

Livre Blanc Job Scheduler

Planification & Ordonnancement des traitements informatiques
En modes SaaS OnDemand & dédié OnPremise



Ce Livre Blanc présente les objectifs à attendre et fonctionnalités à déduire d'une solution de planification & ordonnancement des traitements informatiques (Job Scheduler). Cette solution peut être utilisée soit en SaaS (mode OnDemand) soit de manière dédiée (mode OnPremise).

Document **Livre Blanc Job Scheduler dans le Cloud.pdf**
Version **2.3**

SOMMAIRE

1. Définition des planification & ordonnancement informatiques	3
2. Pourquoi automatiser la production informatique ?	4
3. Mode SaaS ou utilisation dédiée ?	5
4. Critères fonctionnels incontournables	7
5. Critères techniques indispensables	8
6. Spécifications requises	9
COMPATIBILITE TECHNIQUE	10
Interopérabilité et portabilité	10
Intégration avec les applications	10
Gestion des logs applicatifs	10
Performance	11
FONCTIONNALITES APPLICATIVES	12
Planification calendaire	12
Déclenchement évènementiel	12
Lancement d'un traitement	13
Interaction avec les autres traitements	14
Conditions de lancement	15
Gestion manuelle	16
Modélisation générique	17
Vision Métier	17
Visualisation graphique	18
Surveillance, détection et analyse des erreurs	18
Supervision globale	18
Evolutivité du périmètre fonctionnel	19
Evolutivité du périmètre technique	20
Production internationale	20
Gestion des versions	20
Maintenance	20
Fonction de simulation et analyse d'impact	21
Cycle de vie des traitements	21
Nommage des objets	21
Hiérarchisation de serveurs	21
Sécurité de la solution	22
Continuité de service	23
REPORTIG & ERGONOMIE	25
7. Concepts	26
8. Conclusion	28
9. Quelques références d'UNILINK	29
10. Contacts	30

1. Définition des planification & ordonnancement informatiques

Définition

Le Business de l'Entreprise repose largement sur l'exploitation d'applications de gestion métiers (ERP, finance, stock, logistique, etc.) pour organiser la production et la distribution de produits et de services.

L'objectif de la production informatique est de gérer le flux des travaux dans l'infrastructure informatique pour garantir la disponibilité des données auprès des utilisateurs.

Un Automate de Planification & Ordonnancement (Job Scheduler) permet, si possible sans aucune modification des scripts et applications existants :

- ▲ De fluidifier et automatiser l'exécution de traitements inter-applications inter-systèmes
- ▲ D'exploiter et suivre les traitements BATCH de tout type d'applications et ERP de manière centralisée et / ou délocalisée via une Interface Homme-Machine (IHM) ergonomique
- ▲ De maintenir la continuité des services en automatisant la reprise de l'exploitation en cas de souci technique
- ▲ D'assurer le pilotage graphique et le reporting automatique de l'exploitation de ces traitements

Bénéfices

- ▲ Obtenir des gains de compétitivité
- ▲ Anticiper une charge d'exploitation croissante à moyens constants
- ▲ Augmenter l'efficacité et la disponibilité des collaborateurs
- ▲ Tirer profit de la virtualisation des ressources
- ▲ Etre alerté en temps réel si besoin
- ▲ Disposer de moyens d'analyse et d'accès pour agir efficacement
- ▲ Contrôler la production à distance via un simple navigateur internet
- ▲ Avoir une vision claire des traitements en cours, de l'application globale au détail de la tâche
- ▲ Rationaliser les processus de gestion de l'Entreprise
- ▲ Accéder à un référentiel de procédures commun à tous les intervenants, indépendant des systèmes d'exploitation gérés
- ▲ Disposer de la souplesse nécessaire pour accompagner l'évolution de l'Entreprise
- ▲ Réduire les coûts et les risques d'exploitation en supprimant l'empilement d'utilitaires et de fonctions système utilisés pour piloter les traitements BATCH
- ▲ Garantir la disponibilité des données aux utilisateurs et des processus métiers à 99,x%
- ▲ Assurer la continuité de service en mode 24*7

2. Pourquoi automatiser la production informatique ?

La mise en place d'une solution d'ordonnancement doit permettre d'atteindre toute ou partie des objectifs suivants sur des systèmes d'informations hétérogènes :

- ▲.. Contrôler les traitements
- ▲.. Améliorer le suivi des applications
- ▲.. Optimiser et réduire les temps de traitement
- ▲.. Garantir une meilleure distribution des ressources
- ▲.. Minimiser les erreurs de lancement de jobs BATCH
- ▲.. Limiter les interventions humaines dans les chaînes de production
- ▲.. Fournir une documentation à jour pour tous les collaborateurs intéressés
- ▲.. Avoir une vision graphique centralisée sur l'état du lancement des traitements
- ▲.. Etablir des dépendances sans limitation entre les processus et les évènements
- ▲.. Lancer des traitements sur une architecture centralisée et / ou sur une architecture distribuée
- ▲.. Réduire la dépendance des données et résultats de traitements à la disponibilité des collaborateurs en charge de la production
- ▲.. Garantir une totale disponibilité de la chaîne de production en proposant des scénarios de reprise et un fonctionnement en mode dégradé
- ▲.. Disposer d'un référentiel et d'une solution commune à toute l'équipe de production pour assurer plus de souplesse et de rapidité lors des interventions

3. Mode SaaS ou utilisation dédiée ?

La mise en place d'une solution de planification & d'ordonnancement des traitements BATCH pour planifier, exécuter et suivre quotidiennement les traitements informatiques peut être exécutée via des services mutualisés disponibles en SaaS (On Demand) ou classiquement sur des infrastructures dédiées (On Premise). Les avantages des deux solutions sont liés à ces modes :

▲ SaaS :

- Mise en place initiale dans un délai très réduit
- Externalisation des ressources humaines liées au paramétrage
- Qualité d'hébergement très supérieure au niveau moyen constaté
- Facilité d'adaptation au besoin (hausse ou baisse) sans contrainte
- Flexibilité financière liée au transfert du CAPEX à l'OPEX

▲ Dédiée :

- Indépendance totale vis-à-vis d'une infrastructure mutualisée
- Intégration totale de la solution à son système de pilotage
- Capacité de modification des paramètres à tout moment

Dans le cadre d'un tel projet d'automatisation des planification, ordonnancement & lancement des tâches sur des infrastructures hétérogènes, le mode SaaS permet de réduire l'ensemble des ressources nécessaires aux opérations de mise en place, paramétrage, formation et pilotage. Les charges de personnel et d'investissement sont réduites dans des proportions très importantes et rendent donc possibles des projets:

