Lagoon.Markets

The frontend for x402 micro-transactions

x402 resource fragmentation

- No real world resource discovery

- Each x402 must implement it’s own x402 discovery frontend

- Developers need to support more complex frontends like smartphones

- Payment options are limited to US dollar stablecoins

- Web platforms offer poor user retention for subscriptions

- Poor transaction routing choices

Example real world problem

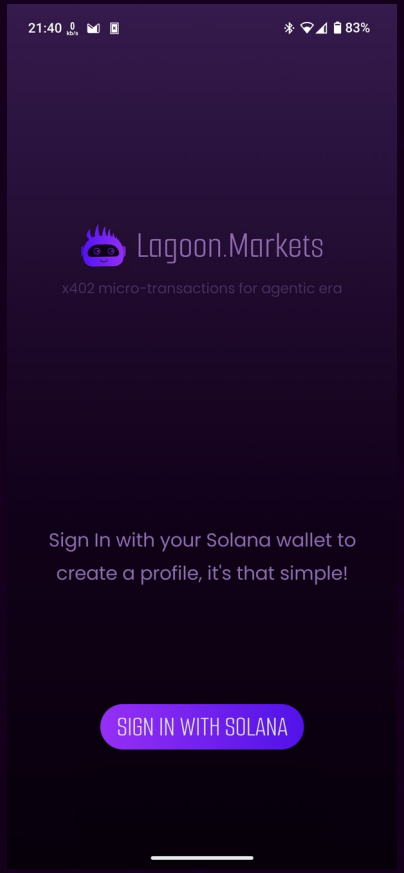
Alice walks past a billboard with her favorite book, **“Conqueror of Blockhains; Taker Of Markets”**, by her favorite author **“Toly the Great”**.

The website publisher who placed the ad on a billboard supports x402 payments but what is Alice to do? She doesn’t know that by simply looking at the billboard.

Alice can perform the tedious task of taking a photo of the billboard then going online to search for a publisher that sells the eBook version, see if she has the currency needed to pay etc. Alice also wants to subscribe to an AI agent to receive updates of new books by the same author but across publishing platforms.

Is there a better way?

x402-URI saves Alice a headache



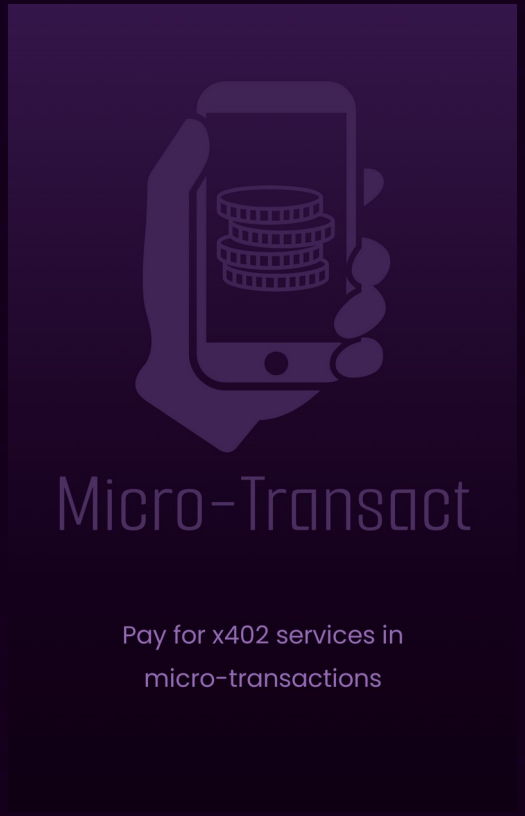
The x402-URI extends the x402 protocol allowing easy discovery of resources.
Resources can be discovered on:
- websites
- mobile apps
- QR-Codes in the real world.

The x402-URI is inspired by **Solana Pay QR Codes** and the HTML **`tel:+`** URI.

A x402-URI is created using the `x402://` scheme where any mobile app implementing the protocol can open the URI

x402-URI Specification ↗️ (Open Link)

Target Market



Millions of users will be able to services that implement the x402 protocol as developers build agentic services. The x402-URI solves this part of the problem.

Developers also need to expose x402 resources using HTTPS endpoints and server-sent events (agenti-to-agent services).

The Lagoon.Markets dapp solves this problem by offering a consistent interface that developers can rely on to show end users an x402 resource.

