



# SAP Inside Track BA

## Modern ABAP Development Using BOPF

Agustin Pighin, SOVANTA AG  
July 28, 2017

PUBLIC

# Agenda

Introduction

Product Qualities

State-Less Applications

Architecture

Hands-On example

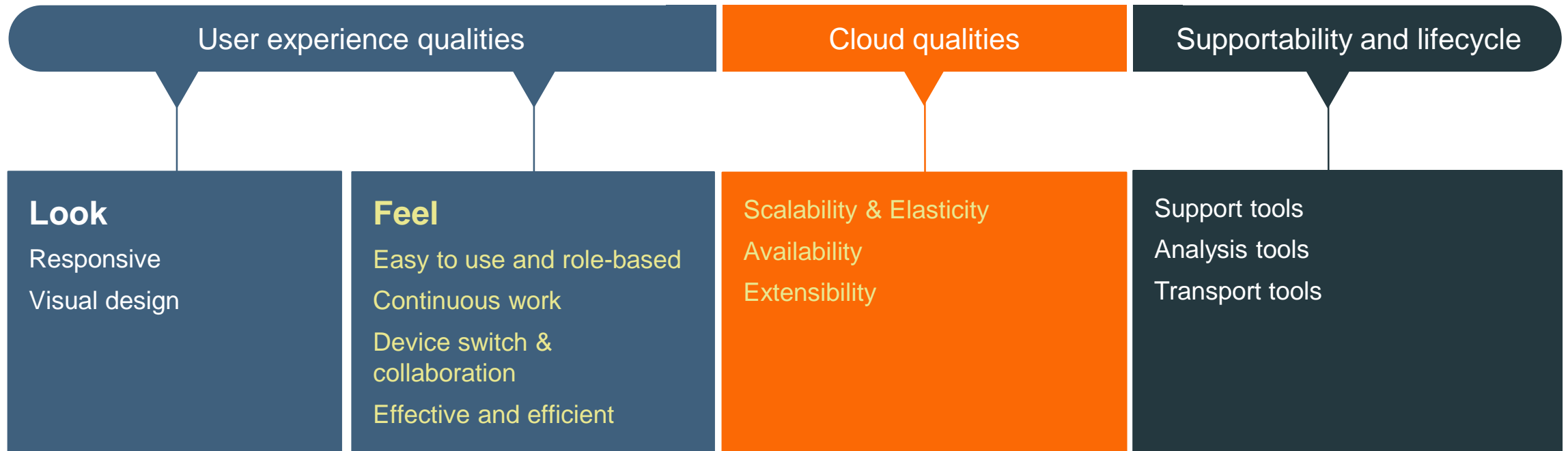
01

# Product Qualities

An Overview

# Product qualities

## Expectations on modern applications

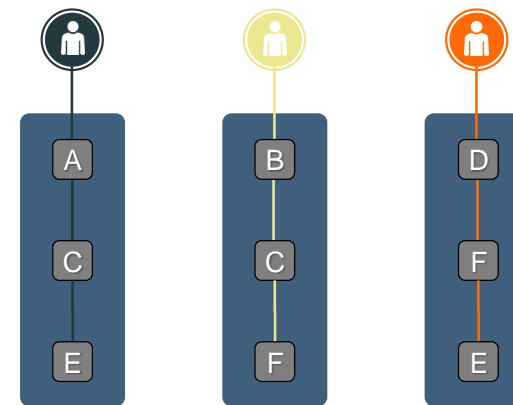
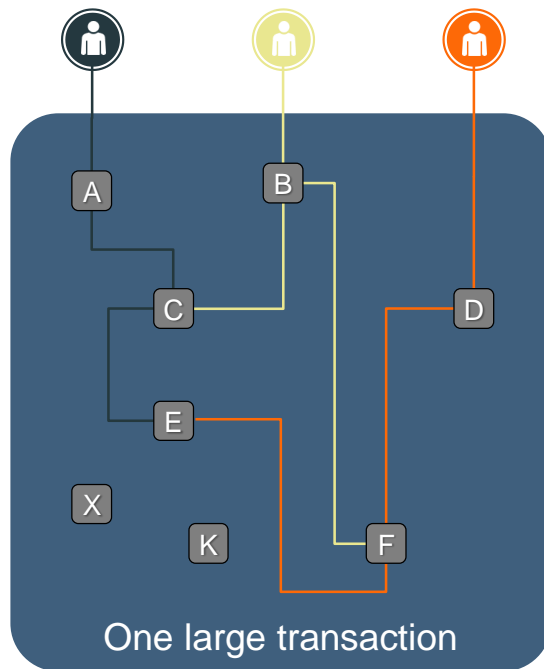


# User experience qualities

## Easy to use and role-based

### Requirement

I want an easy to use application providing only relevant functionality for the tasks of my role



