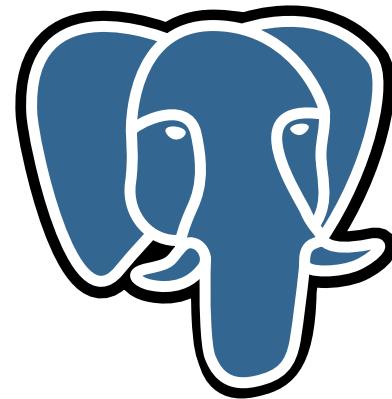


Czech and Slovak PostgreSQL Users Group



PostgreSQL for Oracle DBA

Oracle database logo should be there... but as of this page it can't be:
...Remember that you are generally not able to use Oracle logos unless
you are specifically licensed or authorized to use them. ...



Agenda

- Mind Migration
- Some terminology
- “Architecture”
- Security
- Backup and Recovery
- High Availability / Disaster recovery
- Other unordered stuff to consider

Purpose of these slides



- Understand differences
- Commons in behavior
- Displeasure prevention
- Migration guide
- RDBMS ranking
- Feature matrix

Mind Migration

- PostgreSQL is not an Oracle database
- Oracle and PostgreSQL are both superb databases, their relation is not like Red Hat and CentOS
- Do not expect equivalents for all of the Oracle RDBMS features in PostgreSQL
- Don't hesitate to be impressed by PostgreSQL broad range of data types
- PostgreSQL has extensions



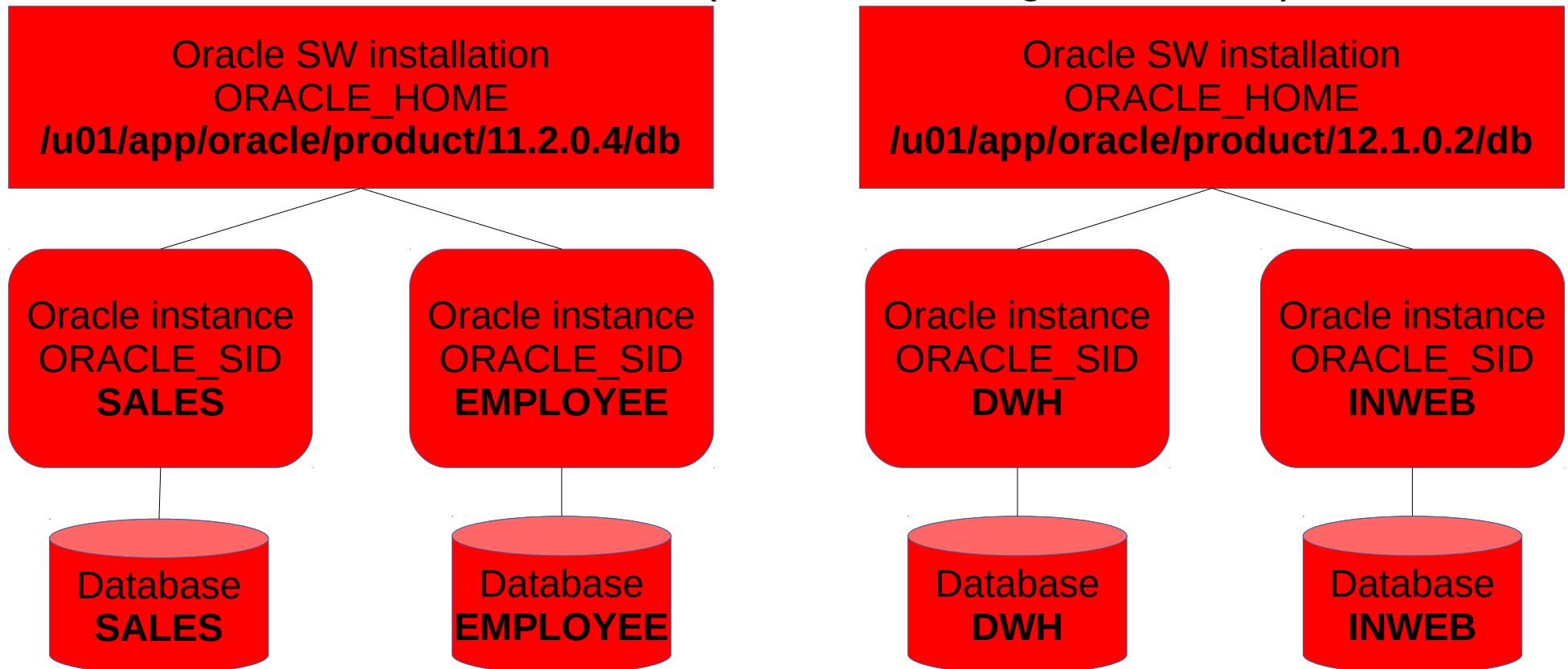
Terminology

- Architecture/concepts
 - Cluster Yes, a cluster can simply mean a grouping of related things, but context is everything.
Shaun M. Thomas
 - Instance
 - Database
 - Tablespace
- Logical
 - Role
 - User
 - Schema
 - Tuple [Wiki](#): A tuple is a finite ordered list of elements
- quick browse through manuals will help and don't take ages
 - Architectural Fundamentals
 - Documentation index



Oracle and DB relation

- From installed software to database (simplified)
 - Instance is software loaded into memory working with **ONE** database (12c PDB changed that rule)





PostgreSQL and DB relation

- From installed software to database (simplified)
 - SW installed from RPM/APT/compiled from source

PostgreSQL installation 9.4
~~POSTGRESQL_HOME~~
/usr/pgsql-9.4/ : /usr/pgsql-9.4/lib

PostgreSQL installation 9.3
~~POSTGRESQL_HOME~~
/usr/bin/postgres : /usr/lib64/libpq.so

