Design and implementation of configuration synchronization and WebGUI for decentralized multi LiDAR setups

Uday Navaneeth Menon
Advisor: Johannes Meßmer
Chair of Decentralized Systems Engineering
https://dse.in.tum.de/

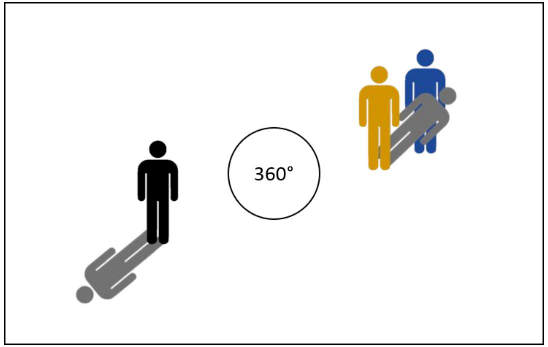
15.12.2021 – 15.08.2022

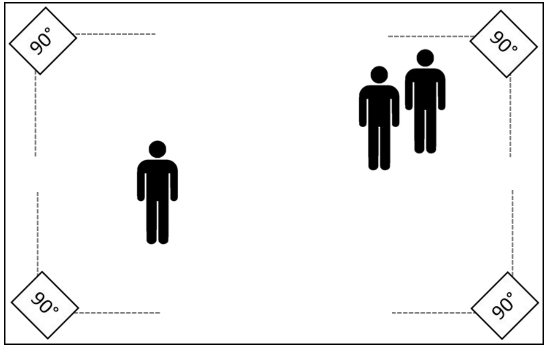
Outline

- **Motivation**
- Overview
- Background
- Design
- Implementation
- Summary
- Future work

Motivation: Research context

- Current LiDAR technologies
  - Mechanical rotating LiDAR
  - Solid state LiDAR
- Distributed LiDAR system (Swarm)
- Possible applications of Swarm
  - People counting
  - Volume measurement
  - Perimeter security
  - Traffic management





Problem statement

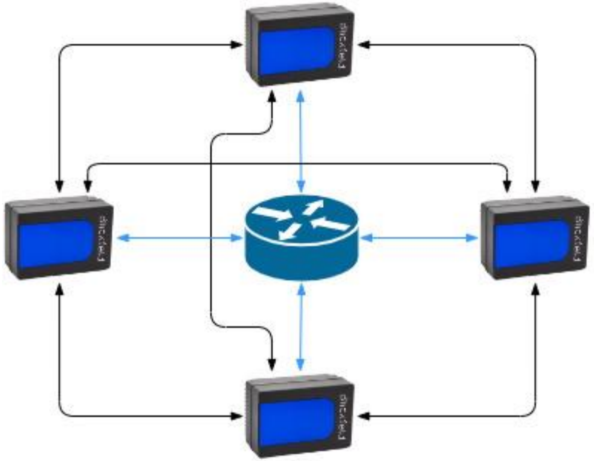
- Configuration synchronization
  - Discovery service to search the network for LiDAR devices and Swarm clusters
  - Configuration service to synchronize configuration of all devices and maintain homogeneity of cluster
- WebGUI
  - Designing a modular system to help improve scalability
  - Component based interface design to facilitate reusability and a streamlined process of adding new elements
  - User experience to enable users to work with the device with little to no training needed

Outline

- Motivation
- **Overview**
- Background
- Design
- Implementation
- Summary
- Future work

Overview

- Blickfeld Qb2 [2]
    - Solid-state LiDAR
    - On-device processing
    - Software defined LiDAR
- Swarm
    - Distributed LiDAR cluster
    - Same physical location
    - Combined output

