The ∃/∃ Framework

Agentic payments and building a shopping assistant



Colton Padden

Member of Technical Staff, Vercel

@coltonpadden @cmpadden /in/colton-padden

Agents are the internet's next users
and they will pay on our behalf

A NEW ERA

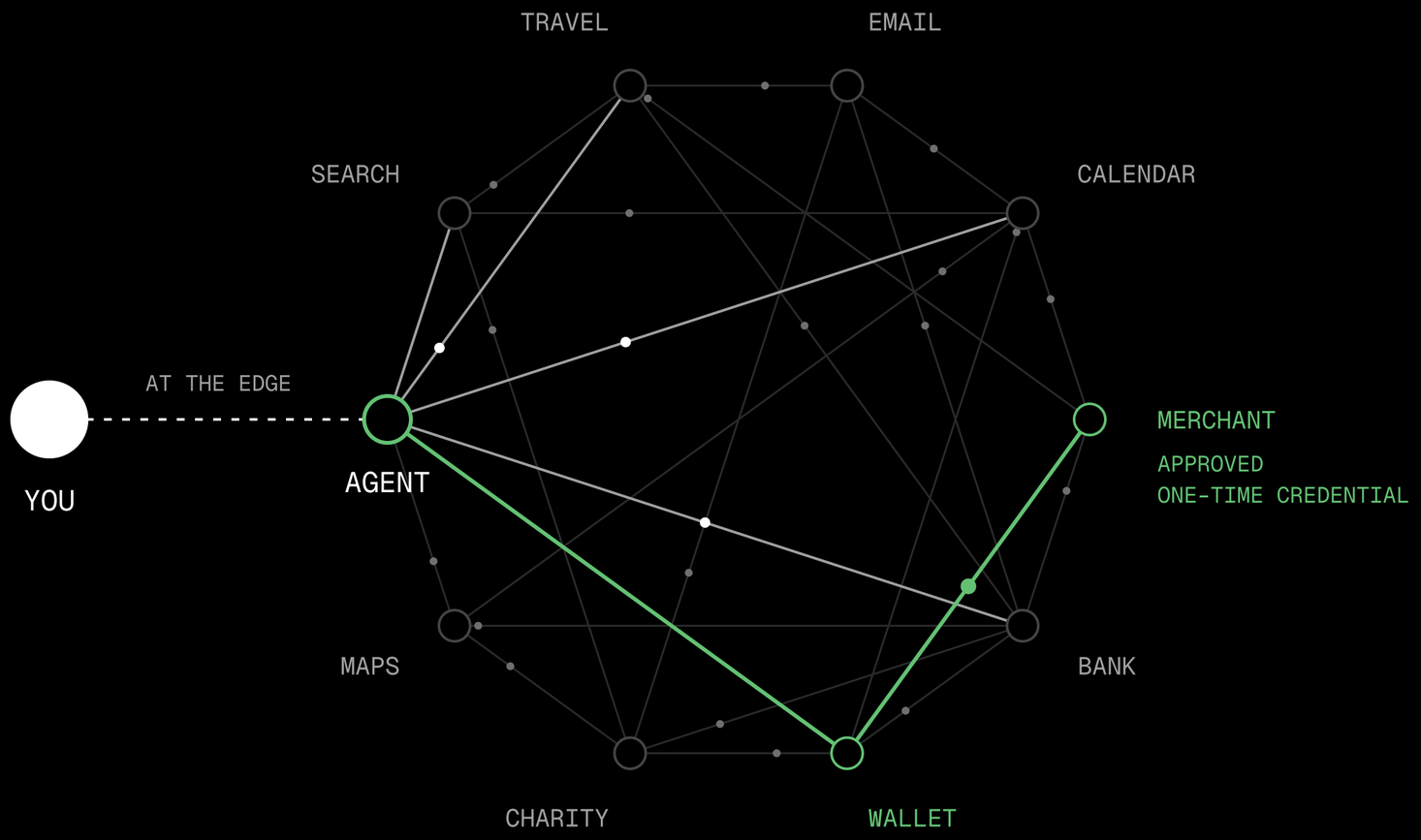
People interact at the edges of the internet, through agents.

AGENTS TRAVERSE

They search, compare, and act across many services for you.

PAYMENTS FOLLOW

Secure payments have to happen on the user's behalf.



Those agents need a harness.
eve is a framework for building them.

