



Guidance for building Apps with the **ABAP RESTful Application Programming Model (RAP)**

André Fischer, ABAP Platform Product Management, SAP SE
October 12, 2021

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.

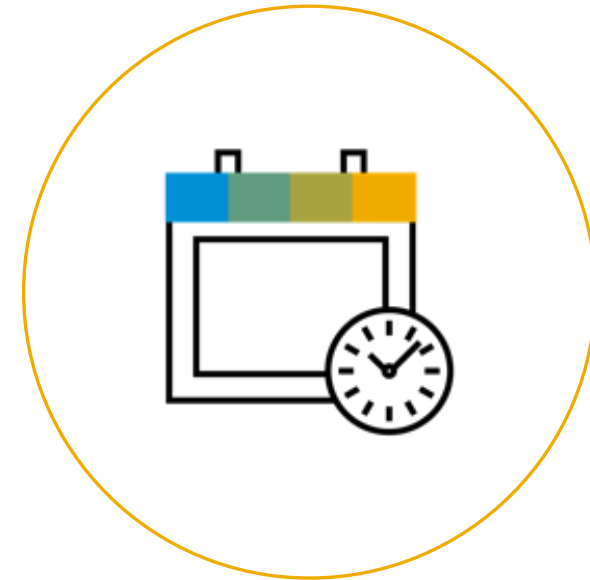
All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Agenda

