**Extract Workflow**

**EmmaHandler**

EmmaHandler downloads new XML with N Missioni

EmmaHandler will parse the XML into N JSON Missioni

EmmaHandler will send the N JSON Missioni over to Kafka

**Transform Workflow**

**EmmaTommyConverter**

EmmaTommyConverter will consume a new JSON Missione from Kafka

EmmaTommyConverter will convert the JSON Missione to a JSON Servizio

EmmaTommyConverter will send the JSON Servizio over to Kafka

**TommyHandler**

TommyHandler will consume a new JSON Servizio from to Kafka

TommyHandler will determine what to do with the new Servizio based on the following cases

* If the servizio (by ID) is already present in the *Persistence DB*
  + If the servizio has the same data (no update), discard it
  + If the servizio has updates, send error notification for manual update
* If the servizio (by ID) is not present in the *Persistence DB*
  + If the servizio (by ID) is already present in the *Staging DB error section*
    - If the servizio has the same data (no update), discard it
    - If the servizio has updates, update the database entry and move it to the new section
  + If the servizio (by ID) is not present in the *Staging DB error section*
    - If the servizio (by ID) is present in the Staging DB new section
      * If the servizio has the same data (no update), discard it
      * If the servizio has updates, update the database entry
    - If the servizio (by ID) is not present in the Staging DB new section
      * Decorate the servizio and create a new DB entry for it

**Load Workflow**

**TommyPoster** (Ciclically every X time)

Lock the *Persistence DB* (if it fails, wait till it can be locked with timeout)

Lock the *Staging DB* (if it fails, unlock the *Persistence DB* and abort)

For all the DB entries in the *Error Section*

* Set the MEZZO\_ID as blocked for posting.

For all the DB entries in the New *Section* (Analysis ordered by servizio ID)

* Check if the Servizio is valid:
  + KM > 0
  + It has to have a MEZZO\_ID
  + The MEZZO\_ID must not be blocked for posting
  + The servizio should not be also present in the *Error Section* *(*Double Checking*)*
  + The servizio should not be also present in the *Persistence DB* *(*Double Checking*)*
* If the servizio is valid add it to the list of servizi to be posted (classified by MEZZO\_ID)
* If the servizio is not valid, set the MEZZO\_ID as blocked for posting

For every MEZZO\_ID list (if not empty)

* If there isn’t a poster for that MEZZO\_ID, spawn it
* Send the list to the poster for that MEZZO\_ID
* Wait for the Poster Response, which can be of three types:
  + Unable to Send Over to REST
  + Sent Over to REST
    - REST accepted the data
      * Send eventual received warnings for manual corrections
      * Write all the posted servizi in to the *Persistence DB*
      * Remove all the posted servizi from the *Staging DB* New *Section*
    - REST refuted the data with error

(the whole list of servizi is not posted if at least one servizio generates an error)

* + - * Move all the the unposted servizi from the *Staging DB* New *Section* to the *Error Section*

Unlock the *Persistence DB*

Unlock the *Staging DB*