The Quark Project is abstracted in 3 levels: -

- General Query Execution Model
- XQuery / XML Specific
- Quark Implementation Specific
General Query Execution Model
Components (Dataflow)

Legend
- component dataflow relation
- component dependency relation
- library dependency relation
- library inheritance relation
- component
- library
General Query Execution Model
Component Dependency

Legend
- component dataflow relation
- component dependency relation
- library dependency relation
- library inheritance relation
- component inheritance relation
- component
- library
Query Engine

$qe$ is the generic Query Engine which only executes a query string that is presented to it.
**YQGM**

**yqgm** defines the basic elements in Yet another Query Graph Model (YQGM), including Operator, Quantifier, ParseTree, PColumn, PFunction, OutputColumn, InputColumn and Expression

**Legend**

- component dataflow relation
- component dependency relation
- library dependency relation
- library inheritance relation
- component
- library
- more generic library

**yqgm_std** defines a number of concrete standard YQGM elements. These include standard operators, quantifiers, expressions and parse trees
yqgm_rules defines a number of rule templates for different types of basic YQGM elements including OperatorRule and QuantifierRule.

yqgm_std_rules defines a number of specific rewrite rules that take advantage of properties specific to certain kinds of elements. For example, SelectMergeRule merges two SPJ operators under certain conditions.
**squ** defines the common interface for SQU which encapsulates the functionality to store data and provide candidate execution plans.

**value_format** defines the interface for different schemes of assigning ValueFormats to stored information items.

**format_transformer** defines the interface for component which can transform one scheme of ValueFormat representation to another.

**Legend**
- component dataflow relation
- component dependency relation
- library dependency relation
- library inheritance relation
- component
- library
- more generic library
- library from another component
Optimizer

- **optimizer** defines the interface for the optimizer component
- **value_format**
- **formatter**
- **yqgm**
- **squ**

Legend:
- Component dataflow relation
- Component dependency relation
- Library dependency relation
- Library inheritance relation
- Component
- Library
- More generic library
- Library from another component

**optimizer_squ** is the implementation of optimizer which works together with Storage/Query Unit (SQU) component
Evaluator

evaluator provides a general evaluator interface

evaluator_std passes an YQGM graph to physical_converter and evaluates the physical graph

iterator_evaluator retrieves tuples iteratively

materialization_strategy

yqgm

yqgm_physical_converter converts YQGM objects to their physical equivalents

physical_model

physical_model_yqgm is a general physical model for evaluation

physical_model_yqgm_yqgm is a general physical model for YQGM

physical_model_yqgm_std

physical_model_yqgm_std is a physical model for yqgm_std

yqgm_std
XML Specific
Components (Dataflow)

Component names in red are XML specific while those in black correspond to general components.
Component names in red are XML specific while those in black correspond to general components.
XQuery Datamodel

Legend

- Component dataflow relation
- Component dependency relation
- Library dependency relation
- Library inheritance relation
- Component
- Library
- More generic library
- Library from another component
XQuery Implementation

\texttt{xquery\_model\_extended} implements Quark specific functions which extend standard XQuery functions

\texttt{xquery\_model\_extended\_mm} is the main memory implementation of \texttt{xquery\_model\_extended}
XQuery YQGM

xquery_model_extended

yqgm

yqgm_std

xquery_model

xquery_yqgm

Legend

(component dataflow relation)
(component dependency relation)
(library dependency relation)
(library inheritance relation)
(component)
(library)
(more generic library)
(library from another component)

xquery_yqgm implements conversion of XQuery functions to YQGM
**XQuery Parser**

xquery_parser implements a parser for XQuery

Legend

- component dataflow relation
- component dependency relation
- library dependency relation
- library inheritance relation
- component
- library
- more generic library
- library from another component
XML Storage

value_format

value_format_xml_std

Legend

component dataflow relation
component dependency relation
library dependency relation
library inheritance relation
component
library
more generic library
library from another component

value_format_xml_std includes a definition of start end tags. It represents the start-end ID scheme used for xml data.
XQuery Rewriter

yqgm_std_rules_xml implements specific rewrite rules for YQGM elements such as View Composition rules.
XQuery Evaluator

yqgm
xquery_model
xquery_yqgm

materialization_strategy
materialization_strategy_mm

physical_model

physical_model_yqgm

physical_model_yqgm_xml

physical_model_yqgm_std

yqgm_std
Component names in black correspond to general components, those in red to XML specific while those in blue to Quark implementation specific.
Component names in black correspond to general components, those in red to XML specific while those in blue to Quark implementation specific.
Quark Query Engine

`qe_quark` is Quark's implementation of a query engine which calls the parser, rewriter and optimizer.
Legend

- component dataflow relation
- component dependency relation
- library dependency relation
- library inheritance relation
- component
- library
- more generic library
- library from another component

Storage

squa

squa_default_xml

squa_rootpath_xml

squa_rootpath_xml includes a SQU which can provide execution plans using RootPath index lookup operation