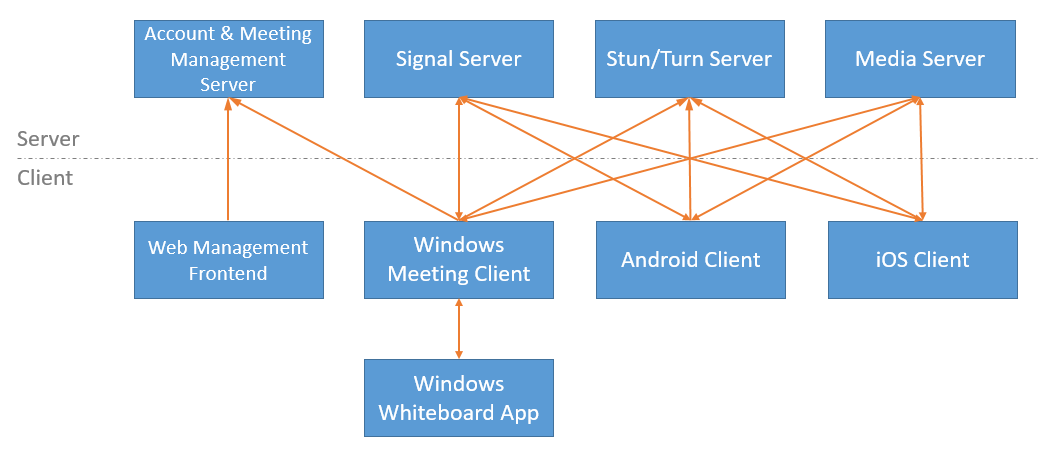
**dynaConnect Meeting Technical Design**

