



Week 4: Dealing with Existing Code

## Unit 1: The Business Scenario

# The Business Scenario

## Topics

### Week 4

**01**

The Business Scenario

**02**

Creating the CDS Data Model

**03**

Defining and Implementing the Business Object Behavior

**04**

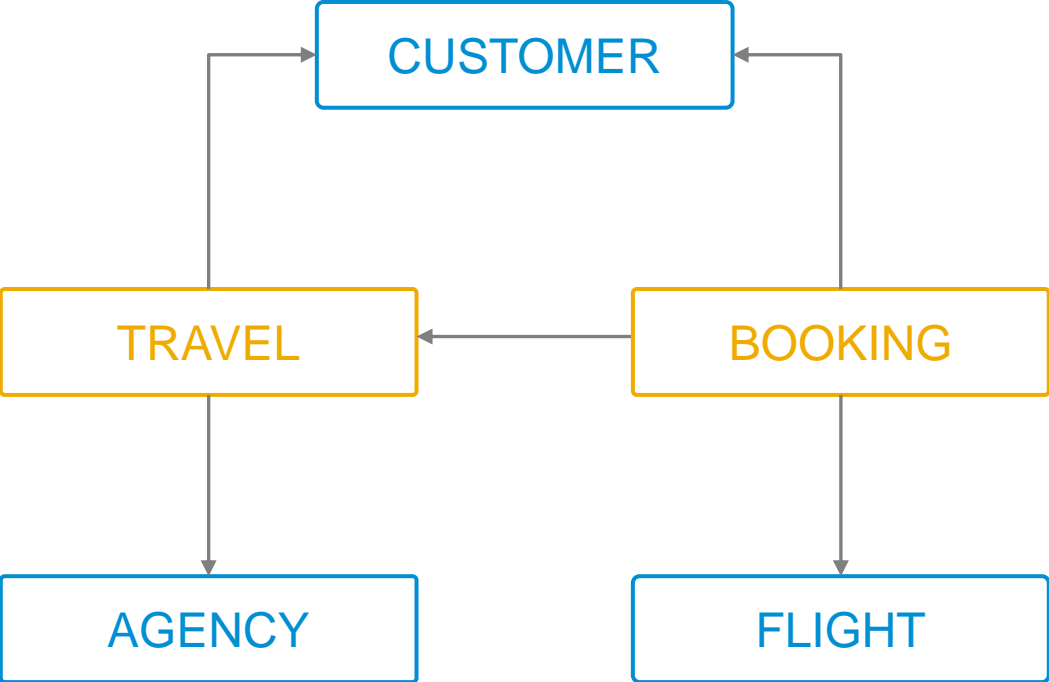
Creating the Business Object Projection

**05**

Building and Previewing the OData UI Service

The Business Scenario

# Simplified flight data model for this openSAP course

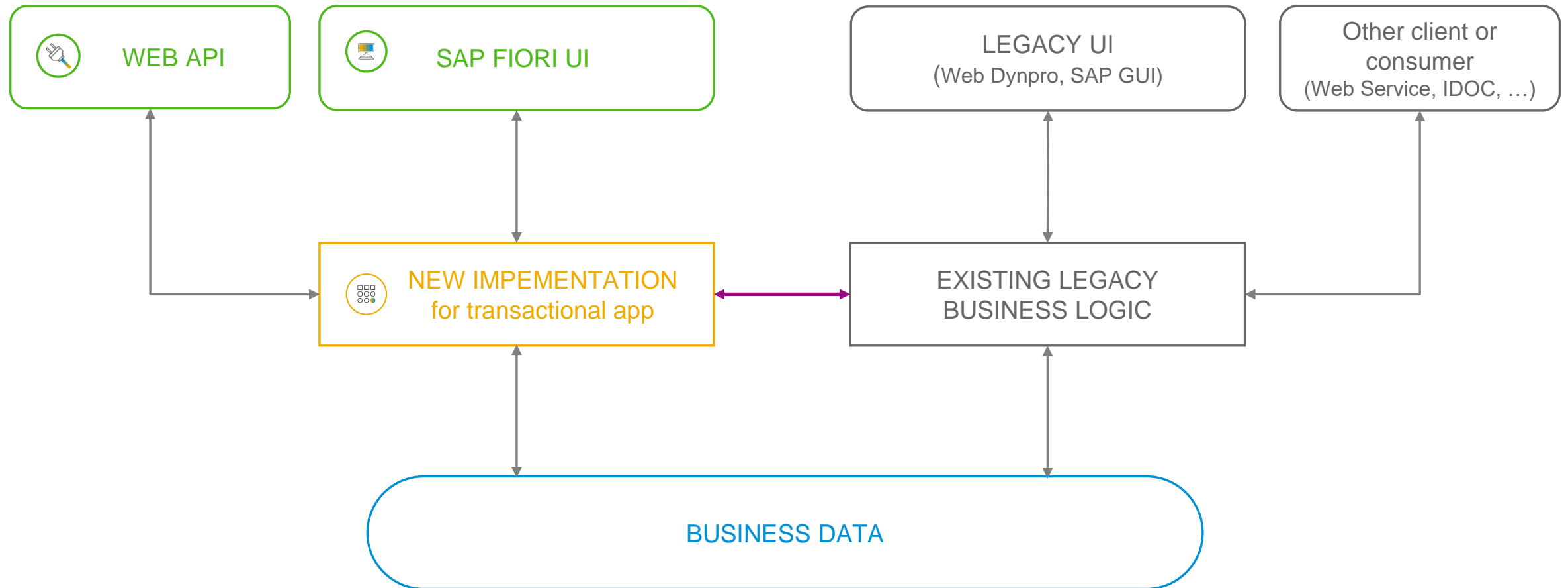


- Main business entities in current scenario
- Secondary business entities in current scenario

Travel	A Travel entity defines general travel data, such as the agency ID or customer ID, status of the travel booking, and the price of travel. The travel data is stored in the database table <code>/DMO/I_TRAVEL</code> .
Booking	The booking data is stored in the database table <code>/DMO/BOOKING</code> . The flight data model defines a 1:n cardinality between the Travel and the Booking entity.
Agency	An Agency entity defines travel agency data, such as the address and contact data. The corresponding data is stored in the database table <code>/DMO/AGENCY</code> . The flight data model defines a 1:n cardinality between Agency and Travel.
Flight	The concrete flight data for each connection is stored in the database table <code>/DMO/FLIGHT</code> . The flight data model defines a 1:n cardinality between the Connection and the Flight entity.
Customer	A Customer entity provides a detailed description of a flight customer (passenger) such as the name, the address, and contact data. The corresponding data is stored in the database table <code>/DMO/CUSTOMER</code> . The flight data model defines a 1:n cardinality between Customer and Travel

# The Business Scenario

## The unmanaged scenario



The Business Scenario

Resulting SAP Fiori elements app

Transactional List Report Travel App

Standard

Search

Adapt Filters

Travels (4,136) Standard

Create Delete

	Travel ID	Agency ID	Customer ID	Starting Date	End Date	Booking Fee	Total Price	Travel Status	
	4136	70025	629	Mar 11, 2021	Mar 11, 2021	20.00 USD	7,424.00 USD	N	
	4135	70014	438	Mar 11, 2021	Mar 11, 2021	10.00 USD	3,713.00 USD	N	
	4134	70028	115	Mar 11, 2021	Mar 11, 2021	20.00 USD	7,375.00 USD	B	
	4133	70043	317	Mar 11, 2021	Mar 11, 2021	20.00 USD	7,391.00 USD	P	
	4132	70007	394	Mar 11, 2021	Mar 11, 2021	20.00 USD	7,346.00 USD	N	
	4131	70029	185	Mar 11, 2021	Mar 11, 2021	10.00 USD	3,633.00 USD	B	
	4130	70006	272	Mar 11, 2021	Mar 11, 2021	30.00 USD	11,008.00 USD	N	
	4129	70048	481	Mar 11, 2021	Mar 11, 2021	30.00 USD	10,797.00 USD	P	
	4128	70009	369	Mar 11, 2021	Mar 11, 2021	20.00 USD	7,101.00 USD	N	

Search

Create

Delete

Object Page

Update

Save

Travel

4136

Edit Delete

Travel

Booking

Travel ID: 4136

Starting Date: Mar 11, 2021

Total Price: 7,424.00 USD

Agency ID: 70025

End Date: Mar 11, 2021

Description: Vacation to USA

Customer ID: 629

Booking Fee: 20.00 USD

Travel Status: N

Booking

Search

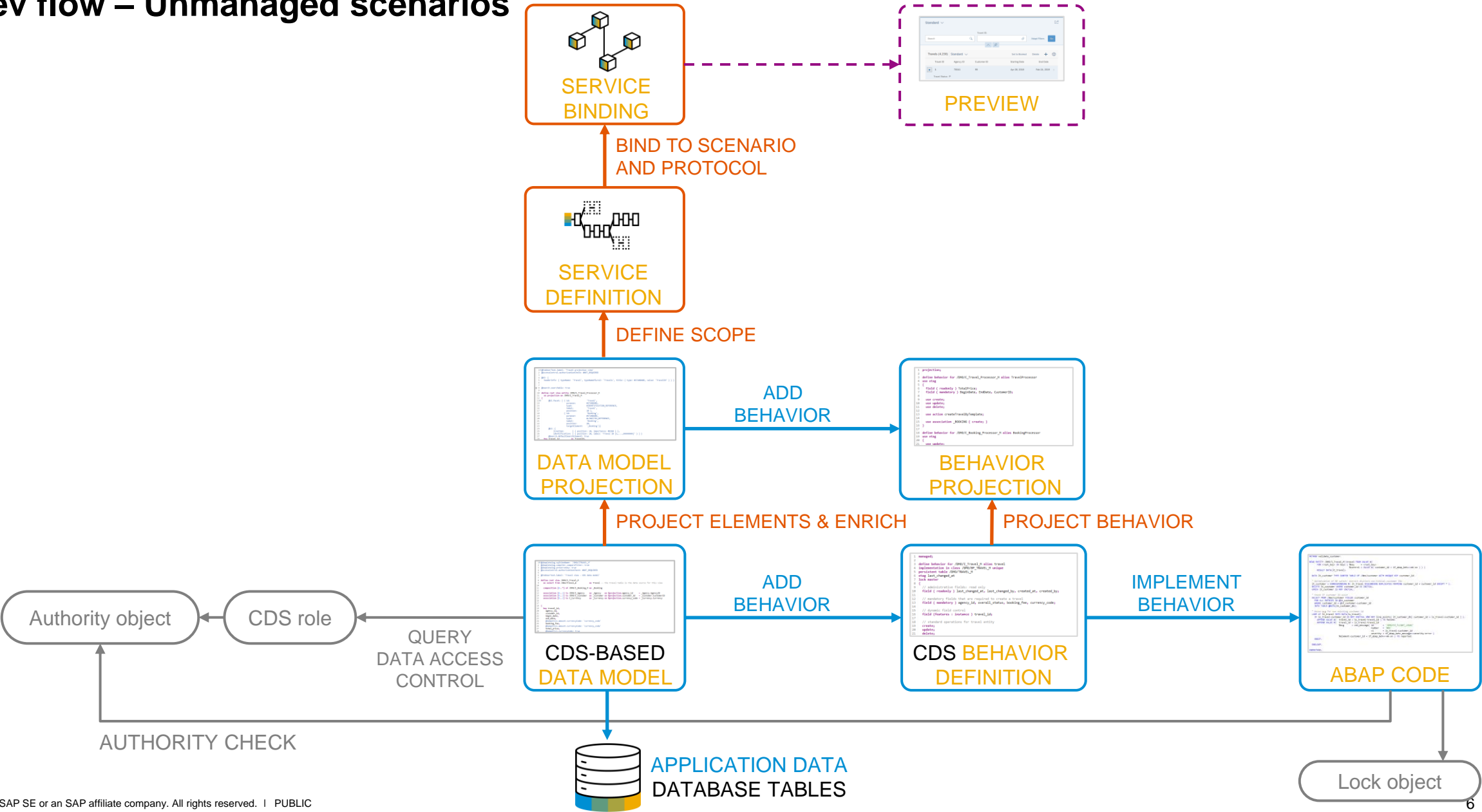
Create Delete

	Booking Number	Booking Date	Customer ID	Airline ID	Flight Number	Flight Date	Flight Price	
	1	Mar 4, 2021	629	AA	18	Mar 11, 2021	3,698.00 USD	
	2	Mar 4, 2021	683	AA	18	Mar 11, 2021	3,698.00 USD	

