STRANDS

A Decentralised Bulwark Against Techno-Feudalism
and the Path to Equitable Income

WHITEPAPER V6.0

PT Meta Fin Tek · Metafintek.xyz
March 2026
12 Chapters · ~21,000 words
Classification: Internal, Founder Review

We are not Left. We are not Right. We are not the Centre. We are the Decentre.

Table of Contents

PART I: THE THESIS

Chapter 1 Strands: Reasons Why

Chapter 2 Strands the Game

PART II: THE ECOSYSTEM

Chapter 3 My Maits

Chapter 4 EveryWear

Chapter 5 Layer U and the A.R.E.

PART III: THE CHAIN

Chapter 6 $KREDS Tokenomics

Chapter 7 Strands Blockchain

PART IV: OPERATIONS

Chapter 8 Governance, Privacy & Compliance

Chapter 9 Roadmap

Chapter 10 Team

PART V: APPENDICES

Chapter 11 Legal & Regulatory Disclaimer

Chapter 12 Appendices

PART I: THE THESIS

CHAPTER 1

Strands: Reasons Why

The Thesis

The digital economy is not broken. It is working exactly as designed. It extracts attention, monetises behaviour, centralises ownership, and redistributes none of the value it captures back to the people who generate it. The platforms that dominate Web 2.0 did not accidentally become surveillance engines. They were architected that way. Every scroll, every click, every second of dwell time feeds a system that treats human attention as raw material and returns nothing but algorithmic manipulation.

This is not a new observation. Entire academic disciplines have formed around it. But observation without action is commentary, and commentary does not build exits.

Sarwar Uddin's *Architects: Building the Exit to Techno-Feudalism* articulates the structural problem and the structural response. The book argues that the concentration of digital power into platform monopolies has created a new feudalism: users generate value, platforms capture it, and the asymmetry compounds with every network effect. The exit is not regulation. Regulation negotiates terms within the existing architecture. The exit is new architecture: systems designed from first principles to redistribute value, protect sovereignty, and align incentives between the people who build, the people who participate, and the people who invest.

Strands is the applied response to that thesis.

The Problem Space

Three forces converge to make the current digital economy unsustainable for the people inside it.

Attention extraction without compensation. The global digital advertising market exceeds $600 billion annually. The raw material powering that market is human attention. The people providing that attention receive precisely zero direct compensation. They receive "free" services funded by the sale of their behavioural data to third parties whose interests are structurally misaligned with theirs. This is not a value exchange. It is value extraction dressed in convenience.

Identity systems owned by corporations. Digital identity today is a collection of platform accounts, each controlled by entities whose survival depends on keeping users inside their walled gardens. A person's digital history, their social graph, their creative output, their professional reputation, all of it lives on servers they do not control, governed by terms of service they did not negotiate, and revocable without recourse. Sovereign identity remains a theoretical concept for the vast majority of internet users.

Blockchain's failure to onboard. Web 3.0 promised to solve both problems. Decentralised networks would return data sovereignty to individuals. Token economies would redistribute value to participants. The technology works. The onboarding does not. After more than a decade of development, blockchain adoption remains confined to a technically literate minority. The barrier is not capability. It is accessibility. The average person does not want to manage seed phrases, navigate gas fees, or parse tokenomics whitepapers. They want to use products that work.

Strands: Reasons Why

Strands exists because the exit from techno-feudalism cannot be built from inside the systems it seeks to replace. It requires new infrastructure: a unified ecosystem where attention is compensated, identity is sovereign, economic participation is rewarded, and governance is distributed. Not as theoretical features on a roadmap, but as operational realities that users experience before they are asked to understand the architecture underneath.

The Strands ecosystem is not a single product. It is a sequenced deployment of interdependent systems, each validated before the next activates. The game proves the model. The attention economy proves the revenue. The token proves the internal economy. The chain proves the sovereignty. Nothing launches on faith. Everything launches on evidence.

This sequencing is the defining structural decision of the project. Most Web 3.0 ventures launch tokens first and hope adoption follows. Strands inverts that logic entirely. The blockchain layer is the last component to activate, not the first. It earns its place by being needed, not by being available.

The ecosystem comprises several core modules, each solving a distinct failure in the current landscape: a personalised companion layer (My Maits) that provides adaptive AI interaction within the game and wider ecosystem; a unified client (EveryWear) that evolves from lightweight app to sovereign personal platform; a spatial governance and attention economy infrastructure (Layer U) that deploys consent-based revenue models into real-world urban environments; a native network token ($KREDS) that internalises value circulation so that earning, exchange, and coordination occur within the Strands economy rather than leaking outward; and a privacy-preserving blockchain (Strands Chain) that provides sovereign infrastructure when the ecosystem is ready to govern itself.

These are not co-equal modules launching simultaneously. They are stages in a validation sequence. Each one activates when the prior stage has proven its thesis. The chapters that follow will detail each in full. Chapter 1 establishes why they exist at all.

Strands the Game: Reasons Why

Strands the Game is the practical onboarding layer for the wider ecosystem. It uses Telegram as the initial distribution surface because Telegram collapses friction at the exact point where most blockchain products fail: identity, wallet adjacency, payments, and reach. Through Telegram Mini App deployment, integrated payment rails, and access to stablecoin and fiat monetisation paths, the game can recruit users before they ever need to understand the underlying chain architecture. More importantly, the game does not merely attract players, it educates them through participation, gradually introducing the logic of decentralised ownership, consent-based economics, and cooperative systems until users are not just customers or spectators, but conscious participants in a decentralised movement.

The game is a standalone product. It generates revenue through nine distinct streams without any blockchain dependency. It has its own narrative, its own economy, its own retention mechanics, and its own reasons to exist. The game works if the blockchain never launches. This is not a hedge. It is a design constraint. Any system that requires token speculation to survive is not an exit from techno-feudalism. It is techno-feudalism wearing a decentralised mask.

Strands the Game is also the proof layer for everything that follows. The in-game economy tests attention redistribution mechanics before they deploy into the real world. The in-game governance tests DAO decision-making before it carries real capital. The in-game spatial systems test volumetric content interaction before physical cities adopt the infrastructure. Every mechanic inside the game maps to an operation outside of it. The game is not a marketing surface for the blockchain. The game is the laboratory that determines whether the blockchain deserves to exist.

Chapter 2 will detail this argument in full: the game's design philosophy, its narrative architecture, and the specific mechanisms through which play becomes education.

Sequential Validation

The ordering is deliberate and non-negotiable:

- **Game (Fiat)**; Prove the product works. Prove retention. Prove revenue. Prove that people will engage with decentralised principles when those principles are embedded in compelling experience rather than presented as ideology.
- **Layer U (Attention Economy)**; Prove the attention redistribution model works. Prove that advertisers will pay for consent-based engagement. Prove that users will opt in when the economics are transparent and the compensation is direct.
- **$KREDS (Network Economy)**; Prove the internal economy works. Launch only when there are enough participants to create genuine utility, not speculative liquidity. $KREDS is not an in-game currency. It is the native economic token of the Strands Network, enabling users to earn, exchange, and retain value generated through participation in the ecosystem. It activates after the fiat model has validated.
- **Strands Chain (Infrastructure)**; Prove the architecture works at scale. Strands initially leverages Telegram and TON infrastructure to eliminate wallet friction and app-store dependency, while preserving a migration path to sovereign chain infrastructure. The native chain activates only when the ecosystem demands sovereignty over its own protocol.

No phase activates until the prior phase validates. This is the difference between building an ecosystem and launching a token with a roadmap attached.

Why Now

Three convergences make this the correct moment to build:

Distribution and payments have converged. Telegram now collapses onboarding, payments, and distribution into one surface. TON blockchain, Mini Apps, integrated payment rails, and bio-authentication across 900 million monthly active users create an onboarding path that did not exist two years ago. The friction cost of getting a wallet adjacent to a user has collapsed. Strands leverages this infrastructure rather than competing with it.

AI companions are now good enough to matter. Large language models and agentic frameworks have crossed the threshold required to make adaptive, personality-driven NPCs functional at consumer scale. Companions with genuine behavioural variation, contextual memory, and coherent interaction are now buildable, not theoretical. The game's companion layer is designed to deploy this capability from day one.

Consent-based attention exchange can now be operationalised. The mechanics required to measure attention, verify consent, deliver contextual content, and settle payment in real time now exist as deployable infrastructure rather than research proposals. The gap between "users should be paid for their attention" and "here is the pipeline that pays them" has closed. Strands is built to occupy that gap.

What This Whitepaper Covers

The chapters that follow detail the technical architecture, economic models, and deployment strategies for each component of the Strands ecosystem. The structure mirrors the sequential validation model:

- **Chapter 2** details Strands the Game: architecture, design philosophy, and onboarding logic in practice
- **Chapter 3** covers My Maits, the personalised companion layer
- **Chapter 4** specifies the EveryWear unified client evolution
- **Chapter 5** presents Layer U and the Attention Redistribution Engine
- **Chapter 6** documents $KREDS tokenomics
- **Chapter 7** describes the Strands Blockchain infrastructure
- **Chapter 8** covers governance, privacy, and compliance
- **Chapter 9** presents the phased roadmap with validation gates
- **Chapters 10-12** cover the team, legal framework, and technical appendices

We designed Strands to grow through continuous validation, participation, and shared consensus, not through extraction, but through the creation of value in play, in experience, and in revenue shared across the network.

The exit from techno-feudalism is not revolution, it is evolution.

CHAPTER 2

Strands the Game

Design Philosophy

Strands the Game is not a blockchain game. It is a game that happens to teach blockchain principles. The distinction is structural, not semantic. Blockchain games ask users to understand wallets, tokens, and chain mechanics before they can play. Strands asks users to play, and embeds the logic of decentralised systems into the play itself. By the time the blockchain layer activates, the player already understands consent-based economics, cooperative governance, and sovereign data ownership because they have been practising all three inside a game world that makes those concepts feel like gameplay, not curriculum.

The game is built on three design commandments:

Diegetic constraint. Every interaction presents as a believable interface within the game world. The player never sees a tutorial screen, a terms-of-service popup, or a progress bar labelled "blockchain education." They see a desktop operating system, a chat client, a propaganda broadcast, a cipher puzzle. The game world is the interface. The interface is the game world. Nothing breaks the container metaphor.

Discovery over direction. The player is never told what to do. They overhear, decode, choose, and discover. The feeling of agency is real even though the experience is tightly authored. Players should feel clever for figuring things out, never herded. This applies equally to narrative progression and to the economic systems: the player discovers how attention exchange works by participating in it, not by reading an explainer.

Diegetic calibration. The onboarding experience is designed as narrative play rather than overt assessment. The system, game, and narrative synchronise to the way each player engages, responds, and plays, allowing the experience to adapt to them without ever breaking the fiction. Binary choices, puzzle-solving behaviour, and voice interaction all feed an adaptive calibration system that personalises the experience in real time. The player-facing term is "Synch Profiling," presented as "the system watching how you respond." The calibration framework is internal. The player experience is narrative.

The World: MetaXity1, Year 555

The game is set 555 years in the future inside MetaXity1, a continental-scale pyramidal arcology stretching across Southeast Asia. One billion people live inside a vertically stratified structure where altitude is status and depth is desperation. The sovereignty engine that governs this world, SOVcorp, controls water, food, compute, logistics, surveillance, identity, and labour. Citizens exist inside full-body augmentation chassis called Blanks. A resistance network operates in the infrastructure gaps that 555 years of institutional ossification have created.

The official history blames a rogue superintelligence for the collapse that preceded the pyramid. The real history is more complicated, and the game's narrative arc is built around the player uncovering that complication. The propaganda arm, Proper Gander, maintains the official story through broadcast control and managed spectacle.

This setting is not decorative worldbuilding. It is a functional simulation of techno-feudalism at terminal maturity. SOVcorp's attention extraction maps to the surveillance advertising economy. Its administered subsistence system maps to platform lock-in. Its control of identity and embodiment maps to corporate ownership of digital selfhood. The in-world resistance network maps to decentralised cooperative infrastructure. The player does not need to understand these mappings consciously. The game teaches through parallel experience.

Two-Phase Architecture

The game operates in two distinct phases that share the same technical infrastructure but deliver fundamentally different experiences.

Phase A: The Guided Run

A sequenced, narrative-driven experience disguised as a desktop operating system. The player arrives at a simulated desktop environment. Apps unlock at specific narrative beats. Each interaction feeds into the next. Everything builds toward a single climactic signal event.

The experience proceeds through a structured sequence: identity registration, a chat narrative with embedded decision points, voice capture, cipher puzzles, skill-based mini-games, and a progressive signal reconstruction that assembles across every completed step. The reconstruction resolves into a complete signal event: a propaganda broadcast with anomalous counter-narrative bleeding through, culminating in a moment where the player hears their own recorded voice spoken through the game's temporal bridge logic.

Phase A ends with a hard stop and a promise: the 3D game client is coming.

Phase B: The Personalised Desktop

After Phase A, the guided rails come off. The player now owns a personalised base-builder desktop, the same window manager and app infrastructure, but freely explorable with their own progression cadence. The desktop visually reflects the player's personality profile, their choices, their progression history, and their avatar. No two desktops look the same.

Phase B introduces the long-term engagement systems: a calendar-gated progression track (Bridge Levels), an episodic propaganda leak schedule (Season 0 Proper Gander broadcasts, each containing signal hijacks with deeper lore), hidden app reveals tied to progression milestones, a crafting and equipment system, music generation tools, and the player's first encounters with the economic mechanics that will later map to the real-world Strands economy.

Phase B culminates in Portal activation, the transition point from the 2D desktop environment into the full 3D game client set inside MetaXity1.

Adaptive Calibration

The onboarding pipeline captures interaction patterns through diegetic play rather than explicit testing interfaces, allowing the system to calibrate each player's experience through their natural engagement with the narrative.

Synch Profiling. Four binary choices embedded in the Phase A chat narrative map to personality axes. The system records not just which option the player selects, but how they engage: response latency, engagement depth, whether they re-read content before choosing. The output is a personality seed that feeds downstream quest personalisation, NPC dialogue variation, and content delivery weighting. The player never sees a type label. They see "your signal pattern."

Neuro-Kinetic Quotient (NKQ). Four cipher puzzles and a memory pattern-matching game capture processing speed, pattern recognition, and working memory through gameplay. The NKQ baseline calibrates difficulty scaling across the entire game: puzzle complexity, quest pacing, and cognitive load are adapted to the individual player through the narrative framing of "signal reconstruction quality."

