



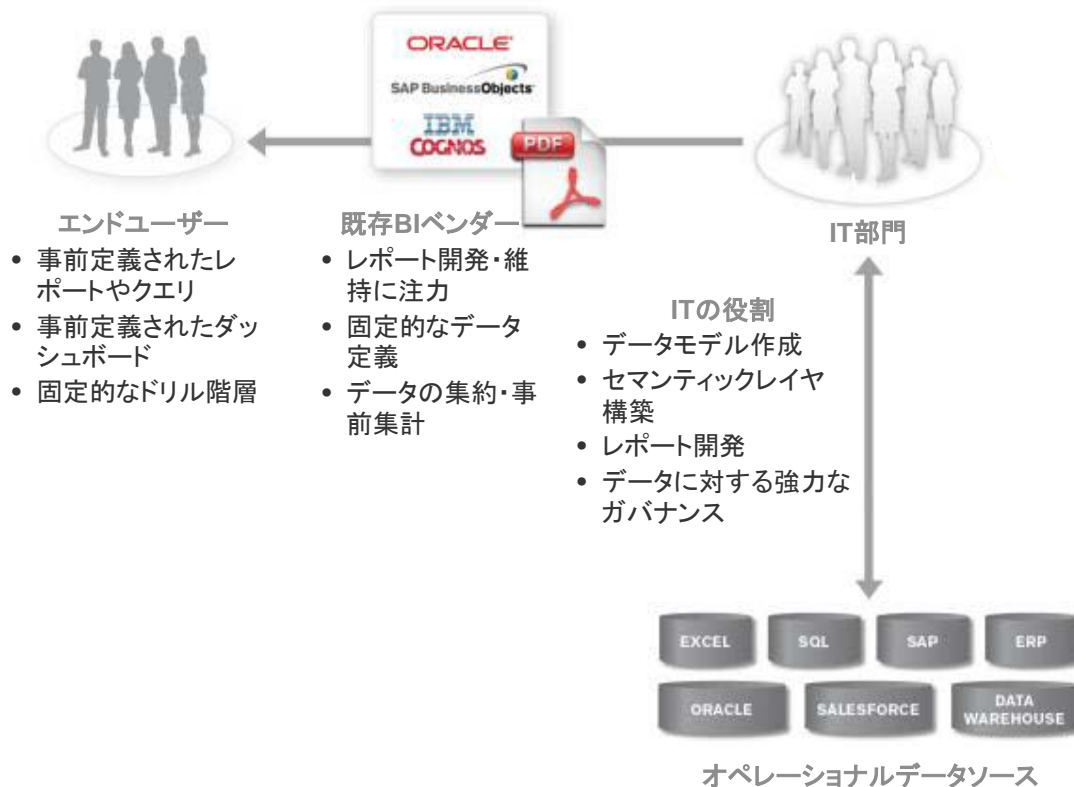
QlikView におけるデータ構成と QVDの利用

クリックテック・ジャパン株式会社

QlikView

そもそもBIはだれの為のものなのか？ ～ BIに革新をもたらすQlikView ～

レポート開発中心のアーキテクチャ (IT主導で管理負担も大きい)

