

Sovereign Space Builder

Full System & Implementation

0. Front Matter

[0.1 Executive Summary](#)

[0.2 Design Principles & Ethos](#)

[0.3 Scope & Non-Goals](#)

[0.4 Audience & Roles](#)

[0.5 Definitions, Abbreviations, Glossary](#)

1. Vision & Use Cases

[1.1 Narrative Vision \(Home → Planetary Ark\)](#)

[1.2 Primary Use Cases](#)

[1.3 Personas](#)

[1.4 Success Metrics](#)

2. System Overview

[2.1 High-Level Architecture \(Client, Edge, Mesh, Cloud-optional\)](#)

[2.2 Core Subsystems](#)

[2.3 Data Flow & Event Bus](#)

[2.4 Trust & Consent Model](#)

3. Game Design & UX

[3.1 Player Loop \(Spawn→Scan, Collect→Design, Sim→Do, Review→Share\)](#)

[3.2 Memory-Palace UX](#)

[3.3 Progression & Rewards](#)

[3.4 Accessibility & Low-Tech Modes](#)

[3.5 Internationalisation & Cultural Modes](#)

4. Spatial Mapping & Digital Twin

[4.1 Onboarding & Scale \(AR, manual, CAD/GeoJSON import\)](#)

[4.2 Anchors & Zones \(Permaculture Z0–Z5, safety zones\)](#)

[4.3 Geometry, Layers & Constraints \(footprints, clearances\)](#)

[4.4 Media Anchoring \(photos, PDFs, CAD, clippings\)](#)

[4.5 Twin Storage \(scene graph, CRDT fields, diffs\)](#)

5. Knowledge Deck (Library-as-Inventory)

[5.1 Card Types \(photo, pdf, bookPage, map, clip, note\)](#)

[5.2 OCR/Tagging/Linking \(auto-suggest, cosine similarity\)](#)

[5.3 Brief Synthesis \(multi-card → recipe/guide\)](#)

[5.4 Provenance & Citations](#)

[5.5 Deck Score & Unlocks](#)

[6. Build System \(Design as Crafting\)](#)

[6.1 Modules Catalog \(beds, towers, aquaponics, compost, water, energy\)](#)

[6.2 Feasibility Engine \(sun/water/power/skill checks\)](#)

[6.3 Companion Planting Matrix & Conflicts](#)

[6.4 IO Graphs \(water, nutrients, compost, harvest routes\)](#)

[6.5 BOM/Costing \(DIY vs salvage, labour/robot time\)](#)

[6.6 Templates & Patterns \(Bioregion packs, meta-patterns\)](#)

[7. Aeroponics/Hydroponics Towers](#)

[7.1 Module Types \(Tower.Aero, Tower.Hydro, Nursery.DWC, SproutRack, Microgreens\)](#)

[7.2 Sensors & Control Loops \(EC, pH, temp, ORP, pressure\)](#)

[7.3 Nutrient Recipes & Micronutrient Flushes](#)

[7.4 CIP/Sanitation & HACCP Logs](#)

[7.5 Failure Modes & Self-Healing](#)

[7.6 UI: Towers Palette, Nutrient Curves, Sanitation Calendar](#)

[8. Simulation & Scenarios](#)

[8.1 Season Engine \(weekly ticks: growth, yield, labour\)](#)

[8.2 Weather & Climate Profiles \(local, heatwaves, drought, cyclone\)](#)

[8.3 Pests, Disease & Biodiversity Index](#)

[8.4 Resource Budgets \(water, energy, nutrients\)](#)

[8.5 Shock Scenarios & Resilience Scores](#)

[8.6 Micro-games \(pollinate, prune, predator release\)](#)

[9. Missions & Operations](#)

[9.1 Mission Generator \(deltas → tasks\)](#)

[9.2 Kanban & Calendar Views](#)

[9.3 Proof-of-Work \(photos, logs\) → Twin Updates](#)

[9.4 Ops Calendar \(sow/transplant/harvest/seed-save\)](#)

[9.5 Disaster Playbooks](#)

[9.6 Review & Export \(one-pagers, checklists\)](#)

[10. Robotics & Automation](#)

[10.1 Robot Catalog](#)

[10.2 Capabilities Matrix](#)

[10.3 Tasks, Routines & Recovery Graphs](#)

[10.4 Fleet Manager \(assignment, energy/water arbitration, locks\)](#)

[10.5 Zones, Policies, HRI Scripts \(Welcome Home, Handover\)](#)

[10.6 Safety, Compliance & Black-Box Logging](#)

[11. Off-World Modes \(Moon/Mars → Earth Welcome\)](#)

[11.1 Commissioning Timeline \(T-30 to launch\)](#)

[11.2 Autonomy Modes \(Assist, Supervised, Scheduled, Adaptive, Resilience\)](#)

- [11.3 Community Mode \(surplus routing, verification, ethics\)](#)
- [11.4 Welcome Prep \(–14d\), Arrival Day Protocols](#)
- [11.5 Latency & Tele-op Considerations](#)
- [12. Fractal Ark \(Decentralised Commons\)](#)
 - [12.1 Topology \(L0 Home, L1 Mesh, L2 Bioregion, L3 Planetary Commons\)](#)
 - [12.2 Schemas & Standards \(Darwin Core, MlXS, OSM, Schema.org\)](#)
 - [12.3 Identity & Consent \(DID/VC, access tiers, revocation\)](#)
 - [12.4 Content Addressing & CRDT Merge \(bundles, tombstones\)](#)
 - [12.5 Reputation & Attestations \(claim-level scoring\)](#)
 - [12.6 Halls of Wisdom Workflow \(submit → review → custodial check → pattern\)](#)
 - [12.7 Cultural & Indigenous Data Sovereignty](#)
- [13. Data Model & APIs](#)
 - [13.1 Core Objects \(Space, Anchor, Module, Plant, Card, Scenario, Mission, SeedBank\)](#)
 - [13.2 Robotics Objects \(Robot, Capability, Task, Routine, Policy, FleetManager, HRI\)](#)
 - [13.3 Towers Objects \(Tower, ResMix, DoseBlock, ValveManifold, HaccpLog, Recipe\)](#)
 - [13.4 Ark Objects \(Taxon, Place, Observation, SeedLot, Practice, SeasonPack, Reputation, AccessPolicy\)](#)
 - [13.5 Event Model \(publish, attest, adopt, revoke, merge\)](#)
 - [13.6 API Endpoints \(local, mesh, gateway\), Webhooks, Schema Validation](#)
 - [13.7 Storage \(content-addressed blobs, indices, encryption at rest\)](#)
- [14. Security, Privacy, Safety](#)
 - [14.1 Threat Model \(device loss, data leak, supply-chain\)](#)
 - [14.2 Encryption, Keys & Backups \(local custody, social recovery\)](#)
 - [14.3 Location Fuzzing & Red-list Shielding](#)
 - [14.4 Safety Interlocks \(zones, tools, pets/children, biosecurity\)](#)
 - [14.5 Audit & Traceability \(HACCP, ops logs, black-box\)](#)
- [15. Compliance & Ethics](#)
 - [15.1 Food Safety & HACCP](#)
 - [15.2 Electrical/Water/Building Codes \(regional variance, checklists\)](#)
 - [15.3 Robotics Safety \(ISO/IEC refs\)](#)
 - [15.4 Accessibility Standards](#)
 - [15.5 Cultural Protocols & Custodial Licences](#)
- [16. Implementation Plan](#)
 - [16.1 MVP Definition \(must-haves for first play\)](#)
 - [16.2 Phased Roadmap](#)
 - [16.3 Milestones & Exit Criteria](#)
 - [16.4 Team Roles & RACI](#)
 - [16.5 Budget & Bill of Materials \(DIY vs off-the-shelf vs custom\)](#)
 - [16.6 Risk Register & Mitigations](#)

17. DevOps & Deployment

- 17.1 Platforms (mobile, desktop, AR/VR)
- 17.2 Offline-First Architecture (sync, conflict, retries)
- 17.3 Mesh/Gateway Nodes (home server, neighbourhood hub)
- 17.4 Telemetry & Privacy-Preserving Analytics
- 17.5 CI/CD, Release Channels, Feature Flags
- 17.6 Backup/Restore & Disaster Recovery

18. Testing & Validation

- 18.1 Unit & Integration Tests (sim, controls, merge)
- 18.2 Field Trials (homes, schools, farms)
- 18.3 Robotics Dry-Runs & Safety Drills
- 18.4 Nutrient/Tower Bench Tests (EC/pH, CIP)
- 18.5 Ark Merge & Revocation Drills
- 18.6 Usability & Accessibility Testing

19. Education & Community

- 19.1 Tutorials, Quests, Curriculum Packs (schools, TAFEs, makerspaces)
- 19.2 Mentor/Guild Programs & Bounties
- 19.3 Seed Library & Tool Library Playbooks
- 19.4 Events & Challenges (Season Packs, Resilience Cups)
- 19.5 Support & Moderation

20. Economics & Incentives

- 20.1 C-Hours Timebank & Local Credits
- 20.2 Surplus Routing & Barter
- 20.3 Optional Web3 (blueprint minting, royalties, guardrails)
- 20.4 Sustainability & Funding Models

21. Appendices

- A. Companion Planting Matrix (starter)
- B. Nutrient Recipes & EC/pH Targets (leafy, fruiting, microgreens)
- C. Capability Matrix (robots × tasks × tools)
- D. Safety Checklists (kitchen, towers, myco, electrical, water)
- E. Seed Saving Protocols & Viability Tables
- F. Data Schemas (full JSON)
- G. API Reference (REST/gRPC/Events)
- H. File Formats & Interop (CAD/GeoJSON/OSM)
- I. Printables (offline cards, QR links, one-pagers)
- J. Glossary of Indigenous & Local Custodial Terms
- K. Bibliography & Source Attribution

Works cited

0. Front Matter

0.1 Executive Summary

This document outlines the vision, architecture, and implementation plan for the Sovereign Space Builder, a gamified digital twin system designed for personal and environmental regeneration. The system begins as a standalone, offline-first application (a "Sovereign Node") that allows an individual to map their self and their immediate environment ("place"). It then scales fractally, connecting with nearby nodes to form community meshes and eventually contributing to a decentralized, planetary-scale knowledge commons—the "Fractal Ark."

The core of the system is a dual digital twin: a Human Digital Twin (HDT) modeling the user's health and a corresponding Environmental Digital Twin (EDT) modeling their living space. Through a gamified loop rooted in permaculture principles, users are guided to observe, design, simulate, and build regenerative systems for their body and their environment.

Technologically, the system is built on a foundation of decentralized principles: peer-to-peer networking, offline-first data synchronization using CRDTs, and user-owned identity via Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs). Governance is fractal, with DAOs operating at the neighborhood, bioregion, and planetary levels. This architecture ensures data sovereignty, resilience, and scalability, creating a tool that is not only powerful but also ethical and community-owned by design.

The vision extends from personal well-being in a home garden to advanced applications in automated agriculture, robotics, and off-world habitats, all contributing to a shared, verifiable, and permanent library of life and regenerative practice.

