Project

This project is to be done in groups of two. Scoring will be individual, and each team member must write some of the code, (100% div 2 is 50%) and understand 100% of it.

Aims

The purpose of this project is to gain experience in designing and building distributed systems. You will be expected to demonstrate that your solution meets the objectives set out in this document.

The Problem

Develop a set of **protocols/messages** and build a reference implementation of devices in a smart environment. Your devices must publicise themselves and discover each other using **Zeroconf**, and they should communicate via **UDP, or TCP**. JSON is to be used for data exchange, GSON can be used for automated conversion to objects. **You can, at your discretion, implement communication between ONE service and its client using MQTT rather than direct TCP sockets.**

On connection they will swap service manifests that describe what services they perform and how to access these services.

Your team should begin by devising your own scenario. You can choose any smart environment. For example, a smart home, a smart office (printers, faxes, PCs), a smart meeting space (projector, laptops, PDAs, mobile phones) or a smart hospital (PDAs, table-style PCs, medicine containers, location trackers). In the above example each services provider i.e. projector can be thought of as an “entity”.

There must be a minimum of 4 separate services. It is key to specify what operations are supported on each “device”, for example it would be sensible for a projector to support and on/off function, a get state function, and a change input function (e.g. VGA, HDMI). Perhaps a media player function would support function to retrieve all playable files, and a corresponding play method, among others.

To demonstrate your implementation a simple GUI with devices and controllers should be built.

Deliverables

Report A short report which details the scenario, and services, you have chosen. Additionally this should specify the message formats for data exchange and service actuation. A template will be provided.

Program Code A NetBeans project, with all code, well commented. Code must also be available in a private GitHub repository, the repo must have a commit history, not a last minute code dump. If student x writes some code then x must submit the code.

Marking Scheme

- Report (10%)
- 4 sufficiently complex services (30%)
- Message format (20%)
- Use of jmDNS (20%)
- User Interface (10%)
- Error Handling (10%)