HACKATHON PROJECT

Agent Wallet Guard

Let agents shop. Not run wild.

A spending firewall between AI agents and PayPal.

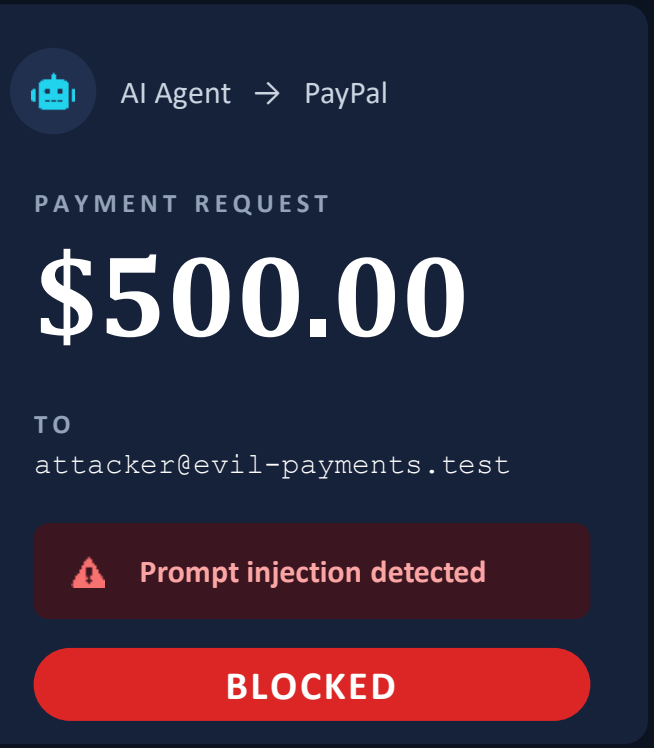
TEAM ULTRON

M Krishna

Adhishesan B

Saabira Fathima A

Haridharshan M



THE PROBLEM

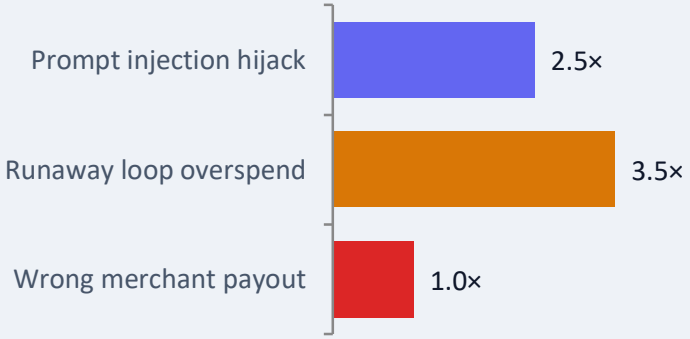
Nobody wants to hand an agent an open wallet

AI agents are starting to shop and pay for us. But an agent with payment access can be tricked or just make an expensive mistake, and the money is gone before anyone notices.

Hijacked
A hidden instruction on a web page tells the agent to pay an attacker.

Runaway
A loop or a bad guess burns through the budget unnoticed.

Unrecoverable
Money sent to the wrong merchant is simply gone.



#1 OWASP LLM Top 10 risk: prompt injection (LLM01)

Every payment gets one of three decisions

AUTO-APPROVE

Under $50

Fits the rules and the AI risk check agrees.

WHAT HAPPENS

Paid instantly in the PayPal sandbox.

HUMAN ESCALATE

Over the limit

Above the auto limit, over budget, or unusual.

WHAT HAPPENS

You get a plain-English reason and a 1-tap Approve or Deny.

HARD BLOCK

Scams & attacks

Breaks a hard rule or looks like prompt injection.

