

Changez de vision et **prenez les bonnes décisions**



MIAGE - Cours Qlikview
Decideom - Jérôme Paul - Novembre 2016

Organisation du cours Qlikview

Partie cours (2h)

- Présentation
- Positionnement de Qlik dans le monde de la BI
- Architecture décisionnelle avec Qlikview
- Partie Script (ETL / Modélisation)
- Partie Restitution (Les bonnes pratiques)
- Partie Serveur (Administration)

Partie TP (4h)

- Création d'une application d'extraction (2h)
 - Alimentation à partir d'une base de données
 - Enrichissement à partir de fichiers
 - Champs calculés et règles de gestion
- Création d'une application de restitution (2h)



Présentation



Présentation Jérôme PAUL

Expériences

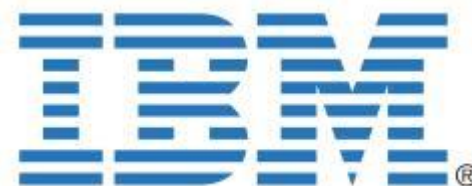
- | | | |
|---------------|-----------------------|-------------------------------------|
| ■ 1998 - 1999 | Consultant | <i>SSII Unis</i> |
| ■ 1999 - 2006 | Chef de projet BI | <i>SSII Business & Decision</i> |
| ■ 2006 - 2008 | Chef de projet BI | <i>Entreprise Kiloutou</i> |
| ■ 2008 - 2010 | Manager BI | <i>SSII Business & Decision</i> |
| ■ 2010 | Responsable Technique | <i>SSII Decideom</i> |



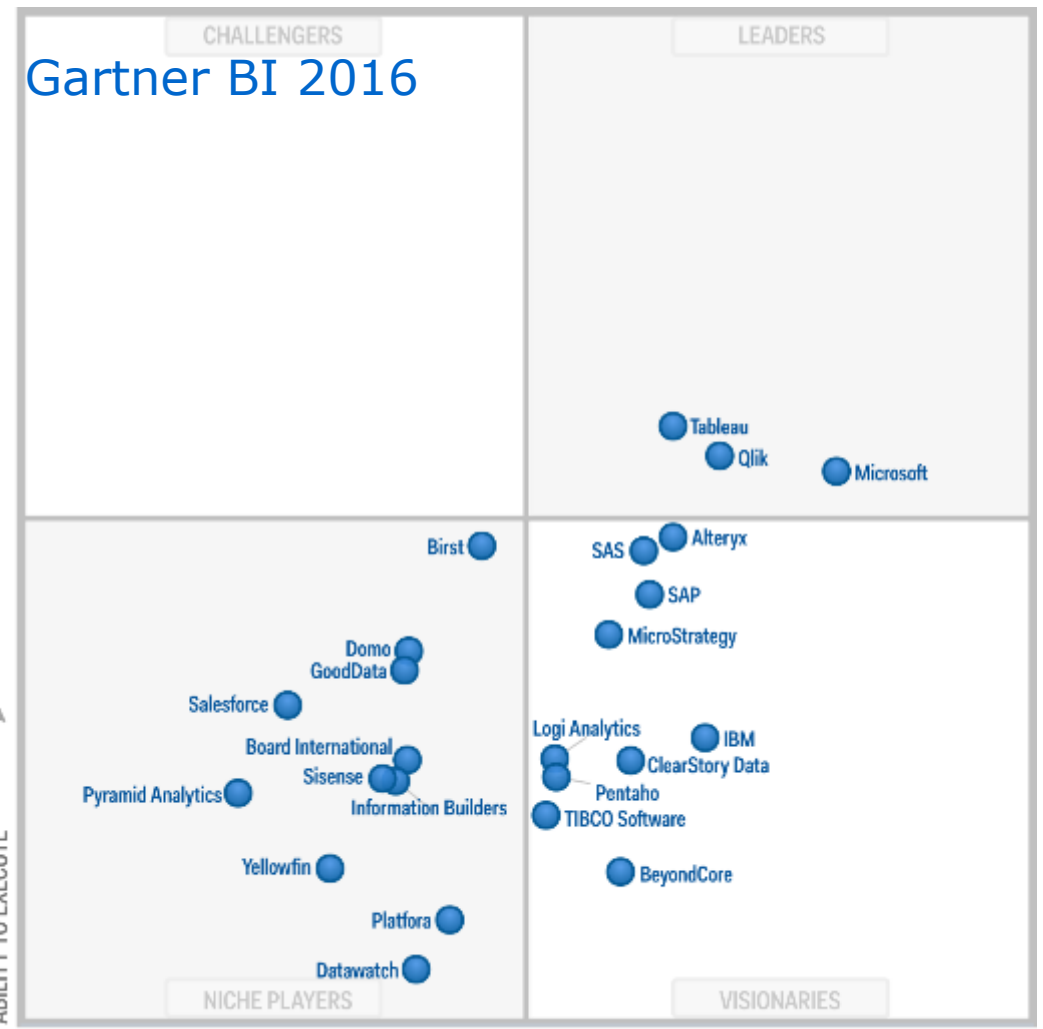
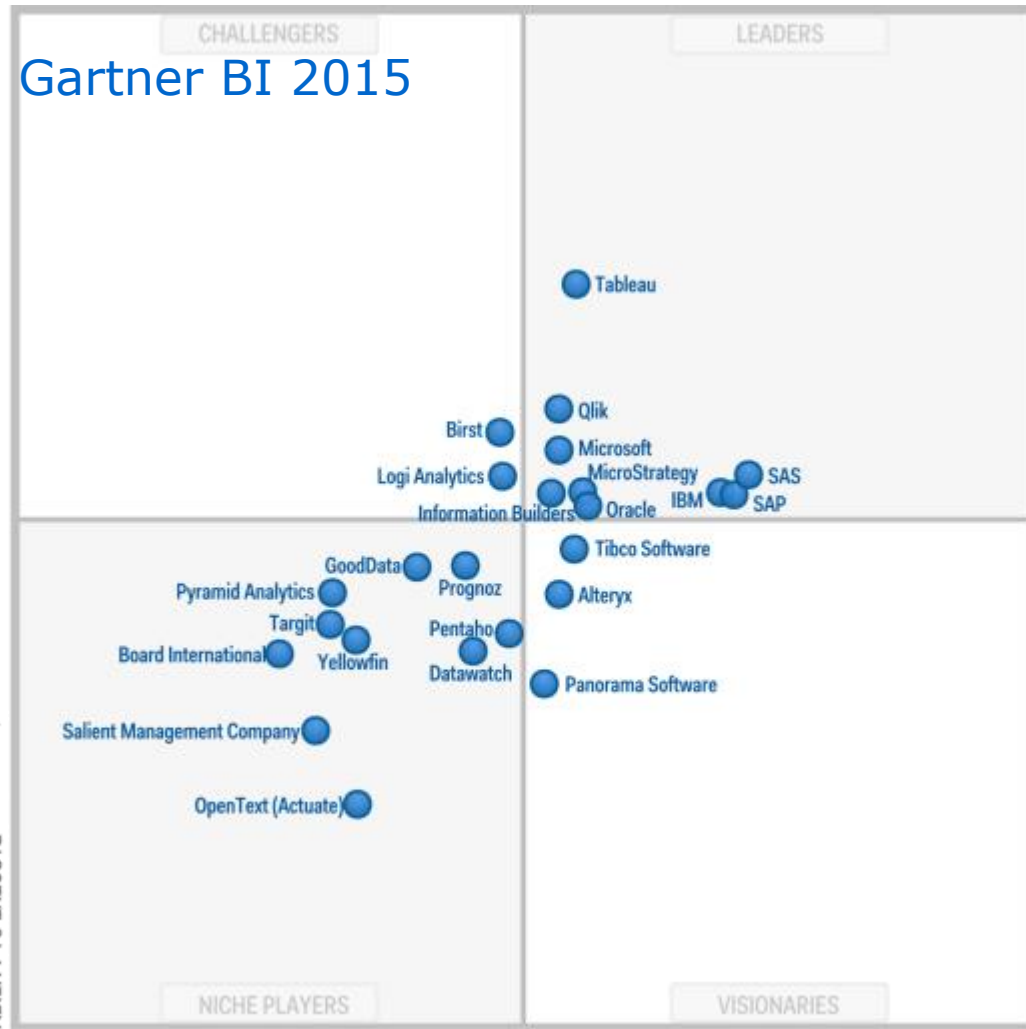
Positionnement de l'éditeur Qlik dans le monde de la BI



Les outils BI dans le nord



Positionnement des éditeurs



À propos de Qlik



Siège aux Etats-Unis, R&D en Suède, et Business Center sur chaque continent



- **Siège social**
- **Bureaux de vente**
- **Bureaux**
- **Associés**



À propos de Qlik

Quelques infos :