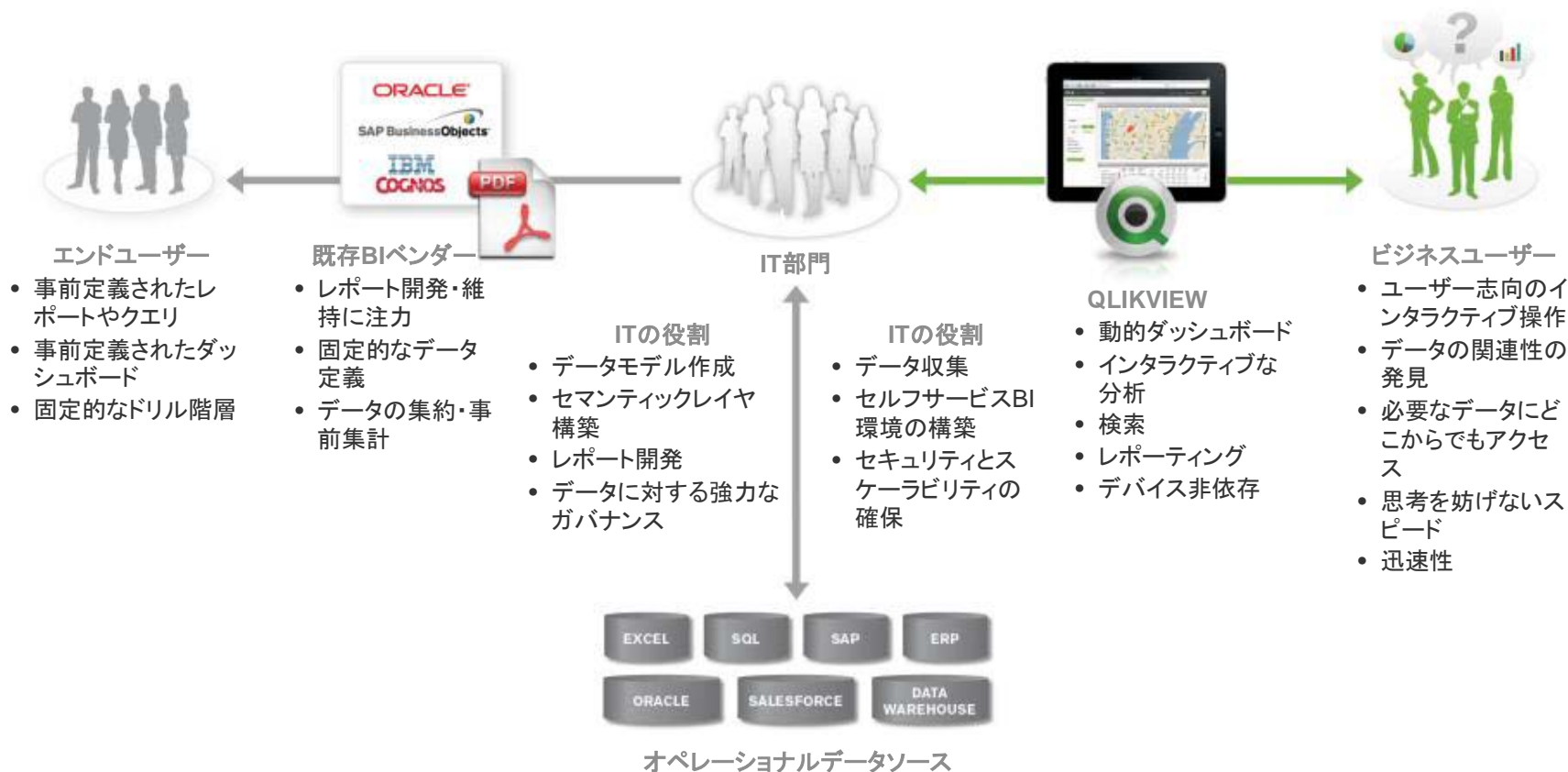


そもそもBIはだれの為のものなのか？

～ BIに革新をもたらすQlikView ～

レポート開発中心のアーキテクチャ
(IT主導で管理負担も大きい)

BUSINESS DISCOVERY アーキテクチャ
(ビジネスユーザー主導のセルフサービス型)



BI運用の主体:ユーザー部門かIT部門か?

