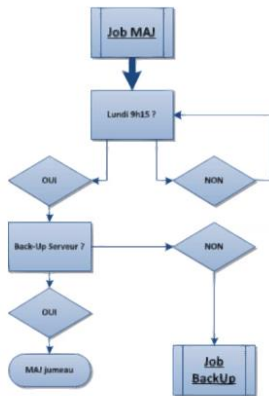


## Orchestrer les opérations de mise en production

Ordonnancement → pas trop de manip... car logiciels = principalement payant

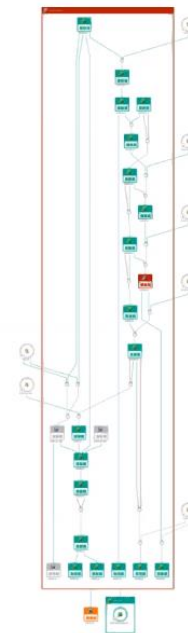


## L'ordonnancement

### L'ordonnancement : définition

L'ordonnancement des tâches informatiques consiste à :

- planifier et synchroniser des travaux batch
- quelle que soit leur nature
- quels que soient les systèmes d'exploitation



= Prioriser les tâches, avec des conditions afin de gérer et optimiser les décisions. i.e. une tâche ne sera lancée que si des conditions précises sont atteintes = si il y a une dépendance avec une tâche antérieure, elle ne pourra pas être lancée

L'ordonnancement, ne fait pas que la planification, mais exécute des tâches → cherche des scripts sur un serveur, lance des logiciels par exemple. Si on a besoin d'extraire et d'envoyer, on fera un job pour chaque action.

## L'ordonnancement : utilité

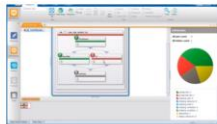
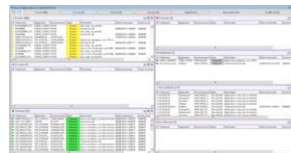
- Etablir des dépendances entre des batchs de différents langages, situés sur différents serveurs et donc scriptés dans des langages différents
- Visualiser et centraliser une chaîne de traitements
  - par ex : extraire des données sur une appli, les transférer sur un serveur via FTP, renommer, envoyer un mail au métier pour mise à dispo
- Optimiser voire réduire des temps de traitements
- Garantir une disponibilité de la production H24
  - minimiser les erreurs de lancement des jobs avec options de reprise et levées d'alertes
- Optimiser le temps de réaction humain pour garantir les SLA

Présentation Title | Author | Date

© Capgemini 2021. All rights reserved | 5

## L'ordonnancement : au quotidien

Suivre, analyser et corriger les plans de production au moyen de logs et de statistiques.



Créer de nouvelles chaînes de traitements pour faire évoluer le plan de production suivant les besoins de l'entreprise

Mettre en place des automatisations pour améliorer la qualité de service (chaînes techniques)



Présentation Title | Author | Date

© Capgemini 2021. All rights reserved | 6

## L'ordonnancement : exécution de tâches

### Crontab sous Unix

```
# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
00 21 * * * root    rm /home/bob/trash/*
```

### Tâches planifiées sous Windows

Nom de la tâche	Prochaine exécution	Déclencheurs	Emplacement
Firefox Default Browser Agent 3080...	11/07/2021 12:37:58	À 12:37 tous les jours	\\Mozila
OneDrive Standalone Update Task...	11/07/2021 13:44:34	À 10:00 le 01/05/1992 - Après le déclenchement, recommencer tous les 1.000000 indéfiniment.	\\
Bookmarks Backup	14/07/2021 12:00:00	À 12:00 tous les mercredi de chaque semaine, à partir du 17/09/2020	\\
BitLockerAD Backup	06/10/2021 12:19:30	S'exécute le Premier mercredi, de chaque janvier, avril, juillet, octobre à partir du 13/10/2020	\\

