metaphacts

Smart Data. Smart Apps. Smart Decisions.

Graph Database Applications

Mastering the Heterogeneity Challenges

Michael Schmidt

DBOnto Workshop, May 27, 2016 (London)

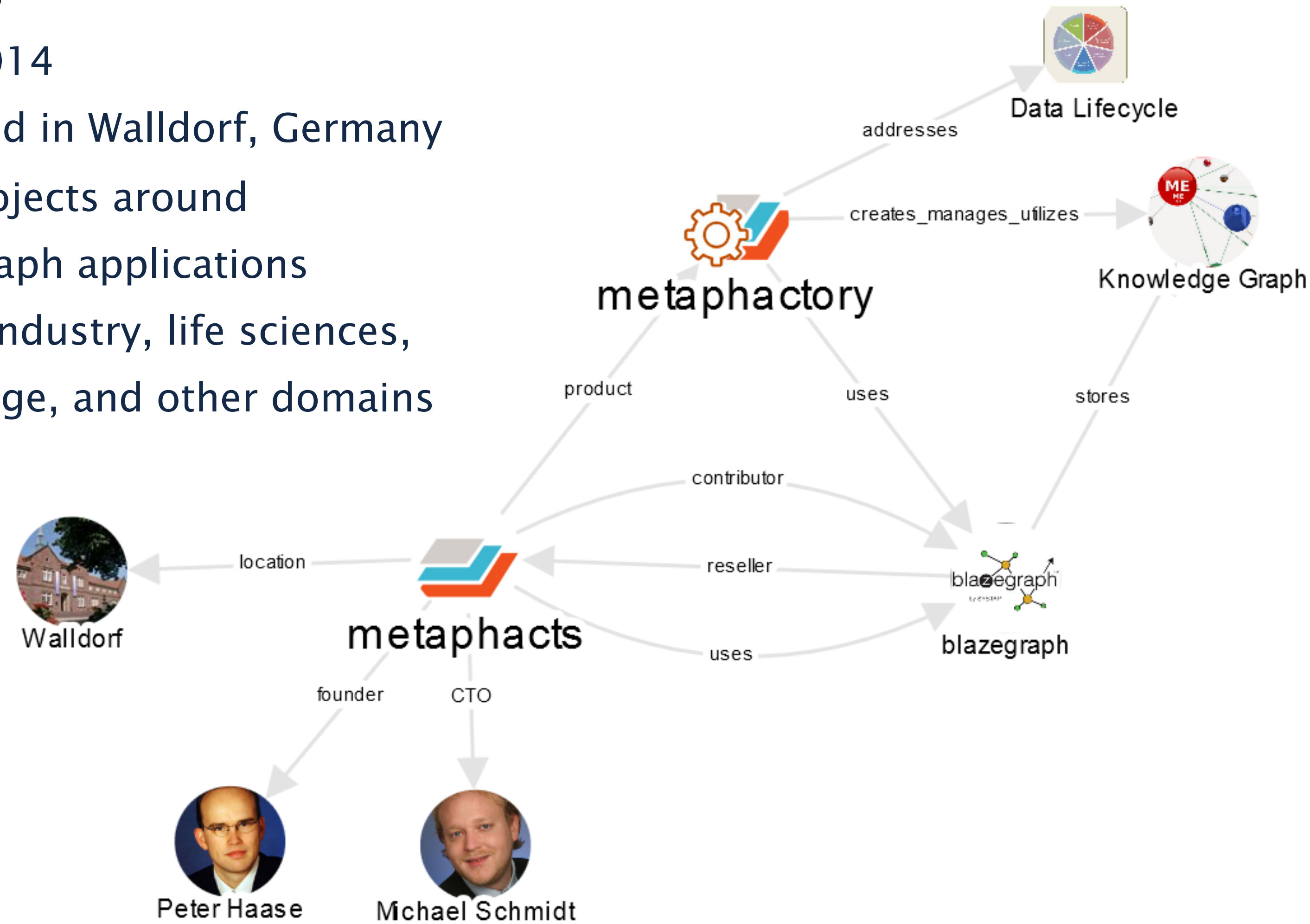
metaphacts at a Glance

metaphacts

Smart Data. Smart Apps. Smart Decisions.

