

Comprehensive Setup Guide: Integrating a SIP Client with a Self-Hosted LiveKit Environment

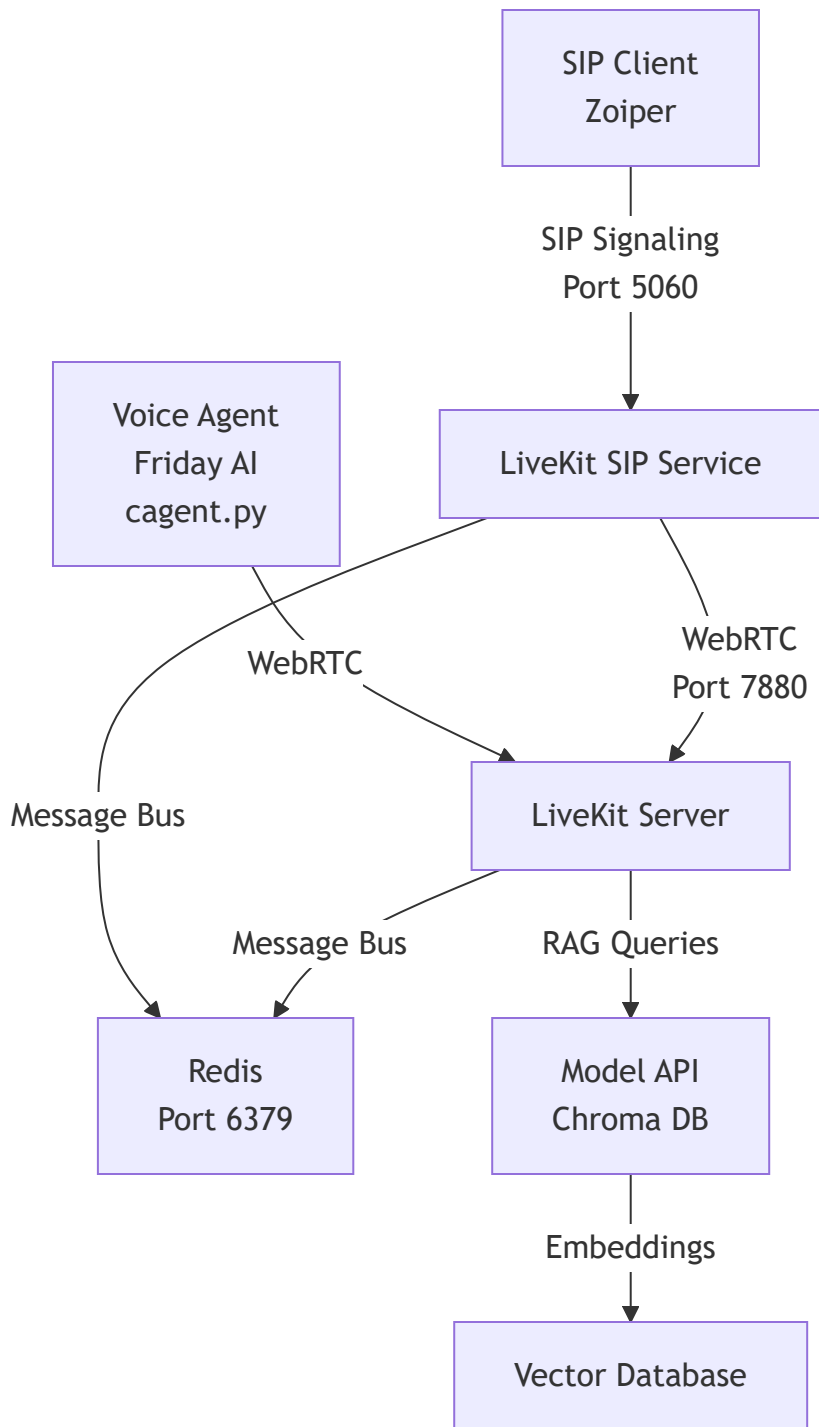
1 Introduction & Architecture

This document provides a detailed, step-by-step guide for setting up a self-hosted LiveKit environment to accept calls from a SIP client (like Zoiper). This setup bridges traditional telephony with modern WebRTC, enabling real-time voice applications.

The architecture consists of several key components that must be configured correctly to communicate with each other:

- **LiveKit Server:** The core WebRTC service that manages rooms, participants, and media tracks. It acts as the central hub for all real-time communication.
- **LiveKit SIP Service:** A separate application that acts as a bridge. It receives calls via the SIP protocol and translates them into a format the LiveKit Server can understand.
- **Redis:** An in-memory data store that serves as the critical message bus between the LiveKit Server and the SIP Service. If they cannot both connect to Redis, the system will fail.
- **lk CLI:** The command-line interface used to manage LiveKit resources like SIP Trunks and Dispatch Rules.
- **Zoiper:** The SIP softphone client used to place test calls into the system.

