WELCOME

Oracle Database 12c New Security Features

Stefan Oehrli
Senior Consultant
Discipline Manager
Trivadis AG

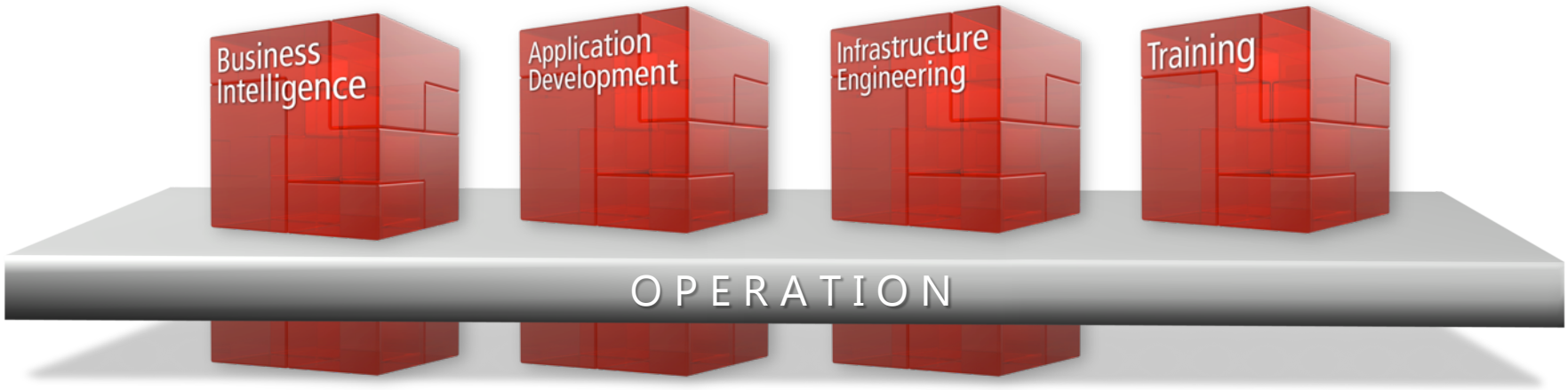
BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

trivadis
makes IT easier.

Our company

Trivadis is a market leader in IT consulting, system integration, solution engineering and the provision of IT services focusing on ORACLE and Microsoft technologies in Switzerland, Germany and Austria.

We offer our services in the following strategic business fields:



Trivadis Services takes over the interacting operation of your IT systems.

With over 600 specialists and IT experts in your region

12 Trivadis branches and more than 600 employees

200 Service Level Agreements

Over 4,000 training participants

Research and development budget: CHF 5.0 / EUR 4 million

Financially self-supporting and sustainably profitable

Experience from more than 1,900 projects per year at over 800 customers

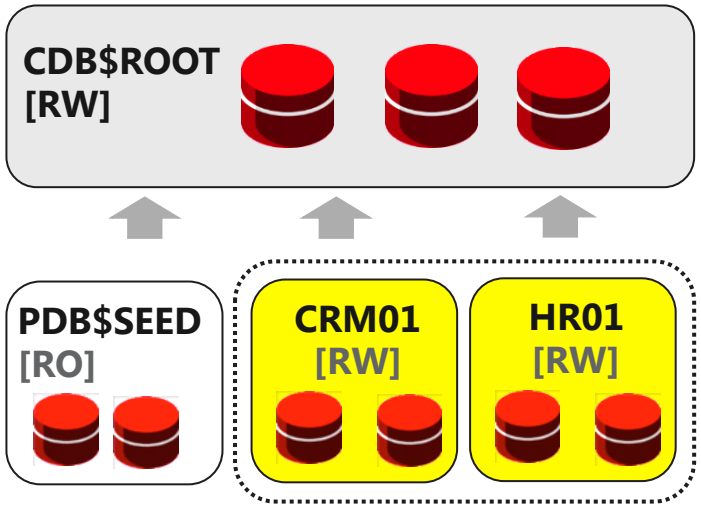
trivadis
makes IT easier.

Agenda

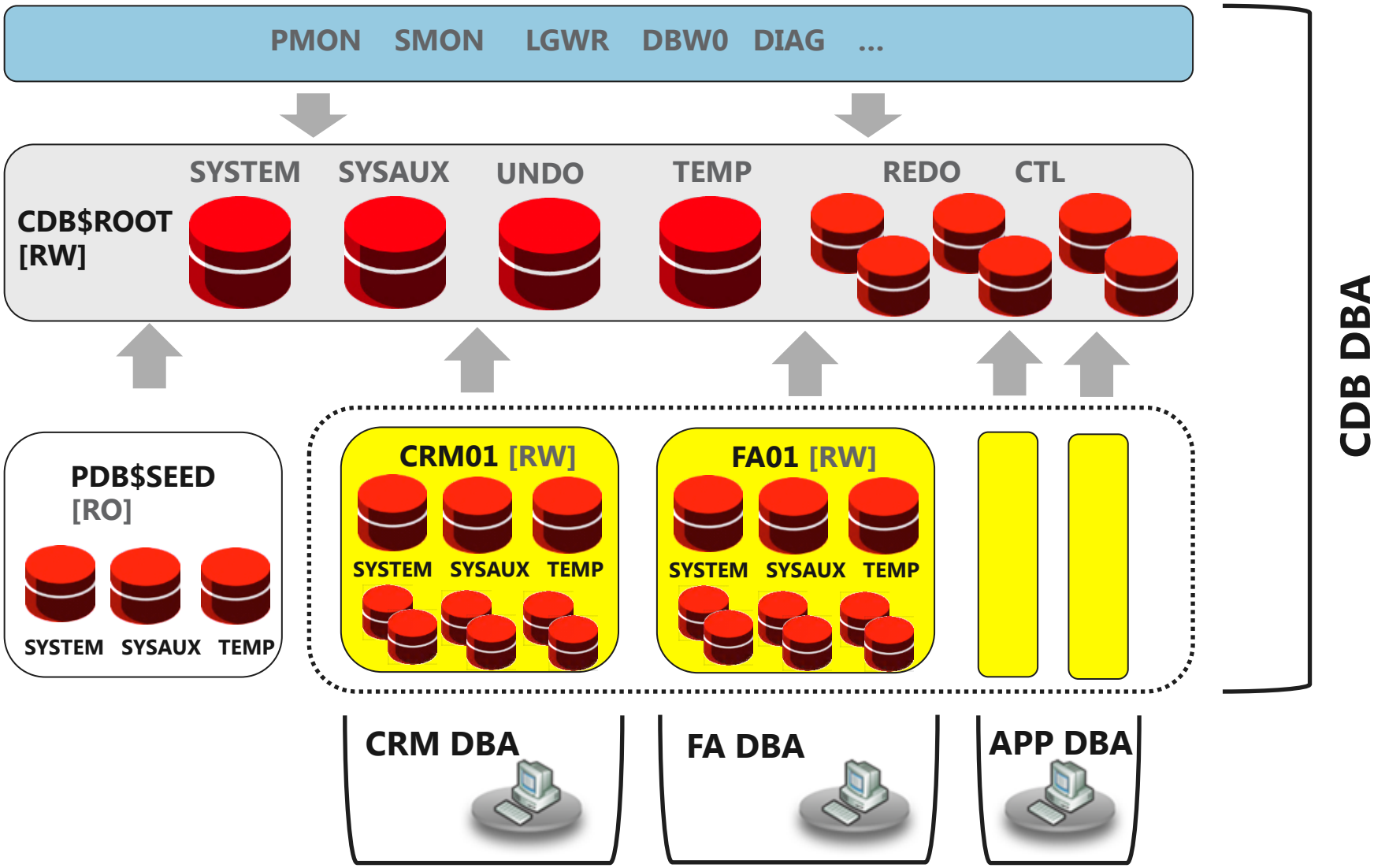
1. Multitenant Architecture
2. General Security Improvements
3. Data Redaction and Transparent Sensitive Data Protection
4. Database Auditing
5. Role and Privilege Analysis
6. Database Vault
7. Key and Wallet Management
8. Other security enhancements and features

