



# 基于go和flutter的实时通信/视频直播解决方案



段维伟

---

湖北捷智云技术有限公司  
创始人



# 目录

背景

01

技术简介

02

客户端

03

服务端

04

开源社区

05

Q&A

06

第一部分

# 背景

# 即将讲述的内容

---

- WebRTC 实时通讯
- Flutter 跨平台UI 开发框架
- 基于Flutter UI 框架的WebRTC 插件 flutter-webrtc
- Go 语言的WebRTC 协议栈 pion/webrtc
- 基于pion/webrtc 的应用级服务框架 pion/ion

# 真实世界的需求点

---

- 5G 时代，实时通讯应用爆发
- 疫情影响，全世界都在使用远程教育，远程办公
- 云游戏，机器人，VR，直播等
- 如何用最容易的方案实现实时通讯
- 漂亮的app，最好全部(mobile, web, desktop)平台都支持.
- 最容易使用的后端技术

WebRTC + Flutter + Go

第二部分

# 技术简介

# WebRTC 是什么

---

- 由 Google 发起的基于浏览器通讯标准
- 基于收购来的 GIPS (6800万美金)的高质量实时音视频引擎
- 支持主流浏览器主流移动设备
- 历时十年成为Web 实时通讯标准
- RTMP 直播协议的低延迟替代方案





# WebRTC 可以做什么

---

- 用实现网页音/视频通话
- 低延迟直播系统(在线课堂)
- 多人视频会议系统
- 高质量SIP/VOIP系统
- 视频监控系统
- 机器学习，视觉计算等



# 如何使用它

---

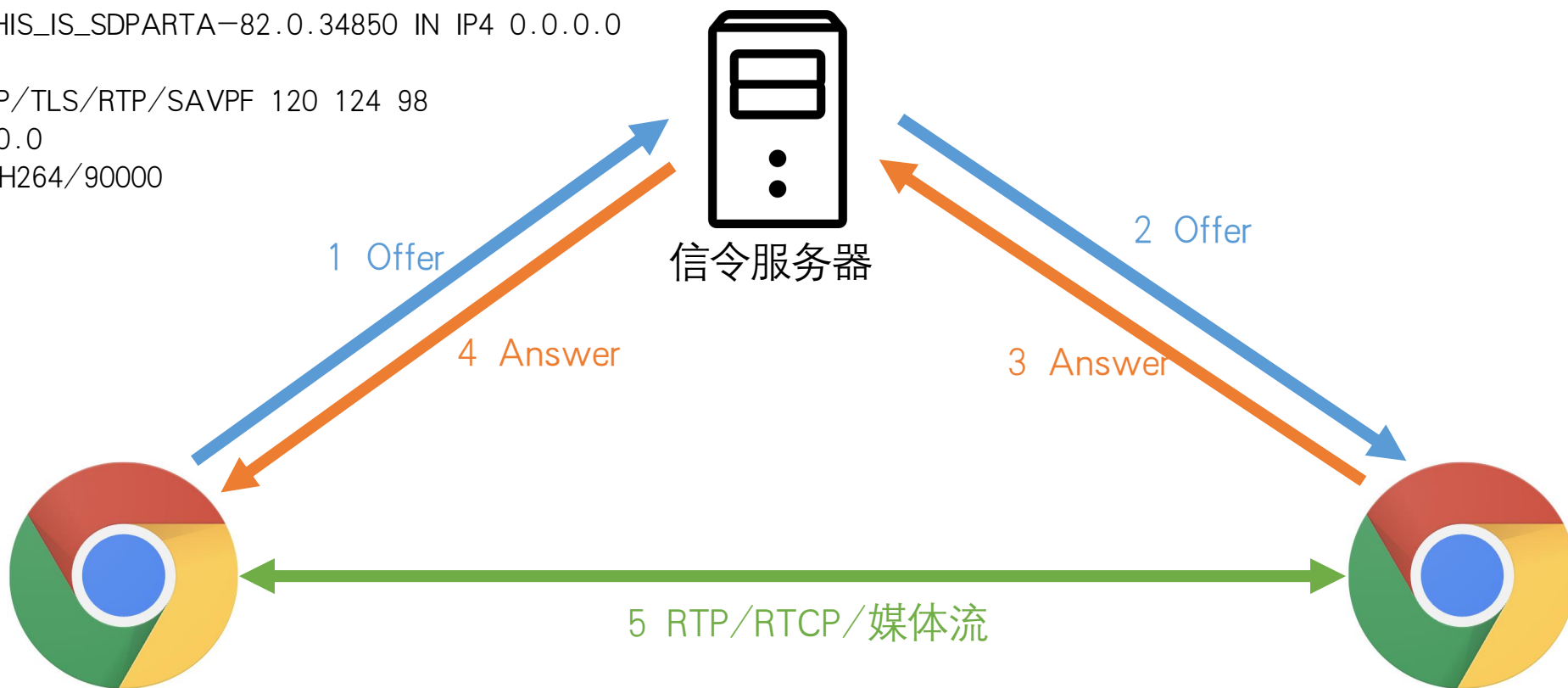
- 在Web中使用JS API
- 基于google libwebrtc实现原生客户端开发 (ios/android/c++)
- 使用第三方堆栈实现兼容功能 (Go)  
<https://github.com/pion/webrtc>



