Keynote: 
First Thoughts on a Data Lake Architecture for an Open Search Infrastructure

Leon Martin, M.Sc. 
Prof. Dr. Andreas Henrich 
University of Bamberg 
Media Informatics 

https://www.uni-bamberg.de/minf/
1. Motivation
2. From Data Warehouses to Data Lakes
3. First Thoughts on an Architecture
4. Next steps & Conclusion
Outline

1. Motivation
2. From Data Warehouses to Data Lakes
3. First Thoughts on an Architecture
4. Next steps & Conclusion
An European Open Search Infrastructure

An open and distributed Internet search in Europe bases on an open search ecosystem – The Open Search Web Index

Open Search Web Index Interface / API

Open Search Web Index

Open Search Internet Crawlers gather information and aggregate it in an Web Index (Database)

Information available in the Internet

User

Service 1

Service 2

Service 3

………………

Service n

https://opensearchfoundation.org/
An **Open Search Index**, as a fundamental and indispensable basis for a large variety of public and private information services.

https://opensearchfoundation.org/
An European Open Search Infrastructure
An European Open Search Infrastructure

- Technological and computational aspects
  - Distributed crawling, indexing, and search
  - Distributed storage of Big Data
  - Security
  - ...

- Societal aspects
  - Right to be forgotten
  - Transparency
  - Access management
  - Fake news detection
  - ...
  - ...

We need a robust architecture!
Benefits of a good architecture:
- Standardized schemata
- Clear interfaces / APIs
- Well defined functional blocks

→ Will reduce risk
→ Will attract various players to contribute
→ Will allow for adaptation and specialization in a generic frame
→ Will foster the Open Search idea
Problem:

Architecture is costly to change later \iff \text{Big-Design-Up-Front is not feasible at this scale and complexity.}

Solution:

- Design an extensible architecture as a starting point that covers key aspects.
1. Motivation

2. From Data Warehouses to Data Lakes

3. First Thoughts on an Architecture

4. Next steps & Conclusion
Open search infrastructure will store heterogenous data ranging from deeply structured to totally unstructured.

The infrastructure has to handle Big Data!
Handling Data using Data Warehouses

- For data analysis
- Data is only added and read
- Data is never updated or deleted
- Use of rigid data models and schemata tailored to specific data mining purpose

Problems:
- Big Data does not fit into predefined data models and schemata
- We have no specific data mining purpose
Handling Data using Data Lakes

- Regarding data **storage**
  - Data is **only added and read**
  - Data is never updated or deleted
  - Data is stored in their **raw format**
  - **Metadata** keeps track of new versions of data

- Regarding data **governance**
  - **Clear-cut componentization** and responsibilities
  - Proper use of metadata is mandatory to avoid **data swamps**
  - A **catalog** takes inventory and stores **management routines**
Handling Data using Data Lakes

- Regarding data **interaction** and **maturation**
  - **Raw data interaction** and view-based interaction
  - Data matures through user interaction

Adopted from [6]; in our case enterprise means the open search infrastructure
Handling Data using Data Lakes

Key benefits:

- **Schema-on-read** defers data modelling and schema definition
- **Data provenance** always comprehensible
- **High level of data accessibility**
- **Immediate access to original raw data**
- **Use case agnostic data management system**
Outline

1. Motivation
2. From Data Warehouses to Data Lakes
3. First Thoughts on an Architecture
4. Next steps & Conclusion
Data Reservoir Overview

Adopted from [3]; the summary of a data reservoir, a data lake architecture by IBM.

No Big-Design-Up-Front!
A Data-Lake-based Search Engine

- Minimum number of components
- Apache Hadoop, Apache Nutch, and Apache Solr as candidate technologies
A Data-Lake-based Search Engine

Note the clear-cut componentization and responsibilities!
Simplified example of a management routine:

- **Crawled Documents**
- **Document Data Store**

- Information Governance monitors operations
- New documents with according description arrive
- Routine for creating/updating the Document Data Store is retrieved from catalog
- Refinement & Annotation issues the execution of the routine
Outline

1. Motivation
2. From Data Warehouses to Data Lakes
3. First Thoughts on an Architecture
4. Next steps & Conclusion
Towards a Data-lake-based Search Infrastructure

We need

- a conceptual model for all metadata
- more (generic) views for accessing the data for other purposes besides search platform (e.g. statistics dashboards)
- more routines for automated refinement (e.g. NLP pipeline)
- statistics, security, …

Again, build a basic extensible architecture first!
Conclusion

- Open search infrastructure → Big Data
- Data lakes could be used as the basis for the envisaged open search infrastructure
- No Big-Design-Up-Front
- Start with a basic extensible architecture, e.g., for a data-lake-based search engine

The points discussed are by no means complete or settled. Instead, our goal is to fuel further discussions.
References

Thank you!
Backup Slides
Data Lake Layers and Consumption Patterns

- Enterprise Data Lake Architecture: What to Consider When Designing
  [Cloud Technology Partners, Sudi Bhattacharya, Neal Matthews
Data Lake Layers and Consumption Patterns

Enterprise Data Lake Architecture: What to Consider When Designing
[Cloud Technology Partners, Sudi Bhattacharya, Neal Matthews

Diagram:
- Raw Data
- Data Lake
- Processed, Standardized, Use Case Specific Data

L. Martin, A. Henrich: First Thoughts on Data Lake Architecture for an Open Search Infrastructure – OSSYM 2020
Data Lake Layers and Consumption Patterns

- Enterprise Data Lake Architecture: What to Consider When Designing
  [Cloud Technology Partners, Sudi Bhattacharya, Neal Matthews

Consumption Pattern

- Machine Learning
- Ad-hoc Analysis
- Reports Dashboard
- Enterprise Search
- Interactive Fast Queries
- Processed, Standardized, Use Case Specific Data
- Raw Data

Data Lake
Data Lake Template for Reference Architecture

Source Systems:
- File Data
- Database Data
- ETL Extracts
- Streaming
- APIs

Storage Solutions:
- CLTP
- Data Warehouse
- Logs
- Cloud

Consumption Zone:
- Business Analyst

Data Lake:
- Loading Zone
- Raw Data
- Refined Data
- Tokenized Data
- Reference Data
- Master Data
- Sandbox
- Metadata
- Security
- Data Quality
- Data Catalog

Links:
- https://buildingvts.com/elasticsearch
- https://www.dragon1.com/demo/data-lake
Key Benefits Of Data Lake

1. Scalability
   - storage from disparate sources like multimedia, binary, XML; …

2. High-velocity Data
   - data stream processing and large volumes of historical data

3. Structure
   - unique arena where structure like metadata, speech tagging etc. can be applied on varied datasets

4. Storage
   - iterative and immediate access to the raw data

5. Schema
   - schemaless write and schema-based read

Factors to consider:

- Data Governance and Security Layer
- Metadata Layer
- Information Lifecycle Management Layer

Tiers to manage data flows:

- Intake Tier
- Management Tier
- Consumption Tier

What is needed according to the CAP theorem?

- Consistency
- Availability
- Partition tolerance
