AngelBlock Protocol Overview Documentation

Introduction

AngelBlock is a non-custodial, protocol based fundraising infrastructure that allows to conduct token based raises in a more transparent, decentralized, and democratized manner. The AngelBlock protocol is comprised of various tools to make DeFi and crypto safer, including on-chain vesting, cap table management, and token distribution systems.

The end goal is the creation of a wide, self-sufficient and decentralized network that will allow for safer, more secure, and truly crypto-native forms of fundraising, vesting, and token distribution, setting a unanimous standard that will be used around the world.

AngelBlock enables:

- On-chain, verifiable and trustless raises secured by governance
- Token based fundraising, vesting, and token distribution infrastructure
- Trading and transfer of vested tokens in a secure, compliant manner with 0 negative price impact
- An opensource framework for full transparency and trust

Protocol

The AngelBlock protocol is a decentralized network aiming to fix the current problematic world of token based funding. The protocol runs natively on Ethereum, although since inception the design has always planned for the protocol to be multichain.

Contributor funds are protected, and on-chain, where capital is paid out after startups deliver on milestones displayed to investors during the fundraising process, in order to minimize the risk of raising and not delivering (all governed by the protocol). If a startup does not deliver or tries to scam / rug, voting on protocol does not occur and funds are returned to contributors.

The current version of the protocol functions on a milestone based model, where startups declare upcoming milestones on the raise page and contributors vote based on completion of milestones. A second, time-based model of capital payout will also be implemented aiding the automation of the payout mechanic, while giving contributors adequate time to halt smart contract payouts if issues arise. This mechanism aims at protecing contributors and funds, while making the process decentralized, transparent, and more secure than what is available on the market. The protocol can easily be adjusted to be used by funds raising capital, startups, as well as grant programs, and more.

Although launching v1.0 on Ethereum mainnet, our protocol design has been multichain since inception. Decentralized, on-chain, and transparent fundraising is needed in each major ecosystem in the space, and each specific Layer 1 will benefit from safer and verifiable infrastructure for conducting raises. We are heavily focused on primarily integrating the privacy-enhancing Layer-1, Aleph Zero into our tech stack and see this as one of most important goals.

Market Problems

The current fundraising landscape in crypto is highly fragmented and inefficient. It is also rife with distrust and questionable individuals.

- Lack of funding is still the leading cause of startup failure
- Lack of protection while investing in crypto startups
    - 2.9 billion USD lost to rugpulls in 2021 alone
    - Account for 37% of all crypto scams
    - Lack of regulatory clarity leads to some projects just taking money, walking away, or simply not doing anything after they raise to the detriment of those who invested

Keeping the fundraising process on-chain and governed by protocol allows investors to have more security over their capital. Our post-raise governance mechanic is milestone based, and startups must keep delivering if funds are to be continued to be paid out from the smart contract they're in. If a project attempts to rugpull (no contact from team, no milestones being met) investors can vote, allowing for remaining funds to be reverted back to contributors.

- Lack of transparent vesting
    - Teams dumping "locked-up" tokens
- Smart contracts allow for transparent, on-chain vesting that is verifiable. No more vesting that is based on empty promises, misplaced trust, and done manually by teams on cold wallets with all tokens already generated.
    - Lack of infrastructural standard for raising, vesting, distributing assets
- 20% of tokens from token sales are sent either to the wrong address or in the wrong amount
- Alameda was able to invest in multiple startups using the same capital (think of it as VC based double-spend problem). They invested but told teams they must keep the funds on FTX, allowing them to use the same fictional capital over and over again. This is not possible using AngelBlock, as capital raised is held in smart contracts, and only distributed to the project that actually raised that capital.
- Token based projects cannot succeed (when creating a liquid trading market) without a mix of Venture Capital, Angel Investors, and community investors, yet there is no infrastructure available that caters to all groups, leading to multiple and various friction points.
    - "Launchpads" predominantly cater to retail, are centralized, lack transparency and do not stand the test of time - you will rarely or ever see a reputable VC using a launchpad.

Governance

Governance is comprised of two major subsects:

- **Protocol Governance**
- **Post Raise Governance**

Protocol Governance

Unlike other DeFi protocols where a transferrable ERC20 token is linked to governance power, AngelBlock will rely on a action-generated, non transferrable token called xTHOL. Users generate xTHOL by performing actions on the protocol, ensuring that protocol decisions stay in the hands of the users.

