Towards a Semantic Message-driven Microservice Platform for Geospatial and Sensor Data

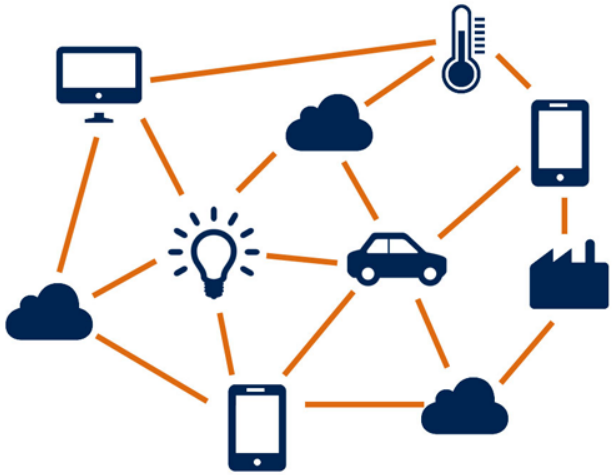
Matthias Wauer and Axel-Cyrille Ngonga Ngomo

AKSW Research Group, University of Leipzig, Germany
Data Science Group (DICE), Paderborn University, Germany

Overview

Motivation

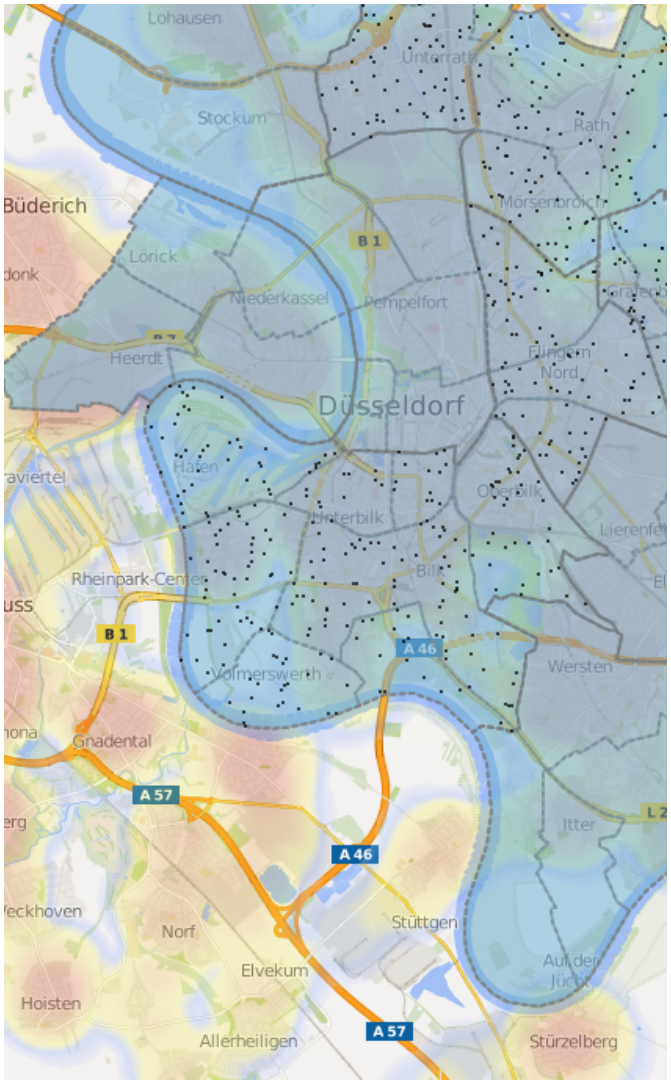
Why GEISER*?*



- Flexible integration of live data
  - Sensor data
  - Social media feeds
- Inherently related to geospatial data
  - Location of sensor measurements
  - Places mentioned in tweets
- Potential applications:
  - Routing
  - Geomarketing
  - Analytics (e.g., business intelligence, predictive maintenance)
- Issues:
  - Small but frequent service requests
  - Flexible and reliable communication
  - Scalability

