



Multi-chain NFT Metadata Caching

World of opportunities

INTRODUCTION

- A high performance NFT Metadata caching layer for all EVM/Non-EVM blockchain
- Easy to set up
- Near-real time processing
- Fault-tolerant

Key Benefits

01

Reduce Development Time

Developers can focus on core business feature and just integrate this metadata caching service

02

Convenience

Chain agnostic service makes it right fit for working with any dApp

03

Reduce Costs

Team plans to run it as SAAS service, which will reduce overall project cost


04

Greater Visibility

Project is open source hence any developer can propose feature updates, bug fixes

Why world needs a smart and high performance metadata service?

NFT Marketplace



Show NFT metadata in nft marketpalces

PFP use case

PFP is just another use cases where applications need the metadata

Gaming

Query NFTs by category, sub-category. Render skins, avatars based on NFT metadata. Render game logic by token metadata.

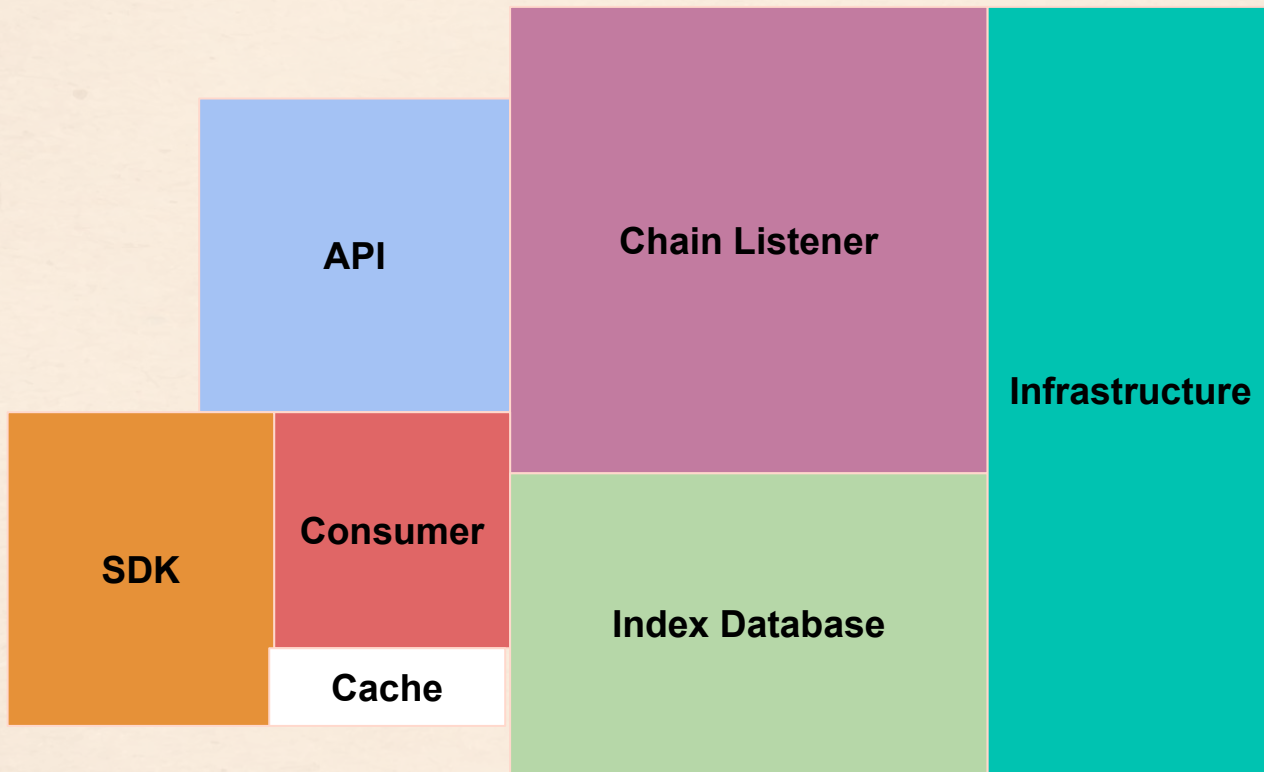
Metaverse

Display NFTs data in art museums, concerts within the metaverse, near real-time and low latency APIs are just what is needed.

