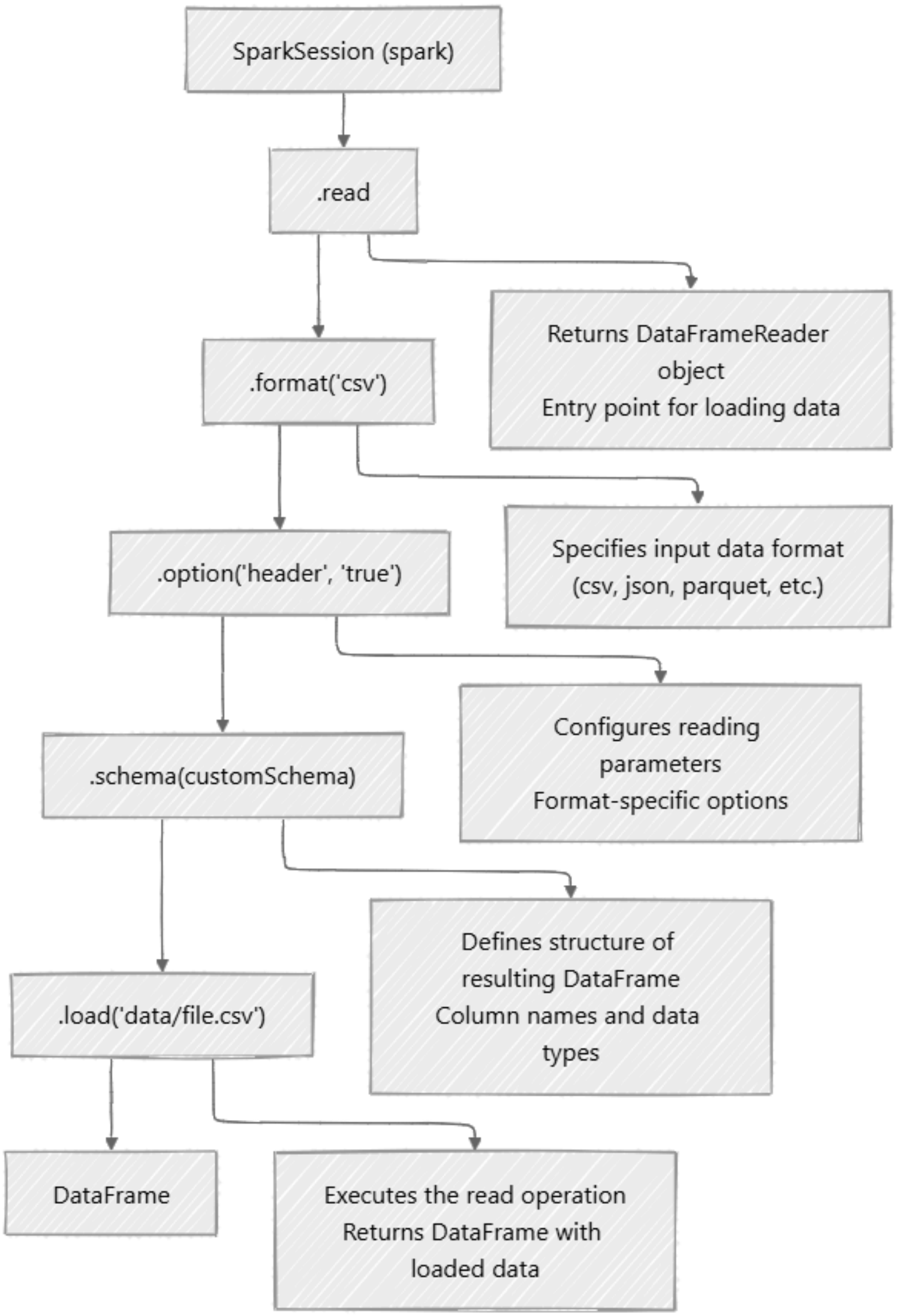
Reading Data in Pyspark

Reading Data

Explained Visually

https://manan.au

Reading Data in PySpark – Overview



.read: Returns a DataFrameReader object, which is the entry point for loading data into Spark.

