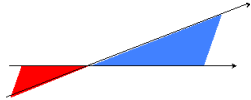


# NPL



## Projet

# Informatisation de l'atelier

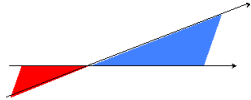


# Introduction

Ce dossier est une contribution à la réflexion d'ensemble du projet d'informatisation de l'atelier de l'usine **NPL** et de ses unités de production déportées.

Il vise à présenter une alternative actualisée, en phase avec les progrès importants enregistrés récemment par l'arrivée sur le marché de matériels de plus en plus puissants, fiables, miniaturisés et robustes.

Associés aux derniers OS de **microsoft**© ces matériels permettent une généralisation de la mise en réseau de l'entreprise incluant l'atelier de fabrication comme tout autre service de l'entreprise.



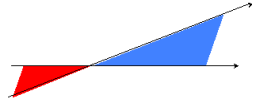
# Architecture serveur-clients

- Serveur

Essentiellement un serveur d'informations constitué d'une base de données et d'un point d'entrée d'accès aux informations via un serveur d'application.

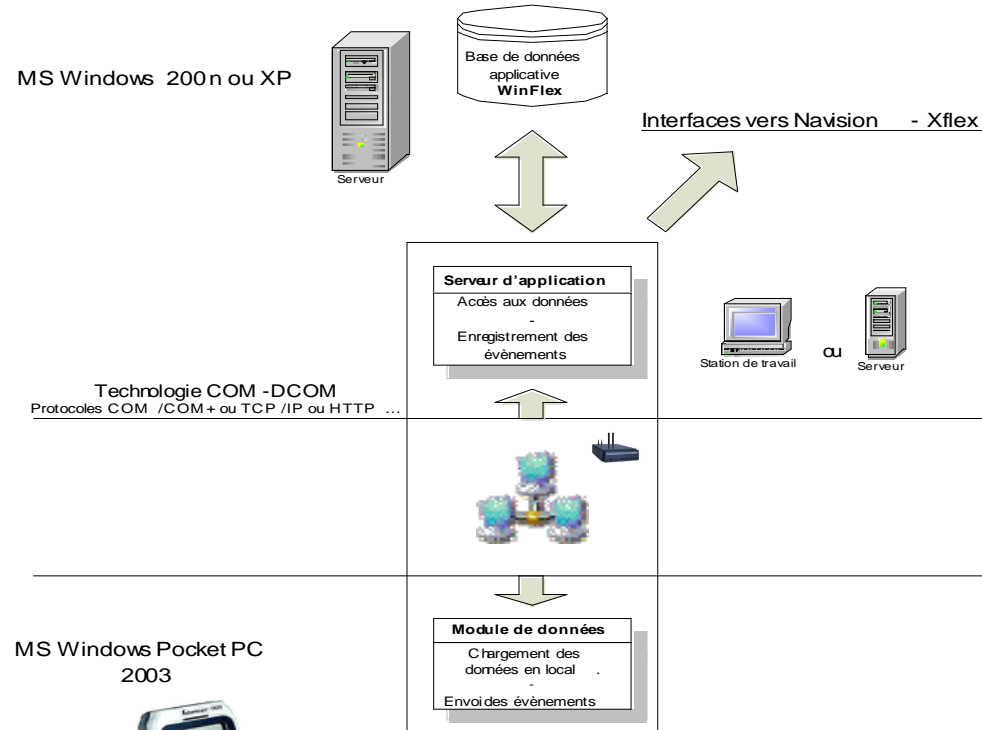
- Clients

Applications clientes constituées d'une interface utilisateur (programme applicatif sur terminal portable) accédant ou transmettant les informations par l'intermédiaire d'un module de données prenant en charge la connexion au serveur d'informations.

