entwickelte Sprache ist ein XML Derivat und nutzt daher ähnliche Regeln, wie sie auch für ASX gelten:

Syntax Beispiel SMIL I:

```
<SMIL> <!-- Kommentar -->
<BODY>
  <VIDEO SRC = "rtsp://zaik-rm.rz.uni-koeln.de/rrzk/streaming/dokumentation/kleinesvideo.rm"/>
</BODY>
</SMIL>
```

- Alle XML Dateien beginnen und enden mit einem Tag zwischen denen alle anderen Tags stehen. (<SMIL> ... </SMIL>)
- Alle SMIL Dateien **müssen** ein <BODY> Tag besitzen, ein <HEAD> Tag ist optional.
- Alle XML Tags und Attribute müssen kleingeschrieben werden. (Achtung: Die Beispiele werden der Übersicht halber GROSS geschrieben.)
- Ein Tag, das kein korrespondierendes End Tag besitzt, muss mit einem Forward Slash ("/") enden. (<VIDEO/>)
- Attributwerte stehen in Anführungszeichen. (SRC="rtsp://.../kleinesvideo.rm")
- Kommentare werden, wie bei HTML, mit <!-- Kommentar --> dargestellt.

Im Header können allgemeine Informationen zur Präsentation angegeben werden, wie etwa Autor, Copyright, Titel, eine allgemeine Beschreibung usw. Im Gegensatz zu ASX, das nur das <REF/> Tag kennt, nutzt SMIL verschiedene Tags, um einen Clip abzuspielen, abhängig vom Typ des Clips, also Video, Audio etc:

Tag	Beschreibung	Beispiel
REF	für alle Arten von Clips	RealVideo (*.rm), JEPEG (*.jpg), RealPix (*.rp), etc.
VIDEO	für Videos	RealVideo (*.rm)
AUDIO	für Audio	RealAudio (*.rm)
IMG	für Bilder	JPEG (*.jpg) oder GIF (*.gif)
TEXT	für Text	Textdateien (*.txt)

TEXTSTREAM	für streaming Text	RealText (*.rt)
ANIMATION	für Flash Animationen	Shockwave Flash (*.swf)

Ein Beispiel für Angaben von <META> Informationen und verschiedenen Clip Typen:  
[playerinfo/beispiel2.smil](#)

Syntax Beispiel SMIL II:

```
<SMIL>
<HEAD>
  <META NAME="author" CONTENT="Sebastian Schetter"/>
  <META NAME="titel" CONTENT="Clips parallel und sequentiell abspielen mit SMIL."/>
  <META NAME="copyright" CONTENT="(c)2002 ZAIK/RRZK"/>
  <META NAME="abstract" CONTENT="Eine kurze Einfuehrung in SMIL (gesprochen:smile)"/>
</HEAD>
<BODY>
  <SEQ>
    <VIDEO SRC="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/kleinesvideo.rm" />
    <TEXTSTREAM ID="ticker"
      SRC="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/ticker2.rt" />
  </SEQ>
</BODY>
</SMIL>
```

Im obigen Beispiel ist schon das <SEQ> Tag genutzt worden. <SEQ> steht für Sequenz und spielt die in ihm enthaltenen Clips der Reihe nach durch. Das Gegenstück ist das <PAR> Tag, welches Clips parallel abspielt. Diese beiden Tags können miteinander kombiniert und ineinander verschachtelt werden.

Ein Beispiel für die Verschachtelung von <SEQ> und <PAR> Tags: Ein <SEQ> Tag beinhaltet ein weiteres <SEQ> Tag als auch ein <PAR> Tag, mit je zwei Clips. Diese beiden Tags werden, weil sie in einem <SEQ> Tag liegen, nacheinander aufgerufen: Erst das <SEQ> Tag, welches nacheinander seine Clips abspielt und dann das <PAR> Tag, welches seine beiden Clips parallel abspielt.

[playerinfo/beispiel3.smil](#)

Syntax Beispiel SMIL III:

```
<SMIL>
<HEAD>
  <META NAME="author" CONTENT="Sebastian Schetter"/>
  <META NAME="titel" CONTENT="Clips parallel und sequentiell abspielen mit SMIL."/>
  <META NAME="copyright" CONTENT="(c)2002 ZAIK/RRZK"/>
  <META NAME="abstract" CONTENT="Eine kurze Einfuehrung in SMIL (gesprochen:smile)"/>
</HEAD>
<BODY>
  <SEQ>
    <SEQ>
      <VIDEO SRC="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/kleinesvideo.rm"/>
      <AUDIO SRC="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/piggy56.rm"/>
    </SEQ>
    <PAR ENDSYNC="id(ticker)" DUR="50s">
      <VIDEO SRC="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/video_lang_56k.rm" />
    </PAR>
  </SEQ>
</BODY>
</SMIL>
```

```

<TEXTSTREAM ID="ticker"
  SRC="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/dokumentation/ticker2.rt" />
</PAR>
</SEQ>
</BODY>
</SMIL>

```

Dies ist legendlich eine sehr kurze Einführung in SMIL, um eine ungefähre Vorstellung der weiteren Möglichkeiten von SMIL zu bekommen können Sie folgendes Beispiel betrachten: (Achtung! Nur geeignet ab einer Bandbreite über 250kbit/s!)

