

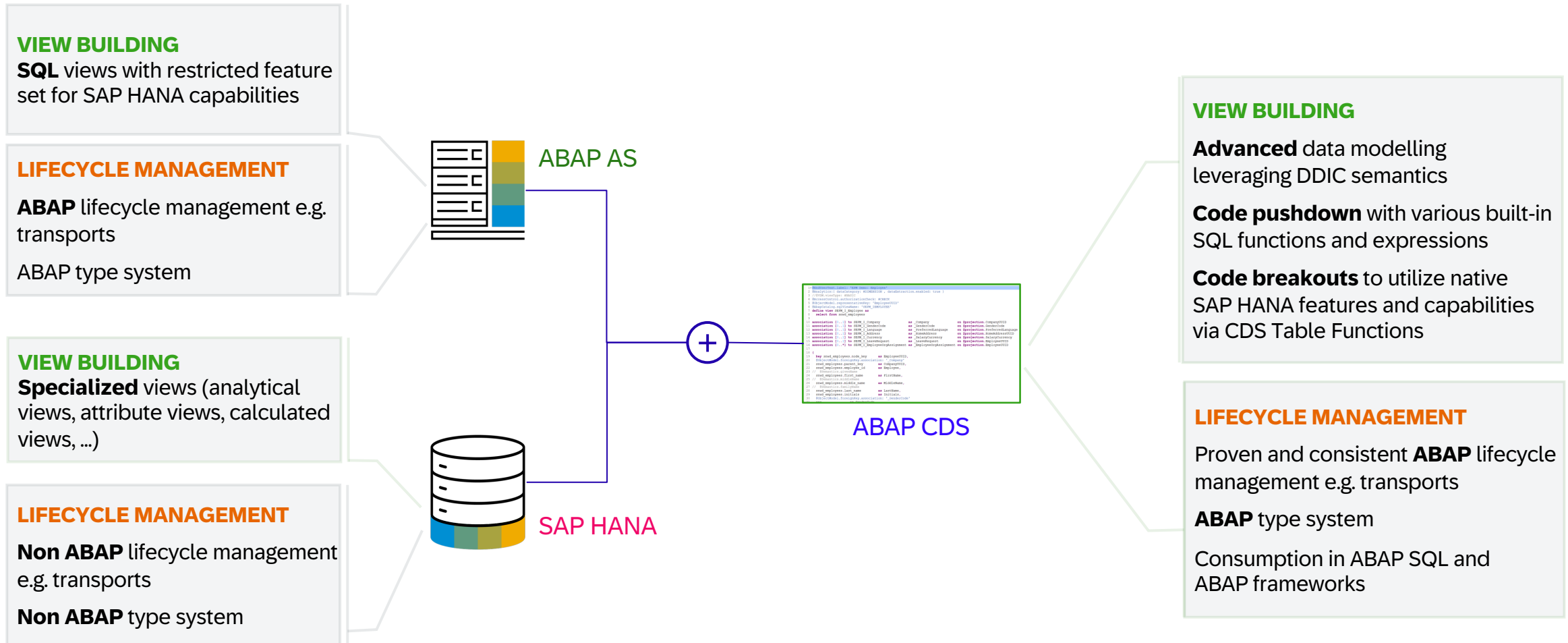
# SAP Fiori Development Enablement

## ABAP Core Data Services (CDS) Overview

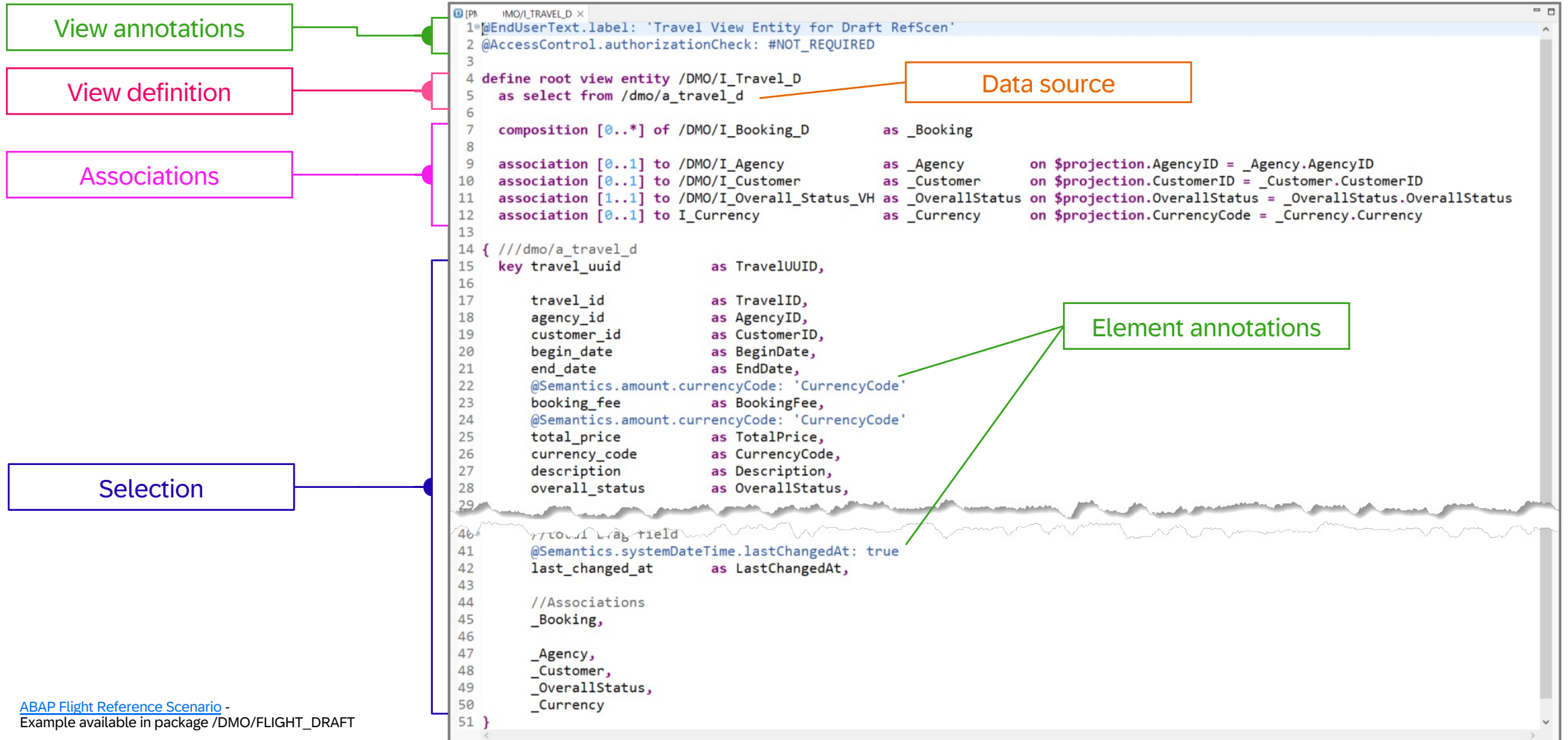
Leonardo Britz, SAP  
Q2 2024

CONFIDENTIAL

# Next generation data modeling and access with ABAP CDS



# Next generation data modeling and access with ABAP CDS – Example



# ABAP CDS Objects

# ABAP CDS View Entities

- An **ABAP CDS view entity** is used to select fields from one or more data sources (database tables or other CDS entities).
- They are used to define the **business objects** of your custom application.
- CDS view entities are the successor of CDS DDIC-based views (**define view** obsolete).
- Right click on the ABAP Package and choose New → Core Data Services → Data Definition.

```
define view entity ${ddl_source_name} as select from ${data_source_name}
{
  ${data_source_elements}${cursor}
}
```

# ABAP CDS Associations

- An **ABAP CDS association** associates **(joins)** the current CDS view entity as association source with the association target using an ON condition.

```
define view entity ${ddl_source_name} as select from ${data_source_name}
association [1] to ${target_data_source_name} as ${_association_name}
on $$projection.${element_name} = ${_association_name}.${target_element_name}
{
  ${data_source_elements}${cursor}
  ${_association_name} // Make association public
}
```