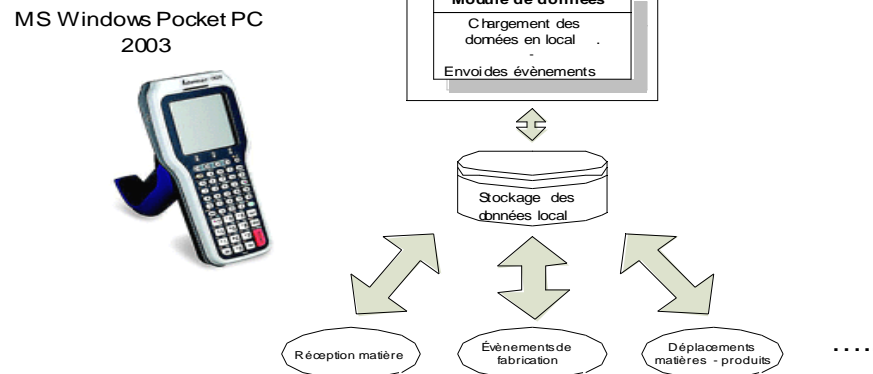


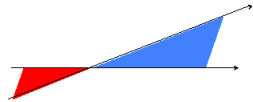
# Schéma architecture serveur-clients

serveur



client





# Architecture serveur - clients

## • **Serveur**

Un serveur est constitué d'un poste du réseau, station de travail ou serveur de données, dédié à la connexion avec la base de données.

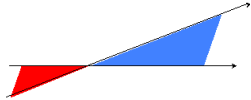
Condition de fonctionnement du serveur :

- Activation du service « serveur d'application ».**
- Le poste de travail est répertorié en paramètres d'exploitation des terminaux « clients ».**

**Dans l'absolu plusieurs stations de travail peuvent simultanément jouer ce rôle.**

## • **Client - Terminal portable**

Poste on line/off line de fonctionnement proche d'une station de travail sous MS Windows, doté de ses propres capacités de stockage permettant un fonctionnement équivalent à un PC classique.



# Serveur d'application

Localisé sur une station de travail ou sur le serveur hébergeant la base de données, le serveur d'application va gérer les échanges d'informations avec les « clients » du réseau.

C'est un programme fonctionnant en service de l'OS Windows constitué à la fois de composants techniques et applicatifs.

## **Les composants techniques :**

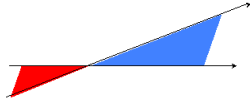
Ils assurent en collaboration avec leur homologue du « module de données » du poste client, la sécurité tant sur le plan protection que sur celui qualitatif dans le transport des informations sur le réseau.

C'est au travers de ces composants techniques, fournis sous forme de modules par les outils de développement logiciel et par l'OS MS Windows que l'on sélectionne le protocole de communication le mieux approprié et que l'on construit les scénarios d'échanges entre les postes du réseaux.

## **Les composants applicatifs :**

La partie privative application va contenir le code spécifique de traitement des informations à envoyer ou à recevoir.

C'est le module regroupant ces composants qui est approprié pour déclencher les contrôles, la tenue du journal des échanges, les alertes sur anomalies, la génération des interfaces vers les applications tiers, etc ....



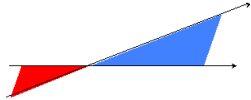
# Serveur d'application (suite)

## Principe de fonctionnement

- La connexion à ce service sera toujours à l'initiative du terminal « client ».
- Il n'y a pas de notion de file d'attente, l'ensemble des sollicitations simultanées de plusieurs terminaux « client » se règle par la création pour chacune d'une « instance » d'exécution qui se détruit après aboutissement. Cette mécanique est pleinement mise en œuvre par les composants techniques issus de l'OS et par les outils de développement logiciel.

-

L'effort réel de l'étude se concentre essentiellement sur la partie privative application, dédiée à la validation des transactions reçues et les traitements qui en découlent.



# Module de données « client »

**Le module de données « client » est la partie service d'accès aux données communes à l'ensemble des postes du réseau.**

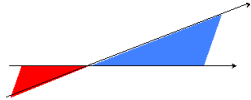
**Comme pour le service « serveur d'application » ce programme fonctionne en service de l'OS Windows et est constitué à la fois de composants techniques et applicatifs.**

## **Les composants techniques :**

Ils assurent en collaboration avec leurs corollaires « serveur d'application » du poste ou des postes « serveur », la sécurité tant sur le plan protection que sur celui qualitatif dans le transport des informations sur le réseau.

