FermoLUG
linuxfm.org

CC BY-NC-SA 4.0

Hack the Headset: Bluetooth e Speech to Speech su Raspberry Pi



Giacomo Vallorani
24 Ottobre 2026

Su di Me



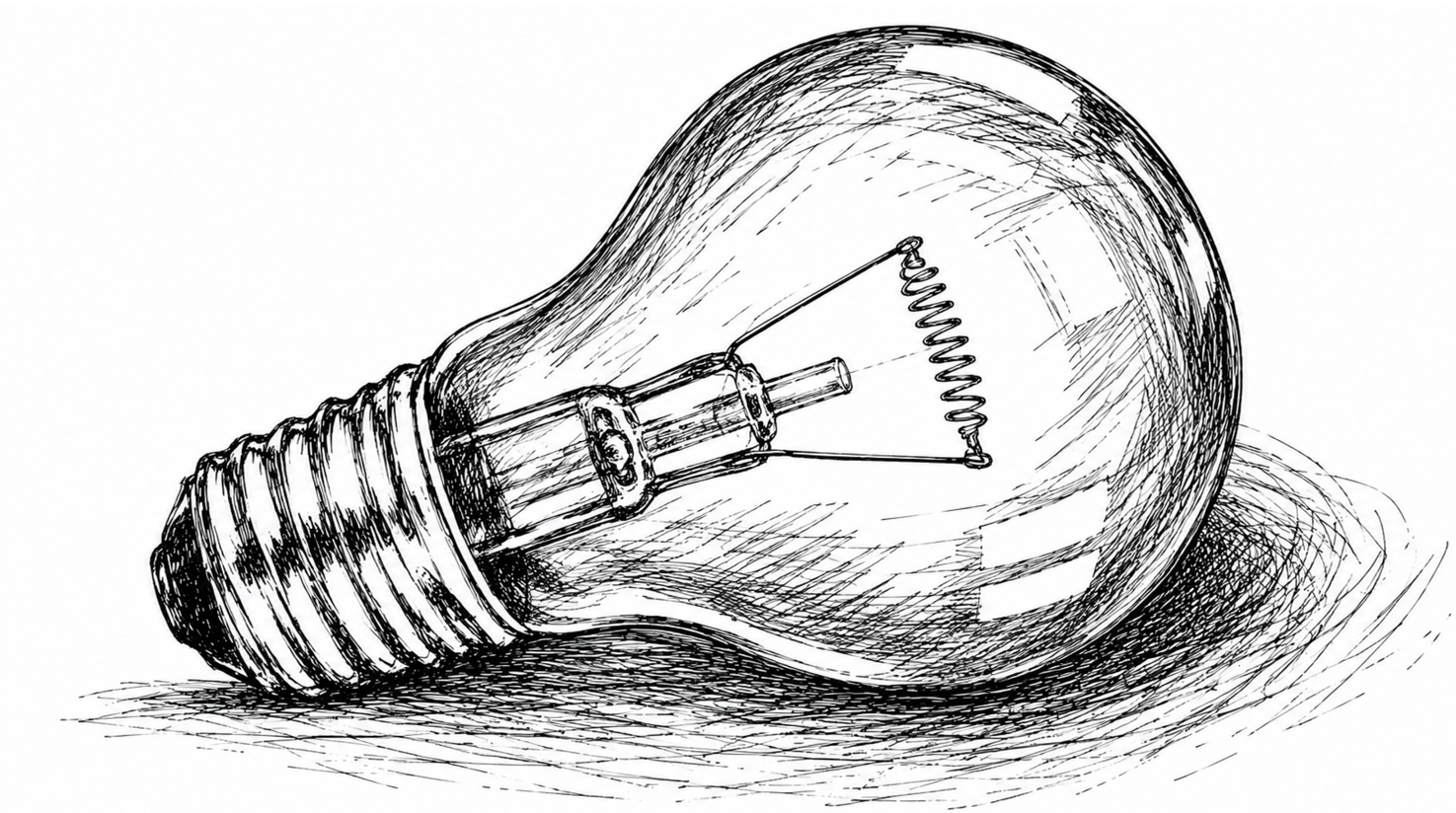
Giacomo Vallorani

- Backend Software Engineer @ PagoPA
- Maker nel tempo libero
- Sistemi distribuiti e IoT

@Vallasc

giacomovallorani

1 L’Idea



OpenAI Realtime API



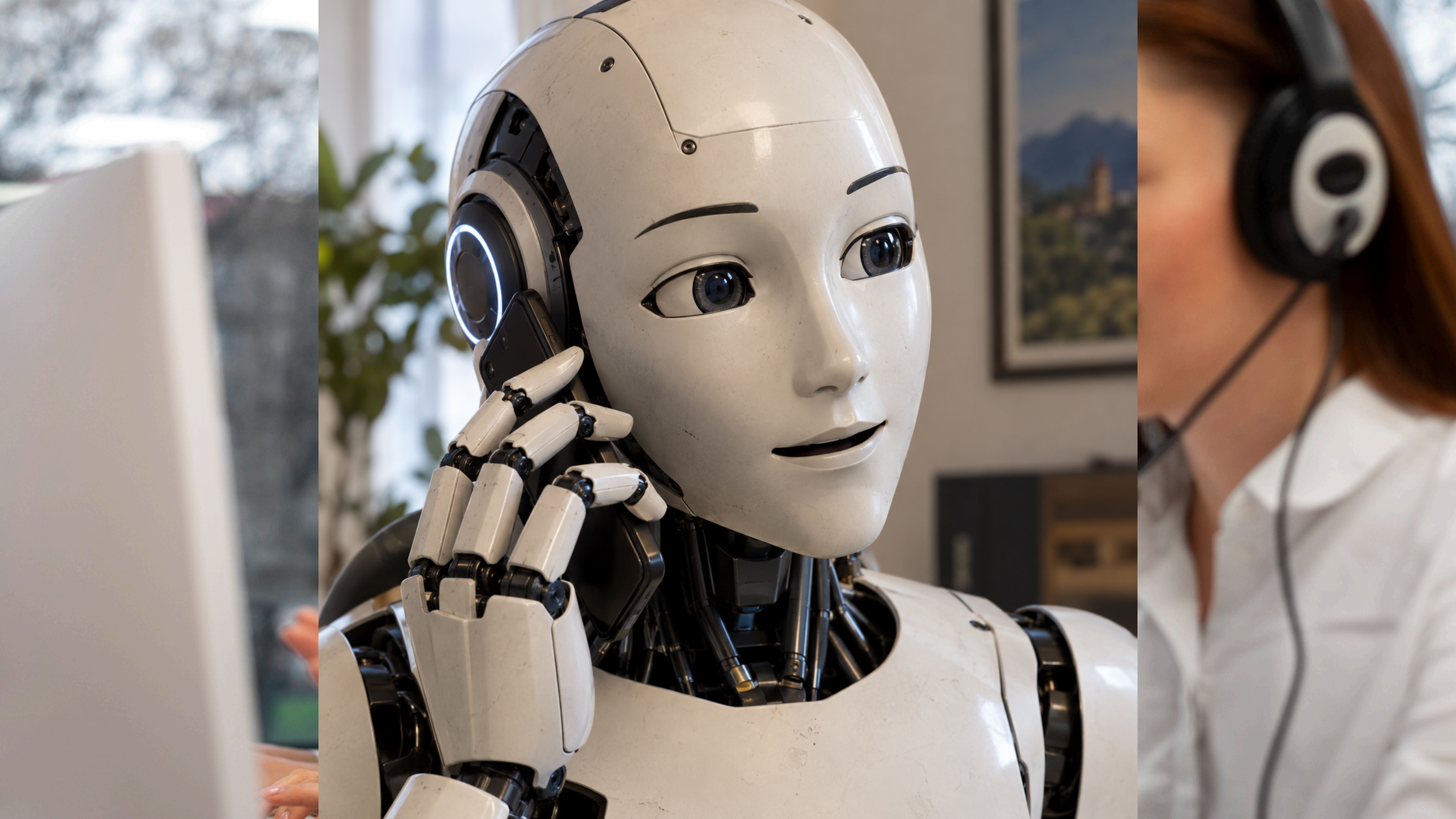
- WebSocket/WebRTC
- PCM a 16 o 24-kHz,
  G.711 μ-law (PCMU)
  A-law (PCMA)
- VAD
  (Voice Activity Detection)
- Tempo di risposta
  da 300ms a 800ms

