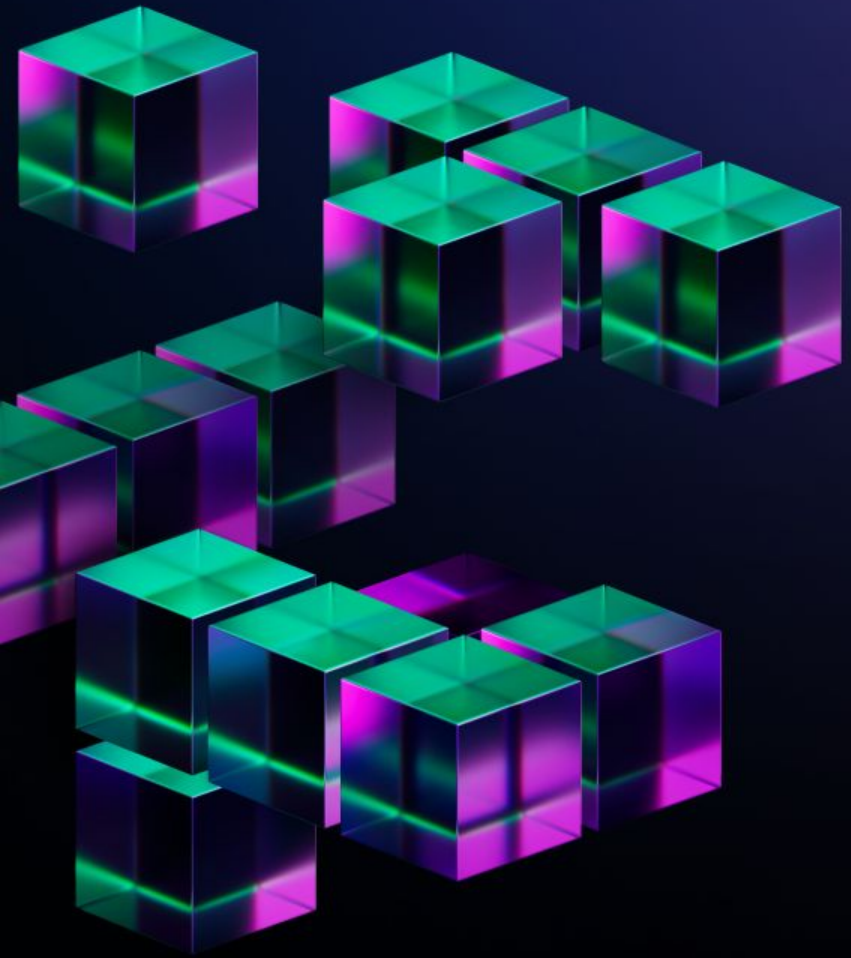
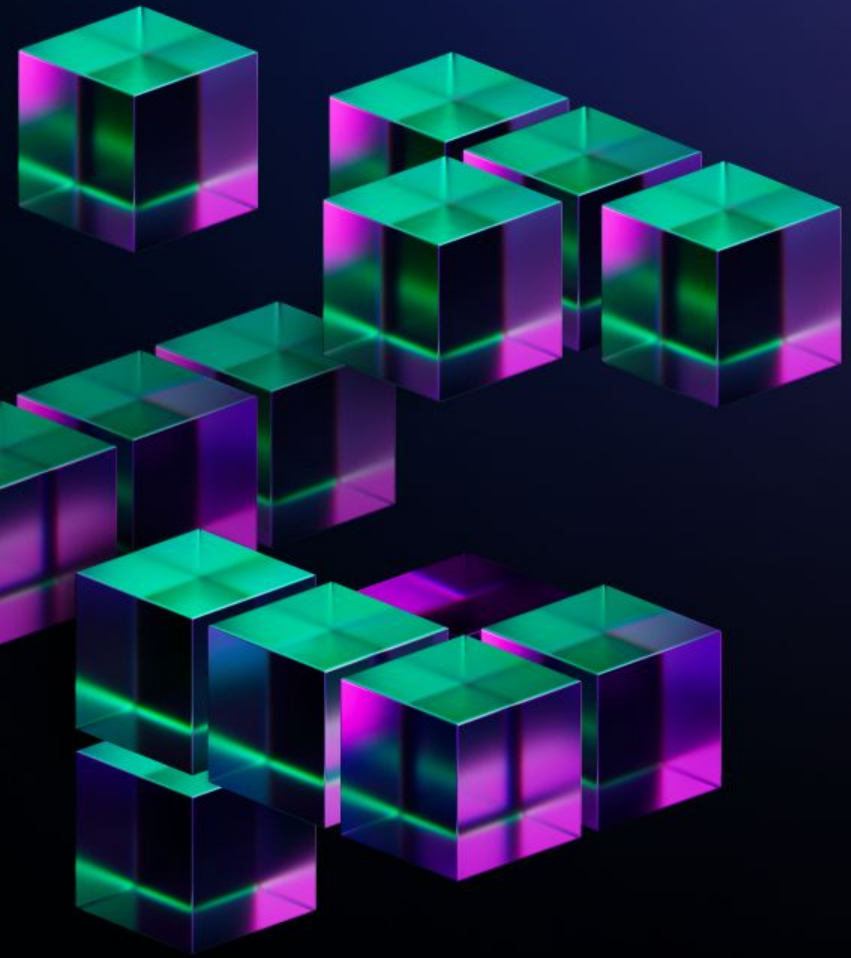


