The ABAP RESTful Programming Model in SAP Cloud Platform ABAP Environment and SAP S/4HANA

Carine Tchoutouo Djomo, SAP

February 25, 2020

PUBLIC

Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP’s intentional or gross negligence.

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, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Agenda

Overview

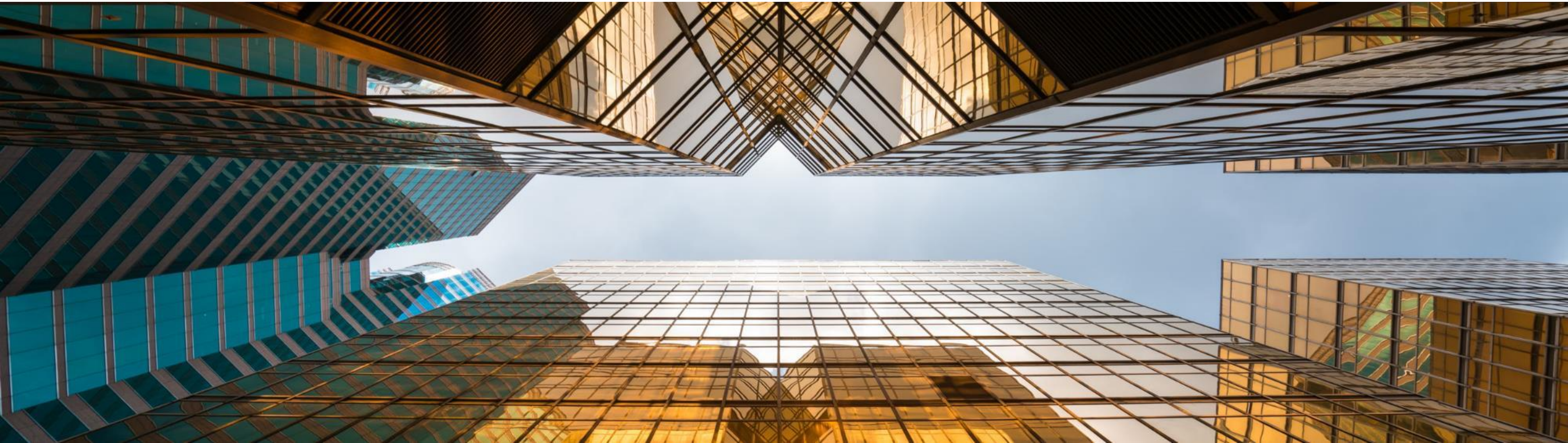
Business Objects

Business Services

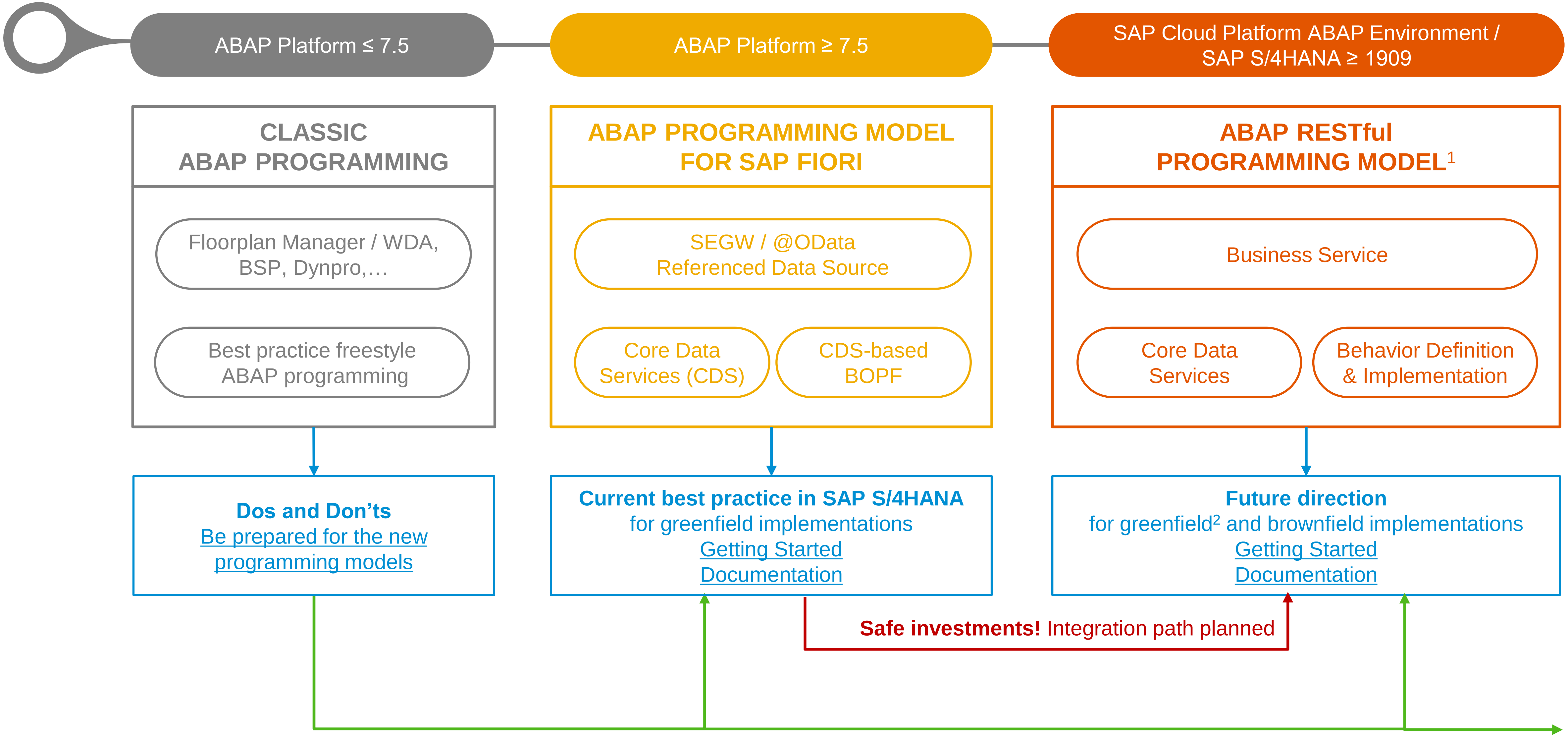
Demo

Outlook & Summary

Overview



Evolution of the ABAP programming model



ABAP Platform ≤ 7.5

ABAP Platform ≥ 7.5

SAP Cloud Platform ABAP Environment / SAP S/4HANA ≥ 1909

CLASSIC ABAP PROGRAMMING

Floorplan Manager / WDA, BSP, Dynpro,…

Best practice freestyle ABAP programming

ABAP PROGRAMMING MODEL FOR SAP FIORI

SEGW / @OData Referenced Data Source

Core Data Services (CDS)

