Calling App - High Level Design

Prioritized requirements

1. Allow users to call each other over **VoIP** or **PSTN**
2. Route call from one user to another and allowing government bodies to record suspicious call
3. Charge users for making calls.
4. Choose a provider for each call.

Requirements not a part of our list

1. Authentication service
2. Allow users to send messages

Capacity Estimation

1. Bandwidth required to handle calls.

- **Assumption** Number of users = **10 million** Number of users making call every day = **25%** Average length of each call = **1 minute** Bandwidth for each PSTN call = **64kbps** Bandwidth for each VoIP call = **100kbps**

- Number of calls per day = 2.5 million calls
- Number of minutes for all calls = 2.5 million minutes
- Number of minutes in a day = 24*60 = **1440**
- Number of calls at any moment = (Total talk time)/(Number of minutes in a day) = 1736 calls ~ **2000 calls**
- Minimum bandwidth for PSTN calls = Number of calls at any moment * Bandwidth of each PSTN call = **128 Mbps**
- For peak hours, bandwidth will be = 128*3 ~ **400Mbps**
- Minimum bandwidth for VoIP calls = Number of calls at any point * Bandwidth of each VoIP call = **200 Mbps**
- To reduce network congestion during peak hours, required bandwidth will be = 200*4 ~ **800Mbps**

Despite VoIP having twice the requirement of the PSTN bandwidth, the VoIP lines are cheaper

2.Total storage required to record calls

- **Assumption** Number of users = **10 million** Number of users making call every day = **25%** Average length of each call = **1 minute** Percentage of suspicious calls = **0.01%** Number of copies for fault

tolerance = **3** Life of each recording = **10 years** Size of 1 minute of audio file = **10 MB**

- Number of calls to be recorded = 0.0001 * 0.25 * 10,000,000 = **250 calls**
- Total storage required = 2.5Gb * 3 * 365 * 10 ~ **30 TB**

Total cache storage required for caching call states

- **Assumption** Number of fields per call = **20** Length of each field = **20 characters** Size of each character = **1 byte** During crisis call volume can become 50 times the normal volume.

- Number of characters for each call = **400 characters**
- Storage required for all calls = 2000 * 400 ~ **1 MB**
- Storage required during crisis = 50 * 1 MB = **50 MB** (*which is negligible*)

Requirement 1: Allowing users to make call over VoIP or PSTN

We have to implement a system that allows user to make calls over both VoIP and PSTN. We also want to separate the business logic from the components that connect the sender and receiver