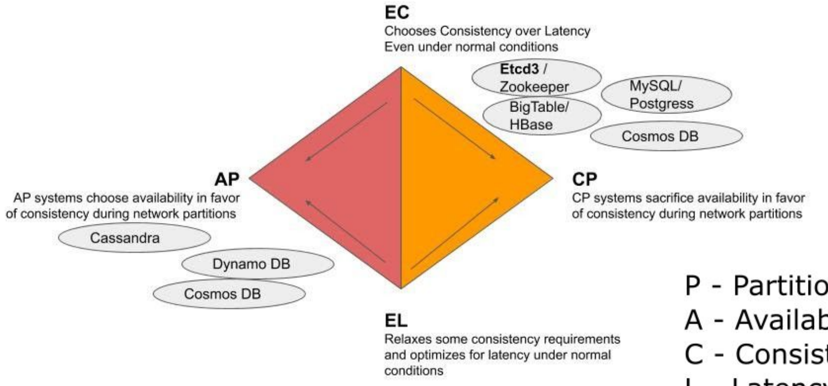


Outline

- Motivation
- Overview
- **Background**
- Design
- Implementation
- Summary
- Future work

Background

- Network discovery
  - Zero configuration networking
  - Multicast Domain Name System
- Distributed key-value storage
- grpc Remote Procedure Call



EC
Chooses Consistency over Latency
Even under normal conditions

Etcd3 / Zookeeper

MySQL/ Postgress

BigTable/ HBase

Cosmos DB

AP
AP systems choose availability in favor
of consistency during network partitions

Cassandra

Dynamo DB

Cosmos DB

CP
CP systems sacrifice availability in favor
of consistency during network partitions

EL
Relaxes some consistency requirements
and optimizes for latency under normal
conditions

P - Partitioning
A - Availability
C - Consistency
L - Latency

PACELC representation of key-value storage technologies [1].

Outline

- Motivation
- Overview
- Background
- **Design**
- Implementation
- Summary
- Future work

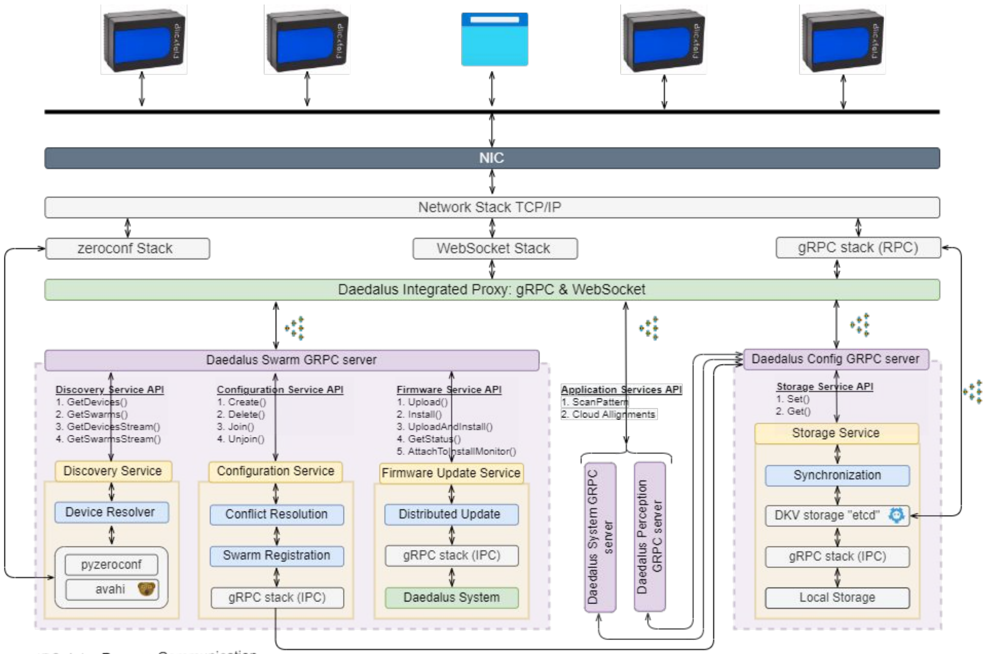
Design overview: Swarm

- Swarm core functions
  - Registration
  - Alignment of point clouds
  - Configuration synchronization
  - Distributed configuration storage
  - Maintenance

- Swarm rules
  - Number of devices
  - Positioning
  - Connection
  - Configuration

