



Operating System

File Synchronization

Presentation 1
Design Documentation
Software Development Life Cycle

— 01





Group Members

— 02



Harshit Kumar
18110063



Shivam Sahni
18110159



Harsh Patel
18110062



Varun Jain
18110185





Requirements

Overview of the Project

While working, often a time we want our data to move with us. We want to work on the data, while not disturbing others workflow. It is important for data to synchronize with people working on it.

We aim to deliver a software that will help clients to work on their systems without worrying about data loss. This software could synchronize folders and files across multiple computers over a local area network.

This solves problems related to data backup, work - home taskflow and also team collaborations.



— 03





Feature Set | Objectives

- Properly be able to synchronize two folders that are mutually exclusive.
- Expanding these folders to over two separate computers.
- Multi-Threaded mode of working.
- Low Latency.
- Expandable to synchronize the folder for N computers.





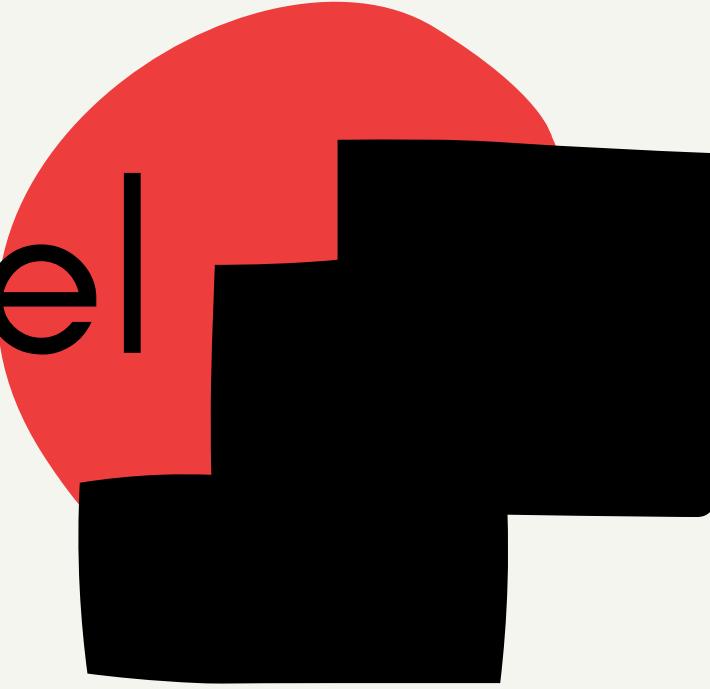
Software Development Life Cycle

— 05





Iterative Model



— 06



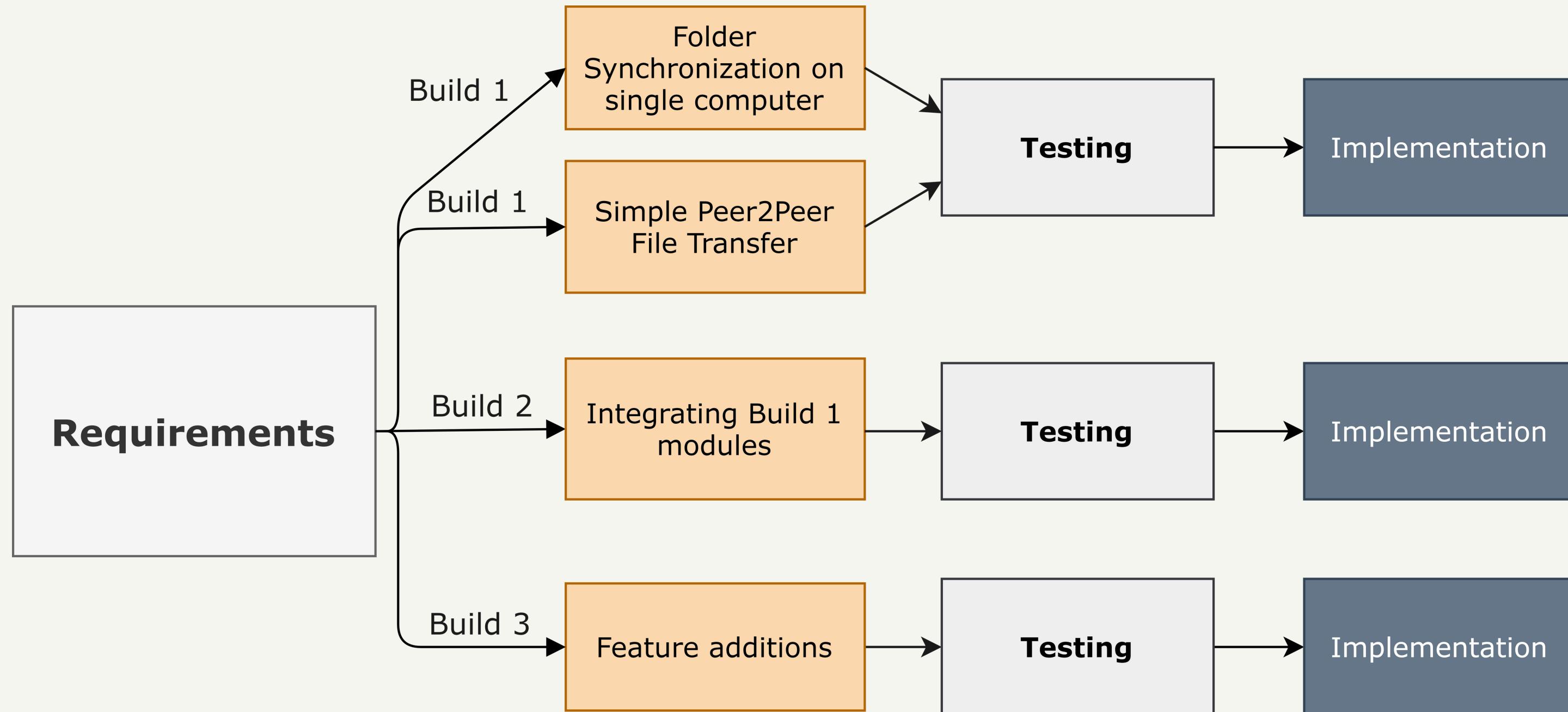
Hypothesis

We'll build the project in steps.
With successive steps, the complexity
and breadth of the project increases.





Iterative Design Flow



— 07





Project Timeline

Oct 28

We start with modeling
the Software
Development Life Cycle
and propose our design
documentation for the
software

Nov 20
Build 2

Testing and Debugging.
Adding more features
and making it bug-proof.