CDS-based BOPF

ABAP RESTful PROGRAMMING MODEL[1]

Business Service

Core Data Services

Behavior Definition & Implementation

Dos and Don’ts
Be prepared for the new programming models

Current best practice in SAP S/4HANA
for greenfield implementations
Getting Started
Documentation

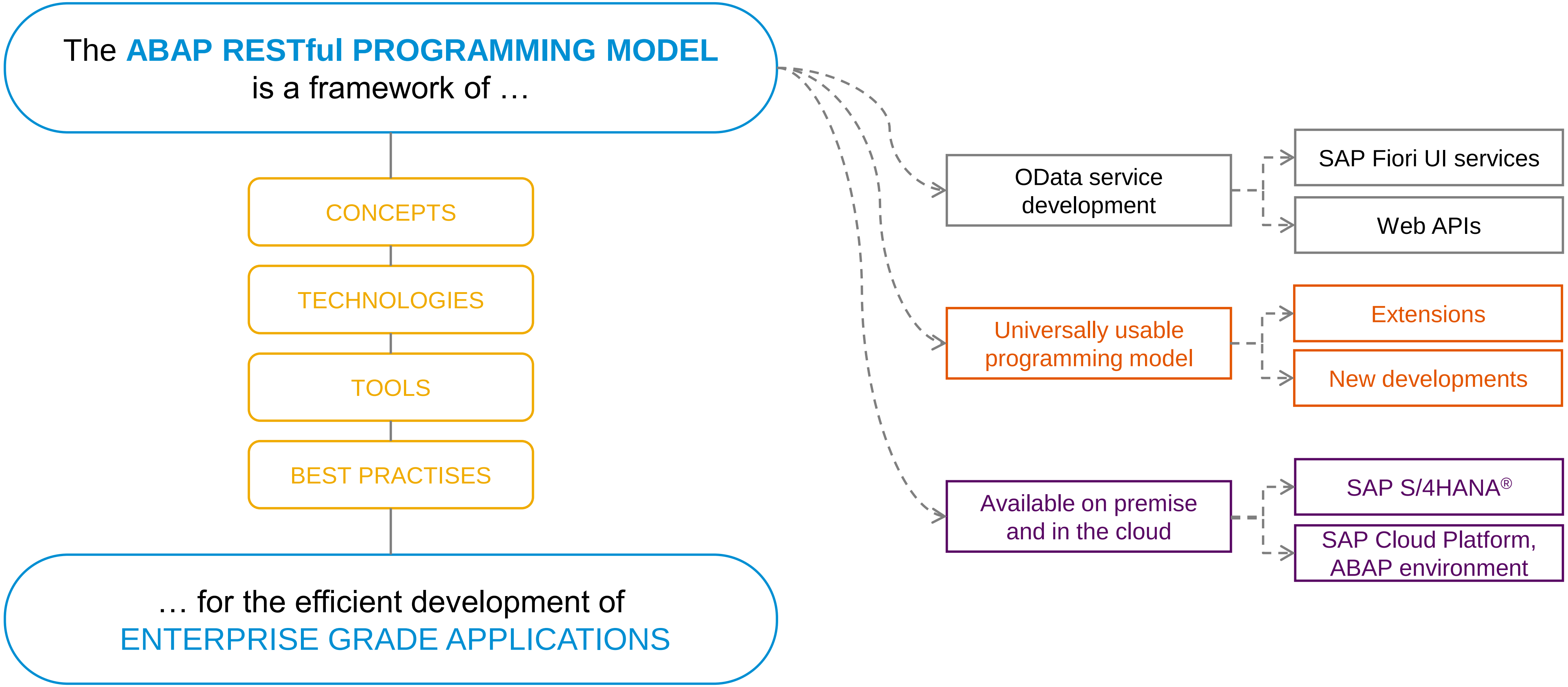
Future direction
for greenfield[2] and brownfield implementations
Getting Started
Documentation

Safe investments! Integration path planned

[1] aka ABAP RESTful Application Programming Model

[2] Greenfield implementations not yet supported on SAP S/4HANA

ABAP RESTful Programming Model – Support for ABAP developers



The **ABAP RESTful PROGRAMMING MODEL** is a framework of …

CONCEPTS

TECHNOLOGIES

TOOLS

BEST PRACTISES

… for the efficient development of ENTERPRISE GRADE APPLICATIONS

- OData service development
  - SAP Fiori UI services
  - Web APIs
- Universally usable programming model
  - Extensions
  - New developments
- Available on premise and in the cloud
  - SAP S/4HANA®
  - SAP Cloud Platform, ABAP environment