～ 両部門のバランスが大切 ～

ユーザ主体運用

システム部主体運用

ユーザー自由度

高い

低い

IT作業負担

少ない

多い

ITガバナンス

低い

高い

情報活用推進

- レポート作成
- テンプレート提供
- 教育・ノウハウ共有
- 個別ニーズの対応



データ管理/運用

- データモデル作成
- 運用ルール決定
- データ品質管理
- セキュリティ管理



QlikViewの
目指す形

これまで

ユーザー部門とIT部門の役割を明確にし、
ユーザー自由度とITガバナンスを両立するBIを目指します



QlikViewにおけるデータ構成で役割を明確化



IT部門

運用・保守の工数削減・効率化

- データ抽出、加工およびユーザー開放により、レポート作成負荷を大幅削減
- セキュリティ・コンテンツ・データ管理



ユーザー部門

自由度・柔軟性の高い情報活用

- 業務ユーザーでも自らレポートを作成・変更することが可能
- 使いやすい柔軟な操作で、あらゆる情報活用シーンで使える

ソースレイヤ



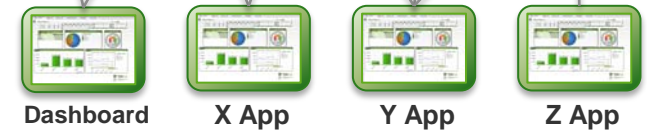
ETLレイヤ



QVDレイヤ



プレゼンテーションレイヤ



IT部門

ユーザー部門

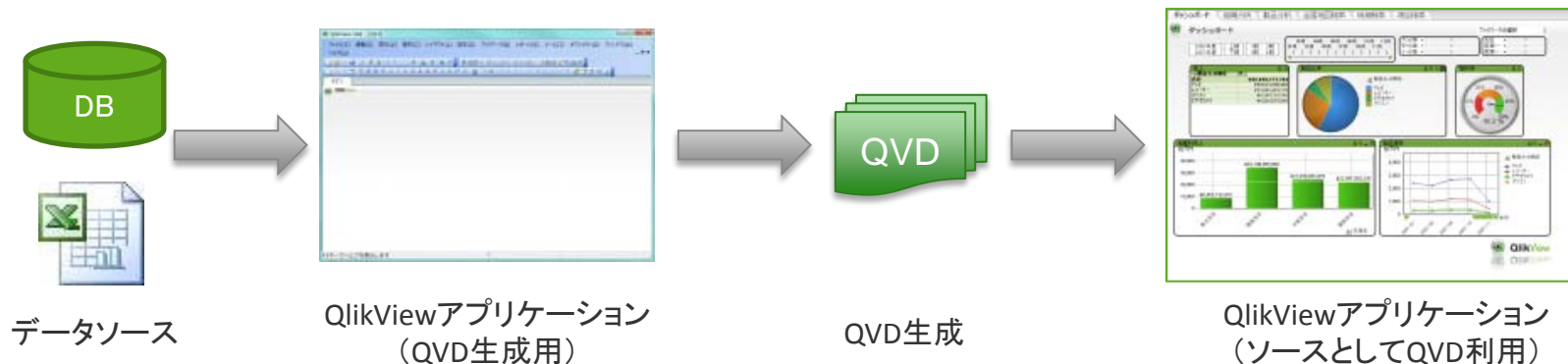


補足：
QVDによるデータ管理と階層分化



QVDの概要

- QVDファイル = QlikView Data ファイル
 - QlikViewから抽出されたテーブルデータを含むファイル
 - 1 QVDファイルにつき1テーブルデータを格納
 - QlikView独自フォーマットであり、QlikViewからのみ読み書き可能
- QVDファイルの作成およびロード
 - QVDファイルは、QlikViewアプリケーション経由で作成され、最終的に別のQlikViewアプリケーションのデータソース(の一部)として利用されます



QVDの概要

- QVDファイル生成スクリプトの例

Data:

Load ID,

Description

FROM C:¥Data.xls (**biff, embedded labels, table is**
[Core Data\$]);

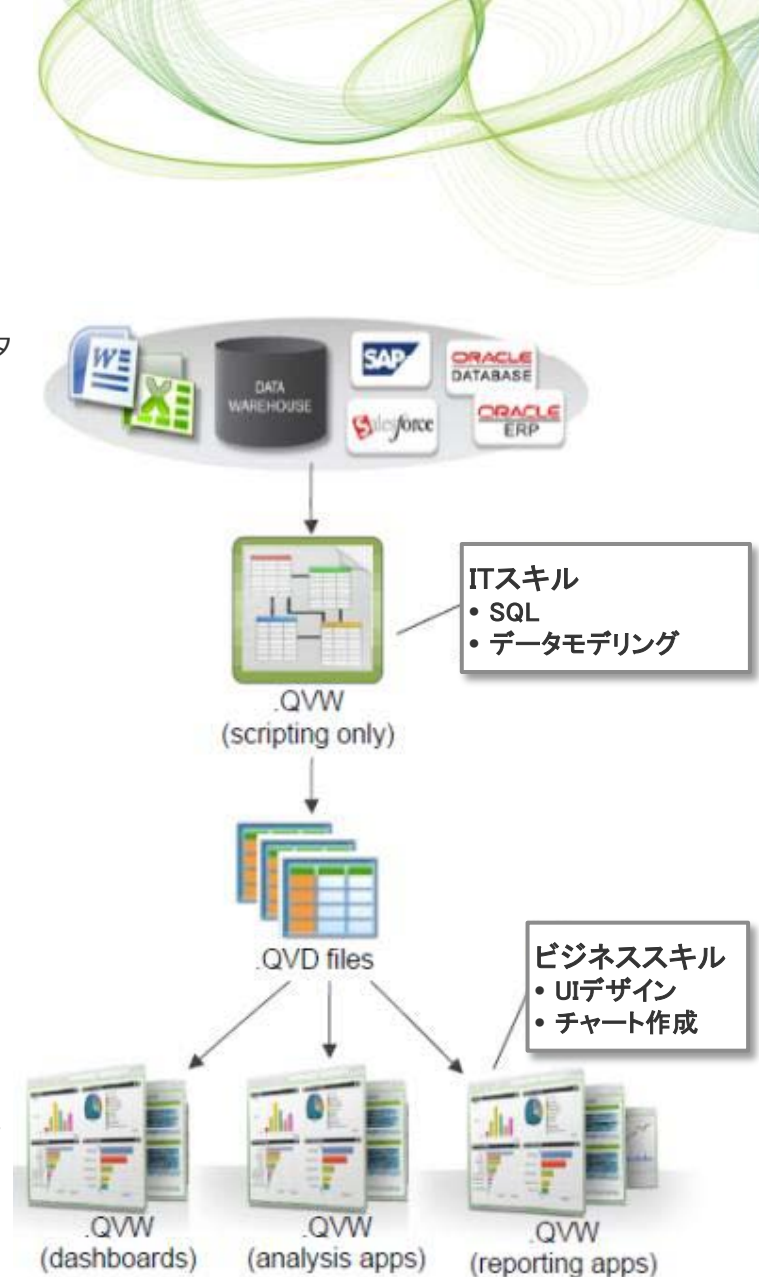
Store Data into C:¥Data.qvd;