Company Facts

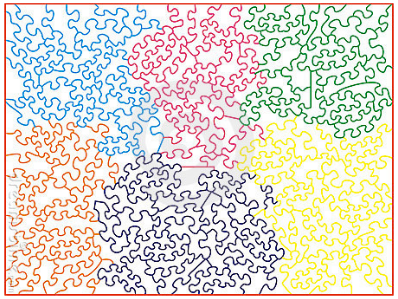
- Founded in 2014
- Headquarterted in Walldorf, Germany
- Software & projects around knowledge graph applications
- Solutions for industry, life sciences, cultural heritage, and other domains



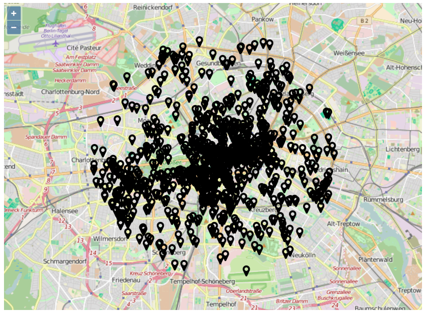
Challenges in Knowledge Graph Application Building

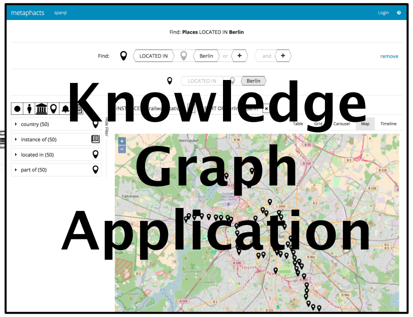
Smart Data. Smart Apps. Smart Decisions.

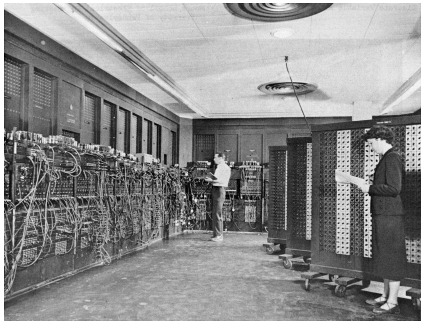
Schema heterogeneity & alignment problems



Different data modalities (geospatial, temporal, …)







Data residing in specialized & legacy systems

Structured Queries vs. (Graph) Analytics

The metaphacts Approach

metaphacts
Smart Data. Smart Apps. Smart Decisions.

Platform for Knowledge Graph Application Development

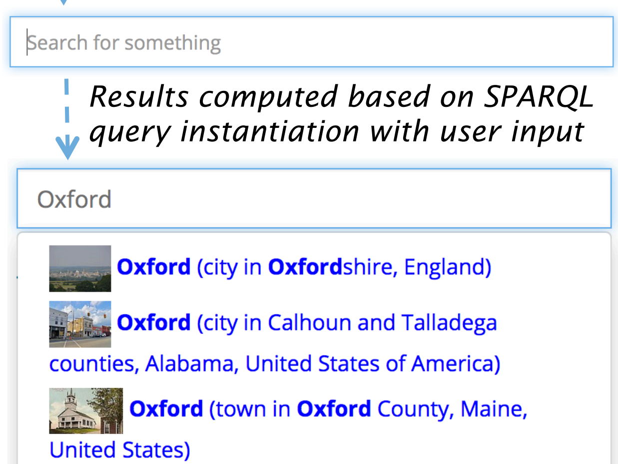
- **RDF, RDFS & OWL** for knowledge representation
  - Graph-based -> easing integration
  - Built-in semantics
- Low-level and higher-level APIs: **SPARQL, LDP, REST, ...**
  - Choice depends on use case and requirements
- **Declarative application development approach**
  - HTML5 based, reusable (and mostly domain independent) semantic Web components
  - Generic, composable & standards-based

```
<semantic-simple-search data-config='{
 "query":"
   SELECT ?result ?label ?desc ?img WHERE {
     ?result rdfs:label ?label .
     ?result rdfs:comment ?desc .
     ?result foaf:thumbnail ?img .
     FILTER(CONTAINS(?label, ?token))
 }",
 "searchTermVariable":"token", // user input
 "template":"
   <span title="{{result.value}}">
     <img src="{{img.value}}" height="30"/>
     {{label.value}} ({{desc.value}})</span>"
}'/>
```