- ▲ Nécessitant un démarrage dans un délai quasi-immédiat
- ▲ Pour des structures ne disposant pas de service dédié à la fonction de Production Informatique
- ▲ De durée limitée ne permettant pas un amortissement raisonnable d'une solution acquise en propre
- ▲ Dont le volume d'infrastructures pilotées est extrêmement variable

Livre Blanc Job Scheduler

De manière plus générale, le mode SaaS autorise l'utilisateur à pouvoir bénéficier de niveaux de service (SLA) tel que pratiqués pour de très grandes structures disposant de moyens financiers en rapport avec les investissements nécessaires. Les fournisseurs de services mutualisés en SaaS utilisent en effet des infrastructures informatique et réseau performantes et hautement sécurisées, permettant de s'engager sur des niveaux de SLA de type 99,95%, voire plus ! Les principaux indicateurs de performance d'un tel environnement sont :

- ▲.. Les alimentations électriques redondantes
- ▲.. Les sauvegardes régulières et multi quotidiennes
- ▲.. L'utilisation d'espace disque physiquement sécurisé en RAID
- ▲.. Les grandes bandes passantes réseau, montante & descendante
- ▲.. La garantie de délais réduits en cas de souci technique tant pour l'intervention que le rétablissement (GTR)
- ▲.. La supervision de l'infrastructure permanente 24h/24 7j/7

La condition minimale pour mettre en place un tel service en SaaS est de pouvoir formaliser les conditions dans lesquelles les traitements doivent être gérés et les communiquer au prestataire. En retour, ce dernier prend en charge les opérations :

- ▲.. D'installation et configuration des modules nécessaires
- ▲.. Du paramétrage initial de l'ordonnancement de vos travaux
- ▲.. De leur planification & pilotage quotidiens
- ▲.. De la production des rapports d'exécution
- ▲.. En cas d'incident, de la communication des éléments en cause

4. Critères fonctionnels incontournables

Les caractéristiques fonctionnelles d'une solution d'automatisation de l'exploitation à l'état de l'art sont principalement:

- ▲.. Le traitement possible simultané de plusieurs journées de production via la définition de "dates de référence" (et non la date système)
- ▲.. L'utilisation de paramètres permettant la création simplifiée voire automatisée d'enchaînements pour les traitements génériques répétitifs
- ▲.. Le déclenchement de traitement sur disponibilité ou contenu de fichiers
- ▲.. Le déclenchement de traitement sur demande utilisateur depuis internet
- ▲.. Le déclenchement de traitement sur réception de courrier électronique ou pièces jointes associées
- ▲.. La mise à jour du référentiel décentralisée et asynchrone par des interlocuteurs intervenant chacun sur leur domaine en mode projet
- ▲.. La possibilité de renommer les objets sans remettre en cause les séquences d'ordonnancement
- ▲.. La disponibilité d'un module de références croisées des objets de l'ordonnanceur
- ▲.. La gestion multi-environnements (développement, qualification, exploitation, etc.)
- ▲.. La gestion sécurisée des objets, profils d'utilisateurs et exploitation (reprise, contrôle, etc.)
- ▲.. La mise en production rapide et sécurisée grâce à la définition de modèles de production uniques et portables sur différents environnements avec les paramètres et contraintes d'exploitation dédiés
- ▲.. La fonction de simulation permettant de tester et visualiser des planifications et ordonnancements
- ▲.. La fonction infocentre permettant de disposer de statistiques pertinentes sur la production informatique
- ▲.. La fonction gestion des astreintes permettant d'intervenir en cas d'anomalie de production

5. Critères techniques indispensables

Les caractéristiques techniques d'une solution d'automatisation de l'exploitation à l'état de l'art sont principalement:

- ▲.. La non adhérence système, ERP et scripts clients n'imposant pas la modification des scripts pour gérer les codes retour
- ▲.. La non adhérence système, ERP et scripts clients n'imposant pas la modification des scripts pour gérer la planification des traitements
- ▲.. La tolérance à l'arrêt / redémarrage du serveur garantissant une reprise automatique et synchronisée des traitements quelle que soit la cause et la durée d'un arrêt
- ▲.. L'utilisation d'un système de gestion de base de données relationnelle (SGBDR) standard autorisant:
 - . La tolérance à un arrêt brutal du serveur en assurant une reprise immédiate, automatique et synchronisée des traitements
 - . L'intégrité des données centralisées grâce à la fonction de sauvegarde on-line ne nécessitant pas d'arrêt de la production
 - . La gestion via des requêtes SQL des suivi de traitements et consultation de statistiques personnalisées
 - . La compatibilité avec les produits de requêtes de SGBDR du marché
- ▲.. L'intégration de toutes les fonctionnalités sans module externe
- ▲.. L'utilisation native de la messagerie de l'Entreprise pour l'envoi de messages et / ou d'alertes sans nécessité de script d'envoi ou d'utilitaire
- ▲.. Une interface IHM disponible tant sous Windows (client lourd) que sous internet (client léger)
- ▲.. Une interface IHM unique et iso fonctionnelle permettant de naviguer en limitant au minimum les ouvertures de fenêtres et / ou l'appel à des menus différents
- ▲.. L'associativité d'un objet à n environnements en le considérant comme unique
- ▲.. La rétro documentation automatisée générant des dossiers de production adaptables à la norme de chaque Entreprise

6. Spécifications requises

Les besoins le plus souvent mis en avant lors de recherche de solution d'automatisation de l'exploitation peuvent être regroupés sous les thèmes suivants :

- .. Compatibilité technique
- .. Fonctionnalités applicatives
- .. Reporting & Ergonomie

COMPATIBILITE TECHNIQUE

Interopérabilité et portabilité

Les systèmes d'Informations mettent en œuvre de nombreuses applications sur différents systèmes d'exploitation et des organisations, centralisées ou décentralisées.

La solution doit donc être compatible avec les différents systèmes d'exploitation du marché et leurs variantes (Windows, Linux, Unix, OS/400, z/OS, GCOS, ...).

En cas de migration du SI vers d'autres plateformes systèmes, la solution doit permettre d'installer l'agent système nécessaire et de transposer le paramétrage technique des tâches à exécuter.

Intégration avec les applications

Le système d'ordonnancement doit appeler et récupérer facilement les codes retour applicatifs de tous types d'applications: ERP (SAP, People Soft, HR Access, JD Edwards, ...), applicatifs verticaux, applications "maison" et solutions techniques (sauvegarde, BI, etc.).

Chaque programme applicatif exécuté génère un code retour récupéré dans le script. La solution doit récupérer ce code nativement via l'agent sur l'initiative du lancement du job. Par défaut, une tâche terminée normale signifie qu'un script a renvoyé le code-retour 0 et une tâche terminée anormale un code-retour différent de 0. Il doit exister la possibilité de gérer différemment les codes-retour, et de considérer que la fin peut être normale quel que soit le code-retour.