OpenAI Realtime API



- WebSocket/WebRTC
- PCM a 16 o 24-kHz,
  G.711 μ-law (PCMU)
  A-law (PCMA)
- VAD
  (Voice Activity Detection)
- Tempo di risposta
  da 300ms a 800ms





L’Audio delle chiamate è blindato

Android

- In entrata:
  `CAPTURE_AUDIO_OUTPUT`,
  riservato alle app di sistema
- In uscita:
  non esiste API pubblica
- Un’app può rispondere solo alla chiamata

iOS

- In entrata:
  non esiste API pubblica
- In uscita:
  non esiste API pubblica

VoIP?

- **Numero da un provider SIP** (deviazione di chiamata dal numero personale)
- Le chiamate arrivano a un server
- Un centralino software o un servizio cloud passa l’audio all’LLM.
- L’accesso all’audio è completo in entrambe le direzioni.

VoIP?

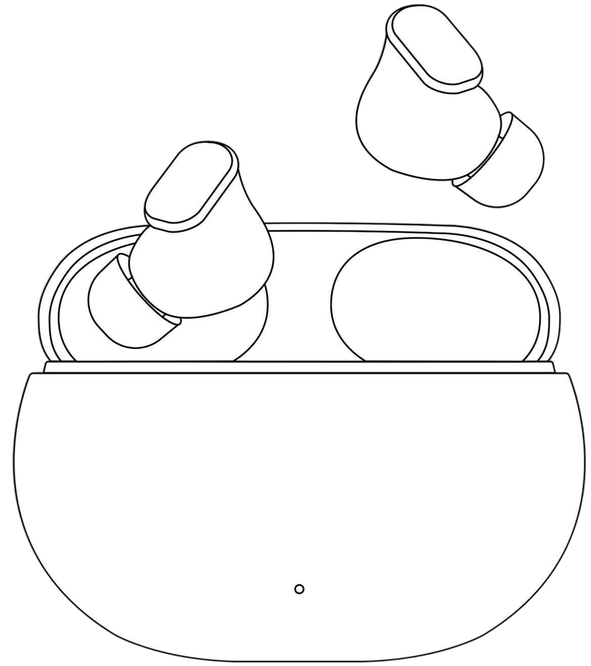
- **Numero da un provider SIP** (deviazione di chiamata dal numero personale)
- Le chiamate arrivano a un server
- Un centralino software o un servizio cloud passa l'audio all'LLM.
- L'accesso all'audio è completo in entrambe le direzioni.

Funziona ma...

- Non è il mio numero
- Costo al minuto + deviazione
- Le chiamate in uscita mostrano il numero VoIP

E se ci fosse un dispositivo che possa intercettare le chiamate?

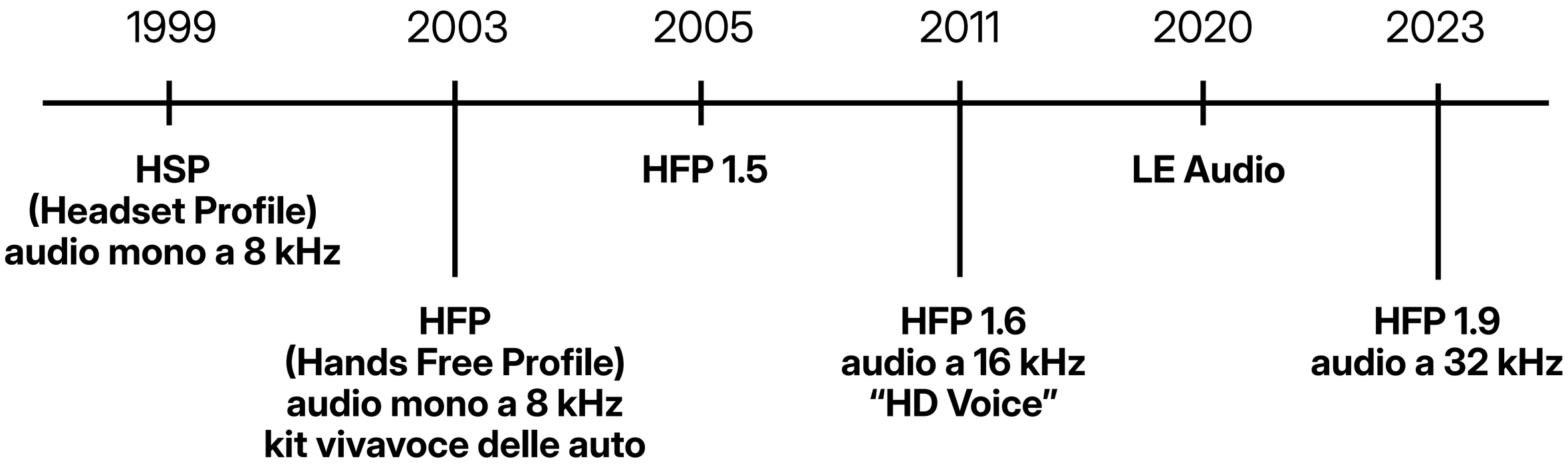
E se ci fosse un dispositivo che possa intercettare le chiamate?