Post Raise Governance

Post Raise Governance refers to the governance mechanic used after the completion of a fundraise on the protocol.

Milestone Mechanics

This document describes the logic of the milestones implementation as a governance system in AngelBlock raises.

Defining milestones

To start the raise at AngelBlock platform startup must define milestones for a project delivery. Each milestone must include:

1. Deliverables
2. Date
3. Budget unlocked
4. Token released

Investor’s eligibility for voting

- User must own the Badge from project’s raise
- User must be verified (KYC approved)
- Voting power of a single user cannot exceed voting power proportional to max. investment ticket within a raise.
    - As a future extension startups will be able to define maximal voting power for a single owner. To learn more about Badges and Voting power check Badges section

Standard voting

All eligible investors can take part in the standard voting when it’s open. They can vote Yes or No with a single action. Additionally vote justification must be described. Each vote has an impact on overall voting proportionally to gained voting power. Each voting has set timeframe of 7 days.

As a future feature we might consider option for startup to define voting length within frames given by AB (5-14).

At the and of voting, based on results:

- If amount of “Yes” votes exceeded “No” votes - Positive result
- If amount of “No” votes exceeded “Yes” votes - Negative result
- In case of a draw - Positive result
- Startup can define quorum in % of available voting power
- If the quorum is not reached - voting has a Negative result

Additional rules:

- Voting cannot be finished before the time passes
- Votes cannot be withdrawn (investor can vote once using all voting power)

Re-planning milestones

Single milestone re-planning

Before the milestone date, the startup can apply for milestone re-planning. Such an action must be approved in investors voting. Replanning can be started only if there is no active voting in the upcoming 1 week (no milestone voting, no repair-plan pending or voting).

Startup, for a single milestone, can:

1. Redefine deliverables
2. Set new date
3. Inherits budget from the milestone
4. Inherits Token released from the milestone

The milestone replanning must be approved by investors following the standard voting rules.

Roadmap re-planning

If there is no active voting in the upcoming 1 week (no milestone voting, no repair-plan pending or voting), the startup can apply for a roadmap re-planning. In such application, the startup can:

1. Redefine deliverables for all upcoming milestones
2. Set new dates for all upcoming milestones
3. Set new budget split for all upcoming milestones (total available amount remains the same)
4. Set new Token share for each upcoming milestone (total amount for upcoming milestones cannot be changed)

Plan above can also assume a different number of milestones.

The roadmap replanning must be approved by investors in a standard voting with an additional condition of at least 50% investors taking part in the voting. Such a voting shall last for one week at least.

Milestone approval / rejection

When a milestone date is reached, then:

1. Startup must update the milestone description - add a delivery report
2. Investors voting starts
3. * Before the 1st milestone startup must provide tokens the latest (if they do not - all milestones gets rejected)

Investors voting for a milestone approval

Milestone approval voting is based on a standard voting rules described above.

Approved milestone

If the milestone was approved, the amount of tokens assigned to the milestone is automatically released

Startup gets the budget defined for a milestone.

Investors can claim tokens assigned to a milestone, proportionally to their investment amount.

There is no deadline to claim tokens.

Budget not claimed is returned to AngelBlock 1 year after the voting. This is a safety mechanism to restore budget if the startup lost their keys.

Rejected milestone - repair plan

If the milestone gets rejected the startup can announce the repair plan within 7 days from the milestone rejection date. The repair plan has:

1. Redefined deliverables
2. New date
3. Inherits budget from the milestone
4. Inherits Token released from the milestone

Investors can approve or reject the repair plan in the standard voting (same rules as with the milestone approval voting). Approved repair plan sets a new date and deliverables for a milestone approval voting.

Rejected repair plan or rejected milestone and no approval plan

Investors can reclaim a budget assigned to the milestone which was rejected without a repair plan or in which repair plan was rejected. If the startup failed to deliver a milestone and trustworthy repair plan - all the invested assets assigned to the failed and remaining milestones will be returned to the investors. Utility tokens generated by the startup and assigned to failed and remaining milestones will be returned to the startup.

Validators in the milestones approval process

Validators, a guardians of the AngelBlock protocol, can audit votes given in any voting, removing the once, which are against platform Terms&Conditions.

Validators will be also able to evaluate startup's performance before the voting starts, assessing milestone delivery.