Par ailleurs, la solution doit permettre de contrôler la présence, l'absence, la taille, la date ou le contenu d'un fichier. L'objet fichier doit pouvoir être utilisé en tant qu'évènement et prendre les valeurs VRAI ou FAUX selon l'état attendu, valeur déterminée par l'agent sur lequel le fichier est testé.

Gestion des logs applicatifs

Les différents traitements au niveau de la chaîne de production génèrent des logs permettant de fournir des informations sur le déroulement d'exécution des processus. Ainsi, le système d'ordonnancement doit exploiter ces logs applicatifs pour signaler les erreurs au niveau de sa console centrale.

Tous les actes de gestion (connexion, modification, évènement, utilisateurs, etc.) et d'exécution de processus doivent être tracés soit dans la base soit dans les journaux de la solution. A ce titre, la solution doit être compatible avec les contraintes d'exploitation de type SOX qui exigent la traçabilité de toutes les actions logicielles.

L'administrateur doit pouvoir définir la gestion des logs (nombre et taille des fichiers), ainsi que les actions (conserver en centralisé, ne pas conserver, à la demande en centralisé, sur incident en centralisé, ...). La solution doit permettre d'assurer automatiquement la rotation de ces logs sur l'espace centralisé du serveur ainsi que sur les agents applicatifs pilotés. La solution ne doit nécessiter aucun script ni commande à exécuter de manière externe.

Livre Blanc Job Scheduler

Performance

La solution installée sur les serveurs de production ne doit pas impacter sensiblement leurs performances (consommation faible en ressources CPU et mémoire RAM notamment.).

La communication entre les différents éléments de la solution ne doit pas impacter non plus les performances du réseau local.

Les échanges entre le serveur et les agents doit s'effectuer par messages en s'appuyant sur le protocole TCP/IP.

La solution doit pouvoir piloter plusieurs milliers de traitements par heure et par localisation.

FONCTIONNALITES APPLICATIVES

Planification calendaire

La solution doit permettre de planifier l'exécution des tâches. Cette planification doit pouvoir être exécutée en fonction de règles calendaires (tous les jours, 3^{ème} lundi du mois, ...). C'est un fonctionnement de type "montée au plan".

La solution doit permettre de définir autant de calendriers spécifiques que souhaités. Les lois calendaires définies peuvent être:

- ▲.. Simples: tous les mardis de chaque semaine
- ▲.. Complexes par la superposition de lois calendaires simples: troisième lundi ouvré du mois
- ▲.. Un intervalle de temps en jours, semaines, mois ou années: les trois premiers jours de la dernière semaine de mars

La modification d'un calendrier doit pouvoir impacter automatiquement et immédiatement toutes les conditions utilisées dans les tâches, y compris celles en production.

Les lois calendaires ne doivent pas avoir de limite de construction.

Déclenchement évènementiel

Il s'agit d'assurer le plus de tâches en un minimum de temps avec le moins d'erreurs possible et un grand taux de parallélisme.

La solution doit permettre de déclencher un traitement sur évènement de différents types : heure, statut de fin d'un autre traitement, fin de traitement de reprise, présence ou contenu d'un fichier, arrivée de courriel, commande envoyée par un programme, demande utilisateur, positionnement d'évènement, valorisation de variable, ... Les tâches sont exécutées dès que les conditions de déclenchement prévues sont satisfaites.

Exemple d'utilisation de cet évènement: attente du fichier de nom TOP du 10 au 15 du mois. S'il arrive, la tâche "traitement TOP" doit partir immédiatement. Si le 15 le fichier TOP n'est pas là, on remonte une alerte.

Variante : on est en attente de n fichiers de nom TOP du 10 au 15, mais le 15 la tâche "traitement TOP" doit se déclencher quel que soit l'état des fichiers (les fichiers TOP non arrivés étant optionnels).

Livre Blanc Job Scheduler

Lancement d'un traitement

Lors des traitements BATCH, le critère de mise en route de traitement des tâches doit être basé sur des évènements. La solution doit savoir gérer des conditions:

- ▲.. De planning: une tâche peut être associée à une loi calendaire. Si la loi calendaire est fausse, la tâche ne sera pas planifiée
- ▲.. D'activation, étudiées après les conditions de planning. Si la condition d'activation est fausse, la tâche passe à l'état de transparence et il y a enchaînement des successeurs. Les conditions d'activation peuvent être:
 - .Une loi calendaire
 - .La valeur d'un évènement
 - .L'état d'une autre tâche
 - .La commande reçue depuis un ordinateur autorisé
- ▲.. D'exécution, dépendant avant tout des liens entre tâches et de la valeur des codes retour qui pourront conditionner l'exécution de tâches suivantes. Les conditions peuvent être:
 - .Une loi calendaire
 - .La valeur d'un évènement
 - .L'état d'une autre tâche
 - .L'existence ou non existence d'un fichier
 - .La réception d'un courrier électronique avec ou sans pièce(s) jointe(s)
 - .. L'heure, au plus tôt ou au plus tard, avec possibilité de choix en cas d'heure au plus tard atteinte:
 - .. De blocage de l'enchaînement
 - .. De poursuite de l'enchaînement
 - .. D'exécution de la tâche
- ▲.. De ressource permettant de conditionner l'exécution simultanée de plusieurs tâches:
 - .Une demande client avec valorisation de paramètre
 - .La disponibilité de charge CPU sur la machine où doit se faire le traitement avec la possibilité de LOAD BALANCING si non disponibilité de CPU
 - .La gestion de ressources logiques de type "sémaphore"

Le calcul d'heure limite de lancement (heure au plus tard) peut être fourni par une durée prévisionnelle, statistique ou réelle.

Livre Blanc Job Scheduler

Interaction avec les autres traitements

La solution doit permettre:

- ▲.. D'interrompre, repousser ou recommencer toute tâche en cours
- ▲.. De grouper plusieurs tâches pour les gérer globalement. Les résultats de l'exécution peuvent être considérés comme condition de démarrage d'autres tâches ou groupe de tâches appartenant à des plateformes et applications différentes
- ▲.. De donner la possibilité de définir des seuils au niveau des temps d'exécution des tâches. Au cas où ce seuil est dépassé, la solution doit offrir la possibilité de définir des actions à effectuer: bloquer le traitement, envoyer une alerte à la console, etc.

Une tâche a, en général, un déroulement naturel guidé par ses propriétés telles que définies pour elle d'une part et associées dans les groupes et plans et des services auxquels elle appartient d'autre part.

Il doit être possible d'intervenir sur son déroulement en raison d'anomalies rencontrées ou de demandes particulières.