0.2 Design Principles & Ethos

The system is guided by a non-negotiable ethos of **Sovereignty and Joyful, Responsible Abundance**. This ethos is expressed through the following core design

principles:

- **Sovereignty (Self-Rule):** Every user owns and controls their data, identity, and the physical systems they build. The architecture is designed to empower the individual and the community, not to extract from them. This aligns with the permaculture ethic of **People Care**.¹
- **Regeneration (Active Healing):** The system's goal is not merely to sustain but to actively improve the health and vitality of both the user and their ecosystem. It is a shift from "doing less harm" to becoming an agent of positive change.² This embodies the permaculture ethic of **Earth Care**.¹
- **Fractal Scaling (Pattern Replication):** The system's logic is designed to work at every scale, from a single room to the entire planet. The patterns of data, governance, and interaction at the individual level are replicated at the community and bioregional levels, ensuring coherence and resilience.
- **Decentralization by Default:** The system avoids single points of failure and control. Data, computation, and governance are distributed across the network of users, creating a resilient, peer-to-peer architecture.
- **Offline-First:** The application must be fully functional without an internet connection. Data is stored locally, and synchronization with the network happens opportunistically. This ensures utility in remote, disconnected, or disaster-prone environments.
- **Data Dignity:** The system respects the user's right to privacy and consent. Data sharing is explicit, purpose-bound, and revocable. It incorporates principles of Indigenous data sovereignty, allowing for local guardianship and tiered access to knowledge.⁴
- **Interoperability & Openness:** The system is built on open standards and schemas to prevent vendor lock-in and encourage a rich ecosystem of compatible tools and services.
- **Joyful, Responsible Abundance:** The user experience is designed to be engaging, empowering, and fun. It celebrates the abundance that comes from regenerative practice while upholding the permaculture ethic of **Fair Share**—setting limits and redistributing surplus for the good of the commons.¹

0.3 Scope & Non-Goals

In Scope:

- A cross-platform application (mobile, desktop) for individual users.
- Tools for mapping physical spaces (manual, AR-assisted).
- A digital twin system for modeling personal health (HDT) and environmental systems (EDT).
- A gamified user experience based on a permaculture design loop.
- A catalog of modular building systems (e.g., hydroponic towers, composters).
- A simulation engine for modeling growth, resource use, and system resilience.
- An offline-first, peer-to-peer data synchronization mechanism.
- A decentralized identity and verifiable claims system.
- A framework for fractal, multi-level DAO-based governance.
- Integration points for a future robotics and automation layer.
- Specialized modes for off-world habitat design and operation.

Non-Goals:

- **Centralized Data Hosting:** The system will not operate a central cloud server for user data storage. All primary data resides on user devices.
- **Manufacturing Physical Hardware:** The platform will provide open-source blueprints and control software for modules (e.g., towers, robots) but will not manufacture the hardware itself.
- **Token-as-Speculation:** While the system uses internal tokens for governance and incentives (C-Hours), it is not designed as a financial speculation platform.
- **Replacing Professional Medical Advice:** The Human Digital Twin is a tool for personal insight and well-being, not a diagnostic medical device. It is not a substitute for professional medical care.
- **Top-Down Control:** The Planetary Stewards (L3) do not have control over lower-level DAOs; their role is to maintain standards and protocols for interoperability.

0.4 Audience & Roles

- **Player:** The primary user. An individual using the system to map their space, track their well-being, and build regenerative systems. They interact with the game loop, complete missions, and contribute data to the Ark.
- **Builder:** A user who fabricates, installs, or maintains physical modules (towers, composters, robots) based on the system's open-source designs.
- **Developer:** An engineer or designer who contributes to the open-source codebase of the application, develops new modules, or builds third-party tools

that integrate with the system's APIs.

- **Researcher:** A scientist or academic who uses anonymized, aggregated data from the Fractal Ark (with appropriate permissions from the governing DAOs) to study ecological patterns, climate resilience, or community health.
- **Custodian:** An individual or group (e.g., an Indigenous community, a bioregional council) entrusted with the guardianship of specific knowledge or data within the Ark. They define and enforce access policies for sensitive information.

0.5 Definitions, Abbreviations, Glossary

- **Ark:** The decentralized, planetary-scale knowledge commons containing all user-contributed data.
- **CID:** Content Identifier. A unique hash derived from the content of a piece of data, used in systems like IPFS.
- **CRDT:** Conflict-free Replicated Data Type. A data structure that allows for concurrent edits on multiple devices to be merged automatically without conflicts.
- **DAO:** Decentralized Autonomous Organization. A member-owned community without centralized leadership, governed by rules encoded on a blockchain.
- **DID:** Decentralized Identifier. A new type of identifier that enables verifiable, decentralized digital identity, controlled by the user.
- **EDT:** Environmental Digital Twin. A virtual replica of the user's physical environment.
- **HDT:** Human Digital Twin. A virtual replica of the user's physical, physiological, and psychological state.⁵
- **IPFS:** InterPlanetary File System. A peer-to-peer protocol for storing and sharing content-addressed data.
- **L0, L1, L2, L3:** Levels of the Fractal Topology (Home, Neighbour, Bioregion, Planetary).
- **Permaculture:** A design philosophy for creating sustainable human environments by mimicking patterns found in nature.¹
- **VC:** Verifiable Credential. A tamper-proof digital credential that can be cryptographically verified.

1. Vision & Use Cases

1.1 Narrative Vision (Home → Planetary Ark)

The journey begins with a single person, Alex, in their small urban apartment. Feeling disconnected, Alex uses the **Sovereign Space Builder** app to map their balcony. The app guides them through a "Scout" mission, using their phone's camera to identify the few hardy plants growing there. This is **L0: The Home Cell**.

Alex then builds their Human Digital Twin, connecting their smartwatch and answering gamified psychological surveys. The system reveals a link between their afternoon energy slumps and the poor air quality near their window. A "Mission" is generated: "Build a Green Lung." Alex uses the **Build System** to design a small hydroponic tower, simulating its growth in the **Sim** environment before ordering the parts.

As Alex's tower thrives, they start generating a small surplus of basil. The app alerts them to the **L1: Neighbour Mesh**. They connect with a neighbor, Maria, and trade their basil for some of her sourdough starter. The mesh network shares anonymized data—not their personal details, but the *learning* that this basil variety grows well in their building's microclimate.

Months later, their entire building has formed a mesh. Their collective data on water usage, pest sightings, and successful plant guilds flows up as an anonymized summary to the **L2: Bioregion Guild**. This guild, covering their entire watershed, uses this data to refine its "Pattern Packs"—templates for resilient urban gardens tuned to their specific climate.

Years pass. Alex's single tower has become a network of thousands. The data from their bioregion, and hundreds of others, has been verified and added to the **L3: Planetary Commons**. This is the **Fractal Ark**: a living, breathing, user-built library of Earth's biology and regenerative know-how. A researcher in another hemisphere, designing a mission for a Martian habitat, downloads a "Drought-Resistant Guild" pattern from the Ark—a pattern whose wisdom began with Alex's single basil plant. From a single room to another world, the system has scaled, sharing wisdom without sacrificing sovereignty.

1.2 Primary Use Cases

- **Room:** A student in a dorm uses the system to build a microgreens rack and a small mushroom fruiting chamber, optimizing their nutrition and air quality in a tiny space.
- **Yard:** A suburban family transforms their lawn into a permaculture food forest, using the Sim engine to plan guilds and the Ark to identify native pollinator-friendly plants. They use the Neighbour Mesh to share surplus produce and organize tool-sharing.
- **Farm:** A small-scale farmer uses the system for advanced crop planning, soil health monitoring, and integrated pest management. They use the robotics layer to automate planting and harvesting, and contribute detailed yield data to the Bioregion Guild.
- **Off-World Base:** An astronaut on a lunar base uses the system to manage a closed-loop life support system. They run advanced shock scenarios in the Sim engine ("What if the water reclaimer fails?") and use the robotics layer for autonomous greenhouse operations, all while drawing on the Ark's vast library of terrestrial agricultural knowledge.

1.3 Personas

- **The Prepper (Zara):** Zara is focused on resilience and self-sufficiency. She uses the system to maximize food and medicine production in her off-grid homestead. She values the offline-first architecture and the disaster playbook features.
- **The Elder (Kenji):** Kenji is a keeper of traditional ecological knowledge. He uses the system as a **Custodian**, mapping culturally significant plants and practices. He uses the tiered access controls to ensure this knowledge is shared appropriately, creating "in-place only" digital licenses.
- **The Educator (Maria):** Maria is a high school science teacher. She uses the system as a curriculum tool, creating "Biome Codex" quests for her students to map the biodiversity of the school grounds and local park.
- **The Astronaut (Dr. Aris):** Aris is the life-support systems specialist on a Mars mission. He is the ultimate power user, relying on the Twin, Sim, and Robotics subsystems to ensure the survival of his crew. He is also a key contributor to the

Ark, sharing novel data on extremophile agriculture.

- **The Community Gardener (Leo):** Leo manages a shared garden plot in a dense city. He uses the L1 Neighbour Mesh DAO to coordinate planting schedules, manage the tool library, and route surplus food to a local shelter.

1.4 Success Metrics

- **Food %:** Percentage of a user's or community's caloric needs met by their own production.
 - **Medicine %:** Percentage of basic first-aid and wellness needs met by homegrown medicinal plants.
 - **Resilience %:** A composite score from the Sim engine, measuring the system's ability to withstand shocks (e.g., drought, power outage, pest outbreak) and maintain yield.
 - **Learning Rate:** The speed at which a user completes "Codex" pages and masters new skills (e.g., composting, seed saving).
 - **Sharing Index:** A measure of a user's or mesh's contribution to the commons, tracked via C-Hours earned and surplus routed.
 - **Biodiversity Score:** A calculated index based on the number and diversity of species (plant, animal, insect) cataloged in the user's EDT.
-

2. System Overview

2.1 High-Level Architecture (Client, Edge, Mesh, Cloud-optional)

The system is a decentralized, peer-to-peer application with four logical layers of operation.

1. **Client (The Sovereign Node):** This is the application running on the user's device (phone, laptop). It is a complete, self-contained software stack that is fully functional offline. It stores the user's data, runs the digital twin and simulation

models, and hosts the user interface.

2. **Edge (Home Server - Optional):** For power users (e.g., farms, off-world bases), an optional edge node can be set up on a local server (like a Raspberry Pi). This node acts as a persistent hub for the Home Cell (LO), managing the robotics fleet, handling more intensive simulations, and providing a stable connection point for the local mesh network.
3. **Mesh (Peer-to-Peer Network):** When online, the client application discovers and connects directly to other peers on the local network (LAN) and the wider internet. This peer-to-peer network is used for data synchronization, federated learning, and DAO governance without relying on a central server.
4. **Cloud-Optional (Replication & Discovery Hubs):** The system does not require the cloud for core functionality. However, optional, community-run "Replication Hubs" can be used to pin important data (like Bioregion Pattern Packs) for higher availability. A lightweight cloud service may also be used as a bootstrap server to help peers discover each other initially.

2.2 Core Subsystems

- **Twin:** The core modeling engine. Manages the Human Digital Twin (HDT) and Environmental Digital Twin (EDT). It ingests sensor data, user logs, and assessment results to maintain a real-time virtual representation of the self-and-place system.
- **Deck:** The user's personal knowledge library. It functions as a collection of digital "cards"—photos, book scans, notes, web clips—that are tagged, linked, and anchored to specific places or objects in the Twin.
- **Build:** The design and crafting interface. Users browse a catalog of open-source modules (beds, towers, etc.) and assemble them into systems within their digital twin. It includes a feasibility engine that checks for constraints (sun, water, skill).
- **Sim:** The predictive simulation engine. It takes the state of the Twin and the designs from the Build system and runs scenarios over time. It models plant growth, resource consumption, pest outbreaks, and system resilience against shocks.
- **Missions:** The gamification and operations engine. It analyzes the delta between the current state of the Twin and the user's goals, generating a dynamic queue of tasks ("Missions" and "Quests") presented in Kanban and calendar views.
- **Robots:** The automation and fleet management layer. It provides the software interface for controlling and coordinating a fleet of heterogeneous robots

(growers, drones, kitchen assistants) based on routines and mission tasks.

