

Servicialo Protocol

v0.9

An Open Standard for Agentic Service Orchestration

Version	0.9	Status	Release Candidate
Date	March 2026	License	Apache 2.0
Registry	servicialo.com/.well-known/agents.json	MCP Package	@servicialo/mcp-server
Reference impl.	Coordinalo (coordinalo.com)	Tools	35 MCP tools

ABSTRACT

Servicialo is an open protocol that enables AI agents to discover, negotiate, and fulfill real-world service interactions — across scheduling, billing, clinical management, and inter-organizational referrals — without human intermediation. Version 0.9 introduces three foundational advances: a 3-Tier Access Model that distinguishes public, authenticated, and restricted capabilities; a Delegated Agency Model that allows agents to act with verifiable, scoped mandates on behalf of humans and organizations; and A2A v0.3 interoperability for composition with any A2A-capable system. The protocol is implemented in production by Coordinalo, the reference implementation. As of v0.9, the tool catalog has grown to 35 MCP tools and the protocol addresses financial inclusion as a design constraint.

Table of Contents

1.	The Problem
2.	Protocol Overview
2.1	Protocol Version History
3.	Core Principles
4.	3-Tier Access Model
4.0	Tier 0 — Public (10 tools)
4.1	Tier 1 — Authenticated (15 tools)
4.2	Tier 2 — Delegated (10 tools)
5.	DNS Resolution System
5.1	Resolution Chain
5.2	6 Resolver Tools
5.3	The x-servicialo Extension
6.	Delegated Agency Model
6.1	ServiceMandate
6.2	Mandate Lifecycle
7.	A2A v0.3 Interoperability
7.1	Key Changes from v0.2
7.2	Booking Flow Example
8.	Genesis Agent Skills
9.	Financial Inclusion
10.	Reference Implementation: Coordinalo
11.	Ecosystem and Registry
12.	Roadmap
13.	Exceptions and Scope Boundaries
14.	Conclusion
App.	
A	Full Tool Catalog
App.	
B	Comparison with Prior Versions

1 The Problem

Service delivery — in healthcare, professional consulting, home services, education — is fragmented at the infrastructure layer. Each clinic, agency, and practice runs its own scheduling system. Each billing platform speaks its own language. Referrals between organizations rely on phone calls, WhatsApp messages, and manual re-entry.

The emergence of AI agents capable of autonomous action has exposed this fragmentation as a systemic failure. An agent that can draft an email, query a database, and call an API cannot book a physical therapy appointment, check a patient's ISAPRE reimbursement status, or route a referral to a specialist — because there is no standard protocol for agents to interact with service providers.

The gap is not AI capability. It is the absence of a service interoperability layer.

Existing protocols partially address adjacent problems:

Protocol	What it solves	What it misses
MCP (Anthropic)	Tool invocation from AI agents	Service-domain semantics, consent, billing
A2A (Google)	Agent-to-agent task delegation	Service discovery, financial flows, patient data
CalDAV / iCal	Calendar interoperability	Agent-native access, authentication, clinical context
HL7 / FHIR	Clinical data exchange	Scheduling, billing, agent interoperability

Servicialo fills the gap between AI agent infrastructure (MCP, A2A) and real-world service execution.

2 Protocol Overview

Servicialo defines:

- A **discovery layer** — how agents find service organizations and their capabilities
- A **tool catalog** — 35 standardized MCP tools that any compliant implementation exposes
- An **access model** — three tiers governing what agents can do without, with, and beyond authentication
- A **delegation model** — how humans grant agents scoped mandates to act on their behalf
- An **interoperability layer** — A2A v0.3 integration for multi-agent orchestration
- A **financial model** — how payments, reimbursements, and financial inclusion are handled

2.1 Protocol Version History

Version	Key Addition
0.1	Core booking intent schema
0.2	MCP tool catalog (20 tools), basic A2A
0.6	DNS Resolution, whitepaper published
0.8	ServiceMandate draft, financial flows
0.9 ★	3-Tier Access, Delegated Agency, A2A v0.3, 35 tools, financial inclusion

3 Core Principles

1 Interoperability by Default

Any compliant agent should interact with any compliant service organization without bilateral integration agreements. The protocol is the contract.

2 Human Authority, Agent Execution

Agents act; humans authorize. No agentic action affecting health, finances, or legal status may proceed without a traceable human mandate. The Delegated Agency Model (§6) formalizes this principle.

