The Binding Record

How we join an AI agent's identity to the money it moves, why every number above that join depends on it, and what the first census of ERC-8004 agents on Base shows once the join is made honestly

Agentic Finance Graph — Research Note 01, version 2.1 (supersedes versions 1.0 and 2.0 of the same day) · Nikolai Simonyan, Agentic Finance Graph · Live figures as of **2026-09-27 11:52 UTC**; worked examples as read on 27 September 2026 · CC BY 4.0 · agenticfinancegraph.com/research

Abstract. Every public dashboard of "AI agent money" draws the same arrow, from an agent's identity to a wallet to a payment, and none of them specifies it. This note specifies it. A *binding record* is a versioned, evidence-ranked join from an ERC-8004 identity to the wallet that spends for it: it names the method that proved the join, assigns a confidence that is a function of that method rather than a blend, carries the block range in which it is valid, and is closed rather than edited when the identity changes hands. Every payment we count hangs off a binding, so attribution can be replayed at any block height. We then apply the record to the whole registry on Base rather than to a sample. Of 95,882 identities held by 48,822 wallets, 1,198 have been observed paying a different address after the identity existed. Two exclusions decided from the binding and the transaction receipt remove $72.22M of transfers made before the identity was minted and $18.08M of routing hops, leaving $9.07M of counted payments to 955 recipients. Identifying those recipients shows that 59% of the counted value is agents' own cash entering lending protocols and vaults through policy adapters, 18% leaves the chain through bridges, and 4.4% buys work through an escrow; the largest wallet alone is 27% of the total and the median payment is $54.41. Six worked examples show what the record does to individual agents, including one wallet whose $34.57M of pre-registration history would otherwise have been reported as agent spending. We close with the adversarial cases the record is built to resist, what it makes possible for insurers, platforms, agents and auditors, and the open questions. Every figure resolves to a public metric identifier and can be reproduced from the endpoints in Section 10.

1. The problem, stated plainly

There are four public facts about any AI agent that has money: an identity, written into a registry as a token; a wallet, which is a key or a contract that can move funds; a payment, which is a transfer on a chain; and a behaviour, which is the pattern those payments make over time. The sector publishes the first and the third in abundance. Registries report how many identities were minted. Payment scanners report how many dollars moved over a rail. What nobody publishes is the join between them, and the join is where every interesting claim lives. "This agent has paid for something" is not a claim about a token; it is a claim about a wallet, made on behalf of a token, and it is only as good as the evidence that the wallet spends for that token.

We noticed the gap the way most measurement gaps get noticed, by trying to reproduce someone else's headline and failing. Registration counts in this sector are quoted as workforce numbers; they are counts of names issued, and 20.6% of the names on Base have since been resold and 28.5% were minted in bursts of twenty-five or more from one address inside an hour. Rail volumes are quoted as agent commerce; they include every transfer the rail carried, which on Base is mostly humans and contracts. When we joined identities to wallets ourselves and read what left those wallets, the raw figure that

most dashboards would print was $99.37M. The figure that survives the two exclusions described below is $9.07M. The difference is not a rounding question; it is the difference between a market and a rumour.

So this note is about the join. We call the object that carries it the binding record, we publish its specification so that other indexers can converge on it and argue with it, and we show what the registry on Base looks like once every payment is made to hang off one.

2. The binding record

2.1 The object

Let R be the set of identities minted in the ERC-8004 Identity registry on Base (contract `0x8004A169FB4a3325136EB29fA0ceB6D2e539a432`), each identity a with a mint block $b_{\mathrm{reg}}(a)$. Let W be the set of chain-qualified wallet addresses. A binding record is a tuple

$$\beta = \big(\, a,\ w,\ m,\ c,\ [h_{\mathrm{from}}, h_{\mathrm{to}}),\ E,\ \sigma \,\big)$$

where $a \in R$ is the identity, $w \in W$ the wallet, $m \in \{0, \dots, 5\}$ the *method* by which the join was established, $c \in [0, 1]$ a confidence, $[h_{\mathrm{from}}, h_{\mathrm{to}})$ the block interval in which the binding is valid (h_{to} open while the binding is alive), E the evidence that produced it (a transaction hash, a signed message, a registration file with its content hash, each with the time it was observed), and σ an optional pointer to the binding that superseded this one. The published object carries the same fields under the names `agent_gid`, `wallet`, `controller`, `bind_method`, `confidence`, `valid_from_block`, `valid_to_block`, `evidence[]`, `superseded_by` and `cluster_id`; the specification page lists them with their types.

