TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI DỘNG NÂNG CAO

TÌM HIỂU VÀ XÂY DỰNG GIẢI PHÁP CHO ĐỀ TÀI VIDEO CONFERENCE BẰNG FLUTTER

Thành viên:
21120140 – Trần Gia Thịnh
21120148 – Trần Tiến

Giảng viên hướng dẫn: Phạm Hoàng Hải

MỤC LỤC

MỤC LỤC BẢNG

MỤC LỤC HÌNH

I. Giới thiệu

1. Thành viên nhóm

Mã số sinh viên	Họ và tên
21120140	Trần Gia Thịnh
21120148	Trần Tiến

Bảng 1: Bảng thông tin thành viên

2. Mục tiêu

Khảo sát và đánh giá các giải pháp để xây dựng hệ thống Video Conference bằng Flutter.

Lựa chọn giải pháp và tìm hiểu sâu cách thức xây dựng hệ thống của giải pháp.

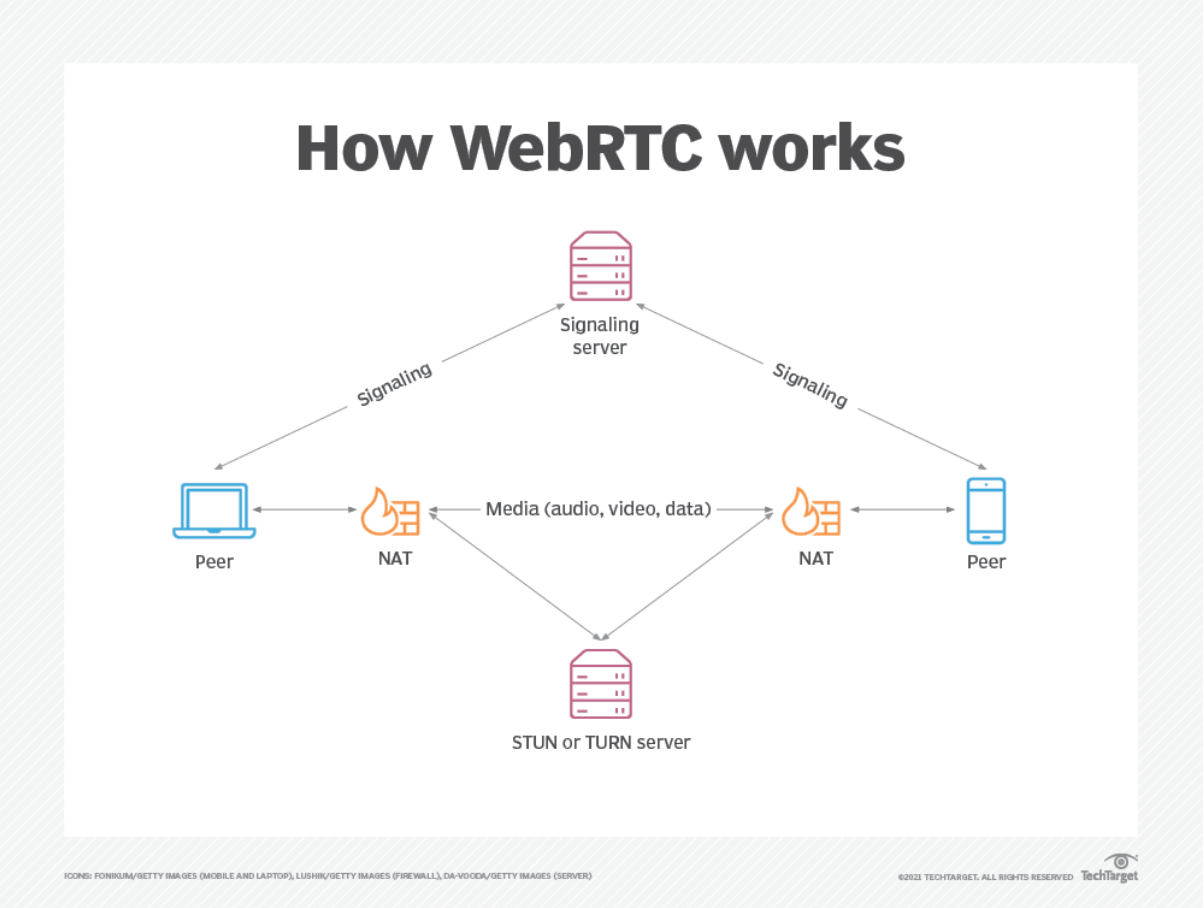
Xây dựng ứng dụng thử nghiệm dựa trên giải pháp đã chọn.

II. Nội dung tìm hiểu

1. Tổng quan về WebRTC

WebRTC (Web Real-Time Communication) là một công nghệ mã nguồn mở cho phép các trình duyệt và ứng dụng di động giao tiếp trực tiếp với nhau qua mạng P2P (peer-to-peer), hỗ trợ truyền tải âm thanh, video và dữ liệu theo thời gian thực mà không cần đến máy chủ trung gian.