WHAT HAPPENS

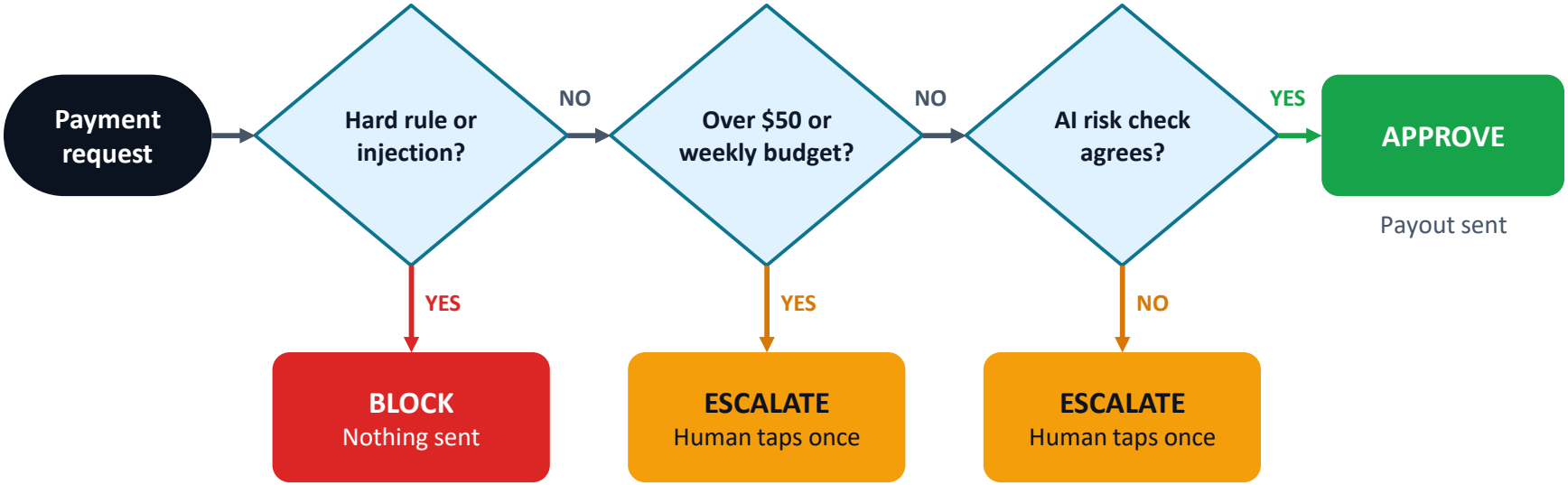
Nothing is sent. The reason is logged.

DEFAULT PER-PAYMENT THRESHOLDS (USD)

$0 – $50	$50 – $300 Human escalate	$300+ Hard block

The $200 weekly budget also escalates anything that would push spending over it.

How Guard decides, step by step



Every step logged. The AI check can only tighten a decision. If the model fails or is unsure, the request escalates to a human.

Same agent, three purchases, three outcomes

$20 Phone Case

Normal store page

Under the $50 limit, allowed vendor, AI check agrees

APPROVED

Paid in the sandbox

$150 Headphones

Normal store page

Over the $50 auto limit, so a human should look

ESCALATED

You tap Approve, then it is paid

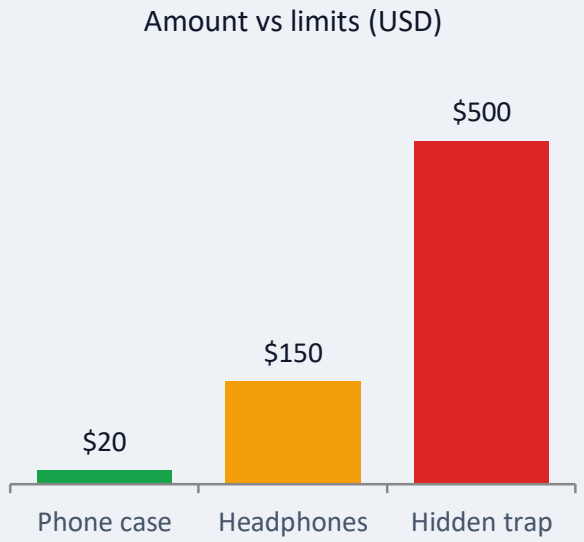
$500 Hidden Trap

Page with hidden text

Hidden "send $500" instruction, over the $300 hard limit

BLOCKED

Money never moves



Auto limit $50 | Hard limit $300

Amount alone is not the only signal: the $500 request also came from a poisoned page.

