

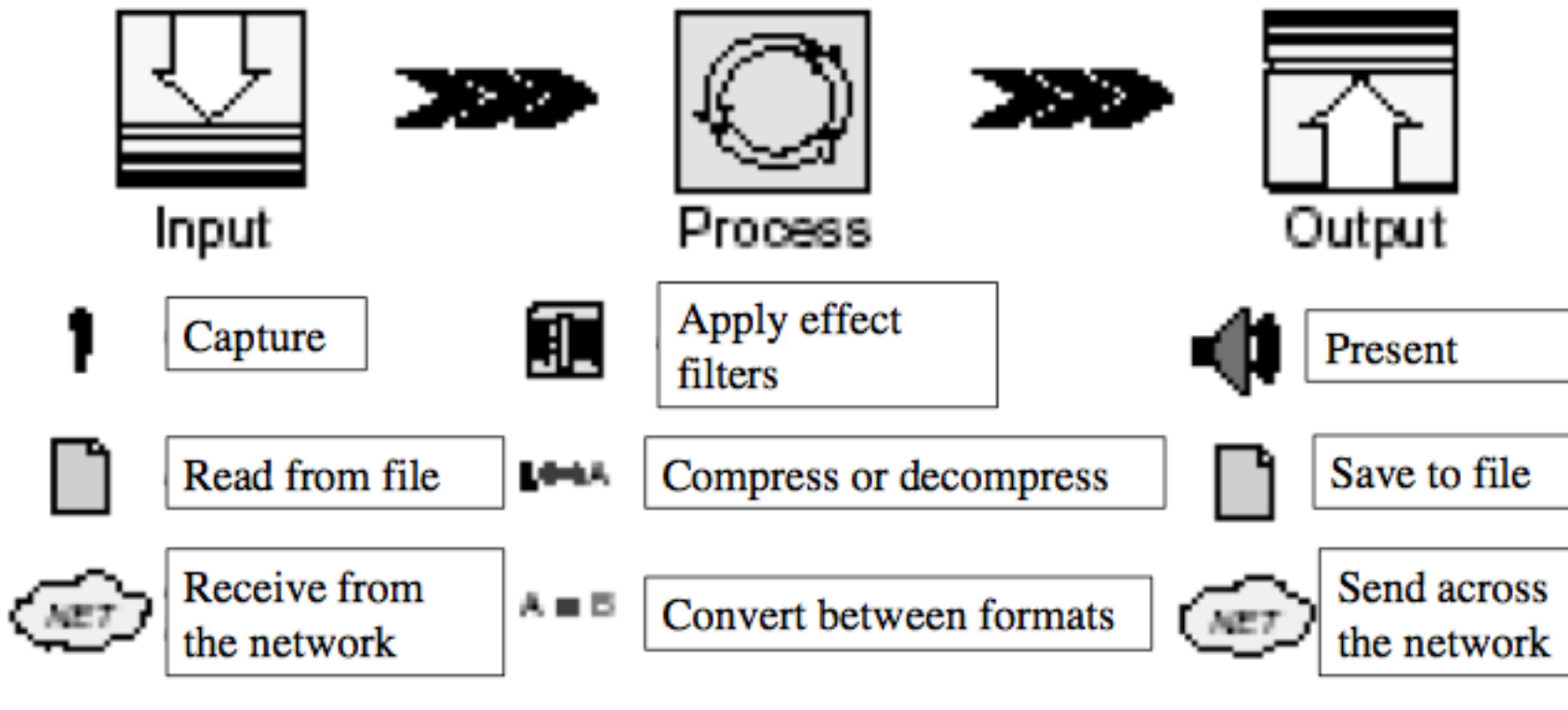
Medientechnik

Übung 4 - Tutorial

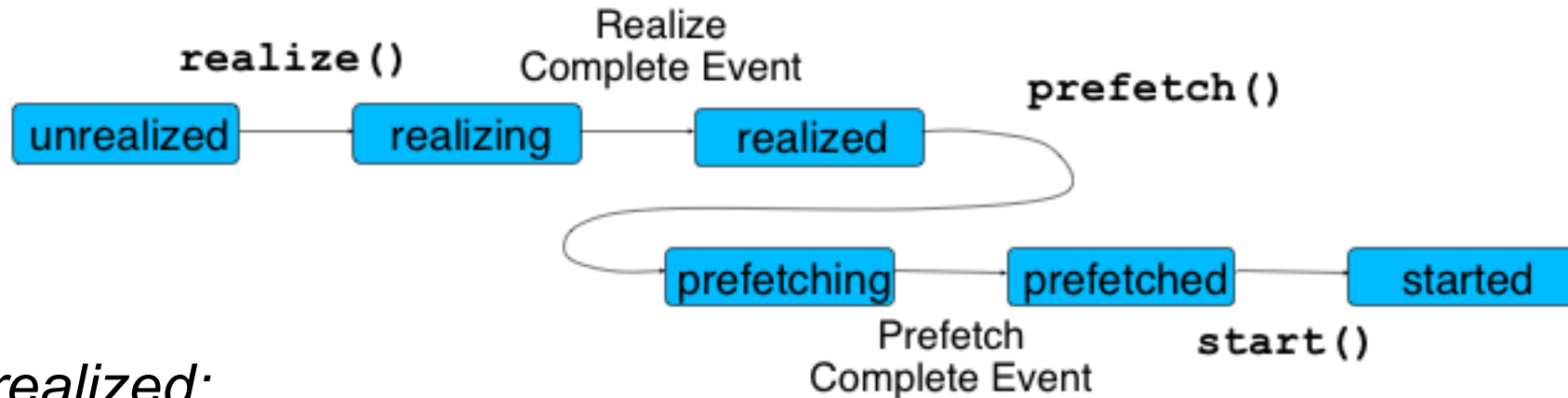
Überblick

- Kurze Wiederholung der Vorlesung
 - Player
 - Processor
- Programmieraufgabe 1:
 - SimplePlayer implementieren
- Programmieraufgabe 2:
 - GrayEffect implementieren

Überblick JMF



Player



Unrealized:

- Anfangszustand

Realizing:

- Medienabhängige Teile des Players werden bereitgestellt

Prefetching:

- Eingabestrom wird soweit gelesen wie nötig, um Puffer zu füllen

Started:

- Verarbeitung läuft

Ereignis-Konzept in JMF

Ereignisse werden wie in AWT/Swing durch *callback* realisiert

Bei einem **Player** werden Objekte mit **addControllerListener** registriert, die Controller-Ereignisse interpretieren

```
public interface javax.media.ControllerListener {  
    public void controllerUpdate(ControllerEvent event)  
}
```

Beispiele für Controller-Ereignisse (Unterklassen von **ControllerEvent**):