Multitenant Architecture – Overview (1)

- The new multitenant architecture enables an Oracle database to function as a **container database** (**CDB**)
- A CDB can host up to 252 **pluggable databases** (PDBs)
  - Each PDB is compatible with a traditional non-CDB (same look and feel from an application point of view)
- The idea
  - Instead of managing many small databases on a server, we consolidate them into one large container database
- New architecture especially useful for
  - consolidation/database virtualization
  - fast and easy database provisioning
  - separation of administrative duties
  - rapid movement of user data (unplug/plug)



Multitenant Architecture – Overview (2)



Multitenant Architecture – Common/Local Entities

- User created **schema objects** (e.g. tables, indexes, PL/SQL code, etc.) are always **local** to a PDB and **not shared** between different containers

```
SQL> SELECT con_id, owner, object_name, object_type, sharing
  2  FROM cdb_objects WHERE object_name='T'
  3  AND owner='CRM01_ADMIN';

CON_ID OWNER        OBJECT_NAME OBJECT_TYPE SHARING
------ ------------ ----------- ----------- -------------
     3 CRM01_ADMIN  T           TABLE       NONE
```

- **Non-schema objects** like users or roles can be created as
  - **common**: The user or role exists in every current and future container
  - **local**: The user or role exists only in one PDB – similar to a non-CDB
- System or object privileges can be granted/revoked **commonly** or **locally**

Multitenant Architecture – Common Users/Roles

- The name of the common user/role must start with **C##** or **c##** (only ASCII or EBCDIC characters)
- **CONTAINER=ALL** clause is optional and the default, while being connected to the ROOT container

```
SQL> CREATE USER C##CDB_ADMIN1 IDENTIFIED BY PWD CONTAINER=ALL;

User created.
```

```
SQL> SELECT con_id, username, user_id, common
  2  FROM cdb_users where username='C##CDB_ADMIN1'
  3  ORDER BY con_id;

    CON_ID USERNAME                       USER_ID COMMON
---------- ------------------------------ ---------- ------
         1 C##CDB_ADMIN1                         112 YES
         3 C##CDB_ADMIN1                         107 YES
```

Multitenant Architecture – Local Users/Roles

- The local user/role name **cannot** begin with **C##** or **c##**
- Optionally use the **CONTAINER=CURRENT** clause (the default while being connected to a PDB)

```
SQL> CREATE USER crm01_admin IDENTIFIED BY pwd
  2  CONTAINER=CURRENT;

User created.
```

```
SQL> SELECT con_id, username, user_id, common
  2  FROM cdb_users where username='CRM01_ADMIN'
  3  ORDER BY con_id;

    CON_ID USERNAME                       USER_ID COMMON
---------- ------------------------------ ---------- ------
         3 CRM01_ADMIN                           108 NO
```

Multitenant Architecture – Granting/Revoking Privileges/Roles

- Privileges (system and objects) **granted commonly**
  - The grantor must be connect to the ROOT container
  - Can be used in all current as well as future database containers
  - Common privileges can be granted by common users **only** to common grantees

```
SQL> GRANT SELECT ANY TABLE TO C##CDB_ADMIN1 CONTAINER=ALL;
```

- Privileges (system and objects) **granted locally**
  - Can be used only in one container database (also locally in the ROOT)
  - Can be granted by common or local users to common or local users/roles
  - You can grant common roles to a local user, but they apply only locally

```
SQL> GRANT CREATE ANY TABLE TO C##CDB_ADMIN1 CONTAINER=CURRENT;
```

- **Omitting** the CONTAINER clause applies the privilege **locally**

Multitenant Architecture – Database Security Challenges

- In general more complex user and role concepts
- There are as well some other challenges like …

```
SQL> conn dbsnmp/dbsnmp
ERROR:
ORA-28001: the password has expired

Changing password for dbsnmp
New password:
Retype new password:
ERROR:
ORA-00600: internal error code, arguments: [kpdbModAdminPasswdInRoot: not
CDB],
[], [], [], [], [], [], [], [], [], [], []

Password unchanged
```

- Known as bug 16901482 and fixed in patch 16901482

Agenda

1. Multitenant Architecture
2. General Security Improvements
3. Data Redaction and Transparent Sensitive Data Protection
4. Database Auditing
5. Role and Privilege Analysis
6. Database Vault
7. Key and Wallet Management
8. Other security enhancements and features

General Security Improvements – ASO Licensing Changes

- Strong authentication services and network encryption are no longer part of Oracle Advanced Security
- Native network encryption and SSL/TLS can be used on any licensed editions of the Oracle Database
  - Simple setup of native SQLNet encryption without any additional costs
  - Kerberos, Radius and PKI can be used as authentication; e.g. it is possible to integrate database accounts with an Microsoft Active Directory
- This is available for 12c and newer databases.
  - Since 11.2.0.4 as well for Oracle 11g R2
- Enterprise User Security remains an Enterprise Feature and requires a corresponding Oracle Identity Management Directory Services Plus

General Security Improvements – Last Login Time

- Displayed on SQL*Plus login or in view DBA_USERS LAST_LOGIN
- Display can be turned off in SQL*Plus with –nologintime
- Is currently not recorded for logins of administrative users resp. password file users like SYSDBA, SYSOPER, SYSASM, SYSBACKUP, SYSDG, SYSKM

```
oracle@urania:~/ [TDB12] sqlplus test/test

SQL*Plus: Release 12.1.0.1.0 Production on Mon Aug 12 12:14:18 2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Last Successful login time: Mon Aug 12 2013 12:02:30 +02:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit
Production
```

General Security Improvements – Password verification (1)

- Oracle has improved/extended the utlpwdmg.sql script
  - Added new functions to simplify custom functions
  - Improved password verification functions for 12c
  - Password profile parameters considering recommendations from Center for Internet Security (CIS Oracle 11g) or Department of Defense (Database STIG v8R1)
- The script is still not automatically executed
- If it is executed, it changes the default profile applicable to all users
- New functions which could be used in custom verification functions
  - string_distance — Function to calculates the Levenshtein distance between two strings
  - complexity_check — Verifies the complexity of a password string

