Week 4 Unit 1: **Native SAP HANA Usage in ABAP**

Native SAP HANA Usage in ABAP
Outline

Content

- Native SQL via ABAP Database Connectivity (ADBC)

- Consumption of SAP HANA Procedures

- Sneak Preview: ABAP Managed Database Procedure Consumption

Public

Native SAP HANA Usage in ABAP
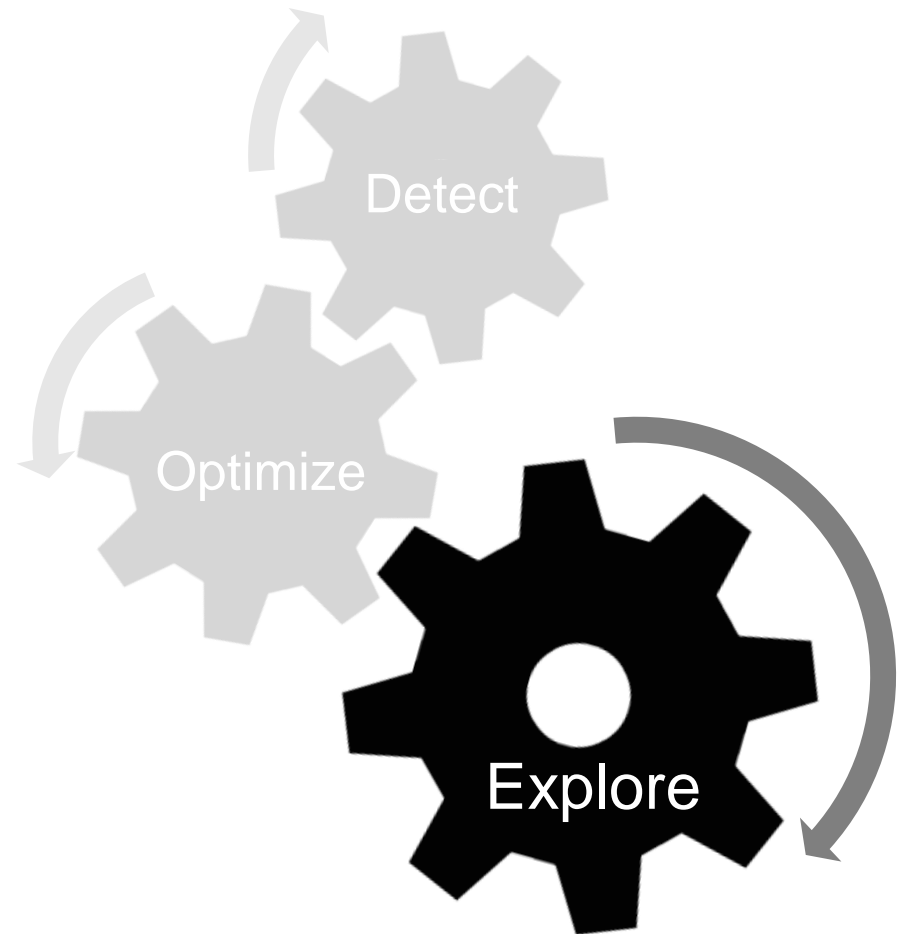## Migration to SAP HANA

1. Detect

- Functional correctness
- Performance optimization potential

2. Optimize

- Database-oriented programming

3. Explore

- Use SAP HANA-specific features
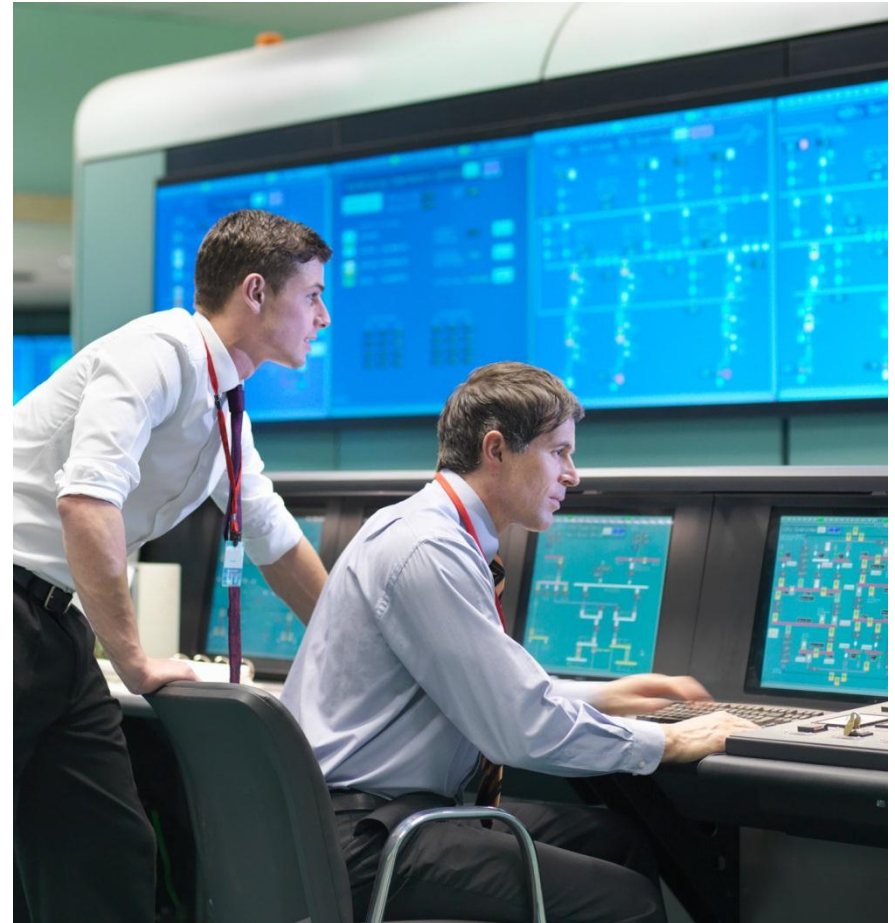- Rethink & innovate

Detect

Optimize

Explore

Native SAP HANA Usage in ABAP
ABAP Database Connectivity (ADBC)

Object-based ABAP API for programming relational database access

- Access to entire SQL functionality
- Dynamic creation and execution of native SQL statements
- Clean concept for multiple DB connections
- Exception handling

Native SAP HANA Usage in ABAP
ABAP Database Connectivity: Demo

ABAP Database Connectivity (ADBC)

- Typical code / call sequence
- Call of an SAP HANA database procedure

More information

- See demo reports ADBC_DEMO and ADBC_DEMO_PROC_CALLS_HDB

Native SAP HANA Usage in ABAP
ABAP Database Connectivity: Typical Code

Typical call sequence

- Target type definition / data declarations

- Concatenation / Definition of the DBSYS-dependent SQL Query

- Create statement object (and DB connection if necessary)

