|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **SIP Proxy** | | The role of the SIP Proxy module is to convert the SIP transport from WebSocket protocol to **UDP**, **TCP** or **TLS** which are supported by all legacy networks. If your provider or hosted server supports SIP over **WebSocket** (e.g. Asterisk or Kamailio) then, you can bypass the module and connect the client directly to the endpoint. Bypassing the SIP Proxy is not recommended if you’re planning to use the RTCWeb Breaker or Media Coder modules as this will requires maintaining two different connections. There are no special requirements for the end server to be able to talk to the Proxy module. | | SIP Proxy architecture | | *SIP Proxy architecture* | |
| |  | | --- | | **RTCWeb Breaker** | | The RTCWeb specifications make support for **ICE** and **DTLS/SRTP** mandatory. The problem is that many SIP-legacy endpoints (e.g. PSTN network) do not support these features. It’s up to the RTCWeb Breaker to negotiate and convert the media stream to allow these two worlds to interop.  For example, if your server doesn't support ICE this means it requires the RTCWeb Breaker in order to be able to connect the browser to a SIP-legacy endpoint.  We highly recommend checking the [Technical Guide](https://www.doubango.org/webrtc2sip/technical-guide-1.0.pdf" \t "_blank) to understand how to avoid **security issues** when using this module. | | SIP Proxy architecture | | *RTCWeb Breaker architecture* | |
| |  | | --- | | **Media Coder** | | The RTCWeb standard defined two MTI (**M**andatory **T**o **I**mplement) audio codecs: **opus** and **g.711**. For now there are intense discussions about the MTI video codecs. The choice is between **VP8** and **H.264**. VP8 is royalty-free but not widely deployed while H.264 AVC is not free but widely deployed. Google has decided to use VP8 in Chrome while Ericsson uses **H.264 AVC** in [Bowser](https://labs.ericsson.com/apps/bowser). Mozilla and Opera Software will probably use VP8 and Microsoft H.264 AVC. As an example, the Media Coder will allow to make video calls between Chrome and Bowser. Another example is calling a Telepresence system (e.g. Cisco) which most likely uses **H.264 SVC** from Chrome. | | SIP Proxy architecture | | *Media Coder architecture* | |
| |  | | --- | | **Click-to-Call** | | This is more a service than a module as it’s a complete SIP [click-to-call](http://en.wikipedia.org/wiki/Click-to-call" \t "_blank) solution based on the three other components. The goal is to allow any person receiving your mails, visiting your website, reading your twitts, watching your Facebook/Google+ profile to call you on your mobile phone with a single click.   [Enjoy our live click-to-call demo »](http://click2dial.org/" \t "_blank) | | Click-to-Call Components | | *Click-to-Call Components* | |