SDCN

Solana Decentralized Compute Network

Decentralized Windows Workspaces and Distributed Compute

CONCEPT WHITE PAPER / VERSION 0.1 / SEPTEMBER 2026

Core proposition

Enable anyone with compute hardware to become a Provider, and let users without nodes obtain Windows workspaces and external compute through tokenized leasing and usage settlement.

This document aligns the project vision, product boundaries, economic model, security assumptions, and evolution path. It is not an engineering implementation specification. All quotas, fees, stake amounts, and performance tiers are illustrative.

Contents

The white paper is organized from product boundaries through architecture, protocol economics, delivery, and open decisions.

Core judgment

The project direction is viable only with a layered execution model: a decentralized compute marketplace, a single-node Anchor VM for each Windows instance, and task-level distribution through a Remote Compute Fabric.

DECISION BRIEF / 0

Executive Summary

Project in one sentence

SDCN is a peer-to-peer cloud computing network in which Solana coordinates economic relationships and settlement while individual users and professional operators supply compute resources. Consumers receive an interactive Windows cloud desktop and can call CPU or GPU capacity from the wider network on demand.

Essential product boundaries

- The network is distributed, but a general-purpose Windows instance should be anchored to one primary Provider at any moment so its kernel, memory, and disk state remain coherent.
- Task-level workloads can be parallelized across nodes; arbitrary Windows instruction execution cannot be transparently split across the internet.
- Solana is the trusted economic state layer, not the system for high-frequency heartbeats, scheduling, or per-second resource metering.
- The incubation stage should prove a verifiable economic loop before pursuing full decentralization.

Decision table

Question	Recommended conclusion
What does Solana do?	On-chain registration, staking, lease records, escrow, settlement, slashing, and reputation anchoring - not real-time compute scheduling.
How does Windows run?	One Windows VM keeps CPU, RAM, and disk state on one Provider; the user connects through RDP, WebRTC, or another streaming layer.
What is genuinely distributed?	A Compute Agent inside Windows can send divisible AI inference, rendering, transcoding, compilation, and data jobs to other Providers.
How does stake affect service?	Consumer Stake sets the eligible performance ceiling, concurrency, priority, and discounts; actual resource use continues to consume tokens.
Why does a Provider stake?	Stake is slashable service collateral against false capacity claims, repeated outages, refusal to perform, and metering fraud.
What should be validated first?	Provider joins -> Consumer leases Windows VM -> usage is metered -> both sides sign a Usage Receipt -> settlement is recorded on-chain.

FOUNDATIONS / 1

Vision and Product Positioning

1.1 Vision

Organize globally distributed, idle, and heterogeneous personal computers, workstations, servers, and GPU nodes into an open compute network. Node owners earn tokens by contributing resources; consumers stake and pay for Windows workspaces and elastic remote compute.

1.2 The product is not Windows on Solana

Windows does not run on Solana validators. Solana is the economic consensus layer; the P2P network is the real-time control layer; KVM, virtualization, and future trusted execution environments form the execution layer; Windows is the user workspace layer.

1.3 Three primary product forms

Product form	User experience	Underlying resources
Windows Workspace	A continuously available Windows cloud desktop after sign-in	One Provider VM plus persistent storage
Elastic Compute	Temporary CPU or GPU expansion when more performance is needed	One or more remote task nodes selected by the network
Compute Marketplace	Choose capacity by price, region, GPU model, and reputation	Provider offers plus Lease and Usage Settlement

NETWORK ROLES / 2

Participants and Incentives

2.1 A Consumer and Provider may be the same party

A user with local hardware can sell idle resources to earn tokens and later buy network capacity when local resources are insufficient. This two-way participation model creates a natural supply-and-demand loop.