Présentation Title | Author | Date

© Capgemini 2021. All rights reserved | 7

Planification sous unix et windows qui existe déjà

## L'ordonnancement : Les différents outils



**Visual TOM**  
**Absyss**



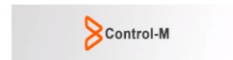
**Dollar Universe**  
**Broadcom**



**Autosys**  
**Broadcom**



**Control-M**  
**BMC**



Presentation Title | Author | Date

© Capgemini 2021. All rights reserved | 8

Evaluation via QCM

Ordonnanceur autosys

Paramétrage visuel via un explorateur. Pas d'IHM a proprement parler. On a des « box » qui contiennent les traitements qui doivent s'enchaîner → matérialisé via des ronds (qui contiennent des scripts).

Couleurs visuelles qui permettent d'avoir une vue globale : Rouge = pb ; Vert = ok ; Orange = en pause ; Bleu = en train d'être exécuté

Ordonnanceur Visual TOM

= réparti en plusieurs niveau d'objets → environnement cloisonné. Cloisonnement logique en fonction du statut de la tâche ? NB : flèche vert entre des tâches → dépendance facultative



## L'ordonnancement : DevOps & Cloud



Les ordonnanceurs s'adaptent et développent des modules complémentaires pour aller plus loin dans la gestion des tâches.

- SAP
- SGDB
- Kubernetes
- AWS/Azure
- Ansible



kubernetes



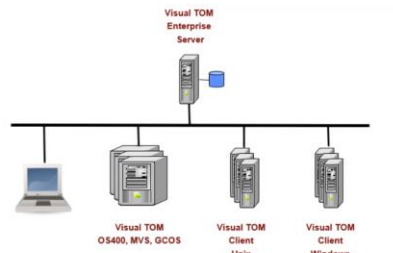
Presentation Title | Author | Date

© Capgemini 2021. All rights reserved | 9

## L'ordonnancement : Architecture centralisée

Une seule base de données regroupe l'ensemble des règles

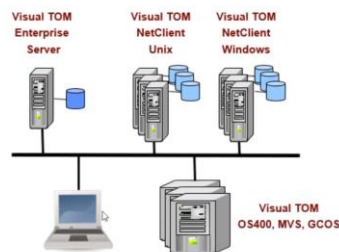
- Le serveur prend les décisions liées à la planification et l'ordonnancement
- L'agent prend en charge l'exécution du traitement sous la tutelle du serveur



- Facilité d'administration : un point focal à sécuriser
- Simplicité de pilotage : vision globale
- Les synchronisations inter-machines sont naturelles

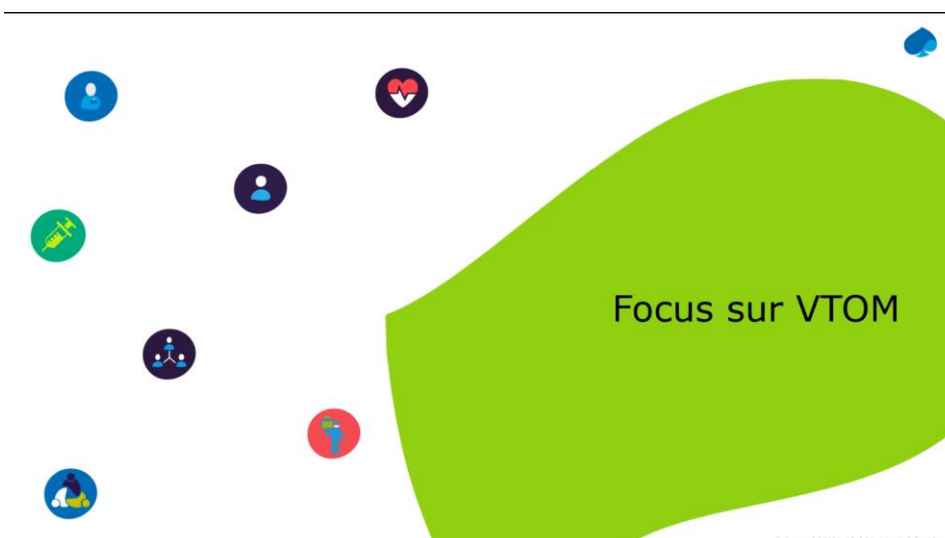
## L'ordonnancement : Architecture distribuée

