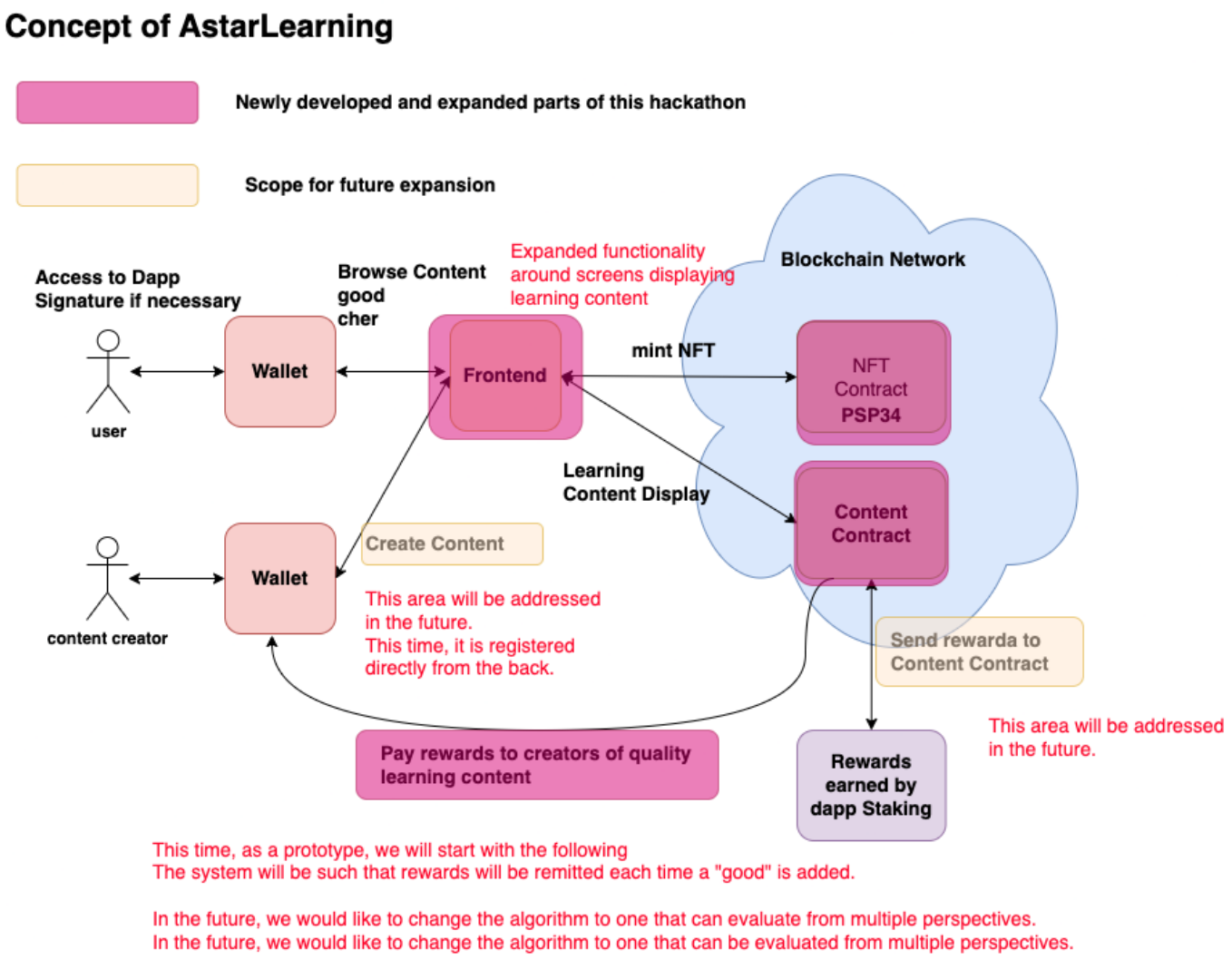
Summary of Astar Learning

proudly built with ink! GITHUB TYPESCRIPT REACT VERCEL POLKADOT
VISUAL STUDIO CODE ESLINT

Concept



Problem to be solved & Solutions

- Problem to be solved

  Our product seeks to solve the following three challenges in expanding the Astar Network ecosystem.

