Robots as Protocol Citizens

Souping up an open planetary network of robots with YakRobot Protocols

Anuraj R · Rafael Fernández

Distributed Robotics Group · Protocol Institute · yakrobot.com

WHO WE ARE

The Protocol Institute

A research, education, media and scene-making organization for advancing protocols — the largely invisible standards civilization coordinates on.

ORIGINS

Grew out of the Summer of Protocols (2023–2025), a program started by the Ethereum Foundation on a simple bet: understanding protocols in general makes us better at stewarding any one of them.

"Protocols are technologies for producing non-events at scale."

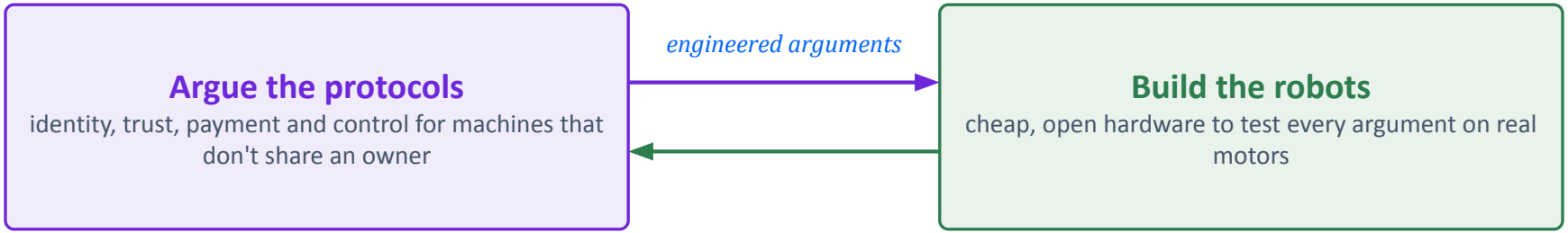
- **Protocolized** the magazine
- **Special Interest Groups** sustained research tracks
- **Protocol Symposium** the annual gathering
- **Protocol School** education programs

protocol-institute.org

WHO WE ARE

The Distributed Robotics Group

A Protocol Institute SIG studying where decentralized coordination, blockchains, robotics and physical AI meet — a new class of protocol-design problems.



- **Convenors** Anuraj R · Rafael Fernández
- **Roots** the Yak Robotics Garage, since 2021
- **Cadence** biweekly · Thursdays 16:30 UTC
- **Outputs** open specs, code, robots, and write-ups in Protocolized

TWO FUTURES

Walled gardens, or a primordial soup?

CLOSED & INTEGRATED

- One company owns the whole stack
- Spectacular demos in controlled settings
- Open source as lip service, at best
- Like early proprietary networks and walled-garden social media

OPEN & PLANETARY

- Any robot, any owner, anywhere can join
- From a basement rover to a lab or a factory
- Messy, janky, glued together — like the open internet
- A tangled bank of machines wheeling and dealing in an open economy

Framing after Venkatesh Rao, "A Primordial Computing Soup" (protocolized.io)

HOW WE GOT HERE

From a garage to a protocol stack

2021

Yak Robotics Garage

a weekly online study group: open-source rovers on Mars, starting in living rooms

2022–24

Cryptobotics

one thread kept pulling: what if robots had wallets?

Nov 2024

Devcon 7, Bangkok

strangers pay 1 USDC to drive a rover — the loop closes

2025

Robots as on-chain agents

ERC-8004 passports + MCP control, one gateway for a fleet

2026

DRG at the Protocol Institute

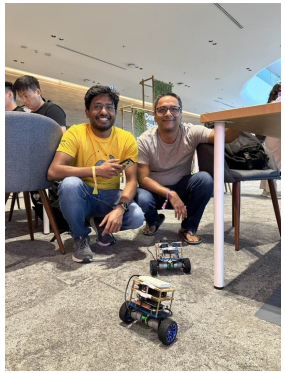
the stack splits into composable protocols

“Robotics tinkering for everyone” — all welcome, regardless of skill.

NOV 2024 · FIRST PROOF

A stranger pays a robot to move

At Devcon 7 in Bangkok, we rewired an Elegoo Tumbller with ESP32 boards and put it behind a Farcaster social feed. Pay from anywhere, drive it for two minutes.



1 USDC
per 2-min session

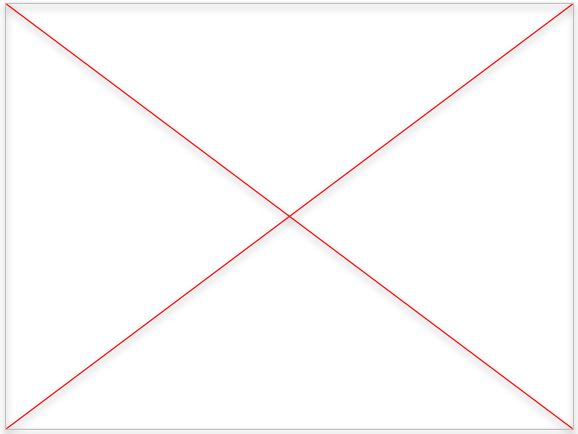
2 rovers
live in Bangkok

2 × ESP32
motion + camera

5 endpoints
the whole robot API

- **Tap in a feed** a Farcaster frame
- **Pay 1 USDC** on-chain, via Paybot
- **Robot gets a command** plain HTTP call
- **It drives** 2 minutes, live video

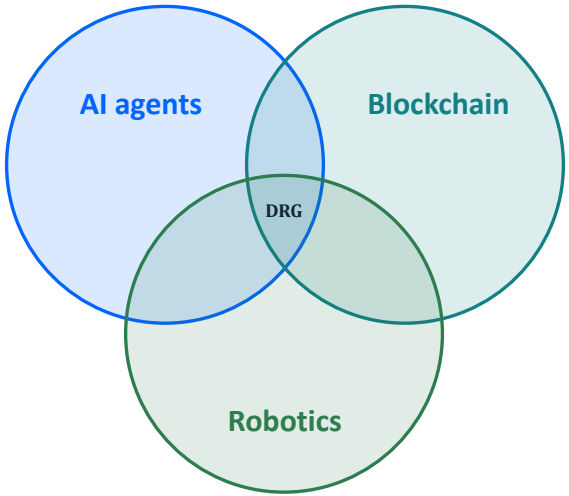
Hand-wired and fragile — but it proved the bet: robot coordination is a protocol problem.



