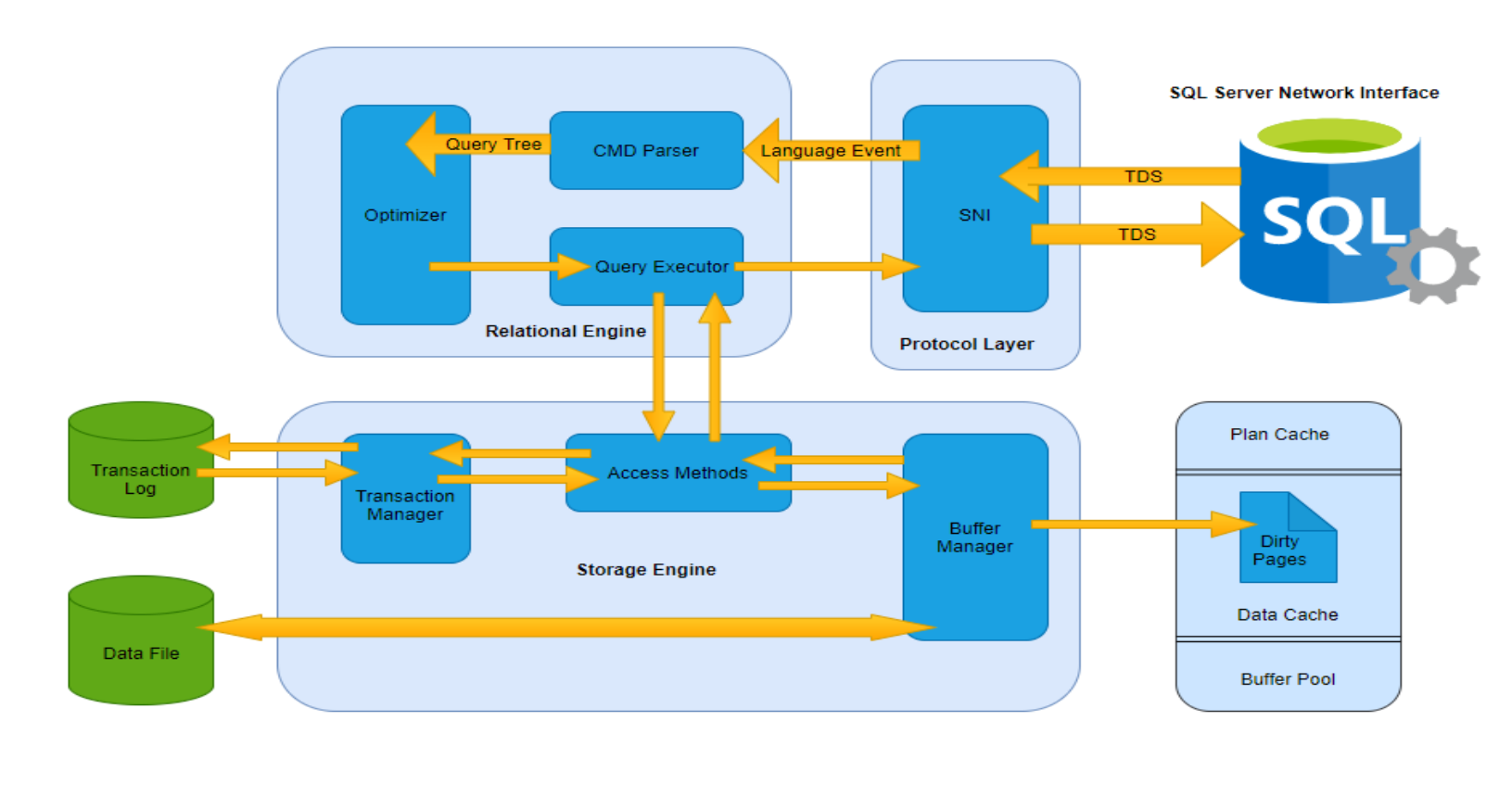
MSSQL Architecture

SQL Server Architecture

Three major components

- Protocol Layer.
- Relational Engine.
- Storage Engine.

