# BLE Mesh Demo

## Purpose

This document will explain the Mesh Demo and the motivations behind the design decisions.

## Architecture

The mesh network realized in this demo has the following points to show:

1. Create a mesh network with two groups of Thingy52s. The two groups are on the same Mesh network (common network key) but they cannot decode the other group’s messages because they have separate application keys. The purpose of creating and bifurcating two groups such as this is to simulate two separate drawers on a station.
2. Demonstrate one Thingy52 from one group to be able to “move” to the other group and be part of the new group. This would demonstrate the cubies switching drawers by wirelessly being reconfigured to belong in the new group.

The above two points should demonstrate isolation of traffic between two drawers while allowing for any cubie to be reconfigured for re-use with either drawer.

Below is a diagram of a node and the capabilities built into it via the firmware:

Please see BLE Mesh profile document for definitions of Elements.

1. Configuration Server
2. Health Server
3. Generic On/Off Server
4. Generic On/Off Client
5. Generic On/Off Server
6. Generic On/Off Client

Element 1

Element 2

In this demo, we only use the Generic On/Off Server on Element 1. In a mesh network, a server controls access to a resource. In this case, the resource is the LED. We will command the Generic On/Off Server via a Generic On/Off Client. In this case, the Client will be the same as the Provisioner but they do not have to be. In other words, another node in the mesh network with the proper network and application key can successfully command the server to switch the LED on and off. The Provisioner is a special node that assigns each node its role. So, in this instance, a provisioner assigns which application key and which group an unprovisioned node belongs to. The demo comes as pre-provisioned nodes, ready to run. The diagram below illustrates

## Provisioning

1. All nodes should be programmed with the firmware and they should come up blinking red:



1. Connect a nRF52-DK to the PC; launch SES with the project in <Mesh SDK 2.2>/examples/serial/. Open the main.c file and modify DSM\_DEVICE\_MAX to surpass or equal the maximum number of nodes you intend to provision using this nRF 52-DK provisioner.