## **Les composants applicatifs :**

Partie privative application qui va contenir le code spécifique de traitement des informations à recevoir ou à envoyer.



## Module de données « client » (suite)

Le module de donnée est activé de deux façons :

ÿ **En automatique à espace de temps paramétré.**

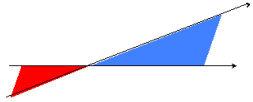
Selon un temps mémorisé dans les paramètres généraux de l'application, ce module effectuera une requête d'interrogation à l'un des serveurs d'application :

*« Y a-t-il un groupe d'informations à importer ? »*

Dans l'affirmative le module réceptionne le ou les fichiers à disposition pour, après le transfert terminé avec succès, procéder à la mise à jour de la base d'informations utilisée par les programmes de collecte.

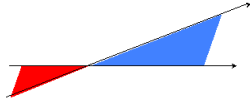
ÿ **Provoqué sur sollicitation d'un des programmes de collecte en atelier.**

Lorsqu'une procédure de saisie d'informations est complète, le programme génère un enregistrement de données qui est mémorisé dans un collecteur d'envoi dans la base de données locale du terminal pour ensuite être dirigé vers le module de données.

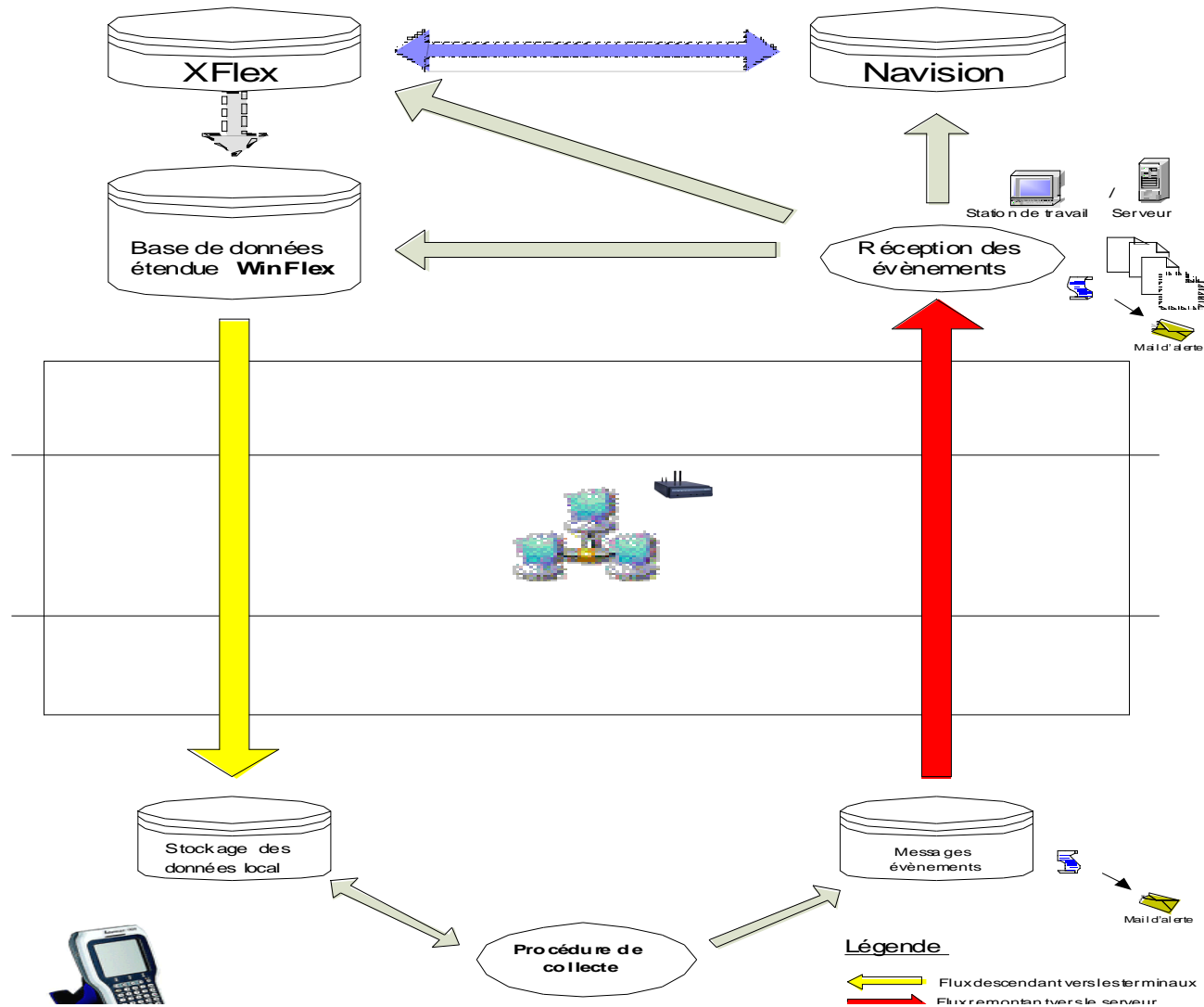


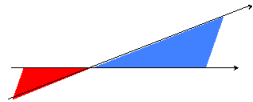
## Module de données « client » (suite)

Comme pour le module serveur d'application, l'effort réel de l'étude se concentre essentiellement sur la partie privative application, dédiée aux procédures de scanning des codes à barres, à l'enchaînement des sélections, des options de travail, des contrôles des codes lus dans chaque contexte, etc ...



# Flux des informations

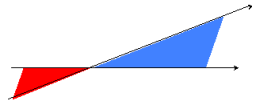




# Flux des informations (suite)

## Base d'informations source

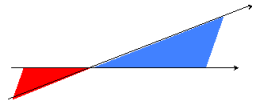
- La base principale, source des informations en référence des procédures de collecte en atelier, est celle du logiciel de GPAO **XFlex**.
- Le référentiel technique est à la base des contrôles de validité des codes saisis.