- Execute query, passing SQL query as string

- Assign internal table for query result

- Retrieve result

- Close result set / free allocated resources

```abap
REPORT zr_adbc_simple.

TYPES:
  BEGIN OF ty_res,
    bp_id              TYPE snwd_bpa-bp_id,
    company_name       TYPE snwd_bpa-company_name,
    currency_code      TYPE snwd_so-currency_code,
    total_gross_amount TYPE snwd_so-gross_amount,
  END OF ty_res.

DATA lv_stmt   TYPE string.
DATA lo_stmt   TYPE REF TO cl_sql_statement.
DATA lo_res    TYPE REF TO cl_sql_result_set.
DATA lt_result TYPE STANDARD TABLE OF ty_res WITH EMPTY KEY.

lv_stmt = |SELECT BP_ID, COMPANY_NAME, SO.CURRENCY_CODE,       | &&
          |        SUM( SO.GROSS_AMOUNT ) as TOTAL_GROSS_AMOUNT | &&
          | FROM SNWD_BPA AS BPA                                | &&
          | INNER JOIN SNWD_SO AS SO                            | &&
          |   ON SO.BUYER_GUID = BPA.NODE_KEY                   | &&
          | GROUP BY BP_ID, COMPANY_NAME, SO.CURRENCY_CODE      |.

TRY.
    lo_stmt = NEW cl_sql_statement( ).
    lo_res  = lo_stmt->execute_query( lv_stmt ).
    lo_res->set_param_table( ref #( lt_result ) ).
    lo_res->next_package( ).
    lo_res->close( ).

    CATCH cx_sql_exception INTO DATA(lx).
    "do some meaningful error handling
    WRITE: lx->sql_message.
ENDTRY.
```

Native SAP HANA Usage in ABAP
Native SQL Pitfalls / Disadvantages

No syntax check at compile time

- SQL query is passed as string

Developer responsible for

- Client handling, accessing correct database schema
- Type mapping, for example, of internal table for result retrieval
- Releasing DB resources
- Proper locking and commit handling

ABAP table buffers bypassed

Coding is database-dependent

- Remember the ABAP Test Cockpit check on Native SQL usage

Native SAP HANA Usage in ABAP
Consumption of an SAP HANA Database Procedure with ADBC

> **i** Additional Material

Consumption

- Consumption of SAP HANA procedures via „CALL <PROCEDURE>"

- Cumbersome, especially for procedures with (several) input / output parameters

More Information

- Demo report ADBC_DEMO_PROC_ CALLS_HDB

```abap
REPORT zr_adbc_proc_call.
"lots of type definitions & data declarations
lv_stmt =
  | CALL "SAPHANAABAP"."ZCL_DEMO_PAID_ON_DATE_AMDP=>PAID_ON_DATE" | &&
  |                  ( '20140401', NULL, NULL, NULL ) WITH OVERVIEW |.

TRY.
    lo_stmt = NEW cl_sql_statement( ).
    lo_res  = lo_stmt->execute_query( lv_stmt ).
    "get a reference of the overview table and prepare the result set
    lo_res->set_param_table( REF #( lt_overview ) ).
    "retrieve the overview table
    lo_res->next_package( ).
    LOOP AT lt_overview INTO DATA(ls_overview).
      "select from the corresponding DB table listed in the overview table
      DATA(lo_res_tab) =
        lo_stmt->execute_query( | select * from { ls_overview-table }| ).
      IF ls_overview-value CS 'ET_INVOICE_ITEM'.
        "prepare the result set
        lo_res_tab->set_param_table( REF #( lt_inv_item ) ).
      ELSEIF ls_overview-value CS 'ET_INVOICE_HEAD'.
        "prepare the result set
        lo_res_tab->set_param_table( REF #( lt_inv_head ) ).
      ELSEIF ls_overview-value CS 'ET_CUSTOMER_INFO'.
        "prepare the result set
        lo_res_tab->set_param_table( REF #( lt_cust_info ) ).
      ENDIF.
      lo_res_tab->next_package( ).
    ENDLOOP.
    lo_res->close( ).
  CATCH cx_sql_exception INTO DATA(lx).
    "do some meaningful error handling
ENDTRY.
```

Native SAP HANA Usage in ABAP
Consumption of an ABAP Managed Database Procedure

Consumption

- Consumption of an ABAP Managed Database Procedure

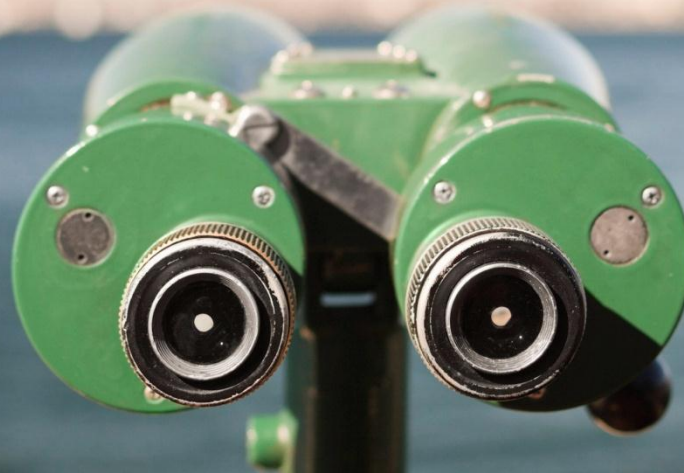## More Information

- Upcoming unit – stay tuned ☺

```abap
REPORT zr_amdp_call.

DATA(lo_info_list) = NEW zcl_demo_paid_on_date_amdp( ).

lo_info_list->paid_on_date(
  EXPORTING
    iv_payment_date = '20140401'
  IMPORTING
    et_customer_info  = DATA(lt_cust_info_amdp)
    et_invoice_header = DATA(lt_inv_head_amdp)
    et_invoice_item   = DATA(lt_inv_item_amdp) ).
```

Native SAP HANA Usage in ABAP
What's Next?

Week 4 Unit 2
ABAP Managed Database Procedures

Thank you

Contact information:

open@sap.com

openSAP

© 2014 SAP AG or an SAP affiliate company. All rights reserved.

Week 4 Unit 2: **ABAP Managed Database Procedures**

ABAP Managed Database Procedures
Outline

Content

- ABAP Managed Database Procedures
  - Concept
  - Definition
  - Consumption
- Exception Handling
- Additional Information:
  - Debugging AMDPs
  - AMDP Extension Concept

