Account system in the Cloud

- Create your first network application
- Develop an Internet Account system using Java (or Python)
  - Note : You can use the Python if you prefer using it.
    - Create client and server programs using socket
    - Get an (arithmetic) expression from client side, and server solves the expression and sends the result back to the client
    - Utilize thread for supporting multi-user
- Define the protocol (communication message formats) for this application

Outline

- Big picture
  - At first, server has 5 accounts info. (you can make your own case)
  - Client sends some expressions to the server
  - Server interprets the expression, executes the expression, and sends the result back to the client
  - Server supports multi users by thread

- Protocol for Command/Response Interaction
  - You should define ASCII-code based message formats
  - Command and responses formats
  - Refer : HTTP request / response formats

Requirements

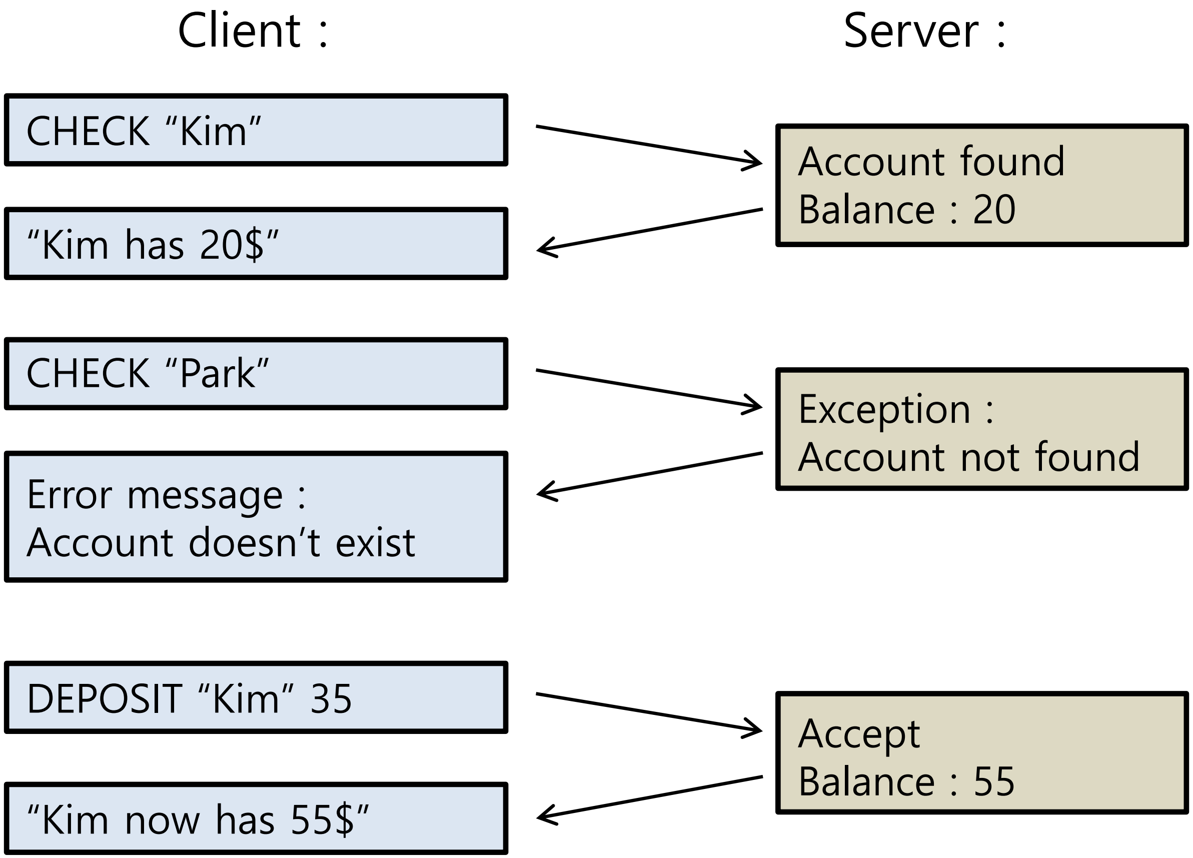
- The four basic operations - check / deposit / withdraw / transfer - should be supported

- Server should support multi users by threads
  - Program should support synchronization for each object

- The response from the server may be either complete message for the expression or error message (and error type)

- Server information at a client is stored with a configuration file (e.g., serverinfo.dat)
  - i.e., client programs read Server IP and port # from the file.