2 Bluetooth Classic: Hands Free Profile (HFP)

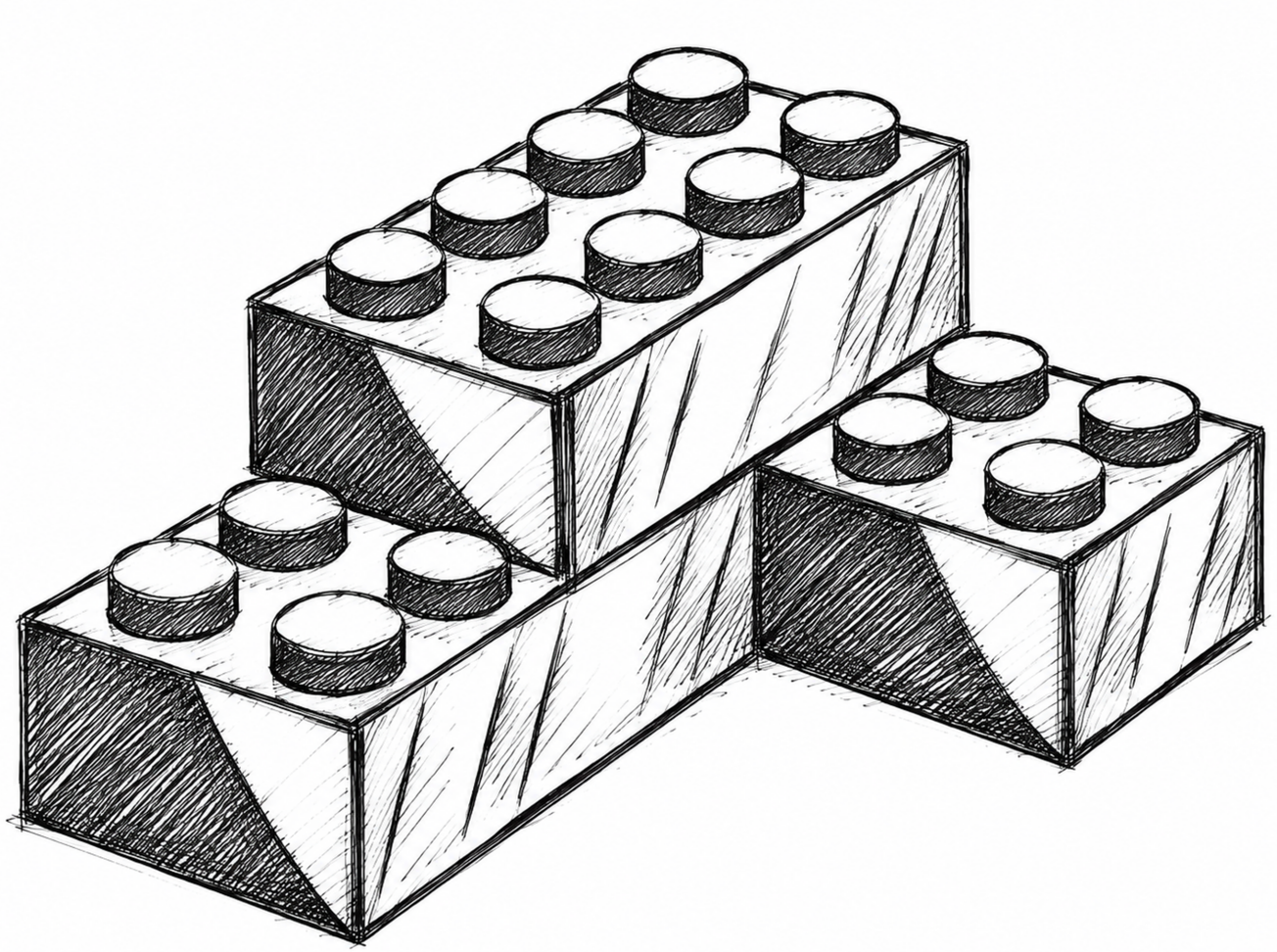
Bluetooth Classic: da HSP a HFP



Come funziona HFP

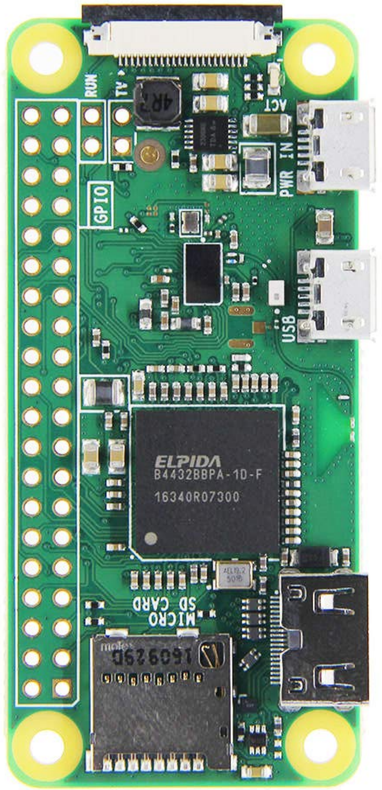
- **Audio Gateway (AG)**: il telefono, che ha la chiamata.
- **Hands-Free unit (HF)**: l'auricolare o l'auto (il nostro dispositivo).
- **Canale di controllo (RFCOMM)**: testo, comandi AT. Chi chiama, rispondi, riattacca, volume.
- **Canale audio (SCO/eSCO)**: un link sincrono a bassa latenza, voce in entrambe le direzioni.

