

物聯網核心網路技術一期末書面報告

第二組

F74082109 梁華軒

F74082010 陳昭穎

Q56114047 簡羽晨

一、情境

本次專案中以霧運算為基礎架構，模擬了農作物之蟲害防治的情境。由於農地面積很大，且蟲害事件發生時並不會使每個區域都產生損害，受損的程度也都不盡相同。因此本專案將農地分割成不同區塊進行管理，並期望能依照不同區域所遭遇的情況動態地進行調整。

而在蟲害防治的情境中會有兩項主要任務：

1. 感測蟲害事件，並啟動防範裝置。
2. 蒐集農作物損害的資訊，並調整防範裝置之強度。

由於啟動防範裝置對於即時性的要求較高，因此會交由較靠近網路邊緣的霧運算判斷是否應該啟動防範裝置；而調整防範裝置的強度所需要的運算能力較強，且對即時性的要求較低，因此會將資料送往雲端進行處理。

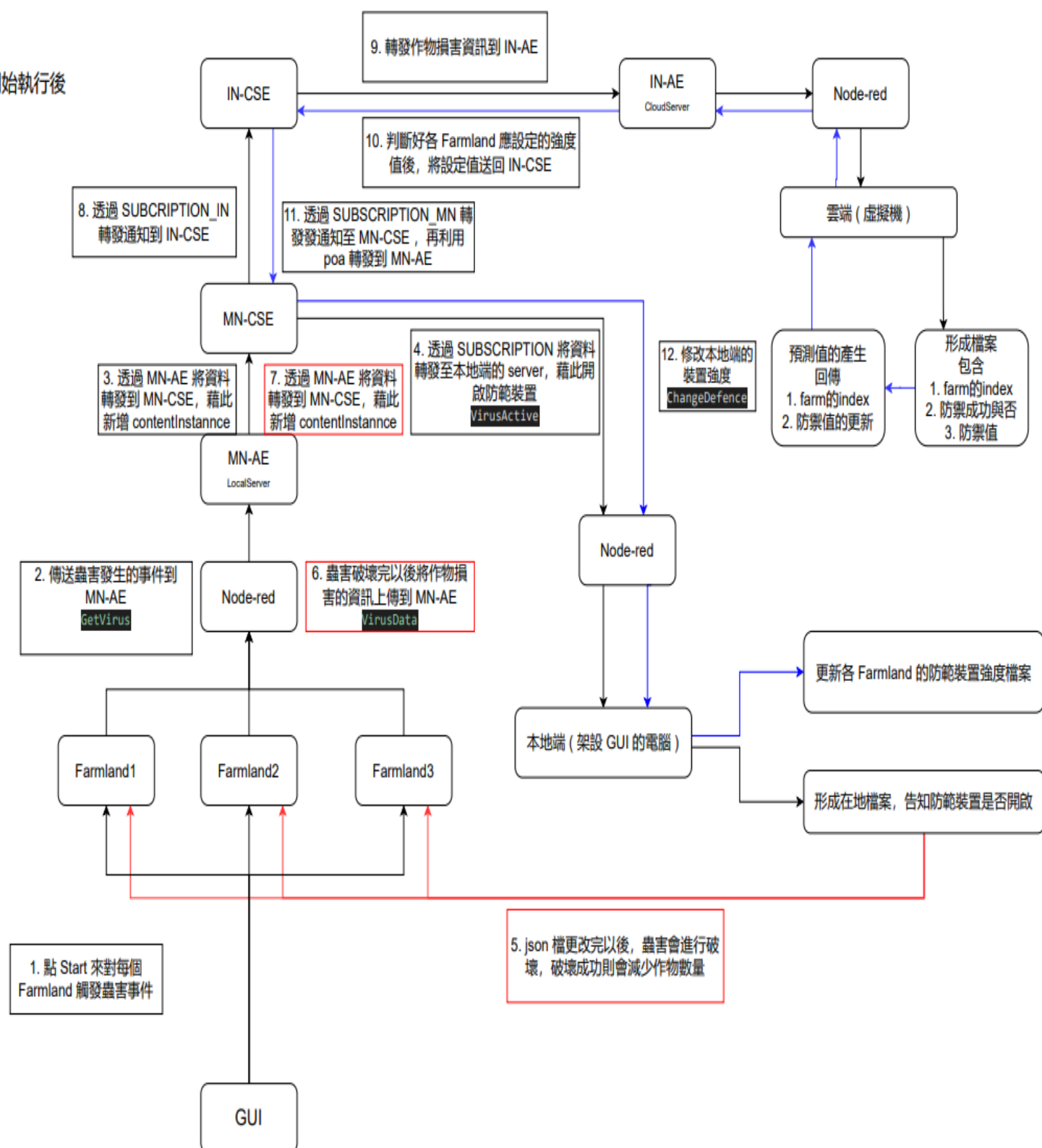
二、環境需求

需安裝以下套件：

- OM2M
