exotel
DOCS

Harmony CX Platform

1. New Capabilities

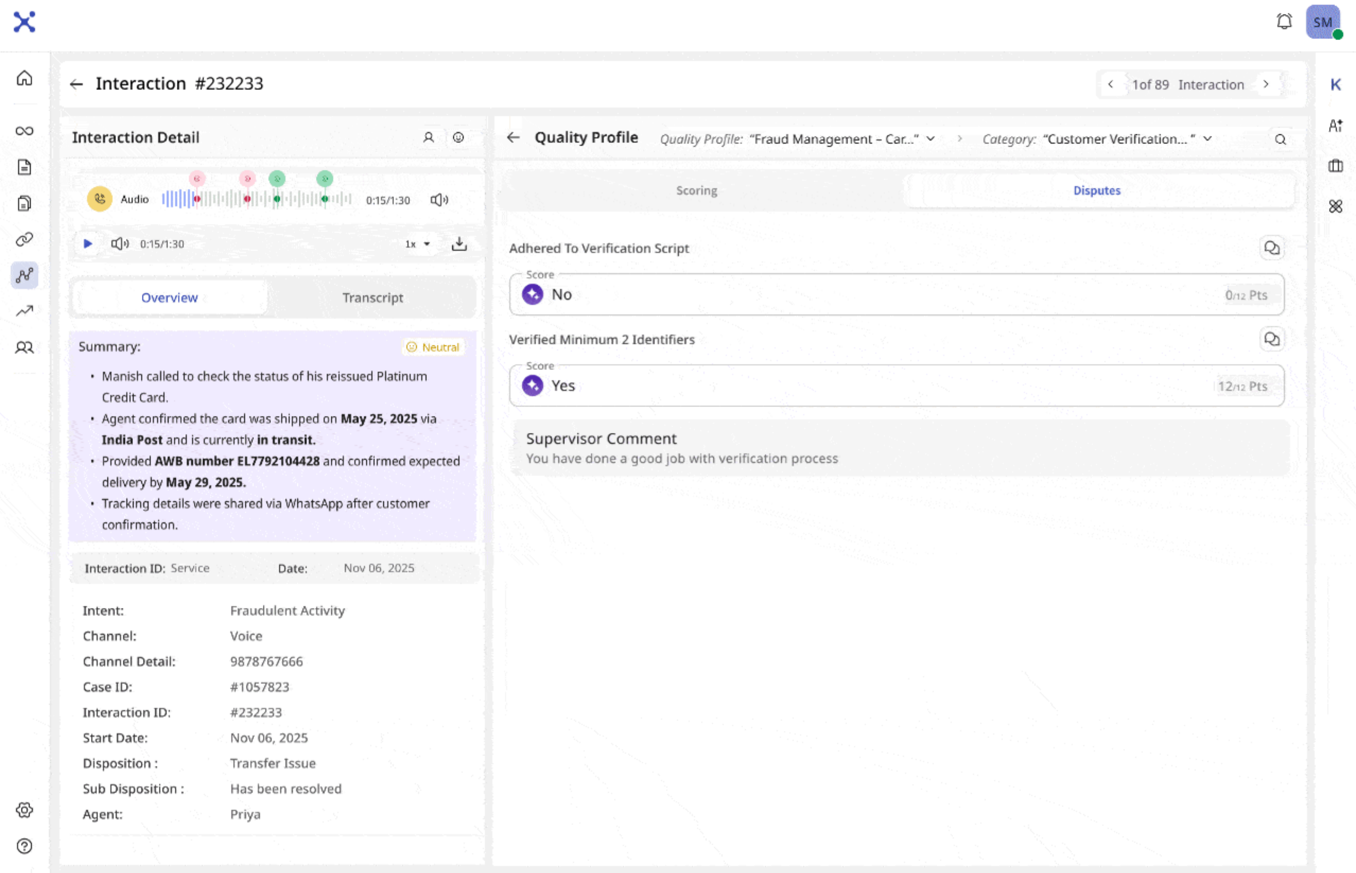
1.1. For Admin/Supervisors/Analys

1.1.1. AI-Powered Quality Analysis

Quality at Scale. Powered by AI.

When enabled, AI Analysis reviews every interaction and scores agents on predefined quality parameters, helping QA teams scale evaluations, reduce bias, and identify coaching opportunities faster.

View Documentation



SM
Interaction #232233
1of 89 Interaction
Interaction Detail
Audio
0:15/1:30
0:15/1:30
1x
Overview
Transcript
Summary:
Neutral
Manish called to check the status of his reissued Platinum Credit Card.
Agent confirmed the card was shipped on May 25, 2025 via India Post and is currently in transit.
Provided AWB number EL7792104428 and confirmed expected delivery by May 29, 2025.
Tracking details were shared via WhatsApp after customer confirmation.
Interaction ID: Service
Date: Nov 06, 2025
Intent: Fraudulent Activity
Channel: Voice
Channel Detail: 9878767666
Case ID: #1057823
Interaction ID: #232233
Start Date: Nov 06, 2025
Disposition : Transfer Issue
Sub Disposition : Has been resolved
Agent: Priya
Quality Profile
Quality Profile: "Fraud Management - Car..."
Category: "Customer Verification... "
Scoring
Disputes
Adhered To Verification Script
Score
No
0/12 Pts
Verified Minimum 2 Identifiers
Score
Yes
12/12 Pts
Supervisor Comment
You have done a good job with verification process

1.2. For Agents/Executives

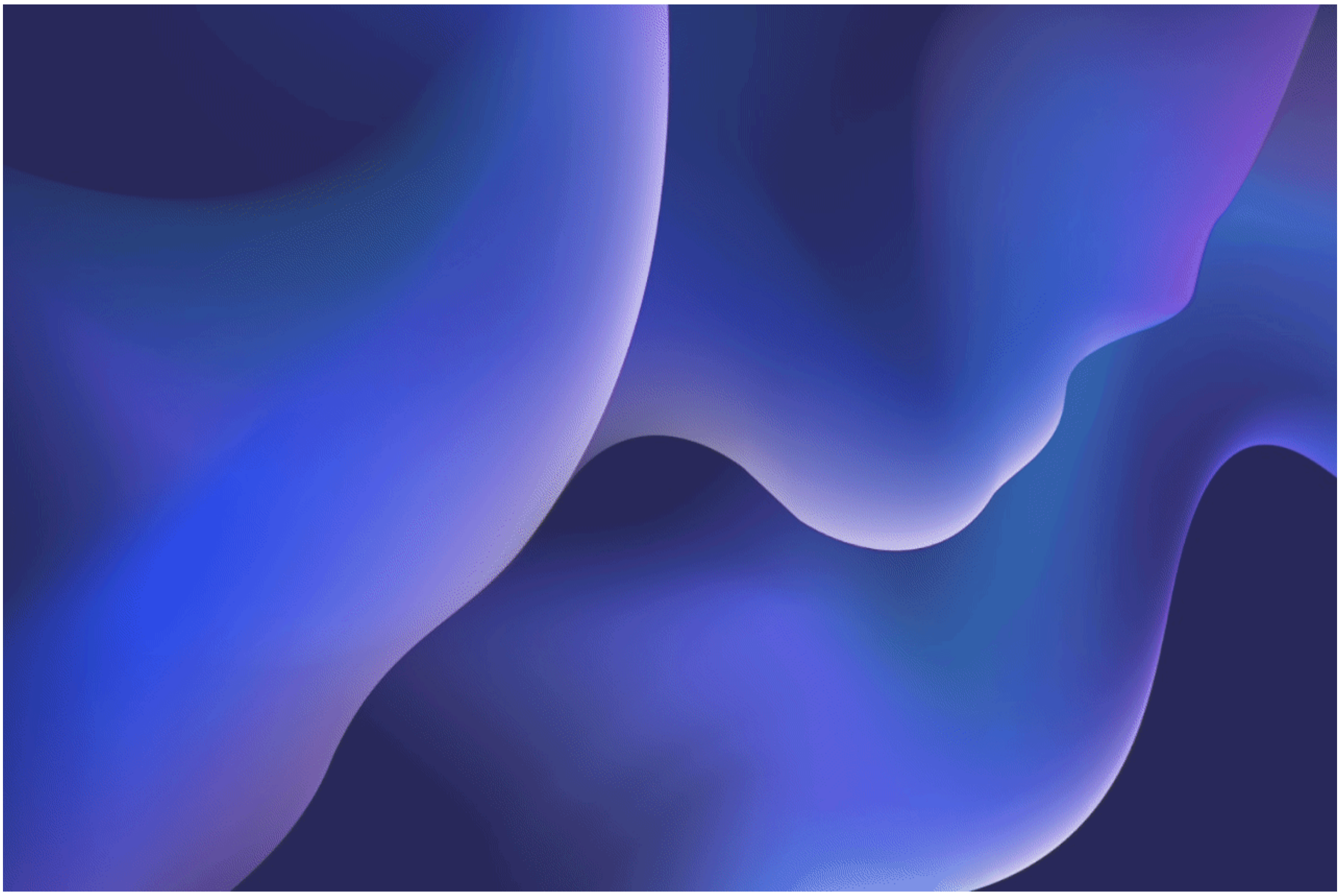
1.2.1. AI Assist

Become a Top Performer Agent

With AI Assist on live calls, agents get real-time transcripts, instant SOP/KB suggestions, auto-generated notes, and smart prompts to stay accurate.

The result: less after-call work, faster closures, better AHT & QA, and more consistent performance

See Interactive Tour

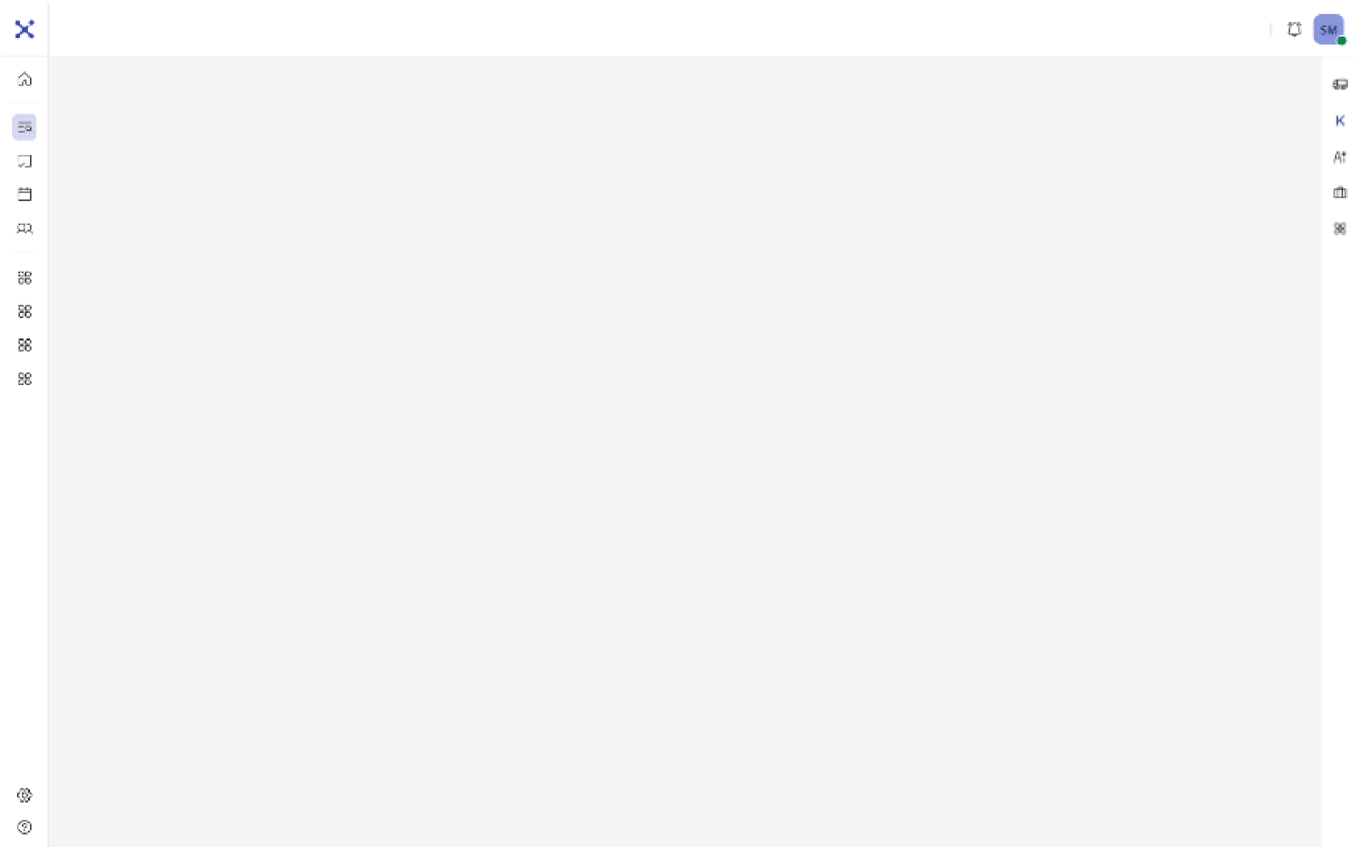


1.2.2. Bot Monitoring

Level up to Monitoring Officer!

As a Monitoring Officer, you let the AI handle the routine while you handle the risks. Step in only when it matters most. Leverage your experience to ensure quality and control the result, not the process

See Interactive Tour



2. The Harmony Design System

2.1. Harmony UX Philosophy

Designing for Human-Centered Enterprise Excellence

What is Harmony?

Harmony is our unified design system that brings visual consistency, usability, and scalability to every experience across the **Exotel** ecosystem. It acts as the single source of truth for our design language—defining how products look, feel, and behave to create seamless experiences for agents, supervisors, and administrators alike.

At its core, Harmony ensures that every interaction feels familiar, efficient, and delightful—reducing cognitive load and empowering users to perform their best.

Harmony UX Philosophy

A Strategic Framework for Scalable, Efficient, and Inclusive User Experience

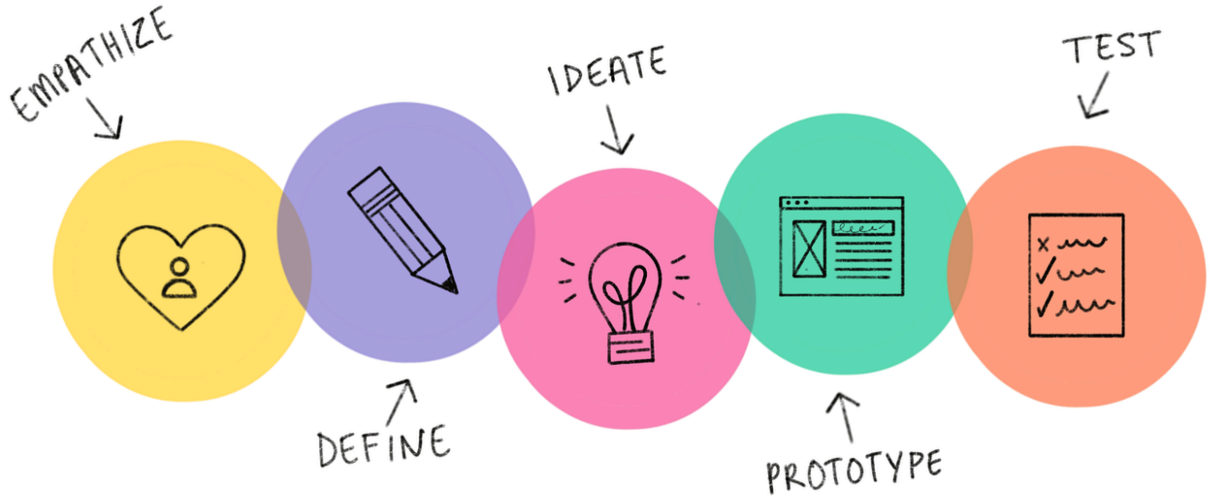
In enterprise environments, users often deal with multiple tools and fragmented workflows. This leads to:

- High cognitive load → slower task completion
- Inconsistent experiences → higher training costs
- Poor usability → decreased productivity

Harmony aims to change that reality. It's not just a visual system – it's a design philosophy rooted in **clarity, consistency, and empathy**. It helps teams build experiences that are simple, predictable, and aligned with how people actually work.
Our vision is to **create enterprise experiences that feel intuitive, inclusive, and intelligently connected** across every touchpoint.

2.2. The Harmony Design Process

Our process combines research, design thinking, and system scalability to create meaningful, repeatable design outcomes.



- **Discover** – Understand user needs, behaviors, and challenges through research and collaboration.
- **Define** – Translate insights into clear design goals and success metrics.
- **Design** – Create patterns and components that are flexible, accessible, and visually aligned.
- **Validate** – Test with real users to ensure usability, performance, and satisfaction.
- **Evolve** – Continuously refine the system through feedback, analytics, and team input.

This iterative process ensures Harmony stays relevant and adaptable as our products and users evolve.

2.3. Tokens

2.4. Components

3. Third party CRM Integration with CC

Overview:

CRM Integration enables seamless connection between a contact center platform and third-party CRM, ERP, LMS, or core systems to provide a unified agent interface and improve customer journey management. This integration allows contact center agents to access customer data, handle interactions across multiple channels, and efficiently manage customer relationships and workflows.

Key Benefits:

- Enhanced agent efficiency through integrated customer data access.
- Omni-channel support providing seamless customer interaction experiences.
- Flexible integration approaches supporting various CRM architectures and requirements.

Functionality:

CRM Integration supports two main host system types: CRM as Host and Ameyo as Host.

1. **CRM as Host:** Ameyo provides voice capabilities embedded within the CRM via a prebuilt or custom toolbar. Agents work primarily on the CRM while using Ameyo for voice calls.
2. **Ameyo as Host:** The agent works within the Ameyo platform and accesses CRM data either through URL-based integration or Web Services-based integration. This approach supports omni-channel communication and richer agent functionalities.

Integration methods include:

- **Web-Toolbar Integration** for **CRM as Host**, embedding Ameyo's toolbar for voice calls. For example, Ameyo Integration with Salesforce, Microsoft Dynamics CRM, Freshdesk, Zendesk, etc
- **URL Based Integration** where CRM pages are displayed within Ameyo via iFrames with URL parameters.

- **Web Services Integration** using RESTful/SOAP APIs to fetch and update customer data on Ameyo's CTI with a custom mini-CRM.

Each method supports specific use cases based on CRM capability and integration needs.

Prerequisites:

- **Web-Toolbar Integration:** Requires a web-based CRM with support for iFrames and JavaScript events, and page refresh restrictions for WebRTC connections.
- **URL Based Integration:** Requires web-based CRM with cross-application iFrame support, acceptance of URL API parameters, and browser plug-ins like "Ignore X-frame."
- **Web Services Integration:** Requires CRM support for RESTful/SOAP APIs capable of two-way data communication.

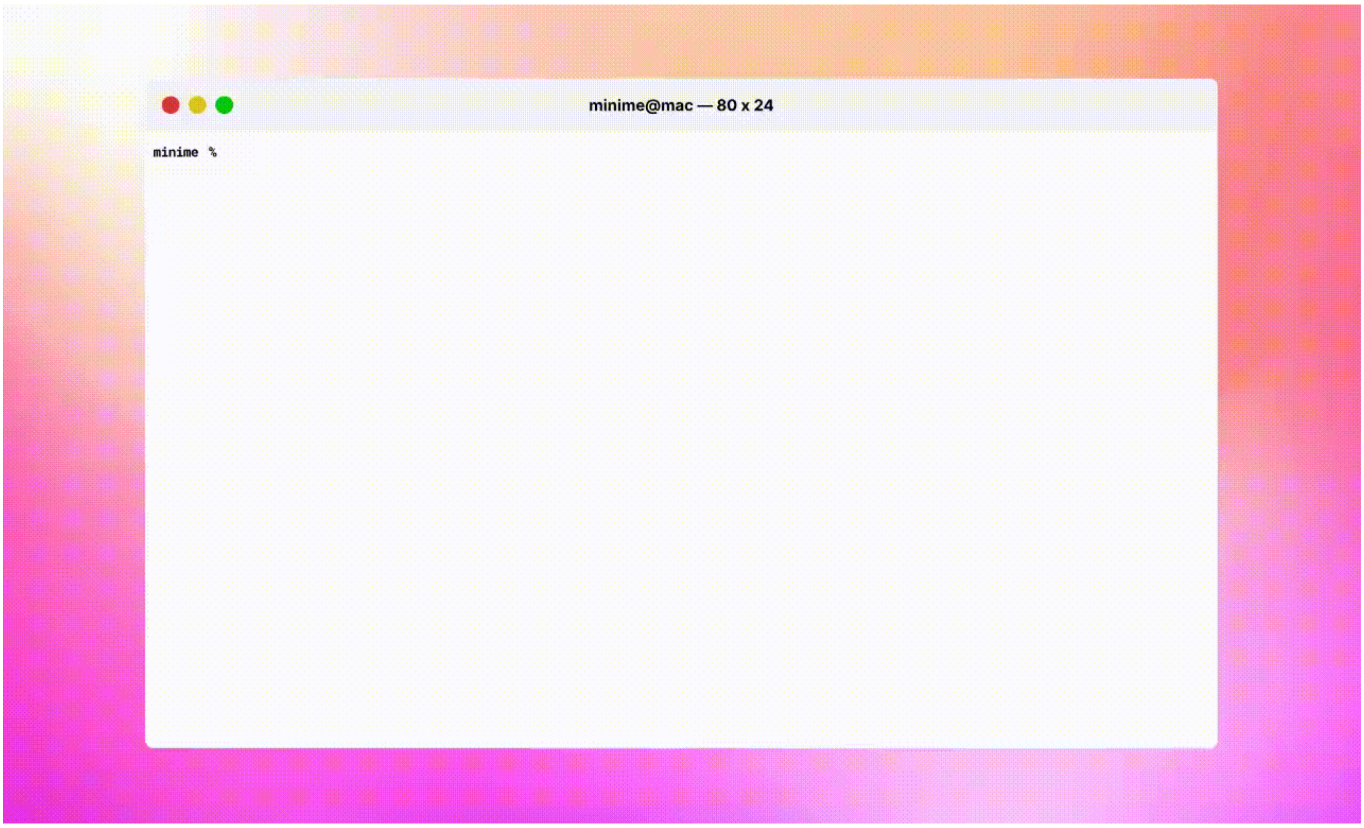
Usage Instructions:

1. Identify the integration host model suitable for your CRM environment.
2. **For CRM as Host with Web-Toolbar:**
   - Embed Ameyo Toolbar iFrame into CRM.
   - Configure toolbar with agent call control features.
   - Ensure CRM meets JavaScript event and page refresh requirements.
3. **For Ameyo as Host with URL Based Integration:**
   - Configure Ameyo to invoke CRM URLs with relevant query parameters during agent login, call handling, and call disposition.
   - Ensure CRM handles and responds to these URLs to display customer info.
4. **For Ameyo as Host with Web Services Integration:**
   - Implement API endpoints in the CRM for GET, POST, and PUSH operations.
   - Configure Ameyo mini-CRM to interact with these APIs.
   - Support both one-way (view only) or two-way (view and update) data exchange scenarios.

Summary

This documentation provides comprehensive guidance for users to understand, deploy, and operate the CRM integration feature within the Ameyo contact center platform, ensuring efficient interaction management aligned with customer journey objectives.

3.1. Omni Custom Apps – Developer Guide



Omni Custom Apps let you extend the contact center by embedding custom functionality directly within the agent workspace. These apps run inside an iframe and connect to the platform using the Omni SDK, which gives you access to real-time events like calls, messages, and agent actions.

In practice, this means you can build things like CRM integrations with screen pop, messaging triggers, call workflows, or lightweight agent assist tools—all directly inside the agent interface.

Getting started (takes a few minutes)

Start by installing the CLI globally:

```
npx create-exotel-omni-app@latest
```

Then generate your app:

```
create-exotel-omni-app
```

You'll be asked whether you want a blank app or a template. If you're exploring, templates are faster—they give you working UI and patterns you can modify instead of starting from zero.

Once the project is created:

```
npm install
npm run dev
```

That's enough to get a running app locally with the SDK already wired up.

When you're ready to deploy:

```
npm run ship
```

This produces an omni-upload.aaex file, which you upload via **Admin → App Manager**.

What's actually happening under the hood

The generated app already handles the hard parts:

- Loading and initializing the Omni SDK
- Setting up event subscriptions
- Wiring the app into the iframe environment
- Packaging the app in the correct .aaex format

You're not building integration plumbing—you're building the actual use case.

The important implication: don't touch the SDK setup unless you know exactly why.

How you should build (this is where most teams go wrong)

The fastest way to build these apps isn't manually—it's with AI.

Every generated project includes a DEVELOPER.md file. This isn't documentation fluff—it's a contract that tells AI tools exactly how to work within the system.

The intended workflow is simple:

- Add DEVELOPER.md in any AI agent console and say 'Let's Begin'

- Describe what you want to build

The agent will:

- Ask a few clarifying questions
- Implement the feature directly in your codebase
- Stay within the allowed SDK and UI patterns

If you follow this, you can go from idea to a working app in minutes. If you ignore it, you'll end up manually wiring things the scaffold already solved.

UI expectations (keep this simple)

Use the Signal Design System (@exotel-npm-dev/signal-design-system) as your default. It's already included—you don't need to install anything extra. Running npm install pulls it in automatically.

These components are built to match the agent interface and behave correctly inside an iframe, which is where most custom UIs break.

If something truly isn't available, you can fall back to MUI—but that should be rare.

The practical rule: start with Signal, only deviate if you hit a real limitation.

Project layout (what you actually need to care about)

You don't need to understand the entire repo. Focus on this:

- app/components/useCases/ → where your feature logic lives
- app/omni/ → SDK integration (leave this alone)
- registerOmniSdk.generated.ts → where events are wired
- manifest.json → controls name, version, and deployment path

Everything else is supporting structure.

Also: keep SDK logic out of your UI components. Use the existing bridge layer. Mixing the two will make your app harder to maintain very quickly.

A couple of non-negotiables

> There are two files you should not modify:
>
> - loadOmniSdk.ts
> - OmniProvider.tsx

They're already aligned with the platform requirements. Changing them is the fastest way to introduce hard-to-debug issues.

Same goes for events:

- Only use documented SDK events and methods
- Don't invent payloads
- Always handle errors properly

System Prerequisites

Before you code

- Node **18+**, **npm**, **zip** (for ship)
- Run npm install in the folder with package.json
- Access to **npm registry** for @exotel-npm-dev/signal-design-system

Before you ship

- npm run ship → omni-upload.aaex
- **App Manager** upload + slot assignment
- manifest.json **name** / **version** match ECC URL

Before CRM works in Omni

- Field keys/types in crmIntegrationConfig.ts
- Test inside **Omni iframe** with a **live interaction**

Deployment details that trip people up

The packaging step (**npm run ship**) gives you a ready-to-upload .aaex. That part is straightforward.

Where things usually break is hosting.

Your app is served from a path like:

```
/si/{name}/{version}/
```

Those values come from manifest.json. If they don't match your hosting setup, you'll get missing JS errors (usually 404s on static files).

Also make sure you're uploading the full dist/ output, including the static/ folder.