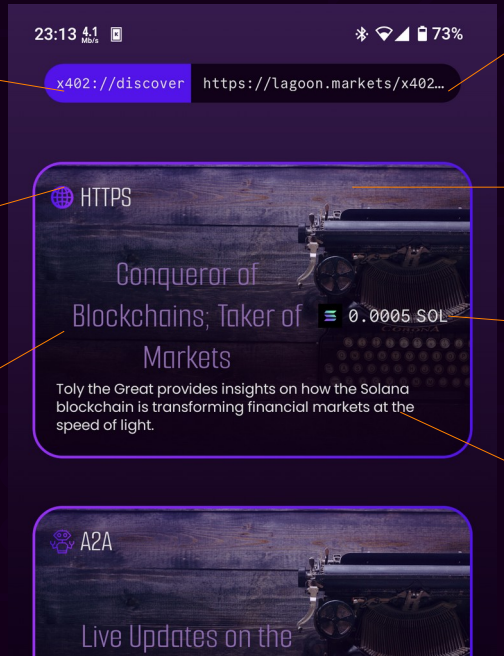
UX: x402 Discovery

The x402 specification defines the **`/discover`** route as a way to get all the resources a server provides. The **Lagoon.Markets** app understands the response and displays the resources in a beautiful list that an end user can just tap to view a discovered resource. The developer doesn't need to build a frontend for this.

x402-URI scheme

The scheme the x402 resource will use. (https, a2a or mcp)

The title of the resource



The URL of the resource provider

Optional banner for the resource

The micro-transaction amount the user will pay to access the resource

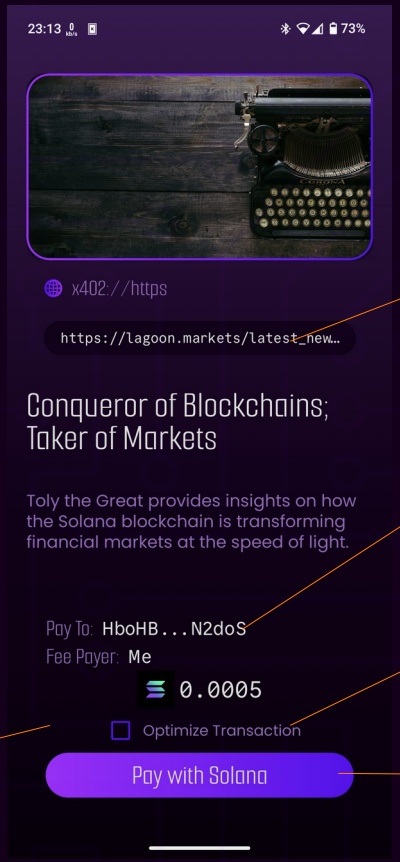
The description of the resource

UX: x402 Discovered Item

A detailed view of the first resource discovered.

For example optimizing transactions with Sanctum gateway or submillisecond settlement using magic block rollups.

These options are provided by the Lagoon.Markets dapp, not the resource server ensuring users don't need to wait for resource providers to provide extra features for transaction signing and routing.



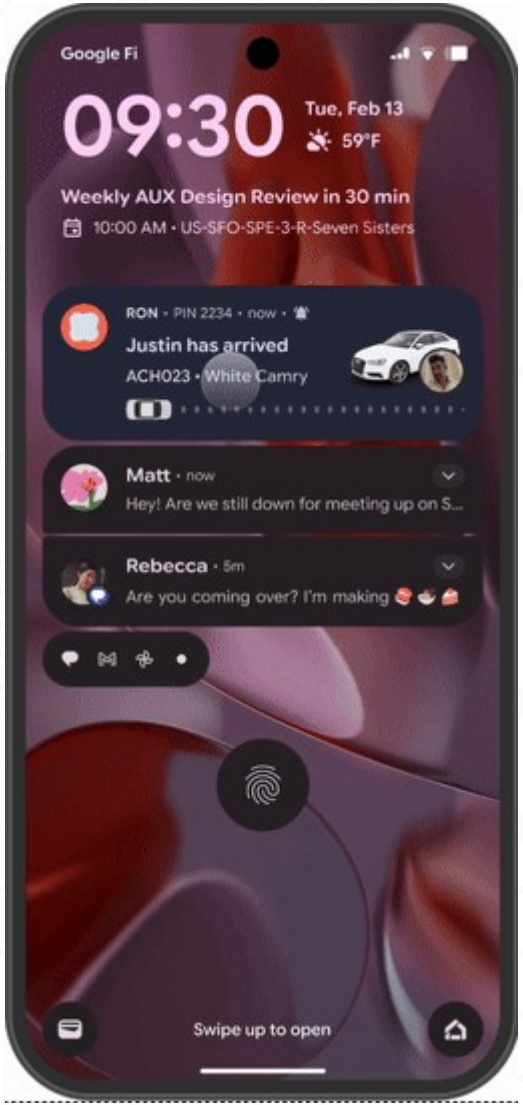
The URL of the resource being provided

The address of the x402 service provider that will receive payment

The app offers extra features to optimize transactions via external services providers

