**2.2.6.2 Reconciler Activity Diagram and Algorithm**

The reconciler is designed to perform files and folders synchronization between the two replicas. Based on the previous and current states of the replicas, the reconciler attempts to reconcile the conflicts between the replicas in the best possible way. In general, the reconciler conflict resolution follows the two key principles:

* If changes occurred only on one replica, propagate the changes directly to another replica.
* If changes occurred on both replicas, handle the conflicts using the pre-defined user settings and conflict resolution policy.

For each replica, detector generates four different lists:

1. List of files that have changes need to propagate over to another replica.
2. List of files that don’t have changes.
3. List of folders that have changes need to propagate over to another replica.
4. List of folders that don’t have changes.

Types of changes on each replica:

1. Creation of files and folders
2. Deletion of files and folders.
3. Modification of files.
4. Renaming of files.

For each synchronization task, the reconciler receives eight lists (four lists for each replica) from the detector and will determine the changes to be propagated between the replicas. The following describes how the algorithm performs the conflict resolution.

Note: For ease of explanation, we name the first replica as *source* and second as *target* and the four lists as following:

1. *DirtyFilesList* - List of files that have changes.
2. *CleanFilesList* – List of files don’t have changes.
3. *DirtyFoldersList* – List of folders that have changes.
4. *CleanFoldersList* – List of folders don’t have changes.

Algorithm (Files):

1. Check *DirtyFilesList* on each replica for renamed files.
   1. For each entry in the *DirtyFilesList*.
      1. Check the entry relative path, flag and hash code.
      2. If the entry with a “CREATE” flag has another corresponding entry (same hash code but different relative path) with a “DELETE” flag, we consider this is a rename file.
2. Traverse through the *DirtyFilesList* on *source* replica to perform reconciling.
   1. For each entry in the *DirtyFilesList*.
      1. Check any similar changes on the *target* replica.
      2. If no changes made on the *target* replica (check the *target CleanFilesList)*, propagate changes from *source* to the *target* replica.
      3. If changes made on the *target* replica (check the *target DirtyFilesList*), this is consider a conflict and the action taken is based on by the user pre-defined setting and the conflict resolution policy.
3. Traverse through the *DirtyFilesList* on *target* replica.
   1. For each entry in the *DirtyFilesList*.
      1. Propagate the changes from the *target* to the *source* replica. Note: Only need to propagate changes from the *target* to *source*, since the above function has already handled all conflicting changes between *source* and *target*.

Once the files-level synchronization is done, the reconciler will perform folder cleanup process. The reconciler detects and performs folder creation, renaming and moving through the file level. Each entry in the file list contains the relative path which has the folder information. The file operation performs will create all required folders during the synchronization. The following describes how the algorithm performs the folder cleanup.

Algorithm (Folder):

1. Traverse through the *DirtyFoldersList* on *source* replica.
   1. For each entry in the *DirtyFoldersList*.
      1. If the entry with a “CREATE” flag, propagate changes to the *target.*
      2. If the entry with a “DELETE” flag and corresponding entry in the *target DirtyFoldersList*, propagate changes to the target.
      3. If the entry with a “DELETE” flag and corresponding entry in the *target CleanFoldersList*, propagate changes to the *target* if the *source* doesn’t have the folder.
2. Traverse through the *DirtyFoldersList* on *target* replica.
   1. For each entry in the *DirtyFoldersList.*
      1. If the entry with a “CREATE” flag, propagate changes to the *source.*
      2. If the entry with a “DELETE” flag and corresponding entry in the *target CleanFoldersList*, propagate changes to the *source* if the *target* doesn’t have the folder.

Conflict Resolution Policy

The following policy is used by the reconciler to automatically handle reconciling conflicting changes:

* Concurrent modified conflict: If existing file is modified independently on each replica, the reconciler prefers to keep both copies and rename both files to prevent name collision, but user can change this default action through the user setting.
* Concurrent modified-delete conflict: If existing file is modified on one replica, and the same file is deleted on another replica. The reconciler prefers to keep the modified file to prevent loss of date. User can change this default action through the user setting.
* Name collision - concurrent create conflict: If files with same name are created on replicas, the files’ last write time and hash code is used to determine whether the content is similar. If the content is similar, no propagation is needed.
* Name collision - concurrent create-rename conflict: If the rename of an existing file collides with the creation of another file on another replica. The reconciler prefers to keep both copies and rename both files to prevent name collision but user can change this default action through the user setting.
* Name collision – concurrent rename-rename conflict: If different files on the replicas are renamed to the same name. The reconciler prefers to keep both copies and rename both files to prevent name collision but user can change this default action through the user setting.
* Folders rename/move: The existing folders on both replicas are renamed to different name. The reconciler is able to detect the folder renaming through the file level and change both folders to the same name.