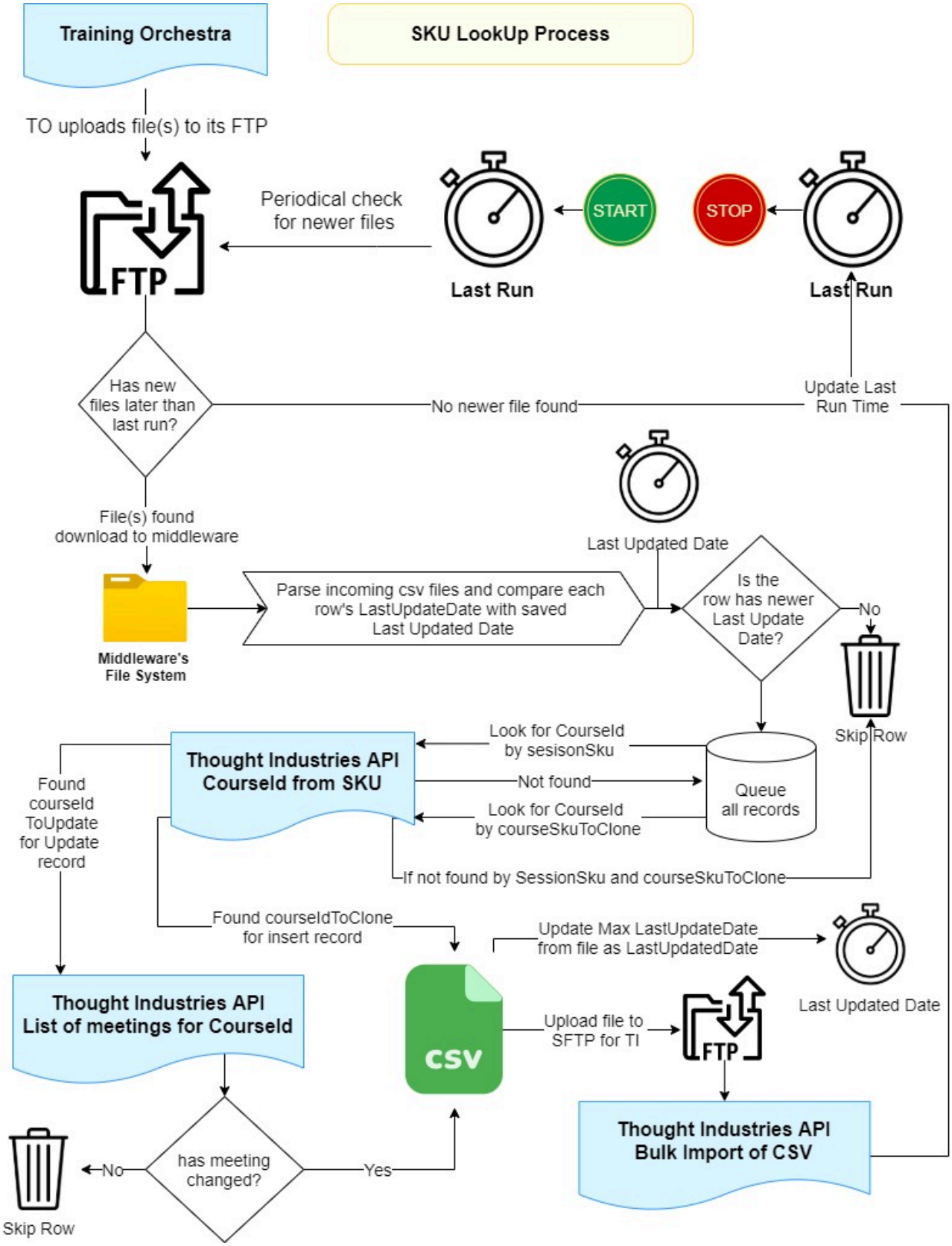
