PostgreSQL Architecture

Surendra Panpaliya

PostgreSQL Architecture

POSTGRESQL IS

POWERFUL OPEN-SOURCE

RELATIONAL DATABASE MANAGEMENT SYSTEM

PostgreSQL Architecture

Known for

Robustness,

Extensibility

Standards compliance

PostgreSQL Architecture

Understanding

its architecture

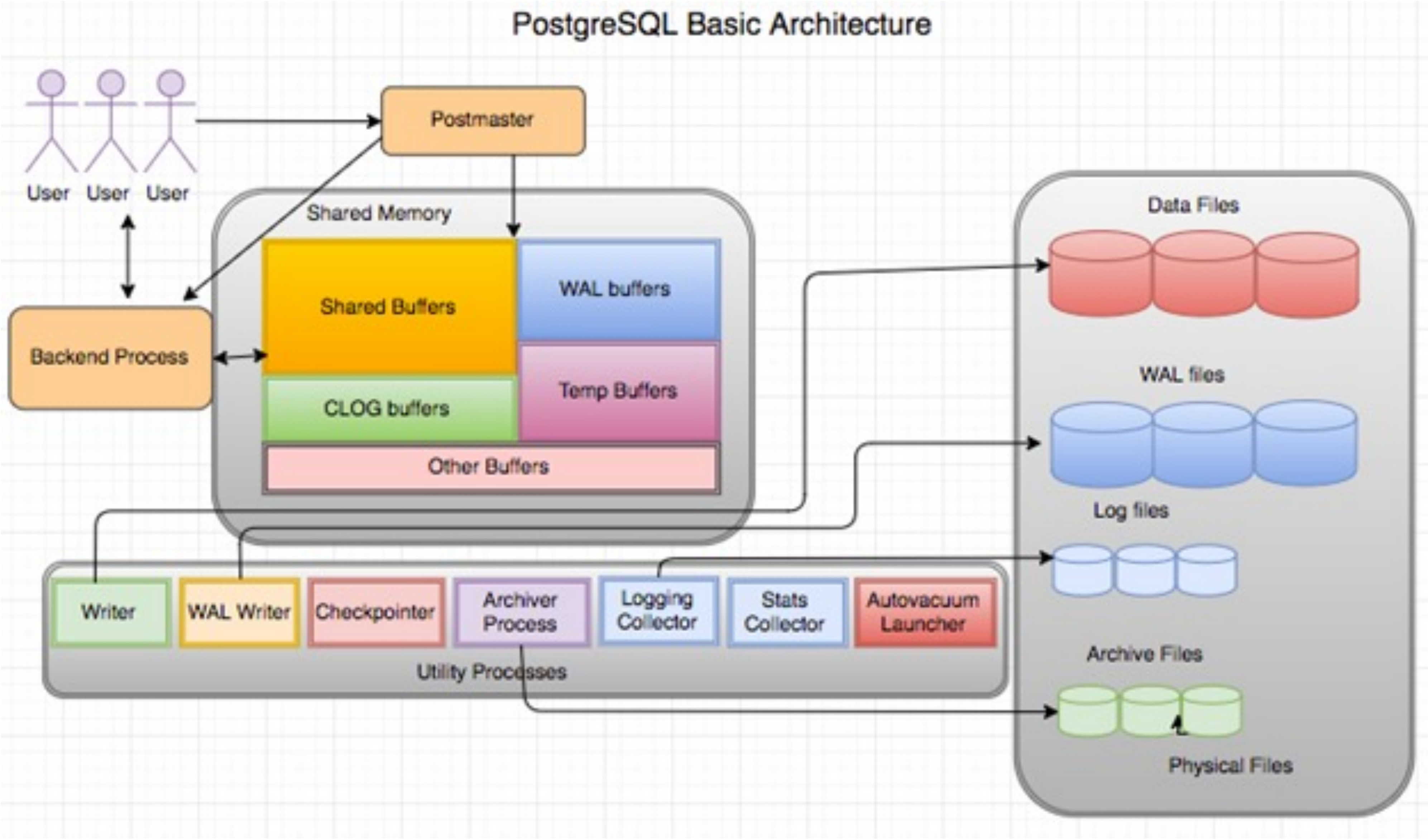
is crucial for effective

Database management

Optimization

PostgreSQL Architecture Components

Surendra Panpaliya


PostgreSQL Basic Architecture
User
User
User
Postmaster
Backend Process
Shared Memory
Shared Buffers
WAL buffers
CLOG buffers
Temp Buffers
Other Buffers
Writer
WAL Writer
Checkpointer
Archiver Process
Logging Collector
Stats Collector
Autovacuum Launcher
Utility Processes
Data Files
WAL files
Log files
Archive Files
Physical Files

PostgreSQL Instance

CONSISTS OF SET OF PROCESS AND MEMORY.

