

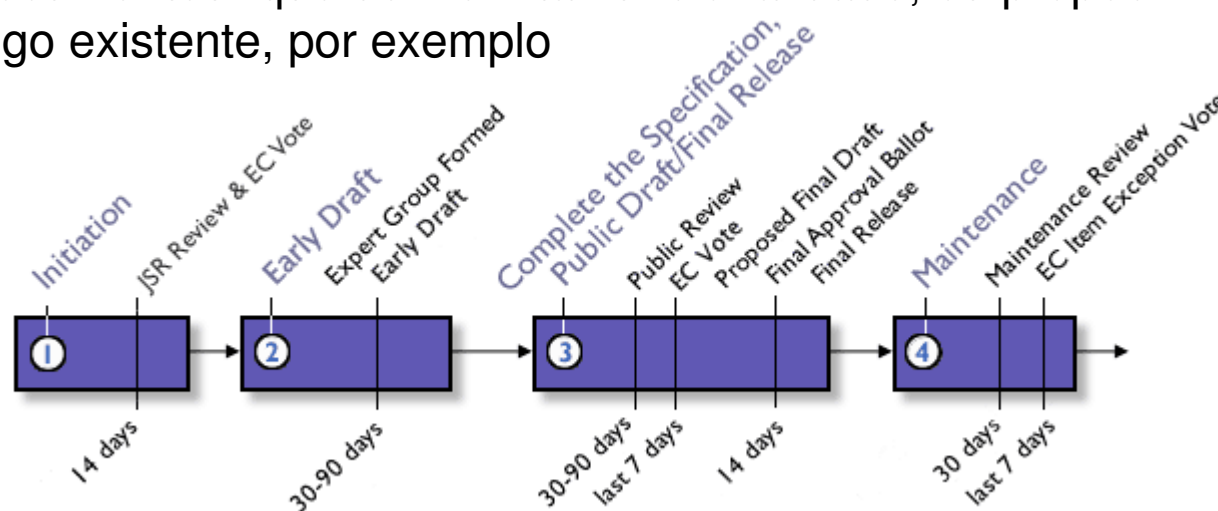
# API Java Stream Assembly (JSA) JSR 158

Raoni Kulesza  
raoni@larc.usp.br

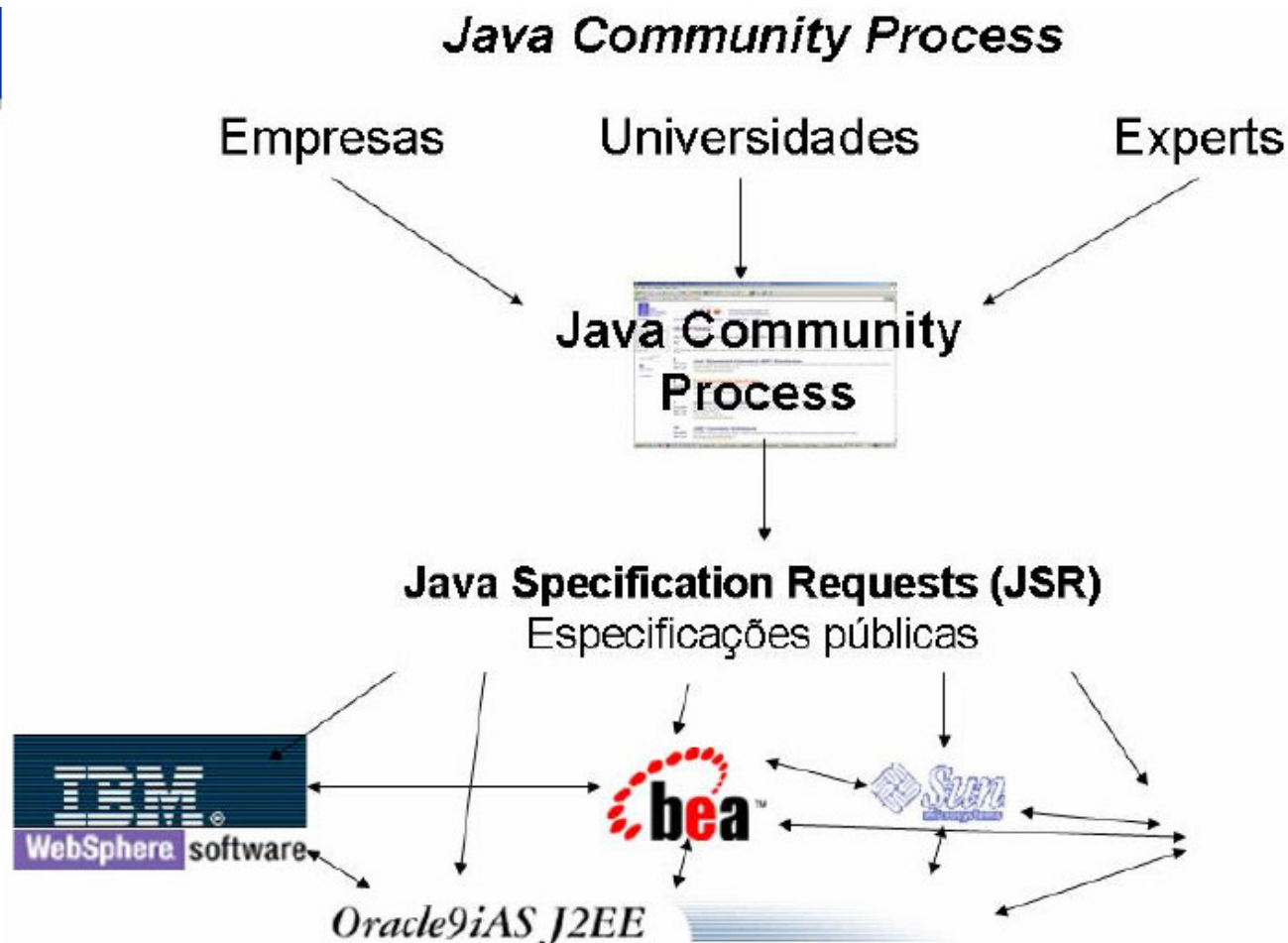
# Java Community Process



- **JCP** (Java Community Process): É uma organização internacional de desenvolvedores e fabricantes cuja missão é manter um processo aberto de desenvolvimento e revisão das especificações, referências e outras implementações relacionadas à tecnologia Java
- **JSR** (Java Specification Request): São as propostas de mudanças e novas implementações para a tecnologia Java. Em outras palavras, são os "documentos" que definem tal funcionalidade, ou propõem mudanças em algo existente, por exemplo



# Java Community Process



# Java Stream Assembly



## Idéia básica

- Definir uma API Java com classes e interfaces para criar, gerenciar e processar fluxos de mídia.
- **JSR: 158 / Status: Em progresso**
  - Public Review
  - Community Draft Ballot
  - Community Review
  - Expert Group Formation
  - JSR Review Ballot
- **Empresas participantes**
  - Alticast
  - Kasenna
  - Strategy & Technology
  - Sun Microsystems
  - Video Propulsion

## Fluxos, Multimídia e Redes

- Um **fluxo de mídia** pode ser um fluxo de informações de:
  - **áudio**, ou **vídeo**, ou **dados** contínuos e dependentes do tempo
- Um **fluxo multimídia multiplexado** é um único fluxo que contém múltiplos e intercalados fluxos de mídia
- **Multiplexação** é o processo de combinar vários fluxos de mídia em um mesmo fluxo que posteriormente possa ser demultiplexado. Pode ser **físico** ou **lógico**
- **MPEG**, Elementary Streams (**ES**), Program Stream (**PS**), Transport Stream (**TS**), Single Program Transport Stream (**SPTS**) e Multiple Program Transport Stream (**MPTS**)
- Por fim, os fluxos devem usar alguma **codificação** (MPEG-2, MPEG-4) e transportados pela rede através de **interfaces físicas** (ASI ou Ethernet) e com utilização de **protocolos** (IP, UDP, RTP, RTCP, RTSP, MPEG-2 TS, DSM-CC)

