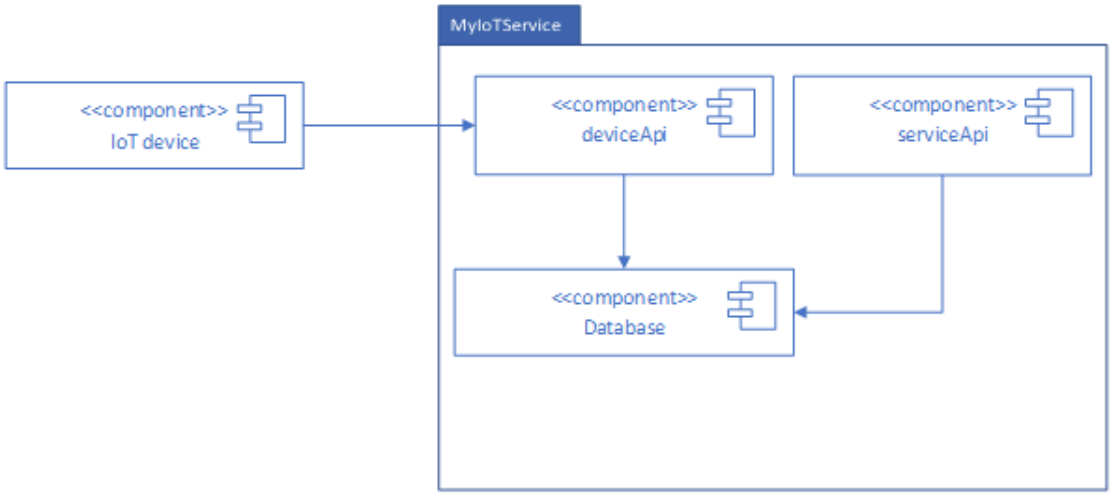
IoT connectivity: MyIoTService

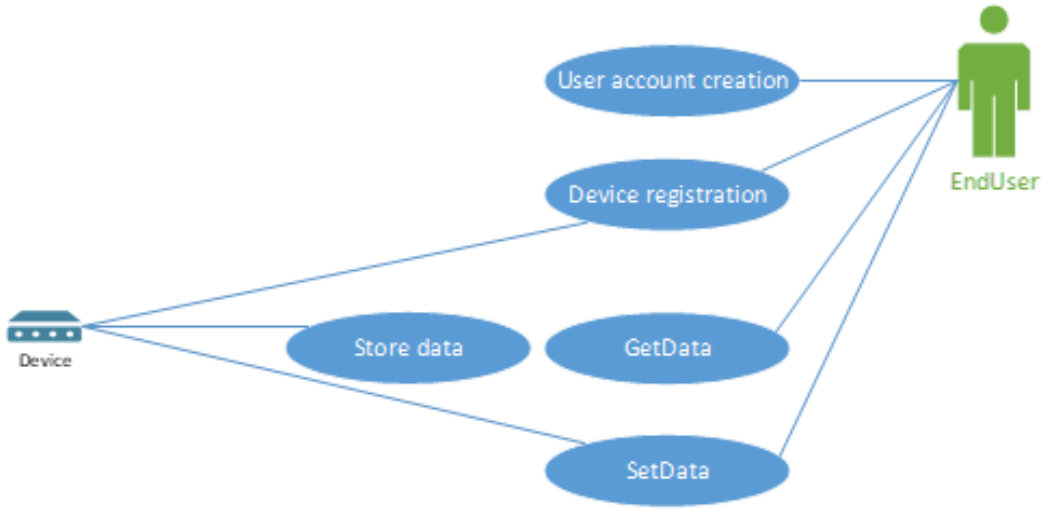
Goal

An ambiguous business goal states: „MyIoTService has to be capable to handle 10 000-50 000 Devices simultaneously and be responsive for 1000 active daily users with further hockey stick growth in future“.
We start from the very blank...
Proposed high level architecture for highly scalable MyIoTService:



Use-case scenarios

MyIoTService – Cloud solution that connects Users and their devices. It receives the data from Device and has to store the current state to database. It provides the data to authenticated User.
EndUser – can create an Account into MyIotService solution. User can log in into the system and make needed configuration to get his Device connected into MyIotService.
Device – smart embedded device. Device has to connect to MyIoTService and send its data.



Device

Hypothetical Client towards MyIoTService. Implement simple Client emulator. In a way how the Device should communicate with MyIoTService.

- It has to have capabilities for authentication in MyIoTService
- It has data that gets sent to MyIoTService
- Some of the data in Device changes in every 1-10seconds
- Some of the data are writable for MyIoTService

Data:

Id	Name	Ranges	
1000	Inside temperature	[-150;150]	
1001	Outside temperature	[-100;100]	
1010	Has outside temperature sensor	[0,1]	If 0, Id 1011 is meaningless
1011	Outside temperature	[-100;100]	
1020	Water temperature	[-100;100]	
2000	OperationTimeInSec	0 ... intMax	Increments in every second
2010	WorkingHour	0... intMax	Increments in every hour
4040	IsOperational	[0;1]	1 – ON, 0 – OFF
4050	SilentMode	[0;1]	1 – Operational mode „Silent“
8000	Machine is broken	[0,1]	1 – broken, 0-healthy
9000	SerialNumber	0..999999	Unique per Device

Device connectivity:

What protocol to use? Free to choose. For example: websockets, Json-RPC, Mqtt, RestApi, etc... up to you.

MyIotService

Has to provide connectivity and data storage for Device. Provide RestAPI endpoints for User. Main scenarios required: account creation, device registration, monitoring the current data and sending new data configuration to Device.

Scope

It is a Backend assignment for .NET. Front-end Web part is not required. It is OK to include minimal web interface as well if that is supportive.

This test assignment leaves free hands to the developer. Be creative and think along. It is also OK to discard all the proposed scenarios and architectural propositions if you can make up better solution and you can describe your approach and ideas in later discussions.

If the scope is too big, then skip the „SetData“ part.

We don't want this technical test to be too hard or too time consuming for you. It is also ok then to skip some other sections if you want, but in that case high-level description on how to approach the missing parts are required.

Deliverables

- Source code of the solution
- Build guide (engineer to engineer)
- Setup guide (engineer to engineer)
- Please use a private git/bitbucket repository to deliver your code. Alternatively, a zipped version of your local git repository
- If you can't or don't want to do everything in this test, then you can describe the missing parts (we don't want this technical test to be traumatic or too hard or too time consuming for you)

If you have any doubts or questions, feel free to reach out to us anytime :)