# [**https://gemini.google.com/u/1/app/4e4c57eff1c5afd3**](https://gemini.google.com/u/1/app/4e4c57eff1c5afd3)

<https://gemini.google.com/u/1/app/4e4c57eff1c5afd3>

<https://gemini.google.com/u/1/app/cfcea269c16d7792>

Project Overview: Via di Francesco Pilgrimage Platform Database (Updated May 18, 2025 -- V2.1)

=============================================================================================

1. 🚀 Executive Summary

The Via di Francesco Pilgrimage Platform aims to be the definitive digital companion for pilgrims, offering comprehensive, reliable, and engaging information for planning and undertaking their journeys. This V2.1 update of the database overview reflects significant progress across all modules, emphasizing standardization towards V2 patterns, enhanced data integrity, and improved support for multilingual content and rich media.

Key advancements in this iteration include the consistent application of standardized audit columns (`created\_by\_profile\_id`, `updated\_by\_profile\_id`, `created\_at`, `updated\_at`) across almost all tables, refined lifecycle management for master data (`is\_active` flags) and transactional data (`deleted\_at` timestamps), a more robust and centralized internationalization (i18n) strategy leveraging the `public.translations` table with primary language (English) content stored directly in entity tables, and enhanced media management through dedicated linking tables and semantic roles. Furthermore, Row-Level Security (RLS) policies have been refined and are consistently designed to integrate with Supabase Auth and standardized helper functions. Many modules now also feature specialized views to optimize data retrieval for API endpoints, particularly for localized content.

2. 🎯 Refined Goals & Success Metrics

This V2.1 update reinforces the project's core goals by enhancing data quality, system maintainability, and the richness of information provided to users. Specific refinements and new objectives supported by these updates include:

- Enhanced Data Integrity & Auditability: Standardized audit columns and lifecycle flags across all modules significantly improve data tracking and governance capabilities.

- Improved Multilingual Support: Centralized translation management with primary language direct storage makes content more accessible and easier to manage globally.

- Richer Media Integration: New media linking strategies allow for more versatile and context-specific image and file associations (e.g., galleries, specific roles for articles).

- Streamlined API Performance: Introduction of localized and summary views aims to optimize data retrieval for common API queries.

- Increased Pilgrim Safety: The Dynamic Conditions module (Module 5) provides more structured and verifiable information about trail warnings and hazards.

- Better Pilgrim Planning: Detailed information for accommodations (Module 4a), attractions, food/water sources, shops/services (Module 4b), transport (Module 4c), and events (Module 4d) allows for more comprehensive journey planning.

- Enhanced User Engagement: Module 6 (User Interaction) enables waypoint voting and moderated tips, while Module 8 (Platform Content) supports rich articles and news.

3. 👥 Primary Personas & User Stories

The primary user roles remain consistent:

- Pilgrims: Planning and undertaking journeys, seeking information on routes, POIs, accommodations, conditions.

- Accommodation Hosts: Managing their listings and availability.

- Regional Content Managers: Updating local information, trail conditions, and POIs.

- Platform Administrators: Overseeing content, system integrity, and user management. Module updates provide richer data and better tools for all personas, for instance, through more detailed filtering for pilgrims or more structured content management for hosts and administrators.

4. 🗺️ Module Map (Snapshot Table)

| Module | Core Purpose | Key Tables | Version | Major Changes (V2.1 Rollup) |

| :-------------------------------------------------- | :------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- | :------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------ | :---------------------------------------- | :-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- |

| 1. User & Content Infrastructure | Manages users, roles (JWT sync), languages, media metadata, and platform-wide translations. | profiles, user\_roles\_master, languages\_master, media, translations. | 2.3 | Standardized audit columns (created\_by\_profile\_id, updated\_by\_profile\_id); icon\_identifier in master tables; primary language (English) content stored directly in main tables, others in public.translations; API returns base field in English & \*\_translations object; profiles.last\_activity\_at added; role synchronization to auth.users.raw\_app\_meta\_data.roles via triggers. |

| 2. Core Trail Hierarchy | Defines structure and characteristics of trails, routes, and segments, including master data for terrain and usage. | trails, routes, segments, route\_segments, terrain\_types\_master, usage\_types\_master, segment\_media. | 2.2 (API Aligned) | GPX file management shifted to media table (via media\_id); segment\_media enhanced with media\_role\_code and translatable captions/alt-text; surrogate id PKs for junction tables (trail\_regions, route\_segments); terrain\_types\_master & usage\_types\_master updated to full V2 spec (is\_active, audit columns, translatable names); all module tables now have full V2 audit columns; new localized views for API support (v\_trails\_detailed\_localized, etc.). |

| 3. Geographical Context | Defines and categorizes geographical areas (regions, provinces, towns) with multilingual support and auditability. | regions, provinces, towns, characteristic\_tags\_master, service\_tags\_master, town\_types\_master. | 2.2 (API-Aware) | Consistent V2 audit columns and lifecycle management (is\_active for master/provinces; deleted\_at & content\_visibility\_status for regions/towns); master tables (characteristic\_tags\_master, etc.) enhanced with is\_active, audit, translatable names, icons; array FK validation triggers (e.g., trigger\_validate\_characteristics\_tags); new region\_media & town\_media linking tables for galleries; localized views (v\_towns\_list\_localized, v\_regions\_list\_localized) for API efficiency. |

| 4. Waypoint Detail Modules (General) | Manages core POI ("waypoint") information, categorization, tagging, and content lifecycle. | waypoints, waypoint\_categories\_master, tags\_master, content\_statuses\_master. | 1.2 | Finalized core tables (waypoints v1.3, master tables v1.2/1.3) to V2 patterns (audit columns, is\_active for master data, deleted\_at for waypoints, i18n support via public.translations); localized views (v\_waypoint\_categories\_localized, v\_tags\_localized) introduced for efficient localized master data retrieval; array FK integrity trigger trigger\_check\_waypoint\_subcategory\_tags for waypoints.waypoint\_subcategory\_tag\_ids. |

| 4a. Waypoint - Accommodations | Manages detailed accommodation-specific information including types, amenities, rooms, pricing, and reviews. | accommodations, accommodation\_reviews, various \*\_master tables (e.g., accommodation\_types\_master), various accommodation\_\* junction tables. | 1.0 (New) | New module. Central accommodations table (v1.5); numerous specialized master tables (e.g., accommodation\_types\_master v1.4, amenities\_master v1.5) aligned with V2 patterns (audit, is\_active, translatable labels/descriptions, icons, sort order); junction tables (e.g., accommodation\_amenities v1.4) with translatable notes and audit columns; new accommodation\_reviews table (v1.0) with moderation workflow; localized views (e.g., v\_accommodation\_types\_localized) for all master tables; accommodations\_capacity\_summary\_view; specific translation cleanup triggers per table. |

