Case study

Technical Report

Draft version 1.0 – May 13th, 2023

Web3 Foundation Grants Program

Contributors

Rafael Brandão (rafael@mobr.ai)
Marcio Moreno (mmoreno@mobr.ai)

1. Introduction

The objectives of this case study are twofold. First, to provide insights for developers and software architects with a roadmap for the design and building of query engine tools, considering use cases for retrieving and analyzing blockchain data from the Polkadot network and its parachains. These use cases will encompass queries supported by a controlled natural language (CNL) and the POnto ontology. Second, we look to assess the perceived value and effectiveness of said query service by participants of the Web3 community, as well as to get insights for building a CNL to support blockchain query services, and for validating the proposed draft ontology.

2. Polkadot Analytics Query Service

Figure 1 presents the proposed architecture for the Polkadot Analytics query service. The architecture encompasses a controlled natural language engine (CNL Engine), which is in charge of parsing input queries and also assisting in query completion itself, supported by the POnto ontology. The role of CNL Engine is to convert queries to a structured format, using the Query Engine component features to process the input. The Informative Artifacts Engine component manages the creation of multimodal artifacts, for visualizing query results. It provides mechanisms to support composition of such artifacts in dashboards. In order to achieve this, the Informative Artifacts Manager is supported by the Multimodal Content Engine component. Users may access the service either by browsing to its webpage frontend or programmatically through an API offering.

In the data layer, a Knowledge Base serves as a central repository of structured and formal knowledge for other componentes, with definitions, domain specific knowledge supported by the POnto ontology, wording disambiguation, and mapping for blockchain functionalities and external sources. A Database is used for caching, aggregating and structuring information when necessary. The DLT Wrapper component is in charge of supporting the Query Engine component to fetch all the necessary data from the Polkadot Relay Chain, parachains, and off chain sources (on the base of the architecture).

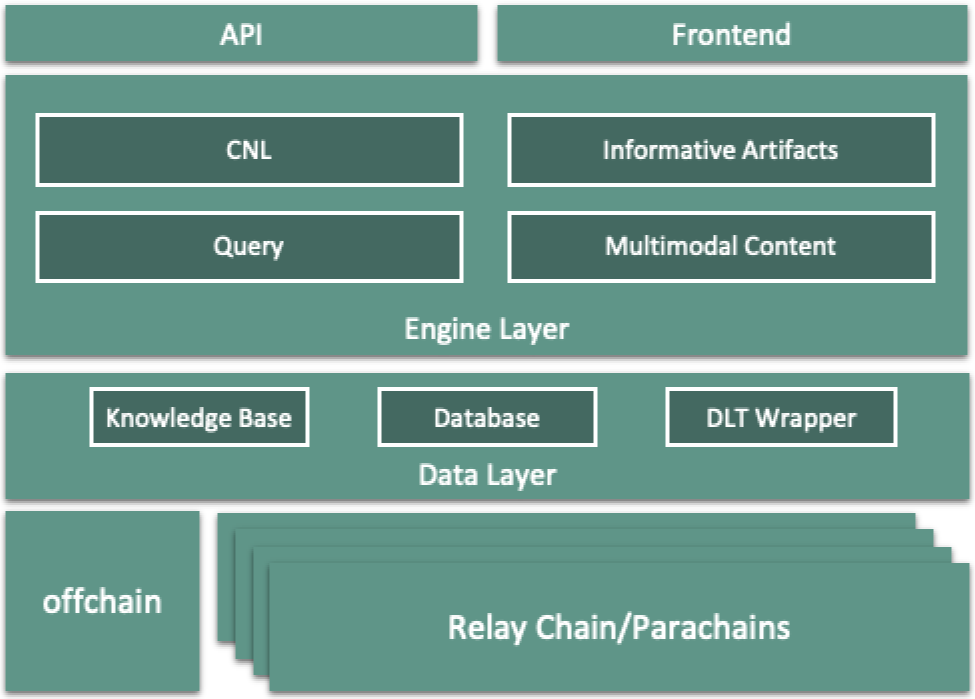

Fig. 1. Proposed architecture for the Polkadot Analytics query service.