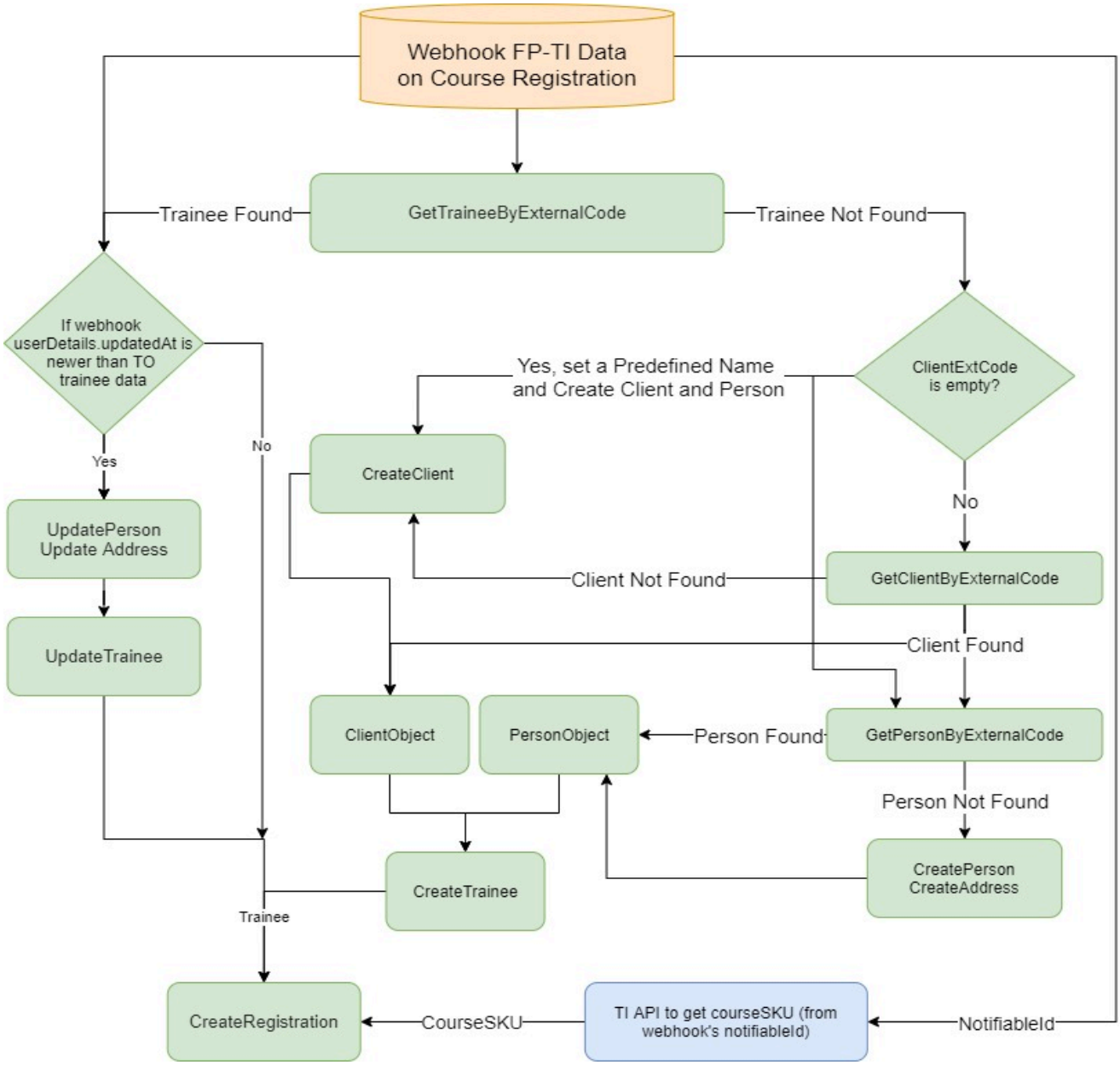

