Fractional Ownership for Digital and Real World Assets
User Stories and On-Chain Requirements

~ Rashmin Chaudhari (GsJYonU5Kz4MJBHZ5UFx9oyStBpXXswnZcFUorktj2yZ)

Part A

Core User Personas

1. **Web3 Investors**
   They are the primary demand drivers, the ones actually buying fractional shares of tokenized assets. Their participation directly validates the key value: shared ownership + automated yield distribution. At the PoC stage, their activity (investing, receiving yield, trading fractions) proves the smart contracts' reliability and UX flow.
2. **NFT Creators / Digital Asset Owners**
   They provide the supply side of the ecosystem, high-value digital assets (NFTs) to fractionalize. Engaging 1–2 trusted creators for the PoC shows how creators can unlock community-backed liquidity and shared ownership without losing full control.
3. **Real-World Asset Owners**
   They help prove real-world application, showing that rent or yield can be tokenized and distributed on-chain. Even one pilot (e.g., small property or co-working space) can make the concept tangible and media-worthy.

Core Function Mapping

1. **Web3 Investors (Primary Demand Side)**

User Story	**Action / Outcome**	**Atomic Function**
Connect Wallet & Verify Identity	User securely connects wallet and verifies address	Wallet integration + on-chain identity verification
Discover & Browse Assets	User sees listed digital or real-world assets, ownership details, expected yield	Fetch assets from on-chain registry and off-chain metadata
Join Ownership Agreement	User contributes capital to become co-owner; ownership percentage stored on-chain	Deposit funds + update ownership registry
View Portfolio & Dashboard	User views ownership records, share %, accumulated yield, transaction history	UI fetches user-specific ownership and yield data

Receive Automated Income Distribution	User receives proportional rent/revenue automatically	Trigger distribute_yield function on-chain
Exit / Transfer Ownership (Future)	User sells or transfers ownership share to another wallet	Planned secondary-market smart contract call

2. **NFT Creators / Digital Asset Owners (Supply Side - Digital Assets)**

User Story	**Action / Outcome**	**Atomic Function**
Connect Wallet & Verify Ownership	Creator proves ownership of NFT / digital asset	On-chain signature verification
Create Shared Ownership Agreement	Deploy on-chain contract defining initial co-ownership	init_ownership smart contract call
Invite or Open Ownership Slots	Allow external investors to join as co-owners	Update ownership slots in smart contract
Receive Creator Revenue	Automatically receive share of periodic income / royalties	On-chain yield distribution function

3. **Real-World Asset Owners / Developers (Supply Side - Physical Assets)**

User Story	**Action / Outcome**	**Atomic Function**
Onboard & Verify Real-World Asset	Submit proof of ownership and verification documents	Off-chain verification + metadata link
Create Shared Ownership Contract	Deploy on-chain ownership registry for property	init_ownership smart contract call
Integrate Rent/Revenue Stream	Route rent/income payments to on-chain escrow	Payment feeder + trigger distribute_yield

Potential On-Chain Requirements

As a Web3 investor
I want to connect my wallet and verify my identity
So that I can securely participate in shared ownership agreements

Potential On-Chain Requirements:

- A function to register and store a wallet address on-chain.
- Signature verification to prove ownership of the wallet.
- On-chain mapping of wallet addresses to co-owner permissions.
- Event logging for wallet connection verification.

As a Web3 investor
I want to discover and browse available digital or real-world assets
So that I can choose assets to invest in

Potential On-Chain Requirements:

- On-chain registry storing all active shared ownership agreements.
- Metadata account for each asset storing ownership percentage, expected yield, and status.
- Query function to fetch all listed assets and their details.
- Event emission when a new asset is listed.

As a Web3 investor
I want to join an ownership agreement by contributing capital
So that I can become a co-owner and earn proportional yield

Potential On-Chain Requirements:

- Deposit function to accept SOL or token transfers from investors.
- Update ownership registry to add new co-owner and recalculate ownership percentages.
- Escrow account to hold deposited funds securely.
- Event logging for investment transactions.
- Validation to prevent joining if ownership slots are full.

As a Web3 investor
I want to receive automated income distribution
So that I can earn proportional yield without manual intervention

Potential On-Chain Requirements:

- Smart contract function (distribute_yield) to calculate each co-owner's share.
- PDA escrow account to store income before distribution.
- On-chain transfer of proportional funds to each co-owner wallet.
- Event emission for each distribution transaction.

As a Web3 investor
I want to exit or transfer my ownership share
So that I can liquidate or sell my position

Potential On-Chain Requirements:

- Function to transfer ownership percentages to a new wallet.
- Validation to ensure only current co-owner can transfer their share.
- Update registry with new ownership percentages.
- Event logging for transfer transactions.