Validator Mechanics

Election

Requirements to participate in the election are staked $THOL tokens and AngelBlock NFTs. Each month, validators will be re-elected based on user voting through delegation of their staked tokens for the chosen candidate. The Top 50 candidates are chosen as new validators and have the possibility of validating on the platform.

Due Diligence

Validators are responsible for performing due diligence for new startups that are aiming to perform fundraising on AngelBlock. With each milestone completed validators will receive part of the reward.

Slashing

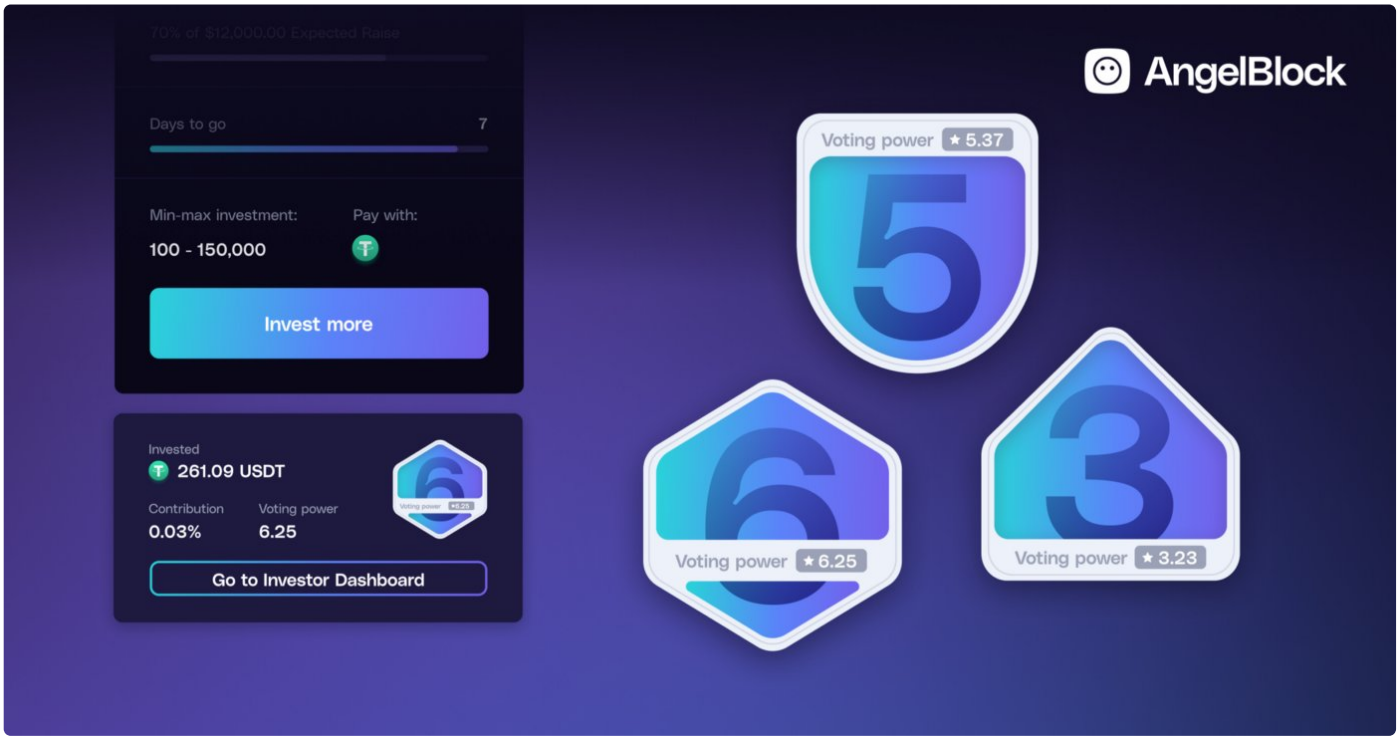
For weak due diligence and allowing poor quality startups, validators are at risk of slashing and losing part of their staked assets (and part of delegation power from users).

Badges

Badges are an ERC-1155 token that are instrumental within the AngelBlock protocol. They not only represent a contributors voting power in a given raise, but also the contributor's position in vested tokens. Vested tokens paid out after the conclusion of a given fundraise are designated by badge location. Badges are NFTs that are automatically generated upon taking part in a given fundraise.

The ERC-1155 token standard allows us to create some truly novel features and mechanics. We are planning to create the first ever vested token marketplace on AngelBlock that allows for new token participants to enter, with zero downside to the project that raised. All outside of an orderbook or active secondary trading market.