**Multiple role-based and simple apps** but with redundant functionality and data access

### Impact on programming model

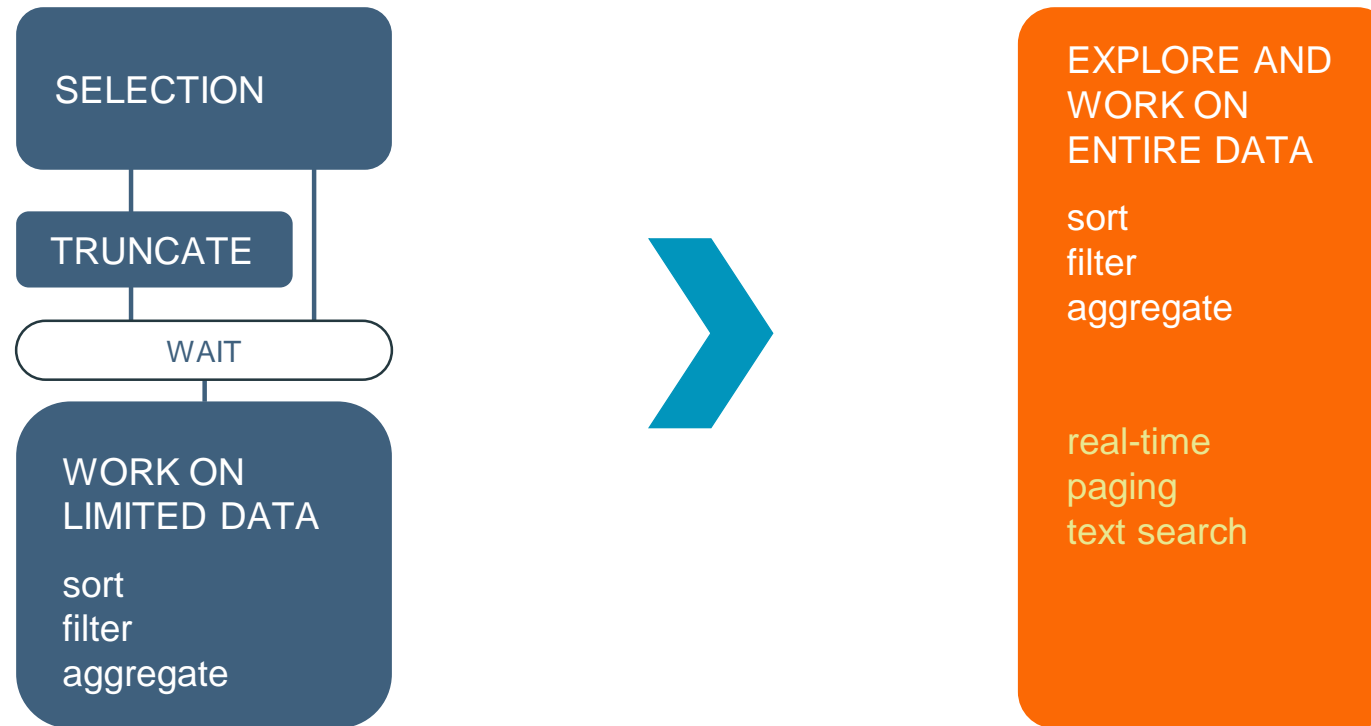
Efficient programming model to support multiple applications on same data and functionality  
Manage redundancies in applications without inconsistencies

# User experience qualities

## Effective and efficient work on entirety of data

### Requirement

I want the freedom to explore and work with entire data – not only on restricted or truncated data



### Impact on programming model

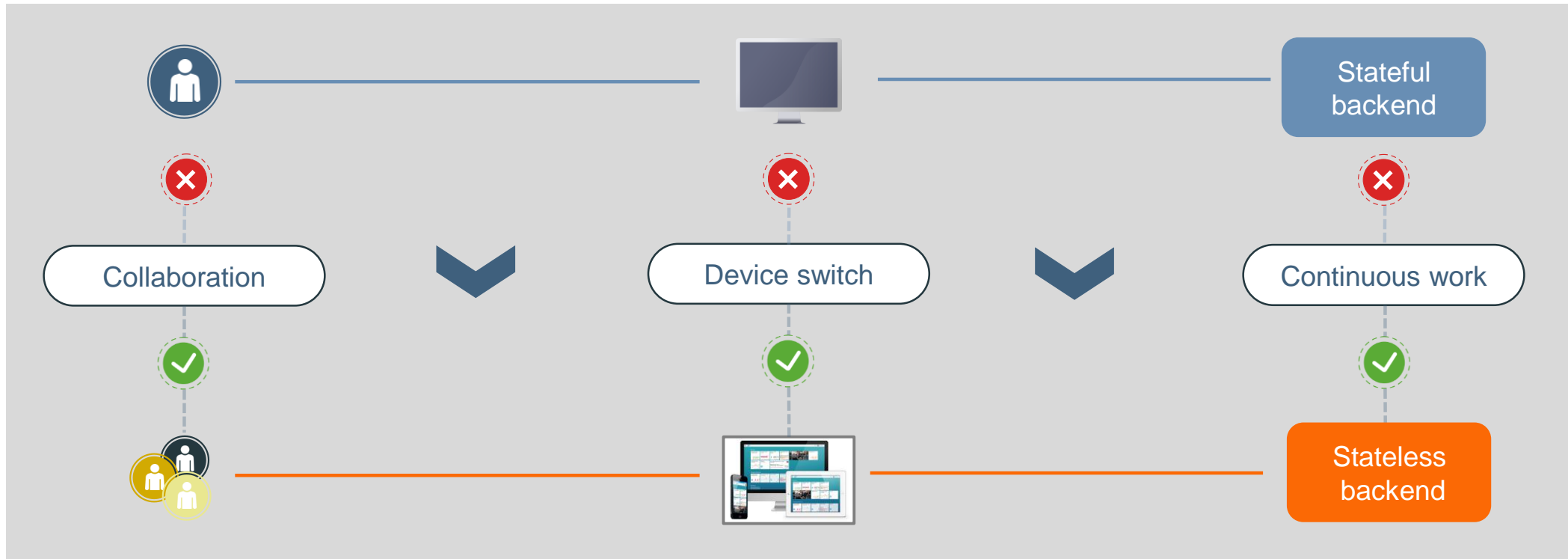
Easily enable navigation, filtering, text search and grouping of data for all applications  
Provide intrinsic support for exploiting SAP HANA features and performance

