ecosmob
makes your vision a reality

8 Ventures

Scope of Work Document

Date: 14/10/2024

ecosmob
makes your vision a reality

www.ecosmob.com
sales@ecosmob.com

Table of Contents

Project Summary

Client name

Crystal Net Pte Ltd

Objectives

1. **Click To Call** : To make a middleware gateway (For click to call) which will connect FreePBX services to the Cloud/CRM panel in which when the user clicks on the client number, It will connect to the agent/extension first and then the customer.

2. **Call Event - Webhook Notification**: To make a solution of webhook notifications (Pop-ups) which will be integrated in the CRM cloud panel. The user can see the call events like (Inbound Call Ringing, Started(Answered) and Call Ended). Primarily it was for inbound calls, during brainstorming meet - It was identified that it will be required for all the call events.

3. **Centralized database for CDR's**: Synchronize CDR data from Asterisk AMI to Kafka Message Broker, then to a centralized database(DBMS), enabling retrieval of CDR information via APIs for the CRM cloud panel.

4. **Storing Call Recordings**: Call Recordings from FreePBX servers will be uploaded on AWS S3 in specific folder structure against Domain/VM, CRM will get the Call Recordings from S3 using which admin can play the Call Recording files on the cloud panel.

5. **Custom Call Status Updates**: Custom changes to be made in the Call Status of CDR's (In Centralized Database) or Additional column to be added (Ex. Success, Rejected, Unavailable, Busy etc). Also Custom Call

status (In Centralized Database) in case of call events happens with Queue Agents

Solutions Overview

Click To Call

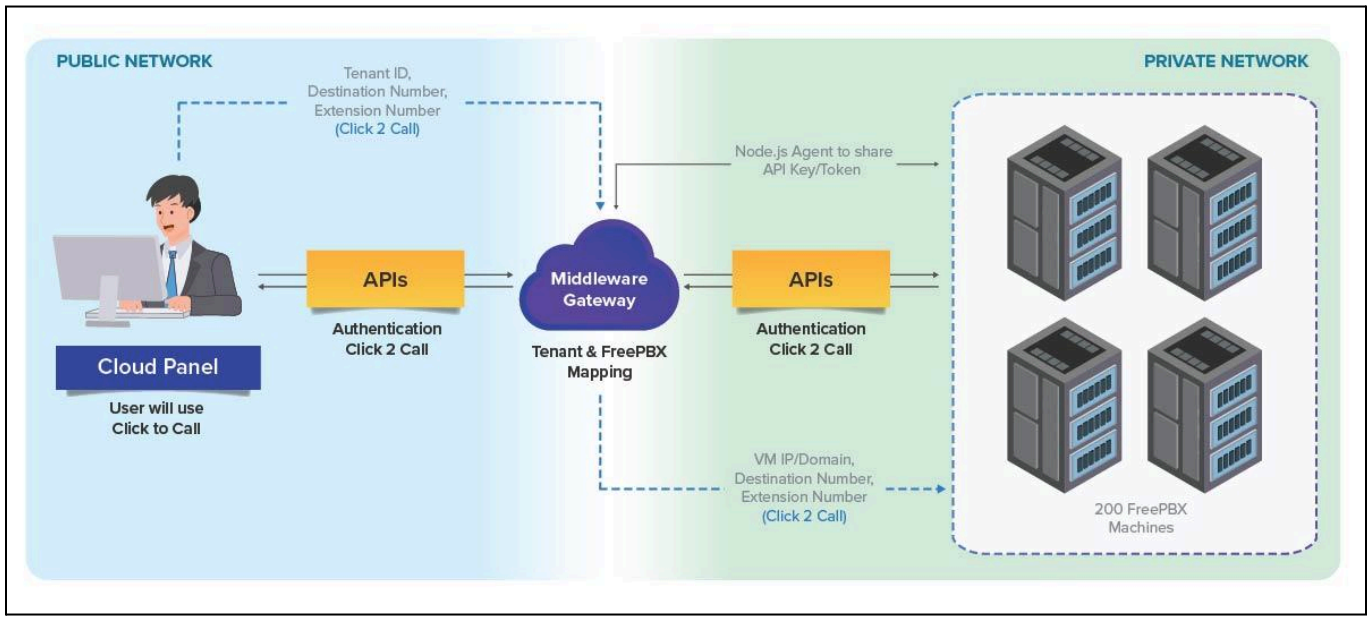
Currently CRM is integrated with the FreePBX system directly using Click 2 Call API's. Client does have 200 numbers of FreePBX/Asterisk machines which are individually integrated with CRM using API Key.