Rendered component

Ex.: declarative spec. of keyword search field driven by SPARQL

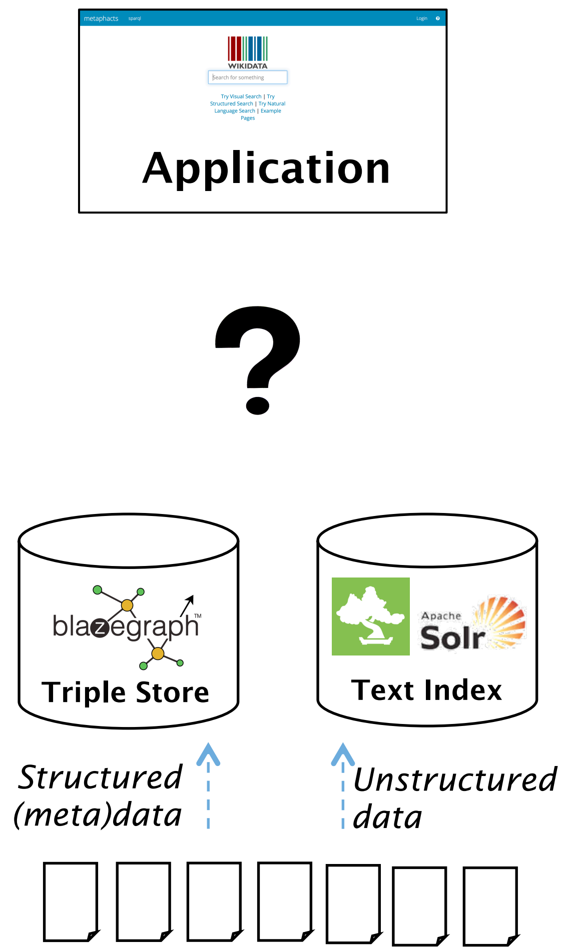
Results computed based on SPARQL query instantiation with user input



Hybrid Query Scenario Challenge

metaphacts
Smart Data. Smart Apps. Smart Decisions.

- **Challenge: supporting hybrid search**
  - Combine free-text search with structured data extraction in SPARQL endpoint
  - Reuse existing systems
    - Non-invasive approach
    - Specialized tools (e.g. for text search) often benefit from years of development & experience
- **Goal**
  - No proprietary, coded solution
  - Still have it declarative

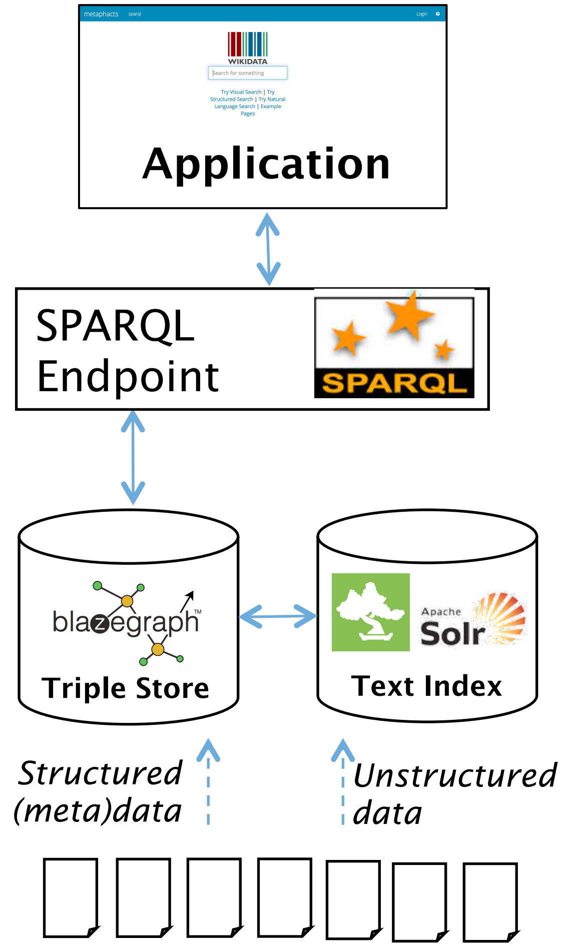


Custom SPARQL SERVICE Extensions

- Key idea: custom **SPARQL SERVICE extensions**
  - Standards-compliant syntax & clear semantics
  - Elegant & easy to understand
  - Extensible

```
SELECT ?res ?type ?author WHERE {
  SERVICE fts:search
  {
    ?res fts:search "London | Queen" .
    ?res fts:endpoint "http://my.solr/select" .
    ?res fts:params "fl=uri,score" .
    ?res fts:scoreField "score" .
    ?res fts:score ?score .
  }
  ?res rdf:type ?type .
  ?res :hasAuthor ?author .
} ORDER BY DESC(?score)
```

Example: returned entities including author & type containing the search terms “London” or “Queen”, ordered by Solr score



Graph Analytics vs. Querying

metaphacts
Smart Data. Smart Apps. Smart Decisions.