ABAP Managed Database Procedures
Migration to SAP HANA

1. **Detect**
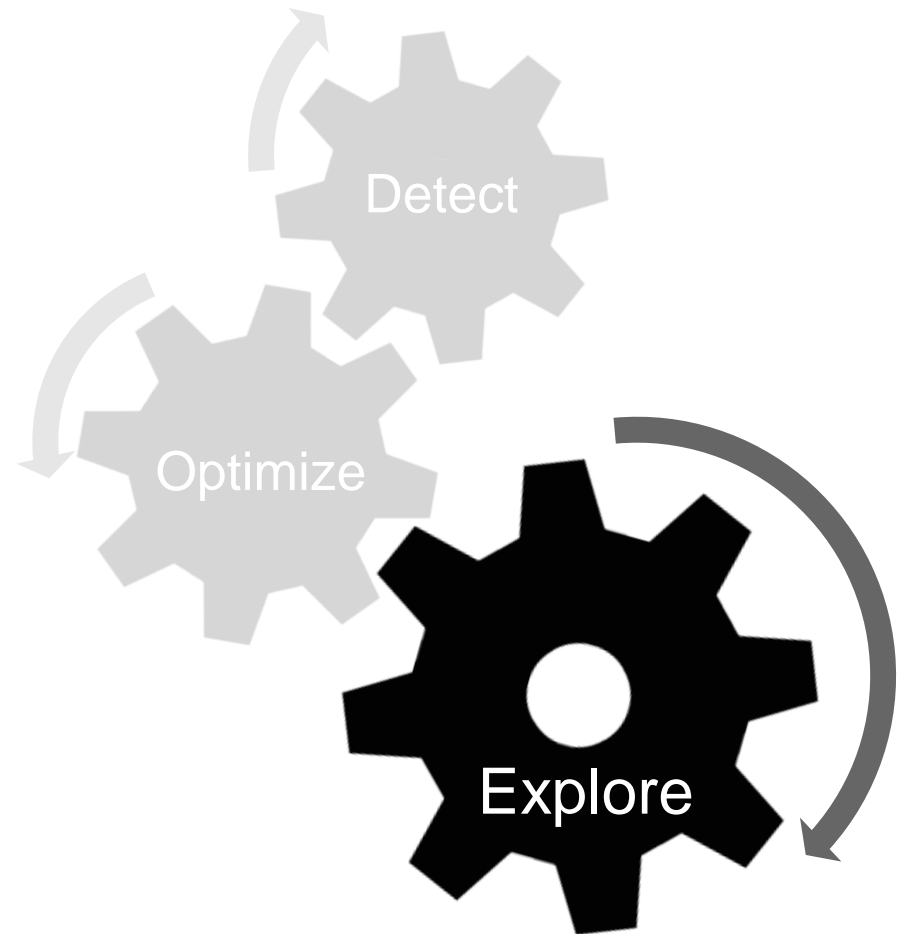   - Functional correctness
   - Performance optimization potential

2. **Optimize**
   - Database-oriented programming

3. **Explore**
   - Use SAP HANA-specific features
   - Rethink & innovate

Detect

Optimize

Explore

ABAP Managed Database Procedures
Introduction of AMDPs

Code-to-Data Paradigm

- Supported through embedding native database procedure coding

Definition & Consumption of AMDPs

- Definition / maintenance via ABAP Development Tools in Eclipse
- Standard ABAP class method as containers for database procedures coding
  → Corresponding SAP HANA artifacts created automatically
- Consumption like any other ABAP class method

Fully integrated into the ABAP infrastructure

- Consistent lifecycle management with all other ABAP artifacts (only transport of ABAP objects required)
- Syntax check provided for SQLScript code
- Detailed analysis of ABAP runtime errors
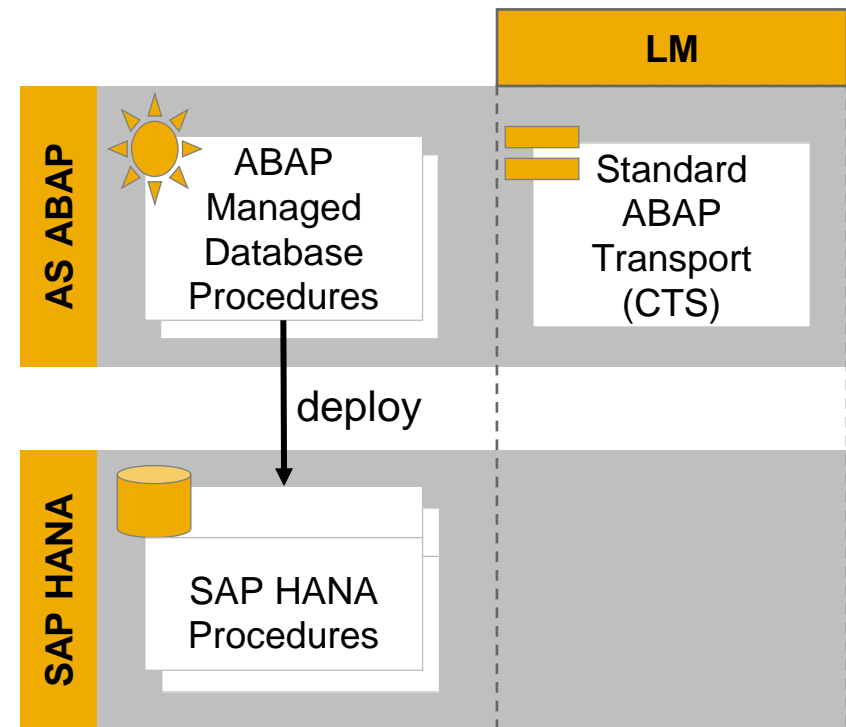
	LM
AS ABAP	ABAP Managed Database Procedures	Standard ABAP Transport (CTS)

deploy

| **SAP HANA** | SAP HANA Procedures | |

ABAP Managed Database Procedures
ABAP Managed Database Procedures: Demo

ABAP Managed Database Procedures

- Definition
- Implementation
- Consumption
- Runtime Error Analysis

More information

- On AMDPs:
  http://scn.sap.com/docs/DOC-51146
- On SQLScript:
  SAP HANA SQLScript Reference

ABAP Managed Database Procedures
Class Definition

Prerequisites in class definition

- Classes with AMDPs must use interface IF_AMDP_MARKER_HDB

- All AMDP method parameters must be passed by value (like RFC)

- All AMDP parameters must be tables with elementary components or scalar types

- AMDPs support (secondary) database connections to the primary database via input parameter **CONNECTION** (type DBCON_NAME)

```abap
CLASS zcl_amdp_simple_00 DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES: if_amdp_marker_hdb.


    METHODS get_customer_infos
      IMPORTING
        VALUE(<input>) TYPE <input_type>
      EXPORTING
        VALUE(<output>) TYPE <output_type>
      RAISING <amdp_exception> .
ENDCLASS.
```

ABAP Managed Database Procedures
Class Implementation

Extended method implementation syntax: BY DATABASE PROCEDURE …

- Indicates method body contains database-specific code not executable on the ABAP server

- Database platform (currently only SAP HANA supported)

- Database procedure language (for example SQLScript)

- Used ABAP Dictionary tables, Dictionary views, and other AMDP methods

- Native SAP HANA SQLScript source code

```
METHOD get_customer_infos
BY DATABASE PROCEDURE
FOR HDB
LANGUAGE SQLSCRIPT
OPTIONS READ-ONLY
USING <dictionary_artifacts>.

--meaningful SQLScript coding

et_customer_info =
 SELECT ... FROM ...;

ENDMETHOD.
```

ABAP Managed Database Procedures
Consumption & Artifact Creation

AMDP consumption like any other ABAP method call

AMDP Runtime:

- At first call of an AMDP, several SAP HANA artifacts are created in the SAP<SID> schema, such as the SAP HANA database procedure

- Artifact creation can alternatively been triggered via ABAP report **RSDBGEN_AMDP**

- When an AMDP is processed, the ABAP stack calls the corresponding database procedure in SAP HANA

```abap
REPORT zr_amdp_01_simple_call.

DATA(lo_amdp) = NEW zcl_amdp_simple_00( ).

lo_amdp->get_customer_infos(
     IMPORTING
       et_customer_info = DATA(lt_result)
     ).
```

ABAP Managed Database Procedures
Exception Handling

Catchable Exceptions

- Several AMDP runtime errors have a corresponding (catchable) exception

- Naming convention:

  <ERROR_NAME> →
  CX_<ERROR_NAME>

- To-Dos for AMDP
  Developers/Consumers:
  - Add RAISING clause to the AMDP method definition
  - Enclose the AMDP call in a TRY… CATCH block


```
"definition
METHODS <method_name>
        <method_interface>
      RAISING cx_amdp_error.
...

"consumption
TRY.
    <method_call>

  CATCH cx_amdp_execution_failed INTO DATA(lx).
    "do some meaningful error handling
ENDTRY.
```

ABAP Managed Database Procedures
ABAP Managed Database Procedures: Demo

ABAP Managed Database Procedures

- Catchable Exceptions

ABAP Managed Database Procedures
AMDP Exception Hierarchy

CX_AMDP_ERROR

  CX_AMDP_VERSION_ERROR

    CX_AMDP_VERSION_MISMATCH
    Version conflict; database procedure has been changed
    during program execution

  CX_AMDP_CREATION_ERROR

    CX_AMDP_DBPROC_CREATE_FAILED
    Database procedure could not be created because a called
    database procedure does not exist on
    the database (any more).

