VCU College of Engineering

CS 25-319 Analyzing and Designing Complex Systems using GraphQL (Part II)

Project Proposal

Prepared for

Shailesh Deshpande

Bank of America

By

Diya Ram Mohan, Houda Lahrouz, Montel Marks, Jayson Urena

Under the supervision of

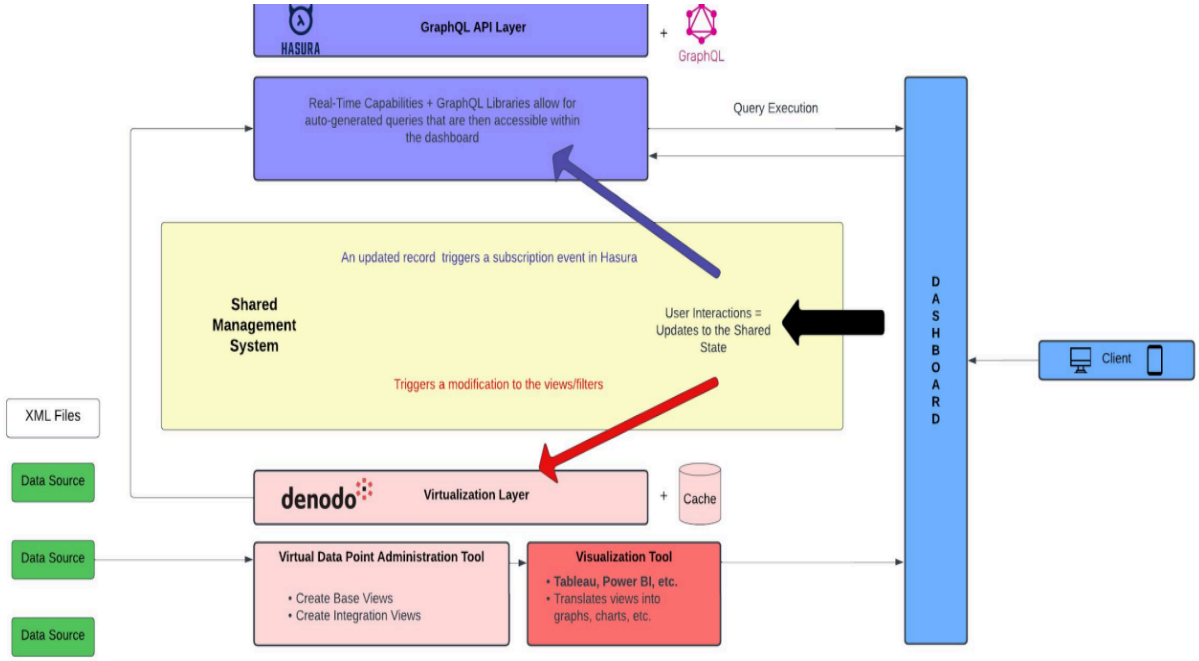
Thomas Gyeera

10/11/2024

Executive Summary

This project is meant to stand as phase two of the Designing Complex Systems using GraphQL capstone project sponsored by Bank of America. Phase two consists of providing necessary features to the current foundation developed in phase one. The result of phase 1 was a fully developed dashboard integrated with SQL databases, with data access provided through the use of a GraphQL API. The goals for phase 2 are connecting the data to Tableau to provide more tools for data visualization, providing data quality assurance through the use of reports, and enforcing data access control measures onto the application. The initial design, timeline, and goals will be developed throughout the semester of the 2024-2025 academic year, with the deliverables and design requirements being set before the initial draft of the proposal is finished. As of today, the team contract has been done, the GitHub repository has been set up, and a substantial amount of material has been collected as part of the project's literature review. The upcoming deliverables for the Fall 2023 semester are the project proposal, the fall design poster, and the preliminary design report.