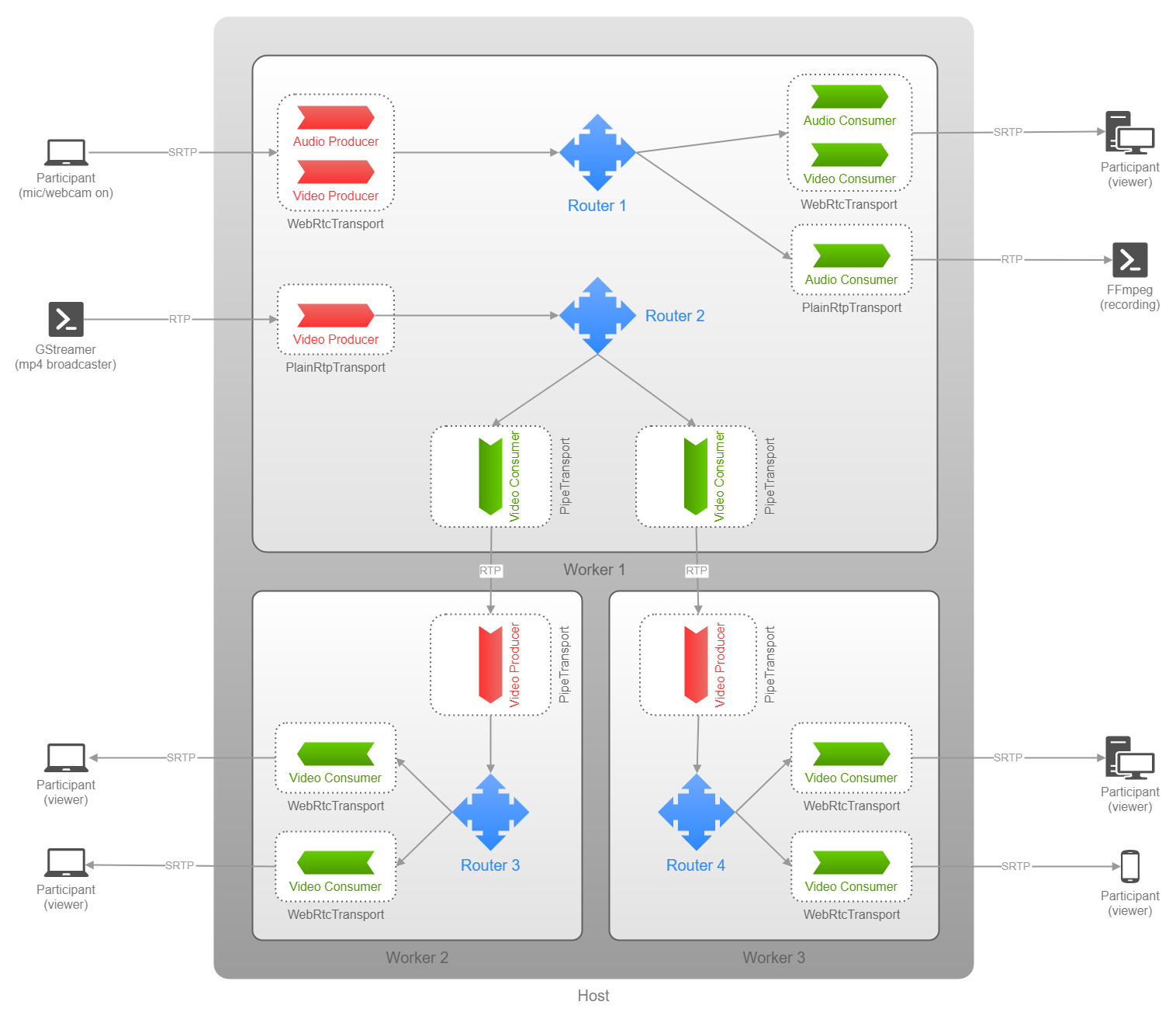
1. **System Architecture**



1. **Component List**

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Framework | Language | Description |
| Account & Meeting Management Server | SpringBoot, Nginx | Java | Manage user account, meeting arrangement and statistics. |
| Signal Server | Node.js, socket.io | JavaScript | Manage meeting rooms and assist WebRTC p2p connection, also support to sync whiteboard data among clients.  <https://codelabs.developers.google.com/codelabs/webrtc-web/>  <https://www.tutorialspoint.com/webrtc/webrtc_signaling.htm>  <https://www.wowza.com/blog/webrtc-signaling-servers> |
| Stun/Turn Server | coturn | C | Support ICE and NAT traversal.  <https://github.com/coturn/coturn>  <https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Protocols>  <https://blog.ivrpowers.com/post/technologies/what-is-stun-turn-server/> |
| Media Server | MediaSoup | C++, JavaScript | SFU media server for WebRTC multipart calls.  <https://mediasoup.org/>  <https://trueconf.com/blog/wiki/sfu>  <https://antmedia.io/webrtc-servers/> |
| Web Management Frontend | React | JavaScript | Browser UI for account, meeting and server management. |
| Windows Meeting Client | Electron, WebRTC | JavaScript | Video conference client, also support screen share. |
| Windows Whiteboard App | WPF | C# | Support Office document share and annotation collaboration. |
| Android Client | React Native, WebRTC | JavaScript | Video conference and whiteboard client, limited functions. |
| iOS Client | React Native, WebRTC | JavaScript | Video conference and whiteboard client, limited functions. |

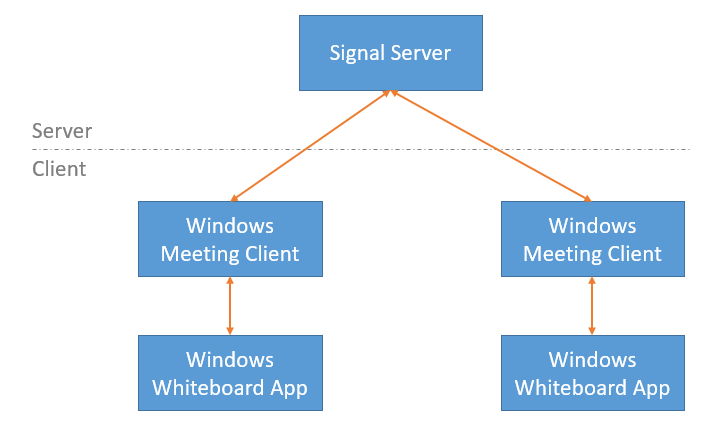
1. **MediaSoup Architecture**



Note: The Windows Meeting Client, Android Client and iOS Client are all participants in the image above.

1. **IPC between Windows Meeting client and Whiteboard app**

Windows Whiteboard app is a standalone WPF app, in order to utilize Windows Ink ability, since the Meeting client has the user accounts information and connections, so the Whiteboard app needs to ask the Meeting client to transfer data with other participants. The data for IPC between Whiteboard app and Meeting client contains document path (actually images path) and strokes, packaged with Protocol Buffers format, transferring with socket on TCP, these data will be forwarded to Signal server and broadcasted to other participants in secure environment (HTTPS), when the Meeting client in another participant receive the whiteboard data, it will forward to Whiteboard app with socket.



1. **Protocols**

|  |  |  |
| --- | --- | --- |
| Protocol | Component | Data to Transfer |
| HTTPS | Web frontend / All platforms meeting client / Signal server 🡨🡪 Account & meeting management server  All platforms meeting client 🡨🡪 Signal server | Account and password for login.  History and reserved meetings.  Whiteboard documents and strokes.  Server configurations.  Media server usage statistics. |
| DTLS-SRTP | All platforms meeting client 🡨🡪 Media server | Mic audio and camera video data.  Screen share video data. |

1. **Codecs**

|  |  |  |
| --- | --- | --- |
| Codec | Component | Usage |
| VP8 | All platforms meeting client | Camera and screen share video encoding & decoding. |
| Opus | All platforms meeting client | Mic audio encoding & decoding. |