— 08

Phase 1

Phase 2

Phase 3

Road Ahead

Nov 10
Build 1

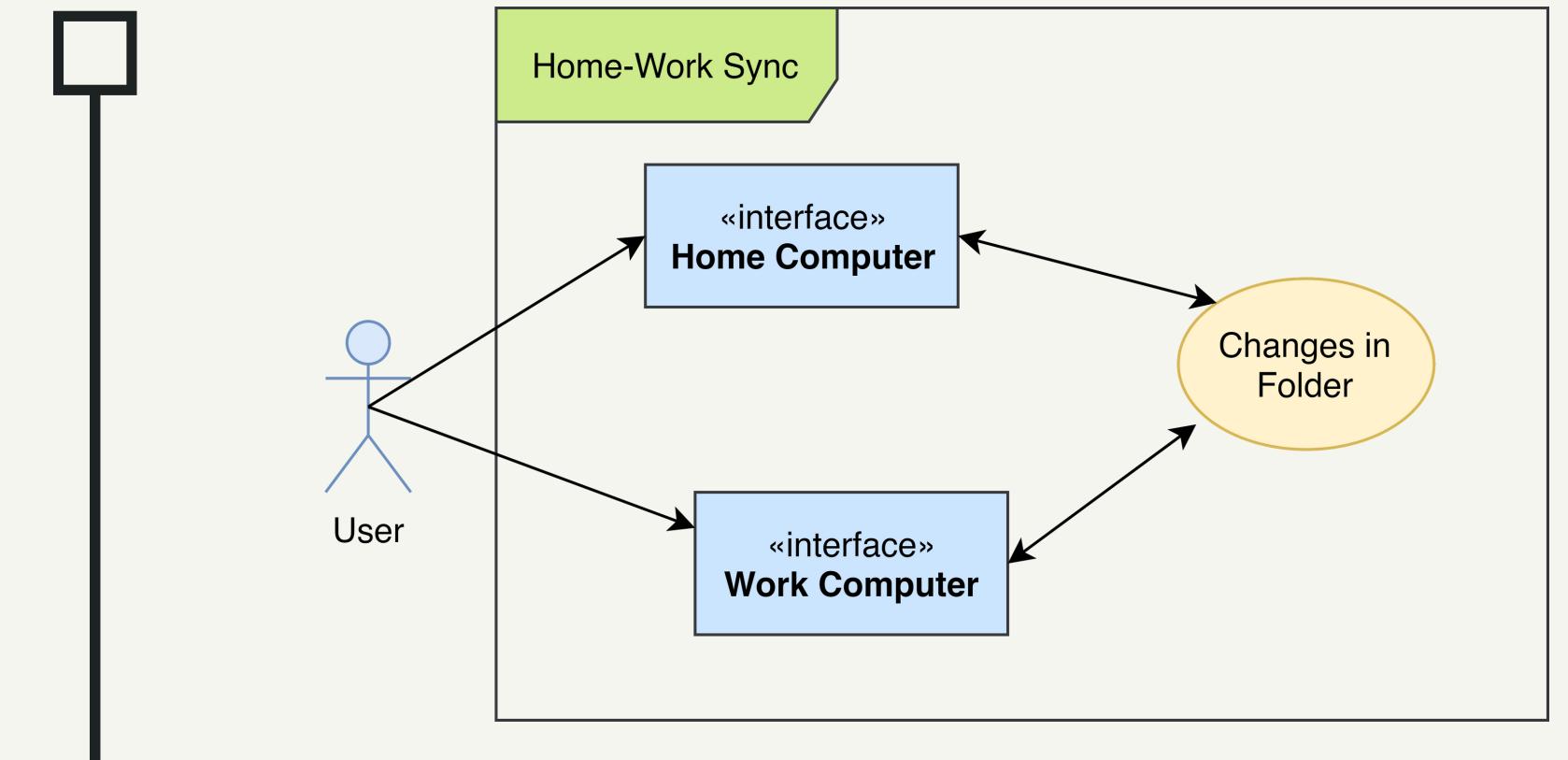
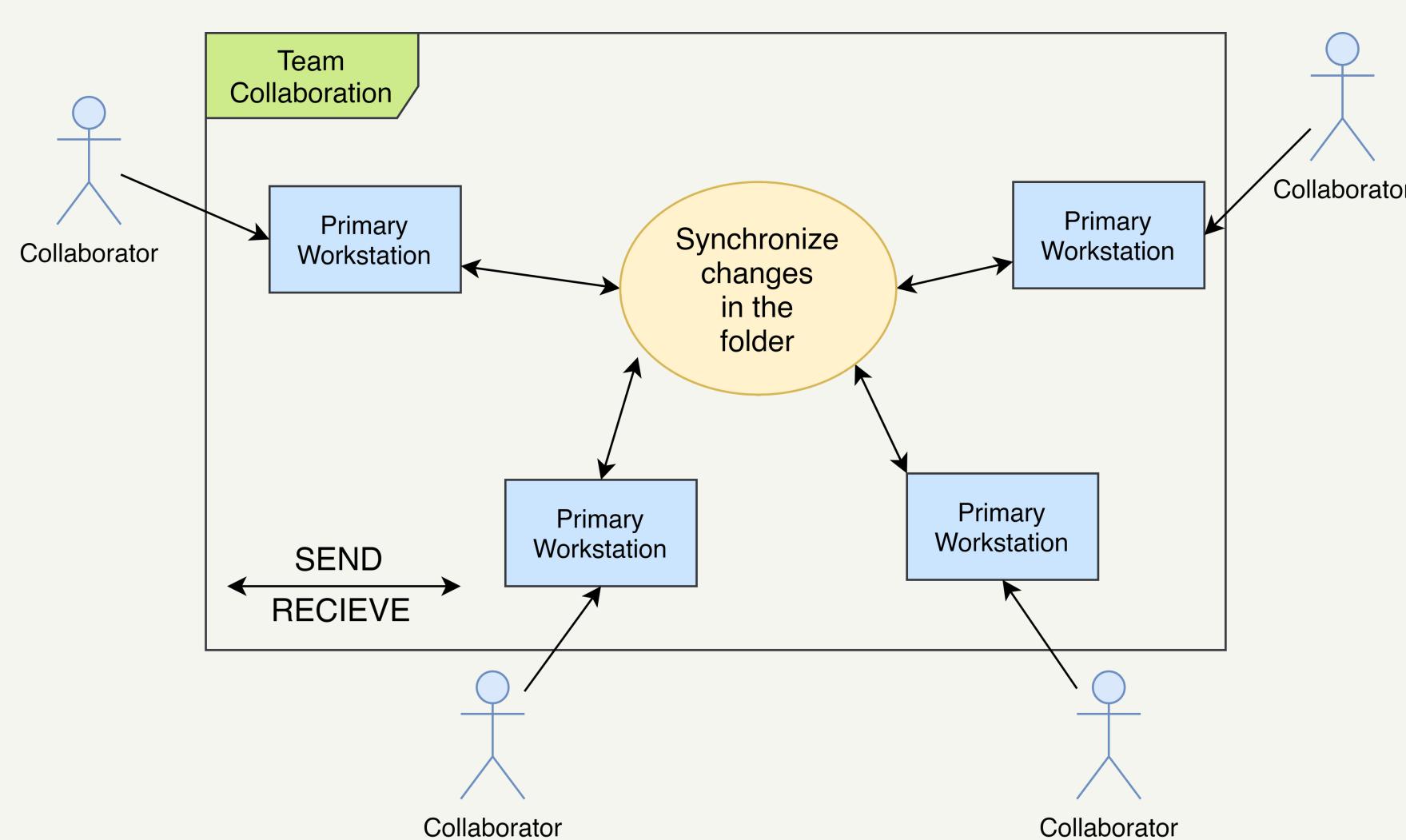
Implementing the
proposed design.
Building the WatchDog
& the Transporter class.

