Etude d'Architecture webMethods
Alimentation Masse
RSCU

Version 1.18-SNAPSHOT du 2018-08-21

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
7. Annexes 8
    7.1. Marques de révision 8

Table 1. Historique

Date	Auteur	Détail
2018-08-21	NeVraX	HTML Asciidoc to Github

Table 2. Historique

Date	Author	Detail
2018-08-21	NeVraX	HTML Asciidoc to Github

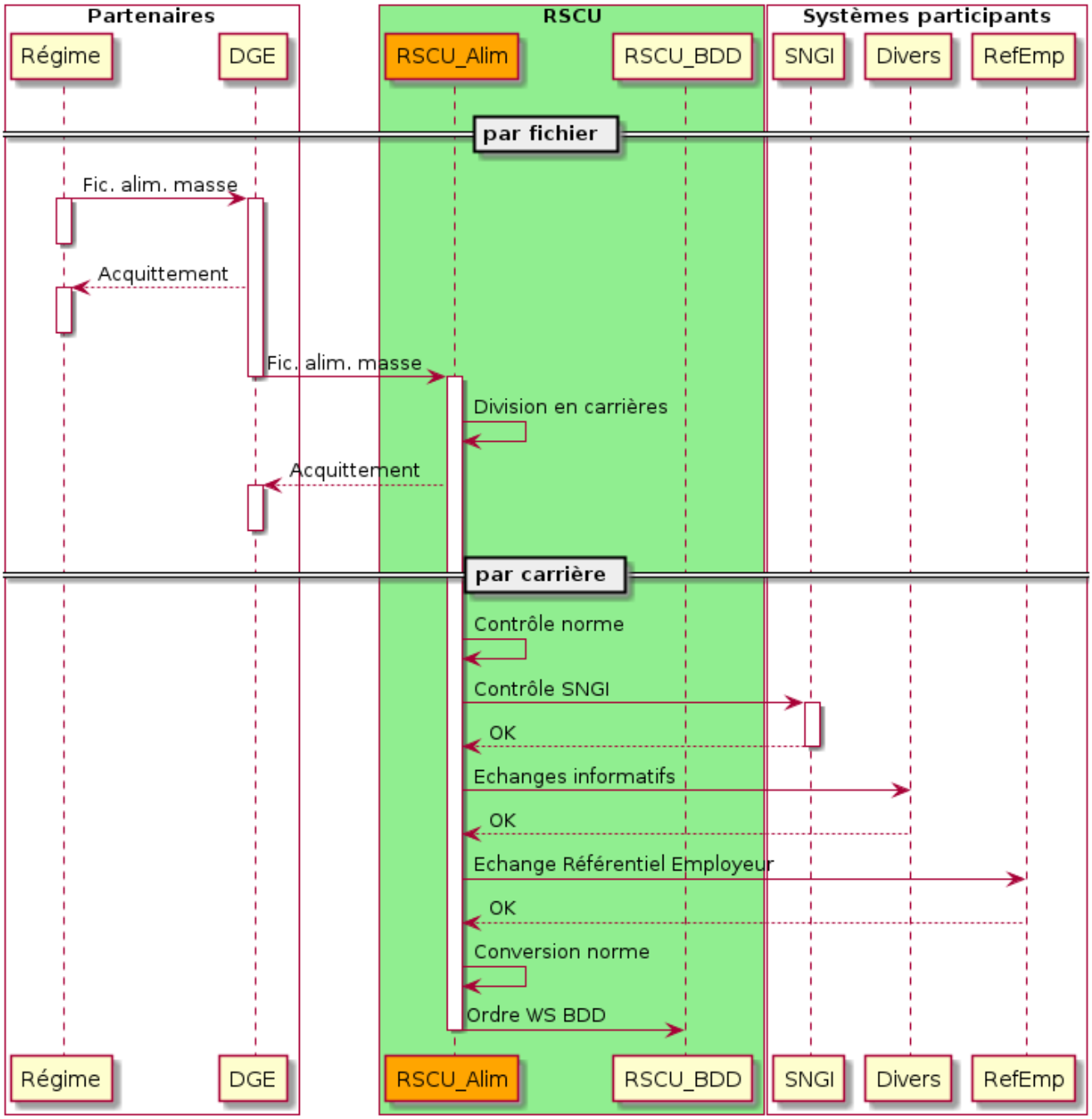
1. Introduction

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



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.

