

AgentOS Technologies Ltd.

The Operating Ecosystem for Autonomous Intelligence

Strategic and technical whitepaper covering product architecture, security, marketplace economics, \$sAGENT policy, the AgentOS Wallet plan, and the token flywheel.

Super AgentOS	DEVELOPER ECONOMY	WALLET + FLYWHEEL
---------------	-------------------	-------------------

Understand	Plan	Authorize	Execute	Verify	Deliver
------------	------	-----------	---------	--------	---------

VERSION 1.1 / SEPTEMBER 2026

Official contract: 2Fob54QUhUbP9jv6h5XAh3PgB1kcULR6LXbxSzuwpump

Document Status and Claim Discipline

This whitepaper defines the AgentOS thesis, product system, architecture, economic design, security boundaries and roadmap. It is not a claim that every described capability is already deployed. Each major statement belongs to one of three classes:

Status	Meaning	Examples
Current foundation	Implemented, documented or represented in the current codebase and product architecture.	Studio, Media Studio, Artifacts, Projects, Library, Vault, stores, sessions, execution records, cancellation and recovery foundations.
Committed direction	Planned capabilities that are announced as available only after validation.	Connected Intelligence selection, production hardening, cross-platform clients, university activation and Hackathon Series 1.
Exploratory research	Strategic work requiring technical, capital, legal, safety or market validation before commitment.	A proprietary AgentOS model family, full FFP deployment, expanded token governance and the 5D Universe.

Non-overclaim rule AgentOS will not describe FFP as live consensus, a proprietary model as trained, an application as published, a token benefit as legally enforceable, or a task as completed until evidence exists.

Abstract

Digital intelligence has advanced faster than the systems used to operate it. Users can converse with capable models, but real work still fragments across chats, applications, automation tools, credentials, files, developer environments and disconnected agents. The result is repeated context, weak permission boundaries, fragile integrations and uncertain accountability.

AgentOS is designed as the operating ecosystem between human intent and autonomous execution. Super AgentOS is the permanent command and control layer. It maintains workspace context, discovers available capabilities, plans supported operations, enforces permissions and approvals, coordinates execution, records provenance, supports cancellation and recovery, and delivers the verified result. Connected Intelligence can extend interpretation, reasoning, generation and verification, but it does not become the product identity and cannot bypass AgentOS control.

The ecosystem includes Apps, Skills, Prime Agents, Primeflows, Projects, Media Studio, Artifacts, Library, Memory, Vault, Universal MCP, developer tooling and a future FFP trust layer. The \$sAGENT token is intended to support ecosystem access and economic coordination where legally and technically appropriate, without being presented as equity, a guaranteed-return instrument or a substitute for product adoption.

The long-term objective is not another conversational interface. It is a persistent, secure and extensible operating environment in which one command can become authorized, observable and recoverable work across web, mobile, desktop and, eventually, spatial interfaces.

Executive Summary

- **Problem.** Intelligence is abundant, but execution remains fragmented across tools, accounts, credentials and interfaces.
- **Solution.** AgentOS unifies command, context, capabilities, permissions, execution, recovery and distribution.
- **Core identity.** Super AgentOS is always active and remains the assistant and operating authority, whether Native or Connected Intelligence is selected.
- **Economic layer.** Apps and Skills are monetizable; Agent Credits account for platform compute; \$sAGENT may coordinate ecosystem utility under transparent legal and treasury controls.
- **Trust layer.** Vault, approvals, provenance, redaction, cancellation, recovery and the Panic control are product-level safety mechanisms.
- **Roadmap.** The immediate programme prioritizes production hardening, AgentOS v1, cross-platform clients, university activation, Mini Hackathon Series 1 and credible CEX readiness.
- **Research.** A proprietary AgentOS model family is under evaluation, not decided. FFP and the 5D Universe remain future directions until validated.

Contents

- 1\. Vision and Market Thesis
- 2\. Product Definition and Principles
- 3\. System Architecture
- 4\. Super AgentOS and Intelligence Modes
- 5\. Core Product Components
- 6\. Execution, Safety and Security
- 7\. Developer and Ecosystem Architecture
- 8\. Business Model and Agent Credits
- 9\. \$sAGENT Utility and Economic Policy
- 10\. FFP: Future Verification Fabric
- 11\. Proprietary Model Research Programme
- 12\. Roadmap and Delivery Gates
- 13\. Adoption Strategy
- 14\. Governance, Metrics and Operations
- 15\. Risks and Mitigations
- 16\. Legal and Regulatory Position
- 17\. Conclusion
- Appendices: Status Matrix, Glossary and Audit

1. Vision and Market Thesis

The current intelligence market is organized around isolated products. A person may use one system for conversation, another for code, another for automation, another for research, and separate applications for communication, finance, storage and deployment. Each system owns only a partial view of the user's work. Context is copied manually, permissions are inconsistent and outputs often stop at advice rather than becoming verified execution.

Core thesis AgentOS is the operating ecosystem where intelligence, capabilities and execution persist. Chat is the command surface; the workspace is the system of record.

1.1 The fragmentation problem

- **Context fragmentation.** Projects, decisions, files and instructions are repeatedly reintroduced across tools.
- **Capability fragmentation.** Applications, agents, APIs and automations are not discoverable through one consistent contract.
- **Credential fragmentation.** Secrets are copied into chats, local files or third-party dashboards without a unified permission model.
- **Execution fragmentation.** Reasoning and action are mixed, making it difficult to know what was proposed, authorized or actually completed.
- **Identity fragmentation.** Every connected model appears as a competing assistant instead of a replaceable resource inside one operating system.
- **Economic fragmentation.** Builders struggle to distribute and monetize reusable capabilities without rebuilding identity, billing, permissions and discovery.

1.2 The AgentOS opportunity

AgentOS can become a neutral operating layer above individual models and below the user's goals. Models, agents, applications and tools can improve or change over time without forcing the user to rebuild their workspace. The long-term defensibility comes from the operating graph: persistent context, installed capabilities, trusted permissions, reusable PrimeFlow executions, developer distribution, execution history and cross-device continuity.

This position avoids unnecessary model-war dependence. AgentOS can improve through Connected Intelligence today, evaluate a proprietary model tomorrow, and adopt future reasoning systems without redefining the product. The product remains AgentOS.

1.3 Target users

User group	Primary need	AgentOS value
Individuals and operators	Complete work without manually switching across multiple systems.	One command surface, persistent context, installed capabilities, approvals, execution and recovery.
Builders and developers	Build, test, publish and distribute reusable capabilities.	SDK, Code Studio, Appstore, Skill Store, Vault, permissions, analytics and versioning.
Teams and enterprises	Control access, data boundaries, integrations and operational reliability.	Workspace policies, role controls, auditability, private connections and shared context.
Institutions and universities	Train builders through real production infrastructure.	Workshops, credits, Apps, Skills, PrimeAgent operators, PrimeFlow executions, incubation and distribution.
External products and agents	Become callable inside a broader workspace.	Universal MCP, bearer-token connectivity and SDK-based Appstore integration.

2. Product Definition and Principles

2.1 Official definition

AgentOS is an operating ecosystem for autonomous intelligence. Super AgentOS is its always-active command, context, permission, execution, recovery and delivery layer. It routes authorized work into Native functions, Connected Intelligence, Apps, Skills, PrimeAgent operators, PrimeFlow executions, Universal MCP tools and future FFP primitives.

Super AgentOS is not a foundation model. It is also not a thin wrapper around one. It is the persistent operating authority that remains functional when no external intelligence connection exists and remains in control when a connection is selected.

2.2 Design principles

Principle	Operational meaning
AgentOS remains the identity	Connected services appear as execution metadata, not as replacement assistants.
Native never disappears	The operating runtime remains usable with zero external connections.
Context is permissioned	Only authorized workspace information is disclosed to any selected intelligence or capability.
Reasoning is not execution	A plan cannot mutate the system until capability, permission, risk and approval checks pass.
Real data or honest emptiness	No fake listings, validators, proof events, ratings, earnings, executions or adoption metrics.
Safety is operational	Stop, cancel, panic, retry, recover, revoke and audit are first-class behaviors.
Server authority	The server resolves identity, context, permissions, credentials, task state and execution.
Replaceability	Intelligence vendors, model versions, tools and consensus systems are replaceable.
Cross-device continuity	Web, mobile and desktop share sessions, Projects, approvals, executions and notifications.
Evidence before claims	A feature is shipped only after real end-to-end verification.

2.3 Product terminology

User-created operators are PrimeAgent operators. Reusable execution graphs are PrimeFlow executions. Connected models, agents or private endpoints are Connected Intelligence. The system may use Native, Single Intelligence or Consensus modes. Apps and Skills are distribution and monetization surfaces; PrimeAgent operators and PrimeFlow executions are not treated as paid marketplace products by default.

Naming boundary AgentOS does not present itself as model-branded or vendor-branded. The message author remains Super AgentOS. An exact selected model may appear only as compact execution metadata.

3. System Architecture

3.1 Layered architecture

Layer	Responsibility	Representative components
Experience	Human interaction across devices.	Home, NL Studio, Primeflow Builder, Code Studio, Media Studio, Artifacts, mobile and desktop clients.
Operating	Identity, context, task planning, permissions, approvals, execution state, recovery and delivery.	Super AgentOS, Context Engine, Task Engine, Execution Engine, notifications and Recovery Center.
Capability	Discoverable and callable resources.	Apps, Skills, Prime Agents, Primeflows, Universal MCP tools and files.
Security	Secrets, grants, policies, redaction, audit and emergency control.	Vault, runtime grants, permission checks, approval tokens, Panic control and provenance.
Intelligence	Optional interpretation, reasoning, generation, verification and synthesis.	Native rules, user-connected models, private endpoints and future proprietary AgentOS models.
Distribution and economics	Publishing, installation, usage accounting and monetization.	Appstore, Skill Store, Artifacts, developer centre, Agent Credits and \$AGENT integration.
Future protocol	Independent verification, coordination and settlement where justified.	FFP-compatible consensus records and future validators.