Below is the diagram illustrating the project's Part 1 architecture:



(Anil et al., 2023)

Table of Contents

Section A. Problem Statement

Bank of America is looking to create a shared services layer to centralize and standardize the way the data is served up to the business. In the first phase of this project, the foundation was established. A dashboard was developed integrated with SQL databases using tools such as Hasura, Denodo, GraphQL, and Python. The second phase of this project aims to integrate more tools, develop more features, and inject more security into this application. Specifically, a lack of data visualization, quality control, and access control are all issues that have been identified surrounding this application. A lack of data visualization capabilities is an issue as a lack of avenues with which employees can interact with the data will lead to a decreased understanding of it and ultimately the best decisions will not be made. Enforcing quality and access control standards will handle issues surrounding costly data leaks and data security concerns. These issues could affect not only Bank of America but also its customers. By utilizing tools such as Tableau, and other tools soon to be determined for both ensuring quality control and enforcing access control these issues will be addressed, leading to a stronger and more robust application.

This project will continue a previous project, more specifically detailed in Appendix 2, that focused on implementing a Hasura-based database with a React front end. Since most middle-tier API services are limited to REST, it was necessary to research the benefits and implementation practices of GraphQL technology instead since it offers solutions to common REST issues such as data-fetching limitations (Lawi et al., 2021). This new technology from Facebook provides a new framework for a data access interface of web applications and a new query language for data requests (Hatrig & Perez, 2017). Additionally, the usage of GraphQL provides benefits for constant time response for each byte sent despite possible limitations due to the large response size (Hartig et al., 2018). In addition, GraphQL's ability to specify precisely what data is needed and its formal semantics allow efficient querying from multiple data sources, which is crucial for optimizing data retrieval in complex systems (Hartig & Perez, 2018). Finally, GraphQL federation enables the integration of multiple GraphQL services into a unified supergraph, allowing for scalable and flexible querying of distributed data sources (Stünkel et al., 2020). Knowing this, our project will continue on this research and analysis of GraphQL through data analysis on data quality throughout the application produced last year and its new updates.

Section B. Engineering Design Requirements

B.1 Project Goals (i.e. Client Needs)

Optimization of the piping from the database structure to the front-facing system

- Look and solve for any data leaks
- Implement as database monitoring tool (PostgreSQL or SQL Server Management Studio)
- Optimize the process for a large volume of calls/ queries
- Implement user roles to control who can update or delete data
- Create a testing platform for the already established dashboard to see what is available to us

Introduce the ability to run the API in outside visualization platforms with security

- Experiment with Hasura and Auth0 to see if any problems need to be addressed
- Hasura is the most important part of connecting the database with the forward-facing data layer
- Experiment with the Tableau platform
- Produce a seamless pipe to Tableau that can run all queries regardless of size and optimize for multi-user scenarios
- Test scalability with large-scale data implementations, live data storage, and multi-user API calls

Visualization Platforms

- Connect the data to Tableau by selecting the data source by importing the dataset. This gives you the ability to clean before processing
- Building visualizations can be accomplished by using the drag-and-drop fields into Rows and columns
- A dashboard can be created by combining multiple visualizations. This allows you to share the dashboard and share among users
- Given issues with Tableau, use alternate visualization platforms such as Power BI platform is similar to the method above

Optional Goals

- Add additional data sources to the data structure (Postgres or NoSQL component)
- Integrate multiple databases into the system (one SQL, one Postgres, and one Oracle)
- Fix prior implementation of GraphQL API to fit new needs of different data

B.2 Design Objectives

1. Optimized data grouping for performance is a big objective. The design should be able to prioritize the logical grouping of related data processes to enhance query performance (with GraphQL as the query language). Optimization at this step reduces complexity and response time.
2. Multi-layered security pipelines are designed to keep user access restricted to relevant information to the defined role. The design will leverage role-based access controls (RBAC) and encryption to secure sensitive data.
3. A primary focus will be on ensuring data integrity and query efficiency. In order to maintain high data quality we must look for data leaks in the system and increase liquidity. We must have efficient indexing strategies, optimized (GraphQL) queries, and engage in active monitoring of the system under a variety of loads.
4. We must evaluate and test the capabilities of the Hasura environment. The database will be deployed and tested within the environment to understand the performance thresholds and limitations.
5. We need to implement a seamless integration of data visualization tools. The database design will prioritize compatibility with data visualization tools like Tableau. Integration points will be planned to allow seamless connection and reporting capabilities, enabling real-time insights and interactive financial dashboards as a part of daily workflow.
6. If the integration of Tableau does not work or is not efficient enough we will switch to a Power BI platform that could allow for integration with modern business tools such as Office 365. Using this tool has less flexibility with data sources when compared to Tableau. However the integration is simpler, the dashboard is more user-friendly, and Microsoft is a legacy company.

