# File Sharing Service

Stefano Borghi | August 16, 2022

**SyncService**: monitors the user folder and writes ClientChange events to the SyncQueue; it also applies applicable ServerChanges (diffing the target resource) as instructed by the QueueWorker

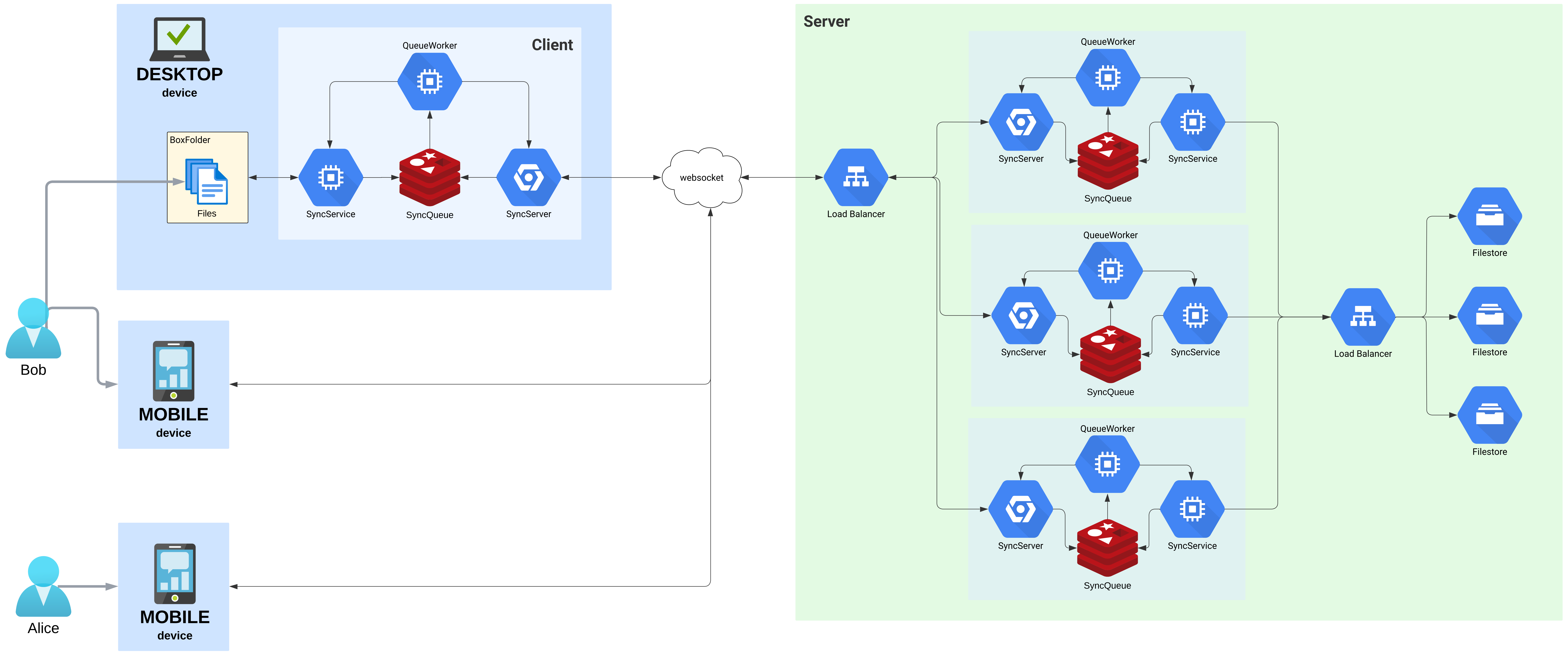
**SyncServer**: writes ServerChange events, received from the internet, to the SyncQueue; it sends ClientChanges to the server if instructed by the QueueWorker

**QueueWorker**: processes the queue and dispaches changes to the local folder or the server

**SyncServer**: receives ClientChange and saves them to the SyncQueue; it also sends ServerChanges, passed by the QueueWorker, to all the user's clients.
It also generate the url link for shaing resources.

**QueueWorker**: processes the queue and dispaches each record as ClientChange to the SyncService and to the SyncServer (as ServerChange) as well, in order to allow all clients (of the same user) to remain in sync

**SyncService**: receives ClientChange from the QueueWorker and pllies them, if necessary, to the FileStorage



## API

### Folders

```
POST folder.add -> folderId
  {userId, deviceId, folderId,
   newFolderName, newFolderContent}

POST folder.delete
  {userId, deviceId, folderId}

PATCH folder.rename
  {userId, deviceId, folderId, newName}

PATCH folder.move
  {userId, deviceId, folderId, newPath}
```

### Files

```
POST file.add -> fileId
  {userId, deviceId, folderId,
   fileName, fileContent}

DELETE file.delete
  {userId, deviceId, fileId}

PUT file.update
  {userId, deviceId, fileId, fileContent}

PATCH file.rename
  {userId, deviceId, fileId, newFileName}

PATCH file.move
  {userId, deviceId, fileId, newPath}
```

### Resources

```
GET resource.link -> url
 {userId, resourceId}

GET resource.find -> [url]
 {userId, resourceName}
```