Drop Table Data;



QVDの利点

- ソースデータの一元化
 - データセットを再利用することで、QlikView環境におけるソースデータの一元化が可能
- 柔軟なデータ管理
 - QVDレイヤーを構築することで、IT部門の管理範囲と、ビジネスユーザー部門の管理範囲を切り分け、双方の責任範囲を明確化することが可能。
- 柔軟な開発
 - QVDをデータベースから切り離されたファイルとして持つことで、開発/検証が容易
- 開発スピードの向上
 - 開発者は毎回データベースにSQLクエリー発行する代わりに、QVDの一覧をライブラリとして必要なデータを容易にかつ高速に取り込み、開発・テストが可能。
- 柔軟なデータ加工
 - QVDは、集約や分割、特定のニーズに応じた加工が容易で、クエリーを再作成するより簡単。
- データロードの高速化
 - 一度QVDを作成しておくで、そのままDBからロードする場合に比べてQVWへのデータ取り込みが10~100倍程度高速化。



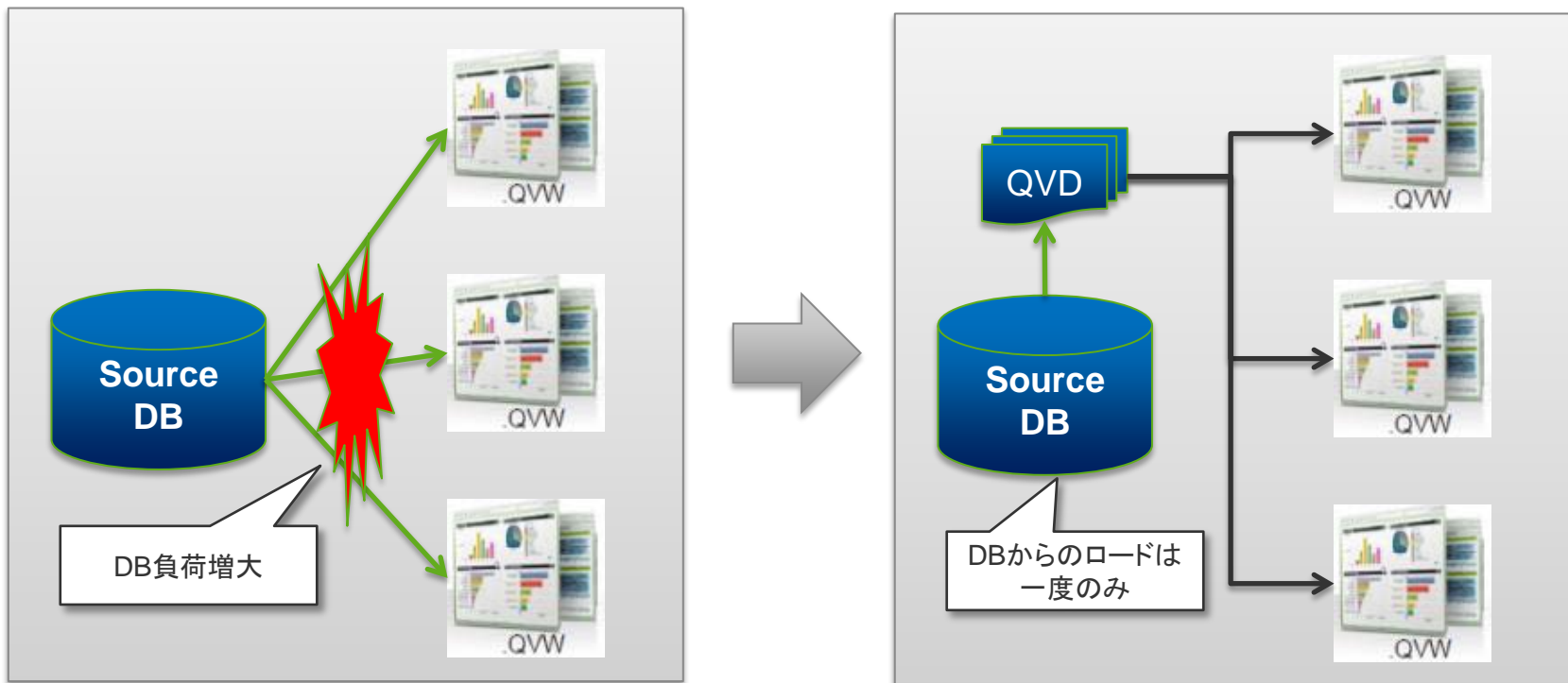
QVDの主な用途：データロードの高速化

- QVDファイルの読み込みは、データベースからの読み込みに比べて高速となります。（10～100倍）
- 多くの場合、アプリケーションの開発時には、スクリプトを繰り返し実行する必要があります。このような状況で QVD ファイルを使用すると、データセットがあまり大きくない場合でも、繰り返される待機時間を大幅に短縮できます。



