avadepin

Useful compute. Clear outcomes.

AvaDepin is being developed to turn available computers into useful, verifiable work.



PRIVATE-BETA ENGINEERING

26 November 2026 · Working-name project

Agree on the work

Clear task and terms

Verify the result

Inspectable acceptance

Account for payment

Explicit settlement states

Proposed economics. Working identity.
Current evidence and future design are identified separately. This paper does not announce a public token launch.

Whitepaper v0.1 · 26 November 2026 · Working-name project

Status: Private-beta engineering with a proposed commercial and token design. The supplied evidence includes isolated local and Avalanche Fuji work. Public paid service, current-release native qualification, and an AvaDepin token launch are not established by this document. The name, symbol, and visual identity are working concepts.

Executive overview

AvaDepin is being developed to turn independently operated computers into a coordinated service for useful, bounded work. A customer should be able to choose a supported task, understand the price and acceptance conditions, and receive an inspectable result. A contributor should be able to connect a compatible computer, choose its limits, pause participation, and understand which completed work was accepted.

The starting point is deliberately narrow: a private coding workbench and invited computers running approved task profiles. The system already contains source implementations for task identity, signed assignments and receipts, independent verification, owner controls, release selection, recovery, and development payment contracts. The next commercial milestone is a repeatable, independently observed journey from an approved task to a useful result and correctly reconciled settlement.

Avalanche provides a practical settlement path. The proposed first public payment integration uses the C-Chain, beginning with its Fuji testnet. It does not require a new blockchain or a new token. An optional token design is documented separately, with explicit allocation and release limits, so reviewers can assess it without mistaking it for an operating economy.

The product promise is simple: **agree on the work, verify the result, account for the payment.** Successful execution, accepted work, settled credit, and a completed withdrawal are distinct events. The interface should make that distinction understandable without requiring people to learn the underlying infrastructure.

1. The problem and initial users

Available computing capacity does not become a useful service merely because it is connected. The difficult work includes matching a task to a capable machine, preserving the original request across failures, checking the output, honoring the contributor's limits, and resolving payment when something goes wrong.

AvaDepin's initial customer hypothesis is small software teams and individual builders with bounded, repeatable work that can be checked. The first contributor hypothesis is an invited owner of compatible hardware who values a predictable setup and control over participation. These are target users, not a claim of current customers or market traction.

The starting use cases are approved coding tasks and structured batch work supported by the existing implementation. Additional inference, media, or other compute profiles must earn their place through an appropriate verifier, operating limits, and measured demand. A familiar task catalog is preferable to promising that any program can safely run on any computer.

Existing cloud services, compute marketplaces, and privately operated clusters are real alternatives. AvaDepin's product hypothesis is that a narrower service can make the full result-and-accounting

journey easier for small teams. This must be tested with users: time to first successful task, accepted output quality, recovery burden, repeat use, and willingness to pay. No market-size, cost-saving, or superiority claim is assumed.

2. A product people can use routinely

The experience has two entry points: **Get work done** and **Contribute a computer**. Technical detail is available when it supports a decision, but it should not obscure the next action.

For a customer, the intended flow is: select a supported task; review the input, deadline, acceptance method and maximum price; authorize it; follow progress; inspect the result; then see the final accounting. A failed or disputed task should explain what happened and what can happen next. A retry should retain the original economic identity rather than accidentally create another charge.

For a contributor, the intended flow is: install a qualified release; connect and complete capability checks; set supported limits; choose when to participate; then review accepted work and any resulting payment. Stop and Pause are controls, not suggestions. After a disconnection or restart, the system must reconcile the previous intent before accepting new work.

The everyday screen should answer four questions: Am I connected? Am I available for work? What was accepted? What needs my attention? The full design can display usage limits and financial history when those data are supported. A mock balance or capacity percentage must never appear as a live reading.

During the current phase, the public website explains the product and provides the whitepaper, roadmap, release status, and an explicitly labeled demonstration. A download becomes available only when a matching, qualified public installer is actually published. The private engineering archive is not a customer download.

3. Current implementation and evidence

The supplied project contains a private coordinator and owner workbench, separately scoped worker and protocol components, Windows/WSL preparation, Linux ARM64 preparation, and development contracts. Public distributions are assembled from an explicit allowlist. Commercial coordination remains proprietary; the licensing and contents of each public release must be stated in that release.