# 一对一视频原理

SDP(Session Description Protocol)

```
v=0  
o=mozilla...THIS_IS_SDPARTA-82.0.34850 IN IP4 0.0.0.0  
s=-t=0 0  
m=video 9 UDP/TLS/RTP/SAVPF 120 124 98  
c=IN IP4 0.0.0.0  
a=rtpmap:120 H264/90000  
b=AS:1000  
a=sendrecv
```



第三部分

# 客户端

# WebRTC客户端开发

---

- Web端 标准JS +html 或通过google 官方提供的源码编译出原生sdk
- iOS/macOS 使用WebRTC.framework + Obj-C/Swift
- Android 使用libwebrtc.aar + java
- Windows 使用libwebrtc.dll + C++

# 实际开发中会遇到的困难

---

- 下载和编译Google WebRTC框架（防火墙，编译环境）
- 原生SDK开发(每平台人力投入)
- UI 的一致性，更新迭代（类似SDK需按平台维护）
- 性能问题(全部使用html5)

---

客户端是否有  
更好的选择？

# 为何选择 Flutter

---



- 同样是 Google 发起的跨全平台高性能UI框架
- 基于 Skia 2D 渲染引擎
- 使用类似JS/TS的Dart 语言开发
- 支持代码编辑后热重载,



# Flutter 支持那些平台



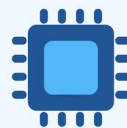
Mobile



Web



Desktop



Embedded

# 使用flutter 开发app意味着什么？



- 无需为每个平台独立维护代码
- 一次编码，多平台运行,效率最大化
- 多平台一致性体验
- 强大的社区资源



# Flutter-WebRTC 插件

# Flutter-WebRTC 支持那些平台

## Functionality

Feature	Android	iOS	Web	macOS	Windows	Linux	Fuchsia
Audio/Video	✓	✓	✓	✓	✓	[WIP]	
Data Channel	✓	✓	✓	✓	✓	[WIP]	
Screen Capture	✓	✓	✓				
Unified-Plan	✓	✓	✓	✓			
Simulcast	✓	✓	✓	✓			
MediaRecorder	⚠	⚠	✓				

# 实现一对一视频通话服务 (基于 Go 开发)

---

## Flutter WebRTC Server

- Support Windows/Linux/macOS
- Built-in web, signaling, [turn server](#)
- Support [REST API For Access To TURN Services](#)
- Use [flutter-webrtc-demo](#) for all platforms.

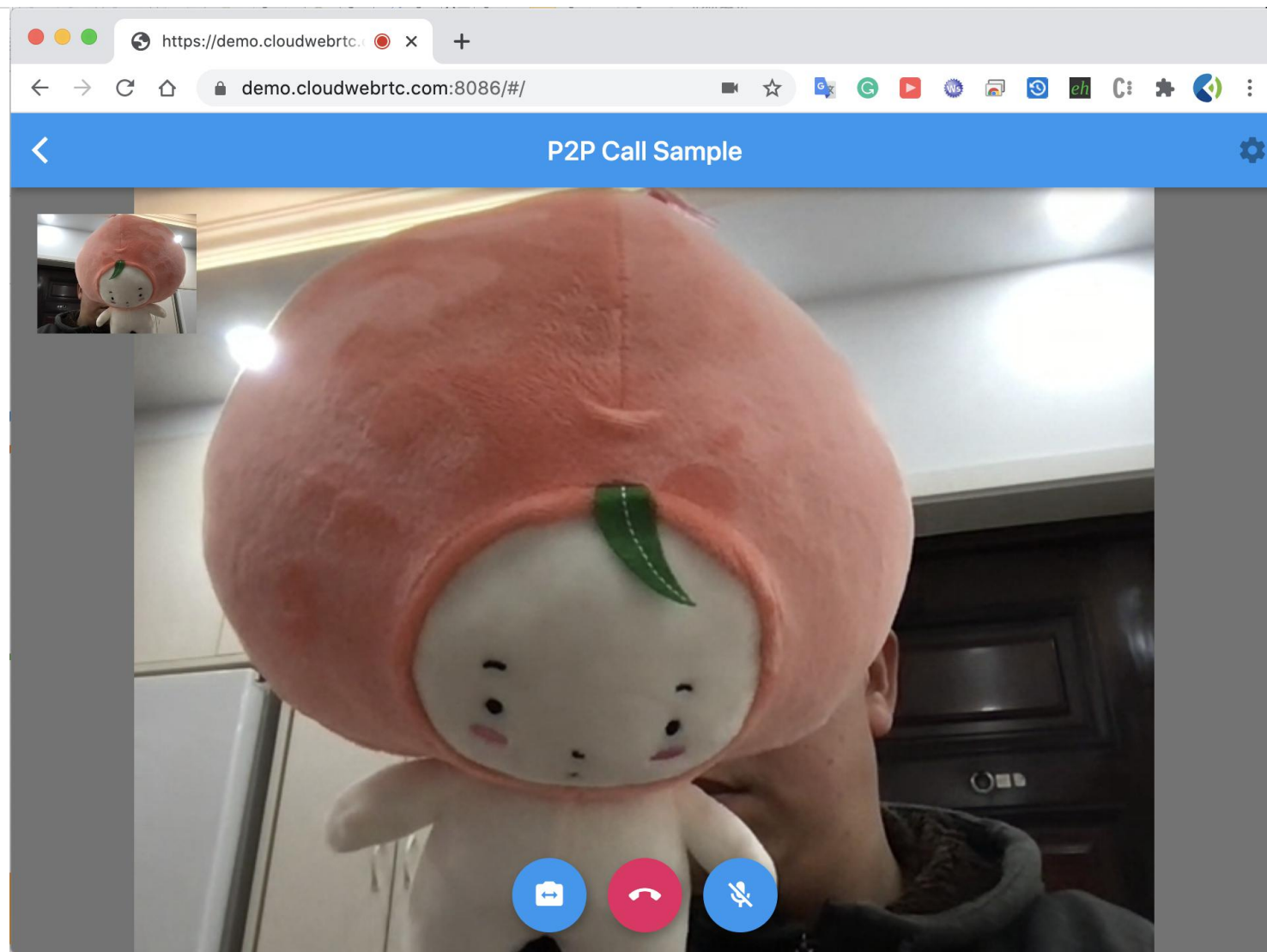
源码:

<https://github.com/flutter-webrtc/flutter-webrtc-server>

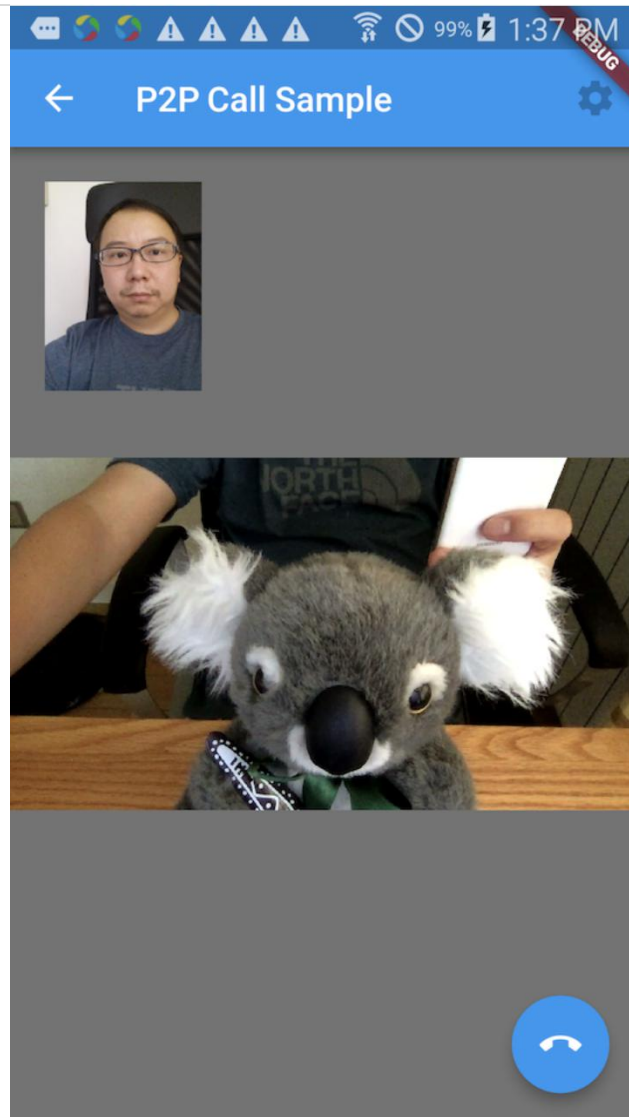
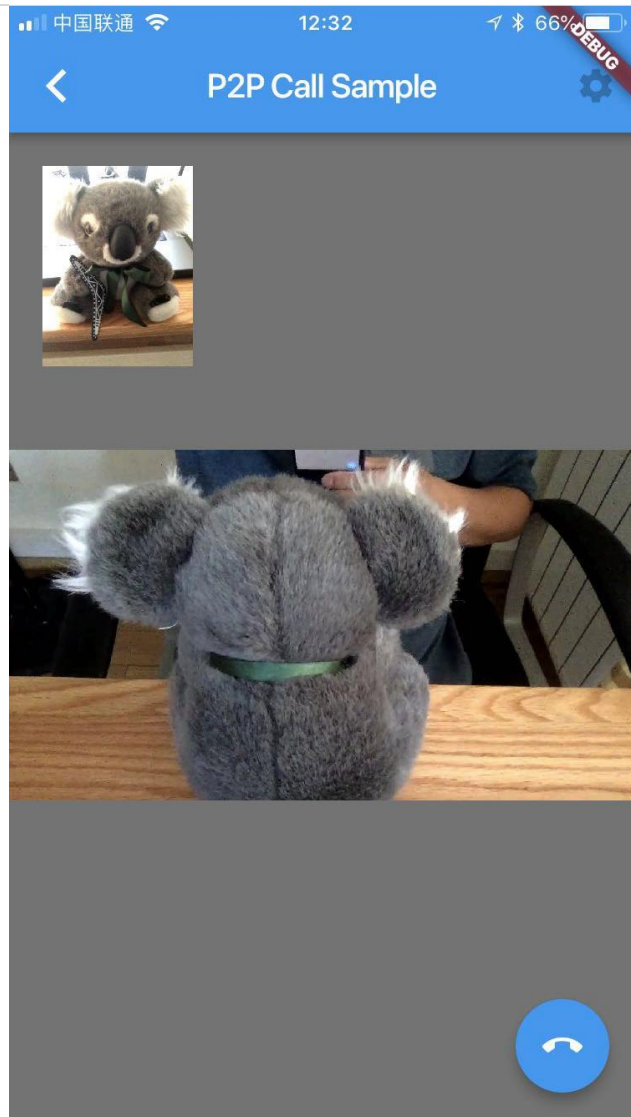
<https://github.com/flutter-webrtc/flutter-webrtc-demo>