A middleware gateway needs to be developed with following features.

- This middleware gateway should be integrated with all 200 individual FreePBX machines
- Requires to develop an **Node.js agent** which will generate and share API keys with middleware gateway
    - Ecosmob to develop **Node.js agent** which can be simply installed on the FreePBX VM's, Installation on 200 Machines will be managed by client using image installation/upgradation or automation primarily. During further brainstorming it was discussed that we needs to share DevOps engineers who should be able to perform the deployment on these VM machines using Ansible or relevant automation tools
- Required to develop this middleware gateway in a way where in it will expose following API's to CRM
    - Authentication API
    - Click 2 Call API

- Listener to receive the API key from agent which is installed on FreePBX machine
- API in Middleware gateway to share the key/token with CRM

Architecture Diagram



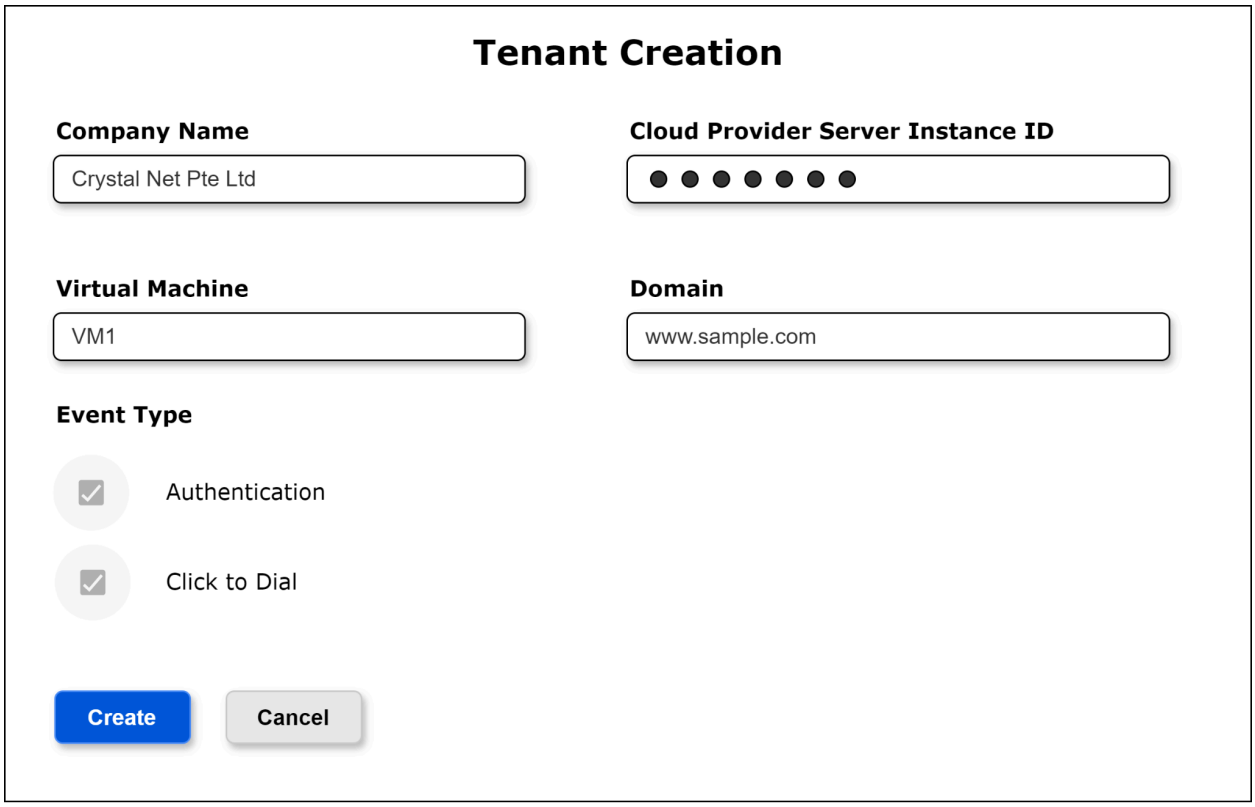
- The CRM user will use **Click to Call** functionality from the CRM portal
  - There will be one portal of middleware gateway where users can enter a Domain/IP to be whitelisted and one API key will be generated in which it can be shared with CRM vendor for executing the various API's
    - Authentication API
    - Click 2 Call API
- CRM to execute the Click 2 Call API of Middleware gateway, Following information are expected
  - Agent or Extension Number Ex. 1001
  - Destination Number(Clients phone number) Ex. 9727794632