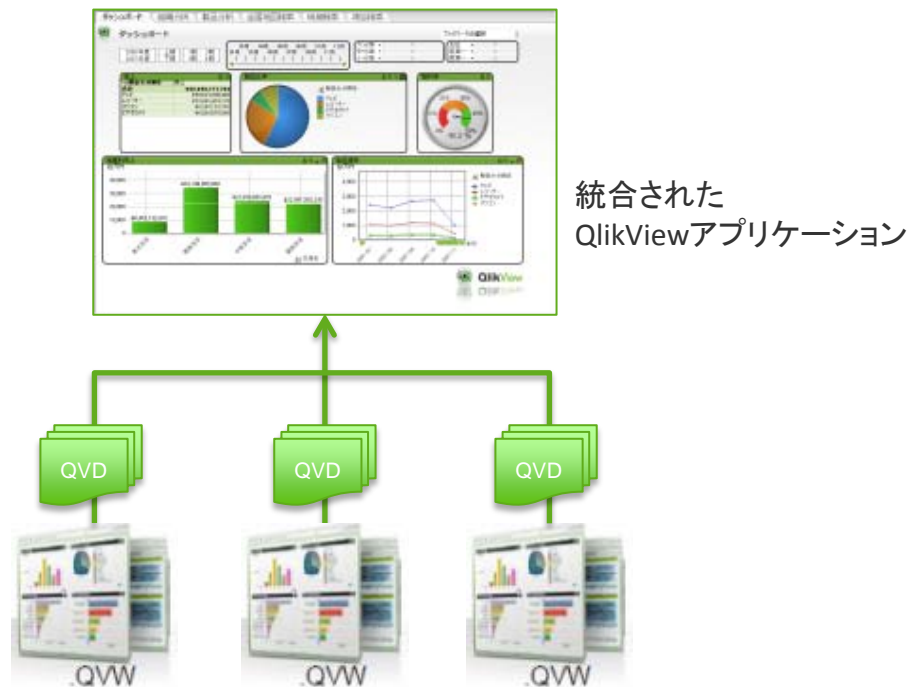
QVDの主な用途：データベースサーバーの負荷軽減

- 複数の QlikView スクリプトが同じデータを共有する場合は、ソースデータベースからデータを一度ロードするだけで済みます。他のアプリケーションは、QVD ファイルからそのデータを使用できます。



