

PDB Isolation and Security

What about security of Oracle multitenant container database (CDB)?

Stefan Oehrli



@stefanoehrli



www.oradba.ch

BASEL | BERN | BRUGG | BUCHAREST | COPENHAGEN | DÜSSELDORF | FRANKFURT A.M. | FREIBURG I.BR.
GENEVA | HAMBURG | LAUSANNE | MANNHEIM | MUNICH | STUTTGART | VIENNA | ZÜRICH

trivadis

Trivadis – Our key figures



- Founded in 1994
- 16 Trivadis locations with more than 650 employees
- Sales of CHF 111 million (EUR 96 million)
- Over 250 Service Level Agreements
- More than 4000 training participants
- Research and development budget: CHF 5.0 million
- More than 1900 projects each year with over 800 customers
- Financially independent and sustainably profitable

Stefan Oehrli

Platform Architect, Trainer and Partner at Trivadis

- Since 1997 active in various IT areas
- Since 2008 with Trivadis AG
- More than 20 years of experience in Oracle databases

Focus: Protecting data and operating databases securely

- Security assessments and reviews
- Database security concepts and their implementation
- Oracle Backup & Recovery concepts and troubleshooting
- Oracle Enterprise User Security, Advanced Security, Database Vault, ...
- Oracle Directory Services

Co-author of the book The Oracle DBA (Hanser, 2016/07)

 @stefanoehrli

 www.oradba.ch



BASEL | BERN | BRUGG | BUCHAREST | COPENHAGEN | DÜSSELDORF | FRANKFURT A.M. | FREIBURG I. BR.
GENEVA | HAMBURG | LAUSANNE | MANNHEIM | MUNICH | STUTTGART | VIENNA | ZÜRICH

trivadis

Agenda



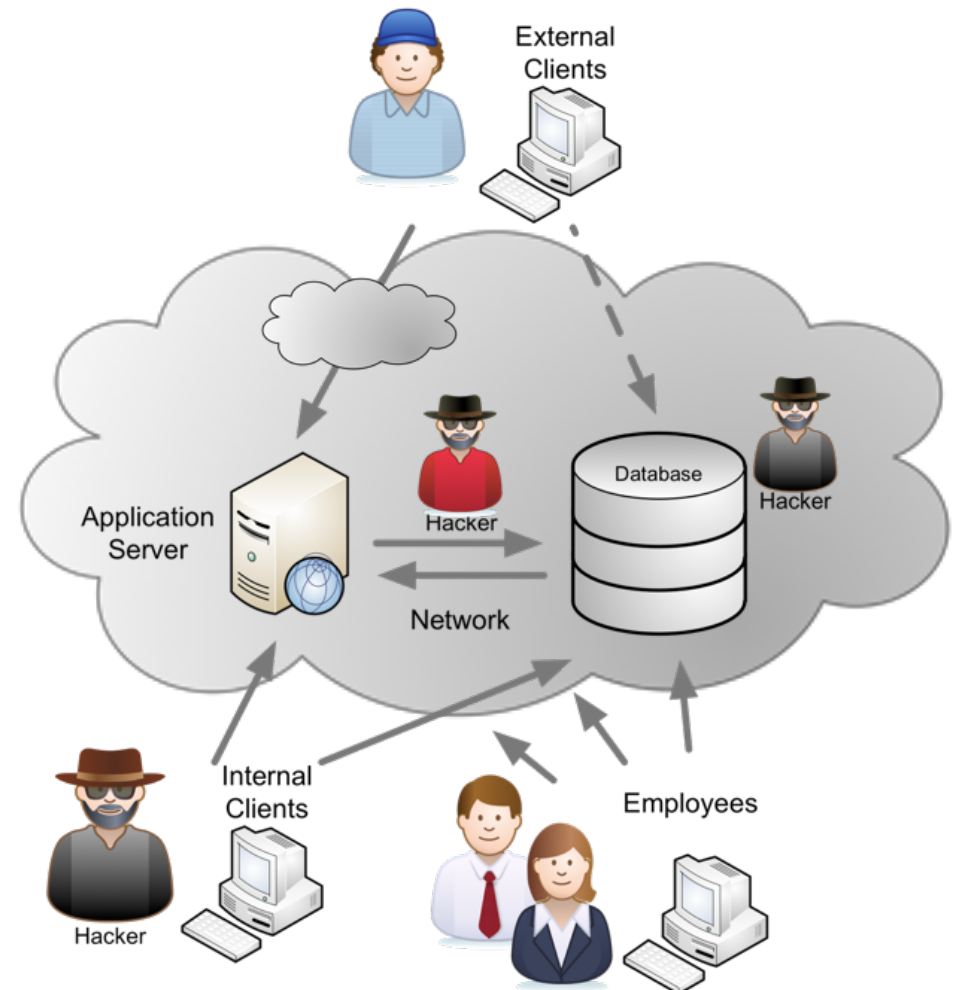
- General Database Security
- Multitenant Container Database Challenges
- Isolation and Security Measures
- Outlook and potential enhancements
- Summary