RealizeCompleteEvent

PrefetchCompleteEvent

StartEvent

StopAtTimeEvent

EndOfMediaEvent

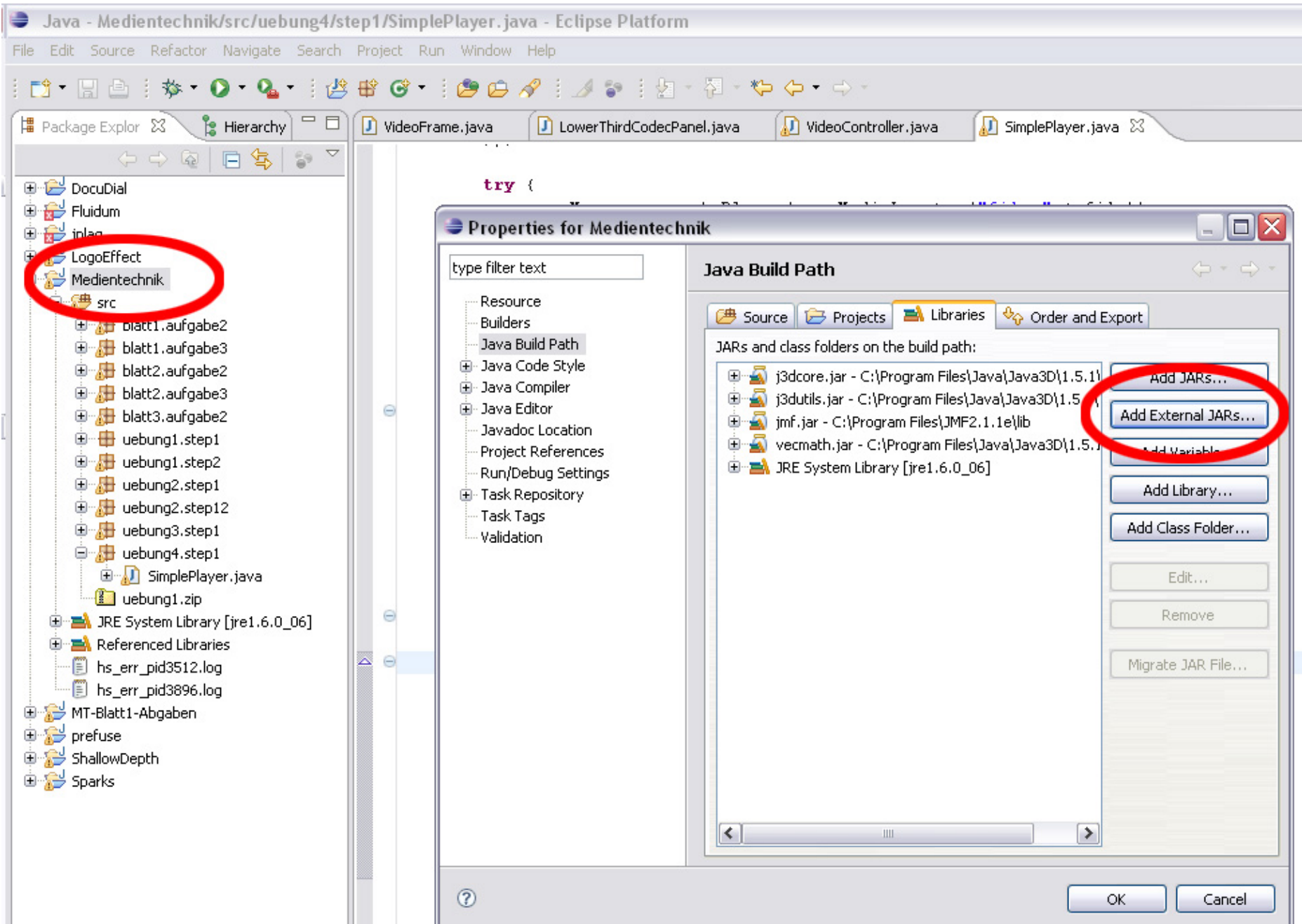
FormatChangeEvent

RateChangeEvent

StopTimeChangeEvent

Programmieraufgabe 1

- Ziel: SimplePlayer realisieren
- Vorgehensweise
 - Vorlage von der Homepage herunterladen
 - Neues Java Projekt anlegen