3 Progressive Disclosure

Public capabilities are discoverable without authentication. Sensitive operations require progressively stronger authorization. The 3-Tier Access Model (§4) implements this principle.

4 Auditability

Every consequential action — booking, payment, clinical note, referral — must produce an auditable record. Tools are designed to emit structured audit events, not just return success/failure.

5 Financial Sovereignty

The protocol does not prescribe a payment provider. Organizations may accept bank transfers, card payments, FONASA vouchers, or any other mechanism. Financial flows are first-class protocol citizens.

6 Clinical Privacy as Default

When handling clinical data, the most restrictive privacy posture must be the default. Agents do not receive clinical information they have not been explicitly granted access to, regardless of authentication level.

7 Collective Intelligence as Common Good

Service interactions generate aggregate intelligence. Implementations that participate in the registry contribute anonymized aggregate signals to the commons. No individual patient data. The intelligence is collective; the privacy is absolute.

4 3-Tier Access Model

Prior versions of Servicialo treated authentication as binary. Version 0.9 formalizes a three-tier model that maps cleanly onto real-world authorization patterns.

Tier 0 — Public | No credentials required

10 public tools — any agent may invoke without credentials. Enables discovery, intent formation, and preliminary scheduling queries.

Tool	Purpose
resolve.lookup	Resolve org slug to endpoints and trust level
resolve.search	Search registered organizations by country and vertical
trust.get_score	Get organization trust score (0-100)
registry.search	Search orgs by vertical, location, country
registry.get_organization	Get public organization details
registry.manifest	Get server manifest and available endpoints
services.list	List public service catalog for an organization
scheduling.check_availability	Query available slots (3-variable scheduler)
a2a.get_agent_card	Get A2A Agent Card for inter-agent discovery
docs.quickstart	Complete getting-started guide as structured data

Tier 1 — Authenticated | Standard API Key or OAuth

25 authenticated tools — require client-level authentication. Execute bookings, manage client records, and handle financial operations.

Tool (subset)	Purpose
scheduling.book	Create a confirmed booking
scheduling.confirm	Confirm booked session
scheduling.reschedule	Move a booking to a new slot
scheduling.cancel	Cancel with reason and policy
lifecycle.get_state	Get current state + available transitions + history
lifecycle.transition	Execute state transition with evidence

Tool (subset)	Purpose
delivery.checkin / checkout	Record arrival/departure with GPS + timestamp
delivery.record_evidence	Register delivery proof by vertical
documentation.create	Generate formal service record
payments.create_sale / record_payment	Financial settlement tools
resource.list / get / create / update	Physical resource management
resolve.register / update_endpoint	Org registration and portability
clients.get_or_create	Resolve or create client identity
telemetry.heartbeat	Send heartbeat to resolver

Tier 2 — Delegated | ServiceMandate required

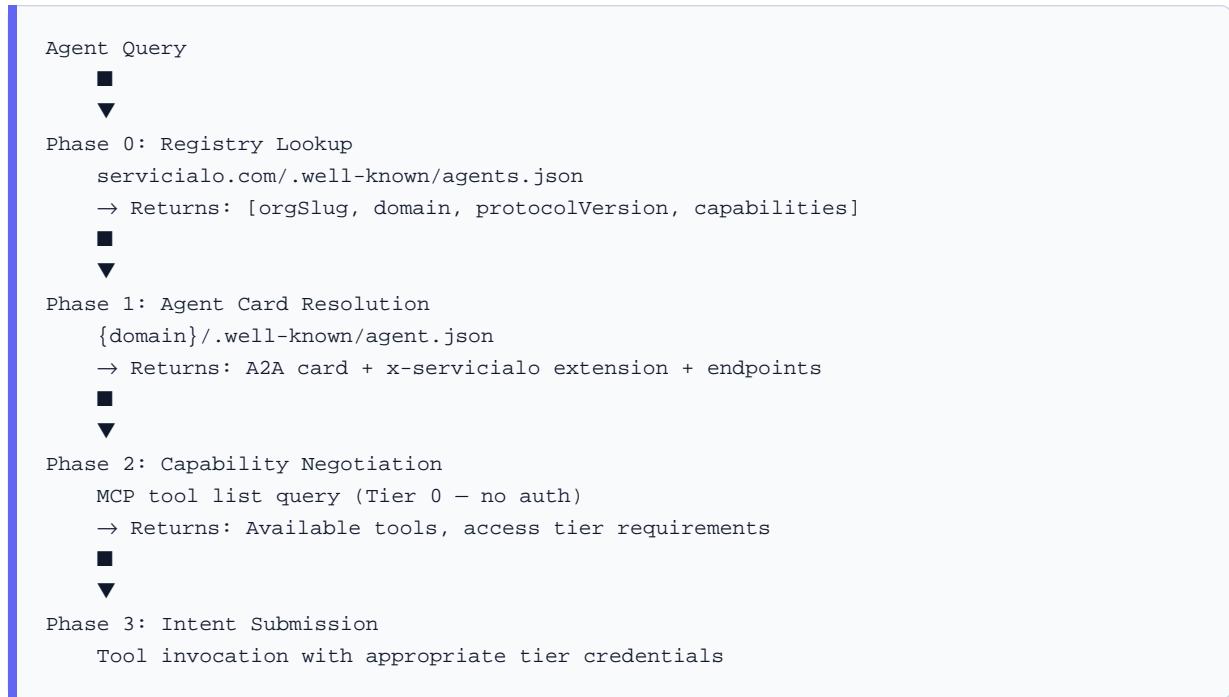
Tier 2 tools require a ServiceMandate object (§6) in addition to Tier 1 credentials. These execute on behalf of a human or organization with explicit, scoped, time-bounded authorization.

