

AGENTOS TECHNOLOGIES LTD.

ECONOMY FRAMEWORK

The AgentOS Economy

Subscriptions, creation, monetisation
and enforceable tokenization rules

**\$sAGENT powers the entire economy
of the AgentOS ecosystem.**

5 OCTOBER 2026

Mandatory build requirements, unique tickers, integrated external launchpads, creator earnings and product-access rules.

CONTENTS

Inside the economy

01 The economic model	3
02 Mandatory economy rules	4
03 What qualifies as a genuine build	5
04 Unique tickers across AgentOS	6
05 Integrated external launchpads	7
06 Enforcement before launch	8
07 People and products	9
08 \$sAGENT, product tokens and credits	10
09 Subscriptions and doing work	11
10 Directly priced products	12
11 Tokenizing a creation	13
12 Creator earnings and rewards	14
13 Holding a product token for access	15
14 Paying with a product token for access	16
15 Comparing the monetisation choices	17
16 A subscriber becomes a creator	18
17 A tokenized movie: two access models	19
18 Changes, failures and continuing compliance	20
19 How value moves through AgentOS	21
20 Revenue and the 50% buyback policy	22
21 Clear offers and operating terms	23
22 Definitions and reference guide	24

The economic model

\$sAGENT, the Super Agents token, powers the entire economy of the AgentOS ecosystem. Users pay for AgentOS subscriptions with it, and creators receive it as the primary payment currency when they sell their products directly. Creators can also tokenize genuine creations through integrated external launchpads and earn under the selected launchpad's creator-reward terms.

Paying to work

A user pays for an AgentOS subscription in \$sAGENT and receives Agent Credits, also called compute, to do work. **Compute and Agent Credits are the same thing.** They are not separate resources, parallel balances or additional purchases. The subscription, the allowance it provides and the work done with that allowance form one connected payment relationship.

Creating and earning

Creators can sell Apps, Skills and artifacts such as movies and pictures. They can attach a direct access price and receive \$sAGENT from buyers, or tokenize a genuine product and participate in the selected integrated launchpad's creator-reward mechanism. Earning is not restricted to executable software or to Apps and Skills.

For a tokenized product, the creator must define access before launch. Holder access lets a user qualify by holding the product's token. Token-payment access requires the user to spend that token. The product token is distinct from \$sAGENT, and creator rewards are distinct from an access payment.

The rules that govern tokenization

No genuine build, no token launch. No memecoin launches. No duplicate AgentOS tickers. No exemption through an external launchpad. Every tokenized product must be attached to a particular genuine build that provides created value. These requirements apply to every AgentOS-originated launch or tokenization route, across all integrated launchpads and ecosystems.

AgentOS does not currently operate its own independent launchpad. It integrates launchpads from different ecosystems so creators have options. AgentOS retains responsibility for enforcing its economy rules before routing a launch; an external provider's willingness to launch a token does not replace that approval.

The revenue connection

The buyback policy allocates **50% of total revenue generated in the ecosystem to buying back \$sAGENT.** This revenue-linked mechanism sits alongside the token's subscription and direct-payment uses. It does not change a creator's access model or make a product token the main currency of AgentOS.

The result is an economy built around people doing work, creating value and earning from real products. Token creation follows the build; it is not a substitute for the build.

Mandatory economy rules

These are mandatory AgentOS economy rules, not optional launchpad preferences. A tokenization request must satisfy all of them before AgentOS initiates, authorizes or hands off a launch. Selecting another ecosystem or launchpad does not change the requirements.

Rule 1 | A genuine build must exist before launch

Every tokenized product must be attached to a particular, genuine build that already provides created value. A creator must present the actual App, Skill, movie, picture or other deliverable associated with the token. An idea, future roadmap, token name or promise to build later is not sufficient. **No build means no launch.**

Rule 2 | No memecoins or productless tokens

AgentOS does not permit memecoin launches or tokens that have no genuine associated product. A token contract, promotional page, logo or unrelated attachment cannot be used to disguise a productless launch. The product must have a clear value of its own; token trading is not a substitute for that created value.

Rule 3 | Every tokenized product has a unique ticker

No two tokenized products may share a ticker within AgentOS. Uniqueness applies across the AgentOS economy, not separately for each chain, launchpad, creator account or product category. A ticker already assigned to one product cannot identify another product through a different route.

Rule 4 | External launchpads do not create exceptions

AgentOS integrates independent launchpads rather than operating its own independent launchpad today. Every AgentOS-originated product routed to an integrated launchpad must still pass the same build, ticker and tokenization checks. External launchpad terms can add requirements; they cannot waive AgentOS rules.

Rule 5 | The approved product remains identifiable

The genuine build, product identity, unique ticker and resulting token must remain linked. A creator cannot secure approval for one build and launch a token for a different or missing build. A launch must not be represented as AgentOS-approved when the completed result differs from the approved request.

Required consequence

A failed or unverifiable rule must block the AgentOS launch route. The request may proceed only after correction and a fresh successful check. A warning, disclaimer, paid fee, privileged account or permissive launchpad is not an exemption.

What qualifies as a genuine build

A genuine build is the actual creation associated with a proposed product token. It must be identifiable, available for verification and capable of providing the value described in its offer. The token is attached to the creation, not to a promise that a creation might appear later.

Software and creative work both count

For an App, the evidence should show the application and the function offered to users. For a Skill, it should show the capability and a verifiable example of what it does. For a movie or picture, the evidence is the actual creative work and the access being offered. Other products must provide evidence appropriate to their form.

Created value is not limited to code or business productivity. A movie can provide entertainment; a picture can provide artistic value. A genuine creative artifact is not excluded because it is not software. Equally, attaching an unrelated image to a productless memecoin does not make that launch eligible.

