08 — RESOURCES · DOCUMENT PATH-WP-001

# The open protocol for the last mile of value transfer

The open protocol for the last mile of value transfer. This paper is the argument — why the formats are open, why a directory is a club, and what is deliberately left blank.

DRAFT

| | |
|---|---|
| DOCUMENT | PATH-WP-001 |
| PROTOCOL | 0.1.0 |
| API | 2026-09-09.genesis |
| STATUS | Draft |
| DATE | 2026-09-09 |
| LANGUAGE | English |

CONTENTS

## Sections

00 · Abstract abstract
01 · The last mile problem
02 · What PATH is not boundary
03 · Architecture architecture
04 · Three rings rings
05 · Reachability reachability
06 · Networks networks
07 · Privacy privacy

08 · Trust and signatures trust

09 · Discovery and conformance discovery

10 · Security security

11 · Status of this document status

00 — ABSTRACT

## Abstract

Value transfer, viewed narrowly, is a solved problem. Instant rails settle in seconds, stablecoins settle in blocks, mobile money reaches people no bank branch ever will. What breaks is everything around the movement: knowing who someone is, where to send them, what they accept, what is owed, who is allowed to act, and whether it arrived. Every one of those questions crosses an organisational boundary. Every one of them is rebuilt, incompatibly, by each operator.

PATH standardises that last mile, and nothing else. It does not move money, hold funds, issue a token, verify identities, or price a transaction. It specifies the objects that two organisations have to agree on if those organisations are not the same company.

The load-bearing split is this. Formats are open: anyone may implement them without asking permission. Directories are closed: being findable by strangers means publishing a resolvable pointer to a customer base, and no regulated institution does that without reciprocity, contract and audit. Both have been true at once in the card industry for forty years. PATH applies the same split to the last mile.

This document is v0.1. Formats may still change. The changelog is the record of that, not a promise of stability. Nothing in it describes identifier hashing as zero knowledge. With a server-side pepper it is not.

01 — THE LAST MILE

## The last mile

Before a single unit of value moves, a payer's application has to establish six facts. None of them is "did the rail settle". All of them are currently private dialects.

FIG. 01 — BEFORE VALUE MOVES

| | | |
|---|---|---|
| 01 | **IDENTITY** | A stable handle that survives changing provider. |
| 02 | **ADDRESS** | Where value reaches them, and on what standing terms. |
| 03 | **ACCEPTANCE** | What they take. The wrong rail is a support ticket. |
| 04 | **DEMAND** | How much is owed, to whom, until when. |
| 05 | **AUTHORITY** | Whether a delegate or agent may act, and within what limits. |
| 06 | **PROOF** | Whether it arrived, in a form an auditor can hold. |

Six questions. Five become pillars. Liquidity is reserved.

A recipient is reachable only inside one system. An identifier belongs to the institution, not the person. A payment request cannot leave its origin. A receipt is only as good as the relationship that produced it. Delegation is all or nothing. Those are not product gaps. They are the cost of treating the last mile as a private integration problem.

### Why the usual fixes do not hold

| FIX | WHAT IT BUYS | WHY IT FAILS |
|---|---|---|
| Bilateral integrations | Two operators can actually pay each other | Cost is quadratic. The network stops growing before it is useful. |
| A single global directory | Anyone can find anyone | One operator holds the map of everyone's customers. No competitor accepts it. No regulator should. |
| A closed network alone | Trust, audit, exclusion | Everyone outside the club is invisible. The original complaint, restated. |

PATH separates the two things that keep getting conflated. Formats are public. Directories belong to networks. A member joins one club rather than integrating fifty peers. An implementer who joins nothing can still speak the protocol with anyone who will speak it back.

02 — WHAT PATH IS NOT

## What PATH is not

Restraint is not modesty. A protocol that also specified how to run a payment company would be implemented by nobody, because it would need renegotiating in every jurisdiction it touched.

FIG. 02 — OUT OF SCOPE

| | | |
|---|---|---|
| 01 | **A BANK, A WALLET, A PROVIDER** | It carries no money and holds no funds. |
| 02 | **A TOKEN** | Nothing needs to be issued or held to use the protocol. |
| 03 | **A KYC VENDOR** | Regulated issuers stay the regulated parties. |
| 04 | **A SINGLE GLOBAL DIRECTORY** | Directories belong to networks. There can be many. |
| 05 | **A PRODUCT** | Portals, apps and payment surfaces belong to operators. |
| 06 | ~~**A ROUTING ENGINE**~~ | Liquidity is reserved, not hidden. |

If a field is not read across an organisational boundary, it is not in PATH.