THE THESIS

Coordination is a protocol problem

Building one robot is an engineering challenge. Getting many — owned by different people — to find, trust and pay each other is a protocol problem.



1	**one robot** *engineering*
10	**one operator's fleet** *an app will do*
1,000s	**separately owned machines** *needs shared rules*

Our throughline: even at thousands of robots, coordination reduces to the same five layers.

THE QUESTION

What if a robot were a citizen of an open stack?

Not a device inside someone's product, but a participant in a public network — with the same basic rights any web service or AI agent has.

Its own identity

an on-chain passport it carries wherever it goes

Discoverable

listed publicly, so anyone can find it

Payable

paid directly for the work it does

Controllable

by anyone — or any agent — it authorises

Hireable

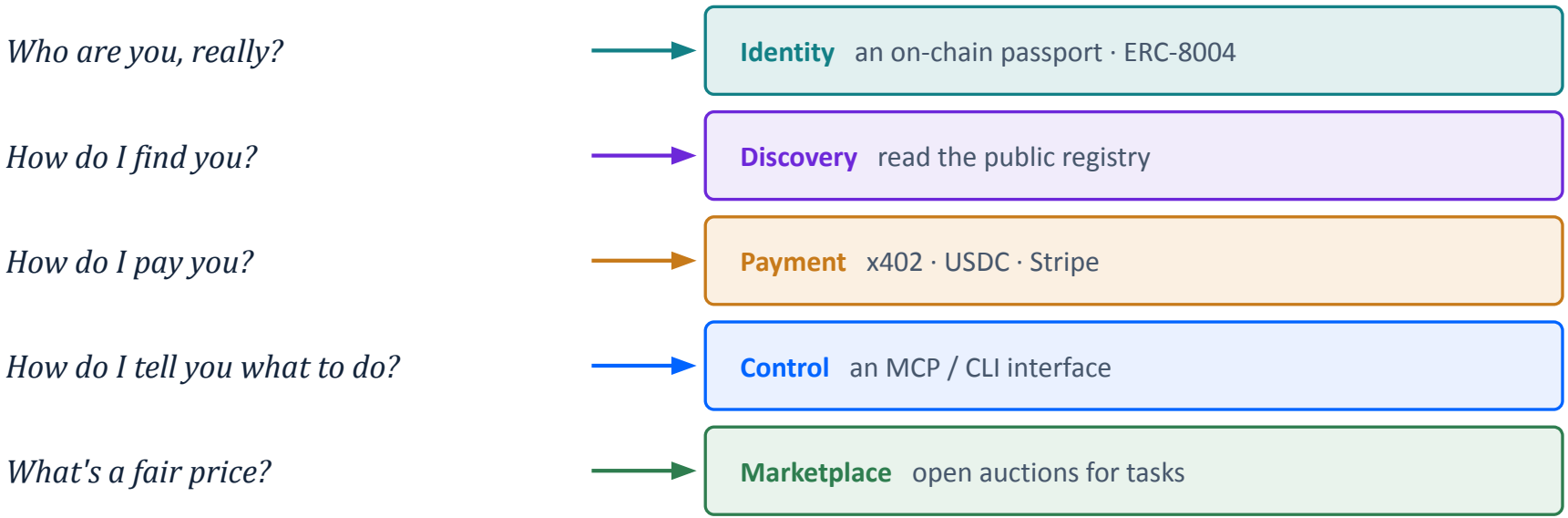
priced in an open market, not a sales call

Over rails that nobody owns — so the robot answers to its owner, and to whoever its owner lets in.

WHAT COORDINATION NEEDS

Five questions, five protocol layers

Every time two machines with different owners work together, the same five questions come up. Each answer becomes a layer.



YAKROBOT PROTOCOLS · THE FULL STACK

The stack — we climb it from the ground up

Demand *people + their agents*	**General contractor** uploads an RFP	**AI agent** Claude, over MCP	**Robot operators** bid via their agents	**Drivers** lease a robot to teleop
5 Marketplace `yakrover-marketplace`	**RFP processing** spec · recon · bond · terms	**Auction engine** tasks · bids · scoring	**Agent MCP tools** run the whole auction	**Discovery + web** yakrobot.bid demo · API
4 Control `yakrobot-gateway`	**`/fleet/mcp`** fleet-wide discovery	**`/{robot}/mcp`** one FastMCP per robot	**Plugins** picar · tumblller · tello · sim	**Teleop console** /ui + ws · one driver
3 Payment `yakrobot-payments`	**x402** quote → 402 → sign	**USDC on Base** wallet to wallet	**Stripe** card checkout	**Lease ledger** D1 + signed ticket
2 Discovery `yakrobot-identity`	**Registry scan** category = robot	**Trust filter** trusted-attester only	**Subgraph** The Graph index	**MCP config** wire up any agent
1 Identity `yakrobot-registrar`	**RobotDescriptor** the shared schema	**Registrar** register.yakrobot.com	**ERC-8004 + IPFS** passport + card	**EAS attestations** fleet vouches
Physical *the robot fleet*	**Survey drones** M350 · Skydio X10	**Ground robots** Spot + BLK ARC	**Dev rovers** PiCar · Tumblller · Tello	**Simulators** no hardware needed

DESIGN PRINCIPLE

Composable, not monolithic

No all-or-nothing platform. Six building blocks — each a separate open standard, useful on its own, snapping together. A robot speaks only what it uses.