Contributors will be able to sell portions of their position in vested tokens (using the fact that ERC-1155 tokens are divisible) and selling them to another market participant. Contributor X is able to sell 10% of his badge to Contributor Y at an agreed upon price and having 10% of the badge (therefore 10% of the position in vested tokens) to be transfered to a new wallet address held by Contributor Y.



Staking Mechanism and Implications

Inclusivity

Anyone will be able to stake $THOL or AngelBlock NFTs to earn staking rewards. AngelBlock does not plan to lock, limit or take fees on staked goods. We believe a healthy ecosystem should incentivize users to enter and have no prerequisites if they want to exit.

Rewards

Stakers will periodically receive auto-compounded $THOL (finite-supply ERC20), which yields an amount that will be dynamically computed based on the current available supply in Staking Rewards Treasury (decreasing APY%, that will raise after each successful raise / $THOL buyback by the team, as well as any unstaking). Additionally, in the future, we plan to add a governance token (xTHOL) as an additional reward for stakers for further influence over the project proposals.

Delegation

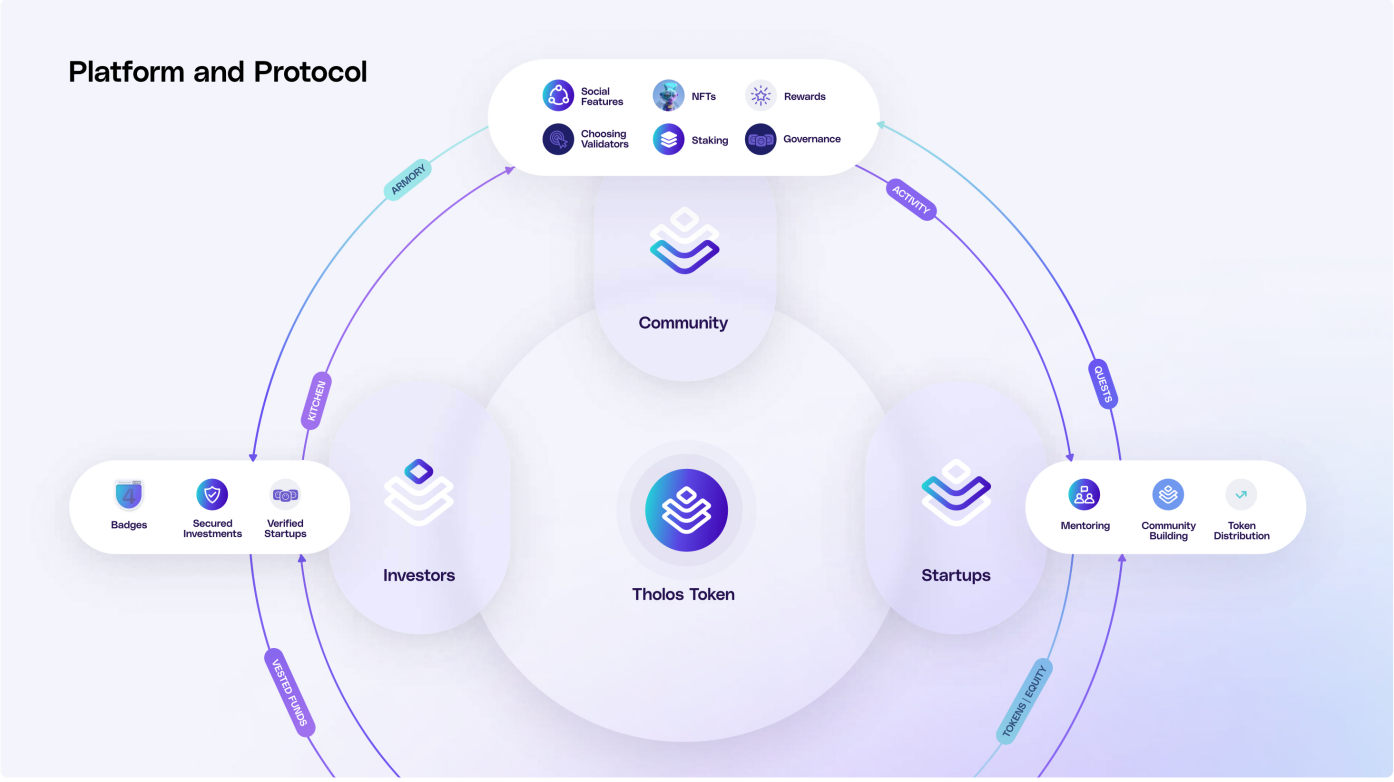
Stakers will be able to delegate the voting power of their staked tokens / NFTs to Validators, who will choose startups for the listing on the platform and perform their due diligence.

