Revised Disco Spec

This document outlines changes to align the Disco service with the ResolvedGroup and ResolvedService items used by the Wifi Direct and MDNS implementations.

The intention here is to use common naming for the fields that are common to reduce confusion. Whilst each protocol has their own implementation with additional fields, there is an interface defining the minimum requirements.

ResolvedGroup

```
public interface ResolvedGroup {
    void addService(ResolvedService resolvedService);
    boolean removeService(ResolvedService resolvedService);
    List<ResolvedService> getResolvedServices();
    String getGroupName();
    String getGroupUid();
}
```

Thus at a group level we have:

- **Group Name** - this is the Description seen in discovery
- **Group UID** - a generated UID
- A list of services

A service in this context is a peer. We don't need records for the additional peers in the same way as the other protocols as the peers files are available from the discovery service, so we don't pick an initial node for the java client to connect to - babble does all of that.

Resolved Service

```
    ResolvedGroup getResolvedGroup();
    void setResolvedGroup(ResolvedGroup resolvedGroup);
    String getAppIdentifier();
    String getGroupName();
    String getGroupUid();
    InetAddress getInetAddress();
    int getPort();
```

We will only have one service per group. There is no need for more as discussed above - the list is a result of mDNS where all the peers broadcast.

- **Group** - Links to parent group. No impact in this exercise
- **AppIdentifier** - FQDN, but we will set exactly as per other protocols.
- **GroupName** - Inherited from Group
- **GroupUID** - inherited from Group
- **InetAddress** - Not needed - will set to the Disco Server
- **Port** - Not needed - will set to the Disco Server Port.

Current Disco Structure

The current Structure is:

```
type group struct {
    ID           string        `json:"ID"`
    Title        string        `json:"Title"`
    Description  string        `json:"Description"`
    Peers        []*peers.Peer `json:"Peers"`
    GenesisPeers []*peers.Peer `json:"InitialPeers"`
}
```

Proposed Disco Structure

```
type group struct {
    GroupUID     string        `json:"GroupUID"`
    GroupName    string        `json:"GroupName"`
    PubKey       string        `json:"PubKey"`
    LastUpdate   int           `json:"LastUpdate"`
    Peers        []*peers.Peer `json:"Peers"`
    GenesisPeers []*peers.Peer `json:"InitialPeers"`
}
```

- **PubKey** is the public key of the group creator. This is not going to be used immediately - but it does pave the way for the signing of update groups
- **LastUpdate** is the last update time for this Group Record. It gives some indication of how stale the record is. We might not use it immediately, but as it is relatively simple to implement, it seems prudent to have that flexibility.

This means that we lose the description field, but we were not using it anyway.

WebRTCResolvedService Additional Fields

- **PubKey** - as above
- **LastUpdate** - as above