POSTGRESQL USES A SIMPLE

“PROCESS PER-USER”

CLIENT/SERVER MODEL.

Postmaster Process

- Main PostgreSQL server process
- Responsible for Initializing the database
- Handling connections
- Starting other background processes

Postmaster Process

LISTENS FOR INCOMING CONNECTIONS

SPAWNS NEW SERVER PROCESSES

FOR EACH CONNECTION.

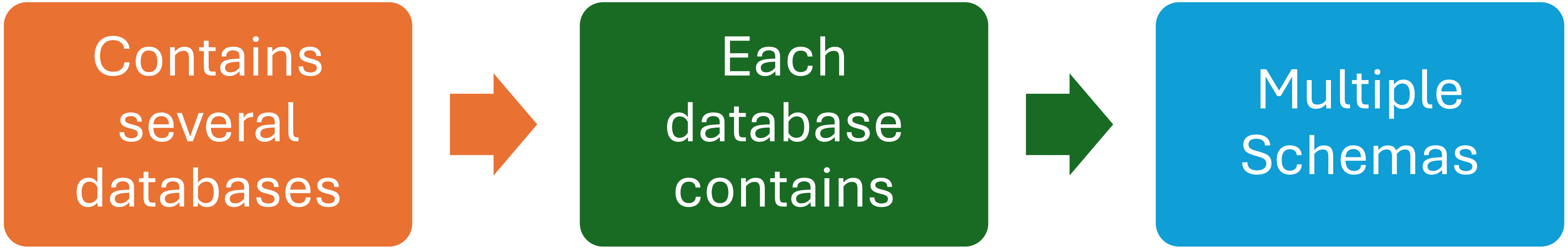
Database Cluster

A COLLECTION OF DATABASES

MANAGED BY A SINGLE

POSTGRESQL SERVER INSTANCE.

