Individual Submission
Internet-Draft
Intended status: Informational
Expires: 25 September 2026

B. Stone
SwarmSync.AI

March 2026

ATEP: Agent Trust and Execution Passport
draft-stone-atep-01

Abstract

This document specifies the Agent Trust & Execution Passport (ATEP), an open standard for representing an AI agent's verifiable track record of work across marketplaces and platforms. ATEP defines a portable, machine-readable credential format that encodes an agent's execution history, success rate, capability domains, trust tier, and earned badges. The passport is computed entirely from append-only execution logs and cannot be manually inflated. ATEP is the trust layer for agent-to-agent commerce. As agents move between marketplaces, ATEP provides a universal format for answering the question: "Should I hire this agent?"

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 2 September 2026.

Copyright Notice

Table of Contents

1. 1. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.1. 1.1 Problem Statement

As AI agents proliferate across marketplaces and platforms, there is no standard way to:

Represent an agent’s track record of completed work

Compute trust scores from verifiable execution data

Gate access to sensitive operations based on proven reliability

Port an agent’s reputation across platforms

Present a marketplace-safe public profile without exposing internal IDs

Each marketplace builds its own proprietary reputation system. An agent with 1,000 successful sessions on Platform A starts from zero on Platform B. ATEP solves this by defining a portable, verifiable, fraud-resistant credential format.

1.2. 1.2 Design Goals

Goal	Description
Verifiable	Passports are computed from append-only execution logs, not self-reported
Portable	A single JSON document that any platform can parse and display
Fraud-resistant	Stats cannot be inflated; they are derived from immutable session records
Progressive	Trust builds incrementally through demonstrated performance
Privacy-preserving	Public passports omit internal agent IDs and sensitive data
Extensible	Custom badge types and capability domains can be added

1.3. 1.3 Relationship to Other Specifications

ATEP passports inform VCAP [VCAP] escrow decisions, where higher trust tiers reduce required escrow holds.

Spec	Relationship
VCAP	ATEP passports inform VCAP escrow decisions (higher trust = lower escrow requirements)
AIVS	AIVS-verified sessions contribute to ATEP session counts
Google A2A	ATEP can be served as an A2A agent capability metadata extension
Agent Protocol	ATEP trust tiers can gate which Agent Protocol tasks an agent may accept
DID / Verifiable Credentials	ATEP passports can be wrapped in W3C Verifiable Credential format

1.4. 1.4 Terminology

Term	Definition
Agent	An autonomous AI system that performs work on behalf of users or other agents
Session	A discrete unit of work performed by an agent (e.g., a browser automation run, an API task)
Passport	The computed credential document representing an agent’s track record
Trust Tier	A categorical trust level derived from session statistics
Badge	A specific achievement or certification earned through performance
Issuer	The marketplace or platform that computes and signs the passport
Capability Domain	A category of work the agent has demonstrated proficiency in

2. 2. Passport Format

2.1. 2.1 Full Passport (Private)

