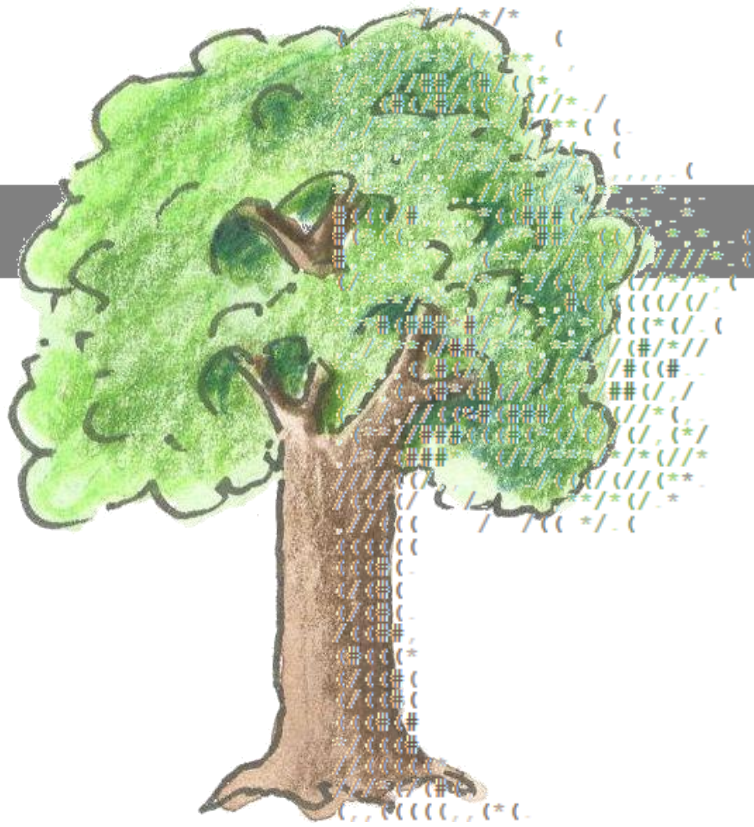
Karlsruhe Institute of Technology

Environmental State Portals at a Glance

Thorsten Schlachter, Aleksandar Jovanovic

Institute for Automation and Applied Informatics (IAI)



Outline

- Partners
- Core Ideas and Principles
- System Architecture
  - Data/Webcache
  - Search Engines
- Technology
- Licences
- Examples

- How to Start
  - „Normal“ Onboarding for new LUPO Partners
  - What do we need?
  - Suggestion for a Procedure
  - Critical points to be clarified

LUPO-Partners

LUPO = **L**andes**u**mwelt**po**rtal
= Federal State Environmental Portal

Federal State	Portal	Mobile App
Baden-Württemberg	✓	✓
Bayern	✓	
Brandenburg	✓	
Nordrhein-Westfalen	✓	
Sachsen-Anhalt	✓	✓
Schleswig-Holstein		✓
Thüringen	✓	✓

Institutions and Companies

- Environmental ministries and authorities in the respective federal states

- Contractors
  - Karlsruhe Institute of Technology (KIT / IAI)
    System architecture, data services, Web components

  - Convotis AG
    Liferay Themes, operation portals and services, development mobile app and reporting backend

  - DECON-network
    Editor full-text search engine

  - *Content-related work*
    IT service Magdeburg
    Web4All
    AHK (preparation of geodata)

No topic for today: Environmental Apps

…based on LUPO services

LUPO: Core Ideas and Principles

- One environmental state portal as **single entry point** to environmental information (for the respective federal state)
- **Use of existing data** - (practically no own/new data)
- **Quick overview** in the portal ➔ linked to **details** in the respective (specialist) system
- Data is provided via "**Web cache**" services ("copy")
- LUPO…
  - is a **Toolbox** of frontend and backend components
  - has a **(Micro)service-oriented architecture** and uses Cloud services
  - uses **Liferay Portal** as container of the Web application
  - benefits from sharing of licenses, operational costs, know how, etc.