Use Cases

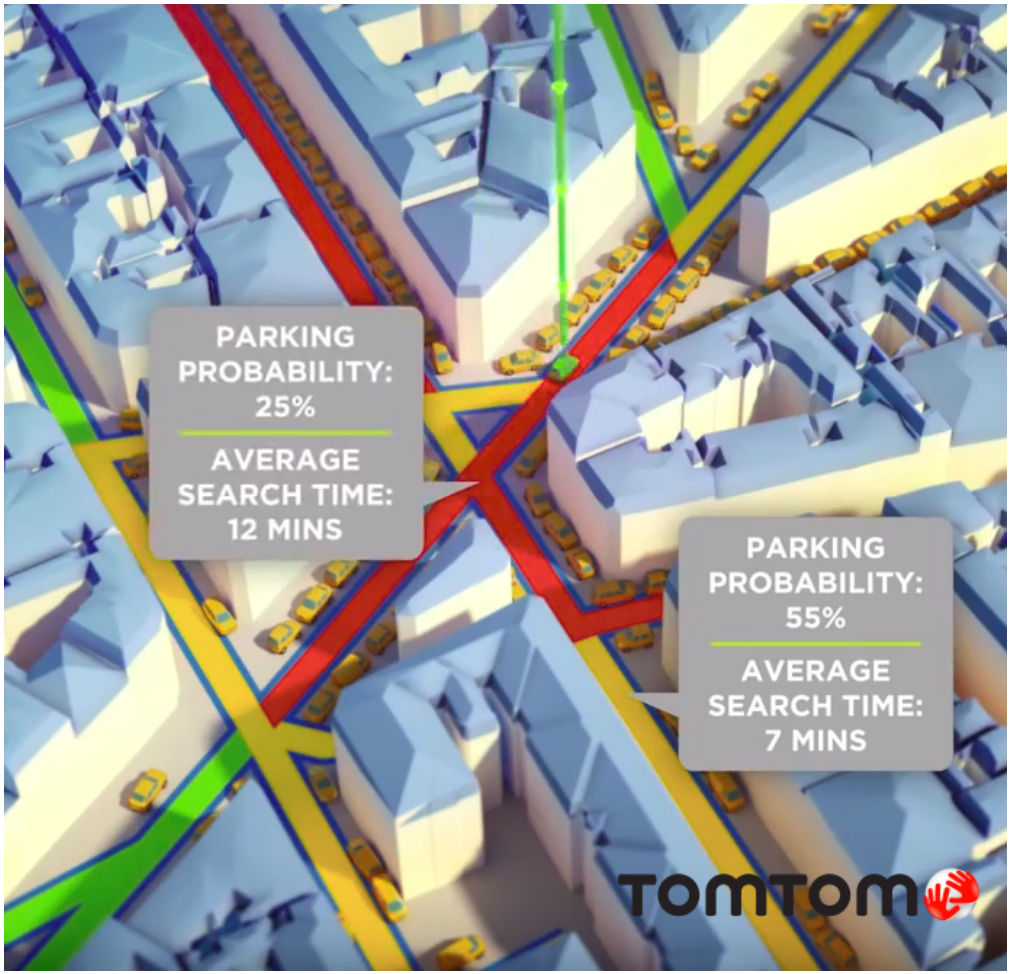
Use Case 1: Data-driven Geomarketing



- Current situation for local businesses:
  - Geomarketing based on static data
  - No fine-grained dynamic adaptations
- Goals:
  - Enable companies to adapt products, services, offers etc. to their audience
  - Integrate and interpret live data
- Relevant data:
  - Events from social media / news
  - Mobility and cellular network data
  - Weather warnings and forecasts
  - Own customer data