Troubleshooting (quick reality checks)

If something breaks, it's usually one of these:

- **JS files not loading** → manifest mismatch
- **Theme looks off** → iframe or root theme issue
- **Events not firing** → wrong event name or silent error
- **Build suddenly unstable** → you ran npm audit fix --force

That last one is worth repeating: don't do it. It tends to upgrade dependencies beyond what the template supports.

Why this setup exists

Without this scaffold, teams consistently run into the same problems:

- SDK initialization issues
- Incorrect event handling
- Broken iframe theming
- Invalid packaging
- UI that doesn't adapt to the agent layout

This setup removes those variables so you can focus on what actually matters: building the workflow.

And if you lean into the AI workflow properly, you're not just saving effort—you're compressing build time from days to minutes.

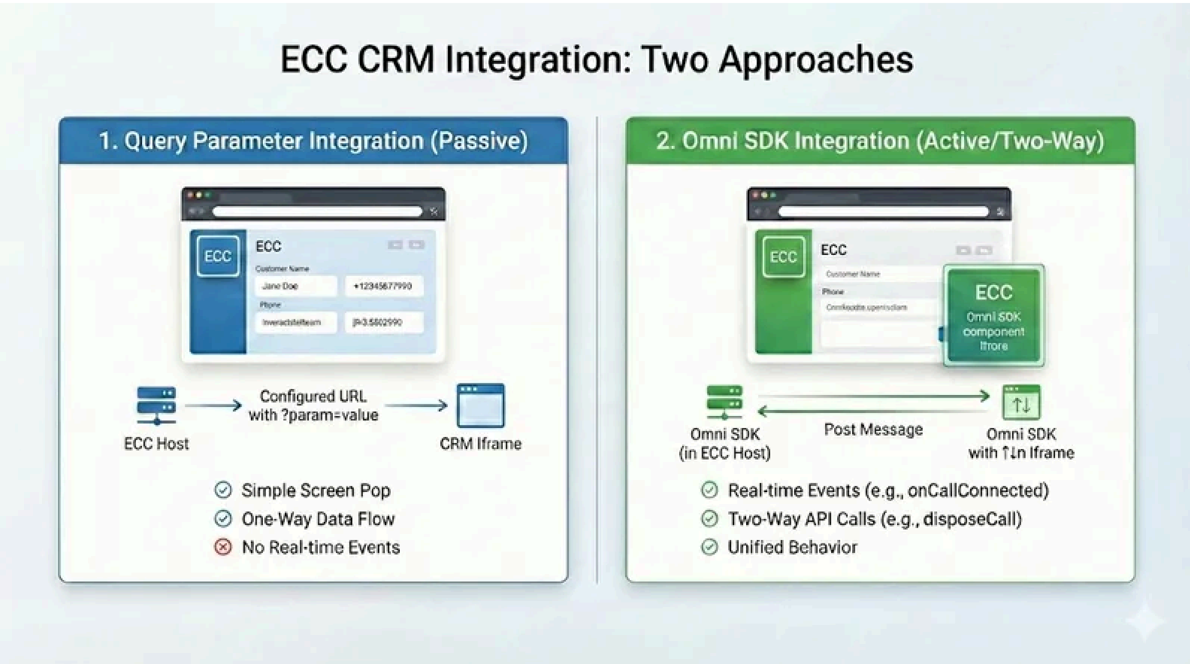
3.2. Omni URL Based CRM integration Guide

Overview

This document details the configuration and structural requirements for integrating a Customer Relationship Management (CRM) application as an embedded page within the Exotel Ameyo Contact Center (ECC) agent console, utilizing the **Omni SDK** for communication.

Introduction and Integration Approach

URL-Based App Integration allows external web applications to be loaded directly inside an **iFrame within the Ameyo ECC agent desktop**.



1. Query Parameter Integration (Passive)

This is the standard approach where Ameyo ECC passes initial context (interaction IDs, customer phone numbers, etc.) by appending **GET request parameters** to the configured CRM URL.

- **Best for:** Simple screen-pops and loading specific customer records based on call data.

- **Limitation:** It is a "one-way" push from Ameyo to the CRM at the moment the page loads.

2. Omni SDK Integration (Active/Two-Way)

By injecting the **Omni SDK** into the CRM page, the embedded application moves beyond a static URL load. It establishes a real-time, two-way communication bridge between the ECC host and the CRM iFrame.

- **Best for:** Real-time event handling (e.g., updating the CRM UI exactly when a call connects) and triggering ECC actions (e.g., disposing a call or dialing a number) directly from CRM buttons.
- **Advantage:** Unified behavior across all integration modes (Toolbar, App, and Iframe).

Key Structural Components

1. **iFrame Integration:** The Ameyo ECC loads the CRM's configured URL into a dedicated iFrame. The CRM must be configured to allow framing by the Ameyo domain.
2. **Context via URL (Approach 1):** Ameyo sends metadata (e.g., phone, interactionId) via URL parameters. The CRM parses these to display relevant information.
3. **Communication Layer (Approach 2):** The CRM initializes the ameyo-sdk.js to listen for events and invoke methods, enabling a seamless "native" feel within the embedded console.

Technical and Security Requirements

The following requirements ensure seamless and secure integration of the third-party CRM application within the ECC.

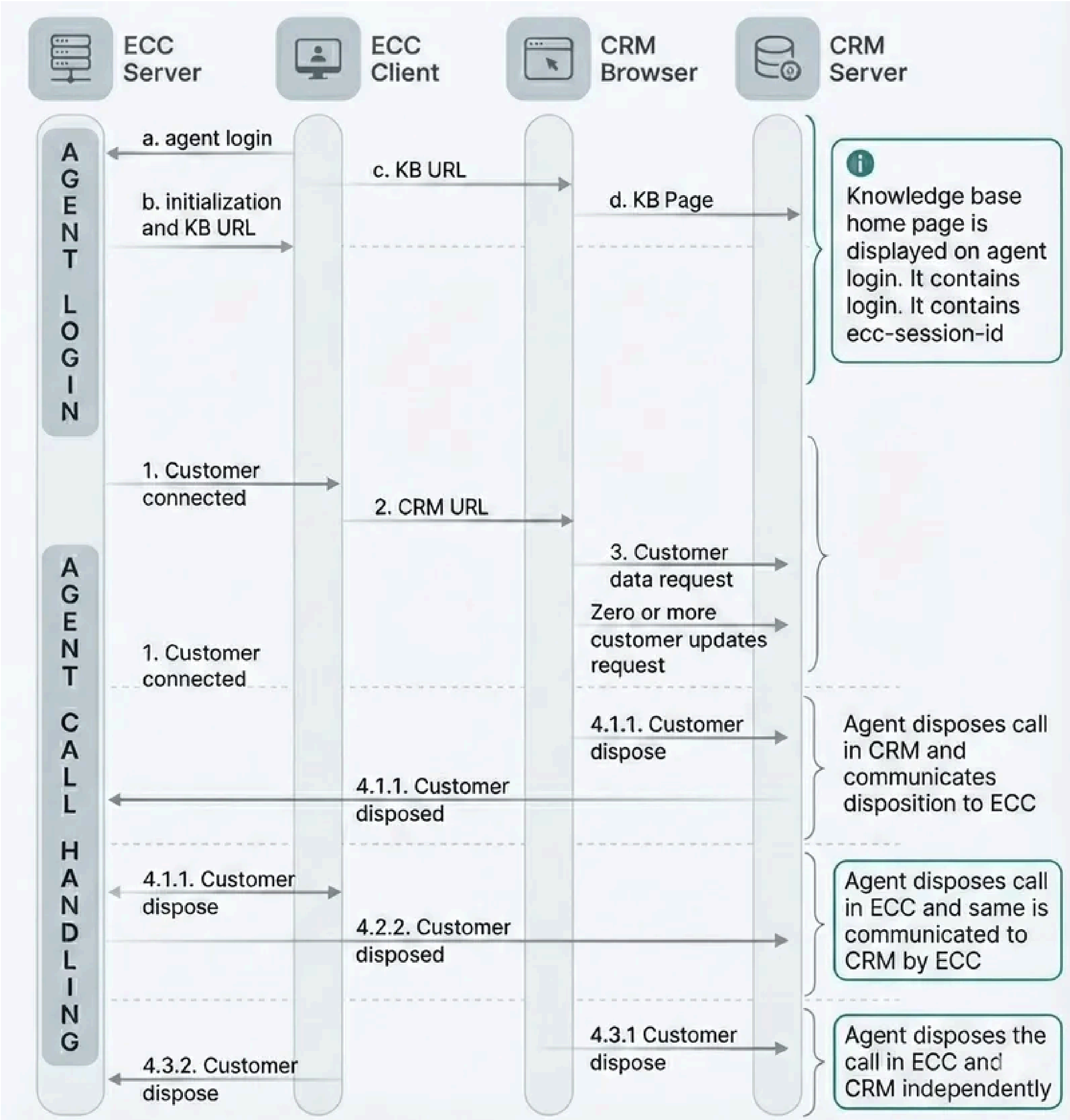
Requirement	Description	Impact on Integration
HTTPS Support	The CRM application's URL must be served over **HTTPS (SSL/TLS)**.	Ensures secure, encrypted communication between the ECC host and the embedded CRM iFrame, preventing browser security errors (mixed content) and protecting sensitive data.
iFrame Compatibility	The CRM page must not use security headers like X-Frame-Options: SAMEORIGIN or Content-Security-Policy: frame-ancestors 'self' that prevent it from being loaded in an iFrame by a different domain (Ameyo ECC).	Essential for the CRM to be successfully displayed within the Ameyo agent console.
Cross-Domain Communication	The Ameyo Omni SDK is built on a **Post Message** architecture, which natively handles communication across different domains (Ameyo $\rightleftarrows$ CRM iFrame).	This negates the need for the developer to implement custom cross-origin messaging, relying solely on the SDK.
Session Management	The CRM must support one of the configured session management policies for the agent: **No Authentication**, **Cookie-Based** (for persistence), or **Single Sign-On (SSO)**.	Determines how the agent's session is maintained or established within the embedded CRM once the page loads.

URL Formats and Integration Points

Ameyo configures specific URLs in its backend to load the CRM page at different points in the call lifecycle. The CRM must be prepared to handle the parameters passed to these URLs.

Integration Point (Ameyo Trigger)	Purpose	Placeholder Parameters (Passed via URL)
Knowledge Base URL	Loads a preconfigured Knowledge Base URL to the embedded browser for agent reference at login, where the CRM Landing Page is opened.	• userId • userName • sessionId • userType • iframeId • campaignId • crmSessionId • userCrtObjectIds • instanceId • locale • origin • contactCenterId • processId
Incoming/Outgoing Call	Loads the CRM to perform a **Screen Pop** and display the customer's record when a call arrives or connects.	• crm_push_generated_time • userCrtObjectId • phone • disableDispositionOnActiveCall • campaignId • associationType • crtObjectId • sessionId • userId

Integration Point (Ameyo Trigger)	Purpose	Placeholder Parameters (Passed via URL)
		• isOrignalPhoneRequiredInCRM • callType • displayPhone • instanceId • session-id • locale • origin • interactionId • iframeId • contactCenterId



Omni SDK Initialization and Event Registration

Since the CRM page is the one loaded inside the iFrame, it must perform the SDK initialization to establish the communication bridge. The APIs and Events remain identical to the standard Omni SDK (Generic Toolbar) integration.

SDK Initialization

The CRM must initialize the SDK using the **crm context** once its page has fully loaded inside the Ameyo iFrame.

```
(function loadSDK() {
//import url of the site where ameyo-sdk.js is hosted
import("https://<yourDomain>/omniapp/sdk/ameyo-sdk.js")
.then((module) => {
const sdk = module.default;
const sdkConfig = {
context: "app",
origin: "https://<crmDomain>",
instanceId: "ameyoIframe"
};
const ameyoOmniSDK = sdk.initialize(sdkConfig);
// Optional: If you want to expose SDK object globally, by default it will be
//available to use wherever you want to use it
window.ameyoOmniSDK = ameyoOmniSDK;
})
.catch((error) => {
console.error("Error in loading OmniSDK", error);
});
})();
```

Event Registration (Listening to Ameyo)

The CRM registers listeners to receive real-time updates from Ameyo (the ECC) based on the agent's activity.

```
ameyoOmniSDK.interaction
.register("interactionInitiated", (data) => {
console.log("Integration interactionInitiated event received", data);
})
.then(() => {
console.log(
"Integration Successfully registered interactionInitiated event"
);
})
.catch((error) => {
console.error(
"Integration Error in registering interactionInitiated event",
error
);
});
```

API Calls (Sending Commands to Ameyo)

The CRM uses API calls to instruct Ameyo (the ECC) to perform actions like disposing of a call or making an outbound dial.

```
ameyoOmniSDK.configuration.getDispositionCodesWithDispositionClass(172)
.then(res =>
console.log("Result", res))
.catch(err =>
console.error(err));
```

Executing Call Actions: SDK & REST API

To trigger telephony actions like dialing or disposing of a call from within your CRM, you have two powerful options. The choice depends on whether you want to use our client-side framework or direct HTTP requests.

Option 1: Client-Side via Omni SDK (Recommended)

The most seamless way to trigger actions is through the **Omni SDK**. This ensures that the Agent Desktop and the CRM stay in perfect sync without manual state management.

- **Capabilities:** Dialing, Disposing, Transferring, and more.
- **Documentation:** For a full list of methods and event listeners, please refer to the **Omni Toolbar**

Option 2: Server-Side via Voice REST APIs

For advanced CTI capabilities or scenarios where the CRM backend needs to trigger actions, we provide a suite of **Voice REST APIs**. These endpoints allow for direct control over the telephony layer.

```
Note: These APIs require the sessionid and userCRTObjectId, which are
provided to the CRM via the initial URL query parameters upon loading.
```

Manual Dial API

Initiates an outbound manual call directly from the agent desktop or CTI toolbar.

Endpoint

Method	URL
POST	https://<your-ameyo-domain>/ameyorestapi/voice/manualdial

Headers

Header	Required	Description
Authorization	**Yes**	The one time / time based authentication token (Basic or Bearer) generated for the account.
sessionid	**Yes**	The active Agent Session ID obtained from the Knowledge Base query parameters.
Content-Type	**Yes**	application/json

Parameters

Field	Type	Required	Description
sessionId	string	**Yes**	The Authentication Token (matches the Authorization header).
shouldAddCustomer	boolean	**Yes**	Whether to auto-create a customer record if one does not exist.
campaignId	number	**Yes**	The active Agent campaign ID obtained from the Knowledge Base query parameters.
requestId	string	**Yes**	A client-generated unique UUID for idempotency.
userCRTObjectId	string	**Yes**	The active Agent userCRT ID obtained from the Knowledge Base query parameters.
searchable	string	**Yes**	The primary search key (usually the phone number).
phone	string	**Yes**	The destination phone number to dial.
additionalParams	object	No	Custom metadata passed from the CRM to the Contact Center application.
selectedSearchables	array	No	List of selected searchables if applicable.

```
curl --location 'https://<your-ameyo-domain>/ameyorestapi/voice/manualdial' \
--header 'authorization: Basic c2FjaGleX2FkcWl5X2F6dXJlOnNhY2hpbl9hZG1pbl9henVyZQ==' \
--header 'content-type: application/json' \
--header 'sessionid: d108-6955296d-ses-Harmony1-appId-App-Server-FYyMSB770C-Lv9Tb0W5Dvv0
--data-raw '{
    "sessionId": "Basic c2FjaGluX2FkbWluX2F6dXJlOnNhY2hpbl9hZG1pbl9henVyZQ==",
    "shouldAddCustomer": false,
    "campaignId": 60,
    "requestId": "manual-dialf09f77ef-ce66-498a-9bf5-a160b7ad7f48",
    "userCRTObjectId":"d108-6955296d-ses-Harmony1-appId-App-Server-FYyMSB770C-Lv9Tb0W5Dv
    "searchable": "8447380993",
    "contextVsContextAdditionalData": {},
    "additionalParams": {},
    "selectedSearchables": [],
    "phone": "8447380993"
}'
```

Dispose Call API

Disposes of an active call without initiating a subsequent action. This endpoint is required for CRM-driven disposition flows.

Endpoint

Method	URL
POST	https://<your-ameyo-domain>/ameyorestapi/voice/disposeCall

Export to Sheets

Headers

Header	Required	Description
Authorization	**Yes**	The one time / time based authentication token (Basic or Bearer) generated for the account.
sessionid	**Yes**	The active Agent Session ID obtained from the Knowledge Base query parameters.
Content-Type	**Yes**	application/json

Parameters

Field	Type	Required	Description
sessionId	string	**Yes**	The Authentication Token (matches the Authorization header).
campaignId	number	**Yes**	The active Agent campaign ID obtained from the CRM URL query parameters.
userCRTObjectId	string	**Yes**	The active Agent userCRT ID obtained from the CRM URLquery parameters.
crtObjectId	string	**Yes**	The active CRT ID obtained from the CRM URL query parameters.
dispositionCodeId	number	**Yes**	The internal numeric ID of the disposition code.
dispositionParams	object	No	Custom attributes associated with the disposition.
notes	string	No	Agent notes or comments regarding the call.
requestId	string	**Yes**	A client-generated unique UUID for idempotency.

```
curl --location 'https://<your-ameyo-domain>/ameyorestapi/voice/disposeCall' \
--header 'authorization: Basic c2FjaGluX2F2bWl2X2F6drJlOnNhY2hpbl9hZG1pbl9henVyZQ==' \
--header 'content-type: application/json' \
--header 'sessionid: d108-6955296d-ses-Harmony1-appId-App-Server-FYyMSB770C-Lv9Tb0W5Dvv0
--header 'Cookie: __METADATA__=e3a3f5a5-71fd-407f-91d7-6556534ead65' \
--data-raw '{
    "sessionId": "Basic c2FjaGluX2FkbWluX2F6dXJlOnNhY2hpbl9hZG1pbl9henVyZQ==",
    "campaignId": 60,
    "userCRTObjectId":"d108-6955296d-ses-Harmony1-appId-App-Server-FYyMSB770C-Lv9Tb0W5Dv
    "crtObjectId": "d604-69202d07-vce-27",
    "dispositionCodeId": 11,
    "dispositionParams": {},
    "requestId": "dispose-call-d0996013-d2a7-41d9-b157-58e254f6d5af",
    "notes": ""
}'
```

Dispose and Manual Dial API

Disposes of the currently active call and immediately triggers a new manual dial request.

Endpoint

Method	URL
POST	https://<your-ameyo-domain>/ameyorestapi/voice/disposeAndManualDial

Headers

Header	Required	Description
Authorization	**Yes**	The one time / time based authentication token (Basic or Bearer) generated for the account.
sessionid	**Yes**	The active Agent Session ID obtained from the Knowledge Base query parameters.
Content-Type	**Yes**	application/json

Parameters

Field	Type	Required	Description
campaignContextId	number	Yes	The numeric ID of the campaign associated with the call being disposed.
userCRTObjectId	string	Yes	The active Agent userCRT ID obtained from the CRM URLquery parameters.
crtObjectId	string	Yes	The active CRT ID obtained from the CRM URL query parameters.
dispositionCode	string	Yes	The string label of the disposition (e.g., "Sale", "Callback").
phone	string	Yes	The next phone number to dial immediately.
customerId	number	No	The active customer Id obtained from the CRM URL query parameters.(send -1 if unknown).
notes	string	No	Agent notes.
requestId	string	Yes	A client-generated unique UUID for idempotency.
additionalParams	object	No	Custom metadata dictionary.

```
curl --location 'https://<your-ameyo-domain>/ameyorestapi/voice/disposeAndManualDial' \
--header 'authorization: c2FjaGluX2FkbWlFX2F6FXJlOn5hY2hpbl9hZG1pbl9henVyZQ==' \
--header 'content-type: application/json' \
--header 'sessionid: d108-6955296d-ses-Harmony1-appId-App-Server-FYyMSB770C-Lv9Tb0W5Dvv0
--data-raw '{
    "campaignContextId": 270,
    "userCRTObjectId": "d108-6955296d-ses-Harmony1-appId-App-Server-FYyMSB770C-Lv9Tb0W5D
    "crtObjectId": "d604-69202d07-vce-22",
    "dispositionCode": "Sale",
    "phone": "7905690272",
    "customerId": "43800317962753",
    "requestId": "15da9c76-4a95-44a2-9972-915b953eabc1",
    "additionalParams": null,
    "notes": ""
}'
```

3.3. Omni Toolbar

Overview

The **Omni Toolbar** is Exotel's unified agent workspace that embeds directly into customer CRMs or internal tools to handle **voice, chat, WhatsApp, and future channels** from a single, consistent interface.

From a customer's point of view, the Omni Toolbar:

- Replaces fragmented CTI and channel-specific widgets
- Preserves **context across channels**
- Reduces agent training and handling time
- Acts as the **single control surface** for all interactions

If you've previously used the **Generic Toolbar Integration**, think of Omni Toolbar as:

```
The same CTI foundation, evolved for omnichannel, scale, and future
extensibility.
```

Who Should Use Omni Toolbar

The Omni Toolbar is designed for:

- Customers embedding Exotel into **CRMs.**
- Customers building **custom agent desktops**
- Customers migrating from **voice-only CTI** to omnichannel
- Enterprises that want **one integration** instead of per-channel integrations

If your agents switch tabs, windows, or tools to manage calls, chats, and WhatsApp today – Omni Toolbar is meant for you.

Core Capabilities

- Embeds as a **floating or docked toolbar** inside your application

- Manages **agent login, presence, and state**
- Handles **inbound and outbound interactions**
- Supports **voice, chat, WhatsApp** (channel-aware UI)
- Triggers **screen pops** and contextual actions
- Maintains **interaction lifecycle events**
- Supports **future extensibility** without reintegration

Setup for CRM Integration (Iframe Model)

The Ameyo Omni Web Toolbar is designed to be embedded within an **iframe** on the third-party CRM's web page.

1. **Iframe Container:** The third-party CRM must have a <div> element with the ID **ameyoIframeDiv** to contain the toolbar iframe.
   Sample URL :- https://<**yourECCDomainName**>/omniapp/toolbar/index.html

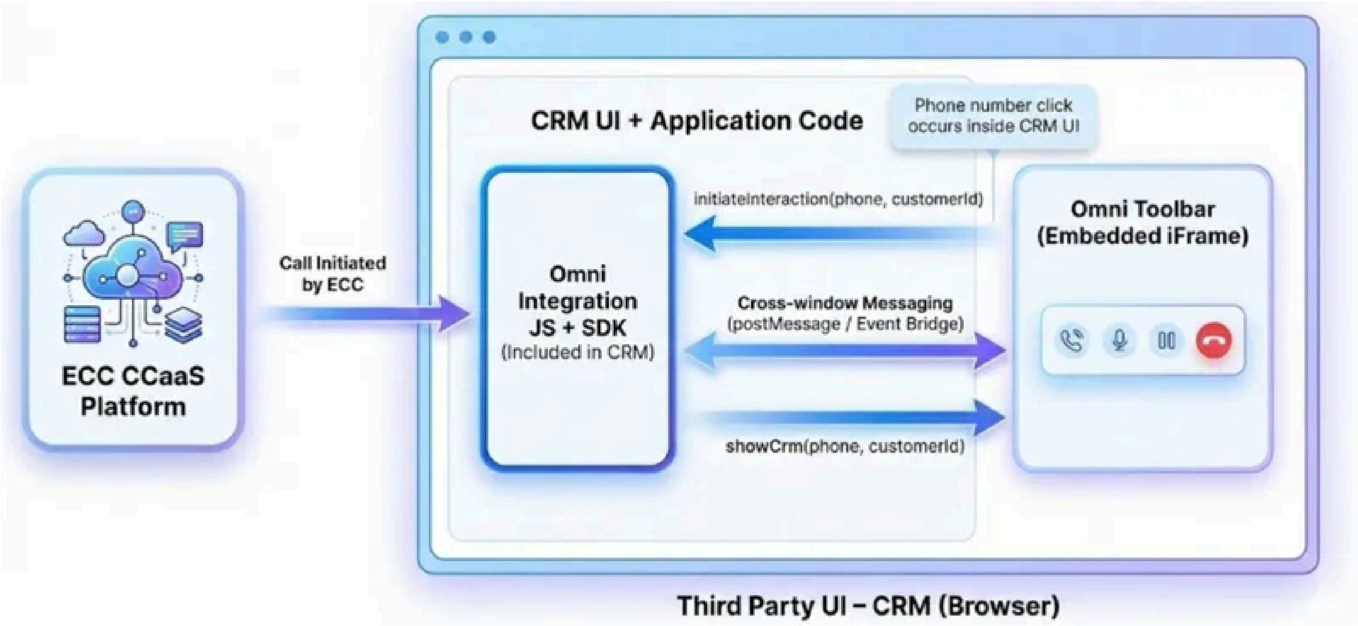
   > Currently, we support a minimum size equivalent to the iPhone SE dimensions i.e., 375 (width) × 667 (height).

2. **Initialization Script:** The integration requires importing the SDK JavaScript file and initializing the SDK with the **crm context**:JavaScript

- Omni Toolbar is embedded into your application UI
- It runs independently but communicates via defined events
- Your application remains the **source of truth for navigation and data**

```
const sdkConfig = {
  context: "crm",
  origin: "https://<your_crm_domain>",
  instanceId: "ameyoIframe" // The iframeId embedding the CRM
};

// After loading SDK module...
sdk.initialize(sdkConfig);
```



Event-Driven Interaction Model

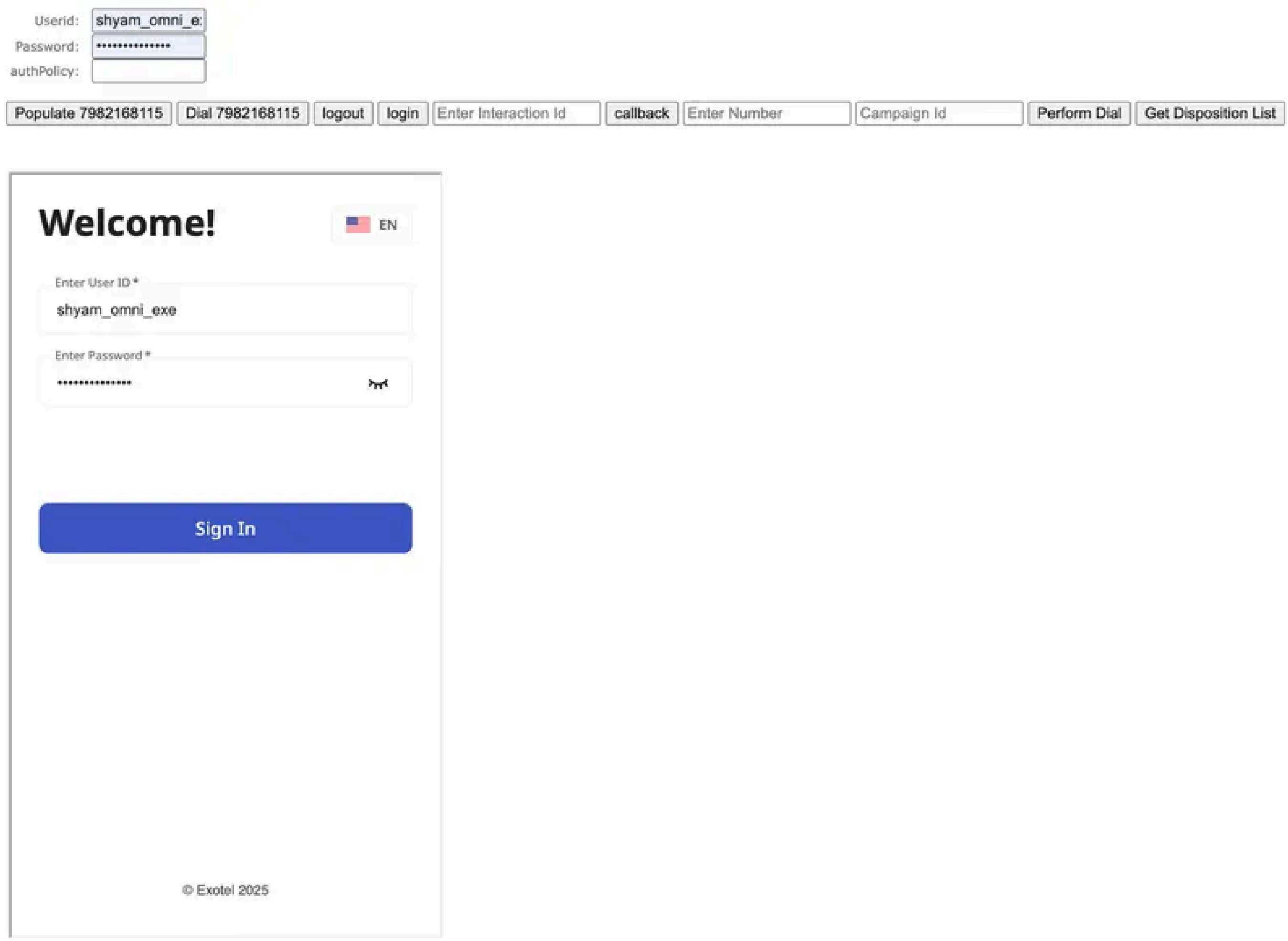
Omni Toolbar communicates via **events**, not tight coupling.