Role map

Role	Primary responsibilities	Primary benefit or cost
Consumer	Stake tokens, create Windows workspaces, submit distributed jobs, and confirm metering	Receives compute resources and pays by usage
Provider	Contribute CPU, RAM, storage, and GPU; run isolated VMs; sign usage records	Earns lease and compute revenue while bearing power, bandwidth, and depreciation costs
Scheduler	Discover nodes, match resources, route jobs, and maintain sessions	May be officially operated at first and later compete in a multi-Scheduler market
Verifier	Run challenges, sample performance, validate disputes, and recompute results	Receives verification rewards or protocol subsidies
Protocol Treasury	Collect a small protocol fee and fund ecosystem incentives and dispute handling	Supports long-term network operation

FOUR-LAYER MODEL / 3

System Architecture

The recommended architecture separates economic consensus, real-time control, execution, and end-user experience. Each layer has a distinct trust and latency profile.

LAYER	SYSTEM LAYER	RESPONSIBILITIES
Layer 4	Experience and Workspace	Windows Workspace, web console, RDP or WebRTC, Compute SDK
Layer 3	Execution	KVM or QEMU, MicroVM, GPU runtime, persistent volume, future TEE
Layer 2	P2P Control Plane	Node discovery, Scheduler, heartbeat, metering, task routing, benchmark
Layer 1	Solana Economic Layer	Registry, Stake, Lease, Escrow, Settlement, Slashing, reputation anchor

3.1 Why the entire system cannot be on-chain

Node heartbeats, millisecond scheduling, and per-second CPU or GPU metering are high-frequency real-time data. Writing each event to Solana would increase cost, latency, and complexity, and on-chain consensus still cannot prove how much work an external CPU actually performed. Usage should be aggregated and signed off-chain; only lease state and final settlement belong on-chain.

ANCHOR VM PLUS COMPUTE FABRIC / 4

Windows Execution Model

4.1 Anchor VM

Each Windows Workspace is assigned to one primary Provider. That Provider supplies virtual CPU, RAM, the system disk, the user's persistent disk, and interactive streaming, preserving operating-system and application state coherently.

4.2 Remote Compute Fabric

Windows includes a Compute Agent. When an application or SDK starts a divisible job, the Agent submits the job description, input data, and resource requirements to the P2P Scheduler. One or more Providers execute the job independently and return results.

Responsibility boundary

Resource	Handled by Anchor VM	Suitable for transparent cross-node split
Windows kernel and process state	Yes	No
RAM	Yes	No
System and user disks	Yes	Not recommended in real time
Desktop display and input	Yes	No
AI inference jobs	Optional	Yes
Render frames and video segments	Optional	Yes
Batch compilation and data processing	Optional	Yes

KEY DISTINCTION

The user sees one Windows machine. Internally, the system combines one Anchor VM with an optional remote compute pool. This preserves the cloud desktop experience while enabling genuinely distributed computation.

CAPABILITY DESCRIPTION / 5

Provider Resource Model

5.1 Provider Resource Manifest

When a Provider Agent starts, it creates and signs a Resource Manifest. The manifest should describe both raw capacity and the policy conditions under which that capacity can be sold.

Category	Example fields
Identity	Provider wallet, Node ID, Agent version

Category	Example fields
CPU	Architecture, logical cores, sellable cores, benchmark score
RAM	Total capacity, sellable capacity, bandwidth tier
GPU	Model, VRAM, driver, sellable share, benchmark score
Storage	Medium, capacity, IOPS or throughput, persistence capability
Network	Public reachability, upstream and downstream bandwidth, region, NAT type
Virtualization	KVM, VT-x or AMD-V, nested virtualization, TEE capability
Policy	Allowed workloads, ports, duration, and content policy

5.2 Compute resources are not fully homogeneous goods

CPU and GPU models, memory bandwidth, storage performance, and network quality vary widely. The protocol should standardize metering dimensions without forcing all hardware into one fixed price. Providers quote offers and market supply and demand determine the clearing price.

ACCESS AND METERING / 6

Consumer Staking and Usage Billing

RECOMMENDED ECONOMIC PRINCIPLE

Stake unlocks a higher performance ceiling, more concurrency, scheduling priority, and discount eligibility. Usage Payment is the resource fee paid to Providers. A model in which Consumers only lock tokens while Providers bear continuous real-world costs would become economically unbalanced.

