

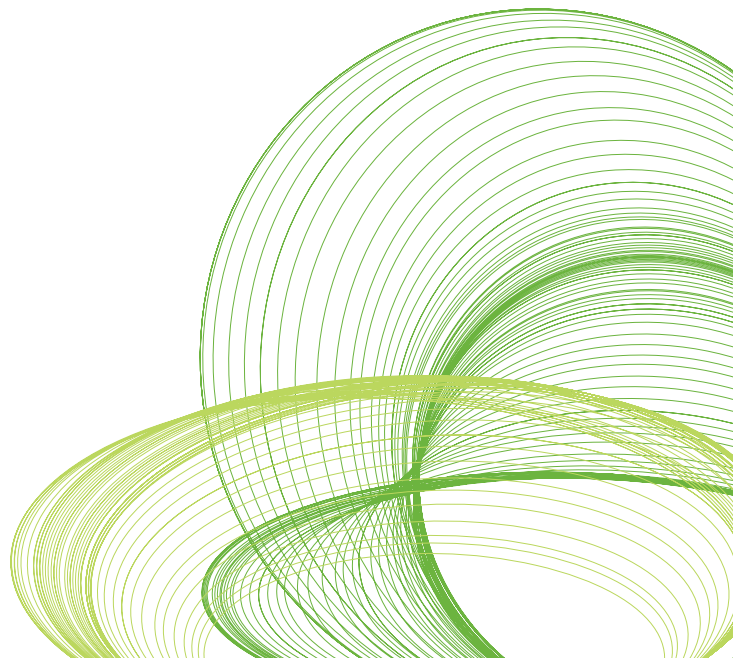


QLIKVIEW DEVELOPMENT AND DEPLOYMENT ARCHITECTURE

A QlikView Technical Brief

Published: February, 2011

qlikview.com



Products Overview

The QlikView Business Discovery platform consists of 3 major components, each playing an important part in designing, developing and implementing almost every QlikView deployment.



QlikView Developer

The QlikView Developer is a Windows-based desktop tool that is used by designers and developers to create 1) a data extract and transformation model and 2) to create the graphical user interface (or presentation layer). It is within this environment where a developer will use a SQL-like scripting environment (augmented by 'wizards') to create the linkages (connection strings) to the source data and to transform the data (e.g. rename fields, apply expressions) so that it can be analyzed within the User Interface, as well as re-used by other QlikView files. The QlikView Developer is also the environment where all user interface and user experience design is developed in a drag-and-drop paradigm: everything from graphs and tables containing slices of data to multi-tab architectures to application of color scheme templates and company logos and so forth is done here.

Previously created SQL code can be ported into the QlikView Developer environment very easily, negating the need for large amounts of re-work or re-coding when beginning to use QlikView within an existing BI framework.

The file type that is created using the QlikView Developer is known as a 'QVW' (.qvw). Upon reload, a QVW can be used to create a data-only 'QVD' file, which is binary and contains no UI.



QlikView Server (QVS)

The QVS is a server-side product that contains the in-memory analytics engine and which handles all client/server communication between a QlikView client (i.e. desktop, IE plugin, AJAX or Mobile) and the server. It includes a Management Console for providing Administrator access to control all aspects of the server deployments (including security, clustering, distribution etc.) and also includes a web server to provide front-end access to the documents within. The web server's user portal is known as "Access Point". (It's important to note that while the QVS contains its own web server, one can also utilize Microsoft's IIS for this purpose too). The QVS handles client authorization against existing directory providers (e.g. AD, eDirectory) and also performs read and write to ACLs for QVW documents.



QlikView Publisher

The QlikView Publisher is a server-side product that performs 2 main functions: 1) It is used to load data directly from data sources defined via connection strings in the source QVW files. 2) It is used as a distribution service to 'reduce' data and applications from source QVW files based on various rules (such as user authorization or data access) and to distribute these newly-created documents to the appropriate QlikView servers or as static reports via email.

