

Introduction

Purpose

- Share files/directories between clients(machine)
- Synchronization of shared files/directories

Scope

- Client and Server program running on each side respectively and allowing user to share files between machines and sync files from server
- Servers backing each other up for failure detection/protection

Success Criteria

- Seamless interaction between clients and server
- Failure protection
- Protection of data integrity

Current System

- DropBox/GoogleSync/SkyDrive
 - Cloud based service that allows you to share files securely between clients of same user or amongst different users.

Proposed System

Overview

- Clients communicate with server to sync files between user clients
- Server provides redundant storage for back up purposes

Functional Requirement

Usability

- Program running processes in the background with minimal user interaction required therefore making the operation seamless

Supportability

- Should run on any operating system that has the version 1.7+ of java virtual machine installed

Interface

- Command-line interface to stop/stop program on Client side
- Command-line interface to stop/start program on Server side plus some administrative functions that may override the regular functionality

Performance

- Fast synchronization with respect to the network bandwidth
- Data integrity
- Failure protection

Packaging

- Runnable JAR file for both client and server programs

Extensibility

- Be able to share files between users
- Peer-to-Peer connectivity based on locality
- Ability to add GUI

Non-functional Requirement

Actors

- User of the software
- Server Administrator

Scenarios

- Login (a.k.a Getting Authenticated)
- Connect to Server (includes Getting Authenticated)
- Connect Shared Folder (includes Connect to Server)
- Client Syncs with Server (includes Connect to Server & Connect Shared Folder)
- Server Syncs with Backup Servers
- Shutting Down

Software Architecture

Overview

Design Goals

- Performance
- Reliability
- Integrity

Boundary Conditions

- Clients gets disconnected from Server
- Client shared folder gets deleted/corrupted
- Server shuts down

Subsystem Decomposition

- API
 - SYNC
 - CONNECTION
 - MANIFEST
- Client
 - Client Driver
 - Sync Manager
 - File Scan Manager
 - Session
 - Scanner
 - Manifest Comparator
 - File List
- Server
 - Server Driver
 - Sync Manager
 - File DAO
 - Client DAO

Persistent Data Management

- Database:
 - User
 - Username
 - Password
 - Profile

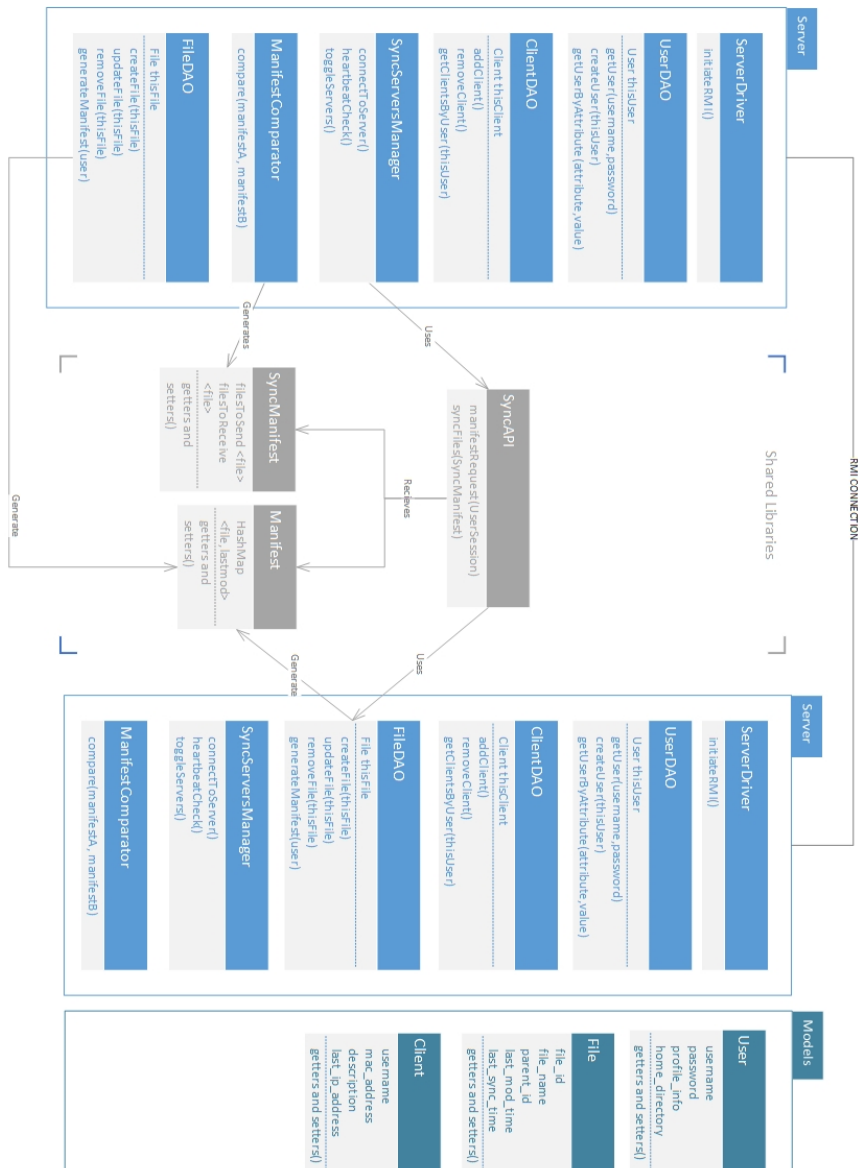
Foreign Key (Client & Files Table)
 - User_Client
 - User Name
 - Device Name
 - MAC Address
 - Files
 - File ID
 - Files Name
 - Parent
 - Last Modify Time
 - Last Sync Time
 - Files_User
 - User Name
 - File Table