Sends the transaction to the user's wallet for signing

The facilitator address paying the fees. If the fee facilitator address is not provided then the app assumes the fee payer is the end user.



UX: Live Updates

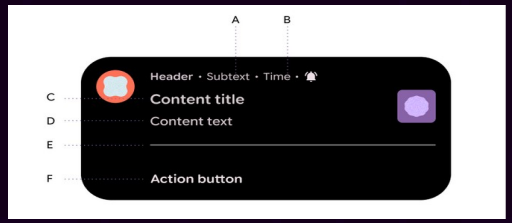
Platforms like **Android**, **iOS** and **HarmonyOS** implement live updates to give live feedback of ongoing events via notifications.

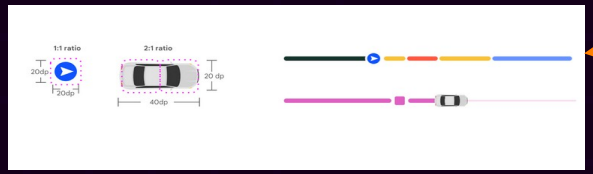
X402 resources can have long lived agentic actions that a user needs to be aware of.

Developers don’t need to implement frontends, they can leverage the live updates spec of the x402-URI specification to provide and the **Lagoon.Markets** dapp will handle all the mobile live update notifications.

UX: Live Updates x402 spec

A developer ships JSON as part of server-sent events which is transformed into the live updates UI described by Android 16 live updates spec. Pre-android 16 devices would simply show normal notifications

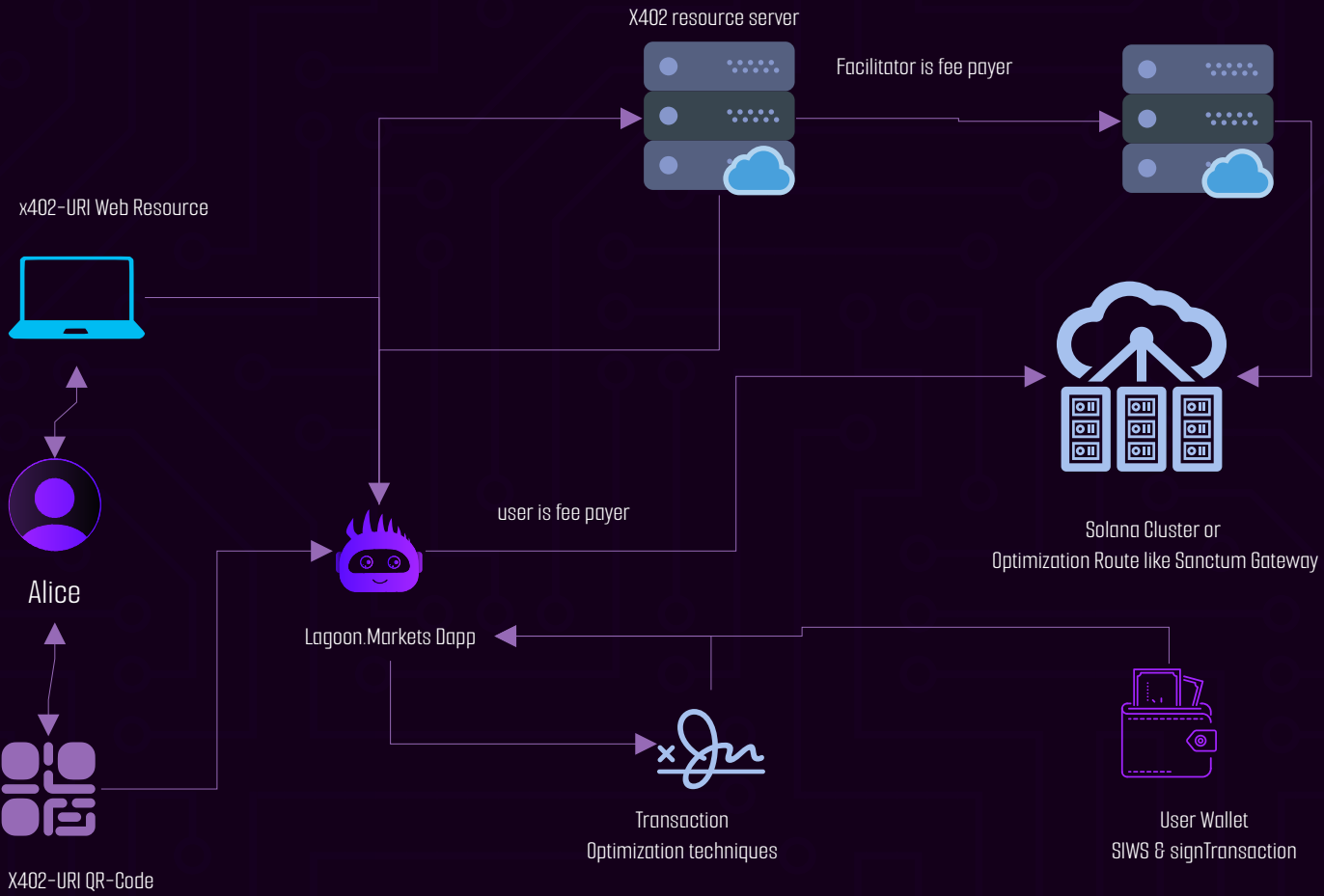





```
{
    "contentTitle": "<String>",
    "contentText": "<String>",
    "shortCriticalText": "<String>",
    "progress": {
        "point": <Int (u32::MAX)>,
        "color": "<String>"
    },
    "isProgressIndeterminate": <Boolean>,
    "actions": ["<String>"],
    "style": {
        "points": [
            {
                "point": <Int (u32::MAX)>,
                "color": "<String>"
            }
        ],
        "segments":[
            {
                "segment": <Int (u32::MAX)>,
                "color": "<String>"
            }
        ]
    }
}
```