# User experience qualities

## Continuous work, device switch and collaboration

### Requirement

I want to work from everywhere, continue on interrupted work and change my devices when needed



### Impact on programming model

Remove tight coupling of client and server to avoid server stickiness

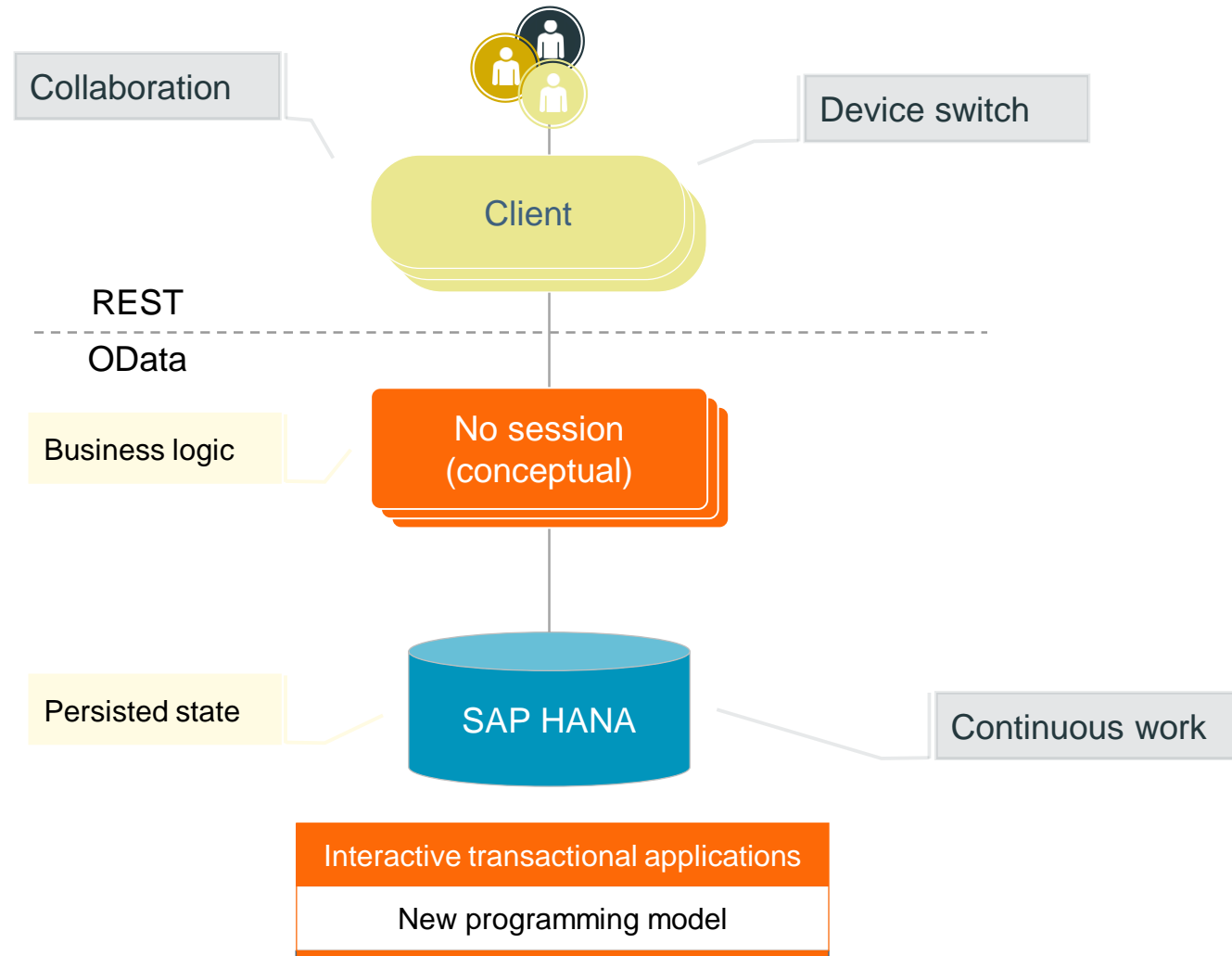
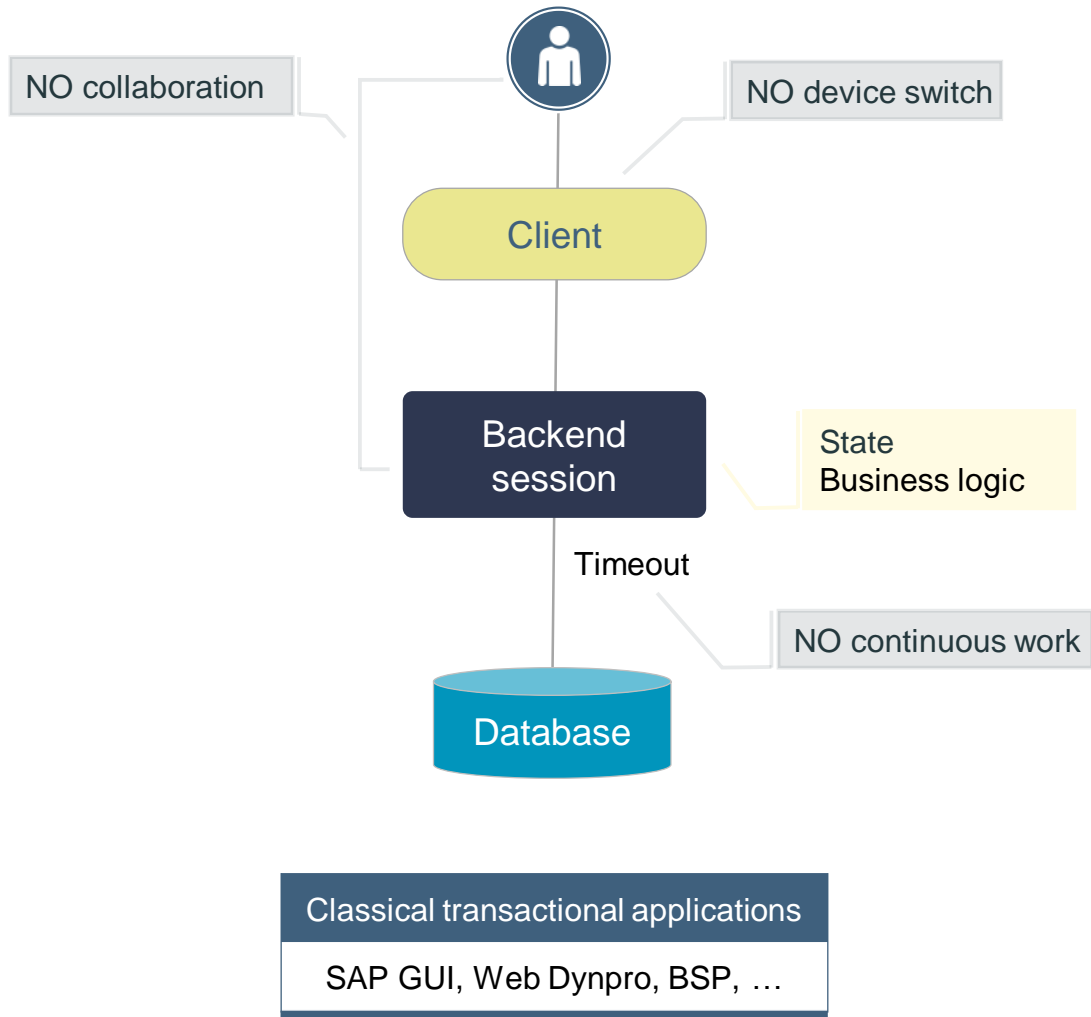
02