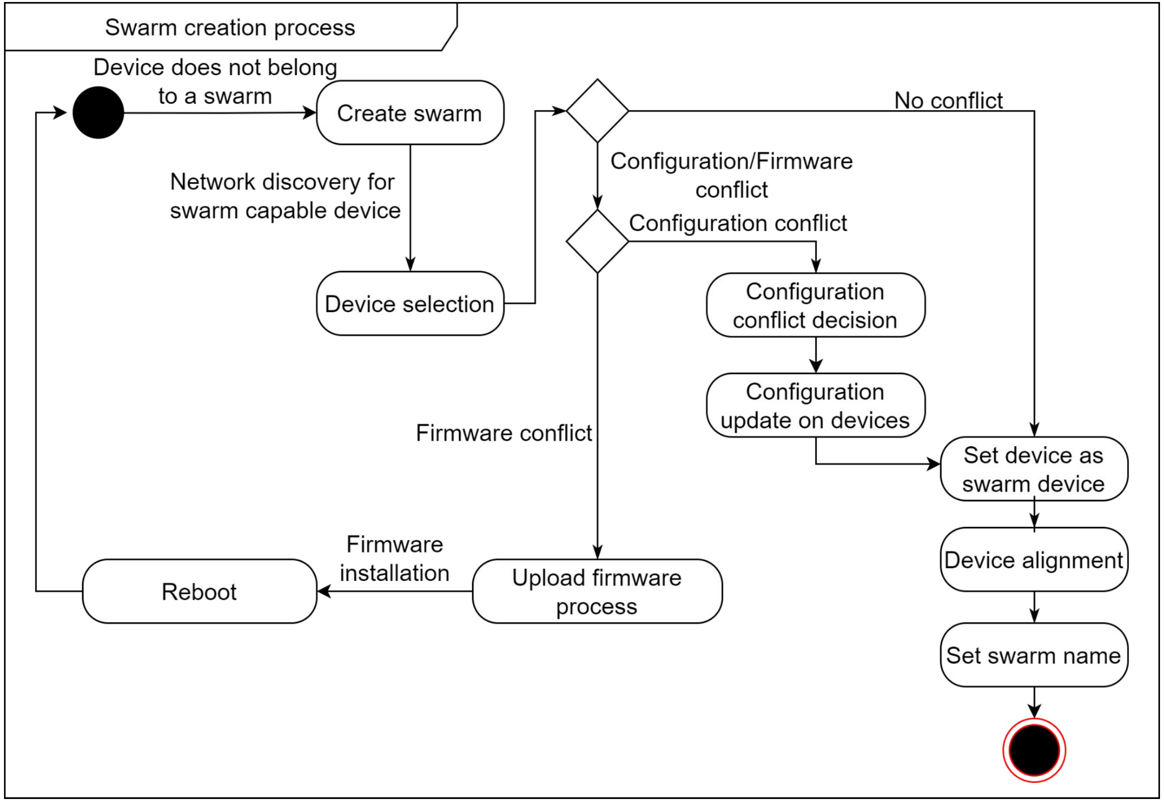
Design overview: Swarm

- Discovery service
  - Python zeroconf
- Configuration service
  - Swarm identifier
  - Distributed configuration storage
  - Time synchronization
  - Frame synchronization
  - Scan pattern settings