https://developer.android.com/develop/ui/views/notifications/progress-centric

Technical Architecture



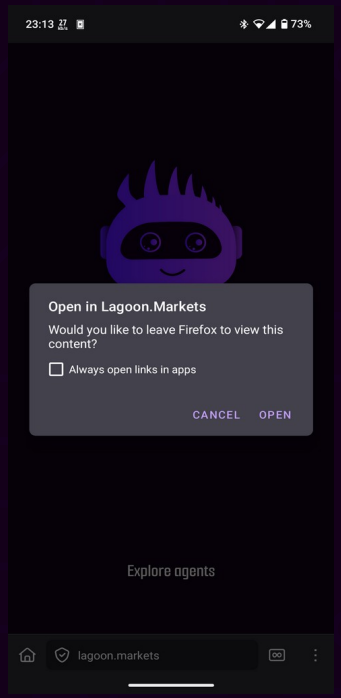
Dev Stack

1. Rust & Kotlin for Mobile development. Rust binding via uniffi

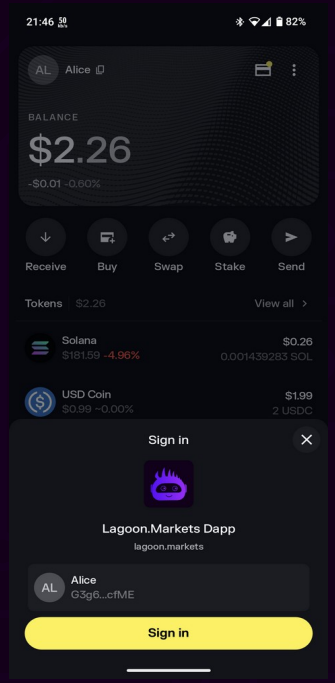
2. Rust for the server

3. Solana official SDKs for transaction building

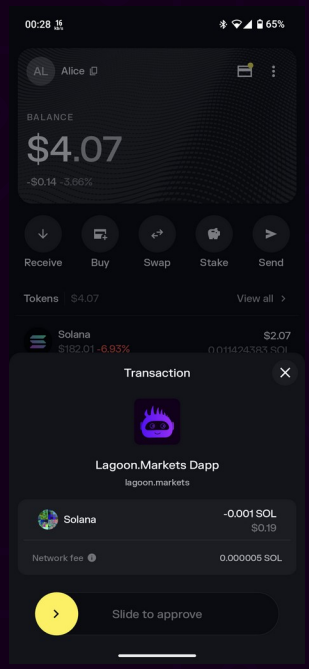
SMS Integration



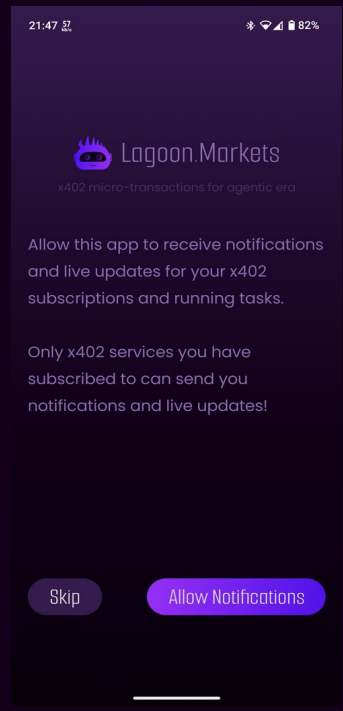
X402:// Deeplinks



Sign In With Solana



Sign Transaction



Notifications & Haptics