    CX_AMDP_NATIVE_DBCALL_FAILED
    SQL error at creation or update of a database procedure
    before it is called.

    CX_AMDP_WRONG_DBSYS
    Database procedure not defined for the current database
    system.

CX_AMDP_ERROR

  CX_AMDP_EXECUTION_ERROR

    CX_AMDP_EXECUTION_FAILED
    Database error during execution of a database
    procedure.

    CX_AMDP_IMPORT_TABLE_ERROR
    Table parameter error during execution of a database
    procedure.

    CX_AMDP_RESULT_TABLE_ERROR
    Error at passing of result table.

CX_AMDP_ERROR

  CX_AMDP_CONNECTION_ERROR

    CX_AMDP_NO_CONNECTION
    No database service connection available.

    CX_AMDP_NO_CONNECTION_FOR_CALL
    No database connection available for calling a
    database procedure.

    CX_AMDP_WRONG_CONNECTION
    Reserved database connection must not be used for
    calling a database procedure.

ABAP Managed Database Procedures
Additional Information

Extension Concept

- Extensibility of ABAP Managed Database Procedures supported via AMDP BAdIs
  → More information on the following slides and on SCN in the (ABAP for SAP HANA Community)

Debugging of AMDPs

- Fully integrated support in the ABAP Development Tools in Eclipse not yet available

- Workaround: External session debugging. For more information see:

  – SCN document:
    http://scn.sap.com/docs/DOC-51790

  – Video tutorial:
    http://www.youtube.com/watch?v=5OEuyI_s5ME

ABAP Managed Database Procedures
Extensibility: AMDP Business Add-Ins (BAdIs)

Modification-free enhancement of AMDP methods using BAdIs and calling points

Special BAdIs provided (AMDP BAdIs)

- AMDP BAdI interface has to include the tag interfaces **IF_AMDP_MARKER_HDB**

- Only AMDP methods allowed

- Fallback method is mandatory

Integrated in the Switch & Enhancement Framework

Filters are currently not supported

Enhancement Spot ZA4H1_AMDP_BADI Change

Enhancement Spot   ZA4H1_AMDP_BADI   Inactive

| Attributes | Enhancem. Implementations | Technical Details | Enh. Spot Element Definitions |

BAdI Definition   ZA4H1_AMDP_BADI   tech.

BAdI Definitions	Description
ZA4H1_AMDP_BADI	BAdI Definition
· Interface	
· Implementations	

Description   BAdI Definition
Interface   ZIF_A4H1_AMDP_BADI

Usability
- ☐ Multiple Use
- ☐ Can only be implemented SAP-internally
- ☐ Limited Filter Use
- ☑ AMDP BAdI

Instance Creation Mode
- ⦿ Newly Creating Instantiation
- ○ Reusing Instantiation
- ○ Context-Specific Instantiation

- ☑ Call fallback if no implementation is executed
- Fallback Class   ZCL_A4H1_AMDP_BADI

Implementation Example Classes

Example Classes	Description
ZCL_A4H1_AMDP_BADI	Class for BAdI: ZA4H1_AMDP_BADI

ABAP Managed Database Procedures
Extensibility: Consumption of an AMDP Business Add-In

Consumption

- Call of AMDP BAdI like any other AMDP call
  - Inclusion in USING list
  - Consumption via CALL <AMDP_NAME>

```
CLASS zcl_a4h1_call_amdp_badi IMPLEMENTATION.          BAdI Caller
  METHOD call_badi
    by DATABASE PROCEDURE
    FOR HDB
    LANGUAGE SQLSCRIPT
    options read-only
    using ZA4H1_AMDP_BADI=>get_customer_infos.

    call "ZA4H1_AMDP_BADI=>GET_CUSTOMER_INFOS"( et_customer_info => et_customer_info );
  ENDMETHOD.
```

Enhancement Spot ZA4H1_AMDP_BADI Display

Enhancement Spot	ZA4H1_AMDP_BADI	Active

| Attributes | Enhancem. Implementations | Technical Details | Enh. Spot Element Definitions |

BAdI Definitions	Description
ZA4H1_AMDP_BAD	BAdI Definition
· Interface	
· Implementations	

Implementations

BAdI Definition	ZA4H1_AMDP_BADI

2 implementations found

Active	Enhancement Implementation	Badi Implementation
🔆	ZCL_A4H1_BADI_IMPL_01	ZCL_A4H1_BADI_IMPL_01
⚬	ZCL_A4H1_BADI_IMPL_02	ZCL_A4H1_BADI_IMPL_02

BAdI Implementation

ABAP Managed Database Procedures
What's Next?

Week 4 Unit 3
Domain-Specific Usage of Core Data Services

Thank you

Contact information:

open@sap.com

openSAP

© 2014 SAP AG or an SAP affiliate company. All rights reserved.

Week 4 Unit 3: **Domain-Specific Usage of Core Data Services**

SAP

openSAP

Domain-Specific Usage of Core Data Services
Outline

Content

- CDS Annotation Concept
- CDS Annotations
  - SQL View Name
  - Client Handling
  - Buffering
  - Currency & Quantity Columns
- CDS meets SAP Gateway

Domain-Specific Usage of Core Data Services
CDS Annotation Concept

Enrich the CDS data models with additional metadata

- Annotations begin with @

Different ABAP-specific information can be maintained

- Name of the associated SQL view
- Buffering
- Client dependency
- Reference information for Currency and Quantity columns