Database Cluster



Contains several databases

Each database contains

Multiple Schemas

Shared Memory

- Shared Buffers
- WAL Buffers
- Work_mem
- Maintenance_work_mem

Shared Buffers

Main memory area

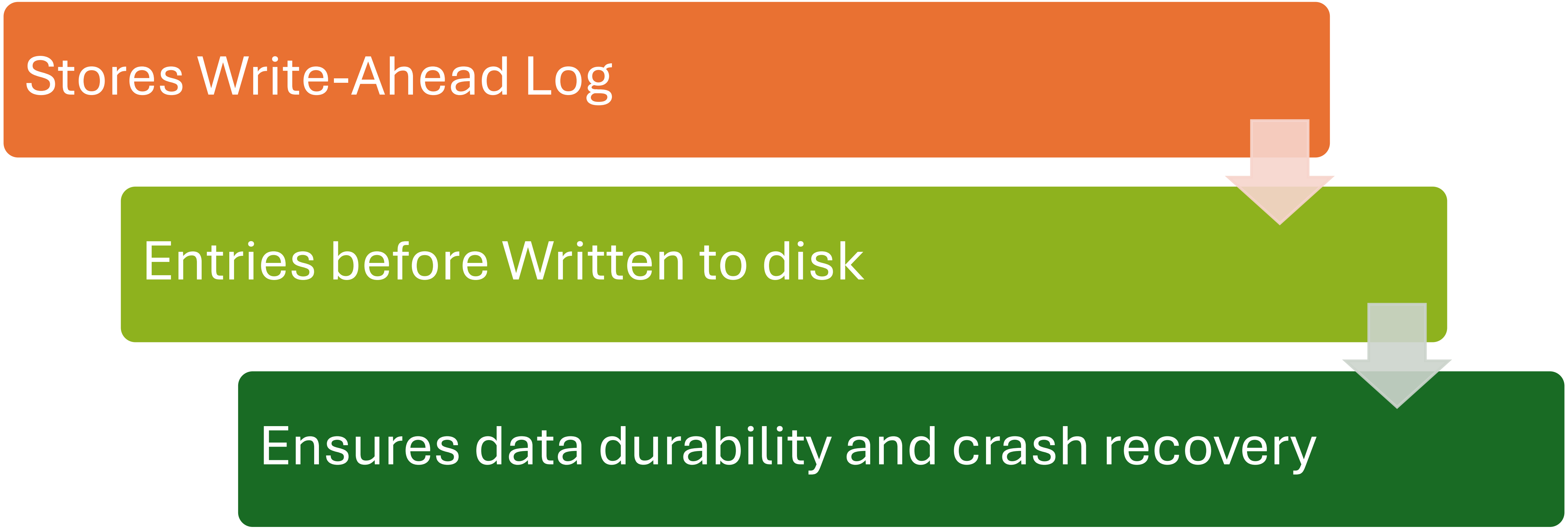
PostgreSQL Caches

Data Blocks Read from the disk.

Improves performance

by reducing disk I/O

WAL Buffers

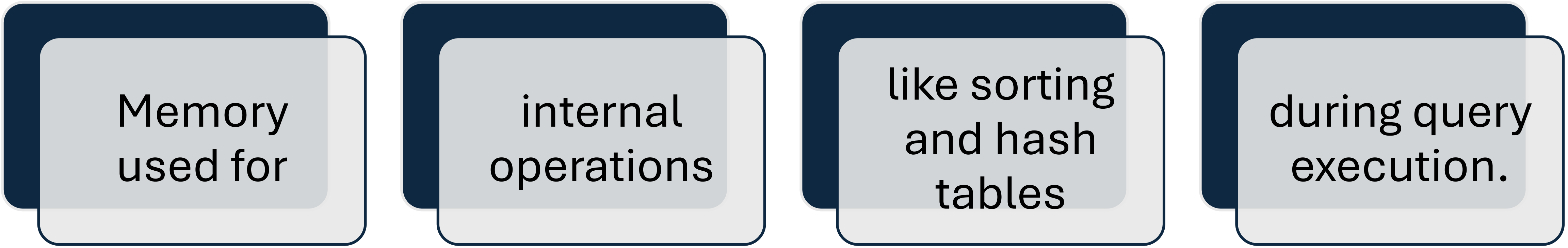


Stores Write-Ahead Log

Entries before Written to disk

Ensures data durability and crash recovery

Work_mem



Memory used for

internal operations

like sorting and hash tables

during query execution.

Maintenance_work_mem

Memory allocated

Maintenance tasks

VACUUM, CREATE INDEX

ANALYZE operations.

Background Processes

- Background Writer (bgwriter)
- WAL Writer (walwriter)
- Autovacuum Daemon
- Stats Collector

Background Writer (bgwriter)

Periodically writes dirty pages

from the shared buffers to the disk

To ensure the buffer cache

has space for new pages.

WAL Writer (walwriter)

WRITES WAL DATA

FROM WAL BUFFERS

TO THE DISK.

Autovacuum Daemon

MANAGES AUTOMATIC VACUUMING OF

DATABASE TO RECLAIM STORAGE

MAINTAIN TABLE STATISTICS.

Stats Collector

Collects and aggregates

Database activity

Performance statistics.

