Lesson 15

Today's topics

- Compressed NFTs
- Debugging Solana Programs
- Pyth Oracle Network
- Eclipse
- Anchor
    - PDA and CPI
    - Further examples

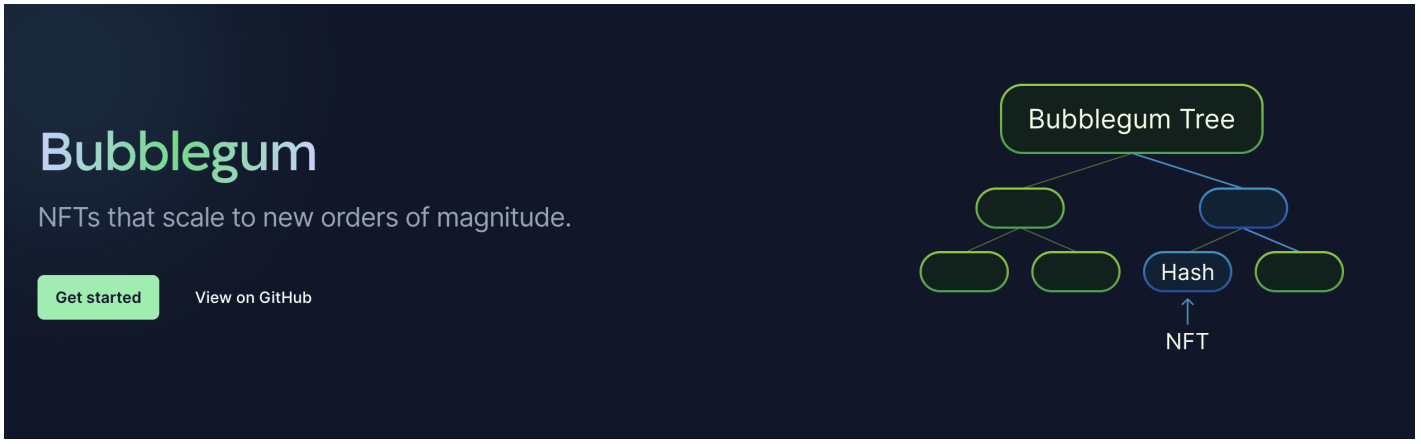
News

Hacker jailed for stealing from Solana DEXs
See article

Compressed NFTs

See Docs
We have seen that NFTs with their metadata is stored on chain.
This can be very expensive if you need to mint large numbers of NFTs
Compressed NFTs were introduced to reduced this cost, they do so by storing the metadata off chain.
They use state compression and merkle trees to store data off chain in a special ledger, but it is still made available for consensus.

Metaplex have created the Bubblegum project to handle compressed NFTs



Compressed NFTs store all of their metadata in the ledger, instead of in traditional accounts like uncompressed NFTs, so indexing services are needed to retrieve the metadata.
This indexing service is available from RPC providers such as

- Helius
- Triton
- SimpleHash

Creating a compressed NFT

- create an NFT collection (or use an existing one)
- create a concurrent merkle tree (using the `@solana/spl-account-compression` SDK)
- mint compressed NFTs into your tree (to any owner's address you want)

Transfering the NFT

1. get the NFT "asset" information (from the indexer)
2. get the NFT's "proof" (from the indexer)
3. get the Merkle tree account (from the Solana blockchain)
4. prepare the asset proof (by parsing and formatting it)
5. build and send the transfer instruction

For a walkthrough of this process see Docs

> Using compressed NFTs can reduce the cost of minting by up to 99%

Debugging Solana Programs

For a guide to debugging Solana programs see Cookbook

Pyth Oracle Network

See site

Pyth Network is an oracle project built on Solana. It focuses on bringing high fidelity market data from trust worthy data provider in a high speed manner.

Market data, as well as a confidence level of that market price are submitted by data provider to be aggregated on Solana chain.
It also has cross chain compatibility using Wormhole as a bridge between ERC20 and SPL tokens.

Example data feed

Integrating a data feed

See Docs
Pyth on Solana Docs

Developers should integrate Pyth into both their on-chain and off-chain code:

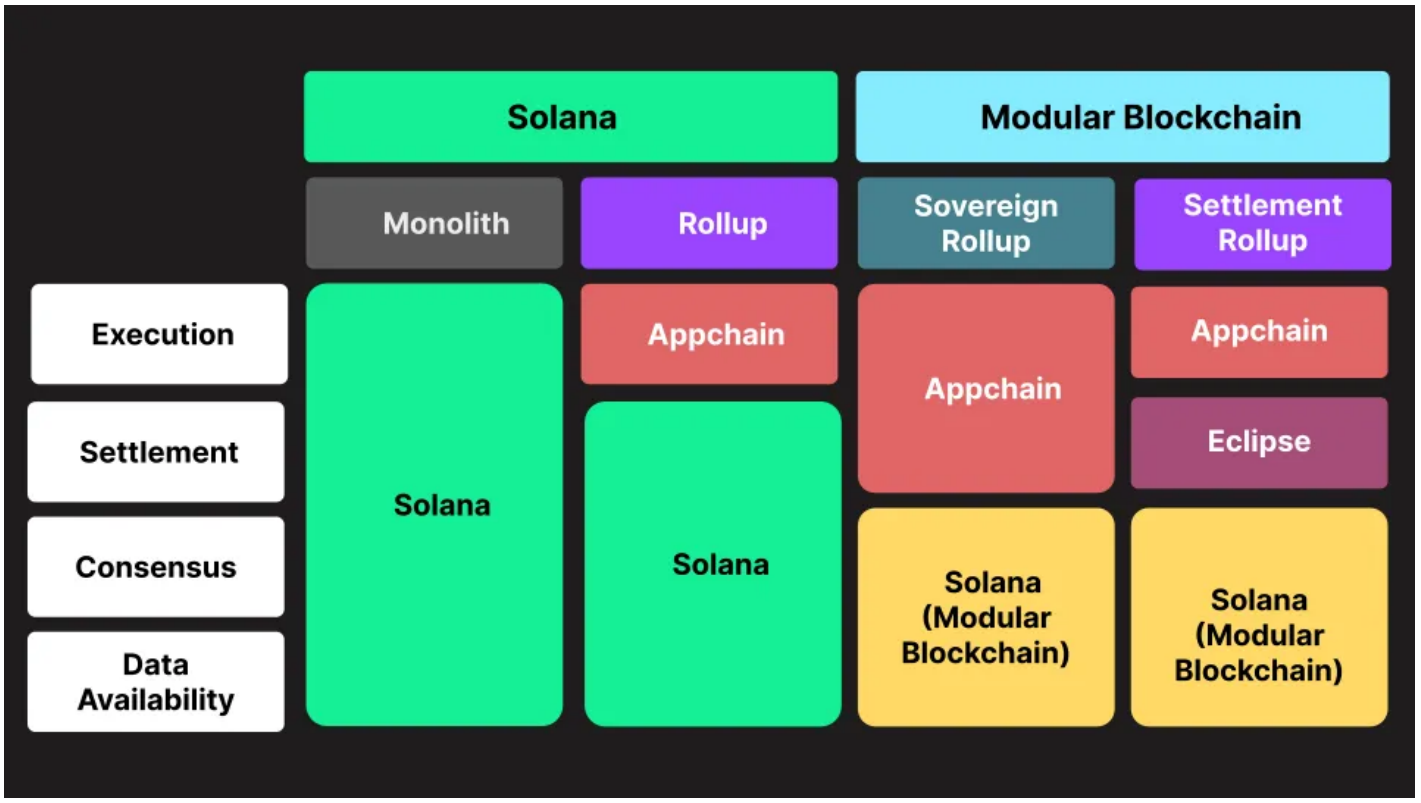
1. On-chain programs should read prices from the Pyth program deployed on the same chain
2. Off-chain frontends and jobs should include Pyth price updates alongside (or within) their application-specific transactions.
3. Pyth provides ecosystem-specific SDKs to assist with both the on- and off-chain pieces of the integration. The easiest way to use Pyth price feeds is to integrate the appropriate SDKs into your application.

Pyth on Solana

Example Anchor code and lib.rs

Eclipse and App specific rollups

See Eclipse Docs
See Article
See Bankless Podcast



Eclipse is a rollup as a service, as an IBC-enabled rollup on Celestia, which is the data availability layer.
Eclipse allows you to spin up your own rollup with the Sealevel VM with a base layer provided by CelestiaOrg , Polygon Avail , or Eigen Layer.
There are also options to run an EVM or the Move VM.

Advantages to using Eclipse

1. **Maximum customisability:** Any dApp can adjust its block times, subsidise gas, or restrict who can use your blockchain.

2. **Shared security:** dApps don’t have to worry about managing infrastructure or bootstrapping validators, because it borrows security from the underlying Layer 1.
3. **Scale horizontally:** This means that dApps can always spin up another execution layer, giving the power of Solana all to themselves.
4. **Cheaper running costs:** The data availability layer doesn’t need to support execution, and therefore the cost of running a node is cheaper promoting decentralisation.

The Eclipse mainnet should be available in Q4 2023

PDA in Anchor

See Anchor book explanation

CPI in Anchor

See Anchor book explanation

Further Anchor examples

Election program with Anchor

Example program in Anchor to run an election
See Article

Staking NFT

See article