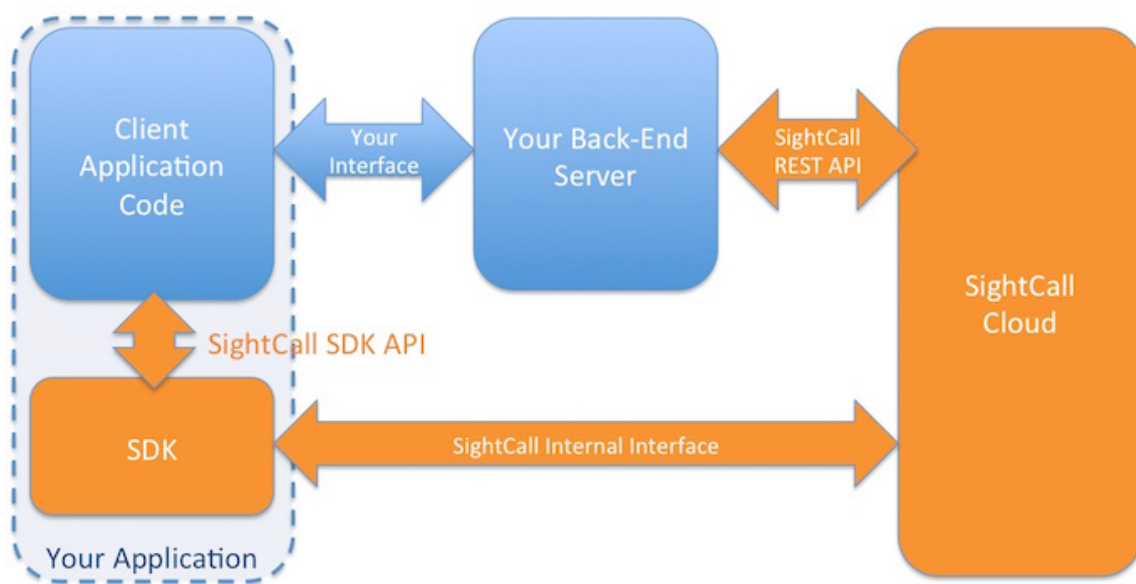


SightCall Design Document

Overview/Architecture

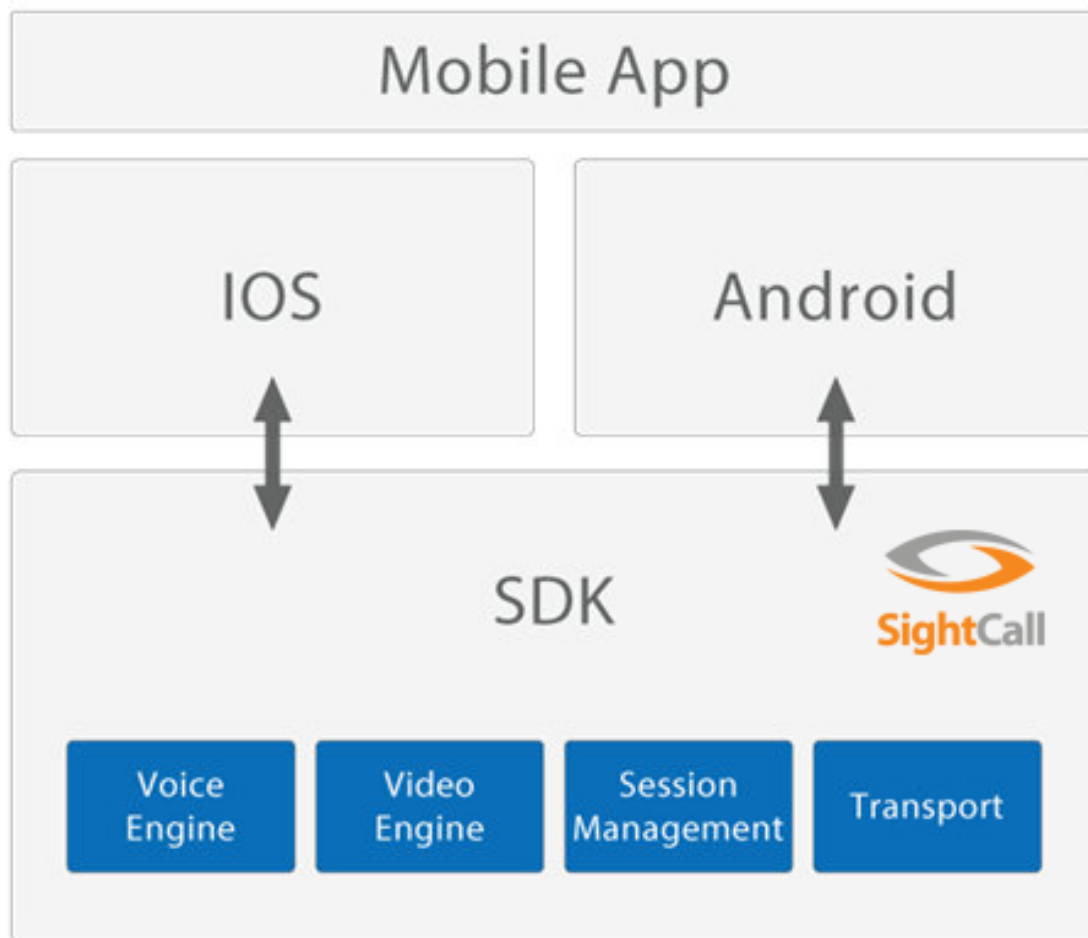
The SightCall's platform is based on SDKs to be embedded in application on client side (i.e. on user side) and on a cloud where most operations take place.

