LiveKit

Build realtime AI. Advanced Voice Mode with OpenAI and LiveKit.

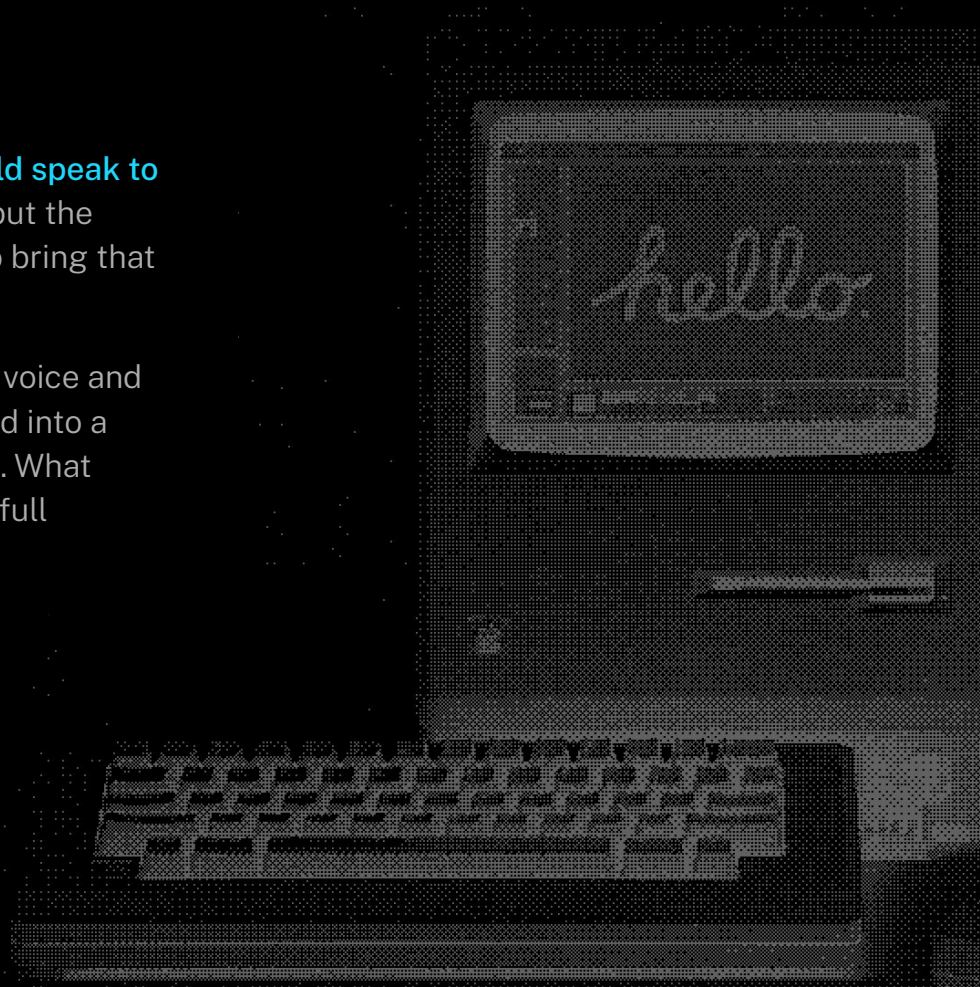
Shayne Parmelee / November 1st / Realtime Voice Workshop

livekit.io

The future of computing

Forty years ago, Steve Jobs unveiled **a computer that could speak to you**. The idea of a computer that adapts to us is not new, but the technology that makes it possible is. We started LiveKit to bring that technology to every developer in the world.