Build 3

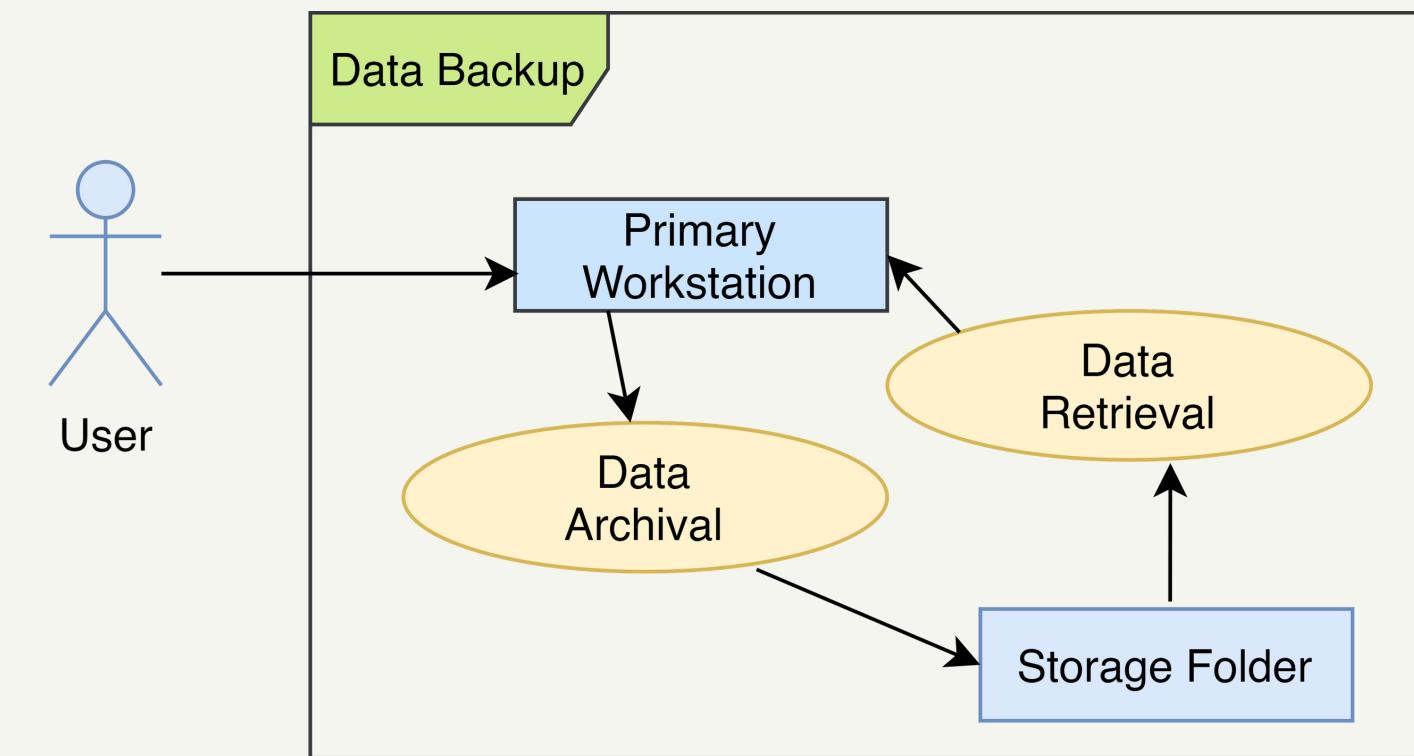
Feature-set addition



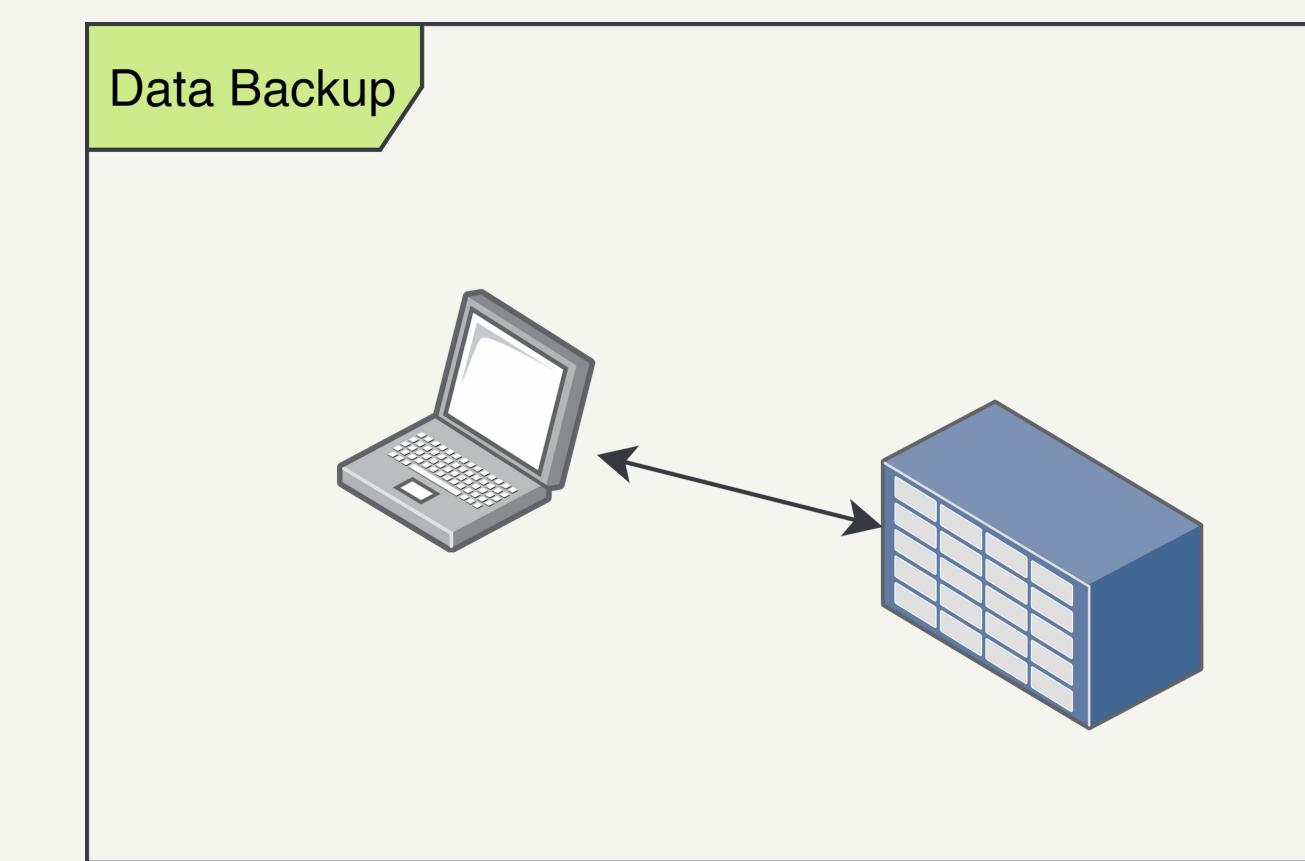
