ontotext

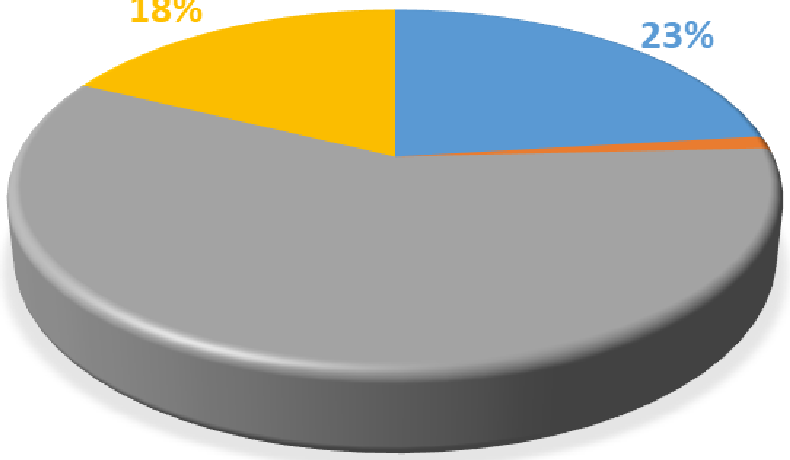
making sense of text and data

KG Powered Digital Twins

Digital Twin Conference Feb, 2024

Ontotext: Visionary Leader with Great Technology

- **Innovator & Leader**
  - ✓ Semantic technology pioneer est. year 2000
  - ✓ The developer of GraphDB™
  - ✓ Center of an ecosystem of 60+ partners
  - ✓ Member of W3C, ODI, STI, EDMC, LDBC, …

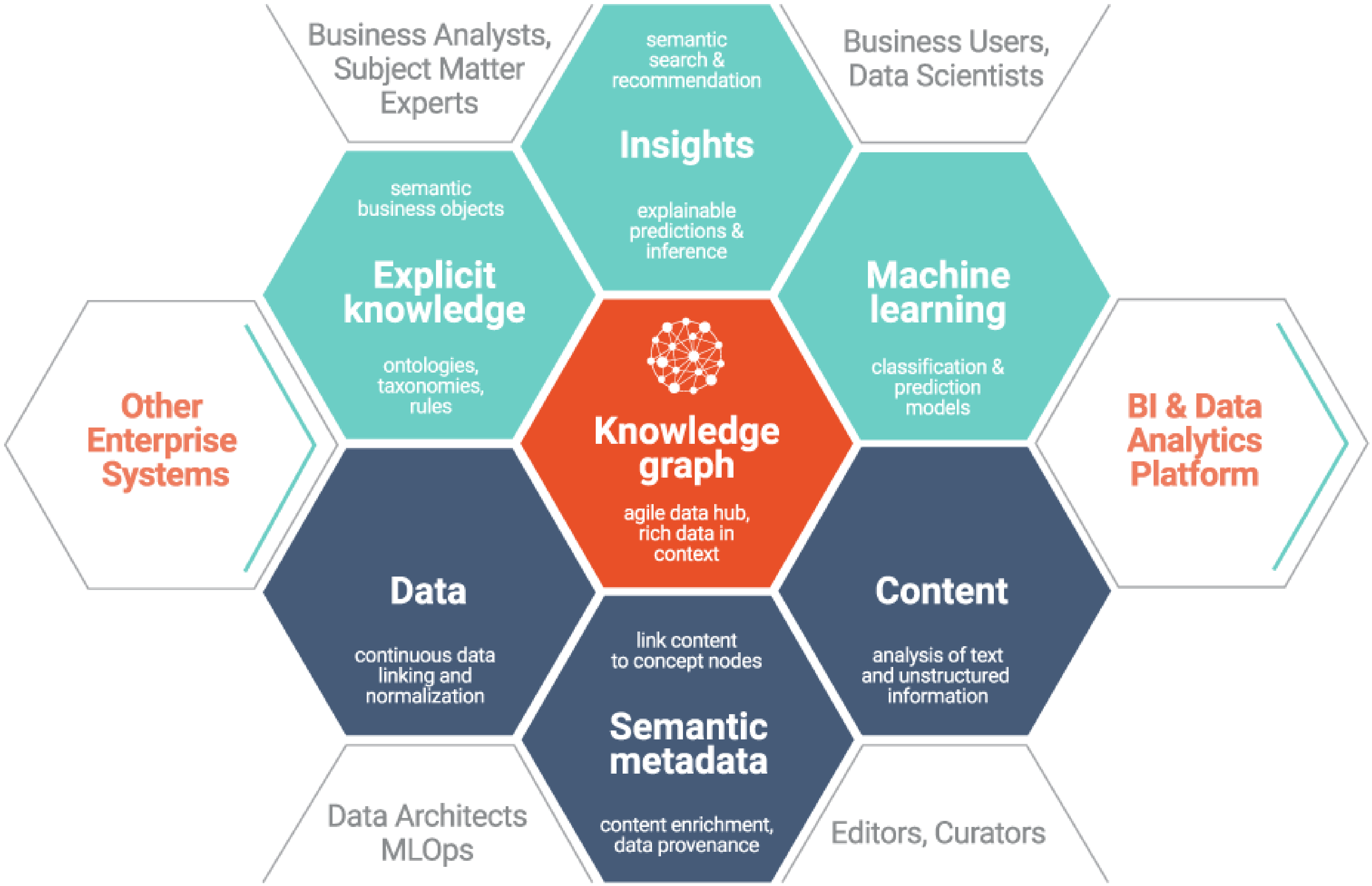


- **Profitable and growing product company**
  - ✓ **ARR** growth 45%**, EBITDA 37%,** over **80% product** revenues
  - ✓ **Global:** 80% of revenue from multinational companies

Agenda

- ↗ The Promise of KG
- ↗ The Fulfillment
- ↗ The Materialization

Value Proposition of Knowledge Graph



Wherefore Digital Twins in a Financial Institution (FI)

A digital twin is a virtual representation of an object or system that spans its lifecycle, uses real-time data, knowledge representation, simulation, machine learning and reasoning to assist just-in-time decision making.

