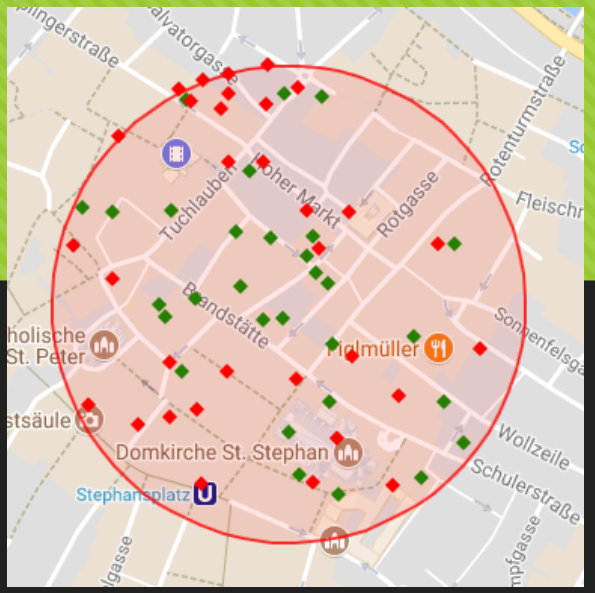
Urban Area Parking Spot Monitoring System

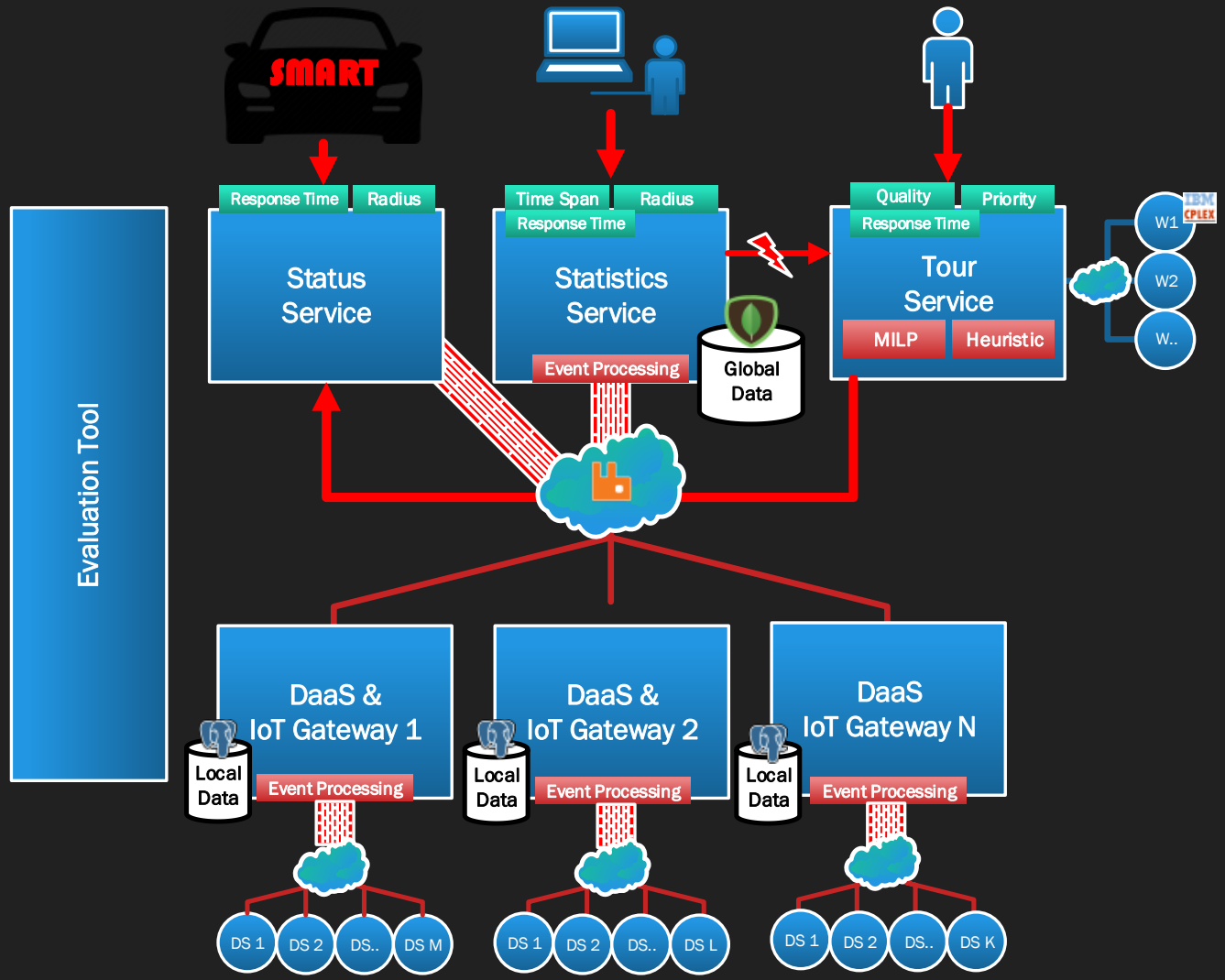
Thomas Kaufmann

Scenario



- Distributed monitoring of parking spots
- Smart parking spots with sensors as data sources
- IoT Gateways retrieve & process real-time data from DS
- Analytics Services retrieve & process data
  - Statistics Service (time period & range-queries, global state analytics)
  - Status Service (query current status for a certain position)
  - Tour Service (request a tour in a certain area)