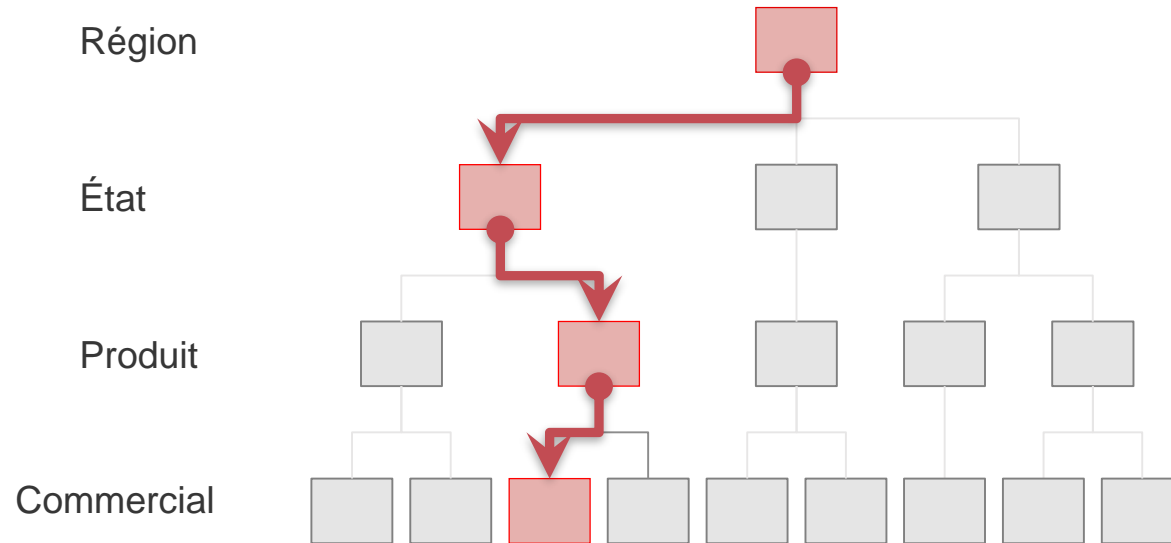


- Éditeur de logiciel de BI
- Fondée en 1993 à Lund (Suède)
- 2 500 employés à travers le monde
- R&D en Suède, Siège au USA (Radnor)
- 65 personnes en France
- 1 700 partenaires à travers le monde
- 39 000 clients à travers le monde
- CA 2015 : 612 M\$



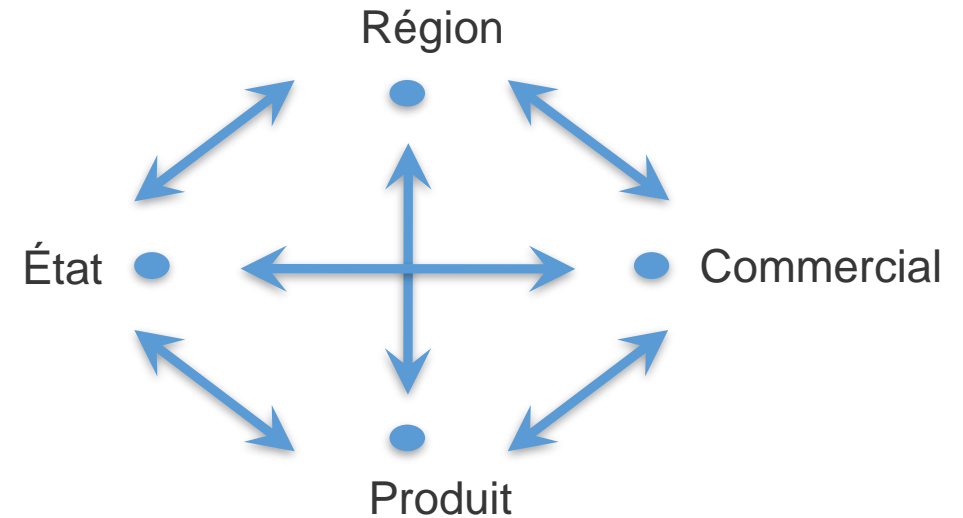
Point fort Qlikview : Expérience associative

Rapports et visualisations



- Chemins linéaires, prédéfinis

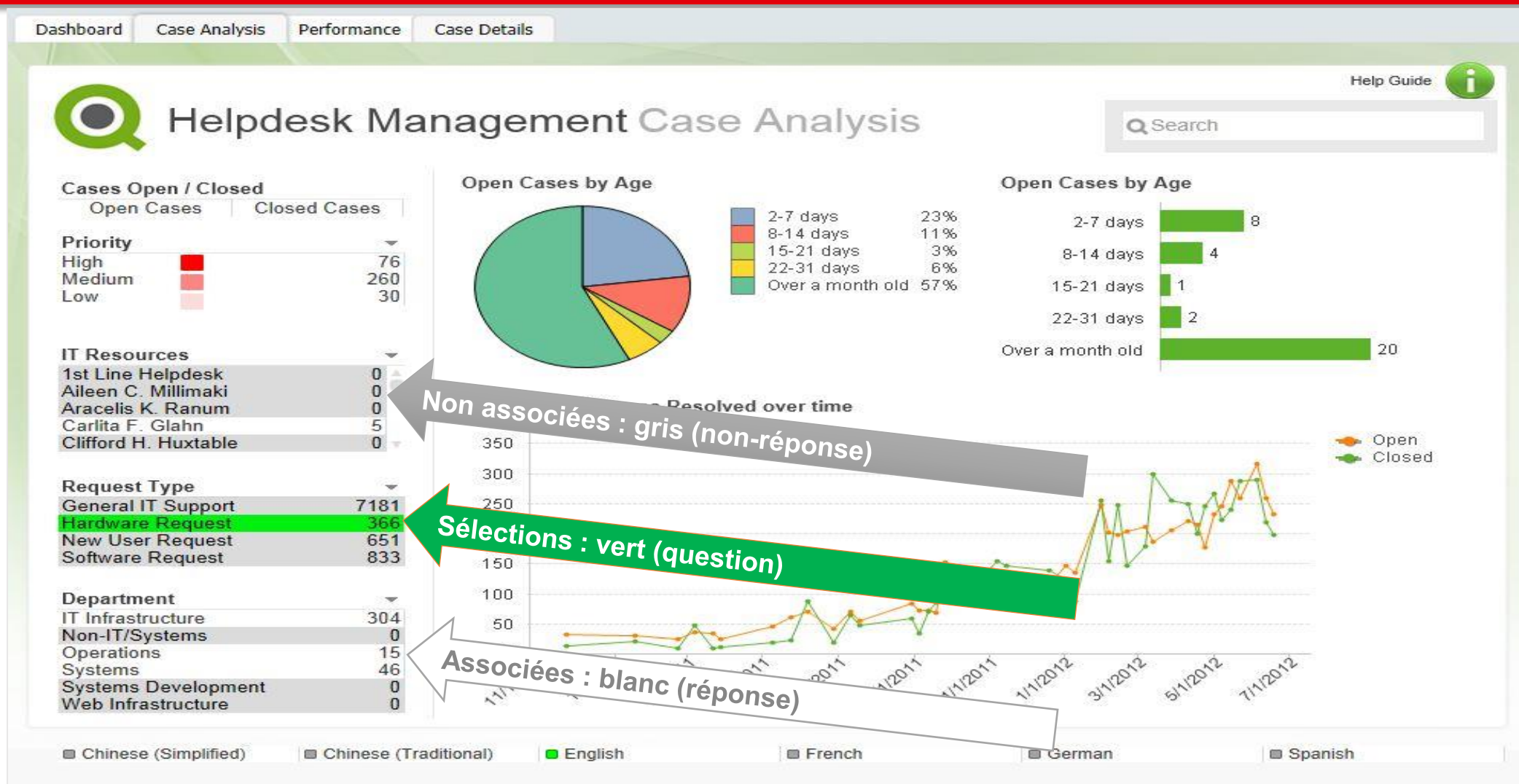
Applications QlikView



- Liberté d'exploration de l'utilisateur



Point fort Qlikview : Expérience associative



Environnement QlikView



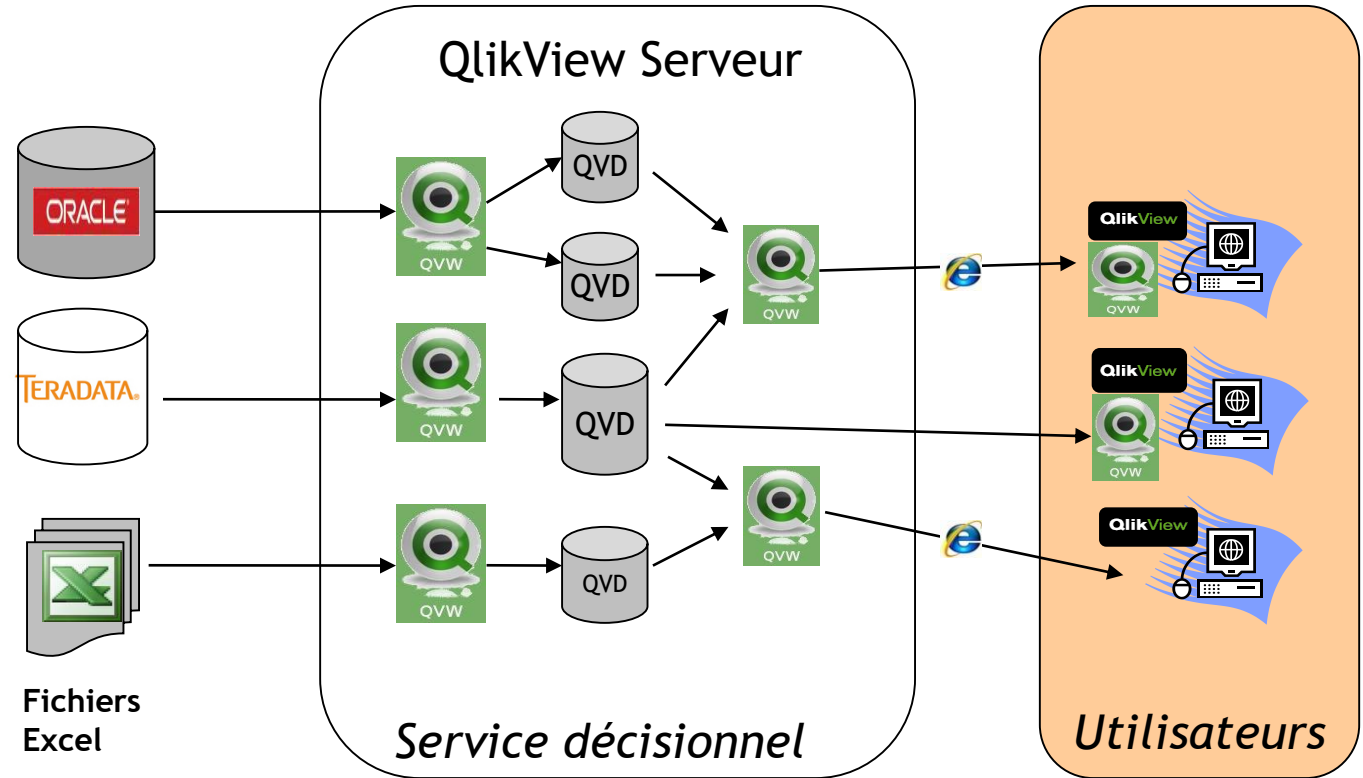
Architecture décisionnelle avec Qlikview