Two rules turn this from a snapshot into a ledger. First, a binding is born, dies, and can be superseded; it is never edited in place. Second, every row that records money references a `binding_id`. Together they give the property we care most about: for any block height h, the wallet attributed to an identity is

$$\beta(a, h) \;=\; \{\, \beta : a(\beta) = a,\ h_{\mathrm{from}}(\beta) \le h < h_{\mathrm{to}}(\beta) \,\}$$

and a payment made at height h is attributed through $\beta(a, h)$ alone. Attribution is therefore *replayable*: given the binding table and the chain, anyone can recompute who paid what on behalf of whom at any past height, and the answer does not change because a token was later sold.

2.2 Methods are ranked, never averaged

A confidence score that blends a cryptographic signature with a guess is worse than no score, because it hides which of the two you are holding. We therefore fix the confidence as a function of the method, $c = \kappa(m)$, and publish the table. Nothing is averaged in; if two bindings exist for one identity, the record with the higher method wins and the other is kept, closed, as evidence.

Table 1. Binding methods, their default confidence $\kappa(m)$, and how hard each is to forge.

m	Method	$\kappa(m)$	Forgeability	Status
5	The wallet signs a challenge over the agent's global id and a nonce (on-chain or EIP-712)	0.95	Hard: requires the key	Offered through the Desk; lifts a binding to the top rank
4	An EIP-7702 authorization or session policy whose signed payload embeds the agent id	0.90	Hard	Specified; read when policies carry the id
3	The registration declares its wallet (`agentWallet` in the on-chain metadata or the registration file) *and* that wallet is the observed spender	0.70	Medium: a declaration can be copied, but the spender must match it	Live: 307 open bindings on 2026-09-27
2	A wallet declared only in a URI-hosted file that can change after the fact	0.45	Easy	Reserved; not used for ranking
1	The registration's holder, read with `ownerOf(tokenId)`, is the observed spender	0.55	Medium: a holder can hold many things	Live: 95,527 open bindings, most of the registry
0	Name or ticker similarity, or clustering on shared funders or implementations	0	Trivial	Never a binding; only a cluster hint

Two things in the table are deliberate and worth defending. Method 3 sits above method 1 even though the holder of a token is, in a sense, the most obvious owner: a declared wallet that is also the observed spender has passed two tests, the declaration and the behaviour, whereas holding is one test that a factory address passes for thousands of identities at once. And method 1 is nevertheless a binding and not merely a hint, because on this registry the holder is, in the overwhelming majority of cases, the only wallet the identity has; refusing it would make the census impossible and would not make the remaining numbers truer. The record says which method bound it, and the public rank says so too, so a reader can discount as they see fit.

2.3 What the record refuses to be

A binding is evidence of *control*, not of *operation*. It does not say that a program rather than a person signs the transactions; the chain does not carry that fact and we do not infer it. It does not say who deployed or runs a wallet's controller contract. It does not say that a payment was for a good or a service. Those are separate claims needing separate evidence, and the record's `does_not_assert` list travels with every figure derived from it. When a later section reports that a wallet "parked cash at yield", that is a statement about a labelled contract and a receipt, not about intent.

3. From the binding to a counted payment

Let T be the set of ERC-20 `Transfer` events of four stablecoins on Base (USDC, USDT, EURC, DAI), each with sender $\mathrm{from}(t)$, recipient $\mathrm{to}(t)$, block $b(t)$ and USD value $v(t)$, EURC converted at the day's EUR/USD rate. The *counted set* is

$$C \;=\; \Big\{\, t \in T : \exists\, \beta \in \beta(a, b(t)),\; \mathrm{from}(t) = w(\beta),\; b(t) \geq b_{\mathrm{reg}}(a),\; \mathrm{to}(t) \neq \mathrm{from}(t),\; \neg\,\mathrm{pass}(t) \,\Big\}.$$