General Security Improvements – Oracle provided functions

Function	Password Length	Characters [a-z] [A-Z]	Upper Case [A-Z]	Lower Case [a-z]	Digits [0-9]	Special Characters	String Difference	Additional checks	Comments
verify_function	4	1	-	-	1	1	3	✔[1]	10g function
verify_function_11G	8	1	-	-	1	-	3	✔[2]	11g function
ora12c_verify_function	8	1	-	-	1	-	3	✔[3]	default
ora12c_strong_verify_function	9	-	2	2	2	1	4	-	

General Security Improvements – Administrative privileges (1)

Administrative Privilege	Usernames	Tasks
SYSDBA	SYS	Same operation as in 11g
SYSOPER	PUBLIC	Same operation as in 11g
SYSASM	SYS	Specific to ASM instances only
SYSBACKUP	SYSBACKUP	Perform RMAN backup &recovery operation from RMAN or through SQL
SYSDG	SYSDG	Perform Data Guard operations with Data Guard Broker or DGMGRL
SYSKM	SYSKM	Manage transparent data encryption wallet operations

General Security Improvements – Privileges, roles and grants

- Access control mechanism based on application code
  - Restricts exercise of privileges within specific code units
  - Minimizes privileges granted to runtime user
- Runtime privilege elevation in PL/SQL program units – Allows owner's roles to be granted to his program units
  - Functions, procedures and packages
  - Invoker rights and definer rights
  - Granted roles enabled during execution of the code

```
GRANT hr_admin TO procedure hr.checksalary_proc
```

- Granting the INHERIT PRIVILEGES privilege to other Users

```
GRANT INHERIT PRIVILEGES ON USER invoking_user TO
procedure_owner
```

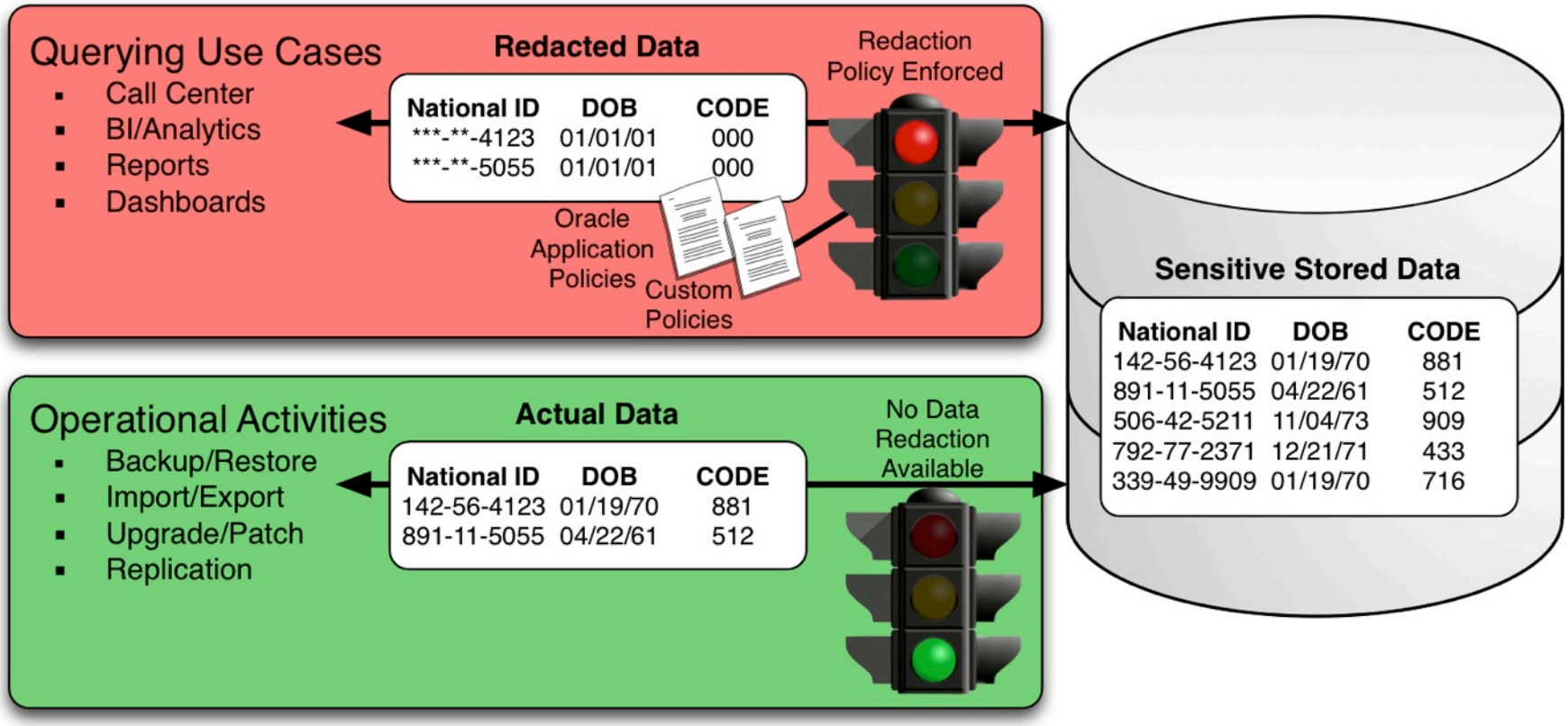
General Security Improvements – Virtual Private Database

- VPD Fine-Grained Context-Sensitive policies
  - Policy uses application contexts to determine which predicate to use
  - Associate VPD policy with one or more context/attribute
  - Policy function will only evaluated when context / attribute gets changed
  - Can be shared over multiple objects
- Support for long identifiers VPD object names
  - DBMS_RLS package and views support now maximum length of 128 bytes