# 一对一 网页端



# 一对一 iOS/Android



第四部分

# 服务端



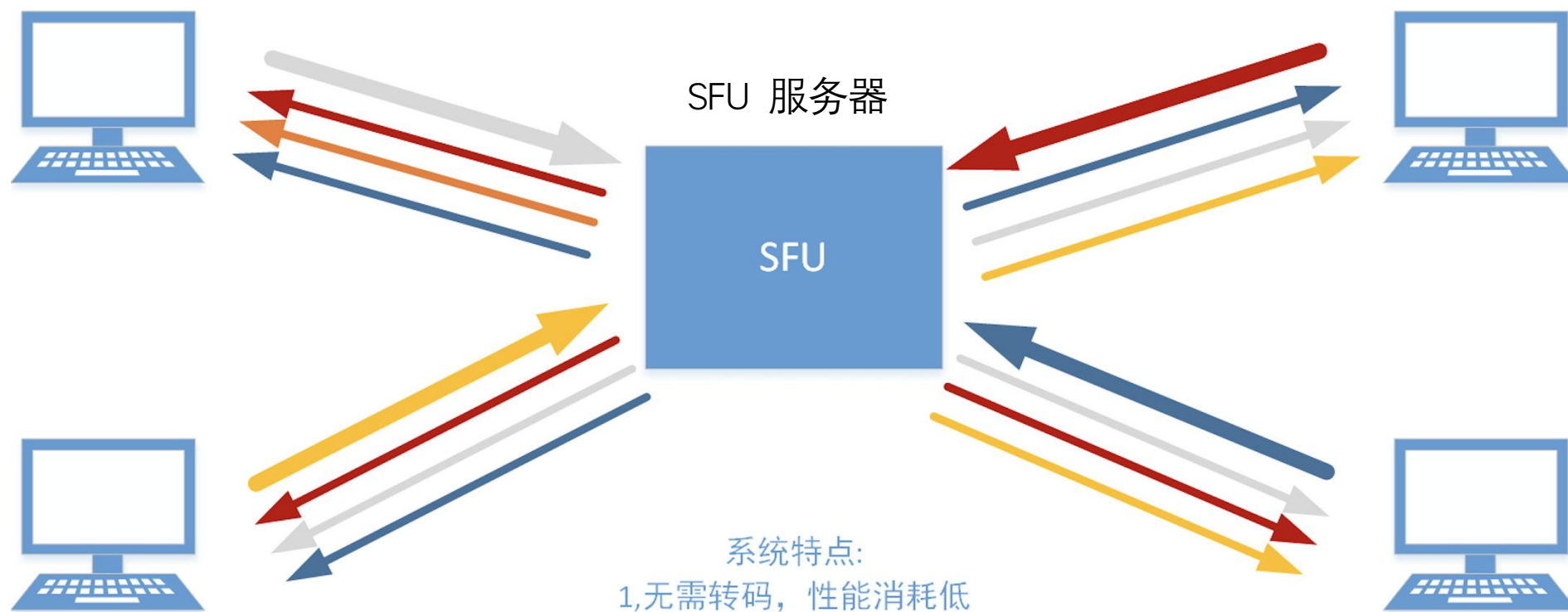
# 如何实现多人视频服务

---

- 一个SFU 服务器  
(pion/ion-sfu)
- 一个信令协议与客户端通讯, 完成SDP的交互  
(Websockets/json-rpc/grpc)
- 一个WebRTC 客户端  
(浏览器/原生客户端/flutter 客户端)

# SFU工作原理

## 基于SFU 的视频对流转发





## Pion 介子

作者: Sean DuBois

The Open Source, Cross Platform Stack for Real-time Media and Data Communication.

