

BOPF Enterprise Procurement Model (EPM)

SAP AG, 2014



Disclaimer

This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document 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. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.

BOPF EPM

Agenda

1. Introduction

2. Business Process

1. Order Process
2. Example Order Process Execution

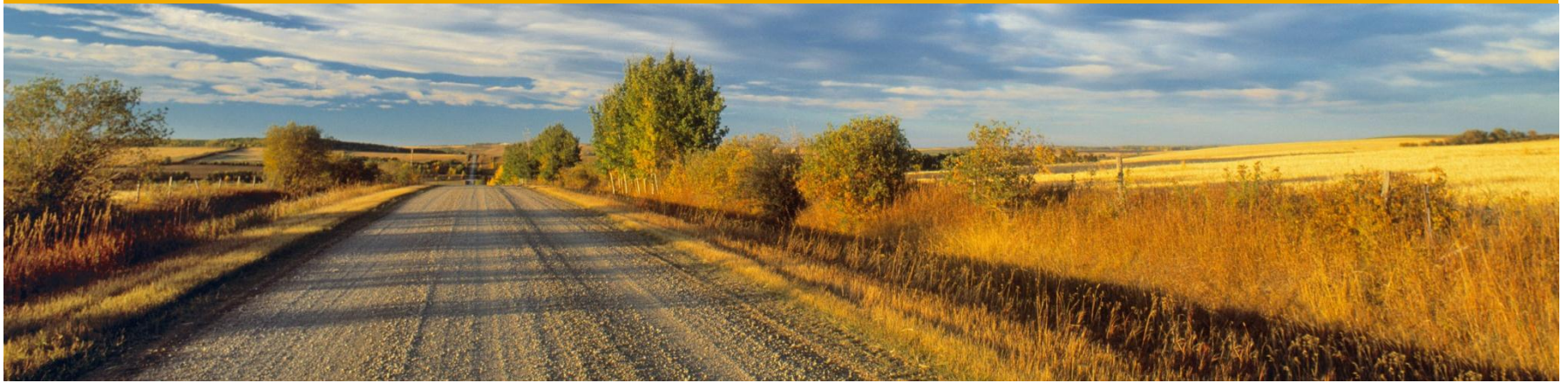
3. Business Objects

1. Business Partner
2. Product
3. Sales Quote
4. Purchase Order
5. Sales Order
6. Sales Invoice

4. Data Generator

nun habe ich endlich ein wenig Zeit gefunden, um mir das BO /BOBF/EPM_SALES_ORDER genauer anzuschauen:

1. ROOT: Die BUYER_ID ist eigentlich eine BUYER_PARTY_ID, da die BUYER_ID die externe Referenz ist. Zusätzlich würde ich noch die BILL_TO und SHIP_TO PartyID aufnehmen (**erledigt**). Optional könnte man noch über die zusätzliche Attribute wie Lieferpriorität oder Absagegrund nachdenken.
2. ITEM: Statusfelder fehlen vollkommen, oder wird nur das Szenario der Komplettlieferung abgebildet? Das wäre aus bwl. Sicht nicht richtig, da der Business Prozess auf Positionsebene stattfindet. Abweichende Empfänger halte ich für optional.



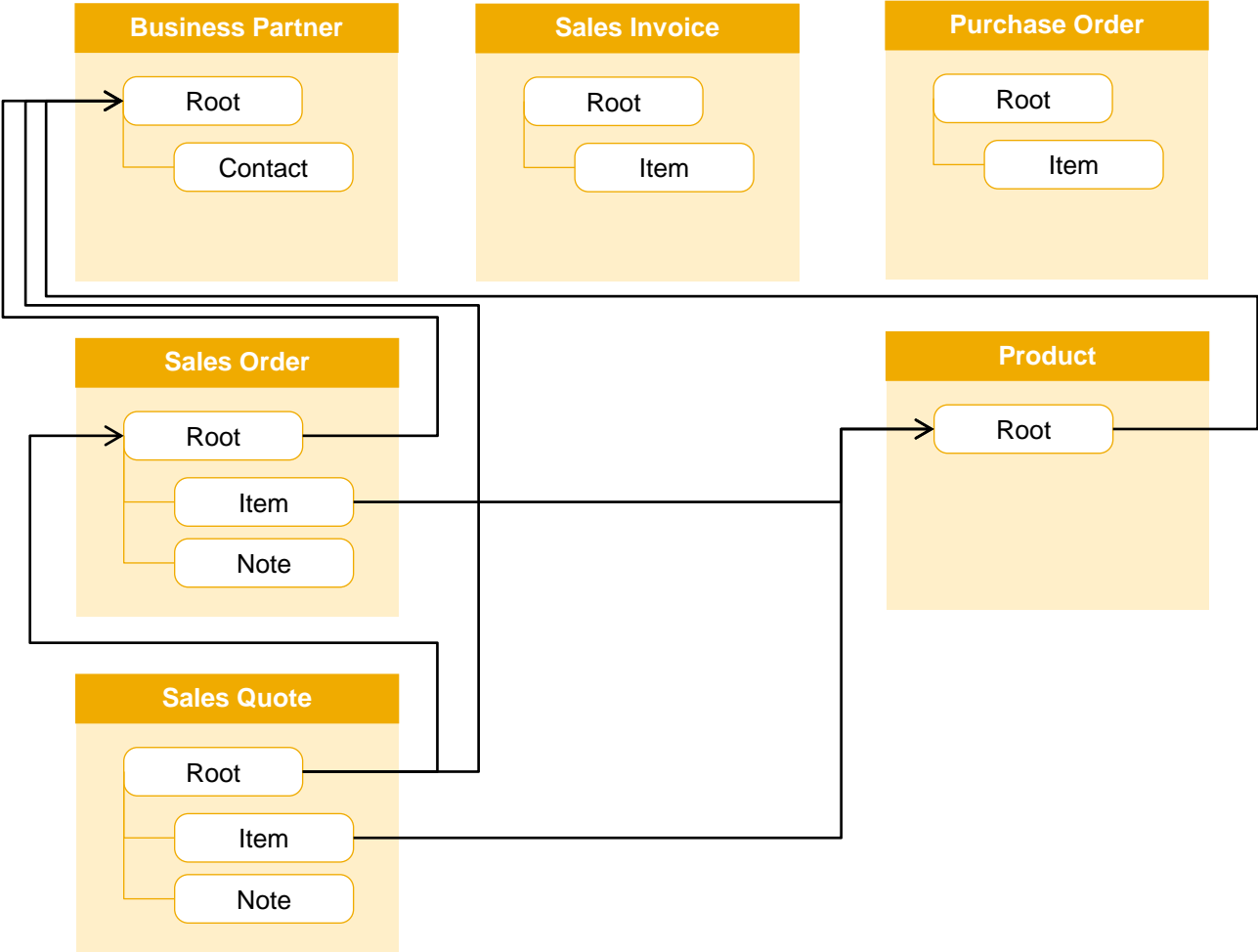
Introduction

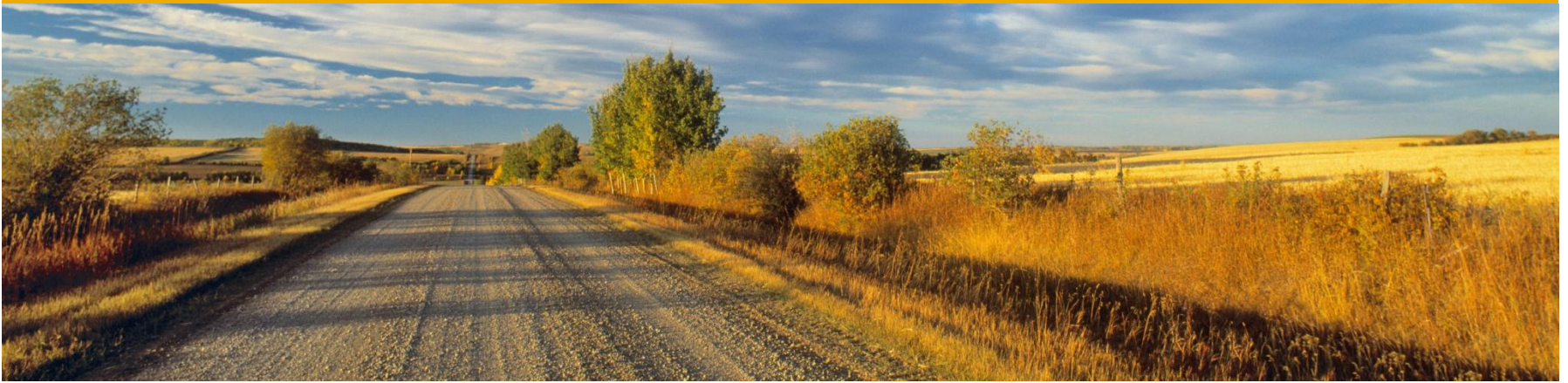
Introduction Enterprise Procurement Model (EPM)

