Table of Contents

[Solution overview 1](#_Toc498717756)

[Server Part 1](#_Toc498717757)

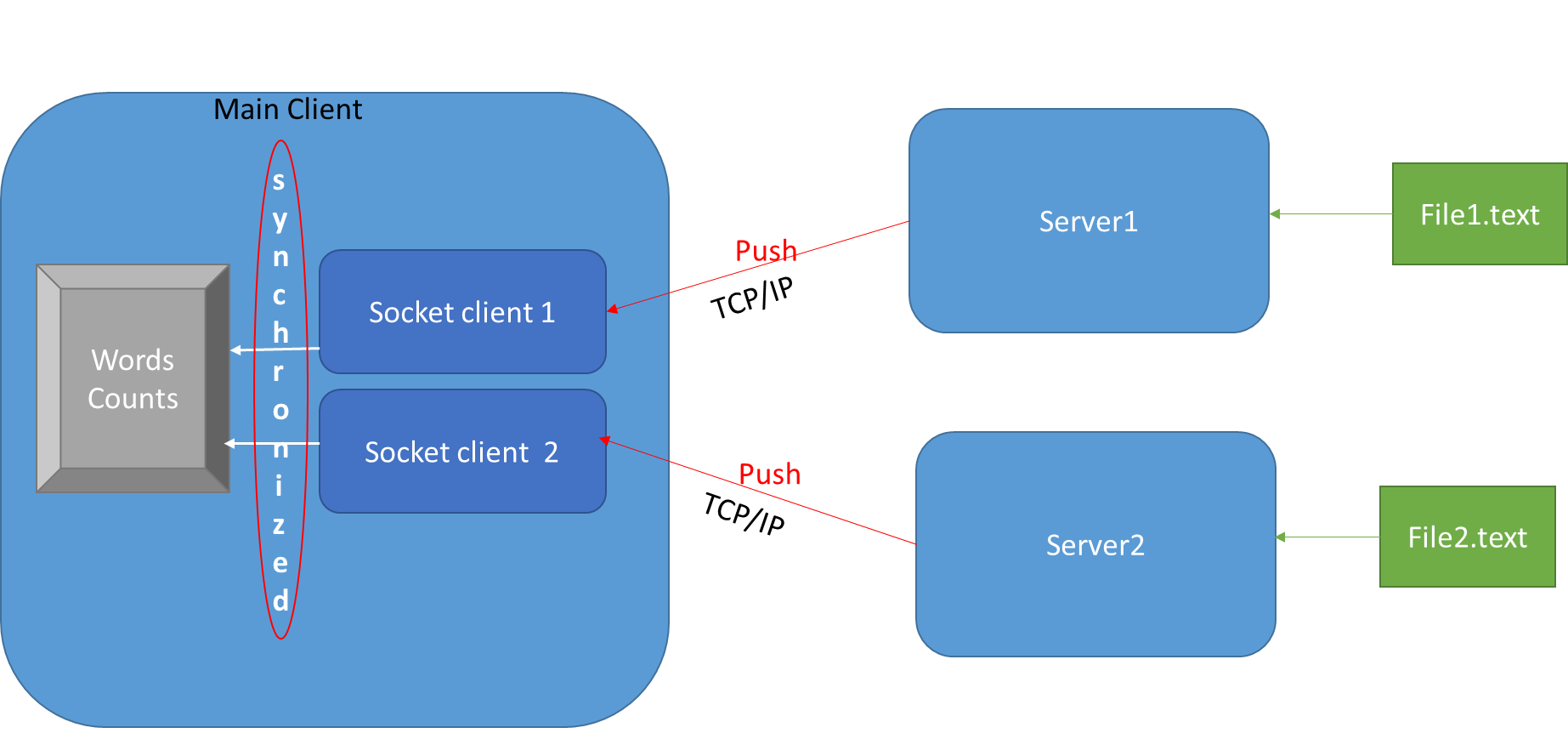
[Client Part 1](#_Toc498717758)

[How to run and Test the solution 2](#_Toc498717759)

[Implementation Idea: 3](#_Toc498717760)

[Future enhancement 3](#_Toc498717761)

# Solution overview



## Server Part

* Server 1 and Server 2 will be responsible to fetch data from files then push to corresponding connected client.

