



## 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.**"

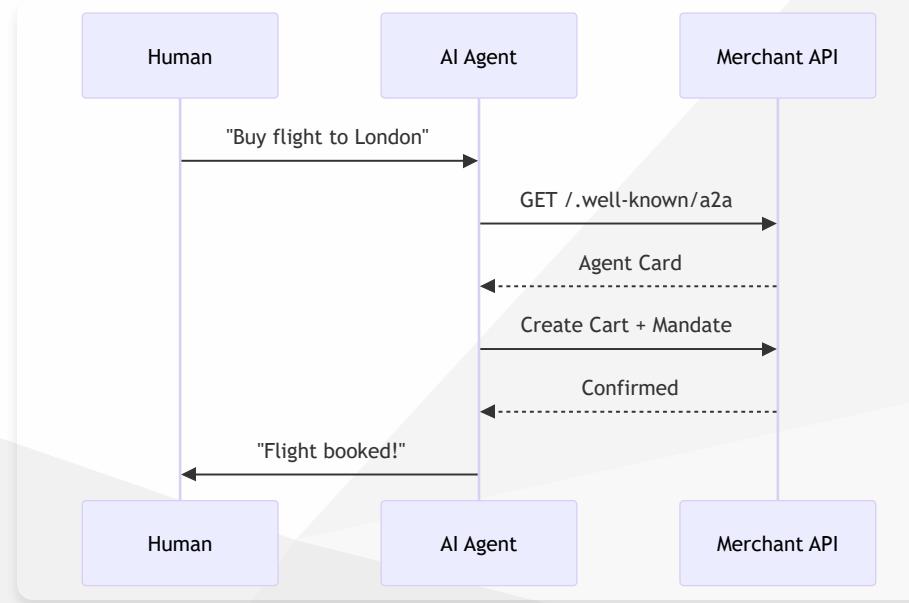
I have worked across product delivery, user research, and cross-agency collaboration. I enjoy solving complex problems and bringing structure to early ideas.

**I care deeply about building products that create real impact.**

# The Vision: Agents Will Shop For You



We are moving from a world where **humans click buttons** to a world where **agents negotiate protocols**.



## The Shift

- **Today:** Humans navigate UI, solve CAPTCHAs, enter details.
- **Tomorrow:** Agents use APIs JSON-RPC, and Crypto Signatures.

# The Reality: A Testing Nightmare

*The infrastructure is here, but it is fragmented.*

Protocol	Company	Approach
AP2	Google	"Agent-to-Agent" mandates
x402	Coinbase	HTTP 402 Micropayments
ACP	Shopify + OpenAI	E-commerce checkout
UCP	Stripe	Universal payment API

“**The Problem:** Developers need to validate signatures, parse headers, and handle mandates across 4 disparate specs.”

# The Developer Pain Point



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

## Option A: Real Test Accounts



- Sign up for Stripe/Shopify.
- Get API keys.
- Configure webhooks.
- **Days of setup.**

## Option B: Build Mock Servers



- Read 4 different PDF specs.
- Code endpoints from scratch.
- Handle encryption manually.
- **Weeks of work.**

**“There was no “Postman for Agent Payments”. So I built one.”**

# Introducing APS



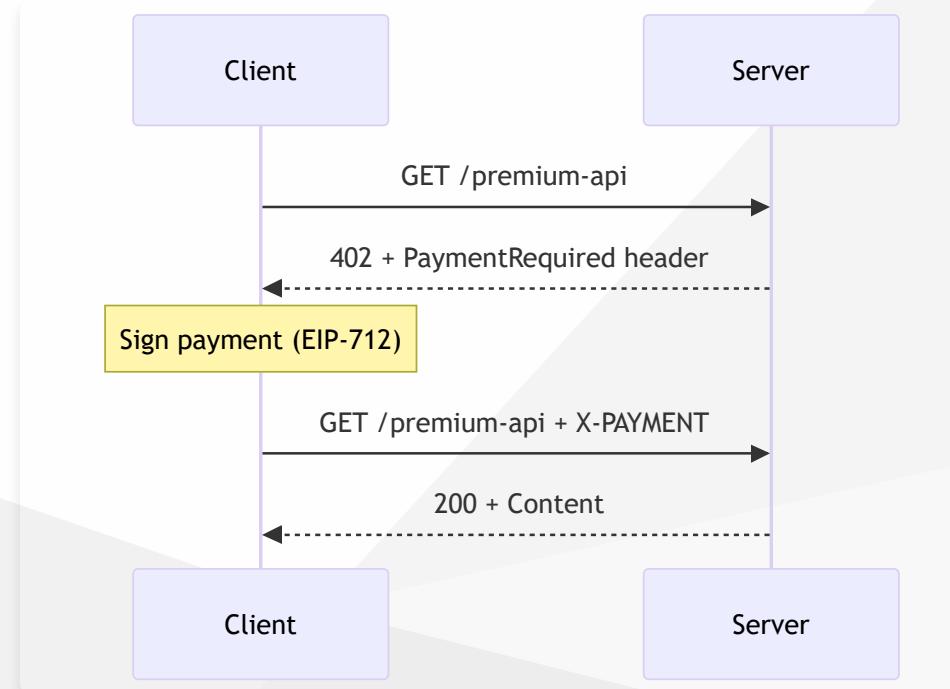
*A Unified Sandbox for Agentic Commerce Protocols*