# JSA – Requisitos do Domínio



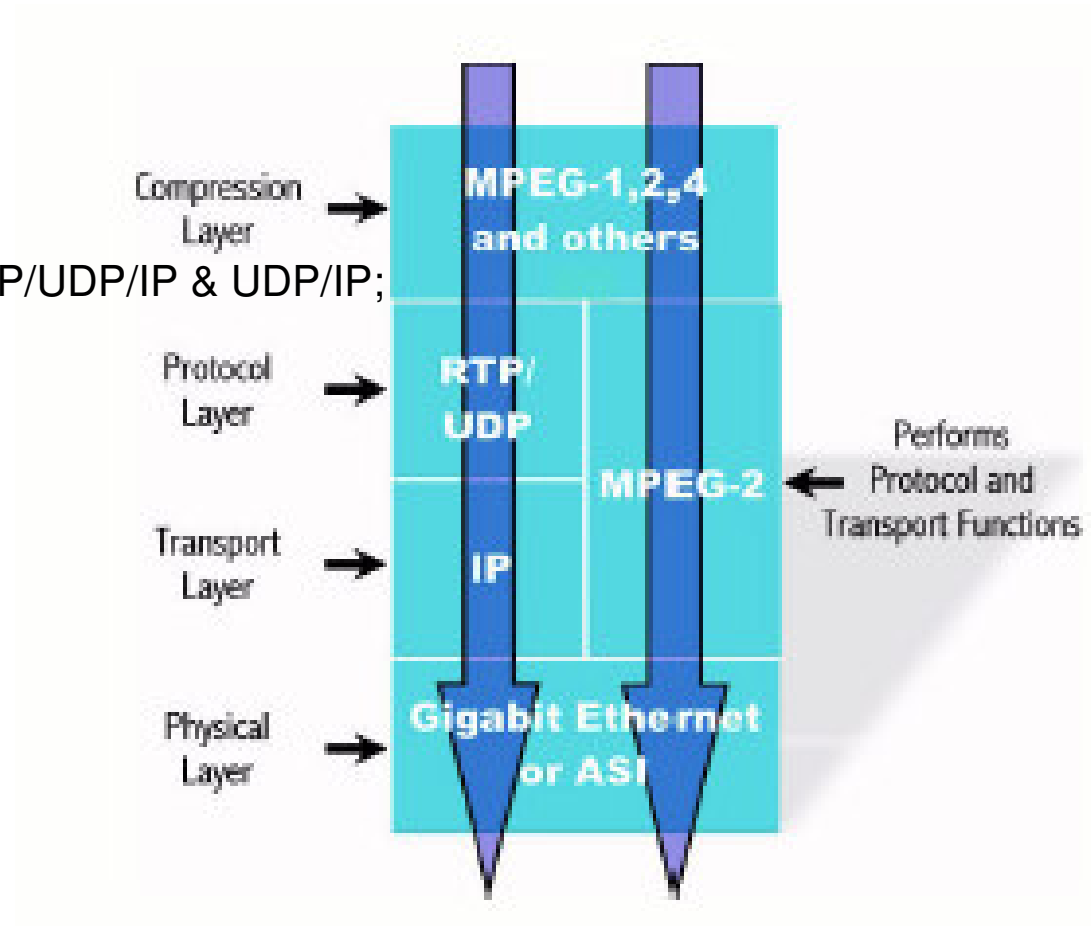
- Dois desafios devem ser superados antes que se possa acessar um fluxo de mídia:
  - Processar e transportar grande volume de dados contendo fluxos de mídia;
  - Os dados, em um fluxos de mídia, devem ser processados em tempo real.
- Algumas funções típicas de manipulação de fluxos são:
  - Combinação de áudio, vídeo e dados para criar um SPTS;
  - Inserção de novos dados que não áudio e vídeo em um TS (por exemplo, informações textuais, propaganda, software ou metadados);
  - Multiplexar SPTSs para criar um MPTS;
  - Remover um programa do MPTS.

# JSA - Escopo



## A API JSA abstrai:

- Camada física de rede
  - ASI, GigE, FastEthernet;
- Protocolos de transmissão
  - MPEG2-TS, IP-based – RTP/UDP/IP & UDP/IP;
- Protocolos de controle
  - DSM-CC, RTSP.



## Finalidade

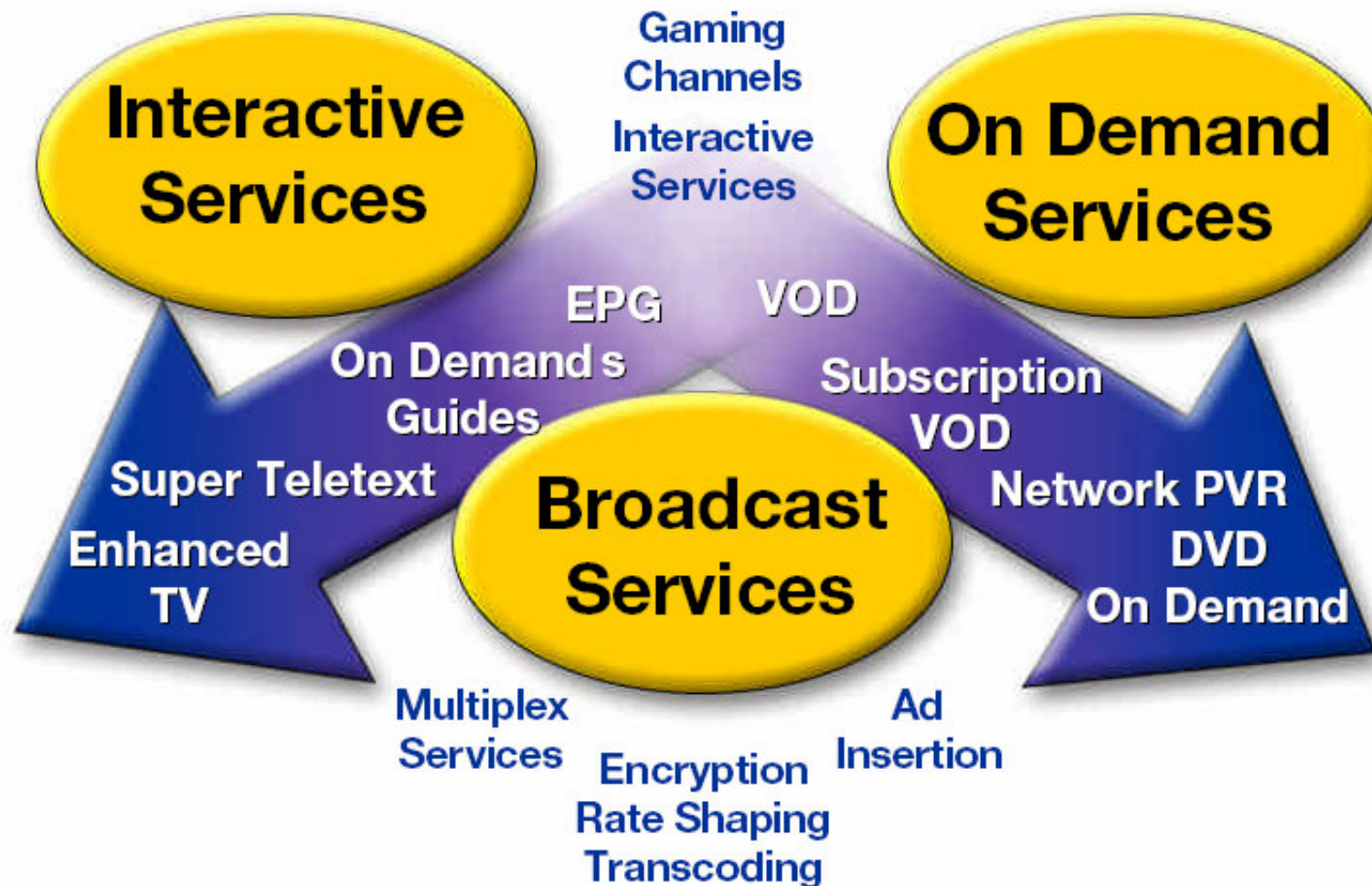
- A API JSA prove uma interface padrão para controlar componentes de hardware e software usados para criar, processar, multiplexar e entregar fluxos de multimídia multiplexados.