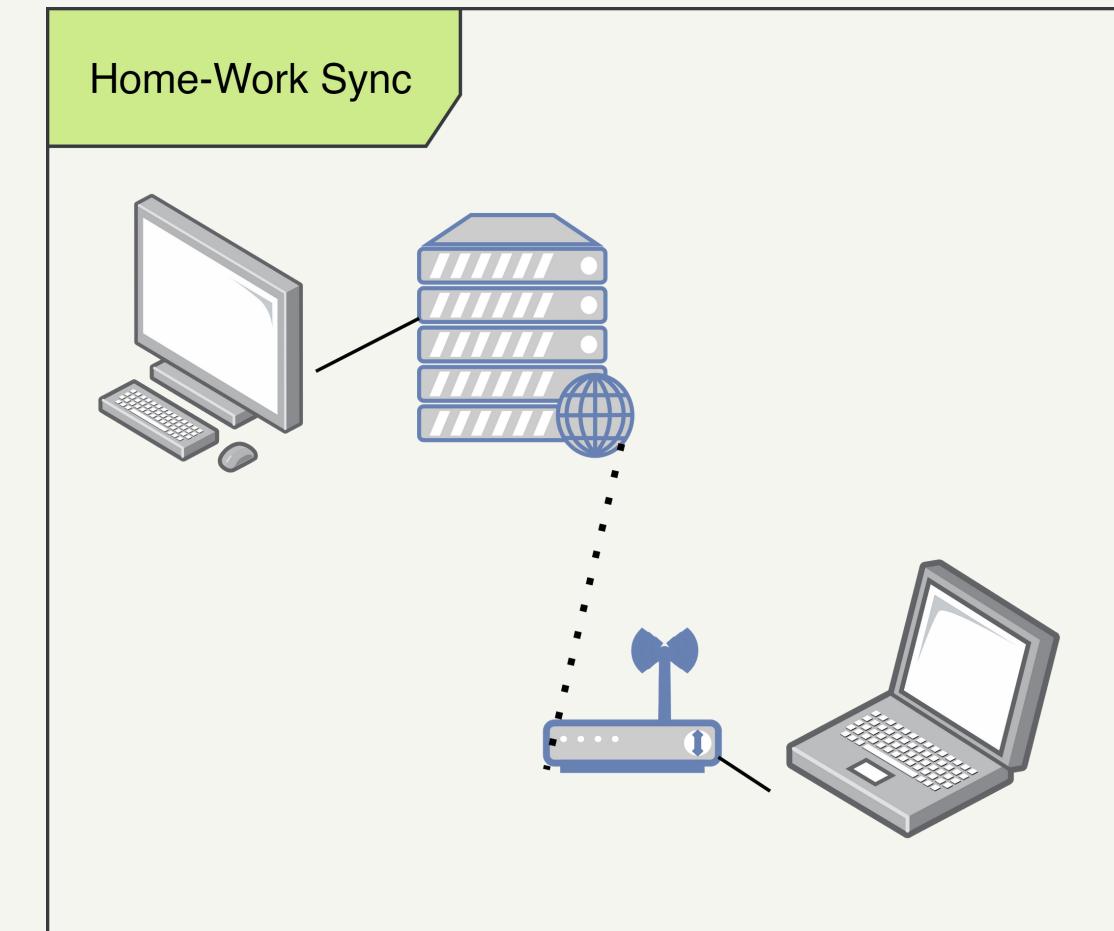
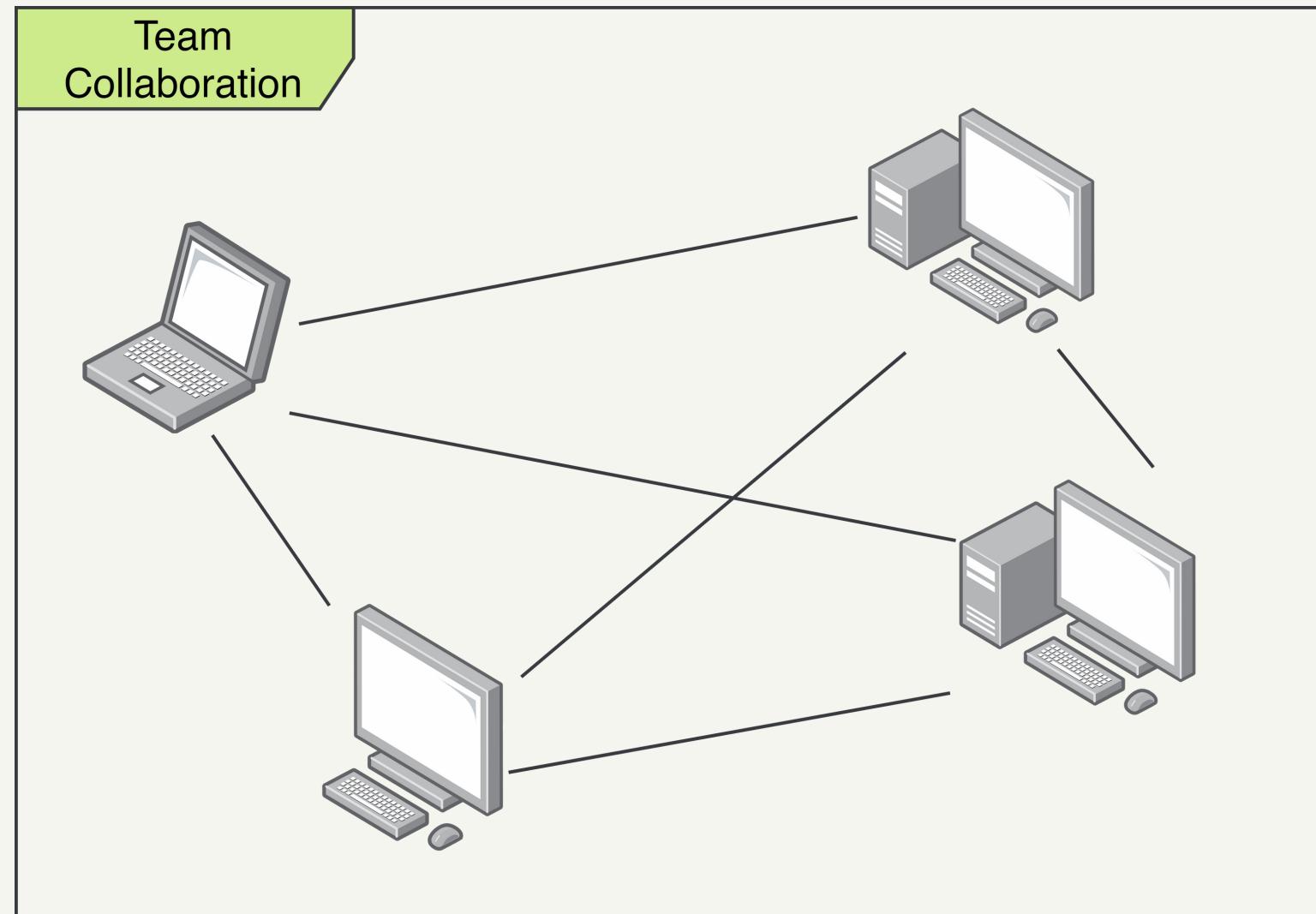
Use Cases