- Node-RED
- Python3.7, PyQt5, requests套件

三、系統架構

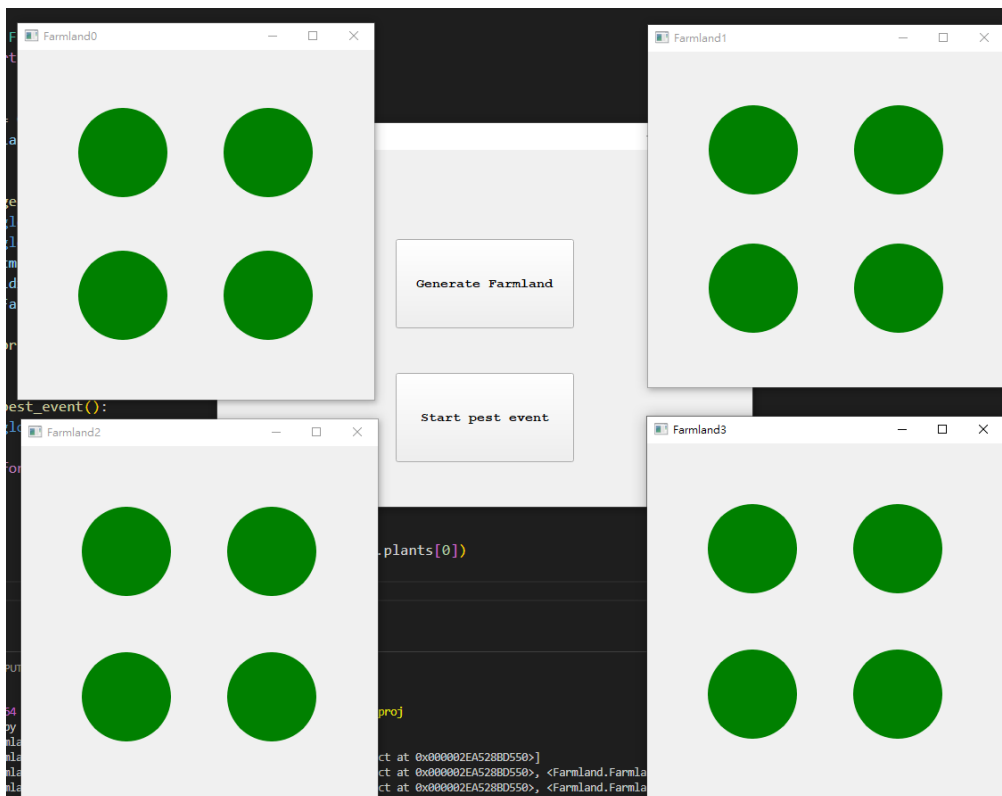
開始執行後



四、使用者操作流程

1. 前置作業

- 於虛擬機上
 - a. 開啟 MN-CSE、IN-CSE、Node-RED
 - b. Deploy Node-RED 的同時, 會架設好所需的 http servers
 - c. 透過 Node-RED 於 MN-CSE 與 IN-CSE 建立所需的 Application 與 Container與訂閱功能
- 於本機上
 - a. 開啟Node-RED
 - b. 執行 main.py , 並藉由 Generate Farmland 按鈕產生農地視窗(數量可自行決定)
 - c. 各個農地生成自己的防禦初始值



2. 開始模擬

1. 點擊 Start pest event 按鈕產生蟲害事件(使用者操作)

