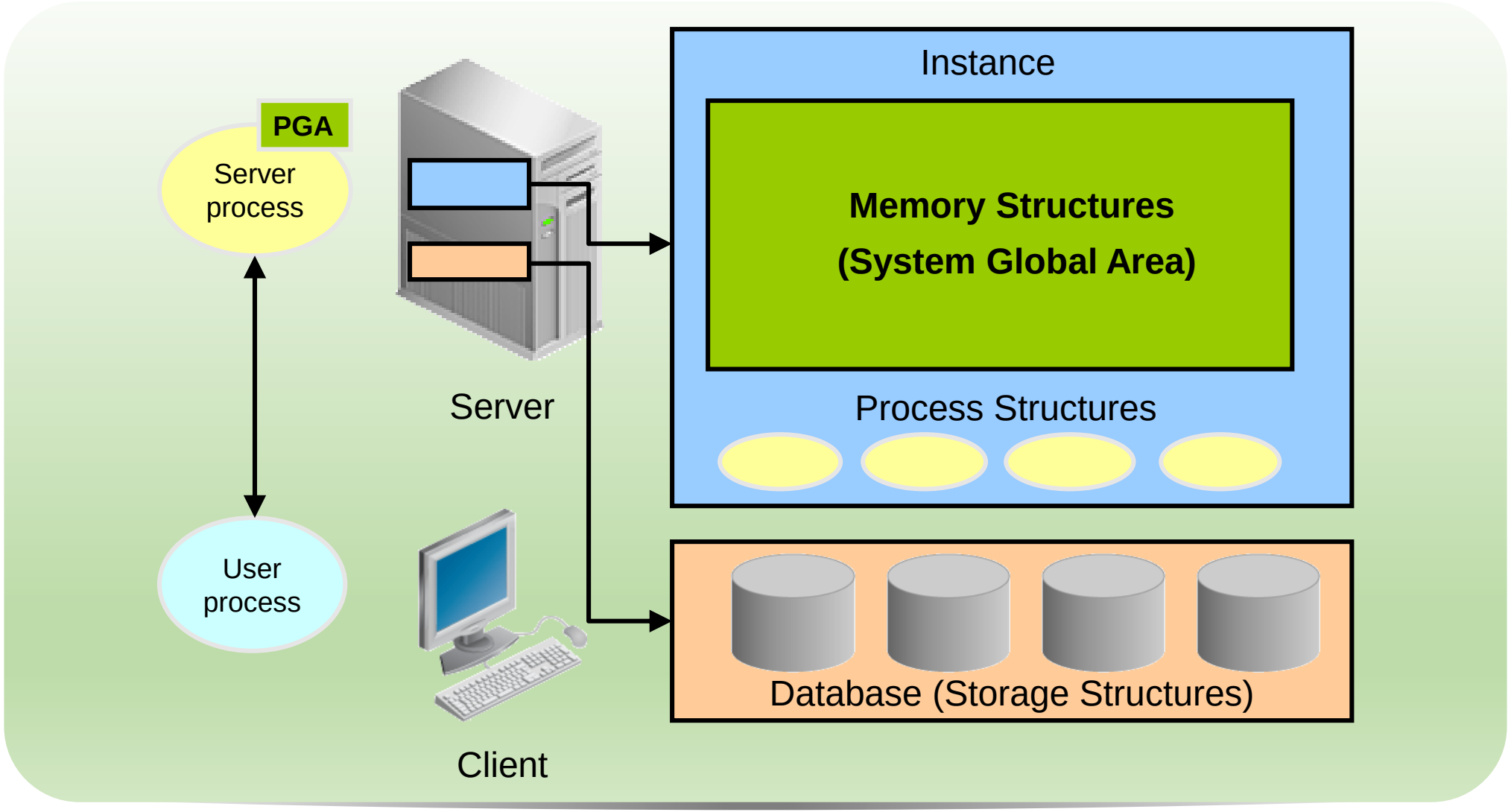
Oracle Database Architecture

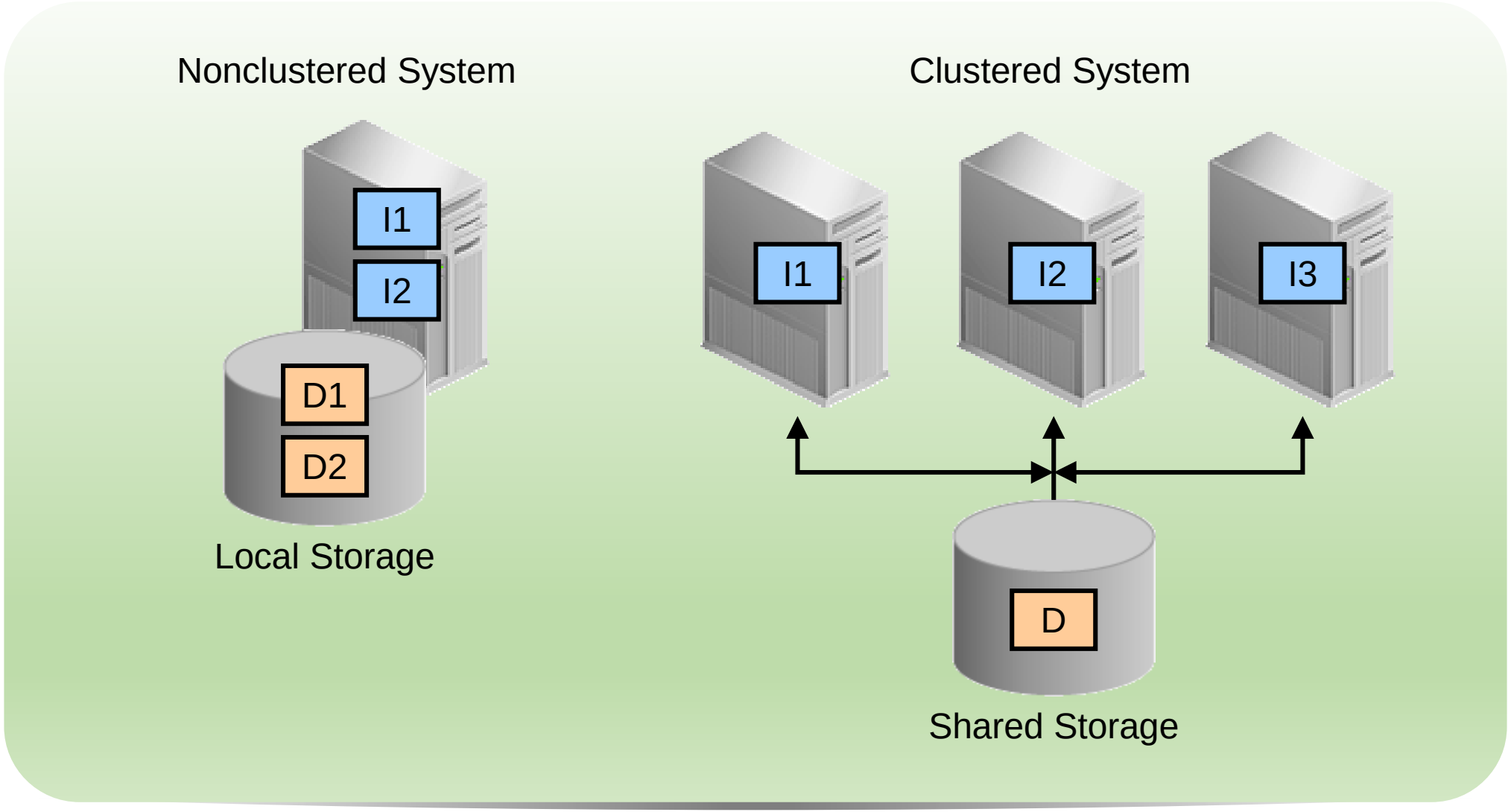
Objectives

- After completing this lesson, you should be able to:
  - List the major architectural components of Oracle Database
  - Explain memory structures
  - Describe background processes
  - Correlate logical and physical storage structures
  - Describe multitenant architecture

Oracle Database Server Architecture: Overview

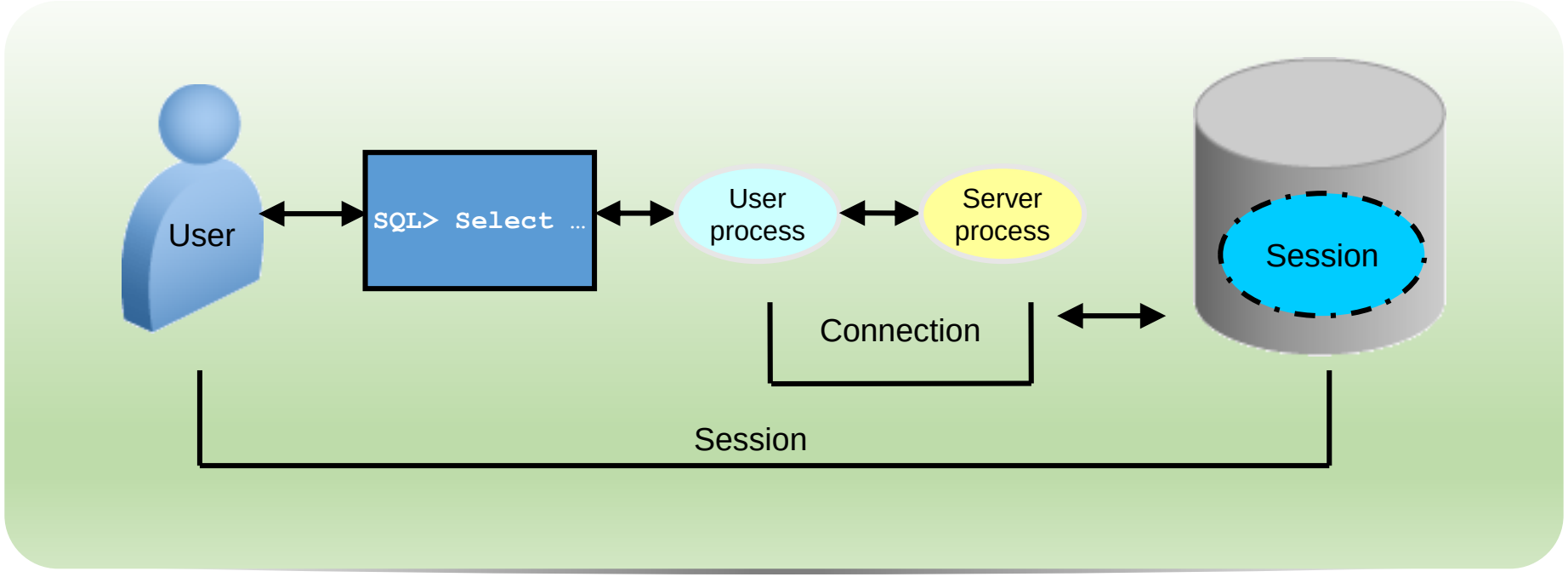


Oracle Database Instance Configurations

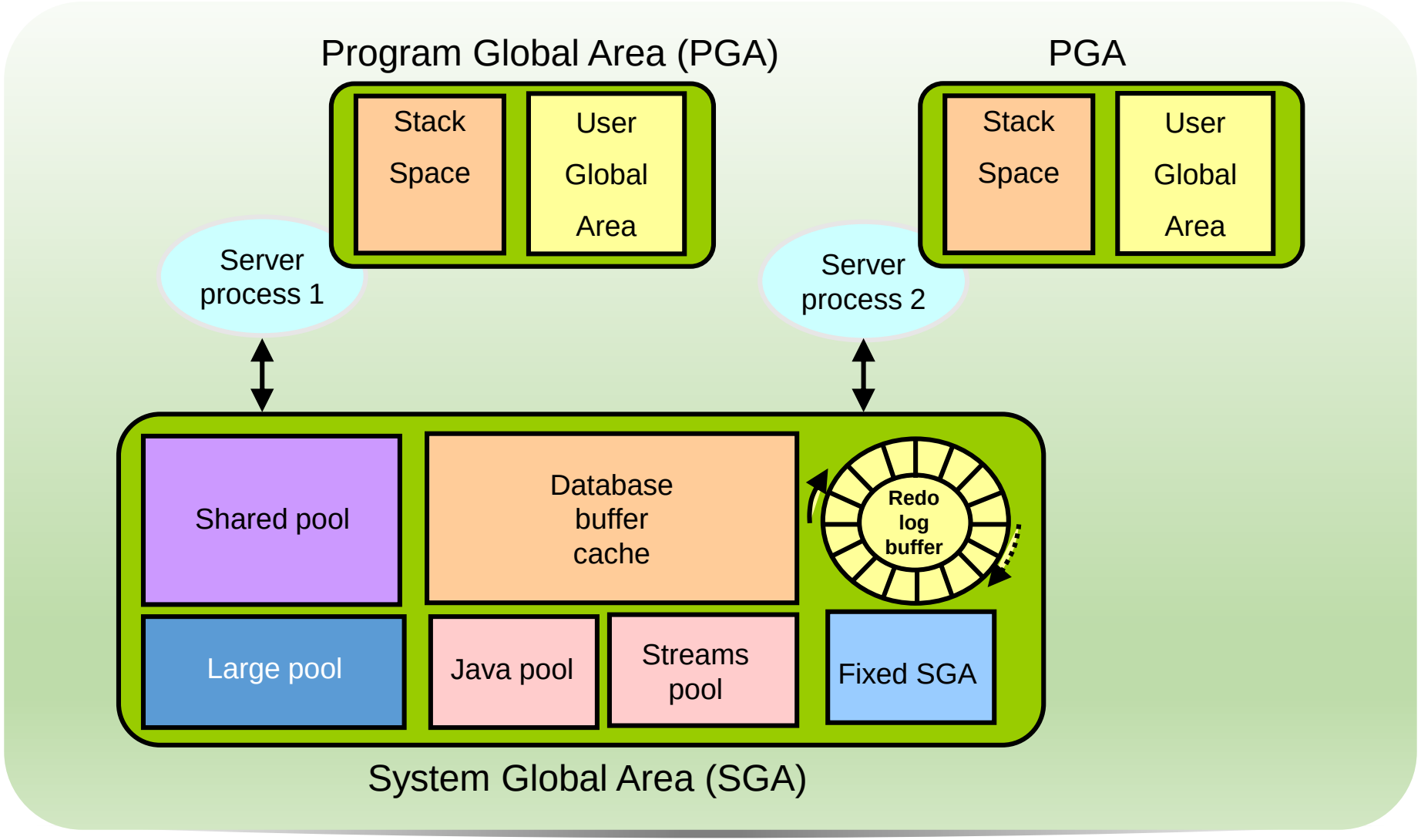


Connecting to the Database Instance

- Connection: Communication between a user process and an instance
- Session: Specific connection of a user to an instance through a user process