A Token based authentication is needed on the client side. This token is obtained from the cloud. Once authenticated, SDKs can perform any operation directly from the commands it receives. Application front end or back end can also request some services directly from the cloud using a Rest API.



SightCall is a global WebRTC platform built on a global real-time video infrastructure. Our platform also has a development framework that encompasses front-end and back-end APIs. The APIs include a JavaScript API that abstracts native WebRTC browser APIs and extends real-time video capabilities to non-WebRTC browsers. A cornerstone of our front-end integration is our, iOS and Android SDKs. The back-end

APIs are intended for authentication, reporting and provisioning.



SightCall's platform helps you bring Real Time Communications to your application. In order to ease integration of those features, SightCall provides SDK for all supported devices and a REST Api for server to server features.

SightCall provides client SDKs for:

JavaScript to embed real time communications in any web application. No matter if your user browser supports WebRTC or not, a single Javascript SDK encapsulates communication stack through WebRTC or through our native plugin for Windows or MacOS. On API, all available technologies.

Android to embed real time communication in your Java Android application.

iOS to embed real time communication in your iOS application. The iOS SDK can be linked with an Objective-C or a Swift application.

In addition to that, SightCall provides a **Rest API** for back-end operation like authentication or conference management, as well as a REST API designed for direct client requests.

Features it supports

As real time communications, SightCall helps you add into your app

- One to One Direct Calls
- Multiparty Conference Calls
- 6 digits Call Routing
- Waiting Room Features
- Chat services
- Inband Data transfer
- Presence and Roster Management
- Annotations features
- Automatic Call Dispatcher features
- Call Recording

One to One Direct Calls

Any authenticated user can call another authenticated user using his uid. The routing and the signalization of the new call is handled by the SightCall's platform. All standard notifications are sent on both side (ringing, proceeding, call active, hangup with reason,...). During a call, both attendees can start or stop video or sound, can start a screenshare or send low-latency messages to the other.

Multiparty Conference Calls

In addition to One to One call, SightCall's platform supports multiparty calls up to 32 participants. A waiting room mechanism helps the management of multiparty calls and control of the attendees. Even if anybody can join a multiparty call, a minimum profile is needed to

6 digits Call Routing

Call routing can also be managed within the Rtcc SDKs. A plus user can create at any moment a 6 digits code that can be given to a non registered user. No need for him to have an account in your website, he can simply authenticate to the Rtcc cloud using this code and will have limited rights: the code authenticated user can only start a call to the user who gave him the code or can join a multiparty call hosted by the user who gave him the code.

Waiting Room Features

A complete set of feature has been introduced in the concept of waiting room. Based on meeting point where users can meet, the host of a call can accept or deny the right to users to enter a call. Users are placed in a waiting room until the host notifies the system of his answer.

Chat services

Chat service are also available in the SightCall's platform. Any user can send chat messages to another user authenticated in the same provider. Chat messages can be acknowledge or not, depending on the application needs and integration.

Inband Data transfer

In addition to chat, attendees of the same call can send inband data message to each others. Inband data messages are send within the video stream (inband) and have the same latency than the video. This feature is dedicated to short, low latency messages in a call. It

can be used for sending coordinates of a pointer for example, but also for any kind of data the application is integrated with (IOT,...)