We began as an open source project for building realtime voice and video applications using WebRTC. Over time, we've evolved into a **global delivery network for any modality of realtime data**. What started with just a media server and some SDKs, is now a full ecosystem of APIs and tools for truly realtime computing.



github.com/livekit/livekit

Let’s see what we’re actually going to build

Agenda for today

1. Doesn’t OpenAI have a Websocket API? Why do we need LiveKit?
2. Creating our first agent with the CLI
3. Using the Developer Sandbox
4. Agent Architecture, and building your first frontend
5. Customizing your Agent
6. Building a frontend from scratch

Doesn’t OpenAI have a Websocket API? Why do we need LiveKit?

Doesn’t OpenAI have a Websocket API? Why do we need LiveKit?

WebRTC

Native media support

High level APIs for capture, built-in codecs for encoding and transmission

Decoding and rendering handled by native code on device — usually hardware accelerated.

UDP — Adaptive streaming and amazing error correction using native media codecs.

Websockets

No native media support

Manual capture, encoding, transmission

Manual reception, reassembly, decoding, rendering

TCP — Significant latency and throughput loss with any changing network conditions

Doesn’t OpenAI have a Websocket API? Why do we need LiveKit?

WebRTC

Native media support

High level APIs for capture, built-in codecs for encoding and transmission

Decoding and rendering handled by native code on device — usually hardware accelerated.

UDP — Adaptive streaming and amazing error correction using native media codecs.

LiveKit

Native media + metadata support

Client SDKs to run on almost any device

Server SDKs for room management and Agent framework for integration with most AI providers

Adaptive streaming via SFU

Doesn’t OpenAI have a Websocket API? Why do we need LiveKit?

TLDR

Websockets are great for many things, media isn’t one of them. When dealing with media, data integrity is less important than real-time delivery.

LiveKit builds on top of the benefits of WebRTC to provide an amazing, production ready developer experience.

Routing and load balancing with an SFU is hard, but LiveKit Cloud makes it trivial.

https://www.cs.utexas.edu/~lam/395t/papers/Mathis1998.pdf

Creating our first agent with the CLI

CLI Setup

Install the LiveKit CLI and test your setup using an example client application.

Install LiveKit CLI

MACOS LINUX WINDOWS FROM SOURCE

```
brew update && brew install livekit-cli
```

`lk` is LiveKit's suite of CLI utilities. It lets you conveniently access server APIs, create tokens, and generate test traffic all from your command line. For more details, refer to our docs in the `livekit-cli` GitHub repo ↗.

Authenticate with Cloud (optional)

For LiveKit Cloud users, you can authenticate the CLI with your Cloud project to create an API key and secret. This allows you to use the CLI without manually providing credentials each time.

```
lk cloud auth
```

Then, follow instructions and log in from a browser.

On this page

Creating our first agent with the CLI

LiveKit

- Analytics
- Sessions
- Egresses
- Ingresses
- Settings
- Sandbox BETA
- Billing

Overview (v2)
Sessions (v2)
Egresses (v2)
Ingresses (v2)

test-app-01-hp3nzyw3.livekit.cloud

FILTER BY LAST 24 HOURS

Test App 01

Get Started

Dismiss

Try sandbox BETA
The fastest way to test cloud apps

Agent guide ↗
Build your first agent in under 10 minutes

OpenAI speech-to-speech playground ↗
Try out our OpenAI realtime playground

Find your SDK quickstart ↗
Quick starts for most platforms

ⓘ The graphs and charts below will be visible once your project has collected enough data.

CONNECTION SUCCESS	CLIENT SDKS	CONNECTION TYPES
No data for this time range	No data for this time range	No data for this time range

Creating our first agent with the CLI

Test App 01 /

Sandbox

Templates

VOICE-ASSISTANT

Voice assistant frontend

A voice assistant showcasing realtime TTS, LLM, and STT

Create app

SERVER

Token server

A hosted token server to help you prototype your mobile applications...

Create app

VOICE-AND-VIDEO

Meet

An open source video conferencing app built on LiveKit Components,...