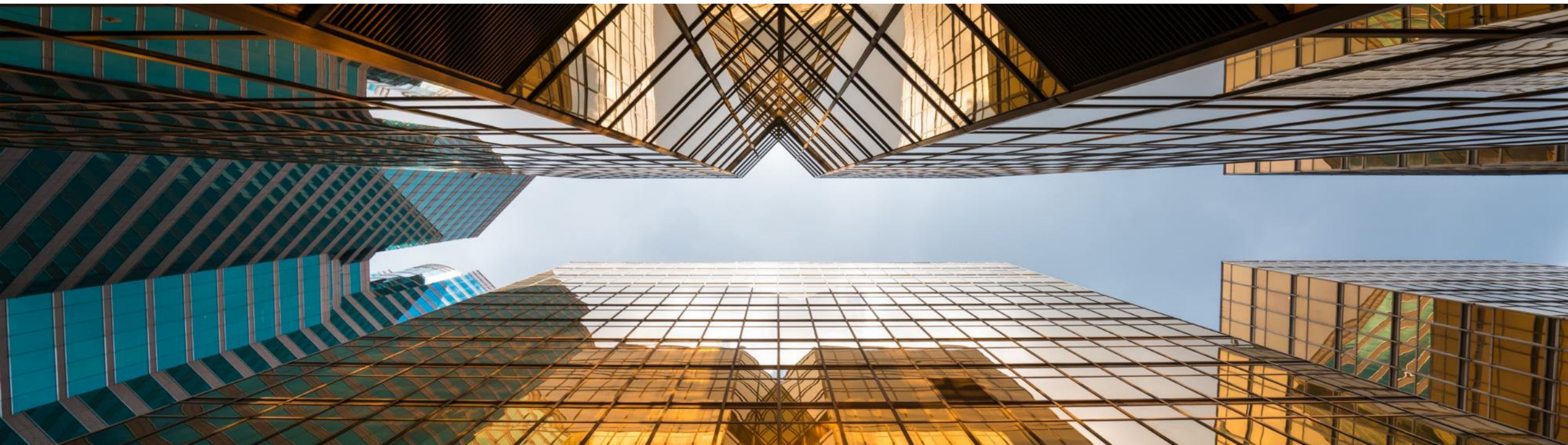
BIG PICTURE

DEMO

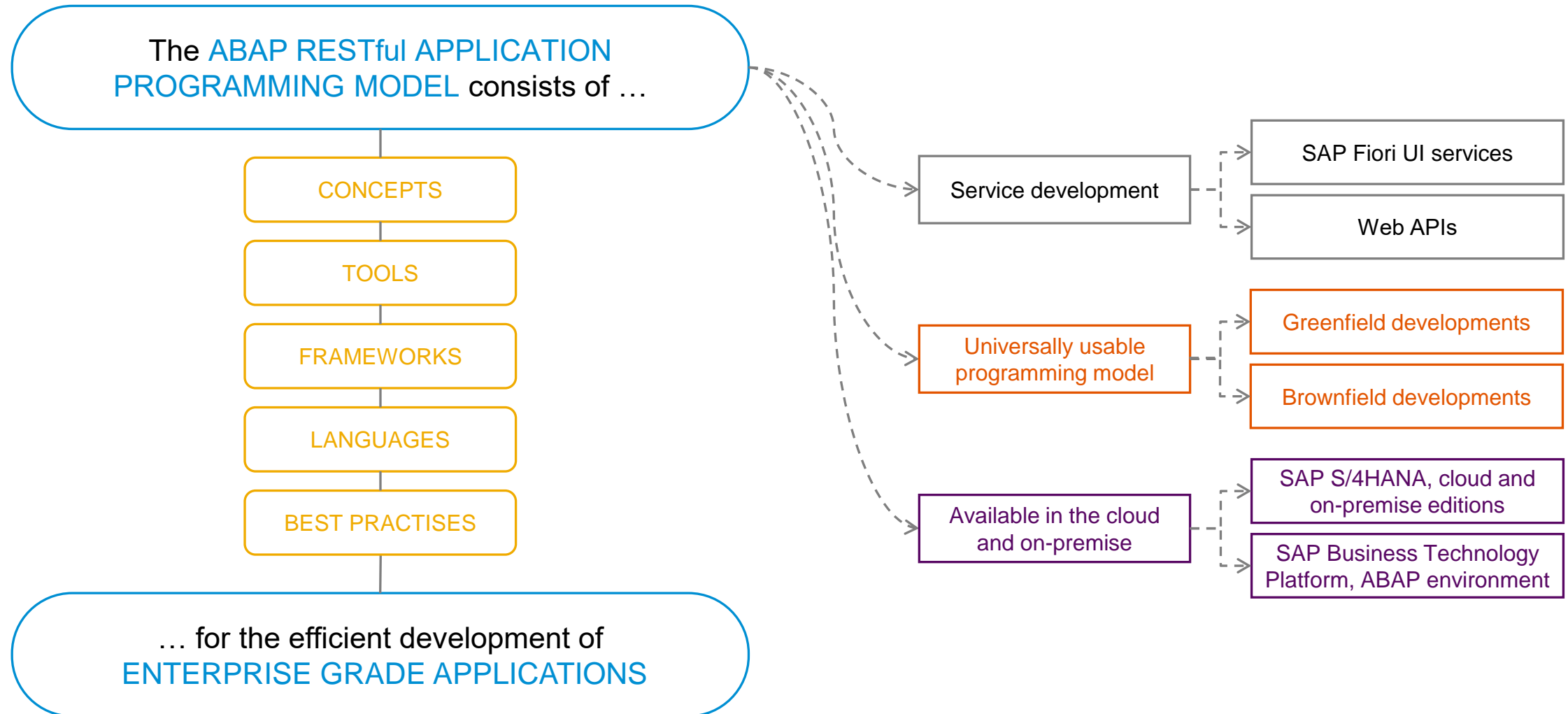
SUMMARY



BIG PICTURE



ABAP RESTful Application Programming Model (RAP) – At a glance



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

The big picture



CONSUMPTION

DATA INTEGRATION *

Consume SQL based services

WEB APIs

Consume OData based services

SAP FIORI UIs

Consume OData based UI services

SAP ANALYTICS CLOUD *

Consume InA based UI services for live data access

BUSINESS SERVICE PROVISIONING