# ABAP CDS Composition

- An **ABAP CDS composition** is a specialized CDS association that defines the current CDS view entity as **parent entity** of the composition target.
- The composition target is the **child entity** and it must define a to-parent association to its parent entity.
- CDS compositions define the **hierarchical structure of a business object** in the context of the ABAP RESTful Application Programming Model.

```
define root view entity ${ddl_source_name} as select from ${data_source_name}
composition of ${target_data_source_name} as ${_association_name}
{
  ${data_source_elements}${cursor}
  ${_association_name} // Make association public
}
```

```
define view entity ${ddl_source_name} as select from ${data_source_name}
association to parent ${target_data_source_name} as ${_association_name}
on $$projection.${element_name} = ${_association_name}.${target_element_name}
{
  ${data_source_elements}${cursor}
  ${_association_name} // Make association public
}
```

# ABAP CDS Projection

- An **ABAP CDS projection** view, exposes a subset of a RAP business object's elements for a dedicated use case.
- The **provider contract** specifies the scenario in which a CDS projection view is used, this determines which syntax checks, features and runtime are used.
- ✓ **transactional\_query**: Enhances the RAP business object with additional data or behavior without changing the underlying RAP business object.
- ✓ **transactional\_interface**: Serves as a stable public interface to the RAP business object, only a subset of existing elements or behaviors can be used. By adding a release contract, only non used elements and behaviors of the underlying RAP business object can be changed, so as to keep the interface stable.

```
define view entity ${ddl_source_name}  
  provider contract transactional_query as projection on  
  as projection on ${data_source_name}  
  {  
    ${data_source_elements}${cursor}  
  }
```



# ABAP CDS Annotations

- A [CDS annotation](#) enriches CDS objects with **metadata** and are added in specific places in a piece of CDS source code or in a CDS metadata extension.
- Annotations are either evaluated by the ABAP runtime or other software components like the SAP Fiori Elements Framework.
- Refer to [CDS Annotations](#) for all annotations.

## Example: **Semantics annotation**

```
define root view entity...
{
...
@Semantics.amount.currencyCode: 'CurrencyCode'
booking_fee as BookingFee,
@Semantics.amount.currencyCode: 'CurrencyCode'
total_price as TotalPrice,
currency_code as CurrencyCode,
..
}
```

## 211

[Travel](#)[Booking](#)

Travel UUID:  
02fecec2-c880-1edd-b2ac-3038ffef1635

Travel ID:  
211

Agency ID:  
Sunshine Travel (70001)

Customer ID:  
Buchholm (1)

Starting Date:  
Mar 23, 2023

End Date:  
May 10, 2023

Booking Fee:  
240.00 EUR

Total Price:  
3,555.95 EUR

# ABAP CDS Annotations

Example: **Consumption, ObjectModel and Search** annotations:

```
@Search.searchable: true
define root view entity...
{
  @Consumption.valueHelpDefinition:
  [{ entity: { name: '/DMO/I_Customer', element: 'CustomerID'} }]
  @ObjectModel.text.element: ['CustomerName']
  @Search.defaultSearchElement: true
  CustomerID,
  _Customer.LastName as CustomerName,...
}
```

Standard

Editing Status: All Travel ID: Agency ID: Customer ID: Go Adapt Filters (1)

Travels (18)

Reject Travel Accept Travel Create Delete

Travel ID	Agency ID	Customer ID	Starting Date	End Date	Booking Fee	Total Price
211	Sunshine Travel (70001)	Buchholm (1)	Mar 23, 2023	May 10, 2023	240.00 EUR	3,555.95 EUR
210	99901	Buchholm (1)	Feb 16, 2023	Feb 23, 2023	100.00 EUR	100.00 EUR
209	Sunshine Travel (70001)	Buchholm (1)	Jan 13, 2023	Jan 20, 2023	1,800.00 EUR	2,308.51 EUR
208	Happy Hopping (70003)	Buchholm (4)	Dec 7, 2022	Dec 14, 2022	0.00	0.00
207	99901	Buchholm (1)	Jun 14, 2023	Jun 16, 2023	180.00 EUR	180.00 EUR
205	99901	Buchholm (1)	Feb 8, 2021	Feb 26, 2021	50.00 EUR	6,285.00 EUR
192	Burns Nuclear (70008)	Sisko (656)	Nov 28, 2019	Nov 28, 2019	30.00 SGD	14,279.00 SGD
168	Real Weird Vacation (70037)	Buehler (228)	Nov 28, 2019	Sep 24, 2020	120.00 SGD	43,477.00 SGD