Les actes de gestion possibles sont :

- ▲.. Modifier les propriétés d'une tâche (modifications effectives uniquement pour la journée de production désignée)
- ▲.. Définir une action de gestion par commande (application immédiate ou différée)
- ▲.. Consulter voire supprimer la liste des commandes appliquées à une tâche nonencore exécutée
- ▲.. Consulter la trace d'exploitation d'une tâche
- ▲.. Consulter les statistiques d'exploitation d'une tâche
- ▲.. Créer ou consulter un bloc-notes

Les actes de gestion ne doivent pas impacter le référentiel s'ils ne concernent qu'une journée. La portée d'une modification du paramétrage doit donc être considérée par défaut comme exceptionnelle et non reproductible.

Livre Blanc Job Scheduler

Conditions de lancement

L'exécution d'une tâche ou d'un groupe de tâches doit pouvoir être conditionnée par des conditions multiples:

- ▲.. Conditions calendaires
- ▲.. Conditions sur évènement (fichier, courriel, valeurs de variables)
- ▲.. Conditions sur des temps d'exécution
- ▲.. Conditions sur la disponibilité de ressource physique (CPU) et de ressource logique (gestion de flux)
- ▲.. Conditions évènementielles
- ▲.. Conditions horaires (heure au plus tôt ou heure au plus tard)

Le lancement d'une tâche peut être défini dans une plage horaire déterminée en fonction:

- ▲.. De sa durée théorique
- ▲.. Du mode apprentissage, correspondant à la durée statistique fournie par la solution après plusieurs exécutions

La solution doit permettre d'associer des conditions horaires (heure au plus tôt, heure au plus tard) avec quatre choix possibles en cas d'heure au plus tard atteinte:

- ▲.. Blocage de l'enchaînement
- ▲.. Poursuite de l'enchaînement
- ▲.. Exécution de la tâche
- ▲.. Terminaison forcée: normale ou anormale

D'autre part, on doit pouvoir interrompre un processus en cours en fonction d'une durée maximale autorisée.

En cas de dépassement, la solution doit permettre d'adresser un message alerte en mode multimédia, avec association de pièce jointe (exemple log d'exploitation et log applicatif).

La solution doit utiliser la messagerie du site. Cela signifie que l'envoi de message au travers de la messagerie de l'Entreprise en fonction de l'état ou de l'avancement des traitements n'implique pas la conception et l'exécution de script spécifique. Par cette technique, la solution doit permettre également de conditionner l'exécution d'un traitement sur réception d'un courrier électronique avec traitement des éventuelles pièces jointes.

La solution doit pouvoir utiliser tout type de messagerie avec les protocoles MAPI, SMTP, POP3, VIM 32 ou IMAP.

Livre Blanc Job Scheduler

Gestion manuelle

La solution doit permettre la saisie de commandes de gestion:

Planifier	Planification de la tâche
Déplanifier	Demande l'annulation de la planification de la tâche
Activer	Activation de la tâche
Exécuter	Exécution de la tâche qui doit être en production
Bloquer	Blocage de la tâche la maintenant dans l'état où elle se trouve quels que soient les évènements se produisant
A. Blocage	Annulation du blocage de la tâche remplaçant une tâche dans l'état où elle se trouvait au moment de son blocage
Effacer	Mise en transparence de la tâche supprimant la procédure, les positionnements d'évènements et les envois de message
A. Effacement	Annulation de l'effacement de la tâche remplaçant une tâche dans l'état où elle se trouvait au moment de son effacement
Terminer	Fin forcée de la tâche permettant que: <ul style="list-style-type: none">▲ La fin de la tâche soit considérée comme normale ou anormale▲ La procédure, si elle est en cours, soit effectivement interrompue sur l'agent cible▲ L'enchaînement de la tâche soit poursuivi ou interrompu
Valider	Validation de la tâche terminée anormalement

Un exploitant doit pouvoir:

- ▲.. Annuler une tâche positionnée dans le futur
- ▲.. Débloquer manuellement une tâche en attente d'une condition ou d'une ressource

Tout traitement déclenché sur évènement temporel n'ayant pas pu s'exécuter (serveur en panne, traitement en erreur, ...) doit pouvoir être relancé avec en paramètre facultatif la date et l'heure de l'exécution souhaitée.

On doit avoir la possibilité de modifier les conditions de relance et l'heure en particulier.

D'autre part, en cas d'incident, la solution doit offrir la possibilité de mettre en œuvre un schéma de reprise automatique avec relance des traitements.

Livre Blanc Job Scheduler

Modélisation générique

Pour une meilleure gestion et suivi des traitements, il faut que la solution offre la possibilité de modéliser une vue globale et unifiée de tous les traitements lancés en production via une console centrale, dans le but d'un diagnostic simple et efficace.

Le serveur central doit permettre la description et la gestion centralisée des travaux (définition des règles de planification, gestion des calendriers...). Il doit permettre de consolider les informations de suivi et transmettre les alertes.

La description des processus d'exploitation doit être indépendante de la configuration matérielle (notion de modèle d'exploitation logique).

Le suivi de l'exécution des jobs doit être possible par une interface graphique qui donne à l'administrateur le choix entre une vision macroscopique et une vision détaillée du déroulement de l'exploitation.

Un suivi sur le statut des ressources, les événements et les files d'attente doit permettre de contrôler efficacement les ressources: mémoire, fichiers, espace disque, processus, ...

L'exploitant doit pouvoir être informé de la réussite ou de l'échec d'un BATCH ainsi que de la nécessité éventuelle d'une intervention humaine.

La solution doit offrir différents niveaux de granularité pour suivre le déroulement de la production des tâches sous forme graphique ou de fenêtre, aussi bien en client lourd qu'en client léger par l'intermédiaire d'un navigateur du marché.

Vision Métier

Pour un pilotage adapté de production importante, il faut que la solution offre la possibilité de modéliser une vue hiérarchisée de tous les traitements gérés. Cette hiérarchie peut être complètement libre, par exemple par application (paye, RH, comptabilité, commercial, ...). De plus, il doit permettre un sous-découpage en groupes à plusieurs niveaux non limités pour une vision détaillée. Ainsi, le suivi des traitements de paye pourrait comporter les sous groupes préparation, calcul & échange organismes.

Le suivi métier doit permettre soit:

- ▲.. Un graphe dynamique du groupe choisi pour une vision de l'avancement en pourcentage (PAYE – 85%)
- ▲.. Un GANTT de l'application basé sur les durées théoriques entrées par le concepteur ou sur les durées statistiques collectées pour une estimation temporelle de prévision de fin (RH – 15h35)
- ▲.. La capacité de détailler les vues en nombre de tâches terminées normales, en erreur, en exécution, en attente ou restant à réaliser

Livre Blanc Job Scheduler

Visualisation graphique

La solution doit disposer pour la supervision des opérations d'un suivi centralisé en mode graphique ou fenêtre (avec rafraîchissement automatique). Ce suivi peut porter sur l'ensemble de la production (tous serveurs confondus) ou être limité à un sous-ensemble paramétrable (métier, application, agent, ...). Cette interface doit donner une vue précise des tâches exécutées, des tâches en cours et des tâches à venir. L'exploitant doit pouvoir disposer de filtres élaborés afin de visionner les tâches suivant l'état de la production.