SERVICE BINDING – Bind to protocol version and scenario



SERVICE DEFINITION – Define scope to be exposed

BUSINESS OBJECT PROJECTION



CDS: BO projection views



BDEF: Behavior projection



ABAP: Behavior implementation

ANALYTICAL PROJECTION



CDS: Analytical projection views

DATA MODELING & BEHAVIOR

CDS ENTITIES



CDS: Data modeling

BUSINESS OBJECTS



CDS: Data modeling



BDEF: Behavior definition



ABAP: Behavior implementation

ANALYTICAL MODEL

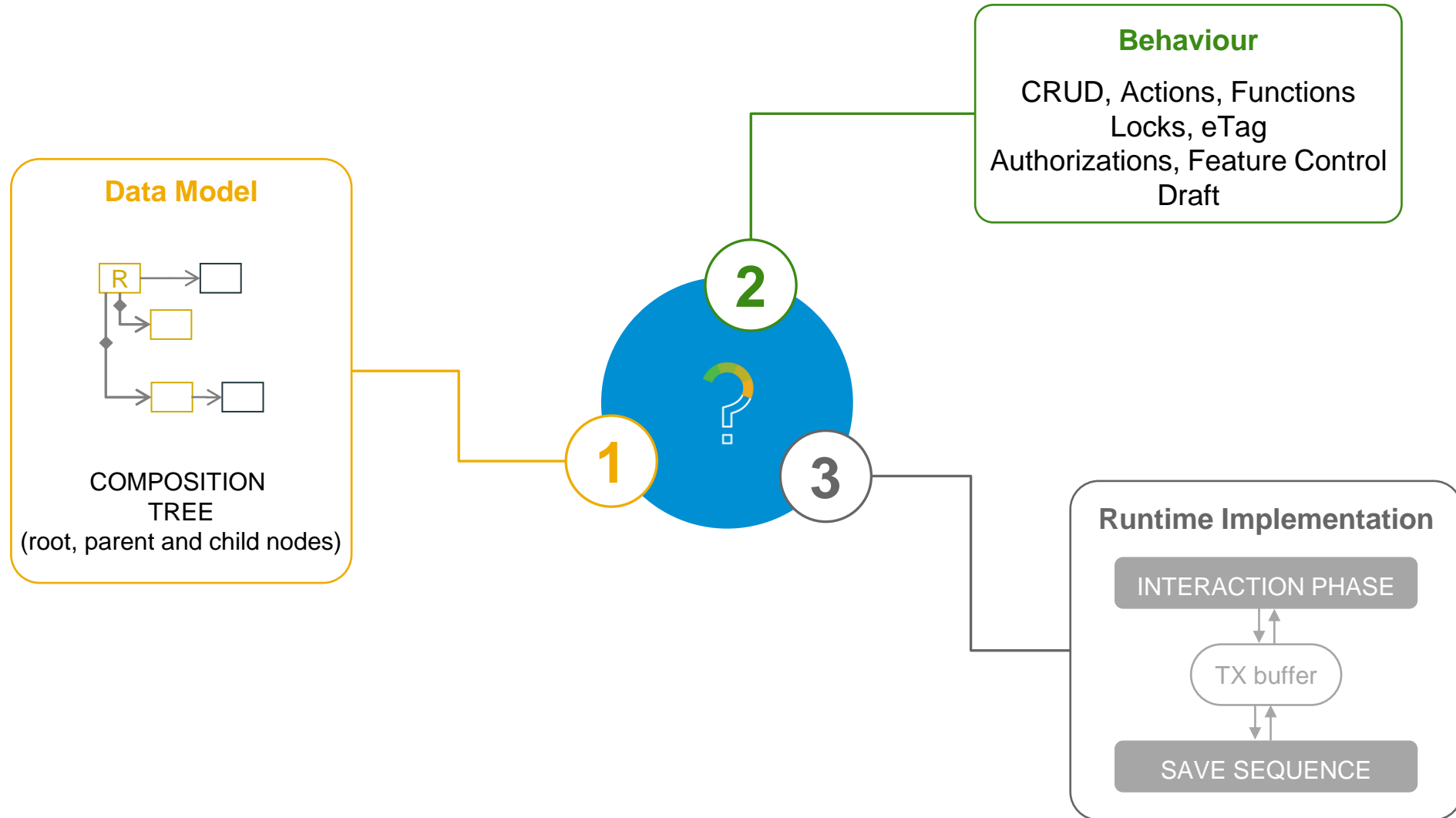


