System Architecture Specification

Edit New Page Jump to bottom

Jannik Schwarz edited this page 17 hours ago · 18 revisions

Change log

Date	Author	Notes
03.11.2019	Jannik Schwarz	Created the document and added the system overview, architectural concept
04.11.2019	Noah Bross	Added system design and systemspecification
06.11.2019	Nicolas Breuninger, Marvin Sonntag, Rene Scholz	Added technical concepts
15.04.2020	Jannik Schwarz, Nicolas Breuninger	Updated the UI part of the system design
14.05.2020	Jannik Schwarz	Updated the module descriptions and specifications

Introduction

Glossary

- API: Application Programming Interface
- JSON: JavaScript Object Notation
- NodeJS: Programming language for server-side JavaScript
- Angular: Frontend programming language

System Overview

System environment

The application is not dependent on other applications in the environment as it draws all the information it uses from the hardware connected to the network.

Hardware environment

Expected is a Profinet-network. That means the application is running on a machine which is in a network with profinet devices.

Software environment

The application is platform indipendent, though it will be developed with Windows 10 in mind. NodeJS and Angular are required.

Architectural Concept

Quality

Efficiency

- The application responds with a list of devices in the network within 0.2 seconds.

Portability

- The application can be used on many different machines. The platform indipendent nature of NodeJS and Angular allow for a quick and easy solution on most operating systems.