Architecture Qlikview

Les autres points forts :

- **Un ETL**
Langage AQL
= SQL Avancé
- **Base vectorielle**
Gain de stockage
- **Base in memory**
Accès rapide à la donnée



un QVD = une table

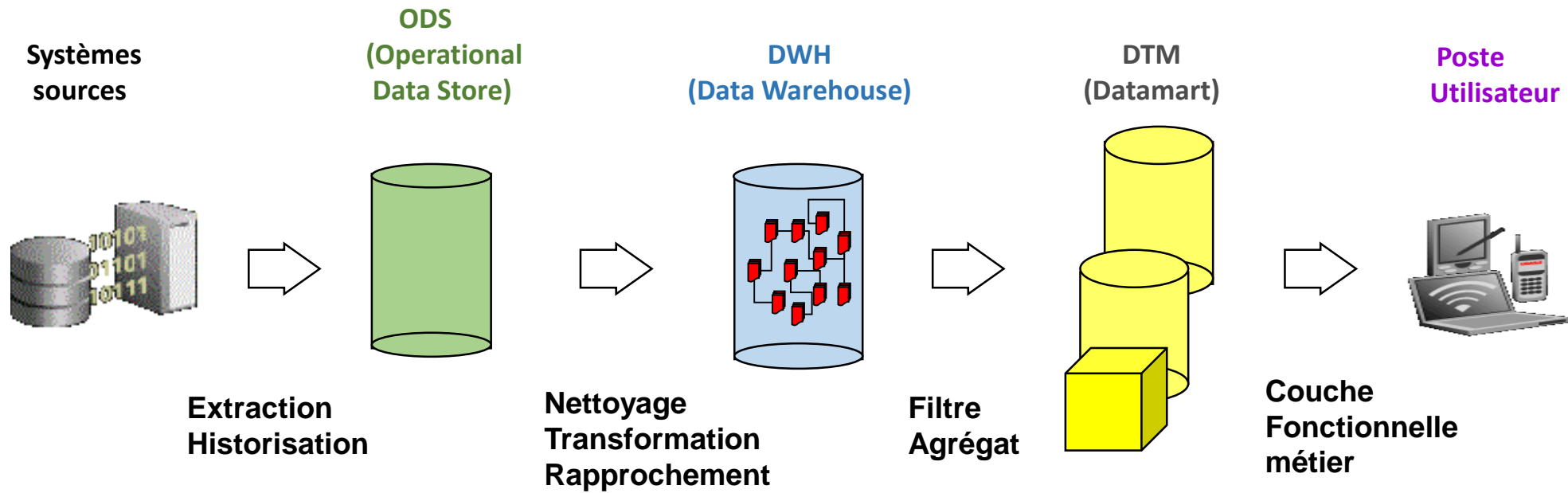
DWH $\Leftrightarrow \Sigma$ QVDs

une application QVW

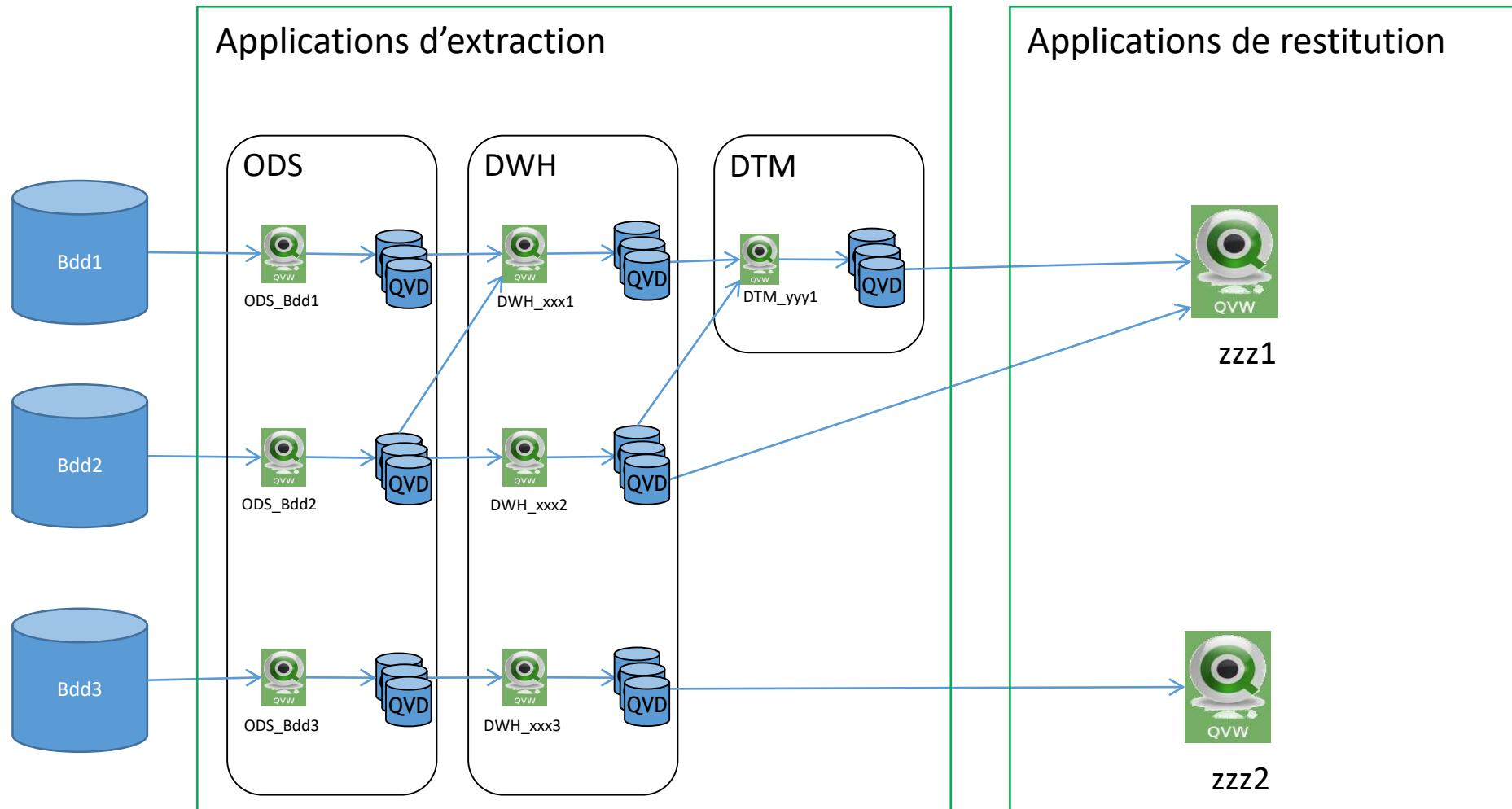
= un datamart



Architecture BI classique

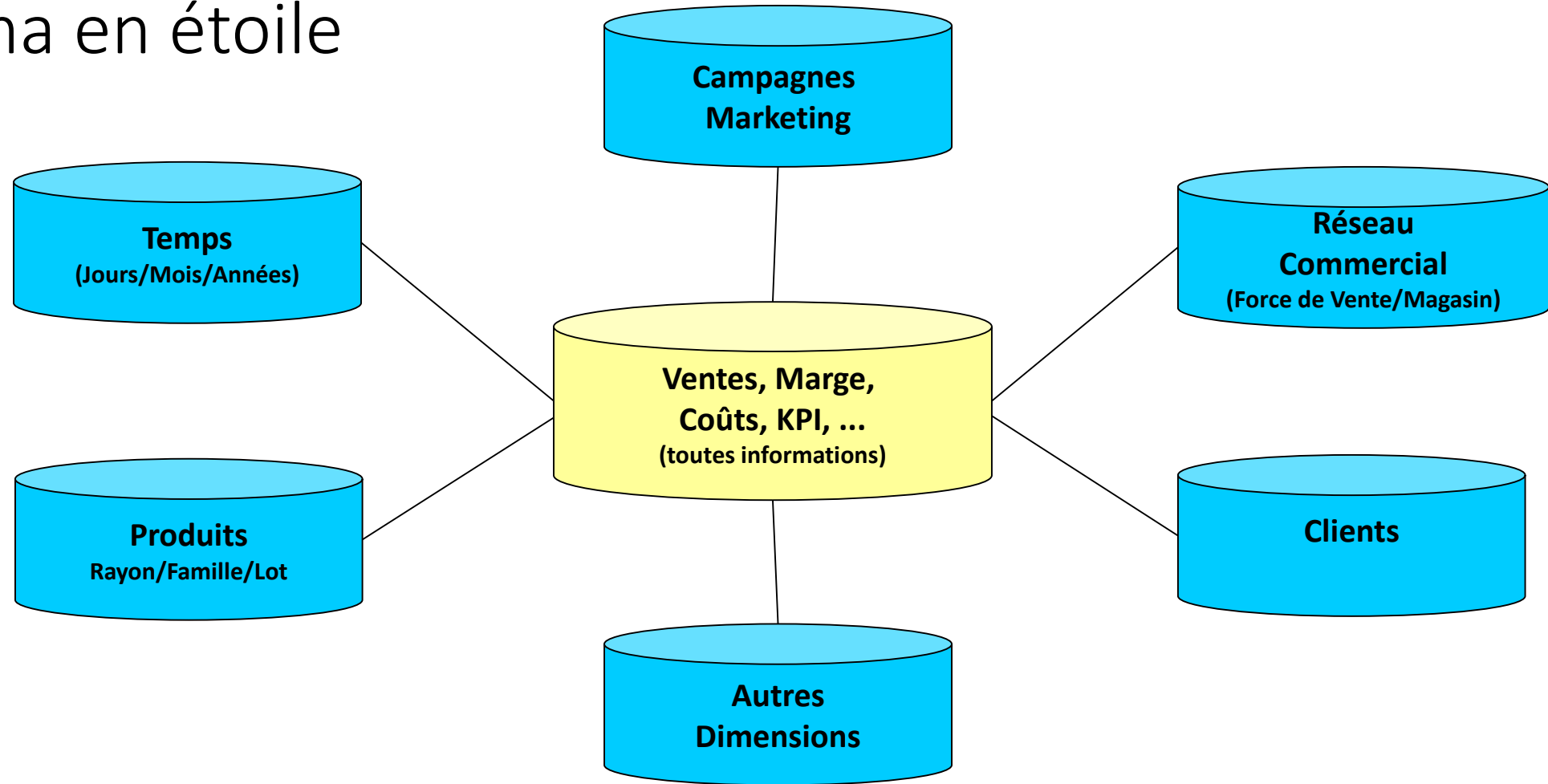


Architecture Qlikview



Modélisation d'une application de restitution

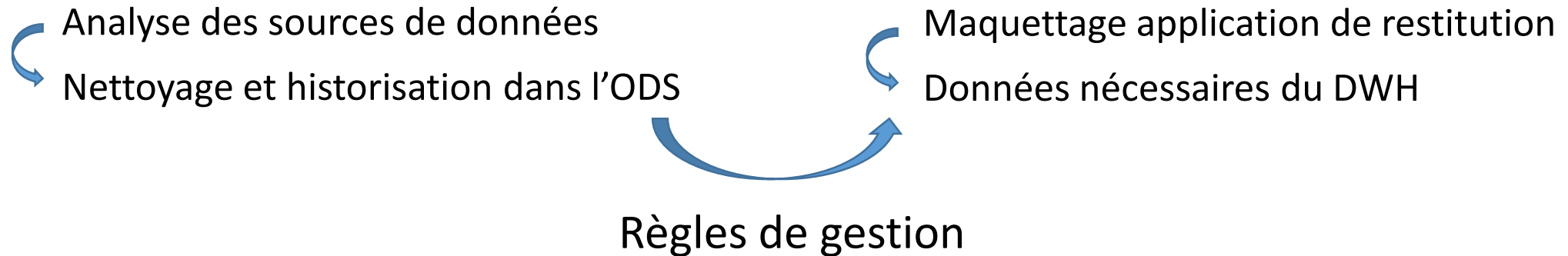
Schéma en étoile



Méthodologie Projet

Méthode projet classique : UML, Merise,

Méthode projet Qlikview :



Découpage en tâches simples pouvant être parallélisées selon la disponibilité des entrants (données, maquettes)

Mode de fonctionnement par itération avec un enrichissement successif des maquettes (évite l'effet tunnel)



Partie Script (ETL / Modélisation)



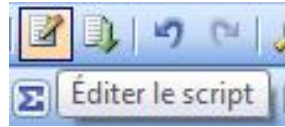
Plan de la partie Script

- Editeur de Script
- Connexion à une base de données
- Récupération des tables
- Ajout de fichiers (XLS, XML, TXT, ...)