> **PATH standardises what crosses a boundary between two organisations. Nothing else.**

Run that line against a payment request and the split is unambiguous. The reference, the amount, the expiry, the disclosed fee, the signature and the receipt are in the protocol. The title, the image, the basket, the tax, the landing page and the provider integration are not. The right-hand column is where products differ. The left-hand column is where they have to agree, or nothing interoperates.

03 — ARCHITECTURE

## Architecture

Five pillars, each answering one question. The questions do not overlap. Four start from the payer. One starts from the receiver — and that one is the part a buyer actually sees.

| FIG. 03 — FIVE PILLARS | | |
|---|---|---|
| 01 | **ID** | Who are you? |
| 02 | **ADDRESS** | Where do I send, and on what standing terms? |
| 03 | **REQUEST** | What is owed, to whom, until when? |
| 04 | **CONNECT** | Who is allowed to act, within what limits? |
| 05 | **SETTLEMENT** | How is the outcome proven? |
| 06 | ~~**LIQUIDITY**~~ | Reserved — receipts include route; the value is null. |

Liquidity is the sixth cell: declared, empty, so it arrives as an extension.

An address carries a policy: I accept this asset on this rail, up to this much. A payment request carries a claim: you owe me this amount, for this order, until Friday. Those are different speech acts. A system that stores them in the same place ends up unable to expire one, revoke the other, or tell a payer which is which.

### Transverse layers

| LAYER | WHAT IT DOES |
|---|---|
| PATH INTEROP | One URI, carried by QR codes, deeplinks and in-app handoffs alike. |
| PATH FINDER | Reachability, in two steps: SONAR then RESOLVER. |
| PATH COMPLIANCE | Counterparty data exchange, on a channel separate from the directory. |
| PATH DISPUTES | Refunds, disputes, and the rulebook that decides who pays. |
| Base layer | Discovery, fee disclosure, portability, error and webhook registries. |

> **PATH INTEROP carries the envelope. The pillar owns the object.**

| URI KIND | THE OBJECT BELONGS TO |
|---|---|
| pay, claim | ADDRESS |
| request, checkout | REQUEST |
| subscribe, connect, fconnect, mandate | CONNECT |
| present | ID |
| receipt | SETTLEMENT |

URI kind to owning pillar.

Two rows repay attention. pay points at a destination while request carries a claim — they look alike in a QR code and behave differently everywhere else. subscribe looks like a payment link but creates a standing permission, which is why it belongs to CONNECT. File it under REQUEST and you will ship without revocation or counters and discover it in production.

### Status labels

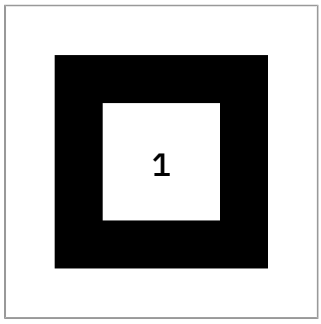
| STAMP | MEANING |
|---|---|
| LIVE | Specified and running in at least one implementation. |
| DRAFT | Specified, subject to change before v1.0. |
| RESERVED | Present in the format, unused in this version. |

RESERVED means the field is present on the object and unused today. PATH LIQUIDITY is the example: every settlement receipt includes route, and in this version the value is null. PIP-0005 defines the only shape it may take.

04 — THREE RINGS

## Three rings

FIG. 04 — THREE RINGS




01 **OPEN**

The formats. Implementable without permission.

02 **THE NETWORK**

Directory, discoverability, compliance, disputes.

03 **THE OPERATOR**

Customer data, KYC documents, wallets, private keys.

The protocol is the core. The operator never enters it.

Ring 1 is the protocol. Identity, addressing, standing intent, requests, authorisation objects, receipts, the interop URI, signatures, error codes. Implementing any of it requires access to nobody's data. You can do it alone, and nobody can stop you.

Ring 2 is a club, and has to be. There is exactly one capability in PATH where the shared resource is not a format but other people's customers: the directory. Entering it means publishing a resolvable pointer to your customer base. No regulated institution does that without reciprocity, a contract, an audit right, and the ability to exclude whoever abuses it.