- Chaque machine dispose des règles qui lui sont propres
- Les décisions liées à la planification et l'ordonnancement sont prises sur chaque agent



- Les règles de distribution des règles sont à la charge de l'exploitant
- Complexité d'administration : chaque agent doit être sécurisé
- Tolérance aux pannes : la disponibilité permanente du réseau n'est pas indispensable
- Pilotage partiel : vision locale
- Les synchronisations inter-machines rompent l'étanchéité

Avantage de cette configuration : si un serveur crash, les tâches s'effectueront quand même au niveau des agents car ils ont leur propre base de données et on les tâches a effectué en mémoire, mais on n'aura plus la visibilité côté humain de ce qui est effectué.



## VTOM : La hiérarchie des objets

Le **Domaine** regroupe tous les objets Visual TOM dans une base

L'Environnement est un ensemble logique d'Applications

Une **Application** est un ensemble cohérent de Traitements

Le **Traitement** est une tâche élémentaire exécutable



Presentation Title | Author | Date

© Capgemini 2021. All rights reserved | 13

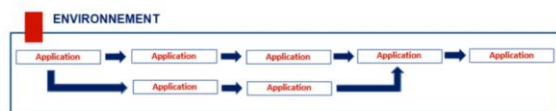
Le Domaine prend en compte tout ce qui existe sous VTOM (quels agents, quels traitements, quelles consignes, etc...) et tout ce qui sera stocker dans la base du serveur où VTOM est stocké.

Environnement => l'organisation est spécifique en fonction du client et de ses besoins, de son mode de fonctionnement. = un regroupement d'objets qui s'exécutent.

## VTOM : L'Environnement

L'Environnement représente un cloisonnement logique.

- Le découpage est choisi par le client en fonction de son organisation
- L'Environnement ne s'exécute pas en tant que tel mais contient des objets qui s'exécutent
- Chaque environnement est géré par un processus dédié : le moteur



© Capgemini 2021. All rights reserved | 14

## VTOM : L'Application



L'Application constitue un ensemble cohérent de traitements.

- On y définit les critères communs des traitements pour éviter de faire sur chacun d'entre eux
- On y définit la planification et les contraintes
- On peut les enchaîner avec des liens
- Deux types de lancement :
  - Cyclique
  - A la demande
- 4 types d'exécution : exécution, simulation, test, stop, traitement
- Comportement en cas d'erreur



Une application a trois types de lancement :

- Contrainte horaire
- Rythme cyclique [C]
- A la demande (i.e. pour relancer/lancer une application qui a été demandée) [D]
- [ ! ] = application stoppée, non planifiée

Comportement en cas d'erreur → on peut soit déplanifier, déplanifier les successeurs à..., ou ne pas déplanifier les successeurs

## VTOM : Le traitement



Le Traitement est le plus petit élément exécutable.

Il hérite des critères de son Application, et est relié par des liens aux autres traitements

On y définit

- des critères spécifiques
- des contraintes de démarrage (horaires, ressources)
- Des paramètres d'exécutions
- un paramétrage en cas d'erreur

Il est exécuté par l'Agent, qui retourne le statut au Serveur.

- 4 types d'exécution : exécution, simulation, test, stop, traitement
- Comportement en cas d'erreur

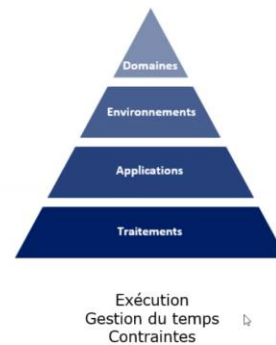


Contraintes : Horaire, lien, ressources

Il y a des consignes qui peuvent être liées au traitement. Par exemple, que faire s'il y a un code erreur et quoi faire en fonction du code erreur.