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_40'
@ClientDependent: true
@AbapCatalog.Buffering.status:
#SWITCHED_OFF
define view zcdsv_annotation_simple as
select from snwd_so
{
  key so_id as customer_id,
  @Semantics.currencyCode
  currency_code,
  @Semantics.amount.currencyCode:
'currency_code'
  gross_amount
}
```

Domain-Specific Usage of Core Data Services
CDS Annotations – Demo

___

Annotations in CDS

- Name of the associated SQL view

- Buffering

- Client dependency

- Reference information for Currency and Quantity columns

Domain-Specific Usage of Core Data Services
Consumption of CDS Views in SAP Gateway

CDS View Consumption

- Direct mapping of CDS views to an entity set in an OData service

- Completely modeled approach; no coding needed for read operations

- OData service can be easily bound to an SAPUI5 control

SAPUI5

Gateway OData

ABAP

Application Programming

Core Data Services

SAP HANA
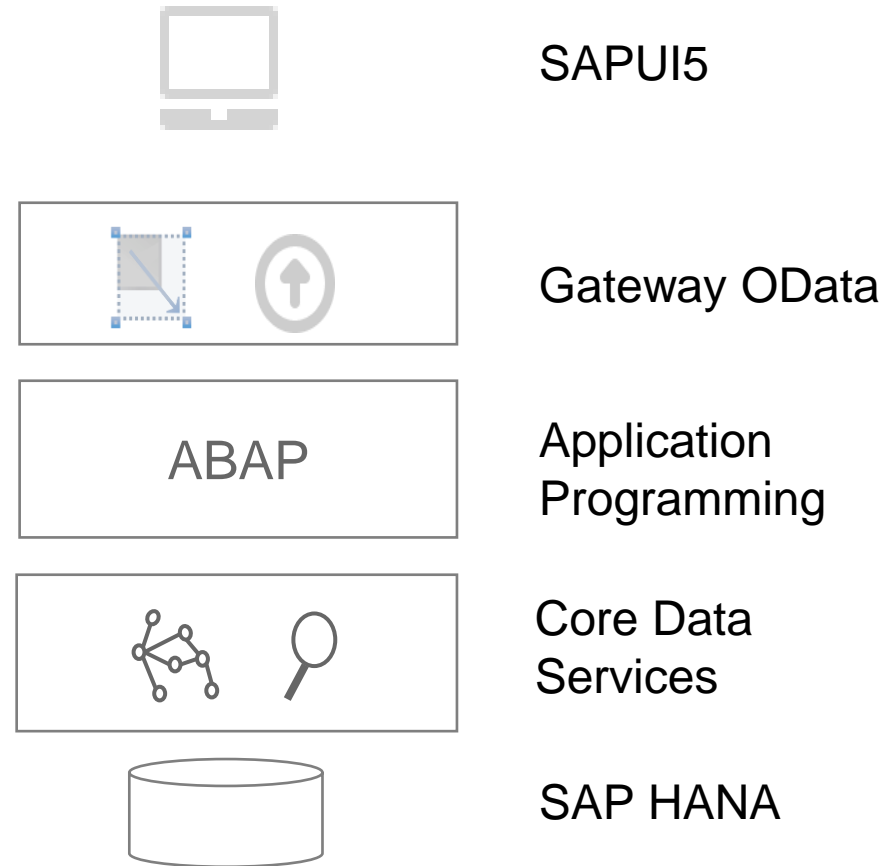
Domain-Specific Usage of Core Data Services
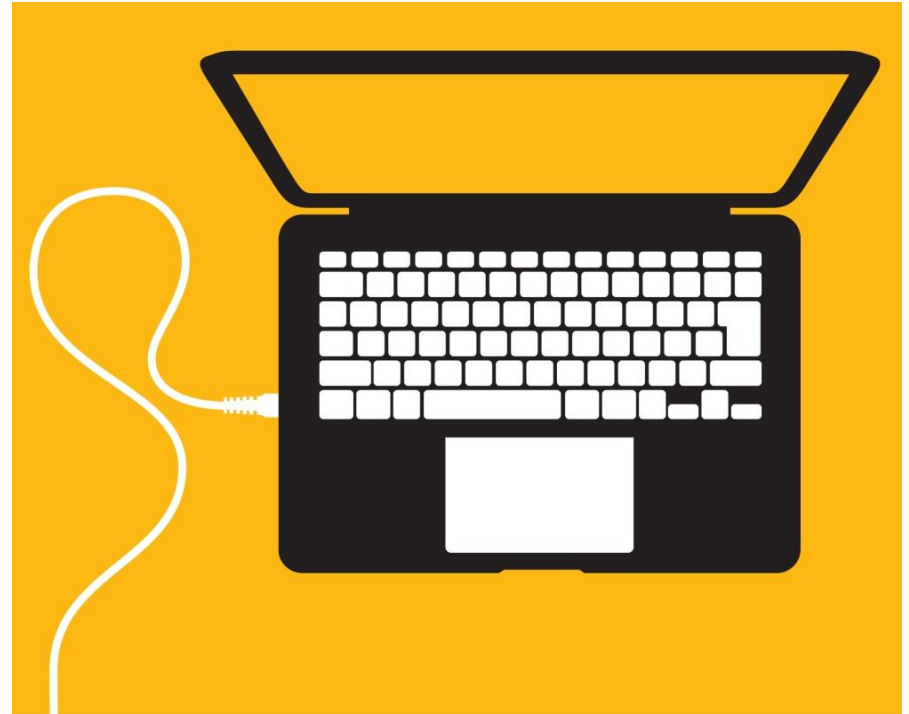CDS View Consumption in SAP Gateway – Demo

CDS in SAP Gateway

- Result of OData request on your trial system:
  http://abapci:50000/sap/opu/odata/sap/ZA4H1_E2E_CUST_INFO_SRV/CustomerInfos?sap-ds-debug=true

Domain-Specific Usage of Core Data Services

Consumption of CDS Views and AMDPs in an SAP Fiori Application

End-to-end application demo

- On your trial system:
http://abapci:50000/sap/bc/ui5_ui5/sap/za4h1_e2e_fiori/index.html?sap-client=001

Guide to build the application on your own:

- http://scn.sap.com/docs/DOC-54541

Domain-Specific Usage of Core Data Services
What's Next?

Week 4 Unit 4
SAP BW and ABAP for SAP HANA: A Perfect Match

Thank you

Contact information:

open@sap.com

openSAP

© 2014 SAP AG or an SAP affiliate company. All rights reserved.

Week 4 Unit 4: **SAP BW and ABAP for SAP HANA: A Perfect Match**

SAP BW and ABAP for SAP HANA: A Perfect Match
Outline

Content

Demo and outlook on the upcoming openSAP course: "SAP Business Warehouse powered by SAP HANA"

Relax and enjoy!