 - Jmf.jar als library hinzufügen:
 - Rechtsklick Projekt -> Build Path -> Configure Build Path
 - Button "Add External JARs..." -> jmf.jar auswählen
 - Neue Klasse "SimplePlayer.java" anlegen
 - Events des Players abfangen
 - Video starten und loopen

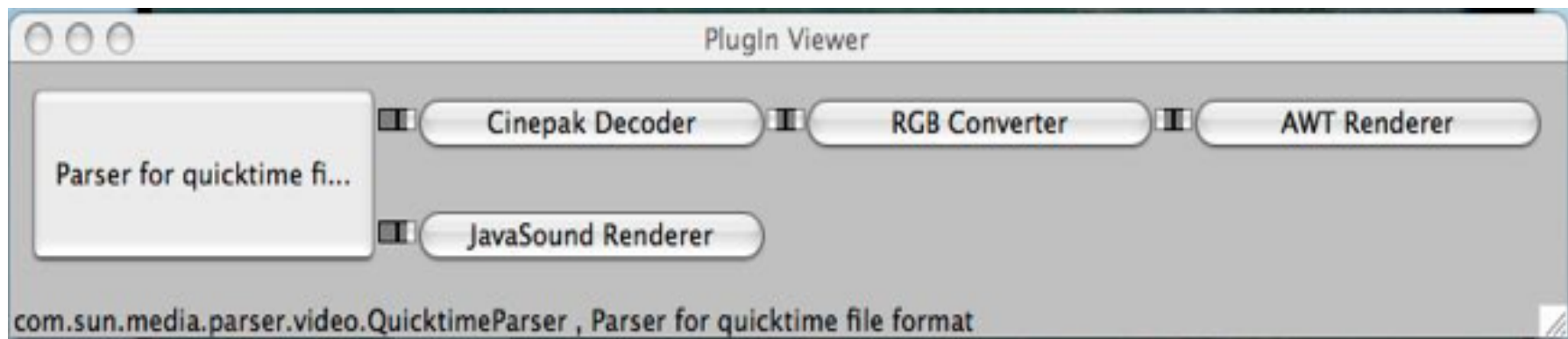
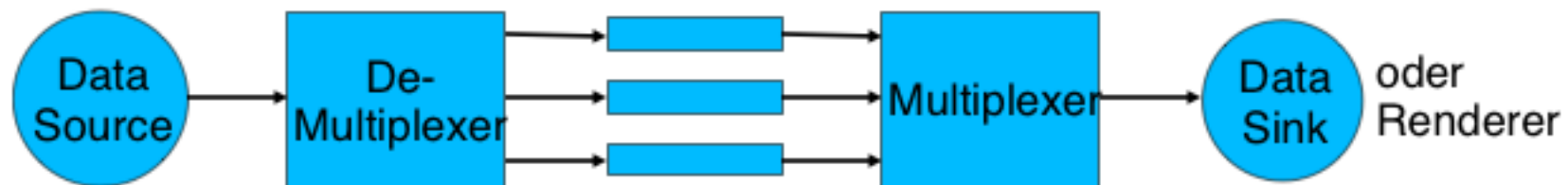