3.1.1 Device-native Home

Authenticated AgentOS opens into one operating Home backed by server-authoritative workspace state. Laptop presentation uses desktop, taskbar, launcher and activity-center patterns; mobile presentation uses an app grid, dock, drawer and notification shade. Both expose the same canonical Apps, Tasks, Projects, Media Studio, Artifacts, Library, permissions and notifications. Library serves as the durable file and capability layer: File Explorer presentation on laptop and Files presentation on mobile may differ without duplicating storage or authority.

3.1.2 Account Center

Settings is the device-adaptive Account Center for identity, workspace roles, permissions, devices, sessions, privacy, notifications, installed-capability readiness, connected intelligence, usage and developer access. It consumes the same server-authoritative records used by execution and security layers, preserves existing deep links, keeps secrets in Vault and leaves system-level Chronos and Panic ownership with Home. Laptop uses a two-pane control panel; mobile uses a full-screen settings list and drill-down without changing those authority boundaries.

3.2 Canonical request flow

USER INTENT	CONTEXT	PLAN	POLICY	EXECUTION	RESULT
-------------	---------	------	--------	-----------	--------

- 1 The user submits a request through Super AgentOS.
- 1 The Context Engine assembles the minimum authorized session, Project, Memory, Library and capability context.
- 1 Native or selected Connected Intelligence interprets the request and produces a response or candidate plan.
- 1 Super AgentOS validates capability existence, schemas, permissions, risk level and approval requirements.
- 1 The Execution Engine runs the authorized operation with idempotency, cancellation, timeout and recovery controls.
- 1 The actual execution result is validated, persisted, summarized and delivered with provenance.

3.3 Capability Graph and Runtime Registry

AgentOS treats executable resources as canonical capabilities rather than scattered integrations. Each capability publishes a versioned contract describing provider, inputs, outputs, permissions, dependencies, compute expectations, health, confidence, priority and fallback policy. The Runtime Registry is the authoritative inventory available to a user or workspace.

A connected model cannot invent an App, Skill, Prime Agent, Primeflow or MCP action that is absent from the graph. It can propose only capabilities supplied to it by AgentOS.

3.4 Context architecture

Context is assembled as a versioned package, not an uncontrolled dump of the workspace. It can include recent conversation, active Project metadata, saved instructions, approved Memory records, selected Library assets, attachments, execution results and capability manifests. Vault secret values, unrelated workspace data and hidden credentials are excluded.

Context category	Default treatment	User control
Recent conversation	Included within bounded limits.	Session and per-message controls.
Project context	Included when a Project is active and authorized.	Attach, switch or detach Project.
Memory	Relevant approved records only.	View, edit, delete and disable.
Library assets	Selected or relevance-matched metadata/content.	Attach or remove items.
Files	Explicit per-message or Project attachment.	Upload, preview, remove and govern access.
Execution results	Included when needed for continuation.	Inspect logs and outputs.
Vault secrets	Never placed in normal context.	Grant, revoke and audit use.

3.5 Task and execution state

AgentOS separates a task-the durable representation of user intent-from an execution-the attempt to perform work. Tasks can have parent and root identifiers, context and graph versions, planner version, priority, retry count, timestamps and metadata. Executions record the actual lifecycle, outputs, failure category, duration, usage and recovery state.

- Queued, planning, awaiting approval, running, paused, completed, partially completed, failed and cancelled states.
- Idempotency keys to prevent duplicate mutations.
- Safe cancellation that propagates to active streams and capability calls.
- Retries that distinguish transient failure from invalid input or revoked permission.
- Recovery records that allow resume, retry, cancel or inspect.

- Provenance that identifies the selected intelligence, capability contract and execution result without exposing secrets.

4. Super AgentOS and Intelligence Modes

4.1 Ownership boundary

Super AgentOS owns conversation and session continuity, active workspace and Project context, Memory, Library context, capability discovery, task and execution state, permissions, approvals, idempotency, retries, cancellation, recovery, logs, notifications, provenance and final delivery.

Connected Intelligence may contribute interpretation, reasoning, decomposition, candidate plans, structured generation, file understanding, verification and synthesis. It must never decrypt credentials, mutate storage directly, bypass permissions, invent execution success, alter task records or expose hidden internal reasoning.

4.2 Native mode

Native mode is always available. It handles supported system commands, deterministic navigation and object management, Project and session operations, Library and Memory lookup, explicitly selected Apps and Skills, Primeflow execution, Prime Agent delegation, MCP invocation, approvals, notifications, stop and panic operations, and known structured intents.

Native honesty rule Native completes every deterministic part it can. When deeper reasoning is required, it states the missing cognitive requirement and offers available Connected Intelligence choices without disabling Super AgentOS.

4.3 Single Intelligence

Single Intelligence uses one user-selected connection. It can interpret, reason, generate, propose, self-review and synthesize. It is not consensus. The user may set a default for the account, workspace or session, or use a one-message override. The exact selected model is visible, while the assistant identity remains Super AgentOS.

Production credentials are user-owned and stored in Vault. They are consumed through short-lived server-side runtime grants. No credential may enter browser state, HTML, messages, logs, events, task metadata or error output.

4.4 Standard Consensus

Standard Consensus is a committed architecture stage after Single Intelligence is stable. It requires at least two eligible connections: one or more independent workers and a verifier. Workers receive the original request, normalized brief, identical authorized context and fixed evaluation criteria. They do not see each other's first-round outputs.

The engine compares claims, evidence, context consistency and contradictions; preserves material dissent; and produces one of four decisions: consensus reached, conditional consensus, split decision or consensus failed. Consensus never authorizes execution. Any proposed action still passes through AgentOS permissions and approval.

Mode	Available when	Primary use	Execution authority
Native	Always	System commands, known operations and deterministic workspace work.	AgentOS only
Single Intelligence	One healthy user connection	Open-ended interpretation, reasoning, generation and synthesis.	AgentOS only
Standard Consensus	Two or more eligible connections and policy permission	Independent worker comparison, verification and dissent preservation.	AgentOS only

Mode	Available when	Primary use	Execution authority
FFP-backed consensus	Future protocol deployment	Potential independent verification and protocol-grade coordination.	Separated from execution

5. Core Product Components

5.1 Studio

Studio is the primary work surface and contains NL Studio, Primeflow Builder and Code Studio under one persistent workspace. Switching modes must preserve the current session, Project, files, capabilities and execution state.

- **NL Studio.** Conversation-first command surface with streaming, semantic command starters, consolidated resource selection, visible context percentage, exact user-owned External Intelligence selection, files, approvals, execution cards, stop, retry and session history.
- **Primeflow Builder.** Visual and natural-language construction of reusable execution graphs with prompts, Skills, Apps, Prime Agents, Vault permission nodes, MCP actions, triggers, branches, retries and outputs.
- **Code Studio.** Developer mode for repository work, SDK apps, tests, deployment preparation, terminal tasks and implementation planning.

5.2 Projects, Library and Memory

Projects are durable context containers for sessions, files, Primeflows, Prime Agents, Apps, Skills and outputs. Library is the durable inventory of installed and created assets. Memory stores user-approved durable knowledge and preferences; it is not a secret store and must remain editable and deletable.

5.3 Apps and Appstore

Apps are larger product surfaces integrated through the AgentOS SDK. A valid App can expose actions, interfaces, permissions, supported devices, dependencies and health. Appstore discovery requires registration, metadata, verification, listing review and working install/open integration. Installation creates durable ownership and places the App in Library.

5.4 Skills and Skill Store

Skills are modular capabilities callable by Super AgentOS, Prime Agents and Primeflows. A Skill has a versioned contract, supported surfaces, permissions, dependencies and examples. Skills may be free or paid where the monetization backend is active. Installed Skills appear in Library and can be governed or removed.

5.5 Prime Agents

Prime Agents are user-created operators. They may be private, incognito or public according to product policy. A Prime Agent can have instructions, permitted tools, Memory scope, Project scope, schedule eligibility and explicit visibility. Prime Agents are not the default monetization layer; privacy and capability isolation are more important than marketplace exposure.

5.6 Primeflows

Primeflows are reusable execution graphs. They can be created manually or from natural language, run on demand or on a schedule, and combine prompts, Skills, Apps, Prime Agents, Vault permissions, MCP tools, triggers and outputs. Public Primeflows may be discovered, starred and forked, but forking must not copy private data, Memory or secrets.

5.7 Vault

Vault is the secure permission layer for credentials and secrets. Secret values remain encrypted and hidden by default. AgentOS exposes labels, assignments, status, permission reason, grants, revocation and audit state-not plaintext. Runtime access is short-lived, server-side, scoped to a specific operation and cleaned after use.

5.8 Universal MCP

Universal MCP connects outside tools and external agents to Super AgentOS routing. MCP tools are not automatically Appstore Apps. SDK Apps and MCP connections have separate registration, permission, health, listing, installation and monetization rules.

5.9 Notifications, Recovery and Panic

AgentOS must notify users when tasks complete, fail, require approval or recover. Recovery Center exposes retry, resume, cancel, rollback where supported and detailed failure state. The Panic control provides a system-wide emergency stop for active work, blocks new actions until reviewed and supports revocation of temporary access. It is an operational safety mechanism, not a decorative button.

6. Execution, Safety and Security

6.1 Action authorization pipeline

PROPOSAL	SCHEMA	PERMISSION	RISK	APPROVAL	EXECUTION
----------	--------	------------	------	----------	-----------

- 1 Confirm that the referenced capability exists and is healthy.
- 1 Validate all inputs against the published contract.
- 1 Resolve the user, workspace and Project permission boundary.
- 1 Classify the action by reversibility, external effect and sensitivity.
- 1 Request explicit approval when policy requires it.
- 1 Execute once using an idempotency key and validate the real result.