— 09



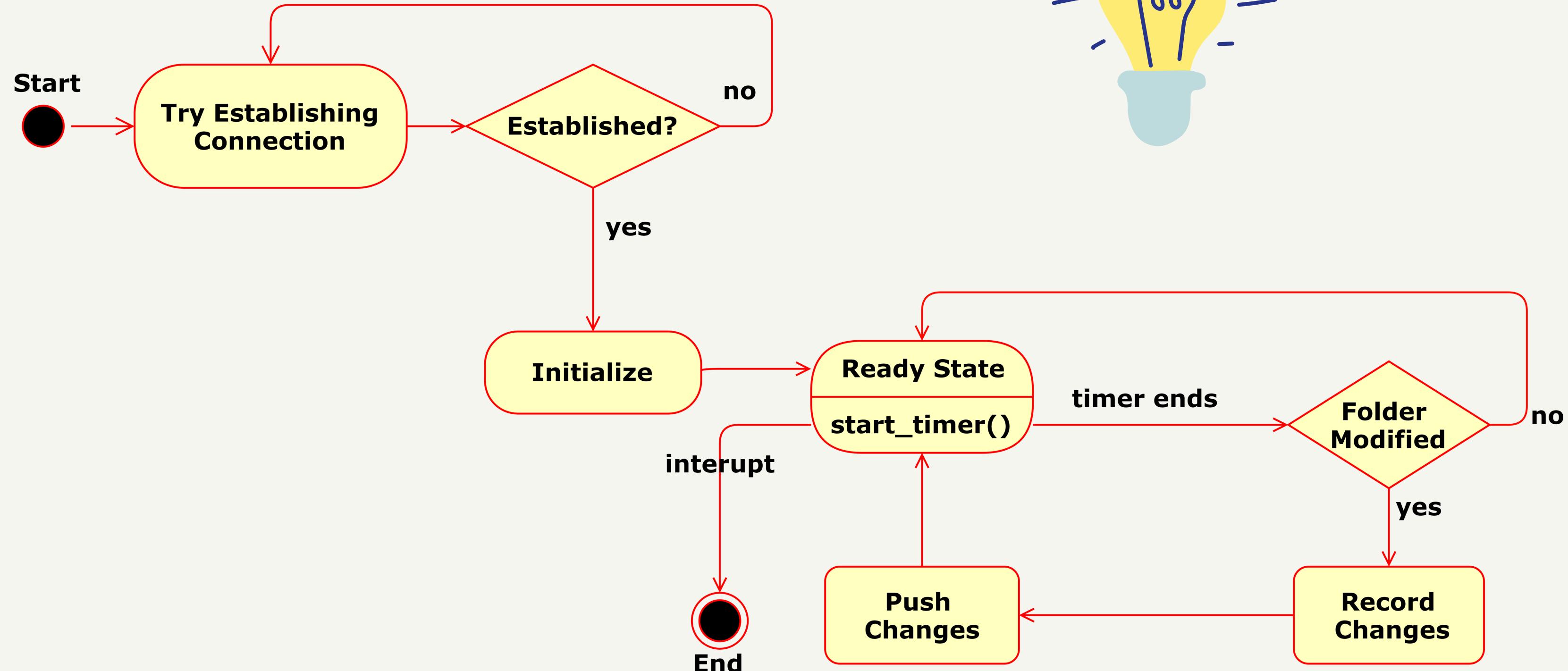
Deployment Scenarios



— 10



Activity Diagram



— 11





Phase-2 Classes

Watcher: A class that keeps track of data inside root directory and notifies the File Handler if data is modified

File Handler: Stores what and how the folder data is changed —¹² and saves the latest hash of all the files

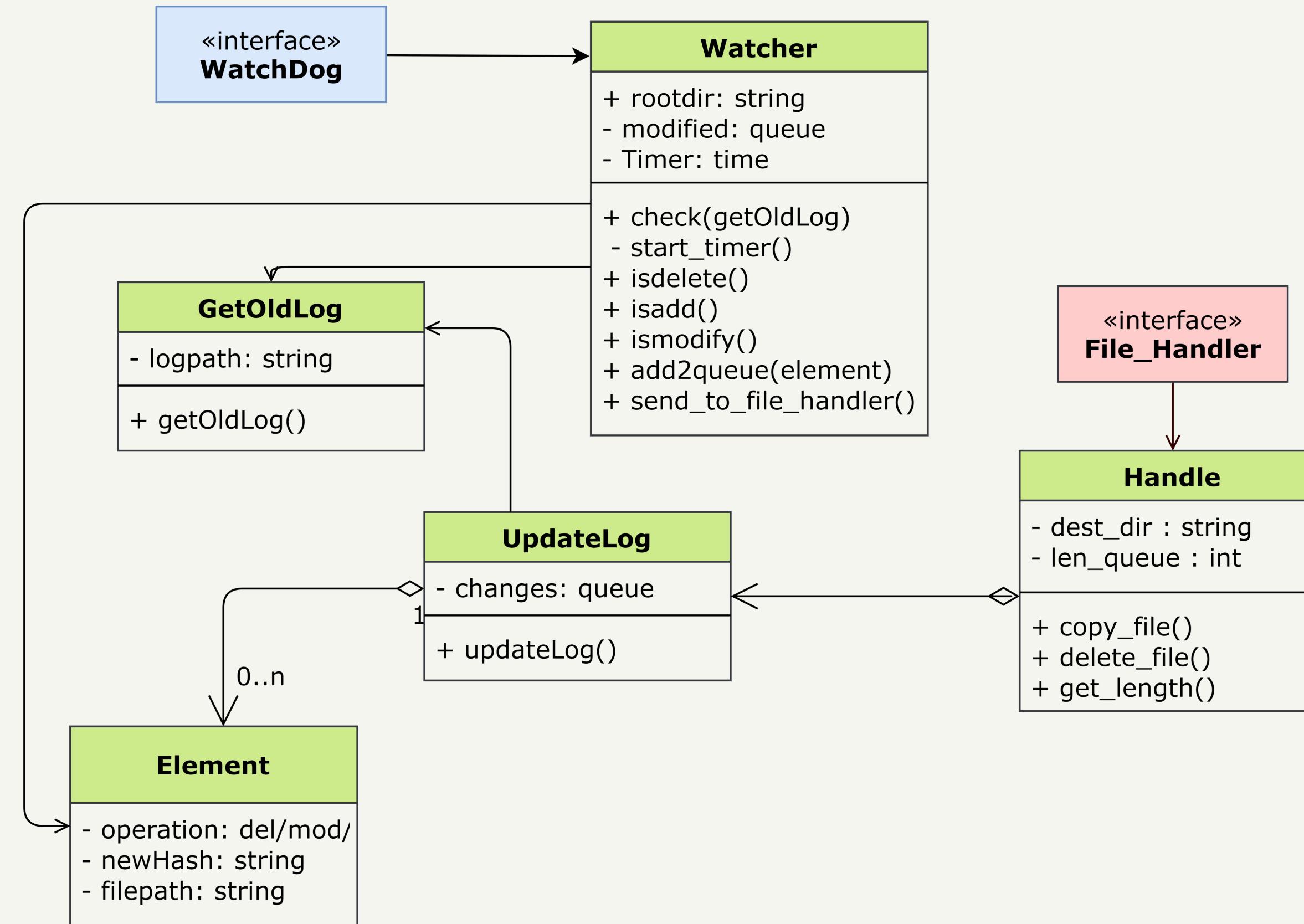
Simple P2P: Handles file transfer between clients



Phase-2 Class Diagrams (Synchronizer)



