

# Voclio System - Diagrams Documentation

Complete API Reference Documentation

Version 1.0.0 | Generated on 2024-01-15

This document explains each diagram in the Voclio system, including their purpose, content, benefits, usage, and when to use them.

---

## 1. Use Case Diagram

### What It Does

The Use Case Diagram illustrates the functional requirements of the Voclio system from a user's perspective. It shows all the actions users can perform with the system and how different actors interact with various use cases.

### What It Contains

- **Actors:** User, Super Admin, External Systems (LLM AI Processor, Calendar Service, Cloud Storage, Speech-to-Text Engine)
- **Use Cases:** Organized into categories:
  - Productivity Features (Smart Suggestions, Daily/Weekly Summary, Focus Mode)
  - Voice & Notes (Record, Transcribe, Search, Tag, Summarize, Convert to Tasks)
  - Task Management (Create, Edit, Delete, Sync with Calendar, Reminders)
  - Profile & Settings (Authentication, Subscription, Theme, Language, Backup)
  - Admin Functions (Security, App Configuration, User Management)
- **Relationships:** Includes, extends, and associations between use cases

### Benefits

- Provides clear overview of system functionality
- Helps stakeholders understand user requirements
- Identifies all user interactions with the system
- Shows dependencies between features

- Useful for requirement validation

## When to Use

- During requirements gathering phase
  - For stakeholder presentations
  - When defining system scope
  - As a reference for developers and testers
  - For project documentation and onboarding
- 

## 2. Entity Relationship Diagram (ERD)

### What It Does

The ERD provides a high-level view of the database structure, showing entities, their attributes, and relationships without detailed constraints. It serves as a simplified database schema overview.

### What It Contains

- **Entities:** User, SuperAdmin, Task, Note, VoiceRecording, Tag, CalendarSync, Reminder, ProductivityStreak, Achievement, FocusSession, Backup, ActivityLog, APIUsage, APIKey, AppConfig, Notification
- **Attributes:** Key fields for each entity (IDs, names, dates, statuses)
- **Relationships:** One-to-one, one-to-many, and many-to-many relationships
- **Primary Keys (PK)** and **Foreign Keys (FK)** indicators

### Benefits

- Quick understanding of database structure
- Easy to read for non-technical stakeholders

- Helps identify entity relationships
- Useful for initial database design discussions
- Lightweight representation of data model

## When to Use

- Initial database design phase
  - Quick reference for data structure
  - Presentations to business stakeholders
  - Documentation for new team members
  - As a simplified overview before detailed design
- 

## 3. Enhanced Entity Relationship Diagram (EERD)

### What It Does

The EERD provides a comprehensive and detailed view of the database schema with all attributes, data types, constraints, indexes, and detailed relationships. It's the complete technical specification for database implementation.

### What It Contains

- **Complete Entity Definitions:** All entities with full attribute details
- **Data Types:** VARCHAR, INT, TIMESTAMP, ENUM, JSON, DECIMAL, etc.
- **Constraints:** NOT NULL, UNIQUE, CHECK constraints
- **Indexes:** Performance optimization indexes
- **Relationships:** Detailed cardinalities (1:1, 1:M, M:M) with descriptions
- **Metadata:** Created\_at, updated\_at, deleted\_at (soft delete)

- **Encryption Indicators:** Fields marked for encryption
- **Business Rules:** Embedded in constraints and validations

## Benefits

- Complete technical specification for database implementation
- Reference for database developers
- Helps identify performance optimization opportunities
- Ensures data integrity through constraints
- Documents business rules in database layer
- Essential for database migration and maintenance

## When to Use

- Database implementation phase
  - When writing database migration scripts
  - For database optimization and indexing
  - During database refactoring
  - As technical reference for backend developers
  - For database security planning
- 

## 4. System Architecture Diagram

### What It Does

The System Architecture Diagram shows the high-level structure of the Voclio system using Clean Architecture principles. It illustrates the separation of concerns across different layers and how components interact.