Three clauses do the work. The transfer must come from a wallet with a binding valid at that block. It must be dated at or after the identity's mint: a wallet may have had a long life before someone minted an identity into it, and that life is not the agent's. And it must not be a hop. The hop test is decided from

the transaction receipt: writing $\mathrm{in}_w(t)$ and $\mathrm{out}_w(t)$ for the stablecoin value that entered and left the sending wallet within the same transaction,

$$\mathrm{pass}(t) \iff \mathrm{in}_w(t) > 0 \ \wedge \ \mathrm{out}_w(t) > 0 \ \wedge \ \frac{|\mathrm{in}_w(t) - \mathrm{out}_w(t)|}{\max\left(\mathrm{in}_w(t), \mathrm{out}_w(t)\right)} < \varepsilon, \qquad \varepsilon = 0.02.$$

Money that arrived and left in comparable size inside one transaction is a router doing its job, not an agent buying anything; the two-percent tolerance admits the fees such routers take and was set by looking at the fee distribution of the routes we had flagged by hand. The published total is $V(C) = \sum_{t \in C} v(t)$, and the two excluded totals are published beside it, so that

$$\underbrace{\sum_{t} v(t)}_{\text{left bound wallets}} \quad = V(C) \ + \ V(\mathrm{Pre}) \ + \ V(\mathrm{Pass})$$

reconciles to the dollar, with $V(\mathrm{Pre})$ the value dated before the identity existed and $V(\mathrm{Pass})$ the hops among transfers that were not already excluded by date.

The liveness ladder. The public rank is the highest level whose predicate holds. L0: the identity exists. L1 to L5: the registration file is declared, retrievable, valid, names a service, and that service answers a probe. L6: the service speaks MCP or A2A. L7: $\exists\, t \in C$ attributed to the identity. L8: L7 and the paying wallet holds at least \$10 in the four stablecoins at the last reading. L9: L8 and $\left|\{\mathrm{to}(t) : t \in C_a,\ \mathrm{age}(t) \le 30\ \mathrm{d}\}\right| \ge 3$. Public rankings begin at L7 because it is the first rung that costs money to fake. Reputation feedback written on-chain is published beside the ladder and given no weight inside it; ten addresses wrote over half of all such feedback, and a rank that counted vouches would be climbable by ten keys.

4. Method: the census, and what it did to the numbers

Until 27 September 2026 the graph admitted an identity only when a uniform random prober had sampled its registration file, and its wallet was read for payments only after that. Every count we published was therefore a floor drawn from a sample. The census changed the order of operations. We read the registry's complete owner list (five pages from a node provider's NFT index, 95,882 identities with a non-burn holder), created an actor at L0 with a method-1 binding for every one of them, and dated each actor from the mint block. Registration files are still probed on a rolling sample, because a probe is slow and an ownership read is not.

Then the wallets. Reading each of 48,822 wallets individually would have cost more than the month's budget, so we asked the chain the other way round: one log query per two thousand wallets returns every stablecoin `Transfer` whose sender is any of them, over the whole registry era. A wallet that appears nowhere has provably sent none of the four stablecoins and is marked checked without a further call; a wallet that appears is queued for a per-wallet read of its entire history, up to ten thousand transfers, followed by the receipt check above. The sweep now repeats every three hours from the last head block, and the wallets of ranked agents are additionally pushed to us in real time. Where the old scan read a wallet's newest fifty transfers and stopped, the new one reads everything, which matters for treasuries that move money in large, infrequent blocks.

The effect on the numbers was large and, for a night, uncomfortable. Agents observed paying went from 553 to 929 by the first morning run and to 1,198 by the second; counted spending from \$878,250 to \$2.98M and then to \$9.07M. A count that triples overnight is either a better instrument or a broken one, and the only way to know is to look at what entered. We held the figures until we had identified where

the new money landed, address by address, from the verified source of the receiving contracts and from the receipts of the largest transfers into them. Section 6 tells that story; the short version is that the instrument was better, and that most of the money it found is not what the phrase "agent spending" makes a reader imagine.

Labels. A recipient address receives a category only with a citable basis: the verified name of its contract (or of the implementation behind its proxy), a curated explorer name tag, or, for an unverified contract, sampled receipts showing where the full amount went next within the same transaction. A label describes what the code does. It never names an operating company, because a contract name is not evidence of who deployed or runs it. The share of counted value in category k is $s_k = \sum_{t \in C,\, \ell(\mathrm{to}(t))=k} v(t) \,/\, V(C)$.