Open
Prioritizing open source and open weights

OPEN SOURCE Apache-2.0. Read it, fork it, run it.	OPEN WEIGHTS Open-weight models through AI Gateway, or your own provider.
ANY INFRASTRUCTURE Deploy to Vercel, or self-host it as a Node service.	PUBLIC APIS eve's own features use the same APIs you do.

Production-ready agents out of the box

MULTIPLAYER

Knows who is asking, and acts with that person's own connections.

AGENT TO AGENT

Publish your agent over MCP, or call someone else's.

DYNAMIC CORE

Each caller gets their own tools, skills, and instructions.

DURABLE EXECUTION

Runs survive restarts and use no compute while they wait.

An agent is a folder
Add a file, and it can do something new

my-agent/agent/

agent.ts	the model and its settings	typescript
instructions.md	what it should do	markdown
tools/	what it can do	typescript
skills/	what it knows, loaded when needed	markdown
channels/	where people reach it	typescript
schedules/	when it wakes up on its own	typescript
subagents/	who it hands work to	

Extensions and the registry
Bundle common utilities to share

TERMINAL

```
# Find a capability
eve registry search browser

# Install it into agent/
eve add extension/link
eve add extension/kernel
eve add channel/slack

# Share your own
eve registry add @acme=https://acme.dev/r/{name}.json
eve add @acme/analytics
```

AN EXTENSION BUNDLES

- tools/
- skills/
- subagents/
- connections/
- channels/
- schedules/
- hooks/
- instructions.md

▶ WALKTHROUGH

Let's build a shopping assistant

LINK

Balances, transactions, approved one-time payment credentials

KERNEL

Cloud browsers with a live view you can watch or take over

EVE

Instructions, confirmation, channels, and durable sessions

Add Link and Kernel

Two extensions from the registry

```
$ eve add extension/link
```

agent/extensions/link.ts

```ts
import link from "@stripe/link-integrations-eve";

export default link({
  accessToken: process.env.LINK_ACCESS_TOKEN!,
});
```

ENV `LINK_ACCESS_TOKEN`

ADDS 15 wallet tools · financial-insights and create-payment-credential skills

```
$ eve add extension/kernel
```

agent/extensions/kernel.ts

```ts
export { default } from "@onkernel/eve-extension";
```

ENV `KERNEL_API_KEY`

ADDS Cloud browser connection · kernel__browse skill with live view

The project

agent/		
agent.ts	model and settings	EVE INIT
instructions.md	safe-payment rules	EVE INIT
channels/slack.ts	Slack, allowlisted	EVE ADD
extensions/link.ts	wallet tools and skills	EVE ADD
extensions/kernel.ts	cloud browser	EVE ADD
tools/ask_question.ts	final confirmation	EVE ADD
tools/link__create_spend_request.ts	test mode override	OURS
sandbox.ts	lightweight sandbox	OURS

agent.ts

```ts
import { defineAgent } from "eve";

export default defineAgent({
  // Routed through AI Gateway
  model: "openai/gpt-6-luna",

  // Not used in this demo:
  // reasoning: "high",
  // compaction: { thresholdPercent: 0.75 },
  // limits: { maxTokenCostUsdPerSession: 1.5 },
});
```

instructions.md

```
# Personal Shopper

Help the user browse stores, buy items they approve, and understand their
connected Link wallet and financial activity.

- Load `link__financial-insights` for balances, funding sources, transactions,
  or insights, and `link__create-payment-credential` for spend requests.
- Keep Eve's confirmation and Link's purchase authorization separate. Present
  Link's approval URL, then retrieve the same spend request to confirm status.
- Before submitting checkout, summarize the merchant, items, delivery details,
  and final total, then use `ask_question` to get explicit confirmation.
  If confirmation is unavailable, stop.
- A spend request or issued credential does not mean a purchase completed.
  Report success only when the merchant confirms the order.
```

Custom tools
Define custom logic with TypeScript

agent/tools/link__create_spend_request.ts · trimmed

```ts
import { create_spend_request } from "@stripe/link-integrations-eve/tools";
import { defineTool } from "eve/tools";

export default defineTool({
  ...create_spend_request,
  execute(input, ctx) {
    const mode = process.env.LINK_TEST_MODE ?? "true";
    const test = mode === "true" || input.test;
    return create_spend_request.execute({ ...input, test }, ctx);
  },
});
```