Solana Actions

💥 with blockchain links 💥



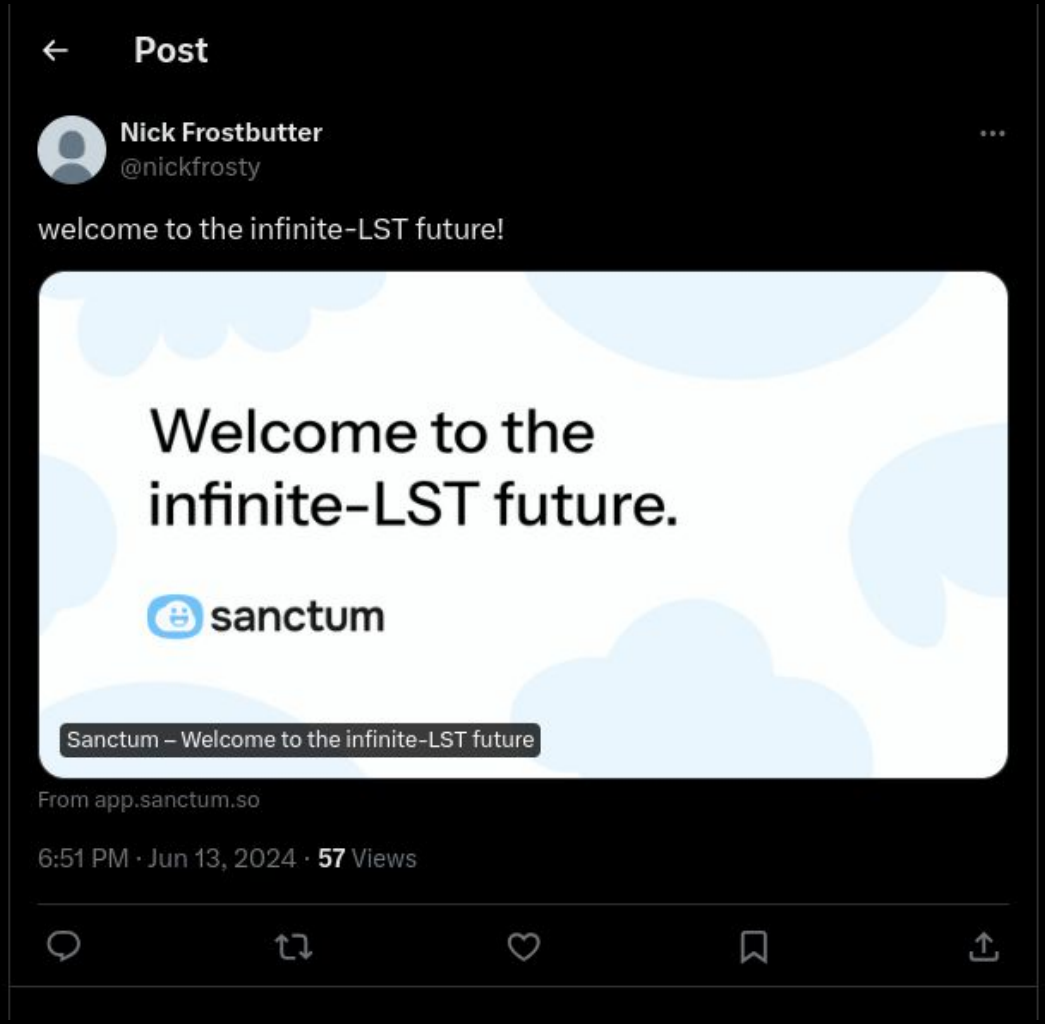
imagine...

Sharing a simple tweet?

SOLANA

You expect this...

SOLANA



Explore
Notifications
Messages
Grok
Premium
Lists
Bookmarks
Communities
Articles
Profile
More
Post

welcome to the infinite-LST future!

app.sanctum.so

Buy hSOL

Buy hSOL with SOL. Choose a SOL amount from the options below, or enter a custom amount.

1 SOL | 5 SOL | 10 SOL

Enter a SOL amount | Buy hSOL

6:51 PM · Jun 13, 2024 · **56** Views

goneinaflash
@gonein400ms
Follow

Explore Beta

Kids Grow Up Fast: A Parent's Chronicle
17 hours ago · Parenting

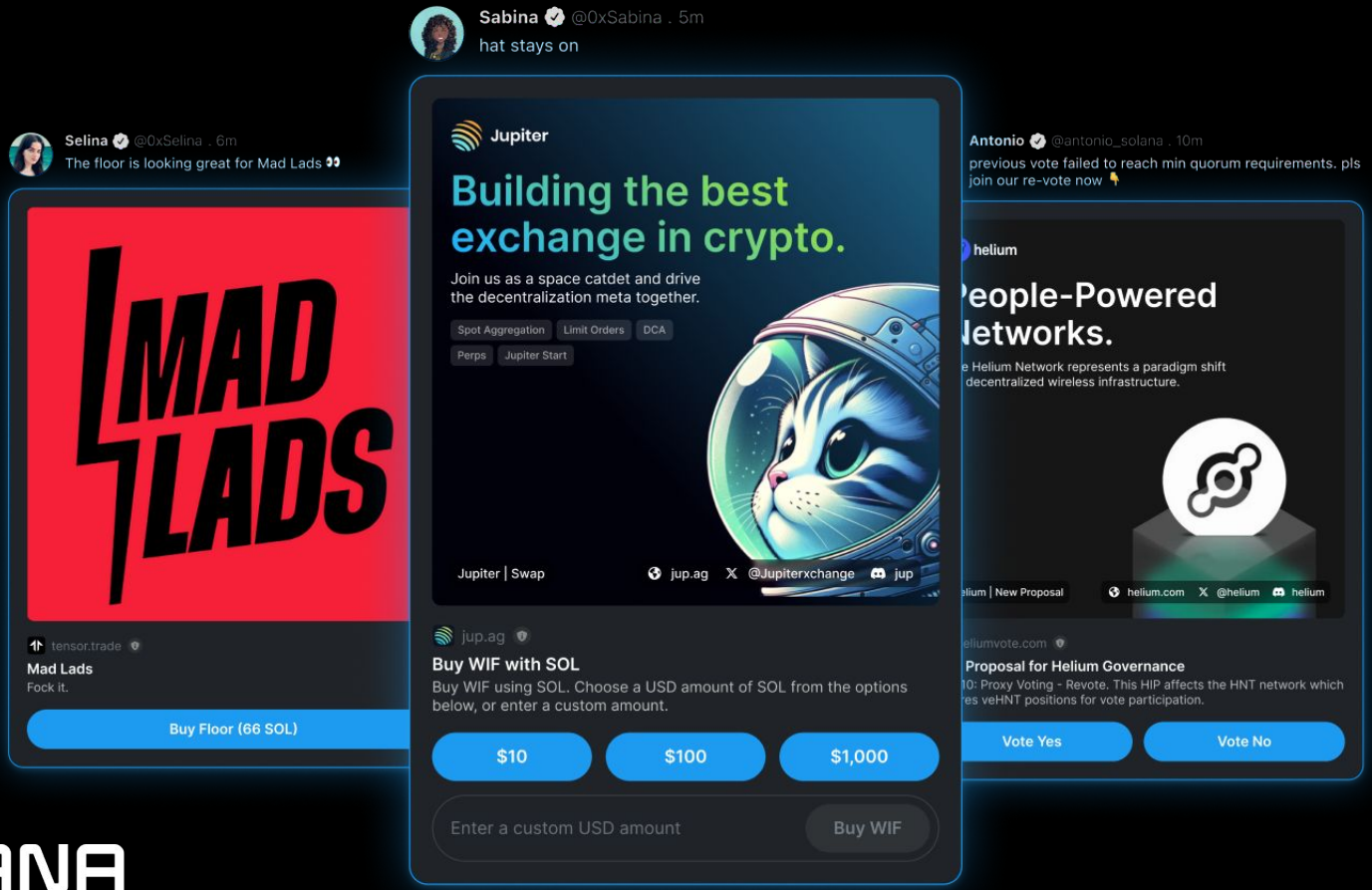
Solana Hits 200-Day EMA: Rally or Retreat?
10 hours ago · Finance