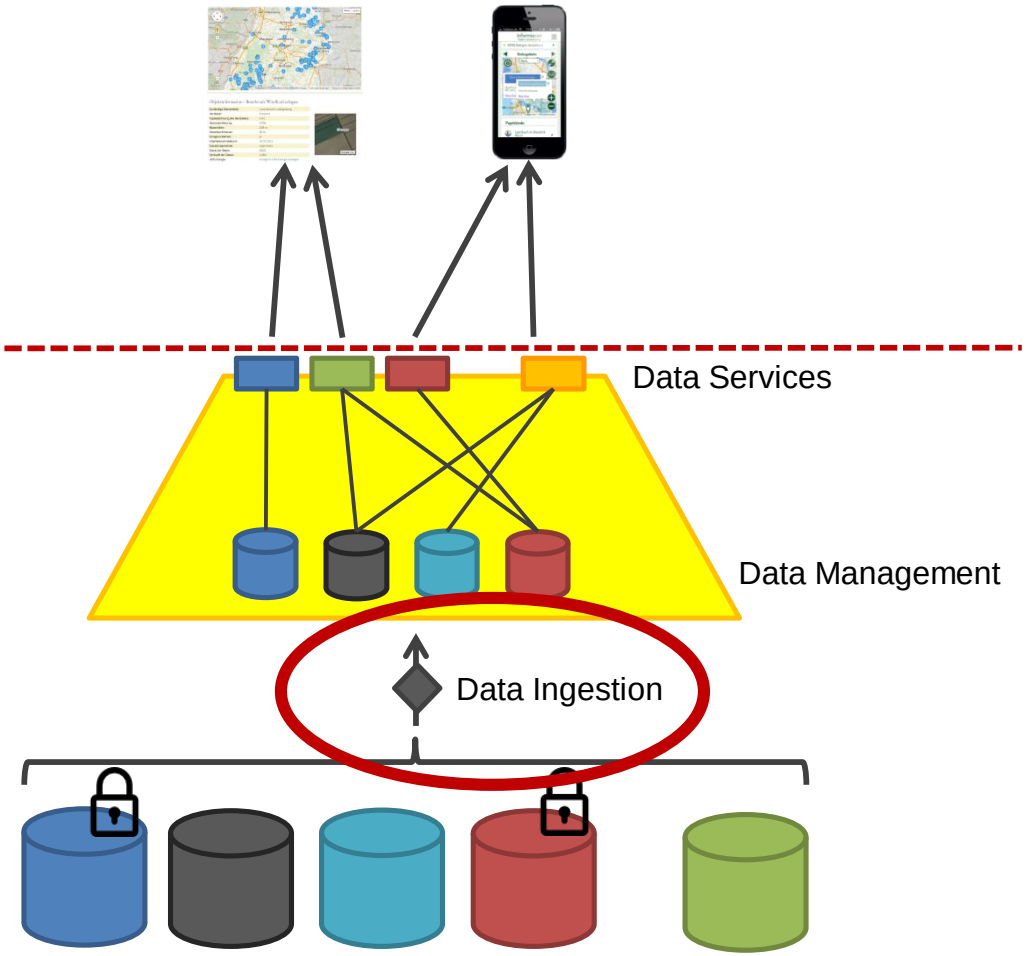
Webcache Architecture

Applications

SoA

Web Cache

(possibly reduced data, read-only, highly available, scalable)



Data Sources

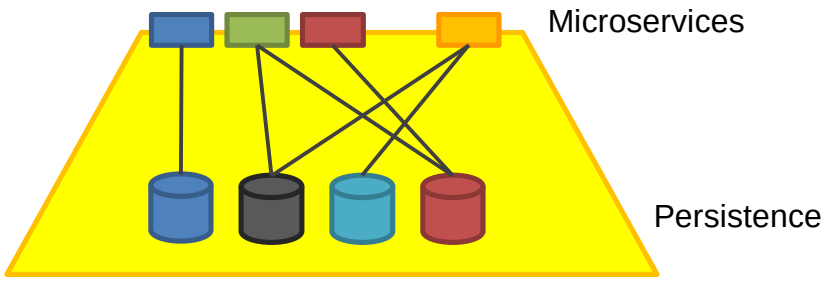
(complete data, read&write)

Types of Data

- **Unstructured and semi-structured data**: full-text search in ...
  - Websites (e.g. administration‘s websites)
  - Specialist portals and applications, document stocks
  - Search cascades, e.g. state statistical office, Service BW
- **Measurement data**
  - Water levels, air quality, weather, pollen, avalanche warning, ...
- **Spatial and/or (structured) object data**
- **Metadata, catalog data**
- **Others**
  - News feeds (e.g. press releases)
  - calendar of events

Webcache and (Micro-)Service Backend

- (Generic) **data services**
  - **Masterdata**
  - **Data schemes** (structure, semanics)
  - **Timeseries**
  - (Media and) **digital assets**
  - (Full text) **search**
  - **Spatial data (CartoDB)**
  - Metadata
  - Links and relations
- Additional services
  - Application configuration
  - Data discovery



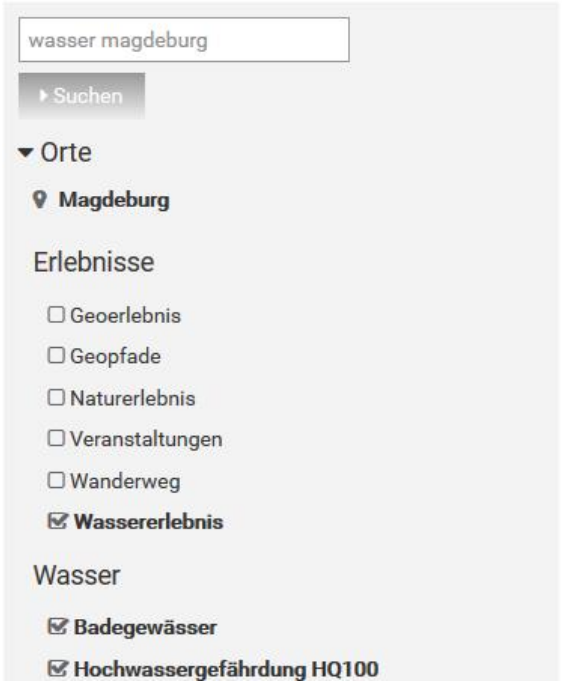
- **Persistence layer**
  - SQL, e.g. MySQL, CloudSQL, PostgreSQL ...
  - NoSQL, e.g. MongoDB, Elastic ...
  - Search engines, e.g. Google Search Appliance, Elastic, iFinder5, Lucidworks Fusion ...

