



## 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 Agentic Commerce Vision



*Your AI assistant wants to book you a flight. What happens next?*

**Today, the agent hits a wall.**

**The Current Reality** 

- Agent navigates to airline website.
- Encounters **CAPTCHA**: "Select all traffic lights."
- **Game Over**. Agent cannot proceed.

**The Emerging Future** 

- Agent discovers airline's **Agent API** (like `/well-known/a2a`).
- Sends structured **payment mandate** with your pre-approved limits.
- Receives confirmation. **You just flew.**

**“This is not science fiction. Four protocols are making this real today.”**

”

# The Paradigm Shift



Just like SWIFT transformed paper-based banking, **agentic protocols** are transforming online commerce.

**It is not about the AI itself. It is about designing infrastructure FOR agents.**

# Error 400: Una

- **From:** "Prove you're human" (CAPTCHAs, forms, clicks).
- **To:** "Prove you're authorized" (Cryptographic signatures, mandates).

# The Problem: No Testing Infrastructure



*I am building an AI shopping agent. How do I test it?*

## Option 1: Real Services



- Sign up for Stripe Test Mode.
- Configure Shopify sandbox.
- Apply for API keys from each provider.
- **Spend days** on setup.

## Option 2: Build Mock Servers



- Read 4 different protocol specs.
- Implement each from scratch.
- Maintain parity with spec changes.
- **Spend weeks** on infrastructure.

## The Gap

There is no "**Postman for agentic commerce**".

Developers cannot:

- ✗ Test flows without real money
- ✗ Validate compliance easily
- ✗ Reproduce edge cases
- ✗ Learn protocols interactively

**“APS fills this gap.”**

## Four Protocols, One Problem



*The agentic commerce landscape is fragmenting. Each solves part of the puzzle.*

# Error 400: u

“**Question:** How do developers test against all 4 without building 4 mock servers?

”

# Introducing APS



*Postman + Chaos Monkey + Case Manager for Agent Payments*

# Error 400: Un

APS provides **mock servers** for all 4 protocols, plus:

- **Inspector**: Validate your implementation against specs.
- **Playground**: Explore protocols interactively.
- **Security Analyzer**: Check signatures and mandates.

# The x402 Flow: Micropayments ⚡

*HTTP 402 "Payment Required" for pay-per-request APIs.*

# Error 400:

**Use Case:** AI model access, premium data APIs, content paywalls.

“APS simulates the **402 response** and **signature verification**.

”

## The AP2 Flow: Multi-Agent

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

# Error 400:

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

“APS includes **OTP challenge** and **credentials provider** simulation.”

## The Technical Architecture

I built a full-stack testing platform to demonstrate end-to-end agent payment flows.

# Error 400: Una

- **Frontend:** React + TypeScript + Vite
- **Backend:** FastAPI + Pydantic (Mock Servers)
- **Docs:** 8 documentation files, 3 Architecture Decision Records

# Demo Mode: Works Everywhere



*The live demo runs on GitHub Pages (static hosting). How?*

## The Challenge

GitHub Pages has **no server**.  
API calls would fail.

## The Solution

- Detect `github.io` hostname.
- Return **realistic mock data**.
- Show clear **Demo Mode** banner.

## Developer Experience

### ⚠ Demo Mode

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

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

“Users understand the limitation and can run locally for full testing.”

# Why This Matters



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

## 1. Problem-First Thinking:

Started with "developers can't test agent payments" - not "let me build cool tech".

## 2. Protocol Expertise:

Deep understanding of 4 competing agentic commerce standards.

## 3. Full-Stack Execution:

Built React frontend, Python backend, documentation, and CI/CD.

## 4. Developer Experience:

Focused on making complex protocols **accessible and testable**.

# 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](http://sivasub.com)
-  **LinkedIn:** [linkedin.com/in/sivasub987](https://linkedin.com/in/sivasub987)
-  **Code:** [github.com/siva-sub/AgentPayment-Sandbox](https://github.com/siva-sub/AgentPayment-Sandbox)
-  **Docs:** [Full Documentation](#)

**Live Demo:**  
[siva-sub.github.io/AgentPayment-Sandbox](https://siva-sub.github.io/AgentPayment-Sandbox)