6.2 Security controls

Control	Required behavior
Tenant isolation	No cross-user, cross-workspace or cross-Project access outside explicit sharing.
Secret protection	No plaintext credentials in browser state, messages, model context, logs, events, tasks or errors.
Approval integrity	Short-lived, scoped, replay-resistant approval tokens tied to the exact action.
Prompt-injection defense	Connected Intelligence cannot request secrets, expand its capability set or bypass policy.
Private endpoint safety	HTTPS, DNS validation, redirect limits, timeout and response limits; private and metadata networks blocked unless enterprise policy explicitly allows.
Redaction	Secret-like values removed from persisted messages, logs and shared outputs.
Cancellation	Active streams and tools receive stop signals and cannot continue silently.
Auditability	Security-relevant grants, denials, approvals, executions and revocations are recorded.
Data lifecycle	Export, deletion, retention and backup behavior are documented and testable.

6.3 Safety model

AgentOS safety is based on control of execution rather than reliance on a model refusing every unsafe request. The operating layer constrains context, capabilities, permissions, approvals and external effects. Connected Intelligence can recommend; AgentOS decides whether a recommendation is valid and executable.

Safety principle A fluent answer is not proof. A plan is not authorization. A consensus result is not execution. Only a validated AgentOS execution record can establish that work occurred.

7. Developer and Ecosystem Architecture

7.1 Developer experience

Enterprise developers receive SDK access, developer tooling and publishing controls. The development lifecycle should support registration, local testing, permission declaration, health checks, screenshots, listing metadata, versioning, release notes, staged publishing, analytics and update management.

7.2 SDK versus MCP

Dimension	AgentOS SDK	Universal MCP
Primary purpose	Build discoverable AgentOS Apps and Skills.	Connect existing tools and external agents.
Distribution	Appstore or Skill Store after validation and review.	Connection management; not automatically listed as an App.
Interface	AgentOS-native contracts, permissions, metadata and lifecycle.	Tool/action contracts exposed by the external service.
Monetization	Eligible for marketplace pricing and developer earnings.	Separate commercial arrangement unless wrapped into an approved product.
User experience	Install, configure, open, update and manage in Library.	Connect, grant permission, invoke, inspect and revoke.

7.3 Developer economics

Paid Apps and Skills may use developer-set pricing. The intended marketplace policy allocates 70% of eligible net marketplace revenue to the developer and 30% to AgentOS, subject to final billing, refund, tax, fraud, legal and platform-fee rules. Free products remain supported. Earnings, installs, ratings and usage must come from real telemetry.

7.4 Ecosystem flywheel

USERS	TASKS	BUILDERS	APPS + SKILLS	DISTRIBUTION	MORE USERS
-------	-------	----------	---------------	--------------	------------

Users create demand through real tasks. Builders convert repeated needs into Apps and Skills. The stores distribute those capabilities. Super AgentOS makes installed capabilities easier to use, increasing successful execution and creating more demand. Universities, hackathons and external products expand the builder base and domain coverage.

8. Business Model and Agent Credits

8.1 Revenue model

Revenue stream	Description	Maturity
Subscriptions	Free, Pro, Enterprise Plus and Enterprise Max plans with different limits, controls and developer access.	Committed product model
Agent Credits	Usage accounting for platform-provided compute, execution and premium runtime functions.	Foundation and committed expansion
Marketplace fees	Platform share on eligible paid Apps and Skills.	Planned with backend enforcement
Enterprise services	Private deployments, advanced controls, integrations, support and institutional programmes.	Strategic
Developer and ecosystem services	Publishing, distribution, incubation and selected partnership programmes.	Strategic
Future protocol services	Verification, routing or settlement fees if FFP becomes operational.	Exploratory

8.2 Agent Credits

Agent Credits are a platform accounting unit, not the same as \$sAGENT and not a generic cash balance. Credits can meter conversations, code tasks, file operations, PrimeAgent runs, PrimeFlow execution, platform-hosted intelligence and other resource-intensive work. User-owned Connected Intelligence usage remains distinguishable from platform credits.

- Display current balance, allowance, UTC reset window, usage history and available-reset count.
- Keep usage activity separate from estimated provider cost.
- Attribute user-owned Connected Intelligence tokens without consuming platform credits.
- Explain which task or capability consumed credits.
- Use honest unavailable states when telemetry is incomplete.
- Keep marketplace product pricing separate from platform compute accounting.
- Avoid claiming exact monetary cost when reliable pricing is not available at execution time.

Credit cycles and available resets

Retail Agent Credit cycles last seven days and enterprise cycles last five days. Every cycle starts with exactly two resets. Unused resets expire at the cycle boundary and never carry over.

A reset is redeemable only after the active balance reaches zero. Redemption atomically consumes one current-cycle reset and restores the full current-plan allowance without restarting or extending the cycle. Row locking and idempotent usage records prevent double redemption and duplicate debits.

Platform executions reserve a credit before work begins; an exhausted balance blocks platform work. Connected Intelligence invoked through a user-owned provider account records model and token activity but consumes zero platform credits.

8.3 Plans and access

Plan	Primary audience	Representative access
Free	New and retail users	Baseline Super AgentOS, core workspace, tighter credits and limits.
Pro	Power users and builders	Higher credits, broader private automation and stronger workspace limits.
Enterprise Plus	Teams and developers	SDK access, publishing, advanced controls, higher limits and integration capability.
Enterprise Max	Large organizations	Maximum policy, administration, support, deployment and integration controls.

9. \$sAGENT Utility and Economic Policy

This section consolidates The AgentOS Economy dated 5 October 2026, especially sections 8-15 and 19-21, with the separate Wallet-holder programme. The economic framework describes the intended payment and buyback model. It does not establish that billing, product-token access, buybacks or holder distributions have been activated.

9.1 Three concrete roles for \$sAGENT

Role	How it works	What the participant receives
AgentOS subscriptions	A user pays the subscription price in \$sAGENT.	Access to AgentOS and the plan's Agent Credits, also called compute, for doing work.
Direct creator payments	A buyer pays a creator's stated access price in \$sAGENT.	The specified App, Skill, movie, picture or other genuine creation under its published access terms. The creator receives the direct-sale payment in \$sAGENT.
Revenue-linked buybacks	The published policy allocates 50% of total revenue generated in the ecosystem to buying back \$sAGENT.	A defined connection between ecosystem revenue and token acquisition, with allocation and executed purchases reported separately.

Example: a user pays for a subscription in \$sAGENT, uses its Agent Credits to create a picture, and sells access to that picture for a creator-set \$sAGENT price. The creator can use the proceeds for a future subscription or another creator's product. A product-token launch is not required for that direct sale.

9.2 Agent Credits and compute are the same allowance

Agent Credits and compute are two names for one platform working allowance. They are not two balances, a second tradable token or two separate purchases. In the published economy model, the subscription provides that allowance; \$sAGENT pays for the subscription.

A subscription does not automatically buy access to every separately priced creation. Creator earnings, product-access payments and Agent Credit accounting remain distinct. [Credit allowances](#), [cycles and resets](#) describe the operational allowance policy.

9.3 Product tokens and the benefits of holding them

A product token belongs to one verified genuine creation. It has its own unique AgentOS ticker and verified token address and network. It is distinct from \$sAGENT; a movie token does not replace the token used to pay for AgentOS subscriptions.

Creators choose and disclose the product's access model before launch:

Access model	What the user does	Benefit or result
Holder access	Holds the required balance of the verified product token. The qualifying tokens are not transferred as an access payment.	Receives the product access specified by the offer while meeting its stated eligibility rules.
Token-payment access	Approves and completes the specified payment in that product's token. Holding the balance alone is insufficient.	Receives the access purchased under the offer's duration, version and usage terms.
Direct pricing	Pays the creator's stated price in \$sAGENT, without requiring a product token.	Receives the access described in the direct-sale offer.

Movie example: a creator may let qualifying movie-token holders watch the movie while retaining their tokens, or require viewers to spend the movie token to purchase access. These are alternative access models. Neither tokenization nor a wallet balance silently chooses one for the user.

A holder-access offer must state the qualifying token, minimum balance, access granted, when eligibility is checked and what happens when the holder falls below the requirement. It does not automatically grant access to every product by that creator, lifetime access or all future versions.

9.4 Creator earnings and launch rules

Direct sales can cover Apps, Skills and creative artifacts such as movies and pictures. The creator receives \$sAGENT under the direct-sale offer. Earnings are not restricted to executable software.

Tokenized creations can earn creator rewards under the selected integrated external launchpad's rules. Qualifying activity, calculation, payout currency and settlement conditions must be disclosed for that provider. There is no universal creator-reward percentage. Product-access payments and launchpad creator rewards are separate receipts and must not be counted as the same payment.

Every token launch through AgentOS requires an attached verified, published genuine project or build. No build means no launch; memecoins and productless tokens are prohibited. AgentOS uses integrated external launchpads and does not claim an independent launchpad of its own. Launchpad discovery includes broader markets from those integrations and has no token-creation action.

Tickers must be unique across AgentOS, including across networks and launchpads. The reviewed build, creator, access terms, reserved ticker and actual resulting token must stay linked. An external launchpad cannot waive these rules. Single-wallet and no-bundling enforcement remains required.

9.5 \$sAGENT holder benefits and the Wallet programme

The published economy gives \$sAGENT a practical payment role: holders can use it for subscriptions and directly priced creations when those payment routes are activated. The 50% ecosystem-revenue buyback allocation is a separate economic mechanism; it is not a cash distribution to each holder.

A separate planned holder revenue distribution programme is described in the Wallet addendum:

- Only \$sAGENT held inside an AgentOS Wallet is intended to qualify. External, linked, exchange, treasury and developer payout wallet holdings do not qualify directly.
- Eligibility is intended to use time-weighted holdings over defined revenue periods. Moving tokens in for a closing snapshot does not create full-period eligibility; moving them out stops future accrual from the applicable measurement point.
- The qualifying holding asset is \$sAGENT. A supported stable settlement asset such as USDC is the recommended distribution asset; it is not an active payout promise.
- No holder distribution rate is active. Holder measurement, entitlements, programme claims and distributions remain unavailable until the financial ledger, versioned policy, reconciliation, security and required financial, tax, legal and jurisdictional gates are complete.