## Requisitos de projeto

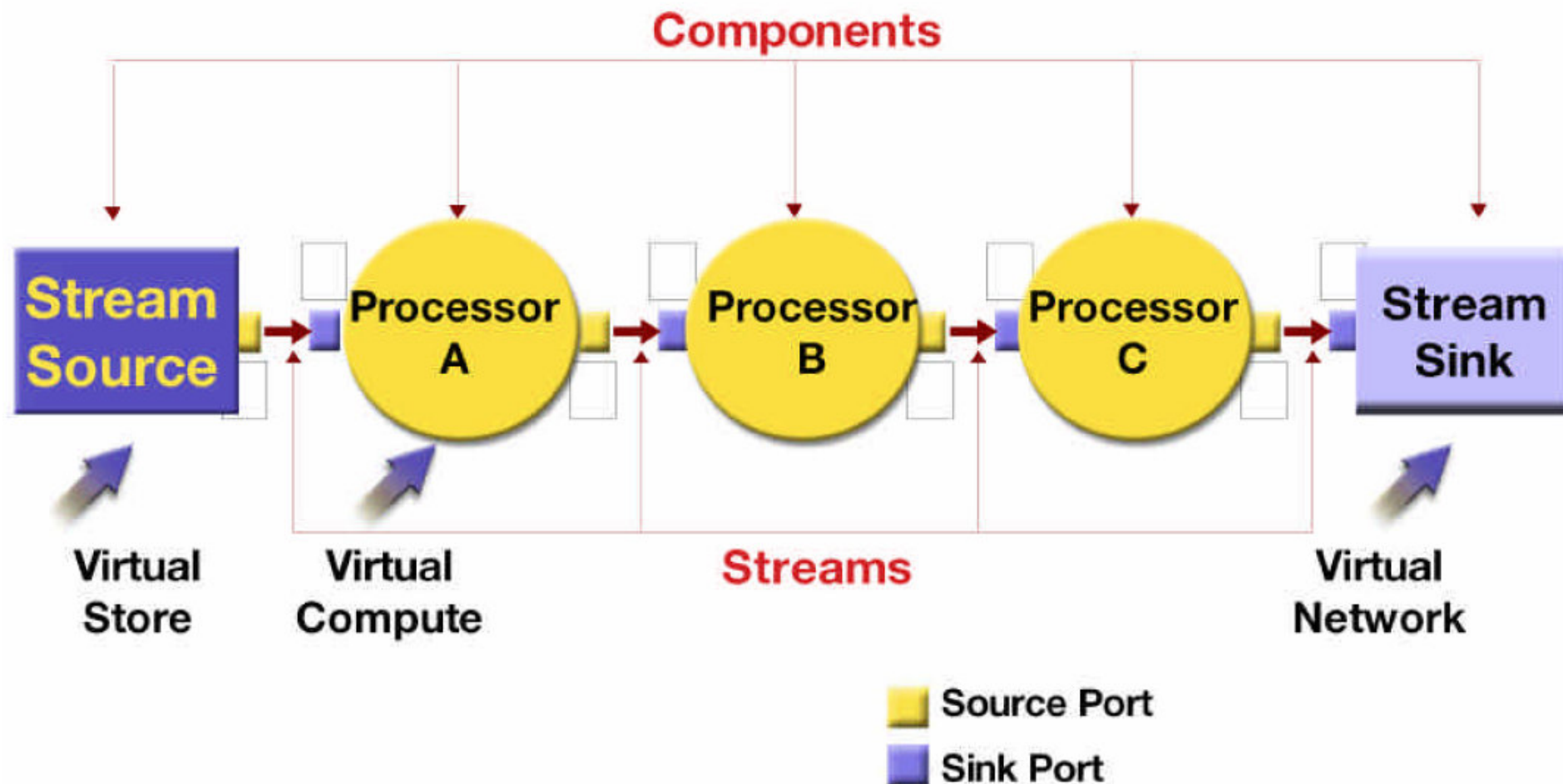
- Deve ser capaz de descobrir e configurar multiplexadores;
- Deve prover funções de controle de montagem, iniciar, parar, adicionar e descartar *fluxos*;
- Deve suportar tanto fluxos de MPEG-2 *transport* quanto fluxos IP;
- Deve suportar a monitoração do relatório de eventos de fluxos , do mesmo modo que eventos de mudanças de pacote;
- Estabelecer e controlar sessões;
- Contemplar aspectos relacionados a tempo real.



# JSA - Cenários de uso



# JSA - Visão Geral

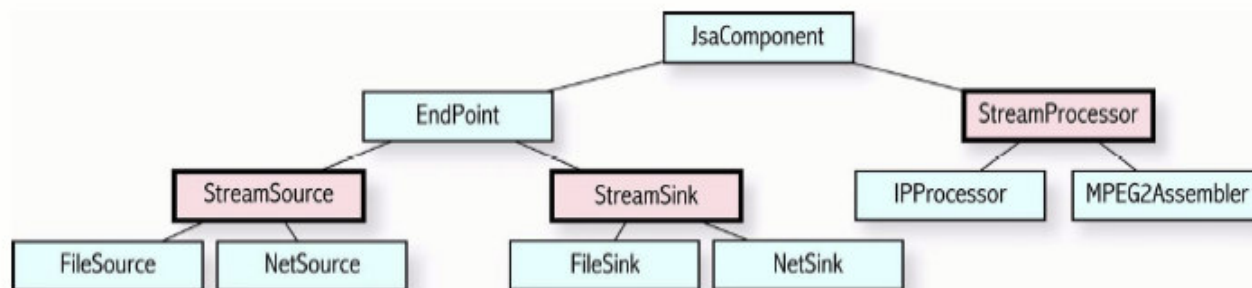


# JSA – Principais Componentes



- Camada de Dados

- Port
- Stream
- StreamSet
- Component
- EndPoint
- Processor



- Camada de Controle

- Control
- Media Position
- Clock Position
- Sample Position
- Session

- Outros:

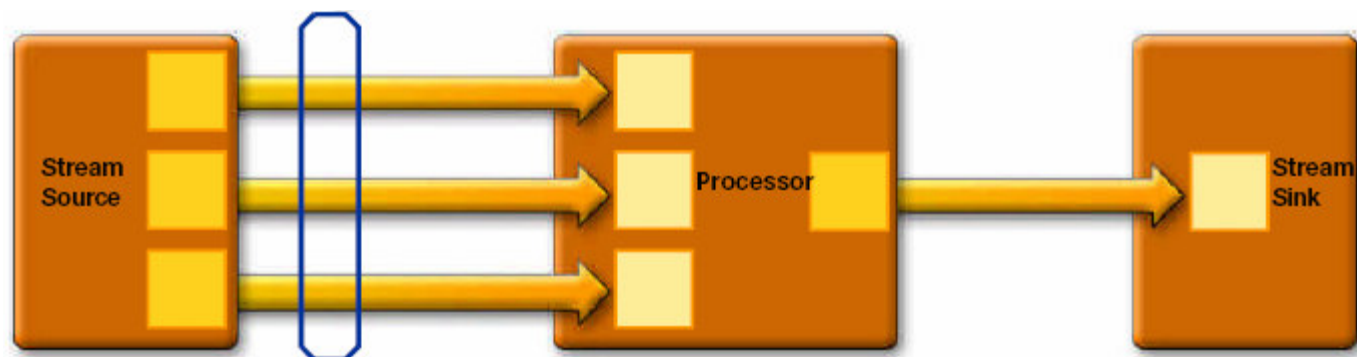
- *Asset, Discovery e Permission* para Componentes