**Stateless Apps**



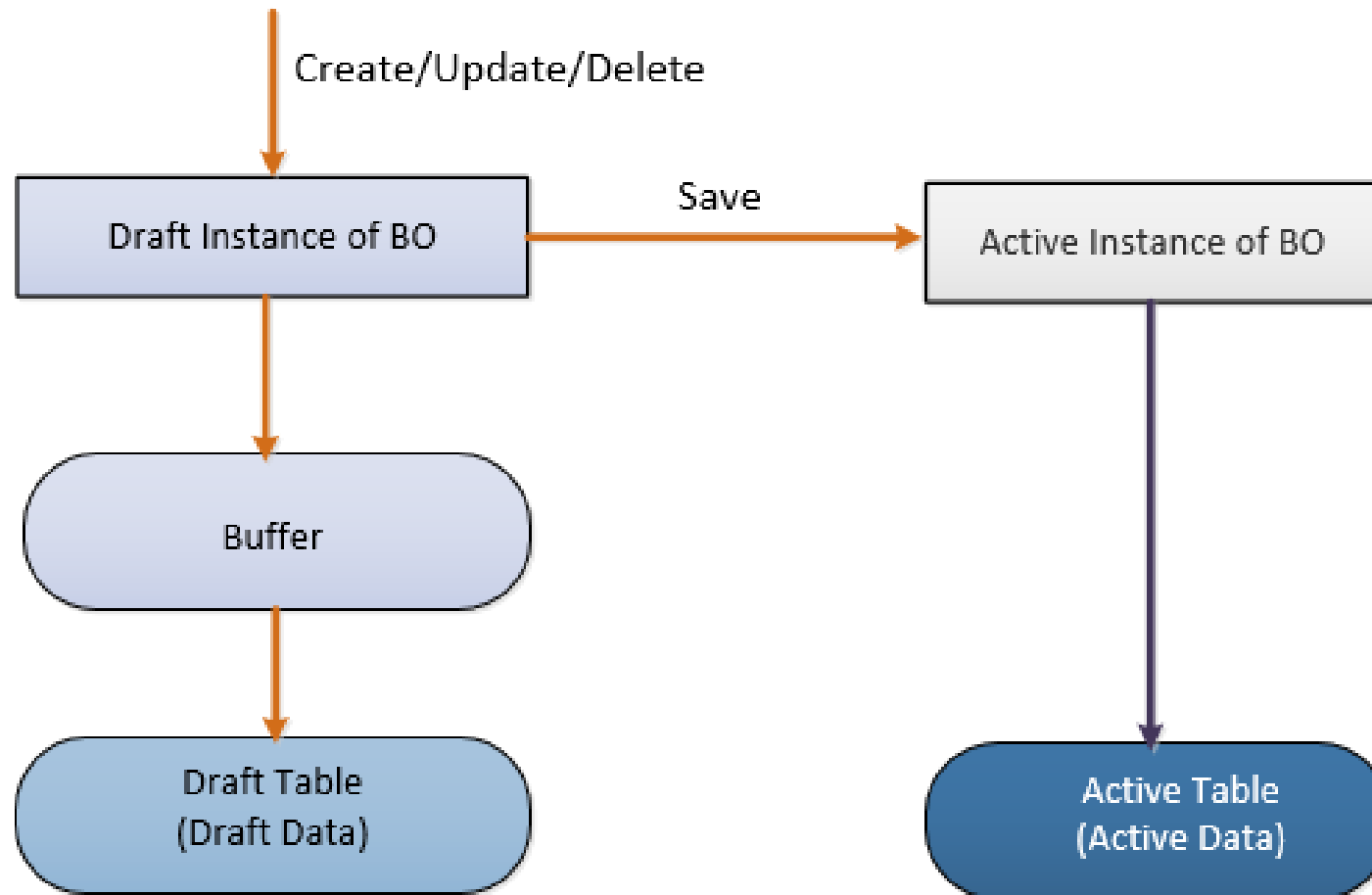
# Impact of a RESTful architecture

## Impact on session, state and business logic



# Draft support

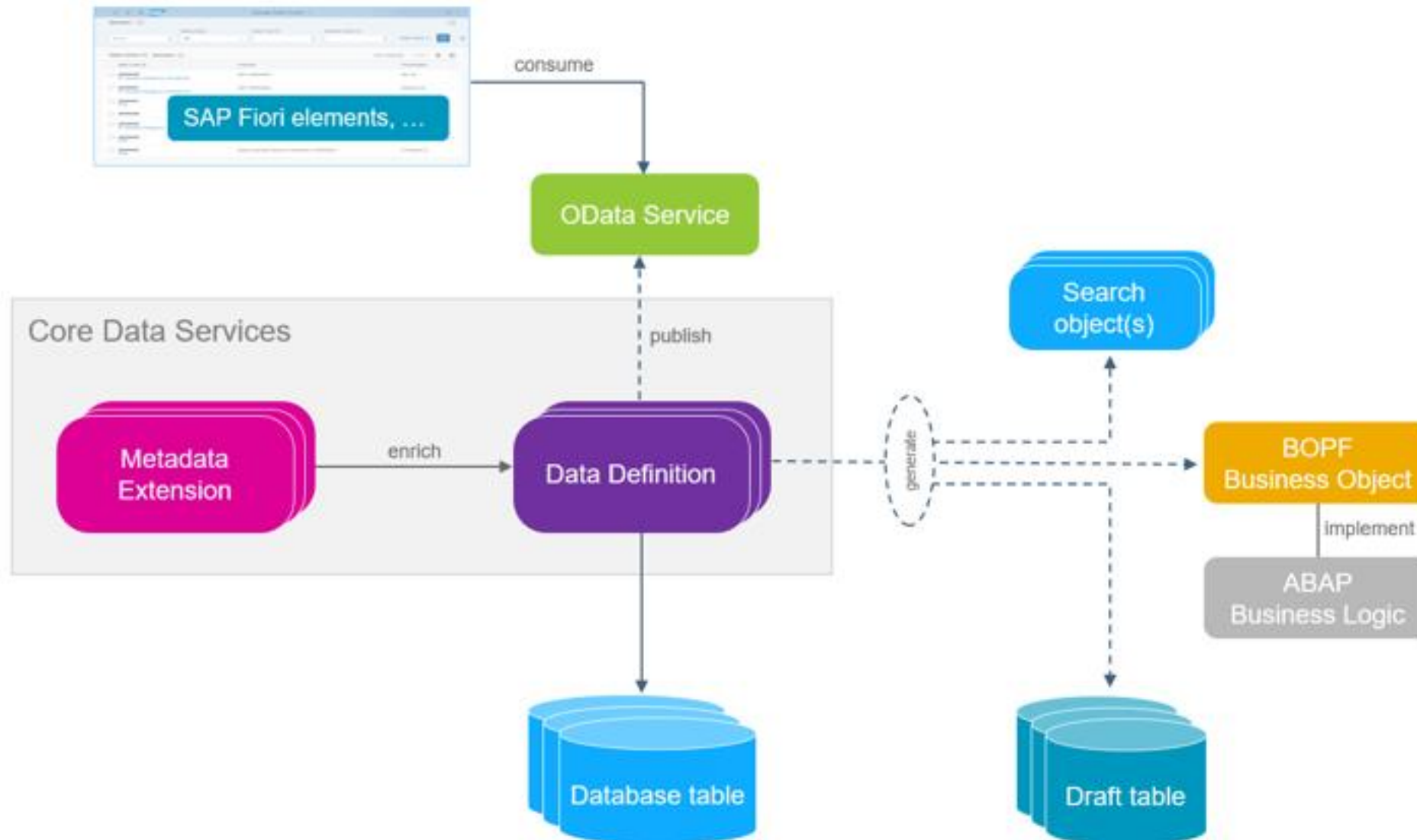
## Active vs draft instances



03

**Architecture**

# Involved technologies



# Advanced data model definition and access

## ABAP Core Data Services (CDS)



### Capture business intent

Reduced complexity  
Extending SQL



### Common data model

Semantically rich  
Declarative  
Close to conceptual thinking

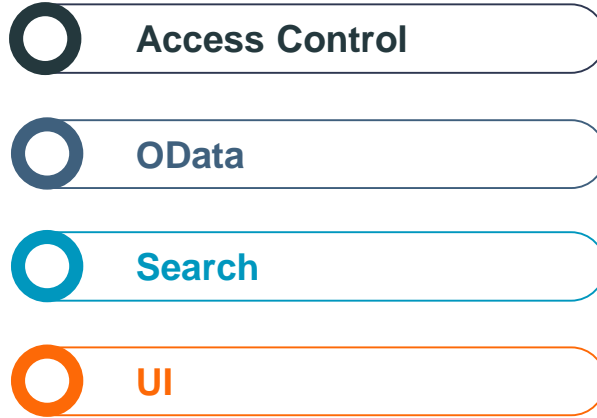


### Improved programming model

For all application domains  
Rich set of build-in functions  
Code pushdown capabilities