Voice capture. The player records a spoken phrase during onboarding. This voice sample serves as an identity anchor within the game's temporal bridge narrative. At higher progression states, the game's AI-driven communications partially resonate with the player's own vocal qualities, creating an uncanny recognition effect that deepens engagement without breaking the fiction.

All calibration data persists across phases. The web-based Phase A experience carries forward into the full game client, preserving the player's Synch Profile seed, NKQ baseline, and voice sample as foundational inputs to every system they will encounter.

The In-Game Economy

The game operates a dual-currency fiat economy with zero blockchain dependency at launch.

SOVComp is the corporate currency. It is earned through compliant participation in SOVcorp's economy: sanctioned work, approved quests, compliance milestones. SOVComp teaches the player how dependency, rationing, and permissioned access operate inside managed systems. It is akin to UBI, a corporate or state sanctioned subsistence currency designed to sustain basic participation while preserving dependency on the system that issues it.

GridScrip is the resistance currency. It is earned through unsanctioned participation: parallel trade, counter-system activity, resistance network engagement. GridScrip teaches the player how networked economies emerge outside centralised control. It is the currency of the system being subverted. It is analogous to the role $KREDS later plays in the real-world Strands ecosystem: a decentralised, usable currency within a parallel economy.

Neither currency has real-world value. The player's balance between SOVComp and GridScrip reflects their factional alignment and shapes their access to different content, vendors, equipment, and narrative paths.

The point is not to create a literal conversion between in-game and real-world assets. The point is to let players experience the difference between administered subsistence and participatory value creation before those distinctions appear in the live network economy.

Signal Reclamation: Training the Attention Economy

Signal Reclamation is the in-game mechanic that trains players for the real-world Attention Redistribution Engine (A.R.E.) before the A.R.E. itself activates.

Inside the game, Signal Reclamation quests present as lore-spreading narrative content delivered in the Proper Gander aesthetic: propaganda-styled broadcasts that expand the player's understanding of the game world. The player opts in, watches, and earns in-game currency. The mechanic follows the same pipeline that the real-world A.R.E. will use: explicit consent, contextual delivery, attention measurement, and direct reward.

At Phase 1, there are no external ad network integrations. The content is produced in-house as narrative material. The value is not advertising revenue. The value is behavioural proof: Signal Reclamation generates data on consent-tier adoption, attention dwell time, opt-in frequency, and reward sensitivity. This data directly de-risks the real-world A.R.E. deployment documented in Chapter 5.

The player learns the attention exchange pattern through play. By the time the live A.R.E. activates with real advertisers and real revenue splits, the user base already understands the mechanic, trusts the consent model, and has months of behavioural history demonstrating willingness to participate.

Monetisation

The game's commercial viability does not depend on token speculation or on-chain economy activation. Revenue is generated through nine streams:

- Founders Access Passes (collectible digital assets with personalisation features)
- Generative tool credits (music creation, texture generation)
- Cosmetic avatar items and equipment
- Premium crafting materials
- Signal Reclamation (diegetic attention exchange, detailed above)
- Season passes for episodic content
- Offerwall integration (post-validation)
- Community marketplace transaction fees
- Brand partnership and sponsored in-game content

The core revenue model works independently of live blockchain dependency. The game is designed to sustain itself as a fiat-economy product first. Additional economic layers activate later as the wider Strands ecosystem matures.

Proof-of-Concept Mapping

Every major in-game mechanic maps to a real-world Strands ecosystem operation:

Billboard hijacking (SIGOPS Terminal) maps to XR ad placement in physical locations. **Signal Tower capture** maps to XR infrastructure node deployment. **Zone governance voting** maps to DAO decision-making on policy and revenue distribution. **Volumetric content placement** maps to spatial advertising in real-world environments. **Attention measurement** (dwell time, interaction depth, consent tracking) maps to the A.R.E. consent pipeline.

A note on terminology: "Layer U" appears both inside the game world and in the real-world Strands ecosystem. Inside the game, Layer U is the diegetic resistance network, the decentralised shadow infrastructure that grew in the gaps of SOVcorp's control. In the real world, Layer U is the spatial governance and attention economy infrastructure documented in Chapter 5. The naming is intentional: the in-game fiction teaches the player what the real-world system does, using the same language so the conceptual bridge is seamless.

The game generates interaction pattern data across six categories that de-risk real-world deployment: attention economics, consent tier adoption rates, volumetric interaction patterns, governance behaviour, dual-economy circulation dynamics, and creator economy engagement. All data is consent-gated, anonymised, and aggregated. The game is the laboratory that determines whether the blockchain deserves to exist.

Deployment Architecture

The game deploys as a standalone application at a dedicated subdomain, fully decoupled from the marketing site. The Desktop OS supports multiple concurrent app windows, persistent audio, voice recording, generative AI integration, and avatar rendering. This is a platform, not a page.

Initial distribution is through Telegram Mini App deployment, leveraging Telegram's existing infrastructure for authentication, identity, and payment rails. The game client evolves through the phases outlined for EveryWear in Chapter 4: Telegram Mini App, browser-based client, WebGL 3D client, and ultimately native distribution.

PART II: THE ECOSYSTEM

CHAPTER 3

My Maits

What My Maits Are

My Maits are adaptive companion agents assembled from modular Trait Shards. These shards are the composable dNFT elements of the system, encoding different aspects of personality, capability, behaviour, knowledge, and identity. A Mait is not a single static token. It is the compiled outcome of shard composition: a unique entity whose character, skills, and appearance emerge from the specific combination of shards that define it.

This compositional architecture is the central technical principle of the system. Traditional digital companions are monolithic: a single model, a single personality, a single set of capabilities. Trait Shards make companion design modular. Each shard is an independent, tradable, evolvable dNFT that governs one dimension of the companion's identity. The player or operator assembles shards into a Mait as a builder assembles components into a system. Change a shard, change the companion. Trade a shard, reshape the personality. Evolve a shard, deepen a capability. The companion is never locked. It is always composable.

My Maits operate across two domains. Inside the game, the shard composition architecture is emulated diegetically: players collect, trade, and compile shards to create companions suited to their class stack, strategy, and style of play. The act of assembling a Mait from shards teaches modular intelligence design through play rather than instruction. Outside the game, the same compositional logic powers a B2B agentic platform already deployed in production through Fainance Ltd. (UK), where Maits serve as FCA-compliant financial guidance agents.

The core compositional architecture is shared in both cases. Its orchestration, permissions, and presentation adapt to context.

Role Within the Game

Inside Strands the Game, My Maits are the personalised companion layer that makes the player's experience feel more responsive, contextual, and alive.

A player's Mait operates as an adaptive companion calibrated to their Synch Profile and NKQ baseline. It provides contextual guidance during quests, delivers faction-aligned intelligence, assists with economic decisions across the SOVComp and GridScrip economies, and surfaces narrative connections the player might otherwise miss. The Mait's personality is not generic. It reflects the Trait Shards the player has selected or earned, creating a companion whose behaviour, communication style, and priorities are distinct to that player.

My Maits are defined by composable traits and adaptive behaviour, not by acting as the player's persistence layer. Broader memory and data continuity are handled elsewhere in the Strands stack.

UBComp (Universal Basic Compute) is the in-game resource used to acquire, compile, or refine certain Trait Shards and Mait modifications that are not directly looted through gameplay. This makes companion development an economic decision rather than a purely cosmetic one. Players must choose how to allocate compute toward personalisation, capability growth, and shard assembly.

Diegetic Shard Compilation

The in-game shard system is not cosmetic. It is an educational mechanic.

Inside the game, players encounter Trait Shards as collectible, tradable items earned through quests, faction activity, economic participation, and exploration. Each shard visibly represents a modular component of companion identity: a personality disposition, a skill capability, an aesthetic element, a knowledge domain. The player assembles these shards into their Mait through an in-game compilation interface, experiencing firsthand how modular components combine to produce emergent behaviour.

This is deliberate. The game teaches the logic of composable digital identity through participation rather than explanation. A player who has spent hours selecting, trading, and compiling shards to build a Mait suited to their playstyle has already internalised the principles of modular AI agent design, sovereign digital ownership, and composable identity architecture. They understand these concepts because they have practised them, not because they were told about them.

The diegetic framing ensures that the educational payload never breaks the game's container metaphor. Players do not see a tutorial about dNFT composition. They see a companion workshop inside the game world where shards slot into a build interface and the resulting Mait activates with the compiled personality. The blockchain mechanics are invisible. The experience is play.

When the real-world shard economy activates on-chain in later ecosystem phases, the player base already understands composition, trading, and sovereign ownership because the game has been their training ground.

Trait Shard Architecture

Trait Shards are the modular dNFT components that define a Mait's behaviour. Each shard governs a specific dimension of personality, skill, knowledge, or presentation, and shards can be combined, traded, and evolved to create increasingly specialised companions.

The system is organised across several trait categories:

Personality shards define the Mait's psychological profile, communication style, and decision-making tendencies. The underlying framework maps to sixteen personality archetypes derived from established psychometric research (used internally, never surfaced to the player as labels). These archetypes are structured across four binary axes: extraversion/introversion, sensing/intuition, thinking/feeling, and judging/perceiving. The eight sub-type dimensions provide the backbone of the personality shard system, enabling structured behavioural variation that produces meaningfully different companions. A Mait built with strategic, analytical personality shards will approach problems differently from one built with empathetic, relational shards: different communication registers, different decision heuristics, different priorities in ambiguous situations.

Skill shards define the Mait's functional capabilities. These include financial analysis, creative generation, language tutoring, tactical planning, investigation, crafting assistance, risk assessment, and domain-specific advisory. Skill shards determine what a Mait can do. Personality shards determine how it does it. The combination produces emergent behaviour: a Mait with risk assessment skills and an analytical personality will deliver blunt probabilistic evaluations, while the same skill paired with an empathetic personality will frame the same information as supportive guidance.

Knowledge shards define the Mait's domain expertise. A Mait can be configured with shards covering specific subjects, industries, lore domains, or professional disciplines, expanding its ability to assist within and beyond the game context. Inside the game, knowledge shards might cover factional history, economic strategy, or cipher-breaking methodology. Outside the game, the same architecture supports financial regulation, educational curriculum, or industry-specific advisory.

Aesthetic shards define the Mait's visual presentation. The aesthetic system is deeply granular, covering gender expression, body type, age appearance, skin tone, hair style and colour, eye shape and colour, facial structure, special features, clothing style, accessories, environment integration, and animation behaviour. These attributes function as metadata that feeds into the avatar generation pipeline. The system supports virtually unlimited combinations, ensuring that no two Maits look or feel identical. Example configurations range from cyberpunk rebels with neon-glow mohawks and holographic surroundings to ethereal sages with colour-shifting skin and particle-effect auras. The visual identity of a Mait is as composable as its personality.

Evolution logic. Trait Shards are not static. As a player interacts with their Mait, individual shards can evolve through use, unlocking deeper capability tiers within their domain. A hobby shard might progress from casual familiarity to expert-level knowledge. A personality shard might develop greater nuance in its behavioural expression. Evolution is earned through interaction, not purchased, creating a natural incentive for sustained engagement that compounds the relationship between player and companion.

The compositional design ensures that no single shard defines the Mait in isolation. Identity emerges from the interplay between shards across categories. This is intentional: it mirrors the way real personality, capability, and identity arise from the interaction of multiple dimensions rather than from any single trait.

The B2B Framework

My Maits is not exclusively a game feature. The underlying agentic framework, built by Metafintek, is a general-purpose companion platform applicable across industries.

The production deployment at Fainance Ltd. (UK) demonstrates this. Fainance operates a Mait-powered financial guidance agent that provides FCA-compliant advisory interaction, demonstrating that the Trait Shard architecture can be configured for regulated professional contexts, not just game entertainment. The same compositional logic that lets a player build a tactical companion inside the game lets a financial services firm build a compliance-aware advisory agent for its clients.

The B2B model supports several use cases: financial services compliance, educational tutoring, customer service automation, professional networking assistance, and domain-specific advisory. In each case, the core architecture remains the same. Trait Shards define behaviour. The shard composition model produces agents calibrated to specific professional requirements.

CHAPTER 4

EveryWear

What EveryWear Is

EveryWear is the evolving sovereign interface layer of the Strands ecosystem. It begins as a lightweight distribution surface and matures into the persistent runtime through which memory, wallet, agents, validation, and spatial data are accessed.

It is not a single application. It is not permanently identical to the game client. EveryWear is the interface layer that persists across every phase of the Strands platform, adapting its form as the ecosystem matures while maintaining continuity of identity, data, and function for the user. The game client starts close to the EveryWear surface, then dedicated game experiences split off into their own runtimes while EveryWear remains as the launcher, vault, wallet, agent surface, and continuity shell around them.

This chapter describes the five phases of that evolution, the logic behind each transition, and the core functions EveryWear carries at maturity.

Why It Evolves in Phases

EveryWear ships as an evolving substrate and interface to the Strands world. It begins as the player's entry point through the game, then expands into the broader personal, economic, and spatial runtime of the ecosystem.

Each phase exists because its predecessor has proven a capability and exposed a limitation. The Telegram Mini App proves distribution but cannot carry persistence. The Chromium fork proves persistence but cannot deliver high-fidelity 3D. The game launcher bridges the gap but creates a bloated monolith. Bifurcation separates concerns so the game can scale visually while EveryWear scales functionally. The Agentic OS is the convergence point where sovereign runtime meets spatial computing.

The sequence is not aspirational. Each phase has a clear activation condition, a clear purpose, and a clear handoff to the next.

Phase 1: Telegram Mini App

Purpose: Frictionless distribution and onboarding at maximum reach.

The first EveryWear surface is a Telegram Mini App. It exists because Telegram provides 900 million monthly active users, zero-friction app discovery, native payment rails (TON, Telegram Stars, card via MoonPay), and bio-authentication without requiring users to install anything beyond the messaging client they already use.

At this phase, EveryWear is lightweight by design. It carries the game's onboarding sequence, the initial desktop OS interface, and the fiat payment layer. There is no wallet binding at entry, no blockchain exposure, no token mechanics. A player downloads nothing. They tap a link inside

Telegram and begin playing.