- **Thành phần chính của WebRTC**
  - STUN Server: Xác định địa chỉ IP công khai của thiết bị để hỗ trợ kết nối trực tiếp.
  - TURN Server: Chuyển tiếp dữ liệu giữa các thiết bị khi kết nối trực tiếp thất bại, thường do NAT/firewall chặn.
  - Signaling Server: Trao đổi thông tin kết nối giữa các thiết bị (như SDP và ICE candidates) để thiết lập và duy trì kết nối.
- **Các bước hoạt động của WebRTC**
  - Signaling: Trao đổi thông tin để thiết lập kết nối (qua Signaling Server).
  - ICE Candidate Gathering: Tìm kiếm đường kết nối khả thi giữa các thiết bị (qua STUN/TURN Servers).
  - Kết nối P2P: Thiết lập kết nối P2P để truyền dữ liệu (nếu kết nối thất bại, TURN Server sẽ chuyển tiếp dữ liệu).



Hình 1: Kiến trúc WebRTC

2. Các giải pháp xây dựng hệ thống Video Conference

2.1. Build from Scratch

2.1.1. Tổng quan

Flutter cùng với thư viện flutter_webrtc, cho phép tích hợp WebRTC vào ứng dụng, cung cấp một nền tảng để xây dựng ứng dụng hội nghị trực tuyến. WebRTC hỗ trợ các cuộc gọi video, thoại và truyền dữ liệu P2P (peer-to-peer), nhưng không có sẵn kiến trúc hội nghị nhiều người (multiparty conferencing). Do đó, việc xây dựng một hệ thống hội nghị từ đầu đòi hỏi thiết kế và phát triển thêm các thành phần cần thiết.

2.1.2. Cách xây dựng hệ thống

- **Thiết lập Signaling Server**

  Signaling Server đóng vai trò trung gian trong việc trao đổi thông tin kết nối giữa các thiết bị tham gia cuộc gọi, bao gồm SDP (Session Description Protocol) và ICE (Interactive Connectivity Establishment) candidates. Các công nghệ có thể được sử dụng cho Signaling Server bao gồm WebSocket, Firebase hoặc giao thức Signaling WebRTC tùy chỉnh. Việc xây dựng

backend server hoặc sử dụng dịch vụ bên thứ ba để quản lý trao đổi thông tin này là rất quan trọng.

- **Sử dụng STUN/TURN Servers**

  STUN server giúp các thiết bị xác định địa chỉ IP công khai, cần thiết cho việc thiết lập kết nối P2P xuyên NAT (Network Address Translation). Trong khi đó, TURN server được sử dụng khi kết nối P2P không khả thi (do firewall/NAT), cho phép chuyển tiếp dữ liệu qua server. Các STUN server miễn phí như của Google có thể được sử dụng, hoặc có thể tự cấu hình TURN server như Coturn.

- **Cấu hình và Tích hợp flutter_webrtc**

  - **Cài Đặt Thư Viện:**

  Thêm thư viện flutter_webrtc vào dự án Flutter của bạn thông qua tệp pubspec.yaml.

  Cấu Hình STUN/TURN Servers:

  Định nghĩa STUN và TURN servers để thiết lập kết nối P2P, giúp nhận diện địa chỉ IP công khai và xử lý trường hợp kết nối không thành công.

  - **Tạo RTCVideoRenderer:**

  Sử dụng RTCVideoRenderer để hiển thị video từ cả hai phía (local và remote), quản lý tài nguyên video trong lifecycle của ứng dụng.

  Thiết lập Kết Nối với RTCPeerConnection:

  Tạo RTCPeerConnection để thiết lập kết nối giữa các người dùng và thêm video stream từ camera local.

  - **Gửi và Nhận SDP và ICE Candidates:**

  Sử dụng Signaling Server để gửi và nhận SDP và ICE candidates, điều này cần thiết để thiết lập kết nối P2P.

  - **Quản Lý Nhiều Luồng Video:**

  Tạo và quản lý nhiều RTCPeerConnection cho các người dùng trong cuộc gọi nhóm, cho phép mở rộng khả năng hội nghị

  Thiết kế Kiến Trúc Gọi Nhóm (Multiparty Video Conference)

  Mesh Network: Mỗi người tham gia kết nối trực tiếp với tất cả các người khác, dễ triển khai nhưng không tối ưu cho các cuộc gọi lớn vì băng thông yêu cầu cao.

Sử dụng Media Server để thu thập và phân phối luồng video, chỉ cần một kết nối cho mỗi người dùng, giúp tối ưu hóa băng thông cho các cuộc gọi lớn.

- **Tạo Giao Diện Người Dùng:**

Thiết kế giao diện người dùng trực quan và thân thiện, cho phép người dùng dễ dàng tham gia các cuộc gọi video.

2.1.3. Ưu và Nhược Điểm

- **Ưu điểm**

Toàn Quyền Kiểm Soát: Xây dựng từ đầu cho phép tùy chỉnh giao diện, kiến trúc backend và logic gọi nhóm theo ý muốn.

Tùy Chỉnh và Mở Rộng Linh Hoạt: Có thể dễ dàng tích hợp các tính năng như chia sẻ màn hình, ghi âm/ghi hình, nhắn tin trong cuộc gọi và các tùy chọn bảo mật.

