

Treasury Proposal: dApp Marvels - Ink! ecosystem growth tool

Proponent: 121k35TZKEpoQeKURnEgt2zqWsyDKxUJkTFuwpZeLoSYUe7o

Date: 20.11.2023

Requested DOT: 25 361,29 DOT

Short description: Tokenguard proposes to build dApp Marvels - an on-chain analytical tool intended for Ink! dApps creators to support their growth and operation through in-depth analysis and insights of on-chain user behavior, which will help to more effectively attract users to the Polkadot ecosystem and support Ink! creators in business and infrastructure development.

Project Category/Type: Infrastructure ▾

1. Context of the proposal

Ink! ecosystem is quickly growing thanks to a huge number of marvelous dApp builders that choose Ink! over Solidity because of its security, transparency and simplicity of use. Each new dApp brings in new users into the ecosystem, creating demand for the underlying native L1 coin which eventually translates into the ecosystem growth.

To maintain this trend and increase the effectiveness of Ink! dApps builders' activities that attract new users, tools supporting growth and operations are needed. This issue was also addressed here <https://forum.polkadot.network/t/we-need-to-make-ink-showcase-projects/592>.

dApp builders currently lack an out-of-the-box solution that would allow them to discover dApp Marvels. What are these? Marvels are data-driven insights concerning the growth of each respective dApp. Whether it's a discovery of power users segment, correlation between two important metrics or understanding which marketing campaign works best - we call them Marvels as they're precious to each dApp owner 💡

Tokenguard is an on-chain analytics tool that already supports the Polkadot ecosystem by providing analytics for several parachains such as Astar, Aleph Zero, Moonbeam and many others through user-friendly no-code/no-SQL dashboards and Ecosystem insights - [Tokenguard app](#). This solution aims to expand the capabilities of the presented metrics with more detailed and valuable insights for dApps built in Ink!. Moreover, the results of this grant constitute a stand-alone tool that can be reused by other teams and projects.

2. Problem statement

Most of the Ink! teams that we spoke with **are wasting hundreds of hours and thousands of \$** to build their own data displaying solutions and maintain them which effectively prevents them from building and developing user-centric features. Most of these solutions are hard to maintain and manage, increasing the technological debt of each project.

In order to facilitate growth, Substrate parachains need a vast and thriving environment of developers creating dApps and smart contracts. Each smart contract generates thousands of transactions that store important technical and business insights. Each new growing dApp, with its own marketing & growth strategy, **brings in new active users to the ecosystem**. These insights are currently hard to discover and the ecosystem needs infrastructure tools that would make it easy for developers and management teams to uncover them, similarly as it is done in EVM environment with tools like Tenderly or Dune Analytics.

Benefits of solving the problem:

1. Allowing Ink! dApp builders to focus on **developing their products**
2. **Time and money savings** on data tracking and visualisation [Building in-house data-analytics = ~\$30k - \$70k]
3. Higher security of dApps as the tool can be used for tracking **security metrics**
4. **dApp transparency** to ecosystem managers and Substrate community
5. Smaller maintenance cost and technological debt
6. **Faster growth of Ink! dApps**

