

Thesis  
B.Sc.

IDP

# Enhancing Distributed REST APIs on the Fly

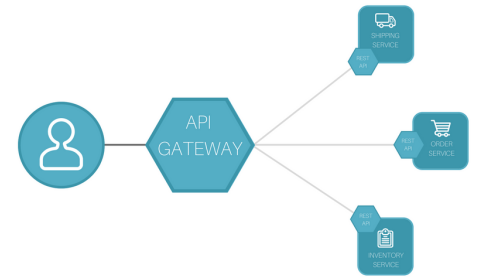
## Motivation

REST [1] APIs are core concepts of today's Internet architecture. They are the link that connects Services among themselves and to applications. However, their design is complex, and sometimes requirements from backend and frontend developers need to be more consistent—e.g., fast development and as few requests as possible vs. an extendable and maintainable architecture.

This thesis shall investigate how an intermediate API gateway [2] can be used to automatically transform and enhance a REST API. Such a gateway can only be built if a certain level of standardization is enforced, e.g., with json:api [3] or OpenAPI [7].

Possible transformations are:

- parsing of HATEOAS responses to a simplified JSON version
  - following of linked data to different services in HATEOAS responses (on demand)
  - providing a websocket interface (e.g., with [4]) to access the HTTP endpoints
  - generation a GraphQL [5] API for clients to access the REST endpoints
- Systematic Literature Research [6]
  - Development of an example application with a REST API
  - Development of an API gateway that can handle the transformations
  - Measurement study on the performance implications in a multi node environment



## Your Task

## Requirements

Bash, Golang, and Python Django or Java Spring

## Contact

Markus Sosnowski [sosnowski@net.in.tum.de](mailto:sosnowski@net.in.tum.de)  
Florian Wiedner [wiedner@net.in.tum.de](mailto:wiedner@net.in.tum.de)

## References

- [1] [https://de.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://de.wikipedia.org/wiki/Representational_State_Transfer)
- [2] <https://www.redhat.com/de/topics/api/what-does-an-api-gateway-do>
- [3] <https://jsonapi.org/>
- [4] <https://socket.io/>
- [5] <https://graphql.org/>
- [6] <https://libguides.csu.edu.au/review/Systematic>
- [7] <https://www.openapis.org/>