CDS: Analytical cubes

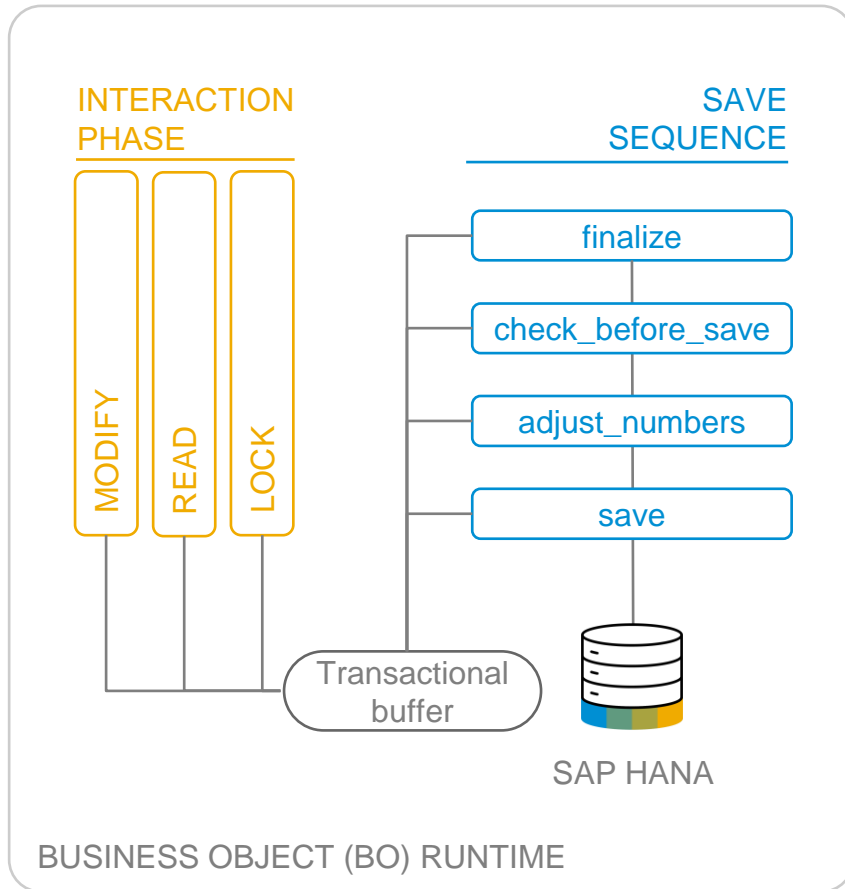


CDS: Analytical dimensions

What is a business object (BO) ?



BO runtime implementation types – Overview



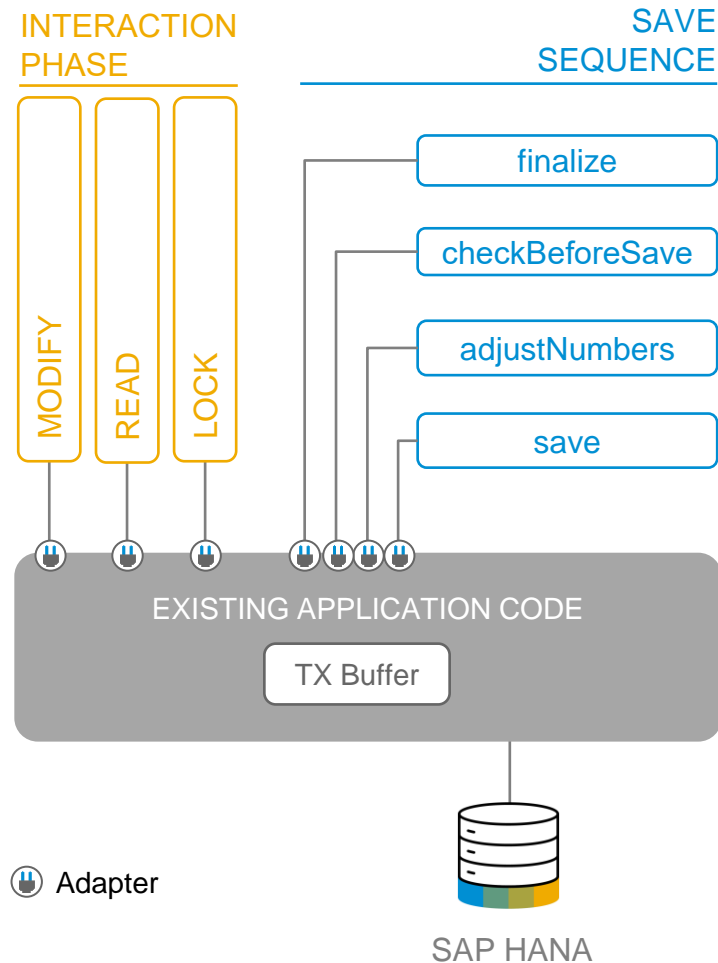
UNMANAGED

Brownfield development with application coding fully available:
Interaction phase + Transactional buffer + Save sequence

MANAGED

Greenfield development with standard implementation
(opt. unmanaged appl. components: DB tables, lock/PFCG object, update task FM)