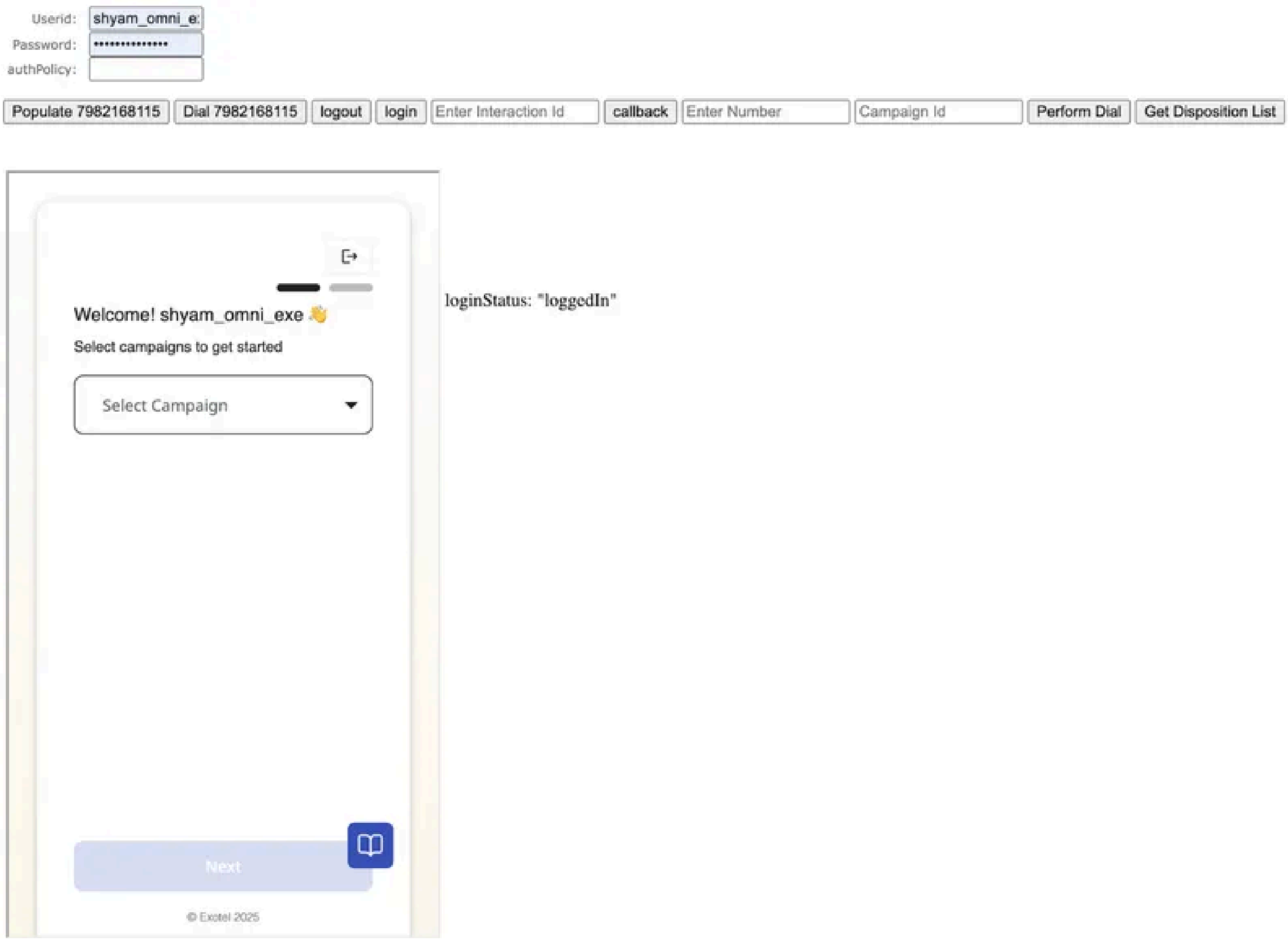
Typical lifecycle:

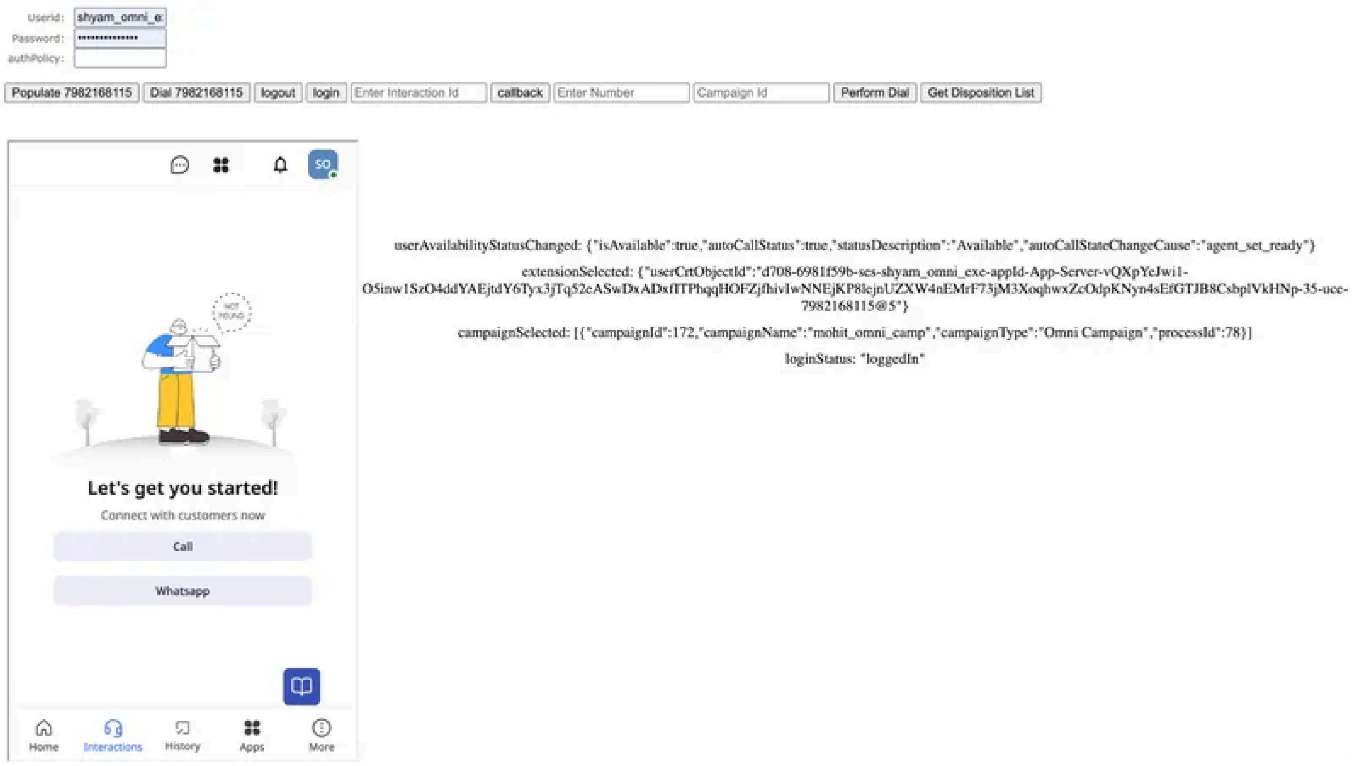
1. Toolbar initializes
2. Agent logs in
3. Interaction arrives (call/chat/WhatsApp)
4. Toolbar emits interaction event
5. Host app reacts (screen pop, fetch data)
6. Agent completes interaction
7. Toolbar emits wrap-up event

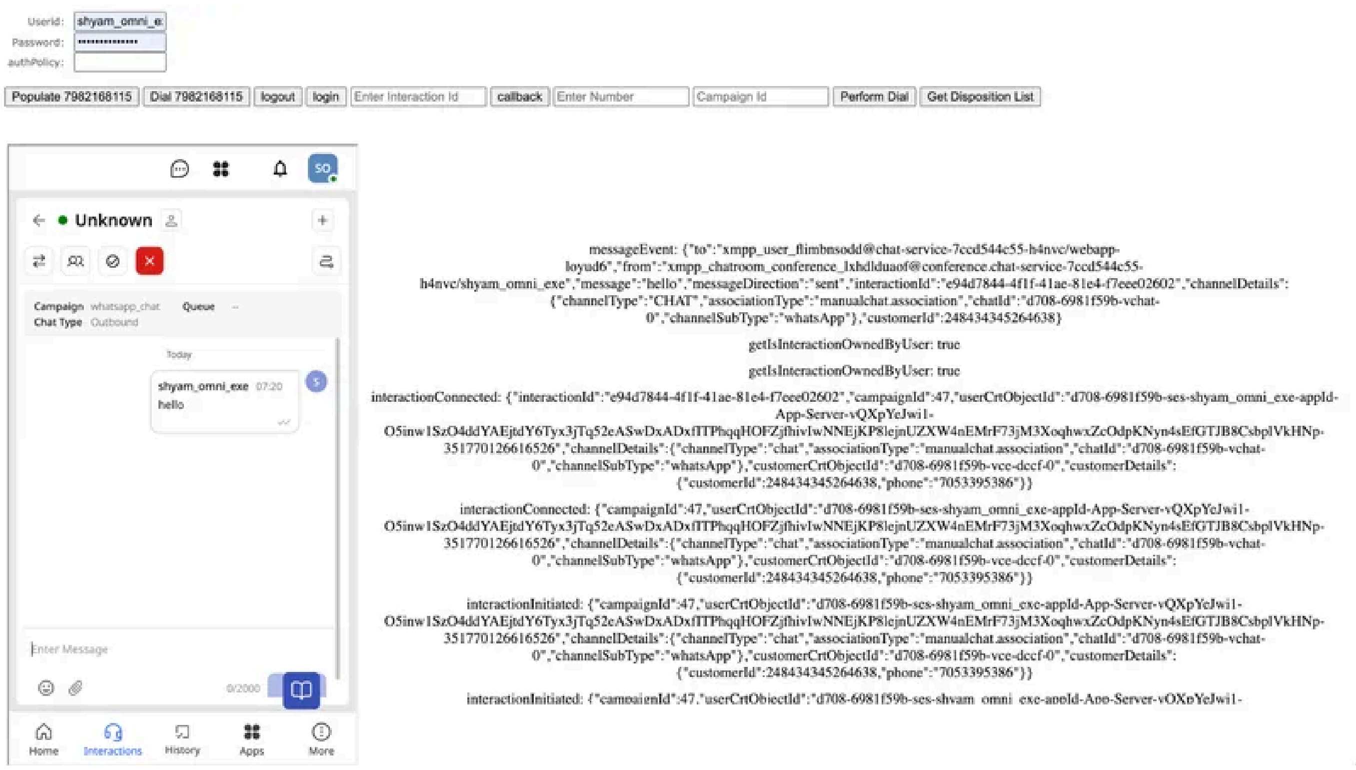
This model keeps integrations stable even as channels evolve.

Screenshots of a sample integration and how events are passed


Userid:
shyam_omni_e:
Password:
authPolicy:
Populate 7982168115
Dial 7982168115
logout
login
Enter Interaction Id
callback
Enter Number
Campaign Id
Perform Dial
Get Disposition List
Welcome!
EN
Enter User ID *
shyam_omni_exe
Enter Password *
Sign In
© Exotel 2025


Userid: shyam_omni_e
Password:
authPolicy:
Populate 7982168115
Dial 7982168115
logout
login
Enter Interaction Id
callback
Enter Number
Campaign Id
Perform Dial
Get Disposition List
Welcome! shyam_omni_exe
Select campaigns to get started
Select Campaign
Next
© Exotel 2025
loginStatus: "loggedIn"


Userid: shyam_omni_e
Password:
authPolicy:
Populate 7982168115
Dial 7982168115
logout
login
Enter Interaction Id
callback
Enter Number
Campaign Id
Perform Dial
Get Disposition List
SO
Let's get you started!
Connect with customers now
Call
Whatsapp
Home
Interactions
History
Apps
More
userAvailabilityStatusChanged: {"isAvailable":true,"autoCallStatus":true,"statusDescription":"Available","autoCallStateChangeCause":"agent_set_ready"}
extensionSelected: {"userCrtObjectId":"d708-6981f59b-ses-shyam_omni_exe-appId-App-Server-vQXpYeJwi1-O5inw1SzO4ddYAEjtdY6Tyx3jTq52eASwDxADxfITPhqqHOFZjfhivIwNNEjKP8lejnUZXW4nEMrF73jM3XoqhwxZcOdpKNyn4sEfGTJB8CsbpIVkHNp-35-uce-7982168115@5"}
campaignSelected: [{"campaignId":172,"campaignName":"mohit_omni_camp","campaignType":"Omni Campaign","processId":78}]
loginStatus: "loggedIn"



Core Integration Usecases & SDK Mapping

The following table maps the actions required for standard Generic Toolbar Integration to the **Omni SDK APIs** and **Events**:

Generic Toolbar Action	SDK Interaction Flow	Omni SDK API / Event	Category
CRM to Ameyo (e.g., Single Sign-On)	The CRM system initiates a request to authenticate and log in the agent into the Omni Toolbar.	login / forceLogin	Session Management
CRM to Ameyo (e.g., Click-to-Dial)	The CRM initiates an action (e.g., clicking a phone number) to start a new outbound interaction.	initiateInteraction	Interaction Management
Ameyo to CRM (e.g., Screen Pop)	The Ameyo platform (on call arrival or connection) sends a real-time notification to the CRM to fetch and display customer details.	showCrm (Event)	CRM & Record Handling
CRM to Ameyo (e.g., Call Disposition)	The CRM collects wrap-up data and sends it back to the Omni Toolbar to officially end the interaction in the Ameyo system.	disposeInteraction	Interaction Management

Initialise SDK

API Name	Params	Description	Sample Response

initialize	config = {
	 context: string ("crm")
	 origin: string
	 instanceId: string (default is ameyoIframe)
	 }
		To initialise the SDK	--

setBaseUrl	url: string		--
getBaseUrl	no params required		--
onLoadHandler	isLoaded: boolean	To inform the toolbar/UI that the App/CRM is loaded	--

Session and Authentication APIs

Session

SSO in Ameyo requires the user ID to be the same in both Ameyo and the CRM. The CRM calls the login API.

API Name	Params	Description	Sample response
login	username: string password: string authPolicy: string (optional)	Login a user	Result Successfully Logged In
force	user	ForceLo	Result Successfully Logged In

API Name	Params	Description	Sample response
Login	name: string password: string authPolicy: string (optional)	gin a user	
getSelect	no param	Get extension	{ callContextId: phone: "1234567890" }

API Name	Params	Description	Sample response
edExtension	s required	data while extension selection or while change selection	
logout	no params required	Logout user	Result Logout successful
getCurren	no param	Get current logg	{ "userId": "executive", "userType": "Executive", "userName": "executive" }

API Name	Params	Description	Sample response
tUser	s required	ed in user data	
getSelectedCampaigns	no params required	Get campaign data while campaign selection or while changing campaign	[{ "campaignId": 47, "campaignName": "whatsapp_chat", "campaignType": "Omni Campaign", "processId": 5 }]
getAge	no para	Get all age	{ "userId": "executive", "userType": "Executive", "terminalInfo": null, "userName": "executive", "breakStatuses": ["Coffee-break", "Dinner-Break", "Lunch-Break"], "readStatuses": ["Available"],

API Name	Params	Description	Sample response
ntSession	ms required	nt session data	"campaignInfos": [{ "campaignName": "whatsapp_chat", "campaignId": 47, "campaignType": "Omni Campaign", "dialTimeOut": 30, "blendGroupId": null, "defaultWorking": false, "primaryCampaign": false, "processId": 5, "wrapTimeout": 10, "acwConnected": 30, "acwNotConnected": 30, "isAcwConnectedEnabled": false, "localIVRInfos": [], "campaignTPVBean": { "campaignId": 47, "tpvName": [], "tpvPhone": [] }, "acwConnectedEnabled": false }], "telephonyChannel": null, "knowledgeBaseURL": "https://www.ameyo.com/", "pingInterval": 20000, "pingNotificationTimeout": 60000, "sessoinTimeout": 180000, "commandTimeout": 20000, "maxPhoneLengthAllowed": 20, "transferConfiguration": null, "conferConfiguration": null, "maxNumberOfCharactersToDisplayInCampaign": 999, "preContextId": null, "prePhone": null, "maskedPrivileges": [], "defaultReady": false, "addCustomerToDial": true, "dialPhoneEnable": true, "allowInternalChat": true, "allowLocalIVR": true, "allowSuccessiveCalls": true, "overrideSelfCallBack": true, "shouldHoldBeforeConfer": false, "shouldUnholdBeforeTransfer": false, "holdAllOnConferEnable": false, "shouldAllowTransferToAssignedCampaignsOnly": false, "shouldDisableOnVrDown": true, "allowScreenRecord": false, "autoSelectSingleAssignedCampaign": true, "webUrl": "http://www.ameyosuite.com", "campaignMaskedPrivilegeMap": { "campaignId_127": [], "campaignId_47": [], "campaignId_172": [] }, "viewPrivileges": null, "loginProperties": {}, "preferences": {}, "srsIp": "localhost", "srsPort": "8889", "agentBreakTimeThreshold": "", }

Session Events

Event Name	Description	Sample Response
loginStatus	Gets the data for login status of the user	Login status event received Successfully logged in
userAvailabilityStatusChanged	Gets the data for change in user availability	User availability status changed: { "isAvailable": true, "autoCallStatus": true, "statusDescription": "Available", "autoCallStateChangeCause": "agent_set_ready" }
campaignSelected	Gets the selected campaign details, including campaign id and name	Campaign selected: [{ "campaignId": 172, "campaignName": "mohit_omni_camp", "campaignType": "Omni Campaign", "processId": 78 }]
extensionSelected	Gets the selected extension details, including extension id and name	Extension selected: { "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-mGaqeOT4CAaxYr1nqW1FaKjcL0gztSuLcfl3ZcCBXM3O2ut6iumdbSoeke0EOosCaaRlWZHdbaGlflgOqKnznhg0RLVX86n0YdVM2mEotM7hp2M1hx7C-6-uce-16" }

Interaction and Call Management APIs

These APIs cover the telephony operations initiated from the CRM (e.g., Click-to-Dial) or the subsequent call management functions.

Dialing and Interaction APIs

API Name	Params	Description	Sample response
initiateInteraction	channelType: string (“voice” | “chat”) channelDetails: { phone: string (optional) name: string (optional)	Initiates an interaction	{ "result": "success", "reason": "success" }

API Name	Params	Description	Sample response
	appId: string (optional) [key: string]: any } campaignId: number customerId: number (optional) sea		

API Name	Params	Description	Sample response
	rchables: string[] (optional) interactionRelationId?: string (optional) channelSubType: string (optional) additio		

API Name	Params	Description	Sample response
	nalInfo: any (optional)		
disposeInteraction	channelType: string (“voice” | “chat”) interactionId: string dispositionCodeId: numb	Disposes an interaction	Result ok

API Name	Params	Description	Sample response
	er notes: string (optional) additionalParams: { callbackTime: string phone: string isSelfCallback: boolean }		

API Name	Params	Description	Sample response
	(optional) callback time example: "2025-07-06T09:56:40+0530" (ISO 8601)		
disposeAndInitiate	interactionId: string dis	Disposes and initiates a new interaction	Result ok

API Name	Params	Description	Sample response
NewInteraction This API is for voice only	positionCodeId: number phone: string notes: string (optional)		
getAllLiveInteractionsForAgent	no params required	Fetches the details of all live interactions of the agent	[{ "data": { "id": "f57fe43b-bd7a-4c31-8498-5724ee2507", "title": null, "customer_id": -1, "last_assigned_user_id": "anshika_agent", "last_assigned_user_name": "anshika_agent", "contact_center_id": 5, "process_id": 78, "last_campaign_id": 172, "last_campaign_name": "mohit_omni_camp", "last_queue_id": 134, "last_queue_name": "dev_queue", "last_disposition": null, "interaction_media_id": "d764-6879fb2e-vchat-70", "channel_name": "CHAT", "sub_channel": "WebChat", "direction": "INCOMING", "date_added": "2025-07-18T11:13:36.753+00:00", "date_modified": "2025-07-18T11:13:37.116+00:00", "date_disposed": null, "interaction_relation_id": "f57fe43b-bd7a-4c31-8498-5724ee2507", "status": "OPEN", "reopen_count": 0,

API Name	Params	Description	Sample response
			"last_reopen_time": null, "first_assigned_date": "2025-07-18T11:13:37.085+00:00", "last_assigned_date": "2025-07-18T11:13:37.085+00:00", "linked_objects": [], "channel_data": null, "related_interactions": null, "linked_connector_id": 23, "last_application_instance_id": "App-Server-fWG2MYiupg", "additional_info": { "XMPP_METADATA": { "users": [{ "id": "anshika_agent", "name": "anshika_agent", "xmppId": "xmpp_user_4qlt4tvgcb@chat-service-7ccd544c55-h4nvc" }], "customer": { "name": "test", "xmppId": "xmpp_customer_conference_j6tncvidns" }, "xmppChatRoomId": "xmpp_chatroom_conference_jpmxwvgq34@conference.chat-service-7ccd544c55-h4nvc" }, "CHAT_CUSTOMER_CRT_OBJECT_ID": "d764-6879fb2e-dccf-37" } }, "http_code": 200, "error_data": null }]
getCurrentInteraction	no params requires	Fetches the details of the interaction that the agent is currently on	{ "data": { "id": "87b41b89-8a79-44dd-a8d1-2ba7882507", "title": null, "customer_id": 735076692678909, "customer_name": "anshika", "last_assigned_user_id": "anshika_agent", "last_assigned_user_name": "anshika_agent", "contact_center_id": 5, "process_id": 78, "last_campaign_id": 172, "last_campaign_name": "mohit_omni_camp", "last_queue_id": 0, "last_disposition": null, "interaction_media_id": "d764-6879fb2e-vcall-105", "channel_name": "VOICE", "sub_channel": null, "direction": "OUTGOING", "date_added": "2025-07-18T11:12:34.898+00:00", "date_modified": "2025-07-18T11:13:04.457+00:00", "date_disposed": null, "interaction_relation_id": "87b41b89-8a79-44dd-a8d1-2ba7882507", "status": "OPEN", "reopen_count": 0, "last_reopen_time": null, "first_assigned_date": "2025-07-18T11:12:34.898+00:00", "last_assigned_date": "2025-07-18T11:13:04.457+00:00", "linked_objects": [], "channel_data": null, "related_interactions": null, "linked_connector_id": 23,

API Name	Params	Description	Sample response
			"last_application_instance_id": "App-Server-fWG2MYiupg", "additional_info": { "VOICE_CUSTOMER_CRT_OBJECT_ID": "d764-6879fb2e-vce-91" } }, "http_code": 200, "error_data": null }
getIsInteractionOwnedByUser	no params required	Will get the ownership status of the interaction, for example, in case of confer, the call lands on agent A and then agent B is added in conference, if we check ownership status at agent B it will show false.	TRUE