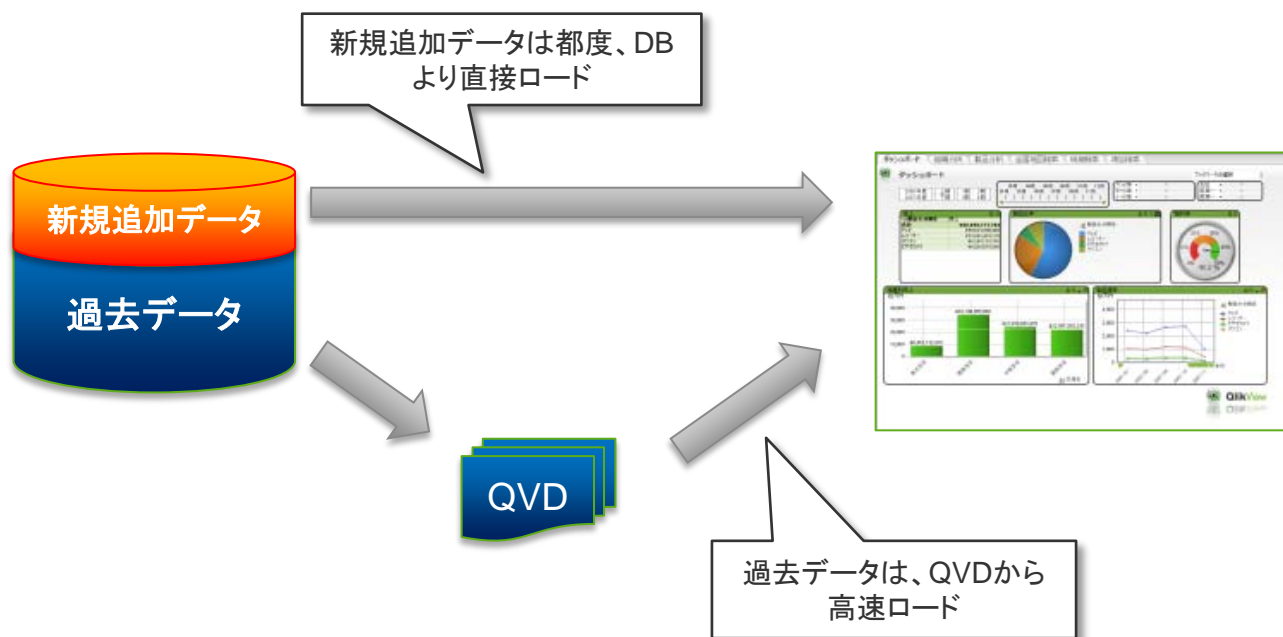
QVDの主な用途：複数QlikViewアプリケーションからのデータ統合

- QVWからQVWへのバイナリロードは1 QVWのみですが、QVD ファイルを使用すれば複数QlikView アプリケーションからのデータを統合できます。
- 例えば、各部署で個別に作成したデータを統合することができます。



QVDの主な用途：差分ロード

- QVD ファイルを経由することにより、増大するデータベースから新しいレコードだけをロードする増分ロードを実装できます。



QlikViewにおける階層型データ構造

- 単一階層(ダイレクトロード)
- 2階層 QVDパターン
- 3階層 QVDパターン

- 2階層 QVWパターン
- 3階層 QVWパターン

- 3階層混在パターン



単一階層(ダイレクトロード)