BO runtime implementation types – unmanaged



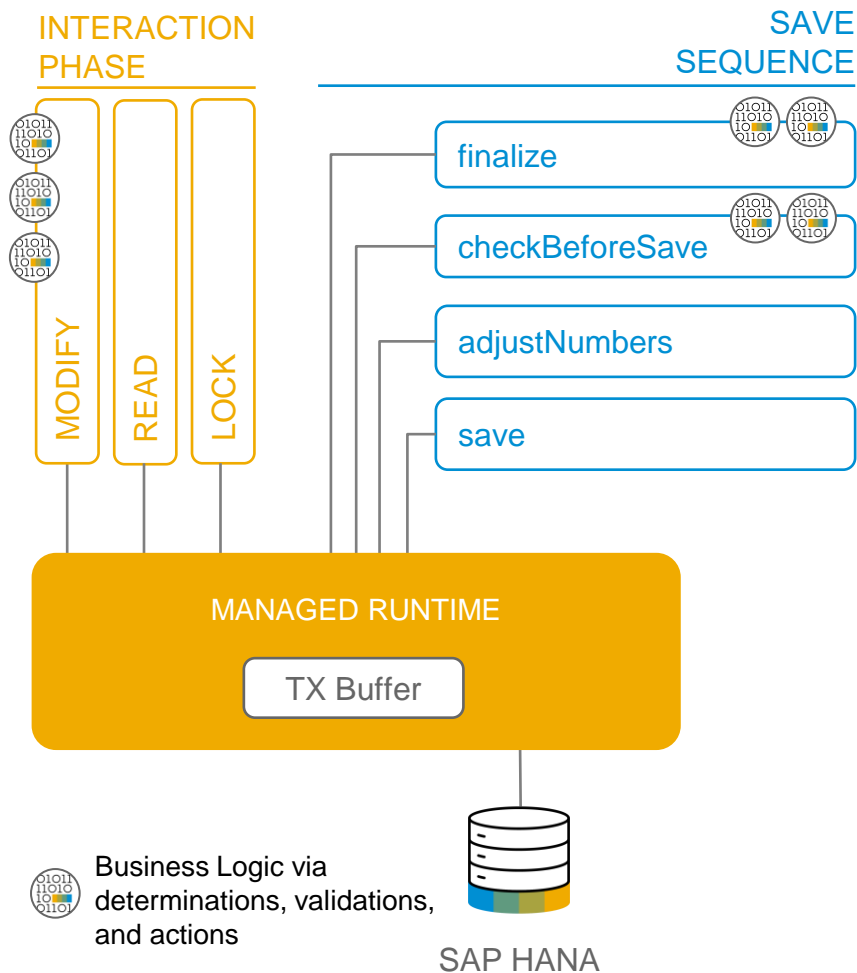
Application coding

- Already available and reusable
- For interaction phase, transactional buffer and save sequence
- Decoupled from UI technology
- Developers fully in charge of BO runtime: CRUD operations
- Existing business logic added using adapters

Examples

- Sales Order, Purchase Order

BO runtime implementation types – managed



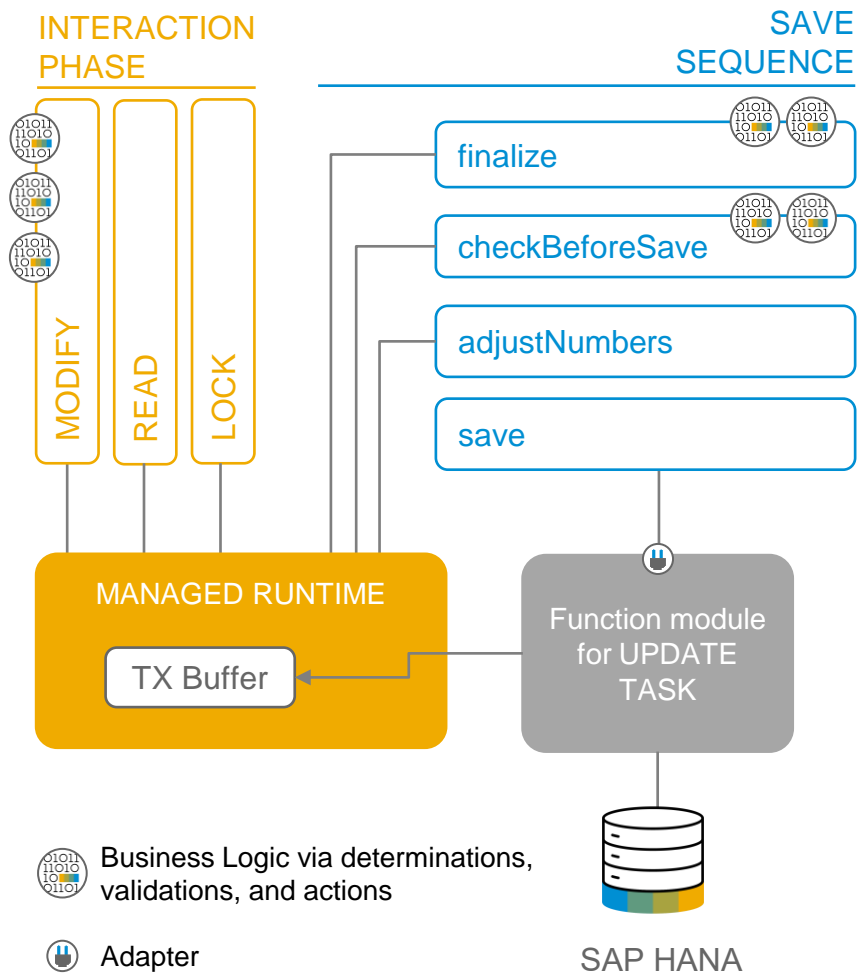
Application coding

- Not yet available – or fine granular reusable code available
- Technical implementation tasks taken over by BO infrastructure
- Standard CRUD operations available out-of-the-box
- Developers' focus on business logic, added via code exits: determinations, validations, and actions.