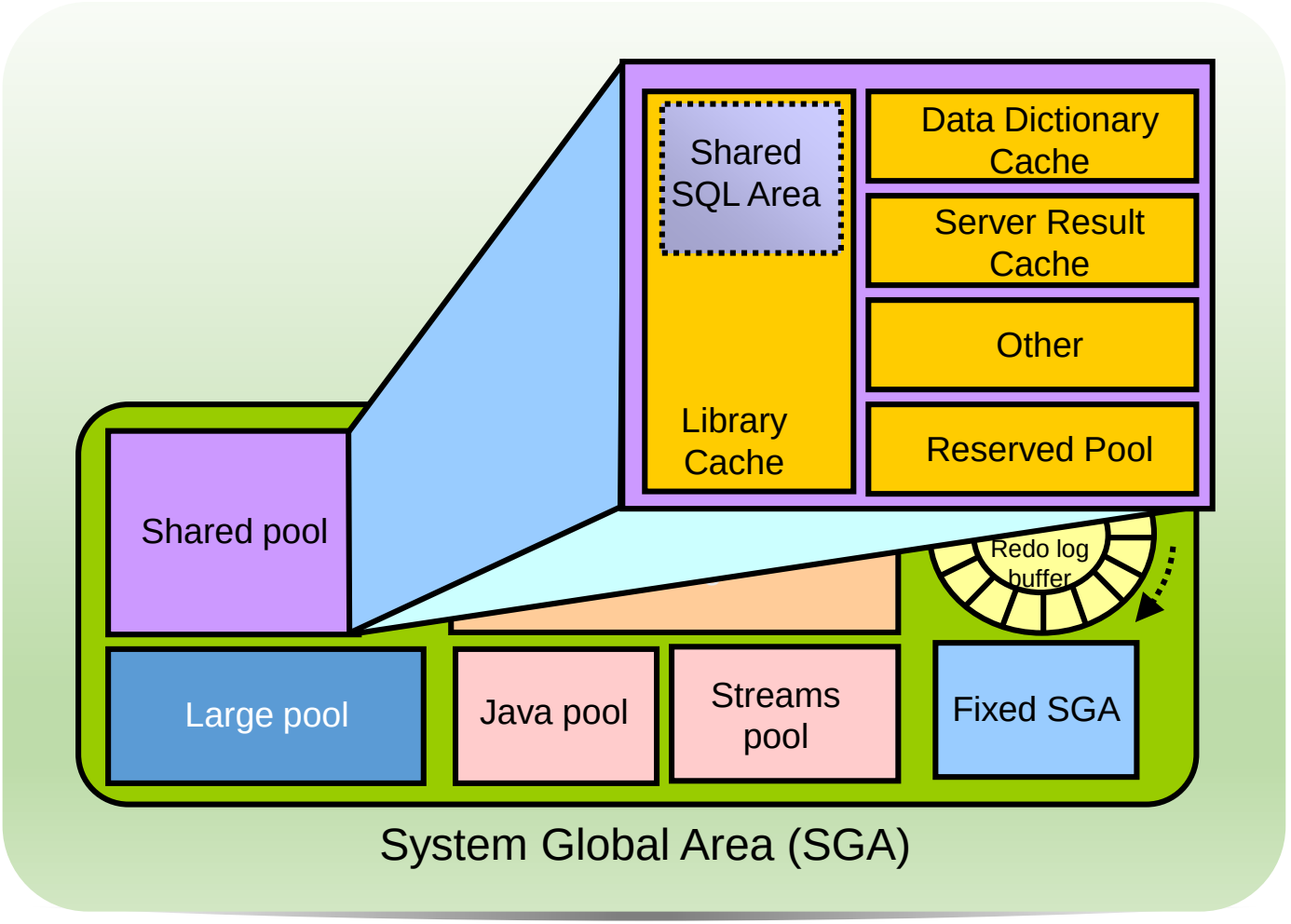


Oracle Database Memory Structures



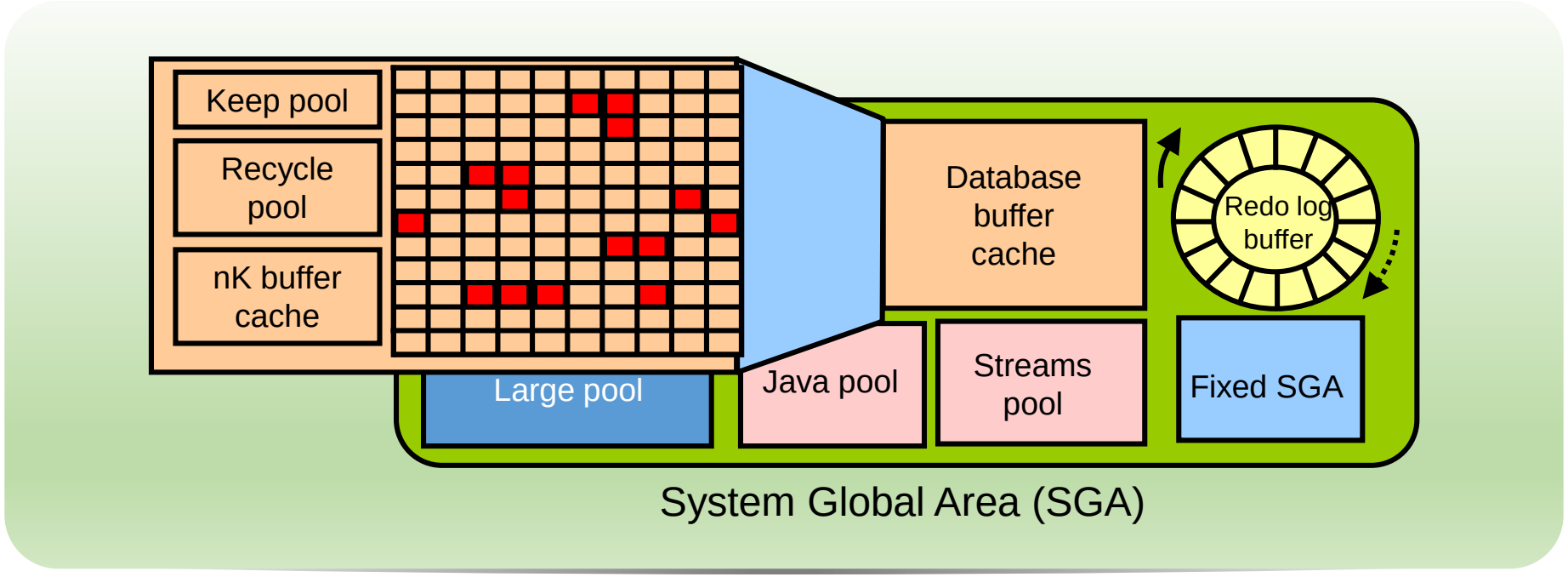
Shared Pool

- Is a portion of the SGA
- Contains:
  - Library cache
    - Shared SQL area
  - Data dictionary cache
  - Server result cache



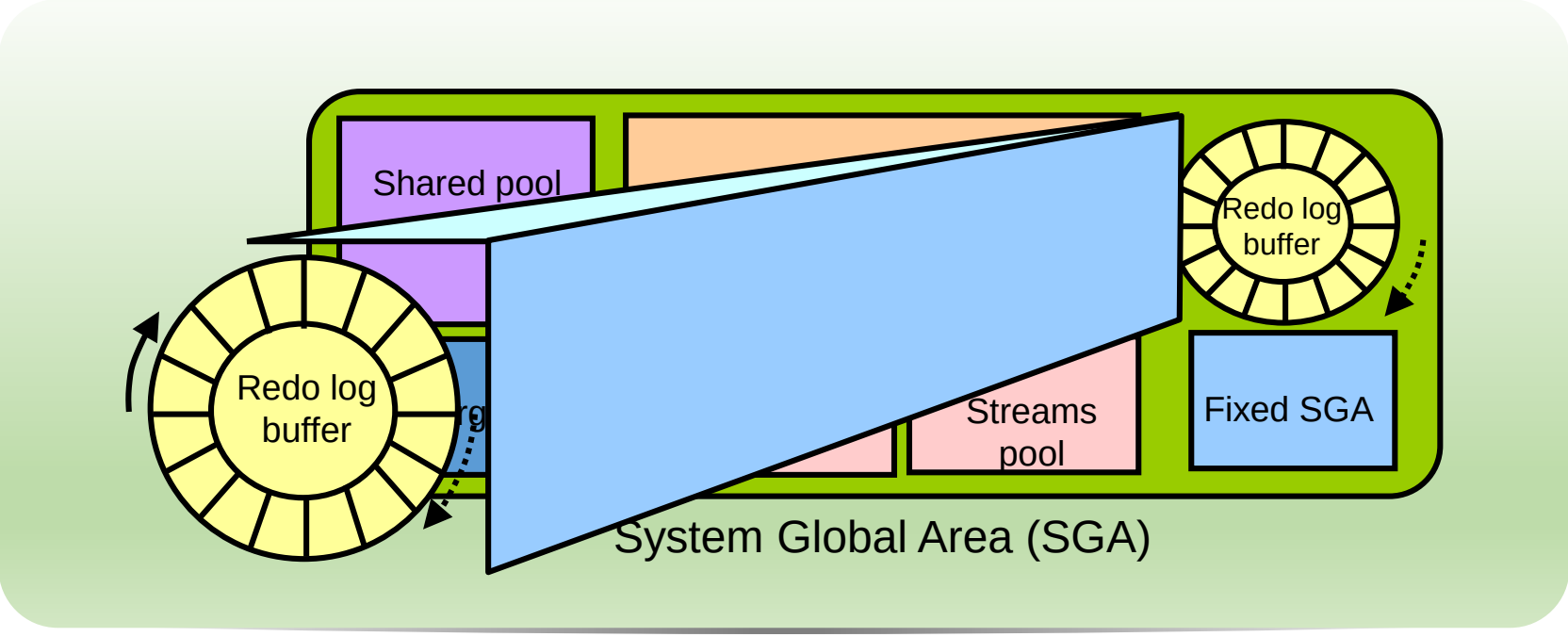
Database Buffer Cache

- Is part of the SGA
- Holds copies of data blocks that are read from data files
- Is shared by all concurrent users



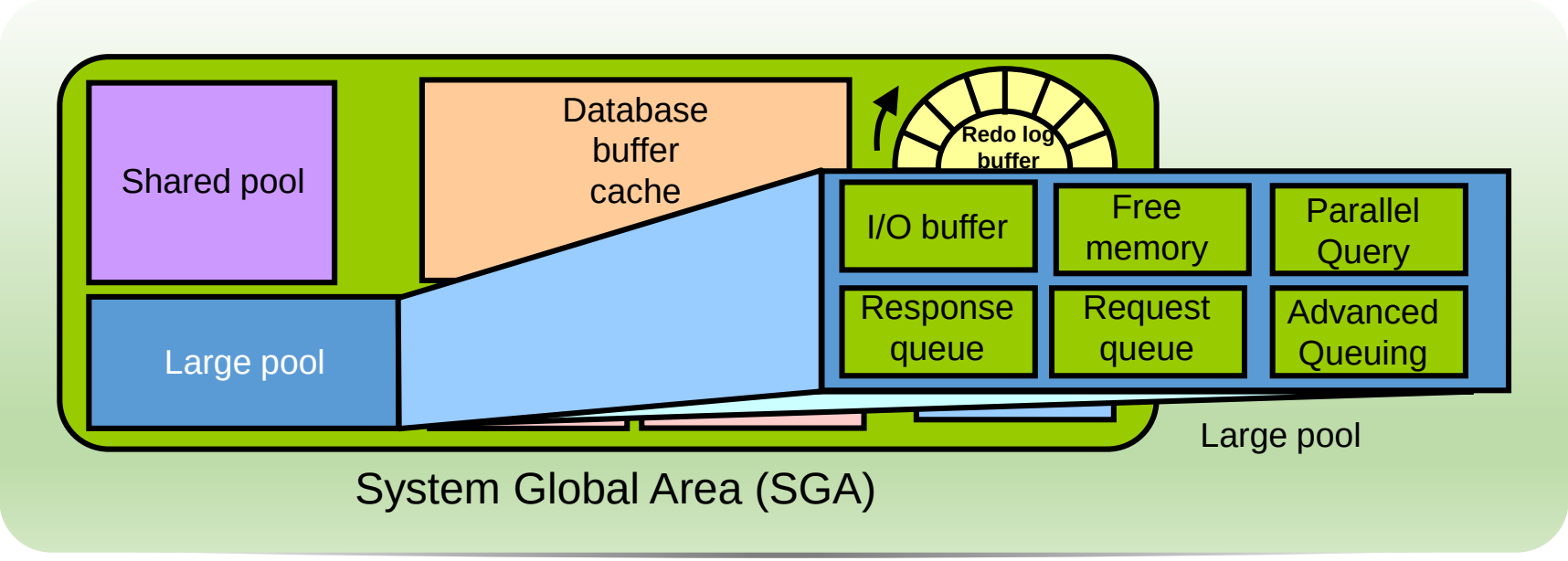
Redo Log Buffer

- Is a circular buffer in the SGA
- Holds information about changes made to the database
- Contains redo entries that have the information to redo changes made by operations such as DML and DDL