Ring 3 never leaves the operator. Customer records, KYC documents, wallets, keys. Not in the protocol, not in a network, not in any circumstance.

FIG. 05 — THE SPLIT

02.1

**OPEN RING**

The formats. Implementable without permission. Testable by a third party.

02.2

**CLOSED NETWORK**

The directory. Admission, contract, audit, exclusion.

ISO 8583 is open. Visa is not. Nobody finds that contradictory.

Closing the directory replaces expensive cryptography with cheap governance. Querying a peer without revealing the identifier, finding which operator holds a number, preventing scraping, knowing who to trust, deciding who pays when something goes wrong: a club with a rulebook can answer those. An open index cannot, except by inventing cryptography that excludes the members who most need to join.

NOTE

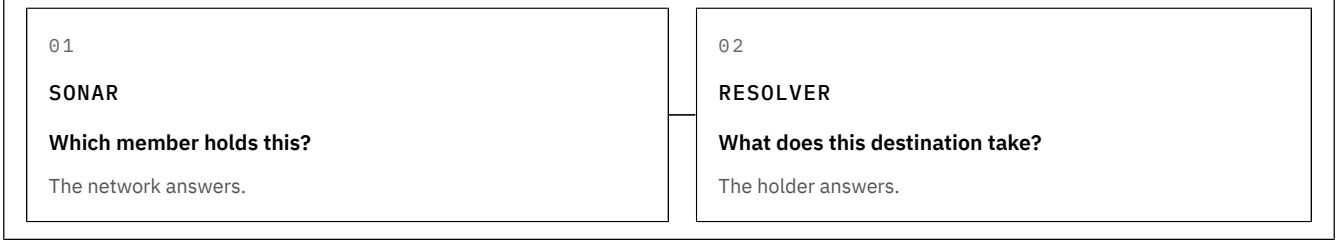
**The bill**

Adoption is capped by admission. The club reintroduces the gravity the open model was meant to escape. Deciding who may join is a supervised power. Prevention becomes deterrence: an audit observes afterwards. And the central point, in the usual configuration, sees every search. None of this invalidates the choice. The open alternative left the directory problem with no answer at all.

# Reachability

Reaching someone is not one question. It is two, and they have different answerers. That is the page to read if you only read one.

FIG. 06 — FINDER




Finder is both, composed on the client. No single party sees the search and the answer.

The obvious simplification is to have the network answer both — one call, one round trip. It is also the one thing that would make the directory unacceptable to its own members. A network that answers step 1 learns who is being looked for. A network that also answers step 2 learns what that destination can receive: rails, assets, limits, and how those change. That is a competitor's product roadmap, arriving daily.

So the composition happens on the caller's side. The SDK makes both calls; the integrator writes one. The cost is a round trip. What it buys is that no single party sees both halves. A network may offer to relay step 2. If it does, it must say so in its discovery document rather than doing it silently.

### The step you can skip

FIG. 07 — PROFILES




Relationship is not a degraded mode. It is the proof that ring 1 needs no permission.

Two parties that already have a relationship resolve directly. They implement addressing, resolution, requests and receipts, and they join nothing. Every claim on this site about openness rests on that path existing and being documented.

### Answers are signed, and constant-shaped

Both steps return signed envelopes. Keep them. Months later, when a payment went somewhere unexpected, the question is what you were told before you sent — and a signed answer turns that from one party's recollection into a document.

SONAR returns the same shape whether a key is unknown or known-but-not-visible to the caller.

06 — NETWORKS

# Networks

There will be more than one network, and the protocol expects them to interoperate. Open Connect is the first PATH network. It is not the protocol. A West African club, a European club, a private banking group: all speak ring 1. Each writes its own rulebook.

| A NETWORK CHOOSES | THE PROTOCOL FIXES |
|---|---|
| Who is admitted, and on what evidence | The format of every object exchanged |
| Whether the directory is credential-gated or open | That negative answers are indistinguishable |
| The reciprocity budget | That searches are logged and retention declared |
| Whether holders co-sign directory answers | That what a destination accepts is answered by the holder |
| The rulebook, disputes, liability | That the rulebook is published and versioned |

What a network chooses, and what it may not.

A network may not answer on behalf of its members about what they accept. A directory that did would hold its members' capability data — ring 3 material sitting in ring 2 — and no second member would sign that.