# ABAP CDS Annotations

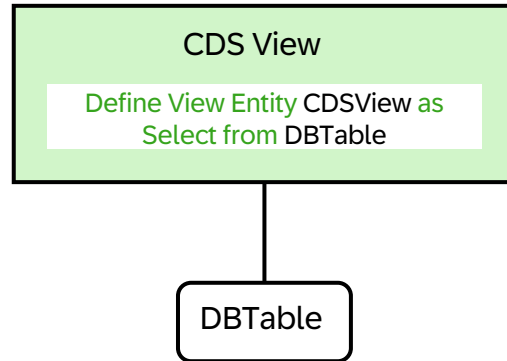
- The annotations of **CDS metadata extensions** expand (or override) the annotations specified in the source code of a CDS entity.
- Right click on the CDS Entity and choose New Metadata Extension.

```
@Metadata.layer: #CORE
annotate entity ...
with
{
  @UI.facet: [ { id:      'Travel',
                 purpose:  #STANDARD,
                 type:     #IDENTIFICATION_REFERENCE,
                 label:    'Travel',
                 position:  10 },
    { id:      'Booking',
      purpose:  #STANDARD,
      type:     #LINEITEM_REFERENCE,
      label:    'Booking',
      position:  20,
      targetElement: '_Booking'} ]
```

211	
<div>Travel   Booking</div>	
Travel UUID: 02fecec2-c880-1edd-b2ac-3038ffef1635	Starting Date: Mar 23, 2023
Travel ID: 211	End Date: May 10, 2023
Agency ID: Sunshine Travel (70001)	Booking Fee: 240.00   EUR
Customer ID: Buchholm (1)	Total Price: 3,555.95   EUR

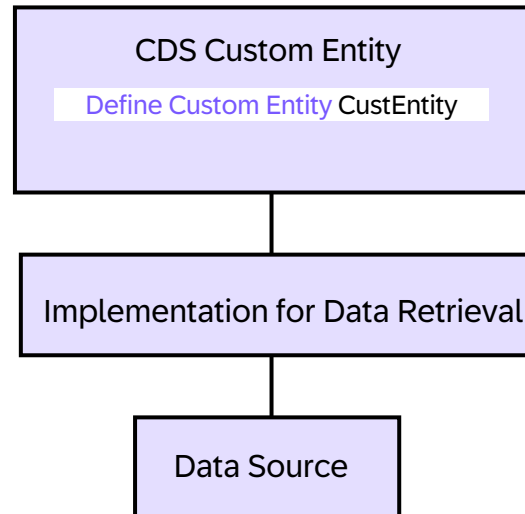
# Query Implementation Types

# Query Implementation Types



## MANAGED

Orchestration framework manages the data access to the database  
→ Application developer does not have to implement anything (only data model and related access controls)