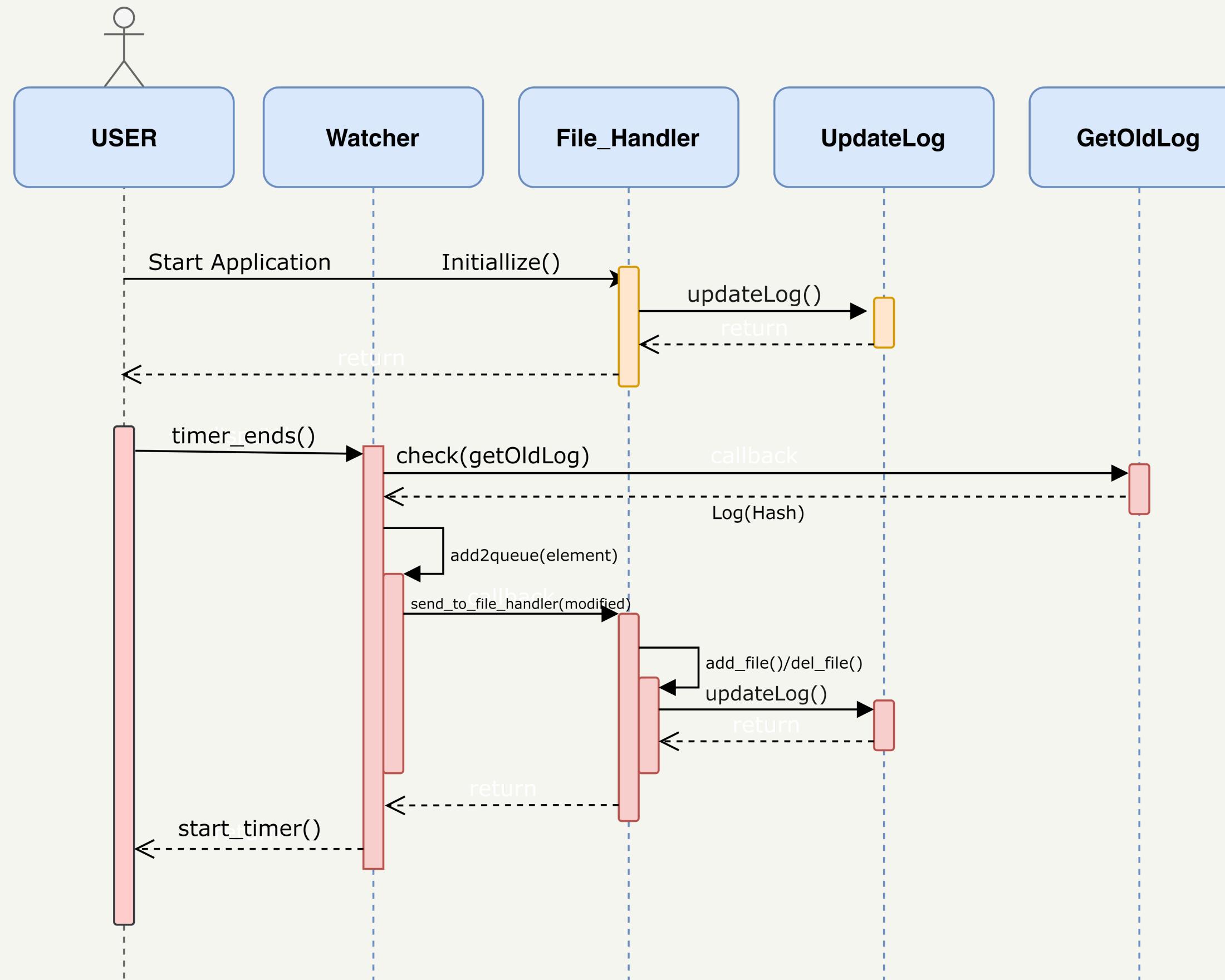
— 13



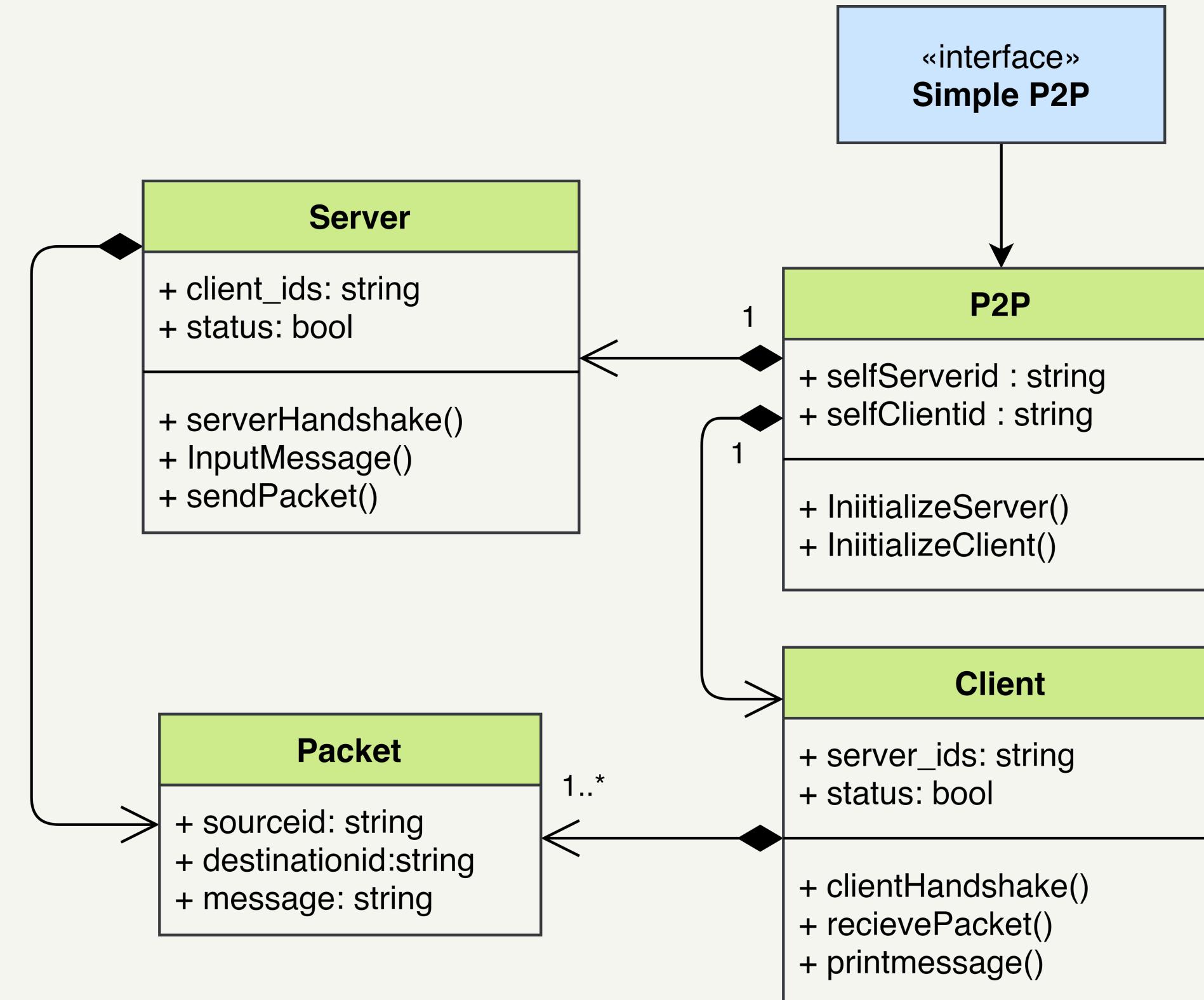
Phase-2 Sequence Diagram (Synchronizer)