Heat Pumps vs ACs: Cooling or Warming, Choices Abound
19 hours ago · Technology

SEC's Crypto Chief David Hirsch Resigns Amid Regulatory Battles
22 hours ago · Finance

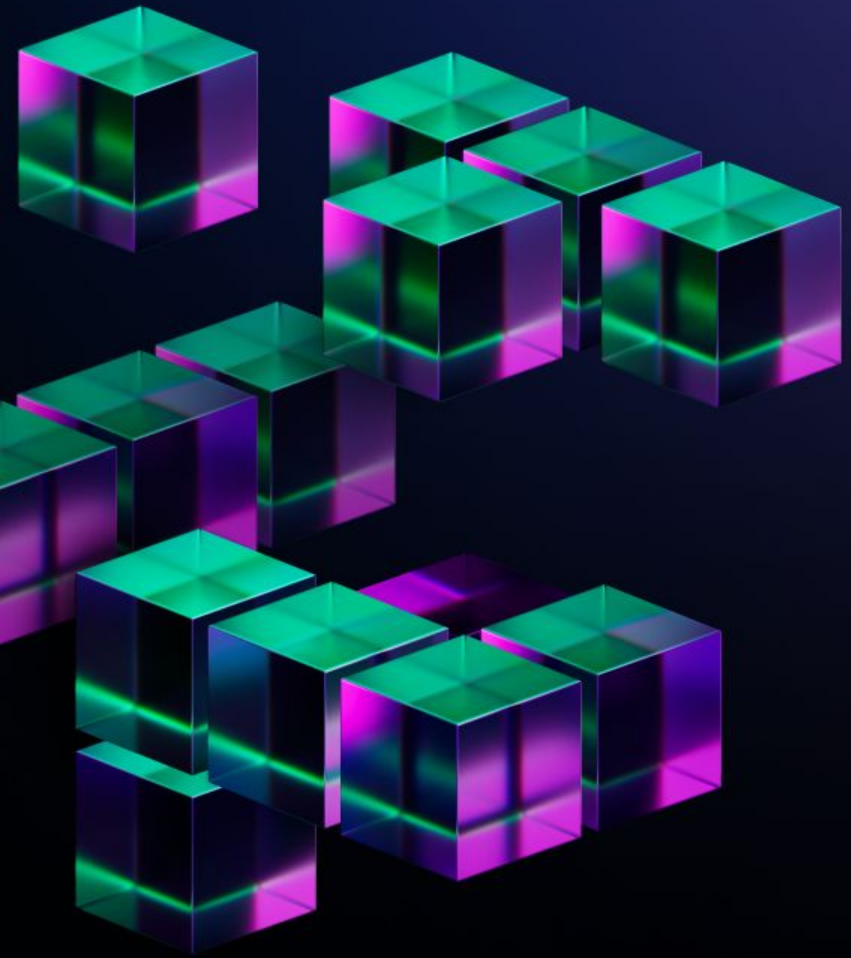
Show more

Terms of Service Privacy Policy Cookie Policy
Accessibility Ads info More ··· © 2024 X Corp.



SOLANA

Starting with Twitter



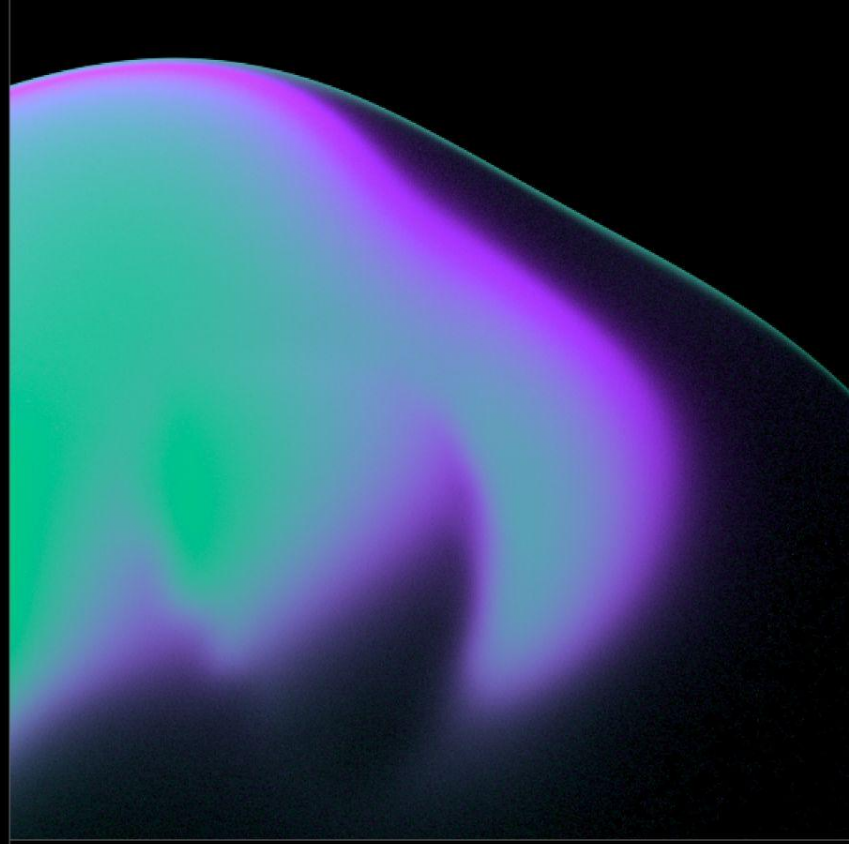
blinks == frontend

Actions == backend

Terminology

Solana Actions - are specification-compliant APIs that return transactions on the Solana blockchain to be previewed, signed, and sent to the network.