- Les ordres de fabrication et de sous-traitance sont également communiqués aux terminaux de collecte de façon à bénéficier d'une saisie « intelligente » à l'instant de l'évènement.
- Les livraisons attendues sont également transmises ce qui permettra tous les contrôles liés aux approvisionnements :  
commande – livraison – facturation.



# Flux des informations (suite)

## Transmission aux terminaux d'atelier

- La base d'information source **XFlex** est transmise régulièrement au système **WinFlex** (& terminaux) , pour prévenir tout déphasage entre les deux systèmes issu d'incidents techniques exceptionnels.
- A chaque événement traité, provenant d'une collecte par terminal atelier ou saisie directement sur un des postes du réseau, **WinFlex** génèrera un fichier de mise à jour contenant les informations réactualisées et déposées dans un répertoire spécifique d'envoi, consulté régulièrement par les terminaux d'atelier.
- Le transfert est pris en charge par les modules techniques du serveur d'application et du module de données « client » de chaque terminal atelier.

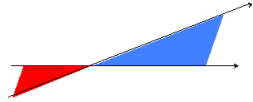


# Flux des informations (suite)

## Gestion des importations sur terminal atelier

Les informations reçues sont stockées sous forme de fichiers de données en mémoire vive et sur un support mémoire non volatile inséré en carte mémoire supplémentaire extractible.

La détection d'une nouvelle version d'information est effectuée automatiquement par les procédures de collecte qui procéderont à leur mise à jour en mémoire vive.



# Flux des informations (suite)

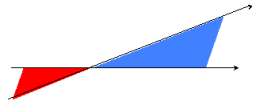
## Génération et transmission des messages « événements »

Une procédure de collecte menée jusqu'à son aboutissement débouche sur la mise en forme d'un message stocké en fichier sur la mémoire additionnelle de chaque appareil (terminal) et simultanément transmis au module de données qui prend en charge l'opération d'envoi à l'un des serveurs d'application présents sur le réseau.