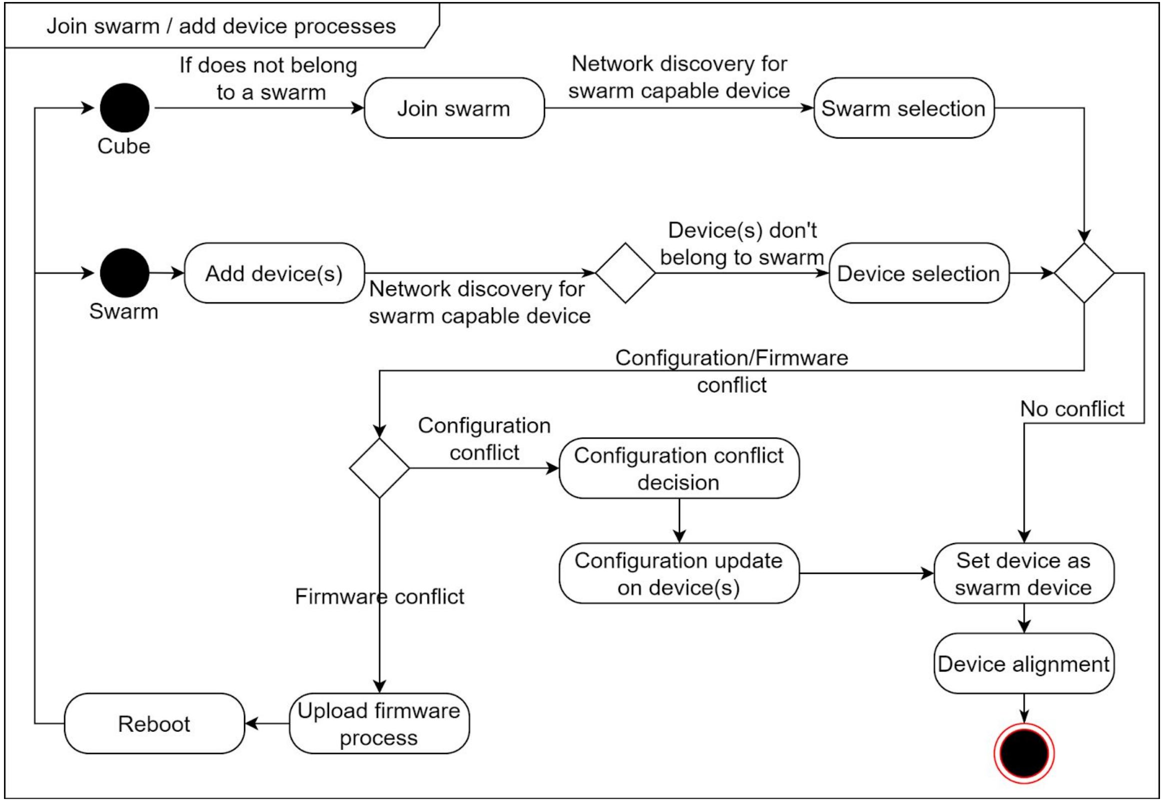


IPC: Inter Process Communication
RPC: Remote Procedure Call
DKV: Distributed Key-Value Store

Swarm workflows

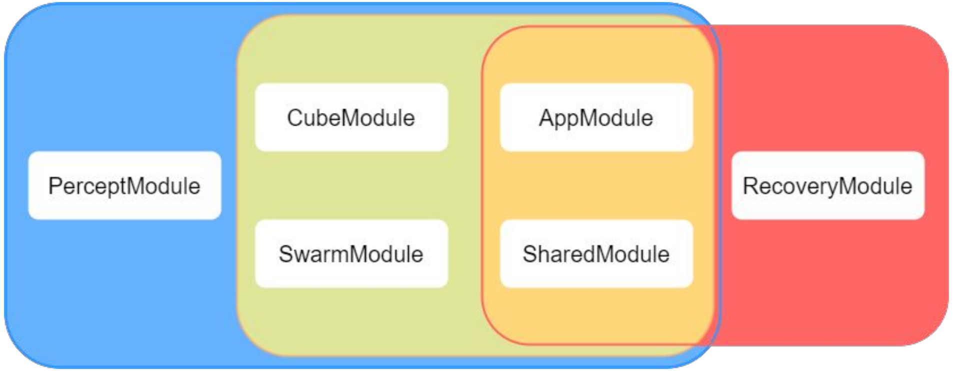

Swarm creation process
Device does not belong to a swarm
Create swarm
Network discovery for swarm capable device
Device selection
No conflict
Configuration/Firmware conflict
Configuration conflict
Configuration conflict decision
Configuration update on devices
Firmware conflict
Set device as swarm device
Device alignment
Set swarm name
Upload firmware process
Firmware installation
Reboot