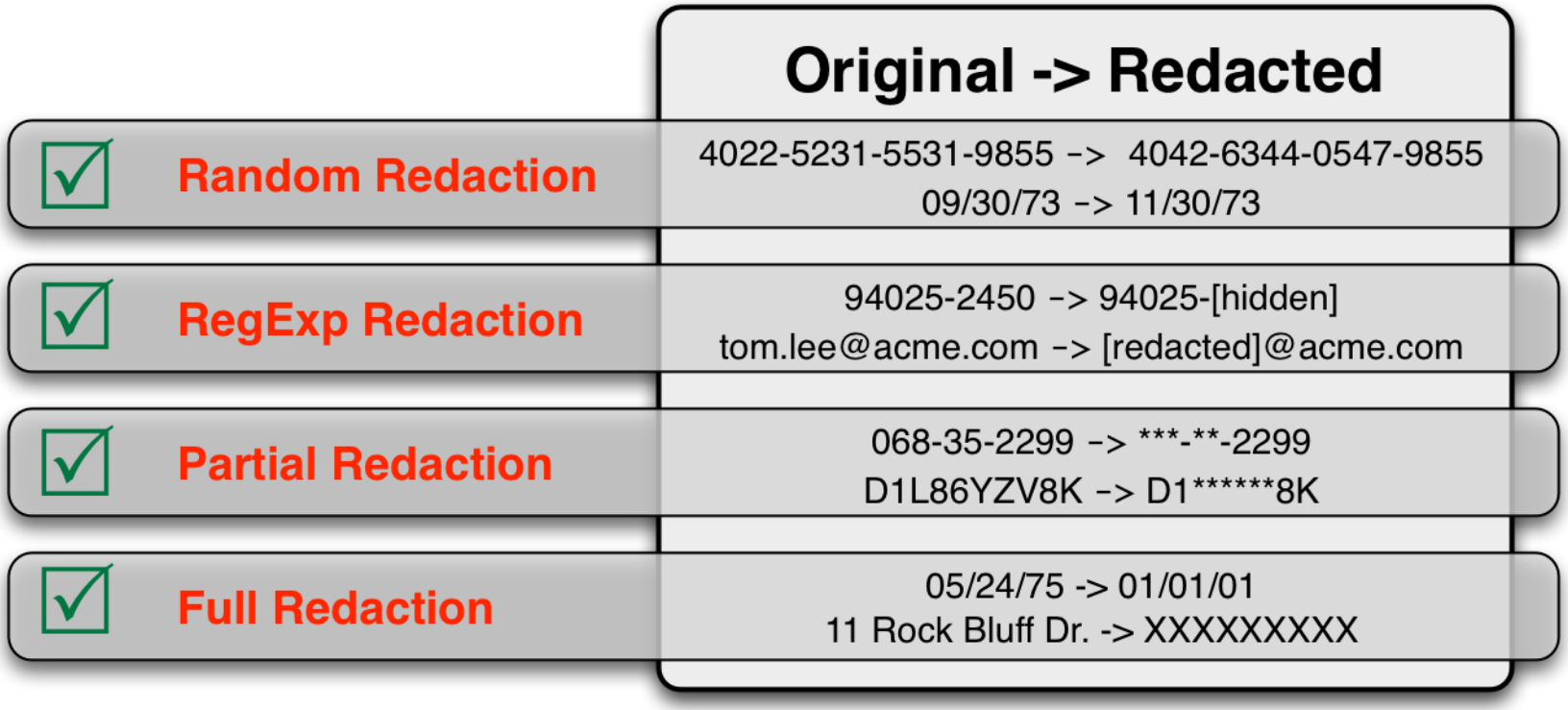
Agenda

1. General Security Improvements
2. Data Redaction
3. Database Auditing
4. Role and Privilege Analysis
5. Database Vault
6. Key and Wallet Management
7. Other security enhancements and features

Data Redaction – Overview



Data Redaction – Features



Data Redaction – Example

- Data redact is done based on a condition
  - Using SYS_CONTEXT to get user/role, IP address, client identifier, …
  - App user/role or other information passed in by the application
  - Supported functions: SYS_CONTEXT(), V(), NV() or DOMINATES ()
    ➔ *no custom PL/SQL*

```
dbms_redact.add_policy(
 object_schema => 'HR',
 object_name   => 'EMPLOYEES',
 column_name   => 'SALARY',
 policy_name   => 'HR_redact_salary',
 function_type => DBMS_REDACT.FULL,
 expression    => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'')!=
,'EUGEN''');
```

- List of existing redaction policies in REDACTION_POLICIES

Data Redaction – Restrictions (1)

- CTAS on redacted table does not work

```
create table hr.emp as select first_name,last_name,salary from hr.employees
where department_id=30
                                                      *
ERROR at line 1:
ORA-28081: Insufficient privileges - the command references a redacted
object.
```

- Export of redacted data with Data Pump is limited

```
ORA-31693: Table data object "HR"."EMPLOYEES" failed to load/unload and is
being skipped due to error:
ORA-28081: Insufficient privileges - the command references a redacted
object.
```

- New system privileges are required to bypass redaction policies
  - EXEMPT REDACTION POLICY
  - EXEMPT DML REDACTION POLICY
  - EXEMPT DDL REDACTION POLICY

Data Redaction – Restrictions (2)

- Not all data types are supported for data redaction
  - Not supported data types: ROWID, RAW, INTERVAL, GRAFIC, user defined types and Oracle supplied types like XML, SPATIAL and MEDIA types
  - BLOB and CLOB are supported for FULL redaction only, shown as *[redacted]*
- Redaction does not work on editioned views
- Data redaction policies apply only on the objects in the current pluggable database in a multitenant environment
- Object types cannot be redacted
- Limitations when using aggregate functions, certain SQL queries cannot take full advantage of database optimizations that presume the row values to be static

Transparent Sensitive Data Protection – Overview

- Define sensitive data types within the database
- Classify the data to be protect
  - E.g. Sensitive column with salary, credit card number etc.
- Protect a given class with TSDP policies
  - Protect data a column level with VPD or data redaction
  - Use/define uniform policy for all classified data
- Export TSDP policies
- Apply TSDP policies across other databases
  - Protect sensitive data company wide

Agenda

1. Multitenant Architecture
2. General Security Improvements
3. Data Redaction and Transparent Sensitive Data Protection
4. Database Auditing
5. Role and Privilege Analysis
6. Database Vault
7. Key and Wallet Management
8. Other security enhancements and features

Database Auditing – The UNIFIED AUDIT

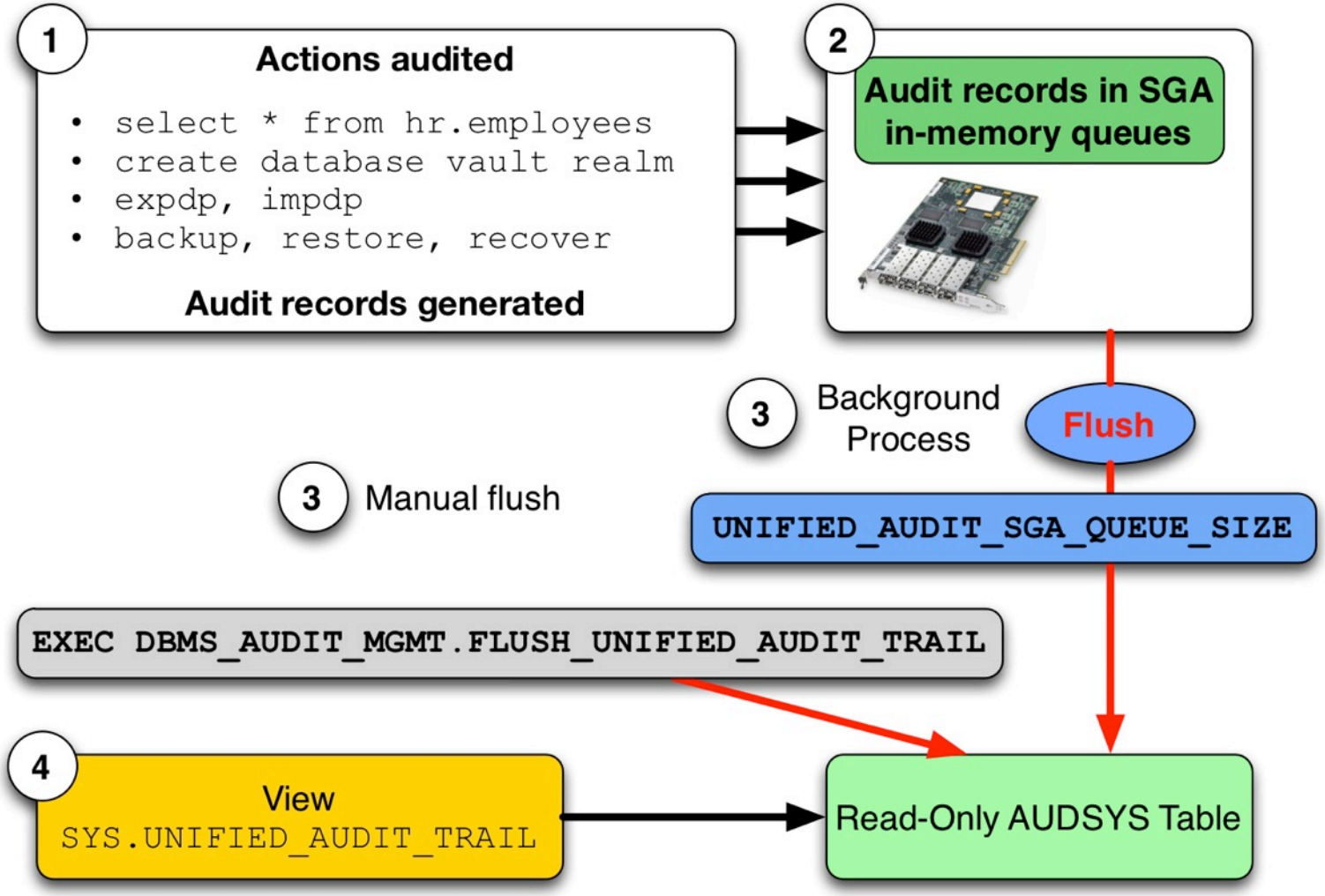
- Oracle introduces the new unified audit trail
  - One single audit trail for any audit data
  - unified_audit_trail view replaces SYS.AUD$, SYS.FGA_LOGS$, DVSYS.AUDIT_TRAIL$, OS audit files in adump, etc
  - All audit data stored in Oracle secure files
  - Security with new AUDITOR and AUDIT_ADMIN accounts
- Always ON auditing
  - No initialization parameters required to enable auditing
  - No need to bounce the database (ehm. At least once… ☺ to link it)
- Audit the audit configuration by default
  - Records every event that modifies the audit configuration
  - Records every modification to audit trail and its settings