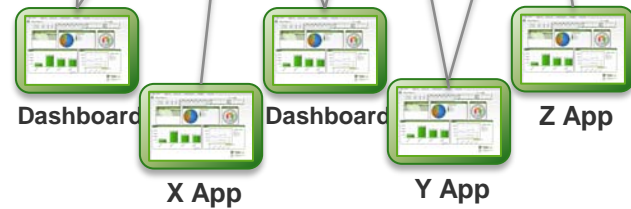
特長

- 階層としては最もシンプル
- 処理効率、メンテナンス効率ともに悪い
- 大規模展開時には推奨されない

ソースレイヤ



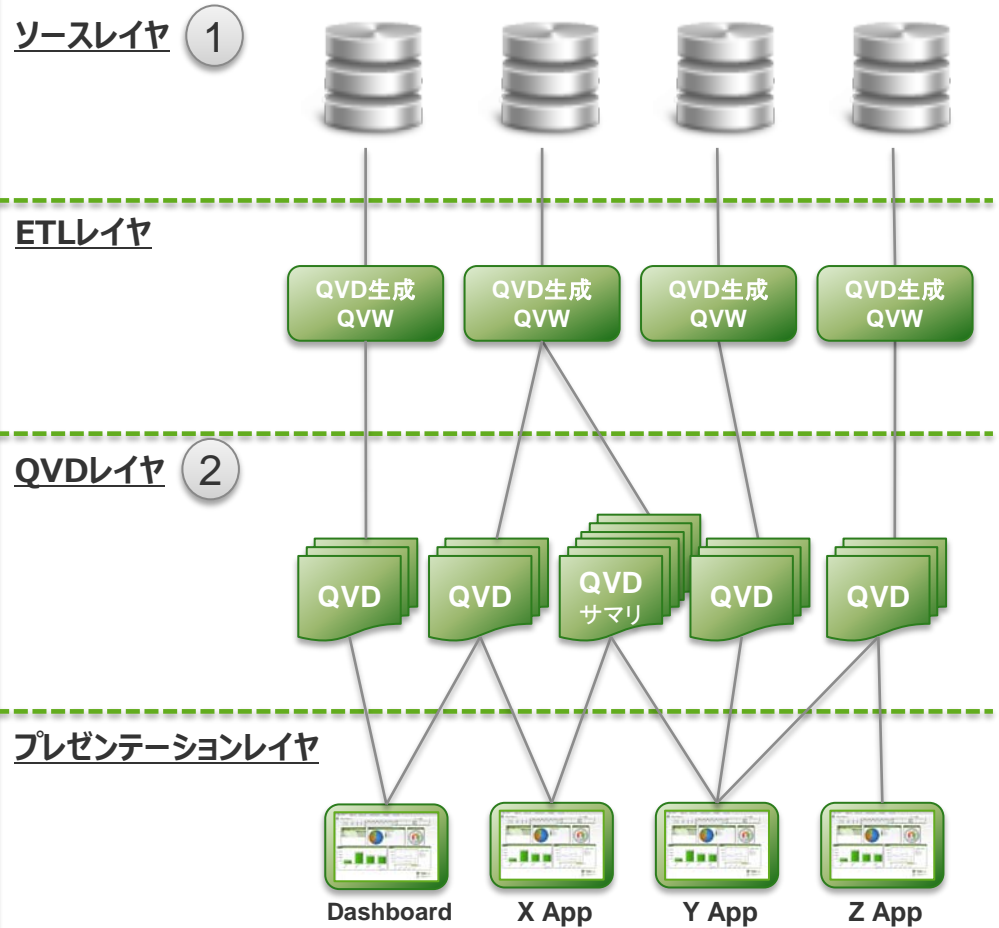
プレゼンテーションレイヤ



2階層QVDパターン

特長

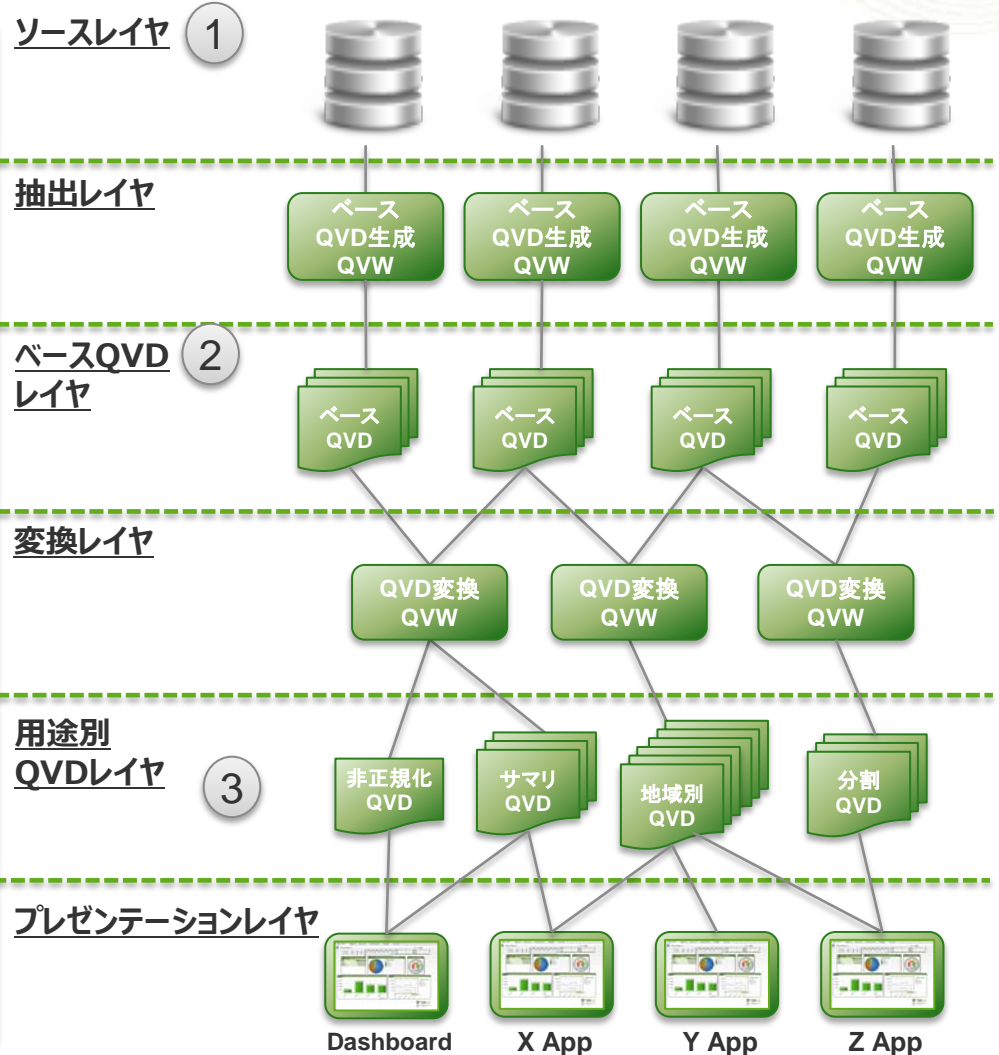
- QVDファイルにて第2階層を構成
- プレゼンテーションレイヤのアプリケーションは複数QVDよりデータを取り込む
- ソースデータへの都度接続は不要
- プレゼンテーションレイヤのQVW開発とデータ抽出レイヤのQVW開発を分離
- 前者はより簡単にQVW作成が可能
- エンドユーザーとIT部門の責任範囲を明確化



3階層QVDパターン

特長

- QVDファイルにて第2階層および第3階層を構成
- ベースQVDレイヤは正規化されたテーブルまたはソースDBのレプリカとする
(変換も集計もないため非常に高速に抽出)
- 用途別QVDレイヤは用途別に非正規化や集計、分割などの変換処理が行われたテーブルとなり、プレゼンテーションレイヤ開発に最適な形となる
- エンドユーザーとIT部門の責任範囲を明確化 (IT部門の管理範囲はより広範)