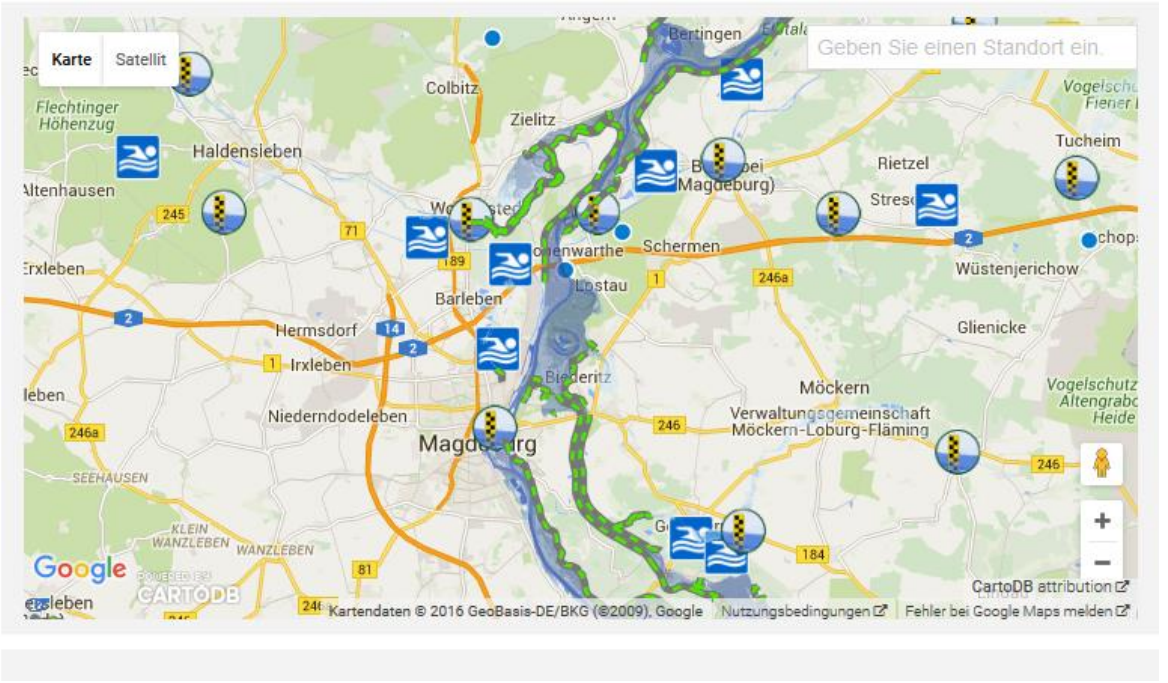
📖 Düpmeier et al.: “A generic microservice architecture for environmental data management” (ISESS 2017, Springer)

📖 Schlachter et al.: “A generic web cache infrastructure for the provision of multifarious environmental data” (ISESS 2017, Springer)