# Componentes JSA



A especificação da JSA consiste de entradas de fluxos de mídia (**sources**), **processors** para manipular os fluxos e saídas de fluxo de mídia (**sinks**)

## Configuração



# Componentes JSA



- *Streams*;
  - *Stream* de mídia que flui entre uma porta de saída para uma porta de entrada (*source* & *sink*)
- Portas;
  - Representam os pontos de entrada e saída de um componente (dispositivo físico ou processos em execução) – *source port* e *sink port*
- Componentes – *Stream Sources* e *Stream Sinks / Processors*.
  - *Stream Source* (StreamSource) – componentes com 1 ou + portas de saída;
  - *Stream Sink* (StreamSink) – componentes com 1 ou + portas de entrada;
  - *Processors* (StreamProcessor) - componentes com 1 ou + portas de entrada e saída. Manipulam os *streams* usando funções da API (adicionar, descartar, inicializar, parar, pausar, etc)

## Processors (StreamProcessor)

Responsáveis por executar alguma operação no *stream*:

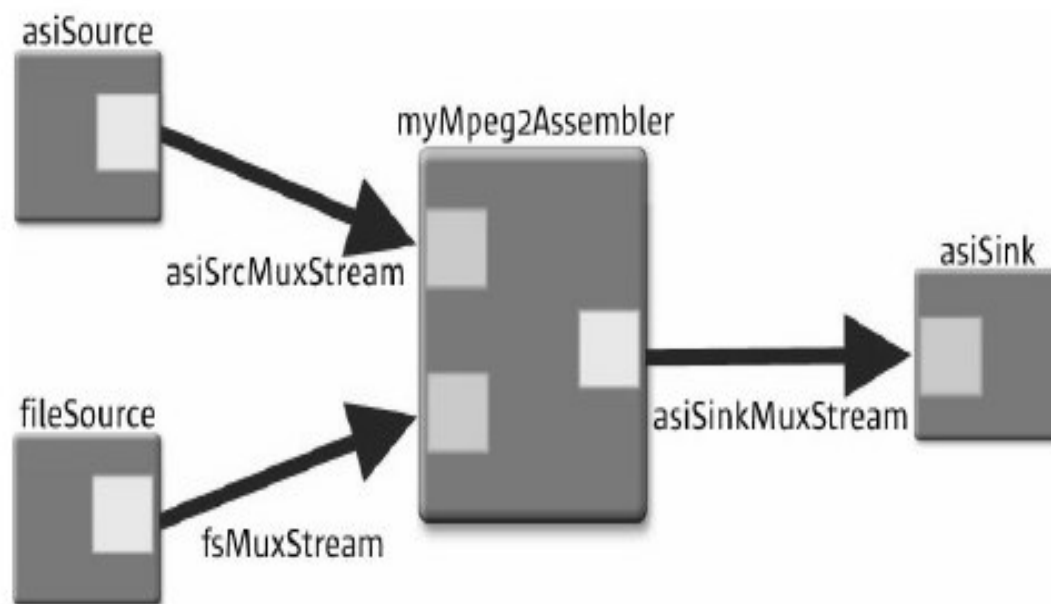
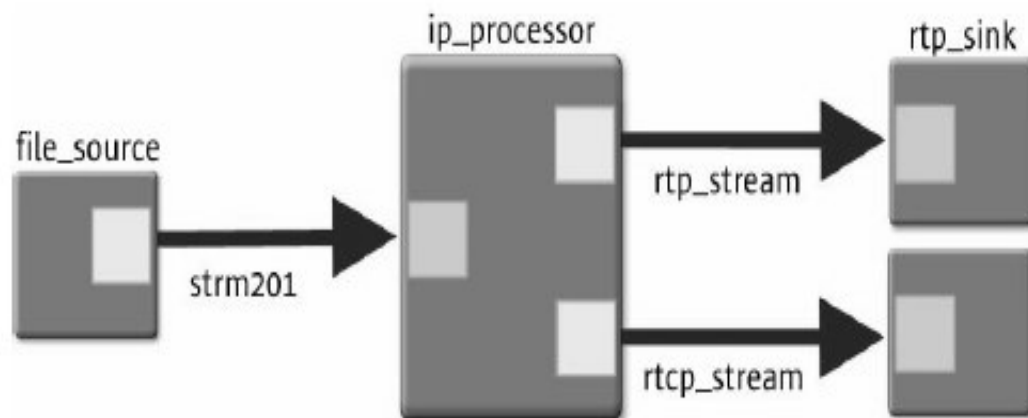
- Conectar um *stream source* a um *stream sink* (pass-through);
- Adicionar/Descartar streams de mídia específicos através dos multiplexadores;
- Renomear determinados PIDs para impedir colisões entre pacotes de diferentes *sources*.

## Control, ControlContext e EventListeners

Responsáveis por executar e interface para controle de componentes JSA

- Dois aspectos: controle e posição (control e position type)
- Controle: ações sobre o que fazer com os fluxos
- Posição: especifica onde atuar sobre o fluxo (position, clock, sample, packet, immediate)
- Controles podem implementar *listeners* para monitoração de eventos

