1. **Kurento API**

**Kurento Media Server** có thể được kiểm soát thông qua API mà nó tiết lộ, vì vậy các nhà phát triển ứng dụng có thể sử dụng ngôn ngữ để tương tác với nó. Dự án Kurento đã cung cấp các triển khai Kurento Client của API này chomột số nền tảng. Nếu bạn thích một ngôn ngữ lập trình khác với những ngôn ngữ được hỗ trợ, bạn có thể triển khai Kurento Client của riêng mình bằng cách sử dụng Giao thức Kurento, dựa trên WebSocket và JSON-RPC.

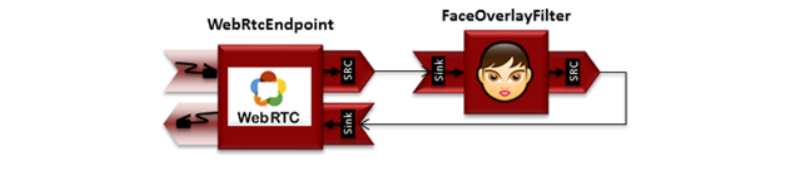
1. **Media Elements and Media Pipelines**

Kurento dựa trên hai khái niệm hoạt động như các khối xây dựng cho các nhà phát triển ứng dụng:

* **Media Elements:** là một đơn vị chức năng thực hiện một hành động cụ thể trên một luồng phương tiện. Media Elements là một cách mọi khả năng được biểu diễn dưới dạng một “hộp đen” (the Media Element) khép kín đối với nhà phát triển ứng dụng, người không cần hiểu các chi tiết cấp thấp của phần tử để sử dụng nó. Media Element có khả năng nhận phương tiện từ các phần tử khác (through media sources) và gửi phương tiện đến các yếu tố khác (through media sinks). Tùy thuộc vào chức năng của chúng, Media Element có thể được chia thành các các nhóm:
* **Input Endpoints:** Media Elements có khả năng nhận phương tiện và đưa nó vào đường ống. Có một số loại điểm cuối đầu vào. Điểm cuối đầu vào tệp lấy phương tiện từ tệp, Điểm cuối đầu vào mạng lấy phương tiện từ mạng và điểm cuối đầu vào Capture có khả năng nắm bắt luồng phương tiện trực tiếp từ máy ảnh hoặc loại tài nguyên phần cứng khác.
* **Filters:** Media Elements phụ trách chuyển đổi hoặc phân tích phương tiện. Do đó có các bộ lọc để thực hiện các hoạt động như trộn, trộn, phân tích, bổ sung, v.v.
* **Hubs:** Media Objects phụ trách quản lý nhiều luồng phương tiện trong một đường ống. Một Hub chứa một HubPort cho từng Media Elements được kết nối. Tùy thuộc vào loại Hub, có những cách khác nhau để điều khiển phương tiện.

Ví dụ: có một Hub có tên là Composite hợp nhất tất cả các đầu vào các luồng video trong một luồng video đầu ra duy nhất, với tất cả các đầu vào được sắp xếp trong một lưới.

* **Output Endpoints:** Media Elements có khả năng lấy luồng phương tiện ra khỏi Media Pineline. Lần nữa, có một số loại điểm cuối đầu ra, chuyên biệt về tệp, mạng, màn hình, v.v.
* **Media Pipeline:** là một chuỗi các Media Elements, trong đó luồng đầu ra được tạo bởi một nguồn phần tử được đưa vào một hoặc nhiều phần tử chìm. Do đó, đường ống đại diện cho một "đường ống" có khả năng thực hiện chuỗi hoạt động trên một luồng.



Điều này có nghĩa là nó dựa trên các Lớp có thể được khởi tạo dưới dạng Đối tượng, các Đối tượng này cung cấp các thuộc tính đại diện cho trạng thái bên trong của máy chủ Kurento và các phương thức hiển thị các hoạt động có thể được thực hiện bởi máy chủ.

1. **Endpoints**

**A WebRtcEndpoint:** là một điểm cuối đầu vào / đầu ra cung cấp truyền phát phương tiện cho Giao tiếp thời gian thực (RTC) thông qua web. Nó thực hiện công nghệ WebRTC để giao tiếp với các trình duyệt.