Each user explicitly creates one AgentOS Wallet. Its supported Solana and EVM addresses belong to that same wallet. Creating a wallet does not activate the holder programme. Creator-fee claims are a separate provider-specific earning route, with reviewed destinations, local signing and verified chain confirmation.

Holding \$sAGENT does not automatically confer company shares, shareholder voting rights, dividends, acquisition proceeds or guaranteed returns. Product-token holder access and the future \$sAGENT holder programme are distinct benefits with different eligibility rules.

9.6 What the 50% buyback policy means

The allocation is **50% of total revenue generated in the ecosystem**, not 50% of profit or a substituted measure of platform fees. For every 100 units of recorded ecosystem revenue, 50 units are allocated to \$sAGENT purchases. This is an illustration of the allocation, not a report of actual revenue or completed buybacks.

Revenue reporting must avoid counting the same receipt twice. Token trading volume and market capitalization are not automatically ecosystem revenue. An allocated amount, an executed purchase and the later use of purchased tokens are separate records. The policy does not by itself mean tokens are burned, distributed to holders or acquired immediately after every payment, and it does not guarantee a market price.

9.7 Published framework and current availability

The subscription, direct-payment, product-token access and 50% buyback mechanics above describe the published Economy framework. Current implementation documentation records no active AgentOS platform, activity or subscription charges; paid checkout and credit purchases are not active. No executed buyback is asserted here. Each payment or access path requires its own verified activation and disclosed offer terms.

AgentOS takes no cut from provider creator fees under the current creator policy. External launchpad and network charges remain separate and must be disclosed. An implemented Wallet or claim adapter, an unsigned simulation or a saved payout destination is not evidence of a settled sale, completed payout or active holder distribution.

The earlier whitepaper references to optional governance, community incentives and future FFP coordination remain longer-term possibilities. They do not replace the specific subscription, direct-payment, access and buyback mechanics in the Economy framework.

9.8 Official identity and source documents

The official \$sAGENT contract on Solana is [2Fob54QUhubP9jv6h5XAh3PgB1kcULR6LXbxSzuwpump](#).

Read [The AgentOS Economy PDF](#), the [original Economy Word document](#), [official provenance](#), [release status](#) and the [complete documentation and resources](#).

10. FFP: Future Verification Fabric

10.1 Intended role

FFP, the Furge Fabric Protocol, is the future trust and coordination direction for AgentOS. AgentOS is the human-facing operating environment; FFP may eventually support independent validation, distributed verification, routing, settlement and proofs for selected tasks.

10.2 Current boundary

FFP is not described as live. AgentOS does not currently claim active validators, independent consensus, proof events, on-chain finality, voting or protocol settlement unless those systems are connected and tested. The current product may expose FFP as Coming Soon and prepare stable, hashable consensus records for future integration.

10.3 Migration path

- 1 Stabilize Standard Consensus in software with durable, versioned records.
- 1 Define canonical task, context, criteria, participant and decision hashes.
- 1 Implement independent validator identity and cryptographic signing.
- 1 Test adversarial behavior, availability, privacy and dispute handling.
- 1 Introduce limited FFP-backed verification for low-risk tasks.
- 1 Expand only after performance, governance, security and economic viability are demonstrated.

11. Proprietary Model Research Programme

11.1 Strategic intent

AgentOS plans to evaluate whether a proprietary model family can improve sovereignty, cost, latency, privacy and execution quality. This direction remains undecided. No final model name, architecture, parameter count, training data mix, compute budget or launch date has been approved.

A proprietary model would not replace the AgentOS operating architecture. It would enter as another intelligence implementation behind the same context, permission, execution and evaluation boundaries. AgentOS must remain model-independent even if an internal model is eventually adopted.

11.2 Decision gates

Gate	Required work	Pass condition
1). Workload definition	Create an AgentOS benchmark covering conversation, planning, capability selection, code, document work, verification and recovery.	Clear measurable advantage targets and representative data.
2). Data governance	Identify lawful, licensed, consented and privacy-preserving training and evaluation data.	Documented provenance, retention, deletion and exclusion controls.
3). Baseline experiments	Compare open-weight adaptation, retrieval, distillation and training approaches.	Evidence that an internal approach can outperform or materially reduce cost on target workloads.
4). Safety and execution evaluation	Test hallucination, tool misuse, prompt injection, privacy leakage and calibration.	Meets AgentOS safety thresholds and cannot bypass the operating layer.
5). Infrastructure case	Estimate training, inference, observability, staffing and maintenance requirements.	Sustainable capital and operations plan.
6). Production pilot	Deploy behind an opt-in, reversible intelligence connection.	Measurable quality, latency, reliability and cost benefit.
7). Adoption decision	Compare pilot evidence against connected alternatives.	Proceed, narrow scope, partner or stop based on evidence.

11.3 Possible staged approach

- Begin with evaluation datasets and domain-specific adapters rather than immediate frontier-scale pretraining.
- Use retrieval and structured AgentOS context to reduce the amount of knowledge that must be embedded in model weights.
- Prioritize planning, capability selection, verification and execution-grounded response quality.
- Keep external connections available so users are not locked into one internal model.
- Publish model cards, limitations and evaluation results before broad production use.

No premature branding The research programme may use internal codenames, but the public whitepaper does not present an undecided model name as a finished product.

12. Roadmap and Delivery Gates

12.1 90-day execution window

The first public production chapter is organized around the execution window 21 July to 18 October 2026. The destination is a production AgentOS v1, an installable cross-device web experience and Windows release candidate, two active Nigerian university collaborations, Mini Hackathon Series 1, and credible first-CEX readiness subject to exchange approval. Native Android, iOS and macOS packages remain gated distribution work rather than claims of this release. Product readiness is the primary dependency.

Phase	Timing	Primary objective	Exit gate
Foundation and hardening	Days 1-30	Freeze v1 scope, close critical defects, secure data, prove core execution, prepare alpha clients and partner packs.	No unresolved P0 issue; alpha clients authenticate, execute and resume; outreach materials approved.
Beta and ecosystem activation	Days 31-60	Controlled beta, store submissions, university scheduling, hackathon registration and CEX applications.	Telemetry, support, legal, security and store requirements complete; beta reliability thresholds met.

Phase	Timing	Primary objective	Exit gate
Public launch and expansion	Days 61-90	Staged AgentOS v1 launch, cross-platform rollout, workshops, hackathon demo day, incubation and exchange target confirmation.	Rollback tested; support staffed; public flows verified; next-quarter plan approved.

12.2 Day-90 committed outcomes

Outcome	Definition of done
AgentOS v1	Public production release with Super AgentOS, Studio, Media Studio, Artifacts, Projects, Library, Vault, Prime Agents, Primeflows, stores and execution controls.
Cross-platform distribution	Installable web delivery on Android and iOS plus a Windows Store candidate; native Android, iOS and macOS packages require separate gates.
University programme	Two active institutional collaborations, four campus ambassadors and at least two technical workshops.
Mini Hackathon Series 1	Qualified submissions, a demo day and an incubation path for selected projects.
\$sAGENT exchange readiness	Legal, technical, token, liquidity, market-making and data-room readiness; first credible CEX target secured subject to approval.
Launch operations	Monitoring, analytics, customer support, incident response, release notes and post-launch reporting operate together.

12.3 Strict v1 scope

- Super AgentOS conversation, planning and task execution.
- NL Studio, Primeflow Builder, Code Studio and direct Media Studio.
- Artifacts distribution for images, video, audio and GIFs with creator-controlled permissions and pricing boundaries.
- Projects, sessions, Memory and search.
- Prime Agents and manual or scheduled Primeflows.
- Appstore, Skill Store and Library installation flows.
- Vault, permissions, redacted logs and Universal MCP connections.
- Execution logs, retries, cancellation, Recovery Center and visible Panic control.
- Free and Pro gating, basic usage metering and account management.
- Responsive, installable web on mobile plus the Windows desktop client; native Android, iOS and macOS packages remain separately gated distribution work.

12.4 Outside the critical path

- Full decentralized FFP deployment.
- AgentOS 5D Universe implementation.
- Complex token governance or dividend-style positioning.
- Large enterprise expansion beyond launch-critical foundations.
- High-volume integrations that cannot be properly tested within the window.
- A proprietary model release before the research gates are passed.

12.5 Medium-term roadmap

Horizon	Priority
0-3 months	Production v1, reliability, clients, institutions, Hackathon Series 1 and exchange readiness.
3-6 months	Marketplace monetization, developer analytics, stronger enterprise controls, Standard Consensus pilot and additional ecosystem integrations.
6-12 months	Regional adoption, richer SDK, verified cross-device continuity, model research pilots and FFP test infrastructure.
12-24 months	Selective FFP-backed verification, broader enterprise deployment, international builder programmes and proprietary-model decision.
24+ months	Spatial and device interfaces, the AgentOS 5D Universe, protocol-scale coordination and deeper autonomous operations where safe.

13. Adoption Strategy

13.1 Nigeria and West Africa

AgentOS can use Nigeria as a high-density builder and real-problem market while maintaining international product standards. The strategy should focus on useful work for education, small businesses, public-service accessibility, agriculture, health administration and developer productivity rather than generic promotional adoption.

13.2 University programme

- Secure accountable institutional coordinators and formal programme dates.
- Train campus ambassadors and faculty or innovation-centre contacts.
- Provide workshops, controlled credits, mentorship and publishing pathways.
- Convert validated student work into Apps, Skills, PrimeAgent operators and PrimeFlow executions.
- Create case studies, internships, contribution opportunities and incubation tracks.