3 Architettura



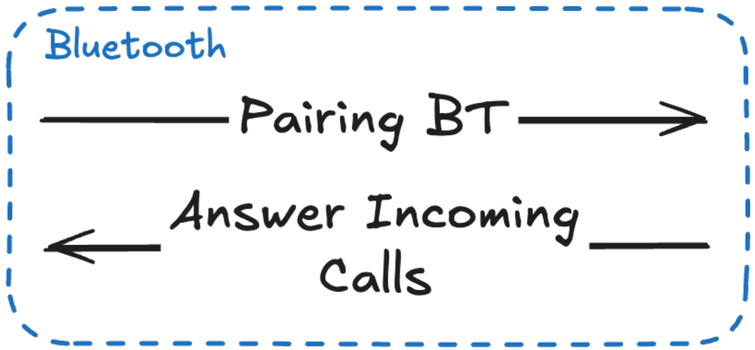
Idea alla base

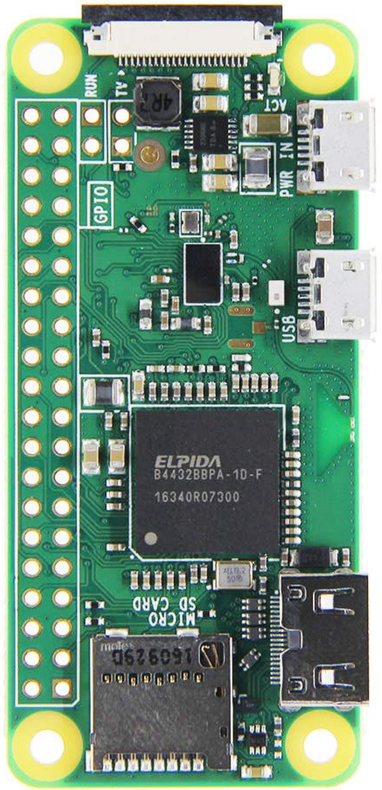




Idea alla base







Idea alla base



Bluetooth

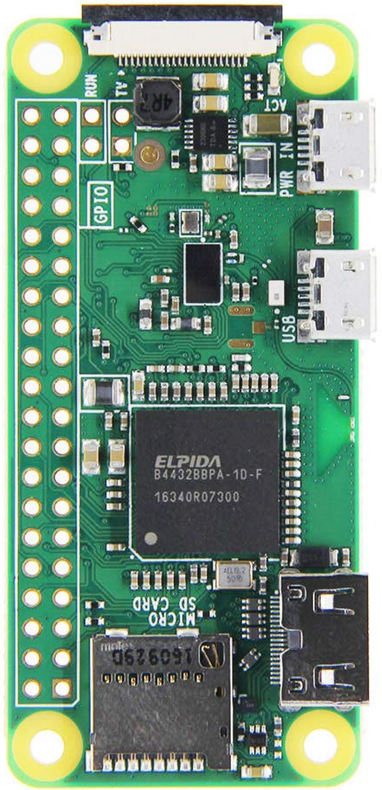
Pairing BT →

← Answer Incoming Calls

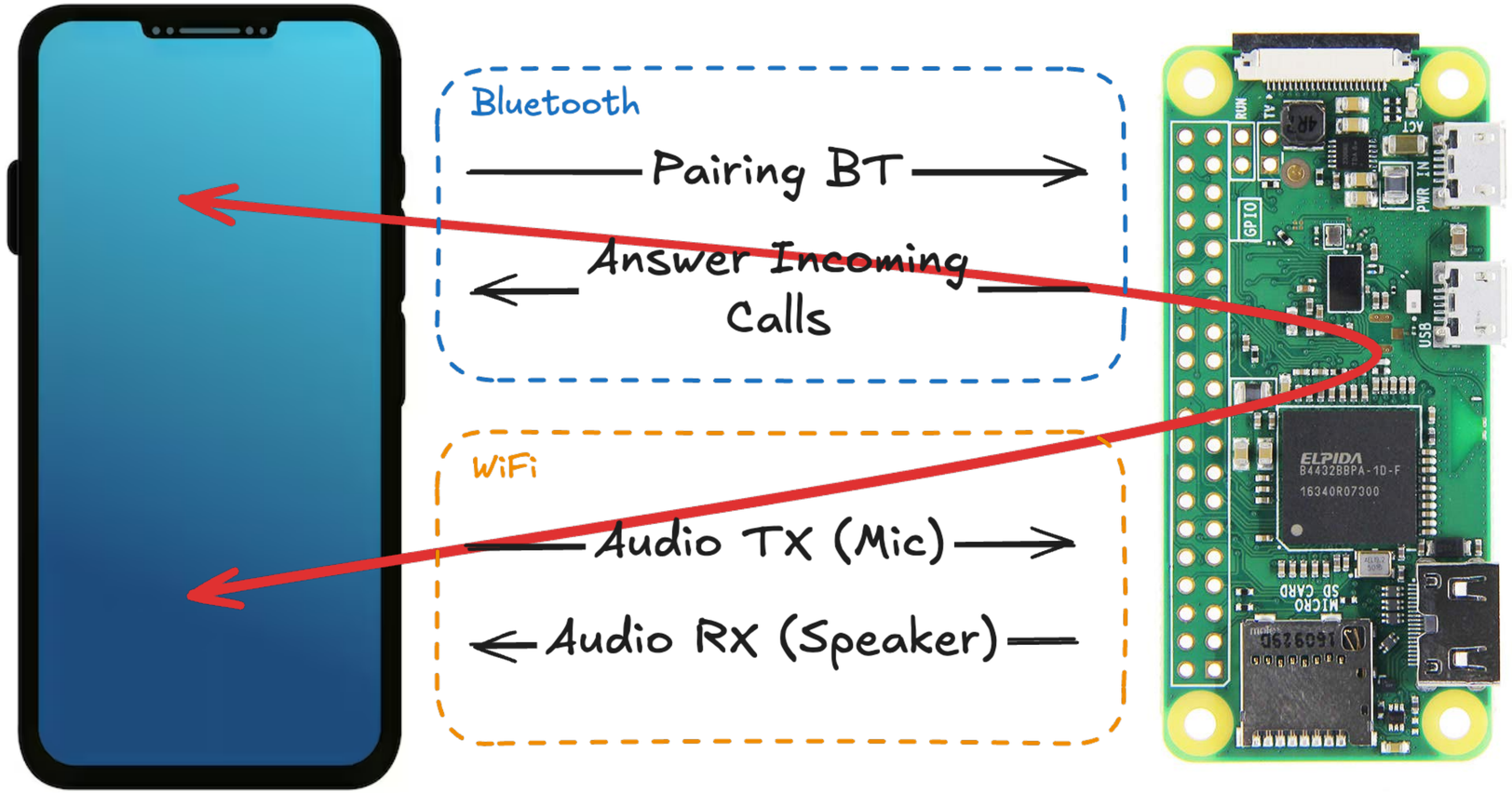
WiFi

Audio TX (Mic) →

← Audio RX (Speaker)



Idea alla base



BlueZ

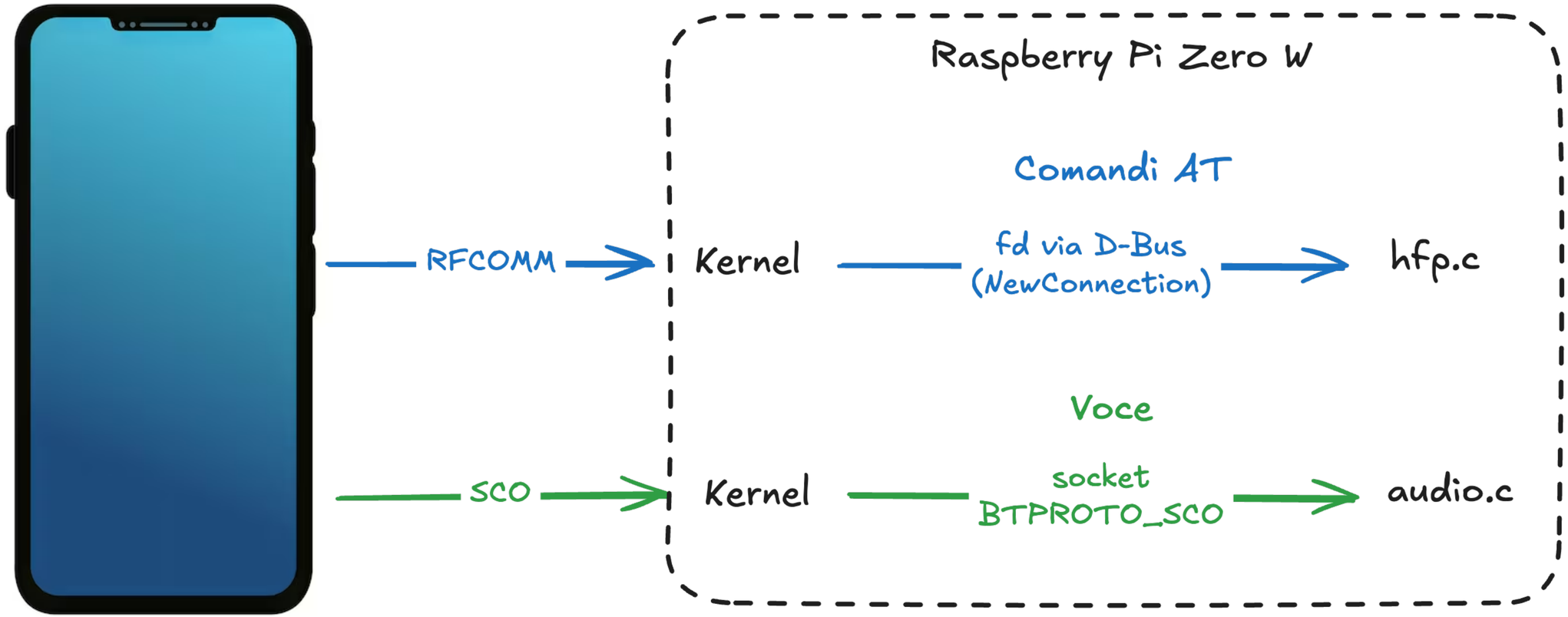
Stack Bluetooth ufficiale di Linux, diviso in due parti:

- **nel kernel**: HCI, L2CAP, RFCOMM e SCO, esposti come socket (`AF_BLUETOOTH`)
- **in userspace**: il demone `bluetoothd`, che gestisce adattatore, pairing, profili e GATT, e pubblica tutto come API su **D-Bus** (le interfacce `org.bluez.*`)