- QR codes
- Buttons + widgets
- Websites
- etc



© ALL RIGHTS RESERVED

Solana Actions

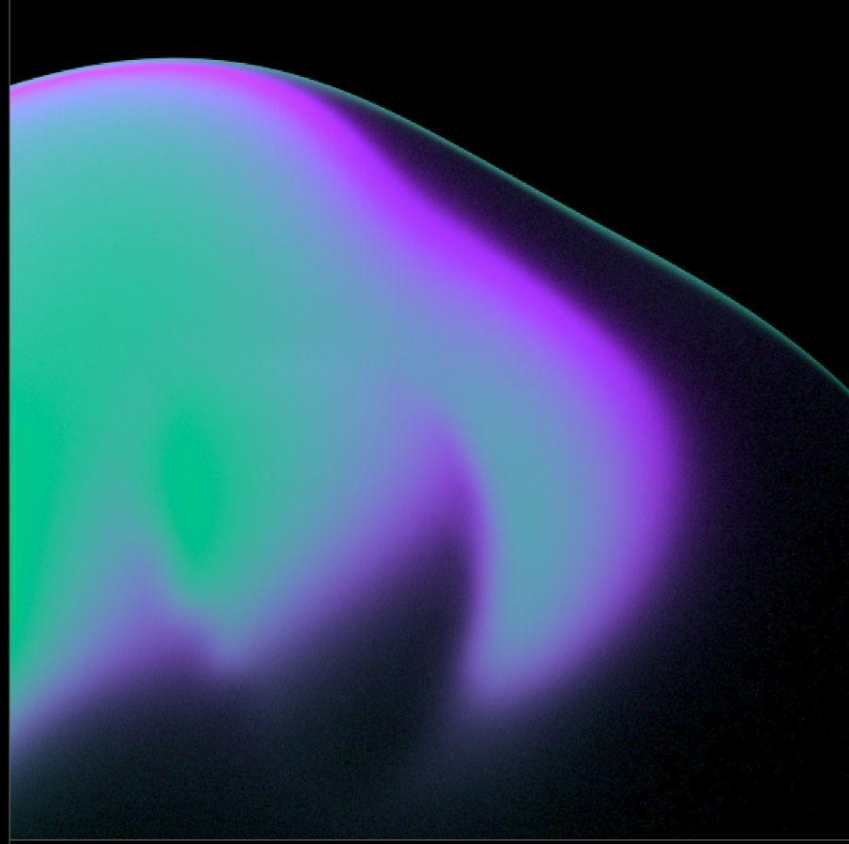
Deliver signable transactions (and eventually messages) to the user

- Public URLs on the internet
- Accept GET and POST requests
- Return metadata
- Return a signable transaction
  - This can do anything!
- Safety + Security
  - Actions come from domains (just like dApps)
  - Transactions are still simulated via wallets

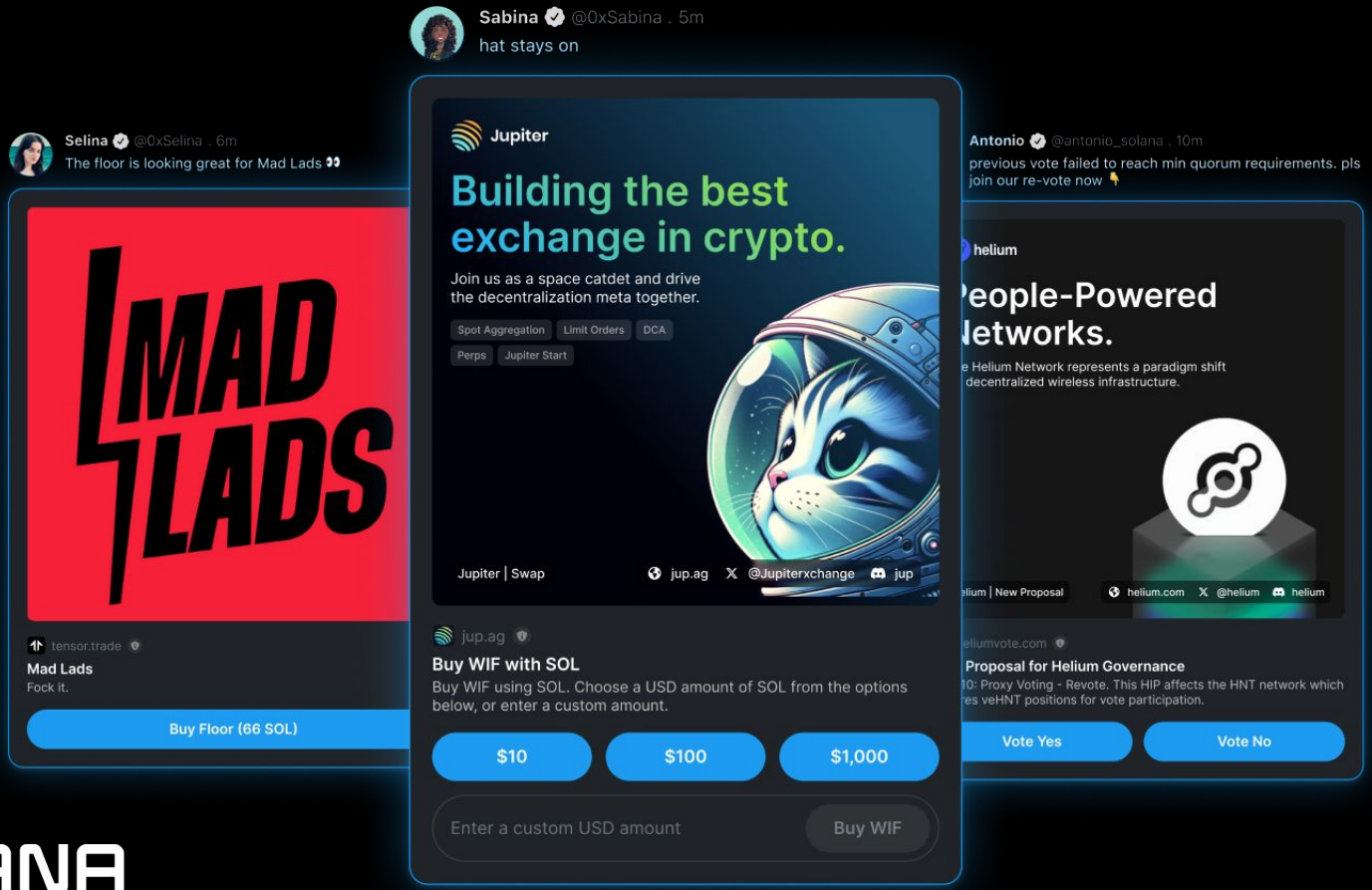
© ALL RIGHTS RESERVED

Example Actions

- Voting on governance proposals
- Staking SOL
- Swapping digital assets
- Claiming digital collectibles
- Donating to support a creator, quadratic funding platform, or charity