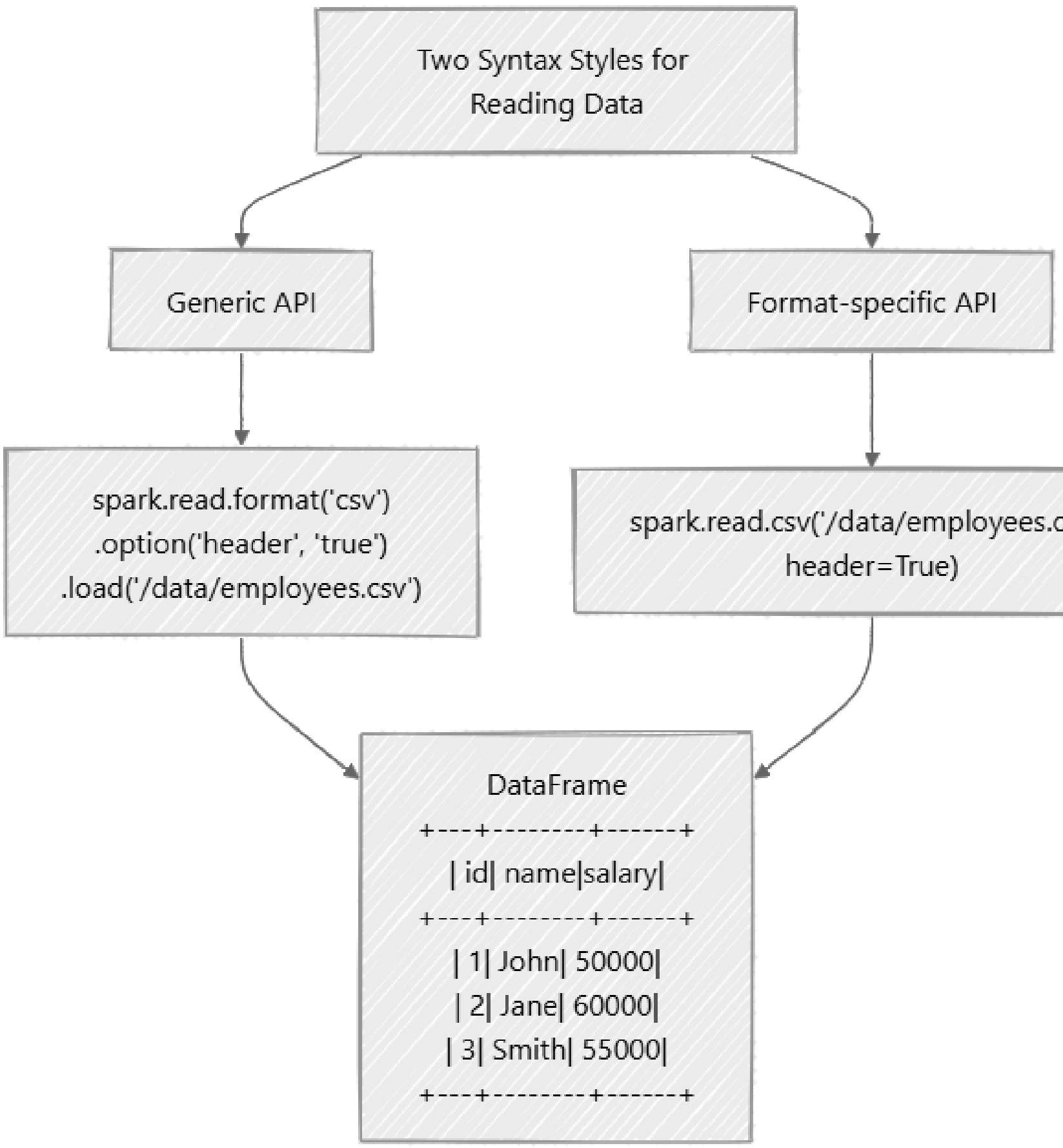
.format(): Specifies the format of the input data (CSV, JSON, Parquet, ORC, etc.)