Une première série de contrôles concomitants peut être effectuée sur le terminal. Le premier contrôle est celui lié aux dépassements d'un seuil de tentatives de connexions non abouties, d'autres concernant éventuellement les données par elles-mêmes.

Le déclenchement d'un mail à destination d'un responsable concerné par le type de problème sera envoyé automatiquement.

En cas de problème matériel sur le terminal lui-même, il sera toujours possible de récupérer l'ensemble des informations non transmises par extraction de la carte mémoire de stockage non volatile.



# Flux des informations (suite)

## Réception des évènements

**Le serveur d'application adressé par le terminal portable reçoit les informations transmises et envoie les données au module d'enregistrement des messages.**

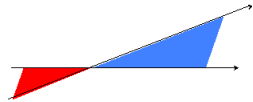
**Chaque transmission donne lieu à l'inscription dans un journal de bord notifiant l'émetteur (terminal), le type du message, les dates et heures etc..**

**Une deuxième série de contrôle est effectuée, à la fois sur la qualité des données transmises, ainsi que sur la cohérence des données collectées par le terminal.**

**Les contrôles de cohérence sont liés à chaque type de procédure de collecte, ce sont :**

- **Pour la fabrication :**
  - **Des quantités produites conformes aux composants alignés**
  - **Des composants correspondant au produit fabriqué (ordre de fabrication).**
  - **Etc ...**
- **Pour la réception des approvisionnements**
  - **Des matières conformes à celles attendues**
  - **Des quantités valides**
  - **Etc ...**
- **Etc ...**

**Le déclenchement d'un mail à destination d'un responsable concerné par le type de problème sera envoyé automatiquement.**

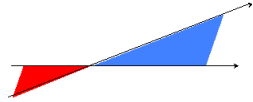


# Flux des informations (suite)

## Réception des événements (suite)

Parallèlement à ces opérations de contrôle et de validation le module d'enregistrement des données procède à la mise à jour des bases de données applicatives :

- **WinFlex**, en effectuant la mise à jour directe dans la base de données.
- **XFlex**, par la fabrication d'un fichier « événement » normalisé et digéré automatiquement par la « sentinelle » d'échange déjà utilisée dans ses procédures avec **Navision**.
- **Navision**, par la fabrication d'un fichier « événement » déposé dans son répertoire d'importation.

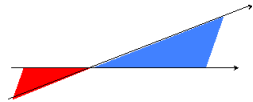


## Flux des informations – synthèse -

L'exposé de la mécanique présenté dans les vues précédentes décrit un cycle fermé de circulation d'informations qui garantit l'intégrité des données.

Chaque envoi d'un message par un terminal aboutit en fin de cycle à l'arrosage de l'ensemble des terminaux après toutes les mises à jour faites dans les bases de données applicatives (XFlex, WinFlex, Navision).

Chaque terminal est rechargé des données impactées par l'action d'un autre, de sorte qu'aucun appareil n'est véritablement dédié à un type de collecte ou à un secteur d'action, ce qui offre une grande souplesse d'utilisation.

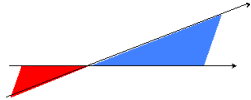


# Terminal - communication -

Le terminal d'atelier peut adresser n'importe quel poste du réseau de l'entreprise.

Il peut notamment envoyer directement une impression vers n'importe quelle imprimante, envoyer un mail, un fichier etc ...

Certaines impressions de type codes à barres peuvent être spécialement dirigées vers un appareil désigné par simple lecture de son code à barres d'identification.



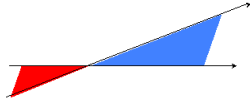
# Scanning des codes à barres

L'ensemble des codes à barres à lire, à l'exception des codes externes à l'entreprise pourrait se structurer de la façon suivante :

- un caractère '@' de reconnaissance code barre interne du client.
- trois caractères de reconnaissance du type lu (exemple : 'MAC' pour machine, 'OUT' pour outil, 'MAT' pour matière etc ...)
- un caractère de séparation code type et valeur origine : ';'.
- La valeur du code d'origine (exemple : code machine Xflex).
- un caractère de séparation code valeur origine et son (ses) complément : ';' (le complément serait par exemple le numéro d'exemplaire d'un outil).