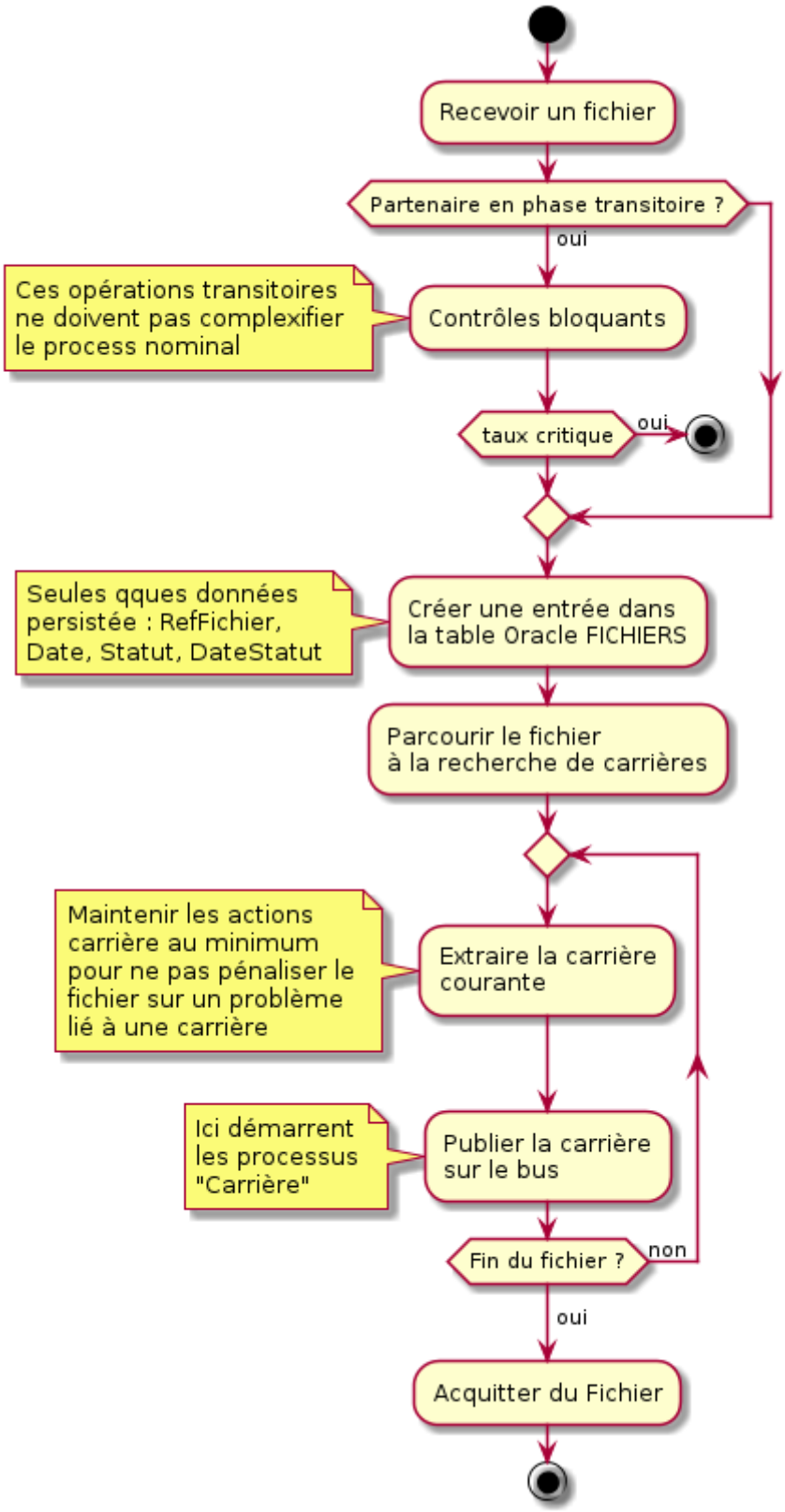
La proposition commerciale était à l'origine prévue sur webMethods pour la masse et l'unitaire factorisés. Dans l'hypothèse de Spring Batch en masse, notre préconisation est d'abandonner webMethods en unitaire pour les raisons suivantes :

- 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.

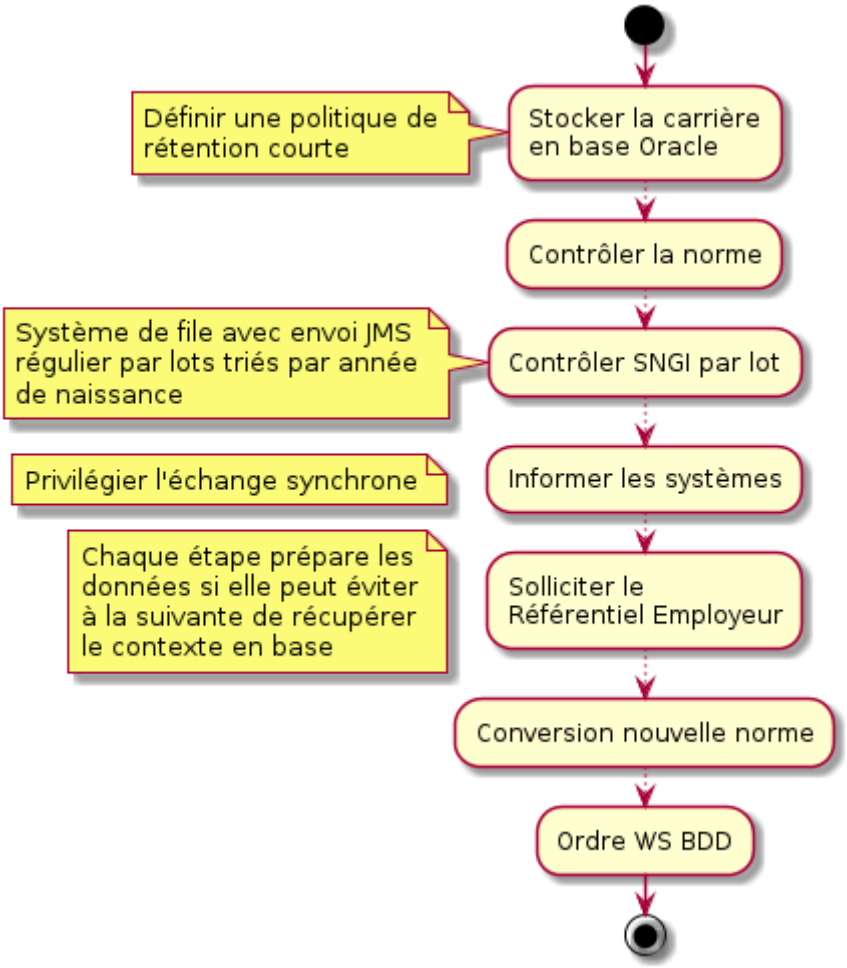


5. Diagramme d’activités macro "Carrière"

Le processus "Carrière" peut être à la fois le traitement d’un élement 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.



6. Annexes

7. Annexes

7.1. Marques de révision

Différences depuis le dernier tag