  - The number of WASM-compliant DApps has not increased significantly.
  - Not much is known about WASM in the first place.
  - Lack of educational content that simultaneously provides knowledge and hands-on experience of WASM-compliant Dapp

- Solutions

  With our product, we would like to provide a place where engineers and non-engineers alike can learn about Astar Network and WASM.

  Incentivize Astar Learning participants by having a system that pays rewards to those who provide quality learning content.

  Currently, it is a direct throwaway from content viewers (learners), but in the future, we will use Astar earned from dapp staking as a source of rewards and combine it well with the evaluation algorithm to pay rewards to content providers.

  In this way, we would like to solve the above issues by making it a semi-permanent learning product with no administrator.

Overview of developed features

No.	Name	Overview
1	Content heading information acquisition function	Functions to be implemented for displaying learning content heading information
2	Contents body information acquisition function	Functions to be implemented to display the main body of the learning content
3	Add "good" function	Ability for users to pay content creators a voluntary reward. Direct Transfer. (The destination address is registered in advance on the content contract side.)
4	Transition function to quiz screen	Function for transitioning from the contents screen to the quiz screen
5	Quiz information acquisition function	Function for displaying the four choices of questions to be displayed on the quiz screen
6	Quiz answer function	Implement a function to select an answer from the choices and validate it, displaying a mint button if the answer is correct and an error message if it is not.
7	NFT mint function	Function to issue NFT.

Developed SmartContract

No.	Contract Name	Contract Address	Network
1	WASM NFT	ZjCB8QVKytLmGRGXVCHCuUMnMiQTWU2V3696zqnQiD9kjMg	Shibuya
2	Astar NFT	a6WkJRaZcy6cVkvRQmRmd1TVhAc1Dfq3d7cmpGUjjn9736d	Shibuya
3	Shiden NFT	bezTq8fCqsG6X49e1KRmeVcatNWNaSHfNM218iHzFKPTxsk	Shibuya
4	Content	av5MGBmkEMfKVfQerD1yjDBcQ1hkgu9GoaMN7DkeAMq4nyP	Shibuya

TechStack

No.	Name	Purpose
1	!ink	Smart Contract Development Language
2	Next.js	Front-end development framework
3	swanky	Smart Contract Development Framework
4	TypeScript	Front-end Development Language
5	NFTStorage	Metadata storage storage for NFT
6	Tailwind CSS	CSS framework for front-end
7	Contract-UI	For deploying and validating smart contracts
8	MonoRepo	Improved development efficiency

Challenges we ran into

In addition to developing a WASM contract using `!ink`, we also developed a front-end program using Next.js+polkadot.js and connected them to develop a dapp that is comparable to using an EVM-compatible contract.

Also, for the first time, the team developed a dapp that supports WASM contracts. It was very difficult to set up the development environment for smart contracts.

What we learned

We learned how to develop WASM contracts using `!ink`, `openbrush` and `contracts-ui`.
In addition, there was much to learn about connecting with the front end, which was an issue, and we learned implementation patterns for connecting with WASM contracts using Next.js and Polkadot.js.

We had a hard time setting up the development environment for WASM contract development (Swanky version, rustup nightly settings, etc.), but we managed to set up a system that allowed us to develop as a team and develop MVP.

We would like to make use of this development experience as a foundation for future WASM development.

Milestones

As for the future of Astar Learning, we plan to upgrade its functionality by taking advantage of opportunities such as hackathons, and we hope to contribute to the expansion of the ecosystem by successfully incorporating dapp staking.

Of course, we have no problem adding content about technologies other than those related to Astar Network and Polkadot, and we would like to make the application an opportunity to share knowledge not only about Web3 but also about AI, quantum computers, and other advanced technologies.

1. Front-end enhancements 【2023 Q3】
    - Add functionality to allow content creation from the front end.
    - Add functionality to allow NFTs to be created from the front end.
2. Smart contract enhancements 【2023 Q3】
    - Add a method to pay rewards from content contracts.
3. Apply for dapp staking 【2023 Q4】
    - Apply to dapp staking.
    - If adopted for dapp staking, the reward should be credited to the content contract and the reward paid from there.
4. increased user 【2024 Q1〜】
    - To gradually increase the number of Astar Learning users.
    - In Japan, there is also a community for WASM contract developers led by official ambassadors, so we will gradually increase the number of users using the service from that starting point.
    - The ultimate goal is to have Dapp revolve completely around user-layer-only activities.

Live Demo

Astar Learning's Live Demo is here (Vercel)