- **Nhược điểm**

Phức Tạp và Tốn Thời Gian: Việc xây dựng từ đầu yêu cầu kiến thức về WebRTC, kiến trúc P2P và quản lý băng thông, là công việc đòi hỏi kỹ thuật cao và thời gian để tối ưu.

Yêu Cầu Media Server cho Các Cuộc Gọi Lớn: Đối với các cuộc gọi có nhiều người tham gia, việc sử dụng Media Server là bắt buộc để tránh vấn đề về băng thông, làm tăng chi phí và phức tạp.

Khó Khăn trong Bảo Trì và Mở Rộng: Việc tự xây dựng và quản lý mọi thành phần sẽ đòi hỏi nguồn lực và công sức lớn để duy trì backend, Media Server và signaling.

2.2. Open Source Solutions:

2.2.1. Tổng quan

Để xây dựng một hệ thống hội nghị truyền hình trên Flutter, có một số nền tảng mã nguồn mở đáng chú ý như Jitsi Meet, OpenVidu. Mỗi giải pháp này đều có ưu và nhược điểm riêng, cùng các công cụ khác nhau hỗ trợ việc truyền tải video, âm thanh và dữ liệu trong thời gian thực. Dưới đây là phân tích chi tiết về các giải pháp mã nguồn mở phổ biến.

2.2.2. Jitsi Meet

Jitsi Meet là một giải pháp hội nghị truyền hình mã nguồn mở, nổi bật với tính dễ dàng tích hợp, giao diện đơn giản và khả năng hoạt động trên cả web và ứng

dụng di động. Được phát triển bởi Jitsi, Jitsi Meet cung cấp tính năng hội nghị video ổn định và cho phép triển khai máy chủ riêng để tăng cường bảo mật.

- **Công nghệ nền tảng**: WebRTC cho truyền tải video, âm thanh thời gian thực.
- **Khả năng mở rộng:** Có thể triển khai nhiều máy chủ Jitsi Video Bridge để chia tải.
- **Ưu điểm**

  Dễ tích hợp và tùy chỉnh: Cung cấp nhiều API và SDK, dễ dàng tích hợp với Flutter.

  Bảo mật cao: Có thể thiết lập máy chủ riêng và sử dụng mã hóa end-to-end.

  Tính năng phong phú: Hỗ trợ tính năng phòng họp, chia sẻ màn hình, chat, và ghi âm.
- **Nhược điểm**

  Hạn chế hiệu năng với nhiều người dùng: Với hội nghị có số lượng người tham gia lớn, có thể gặp vấn đề về hiệu suất.

  Yêu cầu băng thông cao: Cần băng thông ổn định để đảm bảo chất lượng hội nghị, đặc biệt là với nhiều luồng video.

2.2.3. OpenVidu

OpenVidu là một nền tảng mã nguồn mở dựa trên WebRTC, cho phép phát triển và triển khai các ứng dụng hội nghị video, tương tác thời gian thực, và có thể tùy chỉnh linh hoạt theo nhu cầu người dùng.

- **Công nghệ nền tảng:** WebRTC và hỗ trợ bởi OpenVidu Server.
- **Khả năng mở rộng:** Hỗ trợ mở rộng hệ thống thông qua Kubernetes.
- **Ưu điểm**

  Tùy chỉnh cao: hỗ trợ nhiều tùy chỉnh cho luồng video và điều khiển chất lượng video/audio theo nhu cầu.

  Đa nền tảng: tương thích tốt với các nền tảng di động, web và cả các thiết bị IoT.

  Công cụ quản lý mạnh mẽ: Cho phép cấu hình, quản lý và theo dõi các phiên hội nghị video một cách linh hoạt.
- **Nhược điểm**

  Phức tạp trong triển khai: Yêu cầu kiến thức hệ thống và triển khai phức tạp nếu sử dụng các chức năng nâng cao.

Tài nguyên máy chủ yêu cầu lớn: Đòi hỏi tài nguyên máy chủ cao khi có số lượng người tham gia lớn.

2.2.4. Cách xây dựng hệ thống

Để xây dựng hệ thống hội nghị truyền hình sử dụng Flutter và các nền tảng mã nguồn mở, quá trình triển khai thường bao gồm các bước sau:

- **Tích hợp API/SDK:** Sử dụng các plugin hoặc SDK được cung cấp bởi các nền tảng như jitsi_meet, openvidu_flutter, hoặc API của Daily.co.
- **Cấu hình máy chủ:** Với các nền tảng như Jitsi Meet hoặc OpenVidu, cần thiết lập máy chủ riêng để tăng bảo mật và kiểm soát hệ thống.
- **Tùy chỉnh giao diện người dùng:** Thiết kế giao diện hội nghị video trong Flutter, cho phép người dùng tham gia phòng họp, bật/tắt video và âm thanh, và sử dụng các tính năng bổ sung như chia sẻ màn hình.
- **Quản lý và tối ưu hóa băng thông**: Thiết lập các thông số để tối ưu chất lượng video, giảm thiểu độ trễ và đảm bảo ổn định cho số lượng người tham gia lớn.