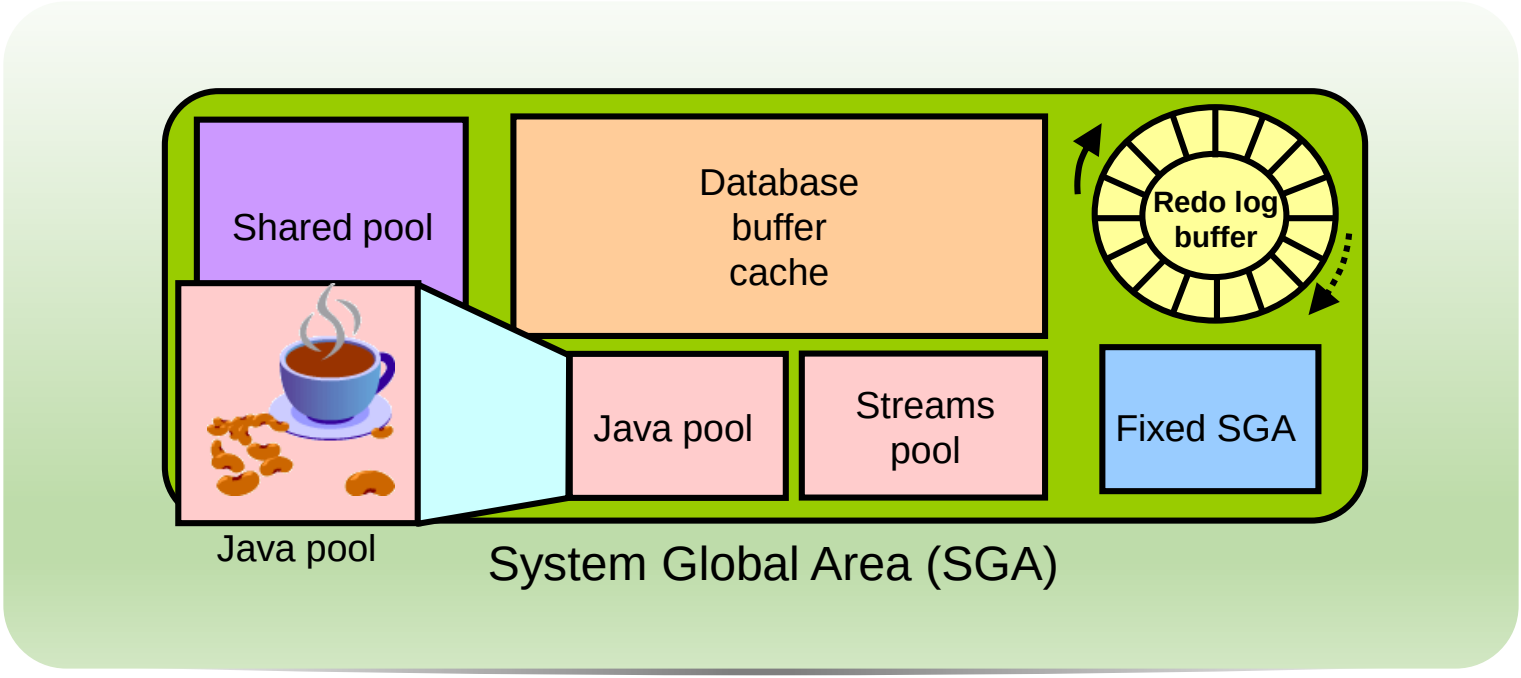
Large Pool

- Provides large memory allocations for:
  - Session memory for the shared server and the Oracle XA interface
  - I/O server processes
  - Oracle Database backup and restore operations



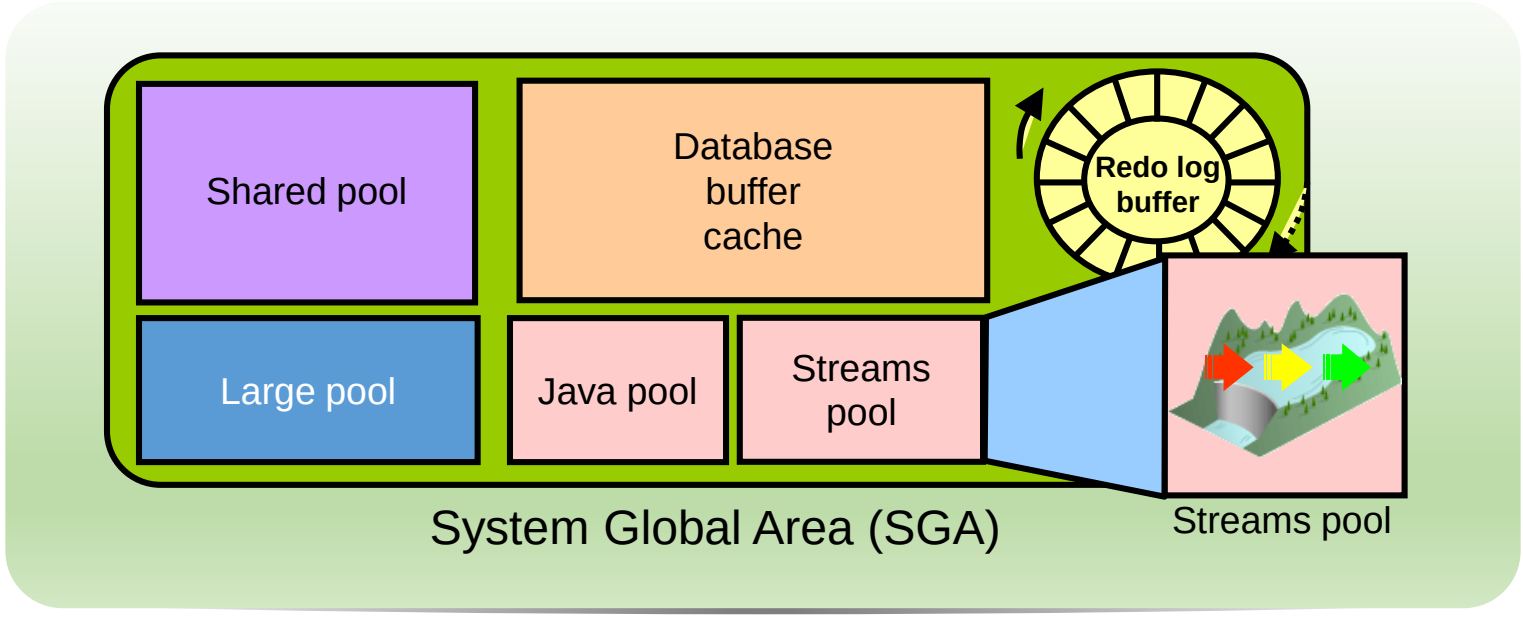
Java Pool

- Java pool memory is used to store all session-specific Java code and data in the JVM.

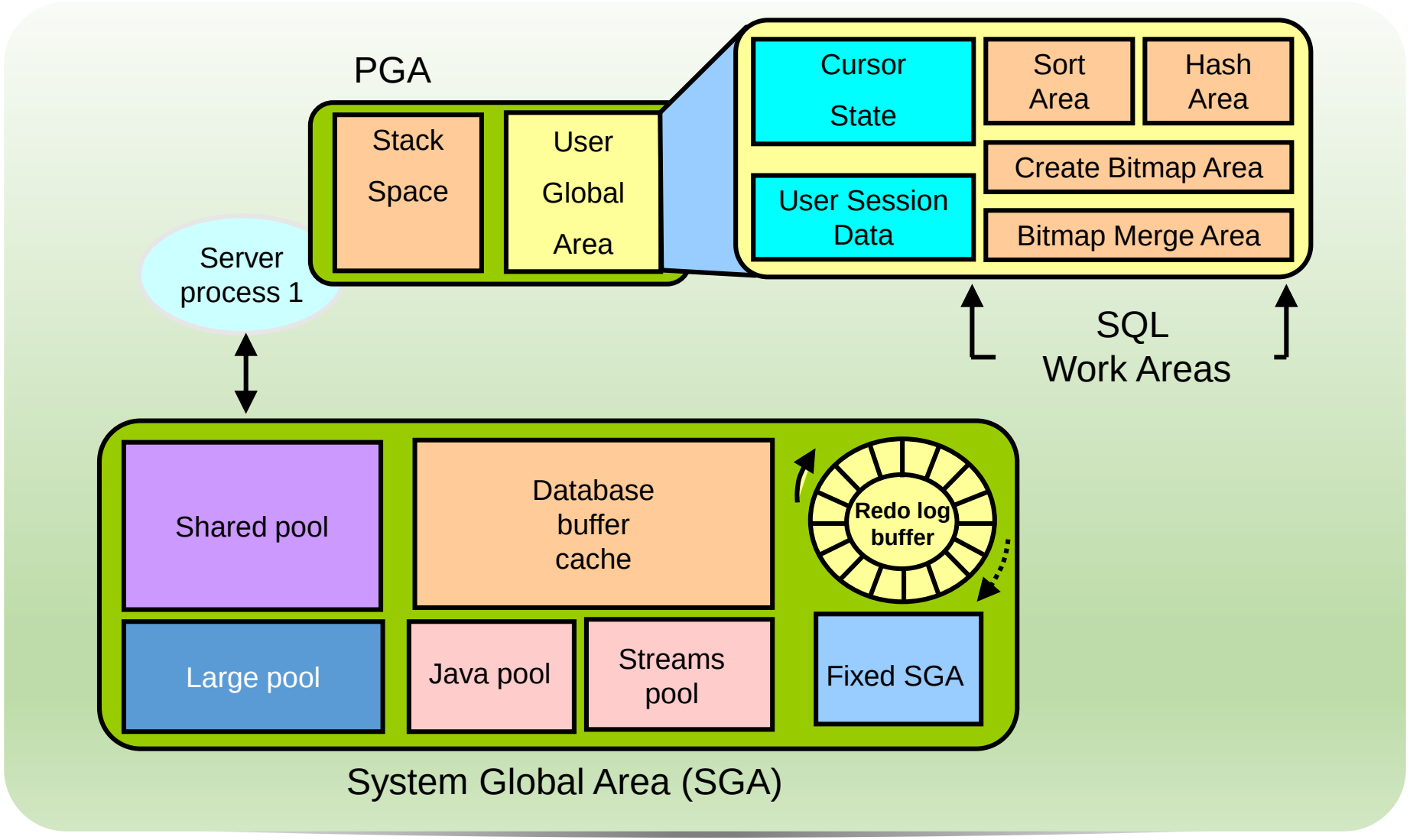


Streams Pool

- Streams pool memory is used exclusively by Oracle Streams to:
  - Store buffered queue messages
  - Provide memory for Oracle Streams processes

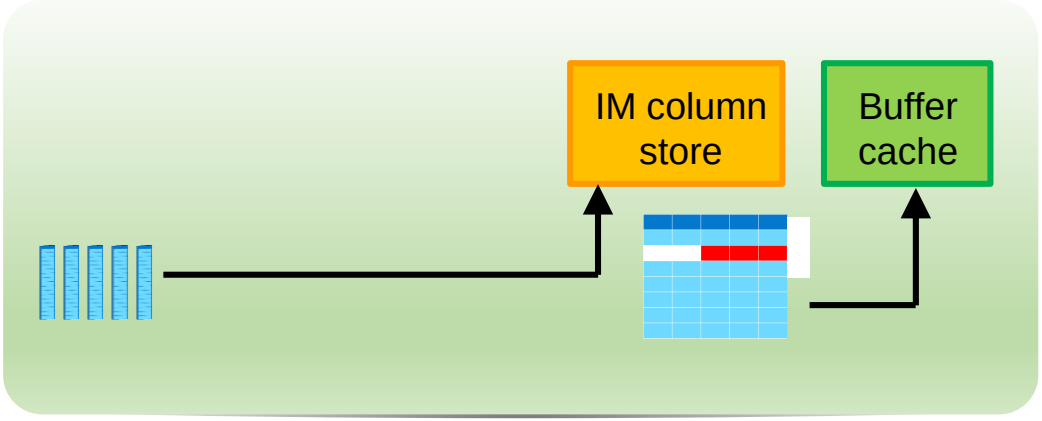


Program Global Area (PGA)



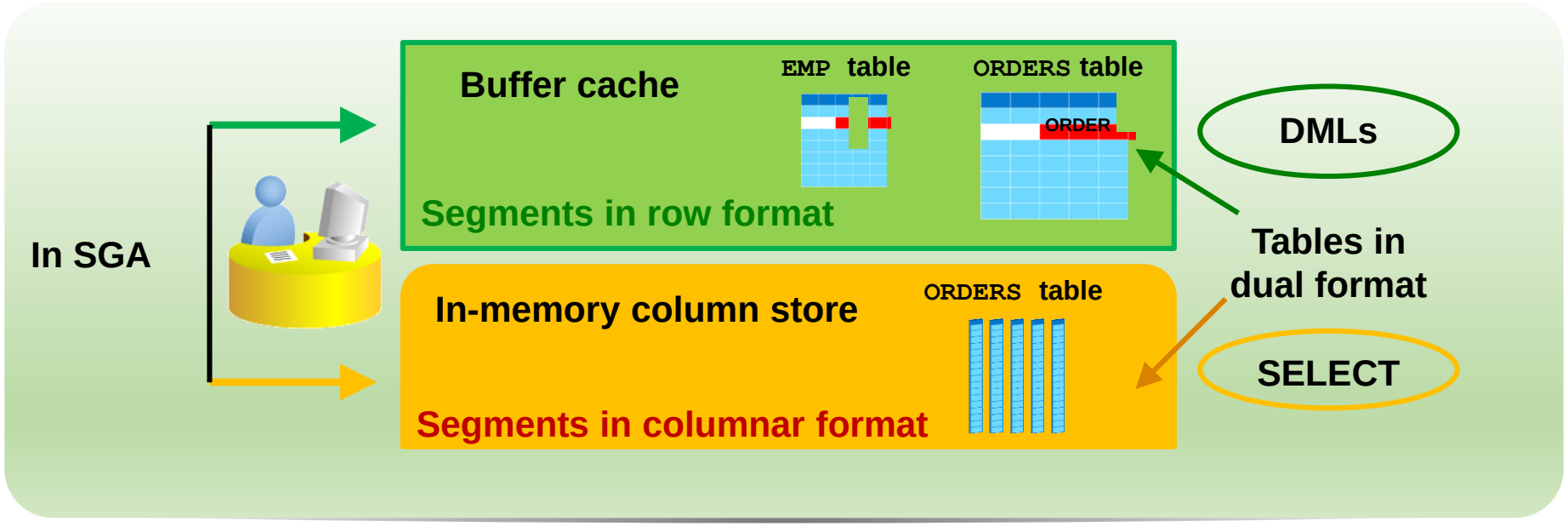
In-Memory Column Store: Introduction

- Instant query response:
  - Faster queries on very large tables on any columns (100x)
  - Use of scans, joins, and aggregates
  - Without indexes
  - Best suited for analytics: few columns, many rows
- Faster DML: Removal of most analytics indexes (3 to 4x)
- Full application transparency
- Easy setup:
  - In-memory column store configuration
  - Segment attributes