Minimum evidence before approval

The launch record must identify the creator, the specific product, its description, the actual build or artifact, the version being reviewed and the proposed relationship between that product and its token. Evidence must be sufficient to inspect or test the claimed value. The creator must establish that they created the work or have permission to offer and tokenize it.

A reproducible version reference, content fingerprint or equivalent immutable record must identify the reviewed build. This prevents an approval from silently being reused for different content. Where a product contains several files or components, the record must identify the complete reviewed package rather than an unrelated sample.

What does not qualify

A name and ticker with no product; a marketing page describing future development; an inaccessible placeholder; an empty or nonfunctional package presented as working software; a copied attachment with no demonstrated authority; or an unrelated artifact added solely to obtain launch approval must not pass.

An early version must be evaluated on the value it already delivers, not on promised future features. A prototype label is neither automatic approval nor automatic rejection: the actual build must satisfy the genuine-build requirement and accurately describe its present limits.

Verification outcome

Approval requires a verified build and a clear product-token link. Missing evidence must place the request on hold or return it for correction; a fabricated build or productless launch must be rejected. Material changes require renewed review before launch. No launch route may offer a "build later" exception.

Unique tickers across AgentOS

Each tokenized product must have its own unique ticker within AgentOS. A ticker identifies one product across the AgentOS economy. A different chain, external launchpad, creator account or product category does not make a duplicate acceptable.

One shared uniqueness check

All AgentOS tokenization routes must use the same authoritative ticker registry. A route must check the requested ticker against existing assignments and active reservations before launch. The ecosystem ticker \$sAGENT must remain associated with the Super Agents token, not be assigned to a creator product.

Checks must compare tickers consistently rather than treating presentation differences as new identities. Capitalization, a display-only dollar sign or surrounding whitespace must not create duplicate exceptions. The permitted ticker format must be defined and enforced consistently; deceptive lookalike characters must not defeat the check.

Reserve before launching

An available ticker must be reserved for the verified product before a launch is authorized. Checking availability and reserving the ticker must be one coordinated operation, so two simultaneous requests cannot both receive approval. The reservation must identify its creator, product, reviewed build and launch request.

Changing the destination launchpad must not create a second reservation or bypass an existing assignment. A retry must refer to the original launch request. If a launch result is uncertain, its ticker must remain protected until that result is reconciled; a timeout alone must not make it available to another product.

Bind the completed token to the product

After a verified launch, the registry must link the product and its ticker to the actual token address and network, together with the selected external launchpad and reviewed build. Product access and payments must use that verified token identity, not trust a ticker label alone.

Updates to an existing build must preserve its product identity. A migration, relaunch or additional network representation must not create an unrelated product under the same ticker. This framework does not grant blanket permission for multiple tokens or deployments; any supported change must preserve the no-duplicate rule.

Prevent reuse and impersonation

A ticker assigned to a successfully tokenized product must not be reassigned to a different product, including after deletion or retirement. Only an unused reservation may be released after a verified cancellation or failure establishes that no token was launched. Published release terms must preserve the record; unresolved reservations remain unavailable.

AgentOS ticker uniqueness covers its own registry and participation routes. It is not a claim that no unrelated token elsewhere can use the same symbol. Such an external token must never be mistaken for the AgentOS-linked product token.

Integrated external launchpads

AgentOS does not currently have its own independent launchpad. It integrates launchpads from different ecosystems so users can choose among supported external routes. The build flow is the AgentOS experience through which that choice is made; the selected external launchpad carries out the launch under its own terms.

What AgentOS is responsible for

AgentOS must verify the genuine build, enforce ticker uniqueness, record the selected access model and approve the specific launch request before handoff. It must retain the link between the product, build, ticker, selected destination and completed token. These responsibilities remain with AgentOS even when the launch interface or transaction execution belongs to another provider.

What the selected launchpad determines

The selected integration determines the available network, launch mechanics and applicable external terms. Creator rewards depend on that launchpad's actual qualifying activity, calculation method, payout currency and settlement conditions. The document does not assign one universal reward percentage or assume every launchpad has an identical rewards programme.

Before the creator chooses a route, AgentOS must present the applicable conditions. A route without a creator-reward mechanism cannot be advertised as providing those rewards. The displayed options must reflect integrations actually available to the user; this framework does not announce particular provider names or networks.

Choice of provider is not choice of policy

All AgentOS-originated launches remain subject to the same economy rules. A creator cannot move from a rejected route to another integrated launchpad to launch the same noncompliant request. A provider's permission to launch a token is necessary for using that provider, but it is not a substitute for AgentOS approval.

A compatible integration does not have to store every AgentOS record itself. AgentOS may retain the authoritative product-build-token record, but it must be able to verify that the completed external launch matches the approved request. Where a route cannot preserve or verify that link, it must not be enabled for AgentOS-originated launches.

The scope of enforcement

These obligations cover every tokenized product originating in or routed through AgentOS: the build flow, an embedded launch experience, a supported handoff, and any other interface that can initiate the action. The policy follows the AgentOS-originated product rather than stopping at the boundary of a particular screen.

AgentOS cannot claim control over entirely independent activity outside its systems. Such activity does not acquire AgentOS approval by reference to a build or ticker. Any later request for AgentOS recognition must pass the applicable admission and identity checks, rather than importing an outside launch as automatically compliant.

Enforcement before launch

A launch must be blocked until the same mandatory checks have passed, whatever interface or external launchpad is used. A front-end notice alone is insufficient. Every initiating route must obtain authorization from the shared AgentOS policy controls.

1 | Register the genuine build

Create a product record containing the creator, actual build or artifact, reviewed version and evidence of created value. Confirm that the product is genuine and that the tokenization proposal is not a memecoin or productless launch. Missing evidence must stop the request.