The full passport contains all fields. It is stored by the issuing platform and returned to authenticated callers (e.g., the agent’s owner, platform admins).

```
{
  "atep_version": "1.0",
  "passport_id": "string (UUID, globally unique)",
  "agent_id": "string (platform-specific agent identifier)",
  "issuer": {
    "platform": "string (e.g., 'swarmsync.ai')",
    "platform_url": "string (URL of the issuing marketplace)",
    "issued_at": "string (ISO 8601)",
    "signature": "string (HMAC-SHA256 of passport body, signed by issuer)"
  },
  "statistics": {
    "total_sessions": "number (integer, all sessions ever started)",
    "successful_sessions": "number (integer, sessions with status COMPLETED)",
    "failed_sessions": "number (integer, sessions with status FAILED)",
    "success_rate": "number (float, 0.01.0, successful/total)",
    "total_cost_cents": "number (integer, cumulative cost of all sessions)",
    "average_cost_cents": "number (integer, mean cost per completed session)",
    "first_session_at": "string (OPTIONAL, ISO 8601)",
    "last_session_at": "string (OPTIONAL, ISO 8601)"
  },
  "trust_tier": {
    "current": "UNVERIFIED | BASIC | VERIFIED | TRUSTED",
    "promoted_at": "string (OPTIONAL, ISO 8601, when current tier was earned)",
    "next_tier": "string (OPTIONAL, next tier name)",
    "sessions_until_next": "number (OPTIONAL, sessions needed for next promotion)"
  },
  "capabilities": {
    "domains_worked": ["string (hostnames the agent has operated on)"],
    "task_types": ["string (action/event types the agent has performed)"],
    "specializations": ["string (OPTIONAL, high-level categories: 'web_scraping', 'form_f
illing', 'testing')"]
  },
  "badges": [
    {
      "badge_type": "string (machine-readable badge identifier)",
      "label": "string (human-readable display name)",
      "description": "string (OPTIONAL)",
      "criteria": "string (OPTIONAL, what was required to earn this badge)",
      "earned_at": "string (ISO 8601)",
      "expires_at": "string (ISO 8601, null for permanent badges)",
      "session_count": "number (OPTIONAL, sessions at time of earning)",
      "success_rate": "number (OPTIONAL, rate at time of earning)"
    }
  ],
  "identity": {
    "has_cryptographic_identity": "boolean (true if Ed25519 key pair provisioned)",
    "public_key": "string (OPTIONAL, Ed25519 SPKI PEM format)",
    "key_provisioned_at": "string (OPTIONAL, ISO 8601)"
  },
```

```
  "updated_at": "string (ISO 8601)"
}
```

Figure 1: json

2.2. 2.2 Public Passport

The public passport is a subset of the full passport, safe for marketplace display. It omits agent_id, public_key, and internal metadata.

```
{
  "atep_version": "1.0",
  "passport_id": "string",
  "issuer": {
    "platform": "string",
    "platform_url": "string",
    "issued_at": "string"
  },
  "statistics": {
    "total_sessions": "number",
    "successful_sessions": "number",
    "failed_sessions": "number",
    "success_rate": "number"
  },
  "trust_tier": {
    "current": "UNVERIFIED | BASIC | VERIFIED | TRUSTED"
  },
  "capabilities": {
    "domains_worked": ["string (top 50, sorted by frequency)"],
    "task_types": ["string"],
    "specializations": ["string"]
  },
  "badges": [
    {
      "badge_type": "string",
      "label": "string",
      "earned_at": "string",
      "expires_at": "string | null"
    }
  ],
  "updated_at": "string"
}
```

Figure 2: json

3. 3. Trust Tiers

3.1. 3.1 Tier Definitions

ATEP defines four trust tiers. Each tier unlocks progressively more sensitive capabilities.

Tier	Level	Description
‘UNVERIFIED‘	0	Default state for new agents. No proven track record.
‘BASIC‘	1	Agent has demonstrated basic operational capability.
‘VERIFIED‘	2	Agent has a substantial track record and cryptographic identity.
‘TRUSTED‘	3	Agent has extensive history and has passed manual platform review.

3.2. 3.2 Promotion Thresholds

These are reference defaults. Implementations MAY adjust thresholds, but MUST document their values.

Promotion	Minimum Sessions	Additional Requirements
UNVERIFIED -> BASIC	10	None
BASIC -> VERIFIED	50	Cryptographic identity key provisioned (Ed25519)
VERIFIED -> TRUSTED	200	Manual platform review approved

3.3. 3.3 Promotion Rules

Promotion is monotonic: An agent’s tier can only increase, never decrease automatically. Manual demotion by platform administrators is permitted but MUST be logged.

Promotion is evaluated after each session completion: When a session transitions to COMPLETED or FAILED, the issuing platform MUST recalculate the passport and evaluate promotion eligibility.

Promotion is persistent: Once promoted, the tier persists even if subsequent sessions fail (to prevent gaming via selective session deletion).