The 26 November source handoff records 36 passing checks selected for its final preparation changes, no failures or skips in that subset, and successful clean dependency installation, backend compilation, and owner-UI compilation on a Linux AMD64 review host. These are bounded source and preparation results. They do not establish current native Windows or Spark qualification, production payment readiness, or an independent security audit.

Dated project records also describe isolated local payment work and a bounded Fuji CPU-task settlement and withdrawal exercise. That history supports the existence of payment engineering beyond a mockup. It does not establish a currently operating payment service. The same requirement record leaves the normal customer payment runtime uncommissioned and production payments unqualified.

The public evidence standard is therefore explicit: **implemented** means source exists; **demonstrated** names its recorded environment and date; **live** requires a current observation of the exact deployed release. **Proposed** describes design work awaiting implementation or approval. An old pass is never silently promoted into a new live claim.

4. How work moves through the system

The coordinator admits supported work under a specific policy, selects an eligible worker, and binds the assignment to the input, expected output conditions, limits, and identities. A logical job, its immutable economic assignment, and each execution attempt remain separate. Transport retries and reconnects must not change the funded terms.

Workers receive scoped assignments and produce signed receipts that bind the attempt and result. The service tracks whether a result was submitted, verified, rejected, or held for review. Unknown effects remain held until they are reconciled. A missing capability, signer, dependency, or observation is a named blocker rather than permission for a substitute execution path.

Independent verification evaluates the evidence required by the task profile. For deterministic public inputs, exact recomputation may be appropriate. Other profiles need domain-specific acceptance criteria or bounded human review. The current verification source expressly distinguishes summary consistency and public-input recomputation from hardware attestation.

A signature proves that a particular key signed a statement; it does not by itself prove that useful computation occurred. Blockchain finality does not establish output correctness. The initial system still trusts approved coordinators, verifiers, and, where selected in advance, an arbitrator. AvaDepin should describe that trust model plainly rather than call it fully trustless.

The owner coding workflow retains separate planning, confirmation, Run, checking, and integration stages. A checked proposal becomes accepted work only through the registered acceptance and update path. Unattended goal generation and recurring autonomous improvement are outside the current beta boundary.

5. Security, privacy, and operational control

The design protects identities, authority, and evidence across restarts. Releases and runtime inputs are pinned; process cleanup is bounded; failed observations remain visible; and recovery reuses the original operation identity. Private owner controls are separated from public contributor functions.

Execution remains a substantial trust boundary. Ordinary executors can read task plaintext while processing it. The current project does not establish confidential computing, a general sandbox proof, or protection from a malicious physical host. Sensitive customer data should therefore be excluded from initial public task profiles unless the particular protection and consent model has been reviewed and demonstrated.

Before a public release, the installer, selected hosting release, signatures, served artifacts, and real contributor journey must agree. Required acceptance includes useful work, Stop behavior, reconnect, ordinary backup and restore, and an agreed observation period on fixed artifacts. Passing a source test or displaying a healthy process is insufficient.

Payment activation needs its own security gate: reviewed contract source and deployment configuration, asset assumptions, signer custody, role separation, replay and recovery behavior, and an independent assessment proportionate to the funds at risk. Contract pausing and administrative powers must be disclosed with the deployment. Existing contracts are development implementations and are not presented as audited.

6. Avalanche integration: C-Chain first

Avalanche's C-Chain supports EVM smart contracts, including Solidity. Official documentation distinguishes C-Chain mainnet, chain ID 43114, from Fuji testnet, chain ID 43113.[1] This makes a Fuji-first integration a concrete route for the existing development contracts and off-chain coordination code.

The planned public journey starts with one supported task profile, an explicitly selected settlement asset, immutable quoted terms, and a bounded dispute policy. On-chain records should commit to the required task and result identities while keeping customer inputs, private conversations, source code, and operational credentials off public infrastructure.

The development escrow already models one economic assignment per job. Terms fix the provider, submission signer, verifier, arbitrator, maximum fees, deadlines, and commitments. Result submission, verifier attestation, challenge, settlement, refund, and withdrawal are separate operations. Its design pauses new funding without closing existing recovery paths. These properties require fresh tests against the exact proposed deployment; they do not constitute a production deployment claim.

An Avalanche L1 remains a possible later architecture. Avalanche L1s can define their own validation and execution rules, but that also introduces a network to operate and secure.[2] AvaDepin would consider that step only after measured usage demonstrates a requirement the C-Chain does not meet. An L1 is not a prerequisite for the grant milestone or the first useful paid task.