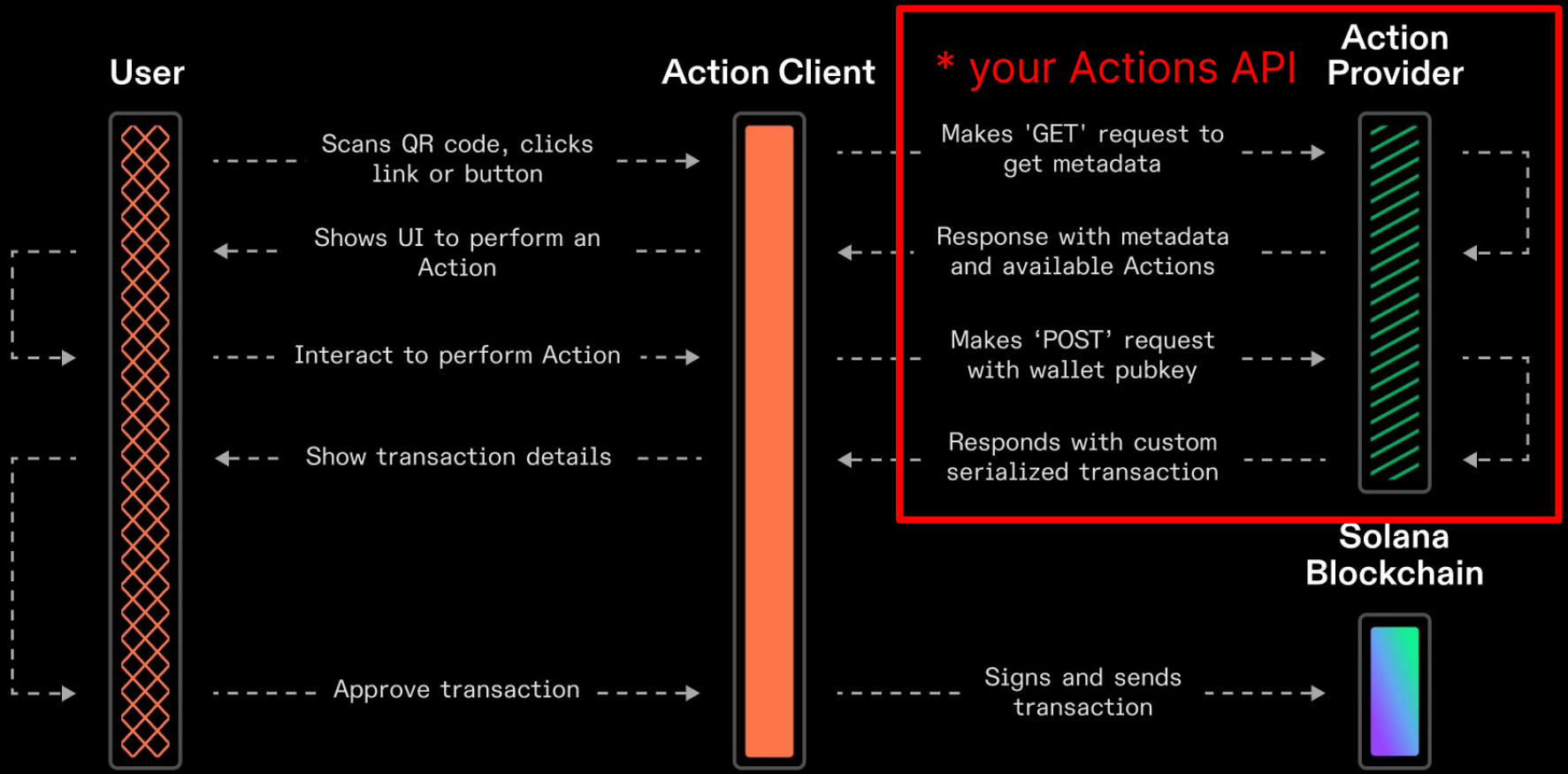
© ALL RIGHTS RESERVED

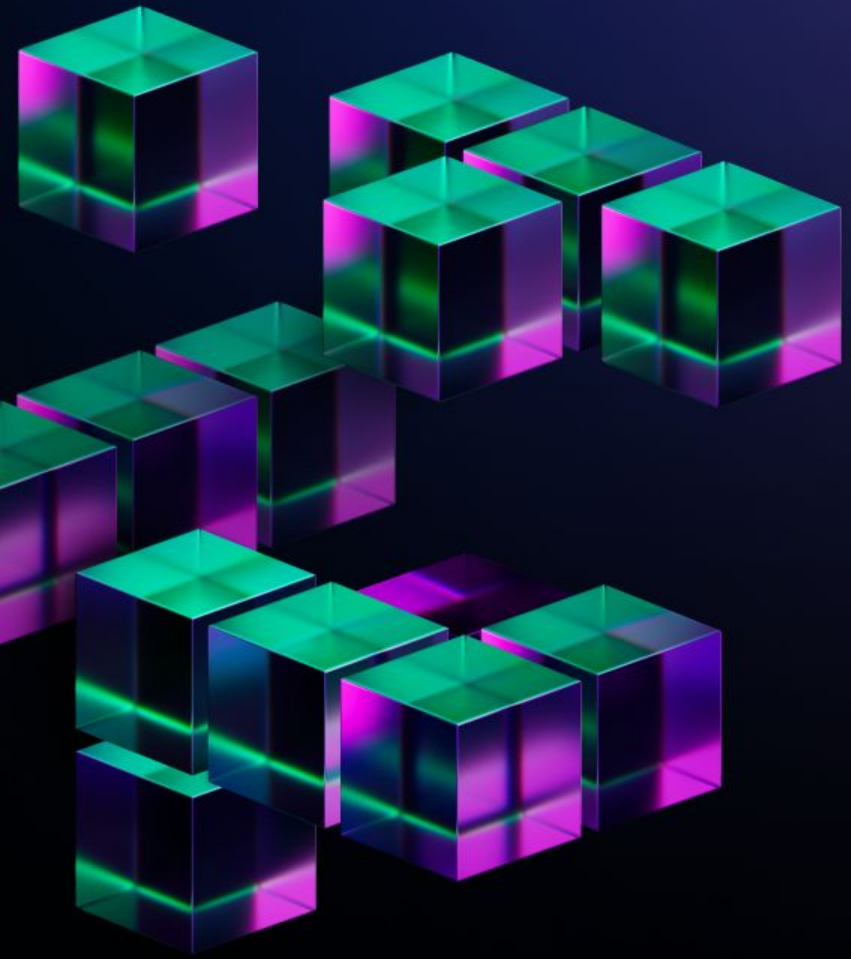


SOLANA

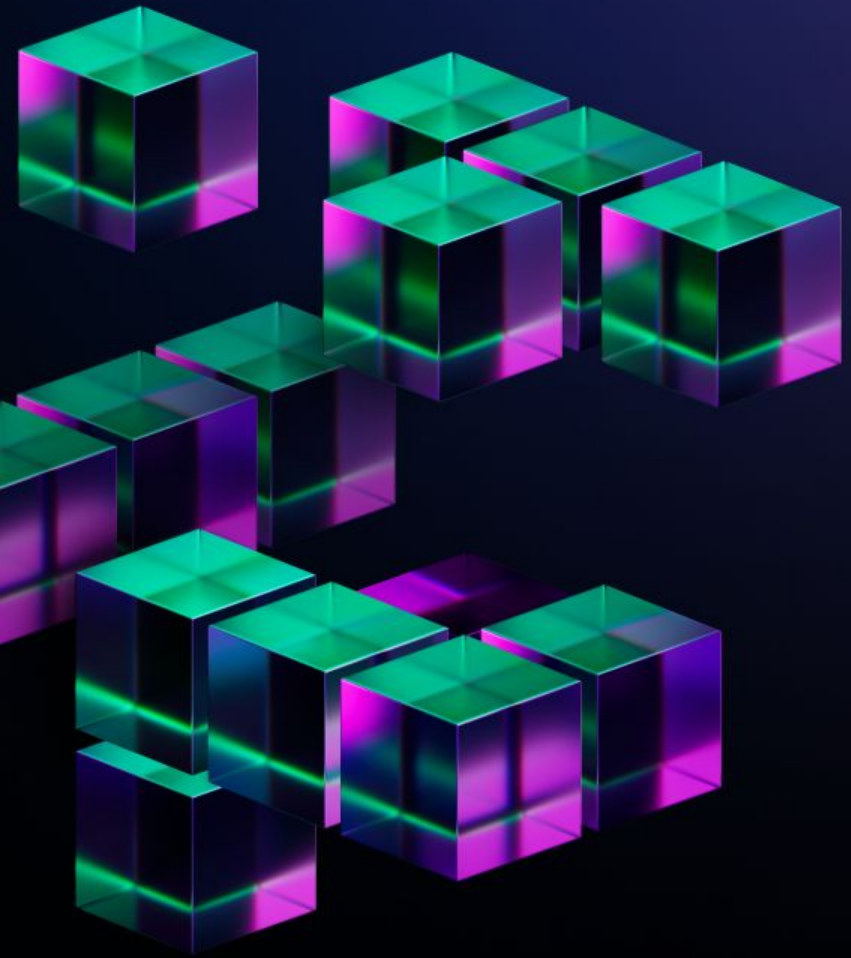
Starting with Twitter

Action Execution Lifecycle





What are blinks?



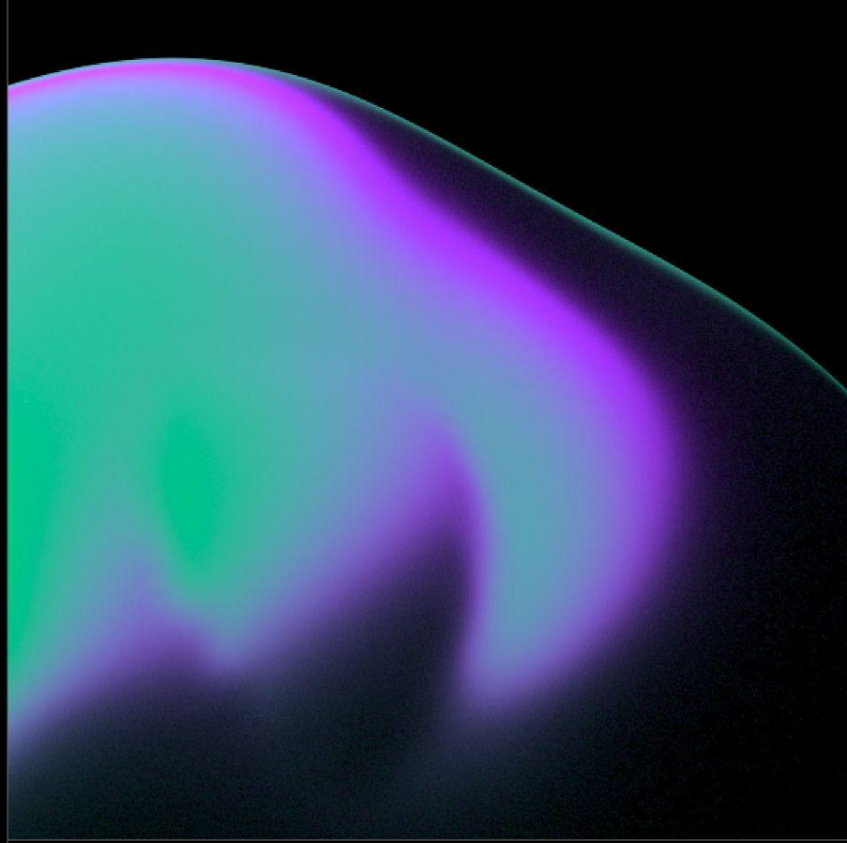
blinks == frontend

Actions == backend

Terminology

Blockchain links – or **blinks** – turn any Solana Action into a shareable, metadata-rich link.

Blinks allow users to execute blockchain transactions directly from URLs, making decentralized applications accessible from any platform or device.