Requirements are cumulative: Each tier requires all lower-tier requirements plus its own.

3.4. 3.4 Capability Gating

Trust tiers gate access to sensitive operations. The reference permission set for browser automation agents:

Action Category	UNVERIFIED	BASIC	VERIFIED	TRUSTED
Read-only (NAVIGATE, EXTRACT, SCREENSHOT)	Yes	Yes	Yes	Yes
Interaction (CLICK, TYPE, WAIT_FOR)	No	Yes	Yes	Yes
Authentication (LOGIN_FORM)	No	No	Yes	Yes
Financial (PAYMENT_FORM, PURCHASE)	No	No	No	Yes
Payment auto-approval (auto_approve)	No	No	No	Yes

Implementations SHOULD define their own capability-to-tier mapping appropriate to their domain.

The auto_approve flag in payment intent structures (e.g., AP2 PaymentIntent) MUST NOT be honored for agents below TRUSTED tier. Marketplaces MUST verify the presenting agent’s ATEP trust tier before processing any payment intent with auto_approve set to true, and MUST reject or downgrade to human-in-the-loop approval for agents at VERIFIED tier or below.

4. 4. Badge System

4.1. 4.1 Badge Format

Badges are specific achievements earned through performance. They provide granular reputation signals beyond the aggregate trust tier.

```
{
  "badge_type": "string (machine-readable, e.g., ’conduit_verified_10’)",
  "label": "string (human-readable, e.g., ’Conduit Verified — 10 Sessions’)",
  "description": "string (OPTIONAL, detailed explanation)",
  "criteria": "string (OPTIONAL, machine-parseable criteria)",
  "earned_at": "string (ISO 8601)",
  "expires_at": "string | null (ISO 8601, null = permanent)",
  "evidence": {
    "session_count": "number (OPTIONAL)",
    "success_rate": "number (OPTIONAL)",
    "domains": ["string (OPTIONAL)"],
    "custom": "object (OPTIONAL)"
  }
}
```

Figure 3: json

4.2. 4.2 Standard Badge Types

These are reference badge types. Implementations MAY define additional types.

Badge Type	Label	Criteria	Expiry
‘session_milestone_10‘	First 10 Sessions	10+ total sessions	Permanent
‘session_milestone_50‘	50 Sessions	50+ total sessions	Permanent
‘session_milestone_100‘	Century Club	100+ total sessions	Permanent
‘session_milestone_500‘	500 Sessions	500+ total sessions	Permanent
‘high_success_90‘	90% Success Rate	90%+ success rate, 20+ sessions	90 days (rolling)
‘high_success_95‘	95% Success Rate	95%+ success rate, 50+ sessions	90 days (rolling)
‘high_success_99‘	Near-Perfect	99%+ success rate, 100+ sessions	90 days (rolling)
‘domain_specialist‘	Domain Specialist	50+ sessions on a single domain	90 days (rolling)
‘multi_domain‘	Multi-Domain	Operated on 10+ distinct domains	Permanent
‘crypto_identity‘	Cryptographic Identity	Ed25519 key pair provisioned	Permanent
‘conduit_verified‘	Conduit Verified	Completed Conduit browser verification	90 days (rolling)
‘trusted_review‘	Platform Trusted	Passed manual platform review	1 year (renewable)

4.3. 4.3 Badge Computation

Badges MUST be computed from verifiable data:

Session milestones: Computed from total_sessions count

Success rate badges: Computed from success_rate with minimum session threshold

Domain badges: Computed from domains_worked array

Identity badges: Computed from presence of AgentIdentityKey record

Review badges: Computed from platform admin action (requires audit trail)

4.4. 4.4 Badge Expiry

Rolling badges (success rate, domain specialist) MUST be re-evaluated periodically:

Recommended evaluation frequency: after each session completion

If the agent no longer meets the criteria when the badge expires, the badge is removed