2 | Reserve its unique ticker

Check the shared ticker registry and reserve the available ticker for that product and launch request. Reject collisions across all AgentOS routes, including conflicting in-progress reservations. A creator may correct the ticker and submit again, but cannot override another product's assignment.

3 | Define access and select the route

Record whether access requires holding the product token or paying with it, the relevant amount or condition, and what users receive. Select a supported external launchpad and network. Record the applicable rewards and launch terms before the creator confirms the request.

4 | Authorize exactly that request

Bind authorization to the approved creator, product, build version, ticker, access terms and destination. Changed parameters require renewed checks. Authorization must not be a reusable permission to launch a different product, replace its build or switch into a noncompliant route.

5 | Recheck at handoff

Immediately before initiation or the signing handoff, confirm that approval and ticker reservation remain valid and that the launch parameters match. If verification is unavailable, inconsistent or expired, stop the launch. An external interface must not be used to alter the approved request without review.

6 | Reconcile before recognition

Record the returned transaction reference and independently verify the completed result against the approved launch. Bind the actual token address and network to the product and ticker. A success message or provider callback alone must not mark the product as successfully tokenized.

7 | Preserve the record and handle failures

Retain approval evidence, the completed token link and decision history. Failed or uncertain launches must enter correction or reconciliation rather than being blindly retried. AgentOS must not list, promote or grant launch-complete status to an unmatched token. A later change must follow the lifecycle controls in section 18.

If AgentOS cannot enforce these checks for a particular integration, that route must remain unavailable for AgentOS-originated tokenization. The user's selection of an external launchpad never removes the rules.

People and products

The AgentOS economy involves more than a platform selling subscriptions to customers. It also involves people creating products, other people accessing those products, and creators receiving earnings. These roles can overlap: a subscriber can become a creator, a creator can buy another person's work, and a customer can hold the token associated with a product they use.

Users and subscribers

A user subscribes because they want to do work through AgentOS. The economic relationship starts with a subscription paid in \$sAGENT and the Agent Credits/compute provided by that subscription. The allowance gives the user the capacity to work within their plan. Their purpose may be to complete a personal task, produce something for their own use or create a product for other people.

A subscription is not itself a sale of every creator's work. Paying AgentOS for the ability to work and paying a creator for access to a particular product serve different purposes. A user can participate in both relationships without either payment becoming a second name for the other.

Developers and creators

Developers can create Apps and Skills. The wider creator category also includes people producing artifacts such as movies and pictures. The economic model does not confine commercial participation to people who publish executable software.

A product is a creation offered to others under a stated monetisation and access model. An artifact is an output such as a movie or picture that can itself become that product. A creation does not need to be described as an App or Skill before its creator can sell or tokenize it.

Buyers, audiences and token holders

A buyer of a directly priced product pays its access price in \$sAGENT. An audience member using a holder-access product meets the stated holding requirement for that product's token. A customer of a token-payment product spends the product's token to obtain access.

These distinctions matter because the same person may take different actions for different products. Buying access to one movie, holding the token for another and paying a third movie's token are different relationships, even though all three involve watching creative work.

AgentOS and integrated launchpads

AgentOS provides the environment in which subscriptions, work, creation and monetisation connect. It integrates external launchpads from different ecosystems; it does not currently operate its own independent launchpad. AgentOS must enforce its economy rules before routing a product to the selected provider and must verify the resulting token-product link.

The practical result is a two-sided economy: people obtain services and creations, and people supplying those creations can earn. The same participant can contribute to both sides rather than being permanently classified only as a consumer or only as a developer.

\$sAGENT, product tokens and credits

Three terms need clear definitions: \$sAGENT, a product token and Agent Credits. They describe different roles. Compute is not a fourth item; it is another name for Agent Credits.

\$sAGENT: the ecosystem payment token

\$sAGENT is the main token of the AgentOS economy. Users pay their AgentOS subscriptions with it. Creators who directly price their products receive payment in it. The revenue-linked buyback policy also purchases \$sAGENT.

These roles connect \$sAGENT to both the platform relationship and creator commerce. A subscription payment goes toward using AgentOS. A direct product payment goes toward accessing a creator's work. The same token is used, but the buyer is purchasing different things.

Product tokens: tokens for particular creations

A product token is launched for a verified genuine creation through a selected integrated external launchpad. It has a unique AgentOS ticker and is linked to that product, its reviewed build and the verified token address and network. It is distinct from \$sAGENT: a movie token does not become the currency for AgentOS subscriptions.

A creator can use the product token as the basis for holder access or require payment in that token for access. The product token also participates in the tokenization route through which the creator earns creator rewards. The exact reward rules belong to the selected integrated launchpad's creator-reward mechanism.

Agent Credits: the subscription's working allowance

Agent Credits are the compute provided through a user's AgentOS subscription. They describe the allowance used to perform work on the platform. **Agent Credits and compute are the same allowance, not parallel balances.**

The distinction from \$sAGENT is one of function. \$sAGENT is used to pay for the subscription. Agent Credits/compute are what that subscription provides for doing work. Receiving an allowance does not make that allowance a second token for selling movies, buying pictures or launching product economies.

Reading a transaction correctly

When a user pays \$sAGENT to subscribe, the outcome is subscription access and the associated Agent Credits/compute. When a user pays \$sAGENT for a directly priced picture, the outcome is access to that picture under its stated terms. When a user holds a movie's product token, the outcome depends on whether that movie offers holder access.

The currency, the purchase and the access condition should always be identified separately. This prevents a wallet balance from being confused with an active subscription, or ownership of a product token from being confused with a completed access payment.