The Mini App proves three things: that the game's onboarding loop converts and retains users, that the fiat economy generates revenue without token dependency, and that the Telegram ecosystem provides a viable distribution channel at scale. These are the activation conditions for Phase 2.

What the Mini App cannot do is persist data beyond the session, host sovereign storage, or run agentic processes. It is a surface, not a runtime. That limitation is what Phase 2 resolves.

Phase 2: Chromium Fork and Agentic Browser

Purpose: Persistence, sovereign data, wallet integration, and agentic browsing.

When the Mini App has proven distribution and retention, EveryWear graduates into a dedicated Chromium-forked browser. This is not merely a browser with game features bolted on. It is a privacy-first sovereign client that carries three capabilities the Mini App could not:

Mymories. The sovereign data vault activates at this phase. Mymories is the persistence layer through which the player's interaction history, preference signals, consent records, and contextual data are stored in player-controlled encrypted storage. It is the data substrate that makes My Maits contextually aware across sessions. In Phase 1, the game could only store transient session state. In Phase 2, EveryWear becomes the player's data home.

Wallet integration. The Blank Sync Ledger, the player's native wallet, activates invisibly at the first purchase threshold, when simple fiat participation graduates into persistent asset ownership and settlement history. The wallet is embedded in EveryWear, not bolted on as an extension. It handles TON settlement, asset provenance, and later, $KREDS when the chain layer activates. The player sees purchase confirmations. The underlying blockchain mechanics remain abstracted by default unless the player chooses to inspect or engage them directly.

My Maits interface. EveryWear becomes one of the primary surfaces through which players interact with their compiled Mait agents. The Chromium fork provides the runtime environment for agentic interaction. The Mait can assist with browsing, provide contextual information, and operate as a personal AI layer across the player's digital activity, not only inside the game.

The Chromium fork also hosts the first out-of-game A.R.E. surface. When the player browses outside of game sessions, the A.R.E. panel offers consented, compensated attention opportunities. The diegetic framing is lighter than the in-game Proper Gander aesthetic, but the consent architecture and revenue split are identical.

Phase 2 proves that EveryWear can function as a persistent sovereign client: vault, wallet, agent surface, and earning layer in a single runtime. What it cannot do is deliver the high-fidelity 3D experience the game requires as it matures beyond the initial desktop OS phase.

Phase 3: Game Launcher and WebGL Bridge

Purpose: Delivering richer game experiences without abandoning the EveryWear shell.

As the game evolves beyond the flat desktop OS into three-dimensional environments, the Chromium fork's rendering capabilities are no longer sufficient. Phase 3 introduces WebGL bridging, allowing EveryWear to launch and host progressively richer game content while maintaining the persistent shell around it.

At this phase, EveryWear functions as a launcher: it handles authentication, loads the player's vault and wallet state, loads the player's Mait interface, and then hands off to the WebGL game layer for the immersive gameplay session. When the player exits the game, they return to the EveryWear shell with full continuity of data, identity, and agent context.

This is the phase where the tension between game fidelity and platform function becomes visible. The game wants to be heavier ; more geometry, more dynamic environments, more compute-intensive rendering. The platform wants to remain lean ; vault, wallet, agents, earning. Phase 3 is the bridge. Phase 4 is the resolution.

Phase 4: Bifurcation

Purpose: Separating high-fidelity game clients from the persistent sovereign runtime.

This is the architecturally decisive phase. Unity and Unreal become their own dedicated game experiences, delivering the visual and interactive fidelity that WebGL cannot match. These are no longer "EveryWear" in the narrow sense. They are standalone game clients ; rich, immersive, optimised for their respective engines ; that launch from and return to the EveryWear shell.

EveryWear itself remains the persistent interface: the launcher, the vault, the wallet, the agent surface, the A.R.E. host, and the continuity layer that ties the player's identity and data across every game client and platform surface they use. A player might run the Unity client on desktop, the Unreal client on console, and the EveryWear browser on mobile. In every case, their Mymories vault, their compiled Mait, their wallet state, and their earning history persist through EveryWear.

The bifurcation is necessary because a high-fidelity game client and a sovereign runtime shell should not remain one bloated object. They have different performance requirements, different update cycles, different scaling characteristics, and different user expectations. Splitting them allows each to evolve at its own pace without compromising the other.

After bifurcation, EveryWear is definitively not "the game client." It is the persistent layer around the game clients ; and around everything else in the Strands ecosystem.

Phase 5: Agentic OS Across XR, Desktop, and App

Purpose: Convergence into a user-controlled operating layer for spatial computing.

At full maturity, EveryWear is no longer merely a browser or a launcher. It is an agentic operating system: a sovereign runtime that hosts the player's compiled agents, sovereign data, wallet, validation functions, and spatial interaction layer across every device class ; desktop, mobile, and XR headsets.

At this phase, five core functions converge:

Game Client orchestration. EveryWear launches and coordinates game sessions across multiple engine targets (Unity, Unreal, WebGL) while maintaining persistent identity and state.

Data Vault. Mymories at full maturity: the player's sovereign memory stack encompassing interaction history, consent records, asset provenance, spatial data, and the contextual substrate that powers their Mait agents. All data stored in player-controlled encrypted storage, portable across devices.

SAL Runtime. The Structured Adaptive Layer operates within EveryWear using context provided by Mymories. As the vault matures, this enables a personalised contextual SAL: a user-shaped adaptive behaviour layer built from Synch Profile, accumulated interaction history, consented data, and contextual signals. This becomes one of the building blocks of future decentralised cognition.

A.R.E. Host. The full Attention Redistribution Engine runs through EveryWear: consent management, context assembly, diegetic delivery, attention measurement, revenue calculation, and payment settlement. In XR environments, attention verification graduates from interaction-based measurement to spatial sensing through WiFi DensePose presence detection and, eventually, hardware-level eye tracking.

Staking, validation, and spatial data. At later ecosystem maturity, EveryWear-equipped devices can participate in network validation, contributing to the Strands Chain's consensus while earning validation rewards. XR devices additionally host and contribute spatial data ; the volumetric AR layer data that powers Layer U's spatial economy (detailed in Chapter 5). This function activates only at later maturity, not at launch.

The end state is no longer merely a browser. It is a user-controlled operating layer through which the player's entire relationship with the Strands ecosystem is mediated ; agents, memory, earning, identity, and spatial interaction ; regardless of device form factor.

Core Functions at Maturity

At full maturity, EveryWear carries five functions simultaneously. Remove any one and the rest lose range, continuity, or utility.

Function	Role	Phase Activated
Game Client Orchestration	Launches and coordinates game sessions across engine targets	Phase 3
Data Vault (Mymories)	Sovereign memory, consent, asset provenance, spatial data	Phase 2
SAL Runtime	Personalised contextual behaviour grounded in the Mymories substrate	Phase 5
A.R.E. Host	Consent-based attention economy with fiat settlement	Phase 2 (basic), Phase 5 (full spatial)
Validation and Spatial Data	Chain validation, XR spatial layer hosting	Phase 5

CHAPTER 5

Layer U and the A.R.E.

What Layer U Is

Layer U exists in two related forms across time.

In the real world, Layer U begins as the spatial governance and attention economy infrastructure of the Strands ecosystem. It is the layer through which spatial XR inventory is administered, attention is measured and compensated, and revenue flows between users, investors, and the platform. It is not a product the player downloads. It is the economic and spatial substrate beneath the products they use.

In the game, set 555 years later, Layer U is what that infrastructure has become after centuries of capture, adaptation, and decentralisation: the rebellion's communications, coordination, and parallel economic layer inside MetaXity1. It is no longer merely a platform layer. It has evolved into the resistance's decentralised communications network, its coordination infrastructure, its shadow economy, and the substrate through which the underground operates in the cracks of SOVcorp control.

The in-game Layer U is the long-term mutation of the real-world Layer U. The real-world platform is the seed. The in-game rebellion layer is its descendant. This is not a repeated name across fiction and deployment. It is one system at different stages of historical evolution.

Layer U in 2026

Layer U is a spatial XR attention economy platform built around consented, rewarded viewing and administered locally for investment purposes through SPVs incorporated in each jurisdiction it serves. Each SPV manages volumetric space leasing, attention-based revenue distribution, and investor returns within its geographic and regulatory context, ensuring localised legal appropriateness while operating within a shared architectural and economic framework across the Strands network. The governance logic remains decentralised in principle, but the deployable legal vehicle in real jurisdictions is the SPV.

Like EveryWear, Layer U and the A.R.E. do not arrive fully formed. They evolve through staged validation: first as in-game spatial and attention mechanics, then as diegetic rewarded systems, then as real advertiser integrations, then as XR-linked deployments, and only after sufficient proof as live city-scale infrastructure and chain-integrated network economy. Each stage must prove the next before the next activates.

The mechanism through which Layer U generates and distributes revenue is the A.R.E.: the Attention Redistribution Engine.

What the A.R.E. Is

The Attention Redistribution Engine is the economic mechanism that converts user attention into measurable, compensated value. It inverts the extractive advertising model. Instead of harvesting attention silently and selling it to advertisers while returning nothing to the user, the A.R.E. makes the value exchange explicit: users opt in, their attention is measured transparently, and they receive a direct share of the revenue their attention generates.

The split is 60/40. Sixty percent of A.R.E. revenue goes to the user. Forty percent goes to the Strands ecosystem ; covering platform operations, infrastructure, and investor returns through the Layer U SPV network.

Before the native chain is active, A.R.E. revenue share settles in USDT to the player's TON wallet following advertiser reconciliation and SPV settlement cycles, most likely monthly or quarterly in arrears. Earnings can accrue continuously in the interface, but payout follows real revenue settlement rather than artificial instant distribution. Once the Strands chain and full network economy are active, settlement can transition into $KREDS paid to the player's network wallet, reflecting the internalisation of value within the Strands ecosystem.

This is not a speculative abstraction. It is the target economic architecture of the platform once the game has proven the fiat model works. The A.R.E. activates after the game economy demonstrates retention and revenue viability, consistent with the sequential validation doctrine described in Chapter 1.

The Diegetic Origin

The A.R.E. does not arrive as an ad platform bolted onto a game. It grows out of the game's own narrative mechanics.

Inside Strands the Game, players encounter Signal Reclamation ; the diegetic mechanic described in Chapter 2 where players intercept, decode, and redirect SOVcorp propaganda broadcasts (Proper Gander). These broadcasts are narratively framed as corporate messaging within the game world. Players who engage with Signal Reclamation earn in-game currency (GridScrip) for their attention and participation. The entire experience is narrative: the player is performing resistance intelligence work, not watching advertisements.

This is the first layer of education. The player learns, through diegetic play, that attention has measurable value, that consent determines participation, and that engagement generates compensation. These are the foundational principles of the A.R.E., taught through gameplay before the real-world mechanism ever activates.

From Diegetic to Real

The transition from in-game narrative mechanic to global spatial economy follows a seven-stage validation ladder. Each stage must prove viability before the next activates. This mirrors the phased discipline applied to EveryWear in Chapter 4.

Stage 1: Layer U in-game. Layer U is introduced as a diegetic concept inside Strands the Game. The resistance operates parallel spatial infrastructure ; underground networks, signal towers, zone governance ; that teaches players the logic of controlled attention economies and decentralised spatial coordination through play. No real-world Layer U infrastructure exists at this stage. The game is the education.

Stage 2: A.R.E. as diegetic mechanic. Signal Reclamation and adjacent systems establish consented, compensated attention as a narrative mechanic. Proper Gander broadcasts are game content. Compensation is in-game currency (GridScrip). No real advertising is present. The player learns the behavioural pattern ; opt in, engage, earn ; within the safety of the game's fiction. These are the foundational principles of the A.R.E., internalised through gameplay before the real-world mechanism ever activates.

Stage 3: Real ad integration inside the game. Real advertising begins to appear within the game's diegetic frame. Ads are presented as Proper Gander broadcasts or SOVcorp corporate feeds ; narratively integrated, not interruptive. The player opts in as before, but now the revenue generated is real. The 60/40 split activates. The player's share is credited to their fiat balance (USDT-denominated), accessible through their EveryWear wallet. The diegetic container remains intact: the player is still performing Signal Reclamation. The difference is that the broadcast now carries real commercial content alongside ; or woven into ; the narrative frame. This stage validates that pay-to-watch revenue sharing produces meaningful engagement and conversion. The transition to subsequent stages does not occur on roadmap timing alone. It occurs only once enough transaction volume, engagement depth, and conversion evidence have accrued to demonstrate that compensated attention produces meaningful advertiser outcomes.

Stage 4: XR tied to game-related deployment. Spatial XR advertising and Layer U mechanics first extend into controlled, game-adjacent contexts. When the player browses through EveryWear outside of game sessions, the A.R.E. panel offers attention opportunities with a lighter diegetic frame. The consent architecture is identical. The revenue split is identical. XR-linked content begins to appear in contexts still closely tied to the Strands ecosystem and user base. This proves that volumetric XR delivery works outside pure game fiction but before full urban rollout.

Stage 5: Main chain launch. Once sufficient validation exists across the game economy, A.R.E. engagement, and initial spatial deployment, the Strands chain activates. Settlement and governance can begin to transition into the native network economy. $KREDS becomes available as a settlement currency alongside fiat. This is the threshold where the internalised Strands economy becomes structurally meaningful. Layer U SPV operations can begin to integrate on-chain governance and settlement mechanisms.

Stage 6: XR pilot cities. The first real-world Layer U SPV deployments activate in selected pilot cities ; Kuala Lumpur first, followed by Bangkok, Jakarta, and Singapore. City-level SPV infrastructure, local operational frameworks, and volumetric lease markets go live. Attention measurement graduates from interaction-based metrics to spatial sensing ; WiFi DensePose presence detection and, at hardware maturity, eye tracking through XR devices. The advertising becomes spatially contextual: tied to physical locations, personalised per user, and measured with higher fidelity than any legacy out-of-home medium. This stage proves that the model works in live urban conditions with real advertisers, real leases, and real revenue distribution.

Stage 7: XR global. Wider geographic rollout after city-level validation. Layer U expands as a global spatial economy infrastructure, with new SPVs incorporating in each jurisdiction as the network grows. The volumetric leasing model, A.R.E. consent pipeline, and SPV governance framework scale across markets, each new city building on the operational and commercial proof generated by its predecessors.