ABAP RESTful Programming Model – The key players

ABAP Development Tools in Eclipse for all development tasks

Easy developer onboarding
End-to-end development flow

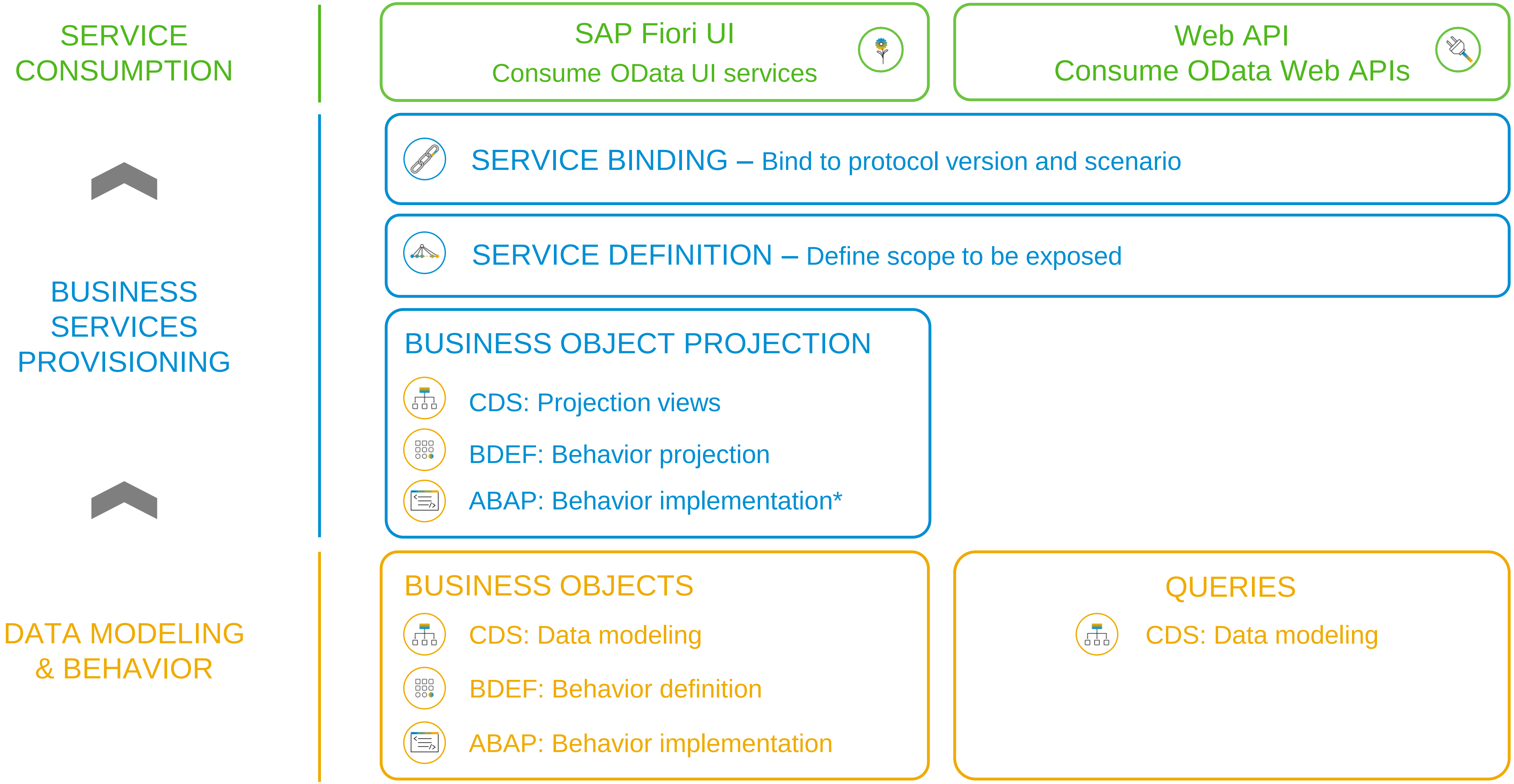
Languages: ABAP and CDS

Standard implementation tasks via typed APIs supporting static code checks, auto-completion, element info

Powerful frameworks

Take over technical implementation tasks
Business logic added in code exits on protocol agnostic layers

ABAP RESTful Programming Model – The big picture



SERVICE CONSUMPTION

SAP Fiori UI

Consume OData UI services

Web API

Consume OData Web APIs

BUSINESS SERVICES PROVISIONING

SERVICE BINDING – Bind to protocol version and scenario

SERVICE DEFINITION – Define scope to be exposed

BUSINESS OBJECT PROJECTION

- CDS: Projection views
- BDEF: Behavior projection
- ABAP: Behavior implementation*

DATA MODELING & BEHAVIOR

BUSINESS OBJECTS

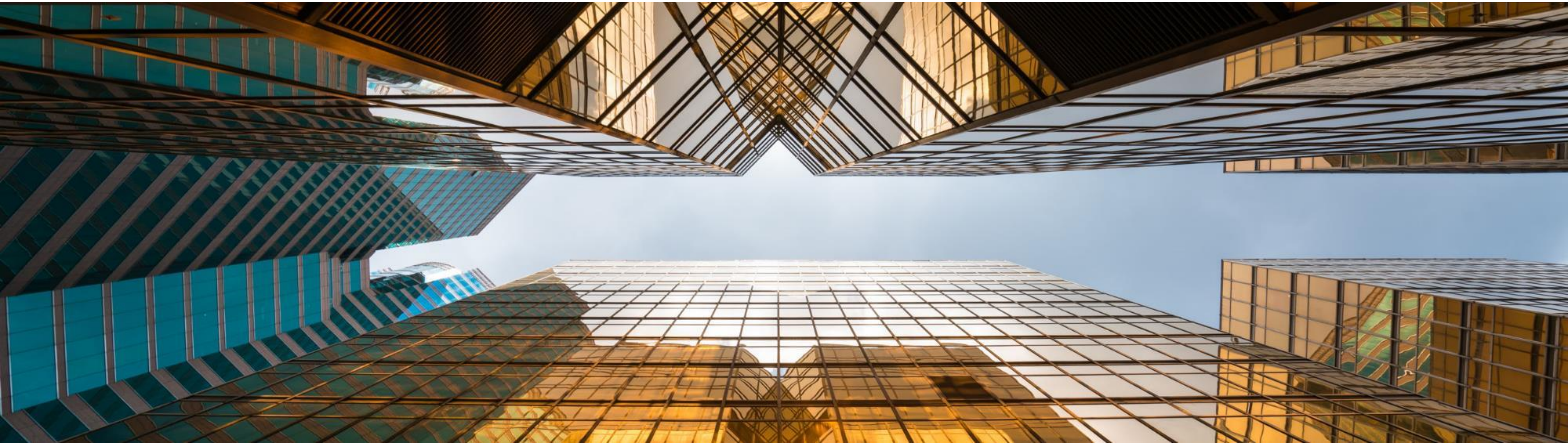
- CDS: Data modeling
- BDEF: Behavior definition
- ABAP: Behavior implementation

