University „Politehnica” Timișoara
Faculty of Automation and Computers
Study Program: Software Engineering

COMPARATIVE ANALYSIS OF REST AND GRAPHQL APIS IN WEB DEVELOPMENT

Disertation

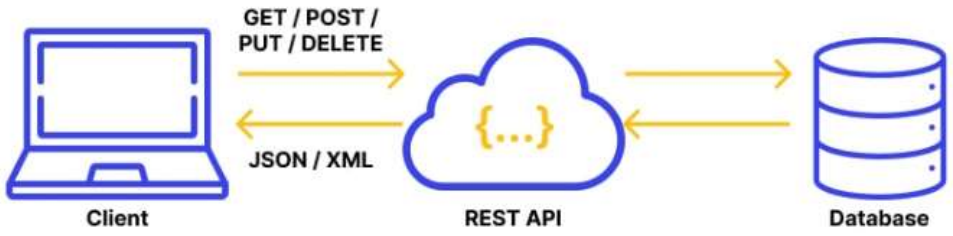
Scientific coordinator:
Prof. Dr. Habil. Ing. Alexandru TOPÎRCEANU

Candidate:
Ing. Bianca-Maria BELEA

Introduction

REST API

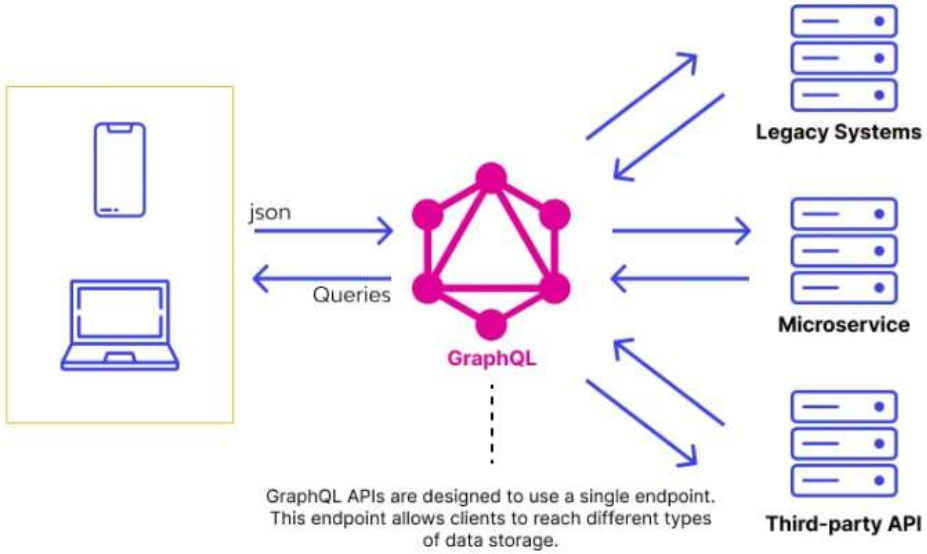
- Standard in API industry
- Resource-based architecture
- Fixed data structure
- Standard HTTP methods
- Uses multiple endpoints



[1]

GraphQL API

- Introduced by Facebook in 2015
- Query-based architecture
- Data structure defined by client
- Uses a single endpoint

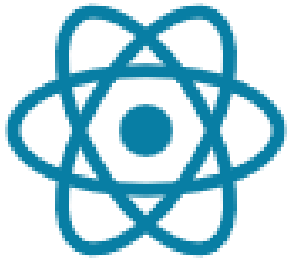


[1]

Implementation

- The same full-stack web application: Mentum – connecting university students in need of academic help with mentors
  - The same front-end
  - The same database
  - Different API implementations: REST & GraphQL

- Technology stack:



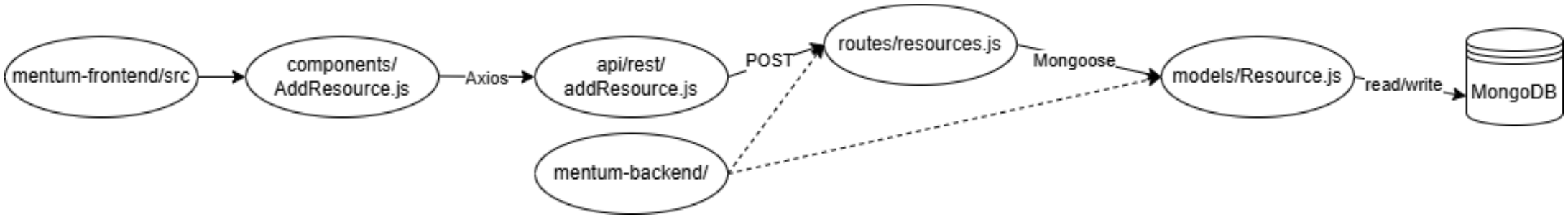
[2]

