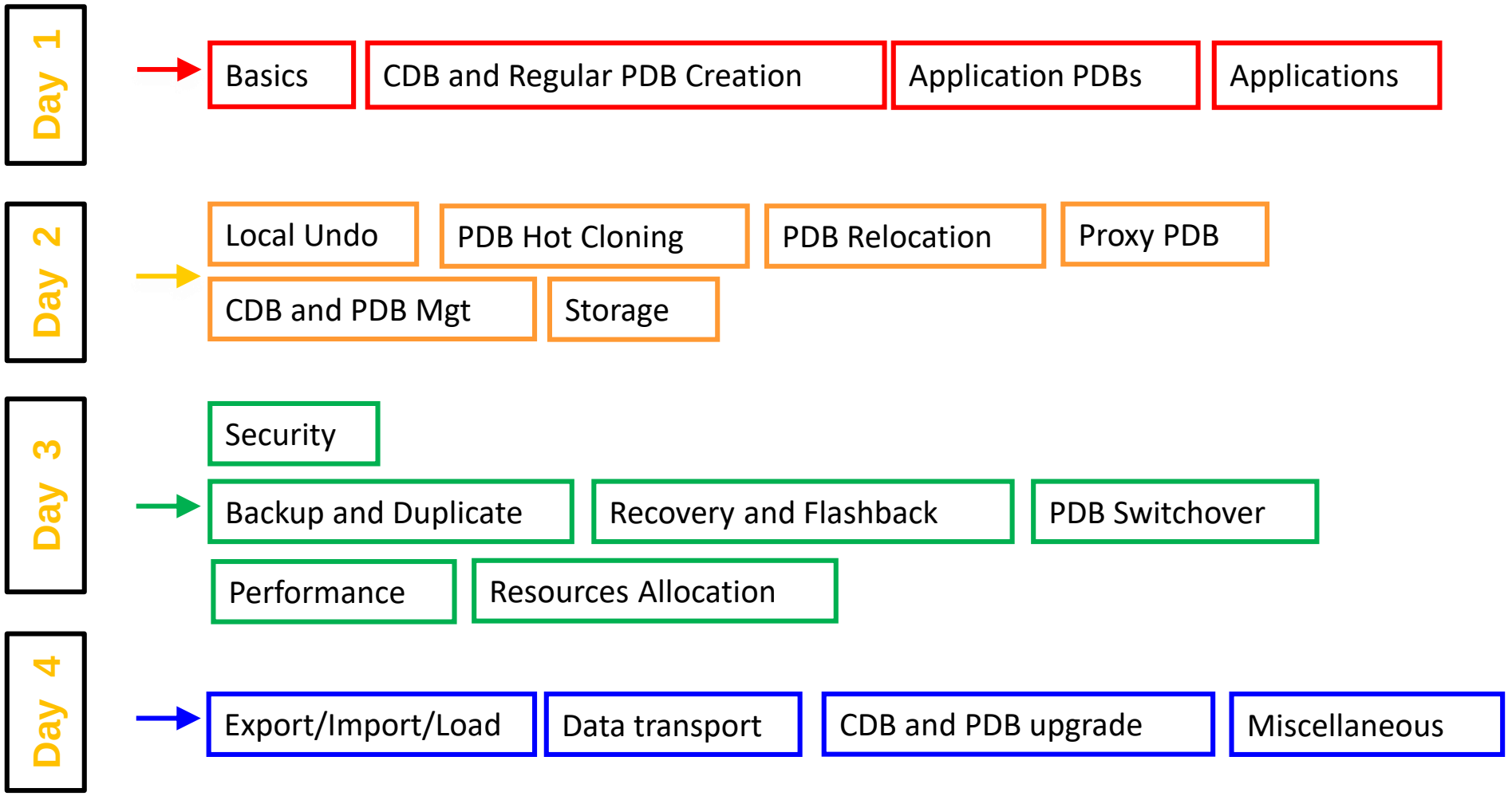
CDB Basics

Oracle Database 19c Multitenant Architecture



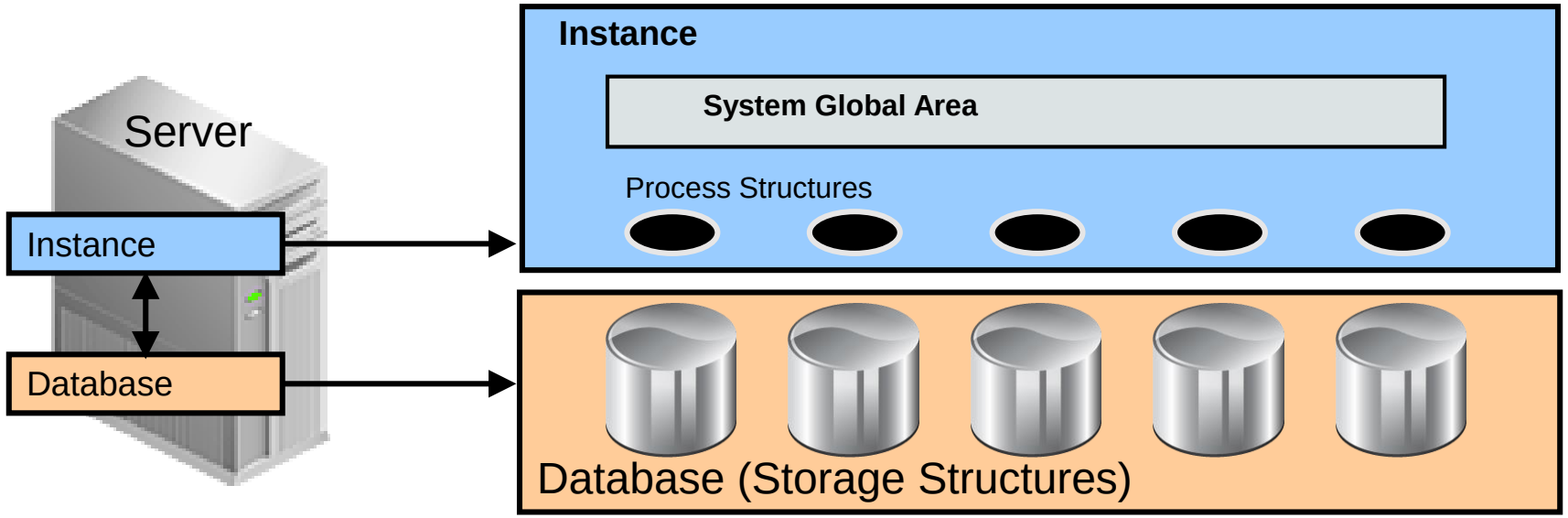
Objectives

- After completing this lesson, you should be able to:
  - Describe the multitenant architecture
  - Describe the CDB root and pluggable database containers
  - Differentiate the CDB root from a pluggable database
  - Explain the terminology of commonality
  - List impacts in various areas

Challenges

- Many Oracle customers have large numbers of “departmental” applications built on Oracle RDBMS that:
  - Do NOT use a significant percentage of the hardware on which they are deployed
  - Have instance and storage overhead preventing large numbers of “departmental” databases from being placed on the same physical and storage server
  - Are NOT sufficiently complex to require 100 percent of the attention of a full-time administrator
  - Do require significant time to patch or upgrade all applications

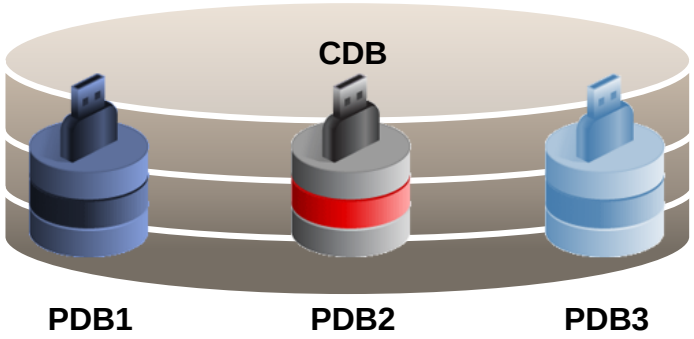
Non-CDB Architecture



- Multiple non-CDBs share nothing:
  - Too many background processes
  - High shared/process memory
  - Many copies of Oracle metadata