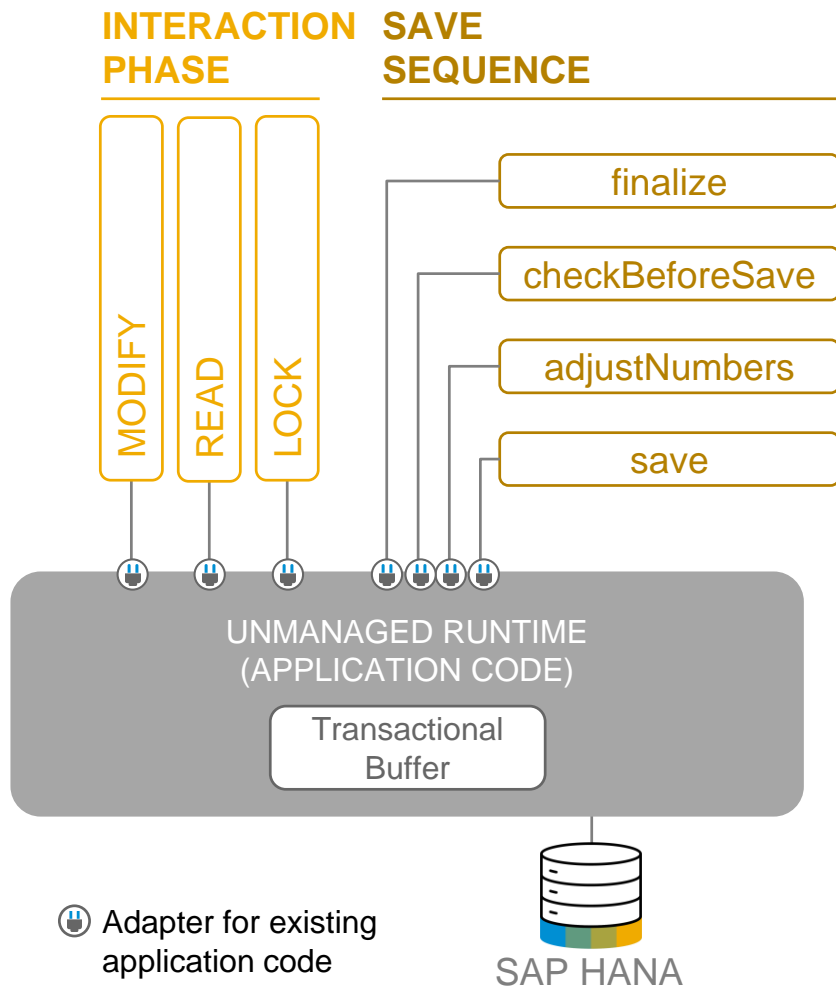
Save Cancel

The Business Scenario

# Dev flow – Unmanaged scenarios



## Unmanaged BO runtime implementation



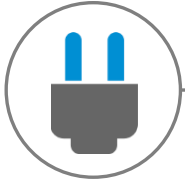
### Application code

- Already available for interaction phase, transactional buffer, and save sequence
- Existing code decoupled from UI and protocol technologies
- Developers fully in charge of BO runtime, i.e. CRUD operations
- Adapters needed to integrate the existing code

### Examples

- Sales Order, Purchase Order

## Unmanaged BO runtime implementation – Adapter for existing application code



```
METHOD create.  
  DATA messages    TYPE /dmo/t_message.  
  DATA legacy_entity_in  TYPE /dmo/travel.  
  DATA legacy_entity_out TYPE /dmo/travel.  
  LOOP AT entities ASSIGNING FIELD-SYMBOL(<entity>).  
    legacy_entity_in = CORRESPONDING #( <entity> MAPPING FROM ENTITY USING CONTROL ).  
    CALL FUNCTION '/DMO/FLIGHT_TRAVEL_CREATE'  
      EXPORTING  
        is_travel    = CORRESPONDING /dmo/s_travel_in( legacy_entity_in )  
      IMPORTING  
        es_travel    = legacy_entity_out  
        et_messages  = messages.  
    IF messages IS INITIAL.  
      APPEND VALUE #( %cid = <entity>-%cid travelid = legacy_entity_out-travel_id )  
        TO mapped-travel.  
    ELSE.  
      "error handling  
      ...  
    ENDIF.  
  ENDLOOP.  
ENDMETHOD.
```



# The Business Scenario

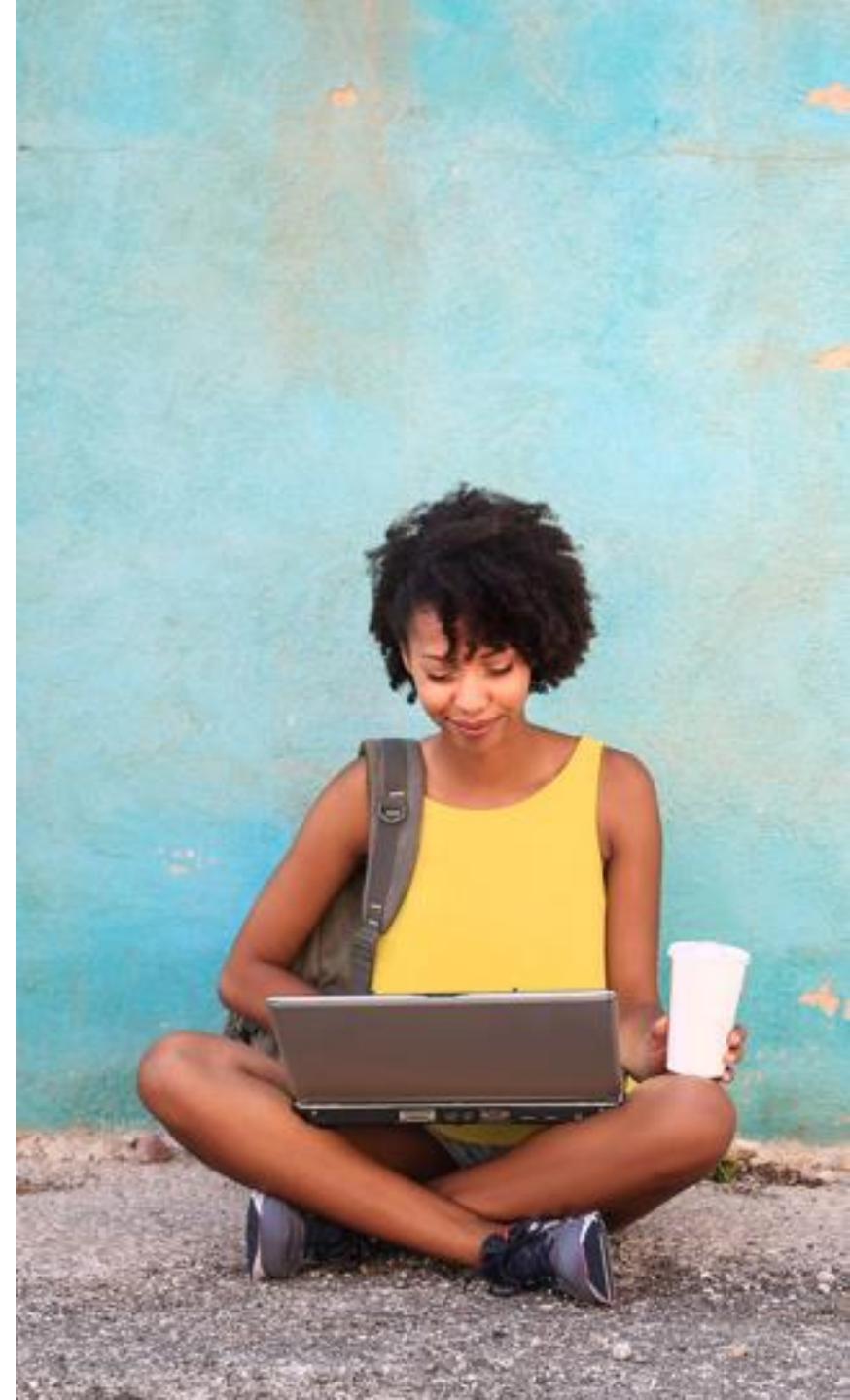
## Wrap-up

### IN THIS UNIT, YOU LEARNED

- How the unmanaged implementation scenario for this week differs from the scenarios from the last units
- How RAP supports this new scenario

### NEXT UNIT

- Week 4 – Unit 2  
Creating the CDS Data Model





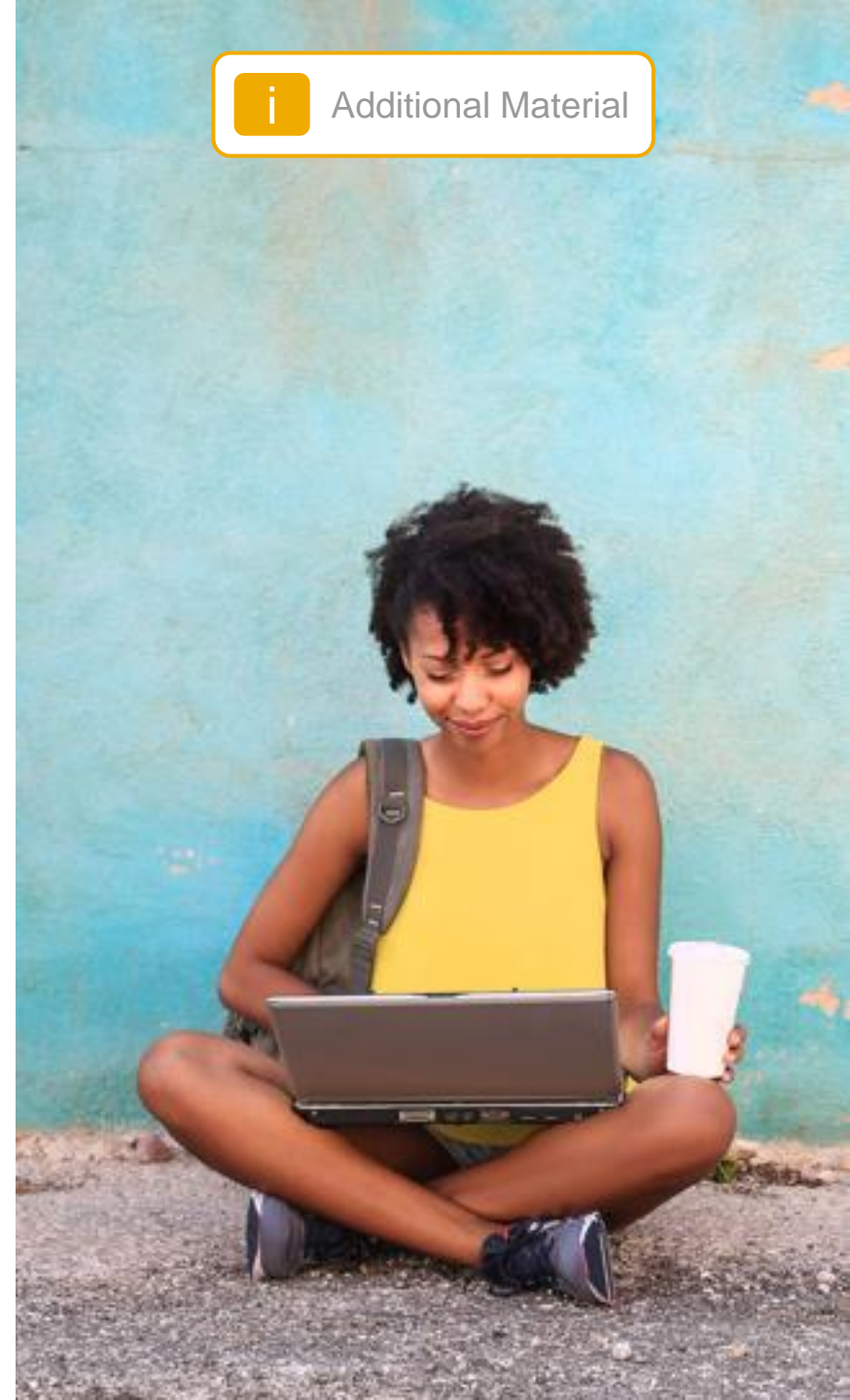
## ABAP RESTful APPLICATION PROGRAMMING MODEL INFORMATION PAGE

---

For more information, links to documentation, tutorials, and more, please visit the RAP at openSAP information page by following the link below

[RAP at openSAP information page \(week 4\)](#)

---



# Thank you.

**Contact information:**

**open@sap.com**

Follow all of SAP



[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.