13.3 Mini Hackathon Series 1

The recommended theme is "Build for Nigeria with AgentOS": practical products for education, SMEs, public services, agriculture, health administration and developer productivity. Entries should produce a working App, Skill, PrimeAgent or PrimeFlow, a demo, a problem statement and deployment instructions. Judging should prioritize technical execution, problem value, usability and ecosystem reusability.

13.4 International positioning

Internationally, AgentOS should be positioned as the operating layer between capable intelligence systems and real execution. The narrative should focus on architecture, safety, distribution and durable context-not unsupported superiority claims. Intel ecosystem engagement, developer programmes, university collaboration and strategic integrations can reinforce the infrastructure thesis.

14. Governance, Metrics and Operations

14.1 Governance separation

Domain	Authority	Principle
Product governance	AgentOS company and designated product owners.	Security, reliability and user value take priority over token price or community pressure.
Workspace governance	Users and enterprise administrators.	Permissions, data, policies and connected resources remain scoped to the workspace.
Marketplace governance	AgentOS review and policy processes.	Listings require factual metadata, working integration and enforceable permissions.
Token policy	Treasury and legally authorized governance mechanisms.	Transparent disclosures; no implied shareholder rights.
Future FFP governance	Defined only when protocol roles and legal structure exist.	No fictitious decentralization before independent participation is real.

14.2 North-star metrics

Metric	Why it matters
Successful task completion	Measures whether AgentOS converts intent into verified outcomes.
Time to first successful task	Measures onboarding clarity and product usability.
Repeat meaningful usage	Distinguishes durable value from one-time curiosity.
Recovery success rate	Measures resilience after failures.
Approval-to-execution accuracy	Detects duplicate, unauthorized or mismatched actions.
Installed capability activation	Shows whether Apps and Skills become useful, not merely downloaded.
Builder retention and earnings	Measures ecosystem health.
Security and privacy incidents	Must trend toward zero; every serious incident has a postmortem.
Cost per successful task	Supports sustainable plan and credit design.
Cross-device continuity success	Measures whether the same workspace truly follows the user.

14.3 Release discipline

- Every phase begins with a baseline audit and ends with scope, architecture, security and UX audits.
- No feature is called production-ready because it compiles or has a mock test.
- Real browser and end-to-end flows are mandatory for launch-critical paths.
- Migrations require dry runs, rollback procedures and data-preservation checks.
- Local code, repository state, deployment commit and live product must be synchronized.
- Visible actions must work, be honestly disabled or clearly explain their state.

15. Risks and Mitigations

Risk	Impact	Mitigation
Product scope expansion	Delays v1 and weakens reliability.	Freeze critical-path scope; require formal change control and gate reviews.
Overdependence on external intelligence	Cost, availability and identity risk.	Native remains functional; user-owned connections; replaceable adapters; proprietary-model research.
Secret or data leakage	Loss of trust and legal exposure.	Vault grants, server authority, redaction, tenant tests, audit and incident response.
Model hallucination or tool invention	Incorrect plans and unsafe actions.	Capability Graph, schema validation, execution separation and real-result verification.
Duplicate or runaway execution	Financial, data or reputational harm.	Idempotency, approvals, cancellation, Panic control and recovery.
Marketplace fraud or low quality	User harm and ecosystem degradation.	Verification, permissions review, factual telemetry, staged publishing and enforcement.
Token-first perception	Weak product credibility and regulatory risk.	Product-first narrative, utility boundaries, transparent treasury policy and no guaranteed-return language.
Store or platform rejection	Distribution delay.	Submit early, maintain PWA fallback and resolve policy issues through staged releases.
University bureaucracy	Partnership slippage.	Multiple targets, named coordinators, dated commitments and clear institutional responsibilities.
FFP overclaim	Technical credibility loss.	Coming Soon status, stable research milestones and evidence before protocol claims.
Model-training cost and failure	Capital loss and distraction.	Gated research programme, baseline experiments and stop decisions.
CEX readiness failure	Reputational and liquidity pressure.	Separate data room, legal review, disclosure, liquidity planning and approval-dependent timelines.

16. Legal and Regulatory Position

This whitepaper is a strategic and technical description, not investment advice, a securities offering, a promise of profit, a guaranteed roadmap or a binding offer. Product features, token utilities, fees, timelines and protocol plans may change after technical, security, legal, financial and market review.

\$sAGENT does not represent company equity, a debt claim, a guaranteed revenue share or an automatic right to dividends or acquisition proceeds. Any future benefit with those characteristics would require separate legal documentation and compliance in each relevant jurisdiction.

AgentOS must implement privacy, consumer protection, intellectual-property, tax, employment, financial, sanctions and digital-asset obligations applicable to its activities. Exchange listings, token programmes, marketplace payouts, institutional partnerships and FFP operations remain subject to third-party approval and law.

Users are responsible for the risks of connecting external services, granting permissions, installing third-party Apps or Skills and transacting with digital assets. AgentOS should provide clear permission disclosures, revocation, audit and risk notices, but no system can eliminate all operational or market risk.

17. Conclusion

The defining AgentOS opportunity is not to win a model benchmark or reproduce another chat product. It is to build the operating ecosystem that makes intelligence useful, controllable and distributable. Super AgentOS provides one persistent command layer; Projects, Memory and Library preserve context; Apps and Skills distribute capabilities; PrimeAgent operators and PrimeFlow executions make work reusable; Vault secures access; Universal MCP connects outside systems; and the Execution Engine turns plans into governed results.

The near-term standard is production discipline: real flows, clear permissions, reliable execution, cross-device continuity and honest product claims. The medium-term standard is ecosystem depth: builders, institutions, marketplace economics and verifiable consensus. The long-term vision is an AgentOS that can operate across models, devices, organizations and spatial environments while keeping the user's context, control and identity intact.

Final statement AgentOS is not another wrapper. It is the operating system for autonomous intelligence: one command, a governed workspace, and execution that can be verified.

Appendix A - Current, Committed and Exploratory Status Matrix

Area	Status	Whitepaper treatment
Super AgentOS operating layer	Current foundation plus production hardening	Central product identity; not described as a foundation model.
Studio, Projects, Library and Vault	Current foundation	Core workspace and security surfaces.
Apps, Skills and developer publishing	Current foundation with monetization expansion	Distribution layer; no fabricated listings or earnings.
Connected Intelligence selection	Committed direction	User-owned connections, exact model selection and Vault-backed credentials.
Standard Consensus	Committed after Single Intelligence stability	Software consensus; preserves dissent; no FFP claim.
FFP	Exploratory and future protocol	Coming Soon until independent protocol execution is proven.
Proprietary AgentOS model	Exploratory research	No final decision, name, scale or launch date.
Installable mobile web and Windows desktop client	Current foundation; Store submission prep	Shared backend authority and cross-device continuity.
University programme and Hackathon Series 1	Committed 90-day programme	Builder activation tied to working products.
CEX listing	Readiness target subject to approval	No guaranteed listing date or outcome.
AgentOS 5D Universe	Long-term research vision	Spatial, contextual and temporal workspace after 2D production maturity.

Appendix B - Glossary

Term	Definition
AgentOS	The operating ecosystem that coordinates context, capabilities, permissions and execution.
Super AgentOS	The permanent command, operating and delivery layer of AgentOS.
Native	Always-available deterministic AgentOS functionality without a Connected Intelligence requirement.
Connected Intelligence	A user-selected model, agent or private endpoint that extends reasoning or generation.
Single Intelligence	One selected connection used for a request or session.
Standard Consensus	Software-based independent worker and verifier evaluation.
Prime Agent	A user-created operator with scoped instructions, context and capabilities.
Primeflow	A reusable execution graph combining prompts, capabilities, triggers, permissions and outputs.
App	A larger SDK-integrated product surface distributed through Appstore.
Skill	A modular installable capability distributed through Skill Store.
Library	The durable inventory of installed, created and generated workspace assets.
Vault	The secure secret and runtime-grant layer.
Universal MCP	The external tool and agent connection layer.
Agent Credits	Platform usage accounting for compute and execution.
\$sAGENT	The intended ecosystem utility token, separate from equity and credits.
FFP	The future Furge Fabric Protocol trust and coordination layer.

Appendix C - Whitepaper Audit Record

The document was audited against the latest AgentOS product directives, master specification and 90-day execution roadmap available at publication. The following consistency checks were applied:

- Super AgentOS is consistently defined as the operating layer, not a model or third-party wrapper.
- Native remains functional with zero Connected Intelligence connections.
- Connected Intelligence cannot own secrets, permissions, execution or product identity.
- Prime Agent and Primeflow terminology is used consistently.
- Apps and Skills are the monetizable marketplace surfaces; Agent Credits remain separate.
- \$sAGENT is separated from equity, guaranteed returns, Agent Credits and automatic holder rights.
- FFP is marked future and not falsely described as live.
- The proprietary-model plan is included as undecided, gated research.
- The 90-day roadmap contains v1, clients, universities, Hackathon Series 1 and CEX readiness with go/no-go gates.
- The 5D Universe is retained as long-term vision outside the critical path.
- Security, privacy, approvals, cancellation, Panic, recovery, observability and no-mock discipline are explicit.
- Legal language avoids promising listings, dividends, price support, timelines or unimplemented capabilities.

Source basis: AgentOS repository documentation (v6.8.0), AgentOS Master Specification Volumes 00-42, Super AgentOS Intelligence Runtime production directive, and AgentOS 90-Day Execution Roadmap, July-October 2026.

Prepared for AgentOS. Public strategic and technical whitepaper, version 1.1, September 2026.

Appendix D - V6.6.9 Primeflows Release Addendum

Primeflows is a first-class discovery and execution surface. Public discovery exposes immutable sanitized versions with real Like, Add, Fork and freshness signals. Like is social state; Add records an asset relationship in Library; Fork creates a new private editable Primeflow from the exact published version and records lineage.

Library remains the durable system of record for owned, published, added and forked workspace assets. Tasks remains the durable system of record for execution. Every manual or scheduled Primeflow run executes its complete ordered plan and records correlated execution logs and ordered Task steps.