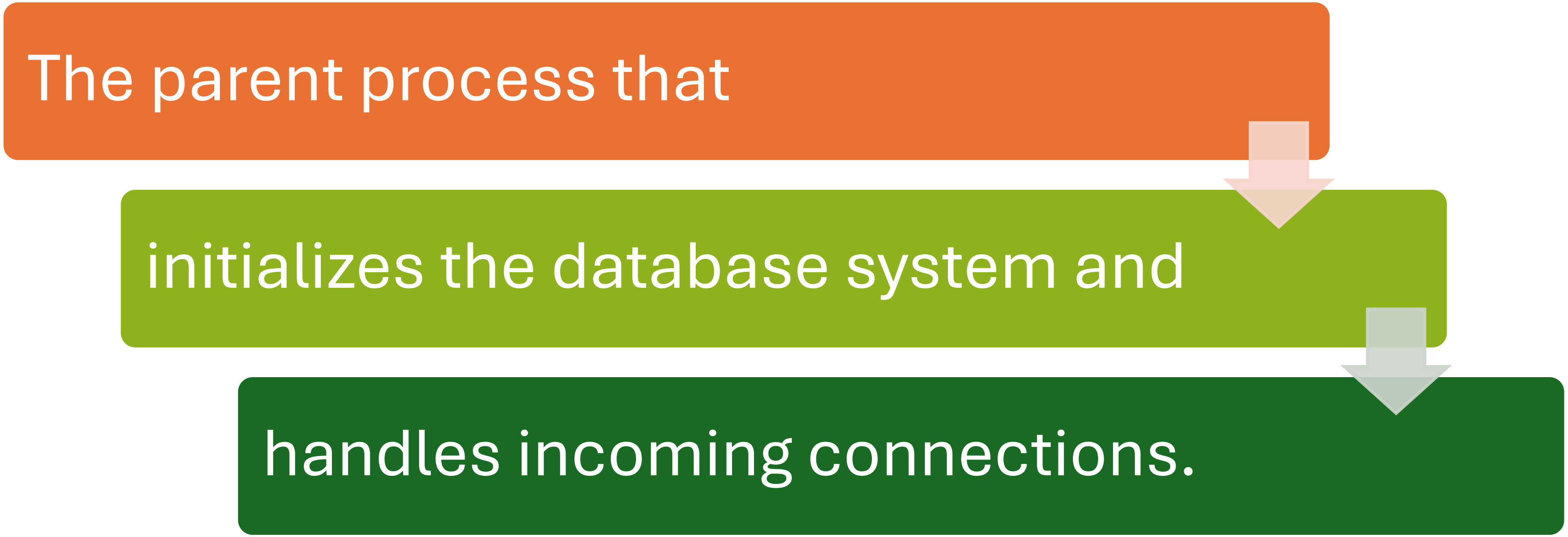
Process Architecture

POSTMASTER PROCESS

BACKEND PROCESSES

BACKGROUND WORKER PROCESSES

Postmaster Process



The parent process that

initializes the database system and

handles incoming connections.

Backend Processes

Individual server processes

Spawned by the Postmaster

To handle client connections.

Backend Processes

EACH CONNECTION

TO THE DATABASE

HAS A CORRESPONDING

BACKEND PROCESS

Background Worker Processes

Custom or built-in

background jobs

perform specific tasks,

such as parallel queries.

Storage

- Data Directory
- Data Files
- WAL Files
- Configuration Files

Data Directory

File system directory

Stores database files

Configuration files

Other metadata

Data Files

- Store actual
- Database tables
- Indexes

WAL Files

Logs of all changes

Made to the database

Used for

Crash Recovery and Replication

Configuration Files

Key configuration files include

postgresql.conf (main configuration)

pg_hba.conf (client authentication)

pg_ident.conf (user identity mapping)