As a digital asset creator
I want to connect my wallet and verify ownership of an NFT
So that I can tokenize and fractionalize it securely

Potential On-Chain Requirements:

- Signature verification to prove ownership of NFT.
- On-chain mapping of creator wallet to asset ID.
- Event logging for verification.

As a digital asset creator
I want to create a shared ownership agreement
So that I can define co-ownership percentages on-chain

Potential On-Chain Requirements:

- init_ownership function to create a new ownership account.
- Store initial co-owners and percentages.
- Lock NFT in PDA escrow.
- Event logging for ownership creation.

As a digital asset creator
I want to invite or open ownership slots for investors
So that my community can become co-owners

Potential On-Chain Requirements:

- Function to add empty ownership slots to registry.
- Validation to prevent exceeding max slots.
- Event logging for slot creation

As a digital asset creator
I want to receive creator revenue automatically
So that I earn my share of yield without manual claim

Potential On-Chain Requirements:

- Smart contract function to distribute yield according to ownership percentages.
- On-chain storage of creator’s share of accumulated revenue.
- Event logging for each revenue distribution.

As a real-world asset owner
I want to onboard and verify my asset
So that it can be safely fractionalized and sold to investors

Potential On-Chain Requirements:

- Metadata account linking off-chain verification documents.
- Function to mark assets as verified.
- Event logging for verification completion.

Part B: PROCESS APPENDIX

Initial User and Function Mapping

I) Manual User Brainstorming
Direct Users: Web3 Investors, NFT Creators, Web3 DAOs
Indirect Users: Tenants paying rent to tokenized real estate asset, Auditors, Appraisers
Administrators: Developers, Compliance Managers, Community Moderators
Stakeholders: Turbin3, Token holders, Investors, Partner Protocols, Real Estate Agencies

II) AI-Assisted User Prioritization

4. **Web3 Investors**
   They are the primary demand drivers, the ones actually buying fractional shares of tokenized assets. Their participation directly validates the key value: shared ownership + automated yield distribution.
   At PoC stage, their activity (investing, receiving yield, trading fractions) proves the smart contracts' reliability and UX flow.
5. **NFT Creators / Digital Asset Owners**
   They provide the supply side of the ecosystem, high-value digital assets (NFTs, IPs) to fractionalize. Engaging 1–2 trusted creators for the PoC shows how creators can unlock community-backed liquidity and shared ownership without losing full control.
6. **Real-World Asset Owners**
   They help prove real-world application, showing that rent or yield can be tokenized and distributed on-chain. Even one pilot (e.g., small property or co-working space) can make the concept tangible and media-worthy.

III) Core Function Mapping

4. **Web3 Investors (Primary Demand Side)**

User Story	Action / Outcome	Atomic Function
Connect Wallet & Verify Identity	User securely connects wallet and verifies address	Wallet integration + on-chain identity verification
Discover & Browse Assets	User sees listed digital or real-world assets, ownership details, expected yield	Fetch assets from on-chain registry and off-chain metadata
Join Ownership Agreement	User contributes capital to become co-owner; ownership percentage stored on-chain	Deposit funds + update ownership registry

View Portfolio & Dashboard	User views ownership records, share %, accumulated yield, transaction history	UI fetches user-specific ownership and yield data
Receive Automated Income Distribution	User receives proportional rent/revenue automatically	Trigger distribute_yield function on-chain
Exit / Transfer Ownership (Future)	User sells or transfers ownership share to another wallet	Planned secondary-market smart contract call

5. **NFT Creators / Digital Asset Owners (Supply Side - Digital Assets)**

User Story	**Action / Outcome**	**Atomic Function**
Connect Wallet & Verify Ownership	Creator proves ownership of NFT / digital asset	On-chain signature verification
Create Shared Ownership Agreement	Deploy on-chain contract defining initial co-ownership	init_ownership smart contract call
Invite or Open Ownership Slots	Allow external investors to join as co-owners	Update ownership slots in smart contract
Receive Creator Revenue	Automatically receive share of periodic income / royalties	On-chain yield distribution function

6. **Real-World Asset Owners / Developers (Supply Side - Physical Assets)**

User Story	**Action / Outcome**	**Atomic Function**
Onboard & Verify Real-World Asset	Submit proof of ownership and verification documents	Off-chain verification + metadata link
Create Shared Ownership Contract	Deploy on-chain ownership registry for property	init_ownership smart contract call
Integrate Rent/Revenue Stream	Route rent/income payments to on-chain escrow	Payment feeder + trigger distribute_yield

IV) Deriving Core POC requirements

Top 2 Critical User Interactions for the PoC

1. An NFT Creator or Asset Owner creates a shared ownership agreement on-chain (defining % ownership among wallets).
2. Web3 Investors join the agreement (by depositing capital) and later receive proportional income (rent/yield) distributed automatically.