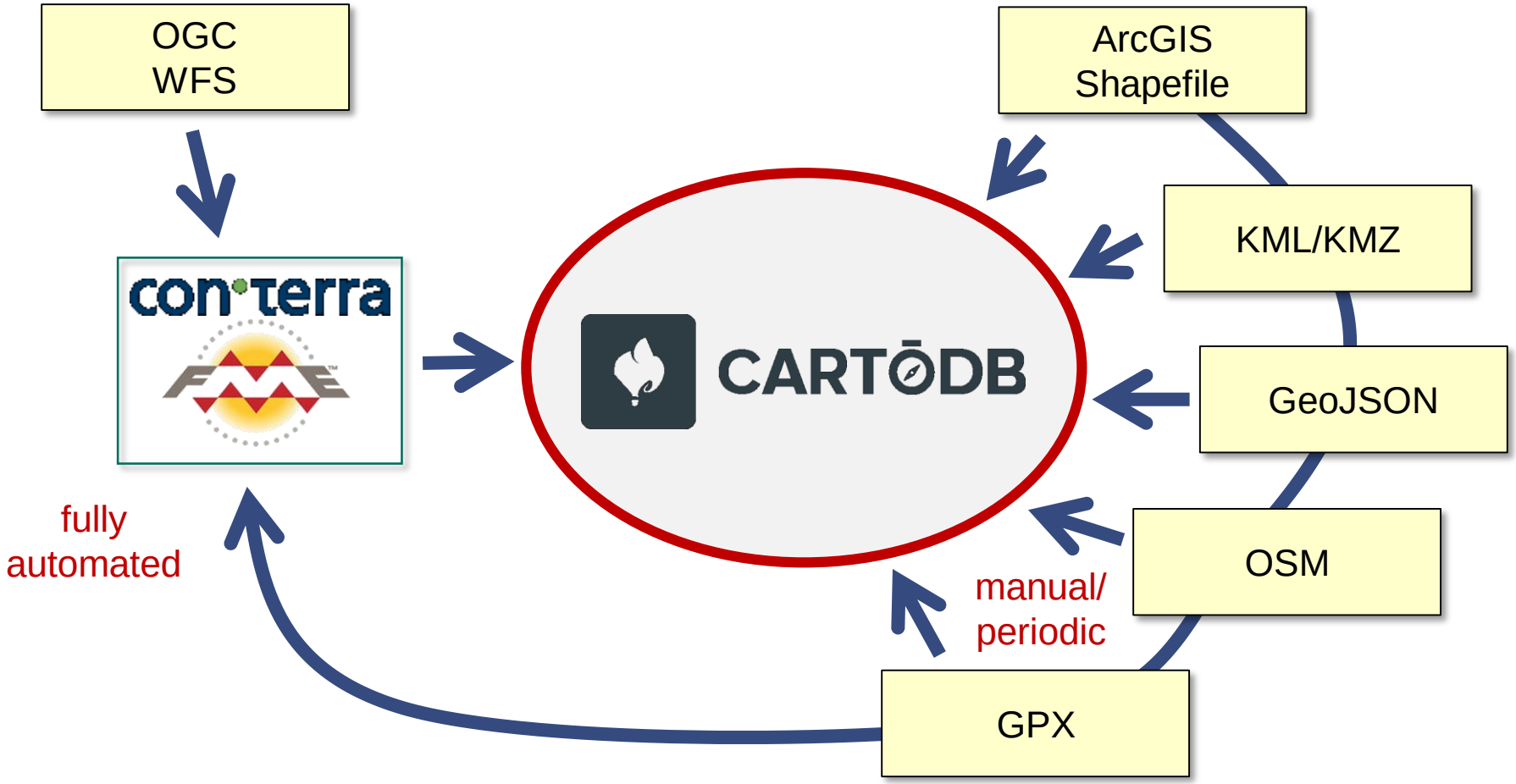
Backend for Spatial Data: Carto/CartoDB

- Cloud hosting, Software as a Service (SaaS)
  - high **performance**
  - **scalable**
  - API, tool support (updates of data, ...)





Data Harvesting and Synchronization



Search Engines

- Full text
  - Google search appliance
    appliance (hard+software)
    to be replaced by:
  - Site Search Pro (iFinder5)
    installation "on premise"

iFinder 5 elastic

elastic

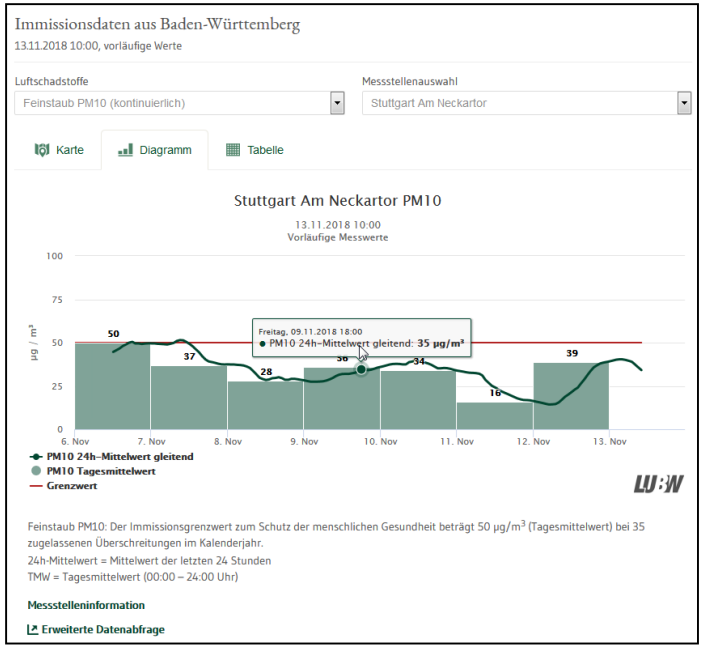
- Structured data / object data
  - Elastic (Search)