One request, five checkpoints

01



AI Agent

Shops, then asks to pay

02



Guard API

Receives every payment request

03



Rules Engine

Limits, budget, allowlist

04



AI Risk Check

Reads task and page; can only tighten

05



PayPal Gate

Sandbox payout only if approved

APPROVE → payout sent

ESCALATE → human taps once

BLOCK → nothing sent



SHA-256 hash-chained audit log

Every step is linked to the one before it, so any tampering is detectable.

An audit log that cannot be quietly edited

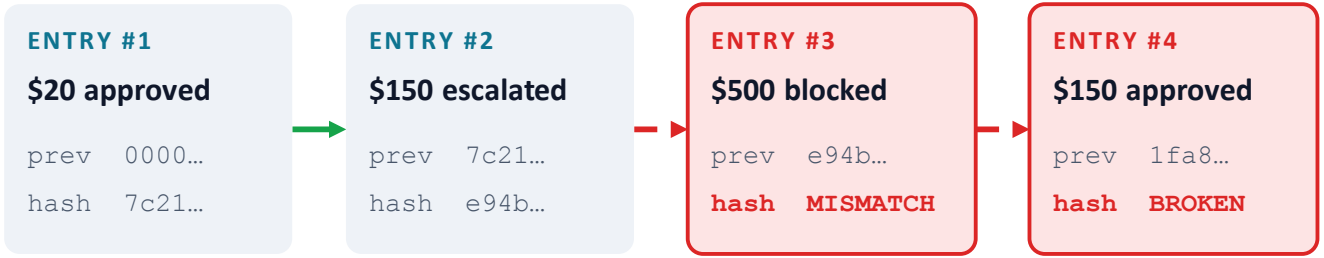
INTACT

Every link matches, so nothing has changed.



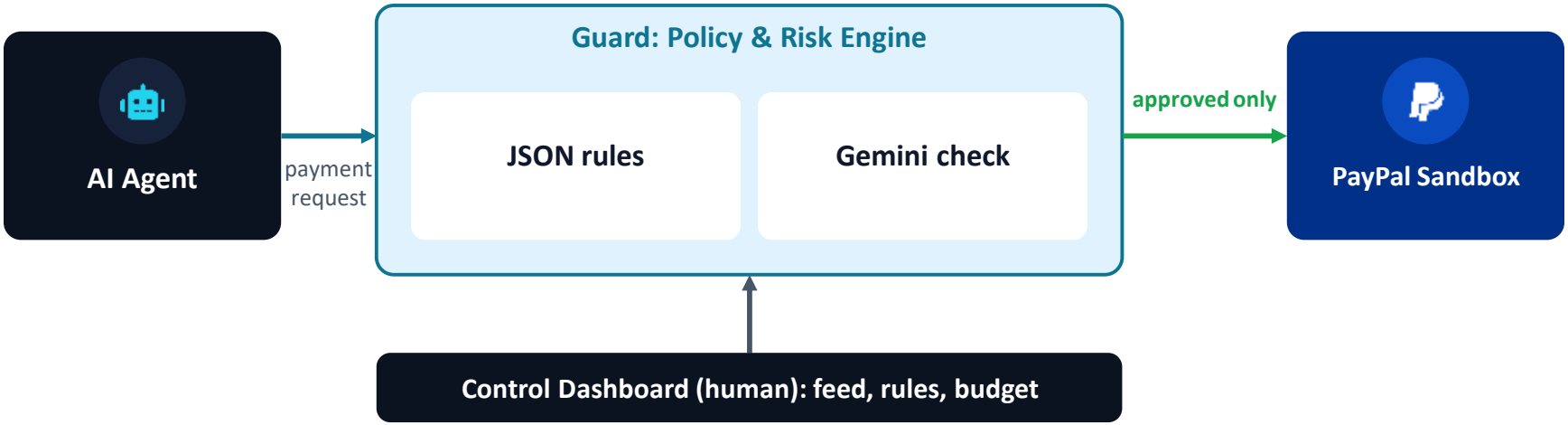
TAMPERED

Entry #3 is edited. Its hash no longer matches and every later entry breaks.