B.3 Design Specifications and Constraints

Design must integrate with some of the previous teams' code and design and build upon the design to an extent. Specifically, the design should include the connection of data to the tableau through the format of the previous design.

Design must include security features to limit data visualization and access from the consumer viewpoint depending on security level or product team.

Data used to model a system must include certain characteristics to accurately test for data quality. Information should include sent time, time/date stamps, and additional information to provide useful visualizations.

Visualizations for the system should be quick to update with information on the quality of information as soon as the user attempts to access the data.

Design must operate within the architectural boundaries of last year's project including using Hasura for GraphQL access, tableau for data connection, and React as front end.

B.4 Codes and Standards

This will be added throughout the academic year as coding continues.

Section C. Scope of Work

C.1 Deliverables

Deliverables requiring campus access:

Presentations for the Fall Poster and Capstone EXPO will require access to campus for physical attendance and setup.

Certain system testing and access to secure databases may also require physical access to campus resources.

Deliverables that can be completed remotely:

- Reports: Writing and preparing reports (Preliminary Design Report, Project Design Report, Final Design Report) can be done remotely.
- Software Development: Code alterations, data governance work, and ensuring data quality can be completed remotely using shared platforms like GitHub and cloud services.
- Data Connection to Tableau: Data connections, integration, and testing for visualizations in Tableau can be worked on remotely, enabling data-driven insights for stakeholders.
- Data Visualizations: Visualization of data through Tableau and the React front end will be developed remotely to present actionable insights.

Resources required for remote work:

Access to licensed software (Tableau, development IDEs like VS Code, version control systems like GitHub).

Shared cloud drives/folders for collaboration.

Reliable internet connection for remote team meetings and access to resources.

Potential obstacles:

Ordering and integrating third-party software tools may require time for procurement.

Any additional hardware (servers for local hosting) may have extended lead times due to supply chain disruptions.

Mitigation strategy:

Place orders for third-party components early to mitigate potential delays.

Have contingency plans for any extended lead times by identifying alternative tools and services.

C.2 Milestones

This project will contain the following main milestones to ensure that we are on track to complete all project deliverables. After the project proposal, the first main milestone will be the understanding and testing of the previous team's code in order to determine what portions will be built upon to complete the necessary deliverables and which ports will not be considered. Following this, the completion of a design poster including information on background, design criteria, and analysis methods for the project by mid-November. The next milestone is the project design in December associated with the deliverable of the Project Design Report. This will contain a literature review and analysis and the completed design for the code to be completed in Spring 2025. Before winter break, the next milestone will consist of a review and any alterations necessary from the previous semester's code to meet the base requirement for our additions (updated base system).

The first milestone for the Spring semester will be the connection of the data to the tableau or a similar implementation of the process. Following that, the milestones will be directly associated with the deliverables for the project including data governance and security work by the start of March and then testing and viewing of data quality by the first of April. Finally, by the end of April, the final milestone of data quality and data visualizations on the React front end should be completed.

Milestone	Due Date
Data Updates and Analysis of Previous Work	November 1
Design Poster	November 15
Project Design Report	December 9
Updated Base System	December 20
Data Connection to Tableau	February 1
Data Governance	March 1
Data Quality Tests	April 1
Data Quality Visualizations	April 20

C.3 Resources

This project will contain the following main resources to ensure a more robust, data solution that adheres to industry standards and scaled efficiently. Here are all the resources:

Hardware Resources

- **High-Performance Computers (HPCs) or Servers**:
  - To host SQL databases, the Hasura GraphQL server, and additional tools for data processing and visualization.
  - On-premises or cloud-hosted servers may be required to ensure scalability and performance.
- **Local Development Machines**:
  - To work on the project, capable of running integrated development environments (IDEs) and local instances of databases or visualization tools.