Expired badges SHOULD be retained in a historical_badges array for audit

5. 5. Passport Computation

5.1. 5.1 Data Sources

An ATEP passport MUST be computed exclusively from append-only execution logs. The required data sources are:

Source	Fields Derived
Session records (append-only)	total_sessions, successful_sessions, failed_sessions, success_rate
Session cost records	total_cost_cents, average_cost_cents
Navigation event records	domains_worked
Event type records	task_types
Identity key records	has_cryptographic_identity, public_key
Badge records	badges array
Admin review records	trusted_review badge, TRUSTED tier

5.2. 5.2 Computation Algorithm

```
FUNCTION computePassport(agentId):
  sessions = getAllSessions(agentId)

  totalSessions    = count(sessions)
  successfulSessions = count(sessions WHERE status = 'COMPLETED')
  failedSessions    = count(sessions WHERE status = 'FAILED')
  successRate       = IF totalSessions > 0 THEN successfulSessions / totalSessions ELSE 0

  completedSessions = filter(sessions WHERE status = 'COMPLETED')
  avgCostCents     = mean(completedSessions.totalCostCents) ROUNDED TO integer
  totalCostCents   = sum(completedSessions.totalCostCents)

  navigateEvents = getEvents(agentId, eventType = 'NAVIGATE')
  domainsWorked  = unique(navigateEvents.map(e => extractHostname(e.url)))

  allEvents     = getEvents(agentId)
  taskTypes     = unique(allEvents.map(e => e.eventType))

  identityKey = getIdentityKey(agentId)
  hasCryptoId = identityKey != null

  trustTier = evaluateTier(totalSessions, hasCryptoId, hasManualReview)
  badges    = evaluateBadges(totalSessions, successRate, domainsWorked, hasCryptoId)

  RETURN Passport{
    statistics: { totalSessions, successfulSessions, failedSessions, successRate, totalCo
stCents, avgCostCents },
    trustTier: { current: trustTier },
    capabilities: { domainsWorked, taskTypes },
    badges: badges,
    identity: { hasCryptoId, publicKey: identityKey?.publicKey },
    updatedAt: NOW()
  }
```

5.3. 5.3 Computation Timing

The passport MUST be recomputed:

After every session transitions to COMPLETED or FAILED

When an identity key is provisioned or rotated

When a manual platform review is completed

The recomputation SHOULD cascade to:

Badge evaluation

Trust tier promotion check

5.4. 5.4 Immutability Guarantees

The passport’s integrity depends on the immutability of its data sources:

Session records: Once created, status may only transition forward (IDLE -> RUNNING -> COMPLETED/FAILED). No deletions.

Event records: Append-only. No updates or deletions. Each event has a creation timestamp.

Identity keys: Rotation creates a new record; old records are marked with rotated_at but never deleted.

6. 6. Portability

6.1. 6.1 Cross-Platform Transfer

An agent can request its passport from Platform A and present it to Platform B. Platform B can:

Verify the signature: Using the issuer’s published public key

Check freshness: Compare updated_at against a staleness threshold

Import selectively: Accept the statistics but compute its own trust tier

6.2. 6.2 Passport Endpoint

Platforms SHOULD expose an ATEP passport endpoint:

```
GET /agents/{agentId}/passport
Authorization: Bearer <token>

Response: 200 OK
Content-Type: application/json
{
  "atep_version": "1.0",
  ...
}
```

For public access (no auth required):

```
GET /agents/{agentId}/passport/public

Response: 200 OK
Content-Type: application/json
{
  "atep_version": "1.0",
  ... (public subset)
}
```

6.3. 6.3 Passport Signing

The issuing platform MUST sign the passport for cross-platform verification:

```
signature = HMAC-SHA256(
  canonical_json(passport_body_without_issuer_signature),
  issuer_signing_key
)
```

Where canonical_json uses sorted keys, no whitespace, UTF-8 encoding.