Week 4: Dealing with Existing Code

## Unit 2: Creating the CDS Data Model

# Creating the CDS Data Model

## Topics

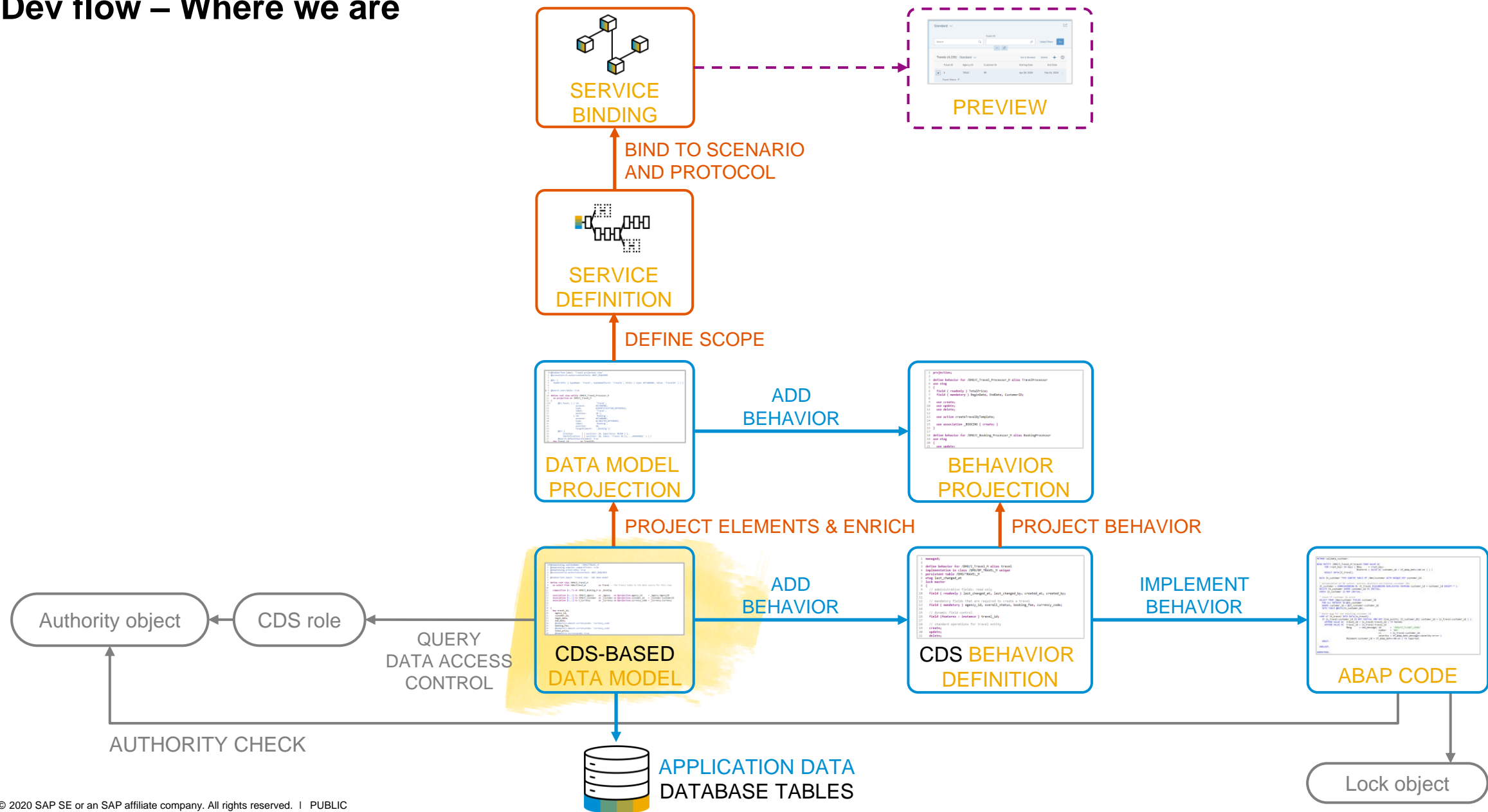
### Week 4

- 01 The Business Scenario
- 02 Creating the CDS Data Model
- 03 Defining and Implementing the Business Object Behavior
- 04 Creating the Business Object Projection
- 05 Building and Previewing the OData UI Service



# Creating the CDS Data Model

## Dev flow – Where we are



# Creating the CDS Data Model

## What we will achieve in this unit

Root  
key word

Composition

Associations

Field list  
with aliases

Data  
preview (F8)

```
1 @EndUserText.label: 'Travel data'
2 @Metadata.ignorePropagatedAnnotations: true
3
4 define root view entity ZI_RAP_Travel_U_1234
5   as select from /dmo/travel
6
7   composition [0..*] of ZI_RAP_Booking_U_1234 as _Booking
8
9   association [0..1] to /DMO/I_Agency      as _Agency on $projection.AgencyID = _Agency.AgencyID
10  association [0..1] to /DMO/I_Customer   as _Customer on $projection.CustomerID = _Customer.CustomerID
11  association [0..1] to I_Currency        as _Currency on $projection.CurrencyCode = _Currency.Currency
12
13  {
14    key travel_id      as TravelID,
15        agency_id     as AgencyID,
16        customer_id   as CustomerID,
17        begin_date    as BeginDate,
18        end_date      as EndDate,
19        @Semantics.amount.currencyCode: 'CurrencyCode'
20        booking_fee    as BookingFee,
21        @Semantics.amount.currencyCode: 'CurrencyCode'
22        total_price    as TotalPrice,
23        currency_code as CurrencyCode,
24        description    as Description,
25        status         as Status,
26        @Semantics.user.createdBy: true
27        createdby      as Createdby,
28        @Semantics.systemDateTime.createdAt: true
29        createdat      as Createdat,
30        @Semantics.user.lastChangedBy: true
31        lastchangedby  as Lastchangedby,
32        @Semantics.systemDateTime.lastChangedAt: true
33        lastchangedat  as Lastchangedat,
34
35    /* associations */
36    _Booking,
37    _Agency,
38    _Customer,
39    _Currency
40  }
41
```

ZI\_RAP\_TRAVEL\_U\_1234

ZI\_RAP\_TRAVEL\_U\_1234

Raw Data

Filter pattern

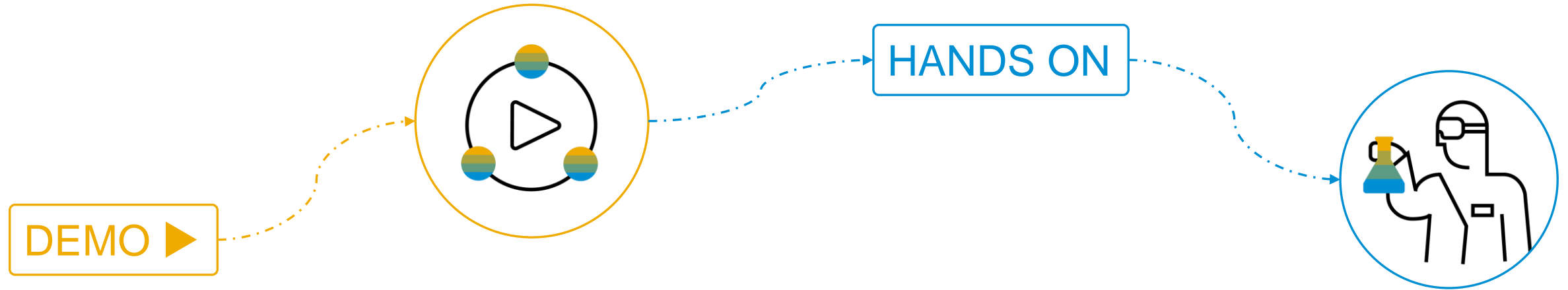
100 rows retrieved - 14 ms (partial result)

AB	TravelID	AB	AgencyID	AB	CustomerID	<div></div> BeginDate	<div></div> EndDate	12	BookingFee
	00000001		070041		000594	2020-05-09	2021-03-07		40.00
	00000002		070007		000608	2020-05-09	2020-05-09		20.00
	00000003		070046		000093	2020-05-09	2021-03-07		80.00
	00000004		070042		000665	2020-05-09	2021-03-07		40.00
	00000005		070007		000225	2020-05-09	2020-05-09		20.00
	00000006		070049		000072	2020-05-09	2021-03-07		120.00
	00000007		070046		000138	2020-05-09	2021-03-07		120.00
	00000008		070012		000705	2020-05-09	2020-05-11		60.00
	00000009		070032		000115	2020-05-09	2021-03-07		120.00



