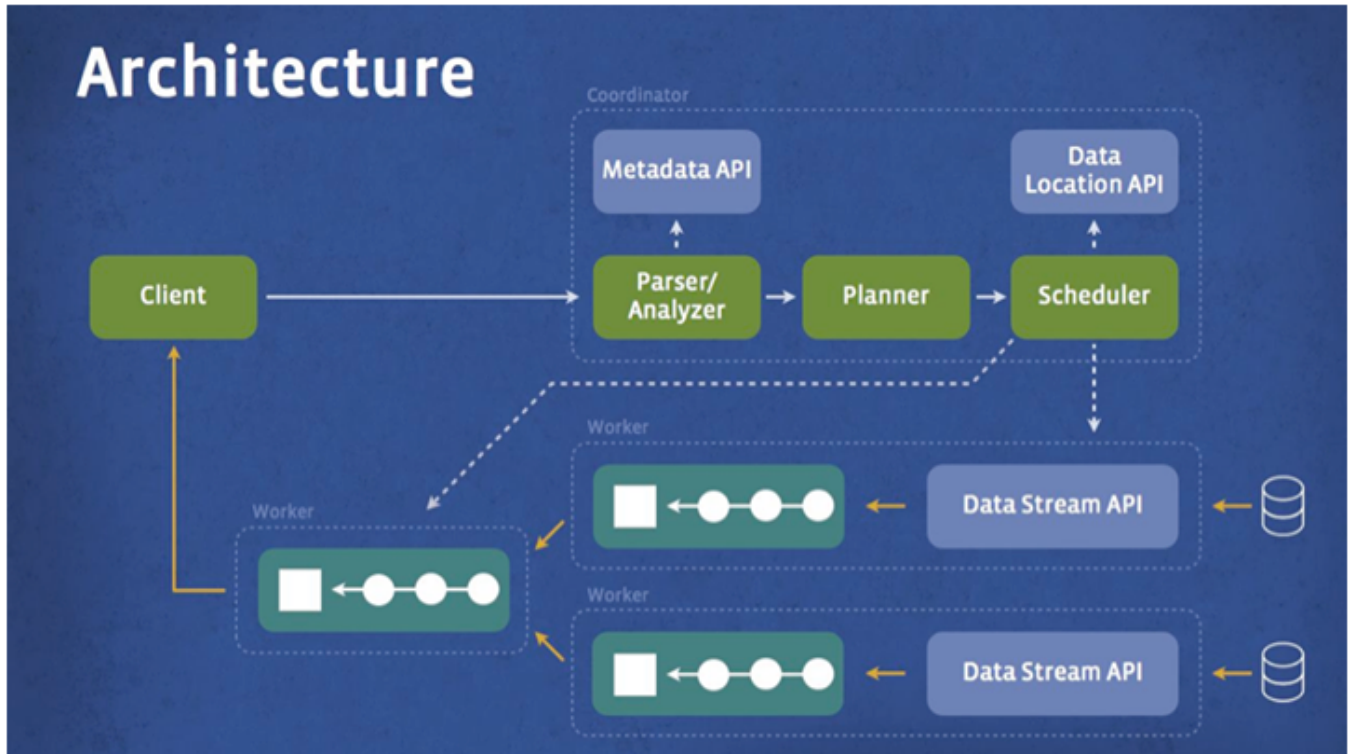


# Presto - a distributed SQL query engine based on pluggable backend at Facebook

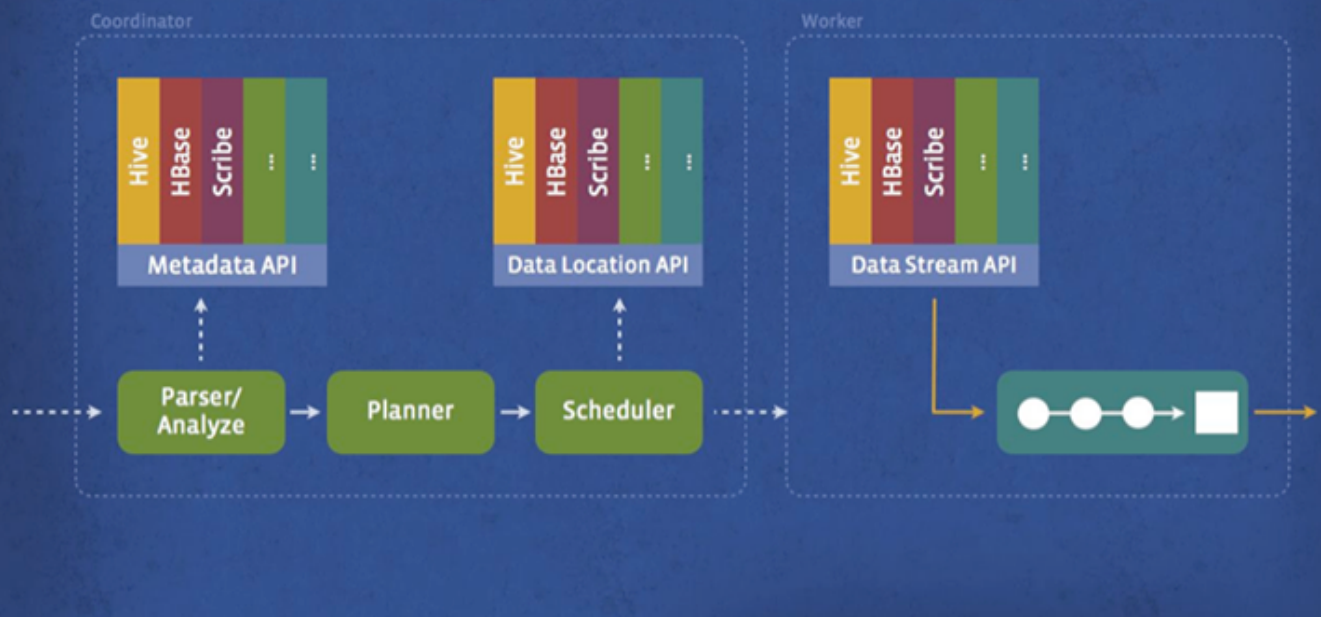
Before elaboration about Presto architecture, you should know Presto is based in Java even though Facebook is known as using PHP as their major server software language.

The diagram below shows the simplified system architecture of Presto. The client sends SQL to the Presto coordinator. The coordinator parses, analyzes, and plans the query execution. The scheduler wires together the execution pipeline, assigns work to nodes closest to the data, and monitors progress. The client pulls data from output stage, which in turn pulls data from underlying stages.



The execution model of Presto is fundamentally different from Hive/MapReduce. Hive translates queries into multiple stages of MapReduce tasks that execute one after another. Each task reads inputs from disk and writes intermediate output back to disk. In contrast, the Presto engine does not use MapReduce. It employs a custom query and execution engine with operators designed to support SQL semantics. In addition to improved scheduling, all processing is in memory and pipelined across the network between stages. This avoids unnecessary I/O and associated latency overhead. The pipelined execution model runs multiple stages at once, and streams data from one stage to the next as it becomes available. This significantly reduces end-to-end latency for many types of queries.

# Pluggable backends



Reference: <https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920/?s=keen-io>