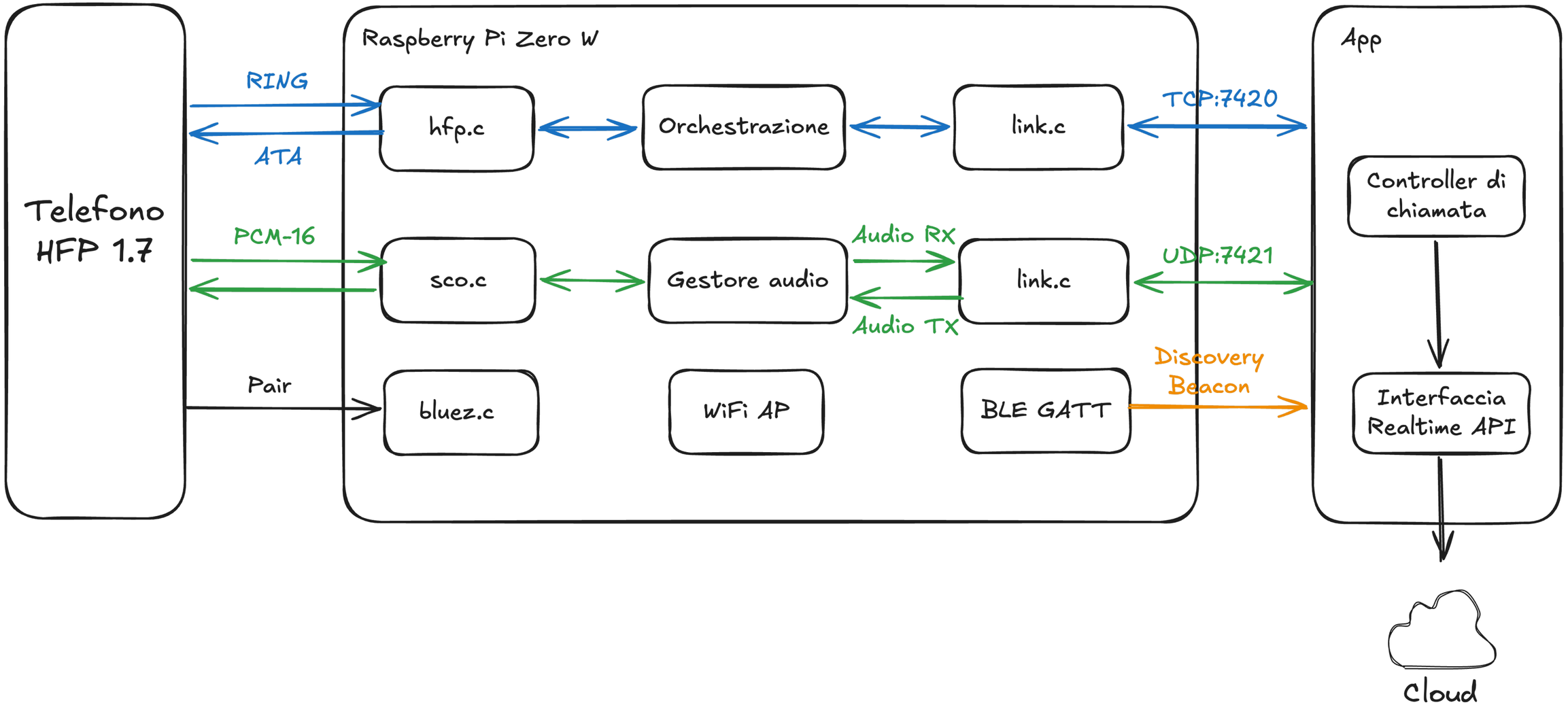


https://github.com/bluez

BlueZ e l'app custom

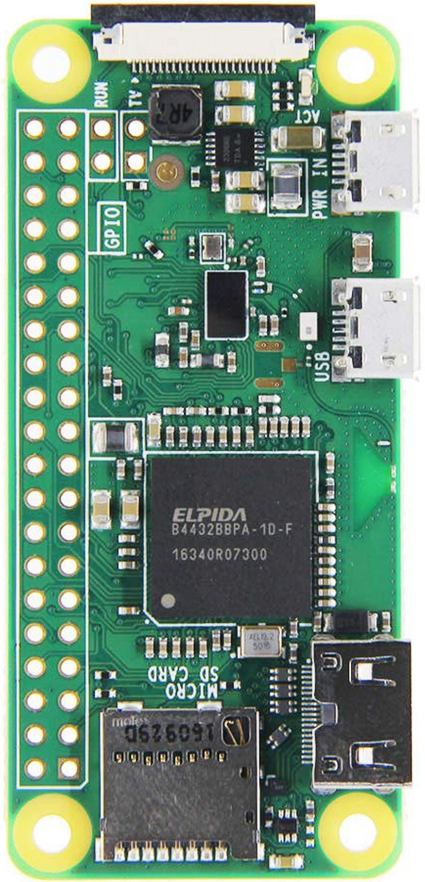


Architettura



Raspberry Pi Zero W (2017)

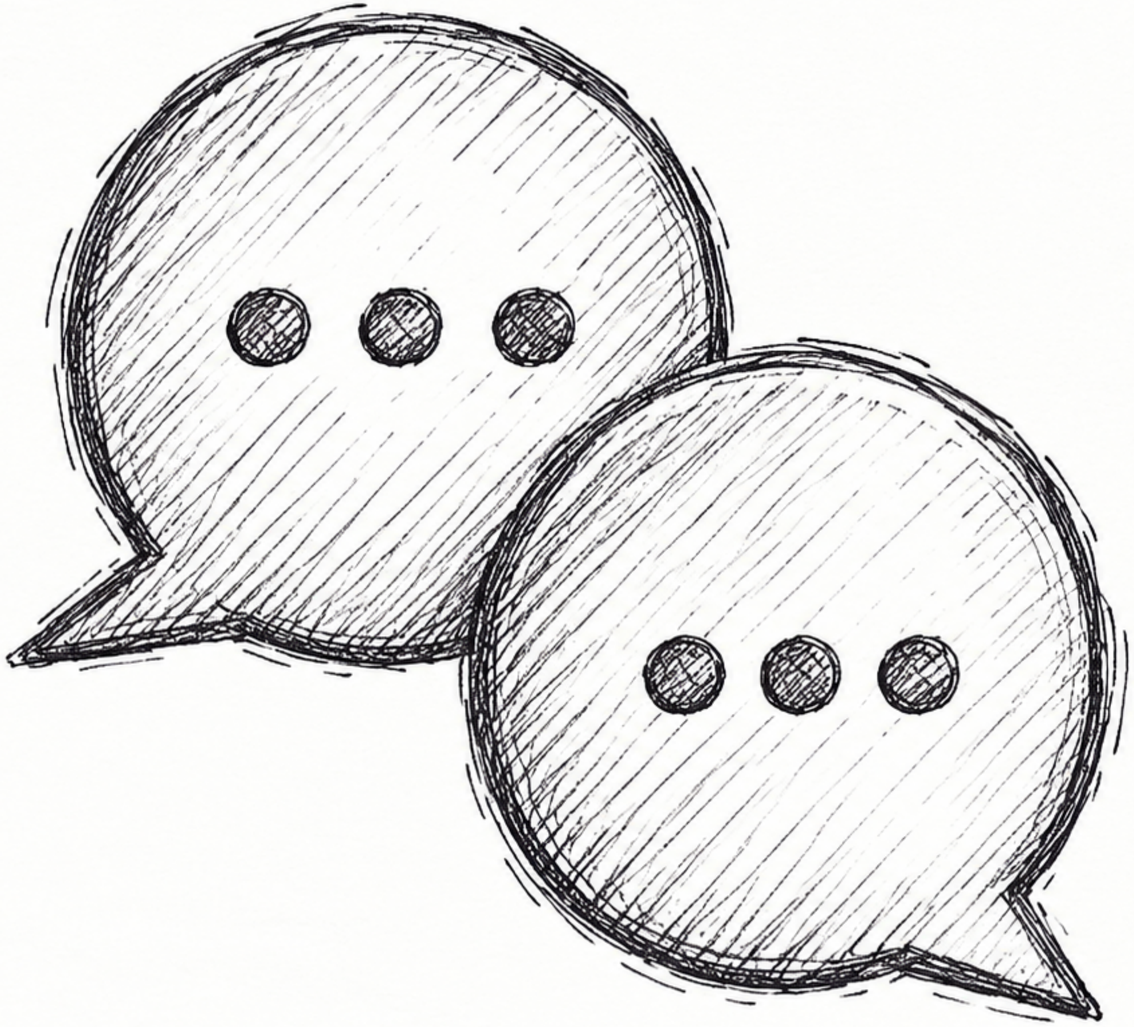
- CPU: ARM11 single-core a 1 GHz (BCM2835)
- RAM: 512 MB
- Radio: Wi-Fi 2.4 GHz + Bluetooth 4.1
- Basso consumo
- Costo: 20€ ??
- Attualmente out of stock