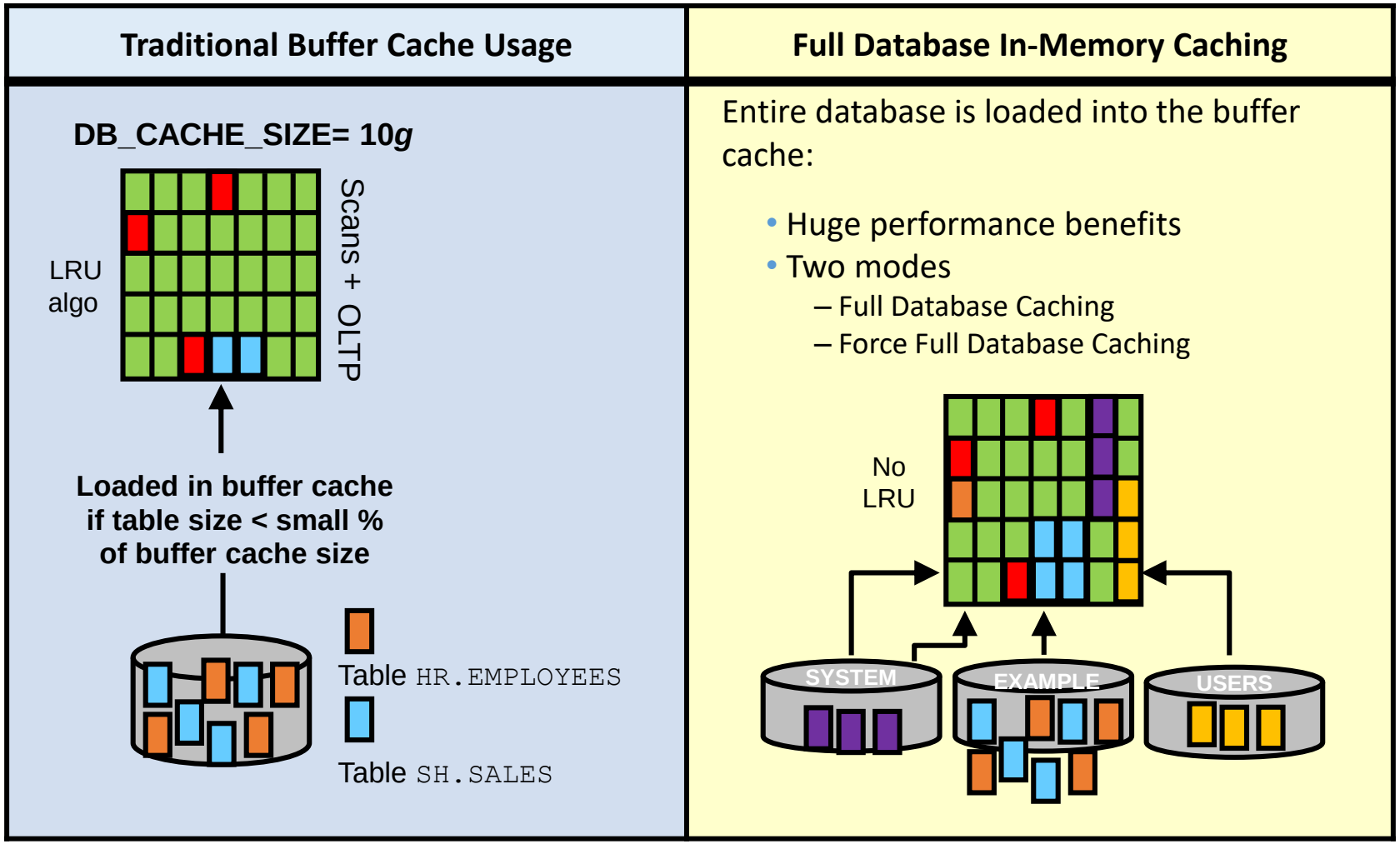


In-Memory Column Store: Overview

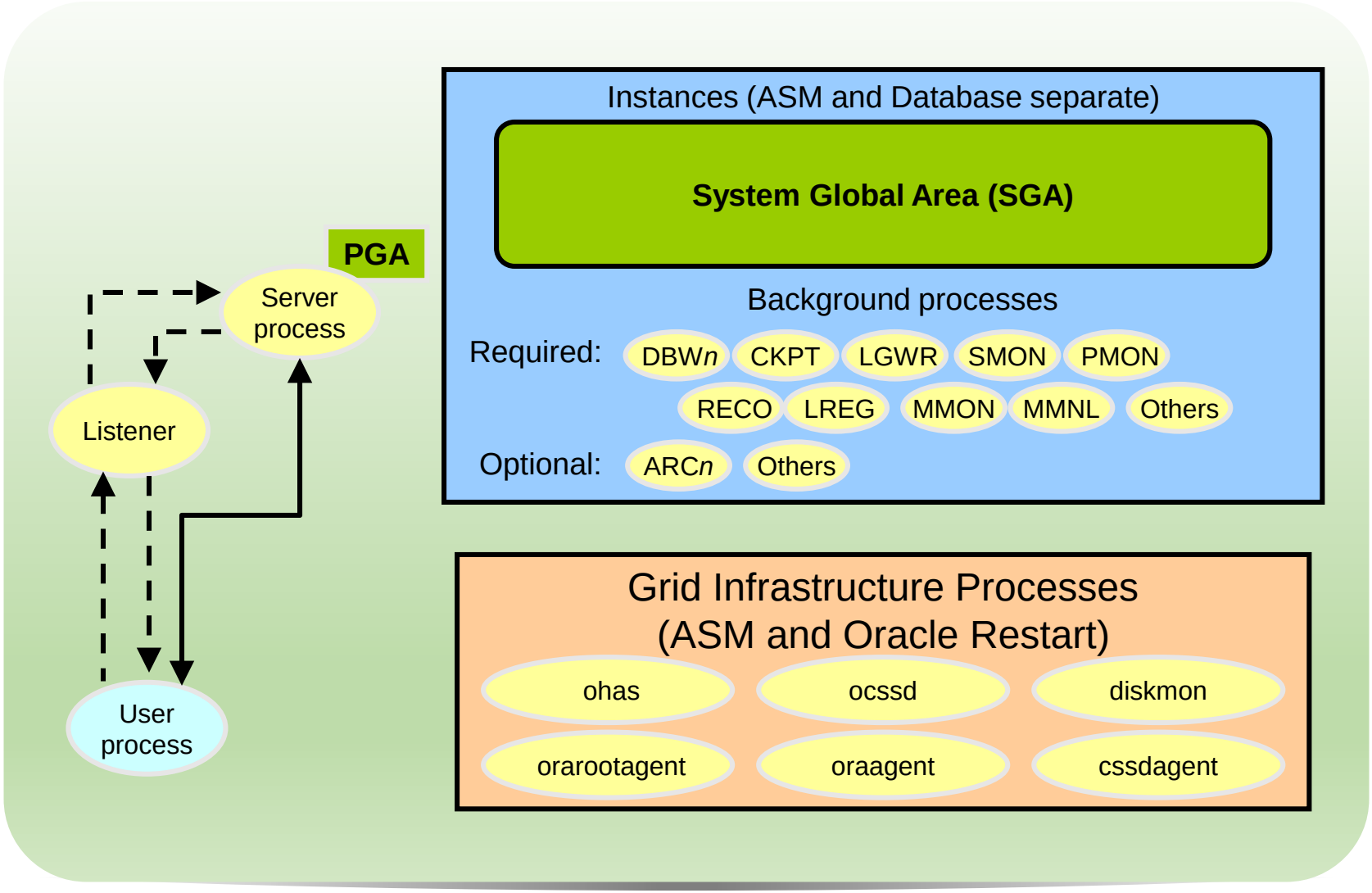
- A pool in the SGA: In-Memory column store
  - Segments populated into the IM column store are converted into a columnar format.
  - In-Memory segments are transactionally consistent with the buffer cache.
- Only one segment on disk and in row format



Full Database In-Memory Caching

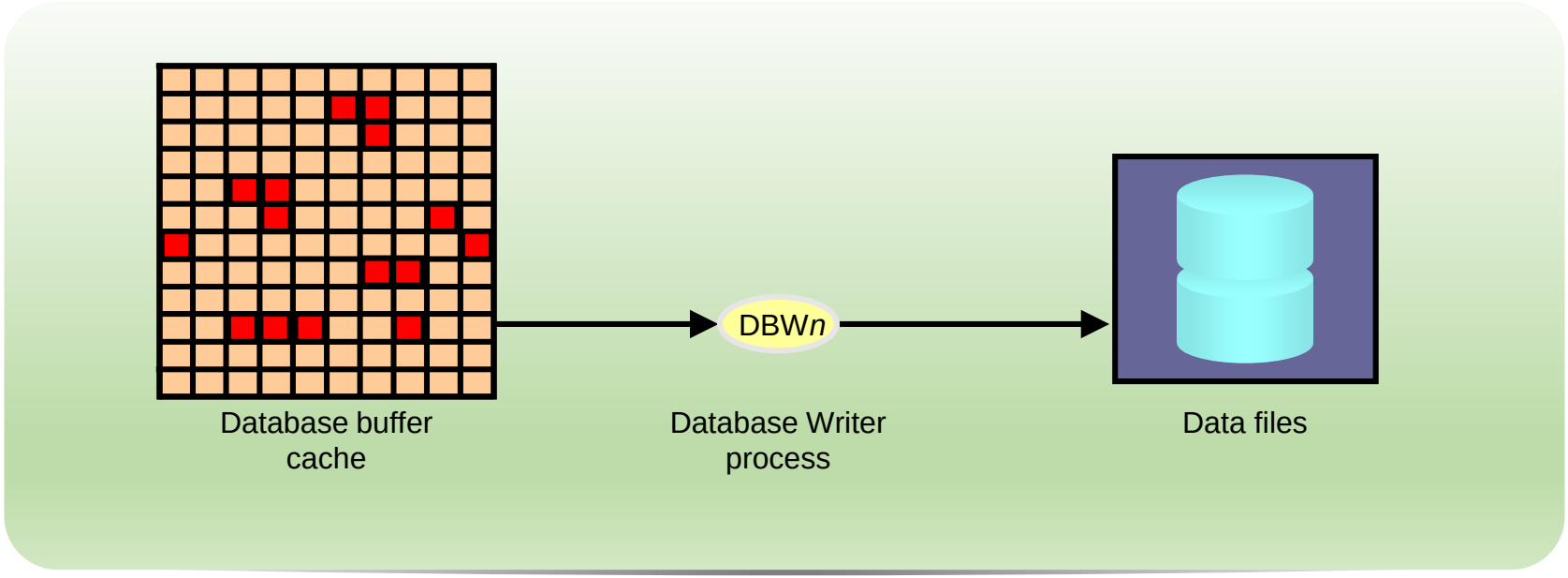


Process Structures



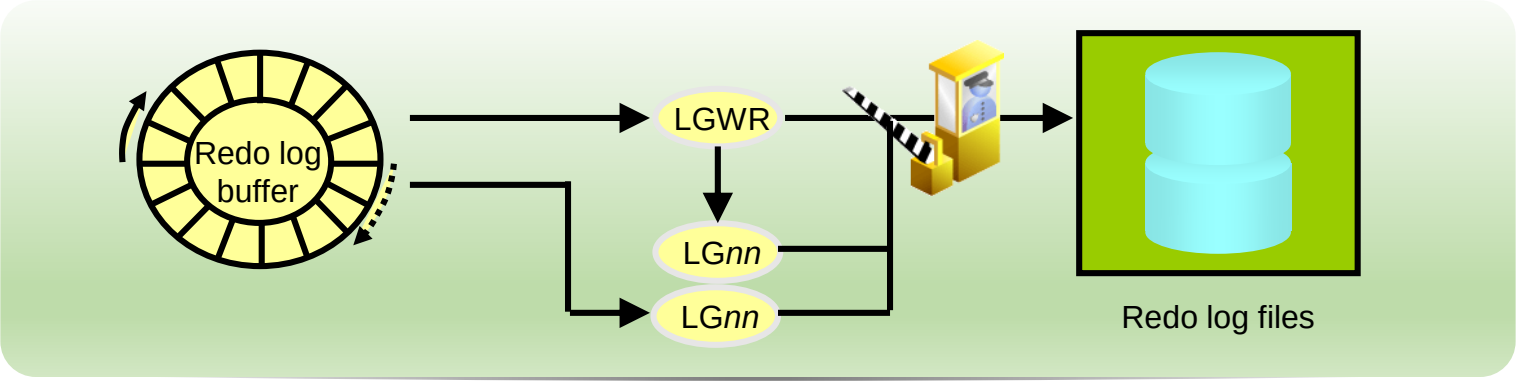
Database Writer Process (DBWn)

- Writes modified (dirty) buffers in the database buffer cache to disk:
  - Asynchronously while performing other processing
  - To advance the checkpoint



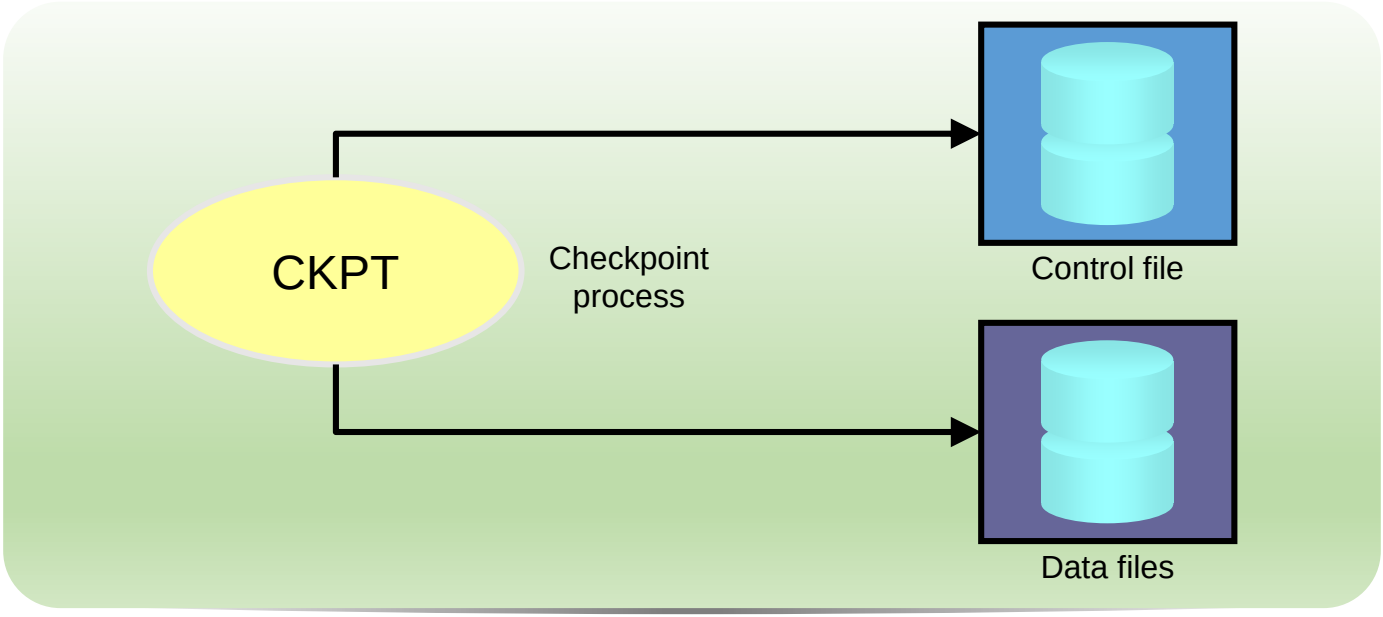
Log Writer Process (LGWR)