- **Knowledge Representation**
- **Meaning Unification**
- **Data Integration**
- **Scenario Simulation**
- **Analytical Foundation**
- **AI Acceleration**
- **Insights Generation**

Semantic Standards (Essential Building Blocks)

- IDENTITY (IRI/URL)
- ONTOLOGIES (Precise Meaning)
- EXPRESSION (RDF, SKOS, OWL)
- BUSINESS RULES (SHACL, SBVR)

Foundational Capabilities

- QUALITY (mathematical validation)
- CONCEPT REUSE (shared meaning)
- CONTEXT (identity + meaning + time + source)
- ENTITLEMENT (datapoint access control)
- LINEAGE (technical and business)
- GOVERNANCE (simplified and embedded)
- EXECUTION (machine-readable)
- CONTINUOUS TESTING (rules-based)

Value Drivers (why)

- Cost Savings (automation, operational resiliency)
- Control Environment (compliance, simplification of governance, reporting)
- Capability Enhancement (flexibility, data science, data quality)
- Business Goals (customer service, product innovation, customer 360)

Use Cases (what)

- Data Integration (lock down meaning, resolve identity)
- Connected Inventory (digitization, system interoperability, supply chain)
- Access Control (privacy, cyber security, entitlements)
- Data Discovery (enterprise search, ad hoc query, risk management)

Knowledge Representation and Meaning Unification

Search

Enter search term here

Filters Clear All

- Resource Type
- Industry
- Topic
- Product
- Application Area
- Audience
- Capability
- Publication Date
- Author

Marketing Resources | Chat | Analytics

Ask Ontotext Chat

Question: *

Using knowledge graphs for domain knowledge representation and meaning unification for digital twins

GPT Answer

Ontotext Knowledge Graph Answer

Knowledge graphs serve as a powerful tool for representing domain knowledge and unifying meaning, particularly in the context of digital twins. By leveraging semantic modeling techniques such as ontologies and controlled vocabularies, knowledge graphs ensure consistent representation of data, regardless of its original format or source. This unified approach provides a common framework for understanding and working with data, which is crucial for digital twins that require a holistic view of all information and an understanding of the relationships between different pieces of data (read more). Additionally, knowledge graphs can enhance proprietary information by integrating external knowledge sources, providing a richer context for interpretation, which is beneficial for the complex systems that digital twins often represent (read more).

Explore further the following main topics:

How do knowledge graphs contribute to better decision-making and analytics in organizations?

What are some of the industry verticals where knowledge graph applications have been successfully implemented?

ontotext

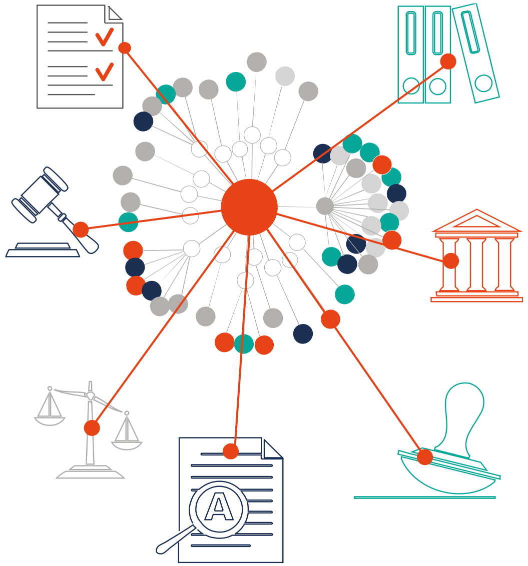
Data Integration Requirements

Structured/Reference Data

- Entities
- Farm and Field
- Property Boundaries
- Land Ownership
- Geographic and Geometric Data
- Legal Information

Time-series

- Historical Soil Moisture
- Historical Precipitation
- Forecasted Precipitation



Knowledge Organization

- Soil Properties and Processes Ontology
- OWL Time Ontology
- OWL GeoSPARQL
- Kinship Ontology
- SWEET – Earth and Environment Ontology
- Financial Industry Business Ontology

Signals from Unstructured content

- TBD

When should each farm field be irrigated to maintain best crop production?

- **Entities:** owners, customers, companies
- **Farm and Field Data:** Information about the specific locations of each farm field and their respective sizes and shapes.
- **Property Boundaries:** Detailed boundaries of each farm field to ensure precise irrigation within the property lines.
- **Geographic and Geometric Data:** GIS data regarding topography and shape of the land
- **Legal Information:** Water rights, usage restrictions, and any legal obligations or constraints regarding irrigation.

- **Soil Properties and Processes Ontology:** Soil type, texture, structure, and porosity
- **OWL Time Ontology:** time-related data, such as planting and harvesting seasons, to align irrigation schedules with critical growth periods of crops.
- **OWL GeoSPARQL:** leverage spatial queries and manipulate geographic information
- **SWEET – Earth and Environment Ontology:** Environmental conditions that affect crop growth and irrigation needs
- **Historical Soil Moisture: Previous soil moisture levels can establish patterns and inform the current moisture status of the soil.**
- **Historical and ForecastedPrecipitation: Historical weather patterns help predict future needs and schedule irrigation to complement natural rainfall.**

