Die bunte Welt
und die dunkle Seite
des

Java Media Frameworks (JMF)

oliver.boehm@agentes.de

Agenda

- Die bunte Welt von JMF
- Architektur
- The Dark Side
- The Bright Side
- Dokumentation

TV is chewing gum for the eyes.
-- Frank Lloyd Wright

Was ist JMF?

agentes

- JMF = Java Media Framework
- Umgang mit Video- und Audio-Daten
- API für Audio-/Video-Player
- Streaming-API

Installation

- http://java.sun.com/products/java-media/jmf/index.js
  - akt. Version: 2.1.1e (v. 2003)
- Framework: lib
  - JARs und Nativ Libs
- Demos: bin
  - JMStudio, JMF-Registry
- Plugin für IE / Mozilla
  - Linux: Handarbeit notwendig

Medienformate

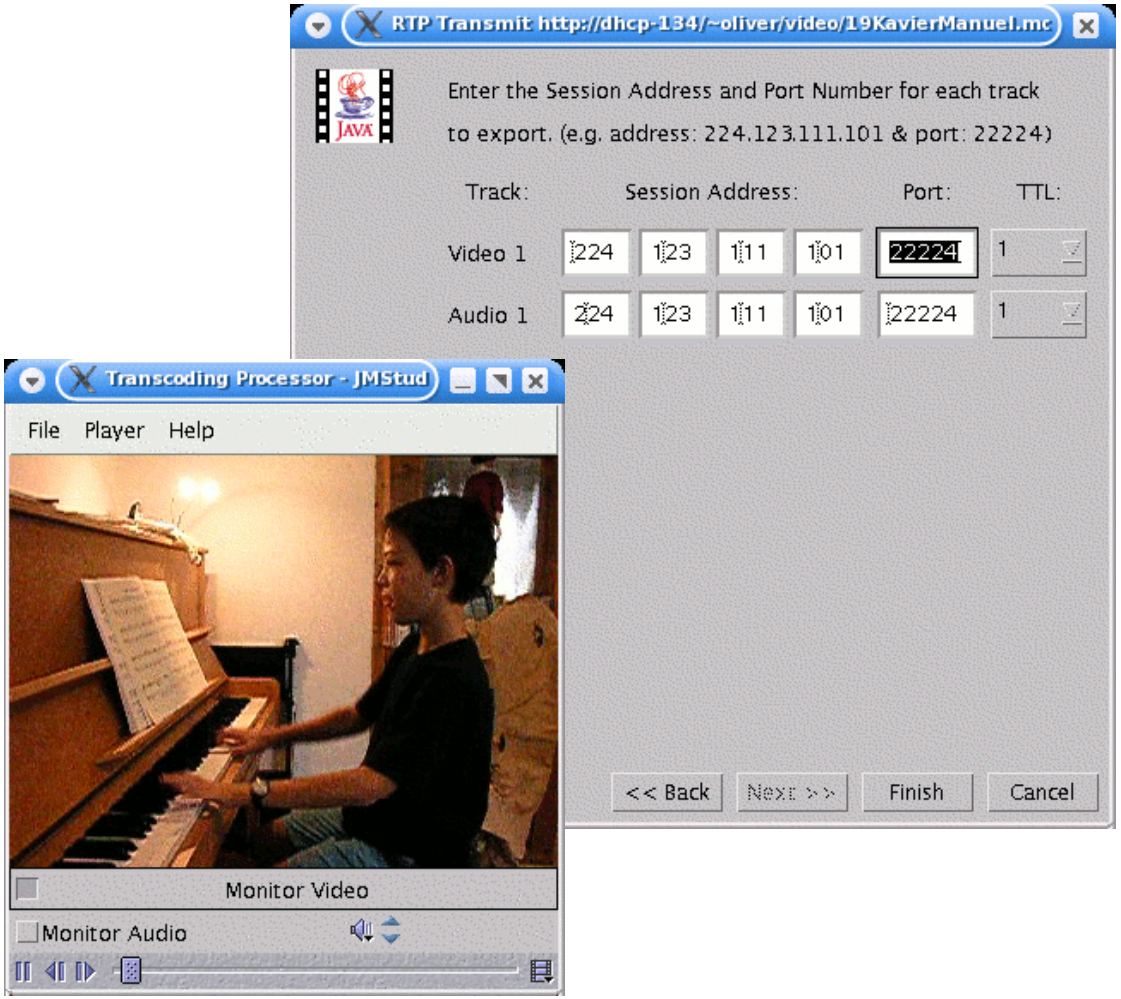
- Audio:
  - G723, GSM, WAV, ULAW
  - MP3: ab JMF 2.1.1b wieder entfernt (aus Lizenz-Gründen)
- Video:
  - CINEPAK, H263, MJPG, MPEG, MOV (QuickTime)
  - streamingfähig:
    - QuickTime, MPEG, WMF
  - nicht streamingfähig
    - AVI

