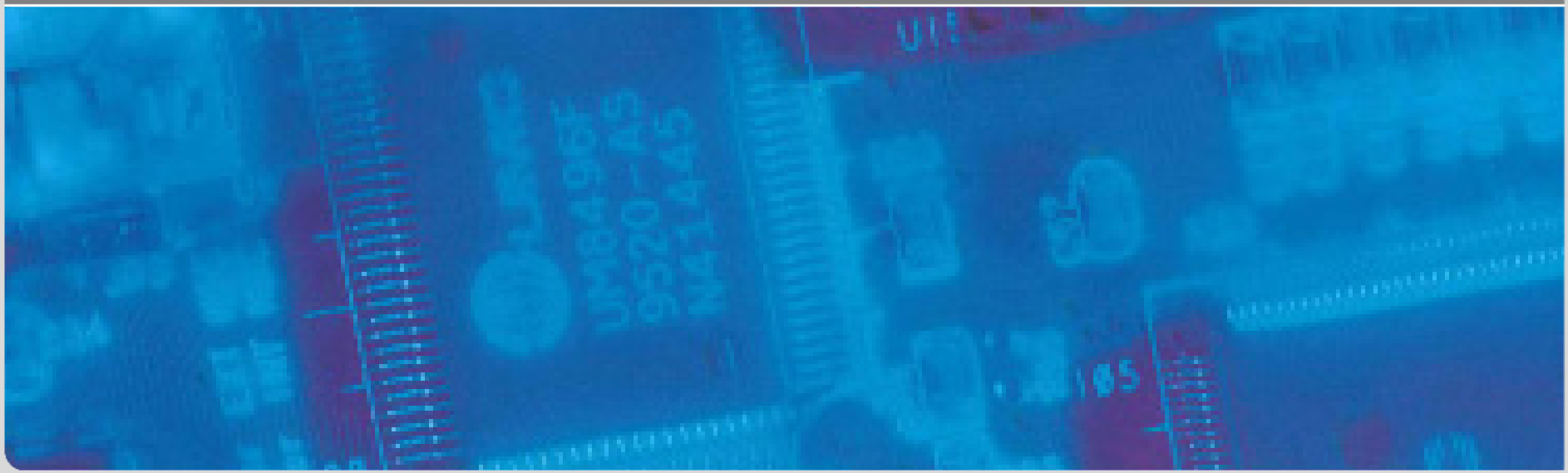
Karlsruhe Institute of Technology

Design and Implementation of an IoT-based Control Device for an off-the-shelf Coffee Machine

Karim Ben Ammar

CES – Chair for Embedded Systems



Outline

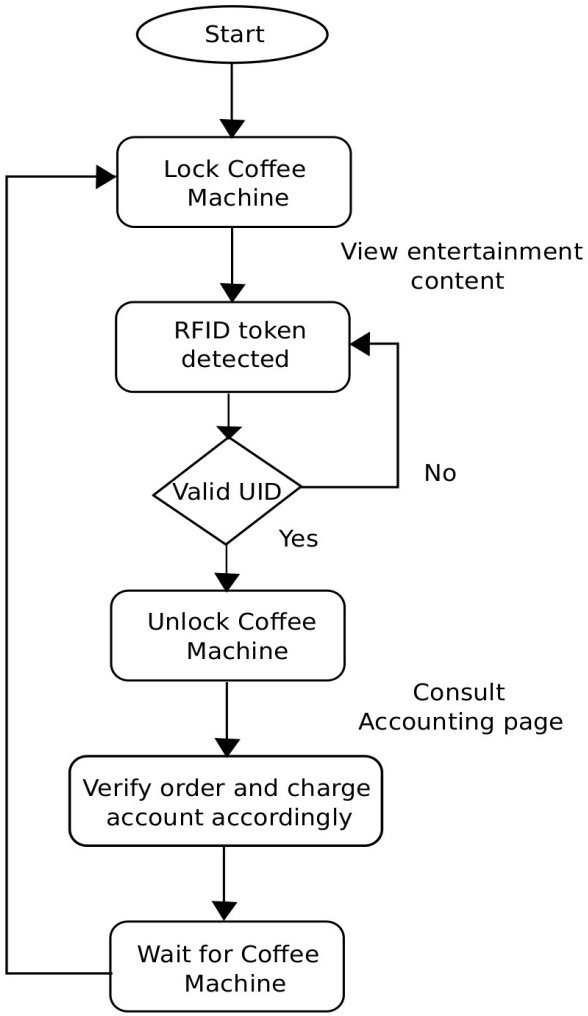
- Motivation
- Design/Implementation
- Challenges
- Conclusion

Motivation

- Limit usage to authorized users
- Prevent money loss due to miscounting the tally sheets or users forgetting to write down their transaction, etc
- The coffee machine may present an unusual behaviour

- Solution: Automated Control System
  - Allow access only to authorized users
  - Automatically detect the user's choice and charge him accordingly
  - Keep track of the coffee machine's behaviour