The BOPF EPM is a demo application delivered to customers. It is based on the Netweaver EPM scenario. However its focus is on the core processes and process steps. It has the following purposes:

- Examples of BOPF features and their corresponding implementation
- Demonstration of integration scenarios with BOPF
- Example of scenario and unittesting of BOPF applications

Application Overview



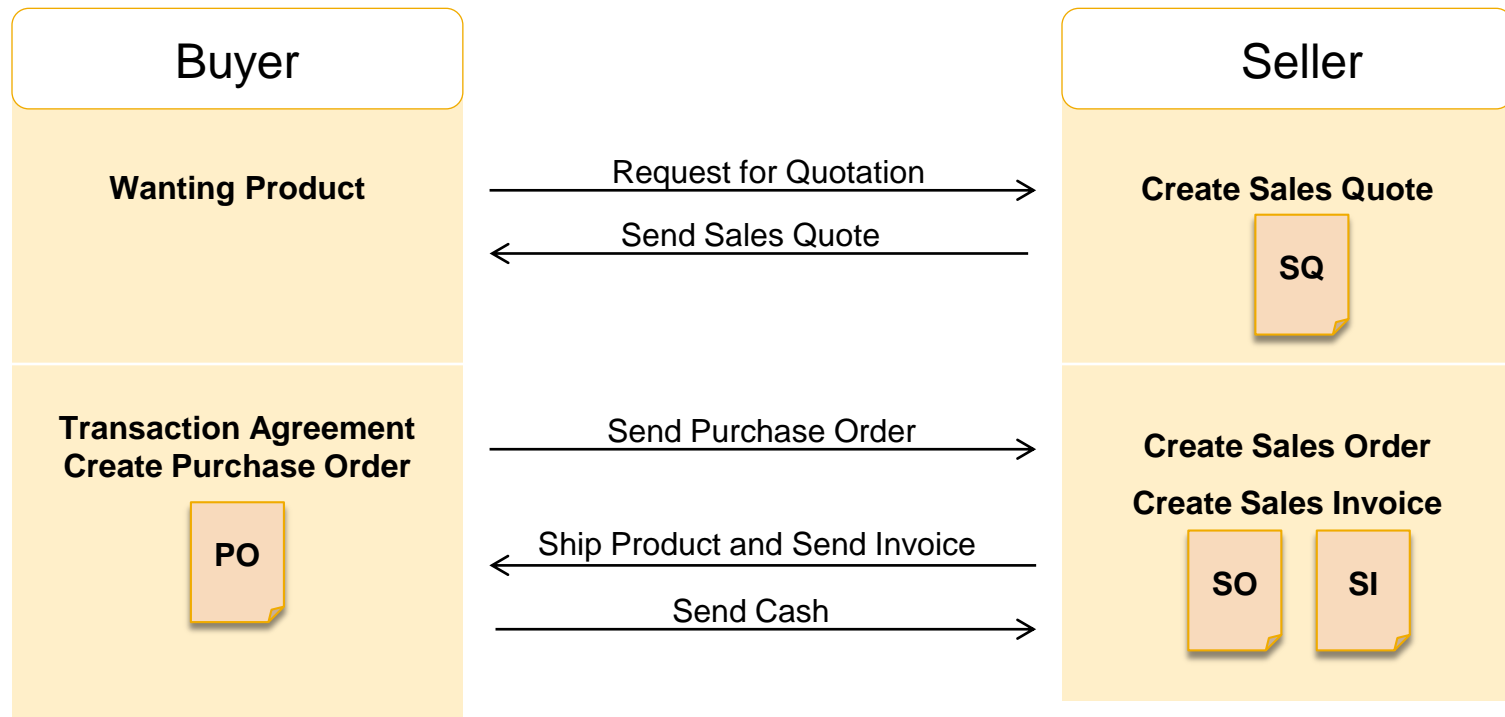


Business Processes

Order Process Description

The EPM applications demonstrates a typical commercial order process. The participating buyer has an intention to buy products from the seller. To be informed about pricing and conditions, a sales quote is requested. Afterwards, the transaction is agreed by creating a purchase order. The seller creates a sales order and the corresponding invoice and delivers the product.

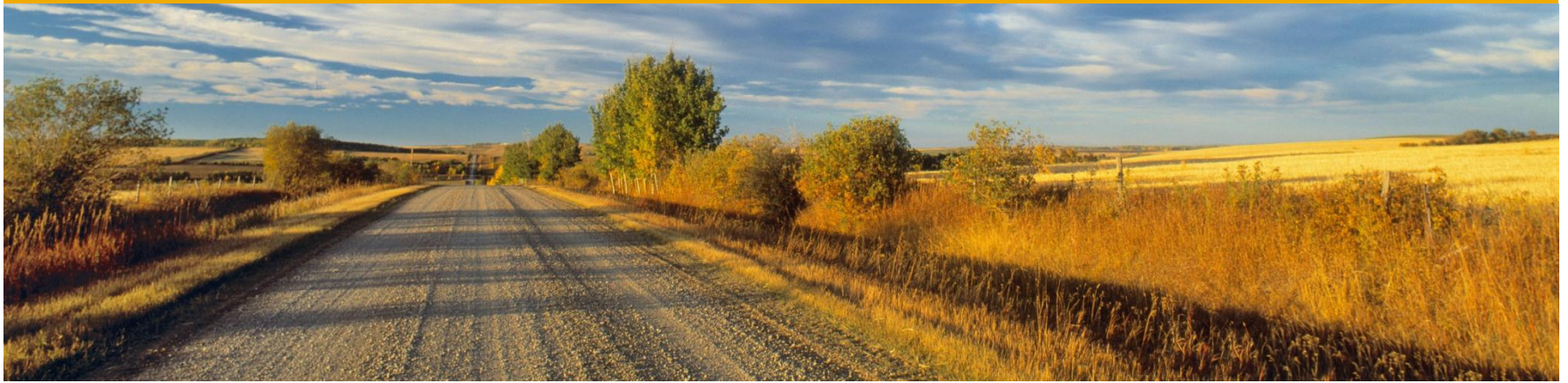
Order Process Detail



- All Business Documents (like e.g. Sales Quote) are represented by an own Business Object
- Buyer and Seller are instances of the “Business Partner” Business Object
- The products shipped to buyer are instances of a “Product” Business Object

Example Order Process Execution

Screenshots of processing the order process in the Test UI. To be done as soon as the EPM BOs are completely implemented.

