Project ECO

JAN 2026

Built for Signal, not Noise.

Developed by Velan

Tel +91-9591290165

Shivajinagar
Bangalore, KA, IN

www.linkedin.com/in/velan-e
velane929@gmail.com

Contents

Professional Documentation & Implementation Guide

Executive Summary

ECO (Executive Context Engine) represents a paradigm shift in professional networking intelligence. The platform aggregates multi-source data streams, Wikipedia profiles, real-time news feeds, social media activity, and AI-synthesized insights, to provide comprehensive executive context in seconds. Built for professionals who demand signal over noise, ECO transforms executive research from hours of manual work into an intelligent, automated workflow.

Platform Architecture

Core Technology Stack

Frontend Framework: Vanilla JavaScript with modern ES6+ features **Styling Engine:** CSS3 with custom design and Tailwind-inspired utilities

Animation Library: Vanta.js for dynamic 3D backgrounds

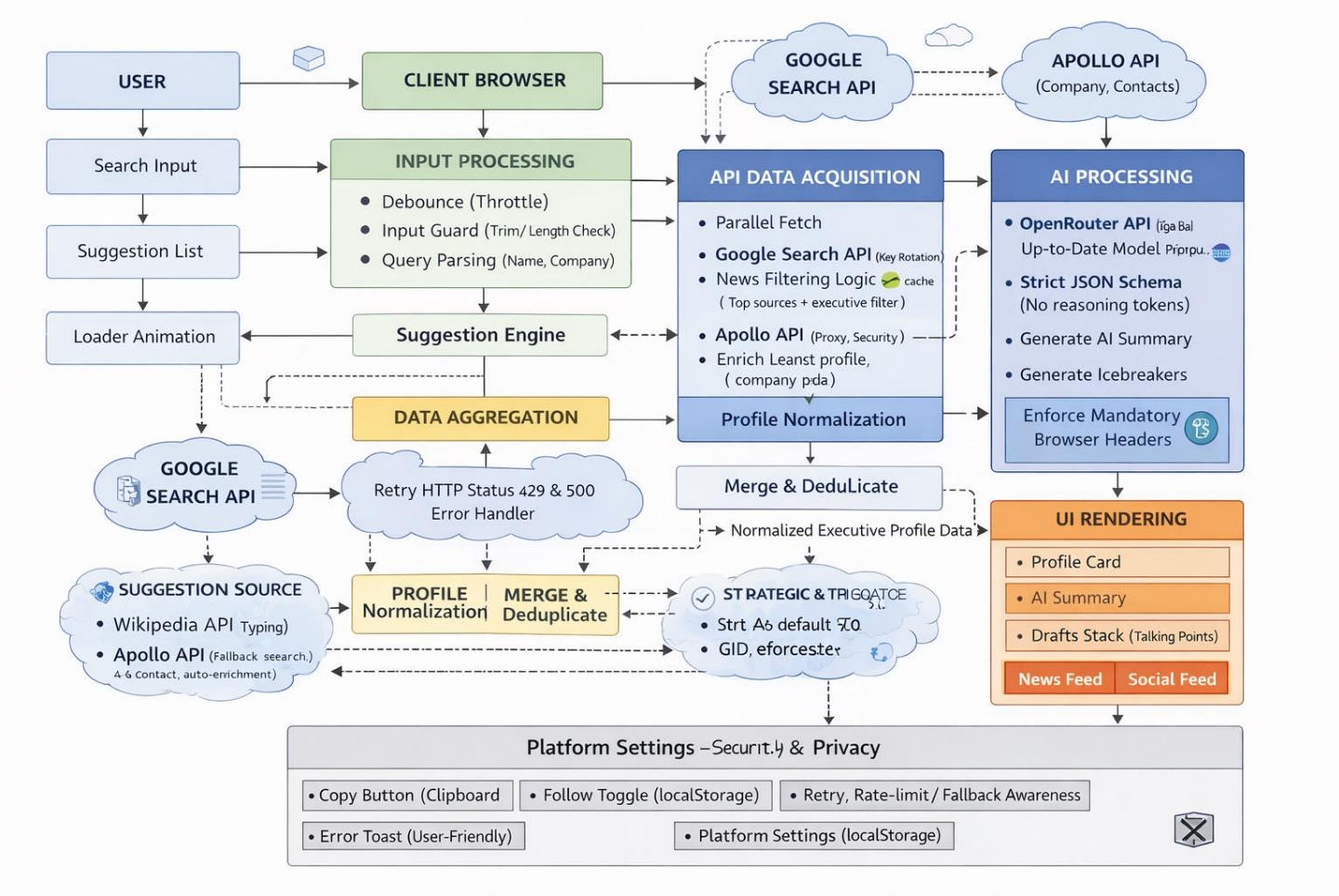
Data Visualization: D3.js v7 for network graph rendering