QUERIES

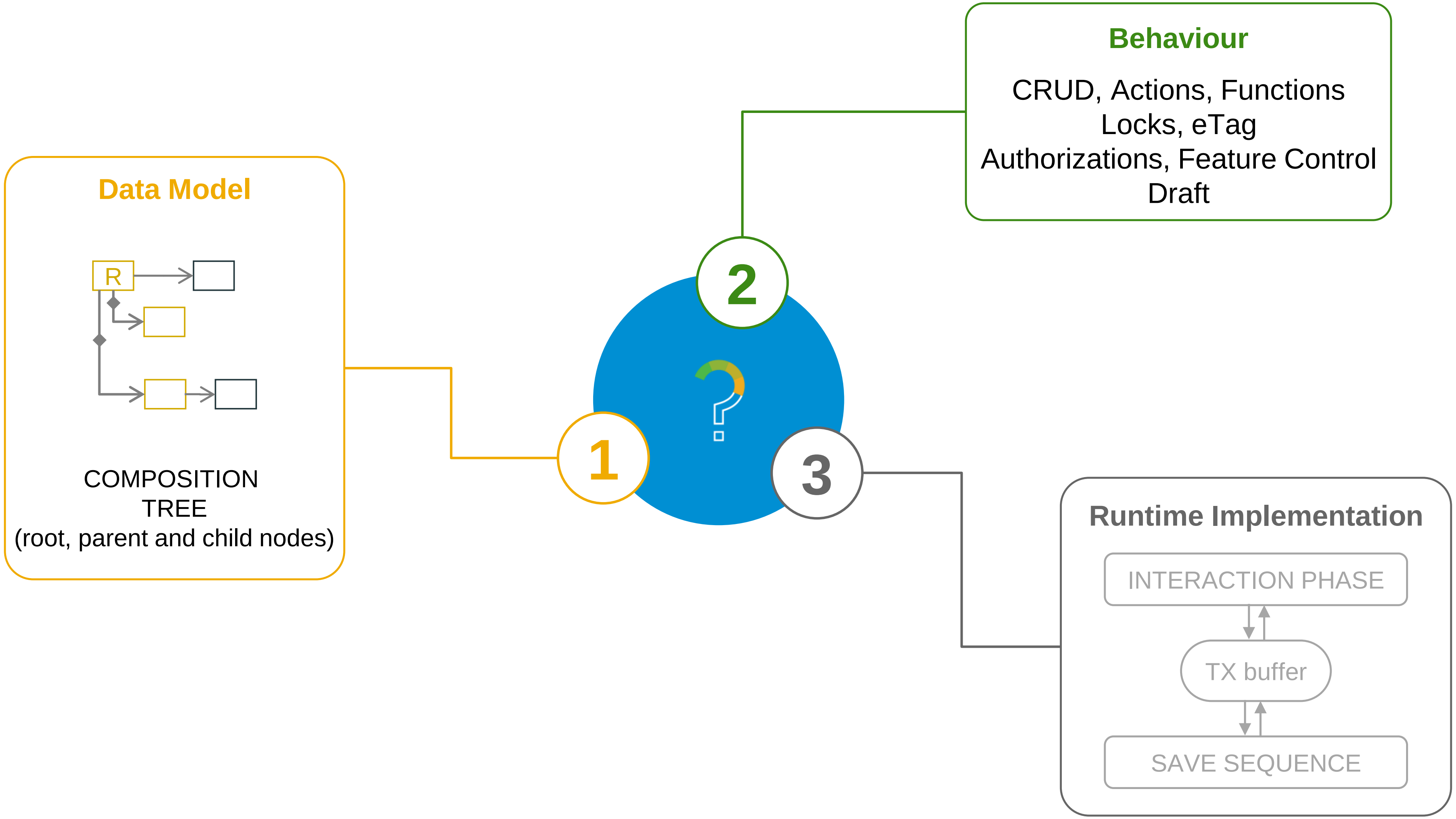
- CDS: Data modeling

* Not yet supported

Business Objects



What is a business object?



Data Model

COMPOSITION
TREE
(root, parent and child nodes)

Behaviour

CRUD, Actions, Functions
Locks, eTag
Authorizations, Feature Control
Draft

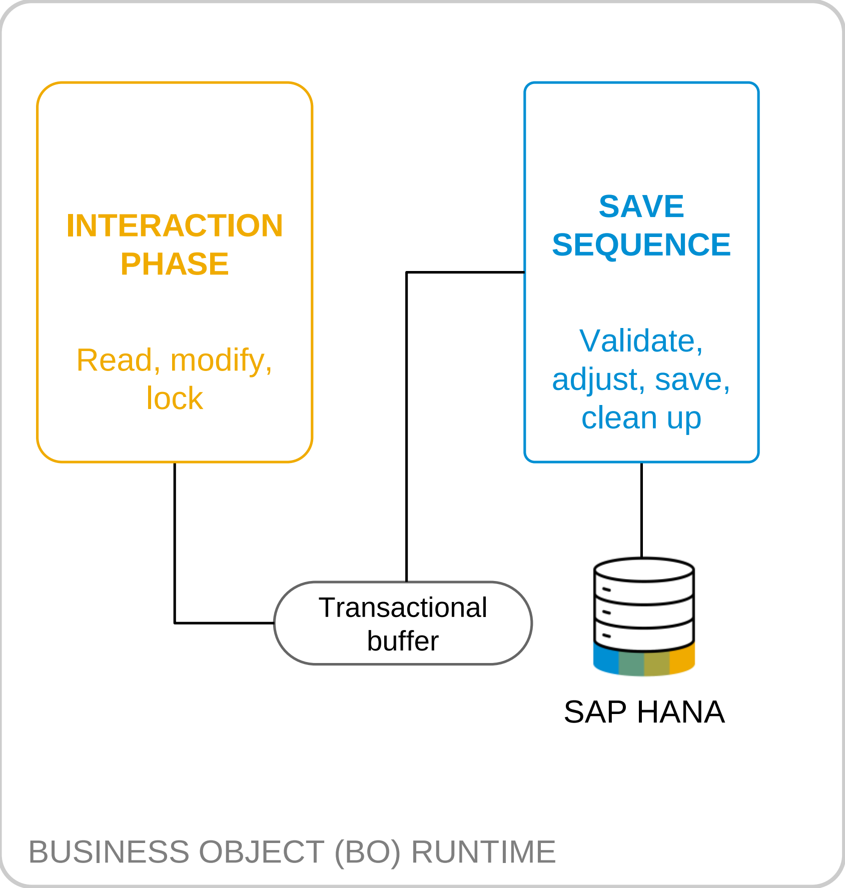
Runtime Implementation

INTERACTION PHASE

TX buffer

SAVE SEQUENCE

Business objects – BO runtime implementation types



INTERACTION PHASE

Read, modify, lock

SAVE SEQUENCE

Validate, adjust, save, clean up

Transactional buffer

SAP HANA

BUSINESS OBJECT (BO) RUNTIME

REUSE EXISTING CODE ➔ UNMANAGED

For brownfield developments with available application code for interaction phase, transactional buffer and save sequence.