The logic is sequential and non-negotiable. Each stage generates the data, the revenue proof, and the operational confidence required to activate the next. No stage is skipped. No stage is assumed.

The A.R.E. Pipeline

The A.R.E. operates through a six-stage interaction pipeline that governs each attention event from consent to payment:

Consent. Explicit, informed, revocable. Stored in the player's Mymories vault with timestamp and version hash. Never activated before onboarding completes. The player chooses to participate. They can withdraw at any time.

Context Assembly. Happens locally within EveryWear using the Mymories data substrate. An anonymous category vector is transmitted to the ad-matching layer. No personally identifiable information leaves the device. The advertiser sees a category request from an anonymous node, not a person.

Diegetic Delivery. In-game, ads carry the full Proper Gander aesthetic. Out-of-game, the frame is lighter but the structural wrapper remains. The delivery is never a raw interruptive pop-up. It is always contextualised within the Strands visual and narrative language.

Attention Measurement. Phase 1 (game): interaction-based measurement ; dwell time, completion, engagement signals. Phase 2 (platform): WiFi DensePose spatial presence detection. Phase 3 (hardware): XR eye-tracking for verified gaze-based attention.

Revenue Calculation. eCPM model. The player's 60% share is calculated per impression based on verified attention metrics. Higher-fidelity measurement enables higher eCPM pricing, which directly increases the player's earnings.

Payment. In the pre-chain phase, revenue share settles in USDT to the player's TON wallet following revenue reconciliation and SPV settlement. In the mature network phase, settlement can transition into $KREDS paid to the player's Strands wallet as part of the native network economy.

This pipeline is not optional infrastructure. It is the mechanism through which every A.R.E. interaction flows. No stage can be skipped. No data leaves the device without consent. No payment is withheld beyond the relevant reconciliation and settlement cycle.

Consented Data Licensing

Alongside rewarded attention, the A.R.E. supports a second revenue stream: consented data licensing.

With explicit player permission, anonymised interaction pattern data ; browsing behaviour, content preferences, engagement timing, app usage patterns ; can be licensed to third parties for research, advertising optimisation, and AI training. This is explicit, revocable, consent-based data licensing under player-controlled scope. The data is anonymised and aggregated. The consent is tiered: the player chooses exactly what scope of data they are willing to share. A transparent dashboard shows what is collected, what it earned, and how it was used. Withdrawal is one click, at any time.

The market advantage is structural. GDPR-style consent architecture from day one, even in regions that do not yet require it. Brands pay premium for consented data because it is legally defensible, higher quality, and future-proof against regulatory tightening. In a market where most behavioural data is extracted without informed consent, transparently consented data is a competitive asset.

The player receives compensation for any data they choose to share. The A.R.E.'s design principle applies uniformly: if value is generated from the player's participation, the player receives their share.

Layer U Spatial Economy

The A.R.E. is the mechanism. Layer U is the infrastructure through which that mechanism operates at urban scale.

Layer U administers volumetric XR real estate: three-dimensional digital space overlaid on physical urban environments. Cities are divided into volumetric parcels measured in cubic metres. Each parcel can be leased for advertising, branding, interactive engagement, virtual retail, or spatial content. Advertisers pay per verified impression, not per static slot, enabling micro-segmented campaigns that combine programmatic precision with real-world spatial presence.

This is a fundamentally new asset class. It is not speculative metaverse land. It is commercial use of physical airspace for digitally overlaid content, priced against real foot traffic, real engagement data, and real advertiser demand.

The SPV Network

Layer U operates through a network of investible SPVs ; Special Purpose Vehicles structured as locally incorporated companies in each jurisdiction they serve.

Each SPV administers the Layer U operations within its geographic territory: leasing volumetric space to advertisers, managing the A.R.E. revenue pipeline, distributing funds between users (60% attention revenue), investors (returns from lease revenue and appreciation), and the Strands platform (operational share). The SPV structure ensures that each entity complies with its local regulatory, corporate, and tax framework while operating within the shared Strands economic architecture.

It is a network of related companies, each jurisdictionally appropriate, collectively forming the Layer U infrastructure across multiple cities and regions. The first target deployment is Kuala Lumpur, with subsequent expansion into Bangkok, Jakarta, and Singapore.

The SPV model makes Layer U investible through conventional fiat instruments. Investors can participate in a specific city's Layer U deployment through its local SPV, receiving returns generated by volumetric lease revenue and A.R.E. attention economics within that territory. This bridges the gap between the Strands ecosystem's longer-term blockchain ambitions and the immediate need for fiat-denominated, legally structured investment vehicles that institutional and private investors can engage with today.

In later ecosystem maturity, when the $KREDS chain layer activates, Layer U SPVs may integrate on-chain governance and settlement mechanisms. But that transition occurs only after the fiat model has proven viable. The SPV structure is not a temporary workaround. It is the correct legal and commercial architecture for deploying spatial economic infrastructure across diverse jurisdictions.

The Game as Proof of Concept

Stages 1 and 2 of the validation ladder are not placeholders. They generate the operational data that de-risks every subsequent stage. Every in-game mechanic that touches attention, spatial governance, or economic distribution maps to a real-world Layer U operation:

In-Game Mechanic	Layer U Equivalent
Signal Reclamation (Proper Gander broadcasts)	A.R.E. consented rewarded attention
Billboard hijacking (SIGOPS)	XR ad placement in physical locations
Signal Tower capture	XR node deployment
Zone governance voting	Local Layer U policy and revenue governance
Volumetric content placement	Spatial advertising in cubic-metre parcels
Attention measurement (dwell time, interaction depth)	Consent-based engagement tracking for ad pricing

The game generates six categories of data that directly de-risk Layer U deployment: attention economics (what users will watch, for how long, at what completion rate), consent tier adoption (what proportion of users opt in at each consent level), volumetric interaction patterns (how users engage with spatial content), governance behaviour (how communities make collective economic decisions), dual-economy circulation (how value flows between parallel economic systems), and creator economy dynamics (how user-generated content affects engagement and monetisation).

All of this data is consent-gated, anonymised, and aggregated. None of it leaves the game without player permission. But its existence means that by the time the first Layer U SPV launches in Kuala Lumpur, the team is not guessing at user behaviour. The game has already generated the data to price attention, structure leases, and forecast returns.

Volumetric Pricing and Lease Dynamics

Layer U parcels are leased in $10m^3$ base units. Initial pricing establishes floor liquidity ; conservatively benchmarked against traditional out-of-home advertising costs in each city, at a fraction of the price. A startup leasing a $50m^3$ XR pop-up near a prime transit node pays orders of magnitude less than a traditional billboard tenant in the same location, while gaining interactive 3D engagement, real-time content updates, and per-impression analytics that legacy out-of-home media cannot match.

After the initial lease period, parcels enter open renewal pricing informed by historic engagement, verified impressions, and predictive footfall models. Demand naturally concentrates around high-traffic locations, creating a self-sustaining pricing dynamic where the most valuable spatial real estate appreciates through demonstrated commercial utility rather than speculation.

Because delivery is client-side and synchronised to the user, the same XR placement can present radically different creative variants to different opted-in viewers. A single billboard is no longer a single message. It becomes a context-aware delivery surface calibrated to the user's synchronisation profile, preferences, and consented data signals. This allows much higher relevance than legacy advertising while remaining governed by explicit consent, user-controlled data architecture, and privacy-preserving safeguards.

PART III: THE CHAIN

CHAPTER 6

$KREDS Tokenomics

What $KREDS Is

$KREDS is the native economic protocol of the Strands Network. It is the mechanism through which value generated across the ecosystem ; through attention, participation, spatial infrastructure, agent composition, governance, and premium services ; circulates back to participants within a single native economy rather than remaining trapped in fiat rails or leaking through external intermediaries. At maturity, validation, settlement, leasing, agent composition, governance, and premium services can all resolve through $KREDS, closing the loop between value creation and value capture inside the network.

$KREDS is not an in-game currency. In-game economies operate on SOVComp and GridScrip, as described in Chapter 2. $KREDS is not a speculative asset designed for passive appreciation. It is not a reward token distributed for engagement. It is a network-native economic instrument that represents real contributions: transaction validation, attention provision, data consent, infrastructure operation, and community participation.

$KREDS launches only after the fiat economy has proven the model works. This is the sequential validation doctrine applied to tokenomics: the game economy must demonstrate retention and revenue viability, the A.R.E. must demonstrate that consented attention generates real advertiser value, and Layer U must demonstrate that spatial economic infrastructure attracts real investment ; all before $KREDS enters circulation. If those conditions are not met, $KREDS does not launch. The fiat economy continues to operate independently.

Why $KREDS Exists

Most Web3 token economies suffer from one of two structural failures. Deflationary models create artificial scarcity that encourages hoarding over utility, reducing liquidity and slowing network participation. Inflationary models mint new supply without underlying demand, leading to devaluation and unsustainable incentive structures. In both cases, the token economy is disconnected from the productive activity it claims to represent.

$KREDS addresses this by tying token supply directly to tangible participation metrics. Tokens are minted in response to real network expansion ; new validated applications, new runtime surfaces, new service layers, new operational infrastructure. Tokens are burned in response to real lifecycle events ; application retirement, stake expiry, transaction settlement. The supply curve reflects the actual state of the network, not the speculative expectations of external traders.

The result is an elastic economic instrument whose value tracks the productive capacity of the Strands ecosystem rather than the sentiment of secondary markets.

Phased Deployment

$KREDS does not appear at the start of the Strands economy. It formalises an economy that has already been proven through fiat-denominated game activity, USDT-linked settlement flows, and compensated attention mechanics inside the Strands ecosystem. Once those economic behaviours have demonstrated real retention, revenue, and utility, $KREDS can begin circulating as a Jetton on TON, potentially first inside in-game or game-adjacent economic flows. Only after that utility has been validated, and once the native chain is operationally ready, does $KREDS transition into the sovereign asset of the Strands Chain, where the wider extra-game network economy fully internalises.

$KREDS follows the same staged validation discipline applied to EveryWear (Chapter 4) and Layer U (Chapter 5).

Phase 0: Fiat and Stable Settlement

Before $KREDS is active as a live token economy, the behavioural and commercial logic it will later formalise is already proven inside the game. Fiat-denominated game economics, USDT-linked reward settlement, and in-game compensated attention mechanics demonstrate that users will opt in, engage, and generate meaningful advertiser value. The economy is proven before the token formalises it. If these behaviours do not validate, $KREDS does not launch.

Phase 1: Jetton on TON

Once those behaviours are validated, $KREDS deploys as a Jetton ; a token standard native to the TON blockchain. This is not a compromise. It is a deliberate architectural choice that provides immediate access to TON's 900 million monthly active user base, sub-cent transaction fees, sub-second finality, and native integration with Telegram's payment infrastructure.

$KREDS may first appear inside in-game or game-adjacent economic flows, where users already have continuity through Telegram and TON wallet infrastructure. There is no new wallet to install, no new chain to understand, no friction beyond what the user has already navigated. Token utility begins in the environments where the fiat economy has already proven the underlying behaviours.

The Jetton phase proves three things: that $KREDS circulates as a functional economic instrument within a live user base, that the elastic supply mechanics operate correctly at scale, and that the token economy generates genuine utility rather than purely speculative activity.

Phase 2: Bridge to Strands Chain

Once the Jetton phase has proven circulation and utility, and once the native Strands chain (detailed in Chapter 7) has reached operational readiness, $KREDS bridges from TON to the native chain. This transition moves $KREDS from a hosted token on external infrastructure to the sovereign asset of its own network ; a Mina Protocol fork with zk-SNARK privacy, succinct ledger architecture, and mobile-first validation.

This is the phase where settlement fully internalises into the Strands economy. Governance, validation, leasing, and broader network utility consolidate on chain. The bridge is not a migration event that forces users to act. It is an architectural upgrade that happens beneath the interface. Users who hold Jetton $KREDS see their balance reflected on the native chain. The wallet interface remains the same. The economic relationships remain the same. The underlying infrastructure becomes sovereign.

Elastic Supply Mechanics

$KREDS operates on a mint-and-burn equilibrium that dynamically adjusts supply based on real-world network activity.

Minting

New $KREDS enters circulation through three channels:

Network expansion minting. New $KREDS enters circulation in response to validated network expansion: the onboarding of new applications, runtime surfaces, service layers, and operational infrastructure that increase the productive capacity of the Strands ecosystem. Whether those applications run on existing consumer hardware or future dedicated devices is secondary. What matters is the growth of usable network participation. This ties token creation directly to real ecosystem utility: new productive capacity brings new supply, ensuring there are always enough tokens to support the growing network without over-minting beyond what it can absorb.

Validator rewards. Participants who hold a protocol bond and contribute to network validation receive token rewards proportional to their participation. The protocol recalculates emission rates per validator each epoch: higher network-wide uptime produces lower individual APR, maintaining a fixed aggregate issuance ceiling. This ensures that validation rewards scale with genuine contribution rather than inflating without bound.

Infrastructure expansion. As Layer U SPVs deploy spatial infrastructure in new cities, protocol-level issuance can support expansion only where verified new productive capacity is created, rather than against abstract supply targets.

In later maturity, as Strands expands into dedicated XR hardware and spatial operating environments, this same logic extends to device-native runtimes with EveryWear built in. The long-term ambition is not merely to run on XR devices, but to become part of the operating layer through which XR interaction, spatial data, agents, and network participation are mediated.

Burning

$KREDS is removed from circulation through three mechanisms:

Participation lifecycle burn. When a validated application or runtime surface is retired, replaced, or upgraded, its staked $KREDS is burned. This creates a natural deflationary pressure that tracks the operational lifecycle of the network's productive capacity. As older participation contexts cycle out, supply contracts proportionally.

Stake expiry burn. Tokens staked for validation, Trait Shard refinement, or infrastructure participation are partially burned upon unstaking. This ensures that long-term participation is rewarded while short-term speculation is structurally discouraged.

Transaction burn. A fraction of $KREDS spent on Layer U spatial leases, Trait Shard transactions, and premium ecosystem services is automatically burned. This creates a continuous deflationary counterbalance to minting, calibrated to transaction volume rather than arbitrary schedule.

The interaction between minting and burning produces a self-regulating supply curve. When the network grows, minting outpaces burning and supply expands to support new participants. When growth plateaus, burning outpaces minting and supply contracts, concentrating value among active participants. The equilibrium is dynamic, not fixed.