Storage

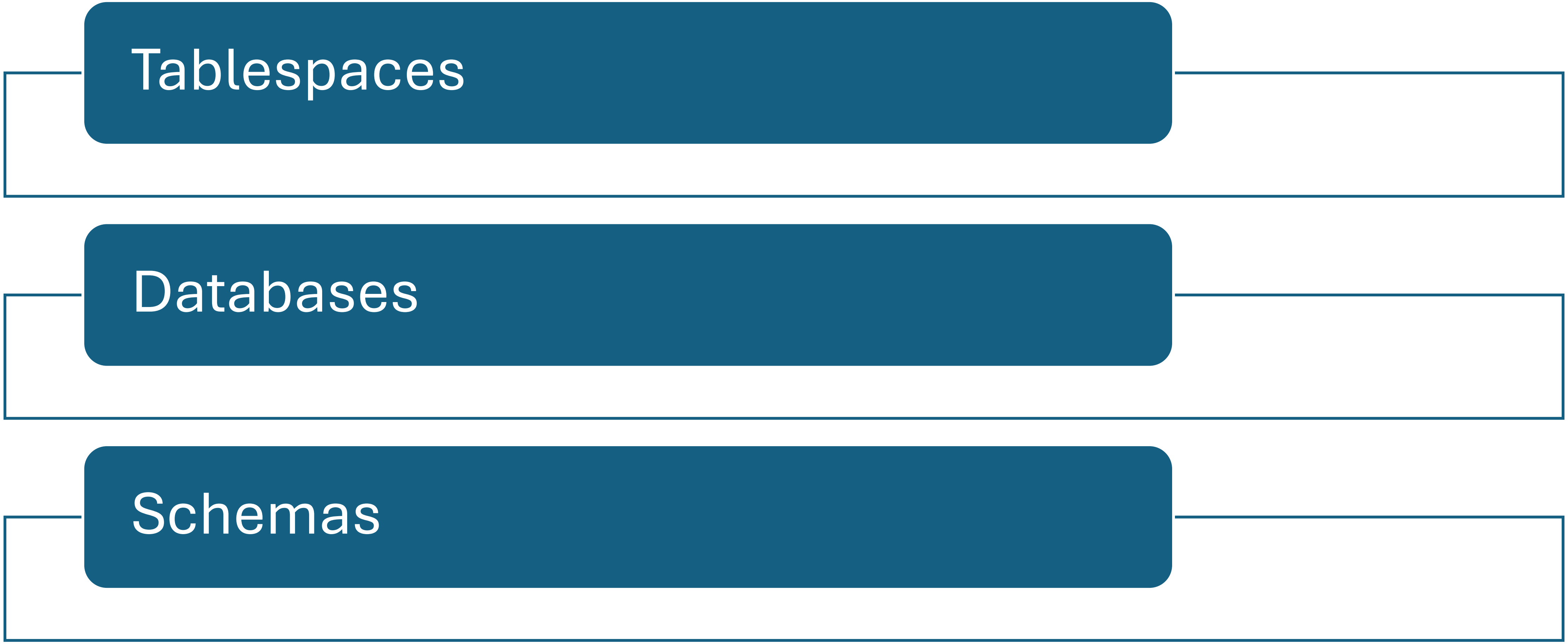
Data Directory: The file system directory where PostgreSQL stores its database files, configuration files, and other metadata.

Data Files: Files that store actual database tables and indexes.

WAL Files: Logs of all changes made to the database. Used for crash recovery and replication.

Configuration Files: Key configuration files include postgresql.conf (main configuration), pg_hba.conf (client authentication), and pg_ident.conf (user identity mapping).

Logical Storage Structures



Tablespaces

Databases

Schemas

Tablespaces

- Logical storage units
- Map to file system locations
- Allow for the distribution of data
- Across different storage locations

Databases

LOGICAL COLLECTIONS OF

SCHEMAS, TABLES, INDEXES

FUNCTIONS, AND

OTHER OBJECTS.

Schemas

NAMESPACE WITHIN A DATABASE

CONTAINS TABLES, VIEWS, FUNCTIONS

SCHEMAS HELP ORGANIZE OBJECTS LOGICALLY

Concurrency Control

MVCC (MULTI-VERSION CONCURRENCY CONTROL)

ENSURES DATA CONSISTENCY AND ISOLATION

BY MAINTAINING MULTIPLE VERSIONS OF DATA ROWS

MVCC

Allows concurrent transactions

to read and write

without blocking each other

using a mechanism called snapshots.