➔ Developers in charge of BO runtime: CRUD operations

➔ Adapters needed to integrate the existing code

START FROM SCRATCH ➔ MANAGED*

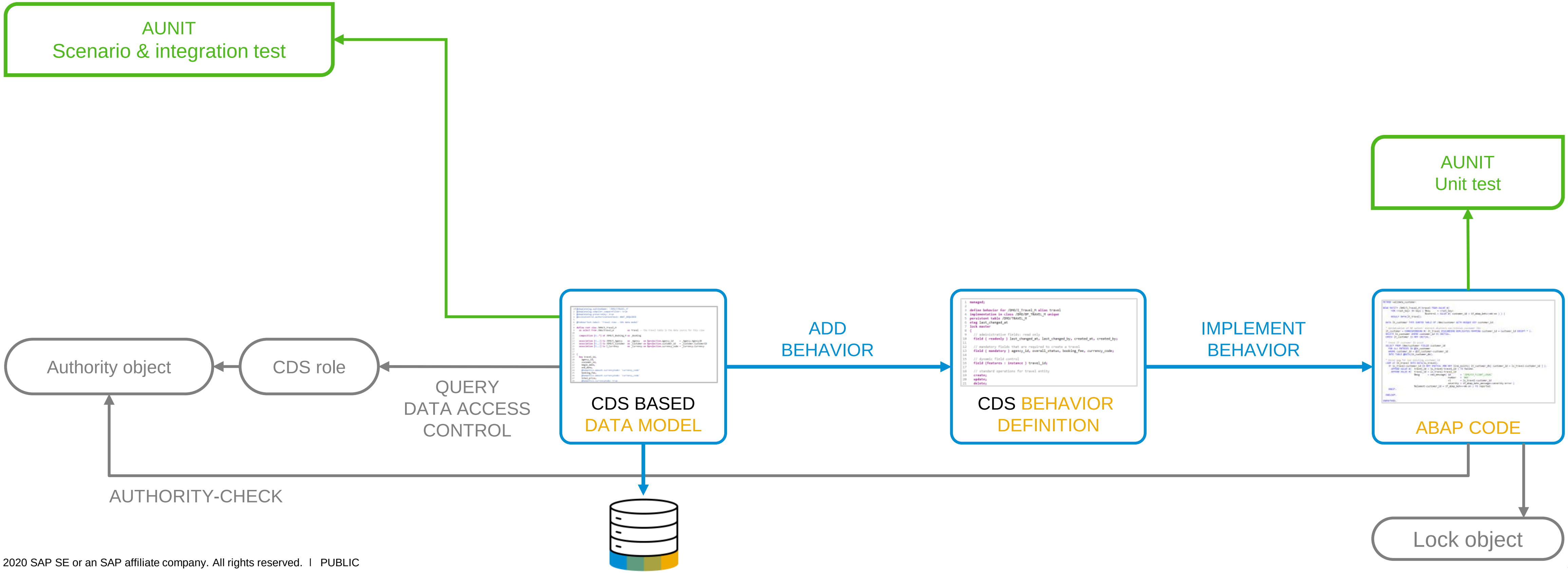
For greenfield developments with standard implementation.
(Opt. unmanaged appl. components: DB tables, lock/PFCG object, update task FM)

➔ Standard CRUD operations works out-of-the-box

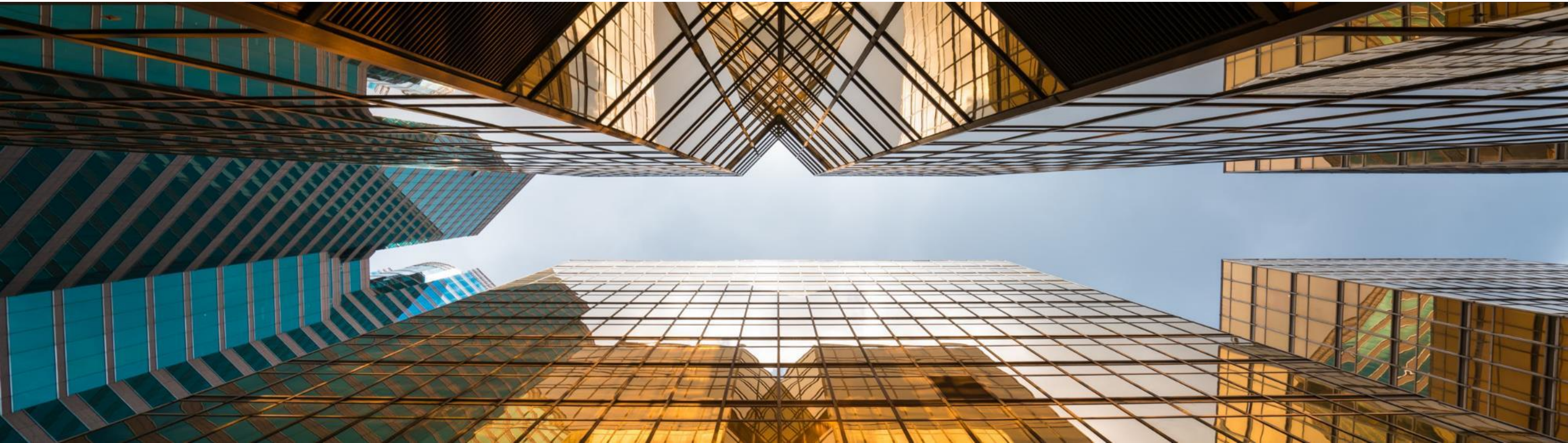
➔ Developers add BO specific business logic