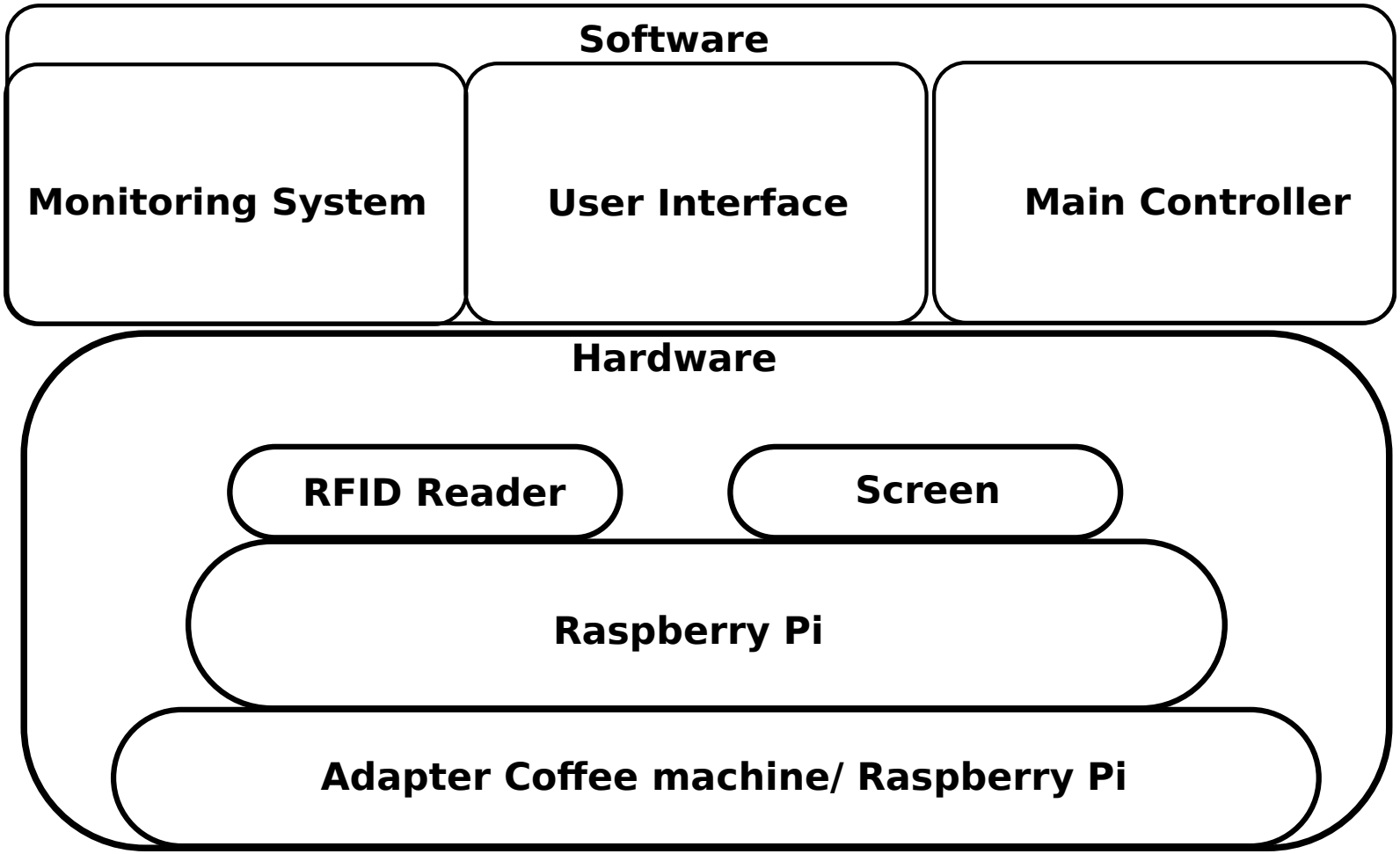
Motivation : Workflow



Design/Implementation

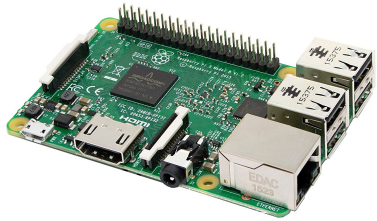
- The system should :
  - Lock the coffee machine until a valid RFID token is provided
  - Automatically detect the state of the coffee machine
  - Charge the user independently of the server's state
  - Continuously monitor the coffee machine's behaviour and log it for future analysis