Indexing

VARIOUS INDEXING METHODS ARE AVAILABLE

B-TREE, HASH, GIST,

SP-GIST, GIN, AND BRIN,

HELP OPTIMIZE QUERY PERFORMANCE

Replication and High Availability

Streaming Replication

Logical Replication

Streaming Replication

Provides real-time

data replication

to standby servers

for high availability and

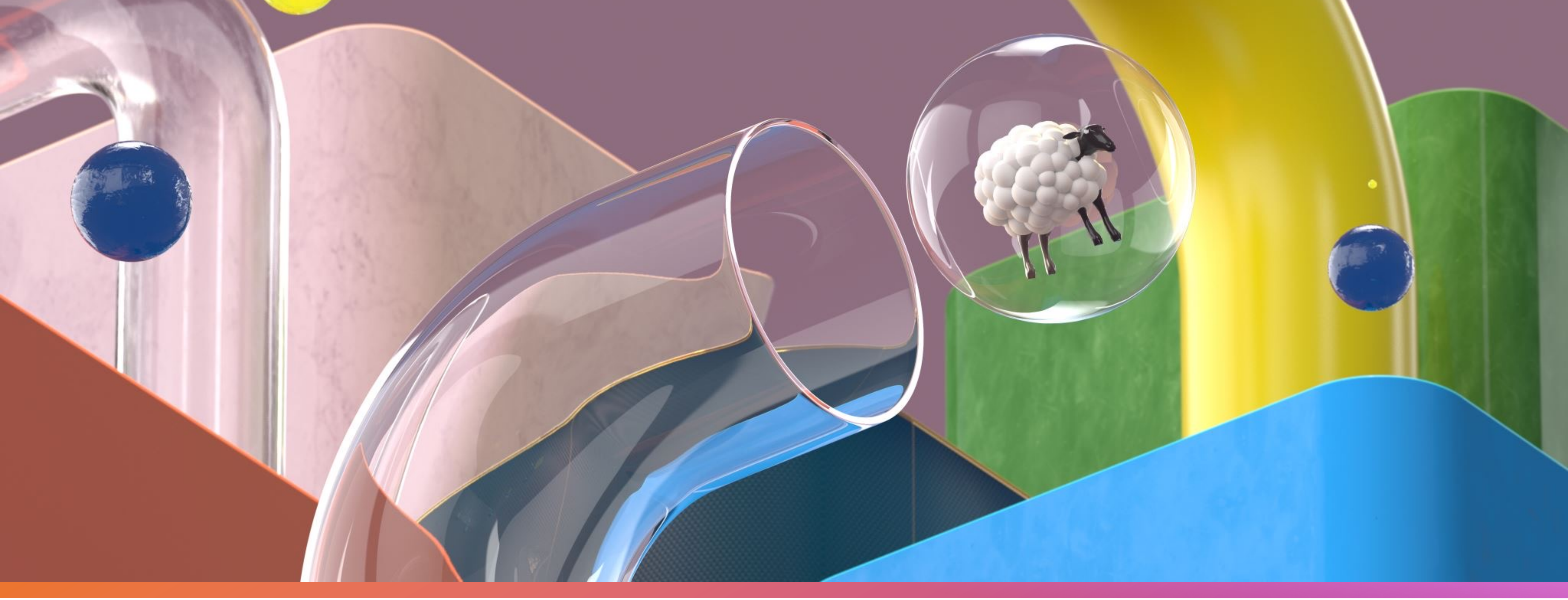
disaster recovery.