- **Ark:** The decentralized commons and governance layer. It manages the peer-to-peer synchronization of data, the verification of claims, the workflow for graduating knowledge into the "Halls of Wisdom," and the operation of the fractal DAOs.

2.3 Data Flow & Event Bus

The system operates on an event-driven, pub/sub model.

1. **Data Capture:** A user performs an action (e.g., takes a photo of a plant, logs a meal, completes a harvest). This generates a new data object.
2. **Local Commit:** The object is structured as a **CRDT** and committed to the user's local, offline-first database. The UI updates immediately.
3. **Content Addressing:** The object is hashed, creating a unique **Content Identifier (CID)**.
4. **Sync Gossip:** When a network connection is available, the user's node "gossips" with its peers, advertising the CIDs of the new objects it has.
5. **Peer Request:** Peers that don't have these objects request them by their CID.
6. **CRDT Merge:** The receiving peer merges the new CRDT object into its own local database. Because CRDTs are used, this merge is mathematically guaranteed to be conflict-free and eventually consistent.
7. **Ark Publication (Optional):** If the user chooses to publish the object to a higher level (e.g., L1 Mesh), a "publish" event is signed with their DID and broadcast to the relevant DAO. This event contains the CID of the data bundle (e.g., a "Season Pack").

2.4 Trust & Consent Model

The trust model is zero-trust and user-centric, built on W3C standards.

- **Identity:** Each Home Cell (LO) has a unique **Decentralized Identifier (DID)**. This DID is the root of the user's identity and is controlled by their cryptographic keys, not by a central server. Humans can remain pseudonymous.
- **Claims:** Every observation, achievement, or piece of knowledge contributed to

the Ark is signed as a **Verifiable Credential (VC)**. This allows any peer to cryptographically verify who made the claim and that it hasn't been tampered with, without needing to contact the original issuer.

- **Consent & Access Control:** Data sharing is governed by explicit, granular consent. When a user shares a "Season Pack" with their L1 Mesh, they are issuing a VC that grants the Mesh DAO's DID access to that specific data bundle. Access policies are encoded in smart contracts, allowing for complex rules like:
 - **Tiered Access:** Public, community-only, or custodial knowledge.
 - **Location Fuzzing:** Raw GPS data never leaves LO by default. L1 sees a fuzzed location (e.g., a 1km hex grid), and L2 only sees the bioregion.
 - **Custodial Licenses:** For culturally sensitive knowledge, access can be restricted to specific DID groups or require physical presence for viewing (in-place only).
 - **Revocation:** A user can "tombstone" any shared data bundle at any time, broadcasting a signed revocation message that instructs all peers to mark the data as withdrawn.
-

3. Game Design & UX

3.1 Player Loop (Spawn→Scan, Collect→Design, Sim→Do, Review→Share)

The core gameplay is an iterative loop designed to be intuitive and motivating.

1. **Spawn & Scan (Observe):** The player "spawns" into their digital twin. Their first action is to scan their environment using their phone's camera (AR) or manual tools. This populates the EDT and unlocks pages in their "Biome Codex."
2. **Collect & Design (Analyze & Design):** The player collects data (biometric, environmental) and knowledge (from their Deck). The system presents "Insight Puzzles" that connect disparate data points. The player then uses the **Build System** to design solutions, crafting new systems from a catalog of modules.
3. **Sim & Do (Simulate & Act):** The player runs their design in the **Sim** engine to test its resilience and predict its yield. Once satisfied, the system generates a "Building Quest" in the **Missions** subsystem, breaking the real-world implementation into small, actionable steps.

4. **Review & Share (Measure & Adapt):** After building, the player completes "Measurement Quests" to update their Twin with real-world data, closing the feedback loop. They can then choose to synthesize their learnings into a "Guild Brief" and share it with their community mesh, earning reputation and C-Hours.

3.2 Memory-Palace UX

The primary user interface is a spatial "memory palace." Instead of navigating menus, the user navigates a 3D representation of their own space.

- **Anchors:** Key physical objects (a specific tree, a hydroponic tower, a compost bin) become persistent "anchors" in the digital twin.
- **Hotspots:** Users can attach information to these anchors. Tapping on the virtual representation of the apple tree reveals hotspots for its watering schedule, links to pruning guides in the Deck, a timeline of past harvest data, and photos of pest sightings.
- **Place-based Recall:** This UX leverages the human brain's natural strength for spatial memory. To find information about composting, the user doesn't search a file system; they simply "walk" to the compost bin in their digital twin and access the information anchored there.

3.3 Progression & Rewards

Motivation is driven by a multi-layered reward system that balances extrinsic and intrinsic drivers, based on Self-Determination Theory.⁶

- **XP & Levels (Competence):** Players earn Experience Points (XP) for completing any action in the loop (scanning, building, harvesting). Gaining levels unlocks new modules in the Build catalog, more advanced simulation scenarios, and new cosmetic options for their avatar/space.
- **Badges & Taxon Ticks (Competence):** Badges are awarded for significant achievements ("First Harvest," "Composter Level 3"). "Taxon Ticks" are earned for identifying and verifying new species for the Ark.
- **Bounties & C-Hours (Autonomy & Relatedness):** The community DAOs can post "Bounties" to fill knowledge gaps (e.g., "Map the flowering time of

elderberries in this hex grid"). Completing these bounties earns **C-Hours**, a non-financial, time-bank token that can be exchanged for help from other community members.

- **Halls of Wisdom (Relatedness & Mastery):** The ultimate intrinsic reward is having a "Guild Brief" successfully peer-reviewed and merged into the Bioregion or Planetary Halls of Wisdom, establishing the player as a recognized expert and contributing to the global commons.

3.4 Accessibility & Low-Tech Modes

To ensure inclusivity, the system is designed to work across a spectrum of technological access.

- **Printables:** The system can generate printable QR codes that can be attached to physical objects. Scanning a QR code on a garden bed with a simple camera phone can bring up a basic web form for data entry.
- **Offline Cards:** For fully disconnected participation, the system can generate printable "Observation Cards" and "Harvest Log" sheets that can be filled out by hand and later entered into the system by a community member with a device.
- **SMS/Photo Intake:** A community mesh can set up a gateway number that allows members to submit observations by sending an SMS message or a photo, which is then parsed and added to the Ark by a bot.

3.5 Internationalisation & Cultural Modes

- **I18n & L10n:** The UI is fully internationalized (I18n) to support multiple languages. Community-driven localization (L10n) efforts are incentivized through C-Hours.
 - **Cultural Protocols:** The system's schemas and governance models are designed to be flexible. A Bioregion Guild can define its own "Custodial Licenses" and cultural protocols, which are enforced by smart contracts. This allows for the respectful handling of traditional ecological knowledge and other culturally sensitive information.
-

4. Spatial Mapping & Digital Twin

4.1 Onboarding & Scale (AR, manual, CAD/GeoJSON import)

The creation of the Environmental Digital Twin (EDT) begins with mapping the user's physical space. The system supports multiple onboarding methods to accommodate different user needs and technical abilities:

- **AR Scan (Augmented Reality):** For users with modern smartphones, the primary method is an AR-powered scan. The user walks around their space, and the app uses the phone's camera and sensors (LiDAR where available) to automatically generate a 3D mesh and a 2D floor plan of the area.
- **Manual Drawing:** Users can use a simple, touch-friendly interface to draw the boundaries of their space, add walls, doors, windows, and define outdoor areas like garden beds or patios.
- **CAD/GeoJSON Import:** For advanced users or those with existing architectural or GIS data, the system allows the import of standard file formats like GeoJSON, KML, or simplified DXF files to create the base map.

4.2 Anchors & Zones (Permaculture Z0–Z5, safety zones)

Once the base map is created, the space is organized into Zones and Anchors.

- **Zones:** These are user-defined polygonal areas that represent different functions or characteristics. The system defaults to a permaculture zoning model (Zone 0: the home, Zone 1: areas visited daily, etc.), but users can create custom zones like "Vegetable Garden," "Workshop," or "Myco Room." Special zones for safety (e.g., "Keep Out Zone" for robotic operations) can also be defined.
- **Anchors:** These are specific points or objects within a zone that serve as the primary attachment points for data. An anchor can be a physical object (e.g., "Apple Tree," "Tower A") or a conceptual point (e.g., "Soil Sample Point 1").

4.3 Geometry, Layers & Constraints (footprints, clearances)

The digital twin stores precise geometric data for all objects. This includes:

- **Footprints:** The 2D area an object occupies on the map.
- **Clearances:** The 3D volume an object requires for operation or access, including vertical height. This is critical for the Feasibility Engine (e.g., preventing a tall module from being placed under a low ceiling) and for robotic navigation.
- **Layers:** Data is organized into thematic layers that can be toggled on or off, such as "Water Pipes," "Electrical Grid," "Sunlight Path (Summer)," and "Pest Sightings."

4.4 Media Anchoring (photos, PDFs, CAD, clippings)

Any piece of information in the user's **Knowledge Deck** can be visually anchored to a specific Zone or Anchor in the digital twin. A user can "pin" a PDF of a plant's care guide to the plant's anchor, attach a photo of a pest to the location it was found, or link a CAD file for a custom-built module to its representation in the twin. This creates the spatial, memory-palace UX.

4.5 Twin Storage (scene graph, CRDT fields, diffs)

The digital twin's state is stored locally as a scene graph. Each object in the scene (Zones, Anchors, Modules) is a node in the graph.

- **CRDT Fields:** Every property of an object (e.g., position, name, nutrient level in a tower) is a **CRDT**. This ensures that any changes made to the twin—whether by the user, a sensor, or a robot—can be merged without conflict during P2P synchronization.
- **Diffs & History:** All changes to the twin are stored as a series of diffs or events. This allows for a complete history of the space to be replayed, enabling features like "undo/redo" and historical analysis of how the space has evolved.

5. Knowledge Deck (Library-as-Inventory)

5.1 Card Types (photo, pdf, bookPage, map, clip, note)

The Knowledge Deck is the user's personal library, composed of "Cards." A Card is a standardized data object that can wrap various media types:

- **Photo Card:** An image with metadata (date, location, user tags).
- **PDF Card:** An imported PDF document, with searchable text.
- **BookPage Card:** A photo of a physical book page, with OCR-processed text.
- **Map Card:** A screenshot or imported image of a map.
- **Clip Card:** A snippet of text or an image clipped from a website.
- **Note Card:** A simple text note.

5.2 OCR/Tagging/Linking (auto-suggest, cosine similarity)

The system uses on-device machine learning to process cards:

- **OCR (Optical Character Recognition):** Text is extracted from all images (photos, book pages) to make it searchable.
- **Auto-Tagging:** The system suggests tags for cards based on their content (e.g., identifying plant names, keywords like "compost").
- **Linking:** The system uses cosine similarity on text content to suggest links between related cards, helping the user discover connections in their knowledge base.

5.3 Brief Synthesis (multi-card → recipe/guide)

Users can synthesize multiple Cards into a new, structured object like a **Practice** or

Recipe. For example, a user could combine a photo of their tower, a web clip about nutrient levels, and a note about their observations into a new, shareable "Lettuce Recipe" for their specific setup. This is the first step in graduating personal knowledge towards the Ark.

5.4 Provenance & Citations

Every Card automatically stores provenance data (e.g., original URL for a web clip, ISBN for a book page). When a user synthesizes a new Practice or Recipe, these sources are carried over as citations, ensuring that knowledge remains attributable as it is shared and remixed.

5.5 Deck Score & Unlocks

To gamify knowledge collection, the system includes a "Deck Score." Users earn points for adding new cards, adding tags, linking cards, and synthesizing new guides. Reaching certain score thresholds unlocks advanced features, such as more powerful search filters or new card types, providing a gentle incentive to build a rich personal library.