2.2.5. Ưu và Nhược điểm

- **Ưu điểm**

  Tiết Kiệm Chi Phí Phát Triển: Sử dụng mã nguồn mở giúp giảm chi phí phát triển, không cần xây dựng từ đầu mà có thể triển khai dựa trên các nền tảng sẵn có như Jitsi Meet, BigBlueButton, hoặc OpenVidu.

  Tích hợp nhanh chóng: Các giải pháp này thường đã bao gồm những tính năng cơ bản như gọi video, chia sẻ màn hình, nhắn tin, và ghi âm/ghi hình, giúp tích hợp vào hệ thống dễ dàng và nhanh chóng hơn.

  Cộng đồng và tài liệu hỗ trợ: Các nền tảng mã nguồn mở thường có cộng đồng lớn và tài liệu phong phú, hỗ trợ nhanh chóng khi gặp sự cố hoặc cần tùy chỉnh.

  Khả năng mở rộng: Nhiều giải pháp mã nguồn mở có kiến trúc linh hoạt, cho phép mở rộng tính năng hoặc tùy chỉnh theo yêu cầu.

- **Nhược điểm**

  Giới hạn tùy chỉnh: Dù mã nguồn mở cho phép tùy chỉnh, nhưng giới hạn sẽ nằm ở kiến trúc có sẵn của nền tảng. Các thay đổi lớn có thể phức tạp và tốn kém công sức, đặc biệt nếu không đồng bộ với bản cập nhật mới từ cộng đồng.

Hiệu năng và tối ưu hóa: Các nền tảng mã nguồn mở không phải lúc nào cũng được tối ưu tốt nhất cho tất cả các trường hợp sử dụng, và có thể yêu cầu điều chỉnh bổ sung nếu muốn đạt hiệu suất cao.

Phụ thuộc vào cộng đồng và bản cập nhật: Sự hỗ trợ và các bản vá lỗi phụ thuộc vào cộng đồng và tổ chức phát triển mã nguồn mở, nên có thể gặp khó khăn nếu các tính năng cần thiết không có sẵn.

Chi phí cho Media Server: Dù là mã nguồn mở, hệ thống vẫn cần các dịch vụ STUN/TURN và Media Server khi quy mô lớn, nên vẫn tốn chi phí hạ tầng.

2.3. Paid Solutions:

2.3.1. Tổng quan

Đây là giải pháp sử dụng các nền tảng có phí và dịch vụ từ các nhà cung cấp. Các nền tảng này, bao gồm Zoom SDK, Twilio Video, Agora, và Vonage (TokBox), cung cấp API và SDK mạnh mẽ để tạo trải nghiệm hội nghị video chất lượng cao, với hỗ trợ đa dạng các tính năng. Các nền tảng trả phí thường đi kèm với dịch vụ hỗ trợ kỹ thuật và tài liệu chi tiết giúp quá trình tích hợp trở nên thuận tiện hơn.

2.3.2. Zoom SDK

Zoom SDK cung cấp giải pháp hội nghị truyền hình chất lượng cao, với tính năng phong phú và bảo mật nâng cao, giúp tích hợp trải nghiệm Zoom vào các ứng dụng di động và web.

- **Công nghệ nền tảng:** Zoom sử dụng WebRTC kết hợp với hệ thống backend chuyên biệt để đảm bảo chất lượng và bảo mật.
- **Khả năng mở rộng:** Zoom có hệ thống cloud mạnh mẽ cho phép hỗ trợ số lượng người dùng lớn mà không cần thiết lập server riêng.
- **Ưu điểm**

  Chất lượng và độ tin cậy cao: Zoom được biết đến với khả năng duy trì chất lượng video và âm thanh ổn định, kể cả trong điều kiện mạng không lý tưởng.

  Hỗ trợ tính năng phong phú: Bao gồm chia sẻ màn hình, ghi âm, breakout room, và quản lý người tham gia.

  Bảo mật mạnh mẽ: Zoom có mã hóa end-to-end và các tính năng kiểm soát quyền truy cập tốt.
- **Nhược điểm**

  Chi phí cao: Zoom có mức phí cao hơn so với các dịch vụ khác, đặc biệt với

các tính năng nâng cao và người tham gia đông.

Phụ thuộc vào Zoom Cloud: Các tính năng và hiệu suất phụ thuộc vào cơ sở hạ tầng của Zoom, không cho phép tùy chỉnh server.

2.3.3. Twilio Video

Twilio Video là một dịch vụ hội nghị truyền hình linh hoạt với API mạnh mẽ, cho phép tùy chỉnh cao và triển khai các giải pháp hội nghị video từ nhỏ đến lớn.

- **Công nghệ nền tảng:** WebRTC.
- **Khả năng mở rộng:** Twilio hỗ trợ mở rộng quy mô linh hoạt qua hệ thống cloud của họ.
- **Ưu điểm**

  API linh hoạt và tùy chỉnh cao: Twilio cho phép kiểm soát chi tiết giao diện và trải nghiệm người dùng.

  Hỗ trợ nhiều tính năng nâng cao: Bao gồm phát hiện mặt, chia sẻ màn hình, ghi âm và điều khiển chất lượng video/audio tự động.