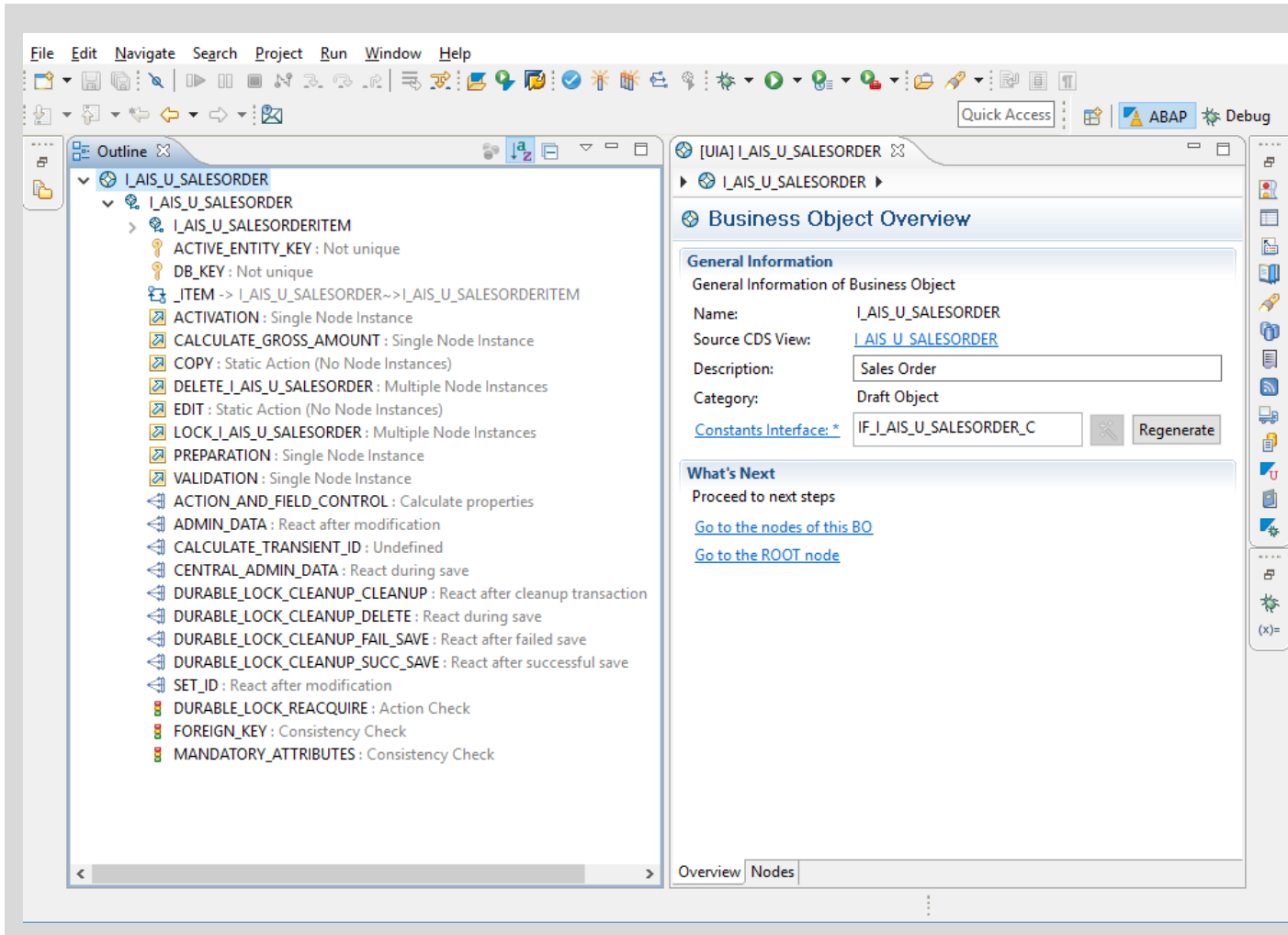
# Contributions to SAP S/4HANA

## SAP S/4HANA applications – Example CDS view with annotations



```
[UIA] ZSALES_ORDER_ITEMS_DDL_SRC [UIA] ZDEMO_DDL_SALES_ORDER_ITEM [UIA] SEPM_CDS_SALES_ORDER_ITEM [UIA] SEPM_CDS_SALES_ORDER [UIA] ZSALES_ORDER_ITEMS
1 @AbapCatalog.sqlViewName: 'ZSQL_SOIL_ADV'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #CHECK
4 @EndUserText.label: 'Advanced List Reporting SO Items'
5
6 @OData.publish: true
7
8 @Search.searchable: true
9
10 @UI.headerInfo: { typeName: 'Sales Order Item', typeNamePlural: 'Sales Order Items' }
11
12
13 define view ZDEMO_SALES_ORDER_ITEM_ADV as
14     select from sepm_cds_sales_order_item as Item {
15         @UI.selectionField.position: 10
16         @UI.lineItem: { importance: #HIGH, label: 'Sales Order', position: 10 }
17         -- @UI.identification: { importance: #HIGH, position: 10 }
18         key Item.sales_order.sales_order_id as SalesOrderID,
19         key Item.so_item_pos as ItemPosition,
20
21         ---@UI.statusInfo.position: 20
22         @UI.lineItem.position: 40
23         Item.sales_order.lifecycle_status as LifecycleStatus,
24
25         @UI.lineItem.position: 90
26         @UI.selectionField.position: 20
27         @Search.defaultSearchElement: true
28         @Search.fuzzinessThreshold: 0.7
29         Item.sales_order.customer.company_name as CustomerName,
30     }
```

# Role of BOPF in programming model



Enables model-based and standardized development approach

Manages transactional data of draft and active instances within one business object