SQL Server Architecture



Protocol Layer (Server Name Identification- SNI)

MS SQL SERVER PROTOCOL LAYER supports 3 protocols and all three use TDS which stands for Tabular Data Stream.

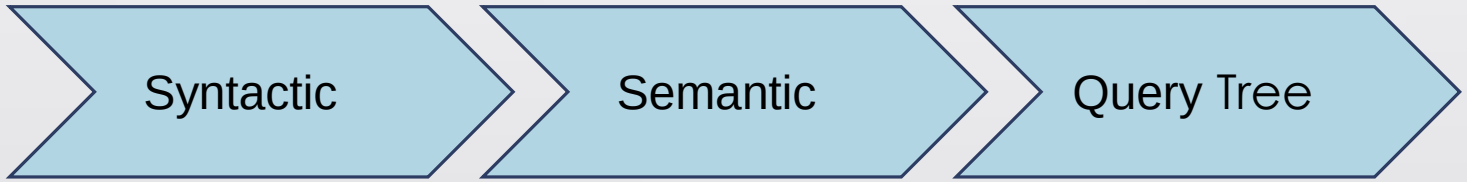
1. Shared Memory : CLIENT and MS SQL server run on the same machine. Both can communicate via Shared Memory protocol.
2. TCP/IP : MS SQL SERVER provides the capability to interact via TCP/IP protocol, where CLIENT and MS SQL Server are remote to each other and installed on a separate machine.
3. Named Pipes : MS SQL SERVER provides the capability to interact via the Named Pipe protocol. Here the CLIENT and MS SQL SERVER are in connection via LAN.

Relational Engine

- The Relational Engine is also known as the Query Processor.
- It has the SQL Server components that determine what exactly a query needs to do and how it can be done best.
- It is responsible for the execution of user queries by requesting data from the storage engine and processing the results that are returned.
- Three Major Components:
  - CMD Parser
  - Optimizer
  - Query Executer

CMD Parser

- Data once received from Protocol Layer is then passed to Relational Engine. **"CMD Parser"** is the first component of Relational Engine to receive the Query data.
- The principal job of CMD Parser is to check the query for **Syntactic and Semantic error.**
- Finally, it **generates a Query Tree**.