Database Auditing – The UNIFIED AUDIT

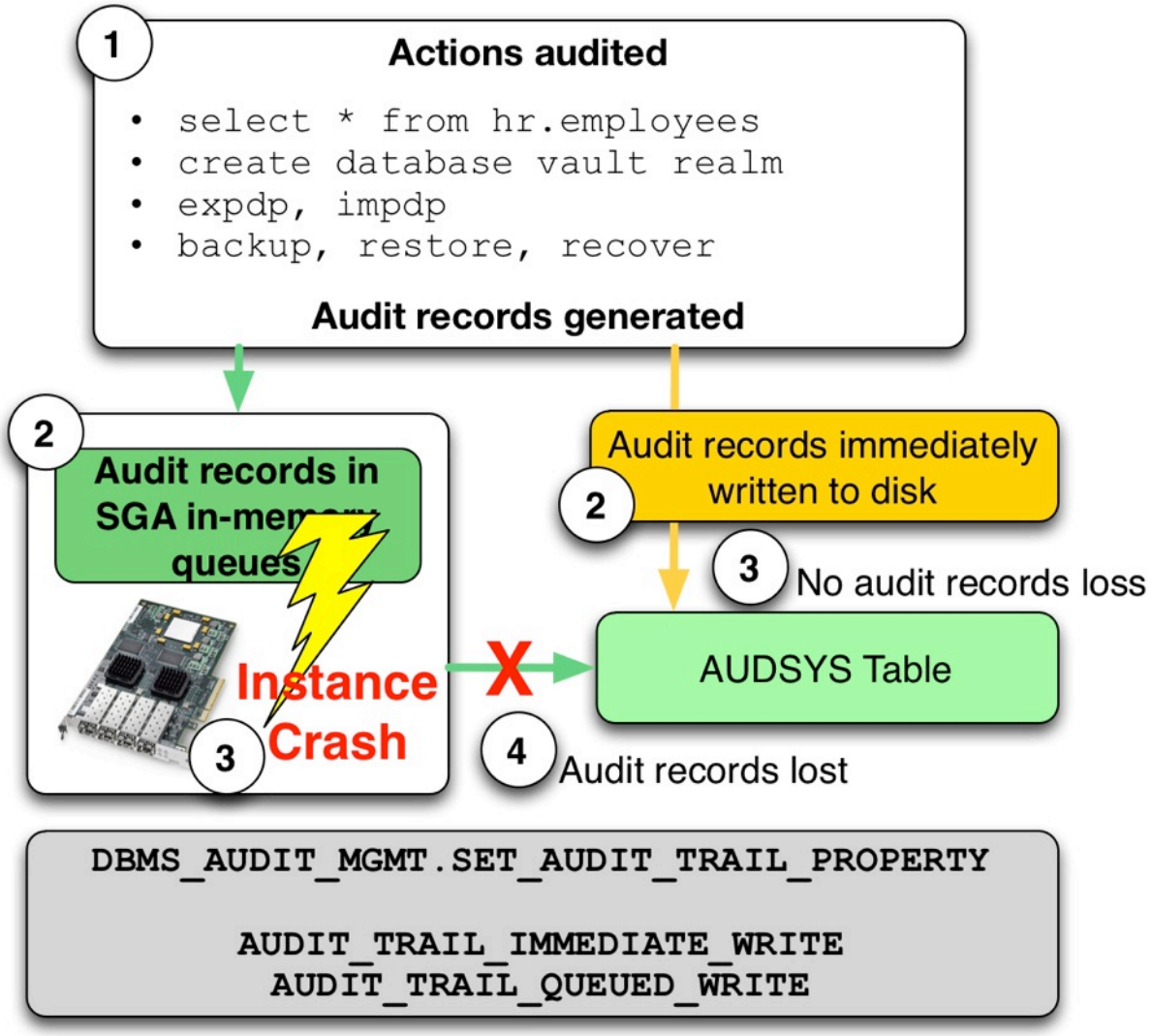
- Fast audit engine, easier access control to DB, increased performance
  - Low processing overhead (records are stored in proprietary format)
  - Low transactional overhead (audit records are buffered)
  - Dynamic views to query audit data stored in proprietary format
- Queued Mode
  - Default mode
  - Audit records stored in SGA and periodically flushed
  - Configured with UNIFIED_AUDIT_SGA_QUEUE_SIZE (1MB to 30MB)
- Immediate Mode
  - Audit records written immediately
- Manual flush queue to disk
  - Connect as user with AUDIT_ADMIN role

```
DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL
```

Database Auditing – Fast audit engine



Database Auditing – Ups…



Database Auditing – Audit policies

- Audit policies are named containers for audit settings
  - Are used to audit ACTIONS, PRIVILEGES, OBJECTS
  - Are based on system wide or object-specific audit options
  - Can contain a role
  - Can contain conditions / exceptions
  - Are enabled / disabled with audit and noaudit statement
- Condition limited to Oracle functions ➔ no custom PL/SQL functions