    - Tenant Name Ex. Ecosmob - Using this tenant identification middleware gateway will be able to send the call to relevant FreePBX machine
    - Middleware to send call request to relevant FreePBX machine in desired format so that call can be initiated between agent/extension and customer number
- Middleware gateway should be in Private Network
    - CRM will be allowed to execute the APIs via Public Network using whitelisted Domain only
    - Middleware will communicate with FreePBX on private network as desired by client
- NodeJS agents who will be sharing API Key and token will be installed through Ansible to all FreePBX instances
    - Upgradation of that Agents can be also done through Ansible.
- **Notes:** Ecosmob will first check installation / Upgradation / of Node JS agents and Testing on one FreePBX test server, then implement it to all other FreePBX servers

The Middleware Web Portal may contain following components:

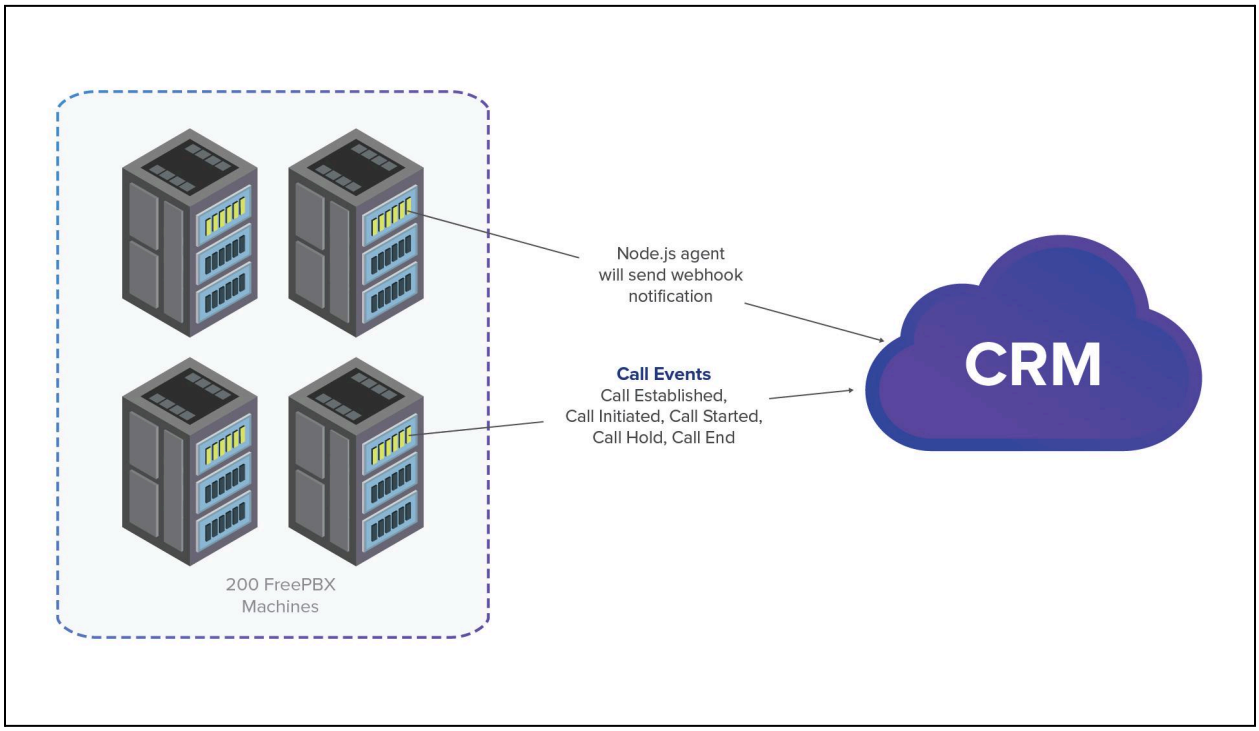
- Landing Page
- Login Page (User and Password will get created from Back-end and we have considered only Admin user for this Portal)
    - User Name
    - Password
    - Remember Me, Forgot Password (Reset option via an email)
- Tenant Management
    - Tenant Listing
        - Company Name / VM Name
        - Domain
        - API Key / Secret Key / Token
        - Modify / Delete
    - Tenant Creation
        - Company Name / Virtual Machine Name
        - Cloud Provider Server Instance ID
        - Domain (Ex. client1.domain.com)
        - Event Type: (Default checked)
            - Authentication - Checkbox
            - Click to Call - Checkbox

