Distributed API Rate Limiting

APIs are an important part of TomTom Telematics' platform strategy. In the longer run, all data will be exposed via service interfaces. This applies to data provided to TomTom's customers as well as data exchanged between internal services.

Rate limiting is a means protecting API servers and downstream services against load peaks. Rate limiting also ensures that the number of requests stays within the quotas booked as part of the customer's service-level agreement.

Distributed rate limiting for APIs comes with a number of interesting challenges:

- TomTom always runs multiple instances of API servers. The rate-limiting algorithm must work distributedly to ensure a given quota across all servers. An option is using Redis as shared data-structure server for that purpose.
- The rate-limiting algorithm should be as fast as possible, should use as little resources as possible (memory, CPU), and should cause as little network traffic as possible.
- The rate-limiting rules should be flexible. It should allow configuring rules that react on different types of requests. For example, a rule may apply to all POST requests, another one to GET requests matching a certain URL pattern.
- The rate-limiting algorithm should work within both Java servers (SpringBoot) and JavaScript servers (node.js). Common functionality may be written as Lua scripts, which are loaded into Redis.

Objectives of this student research project are the comparison of existing rate-limiting algorithms, the design of an adapted algorithm fulfilling the challenges mentioned above, the implementation of a prototypical solution, and finally the test of its correctness and performance.