Multitenant Architecture: Benefits

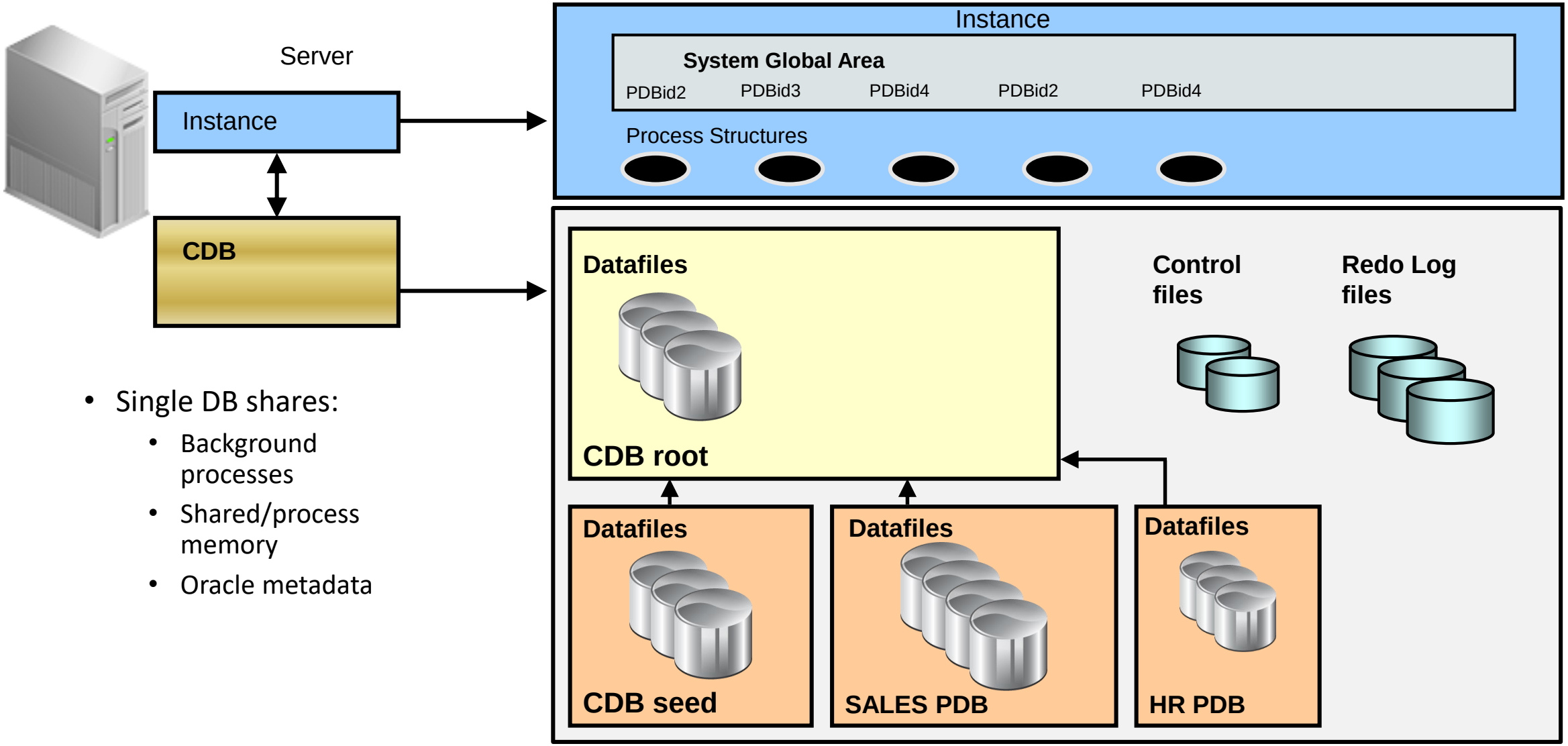
- Operates **multiple databases in a centrally managed platform** to lower costs:
  - Less instance overhead
  - Less storage cost
- Reduces DBA resources costs and maintains security
  - No application changes
  - **Fast and easy provisioning**
    - **Time saving for patching and upgrade**
    - **Separation of duties** between:
      - Different application administrators
      - Application administrators and DBA
      - Users within application
- **Provides isolation**



Other Benefits of Multitenant Architecture

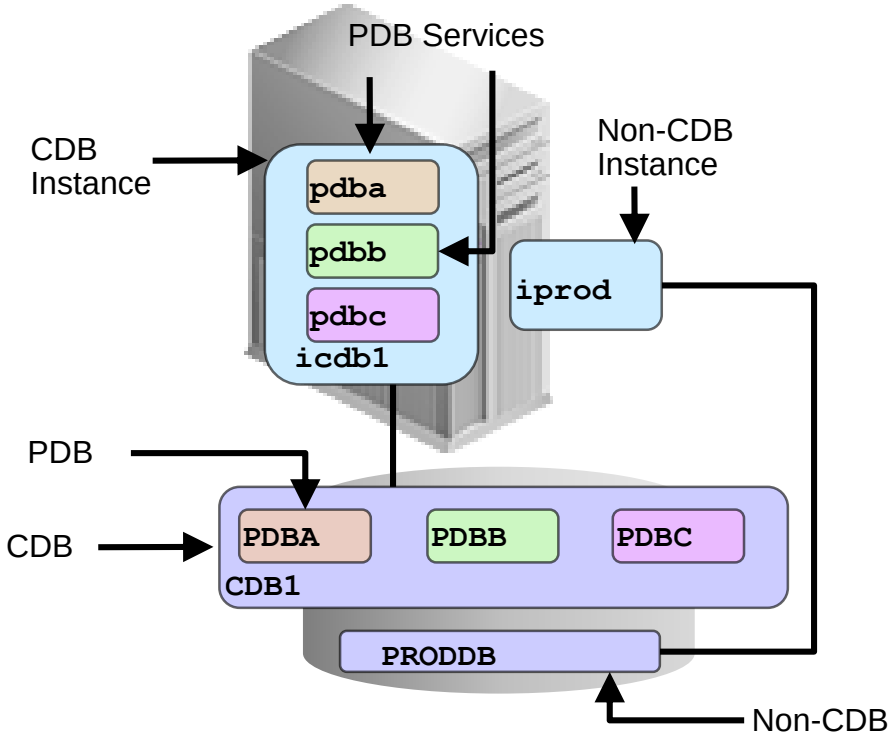
- Ensures **full backward-compatibility** with non-CDBs
- Fully operates with Oracle Real Application Cluster (Oracle RAC) and Data Guard
- Is supported by Oracle Enterprise Manager
- Is integrated with Resource Manager
- Allows central management and administration of multiple databases
  - Backups or disaster recovery
  - Patching and upgrades