- Writes the redo log buffer to a redo log file on disk:
  - When a user process commits a transaction
  - When an online redo log switch occurs
  - When the redo log buffer is one-third full or contains 1 MB of buffered data
  - Before a DBW*n* process writes modified buffers to disk
  - When three seconds have passed since the last write
- Serves as coordinator of LG*nn* processes and ensures correct order for operations that must be ordered



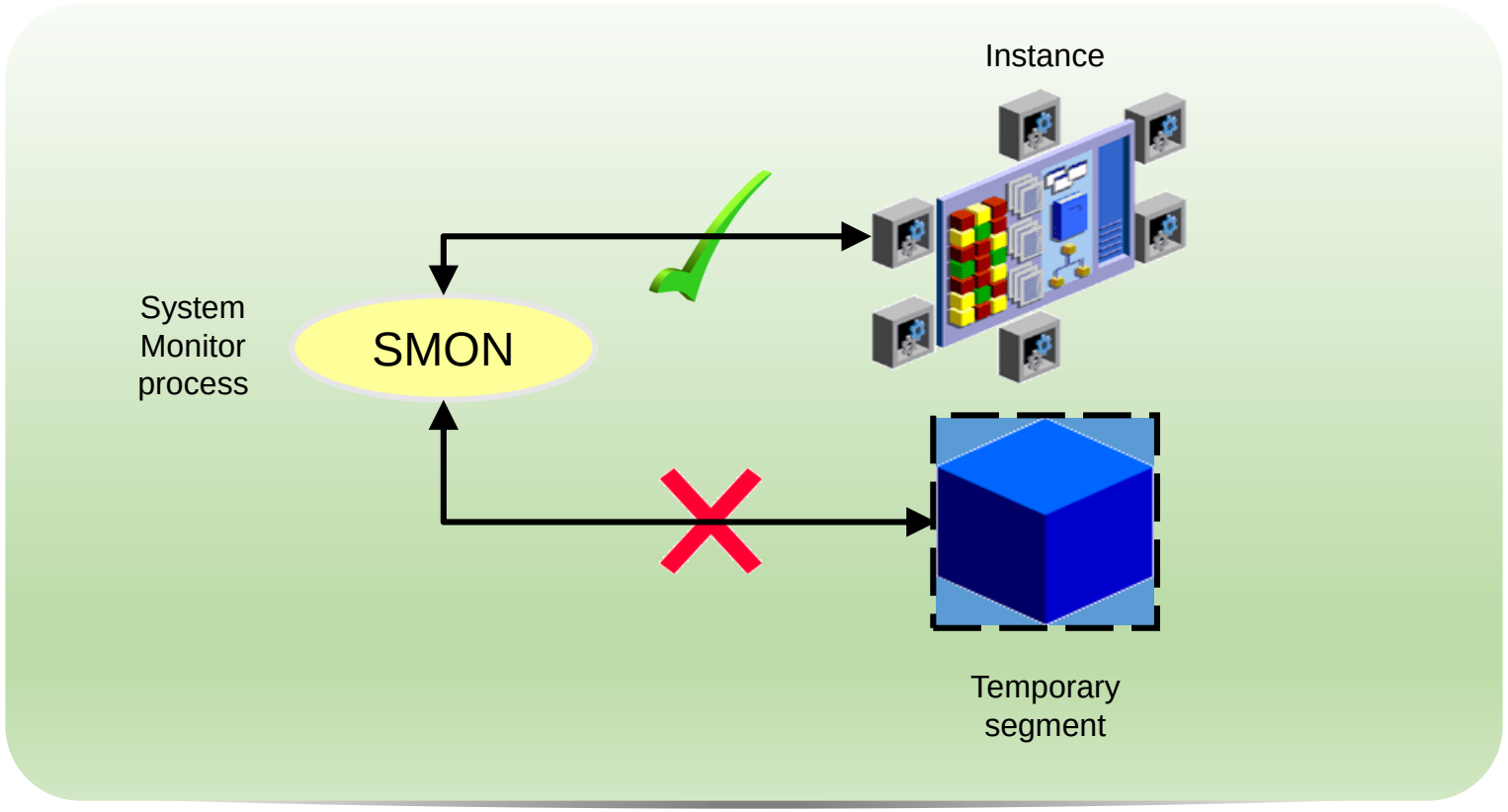
Checkpoint Process (CKPT)

- Records checkpoint information in:
  - The Control file
  - Each data file header
- Signals DBW*n* to write blocks to disk



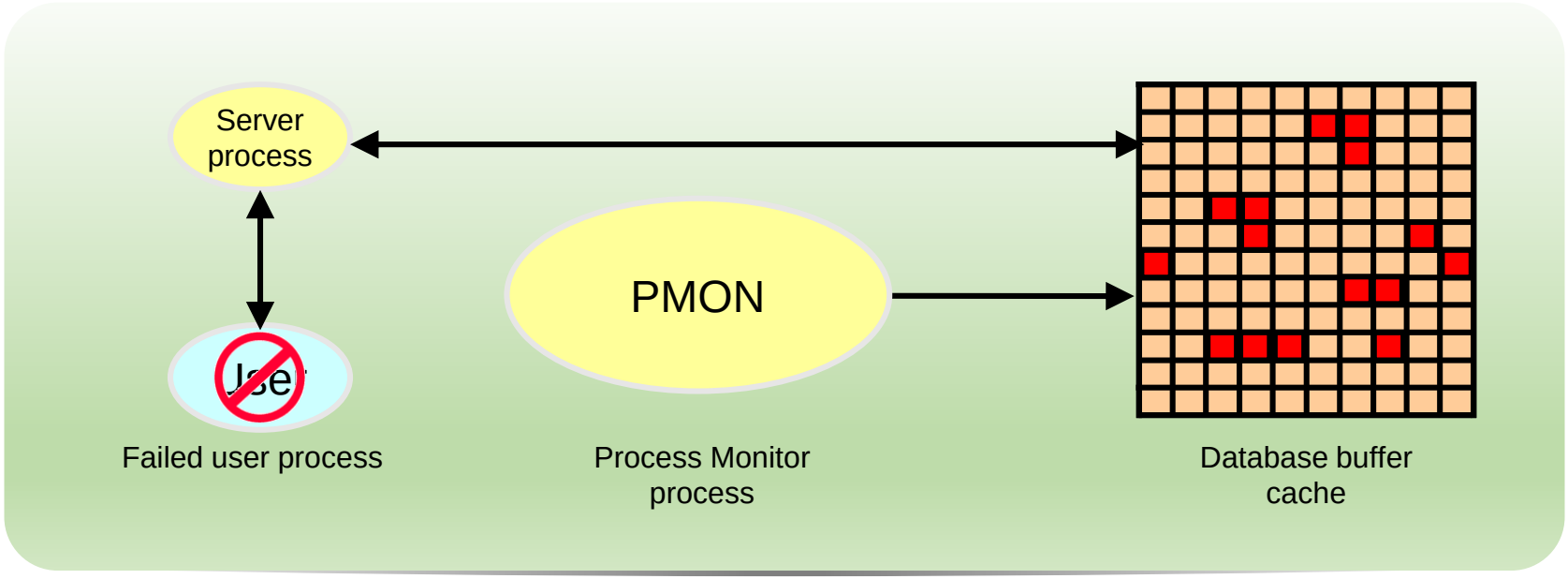
System Monitor Process (SMON)

- Performs recovery at instance startup
- Cleans up unused temporary segments



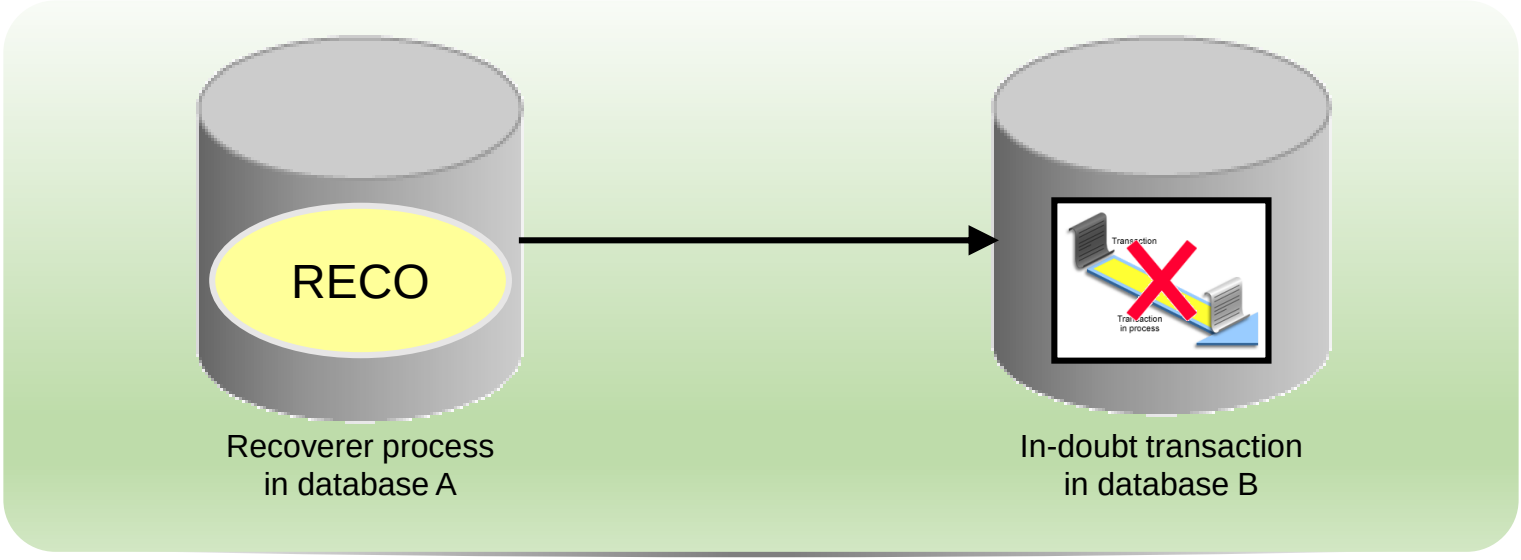
Process Monitor Process (PMON)

- Performs process recovery when a user process fails
  - Cleans up the database buffer cache
  - Frees resources that are used by the user process
- Monitors sessions for idle session timeout



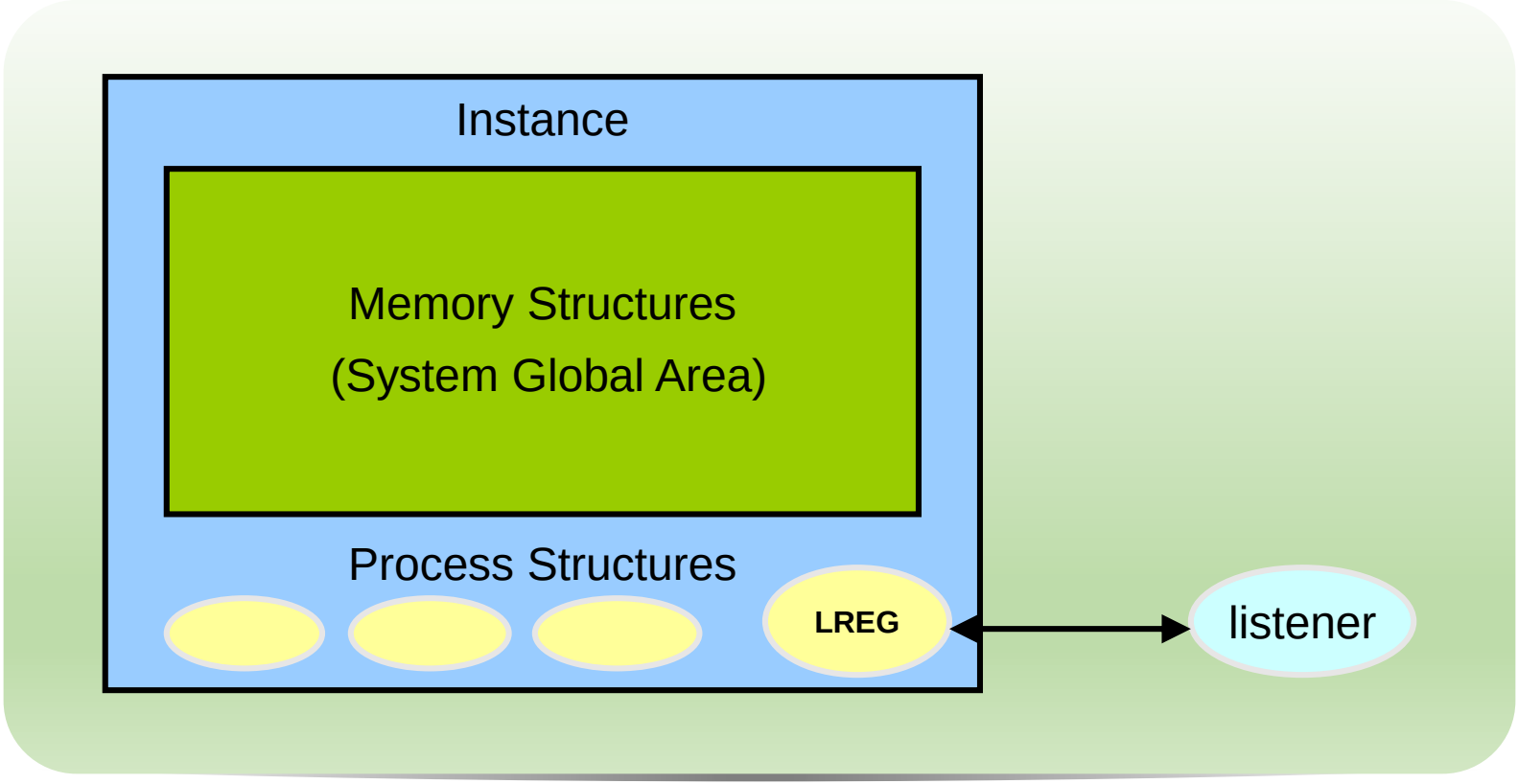
Recoverer Process (RECO)