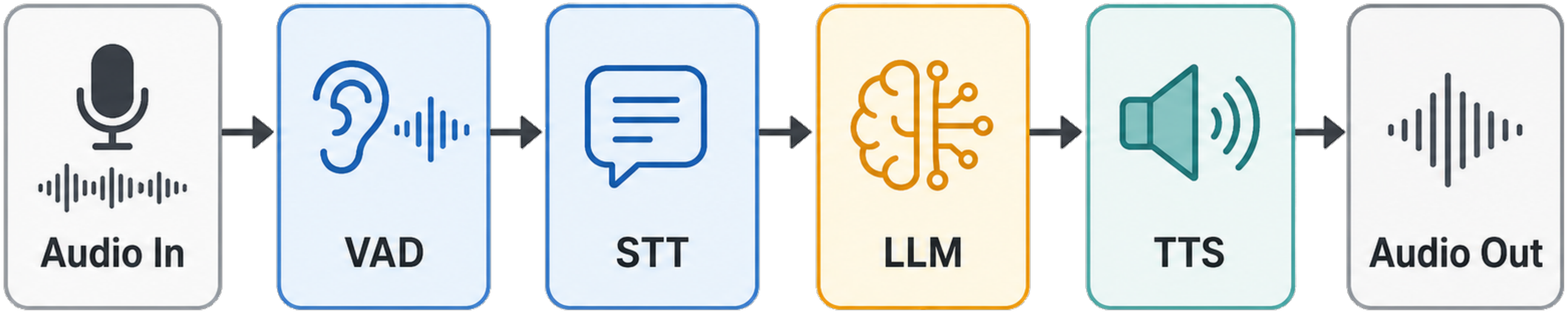
Perché non un ESP32?

- Costa pochi euro e supporta HFP
- Ma Bluetooth e Wi-Fi condividono **una sola radio** a 2.4 GHz, in time-sharing
- Audio in tempo reale + streaming verso l'LLM via Wi-Fi = la radio non sta al passo: audio a scatti e pacchetti persi
- I modelli più recenti (ESP32-S3, C3, C6) hanno solo Bluetooth Low Energy, niente Bluetooth Classic, quindi niente HFP