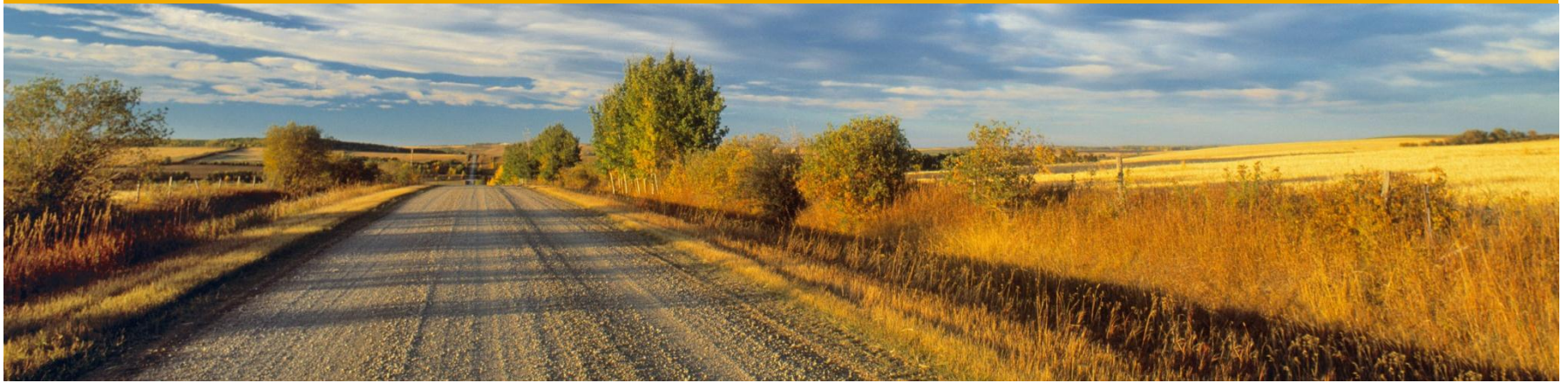


Business Objects

Business Object Overview

- **Business Partner:** /BOBF/EPM_BUSINESS_PARTER
- **Product:** /BOBF/EPM_PRODUCT
- **Sales Quote:** /BOBF/EPM_SALES_QUOTE
- **Purchase Order:** /BOBF/EPM_PURCHASE_ORDER
- **Sales Order:** /BOBF/EPM_SALES_ORDER
- **Sales Invoice:** /BOBF/EPM_SALES_INVOICE

The EPM applications consists of several Business Objects located in sub packages of package “/BOBF/EPM”.



Business Partner

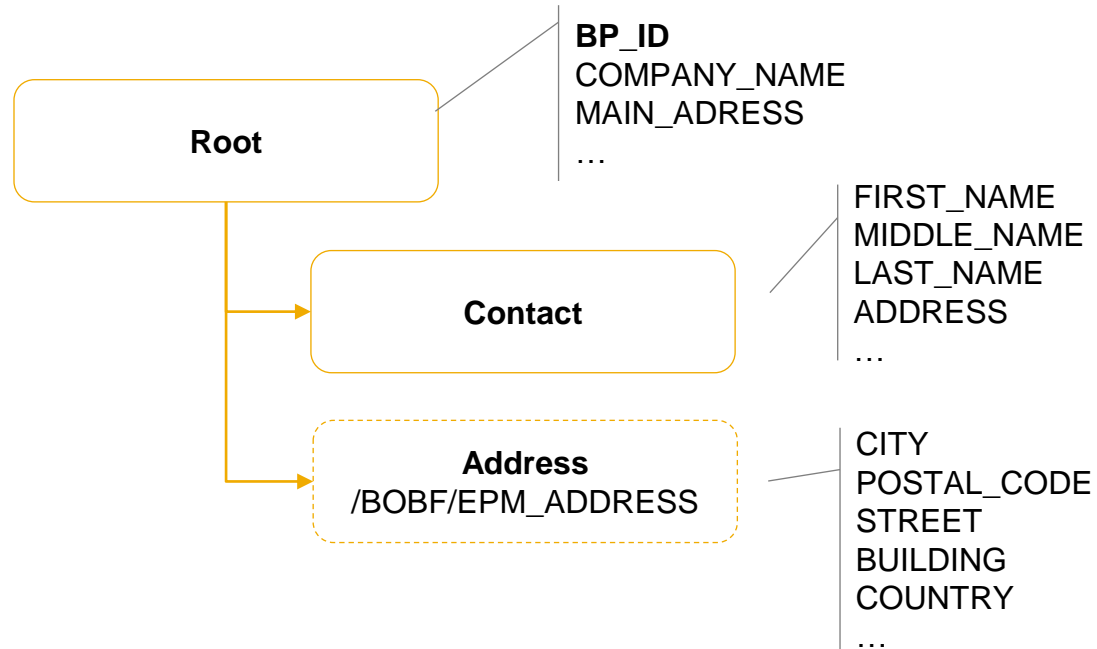
/BOBF/EPM_BUSINESS_PARTNER

Business Partner

Description

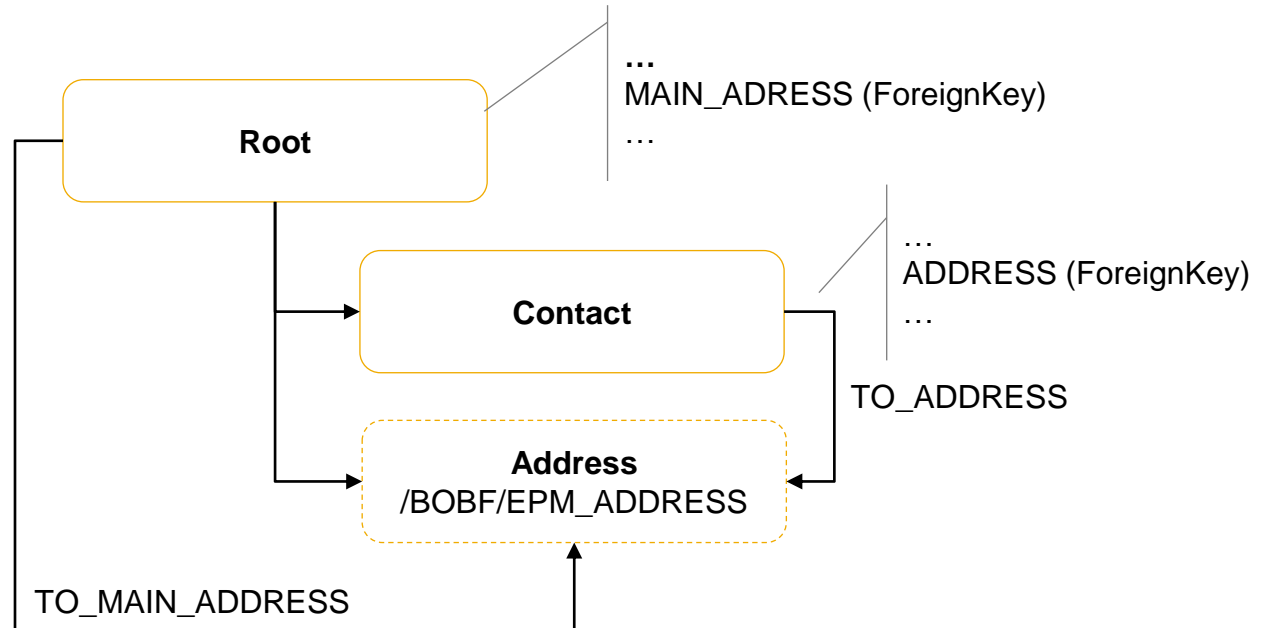
The business partner business object (/BOBF/EPM_BUSINESS_PARTNER) stores all business partners involved in the scenario. Each partner is represented by a business object instance and has a unique ID (BP_ID). For instance both the seller and the buyer are represented as instances of this business object.

Business Partner Nodes



- **Root:** The root node contains the business partner's company details.
- **Contact:** The contact person and its information is stored in the CONTACT node.
- **Address:** The address dependent object (BOBF/EPM_ADDRESS) is included to manage the address data of the company and the contact person.

