

Interaktive web-basierte Visualisierungen für große netzwerkartige Daten

Daniel Graf Hoyos, 29. September 2014 München

Software Engineering für betriebliche Informationssysteme (sebis)
Fakultät für Informatik
Technische Universität München

www.matthes.in.tum.de

1. Szenario

- Bisherige Lösung
- Anforderungen

2. Softwarearchitektur

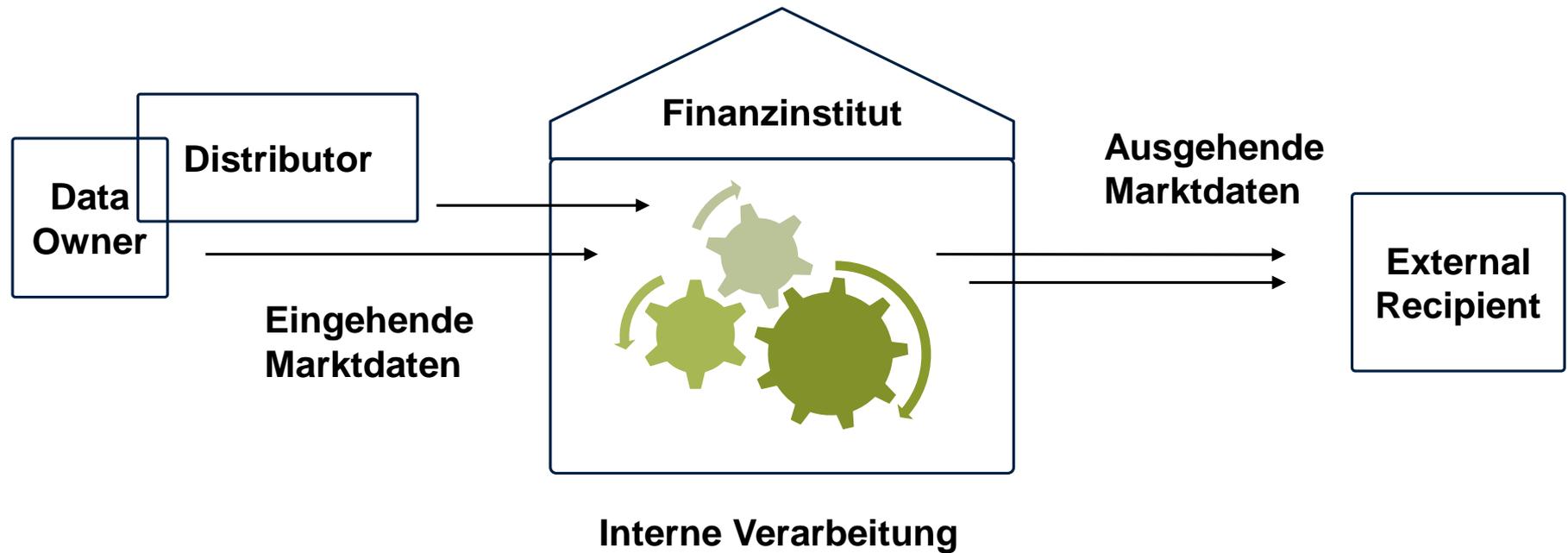
- OData
- Breeze
- AngularJS

3. Komponenten

- Filter
- Graph
- Tabellen

4. Ausblick

5. Prototyp Vorstellung



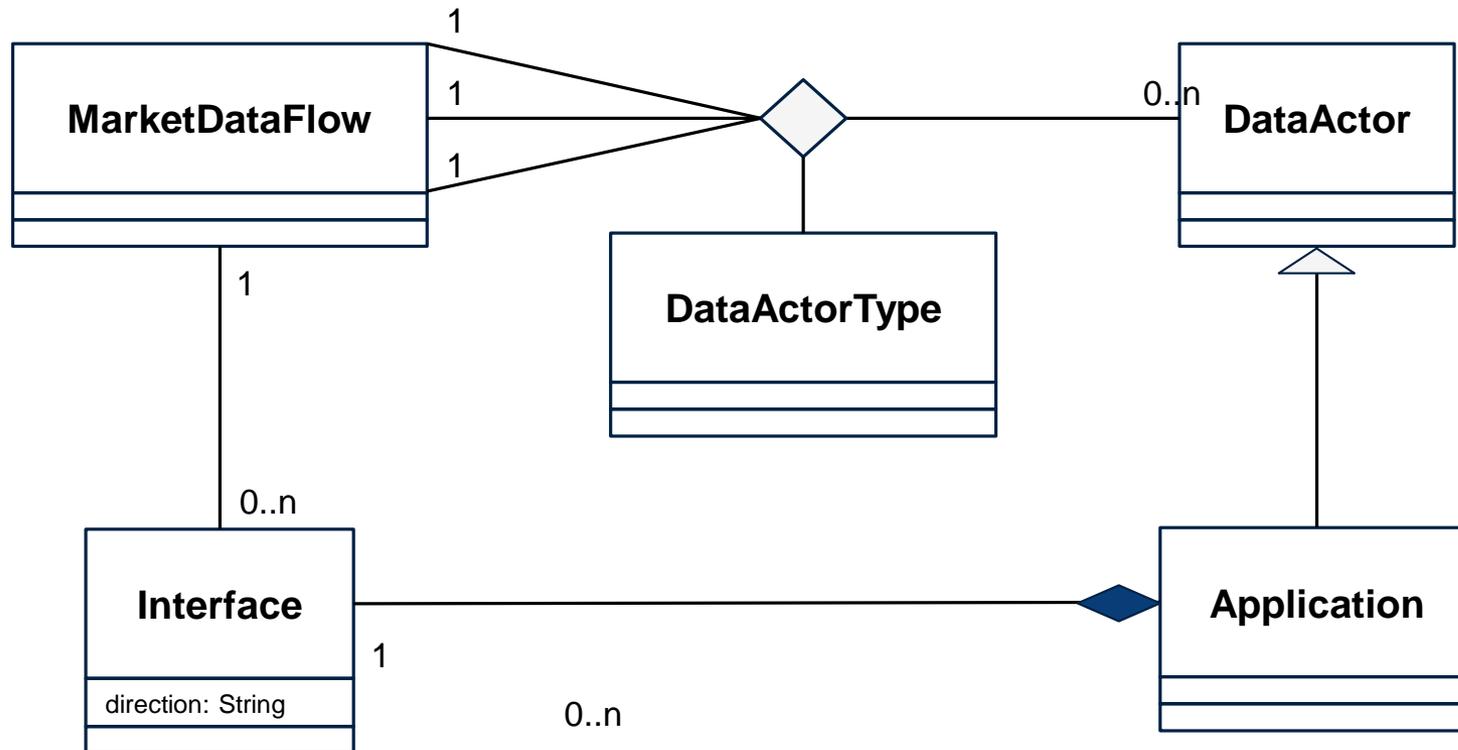
- Excel Fragebögen
- Access Daten Import
- Filter
 - Graphviz markup
- Market data map