# Creating the CDS Data Model

## Demo



---

## Creating the CDS Data Model

---

1. Define the BO composition tree for the Travel and Booking entities
2. Preview the existing data

# Creating the CDS Data Model

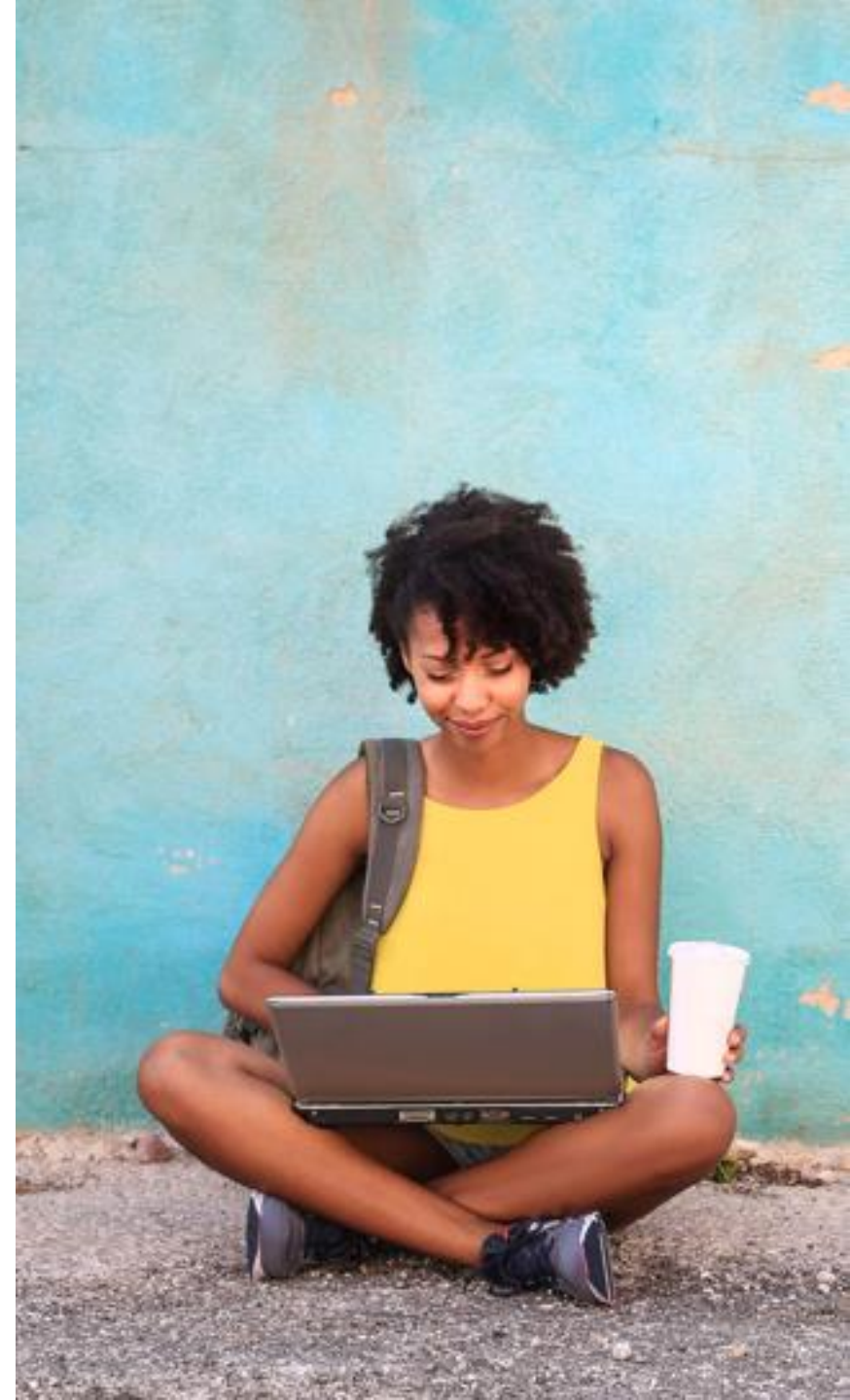
## Wrap-up

### IN THIS UNIT, YOU LEARNED

- What a BO composition tree is and how you define it

### NEXT UNIT

- Week 4 – Unit 3  
Defining and Implementing the Business Object Behavior





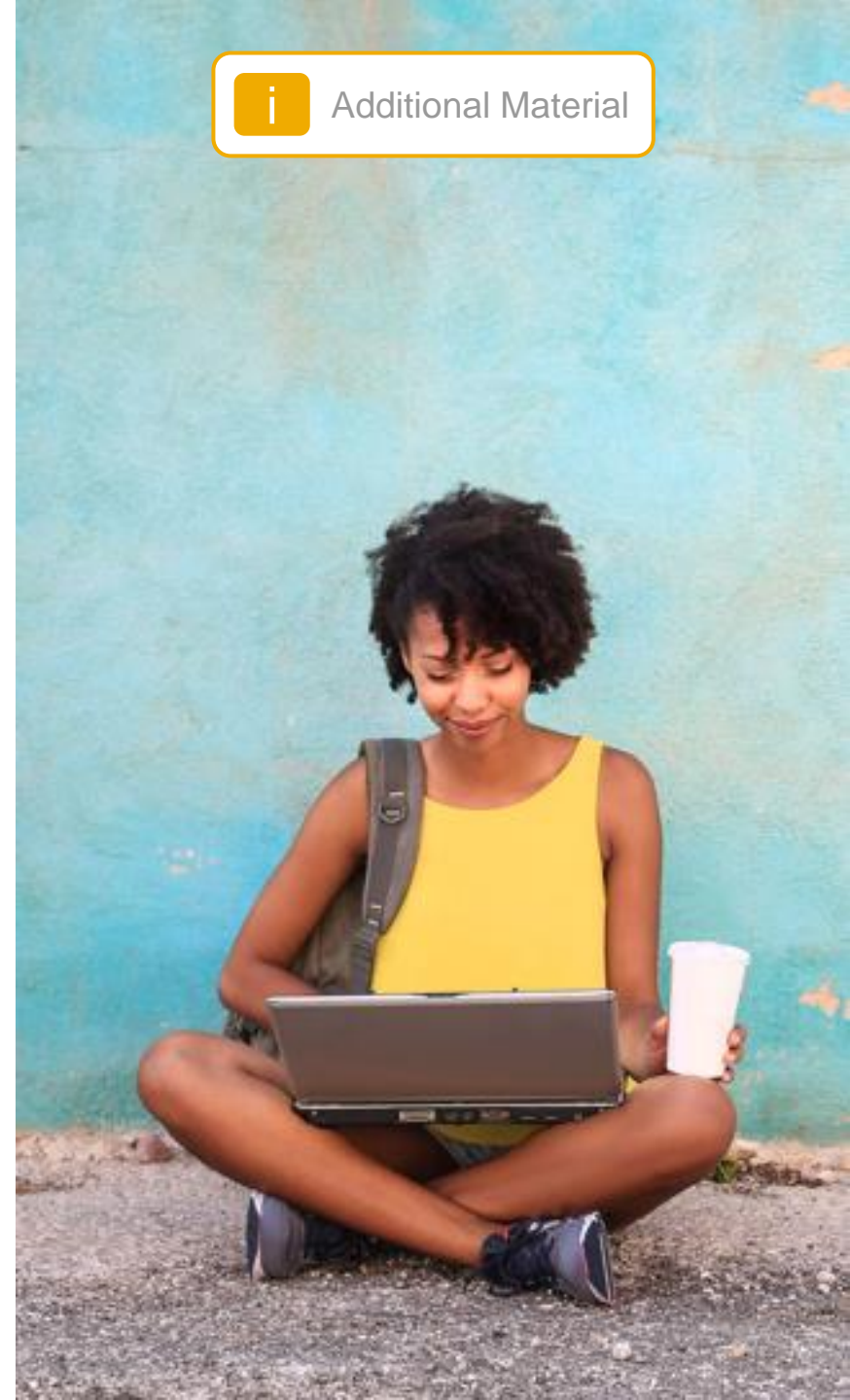
## ABAP RESTful APPLICATION PROGRAMMING MODEL INFORMATION PAGE

---

For more information, links to documentation, tutorials, and more, please visit the RAP at openSAP information page by following the link below

[RAP at openSAP information page \(week 4\)](#)

---



# Thank you.

**Contact information:**

**open@sap.com**

Follow all of SAP



[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.



Week 4: Dealing with Existing Code

## Unit 3: Defining and Implementing the Business Object Behavior

# Defining and Implementing the Business Object Behavior

## Topics

### Week 4

- 01** The Business Scenario
- 02** Creating the CDS Data Model
- 03** Defining and Implementing the Business Object Behavior
- 04** Creating the Business Object Projection
- 05** Building and Previewing the OData UI Service



# Defining and Implementing the Business Object Behavior

## What we will achieve in this unit – Behavior Definition

Implementation  
type unmanaged

Behaviour  
implementation  
classes

Standard  
operations:  
create, update  
and delete

Create enabled  
association

Specify fields that  
are mandatory or  
read-only

central mapping  
for legacy  
business logic

```
[Y08] ZI_RAP_TRAVEL_U_1234
1  unmanaged;
2
3=define behavior for ZI_RAP_Travel_U_1234 alias Travel
4  implementation in class zbp_i_rap_travel_u_1234 unique
5  lock master
6  etag master Lastchangedat
7  {
8    create;
9    update;
10   delete;
11   association _Booking { create; }
12
13   field ( read only ) TravelID;
14   field ( mandatory ) AgencyID, CustomerID, BeginDate, EndDate;
15
16= mapping for /DMO/TRAVEL control zsrp_travel_x_1234
17 {
18   TravelId = travel_id;
19   AgencyId = AGENCY_ID;
20   CustomerId = CUSTOMER_ID;
21   BeginDate = BEGIN_DATE;
22   EndDate = END_DATE;
23   BookingFee = BOOKING_FEE;
24   TotalPrice = TOTAL_PRICE;
25   CurrencyCode = CURRENCY_CODE;
26   Description = DESCRIPTION;
27   Status = STATUS;
28   Createdby = CREATEDBY;
29   Createdat = CREATEDAT;
30   Lastchangedby = LASTCHANGEDBY;
31   Lastchangedat = LASTCHANGEDAT;
32 }
33 }
```

```
35=define behavior for ZI_RAP_Booking_U_1234 alias Booking
36 implementation in class zbp_i_rap_booking_u_1234 unique
37 lock dependent by _Travel
38 etag dependent by _Travel
39 {
40   update;
41   delete;
42   association _Travel;
43
44   field ( read only ) TravelID, BookingID;
45   field ( mandatory ) BookingDate, CustomerID, CarrierId, ConnectionID, FlightDate;
46
47= mapping for /DMO/BOOKING control zsrp_booking_x_1234
48 {
49   TravelId = TRAVEL_ID;
50   BookingId = BOOKING_ID;
51   BookingDate = BOOKING_DATE;
52   CustomerId = CUSTOMER_ID;
53   CarrierId = CARRIER_ID;
54   ConnectionId = CONNECTION_ID;
55   FlightDate = FLIGHT_DATE;
56   FlightPrice = FLIGHT_PRICE;
57   CurrencyCode = CURRENCY_CODE;
58 }
59
60 }
```



## Defining and Implementing the Business Object Behavior

### What we will achieve in this unit – Behavior Implementation

Local handler  
class for buffer  
access

Implementation of  
standard  
operations create,  
update and delete

Implementation  
for read from  
buffer

Create by  
association

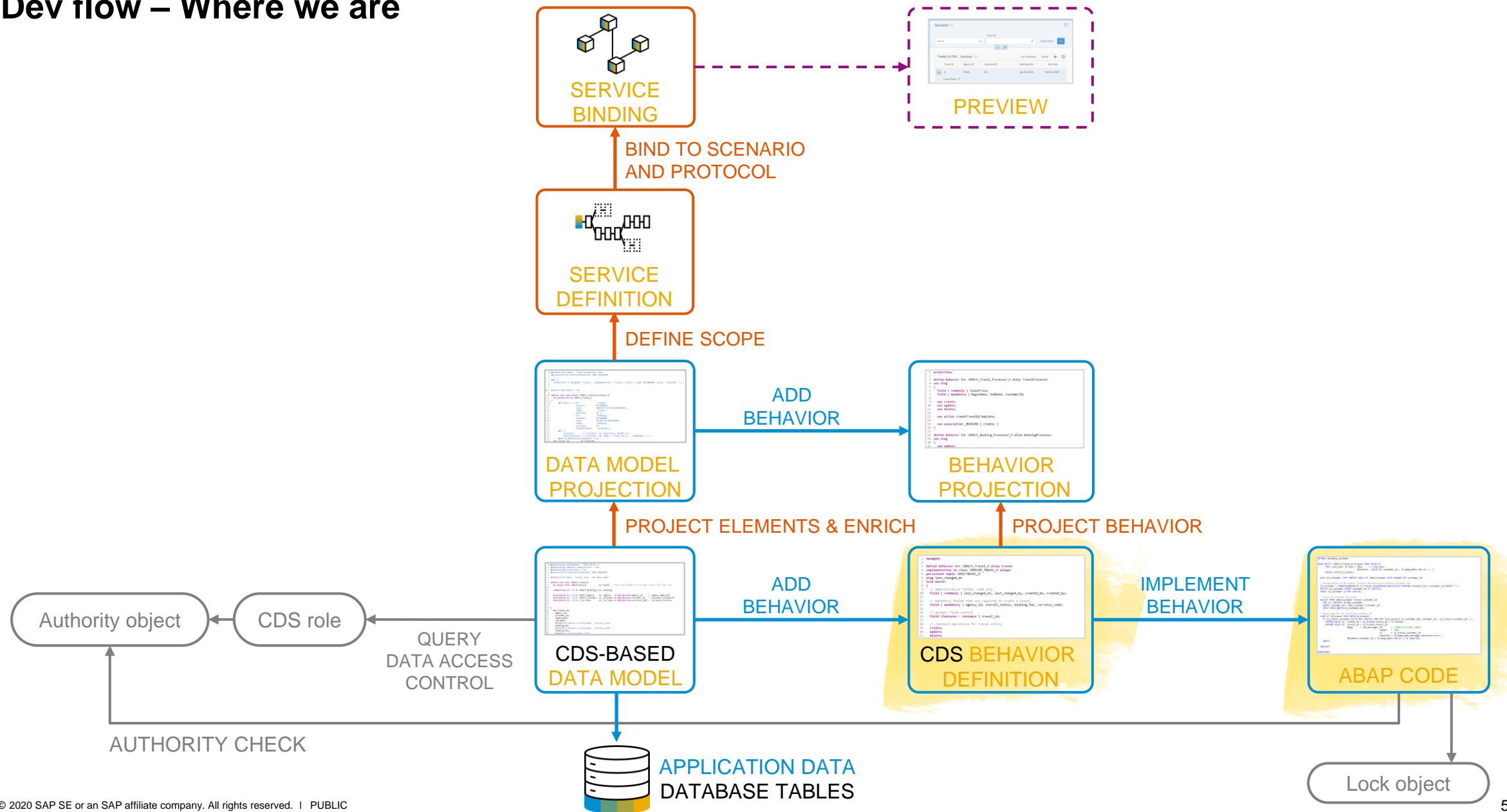
Local saver class  
to save data to  
database tables

```
[Y08] ZBP_I_RAP_TRAVEL_U_1234
  ZBP_I_RAP_TRAVEL_U_1234 > LHC_TRAVEL >
1= CLASS lhc_Travel DEFINITION INHERITING FROM cl_abap_behavior_handler.
2   PRIVATE SECTION.
3
4   METHODS create FOR MODIFY
5     IMPORTING entities FOR CREATE Travel.
6
7   METHODS delete FOR MODIFY
8     IMPORTING keys FOR DELETE Travel.
9
10  METHODS update FOR MODIFY
11    IMPORTING entities FOR UPDATE Travel.
12
13  METHODS lock FOR LOCK
14    IMPORTING keys FOR LOCK Travel.
15
16  METHODS read FOR READ
17    IMPORTING keys FOR READ Travel RESULT result.
18
19  METHODS cba_Booking FOR MODIFY
20    IMPORTING entities_cba FOR CREATE Travel\_Booking.
21
22  METHODS rba_Booking FOR READ
23    IMPORTING keys_rba FOR READ Travel\_Booking FULL result_requested RESULT result
24
25  ENDClass.
```

```
410
411= CLASS lsc_ZI_RAP_Travel_U_1234 DEFINITION INHERITING FROM cl_abap_behavior_saver.
412   PROTECTED SECTION.
413
414   METHODS check_before_save REDEFINITION.
415
416   METHODS finalize          REDEFINITION.
417
418   METHODS save              REDEFINITION.
419
420  ENDClass.
```