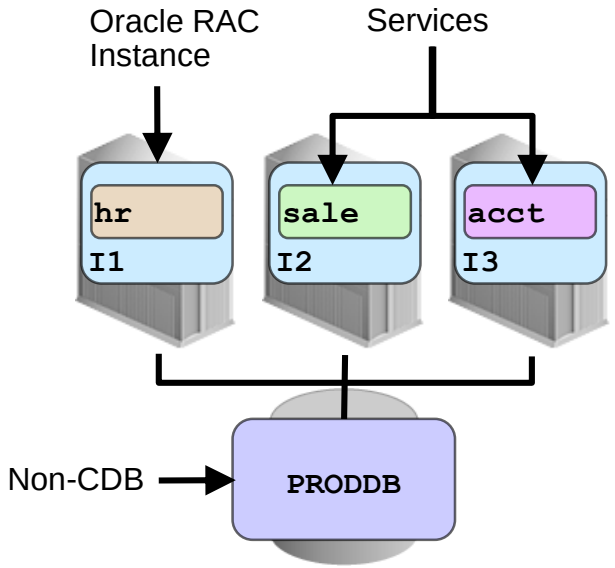
Oracle Multitenant Container Database

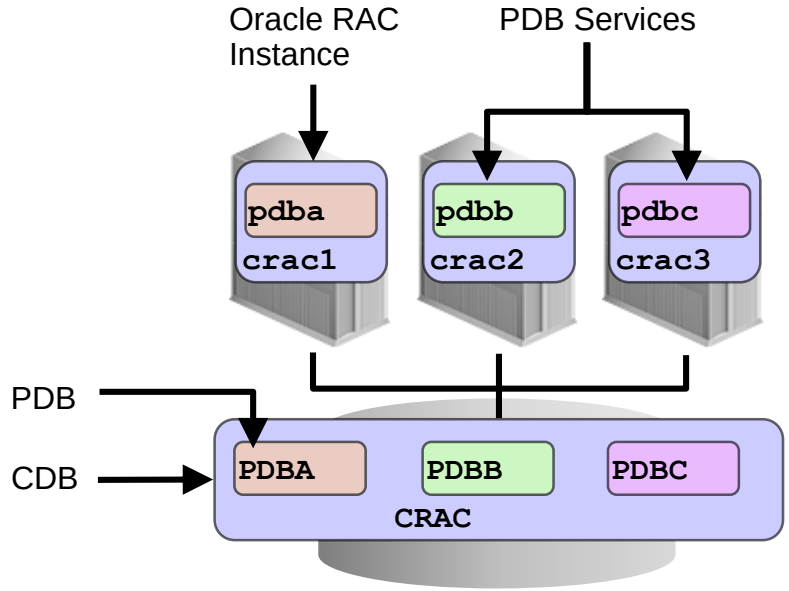


- Single DB shares:
  - Background processes
  - Shared/process memory
  - Oracle metadata