PostgreSQL instance
Running postgres process
PGDATA
/var/lib/pgsql/9.4/data

PostgreSQL instance
Running postgres process
PGDATA
/var/lib/pgsql/9.4/data2

Database
template0

Database
template1

Database
postgres

Database
INWEB

Database
DWH

Database
SALES

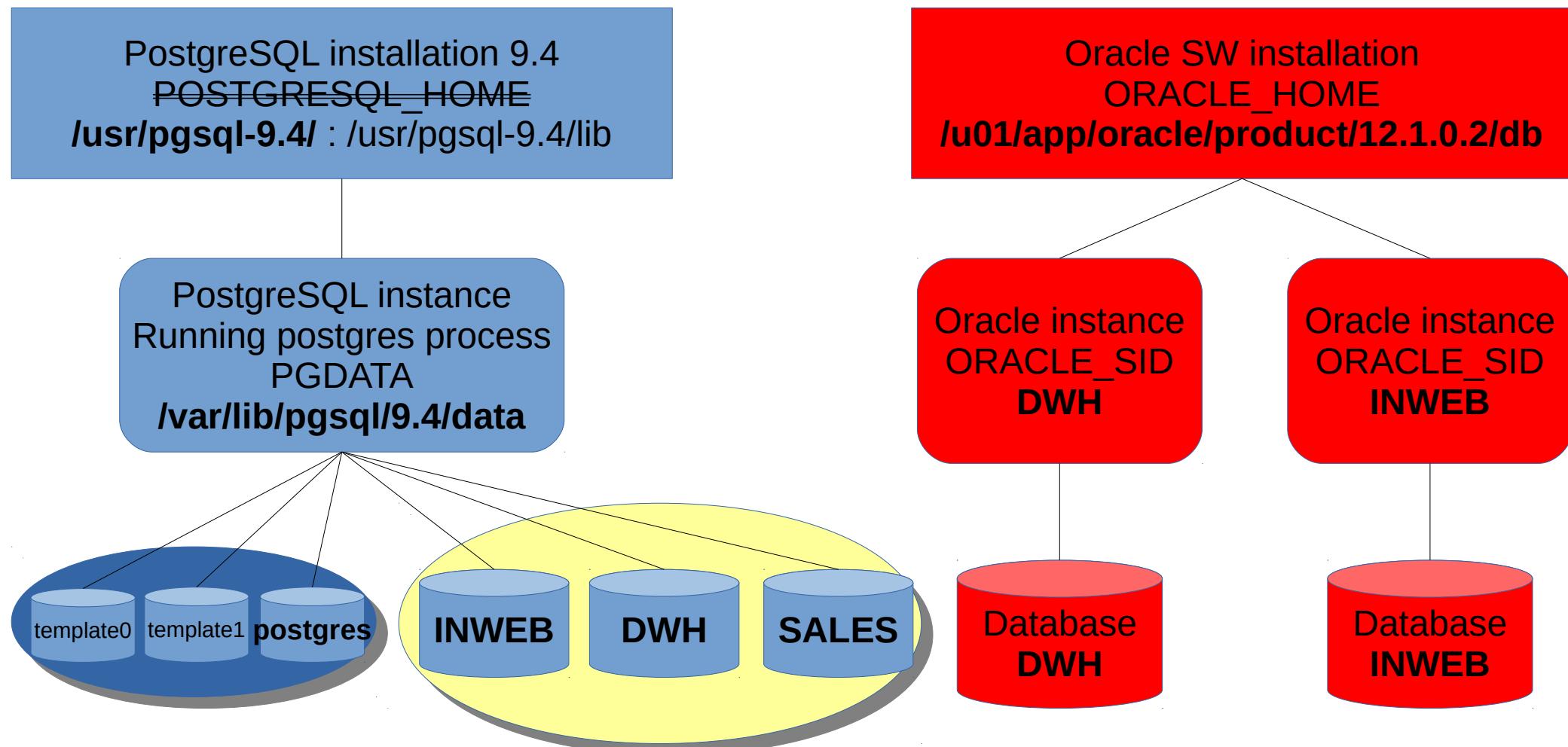
Internal purpose, meta data catalog

Applications databases



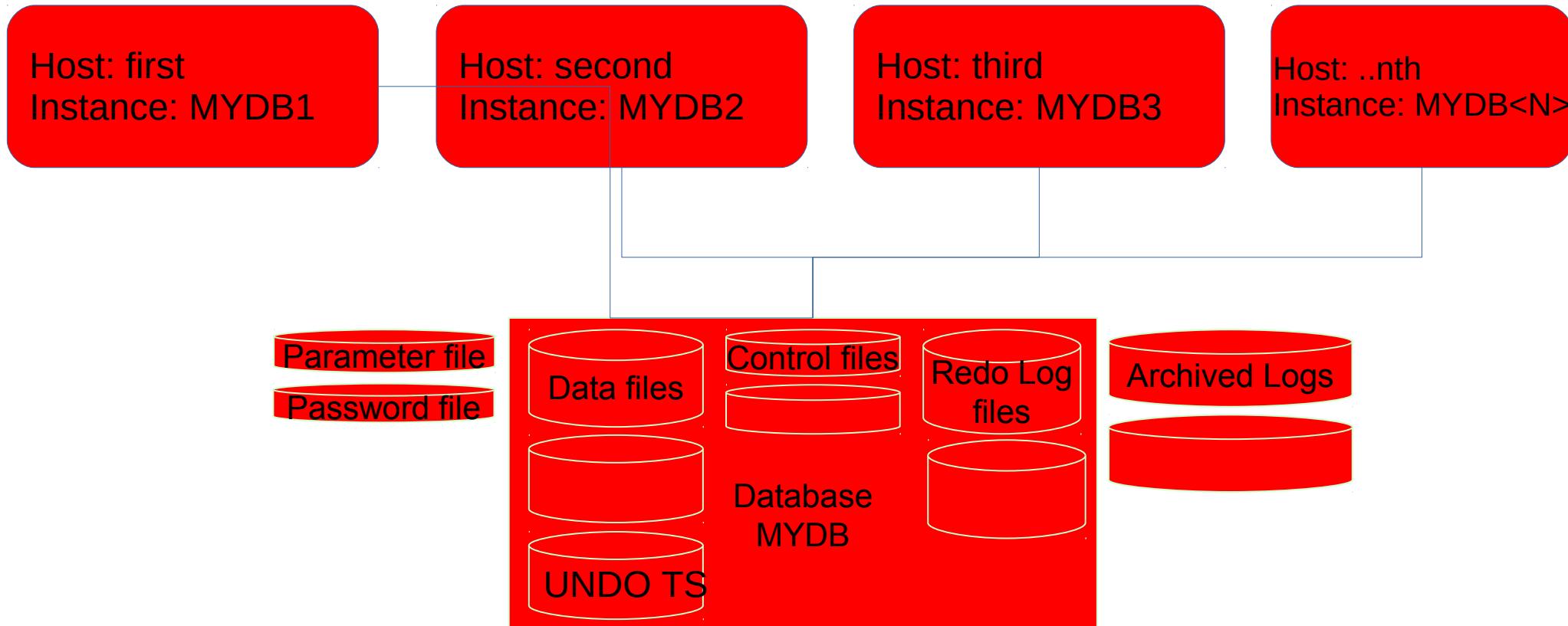
Pg & Oracle – compared

- From installed software to database (simplified)



Oracle – DB physical structure

- Oracle RAC (Real Application Clusters)
 - Even with RAC setup, an Oracle instance serves **ONE** database
 - Multitenant pluggable databases in version 12 breaks that rule...
- Oracle database physical components (files)



Postgres – cluster physical structure

Not means clustering like Oracle RAC

- Postgres uses directory (might be referred by environment variable PGDATA) traditionally called **database cluster** to store all necessary data managed by Postgres instance
 - Which is mandatory parameter for starting Postgres instance
 - Contains configuration files, and in default setup also files for all databases residing within a particular Postgres cluster, see documentation [Database File Layout](#)

```
-bash-4.3$ ps -fu postgres
UID      PID  PPID  C STIME TTY          TIME CMD
postgres 30007     1  0 19:03 ?
00:00:00 /usr/bin/postgres -D /var/lib/pgsql/data -p 5433
```

base/
 global/
 pg_clog/
pg_hba.conf
pg_ident.conf
 pg_log/
 pg_multixact/
 pg_notify/
 pg_serial/
 pg_snapshots/

pg_stat/
 pg_stat_tmp/
 pg_subtrans/
 pg_tblspc/
 pg_twophase/