Use Case 1: Data-driven Geomarketing

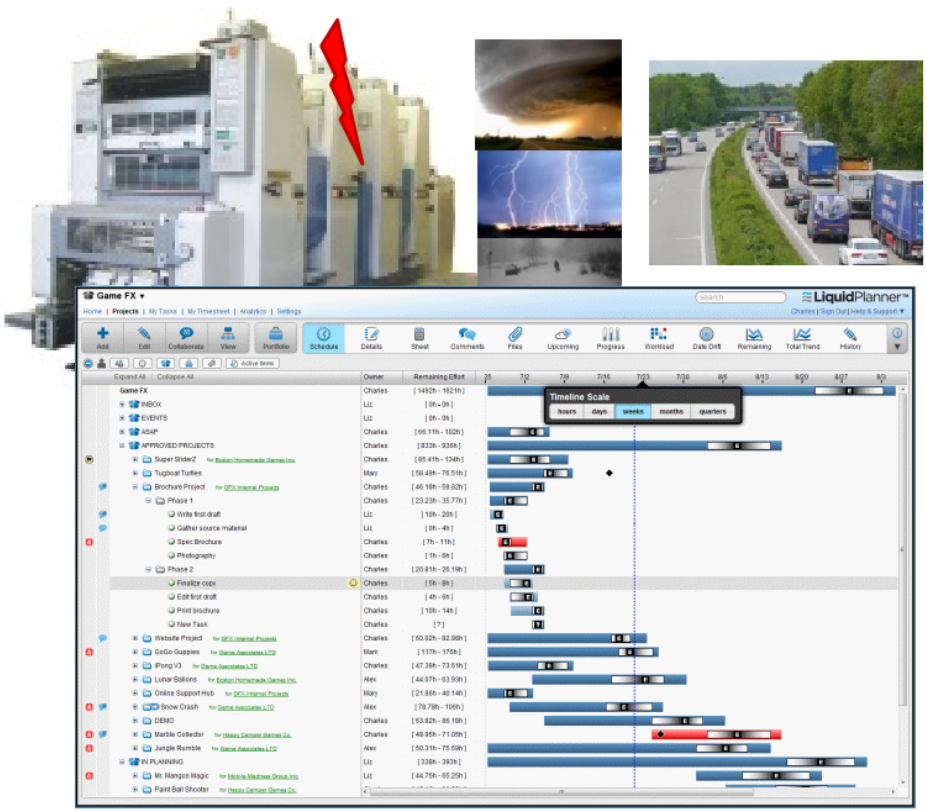
Use Case 2: Intelligent parking



- Current situation:
  - Finding road-side parking in cities is difficult
  - Little support from satnavs
- Goals:
  - Compute parking probabilities
  - Provide approximate routing service on satnav device
- Relevant data:
  - Sensor data (floating car data)
  - Mobility / cellular network data
  - Traffic events
  - Events (e.g., football matches)

Use Case 3: Predictive maintenance and industrial service

Device sensors Weather Traffic



- Current situation:
  - Sensor networks in industrial appliances not connected to service technician scheduling
  - High cost of halted manufacturing processes
- Goals:
  - Link predictive maintenance with service deployment planning and spare part logistics
- Relevant data:
  - Predictive maintenance data
  - Weather forecasts and warnings
  - Traffic information

Approach

GEISER *architecture*

Use Cases & Applications

Deployment & Maintenance

Authentication & Security

Platform & Service Configuration

Dev. Best Practices & SDK

Unified GEISER REST API

GEISER Services

Application Services (only samples included, use case specific)

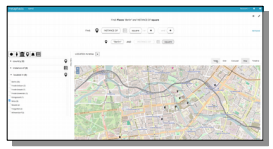
Generic Services, e.g. LIMES, DEER, FOX, AGDISTIS

Data Services (querying, manipulation, geospatial, etc.)

AMQP BUS

GEISER Platform

Compatible Tooling (loosely coupled)



Features:
visualization, exploration, dashboarding, data analytics, semantic search, etc.

Examples:
metaphactory, Facete, mappify, etc.

Private or public cloud
(Semantic Databases, SPARK ecosystem, SANSA ...)