Configurations

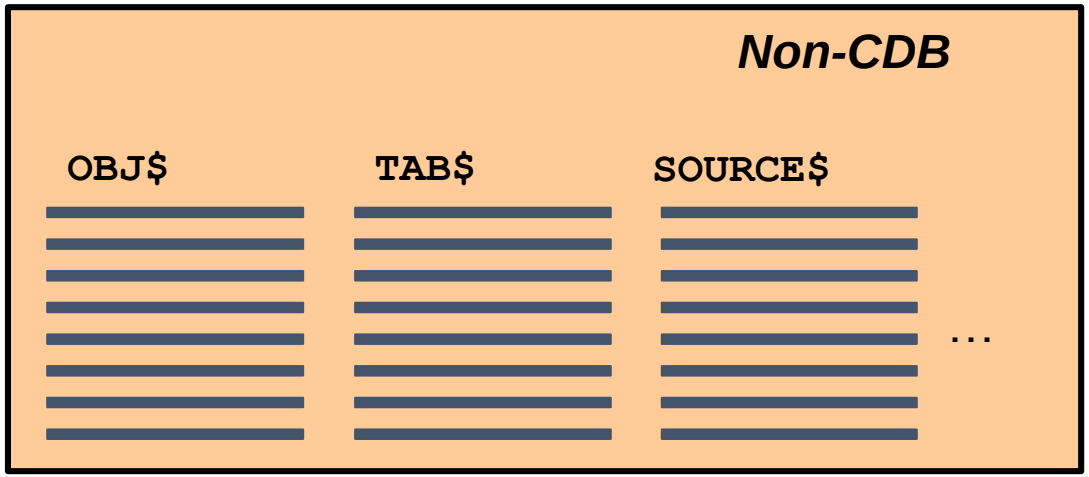






Database Objects in a non-CDB

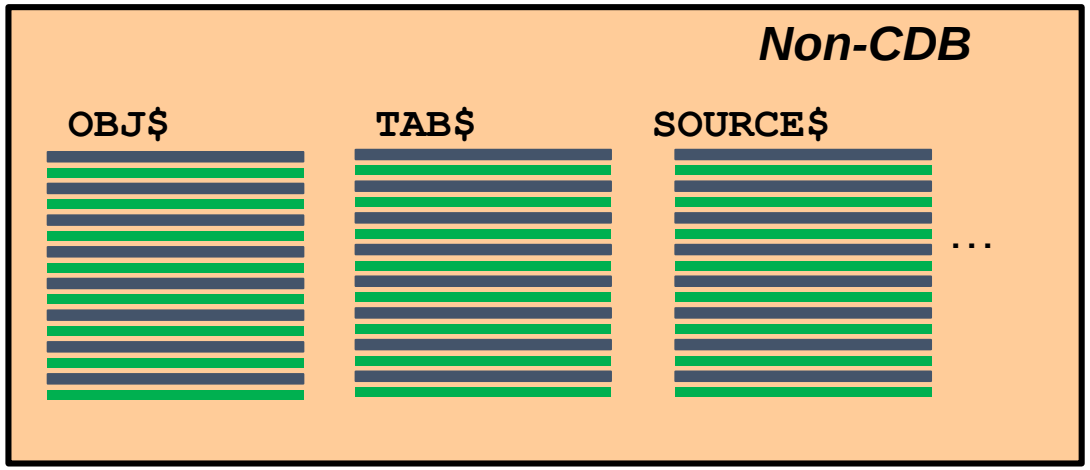
- After the initial database creation, the only objects are Oracle-supplied objects.



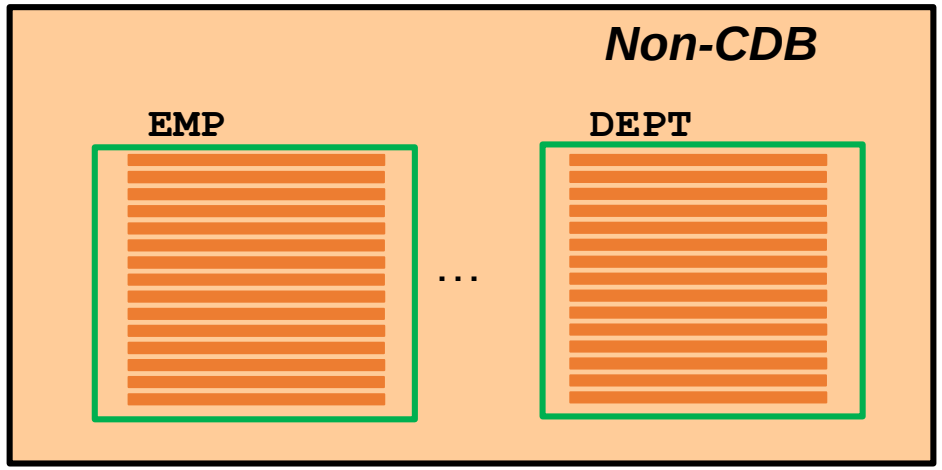
Oracle System data

User-Added Objects to a non-CDB

- In a non-CDB, user data is added: The metadata is mixed with the Oracle-supplied data in the data dictionary.