| 4b. Waypoint - Attractions, Food/Water, Shops/Services | Enriches waypoints with specific details for attractions (incl. religious services), food/water sources, and shops/services. | attractions\_details, religious\_service\_schedules, food\_water\_sources\_details, shops\_and\_services\_details, numerous \*\_master tables, \*\_media linking tables. | 1.1 (New) | New module. Detail tables (attractions\_details v1.3.1, etc.) with translatable fields; numerous master tables (attraction\_types\_master v1.1, etc.) with V2 patterns (is\_active, audit, translatable labels, icons, sort order); dedicated \*\_media linking tables (e.g., attraction\_details\_media v1.0) for galleries with roles and translatable overrides; localized views (v\_waypoint\_attraction\_details\_localized v1.0, etc.) for efficient API data retrieval; "active check" triggers for all FKs to master tables; array FK integrity triggers; shops\_and\_services\_details.deleted\_at for independent soft delete. |

| 4c. Waypoint - Transportation | Manages public transportation stop information, types, facilities, and operational details. | transport\_stops\_details, transport\_stop\_types\_master, transport\_stop\_facilities\_master, view\_transport\_stops\_enriched. | 2.1 (New) | New module. Master tables (transport\_stop\_types\_master v2.1, transport\_stop\_facilities\_master v2.1) with V2 patterns (is\_active, full audit, translatable labels, icons, sort order); transport\_stops\_details table (v2.1) links to waypoints, includes translatable operator\_names\_text array, and stop\_facility\_ids array FK with validation trigger check\_transport\_stop\_facility\_ids; new view\_transport\_stops\_enriched (v2.1) for API efficiency; orphan translation cleanup triggers; standardized API translation model (primary lang field + translations object); transport\_stops\_details.deleted\_at added for soft delete. |

| 4d. Waypoint - Events | Manages detailed information about events associated with waypoints, including classification, timing, and scale. | events\_details, event\_types\_master, event\_recurrence\_frequencies\_master, event\_attendance\_scales\_master, event\_date\_certainty\_levels\_master. | 1.1 (New) | New module. Central events\_details table (v1.3); master tables (event\_types\_master v1.1, etc.) with V2 patterns (code, default\_name (translatable), is\_active, audit columns, sort order, icons); event\_theme\_or\_focus\_tag\_ids (array FK to tags\_master) with validation trigger (trigger\_validate\_event\_theme\_tags); orphan translation cleanup triggers for all module tables; events\_details.deleted\_at for soft deletion; full V2 audit columns on all module tables; primary language content in direct fields, other languages in public.translations. |

| 5. Dynamic Conditions | Manages and disseminates timely, verifiable information about warnings, hazards, closures, and advisories for trail segments. | segment\_warnings, warning\_types\_master, warning\_severities\_master, warning\_source\_types\_master, workflow\_statuses\_master. | 2.1 | Master tables (warning\_types\_master v2.1, etc.) aligned to V2 patterns (id PK, code, display\_name (translatable), is\_active, audit columns by profiles.id, icons, sort order); segment\_warnings table (v2.1) with segment\_id (FK), warning\_type\_id (FK), is\_currently\_active (generated column), workflow\_status\_code (FK), and audit by auth.users.id; public\_active\_segment\_warnings\_view (v1.0) proposed for simplified public access to active warnings; orphan translation cleanup triggers. |

| 6. User Interaction Module | Capturing user votes on waypoints, managing user-submitted tips with moderation, and defining standardized, translatable categories for tips. | user\_waypoint\_votes, user\_waypoint\_short\_tips, tip\_categories\_master. | 2.0 | tip\_categories\_master (v2.1) updated to V2 patterns (category\_code PK, translatable default\_name/description, is\_active, audit); user\_waypoint\_votes (v2.1.1) with trigger update\_waypoint\_vote\_counts to denormalize counts onto public.waypoints; user\_waypoint\_short\_tips (v2.1.1) with is\_publicly\_visible generated column and detailed moderation fields; view\_tip\_categories\_localized for API support. |

| 7. Curated Itinerary Module | Defines, structures, categorizes, and manages the lifecycle of pre-defined pilgrimage itineraries and their associated lookup data. | curated\_itineraries, curated\_itinerary\_segments, itinerary\_categories\_master, seasons\_master, trail\_difficulty\_levels\_master, content\_statuses\_master. | 2.1 | Promotion of ENUMs (difficulty, status, categories, seasons) to full master tables (v2.1) with V2 patterns (code, name (translatable), is\_active, audit, icons, sort order); full V2 audit columns on all module tables; orphan translation cleanup triggers; curated\_itineraries (v3.1) with content\_status\_id FK (NOT NULL); new junction tables curated\_itinerary\_to\_category (v2.1) and curated\_itinerary\_to\_season (v2.1); localized views (v\_curated\_itineraries\_list\_localized, etc.). User GWI Event Manager

- \*\*Module 8: Platform Content | Manages general platform content like news, articles, and blog posts, supporting rich text, multimedia, author attribution, and associations. | `articles`, `media\_roles\_master`, `article\_media`, `published\_articles\_view`, `article\_media\_details\_view`. | 1.0 (New) | New module providing a framework for creating, managing, and displaying dynamic content like news and articles. Supports featured images, media galleries, author attribution, and multilingual capabilities via `public.translations`. Includes RLS policies for content integrity and access. Standard audit columns and soft delete (`deleted\_at`) are included. `media\_roles\_master` defines media use within articles. Plogging data from the user about each of modules to create a summary.

Done.

Now generating the comprehensive project overview document.Project Overview: Via di Francesco Pilgrimage Platform Database (Updated May 18, 2025 -- V2.1)

========================================================================================================================================================

1. 🚀 Executive Summary

The Via di Francesco Pilgrimage Platform aims to be the definitive digital companion for pilgrims, offering comprehensive, reliable, and engaging information for planning and undertaking their journeys. This V2.1 update of the database overview reflects significant progress across all modules, emphasizing standardization towards V2 patterns, enhanced data integrity, and improved support for multilingual content and rich media.

Key advancements in this iteration include the consistent application of standardized audit columns (`created\_by\_profile\_id`, `updated\_by\_profile\_id`, `created\_at`, `updated\_at`) across almost all tables, refined lifecycle management for master data (`is\_active` flags) and transactional data (`deleted\_at` timestamps), a more robust and centralized internationalization (i18n) strategy leveraging the `public.translations` table with primary language (English) content stored directly in entity tables, and enhanced media management through dedicated linking tables and semantic roles. Furthermore, Row-Level Security (RLS) policies have been refined and are consistently designed to integrate with Supabase Auth and standardized helper functions. Many modules now also feature specialized views to optimize data retrieval for API endpoints, particularly for localized content.

2. 🎯 Refined Goals & Success Metrics

This V2.1 update reinforces the project's core goals by enhancing data quality, system maintainability, and the richness of information provided to users. Specific refinements and new objectives supported by these updates include:

- Enhanced Data Integrity & Auditability: Standardized audit columns and lifecycle flags across all modules significantly improve data tracking and governance capabilities.

- Improved Multilingual Support: Centralized translation management with primary language direct storage makes content more accessible and easier to manage globally.

- Richer Media Integration: New media linking strategies allow for more versatile and context-specific image and file associations (e.g., galleries, specific roles for articles).

- Streamlined API Performance: Introduction of localized and summary views aims to optimize data retrieval for common API queries.

- Increased Pilgrim Safety: The Dynamic Conditions module (Module 5) provides more structured and verifiable information about trail warnings and hazards.

- Better Pilgrim Planning: Detailed information for accommodations (Module 4a), attractions, food/water sources, shops/services (Module 4b), transport (Module 4c), and events (Module 4d) allows for more comprehensive journey planning.