- 本機

- i. 向Mn-CSE傳送偵測到蟲害的發生

Attribute	Value						
m	cin_727804741						
ty	4						
ri	/mn-cse/cin-727804741						
pi	/mn-cse/cnt-352681816						
ct	20221227T032854						
lt	20221227T032854						
st	0						
cnf	message						
cs	103						
con	<table> <tr> <th>Attribute</th><th>Value</th></tr> <tr> <td>time</td><td>27/12/2022 19:28:54</td></tr> <tr> <td>MaxNumberOfFarm</td><td>3</td></tr> </table>	Attribute	Value	time	27/12/2022 19:28:54	MaxNumberOfFarm	3
Attribute	Value						
time	27/12/2022 19:28:54						
MaxNumberOfFarm	3						

- 虛擬機

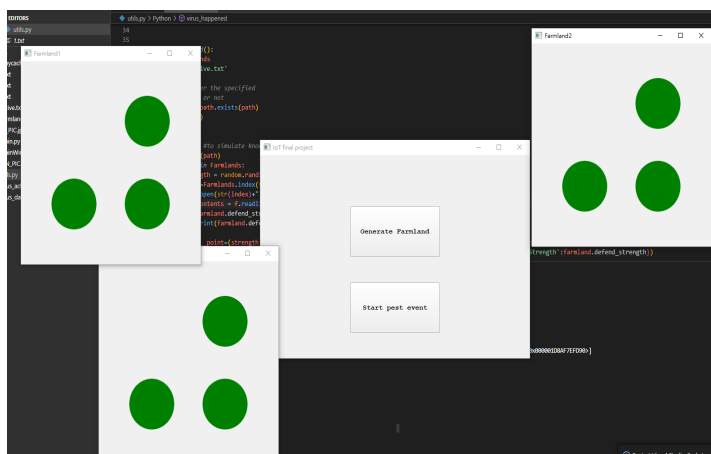
- i. Mn-CSE接收到蟲害發生後，紀錄時間以及農場總數生成新的instance，立刻回傳至本機啟動防禦

2. 判斷防禦機制是否在時間內開啟(自動操作)

- 本機

- i. 生成隨機的蟲害強度

- ii. 判斷是否防禦值(如防禦機制未開啟，值為0)大於蟲害值，呈現在GUI



- iii. 向Mn-CSE傳送防禦的結果(傳送內容{index:農地編號, flag:防禦成功與否, strength:當前防禦值})

Attribute	Value								
m	cin_895847667								
ty	4								
ri	/mn-cse/cin-895847667								
pi	/mn-cse/cnt-174622615								
ct	20221227T032855								
lt	20221227T032855								
st	0								
cnf	message								
cs	124								
con	<table><tr><th>Attribute</th><th>Value</th></tr><tr><td>index</td><td>1</td></tr><tr><td>DefenceCheck</td><td>False</td></tr><tr><td>dfStrength</td><td>40</td></tr></table>	Attribute	Value	index	1	DefenceCheck	False	dfStrength	40
Attribute	Value								
index	1								
DefenceCheck	False								
dfStrength	40								

○ 虛擬機

- Mn-CSE接收到防禦的結果後，生成新的instance，且因為有訂閱的功能，會將生成檔案的資料傳給In-CSE
- In-CSE接收到防禦的結果後，藉由設定的POA傳送給Node-RED，將資料存在虛擬機上

3. (optional) 改變防禦值(使用者操作)

○ 虛擬機上

- 讀取先前所儲存下來的檔案進行分析(在此專題中成功的防禦的防禦值折半，失敗的防禦值加10{index:農場編號, strength:更新後的方玉值})，

向In-CSE傳送更新

Attribute	Value						
m	cin_113205871						
ty	4						
ri	/in-cse/cin-113205871						
pi	/in-cse/cnt-91041734						
ct	20221219T191155						
lt	20221219T191155						
st	0						
cnf	message						
cs	83						
con	<table><tr><th>Attribute</th><th>Value</th></tr><tr><td>index</td><td>0</td></tr><tr><td>df_strength</td><td>50</td></tr></table>	Attribute	Value	index	0	df_strength	50
Attribute	Value						
index	0						
df_strength	50						

- ii. In-CSE形成新的instance, 且因為有訂閱的功能, 會將生成檔案的資料傳給Mn-CSE
- iii. Mn-CSE接收到後, 藉由設定的POA傳送給Node-RED(本地端的)
- 本機
 - i. Node-RED更改各個農地的防禦值