A processing strategy for user queries in the Polkadot Analytics service is illustrated in Figure 2. The process begins with a user submitting a query in a controlled natural language format to the service. For example, the user may enter a query like "What are the parachains running in Polkadot?". The service utilizes NLP techniques to parse the query and extract key entities, relationships, and actions mentioned. This step involves understanding the user's intent and mapping it to the corresponding concepts in the POnto ontology. The parsed query is mapped to relevant concepts and relationships within the ontology. This mapping helps establish a semantic understanding of the user query and allows for effective retrieval and integration of data from different data sources both external and within the Polkadot ecosystem. The service interacts with other APIs and retrieves the relevant data and cross-information from different parachains based on the user query and the ontology mapping. The retrieved data is processed and analyzed to generate a summary of the query results. The Polkadot Analytics web service may perform computations, aggregations, or filtering based on the user's query requirements and the nature of the data. The results are structured and enriched with multimodal content, such as charts, graphs, or visualizations, to enhance the user's understanding and facilitate effective data visualization. The summarized query results, along with the multimodal content, are presented to the user through an interactive user interface. The service provides a visually appealing and user-friendly interface for users to explore and interact with the query results. Users can visualize the data, refine their queries, and further explore the insights provided. Finally, the user has the opportunity to provide feedback or refine his queries based on the presented results. The Polkadot Analytics web service may incorporate user feedback to the knowledge base, enhancing future query processing and improve the system's performance and relevance.

By following this step-by-step process, the Polkadot Analytics web service leverages controlled natural language, ontology mapping, and data retrieval from the Polkadot ecosystem to process user queries, generate meaningful summaries with multimodal content, and facilitate intuitive visualization of the query results.

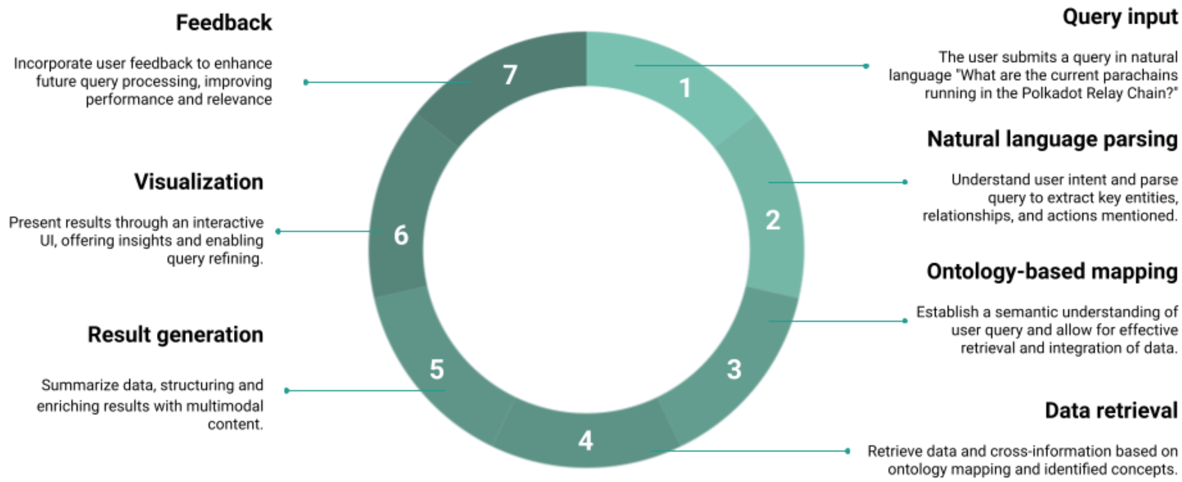

Fig. 2. Input query processing.

Polkadot Analytics foresees data integration in real-time by caching the Polkadot ecosystem data in a triplestore (i.e., Knowledge Base). The idea is to provide up-to-date information retrieval and analysis, as well as significant value in terms of enriching the knowledge base itself. This integration strategy supports complex queries, data analytics, and knowledge discovery based on the combined information from the triplestore and the Polkadot network.