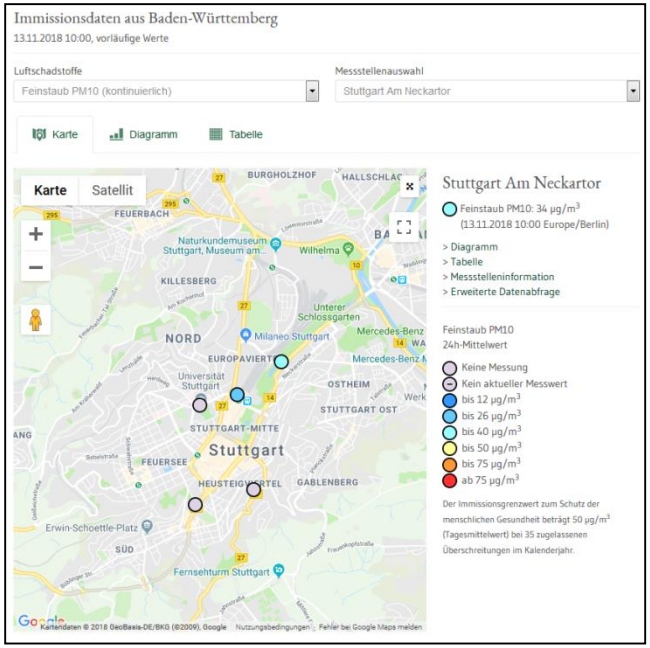
- Search cascades
  - Statistical State Office
  - Service-BW (Citizen services)
  - MetaVer (CSW catalog data harvesting)
    Metadata

Visualization in Web Portals

- Visual presentation by (standardized) front-end modules
  - (HTML5) Web Components
  - Reusable
    - In principle independent of the content management system, however: existing portlet wrapper for Liferay Portal
    - also usable in Web apps and hybrid mobile apps

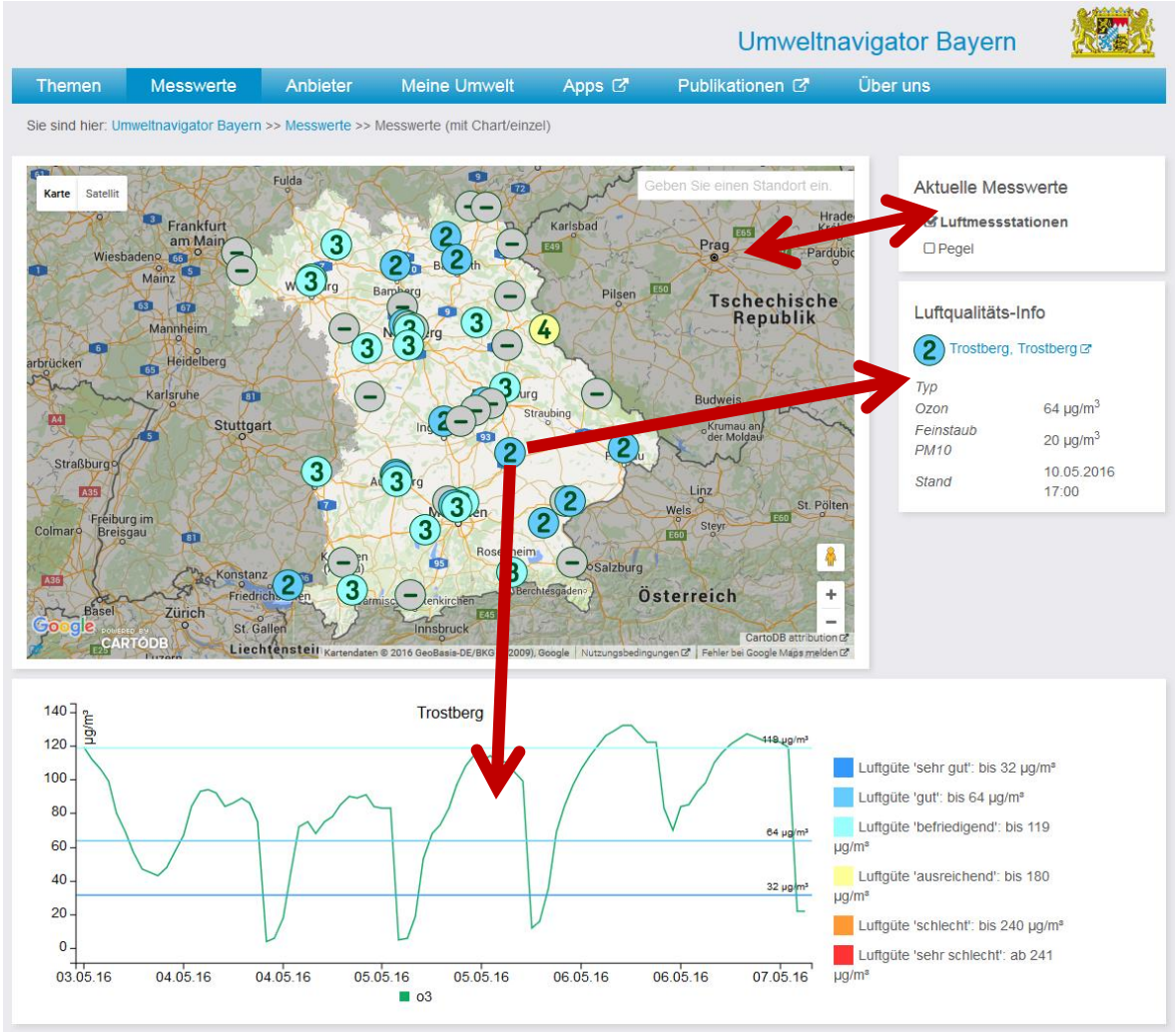
- Examples
  - Diagrams
  - Maps
  - Tables
  - Lists
  - Object details
  - „,,