CRUDE

Locking

Authorization

Orchestrates business logic in well defined exits like

actions

determinations

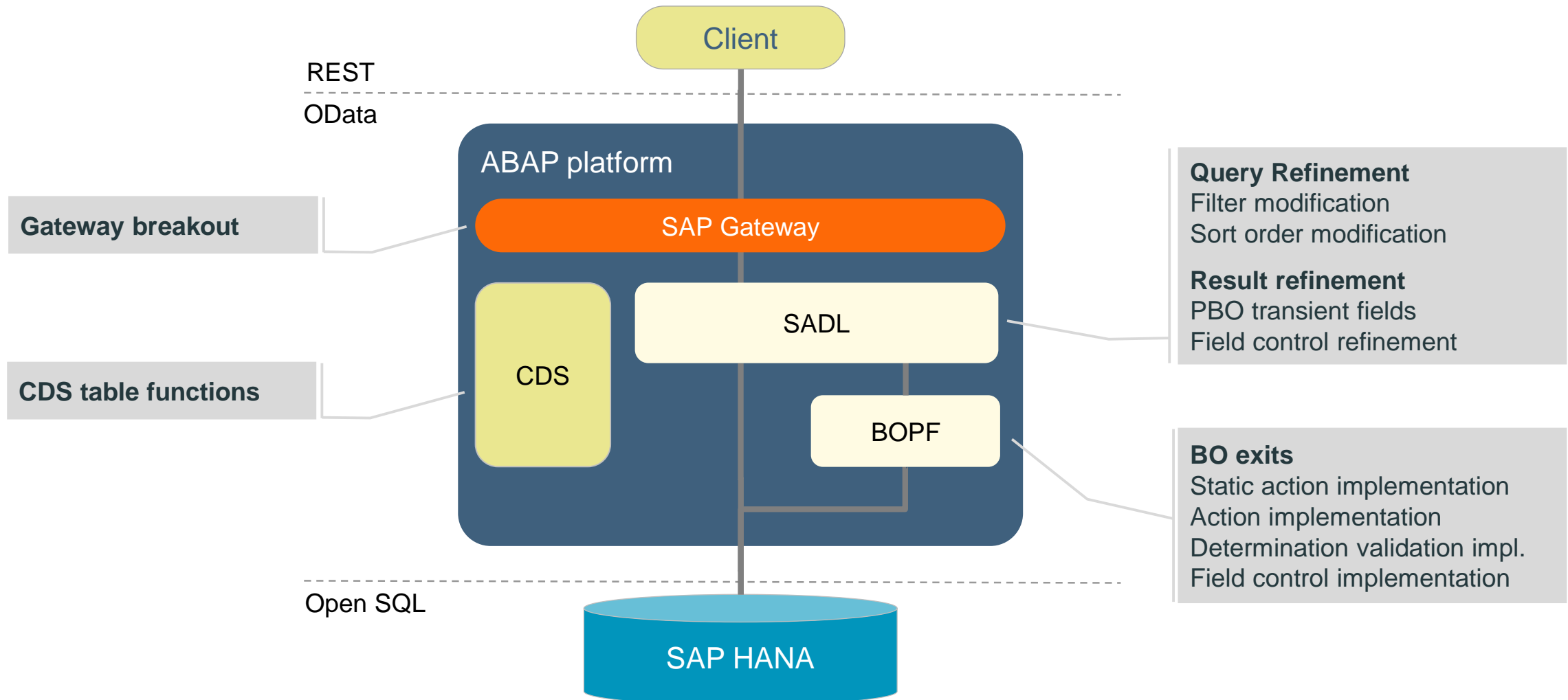
validations

draft handling



# ABAP programming model for SAP Fiori and SAP S/4HANA

## Runtime stack – Exits



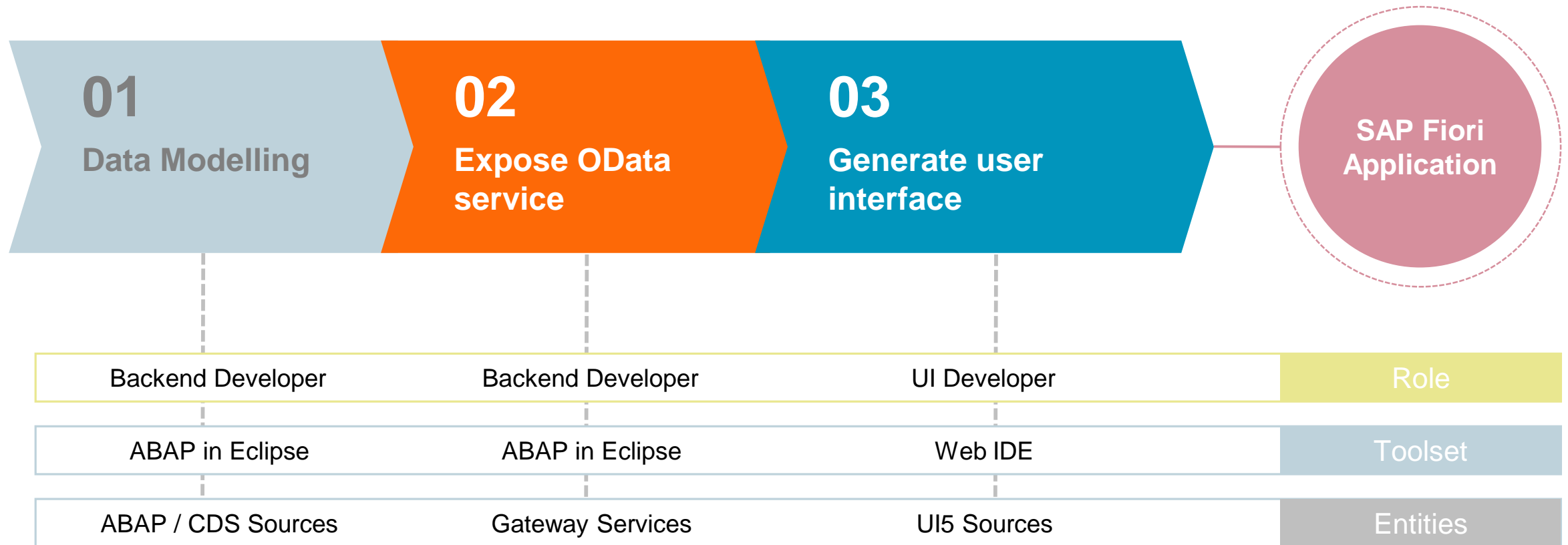


04

**Build an App from  
Scratch**

# Programming model for SAP Fiori

## Major steps and development tools



# BUILDING AN APP FROM SCRATCH

Get the idea

The screenshot displays the SAP Fiori 'Sales Order Invoices' app interface. It features a search bar at the top with the text 'Standard' and a search icon. Below the search bar are input fields for 'Company:', 'Business Partner ID:', and 'Sales Order ID:'. The main content area shows a list of sales order invoices with columns for 'Business Partner ID', 'Sales Order ID', 'Open Days', and 'Gross Amount'. The list is filtered to show 282,524 records. The interface also includes a 'Hide Filter Bar' button and a 'Go' button. A large orange overlay on the right side of the screenshot contains the text: 'INTERACTIVE TRANSACTIONAL SAP FIORI APP INCLUDING LIST REPORTING'.