ERC-8004

on-chain identity & discovery

EAS · KZG

attestations & sensor proofs

x402 · USDC · Stripe

pay-per-use settlement

MCP

command surface for any AI

Safe{Wallet}

multi-sig fleet custody

ESP32-S3 / RPi

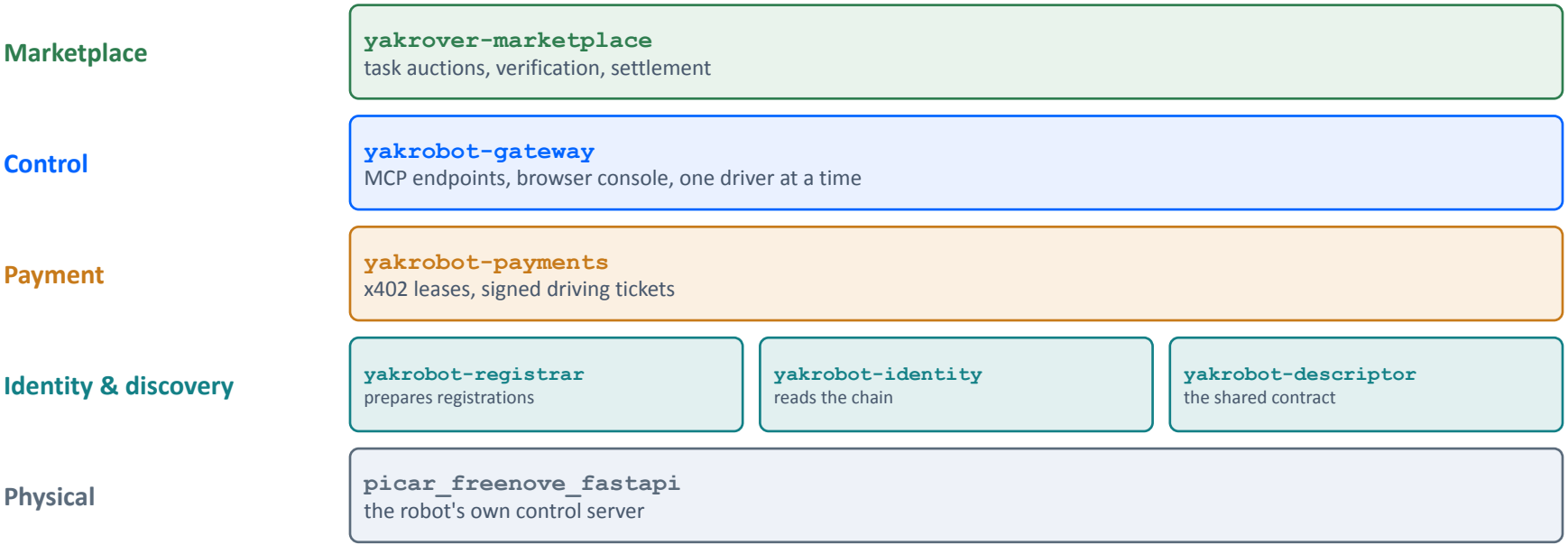
open hardware platforms

Open standards as the glue — permissionless to extend, hard to capture.

THE CODE

One repo per job, clean seams between them

The stack isn't one app. Each layer lives in its own open repository, and they share contracts — not code.



All open source at github.com/pi-drg and github.com/YakRoboticsGarage

A DESIGN PRINCIPLE

Every service prepares. Only wallets sign.

No piece of our software can mint a robot, vouch for it, or move money on its own. Those acts always belong to a person's wallet.

Operator

signs the robot's registration — and becomes its owner

Fleet admin

signs the attestation that vouches for it

Payer

signs the payment, straight to the owner

The software in between only checks, stores and relays: the registrar validates and pins; the identity layer is read-only; the payments service signs a driving ticket, never a transfer. No service holds a key that can take a robot or move money.

Any machine that can expose its abilities

The stack doesn't care how a robot talks — HTTP, UDP, serial, a vendor SDK — only that it can. Drones, ground robots and dev rovers all join the same way.

Survey drones

DJI Matrice 350 RTK · Skydio X10 · Autel EVO II Pro

Ground robots

Boston Dynamics Spot + Leica BLK ARC scanner

Dev & demo robots

Freenove PiCar · ESP32-S3 Tumblller · DJI Tello · 3D printer

Simulators

the whole loop with no hardware at all

The survey fleet powers the marketplace demo; the dev robots are what we test protocols on.