Wireframe Of Tenant Creation



Note - Ecosmob to develop Web Portal using React.js and Back-end will be Node.js

Live Call Notifications For all events

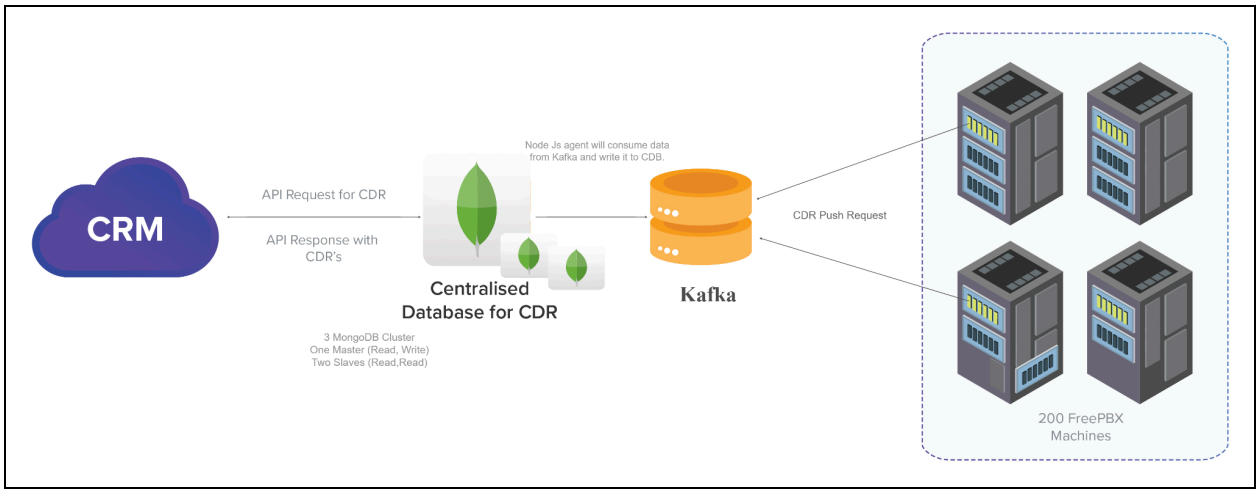


- Develop a webhook notification system to generate and send notifications
    - Agent which we are going to implement in the FreePBX machine, Can also detect and re-transmit both inbound and outbound events to CRM in the form of webhook notifications
- Integrate this system with the CRM cloud panel
- Implement real-time pop-up notifications for call events, such as
    - Call Ringing, Established/Answered, Rejected, Call Completed
    - Following could be the possible CDR's value
        - Unique Call ID, From Number, Destination Number, Caller ID, Call Direction(Inbound/Outbound/Internal), Call Status, Call Duration (Sec), IP Trunk Name, SIP Cause Code, SIP Response Value etc.

- Ensure users receive immediate updates on call events to enhance their ability to track and manage calls directly within the CRM interface - For Inbound Call Handling(Inbound Call Notifications)

Notes : This will be direct integration between FreePBX machine (Nod.js Agent) and CRM, Middleware gateway will not be in this communication flow. If these notifications from FreePBX to the CRM panel does not work appropriately due any technical challenges, We may consider switching with Middleware Gateway for call notifications.

Centralised Database for CDR



- We will Collect CDR data from Asterisk AMI (FreePBX servers) and send it to the Kafka Message Broker
- Then transfer the data from Kafka to a centralized database for storage (Which will be mongoDB)
    - There will be 3 servers required for centralized DB:
        - One master Mongo cluster (Read, Write) and Two Slave Mongo Cluster(Read, Read)
            - CPU: 2 vCPUs per broker
            - Memory (RAM): 4 GB RAM per broker
            - Storage: 100 GB
    - A single Kafka cluster with 3-5 brokers should suffice if the message volume is moderate (e.g., up to a few thousand messages per second).
    - Each Kafka broker node (server) in the cluster should have the following:
        - CPU: 4 vCPUs per broker
        - Memory (RAM): 8 GB RAM per broker