Export market data map to graph

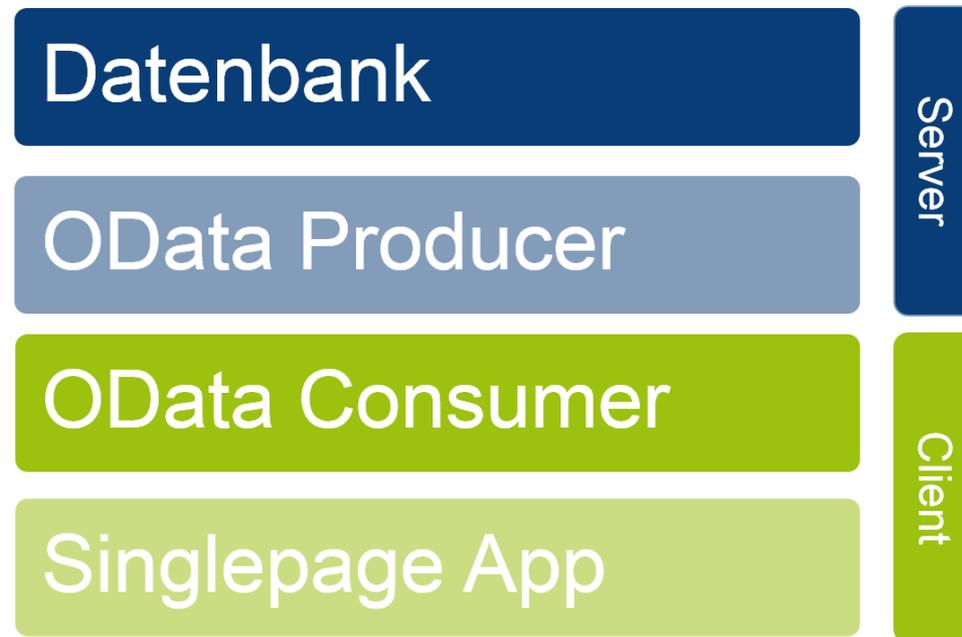
<p>Filter by data owners/systems</p> <p>Select to filter: <input type="text" value="Kondor+Milan"/> <small>no selection = do not filter</small></p> <ul style="list-style-type: none">KoscomLIMA Stock ExchangeLondon Metal ExchangeLSE GroupLucid <p>What to filter:</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Data owners<input type="checkbox"/> Distributors<input type="checkbox"/> Applications<input type="checkbox"/> External recipients <p>Refine filter: <input type="text" value="Six Swiss Exchange"/> <small>flows are only shown if both filters match; no selection = do not filter</small></p> <ul style="list-style-type: none">SmarthouseSnapshotEngineSophisStandard & Poor'sSTATAN <p>What to filter:</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Data owners<input checked="" type="checkbox"/> Distributors<input checked="" type="checkbox"/> Applications<input checked="" type="checkbox"/> External recipients	<p>Filter by flow type</p> <p>Type of transport:</p> <ul style="list-style-type: none"><input type="checkbox"/> Streaming<input type="checkbox"/> Snapshot<input type="checkbox"/> Batch (pull)<input type="checkbox"/> Batch (push) <p>Timeliness:</p> <ul style="list-style-type: none"><input type="checkbox"/> Real-time<input type="checkbox"/> Delayed<input type="checkbox"/> End of day <p><input type="checkbox"/> Aggregate <input type="checkbox"/> Aggregate</p> <p><input checked="" type="checkbox"/> Aggregate by data package <small>(only relevant if not aggregating by type of transport or timeliness)</small></p>	<p>Style options</p> <p>External system font size: <input type="text" value="48"/></p> <p>Internal application font size: <input type="text" value="48"/></p> <p>Market data flow font size: <input type="text" value="16"/></p> <p>Show as data source</p> <ul style="list-style-type: none"><input checked="" type="radio"/> Data owners<input type="radio"/> Distributors <p><input checked="" type="checkbox"/> Show potential compliance violations</p>
---	---	--

Label market data flows

- Data package name
- Data owner name
- Distributor name
- Interface name
- Data types
- Asset classes

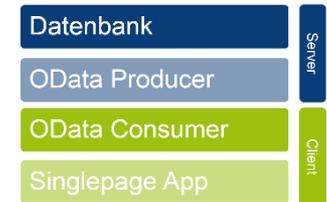


- Optische Unterscheidung (Entitäten und Datenflüsse)
- Navigierbarkeit der Graphen
- Konfigurierbare Darstellung (z.B. Kantenbeschriftungen)
- Filterung von Kanten und Knoten
- Aggregation von Kanten
- Definition und Anwendung von Regeln
- Detailinformationen von Knoten und Kanten
- Ausdruck oder Export
- CRUD Funktionalität
- Erweiterbarkeit des Schemas



