Ocelloids Network Specification

The Zone Council, SO/DA

January 10, 2024

This document is a **work in progress**.

Abstract

This document outlines the functional specifications of the Ocelloids network, a system for secure and transparent leasing of software agents that observe, correlate, and respond to blockchain activity. The network comprises provider nodes hosting agents under negotiated leases with consumers. A continuous service attestation process ensures reliable agent operation, while the network registry controls the admission and revocation processes for service providers and auditors within a trust zone. The document details network roles, leasing processes, and attestation methods, providing a comprehensive understanding of the Ocelloids network's functionalities.

1 Introduction

The Ocelloids network operates as a real-time reactive layer with software agents that observe, correlate, and respond to blockchain activity. These program instances, hosted by provider nodes in a leasing model, naturally connect offchain data with onchain activity. Providers undergo an admission process within a trust zone and operate within defined geographical boundaries, subject to applicable laws. The Service Attestation process ensures the reliable functioning of the agents, enabling providers to earn leasing fees for hosting them.

2 Network

An Ocelloids network is a collection of nodes under a common trust zone with vetted access for participation.

The network is governed through an onchain registry, known as the Network Registry, containing participant information and their Network Roles.

Network participants have well-known identities tied to specific organizations, geolocations, and publicly accessible endpoints.

2.1 Network Registry

The network registry defines a trust zone characterized by designated sovereign and administrative accounts. Each of these accounts is endowed with privileges related to the admission and revocation of service providers and auditors. These procedures play a critical role as they establish connections with entities subject to liabilities and potentially involve the establishment of legally binding contractual agreements.

Each participant of the network has an associated Party Record, which includes essential information for locating service providers and verifying the authenticity of digital signatures.

Table 1: Party Record

Property	Description
Subject	Distinguished name[1].
Accounts	Operator and treasurer blockchain account addresses.
Endpoints	URLs[2] of the public endpoints.

Property	Description
Location	Geographic point location[3].
Role	The node role (section 2.2).

2.2 Network Roles

The network encompasses two node roles:

Provider Node
: Negotiates service agreements, hosts agents in accordance with active leases and receives leasing fees.

Auditor Node
: Verifies the execution of the agents involved in active leases and provides attestations to authorize the payment of leasing fees.

2.3 Program Catalog

The Ocelloids network provides a curated catalog of programs suitable to be hosted. These programs are organized into packages identified by content-addressable identifiers. Each package contains source code and/or binary files, along with a manifest describing the program and its necessary hosting capabilities. An instance of a program is referred to as an agent and possesses a unique identifier.

Reference Ocelloids Node spec. when written.

3 Service Leasing

The service leasing process involves consumers requesting service offers, depositing funds to pay for agent hosting, undergoing continuous service attestations, managing periodic payment claims, and facilitating automatic lease renewals.

3.1 Service Agreement

The service agreement process mandates that the consumer deposit funds to cover at least one period before the provider provisions the agent. The high-level steps are as follows:

1. *Request Quote.* The consumer initiates the leasing process by submitting a quote request to the provider, specifying the desired Program Content ID for execution.

2. *Service Offer.* The provider responds with a service offer, providing details such as the Program Content ID to be executed, the leasing period duration in number of blocks, leasing fee, and minimum deposit required.

3. *Place Deposit.* The consumer submits a deposit to the blockchain, specifying the offer and the transfer amount. The funds for one leasing period are locked, with any remaining funds available for withdrawal by the consumer at any time.

4. *Confirm Deposit.* The blockchain issues a deposit receipt for the offer to the consumer, confirming the deposit. The consumer then sends this receipt to the provider.

5. *Provision Agent.* The provider verifies the deposit and provisions the agent based on the accepted offer.

6. *Confirm Lease.* The provider submits the deposit receipt to the blockchain to formalize the lease, receiving a lease receipt in response.

7. *Activate Lease.* The provider acknowledges the lease activation to the consumer upon receiving the lease receipt.