Interaction Lifecycle Events

These events are crucial for synchronizing the state of a call between the Ameyo toolbar and the external CRM:

interactionInitiated	Gets the data for the interaction, the channel, the user and the customer for which the interaction has been initiated	Integration interactionInitiated event received { "interactionId": "cd787e9a-fac0-4f1e-b823-a998532507", "campaignId": 172, "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54lF7ZKZsX8YysJYDp3iQlNfZLUXCG22OlMkuD4l7coVcm70oA5lKbvMNuAm2tWloo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5", "channelDetails": { "channelType": "voice", "associationType": "manualdial.association", "callId": "d279-6874b064-vcall-20" }, "customerCrtObjectId": "d279-6874b064-vce-6", "customerDetails": { "customerId": 735076692678909, "name": "anshika", "phone": "1234567890" } }
callRinging	Fetches information about the call, the user and the customer to whom it was made while the call is in ringing state	callRinging event received { "interactionId": "7ea1bd1b-5983-4f23-970a-0d99642507", "campaignId": 172, "queueId": null, "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-rrtuxJxpN7ecoIjPtfSj2ubCbmGbWT4PfcAuG03kNDQFXKjfbL7GX0DHRQduSlvEYA7vA4AzPQhlTSSmX8Z6fNw44P8lwuk5bdjknZL4WXi0cLjlyj8J-78-uce-1234567890@5", "channelDetails": { "channelType": "voice", "associationType": "manualdial.association", "callId": "d279-6874b064-vcall-35" }, "customerCrtObjectId": "d279-6874b064-vce-19", "customerDetails": { "customerId": 735076692678909, "name": "anshika", "phone": "1234567890" } }
interactionConnected	Gets the data for the interaction, the channel, the user, and the customer to which the interaction is connected to	interactionConnected event received { "interactionId": "c15a278d-66ff-47ed-a0c0-cba5492507", "campaignId": 172, "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54lF7ZKZsX8YysJYDp3iQlNfZLUXCG22OlMkuD4l7coVcm70oA5lKbvMNuAm2tWloo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5", "channelDetails": { "channelType": "voice", "associationType": "manualdial.association", "callId": "d279-6874b064-vcall-21" }, "customerCrtObjectId": "d279-6874b064-vce-7", "customerDetails": { "customerId": 735076692678909, "name": "anshika", "phone": "1234567890" } }

interactionDisconnected	Gets the data for the interaction, the channel, the user, and the customer from which the interaction is disconnected	interactionDisconnected event received { "interactionId": "cd787e9a-fac0-4f1e-b823-a998532507", "campaignId": 172, "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54IF7ZKZsX8YysJYDp3iQINfZLUXCG22OIMkuD4l7coVcm70oA5lKbvMNuAm2tWIoo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5", "channelDetails": { "channelType": "voice", "associationType": "manualdial.association", "callId": "d279-6874b064-vcall-20" }, "customerCrtObjectId": "d279-6874b064-vce-6", "customerDetails": { "customerId": 735076692678909, "name": "anshika", "phone": "1234567890" } }
interactionDisposed	Gets the data for the interaction, the channel, the user, and the customer for which the interaction is disposed along with the disposition details of the interaction	interactionDisposed event received { "interactionId": "c15a278d-66ff-47ed-a0c0-cba5492507", "campaignId": 172, "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54IF7ZKZsX8YysJYDp3iQINfZLUXCG22OIMkuD4l7coVcm70oA5lKbvMNuAm2tWIoo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5", "channelDetails": { "channelType": "VOICE", "associationType": "manualdial.association", "callId": "d279-6874b064-vcall-21" }, "dispositionDetails": { "dispositionClass": "Closed", "dispositionCode": 588, "callNotes": "" } }
interactionEnded	Gets the data for the interaction, the channel, the user, and the customer for which the interaction is ended	interactionEnded event received { "campaignId": 172, "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54IF7ZKZsX8YysJYDp3iQINfZLUXCG22OIMkuD4l7coVcm70oA5lKbvMNuAm2tWIoo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5", "channelDetails": { "channelType": "voice" }, "customerCrtObjectId": "d279-6874b064-vce-6", "customerDetails": {} }
custom	Gets the data for the	interactionLinked event received { "interactionId": "cd787e9a-fac0-4f1e-b823-a998532507", "campaignId": 172, "userCrtObjectId":

erLinkedToInteraction	interaction and the channel to which the customer is linked to along with the linked customer details	"d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54lF7ZKZsX8YysJYDp3iQlNfZLUXCG22OIMkuD4l7coVcm70oA5lKbvMNuAm2tWloo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5", "channelDetails": { "channelType": "voice", "associationType": "manualdial.association", "callId": "d279-6874b064-vcall-20" }, "customerCrtObjectId": "d279-6874b064-vce-6", "customerDetails": { "customerId": 735076692678909, "name": "anshika", "phone": "1234567890" } }
customerUnlinkedToInteraction	Gets the data for the interaction and the channel from which the customer is unlinked from along with the unlinked customer details	{ "interactionId": "83937022-6ddd-4194-9395-4bc7c72507", "campaignId": 172, "queueId": null, "userCrtObjectId": "d279-68786f8d-ses-executive-appId-App-Server-pHnEm4nGW4-lpuUJTau5NwCQc4keOWJiogD6JtHDZlfvqD2gi4lwtBk5ApW5T8ULxL8dVwFUi6AZVOB5hxY4ITuO1VT80k9aYcExjpcgpVwOY8w6qpq5VGullWInaS7-940-uce-1234567890@5", "channelDetails": { "channelType": "VOICE", "associationType": "manualdial.association", "callId": "d279-68786f8d-vcall-362" }, "customerCrtObjectId": "d279-68786f8d-vce-329", "customerDetails": { "customerId": null, "name": null, "phone": "1234567890" } }
interactionTransferred	Gets the data for the interaction, the user and the channel for which the interaction has been transferred along with the transfer details	interactionTransferred event received { "interactionId": "c15a278d-66ff-47ed-a0c0-cba5492507", "transferType": "transferToUser", "transferContextId": "moh5", "channelType": "VOICE", "channelDetails": { "callId": "d279-6874b064-vcall-21", "crtObjectId": "d279-6874b064-ses-moh5-appId-App-Server-WBOSWo8hqn-mbns0GzYPxAjBULswJQoTSNn7likTcaj8qmjqxrzBvwaL2lisgAw70A48bvDjlbDsj4PZNpo40HUAKxF5Zd4wa7keO8V0dcN2STUW1vqMt6wHA5DuXd0-22-uce-moh5cRJkUZ5Z@6", "memberCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54lF7ZKZsX8YysJYDp3iQlNfZLUXCG22OIMkuD4l7coVcm70oA5lKbvMNuAm2tWloo0cVUU79WCCClhDGFs

		SKQmgc1M2RkHa5y-53-uce-1234567890@5" }, "campaignId": 172 }
interactionConferred	Gets the data for the interaction, the user and the channel for which the interaction has been conferred along with the confer details	Interaction conferred { "interactionId": "0c431e04-e78d-454b-a0ec-bc2aa72507", "conferedType": "conferToPhone", "status": "initialized", "conferedContextId": "8368941654", "channelType": "VOICE", "channelDetails": { "callId": "d279-6874b064-vcall-23", "crtObjectId": "d279-6874b064-vce-11", "memberCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54lF7ZKZsX8YysJYDp3iQlNfZLUXCG22OIMkuD4l7coVcm70oA5lKbvMNuAm2tWloo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5" }, "campaignId": 172 }
getIsInteractionOwnedByUser	Automatically checks ownership status for current interaction, ie, interaction on which agent's screen is	FALSE

CRM & Record Handling

Screen Pop and Customer Information

The **Screen Pop** functionality is a core use case, where the CRM displays customer information upon a call's arrival or initiation.

API Name	Params	Description	Sample response
setRecordInfoForInteraction	interactionId: string recordId: string recordType: string recordName: string additionalInfo: any (optional)	Stores the resolved customer details for the current interaction	{ "data": "{\"crtObjectId\":\"d279-68786f8d-vce-49\",\"info\":{\"recordId\":\"1\",\"recordName\":\"recordName\",\"recordType\":\"contact\"},\"campaignContextId\":172,\"actionType\":\"Add or Update\"}" }
showMultipleRecordsForInteraction	interactionId: string, recordsInfo: CRMRecordInfo[] CRMRecordInfo = { recordId: string, recordType: string, recordName: string,	Displays all identified customers and their associated information on the user interface during a live interaction.	//customer info will be displayed on UI

API Name	Params	Description	Sample response
	additionalInfo: any(optional) }		
showMultipleRecordsForPreview	phone: string, recordsInfo: CRMRecordInfo[] CRMRecordInfo = { recordId: string, recordType: string, recordName: string, additionalInfo: any(optional) }	Displays all customers and their associated information when customer details are previewed for a given phone number.	//customer info will be displayed on UI
sendCurrentC	CRMRecordInfo = { recordId:	sends currently opened	//displays currently opened object's info on UI

API Name	Params	Description	Sample response
RMRecordOpened	string, recordType: string, recordName: string, additionalInfo: any(optional) }	object information to UI	
getAllContacts	connectorId: string, objectId: string	Fetches all contacts for a connector Id and object Id	[[{ "field": "Id", "value": "003dL000000Pun1QAC", "type": "id" }, { "field": "FirstName", "value": "Jack", "type": "string" }, { "field": "LastName", "value": "Rogers", "type": "string" }, { "field": "Phone", "value": "(336) 222-7000", "type": "phone" }, { "field": "OtherPhone", "value": null, "type": "phone" }], [{ "field": "Id", "value": "003dL000000Pun3QAC", "type": "id" }, { "field": "FirstName", "value": "Andy", "type": "string" }, { "field": "LastName", "value": "Young", "type": "string" }, { "field": "Phone", "value": "(785) 241-6200", "type": "phone" }, { "field": "OtherPhone", "value": null, "type": "phone" }]]
getAllCases	connectorId: string, objectId: string	Fetches all cases for a connector Id and object Id	[[{ "field": "Id", "value": "500dL000001oNEhQAM", "type": "id" }, { "field": "Subject", "value": "Performance inadequate for second consecutive week", "type": "string" }, { "field": "Description", "value": null, "type": "textarea" }], [{ "field": "Id", "value": "500dL000001oNEiQAM", "type": "id" }, { "field": "Subject", "value": "Starting generator after electrical

API Name	Params	Description	Sample response
			failure", "type": "string" }, { "field": "Description", "value": null, "type": "textarea" }]]
createCase	objectType: string, data: any, connectorId: string	Creates a new case	--
updateObject	objectType: string, data: any, connectorId: string, objectId: string	Updates the attached object	--
searchCase	connectorId: string, objectType: string, search: string, searchField: string, limit: number, offset: number	Searches the case based on fields	[[{ "field": "Id", "value": "500dL000001oNEhQAM", "type": "id" }, { "field": "Subject", "value": "Performance inadequate for second consecutive week", "type": "string" }, { "field": "Description", "value": null, "type": "textarea" }]]
sendDataToCRM	contextData: any	Sends data from the app to CRM	//shows whatever data is sent in the API

API Name	Params	Description	Sample response
sendDataToApp	contextData: any	Sends data from CRM to app	//shows whatever data is sent in the API

CRM events

Event Name	Description	Sample Response
showCrm	Gets the configured CRM data along with the interaction, user and customer details	showCrm event received { "interactionId": "cd787e9a-fac0-4f1e-b823-a998532507", "userId": "executive", "campaignId": 172, "queueId": null, "userCrtObjectId": "d279-6874b064-ses-executive-appId-App-Server-WBOSWo8hqn-s0qX4novCTO6XZ1JkT54lF7ZKZsX8YysJYDp3iQINfZLUXCG22OIMkuD4l7coVcm70oA5lKbvMNuAm2tWloo0cVUU79WCCClhDGFsSKQmgc1M2RkHa5y-53-uce-1234567890@5", "channelDetails": { "channelType": "VOICE", "associationType": "manualdial.association", "callId": "d279-6874b064-vcall-20" }, "customerCrtObjectId": "d279-6874b064-vce-6", "customerDetails": { "customerId": 735076692678909, "name": "anshika", "phone": "1234567890" } }
openRecord	Opens the record on the CRM	--
onAttachButtonClick	Links current interaction with CRM customer	--
onUnAttachButtonClick	Unlinks current interaction with CRM customer	--
onCreateBut	To create object on CRM	--

Event Name	Description	Sample Response
tonClick		
getCurrentCRMRecordOpened	Fetches current CRM record	--
handlePreview	Searches customer on CRM on preview click	--
handleInteractionDisposed	Sends interaction disposition related information to CRM for activity dumping	--
receiveDataF	CRM receives	//data sent from app is seen in CRM

Event Name	Description	Sample Response
romApp	data from app	
receivedDataFromCRM	App receives data from CRM	//data sent from CRM is shown in App

Disposition Code Retrieval

The CRM can retrieve the two-level disposition codes (Disposition Class and Disposition Code) from Ameyo to display a custom disposition screen.

API Name	Params	Description	Sample response
getDispositionCodesWithDispositionClass	campaignId : number	Gets the disposition class and corresponding codes for a given campaignId	[{ "classId": 588, "className": "Status", "dispositionCodeBean": [{ "dispositionCodeId": 1344, "dispositionCodeName": "Open" }, { "dispositionCodeId": 1345, "dispositionCodeName": "Closed" }, { "dispositionCodeId": 820, "dispositionCodeName": "New" }] }]

Get Configurations

API Name	Params	Description	Sample response
getOmniCrmObjectConfiguration	no params required	Gets the details of all the objects for the crm	[{ "id": 33, "crm_connector_id": 23, "object_type": "Case", "objects_metadata": { "Case": { "is_visible": true, "field_metadata": [{ "field_name": "Id", "label": "Case ID", "crm_field_type": "String", "field_type": "id", "should_show_on_preview": true, "is_required_in_object_creation": false, "is_searchable": true, "standard_field_name": "", "priority": 1, "should_hide_on_object_creation": true, "complex_fields": null, "required_in_object_creation": false, "searchable": true }, { "field_name": "Subject", "label": "Subject", "crm_field_type": "String", "field_type": "string", "should_show_on_preview": true, "is_required_in_object_creation": true, "is_searchable": true, "standard_field_name": "", "priority": 2, "should_hide_on_object_creation": false, "complex_fields": null, "required_in_object_creation": true, "searchable": true }, { "field_name": "Description", "label": "Description", "crm_field_type": "String", "field_type": "textarea", "should_show_on_preview": true, "is_required_in_object_creation": false, "is_searchable": false, "standard_field_name": "", "priority": 3, "should_hide_on_object_creation": false, "complex_fields": null, "required_in_object_creation": false, "searchable": false }], "visible": true } }, "date_added": "2024-08-13T10:48:33.862+00:00", "date_modified": "2024-08-13T10:48:33.862+00:00", "ref": 1, "version": 1 }]
getOmniCrmConfiguration	no params required	Fetches the details of the crm configuration	{ "crm_connector_id": 23, "process_id": 78, "crm_type": "salesforce", "crm_domain": "https://exotel88-dev-ed.develop.my.salesforce.com", "crm_version": "", "syncing_enabled": false, "syncing_mode": "PERIODIC", "record_syncing_interval": 3600000, "record_syncing_complete": true, "batch_size": 1000, "max_number_of_distinct_recordings": 3, "date_added": "2024-08-13T10:46:39.613+00:00", "date_modified": "2025-01-15T10:30:53.334+00:00", "ref": 1, "version": 5 }

API Name	Params	Description	Sample response
getPhoneNumberForSearch	no params required	Fetches the customer phone that is added on the start interaction screen	//will get the phone number entered
getPrimaryColumnName	no params required	Fetches the primary column name	"id"

HTTP Communication

API Name	Params	Description	Sample Response
invokeECCApi	microservice: Microservices url: string method: string (“get” | “post” | “put” | “delete”) queryParams: Record<string, any> (optional) data: Record<string, any> | FormData (optional) isMultipart: boolean (optional) url is api path here Microservices{ APPSERVER = 'appserver', CMS = 'cms', CFS = 'cfs', DATA_ENGINE = 'data-engine', CALLBACK_SERVER = 'callback-server', CUSTOMER_MANAGER = 'customer-manager', CHAT_SERVICE = 'chat-service', INTERACTION_SERVICE = 'interaction-service', CHAT_BOSCH_SERVICE = 'chat-bosch-service' }	This API is used to invoke appserver/cms/cfs/data-engine/callback-server/customer-manager/chat-service/interaction-service/chat-bosch-service	{"searchables": ["id","first_name","phone1"] } //this is a sample response from one Appserver API
invokeThirdPartyApi	url: string method: string (“get” | “post” | “put” | “delete” | “patch”) headers: Record<string, string> (optional) queryParams: Record<string, any> (optional) data: Record<string, any> | FormData (optional) isMultipart: boolean (optional)	This API is used to invoke any third party API	{ "status": 200, "data": { "page": 2, "per_page": 6, "total": 12, "total_pages": 2, "data": [{ "id": 7, "email": "michael.lawson@reqres.in", "first_name": "Michael", "last_name": "Lawson", "avatar": "https://reqres.in/img/faces/7-image.jpg" }], "support": { "url": "https://contentcaddy.io?utm_source=reqres&utm_medium=json&utm_campaign=referral

API Name	Params	Description	Sample Response
			", "text": "Tired of writing endless social media content? Let Content Caddy generate it for you." } }, "statusText": ""

UI Methods

API Name	Params	Description	Sample Response
setLoginInfo	username: string token: string	Populates login info on UI	//data will be visible on login screen UI

populateChannelDetailsForNewInteraction	channelType: string
	channelSubType: string
	channelDetails: {
	 campaignId: string
	 appName: string
	 phone: string
	 name: string
	 }
		Populates the details to set details to start an interaction	//data will be visible on the start interaction screen on UI

setExtensionInfo	callcontextId: number phone: string	Populates the extension info on the select extension and the change extension screen	//data will be visible on the select/change extension screen on UI
triggerToastNotification	message: string variant: string ('success' | 'error' | 'warning' | 'info')	Triggers a toast notification for the user	//A toast notification will be visible on user screen
triggerDesktopNotification	notificationType: string notificationSubtype: string subject: string message: string	Triggers a desktop notification for the user	//A desktop notification will be visible on user screen

UI Events

Event Name	Description	Sample Response
windowVisibilityChanged	Checks if the current window is visible or not	{ isWindowVisible: true }
sendLifecycleEvent	Gets details like app is focused or not, loaded or not, when it is registered	"focused"
receiveData	Receives data from other apps	//Can see whatever is sent from the other app

Notification Events

Event Name	Description	Sample Response
notification	Triggers all the details regarding a bell notification including notification severity, type and message	Notification { "notificationId": "d279-6874b064-notify-2441", "displayMessage": "Call missed!! Customer Name: anshika Contact Number: 01234567890 Missed Call Time: Mon Jul 14 14:01:45 IST 2025", "severity": 2, "type": "application.notification.type", "subType": "notification.missed.call.type", "specificInfo": { "contextId": 52, "context": "MISSED_CALL", "campaignId": 172, "customerId": 735076692678909, "name": "anshika", "contact": "01234567890", "dateAdded": "2025-07-14 14:01:45.899", "phoneInfo": { "phone": "01234567890", "displayPhone": "01234567890" }, "additionalParams": { "notificationId": "d279-6874b064-notify-2441", "context": "MISSED_CALL" } }, "consolidationCount": 0, "dateAdded": "2025-07-14T08:31:48.925Z" }

Internal Communication

For custom logic and data synchronization between the embedded toolbar app and the main CRM page:

Srore Apps Data

API Name	Params	Description	Sample Response
get	key: string	Gets the value of a key from the local storage	Result "qwerty"
set	key: string value: string	Sets the key and it's corresponding value in the local storage	Result successfully set store item //the key can be checked in local storage
delete	key: string	Deletes a key from the local storage	Result successfully deleted store item //the key will now be removed from the local storage

AppInstance

API Name	Params	Description	Sample Response
getAllInfo	appId: string	Fetches information for all loaded apps	
getInfo	instanceId: string	Fetches information for app based on instanceId	
sendDataToApp	instanceId: string, contextData: any	Sends data from one app to the other app based on instanceId	//shows data sent in contextData

Events

Event Name	Description	Sample Response
receiveData	Receives data from other apps	//Can see whatever is sent from the other app

Error Handling

The SDK provides explicit error codes to aid in debugging integration issues.

Error Type	Error Code	Description
TIMEOUT	1000	Request took too long and timed out.
REQUIRED_FIELD_MISSING	2001	A mandatory field (e.g., phone in initiateInteraction) was not provided.
HTTP_ERROR	3000	An HTTP-related issue occurred (e.g., 404, 500).
DATA_NOT_FOUND	4000	Requested data could not be located or does not exist.
INVALID_ORIGIN	1003	Request came from an unauthorized or untrusted source.

4. Configuration Documents

4.1. Agent Guide

Agent Manual: Omni Version1

Overview

The Omni Agent Interface within the ECC product is designed to empower contact center agents with an intuitive and highly efficient interface. This interface enables seamless handling of customer interactions spanning multiple communication channels such as Voice channel, chat channel such as whatsApp, webchat, social chats etc providing agents with comprehensive access to customer information and case management. Integrated AI-powered tools further assist agents to enhance productivity and improve customer satisfaction.

The Agent has access to a web-based interface with telephony controls to make or receive calls of chats. When the agent receives a call, a CTI (Computer Telephony Integration) pop-up appears with customer information.

See AI - Powered Contact Center in Action

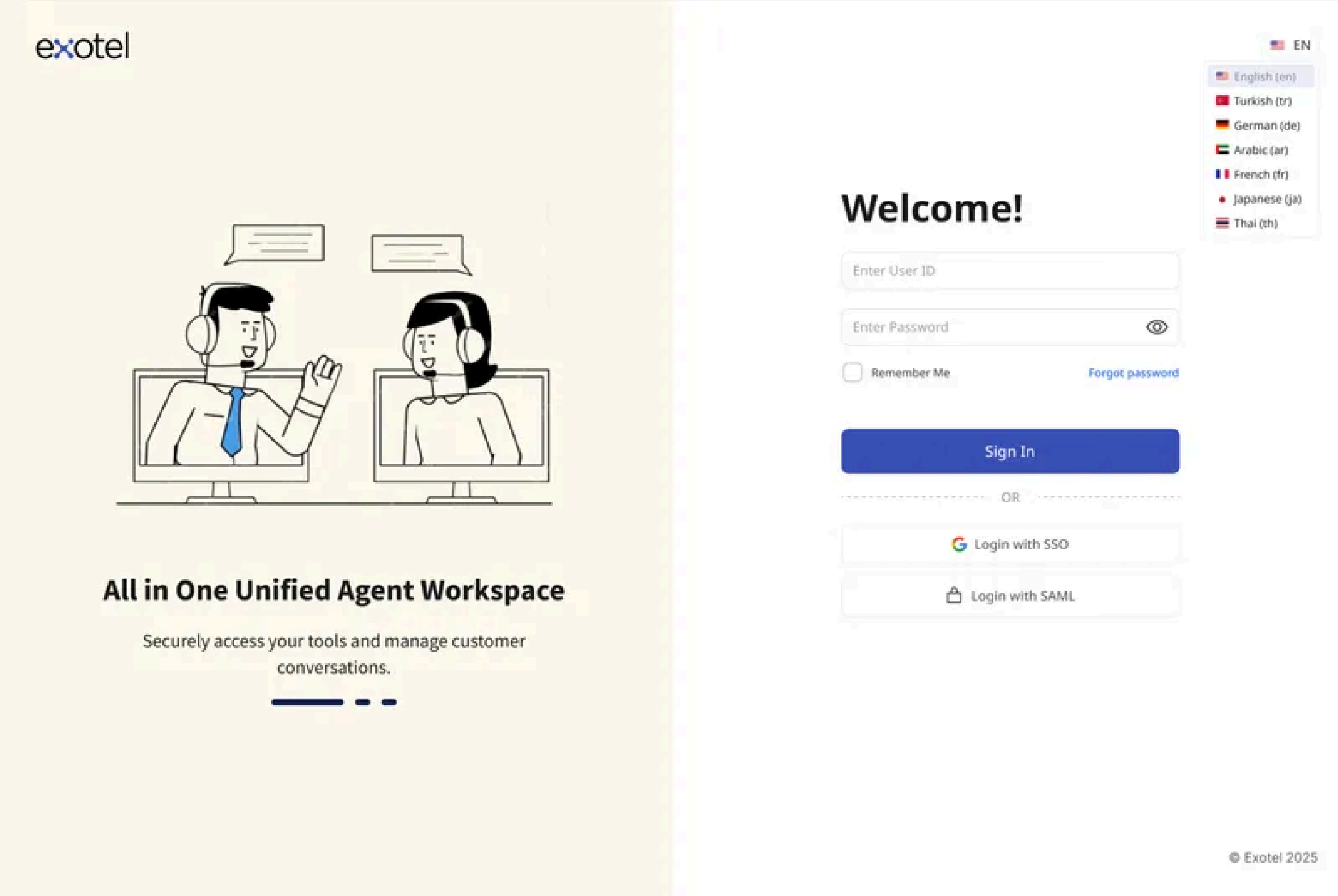
Agent web interface access includes:

1.Login

This section describes how to log in to the Omni Agent workspace.

1. To log in to the agent screen, open the URL provided by your administrator.
2. On the login screen, you can log in using one of the following methods as configured by your administrator:
    - **Normal Login:** Enter the "User ID" and "Password" provided to you and click "Login."

- **SAML (Licensable):** If enabled, click "SAML" to log in using your organization's third-party Identity Provider (IDP).
- **Google (Licensable):** If enabled, click "Google" to log in using your Google account via Single Sign-On.



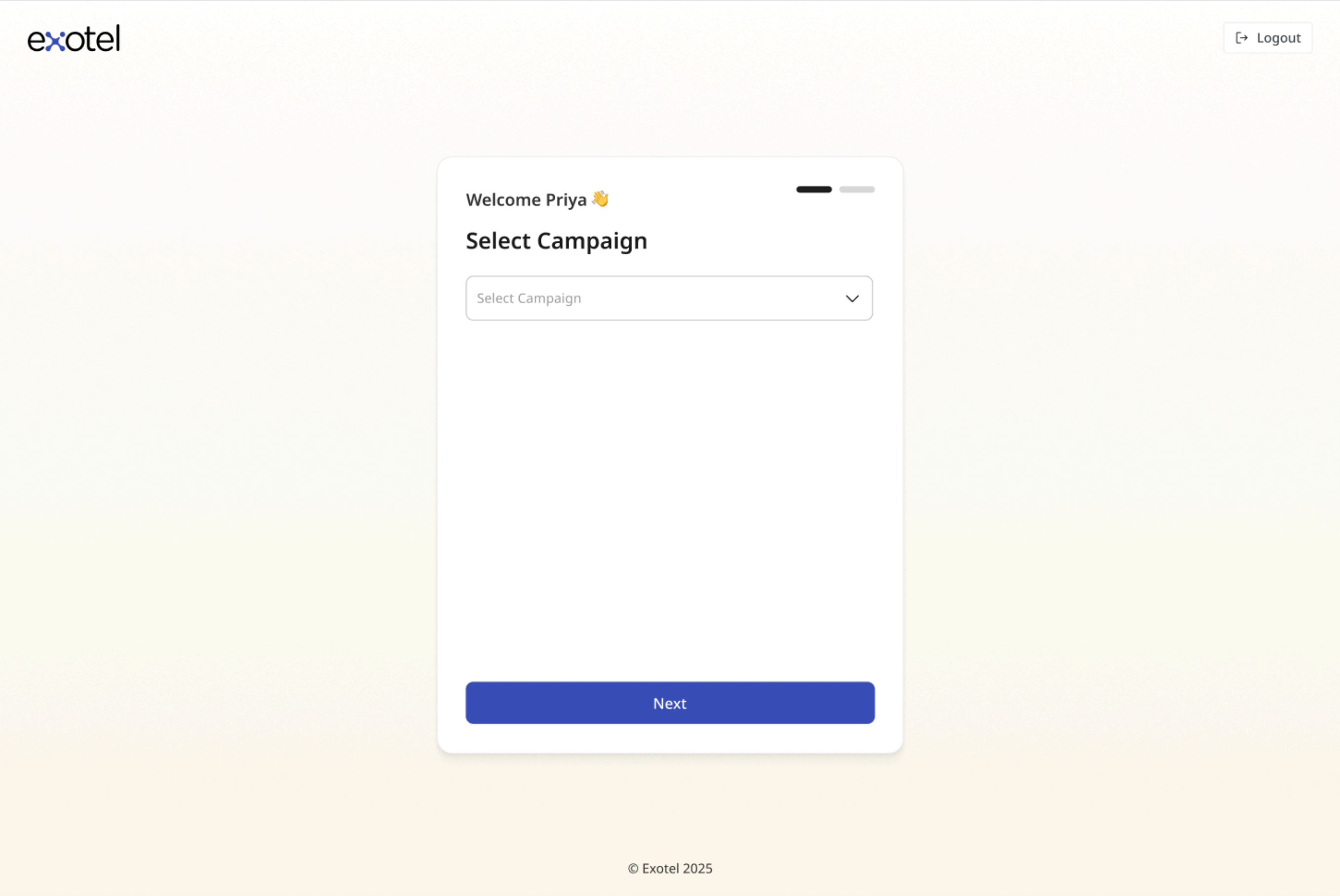
1.1 Selecting Your Campaign and Extension

After you log in for the first time, you may need to complete a one-time setup to select your campaigns and telephone extension.