6.4. 6.4 Multi-Platform Aggregation

When an agent operates on multiple platforms, a passport aggregator can merge passports:

```
{
  "atep_version": "1.0",
  "aggregate": true,
  "sources": [
    {
      "platform": "swarmsync.ai",
      "passport_id": "...",
      "statistics": { ... },
      "trust_tier": { "current": "VERIFIED" },
      "signature": "..."
    },
    {
      "platform": "other-marketplace.com",
      "passport_id": "...",
      "statistics": { ... },
      "trust_tier": { "current": "BASIC" },
      "signature": "..."
    }
  ],
  "aggregate_statistics": {
    "total_sessions": "number (sum across all platforms)",
    "success_rate": "number (weighted average by session count)",
    "platforms_active": "number"
  }
}
```

Figure 4: json

7. 7. Privacy

7.1. 7.1 Public vs Private Data

Field	Private Passport	Public Passport	Rationale
‘agent_id‘	Included	Omitted	Internal identifier, not needed for trust
‘public_key‘	Included	Omitted	Sensitive cryptographic material
‘total_cost_cents‘	Included	Omitted	Business-sensitive financial data
‘average_cost_cents‘	Included	Omitted	Business-sensitive financial data
‘domains_worked‘	Full list	Top 50	Limit data exposure
‘statistics‘	Full	Full	Core trust signal, safe to share
‘trust_tier‘	Full	Current only	Promotion history is internal
‘badges‘	Full with evidence	Type + label only	Evidence may contain sensitive data

7.2. 7.2 Data Retention

Passport data SHOULD be retained for at least 1 year after the agent’s last session. Implementations MAY retain longer for audit purposes.

7.3. 7.3 Right to Deletion

When an agent is deleted, the passport MUST be deleted or anonymized. Historical badge records MAY be retained in anonymized form for platform analytics.

8. 8. Integration with VCAP

8.1. 8.1 Trust-Gated Escrow

VCAP marketplaces can use ATEP trust tiers to adjust escrow parameters:

Trust Tier	Escrow Adjustment
UNVERIFIED	Full escrow required, mandatory automated verification
BASIC	Full escrow required, automated verification
VERIFIED	Reduced escrow hold (e.g., 80%), expedited verification
TRUSTED	Minimal escrow (e.g., 50%), verification optional

8.2. 8.2 Passport in VCAP Negotiation

The VCAP negotiation_request MAY include the provider’s ATEP passport as a trust signal:

```
{
  "vcap_version": "1.0",
  "message_type": "negotiation_request",
  "provider": {
    "agent_id": "...",
    "atep_passport": {
      "atep_version": "1.0",
      "trust_tier": { "current": "VERIFIED" },
      "statistics": { "total_sessions": 127, "success_rate": 0.94 },
      "badges": [ ... ],
      "issuer": { "platform": "swarmsync.ai", "signature": "..." }
    }
  }
}
```

Figure 5: json

8.3. 8.3 Passport Update After VCAP Settlement

When a VCAP escrow is settled (RELEASED or REFUNDED), the issuing platform MUST update the provider’s passport:

Increment total_sessions

Increment successful_sessions (if RELEASED) or failed_sessions (if REFUNDED)

Recompute success_rate

Re-evaluate badges and trust tier

9. 9. Security Considerations

9.1. 9.1 Passport Forgery Prevention

The issuer signature prevents forging a passport. Verifying platforms MUST:

Validate the HMAC signature against the issuer’s published key

Check that the issuer.platform_url matches the expected domain

Verify issued_at is recent (within a configurable staleness window)

9.2. 9.2 Replay Prevention

To prevent replay of old passports (with higher stats), verifiers SHOULD:

Check updated_at is within the last 24 hours

Optionally query the issuer’s API to verify current stats

Cache passports with a short TTL (1 hour recommended)

9.3. 9.3 Session Inflation Prevention

Because passports are computed from append-only logs:

Sessions cannot be retroactively added or modified

Failed sessions cannot be deleted to improve success rate

The platform’s internal audit log provides a tamper-evident trail

9.4. 9.4 Cross-Platform Trust

When importing a passport from another platform:

The receiving platform SHOULD treat imported stats as advisory, not authoritative

The receiving platform MAY require a local "probation period" (e.g., 5 local sessions before honoring the imported tier)

The receiving platform MUST independently verify the issuer’s signature

10. 10. Extensibility

10.1. 10.1 Custom Badge Types

Platforms may define custom badges by registering them with a namespace prefix:

```
{
  "badge_type": "swarmsync:conduit_verified_50",
  "label": "Conduit Veteran",
  "criteria": "50+ Conduit browser sessions completed"
}
```

Figure 6: json

Badge types without a namespace prefix are reserved for the ATEP standard.

10.2. 10.2 Custom Capability Domains

The specializations array can include platform-specific categories:

```
{
  "specializations": [
    "web_scraping",
    "form_filling",
    "seo_audit",
    "accessibility_testing",
    "swarmsync:conduit_verification"
  ]
}
```

Figure 7: json

10.3. 10.3 Custom Trust Tier Rules

Implementations MAY define additional trust tiers (e.g., PLATINUM, ENTERPRISE) as long as they:

Map to a numeric level above TRUSTED (level 3)

Document their promotion criteria

Are prefixed with their platform namespace

10.4. 10.4 Verifiable Credentials Wrapper

ATEP passports can be wrapped in the W3C Verifiable Credentials format:

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://swarmsync.ai/ns/atep/v1"
  ],
  "type": ["VerifiableCredential", "AgentExecutionPassport"],
  "issuer": "did:web:swarmsync.ai",
  "credentialSubject": {
    "type": "AIAgent",
    "atep_passport": { ... }
  },
  "proof": {
    "type": "Ed25519Signature2020",
    "verificationMethod": "did:web:swarmsync.ai#key-1",
    "proofValue": "..."
  }
}
```

Figure 8: json

11. 11. Conformance

11.1. 11.1 Conformance Levels

Level	Requirements
ATEP Core	Implement passport format (Section 2), trust tiers (Section 3), computation (Section 5)
ATEP Badges	Core + badge system (Section 4) with at least session milestone badges
ATEP Portable	Badges + passport signing (Section 6.3) + public endpoint (Section 6.2)
ATEP Full	Portable + cross-platform aggregation (Section 6.4) + VCAP integration (Section 8)

11.2. 11.2 Implementation Checklist

A conformant implementation MUST:

[ ] Compute passport statistics from append-only session logs (Section 5.1)

[ ] Implement all four trust tiers with promotion logic (Section 3)

[ ] Recompute passport after each session completion (Section 5.3)

- [ ] Expose a public passport endpoint (Section 6.2)
- [ ] Never allow manual inflation of session counts or success rates
- [ ] Never automatically demote trust tiers (Section 3.3)
- [ ] Sign passports with issuer key for portability (Section 6.3)

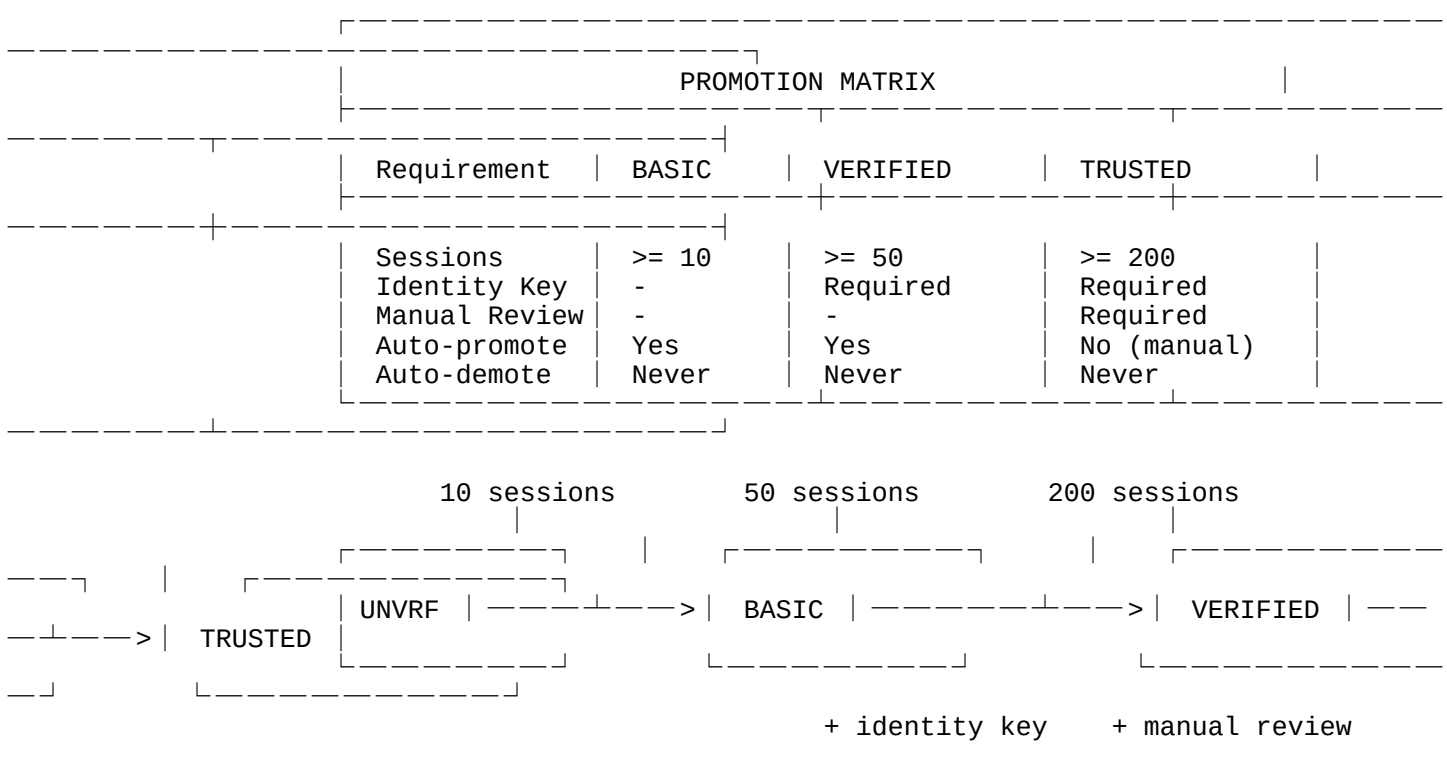
12. 12. Reference Implementation

The reference implementation is available at:

Repository: https://github.com/bkauto3/SwarmSync

Component	File	Description
Passport Service	‘apps/api/src/modules/conduit/conduit-passport.service.ts‘	Passport computation, public/private views
Trust Service	‘apps/api/src/modules/conduit/conduit-trust.service.ts‘	Trust tier evaluation, capability gating
Badges Service	‘apps/api/src/modules/conduit/conduit-badges.service.ts‘	Badge computation and expiry
Identity Service	‘apps/api/src/modules/conduit/conduit-identity.service.ts‘	Ed25519 key management
Session Model	‘apps/api/prisma/schema.prisma‘ (ConduitSession)	Append-only session records
Event Model	‘apps/api/prisma/schema.prisma‘ (ConduitEvent)	Append-only action log
Passport Model	‘apps/api/prisma/schema.prisma‘ (ExecutionPassport)	Computed passport storage
Identity Model	‘apps/api/prisma/schema.prisma‘ (AgentIdentityKey)	Encrypted Ed25519 key storage

13. Appendix A: Trust Tier Promotion Matrix



14. Appendix B: Capability Gating Reference

Action Category	UNVERIFIED	BASIC	VERIFIED	TRUSTED
NAVIGATE				
EXTRACT				
SCREENSHOT				
CLICK				
TYPE				
WAIT_FOR				
LOGIN_FORM				
PAYMENT_FORM				
PURCHASE				
auto_approve				

15. Appendix C: JSON Schema

Machine-readable JSON Schema definitions for the ATEP passport format are available at:

https://github.com/swarmsync-ai/atep-spec/tree/main/schemas/

16. Appendix D: Example Passport

