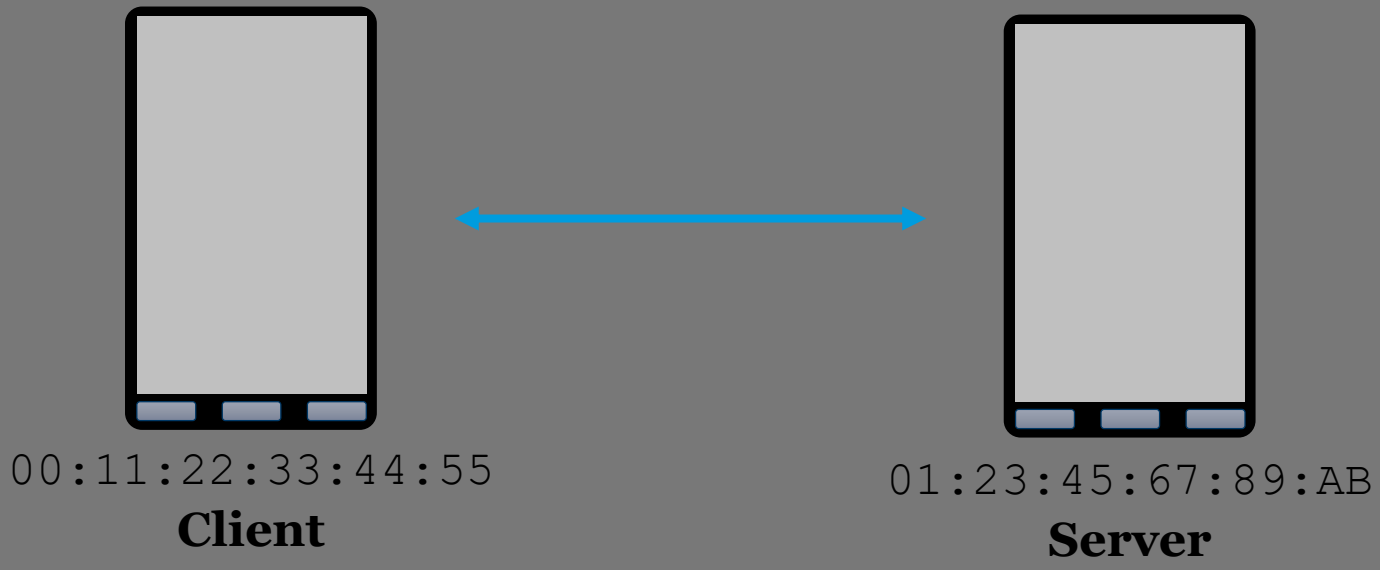
JÖNKÖPING UNIVERSITY

*School of Engineering*

# ANDROID BLUETOOTH CLASSIC

**Peter Larsson-Green**
Jönköping University
Spring 2020

# BLUETOOTH CLASSIC BASICS



# BLUETOOTH PERMISSIONS

- `android.permission.BLUETOOTH_ADMIN` (normal)
  - Enabling discovery & Search and pair with a new device.
- `android.permission.BLUETOOTH` (normal)
  - Connect and communicate with a paired device.
- `android.permission.ACCESS_COARSE_LOCATION` (dangerous)
  - Because of the first permission…
  - From Android 10 and one, instead use:
    - `android.permission.ACCESS_FINE_LOCATION` (dangerous)

# CHECK BLUETOOTH SUPPORT

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if(bluetoothAdapter == null){
    // Bluetooth not supported.
}
```

# ENABLING BLUETOOTH

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if(!bluetoothAdapter.isEnabled()){
    Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableBtIntent, 123);
}
```

```
public void onActivityResult(int requestCode, int resultCode, Intent data){
  if(requestCode == 123){
    if(resultCode == Activity.RESULT_OK){
      // Bluetooth is now enabled ☺
    }else{
      // Bluetooth is still disabled ☹
    }
  }
}
```

# ENABLING BLUETOOTH

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if(!bluetoothAdapter.isEnabled()){
   Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
   startActivityForResult(enableBtIntent, 123);
}
```

```
BroadcastReceiver receiver = new BroadcastReceiver{
   public void onReceive(Context context, Intent intent){
      int state = intent.getIntExtra(BluetoothAdapter.EXTRA_STATE);
      if(state == BluetoothAdapter.STATE_ON){ /* ☺ */ }
      else if(state == BluetoothAdapter.STATE_OFF){ /* ☹ */ }
   }
};
aContext.registerReceiver(receiver,
                          new IntentFilter(BluetoothAdapter.ACTION_STATE_CHANGED));
```

# THE SERVER

## Check if the device is discoverable.

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
int scanMode = bluetoothAdapter.getScanMode();
boolean isDiscoverable = scanMode == BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE;
```

## Making the device discoverable.

```
Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
intent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 60);
startActivityForResult(
```

```
public void onActivityResult(int requestCode, int resultCode, Intent data){
  if(requestCode == 111){
    if(resultCode == Activity.RESULT_OK){
      // Device is now discoverable ☺
    }else{
      // Device is still not discoverable ☹
    }
  }
}
```

# THE SERVER

## Check if the device is discoverable.

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
int scanMode = bluetoothAdapter.getScanMode();
boolean isDiscoverable = scanMode == BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE;
```

## Making the device discoverable.

```
Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
intent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 60);
startActivityForResult(
```

```
BroadcastReceiver receiver = new BroadcastReceiver{
    public void onReceive(Context context, Intent intent){
        int scanMode = intent.getIntExtra(BluetoothAdapter.EXTRA_SCAN_MODE);
        if(scanMode == BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE){ /* ☺ */ }
        else{ /* ☹ */ }
    }
};
aContext.registerReceiver(receiver,
                          new IntentFilter(BluetoothAdapter.ACTION_SCAN_MODE_CHANGED));
```

# THE SERVER

## Listening for incoming connections.

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
BluetoothServerSocket server = bluetoothAdapter.listenUsingRfcommWithServiceRecord(
  "The name",
  UUID.fromString("6a49c8a8-320c-42c4-afbf-8289c70c246d")
);
BluetoothSocket client = server.accept();
// ...
server.close();
```

# THE CLIENT

Discover and connect to the server.

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
bluetoothAdapter.startDiscovery();
BroadcastReceiver receiver = new BroadcastReceiver{
   public void onReceive(Context context, Intent intent){
      BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
      String name = device.getName();
      String macAddress = device.getAddress();
      bluetoothAdapter.cancelDiscovery();
      UUID service = UUID.fromString("6a49c8a8-320c-42c4-afbf-8289c70c246d");
      BluetoothSocket server = device.createRfcommSocketToServiceRecord(service);
      server.connect();
   }
};
aContext.registerReceiver(receiver, new IntentFilter(BluetoothDevice.ACTION_FOUND));
```

# SENDING/RECEIVING DATA

Each device has a `BluetoothSocket` representing the connection to the other device.

**Server**

```
BluetoothSocket client = ...;
String message = "Hello";
OutputStream outputStream = client.getOutputStream();
outputStream.write(message.getBytes(StandardCharsets.UTF_8));
```

**Client**

```
BluetoothSocket server = ...;
InputStream inputStream = server.getInputStream();
byte[] bytes = new byte[512];
int numberOfBytesRead = inputStream.read(bytes);
String message = new String(bytes, StandardCharsets.UTF_8)
```