Example scenario



Client :

Server :

CHECK “Kim”

Account found
Balance : 20

“Kim has 20$”

CHECK “Park”

Exception :
Account not found

Error message :
Account doesn’t exist

DEPOSIT “Kim” 35

Accept
Balance : 55

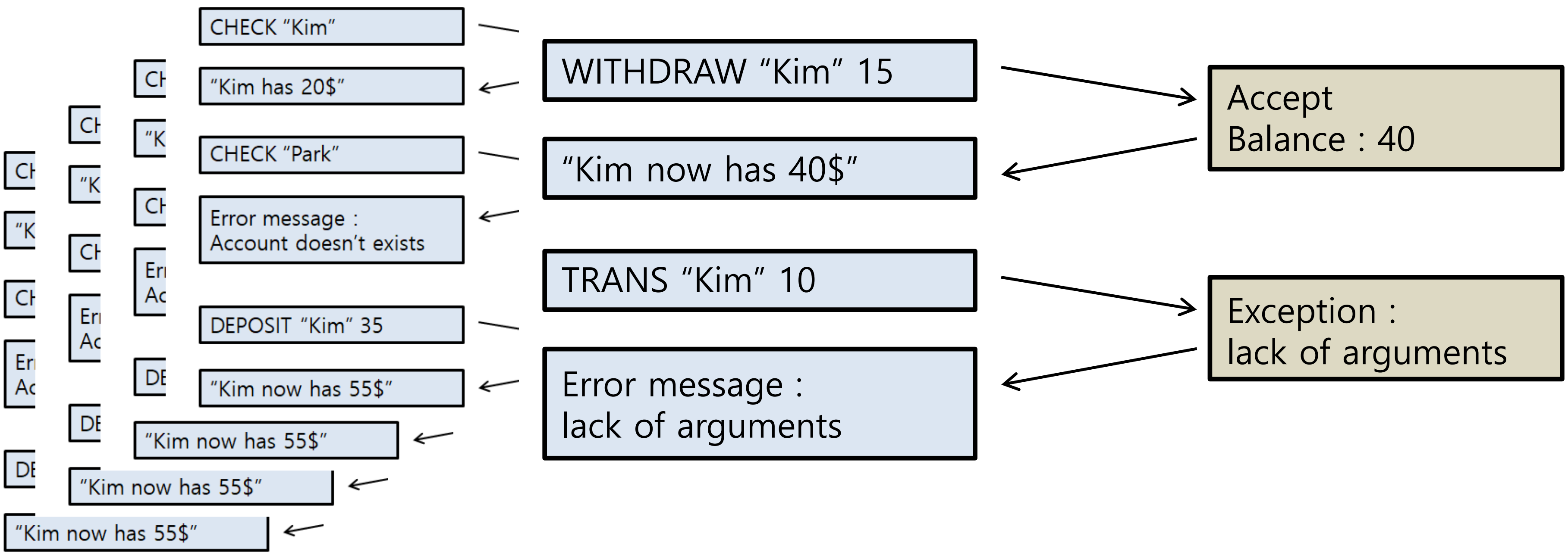
“Kim now has 55$”

Example scenario

Multi-users

Client :

Server :



CHECK “Kim”

“Kim has 20$”

CHECK “Park”

Error message :
Account doesn’t exists

DEPOSIT “Kim” 35

“Kim now has 55$”

“Kim now has 55$”

“Kim now has 55$”

“Kim now has 55$”

WITHDRAW “Kim” 15

“Kim now has 40$”

TRANS “Kim” 10

Error message :
lack of arguments

Accept
Balance : 40

Exception :
lack of arguments

Guidelines

- Submit report (written in English)
  - To describe the basic architecture of your application
  - To explain the message formats (protocol for your application)
  - Include
    - Copy or your source codes at the end of the report
    - Execution screenshots (at least 5 cases)

- Comments in source files

Submission Guideline

- Zipped java project files
  - Export your project (source file + environments files)
  - https://dzone.com/articles/exporting-and-importing (참고 링크)
  - One MS-word report file: Report1_이름.doc (docx)

- Compress the document and project files with a single zip file
  - Name : studentid_이름.zip

Grading

- Code with correct output : 50 points
  - Four basic operations – 25 points
  - Error message – 10 points
  - Multi user support using thread & Synchronization – 15 points

- Report : 30 points
  - System architecture – 10 points
  - Protocol definition – 10 points
  - Copy of your source code & execution screenshots – 10 points

- Comments : 20 points
  - At lease 30% of your source code to explain how it works in detail
  - program codes with comments so that others can understand your code