 PG_VERSION
 pg_xlog/
postgresql.conf
 postmaster.opts
 postmaster.pid

Configuration files **Databases directory**

```
postgres=# select oid, datname from
pg_database;
   oid   | datname
-----+-----
        1 | template1
12968  | template0
12973  | postgres
(3 rows)
```

```
-bash-4.3$ du -sh base/*
6.4M   base/1
6.4M   base/12968
6.5M   base/12973
```

```
tree base | head -5
base
└── 1
    ├── 12706
    ├── 12706_fsm
    └── 12706_vm
```



Postgres – mapping to Oracle files

- Postgres similar to Oracle might create archive of online logs for PITR and other purposes
 - Instead of “Oracle Archiver” server processes, Postgres used to call any external command responsible for copying inactive online log to some other destination

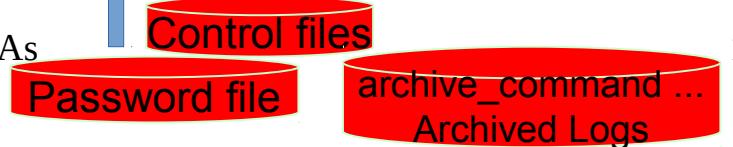
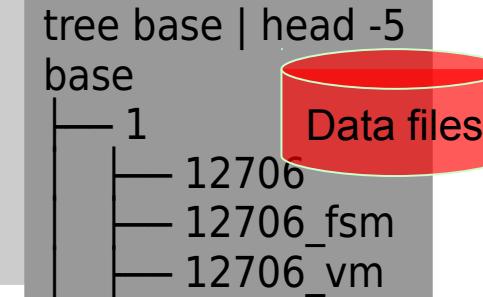
```
-bash-4.3$ ps -fu postgres
UID      PID  PPID  C STIME TTY          TIME CMD
postgres 30007     1  0 19:03 ?
                                                     database cluster
                                                     /var/lib/pgsql/data -p 5433
```

base/
global/
pg_clog/
pg_hba.conf Parameter file
pg_ident.conf file
pg_log/
pg_multixact/
pg_notify/
pg_serial/
pg_snapshots/

pg_stat/
pg_stat_tmp/
pg_subtrans/
pg_tblspc/
pg_twophase/
PG_VERSION
pg_xlog/ Online Redo Log
postgresql.conf Parameter file
postmaster.opts
postmaster.pid

Configuration files **Databases directory**

```
postgres=# select oid, datname from pg_database;
   oid   | datname
-----+-----
       1 | template1
 12968 | template0
 12973 | postgres
(3 rows)
```