The simplest way to remember the relationship is: **\$sAGENT pays for participation and direct purchases; Agent Credits/compute support work; product tokens belong to individual tokenized creations.**

Subscriptions and doing work

AgentOS subscriptions are a direct utility for \$sAGENT. Users pay with the token because they want to use AgentOS to get work done. Their subscription provides Agent Credits, also called compute, as the allowance for that work.

One payment relationship

The economic sequence is: **pay for the subscription in \$sAGENT, receive Agent Credits/compute, and use that allowance to work.** These are stages of the same subscription relationship, not three independently named token utilities.

The subscription connects the payment token to the working allowance. Its purpose is to give the user the capacity to work through AgentOS under a defined plan. The payment and the allowance belong in the same description of what the subscriber is buying.

What the subscriber receives

The subscription establishes the user's access to AgentOS and the allowance associated with their plan. The plan's terms determine the price, the amount of Agent Credits/compute and the relevant subscription conditions. This lets users understand the commitment they are making and the working allowance that accompanies it.

The allowance supports doing work rather than serving as the creator's payout currency. A person can use their subscription to develop an App, prepare a Skill or produce an artifact. What they subsequently charge for their finished creation is a different commercial decision.

Subscription use and creator commerce

A creator's subscription supports the work involved in making a product. When the product is offered to other people, its price or token-access rules determine how those people obtain access. Paying for creation and charging for the finished creation are not the same transaction.

Likewise, a buyer's subscription does not automatically settle the access price of a separately priced product. The subscription provides platform use and the associated working allowance. A creator's product terms identify any separate payment or token condition required to access that product.

The relevant distinction is between **using AgentOS** and **buying someone else's creation**.

Renewal and ongoing use

A subscription relationship can continue when a user renews under the applicable plan. The commercial reason for renewal is continued use of AgentOS: the user wants the platform and the allowance needed to keep working. The economy therefore has a service-payment role for \$sAGENT that does not depend on the user buying a particular creator's token.

The subscription description should make the price, allowance and applicable conditions understandable in one place. Credit expiry, rollover, additional allowances and similar plan details belong in the subscription terms. They describe how the allowance works within a plan, rather than creating another currency in the ecosystem.

Directly priced products

Direct pricing is the clearest sale relationship in the AgentOS economy. A creator makes a product, attaches an access price and receives payment in \$sAGENT from users who purchase access.

The product does not need its own token. Its value to the buyer can be the usefulness of an App, the capability provided by a Skill, the experience of a movie or the appeal of a picture. The transaction is a purchase of access to that creation under its stated terms.

The creator's side of the sale

The creator decides what is being offered and what the buyer receives. The product description and price establish the commercial offer. When a user chooses that offer and completes the required payment, the creator earns through the direct-sale route.

\$sAGENT is the primary payment currency for this route. The economic model does not replace the creator's direct-sale payment with Agent Credits. Credits support the user's work on AgentOS; they are not another name for the proceeds from selling a product.

The buyer's side of the sale

The buyer needs to understand the product, its price and the access being purchased. Paying for access to a movie should be distinguishable from buying access to an App or purchasing a picture. The payment currency can remain \$sAGENT while the product and access terms differ.

A purchase also needs a clear scope. Access could be defined for a particular item, edition or stated period, but those terms must come from the creator's offer. Merely attaching a price does not establish lifetime access, unlimited use or every future version of a product.

Direct pricing across different creations

An App developer can charge for access to an application. A Skill creator can charge for access to a capability. A movie creator can charge viewers to watch their work. A picture creator can make an image available to paying users. These are illustrative applications of the same payment model, not separate token utilities for every file type.

The important expansion is the category of people who can earn. The creator economy includes software and creative output. It is not accurate to describe Apps and Skills as the only products that can generate earnings inside AgentOS.

Why direct sales matter

Direct pricing gives creators a way to earn from customers who want the product itself, without requiring them to launch a product token. It also gives \$sAGENT a commercial role beyond platform subscriptions: the token is used to pay other participants for their work.

The direct-sale relationship is therefore **creation, stated price, payment in \$sAGENT and access**. Each sale should be understandable on those terms, without a token launch being added as a requirement.

Tokenizing a creation

Tokenization is a creator monetisation route in which a genuine product is associated with its own token. **The build must exist and pass AgentOS review before launch.** The creator chooses a supported integrated external launchpad within the build flow and can earn through that provider's applicable creator-reward mechanism.

Start with the actual creation

The App, Skill, movie, picture or other deliverable is the product. The creator must identify it, demonstrate its created value and establish permission to offer it. A future idea, token contract or promotional attachment cannot replace that evidence. Memecoin and productless launches are prohibited.

The build record must identify the version being tokenized. Approval attaches to that product and version, not to a creator's general account. Uploading a genuine build once does not authorize an unlimited series of unrelated token launches.

Set the token identity and terms before launch

The creator proposes a unique ticker and receives a reservation only after the AgentOS-wide collision check passes. The product description, ticker and token relationship must identify the same creation. A duplicate ticker must be corrected before proceeding, even when the external launchpad would otherwise accept it.

The creator must also choose and disclose the access model before authorization: users either qualify by holding the product's token, or purchase access by paying with it under the stated terms. The required balance or payment, the access provided and the selected launchpad's reward terms belong in the approved request.

Use the selected external launchpad

AgentOS verifies the request and routes it only through an integration that can preserve the approved product, ticker and launch conditions. The selected external launchpad performs the launch. Its fees, mechanics and reward terms must be disclosed; those details are not a waiver of AgentOS economy rules.

After launch, AgentOS must verify the completed transaction and actual token identity before recording success. The resulting token must remain linked to the approved product and build. Failed or uncertain launches must be reconciled, not treated as permission to create another token or reuse the ticker.