List of Key Technical Requirements

1. Smart Contract (On-Chain Logic)
   - Ownership Registry Program to record multiple co-owners and their respective percentages.
   - Function to initialize shared ownership (init_ownership) and lock the asset (NFT or real-world token) in escrow.
   - Investment logic allowing new investors to join by depositing funds and updating ownership shares.
   - Automated yield distribution function (distribute_yield) to send proportional payments to co-owners.
   - PDA (Program Derived Address) escrow to securely hold assets and yield deposits.

2. Backend Infrastructure
   - Event listener to monitor ownership creation, investment, and yield distribution events on Solana.
   - Off-chain indexer or database to track asset metadata, ownership percentages, and yield history.
   - Revenue feeder script or cron job to simulate periodic income inflows for testing yield distribution.

3. Frontend Interface
   - Wallet integration (Phantom/Backpack) for both asset owners and investors.
   - UI for asset owners to create shared ownership agreements and define ownership splits.
   - UI for investors to view available assets, invest, and monitor their ownership and yield returns.
   - Real-time dashboard displaying ownership data, yield received, and transaction history.

Core Function Mapping (Refined)

Web3 Investors (Primary Demand Side)

- **Connect Wallet & Verify Identity:**
  Securely connect Solana wallet (Phantom/Backpack) and verify address for co-ownership registration.
- **Discover & Browse Assets:**
  View listed digital or real-world assets available for shared ownership, along with ownership details and expected yield.

- **Join Ownership Agreement:**
  Contribute capital to become a registered co-owner of an asset; percentage ownership is calculated and stored on-chain.
- **View Portfolio & Dashboard:**
  Access ownership records, share percentage, accumulated yield, and transaction history.
- **Receive Automated Income Distribution:**
  Periodically receive proportional rent/revenue directly to their wallet through on-chain yield distribution.
- **Exit / Transfer Ownership (Future scope):**
  Option to sell or transfer ownership share to another wallet (planned for later phases).

NFT Creators / Digital Asset Owners (Supply Side – Digital Assets)

- **Connect Wallet & Verify Ownership:**
  Authenticate asset ownership via on-chain signature verification.
- **Create Shared Ownership Agreement:**
  Initiate on-chain ownership contract defining initial ownership percentages among wallets.
- **Invite or Open Ownership Slots:**
  Optionally allow external investors to join as co-owners by contributing capital.
- **Receive Creator Revenue:**
  Automatically receive their share of periodic revenue or royalties as part of the yield distribution.

Real-World Asset Owners / Developers (Supply Side – Physical Assets)

- **Onboard & Verify Real-World Asset:**
  Submit proof of asset ownership and verification documents off-chain (linked via metadata).
- **Create Shared Ownership Contract:**
  Deploy on-chain ownership registry representing real-world property and define co-ownership terms.
- **Integrate Rent/Revenue Stream:**
  Route rent or income payments into on-chain escrow, triggering automatic yield distribution to all co-owners.

What’s Strong

- Clear separation between user types and technical flow.
- Direct mapping between core interactions (create ownership, join ownership, receive yield) and corresponding contract functions.
- Demonstrates true *on-chain partnership logic* (not fungible token mechanics).

What’s Missing or Needs Clarification

1. **Access Control Logic:** Define who can modify or end an ownership agreement (creator-only, majority vote, or admin).
2. **Dispute / Exit Flow:** Outline how an investor exits or transfers ownership share (even if deferred to later stage).

3. **Compliance Hooks (for Real Assets):** Placeholder for off-chain verification, escrow agent, or document storage.
4. **Testing Plan:** Add explicit test cases for yield precision and ownership updates to ensure contract correctness.

Part C Refinement Log

Before	After	Rationale
"Connect Wallet & Verify Identity"	Split into wallet connection + on-chain verification	Atomicity; makes each step testable
"Discover & Browse Assets"	De-jargonized to "User sees listed digital or real-world assets, ownership details, expected yield"	Simplified for non-technical understanding
"Receive Automated Income Distribution"	Linked explicitly to distribute_yield	Makes technical expectation explicit
"Create Shared Ownership Agreement"	Clarified "deploy on-chain contract defining initial co-ownership"	Removed ambiguous language
"Onboard & Verify Real-World Asset"	Added "submit proof and link metadata"	Added off-chain verification step for compliance
"Invite / Open Ownership Slots"	Clarified "allow external investors to join as co-owners"	Atomic action; clear outcome
Platform admin tasks	Separated into listings, verification, deployment, yield distribution	Atomicity + clarity for dev plan