**Service Definitions** - Use of gRPC (protos)

3 different services/devices proto file

4 different RPC invocation styles

* simple RPC – energy efficiency (where the client **sends a single request** to the server and **gets** a **single response** back, just like a normal function call.)

The lights will be activated with motion, also its intensity will be adjusted according to the quantity of natural light present the data will be collected with sensors in the rooms.

Energy consumption will also be controlled by sensors. After office hours, in the rooms with no staff in it will have all devices considered non-essential turned off.

level of CO will also be controlled. After office hours, HVAC will be turned off in empty rooms.

* server-side streaming RPC where the client **sends** a **request** to the server and **gets** a stream to read a **sequence of messages** back

HVAC will be managed automatically with a pre-determinate temperature, Current temperature check and configuration can be done manually via application by the area manager or building control department.

* client-side streaming RPC where the client writes a **sequence of messages** and **sends** them to the server, again using a provided stream. Once the client has finished writing the messages, it waits for the server to read them and **return** its **response**

The buildings occupancy will be monitored when people pass the security gate with their badges, surveillance cameras in meeting rooms will work with motion detection.

* bidirectional streaming RPC where **both sides send a sequence of messages** using a read-write stream.

The list of visitors and people inside the building can be accessed via application.

**Service Implementations**. Implement 3 sufficiently complex service implementations

Server

Client

**Naming Services**. Use of jmDNS

Marks for registration 3 services

Marks for discovery 3 services

**Remote Error Handling**

appropriate error handling for remote invocations and error messaging

user input validation

**Client - Graphical User Interface (GUI)** (16%)

That allows to

view (e.g., present, discover),

control (parameters)

invoke the services/devices.

**Presentation/Viva**

**GitHub**