Business Partner Association Model



- **TO_MAIN_ADDRESS:** There can be multiple addresses belonging to the same business partner. However, one address is the primary contact address and thus stored in the root element MAIN_ADDRESS.
- **TO_ADDRESS:** Each contact person can have exactly one address assigned and this association navigates to that one.

Business Partner

Business Logic

- **CREATE, RETRIEVE , UPDATE, DELETE**
- **SELECT_ALL, SELECT_BY_ELEMENTS**
- **ADMINISTRATIVE_DATA**: This determination automatically writes the last change date and user as soon as the transaction is saved
- **ALTKEY_UNIQUENESS_CHECK**: This validation entity on the root node ensures that each business partner ID is unique while creating new business partners. In the eclipse design time, this is achieved by a configuration setting of the alternative key.
- The property determination **ROOT_PROPERTY** ensures that business partner IDs can't be updated after the creation but only deleted and newly created.



Product

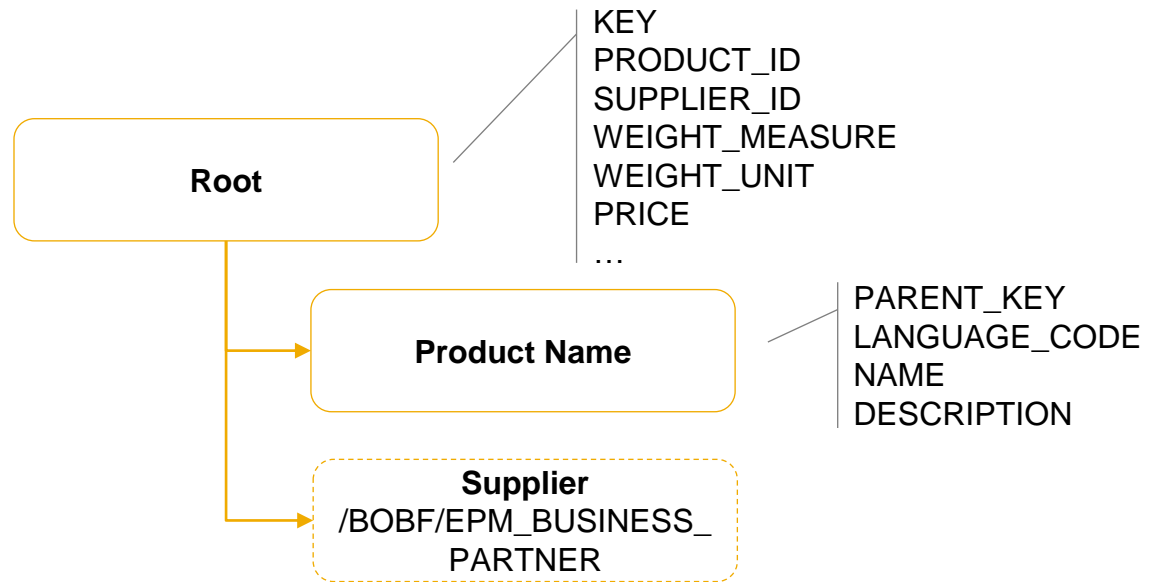
/BOBF/EPM_PRODUCT

Product

Description

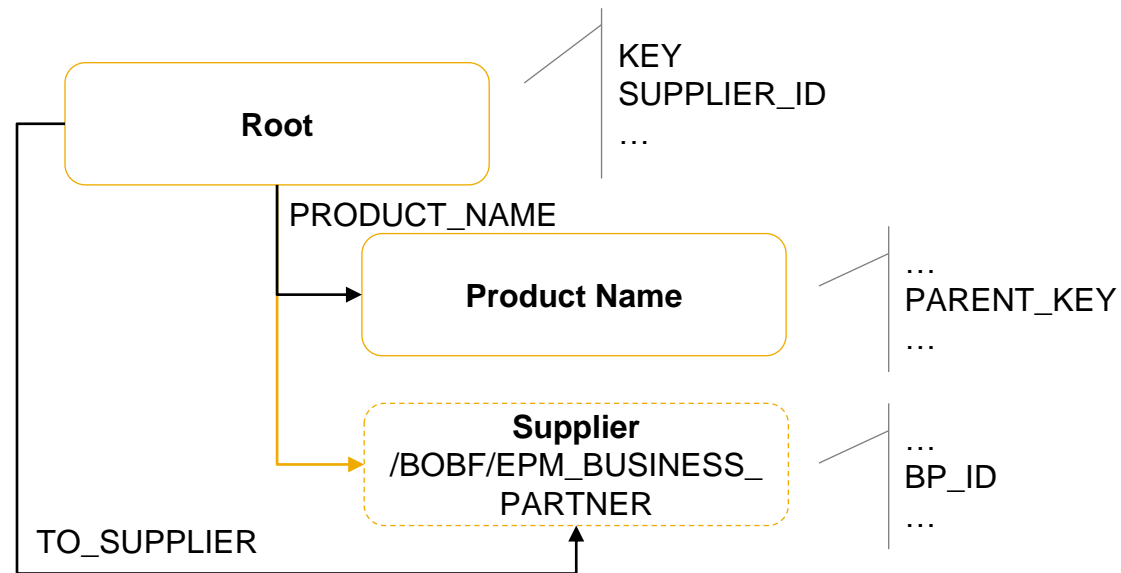
The Product Business Object (/BOBF/EPM_PRODUCT) stores all products which are offered by the application. Each product is represented by a business object instance and has a unique business partner ID. All of its details like for instance price and weight are also stored.

Product Data Model



- **ROOT:** Contains detail information about a product
- **PRODUCT_NAME:** Persists the name of the product in different languages
- **SUPPLIER:** Object representation node representing the business partner which supplies the product.

Product Association Model

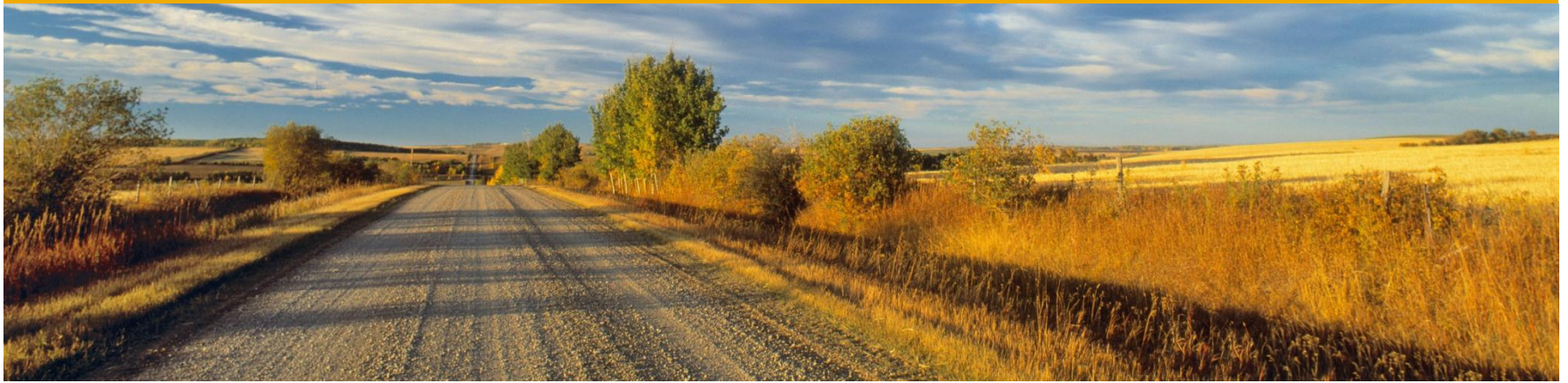


- **TO_SUPPLIER:** This association leads from a product root instance to the corresponding root instance of the business partner business object. The relationship is based on the foreign key “supplier id”
- **PRODUCT_NAME:** This association navigates to the product name node instance containing the product name in the user’s language. It is implemented by the library class `/BOBF/CL_LIB_C_LANGUAGE`.