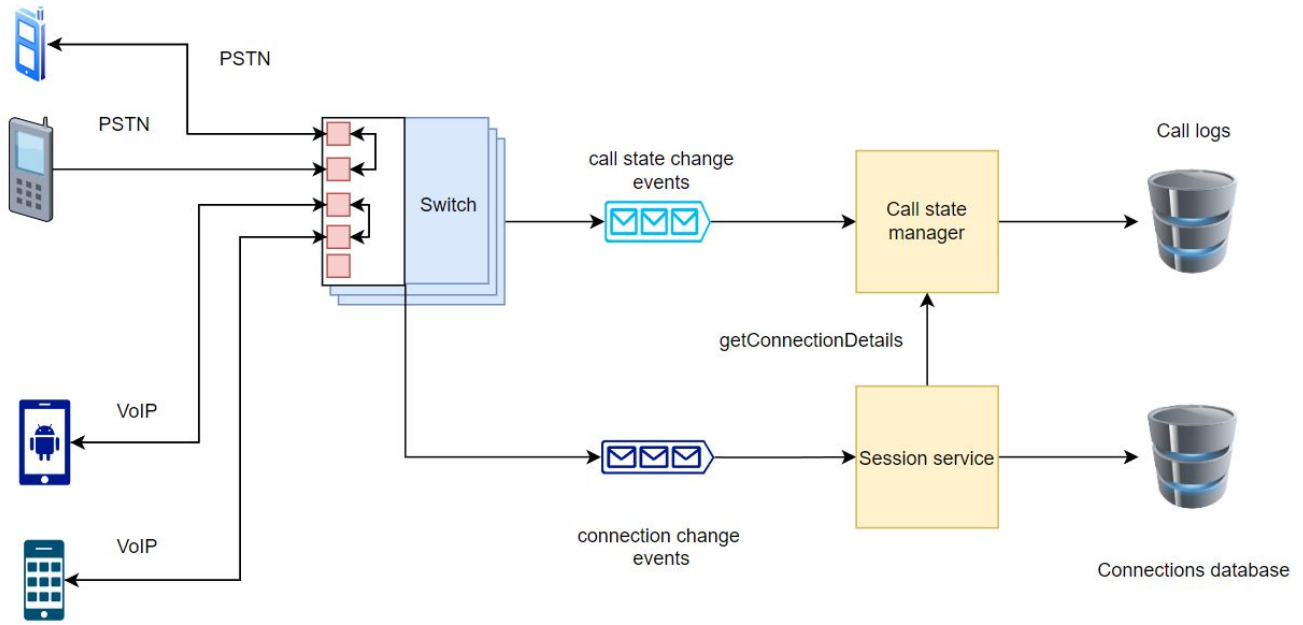
Components required

- **Switch** Whenever there is a call we get a receiver address and sender address. Switch bridges the connection between these two addresses.
- **Message queue**
  We need two message queues.
  - First, whenever there is a change in call state (like user has picked up call or a call is requested) we push it to call state change queue. Call manager subscribes to this message queue and then processes the events one by one.
  - Secondly, whenever a connection is created or terminated we push it to Connection Event Queue.
- **Call state manager**
  This component is the brain of our system. It receives the call event from the switch .It receives the connection ID of both users from the session service and tells the switch which two connections it has to bridge. Apart from that it also checks what is the current state of call. Like, wether the call is terminated or is ongoing. This data is then provided to billing service.
- **Session service** It pulls call events from the connection event queue. It stores the mapping of user to call session. And each session is mapped to a connection.

Comparison between VoIP and PSTN

VoIP	PSTN
Only internet connections is required to make calls	Telephone lines are required to make calls
It requires variable bandwidth	It reserves fixed bandwidth
Cost is not dependent on distance and time	Cost is dependent on distance and time
It is generally free to use.	It is a paid service