Examples

- New apps in SAP BTP ABAP Environment and SAP S/4HANA

BO runtime implementation types – managed with unmanaged save



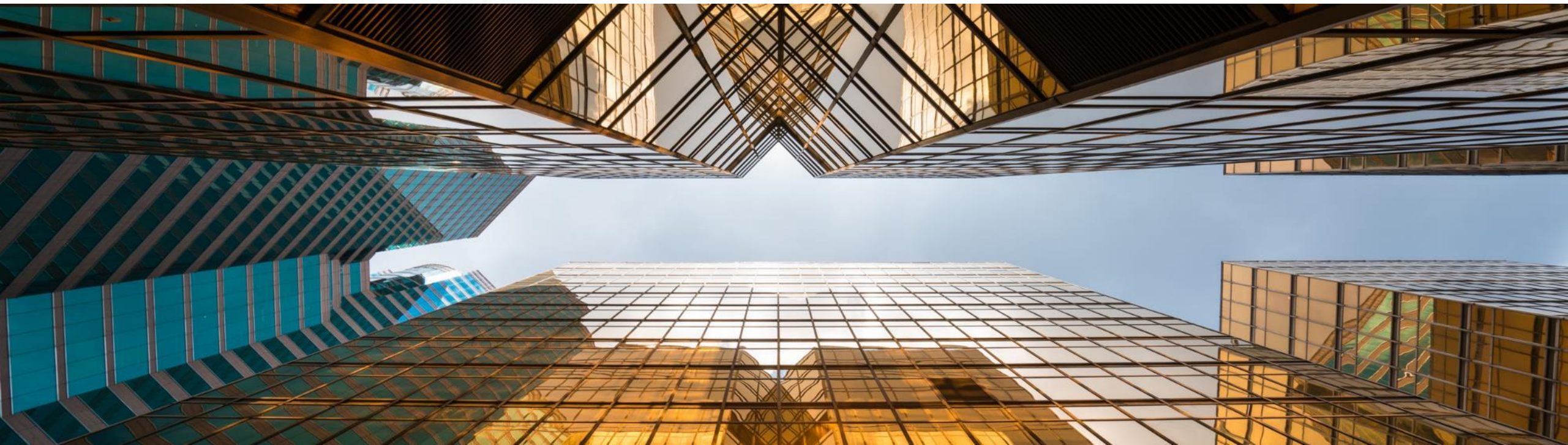
Application coding

- “update-task function module” available and reusable
- Coding for interaction phase not available (e.g. highly coupled in older UI technology: DYNP - PBO / PAI)
- Technical implementation tasks taken over by BO infrastructure
- Developers in charge of persisting the data

Examples

- Business Partner, Product
- [SAP Master Data Governance](#) → Consumes the Business Partner RAP BO

When to use what?



Status Quo: No custom code available

Goal

- New custom application
- New extension
- New API

Recommendation

- **Managed BO implementation**

Status Quo: Custom code available

LOOSELY COUPLED

1. **Managed BO implementation with unmanaged save**
→ for fine-grained, reusable APIs
2. **unmanaged BO implementation**
→ for coarse-grained, reusable APIs
→ when using CDS custom entities

TIGHTLY COUPLED

1. **Managed BO implementation**
→ custom code cannot be reused
2. **Managed BO implementation with unmanaged save implementation**
→ if only the custom code for the update task can be decoupled
3. **Unmanaged BO implementation**
→ when using CDS custom entities

Status Quo: Custom code originating from the ABAP Programming Model for SAP Fiori available

Goal

- Leverage RAP features that are not available in *ABAP Programming Model for SAP Fiori*

Recommendation (*Planned*)

- **BOPF-managed BO implementation**
→ for CDS-based BOPF BOs planned to be delivered in the near future

Status Quo: Building the 101st app but no RAP requirements

Goal

- Build a new app as part of an existing application suite (e.g. another SEGW project)