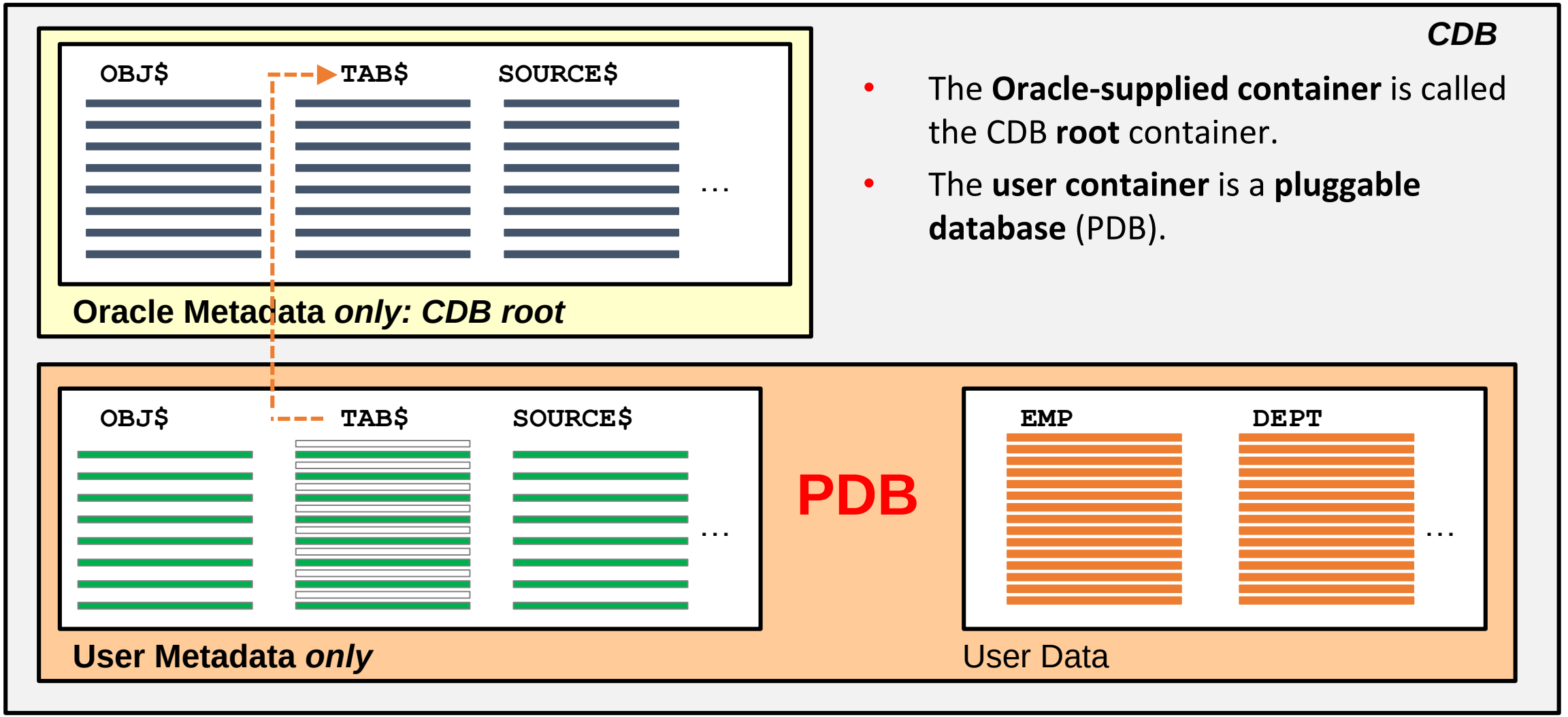


Oracle System data mixed with User metadata



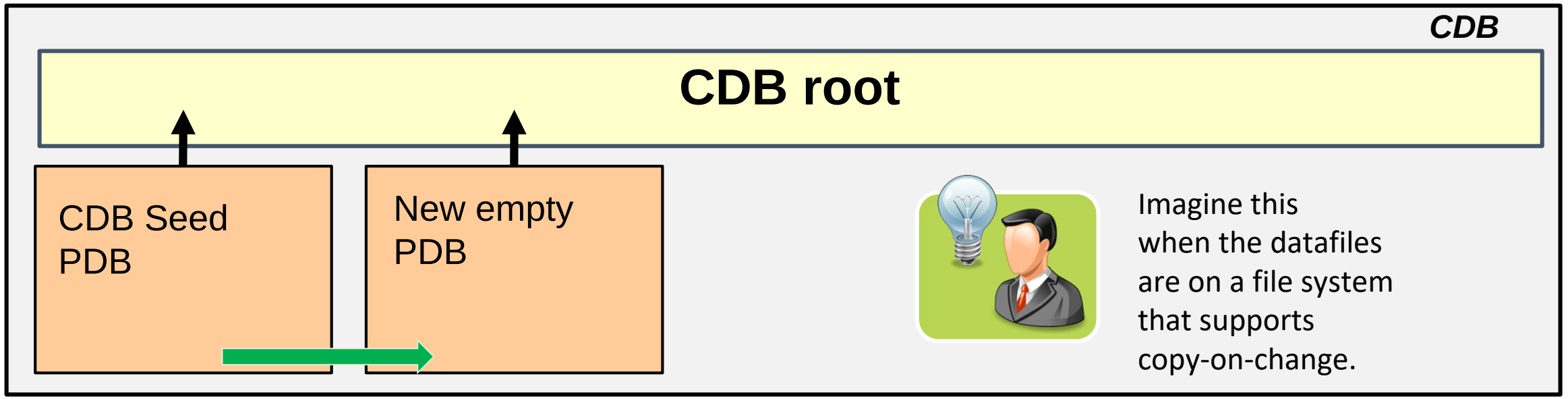
User Data

SYSTEM Objects in the USER Container



- The **Oracle-supplied container** is called the CDB **root** container.
- The **user container** is a **pluggable database** (PDB).