- Modèle de données
- Analyse des chargements
- Création de données calculées sous Qlikview
- Scripts avancés (Join, Concatenate, Exists)
- Fichiers de données QVD



Editeur de script – Généralités

- C'est à partir de l'éditeur de script qu'il sera possible d'intégrer des données dans QlikView. Il est accessible depuis la barre d'outils via l'icône ci-dessous. Il est possible d'utiliser le raccourci (CRL+E)



- Vous utilisez le langage AQL (Associative Query Language) pour intégrer nos données. L'AQL ressemble fortement au SQL avec des fonctionnalités propres à QlikView
- Beaucoup d'assistants facilitent la génération du script
- Tout se passe dans le script, il n'y a aucune configuration ailleurs



Connexion à une source de données

Choix du type de connexion ODBC ou OLE DB

Pour des raisons de performances et pour faciliter le déploiement, il est préconisé d'utiliser la connexion OLEDB plutôt que la connexion ODBC. La chaîne de connexion sera générée automatiquement par QlikView

1

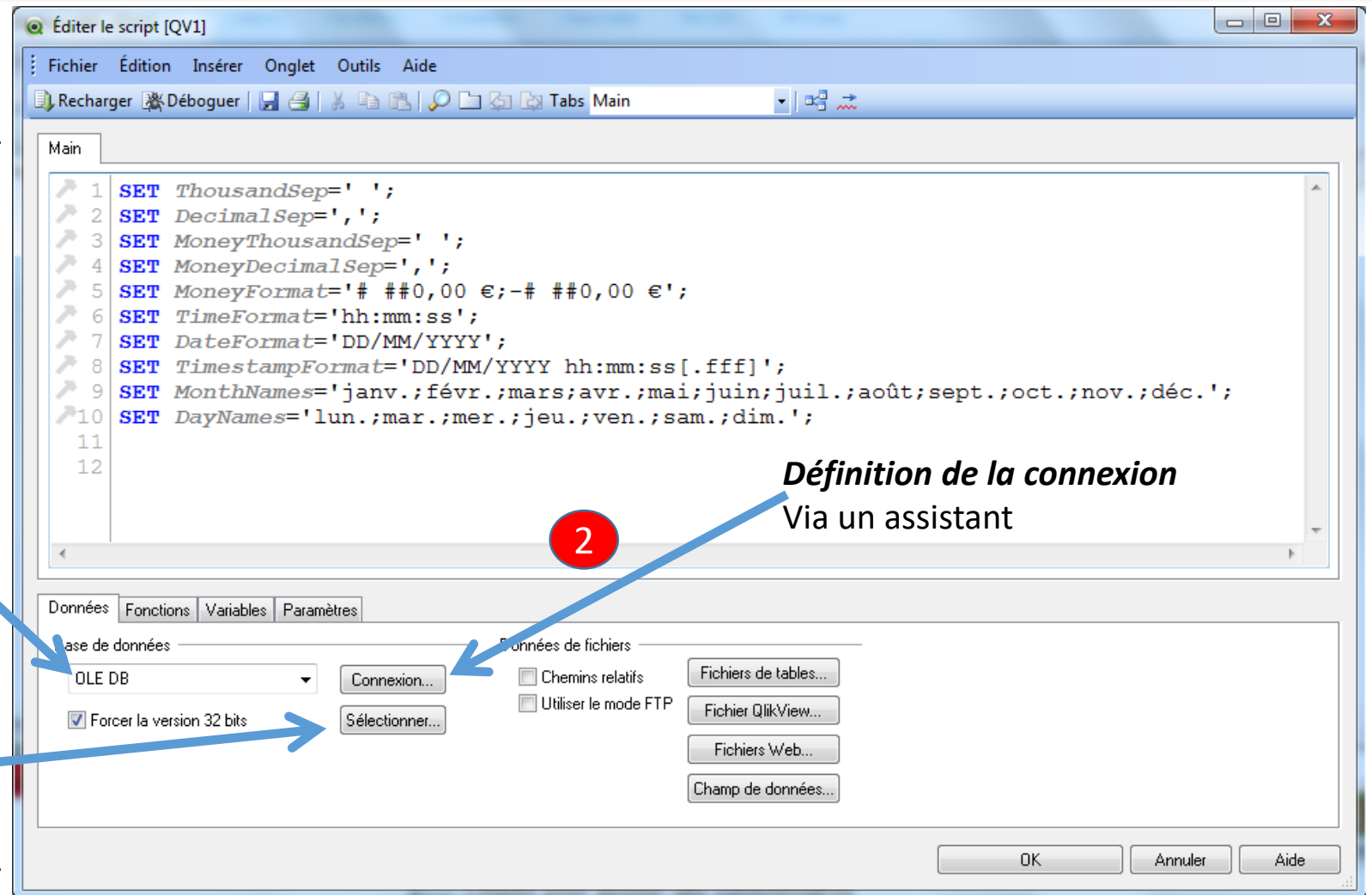
Définition de la connexion
Via un assistant