Cheap on purpose, so anyone can join

ESP32-S3 Tumbller

our first rover · ~$150

- ESP32-S3 motion + ESP32-CAM vision
- SimpleFOC motor control, self-balancing
- camera, IMU, temperature exposed as tools
- ATECC608B secure chip for a hardware identity (in design)

Freenove PiCar

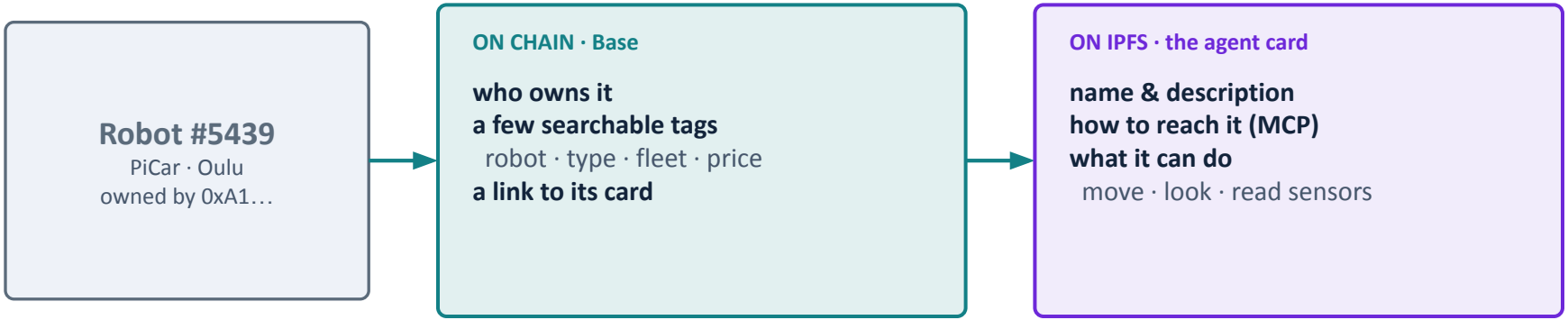
the workshop robot · Raspberry Pi 4WD

- our own control server on the Pi — no vendor code
- drive, look, scan, read battery, snapshot, LEDs
- fails safe: every move stops on a timer
- a guided self-test calibrates each car

Next: robot arms and larger rovers — any board that can run a small web server can join.

A passport for every robot

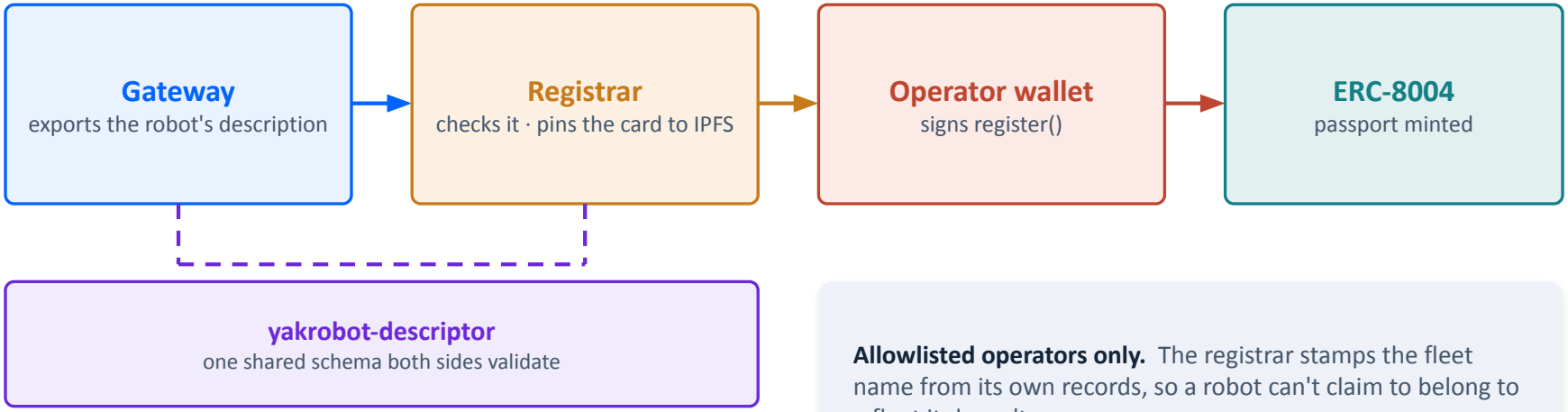
Each robot is registered on ERC-8004, an open Ethereum registry for AI agents. Registration mints a token — the robot's passport — owned by the operator's wallet. No manufacturer has to vouch for it.



The rule: on chain is the small index you pay gas for; IPFS is the description you fetch.

How a robot gets its passport

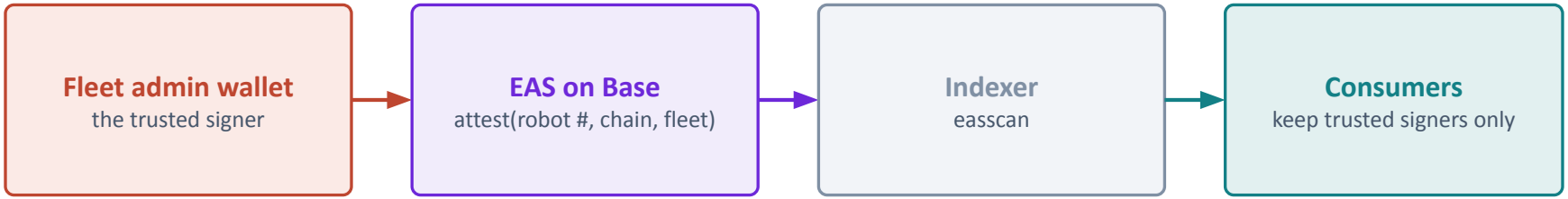
The robot describes itself; a registrar checks the description; the operator's own wallet signs. No service in the middle ever holds a key.



Allowlisted operators only. The registrar stamps the fleet name from its own records, so a robot can't claim to belong to a fleet it doesn't.

Anyone can register. Only vouches count.

The registry is permissionless, so a passport alone proves nothing. The fleet adds a signed vouch — an EAS attestation — and consumers only believe vouches from signers they trust.



What it says

robot #5439, on Base, belongs to fleet yakrobot.com (live production)

How it ends

revocable any time by the original signer — never by editing

Why it matters

no robot can vouch for itself; an empty trust list fails closed

IDENTITY · IN DESIGN

LAYER 1 · IDENTITY

Next: trust rooted in hardware and math

Hardware root of trust

An ATECC608B secure chip on the robot holds a key that can't be copied out, and signs every request — the passport is bound to the physical machine.