## What It Contains

- **Presentation Layer:** Flutter UI, State Management (BLoC/Provider), Localization, Theme Manager
- **Application Layer:** Use Cases (Voice Note, Task, Note, Auth, Focus, Calendar, Analytics)
- **Domain Layer:**
  - Entities (User, Task, Note, VoiceRecording, Tag, FocusSession, ProductivityStreak)
  - Interfaces (Repository and Service interfaces)
- **Infrastructure Layer:** Repository implementations, Service implementations, Local/Cloud Storage
- **External Services:** Whisper API, GPT-5 API, Google/Apple Calendar, Firebase Auth, FCM

## Benefits

- Clear separation of concerns
- Follows Clean Architecture principles
- Easy to understand system layers
- Helps identify dependencies
- Guides development team structure
- Ensures maintainability and testability

## When to Use

- System design phase
- Team onboarding and training
- Architecture reviews
- When planning system refactoring
- For technical documentation
- As a guide for organizing code structure

---

## 5. System Design Diagram

### What It Does

The System Design Diagram shows the component-level design of the Voclio system, illustrating how different services, adapters, and components interact at a more detailed level than the architecture diagram.

### What It Contains

- **Client Layer:** UI Components, State Management, Local Cache, Audio Recorder, Calendar Widget
- **Business Logic Layer:**
  - Services (Authentication, Voice Note, Task, Note, LLM, Calendar Sync, Focus, Productivity, Notification)
  - Data Validation
- **Integration Layer:** API Adapters (Whisper, GPT, Google/Apple Calendar), API Client, File Storage Adapter
- **Data Layer:** Repository, Data Models, Local Database, Cache Manager
- **External Services:** Cloud Database, Whisper API, GPT-5 API, Calendar APIs, File Storage, Push Service, Auth Provider
- **Flow Annotations:** Detailed notes explaining data flow for each major operation

### Benefits

- Detailed view of component interactions
- Shows data flow through the system
- Identifies integration points
- Helps with component design decisions
- Useful for understanding service boundaries

- Guides microservice design

## When to Use

- Detailed system design phase
  - When designing API integrations
  - For component-level discussions
  - During service architecture planning
  - When understanding data flow
  - For integration testing planning
- 

## 6. User Flow Diagram

### What It Does

The User Flow Diagram illustrates the complete user journey through the Voclio application, showing step-by-step interactions, decision points, and different paths users can take.

### What It Contains

- **Authentication Flow:** Email/Google/Apple sign-in processes
- **Voice Note Flow:** Recording, transcription, editing, summarization, task conversion
- **Task Management Flow:** Voice/Manual/Note-based task creation, calendar sync, reminders
- **Note Management Flow:** Search, filter, view, edit, delete, tag operations
- **Focus Mode Flow:** Starting sessions, timer management, ambient sounds, completion tracking
- **Calendar Sync Flow:** Enabling/disabling sync, manual sync, event management
- **Productivity Flow:** Viewing summaries, streaks, achievements, smart suggestions



- **Settings Flow:** Profile updates, theme changes, language settings, backup/restore

## Benefits

- Complete user journey visualization
- Identifies all user interaction paths
- Helps identify UX improvements
- Useful for testing scenarios
- Guides UI/UX design decisions
- Helps create user stories

## When to Use

- UX/UI design phase
  - User testing scenarios
  - Creating user stories and acceptance criteria
  - Identifying edge cases and alternative flows
  - For documentation and training
  - When planning user onboarding
- 

## 7. Data Flow Diagram

### What It Does

The Data Flow Diagram shows how data moves through the system during specific operations, illustrating interactions between components, services, and external systems using sequence diagrams.