- Used with the distributed database configuration
- Automatically connects to other databases involved in in-doubt distributed transactions
- Automatically resolves all in-doubt transactions
- Removes any rows that correspond to in-doubt transactions



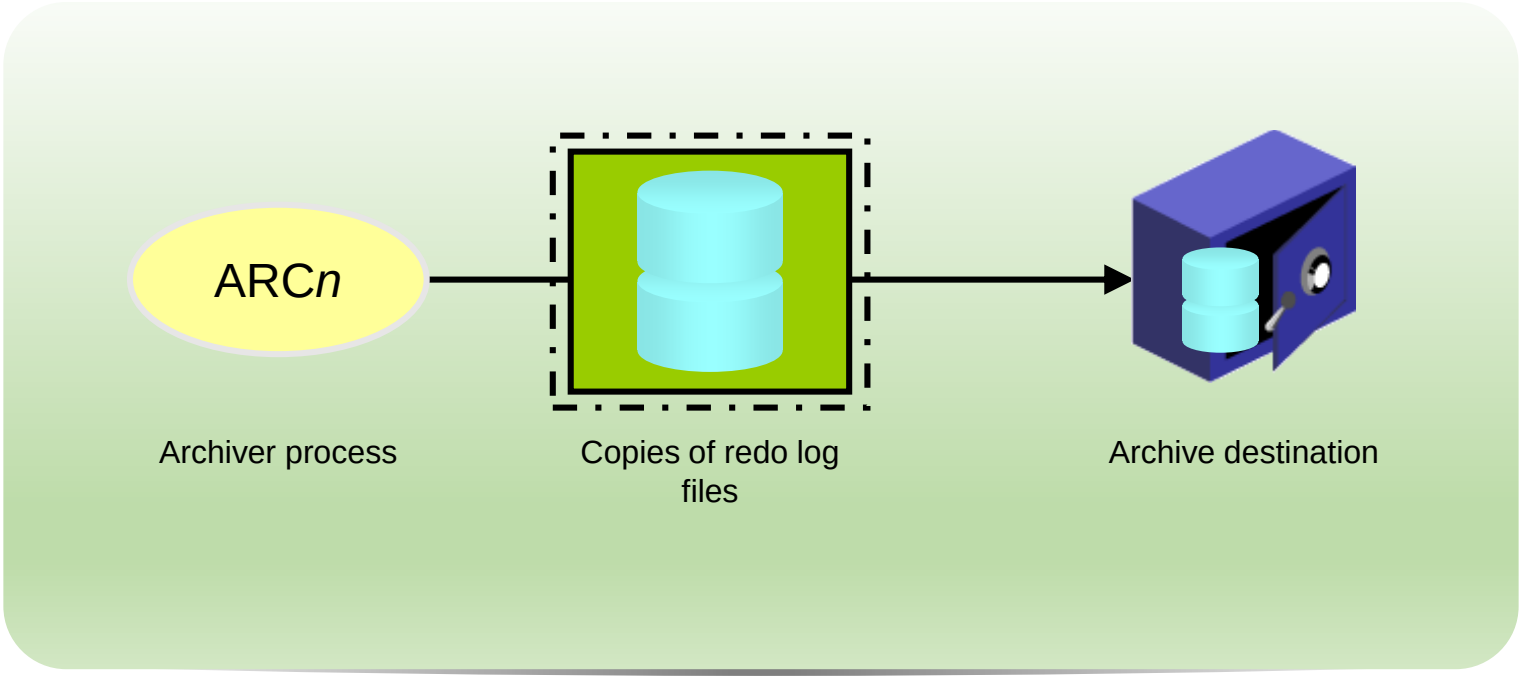
Listener Registration Process (LREG)

- Registers information about the database instance and dispatcher processes with Oracle Net Listener

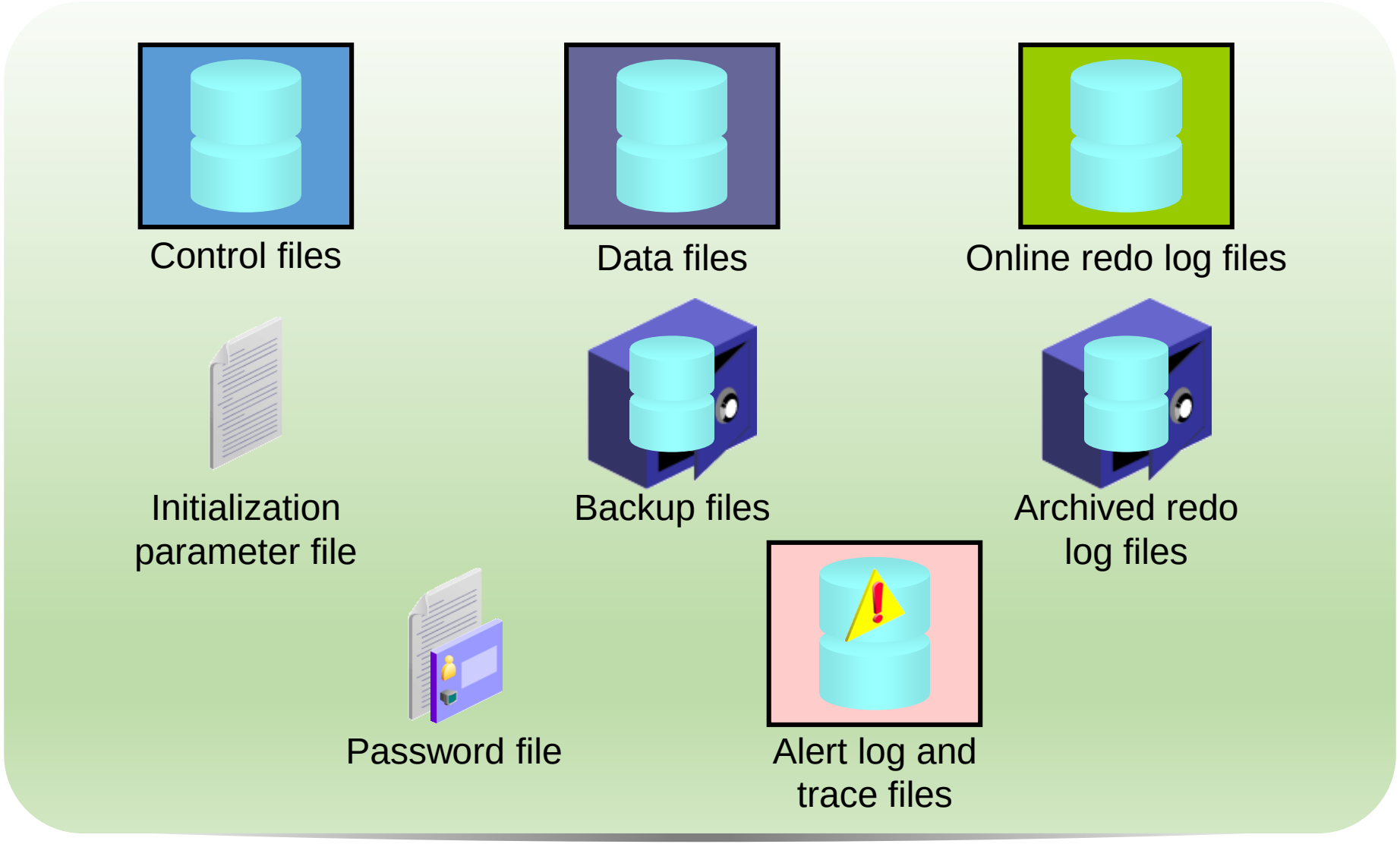


Archiver Processes (ARC*n*)

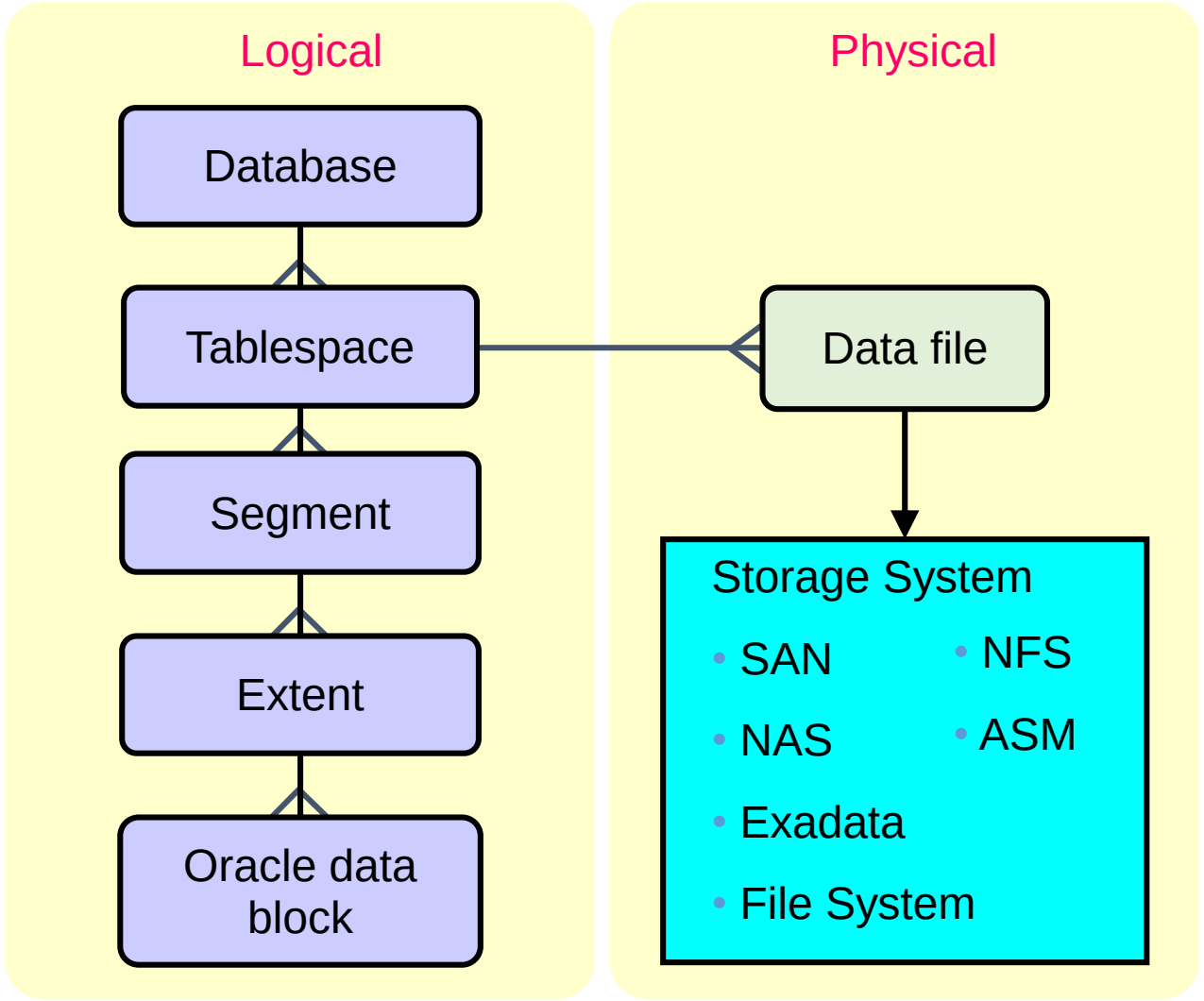
- Copy redo log files to a designated storage device after a log switch has occurred
- Can collect transaction redo data and transmit that data to standby destinations