Diagram

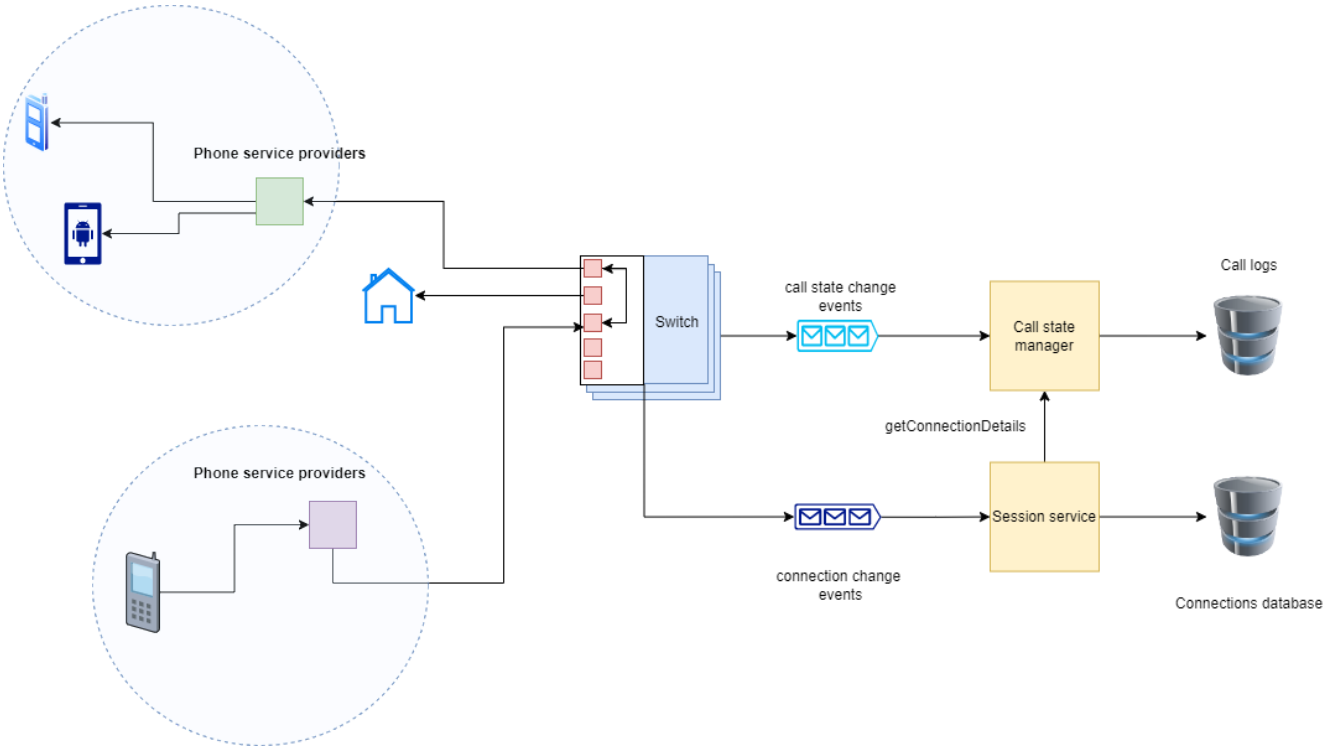


Requirement 2: Routing call from one user to another and allowing government bodies to record suspicious calls

When making a call sender has the address of the receiver. We need to route calls from sender to receiver. To do that there are phone service providers. When users make a call routers forward the call to the next router and so on. This process is repeated until the receiver is found. Our server should allow only authenticated users to make calls and also route calls from sender to receiver.

To record suspicious calls our system should opne a separate connection to them.

Diagrams



Requirement 3: Charging users for making calls

We need to implement a system that checks the current balance and the maximum talk time of user. If users exceed their maximum talk time then the call should be terminated.

We also need to generate invoice for calls made by user.