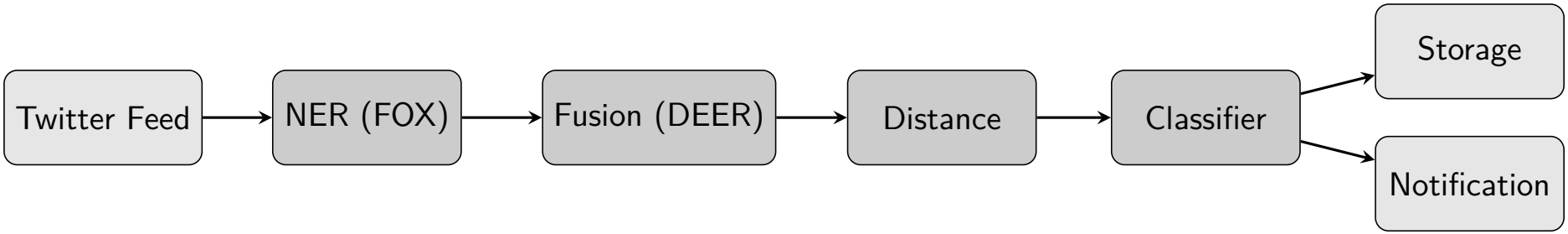
Compute Infrastructure

GEISER *architecture*

- Message bus
  - RabbitMQ based (flexible exchange patterns, reliable asynchronous messaging, simple consumer implementation, MQTT support)
  - Usually AMQP topic exchanges (most flexible)
- Service layers
  - Data services (for accessing, querying and manipulating RDF data)
  - Generic services (e.g., LIMES for interlinking, DEER for data fusion, FOX and AGDISTIS for extracting unstructured data)
  - Application services (e.g., probabilistic routing or weather forecast parsing)
- Usage
  - REST API for external requests (application-specific)
  - External tools like metaphactory

Invocation of services: Geomarketing use case

Example: Extracting and processing tweets for event information



- Twitter Feed pushes new tweets to message bus, e.g.: `"ESWC started in Heraklion"`
- NER annotates identified Named Entities
- Fusion applies an enrichment pipeline, e.g., adding geocoordinates and geometries from related resources (via `owl:sameAs`)
- Distance computes geospatial distance to region of interest, e.g., `st_distance(entity.geom, Greece.geom)`
- Classifier filters messages, e.g., by distance, discards irrelevant tweets
- Storage adds annotated tweet to RDF store

Message passing

- Message passing (instead of RPC between a controller and services):
  - Messages flow through list of services (see example)
  - Invocations specified in AMQP routing key, e.g., for messages from Twitter: `ner-v1.fusion-v1.distance-v1.classifier-v1.storage-v1`
  - Workflow can be modified at runtime

```
// routing key: fusion-v1.distance-v1.classifier-v1.storage-v1
// ... other header fields ...
{
  "@context": "http://example.org/context/fox.jsonld",
  "text": "ESWC started in Heraklion",
  "fox:": [{
    uri: "http://dbpedia.org/resource/Heraklion",
    name: "Heraklion"
  }]
}
```

Example message from NER service in JSON-LD format

Implementation

Implementation

- Implementing services
  - Wrapper for services already exposing a REST API[1]
  - Support library for developing an AMQP interface, based on Spring AMQP:

```
@Component
public class MyService {
  public static final String ROUTING_KEY = "myservice-v1.#";
  public static final String QUEUE_NAME = "myservice-v1";

  @RabbitListener(bindings = @QueueBinding(key = ROUTING_KEY, exchange
      = @Exchange(type = ExchangeTypes.TOPIC, value =
      ServiceUtils.EXCHANGE, durable = "true", autoDelete = "true"),
      value = @Queue(autoDelete = "true", value = QUEUE_NAME)))
  public void handleMsg(@Payload Message message) throws IOException {
    /* handle request */
    rabbitTemplate.send(ServiceUtils.EXCHANGE,
        ServiceUtils.nextRoutingKey(message), MessageBuilder...build());
  }
}
```

[1] https://github.com/AKSW/micropipe-proxy

Deployment

- Services are dockerized and configurable, e.g., regarding inputs
- Set of services required for a use case specified in Docker Compose, e.g.:

```
services:

  geiser-twitter-feed:
    image: mwauer/geiser-twitter-feed
    environment:
      - locations=-122.75,36.8,-121.75,37.8,-74,40,-73,41
      - routing=ner-v1.fusion-v1.distance-v1.classifier-v1.storage-v1
[...]
  storage:
    image: mwauer/geiser-rdf-writer-service

  rabbit:
    image: rabbitmq:management
    ports:
      - 5672:5672
```