En fonction d'un paramétrage, les traces de production doivent pouvoir être conservées le temps souhaité. La solution doit permettre de gérer différentes dates afin d'assurer notamment l'historisation de chaque journée séparément et la conservation de la totalité des actions effectuées durant une journée.

Lorsque toutes les tâches d'une journée sont terminées, la journée doit se terminer automatiquement et pouvoir être archivée. L'archivage de date doit permettre à l'utilisateur, s'il y est habilité, de restaurer la date et également d'exécuter à nouveau des traitements dans le même contexte d'origine.

Les choix d'affichage en cas de nouvelle exécution de journée doivent être les mêmes que pour le suivi temps réel.

Surveillance, détection et analyse des erreurs

A chaque moment l'exploitant doit disposer du déroulement de l'état de l'avancement des tâches et pouvoir agir et / ou réagir sur son état en y apportant les actions nécessaires.

Par ailleurs, l'exploitant doit pouvoir disposer de différents niveaux pour l'établissement d'un diagnostic tels que :

- ▲.. La couleur de la tâche
- ▲.. Le log applicatif
- ▲.. Le bloc-notes comprenant les consignes à prendre

La solution doit disposer d'un graphe réseau qui se rafraîchit automatiquement suivant un intervalle de temps paramétrable. Chaque tâche doit se voir affecter d'une couleur différente en fonction de son avancement (couleur paramétrable par état).

Supervision globale

La solution doit générer des messages dans le journal des évènements du système (Windows ou Linux) sur le fonctionnement de ses différents moteurs ainsi que sur la terminaison des traitements. Une solution de supervision doit pouvoir exploiter le journal de ces évènements.

A l'inverse, il doit être possible d'adresser des messages à la solution pour lui signaler par exemple l'absence de ressources suffisantes sur un serveur. Dans ce cas, la solution doit pouvoir suspendre son activité jusqu'à un retour à la normale.

Livre Blanc Job Scheduler

Evolutivité du périmètre fonctionnel

L'ajout d'une nouvelle tâche ne doit pas remettre en cause toute la planification. La prise en compte de cette nouvelle tâche doit pouvoir être immédiate.

Toute nouvelle tâche intégrée dans le dictionnaire central de référence ne doit pas impacter la planification en cours. Cette intégration doit être possible mais non systématique.

La solution doit permettre de lancer manuellement une tâche non planifiée, tâche qui doit pouvoir s'exécuter sans perturber la planification existante.

L'exécution d'une nouvelle tâche doit pouvoir faire l'objet d'un paramétrage complet : au plus tôt, au plus tard, suite à arrivée de fichier, liée au résultat d'autre traitement, etc. Bien entendu, seule la planification doit être manuelle.

Si la modification est réalisée à partir du module de suivi de production, elle doit être provisoire et valide uniquement pour la journée en cours.

Par ailleurs, tout traitement peut être bloqué, forcé ou mis en attente à n'importe quel moment d'une journée de production. La modification sera prise en compte instantanément: suspendre, reprendre, arrêter, modifier la chaîne, supprimer la chaîne, la soumettre à nouveau à la planification, etc., avec ou sans modification du paramétrage.

La solution doit permettre également de demander ou invalider un traitement sur une journée future.

Dans les cas de déplanification / planification, la solution doit reconstruire automatiquement les liens. Ces actions suffisent en cas d'action d'urgence et ainsi éviter d'agir sur les modules annexes (agents, services ou serveur).

Dans le cas d'un suivi immédiat, il doit être possible de bloquer tous les composants d'un plan ou d'un groupe, avec une action unique en une seule fois puis d'en forcer le déroulement pas à pas (c'est à dire sans exécuter les successeurs de l'objet cible) et de pouvoir intervenir en cas d'incident, afin d'éviter le déclenchement des procédures dégradées.

En cas d'erreur et d'enchaînement sur un schéma de reprise, la solution doit permettre de reprendre le fil normal de l'exploitation au niveau souhaité / prévu par l'administrateur.

Bien entendu, toutes les actions effectuées sur les objets manipulés doivent être régies par une règle de sécurité où l'administrateur ou le responsable d'application ont donné des droits sur ces objets (lecture, modification, création, suppression). Ceci permet de rendre étanche la manipulation des objets d'une application par rapport à une autre ou au contraire de mutualiser certains objets (loi calendaire par exemple).

La solution doit permettre à l'utilisateur final d'agir sur le système soit via une interface graphique soit via un mode commande BATCH.

Livre Blanc Job Scheduler

Evolutivité du périmètre technique

L'ajout d'un nouvel agent cible ne doit pas remettre en cause le paramétrage des tâches réalisées.

La simple création d'un nouvel agent (machine physique cible) associé à une localisation (environnement logique) doit être suffisante pour intégrer la nouvelle machine dans l'environnement traité.

Production internationale

Du fait de la présence possible de l'Entreprise dans différents pays, la solution doit proposer la possibilité d'élaborer des plans de production par pays.

Les lois calendaires doivent pouvoir être conçues selon les calendriers propres aux différents pays, intégrant leurs jours fériés, sans limite. De plus, la solution doit gérer automatiquement les décalages horaires.

Gestion des versions

La solution doit disposer d'une gestion de version des chaînes et des objets qui la compose.

Il doit être possible de figer la définition donnée aux objets et la transmettre à la production. L'ensemble des définitions transmises constitue une version du modèle de production, caractérisée par un numéro de version. La production peut alors créer des journées de production faisant référence à une version.

Les évolutions réalisées ensuite sur le modèle, dans sa version courante, ne doivent pas avoir d'impact sur la production. La notion de version doit permettre de ne passer en production que les objets associés à une version "validée". Ces versions peuvent également être sauvegardées pour en avoir une image.

Il doit être possible d'effectuer facilement un retour arrière sur une version précédente en cas de besoin. La production doit alors pouvoir reprendre sur cette ancienne version.

Maintenance

Un volume, parfois important, de traitements récurrents est effectué pour assurer les sauvegardes, transfert de fichiers, nettoyage de File System, maintenance de BDD, etc. La solution doit permettre de définir des modèles de tâches dans le référentiel de production et de soumettre ces modèles sur les agents cibles avec les paramètres d'exploitation associés.