Polkadot is built using the Substrate framework, which allows for the development of custom blockchain solutions. By mixing information about Substrate in the triplestore alongside Polkadot data, Polkadot Analytics may provide a comprehensive understanding of both the Polkadot network and the underlying Substrate technology. This can be particularly useful for developers, researchers, or enthusiasts who want to explore and analyze the relationships, patterns, and interactions between the two.

One key aspect of the query service is its integration with the POnto conceptual framework. Considering the input query processing mechanism, there is a constant demand for accessing all the relevant concepts present in the knowledge base. Listing 1 presents an example of SPARQL[1] query that could be used to retrieve all classes described.

[1] http://www.w3.org/TR/sparql11-query/

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?concept
WHERE {
  {
    ?concept a owl:Class .
  }
  UNION
  {
    ?concept rdfs:subClassOf ?superClass .
  }
}

```

Listing 1. Example of SPARQL query to access all classes from POnto.

Listing 2 presents an example of SPARQL query to access all Parachain instances specified in the POnto ontology. Queries such as "What are the parachains running in Polkadot?" could be translated to the structured query presented in this listing.

```
PREFIX ponto: <http://www.mobr.ai/ontologies/ponto#>
SELECT DISTINCT ?parachain
WHERE {
  ?parachain a ponto:Parachain .
}
```

Listing 2. Example of SPARQL query to access all Parachain instances from POnto.

The integration with POnto conceptual framework into the Polkadot Analytics, enables retrieving classes and instances, as well as types of classes such as the one exemplified in Listing 3 (i.e., to fetch all types of Wallet). In addition, we foresee exploring the conceptual framework for enabling inference and reasoning, supporting advanced analytics and decision-making.

```
PREFIX ponto: <http://www.mobr.ai/ontologies/ponto#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?wallet
WHERE {
  ?wallet rdfs:subClassOf ponto:Wallet .
}
```

Listing 3. Example of SPARQL query to access all types of Wallet class from POnto.

3. Survey

We planned a survey to gather participants' perspectives about the proposed Polkadot Analytics query engine service.

A questionnaire was designed to understand the profile of users interested in query services for DLT data and to identify the most common use cases for the proposed service. The questions drive participants to identify requirements and relevant features they would like to see in the query engine. Open-ended questions at the end of the questionnaire collect additional query suggestions and feedback that participants may have about the query engine and its development.

The target user groups are experts, developers, and general users from the Web3 community who are interested in retrieving and analyzing blockchain data, including cross-information among different chains. Ideally, we deem the participant selection process should promote a balanced group based on the following criteria:

a. Participants' level of knowledge and experience with DLTs in general;
b. Their experience in retrieving and analyzing blockchain data, and;
c. Their level of familiarity with the Polkadot ecosystem.

This strategy ensures a diverse representation of expertise and experience levels. Participants may be recruited through various channels, such as online forums, social media groups, and especially Polkadot community channels.

4. Final remarks

This case study provides insights for developers and software architects in designing and building query engine tools to retrieve and analyze blockchain data from the Polkadot network and its parachains. The identified use cases, driven by a controlled natural language (CNL) and the POnto ontology, offer a path for tool development.

The study also aims at evaluating the perceived value and effectiveness of the query service among participants from the Web3 community. Their feedback will help shape the CNL for supporting blockchain query services and validate the proposed draft ontology. We hope these collective insights will drive user-centric improvements and provide advanced data retrieval and analysis perspectives within the Polkadot ecosystem.

POnto conceptual framework aims at providing a structured data representation, enabling efficient storage and retrieval. The underlying SPARQL queries allow for precise information extraction, while semantic interoperability enhances data integration across different sources.

By incorporating domain-specific knowledge, POnto can provide accurate and relevant information to promote data integration, scalability, and flexibility, accommodating evolving requirements. Integration with machine learning may enhance Polkadot Analytics capabilities, combining structured knowledge with learning models.

Acknowledgement

This work was supported by a research grant from the Web3 Foundation and is publicly available at [2].

References

[1] Questionnaire – Community Survey. Available at https://docs.google.com/forms/d/e/1FAIpQLSfFXPOiH41HJMTpaK6qE906rQbh3ZHWVeH99pOsUxkTQ1hnxA/viewform?usp=pp_url

[2] Final paper version to be available on arXiv