- Enhanced User Engagement: Module 6 (User Interaction) enables waypoint voting and moderated tips, while Module 8 (Platform Content) supports rich articles and news.

3. 👥 Primary Personas & User Stories

The primary user roles remain consistent:

- Pilgrims: Planning and undertaking journeys, seeking information on routes, POIs, accommodations, conditions.

- Accommodation Hosts: Managing their listings and availability.

- Regional Content Managers: Updating local information, trail conditions, and POIs.

- Platform Administrators: Overseeing content, system integrity, and user management. Module updates provide richer data and better tools for all personas, for instance, through more detailed filtering for pilgrims or more structured content management for hosts and administrators.

4. 🗺️ Module Map (Snapshot Table)

| Module | Core Purpose | Key Tables | Version | Major Changes (V2.1 Rollup) |

| :-------------------------------------------------- | :------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- | :------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------ | :---------------------------------------- | :-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- |

| 1. User & Content Infrastructure | Manages users, roles (JWT sync), languages, media metadata, and platform-wide translations. | profiles, user\_roles\_master, languages\_master, media, translations. | 2.3 | Standardized audit columns (created\_by\_profile\_id, updated\_by\_profile\_id); icon\_identifier in master tables; primary language (English) content stored directly in main tables, others in public.translations; API returns base field in English & \*\_translations object; profiles.last\_activity\_at added; role synchronization to auth.users.raw\_app\_meta\_data.roles via triggers. |

| 2. Core Trail Hierarchy | Defines structure and characteristics of trails, routes, and segments, including master data for terrain and usage. | trails, routes, segments, route\_segments, terrain\_types\_master, usage\_types\_master, segment\_media. | 2.2 (API Aligned) | GPX file management shifted to media table (via media\_id); segment\_media enhanced with media\_role\_code and translatable captions/alt-text; surrogate id PKs for junction tables (trail\_regions, route\_segments); terrain\_types\_master & usage\_types\_master updated to full V2 spec (is\_active, audit columns, translatable names); all module tables now have full V2 audit columns; new localized views for API support (v\_trails\_detailed\_localized, etc.). |

| 3. Geographical Context | Defines and categorizes geographical areas (regions, provinces, towns) with multilingual support and auditability. | regions, provinces, towns, characteristic\_tags\_master, service\_tags\_master, town\_types\_master. | 2.2 (API-Aware) | Consistent V2 audit columns and lifecycle management (is\_active for master/provinces; deleted\_at & content\_visibility\_status for regions/towns); master tables (characteristic\_tags\_master, etc.) enhanced with is\_active, audit, translatable names, icons; array FK validation triggers (e.g., trigger\_validate\_characteristics\_tags); new region\_media & town\_media linking tables for galleries; localized views (v\_towns\_list\_localized, v\_regions\_list\_localized) for API efficiency. |

| 4. Waypoint Detail Modules (General) | Manages core POI ("waypoint") information, categorization, tagging, and content lifecycle. | waypoints, waypoint\_categories\_master, tags\_master, content\_statuses\_master. | 1.2 | Finalized core tables (waypoints v1.3, master tables v1.2/1.3) to V2 patterns (audit columns, is\_active for master data, deleted\_at for waypoints, i18n support via public.translations); localized views (v\_waypoint\_categories\_localized, v\_tags\_localized) introduced for efficient localized master data retrieval; array FK integrity trigger trigger\_check\_waypoint\_subcategory\_tags for waypoints.waypoint\_subcategory\_tag\_ids. |

| 4a. Waypoint - Accommodations | Manages detailed accommodation-specific information including types, amenities, rooms, pricing, and reviews. | accommodations, accommodation\_reviews, various \*\_master tables (e.g., accommodation\_types\_master), various accommodation\_\* junction tables. | 1.0 (New) | New module. Central accommodations table (v1.5); numerous specialized master tables (e.g., accommodation\_types\_master v1.4, amenities\_master v1.5) aligned with V2 patterns (audit, is\_active, translatable labels/descriptions, icons, sort order); junction tables (e.g., accommodation\_amenities v1.4) with translatable notes and audit columns; new accommodation\_reviews table (v1.0) with moderation workflow; localized views (e.g., v\_accommodation\_types\_localized) for all master tables; accommodations\_capacity\_summary\_view; specific translation cleanup triggers per table. |

| 4b. Waypoint - Attractions, Food/Water, Shops/Services | Enriches waypoints with specific details for attractions (incl. religious services), food/water sources, and shops/services. | attractions\_details, religious\_service\_schedules, food\_water\_sources\_details, shops\_and\_services\_details, numerous \*\_master tables, \*\_media linking tables. | 1.1 (New) | New module. Detail tables (attractions\_details v1.3.1, etc.) with translatable fields; numerous master tables (attraction\_types\_master v1.1, etc.) with V2 patterns (is\_active, audit, translatable labels, icons, sort order); dedicated \*\_media linking tables (e.g., attraction\_details\_media v1.0) for galleries with roles and translatable overrides; localized views (v\_waypoint\_attraction\_details\_localized v1.0, etc.) for efficient API data retrieval; "active check" triggers for all FKs to master tables; array FK integrity triggers; shops\_and\_services\_details.deleted\_at for independent soft delete. |

| 4c. Waypoint - Transportation | Manages public transportation stop information, types, facilities, and operational details. | transport\_stops\_details, transport\_stop\_types\_master, transport\_stop\_facilities\_master, view\_transport\_stops\_enriched. | 2.1 (New) | New module. Master tables (transport\_stop\_types\_master v2.1, transport\_stop\_facilities\_master v2.1) with V2 patterns (is\_active, full audit, translatable labels, icons, sort order); transport\_stops\_details table (v2.1) links to waypoints, includes translatable operator\_names\_text array, and stop\_facility\_ids array FK with validation trigger check\_transport\_stop\_facility\_ids; new view\_transport\_stops\_enriched (v2.1) for API efficiency; orphan translation cleanup triggers; standardized API translation model (primary lang field + translations object); transport\_stops\_details.deleted\_at added for soft delete. |

| 4d. Waypoint - Events | Manages detailed information about events associated with waypoints, including classification, timing, and scale. | events\_details, event\_types\_master, event\_recurrence\_frequencies\_master, event\_attendance\_scales\_master, event\_date\_certainty\_levels\_master. | 1.1 (New) | New module. Central events\_details table (v1.3); master tables (event\_types\_master v1.1, etc.) with V2 patterns (code, default\_name (translatable), is\_active, audit columns, sort order, icons); event\_theme\_or\_focus\_tag\_ids (array FK to tags\_master) with validation trigger (trigger\_validate\_event\_theme\_tags); orphan translation cleanup triggers for all module tables; events\_details.deleted\_at for soft deletion; full V2 audit columns on all module tables; primary language content in direct fields, other languages in public.translations. |