Business Partner ID	Sales Order ID	Open Days	Gross Amount
> Company: AVANTEL		27	111,176,838.77 EUR
> Company: African Gold And Diamond Corporation		26	20,519,057.99 EUR
> Company: Alpine Systems		30	24,602,729.34 EUR
> Company: Anav Ideon		21	81,023,573.58 EUR
> Company: Angéré		27	141,678,453.42 EUR
> Company: Asia High tech		25	125,063,847.97 EUR
> Company: Baleda		23	65,306,896.48 EUR
> Company: Bionic Research Lab		22	61,202,508.24 EUR
> Company: Brazil Technologies		22	197,453,817.74 EUR
> Company: Compostela		24	6,705,630.36 EUR

We want to create a sales order overview with **analytics**, **search** and **transactional capabilities**.

05

Final

# Summary

## Key message

### New programming model

New programming model  
available but still evolving

Documentation available

➡ Search for  
[‘ABAP Programming model Fiori’](#)

### Availability of app types

Read-only ➡ ABAP 7.50

‘Batch input’ ➡ Planned for ABAP 7.51

Interactive /  
Draft ➡ Planned for ABAP 7.51 SP01

# Resources

[SAP – Abap Programming Model for SAP Fiori](#)

[BOPF API Reference \( Advance \)](#)