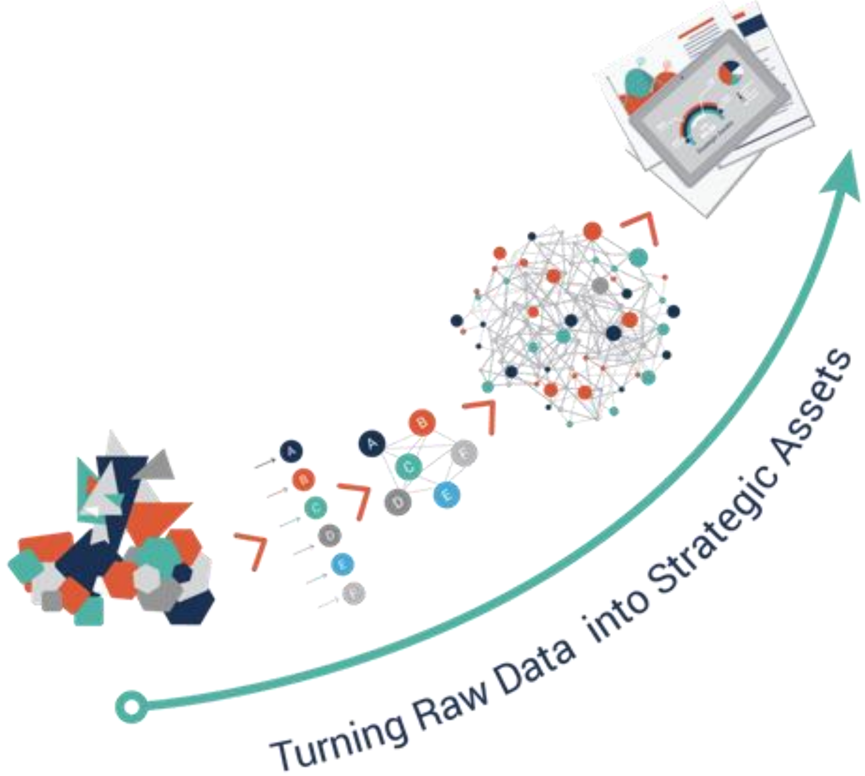
Data Virtualization

Create a single representation of data from multiple, disparate sources without having to copy or move the data

- Abstract different physical model
- Aggregate heterogeneous sources
- Provide a single point of access

Common scenarios:

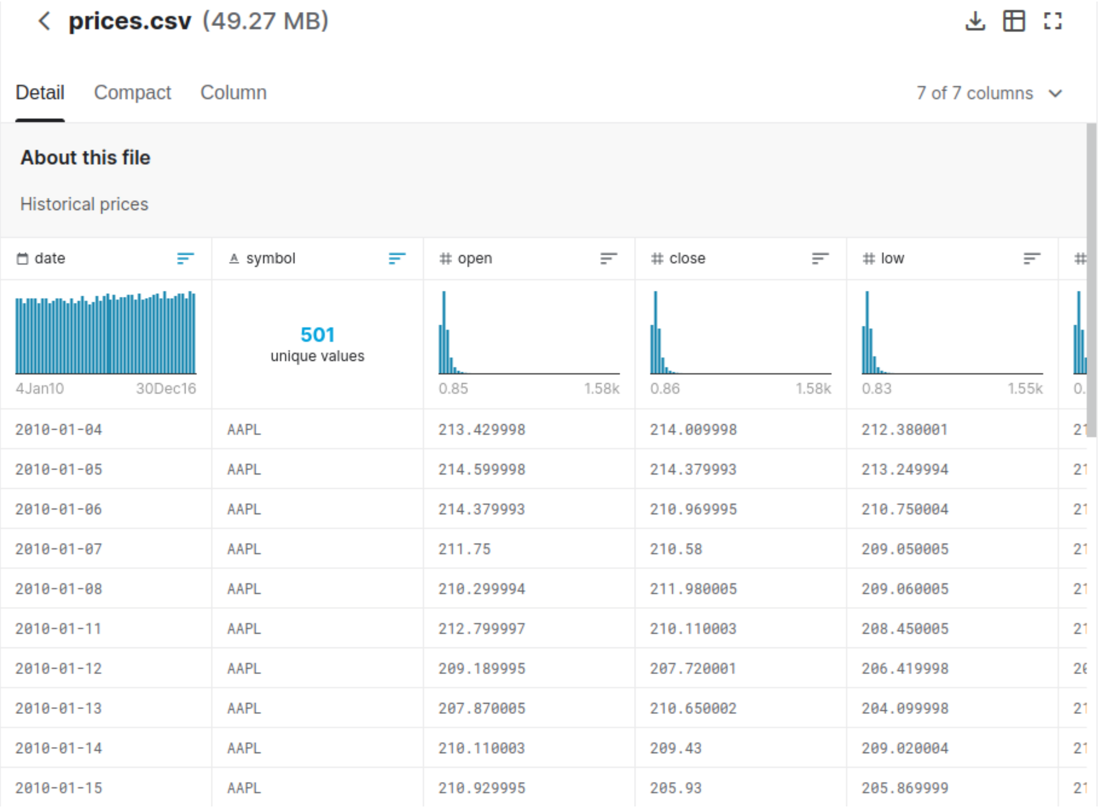
- Access data not available in a single system
- Prototype a BI application
- Data that can't be copied due to security