System Architecture



QoS & Data Concerns (QoD)

- **Statistics Service (QoD)**
  - Timeline (considered history)
  - Granularity (considered range - radius)
- **Tour Service**
  - Elasticity due to different algorithms (optimal vs. heuristic) → Quality of Data (cost)
  - Completeness (abort due to time-out for huge instances)
- **Status Service**
  - Response Time (e.g. avoid *obsolete* results)
  - Granularity (considered range - radius)

→ AOP techniques to trace the response time of service invocations

Elasticity

1. **Statistics Service monitors global state**

   ➔ recurrent analytics task detects peak loads (MapReduce in MongoDB) and notifies Tour Service

   ➔ Tour Service adapts data quality based on number of tour requests & load

   ➔ user is notified

2. **Master-Worker & Priority Task-Queue allows for elastic utilization of resources**

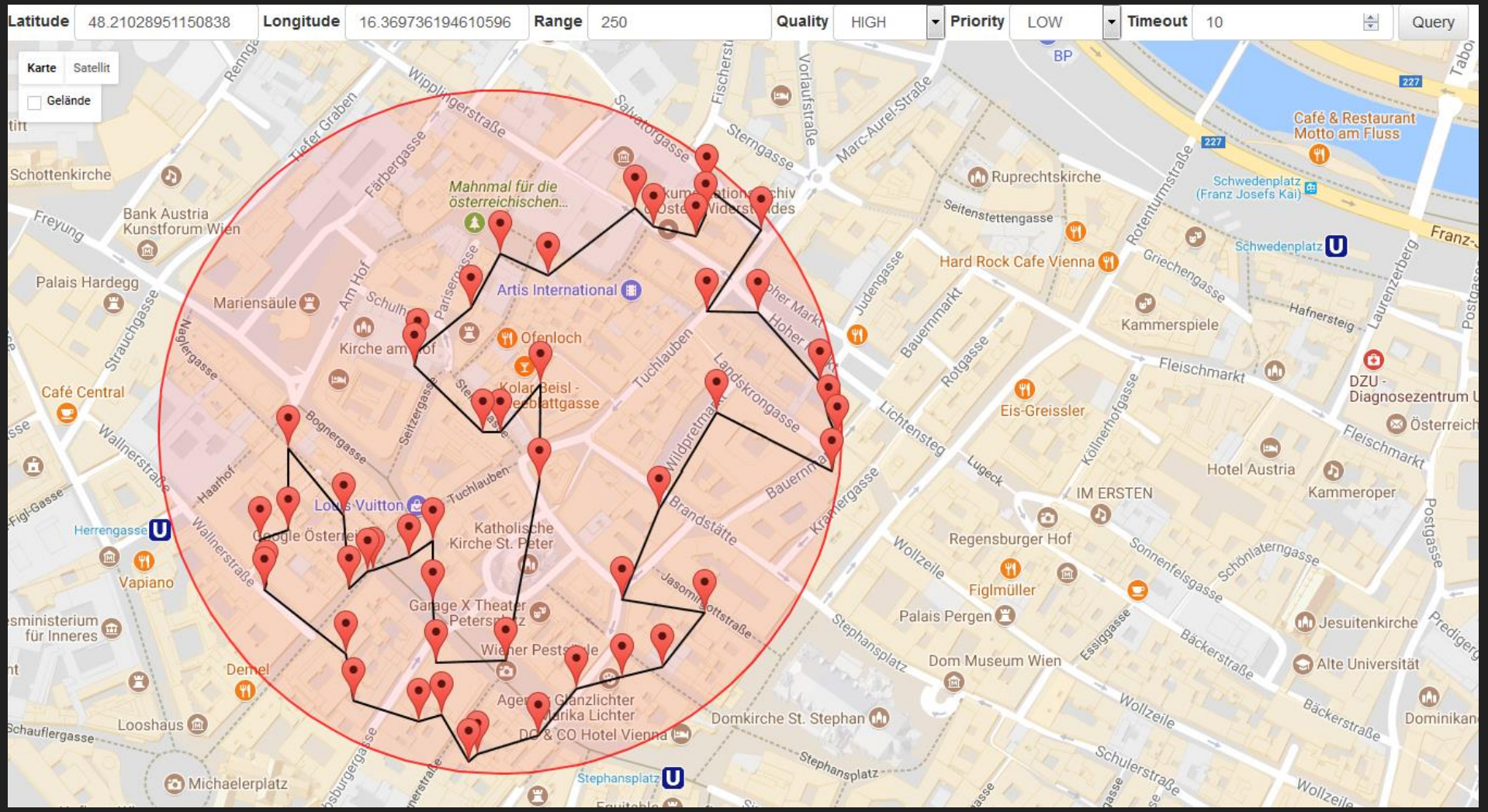
3. **Evaluation tool monitors all service calls (input & output)**

   ➔ data could be used to determine costs & resource demands in the system

Technologies

- Docker
- Spring: Boot, Data, AMQP
- PostgreSQL to persist local
- MongoDB for global status & request monitoring
- RabbitMQ
- Apache Flink
- IBM CPLEX
- 2-opt neighborhood Simulated Annealing + NN-Heuristic
- .NET for Simulation Agents

Tour Service Interface



Status Service Interface