Product

Business Logic

- **CREATE, RETRIEVE , UPDATE, DELETE**
- **SELECT_ALL, SELECT_BY_ELEMENTS**
- The **ADMINISTRATIVE_DATA** determination automatically writes the last change date and user of a product as soon as the transaction is saved
- There is a validation **ALTKEY_UNIQUENESS_CHECK** on the root node ensuring that each business product ID is unique while creating new product instances. In the eclipse design time, this is achieved by a configuration setting of the alternative key.
- The property determination **ROOT_PROPERTY** ensures that product IDs can't be updated after the creation but only deleted and newly created.
- The consistency validation **CHECK_BUSINESS_PARTNER** ensures, that if an invalid business partner ID is maintained, a corresponding message comes and the save is rejected.



Sales Invoice

/BOBF/EPM_SALES_INVOICE

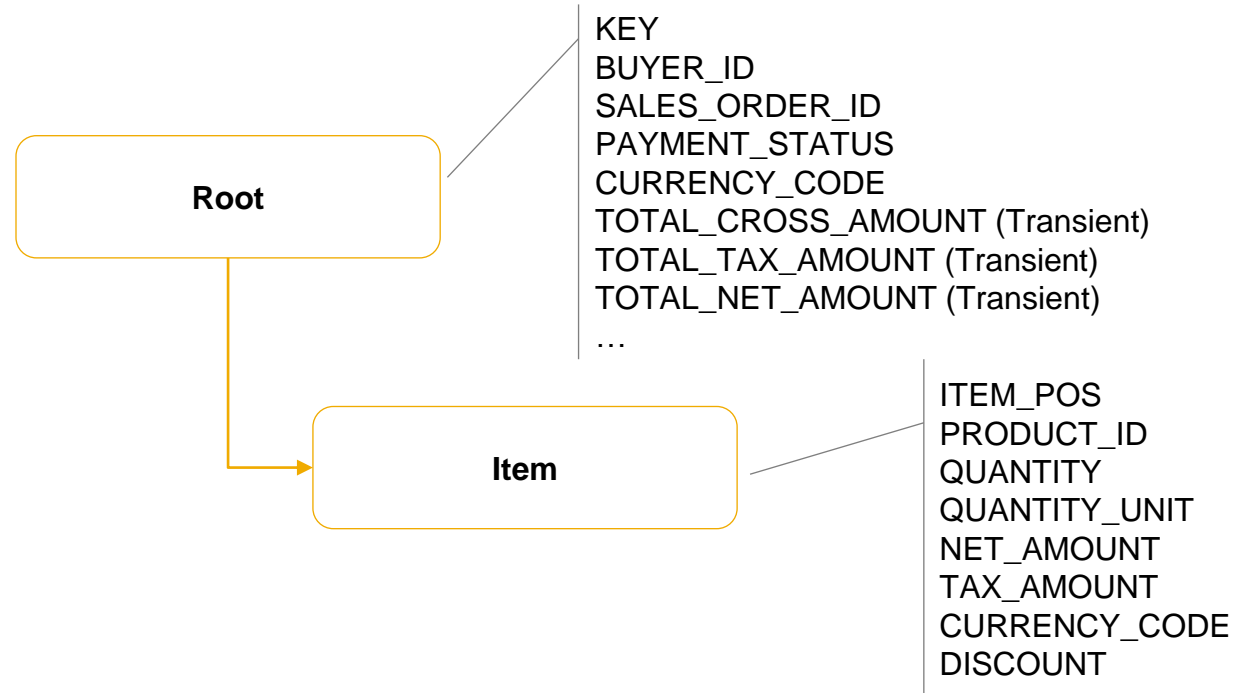
Sales Invoice

Description

The Sales Invoice Business Object (/BOBF/EPM_SALES_INVOICE) is a business document to be sent to the buyer. It contains a copy of the items of the sales order to which it is assigned to.

Sales Invoice

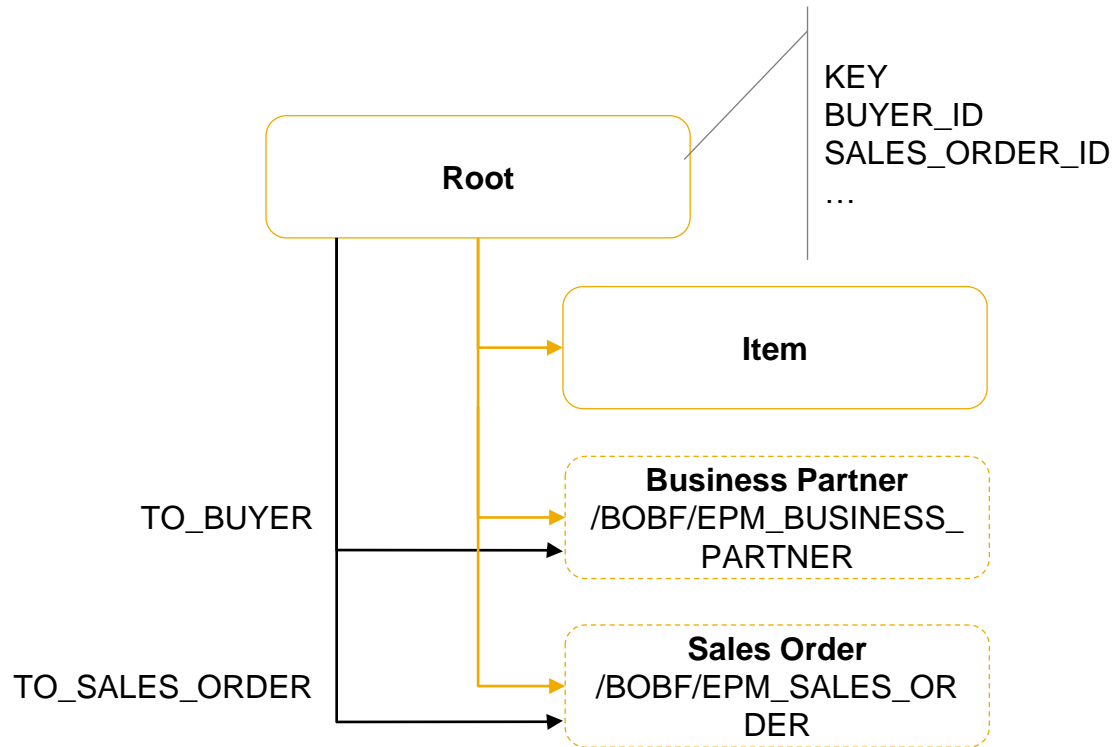
Data Model



- The invoice header data is stored in the ROOT node. The affected business partner and sales order, to which the invoice belongs to are persisted as foreign keys.
- Invoice items are stored in the ITEM node. Also the fixed price, quantity and amount are part of this node.

Sales Invoice

Navigation Model



- **TO_BUYER:** This association leads from the sales invoice to corresponding buyer.
- **TO_SALES_ORDER:** This association lead to the corresponding sales order.