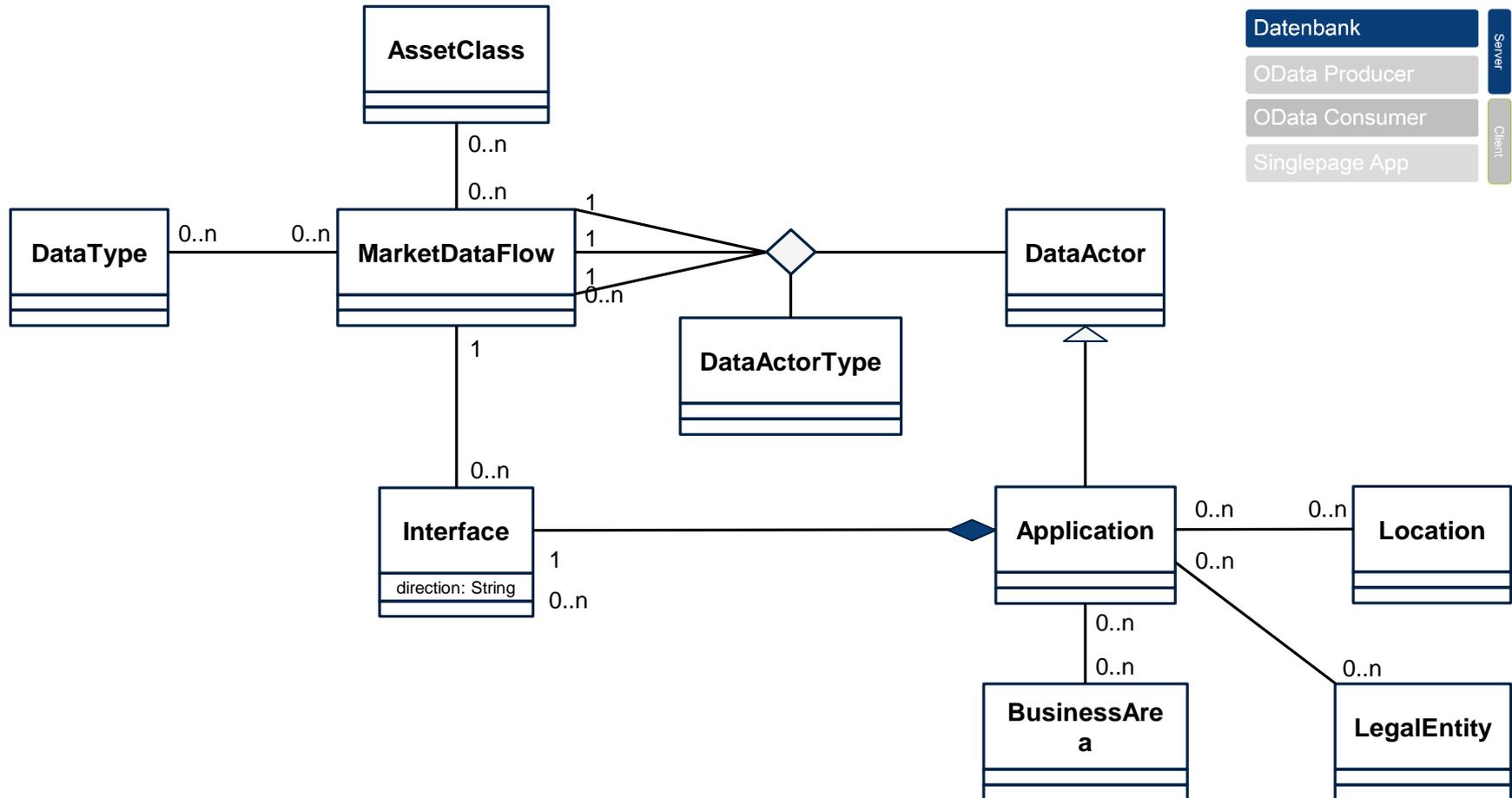
- MySQL
- Apache Olingo
- Tomcat
- breezeJS
 - dataJS
- AngularJS
 - xeditable
- D3
- Dagre
- Twitter Bootstrap
- Font Awesome

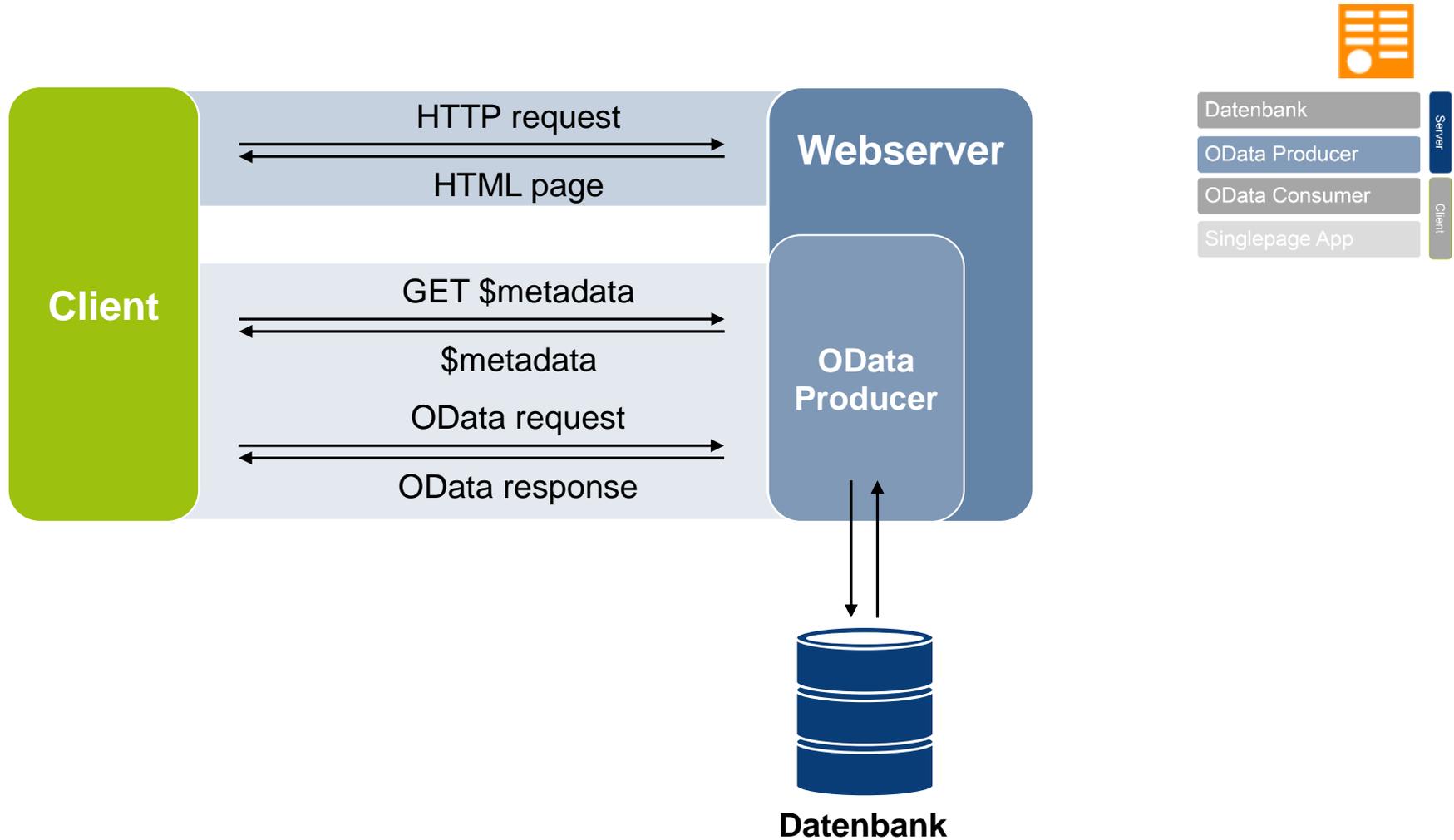
Datenbank
OData Producer
Java Application Server
OData Consumer
HTTP Kommunikation
MVC Framework
Detail Formulare
SVG Verwaltung
Graph Berechnung
CSS Design
Icons