ontotext

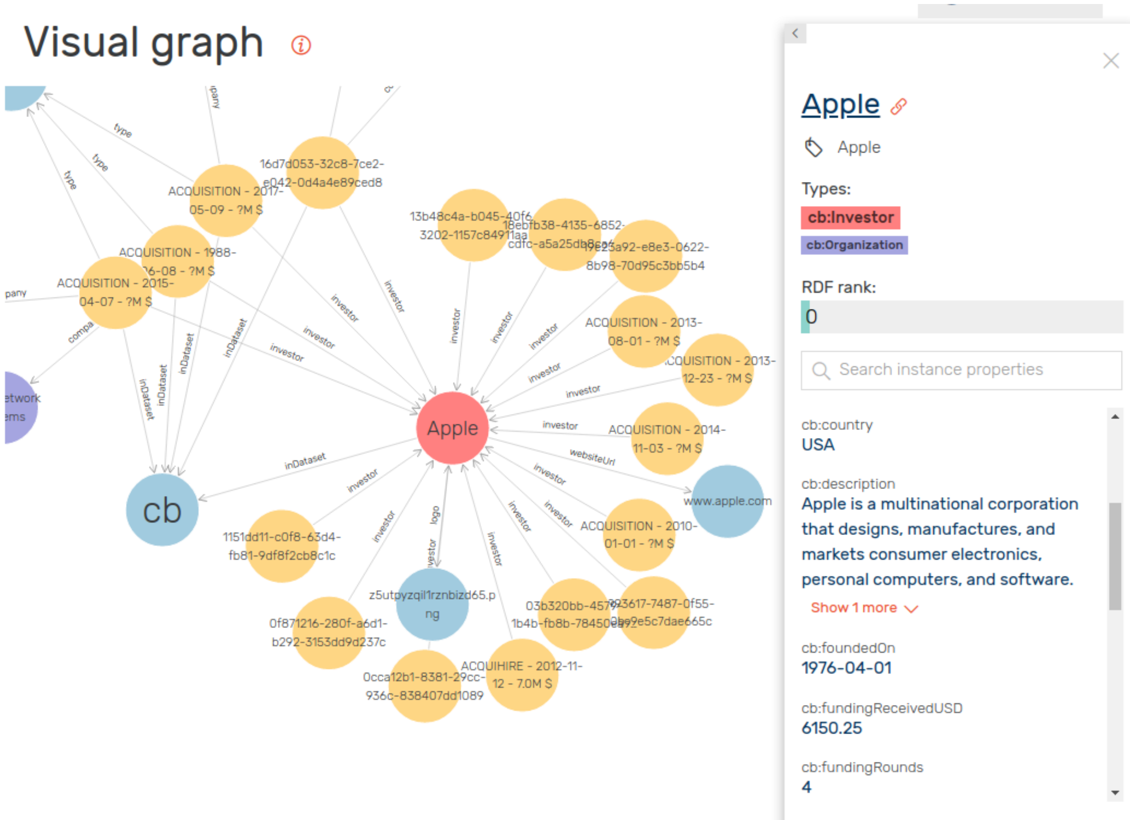
Historical/Forecasted Precipitation at a Location

- Volume and Velocity
- Simple schema no semantics



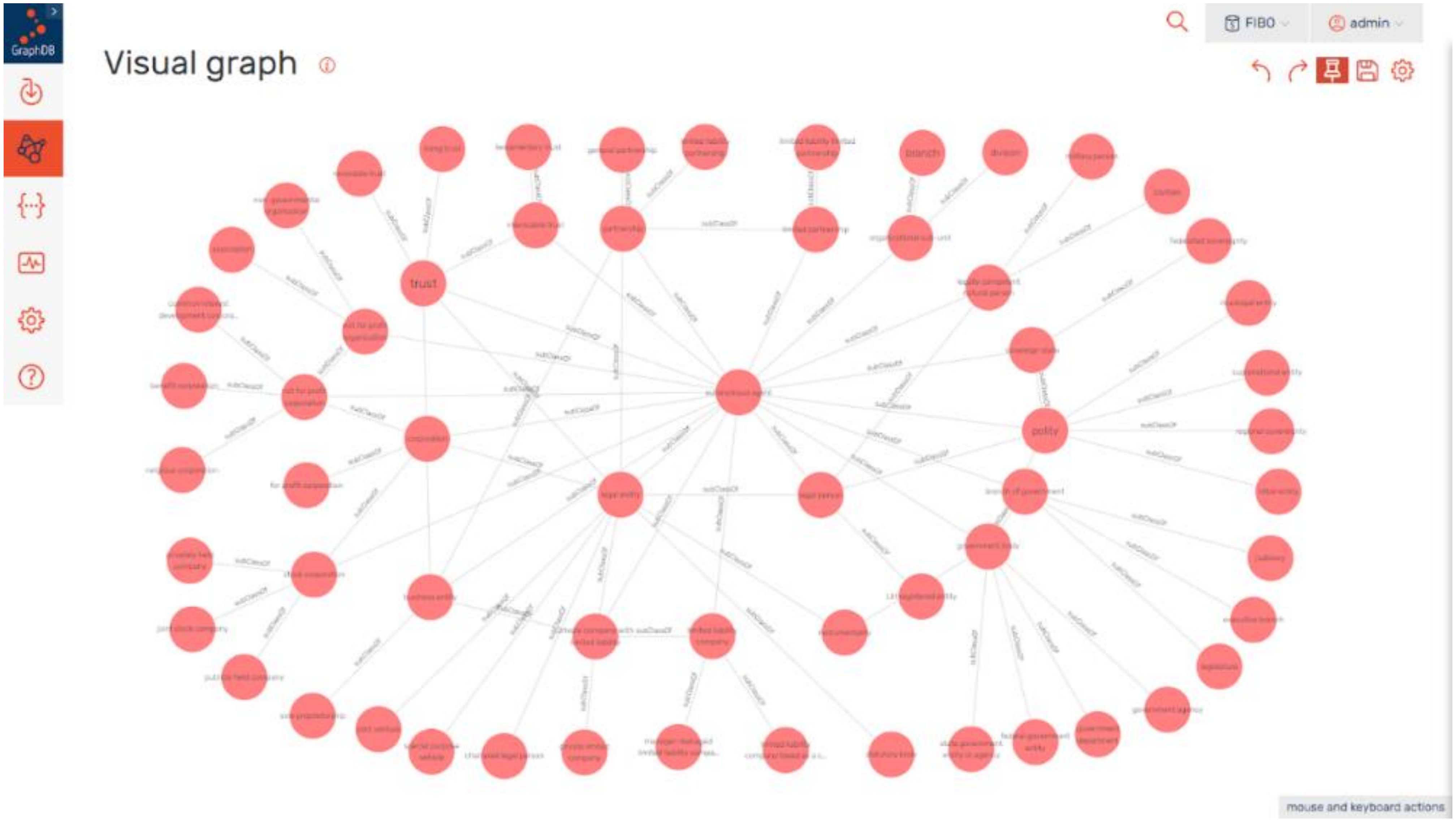
date	symbol	open	close	low	#
2010-01-04	AAPL	213.429998	214.009998	212.380001	21
2010-01-05	AAPL	214.599998	214.379993	213.249994	21
2010-01-06	AAPL	214.379993	210.969995	210.750004	21
2010-01-07	AAPL	211.75	210.58	209.050005	21
2010-01-08	AAPL	210.299994	211.980005	209.060005	21
2010-01-11	AAPL	212.799997	210.110003	208.450005	21
2010-01-12	AAPL	209.189995	207.720001	206.419998	2(
2010-01-13	AAPL	207.870005	210.650002	204.099998	21
2010-01-14	AAPL	210.110003	209.43	209.020004	21
2010-01-15	AAPL	210.929995	205.93	205.869999	21