Keep earnings, access and ecosystem payments distinct

Creator rewards follow the selected launchpad's terms. Holder access does not automatically generate a payment, and token-payment access has its own settlement terms. Neither model replaces \$sAGENT as the payment token for AgentOS subscriptions and directly priced products.

Tokenization expands how creators earn from genuine work. It is not a standalone permission to issue coins without a build, and it does not guarantee a reward amount merely because a token has been launched.

Creator earnings and rewards

The AgentOS economy has two primary creator monetisation routes: directly pricing products and tokenizing genuine products. Direct sales are paid in \$sAGENT. Tokenization can provide creator rewards under the selected external launchpad's terms. Where a tokenized product also requires an access payment, that payment must be accounted for separately from launchpad rewards.

Earnings from direct sales

For a directly priced product, a user pays the access price in \$sAGENT. The creator earns from that product purchase. The sale is tied to an identifiable offer: a particular creation is being made available under stated access terms.

The fact that the creator used Agent Credits/compute while making the product does not change the currency of the sale. The subscription allowance supported the work. The customer's \$sAGENT payment pays for access to the result.

Earnings through tokenization

For a tokenized product, the creator earns through the selected integrated launchpad's creator-reward mechanism. The product token and the reward terms define this earning route. A direct access sale is not the only event through which a creator can build an earning relationship within AgentOS.

Creator rewards need to be described on their own terms. The relevant launchpad information should identify what activity qualifies, how the reward is calculated, what currency is used and when the reward becomes available. The economic model establishes this route without assigning a reward percentage or an automatic payout to every token action.

Rewards are not the access rule

A creator's ability to earn rewards does not, by itself, tell a user how to enter or use the product. Access depends on the chosen holder-access or token-payment conditions. The reward mechanism and the product's access mechanism are related to the same tokenized creation, but they answer different questions.

The reward mechanism answers, "How does the creator earn through tokenization?" The access rule answers, "What must this user do to access the product?" One cannot be substituted for the other.

Earning and participating again

A creator who receives \$sAGENT from direct sales can use those tokens for their own AgentOS subscription or for directly priced products offered by others. This creates a possible path from earning back into participation. It is an available use of the same payment token, not a requirement that creators spend their earnings in a particular way.

Creator-reward proceeds should not automatically be described as \$sAGENT unless the reward terms specify that currency. Product-token holdings, reward balances and direct-sale proceeds are not interchangeable labels.

The creator economy is broader than an App marketplace. Creators can sell access in \$sAGENT or tokenize genuine creations under the mandatory economy rules. A tokenized product's access payments, where required, must remain distinguishable from its creator rewards; they must not be described or counted as the same receipt.

Holding a product token for access

Holder access makes token ownership the condition for using a particular product. A user qualifies by holding the product's token according to the creator's stated requirement. The user does not spend those qualifying tokens as an access payment.

This model answers the question, “Does this user hold what the product requires?” It is different from asking whether the user has completed a payment for the current access request.

What the user does

The user obtains and holds the relevant product token. AgentOS must verify the approved token address and network linked to the product, then check the stated holding condition. A different token with the same display symbol, or a token for another creation, does not satisfy the requirement.

A holding condition may specify a required balance. The creator's terms establish that balance and identify which token qualifies. The important feature is that the qualifying tokens remain held rather than being transferred as an access payment.

What the user receives

The user receives the access described by the product. For a movie, that could be access to watch the stated movie. For another creation, the offer must identify the corresponding access. Holding a product token should not be interpreted as access to every product made by that creator unless the actual offer says so.

The terms also need to explain when the holding condition is checked and what happens if the user no longer meets it. An ongoing holding requirement and a one-time eligibility check are different rules. Making the chosen rule clear tells the audience whether continued access depends on maintaining the qualifying balance.

What the creator receives

The creator's tokenization earning route is the creator-reward mechanism. The act of admitting a qualifying holder is not automatically a direct-sale payment from that holder. An access check does not become a token transfer simply because the creator has monetised the product through tokenization.

This separation is what makes the model distinct. The audience can qualify through ownership, while the creator's earnings follow the selected integrated launchpad reward rules rather than being described as a payment collected on every viewing.

Illustrative movie offer

A creator tokenizes a movie and states that users meeting the movie token's holding requirement may watch it. A qualifying user requests access while retaining their tokens. The requirement is satisfied through ownership, not by sending the balance away.

The offer should therefore be described as **hold the product token to qualify for access**, not **pay the product token to watch**. These phrases describe different actions and should not be used interchangeably in listings, explanations or payment prompts.

Paying with a product token for access

Token-payment access requires a user to spend a product's token to obtain access. Holding the token in a wallet is not enough. The user must complete the payment required by the product's offer.

This model answers the question, "Has the user paid the required amount for this access?" It differs from holder access because the token is used in a transaction, rather than retained solely as a qualifying balance.

What the user does

The user identifies the relevant product, reviews its access price and pays in the verified token linked to that product. The token address and network must match the approved product record. Holding \$AGENT or another token does not complete a payment in the movie's token.

The required payment amount and the scope of access should be clear before the user proceeds. Holding enough tokens makes it possible to pay; it does not mean the payment has already happened.

What the payment buys

A token payment purchases the access described in the offer. The creator must specify whether it covers a particular use, a stated period, a defined product version or another clearly described entitlement. The economic model does not assume that every product charges on every visit or that one payment grants permanent access.

After payment, the relevant purchase terms determine the user's access. A token-payment product should not be treated as a holder-access product merely because the customer still has some tokens left over after paying.

Relationship to creator rewards

The product remains a tokenized creation, so creator rewards belong to its tokenization model. The access payment is a separate, user-facing requirement: the customer must spend the product token to enter or use the product.