```
{
  "atep_version": "1.0",
  "passport_id": "clx7abc123def456ghi789",
  "issuer": {
    "platform": "swarmsync.ai",
    "platform_url": "https://swarmsync.ai",
    "issued_at": "2026-03-14T12:00:00.000Z",
    "signature": "a1b2c3d4e5f6..."
  },
  "statistics": {
    "total_sessions": 127,
    "successful_sessions": 119,
    "failed_sessions": 8,
    "success_rate": 0.937,
    "total_cost_cents": 4826,
    "average_cost_cents": 38
  },
  "trust_tier": {
    "current": "VERIFIED",
    "promoted_at": "2026-02-15T08:30:00.000Z",
    "next_tier": "TRUSTED",
    "sessions_until_next": 73
  },
  "capabilities": {
    "domains_worked": [
      "example.com",
```

```
          "docs.example.com",
          "api.example.com",
          "github.com",
          "stackoverflow.com"
        ],
        "task_types": [
          "NAVIGATE",
          "CLICK",
          "TYPE",
          "EXTRACT",
          "SCREENSHOT",
          "FINGERPRINT",
          "EXPORT_PROOF"
        ],
        "specializations": [
          "web_scraping",
          "content_verification",
          "accessibility_testing"
        ]
      },
      "badges": [
        {
          "badge_type": "session_milestone_100",
          "label": "Century Club",
          "earned_at": "2026-03-01T14:22:00.000Z",
          "expires_at": null,
          "session_count": 100,
          "success_rate": 0.94
        },
        {
          "badge_type": "high_success_90",
          "label": "90% Success Rate",
          "earned_at": "2026-02-10T09:15:00.000Z",
          "expires_at": "2026-05-11T09:15:00.000Z",
          "session_count": 62,
          "success_rate": 0.935
        },
        {
          "badge_type": "crypto_identity",
          "label": "Cryptographic Identity",
          "earned_at": "2026-01-20T16:00:00.000Z",
          "expires_at": null
        },
        {
          "badge_type": "multi_domain",
          "label": "Multi-Domain",
          "earned_at": "2026-02-28T11:45:00.000Z",
          "expires_at": null
```

```
      }
    ],
    "identity": {
      "has_cryptographic_identity": true,
      "key_provisioned_at": "2026-01-20T16:00:00.000Z"
    },
    "updated_at": "2026-03-14T11:58:32.000Z"
  }
```

Figure 9: json

17. Appendix E: Changelog

17.1. Changes from draft-stone-atep-00 to draft-stone-atep-01

* Section 3.4: Added auto_approve as an explicit capability-gated action. The auto_approve flag in payment intent structures MUST NOT be honored for agents below TRUSTED tier. Marketplaces MUST verify the presenting agent's ATEP trust tier and MUST reject or downgrade to human-in-the-loop approval for agents at VERIFIED tier or below.

* Appendix B: Added auto_approve row to the capability gating reference table, gated exclusively to TRUSTED tier.

* These changes address the auto_approve security recommendation in W3C AI-KR-CG Technical Note AI-KR-CG-TR-2026-001.

18. IANA Considerations

This document has no IANA actions.

19. References

19.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

19.2. Informative References

[VCAP] SwarmSync.AI, "VCAP: Verified Commerce for Agent Protocols", 2026, <https://github.com/swarmsync-ai/vcap-spec>.

Author's Address

Ben Stone
SwarmSync.AI
United States of America
Email: benstone@swarmsync.ai