7. Payment model and unit economics

The proposed customer-facing model uses a stable-value service quote. A reviewed supported ERC-20 asset would be selected explicitly; Circle lists native USDC on Avalanche C-Chain, making it one candidate for evaluation.[3] Support, availability, custody, recovery, and restrictions must be checked for the actual release. This paper does not activate USDC payments.

An illustrative starting quote allocates **85% to the provider, 5% to independent verification, and 10% to the protocol**. For a $10 service budget, those amounts are $8.50, $0.50, and $1.00. This is a proposed pricing example, not an existing tariff or earnings forecast. A real quote fixes amounts before authorization; different task profiles may require different verification costs. Chain fees and any other charges are disclosed separately.

Amounts are accounted for in integer asset units. The total funded service amount equals the provider, verifier, and protocol limits. A resolved outcome allocates no more than each agreed limit, and the remaining service funds are refunded. Failed or expired work follows the selected recovery policy; network fees already incurred are separate from the service refund.

Contributor compensation depends on accepted work, actual demand, the agreed price, operating costs, and availability. Connected time or high GPU utilization is not income. Public reporting should distinguish gross provider awards, settled credit, withdrawals, protocol revenue, and subsidies. Testnet tokens and demonstration points are not revenue.

The commercial test is positive contribution margin and repeated useful demand after verification, support, and infrastructure costs. Incentives may help an early pilot, but subsidized activity must be reported separately from customer-funded activity. Growth should not depend on issuing rewards that conceal a service with no paying users.

8. Optional token proposal

No AvaDepin token is launched or offered through this paper. The working symbol **AVDP** and a maximum design supply of **1,000,000,000 units** are a proposal for review. The name and symbol are not cleared. The MVP can operate with an existing settlement asset and AVAX network fees; a new token is not required to buy work or participate in the pilot.

A token should be considered only if it improves a demonstrated coordination problem, such as a transparent budget for earned ecosystem incentives. Proposed future participation or governance functions must specify exactly what they control. Token ownership would not automatically confer equity, profit distributions, ownership of customer funds, a guaranteed service discount, or authority over contributor computers.

The proposed allocation totals 100%. Every pool has zero automatic genesis unlock. Release envelopes are maximums, not promises that tokens will be issued or earned.

Proposed pool	Share	Units	Proposed release envelope
Accepted-work incentives	40%	400,000,000	Maximum budget spread over 96 months from an approved launch; awards require accepted, eligible work
Ecosystem builders	20%	200,000,000	Six-month cliff, then 48 months of linear budget release against published grants
Core team	15%	150,000,000	Twelve-month cliff, then 36 months of linear vesting
Protocol treasury	15%	150,000,000	Twelve-month cliff, then 48 months of linear release under disclosed custody
Community testing	5%	50,000,000	Six-month cliff, then 18 months of capped rewards for verified pilot contributions
Market-access reserve	5%	50,000,000	Locked until a separate funded and approved plan; then no faster than 24 months of linear release
Total	**100%**	**1,000,000,000**	**No automatic genesis allocation**

The accepted-work pool's eight-year envelope caps its annual budget at 5% of maximum supply. Actual monthly awards also depend on the verified work budget and abuse controls. Unearned capacity remains locked and does not automatically enlarge the following month's cap. Ecosystem and treasury envelopes do not make unapproved spending permissible. The market-access reserve commits neither liquidity capital nor an exchange listing.

Before any launch, the owner must approve a final specification, applicable legal and tax treatment must be assessed, contracts and custody must be reviewed, vesting addresses and powers must be disclosed, and customer-funded demand must justify the design. No sale, presale, staking yield, token price, burn schedule, or buyback promise is part of this version. Future economic changes require a dated specification and an explanation of their effect on users.

9. Metrics that show useful progress

The public dashboard should report a small set of clearly defined measures. **Eligible contributors** counts unique admitted contributor identities observed within the stated period, not wallet addresses or historical installations. **Accepted work** counts unique logical tasks accepted by the relevant verifier, with replayed attempts excluded. **Completion rate** uses a published eligible-task denominator and reports cancellations and failures separately.

Financial measures require finalized, reconciled records: customer-funded service volume, provider awards, protocol fees, refunds, and withdrawals. Displayed assets and environments must be explicit. Testnet activity, incentives, and synthetic demonstrations belong in separate views. A dated audit result may be useful evidence, but it is not a live network statistic.