Les avantages de cette solution sont:

- ▲.. De réduire de manière significative l'effort de gestion des tâches récurrentes
- ▲.. De planifier et exploiter de manière automatique et synchronisée ces traitements récurrents avec tout traitement applicatif
- ▲.. De documenter et référencer l'ensemble de ces tâches

Livre Blanc Job Scheduler

Fonction de simulation et analyse d'impact

La solution doit permettre de simuler le déclenchement d'une tâche afin de vérifier la cohérence de la planification d'une chaîne de traitement sur toute journée d'exploitation (passée, présente ou future) afin de vérifier la cohérence de la planification et l'impact sur le plan de production.

La simulation doit prévoir comment les tâches vont se dérouler. Les opérateurs peuvent ainsi anticiper des situations telles que des inter-blocages ou des incohérences sur le temps ou les ressources.

Lors de la création d'une tâche, on doit pouvoir paramétrer le type de fin en mode simulation. La solution doit proposer un affichage du plan de production avec utilisation des temps moyens d'exécution. Ces dispositions permettent d'analyser l'impact d'une nouvelle chaîne.

L'utilisateur doit pouvoir forcer la terminaison (OK, KO, quelconque) et tenir compte de la durée moyenne des n dernières exécutions ou d'une durée théorique fixée à sa guise.

La solution doit inclure une fonction paramétrable ordonnancée et exécutée selon la périodicité désirée permettant d'afficher le planning prévisionnel.

Le planning prévisionnel doit permettre d'obtenir une liste des tâches d'un plan, des tâches d'un groupe, des lois calendaires, des actions différées, des conditions et:

- ▲.. D'informer de 1 à 31 journées de production
- ▲.. De rendre en compte les durées théoriques, prévisionnelles ou réelles
- ▲.. D'obtenir un fichier de format PDF par date

Cycle de vie des traitements

La solution doit permettre de gérer différents environnements (développement, recette, production, ...).

L'organisation mise en œuvre doit être le reflet de l'organisation de l'exploitation des applications de l'Entreprise. Si l'organisation du Système d'Information distingue des environnements de développement, de formation et d'exploitation, il doit être possible de déployer plusieurs serveurs afin:

- ▲.. D'assurer aux utilisateurs des environnements indépendants
- ▲.. De garantir l'étanchéité entre production et pré-production permettant de répondre à des cadencements de montée en charge maîtrisée, suivant des rythmes différents en relation avec les projets, comme l'implémentation de nouvelles versions d'une application, d'un ERP et d'applications spécifiques
- ▲.. De faciliter la conception et la maintenance par le développement unique d'objets ensuite dupliqués sur les autres environnements, complétés par la définition des paramètres propres à chaque environnement

Livre Blanc Job Scheduler

Nommage des objets

La solution doit permettre d'appliquer automatiquement une norme à tout ou partie des objets manipulés. Une fonction de renommage doit être disponible en cas de modification de règles et normes de tâches, ou pour assurer une reprise efficace et cohérente de dictionnaire issu de produits obsolètes.

Cette fonction évite notamment de détruire puis de recréer un objet lorsqu'une erreur a été commise lors de sa création.

Le renommage ne doit pas altérer la synchronisation des tâches, il ne nécessite aucune qualification des objets dont le nom a été modifié.

Hiérarchisation de serveurs

L'utilisation de la solution avec plusieurs serveurs doit permettre de rendre autonome l'exécution des travaux sur plusieurs sites de production, tout en conservant un modèle unique pour la production de l'ensemble.

Un site est constitué par un ensemble d'agents dont la production doit être pilotée directement par un serveur local. Les versions de production utilisées sur ce serveur local sont des copies des versions développées sur le serveur central. La notion de serveur central s'applique uniquement au développement du modèle. Une fois la production démarrée, tous les serveurs, central et locaux, fonctionnent sur un pied d'égalité.

Chaque environnement d'exécution doit être rattaché à un serveur et à un seul. L'activation d'une journée de production et la planification des tâches doit s'effectuer de façon indépendante sur chaque serveur local. Les ressources et les événements doivent donc être gérés de façon indépendante sur chaque serveur local et peuvent avoir des valeurs différentes d'un serveur local à un autre.

Pour fonctionner, un serveur local doit obtenir une copie d'une version du modèle de production. Cette copie doit pouvoir être mise à sa disposition par une action effectuée sur le serveur central. Lorsque nécessaire, le serveur local destinataire se connecte au serveur central, récupère les données mises à sa disposition et les installe.

Lors du transfert d'une version du modèle de production, la totalité des informations associées à cette version doit pouvoir être transférée vers chaque serveur local destinataire. Les objets d'administration doivent pouvoir être également transférés. Si l'objet n'existe pas sur le serveur local destinataire, il doit pouvoir être créé automatiquement. S'il existe, la définition transférée doit remplacer celle existante. Un transfert ne doit occasionner aucune suppression d'objets d'administration.

Sécurité de la solution

La solution doit traiter de manière sécurisée les aspects:

- ▲.. Authentification: l'accès à l'application réclame une authentification forte. La solution doit assurer la traçabilité de toutes les opérations effectuées (nature de l'opération, utilisateur connecté, date et heure)
- ▲.. Confidentialité: l'utilisation de la solution ne devra pas nécessiter la connaissance du mot de passe d'administrateur de la machine (ROOT, ...)
- ▲.. Gestion des profils: le système de gestion des utilisateurs (notion de profil relative aux droits d'accès aux services et aux données) doit permettre de donner aux utilisateurs des accès différenciés au système d'information en fonction de leur profil

La solution doit inclure les fonctions de:

- ▲.. Gestion des droits de création, modification, suppression et lecture sur les données
- ▲.. Création et d'administration de différents profils d'accès
- ▲.. Affectation d'un ou plusieurs profils à un utilisateur
- ▲.. Mise à jour dynamique des profils
- ▲.. Traçabilité des opérations de gestion des droits d'accès

Les scripts doivent pouvoir être lancés avec des comptes et des mots de passe spécifiques. Si le script lancé demande un compte utilisateur différent, la solution doit permettre de transmettre automatiquement l'information définie en paramètre (dans la localisation ou la procédure de la tâche).

La zone de saisie du mot de passe doit être masquée et sa valeur codée.

La solution doit être compatible avec systèmes de type SSO. Les droits des utilisateurs doivent pouvoir être définis spécifiquement pour l'administration, la conception et la gestion de la production, ainsi qu'aux niveaux applications et environnements d'exploitation.

L'accès aux données doit pouvoir être contrôlé via l'accès en lecture / écriture à une application et aux objets la composant:

- ▲.. Le(s) responsable(s) de l'application
- ▲.. Le(s) utilisateur(s) (ou classe de) ayant accès aux objets de l'application
- ▲.. Les objets de production: tâches, groupes, variables, règles calendaires, etc.
- ▲.. Le lieu d'exécution

Les contrôles doivent pouvoir porter également sur l'autorisation de réaliser des actions telles que terminer, suspendre, reprendre, modifier, etc. Toute action de création, modification, suppression ou renommage d'objet doit pouvoir être journalisée et comprendre les informations sur la date et l'heure de la dernière mise à jour ainsi que l'utilisateur qui en est l'auteur.