3.2 Service Attestation

The service attestation process[1] involves auditors continuously verifying the accurate operation and fulfillment of the agents hosted by a providers under the leasing duration. The attestation process operates within the timeframe of a leasing period. During the period, the provider commits a verifiable proof of the processing of each block. Since the blocks could be processed out of order, the provider maintains a local verifiable key-value map independent of the insertion order, such as a sparse Merkle tree[4]. The commitment to the map adds a pair (k, v), where $k = Block_{hash}$ and $v = \text{digest}(Program_{output})$. The provider must anchor the top hash of the verifiable map at the end of the period. The attestation process for each period works as follows:

[1]For non-deterministic sources affecting the program output, a snapshot mechanism must be provided for reproducibility.

1. *Request Service Proofs.* The auditor requests service proofs for a random sample[2] of block hashes within the most recent anchored period. $B_{samp} = \{Block^0_{hash}, \ldots, Block^n_{hash}\}$.

2. *Present Service Proofs.* The provider presents the requested inclusion proofs for the given block hashes. $P_{samp} = \{Proof(b) : b \in B_{samp}\}$, where $Proof$ produces an inclusion proof for the commited value v on the key b.

3. *Verify Service Proofs.* The auditor verifies the inclusion proofs P_{samp}.

   (a) Confirms the inclusion proof using the anchored top hash for the period.

   (b) Independently processes the selected blocks to verify that the digest of the resulting log output matches the value v of the requested block in the proof.

4. (i) **On successful verification**
   Record Attestation. The auditor submits a signed attestation of the verified period. The attestation authorizes[3] the payment of leasing fees[4] by the provider.

   (ii) **Otherwise**
   Record Dispute. The auditor submits a signed dispute with details for further resolution[5].

This continuous attestation process ensures the maintenance of verified operational records, serving as a prerequisite for claiming leasing fees.

3.3 Claim & Renewal

At the end of each leasing period, the provider must claim the leasing fees and an automatic renewal process is initiated.

1. *Claim Fees.* Provider claims the payment of the fees for the leasing period by submitting a transaction to the blockchain; e.g., *ClaimPaymentForPeriod*.

[2] A simple approach would be to use Yamane's method ($n = \frac{N}{1+Ne^2}$) for N blocks in the period, where $n \approx 400$ for a 1-month period.

[3] Variations could require signatures from multiple auditors.

[4] The authorized payment should be captured by the provider and could entail the deducation of a managment fee accrued to the auditor/s.

[5] While operating in a vetted governance model (i.e., without funds at stake), the resolution falls under the discretion of the trust zone sovereign entities, who are responsible for revoking the misbehaving party and transferring funds to the rightful party.

2. *Verify & Transfer.* Blockchain verifies the latest agent operational attestation and transfers the funds to the provider.

3. *Renewal Check.* Provider checks renewal conditions:

   (a) The lease is active.

   (b) Enough funds for next period in the consumer account.

4. (i) **On successful check**
   Renew Lease. Blockchain initiates the renewal process, locking funds for the next period.

   (ii) **Otherwise**
   Cancel Lease. Blockchain cancels the lease, and the provider takes appropriate actions to decommission the agent.

This process ensures a smooth transition between leasing periods, with automatic renewals and the flexibility to cancel if necessary. Consumers are responsible for maintaining sufficient deposit funds to cover renewals.

Glossary

agent
: A program instance running on a provider. 2–4, 6, 7

auditor
: Node responsible for verifying the execution of agents involved in active leases and providing attestations to authorize the payment of leasing fees.. 3–5

consumer
: Party that leases the hosting of an agent on a provider within the network. 3, 4, 6

program
: Package containing executable bytecode or source code published through vetted catalogues. 2, 3

provider
: Node responsible for negotiating service agreements, hosting agents according to active leases, and receiving leasing fees.. 2–7

References

[1] The directory: Selected attribute types. X.520, October 2019. URL https://handle.itu.int/11.1002/1000/14037.

[2] Tim Berners-Lee, Roy T. Fielding, and Larry M Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, January 2005. URL https://www.rfc-editor.org/info/rfc3986.

[3] Standard representation of geographic point location by coordinates. ISO 6709:2022, September 2022. URL https://www.iso.org/standard/75147.html.

[4] Rasmus Dahlberg, Tobias Pulls, and Roel Peeters. Efficient sparse merkle trees: Caching strategies and secure (non-)membership proofs. Cryptology ePrint Archive, Paper 2016/683, 2016. URL https://eprint.iacr.org/2016/683. https://eprint.iacr.org/2016/683.