General Database Security

Top 10 Database Risks and Threats

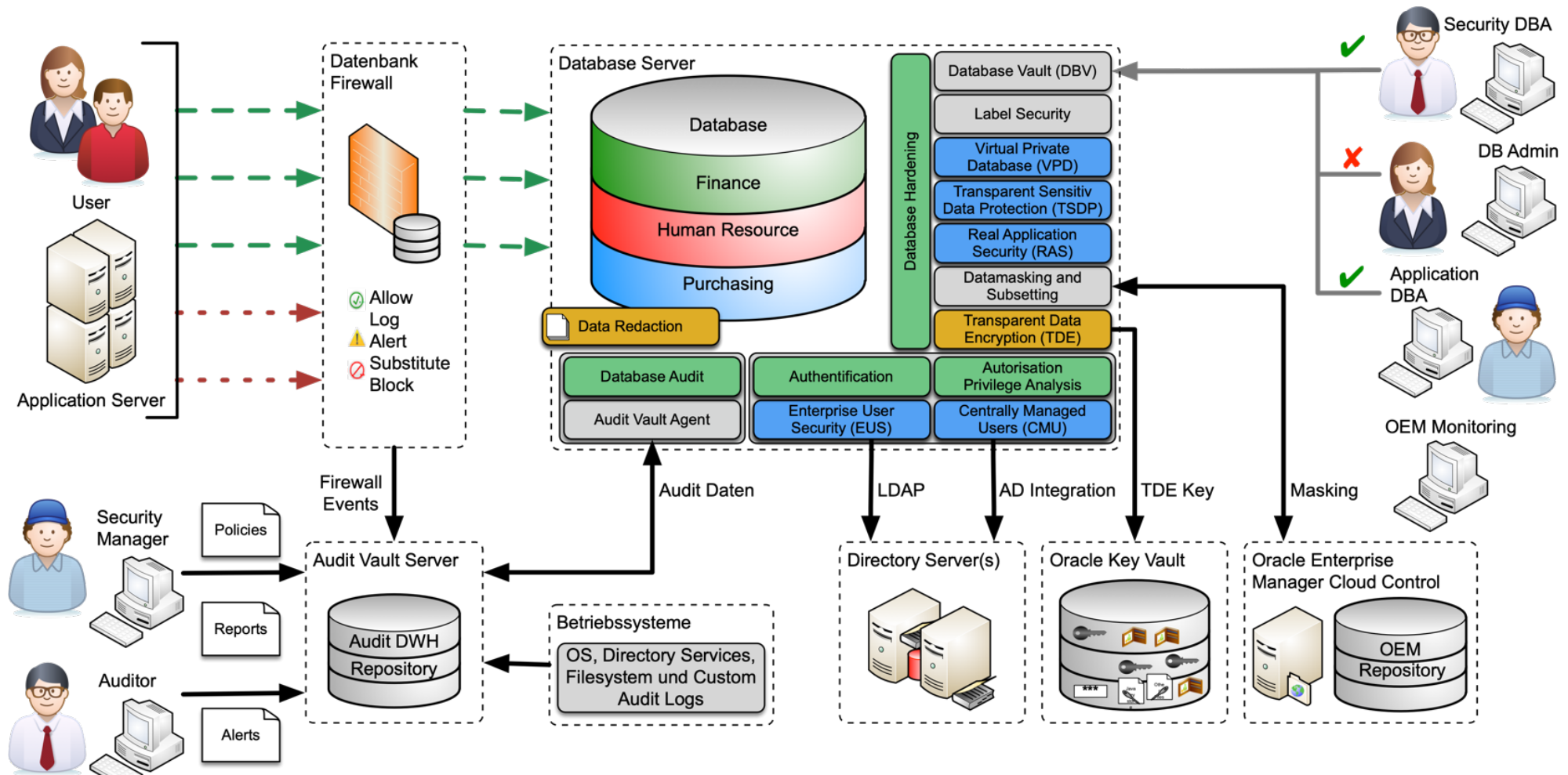
trivadis

- Excessive privileges
- Privilege abuse
- Unauthorized privilege elevation
- Platform vulnerabilities
- SQL injection
- Weak audit
- Denial of service
- Database protocol vulnerabilities
- Weak authentication
- Exposure of backup data