Tool	Purpose
servicialo_mandate_book	Book using delegated authority
servicialo_mandate_pay	Execute payment with mandate
servicialo_bulk_reschedule	Reschedule multiple sessions
servicialo_discharge_client	Close active treatment plan
servicialo_generate_prescription	Create prescription record
servicialo_export_clinical_record	Export patient record (PDF)
servicialo_create_consent_record	Record signed informed consent
servicialo_inter_org_transfer	Transfer client to another org
servicialo_audit_log	Query full audit log
servicialo_revoke_mandate	Revoke active ServiceMandate

5 DNS Resolution System

The Servicialo DNS Resolver provides a standardized discovery chain for agents to locate organizations, verify protocol compliance, and resolve endpoint topology — without prior knowledge or hardcoded URLs.

Resolution Chain



6 Resolver Tools

Tool	Phase	Purpose
resolve.lookup	0	Lookup org by slug, returns endpoints + trust level
resolve.search	0	Search registered orgs by country and vertical
trust.get_score	0	Get org trust score — passive accumulation from delivery history
registry.manifest	1	Get server manifest: capabilities, version, endpoints
registry.get_organization	1	Get org details with services, providers, booking config
resolve.register	2	Register new org in global resolver with unique portable slug

The x-servicialo Extension

Every Servicialo-compliant Agent Card must include the **x-servicialo** extension block. This gives any external agent — from Google ADK, Salesforce Agentforce, or any A2A-compatible system — everything it needs to initiate a service interaction in a single HTTP GET.

```
"x-servicialo": {
  "protocolVersion": "0.9",
  "implementer": {
    "name": "Coordinalo",
    "url": "https://coordinalo.com",
    "status": "verified",
    "type": "reference-implementation"
  },
  "endpoints": {
    "mcp": "https://servicialo.com/api/mcp",
    "rest": "https://servicialo.com/api/v1",
    "agentCard": "https://servicialo.com/.well-known/agent.json",
    "a2a": "https://servicialo.com/api/servicialo/{orgSlug}/a2a"
  },
  "registry": "https://servicialo.com/.well-known/agents.json"
}
```

6 Delegated Agency Model

The Delegated Agency Model is the mechanism by which AI agents gain authority to act on behalf of a human or organization for a bounded set of operations.

The Problem with Implicit Delegation

Prior to v0.9, Servicialo conflated authentication with authorization. A valid API key meant an agent could do anything the key permitted. There was no way to express: "this agent may book sessions for this patient, in this date range, for this service type, up to this cost."