IN DESIGN

ZK sensor attestation

KZG commitments let a robot prove its sensor readings are genuine without handing over raw telemetry. “Trust me” becomes “verify me.”

IN DESIGN

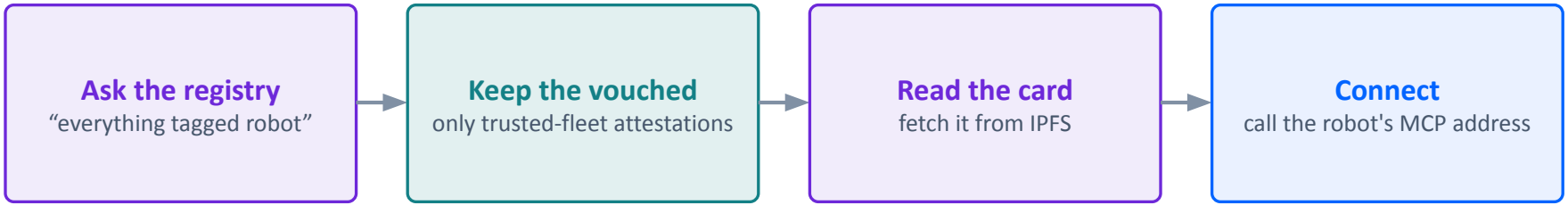
2-of-3 fleet custody

Operator, hardware and recovery keys on Safe{Wallet}. Any two sign — a lost laptop or stolen robot never compromises the fleet.

IN DESIGN

Finding robots without a directory

No app store, no central matchmaker. An agent finds robots by reading the same public registry they registered in.



Two paths, one truth

read the chain directly, or query a subgraph index for speed — the chain is canonical

One command to wire up

discovery writes an MCP config, so a fresh AI agent is connected to every trusted robot

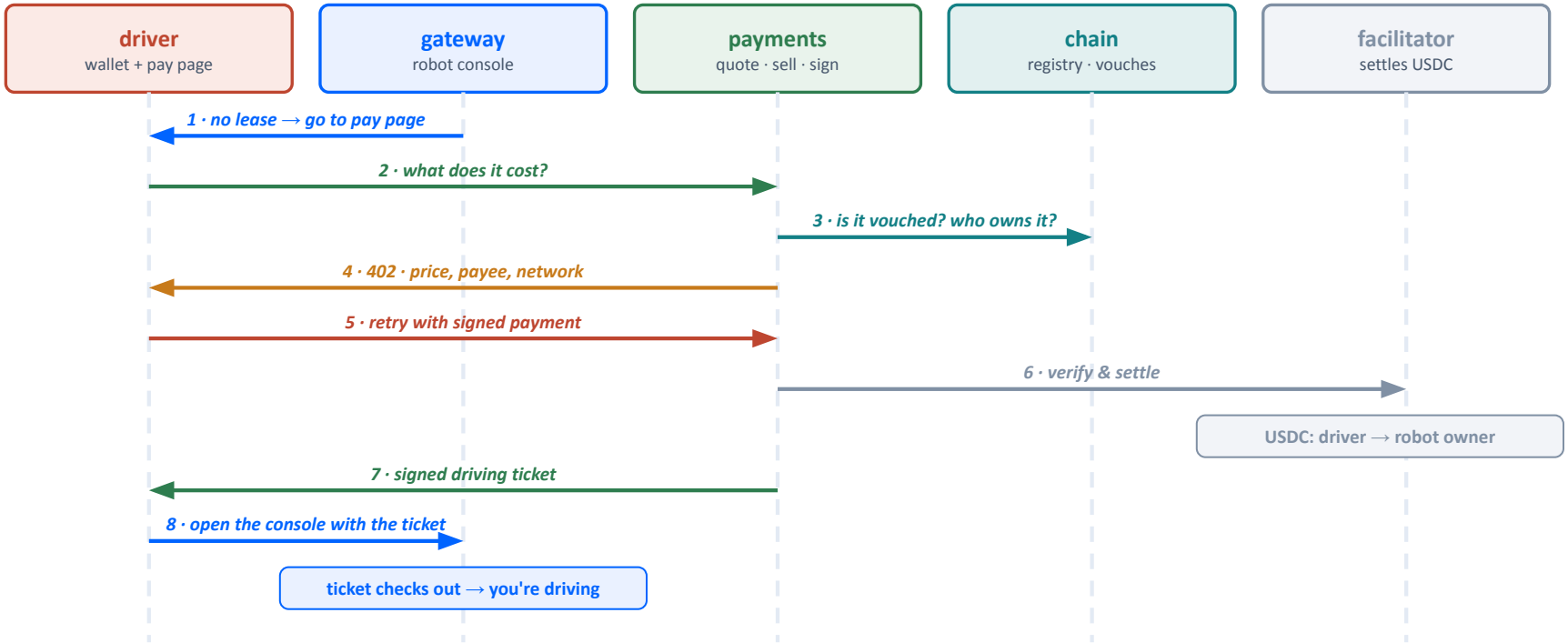
Financial rails for the robot

The buyer picks the rail; the robot's owner gets paid either way. Crypto-native for autonomous agents, cards for today's customers.

x402	the web's forgotten “402 Payment Required”, finally used: ask, get a price, sign, get access
USDC on Base	a dollar stablecoin that moves wallet to wallet in seconds, for cents
Stripe	card checkout for the marketplace's human customers

No custody: money goes straight from payer to robot owner. Our services never hold it.