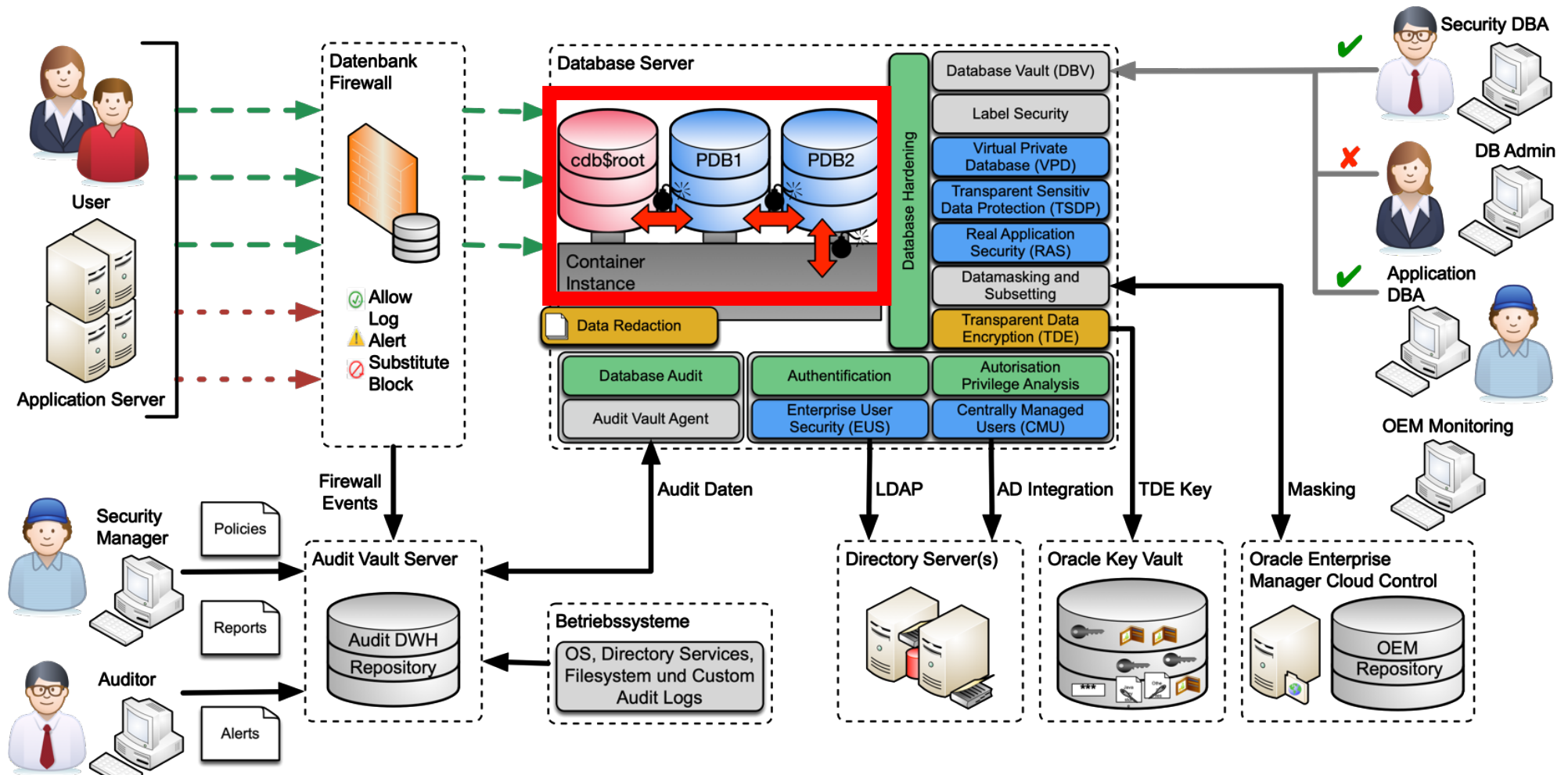
Maximal Data Security Architecture

trivadis



Maximal Data Security Architecture - CDB

trivadis



Multitenant Container Database Challenges

Multitenant Container Database Usage

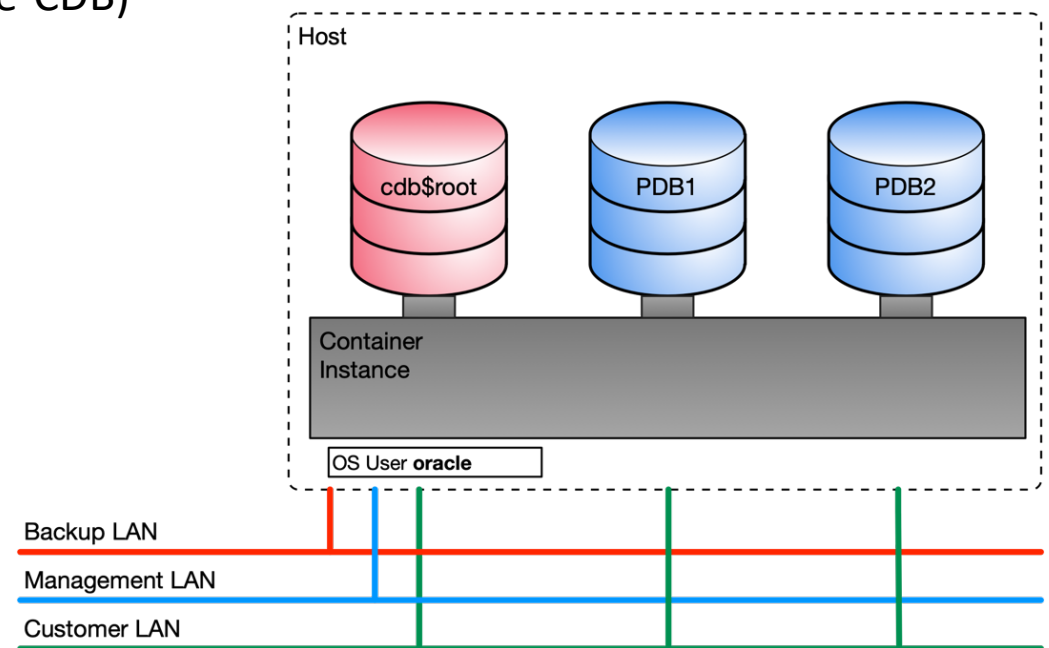


How are the multitenant container databases used?

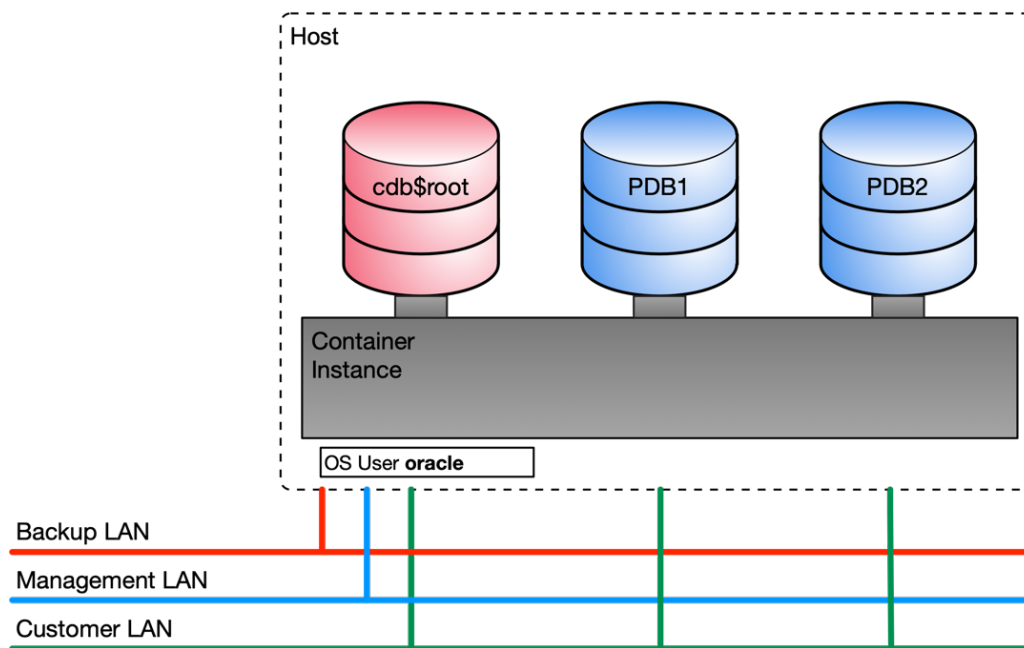
- Simple replacement of the classic database (none-CDB) architecture with just one PDB
- Database consolidation using multiple PDBs
- Private DBAAS
- Public DBAAS