Icon System: Lucide Icons for consistent visual language

Typography: Sora font family for modern, professional aesthetics

API Integration Layer

Search Intelligence: Google Custom Search API with automatic failover **AI Processing:** OpenRouter API with Gemma 2 9B model **Profile Aggregation:** Wikipedia REST API v1 **Contact Discovery:** Apollo.io API (optional) **Social Integration:** Twitter/X oEmbed API, LinkedIn public data



Installation & Deployment

Method 1: Direct Browser Access

Clone the repository to your local environment:

bash

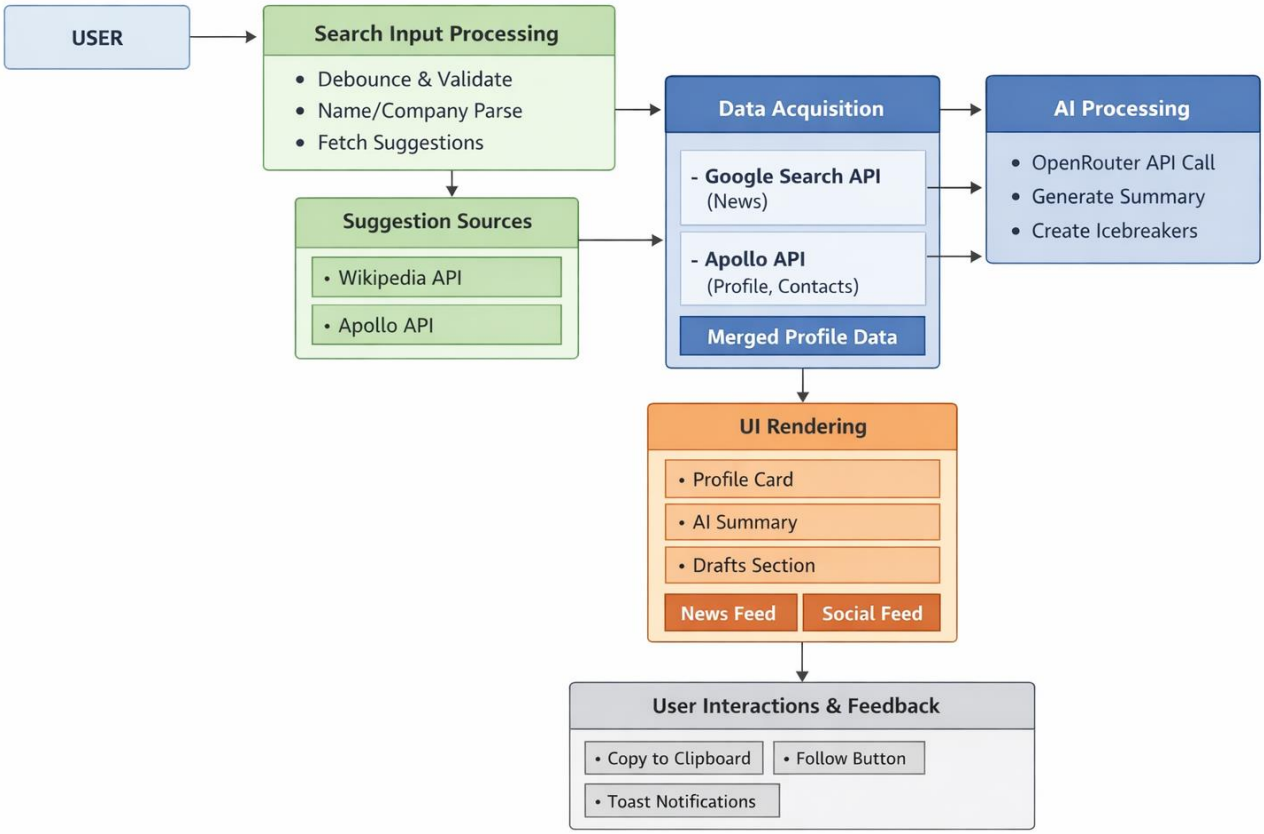
git clone https://github.com/chilly23/Context-First-Networking---Beta

cd eco-platform

Open index.html directly in any modern web browser. The application is fully client-side and requires no compilation or build process.