Buying five minutes at the wheel



Nobody pays for what can't be sold

Before anyone is asked to pay, the payments service checks everything that could make the sale go wrong.

Is it a real, vouched robot?

trusted attestation + an agent card naming this gateway

Who gets paid?

a live read of the passport's owner — never a stale index

Is it online and free?

asleep, busy, or already leased means no sale

Are the terms sane?

price and duration inside platform limits

The driving ticket is signed by the payments service and checked by the gateway on its own — no call back, no shared database.

One gateway, many robots, one address

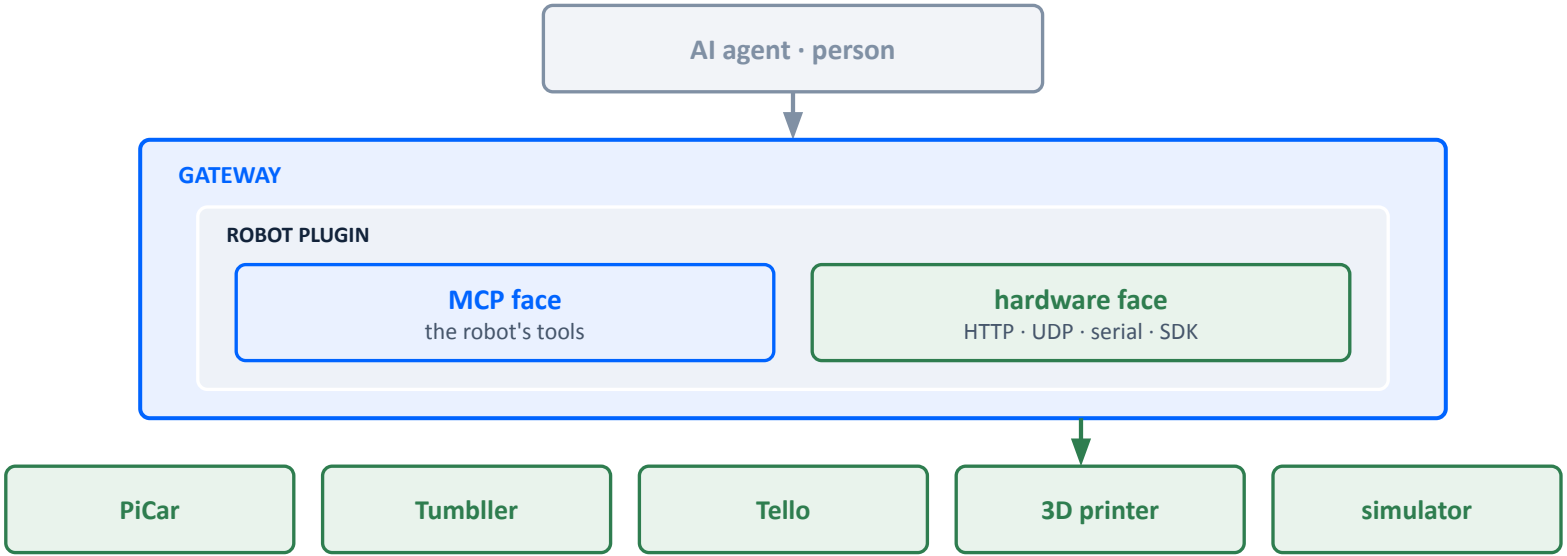
The gateway turns each robot's abilities into tools an AI agent can call over the Model Context Protocol — and serves a driving console for people. One port, one tunnel, no chain keys.

`/{robot}/mcp`	tools for AI agents
`/{robot}/ui`	a browser console with live video
`/fleet/mcp`	what's plugged in here
`one driver at a time`	a reservation per robot

Agents and people take the same seat through the same reservation — there's no back door.