ServiceMandate

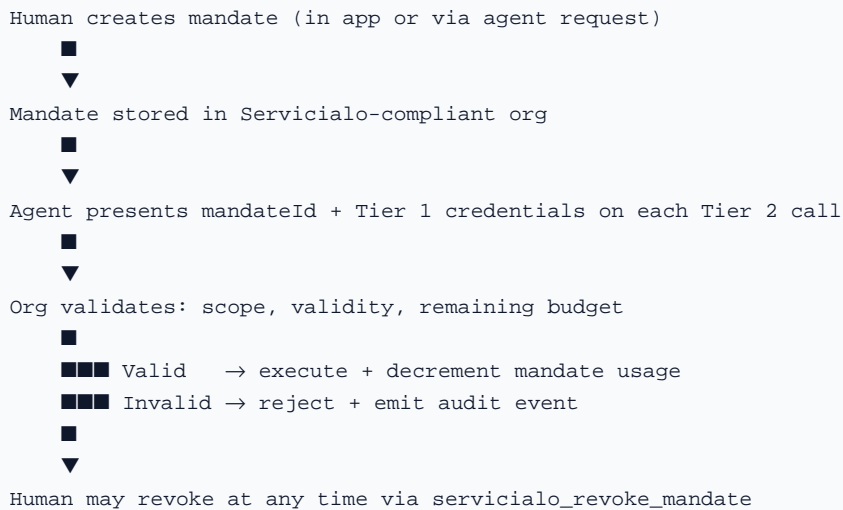
A **ServiceMandate** is a signed, structured authorization object:

```

{
  "mandateId": "mnd_01J...",
  "issuer": {
    "type": "human",
    "id": "client_xyz",
    "name": "María González"
  },
  "agent": {
    "id": "agent_abc",
    "name": "HealthBot Pro",
    "implementer": "acme-health-app"
  },
  "scope": {
    "tools": ["servicialo_mandate_book", "servicialo_reschedule"],
    "services": ["kinesiology"],
    "providers": ["any"],
    "maxCostPerSession": 50000,
    "currency": "CLP"
  },
  "validity": {
    "from": "2026-03-01T00:00:00Z",
    "until": "2026-06-01T00:00:00Z",
    "maxSessions": 12
  },
  "issuedAt": "2026-03-01T09:00:00Z",
  "signature": "sha256:..."
}

```

Mandate Lifecycle



The Delegated Agency Model makes Servicialo-based agents legally and operationally trustworthy. A clinic can allow a health management platform to autonomously book follow-up sessions — bounded by explicit patient authorization, visible in the audit log, revocable at any time.

7 A2A v0.3 Interoperability

Servicialo v0.9 adopts **A2A v0.3** as its multi-agent interoperability layer. A2A (Agent-to-Agent protocol, originated by Google, maintained by Linux Foundation) enables agents from different systems to exchange tasks, delegate subtasks, and share results through a standardized message format.

Key Changes from A2A v0.2

Feature	v0.2	v0.3
Task endpoint	tasks/send	message/send
Part type field	type	kind
Servicialo extension	Not specified	x-servicialo block required
Protocol version	Not declared	protocol_version in all messages

Booking Flow Example (A2A v0.3)

```

POST /api/a2a/message/send
{
  "protocol_version": "0.3",
  "message": {
    "role": "user",
    "parts": [
      {
        "kind": "text",
        "text": "Book a kinesiology session for María González next Tuesday afternoon"
      }
    ]
  },
  "context": {
    "orgSlug": "clinica-example",
    "mandateId": "mnd_01J..."
  }
}
    
```

Cross-System Agent Composition

```

Google ADK Agent
  ■ A2A v0.3 message/send
  ▼
Servicialo-compliant org (via x-servicialo agent card)
  ■ Internal MCP tool invocation
  ▼
Booking confirmed → A2A response with task result
  ■
  ▼
Google ADK receives structured confirmation
    
```

8 Genesis Agent Skills

Genesis defines the five core agentic capabilities that a Servicialo-compliant agent must demonstrate to be considered production-ready. These are not API features — they are emergent behaviors arising from combining the protocol's tool set.

1 Skill 1 — Autonomous Discovery

The agent locates a service organization, resolves its capabilities, and determines whether it can fulfill a need — without any hardcoded configuration.

Tools: resolve.lookup → registry.get_organization → services.list → scheduling.check_availability

2 Skill 2 — End-to-End Booking ✓ PRODUCTION VERIFIED

Completes a full booking cycle: availability query → slot selection → booking confirmation → payment recording — in a single autonomous session.

Tools: scheduling.check_availability → scheduling.book → scheduling.confirm → payments.create_sale

3 Skill 3 — Clinical Context Continuity

Reads a patient's history, active treatment plan, and previous session notes before booking — and includes clinical context in the booking record.