1. Campaign Selection

A "Campaign Selection" pop-up will appear. You must select the campaigns you will be working on from the drop-down lists. You may be required to select:

- A Ticket/Case Campaign
- A Voice Campaign
- A Chat Campaign
  Click "Next" to proceed.



> You will not get the campaign selection pop up if it has been pre assigned from the administrator panel.

2. Extension Selection

If you are working in a Voice Campaign, an "Extension Selection" pop-up will appear.

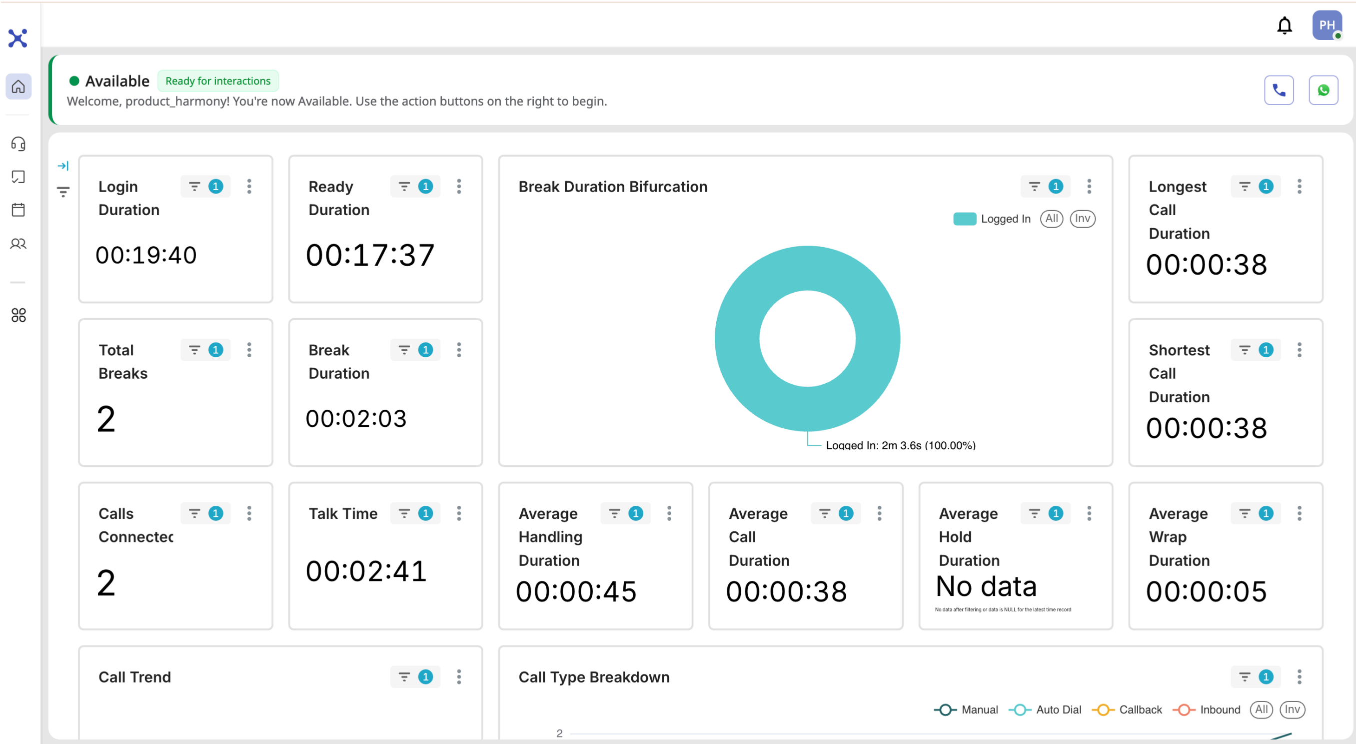
- **For WebRTC:** If your system uses WebRTC (browser-based phone), this step may be automatic.
- **For Softphone/Hardware Phone:** You must select your configured extension (e.g., "softphone1" or your desk phone number "1001") from the drop-down list. If you select a hardware phone, you may need to enter the phone number.

Click **"Next"**. You will now be directed to the Agent Homepage.

2.Getting Started: The Agent Homepage

The Homepage is the first screen you see after logging in. It is a dashboard designed to help you monitor your real-time performance metrics and session activity.

Depending on your assigned campaigns, your Homepage will display either **Voice-Only** or **Omnichannel metrics**. If your organization has CQA/AQM activated, you may also see a widget for your Quality metrics.



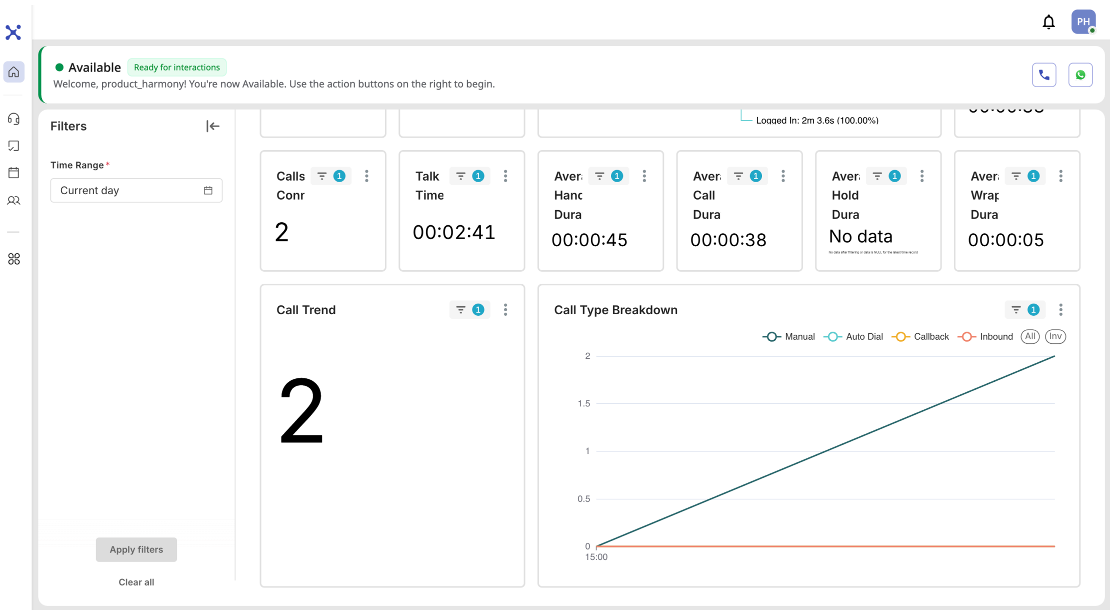
2.1 Voice-Only Self Monitoring Dashboard

If you are assigned to voice-only campaigns, your dashboard will include the following metrics:

- **Call Summary**
  - Total Calls Offered
  - Total Calls Connected
  - Connection Rate (%)
  - Average Call Duration
  - Longest Call Duration
  - Shortest Call Duration
- **Call Type Breakdown**
  - Manual Outbound Calls

- Inbound Calls
- Dialer Calls
- Callback Calls

- **Time Tracking**
  - Login Time
  - Talk Time
  - Idle Time
  - Break Time (Total)
  - Break Bifurcation (e.g., Tea / Lunch / Bio / Others)
- **Call Trend**
  - An hourly trend graph of your connected calls.



2.2 Omnichannel Self Monitoring Dashboard

If you are assigned to handle interactions across multiple channels (e.g., Voice, Chat, WhatsApp, Email), your dashboard will provide a comprehensive view:

- **Interaction Summary by Channel**
  - **Voice:** Total Offered, Connected, Average Duration, Longest Duration
  - **Chat:** Total Offered, Responded, Average Duration, Longest Duration
  - **Email:** Total Offered, Responded, Average Duration, Longest Duration
  - **WhatsApp:** Total Offered, Responded, Average Duration, Longest Duration

- **Channel Engagement Mix**
    - A chart showing the percentage of interactions you have handled per channel.
- **Time Tracking**
    - Login Time
    - Active Time (across all channels)
    - Idle Time
    - Break Time (Total) and its bifurcation
- **Productivity Metrics**
    - Average Concurrent Chats
    - First Response Time (for Chat, Email, WhatsApp)
    - Average Email Response Time
- **Interaction Trend**
    - An hourly trend graph of interactions you have handled, split by channel.

---

3. Understanding Your Workspace

The agent workspace is designed for efficiency, giving you all the tools and information you need in one unified interface. The workspace is made up of three key components: the Left Navigation Bar, the Top Navigation Bar, and the 4-Panel View.

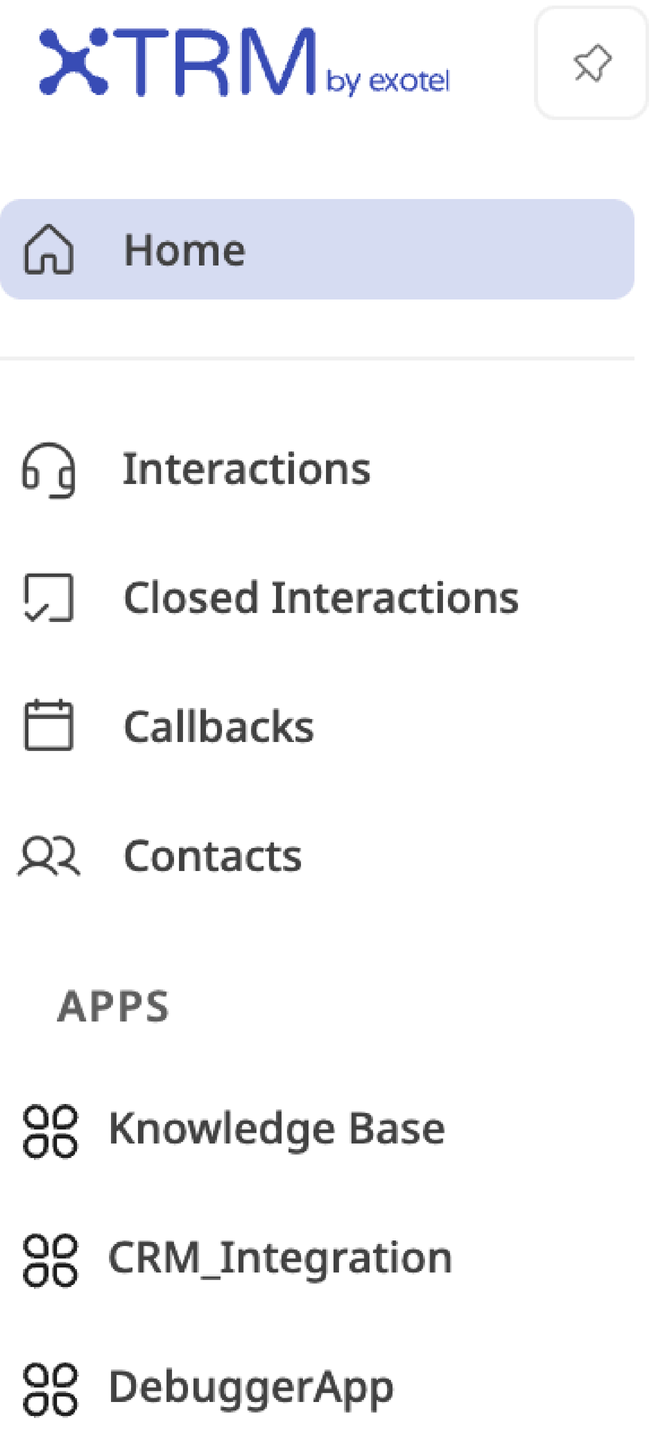
The agent workspace is designed for efficiency, giving you all the tools and information you need in one unified interface. The workspace is made up of the following key components:

- Left navigation panel (collapsible).
- Top navigation bar.
- Main content area (Interactions) with the 4-panel view:
    - Panel 1: Interactions (collapsible).
    - Panel 2: Interaction Details (collapsible).
    - Panel 3: Customer, CRM & Cases (collapsible, except for the CRM section).
    - Panel 4: Configurable App (e.g., Knowmax) (collapsible).
- Right drawer for detailed customer information and past interactions.

3.1 Left Navigation Bar

This is your main navigation menu. It allows you to move between the different sections of the application:

- **Home:** Your performance and metrics dashboard.
- **Interactions:** Your active workbench/workspace for handling all live interactions (voice, chat etc).
- **Interactions History:** A log of all your past closed interactions.
- **Callbacks:** Your list of scheduled callbacks.
- **Contacts:** Your customer directory (Customer 360).
- **App Slots:** Configurable full-screen applications.



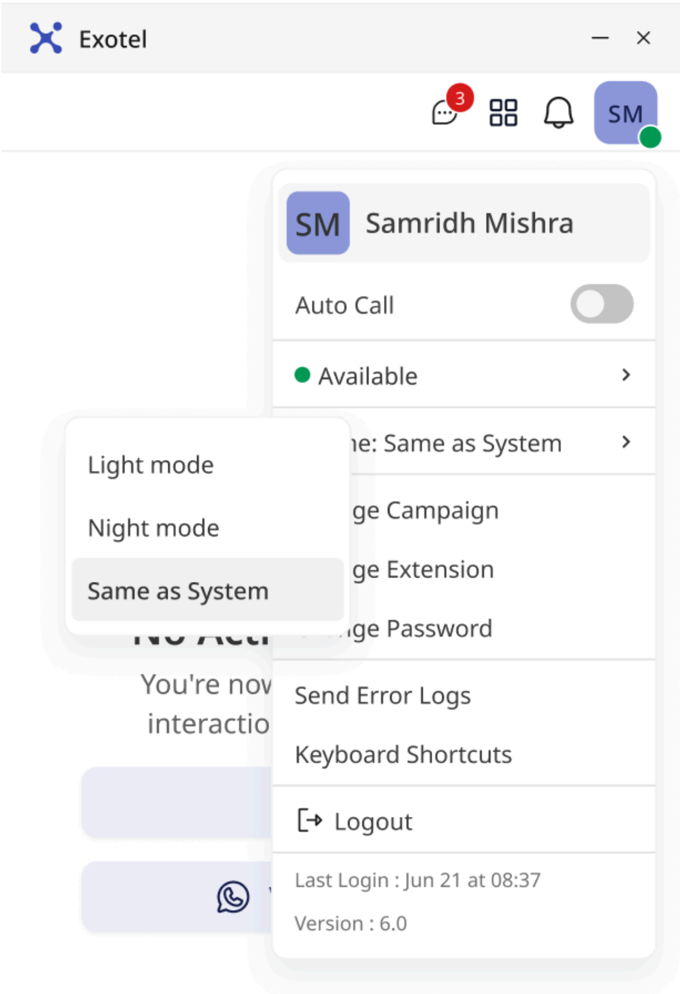
Left Navigation Menu

The navigation bar can be collapsed to an icon-only view, pinned as an expanded, permanently visible menu, or set as an overlay.

3.2 Top Navigation Bar

The top bar provides quick access to essential functions.

- **User Profile:** Where you manage your profile, availability, and theme.

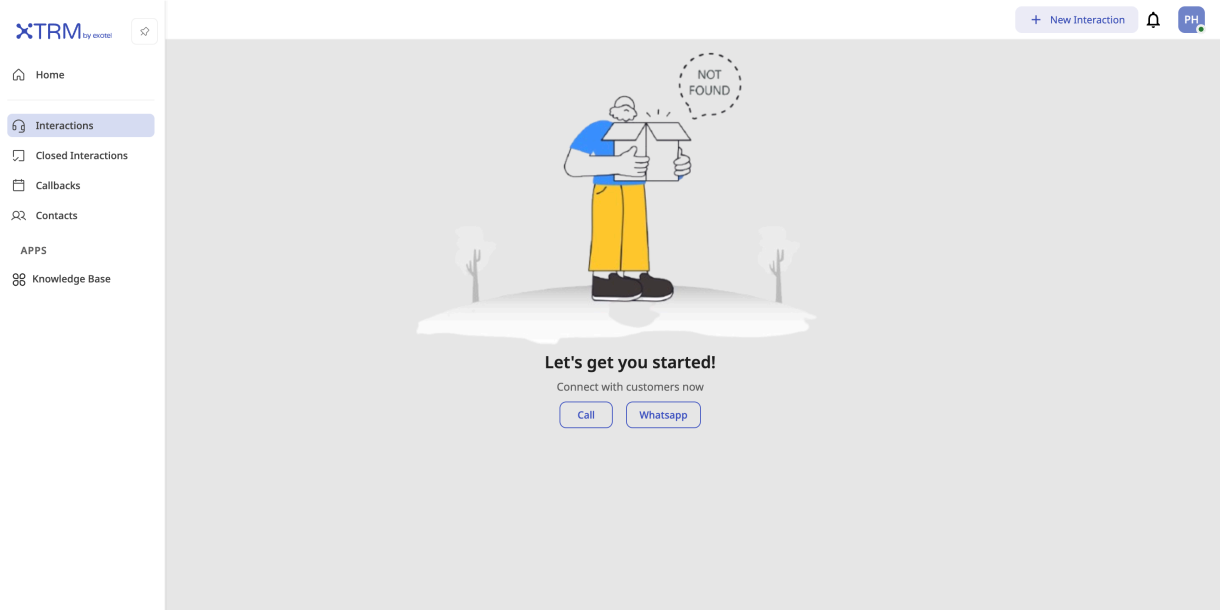


- **Ongoing Interaction Controls:** If you navigate away from the Workbench while on an active call, the call controls (like Mute, Hold, End) will dock here so you can manage the call from anywhere in the app.

3.3 Interaction tab- The 4-Panel View (Agent Workbench)

When you click **Interactions** from the left navigation, you enter your main workspace.

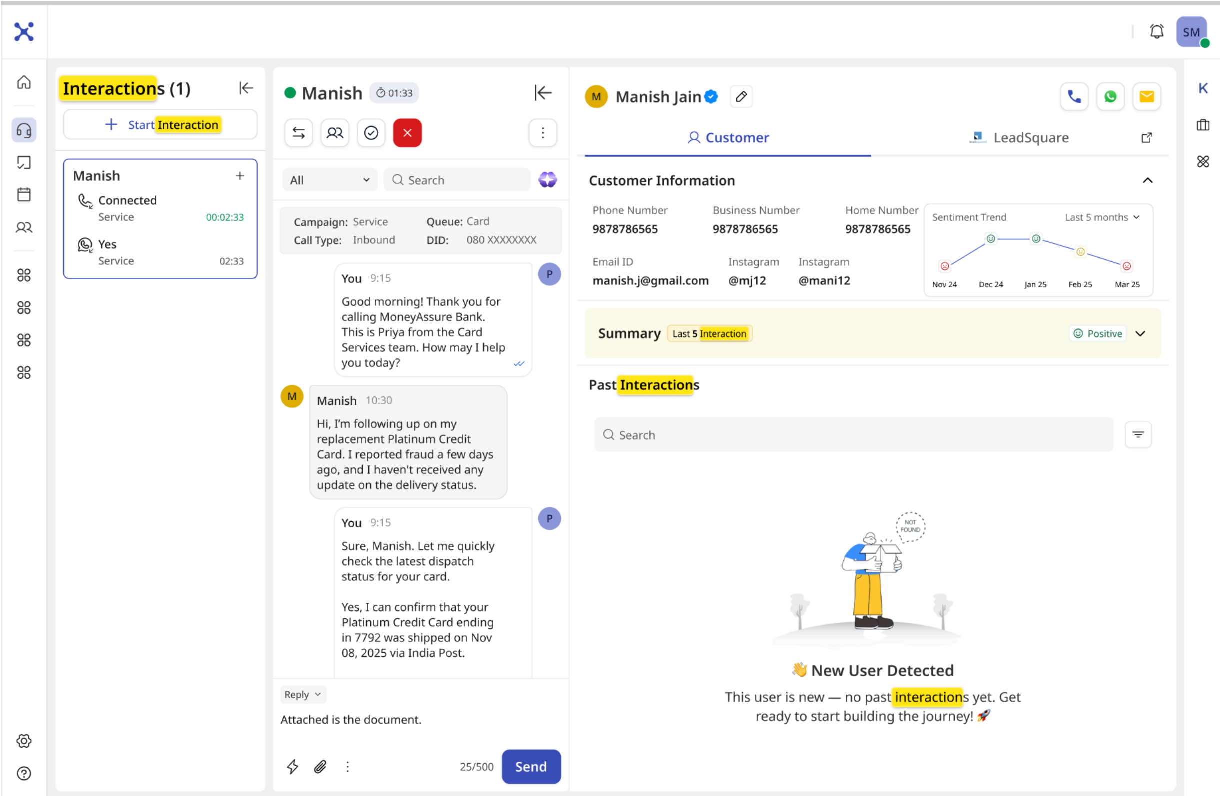
Following in the screenshot when agent is not on any live interaction:



This is a 4-panel layout that gives you a complete view of your customer and tasks.

- **Panel 1: Interactions**
  - This is your interaction queue, located on the far left (first panel).
  - It lists all your ongoing conversations and allows you to switch between them.
  - You can also initiate new interactions from here by clicking the + button.
- **Panel 2: Interaction Details**
  - This panel, located in the middle-left, shows the details of your *currently selected* interaction.
  - It adapts to the interaction channel. For example, it will show the chat transcript and message input for a chat, or a live call transcript for a voice call (if enabled).
- **Panel 3: Customer, CRM & Cases**
  - This middle-right panel provides a 360-degree view of the customer you are interacting with. It contains several tabs:
  - **Customer:** Shows the customer's basic details and interaction history.
  - **CRM:** An integrated view of your company's CRM.
  - **Cases:** A dedicated tab for managing support tickets.
- **Panel 4: Configurable App**
  - This panel, on the far right, is a flexible space that can hold various tools to assist you, such as a Knowledge Base (Knowmax), Agent Assist, or Agent Nudges.

Note: All panels are **collapsible** (except for the CRM section within Panel 3), allowing you to customize your workspace view.

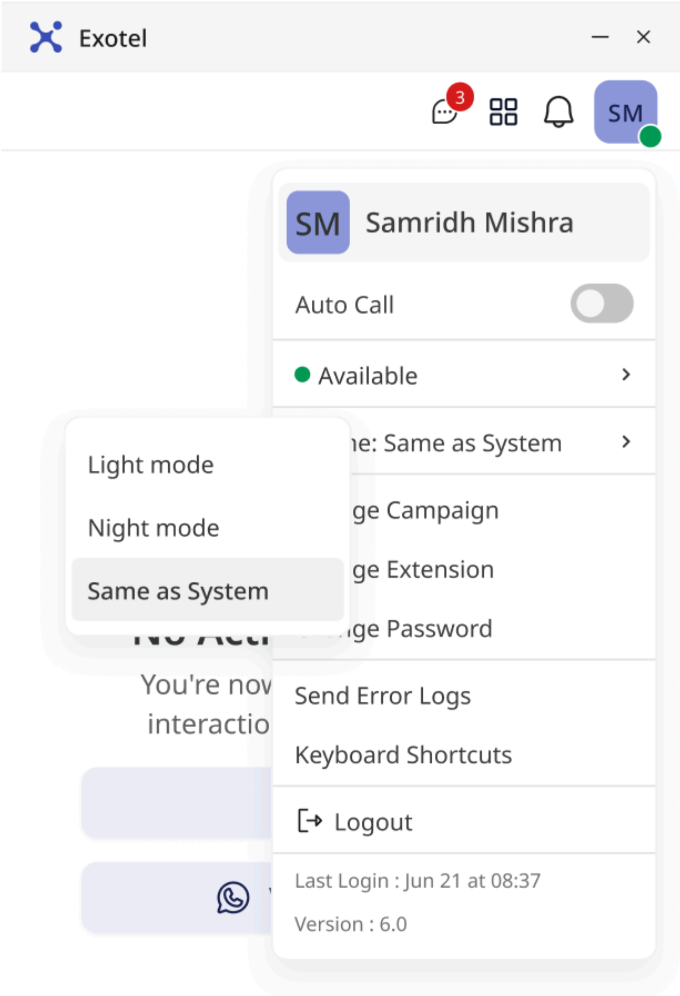


4. Managing Your Status (Availability & Breaks)

Your availability status controls whether you are ready to receive new customer interactions. This is managed from the **Settings** menu.

1. Click **Settings** in the bottom-left navigation bar.
2. Your **User Profile** information will be displayed at the top.
3. Click on your current availability status (e.g., "Available").
4. A list of break reasons will appear. These are set by your administrator but typically include:
    - Lunch
    - Snacks
    - Training
    - Unavailable

5. Select the appropriate break reason when you are stepping away. This pauses new interactions from being assigned to you and informs your supervisor of your status.
6. To start receiving interactions again, return to this menu and set your status back to **"Available"**.
7. In this same section, you can also use the **Auto Call** toggle to enable or disable your participation in auto-dialed calls.



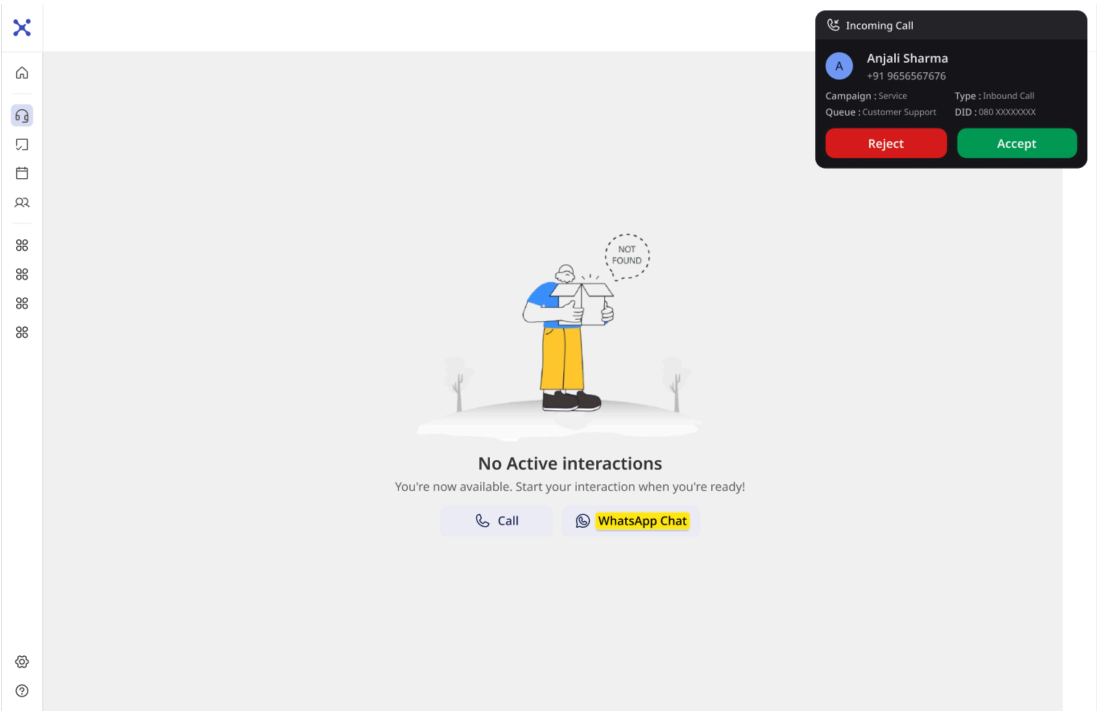
5. Handling Customer Interactions

The workbench is a unified interface that allows you to manage all customer interactions, regardless of the channel, in one place.