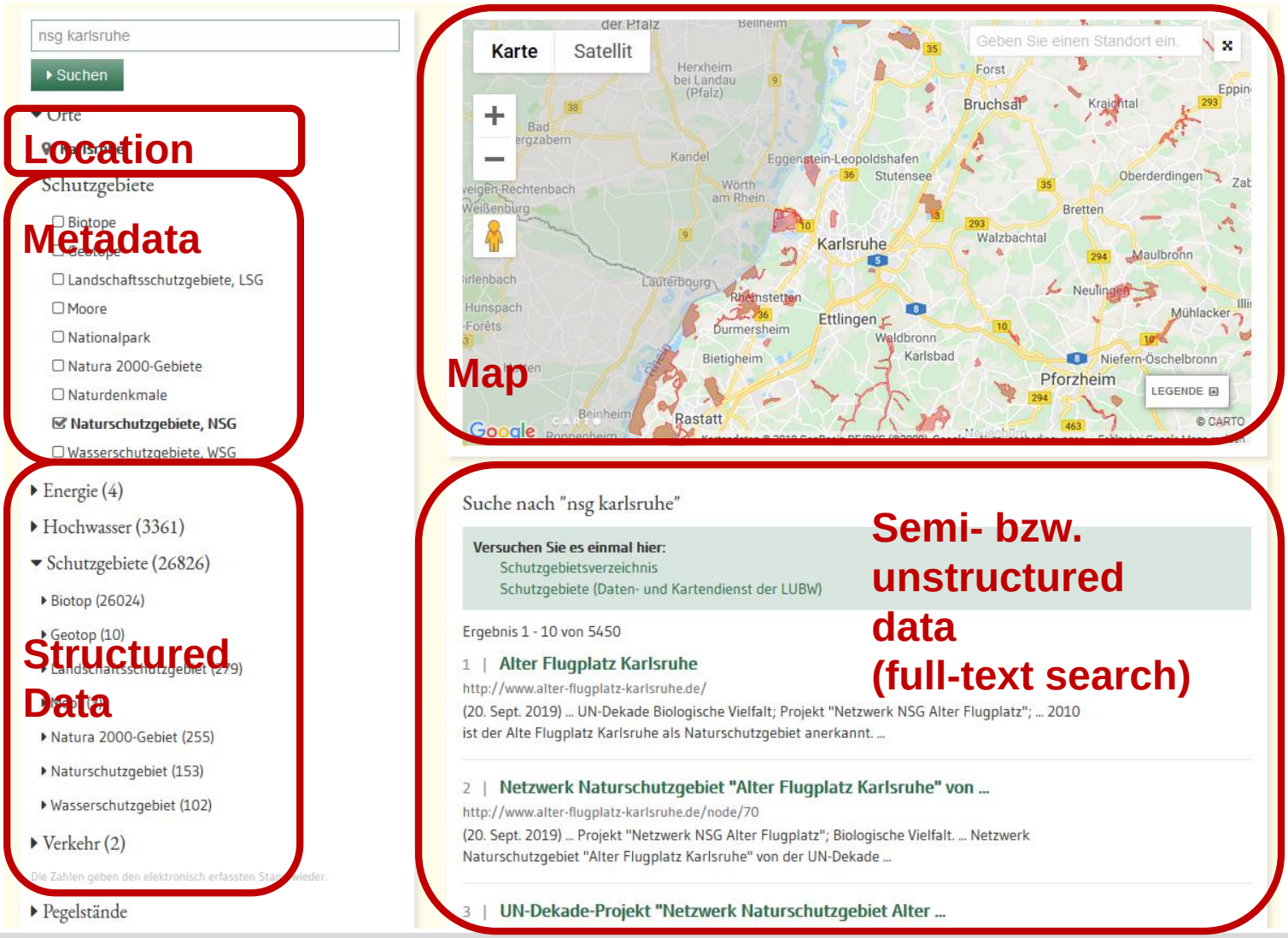


Example: Air Quality Measurements

- Components (selection)
  - Map (showing measuring stations)
  - Layer selection, e.g. Water levels, air quality
  - Detailed information
  - Diagram