### Two directory topologies

| MODE | WHO ANSWERS | WHAT IT COSTS |
|---|---|---|
| central | The network holds the index | Simple to join. The network sees the searches. |
| broadcast | The member asks peers, filtered by coverage | Observation is multiplied, not removed. Forty-nine peers learn that someone is looking. |

Broadcast is workable with declared coverage. Local search against a published index is not part of this design. The hashing secret stays with the network.

### Discoverability belongs to the entry

| LEVEL | WHO CAN FIND IT | TYPICAL FOR |
|---|---|---|
| private | Only a party already in a relationship | The most cautious |
| network | Members of the network | The default for a person |
| public | Anyone | Opt-in — in practice, merchants |

A merchant wants to be publicly findable. A private individual never is by default. Even where the key is public — a company register number is public by construction — the answer is not. "This company banks with that member" is commercial information. A public entry opens the key, not the budget.

07 — PRIVACY

## Privacy

Most privacy failures in payment systems come from collapsing authentication, identifiers and consent into one record. PATH keeps them apart. A disclosure requires all three.

| RETURNED BY A LOOKUP | NEVER RETURNED |
|---|---|
| Which member holds the key | The account holder's name |
| That member's endpoint | Their institution's customer record |
| What the destination accepts | Their KYC tier or documents |
| Applicable limits | Their balance or history |
| A signature and a commitment | Any other identifier of theirs |

Identifiers are normalised, then hashed with a network-held secret — a pepper — using Argon2id. The directory stores hashes; it never stores the identifier.

> **The hash protects the database, not members from the network.**

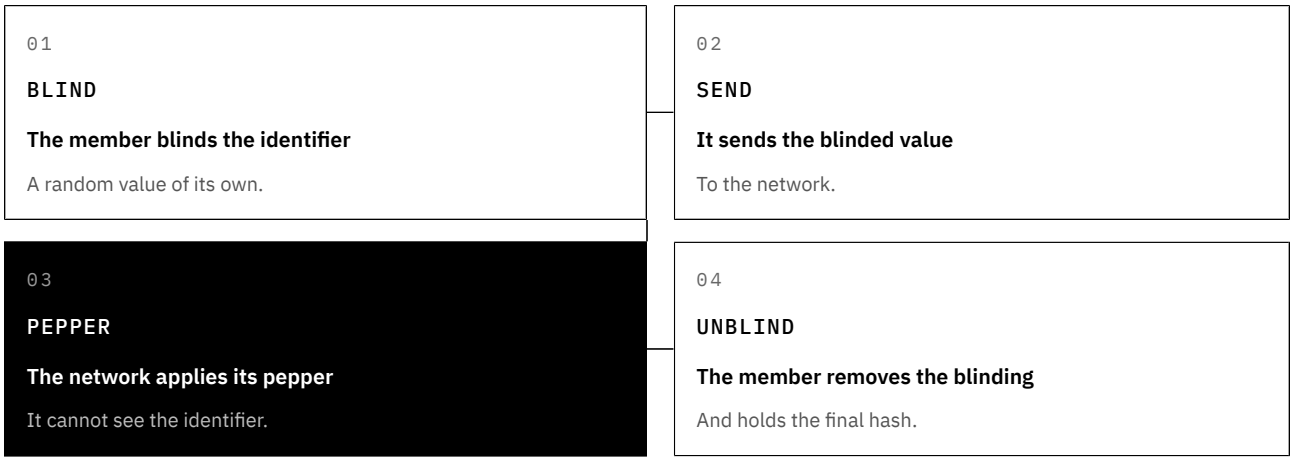
Identifiers are not stored in the clear. The member still sends the identifier to the operator that hashes it. That is not zero knowledge.

NOTE

**The pepper is not distributed**

Members do not receive the hashing secret. Lookups are hashed by the network. Sharing that secret is not part of this design.

FIG. 08 — OBLIVIOUS HASHING




Declared. Not shipped in v0.1. The network sees nothing. The member never learns the pepper.

Until that construction ships, the finder block of a discovery document must say hashing is server_side. A prospective member has a right to know, and to refuse.