Tools: clients.get_or_create → lifecycle.get_state → documentation.create → scheduling.book

4 Skill 4 — Inter-Organizational Referral

Discovers a specialist organization, validates capacity, initiates a referral with consent token, and tracks the outcome — across two independent implementations.

Tools: registry.search → servicialo_initiate_referral → servicialo_get_referral_status

5 Skill 5 — Financial Reconciliation

Identifies unmatched incoming payments, resolves the most probable client mapping using historical patterns, and marks payments as matched or flags for human review.

Tools: payments.get_status → payments.record_payment → finance_reconciliation_engine

9 Financial Inclusion

Financial inclusion is not a feature. It is a design constraint.

Servicialo v0.9 formalizes this constraint with three protocol-level requirements for compliant implementations:

9.1 Multi-Modal Payment Support

Compliant implementations must support at least three payment modalities: card payment, bank transfer, and a publicly-subsidized option (FONASA in Chile, or equivalent national insurance in other jurisdictions). Agents must be able to query and record any of these without implementation-specific customization.

9.2 FONASA / Public Insurance Integration

The protocol defines a `codigoPrestacion` field on service definitions — the billing code required for FONASA reimbursement claims in Chile. This field is first-class in the service schema, not an extension. Similar systems in other countries (CUPS in Colombia, CBHPM in Brazil) are mapped to the same field.

9.3 Partial Payment and Balance Models

Real patients in real economic conditions make partial payments. Servicialo's financial model supports: outstanding balances carried across sessions, partial payment recording, payment plan definition via `TreatmentPlan.projectedSessions`, and dunning (payment recovery) workflows as protocol-native, not implementation-specific.

A scheduling protocol that only works for patients who can pay by card at booking serves a fraction of the population that needs healthcare. The protocol encodes the economic reality of the markets it operates in.

10 Reference Implementation: Coordinalo

Coordinalo is the **verified reference implementation** of the Servicialo protocol. Every protocol feature described in this document exists in production at coordinalo.com.

Implementation Status

Protocol Feature	Coordinalo Status
35 MCP tools (10 public + 25 authenticated)	Live at <code>/api/mcp</code>
3-Tier Access Model	API key + mandate system — Live

Protocol Feature	Coordinalo Status
ServiceMandate (Delegated Agency)	Implemented
A2A v0.3 interoperability	Live at /api/servicialo/{orgSlug}/a2a
x-servicialo agent card extension	Live at /.well-known/agent.json
DNS Resolution (6 resolver tools)	Live — servicialo.com/.well-known/agents.json
Genesis Skills 1–2	Production-verified
Genesis Skills 3–5	In progress
FONASA billing codes	In progress
Informed consent module	Planned
PDF clinical record export	Planned

Architecture

```

coordinalo.com
■■■■ /api/mcp → MCP server (35 tools, Streamable HTTP)
■■■■ /api/servicialo/{orgSlug}/a2a → A2A v0.3 message handler (JSON-RPC 2.0)
■■■■ /api/v1 → REST API (Tier 1 authenticated)
■■■■ /.well-known/
■ ■■■■ agents.json → Global A2A directory (all orgs)
■ ■■■■ mcp.json → MCP discovery config
■■■■ /api/servicialo/{orgSlug}/
    ■■■■ .well-known/
        ■■■■ agent.json → Per-org agent card + x-servicialo block
    
```

Mamá Pro (slug: mamapro) — a women's health kinesiology clinic in Santiago, Chile — serves as the canonical production reference for the Servicialo protocol. All Genesis skill validations are performed against this live deployment.

11 Ecosystem and Registry

The Servicialo Registry

The public registry at <https://servicialo.com/.well-known/agents.json> lists all verified Servicialo-compliant implementations. Registry listing requires:

- A valid /.well-known/agent.json with the x-servicialo extension
- Successful response to all 10 Tier 0 public tools
- Protocol version 0.8 or higher
- At least one Genesis Skill demonstrated in a staging environment

npm Package

```
npm install @servicialo/mcp-server
```

The `@servicialo/mcp-server` package provides a drop-in MCP server that any service organization can embed. It handles tool routing and validation, tier access enforcement, mandate validation, and audit event emission.