Training Orchestra
SKU LookUp Process
TO uploads file(s) to its FTP
FTP
Periodical check for newer files
START
STOP
Last Run
Last Run
Has new files later than last run?
No newer file found
Update Last Run Time
File(s) found download to middleware
Last Updated Date
Middleware's File System
Parse incoming csv files and compare each row's LastUpdateDate with saved Last Updated Date
Is the row has newer Last Update Date?
No
Skip Row
Look for CourseId by sesisonSku
Thought Industries API CourseId from SKU
Not found
Queue all records
Look for CourseId by courseSkuToClone
Found courseId ToUpdate for Update record
If not found by SessionSku and courseSkuToClone
Found courseIdToClone for insert record
Update Max LastUpdateDate from file as LastUpdatedDate
Last Updated Date
Thought Industries API List of meetings for CourseId
CSV
Upload file to SFTP for TI
FTP
Thought Industries API Bulk Import of CSV
has meeting changed?
No
Yes
Skip Row

SKU Lookup Process

1. TO uploads file(s) to TO SFTP location.
2. Periodic sku lookup process starts.
3. Got newer file(s) at TO SFTP location compared to Last Run time.
4. Download newer files to Middleware's folder.
5. Parses downloaded file(s) and filters out records which are older than the Last Updated Date saved on the server.
6. Updates last Updated Date with the maximum LastUpdateDate field of csv.
7. Traverses each of the records and looks for courseId by sessionSku field.
8. If found, the record will be treated as courseIdToUpdate.
9. If not found, it will look for courseId by courseSkuToClone field.
10. If found, the record will be treated as courseIdToClone.
11. If not found by courseSkuToClone too, then skip the record altogether.
12. Arrange all the records into files based on the TI_Bulk_Import_Record limit.
13. For records ready for update, look for meetings by courseIdToUpdate in TI API.
14. If all the TI API meetings match exactly the same as that of file startDate and endDate, then skip the row else keep the record intact as update.
15. On processing all the file records, upload the processed CSV to TI SFTP for bulk importing.
16. Bulk import API is called for insert/update of records.
17. Update Last Run on server.
18. Sku lookup process is complete now.

Registration/Enrollment Process (Webhook)



1) Thought Industries triggers a webhook which is caught by middleware.

2) Middleware triggers the registration process.

3) Check Trainee Exists: Webhook's userDetail.ref7 is used to look for Trainee by hitting TO API => OF_Trainee/GetTraineeByExternalCode?ExternalCode=userDetail.ref7

a) If Trainee found, Check Person Exists: Look for person in TO by using Webhook's userDetail.email
OF_Persons/GetPersonByExternalCode?ExternalCode=userDetail.email

(1) If Person found, Compare Trainee's LastUpdateDate with Webhook's updatedAt

(a) If Webhook's updatedAt is newer then:

(i) Update Person Address with below field maps:

If person.AddressID is present, hit OF_Address/UpdateAddress
If person.AddressID is null, hit OF_Address/CreateAddress

1. Title: "Main Address"
2. Street: userDetail.address1
3. AdditionalInformation1: userDetail.address2
4. Zipcode: userDetail.zipCode
5. City: : userDetail.city
6. Region.code: code (by state name)
7. Region.name: userDetail.state
8. Country.code: code (by country name)
9. Country.name: userDetail.country

(ii) Then Update Person with below field maps:

Hit OF_Persons/UpdatePerson

1. Active: true
2. Surname: userDetail.lastName
3. Birthname: userDetail.firstName + ' ' + userDetail.lastName
4. Firstname: userDetail.firstName
5. EntityID: trainee.EntityID
6. EntityCode: trainee.personExternalCode
7. PhoneNumber: userDetail.ref6
8. ExternalCode: userDetail.email
9. ID: trainee.personID
10. ClientExternalCode: trainee.ClientExternalCode

(iii) Then Update Trainee (OF_Trainee/UpdateTrainee) with below field maps:

1. IsActive: true
2. PersonExternalCode: trainee.PersonExternalCode
3. ClientExternalCode: trainee.ClientExternalCode
4. ExternalCode: trainee.ExternalCode

5. ID: trainee.ID

(iv) Return updated trainee object

(b) If Trainee's LastUpdateDate is newer then return trainee object.