Design/ Implementation

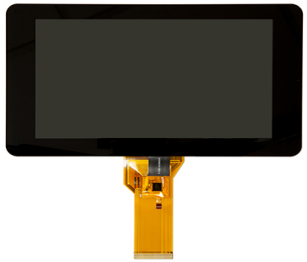


Hardware

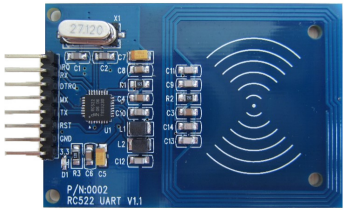
- Raspberry Pi



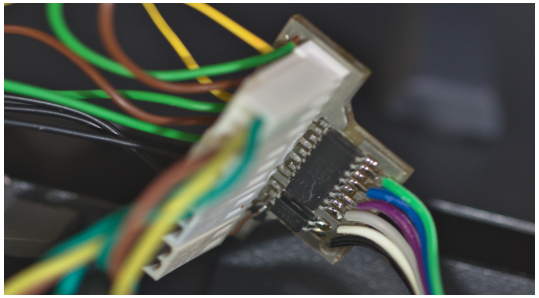
- Official 7" touch screen



- Joy-IT RFID Module

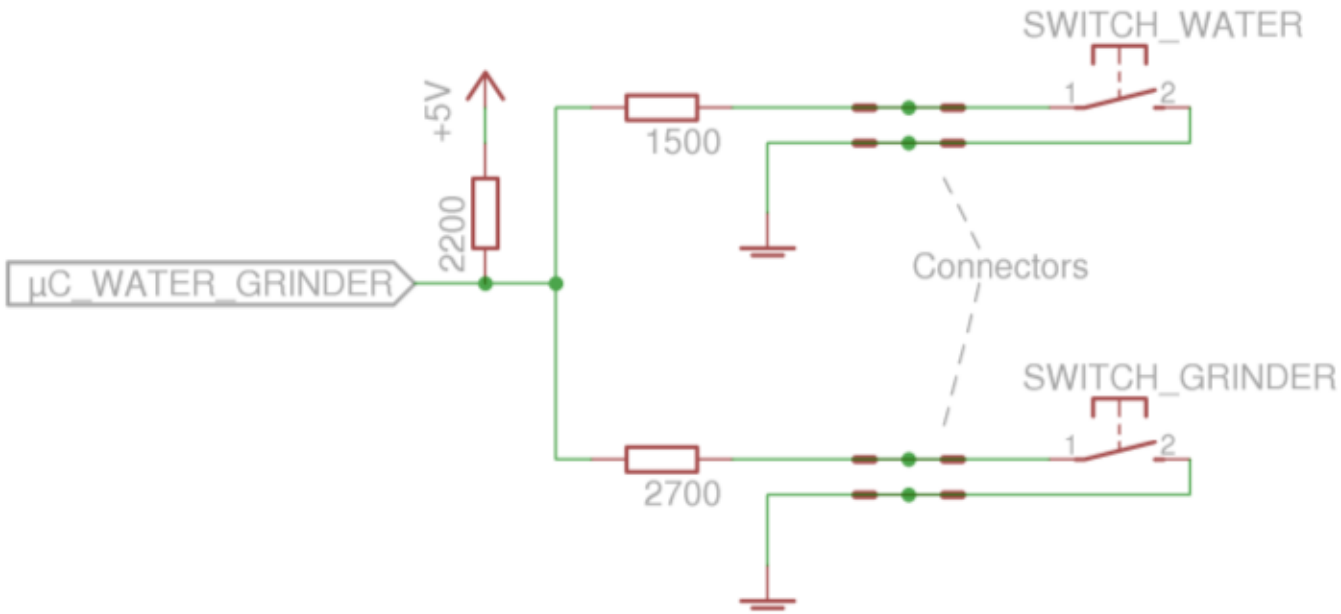


- Custom PCB



Adapter Coffee machine/ Raspberry Pi

- Capture all the sensor's output
- Simulate the water level sensor (Lock/Unlock mechanism)



Tasks of the Raspberry Pi

- Continuously listen to the RFID Reader
- Update the entertainement content
- Manage every transaction
- Log the behaviour of the coffee machine
- Listen for incoming HTTP requests from the Android app

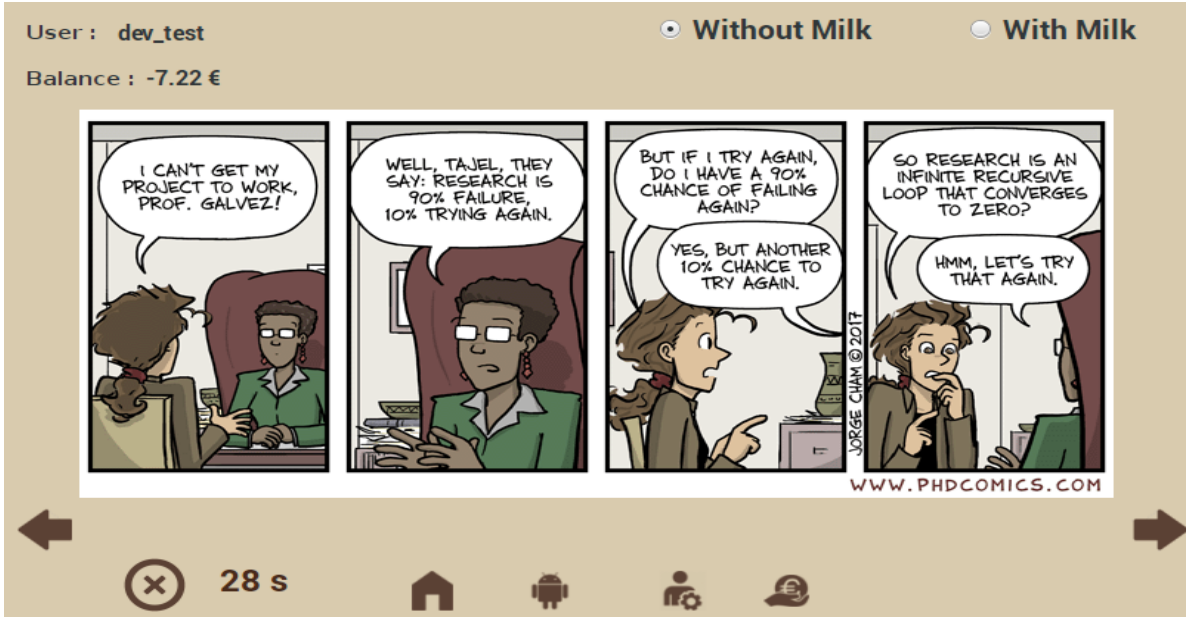
- We can divide the system in 3 main parts : User interface, Main Controller, and Monitoring system