JMStudio

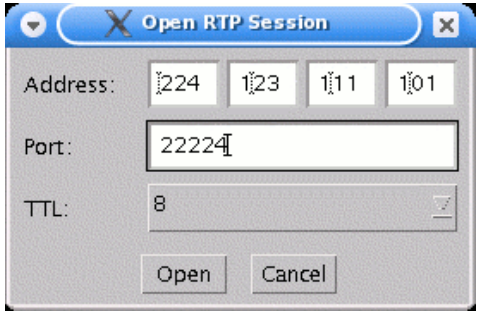
agentes

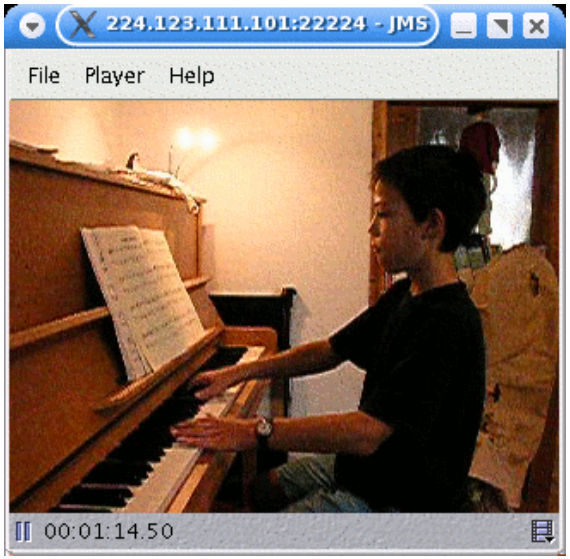
- Audio-/Video-Player
  - Media lokal / über URL adressierbar
- auch als Streamer einsetzbar
- Achtung: läuft nicht unter JDK 1.5
  - NullPointerException
  - ebenfalls anfällig: Sound SPI (MP3SPI v. javazoom)
- Doku-Ersatz bzw. -Ergänzung

JMStudio (Sender)



JMStudio (Empfänger)





Beispiel-Programme

agentes

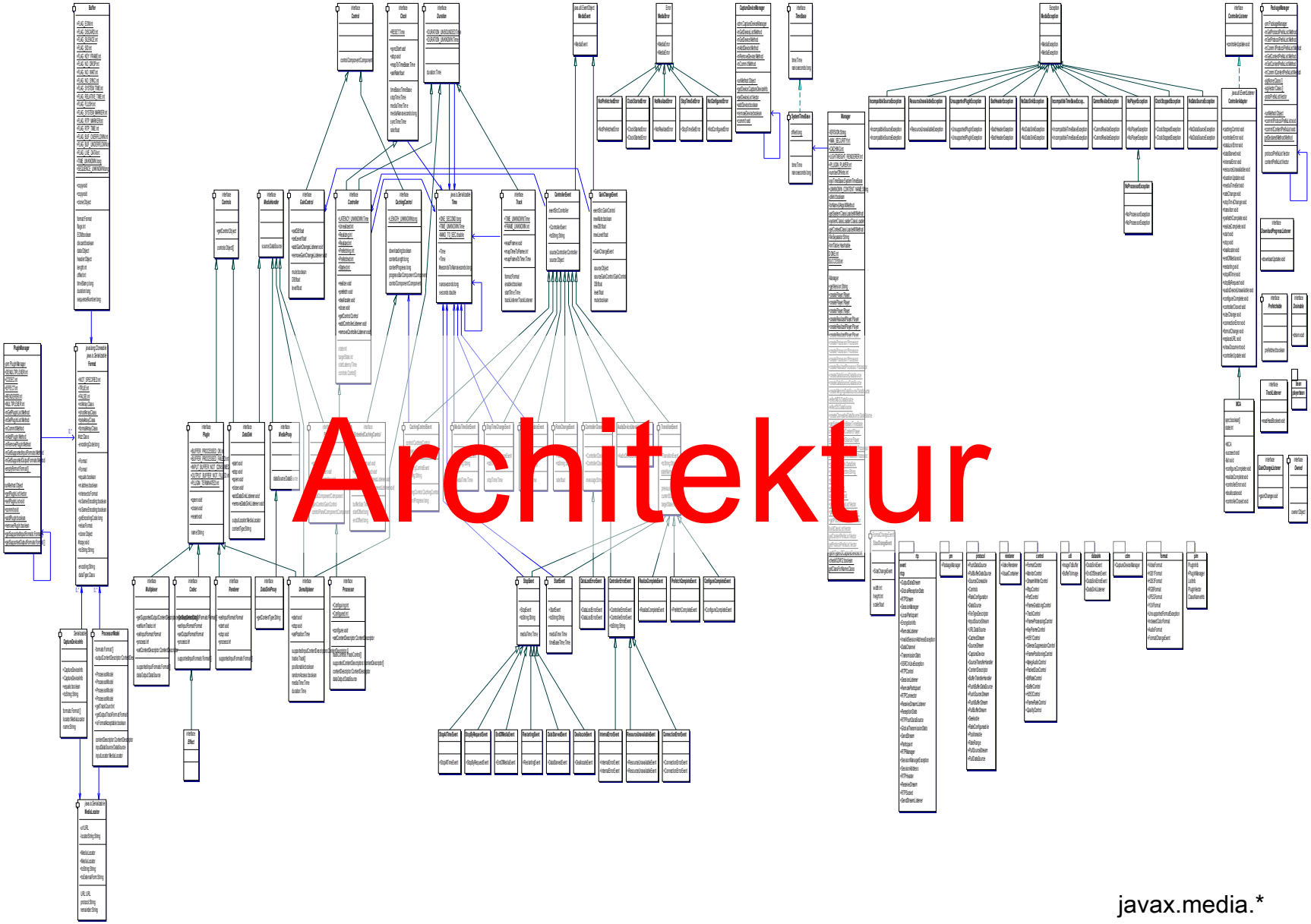
- s. http://java.sun.com/products/java-media/jmf/
- aber:
  - teilweise Mängel in der Programmierung
  - teilweise veraltet
- SessionManager durch RTPManager abgelöst