(2) If person, not found, throw error, script ends.

b) If Trainee not found,

i) Find/Create Client:

(1) If userDetail.ref2 is empty,

(a) then fetch default client by external code "DEFAULTTEXTCODE2". OF_Clients/GetClientByExternalCode

(b) If default client not found then create the default client with following fields:
Create client via OF_Clients/CreateClient
(i) IsActive: true
(ii) Title: "DEFAULT"
(iii) Type.Code: "GRP"
(iv) EntityID: 1003171
(v) ExternalCode: "DEFAULTTEXTCODE2"

(c) If found, return the default client

(2) If userDetail.ref2 is not empty, then fetch client by userDetail.ref8 by hitting TO API OF_Clients/GetClientByExternalCode

(a) If found, return the client.

(b) If not found, then create and return the client with following fields:
Create client via OF_Clients/CreateClient
(i) IsActive: true
(ii) Title: userDetail.ref2
(iii) Type.Code: "GRP"
(iv) EntityID: 1003171
(v) ExternalCode: userDetail.ref8

ii) Find/Create Person

(1) Look for person by userDetail.email via OF_Persons/GetPersonByExternalCode

(2) If found, return person

(3) If person not found,

(a) Create person address via hit OF_Address/CreateAddress

(i) Title: “Main Address”
(ii) Street: userDetail.address1
(iii) AdditionalInformation1: userDetail.address2
(iv) Zipcode: userDetail.zipCode
(v) City: : userDetail.city
(vi) Region.code: code (by state name)
(vii) Region.name: userDetail.state
(viii) Country.code: code (by country name)
(ix) Country.name: userDetail.country

(b) Create person via hit OF_Persons/CreatePerson

(i) Active: true
(ii) Surname: userDetail.lastName
(iii) Birthname: userDetail.firstName + ‘ ’ + userDetail.lastName
(iv) Firstname: userDetail.firstName
(v) EntityID: 1003171
(vi) ContactInformation.PhoneNumber: userDetail.ref6
(vii) ContactInformation.Email: userDetail.email
(viii) ExternalCode: userDetail.email
(ix) AddressID: Address.ID as returned by above person address creation.

(c) Return the Person object as returned by create person above.

iii) Create Trainee

(1) IsActive: true
(2) ClientID: Client.ID (as received from client object above)
(3) ClientExternalCode: Client.ExternalCode (as received from client object above)
(4) PersonID: Person.ID(as received from Person object above)
(5) PersonExternalCode: Person.ExternalCode (as received from Person object above)
(6) EntityID: 1003171
(7) ExternalCode: userDetail.ref7