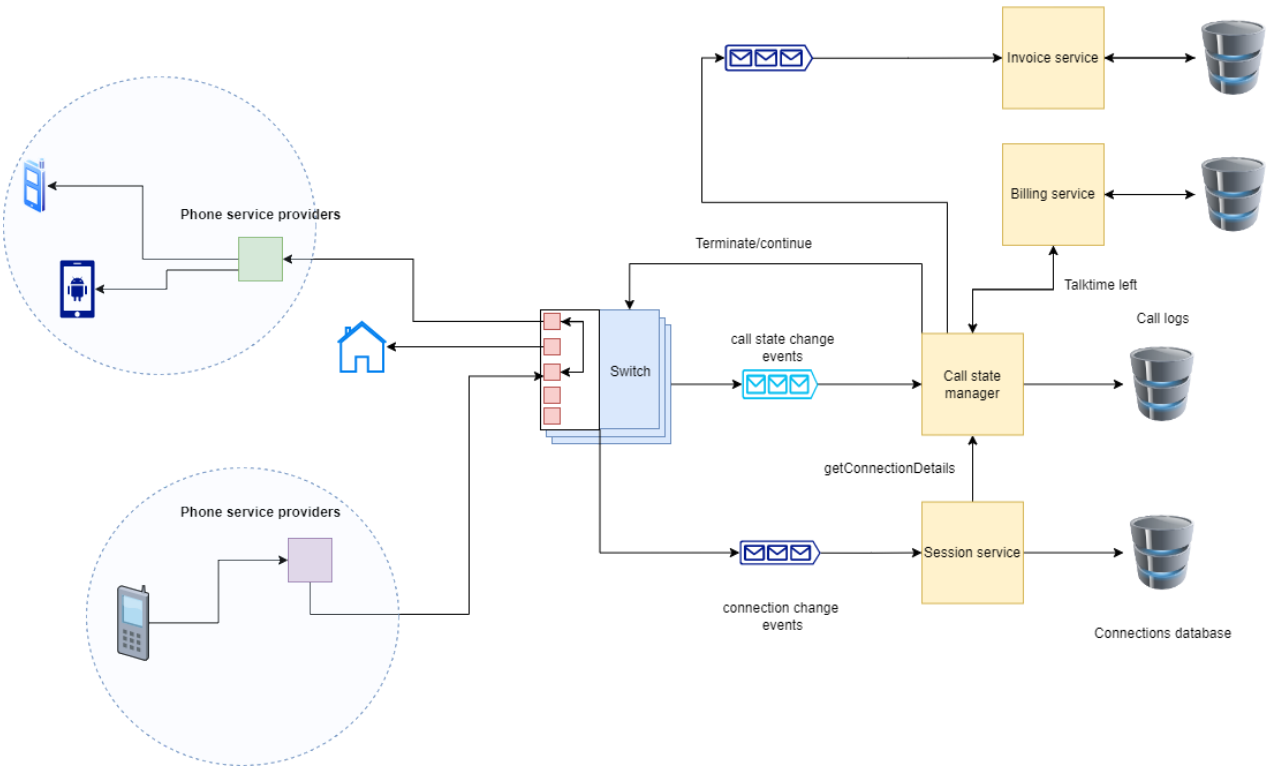
Trade-offs

- **Persisting user data in switch v/s Persisting data in Call state manager**
  If we persist data on switch we will require less API call because if user exceeds the time limit then switch can directly terminate the call. However switch has to take up multiple roles.
  If we persist data on call state manager we will require more API calls. Switch first asks CSM wether to continue call or terminate it and CSM asks the same question to billing service.
  It makes more sense to go with the second approach because CSM is the brain of our system and it makes sense to let it handle the billing and termination. Also we put all data in switch then it has to take up multiple roles and we do not want that. Switch should only perform one task i.e. bridging and terminating calls.

Components required

- **Billing service**
  It keeps current balance and talk time left of a user. It then sends the data to call state manager.
- **Invoice service**
  Once the call is done we want to generate an invoice for that we need an invoice service. It gets all the data (like call duration and cost per minute) from the call state manager and makes an invoice.
- **Message queue**
  To generate an invoice we push call event to this message queue and the invoice service subscribes to this message queue.
  We will push at least one event to this queue but each event will have an unique ID. If the key is already present in invoice service that means we have already generated the invoice. We do this to ensure user is charged only once per call

  Diagram



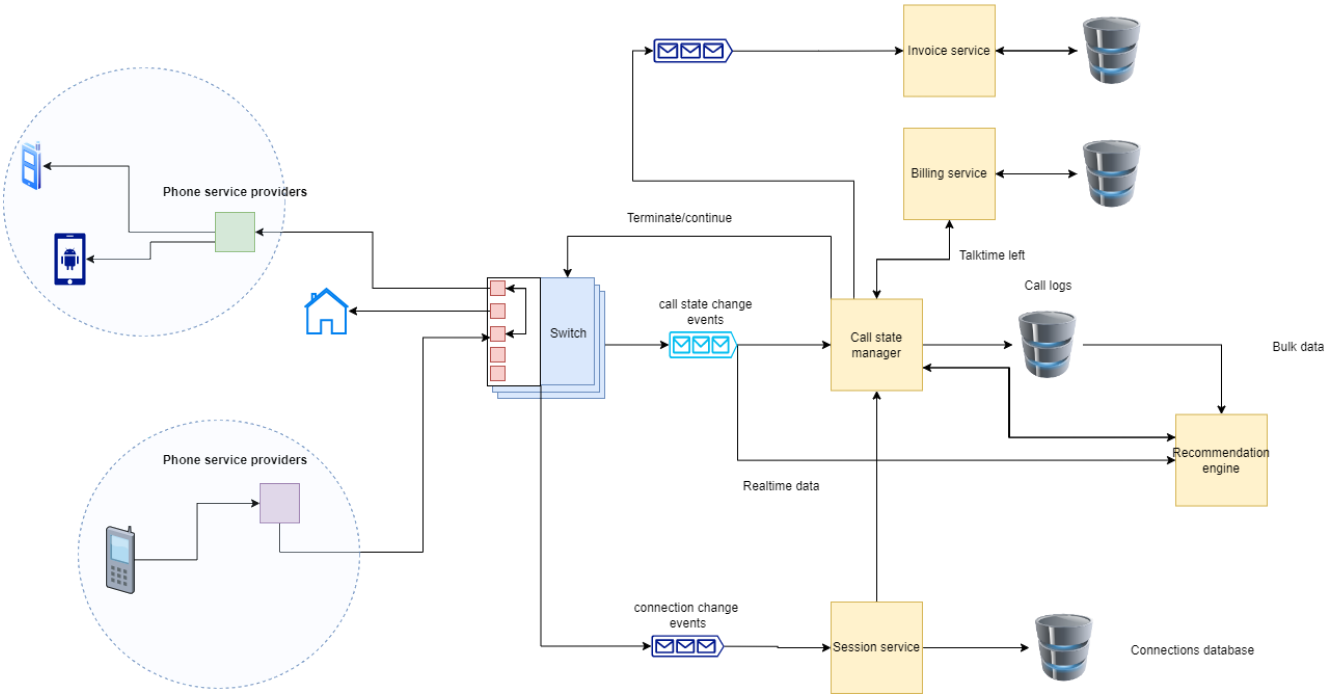
Requirement 4: Choosing right service provider for each call