Access Control Security

- User authentication
- Maintaining list of user clients using MAC addresses

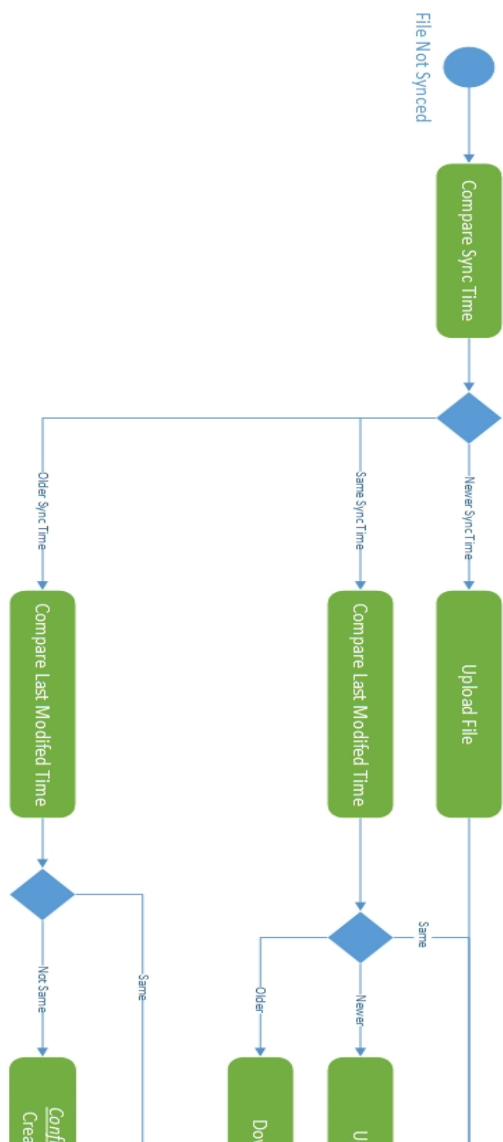
Client-Server Sync