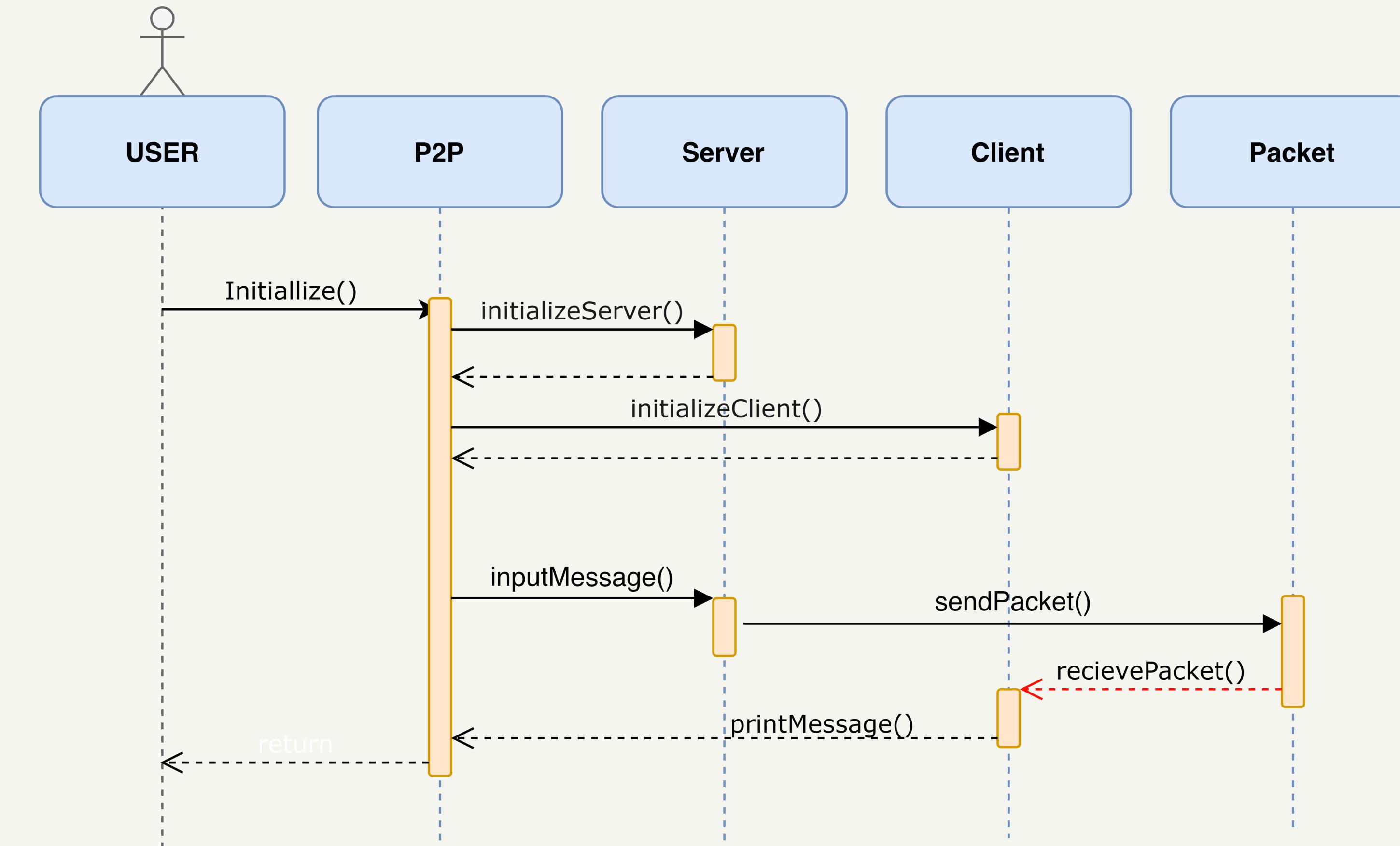
— 14



Phase-2 Class Diagrams (P2P)



Phase-2 Sequence Diagram (P2P)





Phase-3 Classes

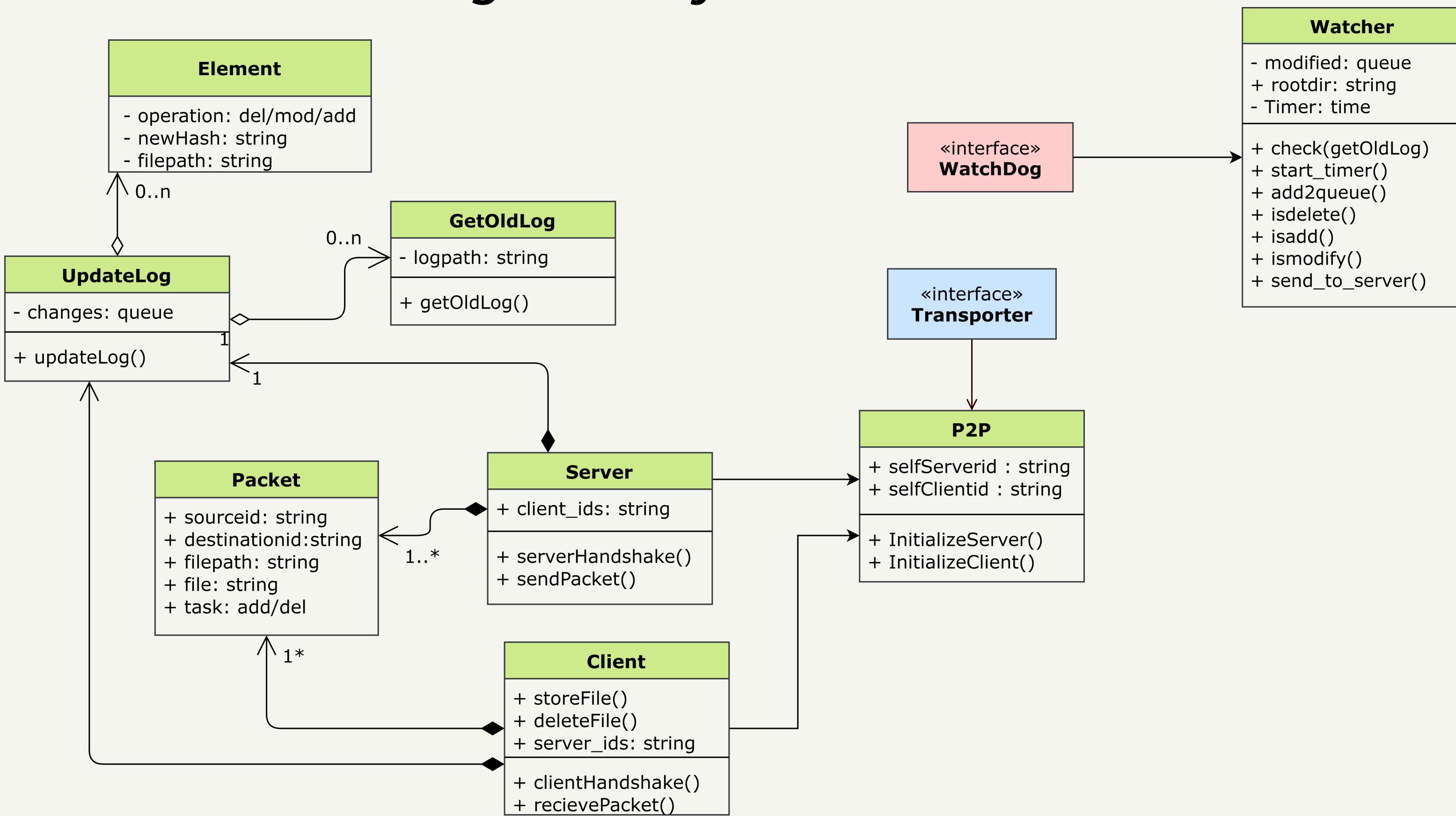
WatchDog - A class that keeps track of data inside root directory and notifies the transporter if data is modified