.option(): Sets various configuration parameters for reading the data (e.g., header, delimiter, quote character)

.schema(): Defines the structure of the resulting DataFrame, including column names and data types

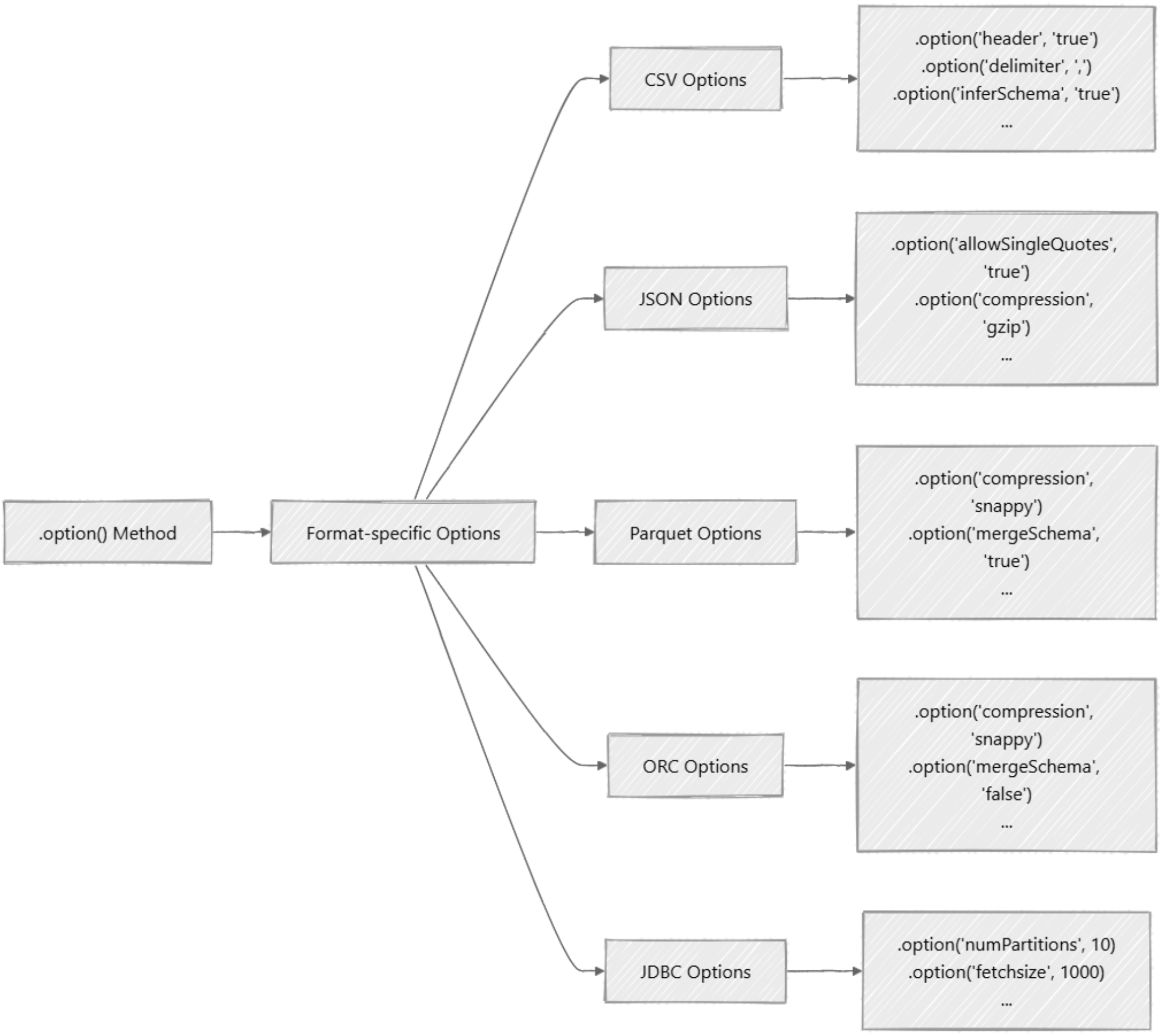
.load(): Executes the read operation with all the specified parameters and returns a DataFrame with the loaded data