PlayerBean

- Beispiel für einen einfachen Video-Player:

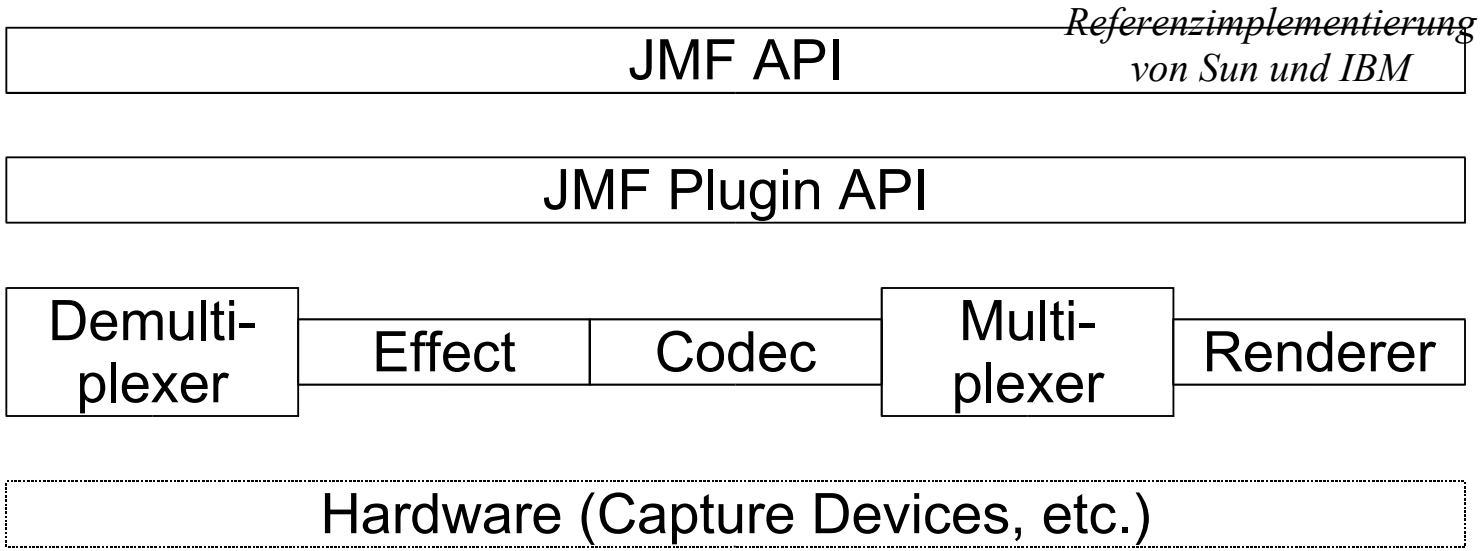
```
MediaPlayer1 = new javax.media.bean.playerbean.MediaPlayer();
MediaPlayer1.setMediaLocation("file://jvideo/media/Sample1.mov");
MediaPlayer1.start();
```

- aus
  http://java.sun.com/products/java-media/jmf/2.1.1/playerbean.html



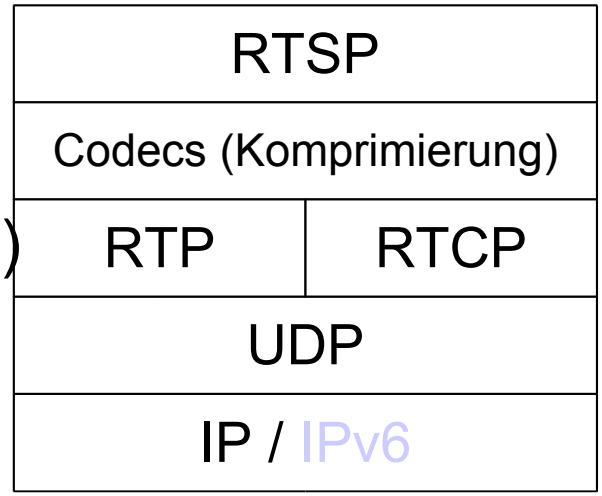
javax.media.*

Architektur



Netzwerk / RTP

- RTSP (Realtime Streaming Protokoll
  - RFC 2326
  - Verbindungs-Auf/Abbau, Steuerung
- RTP (Realtime Transfer Protokoll)
  - RFC 1889 + 1890
  - Daten- (RTP) und Steuerungs-Pakete (RTCP)
- Session-Arten
  - Unicast
  - Multicast (224-239.x.x.x)
  - Broadcast (x.x.x.255)