[3]

[4]

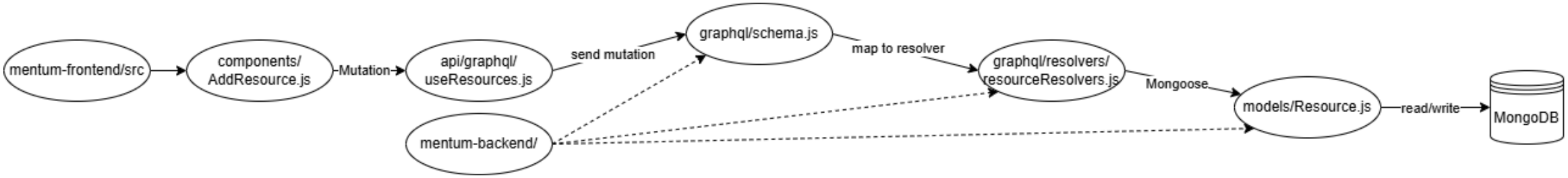
[5]

Implementation of REST API



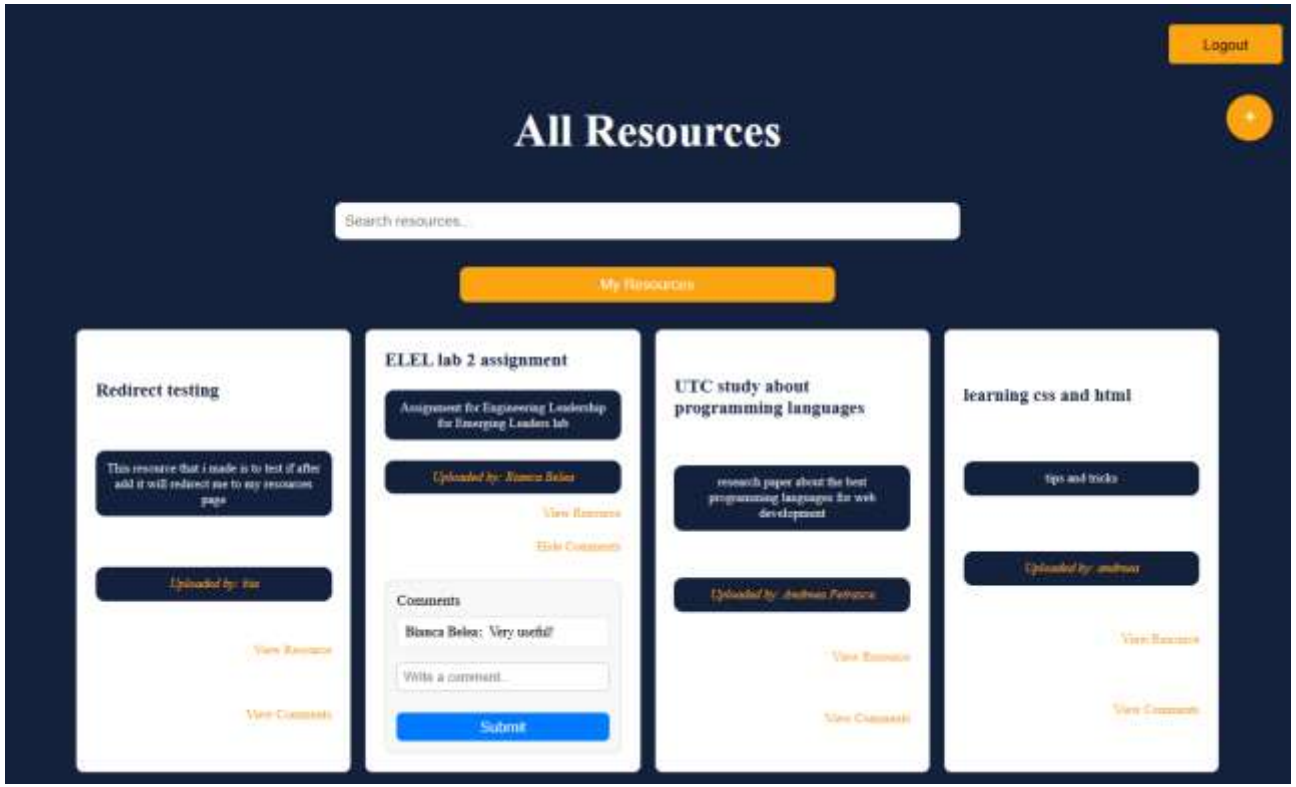
Flow diagram for REST implementation

Implementation of GraphQL API



Flow diagram for GraphQL implementation

Common front-end implementation



All Resources page in the user interface

Testing

- Phase-based execution plan:
  - Light load testing
  - Medium load testing
  - Stress testing
  - Breakpoint testing