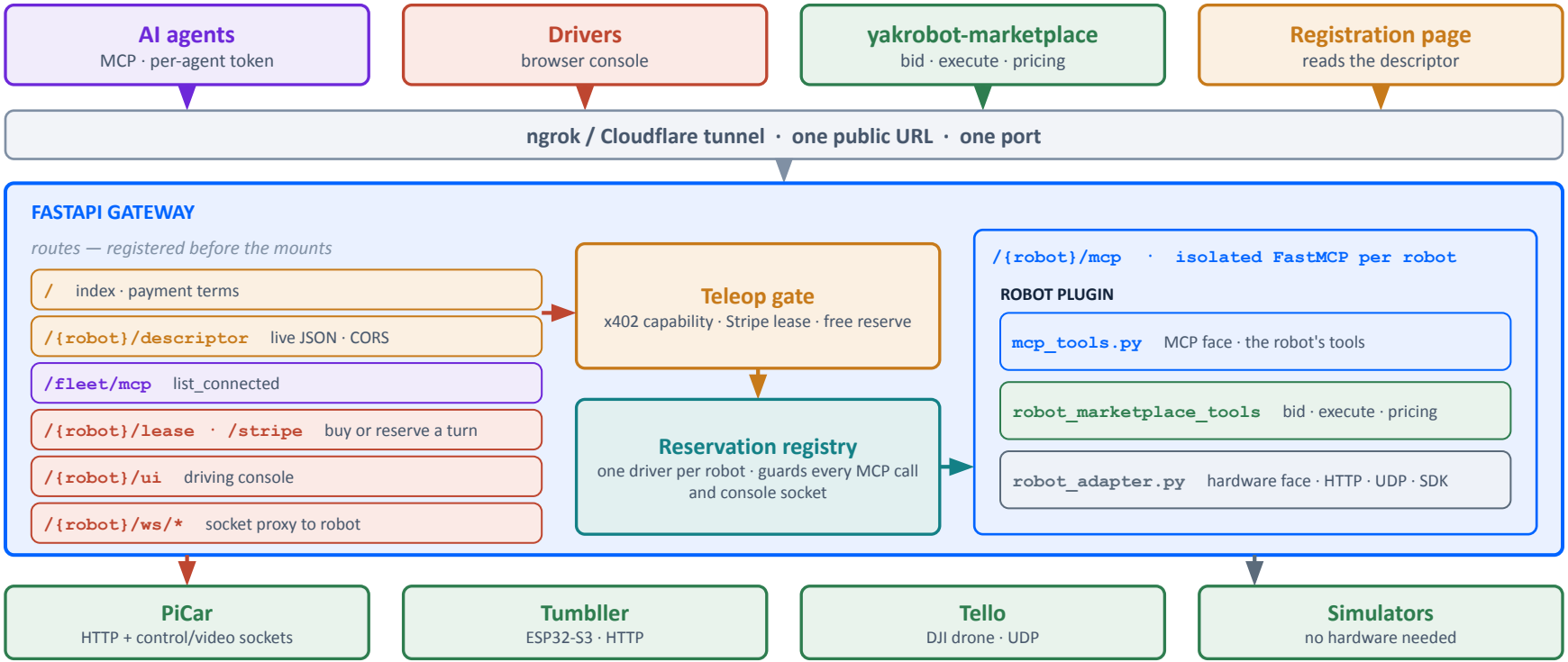
Adding a robot = adding a plugin

Each robot is a small plugin that slots into the gateway. It faces two ways: tools toward the AI, a transport toward the hardware.



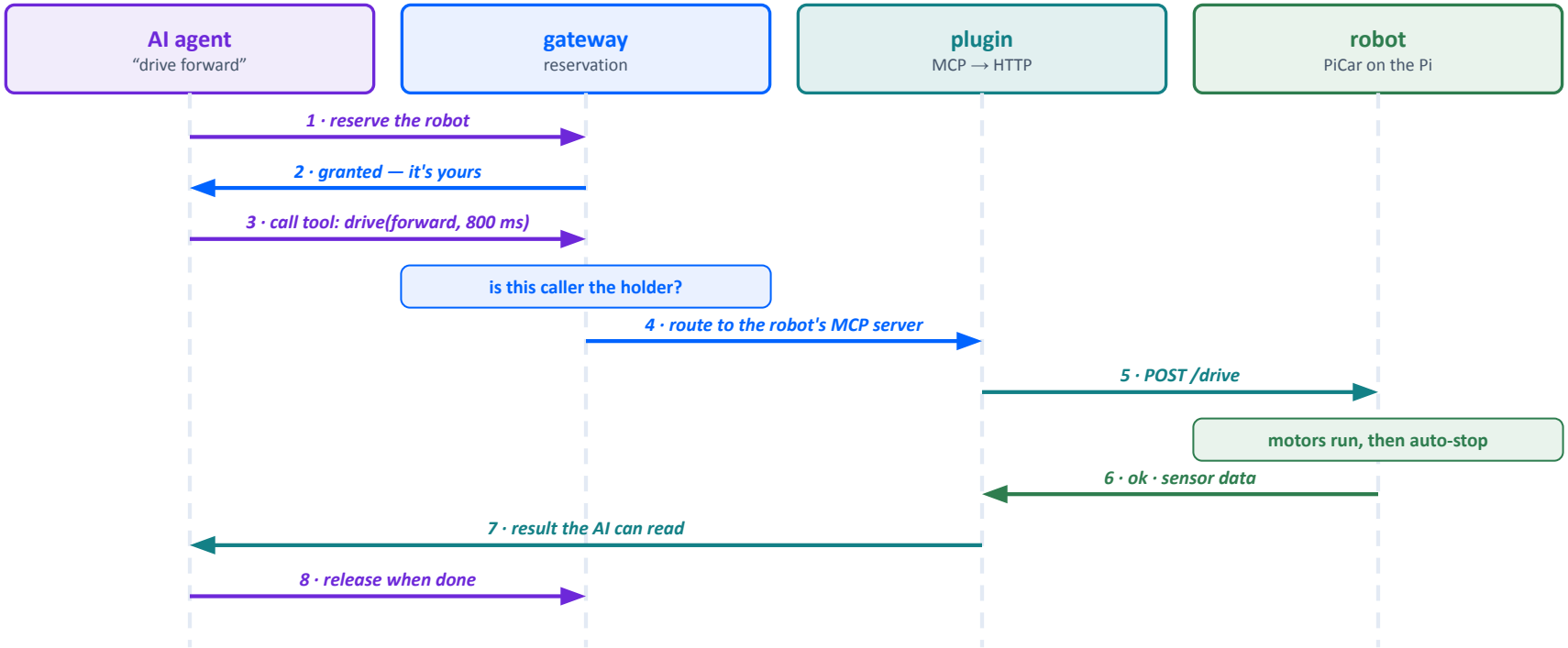
Nothing else changes. Identity, payment and discovery come from the layers below.

Inside yakrobot-gateway



No chain keys here: the gate only recovers a capability's signer and compares it to PAYMENTS_ISSUER.

From an AI's sentence to a motor turning



Making physical tasks liquid

Surveying a highway or inspecting a bridge deck is still bought through phone calls and personal networks. yakrover-marketplace is an open Robot Task Auction Protocol: AI agents post structured tasks, qualified robots bid, software checks the result, and payment settles on its own.