Two Syntax Styles for Reading Data



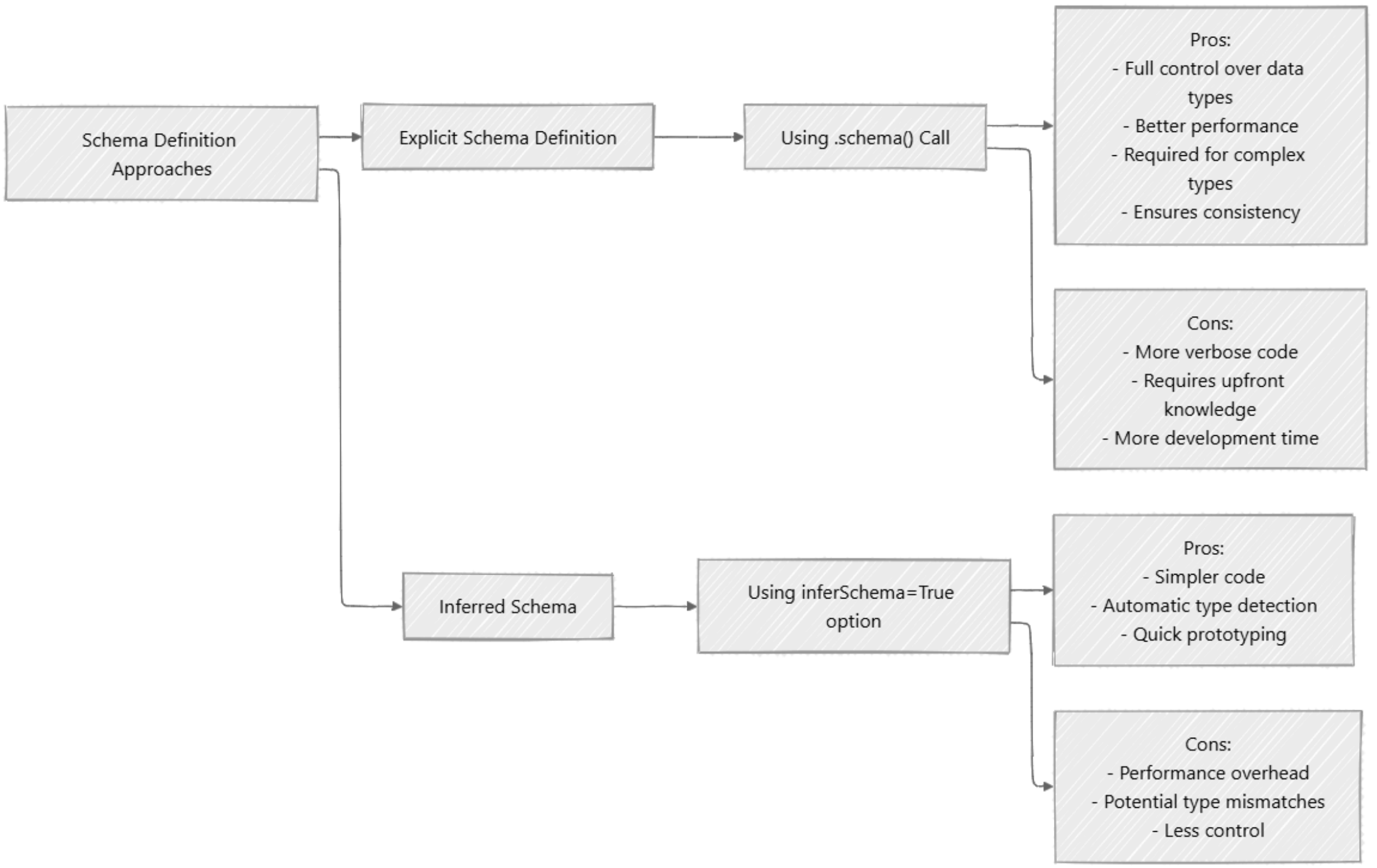
- Generic syntax uses format() to specify the file type explicitly
- Format-specific syntax provides shortcuts for common formats
- Format-specific methods improve code readability