Infrastructure architecture

- Cloud, Hybrid or on-premises?
- Dedicated hardware
- Virtual environments
- Engineered System



Multitenant Container Database Usage



Security related questions

- Comprehensive privileges available in PDBs
- Strict security requirements and standards
- Industry dependent legal and compliance requirements

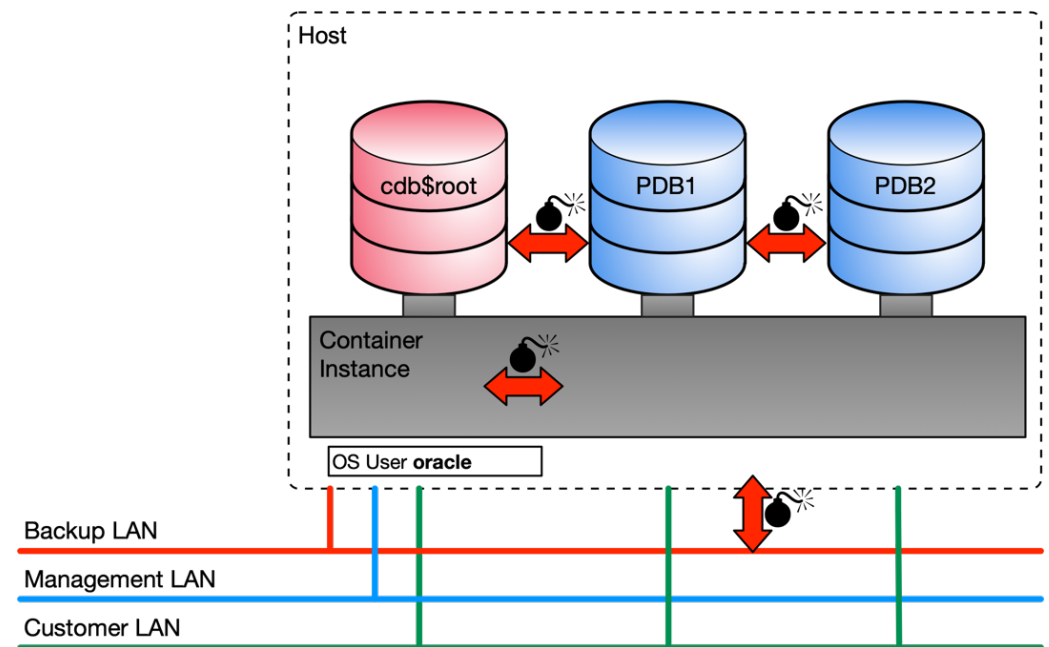
Corporate structure

- All just one legal entity
- More or less independent departments
- Several companies and subsidiaries
- Service provider for different companies

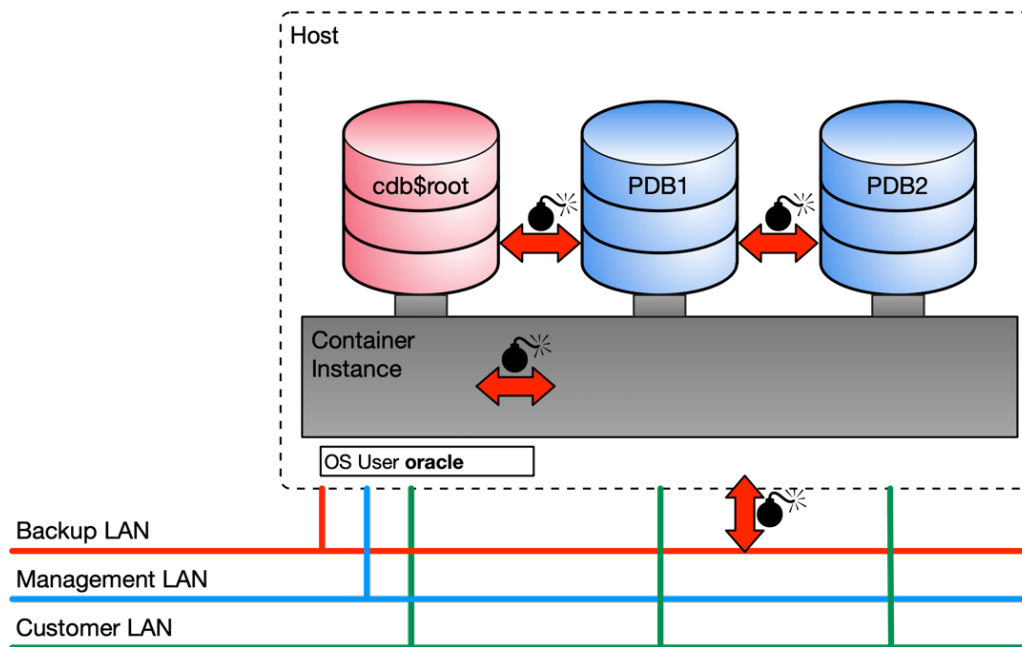
Isolation requirements vary

But why isolation at all?

- Just basic database security is not sufficient.
- PDB admin and user do have comprehensive privileges (DBAAS).
 - Full DBA role
 - ALTER SYSTEM, ALTER SESSION,...
 - PL/SQL packages and procedures
- Oracle bugs **and** feature allow to escape the boundaries of a PDB.
 - Scheduler jobs including OS calls
 - External table pre-processor scripts
 - PL/SQL Library calls
 - Java OS calls
- Resource management beyond the scope of PDBs