| 5. Dynamic Conditions | Manages and disseminates timely, verifiable information about warnings, hazards, closures, and advisories for trail segments. | segment\_warnings, warning\_types\_master, warning\_severities\_master, warning\_source\_types\_master, workflow\_statuses\_master. | 2.1 | Master tables (warning\_types\_master v2.1, etc.) aligned to V2 patterns (id PK, code, display\_name (translatable), is\_active, audit columns by profiles.id, icons, sort order); segment\_warnings table (v2.1) with segment\_id (FK), warning\_type\_id (FK), is\_currently\_active (generated column), workflow\_status\_code (FK), and audit by auth.users.id; public\_active\_segment\_warnings\_view (v1.0) proposed for simplified public access to active warnings; orphan translation cleanup triggers. |

| 6. User Interaction Module | Capturing user votes on waypoints, managing user-submitted tips with moderation, and defining standardized, translatable categories for tips. | user\_waypoint\_votes, user\_waypoint\_short\_tips, tip\_categories\_master. | 2.0 | tip\_categories\_master (v2.1) updated to V2 patterns (category\_code PK, translatable default\_name/description, is\_active, audit); user\_waypoint\_votes (v2.1.1) with trigger update\_waypoint\_vote\_counts to denormalize counts onto public.waypoints; user\_waypoint\_short\_tips (v2.1.1) with is\_publicly\_visible generated column and detailed moderation fields; view\_tip\_categories\_localized for API support. |

| 7. Curated Itinerary Module | Defines, structures, categorizes, and manages the lifecycle of pre-defined pilgrimage itineraries and their associated lookup data. | curated\_itineraries, curated\_itinerary\_segments, itinerary\_categories\_master, seasons\_master, trail\_difficulty\_levels\_master, content\_statuses\_master. | 2.1 | Promotion of ENUMs (difficulty, status, categories, seasons) to full master tables (v2.1) with V2 patterns (code, name (translatable), is\_active, audit, icons, sort order); full V2 audit columns on all module tables; orphan translation cleanup triggers; curated\_itineraries (v3.1) with content\_status\_id FK (NOT NULL); new junction tables curated\_itinerary\_to\_category (v2.1) and curated\_itinerary\_to\_season (v2.1); localized views (v\_curated\_itineraries\_list\_localized, etc.). //

MinifistOK (USS Account for Via di Francesco Pilgrimage Platform database): Okay, sounds good. I'm ready when you are. Please ensure all the context files have been provided. I will be sure to pay close attention to the instructions on the output structure.

Project Overview: Via di Francesco Pilgrimage Platform Database (Updated May 18, 2025 -- V2.1)

================================================================================================

5. 🚀 Executive Summary

The Via di Francesco Pilgrimage Platform aims to be the definitive digital companion for pilgrims, offering comprehensive, reliable, and engaging information for planning and undertaking their journeys. This V2.1 update of the database overview reflects significant progress across all modules, emphasizing standardization towards V2 patterns, enhanced data integrity, and improved support for multilingual content and rich media.

Key advancements in this iteration include the consistent application of standardized audit columns (`created\_by\_profile\_id`, `updated\_by\_profile\_id`, `created\_at`, `updated\_at`) across almost all tables, refined lifecycle management for master data (`is\_active` flags) and transactional data (`deleted\_at` timestamps), a more robust and centralized internationalization (i18n) strategy leveraging the `public.translations` table with primary language (English) content stored directly in entity tables, and enhanced media management through dedicated linking tables and semantic roles. Furthermore, Row-Level Security (RLS) policies have been refined and are consistently designed to integrate with Supabase Auth and standardized helper functions. Many modules now also feature specialized views to optimize data retrieval for API endpoints, particularly for localized content.

6. 🎯 Refined Goals & Success Metrics

This V2.1 update reinforces the project's core goals by enhancing data quality, system maintainability, and the richness of information provided to users. Specific refinements and new objectives supported by these updates include:

- Enhanced Data Integrity & Auditability: Standardized audit columns and lifecycle flags across all modules significantly improve data tracking and governance capabilities.

- Improved Multilingual Support: Centralized translation management with primary language direct storage makes content more accessible and easier to manage globally.

- Richer Media Integration: New media linking strategies allow for more versatile and context-specific image and file associations (e.g., galleries, specific roles for articles).

- Streamlined API Performance: Introduction of localized and summary views aims to optimize data retrieval for common API queries.

- Increased Pilgrim Safety: The Dynamic Conditions module (Module 5) provides more structured and verifiable information about trail warnings and hazards.

- Better Pilgrim Planning: Detailed information for accommodations (Module 4a), attractions, food/water sources, shops/services (Module 4b), transport (Module 4c), and events (Module 4d) allows for more comprehensive journey planning.

- Enhanced User Engagement: Module 6 (User Interaction) enables waypoint voting and moderated tips, while Module 8 (Platform Content) supports rich articles and news.

7. 👥 Primary Personas & User Stories

The primary user roles remain consistent:

- Pilgrims: Planning and undertaking journeys, seeking information on routes, POIs, accommodations, conditions.

- Accommodation Hosts: Managing their listings and availability.

- Regional Content Managers: Updating local information, trail conditions, and POIs.

- Platform Administrators: Overseeing content, system integrity, and user management. Module updates provide richer data and better tools for all personas, for instance, through more detailed filtering for pilgrims or more structured content management for hosts and administrators.

8. 🗺️ Module Map (Snapshot Table)

| Module | Core Purpose | Key Tables | Version | Major Changes (V2.1 Rollup) |

| :-------------------------------------------------- | :------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- | :------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------ | :---------------------------------------- | :-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- |

| 1. User & Content Infrastructure | Manages users, roles (JWT sync), languages, media metadata, and platform-wide translations. | profiles, user\_roles\_master, languages\_master, media, translations. | 2.3 | Standardized audit columns (created\_by\_profile\_id, updated\_by\_profile\_id); icon\_identifier in master tables; primary language (English) content stored directly in main tables, others in public.translations; API returns base field in English & \*\_translations object; profiles.last\_activity\_at added; role synchronization to auth.users.raw\_app\_meta\_data.roles via triggers. |

| 2. Core Trail Hierarchy | Defines structure and characteristics of trails, routes, and segments, including master data for terrain and usage. | trails, routes, segments, route\_segments, terrain\_types\_master, usage\_types\_master, segment\_media. | 2.2 (API Aligned) | GPX file management shifted to media table (via media\_id); segment\_media enhanced with media\_role\_code and translatable captions/alt-text; surrogate id PKs for junction tables (trail\_regions, route\_segments); terrain\_types\_master & usage\_types\_master updated to full V2 spec (is\_active, audit columns, translatable names); all module tables now have full V2 audit columns; new localized views for API support (v\_trails\_detailed\_localized, etc.). |

| 3. Geographical Context | Defines and categorizes geographical areas (regions, provinces, towns) with multilingual support and auditability. | regions, provinces, towns, characteristic\_tags\_master, service\_tags\_master, town\_types\_master. | 2.2 (API-Aware) | Consistent V2 audit columns and lifecycle management (is\_active for master/provinces; deleted\_at & content\_visibility\_status for regions/towns); master tables (characteristic\_tags\_master, etc.) enhanced with is\_active, audit, translatable names, icons; array FK validation triggers (e.g., trigger\_validate\_characteristics\_tags); new region\_media & town\_media linking tables for galleries; localized views (v\_towns\_list\_localized, v\_regions\_list\_localized) for API efficiency. |