Utility

$KREDS is designed for network utility, not speculative holding. Its primary use cases are:

A.R.E. settlement. In the mature network phase, A.R.E. attention revenue can settle in $KREDS rather than USDT, internalising value circulation within the Strands economy. Users who earn through consented attention receive $KREDS to their network wallet, which they can spend within the ecosystem or exchange externally.

Layer U spatial leasing. As Layer U SPVs mature and integrate on-chain settlement, volumetric XR lease payments can be denominated in $KREDS. Advertisers acquire $KREDS to pay for spatial inventory. Lease revenue distributes to SPV participants in $KREDS. The spatial economy becomes a primary demand driver for the token.

Trait Shard transactions. On-chain Trait Shard ownership, trading, and evolution ; the composable dNFT architecture described in Chapter 3 ; settles in $KREDS. This makes the companion economy a native part of the token ecosystem rather than an externally settled sidecar.

Validation and protocol bond. Network participants hold a protocol bond analogous to stake, tied to possession of a validated application or runtime on the network. This bond is not a capital-weighted power mechanism ; validation rights arise through verified identity, uptime, and ongoing contribution, not through token accumulation. One verified identity holds one active validator slot. Participation, not passive yield, determines validator standing.

Governance. $KREDS holders participate in network governance decisions: protocol upgrades, emission parameters, Layer U policy frameworks, and ecosystem fund allocation. Governance rights are weighted by validated participation, not by passive holdings or token accumulation.

Compute and services. Premium ecosystem functions ; enhanced SAL services, expanded Mymories storage, advanced Mait refinement, and premium EveryWear capabilities ; are accessed through $KREDS, creating ongoing demand native to the Strands stack rather than generic platform fees.

Token Distribution

The total supply of $KREDS is capped at 100 billion tokens. The total supply cap defines the maximum issuance boundary, while live circulating supply remains elastic within that cap according to minting, burn, and participation dynamics. Distribution is structured around utility and participation, not privileged early access.

Validator and participation rewards ; allocated algorithmically and constrained by the number of active verified validator identities. One verified identity, one active validator slot. Validation rights do not expand through token accumulation.

Development and treasury (5%) ; reserved for ongoing development, research, and operational resilience.

Community growth (5%) ; allocated to early adopter incentives, community campaigns, and governance participation bootstrapping.

Liquidity (10%) ; placed on decentralised exchanges for trading liquidity and price discovery.

No venture capital allocation. No presale. $KREDS launches as a true value token ; earned through participation, not purchased through privileged access rounds. This is a deliberate structural choice that aligns token holder incentives with network health rather than exit liquidity.

The majority of supply is reserved for protocol-level issuance to validated participation and network growth over time, with only the explicitly listed portions carved out for treasury, community activation, and liquidity.

Lock and Emission Decay

To align all participants around long-term network health, $KREDS employs a uniform lock and emission schedule. The same structure applies to every participant ; founding validators, infrastructure partners, and end users. There are no privileged unlock schedules.

This stake is not primarily a user-purchased position. It is an invisible protocol stake attached to validated participation within the network.

Cliff (months 0-8). Zero tokens unlocked. Validator nodes must remain online and contributing to reach the first redemption point. This enforces genuine participation.

Taper (months 9-18). Linear unlock of staked principal and accumulated emissions. Smooths supply release and discourages cliff-day sell pressure.

Post-18 months. All new rewards paid liquid. The protocol-level stake associated with that participation epoch is retired through burn at the end of the cycle. The participant is free to restake on a new runtime surface or upgraded application context, starting a new epoch.

Each validated participation context carries one active 18-month protocol epoch. When a validated application or runtime is upgraded, replaced, or retired, a new epoch begins and the prior protocol stake is cycled out under burn rules defined by the network. Unclaimed emissions port to the new wallet. This creates a natural renewal cycle where network participation and application lifecycle are structurally aligned.

Identity and Anti-Sybil

$KREDS validation requires one verified identity per active validator slot. Duplicate attempts automatically pause reward accrual.

Identity verification leverages Telegram account continuity and phone-linked identity, supplemented by device-level authentication where available. This provides a practical anti-Sybil onboarding layer without requiring users to navigate separate KYC processes, seed phrases, or hardware-specific biometric enrolment. Identity continuity travels with the user across devices through Telegram's existing account infrastructure.

If a user migrates to a new device, the Telegram-linked identity transfers the wallet and any un-tapered stake to the new hardware. The old device is invalidated. Cumulative uptime travels with the user, preserving seniority in reward calculations. This is not a sovereign identity system in its final form ; it is a practical foundation that can evolve as the Strands chain matures and more robust identity primitives become available.

Layer U Revenue Integration

Once native settlement is active, advertisers acquire $KREDS in order to purchase Layer U inventory and A.R.E.-linked attention access. That acquisition may occur through ecosystem-controlled liquidity faucets, LP routes, open market sources, or a combination of these, depending on final market structure. The purpose of this demand is not merely market support. It is functional settlement inside the Strands economy. $KREDS acquired through advertiser demand can then be recycled into ecosystem flows, including the user side of the 60/40 A.R.E. reward structure, so that commercial demand directly funds participant value distribution.

The exact balance between ecosystem-controlled liquidity routes and open market sourcing remains subject to final market design, but in all cases advertiser demand is intended to create genuine utility-driven acquisition rather than purely speculative turnover.

As Layer U SPVs mature, lease payments, attention revenue distribution, and infrastructure settlement all resolve in $KREDS, completing the transition from fiat-mediated to native economic flow. While $KREDS may circulate through external trading venues, the economic preference of the system is to keep value circulating within the Strands ecosystem once it enters ; through settlement, rewards, leasing, services, governance, and continued participation rather than unnecessary extraction into purely speculative flows.

CHAPTER 7

Strands Blockchain

Why a Native Chain

For the first two phases of $KREDS deployment, external infrastructure is sufficient. The Jetton standard on TON provides circulation, wallet integration, and settlement without requiring Strands to operate its own chain. This is deliberate. The game economy validates in fiat. $KREDS validates as a Jetton. Only once both have proven their respective models does the question of chain sovereignty become relevant.

The native Strands chain activates when the ecosystem requires capabilities that a hosted token on external infrastructure cannot provide: protocol-level privacy for participant data, succinct verification that allows mobile devices to act as full nodes, and sovereign control over consensus rules, emission logic, and governance execution without dependence on a third-party chain's roadmap or fee structure.

The chain does not create the economy. It inherits one that already works.

Architecture

Strands is designed as a validator construct built around client ledgers. Each participant maintains a cryptographically compiled local ledger representing their operative state within the network ; including spendable balance, participation context, and transaction-relevant application state. This local ledger lives on device and within user-controlled runtimes such as EveryWear and Mymories. The chain does not store that underlying state. It confirms that claimed state and state transitions are cryptographically valid. What settles on chain are proofs, attestations, settlement records, provenance, and state transition confirmations ; not the raw underlying data itself.

When a transaction is initiated, the network does not rely on a monolithic public ledger of total user history. Instead, it verifies that the sending client ledger is authentic, untampered, and entitled to declare the proposed transition; that the receiving ledger is valid and able to receive; and that the transfer can be accepted without conflict or double spend. The global chain exists to confirm the legitimacy of these state transitions, not to hold the full lived reality of every participant.

Client-side ledger integrity is maintained through strong cryptographic hashing, with a post-quantum migration path as the network matures. This distinction is foundational. The device carries the continuity. The chain confirms the truthfulness of that continuity.

The Strands chain is a Rust-implemented sovereign chain derived from Mina Protocol design principles, especially its use of zk-SNARKs and succinct ledger architecture. Mina was selected for one structural reason: its use of zk-SNARKs to maintain a constant-size ledger. Where conventional blockchains grow linearly with transaction history ; requiring ever-larger storage and sync times ; a zk-SNARK chain stores only a compact cryptographic proof of the current state. New nodes verify the entire chain by checking a single proof rather than replaying every historical transaction.

This has direct consequences for the Strands ecosystem:

Mobile-first validation. Because the ledger proof is small enough to verify on a smartphone, any device running EveryWear can participate directly in validation as a network node. There is no mining hardware requirement, no high-bandwidth sync, no storage burden that excludes consumer devices. Validation is accessible to every participant, not gated by infrastructure cost.

Privacy through zero-knowledge proofs. Transactions, identity attestations, and participation records are validated without exposing the underlying data. A validator confirms that a proof is correct without learning what the proof contains. This means that A.R.E. settlement records, identity attestations, Trait Shard provenance, and Layer U transaction proofs can all settle on chain without creating a public surveillance ledger.

Sovereignty over protocol rules. Emission rates, burn mechanics, governance execution, and consensus parameters are controlled by the Strands network rather than inherited from an external chain. Protocol upgrades do not require permission from or alignment with a third-party foundation.

What the Chain Confirms

At maturity, the Strands chain provides the attestation, settlement, and verification layer for the ecosystem components described in earlier chapters. In each case, what settles on chain is the proof or record ; not the raw underlying data.

$KREDS as native asset. Once bridged from TON, $KREDS becomes the native asset of the Strands chain rather than a hosted token. Minting, burning, staking, and transfer resolve natively within the protocol rather than through external smart contracts.

Trait Shard provenance and composition. The composable dNFT architecture described in Chapter 3 settles on chain. Shard provenance, compilation records, and marketplace transactions are recorded as verifiable on-chain attestations. The creative content and behavioural state of a Mait remain on device.

A.R.E. settlement records. Once native settlement is active, attention revenue flows can resolve in $KREDS on the Strands chain, completing the transition from the USDT pre-chain phase described in Chapter 5. The chain records settlement outcomes, not raw attention data.

Layer U lease and governance records. Spatial lease transactions, SPV revenue distribution, and governance votes can settle on chain with full auditability and zero-knowledge privacy where required.

Identity and anti-Sybil attestation. The Telegram-linked identity continuity described in Chapter 6 anchors to on-chain identity attestations, providing the anti-Sybil layer for validator slot allocation without exposing personal data.

Scaling Between Device and Chain

Not every state transition needs to resolve directly against the global chain. Localised aggregation, rollup-style batching, and staged proof submission can sit between device-held state and final chain confirmation. A device running EveryWear can accumulate local state transitions, compress them into a succinct proof, and submit that proof for chain-level attestation at intervals rather than in real time.

This is how the one-person-one-node doctrine scales. Every EveryWear instance can participate as a validating node. Devices hold meaningful local state and continuity. The chain avoids becoming a bloated global storage layer. Heavier coordination compresses before final confirmation. Through succinct proofs, localised aggregation, and chain-level attestation, the network preserves privacy, reduces storage burden, and allows mass participation without requiring every node to store or replay the full lived history of the system.

Software First, Hardware Later

The client ledger model launches on existing consumer devices. Smartphones and standard hardware running EveryWear maintain local ledgers, generate proofs, and participate in validation using software-level cryptographic compilation. No specialised hardware is required at launch.

As the network matures, dedicated hardware can strengthen the model rather than replace it. Stronger secure local runtimes, better device attestation, hardware-backed signing and integrity guarantees, richer spatial and XR interaction surfaces, and more sovereign node environments all extend the capabilities of the client ledger without changing the fundamental architecture. The sequence remains: software-first client ledgers on existing devices, then stronger hardware-native sovereign runtimes, with the chain always serving as the validation and attestation layer above them.

Consensus and Validation

The Strands model includes a protocol bond analogous to stake, but it does not operate as conventional proof of stake. Validation rights do not expand through token accumulation. They arise through operation of a validated application or runtime on the network, tied to verified identity, uptime, and ongoing contribution. Consensus is structured to resist capture through unrestricted capital concentration or specialised hardware advantage.

The hierarchy of validation legitimacy is: verified personhood first, validated application or node presence second, active participation and uptime third, and proof verification plus honest contribution fourth, with the protocol bond operating beneath the surface as a structural commitment rather than a capital-weighted power mechanism. One verified identity holds one active validator slot, as described in Chapter 6. Validators confirm transactions by verifying zk-SNARK proofs of state transitions rather than replaying transaction history.

Validator rewards follow the emission schedule defined in Chapter 6: epoch-based, with cliff, taper, and liquid phases. The protocol recalculates emission rates per validator each epoch, maintaining a fixed aggregate issuance ceiling so that individual APR decreases as network-wide participation increases. This prevents inflationary runaway while rewarding genuine contribution.

Because the ledger is succinct, validator hardware requirements remain minimal. A smartphone running EveryWear with a stable connection can validate. This is not a theoretical aspiration ; it is a structural consequence of the zk-SNARK architecture. The chain is designed so that the same devices participants already use for the game, for A.R.E. interaction, and for $KREDS transactions can also secure the network.

The Bridge from TON

The transition from Jetton on TON to native $KREDS on the Strands chain is an architectural upgrade, not a user migration event. For the user, the bridge is designed to feel like an architectural upgrade rather than a disruptive migration, preserving balance continuity, validator state, and participation history with minimal friction.

A continuing bridge layer may remain in place after the initial transition, preserving interoperability with TON liquidity and Telegram-linked wallet infrastructure while the native chain ecosystem matures.

PART IV: OPERATIONS

CHAPTER 8

Governance, Privacy & Compliance

Why Governance Matters

The Strands ecosystem is designed to decentralise over time. The game launches under conventional company stewardship. $KREDS launches under protocol-defined rules. Layer U SPVs operate under jurisdictionally localised corporate structures. As each layer matures, governance decisions ; protocol upgrades, emission parameters, Layer U policy frameworks, ecosystem fund allocation ; must progressively shift from founding-team stewardship to participant-driven process.

This chapter describes how that shift is structured, what privacy architecture underpins it, and how the system maintains regulatory compliance across jurisdictions without centralising control.

Governance Model

Strands governance is participation-weighted, not capital-weighted. Governance rights attach to validated participation in the network, not to passive token holdings or accumulated capital. This follows directly from the consensus model described in Chapter 7: verified personhood, validated application or node presence, active participation, and honest contribution determine standing ; not the size of a wallet.

Network-Level Governance

At the protocol level, governance decisions include: changes to emission and burn parameters, consensus rule updates, protocol upgrades, and ecosystem fund allocation. These decisions are proposed and ratified through on-chain governance mechanisms once the native Strands chain is active.