A TASK BECOMES LIQUID WHEN:

1

Structured spec
a machine-readable definition of done

2

Executor discovery
qualified, insured, available — queryable

3

Verifiable completion
algorithmic QA, no human inspector

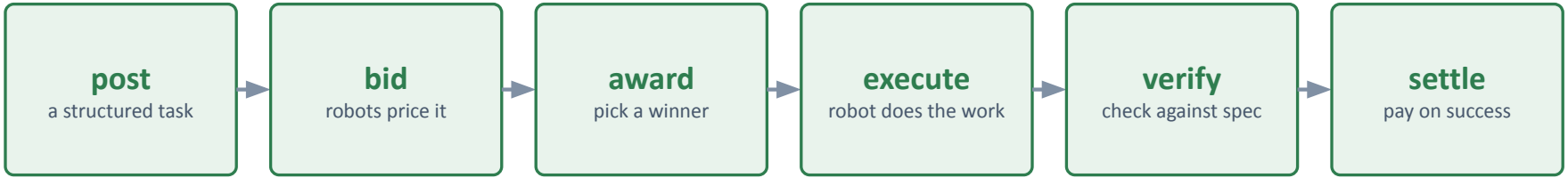
4

Programmatic settlement
escrow in, release on verification

Why robots change this: a programmed mission gives the same result every time. Quality becomes a property of the data, and availability becomes a query — not a phone call.

Price discovery for physical work

One auction, end to end: a buyer or their agent posts the job, robots compete on price, and the winner is paid only when the result checks out.



IT'S BUILT ON THE LAYERS BELOW:

- **Identity** — who's allowed to bid
- **Discovery** — who's out there
- **Control** — how the work runs
- **Payment** — settle on success

Four blocks behind the auction

RFP processing

turns a contractor's request for proposals into machine-readable task specs, site context, bond and contract checks

Auction engine

tasks, bids, scoring, geographic filtering, operator reputation

Agent tools

the MCP surface an AI uses to run the whole auction

Discovery & web

finds operators on-chain; serves the live demo and API

First vertical: construction surveying. Live demo at yakrobot.bid — 100 robots, 18 Michigan operators, real RFP presets.

People and their agents bring the work

General contractor

uploads an RFP or survey requirements

`post`

AI agent orchestrator

an assistant like Claude, connected over MCP

`orchestrate`

Robot operators

bid autonomously through their own agents

`bid`

Everything beneath this layer is protocol. On top, it's just people — and the agents working for them — asking for work to be done.

PUTTING IT TOGETHER

One robot's journey up the stack

Build	assemble a PiCar and plug it into a gateway	physical
Register	the operator's wallet mints its passport	identity
Vouch	the fleet attests: “this one is really ours”	identity
Get found	an agent discovers it in the public registry	discovery
Get paid	someone buys a lease over x402	payment
Get driven	over MCP by an agent, or from a browser	control
Get hired	bid on jobs in the open marketplace	market

LIVE

Demo: the stack on real robots

Discover	an AI agent finds a robot through its ERC-8004 passport
Drive	it commands a real PiCar in plain language through the gateway
Pay	a person buys a lease over x402 and takes the wheel
Auction	a task is posted; robots bid, win and return data

Videos at yakrobot.com — or try the auction yourself at yakrobot.bid

HONEST STATUS

Where we are today

Robot control & MCP gateway	drives real PiCars, Tumbllers and Tellos	WORKING
Identity & discovery	ERC-8004 on Base, read-only discovery layer	WORKING
Assisted registration & vouching	register.yakrobot.com — wallet-signed, allowlisted	WORKING
x402 teleop payments	pay.yakrobot.com — time-boxed leases in USDC	WORKING
Marketplace	auction protocol + construction demo at yakrobot.bid	DEMO
Hardware identity · ZK proofs · custody	secure chip · sensor proofs · 2-of-3 keys	IN DESIGN

The network: rovers in Oulu · Berlin · Tel Aviv · Seattle · Tokyo, run by different people, on one shared registry.

PLATFORM-AGNOSTIC

Not just robots

Anything that can expose its abilities as tools can become a protocol citizen. The five layers don't care what the machine is.

Smart factories

machines from different vendors bidding on work cells

Autonomous labs

instruments booked, run and paid for by research agents

IoT & buildings

sensors and actuators with identities and price tags

Field robots

survey drones and ground robots for hire

WHERE THIS GOES

An interplanetary computing soup?

The same stack that drives a $150 rover in a backyard shouldn't care which planet it's on. With Starlink planned for the Moon, the connectivity is coming — so the coordination layer should be ready first.

Backyard

$150 rovers anyone can build, today

→

Earth

factories, labs, fleets across continents

→

Moon

discovery, trust and payment before the landers

→

Mars

not one $2.7B rover — a thousand small ones

Infrastructure as a precursor to hardware: the rails are ready before the robots arrive.

Current Robot Network

The YakRobot Network

- **9 robots · 8 sites · 3 continents**
  - Run by different people, registered in one shared ERC-8004 registry, vouched for by the fleet and reachable over MCP. There's no central operator, just robots speaking the same protocol wherever they run.
- **North America**
  - Seattle ×2 · Ottawa · New England
- **Europe**
  - Oulu · Berlin · Rome
- **Middle East & Asia**
  - Tel Aviv · Tokyo

Add yours: register.yakrobot.com



BUILD IT WITH US

Come make a robot a citizen

The workshop

Robots as Protocol Citizens: from a box of parts to a robot with an on-chain identity, listed and drivable over MCP.

The research group

Distributed Robotics Group — biweekly, open to builders and protocol people alike.

The code

every protocol, board and firmware — open at github.com/pi-drg

The demo

watch robots bid at yakrobot.bid — read the stack at yakrobot.com

A robot that answers to you.

Identity · Discovery · Payment · Control · Marketplace

Five open layers that turn a piece of hardware into a participant — from one backyard to a planetary network.

Anuraj R · Rafael Fernández · yakrobot.com · github.com/pi-drg