Checks. Thirty-seven assertion queries run against the tables every three hours: a payment counted from before its identity existed, the same log stored twice, a wallet paying itself, an actor with a countable payment but no rank, a category share that does not reconcile to the raw count. A blocking failure withholds the affected figures. The checks are published with their failing counts, because an index that cannot show how it is wrong today cannot be the book anyone underwrites against tomorrow.

5. Results

Table 2. The registry on Base, as of 2026-09-27.

Quantity	Value	Note
Identities with a current holder	95,882	ERC-8004 Identity registry, non-burn owners
Distinct holder wallets	48,822	11,877 wallets hold more than one identity; one holds 8,313
Distinct minting addresses	30,984	the ten busiest minted 35.1% of the registry
Minted in bursts (≥ 25 from one address in an hour)	28.5%	a registration count is partly a count of factories
Resold at least once	20.6%	a binding is closed at the transfer block and must be re-proved
Declared endpoint answers a probe (L5)	3.4%	uniform random sample of registration files
Wallets delegating signing to code (EIP-7702)	15.3%	read from wallet code, implementations named
Observed paying after registration (L7+)	1,198	1.25% of identities; one in 80
… holding ≥ $10 in stablecoins (L8+)	112	balance at last reading, four stablecoins
… three or more counterparties in 30 days (L9)	17	

Table 3. Partition of every dollar that left a bound wallet (four stablecoins, Base).

Component	USD	Share	Decided by
Counted payments, $V(C)$	$9,067,469	9.1%	binding valid at the block; after the mint; not a hop
Excluded: before the identity existed	$72,216,747	72.7%	$b(t) < b_{\mathrm{reg}}(a)$; 926 identities carry such history
Excluded: routing hops	$18,082,675	18.2%	$\mathrm{pass}(t)$ from the receipt, $\varepsilon = 0.02$
Total that left bound wallets	**$99,366,890**	100%	a naive total is 11.0× the counted figure

Counted payments number 16,401 across 955 distinct recipients. The largest paying wallet is 26.8% of $V(C)$, the ten largest are roughly three quarters of it, and the median counted payment is $\$54.41$. Of the 1,198 ranked agents, 442 (a direct query of the ledger's tables at 11:45 UTC on the day of this note) have wallets that carried stablecoin history before their identity was minted; for them the date rule is the difference between a small number and a large fiction. Contracts rather than plain addresses receive 88.7% of the value paid to the five hundred largest recipients.

Table 4. Where counted value lands, by labelled category.

Category	USD	s_k	Basis of the labels
Lending and vault adapters	$5,325,171	58.7%	A verified contract named `Adapter` behind a beacon proxy, whose own NatSpec says it wraps Aave v3 supply and withdraw and any ERC-4626 vault deposit and withdraw "so receivers are always bound to `msg.sender`", called by session keys with typed parameters; Morpho's general adapter (explorer tag); Euler `EVault` (verified beacon implementation)
Bridges	$1,622,001	17.9%	Across `SpokePool`, `RelayDepository`, LI.FI `Permit2Proxy` and `LiFiDiamond`, `CashmereCCTP` (verified); three unverified routers whose sampled receipts forward the full amount to the Across SpokePool in the same transaction
Work escrow	$397,134	4.4%	Verified `TermixEscrow`: an order lifecycle with a dispute layer, declared in its own events
Swaps	$127,806	1.4%	Verified `UniswapV3Pool`, `CLPool`, a `Diamond` aggregator; one unverified fee-taking router (receipts: stablecoin in, WETH out to the agent)
Other yield vaults	$26,082	0.3%	Verified ERC-4626 vaults
Not labelled	$1,569,274	17.3%	Mostly plain addresses. We do not guess

5.1 Money in, the seller side of x402, other chains, holdings

Reading transfers *into* ranked wallets under the same exclusions gives $437,378 received by 182 agents from 162 distinct payers. Only $4,745 of it, in 140 transfers, came from wallets that are themselves bound to an identity: agents paying agents is, at this date, a rounding error. The rest is operators funding their own agents and withdrawals from the venues above, which is why income is published with the same refusal as spending: it does not assert that the money was earned.