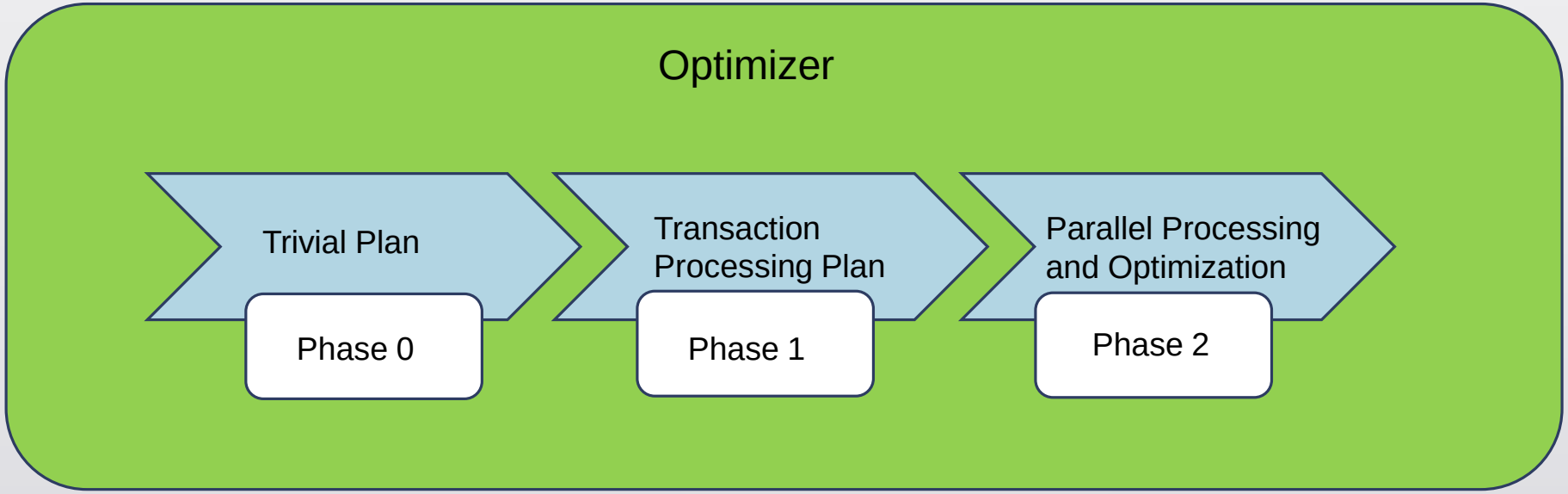


Optimizer

- The work of the optimizer is to create an execution plan for the user's query.
- This is the plan that will determine how the user query will be executed.
- Optimization is done for DML (Data Modification Language) commands like SELECT, INSERT, DELETE, and UPDATE.
- DDL commands like CREATE and ALTER are not optimized, but they are instead compiled into an internal form.
- The query cost is calculated based on factors like CPU usage, Memory usage, and Input/ Output needs.
- Optimizer's role is to find the cheapest, not the best, cost-effective execution plan.

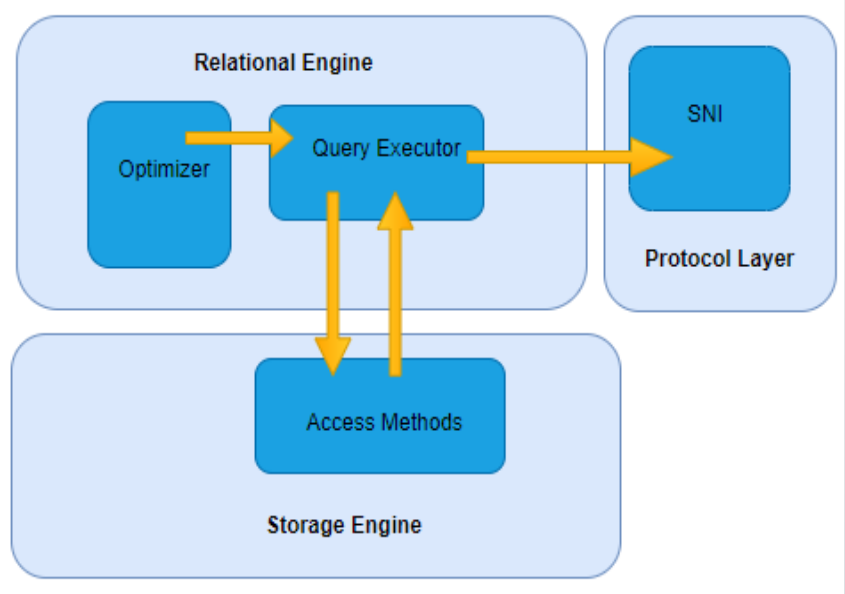
Optimizer (contd)

- Phase 0 : Only one practical, workable plan knows as trivial plan.
- Phase 1 : Search for Simple and Complex Plan.
- Phase 2 : Optimizer searches for Parallel Processing possibilities. If not possible , the final optimization aim is finding all other possible options for executing the query in the best way.



Query Executor

- Query executer calls **Access Method.**
- It provides an execution plan for data fetching logic required for execution.
- Once data is received from Storage Engine, the result gets published to the Protocol layer.
- Finally, data is sent to the end user.

