



Enterprise Voice Operations & Analytics (EVOA)

Architecture Overview

Version 1.10.0

August 2026

Publisher: **MeeraApps Technology**

Classification: **Public / Customer Facing**

Copyright and Legal

Copyright © 2026 MeeraApps Technology. All rights reserved.

Enterprise Voice Operations & Analytics (EVOA) and the MeeraApps logo are trademarks of MeeraApps Technology. All other trademarks, service marks, and trade names referenced in this document are the property of their respective owners.

This document is provided "as is" without warranty of any kind. The publisher reserves the right to update the contents of this document without notice. Product features, screenshots, and command examples in this document reflect the release version noted on the cover; subsequent releases may differ.

Neither this document, in whole or in part, nor any of the information it contains may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, without prior written permission of the publisher.

Feedback on this document is welcome: support@meeraapps.com.

Table of Contents

Right-click the TOC below in Word and pick "Update Field" > "Update entire table" to refresh page numbers after any edit.

Copyright and Legal	2
Table of Contents	3
Introduction	4
About This Document	4
Related Documents	4
High-Level Architecture	4
The Four Tiers	4
Tier 1 — External Systems	5
PBX Systems	5
Mitel CX	5
Mitel Emergency Response (SNMP)	5
Tier 2 — Ingestion (Windows Services)	5
Collection Service	5
SNMP Receiver Service (v1.9.0+)	6
CX Real-Time Client	6
Tier 3 — Persistence (PostgreSQL 17)	6
Master Database	6
Per-Tenant Databases	6
Migrations	6
Time Architecture	6
Tier 4 — Presentation (IIS)	7
Web API (/api)	7
Web UI (/ui) — Blazor WebAssembly	7
IIS Topology	7
Desktop Notifier (Windows Tray)	7
Data Flow	8
CDR Ingestion (Real-Time)	8
Emergency Alert (SNMP Trap)	8
Real-Time Supervisor Wallboard (CX)	8
Multi-Tenant Architecture	9
Hierarchy	9
Roles	9
Security Architecture	10
Authentication	10
Authorization	10
Data Isolation	10
Transport	10
Code Signing	10
Licensing	10
Technology Stack	11
Deployment Architecture	12
Single-Server Deployment (Standard)	12
Split-Tier Deployment (Large / High-Availability)	12
Automated Installer	12
Distribution Channel	12
Contact	13
MeeraApps Technology	13

Introduction

About This Document

This document describes the architecture of Enterprise Voice Operations & Analytics (EVOA) v1.10.0. It is intended for pre-sales engineers, deployment architects, and customer IT teams evaluating the platform. Content is technology-focused and does not cover pricing, licensing SKUs, or feature roll-out plans — see the EVOA Product Brochure and Ordering Guide for that.

Related Documents

- EVOA Installation Guide — step-by-step deployment for the engineer running the install
- EVOA System Requirements — hardware, database, network, sizing tiers
- EVOA PBX Compatibility Matrix — driver, port, and CDR-format reference
- EVOA Product Brochure — business-facing overview of the platform

High-Level Architecture

EVOA is a four-tier architecture with a strict separation between data ingestion and presentation. The key principle: the Windows Services and the Web application communicate **ONLY** through the database. There is no direct network connection between them. Each component can be restarted, updated, or scaled independently.

The Four Tiers

Tier	Purpose	Deployment
1. External Systems	PBX systems (14 supported), Mitel CX	Existing customer infrastructure
2. Ingestion	Collection Service, SNMP Receiver, CX SignalR client	Windows Services on the EVOA host
3. Persistence	PostgreSQL 17 (master + per-tenant databases)	Local to the EVOA host (default)
4. Presentation	IIS /api (Web API) and /ui (Blazor) + SignalR + Hangfire	IIS 10+ on the EVOA host

NOTE For deployments serving 500+ agents or 10+ PBXs, the persistence tier can be moved to a separate SQL host and the presentation tier can be scaled horizontally behind a load balancer. See the System Requirements document for sizing thresholds.

Tier 1 — External Systems

PBX Systems

EVOA ingests CDR / SMDR data from 14 PBX systems across four vendors. Each PBX driver is built and tested against real production output — not paper integrations.

