APS 🤖💳

AgentPayment Sandbox

Sivasubramanian Ramanathan
Product Owner | Fintech & Innovation
Ex-BIS Innovation Hub Singapore

🌏 **Seeking Opportunities in Singapore**
I am looking for roles in **Product Management, Fintech, Payments, RegTech**, and **Digital Assets**.

"I am a Product person. **I build.**"

I have worked across product delivery, user research, and cross-agency collaboration. I enjoy solving complex problems and bringing structure to early ideas.

I care deeply about building products that create real impact.

The Paradigm Shift 💡

We are witnessing a fundamental change in how commerce happens on the internet.

1. The Old World (Human Commerce) 🖱️

- Human → Browser → Click "Buy" → Enter Card → **CAPTCHA** → Confirm.
- **Designed for**: Eyeballs and fingers.
- **Problem**: No standard API for "agent wants to buy this".

2. The New World (Agentic Commerce) 🤖

- Agent → API Discovery → Structured Checkout → **Cryptographic Auth** → Done.
- **Designed for**: Autonomous software agents.
- **Solution**: Mandates and limits replace CAPTCHAs.

> ***Impact***: *AI agents can now shop, pay, and transact autonomously.*

The Problem: Agents Can't Shop 🛒❌

AI agents are everywhere, but commerce infrastructure hasn't caught up.

Current Reality

- 🔴 CAPTCHAs actively **block** automation.
- 🔴 Payment forms require **human interaction**.
- 🔴 No standard **protocol** for agent transactions.
- 🔴 Testing requires **real money** or complex setups.

The Gap

- 🟡 Developers building agent commerce have **no sandbox**.
- 🟡 4 competing protocols, each **partially documented**.
- 🟡 Security testing means **manual code review**.
- 🟡 Edge cases are **impossible to reproduce**.

> *"**Question**: How do you test an agent shopping flow without spending real money?"*

Four Protocols, One Sandbox 🧪

The agentic commerce landscape is fragmenting. I built a unified testing tool.

Protocol	Maintainer	Use Case
AP2	Google	Agent-to-Agent with cryptographic mandates
x402	Coinbase	HTTP 402 micropayments for premium APIs
ACP	Shopify	OpenAPI checkout for complex purchases
UCP	Stripe	Universal commerce for seamless checkout

> ***APS provides mock servers, validators, and an interactive playground for ALL of them.***

Introducing APS 🚀

Postman + Chaos Monkey + Case Manager for Agent Payments

What It Does

- ⚡ **Mock Servers**: Simulate merchants without real infrastructure.
- 🔍 **Inspector**: Validate your implementation against protocol specs.
- 🛡️ **Security Analyzer**: Verify signatures and check for vulnerabilities.
- 🎮 **Playground UI**: Explore protocols interactively.

Who It's For

- Developers building **AI shopping agents**.
- Teams implementing **payment integrations**.
- Engineers learning **agentic commerce protocols**.
- Anyone testing **agent-to-merchant flows**.

The Technical Build 🔧

I architected this to bridge the gap between **Protocol Specs** and **Working Code**.

Architecture

- ⚡ **Frontend**: React + TypeScript + Vite (Interactive Playground UI)
- 🐍 **Backend**: FastAPI + Pydantic (Mock Servers + Validators)
- 📋 **Docs**: 8 documentation files, 3 ADRs

Features Built

- 4 Mock Servers (UCP, ACP, x402, AP2)
- Protocol Inspector with test suites
- Security Analyzer (signature verification)
- State machine for session management
- GitHub Pages deployment

Key Scenarios: x402 Micropayments ⚡

HTTP 402 "Payment Required" enables pay-per-request APIs.

```
Client → GET /premium-api → 402 Payment Required
                          ← X-Payment-Required header

Client builds EIP-712 signed payment payload

Client → GET /premium-api + X-PAYMENT header
                          ← ✅ API Response
```

Use Cases: AI model access, premium data feeds, content paywalls.

> “***APS simulates the entire flow*** *with mock USDC on Base Sepolia.*”

Key Scenarios: AP2 Multi-Agent 🤖🤖🤖

Google's AP2 protocol enables agent-to-agent commerce with user protection.

```
Shopping Agent → Merchant Agent: "Browse laptops"
Merchant Agent → Shopping Agent: CartMandate (requires signature)
Shopping Agent → User: "Approve $2,748.90?"
User → Shopping Agent: OTP via Credentials Provider
Shopping Agent → Payment Processor: Authorize
Payment Processor → Merchant Agent: ✅ Receipt
```

Key Innovation: Mandates ensure agents can't spend beyond limits.

> *"**APS includes OTP challenge simulation** for full flow testing."*

Demo Mode for GitHub Pages 🌐

The live demo works without a backend using intelligent mock data.

The Challenge

GitHub Pages is static hosting. There's **no server** to process API calls.

The Solution

- Detect `github.io` hostname.
- Return **realistic mock data**.
- Show clear **Demo Mode** banner.
- Provide instructions for local backend.

What Users See

```
⚠️ Demo Mode

Running with mock data.
For live API calls, run:

uvicorn app.main:app --port 8080
```

Why This Matters 🌟

This project reflects my ability to bridge **Emerging Tech** and **Real Products**:

1. **Protocol Expertise**:
   Deep understanding of 4 competing agentic commerce standards.
2. **Full-Stack Execution**:
   Built React frontend, Python backend, documentation, and CI/CD.
3. **Developer Experience**:
   Focused on making complex protocols **accessible and testable**.
4. **Future-Oriented**:
   Positioned at the intersection of AI agents, payments, and standards.

About Me 👨‍💼

Sivasubramanian Ramanathan

Product Owner | Fintech, Payments & Digital Innovation

PMP | PSM II | PSPO II

I specialize in taking messy, real-world complexity and structuring it into reliable products.

Open for roles that sit between policy, technology, and stakeholder engagement.

Previous: **BIS Innovation Hub Singapore** - Cross-border payments, CBDCs, regulatory innovation.

Let's Connect 🤝

I am ready to bring this level of product thinking and technical depth to your team.

- 🌐 **Portfolio**: sivasub.com
- 💼 **LinkedIn**: linkedin.com/in/sivasub987
- 💻 **Code**: github.com/siva-sub/AgentPayment-Sandbox
- 📚 **Docs**: Full Documentation

Live Demo:
siva-sub.github.io/AgentPayment-Sandbox