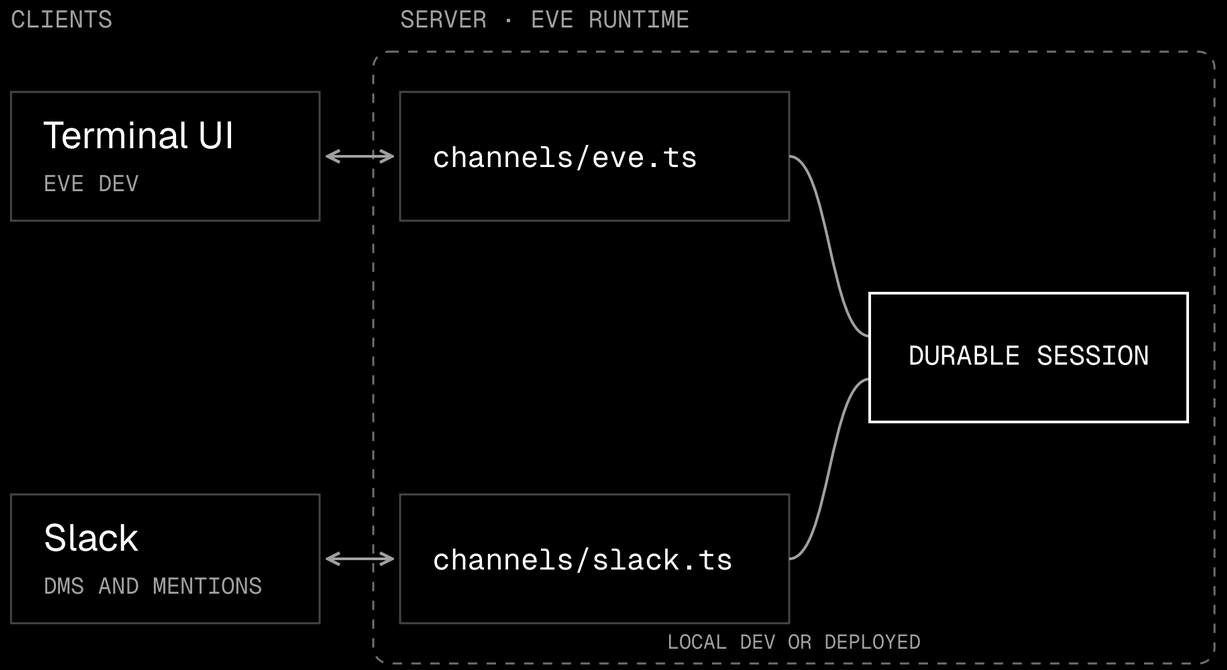
TOOL NAMES ARE DERIVED FROM FILE NAMES

Same name as Link's tool, so ours replaces it, keeping its schema and approval policy.

POLICY IN CODE

Only the operator's environment can allow live payments. The model can't.

Channels
Many frontends to a shared session



agent/channels/slack.ts

```ts
function admit(ctx, message) {
  if (!mayUseSlack(policy, message)) return null;
  return { auth: defaultSlackAuth(message, ctx) };
}

export default slackChannel({
  onDirectMessage: admit,
  onAppMention: admit,
  onInputResponse: admitAnswer,
  approvalChannel: () => "direct-message",
});
```

Multi-tenancy
Who is asking, and whose wallet

TERMINAL	vercel user · local-dev
SLACK	slack:<team>:<user>

Every turn carries a verified principal. Each Slack thread is its own durable session.