## VTOM : La hiérarchie des objets

- L' **Utilisateur** est l'identité sous laquelle est exécuté le Traitement
- L' **Unité de soumission** précise un mode d'exécution
- L' **Agent** précise un lieu d'exécution
- La **Queue Batch** définit le comportement d'exécution du Traitement
- Le **Calendrier** précise les jours d'exception
- La **Date précise** la journée de production
- La **Ressource** conditionne le démarrage des travaux
- La **Période** et le **Jeton** affinent la planification
- Le **Contexte** fournit des variables aux travaux
- L' **Alarme** déclenche une action en fonction d'un événement
- La **Consigne** indique au pilote les actions à effectuer



Unité de soumission → peut être un serveur ou un cluster de serveur = serveur qui fonctionne en binôme, qui se relayent en terme de charge → soit actif-actif, soit système primaire-backup en cas de pépin → soit actif-passif  
 Lieu de l'exécution = le serveur d'exécution (nom lié à l'adresse si déjà renseigné dans sa base de donnée VTOM) VIP = Virtual IP

## VTOM : La hiérarchie des objets

	Exécution	Gestion du temps	Contrainte
Utilisateur	✗		
Agent	✗		
Unité soumission	✗		
Queue Batch	✗		
Calendrier		✗	
Date		✗	
Ressource			✗
Période			✗
Jeton			✗
Contexte			✗
Alarme			
Consigne			

Aucun



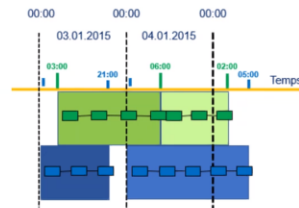


## VTOM : La date



La Date définit la journée de production.

- La journée de production est **indépendante de la journée calendaire**
- Tous les travaux du Domaine ne sont pas nécessairement sur la même journée de production
- 3 types de date : automatique, système, constante
  - **Système** : la journée de production s'étend de minuit à minuit, bascule automatique
  - **Constante** : la journée de production ne se termine « jamais », action manuelle pour modifier la valeur de la date
  - **Automatique** : la journée de production se termine quand tous les travaux sont statuéés, en cas d'erreur, possibilité de bloquer la date et de rattraper



NB : sur autosys → pas de notion de date. « les boxes ne sont pas conscientes du jour qu'il est, elle vont simplement attendre ce qu'il se passe au-dessus d'elles ou de l'heure qu'il est ou si on n'est pas un jour où elle n'est pas sensé fonctionner » contrairement à VTOM où la notion de date existe

Dans VTOM, les journées de production peuvent durer jusqu'à 48h

Date automatique, = on donne une date limite de bascule pour la journée de production (plafonnée à 48h)

On peut cocher une option de rattrapage → si l'heure minimale de bascule est passée, l'action s'exécutera comme si « elle venait de commencer sa journée » = utilisé pour des données critiques qui doivent être soumise/intégrée au jour le jour, mais qui ne gère pas le rattrapage.

Attention, une date ne peut pas basculer, si on n'a pas encore changé de jour sur le serveur. La date de journée de production se base initialement sur la date du serveur. Si on est 23h30 le 12/05, et qu'il y a une bascule à ce moment-là, la date restera le 12/05 pour le début de la prochaine journée de production.