Software Resources

- **Integrated Development Environments (IDEs)**:
  - Tools like **VS Code**, **PyCharm**, or **WebStorm** to develop and test the codebase in languages like Python, JavaScript, and SQL.
- **Hasura**:
  - To continue using and improving the GraphQL API that integrates with the existing SQL databases.
- **Denodo**:
  - For data virtualization and to manage diverse data sources in a unified, virtualized layer.
- **Tableau**:
  - For data visualization, allowing non-technical users to interact with and visualize the data.
  - This will improve visibility into data quality and business insights.
- **Python**:
  - For backend logic and data processing, integrating security and access control mechanisms, and potentially building custom solutions for quality control.
- **GraphQL**:
  - Continue to build on top of the existing Hasura-based GraphQL API for querying and mutating data.
- **Access Control Tools**:
  - Tools such as **Auth0**, **OAuth**, or **AWS IAM** to manage user authentication and authorization, addressing identified security gaps.
- **Quality Control Software/Tools**:
  - Tools like **Great Expectations** or **Datafold** for data quality monitoring and enforcing validation checks on incoming data.
- **Version Control System (VCS)**:

- **Git** and platforms like **GitHub** or **GitLab** are used to manage the codebase, track changes, and facilitate collaboration between team members.

Appendix 1: Team Contract (i.e. Team Organization)

Step 1: Get to Know One Another. Gather Basic Information.

Task: This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other's strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other's communication and working styles.

Team Member Name	***Strengths each member bring to the group***	***Other Info***	***Contact Info***
Jayson Urena	*Backend, Databases, SQL, Python, Scala, Java, C, RestAPI, Elastic Search, Testing.*	*Experience with Agile development, backend, RestAPI, and database integration.*	*urenajm@vcu.edu*
Diya Ram Mohan	*Web development experience React, Angular, Java, Typescript, Python, MSSQL, HTML/CSS*	*Experience with agile methodology, full-stack web development, and web development software like Figma, Swagger, etc.*	*rammd@vcu.edu*
Montel Marks	*Strategy, Statistical Analysis, Java, SQL, Python, SAS*	*Experience with statistical software, Data relations, and hypothesis testing*	*marksm2@vcu.edu*
Houda Lahrouz	*Java,Python,C, C++, SQL, CSS, JavaScript, full stack development*	*Experience with agile methodology, Designing websites, using Figma, Trello, and Github.*	*lahrouzh@vcu.edu*

Other Stakeholders	***Notes***	***Contact Info***
Thomas Gyeera	*Faculty Advisor*	*gyeeratw@vcu.edu*
Michael Karafotis	*Sponsor*	*michael.karafotis@bofa.com*

Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

Task: Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team's culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

Culture Goals	***Actions***	***Warning Signs***
Showing up to group meetings	- *Meeting reminders and confirmations the day before the meeting*	- *Verbal warning if a student misses or is over 20 minutes late to a meeting without warning given.* - *Advisor meeting if the student fails to attend or be on time for meetings.*
Informing the group of any issues or delays in assignments	- *Inform all partners about responsibilities and timeline* - *Set reasonable deadlines and communicate when extensions are needed*	- *Student shows up to meeting with no work done or misses deadline*
Maintain professionalism and respect with all members	- *Ensure all partners get a voice in any group decisions* - *Communicate any displeasures within a group*	- *Members show displease with the group or decisions made by the group*

Step 3: Time Commitments, Meeting Structure, and Communication

Task: Discuss the anticipated time commitments for the group project. Consider the following questions (don't answer these questions in the box below):

- What are reasonable time commitments for everyone to invest in this project?
- What other activities and commitments do group members have in their lives?
- How will we communicate with each other?
- When will we meet as a team? Where will we meet? How Often?
- Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will same person take on that role for the duration of the project?

Required: How often you will meet with your faculty advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings?
See examples below.