Create app

Your Sandbox apps

VOICE ASSISTANT FRONTEND

VOICE-ASSISTANT

augmented-container-mlvzma

Launch

Code

>_ Complete your local setup

```
lk app create \
    --sandbox augmented-container-mlvzma
```

Once you've set up the sandbox app locally, launch it to begin testing and interacting with the application.

Creating our first agent with the CLI

```
shayne@Shaynes-MBP ~ % lk app create --sandbox augmented-container-mlvzma
Using default project [Next Test]
  Select Template
  > multimodal-agent-python
    multimodal-agent-node
    voice-pipeline-agent-python

  Application Name
  > augmented-container-mlvzma

↑ up • ↓ down • / filter • enter select
```

Creating our first agent with the CLI

```
shayne@Shaynes-MBP realtime-workshop % lk app create \
        --sandbox augmented-container-mlvzma
Using default project [Next Test]
Cloning template...
Instantiating environment...
Cleaning up...
To setup and run the agent:

        cd /Users/shayne/Development/Livekit/Demos/realtime-workshop/augmented-container-mlvzma
        python3 -m venv venv
        source venv/bin/activate
        pip install -r requirements.txt
        python3 agent.py dev

shayne@Shaynes-MBP realtime-workshop %
```

Creating our first agent with the CLI

```
yio-4.6.2.post1 attrs-24.2.0 av-13.1.0 certifi-2024.8.30 cffi-1.17.1 click-8.1.7 distro-1.9.0 frozenlist-1.5.0 h11-
0.14.0 httpcore-1.0.6 httpx-0.27.2 idna-3.10 jiter-0.6.1 livekit-0.17.6 livekit-agents-0.10.2 livekit-api-0.7.1 liv
ekit-plugins-openai-0.10.4 livekit-protocol-0.6.0 multidict-6.1.0 openai-1.53.0 pillow-10.3.0 propcache-0.2.0 proto
buf-5.28.3 psutil-5.9.8 pycares-4.4.0 pycparser-2.22 pydantic-2.9.2 pydantic-core-2.23.4 pyjwt-2.9.0 python-dotenv-
1.0.1 sniffio-1.3.1 tqdm-4.66.6 types-protobuf-4.25.0.20240417 typing-extensions-4.12.2 watchfiles-0.24.0 yarl-1.17
.0

[notice] A new release of pip is available: 24.0 -> 24.3.1
[notice] To update, run: pip install --upgrade pip
2024-10-30 15:48:50,178 - DEBUG asyncio - Using selector: KqueueSelector
2024-10-30 15:48:50,179 - DEV  livekit.agents - Watching /Users/shayne/Development/Livekit/Demos/realtime-workshop/
augmented-container-mlvzma
2024-10-30 15:48:50,518 - DEBUG asyncio - Using selector: KqueueSelector
2024-10-30 15:48:50,521 - INFO livekit.agents - starting worker {"version": "0.10.2", "rtc-version": "0.17.6"}
2024-10-30 15:48:50,698 - INFO livekit.agents - registered worker {"id": "AW_2s56mVgdLXdu", "region": "US Central",
 "protocol": 15, "node_id": "NC_OCHICAGO1A_KCCwpn3Mac5Z"}
```

Using the Sandbox

Test App 01 /

Sandbox

Templates

VOICE-ASSISTANT

Voice assistant frontend

A voice assistant showcasing realtime TTS, LLM, and STT

Create app

SERVER

Token server

A hosted token server to help you prototype your mobile applications...

Create app

VOICE-AND-VIDEO

Meet

An open source video conferencing app built on LiveKit Components,...