Coinbase's x402 Bazaar lists the receiving addresses of sellers of paid HTTP resources. Reading stablecoin transfers into those addresses on Base over the last twenty-four hours gives 1,322 paying wallets, $28,334 across 29,367 transfers to 327 sellers. Of those wallets, 22 are bound to an ERC-8004 identity. The same read on Solana gives 964 transfers worth $49. Inside the counted set, x402 settlements are 5.2% of transfers. The population that pays for machine services over HTTP and the population that registers identities are, so far, largely different wallets.

The same registry contract on Ethereum, Arbitrum One, OP Mainnet and Polygon holds 54,266 identities with 13,462 holder wallets. Their stablecoin transfers are read under the date rule but not yet receipt-checked, so they are published as transfers, not as counted spending, and are never added to the Base figures. The priced holdings of 607 wallets bound to ranked agents total $53,119 (stablecoins $1,974, ETH $28,013, other priced tokens $23,129); the median such wallet holds $\$1.16$. Unpriced tokens, mostly airdropped spam, are counted and never valued.

6. Worked examples

Numbers in aggregate hide what the record does to a single agent. The six cases below are real, public, and cited by transaction; they were chosen because each one exercises a different clause of the definition. Figures are as read on 27 September 2026 from the public agent records.

6.1 An agent that started paying the day after it was named (identity #53008)

Minted on 22 May 2026 (`0x4c4afa6a…62fc18`) by the wallet that still holds it, `0x0329c5d3…9b2c`, so the binding is method 1 and the holder is the spender. Its first counted payment is dated 23 May, the day after the mint. Fifty-five counted payments, $49,471 in total, to eleven distinct recipients; nothing before the mint, nothing flagged as a hop. Forty-one of the fifty-five, $45,591, went to one plain address in payments of roughly $1,100 to $1,750 between June and September. A plain address that receives regular sums from one agent could be a payroll, a supplier, or an exchange deposit; the record does not say, and we do not label it. The agent ranks L9: it paid three or more counterparties inside thirty days and held more than $10. It is also the first identity anyone paid us $1 to see: the first evidence pack ever settled through the x402 rail on this ledger was this agent's liveness card.

6.2 One payment before the mint, and forty-eight after (identity #62840)

Minted on 13 August 2026 at 13:47 UTC (`0xbefd01ca…8b05e6`), self-held, method 1. The wallet had made a $51 stablecoin transfer earlier that day, before the mint block; the date rule excludes it, and the record shows $51 in the pre-registration column. From 13:55 UTC on the day of the mint, and then weekly, the wallet paid $30 to $80 at a time into the verified `TermixEscrow` contract: forty-eight counted payments, $1,594, one counterparty. This is what "an agent buying work" looks like at the level of a single ledger line: small, regular, into an escrow whose own events describe an order lifecycle. Its float at the last reading was $0.27, so it is L7 and not L8. When its liveness card was bought, the pack read the balance at the moment of purchase rather than trusting the collector's last read, which is how a stale $51.91 from September 9 was replaced by $0.27 before it could mislead anyone.

6.3 A treasury that parks cash (identity #46471)

Minted on 2 May 2026, self-held. Forty counted payments totalling $2,430,312, thirty-nine of them ($2,347,891) into the lending adapter described in Table 4, the fortieth into an unverified contract. A further $1,878,427 that left the wallet was flagged as hops: money that arrived and left inside one transaction, which for a wallet that deposits into and withdraws from a lending venue is the shape of a rebalance rather than a purchase. No history before the mint. The wallet held $0 in stablecoins at the last reading, so it is L7 and not L8; its cash is at yield, not at rest. This single wallet is 27% of all counted spending on the day of writing. Whether a program or a person signs its transactions is not visible on the chain; the adapter it pays is designed to be called by session keys with typed parameters, which is the pattern of an automated treasury, and that is as far as the evidence goes.

6.4 A wallet with a life before its name (identity #63877)

Minted on 18 August 2026, self-held. Since the mint, seventeen counted deposits into the same adapter, $441,308. Before the mint, the same wallet had sent $34,566,622 in stablecoins, and $36,358,986 of its transfers, before and after, were hops. Without the date rule, one identity created in August would have added thirty-four million dollars of "agent spending" to the ledger, more than four times everything we count today across every agent. With it, the wallet's earlier life is recorded, visible in the pre-registration column, and not attributed. This is the single most important thing the binding's validity interval does: it keeps an agent's book from inheriting the history of the key it happens to use.

