

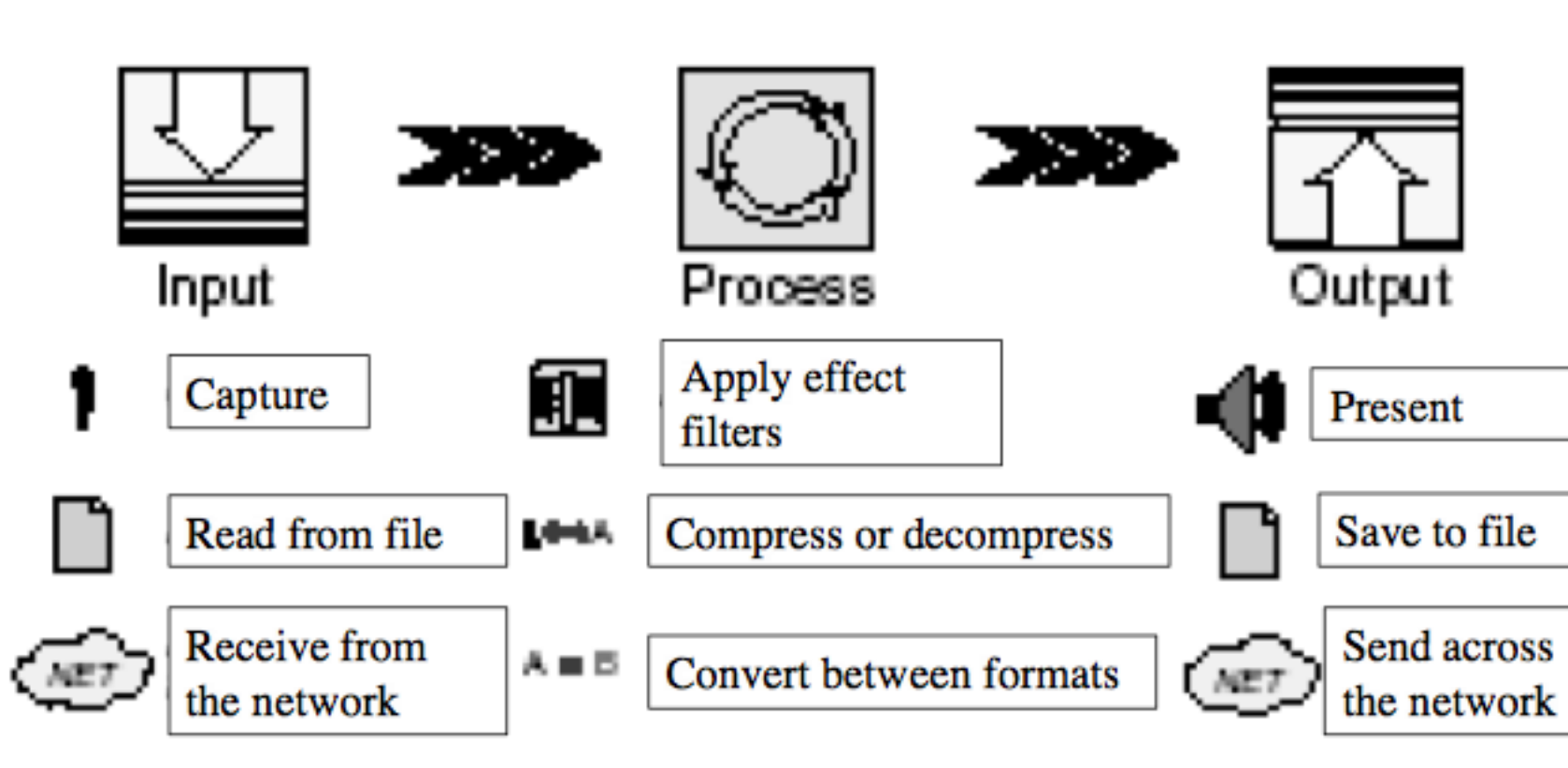
Medientechnik

Übung 4 - Tutorial

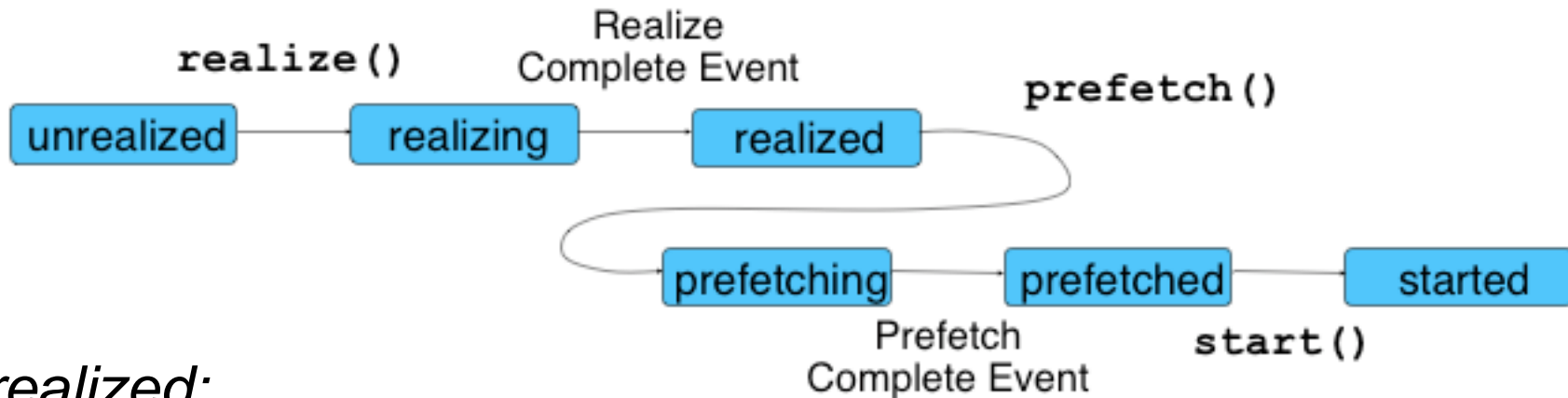
Überblick

- Kurze Wiederholung der Vorlesung
 - Player
 - Processor
- Player implementieren
- Effekt 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)  
}
```

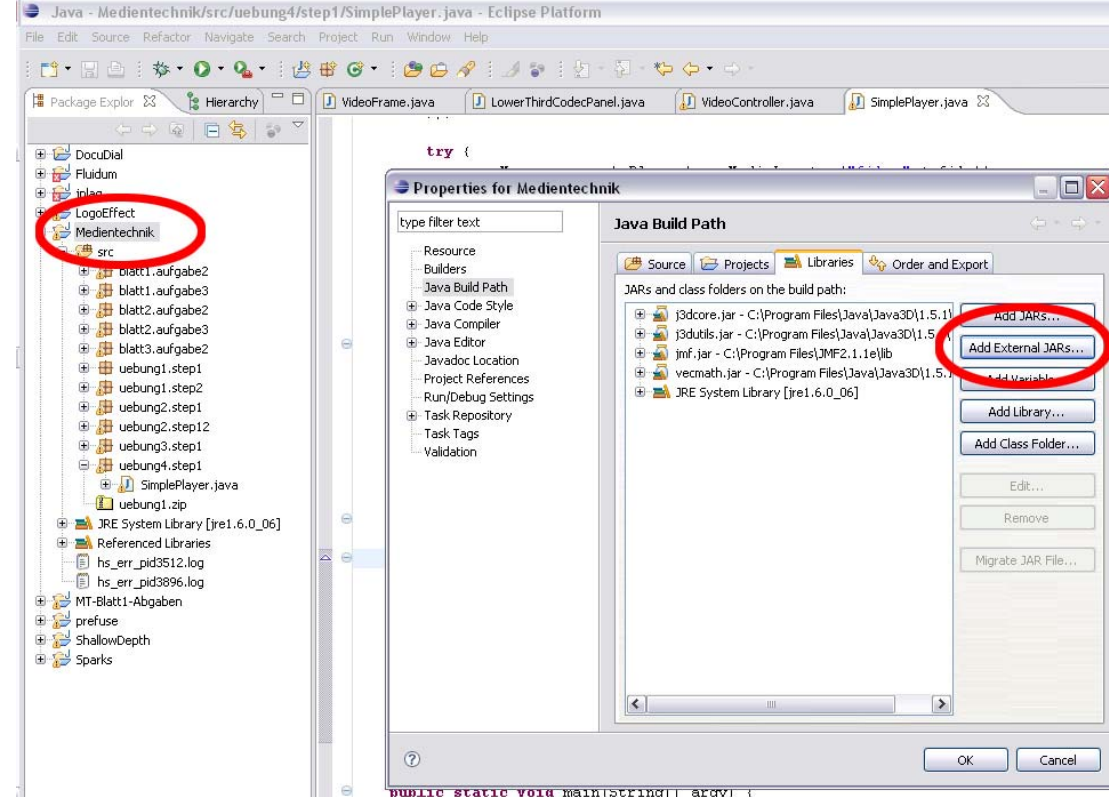
Beispiele für Controller-Ereignisse (Unterklassen von `ControllerEvent`):

- `RealizeCompleteEvent`
- `PrefetchCompleteEvent`
- `StartEvent`
- `StopAtTimeEvent`
- `EndOfMediaEvent`
- `FormatChangeEvent`
- `RateChangeEvent`
- `StopTimeChangeEvent`

Programmieren

- Einfachen Player realisieren
- Events des Players abfangen
- Video starten und loopen

CIP-Pool JMF



Rechtsklick Projekt -> "Properties"

links: "Java Build Path"

Button "Add External JARs..."