Method 2: Local Development Server

For optimal development experience with hot-reload capabilities:



Python Environment:

bash

python -m http.server 8000

Node.js Environment:

bash

npx http-server -p 8000

PHP Environment:

bash

php -S localhost:8000

```

Navigate to `http://localhost:8000` in your browser.
```

Method 3: Production Deployment

Deploy to any static hosting service:
Netlify: Drag and drop the project folder
Vercel: Connect repository and deploy automatically
GitHub Pages: Enable in repository settings
AWS S3: Upload files and configure static website hosting

API Configuration Guide
Google Custom Search API Setup
Navigate to Google Cloud Console and complete the following sequence:

1. Create a new project or select an existing project
2. Enable the Custom Search API from the API Library
3. Generate API credentials under "Credentials" section
4. Create a Custom Search Engine at `https://cse.google.com`
5. Configure search settings to target the entire web
6. Extract the CX identifier from the search engine details

Primary Configuration:
- API Key: Required for search operations
- CX ID: Identifies your custom search engine instance

Secondary Configuration:
- Backup API Key: Automatic failover when primary quota exhausted
- Backup CX ID: Secondary search engine for redundancy

OpenRouter API Configuration

Access `https://openrouter.ai` and complete account setup:

1. Register for a developer account
2. Navigate to API Keys section in dashboard
3. Generate a new API key with appropriate permissions
4. Configure model access (Gemma 2 9B recommended)

The platform uses OpenRouter for:
- Executive summary generation
- Icebreaker message synthesis
- Introduction email drafting

Apollo.io Integration

Access `https://apollo.io` and complete the following:

1. Create an Apollo account with API access tier
2. Navigate to Settings and locate API section
3. Generate an API key with contact search permissions
4. Note: This integration is optional but enhances contact discovery

Configuration Implementation

Access the settings panel via the gear icon in the top-right corner. Enter your API credentials in the designated fields:

Required Credentials:
- Google API Key (Primary)
- Google CX ID (Primary)
- OpenRouter API Key

Optional Credentials:
- Google API Key (Secondary)
- Google CX ID (Secondary)
- Apollo API Key

All credentials are stored locally in browser localStorage and never transmitted to

external servers beyond their intended API endpoints.

User Interface Guide

Search Interface

The primary search interface features an intelligent autocomplete system powered by Wikipedia's OpenSearch API. Begin typing an executive name, and the system presents real-time suggestions with profile images.

Keyboard Navigation:
- Arrow Down: Navigate to next suggestion
- Arrow Up: Navigate to previous suggestion
- Enter: Select highlighted suggestion or execute search
- Escape: Close suggestion dropdown

Search Capabilities:
- Executive names (e.g., "Satya Nadella")
- Company + Title combinations (e.g., "Microsoft CEO")
- Partial name matching with fuzzy search

Results Dashboard

The results view employs a sophisticated Bento Grid layout, organizing information into distinct contextual cards:

Left Column: Profile & News
- Executive profile card with Wikipedia-sourced imagery
- AI-generated synthesis with contextual insights
- Curated news feed with recent articles (past 30 days)
- Visual news cards with source attribution

Right Column: Social Intelligence
- X (Twitter) posts with embedded previews
- LinkedIn activity and recent posts
- Real-time social media engagement

Network Management

Following Functionality:
The Follow button adds executives to your personal network. Followed executives appear in:
- Network Feed (when enabled)

- Network Graph visualization
- Persistent tracking across sessions

Network Graph:
Access via "View Network Graph" in settings. The circular visualization displays:
- Central node: Your position
- Peripheral nodes: Followed executives
- Connections: Direct relationships
- Interactive elements: Hover for details

Network Feed:
Enable via toggle switch in settings. Displays:
- Recent news for followed executives
- Aggregated activity streams
- Personalized content recommendations

AI-Powered Features

Executive Synthesis:
AI-generated summaries provide:
- 2-3 sentence biographical overview
- Key achievements and positions
- Recent notable activities

Icebreaker Messages:
Three AI-generated conversation starters designed for:
- Professional networking contexts
- Email introductions
- LinkedIn connection requests

Draft Introduction:
Personalized email generation based on:
- Your "About Me" profile
- Target executive's background
- Professional context and industry

Customization Options

Theme Selection:
- Dark Mode: Optimized for extended viewing sessions
- Light Mode: Professional aesthetic for presentations

Background Effects:
- Grid: Minimal geometric pattern
- Dots: Animated particle system
- Fog: Ethereal volumetric effect
- Net: Connected network visualization

Profile Configuration:

- Username: Display name throughout interface
- About Me: Professional biography for AI-generated content

Data Architecture

Local Storage Schema

All data persists in browser localStorage with the following structure:

User Configuration:
```
eco_username: string
eco_about: string (textarea)
eco_theme: "dark" | "light"
eco_bg: "grid" | "dots" | "fog" | "net"
```

API Credentials:
```
eco_google_api: string
eco_google_cx: string
eco_google_api_2: string (optional)
eco_google_cx_2: string (optional)
eco_openrouter_api: string
eco_apollo_api: string (optional)
```

Network Data:
```
eco_network: Array<{
  name: string,
  image?: string,
  addedAt: ISO8601 timestamp
}>
```

API Response Processing

Google Search Results:
Processed through `fetchGoogleResults()` function with automatic pagination and error handling. Implements intelligent key rotation when quota limits encountered.