What was Built

- Onboarding user with Sign In With Solana registration allowing a user to construct transactions with that address

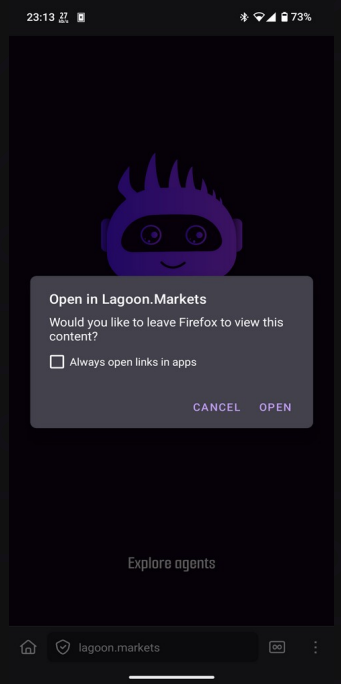
- Handling of x402:// URIs in Android

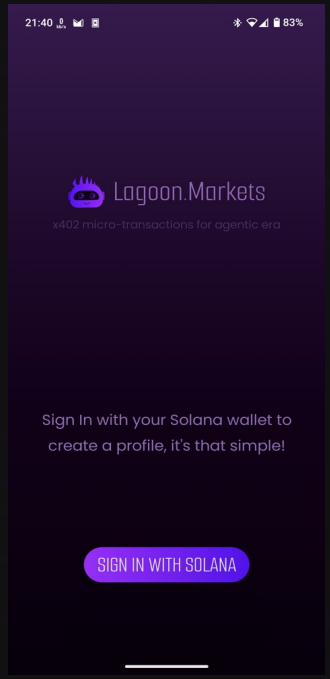
- Discovering of x402 resources from a x402:// URI via deeplinks

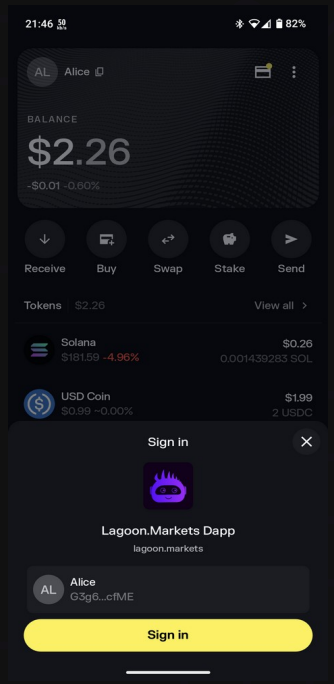
- Implemented of app notifications and android 16 live updates

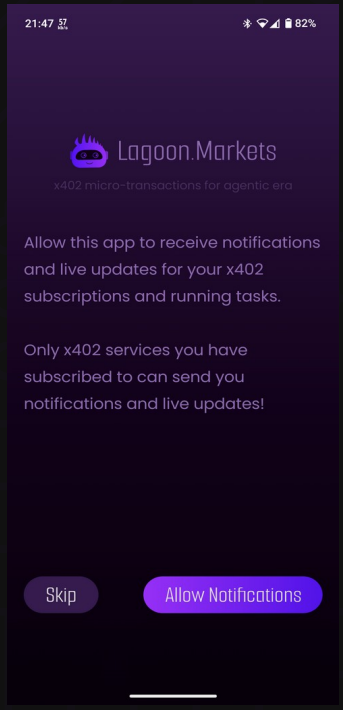
- Integrating server-sent events with android 16 live updates