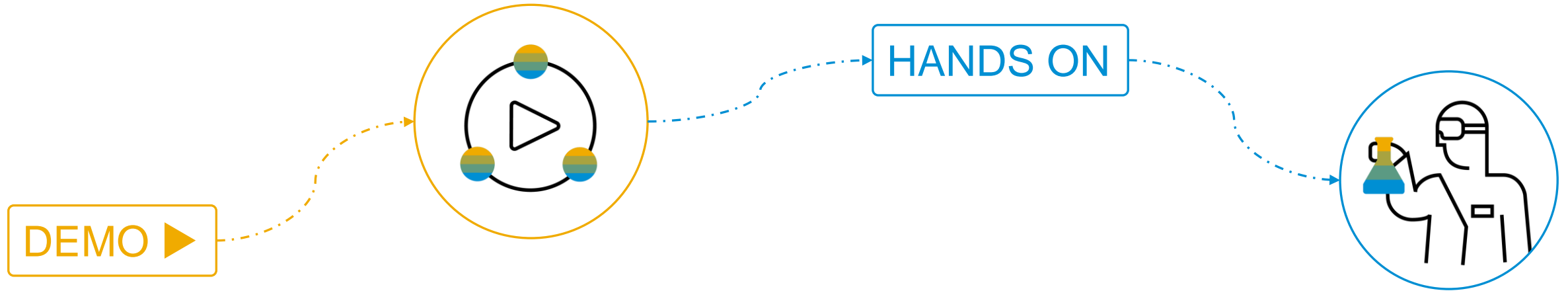
# Defining and Implementing the Business Object Behavior

## Dev flow – Where we are



# Defining and Implementing the Business Object Behavior

## Demo



---

### Create and implement the transactional BO behavior

---

1. Create and define BO behavior definition for the Travel and Booking entities
2. Implement the BO behavior for the Travel and Booking entities
3. Create a unit test to test the BO implementation via EML

# Defining and Implementing the Business Object Behavior

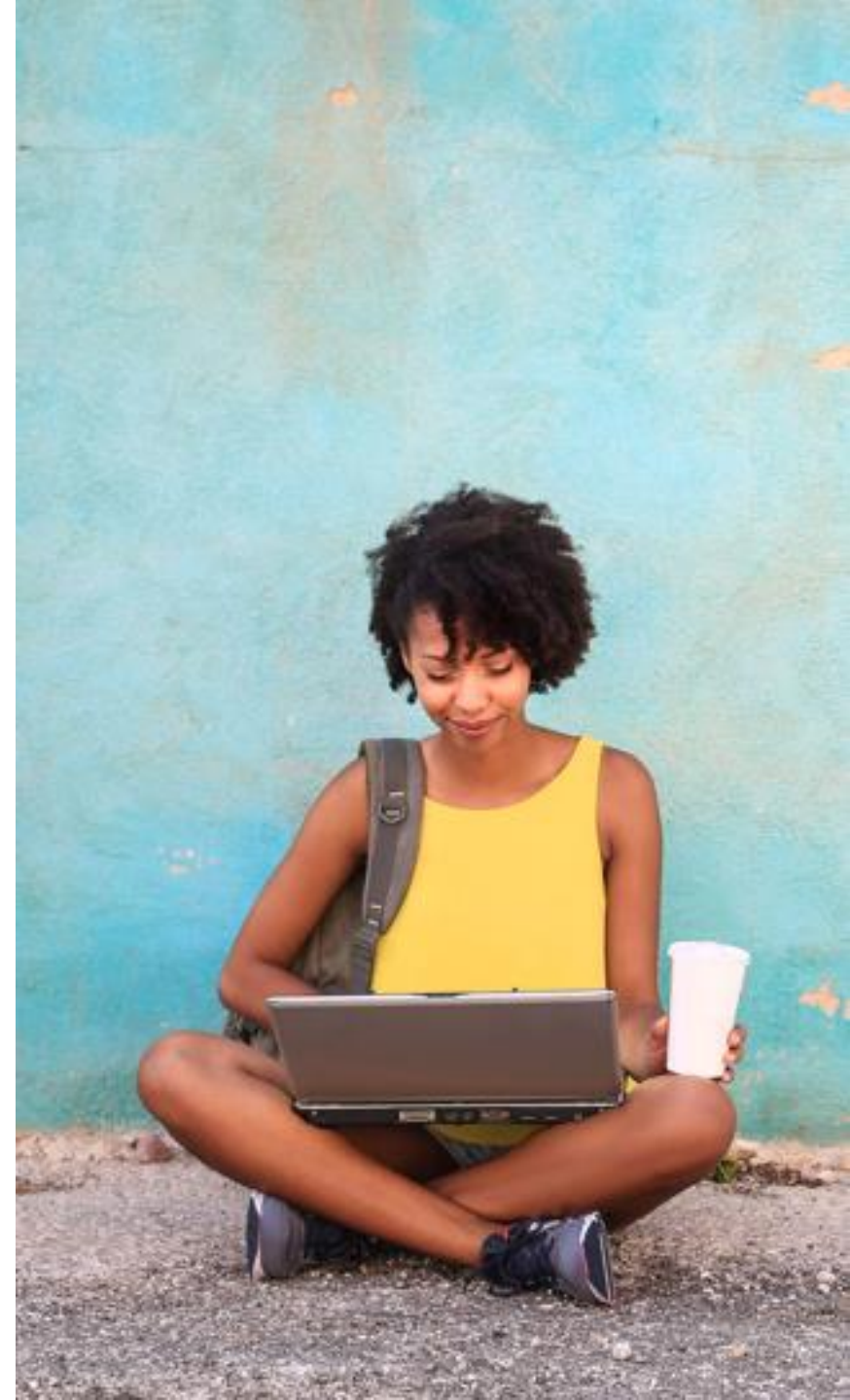
## Wrap-up

### IN THIS UNIT, YOU LEARNED

- How to define the BO behavior definition for the Travel and Booking entities and how to implement it

### NEXT UNIT

- Week 4 – Unit 4  
Creating the Business Object Projection



# Defining and Implementing the Business Object Behavior

## Further reading



Additional Material

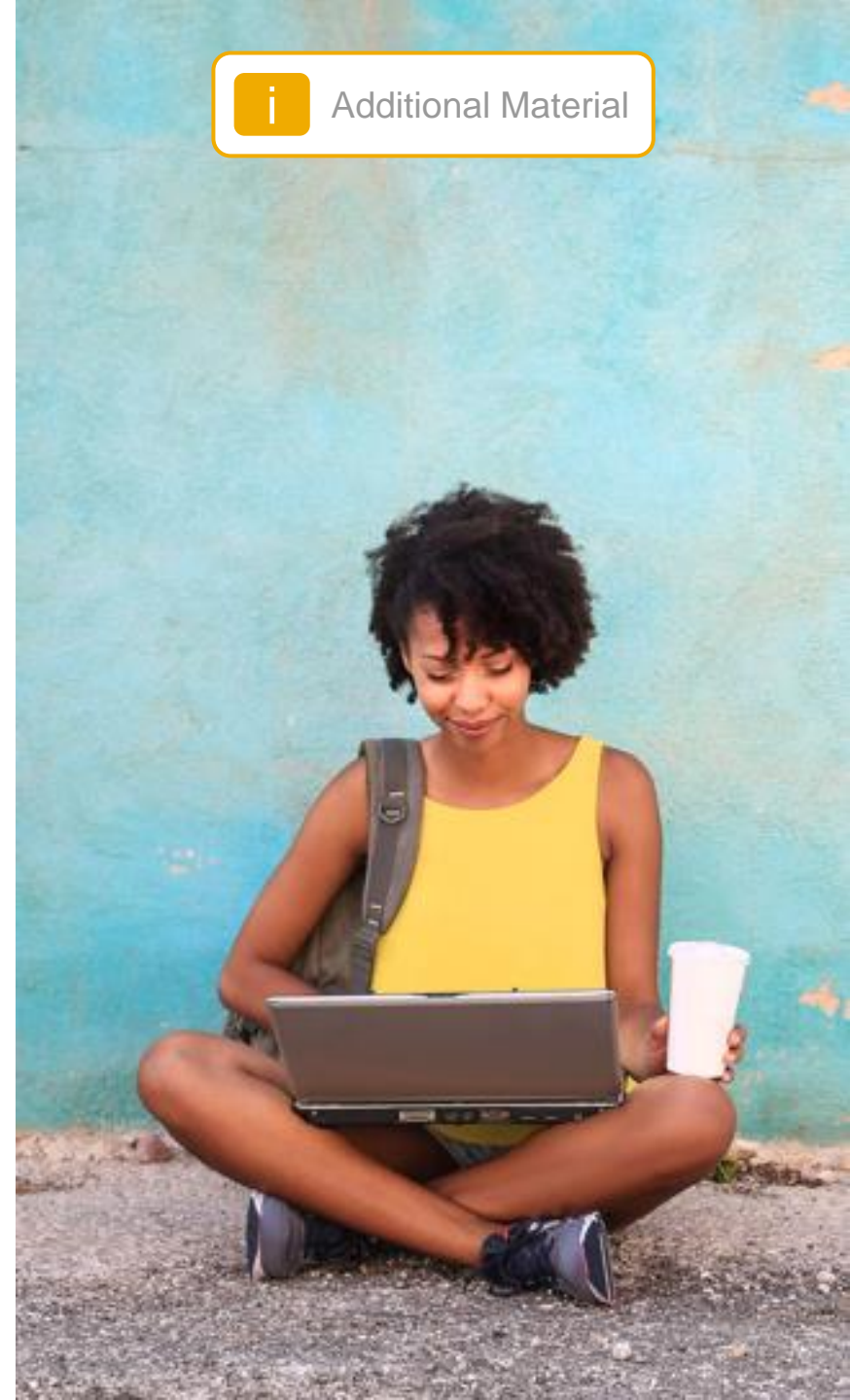
### ABAP RESTful APPLICATION PROGRAMMING MODEL INFORMATION PAGE

---

For more information, links to documentation, tutorials, and more, please visit the RAP at openSAP information page by following the link below

[RAP at openSAP information page \(week 4\)](#)

---



# Thank you.

**Contact information:**

**open@sap.com**

Follow all of SAP



[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.