Architecture Diagram



2 Core Server Configuration

The first step is to ensure both the main server and the SIP service are configured correctly. A mismatch in this foundational step was the source of several issues.

Step 2.1: Configure the LiveKit Server (livekit.yaml)

The main livekit-server must be explicitly configured to connect to Redis. Without this, it cannot communicate with the SIP service, leading to "sip not connected" errors.

1. **Create the configuration file:** In your project root
(/mnt/c/Users/int10281/Desktop/Github/Friday - Copy/), create a file named `livekit.yaml` .
2. **Add the following content:** This configuration defines the server's port, API keys, and the crucial Redis connection.

```
# livekit.yaml

port: 7880
bind_addresses:
  - 0.0.0.0
keys:
  APIntavBoHTqApw: pRkd16t4uYVUs9nSlNeMawSE1qmUzfV2ZkSrMT2aiFM

# This section is required for SIP functionality
redis:
  address: '192.168.109.66:6379'
  db: 0
```

Step 2.2: Configure the SIP Service (config.yaml)

The livekit-sip service needs its own configuration to connect to both Redis and the main server's API. The API keys here **must be identical** to those in `livekit.yaml` .

1. **Navigate to your SIP directory:**

```
cd '/mnt/c/Users/int10281/Desktop/Github/Friday - Copy/sip'
```

2. **Create/Edit the configuration file:** Ensure the file `config.yaml` has the following content. This version is cleaned of any hidden formatting errors and uses the modern Trunk-based approach (the legacy users block has been removed).

```
# config.yaml (Final, Cleaned Version)

api_key: APIntavBoHTqApw
api_secret: pRkd16t4uYVUs9nSlNeMawSE1qmUzfV2ZkSrMT2aiFM
ws_url: ws://192.168.109.66:7880
redis:
  address: 192.168.109.66:6379
sip_port: 5060
rtp_port: 10000-20000
use_external_ip: false
logging:
  level: debug
```

3 Running and Verifying the Services

The services must be started in the correct order.

1. **Start Redis:** Ensure your Redis server is running.

- **Verification Command:**

```
redis-cli ping
```

- **Expected Output:** PONG

2. **Start the LiveKit Server:**

- **Path:** /mnt/c/Users/int10281/Desktop/Github/Friday - Copy/
- **Start Command:**

```
livekit-server --config livekit.yaml
```

- **Verification:** Check the logs for "connecting to redis" and "starting LiveKit server".

3. **Start the SIP Service:**

- **Path:** /mnt/c/Users/int10281/Desktop/Github/Friday - Copy/sip/
- **Start Command:**

```
./livekit-sip --config config.yaml
```

- **Verification:** Check the logs for "connecting to redis" and "sip signaling listening on".

4 Creating the SIP Trunk and Dispatch Rule

Now, we'll configure how the system handles incoming calls using the lk CLI.

Step 4.1: Configure the lk CLI Project

The CLI needs to know which project to work with.

1. Add the project:

```
lk project add friday --url ws://192.168.109.66:7880 --api-key APIntavBoHTqApw --api-secret pRkc
```

2. Set the default project:

```
lk project set-default friday
```

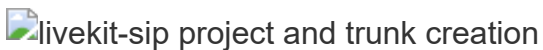
3. Verification Command:

```
lk project list
```

- **Expected Output:** A table showing the `friday` project as the default (marked with an asterisk).

Step 4.2: Create the Inbound SIP Trunk

LiveKit-sip project and trunk creation

livekit-sip project and trunk creation

This trunk defines the credentials Zoiper will use to authenticate.

1. Create the file `inbound_trunk.json` :

- **Path:** `/mnt/c/Users/int10281/Desktop/Github/Friday - Copy/sip/`
- **Content:**

```
{
  "trunk": {
    "name": "Zoiper Local Inbound",
    "auth_username": "1001",
    "auth_password": "1001"
  }
}
```

2. Run the creation command:

```
lk sip inbound create --project friday inbound_trunk.json
```

- **Expected Output:** SIPTrunkID: ST_EmMmjttJHZHP (Your ID will be different). **Save this ID.**

3. Verification Command:

```
lk sip inbound-trunk list
```

- **Expected Output:** A table listing your newly created "Zoiper Local Inbound" trunk.

Step 4.3: Create the SIP Dispatch Rule

This rule routes authenticated calls from the trunk to a specific LiveKit room.

