2.1.2.1 Database Solution Stack

Return to DIDO CLI Background

At the heart of Database Solution Stack is a Database Platform. In this case the Database platform is one of the many RDBMS products, such as Oracle, PostgreSQL, MySQL, SQLServer etc. However, in this stack, the DBMS does not necessarily need to be an RDBMS as long as there is an interface to the database that uses SQL. The Boundaries of the Database Platform are not rigid. For example, Microsoft offers a single ODBC interface for many databases, consequently the ODBC driver may or may not be part of the Database Platform. However, in the big picture what is in the Platform and what is not in the platform is a bit pedantic.

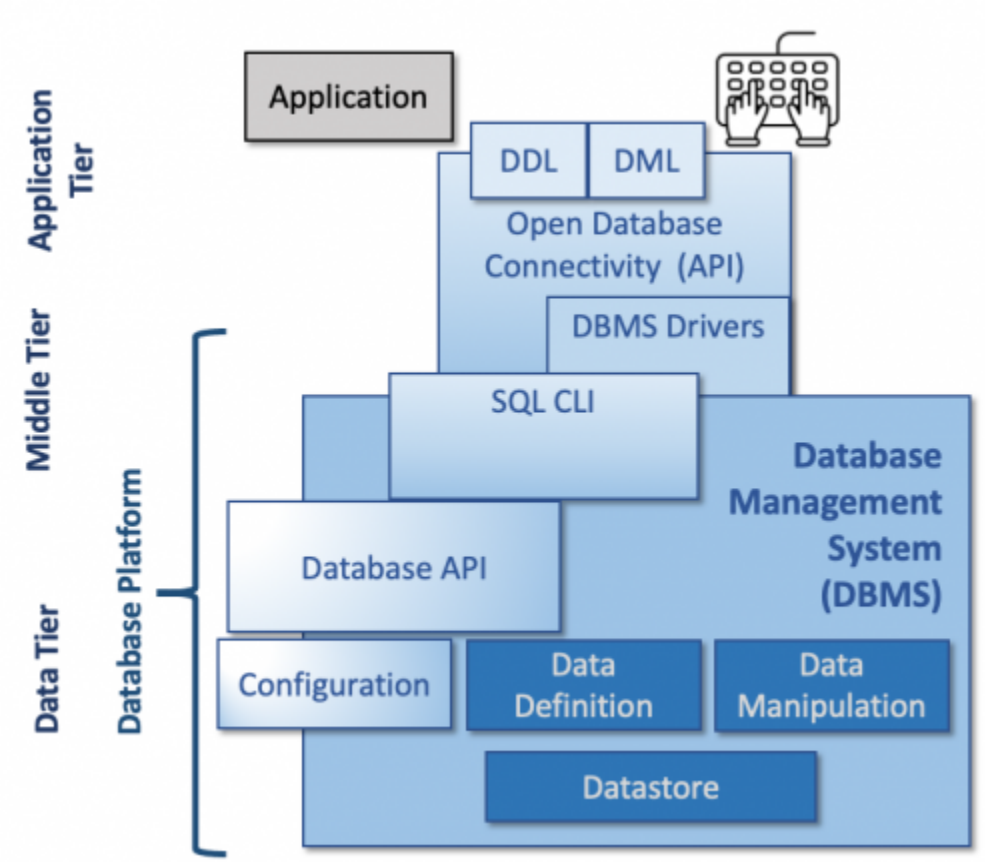


Figure 1: Typical Mature Database “stack”

The following is a brief description of each component in the DBMS Stack. Note, these components are normative in nature and each DBMS may be slightly different. On the left side of the diagram there is a key that describes which tier each component is generally found in.

- **Application** - The application is the application, program, utility, service or microservice that needs to interact with the RDBMS. This is done using standardized Data Definition Language (DDL) or Data Manipulation Language textual commands that adhere to the SQL specifications See: dblang-sql-part1 and related specifications.
- **Users** - And end user can enter SQL compliant DDL or DML text commands into a terminal or even produce scripts that contain as series of commands that sent to the Open Database Connectivity

[dido:public:ra:xapend:xapend.a_glossary:a:api|API]] for processing. Just as with the Application, these commands need to be compliant with the SQL Standard See: dblang-sql-part1.

- **Open Database Connectivity API** - This a utility (i.e., usually a library) that the application calls to process the SQL commands and check for validity and verification that the commands are correct. Examples of an Open Database Connectivity. See Open Database Connectivity (ODBC) and Java Database Connectivity(JDBC).
- **Data Definition Language (DDL)** - The Data Definition Language (DDL) is used to create and modify the structure of database objects in a database. These database objects include views, schemas, tables, indexes, etc. Often, the DDL commands require a differen set of privileges than the Data Manipulation Language (DML).
- **Data Manipulation Language (DML)** - The Data Manipulation Language (DML) includes commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables, modifying existing data and associated data from different tables together. DML is mostly incorporated in SQL databases.
- **DBMS Drivers** - Database Driver is s a computer program that implements a protocol (Open Database Connectivity (ODBC) or Java Database Connectivity(JDBC)) for a database connection.
- **Database Management System** - DataBase Management System (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.
    - **SQL Command Line Interpreter (CLI)** - The SQL Command Line Interpreter translates the tokenized SQL commands into DB Specific instructions.
    - **Database API** - DBMSs generally support APIs that allow a programmer to directly access a databases. For example, Oracle provides a Oracle C++ Call Innterface (OCCI) https://docs.oracle.com/en/database/oracle/oracle-database/18/adobj/oracle-c-cpp-call-interface-odci.html. PostgreSQL supports libpq++ as the C++ API to Postgres https://www.postgresql.org/docs/7.0/libpqplusplus.htm
    - **DB Configuration** - There are always two aspects to configuring a DBMS. The first is setting up the environment externally to the DBMS (i.e., downloading the software, running a wizard to install and configure the DB, etc).
    - **Data Definition** - A Data Definition Language (DDL) is a computer language used to create and modify the structure of database objects in a database. These database objects include views, schemas, tables, indexes, etc.
    - **Data Manipulation** - A Data Manipulation Language (DML) is a family of computer languages including commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.
    - **Datastore** - A data store is a repository for persistently storing and managing collections of data which include not just repositories like databases, but also simpler store types such as simple files, emails etc. … A database is a series of bytes that is managed by a database management system (DBMS).https://en.wikipedia.org/wiki/Data_store