6.1 Stake determines eligibility and limits, not free compute

Illustrative tier	Consumer Stake	Maximum Workspace	Remote job concurrency	Positioning
S0	0	2 vCPU / 4 GB	1	Entry experience
S1	1,000 TOKEN	4 vCPU / 8 GB	2	Standard user
S2	10,000 TOKEN	8 vCPU / 16 GB	5	Advanced user
S3	50,000 TOKEN	16 vCPU / 32 GB	10	Professional or team

PARAMETER NOTICE

These figures illustrate the model only and are not final tokenomics parameters.

6.2 Network Resource Unit

The protocol can define an internal billing abstraction called the Network Resource Unit, or NRU. It records usage across different resource classes, while final prices remain market-driven. NRU must not be confused with Solana Compute Units.

Metering dimension	Suggested base unit	Continuously billed
CPU	vCPU-second	Yes
Memory	GB-second	Yes
GPU	GPU-second or profile-second	Yes
Persistent storage	GB-hour	Yes
Temporary storage	GB-hour	Yes
Network egress	GB	Policy-dependent
Public IPv4	IP-hour	Optional

TRUSTED ECONOMIC STATE / 7

Solana Protocol Design

7.1 On-chain state objects

Object	Purpose	Core field examples
ProviderAccount	Node registry and reputation anchor	owner, stake, status, manifestHash, reputation
ConsumerAccount	User stake and entitlement	owner, stakeTier, creditLimit
Lease	Windows or task rental relationship	consumer, provider, resourceProfile, start, maxSpend
Escrow	Prepaid fee custody	lease, balance, settledAmount
UsageCommitment	Periodic usage commitment	lease, epoch, usageHash, amount, signatures
Dispute	Dispute state	lease, evidenceRoot, challenger, status

7.2 Recommended state machine

A simple Lease state machine is sufficient: Requested -> Matched -> Active -> Closing -> Settled. Exception paths include Expired, ProviderFault, ConsumerFault, and Disputed. High-frequency measurements stay off-chain; the chain receives only usage summaries and monetary amounts.

7.3 Responsibility boundary

- Must be on-chain: stake changes, Lease creation and closure, escrow balance, settlement outcome, slashing, and material reputation events.
- Should stay off-chain: per-second CPU utilization, node heartbeats, live network latency, Scheduler candidate lists, and complete logs.

SCHEDULING AND COORDINATION / 8

P2P Control Plane

8.1 Core control-plane modules

Module	Responsibility
Discovery	Find online Providers and synchronize resource summaries and network addresses
Scheduler	Match nodes by price, region, performance, reputation, stake, and latency
Session Manager	Maintain off-chain sessions for Windows and task Leases
Metering	Collect Host and Guest resource consumption and produce Usage Receipts
Benchmark	Run periodic performance baselines and sample the authenticity of resources
Reputation Relay	Aggregate uptime, refusal rate, and dispute outcomes before anchoring summaries on-chain

8.2 Scheduling does not need to be fully decentralized at first

During incubation, one official Scheduler can coordinate supply and demand while critical lease relationships and funds remain controlled by the Solana Program. The system can later evolve toward multiple Schedulers, a P2P DHT, or open bidding. This sharply reduces early implementation risk.

PROGRESSIVE USAGE ASSURANCE / 9

Proof of Compute

Starting a virtual machine is not the hardest problem. The difficult question is how to show that a Provider delivered the claimed resources and that the Consumer actually used them. Trust should improve in stages.

Phase	Mechanism	Assurance	Appropriate stage
P0	Usage Receipt signed by Provider Host Meter and Guest Agent	Medium	MVP
P1	Random benchmark or challenge plus uptime proof	Medium-high	Public beta
P2	TEE, TPM, or remote attestation	High	High-value nodes
P3	Multi-node recomputation or arbitration for deterministic jobs	High, with high cost	Disputes or high-value jobs

9.1 Usage Receipt logic

Every one to five minutes, the system creates a Usage Epoch. The Host Meter measures resource consumption and the Windows Guest Agent observes key metrics independently. Both sides sign the same usageHash and amount. Matching records can settle automatically; mismatches enter the dispute process.