| 4. Waypoint Detail Modules (General) | Manages core POI ("waypoint") information, categorization, tagging, and content lifecycle. | waypoints, waypoint\_categories\_master, tags\_master, content\_statuses\_master. | 1.2 | Finalized core tables (waypoints v1.3, master tables v1.2/1.3) to V2 patterns (audit columns, is\_active for master data, deleted\_at for waypoints, i18n support via public.translations); localized views (v\_waypoint\_categories\_localized, v\_tags\_localized) introduced for efficient localized master data retrieval; array FK integrity trigger trigger\_check\_waypoint\_subcategory\_tags for waypoints.waypoint\_subcategory\_tag\_ids. |

| 4a. Waypoint - Accommodations | Manages detailed accommodation-specific information including types, amenities, rooms, pricing, and reviews. | accommodations, accommodation\_reviews, various \*\_master tables (e.g., accommodation\_types\_master), various accommodation\_\* junction tables. | 1.0 (New) | New module. Central accommodations table (v1.5); numerous specialized master tables (e.g., accommodation\_types\_master v1.4, amenities\_master v1.5) aligned with V2 patterns (audit, is\_active, translatable labels/descriptions, icons, sort order); junction tables (e.g., accommodation\_amenities v1.4) with translatable notes and audit columns; new accommodation\_reviews table (v1.0) with moderation workflow; localized views (e.g., v\_accommodation\_types\_localized) for all master tables; accommodations\_capacity\_summary\_view; specific translation cleanup triggers per table. |

| 4b. Waypoint - Attractions, Food/Water, Shops/Services | Enriches waypoints with specific details for attractions (incl. religious services), food/water sources, and shops/services. | attractions\_details, religious\_service\_schedules, food\_water\_sources\_details, shops\_and\_services\_details, numerous \*\_master tables, \*\_media linking tables. | 1.1 (New) | New module. Detail tables (attractions\_details v1.3.1, etc.) with translatable fields; numerous master tables (attraction\_types\_master v1.1, etc.) with V2 patterns (is\_active, audit, translatable labels, icons, sort order); dedicated \*\_media linking tables (e.g., attraction\_details\_media v1.0) for galleries with roles and translatable overrides; localized views (v\_waypoint\_attraction\_details\_localized v1.0, etc.) for efficient API data retrieval; "active check" triggers for all FKs to master tables; array FK integrity triggers; shops\_and\_services\_details.deleted\_at for independent soft delete. |

| 4c. Waypoint - Transportation | Manages public transportation stop information, types, facilities, and operational details. | transport\_stops\_details, transport\_stop\_types\_master, transport\_stop\_facilities\_master, view\_transport\_stops\_enriched. | 2.1 (New) | New module. Master tables (transport\_stop\_types\_master v2.1, transport\_stop\_facilities\_master v2.1) with V2 patterns (is\_active, full audit, translatable labels, icons, sort order); transport\_stops\_details table (v2.1) links to waypoints, includes translatable operator\_names\_text array, and stop\_facility\_ids array FK with validation trigger check\_transport\_stop\_facility\_ids; new view\_transport\_stops\_enriched (v2.1) for API efficiency; orphan translation cleanup triggers; standardized API translation model (primary lang field + translations object); transport\_stops\_details.deleted\_at added for soft delete. |

| 4d. Waypoint - Events | Manages detailed information about events associated with waypoints, including classification, timing, and scale. | events\_details, event\_types\_master, event\_recurrence\_frequencies\_master, event\_attendance\_scales\_master, event\_date\_certainty\_levels\_master. | 1.1 (New) | New module. Central events\_details table (v1.3); master tables (event\_types\_master v1.1, etc.) with V2 patterns (code, default\_name (translatable), is\_active, audit columns, sort order, icons); event\_theme\_or\_focus\_tag\_ids (array FK to tags\_master) with validation trigger (trigger\_validate\_event\_theme\_tags); orphan translation cleanup triggers for all module tables; events\_details.deleted\_at for soft deletion; full V2 audit columns on all module tables; primary language content in direct fields, other languages in public.translations. |

| 5. Dynamic Conditions | Manages and disseminates timely, verifiable information about warnings, hazards, closures, and advisories for trail segments. | segment\_warnings, warning\_types\_master, warning\_severities\_master, warning\_source\_types\_master, workflow\_statuses\_master. | 2.1 | Master tables (warning\_types\_master v2.1, etc.) aligned to V2 patterns (id PK, code, display\_name (translatable), is\_active, audit columns by profiles.id, icons, sort order); segment\_warnings table (v2.1) with segment\_id (FK), warning\_type\_id (FK), is\_currently\_active (generated column), workflow\_status\_code (FK), and audit by auth.users.id; public\_active\_segment\_warnings\_view (v1.0) proposed for simplified public access to active warnings; orphan translation cleanup triggers. |

| 6. User Interaction Module | Capturing user votes on waypoints, managing user-submitted tips with moderation, and defining standardized, translatable categories for tips. | user\_waypoint\_votes, user\_waypoint\_short\_tips, tip\_categories\_master. | 2.0 | tip\_categories\_master (v2.1) updated to V2 patterns (category\_code PK, translatable default\_name/description, is\_active, audit); user\_waypoint\_votes (v2.1.1) with trigger update\_waypoint\_vote\_counts to denormalize counts onto public.waypoints; user\_waypoint\_short\_tips (v2.1.1) with is\_publicly\_visible generated column and detailed moderation fields; view\_tip\_categories\_localized for API support. |

| 7. Curated Itinerary Module | Defines, structures, categorizes, and manages the lifecycle of pre-defined pilgrimage itineraries and their associated lookup data. | curated\_itineraries, curated\_itinerary\_segments, itinerary\_categories\_master, seasons\_master, trail\_difficulty\_levels\_master, content\_statuses\_master. | 2.1 | Promotion of ENUMs (difficulty, status, categories, seasons) to full master tables (v2.1) with V2 patterns (code, name (translatable), is\_active, audit, icons, sort order); full V2 audit columns on all module tables; orphan translation cleanup triggers; curated\_itineraries (v3.1) with content\_status\_id FK (NOT NULL); new junction tables curated\_itinerary\_to\_category (v2.1) and curated\_itinerary\_to\_season (v2.1); localized views (v\_curated\_itineraries\_list\_localized, etc.). \_When this user inputs documents from other files, they are uploaded to a shared workspace. This means that the overview document is not accessible to this model. I will update the information using my knowledge and the module overview documents provided.

9. 🏗️ System Architecture Updates

The V2.1 architecture reflects a mature, standardized approach to database design, consistently applied across modules:

- Centralized Translation Model: The `public.translations` table is the definitive source for all translated text beyond the primary reference language (English). Entity tables now store English text directly in their primary descriptive columns (e.g., `trails.name`, `waypoints.description`, `warning\_types\_master.display\_name`). API responses are designed to provide the base field (often resolved to English or requested language) and a corresponding `\*\_translations` object for other languages.

- Standardized Media Linking:

- Singular, critical semantic roles (e.g., `profiles.public\_avatar\_media\_id`, `waypoints.primary\_image\_media\_id`) often use direct Foreign Keys to `public.media.id`.