Database Auditing – Audit policies

- Create audit policy with conditions and exceptions

```
CREATE AUDIT POLICY dba_pol ROLE DBA;

CREATE AUDIT POLICY hr_employees_pol
    PRIVILEGES CREATE TABLE
    ACTIONS UPDATE ON HR.EMPLOYEES
    WHEN 'SYS_CONTEXT(''USERENV'', ''IDENTIFICATION_TYPE'') =
''EXTERNAL''' EVALUATE PER STATEMENT;

AUDIT POLICY hr_employees_pol EXCEPT HR;
```

- Enabled audit policies

```
SELECT * FROM audit_unified_enabled_policies;

USER_NAME    POLICY_NAME          ENABLED_ SUC FAI
-----------  -------------------- -------- --- ---
SCOTT_DBA    ORA_ACCOUNT_MGMT     BY       YES YES
ALL USERS    ORA_SECURECONFIG     BY       YES YES
```

Database Auditing – More on “unified”

- The unified audit trail is also used to store audit information for
  - Fine Grained Audit (FGA)
  - Data Pump
  - Oracle RMAN
  - Oracle Label Security (OLS)
  - Oracle Database Vault (DV)
  - Real Application Security (RAS)
- Component auditing do use dedicated columns
  - RMAN_OPERATION, RMAN_OBJECT_TYPE, RMAN_DEVICE_TYPE
  - DP_TEXT_PARAMETERS1, DP_BOOLEAN_PARAMETERS1
- Can be specified as well in an audit policy

```
CREATE AUDIT POLICY audit_dp
ACTIONS COMPONENT=DATAPUMP ALL
```

Database Auditing – It does get harder to tamper audit

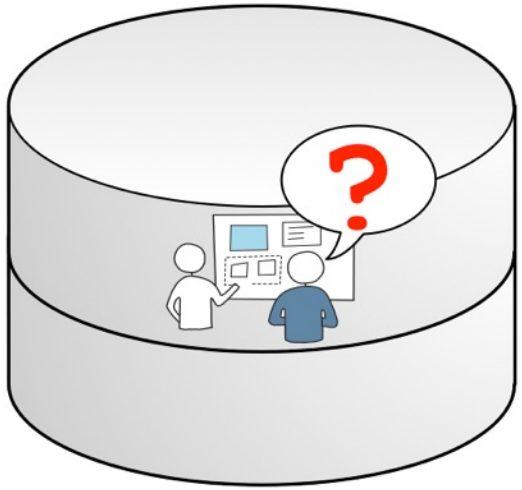
- Unified Audit is part of the oracle kernel
  - switch off require relink / restart
  - Using a different oracle binary at runtime e.g. for sqlplus lead to errors / ORA-00600
  - Auditing is partially available even if relinked with **uniaud_off**
- Memory could be manipulated before it has been flushed
  - Use immediate mode to minimize the risk
- ORADEBUG statements are audited by default
- Unified Audit uses $ORACLE_BASE/audit to store unified audit binary files when DB is not open or writable
  - Transparent access through View UNIFIED_AUDIT_TRAIL
  - Files can be loaded into the database with DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES

Agenda

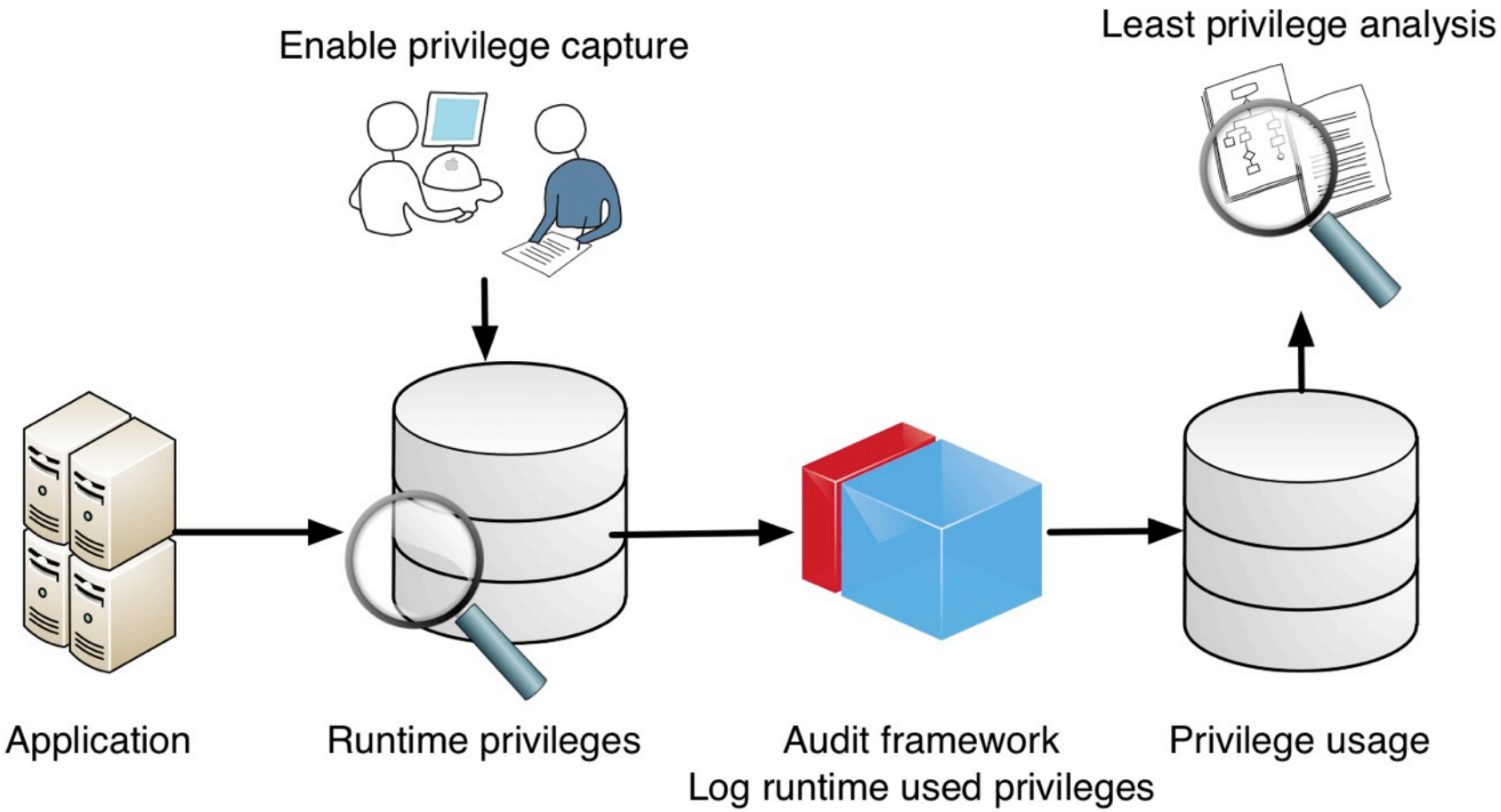
1. Multitenant Architecture
2. General Security Improvements
3. Data Redaction and Transparent Sensitive Data Protection
4. Database Auditing
5. Role and Privilege Analysis
6. Database Vault
7. Key and Wallet Management
8. Other security enhancements and features