### What It Contains

- **Voice Note Creation Flow:** Recording → Transcription → Storage → Optional Summarization
- **Task Creation Flow:** Input → Validation → Local Storage → Calendar Sync → Cloud Sync
- **Note to Task Conversion:** Note Selection → LLM Processing → Task Extraction → Task Creation
- **Calendar Sync Flow:** Authentication → Event Fetching → Task Creation → Bidirectional Sync
- **Focus Session Flow:** Session Start → Timer Tracking → Completion → Streak Update
- **Search & Analytics Flow:** Query → Local Search → Cloud Search → Result Aggregation → AI Summary
- **Backup & Restore Flow:** Backup Creation → Data Export → Storage Upload → Restore Process
- **Admin Operations Flow:** User Management, API Usage Monitoring, Configuration Management

## Benefits

- Detailed understanding of data flow
- Identifies potential bottlenecks
- Shows integration points clearly
- Helps with debugging data issues
- Guides API design decisions
- Useful for performance optimization

## When to Use

- When designing API endpoints
- Understanding data synchronization
- Debugging data flow issues
- Performance optimization
- Integration testing
- API documentation

---

## 8. Sequence Diagram

### What It Does

The Sequence Diagram provides a detailed view of how different components interact over time during a specific operation, showing the order of messages and method calls.

### What It Contains

- **Voice Note to Task Flow:** Complete sequence from voice recording to task creation
- Voice Recording & Transcription (User → App → Whisper API)
- Save as Note (App → Backend → Database)
- Optional Summarization (App → GPT-5 API)
- Convert Note to Tasks (App → LLM → Task Service → Calendar)
- Alternative Direct Task Creation flow
- **Component Interactions:** Mobile App, API Gateway, Services, Database, External APIs
- **Activation Lifelines:** Shows when components are active
- **Message Flow:** Synchronous and asynchronous calls

### Benefits

- Clear visualization of component interactions
- Shows timing and order of operations
- Identifies dependencies between components
- Helps understand asynchronous operations
- Useful for debugging interaction issues

- Guides API design and contract definition

## When to Use

- Detailed design of specific features
  - Understanding complex interactions
  - API contract discussions
  - Debugging interaction issues
  - Performance analysis
  - Integration planning
- 

## 9. Activity Diagram

### What It Does

The Activity Diagram shows the detailed step-by-step process flow for a specific activity (Voice Note Processing), including decision points, parallel activities, and different outcomes.

### What It Contains

- **Recording Phase:** Start recording, continue recording, stop recording, save audio
- **Transcription Phase:** Audio upload, Whisper API call, transcription receipt, error handling
- **Note Creation Phase:** Note object creation, title/content setting, editing, saving, syncing
- **Optional Processing Phase:**
  - AI Summarization (if requested)
  - Tag Addition (if requested)
  - Task Conversion (if requested) with task extraction and calendar sync

- **Notification Phase:** Push notifications and UI updates

## **Benefits**

- Detailed process visualization
- Shows all decision points and branches
- Identifies error handling paths
- Useful for process documentation
- Helps identify optimization opportunities
- Guides implementation details

## **When to Use**

- Implementing specific features
  - Process documentation
  - Identifying edge cases
  - Error handling planning
  - Process optimization
  - Testing scenario creation
- 

# **10. Context Diagram**

## **What It Does**

The Context Diagram shows the Voclio system as a whole and its relationships with external entities and systems. It provides a high-level system boundary view.

## **What It Contains**

- **Voclio System Boundary:** Mobile App, Backend API, Services (Auth, Voice, Task, Note, LLM, Calendar, Analytics), Database, File Storage
- **External Actors:** User, Super Admin
- **External Services:**
  - Whisper API (Speech-to-Text)
  - GPT-5 API (LLM Processing)
  - Google/Apple Calendar APIs
  - Firebase Auth
  - Push Notification Service (FCM)
- **Data Flow:** Shows what data flows between system and external entities
- **Annotations:** Notes explaining each major component's purpose

## Benefits

- Clear system boundary definition
- Identifies all external dependencies
- Shows data flow with external systems
- Helps understand system scope
- Useful for stakeholder communication
- Guides integration planning

## When to Use

- Initial system design
- Stakeholder presentations
- Understanding system scope
- Integration planning

- Architecture overview
  - System boundary discussions
- 

## 11. Deployment Diagram

### What It Does

The Deployment Diagram shows the physical deployment architecture of the Voclio system, illustrating where components are deployed, hardware infrastructure, and deployment topology.

