Gami Protocol: Universal Wallet MVP Technical Specification

Goal: Develop the Minimal Viable Product (MVP) for the Gami Protocol Universal Wallet (Phase 1). This wallet serves as the user's non-custodial Cross-Web Identity, Quest Hub, and Airdrop Gateway.
Design Reference: Contra Wireframe Kit (Figma) - Simple, modern, and mobile-first approach.
Key Constraints:

1. **No $GAMI Balance Display:** The wallet should track XP and claimable Gami Points, but **not** the liquid $GAMI token balance yet (Focus is on utility/progression, not speculation).
2. **XP Ring:** The user profile must feature a dynamic circular XP progression ring.
3. **Airdrop CTA:** A highly visible sign-up mechanism for the upcoming $GAMI Airdrop is mandatory.

1. Design & UI/UX Principles

The design should prioritize simplicity, mobile responsiveness, and gamified engagement.

Component	Figma Kit Reference (Adaptation)	Design Goal
Global Theme	Use "Light Mode" with high contrast and elements adapted from the Gami purple/blue palette (e.g., #6E3CFB for main actions).	Clean, modern, and visually linked to the Gami brand.
Main Layout	Use a standard **Mobile Screen Layout** with a header, scrollable content area, and a bottom navigation bar.	Ensure primary profile and quest actions are accessible on mobile.
Cards/Containers	Use "Card" or "Box" components for Quest Listings and Asset Aggregator. Apply subtle rounded corners and shadows.	Segment information clearly and cleanly.
Buttons	Use "Primary Button" (e.g., the purple fill) for high-priority actions like **"Sign Up for Airdrop"** and **"Claim Reward"**.	Guide the user flow efficiently.

2. Feature Specification: Universal Identity & Profile

This is the central view where the user’s aggregated status and progress are displayed.

2.1 Technical Requirement: Profile & XP Ring

Element	Data Source (MCP Core)	Technical Implementation
Avatar & Level	user_profile.avatar_url, user_profile.current_level	**Avatar:** Standard image display (default placeholder if none). **Level:** Large, centered number.
XP Progression Ring	user_profile.current_xp, user_profile.xp_to_next_level	**Critical Implementation:** Use an **SVG Circle** or **CSS Conic Gradient** to create a progress ring around the avatar. The percentage fill is calculated as: (current_xp / xp_to_next_level) * 100.
Username/Wallet ID	user_profile.username (or truncated wallet_address if no username)	Display the user's primary identifier. Include a **Copy-to-Clipboard** button next to the ID.
Status Score	user_profile.status_score (Single aggregated number)	Display a general score reflecting total engagement/reputation.

UI Layout Sketch (Mobile Top Section):

```
[Header: Gami Logo]
-----------------------------
|         [User Level]        |
|       [XP PROGRESS RING]    |
|       [Avatar Image]        |
|                             |
|    @Username  [Copy Icon]   |
|  Status Score: 850 / VIP 1  |
-----------------------------
```

2.2 Technical Requirement: Asset Aggregator

This section lists non-monetary assets that define the user's identity and loyalty.

Element	Data Source (MCP Core)	Display Logic
Held XP	user_profile.current_xp (Raw total)	Display in a large font with the "XP" label (e.g., **12,450 XP**).
Claimable Gami Points (GP)	user_profile.claimable_gami_points	Display as a pending reward. Include a **Claim GP** button that triggers the GP conversion logic (which will be an API call to the MCP Core in Phase 1).
SBTs / Achievement NFTs	user_assets.sbt_list (Array of objects with name, icon, chain_id)	Display as a horizontal scroll or grid of small cards/badges. Show key achievements (e.g.,

Element	Data Source (MCP Core)	Display Logic
		"Verified Founder," "Level 10 Gamer").

3. Feature Specification: Quest Hub & Community

This view drives user engagement and provides pathways for earning XP.

3.1 Technical Requirement: Quest Feed

Element	Data Source (AI Agent Layer)	UI/UX Requirements
Quest Card	quest.id, quest.title, quest.reward_xp, quest.category	Use a "List Item" or "Card" component from the Figma kit.
Progress Indicator	quest.current_progress_steps, quest.total_steps	Show progress clearly, e.g., "3/5 Steps Completed."
Action Button	N/A (Links to partner app/API)	Label: **"Continue Quest"**. Tapping triggers a deep-link or a modal with instructions.

3.2 Technical Requirement: Airdrop Sign-up

This is a mandatory, persistent Call-to-Action (CTA).

Element	Data Source	Technical Requirement
Airdrop Banner/Modal	Local/Static content	**Highly Visible:** Implement a non-dismissible card or a full-screen modal the first time a user logs in.
Sign Up Button	N/A	**Primary Action:** Label it **"Secure Your Airdrop Slot"**. On click, it performs the following: 1. Displays the user's Public Wallet Address. 2. Sends the wallet address and user ID to the backend **Airdrop Waitlist API**. 3. Changes the button state to **"Airdrop Slot Secured!"**.

3.3 Technical Requirement: Data Consent & Community

Element	Data Source	Technical Requirement
Data Consent Switch	user_profile.data_consent_status	Use a standard Toggle Switch (On/Off). Clearly explain that enabling consent (e.g., sharing fitness data) unlocks specific community quests and rewards.

4. Technical Architecture & Integration

All front-end development must be based on secure, asynchronous calls to the backend APIs.

4.1 Authentication Flow

1. **Onboarding:** User selects Social Login (e.g., Google).
2. **Account Abstraction Layer:** Backend receives OAuth token, creates a linked non-custodial wallet via Account Abstraction (ERC-4337), and securely provisions the key.
3. **Client-Side:** The front-end receives a session token and the user's **Universal Wallet Address**. This address is used for all subsequent API requests.

4.2 Data Fetching (CRUD)

All front-end data is sourced from the Gami Protocol API Gateway, which interfaces with the MCP Core Services.

Feature	Endpoint (Example)	Description
Profile & XP	/v1/users/me/profile	GET request: Fetches current_level, current_xp, xp_to_next_level, and status_score.
Quests	/v1/users/me/quests/active	GET request: Fetches list of active quests, progress, and rewards (powered by AI Agent Layer).
Assets	/v1/users/me/assets	GET request: Fetches claimable_gami_points and list of sbt_list.
Airdrop Sign-up	/v1/airdrop/register	POST request: Submits user_id and wallet_address to the Airdrop Waitlist.
Claim Gami Points	/v1/rewards/claim_gp	POST request: Triggers the conversion logic in the MCP Core for the claimable GP.

5. Next Steps

1. **Wireframing:** Create the three core screens (Profile/Assets, Quests, Airdrop/Settings) using the Contra Wireframe Kit components.
2. **Front-End Build:** Implement the UI using React/TailwindCSS (or similar mobile-first stack), ensuring the XP Progression Ring is a smooth, animated element.
3. **API Integration:** Connect all features to the mock API endpoints listed above, focusing on robust error handling and loading states.