Vendor	Systems	Primary Method
Mitel / Unify	MiVoice Business (MiVB), MiVoice Office 400, MX-ONE, OpenScape 4000, OpenScape Voice, OpenScape Business	TCP Push / Pull
Avaya	Aura Communication Manager (ACM), IP Office (IPO)	TCP Push / Pull
Cisco	Unified Communications Manager (CUCM), CME / CUBE	SFTP / TCP Push
Alcatel-Lucent	OmniPCX Enterprise (OXE), OmniPCX Office (OXO), OpenTouch, 4400	TCP Push

Mitel CX

Mitel CX is a first-class integrated system for real-time supervisor wallboards, not merely a CDR source. EVOA connects to CX via SignalR and consumes `queueConversationChanged`, `employeeStateChanged`, and `queueConversationRemoved` events to render live dashboards. Global (Layer 1) presence is surfaced in v1.4.0; per-group (Layer 2) presence requires a REST fallback for CX BI's short-abandon quirks.

Mitel Emergency Response (SNMP)

EVOA listens for SNMP traps from MiVB Emergency Response on UDP 162. The traps carry the emergency-dialled event (911 / 999 / custom pattern), which EVOA correlates against its CESID location map and fans out to phones, desktop notifiers, and email groups. See Emergency Alerts under Tier 2 below.

Tier 2 — Ingestion (Windows Services)

Collection Service

A Windows Service (`CallAccountingCollectionService`) that manages all connections to PBX systems. It maintains persistent TCP connections in Push mode, initiates connections in Pull mode, polls SFTP servers for CDR files, and parses raw records using vendor-specific drivers into a normalised `CallRecord`.

- Auto-reconnect on PBX failure
- Windows Service configuration: delayed-auto, dependency on network, failure actions, failureflag — baked into `Deploy.ps1` (2026-07-25) for auto-recovery after power loss
- Heartbeat via `ServiceStatus` table (30-second interval)
- No HTTP endpoints — database-only communication with the Web API
- HARD RULE: pass-through of PBX timestamps — EVOA never interprets or converts them, and this is enforced via `grep lint` (see the Pass-through Call Timestamps rule)

SNMP Receiver Service (v1.9.0+)

A dedicated Windows Service (CallAccountingSnmpReceiver) that binds to UDP 162 and decodes traps against the MITEL-ERN MIB. The single-authority allow-list is the Trap Sources DB — an empty JSON allow-list means "forward all," avoiding duplication of admin-managed lists.

- Local dedup on (sourceIp, UnackTableIndex) works around a MIVB firmware quirk where SET-destroy ACK is not honoured (per MIB spec)
- Local trap forwarder writes phx.PollMetrics via ON CONFLICT upsert (fixed 2026-06-30)
- Same delayed-auto + dependency + failure-action config as the Collection Service

CX Real-Time Client

An in-process SignalR client (hosted inside the Web API) that subscribes to Mitel CX event streams for real-time dashboards. Uses the ACD Path string for queue identification; bridges to the legacy AcdAgentGroupId domain via GroupNumber / Reporting when persisting CX REST data. Newtonsoft-serialised CX RealtimeStateSnapshots require lowercase-key parsing on the System.Text.Json side.

Tier 3 — Persistence (PostgreSQL 17)

Master Database

The master database holds shared, system-level data that spans tenants — the tenant list, system users, PBX registration, service status, raw SMDR staging, and the reprocessing queue. Never contains customer call records.

Category	Representative Tables
Identity	Users, Roles, Permissions
Tenancy	Tenants, TenantCatalog, Sites
Infrastructure	PbxSystems, ServiceStatus, RawSmdrRecords, ReprocessingQueue
Realtime	RealtimeDriverStatuses, RealtimeStateSnapshots
Phone Display	phx.Phones, phx.PollMetrics, phx.PhonePushHistory
Emergency Alerts	snmp.TrapSources, snmp.CesidMap, snmp.RecipientGroups, snmp.EmergencyEmailDeliveries

Per-Tenant Databases

Every tenant gets its own database, ensuring complete call-data isolation. Even a SuperAdmin cannot see tenant A's call records while connected to tenant B. Tables include CallRecords, Extensions, Departments, CostCenters, TariffRules, Reports, Schedules, and the tenant-scoped ACD tables.

Migrations

EVOA uses hand-rolled EF migrations with PendingModelChangesWarning suppression and idempotent ALTER TABLE with IF NOT EXISTS. No Designer.cs / snapshot required. Every migration is TZ-portable — always uses AT TIME ZONE current_setting('TimeZone'), never a literal (see Time Architecture Split).

Time Architecture

The system splits time authorities: Mitel CX data uses CX GETDATE(); operator stamps use DateTime.Now on the EVOA host. Both are stored as Kind=Unspecified into timestamp-without-tz columns. This preserves the PBX-native timeline for reporting while keeping the operator UI consistent with the browser's local time.

IMPORTANT EVOA never interprets, adjusts, or converts a PBX-supplied timestamp. Any change to that behaviour requires review — the pass-through invariant is enforced by grep lint.