Database Solution Stack Scenarios

Return to Top Figure 2 represents three different scenarios of interactions of Applications or Users to the Database Solution Stack.

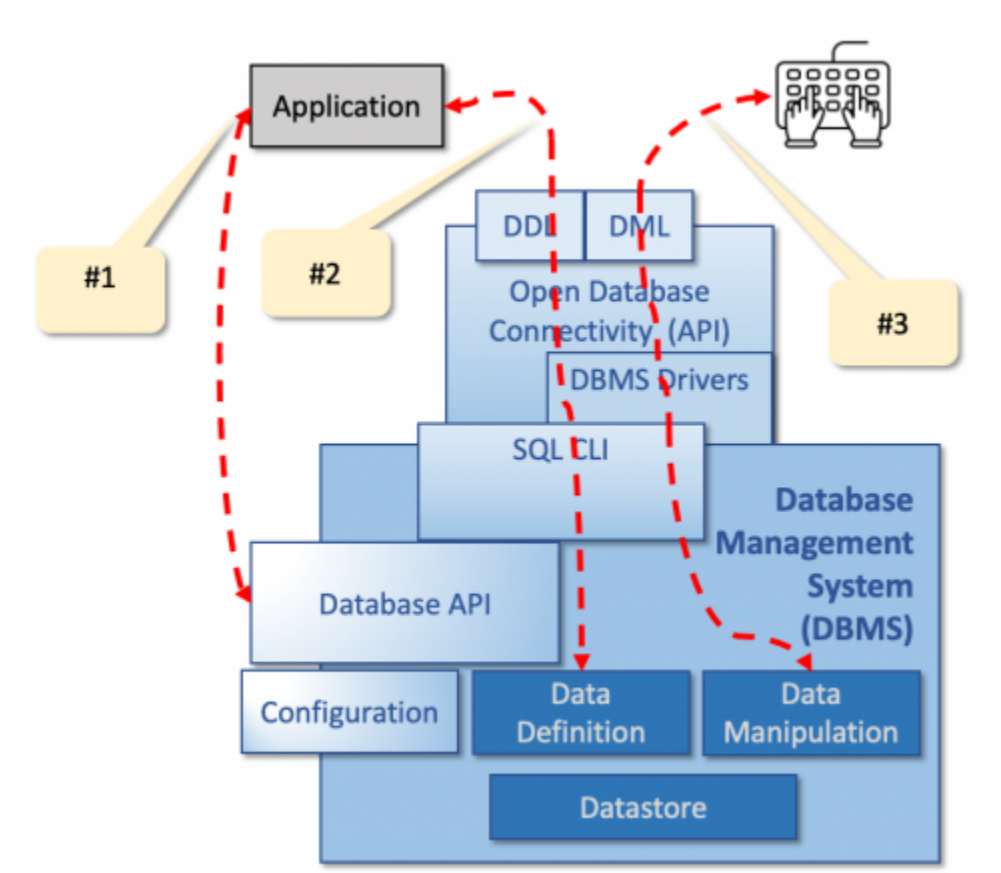


Figure 2: Three different User Scenario Paths through the Database Solution Stack.

Scenario #1

In this scenario, the Application accesses the Database Platform using the Database provided API. This solution is very machine resource efficient since the Application uses code that is optimized to access a specific Database (i.e., Oracle, PostgreSQL, MySQL, SQLServer, etc.). However, all of the error recovery and testing for the code and any changes made to the underlying database schema rests on the application.

Note: This scenario makes it more difficult to migrate from one Database Platform to another (sometimes referred to as vendor lock-in). In order to access the Database, the end user must use the Application.

Scenario #2

In this scenarios, an Application or the End User can create SQL Statements as strings and pass

them through Open Database Connectivity (ODBC) or Java Database Connectivity(JDBC) to access the database. In this scenario, the application or the End User are accessing the Database to invoke Data Definition Language (DDL) functionality for creating, destroying or modifying the database objects defined in the schema (i.e., views, schemas, tables, indexes, etc.).

Note: This Scenario usually relies on a specific Database Driver to access the Database Platform (i.e., a Driver for Oracle, PostgreSQL, MySQL, SQLServer, etc.).

Scenario #3

This scenario is very similar to Scenario #2 except this time the Application or End User are using Data Manipulation Language (DML) for inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data.

Note: This Scenario usually relies on a database Database Driver specific to access the Database Platform (i.e., a Driver for Oracle, PostgreSQL, MySQL, SQLServer, etc.).

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:s_cli:05_contents:01_prt:02_basics:02_solstack:dbstack:start&rev=1627666307

Last update: **2021/07/30 13:31**