- Public and alternative data
- Complex graph with semantics



ontotext

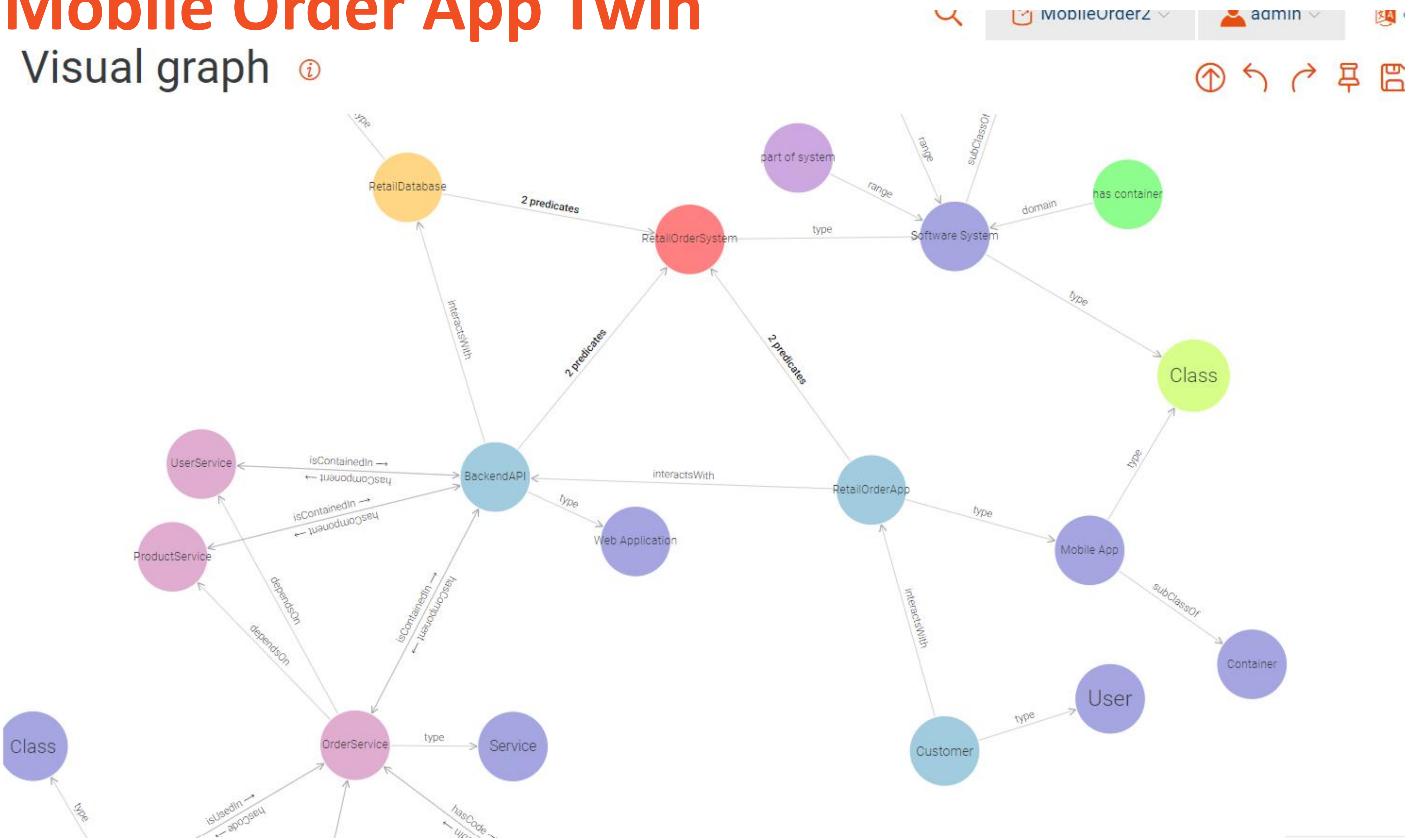
Who is likely to inherit the farm?