Figure 1: Architecture Overview

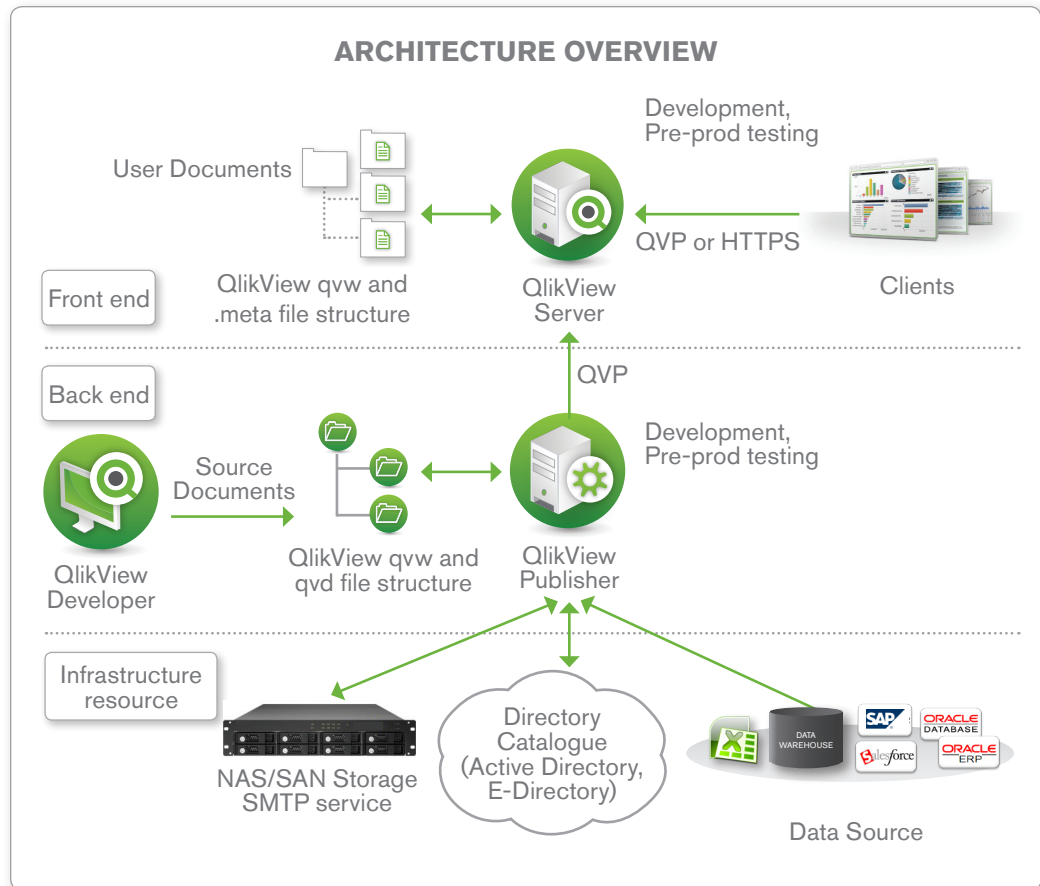


Figure 1 depicts a simplified view of a standard QlikView deployment containing the location of the various QlikView products as well as both data and application locations.

BACK END (INCLUDING INFRASTRUCTURE RESOURCES):

This is where QlikView source documents, created using the QlikView Developer, reside. These source files contain either a) scripts within QVW files to extract data from the various data sources (e.g. data warehouses, Excel files, SAP, Salesforce.com) or b) the actual binary data extracts themselves within QVD files. In addition, .log files are contained here, if active. The main QlikView product component that resides on the Back End is the QlikView Publisher and, as stated above, the Publisher is responsible for data loads and distribution. Within the Back End, the Windows file system is always in charge of authorization (i.e. QlikView is not responsible for access privileges).

The Back End depicted here is suitable for both development, testing and deployment environments.

