LocaNext System Architecture

System Overview

HYBRID DEPLOYMENT MODEL

User's PC (Windows .exe)

- LocaNext.exe (Electron desktop app)
- Embedded Python Backend Server
- **SQLite Database (LOCAL)** - User's projects, files, translation memory
- ALL processing happens locally (FAST, works OFFLINE)
- Optionally sends telemetry to central server

Central Server (Cloud/On-Premise)

- FastAPI Server (Python 3.11+)
- PostgreSQL Database (aggregated data from all users)
- Admin Dashboard (web interface for monitoring)

Communication:

- HTTPS (REST API) for data exchange
- WSS (WebSocket) for real-time updates

Why Both SQLite AND PostgreSQL?

SQLite (In User's Desktop App)

Feature	Benefit
Fast local operations	No network latency

Feature	Benefit
Works completely OFFLINE	No internet required
No database server needed	Just run the .exe
User data stays on PC	Privacy/security
Isolated per user	No conflicts

PostgreSQL (Central Server)

Feature	Benefit
Handles concurrent writes	Many users at once
Aggregates telemetry	All installations
Powers Admin Dashboard	Monitoring
Stores update info	Version management
Production reliability	Enterprise-grade

This is NOT redundancy - they serve different purposes!

Communication Protocols

Layer	Protocol	Port	Purpose
API	REST/HTTPS	443	Data exchange
Real-time	WebSocket (WSS)	443	Live updates
Central DB	PostgreSQL	5432	Central storage
Local DB	SQLite	-	Offline data

Security Layers

1. IP RANGE FILTERING (Primary)

- Only company network IPs allowed (e.g., 192.168.11.0/24)
- All other IPs receive 403 Forbidden

2. CORS (Cross-Origin Resource Sharing)

- Whitelist of allowed origins only
- Blocks requests from unauthorized domains

3. JWT AUTHENTICATION

- Access Token: 30 minutes
- Refresh Token: 7 days
- Secure key generation required

4. TLS/HTTPS ENCRYPTION

- All data encrypted in transit
- TLS 1.2/1.3 supported

5. AUDIT LOGGING

- All login attempts logged
- Failed access attempts tracked
- 1 year retention

6. INPUT VALIDATION

- SQL Injection: Protected (SQLAlchemy ORM)
- XSS: Sanitized inputs
- Path Traversal: Blocked

Database Structure

User's Local (SQLite)

Table	Purpose
users	Local user data
projects	User's projects
files	Uploaded files
translation_memory	TM entries
sessions	Login sessions

Central Server (PostgreSQL)

Table	Purpose
users	All users (aggregated)
projects	All projects
telemetry_events	Usage tracking
telemetry_tool_events	Tool usage
sessions	All sessions
audit_logs	Security logs
error_reports	Error tracking

Total: 17 Tables | 63+ API Endpoints | 912 Tests

Application Components

Component	Technology	Purpose
Desktop App	Electron + Svelte	User interface
Local Backend	FastAPI + Python	API + Processing

Component	Technology	Purpose
Local Database	SQLite	Offline data storage
Central Database	PostgreSQL	Telemetry + Admin
Real-time	Socket.IO	Live updates
Admin Dashboard	SvelteKit	Monitoring

Data Flow

Step 1: User Action

User uploads Excel file in LocaNext desktop app

Step 2: Local Processing

- Python processes file on user's CPU
- **No data sent to cloud**
- Works completely offline
- Fast (no network latency)

Step 3: Result

Output file saved locally

Step 4: Telemetry (Optional)

- Usage statistics sent to central server
- Error tracking
- Update distribution

Test Coverage

Category	Tests	Status
Unit Tests	377+	PASS
API Tests	168	PASS
Security Tests	86	PASS
E2E Tests	164	PASS
Total	**912**	**100%**

LocaNext v2512110832