[playerinfo/beispiel4.smil](http://playerinfo/beispiel4.smil)

SYNTAX Beispiel IV:

```

<SMIL>
<HEAD>
<META NAME="author" CONTENT="Sebastian Schetter"/>
<META NAME="titel" CONTENT="Clips parallel und sequentiell abspielen mit SMIL."/>
<META NAME="copyright" CONTENT="(c)2002 ZAIK/RRZK"/>
<META NAME="abstract" CONTENT="Eine kurze Einfuehrung in SMIL (gesprochen:smile)"/>
<META NAME="base" CONTENT="rtsp://zaik-rm.rrz.uni-koeln.de/rrzk/streaming/" />
<LAYOUT>
<ROOT-LAYOUT WIDTH="400" HEIGHT="375" BACKGROUND-COLOR="red" />
<REGION ID="background" TOP="10" LEFT="10" WIDTH="380" HEIGHT="355"
  BACKGROUND-COLOR="silver" FIT="fill" Z-INDEX="0"/>
<REGION ID="video" TOP="35" LEFT="35" WIDTH="320" HEIGHT="240"
  BACKGROUND-COLOR="transparent" Z-INDEX="1"/>
<REGION ID="ticker" TOP="310" LEFT="35" WIDTH="330" HEIGHT="30"
  BACKGROUND-COLOR="#FFFFFF" Z-INDEX="1"/>
</LAYOUT>
</HEAD>
<BODY>
<PAR ENDSYNC="id(ticker)" DUR="60s">
  <IMG REGION="background"
    SRC="http://www.uni-koeln.de/rrzk/kurse/unterlagen/audio_video/streaming/img/zaik-logo-medium.gif"
    FILL="freeze" />
  <SEQ>
    <VIDEO REGION="video" SRC="Hallo!.rm" CLIP-END="8s" FILL="freeze"/>
    <AUDIO SRC="PiggyLAN.rm" />
  </SEQ>
  <TEXTSTREM REGION="ticker" SRC="ticker.rt" FILL="freeze" />
</PAR>
</BODY>
</SMIL>

```

## 7. Literatur

Dieses Kapitel führt Werke und Links auf, die zur Erstellung dieses Dokumentes hilfreich waren oder aber als Lektüre zu diesen Themen empfohlen werden.

### 7.1 Streamingplattformen

#### 7.1.1 Herstellerseiten

Apple <http://www.apple.com>

Microsoft <http://www.microsoft.de>

Real Networks <http://www.real.com>

Helix Community <https://helixcommunity.org/>

#### 7.1.2 Player

Apple Quicktime <http://www.apple.com/quicktime>

Microsoft Mediaplayer 9 <http://www.microsoft.com/windows/windowsmedia/de/players.asp>

Real Networks  
RealPlayer (free) <http://www.real.com/freeplayer/?rppr=hc.org>

Helix Player <https://player.helixcommunity.org/>

Helix DNA Media Client <https://helix-client.helixcommunity.org/>

#### 7.1.3 Encoder

Microsoft  
Encoder <http://www.microsoft.com/windows/windowsmedia/de/9series/encoder/default.asp>  
9

Real  
Networks <http://www.realnetworks.com/products/producer/>  
Producer

Helix  
DNA <https://helix-producer.helixcommunity.org/>  
Producer

#### 7.1.4 Server

Apple Media [http://www.apple.com/server/media\\_streaming.html](http://www.apple.com/server/media_streaming.html)

## Streaming Server

Apple Streaming  
Server Darwin  
(Open Source  
Projekt)

<http://developer.apple.com/opensource/server/streaming/>

Microsoft Media  
Server

<http://www.microsoft.com/windows/windowsmedia/DE/9series/server.asp>

Real Networks  
Helix Server

[http://www.realnetworks.com/products/media\\_delivery.html](http://www.realnetworks.com/products/media_delivery.html)

Helix DNA  
Server

<https://helix-server.helixcommunity.org/>

### *7.1.5 Einbetten des Players*

Apple [http://developer.apple.com/documentation/QuickTime/Conceptual/QTScripting\\_HTML/QTScripting\\_HTML\\_Document/chapter\\_1000\\_section\\_5.html](http://developer.apple.com/documentation/QuickTime/Conceptual/QTScripting_HTML/QTScripting_HTML_Document/chapter_1000_section_5.html)

Windows

<http://www.microsoft.com/windows/windowsmedia/howto/articles/adsolutions2.aspx>

Real Embedded RealPlayer Extended Functionality Guide

## **7.2 Tutorials**

Ein hervorragendes Online Tutorial und Nachschlagewerk: [SelfHTML](#)