- Approach: **unified, GPU based runtime**
  - Data graph loaded into the GPU at startup
  - Runtime provides highly efficient algebraic core operators
    1. Used to accelerate SPARQL query evaluation
    2. Used to execute graph algorithms (e.g. BFS, SSSP, PageRank, ...)
  - Own algorithms can be specified using a domain-specific functional language
    - Translated into programs over the GPU
  - Algorithms exposed as custom SPARQL SERVICE extensions

Blazegraph GPU bridges the gap between declarative SPARQL queries and functional graph analytics programs.

blazegraph™

```
SERVICE gas:service {
   gas:program gas:gasClass "com.bigdata.rdf.graph.analytics.BFS" .
   gas:program gas:in <ip:/112.174.24.90> . # one or more times, specifies the initial frontier.
   gas:program gas:out ?out . # exactly once - will be bound to the visited vertices.
   gas:program gas:out1 ?depth . # exactly once - will be bound to the depth of the visited vertices.
   gas:program gas:out2 ?predecessor . # exactly once - will be bound to the predecessor.
   gas:program gas:maxIterations 4 . # optional limit on breadth first expansion.
   gas:program gas:maxVisited 2000 . # optional limit on the #of visited vertices.
}
```

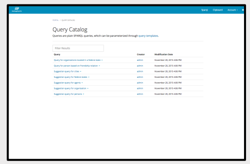
Example: invoking breadth-first search via custom SPARQL SERVICE extension

metaphacts Reference Architecture

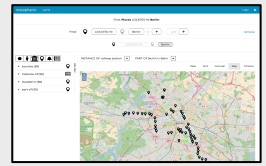
metaphacts
Smart Data. Smart Apps. Smart Decisions.

Data-driven services

Knowledge Graph management UIs (for experts)



Knowledge graph applications (for end users)



Semantic data connectors for external tools

Open for Innovation KNIME

R

SPSS

Unified API Stack

REST APIs (domain specific)

Generic higher-level APIs (e.g. LDP)

Base API: SPARQL Endpoint (implicitly incl. SERVICE extensions)

SPARQL

Access via SPARQL SERVICE (federation) or one-time import

OBDA Endpoint

JDBC

MySQL

ORACLE

...

Relational Data

Graph Database

blazegraph™

- RDF triple store
- Unified management of schema and instance-level data
- GPU-based, unified runtime for querying and graph analytics
- Extensions for geospatial & temporal data
- Open Source & extensible

Graph Data Processing (Querying & Analytics)