# Exemplos JSA

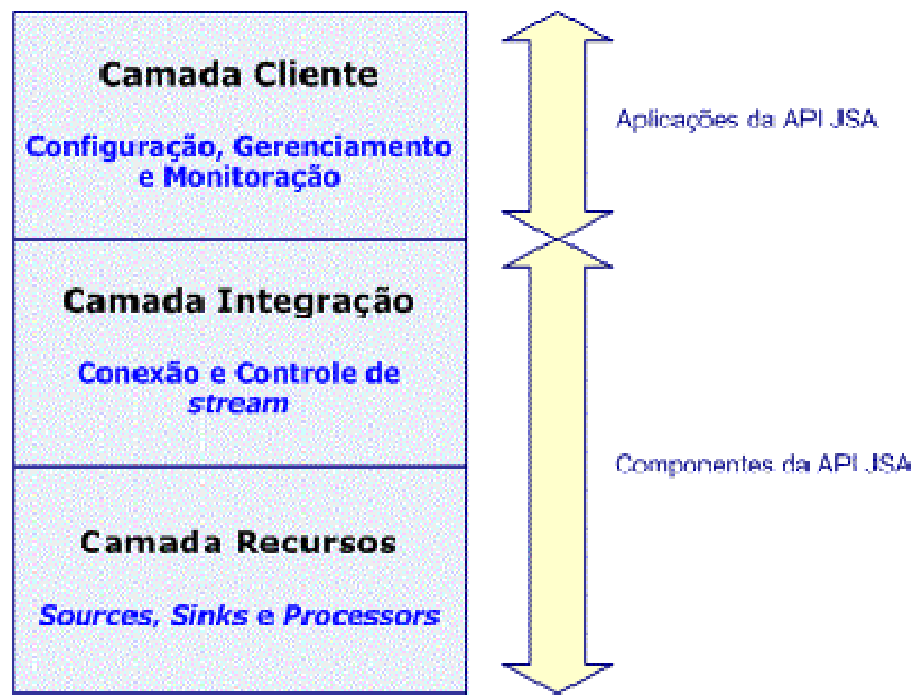


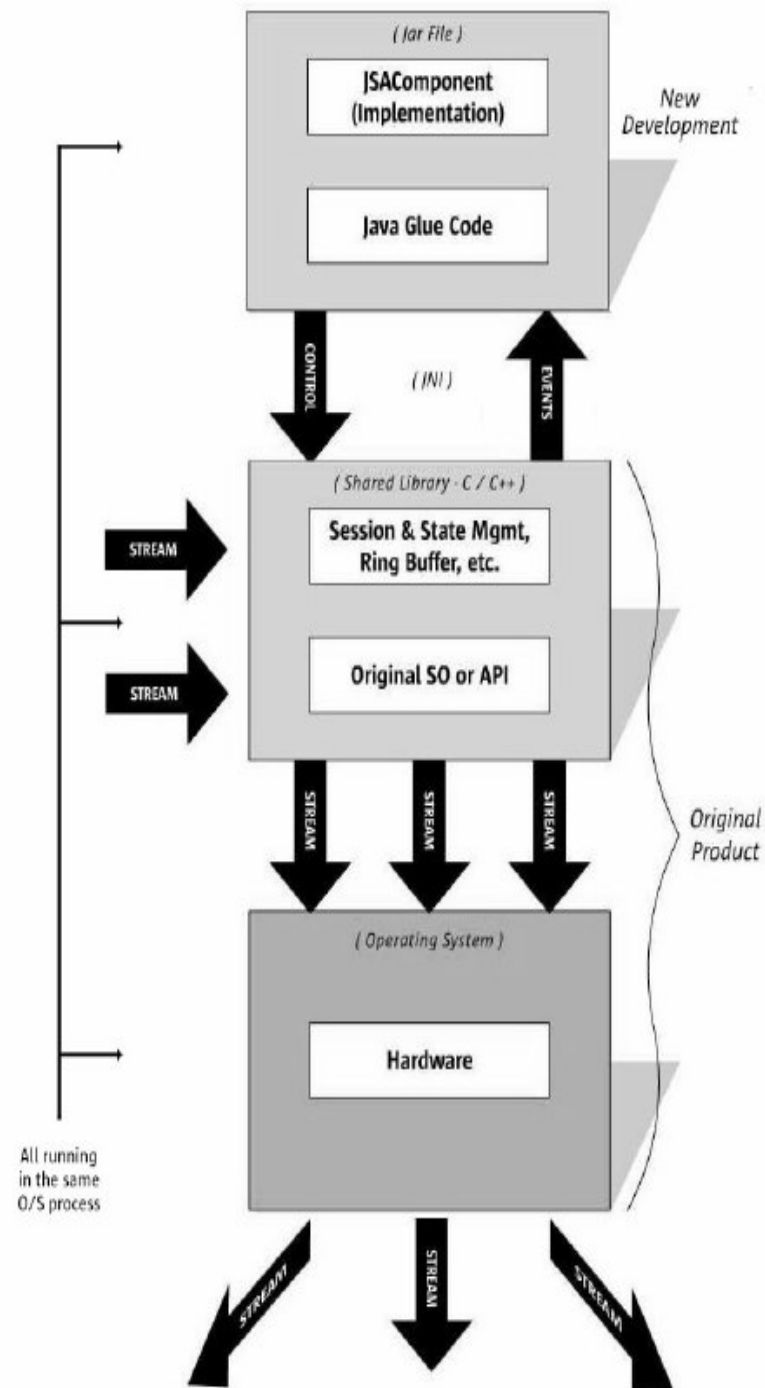


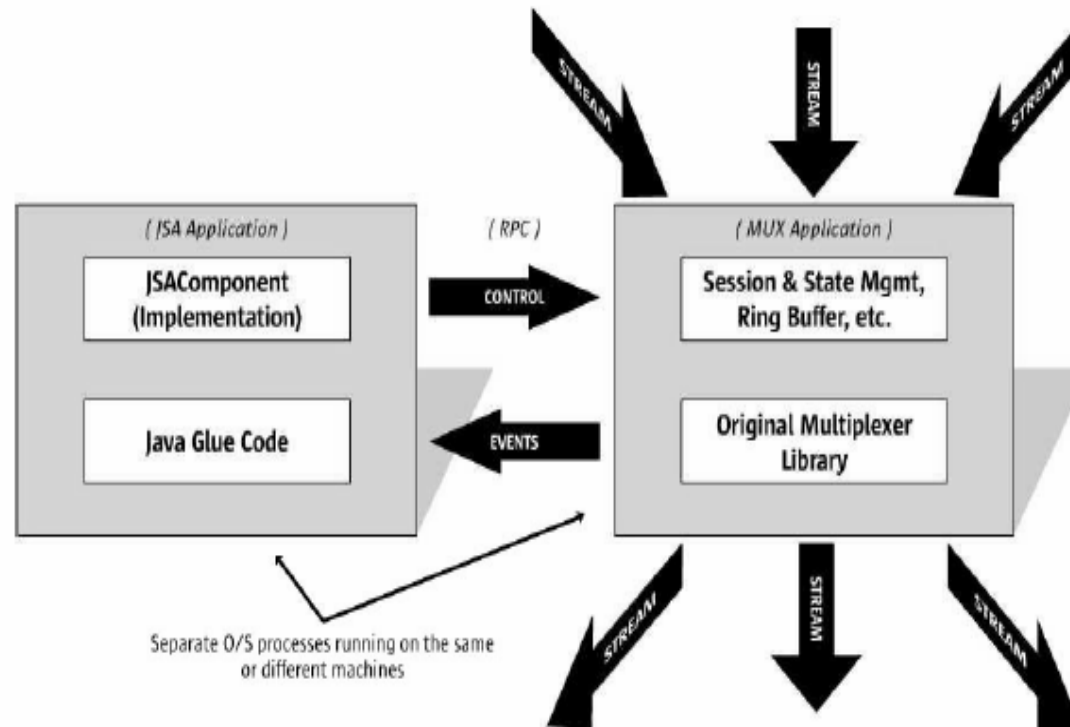
# Camadas – App & Componentes



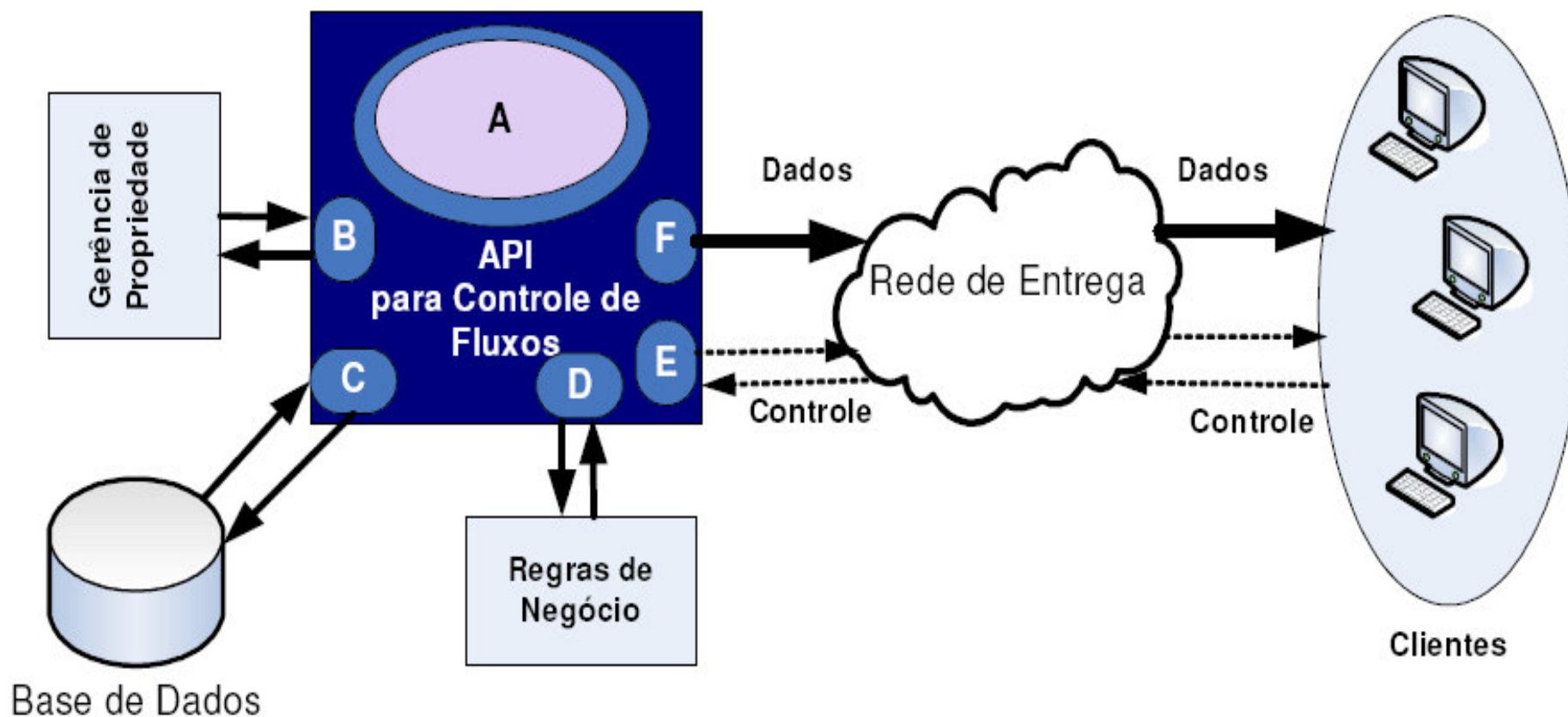
- Segundo a especificação, é conveniente dividir a JSA em 3 camadas.



