

From: Predrag Gruevski predrag@kensho.com
Subject: [Moderator Action] Position Statement for W3C Workshop on Web Standardization for Graph Data
Date: 12 January 2019 at 06:36
To: group-data-ws-pc@w3.org



Dear Members of the Program Committee,

My name is Predrag Gruevski, and I am the lead engineer of the Knowledge Graph Infrastructure Team at Kensho Technologies, a machine learning and financial data analytics company. I would like to propose a discussion on integrating graph, relational, and other database types into a single unified data system.

For the last four years, my team has been building and maintaining Kensho's Knowledge Graph: a large property graph that captures facts about events and relationships in the world (e.g., new product launches by U.S. tech companies), and links them to appropriate time series data (e.g., those companies' stock prices), original documents (e.g., the companies' quarterly report filings), and relational datasets provided by 3rd party data partners. The Knowledge Graph covers a broad and growing spectrum of information, such as geopolitics, natural disasters, supply chains, mergers and acquisitions, and more.

A major pain point my team encountered while designing this system was the need for our engineers to simultaneously interact with a variety of different database systems in order to find and retrieve the data they need. As each type of database has its own query language and API that our engineers needed to master to become productive, inexperience and misunderstandings of these disconnected languages and APIs often resulted in bugs, performance issues, and other negative outcomes.

To address this problem, we designed an intuitive unified query interface based on GraphQL that allows our engineers and customers to retrieve data using the same syntax, regardless of how that data is represented in the underlying set of databases. In order to query the system, the user writes a declarative GraphQL query that can include a handful of custom directives for added expressive power. The system then splits up the query into sub-queries that target only a single database. Each sub-query is compiled to the appropriate database's native query language (SQL, OpenCypher, Gremlin, OrientDB MATCH, etc.), and the received results are post-processed to form the answer to the user's query. This process ensures that the databases' APIs are used correctly, consistently, and in a manner that maximizes query performance and minimizes network round-trips. Leveraging this unified query interface has significantly reduced product development time, while enabling the creation of more sophisticated tools that can handle the diversity and nuance of our clients' analytics needs.

My team at Kensho has open-sourced the portion of this project that compiles GraphQL to database queries:
<https://github.com/kensho-technologies/graphql-compiler>

Additionally, I wrote a blog post on the advantages that our compilation-based GraphQL interface offers over traditional GraphQL and other query interfaces: <https://blog.kensho.com/compiled-graphql-as-a-database-query-language-72e106844282>

My team encountered an interesting variety of technical challenges while designing the Kensho Knowledge Graph architecture. As the attendees of this workshop would likely have encountered similar challenges in a variety of other domains, I would like to facilitate an exchange of ideas regarding:

- approaches to creating systems that span graph, SQL, document store, time series, and other database types, and expose them via a unified query interface;
- tooling to support the use of unified query interfaces while minimizing negative outcomes (e.g., bugs and performance issues);
- proposals to evolve existing and upcoming graph query language standards to make unified query interfaces more performant, more expressive, and easier to build;
- current and future applications of unified query interfaces, including applications in machine learning, data analytics, data quality checks, data provenance tracking, and others;
- best practices in integrating purpose-built and legacy data infrastructure into a unified query interface.

Thank you for your consideration. If you need any further information, please do not hesitate to contact me.

Regards,

Predrag