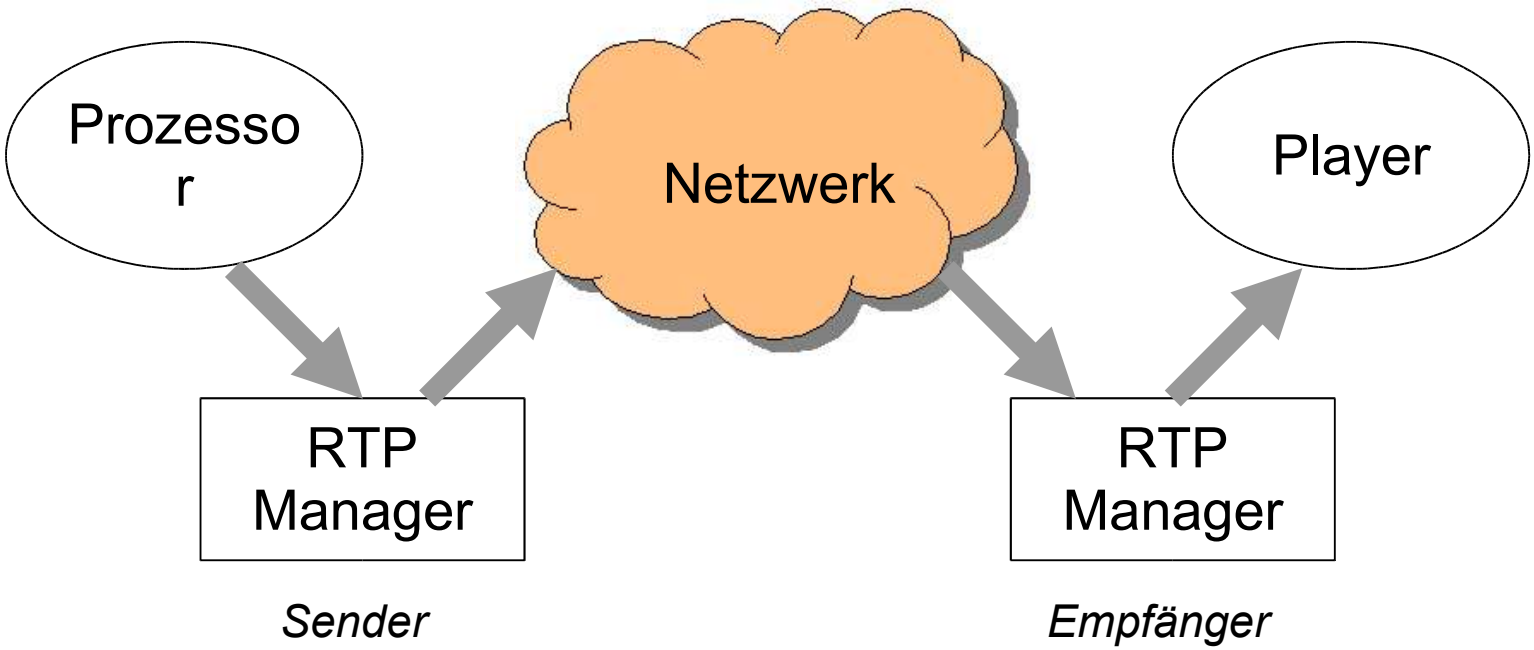


Problem: QoS / Bandbreite
-> „send and pray“

Multicast

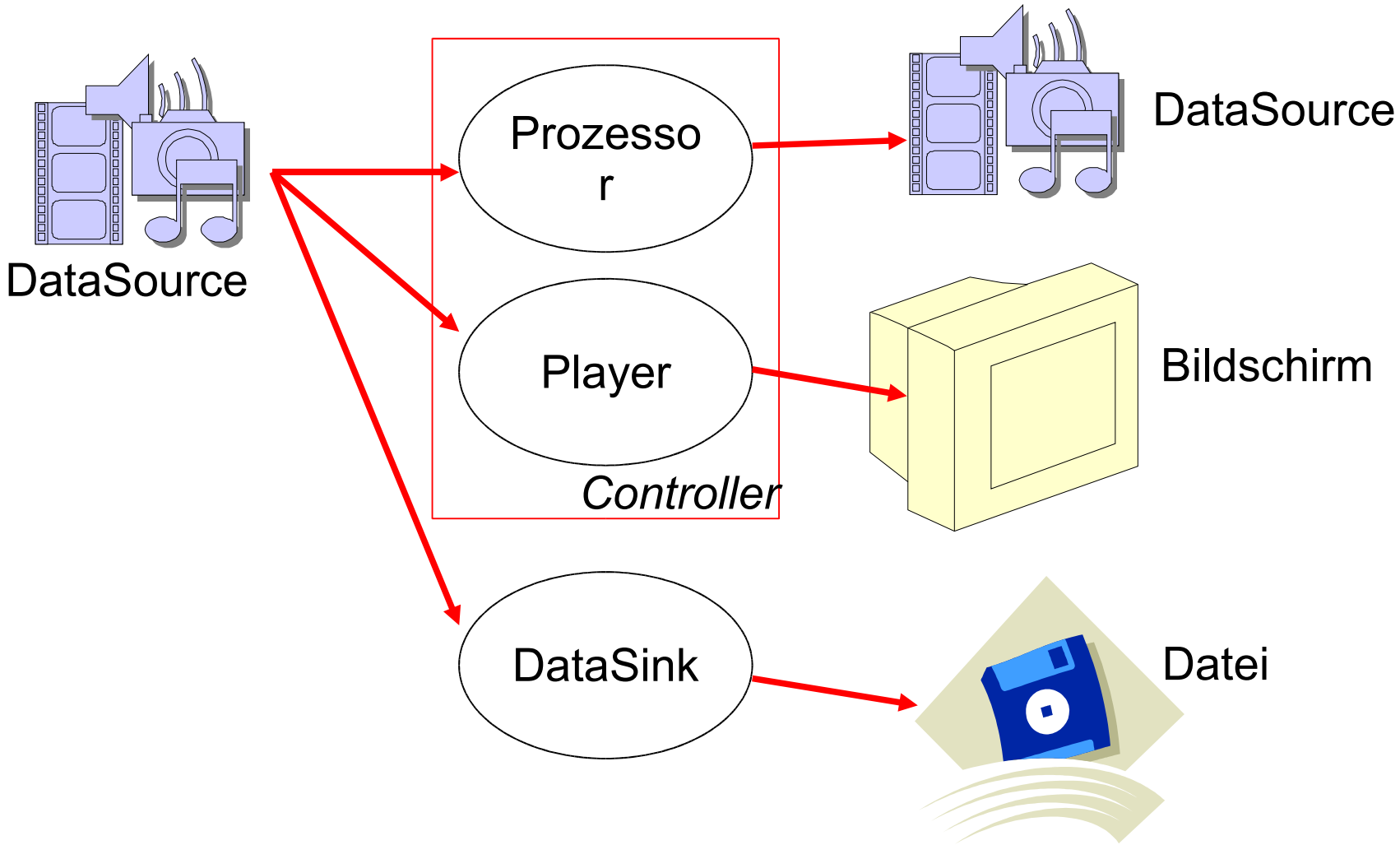
- Adressbereich: 224-239.x.x.x, RFC 1112
- Vorteil:
  - Daten werden nur 1x übertragen (Datenvolumen)
- Vorraussetzungen:
  - multicastfähiges Betriebssystem
  - Multicast-fähige Router
  - Multicast-Adresse muss bekannt sein