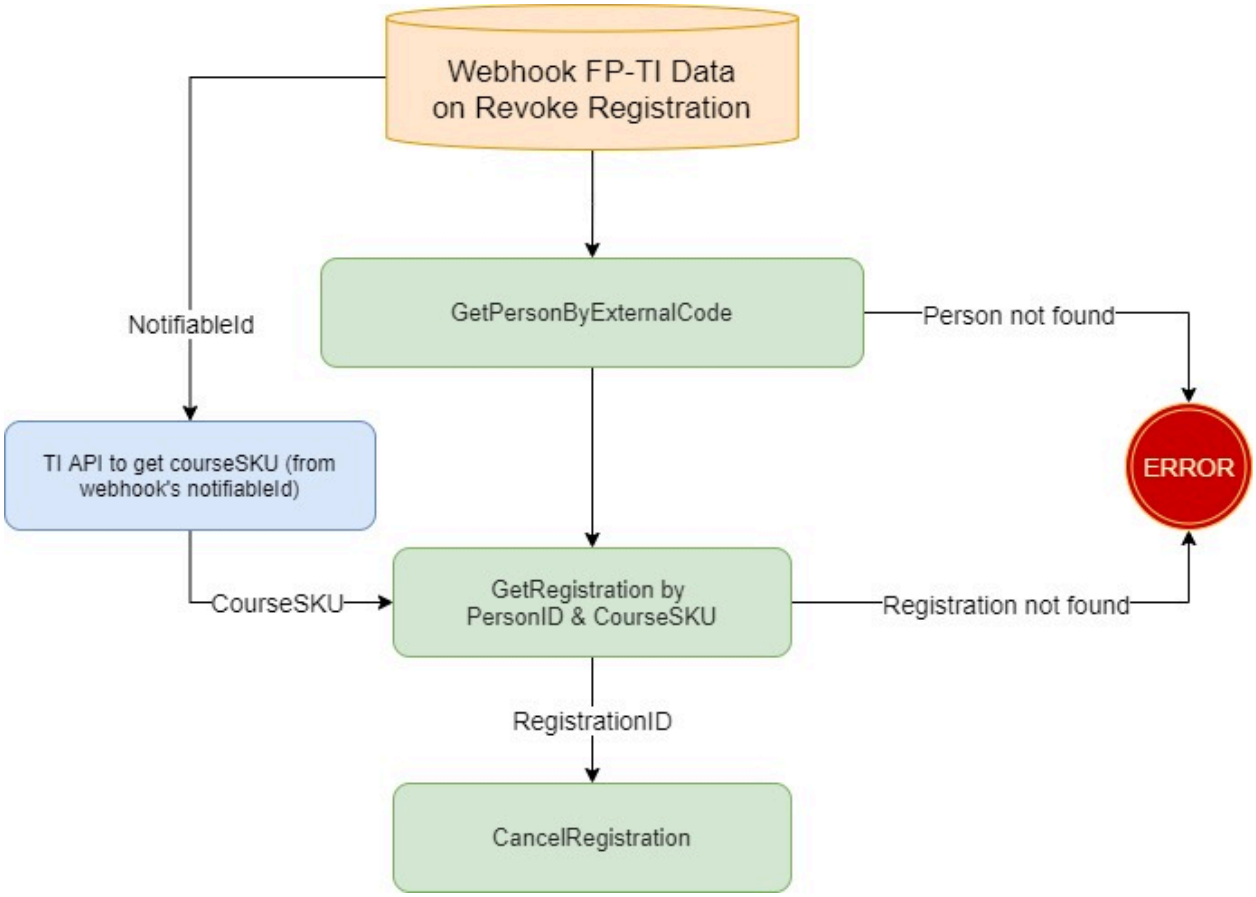
Return the trainee object thus created.

4) Now that we have the trainee object, lets find out the CourseSKUById from TI API by using webhook's notifiableId , return the CourseSKU received back.

5) Final step Create Registration for which we will use trainee object data and courseSKU. TO API OF_Registration/CreateRegistration is used with following fields:

    a) ClientExternalCode: trainee.ClientExternalCode
    b) SessionID: CourseSKU
    c) TraineeID: trainee.PersonID
    d) TraineeExternalCode: trainee.ExternalCode

Revoke Access / De - Enrollment Process (Webhook)



1. CheckPersonExists by ExternalCode from userDetail.email
    a. If not found, then throw error and stop
    b. If found then return Person object

2. ~~Hit TI API to fetch CourseSKU from webhook's notifiableId (CourseID in TI)~~
   Available in webhook's courseSku

3. With PersonID and CourseSKU, we fetch Registration from TO API.
    a. If Registration not found in TO, then throw error and stop
    b. If registration found, then cancel registration with AbsenceReasonCode = ...