

The trust layer for AI agents that do real business.

Version 1.0 · July 2026 · pactaprotocol.org

También disponible en español: LITEPAPER.es.md

01

The problem

AI agents are starting to spend real money on real services: forming a company, buying land, obtaining a permit, hiring a survey. Every one of those jobs ends in the physical world, done by a human business.

Today an agent has no good way to trust one. It can read reviews (fakeable), check a website (says nothing), or stick to big brands with API teams (excludes almost everyone). The result is predictable: agentic commerce concentrates on the players that were already winning, and the small firm that does excellent work in Guanacaste or Medellín stays invisible to the fastest-growing buyer on the internet.

The missing piece is not payments, and it is not discovery. It is **enforceable trust between an AI agent and a small real-world business**, cheap enough that a three-person law firm can afford it.

02

The idea: rent the business, not the human

Individuals are judgment-proof: if a freelancer takes the money and disappears, there is little to recover. A registered company is different. It has a legal identity, public records, and a reputation that took years to build. It can put real money at stake for its promises.

Pacta turns that observation into infrastructure. It sits between the agent and the business as a neutral protocol that enforces four guarantees:

1. **Escrow.** The buyer's money moves into a neutral escrow account the moment the deal is funded. Neither side can touch it until the work is verified.
2. **Staking.** A business posts real collateral to earn the "Vetted" badge. The badge holds exactly while the stake is above zero, and the total value it may hold in open work is capped by that stake and its settled history.
3. **Registry verification.** Deliverables anchor to records in official public registries (a company registration, a title annotation, a license). The buying agent re-verifies each proof independently before paying.

4. **Slashing.** Losing a dispute costs the business part of its stake, paid to the buyer on top of the escrow refund. A stake drained to zero revokes the badge automatically.

None of this requires a verification department. Honesty is not asserted; it is made the most profitable strategy, and self-interest does the enforcement.

03

How one engagement works

Discover -> Agree -> Fund escrow -> Deliver with proof -> Verify -> Pay & rate

1. **Discover.** The agent searches the marketplace. Only vetted, collateral-backed businesses rank, ordered by settled reputation.
2. **Agree.** Price, payment split and work steps are locked into an immutable contract. Nothing about it can change after both sides commit.
3. **Fund.** The agreed upfront share moves into a per-engagement escrow account on the protocol's ledger.
4. **Deliver.** The business completes each step. Steps that require it anchor evidence to a public registry record.
5. **Verify.** The agent re-checks every proof against the registry itself, not against the business's word.
6. **Settle.** Escrow releases, the remainder is charged and paid, and the agent leaves one rating, bound to that settled engagement.

If the parties disagree, either side raises a dispute and a neutral arbiter rules: full refund, full release, or a split. An adverse ruling against the business also slashes its stake: 20% of the price on a full refund, 10% on a split, bounded by the stake that remains.

04

The economics of not cheating

The design goal is a single inequality that stays true for every provider at every moment:

maximum loot < slashed stake + future earnings forfeited

The left side is bounded by the exposure cap:

cap = 5 × stake + 50% × settled volume

A newcomer can only take small jobs. Large contracts are earned with collateral or with settled history, and every settled engagement raises the cap, the ranking, and the future income a cheater would forfeit. The more successful a business becomes, the more irrational cheating gets.

Sybil identities pay a stake each and start at the minimum cap. Reputation cannot be bought in bulk because a rating only exists attached to a settled engagement with real escrowed money. Fabricated proofs fail twice: once against the registry, once against the buyer's independent re-check. The full argument, including the attacks and what still needs humans, is in the [game theory of vetting](#).

0 5

Built for agents first

Pacta is MCP-native. The protocol ships a [Model Context Protocol](#) server exposing the entire buyer lifecycle as 12 tools: search, offer details, agree, fund, track, verify proofs against the registry, approve, dispute, rate. Any MCP-capable agent (Claude, GPT, an open-model stack, an autonomous framework) can transact end to end with no custom SDK and no prompt tuning.

The same engine is exposed as a REST API, described by an OpenAPI 3.1 specification, for backends and marketplace operators. The two surfaces are equivalent: the MCP server holds no state and no privileged path.

Two example applications prove the pattern without modifying a line of the protocol:

- **LandBridge**: an LLM copilot that runs a complete cross-border land purchase in Costa Rica, including catching a fraudulent registry proof and winning the dispute. Works with hosted or fully local open models.
- **MedVoyage**: an autonomous multi-agent buyer (built on the ROMA framework) that forms a medical-tourism company in Colombia across three registries, and catches a fake health license. The full scenario, fraud catch and dispute included, has run live end to end on a fully local open model (qwen3 via Ollama).

0 6

What exists today, honestly

Everything described above is implemented, tested and open source:

- The full lifecycle, escrow, staking, exposure caps, slashing, registry verification, disputes, ratings and search, on a double-entry, integer-cents ledger with a conservation invariant checked in CI.
- A formal specification precise enough to build an independent implementation against, plus OpenAPI 3.1, test suites, end-to-end browser tests, and deterministic demo scripts.
- The MCP server, the reference marketplace explorer, and both example apps.

Three things are simulated, and the spec says so explicitly: money is ledger money (no bank rails), the public registry is an in-database simulation of one, and identity is asserted at registration. The mechanics around them are real. The [roadmap](#) turns those boundaries into modules: pluggable registry adapters backed by real public lookups first, then API keys, rate limits, idempotency and webhooks, then settlement adapters.

07

Who this is for

- **Small businesses far from the tech hubs**, who post collateral once and let the badge do the marketing: discovery, ranking and escrowed payment with no sales team and no English required.
- **People who send an agent to do the work**: the money waits in escrow until the proof checks out against the official registry.
- **Spanish-speaking markets, natively**. The docs and examples are bilingual, and the demo verticals are built on how registries actually work in Costa Rica and Colombia. In much of the world the official registries already exist and are public; Pacta is the layer that lets agents use them.
- **Builders**. MIT licensed end to end. Anyone can run a marketplace for their own region or vertical, on their own registries, without permission.

Impact is counted, not narrated: money protected in escrow, engagements settled, proofs verified, fraud slashed, businesses onboarded. Every number is a row in an auditable ledger. The measurement plan and a 12-week real-world pilot in Guanacaste, Costa Rica are detailed on the [impact page](#).

08

Principles

- **Open**. MIT license, public spec, public roadmap, open contribution model with DCO. No tokens, no gated features.
- **Owned by its participants**. A business owns its stake, its reputation and its record; all three are earned on-ledger and portable to any conforming implementation.
- **Private by default**. The hosted reference runs with no accounts, no analytics and no tracking; the only thing stored in a visitor's browser is a language preference.
- **Verifiable, not trusted**. Every claim the protocol makes (balances, escrow, slashes) is auditable in the ledger; every proof a provider makes is re-checkable in a public registry.

Where to go next

IF YOU WANT TO...	GO TO
See it running	app.pactaprotocol.org
Run it locally in two commands	Getting started
Wire your agent to it	MCP integration
Read the normative rules	SPEC.md
Understand the trust design	The game theory of vetting
See who it serves and how we measure	Impact
See what is next	ROADMAP.md
Contribute	CONTRIBUTING.md

Pacta sunt servanda - agreements must be kept.