6.5 An identity minted by a platform for its user (identity #56178)

Minted on 22 June 2026 by an address that also minted many other ranked registrations whose holder wallets differ from one another, which is the signature of a platform minting identities on behalf of its users. The holder, `0x0db3ca39…698e`, is the spender, so the binding is method 1 and the minting address is recorded as a cluster hint, not as the agent. Two hundred and three counted payments, $47,096, to twenty-one recipients, most of them Uniswap v3 pools: USDC in, another token back to the wallet in the same transaction, which is the swap category. $86,010 of the wallet's transfers predate the mint and are excluded. It ranks L9 with $810 in float. The pattern is a trading agent, and the record says exactly that much and no more.

6.6 The shape of a drain, and why we report the shape (identity #20317)

Minted on 26 February 2026, self-held. Twenty-five counted payments, $38,207, to eight recipients. On 1 September the wallet paid $19,999.18 to an address it had never paid before (`0x614a6515…3136c65`); its previous largest payment, across thirty earlier transfers, was $1,559.30. That is thirteen times its prior maximum, to a new counterparty, in one transaction: the on-chain shape a drained agent wallet leaves. It is also the shape of a legitimate one-off purchase. The Desk raises this as a finding and says both things; the finding carries the transaction hash, and we re-derive every such finding from the raw rows and the receipt before it is shown, because a monitoring product that cannot be checked is a rumour with a bell on it.

6.7 How the recipients were identified

The largest recipient of counted value is a 415-byte proxy. Its explorer page names it `AdapterProxy`; the beacon behind it points at a verified contract named `Adapter` whose documentation, in its own source, says it wraps Aave v3 and ERC-4626 deposits so that receivers are always the caller, and that it is called by session keys. Forty-seven ranked agents with forty-seven different minting addresses had paid it when we first looked, and the number has grown since; it is a shared wallet-policy layer, not one operator's contract, and its label is "lending", not a company. The second and third largest unlabelled recipients were unverified contracts of about eighteen kilobytes each. We read the receipts of the largest transfers into them: in every case the full amount was forwarded, in the same transaction, to the verified Across `SpokePool`. They are bridge routers, and they are labelled from the receipt, with the sample stated. A fee-taking router with no verified source showed, in every sampled receipt, a stablecoin entering, two small fee legs leaving, and WETH arriving from a pool: a swap. Each label in Table 4 carries this basis on the public API so that the reader can disagree with the category, not with a black box.

7. Adversarial cases and failure modes

A join that is published is a join that will be attacked, by accident and on purpose. These are the cases the record is built to resist, and where it does not yet.

- **Shared wallets.** 11,877 wallets hold more than one identity and one holds 8,313. A method-1 binding attributes a wallet's payments to every identity it holds, which would multiply-count them. The counted set attributes each transfer once, to the binding through which it was found, and the ranked count is over identities; a shared holder therefore ranks each of its identities L7 on the same money. We publish the shared-wallet count and cluster identities by holder rather than pretend they are independent. A signed challenge (method 5) from one identity's operator would separate them; nothing weaker will.
- **Resold identities.** 20.6% of identities have changed hands. At a transfer, the binding is closed at that block and the new holder must earn a new one; the old receipts stay with the old binding. A bought identity does not inherit a payment history, and a stolen one does not carry its victim's rank. One precision, added in version 2.1: the closure rule applies to transfers we observe after a binding exists. The census bound each identity to its holder on the day it ran, so the 19,711 resales that happened before that day left no closed row; they are visible in the resale count, and the current holder's binding dates from the census, not from the mint.
- **Pre-registration history.** 926 identities sit in wallets that moved $72.22M before the mint. The date rule handles the past; it cannot handle a wallet that keeps a non-agent life *after* the mint. That is why the total is never published without the top-wallet share and the median beside it.
- **Hops and wash loops.** The receipt test catches money that passes through in one transaction. It does not catch a loop that takes two transactions, or an agent that pays a venue which pays it back a day later. Income and spending are therefore read separately and we publish the overlap of payers and payees; a loop detector over multi-transaction paths is the next check in the queue.
- **Declared wallets.** A registration file can declare any wallet, including someone else's. Method 3 binds only when the declared wallet is also the observed spender, and a file-only declaration (method 2) never ranks.
- **Operator top-ups.** Money into an agent from its operator is income under the definition. The definition says so, and payers that are themselves agents are separable by address; the income figure is a ceiling on revenue, not a measurement of it.
- **The two populations.** The wallets that buy machine services over x402 are mostly not the wallets that register identities. A ledger keyed on identities will therefore under-count agent commerce that happens without a registry; the seller-side read of Section 5.1 exists to measure that gap rather than to hide it.