The access payment and creator reward need separate descriptions. The product and selected integrated launchpad terms establish where the payment goes, how it is settled and how creator rewards are calculated. This lets users understand the access purchase and creators understand the earnings associated with their product.

Illustrative movie offer

A creator tokenizes a movie and requires viewers to pay its access price in the movie's token. A user with a sufficient token balance still needs to approve and complete that payment. Access follows the purchase under the movie's stated terms.

The offer should therefore be described as **pay with the product token to access the product**. It should not suggest that merely holding the token is enough.

The distinction is practical: holder access uses ownership as the qualification, while token-payment access uses a completed payment as the qualification. The creator chooses the model; tokenization alone does not choose it.

Comparing the monetisation choices

A creator makes two related decisions. First, they choose the route by which the product will be monetised. Second, where the product is tokenized, they define the action required for access. Keeping these decisions separate avoids treating tokenization as a complete access policy.

The monetisation decision

Under direct pricing, the creator attaches a price to the product and receives \$sAGENT from users who purchase access. A product token is not required for this route.

Under tokenization, AgentOS verifies the genuine build and reserves its unique ticker; the creator defines access. AgentOS then authorizes a launch through a selected integrated external launchpad. Creator rewards follow that provider's terms. Launchpad selection does not exempt the product from AgentOS rules.

The access decision

For a tokenized product, holder access makes qualifying ownership the condition. Token-payment access requires users to spend the product token. These choices create different user experiences and should be named accordingly.

Offer	What the user does	Token involved	Creator earning route
Directly priced product	Pays the stated access price.	\$sAGENT	Direct-sale payment in \$sAGENT.
Tokenized product: holder access	Holds the qualifying balance; does not spend it as the access payment.	The product's own token	Creator rewards under the selected integrated launchpad rules.
Tokenized product: token-payment access	Spends the required amount to purchase access.	The product's own token	Creator rewards; the access payment has separately defined settlement terms.

Matching the offer to the intended relationship

Direct pricing expresses a conventional product purchase: the user pays the stated price to receive the stated access. Holder access expresses an ownership-based relationship: the user qualifies by keeping the required product-token balance. Token-payment access expresses a purchase made in the product's own token.

The creator's choice should follow the relationship they intend to offer. A movie available to qualifying holders is not the same offer as a movie that requires a token payment for each access purchase. Neither should be described as the other simply to simplify a listing.

An offer should state its complete access condition before the user proceeds. Where a creator describes holder access, the audience should see the holding requirement. Where the creator describes a purchase, the audience should see the required payment. Additional conditions belong in the offer itself, not in an unexpected step after joining or paying.

A subscriber becomes a creator

The following journey is illustrative. It explains how the defined relationships connect without assigning actual prices, credit allowances or product names to AgentOS.

Starting with a subscription

A person wants to create work using AgentOS. They select the appropriate subscription and pay for it in \$sAGENT. The subscription provides Agent Credits/compute, which they use as their allowance for doing work.

The person begins as a subscriber rather than as a seller. Their immediate aim is to use the platform and turn an idea into a finished creation. Whether that creation later becomes a product is a separate decision they make as the work develops.

Producing an artifact

The subscriber creates a picture. At this stage, the artifact is the result of their work. The creator then decides to offer it to other users as a directly priced product. They describe what a buyer will receive and attach an access price.

This does not require the picture to be relabelled as an App or Skill. Its status as a creative artifact does not exclude it from monetisation. The creator's decision is about making the work available on stated commercial terms.

Receiving a direct payment

Another user wants the picture and accepts the offer. That user pays its access price in \$sAGENT. The creator receives payment through the direct-sale route, and the buyer receives the access described in the offer.

The buyer is paying for the finished picture, not funding a new subscription for themselves. The creator's earlier subscription and this later sale have different purposes, even though both use \$sAGENT as the payment currency.

Using earnings within the ecosystem

The creator can then use their \$sAGENT earnings for a subscription or to buy a directly priced product from someone else. For example, they might purchase access to a Skill that supports another project. This is a possible circulation path, not a rule requiring creators to reinvest or remain within AgentOS.

The original customer, the picture creator and the Skill creator now occupy different roles in connected transactions. The picture creator has moved from subscriber to seller to buyer without needing a different ecosystem currency for each direct payment.

What the journey demonstrates

The subscription provides the working allowance. The allowance supports creation. A direct sale turns that creation into earnings. Those earnings can support further participation through the same main payment token.

The App and Skill economy remains part of this picture, but it is not the boundary of the creator economy. A picture can generate a paid transaction just as an App or Skill can. The wider model connects people who create different kinds of value rather than limiting earnings to one category of product.

A tokenized movie: two access models

This illustrative journey follows a creator who has already made a movie and wants to tokenize it. It does not announce a real movie, ticker, price or integrated provider. The two access models below are alternatives, not two successive charges that every viewer must satisfy.

Before either model can launch

The creator submits the actual movie, demonstrates the value being offered and establishes permission to tokenize the work. AgentOS verifies the build, records the movie's identity and version, and reserves an available ticker that is unique across AgentOS. A promotional page for an unmade movie does not qualify.

The creator selects an available integrated external launchpad and reviews its launch and creator-reward terms. They choose one of the following access models before authorization. AgentOS then approves the specific build, ticker, access terms and destination together; the completed launch must match that approval.

Alternative A | Hold the movie token

The offer states that viewers who meet the movie token's holding requirement may watch the movie. A viewer obtains and holds the verified token. AgentOS checks the approved holding condition; the viewer does not transfer those qualifying tokens as an access payment.

The offer explains the required balance, what access it grants, when eligibility is checked and what happens if the balance no longer qualifies. A qualifying viewer opening the movie is not automatically a direct sale. The creator's reward earnings follow the selected integrated launchpad's rules.