SDKs

Language	Package	Status
TypeScript	@servicialo/sdk	Available
Python	servicialo (PyPI)	Available

12 Roadmap

v1.0 — Q3 2026 — Stable Protocol

- Freeze tool catalog
- Formal compliance test suite
- Multi-country financial codes (Colombia, Brazil, Mexico)
- Signed mandate verification (cryptographic)
- Second verified reference implementation (non-Coordinalo)

v1.1 — Q4 2026

- Real-time agent presence (agents notify when actively managing a booking)
- Aggregate intelligence API (anonymized cross-org signals, Principle 7)
- Native FHIR export from clinical notes

v2.0 — 2027

- Fully autonomous treatment lifecycle management
- Cross-country referral flows
- Decentralized mandate registry

13 Exceptions and Scope Boundaries

Servicialo v0.9 explicitly does not define:

- **The user interface** — how humans interact with Servicialo is outside scope
- **The AI model** — any LLM or agent framework may implement the protocol
- **The payment processor** — Stripe, Fintoc, bank transfer, or any other mechanism is valid
- **Clinical decision support** — the protocol stores and routes clinical data but does not interpret it
- **Legal compliance** — implementations must comply with local health data regulations
- **The physical service** — Servicialo orchestrates the interaction but does not deliver the service

14 Conclusion

Servicialo v0.9 is the first open protocol that treats AI-native service orchestration — not as an extension of existing booking APIs, but as the primary design target.

The three advances in this release — the 3-Tier Access Model, the Delegated Agency Model, and A2A v0.3 interoperability — together make a substantive claim: that AI agents can be trusted with consequential service interactions when the protocol enforces human authority, scoped mandates, and complete audit trails.

The protocol is not theoretical. Coordinalo runs it in production. Mamá Pro books real patients with it. Genesis Skill 2 — the first fully autonomous end-to-end booking — has been executed in a live environment.

The question is no longer whether agentic service orchestration is possible. It is how quickly the ecosystem converges on an open standard.

Servicialo is that standard.

App .A Full Tool Catalog

35 tools across 3 tiers. Tools 1–10 are public (Tier 0). Tools 11–25 require authentication (Tier 1). Tools 26–35 require a ServiceMandate (Tier 2).

#	Tool	Tier	Domain
1	resolve.lookup	0	DNS
2	resolve.search	0	DNS
3	trust.get_score	0	Trust
4	registry.search	0	Discovery
5	registry.get_organization	0	Discovery
6	registry.manifest	0	Discovery
7	services.list	0	Discovery
8	scheduling.check_availability	0	Scheduling
9	a2a.get_agent_card	0	A2A
10	docs.quickstart	0	Docs
11	service.get	1	Services
12	contract.get	1	Services
13	clients.get_or_create	1	CRM
14	scheduling.book	1	Booking
15	scheduling.confirm	1	Booking
16	lifecycle.get_state	1	Lifecycle
17	lifecycle.transition	1	Lifecycle
18	scheduling.reschedule	1	Booking
19	scheduling.cancel	1	Booking
20	delivery.checkin	1	Delivery
21	delivery.checkout	1	Delivery
22	delivery.record_evidence	1	Delivery
23	documentation.create	1	Docs
24	payments.create_sale	1	Finance

#	Tool	Tier	Domain
25	payments.record_payment	1	Finance
26	payments.get_status	1	Finance
27	resource.list	1	Resources
28	resource.get	1	Resources
29	resource.create	1	Resources
30	resource.update	1	Resources
31	resource.delete	1	Resources
32	resource.get_availability	1	Resources
33	resolve.register	1	DNS
34	resolve.update_endpoint	1	DNS
35	telemetry.heartbeat	1	Telemetry

App .B Comparison with Prior Versions

Dimension	v0.6	v0.9
MCP tools	20	35 (10 public + 25 authenticated)
Core principles	6	7 (+ collective intelligence as common good)
Exception flows	7	6 (reclassified)
A2A version	0.2 (basic)	0.3 (x-servicialo extension required)
Delegation model	None	ServiceMandate (Tier 2)
Access tiers	Binary (auth/unauth)	Tier 0 / Tier 1 / Tier 2
DNS resolution	Draft	Phase 0 with 6 resolver tools
Financial inclusion	Not addressed	Protocol-level design constraint
Genesis skills	Not defined	5 skills — Skill 2 production-verified
Registry listing	Not required	Required for verified status
Agent card extension	None	x-servicialo block (implementer + endpoints)

Servicialo is an open protocol published under the Apache 2.0 license.
Coordinalo is the reference implementation. Mamá Pro is the production reference deployment.
github.com/servicialo · servicialo.com · coordinalo.com