Presence and Roster Management

SightCall's platform also provides an advanced presence management. When a user connects to the cloud, he is automatically registered to the presence servers with a presence value equals to 0. The user, or his application, can set this value to any value between 0 and 255, 0 meaning not present. Any user on the cloud can request the presence value of another user (presence feature) and can also subscribe to changes of this value (Roster feature).

Annotations features

During a call, annotation are available, i.e. a user can annotate a paused video or a screen share to highlight some points on what he wants to show.

Automatic Call Dispatcher features

An automatic call dispatcher (ACD) is also available. Agents can register with several criterias and users can request to start a call with an agent that matches requested criterias. The ACD then dispatches requests to matching agents that can accept or deny the call request. This feature can be linked to presence to add the presence as a criteria for the dispatch.

Call Recording

SightCall's platform can also record multiparty call, either locally and forward the recording on an Amazon storage, or via chunk http post request to a provider server.

SDK Implementation

To start an implementation based on the Rtcc SDKs and on the SightCall's platform, you will find all that you need on the portal:account.sightcall.com.

SAPortal

The account.sightcall.com website contains a summary of all the mandatory values that are required to connect to the cloud and initialize the clients SDKs.

AppId

The AppId uniquely identifies your account. It is also known as your provider identifier. This key is not meant to be secret and is used by SDK to identify your account to the Cloud.

How is it built(Design)

1.How to make Make a one-to-one call

Once connected and authenticated, SightCall clients can call another connected and authenticated client based on the uid he/she is connected with. The callee SDK is notified of an incoming call and can pick up the call (audio video, audio only) or decline it. During the call, any peer can also start a screen share to share his/her desktop or any window (desktop clients: plugin or WebRTC) or any view (mobile clients: android or iOS).

Create a call

Calling someone else is done using a createCall method of the Rtcc

object (JS,Android,iOS). This method can take several parameters but the most important ones are:

uidToCall: uid of the user to call.

displayNameToCall: string displayed in the call window until the callee has been found (i.e. until the call is ringing). It will then be replaced by the displayname that the callee has set up himself.

Receive a Call

For mobile devices, as only one call can be done at a time, actions are done on the current active call.

Subscribe to Events of the call object

Although no implementation other than the create call is necessary to make or receive calls, the application may want to add additional behaviors. Several events occur during the call that the application may want to subscribe to, including:

local video start or stop

remote video start or stop

screen share start or stop

call terminated

Hang up a call

Hanging up a call is done using the hangUp methods of the SDKs on the call object.

2.Manage a Waiting Room

The SightCall platform proposes a Waiting Room feature. A user can create a virtual place dedicated to wait for a meeting to start. This place is called a Waiting Room and is reachable by a Meeting

Point (identified by its Meeting Point ID aka MPI).

Except for the hostless case, attending a call behind a waiting room always happens in two steps:

the host creates a meeting point: users can then enter the waiting room, but are not able to enter the conference.

the host effectively hosts the call: only then can users e

A meeting point can also be in one of the three following modes:

auto-accept: every attendee entering the meeting point is immediately accepted onto the call (once hosted)

default: to enter the call the attendee is waiting for approval from the host

locked: nobody can enter the call once the meeting point has been locked

3.Chat Feature

- Receive a Message

Received messages trigger the event

`message.receive`.

```
//  
To receive message the delegate shall be  
set  
Rtcc.instance().dataDelegate = theDataDel  
egate  
  
// in RtccDataDelegate's  
func rtccReceivedData(data: NSData!, from  
contactUID: String!, withID messageID: U  
Int8) {  
    ...
```



```
}
```

- Acknowledge a Message

4.Presence Values

When a user is not connected, his presence is set to 0. The meaning of the 0 value is reserved and cannot be changed.

When a user connects, his/her presence value remains 0 until he/she changes it to any other value.

Presence has to be explicitly changed to any other value by the web application.

Except for 0, it is the responsibility of the web application to define what the presence values mean. For example an application 'A' can set up the following lookup table:

Presence	Meaning
0	Not connected
1	Available
2	Busy
3	Already on a call
4	Will be back soon

And an application 'B' can use very different values:

Presence	Meaning
0	Not connected
10	Available
20	Writing a message
30	Responding to a message
40	Asleep, do not disturb

41	Asleep, call only for urgent matter
42	Asleep, please wake me up with a call