Create app

Your Sandbox apps

VOICE ASSISTANT FRONTEND

VOICE-ASSISTANT

augmented-container-mlvzma

Launch

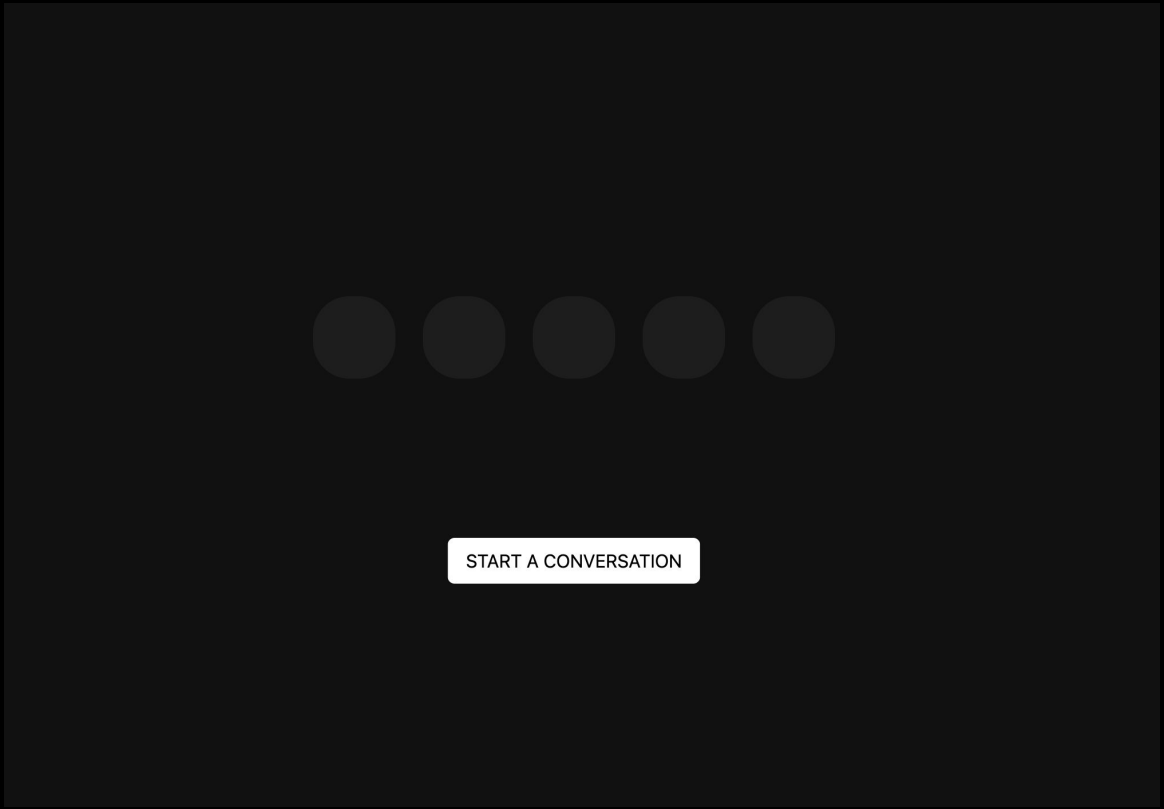
Code

>_ Complete your local setup

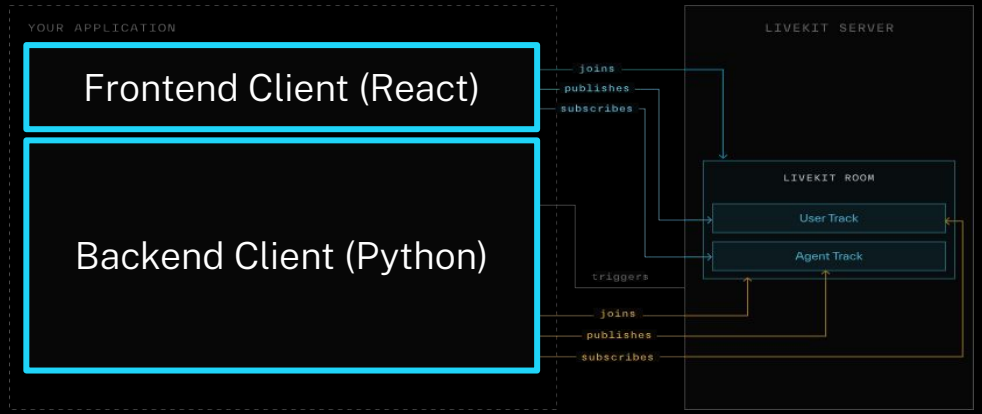
```
lk app create \
    --sandbox augmented-container-mlvzma
```

Once you've set up the sandbox app locally, launch it to begin testing and interacting with the application.