Meeting Participants	***Frequency Dates and Times / Locations***	***Meeting Goals Responsible Party***
Students Only	*Tuesdays After 4:00 Location will vary between Engineering Building and Online*	*Talk about over-the-weekend work/create a weekly goal*
Students Only	*Thursdays at 4:00 pm Location will vary between Engineering Building and Online*	*Problem-solving / Questions and adjustments to weekly goals or deadlines*
Students + Faculty advisor	*Fridays from 11 am to 12 pm varying locations between Zoom and advisors office*	*Update advisor on changes over the past week and any questions on the project*
Project Sponsor	*Thursday from 3:30 pm to 4:00 pm*	*Update sponsor on progress with the project, ask any questions, confirm project work is up to standards and timeline is maintained*

Step 4: Determine Individual Roles and Responsibilities

Task: As part of the Capstone Team experience, each member will take on a leadership role, ***in addition to*** contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

- **Before meeting with your team**, take some time to ask yourself: what is my "natural" role in this group (strengths)? How can I use this experience to help me grow and develop more?
- **As a group,** discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

Communication Leaders

Suggested: Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

Montel Marks

Suggested: Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

Jayson Urena

Common Leadership Roles for Capstone

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.
   Required: On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.
2. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.
3. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.
4. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of sub-systems in the prototype; develops and recommends system architecture and manages product interfaces.
5. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.
6. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

Team Member	Role(s)	Responsibilities
Diya Ram Mohan	Project Manager	- *Keep a detailed record of meeting notes and share them with the group* - *Manages schedules to ensure tasks are completed on time* - *helps write agendas and run group meetings*
Houda Lahrouz	*Logistics Manager*	- *coordinates all internal and external interactions;* - *following up on communication of commitments, obtaining information for the team;* - *Documents meeting minutes; manages facility and resource usage.* - *lead in establishing contact within and outside of the organization*
Montel Marks	*Financial Manager & Client Communication Leader*	- *Keeping track of spending and sponsor coordination* - *Keep weekly professional update summaries to send to sponsor* - *Relational analysis and structured categorizations*
Jayson Urena	*Faculty Advisor Communication Leader & Systems Engineer Test Engineer*	- *Managing communication with the Faculty advisor* - *Analyzing client design specifications and leading the design and integration of subsystems.* - *Overviewing testing procedures, protocols and schedules*

Step 5: Agree to the above team contract

Team Member: Diya Ram Mohan *Signature:* Diya Ram Mohan

Team Member: Houda Lahrouz *Signature:* Houda Lahrouz

Team Member: Montel Marks *Signature:* Montel Marks

Team Member: Jayson Urena *Signature: Jayson Urena*

Appendix 2: Past Project Reference

This project is the second part of a previous existing Bank of America capstone project from the past academic year. The previous project *CS 24-314 GraphQL - Providing Different Perspectives* created an application using mock Airbnb data to show a React frontend containing statistical information on the data, and a Hasura/GraphQL backend to access and format the data. The data from XML files are connected to Denodo and are mapped to create derived views that Tableau should be able to use and visualize for the dashboard. Additionally, a GraphQL schema was produced in Denodo to map the XML data to GraphQL. Denodo also provides functionality to publish APIs from GraphQL so Harusa can then configure Denoted as a remote schema and apply it. The resulting GraphQL API within Harsura is connected to the React dashboard as a shared state so any subscribed changes from Hasura are received and updated in real time. Additionally, the previous team implemented Auth0 authentication to safeguard data.

However, some portions of the project, in particular, will be continued this year including the Tableau visualizations as that portion was unable to be completed the previous year.

References

Anil, A., Gandhi, M., Goyal, N., Johnson, G. (2023). *Capstone 24-314 Architecture Diagram.*

Besta, M., Peter, E.K., Gerstenberger, R., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., & Hoefler, T. (2019). Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. *ACM Computing Surveys, 56*(1), 1 - 40.

Hartig, O., & Pérez, J. (2017). An initial analysis of Facebook's GraphQL language. *Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Databases and the Web (AMW)*.

Hartig, O., & Pérez, J. (2018). Semantics and complexity of GraphQL. *Proceedings of the 2018 World Wide Web Conference (WWW '18)*, 1155–1164. https://doi.org/10.1145/3178876.3186014

Lawi, A., Panggabean, B.L.E., Yoshida, T. (2021). Evaluating GraphQL and REST API Services Performance in a Massive and Intensive Accessible Information System. *Computers*, *10*(11), 138. https://doi.org/10.3390/computers10110138

Stünkel, P., von Bargen, O., Rutle, A., & Lamo, Y. (2020). GraphQL federation: A model-based approach. *Journal of Object Technology, 19*(2), 1-21. https://doi.org/10.5381/jot.2020.19.2.a18