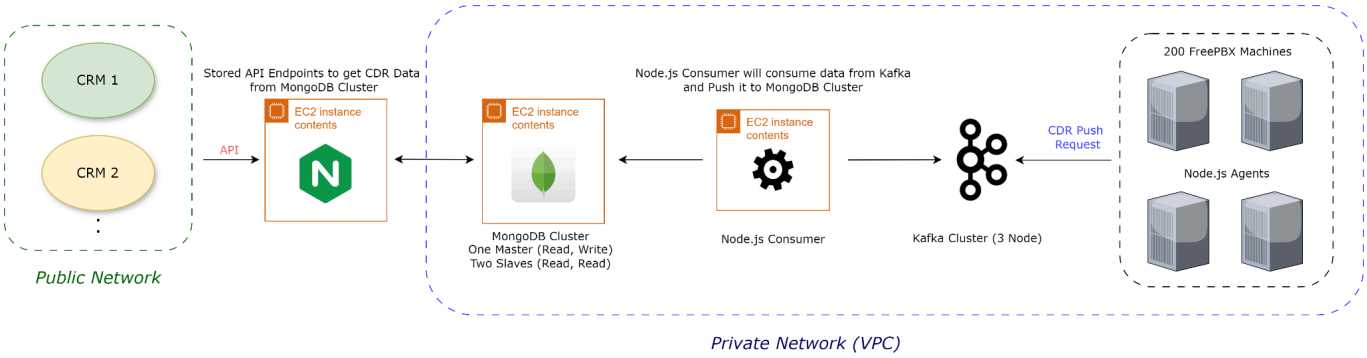
        - Storage: 50 GB
    - We will install **Node.js agent** to consume data from **Kafka** and write it to the **centralized MongoDB database**. The agent will serve as the intermediary between the Kafka message broker and MongoDB

- Also Develop API Endpoint at Centralize DB so CRM can retrieve the CDR data using these API's
    - Recording links will be stored by node js agent in centralized DB, So cloud panel can fetch alongside CDR.

Notes:

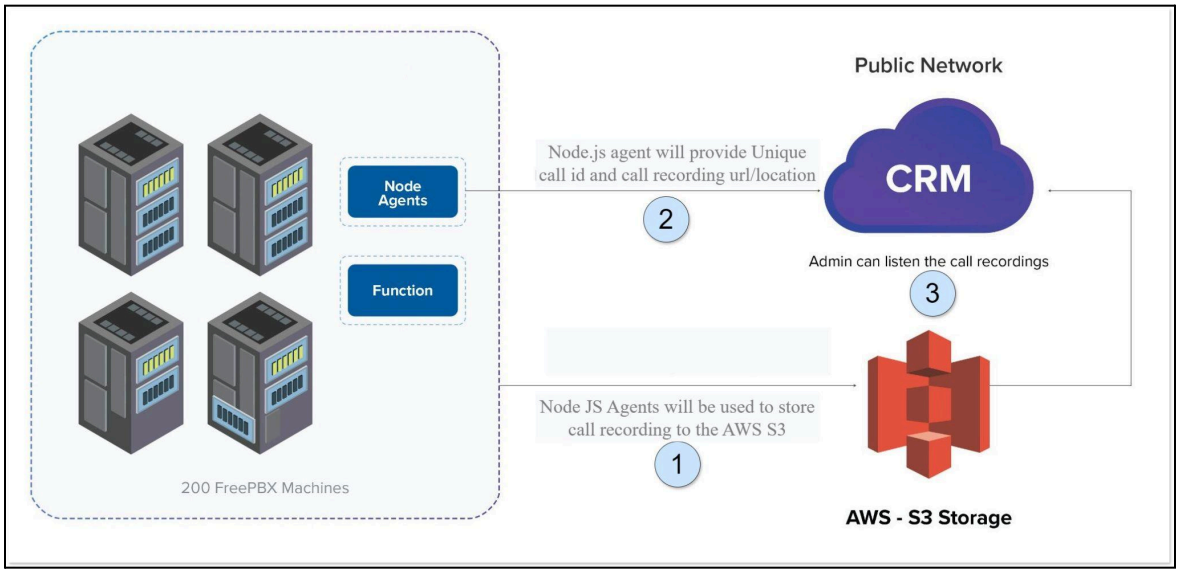
- We assume that each FreePBX server generates 1 CDR per second, that would be 200 CDRs per second across all servers.
- **Exclusion** - Web Portal is not considered to show these CDR's, But yes API is considered so that using that API CRM can pull or show CDR's information on their CRM Panel/Portal