2

3

Choix des tables et des champs à récupérer

Pour des raisons de performances et pour faciliter



Récupération des tables (Assistant)

Choix des types de tables à afficher →

Choix de la table à intégrer →

Script généré →

Valide la sélection →

Choix des champs à sélectionner
Ne jamais utiliser *

Options de génération du script
Utiliser Colonne et cocher « Avant Load »

Ajoute la table sélectionnée
Permet de passer ensuite à une autre table

```
LOAD CategoryID,
CategoryName,
Description;
SQL SELECT *
FROM Categories;
```



Ajout de fichiers

Sélection du fichier à récupérer

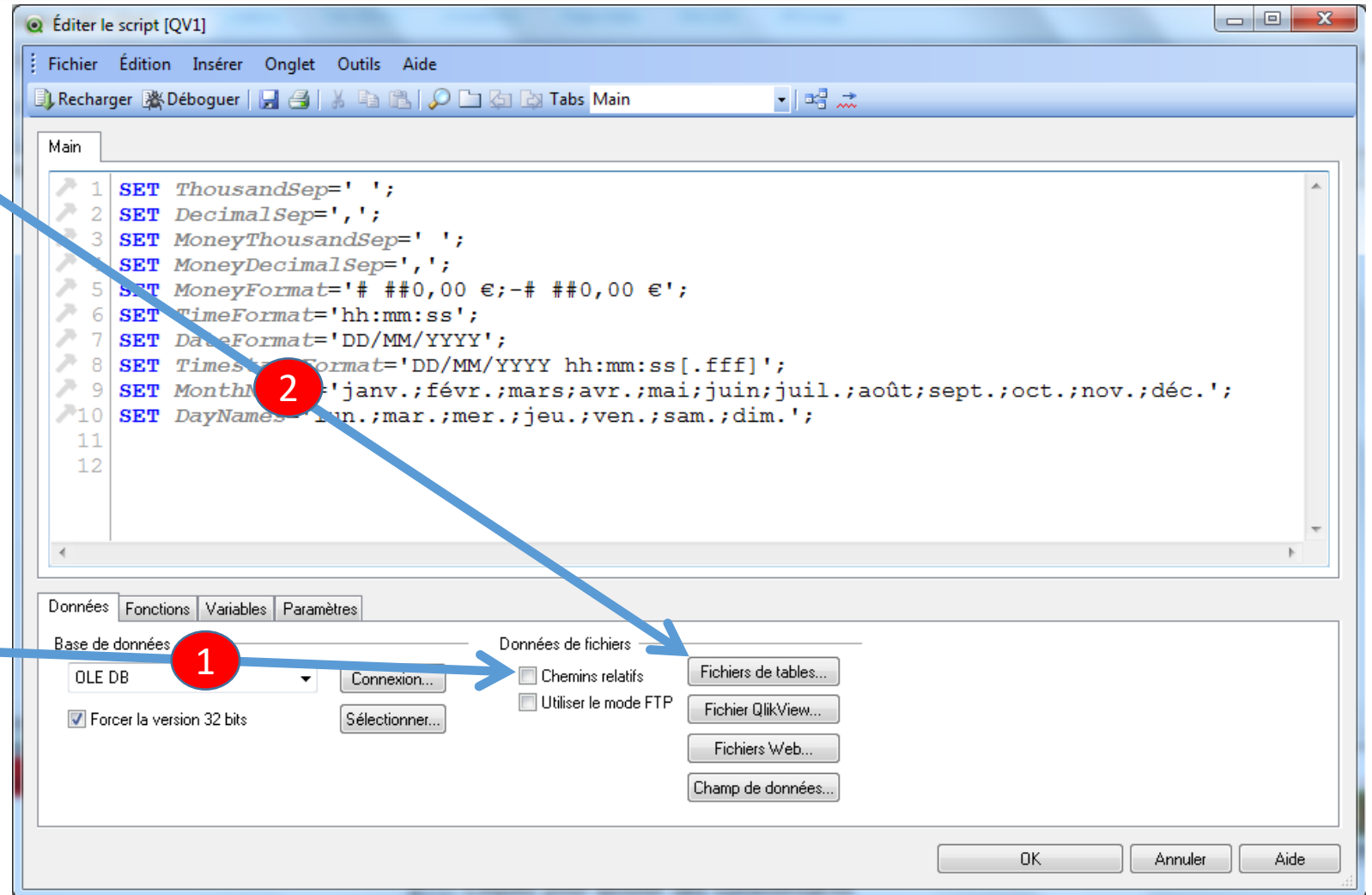
Lance un assistant pour guider intégration

On peut charger des fichiers de type :

- TXT délimité par un séparateur,
- TXT à champs fixes,
- DIF (gros système),
- XLS/XLSX,
- Html,
- XML,
- QVD/QVX (format propriétaire Qlikview)

Choix du type d'accès pour le répertoire

Chemins relatifs prend en compte le répertoire où est stocké l'application Qlikview



Editeur de script - Les instructions simples

- Définir une variable : **SET** (variable en dur) / **LET** (variable résultat d'une fonction ou d'un calcul)

```
SET Chemin='C:\Projet\';
```

```
LET vAnnéeCourante = Year (today ());
```

- Charger un fichier : **Load**

```
Load * from 'c:\userfiles\data2.txt'(norme ANSI, txt,le délimiteur est « \ t »,étiquettes incluses);
```

- Charger une table : **SQL**

```
SQL SELECT DISTINCT
```

```
    I. AddressID, Nom, Adresse, PostalCode
```

```
FROM [Facture] I, [Adresse] A WHERE I.FactureType is not null and I.AddressID = A.AddressID;
```

- Renommer un champ : **AS**

```
Load
```

```
    Capitale as 'Nom de la Capitale',
```

```
    Pop as Population
```

```
From Pays. CSV (ANSI, txt...
```



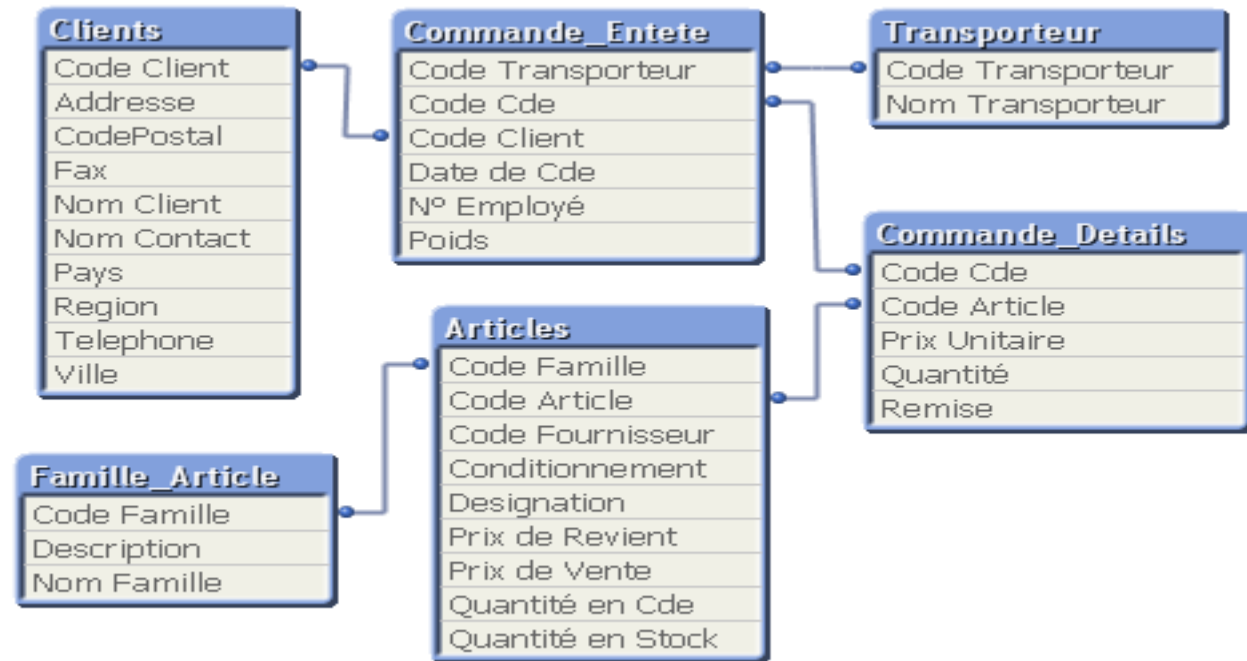
Les structures de données

Le visionneur le table permet d'observer et d'analyser les données chargées dans l'application Qlikview
CTRL + T permet d'y accéder

Les Jointures entre les tables sont créées automatiquement par homonymie

Obligation de renommer les champs pour créer ou casser des jointures

Info : Laisser la souris sur un champ ou l'entête d'une table permet d'obtenir des infos supplémentaires



Analyse des chargements - Tables synthétiques

- **Gestion des tables synthétiques**

Les tables synthétiques sont générées automatiquement par QlikView lorsqu'il détecte dans le modèle de données des tables ayant plusieurs clés (champ commun à plusieurs tables).

Elles provoquent une augmentation de la RAM et de la CPU utilisées ; il faut les supprimer.

Pour les supprimer, 2 solutions :

Il s'agit d'une information redondante : Suppression de l'un des champs

Il s'agit d'une vraie clef composée : Utilisation d'une fonction permettant d'utiliser un clef unique et supprimer ou renommer les constituants de la clef selon le besoin.