- For galleries or multiple media associations, new `[entity\_name]\_media` linking tables are being introduced (e.g., `trail\_media`, `segment\_media`, `region\_media`, `town\_media`, `attraction\_details\_media`, `food\_water\_source\_media`, `shop\_service\_media`, `article\_media`). These linking tables include `media\_role\_code` (FK to `public.media\_roles\_master`), `display\_order`, and translatable caption/alt-text overrides.

- GPX files are now consistently managed via `media\_id` Foreign Keys in `segments` and `routes` tables, pointing to the `public.media` table.

- Enhanced Row-Level Security (RLS):

- RLS policies are extensively used across all modules, with increasing standardization.

- Reliance on helper functions like `public.has\_role(TEXT)` or `public.has\_role\_on\_profile(UUID, TEXT)` (checking `auth.uid()` against `public.profiles.roles` or JWT claims) is standard.

- Synchronization of `public.profiles.roles` to `auth.users.raw\_app\_meta\_data.roles` for JWT inclusion is a key feature of Module 1.

- Data Integrity via Triggers:

- Standard `updated\_at` triggers are universally applied.

- Orphaned translation cleanup `AFTER DELETE` triggers are mandated for all tables with translatable content.

- Array Foreign Key integrity triggers (checking existence and `is\_active` status of master records) are implemented for columns like `waypoints.waypoint\_subcategory\_tag\_ids`, `events\_details.event\_theme\_or\_focus\_tag\_ids`, and `transport\_stops\_details.stop\_facility\_ids`. Similar "active check" triggers are implemented for non-array Foreign Keys to master tables.

- Promotion of ENUMs to Master Tables: Many former ENUM types have been promoted to full `\*\_master` tables with `is\_active` flags, audit columns, translatable labels, icons, and sort order (e.g., for waypoint categories, tags, content statuses, warning types, event types, transport types/facilities, itinerary categories, seasons, difficulty levels).

10. 📚 Data Governance & Cross‑Cutting Concerns

- Translations & i18n:

- A central `public.translations` table stores all translated content for fields designated as translatable across all modules.

- Primary reference language (English) content is stored directly in the main table columns.

- API responses generally include the base field (primary language or requested language) and a `\*\_translations` object for other available languages.

- `AFTER DELETE` triggers on parent tables ensure orphaned translations are removed from `public.translations`.

- Standard Audit Columns:

- All relevant tables include `created\_at`, `updated\_at` (auto-triggered), `created\_by\_profile\_id`, and `updated\_by\_profile\_id`.

- For `media`, `uploader\_profile\_id` serves as `created\_by\_profile\_id`.

- For `segment\_warnings`, user audit links to `auth.users.id`.

- Soft-Delete & Lifecycle Management:

- Transactional/user data (e.g., `waypoints`, `events\_details`, `articles`, `transport\_stops\_details`, `user\_waypoint\_votes`, `user\_waypoint\_short\_tips`, `curated\_itineraries` ) typically uses `deleted\_at TIMESTAMPTZ NULL`.

- Master/lookup data uniformly uses `is\_active BOOLEAN NOT NULL DEFAULT true`.

- Content status tables (e.g., `content\_statuses\_master`, `workflow\_statuses\_master` ) manage complex content lifecycles.

- Media Handling Strategy:

- A central `public.media` table stores metadata for original uploads and `image\_variants\_json` for derivatives. Backend processes generate variants.

- Direct FKs for singular roles (e.g., avatar, primary image) and `[entity\_name]\_media` linking tables for galleries/multiple roles (with `media\_role\_code` FK to `public.media\_roles\_master`).

- Access Control (RLS):

- RLS is extensively applied using Supabase Auth, JWT claims (from `public.profiles.roles` ), and helper functions like `public.has\_role(TEXT)`.

11. 📈 Performance & Scalability Notes

- Indexing: Comprehensive indexing strategies are defined per module, including:

- PKs, FKs, and unique `code` columns on master tables are indexed.

- GIST indexes for geospatial data (`segments.path\_geom`, `waypoints.geom` ).

- GIN indexes for array FKs (e.g., `waypoints.waypoint\_subcategory\_tag\_ids`, `transport\_stops\_details.stop\_facility\_ids`, `events\_details.event\_theme\_or\_focus\_tag\_ids` ) and FTS (planned).

- Composite indexes for filtering/sorting (e.g., on `is\_active, sort\_order` in master tables; `is\_currently\_active, workflow\_status\_code` in `segment\_warnings` ).

- Critical indexes on `public.translations` for lookup performance.

- Views for API Performance: Localized views (e.g., `v\_waypoint\_categories\_localized`, `v\_towns\_list\_localized` ), summary views (e.g., `accommodations\_capacity\_summary\_view` ), and enriched views (e.g., `view\_transport\_stops\_enriched`, `public\_active\_segment\_warnings\_view`, `published\_articles\_view` ) are increasingly used to denormalize data, pre-join tables, and aggregate translations, simplifying API logic and optimizing read performance.

- Database Functions for Complex GETs: Recommended for endpoints requiring significant aggregation and translation resolution (e.g., accommodation details, region/town details, article details ).

- Partitioning Strategies: `segments`, `media`, and `translations` tables are noted as V2+ candidates for partitioning if data volumes grow excessively. `segment\_warnings`, `events\_details`, and `accommodation\_reviews` are also potential candidates.

- Caching: Application-level caching is recommended for frequently accessed, relatively static master data (e.g., `languages\_master`, `user\_roles\_master`, and various `\*\_master` tables from Module 4, 5, 7). Materialized views are a future option for complex localized views if standard views face performance issues.

12. 🔐 Security & Compliance

- Authentication & Authorization: Supabase Auth is the authentication provider, leveraging JWTs. Application-level roles stored in `public.profiles.roles` are synchronized to JWT custom claims for use in RLS policies.

- Row-Level Security (RLS): Extensively applied across all tables in all modules, with policies tailored to roles (Public, Authenticated User, Owner, various Manager/Admin levels) and data status (e.g., `is\_active`, `content\_visibility\_status`, `deleted\_at` ).

- RLS Helper Functions: Standardized SQL helper functions (e.g., `public.has\_role(TEXT)`, `public.is\_platform\_admin()`, `public.user\_manages\_segment()` ) are crucial for implementing RLS policies. Secure implementation of `SECURITY DEFINER` functions is noted where necessary, with careful `search\_path` management.

- Data Integrity: Database-level constraints (`CHECK`, `FOREIGN KEY` with appropriate `ON DELETE` actions, `NOT NULL`), "active check" triggers for FKs to master tables, and array FK integrity triggers enhance data reliability.

- Compliance: No specific new compliance requirements (e.g., HIPAA, GDPR beyond standard data protection) were noted in the V2.1 module updates. General principles of data minimization and purpose limitation should be followed.

13. 🛣️ Roadmap & Next Steps

P0 (Immediate Blockers/Highest Priority):

- Global Auth & RLS Foundations:

- Implement and rigorously test all core RLS helper functions (`public.has\_role(TEXT)`, `public.has\_role\_on\_profile(UUID, TEXT)`, `is\_platform\_admin()`, `user\_manages\_segment()`, etc.), ensuring their security and performance, especially for `SECURITY DEFINER` contexts.

- Verify and test the `public.profiles.roles` to `auth.users.raw\_app\_meta\_data.roles` synchronization mechanism.