Week 4: Dealing with Existing Code

## Unit 4: Creating the Business Object Projection



# Creating the Business Object Projection

## Topics

### Week 4

- 01** The Business Scenario
- 02** Creating the CDS Data Model
- 03** Defining and Implementing the Business Object Behavior
- 04** Creating the Business Object Projection
- 05** Building and Previewing the OData UI Service

# Creating the Business Object Projection

## What we will achieve in this unit – Data Model Projection

Enable Search  
@Search.searchable:  
true

Allow use of  
MetaDataExtensions

Define value help  
@Consumption.  
valueHelpDefinition

Define association as  
redirected to  
composition child

Define association as  
redirected to parent

```
[Y08] ZC_RAP_TRAVEL_U_1234
1=@EndUserText.label: 'Travel data'
2 @AccessControl.authorizationCheck: #CHECK
3 @Search.searchable: true
4 @Metadata.allowExtensions: true
5
6 define root view entity ZC_RAP_Travel_U_1234
7   as projection on ZI_RAP_Travel_U_1234
8   {
9     //ZI_RAP_TRAVEL_U_1234
10    key TravelID,
11*    @Consumption.valueHelpDefinition: [ { entity: { name: '/DMO/I_Agency', element: 'AgencyID' } } ]
12    @Search.defaultSearchElement: true
13    AgencyID,
14*    @Consumption.valueHelpDefinition: [ { entity: { name: '/DMO/I_Customer', element: 'CustomerID' } } ]
15    @Search.defaultSearchElement: true
16    CustomerID,
17    BeginDate,
18    EndDate,
19    @Semantics.amount.currencyCode: 'CurrencyCode'
20    BookingFee,
21    @Semantics.amount.currencyCode: 'CurrencyCode'
22    TotalPrice,
23    @Consumption.valueHelpDefinition: [ { entity: { name: 'I_Currency', element: 'Currency' } } ]
24    CurrencyCode,
25    Description,
26    Status,
27    Createdby,
28    Createdat,
29    Lastchangedby,
30    Lastchangedat,
31
32*    /* Associations */
33    //ZI_RAP_TRAVEL_U_1234
34    _Agency,
35    _Booking : redirected to composition child ZC_RAP_Booking_U_1234,
36    _Currency,
37    _Customer
38  }
39
```

```
[Y08] ZC_RAP_BOOKING_U_1234
1=@EndUserText.label: 'Booking'
2 @AccessControl.authorizationCheck: #CHECK
3 @Search.searchable: true
4 @Metadata.allowExtensions: true
5
6 define view entity ZC_RAP_Booking_U_1234
7   as projection on ZI_RAP_Booking_U_1234
8   {
9     //ZI_RAP_BOOKING_U_1234
10    key BookingID,
11*    @Consumption.valueHelpDefinition: [ { entity: { name: '/DMO/I_Agency', element: 'AgencyID' } } ]
12    @Search.defaultSearchElement: true
13    AgencyID,
14*    @Consumption.valueHelpDefinition: [ { entity: { name: '/DMO/I_Customer', element: 'CustomerID' } } ]
15    @Search.defaultSearchElement: true
16    CustomerID,
17    BeginDate,
18    EndDate,
19    @Semantics.amount.currencyCode: 'CurrencyCode'
20    BookingFee,
21    @Semantics.amount.currencyCode: 'CurrencyCode'
22    TotalPrice,
23    @Consumption.valueHelpDefinition: [ { entity: { name: 'I_Currency', element: 'Currency' } } ]
24    CurrencyCode,
25    Description,
26    Status,
27    Createdby,
28    Createdat,
29    Lastchangedby,
30    Lastchangedat,
31
32*    /* Associations */
33    //ZI_RAP_BOOKING_U_1234
34    _Carrier,
35    _Customer,
36    _Travel : redirected to parent ZC_RAP_Travel_U_1234
37  }
38
39
```

# Creating the Business Object Projection

## What we will achieve in this unit – Metadata Extension

@UI.headerInfo

@UI.presentation  
Variant  
defines default sort  
order

@UI.facet  
annotations

```
[Y08] ZC_RAP_TRAVEL_U_1234
1* @Metadata.layer: #CUSTOMER
2 @UI: {
3   headerInfo: {
4     typeName: 'Travel',
5     typeNamePlural: 'Travels',
6     title: {
7       type: #STANDARD,
8       label: 'Travel',
9       value: 'TravelID'
10    }
11  },
12  presentationVariant: [ { sortOrder: [ { by: 'TravelID',
13                                         direction: #DESC } ] } ]
14 }
15 annotate view ZC_RAP_Travel_U_1234 with
16 {
17   @UI.facet: [ { id: 'Travel',
18                  purpose: #STANDARD,
19                  type: #IDENTIFICATION_REFERENCE,
20                  label: 'Travel',
21                  position: 10 },
22                { id: 'Booking',
23                  purpose: #STANDARD,
24                  type: #LINEITEM_REFERENCE,
25                  label: 'Booking',
26                  position: 20,
27                  targetElement: '_Booking' } ]
28
29   @UI: { lineItem: [ { position: 10 } ],
30          identification: [ { position: 10 } ] }
31   TravelID;
32
33   @UI: { lineItem: [ { position: 20 } ],
34          identification: [ { position: 20 } ] }
35   AgencyID;
36
37   @UI: { lineItem: [ { position: 30 } ],
38          identification: [ { position: 30 } ] }
39   CustomerID;
40 }
```

### What we will achieve in this unit – Behavior Projection

Define that etag is used

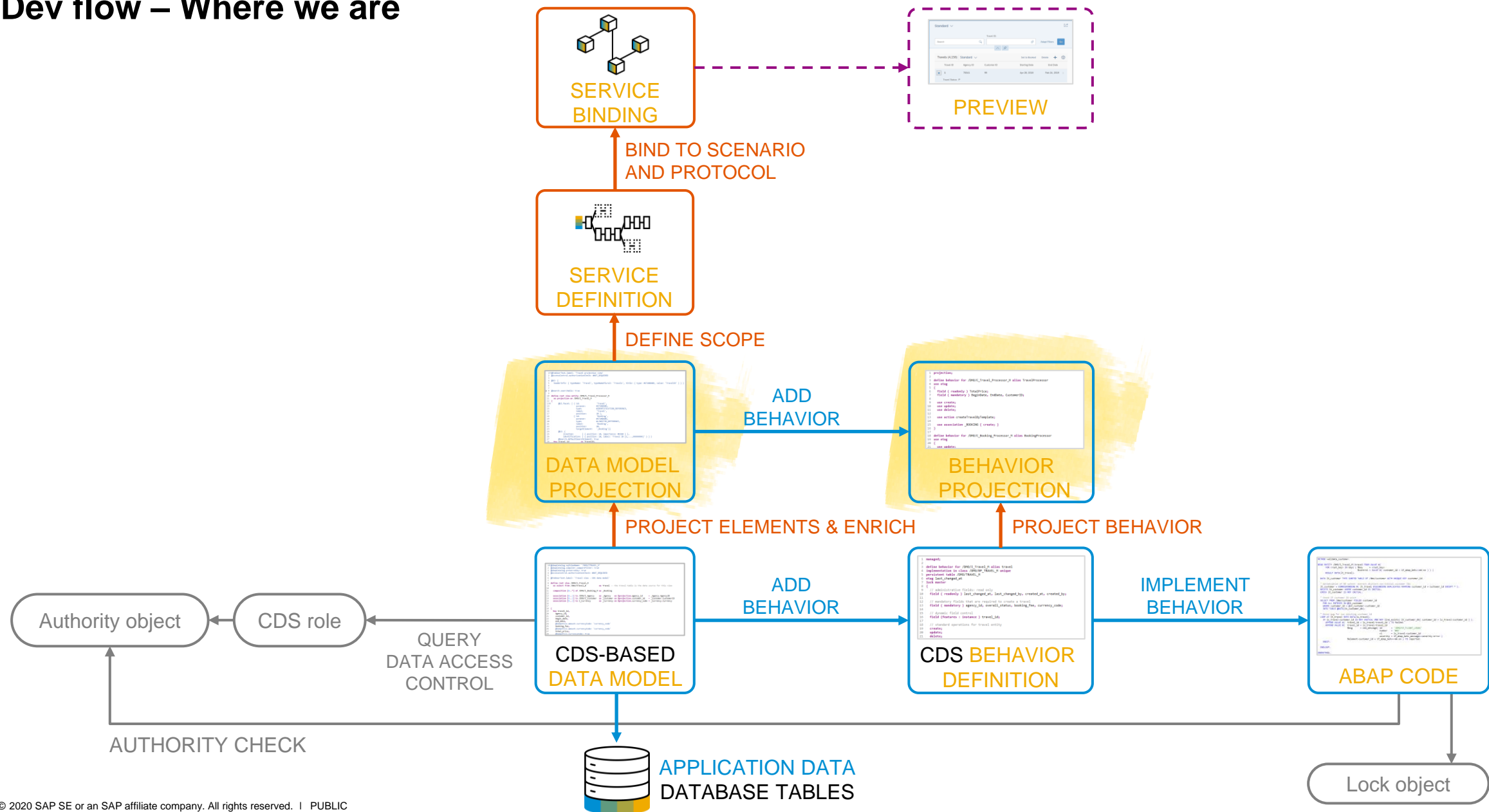
Define which standard operations are used

Define that create enabled association is used

```
[Y08] ZC_RAP_TRAVEL_U_1234 33
1 projection;
2
3 define behavior for ZC_RAP_Travel_U_1234 alias Travel
4 use etag
5 {
6     use create;
7     use update;
8     use delete;
9
10    use association _Booking { create; }
11 }
12
13 define behavior for ZC_RAP_Booking_U_1234 alias Booking
14 use etag
15 {
16     use update;
17     use delete;
18
19     use association _Travel;
20 }
```

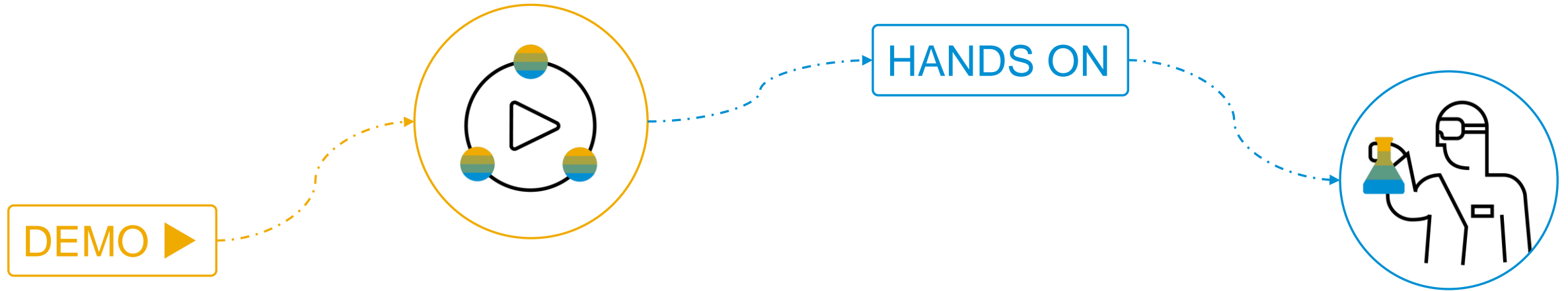
# Creating the Business Object Projection

## Dev flow – Where we are



# Creating the Business Object Projection

## Demo



---

### Define the business object BO projection

---

1. Define the BO data model projection for the Travel and Booking entities
2. Define Metadata Extensions for the BO data model projection
3. Define the BO behavior projection for the Travel and Booking entities