Tier 4 — Presentation (IIS)

Web API (/api)

An ASP.NET Core Web API on .NET 10, hosted under IIS at /api. Serves the Blazor UI, the phones, the TV wallboards, the Desktop Notifier, and any external REST clients.

- JWT Bearer authentication (unique RSA-generated secret per install)
- RBAC with SuperAdmin, Admin, Sales, Viewer + 6 partner-tier roles
- REST endpoints for CRUD across every entity
- Hangfire background jobs for reporting, emergency-email fan-out, and scheduled tasks
- SignalR hubs for real-time updates to browser, phone, TV, and notifier surfaces
- Swagger / OpenAPI documentation exposed at /api/swagger

Web UI (/ui) — Blazor WebAssembly

A Blazor WebAssembly SPA on .NET 10, hosted under IIS at /ui with SPA 404-to-index fallback. Runs entirely in the browser after initial load; communicates with the API over HTTPS using JWT.

- Dashboard with real-time KPI widgets (ApexCharts)
- Call Records grid with server-side filtering, sorting, and export
- Report builder + scheduled reports
- PBX system configuration
- Phone Display admin (Phones, Desktop Stations, Emergency Alerts, Caller Tags)
- Real-Time Wallboards with widget composer
- Dark mode, RTL for Arabic, 8 languages with dynamic switching

IIS Topology

EVOA's IIS split at /api and /ui is a documented topology. External clients must include the /api prefix on REST + SignalR URLs (see the IIS Topology memory for the SPA fallback wiring).

Path	Purpose
/api	Web API (REST + SignalR hubs)
/ui	Blazor WebAssembly SPA (404-to-index fallback)
/api/swagger	OpenAPI documentation
/wwwroot/help-content/	Per-locale help pages served as static assets
/wwwroot/downloads/notifier/	Signed Desktop Notifier installers for auto-update discovery

Desktop Notifier (Windows Tray)

A signed Windows tray application distributed alongside the server release. Connects to the /api SignalR hub, participates in the full-mesh acknowledgement fabric, silently auto-updates by polling /api/v1/desktop-notifier/version every 24 hours.

- Signed with the MeeraApps Certum EV certificate — no SmartScreen warnings
- Per-category custom alert sounds (built-in WAVs + custom upload)
- Full-mesh acknowledgement with phone / browser / TV
- SignalR reconnect: built-in schedule + perpetual 5–60 sec exponential backoff (v1.10.0)
- Lock-screen constraint: Toast + sound + display-wake work; full popup cannot render (Windows Session 0)

Data Flow

CDR Ingestion (Real-Time)

1. PBX sends raw CDR / SMDR over TCP to the Collection Service, or Collection Service polls the PBX / SFTP source.
2. The appropriate vendor driver parses the raw data into a normalised SmdrParseResult.
3. The Collection Service classifies the call (inbound / outbound / internal) and enriches with site + PBX metadata.
4. The parsed CallRecord is written to the tenant's CallRecords table.
5. If raw SMDR storage is enabled, the raw text is also written to RawSmdrRecords (default 365-day retention).

Emergency Alert (SNMP Trap)

1. MiVoice Business fires an SNMP trap on 911 / 999 dial.
2. The SNMP Receiver Service decodes the trap and validates against the Trap Sources allow-list.
3. The service correlates the CESID to a location entry and identifies the matching Recipient Group.
4. The Web API fans out the event to every configured channel — phones, Desktop Notifier, email group.
5. Any surface acknowledges; the ACK propagates through the full-mesh fabric and closes the event everywhere within a second.

Real-Time Supervisor Wallboard (CX)

1. The Web API's CX client subscribes to queueConversationChanged / employeeStateChanged / queueConversationRemoved on the CX SignalR hub.
2. Received events are transformed into RealtimeStateSnapshots and pushed onto the EVOA SignalR hub.
3. The Blazor dashboard's widgets receive the push and update the ApexCharts in real time — no polling loop.
4. Per-widget supervisor scope is narrowed via IUserAcScopeService at every endpoint.

Multi-Tenant Architecture

A single EVOA installation serves multiple tenants, each with complete data isolation. Every tenant has its own database, sites, PBX systems, extensions, departments, cost centers, tariffs, branding, and reporting scope.

Hierarchy

The License Manager v1.4.0 introduced a 3-tier partnership model: Direct Partners (PTR-), Distributors (DIS-), and Sub-Partners (SUB-) with hard role-to-type binding. Distributor and Sub-Partner tiers roll up into consolidated dashboards. Only SuperAdmin can move customers or reparent Sub-Partners.