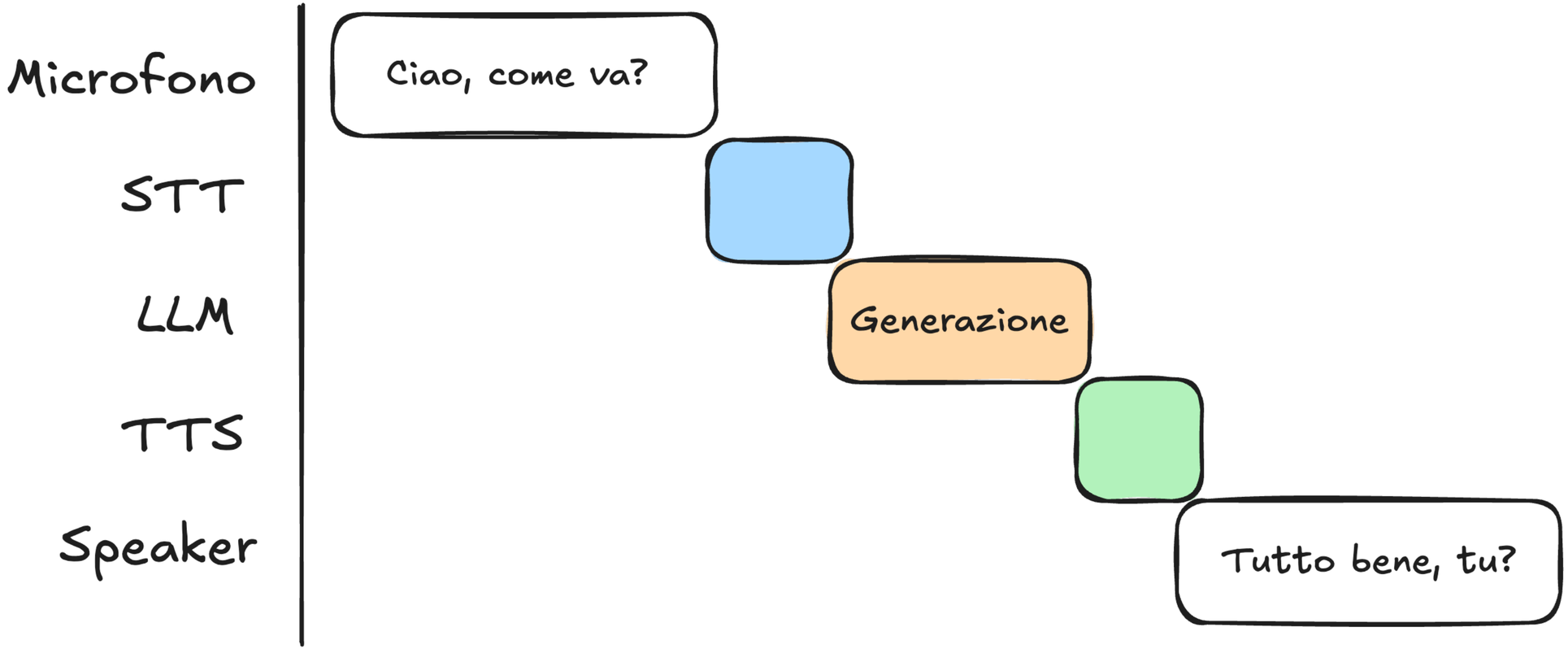
4 Speech to Speech



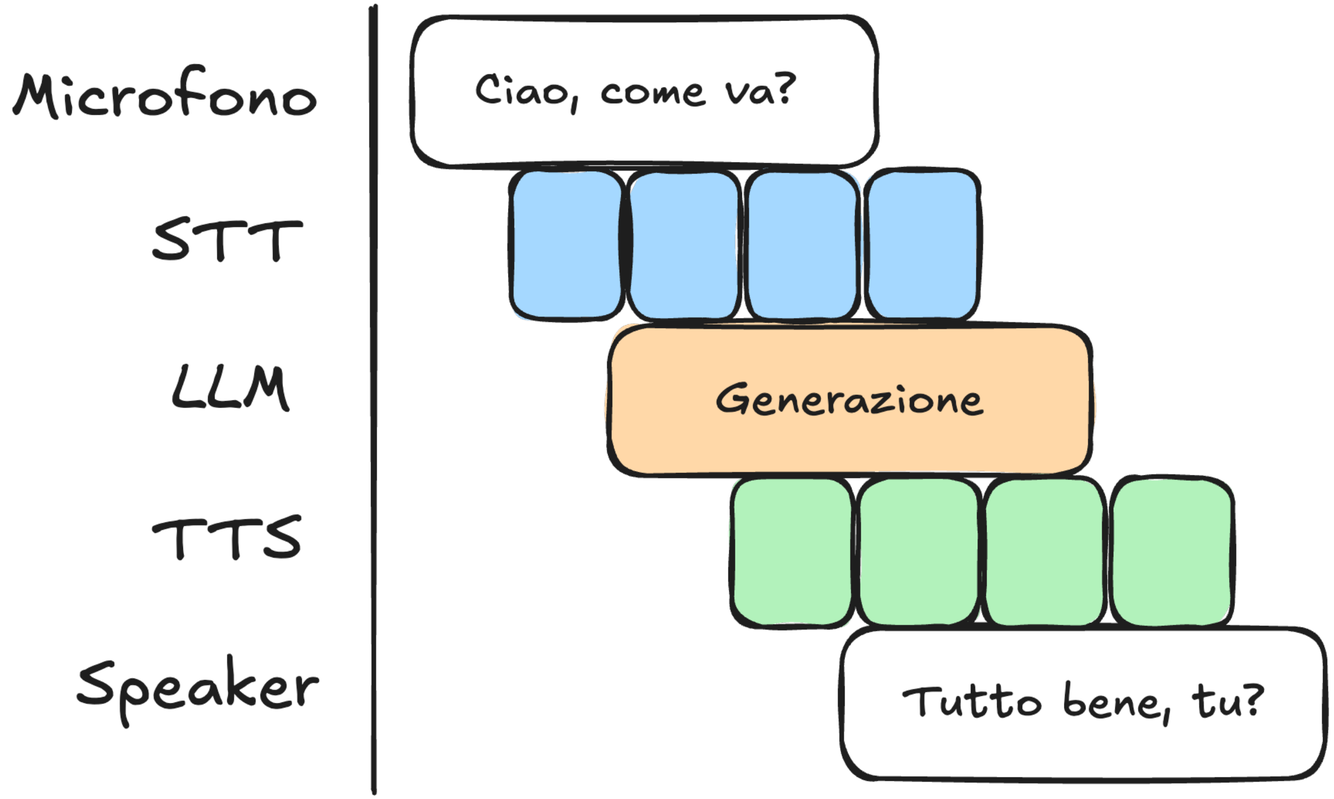
Pipeline STS



Pipeline STS a cascata



Pipeline STS streaming



OpenAI Realtime Protocol su WebSocket

- **Connessione persistente** tra client e API tramite WebSocket.
- **Comunicazione bidirezionale**: invio dell'audio in ingresso e ricezione dell'audio generato in tempo reale.
- **Protocollo a eventi**: messaggi JSON per controllare la sessione e trasmettere dati audio in chunk codificati in Base64.
- **Gestione del flusso audio**: invio progressivo dell'audio e ricezione della risposta senza attendere il completamento dell'intera elaborazione.
- **Bassa latenza**: interazione speech-to-speech senza dover orchestrare

HF: speech-to-speech

README Apache-2.0 license



Speech To Speech: Build voice agents with open-source models

GITHUB TRENDING #1 REPOSITORY OF THE DAY

pypi v1.0.0 python 3.10 | 3.11 | 3.12 license Apache 2.0

A low-latency voice agent built from open models that **runs entirely on your computer**. One command starts a conversation: no API key, no cloud service.

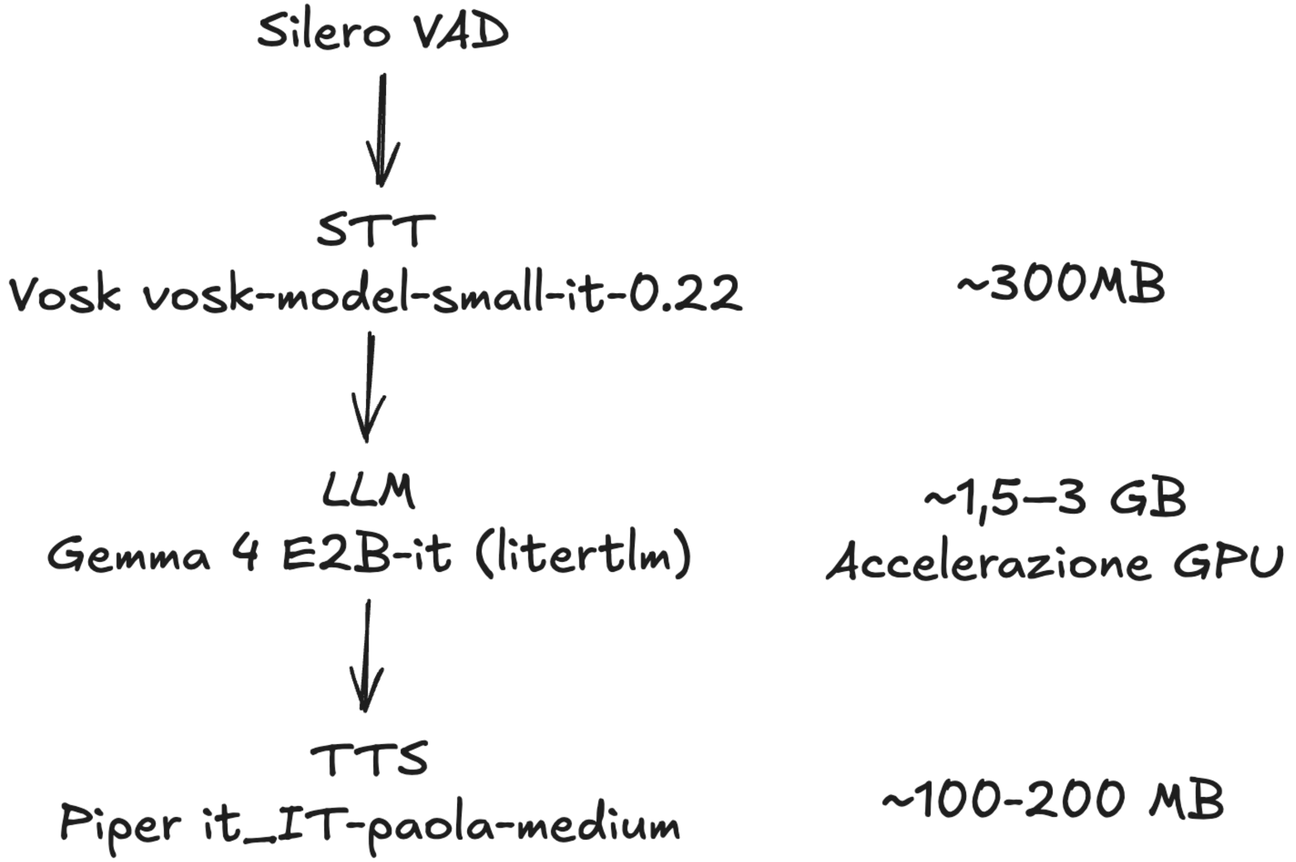
The default pipeline is **Silero VAD → Parakeet TDT → Qwen3.5-4B → Qwen3-TTS**, and every stage is swappable. The server speaks the **OpenAI Realtime protocol** over WebSocket and WebRTC, so apps built for OpenAI's Realtime API can point at your own machine instead.

Component	Backend	Languages
STT	Parakeet TDT (default)	25 European languages
STT	Whisper / Whisper MLX / Faster Whisper	Broad multilingual coverage, depending on the selected Whisper checkpoint
STT	Paraformer	Depends on the selected FunASR checkpoint; the default is Chinese-oriented
TTS	Qwen3-TTS (default)	Multilingual, with `--qwen3_tts_language auto` by default
TTS	Kokoro	Multiple language/voice mappings, depending on backend availability
TTS	KittenTTS	English
TTS	ChatTTS	English and Chinese
TTS	MMS TTS	Broad multilingual coverage through MMS checkpoints
TTS	OmniVoice	600+ languages; voice cloning, design, and automatic voice selection
TTS	Supertonic	32 languages
TTS	Pocket TTS	English, French, German, Portuguese, Italian, Spanish
TTS	OpenAI-compatible `/v1/audio/speech` endpoint	Depends on the connected TTS server/model