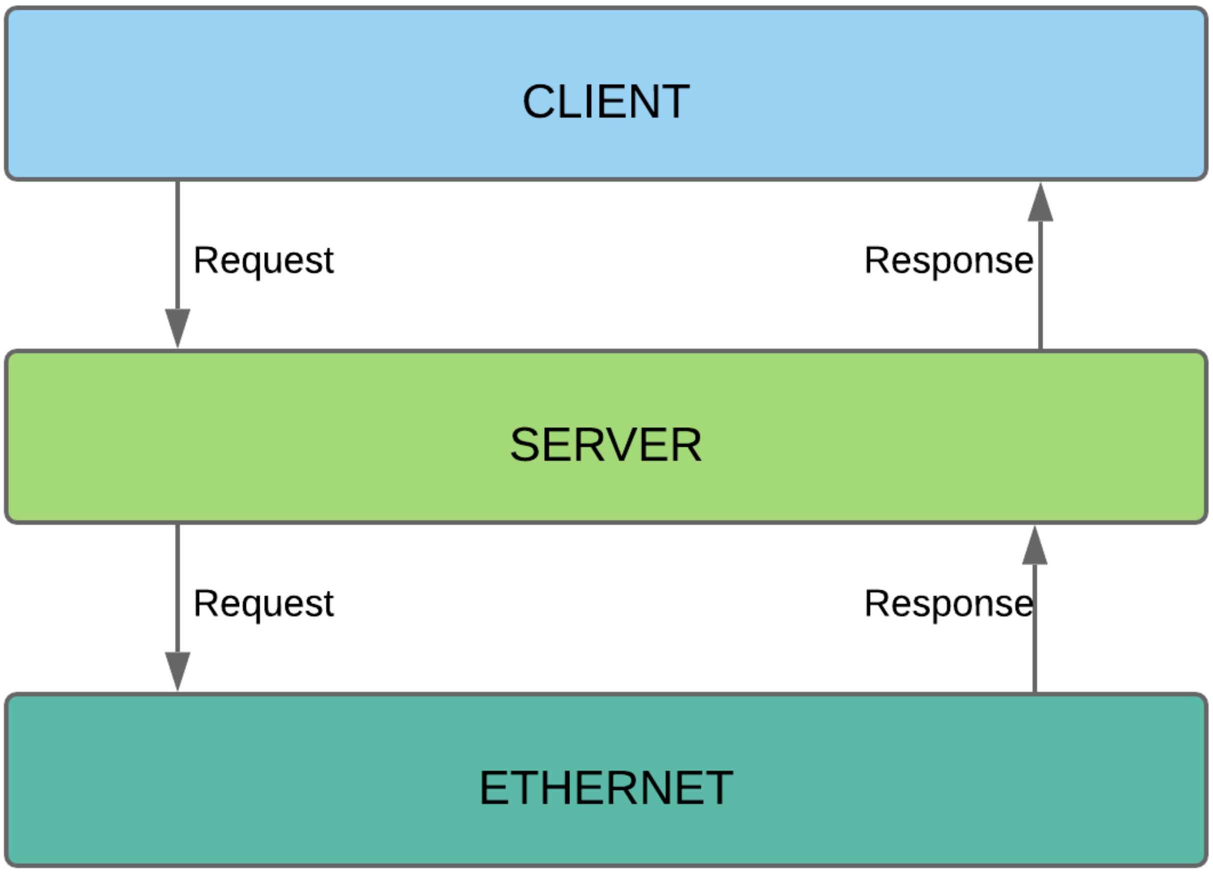
Reliability

- Before changing values of a device, the entered parameters will be checked for correctness by the server. Misconfiguration will be reduced to a minimum by this procedure, helping with possible data faults and system reliability.

Usability

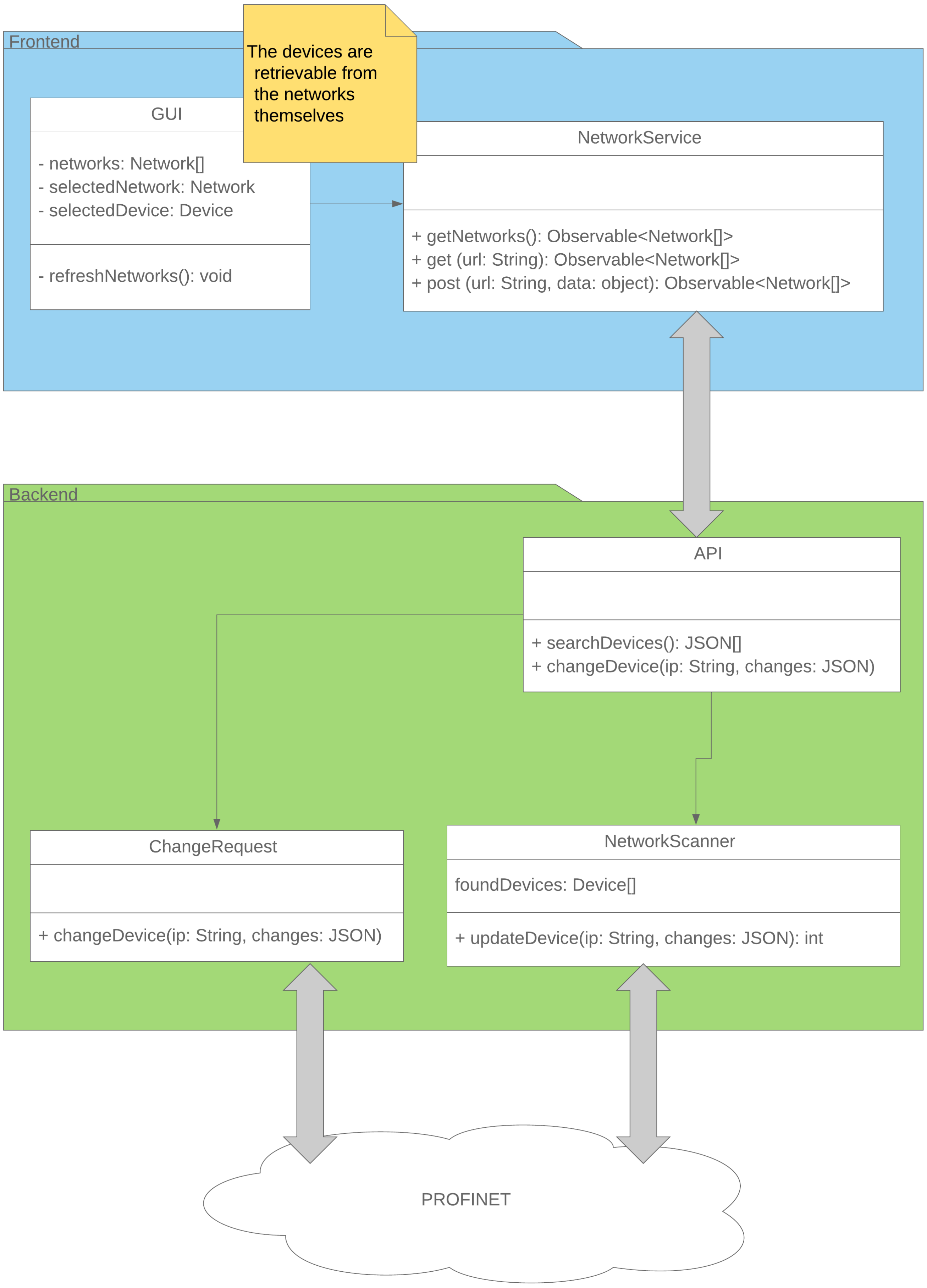
- The minimalistic design makes it easy to navigate the client and quickly understand the application.
- Without an overloaded interface the application will be easy to understand.