Role and Privilege Analysis – Overview

- Capture and report on database privilege usage at runtime
  - For users, sessions, roles, PUBLIC
  - Show used system, object, and PUBLIC privileges
  - Show how the user got the privilege
- Show unused privileges:
  - System and object
- Achieve least privilege model
  - Make the database and applications more secure



Role and Privilege Analysis – Architecture



Role and Privilege Analysis

- Create the capture policy

```
DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
NAME => 'dba_privilege_analysis', type => DBMS_PRIVILEGE_CAPTURE.G_CONTEXT,
CONDITION=> 'SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ''SCOTT''')
```

- Enable the capture policy

```
DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE('dba_privilege_analysis')
```

- Run Job, Task etc which has to be analyzed
- Disable the capture policy

```
DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE('dba_privilege_analysis')
```

- Generate report

```
DBMS_PRIVILEG_CAPTURE.GENERATE_RESULT('dba_privilege_analysis')
```

- Review views DBA_USED_% and DBA_UNUSED_%

Agenda

1. Multitenant Architecture
2. General Security Improvements
3. Data Redaction and Transparent Sensitive Data Protection
4. Database Auditing
5. Role and Privilege Analysis
6. Database Vault
7. Key and Wallet Management
8. Other security enhancements and features

Database Vault – Improvements in 12c

- Manageability
  - Streamline controls enforcement via Enterprise Manager 12c (☺☹)
  - One command enablement, no special installation required
  - New delivered realms to protect sensitive metadata
- New mandatory realms feature
  - Block all privileges from accessing data – even owner
  - Patching, maintenance, sensitive information eg. role DV_PATCH_ADMIN
- Improved performance
- Installation
  - Installed by default but not configured ➔ removes reliance on OS for linking
  - Protection is always on no matter where you restore DB backup
  - Support regular and container databases

Database Vault – Configuration (1)

- Create a security admin user as DBA

```
GRANT CREATE SESSION TO sec_admin identified by manager
```

- Create an accounts admin

```
GRANT CREATE SESSION TO accts_admin identified by manager;
```

- One command to configure as SYS

```
dvsys.configure_dv(dvowner_uname => 'sec_ADMIN',
dvacctmgr_uname => 'accts_admin')
```

- Then enable as security admin sec_admin

```
dvsys.dbms_macadm.enable_dv
```

- Restart the database as SYSDBA

Database Vault – Configuration (2)

- Container Database provide common DVSYS and DVF users
- DB Vault policies are scoped to individual PDB
  - Each PBD has its own database vault metadata
- DB Vault is configured and enabled at PDB level
  - Database Vault must first be enabled in root container
  - ORA-47503: Database Vault is not enabled on CDB$ROOT
- V$OPTION does show at container level if database vault is enabled
  - Require to set the container first

```
SELECT * FROM v$option WHERE parameter = 'Oracle Database Vault';

PARAMETER                      VALUE
------------------------------ ----------
Oracle Database Vault          TRUE
```

Agenda

1. Multitenant Architecture
2. General Security Improvements
3. Data Redaction and Transparent Sensitive Data Protection
4. Database Auditing
5. Role and Privilege Analysis
6. Database Vault
7. Key and Wallet Management
8. Other security enhancements and features

Advanced Security Key Management

- New key attributes to help track key expiration, last rekey, etc.
- Additional data dictionary views to summarize keys and their attributes
- New commands to consolidate actions previously in distinct utilities
- Import/export feature to move individual keys between wallets
- Migrate/reverse migrate to move keys between wallet & HSM
- Automatic backup of wallet-based keystores
- Updated TDE page in EM12c for simple management of keys and key stores
- TDE master keys are managed independently within the wallet
  - Are rotated within the wallet independently
  - Can be imported/exported between wallets

Advanced Security Key Management – New commands

- Create a new password-based wallet / key store

```
ADMINISTER KEY MANAGEMENT
CREATE KEYSTORE '/u00/app/oracle/etc/wallets/TDB12'
IDENTIFIED BY "manager"
```

- Creating and Activating a Master Encryption Key with a backup

```
ADMINISTER KEY MANAGEMENT SET KEY USING TAG 'TDB12Master'
IDENTIFIED BY "manager" WITH BACKUP
```

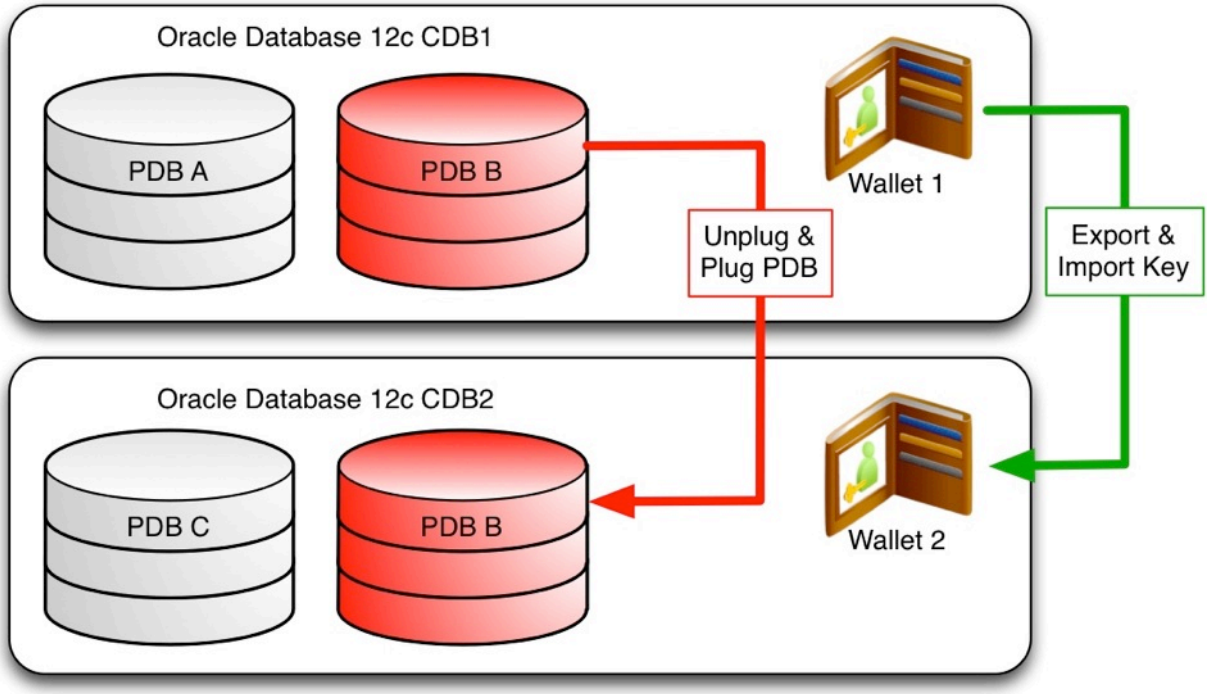
- Query some new key attributes

```
SELECT tag,crator,backed_up FROM v$encryption_keys

TAG                  CREATOR    BACKED_UP
-------------------- ---------- ---------
TDB12Master_TEST     SYS        NO
TDB12Master          SYS        YES
```

Advanced Security Key Management – Multitenant Databases

- The wallet lives in the host environment, not within PDB
  - A single wallet accessed by multiple PDB running on the host
  - Each PDB using encryption has a TDE master key stored in the wallet