* Currently only available on SAP Cloud Platform ABAP Environment

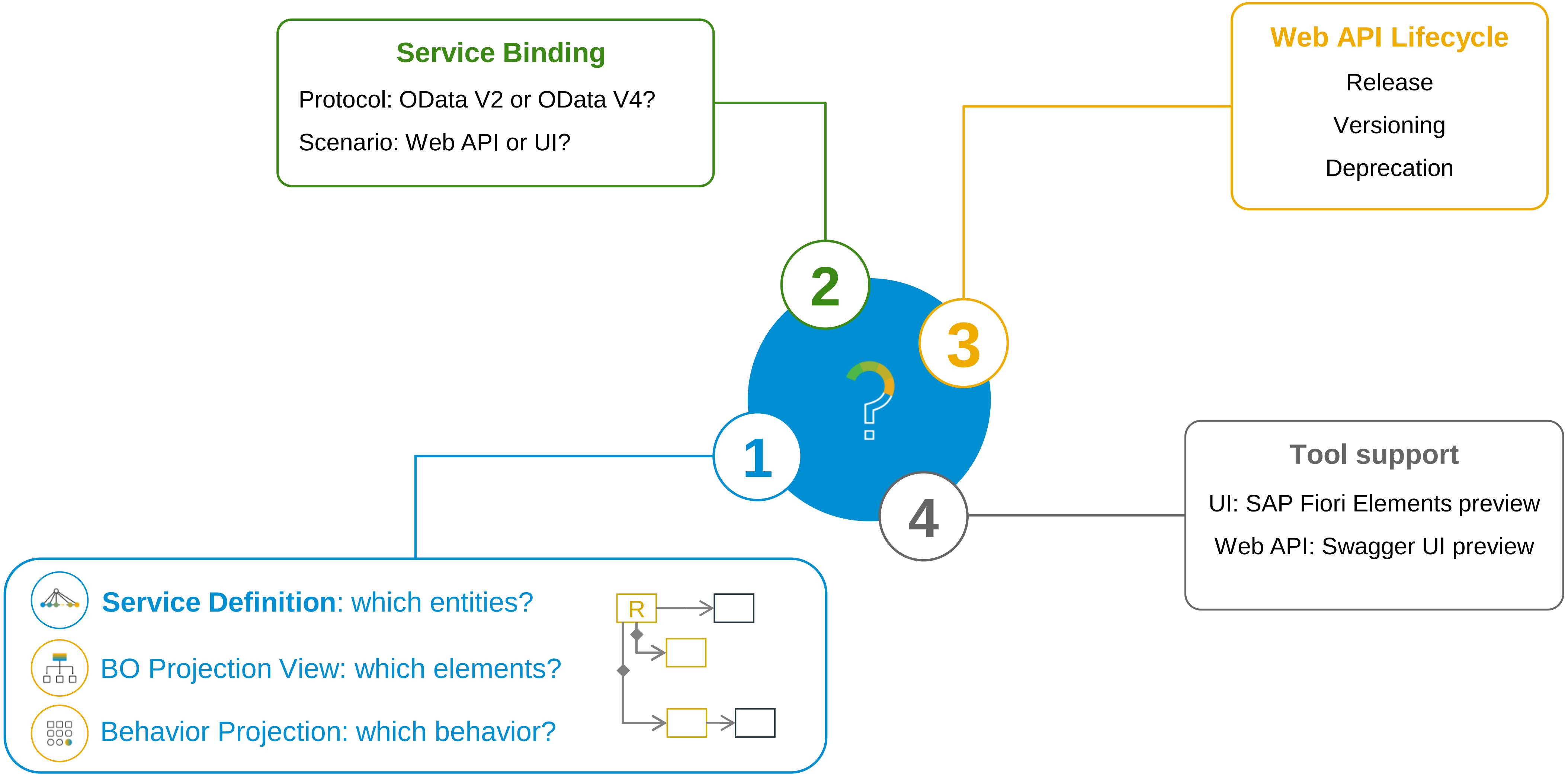
ABAP RESTful Programming Model – Development flow



Business Services



What is a business service



Service Binding

Protocol: OData V2 or OData V4?

Scenario: Web API or UI?

Web API Lifecycle

Release

Versioning

Deprecation

Service Definition: which entities?

BO Projection View: which elements?

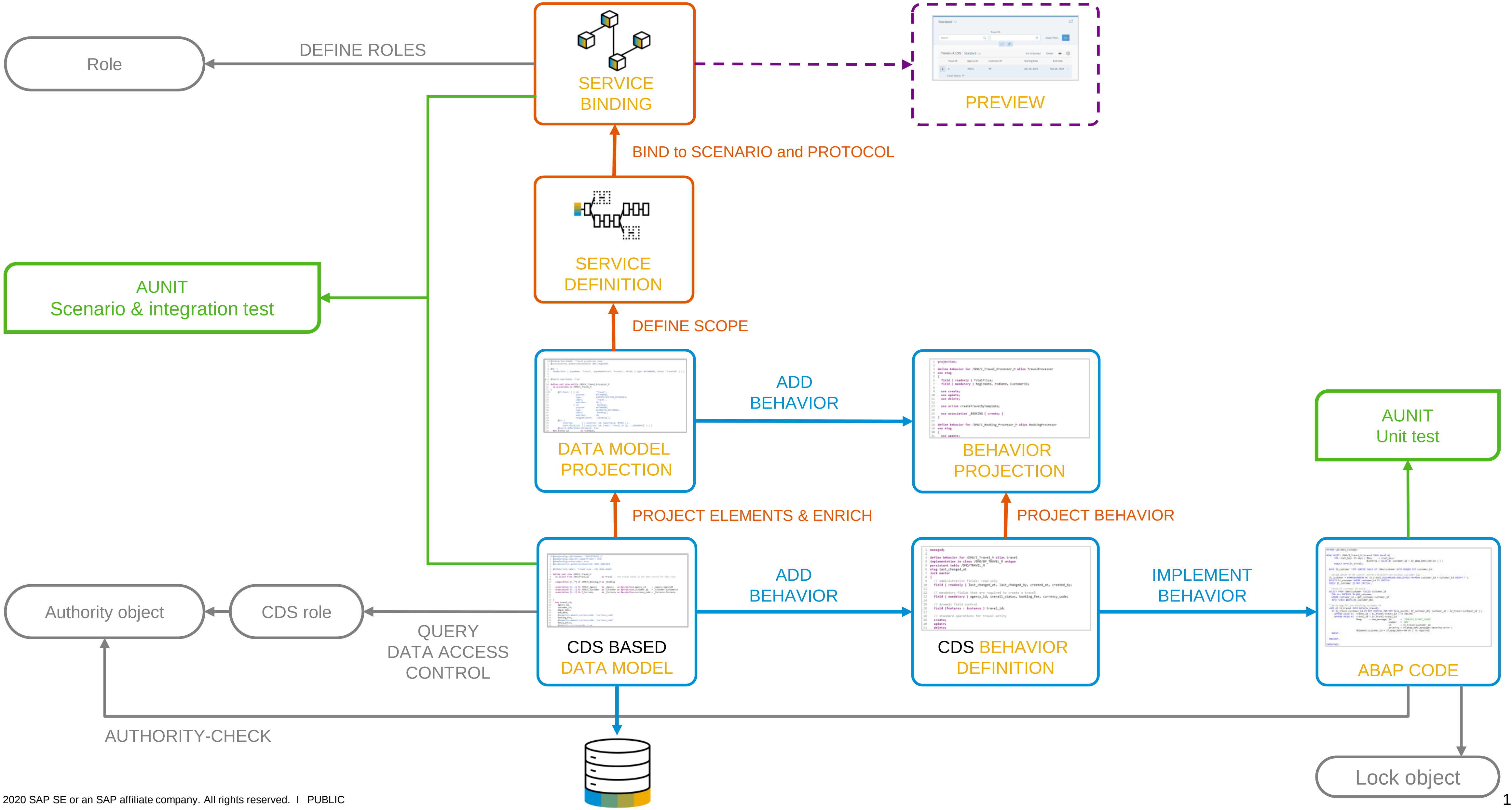
Behavior Projection: which behavior?