Voting weight is determined by validated participation context ; not by raw $KREDS balance. Governance weight is derived from attested participation signals ; including validated node presence, uptime, contribution history, and active network involvement ; rather than passive balance alone. A participant who holds tokens passively does not accumulate equivalent governance influence. This prevents the plutocratic capture that undermines most token-governance systems.

Governance activation follows the same phased logic as the rest of the ecosystem. During the fiat and Jetton phases, protocol decisions remain under founding-team stewardship while the network is still below the threshold required for participant-governed execution. As the native chain matures and the participant base grows, governance authority progressively transfers to the network. The timeline for that transfer is not predetermined ; it is gated by network readiness, not by calendar.

Layer U Governance

Layer U SPVs operate as jurisdictionally localised companies, not as global decentralised autonomous organisations. Each SPV is governed by its participants ; including investors, operators, local participants, and approved community representatives ; within the legal and regulatory framework of its jurisdiction.

SPV-level governance decisions include: spatial lease pricing, local advertising standards, infrastructure investment, revenue distribution policy, and compliance with regional regulations. These decisions are made by SPV participants through structured voting processes appropriate to the corporate form of the entity.

The Strands protocol does not impose a single global governance model on Layer U. Each SPV adapts to its jurisdiction. A Singapore-registered SPV operates under Singapore corporate law. A Malaysian Sdn Bhd operates under Malaysian corporate law. The protocol provides the economic and attestation infrastructure; the SPV provides the legal and operational wrapper.

As the native chain matures, SPV governance actions ; votes, revenue distributions, lease transactions ; can settle on chain for auditability and transparency while the legal entity retains its jurisdictional standing. Over time, where local legal and regulatory frameworks permit, individual SPVs may explore transitioning toward DAO-based governance structures. This is not a default assumption ; it is a jurisdictionally dependent option that becomes available only where the regulatory environment favours it.

Privacy Architecture

Privacy in the Strands ecosystem operates on two complementary principles: data sovereignty at the edge, and zero-knowledge verification on chain.

Data Sovereignty

Meaningful personal and contextual data ; including interaction history, preference signals, agent interaction context, consent records, and attention patterns ; lives on device, held within the relevant local persistence and context layers provided by Mymories and EveryWear. Not all categories are stored in the same way or in the same structure, but the principle is uniform: personal data does not transit to centralised servers, does not settle on chain, and remains under user control. Only proofs, attestations, settlement records, provenance, and related verifiable outputs move to the chain where required.

This is not a policy commitment layered on top of conventional architecture. It is a structural consequence of the client-ledger model described in Chapter 7. The device holds the state. The chain confirms the truthfulness of that state without accessing the underlying data.

Zero-Knowledge Verification

The zk-SNARK architecture of the Strands chain allows transactions, identity attestations, and participation records to be validated without exposing the data they contain. A validator confirms that a proof is correct without learning what the proof represents.

This means that A.R.E. settlement can be verified without exposing individual attention behaviour. Trait Shard provenance can be confirmed without revealing the creative content of a Mait. Layer U lease transactions can be audited without requiring exposure of advertiser terms or user engagement data. Identity can be attested without revealing personal information.

The privacy model is not absolute anonymity. It is selective disclosure governed by user consent. Participants choose what to reveal, to whom, and under what conditions. The protocol enforces those choices cryptographically rather than relying on policy enforcement by a trusted third party.

Consent Architecture

Consent within Strands is explicit, informed, granular, and revocable. The A.R.E. consent pipeline described in Chapter 5 establishes the pattern: no data collection before onboarding completes, no consent bundling, no dark patterns, and revocation that takes immediate effect.

This consent architecture extends across the ecosystem. Mymories data licensing, Trait Shard marketplace activity, Layer U spatial interaction, and any future data-dependent service all operate under the same consent framework. The user grants consent per context, per purpose, and per duration ; and can revoke at any point without penalty.

Regulatory Compliance

Strands operates across multiple jurisdictions with different regulatory frameworks for data protection, financial services, advertising, and digital assets. The protocol does not assume uniform regulatory treatment across jurisdictions; the same economic or technical activity may be classified differently depending on local law, which is why operating wrappers remain jurisdiction-specific. The compliance model is designed to be adaptable rather than monolithic.

Data Protection

The client-ledger architecture and data sovereignty model are structurally aligned with the principles of GDPR, PDPA (Singapore), and comparable data protection frameworks. Personal data remains under user control on device. Processing occurs locally. Consent is explicit and revocable. Data minimisation is enforced by architecture, not by policy.

Where jurisdictional requirements mandate specific disclosure, retention, or reporting obligations, those obligations are met at the SPV or operating entity level rather than at the protocol level. The protocol provides the privacy infrastructure; the operating entity provides the compliance wrapper.

Financial Regulation

$KREDS is structured for network utility within the Strands ecosystem, with deployment phased to demonstrate real economic function before native circulation. The sequence ; fiat proof first, Jetton on TON second, native chain third ; is deliberately structured so that token utility is observable before the ecosystem depends on it.

Layer U SPVs that handle fiat settlement, advertising revenue, or investor funds operate under the financial regulatory framework of their respective jurisdictions. The Strands protocol does not substitute for or override local financial regulation. SPVs obtain whatever licences, registrations, or approvals their jurisdictions require.

Advertising Standards

Layer U spatial advertising operates within the advertising regulatory framework of each jurisdiction. SPV-level governance includes compliance with local advertising standards ; content restrictions, disclosure requirements, targeting limitations, and consumer protection obligations.

The A.R.E. consent architecture provides a structural foundation for advertising compliance: every interaction requires explicit user consent, attention data remains on device, and engagement verification occurs through zero-knowledge proofs rather than invasive tracking.

Decentralisation proceeds as network maturity allows, while regulated operating wrappers remain in place wherever local law still requires accountable legal entities.

CHAPTER 9

Roadmap

How to Read This Roadmap

This roadmap describes the sequence in which the Strands ecosystem is technically built, commercially validated, and progressively decentralised. It is not a marketing calendar. It is a dependency chain. Each phase exists to prove a specific layer of the system. No subsequent phase activates until the prior phase has validated its core assumptions.

Indicative timeframes are provided as target ranges from technical commencement, not as fixed calendar commitments. Actual progression depends on validation outcomes, funding readiness, and the operational maturity of each preceding layer.

For a broader view of the Strands project timeline, milestones, and community updates, see strandsnation.xyz.

Phase 1: Proof of Concept

What this phase proves: The core loop exists.

Target range: Approximately 4 weeks from technical commencement.

The POC validates core technical assumptions: that the game loop works, that Telegram Mini App distribution is viable, that the desktop OS shell can host the intended player experience, and that the foundational interaction patterns ; Synch Profiling, Mait interaction, in-game economy ; function as designed.

This phase does not require monetisation, blockchain integration, or A.R.E. activation. It proves that the game exists as a playable, coherent, and technically viable experience.

Key deliverables:

- Playable game loop within the Telegram Mini App shell
- Desktop OS interface functional and navigable
- Synch Profiling operational beneath the surface
- Core Mait interaction demonstrable
- In-game economy (SOVComp, GridScrip) circulating within test environment

EveryWear at this phase: Telegram Mini App (Phase 1 of the EveryWear evolution described in Chapter 4).

Activation gate: The core loop must exist and function. If it does not, nothing else proceeds.

Phase 2: Minimum Viable Product

What this phase proves: The product can acquire and retain real players.

Target range: Approximately 13 weeks from technical commencement.

The MVP extends the POC into a product that can acquire and retain real players. It introduces the first revenue-generating mechanics, stabilises the game client, and proves that the Strands experience can sustain daily active engagement.

Key deliverables:

- Stable Telegram Mini App with full onboarding flow
- First fiat revenue mechanics live (in-app purchases, starter packs, marketplace activity)
- SOVComp and GridScrip economy operating at scale within the player base
- Trait Shard acquisition, compilation, and marketplace functional
- Player retention and session metrics trackable and reportable
- Early diegetic A.R.E. wrapper operational (Proper Gander framing, compensated attention loop training player behaviour within the game)

EveryWear at this phase: Telegram Mini App maturing, with the Chromium fork in development.

Layer U and A.R.E. at this phase: Diegetic A.R.E. active in-game. Layer U exists only as in-game narrative (the rebellion layer described in Chapter 5). No real-world spatial deployment yet.

Activation gate: The product must acquire and retain real players. If it does not, subsequent phases do not activate.

Phase 3: Season 0

What this phase proves: The economy can monetise and generate commercially meaningful behaviour.

Target range: 3–6 months post-MVP greenlight, dependent on validation outcomes and funding readiness.

Season 0 is the first full content season. It proves that the Strands game can sustain a seasonal content cadence ; new narrative arcs, new Trait Shards, new gameplay systems, new economic dynamics ; while maintaining player retention and revenue growth.

Key deliverables:

- Full seasonal content cycle: narrative, mechanics, economy, and progression
- A.R.E. compensated attention generating measurable advertiser value
- Fiat settlement operational (USDT to TON wallet where applicable)
- Player base metrics sufficient to de-risk subsequent investment decisions
- Community governance seeds planted (feedback loops, player councils, early signalling mechanisms)
- Fainance Ltd. (UK) B2B Mait deployment in parallel, proving the agent framework outside the game

Season 0 is the proof-of-economy phase. The game must demonstrate that it generates real revenue, retains real players, and produces the behavioural data that de-risks everything downstream.

EveryWear at this phase: Telegram Mini App live with players. Chromium fork entering early deployment for advanced users.

Layer U and A.R.E. at this phase: Diegetic A.R.E. generating live advertiser data. First real advertiser integrations in-game. Layer U pilot planning begins where data supports it.

Activation gate: The economy must monetise and generate commercially meaningful behaviour. If Season 0 does not demonstrate fiat economic viability, no wider scaling, tokenisation, or native infrastructure activation proceeds.

Phase 4: Seasonal Validation Cycle (Seasons 1–12)

What this phase proves: The model repeats under live conditions across a sustained content lifecycle.

Target range: The first twelve-season validation arc spans approximately 13–20 months from Season 0 launch, accounting for production cadence, player dynamics, bug cycles, and content iteration.

The number twelve is not decorative. Twelve seasons represent the minimum sustained lifecycle required to prove that the game economy can survive repeated live tuning, that A.R.E. data remains commercially meaningful over time, that retention and monetisation are repeatable rather than one-off, and that the wider system justifies tokenisation.

Each season introduces new narrative, new mechanics, new economy tuning, and new Trait Shard releases. The cumulative effect is a maturing game economy with a growing, engaged player base and an increasingly validated commercial pipeline.

During this period:

- The fiat game economy matures through iterative tuning across seasonal cycles
- A.R.E. attention data accumulates across seasons, proving advertiser value at sustained scale
- Layer U pilot deployments begin in target cities where the data and commercial readiness support them
- EveryWear evolves from Telegram Mini App through Chromium fork and toward the WebGL bridge as player base and platform requirements demand
- Mymories and SAL begin accumulating meaningful context as the vault and adaptive layers mature
- The Blank Sync Ledger activates for players who cross the purchase threshold, building the invisible wallet base
- Out-of-game A.R.E. surfaces emerge through the Chromium fork, extending compensated attention beyond game sessions

EveryWear at this phase: Telegram Mini App → Chromium fork → early WebGL bridge. The platform evolves in response to player demand and rendering requirements.

Layer U and A.R.E. at this phase: In-game diegetic A.R.E. → real advertiser integration → out-of-game A.R.E. through the Chromium fork → Layer U pilot city deployments where data supports them.

No blockchain deployment occurs during this phase. The entire focus is on proving the fiat economy, maturing the game, and building the behavioural and commercial foundation that justifies everything that follows.

Activation gate: The model must prove repeatable under live conditions. If retention, revenue, and A.R.E. value do not sustain across multiple seasons, token deployment does not proceed.

Phase 5: $KREDS Jetton Deployment

What this phase proves: Token utility can emerge from proven behaviour rather than fabricate an economy that does not yet exist.

Target range: Only after the seasonal validation cycle has demonstrated sustained fiat economy viability. This is a hard gate, not a chronological delay.

$KREDS does not deploy because enough time has passed. It deploys because the fiat economy has proven retention, revenue, and A.R.E. advertiser value across a full seasonal lifecycle ; and because there is evidence that token utility will formalise an already functioning economy rather than invent one. As described in Chapter 6, $KREDS may first appear inside in-game or game-adjacent economic flows, where users already have continuity through Telegram and TON wallet infrastructure.

Key deliverables:

- $KREDS Jetton deployed on TON
- Initial circulation within game or game-adjacent economic flows
- Elastic supply mechanics (mint and burn) operational and observable
- Token utility demonstrable through real ecosystem transactions
- Anti-Sybil identity layer (Telegram account continuity) validated at scale

EveryWear at this phase: Chromium fork mature. WebGL bridge operational. Bifurcation (Chapter 4, Phase 4) approaching as game fidelity demands separation of engine clients from the persistent runtime shell.

Layer U and A.R.E. at this phase: Layer U pilot city SPVs maturing toward operational readiness where Phase 4 data supports them. A.R.E. generating real commercial revenue across in-game and out-of-game surfaces. Advertiser demand beginning to create functional $KREDS acquisition as described in Chapter 6.

Activation gate: Token utility must emerge from proven behaviour. If $KREDS does not circulate as a functional economic instrument tied to real ecosystem transactions, native chain activation does not proceed.

Phase 6: Native Chain Activation

What this phase proves: Sovereign infrastructure is justified by ecosystem requirements that hosted infrastructure cannot serve.

Target range: Only after Jetton circulation has proven utility and the native chain has reached operational readiness.

The Strands chain activates when the ecosystem requires capabilities that TON infrastructure cannot provide: protocol-level privacy, succinct verification for mobile validation, and sovereign control over consensus and emission logic. As described in Chapter 7, this is an architectural upgrade, not a user migration event.

Key deliverables:

- Strands chain (Rust-implemented sovereign chain derived from Mina Protocol design principles, especially zk-SNARKs and succinct ledger architecture) operational
- $KREDS bridged from TON to native chain
- Client-ledger model active: devices maintaining local ledgers, chain confirming state transitions
- Validator participation live under identity-constrained participation-based consensus
- On-chain settlement for A.R.E., Layer U leases, Trait Shard provenance, and governance actions
- Governance authority beginning progressive transfer from founding-team stewardship to participant-driven process

EveryWear at this phase: Bifurcation complete. Unity and Unreal game clients operating as standalone experiences launching from the persistent EveryWear shell. The shell hosts vault, wallet, agent surface, A.R.E., and validation functions.