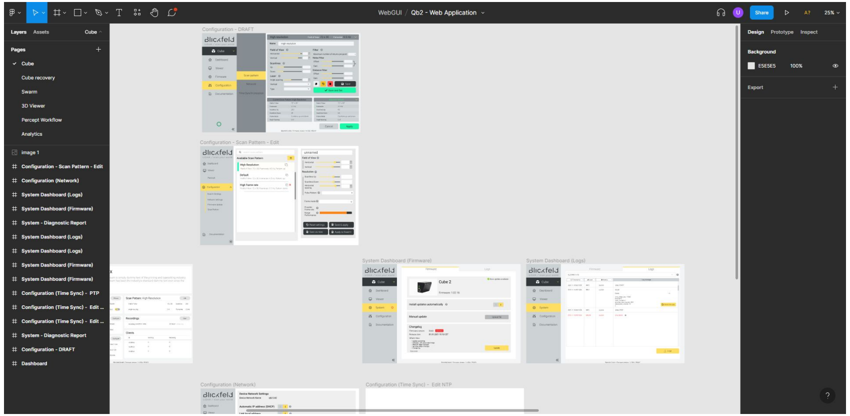
Swarm workflows


Join swarm / add device processes
If does not belong to a swarm
Join swarm
Network discovery for swarm capable device
Swarm selection
Cube
Swarm
Add device(s)
Network discovery for swarm capable device
Device(s) don't belong to swarm
Device selection
Configuration/Firmware conflict
No conflict
Configuration conflict
Configuration conflict decision
Configuration update on device(s)
Set device as swarm device
Device alignment
Firmware conflict
Upload firmware process
Reboot

Design overview: Web-GUI

- Design goals
  - Modular structure
  - Development rules
  - Plugin based architecture
- Design process
- User experience (UX)
- User interface (UI)