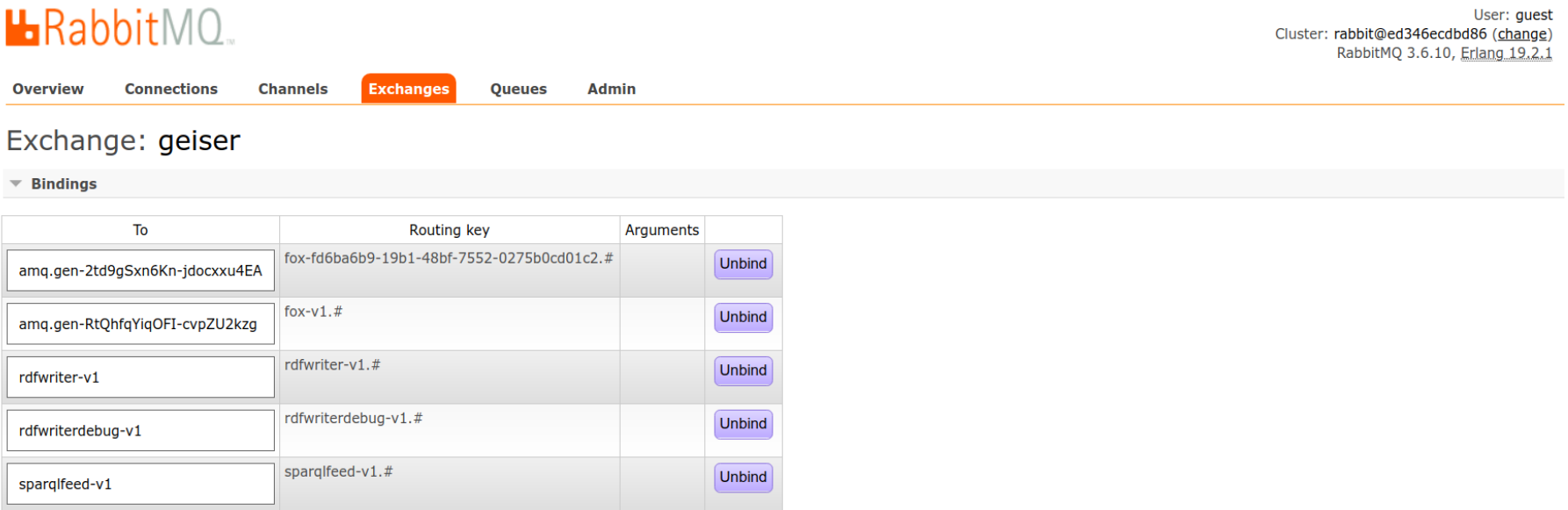
Available services

- Generic services
  - NER and Relation Extraction (FOX/AGDISTIS)
  - Translation (APIs like Yandex or DeepL)
  - Interlinking (LIMES with RADON)
  - Fusion (DEER)
  - Transformation (Sparqlify)
- Data services
  - Basic RDF reading and writing services
  - RDF graph store, repository, SPARQL (in work)
  - Twitter feed (in work)
- Application services (internal)
  - Weather (openweathermap.org, DWD)
  - Pollen warning (DWD)
  - Routing (TomTom)
  - Traffic (TomTom)
  - Reachable areas (TomTom)

Evaluation

Validation

- Functional validation
  - Running service composition similar to data-driven geomarketing scenario
- Component evaluation:
  - NER: FOX at OKE 2017 challenge
  - Semi-automatic extraction of RDF from tables: TAIPAN
  - Interlinking: RADON (rapid discovery of topological relations)



Related Work

Related Work

- Semantic Service Infrastructures
  - METEOR-S
  - SADI (Semantic Automated Discovery and Integration)
  - SSWAP (Simple Semantic Web Architecture and Protocol)
  - MicroWSMO / hRESTS
- Semantic Data Processing
  - LinkedPipes ETL / UnifiedViews
  - GeoKnow
  - EW-Shopp

Conclusions and Future Work

Conclusions and Future Work

- GEISER, a platform for
  - Flexible processing of geospatial and sensor data
  - Using semantic technologies
  - Message-passing approach
- Future Work:
  - Implementation of the use cases
  - Scalability of the platform
  - Support in components (e.g., interlinking on data streams)

Matthias Wauer
wauer@informatik.uni-leipzig.de
AKSW Research Group at Leipzig University
DICE Group at Paderborn University
http://aksw.org/MatthiasWauer

The research and development project that forms the basis for this report is funded under project No. 01MD16014E (GEISER) within the scope of the Smart Services World technology programme run by the Federal Ministry for Economic Affairs and Energy and is managed by the project management agency at the German Aerospace Center (DLR-PT). The authors are responsible for the contents of this publication.