Datenbank	Server
OData Producer	
OData Consumer	Client
Singlepage App	





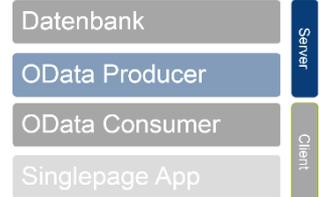
- OData 2.0 Unterstützung
- OASIS OData 4.0 in Entwicklung
- JPA model processor
- Entwickler: Apache Foundation unterstützt durch SAP

REST Operationen

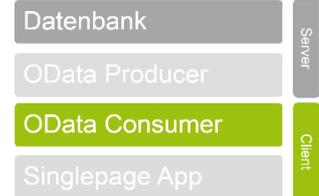
- GET, POST, PUT, PATCH, DELETE
- \$batch für gesammelte http anfragen

Nicht unterstützte Funktionen

- Keine groupBy unterstützung
- JoinColumns (nur olingo)



- JavaScript Bibliothek
- Asynchrone Kommunikation (Promises)
- Nachverfolgen von Änderungen
- Lokale Datenbankabfragen im Cache



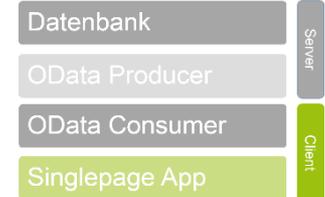
Asynchrone Anfrage

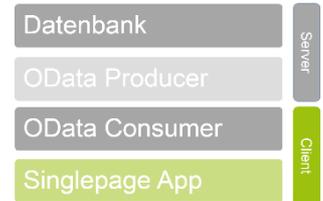
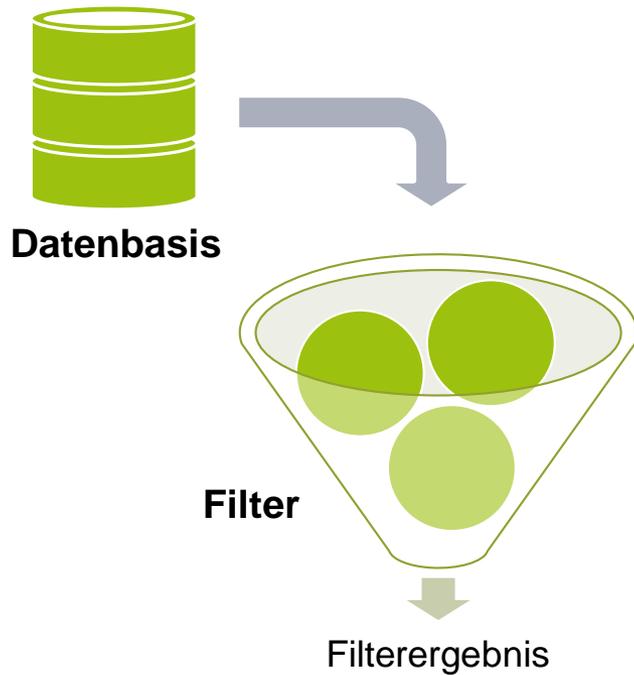
```
query = new breeze.EntityQuery ()
    .from('Interfaces')
    .where('applicationID', 'Eq', 1234);
manager.executeQuery(query)
    .then(function(data) {
        interfaces = data.results;
    });
```

Synchrone Anfrage

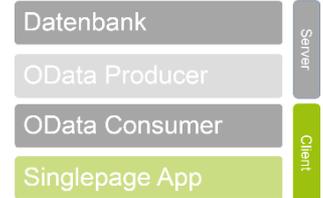
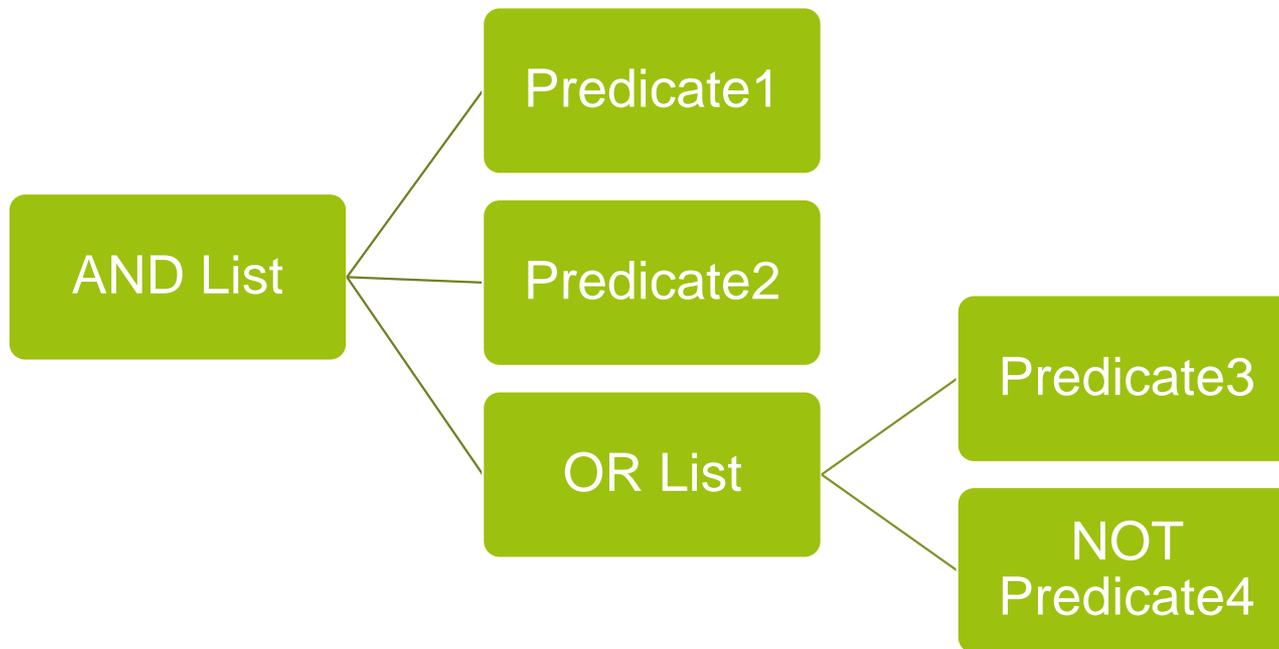
```
interfaces = breeze.EntityQuery
    .from('Interfaces')
    .where('applicationID', 'Eq', 1234)
    .using(manager)
    .executeLocally();
```