Agenda

1. Multitenant Architecture
2. General Security Improvements
3. Data Redaction and Transparent Sensitive Data Protection
4. Database Auditing
5. Role and Privilege Analysis
6. Database Vault
7. Key and Wallet Management
8. Other security enhancements and features

Other security enhancements

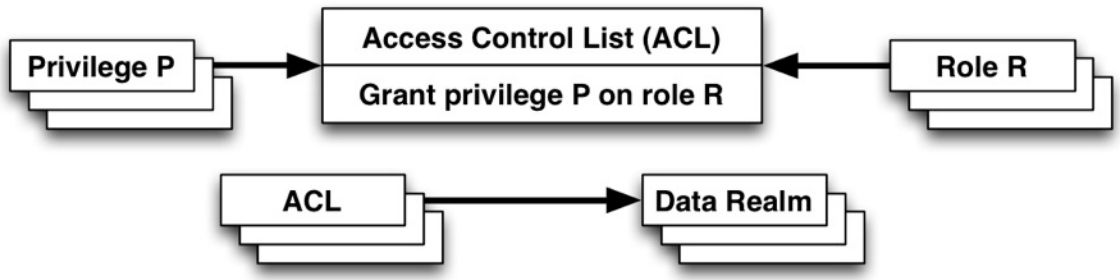
- Sensitive data dictionary tables
  - The SELECT ANY DICTIONARY privilege no longer permits access to security sensitive data dictionary tables DEFAULT_PWD$, ENC$, LINK$, USER$, USER_HISTORY$, and XS$VERIFIERS.
- UNLIMITED TABLESPACE
  - RESOURCE Role does not grant UNLIMITED TABLESPACE any more
  - UNLIMITED TABLESPACE must be granted manually if required
- Partially support for SHA-2
  - SHA-2 as the hashing algorithm to sign security certificates for use with SSL
  - PL/SQL DBMS_CRYPTO and JVM do both support SHA-2 algorithm
  - SHA-2 for Database Authentication is not yet available

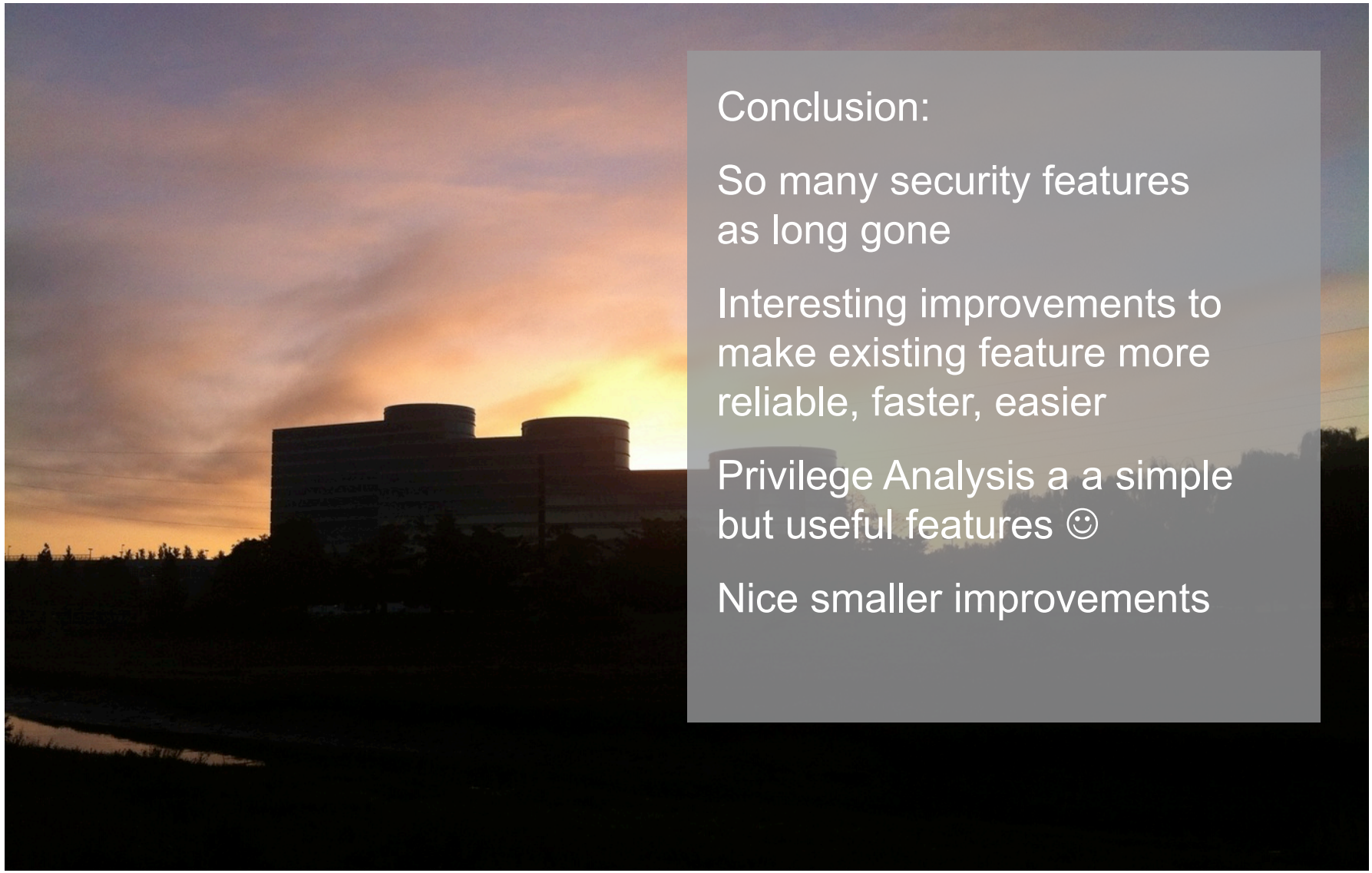
Other security enhancements

- Multiple authentication support
  - Database will fall back to password authentication
- New Kerberos stack
  - Replaced old Kerberos implementation
- Hardware acceleration support extended beyond TDE
  - Now supported for Network Encryption and DBMS_CRYPTO
- New Secure Sockets Layer Cipher Suites
  - Support for Elliptic curve Diffie–Hellman (ECDHE) and Elliptic Curve Digital Signature Algorithm (ECDSA)
- ASM now supports storing password files inside ASM disk groups
  - Password files for ASM or Database can be stored in a disk group
  - Migration of exiting password files with orapwd

Other security enhancements

- Better security for external procedures
  - Run with designated OS credentials
  - Configured with the new DBMS_CREDENTIAL
  - Enhanced CREATE LIBRARY associate EXTPROC user with a library
- DBMS_NETWORK_ACL_ADMIN
  - Update procedures for host ACL
  - New procedures for wallet ACL
- Real Application Security
  - Efficient db-enforced data access control
  - Application privileges and roles
  - Application users and sessions





THANK YOU.

Trivadis AG

Stefan Oehrli

Europa-Strasse 5
CH-8152 Glattbrugg

www.trivadis.com
www.oradba.ch

BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

trivadis
makes IT easier.