<https://pion.ly/>

<https://github.com/pion>

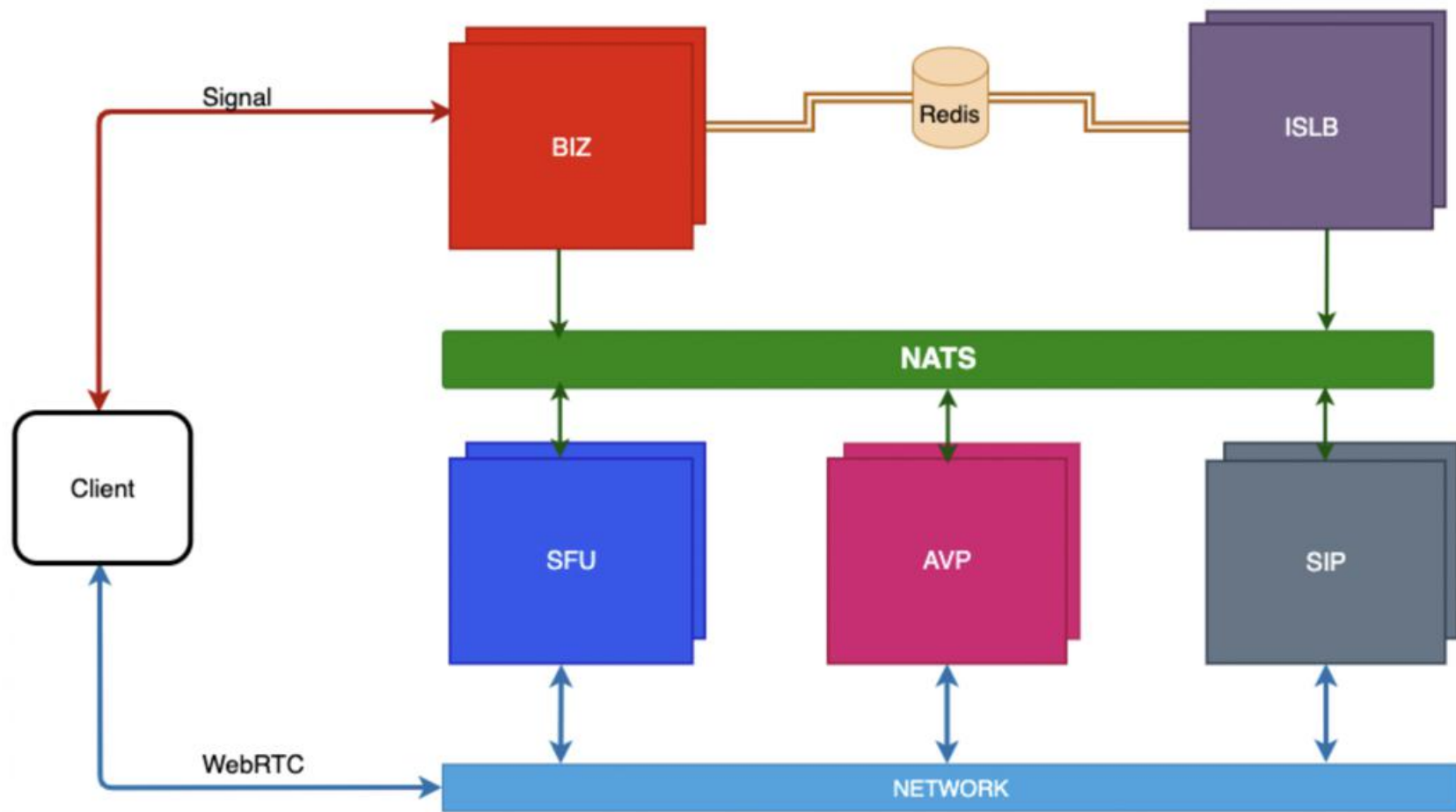
# ION 离子之光

分布式实时通讯系统  
<https://github.com/pion/ion>



- 基于pion/webrtc 开发 pion/ion-sfu
- 分布式架构
- 基于grpc over NATS mq
- 使用redis 存储媒体流全局位置
- 支持业务自定义开发
- 高性能, 单个ion-sfu节点 1k 并发仅需 0.5核

# ION 架构 多node 架构



# 主要模块



- ISLB 服务发现，负载均衡，媒体信息全局存储
- Biz 业务接入模块
- SFU 节点（用于转发webrtc 流，与biz模块配合创建视频会议系统）
- AVP 节点（用于从SFU节点拉取数据进行数据处理，视频流存储，音视频混合，CV，语音识别，图形识别等）
- 网关节点与go-sip stack 或rtmp/hls/srt等协议组合成协议网关
- 转换RTC流成直播流推送到CDN