SAP BW and ABAP for SAP HANA: A Perfect Match
AMDP Usage Within SAP Business Warehouse (BW)

AMDP in SAP BW Data Flows

- Native SAP HANA coding in data loading processes

- Integrated in the SAP BW ETL processes (routine type "procedure", SAP HANA analysis process)

- Available as of SP8 for SAP BW 7.40 powered by SAP HANA

Advantages

- Optimized data loads for SAP HANA by leveraging full flexibility of SQLScript procedures

- Fully integrated in SAP BW Lifecycle and Process Management

SAP Business Warehouse

SAP BW Process Management

Info-Provider

AMDP

Info-Provider

SAP HANA

SAP BW and ABAP for SAP HANA: A Perfect Match
AMDP Usage Within SAP BW: Demo

ABAP Managed Database Procedures in SAP BW powered by SAP HANA

SAP HANA expert script processing

More Information

Upcoming OpenSAP course:
SAP Business Warehouse powered by
SAP HANA → Stay tuned!

SAP BW and ABAP for SAP HANA: A Perfect Match
What's Next?
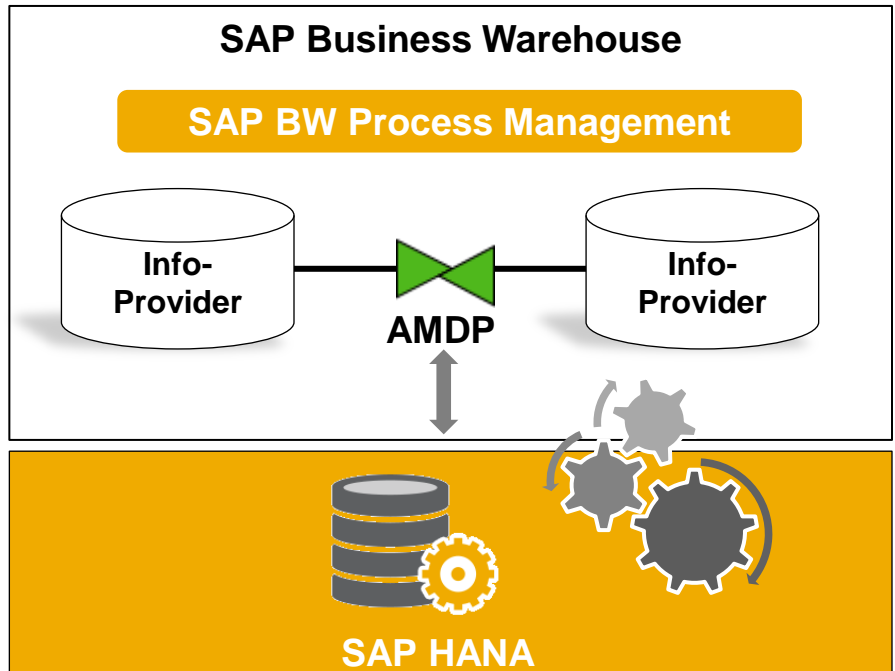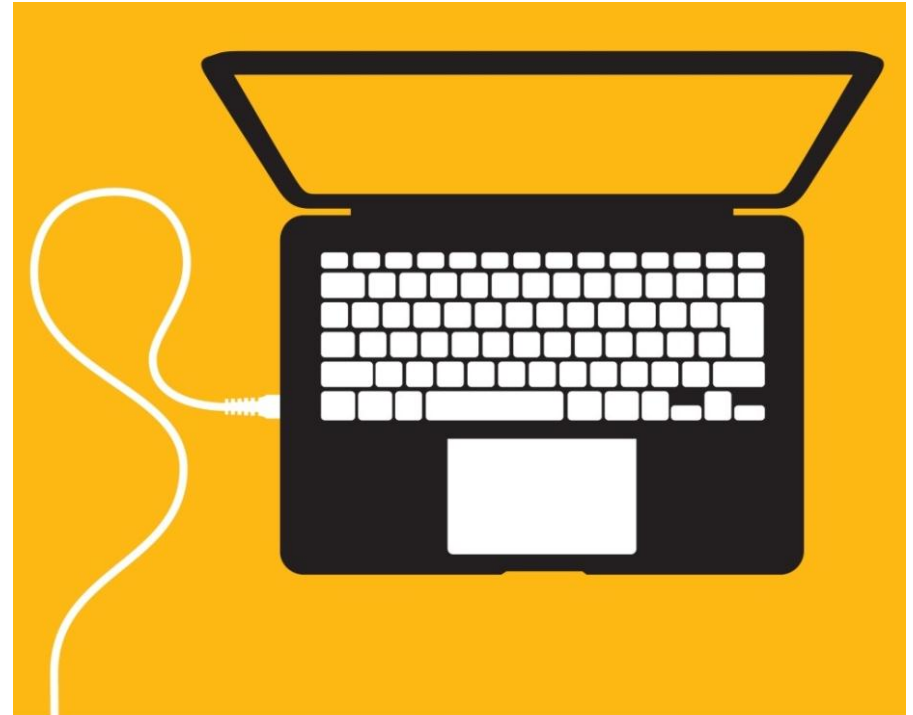
Week 4 Unit 5
Looking Ahead

Thank you

Contact information:

open@sap.com

openSAP

© 2014 SAP AG 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 AG or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG (or an SAP affiliate company) in Germany and other countries. Please see http://global12.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.

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

National product specifications may vary.

These materials are provided by SAP AG or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP AG or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP AG 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 AG 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 AG's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP AG 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, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Week 4 Unit 5:
Looking Ahead

SAP

openSAP

Looking Ahead
Outline

Content

- Recap
- Lab Preview
- More Information

Looking Ahead
Recap: Code-to-Data Paradigm