RTP & JMF

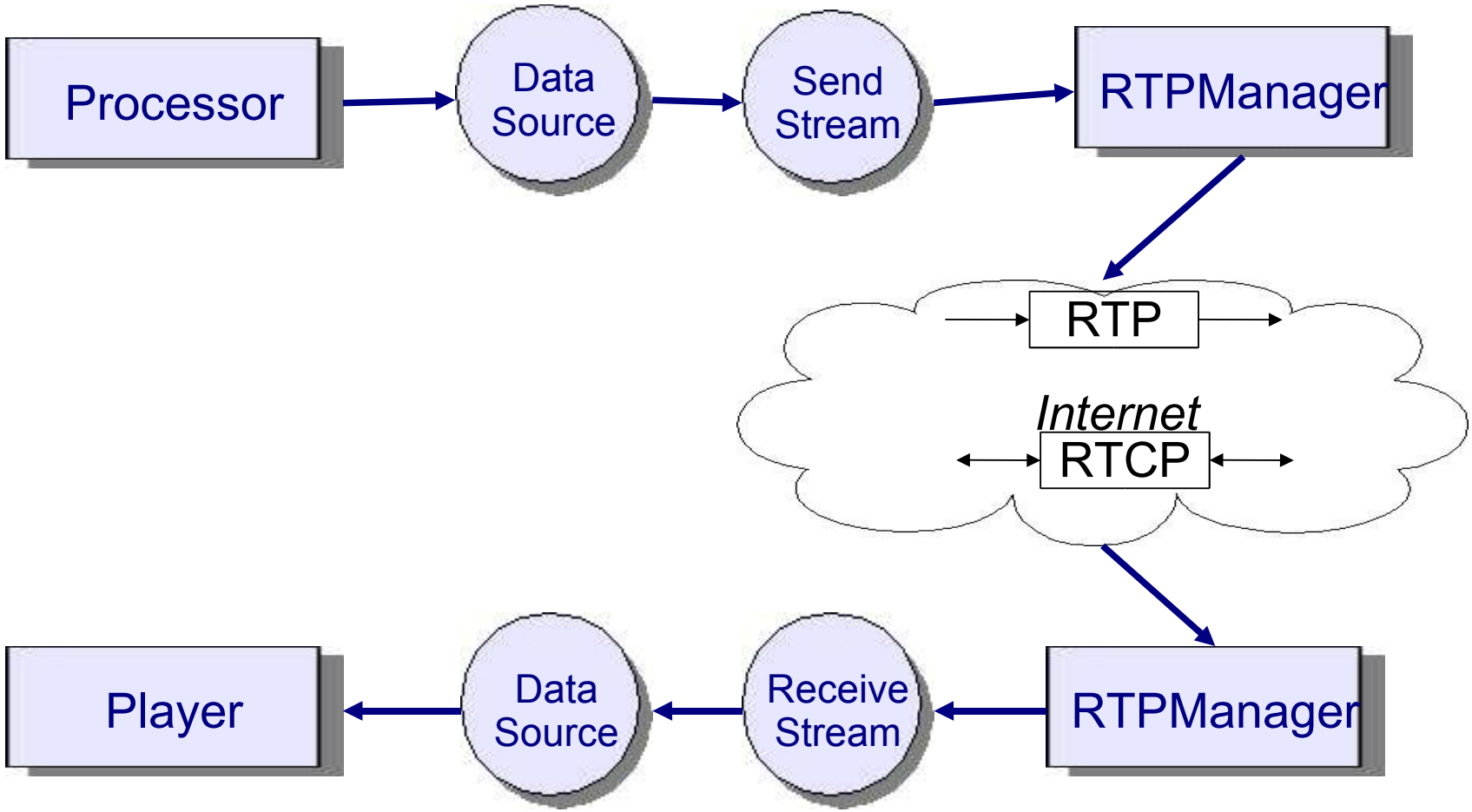


RTPManager ersetzt SessionManager (alt)