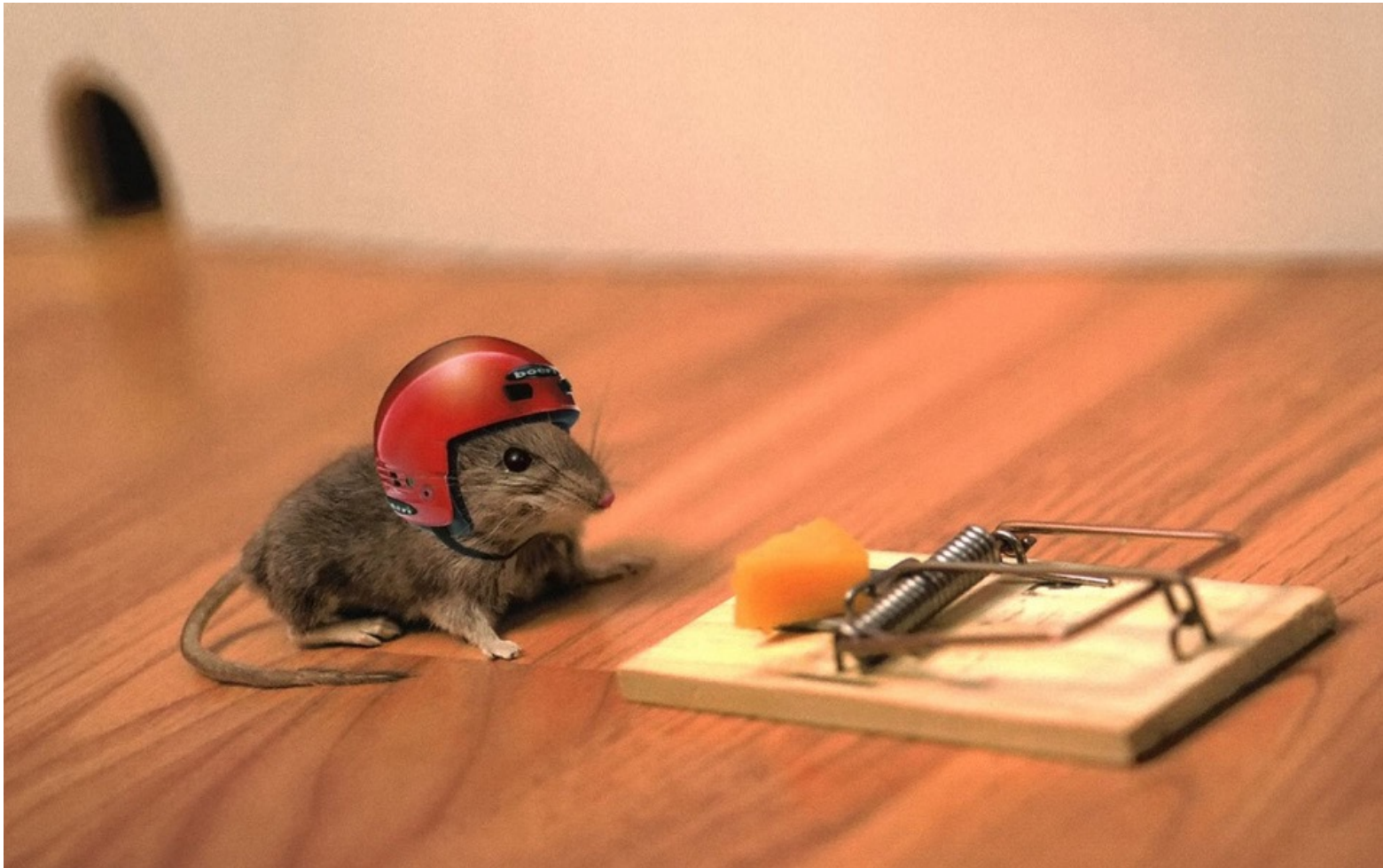
Risks in a DBAAS environment



- PDB admin use **privilege escalation**.
- **Excessive** use of shared resources.
- Access sensitive data via shared resources e.g. backup or management LAN.
- Break out of PDB and get OS access as **oracle**.
- Gain access to the **root container** (cdb\$root)
- Gain access to other PDBs.
- Gain access to the **network**.
- Use of **critical features** like
 - Administration features
 - Oracle JVM
 - DBMS_SCHEDULER
 - External table pre-processor

Risk all cleared?

trivadis



Isolation and Security Measures

Will we cook or why onions?

- Measures on several levels are **absolutely** essential.
- Layered structure similar to an onion. E.g.
 - Secure and correct data
 - Secure application design
 - PDB hardening
 - PDB isolation and security
 - General CDB architecture
 - CDB hardening
 - OS Hardening
 - Network security
- The following functions such as lockdown profiles, path prefix and PDB OS credentials are only part of a holistic approach.

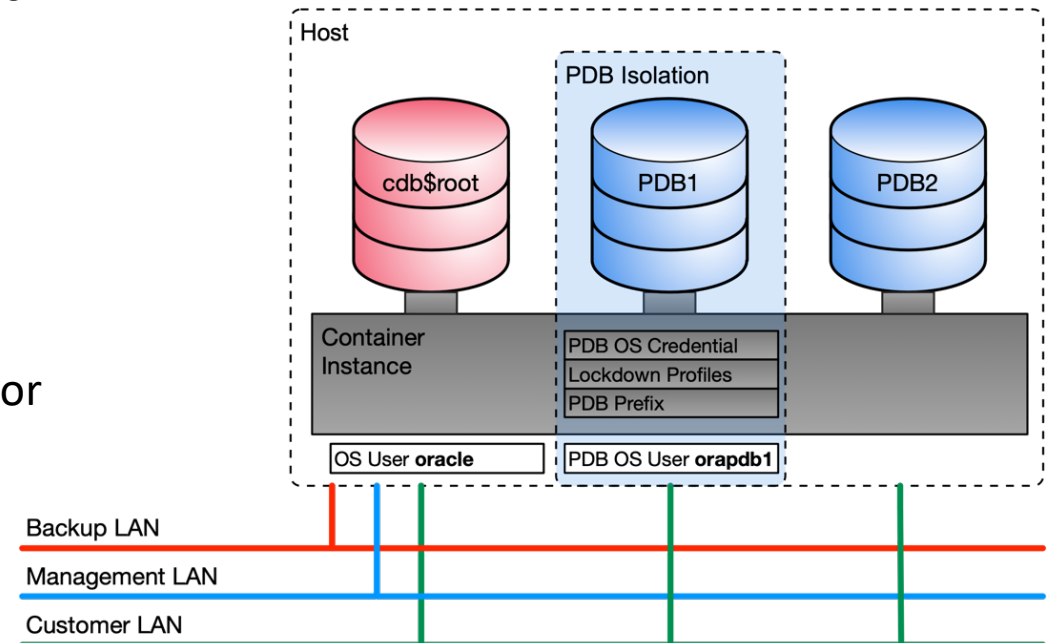


Possibilities for risk mitigation

A multitenant container database provides the following features beyond regular security measures:

- **PATH_PREFIX** and **CREATE_FILE_DEST** clause to limit data files and directory objects to certain paths.
- **PDB_OS_CREDENTIAL** parameter assigning a dedicated user account for OS interactions
- **Lockdown profiles** to restrict certain operations or functionalities in a PDBs

But what can you do with them...?