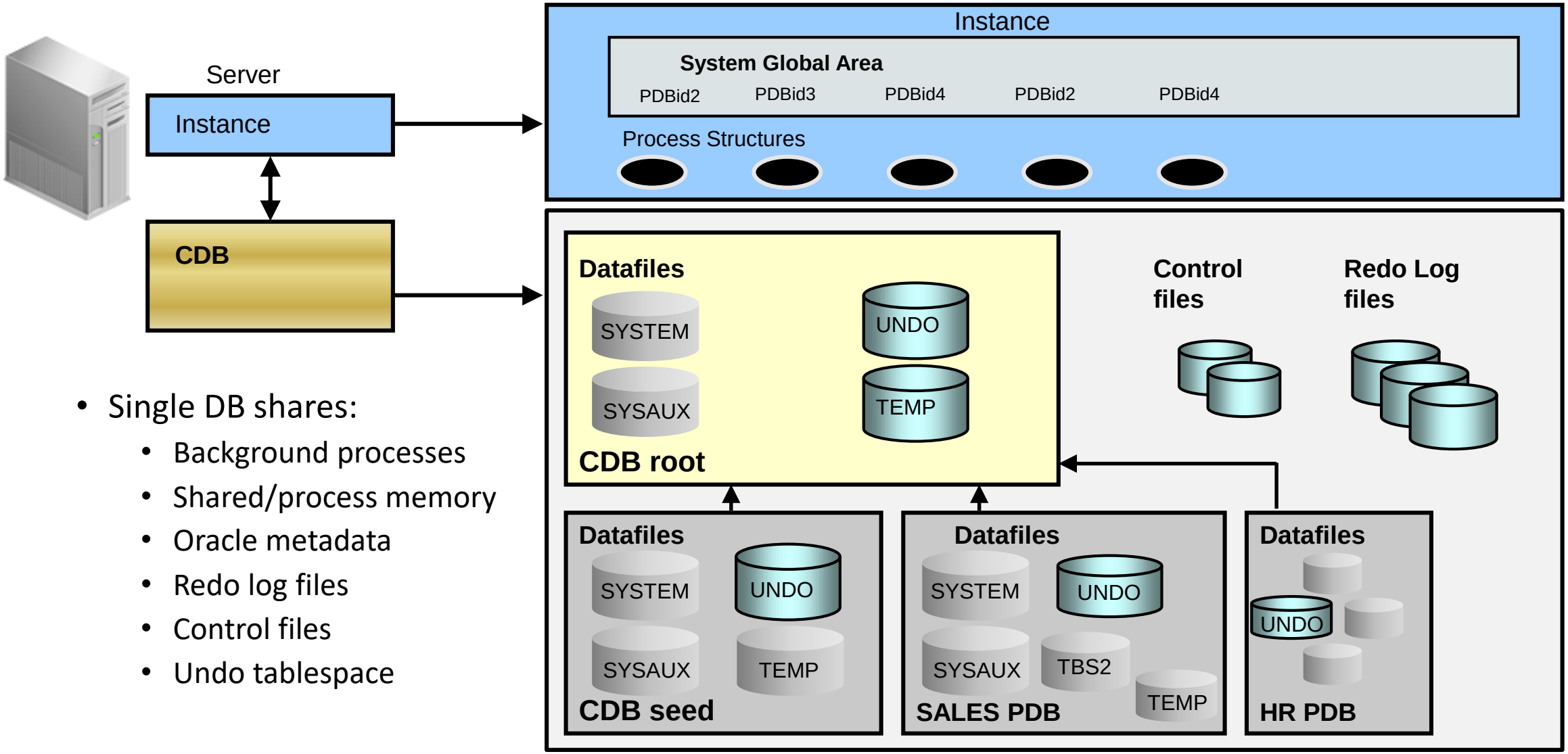
Provisioning a Pluggable Database



Different methods:

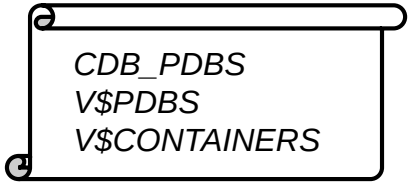
- **Create** new PDB from **CDB** seed pluggable database.
- **Plug** or **clone** a non-CDB as a PDB into a CDB.
- **Clone** or **relocate** a PDB from another PDB into the same or another CDB.
- **Plug an unplugged PDB** into the same or another CDB.
- **Create** a PDB as a **proxy** PDB.

Multitenant Container Database Architecture



- Single DB shares:
  - Background processes
  - Shared/process memory
  - Oracle metadata
  - Redo log files
  - Control files
  - Undo tablespace

Containers

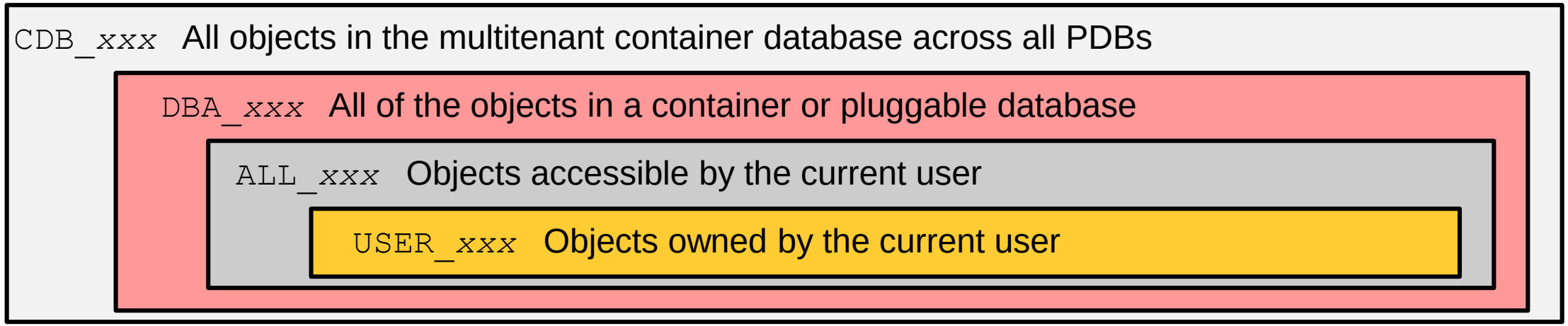


- Types of containers in `V$CONTAINERS`:
  - **CDB root**
    - The first mandatory container created at CDB creation
    - Oracle system–supplied common objects and metadata
    - Oracle system–supplied common users and roles
  - **Pluggable database** (PDBs)
    - Tablespaces (permanent and temporary)
    - Schemas / Objects / Privileges
    - Created / cloned / unplugged / plugged / proxied
    - Particular PDB: CDB seed (`PDB$SEED`) used for fast provisioning of a new PDB