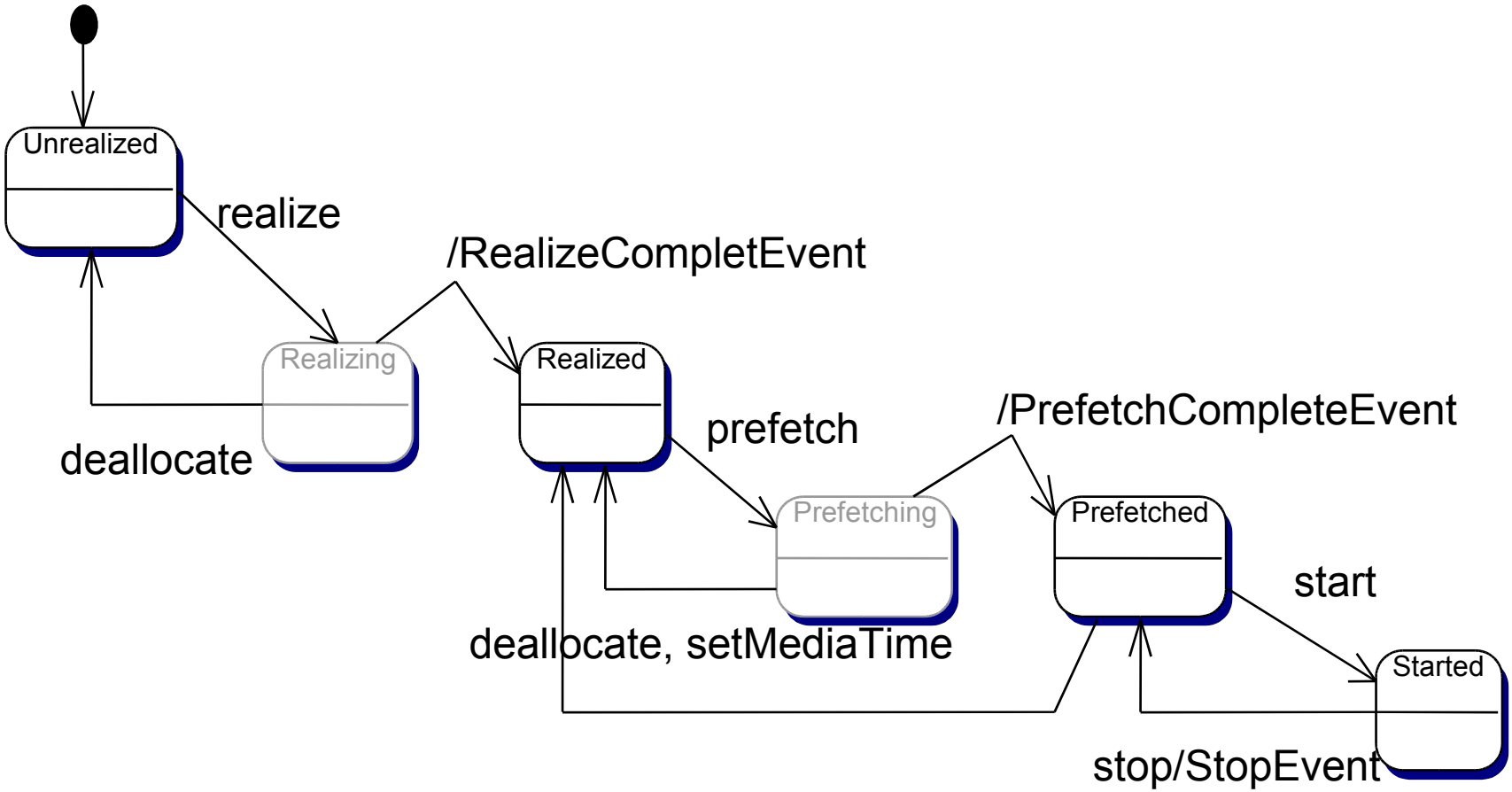
Arten von Controllern



RTP-Streaming in JMF



Zustände (Player)



Zeit & Synchronisation

agentes

- Zeitmodell in JMF
  - Clock
  - SystemTimeBase
  - TimeBase
  - Time
  - Duration
- Aktuelle Zeit
  - getMediaTime()
  - setMediaTime()

- Intramedia-Synchronisation
  1. TimeBase Synchronisation
  2. Kontrolle eines Players durch einen anderen
  3. direkte Synchronisation


the dark side

Die dunkle Seite von JMF

agentes

- problembehaftete Beispiele:
  - laufen nicht immer
  - verwenden undokumentierte Methoden
    - z.B. dataSource.getControl("javax.media.rtp.RTPControl")
  - referenzierte Klassen fehlen, z.B. VFWCapture
- Sourcen unter SCSL, aber unvollständig
  - z.B. com.sun.media.rtp.RTPSessionMgr fehlt
  - aber in JMF.jar vorhanden
  - ➔ *Sourcen stimmen nicht mit JMF.jar überein*

Die dunkle Seite (2)

- NullPointExceptions are the frameworks best friends!!!
- unnötige Casts
  - Object als Rückgabewert
- dyn. Laden von Klassen
  - Übergabe als String!
- Threads “verhungern” manchmal
- Mängel in der Programmierung