Managing OS Access



- Define a dedicated OS user account for system interactions.
- Based on **DBMS_CREDENTIAL** package and init parameter **PDB_OS_CREDENTIAL**.
- Allows to define dedicated OS credential for:
 - External jobs that do not yet have an OS credential specified
 - External table pre-processors
 - PL/SQL library executions (EXTPROC)
- Can be defined globally or per PDB.
- Introduced with Oracle 12.2.0.1.

As far as theory goes...

- ... unfortunately there are still bugs (12g-19c)
- ... PDB_OS_CREDENTIAL does not work as expected for table pre-processor and EXTPROC.

Define credentials

- Create a credential in the *cdb\$root* for a corresponding OS user

```
BEGIN
  dbms_credential.create_credential(
    credential_name => 'PDBSEC_OS_USER',
    username        => 'orapdbsec',
    password        => 'manager');
END;
/
```

- Set the parameter at CDB level (older version did require a change via pfile)

```
ALTER SYSTEM SET pdb_os_credential=GENERIC_PDB_OS_USER SCOPE=SPFILE;
```

- Or individually per PDB

```
ALTER SESSION SET CONTAINER=pdbsec;
ALTER SYSTEM SET pdb_os_credential=PDBSEC_OS_USER SCOPE=SPFILE;
```

Manage File Access



PATH_PREFIX clause to restrict directory object paths

- Set per PDB at creation time. Can not be changed later.
- Restrictions and limitations when **PATH_PREFIX** is used:
 - Existing directory objects might not work as expected.
 - **PATH_PREFIX** string is always added as a prefix to all local directory objects in the PDB.
 - It does not apply to Oracle-supplied directory objects.
 - Does not affect files created by Oracle Managed Files.
 - Does not apply to data files or temporary files. Use **CREATE_FILE_DEST**

CREATE_FILE_DEST clause to restrict directory path for data files or temporary files.

- Implicit set DB_CREATE_FILE_DEST. => can be modified / requires lockdown profile
- Enable Oracle Managed Files for the PDB.
- Specifies the default file system directory or Oracle ASM disk group for PDB files.

Configure File Access

- Manually create a PDB based on PDB\$SEED

```
CREATE PLUGGABLE DATABASE pdbsec
  ADMIN USER pdbadmin IDENTIFIED BY LAB01schulung ROLES=(dba)
  PATH_PREFIX = '/u01/oradata/pdbsec/directories/'
  CREATE_FILE_DEST = '/u01/oradata/pdbsec/';
```

- Alternatively create a PDB based on a template PDB, dbca etc.
- Check current setting of PATH_PREFIX in database_properties

```
ALTER SESSION SET CONTAINER=pdbsec;
SELECT * FROM database_properties WHERE property_name='PATH_PREFIX';
```

PROPERTY_NAME	PROPERTY_VALUE	DESCRIPTION
PATH_PREFIX	/u01/oradata/PDBSEC/directories	All paths for objects such as directories are relative to this

Verify File Access – PATH_PREFIX



- Create an Oracle directory object outside PATH_PREFIX

```
SQL> CREATE DIRECTORY wrong_prefix AS '/tmp/test';
CREATE DIRECTORY wrong_prefix AS '/tmp/test'
*
ERROR at line 1:
ORA-65254: invalid path specified for the directory
```

- Create an Oracle directory object inside PATH_PREFIX

```
SQL> CREATE DIRECTORY no_prefix AS 'no_prefix';

Directory created.

SQL> SELECT directory_name, directory_path FROM dba_directories
       2  WHERE origin_con_id=(SELECT con_id FROM v$pdb);

DIRECTORY_NAME  DIRECTORY_PATH
-----
NO_PREFIX      /u01/oradata/PDBSEC/directories/no_prefix
```


Verify File Access – CREATE_FILE_DEST



- Create an Oracle tablespace with an absolute path outside CREATE_FILE_DEST

```
SQL> CREATE TABLESPACE wrong_prefix DATAFILE '/tmp/wrong_prefix.dbf' SIZE 1M;  
CREATE TABLESPACE wrong_prefix DATAFILE '/tmp/wrong_prefix.dbf' SIZE 1M  
*  
ERROR at line 1:  
ORA-65250: invalid path specified for file - /tmp/wrong_prefix.dbf
```

- Create an Oracle tablespace with an absolute path within CREATE_FILE_DEST and with OMF

```
SQL> CREATE TABLESPACE right_prefix DATAFILE  
2  '/u01/oradata/PDBSEC/right_prefix.dbf' SIZE 1M;  
  
Tablespace created.  
  
SQL> CREATE TABLESPACE no_prefix datafile SIZE 1M;  
  
Tablespace created.
```

Lockdown Profiles



- Oracle introduced lockdown profiles with 12.1. However, they become interesting only in 12.2 or the better 18c/19c
- Lockdown profiles allows to restrict user operation in PDBs in a multitenant container database.
- It is possible to assign lockdown profiles to...
 - ... individual PDBs, if PDB_LOCKDOWN is set while connected to a particular PDB.
 - ... all PDBs in a CDB, if PDB_LOCKDOWN is set while connected to the CDB root.
 - ... application container, if PDB_LOCKDOWN is set while connected to an application root
- The `CREATE LOCKDOWN PROFILE` statement must be issued from CDB or application root.
- Usage of lockdown profiles includes 3 steps:
 - Create a lockdown profile using `CREATE LOCKDOWN PROFILE`
 - Enable / disable user operations using `ALTER LOCKDOWN PROFILE`
 - Enable the corresponding lockdown profile using `ALTER SYSTEM SET pdb_lockdown=`