Illustrative receipt

Field	Example
leaseId	912881
epoch	38191
cpuSeconds	1687
memoryGbSeconds	4672
gpuProfileSeconds	93
egressMb	203
cost	0.01892 TOKEN
signatures	providerSig + guestSig

Provider Stake and Reputation

10.1 The role of Provider Stake

Provider Stake is service-quality collateral. The amount of capacity a Provider may sell, the maximum value of Leases it may accept, and the speed at which its reputation recovers can all be linked to stake.

Reputation and enforcement events

Event	Recommended response
Short outage	Stop billing and reduce the uptime score
Refuses to start after accepting an order	Refund the Consumer, reduce reputation, and slash after repeated offenses
Falsifies GPU or CPU capacity	Slash and suspend the node after a failed challenge
Metering anomaly	Freeze the related settlement and open a Dispute
Malicious Guest tampering	Slash according to attestation or arbitration outcome
Sustained high-quality performance	Increase reputation weight and exposure to higher-value orders

Security and Abuse Isolation

DEFAULT THREAT MODEL

A Provider must assume the Consumer may be malicious, and a Consumer must assume the Provider may inspect data, throttle service, falsify capacity, or tamper with software. Security cannot depend on either party behaving voluntarily.

11.1 Protect the Provider

- Run user code only inside strongly isolated KVM, QEMU, MicroVM, or equivalent environments; never grant direct Host access.
- Restrict device passthrough, system calls, disk mounts, network privileges, and management interfaces.
- Let Providers publish Policy Profiles covering public inbound traffic, SMTP, bulk scanning, GPU passthrough, and other sensitive capabilities.
- Detect anomalous traffic, abusive resource behavior, and malicious workloads at the platform layer.

11.2 Protect the Consumer

- Encrypt persistent user volumes by default and destroy keys when an ephemeral VM is closed.
- Introduce confidential-computing capabilities such as AMD SEV-SNP or Intel TDX for high-security nodes in later phases.
- Use the Windows Guest Agent to verify the boot image, Agent version, and metering-module integrity.
- Allow sensitive jobs to select only Providers with hardware attestation.

Windows Workspace Lifecycle

12.1 Two instance modes

Mode	Characteristics	Best fit
Ephemeral Workspace	Fast start; destroyed at shutdown; data stored on an independent encrypted Volume	Temporary office work, testing, short jobs
Persistent Workspace	Stable system and user disks; resumes across sessions; may require migration	Long-term personal cloud desktop

12.2 User flow

01 Sign in with a wallet

02 Check the Consumer Stake tier

03 Choose Windows performance and budget

04 Lock funds in Escrow

05 Match with a Provider through the Scheduler

06 Start the VM on the Provider

07 Return encrypted session credentials

08 Connect through RDP or WebRTC

09 Produce periodic Usage Receipts

12.3 Failure and migration

The incubation release does not need seamless live migration. The first version can stop billing immediately when a Provider goes offline and let the Consumer restart the Workspace on another Provider. With an encrypted persistent Volume, the Workspace resumes from the latest synchronized state. Snapshot replication, incremental synchronization, and live migration can follow later.

WORKLOADS THAT DIVIDE CLEANLY / 13

Distributed Job Model

13.1 First workloads to support

Workload	Partitioning method	Result verification difficulty
AI inference	Request or batch distribution	Medium
Blender or 3D rendering	Frame or tile	Low-medium
Video transcoding	Time slice or segment	Low
Batch image processing	File	Low
Code compilation	Compilation unit or remote cache	Medium
Data processing	Map or shard	Medium

13.2 Workloads that are not an early priority

Workloads that require strongly shared memory, hard real-time synchronization, or extremely low-latency interconnects are poor fits for the first phase. General Windows applications also cannot transparently migrate arbitrary threads to remote nodes.