- JavaScript Framework
- HTML = static web → AngularJS = dynamic web
- Model-View-Controller pattern (MVC)
 - Separation of concerns
 - Austauschbare Komponenten
 - Geschäftslogik getrennt von Anzeigelogik
- Custom HTML Tags (directives)
<myTag></myTag>





Data Actor Filter



Data Actor Filter

AND OR

Application: Application XY, Application F

Data Owner
 Distributor
 Application
 External Recipient

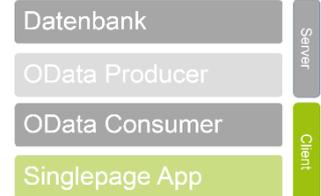
Search Field

- Application XY
- News Agency C
- Data Producer Y
- Vendor ZXY
- Stock Exchange ABC

AND OR

Data Owner: Stock Exchange ABC

Distributor: Vendor ZXY, Stock Exchange ABC



- Alternative Data Actor Rollen wählbar
- Alternative Data Actors wählbar
- AND bzw. OR Verknüpfungen
- Jede Ebene negierbar
- Fehlerwarnungen

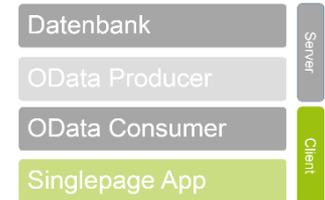
Markdatenfluss Filter

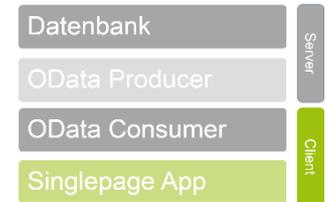
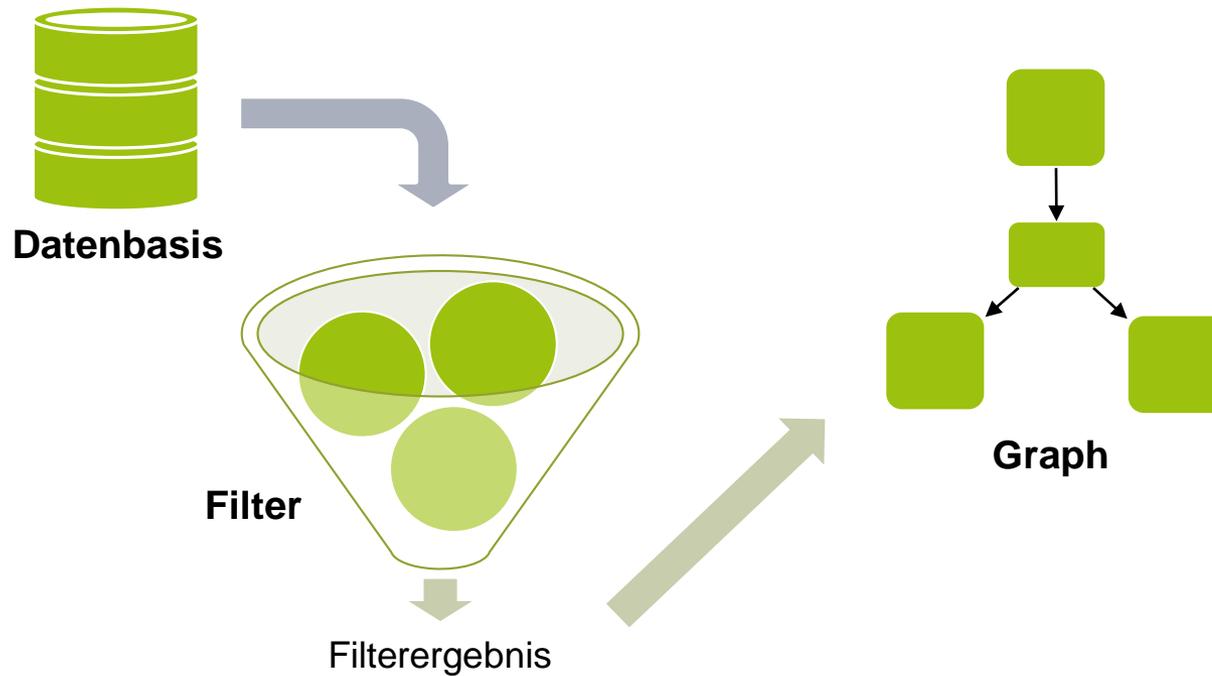
Timeliness <input checked="" type="checkbox"/> real-time <input checked="" type="checkbox"/> delayed <input type="checkbox"/> end of the day <input type="checkbox"/> historical	Type of transport <input type="checkbox"/> streaming <input checked="" type="checkbox"/> snapshot <input type="checkbox"/> batch (pull) <input type="checkbox"/> batch (push)
---	--