Using the Sandbox



Agent architecture, and building your first frontend



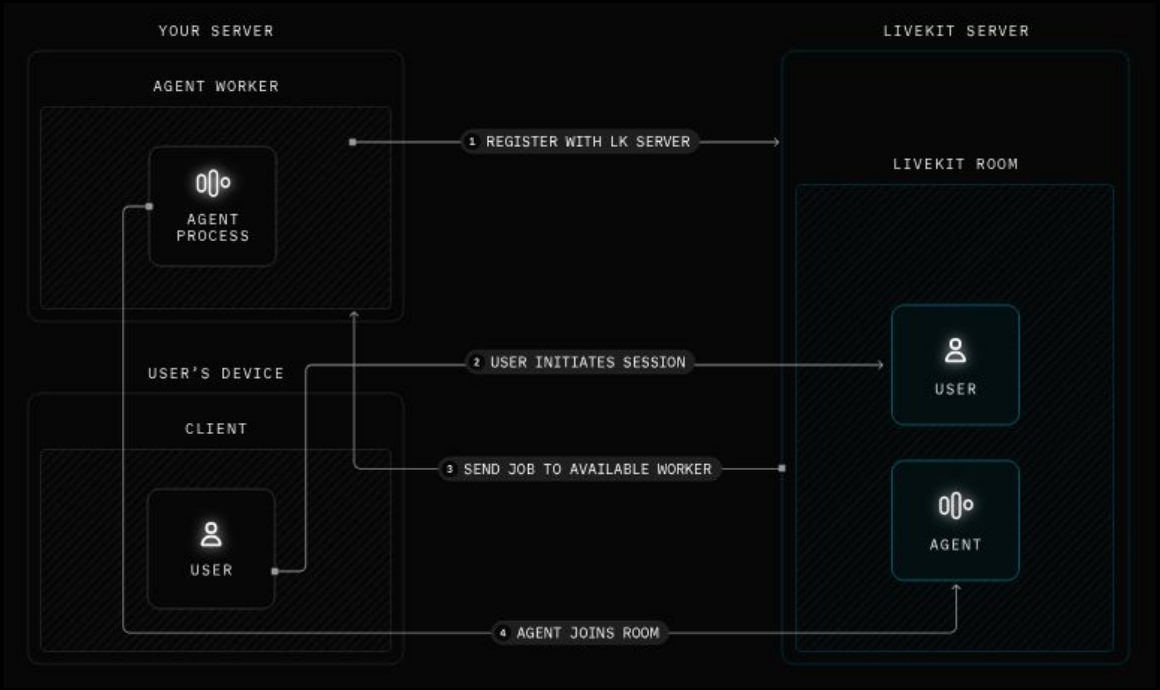
1. `Client` joins `Room`, publishes audio
2. `Agent` joins `Room`, subscribes to `Client` audio and publishes its own.
3. `Client` subscribes to `Agent` audio

Now we have one track of audio being streamed by each of the `Agent` and `Client`. These tracks are published to a `Room`.

Agent architecture, and building your first frontend

Agent architecture, and building your first frontend

Agent architecture, and building your first frontend



99.99%

Uptime

Agent architecture, and building your first frontend

```
lk app create --template voice-assistant-frontend
```

Customizing your agent

Building your frontend from scratch

Ready to build?

Start building your realtime application with a free account.
No credit card required • 50GB free monthly