1. Create the file sip_dispatch.json :

- **Path:** /mnt/c/Users/int10281/Desktop/Github/Friday - Copy/sip/
- **Content:** Replace ST_EmMmjttJHZHP with the Trunk ID you saved.

```
{
  "dispatch_rule": {
    "name": "Zoiper to Assistant Room Rule",
    "trunk_ids": ["ST_EmMmjttJHZHP"],
    "rule": {
      "dispatchRuleDirect": {
        "roomName": "friday-assistant-room"
      }
    }
  }
}
```

2. Run the creation command:

```
lk sip dispatch create --project friday sip_dispatch.json
```

- **Expected Output:** SIPDispatchRuleID: SDR_z8poRnoYDZPB (Your ID will be different).

5 SIP Client (Zoiper) Configuration

Configure your Zoiper client to register with your livekit-sip service.

1. Go to **Settings** → **Accounts** → **Add Account** and choose **Manual Configuration** for a **SIP** account.
2. Enter the following details:
 - **Host / Domain:** 192.168.109.66
 - **Username:** 1001
 - **Password:** 1001
3. Save the account.
4. **Verification:** Look for a green checkmark or a "Registered" status next to the account in Zoiper. You should also see Inbound SIP REGISTER messages in your livekit-sip service logs.

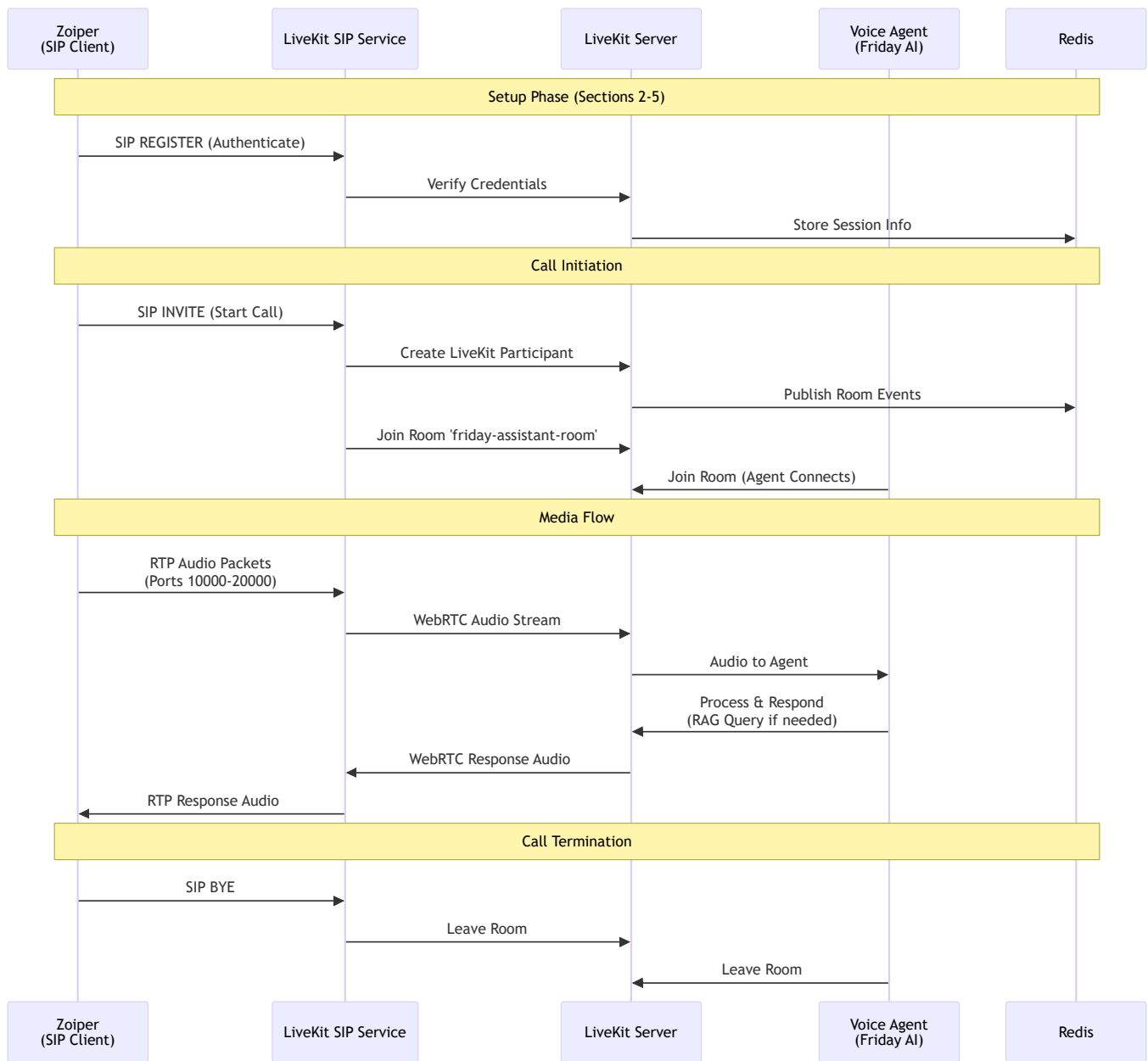
6 End-to-End Test

With all services running and Zoiper registered, dial any number from the Zoiper client. The call should connect, and you will be placed into the `friday-assistant-room`.