Architecture



- A Kafka cluster will receive **CDR data** from **Node.js agents** running on **FreePBX servers**.
- A **Node.js-based consumer** (CPU: 2 VCPUs, Memory (RAM): 2 GB) hosted on an **EC2 instance** will consume this data from the Kafka cluster and push it to a **MongoDB cluster**.
- **API endpoints** to fetch CDR data will be hosted on the **EC2 instance**, allowing the **CRM** to request data from the MongoDB cluster.
- We will **whitelist the CRM's domain**, ensuring that only authorized traffic from the CRM can access the data.
- This setup will solve the issue of securely fetching data from a **public network** to a **private network**, while maintaining high security and controlled access.

Call Recording

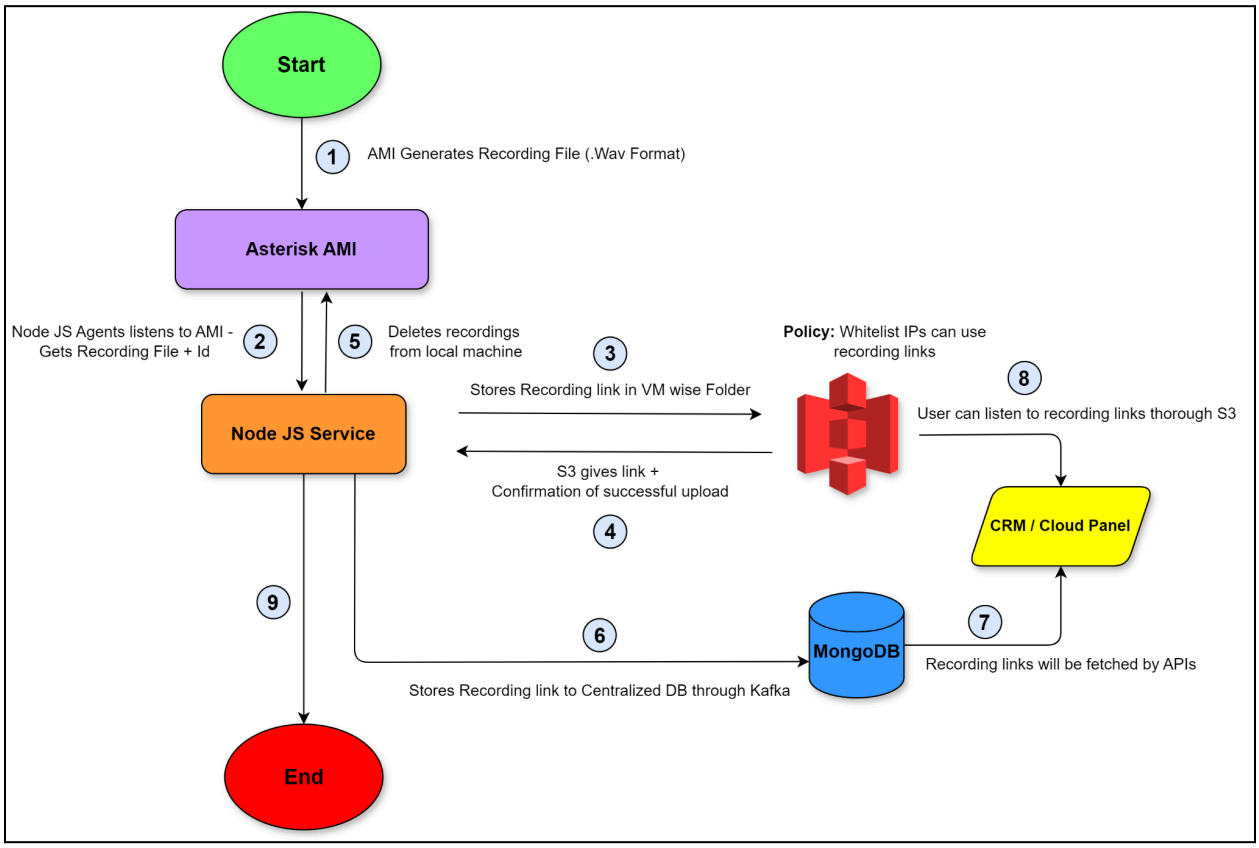


1. A **Node.js service** will be installed on all FreePBX servers via **Ansible**.
2. This **Node.js agent** will get Call Recordings (.Wav File) from asterisk AMI.
3. A **Node.js agent** creates VM-specific directories (Naming files as domain names) within an **S3 bucket** and upload call recordings from each FreePBX machine to the designated directory in S3.
    - **Fetch EC2 Metadata**: Use the EC2 Metadata service to get the instance's public hostname or ID to uniquely identify each domain.