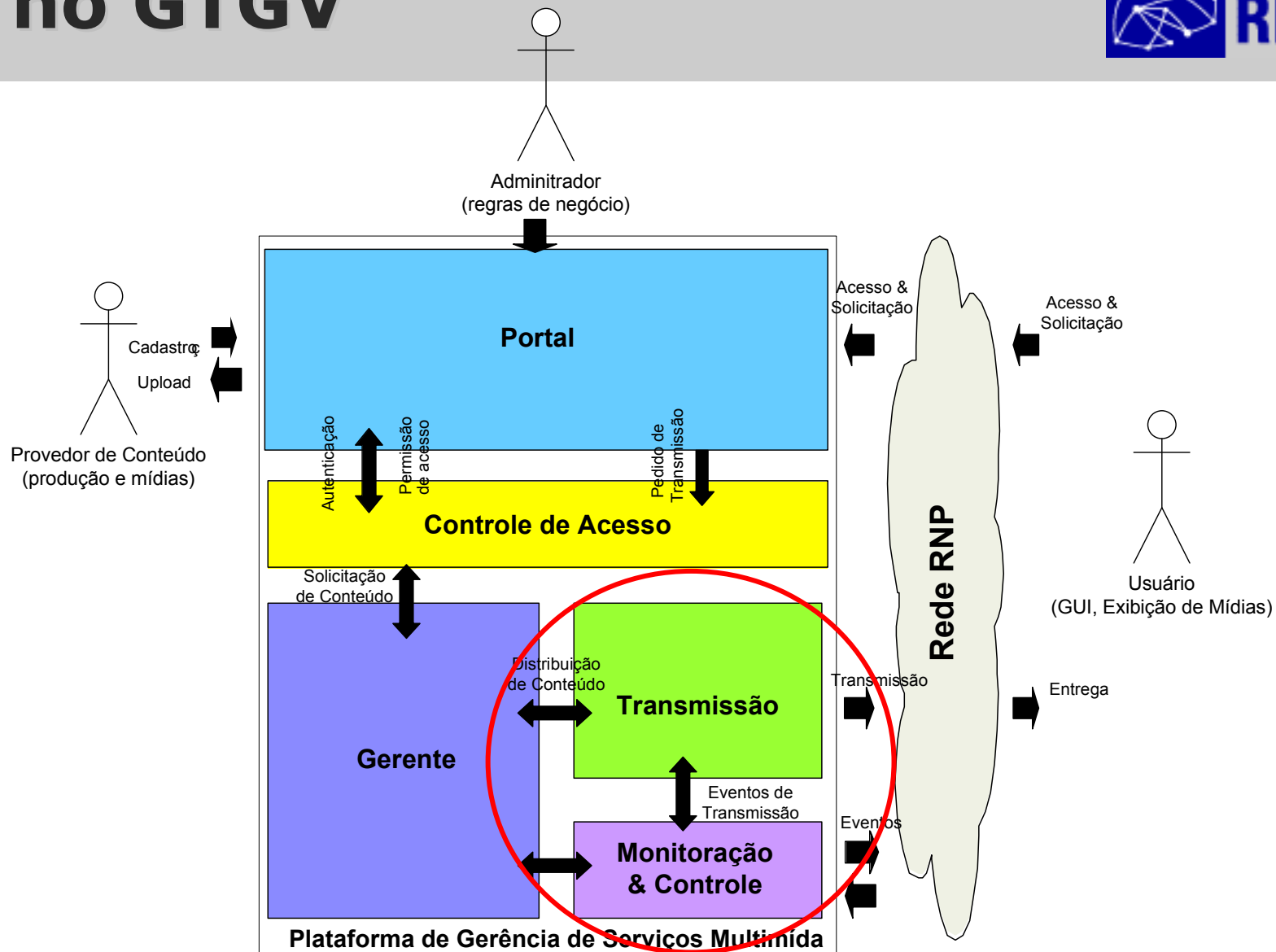




# JSA no GTGV



# JSA no GTGV



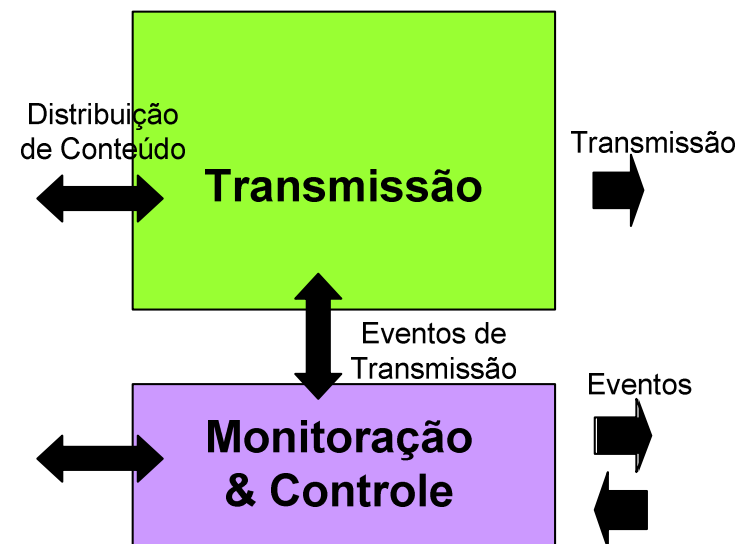
## Interfaces com Módulo Gerente

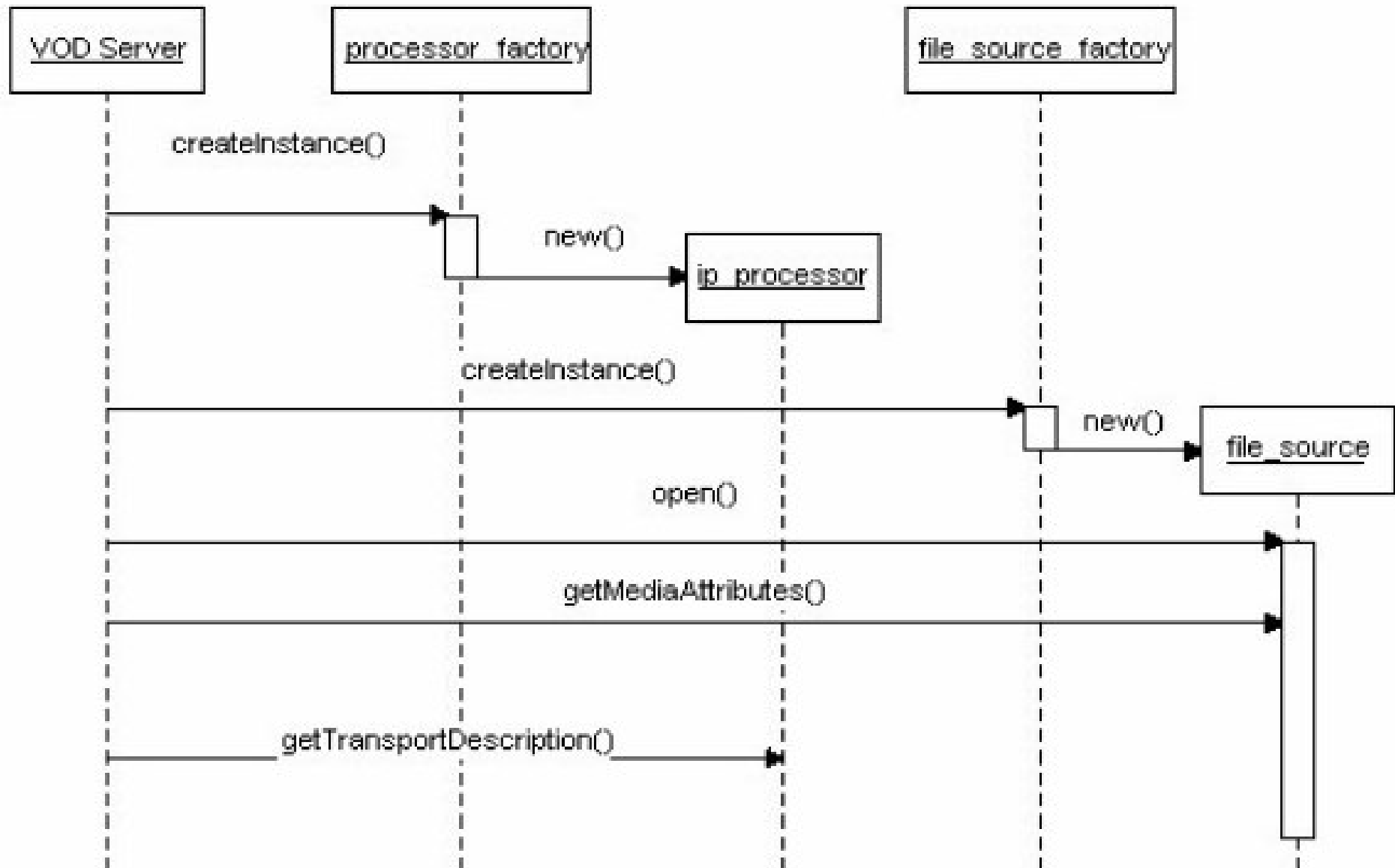