THOL Token

The AngelBlock Protocol will heavily rely on use of the native token, THOL. The token itself will be an integral part of the protocol, allowing for users to participate in AngelBlock’s protocol (in form of gas), governance, influencing development, gaining access to additional features and more.

At the time of the Token Generation Event (TGE), the Tholos Token will have the following functionality:

- Token Staking – earning deflationary APY in exchange for nominating Validators of Startups
- THOL will be used to pay gas in the protocol and not requiring ETH on the user's side
    - THOL will be required for all protocol interactions, including but not limited to, taking part in raises, buying and selling badges, claiming tokens.
- Additional privileges based on the amount of tokens held in wallet
  Some additional utility will be implemented after Token Generation Event:
- Priority deals and ticket sizes based on staking tiers
- Community building incentivization
- Rewards for completing task / ticket / contribution by the community members for startups
- Offer rewards for submitting / fixing security issues and loopholes in tokenomics of Startups
- Liquidity sourcing - Creating new liquidity pools with $THOL in pair
- Incentivizing user activity to support listed Startups



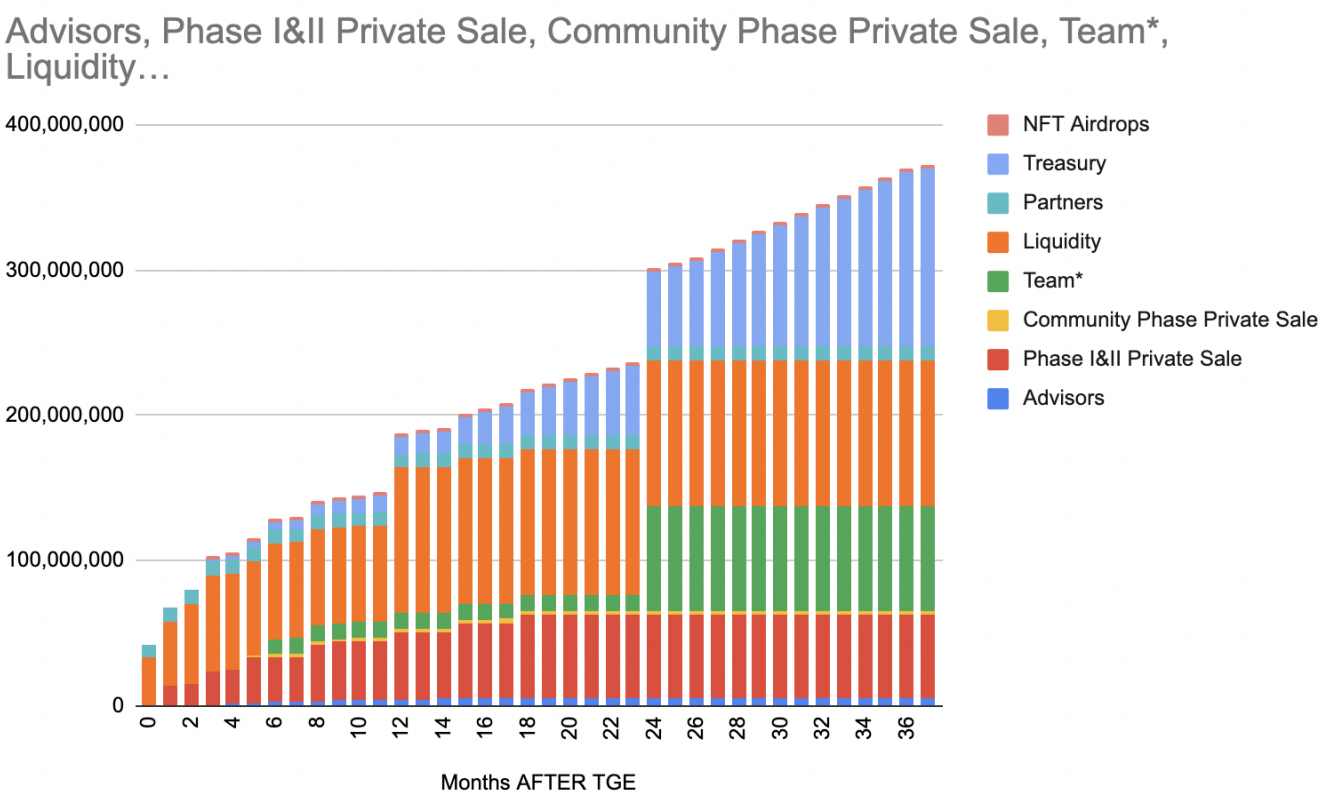
THOL is capped at 400,000,000 with no inflation model. The token supply can decrease over time as a result of the gas mechanic.