Back-Up Server Sync

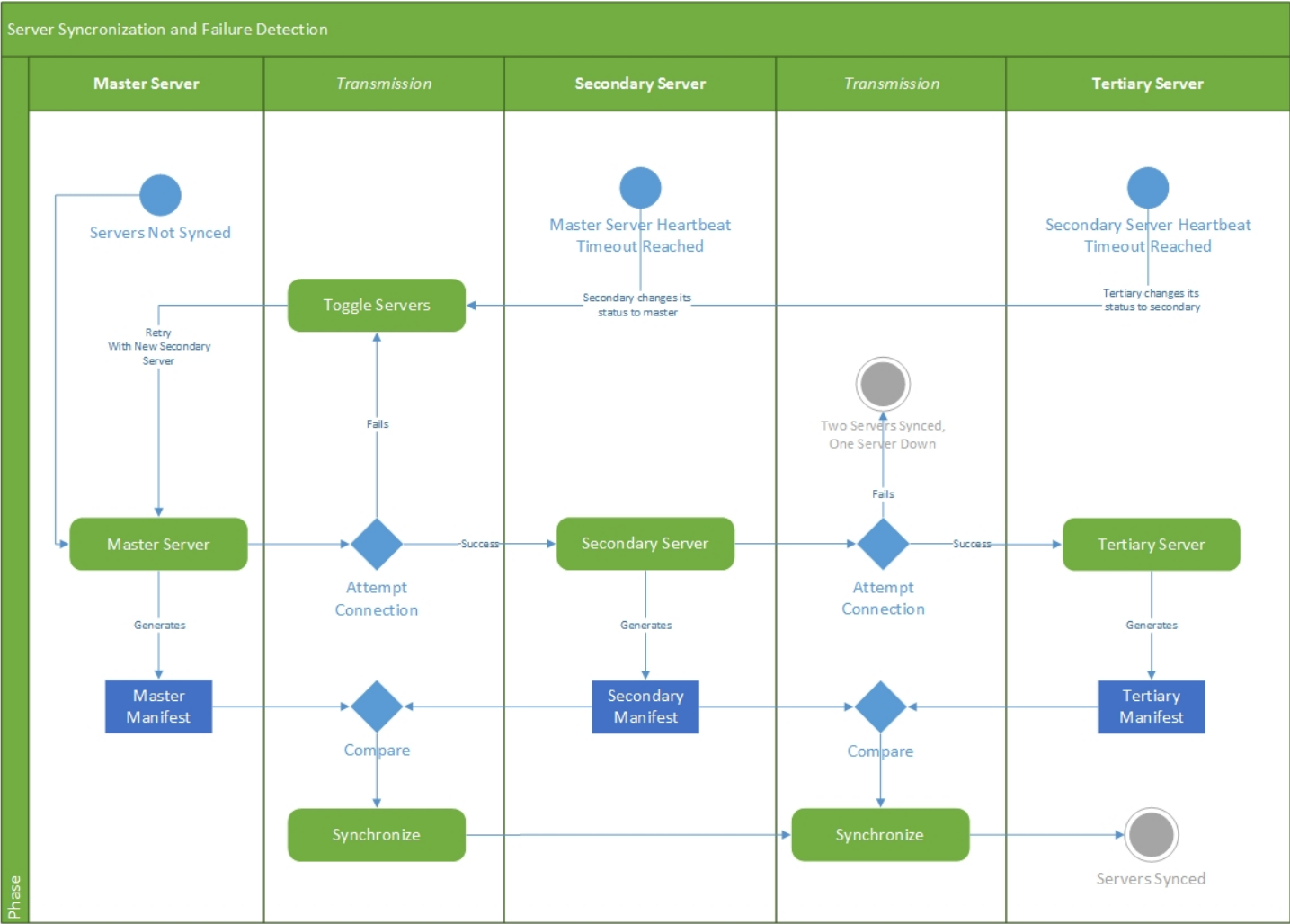


Sequence Diagrams

File Sync: Client and Server

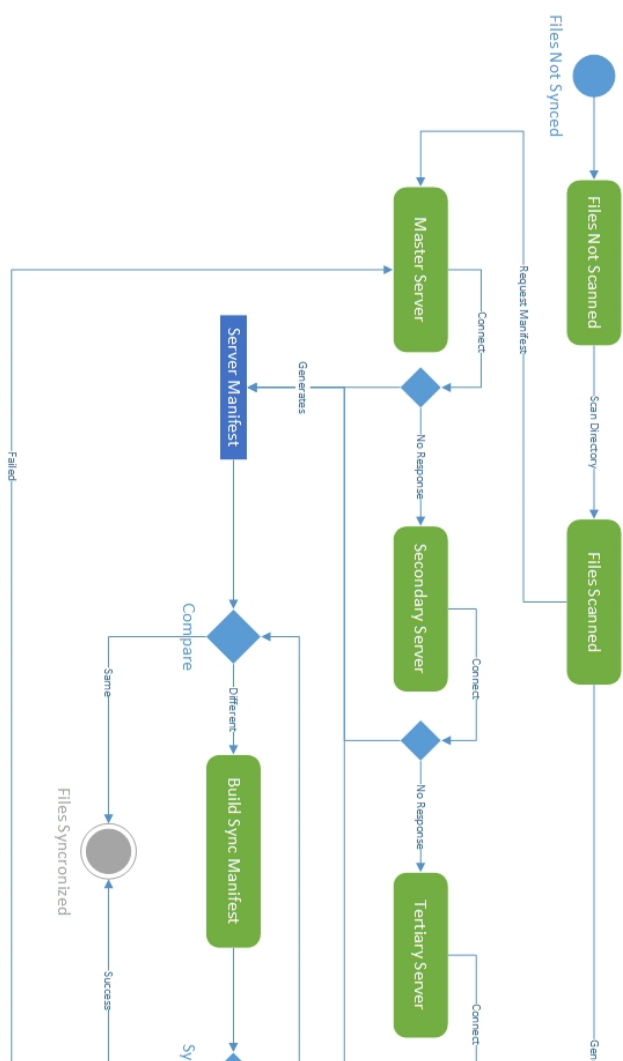