User Experience Flow

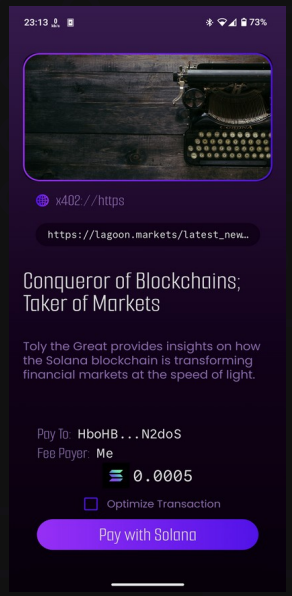


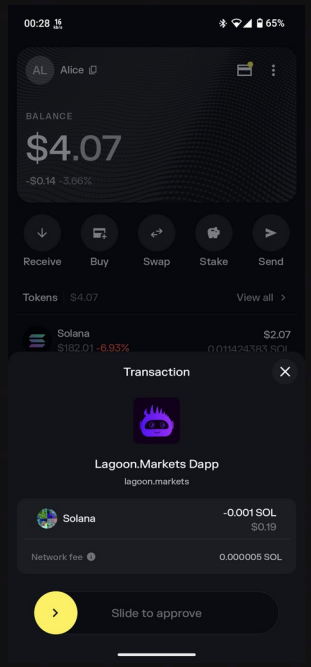




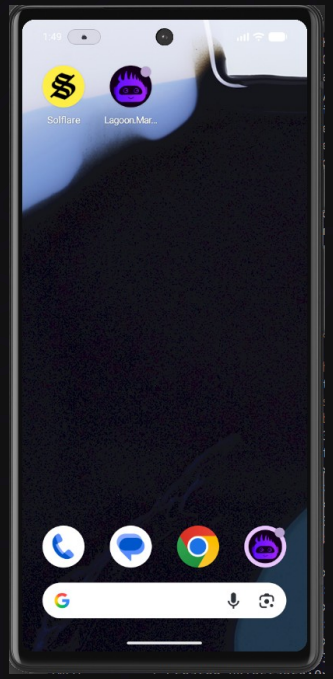


User Experience Flow: Transaction



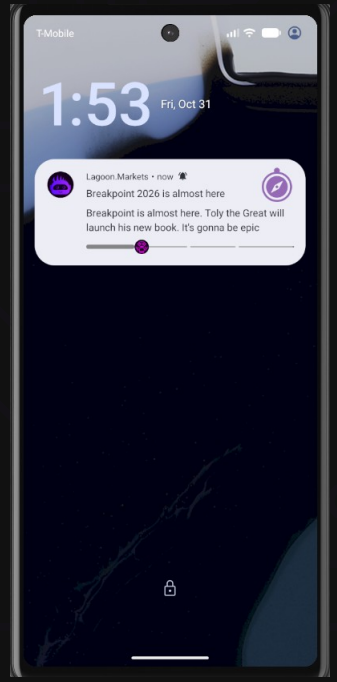


User Experience Flow: Live Updates



Short title

Expanded live updates



Lockscreen live updates

Coming Soon

- History of x402 resources a user interacted with

- In app scanning of x402-URI QR Codes

- A discovery view that lists x402 resources from well known sources like as a user feed

- More transaction optimization features for faster transaction times

- User to agent chats allowing a user to chat with an agent from within the app

- Subscription to AI agents especially for long running tasks

- More complex widgets like graphs and graphics for better agent interactions

Market Size & User Research

Version 1:

- Solana dapp store with 150,000+ users.
- Developers building AI agents for DeFi and Solana ecommerce checkout payments.

Version 2:

- Expands to users on Google play store and Apple app store targeting 10million+ wallet installs.
- Developers building AI agents for anything as long as it implements the x402-URI specification

Overall, the market size of mobile users is 10 million+ across Solana mobile, Android and iOS combined.

Making Money

The app also adds a transfer instruction that charges a user a micro-transaction for each successful request.

Team

Charles Chege

Github Profile: @448-OG

Building open source Rust tools for Solana devs at JamiiDao

Github: @JamiiDao

Resources

Project Page

Download Mobile Dapp

Visit lagoon.markets to explore x402 resources

x402-URI specification (extension of x402 protocol using URIs)

X402 protocol spec

Project implementation of x402 protocol