  Khả năng kiểm soát băng thông thông minh: Tự động điều chỉnh chất lượng dựa trên điều kiện mạng.
- **Nhược điểm**

  Chi phí theo sử dụng: Twilio Video tính phí theo thời lượng sử dụng, dễ dẫn đến chi phí cao cho các ứng dụng có lượng sử dụng lớn.

  Yêu cầu thiết lập phức tạp hơn: Twilio có tài liệu kỹ thuật chi tiết, nhưng yêu cầu người dùng có kiến thức tốt về hệ thống WebRTC và lập trình video.

2.3.4. Agora

Agora là nền tảng video call và streaming chuyên dụng với khả năng mở rộng linh hoạt và hiệu suất tốt, đặc biệt phù hợp cho các ứng dụng yêu cầu độ trễ thấp như phát sóng trực tiếp.

- **Công nghệ nền tảng:** WebRTC, với nhiều cải tiến để giảm độ trễ và tối ưu băng thông.
- **Khả năng mở rộng:** Agora có hạ tầng cloud rộng khắp toàn cầu, cho phép mở rộng hiệu quả.
- **Ưu điểm**

  Chất lượng video cao và độ trễ thấp: Agora cung cấp các giải pháp với độ trễ cực thấp, đặc biệt hiệu quả cho các sự kiện livestream hoặc video call đa

phương tiện.

Dễ tích hợp: Agora có sẵn plugin cho Flutter và tài liệu hướng dẫn chi tiết.

Hỗ trợ đa nền tảng: Agora hoạt động tốt trên cả Android, iOS, và web.

- **Nhược điểm**

Chi phí theo mức độ sử dụng: Chi phí có thể cao nếu sử dụng tính năng cao cấp và yêu cầu băng thông lớn.

Tùy chỉnh giao diện hạn chế hơn: Một số hạn chế khi tuỳ chỉnh giao diện của Agora SDK so với các nền tảng khác như Twilio.

2.3.5. Cách xây dựng hệ thống

- **Tích hợp API/SDK:** Sử dụng các SDK của từng nền tảng, như Zoom SDK, Twilio Video SDK, Agora Flutter SDK, để tích hợp vào ứng dụng Flutter.
- **Cấu hình ứng dụng:** Cấu hình các thông số API, như thông tin phòng họp, bảo mật, và quản lý người dùng.
- **Thiết lập tính năng hội nghị video:** Sử dụng các API để tạo phòng họp, bật/tắt video và âm thanh, chia sẻ màn hình và chat.
- **Quản lý băng thông:** Đối với các nền tảng như Twilio và Agora, có thể điều chỉnh chất lượng video/audio để tối ưu băng thông.
- **Đảm bảo bảo mật:** Thiết lập các chính sách bảo mật nâng cao như mã hóa end-to-end và xác thực người dùng nếu cần.

2.3.6. Ưu và Nhược điểm

- **Ưu điểm**

Dễ dàng triển khai và tích hợp: Các nền tảng trả phí cung cấp SDK và API với tài liệu đầy đủ, dễ sử dụng, giúp tích hợp nhanh chóng vào ứng dụng mà không cần chuyên sâu về WebRTC hay các công nghệ nền tảng.

Hiệu năng và ổn định cao: Các giải pháp này thường đã được tối ưu cho hiệu năng cao và độ ổn định, đảm bảo chất lượng video và âm thanh tốt ngay cả khi số lượng người dùng lớn.

Công nghệ hiện đại và thường xuyên cập nhật: Các dịch vụ trả phí được cập nhật liên tục với các tính năng mới nhất, như tối ưu băng thông, AI nâng cao chất lượng hình ảnh và âm thanh, các tính năng bảo mật bổ sung, v.v.

Hỗ trợ kĩ thuật và SLA: Các nền tảng trả phí thường cung cấp hỗ trợ kỹ thuật và cam kết chất lượng dịch vụ (SLA), giúp giải quyết sự cố nhanh chóng,

đảm bảo trải nghiệm người dùng.

Bảo mật mạnh mẽ: Các nền tảng này đầu tư vào bảo mật với các tính năng như mã hóa đầu cuối, xác thực, quản lý quyền truy cập để bảo vệ thông tin người dùng tốt hơn.

- **Nhược điểm**

Chi phí cao: Các giải pháp trả phí tính chi phí theo mức sử dụng hoặc số lượng người dùng, có thể tốn kém khi lượng người dùng tăng cao hoặc cần tính năng nâng cao như ghi âm, phiên dịch thời gian thực.

Giới hạn tùy chỉnh và kiểm soát: Mặc dù có nhiều tính năng, các giải pháp trả phí thường giới hạn về khả năng tùy chỉnh, không cho phép thay đổi toàn bộ cấu trúc nền tảng hay giao diện theo ý muốn.

Phụ thuộc vào bên cung cấp dịch vụ: Nếu nhà cung cấp gặp sự cố kỹ thuật, hệ thống của bạn có thể bị ảnh hưởng. Ngoài ra, thay đổi về giá cả hoặc chính sách của bên thứ ba cũng có thể tác động đến dịch vụ của bạn.

Bảo mật phụ thuộc vào nhà cung cấp: Các biện pháp bảo mật do nhà cung cấp triển khai, nên bạn có ít khả năng kiểm soát bảo mật ở mức độ sâu hơn hoặc tùy chỉnh theo quy chuẩn nội bộ.

