1. Utility tree of ASRs

	Quality Attribute	Attribute Refinement	Summary Rationale ***Business Value*** Rationale ***Architectural Impact***
1	Performance	Latency	Critical events (high priority) within the infrastructure should be delivered to the application instances within 2 seconds. For example: fire detection in a sensor is a critical event.
			H: *It's important for critical events to be delivered quickly, so application developers can rely on this promise and create life-saving applications.*
			H: *No other processes may delay an application instance of seeing a critical event.*
2		Latency	Critical events (high priority) within the infrastructure should be notified to the infrastructure owner within 30 seconds.
			M: *The infrastructure owner should always know what is going on.*
			M: *No other processes may greatly delay this notification.*
3		Latency	Normal priority events that have been sampled by a sensor should be delivered to the application instances under 4 minutes
			M: *Not crucial, but still important for application developers.*
			L: *Resources can be added to achieve smaller response times without changing much of the existing architecture.*
4		Latency	Commands that should respond realtime (some actuators, like an automatic door), should respond in under 2 seconds.
			M: *It's important to have actuators acting quickly, but not crucial.*
			H: *This limit of response time needs to be achieved in a special way and therefore will have great impact on the architecture.*
5		Latency	Scheduled commands should not deviate more than 5 minutes of the scheduled time. In 90% of the cases, this deviation will average 20 seconds.
			M: *To conform to end-user's expectations of the scheduled time, we can't deviate much from the given timestamp, but it's not a realtime command either.*
			L: *Not a lot of extra measures should be taken to achieve this timing.*
6		Throughput	Each application instance on the Online Service gets a capacity of handling 1 Gbps.
			H: *The more computational power, the more applications we can run and therefore profit from.*
			H: *Important part of the architecture to achieve fast computation times.*
7	Availability	Network failure	In case of network failure to the online service, the gateway still needs to perform some critical operations as if the network failure did not occur.
			H: *The customers have the insurance that their critical operations will always work.*
			H: *Some logic will definitely need to be implemented in the gateway so it can carry out these critical operations to the actuators when needed.*

	Quality Attribute	Attribute Refinement	Summary Rationale ***Business Value*** Rationale ***Architectural Impact***
8		Network failure	Within reasonable limits, the delivery of commands from the Online Service to the gateway is guaranteed when the network between the online service and the gateway fails.
			L: *These commands can still be issued again and are therefore not of high business value.*
			L: *The reliability of the underlying network protocol will ensure the reliable delivery within reasonable limits and won't have impact on our architecture.*
9		Network failure	In case of network failure between the gateway and the online service, a moving window of 1000 events are guaranteed to be eventually delivered.
			M: *Customers need to know how much of the events still come through when communication fails.*
			L: *Only some extra cache might need to be reserved for the delayed events.*
10		Downtime	The Online service should be available 99.99% of the time.
			H: *The Online Service is a critical component for the whole system, when it goes down we lose money.*
			H: *Extra servers need to be acquired to act as a backup. More over the system needs to be able to handle switching hosts for the online service*
11		Downtime	The gateways should each be available 99.5% of the time.
			H: *The gateways are important for the connection between applications and hardware. They also need to handle critical operations in some situations and therefore need to be reliable. Because a gateway failure will only have local impact, this 99,5% is good trade-off between up-time and cost.*
			L: *Changes for improving gateway up-time should be localized on the gateway itself or should not have that much impact on the existing architectural components.*
12		Downtime	The whole SIoTIP system should continue its service without suffering any downtime in case of installing new hardware.
			H: *Adding/re-enabling hardware is rather the norm than the exception. This both because of scaling and failure of hardware. Therefore it is crucial that the system keeps operating while devices are added or removed.*
			M: *MicroPnP already supports plug-and-play functionality which will ease achieving support for this plug-and-play functionality in our system.*
13	Interoperability	Data standardization	The communication between the sensors/actuators and the online service via the gateway will work correctly 99.9% of the time.
			H: *Application developers should know in what format the data will come for processing and how commands should be structured by means of a well defined API. Keeping data in a consistent way is also interesting for big data purposes.*
			L: *This just requires some light processing by the API.*
14	Security	Application compromised	When an application is compromised, only the data and sensors it has access to, may be compromised.
			H: *Customers need to aware of permissions granted to applications and thus should know what data can be accessed by these applications.*

