

### **Pages**

Introduction

Introduction

Stage 1

Stage 2

Stage 3

Stage 4

The following pages outline several data architectures that can be deployed as part of QlikView solutions. These pages do not depict every possible architecture. In fact, they may only represent a small portion of the possible architectures. They do, however, represent the concepts that are most often employed (with the exception of the Stage 1 Architecture) to promote scalability, re-use and consistency.

These pages are meant to help provide a backdrop for design and architecture discussions related to enterprise deployments with QlikView.



## **Stage 1 Architecture**

#### **Description:**

This architecture involves only direct queries against source databases. All presentation layer applications make database connections and contain their own SQL queries and QlikView scripts to load, transform and aggregate any data needed.

#### When to use:

This is the simplest QlikView architecture, but also the most costly to maintain. Queries are often repeated across many applications. Reloads of applications may be competing for the same database resources. Intraday reloads will need access to source databases to load from.

Source Layer

Databases and other data sources

Presentation Layer

Production applications. Each one is scripted with source database queries and any transformational scripting needed for the final interface. Many elements and metrics overlap across applications, requiring redundant code and processing.

# DATABASE **VAREHOUSE** ORACL **C**alesforce ERP Dashboard Dashboard Dashboard Analysis App

#### **PLEASE NOTE:**

This architecture is not a recommended best practice for enterprise QlikView deployments.

© 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 or Rules, The Uncontrollable Smile and other 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.

Author: BPN. QlikTech NA



DATABASE

ORACL

ERP

## **Stage 2 Architecture**

#### **Description:**

This architecture includes the use of QVD files for a 2<sup>nd</sup> data tier. The presentation layer applications will still need to generate data models from several QVD files, but they will not need to extract directly from source databases. This helps promote re-use and consistency across presentation layer applications.

#### When to use:

Use this architecture when you want to shield presentation layer developers from direct database retrievals and promote re-use. Some data modeling and scripting expertise will be needed at the presentation layer, but not necessarily any source database query knowledge or skills.

Databases and other data sources

QlikView applications that extract and (optionally) denormalize source tables into QlikView QVD files.

t rce

WAREHOUSE



( ales force

QVD Generator QVD Generator QVD Generator QVD Generator

QVD Lay

**Presentation Layer** 

QVD Files – QlikView data file layer. QVDs can be one-to-one match with source tables or denormalized "views" of several source tables

Production applications built from QVDs in the layer above. No direct database queries are needed in these applications and re-use of common QVDs is promoted. Some understanding of data model best practices is still required to optimize application performance.

QVD Files QVD Files QVD Files







Analysis App

Dashboard

© 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 PlikTech 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.

Author: BPN. QlikTech NA



## **Stage 3 Architecture**

#### **Description:**

This architecture includes the use of QVD files for the 2<sup>nd</sup> and 3<sup>rd</sup> tiers. The first QVD layer can be very normalized, in some cases it can be a one-to-one match of source DB tables. This layer accommodates very fast extractions with minimal transformations and aggregations. The second QVD layer is where aggregated and denormalized QVDs are exposed to presentation layer developers. This helps support codevelopment of QlikView between IT and business teams.

Use this architecture when a short batch window is desired (QVD layer is fast) and you want to create the presentation layer data models from a QVD layer, but you want this layer to be optimized for re-use and performance.

Source Databases and other data sources

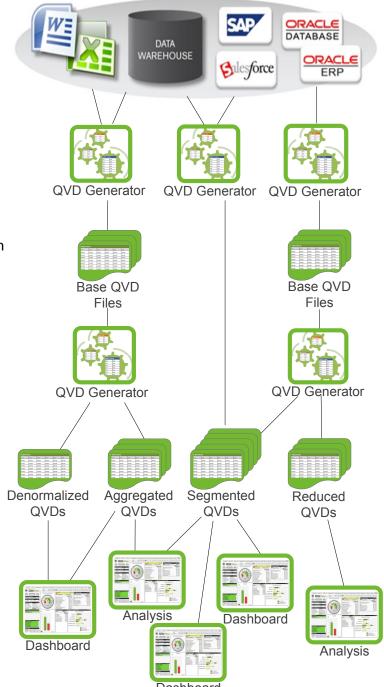
QlikView applications that extract and (optionally) denormalize source tables into QlikView QVD files.

QVD Files - QlikView data file layer. QVDs can be one-to-one match with source tables or denormalized "views" of several source tables

**Transformation** These QlikView apps read the base QVDs and perform aggregation and/or denormalization, writing out the final QVDs to be used by presentation layer applications.

Transformed These QVDs are aggregated, regionalized, reduced or denormalized to optimize re-use, ease of deployment and performance. These are the QVDs that are exposed to presentation layer developers.

Production applications built from QVDs in the layer above. Some understanding of data model best practices is still required to optimize application performance.



#### When to use:

# **QlikView**

#### **Description:**

This architecture includes the use of "QlikMart" applications. These QlikMarts are just QlikView applications without a completed user interface. They act as data models that can be loaded into a presentation layer application with a single line of code (called a binary load). This helps support co-development of QlikView between IT and business teams.

#### When to use:

This architecture should be used when "self-service BI" is a goal, or when many dashboards will overlap in data usage and IT wants to control the consistency and predictability of dashboard performance. This architecture shields the presentation layer designers from data model preparation, except in cases where a QlikMart has not yet been built

Stage 4 Architecture

Databases and other data sources

QlikView applications that extract and (optionally) denormalize source tables into QlikView QVD files.

Rich QVD Layer

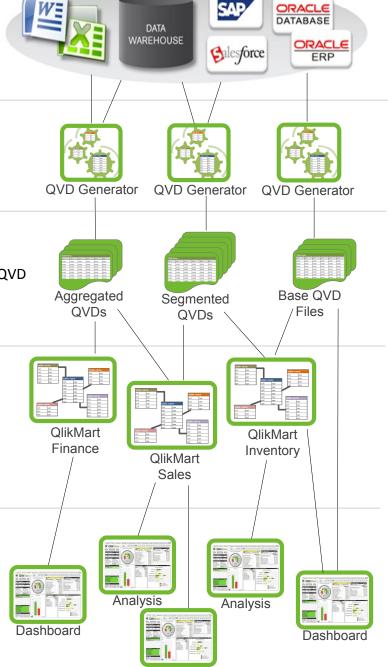
QVD Files – combination of base QVD files, aggregated, segmented and denormalized QVD files.

A "QlikMart" is a QlikView application that simply holds a data model that can be binary loaded as the base data model for end-user applications. Examples might be a Finance QlikMart, a Sales QlikMart and

an Inventory QlikMart.

Presentation Layer

Production applications. Built from QlikMart applications where possible. Minimal (if any) scripting. Can optionally retrieve from QVD layer as well as QlikMart layer.



Dashboard