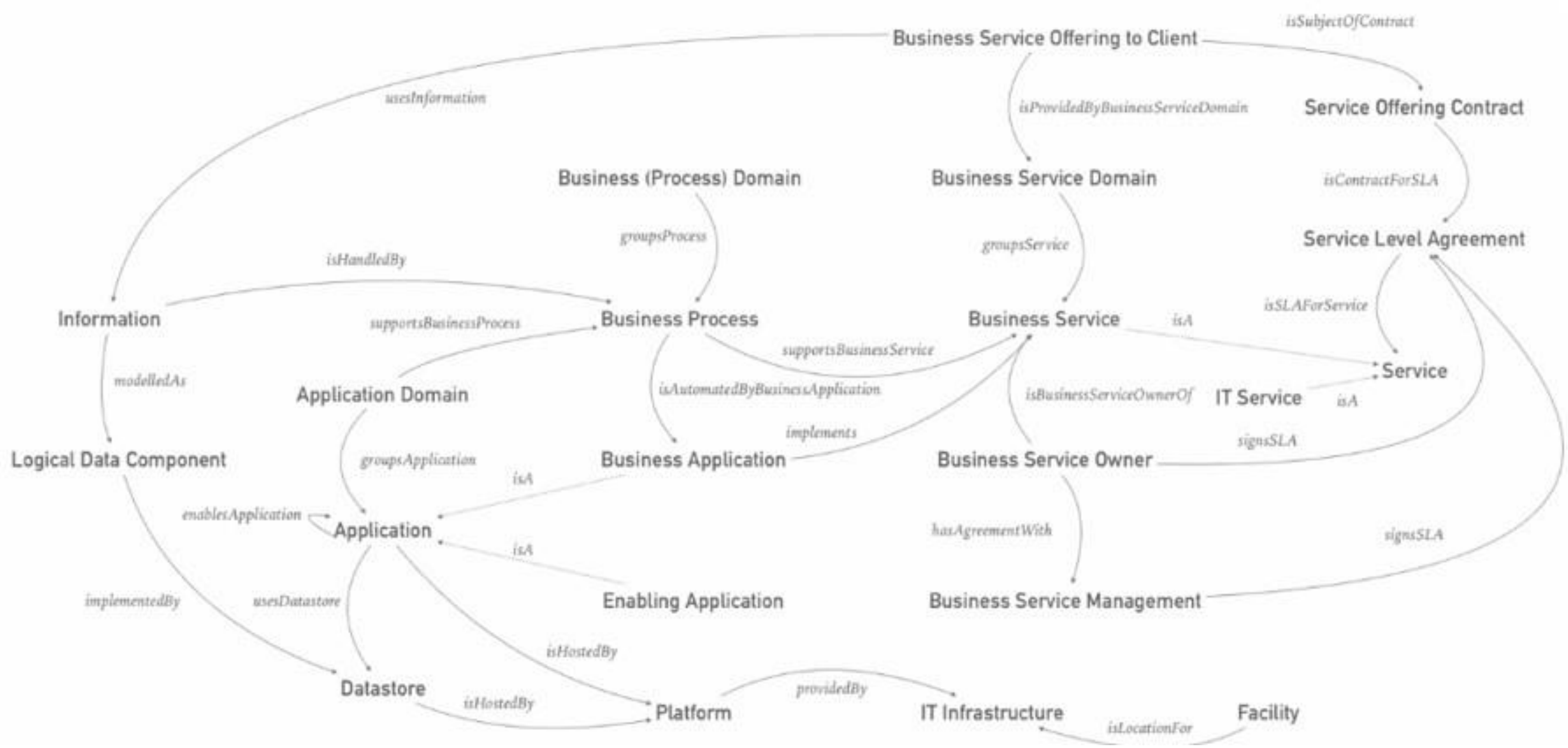
- Formalization
- Axiomatization
- Validation
- Verification
- Reasoning