Libraries

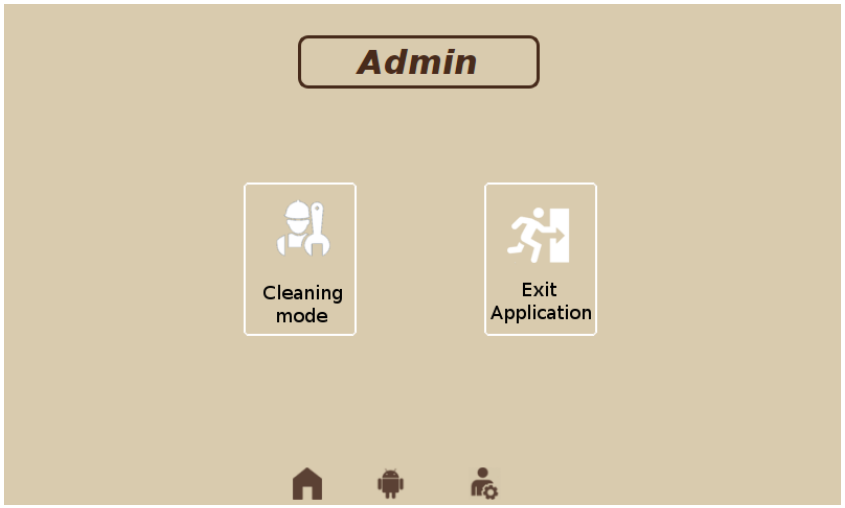
- MFRC522 library : RFID reading
- Beautifulsoup library : Find the image extension before downloading the entertainement content
- PyQt : User Interface
- Transitions : Generate the Finite State Machine
- Tornado library : Intercepting HTTP Request

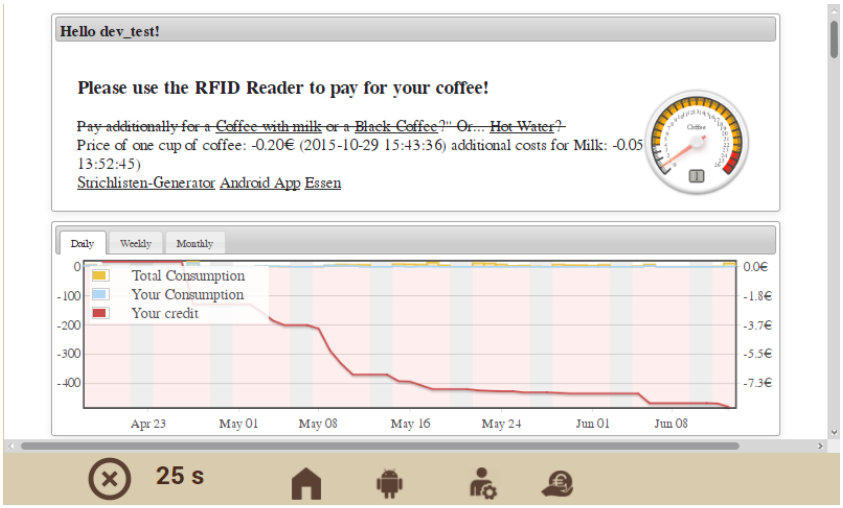
Software: User interface

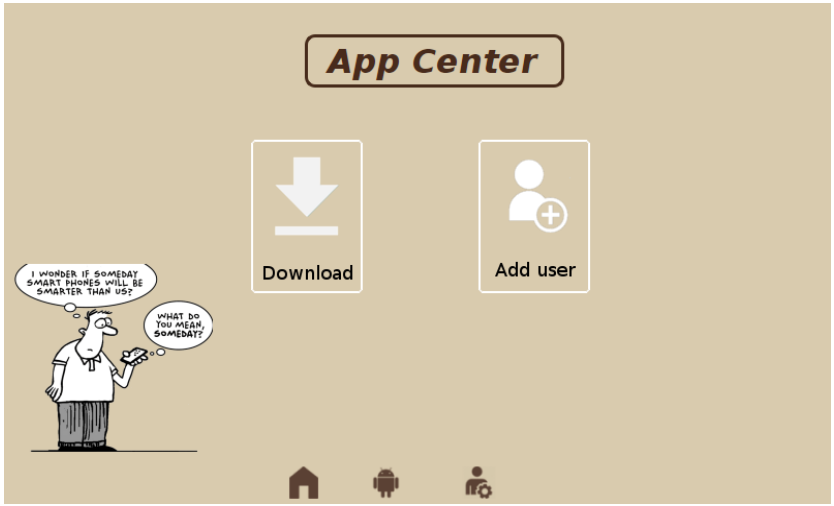
- Provide entertainment content
- Show the accounting page of the current user
- Display the user's informations & order
- Milk choice