ECONOMIC FRAMEWORK / 14

Tokenomics and Protocol Revenue

14.1 Three sources of token demand

Demand source	Function
Consumer Stake	Unlock performance tiers, concurrency, priority, discounts, and credit limits
Provider Stake	Back service quality, set capacity limits and reputation weight, and fund slashing collateral
Usage Payment	Pay the variable cost of CPU, RAM, GPU, storage, and network usage

14.2 Illustrative fee allocation

Destination	Illustrative share	Rationale
Provider	94%	Compute, bandwidth, storage, electricity, and hardware return
Protocol Treasury	4%	Protocol development, operations, and security
Verifier and incentives	2%	Challenges, verification, and early-stage incentives

PARAMETER NOTICE

The allocation is conceptual. Production economics must be rebuilt around Provider margin, token volatility, on-chain fees, refunds, and dispute costs.

14.3 Price formation

The protocol should not impose one hardware price. Providers quote by Resource Profile, and the Scheduler matches offers against the Consumer's maximum budget, region, latency, reputation, and performance needs. A protocol reference price or fixed packages may reduce user-facing volatility, with the protocol assuming part of the matching risk.

END-TO-END FLOWS / 15

Three Core Business Loops

15.1 Provider joins the network

01 Create wallet

02 Install Provider Agent

03 Run hardware checks and benchmark

04 Select sellable resources

05 Post Provider Stake

06 Register ProviderAccount

07 Publish heartbeat

08 Accept Lease

09 Run VM or task

10 Receive settlement

15.2 Consumer starts Windows

01 Wallet sign-in

02 Meet target stake tier

03 Fund or authorize Usage Budget

04 Create Lease Request

05 Scheduler match

06 Lock Escrow

07 Provider starts Windows

08 Consumer connects

09 Meter usage

10 Settle periodically

11 Shut down or renew

15.3 Windows calls network compute

01 Windows application or SDK submits a Remote Job

02 Compute Agent describes requirements

03 Scheduler selects one or more Providers

04 Encrypt and transfer input data

05 Execute

06 Verify results

07 Return results to Windows

08 Settle usage

SEQUENCE RISK OUT OF THE BUILD / 16

Incubation MVP and Roadmap

MVP PRINCIPLE

The first release should prove that unfamiliar nodes can complete a measurable Windows compute lease without either party taking custody of the other's funds. It should not also attempt cross-node Windows execution, complex DAO governance, fully automated disputes, and confidential computing.

16.1 Minimum M1 technical scope

- Solana Program: Provider Registry, Stake, Lease, Escrow, and Settlement.
- Provider Agent: hardware reporting, KVM VM creation, metering, signatures, and session management.
- Windows Image: preinstalled Guest Agent, dual-signed Usage Receipts, and remote access.
- Scheduler: centralized in the first stage, responsible for node selection and Lease sessions.
- Web Console: wallet sign-in, plan selection, balance and usage, and Workspace start or stop.

Evolution path

Phase	Scope	Question that must be answered
M0 Simulation	Two or three local nodes plus a Devnet contract model	Are Lease, Stake, and Usage data structures workable?
M1 Windows Lease	Linux Provider, KVM Windows, remote desktop	Can users obtain a stable VM and does the billing loop close?
M2 P2P Provider	Open third-party nodes, reputation, benchmark, and refunds	Can a heterogeneous network schedule and recover reliably?
M3 Remote Jobs	GPU, rendering, and transcoding Job API	Is there real demand for task-level distributed compute?
M4 Trust Upgrade	TEE, Challenge, Dispute, and multiple Schedulers	How can bilateral distrust and centralized scheduling risk be reduced?

KNOWN CONSTRAINTS / 17

Core Risk Matrix

Risk	Severity	Early response
Provider inspects user data	High	Disk encryption, restrict sensitive jobs to approved nodes, introduce TEE later
Consumer abuses a node	High	Strong VM isolation, network policy, risk controls, and deny lists
Metering fraud	High	Host and Guest dual signatures, challenges, and disputed-fund freeze
Frequent node outages	Medium-high	Uptime reputation, immediate billing stop, and instance rebuild
False GPU or CPU claims	Medium-high	Benchmarks, random challenges, and reputation decay
Token price volatility	High	USD reference pricing, short quote windows, and stablecoin settlement option
Windows licensing and image distribution	High	Concept validation only during incubation; dedicated compliance plan before commercialization
Centralized Scheduler	Medium	Accept temporarily, keep critical funds on-chain, and add multiple Schedulers later
Runaway complexity	High	Strictly constrain the MVP to the Windows Lease loop

EVIDENCE BEFORE SCALE / 18

Incubation Metrics

18.1 Supply side

- Online Provider count and sellable vCPU, RAM, and GPU capacity.
- Seven-day Provider uptime, order acceptance rate, and average VM startup time.