ontotext

Mobile Order App Twin



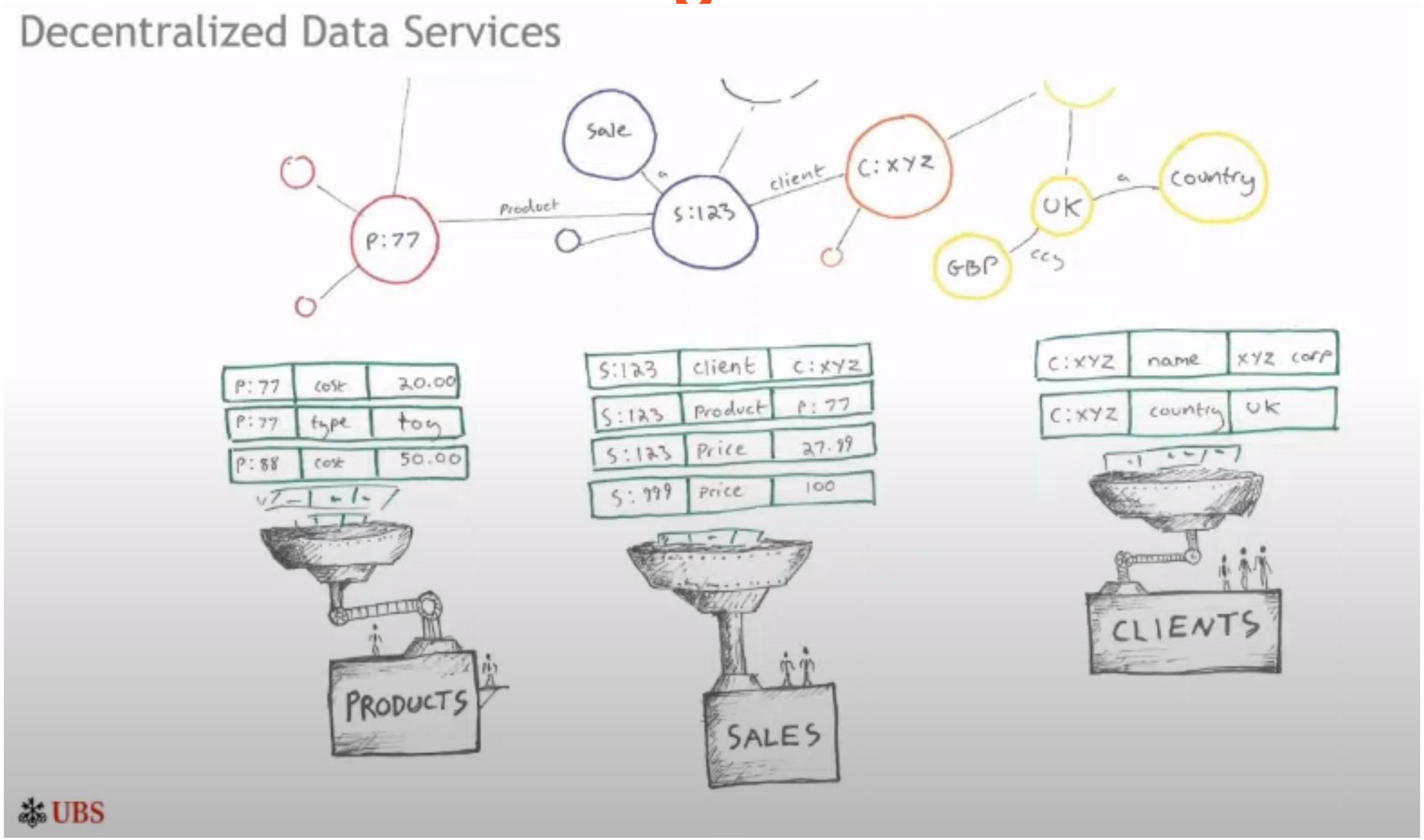
JPMC – Resilient Infrastructure



Highlights

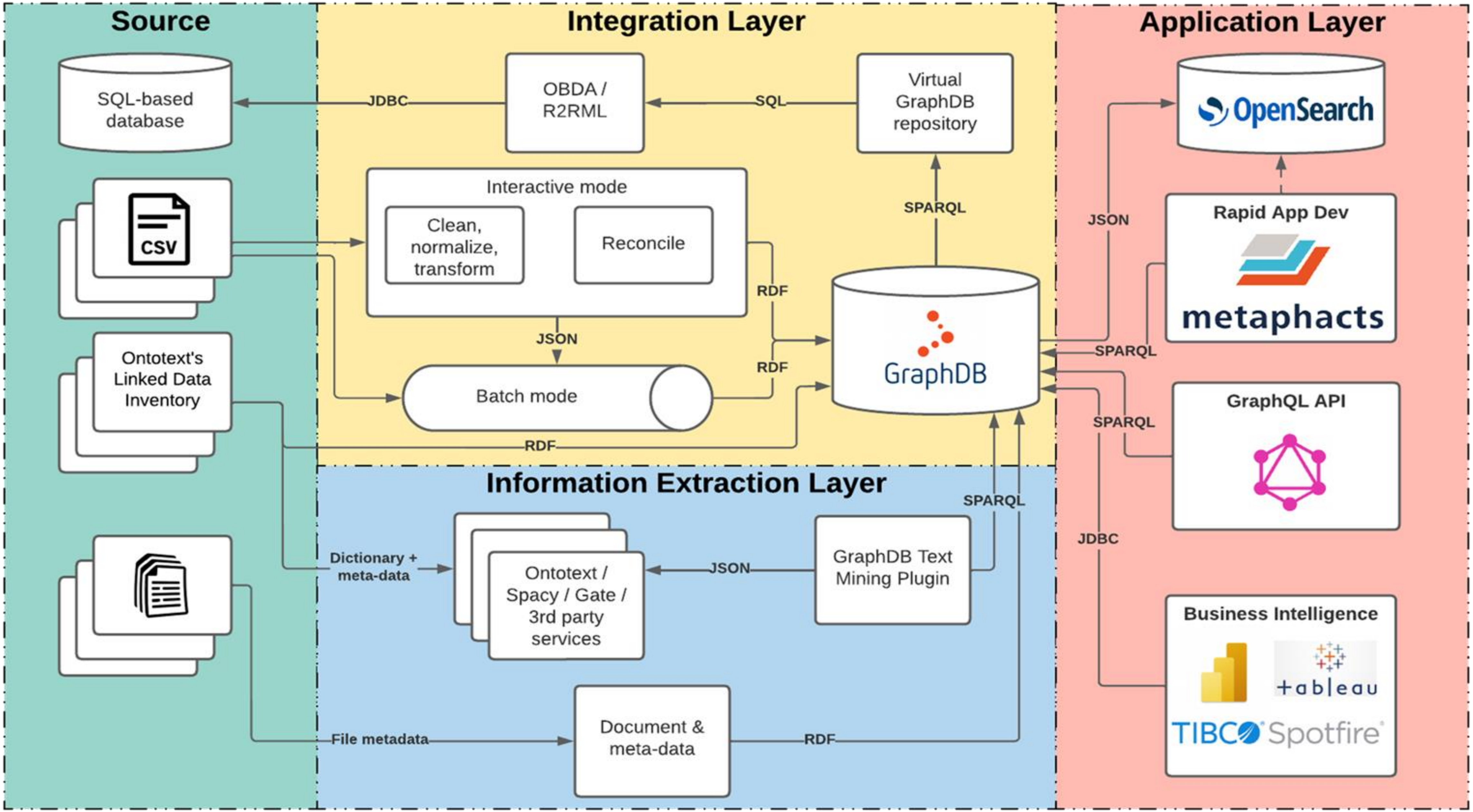
- Resilient, interconnected assets
- Impact prediction
- Reference data system

UBS IT Asset Management



Highlights

- Holistic view of IT assets
- Powerful analytics
- Support UBS and CS merger


Source
Integration Layer
Application Layer
SQL-based database
JDBC
OBDA / R2RML
SQL
Virtual GraphDB repository
OpenSearch
CSV
Interactive mode
Clean, normalize, transform
Reconcile
SPARQL
JSON
Rapid App Dev
metaphacts
RDF
GraphDB
Ontotext's Linked Data Inventory
JSON
Batch mode
RDF
SPARQL
GraphQL API
RDF
SPARQL
Information Extraction Layer
SPARQL
JDBC
Dictionary + meta-data
Ontotext / Spacy / Gate / 3rd party services
JSON
GraphDB Text Mining Plugin
Business Intelligence
+ableau
TIBCO Spotfire
File metadata
Document & meta-data
RDF

ontotext

Predictable Management of Diverse Data

Big Graphs & Metadata

- ✓ 10B statements of master data
- ✓ 1B entities and concepts
- ✓ Entity linking across 5+ data sources
- ✓ 10M documents, 100 KG tags/doc.

Text & Graph Analytics

- ✓ Extract new entities and facts from text
- ✓ Retrieval of similar documents and entities
- ✓ Automatic classification and link prediction
- ✓ Relevance and importance ranking

Query Performance

- ✓ 10 transactional updates/sec on master data
- ✓ 500 updates/sec for documents and metadata
- ✓ 100 graph queries/sec/node, incl. inferred facts
- ✓ RDFS+ reasoning: instant and transparent
- ✓ 1000 full-text searches/sec across docs and data