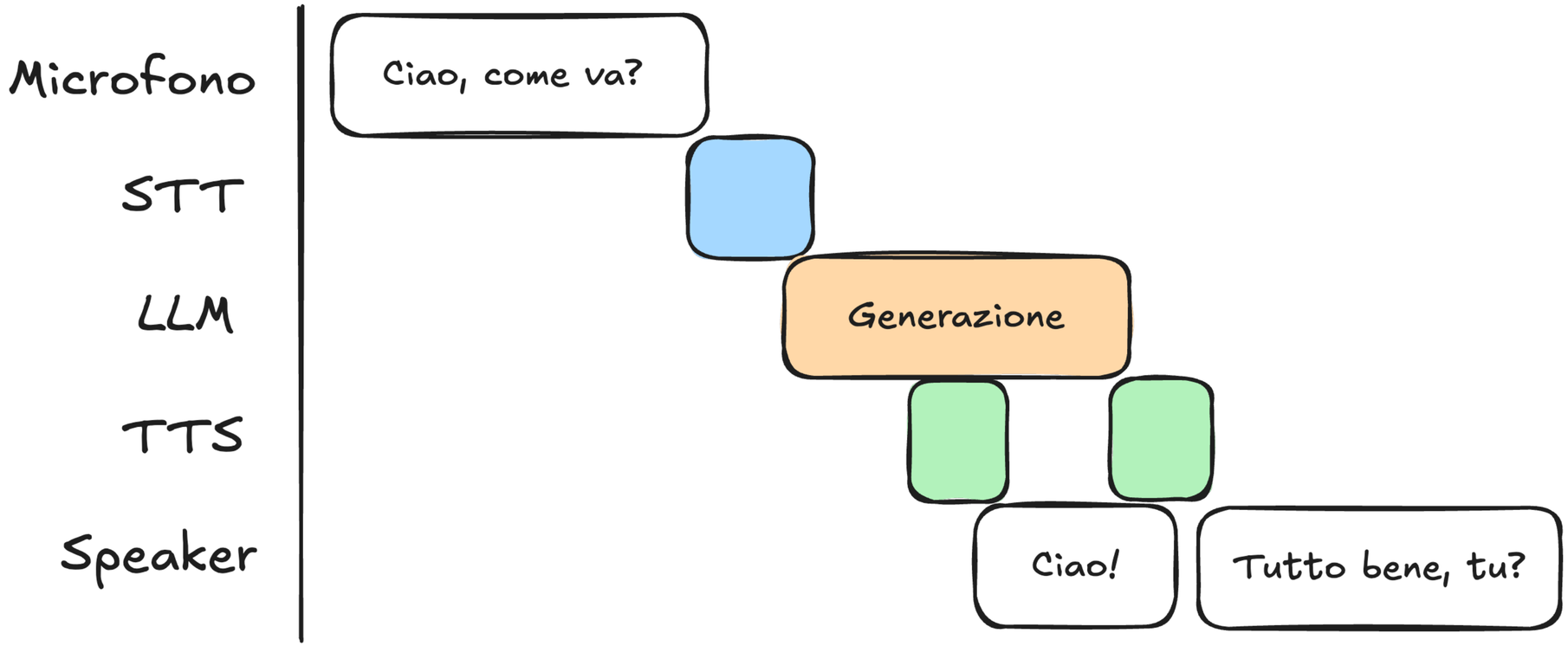
https://github.com/huggingface/speech-to-speech

Su mobile? Locale?

- Requisiti:
  - Realtime (o quasi)
  - Voce con pronuncia italiana (decente)
  - Footprint basso in memoria



Streaming Pipeline? Quasi



Possibili casi d’uso



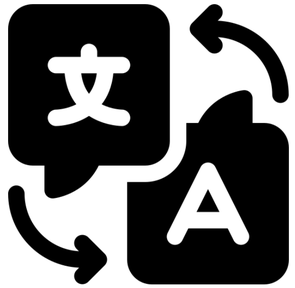
Accessibilità



Filtro anti-spam



Segreteria / Prenotazioni



Traduzioni realtime

5 Privacy ed etica



Privacy: cloud o locale?

API Cloud

Vantaggi

- Modelli potenti
- Meno risorse sul dispositivo

Trade-off

- I dati escono dal dispositivo
- Dipendenza dal provider e dalla connessione
- Verifica uso dei dati e condizioni del servizio

Locale

Vantaggi

- Dati elaborati offline
- Maggiore controllo sul trattamento
- Costo API azzerato

Trade-off

- Risorse hardware e consumi
- Qualità e latenza dei modelli

AI Act: articolo 50

Chi fornisce un sistema di AI che interagisce con le persone deve far sapere che stanno parlando con un'AI, salvo che sia evidente dal contesto.

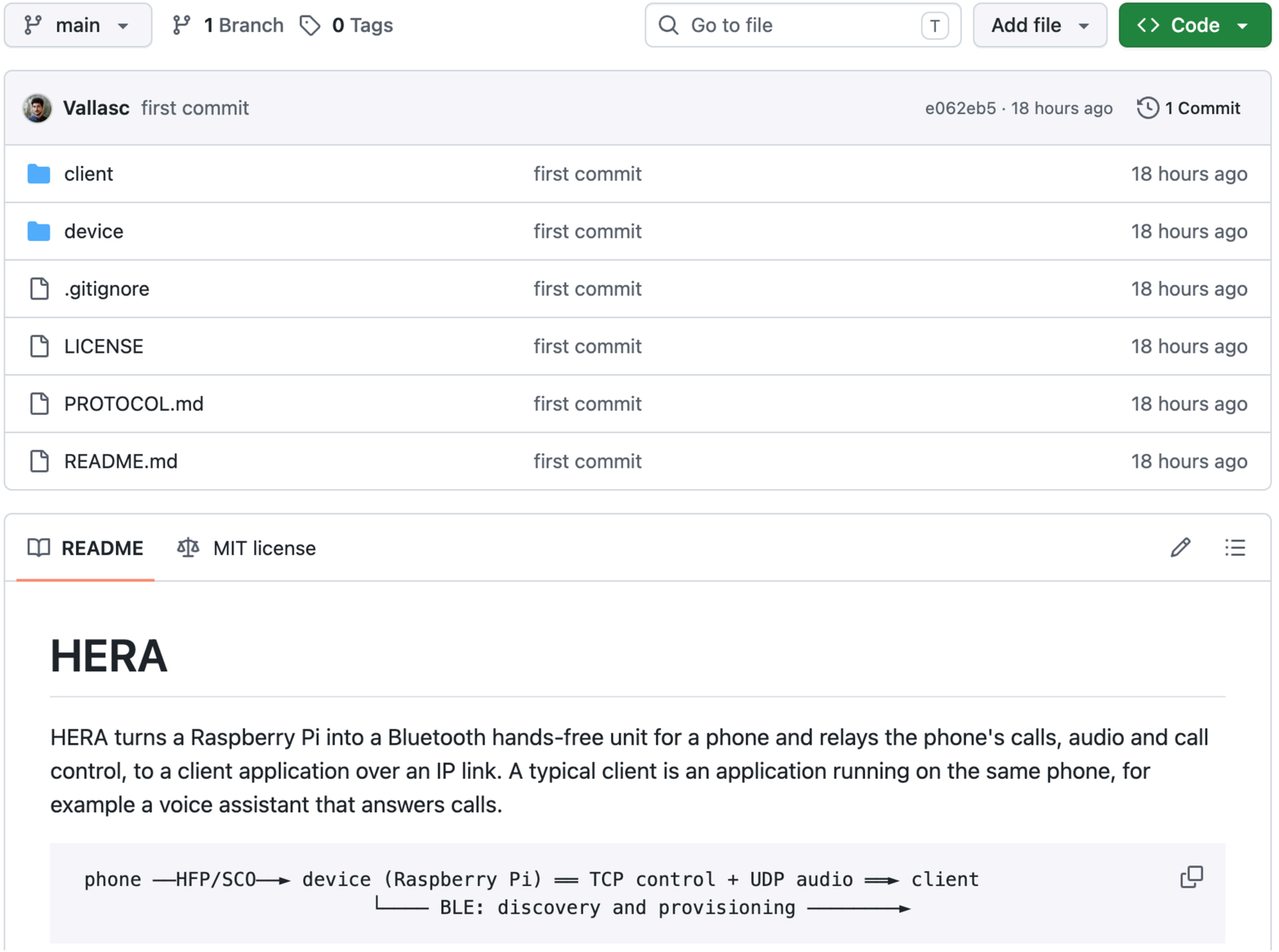
https://ai-act-service-desk.ec.europa.eu/it/ai-act/article-50

Stesso strumento, due usi

Utile	Pericoloso
Accessibilità	Truffe telefoniche (vishing)
Filtro per lo spam	Spam automatizzato su larga scala
Assistente personale	Impersonare qualcuno

HERA

Human-simulating Earbuds Realtime Answering



https://github.com/Vallasc/hera


Demo