08 — TRUST AND SIGNATURES

## Trust and signatures

> **Anything a counterparty may quote back is signed.**

Resolver answers, directory answers, payment requests, settlement receipts, holder confirmations, attestations. If a statement might matter later, it carries a signature. TLS proves you talked to the right server. It says nothing about the statement that server handed you, and it leaves nothing behind.

| WHAT A SIGNATURE BUYS | WHAT TLS DOES NOT |
|---|---|
| It survives the connection | Six months later the statement is a document, not a recollection. |
| It is checkable by someone who was not there | An auditor, a regulator, a counterparty's counterparty. |
| It survives a compromised intermediary | A hostile proxy cannot produce a different answer that verifies. |

Signatures are Ed25519. Not an HMAC. A shared secret proves that someone holding the secret produced the value — and the recipient holds it too, so the recipient could have produced it. That is fine for a session cookie and useless for a statement a third party has to rely on.

Everything signed is serialised as RFC 8785 canonical JSON before signing: keys sorted, no insignificant whitespace, undefined dropped, numbers in shortest round-trip form. JSON.stringify preserves insertion order. Two implementations produce different bytes for the same object, therefore different signatures, and neither can verify the other. It works in testing — one implementation talking to itself — and fails the day a second one appears.

09 — DISCOVERY AND CONFORMANCE

## Discovery and conformance

An operator publishes a discovery document at /.well-known/path-configuration. Unauthenticated, always. Anyone considering an integration has to read it before there is a relationship to authenticate. A discovery document behind a key is a brochure.

Declaring is committing. A profile announced and not honoured is a non-conformity, not an approximation. The finder block is the material part for a prospective member: sonar mode, hashing, who may query, the reciprocity budget, how long searches are kept. Those answers change whether an institution can join, and they belong in public rather than in a contract annex.

| PROFILE | WHAT IT ASSERTS |
|---|---|
| PATH-CORE.Discovery | The document is published and complete. |
| PATH-ID.Core | Subjects, identifiers, attestations. |
| PATH-ADDR.Format / Standing | Addresses and standing terms. |
| PATH-REQ.Link / Payer / Checkout | Requests a wallet can read, pay and hand back. |
| PATH-FINDER.Relationship / Directory | Resolver alone, or Sonar then Resolver. |
| PATH-INTEROP.Read / Issue / Pay | Parse both forms, issue signed HTTPS, pay with declared rails. |
| PATH-SETTLE.Receipts | One signed receipt format across rails. |

Conformance is named, not implied.

The published register of networks is a small public list of clubs, not an admission desk. PATH lists them. Each network decides who sits in it.

10 — SECURITY

## Security

The realistic directory attacker is not an anonymous outsider. It is an authenticated participant, or someone holding their key.

| ADVERSARY | CAPABILITY | PRIMARY DEFENCE |
|---|---|---|
| A legitimate member, curious | Valid credential, patience | Reciprocity budgets, logging, the rulebook |
| A stolen credential | Everything the member could do | Request signing, short freshness, revocation |
| A crafted payload | A QR code or deeplink fed to a reader | Anchored parsing |
| A network operator | Sees searches; holds the index | Declared retention, audit, contract — commitments, not properties |
| A compromised resolver | Answers with the wrong destination | Signatures and commitments |
| A passive observer | Watches traffic | TLS |

### Mistakes that cause incidents

01 Unanchored URI parsing. Match the complete string, at both ends.

02 Symmetric signatures on public statements. An HMAC the recipient could have forged proves nothing to a third party.

03 Non-canonical serialisation. One implementation talking to itself hides the bug until a second language appears.

04 Returning a profile from a lookup. A routing answer that includes a name is a disclosure.

05 Looking up an address book on install. The reciprocity budget exists because this is the default instinct, and it is the wrong one.

Security issues in the specification or the reference implementation go to security@pathprotocol.dev. Issues in a specific operator's deployment go to that operator — its contact is in its discovery document.

11 — STATUS OF THIS DOCUMENT

## Status of this document

This paper describes PATH 0.1.0, a draft. Later objects are in the changelog. Protocol versions and API versions move independently: the first changes the objects, the second changes this implementation's HTTP surface.

Out of scope for v0.1, and declared so: liquidity routing; any settlement token; selective disclosure over attestations; the full checkout lifecycle (partial payments, multiple attempts, refunds tied to an order). The hand-back is specified. The rest waits. A specification that covers three things well beats one that announces five and delivers one.

The reference implementation, the TypeScript SDK, and two MCP servers are published in the same repository as this paper. They are a way to contradict the specification, not a product. Open Connect is the first network to speak it. Other networks will. They will not be PATH, and PATH will not be them.

> **The protocol is open. A directory is a club.**