Motivation zur Aufgabe 2

- Bisher (mit Player) nur einfaches Abspielen, Vor-/Zurückspulen, Stop möglich
 - => Wie können wir das Video direkt manipulieren?
 - => Wie können wir etwas in die Verarbeitungskette des Videos einfügen?
 - => Wie können wir eigene Effekte implementieren?

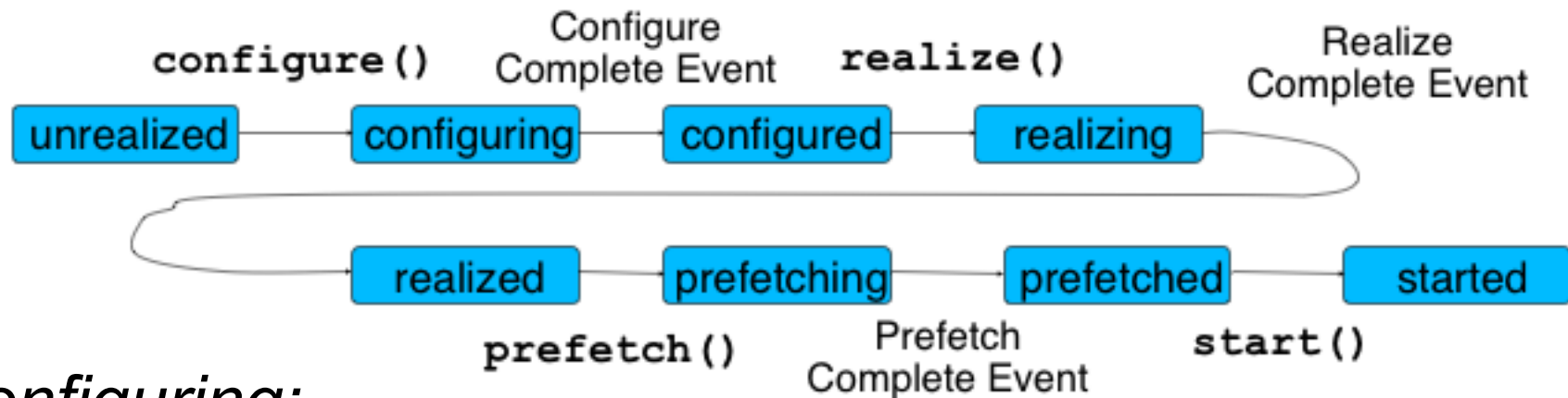
Verarbeitungsketten

- Datenquelle wird in einzelne Tracks (Video, Audio) aufgespalten (Demultiplexer)
- Die Tracks können nun separat bearbeitet werden (z.B. Effekte hinzufügen)



Processor

- Detaillierteres Event-Modell als Player
- Einfügen von eigenen Demultiplexern, Codecs, Effekten, Multiplexern etc. möglich



Configuring:

- Die Eingabe wird auf die enthaltenen Medien (Spuren, *tracks*) analysiert

Configured:

- Bearbeitung für die einzelnen Spuren kann separat definiert

Programmieraufgabe 2

- Ziel: Gray-Effect implementieren
- Vorgehensweise
 - GrayEffect.java öffnen
 - In der process(..)-Methode des Effects jeden Pixel auf Grauwert setzen (ähnlich zum ÜB3)