Primeflow publication cannot include credentials, tokens, cookies, passwords, private keys, grants, connections, workspace or Project bindings, sessions, Memory identifiers, file identifiers or other private context. Safe credential placeholders remain empty and must be configured by the receiving user.

Appendix E - V6.6.10 AgentOS Wallet and Token Flywheel Addendum

Status

AgentOS Wallet is now visible as a first-class navigation and roadmap surface. The wallet runtime remains Coming Soon. V6.6.10 does not create wallet addresses, keys, balances, token holdings, entitlements, claims, transfers, distributions, or transaction history.

When the wallet runtime is activated, every AgentOS account is intended to receive an AgentOS Wallet. Only \$sAGENT held inside AgentOS Wallet is intended to qualify for the planned holder revenue distribution programme. Holdings in external, linked, exchange, treasury, or developer payout wallets will not qualify directly.

Product-backed flywheel

The planned flywheel begins with real use cases and Agent Credit purchases. Users buy Agent Credits to power platform compute and execution, creating AgentOS platform revenue. Paid Apps and Skills separately create marketplace revenue; eligible net App and Skill revenue is intended to split 70% to builders and 30% to AgentOS. Subject to separate approval, eligible AgentOS revenue may support \$sAGENT buybacks, builder incentives, treasury programmes, and qualifying AgentOS Wallet-holder distributions without silently reducing or reclassifying the builder share.

Primeflows and Prime Agents remain reusable execution assets rather than direct monetization products. When either uses a paid App or Skill, the paid capability remains the attributable monetization source. Agent Credits remain platform compute accounting, their purchase is a direct AgentOS revenue input to the flywheel, and they remain separate from \$sAGENT, marketplace balances, developer earnings, and holder distributions.

Concrete use case

A user gives Super AgentOS one command to find and execute a revenue opportunity across connected accounts. The user buys Agent Credits to power the required platform compute and execution. Super AgentOS can then coordinate up to five user-connected External Intelligences, use only explicitly approved app credentials, and apply Apps, Skills, and Primeflows to research, test, debug, and act across the selected accounts.

The Agent Credit purchase creates AgentOS platform revenue. If the work also produces a \$10 eligible net App or Skill charge, \$7 is allocated to the builder and \$3 to AgentOS. Subject to separate approval, eligible AgentOS revenue may support \$sAGENT buybacks, builder incentives, treasury programmes, and qualifying AgentOS Wallet holders after the programme is active. Stronger AgentOS and builder economics improve the capabilities that attract more use cases and Credit purchases. Any user proceeds can enter AgentOS Wallet only after the Wallet and relevant payment runtime are live.

Eligibility and activation boundary

Future eligibility should use time-weighted \$sAGENT holdings across defined revenue periods rather than a single closing snapshot. A programme rate must be versioned, effective-dated, auditable, and applied only after refunds, chargebacks, taxes, payment costs, reconciliation, treasury controls, jurisdictional eligibility, security review, and legal approval. Until those gates pass, AgentOS must describe the Wallet and holder programme as planned, display honest inactive states, and avoid representing token holders as current shareholders or recipients of guaranteed revenue.

Appendix F - V6.6.12 App Session Reliability Addendum

AgentOS Apps use owner- and workspace-scoped shortcuts and an auto-hidden edge dock. Every opening and return rechecks authentication, installation, workspace membership and readiness before launching. The dock uses bundled official logos, visible focus and labels, touch targets, and automatic hiding that pauses while keyboard focus remains inside it.

While the original AgentOS document remains open, the controller returns to the same live external window without resetting its route, scroll or unsaved draft when the provider preserves a controllable handle. Switching marks the previous App minimized. If provider browser isolation severs that handle, AgentOS opens no duplicate and directs the user to the existing browser tab or to end the shortcut before reopening. Reloading AgentOS loses live window references; selecting a saved shortcut then opens the provider's validated entry point. The provider controls recovery of its content and any private state.

After 24 hours without AgentOS activation, a shortcut is marked idle. This is an activity label, not runtime hibernation: AgentOS does not suspend cross-origin pages, stop their timers, or transfer their work into Tasks. Independently scheduled Tasks retain their existing execution model. Ending a session, uninstalling an App or signing out removes AgentOS tracking; external browser tabs remain under the user's control.

Saved records contain a non-secret owner scope, App identity, workspace, bundled logo path, state and timestamps. Launch URLs remain transient; credentials, query tokens, cookies, card data, model context and third-party contents are excluded from storage. Legacy records containing launch targets are discarded. Owner changes and cross-tab sign-out invalidate pending launches, and blocked pop-ups produce a visible error without a phantom session.

The controller tracks up to eight live windows and forty saved shortcuts. These limits apply to the current AgentOS document and do not measure or control provider CPU or memory. Provider navigation, session expiry and browser isolation policies may require a fresh opening.

Curated external Apps retain durable catalog identities for detail review, installation and governed launch. Existing Wallet, Primeflows, capability, execution, audit and recovery contracts remain in place.

Appendix G - V6.6.13 Release Integrity

AgentOS integrates Eyebrow 0.4.3 with its release and execution controls. The scanner binary is downloaded from the upstream release and checked against a pinned archive checksum. Stored skill source is inspected in a temporary directory without executing it or receiving Vault credentials. Medium, high and critical findings require corrected source; low and informational findings remain visible advisories. Static inspection does not guarantee code safety.

Published app manifests, stored skill source and declared access, and registered MCP destinations and tool declarations receive immutable release snapshots. Changed contents require a new version. Skill installations record the approved digest. Every skill run checks the release, revocation and approved digest before source inspection and Vault access; unavailable checks pause execution. Existing installations require a fresh review. App updates require review before opening the changed version.

The public Appstore monitor compares the complete catalog with a reviewed Eyebrow baseline every six hours. It distinguishes metadata drift from unavailable inspection. TLS key changes and failed TLS checks are reported separately because they do not establish a change in application code. Remote websites and hosted MCP implementation are not obtained or sandboxed by this integration. Publisher signatures and local MCP process wrapping remain outside this release.

Skill execution forms collect required inputs and display the returned result. App and Skill details expose the coverage, recorded version, evidence time, digest and review state. A pinned listing means its published declaration is recorded; it is not a certificate that remote code is safe. The first snapshot records the contents present at enrollment and does not retroactively certify earlier history.

Primeflow catalog correction

Public Primeflow pages resolve their exact published snapshot independently of the active workspace. The original twelve official catalog entries were withdrawn after their advertised outcomes proved unverified. Three manual report flows replace them: Lagos Weather Snapshot, Bitcoin Price Snapshot, and Workspace File Inventory. Each captures source data, writes a JSON file to AgentOS storage, and reads the saved file back. These flows do not claim research generation, wallet monitoring, backup validation, release approval or notification delivery. Historical versions and user-owned private forks are preserved.

Publication checks reject unsupported tools, schedules and output references. Structured file content is serialized and missing content fails explicitly. Discovery no longer republishes legacy rows on read. Absence of credential placeholders is not evidence of an outcome or runtime guarantee.

The console uses distinct section accents across navigation, shared page headers, cards and actions. Dark and light themes, collapsed navigation and mobile drawers retain the same section identity. Active-page markers and visible keyboard focus accompany color.

Appendix H - V6.8 App Experience

AgentOS presents one continuous operating environment across laptop and mobile. Successful web sign-ins use a 30-day browser access session backed by a 90-day refresh window, while concurrent session checks share refresh work. Its built-in apps open in movable, resizable laptop windows with a persistent taskbar. On phones, the same app identities use full-screen views with Home, Back and Recents. Original AgentOS logos and section colors remain the identity of the system.

A system host owns app activation, focus, window bounds and workspace identity. Trusted same-origin app documents retain their existing routers, providers, APIs and permission checks. Focus changes visual stacking without moving or reloading the app document. Library, Super AgentOS, SpaceGo, Tasks, Projects, Appstore, Skills, Prime Agents, Primeflows, Vault, Universal MCP, Search and Settings share the system. Account, documentation and compatibility routes have explicit owners.

SpaceGo uses the Go logo in its launcher, app window, taskbar and standalone Windows package. The updated AgentOS desktop client embeds its native browser inside the normal app window. Minimize, restore, move, resize and app switching preserve the same native session; closing the app or changing account destroys it. Playground and system dialogs hide the native view. Browser sessions remain memory-only, while selected settings and script drafts use scoped session recovery. SpaceGo provides a real Windows HTTPS browser with Back, Forward, Reload, Stop, and memory-only Spaces cleared on close. Permissions and downloads are denied; guest pages receive no AgentOS preload, Vault, or filesystem API. The web companion opens external tabs in the existing browser profile and cannot inspect them. Playground exports Playwright scripts and runs policy-checked inspect, screenshot, click, type, select, scroll and confirmed-submit actions against the live SpaceGo page. Super AgentOS can hand off HTTPS navigation, inspection and screenshot requests, including repeated requests to an already-open app. The desktop host enforces the current origin, control mode, capability expiry, source, input limits and submit confirmation. Page content cannot authorize itself. Computer Use remains browser-scoped: file transfer, Vault secrets, arbitrary Windows control, long autonomous plans and unattended remote workers are not enabled.

Recovery metadata is separate from private drafts. The host stores app and workspace identifiers, allowlisted route parameters and window bounds. Supported Studio drafts, editor tabs, Library views, catalog filters and account-form drafts use account- and instance-scoped browser session storage. Vault values, credentials, tokens and arbitrary forms are excluded. Unsaved forms without recovery remain open and warn before closing. Recovery depends on browser storage being available; signing out or changing account invalidates private recovery state.

Closing a running Studio view retains its app document in the background until its response finishes. It does not issue a stop request. Explicit execution controls and Panic keep their existing authority. A browser termination or device shutdown can still interrupt a browser-bound response; this release does not convert every execution into a durable background worker.

Library opens selected files and can attach an authorized file to Studio. Installed apps use server validation of installation, membership, readiness and launch address. External provider state remains outside AgentOS recovery guarantees. Notifications resolve the exact workspace, session or execution rather than a generic screen.