Capabilities of Lockdown Profiles



- `ALTER LOCKDOWN PROFILE` statement allows to enable or disable the following functions :
 - `LOCKDOWN_OPTIONS` clause: User operations associated with certain database options.
 - `LOCKDOWN_FEATURES` clause: User operations associated with certain database features.
 - `LOCKDOWN_STATEMENTS` clause: The issuance of certain SQL statements.
- Function can either explicitly be disable or enable.
- Combination of ALL and EXCEPT is possible.
- Oracle 18c introduced a couple of enhancements for lockdown profiles:
 - Restrict user operation for ALL, LOCAL or COMMON user.
 - Create lockdown profiles based existing profiles as static copy or dynamic link.

LOCKDOWN_OPTIONS clause



- The following database options can be restricted:
 - DATABASE QUEUING – User operations associated with Oracle Database Advanced Queuing option
 - PARTITIONING – User operations associated with Oracle Partitioning option
- Restriction explicit or with exclusion:
 - Use ALL to specify all options.
 - Use ALL EXCEPT to specify all options except the specified options.
 - Default is ENABLE OPTION ALL.
- Enable all options except DATABASE QUEUING

```
ALTER LOCKDOWN PROFILE sec_default ENABLE OPTION  
ALL EXCEPT = ('DATABASE QUEUING');
```

LOCKDOWN_FEATURES clause



- *LOCKDOWN_FEATURES* clause disable or enable user operations associated with certain database features.
- Supports a comprehensive list of database features and feature bundle e.g. AWR_ACCESS, CONNECTIONS, JAVA, JAVA_RUNTIME, NETWORK_ACCESS, OS_ACCESS, etc.
 - See Oracle documentation [ALTER LOCKDOWN PROFILE](#) for a complete list.
- Restriction explicit or with exclusion:
 - Use ALL to specify all features.
 - Use ALL EXCEPT to specify all features except the specified feature.
 - Default is ENABLE ALL.
- Disable OS_ACCESS but explicitly enable TRACE_VIEW_ACCESS

```
ALTER LOCKDOWN PROFILE sec_default DISABLE FEATURE = ('OS_ACCESS');  
ALTER LOCKDOWN PROFILE sec_default ENABLE FEATURE = ('TRACE_VIEW_ACCESS');
```

LOCKDOWN_STATEMENTS clause



- *LOCKDOWN_STATEMENTS* clause disable or enable issuance of certain SQL statements.
 - Use `DISABLE` to disable the execution of specified SQL statements.
 - Use `ENABLE` to enable the execution of specified SQL statements.
- Restriction explicit or with exclusion:
 - Use `ALL` to specify all statements.
 - Use `ALL EXCEPT` to specify all statements except the specified statements.
 - Default is `ENABLE STATEMENT ALL`.
- The `STATEMENT_CLAUSES` lets disable or enable specific clauses of the specified SQL statement.
- The `OPTION_CLAUSES` lets disable or enable setting or modification of specific options.
 - E.g. disable all `ALTER SYSTEM` but allow `ALTER SYSTEM SET cursor_sharing`
- Challenging to cover all critical statements, clause and options and not open a security flaw.

A few more examples

- Disable ALTER SYSTEM but allow it for COMMON user and for clause KILL SESSION.

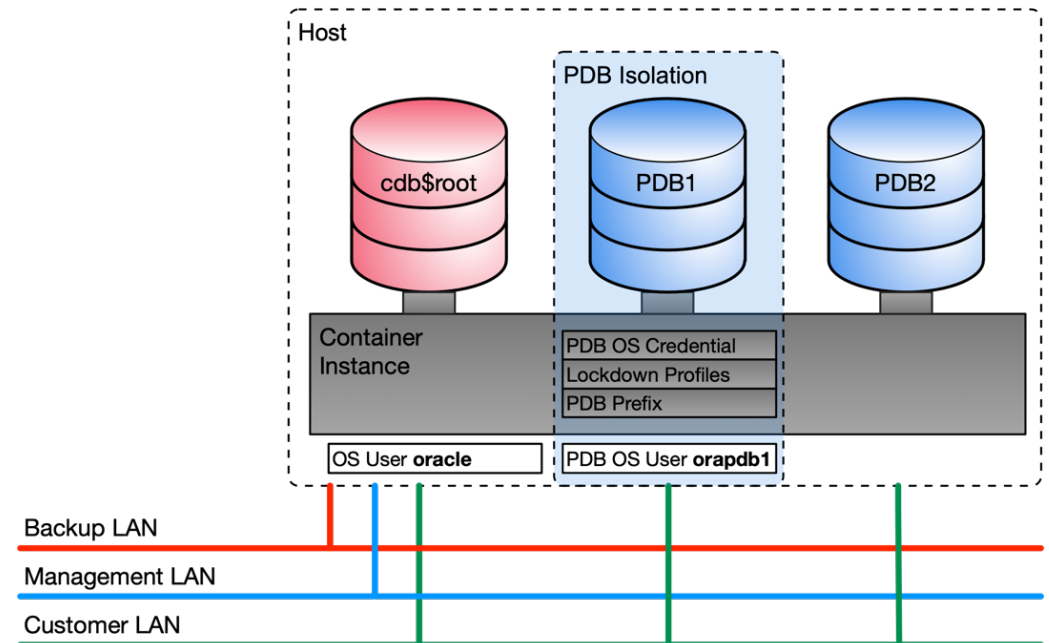
```
ALTER LOCKDOWN PROFILE sec_default DISABLE STATEMENT = ('ALTER SYSTEM');  
ALTER LOCKDOWN PROFILE sec_default ENABLE  
    STATEMENT = ('ALTER SYSTEM') CLAUSE = ('SET') USERS=COMMON;  
ALTER LOCKDOWN PROFILE sec_default ENABLE  
    STATEMENT = ('ALTER SYSTEM') CLAUSE = ('KILL SESSION');
```