2.4. So sánh giữa các giải pháp

Tiêu chí	Build from scratch	Open sources	Paid solution
Chi Phí Phát Triển	Cao (cần phát triển hoàn toàn)	Thấp hơn, tận dụng nền tảng có sẵn	Cao (theo mức sử dụng và tính năng)
Tùy Chỉnh	Tối đa, hoàn toàn theo yêu cầu	Giới hạn theo kiến trúc sẵn có	Giới hạn, không cho phép tùy chỉnh sâu
Triển Khai	Phức tạp và mất thời gian	Dễ dàng và nhanh chóng	Dễ dàng với SDK và tài liệu đầy đủ
Hỗ Trợ Kỹ Thuật	Không có hỗ trợ chính thức, tự duy trì	Hỗ trợ từ cộng đồng và tài liệu	Hỗ trợ kỹ thuật và SLA từ nhà cung cấp
Bảo Mật	Tùy thuộc vào khả	Phụ thuộc vào cộng đồng và	Mạnh mẽ, với các tính năng bảo mật

	năng tự xây dựng và quản lý	tính năng có sẵn	nâng cao
Hiệu Năng	Có thể tối ưu hóa, nhưng yêu cầu kiến thức chuyên sâu	Không tối ưu hóa cho tất cả trường hợp sử dụng	Thường tối ưu và ổn định cho hiệu suất cao
Khả Năng Mở Rộng	Có thể mở rộng nhưng đòi hỏi nhiều nguồn lực	Có thể mở rộng, nhưng phụ thuộc vào kiến trúc nền tảng	Dễ dàng mở rộng với nhiều tính năng mới
Phụ Thuộc	Không phụ thuộc vào bên thứ ba	Phụ thuộc vào cộng đồng và bản cập nhật	Phụ thuộc vào nhà cung cấp, có thể ảnh hưởng đến dịch vụ

Bảng 2: Bảng so sánh giữa các giải pháp

2.5. Kết luận

- **Giải pháp xây dựng từ đầu** phù hợp cho những ai cần kiểm soát hoàn toàn và có thể đầu tư thời gian và nguồn lực.
- **Giải pháp mã nguồn mở** là lựa chọn tốt cho những ai muốn tiết kiệm chi phí và thời gian nhưng không cần tùy chỉnh quá phức tạp.
- **Giải pháp trả phí** thích hợp cho các tổ chức cần nhanh chóng triển khai một hệ thống ổn định và có hỗ trợ kỹ thuật chuyên nghiệp.

3. Lựa chọn giải pháp và cách thức xây dựng hệ thống

Nhóm quyết định chọn giải pháp sử dụng mã nguồn mở cụ thể là Jitsi Meet. Bởi vì Đây là nền tảng miễn phí phù hợp với kinh phí của nhóm.

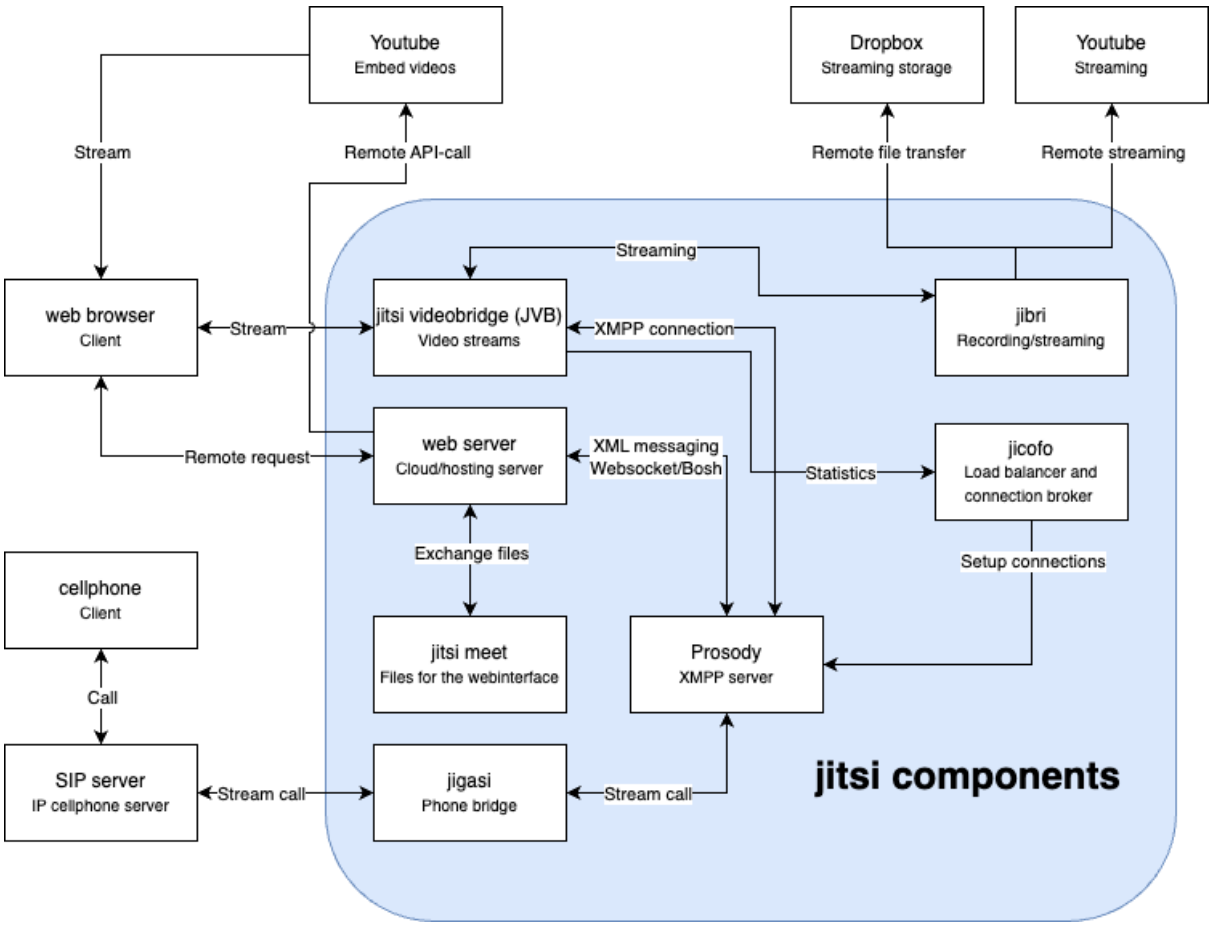
Khả năng tích hợp nhanh chóng, không tốn nhiều thời gian triển khai như xây dựng hệ thống từ đầu.