6. Build System (Design as Crafting)

6.1 Modules Catalog (beds, towers, aquaponics, compost, water, energy)

The Build system is a design interface where users "craft" their environment by placing modules from a catalog. The catalog contains open-source blueprints for physical systems, including:

- **Growing:** Raised beds, hydroponic/aeroponic towers, aquaponics systems, mushroom rooms.
- **Waste & Soil:** Compost bins (hot, vermi-, Bokashi), bio-digesters.
- **Water:** Rainwater harvesting tanks, greywater filters, swales, ponds.
- **Energy:** Solar panel mounts, micro-wind turbine stands, battery enclosures.

6.2 Feasibility Engine (sun/water/power/skill checks)

When a user places a module in their digital twin, the Feasibility Engine runs a series of checks against the twin's data:

- **Sun Check:** Uses sun path data layered onto the twin to verify if a plant module will receive adequate light.
- **Water/Power Check:** Checks if the module is placed within reach of necessary water and power connections defined in the twin's layers.
- **Skill Check:** Cross-references the module's required skills (e.g., "Basic Plumbing," "Soldering") with the user's self-assessed skills in their HDT. It will flag a potential skill gap and link to relevant guides in the Deck or Ark.

6.3 Companion Planting Matrix & Conflicts

For growing modules, the system includes a companion planting database (see Appendix A). When a user adds a plant to a bed or tower, the system visually highlights synergistic or antagonistic relationships with neighboring plants, guiding the user towards creating beneficial plant guilds.

6.4 IO Graphs (water, nutrients, compost, harvest routes)

The Build system allows users to visualize the flows within their design. Users can draw connections between modules to create Input/Output (IO) graphs. For example, a user can draw a line from a "Compost Bin" output to a "Garden Bed" input, defining a nutrient flow. This makes the systems-thinking aspect of permaculture tangible and

helps in designing closed-loop, regenerative systems.⁷

6.5 BOM/Costing (DIY vs salvage, labour/robot time)

Once a design is complete, the system can generate a Bill of Materials (BOM). For each component, the user can tag it as "to buy," "DIY," or "salvaged." The system provides fields for estimating costs and labor time (for both humans and robots), allowing for basic project planning and budgeting.

6.6 Templates & Patterns (Bioregion packs, meta-patterns)

New users don't start from scratch. The Build system comes pre-loaded with starter templates. More importantly, it can directly import "Pattern Packs" from the Fractal Ark. A user can download a "Drought-Tolerant Balcony Garden" pattern from their L2 Bioregion Guild, and the system will automatically populate their Build canvas with a proven, locally-attuned design that they can then customize.

7. Aeroponics/Hydroponics Towers

7.1 Module Types (Tower.Aero, Tower.Hydro, Nursery.DWC, SproutRack, Microgreens)

This subsystem provides detailed digital models and control logic for various soilless growing modules. The catalog includes specific subtypes:

- **Tower.Aero:** High-pressure aeroponics tower for fruiting plants.
- **Tower.Hydro:** Nutrient Film Technique (NFT) or Drip hydroponic tower for leafy greens.

- **Nursery.DWC:** Deep Water Culture module for starting seedlings.
- **SproutRack:** Automated sprouting tray system.
- **Microgreens:** Stacked microgreens growing trays with automated lighting/watering.

7.2 Sensors & Control Loops (EC, pH, temp, ORP, pressure)

Each tower module in the digital twin has defined data inputs for key sensors, including Electrical Conductivity (EC), pH, water temperature, ambient temperature, and water pressure. The system's software provides control loops that can be run on an Edge device (e.g., Raspberry Pi) to automate the tower's operation, such as activating pumps or nutrient dosers to maintain target setpoints.

7.3 Nutrient Recipes & Micronutrient Flushes

The system uses the Recipe object type to manage nutrient schedules. A recipe is a timeline of target EC and pH values for different growth stages (e.g., "Nursery," "Vegetative," "Fruiting"). It also includes schedules for periodic micronutrient additions or system flushes to prevent salt buildup. Users can create their own recipes or download proven ones from the Ark.

7.4 CIP/Sanitation & HACCP Logs

To ensure food safety, the tower subsystem includes a module for Clean-in-Place (CIP) and sanitation protocols. Users can schedule and log sanitation events (e.g., "Reservoir Bleach Cycle," "Line Flush"). For users selling produce, the system can generate logs compliant with Hazard Analysis and Critical Control Points (HACCP) principles, documenting that critical sanitation tasks were completed.

7.5 Failure Modes & Self-Healing

The system's control software includes logic for common failure modes (e.g., pump failure, clogged nozzle, sensor drift). It can trigger alerts to the user's device and, in some cases, attempt self-healing actions. For example, if a pressure sensor detects a clog, the system might trigger a short, high-pressure "purge" cycle on the pump before alerting the user.

7.6 UI: Towers Palette, Nutrient Curves, Sanitation Calendar

The user interface for this subsystem provides:

- **Towers Palette:** A dashboard showing the real-time status of all connected towers.
 - **Nutrient Curves:** A graphing tool to visualize historical sensor data (EC, pH) against the target curves defined in the active Recipe.
 - **Sanitation Calendar:** A calendar view showing past and upcoming scheduled sanitation tasks.
-

8. Simulation & Scenarios

8.1 Season Engine (weekly ticks; growth, yield, labour)

The Sim engine operates on a discrete-time model, advancing in "weekly ticks." At each tick, it updates the state of the digital twin based on probabilistic models for:

- **Growth:** Plant growth is calculated based on available light (from sun path data), water, and nutrients, using simplified growth models for common plant types.
- **Yield:** The engine predicts harvest quantities and timing.
- **Labour:** It estimates the human or robotic labor hours required for tasks like

planting, pruning, and harvesting.

8.2 Weather & Climate Profiles (local, heatwaves, drought, cyclone)

The simulation can be run against different weather profiles. The default is the historical average for the user's location (pulled from public data). Users can also apply stress-test profiles like "2-Week Heatwave," "Extended Drought," or "Cyclone Event" to see how their system design holds up.

8.3 Pests, Disease & Biodiversity Index

The Sim includes a simplified ecological model. It simulates the probability of pest or disease outbreaks based on the diversity of plantings (monocultures are more vulnerable). A higher biodiversity score in the EDT reduces the chance of negative events and increases the chance of positive ones (e.g., the "arrival" of beneficial predatory insects).

8.4 Resource Budgets (water, energy, nutrients)

Users can set budgets for key resources. The Sim will track the consumption of water, energy (for pumps and lights), and nutrients against these budgets, providing a forecast of when resources might be depleted.

8.5 Shock Scenarios & Resilience Scores

This is the core of the Sim's utility for resilience planning. Users can run pre-defined shock scenarios, such as:

- **"Grid Down"**: Simulates a 72-hour power outage.

- **"Supply Chain Disruption"**: Simulates the inability to purchase new nutrients or supplies for 1 month.
- **"Water Rationing"**: Simulates a 50% reduction in available water.

After running a scenario, the system provides a **Resilience Score**, a composite metric indicating how well the system maintained food production and viability during the shock.

8.6 Micro-games (pollinate, prune, predator release)

To make complex concepts more engaging, the Sim includes interactive micro-games. For example, to teach the importance of pollinators, a user might have to play a mini-game where they manually pollinate blossoms to ensure a fruit harvest. These "serious games" serve as educational tools embedded within the simulation.⁹

9. Missions & Operations

9.1 Mission Generator (deltas → tasks)

The Missions subsystem is the user's primary task manager. It automatically generates tasks by comparing the current state of the digital twin to the desired state or a pre-defined schedule.

- **Delta-based Tasks**: If the Sim predicts a nutrient deficiency in a tower next week, it generates a mission: "Add micronutrients to Tower A."
- **Scheduled Tasks**: Based on a Recipe or Ops Calendar, it generates routine tasks like "Check pH levels" or "Clean pump filter."
- **User-generated Tasks**: Users can create their own one-off or recurring tasks.

9.2 Kanban & Calendar Views

Missions are presented in user-friendly formats. A **Kanban board** allows users to drag tasks between columns ("To Do," "In Progress," "Done"). A **Calendar view** shows time-sensitive tasks on a daily, weekly, or monthly layout.

9.3 Proof-of-Work (photos, logs) → Twin Updates

To complete a mission, the user must provide a simple "Proof-of-Work." This could be:

- **Photo:** Taking a photo of the completed task (e.g., a newly planted bed).
- **Log Entry:** Entering a value (e.g., the new pH reading).
- **Check-off:** Simply marking the task as done.

Completing a mission automatically updates the state of the digital twin. For example, logging a new pH reading updates the pH CRDT field for that tower's water reservoir.

9.4 Ops Calendar (sow/transplant/harvest/seed-save)

This is a specialized calendar focused on the lifecycle of plants. It automatically populates with optimal dates for sowing, transplanting, harvesting, and seed saving, based on the plant types in the user's twin and the local climate data from the L2 Bioregion Guild.

9.5 Disaster Playbooks

A Playbook is a pre-packaged set of missions designed to respond to a specific disaster scenario. For example, if a "Wildfire Smoke" alert is received from an external data feed, the system can activate the "Air Quality Playbook," which might include missions like "Close all greenhouse vents," "Check indoor air filter," and "Harvest

sensitive leafy greens."

9.6 Review & Export (one-pagers, checklists)

The system allows users to export operational data in simple formats. A user could export a "Weekly Harvest Report" as a PDF or generate a printable "Daily Maintenance Checklist" for a less tech-savvy family member to use.

10. Robotics & Automation

10.1 Robot Catalog

The system is designed to be extensible with a robotics layer. The software includes a catalog of digital twin models for various open-source or commercial robot types, categorized by function:

- **Humanoid (R-H1):** General-purpose bipedal robot for complex manipulation tasks.
- **Grower UGV (R-G2):** Unmanned Ground Vehicle with robotic arm for planting, weeding, and harvesting.
- **Drone (R-D1):** Quadcopter for aerial surveying, sensor reading, and spot-spraying.
- **Kitchen (R-K1):** Robotic arm in a fixed location for food processing and prep.
- **Myco (R-M1):** Specialized robot for managing mushroom fruiting chambers.
- **Wash/Waste (R-W1):** Robot for cleaning modules and managing compost.
- **Courier (R-C1):** Small UGV for moving items between zones.
- **Engineer (R-E1):** Robot equipped for basic repairs and maintenance.
- **Sentry (R-S0):** Mobile sensor platform for security and environmental monitoring.

10.2 Capabilities Matrix

Each robot model in the catalog has a defined **Capabilities Matrix** (see Appendix C). This matrix lists the tasks the robot can perform (e.g., "Grasp Object," "Measure pH," "Navigate to Anchor") and the tools it requires. This allows the Fleet Manager to assign tasks only to capable robots.

10.3 Tasks, Routines & Recovery Graphs

- **Tasks:** The smallest unit of robotic work, equivalent to a single command (e.g., `moveTo(x,y,z)`).
- **Routines:** A sequence of tasks that accomplishes a Mission (e.g., the "Harvest Tomato" routine).
- **Recovery Graphs:** For each routine, there is a corresponding graph of recovery behaviors for potential failures. If a "Grasp Object" task fails, the recovery graph might define steps like "Retry," "Adjust Gripper," "Scan for Object," and finally "Alert Human."

