**I.Giới thiệu về WebRTC**

**1.Sơ lược lịch sử của WebRTC**

Ý tưởng phát triển WebRTC được nhóm kỹ sư chịu trách nhiệm cho Google Hangouts đưa ra từ tận năm 2009. Vào thời gian đó, để truyền tải video, hình ảnh trên web thì người ta thường phải xài đến Flash. Nhóm kỹ sơ Hangouts lại không muốn sử dụng công nghệ này, và họ bắt đầu tự làm một chuẩn riêng cho mình. Đến năm 2010, Google thâu tóm hai công ty On2 và Global IP Solutions (GIPS) để lấy công nghệ truyền dữ liệu thời gian thực làm nền tảng cho WebRTC về sau.

Vào tháng 5/2011, Google ra mắt một dự án nguồn mở dành cho việc giao tiếp thời gian thực giữa trình duyệt với nhau, và từ lúc này dự án mang tên WebRTC. Song song đó, Hiệp hội World Wide Web (W3C) và Hiệp hội Kĩ sư quốc tế (IETF) cũng đang phát triển một số giao thức để dùng cho việc việc kết nối thời gian thực, thế nên họ bắt tay nhau tiếp tục hoàn thiện để rồi quyết định kết hợp chung vào WebRTC.

Đến 27/10/2011, W3C ra mắt bản nháp đầu tiên của WebRTC. Tháng 11/2011, Chrome 23 ra mắt, trở thành trình duyệt đầu tiên có tích hợp WebRTC ngay từ bên trong. Và tính đến thời mà mình viết bài này thì WebRTC vẫn còn đang tiếp tục được phát triển chứ chưa hoàn thiện một cách chính thức.

**2. WebRTC là gì?**

WebRTC là viết tắt của cụm từ Web Real-Time Communication. Là một web API được phát triển bởi World Wide Web Consortium (W3C), khả năng hỗ trợ trình duyệt (browser) giao tiếp với nhau thông qua VideoCall, VoiceCall hay transfer data Peer-to-Peer (P2P) mà không cần browser phải cài thêm plugins hay phần mềm hỗ trợ nào từ bên ngoài.

WebRTC là một dự án mở, miễn phí, cung cấp các trình duyệt và ứng dụng di động với khả năng Truyền thông thời gian thực (RTC) thông qua các API đơn giản. Các thành phần WebRTC đã được tối ưu hóa để phục vụ tốt nhất cho mục đích này.



**3.Các phần chính của WebRTC và chức năng của WebRTC API.**

Các phần chính của WebRTC bao gồm:

•getUserMedia, cho phép trình duyệt web truy cập vào camera và/hoặc microphone để lấy dữ liệu hình ảnh âm thanh cho việc truyền tải.

•RTCPeerConnection dùng để cài đặt videocall/voicecall dùng cho việc truyền tải.

•RTCDataChannel cho phép trình duyệt chia sẻ dữ liệu peer-to-peer.

WebRTC API bao gồm chức năng:

•getStats cho phép ứng dụng web lấy tập hợp các số liệu thống kê về các session WebRTC.

**4.WebRTC dùng để làm gì ?**

WebRTC có thể được sử dụng cho truyền tải video, âm thanh cho đến gửi dữ liệu theo thời gian thực giữa hai hoặc nhiều thiết bị với nhau mà không nhất thiết phải đi qua server trung gian.

**II.Giới thiệu về Kurento.**

**1. Kurento là gì?**

Kurento là một máy chủ truyền thông WebRTC và một bộ API khách hàng giúp đơn giản hóa việc phát triển các ứng dụng video nâng cao cho các nền tảng web và điện thoại thông minh. Các tính năng của nó bao gồm liên lạc nhóm, chuyển mã, ghi âm, trộn, phát sóng và định tuyến của dòng nghe nhìn. Kurento cung cấp một khung đa phương tiện giúp giảm bớt nhiệm vụ xây dựng các ứng dụng đa phương tiện với những tính năng, đặc điểm sau đây:

•Dynamic WebRTC Media pipelines: Kurento cho phép media pipelines tùy chỉnh được kết nối với các peers WebRTC như trình duyệt web và ứng dụng di động. Các media pipelines này dựa trên các yếu tố có thể kết hợp như người chơi, máy ghi âm, máy trộn, v.v … có thể được trộn và kết hợp, kích hoạt hoặc hủy kích hoạt tại bất kỳ thời điểm nào.

•Client/Server Architecture: Các ứng dụng được phát triển với Kurento tuân theo kiến trúc client/server. Kurento Media Server (KMS) là máy chủ và cung cấp giao diện WebSocket thực hiện giao thức Kurento, cho phép ứng dụng client để xác định cấu trúc liên kết pipelines.

•Java and JavaScript Client Applications: Trường hợp sử dụng điển hình của việc triển khai KMS bao gồm một trình duyệt kiến trúc, nơi trình duyệt người dùng tương tác với máy chủ KMS bằng Máy khách trung gian ứng dụng. Có một số thư viện Kurento chính thức, hỗ trợ việc sử dụng Java và JavaScript cho ứng dụng client. Các client cho các ngôn ngữ khác có thể dễ dàng thực hiện theo giao thức WebSocket.

•Third party Modules: Kurento Media Server có kiến trúc mở rộng dựa trên các plugin, cho phép các bên thứ ba để thực hiện các mô-đun có thể được thêm vào đường ống truyền thông của họ. Điều này cho phép tích hợp phương tiện truyền thông xử lý các thuật toán cho bất kỳ ứng dụng WebRTC nào, như tích hợp Thị giác máy tính, Công nghệ thực tế ảo, Video, và phân tích giọng nói. Tất cả cần thiết là tạo ra một yếu tố Kurento mới và sử dụng nó trong bất kỳ media pipelines hiện có.

**2. WebRTC media servers là gì?**

WebRTC là một tập hợp các giao thức, cơ chế và API cung cấp trình duyệt và ứng dụng di động với Real-Time Communications (RTC) thông qua kết nối peer-to-peer. Nó đã được hình thành như một công nghệ cho phép trình duyệt để giao tiếp trực tiếp mà không cần qua trung gian của bất kỳ loại cơ sở hạ tầng nào. Tuy nhiên, mô hình này chỉ đủ để tạo các ứng dụng web cơ bản; các tính năng như liên lạc nhóm, ghi dòng phương tiện, phương tiện truyền thông phát sóng, hoặc chuyển mã phương tiện rất khó thực hiện. Vì lý do này, nhiều ứng dụng yêu cầu một máy chủ phương tiện trung gian.

**Các WebRTC media servers phổ biến**

Một số media server đang được sử dụng khá phổ biến hiện tại có thể kể tên:

* Jitsi
* Kurento
* Janus