blinks are like embedded dApp widgets. sort of.

© ALL RIGHTS RESERVED

Blinks

client applications that introspect Action APIs and construct user interfaces around interacting with and executing Actions

- ❏ Client detects Actions and displays standardized UI
- ❏ Currently via browser extensions that inject this UI
- ❏ To be a blink:
  1. `solana-action:` uri
  2. actions.json file
  3. interstitial website using the `action` query parameter

© ALL RIGHTS RESERVED

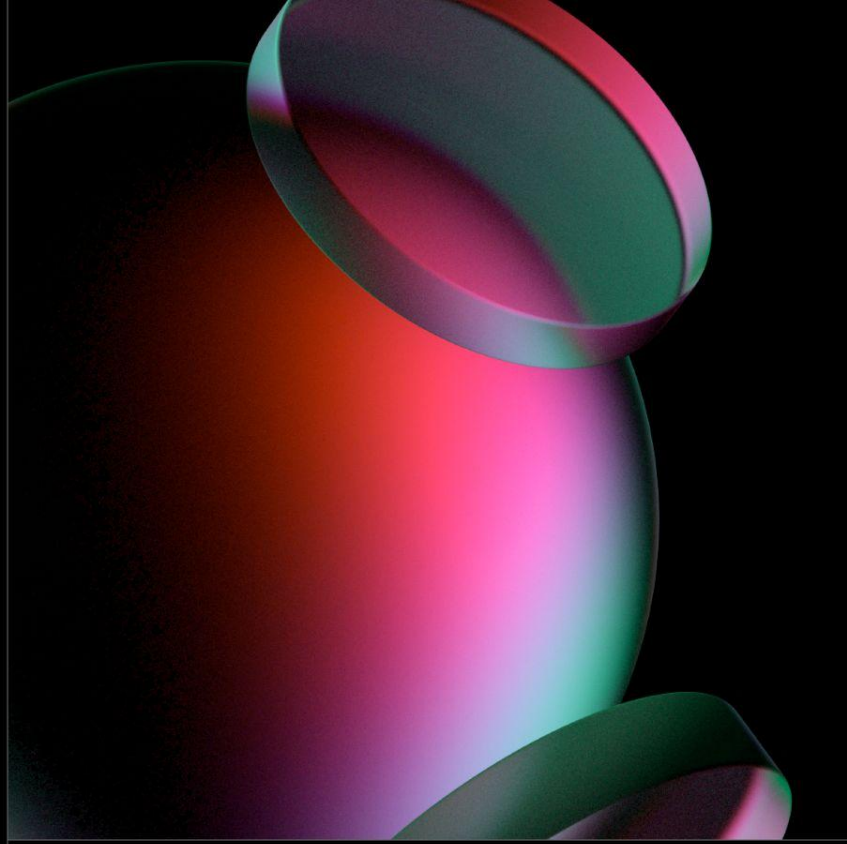
`solana-action:` uri scheme

```
solana-action:https://actions.dialect.to/api/jupiter/swap/USDC-SOL
```

