HamsTON - A TON toolkit platform to onboard thousands of developers into the TON ecosystem

Funding Ask 25,000 USD Milestones 5

Details

### **The previous application was rejected because of the unclear statement of our proposal. Therefore, we would like to re-submit the up-to-date version with more details for the community and committee to evaluate.**

### **HamsTON is a TON toolkit platform to onboard thousands of developers into the TON ecosystem.**

Deck:

[**https://docs.google.com/presentation/d/1Coipp9GyqVNXAHuqx4lmkJO6G3ZP-iLj/edit?usp=sharing&ouid=117313761802924893810&rtpof=true&sd=true**](https://docs.google.com/presentation/d/1Coipp9GyqVNXAHuqx4lmkJO6G3ZP-iLj/edit?usp=sharing&ouid=117313761802924893810&rtpof=true&sd=true)

Not only does HamsTON help developers improve the efficiency of the development process, but it also helps them get started with the Func language quickly. With our solution, developers can lower the barrier to entry into the TON ecosystem at an early stage and get up and running quickly. In addition, our solution will also support the rapid development capability of DApps for Telegram Bots in the future, providing developers with a more convenient development environment and tools. This will accelerate the learning curve for developers and enhance their ability to build applications and integrate Telegram Bots on the TON platform. By providing a streamlined workflow and development tools, our CI/CD solution will help developers develop and deploy their applications more efficiently, driving growth and innovation in the TON ecosystem.

HamsTON is focused on providing TON with the following features:

**1. Quickly create FunC contract projects:** We provide convenient tools to help developers quickly start FunC contract projects. By providing templates and sample code, developers can quickly start writing their own contracts.

**2. Code checking and building:** We integrate code checking and building features to help developers quickly check the quality of contract code and potential problems, and generate executable versions. This helps ensure the stability and reliability of the contract.

**3. Rapid contract deployment and interface interaction:** We simplify the contract deployment process and provide a user-friendly interface that allows developers to interact with contracts directly. Through simple operations, developers can deploy contracts to the TON network and use our tool interface to invoke and test contract functionality.

**4. Front-end static code hosting:** With our tools, developers can easily upload their front-end static code to **TON's storage network**. This process is very simple and efficient. Our tools are integrated with TON storage, providing developers with a direct channel to securely store their front-end code on the distributed TON storage network.

**5. Ton’s integration with Telegram:** Our tool will provide a user-friendly interface and tools that allow developers to easily make modular combinations. Developers can quickly build their own Telegram bots with simple operations and configuration, and use TON contracts as the backend to support various functions. This quick build will accelerate the development cycle and improve development efficiency, enabling developers to better meet user needs and create innovative applications.

**6, Onboarding more Web2.0 developers into the TON ecosystem:** As one of the core mods, HamsTON has successfully integrated over 420,000 Web 2.0 developer resources and details. None of these developers are currently building in the Ton ecosystem. With our onboarding module, these developers can be brought into the TON ecosystem.

With these features, our developer tools provide an efficient and convenient development environment that supports developers to quickly create, test and deploy TON contracts and enable modular combinations with Telegram bots and the interaction of Ton contract data with other chain networks. This will drive the development of the TON ecosystem and provide developers with more opportunities to innovate and scale.

**Team:**

* Porsche (CoFounder and CTO):
* Ex CTO of a listed Fintech company, leading 6000+ IT engineers and supporting 1000+ large projects.
* https://www.linkedin.com/in/porsche-shi-41832446/
* Nova (CoFounder and CMO):
* Founder partner of industry funds.
* Graduated from Shanghai Jiaotong University.
* https://www.linkedin.com/in/nova-wong-92b48359/
* Alvis (CoFounder and COO):
* Previous Co-Founder and core member of 2x Web3 projects.
* PhD in Economics and Management & Master in IT from top European Universities.
* https://www.linkedin.com/in/alvis-tsui-69952341/
* Liam (VP and Head of Tech):
* 4+ years’ experience in crypto.
* 8+ years software coding experience in java/Go/Rust/FunC, DevOps and Cloud Computing.
* <https://github.com/ltmuyuan>
* https://www.linkedin.com/in/liam-liang-092950245/
* Tom (Backend Developer):
* over 10 years of software development experience being familiar with Go/Rust/Java/Python/JavaScript/Typescript/FunC, cloud computing, cloud native, and DevOps.
* <https://github.com/mohaijiang>
* 8 more Devs and 2 more marketing

**Telegram Handle**

https://t.me/Alvis\_Tsui

**Reference point of contact**

Nan Wang and John Z

**Overview of the technology stack to be used**

The backend uses golang, mysql, redis, FunC and the frontend uses Typescript, vue and ton libraries (ton, ton-crypto, ton-core, ton-connect)

**How does your project implement the TON blockchain and which parts will be on-chain?**

Our project can deploy ton smart contracts for users. We can create or import ton smart contracts for users, and compile ton smart contracts for users in the background. Finally, users can deploy their own smart contracts to the test network or main network through wallets (such as Tonkeeper).

**GitHub repository of the team/project**

https://github.com/HamsTON-Space

**Data models/API specifications of the core functionality?**

Our core function is to build pipelines and deploy contracts online. Including Ton contract compilation, cell file collection, ton contract deployment

**What are success metrics for your project and who are the target users?**

To grow together with the TON EcoSystem. Our target users are all development teams or developers in other ecosystems that would like to build in the TON ecosystem, and also the 420000 web2 developers who are already on-boarded in the HamsTON developer database.

Milestones:

Milestone 1:

a.Design the basic architecture of the application

b.Complete the workflow diagram of the user process

c.Build the contract template function

d.Complete FunC project creation function

5000 USD

Milestone 2:

a.Investigate mainstream code checking usage and workflow

b.Complete the code checking function, so that developers can quickly complete the code checking of FunC contracts through the interface

5000 USD

Milestone 3:

a.Complete FunC interface contract deployment and contract interaction features

b. Completion of static code hosting functionality that allows developers to securely store their front-end code on a distributed TON storage network

5000 USD

Milestone 4:

a. Complete front-end code addition, allowing users to add their own front-end code to the platform

b.Complete fast front-end code and contracts for access: Integration of Ton with Telegram by using Telegram Bot API and related APIs of Ton

5000 USD

Milestone 5:

a. Ongoing Telegram bot integration functionality, where developers can use tools to automate the building and deployment of contracts, associate front-end code with contracts, and embed front-end interfaces into Telegram bots.

b,integrate Web2.0 developers into Ton Ecosystem

5000 USD