**Ainsi construit chaque code lu permet le déclenchement automatique d'une procédure liée à son type et tous les contrôles qui vont avec.**

**De plus, une telle construction évite la lecture dans un fichier le référençant, ce qui établi instantanément sa validité.**

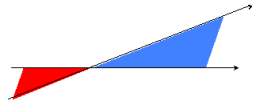


# Procédures de collecte

La technique de programmation est de type événementiel, c'est à dire dont les actions sont déclenchées à la suite d'évènements de lecture dont le déchiffrement des codes désigne implicitement les usages à en faire. Les traitements sont tantôt déclenchés immédiatement derrière la lecture ou soit suivi d'une liste d'option de travail à sélectionner.

Ce mode de conception de programme agit au final comme un véritable guide d'utilisation en excluant au maximum les interrogations de la part des utilisateurs.

C'est ce type de programmation qui est appliqué aujourd'hui dans le monde de la téléphonie mobile et qui permet à un public des plus divers d'utiliser avec la plus grande facilité toutes les possibilités offertes par les téléphones modernes.



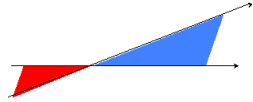
## Procédures de collecte (suite)

Un des aspects les plus importants dans la collecte d'informations, outre celui lié au contrôle de validité de l'information scanné qui est assuré par la structure du code lu, concerne la cohérence d'un groupe d'informations nécessaire à un contexte d'un événement précis.

Une quantité fabriquée est liée à un ordre de fabrication désignant un produit et un ensemble de composants. Dans ce type de déclaration (quantité fabriquée), il y a lieu d'effectuer des vérifications en cascades, comme s'assurer que l'ensemble des composants a bien été désigné, que la quantité produite est conforme aux consommations calculées à partir des stocks des composants déclarés etc ...

Ces vérifications ne sont possibles qu'à partir du moment où le terminal d'atelier a en mémoire le descriptif complet de l'ordre de fabrication ce qui justifie donc l'importation régulière des bases d'informations mises à jour, comme décrit dans les vues précédentes.

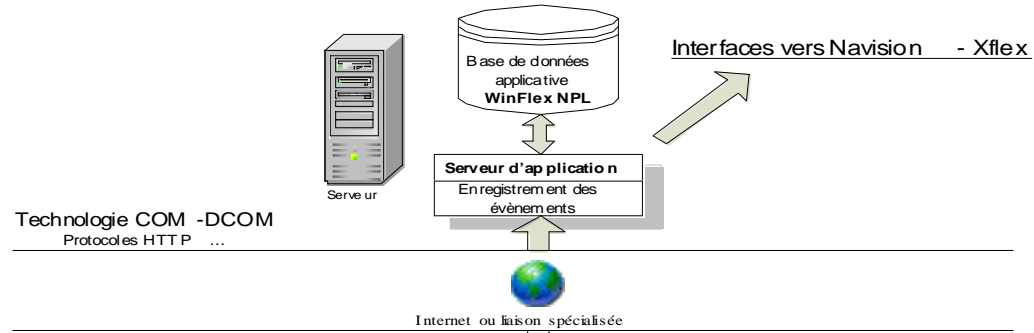
Ceci s'applique pour les autres programmes de collecte comme la saisie des réceptions matières en ayant en ligne les approvisionnements attendus.



