Shared Internet-of-Things Infrastructure Platform (SIoTIP)

Part 2: Architecture evaluation

The second part of the project involves analysis and evaluation of an initial architecture for SIoTIP.

1 Part 2 - analysis and evaluation

The aim of this assignment is to establish a solid understanding of the initial SIoTIP architecture. We've compiled a list of targeted questions that will help you to achieve this goal. Use the architectural description document to formulate an answer to these questions.

Some pointers and guidelines:

- Don't just answer a question with yes/no or with a component name, but share your insights and discuss how you would address the issue at hand.
- Write down the reasoning you followed and highlight the evidence found in the initial architecture. Don't just echo or copy relevant diagrams that you've found in the description document, or translate them into text; just refer to them with their number (section number, page number, figure number, etc).
- Focus on assessing whether the architectural description is clear, complete and correct, or whether it misses important aspects and/or has inconsistencies with other parts.
- If you cannot fully answer a question, state what information is missing, and how you could obtain it. It's OK to make assumptions that cannot be backed directly with evidence from the architectural description, but make sure to state them clearly.
- You are expected to include your opinion on the design choices that have been made into your answers. Did you think a good choice has been made, or do you foresee problems?

1.1 Sensors and actuators

Q1. Do applications need to be aware of the specifics of the device type (e.g., format of the commands accepted by the actuator to which they want to send an actuation command, or sensor measurement unit of the sensor readings)? Why (not)?
Q2. Which components of the system come into contact with sensor data (measurements)?
Q3. Which components would need to change when support for a new type of sensor (e.g., a push button) is introduced, and why would they need to change?
Q4. Which component(s) will deal with MicroPnP-specifics? (Which components directly interact with the MicroPnP middleware, which components are coupled to MicroPnP interfaces or data types, specific commands, etc?).
Q5. What happens when the wrong command is sent to an actuator? (e.g., a command not understood by the actuator, or badly-formatted). Can this occur due to a buggy or faulty application?

1.2 Device installation and failures

Q6. Which component is responsible for detecting that a sensor has failed? How does it do so?
Q7. Are applications automatically activated again when a required sensor is restored? Which component(s) is/are responsible for this?

Q8. Describe in detail how the system deals with failure of a sensor. Make sure to address application instances that rely on the failed sensor being turned off (both their online and gateway parts), determining whether a redundant sensor is available, and switching to that sensor.
Q9. What would happen when a sensor fails just when the connection between the Gateway and the Online Service is disrupted?

1.3 Application development and uploading

Q10. Against which interfaces or APIs are applications developed, i.e. which interfaces can be used by applications?
Q11. What is the relation between the API provided to application developers, and the dashboard used for uploading applications?
Q12. How can an application developer express the user roles for the application? Is there distinction between optional and mandatory roles (i.e., roles that must be filled in before activating an instance of the application)?
Q13. How can an application developer express the requirements of an application in terms of the topology (e.g., the fire detection application needs at least a smoke detector or a temperature sensor, and a buzzer)?
Q14. Is the system aware of the version of a running application (on the Online Service, on the gateway, and for mobile apps?)
Q15. When is a version update complete, and what happens then (with the old version)? Can different customer organizations use different versions of the same application at the same time?
Q16. Write down in pseudo-code the implementation of a SIoTIP light management application that exhibits the following behavior:

1. A sensor reading indicating that presence of a person in a hallway, i.e. from a passive infrared sensor, is delivered to the application instance.

2. The application instance checks the most recent values received from the light sensors outside the building.

3. If it is sufficiently dark outside, the application instance switches on the lights in the rooms adjacent to the hallway.

4. If no presence is detected in a room for 10 minutes, the lights are turned off again.

Make sure that this pseudo-code is based directly on the SIoTIP development API, and is consistent with the signatures and semantics of the methods. Focus exclusively on the interactions between the application instance and the SIoTIP platform, and not the internal SIoTIP interactions.
Q17. What is expected of the developer in terms of testability of his/her applications? What must the application adhere or comply to, to make it testable?

1.4 Running applications

Q18. Explain the position of the `Application`, `GWApplication` and `MobileApp` components on both the client-server (Fig. 1.1) and deployment context (Fig. 3.1) diagrams.
Q19. Which SIoTIP components and nodes are involved in running applications on the online system, on the gateway, and the mobile app, and what are their responsibilities?
Q20. What happens when an application fails, e.g. when it crashes? What happens with the application instance state?
Q21. What happens when a customer organization unsubscribes? Which component is responsible for (1) turning off the application instance, and (2) freeing up resources?
Q22. Can an end-user, on his or her own accord, unsubscribe from the role he or she was assigned within an application instance?
Q23. Describe how a new subscription entered at the dashboard leads to application code being deployed and started on the gateway(s) of that customer organization.
Q24. Are applications running all the time, also when no sensor data or commands are coming in, or can they be stopped and (re)started on demand? Why (not)?
Q25. Consider the following scenario for an instance of the smart heating application:

1. Based on the manual adjustments by end-users, the application instance has learned that the comfort temperature (i.e. the temperature maintained when people are present) for a specific office should be increased.
2. To this end, the application instance part on the Online Service must inform the application instance part of the Gateway located in that room.