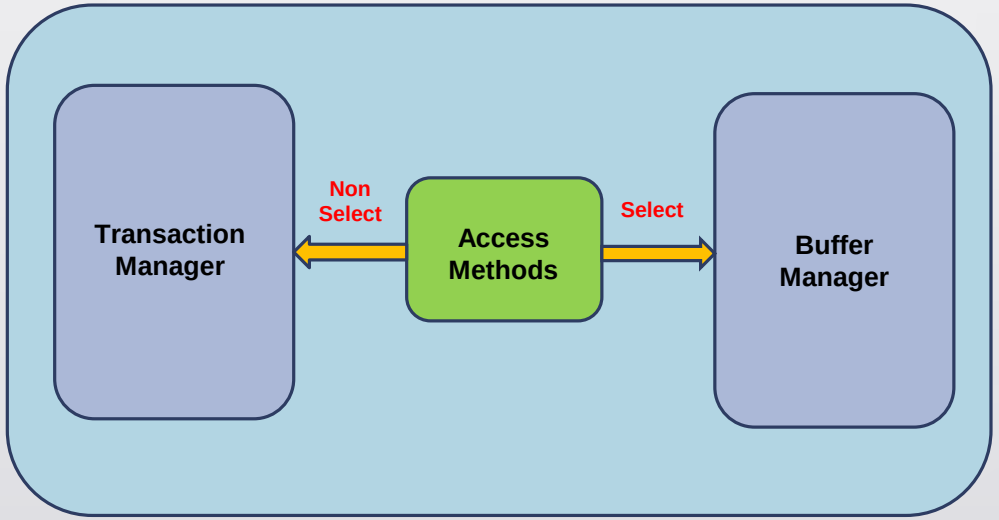


Storage Engine

- The work of the Storage Engine is to store data in a storage system like Disk or SAN and retrieve the data when needed.
- Three Major Components:
  - Access Method.
  - Buffer Manager.
  - Transaction Manager.

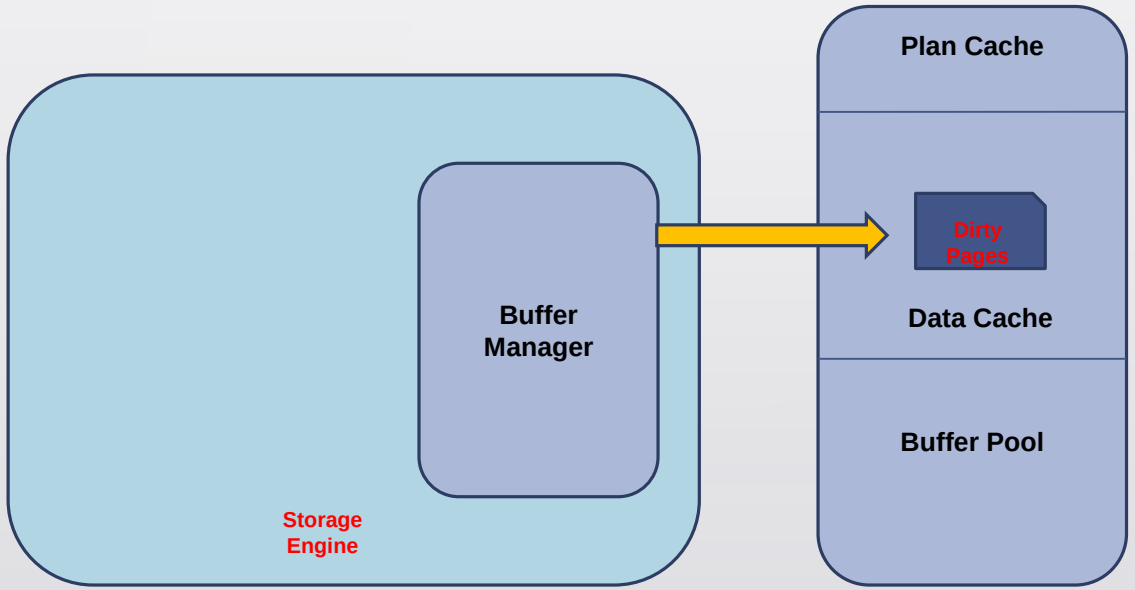
Access Method

- It acts as an interface between query executor and Buffer Manager/Transaction Logs and itself don’t do any Transaction.
- **SELECT statements** sent to Buffer Manager for further processing.
- **NON-SELECT statement**, the query is pass to Transaction Manager. This mostly includes the UPDATE statement.



Buffer Manager

- Buffer manager manages core functions for modules below:
  - Plan Cache.
  - Data Parsing : Buffer Cache & Data storage.
  - Dirty Page



Transaction Manager

- Transaction Manager is invoked when access method determines that Query is a Non-Select statement.
- Log Manager: Log Manager keeps a track of all updates done in the system via logs in Transaction Logs.
- Lock Manager : During Transaction, the associated data in Data Storage is in the Lock state. This process is handled by Lock Manager.