  - Exceptions werden als Rückgabewert verschleiert (C-Stil!)
  - IOExceptions können verschiedenes bedeuten:
    - File konnte nicht geladen werden
    - JMF konnte keine Klasse nachladen
    - Framework hatte gerade keine Lust

Erfahrungswerte

agentes

- Controller nie im falschen Status aufrufen
  - keine Exception, nur undefiniertes Verhalten
- JMF programmieren ist wie um Hand anhalten:
  - Blumen nicht vergessen (für Schwiegermama)
  - nichts Falsches im falschen Moment sagen
  - keine Reaktion kann bedeuten:
    - alles ok, es fehlt nur noch der Gehaltsnachweis
    - Ablehnung
  - am Ende bitte das Geschirr aufräumen

Dokumentation?

agentes

You must work hard what the only that existis: JMF guide, sun code examples and jmf studio source code.

After a lot of hours you will feel better!

Antwort auf eine Anfrage im JMF-Forum bzgl. Dokumentation

API-Guide

agentes

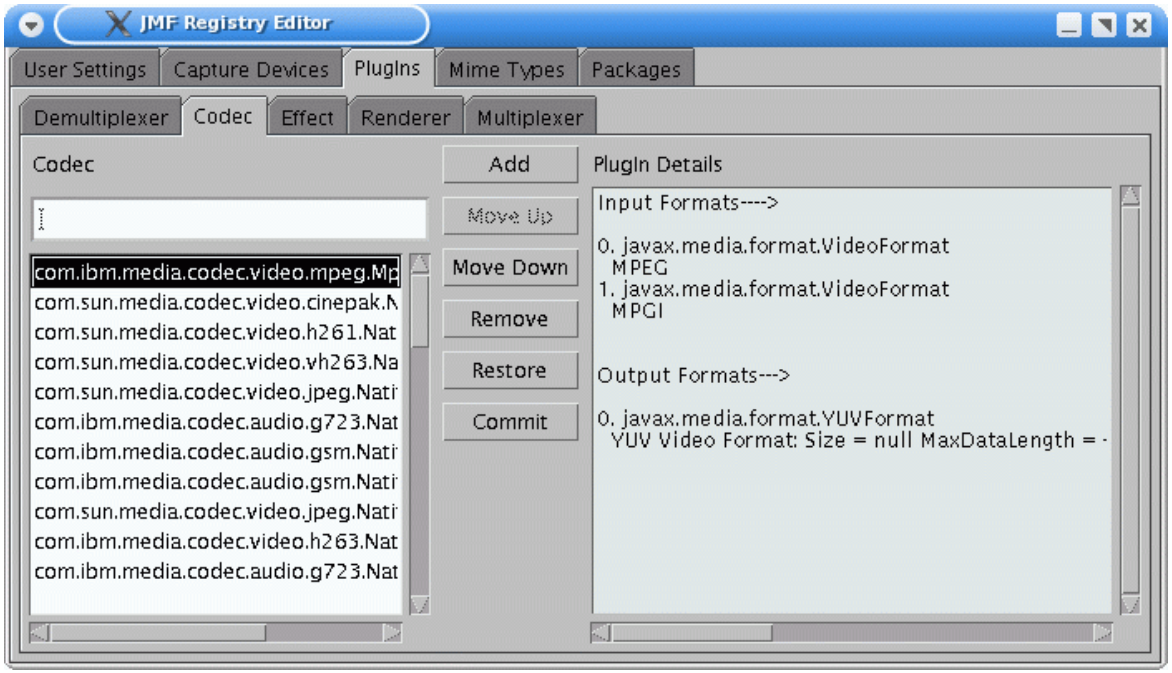
- als erster Einstieg geeignet, aber
- veraltet
  - RTP-Dokumentation
  - SessionManager durch RTPManager abgelöst
- unvollständig
  - MimeManager fehlt
  - RGBScaler