### What It Contains

- **Client Devices:** iOS App, Android App, Local Storage
- **Cloud Infrastructure:**
  - Load Balancer (API Gateway)
  - Application Servers (Microservices: Backend API, Auth, Voice, LLM, Task, Note, Calendar, Analytics)
  - Message Queue (Redis, RabbitMQ)
  - Database Cluster (PostgreSQL Primary/Replica, Elasticsearch)
  - File Storage (S3/Object Storage, CDN)
  - Monitoring & Logging (Prometheus, Grafana, ELK Stack)
- **External Services:** Whisper API, GPT-5 API, Firebase Auth, FCM, Calendar APIs
- **Connections:** Shows protocols (HTTPS, HTTP) and data flow
- **Deployment Notes:** Container orchestration, auto-scaling, health checks

### Benefits

- Complete deployment architecture view

- Identifies infrastructure requirements
- Shows scalability considerations
- Guides DevOps and infrastructure planning
- Helps with cost estimation
- Useful for deployment documentation

### **When to Use**

- Infrastructure planning
  - DevOps and deployment discussions
  - Cloud architecture design
  - Cost estimation
  - Scalability planning
  - Infrastructure documentation
- 

## **12. API Endpoints Diagram**

### **What It Does**

The API Endpoints Diagram provides a visual catalog of all REST API endpoints organized by functional categories, serving as a quick reference for API structure.

### **What It Contains**

- **Endpoint Categories:**
- Authentication Endpoints (Register, Login, OTP, Profile)
- Voice & Notes Endpoints (Upload, Transcribe, CRUD, Search, Summarize)



- Task Management Endpoints (CRUD, Stats, Bulk Operations)
- Calendar Sync Endpoints (Enable, Disable, Sync, Events)
- Tag Endpoints (CRUD operations)
- Focus Mode Endpoints (Session management, Stats)
- Productivity Endpoints (Streak, Achievements, Summary, Suggestions)
- Backup & Restore Endpoints (Create, List, Restore, Delete)
- Reminder Endpoints (CRUD, Snooze, Dismiss)
- Notification Endpoints (Email, WhatsApp, Push, Preferences)
- Settings Endpoints (Theme, Language, Subscription, Notifications)
- Admin Endpoints (User Management, API Usage, API Keys, Logs, Config)
- **Request/Response Examples:** Sample data structures for key endpoints

## Benefits

- Quick API reference
- Shows complete API structure
- Organized by functionality
- Helps with API discovery
- Useful for frontend development
- Guides API testing

## When to Use

- Frontend development
- API integration
- API testing and documentation
- Quick API reference

- When planning API usage
- For API versioning discussions

---

## Summary: When to Use Each Diagram

| Diagram | Primary Use Case | Audience |

|-----|-----|-----|

| **Use Case** | Requirements gathering, stakeholder communication | Business Analysts, Product Managers, Stakeholders |

| **ERD** | Quick database overview | All team members, Business stakeholders |

| **EERD** | Database implementation | Database developers, Backend developers |

| **System Architecture** | System design, team onboarding | Developers, Architects, Tech leads |

| **System Design** | Component design, integration planning | Developers, System architects |

| **User Flow** | UX/UI design, testing scenarios | UX designers, QA, Product managers |

| **Data Flow** | API design, integration testing | Backend developers, API designers |

| **Sequence** | Feature implementation, debugging | Developers, QA engineers |

| **Activity** | Process documentation, implementation | Developers, Business analysts |

| **Context** | System overview, stakeholder communication | All team members, Stakeholders |

| **Deployment** | Infrastructure planning, DevOps | DevOps engineers, System architects |

| **API Endpoints** | Frontend development, API integration | Frontend developers, API consumers |

---

**Document Version:** 1.0.0

**Last Updated:** 2024-01-15

**System:** Voclio