THIS DEMO · ONE OWNER

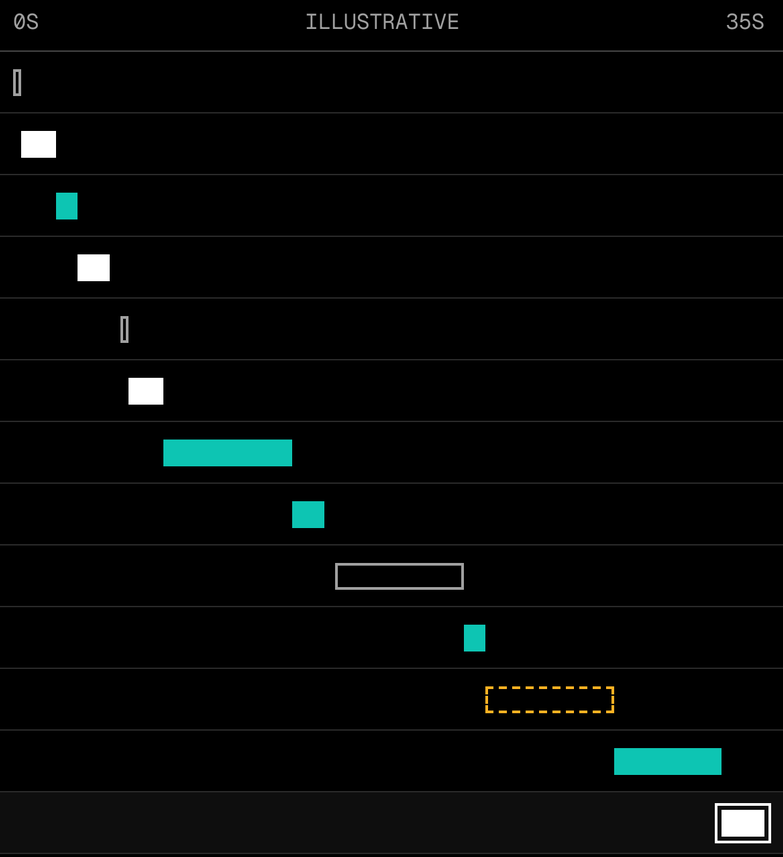
One Link token, mine. Every session spends from it, so Slack is allowlisted and fails closed.

MULTIPLAYER · PER-PERSON CREDENTIALS

User-scoped connections key tokens to the principal. Each person authorizes their own account; the model never sees a token.

Session walkthrough
One run, step by step

KIND	STEP
USER	“What's my balance?”
MODEL	Load financial-insights
TOOL	`link__list_balances`
MODEL	Reply with balances
USER	“Donate $1 to Stripe Climate”
MODEL	Load browse + payment skills
TOOL	`Kernel: open contribute page`
TOOL	`link__create_spend_request`
USER	Approve in the Link app
TOOL	`link__retrieve_spend_request`
PARKED	`ask_question: confirm $1`
TOOL	`Kernel: submit checkout`
MODEL	Verify and reply



MODEL · STEP 13 OF 13

Verify and reply

Success only from the merchant's confirmation. No blind retries on an uncertain payment.

CONTEXT · 15 ITEMS

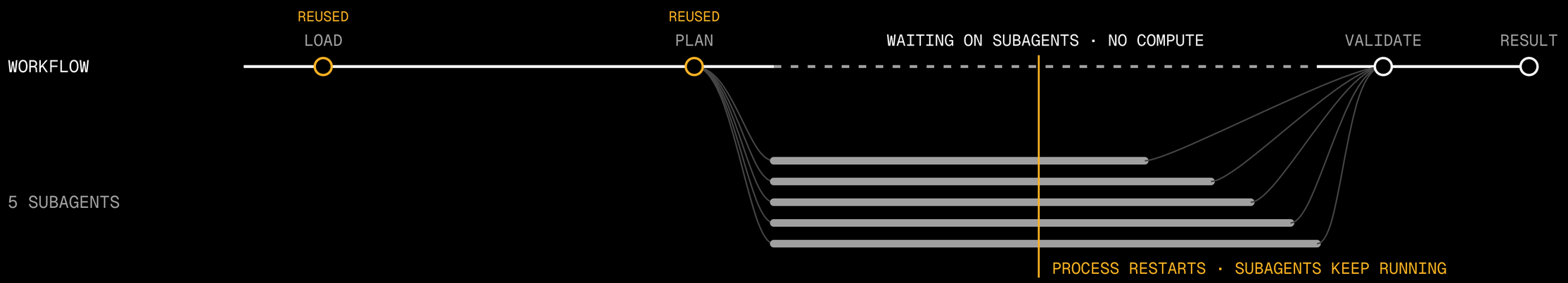
+ 9 earlier

`Page snapshot (untrusted)`
`Spend request + approval link`
`User: approved in Link`
`Spend request: approved`
`Answer: Confirm`
`Merchant result page`

AVAILABLE CAPABILITIES

`link__*` 15 wallet tools	`kernel__*` cloud browser
`ask_question` durable question	`skills` loaded on demand

Every step is a checkpoint
How eve runs durable work



CHECKPOINTED STEPS

Each step saves its result as soon as it finishes.

WAITS RELEASE COMPUTE

While subagents work, or a person approves, it holds no compute.

RESTARTS REPLAY

If the server dies, a new one reuses the finished steps.



Build your own agent

```
npx eve@latest init personal-shopping-assistant
cd personal-shopping-assistant
npx eve add extension/link
```