Alternative B | Pay with the movie token

The offer instead requires a payment in the movie's token. A viewer obtains enough of the verified token, reviews the access price and completes the required transaction. Holding the balance alone does not buy access. The purchase terms define what the payment covers and where it is settled.

The access payment is distinct from any creator reward paid under the launchpad's mechanism. It must not be described as the same receipt or assumed to trigger a reward automatically. Access is recognized after the required payment is verified against the approved product-token identity.

The rules do not change between alternatives

Both alternatives require the same genuine movie, unique ticker and AgentOS approval. Switching launchpads, access models or networks cannot turn a rejected build into an approved launch. Any material change before launch must be checked again.

The movie's product token remains distinct from \$sAGENT. The difference between the alternatives is the viewer's action: in one, the viewer holds the token; in the other, the viewer spends it.

Changes, failures and continuing compliance

AgentOS economy rules must remain effective after initial review. An approved request is not permission to substitute a build, create duplicate tokens or disguise a different launch. The product, reviewed build, ticker and token must remain traceable through changes and failures.

Changes before execution

If the creator changes the build, product identity, ticker, network, launchpad or material access terms, the prior authorization must no longer be sufficient. AgentOS must check the amended request and confirm the ticker reservation before any launch. A previously verified attachment must not be reused as cover for a different product.

Timeouts and retries

A failed screen or delayed provider response does not prove that no token was launched. AgentOS must check the transaction outcome before retrying or releasing a ticker. The same launch request must not accidentally create multiple tokens when a user clicks twice, an automated task retries or an integration sends repeated callbacks.

Where the outcome remains uncertain, the request must stay in reconciliation. A verified successful launch completes the existing product-token link. A verified failure can enter the documented correction and reservation process. Uncertainty must never be treated as a fresh approval.

Build updates and product identity

Legitimate updates must retain a history of the original approved build and the changed version. A replacement must continue to represent the same product; renaming or updating a record must not silently attach its token to an unrelated creation. Material changes need renewed verification before they are offered through AgentOS.

An unavailable or withdrawn build must trigger review of the product's status and any affected access offers. AgentOS must stop new noncompliant launches or offers and make the problem visible to affected users. The response must preserve transaction records and existing customer claims rather than silently deleting their history.

No retrospective approval by import

An external token cannot become AgentOS-approved merely because a creator later supplies a product link, or because the external launchpad accepted it. Any request for AgentOS recognition must satisfy the genuine-build, ticker and identity requirements. Launching elsewhere after an AgentOS rejection is not a valid bypass into AgentOS.

Limits of platform enforcement

AgentOS can block its own authorizations, integrations, listings and access offers where policy requirements fail. It must record the reason and require correction or renewed review before restoring the affected route. An administrative action must not waive the core economy rules.

These controls do not imply that AgentOS can reverse an external blockchain transaction, erase an externally issued token or seize user holdings. Product retirement, customer remedies and release of genuinely unused reservations require explicit operating terms. Enforcement must remain accurate about what the platform can actually control.

How value moves through AgentOS

The economy connects demand for useful work with the people and products that meet that demand. Its different payment relationships should be understood as connected activities rather than an inflated list of unrelated token functions.

Subscription activity

A user spends \$sAGENT on an AgentOS subscription because they want to work through the platform. The subscription provides Agent Credits/compute. Continued use can lead to renewal under the plan's terms, creating an ongoing service-payment relationship between the user and AgentOS.

This relationship exists whether the user creates something for sale, completes a private task or purchases no creator products at all. The subscription is a utility in its own right because it pays for platform use and the associated working allowance.

Direct creator commerce

When users buy directly priced products, \$sAGENT also moves through creator commerce. A customer pays for work made available by another participant. The creator earns and may use the same token for subscriptions or other direct purchases.

This creates the possibility of circulation across multiple participants. It does not require every recipient to retain or spend the tokens in a particular way. The important economic function is that a common payment token can support both paying the platform and paying creators.

Product-specific token activity

Tokenized creations introduce their own tokens and access relationships. Users may hold those tokens to qualify for products or pay with them to purchase access. Creators earn through the selected integrated launchpad's creator-reward mechanism.

This activity should be described accurately rather than attributed automatically to direct \$sAGENT payments. A payment in a movie's token is a payment in that movie's token. The wider connection to \$sAGENT comes through its established ecosystem roles and the revenue-linked buyback policy, not through pretending every token action uses the same currency.

Product value and economic participation

The reason to subscribe is the ability to do work. The reason to buy a directly priced product is the value of that creation to the customer. The reason to use a tokenized product depends on the creation and the access relationship it offers.

A larger number of tokens or listings does not replace useful products. Every AgentOS-originated tokenized product must remain attached to a verified genuine build with its own unique ticker, regardless of launchpad. The economy rules make created value a condition of participation rather than an optional description.

Economic activity is also not the same as a guaranteed increase in token price. The document defines payment uses, earning routes and the buyback allocation. It does not turn those uses into a promise about market outcomes or require a price forecast to explain why the payment model matters.

Revenue and the 50% buyback policy

The AgentOS buyback policy allocates **50% of total revenue generated in the ecosystem to buying back \$sAGENT**. This is the stated revenue connection in the economic model.

The policy sits alongside the token's payment uses. Subscribers use \$sAGENT to pay for AgentOS. Buyers use it for directly priced products. The buyback allocation provides a separate route by which ecosystem revenue is used to purchase the main token.

The allocation

For every 100 units of ecosystem revenue, the policy allocates 50 units to \$sAGENT buybacks. This is a simple illustration of the percentage, not a report of actual revenue or a statement of a particular reporting currency.

The revenue basis is total revenue generated in the ecosystem, not a substituted measure of platform fees or profit. The reporting method must define how revenue is recorded across the ecosystem without counting the same receipt twice. This accounting requirement does not change the stated 50% allocation.