Tools

	SQL*Plus	OUI	DBCA	EM Cloud Control	EM Database Express	SQL Developer	DBUA
Create a new CDB or PDB	Yes	Yes	Yes	Yes (PDB only)	Yes (PDB only)	Yes (PDB only)	
Explore CDB instance, architecture, and PDBs	Yes			Yes	Yes	Yes	
Upgrade a 12c CDB to 19c CDB				Yes			Yes

Data Dictionary and Dynamic Views



- CDB_pdbs: All PDBS within CDB
- CDB_tablespaces: All tablespaces within CDB
- CDB_users: All users within CDB (common and local)

```
SQL> select OBJECT_ID, ORACLE_USERNAME, LOCKED_MODE, CON_ID from V$LOCKED_OBJECT;

OBJECT_ID  ORACLE_USERNAME  LOCKED_MODE  CON_ID
---------- --------------- ----------- -------
     83711 SYS                       3       3 <-- PDB1
     83710 DOM                       3       4 <-- PDB2
```

Terminology

- DBA, CDB_DBA, and PDB_DBA (roles assigned to administrators at different levels)
- Common vs Local:
  - Users
  - Privileges / Roles
  - Objects
  - Profiles
- CDB vs PDB level:
  - CDB Resource Manager plan vs PDB RM plan
  - Unified audit policies at CDB or PDB level
  - Encryption master keys at CDB and PDB level
  - Database Vault realms and command rules at CDB or PDB level
  - XStream at CDB or PDB level

Impacts

- Define a character set for the CDB and per PDB.
- Define PDB initialization parameters in a single SPFILE.
- Do not use PDB-qualified database object names. Instead use database links.

```
SQL> SELECT * FROM HR.apps.tab1;        SQL> SELECT * FROM apps.tab1@HR;
```

- Implement subset standbys at the PDB level.
- Configure Oracle Database Vault per PDB and on common objects.
- Create one TDE master encryption key per PDB to encrypt PDB data.
- Configure unified audit at CDB and PDB level.
- Benefit from Heat Maps and Automatic Data Optimization.
- Use Logminer for objects at all levels.
- Configure replication at PDB and application level with XStream and Oracle GoldenGate.

Summary

- In this lesson, you should have learned how to:
  - Describe the multitenant architecture
  - Describe the CDB root and pluggable database containers
  - Differentiate the CDB root from a pluggable database
  - Explain the concept of commonality
  - List impacts in various areas

Practices Environment – 1

Each student has two virtual machines (VM1 and VM2) to use during class.

- **On VM1**:
  - Oracle Database 19c installed:
    - With one CDB named **`ORCL`** and one PDB named **`PDB1`**. The CDB has been set up for you in advance so that you can explore an existing configuration.
    - You will create another CDB, **`CDB18`**, and its PDB, **`PDB18`**.
  - Oracle Database 12.2 installed:
    - With one CDB named `CDB12` and one PDB named `PDB12` to perform upgrade operations to Oracle Database 19c.
    - With a non-CDB named `NONCDB` to perform plugging or cloning operations into one of the Oracle Database 19c CDBs. This shows you how to migrate a non-CDB into a CDB.
- **On VM2**:
  - Enterprise Manager 13*c* deployed to monitor Oracle Database 19c targets.
  - Oracle Database 12.1 installed with one CDB named `cdbem` and one PDB named `PDBEM` used as the Oracle Management Repository (OMR).

Practices Environment - 2

- Pre-created databases and instances with their respective PDBs:
  - Datafiles in `/u02/app/oracle/oradata/`
  - CDB root datafiles in `/u02/app/oracle/oradata/<db_name>`
  - PDB datafiles in `/u02/app/oracle/oradata/<db_name>/<pdb_name>`
  - Control files in `/u02/app/oracle/oradata/<db_name>` and `/u03/app/oracle/fast_recovery_area/<db_name>`
  - All redo log files in `/u04/app/oracle/redo/<db_name>`
  - All backup files in `/u03/app/oracle/fast_recovery_area/<db_name>`
  - Password and init files in `$ORACLE_HOME/dbs`
  - Diagnostics files in `/u01/app/oracle/diag/rdbms/orcl/ORCL/…`
  - TDE wallet in `/u01/app/oracle/admin/<db_name>/tde_wallet`
  - Net files in `$ORACLE_HOME/network/admin`

Practice 1: Overview

- 1-1: Discovering practices environment
- 1-2: Adding a CDB as a new target in Enterprise Manager Cloud Control
- 1-3: Checking named credentials
- 1-4: Using Enterprise Manager Express

- **Note:**
- In most of the practices, you will have to execute setup and cleanup shell scripts and SQL scripts. The scripts may generate false errors in the following occurrences:
  - A tablespace dropped because the tablespace was not created
  - A pluggable database dropped because the pluggable database was not created
  - A directory creation dropped because the directory already exists
    Do not pay attention to the errors.

Multitenant Architecture Poster