Database Storage Architecture

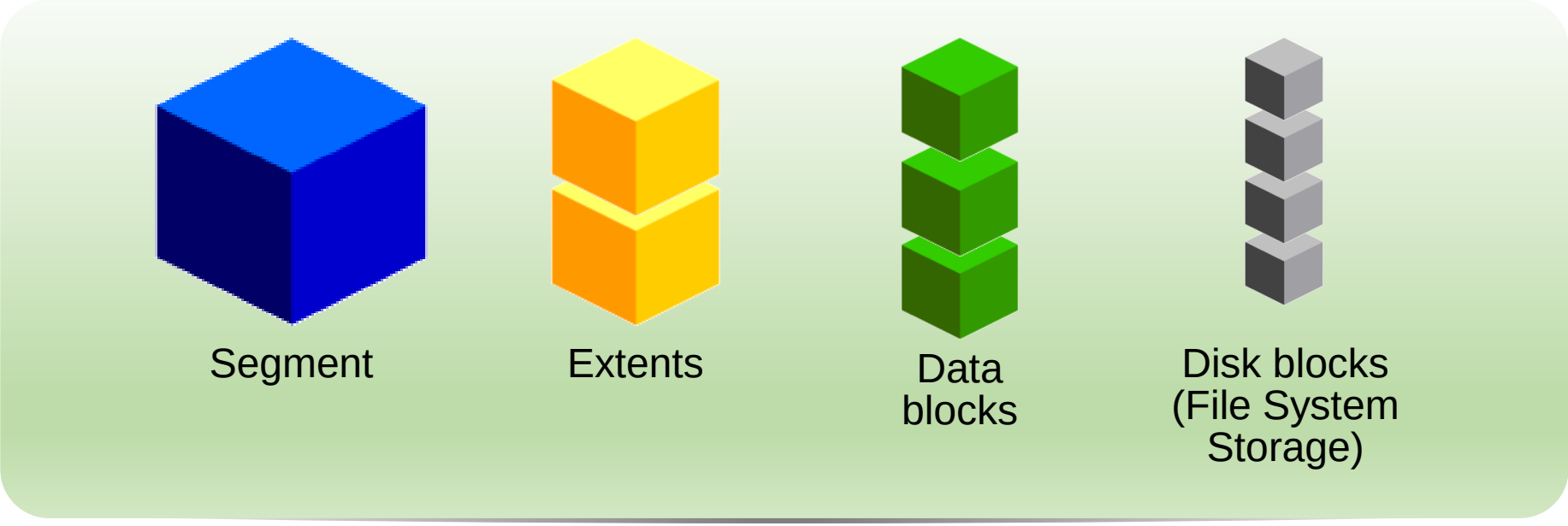


Logical and Physical Database Structures

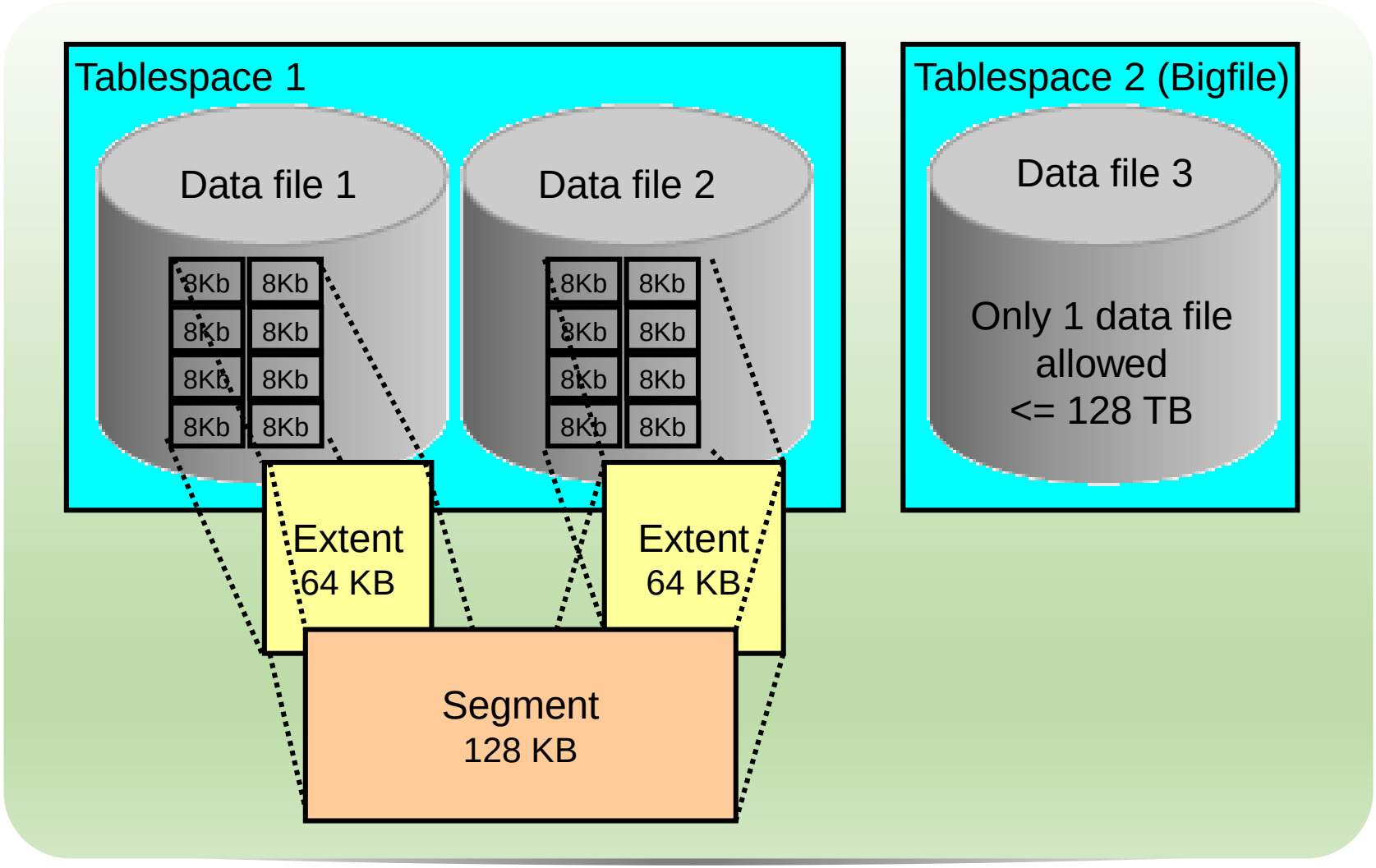


Segments, Extents, and Blocks

- Segments exist in a tablespace.
- Segments are collections of extents.
- Extents are collections of data blocks.
- Data blocks are mapped to disk blocks.



Tablespaces and Data Files



Default Tablespaces

Tablespace	Description
SYSTEM	The SYSTEM tablespace is used for core functionality.
SYSAUX	The SYSAUX tablespace is an auxiliary tablespace to the SYSTEM tablespace and helps reduce the load on the SYSTEM tablespace.
TEMP	The TEMP tablespace contains schema objects only for a session's duration.
UNDO	The UNDO tablespace stores the data needed to roll back, or undo, changes to the database.
USERS	The USERS tablespace stores user objects and data.

SYSTEM and SYSAUX Tablespaces

- The SYSTEM and SYSAUX tablespaces are mandatory tablespaces that are created at the time of database creation. They must be online.
- The SYSTEM tablespace is used for core functionality (for example, data dictionary tables).
- The auxiliary SYSAUX tablespace is used for additional database components.
- The SYSTEM and SYSAUX tablespaces should not be used for application data.

Implementing Oracle Managed Files (OMF)

- Specify file operations in terms of database objects rather than file names.

Parameter	Description
DB_CREATE_FILE_DEST	Defines the location of the default file system directory for data files and temporary files
DB_CREATE_ONLINE_LOG_DEST_n	Defines the location for redo log files and control file creation
DB_RECOVERY_FILE_DEST	Gives the default location for the fast recovery area

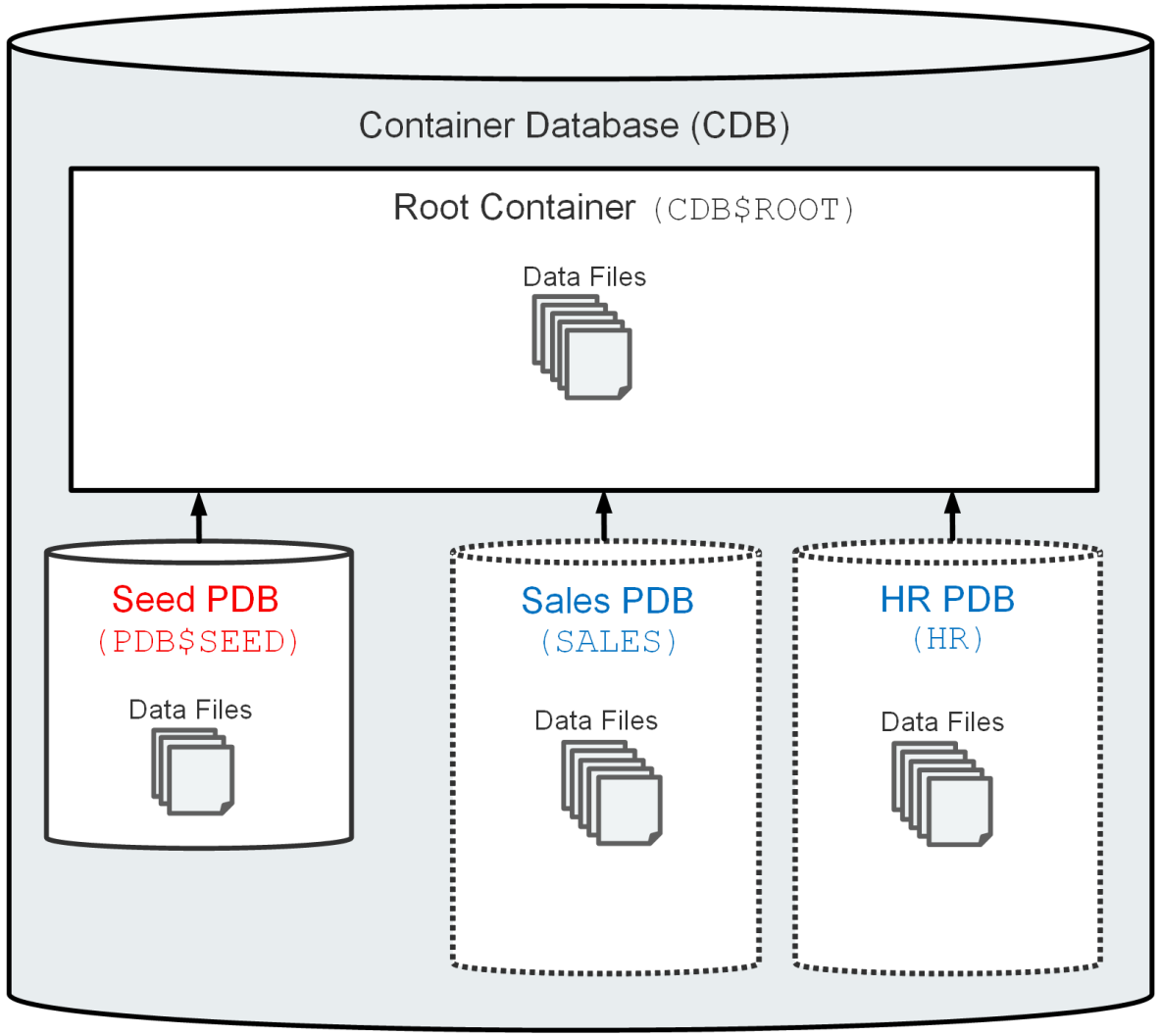
Example:

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST='/u01/app/oracle/oradata';
SQL> CREATE TABLESPACE tbs_1;
```

Oracle Container Database: Introduction

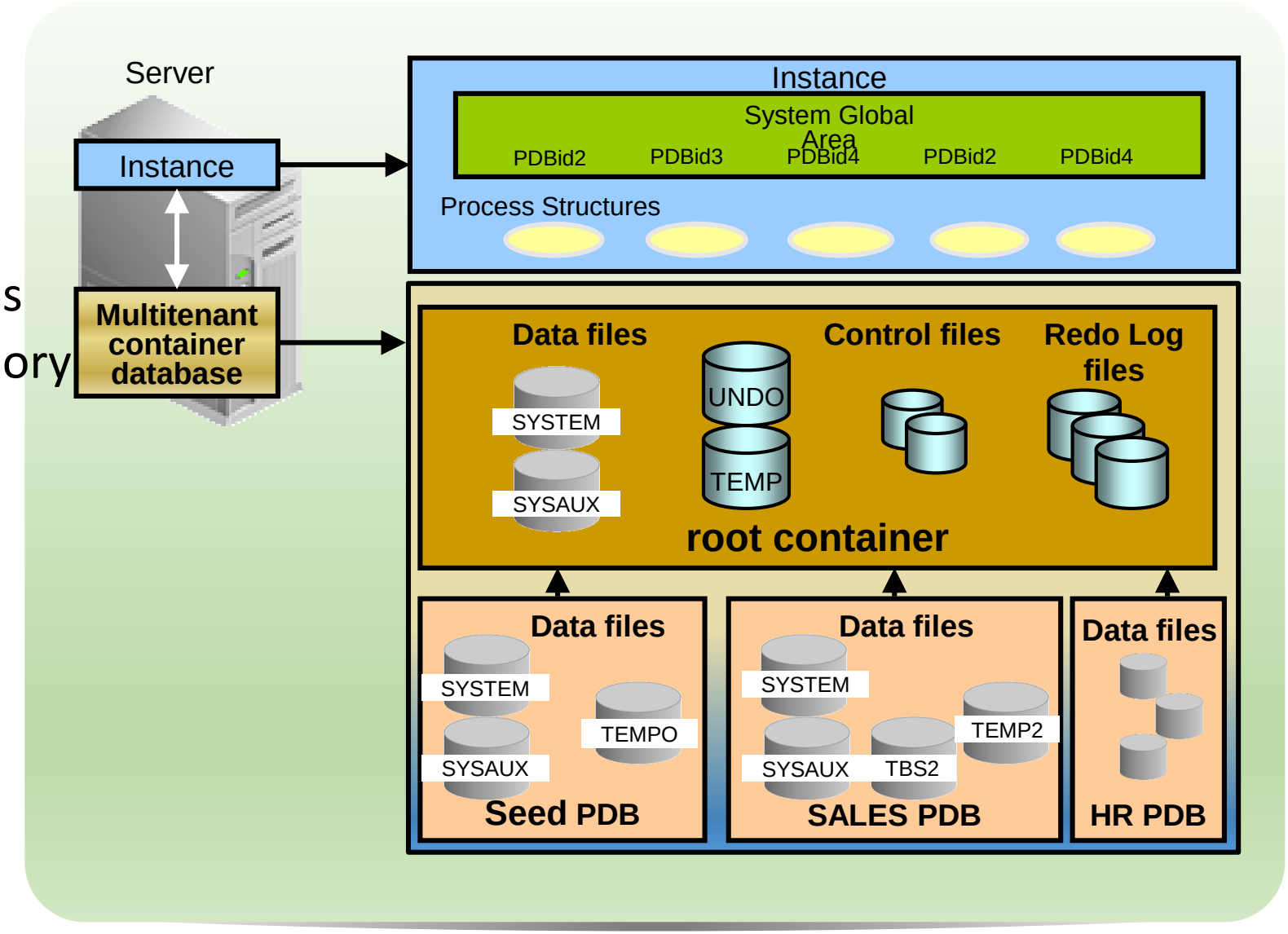
- *Pluggable database*: Is a set of database schemas that appears logically to users and applications as a separate database
- *Multitenant container database*: Has a database instance and database files at the physical level
- All pluggable databases share:
  - Background processes
  - Shared/process memory
  - Oracle metadata

Multitenant Database



Multitenant Architecture

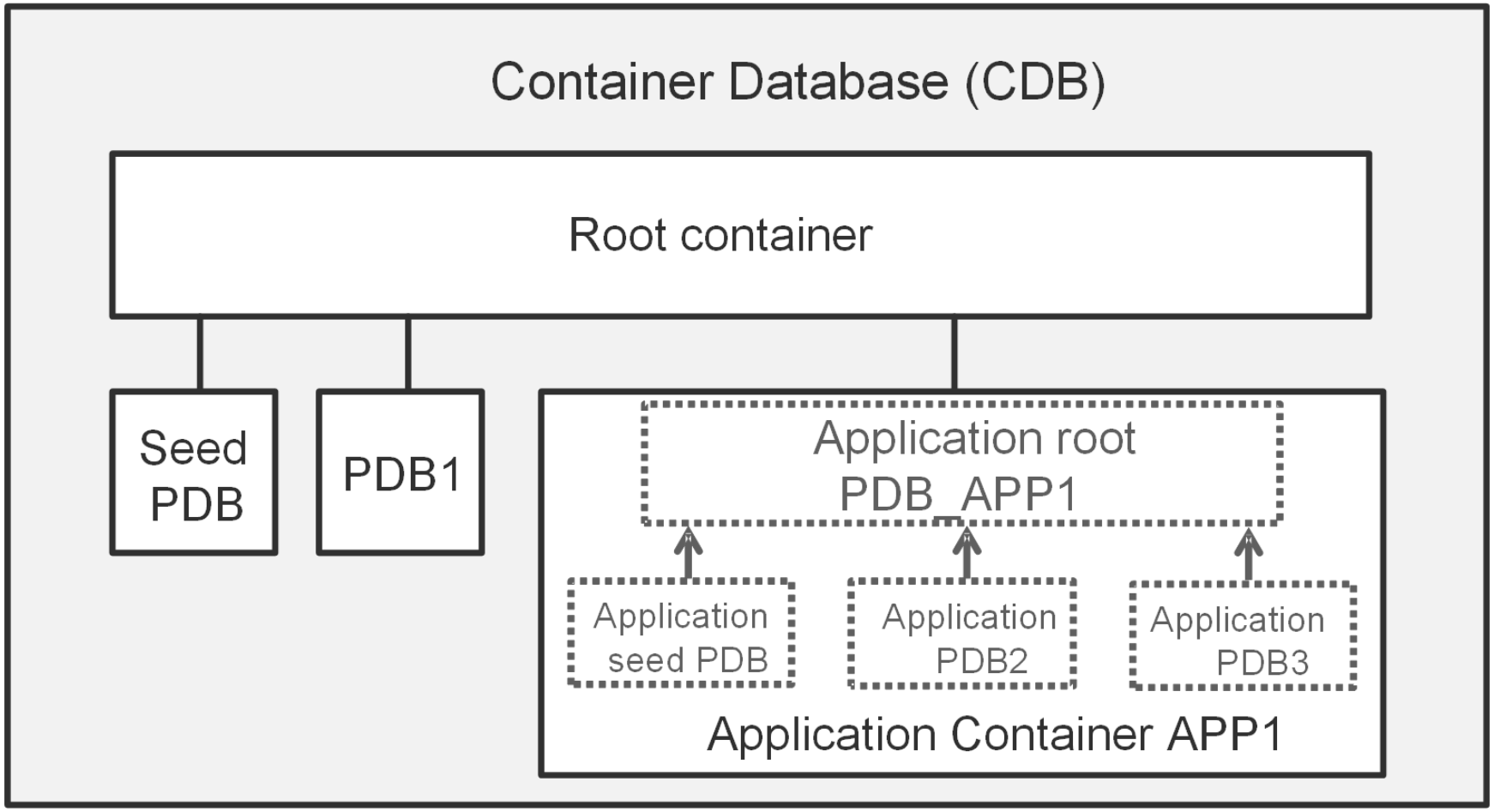
- All PDBs share:
  - Background processes
  - Shared/process memory
  - Oracle metadata
  - Redo log files
  - Control files
  - Undo tablespace



Default Tablespaces in the Multitenant Architecture

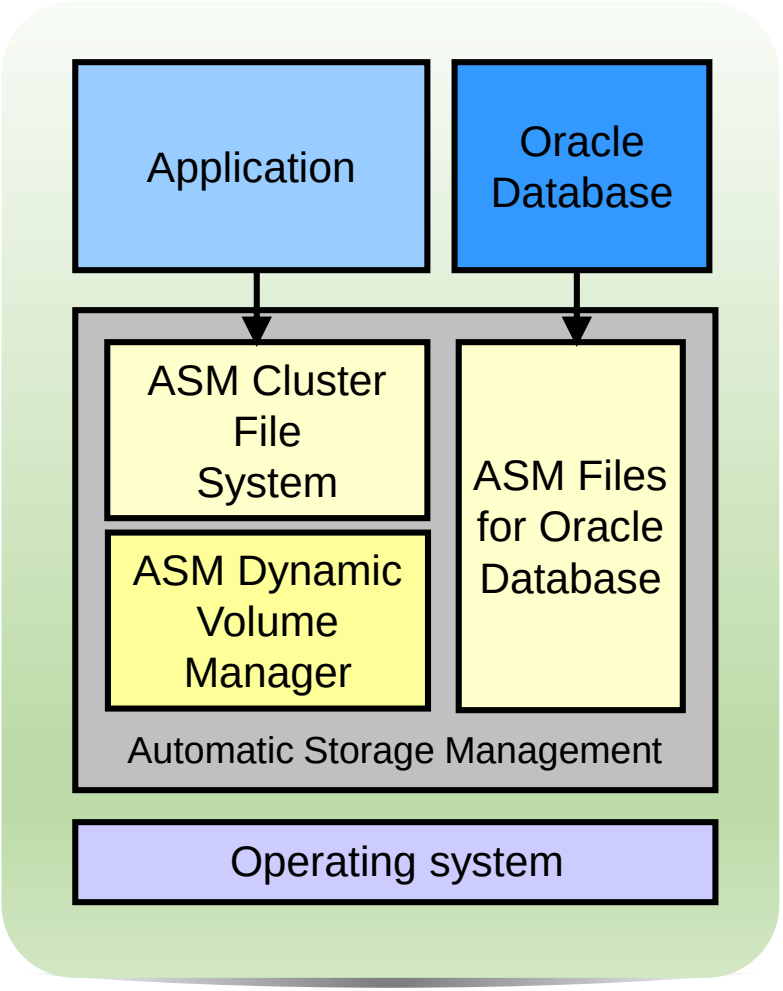
Tablespace	Description
SYSTEM	In the root container, it contains Oracle-supplied metadata. In a PDB, it contains user metadata.
SYSAUX	It exists in the root container and in each PDB.
TEMP	By default, the root container has a single default temporary tablespace that every PDB uses. You can create separate temporary tablespaces in PDBs.
UNDO	One active undo tablespace exists in the root container. It is recommended that you have a local undo tablespace in each PDB.
USERS	The root container and each PDB have a USERS tablespace.

Application Containers

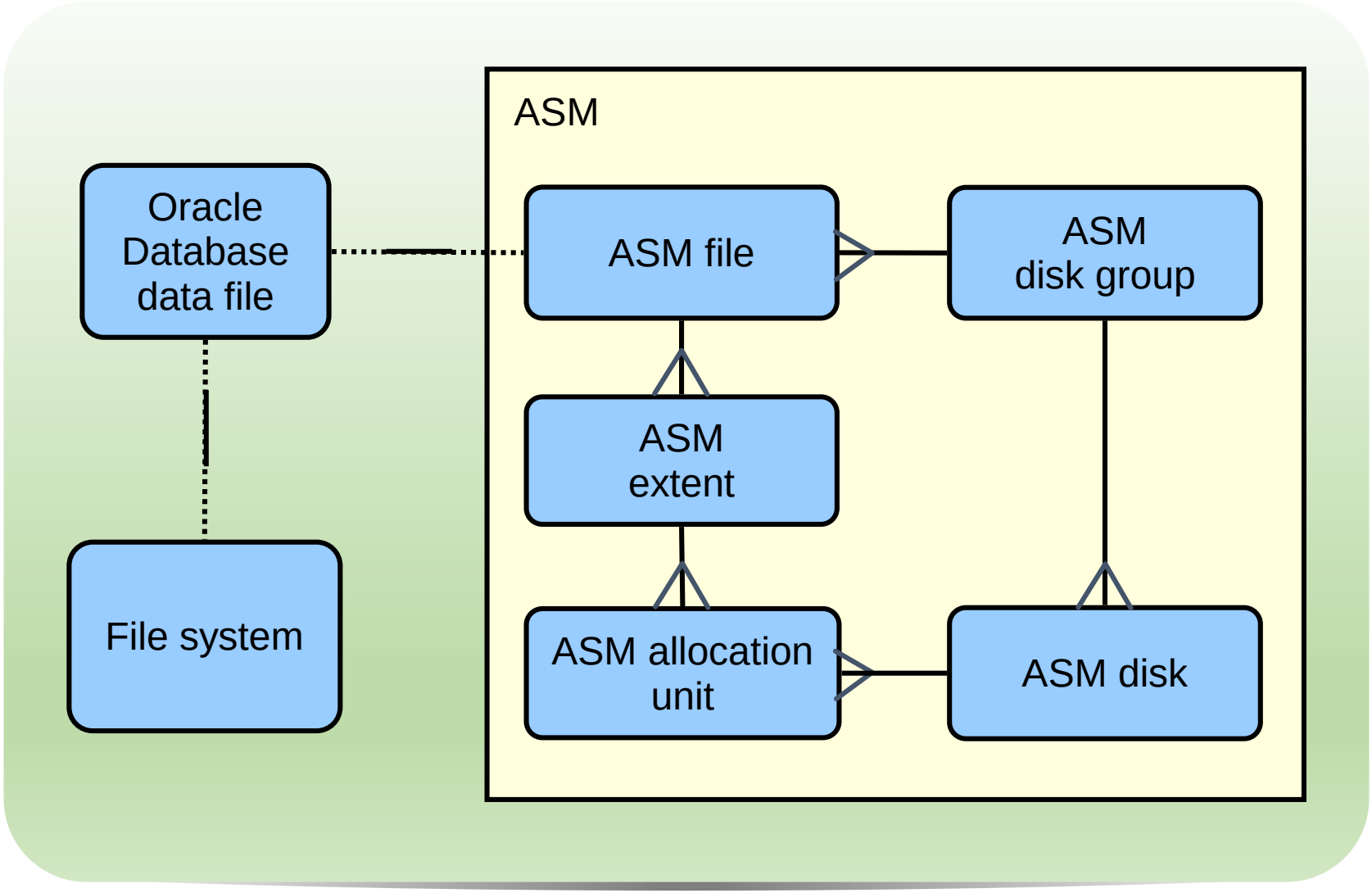


Automatic Storage Management

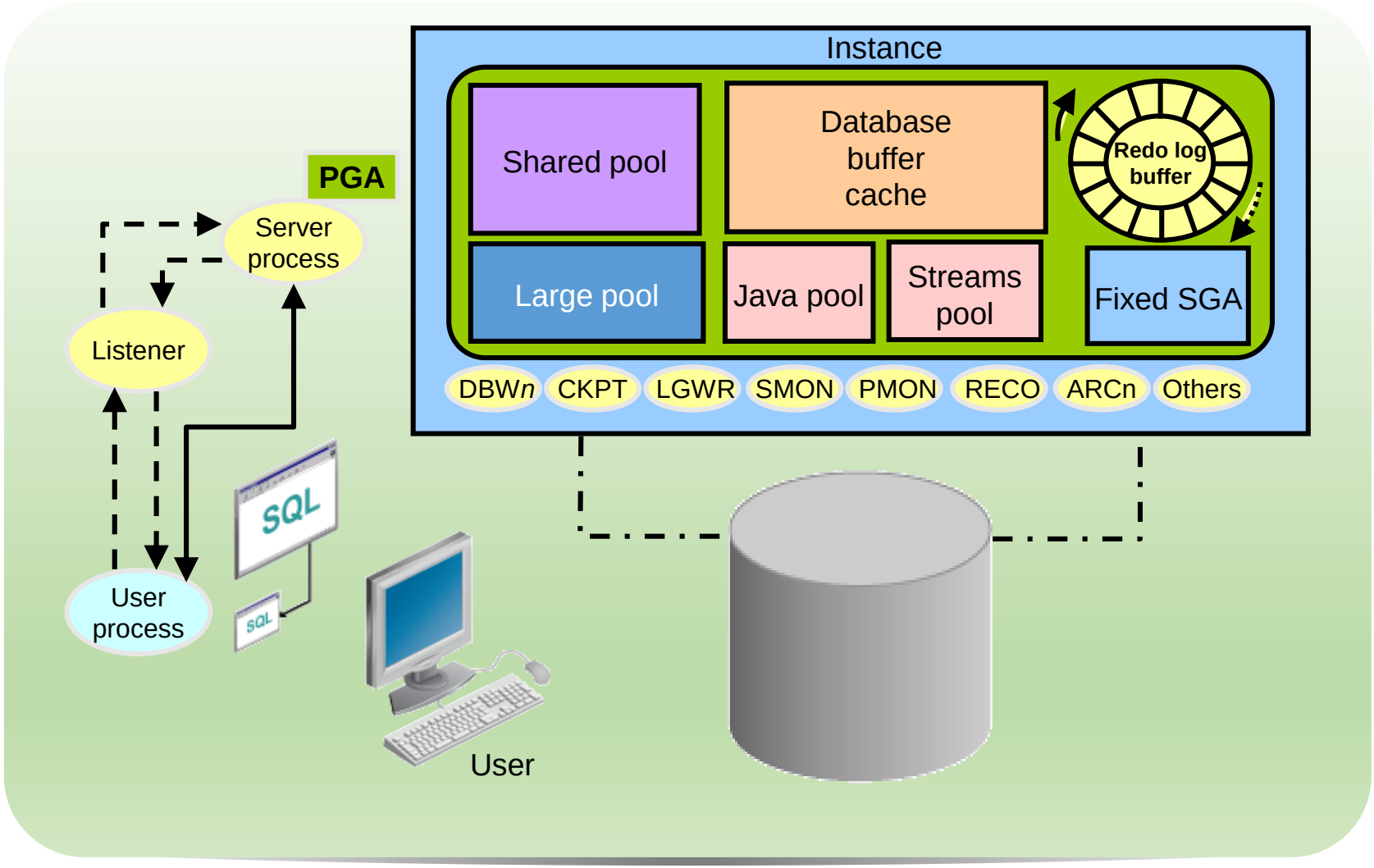
- Is a portable and high-performance cluster file system
- Manages Oracle database files
- Manages application files with ASM Cluster File System (ACFS)
- Spreads data across disks to balance load
- Mirrors data in case of failures
- Solves storage management challenges



ASM Storage Components



Interacting with an Oracle Database: Memory, Processes, and Storage



Summary

- In this lesson, you should have learned how to:
  - List the major architectural components of Oracle Database
  - Explain memory structures
  - Describe background processes
  - Correlate logical and physical storage structures
  - Describe multitenant architecture