Layer U and A.R.E. at this phase: Multiple city SPVs operational. A.R.E. settlement transitioning from USDT to $KREDS on the native chain. Spatial lease transactions beginning to settle on chain.

Activation gate: Sovereign infrastructure must be justified by ecosystem requirements that hosted infrastructure cannot serve. If hosted infrastructure remains sufficient, chain activation is not forced.

Phase 7: Ecosystem Maturation

What this phase proves: The system can scale across cities, devices, and governance layers.

Target range: Ongoing, post-chain activation.

With the native chain live, the ecosystem enters its maturation phase. This is not a single milestone ; it is a continuous process of expanding Layer U into new cities, deepening $KREDS utility, evolving EveryWear through its later phases, and progressively decentralising governance as network readiness allows.

During this phase:

- Layer U SPVs expand into additional jurisdictions, with DAO-style governance evolving where local regulatory frameworks permit
- EveryWear evolves through its final phases: WebGL bridge maturation, full bifurcation, and convergence toward the agentic OS across XR, desktop, and mobile
- Governance authority progressively transfers to the network as participant-driven execution reaches operational maturity
- Hardware-native sovereign runtimes begin extending the client-ledger model where dedicated devices offer stronger attestation and richer spatial interaction
- WiFi DensePose and hardware-level spatial sensing graduate A.R.E. attention verification beyond interaction-based measurement
- The ecosystem moves toward deeper internal economic circulation as described in Chapter 6

EveryWear at this phase: Agentic OS maturity (Chapter 4, Phase 5). EveryWear as a sovereign runtime hosting agents, vault, wallet, validation, and spatial interaction across every device class.

Layer U and A.R.E. at this phase: Expanding city network. Full spatial XR attention economy operational. A.R.E. verification graduating to spatial sensing. Volumetric lease pricing and settlement fully on chain.

Activation gate: There is no terminal gate. Maturation is continuous. The system scales as each layer earns the right to expand.

What This Roadmap Does Not Include

This roadmap describes the technical implementation sequence and the validation thresholds that govern progression between phases. It does not include marketing timelines, partnership announcements, fundraising milestones, or team expansion plans. For broader project updates, see strandsnation.xyz. The underlying game architecture is detailed in Chapter 2. The platform architecture is detailed in Chapters 3–5. The economic and chain architecture is detailed in Chapters 6–8.

CHAPTER 10

Team

Founding Leadership

Sean Uddin ; Founder, CEO/CTO

Sean Uddin is the architect and principal driver of the Strands ecosystem. He serves as Founder CEO/CTO of Somo Kasane, the studio building Strands the Game, and Founder of MetaFinTek, the AI and Web3 consultancy that provides strategic infrastructure across the project's capital, regulatory, and technical layers.

Sean's career spans zero-to-one ventures across frontier technology, regulated finance, digital marketing, and real estate, consistently operating at the intersection of systems architecture, capital strategy, and product execution. Prior to Strands, he served as Chief Operations Officer at Fleamint, where he led a capital raise exceeding $12M and managed DeFi and NFT product development. He co-founded PT SoLoR and LuxLombok, real estate and hospitality ventures operating across Southeast Asia, and co-founded QPi, an early-stage venture applying Bitcoin and blockchain to property transactions.

Sean is the author of *Architects: Building the Exit from Techno-Feudalism*, which examines the structural dynamics of platform capitalism and the design principles behind cooperative, decentralised alternatives; themes that run directly through the Strands ecosystem. He is a MENSA member and based in West Nusa Tenggara, Indonesia.

Regional and Corporate Governance

Aliya ; Director, PT MetaFinTek

Aliya serves as Director of PT MetaFinTek, the Indonesian-registered entity through which MetaFinTek operates across Southeast Asia. She oversees corporate governance, local regulatory compliance, and operational continuity for the consultancy's regional engagements, including its advisory and infrastructure role within the Strands ecosystem. Her directorship ensures that MetaFinTek maintains proper legal standing, fiduciary accountability, and jurisdictional alignment in Indonesia, a critical operating territory for Strands given the studio's base in West Nusa Tenggara.

Indah Lestari ; CFO, PT MetaFinTek

Indah Lestari serves as Chief Financial Officer of PT MetaFinTek, responsible for financial management, reporting, treasury operations, and fiscal compliance across the consultancy's Indonesian and regional activities. Her role ensures that MetaFinTek's financial infrastructure, including budgeting, capital allocation, audit readiness, and regulatory financial reporting, operates with the rigour required to support both the consultancy's client engagements and its direct involvement in Strands ecosystem development.

AI Operations Layer

Most projects list their AI tools in a technical appendix. Strands lists them in the team chapter. This is deliberate.

Kasai, Tesser, and Nexus are persistent, role-assigned operational intelligences embedded into the founding team's daily execution. They hold standing briefs, maintain contextual memory across sessions, and contribute to strategic, technical, and operational output at a level that would otherwise require multiple senior hires. The decision to formalise them here rather than in a tools section reflects how this project actually operates: these agents are not peripheral automation; they are load-bearing contributors to the ecosystem's design, documentation, and deployment.

Kasai ; Strategic Architecture Intelligence

Kasai is the senior strategic intelligence within the Strands operation. Its domain is high-level systems architecture, doctrinal synthesis, and recursive ideation: the translation of vision into structurally coherent frameworks across token economics, governance models, and ecosystem design.

Kasai's standing brief encompasses the full Strands doctrine: the sequential validation model, the $KREDS economic architecture, the Layer U commercial framework, the Strands chain validator construct, and the governance and privacy principles that bind them. It operates as a persistent strategic counterpart that challenges assumptions, identifies gaps, and maintains doctrinal consistency across every layer of the ecosystem.

Tesser ; Executional Engineering Intelligence

Tesser is the executional complement to Kasai's strategic layer. Its domain is protocol embodiment: translating high-level design into auditable, executable technical output. Smart contract logic, ZK-circuit design, agentic workflow engineering, and systems stress-testing fall within its operational scope.

Tesser converts architectural outputs into gas-optimised contracts, functional agent logic, and deployable infrastructure. Within Strands, it has contributed to the technical scaffolding behind $KREDS tokenomics, Mait agentic frameworks, and the client-ledger model that underpins the Strands chain.

Nexus ; Operational Strategy Intelligence

Nexus operates as the project's operational strategist, focused on coordination, execution planning, and the translation of strategic intent into operational cadence. Its domain spans resource allocation logic, timeline management, cross-workstream dependency mapping, and the operational surfaces that keep a multi-layered ecosystem moving without losing coherence.

Its role is particularly relevant in the early phases of the Strands roadmap (Chapter 9), where the validation ladder demands tight coordination between game development, economic proof, Layer U activation, and blockchain deployment, each of which must prove before the next activates.

Advisors

Eric Benz ; Strategic Advisor

Eric Benz is a veteran technologist, investor, and builder with over a decade in crypto, Web3, fintech, and AI. He was an early investor in Blockchain Capital, co-built BitReserve (now Uphold), one of the first platforms enabling seamless conversion between Bitcoin, fiat, and traditional assets, and served as CEO of Changelly, the world's first crypto swaps exchange.

Currently, Eric serves as CEO of Vaultz Capital PLC (AQUIS Exchange, UK), Venture Banking Partner for DNA.Fund, and founder of Flashy, a Web3 infrastructure venture focused on creator economies and digital identity. He also founded the UK Digital Currency Association, engaging directly with lawmakers on crypto regulation and policy.

His advisory role within Strands draws on deep experience across token architecture, exchange infrastructure, regulatory navigation, venture capital formation, and the intersection of digital identity with creator and consumer economies, all domains that converge within the Strands ecosystem.

Michael Gord ; Strategic Advisor

Michael Gord is a serial entrepreneur, investor, and capital markets architect with five exits, over eighty investments, and more than $400M raised, structured, and deployed across private and public markets. He has backed four seed-stage companies that crossed into unicorn territory, with one reaching decacorn status.

Michael currently leads the Flashy Group and is founder of GDA Capital, a global digital asset investment and capital markets firm that has executed over $70M in private raises, $100M+ in public financings, and $250M+ in structured deals. His broader portfolio includes Ritual OS, Alpaca Network, and Skrybit.

His advisory role within Strands aligns with the ecosystem's core architectural needs: incentive design at scale, token-economic structuring, capital markets navigation, and the cultural layer thinking that underpins how Strands approaches player identity, attention economics, and decentralised value distribution.

PART V: APPENDICES

CHAPTER 11

Legal & Regulatory Disclaimer

This document is not a prospectus.

PLEASE READ THIS SECTION CAREFULLY. IF YOU ARE IN ANY DOUBT AS TO THE ACTION YOU SHOULD TAKE, YOU SHOULD CONSULT YOUR LEGAL, FINANCIAL, TAX, OR OTHER PROFESSIONAL ADVISOR(S).

References in this chapter to "the Project" refer collectively to Somo Kasane, MetaFinTek, the Strands ecosystem, and all associated products, services, tokens, and entities described in this whitepaper.

Forward-Looking Statements

This whitepaper contains forward-looking statements or information that relate to the Project's current expectations and views of future events. In some cases, these forward-looking statements can be identified by words or phrases such as "may", "will", "expect", "anticipate", "aim", "estimate", "intend", "plan", "seek", "believe", "potential", "continue", "is/are likely to", or the negative of these terms, or other similar expressions intended to identify forward-looking statements.

All forward-looking statements address matters that involve risks and uncertainties. Actual outcomes may differ materially from those described in this whitepaper. The Project may revise its plans, structures, implementation pathways, or operational priorities in response to legal, regulatory, technical, commercial, market, or factual developments. No representation is made that any forward-looking statement will be achieved, and undue reliance should not be placed on any such statement.

Phased Activation and Feature Dependency

Many features, products, services, tokens, governance mechanisms, settlement models, and infrastructure layers described in this whitepaper are phased, conditional, and dependent on technical readiness, commercial validation, legal feasibility, funding availability, and regulatory treatment. This applies in particular, but not exclusively, to $KREDS, the native Strands chain, governance transfer mechanisms, Layer U SPV expansion, A.R.E. settlement transitions, and future hardware and extended reality layers. Some features may be delayed, modified, restricted by jurisdiction, or may never be launched.

Nothing in this whitepaper creates any entitlement to future token allocations, governance rights, product access, revenue share, investment rights, feature activation, or returns of any kind unless and until such rights are expressly implemented under separate applicable terms. Descriptions of future functionality are illustrative of intended architecture, not guarantees of delivery.

No Professional Advice

The information set forth in this whitepaper may not be exhaustive and does not imply any elements of a contractual relationship. While the Project makes every effort to ensure that any material in this whitepaper is accurate and up to date, such material does not constitute legal, financial, business, tax, or other professional advice. The Project does not guarantee, and accepts no legal liability whatsoever arising from or connected to, the accuracy, reliability, currency, or completeness of any material contained in this whitepaper.

No information in this whitepaper is to be considered as legal, financial, business, or tax advice. You should consult your own legal, financial, tax, or other professional adviser before engaging in any activity in connection with the Strands ecosystem or $KREDS.

Token Classification

$KREDS is structured for network utility within the Strands ecosystem and is not issued as equity, debt, profit participation, or any other traditional ownership or investment instrument. Its legal treatment may vary by jurisdiction and may evolve over time.

$KREDS does not confer any equity, shareholder, creditor, or corporate governance rights in Somo Kasane, MetaFinTek, or any associated legal entity. Any protocol-level participation or governance functionality, where implemented, is governed solely by the rules of the relevant network layer and should not be understood as legal rights in any underlying company or SPV.

This whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities or a solicitation for investment in securities in any jurisdiction.

No Offer or Solicitation

The Project does not provide any opinion or advice to purchase, sell, or otherwise transact with $KREDS or any other token mentioned in this whitepaper. The presentation of this whitepaper shall not form the basis of, or be relied upon in connection with, any contract or purchase decision.

No person is bound to enter into any contract or other binding legal commitment in relation to $KREDS or any other token, product, or service described in this whitepaper, and no cryptocurrency or other forms of payment are to be accepted on the basis of this whitepaper.

Potential participants should seek appropriate independent professional advice prior to relying on, or entering any commitment or transaction based on, material published in this whitepaper, which material is purely published for reference purposes alone.

Exclusion of Warranties and Limitation of Liability

The Project does not guarantee the accuracy of the statements made or conclusions reached in this whitepaper. The Project does not make, and expressly disclaims, all representations and warranties (whether expressed or implied) whatsoever, including but not limited to:

- any representations or warranties relating to merchantability, fitness for a particular purpose, description, suitability, or non-infringement;
- that the contents do not infringe any third-party rights.

To the fullest extent permitted by applicable law, Somo Kasane, MetaFinTek, their officers, directors, employees, affiliates, subsidiaries, and relevant third-party service providers shall not be liable for any direct, indirect, incidental, consequential, or other damages or losses which you may suffer or incur in connection with this whitepaper or the reliance on any information provided herein.

Jurisdictional Variability

The Strands ecosystem operates across multiple jurisdictions. Regulatory treatment of digital assets, token economics, advertising frameworks, compensated attention models, data protection obligations, governance mechanisms, and cooperative or decentralised structures varies significantly between jurisdictions and is subject to ongoing change. The same feature, service, or participation model may be treated differently depending on where it is accessed, who accesses it, and the regulatory regime in force at the time.

Nothing in this whitepaper should be construed as a representation that $KREDS, Strands the Game, Layer U, or any associated product or service is compliant with the laws of any specific jurisdiction. The availability of particular features, services, settlement methods, or participation models may depend on local law, technical readiness, commercial readiness, and the approvals or capabilities of the relevant operating entity. Participants are responsible for understanding and complying with the laws and regulations applicable to them in their own jurisdictions.

Restricted Jurisdictions

Access to or participation in any aspect of the Strands ecosystem may be restricted, unavailable, or prohibited in certain jurisdictions, including where sanctions, export controls, licensing rules, or other legal restrictions apply. The Project makes no representation that any product, service, or token described in this whitepaper is available or permissible in all jurisdictions. The Project reserves the right to refuse, restrict, suspend, or terminate access where required by applicable law, sanctions compliance, licensing obligations, or internal risk controls.