- Ensure `public.handle\_new\_user()` correctly initializes `profiles.roles` and `auth.users.raw\_app\_meta\_data.roles`.

- Translation Infrastructure:

- Ensure the global `public.cleanup\_related\_translations()` function is robust for all PK types (TEXT, INTEGER, UUID) and correctly implemented as `AFTER DELETE` triggers on ALL translatable tables across ALL modules.

- Array FK Integrity Triggers & Active Check FK Triggers: Implement and test all array FK integrity validation triggers (e.g., `check\_waypoint\_subcategory\_tags`, `check\_event\_theme\_tags\_exist`, `check\_transport\_stop\_facility\_ids` ) and general "active check" FK triggers ensuring master records are active.

P1 (Core V2.1 Implementation & Stabilization):

- Module DDL Execution: Systematically deploy all finalized DDL (tables, views, functions, triggers, indexes) for each module according to their specified build orders and V2.1/latest versions.

- Seed Data Population: Populate all `\*\_master` tables across all modules with agreed initial V2.1 data, including `icon\_identifier`, `sort\_order`, and correct audit column values (using a designated admin UUID).

- RLS Policy Implementation & Testing: Apply and rigorously test all defined RLS policies for all tables and views across all modules with diverse user roles.

- API Backend Logic & View Integration:

- Develop/refine API backend logic to utilize the new/updated localized views and database functions for data retrieval.

- Ensure consistent handling of the `lang` parameter and the API translation model (base field + `\*\_translations` object).

- Finalize `waypoints` Vote Count Integration: Add `up\_vote\_count` and `down\_vote\_count` to `public.waypoints`; ensure `public.update\_waypoint\_vote\_counts()` trigger is correct.

- Slug & Default Status Logic: Implement for `curated\_itineraries` and `articles`.

- JSON Schema Definition: Formally define, document, and validate schemas for JSONB fields like `media.image\_variants\_json` and `opening\_hours\_structured`.

P2 (Further Enhancements & Broader Scope):

- Backend Media Variant Generation: Develop and deploy the backend process for `media.image\_variants\_json` population.

- User Tip & Review Edit Policy/RLS: Finalize and implement for `user\_waypoint\_short\_tips` and `accommodation\_reviews`.

- Data Migration from Old ENUMs (if applicable).

- URL Validation Constraints: Implement DB-level `CHECK` constraints for all relevant URL fields.

- Review and Standardize `ON DELETE` Actions: For all FKs across the database.

- Consistent Lifecycle Management: Ensure consistent use of `is\_active` / `deleted\_at` / status fields across all tables.

- Define Population Logic for Audit Fields: Ensure robust application/admin tool logic for setting `created\_by\_profile\_id`, `updated\_by\_profile\_id`, especially for `\*\_master` tables and system actions.

- `profiles.last\_activity\_at` Update Mechanism: Define and implement.

- Define clear conventions for `icon\_identifier` usage and asset sources.

- Plan and Implement Triggers for `media.last\_linked\_or\_used\_at`.

- Full Specification of Waypoint Detail Extension Tables: Complete V2 review for `accommodations` (beyond v1.0 current spec), `attractions\_details` (beyond v1.3.1), `food\_water\_sources\_details` (beyond v1.3.1), etc.

- Editorial Workflow for `data\_last\_verified\_at`: (e.g., for transport stops, events ).

14. 📋 Appendix A -- Module Change Log

- Overall Project (V2.0 to V2.1): Comprehensive update reflecting standardization to V2 patterns across all modules. Key themes include consistent audit columns, `is\_active` flags for master data, refined `deleted\_at` usage, centralized `public.translations` strategy with primary language in-table, enhanced media linking with roles, new localized/summary views for API optimization, and more robust RLS policies with standardized helper functions. Array FK and "active check" FK integrity triggers widely implemented.

- Module 1: User & Content Infrastructure (to V2.3): Standardized audit columns (`created\_by\_profile\_id`, `updated\_by\_profile\_id`) and `icon\_identifier` added to master tables. Adopted "English in main table, other languages in `public.translations`" model with clear API output structure. Added `profiles.last\_activity\_at`. Implemented role synchronization from `profiles.roles` to `auth.users.raw\_app\_meta\_data.roles` for JWTs via triggers.

- Module 2: Core Trail Hierarchy (to V2.2 API Aligned): GPX file management fully transitioned to `public.media` via `media\_id` FKs. `segment\_media` table enhanced with `media\_role\_code` and translatable `caption`/`alt\_text`. Surrogate `id` PKs added to junction tables (`trail\_regions`, `route\_segments`). `terrain\_types\_master` and `usage\_types\_master` fully updated to V2 spec (`is\_active`, full audit columns, translatable names/descriptions). All module tables now include full standard V2 audit columns. New localized views (`v\_trails\_detailed\_localized`, `v\_routes\_detailed\_localized`, `v\_segments\_detailed\_localized`) introduced for API support.

- Module 3: Geographical Context (to V2.2 API-Aware): Consistent V2 audit columns and lifecycle management (`is\_active` for master tables & `provinces`; `deleted\_at` and `content\_visibility\_status` for `regions` & `towns`). Master tables (`characteristic\_tags\_master`, `service\_tags\_master`, `town\_types\_master`) enhanced with `is\_active`, audit columns, translatable names, icons. Array FK validation triggers (e.g., for `regions.characteristics\_tags`) added. New `region\_media` and `town\_media` linking tables for galleries. Localized views (`v\_towns\_list\_localized`, `v\_regions\_list\_localized`) for API efficiency.

- Module 4: Waypoint Detail Modules (General) (to V1.2): Finalized core tables `waypoints` (v1.3), `waypoint\_categories\_master` (v1.2), `tags\_master` (v1.3), `content\_statuses\_master` (v1.2) to V2 patterns (standard audit columns, `is\_active` for master data, `deleted\_at` for `waypoints`, i18n support via `public.translations` ). Localized views (`v\_waypoint\_categories\_localized`, `v\_tags\_localized`) introduced for efficient retrieval of translated master data. Array FK integrity trigger `trigger\_check\_waypoint\_subcategory\_tags` specified for `waypoints.waypoint\_subcategory\_tag\_ids`.

- Module 4a: Waypoint - Accommodations (to V1.0 - New Module): Introduced central `accommodations` table (v1.5); numerous specialized master tables (e.g., `accommodation\_types\_master` v1.4, `amenities\_master` v1.5 ) aligned with V2 patterns (audit, `is\_active`, translatable labels/descriptions, icons, sort order); junction tables (e.g., `accommodation\_amenities` v1.4 ) with translatable notes and audit columns; new `accommodation\_reviews` table (v1.0) with moderation workflow; localized views (e.g., `v\_accommodation\_types\_localized`) for all master tables; `accommodations\_capacity\_summary\_view`; specific translation cleanup triggers per table.

- Module 4b: Waypoint - Attractions, Food/Water, Shops/Services (to V1.1 - New Module): Introduced detail tables (`attractions\_details` v1.3.1, `religious\_service\_schedules` v1.3.1, `food\_water\_sources\_details` v1.3.1, `shops\_and\_services\_details` v1.3.1 ) with translatable fields; numerous master tables (`attraction\_types\_master` v1.1, etc.) with V2 patterns (`is\_active`, audit, translatable labels, icons, sort order); dedicated `\*\_media` linking tables (e.g., `attraction\_details\_media` v1.0 ) for galleries with roles and translatable overrides; localized views (`v\_waypoint\_attraction\_details\_localized` v1.0, etc.) for API data retrieval; "active check" triggers for all FKs to master tables; array FK integrity triggers; `shops\_and\_services\_details.deleted\_at` for independent soft delete.