Until an authorized public aggregate feed is connected and validated, live figures remain **Unavailable** with a plain explanation. The dashboard may include a clearly marked interactive example to teach the workflow, but it must not label those example values as observed activity. No private endpoint, wallet identity, task content, or machine address should be exposed to populate a marketing page.

10. Grant plan and acceptance milestones

Team1's current public program lists open Mini Grant applications up to $10,000; Accelerator Grants up to $30,000 are committee-proposed rather than an open application track.[4] The program's published overview emphasizes a prototype, documented product thinking, milestone development, and user feedback.[5] AvaDepin proposes a **$10,000 milestone budget** for review. No award, endorsement, or program acceptance is claimed.

Proposed milestone	Budget	Concrete acceptance evidence
Reproducible invited beta	$2,500	One matched release installed on the coordinator and two compatible invited contributors; Stop, reconnect, useful task and recovery observations recorded
Fuji settlement journey	$3,500	One successful funded task and one separately canceled/expired task; exact terms, verification, settlement/refund, withdrawal and balanced accounting; replay and lost-response cases reconciled
Small user pilot and public evidence	$2,500	Five consented pilot participants targeted; task success and onboarding feedback reported with actual denominators; sanitized aggregate feed and qualified release publication
Security review and next-stage decision	$1,500	Independent scoped review, risk triage, published fixes/holds and a documented go/no-go decision for a capped payment pilot
Total proposed	**$10,000**	**Release against agreed evidence, subject to grant terms**

These are proposed funding allocations, not guaranteed vendor quotes or a claim that $1,500 buys a comprehensive contract audit. A full audit or operational work beyond the funded scope requires a separate budget. The five-person pilot is an acquisition target, not current adoption.

The grant should produce reusable Avalanche integration knowledge: a documented task-to-settlement model, public contract interfaces where permitted, deterministic verification

examples, and a clear separation between testnet demonstrations and commercial activation. The proprietary coordinator can remain private while published components and evidence have explicit licenses and boundaries.

11. Roadmap, decisions, and risks

The immediate sequence is evidence-led: qualify the existing invited beta; repeat the complete application-driven Fuji journey; improve onboarding with real pilot feedback; publish truthful aggregate observations; then decide whether a tightly capped commercial pilot is justified. A broader task catalog, new token, custom Avalanche L1, and decentralized coordination are later decisions with separate acceptance criteria.

Technical risks include heterogeneous hardware, failures during reconnect or recovery, malicious or faulty workers, incorrect verification, and compromised signing or administrative authority. The coordinator and approved verifier set create concentration of trust. Contract bugs, unsupported token behavior, freezes, fee volatility, and observer disagreement can affect payment availability. No single metric or audit eliminates those risks.

Commercial risks include insufficient paying demand, support costs exceeding fees, incentives attracting activity without retention, and contributors whose operating costs exceed awards. Brand clearance, distribution terms, data handling, and payment obligations must be resolved for the intended audience and jurisdictions. These are launch decisions, not reasons to display unearned confidence on the website.

The project advances when a user can repeatedly get a useful result, a contributor can retain control of their computer, and the accounting explains exactly what happened. That is the standard for the next milestone.

Sources and evidence notes

External sources were checked on 26 November 2026. They describe the referenced platform or program, not AvaDepin endorsement or implementation status.

1. Avalanche Builder Hub, **Primary Network / C-Chain**. https://build.avax.network/docs/primary-network
2. Avalanche Builder Hub, **Avalanche L1s**. https://build.avax.network/docs/avalanche-l1s
3. Circle Developer Docs, **USDC contract addresses**. https://developers.circle.com/stablecoins/usdc-contract-addresses
4. Avalanche Team1, **Builder Grants**. https://www.team1.network/grants
5. Avalanche Team1, **Grants program overview**. https://grants.team1.network/
6. Avalanche Builder Hub, **Team1 Mini Grants**. https://build.avax.network/grants/team1-mini-grants

Project evidence: the dated 26 November 2026 source-handoff validation; the development contract documentation and implementations; the project requirement and release records. These were reviewed as supplied source evidence. Current live state was not independently queried for this paper. Private operational material remains in the engineering handoff and is excluded from the public site.

AvaDepin and AVDP are working names. This document presents a product and economic design, not an operating token offering, investment return, or funding award.