Sales Invoice

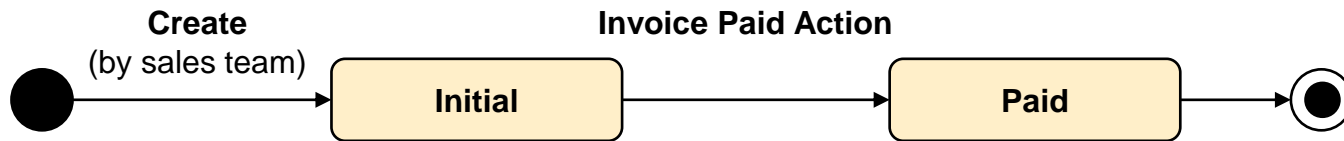
Business Logic

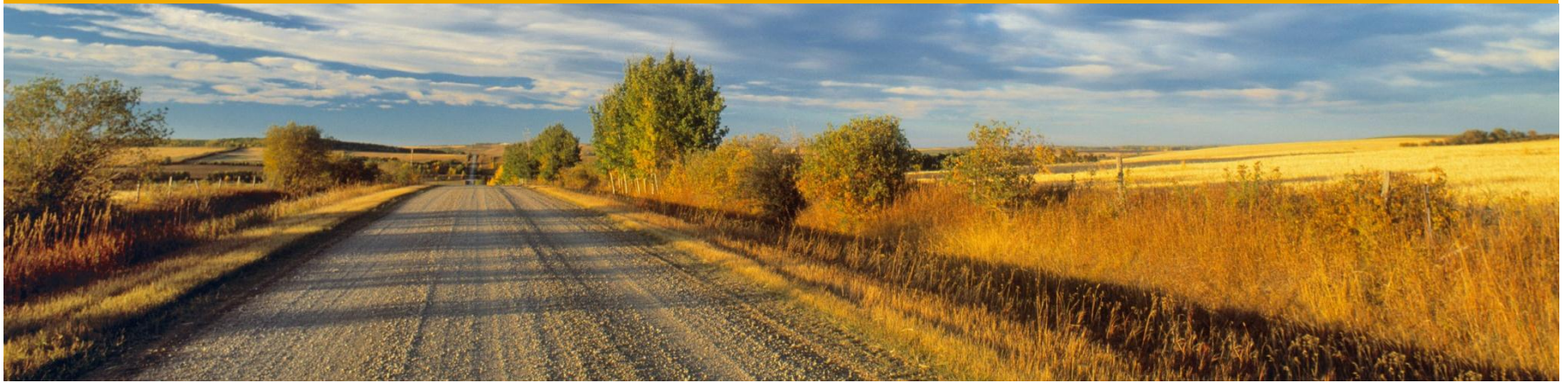
- **CREATE, RETRIEVE (only ROOT node: UPDATE, DELETE)**
- **SELECT_ALL, SELECT_BY_ELEMENTS**
- The **ADMINISTRATIVE_DATA** determination automatically writes the last change date and user of a sales invoice as soon as the transaction is saved
- There is a validation **ALTKEY_UNIQUENESS_CHECK** on the root node ensuring that each sales invoice ID is unique while creating new invoice instances. In the eclipse design time, this is achieved by a configuration setting of the alternative key.
- The property determination **ROOT_PROPERTY** ensures that after an invoice is saved, the invoice including the item instances are update and delete disabled. Furthermore, it is not possible to create further items.
- The determination **UPDATE_SALES_ORDER** is invoked as soon as an invoice is created and saved for the first time to inform the corresponding sales order about this.
- After an invoice is saved, the action **MARK_AS_PAID** can be called by indicate that the invoice has been paid.
- The determination **COPY_SALES_ORDER_ITEM** is triggered as soon as the sales order id is assigned to the ROOT node. It deletes all existing item instances and copies all items from the sales order.

Sales Invoice

State Model

Payment Status (on ROOT node)





Sales Order

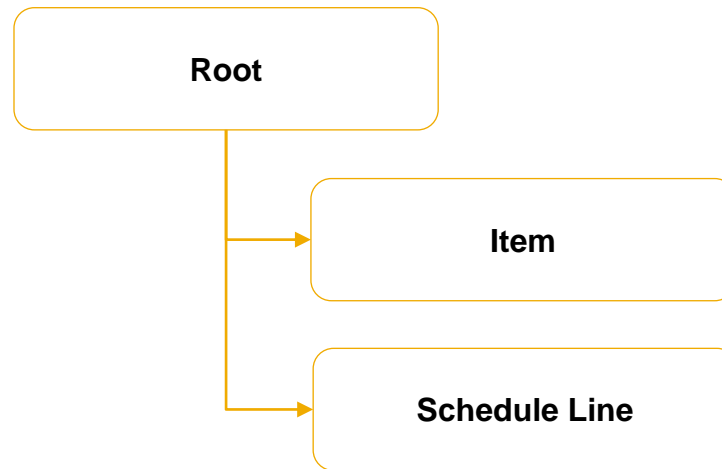
/BOBF/EPM_SALES_ORDER

Sales Order

Description

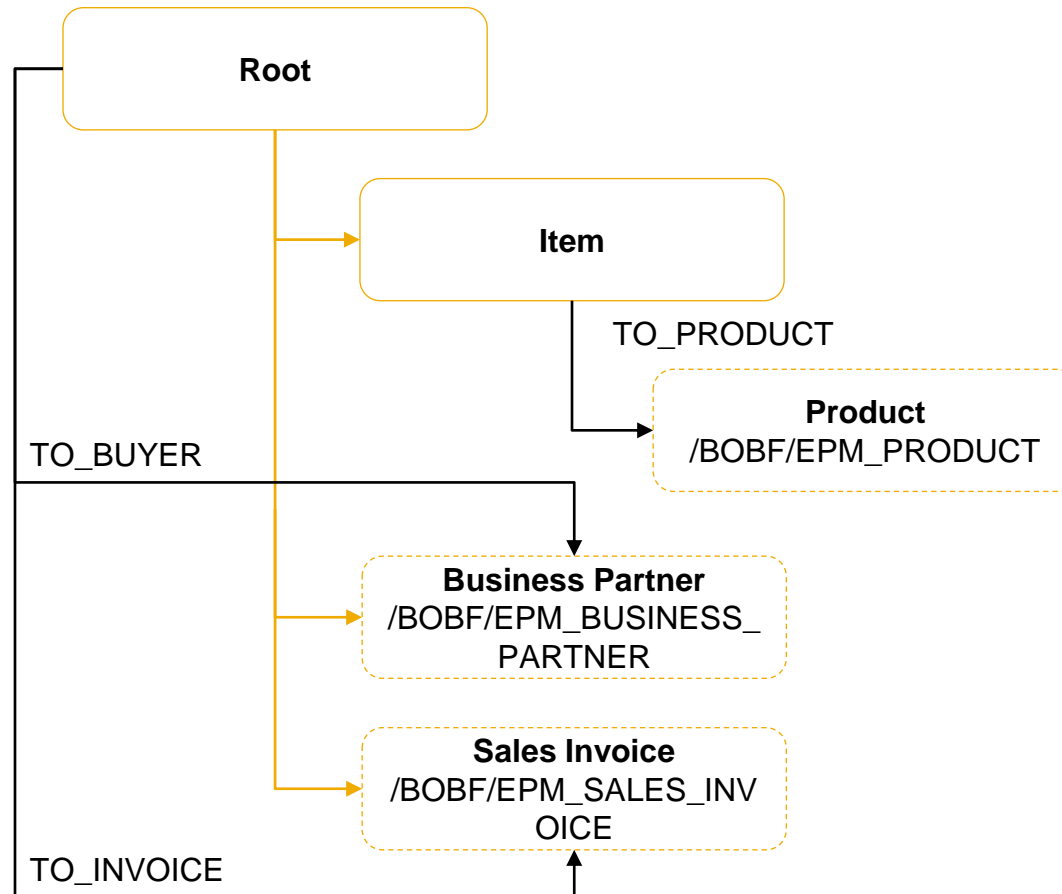
The Sales Order Business Object (/BOBF/EPM_SALES_ORDER) is an internal document created by the seller in order to record the buyers purchase orders. It can be either manually built by the sales team or automatically by the help of an inbound process.

Sales Order Data Model



- **ROOT:** Contains detail information about the status and header information like for instance business partner
- **ITEM:** Contains product information like for instance price
- **SCHEDULE LINE:** Contains information about the expected delivery time and quantity.