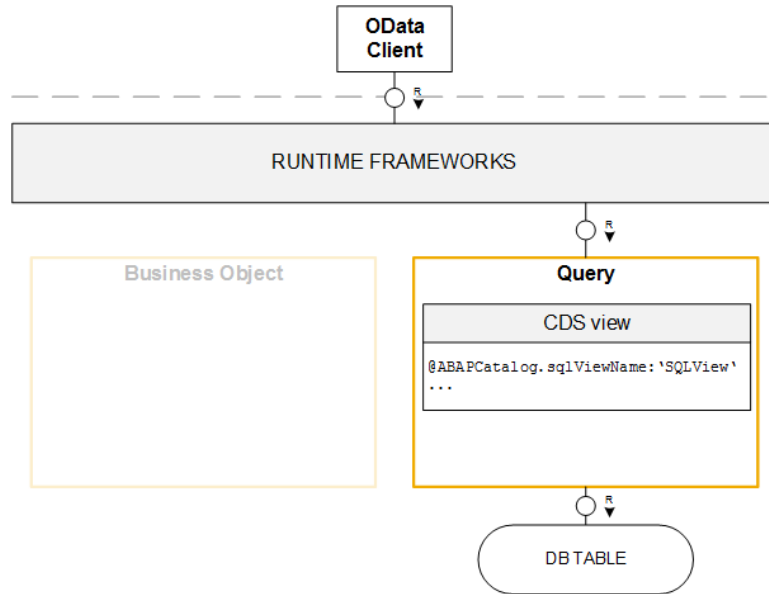


## UNMANAGED

Application developer manages the data access to the database  
→ Data source is another OData service (not a database table), which is reached by an OData client proxy  
→ CDS Custom Entity (no SQL view) with ABAP class that implements the query

# Query Implementation Types

## Managed

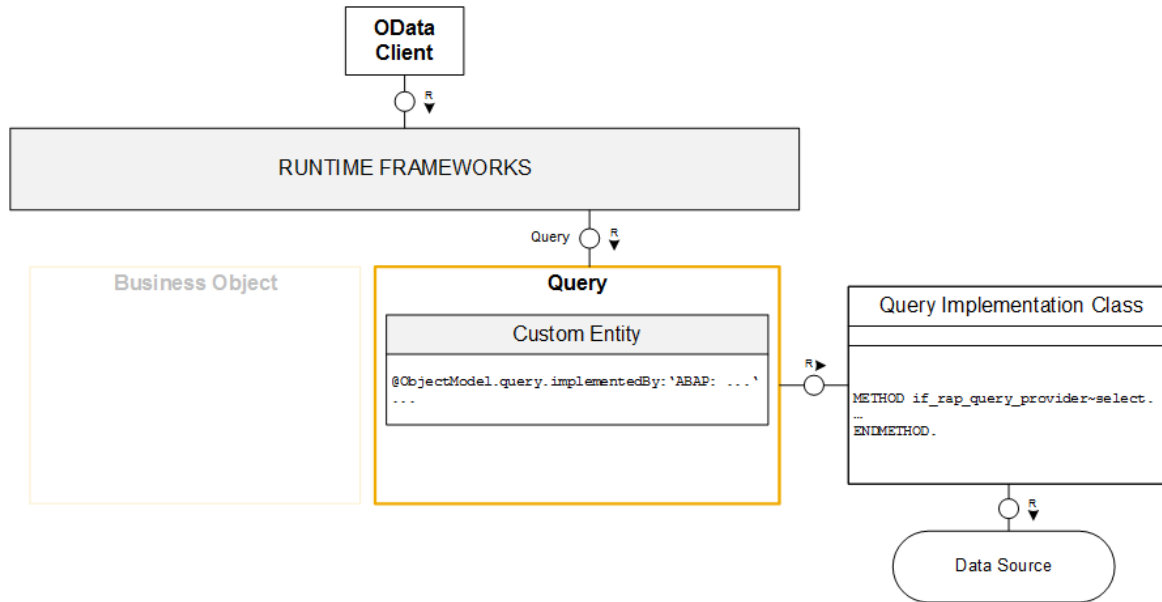


### SQL DB Access

- Managed query is default case
- Considers query capabilities from OData query options (\$orderby, \$top, \$skip ...)
- Considers authorizations derived from access control
- Orchestration framework triggers the query to be executed
- Query framework creates the SQL statement, optionally extends with where clause and evaluates authorization

# Query Implementation Types

## Unmanaged



### Protocol Agnostic

- Standard SQL push-down not possible or sufficient
- CDS Custom Entity references ABAP implementation class
- Implements ABAP interface (IF\_RAP\_QUERY\_PROVIDER) to handle request and response data
- Can have parameters, elements and associations, be a root, parent, or a child entity
- Cannot be used in ABAP SQL SELECT (no DB representation)
- Application developer has to manually take care of query options and access control

### Use Cases

- Data source is not a database table, but another OData service, reached by an OData client proxy
- Using AMDPs with a query push-down SQL script implementation

# Thank you.

Contact information:

Leonardo Britz  
Senior UX Consultant  
[leonardo.britz@sap.com](mailto:leonardo.britz@sap.com)