    - Ex:`curl http://169.254.169.254/latest/meta-data/public-hostname`
    - **Create Domain-Specific Folder** : Use `*fs.mkdirSync*` to create a folder locally for each domain based on the retrieved metadata.
    - **Save Recordings** : Save each recording in the appropriate domain folder using `path.join(domain, fileName)`.
        - Note: Just for Example
    - **Upload to S3** : Use AWS SDK in Node.js to upload the files from the domain folder to S3, structuring the folders by domain.

4. Once the call recordings are successfully uploaded to **S3**, the node.js agent will confirm the recording file upload and then delete the recordings from the corresponding local drive of Freepbx servers.
5. The **Node.js agent** will also push the call recording links to a **centralized database** using **Kafka**. From there, the **CRM (Cloud Panel)** will fetch the recording links, allowing administrators to play the recordings directly through the cloud panel interface.

- **Note:** The recording links will be secure and accessible only to authorized users. Any attempts to access the links from outside the organization will be blocked.
    - It will be managed from AWS Policy, Where we can set IPs that can access the Recordings.

Flow for call recording



Call Status

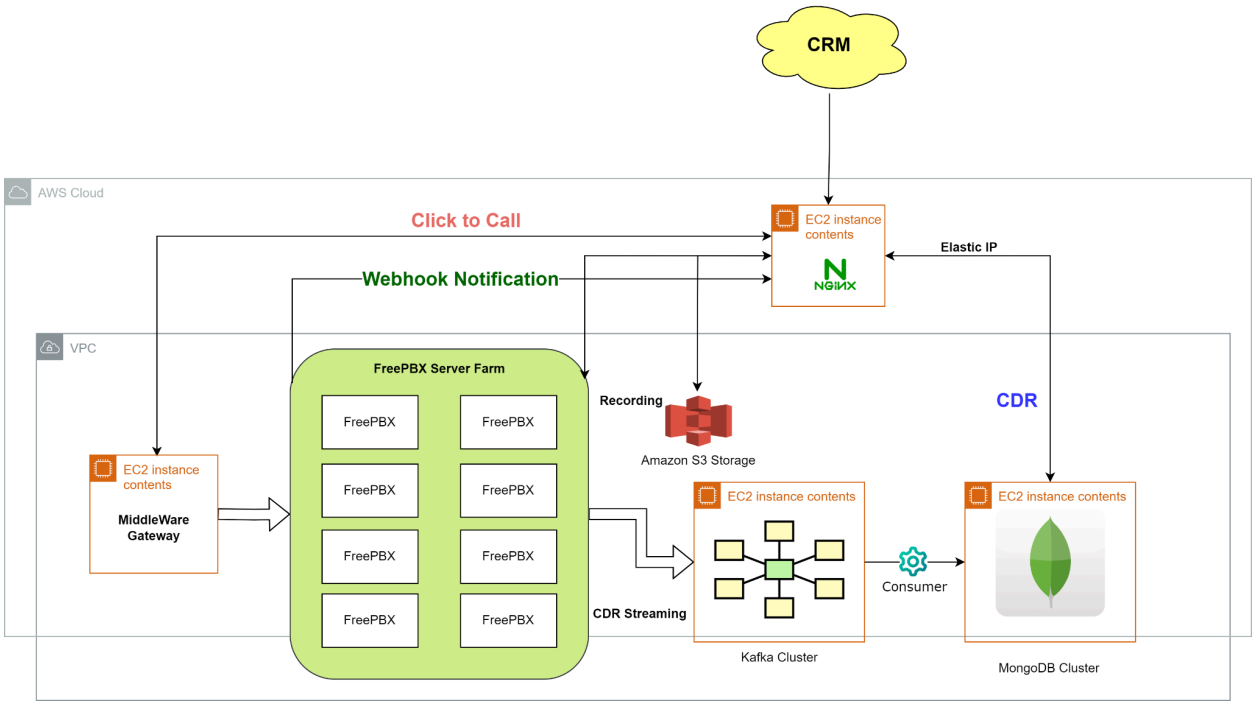
Call status:

- Client will provide the different SIP Cause and SIP response code received by different Call Providers
- Ecosmob will make custom status changes in CDR's as suggested by the client
- Eg. We will perform the changes in the MongoDB (Centralized DB) for call status like success(Call Answered), unavailable(No Answer/Unavailable),Busy and Rejected. It is recommended to add new column and perform changes in final cause codes there instead of performing the changes in actual ISDN/SIP Cause codes

Call Status (For Queue)

- In a scenario where we have to show call status where there is a ring group where one agent answers the call and others won't
- Ecosmob will make custom call status for that which can help admin (from CRM Panel) identify the call event
- These will be again changes in the MongoDB (Centralized DB), One more column can be added or changes in the existing field can be done as required from client side
    - **Note**: For call status, Ecosmob can be discussed with the client after project kickoff and find the best solution.

General Architecture



Technology Stack

Modules	Technology
Frontend	React.js
API Gateway and Consumer	Node.js
Que System / Message Broker	Kafka (Cluster)
Database	MySql - For configurational data MongoDB - For Big Data (Cluster)
VoIP	FreePBX (Asterisk)

Tech Specifications

Specifications	Description
Server OS	Debian 12
Kafka Cluster	3-5 Brokers CPU: 4 vCPUs per broker Memory (RAM): 8 GB RAM per broker Storage: 50 GB
MongoDB Cluster	One Master (Read, Write) Two Slave (Read, Read) CPU: 4 vCPUs per broker Memory (RAM): 8 GB RAM per broker Storage: 100 GB
Node.js Consumer (For Kafka - MongoDB)	CPU: 2 vCPUs Memory (RAM): 2 GB (Can scale as per requirement)
Node.js	Version: 20.17.0 Port: 3000

Tentative Timeline

Modules	Timeline
Middleware Gateway for Click to Call	4 Weeks
Live Call Notifications for call events - Node.js Agent Deployment using Ansible	
Change in Asterisk - To push CDR's to Kafka Cluster Centralised Database for CDR - Kafka Cluster - Node.js consumer - MongoDB Cluster API to fetch CDR's from DBMS	
Uploading Call Recording on the AWS S3 Bucket and Notification to S3 for call recording links and pushing the same link to mongo. Call Status (Normal Call Status, Queue Call Status)	
UAT (To be Performed by Customer, Post Sales Team and QA will share the Test Cases, Test Case Amendment can be done by client)	2 weeks

Notes : It will be approx 5 to 7 Weeks

Notes/Assumptions

1. Ecosmob will not be responsible for any 3rd party service or API procurement and its license or subscription cost
2. Server and FreePBX upgradation are not considered in this scope, It will be managed by 8 Ventures/Crystal Net
3. Ecosmob to develop Web Portal using React.js and Back-end will be Node.js for Middleware gateway management only
4. Webhook Notification : This will be direct integration between FreePBX machine (Agent) and CRM, Middleware gateway will not be in this communication flow
5. Web Portal is not considered for centralized CDR system, There will be API using which CRM will be able to fetch the CDR's
6. For monitoring auto log errors and auto restart itself if services fail, Ecosmob will install Supervisor which will be monitoring and restarting if it fails and there will be logs for everything.
7. The monitoring will be sending email notifications if service fails or restart itself. The Client will share SMTP details for email notifications and Email/User ID and Password for authentication.

About us

Ecosmob Technologies Pvt. Ltd. is a world-renowned provider of carrier-grade software solutions and services. We strive to deliver innovative and client-centric solutions. We help our clients modernize their networks to improve competitive positioning and business outcomes.

Contact us

INDIA
501-503, Binori B Square 1, Nr. Neptune House,
Ambli -Bopal Road, Ahmedabad-380058, Gujarat, India.

USA
300 SE 2nd Street, Suite 600 Ft. Lauderdale,
Florida, USA 33301

SOUTH AFRICA
158 Kiepersol street, Grootvlei, Pretoria 0120, South Africa.

CANADA
1285 West Broadway, Suite 600, Vancouver, BC, V6H 3X8

PHONE
+1-303-997-3139
+91-7778842856
+1 (604) 900-8870

MAIL US AT
sales@ecosmob.com

VISIT US

ecosmob
makes your vision a reality