## Client Part

* Will have 2 Socket client using one shared HashMap .
* Will lunch 2 threads to receive data in parallel.
* Will processing incoming line by splitting to words delimited “ “ space .
* Check if the word already exist on shared map will increase count for this word.
* If not exist will add new one with count 1.
* Keep reading data as long as threads still running.
* Once threads terminated start to sort the HashMap and filter top 5.
* All operation over shared Resource are synchronized.

How to run and Test the solution:

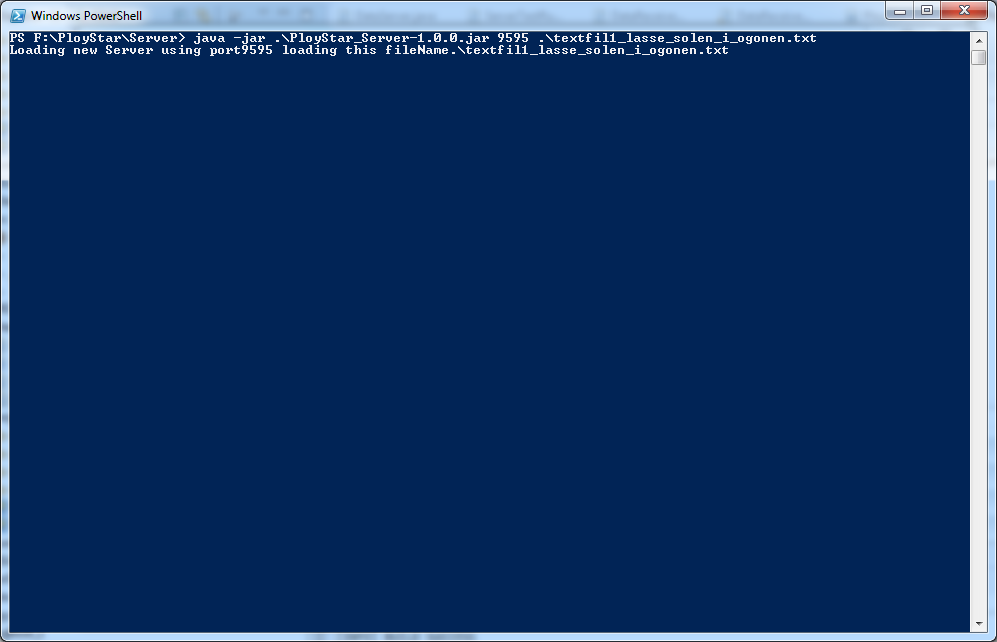
Note: you can find the

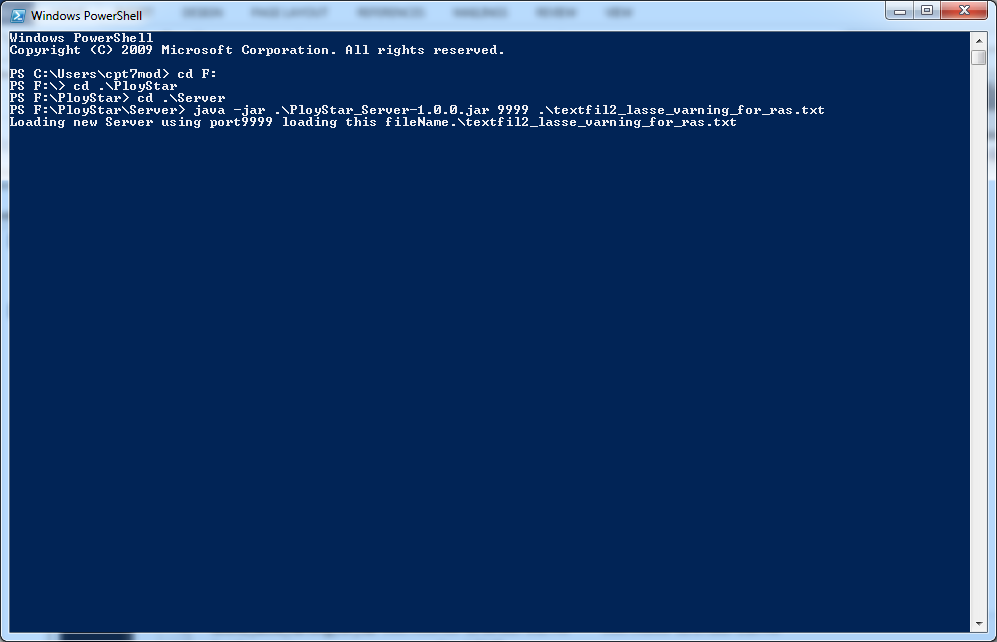
This program is built based on the requirements based on Doc. Nr OSIX-041214-T1