Tool support

UI: SAP Fiori Elements preview

Web API: Swagger UI preview

ABAP RESTful Programming Model – Development flow



Demo

Building a SAP Fiori elements UI service with the managed BO runtime implementation

Modern ABAP development in Eclipse

MODERN DEVELOPMENT TOOLSET

Fully eclipse-based

Syntax check, Code completion

Syntax highlighting, Pretty printing

Navigation, Search, Quick Fixes

QUALITY ASSURANCE

Static code checks (CVA, ATC) with remote and local scenarios

Unit testing incl. isolation frameworks

Test seams and injections

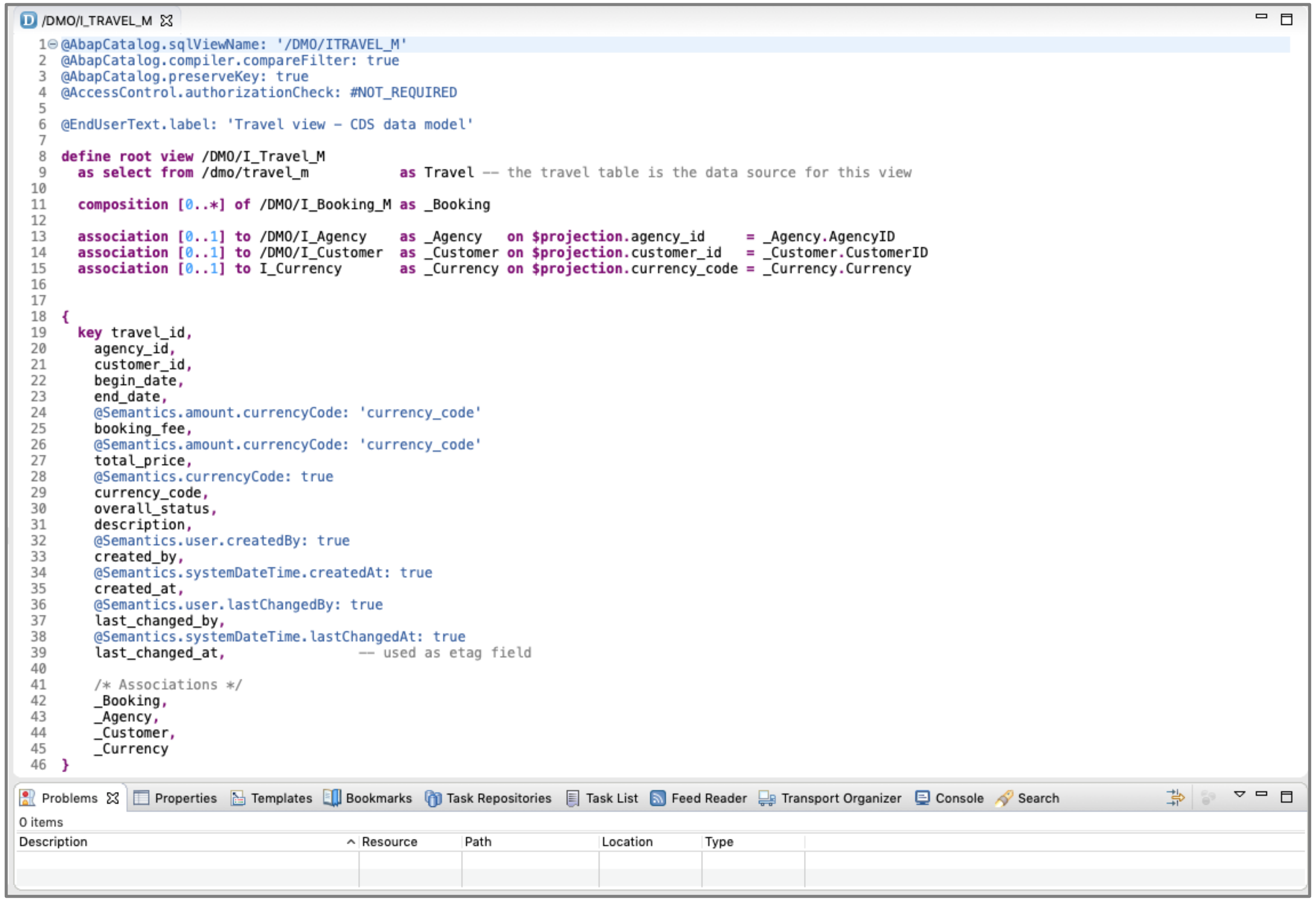
SUPPORTABILITY

Debugging, profiling

Static and dynamic logging

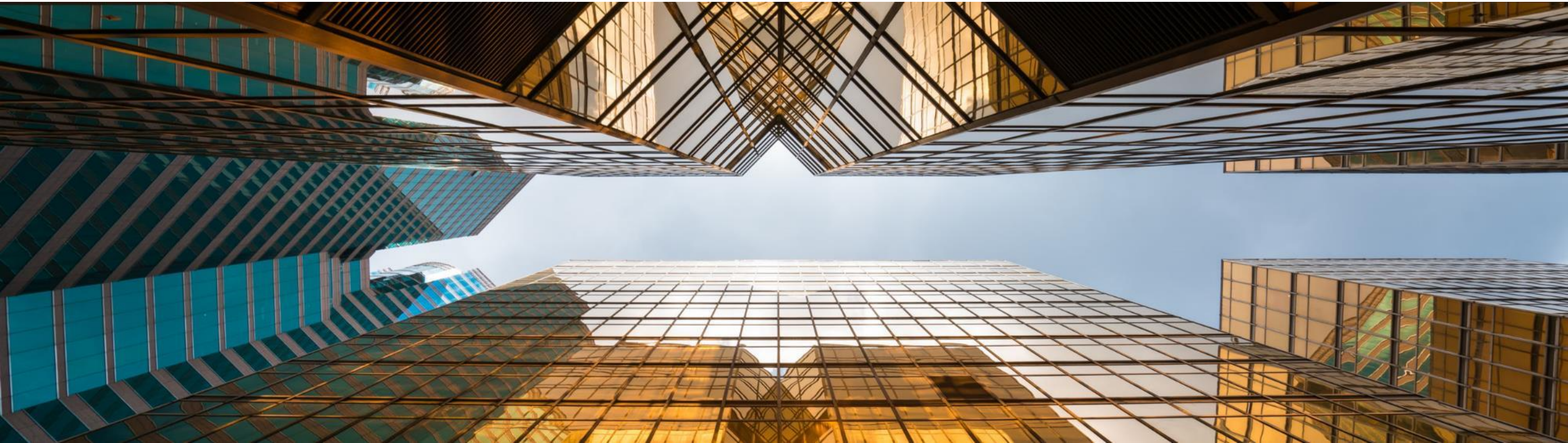
Runtime monitoring and analysis

ABAP DEVELOPMENT TOOLS (ADT)



```
@AbapCatalog.sqlViewName: '/DMO/ITRAVEL_M'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED

@EndUserText.label: 'Travel view - CDS data model'

define root view /DMO/I_Travel_M
  as select from /dmo/travel_m          as Travel -- the travel table is the data source for this view

  composition [0..*] of /DMO/I_Booking_M as _Booking

  association [0..1] to /DMO/I_Agency    as _Agency   on $projection.agency_id     = _Agency.AgencyID
  association [0..1] to /DMO/I_Customer  as _Customer on $projection.customer_id   = _Customer.CustomerID
  association [0..1] to I_Currency       as _Currency on $projection.currency_code = _Currency.Currency


{
  key travel_id,
    agency_id,
    customer_id,
    begin_date,
    end_date,
    @Semantics.amount.currencyCode: 'currency_code'
    booking_fee,
    @Semantics.amount.currencyCode: 'currency_code'
    total_price,
    @Semantics.currencyCode: true
    currency_code,
    overall_status,
    description,
    @Semantics.user.createdBy: true
    created_by,
    @Semantics.systemDateTime.createdAt: true
    created_at,
    @Semantics.user.lastChangedBy: true
    last_changed_by,
    @Semantics.systemDateTime.lastChangedAt: true
    last_changed_at,                -- used as etag field