- **Verification:**
 - The livekit-sip logs will show "SIP invite authentication successful" followed by "Joining room".
 - The livekit-server logs will show events for a new participant joining the `friday-assistant-room`.

Once the call is connected, ensure the voice agent is running to handle the conversation. Run `python cagent.py` in the project root to start the Friday AI voice assistant.

Call Flow Diagram



7 Troubleshooting Common Issues

- **Error: twirp error... sip not connected (redis required)**
 - **Cause:** The main livekit-server is not connected to Redis.
 - **Solution:** Ensure your `livekit.yaml` has the correct `redis` block and that you are starting the server with `livekit-server --config livekit.yaml`.
- **Error: Cannot create LiveKit participant... "error": "missing API key or secret key"**

- **Cause:** The API key/secret in `config.yaml` does not match `livekit.yaml` , often due to a typo or a hidden formatting error.
- **Solution:** Stop both services. Carefully copy the keys from `livekit.yaml` and paste them into the `livekit` block in `config.yaml` . Use the cleaned `config.yaml` from this guide to avoid formatting issues.
- **Zoiper Fails to Register**
 - **Cause:** Firewall blocking port 5060, incorrect IP in Zoiper's "Host" field, or the `livekit-sip` service is not running.
 - **Solution:** Verify the host IP is `192.168.109.66` . Check that the `livekit-sip` process is active and listening. Ensure no firewalls are blocking UDP/TCP traffic on port 5060.
- **Call session established but voice not transmitted from caller to bot or bot not listening**
 - **Cause:** The voice agent (Friday AI) is not running or not connected to the LiveKit room.
 - **Solution:**
 - a. Start the voice agent: In the project root
(`/mnt/c/Users/int10281/Desktop/Github/Friday - Copy/`), run `python cagent.py` .
 - b. Verify the agent is in the room: Check LiveKit server logs or use CLI:

```
lk room participants --room friday-assistant-room
```

You should see the SIP participant and the agent participant.

- c. Check agent logs for any connection errors.
- **Cause:** RTP media ports blocked by firewall.
 - **Solution:** Ensure UDP ports 10000-20000 are open and the `livekit-sip` service can bind to them. Use tools like `sngrep` to monitor SIP traffic and check for RTP packets.

8 Example call and generated lead (validation)

Below is an example call transcript taken from a real test session and the lead JSON that the assistant created. Use this as a validation reference to confirm the end-to-end flow (SIP -> `livekit-sip` -> LiveKit -> voice agent -> lead storage).

Example conversation excerpt

Agent: Namaste! Main Triotech ki Sales Assistant hoon. Main aapki kis tarah help kar sakti hoon?

User: Hello?

Agent: Ji, hello! Main Triotech ki Sales Assistant hoon. Main aapki kis tarah help kar sakti hoon?

User: आप मेरे को try tech के products के बारे में बता सकते हैं try

Agent: Maaf kijiye, abhi is query ke liye exact products ki list available nahi hai. Lekin, main

User: रिंगिंगों के बारे में जानना है.

Agent: Ringingo ek AI Dialer hai jo automated call handling, smart routing, aur real-time metrics

User: नहीं मेरे को sales team के साथ connect कर दीजिए please.

Agent: Ji, bilkul! Sales team se connect karne ke liye, kya main aapka naam, email, company ka r

User: मेरा नाम मलय जैन है. मेरा company का नाम

Agent: Shukriya, Malay Jain ji! Aapki details note kar li hain: Name: Malay Jain, Company: 3inov

User (spoken): My phone number is six two three two one, double five, triple eight.

Agent: Shukriya, Malay Jain ji! Aapki complete details ab mere paas hain: Name: Malay Jain, Ema:

Agent: Shukriya! Aapki details hamari sales team ke saath share kar di gayi hain. Hamari sales t

Generated lead (leads/lead_20251007_164608.json)

```
{  
  "timestamp": "2025-10-07T16:46:08.546889",  
  "source": "Friday AI Assistant",  
  "status": "new",  
  "name": "Malay Jain",  
  "email": "malayjain1234@gmail.com",  
  "company": "3inovi",  
  "interest": "Ringingo",  
  "phone": "6232155888",  
  "job_title": "",  
  "budget": "4-5 Lakhs",  
  "timeline": ""  
}
```

sngrep capture

 sngrep capture

Works cited

1. How to set SIP/IAX2 account with Zoiper - Frequently Asked Questions :: Zoiper, accessed October 6, 2025,
<https://www.zoiper.com/en/support/home/article/192/How%20to%20set%20SIP%2FIAX2%20account%20with%20Zoiper>