We need to consider two factors when choosing the right service provider. Call quality and service provider fee. Depending on what our needs are we need to choose the best service provider.

Components required

- **Recommendation service**
  It derives insights from the previous data and based on that finds a service provider with best call quality and minimum fee. It gets bulk data from call logs. It also subscribes to the call state change queue to get real time events (*This architecture is called lambda architecture*).

Diagram



API Contracts

Session Service

- Session createSession(userID, connectionID)
- Session[] getSessions(user_id)
- void logoutSession(session_id)

Call State Manager

- Call startCall(User from , User to)
- Bool continueCall(call)

Switch

- Call startCall(callStateMachine)

Invoice service

- Invoice createInvoice(callID, talkTime, currency, pricePerMinute)

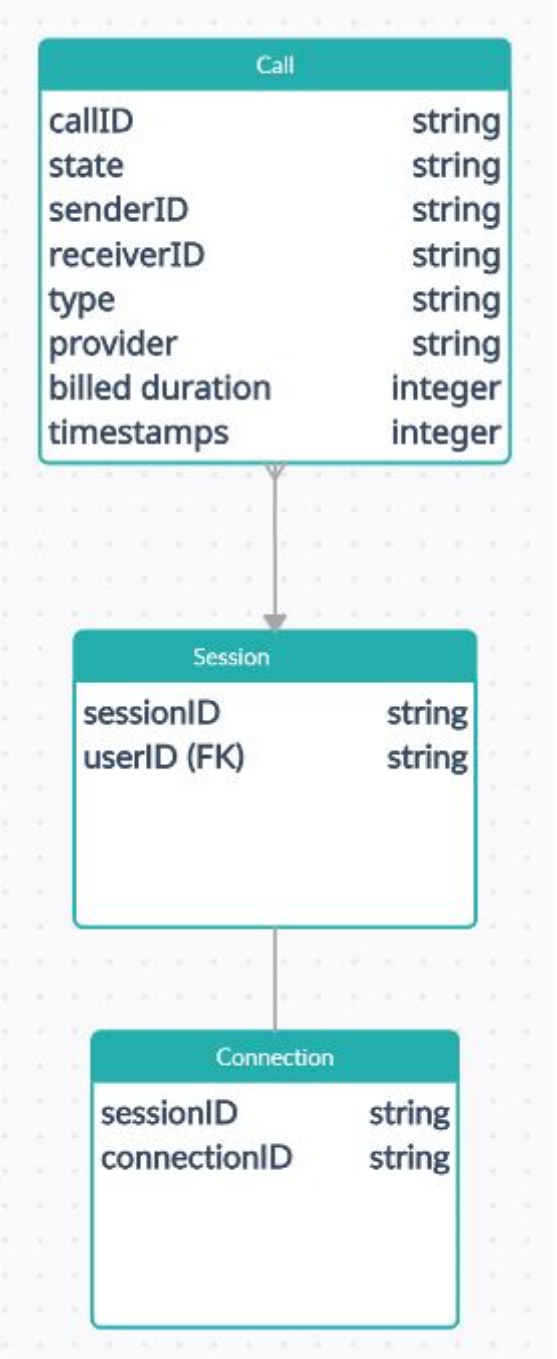
Billing service

- void addBalance(userID, amount)
- getBalance(userID)

Recommendation engine

- ServiceProvider getRoute(user sender, user receiver)

Database design



That's it for now!

You can check out more designs at InterviewReady.