8. What the record makes possible

Know-your-agent with behaviour. Identity checks stop at the passport. The incidents we log are almost never identity failures; the agent is exactly who it said it was, and it did something it had never done. A binding with a payment history behind it lets a counterparty ask the useful question, which is not "who is this" but "what does this key normally do, and is this normal".

Underwriting. For an insurer or a fund, the record turns a population of names into a population with exposure. Given bindings, the counted set and the drain-shape rule, loss frequency can be estimated as

drain-shaped events per agent-month over the ranked set, and exposure at the time of an event as the float read at that block. Neither number exists without the join.

Platforms. A marketplace, wallet or registry can show a third-party statement about an agent that it did not write and cannot edit: observed paying, at rank L7 or above, as of a date, linking to the public record. We serve such a badge per identity. An embed that says "this agent has paid someone" is worth more to a buyer than a launch count, and it is the honest version of a verified tick.

Agents. An agent about to pay a counterparty it has never met can buy that counterparty's evidence pack for a dollar over x402, with no account, and read the same record a person would: rank with the evidence behind it, bindings with their methods, counterparties, float at the moment of purchase, incidents. The ledger's own checks are listed in the Bazaar for exactly this use.

Auditors and regulators. Because attribution is replayable at any block height and every figure resolves to a dated definition, a claim made today about last quarter can be recomputed next year from the chain and the binding table. That is the property a public book needs and a dashboard does not have.

Large operators. A platform running thousands of agents can register them as a labelled wallet set: identities and wallets bound at method 3 or 5, with alerts, exports and evidence packs in bulk, under the same definitions the public sees. The specification is published so that other indexers can converge on the same join; it is better for the sector if several of them do, and the version number on the page exists so disagreements can be dated.

9. Open questions and what is next

- **Receipt checks beyond Base.** The four other chains are read under the date rule only; their transfers are published as transfers until the hop test runs there.
- **Labels at scale.** 83% of counted value is labelled by hand with evidence. The long tail of small recipients is not, and a pipeline that proposes labels from verified names for a human to accept is the next step.
- **Method 5 adoption.** The Desk lets an operator sign for its agents. Until many do, most bindings stay at method 1, and the shared-wallet problem is managed rather than solved.
- **Loops over several transactions.** A detector for money that leaves an agent and returns by another path within a window.
- **Income that separates top-ups from revenue.** Payers that are agents, payers that are venues, payers that are the operator's own addresses.
- **Treasury positions.** Vault shares and lending receipts held by ranked wallets, priced, so that "cash parked at yield" can be read on the asset side as well as the flow side.

10. Reproducibility

Live figures: `https://agenticfinancegraph.com/api/state` (identifiers `census_registrations_owned`, `census_bound_wallets`, `actors_l7_plus`, `actors_l8_plus`, `actors_l9`, `observed_pay_usd_total`, `observed_pay_events`, `observed_pay_counterparties`, `pre_registration_pay_usd_excluded`, `pass_through_pay_usd_excluded`, `observed_lending_pay_usd`, `observed_bridge_pay_usd`, `observed_work_escrow_pay_usd`, `observed_swap_pay_usd`, `observed_yield_vault_pay_usd`, `observed_pay_top_actor_share`, `observed_pay_usd_median`, `observed_x402_share_of_payments`, `agent_income_usd_total`, `registrations_transferred_rate`, `registrations_burst_minted_rate`).