The connection to the economy

The policy links revenue generation to purchases of \$sAGENT. Revenue from the ecosystem is not disconnected from the main token simply because some activities involve product tokens or different creator relationships. The allocation creates an explicit revenue-linked component alongside subscriptions and direct product commerce.

However, the percentage is an allocation rule, not a statement that every individual payment immediately triggers a separate market order. Purchase timing, execution and reporting belong to the operating arrangements for the policy.

Allocation and execution

A clear account of the policy should distinguish the amount allocated for buybacks from the purchases actually completed. Stating an allocation is not the same as reporting an executed purchase. Reporting should make it possible to understand the revenue basis, the corresponding 50% allocation and the execution against that allocation.

The document also does not treat token market capitalization or token transfer volume as if either automatically equals ecosystem revenue. Revenue reporting needs to use the actual revenue being recorded, rather than a market activity figure substituted for it.

What a buyback means

A buyback is a purchase of \$sAGENT. It does not, by itself, state that the purchased tokens are burned, distributed to holders or assigned to a particular treasury use. Those are separate decisions and should be described separately where applicable.

The economic role of the policy is therefore precise: **half of ecosystem revenue is allocated to buying back the ecosystem's main token**. This is a revenue allocation for token acquisition. It sits alongside creator earnings and product payments without replacing the rules that govern those separate relationships.

Clear offers and operating terms

Users must understand what they are buying, and creators must understand how they earn, before a subscription, purchase or launch is confirmed. Product terms and external launchpad terms supply the operating detail; they cannot remove the mandatory AgentOS economy rules.

Subscription offers

An AgentOS subscription offer must state its price in \$sAGENT, the Agent Credits/compute it provides and its plan conditions. The terminology must describe one working allowance. Renewal, expiry, rollover and any additional-allowance arrangements need explicit terms; this document does not introduce another resource called compute.

Directly priced products

A direct-sale offer must identify the actual creation, its access price in \$sAGENT and what the buyer receives. Access duration, covered versions, allowed uses and relevant conditions must be clear. A platform subscription does not silently include every separately priced creation, and direct pricing does not require a product-token launch.

Tokenized products

The offer must identify the genuine build, reviewed version, unique AgentOS ticker, selected external launchpad and network, and the verified token address once launched. Before authorization, the creator must state the access model, its conditions and the relationship between the token and the creation.

For holder access, disclose the qualifying token and balance, when the holding condition is checked, the access granted and the effect of losing eligibility. For token-payment access, disclose the amount, payment token, recipient or settlement route, what the purchase covers and when access is recognized. An ownership check must not be presented as a payment.

External launch and creator-reward terms

Disclose the selected integration's launch mechanics, applicable charges and actual creator-reward rules. State qualifying activity, calculation method, payout currency and settlement conditions. Do not promise a universal reward rate or describe a provider without rewards as offering them. Product access payments and creator rewards must remain distinguishable.

Records, changes and reporting

Retain the approved build, ticker reservation, policy decision, destination, launch result and product-token link. Material changes must be reviewed and recorded. Offers must not make a failed or unverified launch appear approved, or present an externally launched token as exempt from the rules.

Buyback reporting must preserve the policy of 50% of total ecosystem revenue and distinguish the allocation from completed purchases. It must not count token volume as revenue automatically, or count one receipt repeatedly as it moves between participants.

Detailed terms tell each participant what they pay, receive, hold or spend. They explain implementation without changing the economy's core rules.

Definitions and reference guide

AgentOS economy. The system through which people subscribe, work, create, sell and tokenize genuine products. \$sAGENT is its main payment token.

\$sAGENT / Super Agents token. The token used for AgentOS subscriptions, the primary payment currency for directly priced creations, and the token purchased under the buyback policy.

Subscription; Agent Credits / compute. A subscription is paid in \$sAGENT and provides the allowance used to work. Agent Credits and compute are two names for that same allowance, not separate purchases.

Creator; product; artifact. A creator makes work that can be offered to others. A product is that creation under stated access terms. Artifacts include creative outputs such as movies and pictures. Apps, Skills and artifacts can all be products in the AgentOS economy.

Genuine build. The actual, verifiable creation that provides the value associated with a tokenized product. A token name, future promise or unrelated attachment is not a substitute.

Direct pricing. Attaching an access price to a product, paid in \$sAGENT. The creator earns from the sale; a product token is not required.

Tokenization; product token. Launching a token for a verified genuine build through an integrated external launchpad. The resulting product token is distinct from \$sAGENT.

Integrated launchpad; build flow. An external provider supplies the launch service. The AgentOS build flow connects creation, review and selection of that provider. AgentOS does not currently operate its own independent launchpad.

Unique ticker; ticker registry. A ticker identifies one tokenized product across AgentOS. The shared registry records reservations, assignments and links to verified token identities. Another chain or launchpad does not permit duplication.

Product-build-token link. The recorded association between the product, reviewed build, ticker, actual token address and network. It must survive handoff, retries and legitimate updates.

Creator rewards. Earnings under the selected integrated launchpad's mechanism, with qualifying activity, calculation and payout terms specific to that provider. They are not automatically the same as product-access payments.

Holder access. Users qualify by holding the required product token. They do not spend the qualifying tokens as an access payment.

Token-payment access. Users spend the required product token to buy the stated access. Holding it alone does not complete a purchase.

Buyback allocation. 50% of total revenue generated in the ecosystem is allocated to buying back \$sAGENT. Allocation, execution and later use of purchased tokens are separate records.

AgentOS-originated route. Any tokenization initiated, authorized or routed through AgentOS, including a handoff to an external provider. The same economy rules apply throughout.