Logical Replication

Allows selective replication of

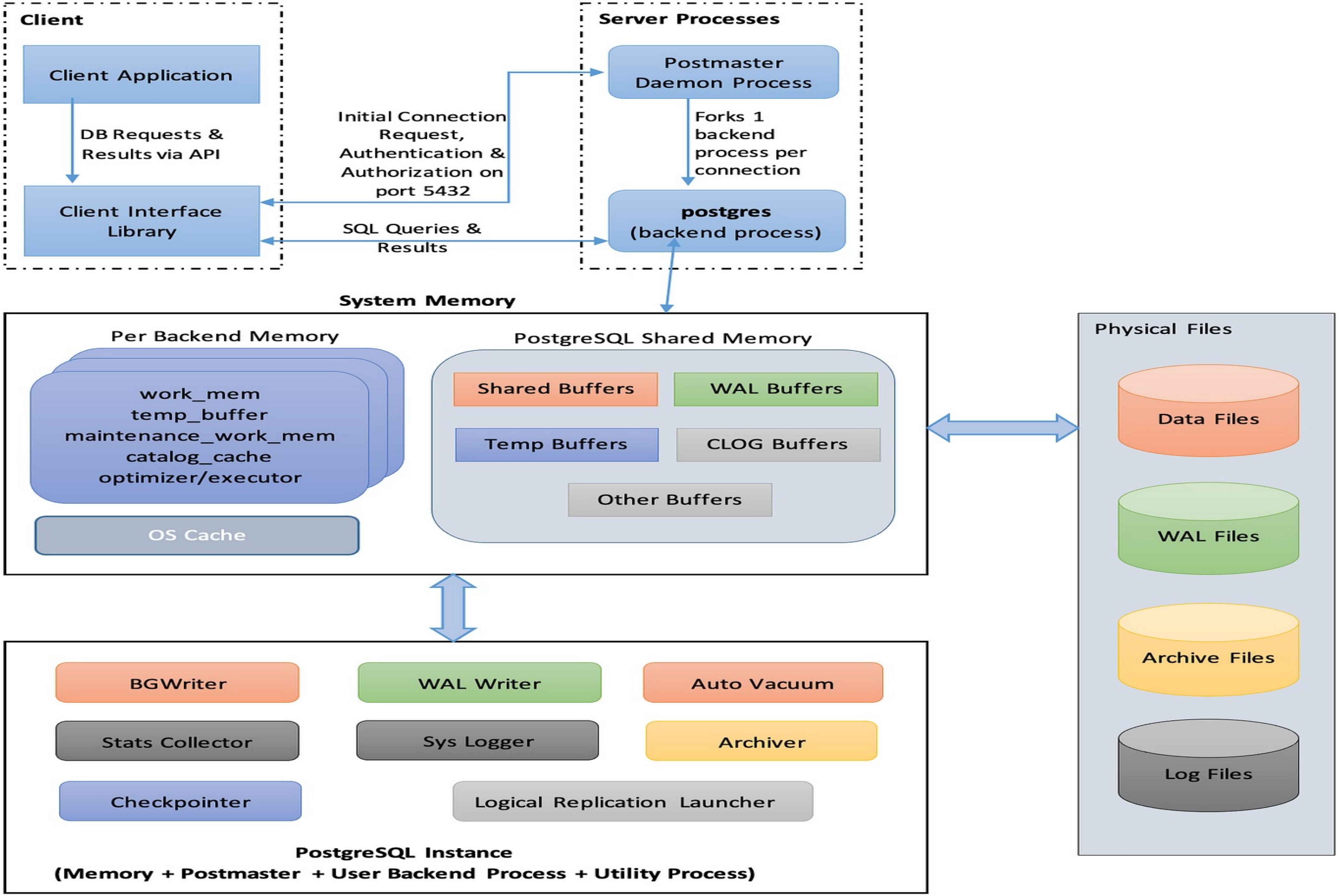
data changes based on

user-defined configurations



PostgreSQL Process Flow

Surendra Panpaliya


Client
Client Application
DB Requests & Results via API
Client Interface Library
Initial Connection Request, Authentication & Authorization on port 5432
SQL Queries & Results
Server Processes
Postmaster Daemon Process
Forks 1 backend process per connection
postgres (backend process)
System Memory
Per Backend Memory
work_mem
temp_buffer
maintenance_work_mem
catalog_cache
optimizer/executor
OS Cache
PostgreSQL Shared Memory
Shared Buffers
WAL Buffers
Temp Buffers
CLOG Buffers
Other Buffers
Physical Files
Data Files
WAL Files
Archive Files
Log Files
BGWriter
WAL Writer
Auto Vacuum
Stats Collector
Sys Logger
Archiver
Checkpointer
Logical Replication Launcher
PostgreSQL Instance
(Memory + Postmaster + User Backend Process + Utility Process)

Client Connection

When a client connects to PostgreSQL

Postmaster process accepts the connection

Spawns a new backend process to handle it

Query Processing

Parser

Planner/Optimizer

Executor

Parser

Converts

SQL queries into

Parse tree

Planner/Optimizer

Converts the parse tree

into an execution plan

optimizing for performance

Executor

Executes the plan

By retrieving

Manipulating data as needed.

Buffer Management

Data is fetched from the disk

into the shared buffers.

Buffer Management

ENSURES

FREQUENTLY ACCESSED DATA

CACHED IN MEMORY

Transaction Management

ENSURES

ACID PROPERTIES

THROUGH

MVCC AND WAL

Background Tasks

PROCESSES

LIKE

BACKGROUND WRITER.

Background Tasks

Autovacuum daemon

Ensure data integrity

Optimal performance

by Managing

Memory and storage

Summary

PostgreSQL Architecture

designed for reliability,

performance, and

extensibility.

Summary

Uses a combination of

shared memory and process-based management

to handle client connections,

perform query execution

manage data storage.

Summary

Its robust concurrency control,

indexing methods, and replication mechanisms

Make it suitable for

various applications,

including large-scale enterprise systems.

Thank you for your support and patience

Surendra Panpaliya

Founder and CEO

GKTCS Innovations

https://www.gktcs.com