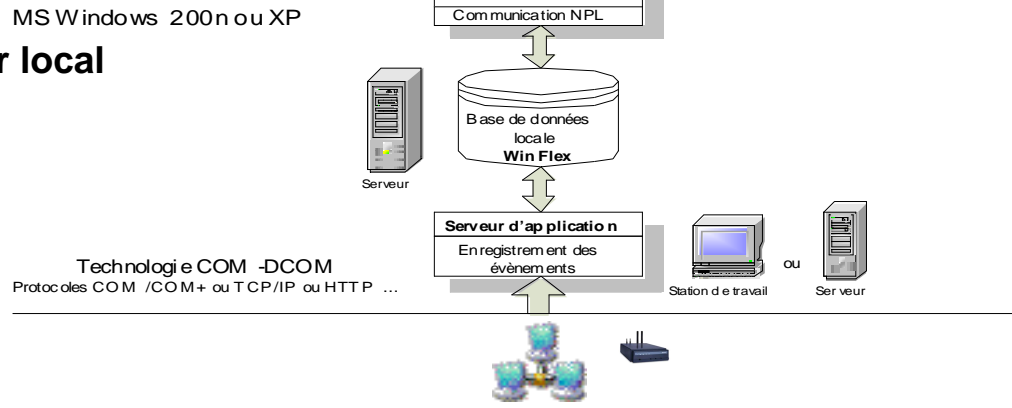
# Gestion des unités de production déportées

## Schéma

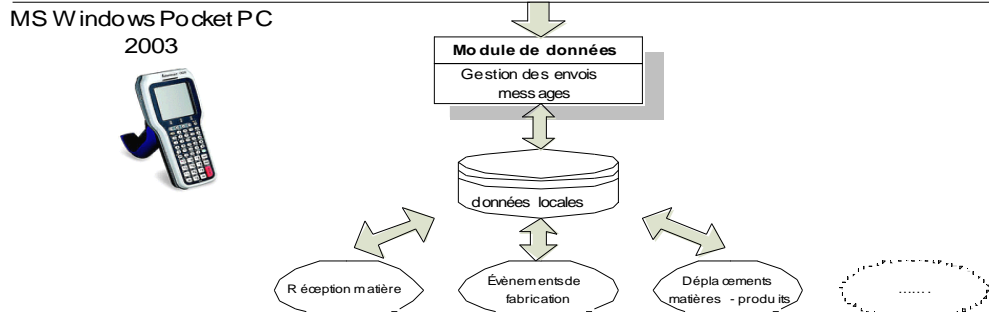
### Serveur NPL

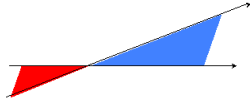


### Client NPL – Serveur local



### Client local



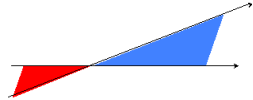


## Gestion des unités de production déportées

Le schéma de la vue précédente reprend en grande partie celui de l'architecture « serveur – clients » du début de ce dossier. Il comporte en fait quelques différences comme l'ajout d'un niveau de communication entre l'usine principale du groupe **NPL** qui gère l'ensemble de la production et les unités déportées, ou encore l'absence en local d'une génération de messages à destination des logiciels associés « **XFlex** » et « **Navision** ». Ces dernières procédures étant prises en charge lors de l'importation par l'usine principale **NPL**.

Le point le plus important dans ce schéma est qu'il révèle l'utilisation des mêmes technologies COM-DCOM en communication réseau local et distante. Ce sont les mêmes modules qui sont activés dans les deux cas, avec quelques variantes dans les procédures applicatives.

Le serveur local disposant du « module de données » caractérisant le client de la liaison entre l'usine NPL et l'usine déportée, se connecte au serveur d'application **NPL** spécifique à ce type de liaison (protocole http en liaison internet).



# Synthèse générale

Ce dossier expose une solution exploitant les toutes dernières technologies logiciels et matériels par l'arrivée des équipements portatifs « intelligents » et l'utilisation du réseau mondial Internet.

Les solutions toutes faites sur le marché sont pour la plupart héritées de concepts de systèmes centralisés type multi-postes où le recours à des terminaux inertes à capacité de stockage temporaire était la règle.

Il n'est plus nécessaire de disposer d'un ensemble complexe de logiciels de pilotage des postes en atelier comme cela est souvent proposé. Toutes les fonctions de contrôles et de fiabilisation des échanges sur le réseau sont dévolus aux systèmes d'exploitations (MS Windows XP, 200n, CE et Pocket PC 2003) qui dialoguent ensemble.

Ces difficultés techniques levées, toute l'attention peut être portée sur la conception des procédures à usage des utilisateurs car ce sont celles-ci qui au final assureront le meilleur rendement du projet dans sa globalité.