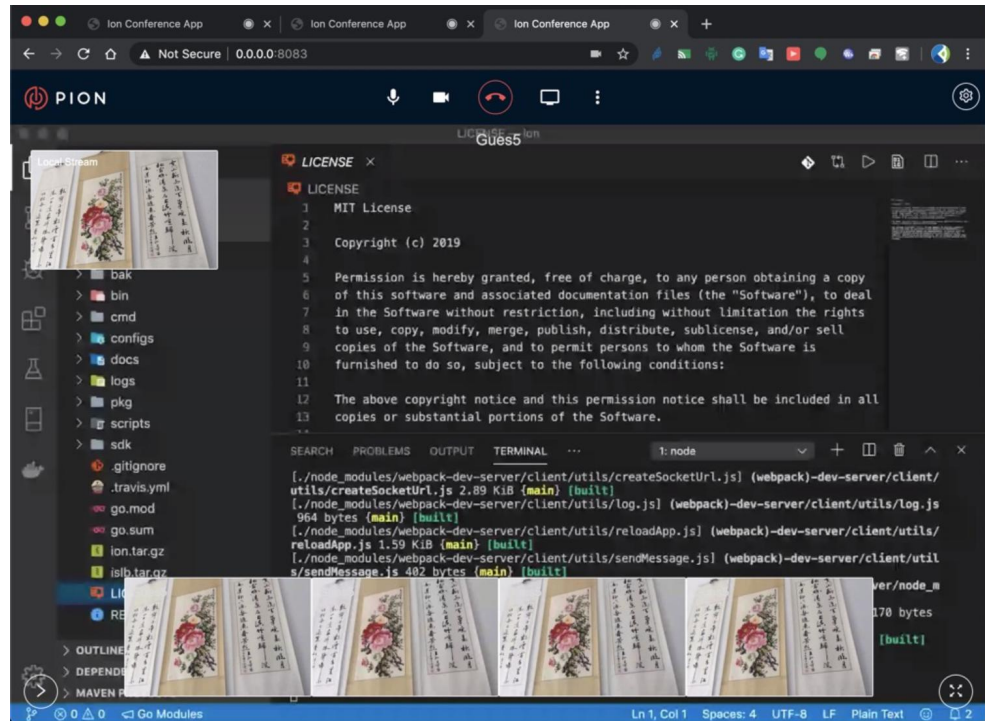
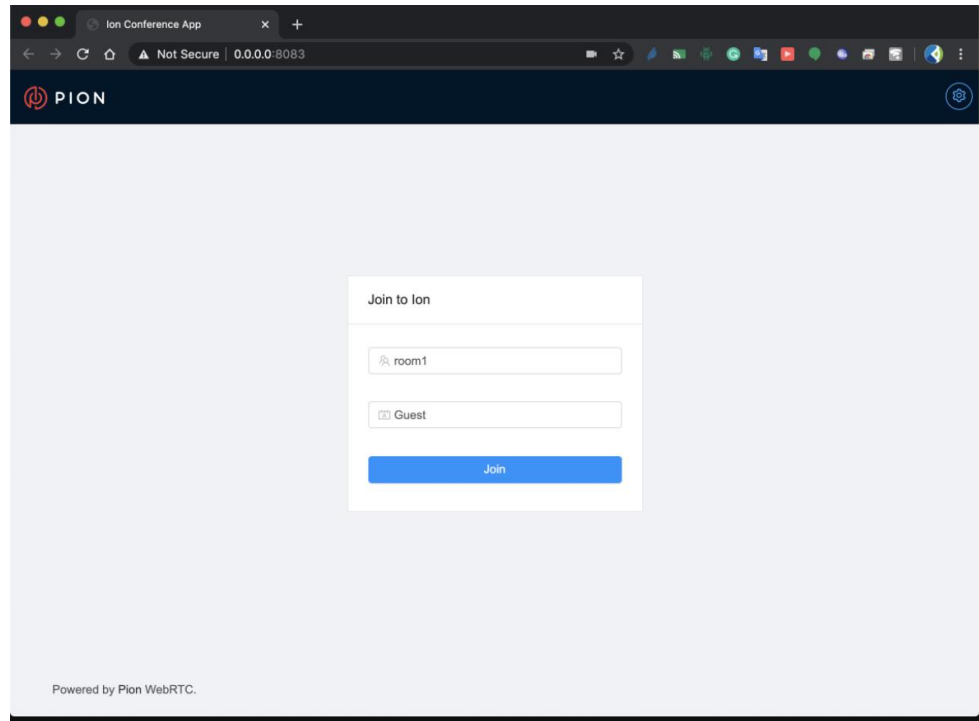
# ION客户端支持

---



- `lon-sdk-js` 网页端SDK
- `lon-sdk-flutter` 全平台flutter SDK
- `lon-sdk-go` 用于开发推流工具, 压力测试, avp 节点开发
- `lon-app-web`
- `lon-app-flutter`