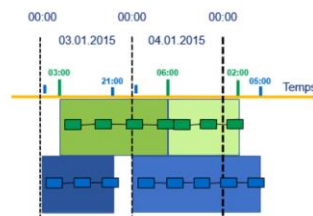
NB : sur le log, on a si on est à la 32<sup>e</sup> h du jour de travail, affiché la date de début de la journée de travail : 12/05 par exemple et l'horaire à 32hxxmin si on est le 12/05 à 8hxxmin du matin

## VTOM : La date



4 conditions pour que la date bascule :

- il faut qu'on soit au jour +1 (date système supérieure à la date qu'on veut basculer)
- tous les éléments doivent être statuéés
- aucun élément doit avoir l'option « bloquer l'avancement de la date »
- le moteur doit être démarré.



Ces 4 conditions concernent la date automatique.



## VTOM : La planification



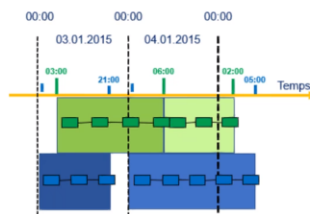
Différents objets définissent la planification sous VTOM.  
C'est le moteur qui effectue la planification.

Le Calendrier représente la liste des jours d'exception.

- Jours chômes
- Groupes de jours fériés
- Jours spécifiques de fermeture

Le Planning représente la règle de planification  
Il s'appuie sur le Calendrier pour traiter les exceptions

La Formule permet d'affiner la planification.  
C'est une règle basée sur un langage syntaxique. Elle s'appuie sur le Calendrier, et peut éventuellement s'appuyer sur le planning.



Pour la planification, il est possible de mettre en place une formule pour définir la planification au lieu de l'outil par défaut de VTOM. Ou d'utiliser l'outil par défaut et affiner avec une formule par exemple.

Par exemple :

- `test(today=premier.lundi.calendaire.mois)` pour avoir le premier lundi calendaire du mois
- `test(today=25.12)` → le jour d'exécution sera le 25 décembre

Attention, en mode hebdomadaire → le job se lance une fois par semaine par rapport au moment où il a été lancé la dernière fois. Si on décide pour une raison x ou y de relancer le job manuellement en cours de semaine, le job s'exécutera à nouveau une semaine à partir de la dernière exécution → dans ce cas, la périodicité est modifiée. Pour éviter cela, on évite de faire cela et on définit plutôt un jour d'exécution dans la semaine, toutes les semaines si on veut une périodicité hebdomadaire.

On peut entrer une durée max → l'un des intérêts : = pour kill un job s'il ne s'exécute pas et ne ralentisse pas le reste des jobs à exécuter après si ce dernier n'est pas crucial. Autre raison possible : quand un script est figé, mais comme l'action est distante et qu'il n'a pas de retour du logiciel à distance et est en attente du logiciel distant, il ne s'arrêtera pas car il n'a pas de retour = sorte de timeout.

Un job peut contenir des paramètres qui sont numérotés = les paramètres du script → sert de fichier de configuration. Permet d'avoir un script évolutif en fonction du temps ou de copier un job pour le réutiliser autre part en adaptant ses paramètres à son autre utilisation.

Fin de traitement : paramétrage du comportement en fin de traitement. Par exemple : si on veut une relance automatique => on peut définir le nombre de relance, un script de rattrapage

On peut également définir la gestion des logs en fin de traitement → durée, nombre d'occurrence

Alarmes en fin de traitement = alarmes spécifiques qui permet de gérer les alarmes et remontées d'alertes.

Pour faire une modification/action ponctuelle, sous VTOM, possibilité de mettre en place une modification temporaire. Attention, il ne faut pas oublier de modifier le planning pour la modification temporaire.

NB : chemin du script → si présence d'un #/chemin du script = veut dire que c'est une chemin unique/complet, on ignore le chemin donné dans la partie environnement

## VTOM : Les changements d'états

- 6 statuts qui donnent une indication sur le déroulement du traitement
- Une couleur représente un statut pour un état ou une application
- 4 statuts finaux :
  - Terminé
  - En erreur
  - Déplanifié
  - Non planifié



NB : on peut avoir une déplanification en cas de contrainte non validée.