The Quark database system can be divided into 4 levels of abstraction:

**Quark XML DB System**
- main() function
- Quark Parser

**Relational DB System**
- main() function
- Command Parser

**Quark XML SQUs**
- File System SQU
- Shore XML SQU
- Default SQU

**Relational SQUs**
- Shore RDB SQU
- ODBC SQU

**Quark XML DB Infrastructure**
- XML Data Model
- XQuery Parser
- XML Rewriter

**Relational DB Infrastructure**
- Relational Data Model
- SQL Parser
- Relational Rewriter

**Quark DB Infrastructure**
- Server
- Client
- Command Line Processor
- Command Processor
- YQGM
- Rewriter
- Storage
- Optimizer
- Evaluator
The 3 levels of abstraction next to the XML data model specific portions of Quark depict how Quark can be extended to work with a relational data.

**Quark DB Infrastructure**
The Generic query engine of Quark, also called the Quark DB Infrastructure, forms the core data model-independent portion of Quark. It contains a set of components which take the query in the form of a graph called a YQGM (Yet another Query Graph Model) graph and see it through to evaluation. The main components are the Rewriter (which rewrites the YQGM graph into a simpler form), Storage (which handles the storage and retrieval of data), Optimizer (which provides evaluation plans for the evaluator) and Evaluator (which converts the YQGM graph to a physical graph and evaluates it). YQGM is independent of the underlying data model and is hence used to represent the queries in Quark.

**Quark XML DB Infrastructure**
The Quark XML DB Infrastructure is the XML specific component of Quark. It implements the XML data model that is used by the XML specific versions of the various components in the Quark DB Infrastructure. It has XML specific extensions to YQGM, a parser to parse XQuery and TeXQuery queries and a Rewriter implementation for rewriting YQGM graphs containing special configurations of XML specific extensions to YQGM.

**Quark XML SQUs**
The Quark XML SQUs (Storage Query Units) form the storage component of Quark, and contain the implementations for the SQUs that can be used in Quark. SQUs have the function of storing a piece of data and returning an ID corresponding to it, retrieving the piece of data stored when given its ID, and returning optimized evaluation plans for a part of a YQGM graph given the root of some sub tree of the graph. The libraries in this layer of Quark are squ (which is an interface for SQU implementations to follow), and squ_default (which is the implementation of a default SQU that guarantees an evaluation plan given any part of the YQGM graph corresponding to any valid query).

**Quark XML DB System**
The Quark XML DB system links the generic DB infrastructure with the XML SQUs in Quark. All SQUs are registered in this part of Quark. New SQUs that users wish to register and the SQUs provided with Quark are all registered here. It also contains the main() function which is the entry point for the execution of Quark, and decides whether to start Quark’s Server or Command Line Processor.