5.1 Accepting or Rejecting Interactions

When you are in an "Available" status, the system will offer you new interactions.

1. An **"Accept/Reject" popup** will appear on the top right of your screen (if configured from the admin panel). This notification will display key details about the incoming interaction, such as the customer's ID and the channel.
2. Click **"Accept"** to accept the interaction or **"Reject"** to decline it.

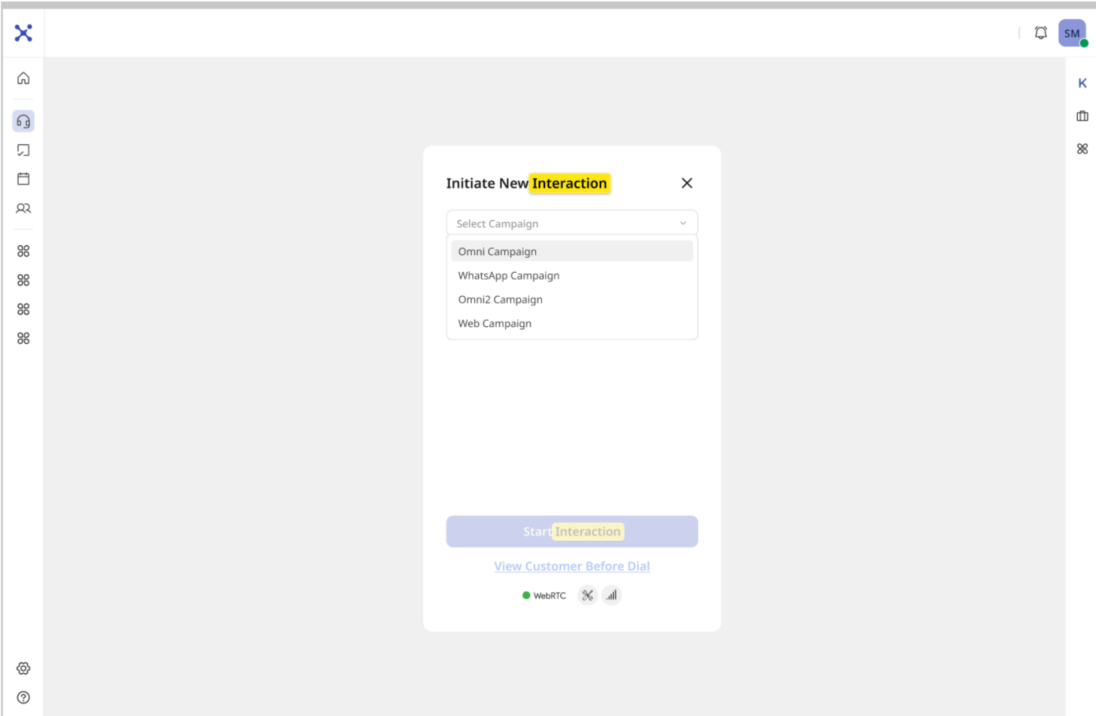


3. When you accept an interaction, it opens in your **Agent Workbench (Interactions tab)**, and the system performs an immediate CTI (Computer Telephony Integration) action:
   - **New Customer:** A blank customer profile will appear, allowing you to create a new contact.
   - **Multiple Customers:** If the phone number matches multiple customers, a list may appear asking you to select the correct one.

5.2 Initiating a New Interaction (Outbound)

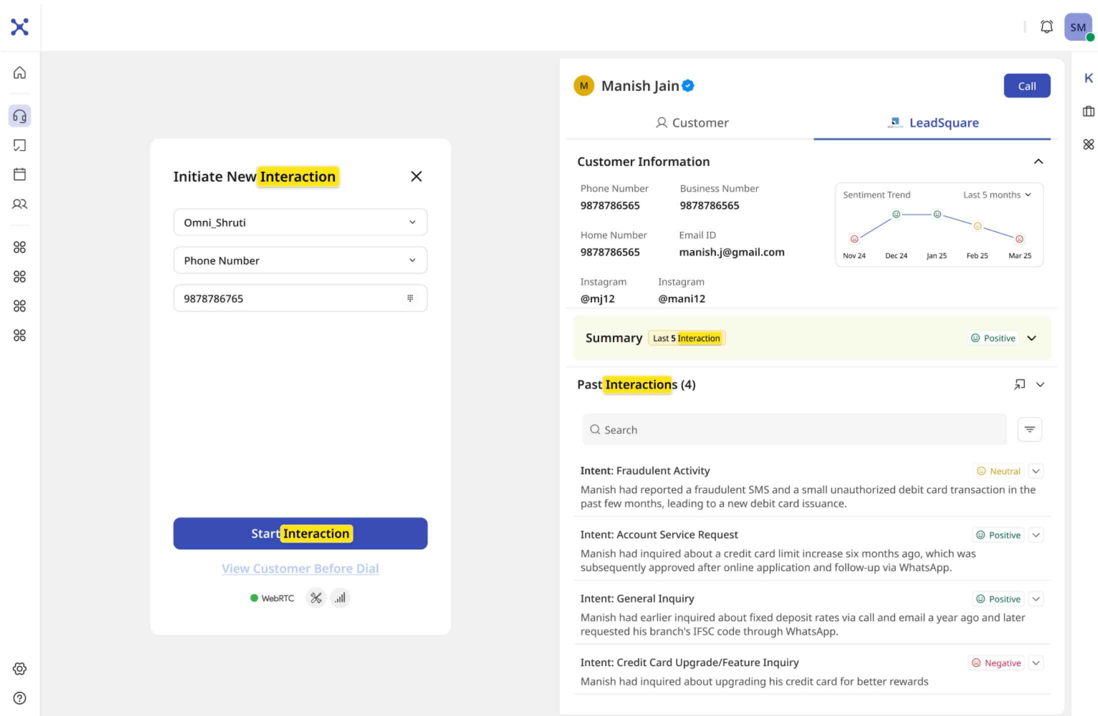
You can start a new outbound interaction (like a voice call or WhatsApp message) at any time from the workbench.

1. Click the **"New Interaction"** button, located in Panel 1 of the workbench.
2. A modal will appear, prompting you to select the **Channel** (e.g., Voice, WhatsApp).



3. Next, you must look up the customer:

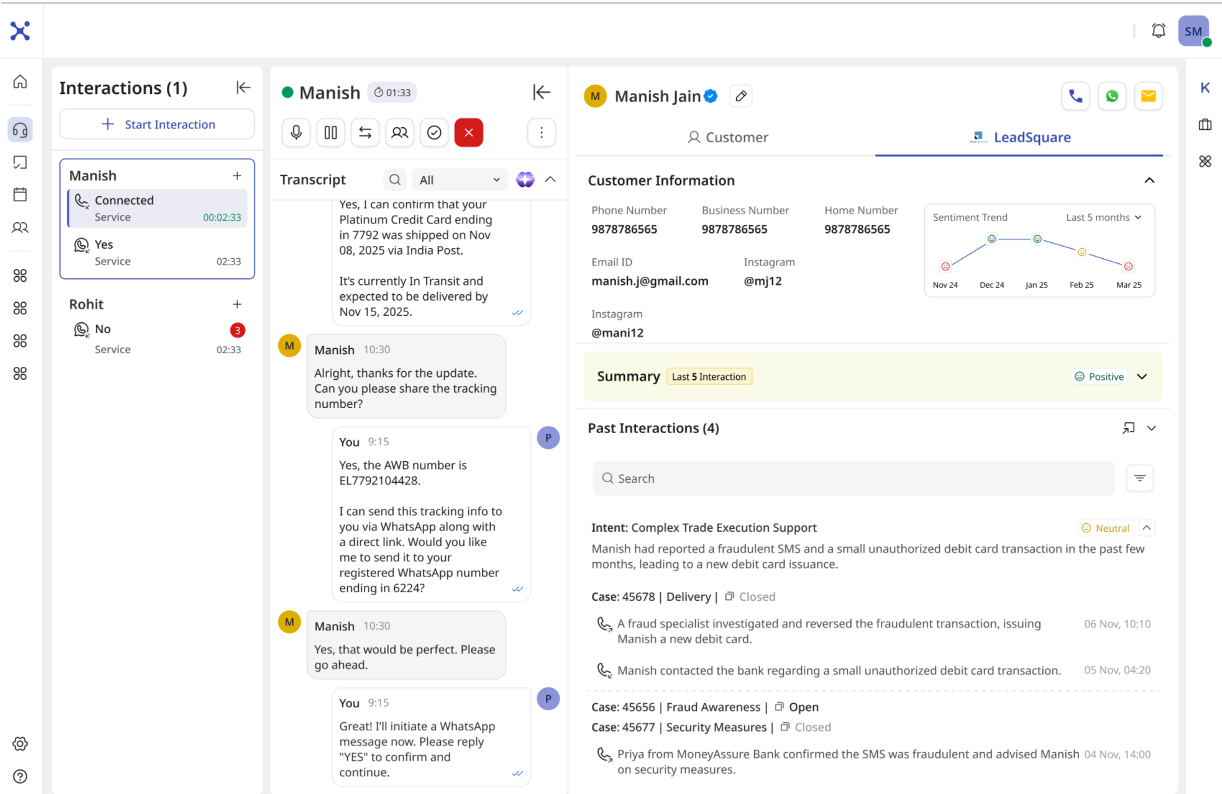
4. Click **"Start"** to begin the interaction.



5.3 Handling Multiple Interactions (Omnichannel)

The omni-channel workbench allows you to handle interactions on different channels at the same time. For example, you can start a WhatsApp chat while on an active voice call.

1. While you are engaged in a voice conversation with a customer, the interaction will be active in your workbench.
2. To add a new channel, click the **+ button** in the interaction bubble (in Panel 1).
3. Select the new channel, for example, **"WhatsApp"**.
4. A new window will appear to start the WhatsApp chat. You can select a template or type a message.
5. Click **"Start"** to begin the chat.
6. You can now switch between the active voice call and the active WhatsApp chat by clicking them in Panel 1. The details for the selected interaction will appear in Panel 2.

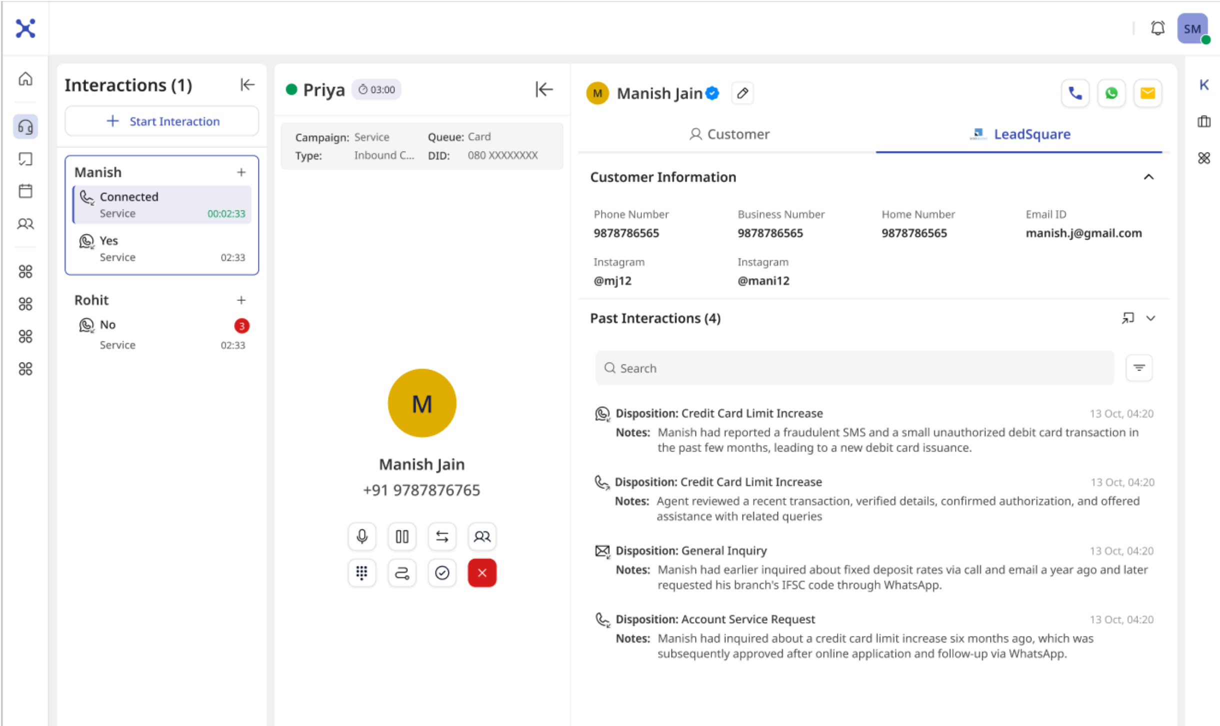


5.4 Handling Voice Interactions

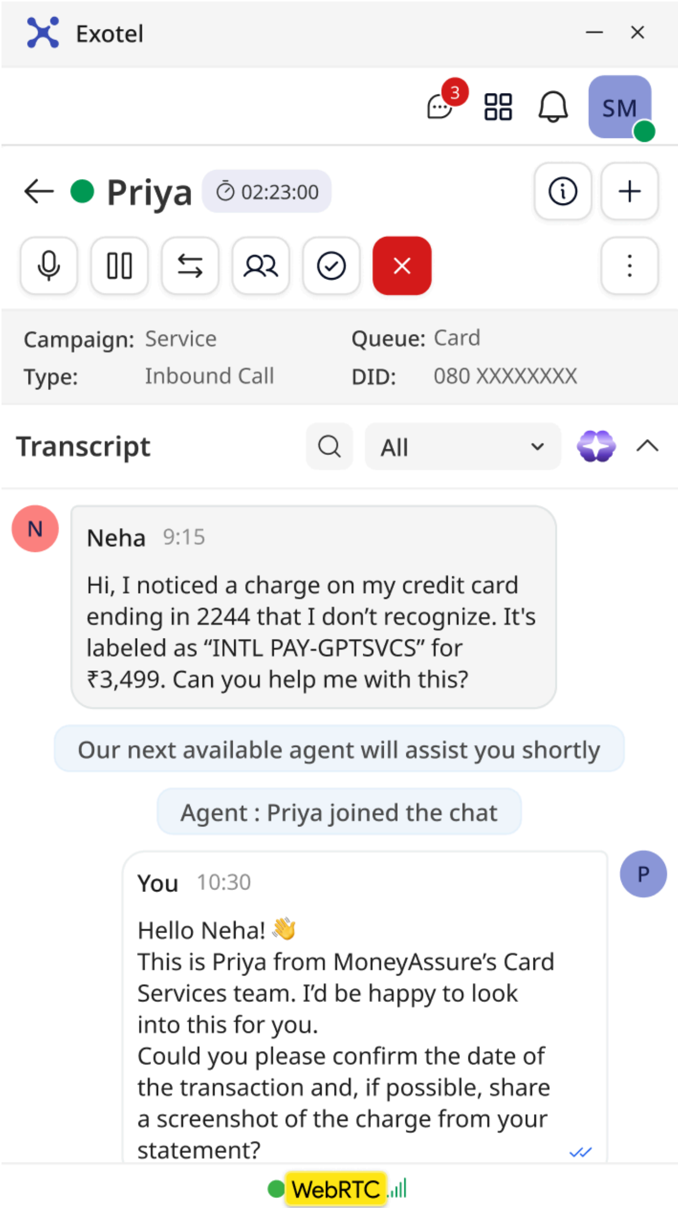
When you accept or start a voice call, the interaction becomes active in your workbench (interactions tab)

- **Panel 2 (Interaction Details):** This panel will display the live call transcript, if this feature is enabled for your account.

- **Panel 3 (Customer 360):** The customer's profile, including past interactions and case history, will automatically load here.



- **Call Controls:** Standard call controls (Mute, Hold, Transfer, End Call) will be available. If you navigate away from the workbench, these controls will dock in the top navigation bar.



5.4.1 Call Controls (Hold, Mute)

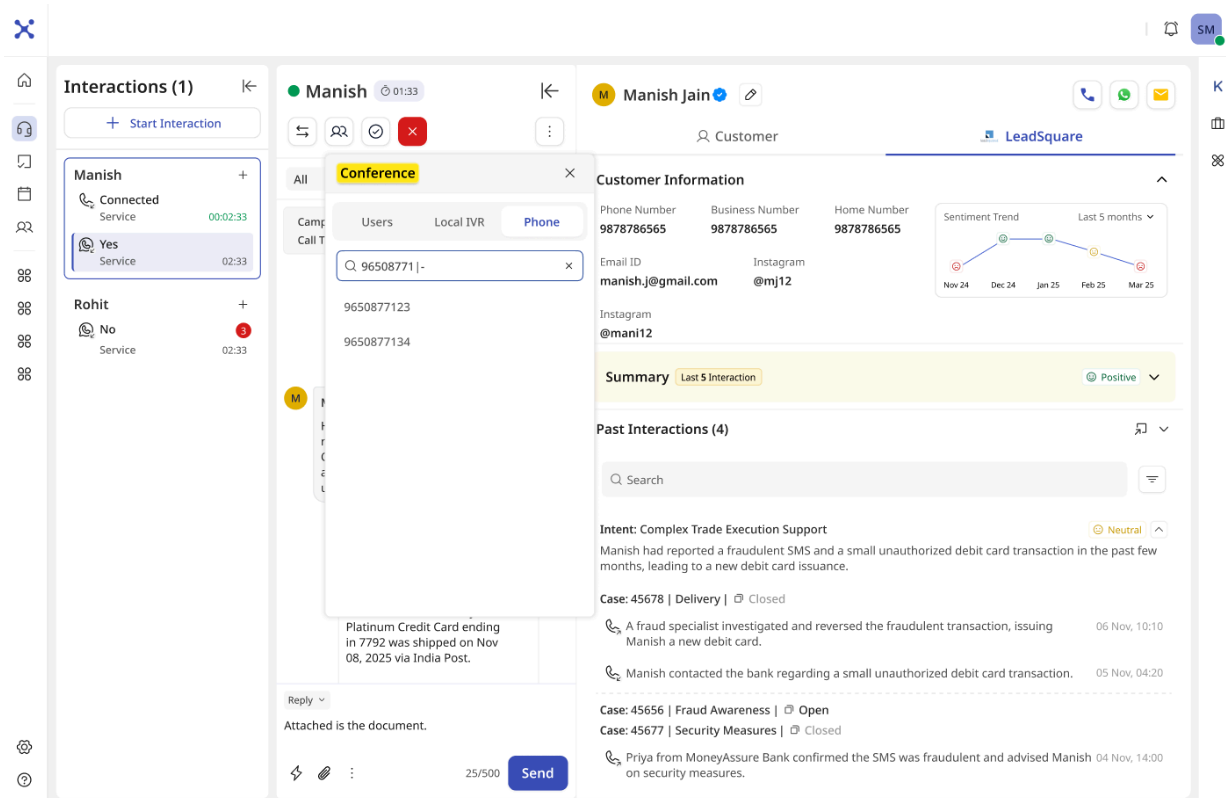
- **Call Hold/Talk:** While on a live call, you can put the customer on hold by clicking the **Hold** button. The customer will hear music, and you can resume the call by clicking the **Talk** (or Unhold) button.
- **Call Mute:** You can mute yourself by clicking the **Mute** button. The customer will not be able to hear you, but you can still hear them. Click the button again to unmute.

5.4.2 Conferencing a Call

The conference feature allows you to add another person to the call.

1. While on the call, click the **Conference** button.
2. You can choose to conference with:

- **A User or Supervisor:** Select from the list of available, logged-in users. You can speak to the new user before unholding the customer to create a three-way conference.
- **An External Number:** Enter an external phone number, click "Confer To Phone," and then unhold the customer once the third party is connected.
- **An IVR:** Select a local IVR from the drop-down menu to conference both yourself and the customer into the IVR, often used for payments or surveys.

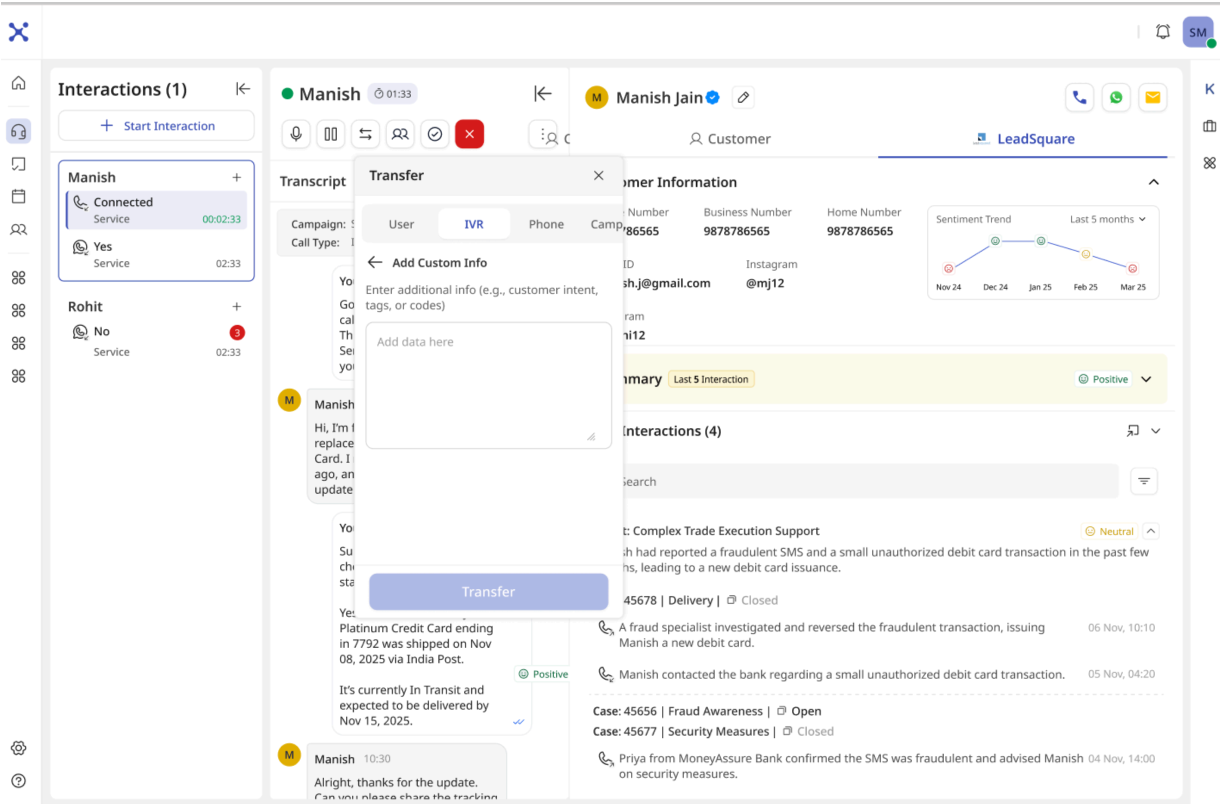


5.4.3 Transferring a Call

You can transfer a call to another party.

1. While on the call, click the **Transfer** button.
2. You can perform a **Warm Transfer** (where you speak to the new agent first) or a Cold Transfer.
3. You can transfer to:
   - **User or Supervisor:** Select a logged-in user from the list. You can speak to them, then click "End Call" to leave, leaving the customer connected to the new user.

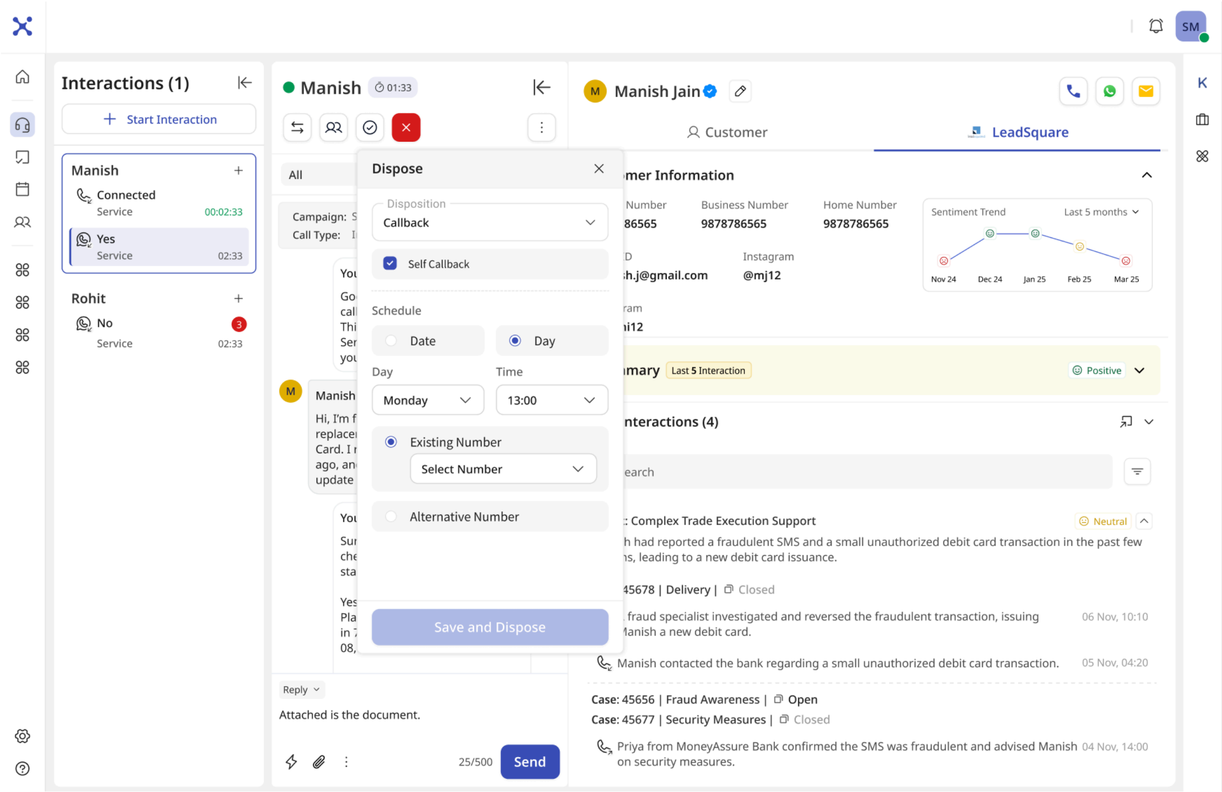
- **External Number:** Similar to a conference, you can dial an external number and then transfer ownership of the call.
- **IVR:** Select an IVR from the drop-down to transfer the customer directly into it.
- **Campaign or Queue:** Select a different campaign or queue from the list to re-route the call to any available agent in that group.



5.4.4 Scheduling a Callback

If the call requires a follow-up, you can schedule a callback during the disposition process.