Additional Types of Native Tokens

The following below are the types of tokens we plan to facilitate within the AngelBlock ecosystem:

- $THOL (Tholos token) - the main token in terms of monetary value and core ERC20 of AngelBlock. Earned as a reward from staking or community-building tools for most active users on the platform.
- wTHOL (Wrapped Tholos) - correlated incentivization asset aimed to be utilized for cross- user interactions on the platform and essentially allowing the acquiring of $THOL from the AngelBlock Treasury for activity and participation by most active users.
- xTHOL - a future governance token, whose emissions and treasury will be fully controlled and governed by a DAO. This is planned as an additional reward from the staking program that will allow participation in growth and development of AngelBlock and its ecosystem.
- AngelBlock NFTs - a collection of ERC721 tokens that will accompany AngelBlock long term, remain tightly integrated with the platform and strongly evolve over time by consistently revealing “hidden traits” during the growth of the application.
- Badges - ERC1155 certificates unlocked by investors after fundraising, strictly connected with project and amount invested. Will equally conform to voting during startup milestone evaluation and the amount of released project’s ERC20.

Token Unlocks



Fee Structure

The charged fee will be divided in three parts:

- 50% will be AngelBlock’s part of the fee and will constitute its income
- 25% will be used to reward the Validators of the Startups
- 25% will be used to buy THOL from the market to replenish the treasury used for the payments of staking rewards
  AngelBlock’s part of the fee will be programmed on smart contract and it is charged as soon as the fundraise for startup is finished, regardless of vesting milestones etc.
  25% of the collected fee will be paid to Validators as a remuneration for their due diligence process of the startup to ensure the quality of accepted projects. At first the only Validator will be the AngelBlock. In later iterations of the Platform we will allow listing of different validators – like physical persons, first pre approved by AngelBlock. This will provide a smooth transition to a fully decentralized startup due diligence model.
  Validators will receive remuneration as milestones are achieved and vesting periods for investors completed.
  Validators will be motivated to:
- Performing proper due diligence – attempting to input fraudulent data in the process will result in slashing.
- “moderate” the voting and milestones – engage with the community, motivate startups etc.

Private Raises

From the start, our goal was to create a more efficient and secure infrastructure to conduct raises for the crypto economy. This cannot stop on AngelBlock's frontend. AngelBlock's protocol infrastructure will be available to be used by startups, venture capitalists, grant programs, and more in the form of invite based, private raises.

This way we can scale above what is simply available on angelblock.io and have our infrastructure become the industry standard for tokenized raises. This is also the route for increasing our total addressable market as some jurisdictions impose regulatory limitations to public accesibility of raises.

Multiple entities will be able to host private raises for various reasons. We're not stopping at fundraising for startups. On-chain raising with transparent vesting and automated token distribution should be the industry standard and we won't stop at anything less than that.

AngelBlockDAO & NFTs

The AngelBlock NFT are ERC721 tokens that provide their holders with utility features on the AngelBlock platform as well as other benefits which will be periodically assessed in roadmap updates. The collection consists of 6,900 unique tokens and it will be released for minting in two batches in order to optimize the distribution among users and capitalize on marketing potential.

Holders will own the IP to the tokens enabling them to creatively utilize the NFTs outside of the AngelBlock ecosystem. The team strongly believes that NFTs are a key part of the ethos of Web 3.0 and could possibly be used to develop a brand in order to attract users and provide a gateway to ownership of the platform.

Mint price for the first batch is set at 0.069 ETH. The price of the second batch will be determined after considering market conditions and the trade volume of the first batch secondary market.

The NFTs will be used not only as a branding mechanism for AngelBlock but also as an additional revenue stream - every secondary market transaction will generate an additional 4.2 % fee. The funds raised from the AngelBlock NFT mint and secondary sales will be utilized to fund the endeavors in publications of the roadmap.

