Chair of Computational Modeling and Simulation
TUM Department of Civil, Geo and Environmental Engineering
Technical University of Munich

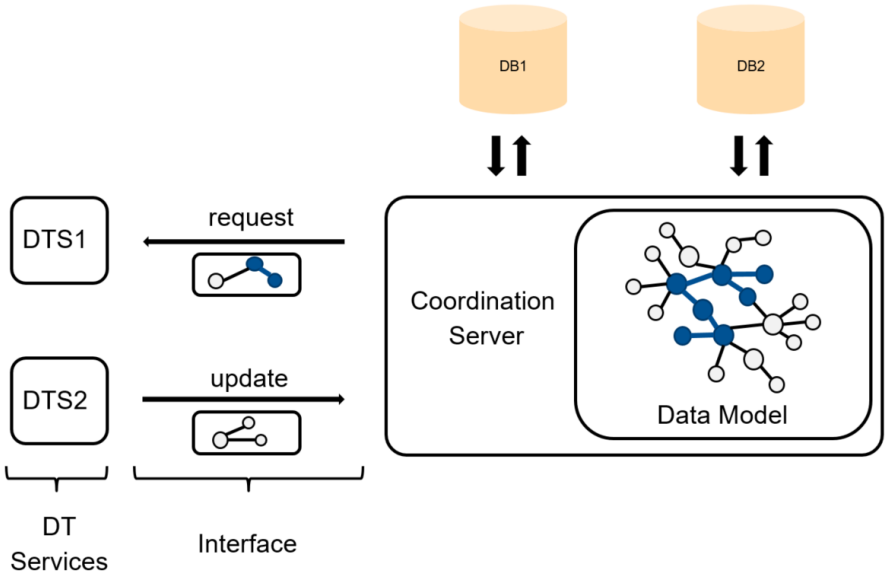
Modeling:
Mathematics:
Programming:
Science:

Software Lab:

Flexible Interface to access heterogeneous Digital Twin Data

Description

In the context of a digital twin in construction, a coordination server becomes necessary to manage the significant amount of heterogeneous data collected during the design and execution phase. A central data model is used to ensure that all of the data is well-organized and distributed across multiple databases*. So-called digital twin services will interact with the coordination server to request information from or send information to the central data storage through the means of an API. Depending on the construction project and its goals, a different set of digital twin services will come into play for every digital twin-supported project. However, for a single digital twin service, only a small subsection of the central data model is usually relevant. Using RestAPI endpoints would result in a great amount of manual implementation to supply all of the needs of every digital twin service. In the frame of this SoftwareLab project, a more flexible interface between the coordination server and digital twin services shall be implemented that supports the modular service structure.*



Task

Set up a simple coordination server with a flexible API that can be easily extended based on client-side requests. This will require the following steps:

- Set up a server with an API, a database, and a data model
- Set up an API client that will send requests and updates to the coordination server
- Design a flexible API that is able to handle access to subsections of the data model without manually defining a new API endpoint for every digital twin service individually (GraphQL might be well-suited for this problem and could be tried as one possible solution)
- Start with simple data requests and work your way to more complex queries and data updates
- Think about possibilities for API extensions to handle digital twin services that exceed the interface's current capabilities

Supervisors

Jonas Schlenger, Chair of Computational Modeling and Simulation, jonas.schlenger@tum.de
Sebastian Esser, Chair of Computational Modeling and Simulation, sebastian.esser@tum.de

References

[1] G. Block, P. Cibraro, P. Félix, H. Dierking, and D. Miller, Designing Evolvable Web APIs with ASP.NET. Sebastopol, CA 95472: O'Reilly Media, 2014.

[2] GraphQL, https://graphql.org/