

**CONFIDENTIAL**



## Scope of Work

**16 March 2026**

### Tokenized Rental Property Pool – Onchain MVP

#### Project Summary

Build a minimal, production-ready onchain MVP for a tokenized real estate investment platform where investors receive ERC-20 pool tokens representing participation in a rental property pool instead of receiving apartments directly.

The system enables:

- Fundraising and token allocation through a permissioned onchain flow
- Permissioned presale participation for early investors, including discounted private token allocations before public trading
- Pool token issuance (ERC-20 standard) to approved investors
- Onchain investor registry and permission management
- Onchain property pool registry for portfolio representation
- Basic onchain income distribution and investor claiming
- Admin-side management of investors, properties, allocations, and distributions

Single EVM-compatible blockchain, single ERC-20 pool token, and single whitelisted payment token if purchase flow is enabled. KYC or identity checks may be performed by third-party providers, while investor approval status is stored and enforced onchain.

#### User Roles

##### Investors

- Connect wallet
- Complete permissioned investor registration
- Receive approval to participate
- Receive or purchase pool tokens

- View token allocation and portfolio exposure
- Claim available income distributions
- Transfer tokens subject to the chosen MVP transfer logic

## Administrators

- Approve or revoke investors
- Add and update properties in the pool
- Mint and allocate pool tokens
- Manage pool settings
- Fund and publish income distributions
- Monitor core platform activity

## Presale Investors

- Connect wallet
- Complete permissioned investor registration
- Purchase pool tokens under special early-stage pricing terms
- Receive token allocations subject to claim, lock, or vesting conditions depending on the final MVP configuration
- View presale allocation and token status in the frontend
- Claim or unlock tokens according to the configured schedule, if applicable

# 1. Smart Contracts Architecture

- **PoolToken (ERC-20)** acts as the main token representing participation in the rental property pool. Minting and burning are admin-controlled through dedicated access roles. Transfer restrictions may be applied depending on final MVP policy.
- **InvestorRegistry** stores approved investor wallets and basic identity reference hashes or verification references, and acts as the permission layer for participation.
- **PropertyPoolRegistry** stores the list of underlying properties and their basic onchain metadata references, including valuation, status, expected yield, occupancy status, and active or inactive state.
- **AllocationManager** handles token allocation to investors, including direct allocations for approved users and batch allocation workflows where needed.
- **PresaleVestingManager** handles early-stage private token sale flows for approved presale participants before public trading. The contract supports whitelist-based participation, configurable token pricing or discount logic, payment in a single whitelisted token, allocation accounting, and token claim, lock, or vesting logic based on the final implementation.
- **DistributionManager** records and manages investor income distributions. Depending on final implementation, this contract either supports direct stablecoin claim flow or lightweight distribution accounting with investor claims.

- **Treasury**, optionally holds incoming stablecoin payments and serves as the source wallet or contract for distribution funding and treasury-controlled actions.

The above architecture is functional and serves as a basis for the blockchain engineers, rather than imperative smart contracts division, which developers decide based on software development principles.

## Testing and Deployment

- Unit tests and integration tests, fuzz and invariant testing, functional testing according to the best standards in web3 industry
- Testnet deployment for tests and Mainnet development into one EVM/TVM network
- Deployment scripts
- Technical handover and documentation
- Basic Continuous Integration and Continuous Development pipeline

## 2. Frontend and UI/UX

Frontend development and UI/UX design are not included in this scope, because frontend effort cannot be estimated without defined wireframes, UI/UX Figma Design and user flows.

However, for MVP planning purposes, the expected frontend structure would normally include the following pages:

### Public pages

- Landing page, Pool Overview page
- Properties page
- Single Property page

### Investor pages

- Connect / Onboarding page / KYC third party integrations
- Investor Dashboard
  - My Allocations, distributions, income, chart for income(if possible)
  - Transactions / Activity page
  - Pre-sale pages

### Admin pages

- Admin Dashboard
- Investor Management page
- Property Management page
- Token Allocation page

- Distribution Management page
- Settings page

Frontend development we estimate will consist of 8-10 SPA pages for this project normally would be 1.5-3 months and ~\$8-16k, depending on the features to be included into MVP.

### 3. UI/UX Design

- Option A: Web3 UI/UX Designer available on a pay-as-you-go basis with the rate \$40/hour
- Option B: we run a discovery phase (subject to a separate agreement based on the exact scope) where we develop exact tech tasks for the designer and the designer implements them in Figma Design. That would be based on a fixed price proposal offered separately (~\$3.5-6k).

### Milestones and Schedule

- **Weeks 1–4:** Smart contract implementation and internal review, event design, core permissioning and allocation logic, testing,
- **Weeks 5-6:** distribution logic, deployment scripts, documentation, and QA (rigorous testing, scrutinize tests, edge cases).
- **Weeks 6–7:** Testnet deployment, bug fixes, polishing, and contingency buffer

### Commercial Summary

#### Smart Contracts Protocol

**Timeline:** ~6 weeks

**Cost:** \$9,600 fixed

**Payment mode:** (1) 50% pre-pay and 50% upon deployment readiness or (2) weekly upfront payments of \$1,600 each over the course of 6 weeks.

#### Frontend Development (separate scope)

**Timeline:** ~1.5-3 months

**Budget:** ~\$8,000-16,000 (depending on MVP requirements)

#### UI/UX Design (separate scope)

**(1) Time & Material:** \$40/hour or

**(2) Fixed-Price Discovery + Figma phase:** ~\$3,500-6,000 (depending on requirements)

**Payment currency:** USDT/USDC on ETH, BSC, Polygon, TRON

## Signatures

**For the Client (Jozef Staszal / Consolidated Properties):**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

**For the Developer (Crabit.io):**

A handwritten signature in blue ink, consisting of a large, sweeping loop on the left and several sharp, intersecting strokes on the right.

Signature: \_\_\_\_\_

**Dmytro Bogdan**

Chief Technical Officer

Date: 15 Mar 2026