1. After the call, in the disposition window, select the "Callback" disposition.
2. The callback scheduling options will appear.
3. Check the **"Self Callback"** box if you want the callback to be routed only to you.
4. Choose the time:
   - **Specify date:** Select a specific date and time from the calendar.
   - **Specify time:** Schedule the callback relative to the current time.
5. Enter the correct phone number ("Existing Numbers" or "Alternate Number").
6. Click **"Save And Dispose"** to schedule the callback.



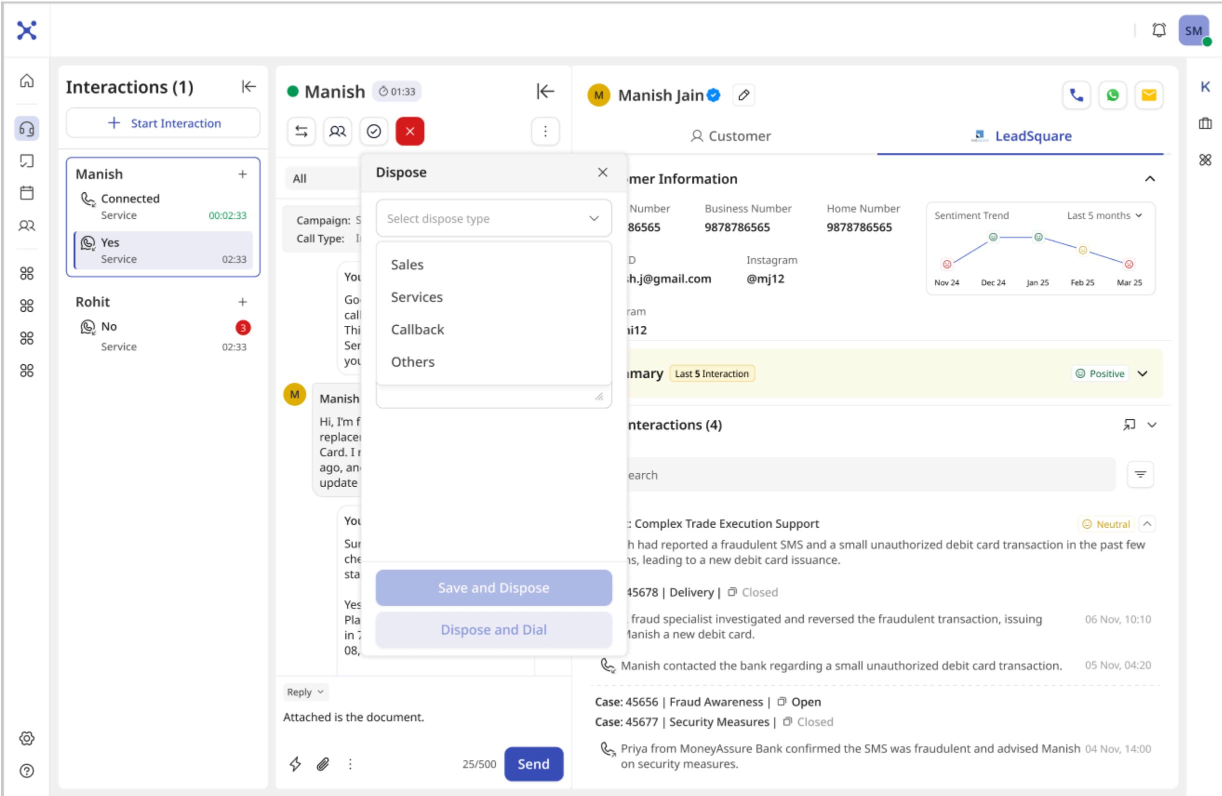
5.4.5 Adding Call Notes

You can add notes during or after a call. Enter your notes in the "Add note" textbox. If you add notes *during* the call, they will be saved automatically when you dispose of the call.

5.4.6 Call Disposition

After completing a call, you must apply a disposition.

1. The disposition menu will appear.
2. Select the **Disposition Class** (the main category) from the first drop-down.
3. Select the **Disposition Code** (the specific outcome) from the second drop-down.
4. If the customer's number should be blacklisted, select the "Blacklist" disposition if configured.
5. Click **"Save and Dispose"** to finish the interaction.



5.5 Handling Chat & Social Media Interactions

When you accept a non-voice interaction, it becomes active in your workbench (interactions tab)

5.5.1 Responding to Chat (WhatsApp, WebChat)

- **Panel 2 (Interaction Details):** This is your main chat window. It will display the conversation transcript, the message input box, and options to send attachments.
- **Initiating with Templates (WhatsApp):** When starting a new outbound WhatsApp chat, you will be shown a list of WABA (WhatsApp Business Account) templates to choose from. You must select a template before clicking "Next" to start the conversation.
- **Load Previous Chats:** If the customer has chatted before, you can click "Load Previous Chats" in the chat window to see their history.

5.5.2 Responding to Social Media (Facebook, Twitter)

- **Panel 2 (Interaction Details):** This panel will show the customer's post or direct message.

- **Like/Unlike:** You can react to a customer's post/comment. For example, on a Twitter ticket, you can click the **"Like"** button on the post. Clicking it again will "Unlike" the post.
- **Hide/Delete Post (if enabled):** For public posts (like on a Facebook page), you may have the option to **"Hide"** or **"Delete"** a customer's comment if it is inappropriate. This action is maintained in the ticket history.
- **Reply:** You can reply directly to the post or send a Direct Message (DM).

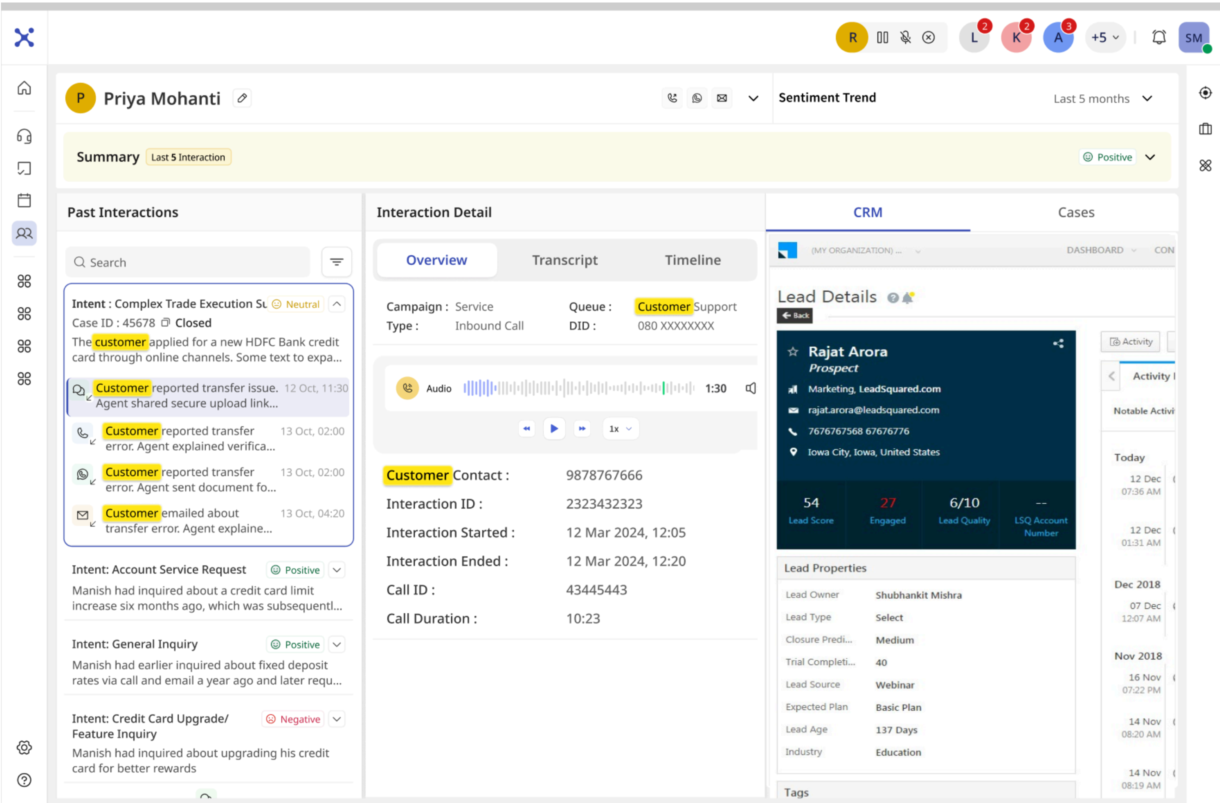
5.5.3 File & Character Limits

Be aware of channel-specific limitations:

- **Twitter:** Replies are limited to 280 characters. You can attach up to 4 images (max 5MB each) or 1 video (max 15MB).
- **Facebook:** Post replies have a character limit of 8,000. You can attach one image (max 4MB).

6. Using the Customer 360 View

The **Customer 360 View** (accessed from the **Contacts** tab) provides a complete overview of a customer, including their interaction history, sentiment, and case details, allowing you to provide personalized and efficient support.



6.1 Accessing Customer 360

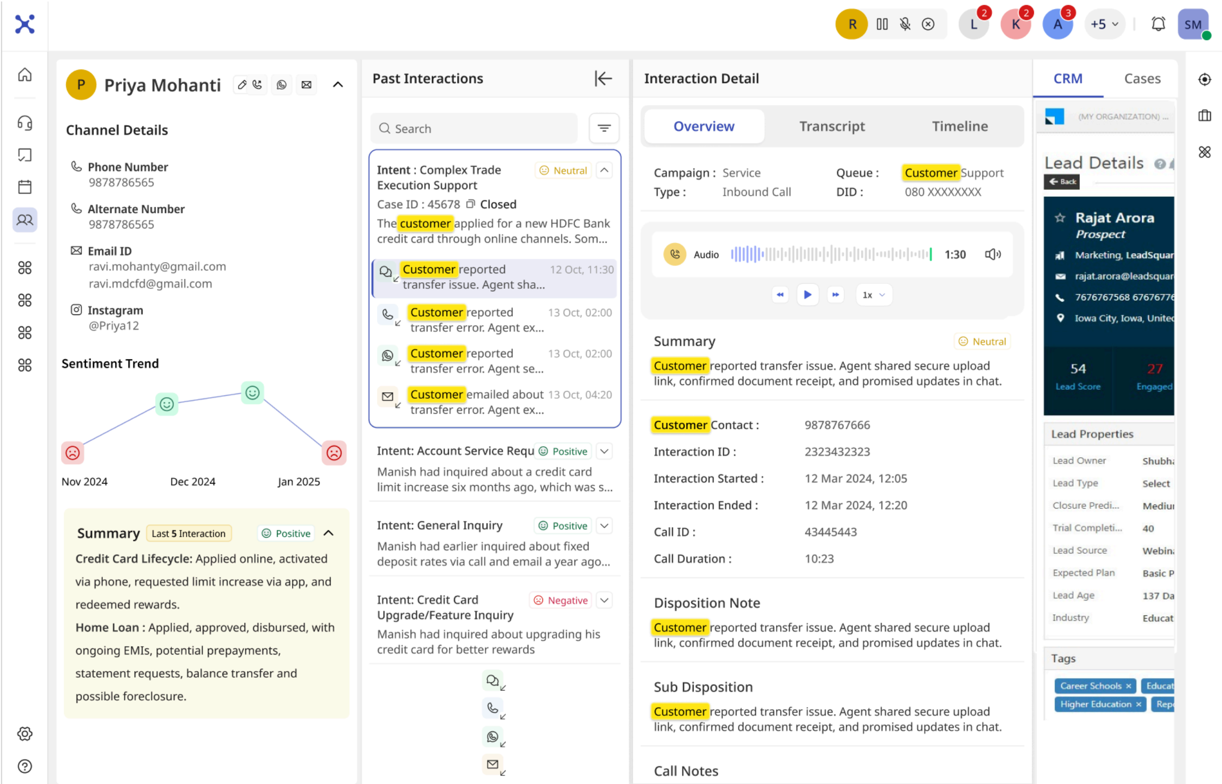
You can access the Customer 360 view in two ways:

1. **From the "Contacts" Tab:**
    - Click **Contacts** on the left navigation bar.
    - This opens the Customer Table. You can search for a customer by name, phone number, email, or other criteria.
    - Clicking on any customer in the table will open their detailed Customer 360 profile in a **Right Drawer** that slides into view.
2. **During an Interaction:**
    - When you accept an interaction, the system automatically identifies the customer and opens their Customer 360 profile in **Panel 3** of the workbench.

6.2 Understanding the Customer 360 Profile

The Customer 360 profile is organized into tabs:

- **Overview:** Shows the customer's name and channel details at the top.
- **Past Interactions Tab:** A list of all previous interactions with the customer.
- **Details Tab:** This tab shows in-depth information about an interaction selected from the "Past Interactions" list, including an overview, the full transcript, and an interaction timeline.
- **Cases Tab:** A list of all support cases (past and present) associated with this customer, fetched from the CRM.
- **CRM Tab:** An embedded Iframe view of the customer's full profile from the integrated CRM.



6.3 AI-Powered Insights: Sentiment & Summaries

The Customer 360 view is enhanced with AI to help you quickly understand the customer's journey:

- **Sentiment Trend:** A graph showing the customer's sentiment (Positive, Negative, Neutral) over the last 6 months.
- **Overall Sentiment:** A single indicator showing the customer's current overall sentiment.
- **AI-Generated Summaries:** (If enabled) Instead of reading long transcripts of past interactions, the AI provides a short, 3-5 line summary for each one. This summary includes:
    - **Intent:** The reason the customer contacted (e.g., "Margin Call Assistance").
    - **Summary:** A brief overview of what was discussed and resolved.
    - **Sentiment:** The detected sentiment of that specific interaction.
    - **Case Number:** Any case ID linked to the interaction.

6.4 Viewing Past Interactions

When you are in the **"Past Interactions"** tab of the Customer 360 profile, you can review all previous customer touchpoints.

- **For a New Customer:** If there is no history, a message will be displayed: "New User Detected - no past interactions yet. Get ready to start building the journey!".
- **For an Existing Customer:** A comprehensive list of past interactions is shown. The level of detail depends on your system's configuration:
  - **AI Enabled, Case Management Enabled:** Interactions are grouped by "Intent" and include the AI summary, sentiment, Case Number, campaign, queue, disposition, and call notes.
  - **AI Disabled, Case Management Disabled:** Interactions are listed chronologically and include only the campaign, queue, disposition, and call notes.

To see more detail, click on any single past interaction. This will populate the **"Details"** tab with the full transcript and interaction timeline.

7. Reviewing Your History

The workspace provides two dedicated sections on the left navigation bar for reviewing your past work: **Interactions History** and **Callbacks**.

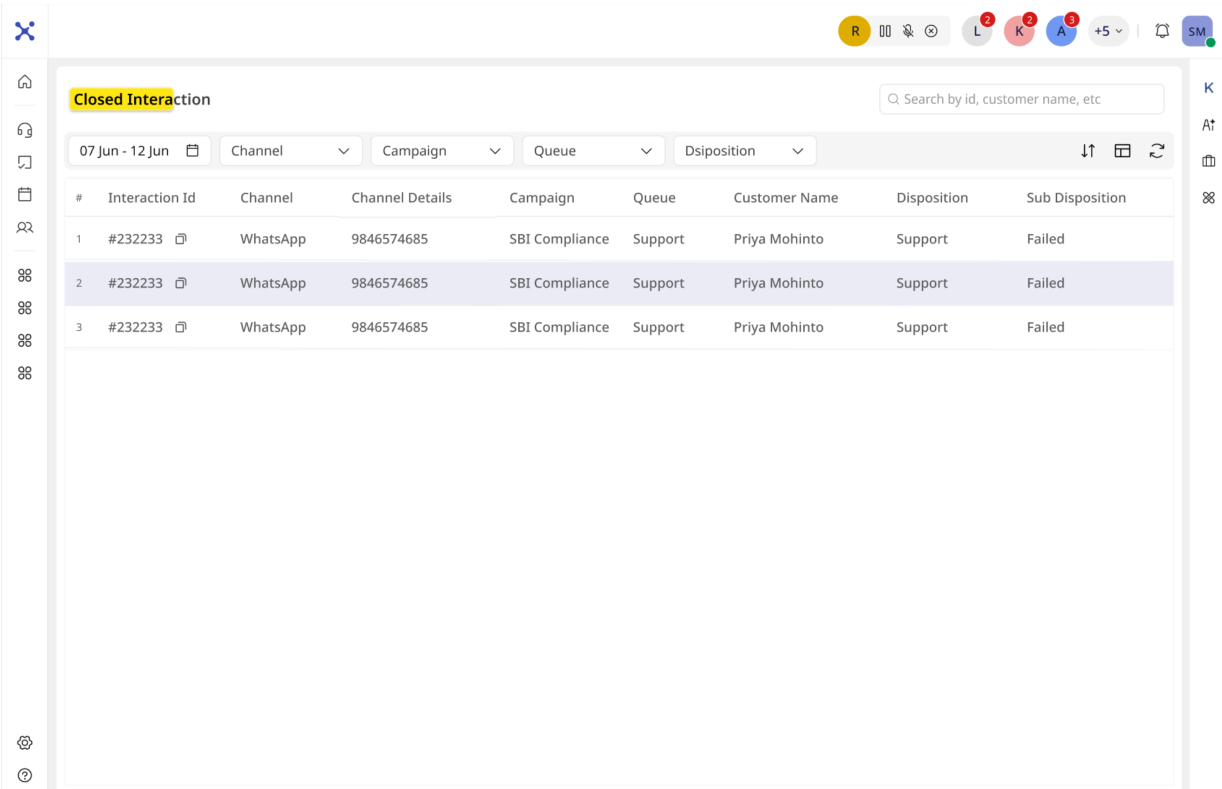
7.1 Interaction History (Closed Interactions)

This screen displays a comprehensive and searchable log of all your completed (closed) interactions from all channels.

- **View:** The history is presented in a paginated table view.
- **Columns:** By default, you will see key metadata for each interaction, including:
  - Channel
  - Campaign
  - Queue
  - Agent
  - Disposition
  - Date/Time
- **Customize View:** You can rearrange the columns using the Column Arrangement feature.

Features:

- **Search:** A search bar at the top allows you to quickly find interactions.
- **Quick Filters:** Located above the table, these dropdowns let you instantly filter the list by:
    - Date Range
    - Campaign
    - Queue
    - Disposition
    - Channel
- **Download Status:** You can export your interaction data. This icon will display the status of your data export.
- **Clear Filters / Undo:** An option is available to reset all applied filters and return to the default view.



7.1.1 Listening to Recordings

For voice interactions, you can listen to the call recording.

1. In the history list, find the call you want to review.
2. Click the **Listen** icon (play button).
3. A pop-up player will appear, allowing you to listen to the call.
4. From this pop-up, you can also **download** the voicelog (recording file) to your computer.

7.1.2 Click-to-Dial from History

You can initiate a new call directly from the call history.

1. Find the call record in the list.
2. Click the **Click-to-Dial** icon next to the customer's number.
3. The system will initiate a new outbound call to that number.

7.2 Managing Callbacks

This screen shows a list of all callbacks that you have scheduled for yourself.

- **View:** Callbacks are listed in a table, showing details like the scheduled date and campaign.
- **Quick Filters:** You can quickly filter your list by **Date** or by **Campaign** using the dropdowns.
- **Filter Chips:** Any filters you apply will appear as removable chips above the list.

7.2.1 Actions on Callbacks (Dial, Reschedule, Delete)

You can manage your scheduled callbacks directly from this screen.

- **Click-to-Dial:** Click the **dial icon** on a callback to initiate the call immediately. The call will open in your workbench, and the callback will be removed from the list.
- **Reschedule:** Click the **reschedule icon** (calendar). A pop-up will appear allowing you to select a new date and time and provide a reason for the change.
- **Delete:** Click the **delete icon** (trash can). You must enter a reason for the deletion before you can confirm.

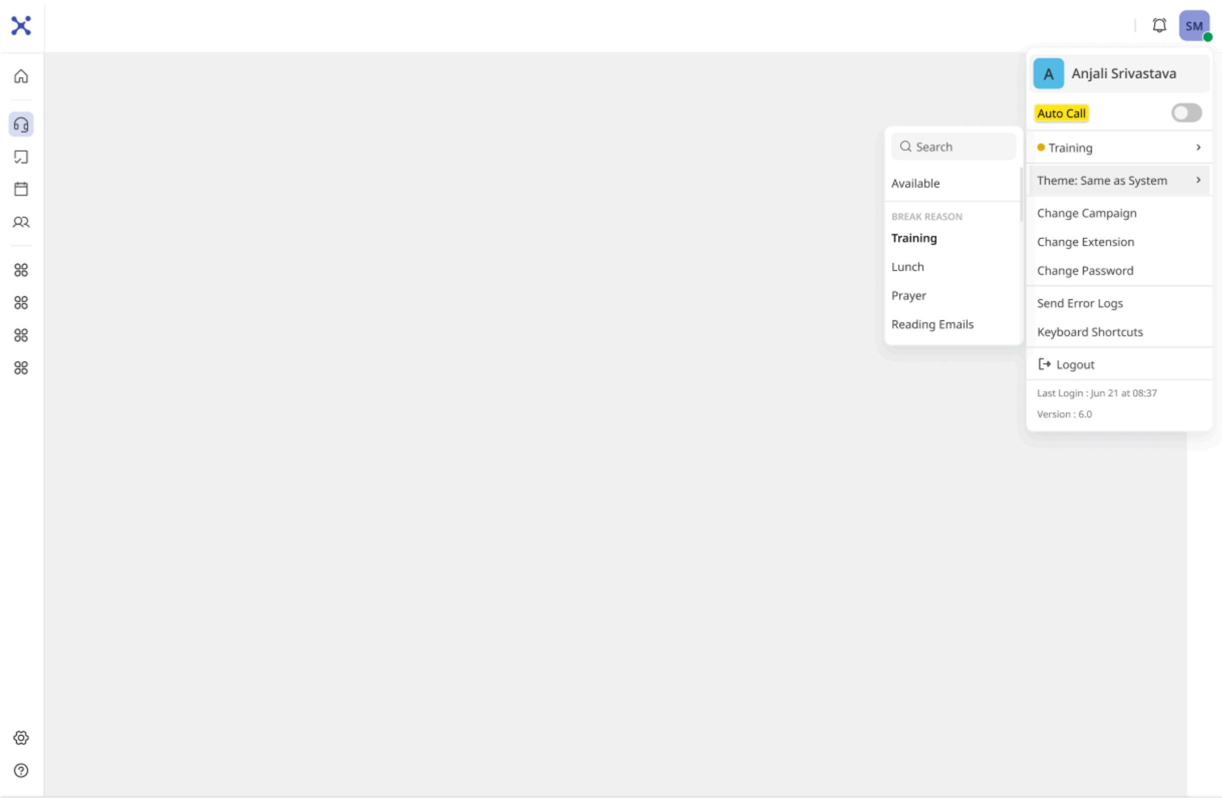
8. Managing Your Profile & Settings

The **Settings** menu, located at the bottom of the left navigation bar, allows you to configure your workspace, manage your status, and access support utilities.

8.1 User Profile Section

This section displays your personal information and manages your work status.

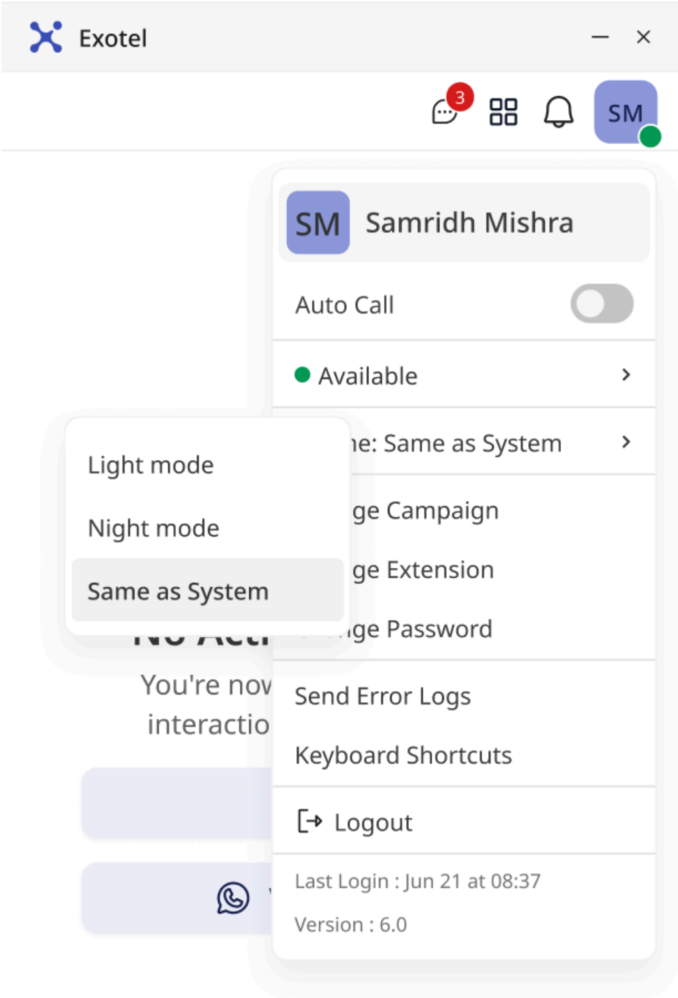
- **Profile:** Shows your agent name and profile initials.
- **Auto Call:** A quick toggle to enable or disable your participation in auto-dialed calls.
- **Availability Status:** Shows your current status (e.g., "Available"). Click this to select a break reason.