In-memory computing imperative:

- Avoid (unnecessary) movement of large data volumes
- Perform data-intensive calculations in the database

Today:

Data-intensive computations in application layer

calculate

Application Layer

Data Layer

SAP HANA:

Delegate data-intensive operations to data layer

calculate

Looking Ahead
Recap: Migration to SAP HANA & Code-to-Data Capabilities
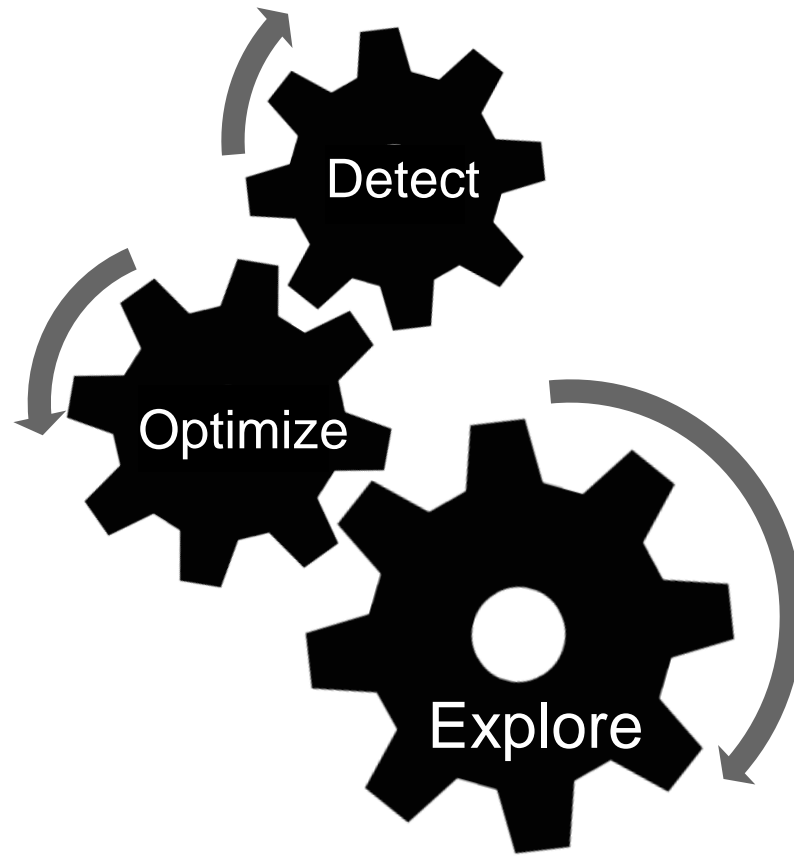
Looking Ahead
Recap: Performance Improvements vs. Code Adjustments

Code Complexity Adjustments (vertical axis, from (less) to (more))

Performance Improvement (horizontal axis, from (low) to (high))

Transparent Optimizations
- Fast Data Access
- Table Buffer

Advanced SQL in ABAP
- Open SQL Enhancements
- Advanced View Definition

SAP HANA Native Features
- ABAP Managed Database Procedures
- Native SQL
- …
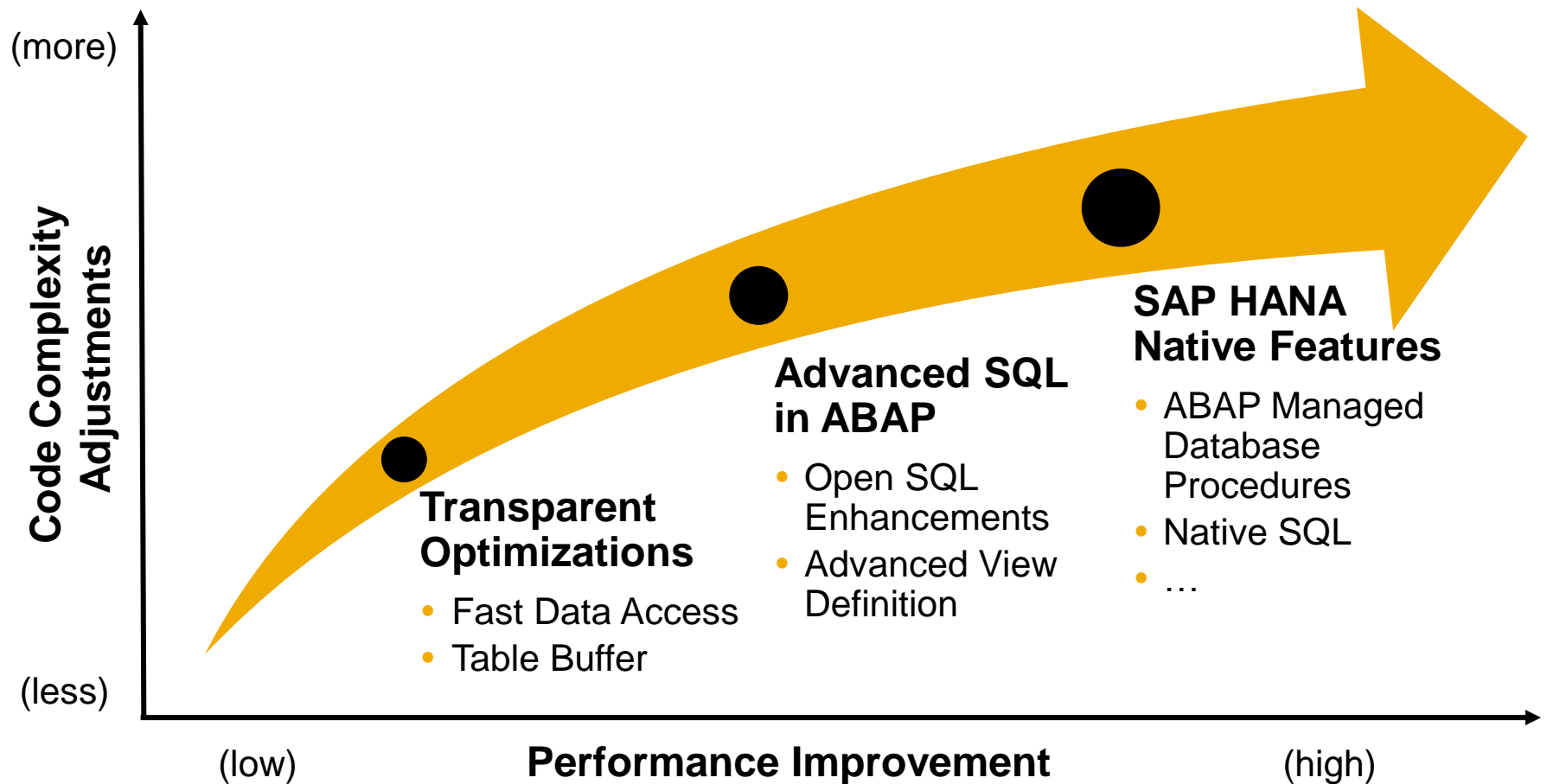
Looking Ahead
Recap: Top-Down Approach (≥ NW AS ABAP 7.40 SP05)
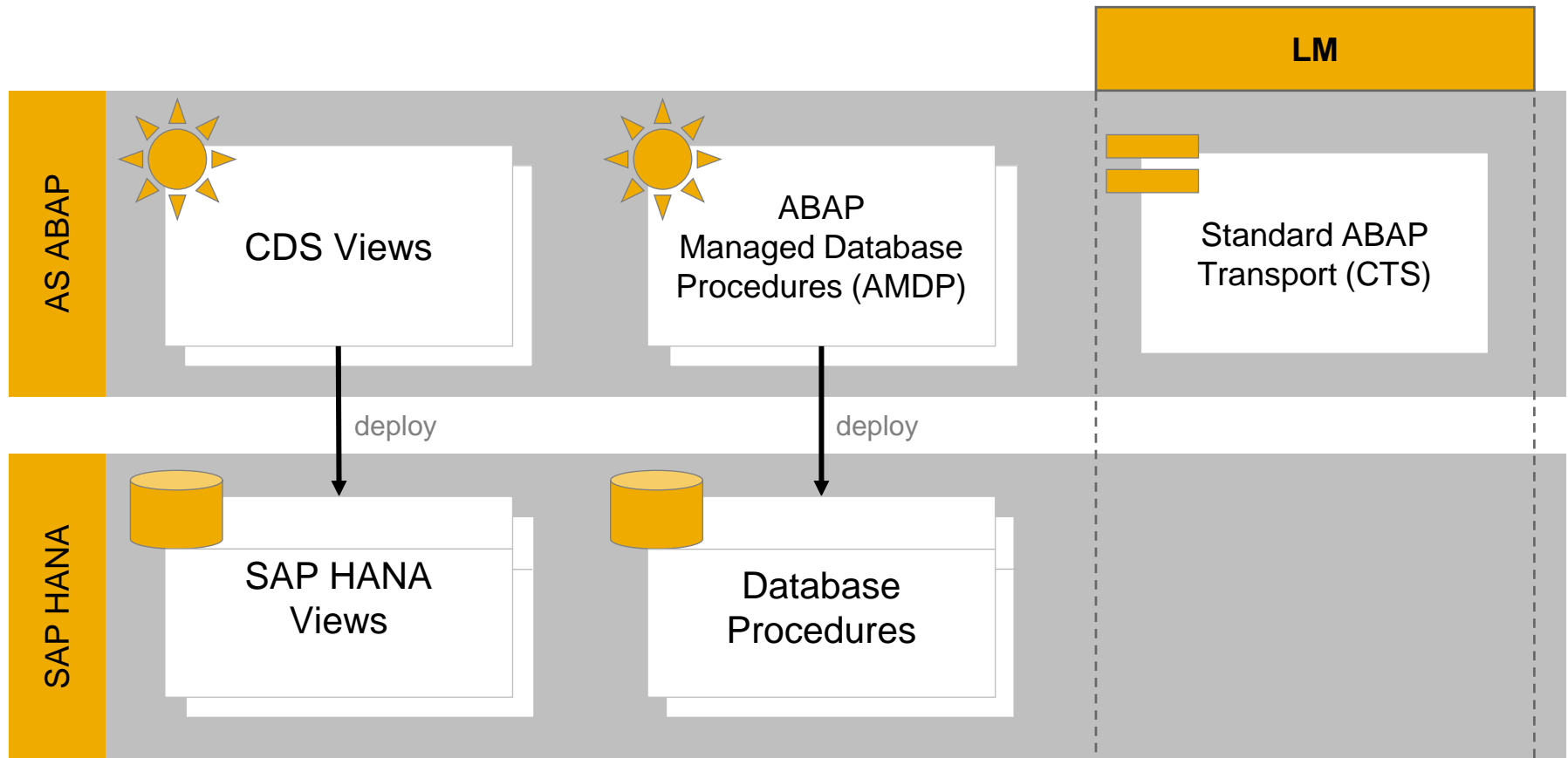
Looking Ahead
Lab Preview – Where Are We Heading?

1

Optimized Data Transfer

- Fast data access
- `SELECT ... FOR ALL ENTRIES`
- Data exchange for stored procedures

2

Code Pushdown

- Type harmonization
- CDS views
- Database procedures and functions
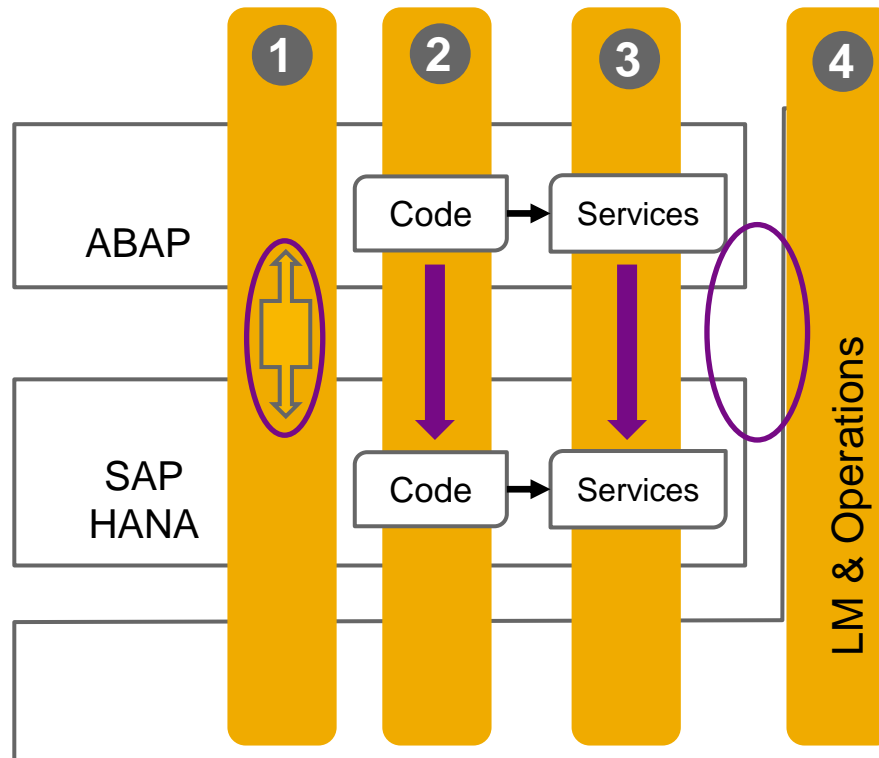
3

Service Pushdown

- Number ranges, GUIDs
- Enqueue service
- Calendars (factory, fiscal)

4

Improved Operations

- Co-deployment
- Aligned HA setup
- Zero downtime mgmt
- Workload mgmt

Looking Ahead
More Information

Learn more about ABAP Development for SAP HANA

- SAP Training and Certification Shop
- SAP HANA Academy
- SAP Community Network (SCN):
  - Reference Scenario:
    http://scn.sap.com/docs/DOC-35518
    (including guides, how-tos, video tutorials, …)
  - Video Tutorials:
    http://scn.sap.com/docs/DOC-47994
  - ABAP YouTube Channel

Join the
ABAP for SAP HANA Community
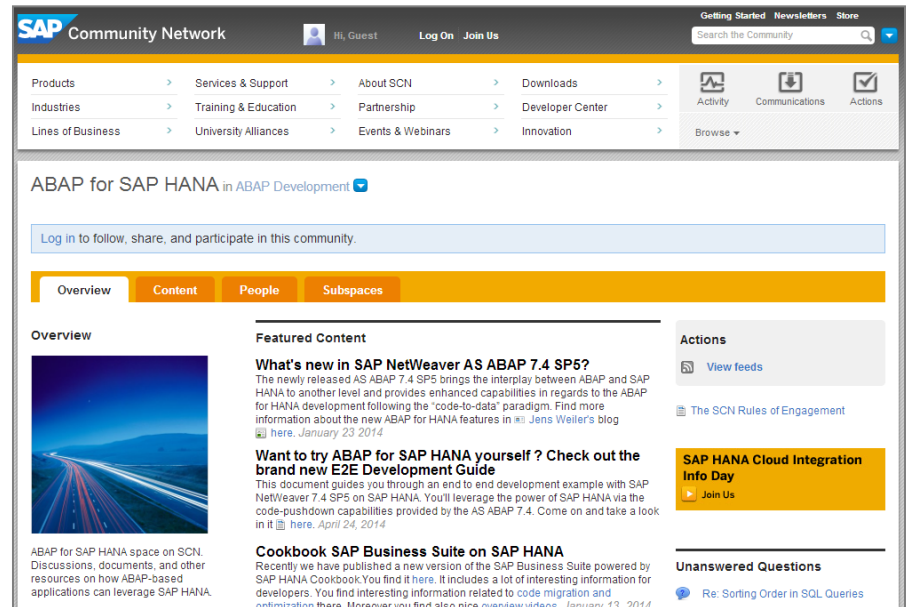
Looking Ahead
ABAP for SAP HANA CodeJam Events

Want to meet with ABAP for SAP HANA experts?

Join a 5 – 6 hour hands-on coding and networking event aimed at the developer community!