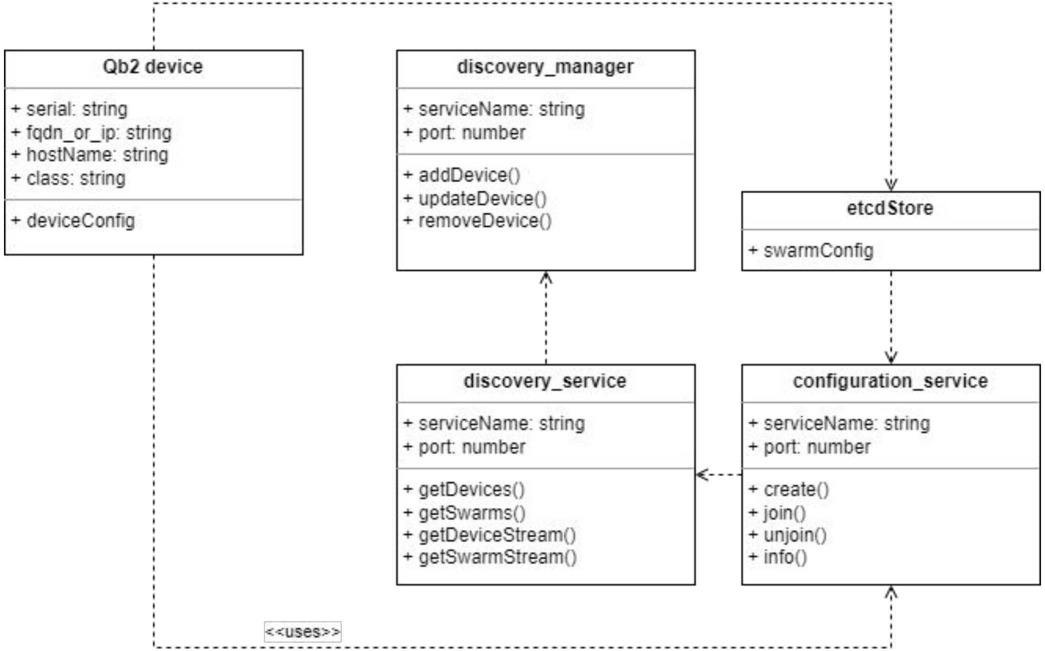

WebGUI / Qb2 - Web Application
Share
25%
Layers
Assets
Cube
Pages
Cube
Cube recovery
Swarm
3D Viewer
Percept Workflow
Analytics
image 1
Configuration - Scan Pattern - Edit
Configuration (Network)
System Dashboard (Logs)
System Dashboard (Firmware)
System - Diagnostic Report
System Dashboard (Logs)
System Dashboard (Logs)
System Dashboard (Firmware)
System Dashboard (Firmware)
Configuration (Time Sync) - PTP
Configuration (Time Sync) - Edit ...
Configuration (Time Sync) - Edit ...
System - Diagnostic Report
Configuration - DRAFT
Dashboard
Configuration - DRAFT
Configuration - Scan Pattern - Edit
System Dashboard (Firmware)
System Dashboard (Logs)
Configuration (Network)
Configuration (Time Sync) - Edit NTP
Design
Prototype
Inspect
Background
E5E5E5
100%
Export

Outline

- Motivation
- Overview
- Background
- Design
- **Implementation**
- Summary
- Future work

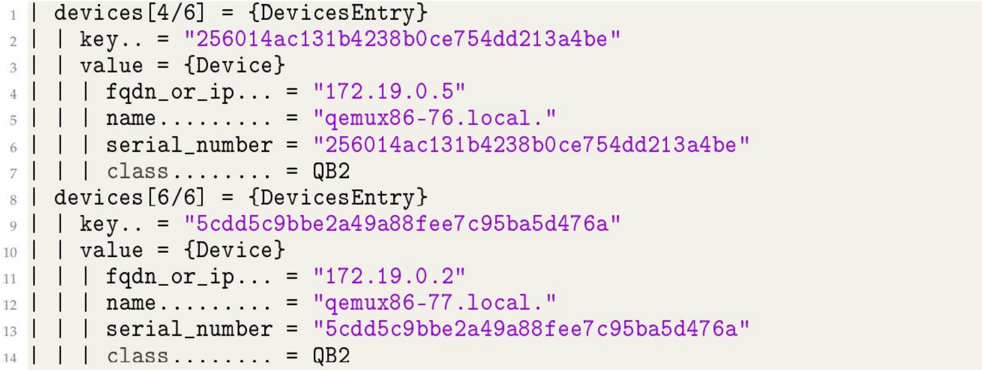
Implementation: Configuration synchronization

- Discovery service
    - Add device
    - Remove device
    - Update device
- Configuration service
    - Create
    - Join and Unjoin
    - Configuration synchronization



Result

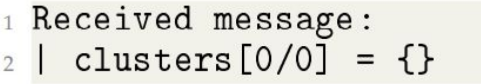
These results were obtained by using gWhisper.



```
| devices[4/6] = {DevicesEntry}
| | key.. = "256014ac131b4238b0ce754dd213a4be"
| | value = {Device}
| | | fqdn_or_ip... = "172.19.0.5"
| | | name......... = "qemux86-76.local."
| | | serial_number = "256014ac131b4238b0ce754dd213a4be"
| | | class........ = QB2
| devices[6/6] = {DevicesEntry}
| | key.. = "5cdd5c9bbe2a49a88fee7c95ba5d476a"
| | value = {Device}
| | | fqdn_or_ip... = "172.19.0.2"
| | | name......... = "qemux86-77.local."
| | | serial_number = "5cdd5c9bbe2a49a88fee7c95ba5d476a"
| | | class........ = QB2
```

List of devices on network: **blickfeld.swarm.services.Discovery GetDevices**

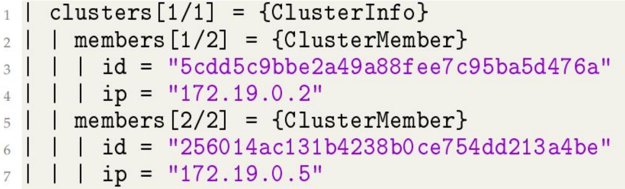
Discovery service returns all the Qb2 devices and Swarms that exist on the network



```
Received message:
| clusters[0/0] = {}
```