- Metrics analyzed in the process:
  - Average Response Time (ms)
  - Data Sent (KB)
  - Requests Per Second (RPS)

Results

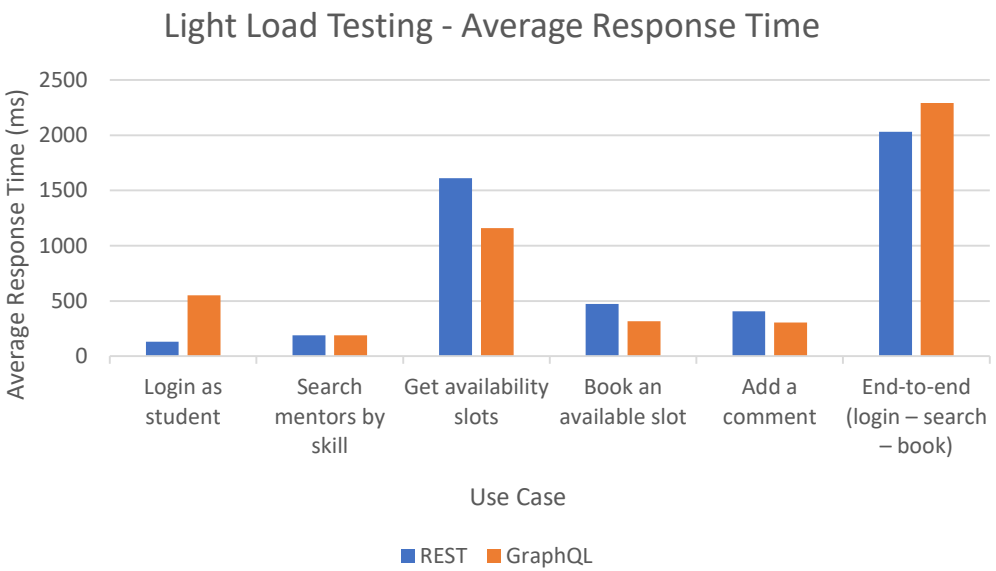
Metric / API Implementation	REST	GraphQL
Average Response Time (ms)	131.33	550.80
95th Percentile (ms)	151.75	563.88
Max Latency (ms)	489.60	2120
Requests/Second (RPS)	75.73	18.02
Failure Rate (%)	0.00	0.00
Data Sent (KB)	267	106
Data Received (KB)	1100	180

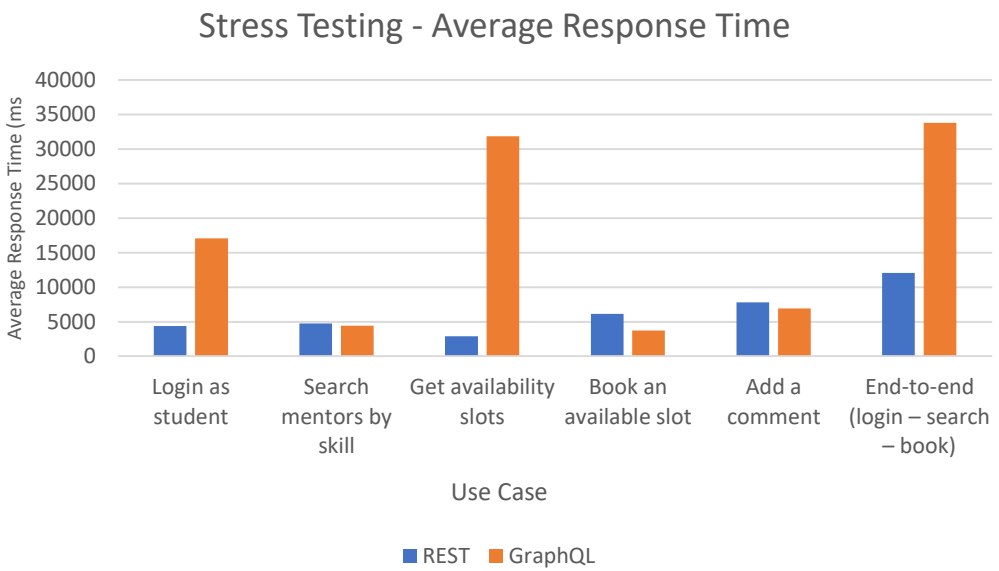
Metrics for light load testing, login as student