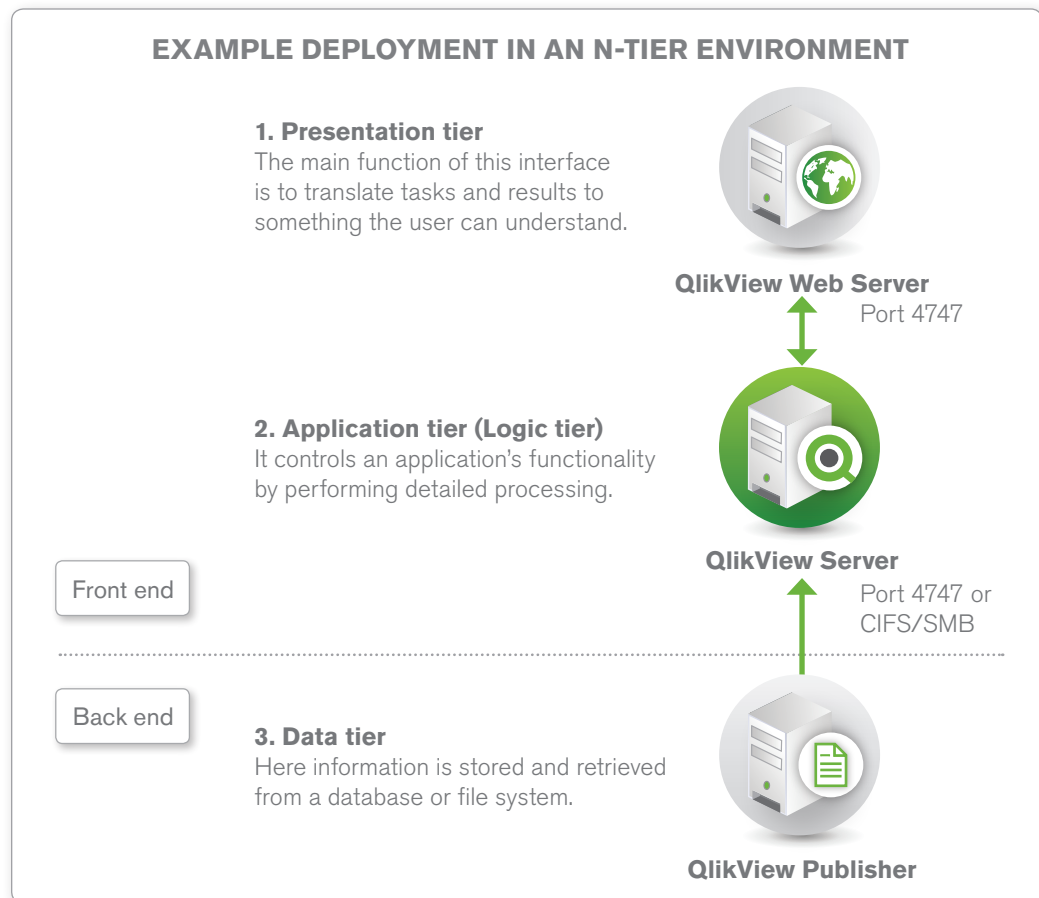
FRONT END

The Front End is where end users interact with the documents and data that they are authorized to see via the QlikView Server. It contains the QlikView user documents that have been created via the QlikView Publisher on the back end. The file types seen on the Front End are QVW, .meta and .shared documents. All communication between the client and server occurs here and is handled either via HTTPS (in the case of the AJAX client) or via the QlikView proprietary QVP protocol (in the case of the plugin or Windows client). Within the Front End, the QVS is responsible for client security.

From a security standpoint, it's important to understand that the Front End does not have any open ports to the Back End. It does not send any queries to data sources on the back end, nor do any of the user documents (QVW's) contain any connection strings to data sources located on the back end. End users can only access QlikView documents that exist on the Front End, and never in the Back End.

Since all documents in the Front End are created via Publisher tasks, it is very straightforward to recreate all user documents by simply running these tasks again.

Figure 2: Three Tier Architecture



A standard QlikView deployment using AJAX employs a Three-Tier architecture where data, application and presentation layers are maintained in distinct, separate environments. In Figure 2, the highly important Data tier contains only the QlikView Publisher, the product that provides for data loading and application and data distribution. As mentioned above, access to this tier is very limited and no end users or applications within the other 2 tiers have access to this layer and the data or files within. The Application (or Logic) tier contains the QlikView Server(s) and the user documents that have been distributed via the Publisher in the Data tier. Communication from the Publisher to the Server is done securely via port 4747. Finally, the Presentation tier contains the Web Server(s) (either the QlikView Web Server or IIS). This is where user requests for application access are handled via the 'Access Point' portal.

It's important to note that within the Data tier, there are many different approaches to handling data access and modeling and one must carefully consider the various Data Architecture approaches covered in documents such as QlikView Data Architectures.pdf to fully understand the optimal approach for an individual deployment scenario.

© 2010 QlikTech International AB. All rights reserved. QlikTech, QlikView, Qlik, Q, Simplifying Analysis for Everyone, Power of Simplicity, New Rules, The Uncontrollable Smile and other QlikTech products and services as well as their respective logos are trademarks or registered trademarks of QlikTech International AB. All other company names, products and services used herein are trademarks or registered trademarks of their respective owners. The information published herein is subject to change without notice. This publication is for informational purposes only, without representation or warranty of any kind, and QlikTech shall not be liable for errors or omissions with respect to this publication. The only warranties for QlikTech products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting any additional warranty.