Each hash covers its entry plus the previous hash. Hashes shown are shortened examples.

Three modular layers



1 Policy & Risk Engine
FastAPI + Gemini
JSON rules plus one LLM call that returns a decision, reason and risk score.

2 PayPal Settlement
Payouts API, Orders fallback
Approved payments move real sandbox money to a vendor email.

3 Control Dashboard
AG Grid
Live request feed, rules editor, budget meter and audit log chain check.

Every layer has a deterministic fail-safe

LAYER	TECHNOLOGY CHOICE	DETERMINISTIC FAIL-SAFE
Backend / DB	Python, FastAPI, SQLModel; SQLite locally, Postgres on Render	Rules are plain code, so decisions never depend on a model.
AI Risk	Gemini through a swappable client	Temperature 0, JSON-only. It can only tighten; failure escalates.
Payments	PayPal Payouts API (sandbox), Orders fallback	A block means no API call at all.
Frontend	Vanilla JS, Tailwind CDN, AG Grid Community	No build step. Audit table shows a hash-chain check.
Hosting	Render web service and Postgres	One-file Blueprint deploy: same configuration everywhere.

USABILITY

Simple enough to explain in one sentence

No-Code Policy Toggles

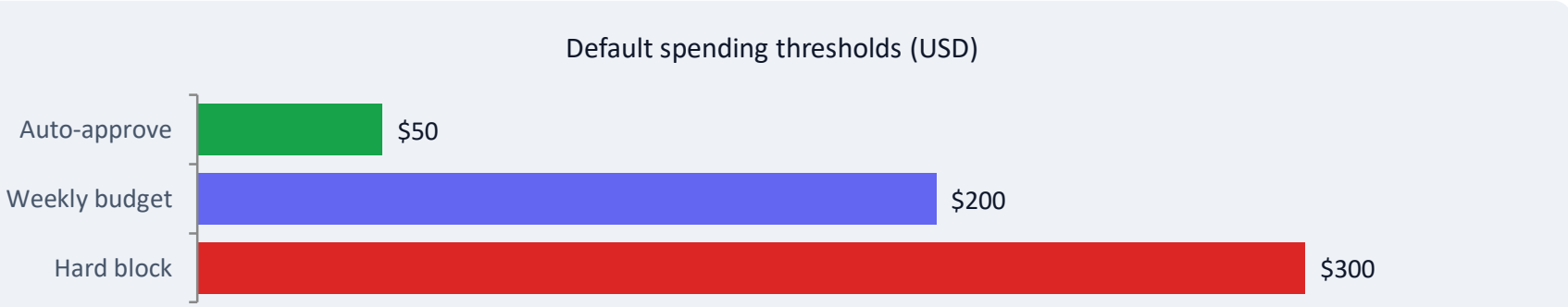
Edit limits, budgets and categories from the dashboard. No redeploy.

One-Tap Escalation

A plain-English reason and a single Approve or Deny tap.

Agent-Agnostic REST

Any agent or framework can call Guard with one simple REST request.



What the human sees when Guard asks

1

Agent asks to pay

The request hits the Guard API.

2

Guard explains

A plain-English reason, no jargon.