- Create a new lockdown profile **sec_jvm** based on **sec_default**
- Enable Java and Java runtime

```
CREATE LOCKDOWN PROFILE sec_jvm FROM sec_default;  
ALTER LOCKDOWN PROFILE sec_jvm ENABLE FEATURE = ('JAVA_RUNTIME');  
ALTER LOCKDOWN PROFILE sec_jvm ENABLE FEATURE = ('JAVA');
```


Missing parts

- PDB isolation is no a replacement for a comprehensive security concept.
- Resource management is not covered at all.
- TDE, in particular TDE key management requires additional measures
- OS specific admin activities are not separated. e.g. Backup & Recover
- Although user are distinct by PDB_OS_CREDENTIAL, measures on the OS are still mandatory. => Layered Security
- Flaw in PDB isolation if poor lockdown profiles have been defined.



Outlook and potential enhancements

Outlook and potential enhancements

- In short term, waiting for the problems around PDB_OS_CREDENTIAL to be solved is still required.
 - [Bug 25820082](#) - PDB_OS_CREDENTIAL PARAMETER NOT WORKING
 - [Bug 29926986](#) - LISTENER SPAWNED EXTPROC FAILS WITH ORA-28575
 - [Bug 29922316](#) - EXTPROC FAILS WITH HS: CHILD EXTPROC FINISHED CALLING JSSU! STATUS = -1 WHEN DBMS_CREDENTIAL CONFIGURED
 - Doc 29938722 - PDB_OS_CREDENTIAL DOCUMENTATION NEEDS TO INCLUDE CONFIGURATION REQUIREMENTS AND EXAMPLES
- Although the Java challenges are not yet covered by these bugs
- Oracle does struggle with similar challenges for their cloud services e.g. Oracle Autonomous Database
- Discussions with the Oracle PM for PDBs have shown that solutions are needed and being developed. e.g. Linux CGROUPS, Linux Security Module (LSM) etc.



Oracle DB Nest

- A possible hint in Oracle 19c based on a few hidden parameter

Parameter	Instance	Description
-----	-----	-----
_dbnest_enable	NONE	dbNest enable
_dbnest_pdb_fs_conf		PDB Filesystem configuration
_dbnest_pdb_fs_type	DEFAULT	PDB FS Type
_dbnest_pdb_scm_conf		PDB SCM configuration
_dbnest_pdb_scm_level	STRICT1	PDB SCM Level
_dbnest_stage_dir		Staging directory configuration
_instance_dbnest_name		Instance dbNest Name

- Reveals a functionality named DB Nest.
- Presentations from OOW 2018 have mentioned DB Nest as a security enhancement for Multitenant Databases

Oracle DB Nest

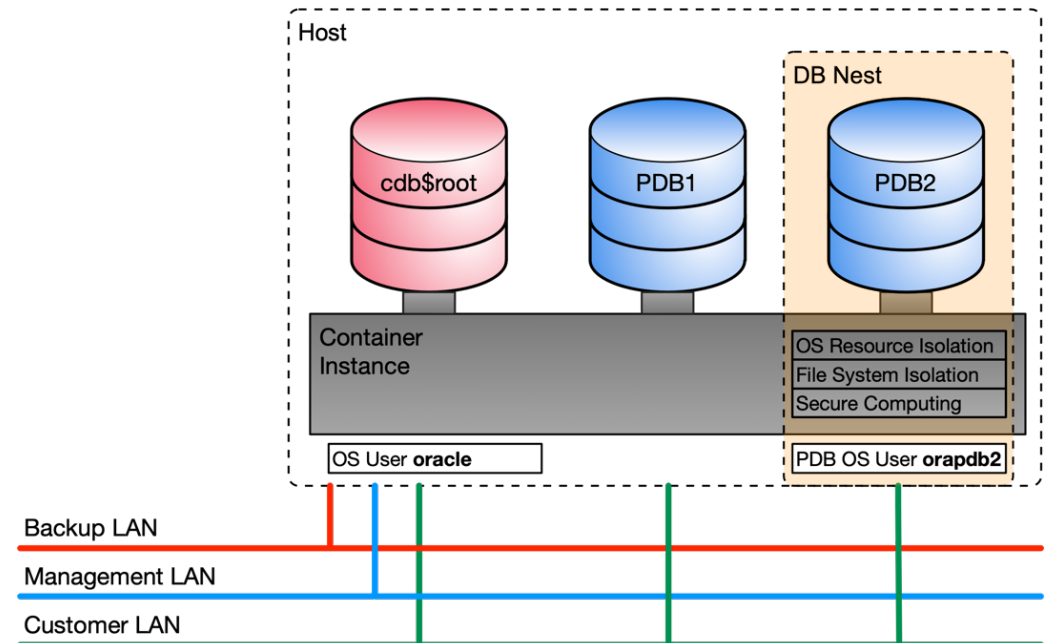


The exact functionality can only be guessed at the moment.

Control and isolation of...

- OS resources used by a PDB
- File system isolation per PDB
- Secure computing
- ...?

Oracle Open World 2019 is soon 😊



Summary

Summary



- It is not exclusively a customer challenge. Oracle has to solve similar issues for its cloud products.
- Lockdown profiles, PATH_PREFIX, PDB_OS_CREDENTIAL etc. do provide basic functionality to implement PDB Isolation. But...
 - ... there are still a few bugs.
 - ... not all use cases are covered e.g. resource management, java etc.
- What used to be true is still true.
 - **Do not install** Oracle JVM in security-critical databases!
 - If it is not absolutely necessary.
- The opening of an SR on this topic is simply cumbersome and laborious.
- The new functionality DB Nest does look promising.

Despite the limitations, these functions offer basic functionalities to make the PDB more secure and to start implementing isolation.



Making a **WORLD** possible
in which **intelligent IT**
facilitates **LIFE and WORK** as a
matter of course.