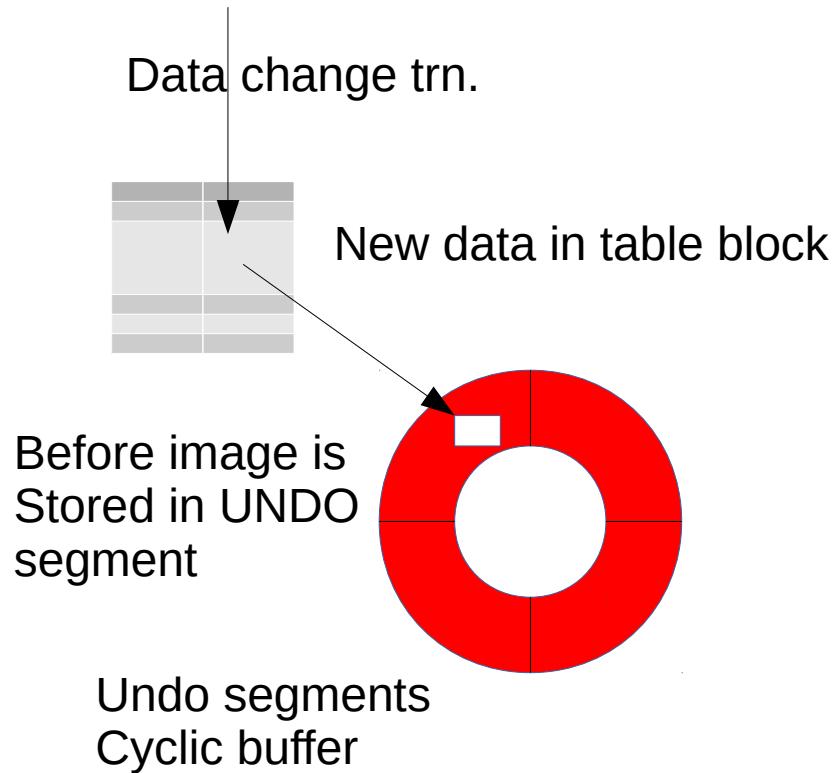
Tablespaces and filesystems

- Tablespaces might reside on different filesystem
 - Outage prevention
 - Data & storage tier life cycle management
 - Online active data on SSD
 - Archive data on rotating disks
 - Tablespace for temporary files – fast might be unprotected storage, no data loss risk
- Wiki page - [File System Layouts](#)



Where is UNDO tablespace?

- Answer: inside the data files
- Is this feature free of charge?
- No, space maintenance (vacuum) is needed to avoid table bloat.



- Postgres manages data consistency using MVCC model (Multiversion Concurrency Control)
 - Transaction isolation for each session
 - Snapshot of data visible to each session based on transaction number
 - Minimize locking contention
 - Readers never blocks writes
 - Serializable snapshot Isolation is available

Online REDO ~ WAL files

- Online REDOlogs are cyclic buffer in Oracle
 - “cleaned up” by archiver process
 - Static amount of redolog Groups each with one or more members within a redolog group
 - Log switch tuning
- WAL – Write Ahead Log files (XLOGs)
 - “cyclic buffer space” with only soft limit in size
 - File reuse – rename already archived file
 - archive_command is used called each time WAL is switched to new file (there is no “archiver” process in postgres)

WAL files

- Place them on separate filesystem
 - Up to 9.4 space requirement for XLOG filesystem
 - **(2 + checkpoint_completion_target) * checkpoint_segments + 1 or checkpoint_segments + wal_keep_segments + 1** files. Each segment file is normally 16 MB.
 - Starting with 9.5
 - **wal_min_size** (default 80MB ~ 5 xlogs) and **wal_max_size** (default 1GB ~ 64 xlogs)
 - Amount of WAL segments between automatic WAL checkpoint (higher values ~ potentially more data for datafiles recovery after server crash)
 - Both configurations are SOFT limit only



Archive_mode and WAL_level

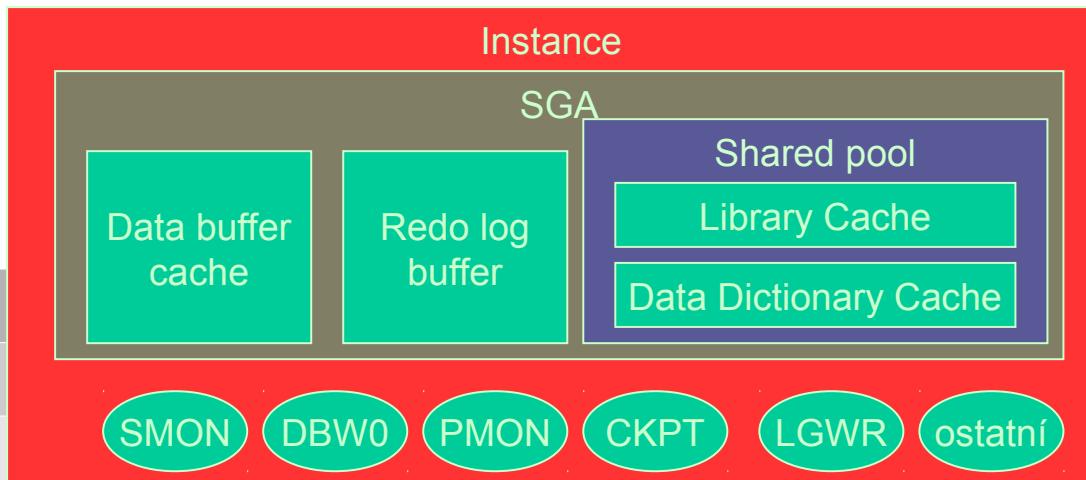
- Archive_mode
 - off, on, always (archive again on streaming replica)
- wal_level
 - minimal
 - Used for crash recovery only ~ Oracle noarchivelog
 - Archive
 - Used for streaming replication ~ Oracle DataGuard
 - hot_standby
 - Used for streaming replication with read only access to replica ~ Oracle Active DataGuard

Memory

Oracle structure:

Parameters relation....

Oracle	Postgres
db_cache_size	shared_buffers
sort_area_size (pga_aggregate_target)	work_mem, temp_buffers
log_buffer_size	wal_buffers
	maintetance_work_mem
	effective_cache_size



Postgres structure:

Nice description can be
found at link:
[PostgreSQL 9.0 Architecture](#)