# ion-app-web

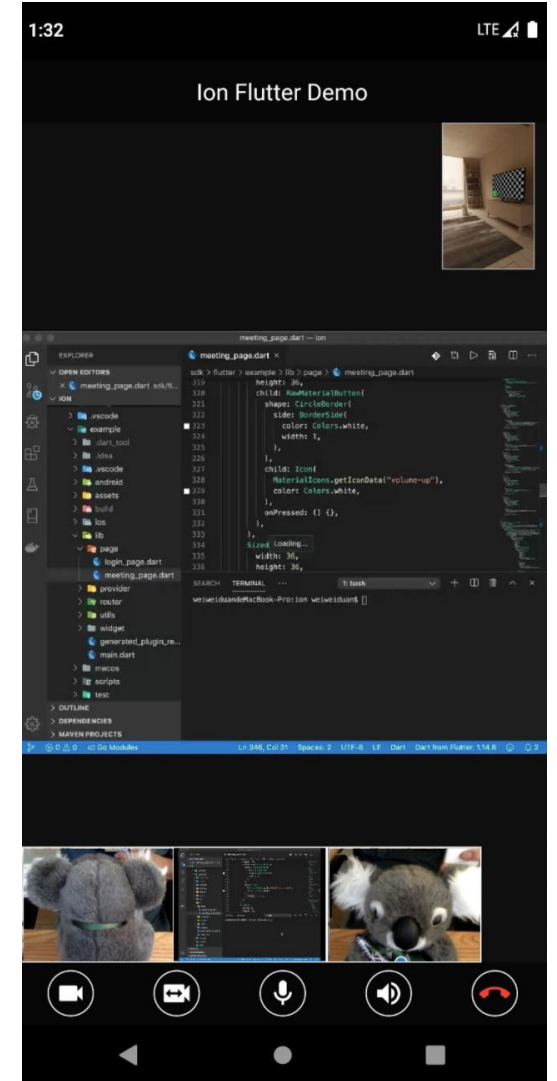
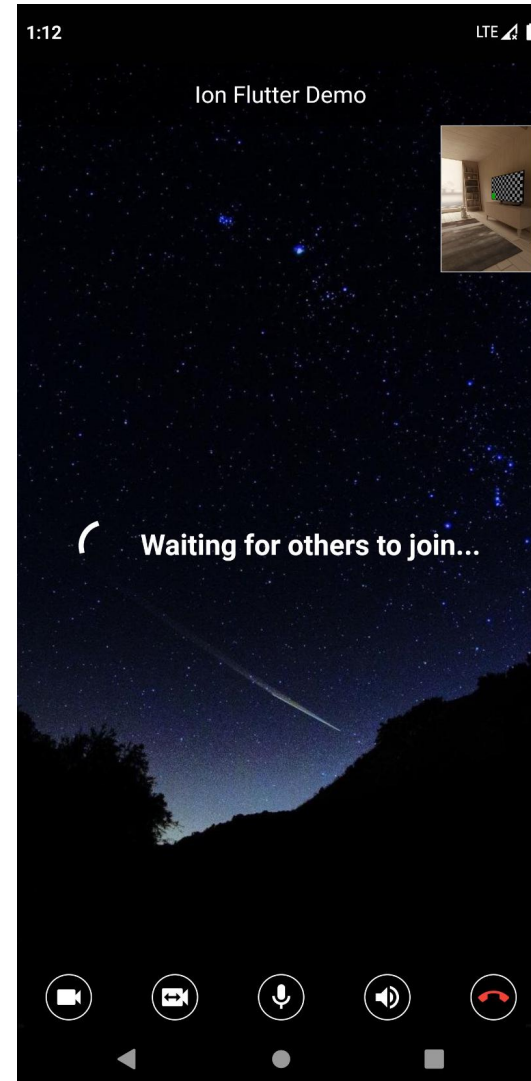
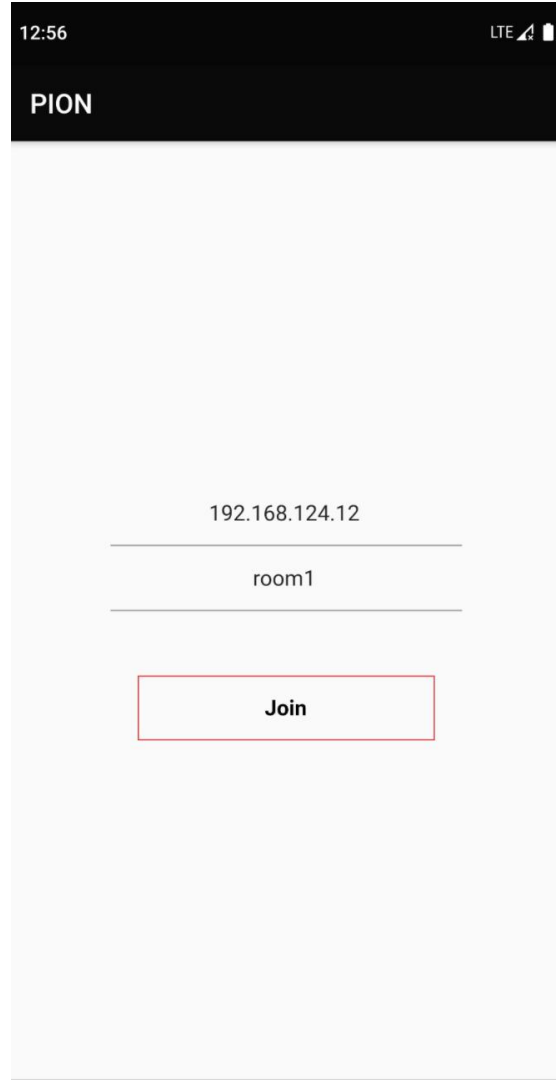