Software: User interface

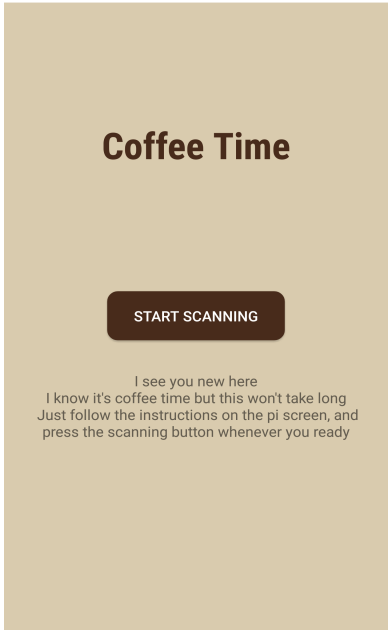


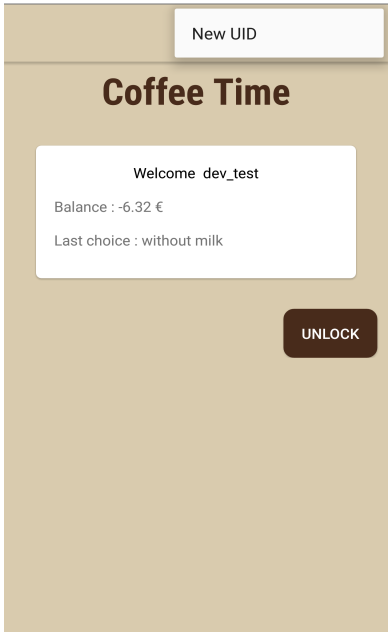




Software: User interface

- Use a smartphone as an identification mechanism
- Manage between the 2 identification processes





Software: Main controller

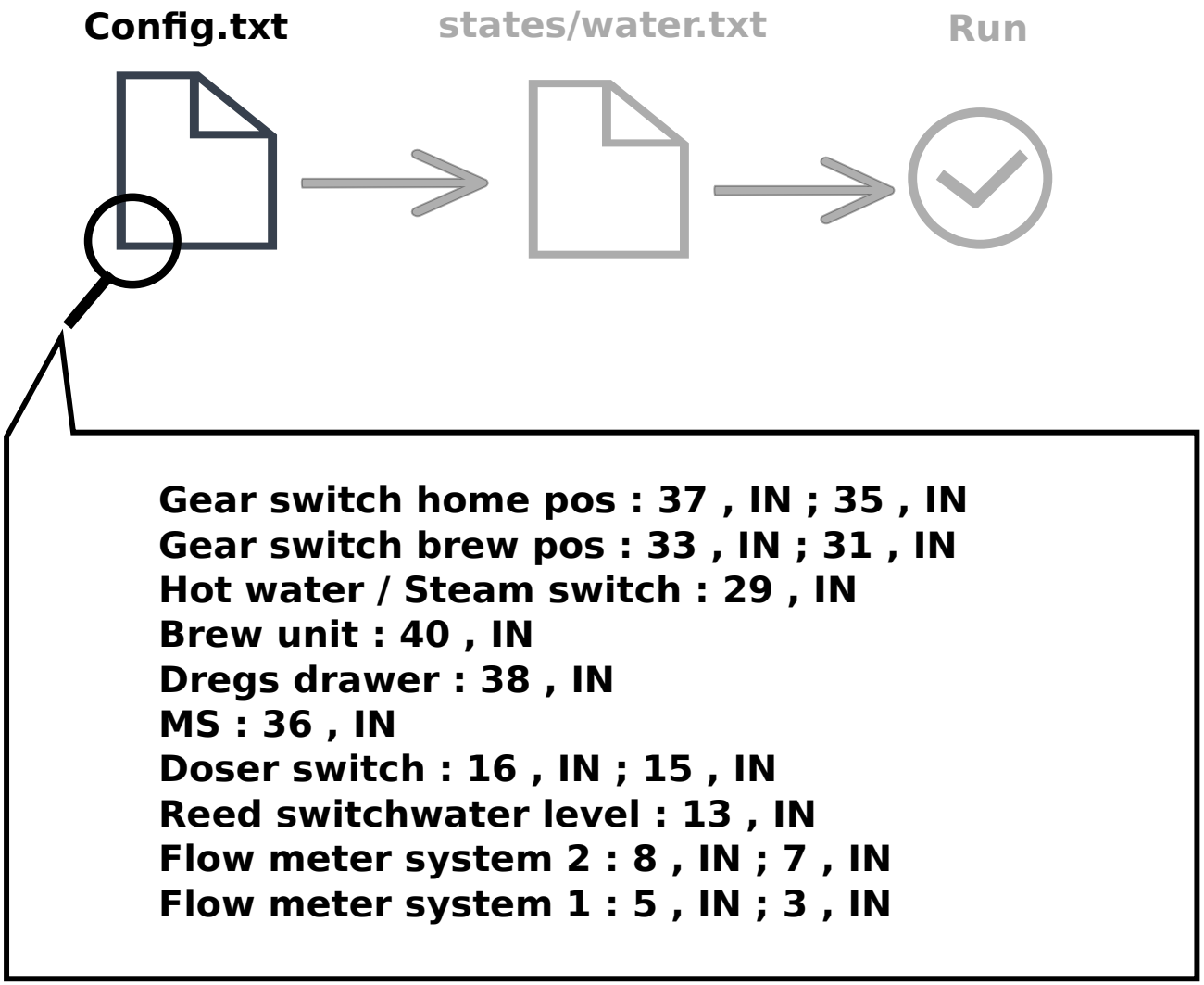
- Communication with the server : Get user information (Name & Balance) / Forwarding the user's order to the server
- Lock/Unlock the coffee machine + Buzzer
- RFID reading
- Daily check up :
  - Download the daily entertainment content
  - Forward any offline transaction to the server

