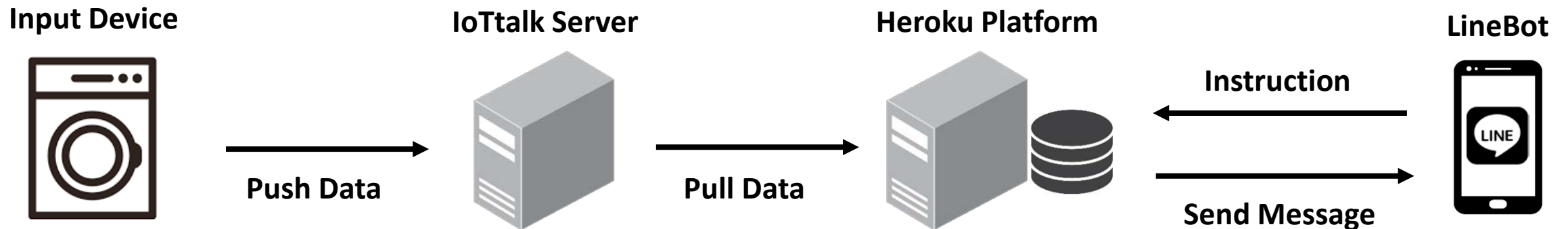# Mini Project 4 - Goal

- Design an IoT-based **LineBot** Application with IoTtalk server

  - by deploying the application to the **Heroku** cloud platform through **Git** method

- Check your Heroku and LineBot can work properly before starting the mini project 4

Input Device   **IoTtalk Server**   **Heroku Platform**   **LineBot**

Push Data   **Pull Data**   **Instruction**

Send Message

Mini Project 4 - Package

- **Procfile**

  Heroku apps include a Procfile that specifies the commands that are executed by the app on startup.

  Procfile Format >> web: [language] [file to be uploaded]

  - E.g. web: python app.py

- **requirements.txt**

  List all the packages we could use, and Heroku will install these based on the document.

- **csmapi.py/DAN.py**

  Some useful APIs about IoTtalk connection

- **app.py**

  You can follow the comment and modify this file to finish mini project 4.

  The function handle_message() is used to control the message reply.

Spec

- Design the **scenario**

  - E.g. Query the status of machines, the information of something, the location of something …

- IoTtalk

  - Design your own **Device Model** (at least one)

  - Implement the **Join Function**

- LineBot & DAI

  - Your LineBot need to interact with IoTtalk(send/receive data to/from IoTtalk)

  - You cam modify '**app.py**' to finish mini project 4

- Note

  - "csmapi.py/DAN.py" are the API about IoTtalk connection, you may use it in 'app.py'. Don't need to modify them.

Grading Policy - 1

- LineBot & DAI (40%)

  - (5%) IoTtalk Devices Registration

  - (15%) Push data to IoTtalk(IDF)

    - You can random some data to push to IoTtalk, or get data from other

  - (15%) Pull data from IoTtalk(ODF)

  - (5%) Get result from LineBot App successfully

Grading Policy - 2

- Report (50%)

  - (5%) Scenario Description

  - (20%) How to Design

  - (15%) Screenshots

    - IoTtalk GUI connection, DM/DF creation, join connection

    - IDF/ODF Monitor

    - LineBot result

  - (10%) What you learn

- Demo (10%)

Submission

- Please upload your mini project 4 to eLearn: **Deadline: 2021-01-14(Thu.) 23:59**

- Your project must be named as follows:

  - Program

    - <Student_ID>_project4.zip
      - csmapi.py
      - DAN.py
      - Procfile
      - requirements.txt
      - <Student_ID>_project4.py (This file is modified from '**app.py**')

  - Report

    - The report filename must be "<Student_ID>_project4.pdf"

- Plagiarism Avoidance: Discussion is encouraged. However, plagiarism is not allowed.

  We will use, e.g., "Moss" for similarity comparison and 0 points will be given if plagiarism.

IoTtalk Server

- The IoTtalk project should be named as "**Your_Student_ID**"

- **IoTtalk server**

  - **Student ID: 106062131~109062512**

    - http://140.114.77.89:9999/

  - **Student ID: 106062547~109062650**

    - http://140.114.77.90:9999/

  - **Student ID: 109062665~109064511、R124683364**

    - http://140.114.77.91:9999/

Note

- Please be careful that not to delete the device information of other people

- It is recommended to back up the information you set on IoTtalk

  - E.g. join function setting, device setting…