List of Swarms on network: **blickfeld.swarm.services.Discovery GetSwarms**

Using the **"blickfeld.swarm.services.Cluster Create"** service

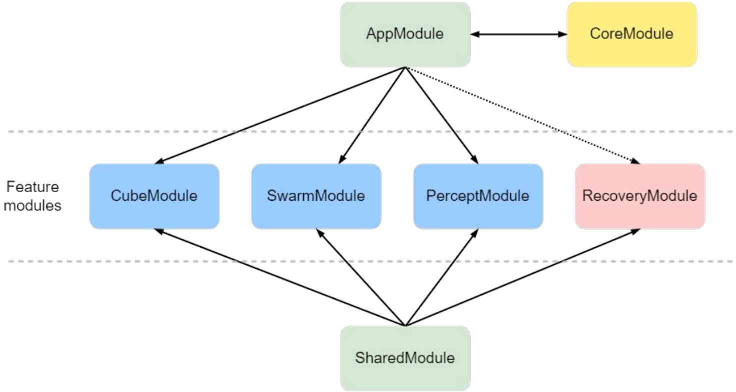


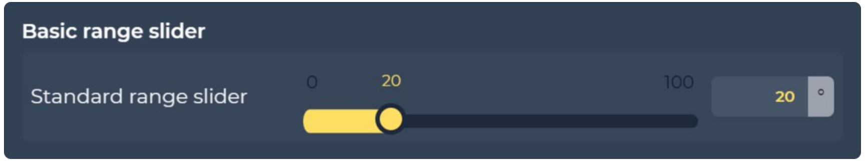
```
| clusters[1/1] = {ClusterInfo}
| | members[1/2] = {ClusterMember}
| | | id = "5cdd5c9bbe2a49a88fee7c95ba5d476a"
| | | ip = "172.19.0.2"
| | members[2/2] = {ClusterMember}
| | | id = "256014ac131b4238b0ce754dd213a4be"
| | | ip = "172.19.0.5"
```

List of Swarms on network after swarm is created

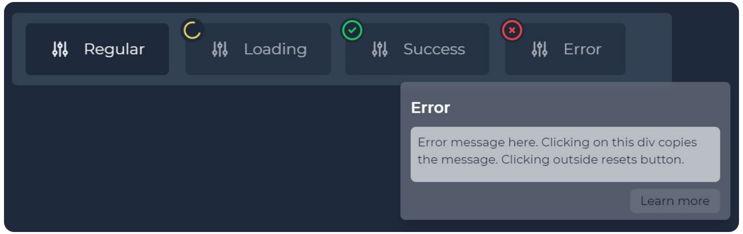
Implementation: WebGUI

- Technologies
  - Angular
  - Typescript
  - Tailwind CSS
- Connection
  - Websocket gRPC proxy
  - Routing
- Dynamic components