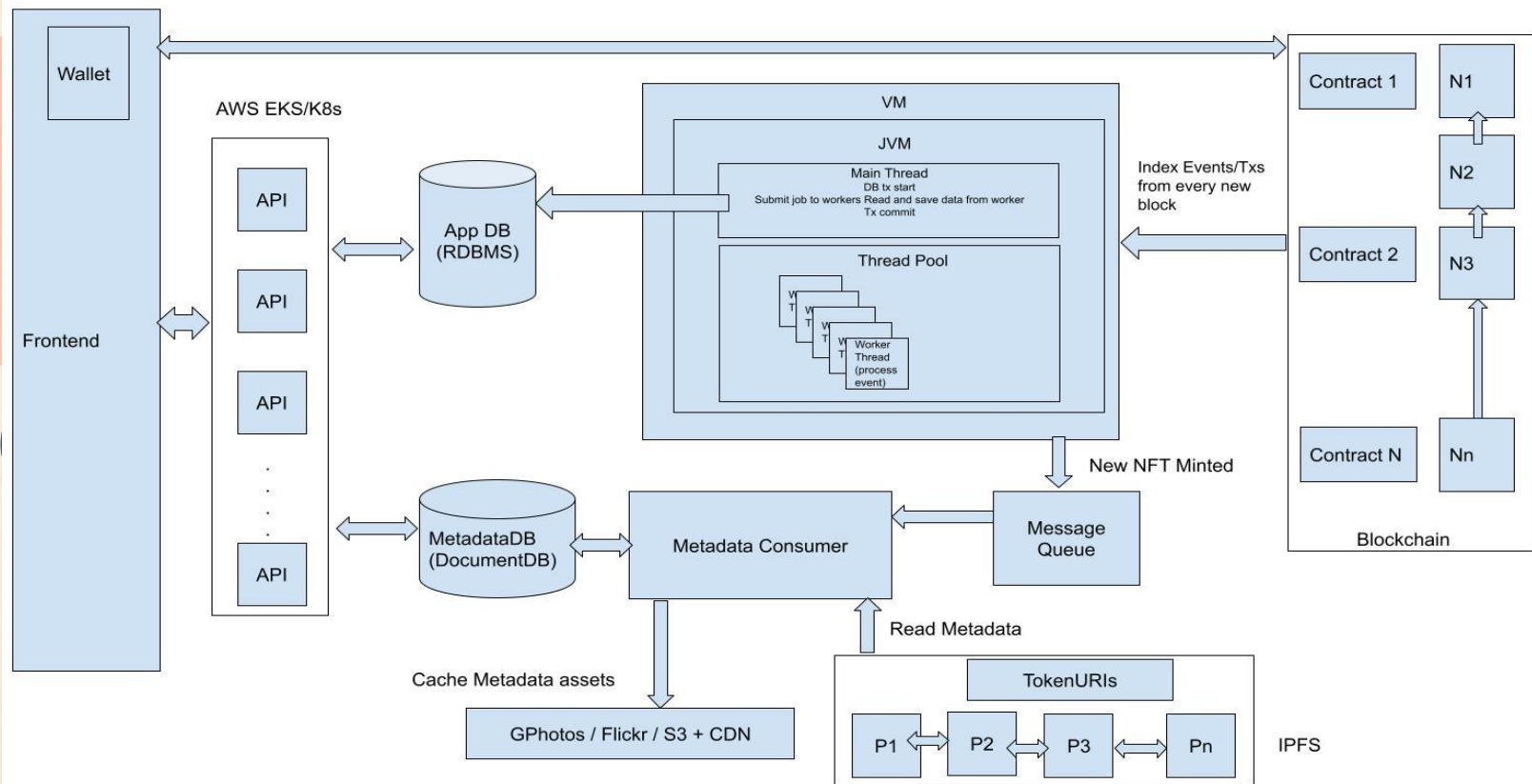
The background is a light beige textured surface. It is decorated with several geometric shapes: a large light pink rectangle on the left, a dark blue circle on the left, a dark blue square in the top right, a light orange square in the top left, a light orange square in the top right, a light orange square in the bottom left, a light orange square in the bottom right, a light pink rectangle in the bottom left, and a row of eight dark blue triangles along the bottom right.

One High
Performance
Service built for
entire ecosystem

Building Blocks Of Metadata Service



High Level Architecture



Features

Non-Fungible Tokens

- Inquire about specific NFT contract information
- Retrieve all token information of specific NFT contract
- Query specific NFT token information
- Query NFT by category/metadata
- Metadata caching

Key Highlights of architecture

01

High Performance

Core indexer listener would be written in java/Go multithreaded architecture for faster computing

03

High Availability

Leader election is at the core of system design. If active listener dies/hangs, standby would immediately takeover.

02

Fault Tolerant

It takes care of persisting and processing events block by block leaving no chance of missing any events event if process dies.

04

Chain reorgs

Block hashes are lazily verified and if at any point chain reorgs is detected it would immediately discard forward blocks and start from where reorg is detected



THANKS!

For any further questions please send email to narendra in
telegram @nindz or twitter @Narendra_SoniK