Landing Page for Search Requests



Example: Environmental Portal of Baden-Württemberg

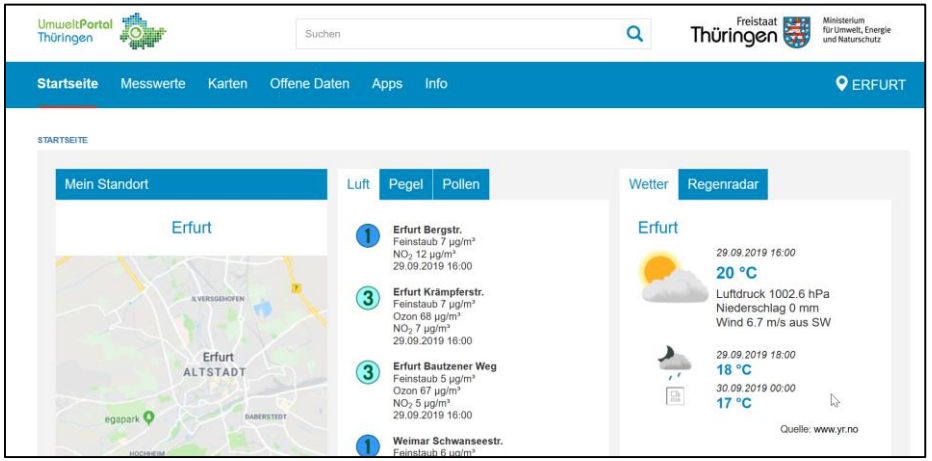
Liferay Portal as Container Application

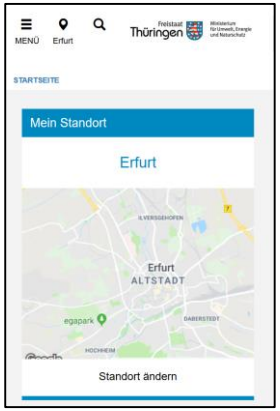
- Community Edition (free of charge)
- Java based

- Editorial interface for
  - Page structure
  - Layout (arrangement of portlets)
  - Wrapper for Web components
  - "Small content“

- Presentation
  - Themes (country specific)
  - Responsive

- Operation in *Docker* containers





References:
https://liferay.dev/-/portal
https://www.docker.com

Microservices

- Each service is an independent application
  - Largely independent from each other
  - Multiple usages possible
- Stable APIs
- Scalable (if necessary)

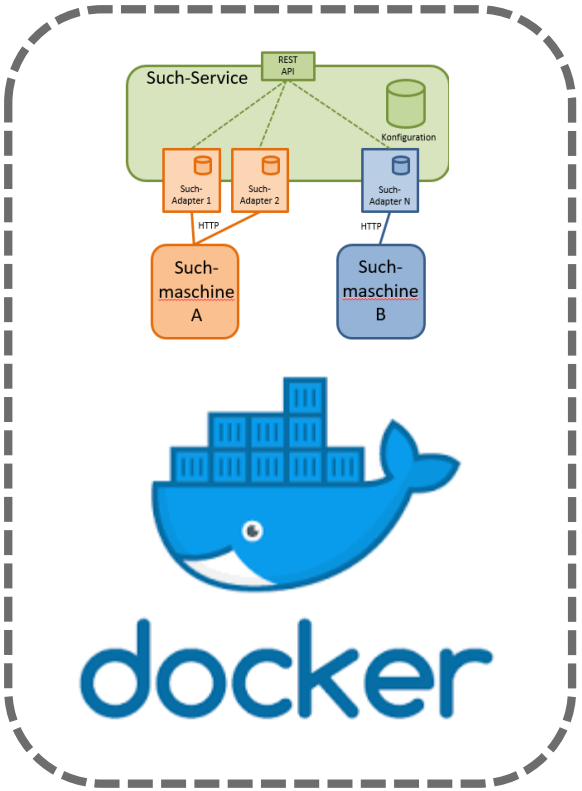
- Manageable code base
- REST description with “*swagger*”
- Java implementation with “*spring boot*”
- Configurable for different applications
  - e.g. State-specific characteristics

References:
https://spring.io/projects/spring-boot
https://swagger.io

Container Virtualization

- Container virtualization with *Docker*
- Each Microservice packed in *Docker* container
- Operation on different infrastructures, e.g. *Kubernetes* infrastructures
  - Server
  - Cluster, data center
  - Cloud
- Easy deployment and creation of new instances
- “Inheritance” of containers simplifies updates

- Also for
  - *Liferay* portals
  - *Liferay* databases



kubernetes

References:
https://www.docker.com
https://kubernetes.io

Licences for Software and Services

- **Google**
  - Maps JavaScript, Geocoding API
  - Static Maps
  - CloudSQL
  - App Engine, Compute Engine

- **Site Search Pro** (iFinder5 elastic full-text searcg)
- (Google Search Appliance - running out)

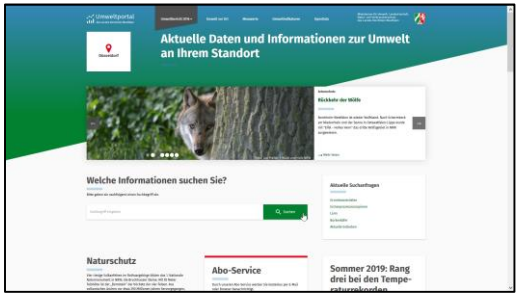
- **Carto** (map server)

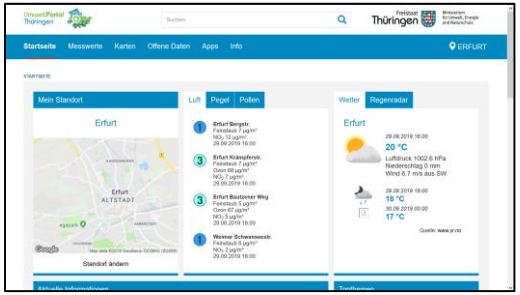
- Jointly licensed by the participating federal states.