10.4 Fleet Manager (assignment, energy/water arbitration, locks)

The Fleet Manager is the central brain of the robotics subsystem, typically running on an Edge node. Its responsibilities include:

- **Task Assignment:** Assigning missions from the Ops queue to available and capable robots.
- **Arbitration:** Managing shared resources. If two robots need to use the charging station, the Fleet Manager decides the priority based on their current energy levels and upcoming tasks.
- **Locks:** Implementing spatial locks to prevent collisions. Before a robot enters a narrow corridor, it requests a lock from the Fleet Manager, ensuring no other robot will enter at the same time.

10.5 Zones, Policies, HRI Scripts (Welcome Home, Handover)

- **Zones & Policies:** The digital twin's zones are used to define robot behavior. A "Human-Only Zone" policy would prevent any robot from entering that area. A "Quiet Zone" policy would restrict noisy operations during certain hours.
- **HRI Scripts:** Human-Robot Interaction is managed through pre-defined scripts. A "Handover" script defines the precise movements for a robot to safely pass a tool or a harvested vegetable to a human. A "Welcome Home" script might trigger a robot to perform a specific greeting or status report when the user arrives.

10.6 Safety, Compliance & Black-Box Logging

Safety is paramount. The robotics subsystem includes:

- **E-Stops:** A virtual "Emergency Stop" button in the app that immediately halts all robotic activity.
 - **Compliance Checklists:** For commercial use, the system includes checklists based on relevant ISO/IEC safety standards.
 - **Black-Box Logging:** All robot sensor data and commands are stored in a rolling, encrypted log on the Edge device. In the event of an accident, this log can be reviewed to determine the cause.
-

11. Off-World Modes (Moon/Mars → Earth Welcome)

11.1 Commissioning Timeline (T-30 to launch)

For off-world applications, the system includes a specialized project management timeline. This "Commissioning Timeline" starts before the mission (T-minus 30 days)

and includes missions for final system checks, data loading, and pre-positioning of robotic assets.

11.2 Autonomy Modes (Assist, Supervised, Scheduled, Adaptive, Resilience)

The system can operate in several autonomy modes, critical for managing high-latency communication with an off-world base:

- **Assist:** A human operator directly controls a robot (tele-operation).
- **Supervised:** A robot executes a routine, but requires human confirmation at key decision points.
- **Scheduled:** The Fleet Manager executes the Ops Calendar autonomously, with no human intervention required unless an error occurs.
- **Adaptive:** The system's AI can modify the schedule in response to real-time data (e.g., delaying an outdoor task if a solar flare is detected).
- **Resilience:** If communication with Earth is lost, the system enters a fully autonomous mode, using the Sim engine and pre-loaded Disaster Playbooks to make decisions that maximize the probability of crew survival.

11.3 Community Mode (surplus routing, verification, ethics)

Even in an off-world context, the system connects to the Ark. The base can:

- **Route Surplus:** Log surplus resources (e.g., excess oxygen, water, data) that could be made available to future missions.
- **Verify Data:** Dedicate spare computing power to help verify claims and maintain the integrity of the Ark's blockchain ledger.
- **Propose Ethical Frameworks:** Contribute to the L3 Planetary Stewards DAO, proposing new ethical guidelines for off-world resource use based on their unique experience.

11.4 Welcome Prep (–14d), Arrival Day Protocols

This mode is designed for preparing a habitat for human arrival. Starting 14 days before arrival, the "Welcome Prep" playbook activates, running missions to ensure life support systems are optimal, initial crops are ready for harvest, and the environment is stable. On arrival day, specific protocols are executed to greet the crew and provide a full system status briefing.

11.5 Latency & Tele-op Considerations

The UI for tele-operation includes features specifically designed to handle high communication latency. Operators see a "ghost" of the robot showing its predicted position based on the last command sent, while the "real" robot's position is updated when data is received. This allows for more intuitive control despite the time lag.

12. Fractal Ark (Decentralised Commons)

12.1 Topology (L0 Home, L1 Mesh, L2 Bioregion, L3 Planetary Commons)

The Ark is built on the four-layer fractal topology, which allows information to scale from the personal to the planetary level while maintaining local autonomy.

- **L0 Home Cell:** The user's sovereign node. All data is created and stored here first.
- **L1 Neighbour Mesh:** A peer-to-peer network of 8-50 nearby homes. Data is shared via gossip protocols for local needs like seed swaps and surplus routing.
- **L2 Bioregion Guild:** A larger network, often defined by a watershed or ecoregion. Anonymized data summaries from L1 meshes are aggregated here to create climate-specific Pattern Packs and seasonal calendars.
- **L3 Planetary Commons:** The global layer. It maintains the master taxonomic indices and open standards. It also serves as a replication hub for the most

valuable data from all bioregions.

12.2 Schemas & Standards (Darwin Core, MlxS, OSM, Schema.org)

To ensure interoperability, the Ark's data objects are built on established open standards:

- **Taxon:** Extends **Darwin Core (DwC)** for biological classification.
- **Microbial Data:** Will incorporate **Minimum Information about any (x) Sequence (MlxS)** standards.
- **Place:** Uses **OpenStreetMap (OSM)** tags for features and GeoJSON for geometry.
- **Recipe & Practice:** Uses **Schema.org** vocabulary for recipes and how-to guides to ensure machine readability.

12.3 Identity & Consent (DID/VC, access tiers, revocation)

The trust layer is built on W3C standards for Self-Sovereign Identity.

- **Identity:** Every node (user, DAO) has a **DID**.
- **Claims:** All data shared to the Ark is packaged as a **VC**, signed by the creator's DID.
- **Access Tiers:** Access is controlled by smart contracts, allowing for public, community-only, and custodial (guardian-controlled) knowledge tiers.
- **Revocation:** Users can broadcast a signed "tombstone" message to revoke any VC they have previously shared.

12.4 Content Addressing & CRDT Merge (bundles, tombstones)

The data mechanics ensure resilience and verifiability.

- **Content Addressing:** All data objects are stored in a decentralized file storage system (like IPFS) and are addressed by their content hash (CID).

- **Merkle Bundles:** Data is shared in "Season Packs," which are **Merkle-DAGs**. The root hash of the DAG guarantees that the entire bundle is tamper-proof.
- **CRDT Merge:** All individual data objects are **CRDTs**, allowing for conflict-free, automatic merging of data from different peers.

12.5 Reputation & Attestations (claim-level scoring)

Trust in the Ark's data is not based on who a person is, but on the verification of their claims.

- **Attestations:** Other users (or robots) can "attest" to an observation by signing it with their own DID, creating a peer-review VC.
- **Reputation Score:** Each claim (e.g., an Observation object) accumulates a reputation score based on the number and quality of attestations it receives. This allows users to filter for highly-verified data.

12.6 Halls of Wisdom Workflow (submit → review → custodial check → pattern)

This is the process by which local knowledge becomes a global public good.

1. **Submission:** A user or L1 Mesh synthesizes a "Guild Brief" (a Practice bundle with supporting evidence) and submits it to their L2 Bioregion Guild DAO.
2. **Peer Review:** The DAO triggers a review process, where other members of the bioregion can attest to the brief.
3. **Custodial Check:** If the brief contains culturally sensitive information, it is automatically flagged for approval by the designated Custodial DID for that knowledge type.
4. **Pattern Creation:** If accepted, the brief is versioned and published as an official "Bioregion Pattern," making it available for others to download and use.

12.7 Cultural & Indigenous Data Sovereignty

The system is explicitly designed to protect cultural and Indigenous knowledge. The **AccessPolicy** object allows Custodians to define granular rules, such as making certain SeedLot or Practice objects "custodial," meaning they can only be accessed by members of a specific DID group, or even making a Place object's data only viewable when the user is physically at that location.

13. Data Model & APIs

13.1 Core Objects (Space, Anchor, Module, Plant, Card, Scenario, Mission, SeedBank)

- **Space:** The root object for a user's digital twin, containing all zones and anchors.
- **Anchor:** A specific point or object in the space.
- **Module:** A physical system component (e.g., a tower, a compost bin).
- **Plant:** An instance of a biological plant, linked to a Taxon object.
- **Card:** A single piece of knowledge in the user's Deck.
- **Scenario:** A simulation profile (e.g., "Drought").
- **Mission:** A single task in the operations queue.
- **SeedBank:** A user's personal inventory of SeedLot objects.

13.2 Robotics Objects (Robot, Capability, Task, Routine, Policy, FleetManager, HRI)

- **Robot:** An instance of a specific robot model.
- **Capability:** A single function a robot can perform.
- **Task:** The smallest unit of robotic work.
- **Routine:** A sequence of tasks to complete a mission.
- **Policy:** A rule governing robot behavior in a specific zone.
- **FleetManager:** The central control object for the robotics subsystem.
- **HRI:** A Human-Robot Interaction script.

13.3 Towers Objects (**Tower**, **ResMix**, **DoseBlock**, **ValveManifold**, **HaccpLog**, **Recipe**)

- **Tower:** The main object for an aeroponic/hydroponic module.
- **ResMix:** Represents the nutrient reservoir mix, tracking EC, pH, and volume.
- **DoseBlock:** A record of a nutrient dosing event.
- **ValveManifold:** The digital representation of the plumbing manifold.
- **HaccpLog:** A log entry for a food safety-related task.
- **Recipe:** A timeline of target nutrient and environmental conditions.

13.4 Ark Objects (**Taxon**, **Place**, **Observation**, **SeedLot**, **Practice**, **SeasonPack**, **Reputation**, **AccessPolicy**)

(See Appendix F for full JSON Schemas)

- **Taxon:** Biological classification, extends Darwin Core.
- **Place:** A geo-referenced location with privacy controls.
- **Observation:** A record of a Taxon at a Place.
- **SeedLot:** A batch of seeds with provenance and access rules.
- **Practice:** A how-to guide or protocol.
- **SeasonPack:** A shareable, content-addressed bundle of other Ark objects.
- **Reputation:** A score attached to a specific claim.
- **AccessPolicy:** A smart contract defining access rules for custodial data.

13.5 Event Model (**publish**, **attest**, **adopt**, **revoke**, **merge**)

(See Appendix G for API Reference)

- **publish:** Announce a new shareable bundle to a DAO.
- **attest:** Peer-review and verify another user's claim.
- **adopt:** Import a pattern from the commons into a local space.
- **revoke:** Tombstone a previously published object.

- **merge:** Low-level event for CRDT synchronization.

13.6 API Endpoints (local, mesh, gateway), Webhooks, Schema Validation

- **Local API:** The application exposes a local API (e.g., via localhost) for user scripts and third-party tool integration.
- **Mesh API:** Communication between peers happens over a defined set of P2P API endpoints.
- **Gateway API:** Optional gateway nodes can expose a public API for interacting with a specific Bioregion Guild.
- **Webhooks:** Users can configure webhooks to be triggered by system events (e.g., "Harvest Ready," "Robot Task Failed").
- **Schema Validation:** All data objects are validated against the published JSON schemas before being committed or synced.

13.7 Storage (content-addressed blobs, indices, encryption at rest)

- **Blob Storage:** All raw data (images, PDFs, etc.) is stored as content-addressed blobs in a decentralized file storage system like IPFS.
 - **Indices:** The local database (e.g., SQLite) stores the CRDT data and acts as an index, pointing to the CIDs of the blobs.
 - **Encryption at Rest:** All data on the user's device is encrypted by default using keys controlled by the user.
-

14. Security, Privacy, Safety

14.1 Threat Model (device loss, data leak, supply-chain)

The security model addresses several key threats:

- **Device Loss/Theft:** Mitigated by strong encryption at rest. Data is unreadable without the user's keys.
- **Data Leak (from peers):** Mitigated by the privacy-preserving design. Raw, sensitive data is not shared by default, only anonymized learnings or explicitly published VCs.
- **Malicious Peer:** Mitigated by the use of VCs and a reputation system. Data from untrusted peers can be ignored.
- **Supply-Chain Attack (on software):** Mitigated by using open-source software, allowing for community audits of the codebase.
- **Supply-Chain Attack (on hardware):** Mitigated by providing open-source hardware designs, allowing users to build and verify their own modules.