Sales Order Association Model



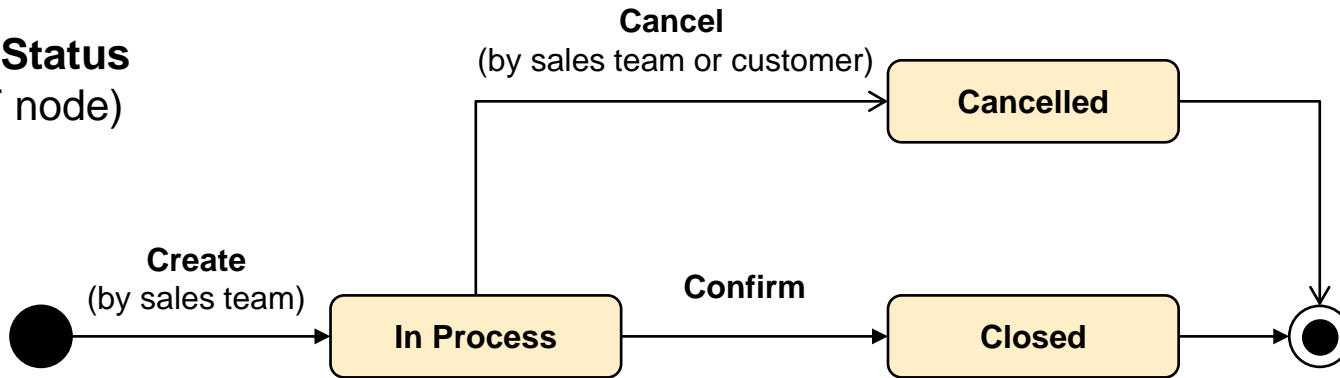
Sales Order

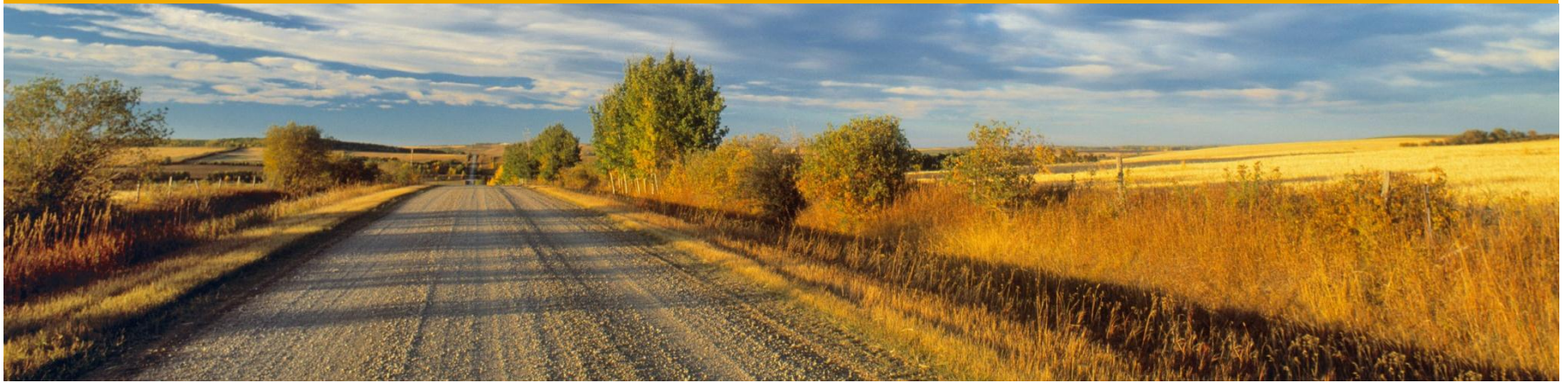
Business Logic

- **CREATE, RETRIEVE, UPDATE, DELETE**
- **SELECT_ALL, SELECT_BY_ELEMENTS**
- The **ADMINISTRATIVE_DATA** determination automatically writes the last change date and user of a product as soon as the transaction is saved
- There is a validation **ALTKEY_UNIQUENESS_CHECK** on the root node ensuring that each business product ID is unique while creating new product instances.
- The property determination **ROOT_PROPERTY** ensures that the business partner can not be updated after saving the instance for the first time. Furthermore, the sales order can only be deleted if the lifecycle status is NEW. A subtree property is used to disable the deletion and update of ITEM instances after a sales order is confirmed.
- The action **INVOICE_CREATED** is called by each sales invoice instance as soon as that invoices are saved for the first time.
- Action **CONFIRM** is used by the seller to indicate the finished process
- Action **CANCEL** is used by the seller to cancel an already existing sales order (e.g. because the order takes too long to be executed)

Sales Order State Model

Lifecycle Status
(on ROOT node)





Purchase Order

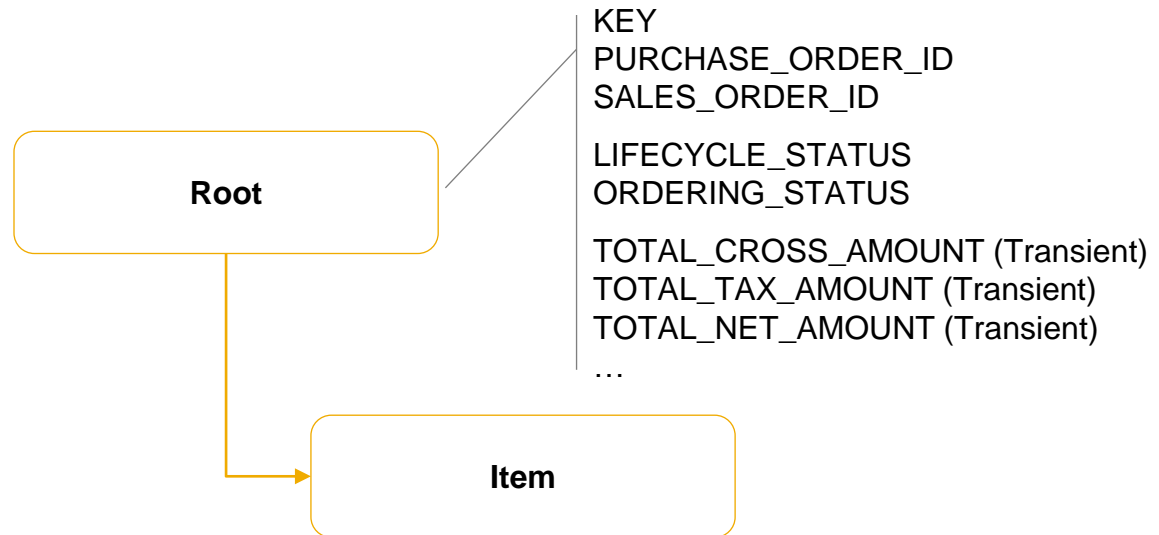
/BOBF/EPM_PURCHASE_ORDER

Purchase Order

Description

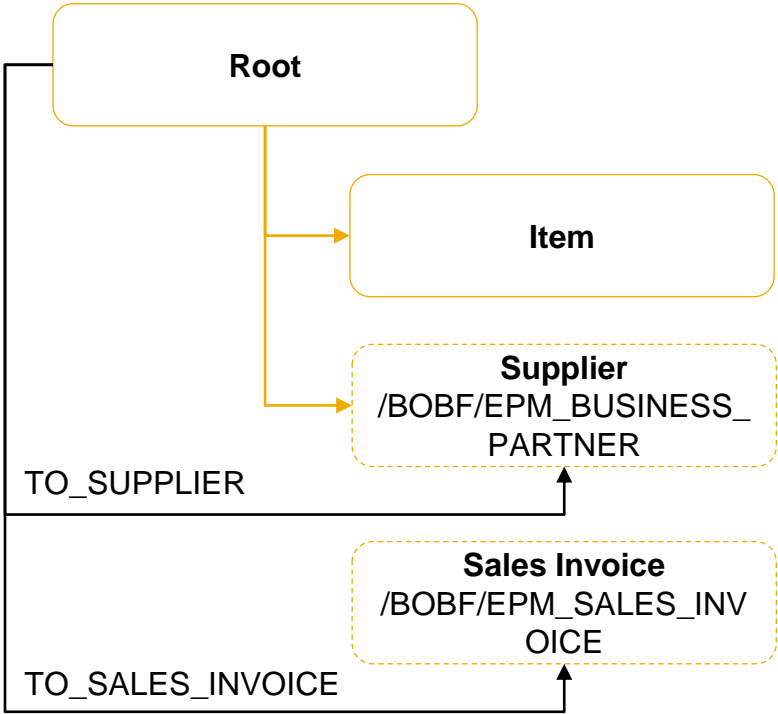
A Purchase Order is a document indicating the products and their quantities and prices that a buyer intends to buy. It is created by the buyer and forwarded to the seller. If the seller accepts the purchase order the buying contract is formed.