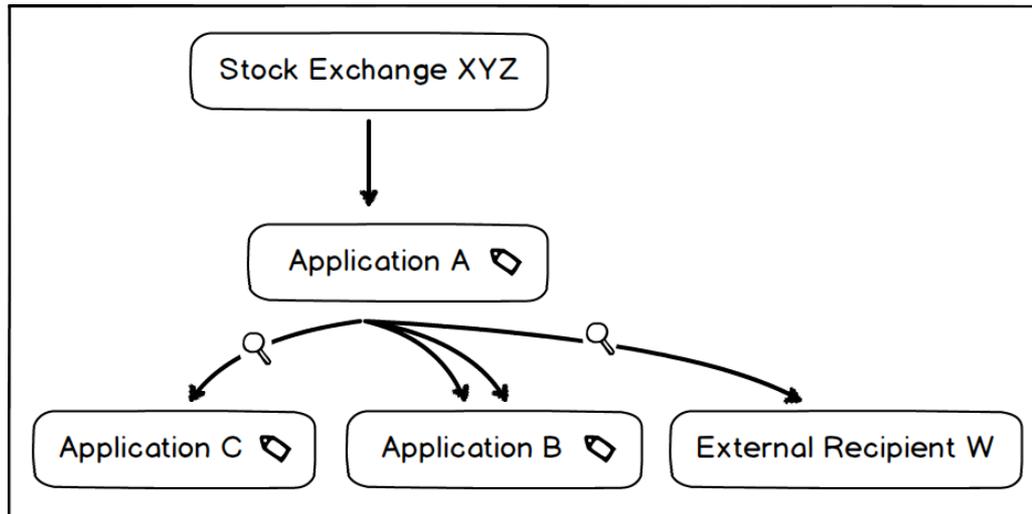
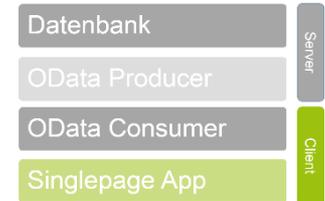
- Alternative Aggregation oder Aggregieren aller Kanten
- Frei kombinierbare Labels
- Graphenorientierung wählbar
- Fehlerwarnungen

Graph Einstellungen

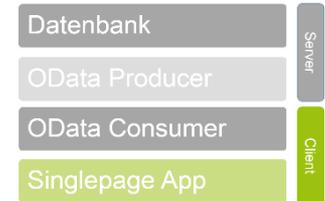
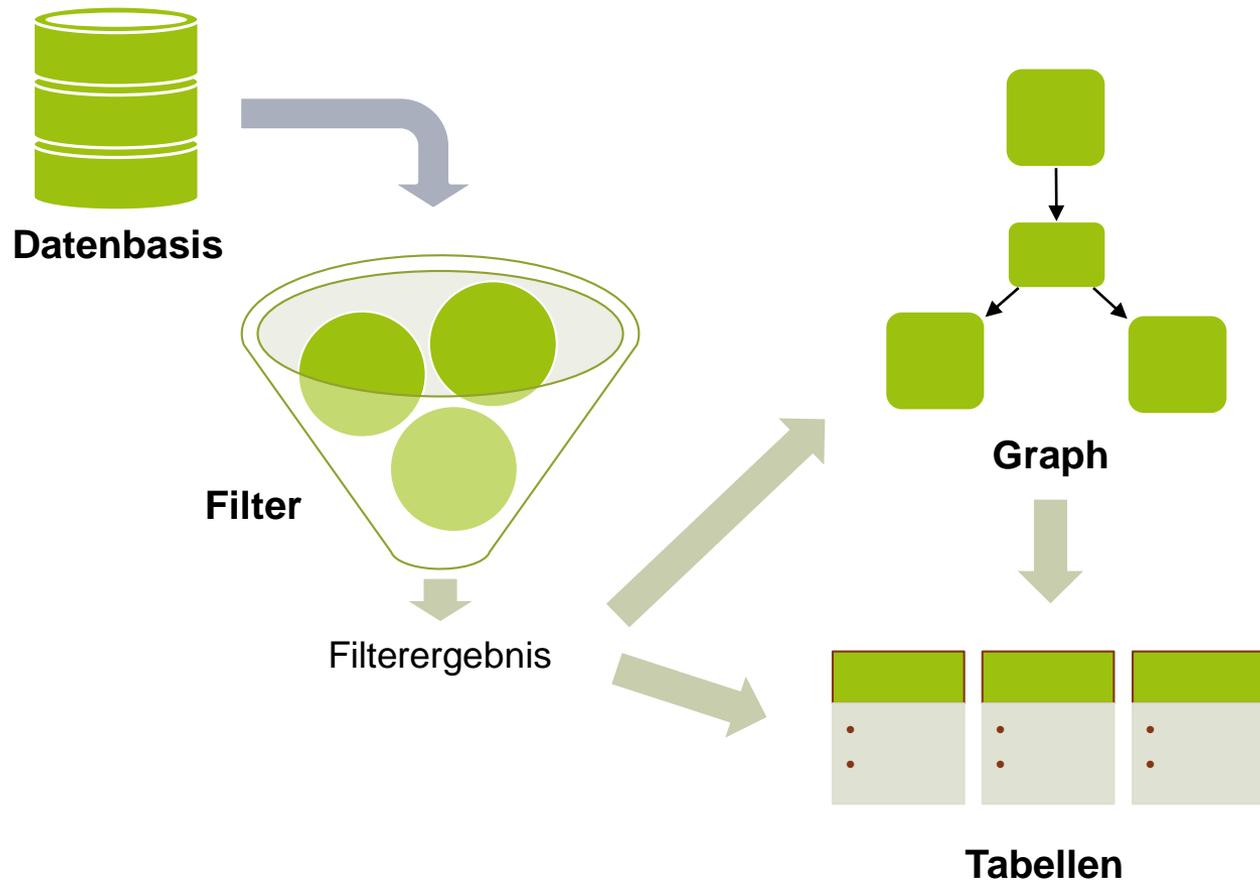
Aggregate edges <input type="radio"/> by data packages <input checked="" type="radio"/> by timeliness <input type="radio"/> by type of transport	Use as data source <input checked="" type="radio"/> data owner <input type="radio"/> distributor
--	---