3. Proposal objective(s) or solution(s)

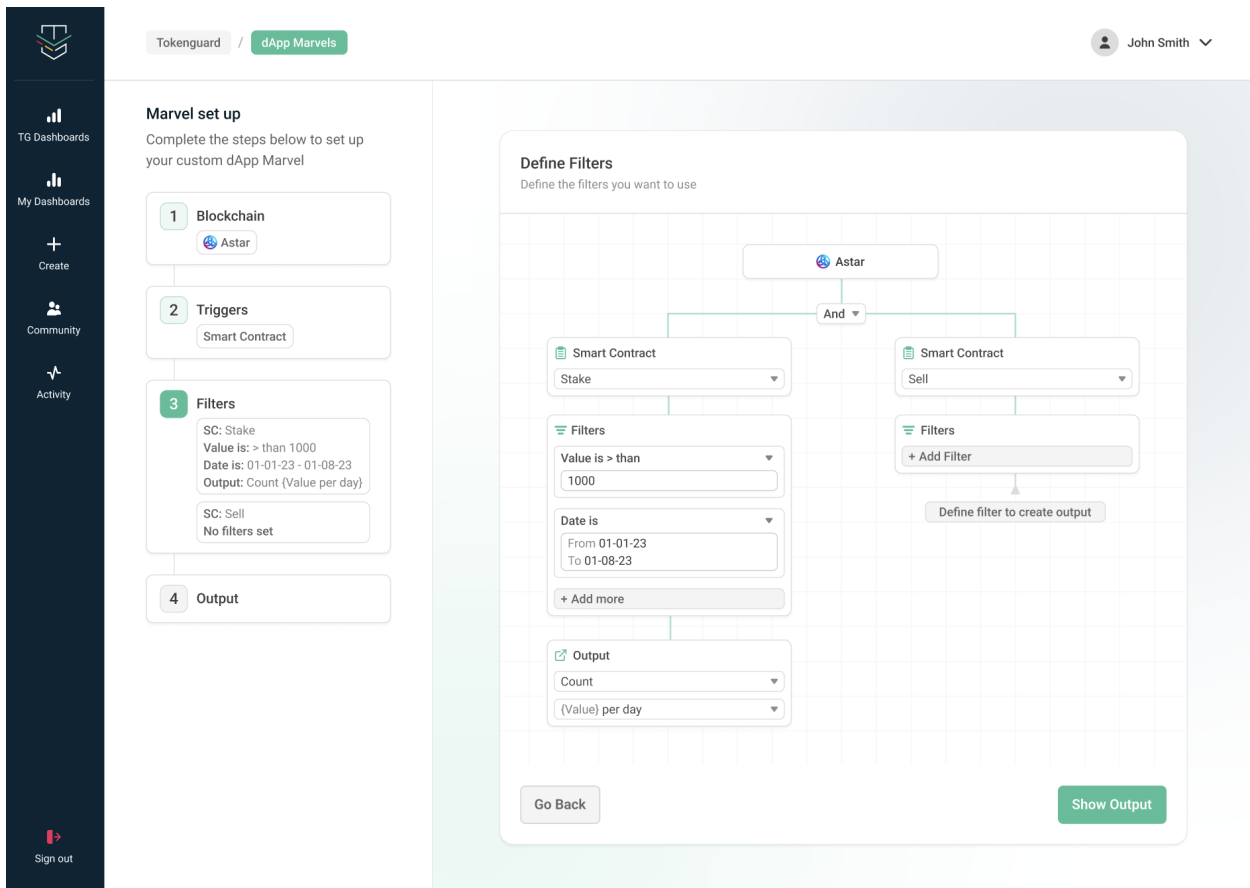
We propose creation of a standalone tool that will allow dApp creators & analytical tools index data coming from their smart contracts, aggregate them in a predefined human-readable format and serve them in plot-ready format directly to your frontend. Such tool is going to run a number of steps to make the Marvel 💎 discovery easy and fun for dApp creators.

1. After smart contract deployment, dApp creator runs the Ink! dApp Marvels analytics by uploading contracts' ABIs.

2. The ABIs are decoded and translated into a SubSquid powered data processor with smart contract decoder.
3. Using the API interface dApp creator specifies potential dApp Marvels that should be tracked - this can be an event with its respective arguments (or any mixture of both). Marvels can include actions invoked in more than 1 smart contract.
4. Aggregated database of Marvels is created and automatically populated in case any of these events happen on-chain.
5. dApp creator can query the database using API or display the data in a human-readable format using JSONs that can be then plotted on the frontend of your app.

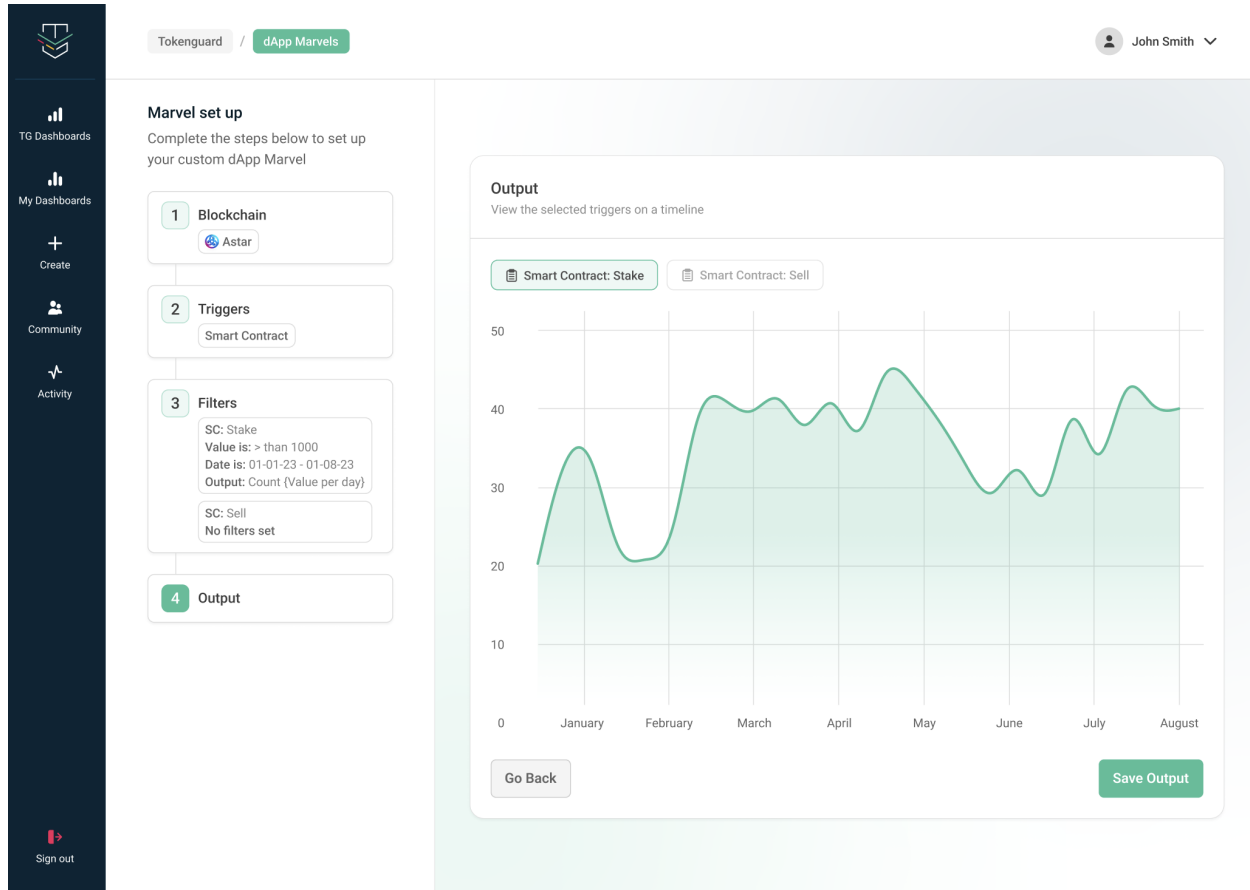
The solution can be used as a standalone tool or parsed by data analytics tools available in the space such as Grafana, Tokenguard, Databox and others. Thanks to the API, dApp creators will be able to display their dApp data actively on websites to increase transparency for users.