Recommendation

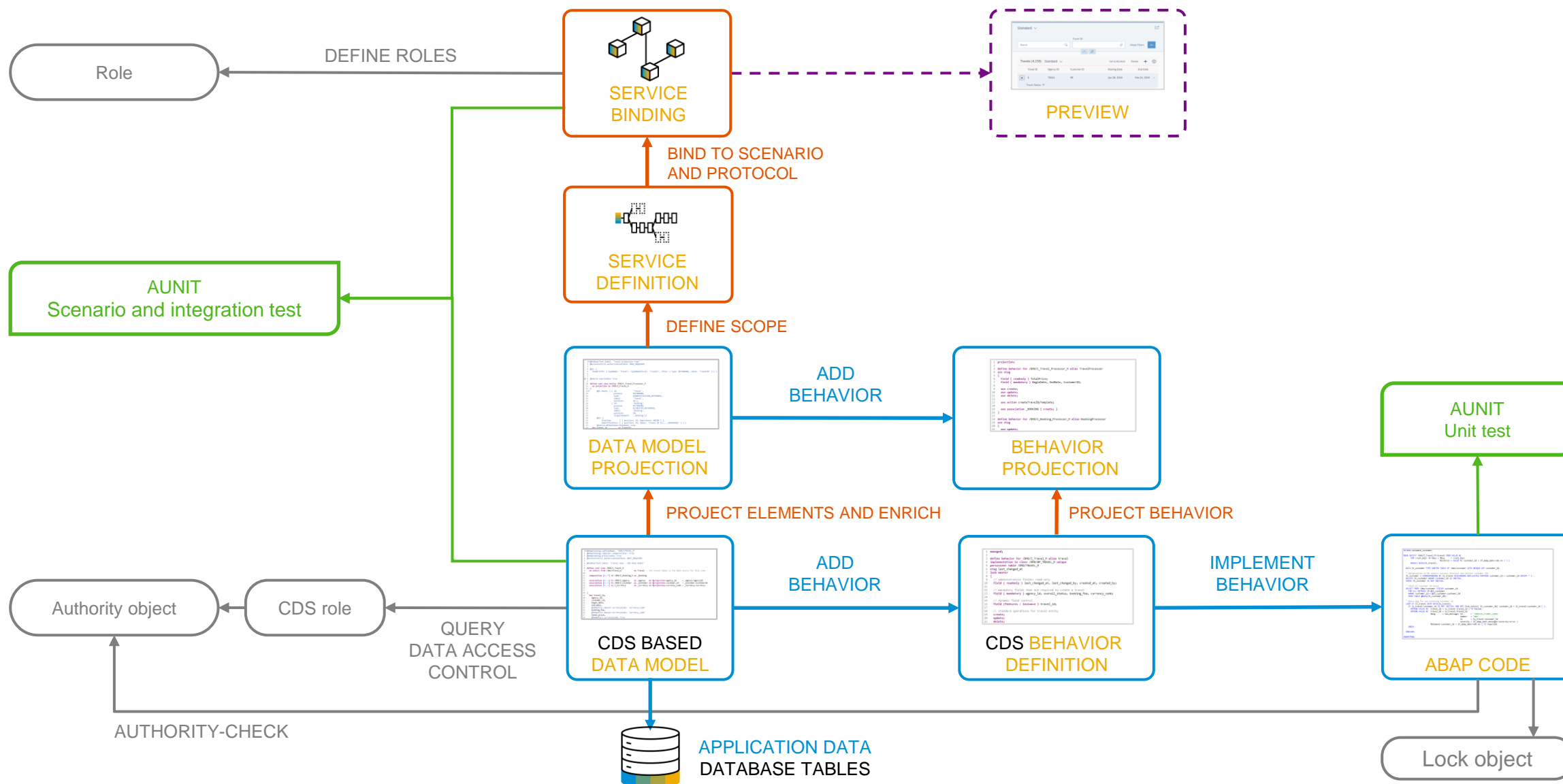
- Use the same technology
- If you want and can try out something new
 - Use the ABAP RESTful application programming model