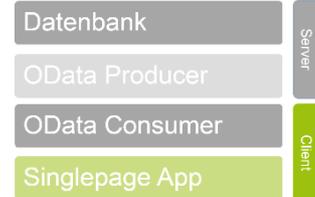




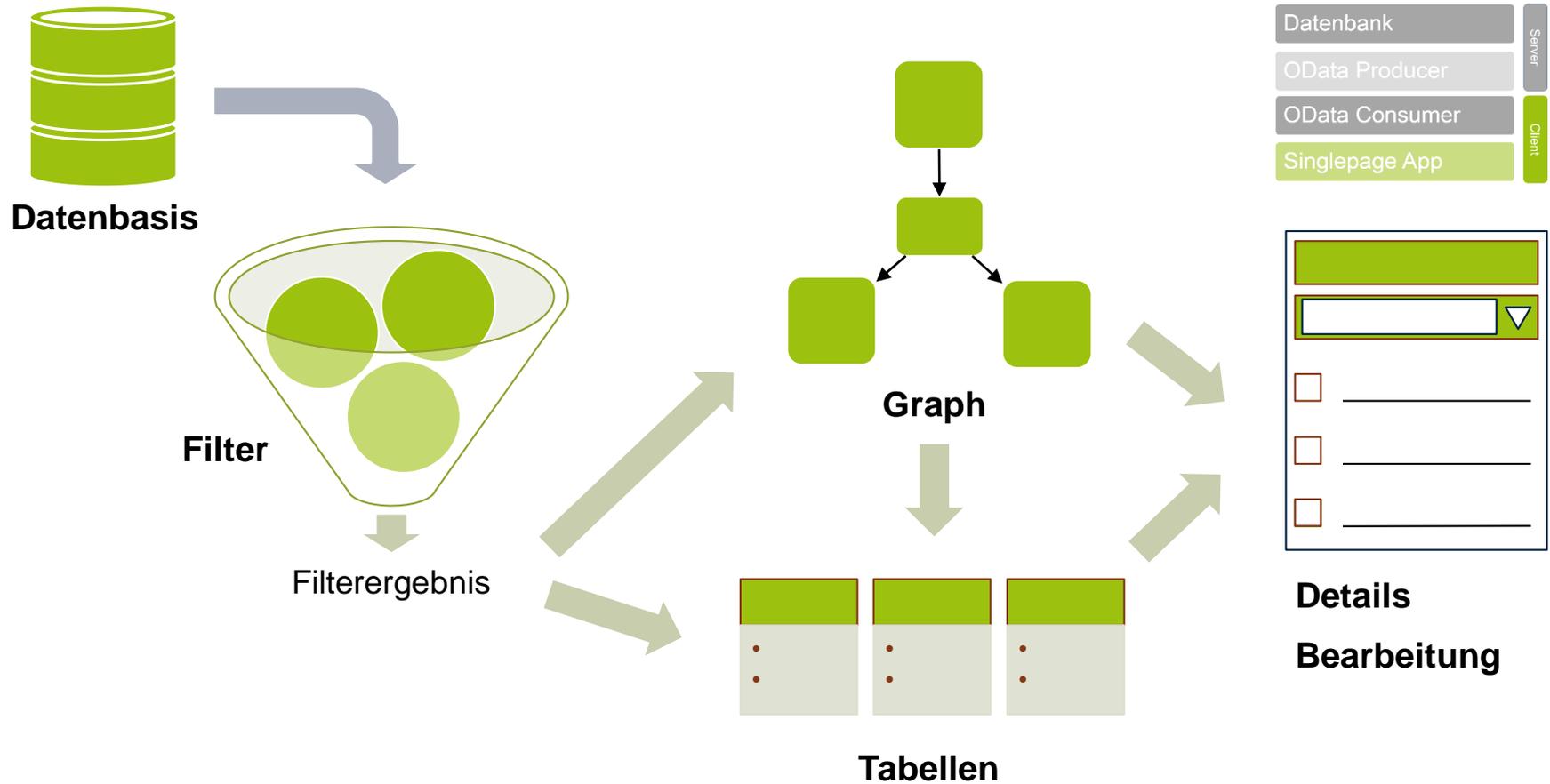
- Data Actors sind Knoten
- Marktdatenflüssen sind gerichtete Kanten
- Applikationen sind editierbar
- Einzelne Kanten sind editierbar
- Ursprung einer Kante ist Data Owner oder Distributor
- Navigierbar (drag'n'zoom)



Application X	ID X	User Number X	Location ▾ + add field
<input type="text" value="search"/>	<input type="text" value="search"/>	<input type="text" value="search"/>	
Application XY	1234	2.153	 
Risk Mgmt ABC	2089	121	 
Broker Tool YZ	1047	ca. 897	 



- Konfigurierbare Spalten
- Nach Feldern sortierbar
- Mehrfaches Eingrenzen durch Suchfelder
- Detailansichten
- Sublisten



Application XY

Administrative Data

Interviewer: Person AB
Interviewee: Person CD
Date of last change: 17.05.2007

Application Info

ID: XY
Name: Application XY
Data Owner: Stock Exchange XYZ
Distributor: Vendor PQ

Location

Wien, London

Application XY

Administrative Data

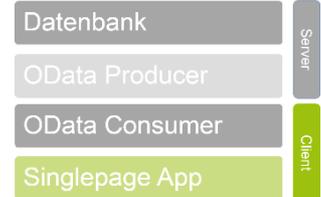
Interviewer
Interviewee
Date of last change