API-Doc

agentes

- Offline-Version
- besser: Online-Version (neuerer Stand)
- noch besser:
  - Sourcen herunterladen
  - Javadoc selber generieren (aus „Medienverarbeitung in Java“)
  - zusätzlich: Klassendiagramm (z.B. über Together)

JMF Registry



- für alles, was nicht in der Doku steht:
  - MIME-Types
  - Codecs
  - ...

Testen

agentes

- „write once ~~run~~ test anywhere“
- unterschiedliches Verhalten unter Windows / Linux
  - Beispiele (.au/.wav) laufen unter Linux, aber nicht unter Windows (und umgekehrt)
  - kein mitgeliefertes MP3 unter Linux
  - Broadcast geht nicht
- automatische JUnit-Tests


the bright side

Pro's:

- leistungsfähige Architektur
  - an JavaSound angelehnt
- konkurrenzlos
  - einziges plattform-unabhängige Media-Framework
  - MMAPI (Mobile Media API) stark an JMF angelehnt
- Erweiterungen / Plugins für JMF erhältlich

Voraussetzungen für Einsatz

agentes

- leistungsfähiges Logging (z.B. log4j)
  - normales Debuggen ist nicht möglich
- Thread-Debugger
- sauberes Exception-Handling
- automatische Tests
  - write once test anywhere
- Netzwerk-Monitor
  - tcpdump, ethereal, ...

Hilfen zur Einarbeitung

agentes

- viele Beispiele
- API-Doc zum Nachschlagen
- JMStudio debuggen
- Log-Aspekte / AspectJ

JMF-Erweiterungen

agentes

- Fobs4JMF
  - FFMpeg-Wrapper
  - FFMpeg = Library zum Aufnahmen, Verarbeiten und Streamen von Media-Daten (http://ffmpeg.sf.net)
  - http://fobs.sf.net
- JLayer
  - MP3 Library
  - http://www.javazoom.net/projects.html

JMF-Anwendungen

agentes

- NeburStream
  - Java streaming server over JMF (leider auf Spanisch)
  - http://sourceforge.net/projects/neburstream/
  - baut auf http://vod.sf.net auf
- VASTE (Video And Slideshow Teaching Environment)
  - http://www.wire.tu-bs.de/OLDWEB/multimedia/
- X-Smiles (XML-Browser + SMIL-Player)
  - http://www.xsmiles.org/
- TimCam (WebCam-Anwendung, http://timcam.sf.net)
- einige Diplomarbeiten

SMIL

- Synchronized Multimedia Integration Language
  - SMIL (sprich: smile :-)
- W3C-Standard
- Beschreibungssprache für Media-Daten/-Präsentationen
  - z.B. für Playlist
  - für Präsentationen

Streaming-Lösungen (RTP, RTSP)

agentes

- kommerziell (C++)
  - Darwin Streaming Server
    - freie Port d. Apple QuickTime Servers
  - Reals HELIX Server
    - open multi-format platform for digital media creation
- OpenSource (C, C++)
  - VideoLAN
    - http://www.videolan.org
    - RTSP?
  - LIVE.COM Streaming Media
    - http://live.sf.net

Doku

- Medienverarbeitung in Java
  - s. http://www.dpunkt.de/buch/3-89864-184-8.html
  - Grundlage für Vorlesung an der TU Wien
  - Beispiele unvollständig
- JMF API Guide
  - http://java.sun.com/products/java-media/jmf/2.1.1/specdownload.html
  - teilweise veraltet
- API-Dok
  - online: http://java.sun.com/products/java-media/jmf/2.1.1/apidocs/index.html
  - aktueller als Offline-Version

Links

agentes

- MCRL > MCRL Media > Media Projects > Java Media Framework
  - http://pirobase.pironet-ndh.com/servlet/PB/menu/1330640/index.html
- JMF Home / FAQ
  - http://java.sun.com/products/java-media/jmf/index.jsp
  - http://java.sun.com/products/java-media/jmf/reference/faqs/index.html
- SMIL
  - http://www.informatik.fernuni-hagen.de/import/pi3/peter/smil.htm

Vielen Dank

agentes AG

Oliver Böhm

Oliver.boehm@agentes.de

Räpplenstaße 17

70191 Stuttgart