



zenika
<animés par la passion>

Etude d'Architecture webMethods Alimentation Masse *RSCU*

Version 1.18-SNAPSHOT du 2018-11-01

Sommaire

1. Introduction	2
2. Exigences et contraintes	2
3. Diagramme de séquence macro	3
3.1. Hypothèse Spring Batch	3
4. Diagramme d'activités macro "Fichier"	5
5. Diagramme d'activités macro "Carrière"	6
6. Annexes	7
6.1. Marques de révision	7

Table 1. Historique

Date	Auteur	Détail
2018-08-24	bcouetil	Icones added for download + favicon added for webpage
2018-08-23	bcouetil	Initial commit

1. Introduction

Il s'agit du dossier d'étude d'architecture du RSCU de point de vue du pôle expertise webMethods de Capgemini.

2. Exigences et contraintes

- ☑ Performances (y compris pour une alimentation unitaire en cours de batch d'un autre régime)
- ☑ Scalabilité
- ☑ Intégrité des données
- ☑ Reprise sur erreur sans redémarrage à zéro
- ☑ Suivi temps réel de l'évolution des fichiers et des éléments de carrière
- ☑ Pas de pénalisation des autres éléments de carrière si le traitement de l'un d'eux tombe en erreur
- ☑ Lotissement intelligent (années de naissance) pour appel SNGI

3. Diagramme de séquence macro

```
box "Partenaires" #White
  participant Régime
  participant DGE
end box
box "RSCU" #LightGreen
  participant RSCU_Alim #Orange
  participant RSCU_BDD
end box
box "Systèmes participants" #White
  participant SNGI
  participant Divers
  participant RefEmp
end box

== par fichier ==
Régime -> DGE: Fic. alim. masse
activate Régime
deactivate Régime
activate DGE
DGE --> Régime: Acquittement
activate Régime
deactivate Régime
DGE -> RSCU_Alim: Fic. alim. masse
deactivate DGE
activate RSCU_Alim
RSCU_Alim -> RSCU_Alim: Division en carrières
RSCU_Alim --> DGE: Acquittement
activate DGE
deactivate DGE

== par carrière ==
RSCU_Alim -> RSCU_Alim: Contrôle norme
RSCU_Alim -> SNGI: Contrôle SNGI
activate SNGI
SNGI --> RSCU_Alim: OK
deactivate SNGI
RSCU_Alim -> Divers: Echanges informatifs
Divers --> RSCU_Alim: OK
RSCU_Alim -> RefEmp: Echange Référentiel Employeur
RefEmp --> RSCU_Alim: OK
RSCU_Alim -> RSCU_Alim: Conversion norme
RSCU_Alim -> RSCU_BDD: Ordre WS BDD
deactivate RSCU_Alim
```

3.1. Hypothèse Spring Batch

Dans l'hypothèse de traitement du fichier en Spring Batch dans le cas de la masse, Spring Batch effectue via des appels de service java les opérations sur chaque carrière. Ces opérations unitaires sont communes à l'unitaire et à la masse. De plus, ces opérations sont prévues en synchrones en mode unitaire.

La proposition commerciale était à l'origine prévue sur webMethods pour la masse et l'unitaire factorisés. Dans l'hypothèse de Spring Batch en masse, notre préconisation est d'abandonner webMethods en unitaire pour les raisons suivantes :

- webMethods se justifie sur des traitements multiples en parallèle
- webMethods se justifie sur des traitements asynchrones

- La masse et l'unitaire sont a homogénéiser en terme de techno pour des raisons de :
 - Cohésion de conception
 - Factorisation de code
 - Coût de réalisation Capgemini
 - Coût de maintenance CNAV

4. Diagramme d'activités macro "Fichier"

Ce processus regroupe les opérations faites au niveau fichier, et délègue au processus "Carrière" au fil de l'eau pendant le découpage.

```
start
:Recevoir un fichier;
if (Partenaire en phase transitoire ?) then (oui)
  :Contrôles bloquants;
  note left
    Ces opérations transitoires
    ne doivent pas complexifier
    le process nominal
  endnote
  if (taux critique) then (oui)
    stop;
  endif
endif
:Créer une entrée dans
la table Oracle FICHIERS;
note left
  Seules qqes données
  persistée : RefFichier,
  Date, Statut, DateStatut
endnote
:Parcourir le fichier
à la recherche de carrières;
repeat
  :Extraire la carrière
  courante;
  note
    Maintenir les actions
    carrière au minimum
    pour ne pas pénaliser le
    fichier sur un problème
    lié à une carrière
  endnote
  :Publier la carrière
  sur le bus;
  note left
    Ici démarrent
    les processus
    "Carrière"
  endnote
repeat while (Fin du fichier ?) is (non)
-> oui;
:Acquitter du Fichier;
stop
```

5. Diagramme d'activités macro "Carrière"

Le processus "Carrière" peut être à la fois le traitement d'un élément de masse et unitaire. Il est intégralement asynchrone. Dans cette proposition, les carrières sont stockées en base. Ceci est plus avantageux si :

- Les données d'une carrière ne sont pas toutes transmises aux systèmes participants
- Plusieurs systèmes participants sont sollicités en asynchrone (JMS, fichier, WS aller + WS retour)

Sinon privilégier le transport via le bus, hors proposition.

```
start
:Stocker la carrière
en base Oracle;
note
    Définir une politique de
    rétention courte
endnote
-[dotted]->
:Contrôler la norme;
-[dotted]->
:Contrôler SNGI par lot;
note left
    Système de file avec envoi JMS
    régulier par lots triés par année
    de naissance
endnote
-[dotted]->
:Informar les systèmes;
floating note left
    Privilégier l'échange synchrone
endnote
-[dotted]->
:Solliciter le
Référentiel Employeur;
floating note left
    Chaque étape prépare les
    données si elle peut éviter
    à la suivante de récupérer
    le contexte en base
endnote
-[dotted]->
:Conversion nouvelle norme;
-[dotted]->
:Ordre WS BDD;
stop
```


6. Annexes

6.1. Marques de révision

Différences depuis le dernier tag