14.2 Encryption, Keys & Backups (local custody, social recovery)

- **Local Custody:** The user's private keys for their DID are generated and stored on their device, never shared with a central server.
- **Encryption:** All P2P communication is encrypted end-to-end.
- **Backups:** The system provides tools for creating encrypted backups of the user's local data.
- **Social Recovery:** Users can implement a social recovery mechanism, where a user's keys can be recovered with the help of a pre-selected group of trusted peers (other DIDs), without any central authority.

14.3 Location Fuzzing & Red-list Shielding

- **Location Fuzzing:** As described in the Trust Model, precise GPS coordinates are fuzzed to a lower-resolution grid before being shared to higher-level network tiers.
- **Red-list Shielding:** The system cross-references all Taxon observations against global and regional lists of threatened or endangered species. If a match is found, the location data for that observation is automatically obscured or removed before it can be published to the Ark, preventing poachers or others from using the data to find and harm vulnerable populations.

14.4 Safety Interlocks (zones, tools, pets/children, biosecurity)

- **Robotic Safety:** The system enforces hardware-level safety interlocks. A robot cannot operate a sharp tool unless it is within a designated "Workshop Zone" and has confirmed via its sensors that no humans, pets, or children are within its operating envelope.
- **Biosecurity:** When a user logs a pest or disease observation, the system can automatically generate a "Biosecurity Protocol" mission. This may include tasks like "Quarantine affected plants" and "Sanitize tools after use." The system will prevent a robot from moving from a quarantined zone to a clean zone without first passing through a designated sanitation station.

14.5 Audit & Traceability (HACCP, ops logs, black-box)

- **Immutable Ledger:** All key events (harvests, sanitation logs, data publications) are recorded as transactions on the Ark's blockchain layer. This creates a permanent, tamper-proof audit trail.
 - **HACCP Compliance:** The system's logging capabilities are designed to meet HACCP requirements for commercial food producers.
 - **Black-Box Logging:** The robotics' black-box logs provide full traceability for accident investigation and performance analysis.
-

15. Compliance & Ethics

15.1 Food Safety & HACCP

The system provides tools and checklists to help users comply with local food safety regulations. The HACCP logging features are a core component of this, allowing for

auditable records of food production and sanitation processes.

15.2 Electrical/Water/Building Codes (regional variance, checklists)

The open-source blueprints in the **Build System** catalog include notes and checklists regarding common electrical, water, and building codes. However, the system clearly states that the user is ultimately responsible for ensuring their creations comply with all local regulations.

15.3 Robotics Safety (ISO/IEC refs)

The design of the robotics subsystem is guided by principles from relevant safety standards, such as ISO 10218 (Robots and robotic devices — Safety requirements for industrial robots). Documentation will reference these standards and provide guidance on implementing safe robotic systems.

15.4 Accessibility Standards

The user interface will be designed to meet Web Content Accessibility Guidelines (WCAG) 2.1 AA standards, ensuring it is usable by people with a wide range of disabilities. This includes support for screen readers, keyboard navigation, and sufficient color contrast.

15.5 Cultural Protocols & Custodial Licences

The system's design is rooted in the ethical principle of respecting cultural knowledge. The **Custodial License** is a core feature, allowing Indigenous and other communities to encode their specific data governance protocols directly into the system's smart

contracts. This moves beyond simple access control to enable true data sovereignty.

16. Implementation Plan

16.1 MVP Definition (must-haves for first play)

The Minimum Viable Product (MVP) will focus on the core single-player experience (LO) to validate the player loop and core technologies. It will include:

- Manual 2D spatial mapping.
- Basic HDT (manual entry for mood, sleep) and EDT (placing anchors).
- The Knowledge Deck with Photo and Note card types.
- A limited Build System catalog (1 bed type, 1 tower type).
- A simplified Mission generator for basic gardening tasks.
- The core player loop: Scan (add a plant), Design (place a bed), Do (complete mission), Review (log harvest).

16.2 Phased Roadmap

- **P0 Initiation (Q1):** Finalize core team, establish legal entity (e.g., a foundation), secure initial funding.
- **P1 Design (Q2-Q3):** Detailed UX/UI design, finalize technical architecture and schema definitions.
- **P2 Construction (Q4-Q2 next year):** MVP development.
- **P3 Operations (Q3 next year):** MVP launch to a closed beta group. Gather feedback and iterate.
- **P4 Ark Mesh (Q4 next year):** Develop and launch L1 and L2 networking features (P2P sync, DAOs).
- **P5 Robotics (Year 3):** Develop and release the robotics subsystem and open-source hardware designs.
- **P6 Off-World (Year 4+):** Partner with space agencies/companies to develop and

test the off-world modes.

16.3 Milestones & Exit Criteria

Each phase has clear exit criteria. For example, P2 (Construction) is complete when the MVP passes all integration tests and is successfully deployed to internal test devices. P3 (Operations) is complete when the beta group shows a 20% week-over-week retention rate for at least one month.

16.4 Team Roles & RACI

A standard RACI (Responsible, Accountable, Consulted, Informed) matrix will be developed for the core team, covering roles like Product Lead, Engineering Lead, Design Lead, and Community Manager.

16.5 Budget & Bill of Materials (DIY vs off-the-shelf vs custom)

An initial budget will be developed to cover P0-P3. A Bill of Materials for the "Starter Ark Kit" hardware (e.g., for a basic hydroponic tower) will be created, providing cost estimates for users choosing DIY, off-the-shelf, or custom-fabricated components.

16.6 Risk Register & Mitigations

A risk register will be maintained, tracking technical, financial, and community risks. Key risks include:

- **Technical Risk:** CRDT merge complexity. **Mitigation:** Use a well-supported library like Automerge or Y.js and build extensive test suites.
- **Community Risk:** Failure to attract an initial user base. **Mitigation:** Partner with

existing permaculture and citizen science communities for the beta launch.

- **Financial Risk:** Running out of funding before P4. **Mitigation:** Focus on a lean MVP and explore grant funding opportunities for the open-source components.
-

17. DevOps & Deployment

17.1 Platforms (mobile, desktop, AR/VR)

The initial platform targets will be iOS, Android, and desktop (via Electron). Future development will include dedicated AR/VR interfaces for a more immersive memory-palace experience.

17.2 Offline-First Architecture (sync, conflict, retries)

The application will be built using an offline-first architecture. All data is written to a local database first, and the UI reads only from this local source. A background service will handle synchronization with peers, managing retries and back-off strategies for intermittent network connections.

17.3 Mesh/Gateway Nodes (home server, neighbourhood hub)

While the application is fully P2P, users can set up optional persistent nodes:

- **Home Server (Edge Node):** A user can run the node software on a home server (e.g., Raspberry Pi) to provide a stable peer for their own devices and to run the robotics Fleet Manager.
- **Neighbourhood Hub:** A community can set up a shared hub that acts as a bootstrap and replication node for their L1 Mesh, improving discovery and data

availability.

17.4 Telemetry & Privacy-Preserving Analytics

To improve the application, anonymous usage data is needed. This will be implemented using **federated analytics**. The application will compute aggregate, non-identifiable statistics locally (e.g., "number of towers created," "average mission completion time") and only share these aggregated results, not the underlying user data.

17.5 CI/CD, Release Channels, Feature Flags

A standard CI/CD (Continuous Integration/Continuous Deployment) pipeline will be used for automated testing and builds. The project will maintain stable, beta, and nightly release channels. New features will be deployed behind feature flags, allowing for gradual rollouts and A/B testing.

17.6 Backup/Restore & Disaster Recovery

The application will include a user-controlled backup and restore function. Users can export their entire local database as a single, encrypted file, which they can store securely. The restore function will allow them to fully recover their Sovereign Node on a new device.

18. Testing & Validation

18.1 Unit & Integration Tests (sim, controls, merge)

The codebase will have comprehensive test coverage, including:

- **Unit Tests:** For individual functions and components.
- **Integration Tests:** For interactions between subsystems (e.g., testing that a completed Mission correctly updates the Twin).
- **Simulation Tests:** The Sim engine will be tested against known outcomes.
- **Merge Tests:** Extensive testing of the CRDT merge logic under various network partition and concurrency scenarios.

18.2 Field Trials (homes, schools, farms)

After the initial closed beta, the system will be tested in a series of field trials with different user groups (e.g., urban apartment dwellers, suburban families, a community garden, a small farm) to gather qualitative feedback and validate its utility in diverse real-world contexts.

18.3 Robotics Dry-Runs & Safety Drills

Before any robot is allowed to operate autonomously, it must pass a series of dry-run tests within the digital twin. Once deployed, regular automated safety drills (e.g., testing the E-stop functionality) will be conducted.

18.4 Nutrient/Tower Bench Tests (EC/pH, CIP)

The open-source hardware designs will be accompanied by detailed bench testing protocols. These will provide instructions for calibrating sensors and validating the performance of the nutrient dosing and Clean-in-Place systems.

18.5 Ark Merge & Revocation Drills

The decentralized Ark network will be tested with large-scale drills to ensure the P2P sync, CRDT merge, and data revocation mechanisms are robust and performant under heavy load.

18.6 Usability & Accessibility Testing

The UI/UX will be continuously tested with real users, including users with disabilities, to ensure the application is intuitive, efficient, and accessible to the widest possible audience.

19. Education & Community

19.1 Tutorials, Quests, Curriculum Packs (schools, TAFEs, makerspaces)

The system will include a rich set of educational resources:

- **Tutorials:** The onboarding process will be a series of interactive, gamified tutorials.
- **Quests:** In-game quests will be designed to teach core permaculture and systems thinking concepts.
- **Curriculum Packs:** Downloadable curriculum packs will be created for educators in schools, vocational colleges (TAFEs), and makerspaces to use the system as a hands-on STEM learning tool.

19.2 Mentor/Guild Programs & Bounties

Community engagement will be fostered through:

- **Mentor Program:** Experienced users can sign up to be mentors, earning C-Hours for helping new users.
- **Guilds:** Users can form special interest groups or "Guilds" (e.g., "Mycology Guild," "Robot Builders Guild").
- **Bounties:** DAOs can post bounties for specific tasks, such as translating the app into a new language or designing a new open-source module.

19.3 Seed Library & Tool Library Playbooks

To support L1 Neighbour Meshes, the system will provide "Playbooks"—step-by-step guides for establishing and managing physical community resources like a shared seed library or tool library, including model governance rules for the L1 DAO.

19.4 Events & Challenges (Season Packs, Resilience Cups)

Regular community events will be run to drive engagement:

- **Season Pack Showcase:** A quarterly event where users can share their most successful "Season Packs."
- **Resilience Cup:** An annual challenge where users compete to see whose system design achieves the highest Resilience Score in a standardized shock scenario.

19.5 Support & Moderation

Community support will be provided through a public forum and chat channels. A moderation team (initially volunteers, incentivized with C-Hours) will enforce a community code of conduct to ensure a safe and welcoming environment.

20. Economics & Incentives

20.1 C-Hours Timebank & Local Credits

The primary economic layer is a non-financial, reputation-based system.

- **C-Hours (Community Hours):** Users earn C-Hours for actions that benefit the commons (e.g., verifying Ark data, mentoring, contributing to open-source designs). These can be exchanged for help from other users, creating a peer-to-peer timebank.

20.2 Surplus Routing & Barter

The L1 Neighbour Mesh facilitates a local, circular economy. The system includes a "Surplus" feature where users can list items they have in excess (e.g., produce, seeds, compost). Other mesh members can then claim these items, arranging for barter or gifting.