    /* Associations */
    _Booking,
    _Agency,
    _Customer,
    _Currency
}
```

Outlook & Summary



Outlook – Next steps planned*

Goals
2020

SAP CLOUD PLATFORM ABAP ENVIRONMENT

Enhanced managed scenario (e.g. early numbering based on UUIDs, draft support)

Markdown editor for documentation of behavior definitions

OData V4 support

SAP S/4HANA

Managed scenario for supporting greenfield development with draft support and early numbering based on UUIDs

Enhanced unmanaged scenario

OData V4 support

Read more in the ABAP Platform Road Map

* This is the current state of planning and may be changed by SAP at any time.

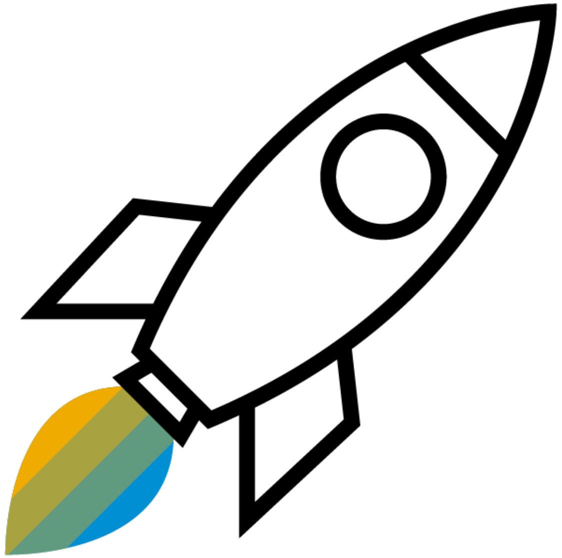
Summary – Key takeaways

The **ABAP RESTful Programming Model** is available starting with SAP Cloud Platform ABAP Environment 1808: Cloud-first delivery.

The **ABAP RESTful Programming Model** is available on-premise starting with SAP S/4HANA 1909 – with reduced feature scope: *UNMANAGED BO IMPLEMENTATION.*

The feature scope of the **ABAP RESTful Programming Model** is enhanced on a quarterly basis in SAP Cloud Platform ABAP Environment and on-premise on a yearly basis in SAP S/4HANA.

The **ABAP Programming Model for SAP Fiori** *WITH DRAFT SUPPORT* is available as of AS ABAP 7.51 and remains the current best practice for greenfield implementations in SAP S/4HANA.



GETTING STARTED

With the

ABAP RESTful Programming Model

Further information

- Getting Started with the ABAP RESTful Programming Model
- Get started with ABAP Core Data Services (CDS)
- Evolution about the ABAP Programming Model
- Be prepared for the new ABAP Programming Models
- ABAP Development Community
- SAP Cloud Platform, ABAP Environment
- SAP Developer Center
- SAP TechEd Online (e.g. session CAA103, CAA100 and CAA101)
- SAP Training & Certifications Opportunities

Thank you.

Contact Information:

Carine Tchoutouo Djomo
Product Manager for ABAP Platform
SAP SE
carine.tchoutouo.djomo@sap.com