- Actual resource utilization and Provider return per hardware unit.

18.2 Demand side

- Workspace creation success rate, average session length, repeat purchase rate, and renewal rate.
- Average Usage Spend per user and the distribution of Consumer Stake tiers.
- Remote-job adoption: the share of Windows users who actually use the Compute Fabric.

18.3 Protocol side

- Usage Receipt agreement rate, Dispute rate, refund rate, and slashing frequency.
- P50 and P95 time from clicking Start to a usable desktop.
- On-chain settlement cost and off-chain metering cost per settlement.

RECOMMENDED CURRENT DIRECTION / 19

Decisions Still Under Incubation

Decision	Current recommendation
Must the payment asset be a native token?	Do not lock the design yet. Model token utility while reserving stablecoin support in the payment layer.
Must Windows persistence exist on day one?	Start with Ephemeral Workspace plus an independent Volume, then add full persistence.
Must the Scheduler be decentralized?	No. Early centralization helps validate supply, demand, and metering.
Are home nodes behind NAT allowed?	Yes, but restrict eligible jobs according to public reachability and network quality.
Is GPU passthrough required?	It can be omitted from M1 and introduced in M2 or M3.
Should slashing be automated immediately?	No. Record evidence and use human arbitration first, then automate gradually.
Will SDCN implement instruction-level distributed Windows?	No. The product is Workspace plus task-level Compute Fabric.

CONCLUSION / 20

Recommended Product Definition

SDCN is a decentralized cloud computing protocol in which Solana manages economic relationships and an open Provider network supplies real compute resources. Its first product is Windows Workspace. Its long-term capability is a distributed Compute Fabric callable by Windows, SDKs, and third-party applications.

The definition preserves four core values

- Users can contribute idle capacity from their own nodes and earn revenue.

- Users without nodes can pay for a complete Windows virtual system.
- Consumer Stake determines performance tiers and network entitlements, while actual consumption is settled with usage payments.
- Solana provides trusted economic state for identity, staking, leases, escrow, usage settlement, and penalties.

The Anchor VM plus Remote Compute Fabric design avoids relying on impractical instruction-level Windows execution across arbitrary internet nodes. It preserves the decentralized compute-market thesis and leaves room for GPU, AI, rendering, transcoding, compilation, confidential computing, and multiple Schedulers.

APPENDIX A

Concept Glossary

Term	Definition
Anchor VM	The primary virtual machine that holds one user's Windows kernel, RAM, and system state.
Compute Fabric	A network of Providers that executes divisible remote jobs.
Provider Agent	A daemon on a compute node that handles registration, virtualization, metering, and signing.
Guest Agent	An in-Windows agent for session handling, metering cross-checks, and Remote Jobs.
Lease	A rental relationship between Consumer and Provider under defined resource, price, and budget constraints.
Usage Receipt	A signed summary of resource use and charges for one metering period.
NRU	Network Resource Unit, a provisional internal abstraction for resource metering.
Attestation	Evidence from hardware or a security module that a node, VM, or Agent is in an expected state.

APPENDIX B

Explicitly Out of Scope

- Final token supply, release schedule, governance weight, or exchange-listing mechanics.
- A commercial Windows licensing plan.
- A commitment to decentralized scheduling, cross-Provider live migration, or fully trustless metering.
- A final virtualization stack or P2P protocol implementation.
- Transparent instruction-level parallelism for general Windows applications.
- Treating incubation-stage parameters as production security or economic parameters.

END OF CONCEPT DOCUMENT