Visualisation of the dApp Marvel set up @ Tokenguard interface



The screenshot displays the Tokenguard dApp Marvels setup interface. On the left, a sidebar contains navigation links: TG Dashboards, My Dashboards, Create, Community, and Activity. The main content area is titled "Marvel set up" and includes a sub-header "Complete the steps below to set up your custom dApp Marvel". The setup process is divided into four steps: 1. Blockchain (Astar), 2. Triggers (Smart Contract), 3. Filters, and 4. Output. Step 3, "Filters", is currently active and shows a configuration for the "Stake" event. The filters are defined as "Value is > than 1000" and "Date is From 01-01-23 To 01-08-23". The output is set to "Count (Value) per day". The interface also includes a "Define Filters" section with a visual flowchart showing the logic: Astar -> And -> Smart Contract (Stake) -> Filters (Value is > than 1000, Date is From 01-01-23 To 01-08-23) -> Output (Count (Value) per day). A "Go Back" button is located at the bottom left, and a "Show Output" button is at the bottom right.

Visualisation of the dApp Marvel set up @ Tokenguard interface



Milestones, Deliverables & Reporting

The delivery of this proposal is an infrastructure that allows anyone to create a standalone analytics tool for "Ink! smart contracts", while also being a ready solution to be integrated as an additional feature in existing analytics tools. The entire delivery will be available in open source repositories on [Github](https://github.com)

Milestones	Tasks	Task Description	Deliverables & Reporting
Ink! Smart Contract Decoder	Subsquid processor	- Using the SubSquid ecosystem we will prepare contract-agnostic squid which indexes normalized data about contracts from provided ABI. - We will prepare GraphQL access interface to	Delivery: Public GitHub repository.

		indexed data for builders usage.	Reporting: The updates will be posted on the corresponding proposal post.
dApp Marvels storage	Database	1. Developing a service of automated SQL database creation based on dApp Marvels defined by user 2. Connecting the database to SubSquid smart contract decoder: 1. collecting raw data separately as smart contract events with their respective arguments 2. aggregating the data based on defined dApp marvels 3. Implementing automated schema changes to the database based on real-time user changes and newly defined Marvels	Delivery: Public GitHub repository. Reporting: The updates will be posted on the corresponding proposal post.
	Data processing and handling	1. Prepare a data pipeline that allows accessing the data with low delay 1. automated data aggregation pipeline 2. processing events coming from numerous smart contracts based on user's dApp Marvel definition	Delivery: Public GitHub repository. Reporting: The updates will be posted on the corresponding proposal post.
dApp Marvels definition	Translating smart contract events into dApp Marvels	1. Create a database that will collect ABIs of dApp's smart contracts and automatically process them using Ink! SubSquid decoder 2. Create a template to connect events from multiple smart contracts and set up required arguments & filters using logic rules 3. Design universal data model for transforming smart contract calls and methods with their respective arguments into previously user-specified dApp Marvels that need to be tracked & discovered 1. defining the database altering schema based on user changes of dApp Marvels	Delivery: Public GitHub repository. Reporting: The updates will be posted on the corresponding proposal post.
REST API	Creation of	1. Creating a REST API service that will allow users	Delivery: Public