Chronos retains its existing implementation, preferences, timezone, weather, battery and fullscreen behavior. Panic remains a system control with touch and viewport handling. The public landing page introduces AgentOS; authenticated Home is its operating surface. Device boot, kernel services, unattended remote browser workers and unrestricted hardware control are separate platform work and are not claimed by this release.

September 12 reliability boundaries

Browser access is validated against durable session records, and rotation and revocation share atomic account locks. Storage failure is an unavailable state, not proof of sign-out or an empty workspace. Search history uses account-scoped browser keys; server search limits private execution history to its owner and reports unavailable categories.

PRIME command execution uses explicit supported native commands. The profile does not imply a configured language model or unrestricted permissions. Invoking a shared PRIME retains the caller identity and limits memory to the invoked agent namespace. Sharing requires source ownership, canonical workspace membership, and an allowed permission.

Code Studio hosted commands require an isolated runtime connection. The sandbox has a bounded lifetime, denied outbound networking, bounded project file synchronization, and verified whole-sandbox cancellation. Each command starts a fresh shell. Retry starts a supported new attempt; pause, resume, and rollback require actual runtime support. Primeflow schedule updates are atomic, and durable runs retain their workspace and project.

Media Studio video export uses fixed frame timestamps and supported browser codecs for H.264/AAC MP4. Codec availability is checked explicitly, while GIF and image exports remain independent. Clip trim, output duration, and audio gain are verified against produced files. Developers and creators retain control of prices; unsupported paid distribution is not represented as completed checkout.

The September 12 release candidate restores chat recovery through Active, Archived, and Trash views. Restore checks the chat owner and clears archive and deletion markers; failed rename or project attachment retains input and reports the error. Primeflow Prompt steps execute supported native Studio commands with the caller permissions and stop when confirmation is required. Code Studio reconnects only to a matching saved project terminal; a temporary failure does not resubmit the command. Explicit Stop and Panic report successful cancellation only after the runtime provider confirms it. These corrections require verification on the exact production deployment.

Recorded browser audits and automated regressions provide evidence for tested paths. They do not certify unavailable providers, unsupported codecs, external billing, or unrestricted autonomous operation. Password-reset delivery requires an approved mail connection; durable token consumption alone is not evidence that an email was delivered.

AgentOS MODES and composer work intent

AgentOS MODES provides eight soft work contexts: General, Trading, Coding, Gaming, Chat, Bounty, Research, and Artifact. These do not replace NL Studio, Code Studio, or Primeflow navigation. A mode prioritizes relevant available resources while explicit requests, permissions, approvals, and provider availability retain authority.

Personal defaults are scoped to the account and workspace; saved conversations retain their own mode. An optional administrator recommendation is a fallback, not an enforced override. Mode context accompanies supported native and connected-intelligence execution through the shared hosted runtime.

Chronos adds an account-wide view of the signed-in user's own recorded activity. A transparent SVG prism represents messages, completed runs, and actual mode transitions, with an exact-value table and 7-day, 30-day, 90-day, and custom reporting periods. Analytics store event metadata, not prompt contents, and show empty or unavailable states honestly.

Separate Ask, Plan, and Goal controls live on the composer. Plan saves a review checklist without running work. Goal saves a tracked outcome without claiming autonomous background execution. The saved record appears above the composer. Without planning intelligence, a resource-based scope checklist is not represented as a model-generated task decomposition. Verification, execution, and completion remain distinct states.

Deployment requires the additive schema migration and live verification. The same hosted implementation is intended for web and connected app clients; package-specific update behavior must be verified on the actual client.

Account onboarding recovery

AgentOS uses staged signup with atomic account and workspace provisioning. Duplicate email or username details return one neutral account-conflict response. Users can edit either detail or move directly to sign-in and recovery without losing their chosen plan. Interrupted registration requests are not automatically repeated because the account may already have been created. Passwords are never persisted by the staged form. Password-reset delivery remains dependent on a configured email provider.

Studio draft editing

The shared Studio composer grows with the draft within the available viewport. Longer text remains editable through scrolling without a client-side character cap. Focus is indicated around the whole composer rather than a nested text box. Changing window or rail width reflows the text. Shift+Enter inserts a new line, and confirming IME composition does not send the draft. Modes, resource selection and permission boundaries are unchanged.

Focused work surfaces and personal settings

AgentOS keeps its desktop, launcher, dock and original logo while simplifying the applications inside them. Library and Projects separate primary work from contextual inspection. Tasks distinguishes active work, failures requiring attention, and history without changing actual execution states. Search supports keyboard result navigation. MCP registry availability remains distinct from an authenticated connection. Stores preserve listings, permissions and prices while simplifying discovery.

Settings adds personal AgentOS wallpaper presets and raster-image uploads, fit and dimming, text scaling, reduced motion and contrast. These preferences are saved through an authenticated, account-scoped API and reused by hosted web and app surfaces. The original desktop stays the default. Device Wi-Fi, audio and other operating-system controls remain under the device settings, not simulated inside AgentOS. Code Studio prioritizes the editor with a resizable terminal/results area; Primeflow Builder prioritizes the workflow canvas.

Current models and typed decisions

External Intelligence includes GPT-6 Astra, Sol and Luna and Claude Fable 5.1 and Opus 5.5 while retaining existing model selections. Vault refreshes models from the selected provider credential without choosing a replacement silently. Availability is account-specific; a catalogue entry is not evidence of a connected or successfully executed provider.

TypeSafe AI's Jev is an explicitly selected structured-evaluation capability, not a conversational generator. Users enter state and Choice, Score or Noul questions and confirm sending that material to TypeSafe. AgentOS does not automatically export chat history, files or workspace contents. Typed results, resolved model identity and actual token usage are returned. Vault credentials stay server-side. Task Center and Panic cancellation remain connected to the tracked evaluation; cancellation cannot undo work already processed by the provider. Failed or cancelled evaluations never report successful completion, and no additional provider or automatic retry is invoked. TypeSafe account billing remains separate; unavailable monetary cost is shown as unknown rather than fabricated.

Enterprise settings and governance

Enterprise Plus and Enterprise Max support organisation administration while retaining their existing account quotas. Organisation settings are separate from personal profile, wallpaper, appearance, accessibility, notifications and regional preferences. AgentOS does not introduce a shared credit wallet, device-management service or SCIM provisioning in this release.

An organisation can link multiple workspaces. Both the workspace owner and organisation owner approve the same policy impact before linking. Existing content is preserved; organisation membership alone does not grant access to private workspaces. Owner, admin, billing manager, auditor and member roles have different server-enforced

permissions. Expiring, single-use invitations, groups and administrative changes retain an audit record. The last owner cannot be removed.

The eight organisation settings sections cover workspaces, people, security, automation, intelligence, budgets, data and developer access. Current workspace context selects the affected resources. Unsaved policy changes survive a stale-version conflict instead of overwriting another administrator's changes.

Identity and execution boundaries

OIDC and SAML provider configuration must pass a real sign-in test before activation. Requiring organisation SSO also requires owner recovery. Verified MFA is separate from a provider merely returning a password or OTP label. Organisation assurance is tied to the current browser session or an explicitly provisioned, scoped service identity. Personal authentication remains available for unrelated personal resources.

Organisation policies bound available providers, models, apps, integrations, tools, credentials and external sharing. A workspace may tighten a policy, not weaken it. Explicit user requests and MODES do not bypass these controls. Consequential work can require a one-time approval bound to the exact action, input, identity and current policy. Task Center presents pending decisions; approving unrelated work never grants blanket authority. Emergency stop reports completed cancellations separately from pending or failed stops.

Spending, data and records

Organisation, workspace and member budgets measure AgentOS credits within UTC calendar months. Hard caps count outstanding reservations and settled charges. Finalisation releases unused reservations, not already charged work. Uncertain costs stay reserved until reconciled. Warnings are issued at configured thresholds. Externally billed provider invoices remain separate and are not controlled by these caps.

Retention starts with an explicitly confirmed 30-, 90- or 365-day schedule. Ninety days is preselected. Eligible expired organisation content enters a 30-day recoverable quarantine; preservation holds prevent expiry. Personal content, active work, published releases, credentials, financial records and governance audit records are excluded. Previously linked workspace content is not retrospectively expired without an owner-confirmed affected-data preview. Automatic processing is separately gated and Settings reports when it is disabled.

Content exports remain scoped to the requesting account's accessible records. File exports provide a manifest with authenticated download routes, not a claim that a JSON file contains every stored byte. Administrative audit exports require the appropriate organisation role. Service keys are scoped, expiring and revocable; cleartext appears only once after durable issuance. Secrets and prompt content are excluded from governance audit records.

Source implementation and isolated tests are not proof of a production rollout. Release evidence must separately confirm the deployed commit, actual identity-provider journeys, enforcement, storage lifecycle and browser or hosted-app behaviour before these capabilities are announced as generally available.

Chronos at a glance

A compact transparent mode prism remains visible in the Chronos control before expansion. Its eight nodes reflect the account's real 30-day message and completed-run activity; connections represent recorded mode switches. An empty history shows neutral nodes, not sample usage. The expanded Modes view keeps period filters and an exact-value table. Time, weather and device battery remain alongside the prism, and detailed analytics stay inside Chronos.

Gaming Studio

Gaming Studio brings Play, Create and Releases into the AgentOS workspace. Prism Run and Lumen Rooms are original 2D and 3D starter projects. Users can play, keep personal progress, edit Project files, build in an isolated runtime, playtest and review immutable versions before publishing. Code Studio and Media Studio open the same Project files. A successful build remains distinct from a playtest.

Trading Studio

AgentOS Paper provides virtual USD and USDT accounts without broker credentials. Reviewed paper buys can carry multiple take-profit and stop-loss levels. Active plans use fresh reference bids on scheduled checks; a target touch or stop level does not guarantee execution or a particular price. Filled legs, cancellations and remaining quantities are shown with the order record. Discretionary sales cancel remaining virtual protection, while Panic pauses checks. Strategy backtests do not create orders. Funded broker execution remains separately gated behind eligible provider accounts and verification.