# Creating the Business Object Projection

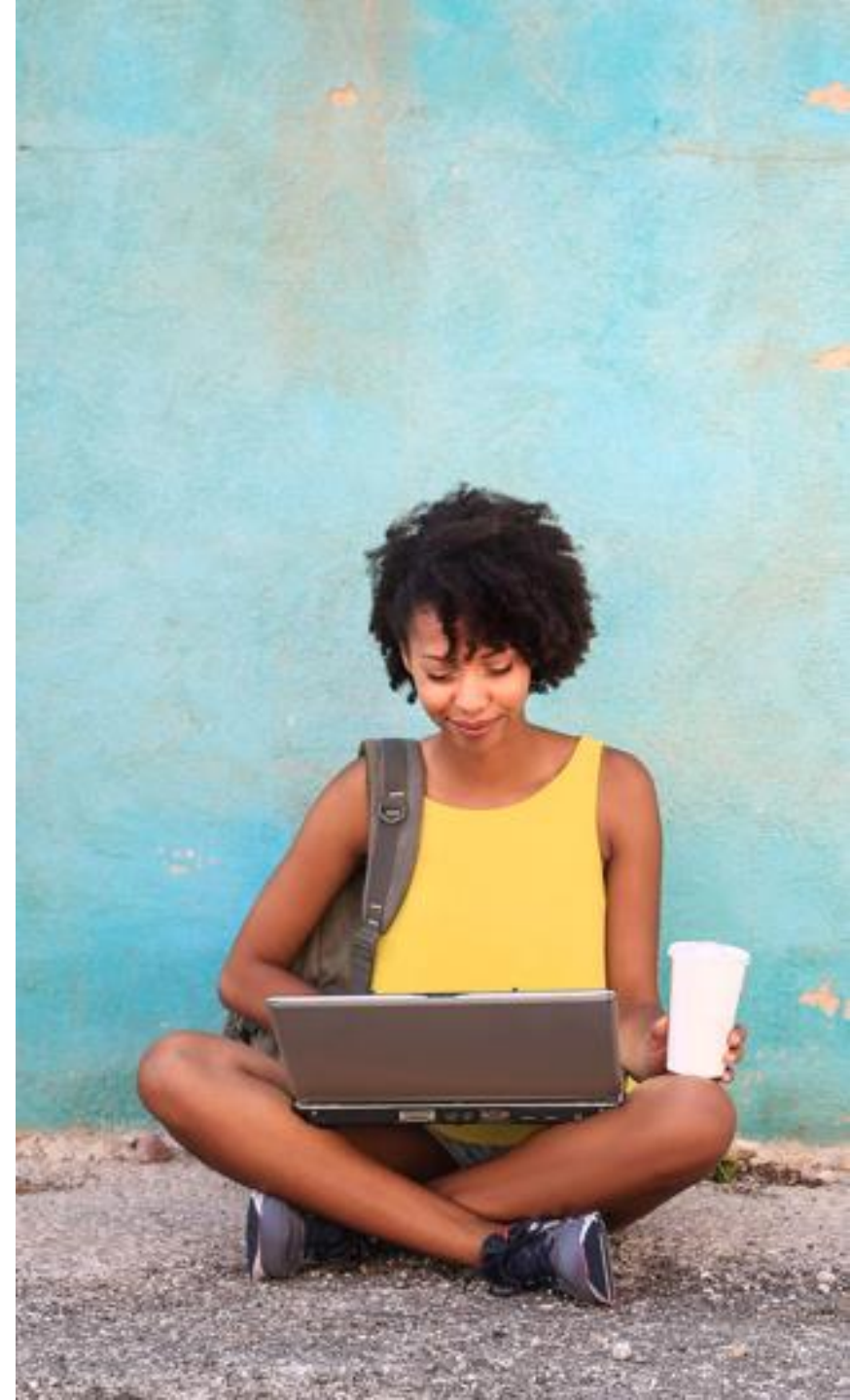
## Wrap-up

### IN THIS UNIT, YOU LEARNED

- How to define the BO data model projection and the BO behavior projection for the Travel and Booking entities

### NEXT UNIT

- Week 4 – Unit 5  
Building and Previewing the OData UI Service





# Creating the Business Object Projection

## Further reading



Additional Material

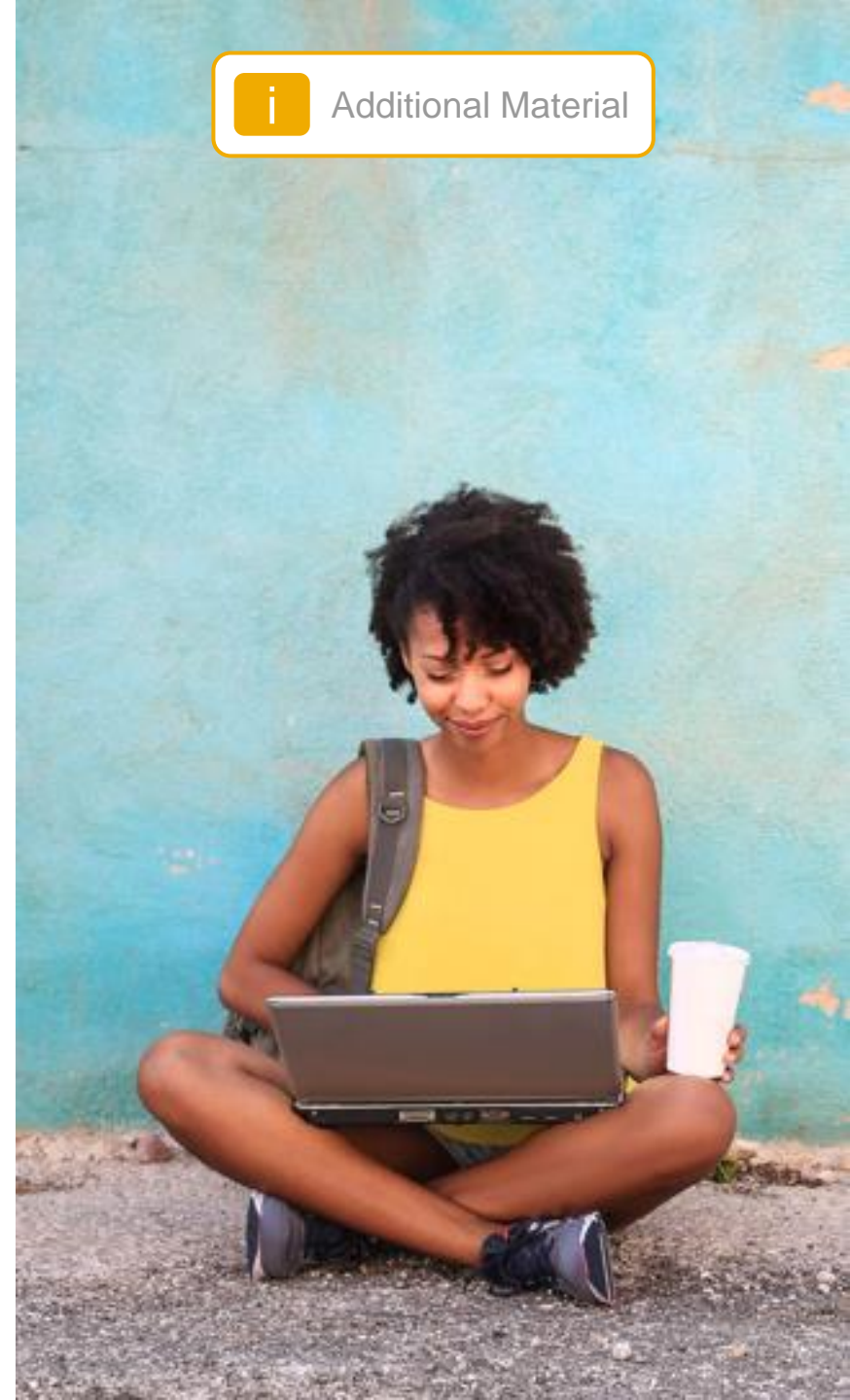
### ABAP RESTful APPLICATION PROGRAMMING MODEL INFORMATION PAGE

---

For more information, links to documentation, tutorials, and more, please visit the RAP at openSAP information page by following the link below

[RAP at openSAP information page \(week 4\)](#)

---





# Thank you.

**Contact information:**

**open@sap.com**

Follow all of SAP



[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.



Week 4: Dealing with Existing Code

## Unit 5: Building and Previewing the OData UI Service

## Topics

### Week 4

**01**

The Business Scenario

**02**

Creating the CDS Data Model

**03**

Defining and Implementing the Business Object Behavior

**04**

Creating the Business Object Projection

**05**

Building and Previewing the OData UI Service

## What we will achieve in this unit – Service Definition

Define scope of service

```
[Y08] ZUI_RAP_TRAVEL_U_O2_1234 ⌵  
1 @EndUserText.label: 'Travel'  
2 define service ZUI_RAP_TRAVEL_U_O2_1234 {  
3     expose ZC_RAP_Travel_U_1234 as Travel;  
4     expose ZC_RAP_Booking_U_1234 as Booking;  
5     expose /DMO/I_Customer as Passenger;  
6     expose /DMO/I_Agency as TravelAgency;  
7     expose /DMO/I_Carrier as Airline;  
8     expose /DMO/I_Flight as Flight;  
9     expose /DMO/I_Connection as Connection;  
10    expose /DMO/I_Airport as Airport;  
11    expose I_Currency as Currency;  
12 }
```

## What we will achieve in this unit – Service Definition, Service Binding, Preview

Define scope of service

```
[Y08] ZUI_RAP_TRAVEL_U_1234 ⌵  
1 @EndUserText.label: 'Travel'  
2 define service ZUI_RAP_TRAVEL_U_1234 {  
3   expose ZC_RAP_Travel_U_1234 as Travel;  
4   expose ZC_RAP_Booking_U_1234 as Booking;  
5   expose /DMO/I_Customer as Passenger;  
6   expose /DMO/I_Agency as TravelAgency;  
7   expose /DMO/I_Carrier as Airline;  
8   expose /DMO/I_Flight as Flight;  
9   expose /DMO/I_Connection as Connection;  
10  expose /DMO/I_Airport as Airport;  
11  expose I_Currency as Currency;  
12 }
```

Define binding type

[Y08] ZUI\_RAP\_TRAVEL\_U\_O2\_1234 ⌵

### Service Binding: ZUI\_RAP\_TRAVEL\_U\_O2\_1234

This section describes general information about this service binding

Binding Type:

#### Service Versions

Define service versions associated with the service binding

Version	Service Definition
0001	ZUI_RAP_TRAVEL_U_O2_1234

#### Service Version Details

View information on selected service version

[Default Authorization Values:](#)

Local Service Endpoint:

#### Service Information

[Service URL:](#)

#### Entity Set and Association

- 
- 
- 
- 
- 
- 
- 
- 

SAP Fiori Elements Preview

Travel

4132

**Travel** Booking

Travel ID: 4132 Starting Date: Feb 18, 2021 Total Price: 7,346.00 USD

Agency ID: 70007 End Date: Feb 18, 2021 Description: Business Trip for Salvador, Christine

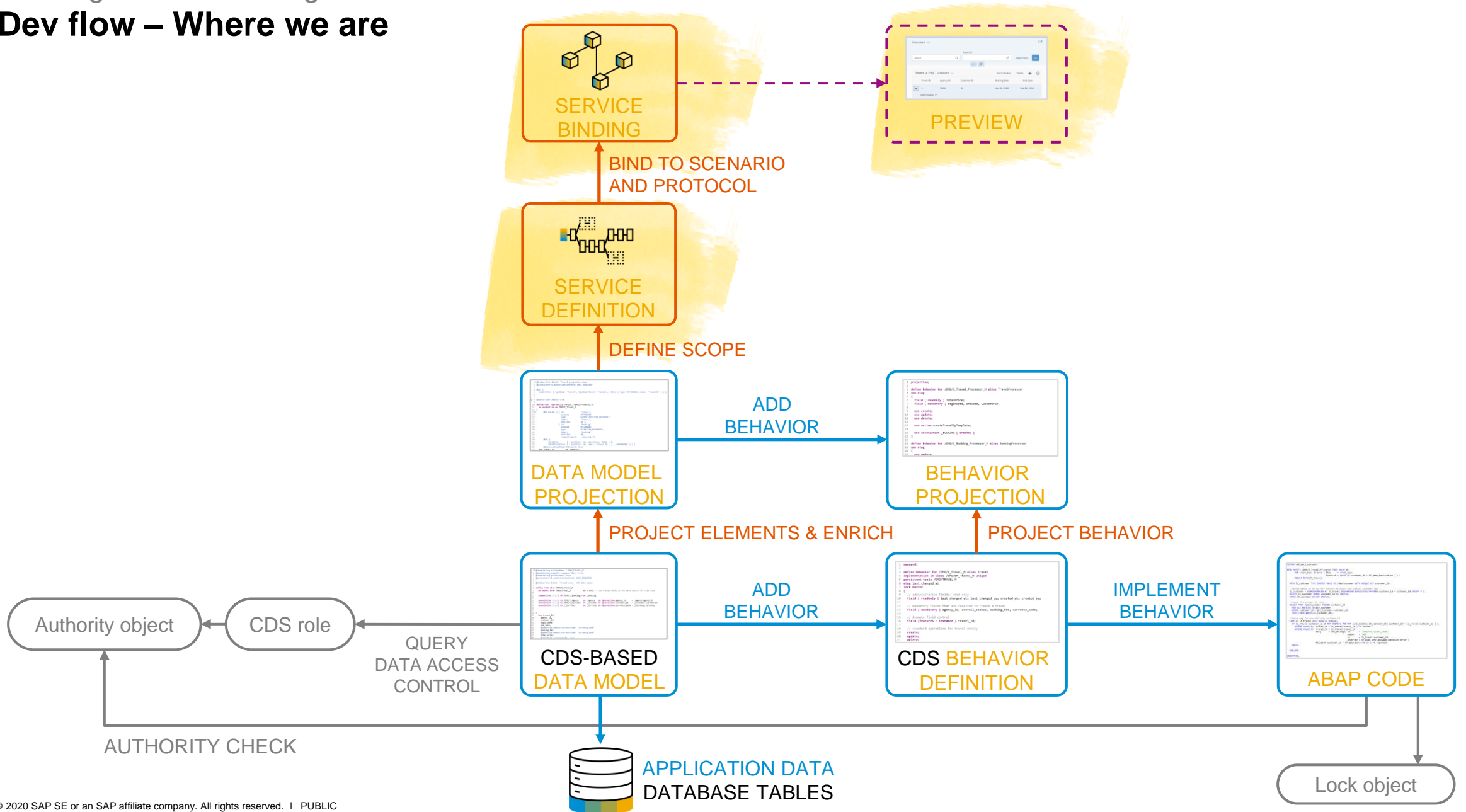
Customer ID: 394 Booking Fee: 20.00 USD Travel Status: N