20.3 Optional Web3 (blueprint minting, royalties, guardrails)

For users who opt-in, the system provides a bridge to Web3 economies:

- **Blueprint Minting:** A user who designs a particularly successful or innovative module can mint its blueprint as a Non-Fungible Token (NFT).
- **Royalties:** The smart contract for the NFT can include a royalty, where the original designer receives a small percentage every time the blueprint is used or sold.
- **Guardrails:** The system's DAOs will establish ethical guardrails to prevent extractive financialization and ensure these features align with the core ethos of

the project.

20.4 Sustainability & Funding Models

The project will be sustained through a hybrid model:

- **Grants:** Seeking funding from foundations that support open-source, environmental, and decentralized technology projects.
 - **Service Subscriptions:** Offering paid subscriptions for optional, value-added services, such as dedicated gateway nodes, enterprise-level support for farms, or advanced analytics for researchers.
 - **Donations:** Accepting donations from the community to support public infrastructure like replication hubs and the L3 Planetary Stewards DAO.
-

21. Appendices

A. Companion Planting Matrix (starter)

This matrix provides a basic guide to synergistic and antagonistic relationships between common garden plants, a core concept in designing resilient plant guilds.

Plant	Companions (Beneficial)	Antagonists (Avoid Planting Nearby)	Notes
Tomatoes	Basil, Carrots, Marigolds, Chives, Borage	Cabbage family, Potatoes, Fennel, Corn	Basil repels tomato hornworms. Marigolds deter nematodes.
Beans (Bush)	Corn, Squash, Carrots, Cucumber, Potatoes	Onions, Garlic, Peppers, Fennel	Part of the "Three Sisters" guild with corn and squash.

Lettuce	Carrots, Radishes, Strawberries, Cucumbers	Cabbage family (e.g., Broccoli, Cabbage)	Shaded by taller plants like carrots in hot weather.
Carrots	Beans, Lettuce, Onions, Rosemary, Sage	Dill, Fennel, Parsnips	Rosemary and sage help deter the carrot rust fly.
Cucumbers	Beans, Corn, Peas, Radishes, Sunflowers	Aromatic herbs (e.g., Sage), Potatoes	Sunflowers can provide a natural trellis.
Peppers	Basil, Carrots, Onions, Spinach, Tomatoes	Beans, Fennel, Kohlrabi	Basil can help deter some pests.
Onions/Garlic	Carrots, Beets, Lettuce, Chamomile	Beans, Peas, Asparagus	The strong scent can deter pests from companion plants.
Squash	Corn, Beans, Peas, Radishes, Marigolds	Potatoes	Part of the "Three Sisters" guild. Marigolds deter pests.

B. Nutrient Recipes & EC/pH Targets (leafy, fruiting, microgreens)

This table provides starter nutrient targets for common hydroponic and aeroponic systems. These are general guidelines and should be adjusted based on specific plant needs and observations.

Crop Type	Growth Stage	Target EC (mS/cm)	Target pH	Notes
Leafy Greens	Seedling	0.6 - 1.0	5.8 - 6.2	Keep nutrients low to avoid burning young roots.

(Lettuce, Spinach)	Vegetative	1.2 - 1.8	6.0 - 6.5	Maintain stable pH for optimal nutrient uptake.
Fruiting Plants	Seedling	0.8 - 1.2	5.8 - 6.2	Slightly higher EC for initial growth.
(Tomatoes, Peppers)	Vegetative	1.8 - 2.5	5.5 - 6.0	Lower pH range helps with micronutrient availability.
	Fruiting	2.5 - 3.5	5.5 - 6.0	Higher EC supports fruit development. Monitor for salt buildup.
Herbs	All Stages	1.0 - 1.6	6.0 - 6.8	Generally prefer lower nutrient concentrations.
(Basil, Mint)				
Microgreens	All Stages	0.5 - 0.8	6.0 - 6.5	Very low nutrient requirements; often grown in plain water.

C. Capability Matrix (robots × tasks × tools)

This matrix outlines the core capabilities of the defined robot models and the tools required for them to perform specific tasks. The Fleet Manager uses this data for task assignment.

Task	R-H1	R-G2	R-D1 (Drone)	R-K1	Tool
------	------	------	--------------	------	------

	(Humanoid)	(Grower)		(Kitchen)	Required
Harvest Tomato	Yes	Yes	No	Yes	Gripper-Cutter Toolhead
Measure Soil pH	Yes	Yes	No	No	Soil Probe Sensor
Aerial Survey	No	No	Yes	No	High-Res Camera Payload
Prune Plant	Yes	Yes	No	No	Precision Pruner Toolhead
Process Vegetables	Yes	No	No	Yes	Knife/Chopper Toolhead
Transport Bin	Yes	Yes	No	No	Heavy Gripper Attachment
Inspect for Pests	Yes	Yes	Yes	No	Magnification Camera
Apply Foliar Spray	Yes	Yes	Yes	No	Spray Nozzle Toolhead

D. Safety Checklists (kitchen, towers, myco, electrical, water)

These are starter checklists for ensuring the safe operation of key modules. Users will be prompted to complete these during module setup and for periodic maintenance missions.

General Electrical Safety:

- ☐ All electrical connections are housed in waterproof enclosures.
- ☐ All circuits connected to water systems are protected by a Ground-Fault Circuit

Interrupter (GFCI).

- ☐ Cables are routed to prevent tripping hazards and are protected from physical damage.
- ☐ System has a clearly marked and accessible emergency power-off switch.

Hydroponic/Aeroponic Tower Safety:

- ☐ Reservoir lid is secure to prevent access by children/pets and to block light (preventing algae).
- ☐ Water pump is fully submersible and correctly installed.
- ☐ Tubing and connections are secure and free from leaks.
- ☐ Nutrient solutions are stored in clearly labeled, food-safe containers, away from children/pets.

Mycology (Mushroom Room) Safety:

- ☐ The room has adequate ventilation to prevent CO2 buildup.
- ☐ All surfaces are non-porous and can be easily sanitized.
- ☐ A dedicated sanitation protocol is in place for entering/exiting the room to prevent contamination.
- ☐ If using a pressure cooker for sterilization, it is operated according to manufacturer instructions.

E. Seed Saving Protocols & Viability Tables

Basic Seed Saving Protocols:

1. **Select Healthy Plants:** Always save seeds from your healthiest, most productive, and best-tasting plants. This selects for good genetics.
2. **Ensure Proper Pollination:**
 - **Self-Pollinating (e.g., Tomatoes, Beans, Peas):** These are easiest. You can save seeds from one plant.
 - **Cross-Pollinating (e.g., Squash, Corn, Cucumbers):** These require pollen from another plant. To save pure seed, you must either isolate the plants (by distance or bagging flowers) or hand-pollinate.
3. **Harvest at Full Maturity:** Let the fruits (for wet seeds) or pods (for dry seeds) fully ripen on the plant, often past the point where you would normally eat them.
4. **Processing:**
 - **Dry Seeds (e.g., Beans, Lettuce, Radish):** Allow seed pods or heads to dry

completely on the plant. Shell the seeds and winnow away the chaff.

- **Wet Seeds (e.g., Tomatoes, Cucumbers):** Scoop out the seeds and pulp into a jar. Add a little water and let it ferment for 2-4 days. This process removes the germination-inhibiting gel around the seed. Skim off the mold, rinse the clean seeds thoroughly, and spread them on a non-stick surface to dry completely.

5. **Storage:** Store clean, completely dry seeds in a cool, dark, and dry place. A labeled paper envelope inside a sealed glass jar is ideal.

Seed Viability Table (Estimated):

Plant	Average Viability (Years)
Beans	3 - 4
Carrots	3
Corn	2
Cucumber	5
Lettuce	4 - 5
Onions	1 - 2
Peas	3
Peppers	2 - 3
Radish	4 - 5
Spinach	3 - 5
Squash	4
Tomatoes	4 - 7

F. Data Schemas (full JSON)

This section contains the full JSON schema definitions for all data objects.

JSON

```
{
  "definitions": {
    "BaseObject": {
      "type": "object",
      "properties": {
        "id": { "type": "string", "description": "Content ID (CID) of the object." },
        "type": { "type": "string", "description": "The type of the object." },
        "created": { "type": "string", "format": "date-time" },
        "updated": { "type": "string", "format": "date-time" },
        "by": { "type": "string", "description": "DID of the creator." },
        "sig": { "type": "string", "description": "JWS signature of the object." }
      },
      "required": ["id", "type", "created", "updated", "by", "sig"]
    },
    "CoreObjects": {
      "Space": {
        "allOf": [
          {
            "properties": { "name": { "type": "string" }, "zones": { "type": "array", "items": { "type": "string" }, "description": "Array of Zone CIDs." } } }
        ],
        "Anchor": {
          "allOf": [
            {
              "properties": { "name": { "type": "string" }, "geom": { "$ref": "#/ArkObjects/Place/properties/geom" }, "linkedCards": { "type": "array", "items": { "type": "string" } } }
            }
          ]
        }
      },
      "Module": {
        "allOf": [
```

```
    "properties": { "name": { "type": "string" }, "moduleType": { "type": "string" }, "placeId": { "type":  
"string" } }  
  },  
  "Plant": {  
    "allOf": [  
      "properties": { "taxonId": { "type": "string" }, "placeId": { "type": "string" }, "sowDate": { "type":  
"string", "format": "date" } }  
    ],  
    "Card": {  
      "allOf": [  
        "properties": { "cardType": { "enum": ["photo", "pdf", "bookPage", "map", "clip", "note"] },  
"contentCid": { "type": "string" }, "ocrText": { "type": "string" } }  
      ],  
      "Scenario": {  
        "allOf": [  
          "properties": { "name": { "type": "string" }, "description": { "type": "string" } }  
        ],  
        "Mission": {  
          "allOf": [  
            "properties": { "title": { "type": "string" }, "status": { "enum": ["todo", "inprogress", "done"] },  
"proof": { "type": "array", "items": { "type": "string" } } }  
          ],  
          "SeedBank": {  
            "allOf": [  
              "properties": { "seedLots": { "type": "array", "items": { "type": "string", "description": "Array of  
SeedLot CIDs." } } }  
            ],  
          },  
          "RoboticsObjects": {  
            "Robot": {  
              "allOf": [  
                "properties": { "name": { "type": "string" }, "model": { "type": "string" }, "status": { "enum":  
["idle", "working", "charging", "error"] } }  
              ],  
              "Capability": {  
                "allOf": [  
                  "properties": { "name": { "type": "string" }, "description": { "type": "string" } }  
                ],  
                "Task": {
```



```

    "allOf":;,
    "properties": { "command": { "type": "string" }, "params": { "type": "object" } }
  },
  "Routine": {
    "allOf":;,
    "properties": { "name": { "type": "string" }, "tasks": { "type": "array", "items": { "type": "string" } }
  }
  },
  "Policy": {
    "allOf":;,
    "properties": { "zoneId": { "type": "string" }, "rule": { "type": "string" } }
  }
  },
  "TowersObjects": {
    "Tower": {
      "allOf": [{ "$ref": "#/CoreObjects/Module" }],
      "properties": { "towerType": { "enum": } }
    },
    "ResMix": {
      "allOf":;,
      "properties": { "towerId": { "type": "string" }, "ec": { "type": "number" }, "ph": { "type": "number"
}, "volume": { "type": "number" } }
    },
    "HaccpLog": {
      "allOf":;,
      "properties": { "towerId": { "type": "string" }, "action": { "type": "string" }, "result": { "type":
"string" } }
    }
  },
  "ArkObjects": {
    "ArkObject": { "$ref": "#/definitions/BaseObject" },
    "Taxon": {
      "allOf":;,
      "properties": {
        "dwc": { "properties": { "scientificName": { "type": "string" }, "rank": { "type": "string" },
"vernacularNames": { "type": "array", "items": { "type": "string" } }, "links": { "properties": { "gbif": {
"type": "string", "format": "uri" }, "tpl": { "type": "string", "format": "uri" } } } } },
        "status": { "properties": { "iucn": { "enum": } } }
      }
    }
  }