Tier	Owns Customers?	Buys From
Direct Partner	Yes	MeeraApps
Distributor	No (via SUBs)	MeeraApps
Sub-Partner	Yes (own only)	Its parent Distributor

Roles

Admin	Full access within assigned tenant
Sales	Quotes, orders, partners (no settings / users)
Viewer	Read-only across all data
PartnerAdmin/User	Direct-partner sub-user
DistributorAdmin/User	Roll-up across own + Sub-Partners
SubPartnerAdmin/User	Own scope only

Security Architecture

Authentication

- JWT Bearer tokens; configurable expiry (default 60 min)
- Unique RSA-generated JWT secret per installation
- Brute-force protection on the login page

Authorization

- Role-Based Access Control with tenant-scope enforcement
- SuperAdmin operations (delete / deactivate / move / force-change) hidden from tenant-tier admins
- Supervisor scope narrowing via IUserAcScopeService (intersected as the last narrowing step)

Data Isolation

- Per-tenant databases enforce hard isolation
- Cross-tenant queries are structurally impossible — no shared call-record tables

Transport

- HTTPS (TLS 1.2+) recommended for all web traffic
- Certificates managed via IIS
- SMTP with TLS/SSL for scheduled reports and emergency-alert email fan-out

Code Signing

Every first-party binary shipped to a customer is signed with the MeeraApps Certum EV certificate: EVOA server binaries, License Manager, LM installer, Desktop Notifier installer, and CallAccounting.Licensing shared library. Signing uses jsign + PKCS#11 + RFC3161 timestamping. Windows SmartScreen recognises the signature immediately.

Licensing

- RSA-SHA256 digital signature on license files
- Optional hardware binding (machine-specific validation)
- Offline validation — no internet required at runtime
- 30-day trial + 30-day grace on expiration

Technology Stack

Layer	Technology
Runtime	.NET 10
Web framework	ASP.NET Core 10
Frontend	Blazor WebAssembly (.NET 10)
Database ORM	Entity Framework Core 10
Primary database	PostgreSQL 17
Alternate database	Microsoft SQL Server 2019+
Web server	IIS 10+ (Windows)
Background jobs	Hangfire 1.8
Real-time	SignalR (over WSS)
PDF reports	QuestPDF
Excel reports	EPPlus
Charts	ApexCharts (Blazor-ApexCharts)
SFTP client	SSH.NET
SNMP receiver	SharpSnmpLib (UDP 162 trap listener)
Logging	Serilog (structured)
Code signing	jsgn + PKCS#11 (Certum EV cert) + RFC3161 timestamping
API docs	Swagger / OpenAPI

Deployment Architecture

Single-Server Deployment (Standard)

All components run on one Windows Server. Recommended for most deployments — up to 500 agents, 10 PBXs, and moderate reporting load.

Component	Location
Web API	IIS — /api
Blazor UI	IIS — /ui
Collection Service	Windows Service — CallAccountingCollectionService
SNMP Receiver	Windows Service — CallAccountingSnmpReceiver
PostgreSQL	Local instance on the same host
Data + logs + backups	C:\CallAccountingData\
Help content	C:\inetpub\wwwroot\CallAccounting\ui\help-content\ (IIS live)

Split-Tier Deployment (Large / High-Availability)

For 500+ agents, 10+ PBXs, or when high availability is required, the tiers can be split across separate hosts:

- Presentation tier: IIS on a Windows Server, optionally behind a load balancer
- Persistence tier: PostgreSQL on a separate host (or a managed PG service)
- Ingestion tier: Collection Service + SNMP Receiver on a dedicated Windows host in the same VLAN as the PBXs

TIP The Windows Service configuration for the Collection Service and SNMP Receiver — delayed-auto, dependency on network, failure actions, failureflag — is baked into Deploy.ps1 and applied consistently on both single-server and split-tier deployments.

Automated Installer

The EVOA installer (EnterpriseVoiceOperationsAnalyticsSetup.exe) performs every deployment step. Manual deployment via Deploy.ps1 is available for administrators who prefer scripted installs — see the Installation Guide for both paths.

Distribution Channel

- Server: signed .exe installer via the customer release package
- Desktop Notifier: standalone signed .exe (bypass SmartScreen prompts)
- Notifier updates: silent auto-update via /api/v1/desktop-notifier/version poll
- License Manager: separate on-premise install; edit-in-EVOA + sync-to-LM script keeps CallAccounting.Licensing DLL aligned across both trees

Contact

MeeraApps Technology

Website	https://www.meeraapps.com
Sales enquiry	info@meeraapps.com
Technical support	support@meeraapps.com
Phone	+966 573516885