**Booking**

Booking Number	Booking Date	Customer ID	Airline ID	Flight Number	Flight Date	Flight Price
<input type="radio"/> 1	Feb 12, 2021	262	AA	18	Feb 18, 2021	3,657.00 USD >
<input type="radio"/> 2	Feb 12, 2021	390	AA	18	Feb 18, 2021	3,657.00 USD >

## Building and Previewing the OData UI Service

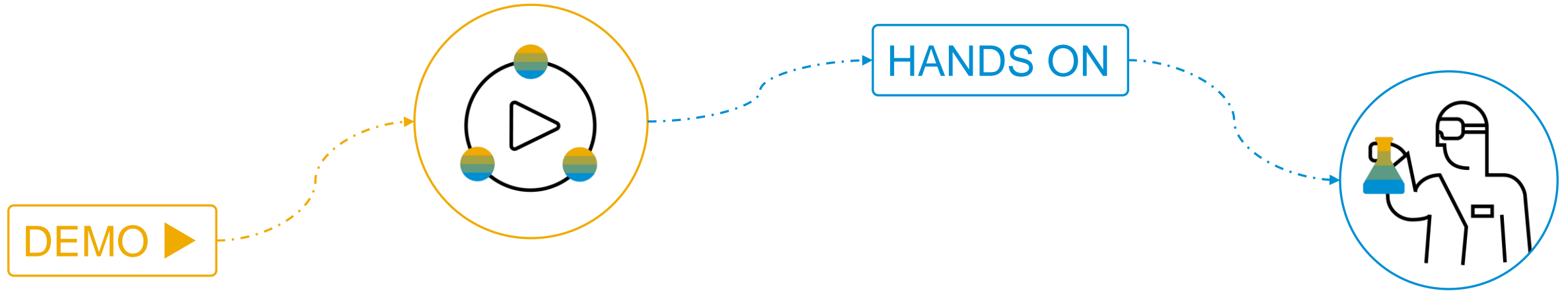
### Dev flow – Where we are





# Building and Previewing the OData UI Service

## Demo



---

### Create and preview the OData UI service

---

1. Create the service definition from the data model projection
2. Create the service binding
3. Preview your SAP Fiori elements app



## Building and Previewing the OData UI Service

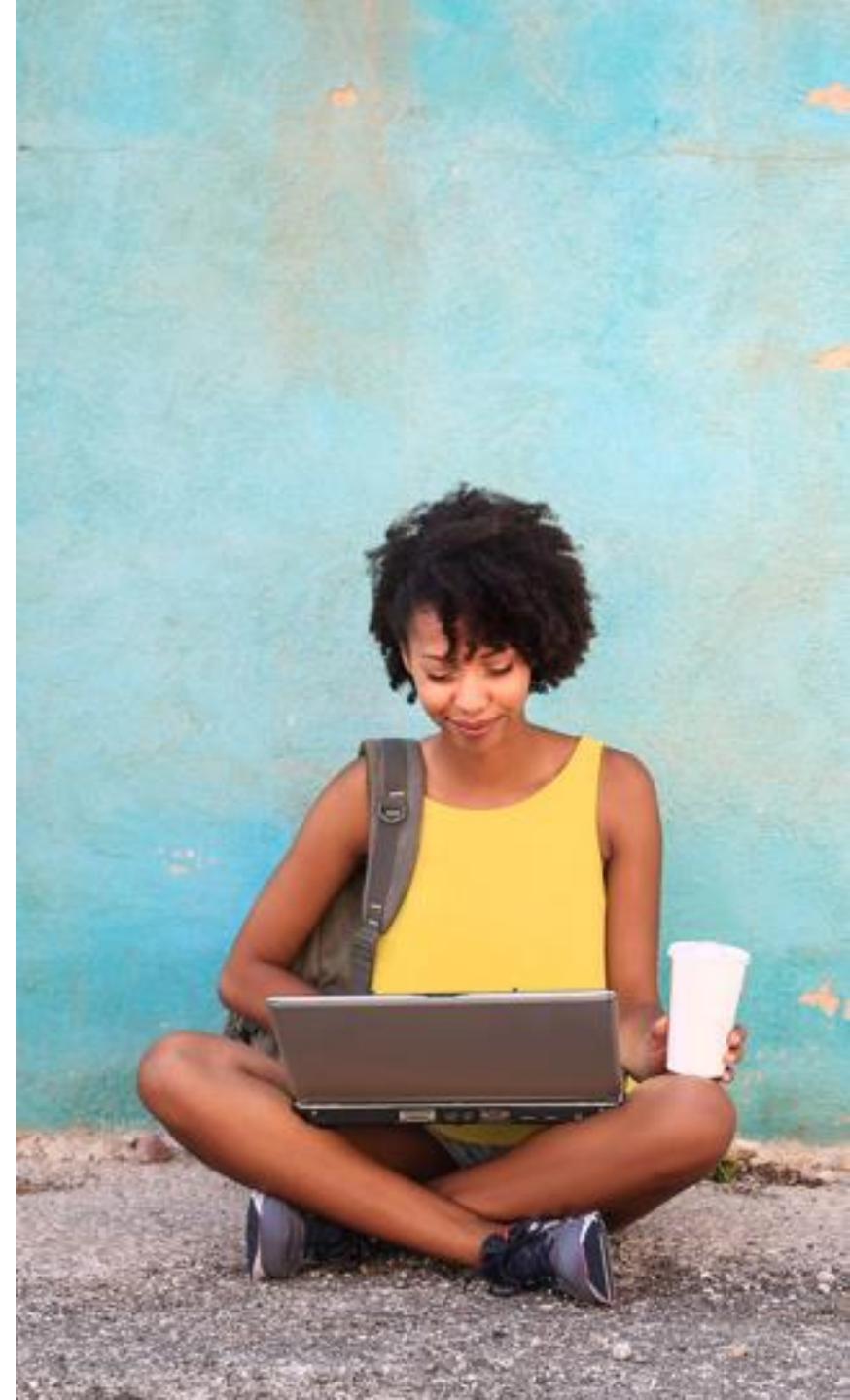
### Wrap-up

#### IN THIS UNIT, YOU LEARNED

- How to create the service definition from the data model projection
- How to create the service binding and test the service

#### NEXT UNIT

- Week 5  
Service Consumption and Web APIs



## Building and Previewing the OData UI Service

### Further reading



Additional Material

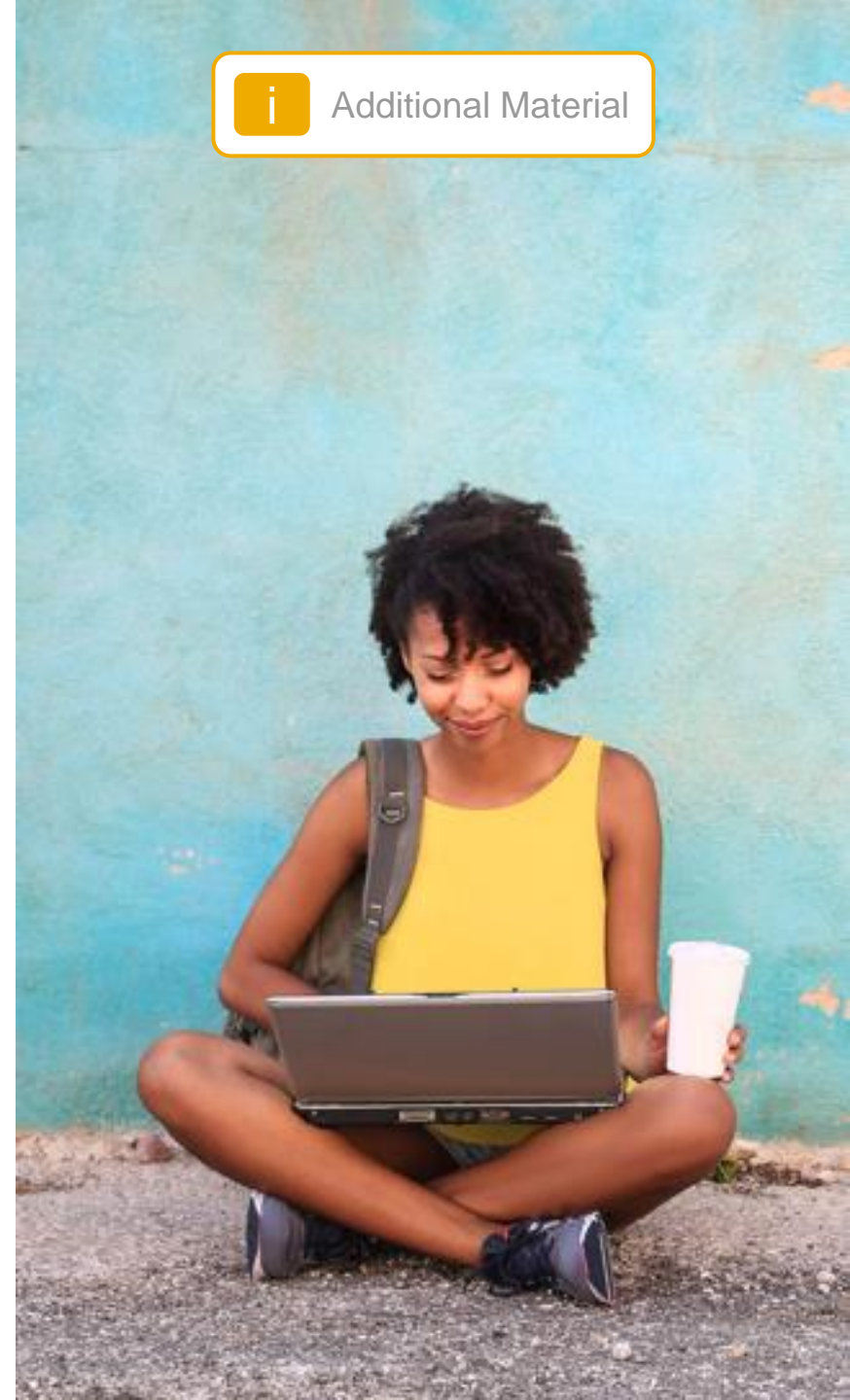
### ABAP RESTful APPLICATION PROGRAMMING MODEL INFORMATION PAGE

---

For more information, links to documentation, tutorials, and more, please visit the RAP at openSAP information page by following the link below

[RAP at openSAP information page \(week 4\)](#)

---



# Thank you.

**Contact information:**

**open@sap.com**

Follow all of SAP



[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.