- No fallback website
- Least supported
- Least recommended

actions.json

- Map normal urls on your website to Action API endpoints
  - local relative links
  - external absolute links
  - wildcards are supported
- Allow anyone to share a normal link to your website, which will be converted to a blink
- Stored at the root of your domain
- Your website is the fallback UI



© ALL RIGHTS RESERVED

Example actions.json file

- Jupiter Swap website: https://jup.ag/swap/USDC-SOL
  - this is the link you share on social media and across the internet
- actions.json file at https://jup.ag/actions.json
- Maps (using wildcards):
  - `/swap/USDC-SOL` to
  - `https://actions.dialect.to/api/jupiter/swap/USDC-SOL`

actions.json file

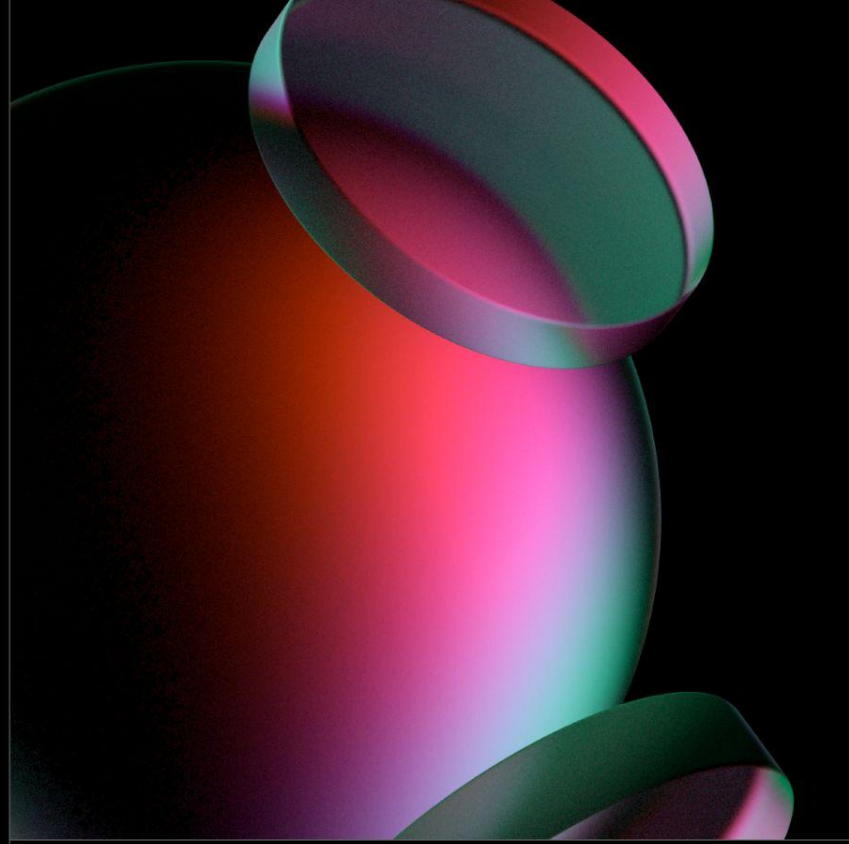
```
{
  "rules": [
    {
      "pathPattern": "/swap/**",
      "apiPath": "https://actions.dialect.to/api/jupiter/swap/**"
    },
    {
      "pathPattern": "/collection/*",
      "apiPath": "/api/actions/collection/*"
    }
  ]
}
```

SOLANA

- External absolute urls
- Internal relative urls
- Wildcards `*` and `**`

Interstitial website

- Uses `action` query parameter
  - Value of `solana-action:` uri
- Interstitial website is the fallback UI
- Standardized UI
- Allow you to build your Actions without building your own frontend UI and wallet connect flow



© ALL RIGHTS RESERVED

Interstitial website

```
https://dial.to/?action=solana-
action:https://actions.dialect.to/api/jupiter/swap/USDC-SOL
```

- `action` query parameter
- value of the `solana-action:` uri scheme
- Interstitial is the fallback
- No need to build your own UI and wallet connect

SOLANA

SOLANA 2023

How do Blinks unfurl/render?

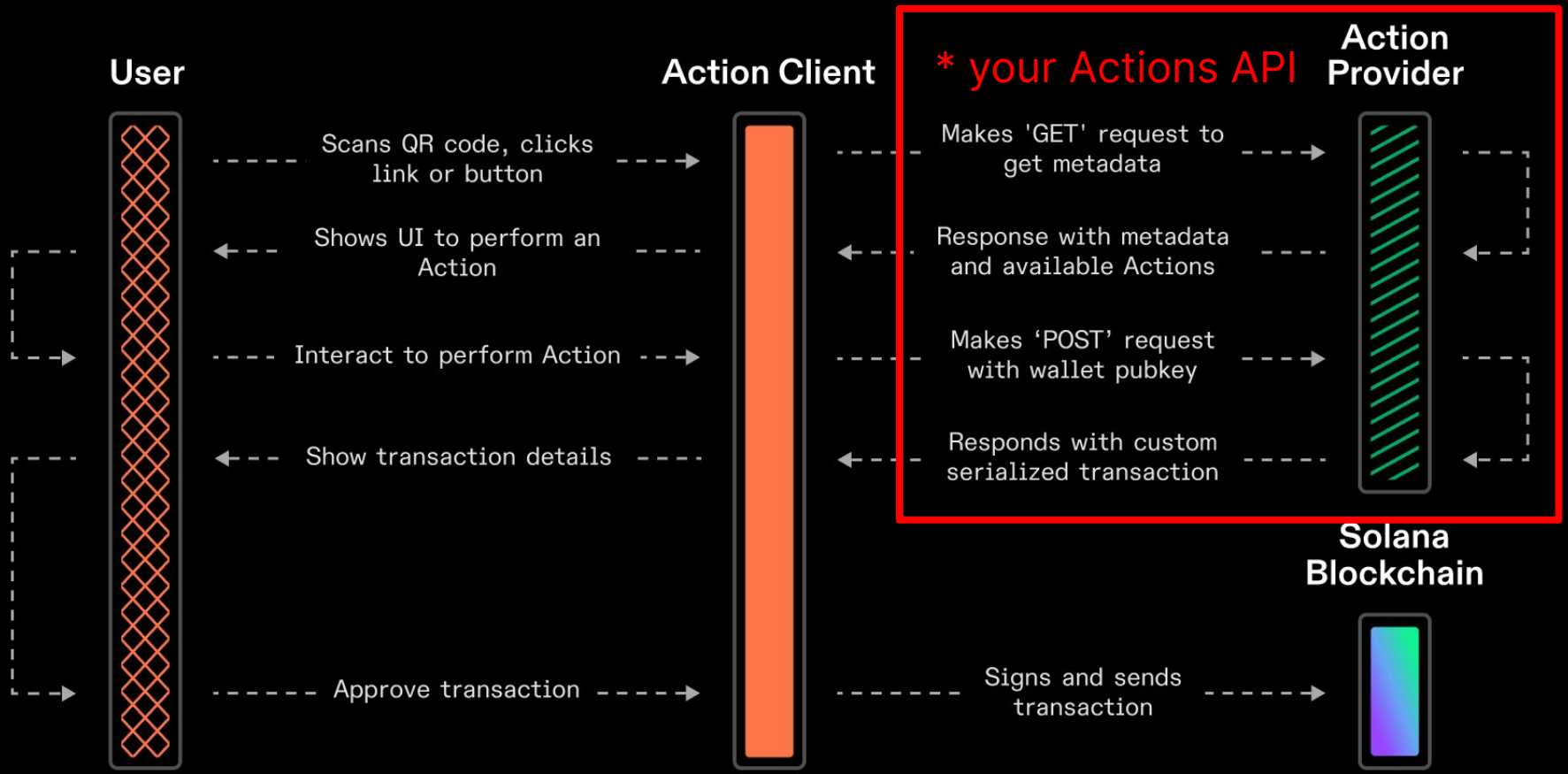
unfurl - unroll, unfold, or spread out or be unrolled, unfolded, or spread out from a furled state. “unfurl a banner”