Wikipedia Profiles:
Fetched via REST API v1 with fallback placeholder images for executives without profile photos.

AI Summaries:

Structured JSON response from OpenRouter with fields:
- summary: string (2-3 sentences)
- icebreakers: string[] (array of 3 messages)
- error: string | null

Apollo Contacts:
Returns structured contact data:
```
{
  email: string | null,
  linkedin: string | null,
  twitter: string | null,
  phone: string | null
}
```

Performance Optimization

Loading Strategy

The application implements a minimum 4-second loading period to prevent jarring transitions. This strategic delay:

- Ensures all API calls complete
- Provides smooth user experience
- Allows proper data aggregation
- Prevents partial render states

API Rate Limiting

Google Search Quota Management:

- Primary/secondary key rotation
- Automatic failover on 429 responses
- Per-page retry logic
- Graceful degradation

Request Optimization:

- Parallel API calls via Promise.all()
- Single autocomplete requests (debounced 300ms)
- Cached Wikipedia profile images
- Lazy-loaded social media embeds

Browser Compatibility

Tested and optimized for:

- Chrome 90+ (recommended)
- Firefox 88+
- Safari 14+
- Edge 90+

Security Considerations

API Key Protection

API keys stored in localStorage remain client-side only. The application:

- Never transmits keys to third-party servers
- Uses HTTPS for all API communications

- Implements no server-side key storage
- Provides user-controlled key management

Data Privacy

User data handling principles:

- No account creation required
- No server-side data storage
- Local-only persistent storage
- User-controlled data deletion via browser tools

CORS Handling

All external APIs accessed directly from client:

- Wikipedia API: CORS-enabled by default
- Google Custom Search: API key authentication
- OpenRouter: Bearer token authentication
- Twitter oEmbed: Public endpoint with CORS support

Troubleshooting Guide

Common Configuration Issues

Search Returns No Results: Verify Google API credentials are correctly entered and Custom Search Engine targets "entire web" rather than specific sites.

AI Features Unavailable: Confirm OpenRouter API key has sufficient credits and model access permissions.

Contact Information Missing: Apollo.io integration is optional. Without valid Apollo API key, contact buttons link to search pages.

Theme Not Persisting: Clear browser cache and localStorage, then reconfigure preferences.

API Quota Management

Google Search Limits: Free tier: 100 queries per day Configure secondary API key for automatic failover

OpenRouter Limits: Varies by account tier and model selection Monitor usage in OpenRouter dashboard

Performance Issues

Slow Search Results: Check network connectivity and API response times Verify Google Custom Search Engine configuration

Failed Network Graph: Ensure D3.js library loads successfully Check browser console for JavaScript errors

---

Production Deployment Checklist

Pre-Deployment:

- Remove hardcoded API keys from config.js
- Minimize CSS and JavaScript files
- Optimize image assets
- Configure error tracking
- Set up analytics integration

Post-Deployment:

- Verify all API endpoints accessible
- Test search functionality across devices
- Confirm theme persistence
- Validate network graph rendering
- Monitor console for errors

Ongoing Maintenance:

- Monitor API quota usage
- Update dependencies periodically
- Review error logs
- Collect user feedback

- Optimize based on usage patterns

Technical Support

For implementation assistance, configuration guidance, or technical questions, reference the following resources:

API Documentation:

- Google Custom Search: https://developers.google.com/custom-search
- OpenRouter: https://openrouter.ai/docs
- Wikipedia API: https://www.mediawiki.org/wiki/API
- Apollo.io: https://apolloio.github.io/apollo-api-docs

Library Documentation:

- Vanta.js: https://www.vantajs.com
- D3.js: https://d3js.org
- Lucide Icons: https://lucide.dev

License & Attribution

This project utilizes open-source libraries and public APIs. Ensure compliance with respective license terms when deploying to production environments.

Third-Party Services: All API providers require individual account creation and adherence to their terms of service. Review each provider's documentation for usage limits, pricing tiers, and compliance requirements

Version: 1.0.0 (Beta) **Last Updated:** January 2026 **Minimum Browser Version:** Chrome 90+, Firefox 88+, Safari 14+

Company Information

Developed by Velan
Shivajinagar
Bangalore, KA, IN
Tel +91-9591290165
Fax [Fax]
www.linkedin.com/in/velan-e