```

```

    },
    "Place": {
      "allOf":,
      "properties": {
        "name": { "type": "string" }, "kind": { "enum": ["bed", "tower", "pond", "tree", "wildpatch",
"room"] },
        "geom": { "properties": { "type": { "enum": ["Polygon", "Point"] }, "coordinates": { "type": "array"
} } },
        "privacy": { "properties": { "loc": { "type": "string" }, "share": { "enum": ["L0", "L1", "L2", "L3"] }
} }
      }
    },
    "Observation": {
      "allOf":,
      "properties": {
        "taxonId": { "type": "string" }, "placeId": { "type": "string" }, "when": { "type": "string", "format":
"date-time" },
        "what": { "properties": { "stage": { "enum": ["seed", "seedling", "veg", "flower", "fruit",
"dormant"] }, "phenophase": { "type": "array", "items": { "type": "string" } } } },
        "evidence": { "type": "array", "items": { "properties": { "kind": { "enum": ["photo", "audio", "log"]
}, "cardId": { "type": "string" } } } },
        "confidence": { "properties": { "self": { "type": "number" }, "peer": { "type": "number" }, "auto":
{ "type": "number" } } },
        "attestations": { "type": "array", "items": { "properties": { "by": { "type": "string" }, "type": {
"enum": ["peer_review", "expert_review", "robot_scan"] }, "score": { "type": "number" } } } }
      }
    },
    "SeedLot": {
      "allOf":,
      "properties": {
        "taxonId": { "type": "string" }, "source": { "enum": ["garden", "wild", "swap", "commercial"] },
        "harvest": { "type": "string", "format": "date" }, "viability": { "type": "number" }, "qty": { "type":
"integer" },
        "provenance": { "properties": { "bioregion": { "type": "string" }, "custodian": { "type": "string" }
} },
        "restrictions": { "properties": { "share": { "enum": ["public", "mesh", "custodial"] } } }
      }
    },
    "Recipe": {

```

```

    "allOf":,
    "properties": {
      "name": { "type": "string" }, "scope": { "enum": ["tower", "bed", "mushroom", "preserve"] },
      "stages": { "type": "array", "items": { "properties": { "name": { "type": "string" }, "days": {
"type": "integer" }, "targets": { "properties": { "ec": { "type": "number" }, "ph": { "type": "number" } }
} } } },
      "haccp": { "properties": { "required": { "type": "boolean" } } }
    }
  },
  "Practice": {
    "allOf":,
    "properties": {
      "kind": { "enum": ["grow", "medicine", "preserve", "build"] }, "title": { "type": "string" }, "steps":
{ "type": "array", "items": { "type": "string" } }, "media": { "type": "array", "items": { "type": "string" }
},
      "licenses": { "properties": { "use": { "enum": } } }
    }
  },
  "SeasonPack": {
    "allOf":,
    "properties": {
      "index": { "properties": { "places": { "type": "array", "items": { "type": "string" } },
"observations": { "type": "array", "items": { "type": "string" } }, "seedLots": { "type": "array", "items":
{ "type": "string" } }, "recipes": { "type": "array", "items": { "type": "string" } }, "practices": { "type":
"array", "items": { "type": "string" } } } },
      "merkleRoot": { "type": "string" }, "scope": { "enum": ["L0", "L1", "L2", "L3"] }
    }
  },
  "Reputation": {
    "allOf":,
    "properties": {
      "claimId": { "type": "string" }, "scores": { "properties": { "accuracy": { "type": "number" },
"safety": { "type": "number" }, "usefulness": { "type": "number" } } },
      "justifications": { "type": "array", "items": { "type": "string" } }
    }
  },
  "AccessPolicy": {
    "allOf":,
    "properties": {

```

```

    "rules": { "type": "array", "items": { "properties": { "objType": { "type": "string" }, "scope": {
"type": "string" }, "allow": { "type": "array", "items": { "type": "string" } }, "shareMax": { "enum":
["L0", "L1", "L2", "L3"] } } } }
  }
}
}
}

```

G. API Reference (REST/gRPC/Events)

Local API (Conceptual Endpoints)

These endpoints would be exposed on the user's local device for integration with other tools or scripts.

- GET /twin/state: Retrieves the full current state of the user's digital twin (HDT and EDT) as a JSON object.
- POST /missions: Creates a new user-defined mission.
- PUT /missions/{id}/complete: Marks a mission as complete, optionally with proof-of-work data.
- GET /deck/cards: Lists all cards in the user's knowledge deck.
- POST /deck/cards: Adds a new card to the deck.
- POST /sim/run: Initiates a simulation run with a given scenario.

Mesh API (P2P Events)

These are the primary events broadcast over the peer-to-peer network for data synchronization and governance.

- **ark:publish:** Broadcasts a new data bundle (e.g., SeasonPack) to a specific governance scope (e.g., L1 Mesh DAO).
 - **Payload:** { "object": "SeasonPack", "cid": "bafy...", "scope": "L1" }
- **ark:attest:** Broadcasts a peer review or verification of another user's claim.
 - **Payload:** { "claimId": "cid_of_observation", "type": "peer_review", "score": 0.9 }
- **ark:adopt:** Logs the action of a user importing a pattern from a higher-level commons into their local space.
 - **Payload:** { "patternId": "cid_of_practice", "source": "L2:BioregionGuild" }

- **ark:revoke:** Broadcasts a tombstone message to invalidate a previously published object.
 - **Payload:** { "tombstoneFor": "cid_of_season_pack", "reason": "Data entered in error." }

H. File Formats & Interop (CAD/GeoJSON/OSM)

The system prioritizes open and standard file formats to ensure interoperability and prevent data lock-in.

- **Spatial Data Import/Export:**
 - **GeoJSON (RFC 7946):** The primary format for importing and exporting vector spatial data (zones, footprints, paths).
 - **KML:** Supported for easy interoperability with Google Earth.
 - **OSM XML:** Supported for importing crowdsourced data from OpenStreetMap.
- **3D Model/CAD Import:**
 - **glTF:** The preferred format for importing 3D models of modules or robots due to its efficiency for web and mobile rendering.
 - **DXF (simplified):** Supported for importing 2D layouts from architectural or CAD software.
- **Data Log Export:**
 - **CSV (Comma-Separated Values):** All time-series data (sensor readings, harvest logs, etc.) can be exported as CSV for analysis in external tools like spreadsheets or data analysis software.
- **Knowledge Deck Export:**
 - **Markdown:** Synthesized Practice and Recipe objects can be exported as Markdown files, preserving text formatting and links to media.

I. Printables (offline cards, QR links, one-pagers)

To support low-tech and offline participation, the system includes a "Print Center" that can generate:

- **Observation Cards:** A5-sized cards with fields for recording a single Observation (Species, Location, Date, Notes, Sketch). These can be filled out in the field and

entered into the system later.

- **QR Code Labels:** Weatherproof labels with QR codes linked to specific Anchors in the digital twin. Scanning the code with a phone opens a simple web form to log data for that specific anchor (e.g., "Watered this plant," "Harvested 5 tomatoes").
- **One-Pagers:** Printable summaries of Recipes or Practices, such as a "Composting Quick Guide" or a "Tower Sanitation Checklist," which can be posted in the relevant physical location.
- **Starter Kit Guides:** One-page, printable how-to guides for the activities in the "Starter Ark Kit," such as how to perform a soil test or start a seed library.

J. Glossary of Indigenous & Local Custodial Terms

This appendix is intentionally left as a framework to be co-developed with custodian communities. Its purpose is to create a shared, respectful vocabulary for knowledge that is under local or Indigenous guardianship.

- **Purpose:** To define and explain the terms used in Custodial Licences and AccessPolicy objects, ensuring that all network participants understand the specific cultural protocols and restrictions associated with certain types of knowledge.
- **Development Process:** The glossary will be built collaboratively. When a new Custodian group joins the Ark, they will be invited to define the terms relevant to their community's knowledge. These definitions will be ratified by the L2 Bioregion Council and added to this living document.
- **Example Entry (Conceptual):**
 - **Term:** *Rahui* (as defined by a specific Māori community)
 - **Definition:** A temporary ritual prohibition or restriction placed on a specific area or resource to allow for its regeneration and protection. Data marked with a Rahui license may be temporarily inaccessible or have its location obscured.

K. Bibliography & Source Attribution

This document was synthesized from a wide range of research. Key sources and foundational concepts include:

1. **Regenerative Framework & Permaculture:** Principles derived from the work of Bill Mollison and David Holmgren, focusing on the core ethics of Earth Care, People Care, and Fair Share, and the shift from sustainability to active regeneration .
2. **Systems Thinking:** Application of systems thinking principles (interconnectedness, feedback loops, stocks, flows) to model the self-and-place system .
3. **Digital Twin Architecture:** Concepts for Human and Environmental Digital Twins, including multi-layer architectures, data aggregation, and simulation, are drawn from emerging research in personalized healthcare and smart manufacturing .
4. **Personal & Health Data:** Methodologies for collecting biometric, psychological, and behavioral data are informed by the Quantified Self movement and established digital psychological assessment tools .
5. **Environmental & Citizen Science Data:** Data collection protocols are based on citizen science methodologies for biodiversity audits, soil testing, and water quality monitoring .
6. **Gamification & Behavior Change:** The design of the gamification engine is based on Self-Determination Theory and research into serious games for learning and sustainable behavior change .
7. **Decentralized Technology:** The architecture for the Fractal Ark utilizes principles of P2P networking, CRDTs (Automerger/Y.js), DIDs/VCs, blockchain for data integrity

Works cited

1. Regenerative Systems through Permaculture - Number Analytics, accessed August 12, 2025, <https://www.numberanalytics.com/blog/regenerative-systems-through-permaculture>
2. Regenerative Systems Thinking → Term, accessed August 12, 2025, <https://lifestyle.sustainability-directory.com/term/regenerative-systems-thinking/>
3. Regenerative System Thinking → Term - Lifestyle → Sustainability Directory, accessed August 12, 2025, <https://lifestyle.sustainability-directory.com/term/regenerative-system-thinking/>
4. Ethical Issues of Digital Twins for Personalized Health Care Service ..., accessed August 12, 2025, <https://pmc.ncbi.nlm.nih.gov/articles/PMC8844982/>
5. (PDF) Human Digital Twins in Personalized Healthcare: An ..., accessed August 12, 2025, https://www.researchgate.net/publication/389916943_Human_Digital_Twins_in_Pe

[Personalized Healthcare: An Overview and Future Perspectives](#)

6. Behavior Change through Gamification - Number Analytics, accessed August 12, 2025,
<https://www.numberanalytics.com/blog/behavior-change-through-gamification>
7. Systems Thinking, Critical Thinking, and Personal Resilience - Resilience.org, accessed August 12, 2025,
<https://www.resilience.org/stories/2018-05-24/systems-thinking-critical-thinking-and-personal-resilience/>
8. Systems Thinking | THRIVE Project, accessed August 12, 2025,
<https://thrivabilitymatters.org/thrive-framework/thrive-framework-systems-thinking/>
9. Serious games as a tool to engage people | Earth Sciences New ..., accessed August 12, 2025,
<https://niwa.co.nz/climate-and-weather/serious-games-tool-engage-people>