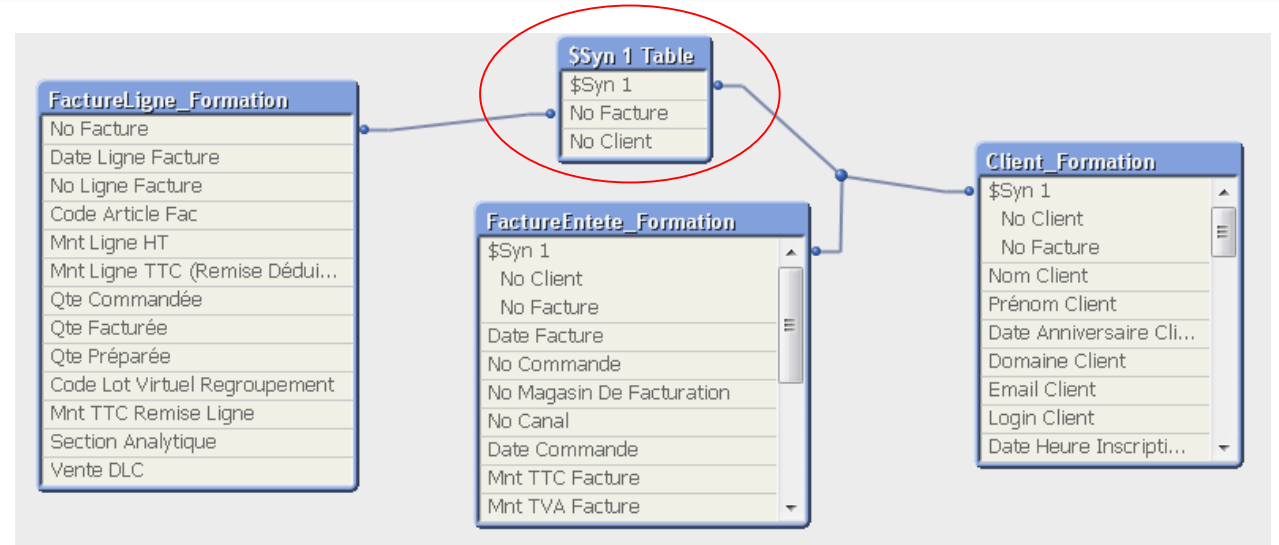
Exemple de fonction :

Concaténation :

[No Facture] & ' ' & [No Client] as Clef

Fonction de Hashing :

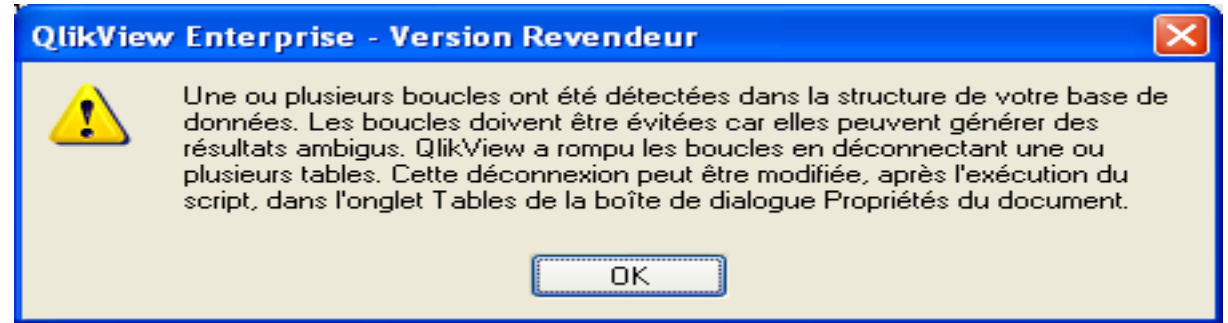
Hash128([No Facture], [No Client]) as Clef



Analyse des chargements - Boucle

- ***Gestion des boucles***

Les boucles ne sont pas autorisées dans Qlikview, si votre modèle en contient, l'application n'est pas utilisable.



Pour les supprimer, 2 solutions :

Il s'agit d'une information redondante : Suppression de l'un des champs

Il s'agit d'une information utile : renommage de l'un des champs



Analyse des chargements - Table système

- **L'objet table système**

Utilisable dans la partie restitution, cet objet graphique permet de visualiser les croisements entre les tables et les champs

Utilisé quand le modèle devient illisible.

Pour l'afficher :

Faire un clic droit dans une feuille et choisir ajout d'un nouvel objet, **Table système**.

Un tableau croisé dynamique sera créé avec les dimensions *\$Field* et *\$Table*.

Table système							
\$Field	\$Table	Clients	Transporteurs	Articles	Familles_Ar...	entetes_Cde	Details_Cde
Code Client		Code Client	-	-	-	Code Client	-
Code...		-	Code...	-	-	Code...	-
Code Article		-	-	Code Article	-	-	Code Article
Code Famille		-	-	Code Famille	Code Famille	-	-
Code Cde		-	-	-	-	Code Cde	Code Cde
Adresse		Adresse	-	-	-	-	-
CodePostal		CodePostal	-	-	-	-	-
Fax		Fax	-	-	-	-	-
Nom Client		Nom Client	-	-	-	-	-
Nom Contact		Nom Contact	-	-	-	-	-
Pays		Pays	-	-	-	-	-
Region		Region	-	-	-	-	-
Telephone		Telephone	-	-	-	-	-
Ville		Ville	-	-	-	-	-
Nom...		-	Nom...	-	-	-	-
Code...		-	-	Code...	-	-	-
Conditionnem...		-	-	Conditionne...	-	-	-
Designation		-	-	Designation	-	-	-
Prix de Revient		-	-	Prix de...	-	-	-
Prix de Vente		-	-	Prix de Vente	-	-	-
Quantité en Cde		-	-	Quantité en...	-	-	-
Quantité en...		-	-	Quantité en...	-	-	-
Description		-	-	-	Description	-	-
Nom Famille		-	-	-	Nom Famille	-	-
Code Vendeur		-	-	-	-	Code Vendeur	-
Date de Cde		-	-	-	-	Date de Cde	-
année		-	-	-	-	année	-
Mois		-	-	-	-	Mois	-
jour		-	-	-	-	jour	-
Poids		-	-	-	-	Poids	-



Analyse des chargements - Les champs systèmes

Les champs systèmes sont utilisables dans la partie restitution :

- *\$Field* montre les noms de tous champs chargés
- *\$Table* montre les noms de toutes les tables chargées
- *\$Rows* montre le nombre de d'enregistrements dans les tables
- *\$Fields* montre le nombre de champs dans les diverses tables
- *\$FieldsNo* montre les positions des champs dans les tables (le numéro de colonne)
- *\$Info* montre les noms des informations sur les tables chargées



Création de données calculées sous Qlikview

- Resident

Permet de créer une table à partir d'une table précédemment chargée.

- IF (condition, résultat, sinon)

Permet de créer un champ à partir d'une condition.

Exemple de création d'une table interne avec la commande « Resident » :

Commerciaux :

Load [Num Employé],

Nom & ' ' & Prénom as [Nom Commercial],

IF(Fonction=1,'Sales Manager','Sales Coordinator') as [Titre Commercial]

Resident Employés where Fonction=1 OR Fonction=2;



Création de données calculées sous Qlikview

- Agrégation des valeurs

Comme en SQL, on peut agréger les données d'une table dans une nouvelle table.

Exemple d'agrégation :

CommandesParAnnée :

Load

Année, CommandeID,

Sum (Montant_net_Ligne) as Montant_net_Année

RESIDENT

Entetes_Cde

Group by

Année, CommandeID;



Script avancé - Join

- Jointure : JOIN

La commande [Outer/Inner/Left/Right] JOIN permet de fusionner les champs de 2 tables.

Syntaxe1 : Définition_Table1 JOIN Définition_Table2

La définition des 2 tables se suivent

Syntaxe2 : JOIN (Nom_Table1) Définition_Table2

Table1 est définie bien avant dans le script

```
/* Commentaire : chargement de la source de Variation des ventes avec jointure JOIN*/
```

```
Entetes_Cde_2_Ans:
```

```
Load      `Code Cde`, Année
```

```
RESIDENT Entetes_Cde
```

```
Where     Année = 2013 or Année = 2014; /*Nous chargeons uniquement les années 2013 et 2014*/
```

```
Left Join (Entetes_Cde_2_Ans)          /* Jointure à gauche que le champ commun 'Code Cde' */
```

```
Load
```

```
      `Code Cde`, Montant_net_Ligne
```

```
RESIDENT Details_Cde;
```



Script avancé - Concatenate

- Concaténation : CONCATENATE

La commande Concatenate permet de fusionner les enregistrements de 2 tables.

Syntaxe1 : Définition_Table1 CONCATENATE Définition_Table2

La définition des 2 tables se suivent

Syntaxe2 : CONCATENATE (Nom_Table1) Définition_Table2

Table1 est définie bien avant dans le script

```
/* Entetes_Cde contient les enregistrements des tables Entetes_Cde_2013 et Entetes_Cde_2014 */
```

```
Entetes_Cde:
```

```
Load `Code Cde`, Année, Montant_net_Ligne
```

```
RESIDENT Entetes_Cde_2013;
```

```
Concatenate
```

```
Load `Code Cde`, Année, Montant_net_Ligne
```

```
RESIDENT Entetes_Cde_2014;
```

```
Drop Tables Entetes_Cde_2013, Entetes_Cde_2014; /* Suppression de tables devenues inutiles */
```



Script avancé - Exists

- Exists

Syntaxe : Exists(champ [, expr])

S'utilise dans une clause WHERE ; Cette fonction détermine si la valeur d'un champ existe dans un champ parmi les données déjà chargées.

champ est un nom ou une expression de chaîne qui donne le nom d'un champ existant.