- Module 4c: Waypoint - Transportation (to V2.1 - New Module): Introduced master tables (`transport\_stop\_types\_master` v2.1, `transport\_stop\_facilities\_master` v2.1 ) with V2 patterns (`is\_active`, full audit, translatable labels, icons, sort order). `transport\_stops\_details` table (v2.1) links to waypoints, includes translatable `operator\_names\_text` array, and `stop\_facility\_ids` array FK with validation trigger `check\_transport\_stop\_facility\_ids`. New `view\_transport\_stops\_enriched` (v2.1) for API efficiency. Orphan translation cleanup triggers and standardized API translation model (primary lang field + `translations` object) implemented. `transport\_stops\_details.deleted\_at` added for soft delete.

- Module 4d: Waypoint - Events (to V1.1 - New Module): Introduced central `events\_details` table (v1.3). Master tables (`event\_types\_master` v1.1, etc.) with V2 patterns (`code`, `default\_name` (translatable), `is\_active`, audit columns, sort order, icons). `event\_theme\_or\_focus\_tag\_ids` array FK (to `tags\_master.id`) with validation trigger (`trigger\_validate\_event\_theme\_tags`). Orphan translation cleanup triggers for all module tables. `events\_details.deleted\_at` for soft deletion. Full V2 audit columns on all module tables. Primary language content in direct fields, other languages in `public.translations`.

- Module 5: Dynamic Conditions (to V2.1): Master tables (`warning\_types\_master` v2.1, etc.) aligned to V2 patterns (`id` PK, `code`, `display\_name` (translatable), `is\_active`, audit columns by `profiles.id`, icons, sort order). `segment\_warnings` table (v2.1) with `segment\_id` (FK), `warning\_type\_id` (FK), `is\_currently\_active` (generated column), `workflow\_status\_code` (FK), and audit by `auth.users.id`. `public\_active\_segment\_warnings\_view` (v1.0) proposed for simplified public access to active warnings. Orphan translation cleanup triggers.

- Module 6: User Interaction (to V2.0): `tip\_categories\_master` (v2.1) updated to V2 patterns (`category\_code` PK, translatable `default\_name`/`description`, `is\_active`, audit). `user\_waypoint\_votes` (v2.1.1) with trigger `update\_waypoint\_vote\_counts` to denormalize counts onto `public.waypoints`. `user\_waypoint\_short\_tips` (v2.1.1) with `is\_publicly\_visible` generated column and detailed moderation fields. `view\_tip\_categories\_localized` for API support.

- Module 7: Curated Itinerary (to V2.1): Promotion of ENUMs (difficulty, status, categories, seasons) to full master tables (v2.1) with V2 patterns (`code`, `name` (translatable), `is\_active`, audit, icons, sort order). Full V2 audit columns on all module tables. Orphan translation cleanup triggers. `curated\_itineraries` (v3.1) with `content\_status\_id` FK (NOT NULL). New junction tables `curated\_itinerary\_to\_category` (v2.1) and `curated\_itinerary\_to\_season` (v2.1). Localized views (`v\_curated\_itineraries\_list\_localized`, etc.).

- Module 8: Platform Content (to V1.0 - New Module): New module. `articles` table (v1.0) for news/blog posts with translatable text, featured image, associations, audit columns, status, and soft delete (`deleted\_at`). `media\_roles\_master` (v1.0) defines semantic roles for media within articles, with translatable names, `is\_active` flag, and audit columns. `article\_media` (v1.0) linking table connects articles to `public.media` items, assigning a role, display order, and allowing translatable caption/alt-text overrides, with audit columns. Views `published\_articles\_view` and `article\_media\_details\_view` for denormalized access. Translation cleanup triggers on all module tables.

15. 📋 Appendix B -- Glossary / ENUM Registry

- General Trend: Many former database `ENUM` types have been promoted to dedicated `\*\_master` lookup tables across modules (e.g., Waypoint Categories, Tags, Content Statuses, Warning Types, Event Types, Transportation Types/Facilities, Itinerary Categories, Seasons, Difficulty Levels ). These master tables typically include `id` (PK), `code` (unique text identifier), `label`/`name` (translatable, English primary), `description` (translatable, English primary, optional), `icon\_identifier` (optional), `sort\_order`, `is\_active` (boolean for lifecycle), and full audit columns (`created\_at`, `updated\_at`, `created\_by\_profile\_id`, `updated\_by\_profile\_id`).

- Key Remaining Global ENUMs (examples from modules):

- `public.units\_preference\_enum` ('metric', 'imperial')

- `public.pilgrim\_experience\_enum` ('novice\_first\_pilgrimage', ...)

- `public.user\_account\_status\_enum` ('active', 'suspended\_by\_admin', ...)

- `public.media\_asset\_type\_enum` ('image', 'gpx\_file', ...)

- `public.media\_licence\_enum` ('all\_rights\_reserved', 'cc\_by', ...)

- `public.media\_status\_enum` ('pending\_review', 'published\_approved', ...)

- `public.trail\_difficulty\_enum` (used by Core Trail Hierarchy & Curated Itineraries - though Module 7 now uses `trail\_difficulty\_levels\_master`)

- `public.trail\_operational\_status\_enum`

- `public.content\_visibility\_status\_enum` (used by multiple modules for main entities like trails, routes, segments, regions, towns, articles - though Module 7 now uses `content\_statuses\_master`)

- `public.route\_category\_enum` ('main\_section', 'official\_variant', ...)

- `public.segment\_sun\_exposure\_enum` ('mostly\_shaded', 'mostly\_exposed', ...)

- `public.segment\_travel\_direction\_enum` ('bidirectional', 'northbound\_only', ...)

- `public.weekday\_enum` (used by religious service schedules, etc.)

- `public.vote\_type\_enum` ('up', 'down') (used by user interactions, accommodation reviews)

- `public.content\_moderation\_status\_enum` ('pending\_approval', 'approved\_visible', ...) (used by user interactions, accommodation reviews)

- `is\_active` (BOOLEAN): Standard lifecycle flag for master/lookup data. `TRUE` means the entry is current and can be referenced. `FALSE` means it's retired/historical.

- `deleted\_at` (TIMESTAMPTZ): Standard flag for soft-deleting transactional data. `NULL` means active. A timestamp indicates when it was soft-deleted.

- `icon\_identifier` (TEXT): A textual reference (e.g., CSS class name, SVG file name, material icon code) used by the frontend to display an appropriate icon. Introduced in many `\*\_master` tables.

- Primary Language / Reference Language: English. Stored directly in entity table columns.

- `code` (TEXT): A stable, machine-readable, unique identifier (often `snake\_case`) used in master tables (e.g., `user\_roles\_master.role\_code`, `warning\_types\_master.code`, `media\_roles\_master.role\_code` ). Essential for system integrations and referencing specific master data entries programmatically.