Server Back Up & Failure Protection



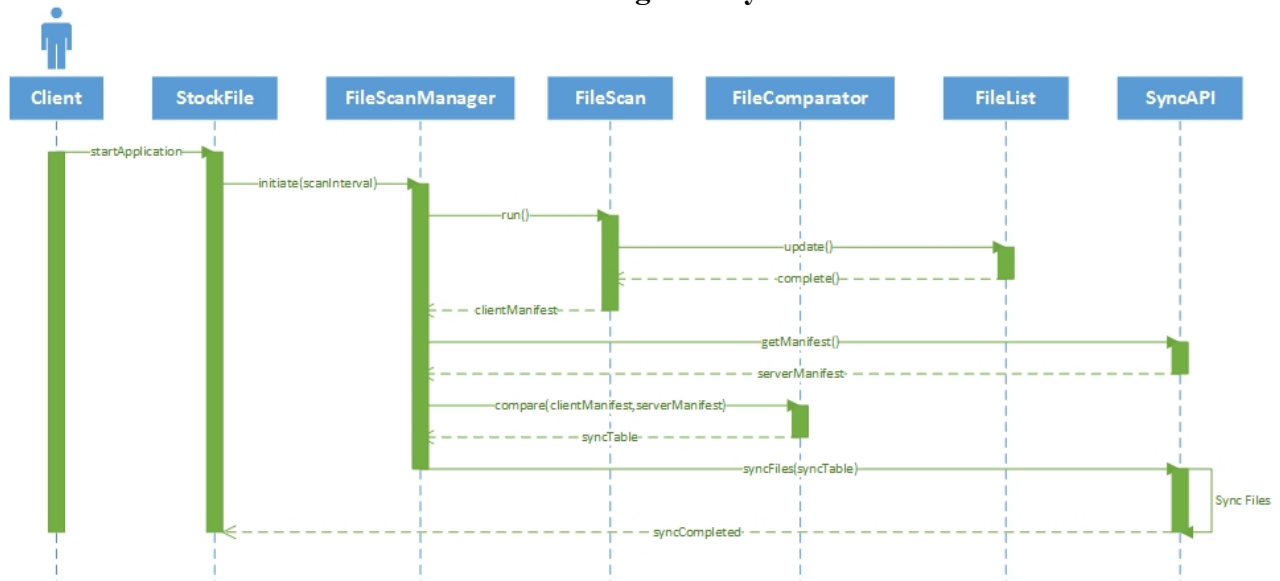
Phase

Server Back Up & Failure Protection



State Diagrams

Client Program: Sync



Client-Server API