The Option() Method for Data Reading



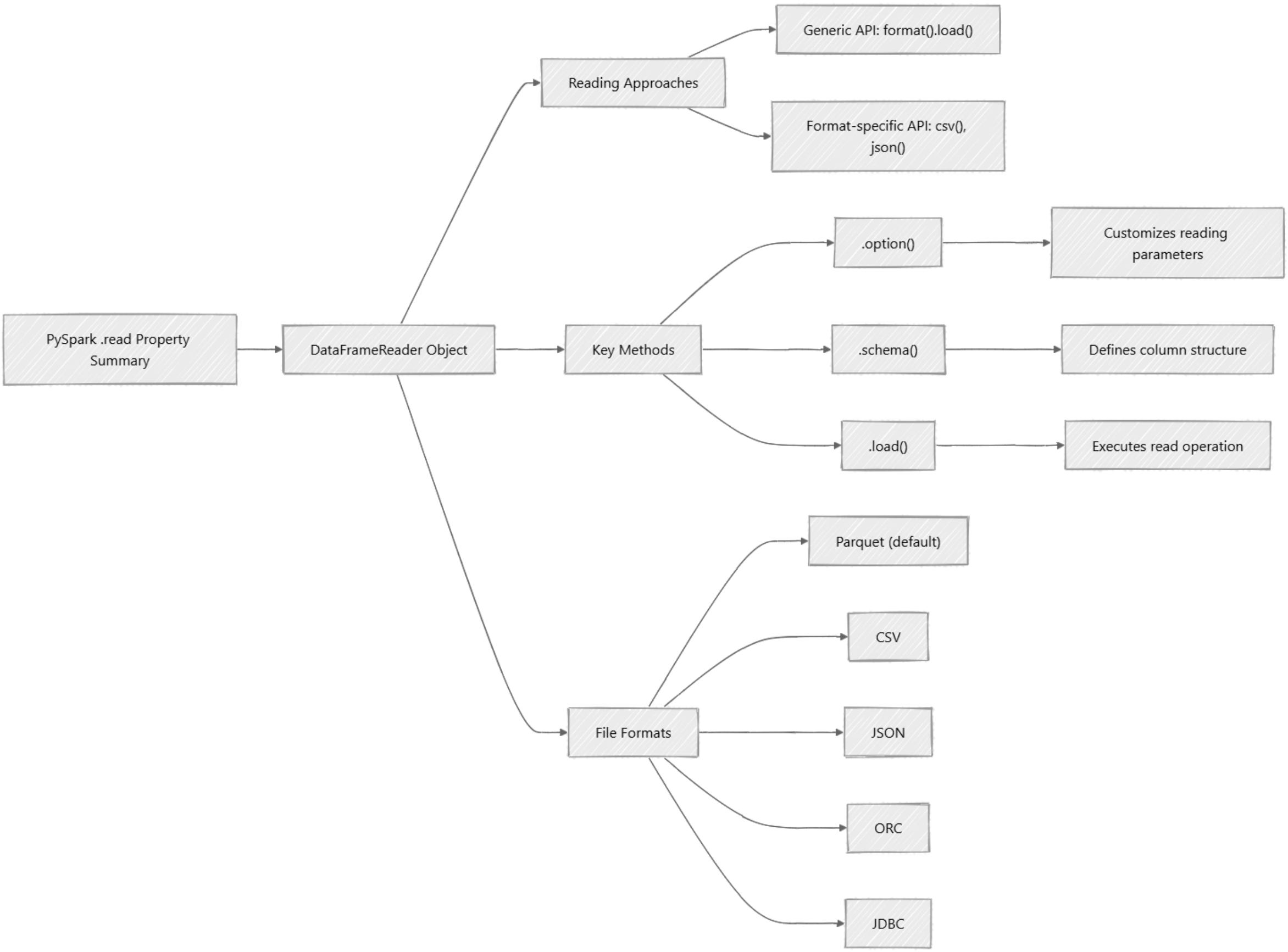
- option() customizes how data is read and interpreted
- Different file formats have their own specific options
- Multiple options can be specified by chaining method calls

The .schema() method for specifying Schema



- Two approaches to schema handling: Explicit definition with .schema() or automatic inference
- Explicit schemas provide better control: Ensures correct data types and prevents misinterpretations
- inferSchema option: Automatically detects column data types by sampling the data, which is convenient but can impact performance

Summary Reading Data



1. The .read property returns a DataFrameReader object, which is the entry point for loading data into Spark.
2. Two syntax styles are available: generic (format().load()) and format-specific shortcuts (csv(), json(), etc.).
3. The .option() method customizes how data is read with format-specific parameters.
4. Schema handling can be explicit (.schema()) or inferred (inferSchema=true).
5. Various file formats are supported with Parquet being the default.