Server Side:

* You can start the Server Side Application by doing the following command
  + Java –jar PloyStar\_Server-0.0.1-SNAPSHOT.jar port\_Number file\_path

Example java -jar .\PloyStar\_Server-1.0.0.jar 9595 ..\files\textfil1\_lasse\_solen



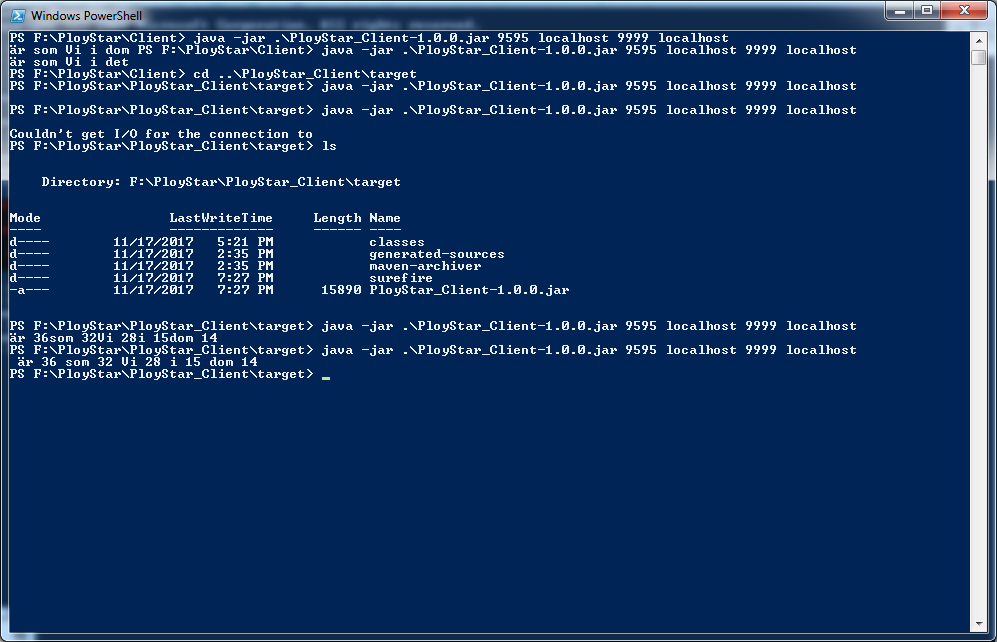


PolyStar Programming Java

This program is built based on the requirements based on Doc. Nr OSIX-041214-T1

Client Side:

* You can start the Client Side Application by doing the following command
  + Java –jar PloyStar\_Client-1.0.0.jar port\_Number hostname port\_number hostname
  + F:\PloyStar\Client> java -jar .\PloyStar\_Client-1.0.0.jar 9595 localhost 9999 localhost



# Implementation Idea

The solution could be used to collect information from NE then (aggregate, correlate or validate) the incoming data (ex: collect voice calls CDRs from different MSCs and aggregate them then send to Billing).

# Future enhancement

* Build Test cases for Server and Client side.
  + Server
    - Check if file exists
    - Check can able to establish the server
    - Check if reading total number of lines in file
  + Client
    - Check if connecting to correct server.
    - Check if synchronized data well for words counts
    - Check if return accurate result.
* Activate java logging messags.
* Configure the number of socket client would be dynamic.
* Configure Top 5 to be Top N.
* Extra Validation for the input parameters and files.