Purchase Order Data Model



- **ROOT:** Contains detail information about the status and header information like for instance business partner
- **ITEM:** Contains product information like for instance price

Purchase Order Association Model



Purchase Order

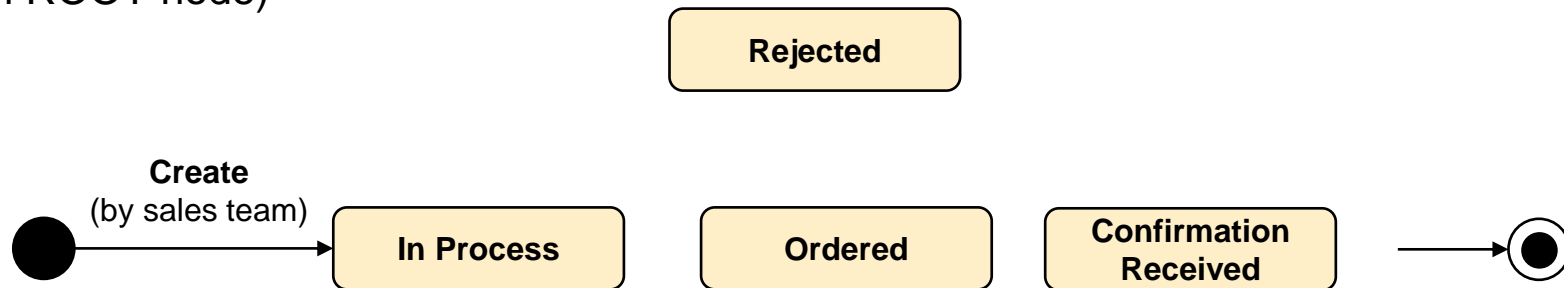
Business Logic

- **CREATE, RETRIEVE, UPDATE, DELETE**
- **SELECT_ALL, SELECT_BY_ELEMENTS**
- The **ADMINISTRATIVE_DATA** determination automatically writes the last change date and user of a purchase order as soon as the transaction is saved
- There is a validation **ALTKEY_UNIQUENESS_CHECK** on the root node ensuring that each purchase order ID is unique while creating new product instances.
- The actions...
 - **APPROVE: Set Status to Approved**
 - **CONFIRM: Set Status to Confirmed**
 - **ORDER: Set Status to Ordered**
 - **REJECT: Set Status to Rejected**

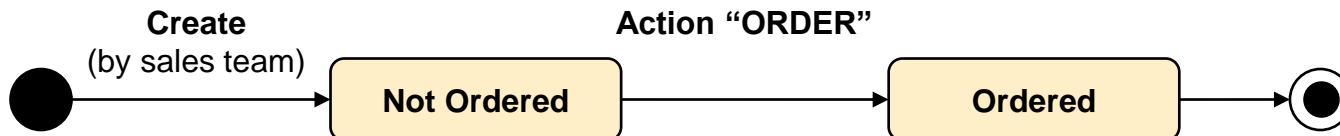
Purchase Order State Model

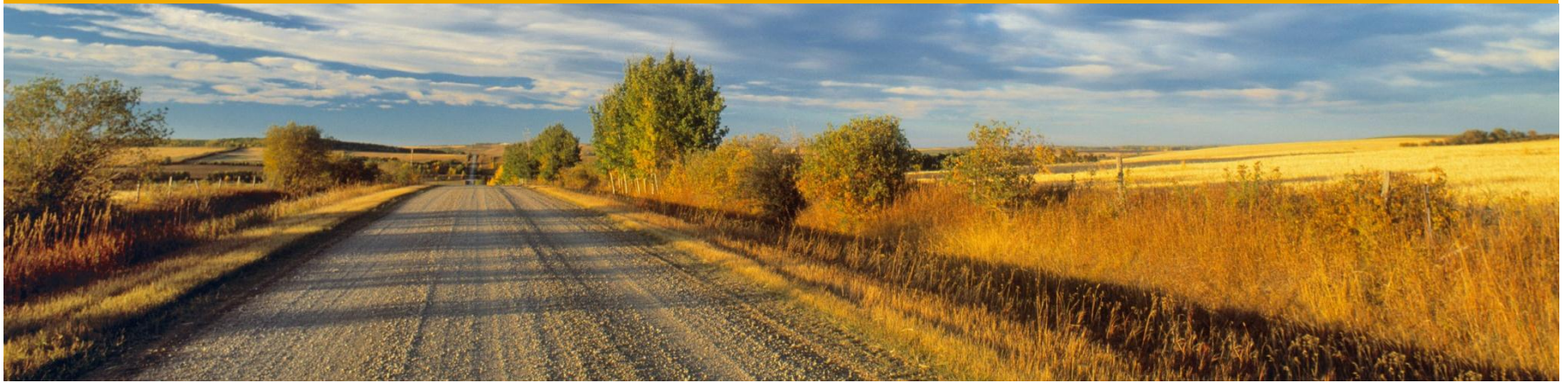
To be checked

Lifecycle Status (on ROOT node)



Ordering Status (on ITEM node and on ROOT node)





Sales Quote

/BOBF/EPM_SALES_QUOTE



Test Data Generator

Test Data Generation

In order to have some sample data to process orders a test data generator can be used. It removes all existing data from the BOPF EPM application database tables and afterwards generates some new random application data.

The generator can be used by executing the report:
`/BOBF/EPM_DATA_GENERATOR`



Thank you

© 2012 SAP AG. 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 AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, PowerPoint, Silverlight, and Visual Studio are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, z10, z/VM, z/OS, OS/390, zEnterprise, PowerVM, Power Architecture, Power Systems, POWER7, POWER6+, POWER6, POWER, PowerHA, pureScale, PowerPC, BladeCenter, System Storage, Storwize, XIV, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, AIX, Intelligent Miner, WebSphere, Tivoli, Informix, and Smarter Planet are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are trademarks or registered trademarks of Adobe Systems Incorporated in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and its affiliates.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems Inc.

HTML, XML, XHTML, and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Apple, App Store, iBooks, iPad, iPhone, iPhoto, iPod, iTunes, Multi-Touch, Objective-C, Retina, Safari, Siri, and Xcode are trademarks or registered trademarks of Apple Inc.

IOS is a registered trademark of Cisco Systems Inc.

RIM, BlackBerry, BBM, BlackBerry Curve, BlackBerry Bold, BlackBerry Pearl, BlackBerry Torch, BlackBerry Storm, BlackBerry Storm2, BlackBerry PlayBook, and BlackBerry App World are trademarks or registered trademarks of Research in Motion Limited.

Google App Engine, Google Apps, Google Checkout, Google Data API, Google Maps, Google Mobile Ads, Google Mobile Updater, Google Mobile, Google Store, Google Sync, Google Updater, Google Voice, Google Mail, Gmail, YouTube, Dalvik and Android are trademarks or registered trademarks of Google Inc.

INTERMEC is a registered trademark of Intermec Technologies Corporation.

Wi-Fi is a registered trademark of Wi-Fi Alliance.

Bluetooth is a registered trademark of Bluetooth SIG Inc.

Motorola is a registered trademark of Motorola Trademark Holdings LLC.

Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.