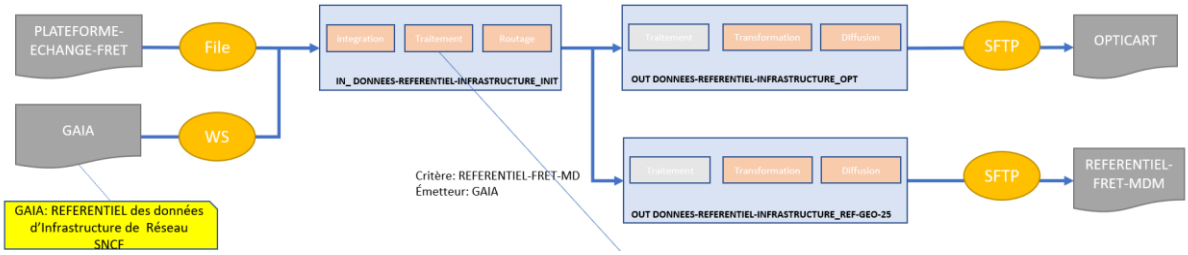

1. **Le fichier en entrée est de type file, il passe à la phase suivante grâce à un quartz.**

On récupère les informations du quartz grâce a ce code qui permet de filtrer le fichiers xml comprenant les balises quartz :

```
 public class FluxQuartz {

        public static void main(String[] args) throws
ParserConfigurationException, SAXException {

            try{
                System.out.println(ProcessQuartz());
            }
            catch(IOException e){
                System.out.println(e);
            }

        }
        public static List<String> ProcessQuartz() throws
ParserConfigurationException, SAXException, IOException{


            try{
                File file = new
File("C:\\Users\\0401281A\\Stage\\ProcessQuartz\\Quartz.xml");
                /*instanciassion necessaire*/
                DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
                DocumentBuilder db = dbf.newDocumentBuilder();
                Document document = db.parse(file);
                List<String> ListElement = new ArrayList<>();


                document.getDocumentElement().normalize();//normalisation du
document
                System.out.println("Balise racine : " +
document.getDocumentElement().getNodeName());
                NodeList nList = document.getElementsByTagName("quartz");
                System.out.println(nList.getLength());
                for (int i = 0; i < nList.getLength(); i++) {
                    Node nNode = nList.item(i);
```

```
                    if (nNode.getNodeType() == Node.ELEMENT_NODE) /*verifie
qu'il s'agit d'un element XML*/ {
                            nNode = nList.item(i);
                            Element eElement = (Element) nNode; //Cast
permettant d'utiliser les fonctions suivantes
                            String nomBalise =
eElement.getTagName();//recupere le nom de la balise
                            String contenuBalise =
eElement.getTextContent();//recupere le contenue de la balise
                            ListElement.add(nomBalise);
                            ListElement.add(contenuBalise);
                        }
                    return ListElement;
                }

            }
            catch(ParserConfigurationException e){
                System.out.println(e);
            }
            return null;

            }
        }
```

Puis on copie les balises quartz récupérer dans un autre fichier :

```
public class ProcessCopieQuartz {

    public static void main (String args[]) throws
ParserConfigurationException, SAXException, IOException{

        File src = new
File("c:\\Users\\0401281A\\Stage\\ProcessQuartz\\iufibi.xml");
        File dest = new
File("c:\\\\Users\\\\0401281A\\\\Stage\\\\ProcessQuartz\\\\CopieQuartz.xml");
//copie du fichier
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        DocumentBuilder db = dbf.newDocumentBuilder();
        Document document = db.parse(src);//anlyse le fichier xml
        InputStream is = null;//permet de lire des donnees
        OutputStream os = null;//permet d'ecrire des donnees

        //renvoie le nombre de balise quartz
        NodeList nList = document.getElementsByTagName("quartz");
                System.out.println(nList.getLength());
        try {
            is = new FileInputStream(src); //pointe vers FileInputStream
            os = new FileOutputStream(dest); //pointe vers FileInputStream
            byte[] buffer = new byte[1024]; //cree un tableau de byte
            int len;
```

```
            while ((len = is.read(buffer)) > 0) /*recupere les donnes de is et
les stocke dans len puis les ecrit dans os */ {
                os.write(buffer, 0, len);
            }
            is.close();
            os.close();
        }
        catch(IOException e){
            e.printStackTrace();
        }
    }
    }
```

```
<congif_quartz>
        <quartz ExpressionQuartz=«0+0+20+*+*+?» CodeFluxService=«rebarareg»> Format=«csv» Destinataire=«REFERENTIEL-FRET-MDM»
PatternDestinataire=«RGPRALPR_GAIA_[AAAA][MM][JJ][HH][MI][SS].txt» EncodingDestinataire=«UTF-8»>
        <quartz ExpressionQuartz=«0+0+21+*+*+?» CodeFluxService=«repralpr» Format=«csv» Destinataire=«OPTICART»
PatternDestinataire=«REPRALPR_GAIA_[AAAA][MM][JJ][HH][MI][SS].txt» EncodingDestinataire=«UTF-8»>
        …
</congif_quartz>
```

Au démarrage du bundle:

- On charge le fichier xml «FluxQuartz.xml »
- On Garde en cache les informations du fichier xml «FluxQuartz.xml » (notamment pour récupérer l'encoding du flux de sortie dans les bundles de sortie)
- On créait un process quartz pour chaque balise <quartz>
- Chaque quartz se déclenche à l'heure correspondant sont expression quartz et génère un fichier INIT_< CodeFluxService >.xml au format xml suivant:

2. Dans cette phase il passe par :

- **L'intégration :**
    - Consommation du fichier INIT par l'adaptateur (Pattern INIT_*.xml)
    - Les informations du fichier INIT doivent être propagé dans le Header Technique des fils GMS* dans toutes les phases du flux.

- **Le traitement**
    - **Création des Paramètres métiers*:**
        - CODE_FLUX:XPATH('Init_ Donnees_Referentiel_Infrastructure /CodeFluxService')
        - FORMAT:XPATH('Init_ Donnees_Referentiel_Infrastructure /Format')
        - DESTINATAIRE:XPATH('Init_ Donnees_Referentiel_Infrastructure /Destinataire')
        - PATTERN_DESTINATAIRE:XPATH('Init_ Donnees_Referentiel_Infrastructure /PatternDestinataire')
        - NB_LIGNE:Nombre de ligne retournée par la requête WS à GAIA
    - **Appel du WebService GAIA:**
        - curl -X GET "https://gateway.dgexsol.fr/precomputed_query/reseau/<Code FluxService>/?format=<Format>" -H "Accept: application/json" -H "Authorization: Bearer {TOKEN}
        - SI TimeOut ou Code Erreur → KO_TEC*

- SI les para mètres <CodeFluxService> et <Format> sont pas conforme KO_FONC*

- **Le routage**

3. **En fonction du critère il passera ensuite en phase de :**
   - **Traitement :**
   - **Transformation :**
   - **Diffusions :**
4. **Puis est transmis via SFTP**

* Fils GMS : tunnels par lesquels vont transiter les différentes étapes à partir de l'intégration jusqu'à la diffusion.

*Paramètre métier : Différents paramètres permettant de retrouver le fichier.

*Ko_Tec : Problème technique lié au processus. (ex : serveur éteint, fichier non envoyé…)
*Ko_Fonc : Problème lié aux données. (ex : données non conforme)