interface & Marvel display	interactive REST API interface	to: 1. set up new trackable marvels 2. collect information about existing marvels 3. modify existing marvels	GitHub repository. Reporting: The updates will be posted on the corresponding proposal post.
	API <> Database connection	1. Connecting the API to the database for: 1. automated schema altering 2. events notifications 3. data processing 2. Adding a JSON outputting service to the database to allow displaying the data in a format readable to analytics tools such as Grafana, Tokenguard and others.	Delivery: Public GitHub repository. Reporting: The updates will be posted on the corresponding proposal post.

4. Timelines, Budgets & Team

This proposal is planned to create an Ink! dApp Marvels analytical tool by delivering the above-mentioned Milestones within 4 months of acceptance and receipt of funds.

To achieve this, we will engage a team consisting of: data analyst, front-end developer, back-end developer, blockchain developer, project manager, QA, graphic designer, where the average hourly rate is \$80/h. The time commitment of PM, QA and graphic designer is approximately 1/4 - 1/2 FTE depending on the demand, while devs and data analysts are FTE.

The specific development costs and time breakdown for each milestone are as follows:

Milestones	Tasks	Duration	Hours effort	Costs
Ink! Smart Contract Decoder	Subsquid processor	1.5 months	250h	\$20,000
dApp Marvels storage	Database	1 month	155h	\$12,400

	Data processing and handling		75h	\$6000
dApp Marvels definition	Translating smart contract events into dApp Marvels	2 months	635h	\$50,800
REST API interface & Marvel display	Creation of interactive REST API interface	1 month	125h	\$10,000
	API <> Database connection		317,5h	\$25,400
TOTAL		4 months	1557,5h	\$124,600

Team

Tokenguard's clients include Swiss Sygnum Bank, [Bitcoin.com](https://www.bitcoin.com/), Astar, Aleph Zero and many others recognizable brands. Each of the cofounders has tremendous web3 experience - Damian was co-creating DeFi analytics platform Kasuria. Jacob was working at OpenLoyalty, helping web3 projects engage their users and Kamil created web3 security analytics solutions, conducting more than 30+ audits.

Team members:

Kamil Gorski - CEO (<https://www.linkedin.com/in/gorskikamil/>)

Jakub Rojek - CTO / Backend developer (<https://www.linkedin.com/in/jakub-rojek-dev/>)

Damian Rolek - CDO / Data Analyst (<https://www.linkedin.com/in/damianrolek/>)

Matt Saczewski - Business Development Manager (<https://www.linkedin.com/in/msaczewski/>)

Jacob Piorecki - UI/UX Designer (<https://www.linkedin.com/in/jacobpiorecki/>)

Piotr Józefowicz - Backend Developer (<https://www.linkedin.com/in/piotr-jozefowicz-529669220/>)

Adam Król - Backend Developer (<https://www.linkedin.com/in/adam-kr%C3%B3l-7b3a831b6/>)

Krystian Trepczyński - Frontend Developer (<https://www.linkedin.com/in/krystiantrepczynski/>)

Adam Denes - Data Scientist

5. Payment conditions

The funding request for a continuous 4-month development grant is \$124,600 USD in DOT, calculated based on the EMA30 rate provided by Subscan at the time of submission. We are seeking this allocation as a one-time upfront payment, intended to cover infrastructure development and maintenance expenses, as well as the remuneration costs for a team of developers working on the implementation.

Beneficiary Information

Registered Legal Entity: Block Solutions Sp z o.o. (Tokenguard)

Registered Address: Foksal 3/5 St., 00-366 Warsaw, Poland, EU VAT ID PL7252284130

Fund recipient address: 167fMsTnjc28SUmEYXHCMV8pGUJhUWXtmBPfC66etfiHoHAK

Fund manager contact: Kamil Górski, CEO @ Tokenguard contact@tokenguard.io

6. Comments, Qs&As

Please see the discussion <https://polkadot.polkassembly.io/post/1983> on Polkasassembly for this proposal. We have engaged each of the commenters who have requested further discussion.

Additionally for any project related topics Matt is available on [Polkadot Direction](#):

@mattsac:matrix.org or via email: matt@tokanguard.io

7. We'd love to hear about how you got to know about the Polkadot on-chain treasury.

The treasury is a well-known source of funding. As a team providing on-chain analytics for the Polkadot ecosystem, we are in contact with several parachains, dApps, as well as the W3F team and the Parity team. Delivery of this grant specifically supports some of the most popular parachains with Smart Contract functionality - Astar, Aleph Zero and Phala Network and helps to grow & develop the WASM ecosystem.