Discuss the sequence of all operations called to achieve this scenario. Do you think that the provided APIs provide sufficient support to easily develop applications containing this kind of behavior? Why (not)? What kind of improvements do you have in mind?

Q26. How can you scale up the system when more application instances are needed? Is this a manual or an automated process?

Q27. What constraints are enforced onto applications running on the Online Service? To what extent can they misbehave (e.g. triggering a Denial of Service attack by issuing many requests simultaneously, requesting sensor data that does not belong to the customer organization for which they are running, etc)?

1.5 Application coordination

Q28. How does the `GatewayApplication` instance communicate to its corresponding online counterpart? Explain step-by-step.

Q29. How does the SIoTIP application instance part at the Online Service find a specific gateway application to share information with?

Q30. What happens to application communication when the communication channel between the gateway and the Online Service is down?

Q31. How is a mobile app instance linked to a SIoTIP application? How can a mobile app instance fill in a certain user role in the context of that SIoTIP application?

Q32. What commands can a mobile app send to the system? Are there any limitations on the recipients of these commands?

1.6 Gateway-Online Service communication

Q33. Which components from the gateway communicate with which components from the Online Service? Where did you find this information?

Q34. How does a gateway deal with (detect, react) failure of the communication channel between itself and the Online Service? Why do you think this strategy was chosen?

Q35. What happens with regular communication from gateway to the Online Service (sensor readings, application messages, ...) in case of failure of the communication channel?

Q36. What happens with regular communication from the Online Service to gateway (actuation commands, application updates, reconfigurations, ...) in case of failure of the communication channel?

Q37. How does the Online Service deal with (detect, react to) failure of the communication channel?

Q38. After how much downtime will the system, i.e. Online Service and Gateways, start to lose data? How did you determine this number? Which assumptions did you make and/or which parameters mainly influence this number?

1.7 Resource provisioning and load management

Q39. Which component ensures that not too many applications are running on the gateway?

Q40. What happens when core gateway functionality (i.e. dealing with sensor data, running applications, the MicroPnP firmware itself) crashes/fails?

Q41. Which node(s) are single-points-of-failure, i.e. failure of the node(s) brings down considerable functionality of the system? Is this acceptable/reasonable? How did you determine these? How could you resolve this?

Q42. What happens when one `SensorDBNode` fails? What happens when that instance starts again? (How) is the distributed database kept consistent?

Q43. What constraints are enforced onto applications running on the Gateway? To what extent can they misbehave, e.g. triggering a Denial of Service attack by issuing many requests simultaneously or issuing commands to actuators randomly?

1.8 General

Q44. Which architectural tactics or patterns did you recognize? For which quality attributes do you think they are introduced?
Q45. Did you find inconsistencies in the documentation?
Q46. How would you summarize the strengths and weaknesses of this architecture? Are there other aspects of the architecture that strike you as either genius or odd? If so, list these below and briefly explain.
Q47. Which follow-up questions would you ask the software architect that has created this initial architecture (if any)? Which assumptions or interpretations made during the evaluation exercise would you like to double-check?

2 Instructions

For part 2, you are asked to create a report with motivated answers to the questions above.

Formatting rules You can use the tool of your choice to author the report, but you must deliver a single PDF file, preferably with indexes enabled for easy navigation (easy to achieve in LaTeX with the hyperref package). No other digital formats are accepted. The file must be self-contained (e.g., no extra figures in attachment) and readable (e.g., font size in pictures is adequate). Pages must be numbered. The cover page must mention the course name and the team member names (including their student id numbers). If you decide to use LaTeX, you can use the template that has been provided (on Toledo).

Answer each question that you've chosen with the following template:

> **Q123** (you may copy the question text, but don't have to)
>
> - **Answer:** Your answer to the question.
> - **Evidence and assumptions:** Pointers to the evidence that you've used to answer the question (figure numbers, section numbers, etc.), as well as assumptions that you've made but that aren't in the architectural description.
> - **Opinion and improvements:** Your opinion on how the aspect you've investigated is handled in the current architecture, and possible improvements you'd propose.

Delivery The deadline to turn in your report for part 2 is **April 03, 2019 at noon**. You are expected to (i) upload a self-contained PDF document on Toledo, and (ii) deliver a printed copy of this report in project post boxes (in the student room A00.03) on the main floor of the Department of Computer Science.

Good luck!

The Software Architecture team.