Basic range slider

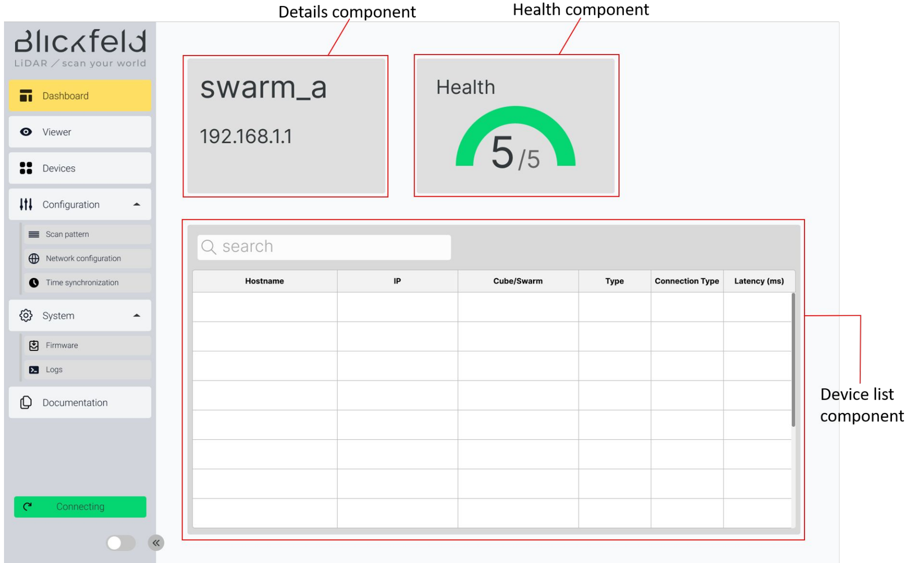
Standard range slider 20



Error

Error message here. Clicking on this div copies the message. Clicking outside resets button.

Result



Details component

Health component

Device list component

swarm_a

192.168.1.1

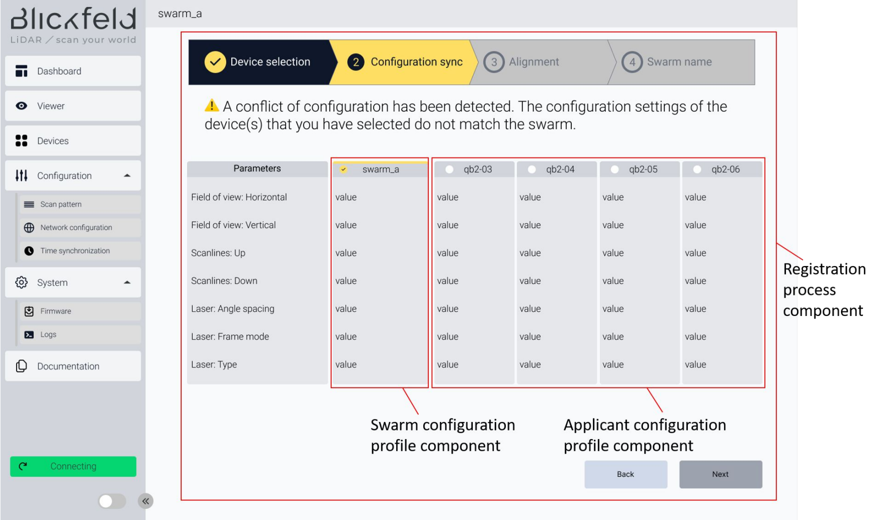
Health

5 /5

search

Hostname	IP	Cube/Swarm	Type	Connection Type	Latency (ms)

Swarm dashboard



Registration process component

Swarm configuration profile component

Applicant configuration profile component

⚠ A conflict of configuration has been detected. The configuration settings of the device(s) that you have selected do not match the swarm.

Parameters	☑ swarm_a	○ qb2-03	○ qb2-04	○ qb2-05	○ qb2-06
Field of view: Horizontal	value	value	value	value	value
Field of view: Vertical	value	value	value	value	value
Scanlines: Up	value	value	value	value	value
Scanlines: Down	value	value	value	value	value
Laser: Angle spacing	value	value	value	value	value
Laser: Frame mode	value	value	value	value	value
Laser: Type	value	value	value	value	value

Configuration conflict resolution

Outline

- Motivation
- Overview
- Background
- Design
- Implementation
- **Summary**
- Future work

Summary

The proposed concept of decentralized LiDAR system (Swarm) provides a scalable and flexible solution for combining outputs from multiple LiDAR devices

In this project the focus was on

- Designing initial architecture
- Discovery service
- Configuration service
- Modular web interface
- Dynamic components

Outline

- Motivation
- Overview
- Background
- Design
- Implementation
- Summary
- **Future work**

Future Work

- Combining point cloud data
- Perception processor

References

[1] D. Ongaro and J. Ousterhout. “In Search of an Understandable Consensus Algorithm (Extended Version).” In: Stanford University.

[2] B. GmbH. Qb2. https://www.blickfeld.com/lidar-sensor-products/qb2/