Application Info

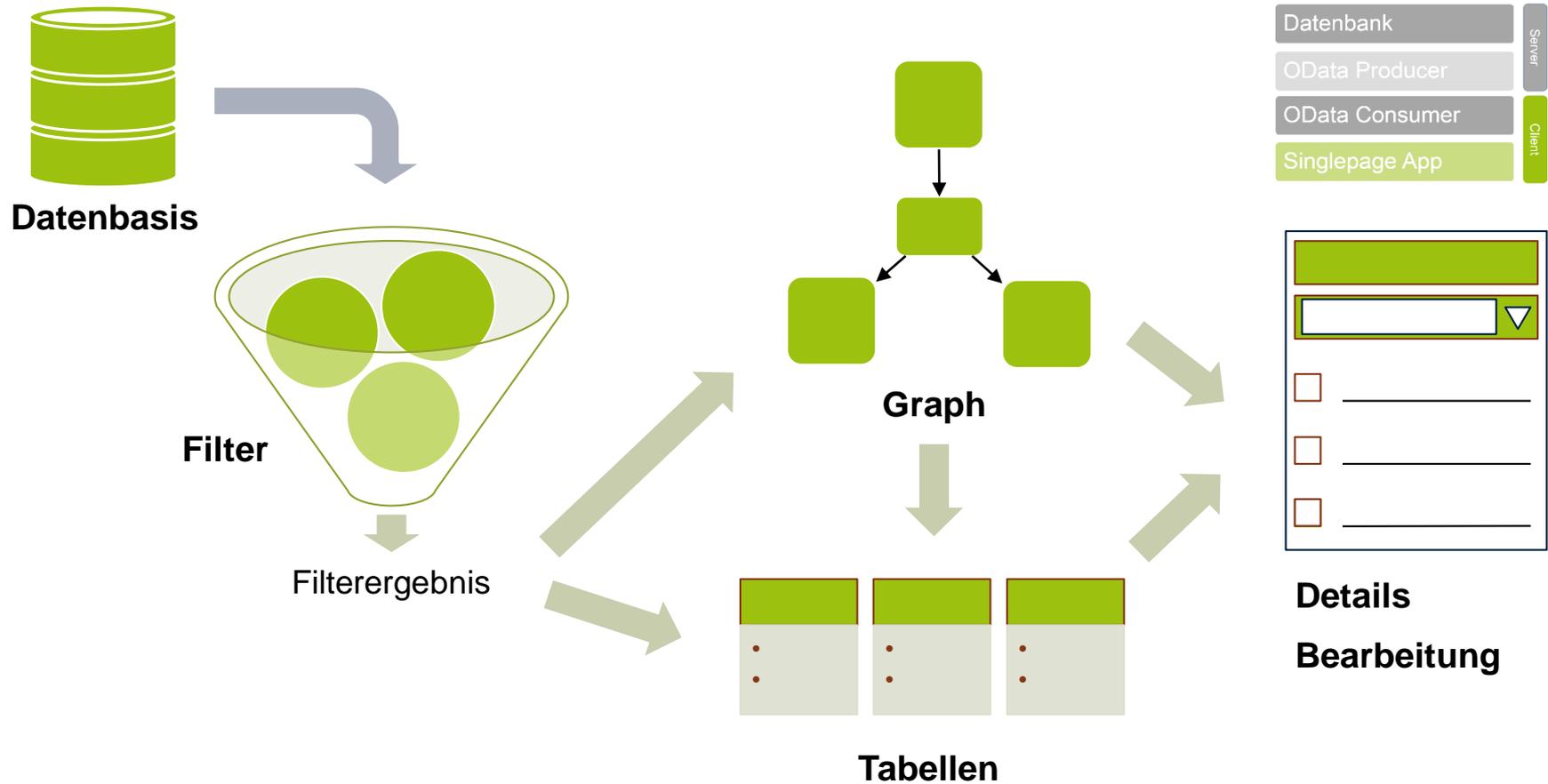
ID
Name
Data Owner ▼
Distributor ▼

Location

Berlin
 Wien
 London

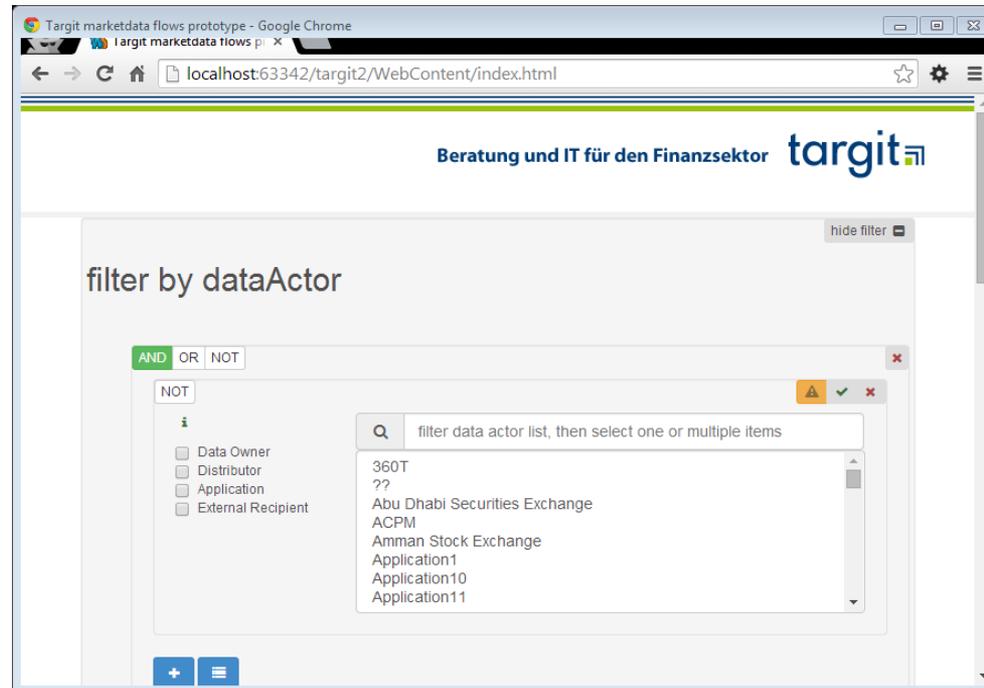


- Temporäres bearbeiten
- Freitextfelder
- Dropdown
- Checkbox und CheckList
- RadioList
- Löschen



Mögliche Erweiterungen

- Rechteverwaltung
- Filter Templates speichern / exportieren / importieren
- Mini Map zur besseren Navigierbarkeit
- Komplexere Kantenbeschriftungen bei Aggregation



Fragen – Anmerkungen – Diskussion



Daniel Graf Hoyos
daniel.hoyos@tum.de



Technische Universität München
Fakultät für Informatik
Lehrstuhl Software Engineering für
betriebliche Informationssysteme

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.
Fax +49.89.289.17136

wwwmatthes.in.tum.de