	Quality Attribute	Attribute Refinement	Summary Rationale ***Business Value*** Rationale ***Architectural Impact***
			M: *Extra components may need to be introduced to make sure a (compromised) application only accesses it's permitted hardware.*
15	Usability	Proficiency of development	Getting a basic application up and running on a test environment should be possible in less than 30 minutes for an experienced developer. Such a basic application can for example be a variant of 'Hello World' for IoT.
			H: *Attracting application developers is key to get a wide range of applications on our platform. Such a variety of applications will surely attract a lot of customer organizations which is very important for our business. Allowing a smooth learning curve can be a crucial factor for attracting developers.*
			L: *Providing a starter application and documentation/guide should suffice.*
16		Manual configuration	The act of deploying (starting the deployment process) an application on the SIoTIP platform should be performable in one click / in less than 1 minute. The deployment itself can however take longer, but should not require further interaction.
			M: *We should provide a hassle-free means for deploying applications as an extra advantage of our platform.*
			M: *Extra components should be introduced to allow an automated deployment of the application.*
17		User experience	Performing tasks on the customer organization dashboard (e.g. subscribing to applications and changing end user roles) should require no training at all for a user.
			H: *Customer organizations want a hassle-free means for interacting with our system that is as intuitive as possible. If the interface is not intuitive, customer organizations will refrain from using our system and will hinder adoption.*
			M: *We should allow transparent hardware installation to the point of view of the customer organization and require as less manual configuration as possible.*
18		User experience	Performing tasks on the infrastructure owner dashboard (e.g. monitoring/configuring hardware and configuring their topology) should require less than 8 hours of training for an infrastructure owner.
			M: *If their are enough users, infrastructure owners will have an incentive to get trained, so it's less important to have this dashboard as intuitive as possible. However, with enough training an infrastructure owner should be able to have full control.*
			L: *The complexity of this dashboard should not have much impact on the architecture.*
19	Safety		Application instances can never overwrite commands with a higher priority.
			H: *It is important that customer organizations can rely on the correct but safe interaction between multiple applications. This is also highly important for accountability purposes.*
			M: *Extra components may need to be introduced in order to support command priorities.*
20	Portability	Hardware changes	When microPnP hardware is replaced with hardware from another vendor, the only software that may be affected is the part on the gateway it's connected to and responsible for its communication.

	Quality Attribute	Attribute Refinement	Summary Rationale ***Business Value*** Rationale ***Architectural Impact***
			H: *Our choice for MicroPnP should not put us in the dangerous situation of being totally dependent on their hardware.*
			M: *It should only require some sort of abstraction layer on top of the hardware.*
21	Testability	Cost/time	Setting up a test-environment should take less than 20 minutes for an experienced developer. Also it shouldn't cost anything at all.
			H: *Testing an application is key to the development of qualitative applications. But in order to not hinder the application testing, we should provide a cheap environment where for example no real hardware should be bought.*
			M: *Some hardware and software components must be able to be simulated in virtual environment.*
22		Controllability	For an experienced developer, it should not take more than a minute to simulate certain events (e.g. detection of fire) or mimicking other sensor data in the test environment. A developer should be able to set up all kinds of test-cases in an effortless manner.
			H: *Both application developers and customer organizations want to be sure about the correct working of their applications. For this purpose easy simulation of events and mimicking sensor data will be an attractive feature of our platform.*
			M: *Extra simulation components will be needed to support the simulation of events.*
23	Scalability	Scale	For the first year, the SIoTIP platform should be able to handle 300 customer organizations and 80 applications.
			H: *The system should be able to support these numbers in order to meet our short term business goals while ensuring a short time to market.*
			M: *This amount of users needs to be taken into account when enrolling our system, we already need to be prepared for this number.*
24		Scale	A gateway must be able to handle at least 30 sensors.
			M: *For a standard small-scale infrastructure setup (for example for one floor) only one gateway should be sufficient.*
			L: *We should look into having sufficiently good hardware rather than changing our software architecture to achieve this number.*
25		Elasticity	The capacity of the online service can be increased/decreased within 2 weeks, and should have a fairly linear marginal cost.
			H: *We aim for rapid growth to roll out a worldwide platform that is the primary means for IoT developers to deliver their applications. So scaling the system rapidly along with the users is of high importance.*
			H: *Our software components should be able to work with multiple instances of our service. A very different approach should be taken to support this requirement, which can have a drastic impact.*
26		Elasticity	The capacity of the online service can be increased/decreased within 20 minutes, while ensuring the cost-effectiveness of this operation.
			M: *From the business perspective it could be very attractive to scale down at night when probably less traffic will happen to keep costs as low as possible. At peak hours it should also be possible then to scale up very quickly.*