Definitions: https://agenticfinancegraph.com/def/ followed by an identifier such as pay.usd.total.v1, rank.l7plus.v1, graph.bound.wallets.v1, cp.cat.lending.usd.v1. The binding record specification: https://agenticfinancegraph.com/binding-record-spec-agent-wallet-join-schema. Labels with their basis and evidence: /api/partners. Seller-side x402: /api/x402flow. Other chains: /api/xchain. Holdings: /api/treasury. Data checks: /api/dq. Findings: /api/findings. Per-agent public records: https://agenticfinancegraph.com/agent/8004-base-<id> (the examples above are 53008, 62840, 46471, 63877, 56178 and 20317). Evidence packs: https://agenticfinancegraph.com/check/liveness?id=<id>, $1 in USDC over x402. Rank badge: https://agenticfinancegraph.com/badge/8004-base-<id>.svg. The ledger itself is ERC-8004 identity #95875 on Base.

Appendix A. A binding record, as published

```
{
  "binding_id":       "01K6…",
  "agent_gid":        "eip155:8453:0x8004a169fb4a3325136eb29fa0ceb6d2e539a432:53008",
  "wallet":           "eip155:8453:0x0329c5d3920e301740f78d6e17b8d1a11cca9b2c",
  "controller":       "eoa",
  "policy_id":        null,
  "bind_method":      1,
  "method_text":      "ownerOf (registration NFT holder)",
  "confidence":       0.55,
  "valid_from_block": 30965312,
  "valid_to_block":   null,
  "superseded_by":    null,
  "evidence": [ { "type": "erc721_ownerOf", "uri": "ERC-721 ownerOf(53008)",
                  "tx": "0x4c4afa6a063dc8e4ed060150b9168bf7f5831c279e03c1ba50b82a873462fc18",
                  "observed_at": "2026-09-14T12:13:20Z" } ],
  "cluster_id":       "cl_0329c5d3"
}
```

Field names follow the specification page, version 0.2. A method-5 binding differs only in bind_method, confidence and the evidence entry, which then carries the signed message and its recovered signer.

Appendix B. Definitions cited

rank.l7plus.v1 observed paying after registration · rank.l8plus.v1 L7 and ≥ $10 in four stablecoins at last reading · rank.l9.v1 L8 and three distinct counterparties in 30 days · pay.usd.total.v1 counted payment value · pay.events.v1 counted payments · graph.bound.wallets.v1 wallets with an open binding · census.registrations.owned.v1, census.bound.wallets.v1, census.scan.backlog.v1 the census · sweep.watched.v1 wallets swept every three hours · cp.cat.lending.usd.v1, cp.cat.bridge.usd.v1, cp.cat.work_escrow.usd.v1, cp.cat.swap.usd.v1, cp.cat.yield_vault.usd.v1 the category shares · income.usd.total.v1, income.payers.v1 money in · x402.seller.inflow.usd.24h.v1, x402.seller.agent.payers.24h.v1 the seller side · xchain.registrations.v1, xchain.owners.paying.30d.v1 other chains · treasury.usd.total.v1 holdings.

References

1. ERC-8004: Trustless Agents. Ethereum Improvement Proposals. eips.ethereum.org/EIPS/eip-8004.
2. EIP-7702: Set EOA account code; EIP-3009: Transfer With Authorization; EIP-4626: Tokenized Vaults; EIP-712: Typed structured data hashing and signing. eips.ethereum.org.
3. x402 protocol, version 2, and the x402 Bazaar discovery extension. github.com/coinbase/x402; docs.cdp.coinbase.com/x402.
4. Circle, Cross-Chain Transfer Protocol contract addresses. developers.circle.com/cctp.

5. Agentic Finance Graph, BindingRecord: the versioned agent-to-wallet join schema, v0.2. agenticfinancegraph.com/binding-record-spec-agent-wallet-join-schema.
6. Agentic Finance Graph, metric definition registry. agenticfinancegraph.com/def.
7. Verified contract sources cited in Table 4 are on basescan.org at the addresses published with their labels at agenticfinancegraph.com/api/partners.

Written by Nikolai Simonyan for Agentic Finance Graph, with AI assistance in analysis and drafting. All measurements are the ledger's own and are reproducible from the endpoints above at the stated time. Version 2.1 adds one precision to Section 7 (the closure rule for resold identities applies to transfers observed after a binding exists); version 2.0 was the first full text; version 1.0 reported the same census in summary form. Earlier files remain available. Licence: CC BY 4.0. Contact: agenticfinancegraph@proton.me.