Dưới đây là phân tích kỹ thuật chi tiết về các thành phần chính, cách tích hợp với Flutter, các API và thư viện hỗ trợ, cũng như các hạn chế và yêu cầu về băng thông, bảo mật, và khả năng mở rộng.

3.1. Kiến trúc hệ thống của Jitsi Meet

Jitsi Meet sử dụng công nghệ WebRTC để truyền tải âm thanh, video và dữ liệu qua giao thức web, cùng các thành phần chính sau:

- **Prosody (Signaling Server)**: Đảm nhiệm vai trò signaling, trao đổi thông tin kết nối ban đầu giữa các thiết bị, và quản lý người dùng thông qua XMPP (Extensible Messaging and Presence Protocol).
- **Jitsi Video Bridge (JVB)**: Thành phần xử lý chính của Jitsi Meet, JVB nhận và chuyển tiếp các luồng dữ liệu video và âm thanh đến từng người tham gia trong phòng. Để tối ưu hiệu suất, JVB hoạt động như một SFU (Selective Forwarding Unit), giúp các luồng video không phải mã hóa lại mà chỉ cần chuyển tiếp đến các thiết bị đầu cuối.
- **Jicofo**: Điều phối viên của hội nghị, có vai trò kết nối, đồng bộ hóa các thiết bị, và điều phối các luồng dữ liệu trong phòng họp.
- **Jitsi Meet API**: Cung cấp các giao diện lập trình ứng dụng (API) để tích hợp Jitsi Meet vào ứng dụng web và di động, giúp khởi tạo cuộc họp, quản lý thông tin người dùng, và xử lý sự kiện hội nghị.



Hình 2: Kiến trúc Jitsi

3.2. WebRTC trong Jitsi Meet

WebRTC là nền tảng cốt lõi giúp Jitsi Meet truyền tải luồng dữ liệu một cách an toàn và thời gian thực giữa các thiết bị. WebRTC cung cấp các chức năng sau cho Jitsi Meet:

- **ICE (Interactive Connectivity Establishment)**: Tìm kiếm con đường tốt nhất kết nối các thiết bị qua mạng, giúp đảm bảo kết nối ổn định.
- **DTLS-SRTP (Datagram Transport Layer Security - Secure Real-time Transport Protocol)**: Mã hóa dữ liệu để đảm bảo tính bảo mật.
- **STUN/TURN Servers**: Hỗ trợ vượt qua NAT và firewall, đảm bảo các thiết bị có thể kết nối với nhau trong các môi trường mạng khác nhau.

3.3. Flutter và SDK Jitsi_meet

SDK Jitsi Meet có thể được tích hợp vào ứng dụng Flutter để triển khai các tính năng hội nghị video. SDK này cung cấp khả năng điều khiển các cuộc họp từ ứng dụng Flutter, khởi tạo cuộc họp, theo dõi sự kiện và điều khiển các tùy chọn âm thanh, video.

3.4. Cách thức sử dụng API và thư viện

3.4.1. Cài đặt và cấu hình plugin jitsi_meet trong Flutter

Để bắt đầu sử dụng SDK Jitsi Meet, cần thêm thư viện vào tệp pubspec.yaml:

```
dependencies:
jitsi_meet_flutter_sdk: ^latest_version
jitsi_meet_flutter_sdk: ^10.2.1
```

Sau đó, tải thư viện qua lệnh sau:

```
flutter pub get
```

3.4.2. Tạo giao diện hội nghị video trong Flutter

Sử dụng thư viện jitsi_meet, bạn có thể tạo giao diện và các chức năng cần thiết cho hội nghị video như tham gia phòng họp, tắt/mở âm thanh và video.