jmf.jar aus

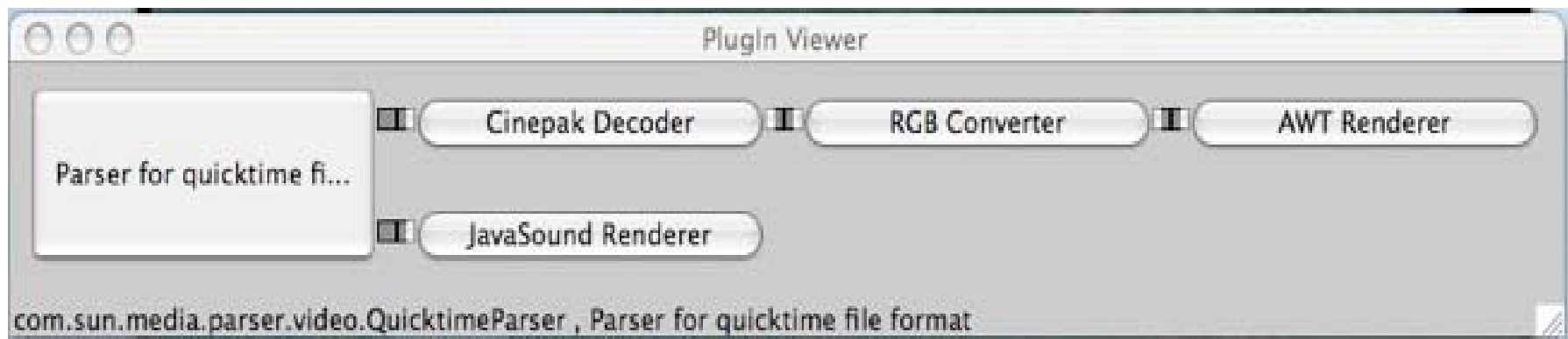
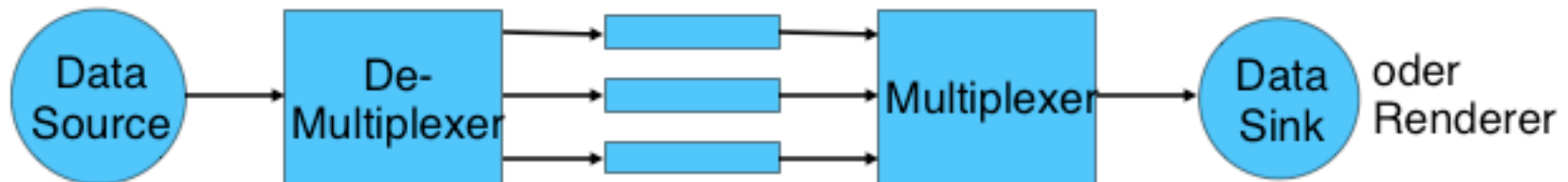
**/soft/IFI/lang/j2sdk-1.4.2_03-sun/iX86-unknown-
linux/jre/lib/ext**

Motivation

- 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?

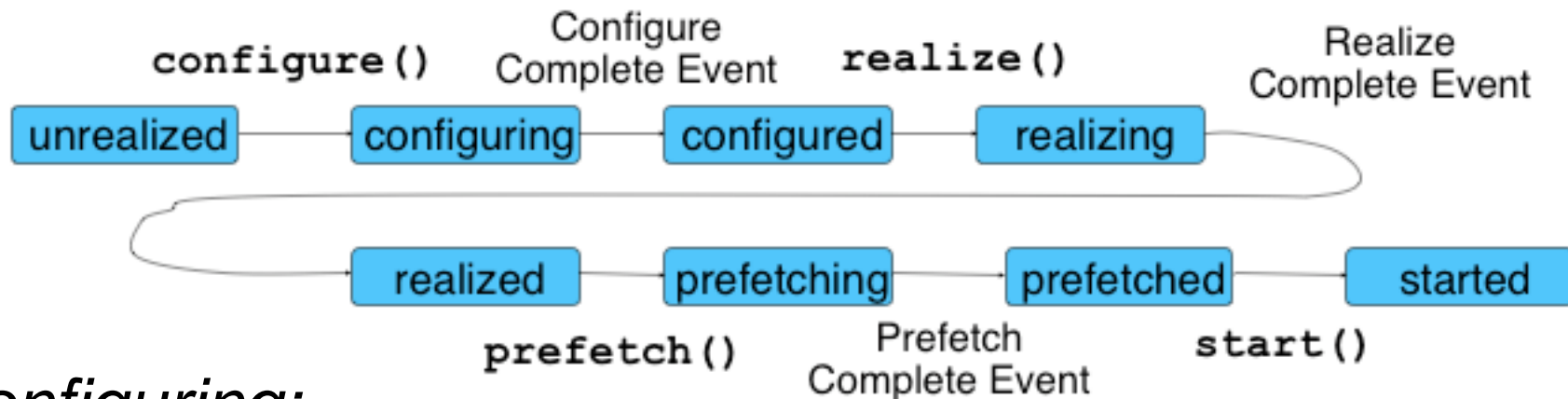
Verarbeitungsketten

- 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 werden

Programmieren

- Vorlage von der Homepage
- Grey-Effect implementieren
- In der process(..)-Methode des Effects jeden Pixel auf Grauwert setzen (ähnlich zum ÜB3)