Software: Monitoring System

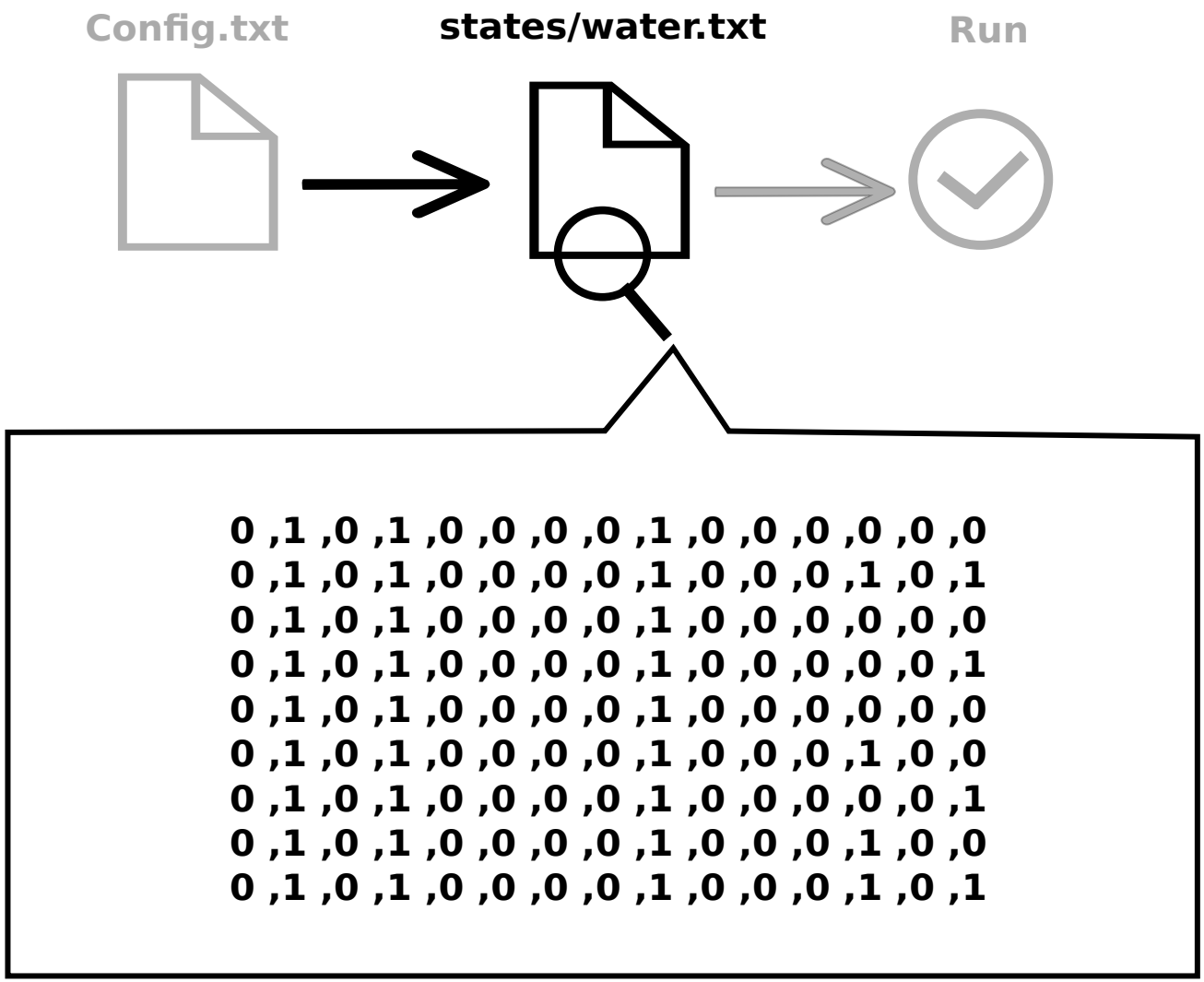
- Continuously read from the coffee machine's sensors
- Generate the machine's corresponding Finite state machine (FSM)

- Configuration :
  - Config file : Sensors name and pin number
  - States folder : Different sensors input for each state in a seperate file
  - Run the program