- 基于 React.js + ant-design
- 多人视频
- 屏幕共享
- 文字聊天



# ion-app-flutter

- 多人视频
- 文字聊天
- 屏幕共享
- iOS/Android
- macOS/Windows

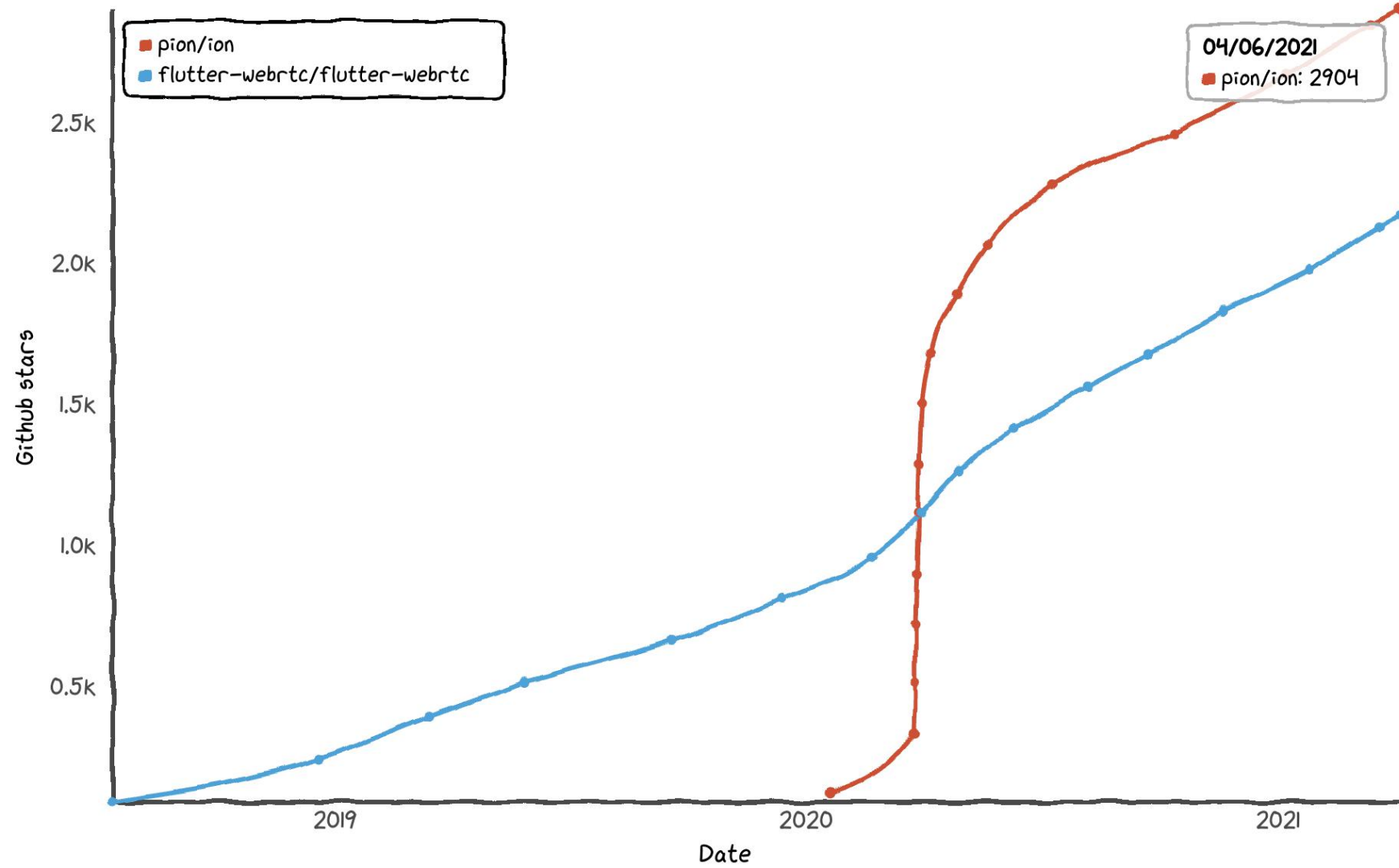


第五部分

# 开源社区



## Star history



# ion-\* 社区主要贡献者

- Adwpc <https://github.com/adwpc>
- Kangshaojun <https://github.com/kangshaojun>
- Jin Gong <https://github.com/cgojin>
- jason-shen <https://github.com/jason-shen>
- Tarrence van As <https://github.com/tarrencev>
- Jason Brady <https://github.com/jbrady42>
- Leeward Bound <https://github.com/leewardbound>
- Orlando Co <https://github.com/OrlandoCo>
- CloudWebRTC <https://github.com/cloudwebrtc>

pion 和 ion 社区的所有代码均为MIT授权,

相关github 地址如下:

<https://github.com/pion/ion>

<https://github.com/flutter-webrtc>

<https://github.com/pion/ion-app-web>

<https://github.com/pion/ion-app-flutter>

欢迎有兴趣的小伙伴参与到ion社区.

邮件:duanweiwei1982@gmail.com



# Q&A