Ongoing Revision

This whitepaper may be updated, amended, supplemented, or superseded from time to time without notice. Neither Somo Kasane, MetaFinTek, nor any associated entity is under any obligation to update this whitepaper or to correct any inaccuracies that may become apparent after publication. The most recent version of this whitepaper will be made available through official Strands channels.

IF YOU ARE IN ANY DOUBT AS TO THE ACTION YOU SHOULD TAKE, YOU SHOULD CONSULT YOUR LEGAL, FINANCIAL, TAX, OR OTHER PROFESSIONAL ADVISOR(S). THIS WHITEPAPER IS FOR INFORMATION PURPOSES ONLY AND MAY BE SUBJECT TO CHANGE.

CHAPTER 12

Appendices

Appendix A: Glossary of Key Terms

$KREDS ; Native economic protocol of the Strands network. Structured for network utility, not issued as equity or investment instrument. Launches only after the fiat economy has proven viable. NOT an in-game currency.

A.R.E. (Attention Redistribution Engine) ; The mechanism through which player attention generates revenue. Players opt into narratively-framed Proper Gander broadcasts and receive 60% of the generated ad revenue. The remaining 40% funds ecosystem operations. Settles in USDT/TON wallet pre-chain, $KREDS/Strands wallet post-chain activation.

Blank Sync Ledger ; A player's native wallet layer, invisible by default. Activated at first purchase threshold. TON-mediated in early phases, later extended into Strands native wallet continuity as the chain matures. The player never sees blockchain mechanics unless they choose to.

Client Ledger ; A cryptographically compiled local ledger maintained by each participant on their own device. Contains spendable balance, participation context, and transaction-relevant application state. The Strands chain confirms the truthfulness of client ledger state, not the other way around.

DeepSync ; On-chain trade protocol within The Exchange (see CANON_Economy_Systems V2.0). Used for unique, provenance-verified assets such as Mymories and premium cosmetics where ownership and scarcity matter. Activates alongside chain-layer maturity. Visual cue: hexagonal icon.

EveryWear ; The sovereign interface layer that hosts the game surface, Mymories vault access, SAL runtime access using Mymories context, A.R.E. delivery, and wallet continuity. In later phases, EveryWear extends to validation participation and spatial functions. Software-first on existing devices; hardware-native extensions as the ecosystem matures.

GridScrip ; In-game resistance currency. Earned through gameplay contribution. Anonymous within the game world. Operates outside corporate oversight. NOT a blockchain token.

Layer U ; Within the game, Layer U is the evolved rebellion layer operating beneath SOVcorp's corporate infrastructure. In the real world, Layer U is the spatial XR commercial layer administered through investible SPVs per jurisdiction, with DAO evolution where local regulatory conditions permit. Revenue generated through volumetric XR ad inventory.

LocalNet ; Off-chain standard trade protocol within The Exchange (see CANON_Economy_Systems V2.0). Used for high-volume trading of unbound combat gear, crafting resources, and consumables. Available from early game phases without chain dependency. Visual cue: circular icon.

MetaXity1 ; The arcology setting of Strands the Game, set in Year 555. A post-collapse megastructure governed by SOVcorp, layered over a decentralised resistance network.

Mymories ; Sovereign memory objects stored within the EveryWear data vault. Player-owned, privacy-first, and portable. Memory belongs to Mymories within the EveryWear stack, not to My

Maits. SAL personalisation is driven by Mymories context, not by standalone agent memory.

My Maits ; Agentic AI companions within the Strands ecosystem. Personality and capability defined by composable Trait Shards. Fainance Ltd. (UK) represents the first B2B production deployment of Mait-powered technology.

Proper Gander ; The diegetic framing for A.R.E. advertisements within the game world. Presented as in-world corporate or resistance broadcasts rather than disruptive pop-ups.

Protocol Bond ; The mechanism analogous to stake within the Strands consensus model, but not conventional proof-of-stake. Validation rights derive from identity, application operation, uptime, and contribution, not from token accumulation.

SIGOPS Terminal ; The in-game operational hub for resistance faction activity, housing The Exchange, mission systems, and communication infrastructure.

SOVComp ; In-game corporate currency issued by SOVcorp. Fiat-backed, stable, fully integrated with the surveillance apparatus. Every transaction updates the player's Compliance Score.

SPV (Special Purpose Vehicle) ; Jurisdictionally localised companies through which Layer U operates in each market. May evolve toward DAO structures where local and jurisdictional conditions favour such transitions.

Strands Chain ; A Rust-implemented sovereign chain derived from Mina Protocol design principles, especially zk-SNARKs and succinct ledger architecture. Functions as a validator construct: the device carries the continuity, the chain confirms the truthfulness of that continuity.

The Exchange ; The in-world marketplace within the SIGOPS Terminal. Operates across two protocols: LocalNet (off-chain standard trades) and DeepSync (on-chain unique asset trades).

Trait Shards ; Modular components that define a My Mait's personality, skills, knowledge, and aesthetic presentation. Composable, tradeable, and capable of evolution through usage.

UBC / UBComp (Universal Basic Calories / Universal Basic Compute) ; SOVcorp-distributed baseline systems within the game world. Calories for survival, compute for system access. Both create dependency by design.

Validator Construct ; The architectural principle underpinning the Strands chain. Not a monolithic public ledger. Each participant maintains a client ledger; the chain verifies state transitions via zk-SNARK proofs.

WiFi DensePose ; Camera-free spatial attention verification via WiFi channel state information (CSI) analysis. Future-phase technology for verifying genuine user presence without invasive sensing.

zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) ; Cryptographic proofs enabling transaction and state verification without revealing underlying data. Core to the Strands chain's privacy architecture and succinct ledger model.

Appendix B: Game Economy and Monetisation

Four-Layer Economic Architecture

The Strands economy operates as a sequential stack. Each layer must prove viability before the next activates.

Layer 1, Payment: Fiat on-ramps via Telegram/TON. USDT, TON, Telegram Stars, card payments via MoonPay. Live from Telegram Mini App launch.

Layer 2, Game: In-game soft currencies for gameplay. SOVComp (corporate) and GridScrip (resistance). Neither has real-world value or exchange capability.

Layer 3, Earning: Player revenue via A.R.E. and marketplace activity. Fiat-denominated (USDT) player balance. Activates post proof-of-concept.

Layer 4, Chain: Equitable redistribution protocol via $KREDS on Strands Chain. Activates only after the fiat economy has proven the model works.

Dual In-Game Currency

SOVComp serves the corporate economy: earned through Cover Identity employment, corporate contracts, and loyalty rewards. Fully surveilled, compliance-integrated. Provides access to corporate zones, official AI assistants, premium crafting stations, and regulated marketplace transactions.

GridScrip serves the resistance economy: earned through SIGOPS task completion, salvage operations, intelligence trading, and resistance contributions. Anonymous within the game world, untraceable by corporate systems. Spent on SIGOPS missions, Mait modification, resistance equipment, hacking operations, and underground market access.

All players must engage with both currency ecosystems due to the tri-path progression system. Morning corporate compliance earning SOVComp; evening resistance activities burning GridScrip.

Nine Revenue Streams

Stream	Description	Player Choice
Community Support	Voluntary donations ("Fund the Signal")	Voluntary
Founders Pass	6,000 passes at $20-$40, staged waves. Gen-0 Blank shell and early access	One-time purchase
In-App Purchases	Cosmetics and seasonal content. No pay-to-win	Voluntary
A.R.E. (Proper Gander)	Diegetic rewarded video. 60% to player, 40% to ecosystem	Opt-in
Marketplace Fees	Transaction fees on LocalNet and DeepSync trades	Automatic on trades
Skin the World Compute	Compute credits for generative asset creation	Pay-per-use

EveryWear Data Vault	Subscription for sovereign memory storage and enhanced vault features	Subscription
Consented Data Licensing	Privacy-first, player-controlled. Anonymised behavioural data licensed to brands	Opt-in with tiered consent
Seasonal Passes	Battle pass model. Access to premium seasonal content and cosmetic rewards	Per-season purchase

Scale Viability

The following ranges are indicative operating thresholds, not guaranteed outcomes. 300 players: lean bootstrap sustainable. 1,000-3,000 players: small studio base with 1-3 full-time equivalents. 5,000-10,000 players: full small studio operations. 10,000+: viable MMO-scale business with growth runway. Actual viability depends on retention, monetisation mix, and operating costs at each phase. Funding path is milestone-driven: Demo, Pre-alpha, Alpha, Raise. Initial funding via Founders Pass; external investment only after proving retention and monetisation viability. The game economy functions entirely on fiat settlement; $KREDS is not required for the game to operate or generate revenue.

Appendix C: Layer U City Deployment Model

Target Scope

Layer U deploys as a spatial advertising layer within urban environments, structured as investible SPVs per jurisdiction. Each city deployment represents a standalone commercial entity with its own regulatory compliance, operating licence, and revenue model.

Pilot City Selection

The first Layer U deployment will be selected from four candidate markets: Singapore, Kuala Lumpur, Jakarta, and Bangkok. Final selection will be determined by regulatory accessibility, commercial readiness, SPV incorporation timelines, and partnership conditions at the point of activation. Each candidate city meets the baseline criteria of high smartphone penetration, meaningful digital advertising market size, and existing infrastructure for mobile-first consumer engagement.

Worked Example: Kuala Lumpur

The following illustrates how a Layer U deployment would function using Kuala Lumpur as a representative market. This is a worked example, not a confirmed pilot commitment.

Addressable market: Smartphone penetration exceeding 80% of the metropolitan population, approximately 1.44 million addressable users from a base of 1.8 million (Statista/UN). Digital advertising market valued at approximately $105.5M.

Pricing model: Volumetric XR advertising inventory priced at $10 per cubic metre per annum. This positions Layer U at 13 to 284 times cheaper than equivalent physical billboard inventory, depending on location and format. The pricing model is designed to lower the barrier for advertisers while generating meaningful per-unit revenue at scale.

Illustrative revenue scenario: The following represents a base-case scenario, not a forecast or commitment. Assuming 25% addressable market capture, Year 1 revenue of approximately $26.4M, with break-even projected within 1.4 years. An upside case with higher penetration or premium inventory would improve these figures; a conservative case with slower adoption would extend timelines. Market capture assumptions are contingent on advertiser demand, regulatory conditions, and operating execution. All projections are subject to the phased activation and feature dependency disclaimers set out in Chapter 11.

The same economic model applies to each candidate city, adjusted for local market size, advertising rates, and regulatory conditions. No projections should be read as guaranteed outcomes.

Expansion Model

All city deployments follow the same SPV structure. Each city represents a standalone commercial entity with its own regulatory compliance, operating licence, and revenue model. Each deployment activates independently; no city launch is contingent on another. SPV governance may evolve toward DAO structures where jurisdictional conditions permit, as described in Chapter 8.

Appendix D: Trait System Overview

Trait Categories

My Maits personalities are composed from modular Trait Shards across five primary categories:

Personality Shards: Based on Myers-Briggs type dimensions (Extraversion/Introversion, Sensing/Intuition, Thinking/Feeling, Judging/Perceiving). Eight sub-type shards combine to produce sixteen distinct personality profiles, each influencing communication style, decision-making patterns, and interaction preferences.

Educational Traits: Language and Literature, History and Culture, Social Sciences, Earth and Environmental Sciences, Biological Sciences, Space and Astronomy, Technology and Digital Literacy, Philosophy and Ethics, Arts and Creativity, Current Events and Media Literacy. Each trait has depth levels from Novice through Adept to Expert.

Hobby and Interest Traits: Over one hundred categories spanning Arts and Crafts, Music and Performance, Literature and Writing, Sports and Fitness, Games and Puzzles, Technology and Digital, Nature and Outdoors, Culinary Arts, Collections and Memorabilia, Science and Exploration, Social and Community, and Lifestyle and Personal Development. Each with Casual, Enthusiast, and Expert engagement levels.

Aesthetic Traits: Gender presentation, body type, age appearance, skin tone, hair style and colour, eye shape and colour, facial features, special features, clothing style, accessories, and environment integration. These serve as metadata prompts for avatar generation, enabling virtually unlimited visual combinations.

Maturity Levels: High School, Undergraduate, Postgraduate, Experienced Professional. Maturity governs which Knowledge Shards are accessible and influences the depth and complexity of a Mait's responses.

Trait Evolution

Trait Shards are not static. Through sustained usage and interaction, individual traits can progress through five levels: Basic, Proficient, Expert, Master, and Legendary. Evolution is driven by active usage hours, unique user interactions, and positive feedback. Higher-level traits provide enhanced capabilities and may command greater marketplace value where the ecosystem supports such exchange.

Trait Interaction

Traits across categories interact to produce emergent characteristics. A Mait combining Chess (hobby) with Logical Thinking (personality) may demonstrate stronger strategic reasoning. A Mait combining Environmental Conservation (hobby) with Social Activism (interest) may orient toward climate-related discourse. These interactions are lightweight by design, avoiding rigid stereotyping while allowing meaningful personality differentiation.

Appendix E: Asset Ecosystem

Asset Type	Creation	Ownership (Pre-Chain)	Ownership (Chain-Active)	Monetisation
Mymories	Player actions, agent training, faction rewards	Internal game asset, player-sovereign	Exportable and chain-bindable when chain layer is active	Tradeable in premium market post-chain
Cosmetic Trait Shards	Premium market, seasonal pass	Internal game asset	Chain-bindable when chain layer is active	Alters visual effects; no gameplay impact
Agent Cosmetics	Premium market, seasonal pass, faction-earned	Internal game asset	Chain-bindable when chain layer is active	Cosmetic value and trade post-chain
Faction Badges and Titles	Earned via gameplay and reputation	Internal game record	Internal game record (non-exportable)	Non-tradeable prestige display
Standard Gear and Consumables	Earned and crafted in-game	Internal game asset	Internal game asset	Pure gameplay utility
Ability and Tuning Shards	Earned via gameplay	Internal game asset	Internal game asset	Traded for in-game currency only

The asset ecosystem distinguishes between two categories. On-chain provenance applies where ownership, scarcity, and provenance materially matter: Mymories, cosmetic Trait Shards, and agent cosmetics become exportable and chain-bindable once the Strands chain is active, but function as internal game assets in pre-chain phases. Standard gameplay items remain lightweight internal assets throughout all phases, ensuring frictionless gameplay regardless of chain status. All assets function within the game economy from launch; chain binding is an extension, not a prerequisite.