Expr est une expression qui fournit la valeur à chercher dans le champ spécifié.

Exemple :

Vendeur:

LOAD

EmployeeID,

nom as nom_vendeur

Resident Employé

Where

Exists(Empvendeur,EmployeeID);



Fichiers de données QVD

- Store

Store permet d'exporter une table (tous ses champs ou une partie) dans un fichier .QVD

La syntaxe pour la commande :

```
store [( * | <liste de champs> )] from <table> into <nom du fichier> ;
```

Un fichier QVD contient une table de données exportées de QlikView. QVD est un format propriétaire de QlikView.

Lire les données à partir d'un fichier QVD est en général 10-100 fois plus rapide qu'une lecture à partir d'une autre source.

4 utilisations principales des QVD :

Augmentation de la vitesse de chargement

Diminution de la charge sur les serveurs de base de données

Consolidation des données des applications multiples QlikView

Chargement incrémental



Partie Restitution (Les bonnes pratiques)



Application Restitution - Les bonnes pratiques

- Organisation

Pour bien organiser des applications, il faut respecter le découpage suivant :

Une application = une problématique métier

- Contenu

Pour optimiser l'application, elle ne doit contenir que les champs utilisés dans les graphiques et listes de sélections ; tout ce qui n'est pas utile ou ayant servi à effectuer des calculs, doit être retiré de l'application (mise en commentaire des champs lors de l'import des QVDs)



Application Restitution - Les bonnes pratiques

- Paramétrage de l'application de restitution

Pour uniformiser les applications, le plus simple est d'utiliser une application modèle.

Cette application se trouvera sur le serveur Qlikview dans un répertoire partagé

Exemple : D:\QV_PROD\Documentation\

Pour faciliter la création des objets graphiques, il est fortement conseillé de :

Prédéfinir les propriétés des objets au niveau du document (format, tri, ...)

Préciser pour chaque champ s'il est de type dimension ou de type indicateur.



Les feuilles - Les bonnes pratiques

- Organisation des feuilles

Pour bien organiser une application de restitution, il faut respecter le découpage suivant :

Une feuille = une analyse métier

La première feuille contiendra le descriptif de l'application et les explications fonctionnelles. Elle ne contiendra pas de graphique.

Les feuilles suivantes seront ordonnancées de celle ayant l'analyse la plus générale à celle ayant l'analyse la plus détaillée.

Il faut éviter d'avoir trop de feuilles dans une application ; maximum 10 feuilles.



Les feuilles - Les bonnes pratiques

- Contenu des feuilles

Toujours mettre la sélection active en haut à gauche de chaque feuille (sauf sur la première)

Toujours préciser la dernière date de rechargement de l'application

Pour l'organisation des listes de sélection, de préférence on mettra toujours :

- Le ou les axes temps en haut

- Les autres listes de sélections seront à gauche

- Les objets identiques existants dans plusieurs feuilles, devront être positionnés au même endroit.

- Les listes de sélection à choix unique seront de type Multibox

Une feuille ne doit pas être surchargée en graphiques ; elle doit être lisible en prenant en compte la résolution des postes utilisateurs (ex : 1024x768) et éventuellement la résolution des tablettes si elles sont utilisées.

On doit pouvoir accéder directement à tous les graphiques sans devoir les rechercher.

Il faut éviter d'avoir trop de graphiques ouverts en même temps sur une feuille; donc grand maximum 8 petits graphiques ou 4 grands.



Les dimensions et expressions - Les bonnes pratiques

- Dimension calculée

L'utilisation de dimension calculée dans la partie restitution est interdite.

Les dimensions dans les objets (Liste de sélection, Liste de recherche, graphique, ...) doivent se faire uniquement sur des champs existants dans l'application (calculés dans le script).

Exception : 2 fonctions sont utilisables dans une dimension calculée : Class et Aggr.

En cas de besoin, il faut demander l'ajout d'un champ correspondant au calcul souhaité directement dans le QVD de la dimension concernée.

- Expression calculée

Il faut éviter l'utilisation des expressions calculées complexes ; pour cela il faut ajouter le champ correspondant (ou de ses constituants) dans la table de faits concernée pour simplifier le calcul.



Partie Serveur (Administration)



Architecture Serveur

Selon les besoins,
Qlikview peut s'utiliser dans plusieurs architectures évolutives.

Différentes architectures existantes :

- Le mode autonome
- Le mode serveur
- Le mode serveur/publisher



Architecture Serveur

QlikView en mode Autonome



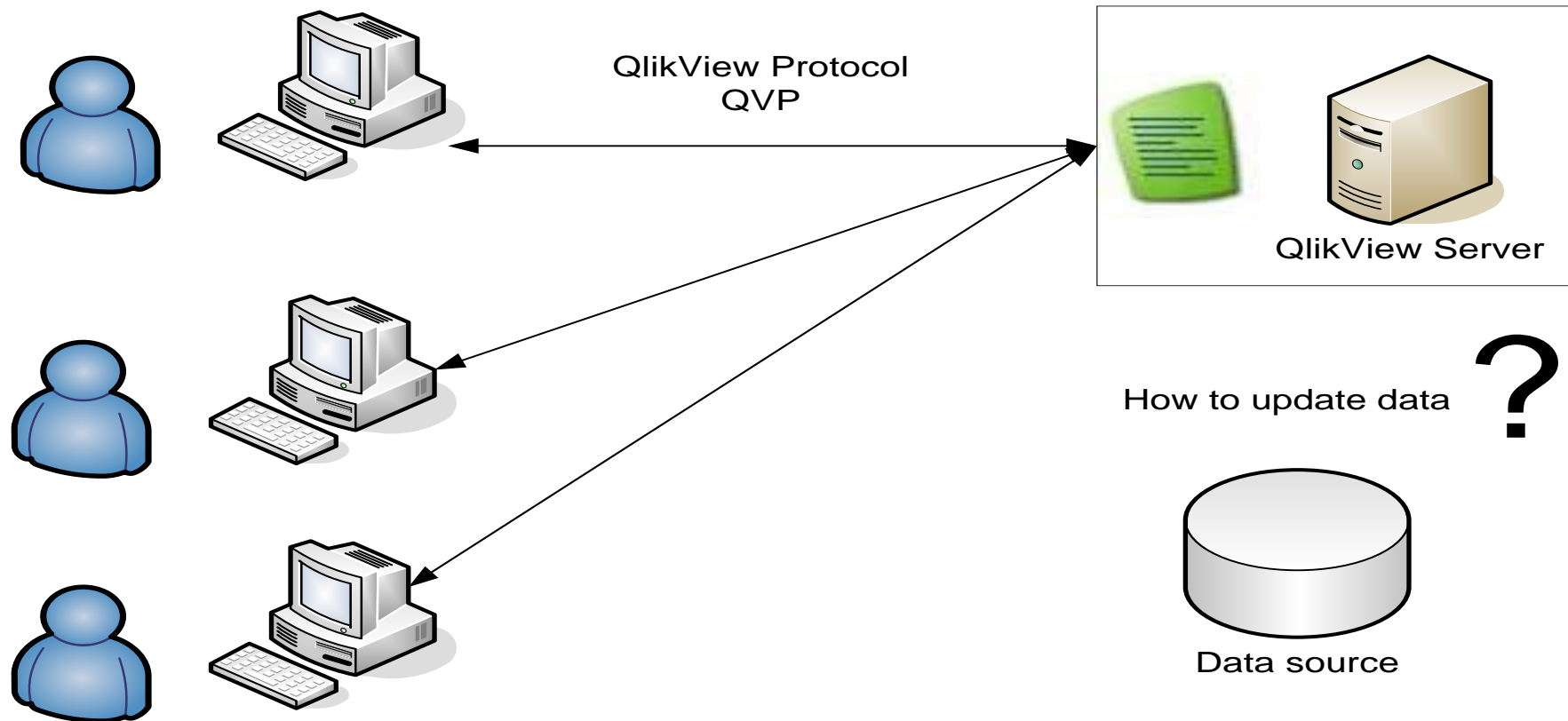
Utilisation du client lourd : création de .QVW

- Version gratuite : complète mais pas d'échange de .QVW
- Version payante : échange possible de .QVW