References:
https://cloud.google.com
https://www.intrafind.de/index_en
https://carto.com

References and Examples

- Umweltnavigator Bayern
  https://www.umweltnavigator.bayern.de/
- Umweltportal Baden-Württemberg
  https://www.umwelt-bw.de/
- Umweltportal Nordrhein-Westfalen
  https://www.umweltportal.nrw.de/
- Umweltinformationsnetz Sachsen-Anhalt
  https://www.umwelt.sachsen-anhalt.de/
- Umweltportal Thüringen
  https://www.umweltportal.thueringen.de/





HOW TO START

„Normal“ Onboarding for new LUPO Partners

Phase 1: Organizational Planning and General Conditions

- Contracts and formalities
  - Kick-off conversation
  - Definition of goals
  - Requirements definition
  - Presentation of strategy and planned developments

- Organizational framework
  - Contact persons
  - Create accounts
  - Invitation to regular communication (e.g. Web conferences)
  - Training for existing systems

Onboarding (cont.)

Phase 2: Design Phase and Implementation Planning

- Technical concept
  - Definition of the features to be implemented
  - Definition of the necessary additional work on the part of partners
  - Coordination about systems to be connected (import / export)
- Definition of the collaboration (e.g. with KIT, Convotis)
- Scheduling in coordination with all partners
  - Start of development
  - Code freeze
  - Test phases (internal / external)
  - Beta release
  - Production release
- Convotis/KIT commissioned to implement the coordinated requirements

Onboarding (cont.)

Phase 3: Implementation Phase

- Start of Implementation
- Continuous communication of development teams with contact persons and project management
  - Clarification of any questions
  - Communication of possible delays with reasons
  - Blocker
- Internal test phase development
- Provision of staging version for testing by the project group

Onboarding (cont.)

Phase 4: Testing and Release

- Start testing for project group
- Feedback to project management and development teams
  - Technical changes
  - Feature requests
  - Bugs
- Fixes
- Second test phase for project group
- Release by individual partner
  - Preparations for release
  - Beta release
  - Production release

What do we need?

- Overal ideas
- (Functional) Requirements, e.g.
  - Which measuring networks should be integrated?
  - Which spatial data should be found through the portal?
  - Is there any specific content for the portal? Which?
  - Do you already have specific ideas for the frontend application?
    e.g. mockups for specific pages?
- Data sources, e.g.
  - Which data sources and providers are out there? Contact persons!
  - Which interfaces do they provide?
  - Which data formats are used?
  - Are there existing landing pages in specific systems?
  - Are there any restrictions on the use of data?
- Others
  - Are there any legal restrictions?
    (e.g. data protection, copyright, licenses, use of cloud services, etc.)

Possible Topics and Data Sources

- Protected areas
- Nature reserves
- Natura 2000 areas
- Biosphere reserves
- National parks
- Forest areas
- Neophytes
- Land use
- Floodplains
- Flood areas
- Water protection areas
- Bathing water
- Traffic noise
- Nature trails
- Agriculture
- Renewable energy
- Wind turbines
- Wind potential
- Solar potential
- Geothermal energy
- Biomass power plants
- Energy consultants
- Events
- News, Press releases
- Environmental education
- Planning documents
- Construction areas
- Environmental reports
- Research reports …

Suggestion for a Procedure

1. First information (today)
2. Collection of ideas and requirements (few weeks)
3. Determination of possible data stocks
   (including interfaces, formats, restrictions) (few weeks)
4. Workshop (e.g. 03/2020)
   - *Main goal: Definition of a prototype*
   - Assessment of ideas and requirements
   - Evaluation of the existing solution with regard to ideas and requirements, identification of implementation gaps
   - Decision about the implementation of a prototype
   - Specification of prototype features
   - Planning of next steps
   - Basic training (optional)
5. Implementation of a prototype (3-5 months)
   - Agile software development, Ongoing communication
6. Evaluation Workshop (summer/autumn 2020)
   - Decision on if and how to proceed

Critical points to be clarified

Currently we see the following points to be clarified for the successful implementation of a prototype and beyond:

- Participation of the environmental portal for Serbia in the LUPO cooperation, e.g. as affiliated partner
  - Re-use of software and licenses
  - Free capacities of the contractors
- Provision of data by Serbian authorities (generally and in time)
  - Clarification of all possible legal aspects
- Language and localization
  - German only at this point
  - Translation of content and templates
- Design
  - No budget for prototype-styling: Very basic design and styling.

Contact

Dr.-Ing. Thorsten Schlachter

Karlsruhe Institute of Technology (KIT)
Institute for Automation und Applied Informatics (IAI)
Web-based Information Systems (WebIS)

+49 721 608-25769 • thorsten.schlachter@kit.edu