Find more information here:

- SAP Developer Program
  http://developers.sap.com

- Pictures of recent CodeJam events
  http://www.facebook.com/sapcodejam

- Upcoming CodeJam events
  http://scn.sap.com/community/events/codejam

- Interested in hosting your own event?
  Visit http://scn.sap.com/docs/DOC-37775
  or contact sapcodejam@sap.com

SAP CodeJam: 2012 – 2013 stats

52 cities | 9,500 hours of coding | Over 3,200 developer connections made


USA	POLAND
Chicago	Warsaw
Newtown Square	Wroclaw
Cleveland	SLOVAKIA
Miami	Bratislava
CANADA	TURKEY
Montreal	Istanbul
Toronto	MACEDONIA
PERU	Skopje
Lima	BULGARIA
COLOMBIA	Sofia
Bogota	RUSSIA
ARGENTINA	Moscow
Buenos Aires	ISREAL
CHILE	Tel Aviv
Santiago	Ra'anana
BRAZIL	U.A.E.
São Leopoldo	Dubai
UK	Abu Dhabi
London	CHINA
NORWAY	Shanghai
Oslo	Beijing
NETHERLANDS	SOUTH KOREA
Huizen	Daegu
Eindhoven	SINGAPORE
BELGIUM	SOUTH AFRICA
Antwerp	Durban
GERMANY	Cape Town
Bielefeld Hamburg	Johannesburg
Berlin Munich	AUSTRALIA
Stuttgart Walldorf	Melbourne
Hanover Frankfurt	Sydney
Ratingen Karlsruhe	INDIA
	Bangalore
	Pune
SPAIN	Mumbai
Madrid	
ITALY	SAUDI ARABIA
Rome	Al Khobar

Looking Ahead
What's Next?

Good luck for your final exam!

Thank you

Contact information:

open@sap.com

openSAP

© 2014 SAP AG 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 AG or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG (or an SAP affiliate company) in Germany and other countries. Please see http://global12.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.

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

National product specifications may vary.

These materials are provided by SAP AG or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP AG or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP AG 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 AG 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 AG's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP AG 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, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.