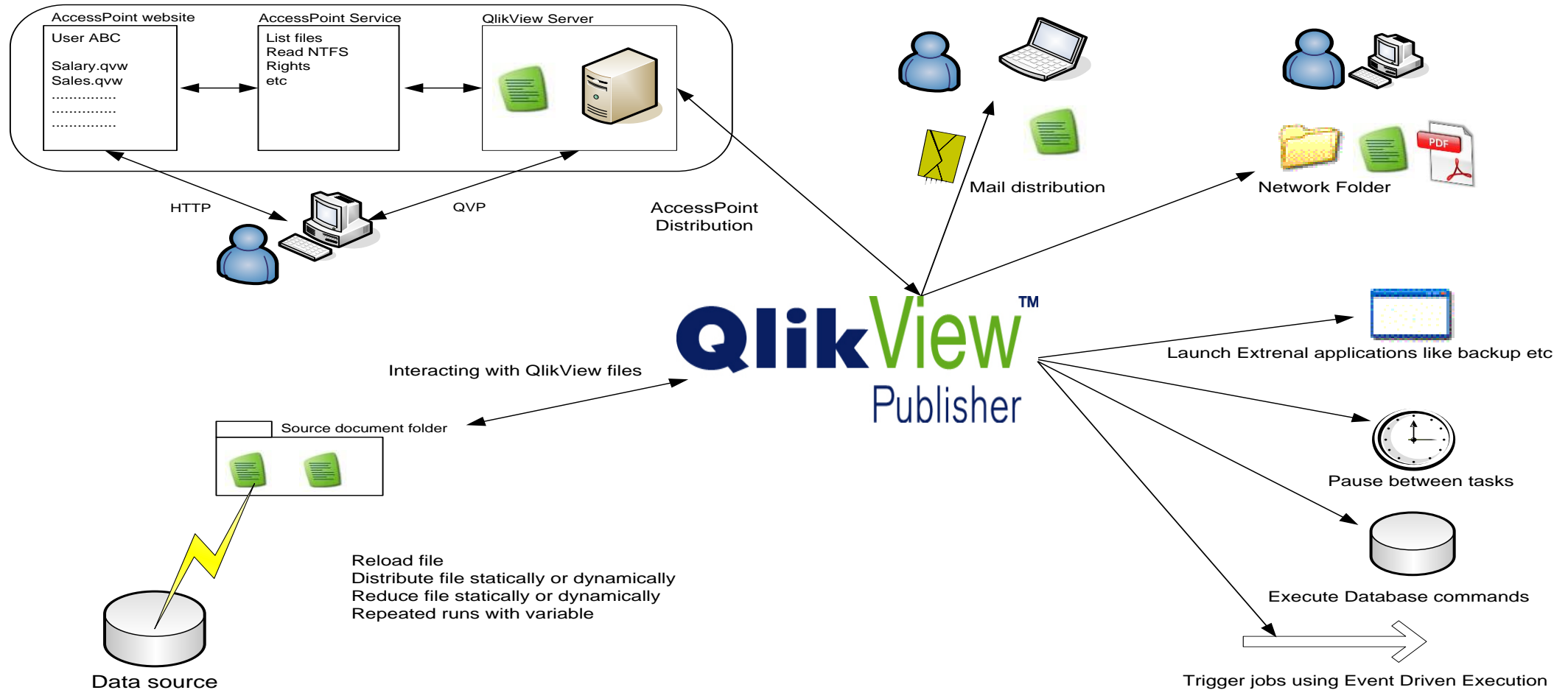
Architecture Serveur

QlikView en mode Serveur

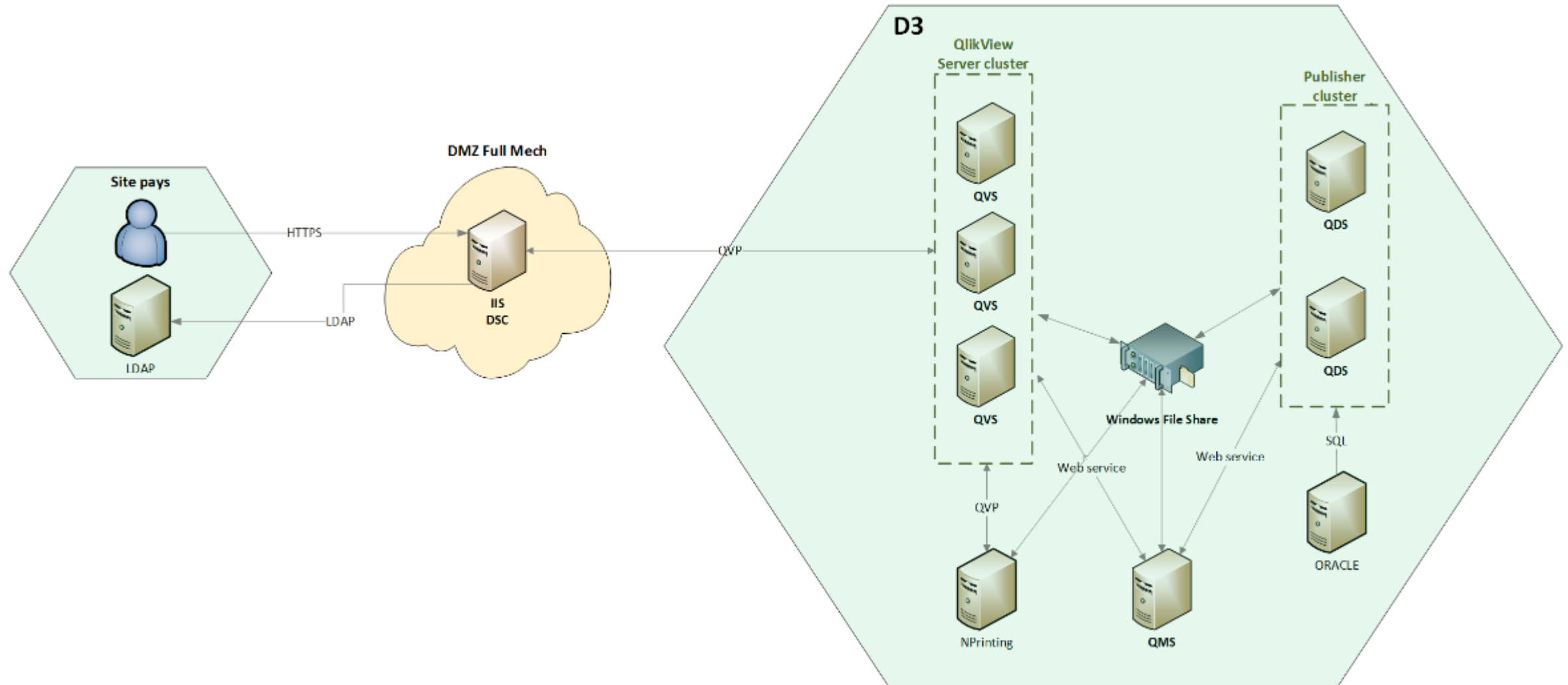


Architecture Serveur

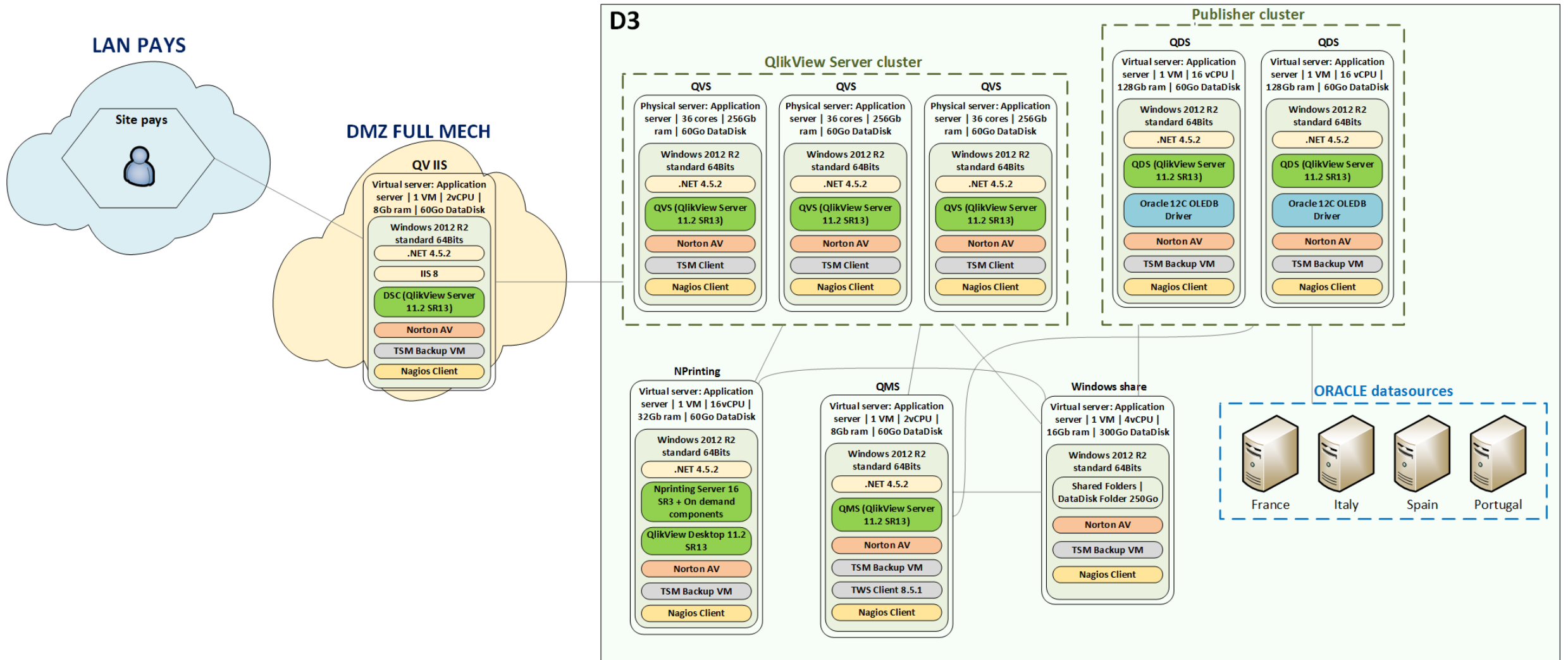
QlikView, Serveur et Publisher



Exemple Architecture Serveur



Exemple Architecture Serveur



Questions / Réponses