Per process MEMORY

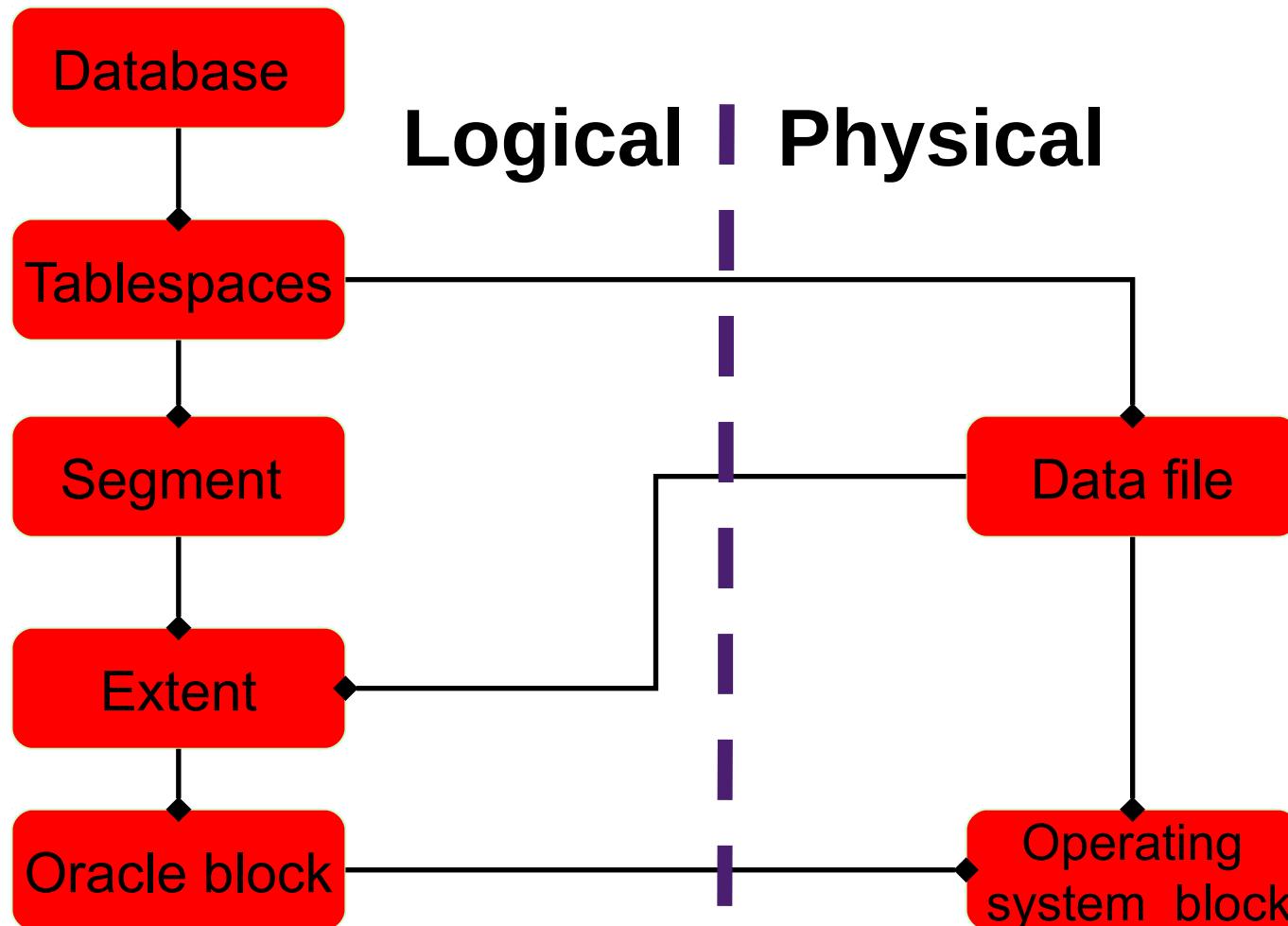
- Work mem
- Maintenance work mem
- Temp buffer
- Catalog cache
- Optimizer/executor