Metric / API Implementation	REST	GraphQL
Average Response Time (ms)	4360	17050
95th Percentile (ms)	8210	34190
Max Latency (ms)	8380	47230
Requests/Second (RPS)	74.85	17.43
Failure Rate (%)	0.00	0.00
Data Sent (KB)	802	448
Data Received (KB)	3584	714

Metrics for stress testing, login as student

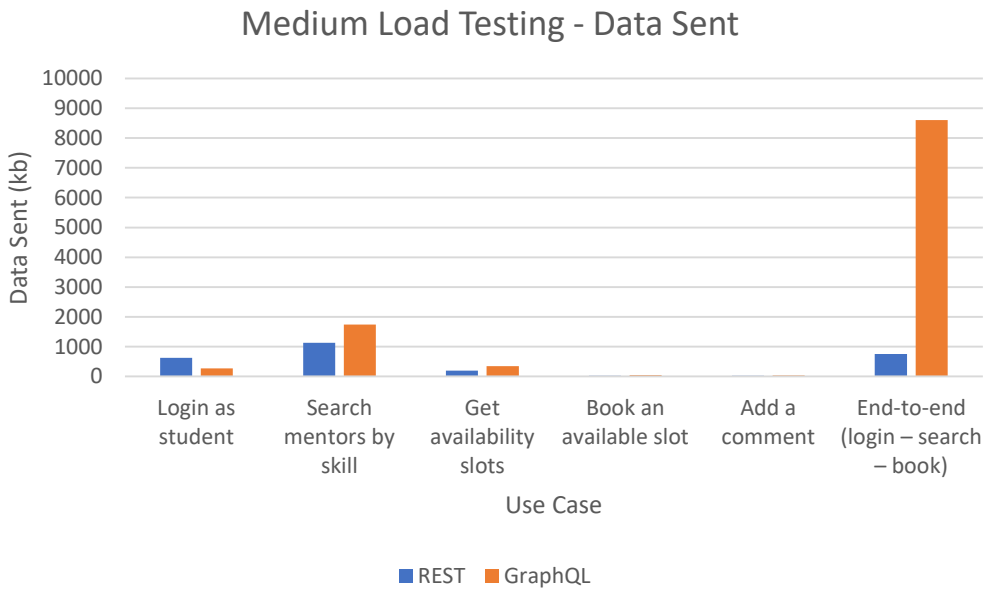
Results – Average Response Time

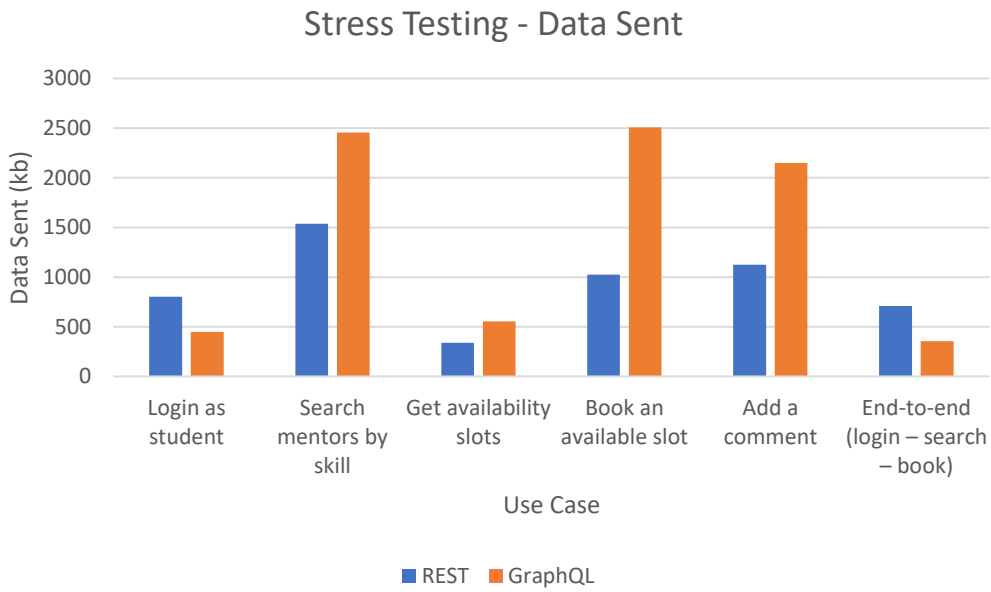




- REST is on average 30-50% faster than GraphQL
- REST maintains its stability in stress conditions
- GraphQL degrades under stress conditions, exceeding 30-40 seconds