Ví dụ: Khởi tạo một phiên hội nghị

```
var options = JitsiMeetConferenceOptions(
serverURL: "https://meet.jit.si",
room: roomName, // Room identifier
configOverrides: {
```

```
    "startWithAudioMuted": true,
    "startWithVideoMuted": true,
    "subject": meetingName, // Use the entered meeting name as the subject
    "localSubject": "Local - $meetingName",
   },
   featureFlags: {
    "unsaferoomwarning.enabled": false,
    "security-options.enabled": false
   },
   userInfo: JitsiMeetUserInfo(
      displayName: "Flutter user", email: "user@example.com"),
  );
  jitsiMeet.join(options);
```

3.4.3. Cấu hình máy chủ Jitsi Meet tùy chỉnh

- var options = JitsiMeetConferenceOptions(
  serverURL: "https://your-custom-server.com",
  room: 'flutter_meeting_room',);

3.4.4. Các thông số tùy chọn của JitsiMeetingOptions

- **serverURL**: URL của máy chủ Jitsi Meet mà ứng dụng sẽ kết nối đến. Ví dụ, "https://meet.jit.si" là máy chủ công khai của Jitsi.
- **room**: Tên phòng họp (Room identifier). Đây là chuỗi xác định phòng họp cụ thể mà người dùng sẽ tham gia.
- **configOverrides**: Cấu hình bổ sung để tùy chỉnh hành vi của cuộc họp. Các thông số trong configOverrides bao gồm:
    - **startWithAudioMuted**: Bắt đầu cuộc họp với âm thanh bị tắt (true để tắt âm thanh).
    - **startWithVideoMuted**: Bắt đầu cuộc họp với video bị tắt (true để tắt video).
    - **subject**: Chủ đề (tên) của cuộc họp.
    - **localSubject**: Chủ đề cục bộ hiển thị (ví dụ: "Local - $meetingName").
- **featureFlags**: Tùy chọn bật/tắt các tính năng nhất định trong cuộc họp. Trong

đoạn mã này:

  - **unsaferoomwarning.enabled**: Vô hiệu hóa cảnh báo về phòng không an toàn (khi đặt là false).
  - **security-options.enabled**: Vô hiệu hóa các tùy chọn bảo mật (khi đặt là false).

- **userInfo**: Thông tin người dùng tham gia cuộc họp, bao gồm:
  - **displayName**: Tên hiển thị của người dùng trong cuộc họp (ví dụ: "Flutter user").
  - **email**: Địa chỉ email của người dùng (ví dụ: "user@example.com")

3.5. Hạn chế kỹ thuật

- **Phụ thuộc vào hiệu năng thiết bị**

  Jitsi Video Bridge cần xử lý lượng lớn dữ liệu khi thực hiện hội nghị video trên nhiều thiết bị. Nếu thiết bị hoặc mạng không đủ mạnh, có thể dẫn đến gián đoạn.

- **Không có tính năng AI nâng cao**

  Jitsi Meet không tích hợp sẵn các tính năng AI như lọc tiếng ồn, nhận diện khuôn mặt, hoặc làm mờ nền. Các tính năng này có thể cần thiết cho một số ứng dụng chuyên nghiệp và có thể cần bổ sung công nghệ khác như Google AI hoặc AWS Machine Learning để hỗ trợ.

- **Hạn chế về bảo mật**

  Khi sử dụng SDK Jitsi, việc đảm bảo quyền riêng tư và bảo mật cho người dùng là một vấn đề cần chú ý. Đảm bảo rằng quyền truy cập vào camera, micro, và dữ liệu người dùng được xử lý đúng cách là rất quan trọng.

- **Khả năng mở rộng**

  Jitsi Meet và các plugin liên quan thường xuyên được cập nhật, dẫn đến việc ứng dụng cần phải điều chỉnh theo để duy trì tính ổn định và khả năng tương thích.

4. Xây dựng ứng dụng thử nghiệm dựa trên giải pháp đã chọn.

Video demo:
https://drive.google.com/file/d/1LhAH3WxipfVGbkUxUfx6CWG5j_aQ8fmU/view?usp=drive_link

TÀI LIỆU THAM KHẢO

[1] "WebRTC (Web Real-Time Communications)," [Online]. Available: https://www.techtarget.com/searchunifiedcommunications/definition/WebRTC-Web-Real-Time-Communications.

[2] "Flutter-WebRTC: A Complete Guide," [Online]. Available: https://www.zegocloud.com/blog/flutter-webrtc.

[3] "Jitsi Meet Handbook," [Online]. Available: https://jitsi.github.io/handbook/docs/intro.

[4] "Zoom Video SDK for Flutter," [Online]. Available: https://developers.zoom.us/docs/video-sdk/flutter/.

[5] "OpenVidu Docs," [Online]. Available: https://openvidu.io/latest/docs/getting-started/.

[6] "BigBlueButton Development Guide," [Online]. Available: https://docs.bigbluebutton.org/development/guide/.

[7] "Twilio Flutter," [Online]. Available: https://pub.dev/documentation/twilio_flutter/latest/.