AngelBlock DAO will be a token-gated organization requiring 1 AngelBlock NFT in order to become a member. The DAO will serve as the main governing body within the AngelBlock ecosystem.

Architecture I

Frontend

AngelBlock is a responsive web application hosted on AWS, that will mainly support authentication by crypto wallets and forward co-signed user requests to smart contracts. Our language of choice for everything on the frontend side is TypeScript - we have a few bits in pure JS, but all application/ test code is written TS. From the libraries/tools point of view we're using:

- NX (https://nx.dev/) to organize code in monorepo and share build tools configuration
- Next.js (https://nextjs.org/) as a meta framework to build apps with React - all apps are exported statically as multi-page apps
- Cypress (https://www.cypress.io/) to write integration and (in the future) e2e tests
- Storybook (https://storybook.js.org/) to showcase components and as a basis for some integration tests and visual regression tests (run by loki - https://loki.js.org/)
- Vanilla extract (https://vanilla-extract.style/) - as a basis for our own design system and as the main tool to handle styling with CSS in various apps
- framer motion (https://www.framer.com/developers/) - as a tool to create all complex animations/transitions
- eslint (https://eslint.org/) - to handle linting
- prettier (https://prettier.io/) - to ensure common code format
- many other libraries that are quite popular in the React ecosystem - this includes, but is not limited to:
  - https://mdxjs.com/
  - https://react-hook-form.com/
  - https://popper.js.org/
  - https://plate.udecode.io/
  - https://github.com/Uniswap/web3-react
  - https://github.com/axios/axios
  - https://docs.ethers.io/v5/
  - https://date-fns.org/
  - and many others

Architecture II

Backend

In the short term, AngelBlock will be a centralized, off-chain back-office hosted on AWS, but in the long term we plan to decentralize the governance and migrate to one of the L2s / L1s (with privacy enhanced features like Aleph Zero) that fit our needs and facilitate our goals for decentralization, without impacting our uptime or user satisfaction). Initially AngelBlock will be responsible for user verification (KYC / AML) and startup validation (end to end process dedicated to illuminating the space with worthy users). Below are the tools and platforms we are using for backend development:

- Docker + docker-compose (https://www.docker.com/ / https://docs.docker.com/ compose/) - organizes the microservices in the local environment and for development purposes
- Python 3.10 + Django + DRF (https://www.python.org/ / https://www.djangoproject.com/ / https://www.django-rest-framework.org/) - framework setup that is the base structure for the entire backend project
- Swagger (https://swagger.io/) - APIs schema description provider
- PostgreSQL 13 (https://www.postgresql.org/) - database engine
- AWS + Github Actions + Jenkins + ArgoCD + Kubernetes - is our CI and CD automation of our remote environments application setup
- Pytest + FactoryBoy + MyPy + Black + Isort + Flake8 + Coverage - to perform tests and quality assurance of our code base
- Sentry (https://sentry.io/) - error tracking and performance insights
- Web3 + eth-account (https://github.com/ethereum/web3.py / https://github.com/ethereum/eth-account) - blockchain integrations
- Pydantic (https://pydantic-docs.helpmanual.io/) - DTOs and schema validation helpers
- Simple JWT (https://django-rest-framework-simplejwt.readthedocs.io/en/latest/) - tools for authentication

Architecture III

Smart Contracts

We plan to support EVM and Aleph Zero environments. For our MVP our target is Ethereum, which codebase will focus on:

• modular / extendable design
• permissioned and controlled upgradability
• incentivisation framework for platform’s actors • security and risk management of funds
• administration driven by DAO

Tech stack:

- Docker + docker-compose (hardhat node)
- TheGraph (hardhat-graph plugin)
- IPFS (TheGraph backend and NFT storage)
- Postgres (TheGraph dependency)
- Node.js + Typescript + Typechain
- Hardhat framework with following plugins:
  - Waffle (tests)
  - Dodoc (documentation)
  - Etherscan (verification)
  - ABI exporter
  - Contract sizer
  - Gas reporter
  - Solidity coverage
  - Graph plugin (subgraph generation)
  - Smock (mocking suite)
  - Alchemy provider
- Ethers.js
- OpenZeppelin contracts
- Mustache / Mocha / Chai
- Foundry (fuzzy testing suite)