Quality Attribute	Attribute Refinement	Summary Rationale ***Business Value*** Rationale ***Architectural Impact***
		H: *Start up time for new instances should be as low as possible. Our software components should be able to work with multiple instances of our service.*

2. Quality Attribute Scenarios

2.1 Availability: Local processing of critical events

ASR: 7

Some events are extremely time-sensitive (e.g. a fire is detected by a smoke alarm), for these events it is of utmost importance that some basic actions happens immediately (e.g. triggering the fire alarm). These critical action need to happen regardless of network availability.

- **Source:** A sensor with critical data.
- **Stimulus:**
  - Critical data that can be directly processed by a lightweight application running on the gateway is pushed to the gateway.
- **Artifact:** The sensors that need to communicate, and the gateway.
- **Environment:** Run-time, more specifically during normal operations as well as in an environment with degraded network capability.
- **Response:**
  - The gateway receives data from the sensor and does some preliminary processing to transform the data into a system event.
  - A lightweight application running on the gateway processes the event, and if it is deemed critical for another application running on the gateway the event is directly passed to that application.
  - The receiving application can view the event and take limited action (e.g. triggering an actuator).
- **Response measure:**
  - Response time between data being read on the sensor and a command reaching the actuator should be the same regardless of network availability when the gateway can do the processing required.

2.2 Scalability: Scaling quickly

ASR: 26

From the business perspective it could be very attractive to scale down at night when probably less traffic will happen to keep costs as low as possible. At peak hours it should also be possible then to scale up very quickly.

- **Source:** System users.
- **Stimulus:**
  - Cyclical usage pattern (daily, weekly, etc) causes rise and fall in system load.
- **Artifact:** Online service.
- **Environment:** Run-time, high/low demand.
- **Response:**
  - System load is monitored.
  - A system load above 0.8 (ratio) is detected.
  - New instances of the online service are spun up until load is sufficiently below the threshold value.
- **Response measure:**

  - The online service responds to a change in load within 20 minutes.
  - Adding new instances to the online service has a linear marginal cost.
  - Existing users are not affected.

2.3 Safety: Safely sharing hardware

ASR: 19

One important requirement of the overall system is that hardware sensor are shared across applications. It is crucial however that applications do not interfere with one another, more specifically from a safety or security standpoint. For example, the IoT door locks should not prevent the fire department from entering the building, while the building is on fire.

- **Source:** An application (potentially malicious) with low physical priority access to a shared actuator.
- **Stimulus:**
  - A conflicting request is sent to the shared actuator (conflicting is defined here as opposite actions, e.g. open/close), by a unknowing rule abiding application (meaning that no harm is indented).
  - A conflicting request is sent to the shared actuator by an application that intends to do harm.
- **Artifact:** Local mesh network.
- **Environment:** Pre run-time / Run-time, assuming the gateway is not compromised.
- **Response:**
  - Each application requests a priority for the commands it will be executing. This happens pre run-time.
  - The lowest priority is granted per default, higher priorities will require review.
  - At run-time when two conflicting requests arrive at the same actuator, the request with the highest priority is executed. As an example, the command to open all doors in case of a fire has a higher priority than the command to lock all doors unless the actor has a key.
- **Response measure:**
  - The system should never execute a command with a lower physical priority than one with higher physical priority.

2.4 Performance: React quickly to critical events

ASR: 1

As the system grows and grows, performance keeps being an important factor. Critical events need to reach the application instances as quick as possible so they can react to it. For example: a critical event might be a fire alarm going off, after which an application instance could turn on the sprinklers. This needs to happen as quick as possible.

- **Source:** A sensor with critical data
- **Stimulus:**
  - The sensor sampled data which is considered critical.
- **Artifact:** The sensor that samples the data and the gateway it is connected to.
- **Environment:** Run-time, normal operation.
- **Response:**
  - The sensor sends data over the mesh network to its corresponding gateway.

  - The gateway receives the data and sees it is considered critical. It creates a critical event of this and gives it to the application(s) that considers this critical.
  - Delivery of critical events has a high priority, and as such is placed in a different queue. The system also dedicates extra resources to handling this prioritized queue to ensure performance remains within spec even at high load.

- **Response measure:**
  - The critical event is delivered to the application instance(s) within 2 seconds.