**An RtpEndpoint:** là một điểm cuối đầu vào / đầu ra cung cấp khả năng phân phối nội dung hai chiều với điều khiển từ xa mạng ngang hàng, thông qua giao thức RTP. Nó sử dụng SDP để đàm phán phương tiện.

**An HttpPostEndpoint:** là một điểm cuối đầu vào chấp nhận phương tiện sử dụng yêu cầu HTTP POST như tải lên tệp HTTP chức năng.

**A PlayerEndpoint**: là một điểm cuối đầu vào lấy nội dung từ hệ thống tệp, URL HTTP hoặc URL RTSP và đưa vào nó vào Media Pipeline.

**A RecorderEndpoint:** là một điểm cuối đầu ra cung cấp chức năng lưu trữ nội dung ở chế độ đáng tin cậy (không loại bỏ dữ liệu). Nó chứa các miếng đệm Media Sink cho âm thanh và video.

1. **Filters**

Filters: là các Media Elements thực hiện xử lý phương tiện, thị giác máy tính v.v.

**The ZbarFilter:** filters phát hiện QR và mã vạch trong luồng video. Khi một mã được tìm thấy, bộ lọc tăng CodeFoundEvent. Clients có thể thêm người nghe vào sự kiện này để thực hiện một số hành động.

**The FaceOverlayFilter:** filtersphát hiện các khuôn mặt trong luồng video và phủ lên nó bằng một hình ảnh có thể định cấu hình**.**

**GstreamerFilter:** là một giao diện bộ lọc chung cho phép đưa bất kỳ phần tử GStreamer nào vào Kurento Media Pipeline. Tuy nhiên, lưu ý rằng việc triển khai GStreamerFilter hiện tại chỉ cho phép các phần tử đơn lẻ được đưa vào, không thể chỉ ra nhiều hơn một cùng một lúc; sử dụng một số GStreamerFilters nếu bạn cần tiêm thêm nhiều hơn một phần tử cùng một lúc.

1. **Hubs**

Hub là đối tượng phương tiện chịu trách nhiệm quản lý nhiều luồng phương tiện trong một đường ống. Hub có một số cổng trung tâm, nơi các Media Elements khác được kết nối.

**Composite:** là một Hub trộn luồng âm thanh của các đầu vào được kết nối của nó và tạo lưới với các luồng video của họ.

**DispatcherOneToMany:** là một Hub gửi một đầu vào nhất định cho tất cả các HubPorts đầu ra được kết nối.

**Dispatcher:** là một Hub cho phép định tuyến giữa các cặp HubPort đầu vào-đầu ra tùy ý.

1. **Kurento Client**

Hiện tại, dự án Kurento cung cấp triển khai API Kurento cho hai ngôn ngữ lập trình: Java

và JavaScript. Trong tương lai, khách hàng Kurento bổ sung có thể được tạo, thể hiện cùng một loại mô-đun trong các ngôn ngữ khác như Python, C / C ++, PHP, v.v.

1. **Kurento Java Client**

Kurento Java Client là một lớp Java SE sử dụng API Kurento và thể hiện các khả năng của nó thông qua giao diện dễ sử dụng dựa trên Java POJO đại diện cho các Phần tử Phương tiện và Đường ống Phương tiện. API này trừu tượng theo nghĩa là tất cả những phức tạp vốn có không trực quan của Giao thức Kurento nội bộ hoạt động được trừu tượng hóa và các nhà phát triển không cần phải xử lý chúng khi tạo ứng dụng. Sử dụng Kurento Java Client chỉ yêu cầu thêm phần phụ thuộc thích hợp vào dự án Maven hoặc tải xuống jar tương ứng vào Java Classpath của ứng dụng. Điều quan trọng cần nhận xét là Kurento Java Client là một API điều khiển mặt phẳng phương tiện. Nói cách khác, mục tiêu của nó là cho thấy khả năng quản lý các đối tượng phương tiện, nhưng nó không cung cấp bất kỳ khả năng nào của mặt phẳng báo hiệu.

1. **Kurento JavaScript Client**

**Kurento JavaScript Client** là một lớp JavaScript sử dụng API Kurento và cho thấy các khả năng của nó. Nó cho phép xây dựng Node.js và các ứng dụng dựa trên trình duyệt.