Paper perpetual execution and risk controls

Trading Studio supports isolated virtual perpetual positions in USDT, using explicitly selected Hyperliquid or Aster market references. Mainnet and testnet prices remain distinct; a mainnet price does not authorize a funded order. Review binds account, symbol, direction, quantity, leverage and venue. Confirmation replay returns its existing receipt without duplicating execution.

New risk-managed entries may attach five take-profit and three stop-loss levels. Partial reductions, risk events and balances persist in the shared runtime. Liquidation uses disclosed paper assumptions: one percent maintenance margin, five basis points adverse slippage and five basis points assumed venue fees. It is not a provider's liquidation engine; funding payments are not simulated. Older positions remain manually monitored. The five-minute scheduler and explicit reconciliation use fresh marks, current access and Panic state. Checks can be delayed and miss price touches. No AgentOS fee or real exchange transaction is generated.

Paper trading uses virtual funds. It does not enable live orders, Wallet transfers or token launches. Those features have their own availability and approval requirements.

Account-owned Wallet implementation

AgentOS Wallet provides explicit browser-side generation, encrypted account-bound backup and password-checked recovery-phrase/private-key export. Private keys and recovery phrases are not sent to the server. Reviewed native transfers bind the chain, sender, recipient, amount and maximum network fee before local signing. Durable intents distinguish submission from confirmed execution; canonical finalized receipts are required before a terminal success is recorded. Funded transfers are still being validated; network configuration alone does not make them available.

Balances load on entry and network changes. Reads verify network identity, prefer dedicated connections and use labelled public connections only when none is configured. Solana balances use finalized commitment; EVM balances reflect the current chain head. Provider failures never become estimated or zero balances. Public read access does not enable unconfigured transaction signing or submission.

Creator claims, swaps, build-token payments and holder distributions remain unavailable. Available Wallet features are shown in the app for the selected network and account. Agent Credits and sAGENT remain separate.

Desktop layout preferences

Home apps rearrange by dragging an icon or label directly; no edit mode or menu is needed. The app follows the cursor, gaps between tiles accept drops, and dropping outside the grid cancels. A drag does not launch the app. Alt+Arrow remains a keyboard alternative. Order is stored on the current device per account and workspace, without changing permissions or synchronizing positions between devices. Desktop Panic supports drag positioning, viewport bounds, keyboard movement and Ctrl+P access. Moving it does not execute a safety action. Desktop windows use the available work area. Exact legacy automatic sizes are upgraded; custom sizes are preserved. Retired preview preferences no longer restore the previous console. Shared application headers, cards and nested spacing remain compact across hosted clients. Phone Panic keeps its navigation-aware placement and 44px touch target. Release evidence identifies the deployed commit and verified clients.

Wallet security update

Each user has one AgentOS Wallet with Solana and EVM addresses. Wallet creation, encrypted backup and export preserve existing keys and transaction records. Network selection does not create another wallet.

The update validates decoded swap instructions before review and signing, encrypts the approved signed bytes for bounded recovery, and reconciles final status against chain evidence. Recovery cannot sign a replacement transaction. Collectible ownership is checked independently of indexer metadata; transaction history uses stable account-scoped pagination.

Wallet verifies the chain identity of public and configured connections. Base mainnet uses the verified PublicNode default; configured providers retain precedence without silent fallback. A connection does not authorize signing or establish transaction completion.

Creator-fee claims follow each provider's eligibility, destination and signing rules. Pump.fun/PumpSwap, pons, Bankr, StonkFun/Stonks and Uniswap have protocol-specific implementations with separate verification gates. StonkPad creator claims remain unavailable. This update does not release Launchpad. Production availability requires verified deployment and execution evidence.

Wallet indexing verification

Configured Alchemy connections provide token and transfer indexing for HyperEVM and Robinhood. Chain identity, owner scope, exact token quantities and independent incoming/outgoing pagination are validated. Missing timestamps are resolved from actual blocks; unsupported precision and incomplete coverage remain visible. Live read-only reference evidence is separate from signed execution. Connection settings are saved in production, but the indexing update is pending release verification and deployment.

The pons v2 claim path binds native and quote-token withdrawals to the authenticated beneficiary and published fee escrow. Only settled balances are included; unswept fees and unreleased buyback vesting are not presented as claimable. A positive public-reference unsigned simulation has passed, without signing or broadcast.

October 2026 studio upgrade: saved work and selectable brains

This upgrade refines the existing AgentOS operating system and preserves its six studios, eight MODES, tools, original logo and execution controls. The compact composer retains context, attachments, sending and cancellation. Separate Plan and Goal controls hang above the typing box, begin closed, preview on hover and pin on click. Touch, keyboard, explicit closing and Escape are supported.

Task Center stores distinct named plans and desired-result goals. Saving records does not start work. Multiple plans keep separate steps and progress; a plan may belong to a goal or stand alone. Execution binds to the exact owned plan in its conversation, workspace and project. Only completed steps with recorded AgentOS verification receive ticks. Failed, blocked and unverified steps remain unticked. A completed linked plan does not automatically verify its goal.

Action review is a compact strip with full expandable details. Repeated clicks cannot submit twice. The strip clears only after an explicit server acknowledgement; failures retain recovery controls. Work details retain estimates, actual usage, evidence, approvals, errors, funding and task controls. The separate Summary keeps Mission, Objective, Decisions, Next steps, Work & files, execution records, source links and context access. On iPad, the dock reserves Panic space; saved positions move clear of visible apps and the dock in both orientations.

Single uses one selected model. Compare displays separate answers from two to four models. Team uses two to four models with an explicitly selected lead, one combined answer and expandable contributions. Missing participants and disagreements are stated; an unavailable lead is never silently replaced. These choices preserve legacy parallel research, Standard Consensus, one-message overrides and typed Jev decisions.

Included access uses protected server credentials. Models appear only after validation for the configured connection. OpenRouter free routes have zero reported pricing and enforce a zero-price request ceiling. Groq access requires a confirmed free account. Z.ai access probes only the explicitly free Flash allowlist. Shared capacity is limited and can become unavailable. Exhausted free access stops without switching to paid service; trials need a real future expiry.

Super AgentOS remains the default until the user chooses another available brain. AgentOS retains context, permissions, approvals, execution, memory, cancellation, records and verified delivery. Model outputs remain advisory and cannot approve actions. Trading wallet rules and Launchpad's verified genuine published-project requirement remain authoritative.

Model availability depends on the connected provider and remaining allowance. The picker shows the access currently available to the user.

Working across AgentOS Studios

AgentOS brings Natural Language, Code, Primeflow, Media, Gaming and Trading into one workspace. Each studio keeps its own tools and working area. The shared composer carries the conversation and selected Project into the work, while studio-specific resources stay attached to their original task.

Users can work with one available brain through Single, compare separate answers through Compare, or bring contributions together through Team. Compare and Team use a chosen lead when turning proposals into one execution plan. Available models and access limits are shown in the model picker. Super AgentOS remains the default until the user chooses another brain.

Plan and Goal serve different purposes. A Goal describes the desired result; a Plan contains the steps for reaching it. Users can keep several named Plans, link them to a Goal or use them independently. Saving a Plan or Goal does not start work. Their compact controls sit above the composer, while Summary keeps the conversation's decisions, files, sources and next steps available separately.

Actions that need permission show a compact review with expandable details. Users can approve the requested action, inspect its progress and stop further work. When an outcome is uncertain, AgentOS keeps that work unresolved until its result can be checked.

Progress reflects recorded work. Files belong to the selected Project, workflow results remain available in Primeflow, saved media appears in its project, and game changes retain their preview. Trading keeps its account, wallet and approval controls. A submitted order and a filled order remain distinct states.

The aim is a consistent way to plan, act and review results across studios, with more usable space and clear access to the details when needed.

Universal model execution candidate

Universal execution is an open release candidate covering the complete available model catalogue, including native Super AgentOS and the Argon, Muse and Quiver integrations. Each available reasoning model is intended to direct authorized actions in Natural Language, Code, Primeflow, Media, Gaming and Trading. Specialist models perform their supported operations from any Studio and deliver results into the originating Project. Availability and supported operations depend on the exact connected provider and model.

Vault, Studio selectors and execution share exact model identity, supported operations and provider limits. The selected reasoning model receives project context and recorded tool results through native functions or a validated action contract. Single, Compare, Team and Once retain their selected model or lead. User-owned operations use the selected Vault account; existing included access keeps its existing restrictions. This candidate adds no AgentOS-funded provider usage. Credentials stay server-side.

Studio	Required delivered result
Natural Language	Saved project outputs and connected-tool receipts.
Code	Changed files, successful checks and a working build or preview.
Primeflow	Saved workflows, executed nodes and delivered outputs.
Media	Generated or edited assets, valid previews and downloadable exports.
Gaming	Saved game changes, a successful build and playable preview.
Trading	Authorized broker orders, reconciled status, fills and positions.

Specialist operations include image, video, SVG, audio, transcription, segmentation, evaluation and supported embedding or moderation. Jev decisions select explicitly configured action branches, which require their normal review and actual execution receipts. Provider completion does not mark an unexecuted branch complete. Output continuation

follows provider capabilities, while interrupted or uncertain submissions retain the same recorded job for recovery.

Trading continues to enforce the connected account, exact order review or delegated authority, expiry, wallet binding and duplicate prevention. An uncertain submission must be reconciled before retry; rejected, submitted, partially filled, filled and cancelled orders keep distinct statuses.

Release verification remains incomplete. Genuine provider receipts, delivered outputs across all six Studios, live broker acknowledgement and reconciled status, and final graphics from verified production screens are still required.

Catalogue visibility, a proposed action or a successful deterministic check does not certify completed execution.