Architectural Model



The layered model as chosen to display the system architecture because it gives a clear overview over the topology of requests. The client sends requests to the server. The server then either communicates with the network it is in and waits for responding devices before responding itself, or responds to the request directly with already gathered information.

Systemdesign


Frontend
The devices are retrievable from the networks themselves
GUI
- networks: Network[]
- selectedNetwork: Network
- selectedDevice: Device
- refreshNetworks(): void
NetworkService
+ getNetworks(): Observable<Network[]>
+ get (url: String): Observable<Network[]>
+ post (url: String, data: object): Observable<Network[]>
Backend
API
+ searchDevices(): JSON[]
+ changeDevice(ip: String, changes: JSON)
ChangeRequest
+ changeDevice(ip: String, changes: JSON)
NetworkScanner
foundDevices: Device[]
+ updateDevice(ip: String, changes: JSON): int
PROFINET

Subsystemspecification

<MOD-ID 1> GUI

Descriptor	Description
System requirements covered	LF10, LF20
Usage	Gives validated data to the UI and checks for user input and stores retrieved data by the API-Caller to allow future look up of data
Access	GUI-Components
external data	Networks from the NetworkService-module
MOD	here

<MOD-ID 2> NetworkService

Descriptor	Description
System requirements covered	LF10, LF20
Usage	Used to store the network data. It also holds the data which is displayed in the GUI. Additionally it handles the communication between front- and backend.
Access	AppComponent
external data	API-data from requests to backend.
MOD	here

<MOD-ID 3> API

Descriptor	Description
System requirements covered	LF10, LF20
Usage	Sends and receives data about the current network. It is the connecting layer of front- and backend.
Access	NetworkService, ChangeRequest, NetworkScanner

Descriptor	Description
external data	NetworkScanner, NetworkService
MOD	here

<MOD-ID 4> ChangeRequest

Descriptor	Description
System requirements covered	LF20
Usage	This module changes the information of a device in this network.
Access	local network
external data	API (change request data)
MOD	here

<MOD-ID 5> NetworkScanner

Descriptor	Description
System requirements covered	LF10
Usage	This module scans the local network for profinet-devices.
Access	
external data	local network
MOD	here

Technical Concepts

User Interface and ergonomy

When creating the UI simplicity was a key value we had in mind. All information should be easily accessable without much trouble. For that reason we thought to make a UI that guides the user through the application step by step. The application will be used from left to right. Starting at the network, going to a list of devices and ending with details of a selected device.

Data Validation

For every entry there will be a type and validity test. The type test will find out what type of data, String or number or boolean, we're dealing with. The validity test looks at the plausability of values. For example an IP-adress must not be "0.0.0.0", as well as every IP-adress found must be in the same subnet.

Configurability

The backend will be configurable with a config file. The IP-adress of the Profinet-System can be configured. The frontend will be configurable when compiling. Then a IP or hostname can be set to connect to the backend.

Internationalisation

Angular, the frontend framework, allows for support of internationalisation. With translation tables it is possible to use many different languages.

Testability

With modularity in mind everything can be tested with unit-tests. That should garuantee the correctness of the application

Scalability

With containerisation our frontend and backend services can potentially be replicated indefinetly. Thusly the services could scale to near infinity, if the resources are given. So the only bottleneck visible at the moment is the Profinet-Bus.

Availibility

Our services can be deployed on various servers and with help of load balancers grant a very high availibility. The only reasons the availibility might get worse are a black out, taking out the Profinet-sytem, or a hardware defect.

Pages 4

Find a Page...

+ Add a custom sidebar

Clone this wiki locally

```
https://github.com/nicolasbreuni/Tinf18C_Team_2_Profinet_DCP_Client.wiki.git
```