L'accès à l'exploitation et au suivi des traitements doit être paramétrable en fonction du profil de l'utilisateur connecté.

Livre Blanc Job Scheduler

Continuité de service

La perte de l'éventuel point central du système (Single Point Of Failure - SPOF) ne doit pas paralyser la production. Ainsi, le déroulement des chaînes locales devra donc être assuré.

S'il y a déconnexion avec le serveur central, le résultat des jobs exécutés par un agent doit être stocké localement. L'agent doit se synchroniser automatiquement avec le serveur central au rétablissement de la liaison avec celui-ci

L'administration de l'ordonnanceur peut être réalisée aussi bien en local qu'à distance en utilisant un système de base de données relationnelle du marché. L'utilisation d'un SGBDR présente deux avantages majeurs:

- ▲.. Garantie de l'intégrité des données centralisées grâce aux fonctions de sauvegarde on-line ne nécessitant pas l'arrêt de la production
- ▲.. Tolérance à un arrêt brutal en assurant une reprise immédiate et synchronisée des traitements

La base de données devant être sécurisée, il est conseillé de l'installer dans une baie SAN.

La solution doit assurer une permanence de fonctionnement: chaque machine doit être totalement autonome et pouvoir continuer le traitement local des jobs en cas d'indisponibilité du réseau.

Dans le cadre d'une architecture multiserveur, cette fonction est essentielle puisqu'elle permet de réaliser très simplement une sauvegarde d'un ensemble cohérent, et donc de restaurer un ensemble cohérent en cas d'incident, selon le cycle choisi par l'administrateur.

La solution doit pouvoir être mise en œuvre sur des architectures centralisées et / ou distribuées, distinguant éventuellement n environnements de développement et de production.

Le crash d'une machine cible ne doit impacter que les traitements en cours sur cette machine et ne doit pas arrêter la production.

Par ailleurs, l'indisponibilité d'un agent doit être détectée par le serveur. Il peut alors soit suspendre l'exécution des traitements qui seront repris une fois l'agent disponible (avec synchronisation automatique à partir de l'arrêt), soit basculer les traitements sur un autre agent associé au premier dans le cadre d'un pool d'agents. Cette fonction de LOAD BALANCING doit pouvoir également être mise en œuvre en cas de ressources insuffisantes sur un agent de production.

REPORTIG & ERGONOMIE

Pour la supervision des opérations, les utilisateurs doivent pouvoir disposer d'une fonction de suivi central en mode graphique. Différents niveaux d'information (vue passée, journée en cours, analyse des temps moyens d'exécution des chaînes et des processus, volumétrie traitée...) doivent permettre l'établissement d'un diagnostic afin de lancer les actions correctives appropriées.

Un journal consultable doit permettre de conserver une trace de toutes les étapes de l'exécution des chaînes ainsi qu'un historique détaillé.

La solution doit permettre l'élaboration et la production de tableaux de bord statistiques de différentes formes sur différentes périodes.

Ce système doit comprendre des rapports prédéfinis exploitables depuis toute solution de BI du marché ou depuis un outil fourni avec la solution.

La solution doit permettre de définir de manière simple et rapide ses propres rapports et d'intégrer tout type de données.

Le système doit être d'une utilisation particulièrement simple pour les utilisateurs et les administrateurs.

La solution doit facilement être paramétrable par un administrateur ne disposant pas de qualifications techniques particulières et ne demandant pas trop de navigation et d'ouverture de fenêtres pour aboutir à l'information souhaitée.

La conception des écrans doit permettre aux exploitants de se repérer facilement dans les différentes étapes du processus, compte tenu de la multiplicité des informations traitées.

Les actions doivent pouvoir être saisies à partir d'un point central.

La solution doit disposer d'une console permettant son administration et d'une ou plusieurs consoles permettant le suivi d'exploitation. Ces différentes consoles pourront être personnalisées par "processus métier".

Cette interface doit être disponible sous deux formats: Windows et HTTP.

Le nombre de postes clients ne doit pas être limité. Il doit ainsi être possible de mettre en œuvre des consoles de suivi dédiées, par exemple par métier ou d'organiser un bureau de travail multi applications sur les postes utilisateurs.

7. Concepts

Localisation

Le concept de localisation permet la duplication automatique d'un ordonnancement à partir de la saisie du modèle de référence et la définition de paramètres spécifiques.

Pour deux localisations (ou environnements d'exploitation) le gain est de 50% (de règles en moins) sur un JOB SCHEDULER classique et de 70% sur un pilotage sans JOB SCHEDULER.

Cette définition en mode objet facilite et accroît les performances de la phase d'industrialisation et de maintenance des traitements applicatifs.

Journée de production

Certains produits ne fonctionnent pas avec la notion de journée de production et ne gèrent pas les variables de type "date" sur des journées passées ou futures : ce principe interdit tout parallélisme de journées de production et contraint à une gestion de variables manuelles dès lors que la date en cours est "passée".

La solution doit permettre de gérer de manière totalement étanche plusieurs environnements de production en parallèle, en pouvant faire glisser une journée de production, afin de laisser la possibilité de boucler une journée de traitement sans être contraint par le temps. Cette notion doit permettre de traiter n journées de production en parallèle, sans contrainte de temps de type "début" ou "fin" de journée.

Une journée de production n'est pas limitée à 24 h. Elle doit s'activer à une heure donnée (qui est paramétrable) et se terminer quand tous les traitements liés à cette journée se sont exécutés et terminés normalement (ou que toutes les terminaisons anormales aient bien été validées). Par exemple, la date de référence du 13 avril s'activera à minuit et se terminera peut être le 14 avril à 11 h du matin. Celle du 14 avril se sera activée toujours à minuit et les traitements planifiés pour cette journée seront exécutés en parallèle avec les jobs de la journée du 13 avril toujours active.

Cette fonction est particulièrement intéressante pour les services informatiques en charge d'exploitation répartie sur plusieurs fuseaux horaires en leur permettant de se "retrouver" au sein de la journée de production virtuelle.

Livre Blanc Job Scheduler

Serveurs et agents

▲.. Serveur de pré-production

Ce serveur est utilisé pour la conception de l'automatisation et les tests d'intégration de l'ensemble des sociétés.

L'organisation du référentiel doit être assurée avec les objets de type "APPLICATION" et "PROJET".

L'objet APPLICATION doit permettre d'associer des droits utilisateurs (lecture / écriture) aux objets de production.

L'objet PROJET doit permettre une mise à jour désynchronisée du dictionnaire de référence.

▲.. Serveur central de production

Ce serveur "reçoit" les mises à jour des projets du serveur de pré production et assure la création d'une nouvelle version du dictionnaire de référence (sans écraser les précédentes).