Access via custom SPARQL SERVICE extensions

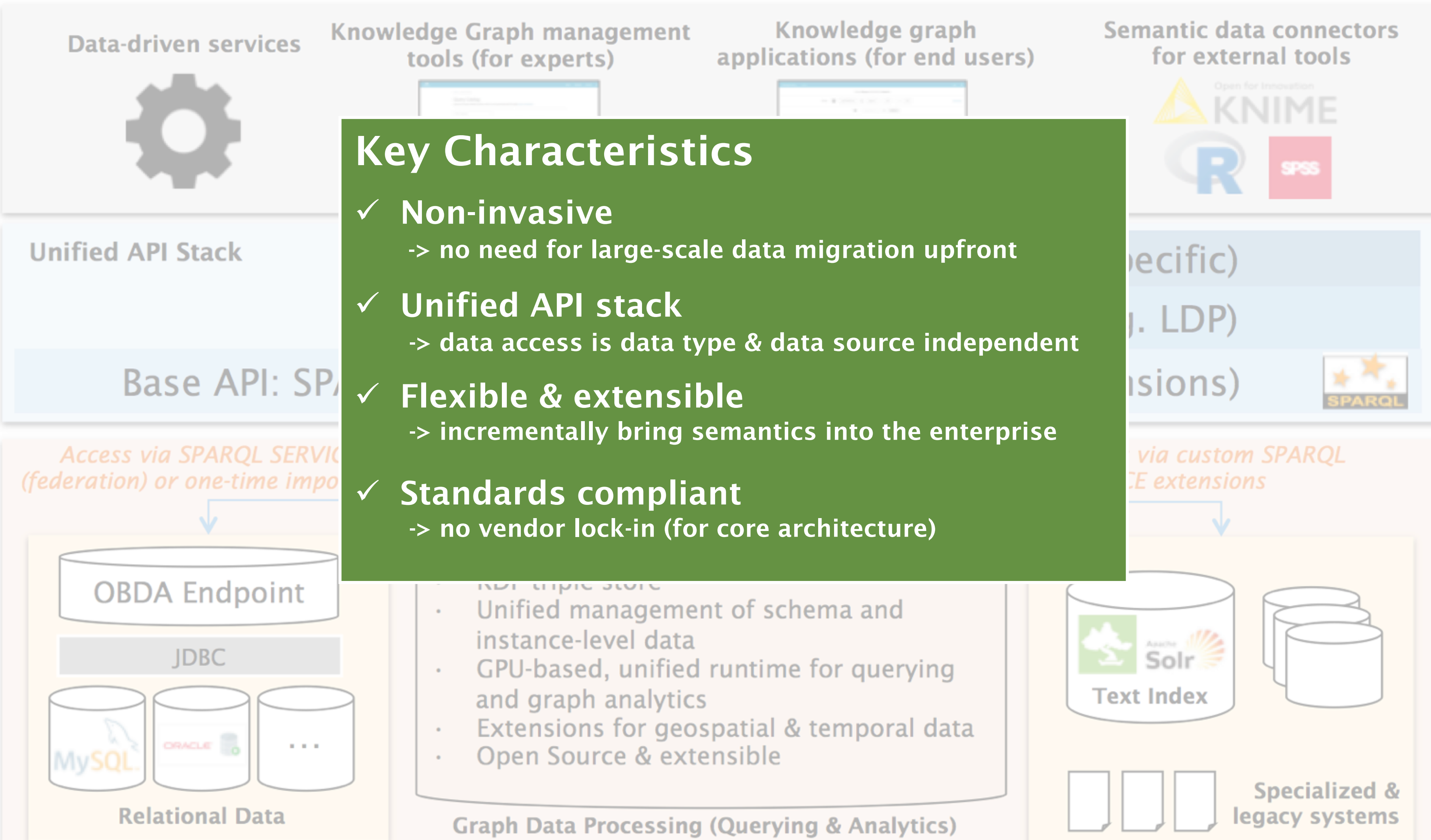
Apache Solr

Text Index

Specialized & legacy systems

metaphacts Reference Architecture

metaphacts
Smart Data. Smart Apps. Smart Decisions.



Key Characteristics

✓ **Non-invasive**
-> no need for large-scale data migration upfront

✓ **Unified API stack**
-> data access is data type & data source independent

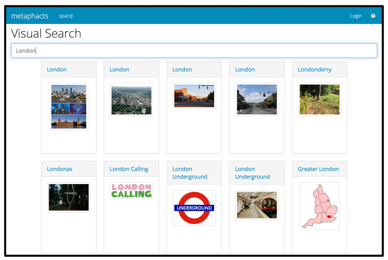
✓ **Flexible & extensible**
-> incrementally bring semantics into the enterprise

✓ **Standards compliant**
-> no vendor lock-in (for core architecture)

Research @ metaphacts

metaphacts
Smart Data. Smart Apps. Smart Decisions.

DIESEL eurostars



What:

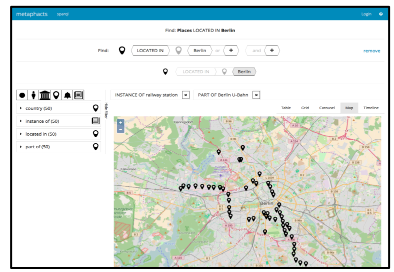
Design a scalable, federated semantic enterprise search system over distributed, heterogeneous data sources.

Key challenges:

- Integration of specialized and legacy systems
- Efficient federated query evaluation
- Design & implement generic APIs for search

GEISER

Federal Ministry
for Economic Affairs
and Energy



What:

Build an open, service-based platform for management and efficient processing of sensor based geo data.

Key challenges:

- Scalable backend services for the storage, retrieval, and processing of semantic geo data
- Flexible, micro-service based architecture

Looking for partners to address the challenges ahead!

Data level

- Efficiency and query optimization
- GPU acceleration & analytics

Architectural level

- Integration with Big Data frameworks (SPARK, …)

Application level

- Supporting management of semantic assets (queries, ontologies, mappings, …)
- Abstraction layers & APIs