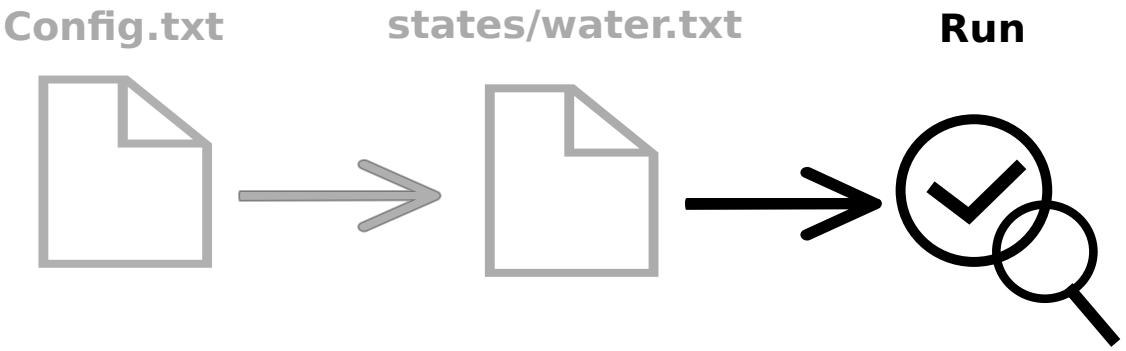
Software: FSM



Software: FSM

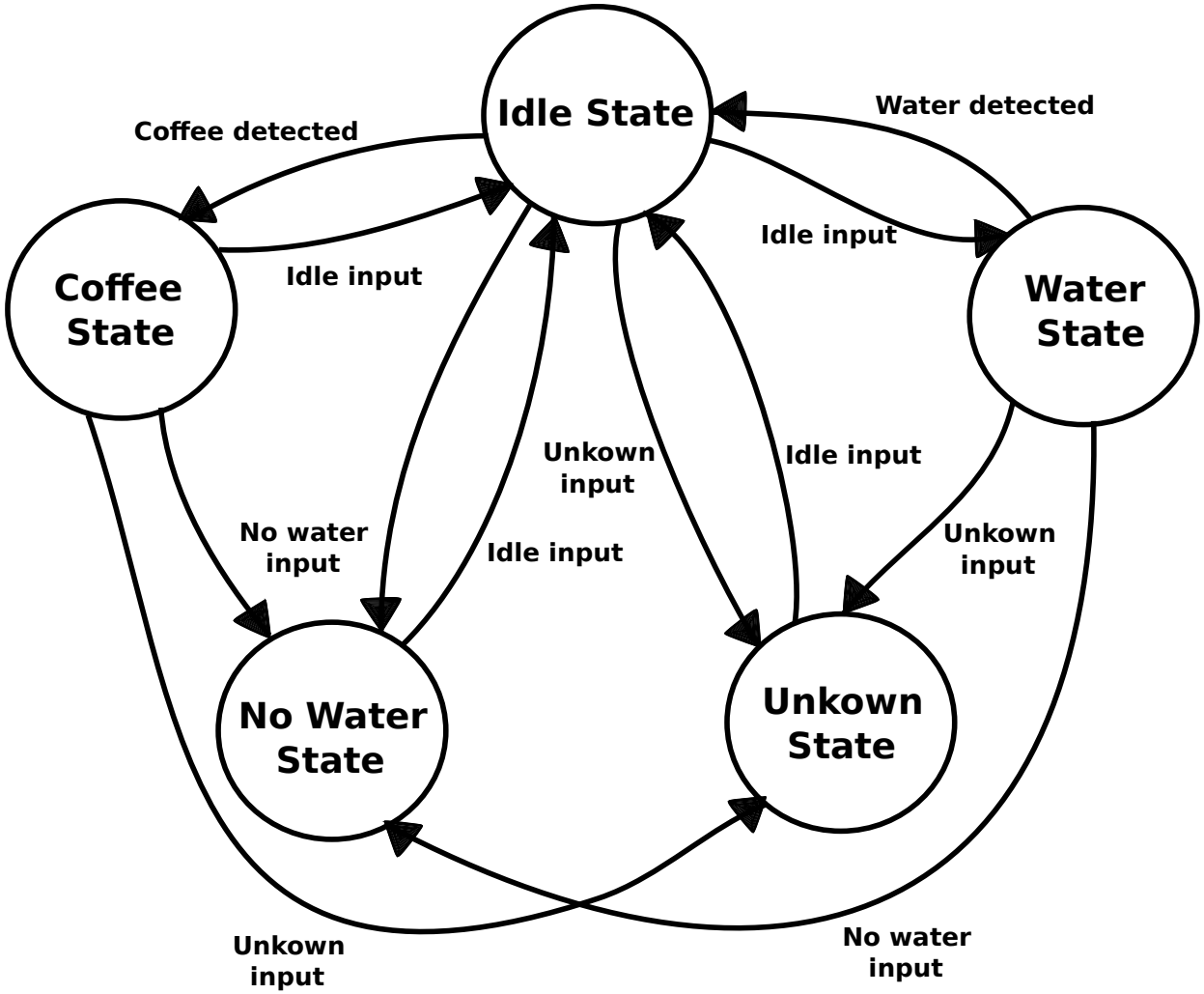


Software: FSM



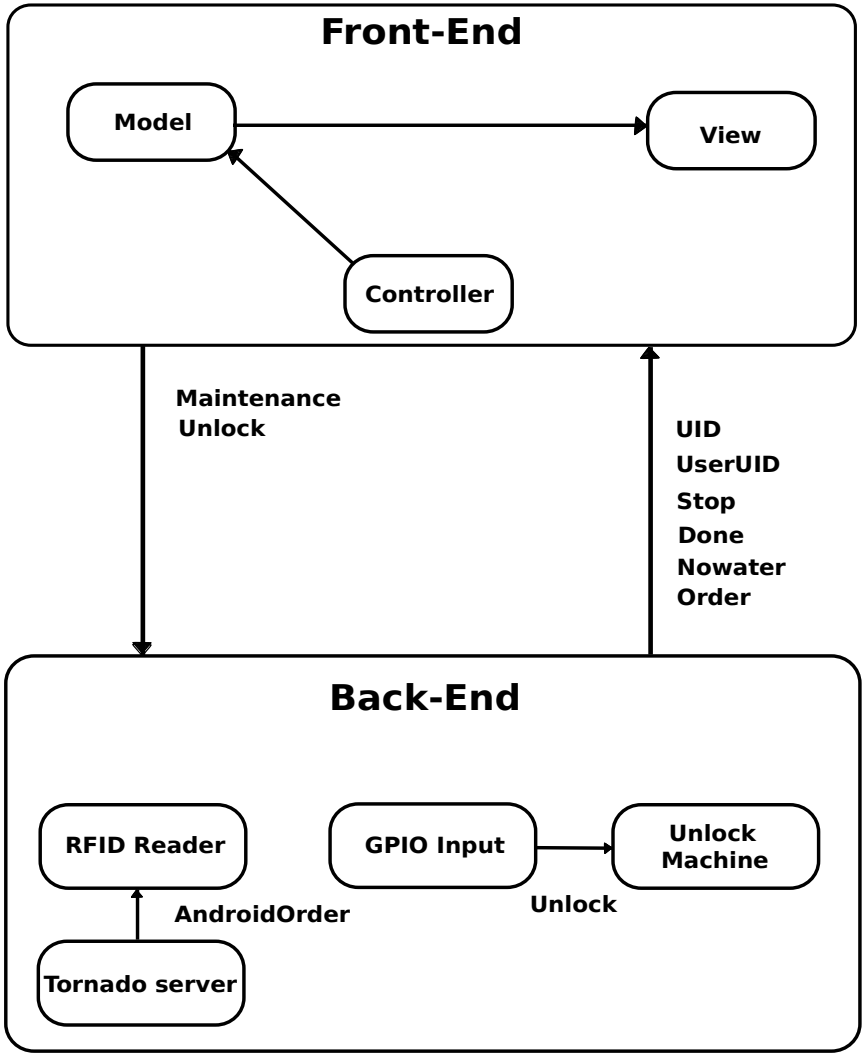

```
try :
    moveToState = getattr ( coffeeMachine ,
    coffeeMachine.convertToFunc( s ) )

    moveToState( )
exceptAttributeError as e :
    print (str ( e ))
except MachineError :
    coffeeMachine.toUnkown( )
```

Software: FSM



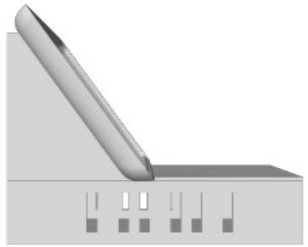
Process communication

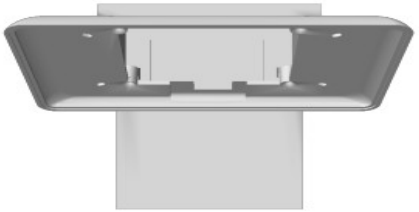
- Shared memory
- Sockets
- Pipes
- Text files

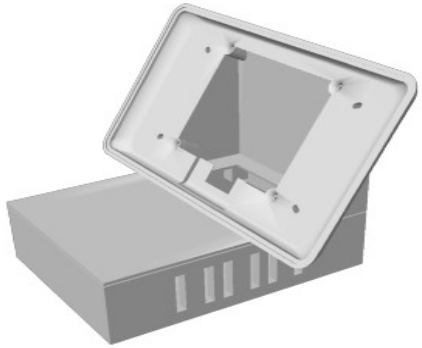


3D Case

- Printed with *Formlab's Form 2*
- Modeling : Blender
- The whole case is composed from 3 separate parts
- Printing Time for each part : 15h.30min







Challenges

- No threads possible : separate processes
- No stable input detected
- Blocking Firewall
- Failing RFID Reader

Conclusion

- We created an automated control system that can be easily plugged to a coffee machine
- RFID Tokens based authentication system
- Smartphone based authentication system
- Offline mode
- Entertainment content
- Automated order detection -> charging the user accordingly
- Monitoring the coffee machine's behaviour, logging, etc
- Sound based state detection ?

Thank you for your attention !