— 17

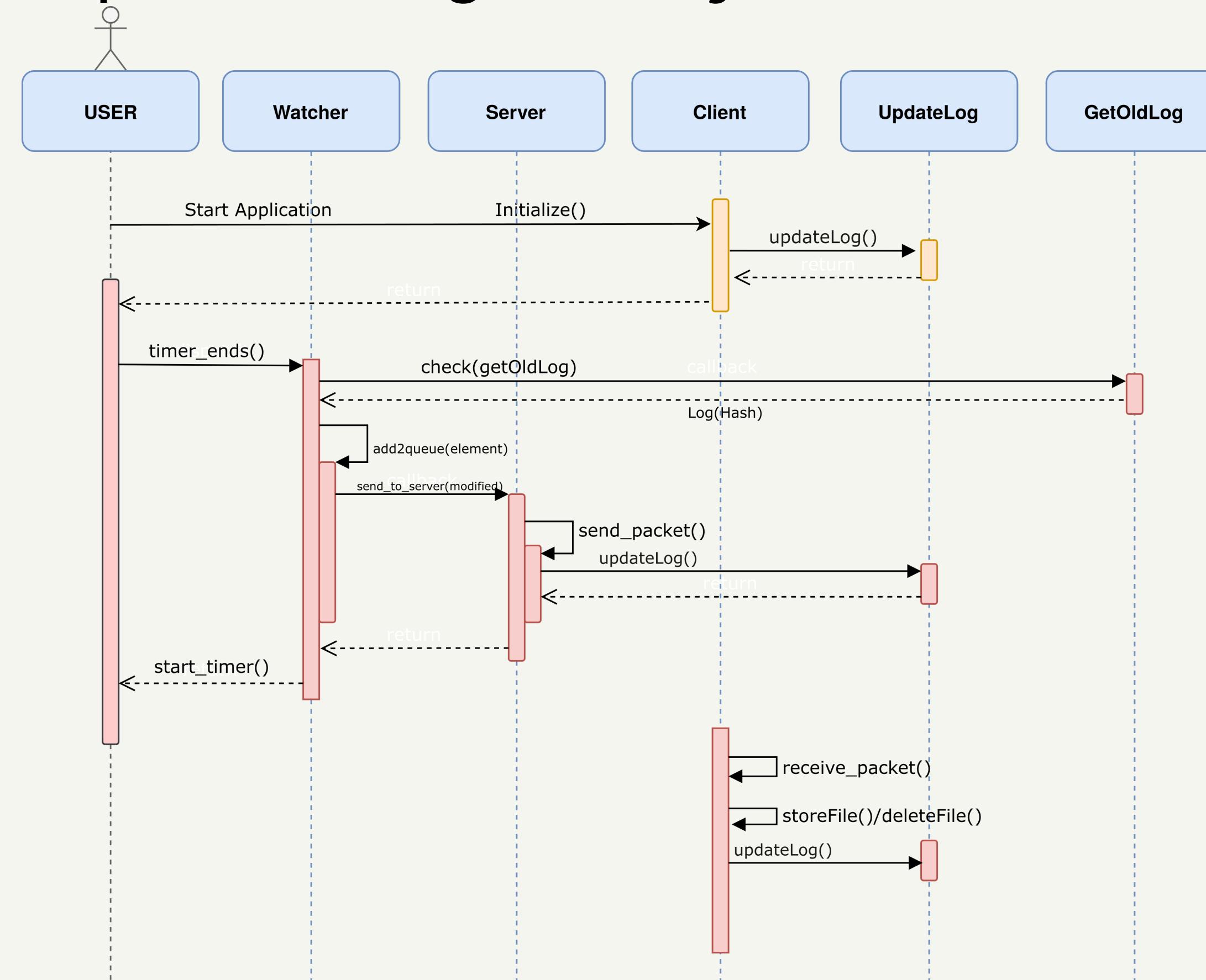
Transporter - A class that manages the server and client threads for Peer2Peer communication. It delivers the modified files and maintains the logs for next iteration.



Phase-3 Class Diagram (Synchronizer)

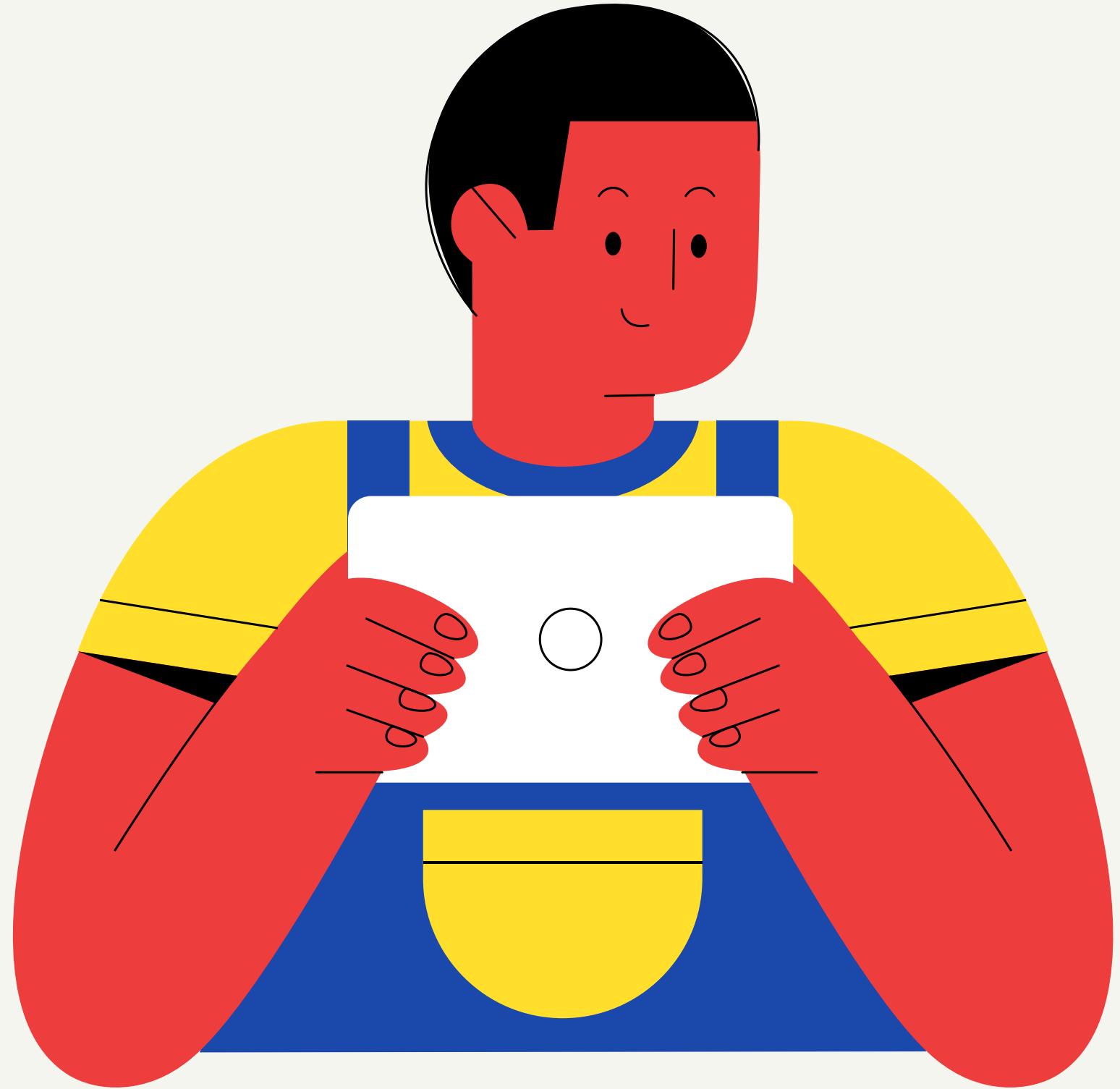


Phase-3 Sequence Diagram (Synchronizer)





Thankyou



— 20