- ❏ Blink client detects link on page
- ❏ Currently on X.com:
  1. Link must show social image
- ❏ Client checks:
  1. actions.json file or `action` query param
  2. Dialect’s registry
  3. Fetch metadata
- ❏ If all checks pass ⇒
  - render blink
- ❏ If any check fails ⇒
  - everyone sees a regular link
  - (just like before)

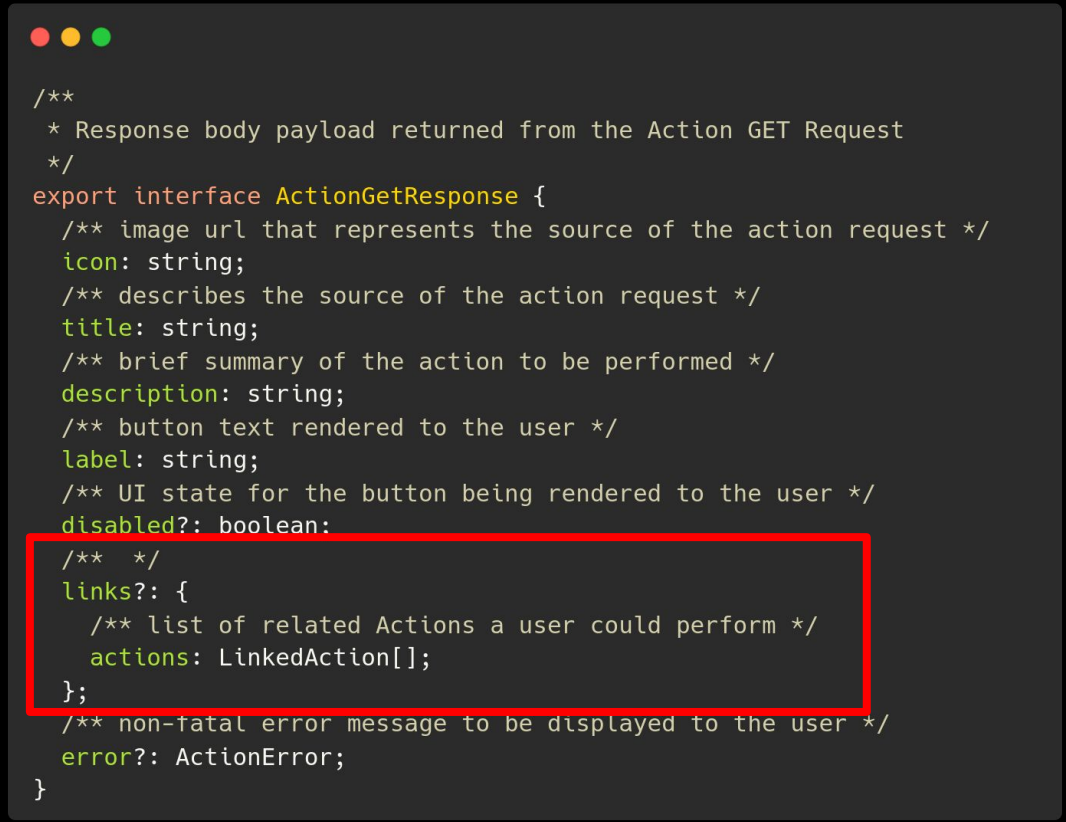
© ALL RIGHTS RESERVED

Let’s get nerdy about Actions

Action Execution Lifecycle



GET Response



```ts
/**
 * Response body payload returned from the Action GET Request
 */
export interface ActionGetResponse {
  /** image url that represents the source of the action request */
  icon: string;
  /** describes the source of the action request */
  title: string;
  /** brief summary of the action to be performed */
  description: string;
  /** button text rendered to the user */
  label: string;
  /** UI state for the button being rendered to the user */
  disabled?: boolean;
  /**  */
  links?: {
    /** list of related Actions a user could perform */
    actions: LinkedAction[];
  };
  /** non-fatal error message to be displayed to the user */
  error?: ActionError;
}
```

SOLANA

POST Request

```ts
/**
 * Response body payload sent via the Action POST Request
 */
export interface ActionPostRequest {
  /** base58-encoded public key of an account that may sign the transaction */
  account: string;
}
```

SOLANA

POST Response

```ts
/**
 * Response body payload returned from the Action POST Request
 */
export interface ActionPostResponse {
  /** base64 encoded serialized transaction */
  transaction: string;
  /** describes the nature of the transaction */
  message?: string;
}
```

SOLANA

Let’s look at code!

https://github.com/solana-developers/solana-actions/blob/6a34a8d99af87119d1ce71ab5321ebc90a531a69/examples/next-js/src/app/api/actions/transfer-sol/route.ts

Solana Actions SDK

```
npm install @solana/actions
```

SOLANA

More Actions and Blinks

YouTube Playlist

Getting Started Guide

SOLANA

More Resources

- Docs - https://solana.com/docs/advanced/actions
- Code Examples - https://github.com/solana-developers/solana-actions/tree/main/examples
  - different frameworks - including NextJS, Express, and Axum
- Blink Inspector Tool (debug your blinks/actions)
  - https://www.blinks.xyz/inspector
- Awesome Blinks - https://github.com/solana-developers/awesome-blinks
  - If you build something awesome, open a PR

Thank You!

Ask questions on StackExchange:
https://solana.stackexchange.com

Follow *@solana_devs* on Twitter / X:
https://x.com/solana_devs

Follow *@nickfrosty* on Twitter / X:
https://x.com/nickfrosty

Developer Resources

© ALL RIGHTS RESERVED