# Error 400: Un

- **Mock Servers:** Instant endpoints for UCP, ACP, x402, AP2.
- **Inspector:** Validates your requests against official schemas.
- **Security Overview:** Verifies cryptographic signatures (EIP-712).
- **Playground:** Learn by doing in the UI.

# Testing Flow 1: x402 (Coinbase) ⚡

*The Standard for Micropayments & Metered Access.*

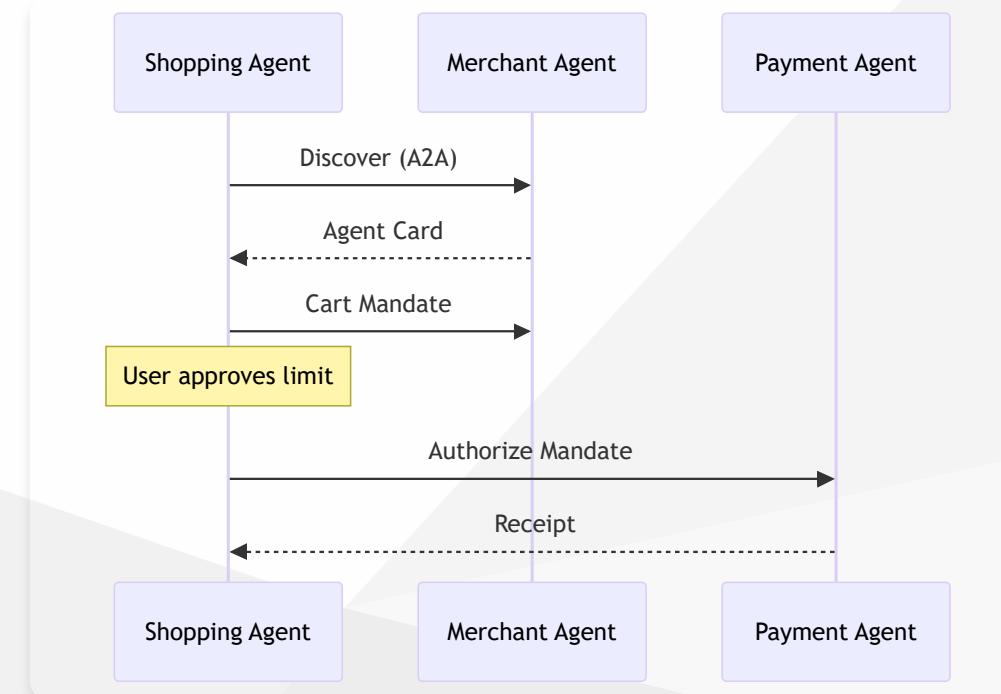


## What APS Tests:

- Header Parsing:** Does the agent correctly parse **402 Payment Required** headers?
- Signature Validation:** Is the **X-Payment** header signature valid?
- Payload Structure:** Does the payment payload match the resource price?