SHARED MEMORY

- shared buffers
- wal buffers
- CLOG buffers
- Locks space
- Other buffers

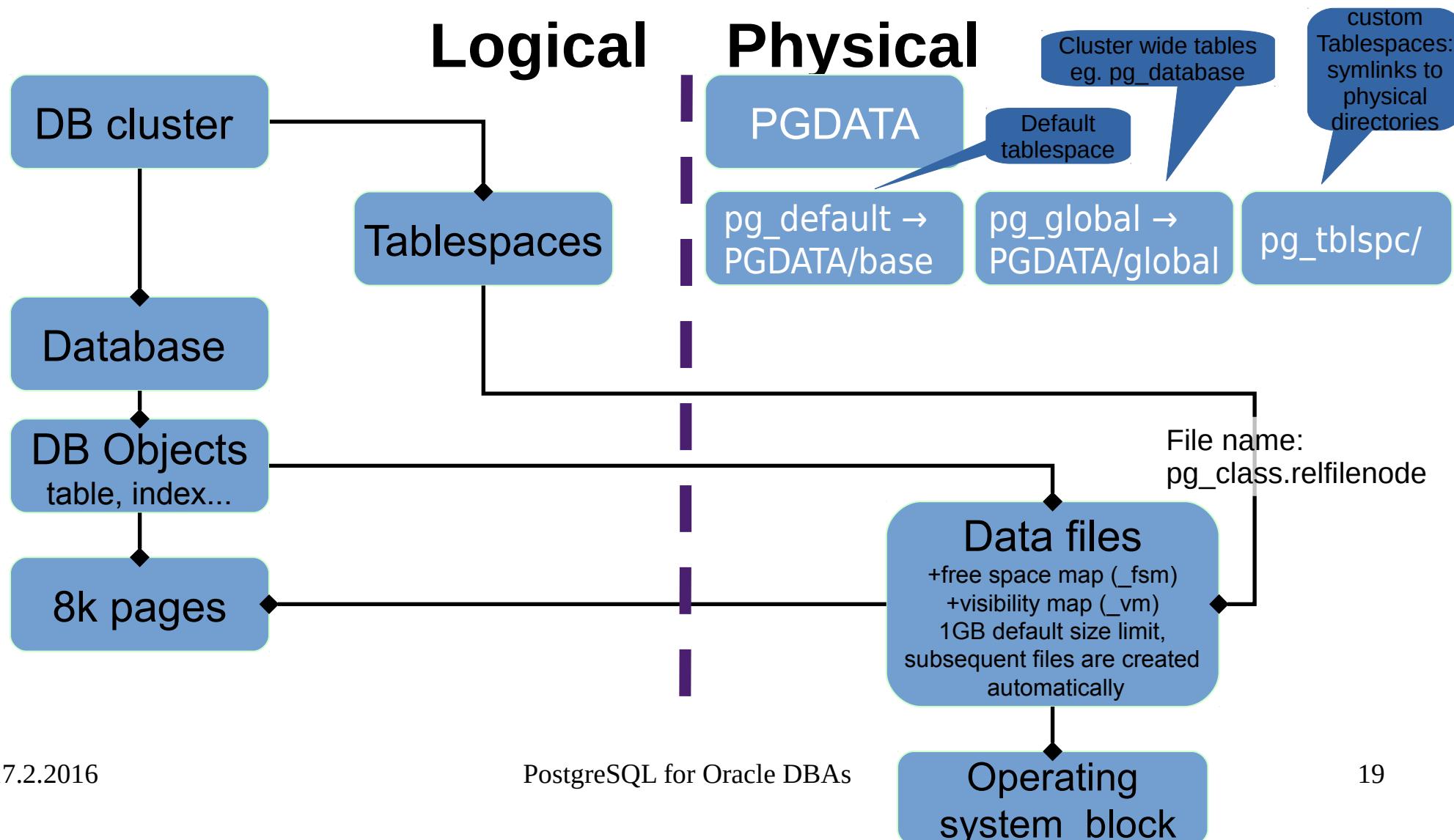
Architecture – database structure

- Oracle database structure (simplified)



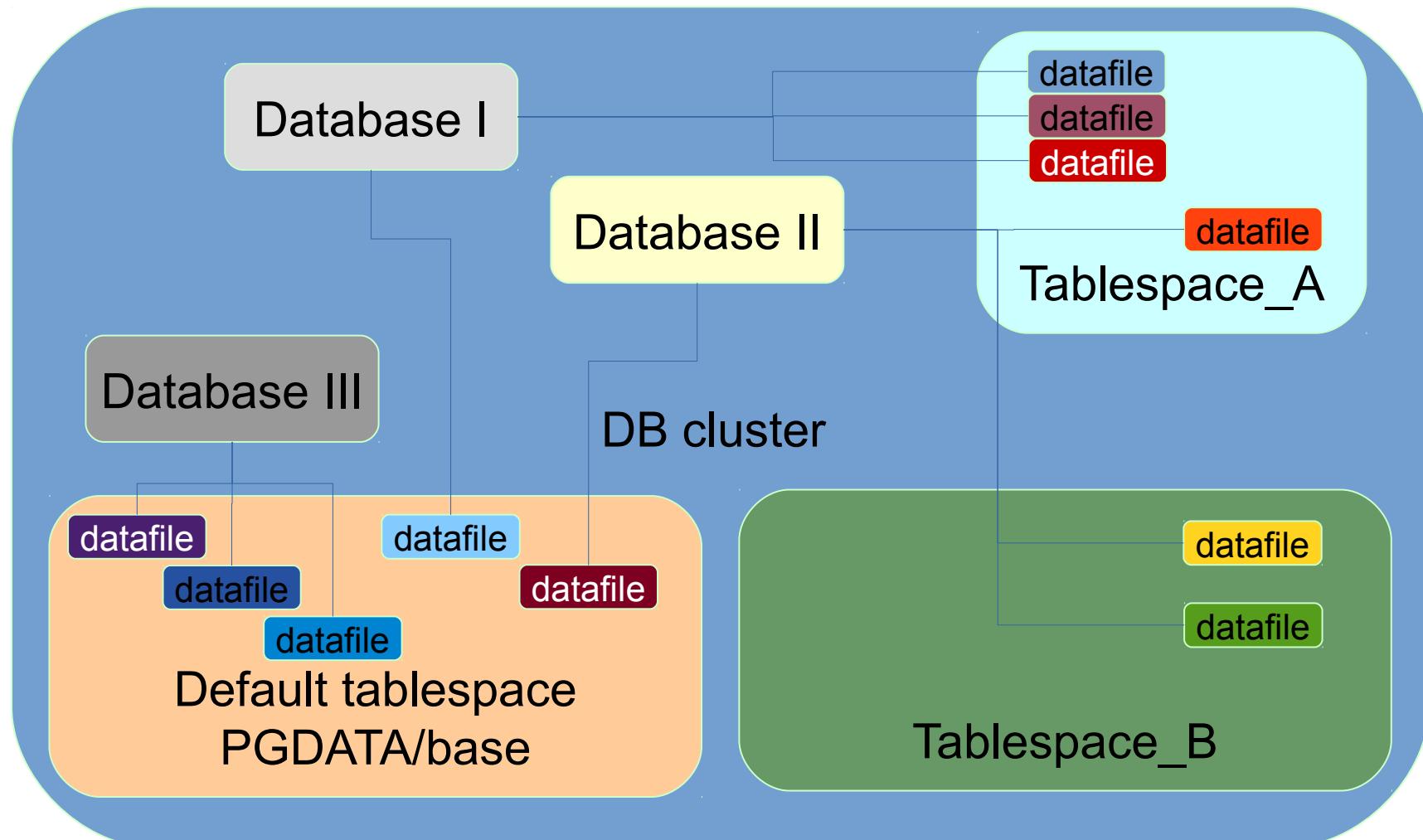
Architecture – database structure

- PostgreSQL database structure (simplified)



Architecture – database structure

- PostgreSQL database structure (simplified)



Architecture - connections

- Oracle
 - Process named **listener** is responsible to handle new connections
 - listener.ora (network restrictions, TCP port)
 - sqlnet.ora (protocol configuration, kerberos...)
 - Dedicated server processes per client
 - Multi-threaded server
 - Always used on Windows
- PostgreSQL
 - Master process **postgres** listens for new connections
 - pg_hba.conf (user/database/network restrictions)
 - postgresql.conf (TCP port, kerberos, RDBMS configuration...)
 - Dedicated server only
 - Shared memory and semaphores are used for inter process synchronization
 - Connection pooling by other products
 - PgBouncer
 - pgpool-II

Architecture notes

- Oracle
 - Decided that RDBMS is right and only place to manage database buffers
 - Promotes its ASM to have a direct control on file management (ASM is kind of LVM dedicated to Oracle)
- PostgreSQL
 - Relies on (believes to) OS file cache management
 - Do not re-implement features already implemented in OS, thus it use file system to store its data files (no RAW device support)