Operations & Data Quality

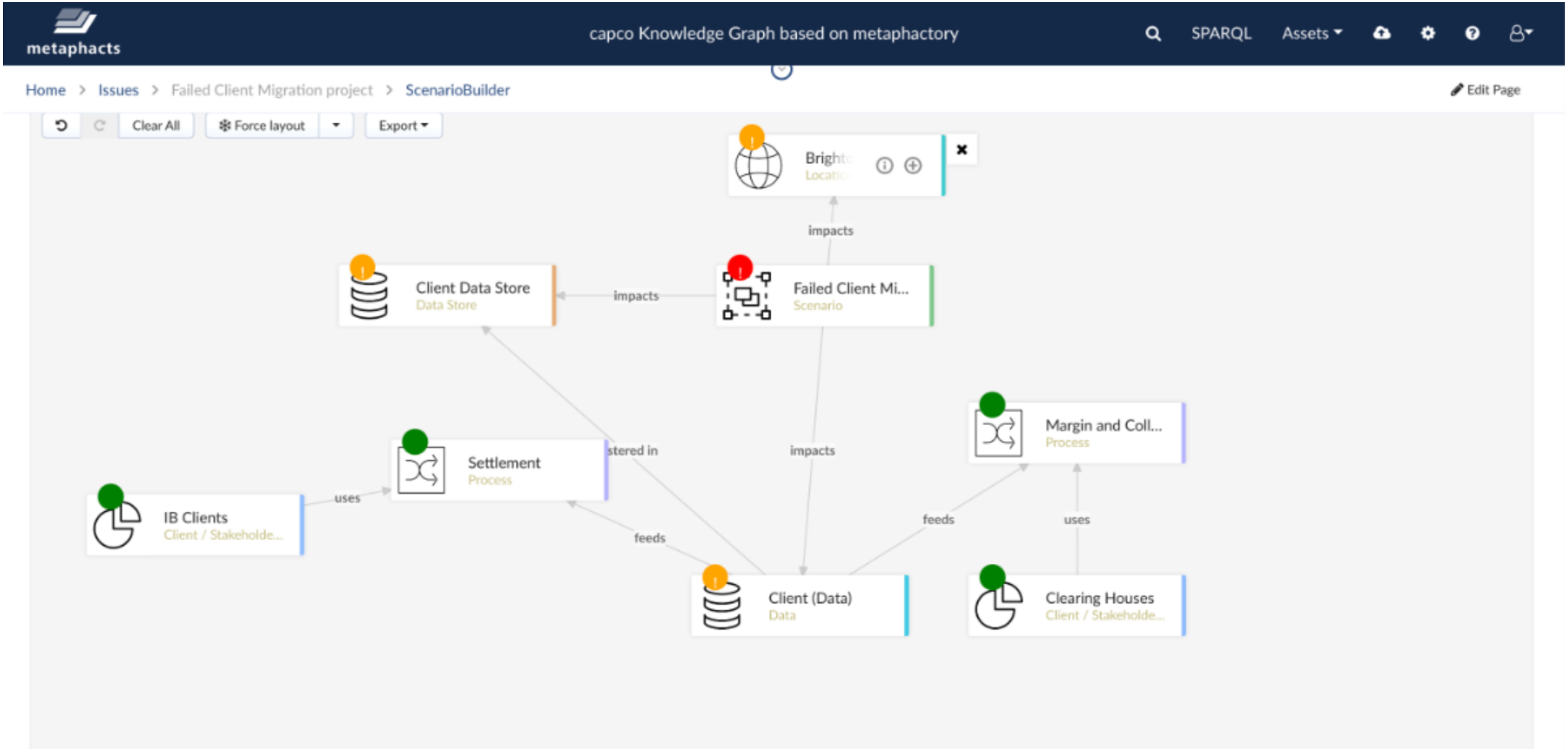
- ✓ Multi-DC deployment across continents
- ✓ Commodity workers: 16 vCPU, 32GB RAM
- ✓ Daily updates from external data sources
- ✓ Maintain quality of linking and text analysis
- ✓ Metadata and instance data curation

SMOOTH DATA INTEGRATION

Thank you!

ontotext

ontotext


metaphacts
capco Knowledge Graph based on metaphactory
SPARQL
Assets
Home
Issues
Failed Client Migration project
ScenarioBuilder
Edit Page
Clear All
Force layout
Export
Bright
impacts
Client Data Store
Data Store
impacts
Failed Client Mi...
Scenario
Margin and Coll...
Process
Settlement
Process
stered in
impacts
IB Clients
Client / Stakeholde...
uses
feeds
feeds
uses
Client (Data)
Data
Clearing Houses
Client / Stakeholde...

ontotext

Understand Schema

```
answer_question(df, question="What are the classes from the ontology?", debug=False)
```

```
'c4:Entity, c4:Agent, c4:Class, c4:Context, c4:User, c4:MessageBus, c4:WebApplication, c4:SystemElement, s
```

```
answer_question(df, question="What are the properties from the ontology?", debug=False)
```

```
'skos:example, skos:broaderTransitive, skos:definition, skos:hasTopConcept, skos:semanticRelation, skos:na
```

```
answer_question(df, question="How are c4:Context and c4:Container related?", debug=False)
```

```
'c4:Context is a subclass of c4:Entity, and c4:Container is a subclass of c4:SystemElement, which is a sub
```

```
answer_question(df, question="How are :RetailDatabase and c4:Container related?", debug=False)
```

```
':RetailDatabase is a subclass of c4:Container.'
```

ontotext

Write Queries

```
answer_question(df, question="Write SPARQL query to select all c4:Container", debug=False)
```

```
'SELECT ?container WHERE {\n  ?container a c4:Container .\n}'
```

```
answer_question(df, question="Write SPARQL query to select all c4:Container", debug=False)
```

```
'SELECT ?container WHERE {\n  ?container a c4:Container .\n}'
```

```
answer_question(df, question="Write SPARQL query to select all the dependencies of the Retail Order System",
```

```
'SELECT ?dependency WHERE {\n  ?dependency c4:hasRelationshipWith :RetailOrderSystem .\n}'
```

Explain

Graph

```python
answer_question(df, question="How the User interacts with the BackendAPI", debug=False)
```

'The User interacts with the BackendAPI through the Retail Order Mobile App.'

+ Code + Markdown

```python
answer_question(df, question="How the User is related to the OrderService?", debug=False)
```

'The User interacts with the RetailOrderApp, which interacts with the BackendAPI, which contains the OrderServi

```python
estion="Enumerate all system elements which will be affected if the database is no longer available?", debug
```

the database is no longer available are :RetailDatabase, :BackendAPI, :OrderService, :ProductService, and :Ret

ontotext