# Testing Flow 2: AP2 (Google)

*High-Trust Commerce with User Mandates.*



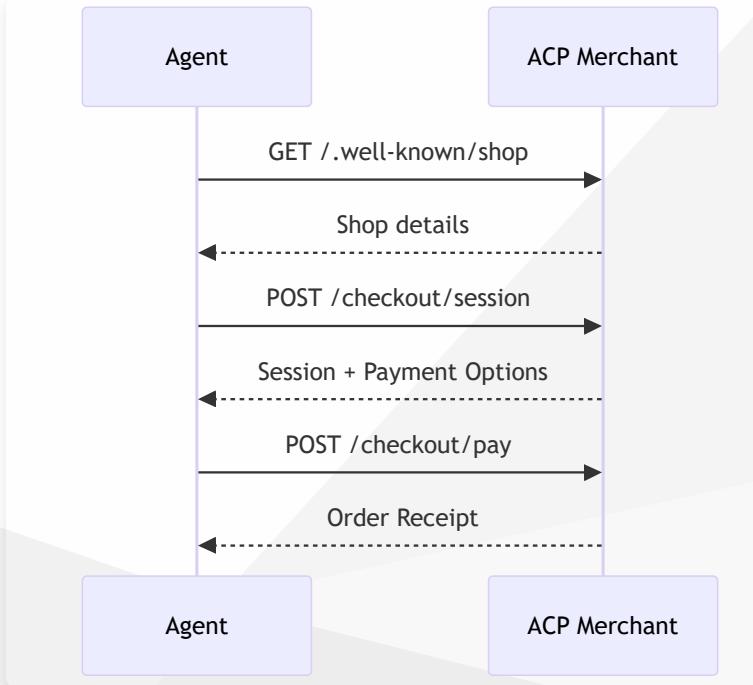
## What APS Tests:

1. **Discovery:** Can the agent find and parse the `/.well-known/a2a` card?
2. **Mandate Limits:** Does the agent respect the spending limit in the Mandate?
3. **OTR Challenge:** Can the agent handle user intervention flows?

# Testing Flow 3: ACP (Shopify + OpenAI)



*Agentic Commerce Protocol: Specialized for E-commerce.*



## What APS Tests:

1. **Session Management:** Can the agent maintain session state across requests?
2. **Product Selection:** Does the agent correctly select variants (size/color)?
3. **Checkout Validity:** Is the shipping address and payment method valid?

# Testing Flow 4: UCP (Stripe)

*Universal Commerce Protocol: The "Stripe for Agents".*

**Error 400: SyntaxError: No diagram type detected matching given configuration for text: sequence**

```
participant Agent
participant UCP Merchant
Agent->>UCP Merchant: GET /.well-known/ucp
UCP Merchant-->>Agent: Capabilities & Keys
Agent->>UCP Merchant: POST /checkout-sessions
UCP Merchant-->>Agent: Session Created
Agent->>UCP Merchant: POST /sessions/{id}/complete
UCP Merchant-->>Agent: Order Confirmed
Error: Syntax error in graph
at Worker.convert (file:///usr/local/kroki/src/worker.js:44:15)
at async file:///usr/local/kroki/src/index.js:31:28
```

## What APS Tests:

1. **Capability Discovery:** Can the agent identify supported payment methods?
2. **Idempotency:** Does the agent safely retry requests without double-charging?
3. **Card Validation:** Are the card details (simulated) correctly formatted?

# Demo Mode: GitHub Pages

*How I deployed a backend-heavy app to static hosting.*

## The Challenge

GitHub Pages = No Server.  
API calls usually fail.

## The Solution

- **Context-Aware API:** Detects `github.io`.
- **Service Worker Logic:** Intercepts requests.
- **Mock Data Layer:** Returns valid JSON responses.

```
// frontend/src/services/api.ts
const IS_DEMO = window.location.hostname
  .includes('github.io');

if (IS_DEMO) {
  return MOCK_DATA.ap2_response;
}
```

“**Result:** A frictionless live demo for recruiters and developers.”

# About Me



**Sivasubramanian Ramanathan**

*Product Owner | Fintech, Payments & Digital Innovation*

I specialize in **simplifying complexity**.

- **Experience:** BIS Innovation Hub Singapore (CBDCs, Cross-Border Payments).
- **Skills:** Product Management, Rapid Prototyping, Technical Strategy.
- **Focus:** Bridging the gap between Policy/Regulation and Engineering.

## Let's Connect

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

-  **Portfolio:** [sivasub.com](http://sivasub.com)
-  **LinkedIn:** [linkedin.com/in/sivasub987](https://linkedin.com/in/sivasub987)
-  **Code:** [GitHub/APS](#)
-  **Docs:** [Documentation](#)

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