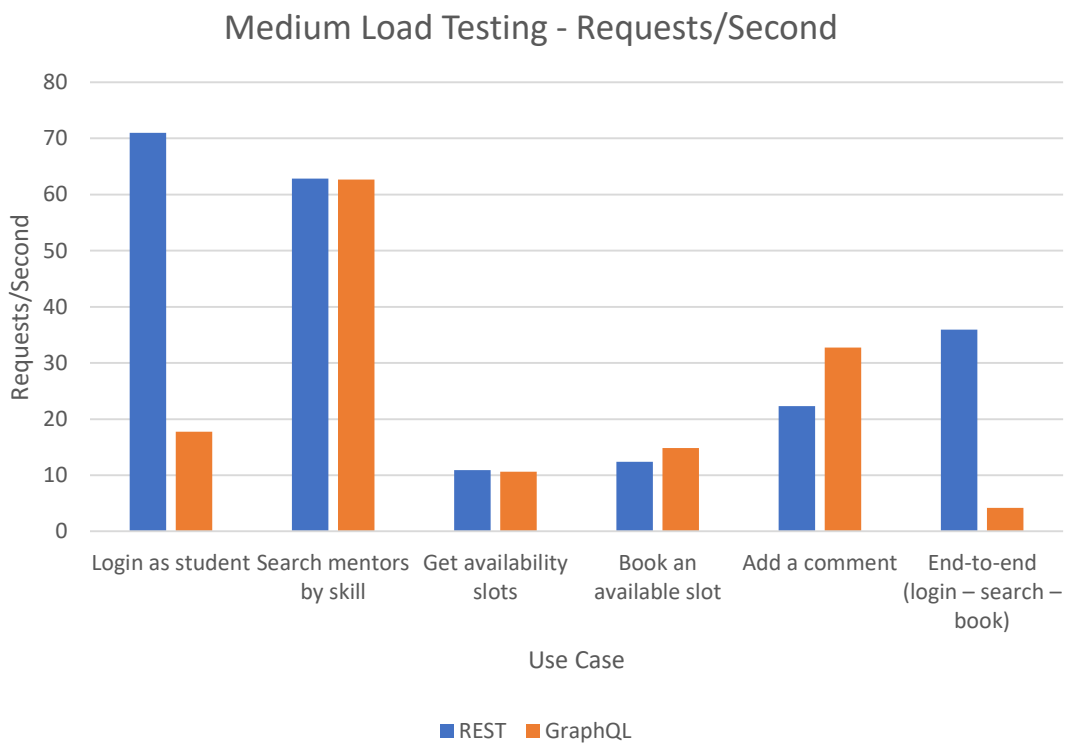
Results – Data Sent





- GraphQL experiences a huge spike for *end-to-end* flow
- Consequence of its detailed queries
- REST uses static routes, resulting in a smaller request body

Results – Requests Per Second



- This metric is strongly linked to *failure rate*
- Similar results for “read” operations
- REST is slightly more stable than GraphQL in use cases with nested calls

Conclusions

- The aspects presented in this research align with the existing state-of-the-art papers.
- REST API has a speed advantage compared to GraphQL.
- GraphQL is more flexible but its performance downgrades under stress conditions.

Metric	REST API	GraphQL API
Average Response Time	Faster	Slower in complex use cases
Throughput	Higher, more stable	Fluctuates under stress loads
Data Payload	Compact	Verbose queries
Failure Handling	Predictably	Sharp
Frontend Flexibility	Less dynamic	High granularity
Learning Curve	Beginner-friendly	Deeper back-end knowledge

Visual comparative summary between REST and GraphQL

When to use them?

- Use REST API for:
  - High-load environments
  - Low-bandwidth requirements
  - Mobile applications
  - Real life example: hospital management system

- Use GraphQL API for:
  - Dynamic front-ends
  - Applications where data evolves constantly
  - Concerns about over-fetching data
  - Real life example: e-learning platform

Image Cites

[1] https://www.wallarm.com/what/graphql-vs-rest-all-that-you-must-know

[2] https://commons.wikimedia.org/wiki/File:React_Logo_SVG.svg

[3] https://en.m.wikipedia.org/wiki/File:Node.js_logo.svg

[4] https://commons.wikimedia.org/wiki/File:MongoDB_Logo.svg

[5] https://en.wikipedia.org/wiki/File:K6-load-testing-tool-logo.svg