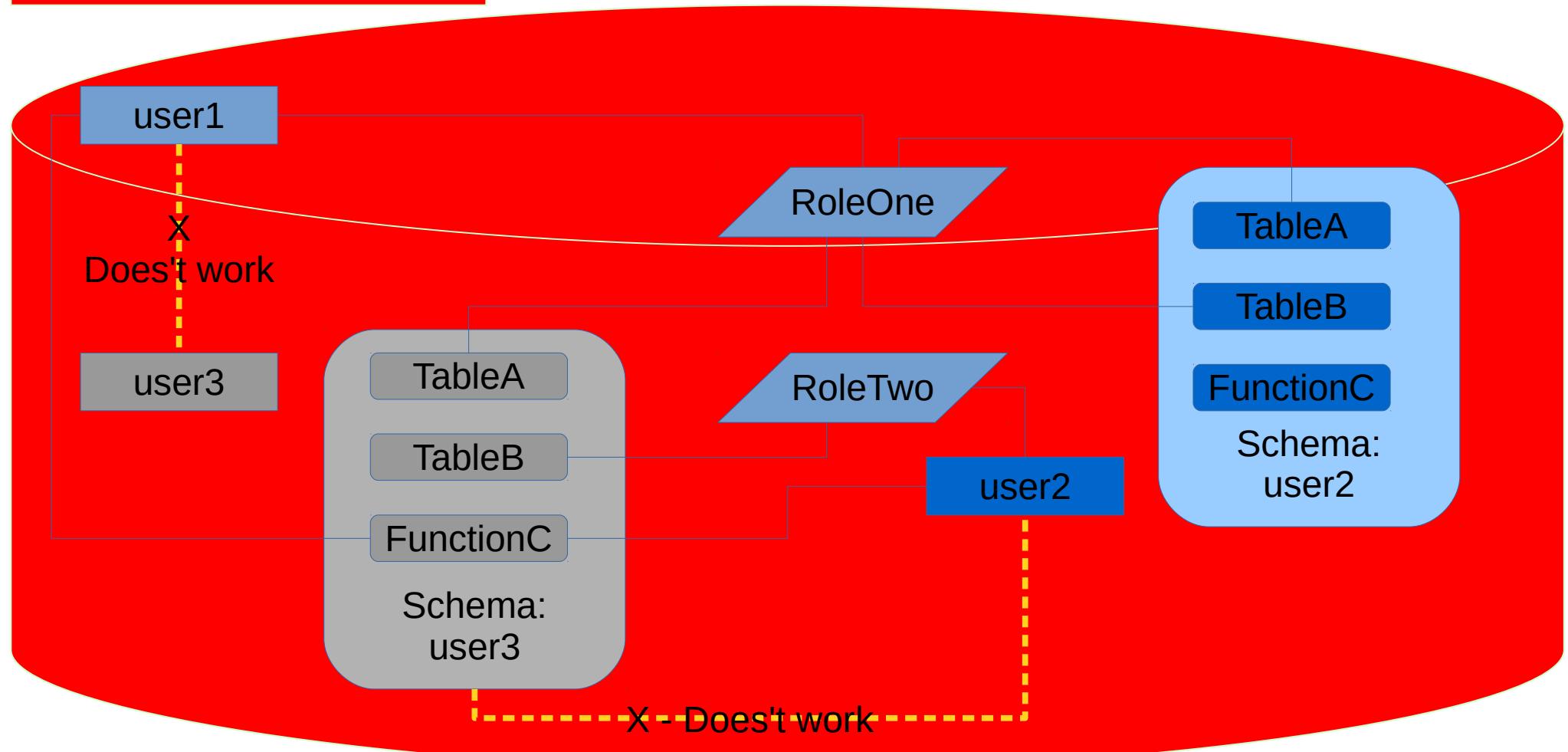
Security observations I

- Oracle has users and roles
 - Users and Roles are defined on DB level (not applies for PDB)
 - Users and Roles are different entities
- Postgres has roles only
 - Some roles might be granted “with login” permission
- Oracle schema consist from a single user objects (schema = user)
 - Schema is not an object, so it can't be granted
- Postgres schema is a grantable name-space object
 - Ownership and usage on schema might be granted to roles
 - Objects owned by different roles (users) might reside within a single schema
 - Public schema might (and should) be dropped



Security observations

SW installation:
`/oracle/product/12.1.0.2/db_1`
OS access control to files



Security observations II

- Oracle tablespace always belongs to a database
 - quotas might be used to limit tablespace usage by users
- Postgres tablespace is defined at cluster level
 - “create” on TS might be granted to a role
 - TS ownership to a role might be defined
 - There are no space usage quotas on tablespace, check FS free space
- Oracle database contains users defined inside DB, there is no database ownership concept
 - Grant scope is always within a database (PDB global users exception exists)
- Postgres database might be owned by a specific role
 - One role might have granted access on objects from multiple databases
 - Role attributes possible in scope of database – alter role XXX se serach_path = YYY,ZZZ in MY_DATABASE

Security observations III

- Oracle distinguish
 - System privileges (create table..., select any ...)
 - Object privileges (grant select on ...)
- Postgres does not have such strong difference
 - Login permission is cluster wide kind of “system” privilege
 - Mostly all privileges are related to some objects including database object itself
 - Grant connect on database myDB
 - Grant usage on ...
 - Grant create on ...