DEMO

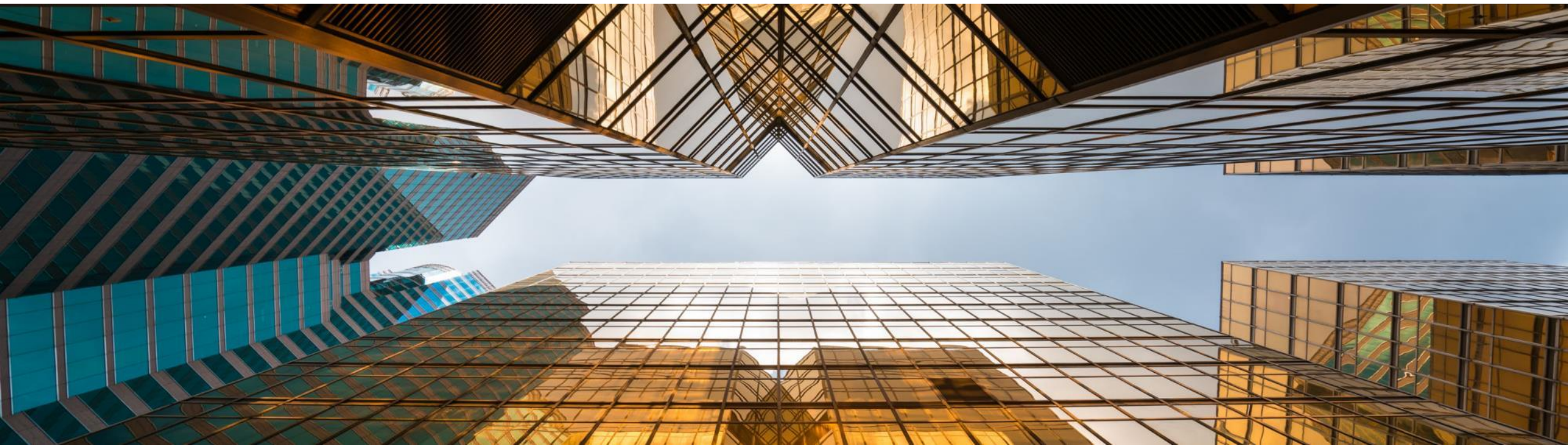
Managed BO with unmanaged save



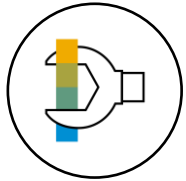
RAP development flow



SUMMARY



Summary – Key takeaways



The **ABAP RESTful Application Programming Model (RAP)** helps to efficiently and rapidly build enterprise-grade services with intrinsic built-in cloud qualities. RAP best support **SAP HANA** and **SAP Fiori elements**.

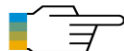


RAP is **available** on **SAP BTP ABAP Environment** and **SAP S/4HANA**, cloud and on-premise editions.

RAP is available SAP S/4HANA on-premise starting with edition 1909.



The **feature scope** of RAP is enhanced on a **quarterly** basis in SAP BTP ABAP Environment and SAP S/4HANA Cloud, and on a **yearly** basis in SAP S/4HANA on-premise.



Check out the [SAP Road Map Explorer](#) for updates.

Guidance for RAP usage in SAP S/4HANA on-premise releases

SAP S/4HANA 1909

FIRST RAP DELIVERY WITH LIMITED FEATURE SET

managed scenario for greenfield development **not** supported,
no draft support for SAP Fiori development

GUIDANCE

Delivery with SAP S/4HANA 1909

 Use the *ABAP Programming Model for SAP Fiori*

SAP S/4HANA 2020

BASIC RAP DELIVERY

incl. **managed** scenario (*with major limitations regarding key layout / numbering*),
draft support for managed or unmanaged BOs to build transactional SAP Fiori apps
OData V4 support for Web APIs and SAP Fiori UIs with FPS01

GUIDANCE

If limitations are crucial

 Use the *ABAP Programming Model for SAP Fiori*

If limitations are not critical

 Use RAP

SAP S/4HANA 2021 (planned)

MASS ADOPTION READY RAP DELIVERY

managed scenario now supports nearly all key layouts;
late numbering for a managed BO (with or without draft) in progress with FPS01
Enhanced testability, documentability and supportability support

GUIDANCE

 Exclusively use RAP



FREE openSAP COURSE



Building Apps with the ABAP RESTful Application Programming Model (RAP)



Self-paced mode

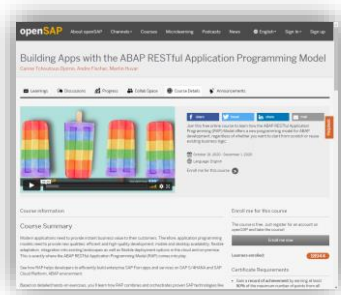


Week 1: Introduction
Week 2: Developing a Read-Only List Report App
Week 3: Enabling the Transactional Behavior of an App
Week 4: Dealing with Existing Code
Week 5: Service Consumption and Web APIs
Week 6: Final Exam



ENROLL NOW!

<https://open.sap.com/courses/cp13>



More information



Further information

[Getting started with the ABAP RESTful Application Programming Model \(RAP\)](#)

[Building Apps with the ABAP RESTful Application Programming Model](#) (openSAP Course)

[Testing Business Objects built with RAP](#)

Public SAP Web sites

ABAP Development community: www.sap.com/community/topic/abap.html

SAP BTP ABAP Environment community: <https://community.sap.com/topics/cloud-platform-abap-environment>

ABAP Testing and Analysis Community: <https://community.sap.com/topics/abap-testing-analysis>

SAP products: www.sap.com/products

SAP training and certification opportunities

www.sap.com/education

Thank you.

Contact information:

Carine Tchoutouo Djomo

Product Manager for ABAP Platform

carine.tchoutouo.djomo@sap.com

André Fischer

Product Manager for ABAP Platform

andre.fischer@sap.com