### Módulo de Transmissão

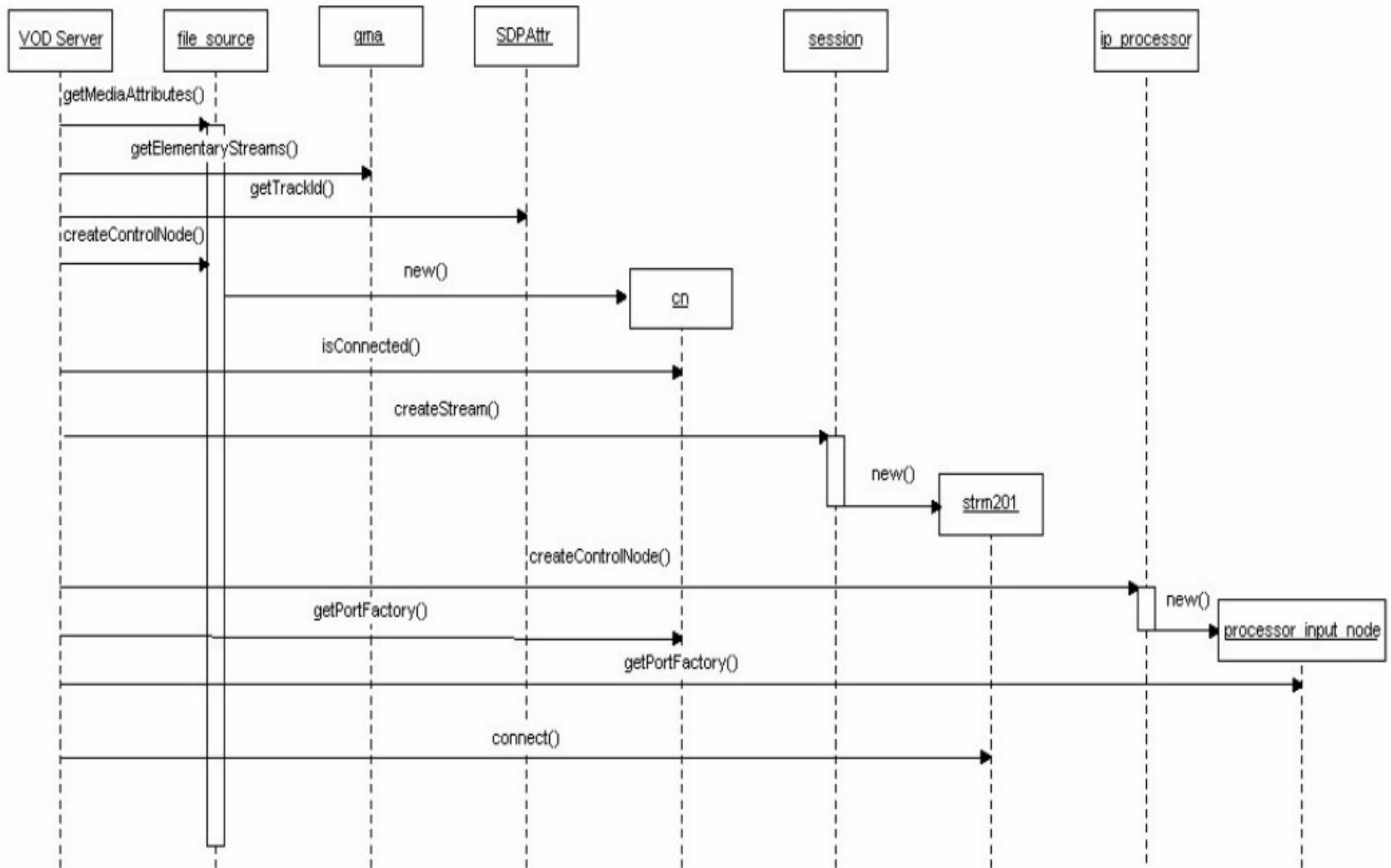
- Controle de Sessões de Transmissão
- Controle de Fluxos de Mídias
- Protocolos de Transporte e Rede