Cette version est ensuite diffusée sur les serveurs locaux de production.

La centralisation du processus de diffusion sur ce serveur central permet de centraliser le suivi la production de tous les éventuels serveurs locaux. Cela permet également de centraliser l'initialisation de journées de production et de ce fait, garantir la cohérence des versions de dictionnaire utilisées par les serveurs locaux s'il y a des synchronisations inter serveurs.

▲.. Serveur local de production

Ce serveur reçoit les mises à jour du serveur central de production. Il assure la production de toutes les tâches qui lui sont affectées. Ce serveur est autonome pour ses journées de production.

▲.. Agent de production multi système

L'agent est chargé d'exécuter les tâches de procédures demandées par un serveur. Il soumet et suit l'exécution des traitements de manière autonome par rapport au serveur. Il reste totalement synchronisé avec le serveur local et par son intermédiaire, avec d'autres agents. L'agent reçoit les ordres de soumission, exécute la demande et retourne au serveur le code de terminaison.

8. Conclusion

L'ensemble de ces critères doit concourir à faire d'une production informatique un ensemble cohérent, simple à définir, simple à maintenir et offrant aux utilisateurs une large gamme de fonctionnalités permettant :

- ▲ Une économie importante des temps et des coûts d'intégration
- ▲ Une meilleure qualité du référentiel de production par le respect de normes et de méthodes de production simplifiées et intuitives, induites par les fonctionnalités de la solution
- ▲ La simplification du suivi de production par des alertes et des tableaux de bord automatisés
- ▲ Une visibilité lors de l'intégration de nouvelles applications ou de nouvelles versions d'applications grâce à ses modules statistiques permettant d'obtenir des graphiques de la production et ainsi de s'assurer de la possibilité d'intégration de nouveaux travaux dans la fenêtre BATCH, garantissant la disponibilité des applications aux utilisateurs

Livre Blanc Job Scheduler

9. Quelques références d'UNILINK

UNILINK est éditeur de logiciels dans le secteur de l'automatisation de la Production Informatique depuis 1983. Son équipe est constituée de chefs de projets, d'architectes et d'experts produits, la majorité issue d'entreprises utilisatrices ayant une réelle connaissance des besoins et contraintes "terrain".

L'ambition d'UNILINK est d'aider ses Clients à réussir leur projet d'intégration des produits tant d'un point de vue mise en œuvre (obtention rapide de résultats concrets et optimisation des coûts), fonctionnelle et technique (meilleure utilisation possible en fonction de l'organisation Client) que d'un point de vue humain (maîtrise et acceptation des produits).



En savoir plus

10. Contacts

Informations générales

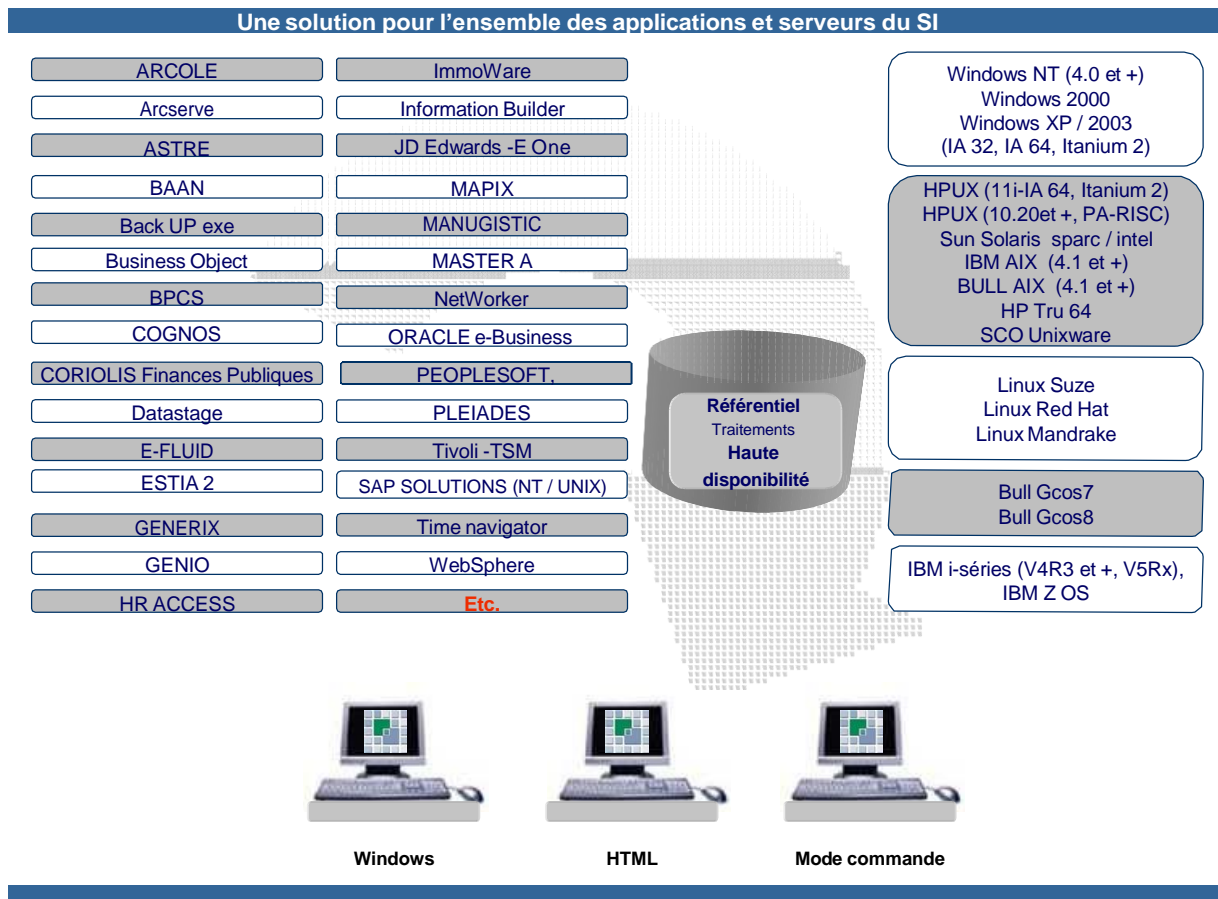
Société <http://www.unilink.fr>

Informations solutions

VegaSaaS <http://www.unilink.fr/vegasaas/>

VEGA <http://www.unilink.fr/descriptif-vega-2/>

Direction Commerciale et Services
 +33.(0)1 72 69 54 64
contact@unilink.fr



Livre Blanc Job Scheduler



UNILINK

169, Bureaux de la Colline – 92213 Saint-Cloud

Téléphone : 01.72.69.54.50

Télécopie : 01.72.69.54.51

Web : <http://www.unilink.fr>