8.2 Theme Settings

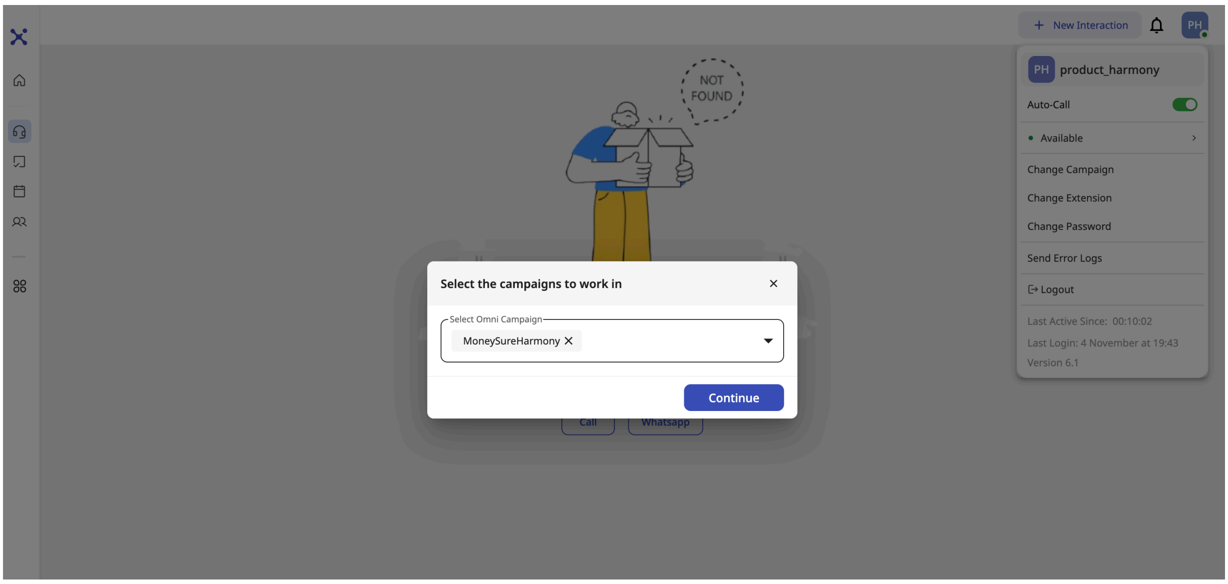
This allows you to change the appearance of your agent workspace:

- Light Mode
- Night Mode
- Same as System

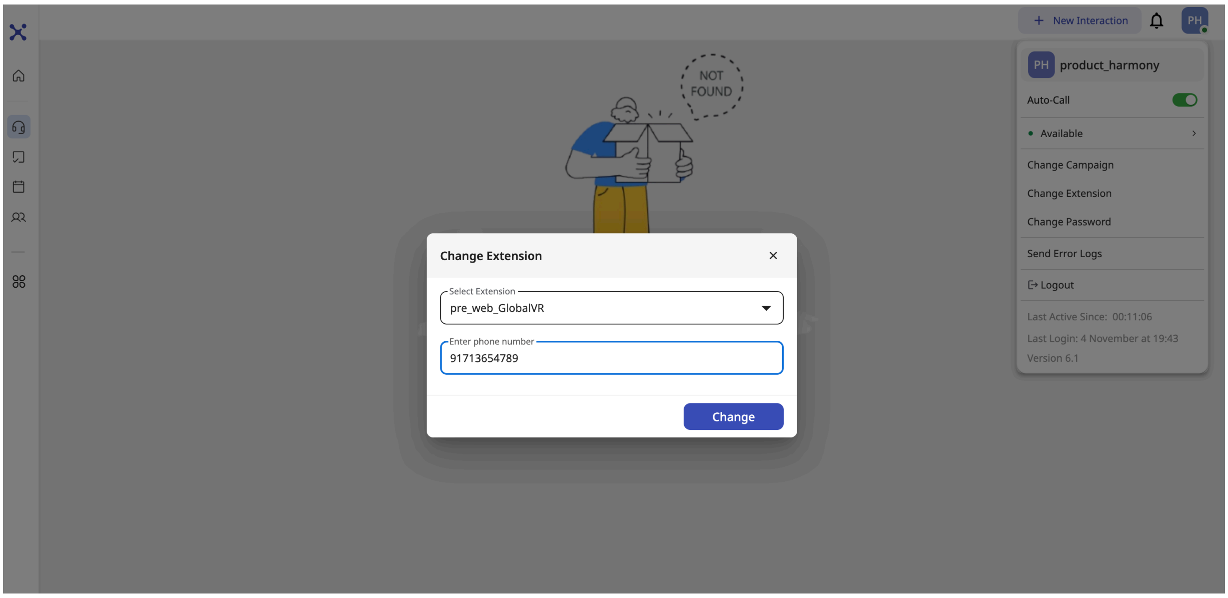


8.3 Session/Account Management Options

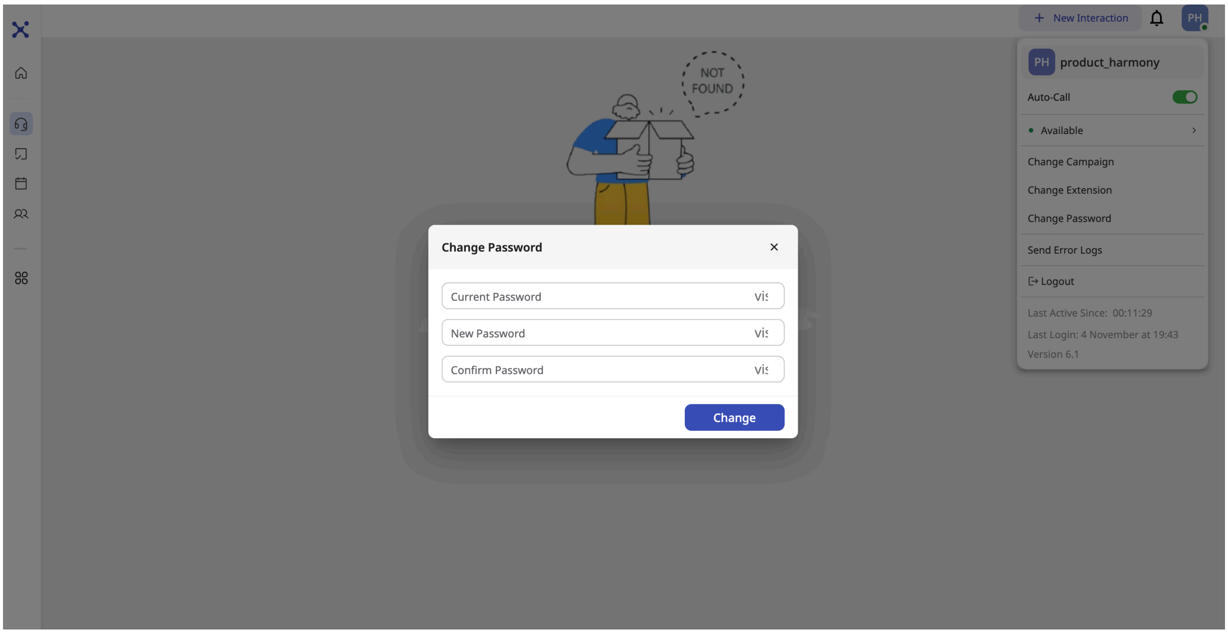
- **Change Campaign:** Allows you to switch your assigned campaign. Click the button and select the new campaign from the pop-up.



- **Change Extension:** Lets you modify your phone extension settings.
- Click on Change extension button and you will get a pop up window as shown in the screenshot.

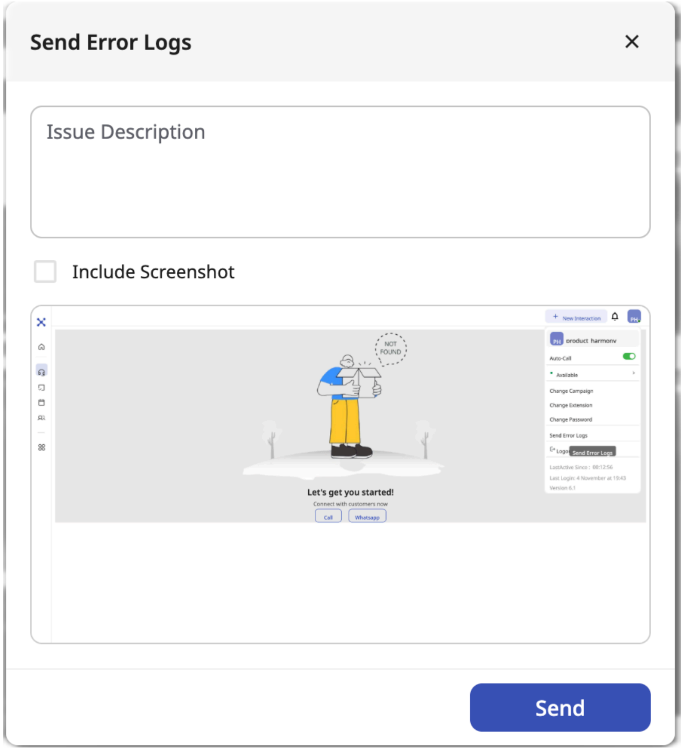


- **Change Password:** Provides a screen to update your login credentials. You will need to enter your current password and then your new password twice.
- Click on Change password button and you will get a pop up window as shown in the screenshot.

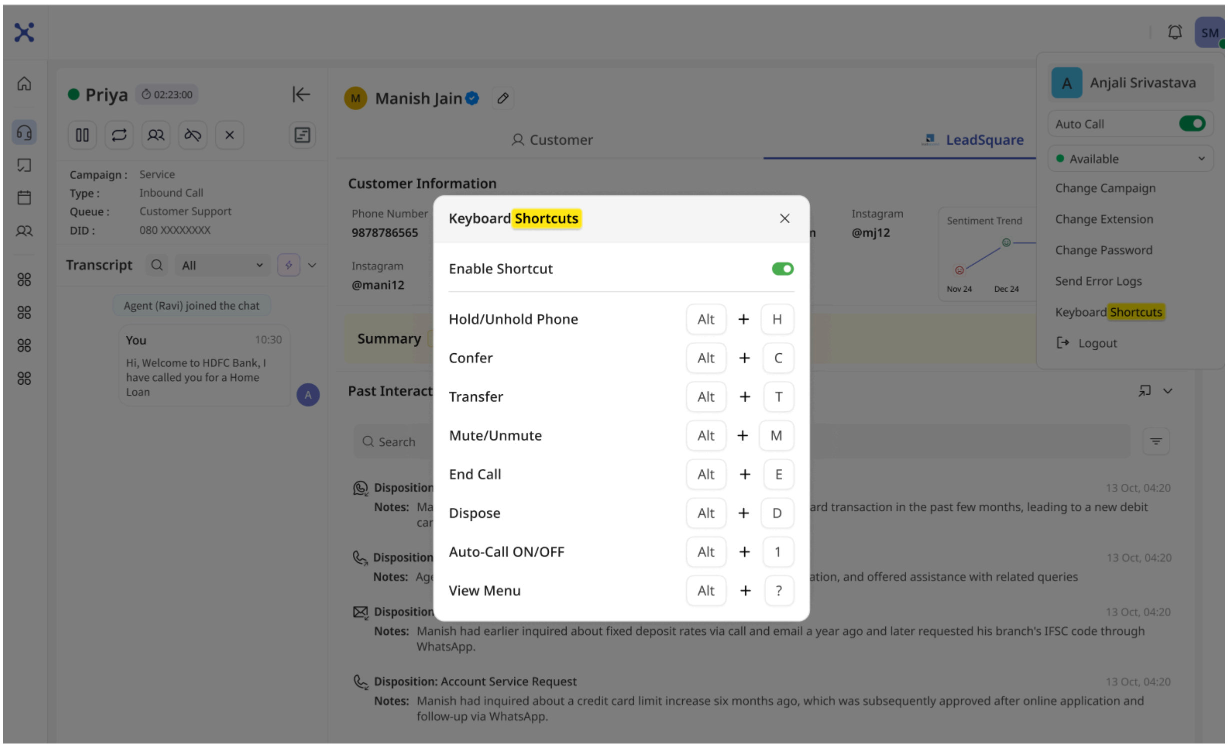


8.4 Support and Utilities

- **Send Error Logs:** If you encounter a bug, click this to send system logs to your technical support team. You should provide a description of the error and can choose to include a screenshot.



- **Keyboard Shortcuts:** View a list of available shortcuts to improve your efficiency. Shortcuts include:
    - **Alt + H:** Hold/Unhold Call
    - **Alt + C:** Confer Call
    - **Alt + T:** Transfer Call
    - **Alt + M:** Mute/Unmute Call
    - **Alt + E:** End Call
    - **Alt + D:** Dispose Call
- **Logout:** Securely log out of the application.



8.5 Managing Your Acronyms

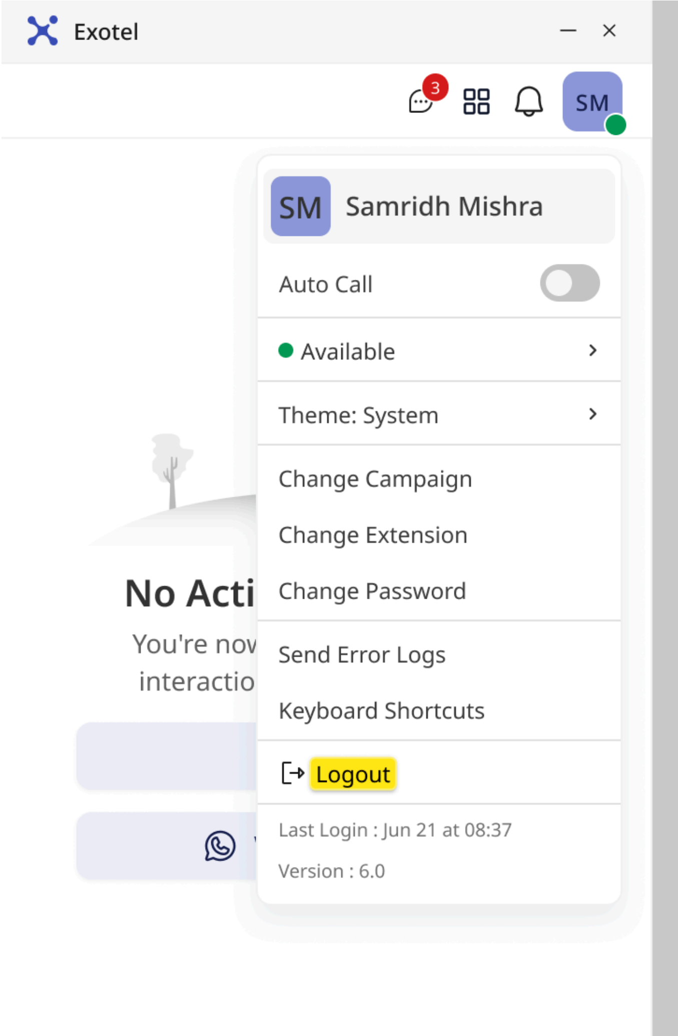
Acronyms are short codes you can create to quickly insert longer, common phrases (e.g., typing "hithere" could insert "Hello, thank you for contacting support. How can I help you today?").

1. From the settings menu, find and open **"Manage Acronyms"**.
2. You will see a list of "Campaign Acronyms" (created by your supervisor) and "Self-Acronyms" (created by you).
3. To create your own acronym:
    - Enter the shortcut in the **"Short Term"** box.
    - Enter the full phrase in the **"Full Term"** box.
    - Click **"Create"**.
4. You can delete any "Self-Acronym" you have created by clicking the delete icon.

9. Log Out

You can securely log out of the agent workspace at any time.

1. Navigate to the User profile menu at the top right
2. Click **"Logout"** to end your session.



Note: The logout option may be disabled if you are on an active call or in an active chat. You must dispose of all active interactions before you can log out.

4.2. Supervisor Guide

Supervisor Manual: Omni Version1

Overview

The Contact Center Supervisor Interface is a unified, web-based dashboard designed to give team leaders and managers complete real-time visibility and control over contact center operations. It serves as a command center for monitoring agent performance, tracking campaign metrics, and intervening in interactions to ensure quality.

Here are the key capabilities of the interface:

- **Real-Time Monitoring:** Supervisors can view live agent statuses (e.g., Ready, Talking, On Break), queue wait times, and active call details through customizable wallboards.
- **Call Intervention Tools:** It provides features to manage live interactions, including **Snoop/Listen** (monitor without detection), **Whisper** (coach the agent without the customer hearing), and **Barge-in** (join the call directly to assist).
- **Agent Management:** Supervisors can force-logout agents, change their work states, or reassign them to different campaigns dynamically based on traffic volume.
- **Analytics & Reporting:** The interface offers access to historical data and real-time KPIs —such as Average Handling Time (AHT) and First Call Resolution (FCR)—allowing for immediate data-driven decision-making.

Supervisor web interface access includes:

1. **Supervisor Login**
2. **Supervisor Console**
3. **Process Settings**
4. **Monitor Tab Introduction**
    - **Monitor Inbound Campaign**
    - **Monitor Outbound Campaign**
    - **Monitor Omni Campaign**

4.2.1. Monitor Omni Campaign

Omni Live Monitoring Configuration Guide (Supervisor)

This dashboard provides supervisors with real-time visibility into agent activities and interaction statuses across Omni channels (selected omni campaign). It is designed to track agent availability, interaction loads, and channel-specific performance.

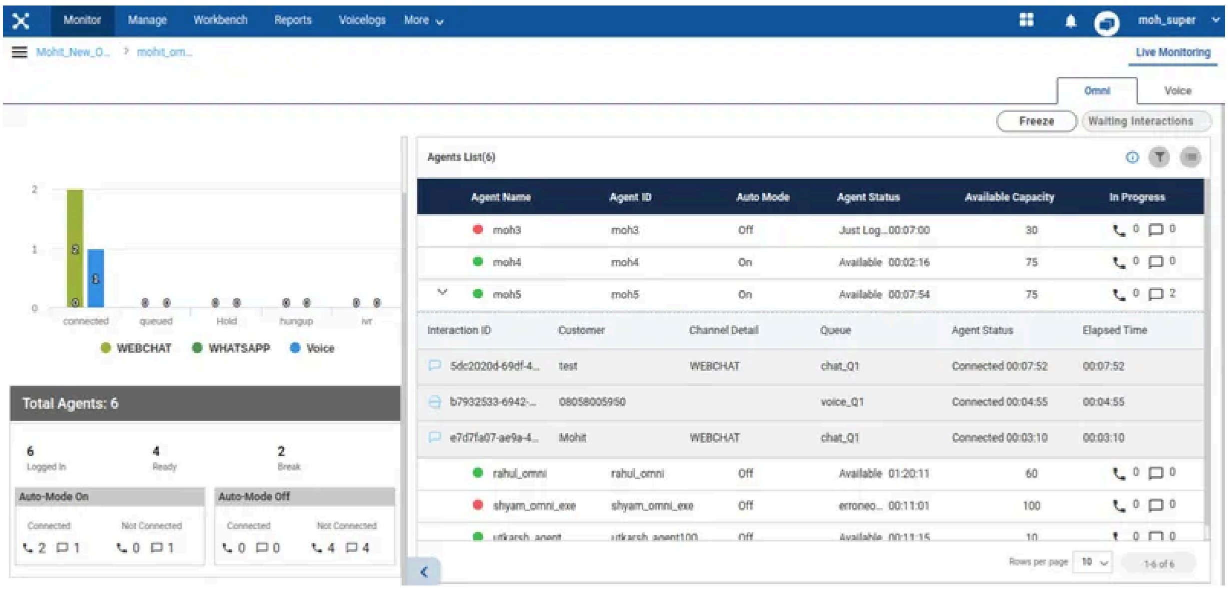
1. Interface Overview

The Live Monitoring dashboard for Omni Campaign is divided into two primary sections:

- **Left Panel (Summary & Stats):** Displays graphical interaction summaries and aggregate agent statistics (Logged In, Ready, Break, and Auto-Mode connectivity).
- **Right Panel (Agent List):** A detailed, tabular view of individual agents, their current status, capacity, and active interactions.

Data Refresh Indicators:

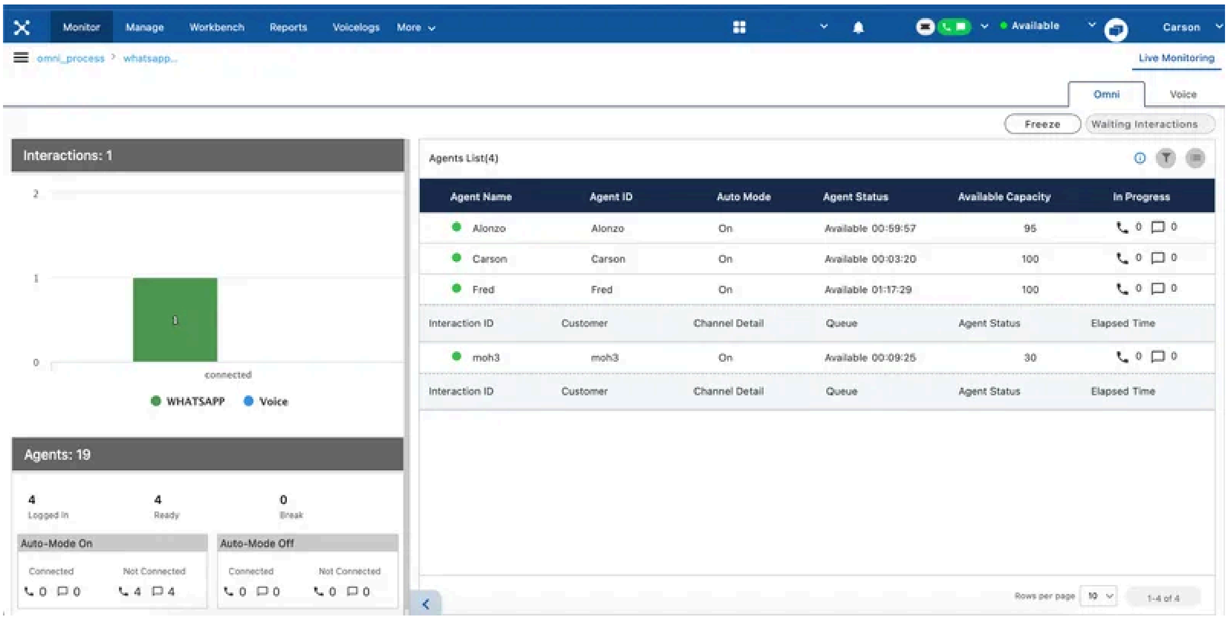
- **Live Monitoring:** The page is labeled "Live Monitoring" in the top right, indicating real-time data updates.
- **Freeze Button:** Allows the supervisor to pause data refreshing to analyze a specific snapshot in time.



2. Interaction Summary (Left Panel)

This section provides a high-level visual representation of current interaction states.

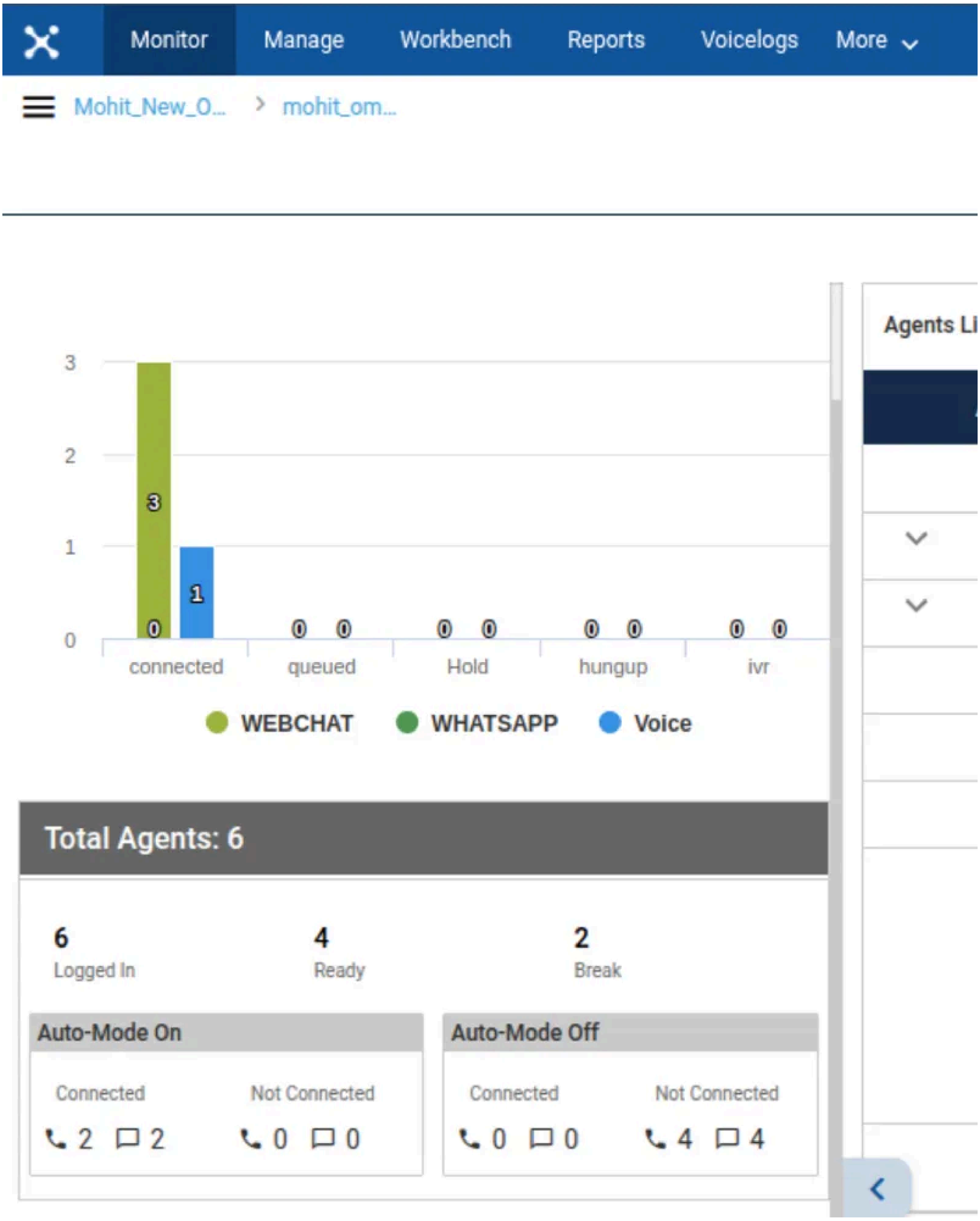
- **Interaction Graph:** A bar chart displays the count of interactions in various states.
    - **States Tracked:** Connected, Queued, Hold, Hungup, and IVR.
    - **Channel Indicators:** Color-coded legends distinguish between channels, such as **WHATSAPP** (Green) and **Voice** (Blue).



3. Agent Aggregate Stats (Left Panel)

Below the interaction graph, the dashboard summarizes agent availability.

- **Total Agents:** Displays the total count of agents currently tracked in the campaign.
- **Status Counters:**
    - **Logged In:** Total agents currently logged into the system.
    - **Ready:** Agents available to accept new interactions.
    - **Break:** Agents currently on break status.
- **Auto-Mode Connectivity:** A matrix separating agents by "Auto-Mode On" vs. "Auto-Mode Off" and their connection status (Connected vs. Not Connected). It also breaks down counts by channel icons (Voice handset vs. Chat bubble).



4. Agent List & Interaction Details (Right Panel)

This table provides granular details for every agent.

A. Agent Columns

Column Name	Definition
Agent Name	Displays the agent's display name along with a status color indicator (e.g., Green for Available/Ready, Red for Busy/Just Logged Out).
Agent ID	The unique system identifier for the agent
Auto Mode	Indicates if the agent is in Auto Call/interaction mode (**On** or **Off**).
Agent Status	Shows the agent's current state and duration
Available Capacity	A numerical score (0-100) indicating the agent's remaining capacity to handle new interactions.
In Progress	A visual count of active interactions per channel type (Voice icon for calls, Chat bubble for messages).

B. Interaction Drill-Down

Clicking the arrow next to an agent expands a detailed view of their active interaction.

Field Name	Definition
Interaction ID	Unique identifier for the specific interaction session
Customer	The customer's identifier, typically a phone number
Channel Detail	Additional info on the specific channel used for communication
Queue	The specific queue handling the interaction
Agent Status	The status of the agent *within* that specific interaction (e.g., Connected etc)
Elapsed Time	Total duration of the current interaction state.

5. Additional Tools

- **Filter & List Options:** Icons in the top right of the Agent List allow supervisors to filter agents or change list view settings.

- **Pagination:** Controls at the bottom right allow navigation through large lists of agents (e.g., "1-4 of 4", "Rows per page 10").
- **Channel Tabs:** Dedicated tabs for **Omni** and **Voice** allow isolation of metrics relevant to specific workflows.