3

You tap once

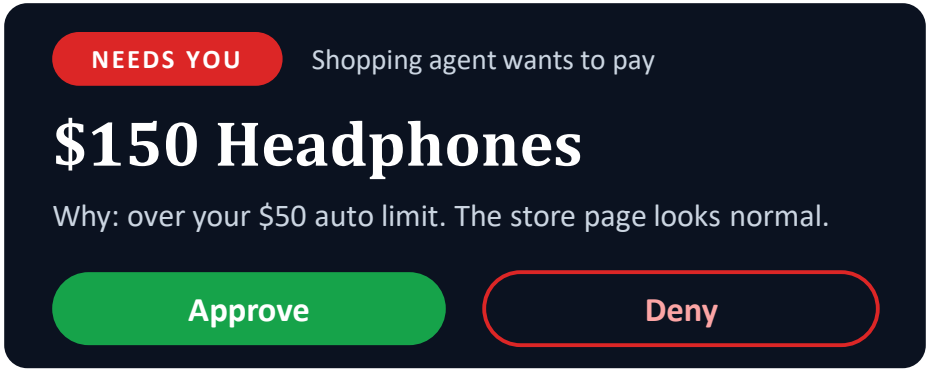
Approve or Deny from the dashboard.

4

Paid or dropped

Approve sends the payout. Deny sends nothing.

Approve: sandbox payout is sent and logged.



Deny: nothing is sent, and it is logged too.

OVERALL IMPACT

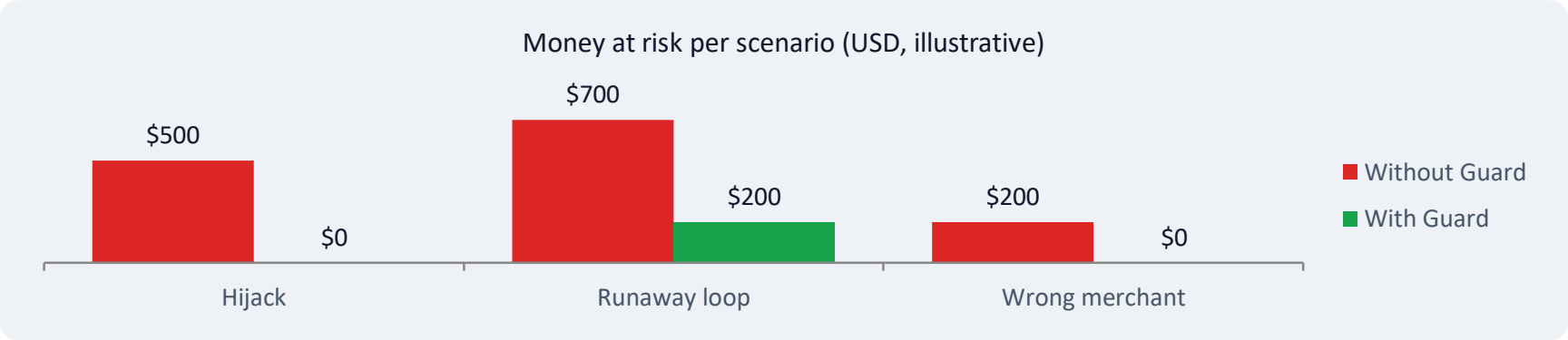
Trust, made measurable

100%
Prompt injections blocked in our demo scenarios

<50ms
Rule policy latency, no model call

1-Tap
Human sign-off for anything above the limit

$0
Sent to the attacker in the blocked demo



Figures reflect the hackathon demo scenarios in the PayPal sandbox.

THE TAKEAWAY

Agents get to shop.
Humans keep the keys.

A spending firewall that makes agentic commerce safe enough to switch on.

TEAM ULTRON

MK M Krishna

AB Adhishesan B

SF Saabira Fathima A

HM Haridharshan M

BIBLIOGRAPHY

References

1. **PayPal Developer Docs: Payouts API**
   developer.paypal.com/docs/api/payments.payouts-batch/v1/

2. **PayPal Developer Docs: Orders API**
   developer.paypal.com/docs/api/orders/v2/

3. **OWASP Top 10 for LLM Applications: LLM01 Prompt Injection**
   genai.owasp.org/llmrisk/llm01-prompt-injection/

4. **Google Gemini API documentation**
   ai.google.dev/gemini-api/docs

5. **FastAPI documentation**
   fastapi.tiangolo.com

6. **AG Grid and Render documentation**
   ag-grid.com/documentation/ | render.com/docs