### Módulo de Monitoração

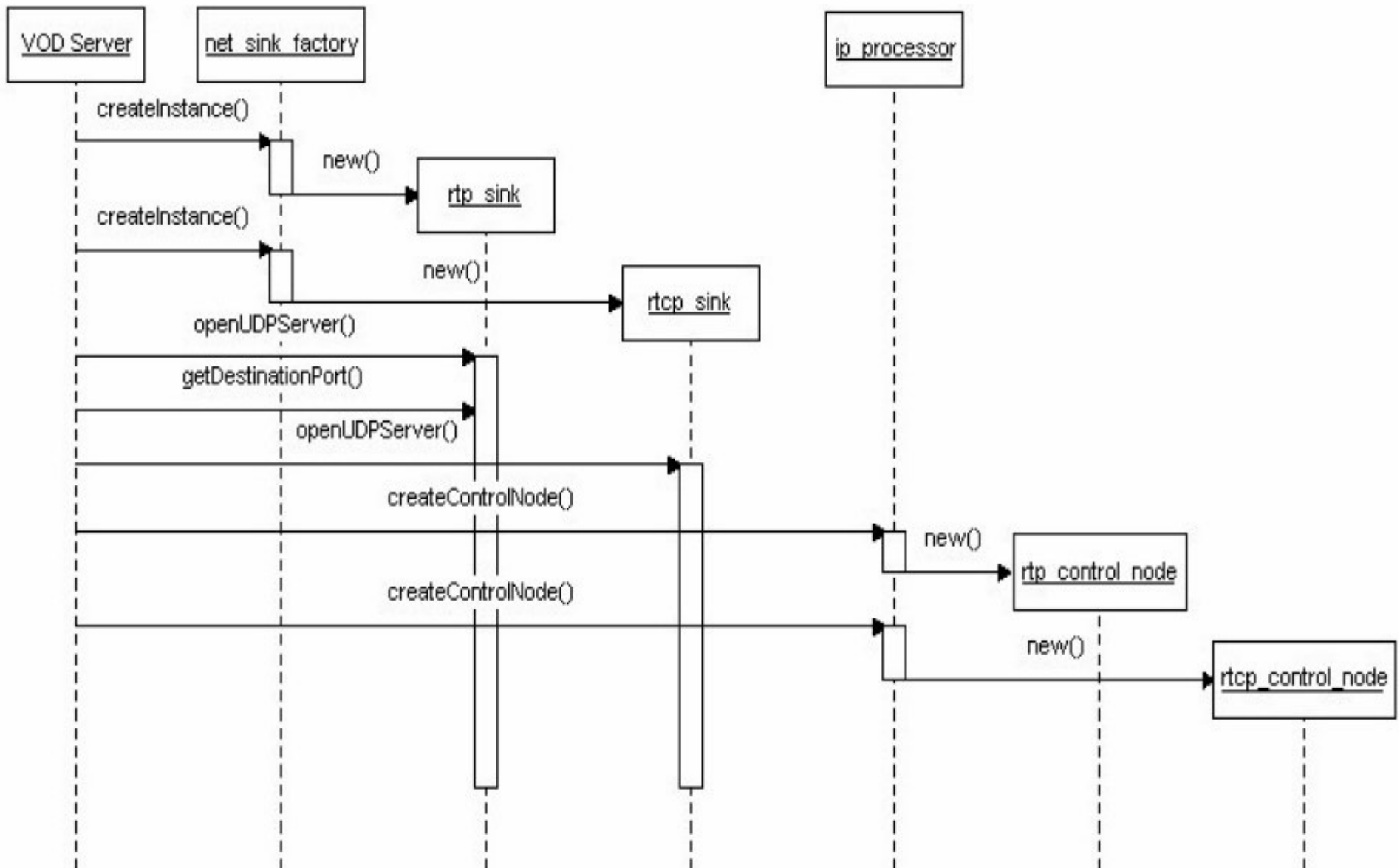
- Monitoração e Controle de Transmissão

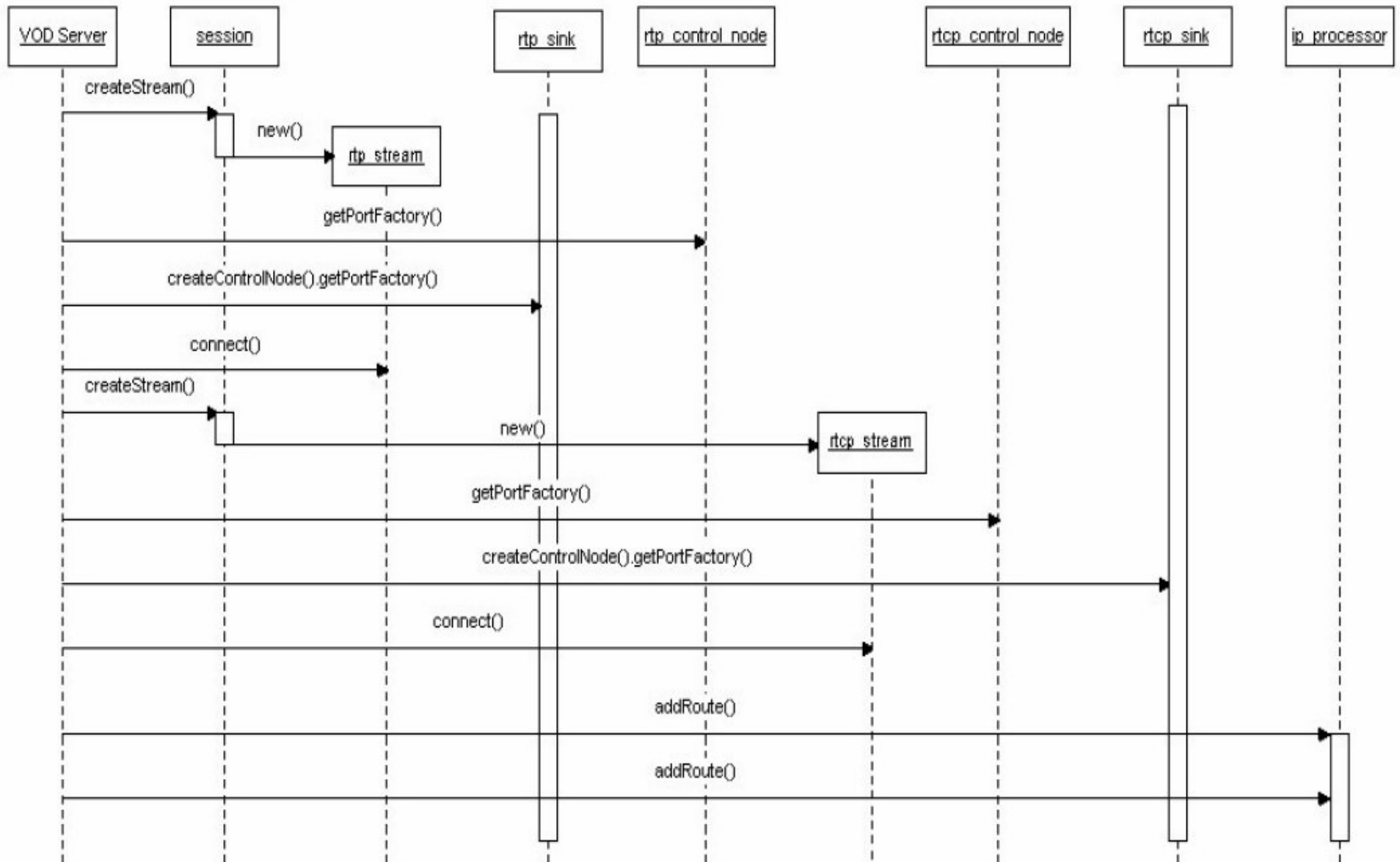












# Referências



- Site da JSR
  - <http://www.jcp.org/en/jsr/detail?id=158>
- Reference Implementation, Early Access
- Java Stream Assembly API – Programmer’s Guide, version 1.0, Public Review, 2003
- JavaOne 2003 – Session 2453