2階層QVWパターン

特長

- QlikMartファイルにて第2階層を構成
- QlikMartは、画面UIなしのQVWでデータマートとしての役割を担う
- プレゼンテーションレイヤのQVWからはバイナリロードで取り込む
- 差分ロード用途等でのQVWが不要な場合に最適
- プレゼンテーションレイヤのQVW開発とQlikMartレイヤのQVW開発を分離
- エンドユーザーとIT部門の責任範囲を明確化

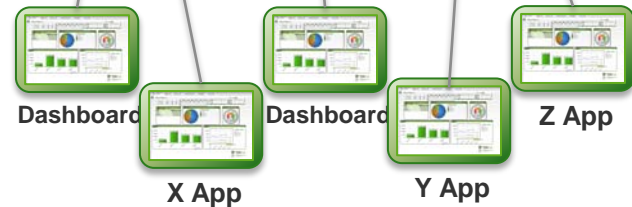
ソースレイヤ ①



QlikMartレイヤ ②



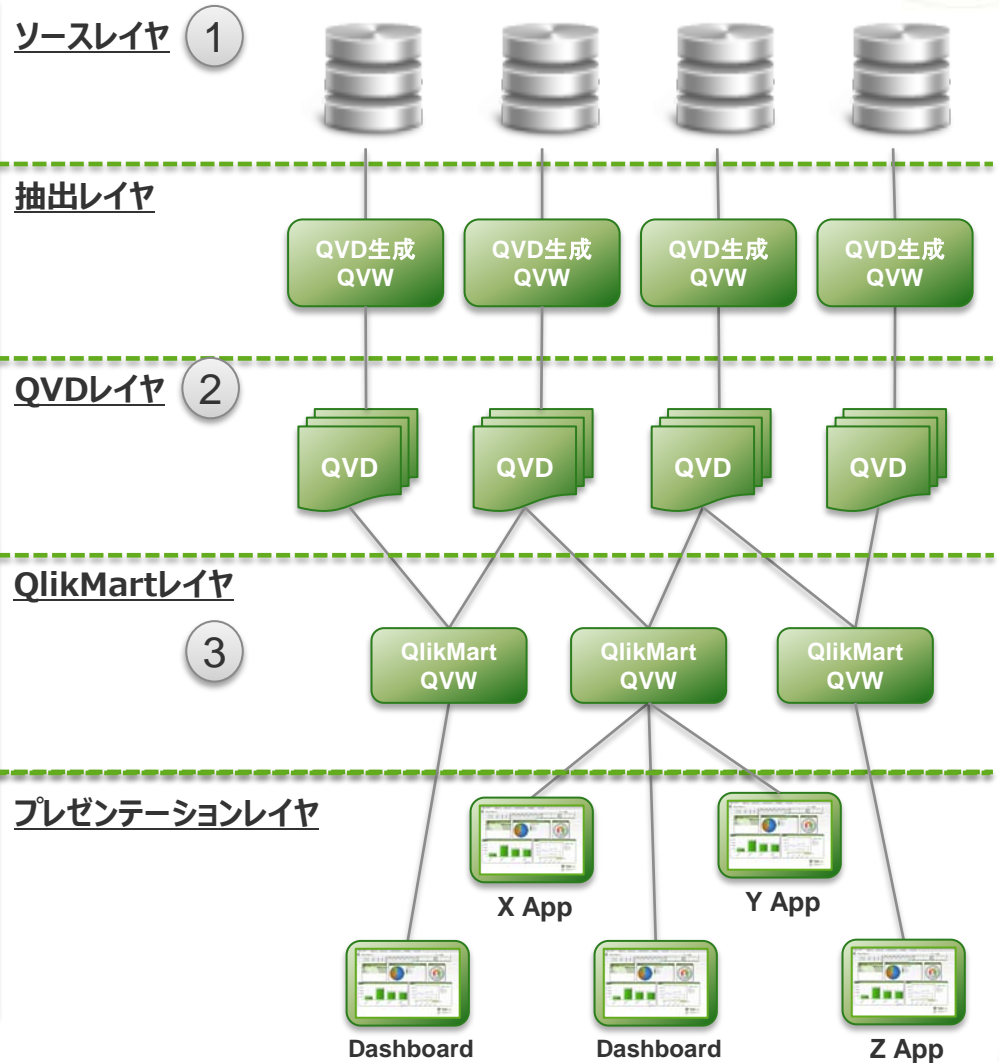
プレゼンテーションレイヤ



3階層QVWパターン

特長

- QVDファイルにて第2階層を構成
- さらにQVDからQlikMartにて第3階層を構成
- プレゼンテーションレイヤのQVWからはバイナリロードで取り込む
- プレゼンテーションレイヤのQVW開発とQlikMartレイヤのQVW開発を分離
- エンドユーザーとIT部門の責任範囲を明確化





Thank you!