Security observations IV

- Oracle Advanced Security
 - Transparent Data Encryption
 - Kerberos (MS AD integration) is available without Advanced security as of 12.1 release, applies to older releases
 - Many other security features (VPD, RLS...)
- Postgres
 - SSO available
 - Row Security Policies are available with 9.5 release
 - TDE is not available
 - Encryption is covered by separate module pgcrypto

Security observations V

- Oracle remote access control
 - IP address level: sqlnet.ora
 - tcp.validnode_checking = yes
 - tcp.invited_nodes = (hostname1, hostname2)
 - tcp.excluded_nodes = (192.168.10.3)
 - username password and create session is evaluated as next step
- Postgres
 - [pg_hba.conf](#) File
 - username/role_membership, database name, source IP address and authentication method is evaluated prior password validation
 - Password is evaluated as next step

Security observations VI

- Oracle [public] synonyms
 - Synonyms are used to reference another user (schema) objects
 - Might be defined as public – accessible to all users
 - Postgres
 - search_path session environment is used to define scope of visible objects, used similar to PATH in OS
 - Might be defined at cluster level
 - Users might have specified different search path values in particular databases
- ```
ALTER ROLE { role_specification | ALL } [IN
DATABASE database_name] SET
configuration_parameter { TO | = } { value |
DEFAULT }
```

# Security features...

- ALL macro in grant commands
  - Expands to all at time of execution existing objects satisfying grant scope criteria
    - Grant execute on **ALL functions** in schema my\_schema to ...
- Alter default privileges
  - Does not affect existing objects, applied to newly created ones
  - Doc: [ALTER DEFAULT PRIVILEGES](#)

```
ALTER DEFAULT PRIVILEGES
 [FOR { ROLE | USER } target_role
 [, . . .]]
 [IN SCHEMA schema_name [, . . .]]
abbreviated_grant_or_revoke
```

# Backup ... and recovery

- Database [full or partial] dump
  - Oracle exp/imp, expdp/impdp
  - Postgres
    - `pg_dump` / `pg_restore`
      - “directory” format supports parallel dumps
    - `pg_dumpall` (use it for cluster globals only)
      - Load dump by call to `psql`
    - Thanks to MVCC, there is no “ORA-1555” risk during dump
      - For sure, the backup is consistent even if the database is used during the dump



# Binary backups and recovery

- Offline! Works for Oracle, Postgres...
- Online Oracle database backups
  - Manual
    - Alter database (tablespace) begin backup, Copy corresponding datafiles, alter database (tablespace) end backup, store archived redologs needed for recovery
  - **Or use Oracle RMAN utility**
- Online Postgres cluster backup
  - **Backup Control Functions**
    - `pg_start_backup()`, `pg_stop_backup()`, same as above for Oracle [no TS level available]
  - **`pg_basebackup`**
    - Handle calls to backup control functions and might produce copy of postgres cluster or tar archive with the backup. Some features are available like tablespace mapping for convenient backup procedure/higher flexibility



# Binary backups and recovery

- pgBarman
  - Some features similar to oracle RMAN
    - Recovery window / # of copies
    - Stores archived WALs together with Barman backups
    - Backup reports
    - Does not use “rman catalog”, backed up files with some barman metadata files are enough
    - Single backup might be archived to tape (tape integration is not part of pgBarman) – it disappears from backup reports, once retrieved from tape, pgBarman can use the backup again
- pgBackRest
  - More complicated configuration than Barman, incremental backups seems to be implemented slightly better

# HA & DR

- OS clusterware (RHEL Pace Maker, PowerHA...)
  - Simply works
- There is no usable technology like Oracle RAC for PostgreSQL server
  - Sharding (Postgres XL) is not about sharing data files between nodes
  - Oracle 12.2 seems to provide some support for sharding

# Oracle replication

- Oracle DataGuard
  - Log shipping (`log_archive_dest_n`) by archiver
    - `ARCHIVE_LAG_TARGET`
  - Redo transmit by LGWR
    - ASYNC
    - SYNC
    - Delayed recovery - `DELAY=minutes` attribute of the `LOG_ARCHIVE_DEST_n`
  - Logical standby
  - Active Data Guard
  - Golden Gate



# RDBMS replication

- Postgres
  - Log-Shipping Standby Servers
    - archive\_timeout
  - Streaming Replication
    - ASYNC (default)
    - SYNC - Synchronous Replication
    - Standby Server Settings recovery\_min\_apply\_delay available from 9.4
  - Logical Standby
    - Slony, Bucardo, logical decoding framework
  - Hot Standby ( read only accessible standby )
  - BDR provides asynchronous multi-master logical replication.

# Others... I

- psql command line client
  - Comfortable interface, but be aware of default AUTOCOMMIT behavior
- CZ fast reference by Pavel Stěhule
- --data-checksums initdb option
  - Page check-sums are calculated for all object in all databases in cluster
  - use pgbench to verify performance impact
  - Checksum is calculated on page read
  - Backup operate at file level, checksums are not calculated during backup

# Others... II

- Oracle dual table
  - `select function() from dual;`
  - SQL Loader, External tables
  - db links
- PostgreSQL
  - `select function(); select 5/8;`
  - copy command (client side, server side), `file_fdw` for CSV files, format compatible with `COPY` command required
  - Foreign Data Wrappers for many kinds of data sources, including Oracle database



# Others... III

- Porting from Oracle PL/SQL
  - Oracle / Postgres – often similar, not always the same
    - ORA: `trunc(date_variable, format)`
    - PG: `date_trunc('field', source)`
- Pipelined functions are not implemented
- Group by can use column alias in PostgreSQL

```
open2300db=> select date_trunc('hour', rec_datetime) as record_time,
 round(avg(temp_out), 2) as avg_temp,
 max(wind_speed_max) as max_wind_max
 from open2300.weather where
 rec_datetime > now() - interval '3 hour'
 group by record_time
 order by record_time desc;
 record_time | avg_temp | max_wind_max
-----+-----+-----
2016-02-15 22:00:00+01 | 3.04 | 2.4
2016-02-15 21:00:00+01 | 3.23 | 2.5
2016-02-15 20:00:00+01 | 3.66 | 2.3
2016-02-15 19:00:00+01 | 4.11 | 2.6
```

# Others... partitioning

- Postgres partitioning is implemented on top of inheritance feature
  - Declarative partitioning like in Oracle is not available
    - Some basic development for 9.6
- Constraint on child tables
- Trigger on master table
  - Static IF... requires trigger compilation if new child partition tables are added
  - Trigger builds dynamic SQL – more overhead
- No global indexes on partitioned tables



# Others... get table filename

- Bonus link:  
[How to find out which PostgreSQL table a file on disk corresponds to](#)
- Q & A /\* end of slides \*/