ENTWICKLUNG EINER AUGMENTED REALITY APP FÜR IOS-GERÄTE ZUR BEDIENUNG VON IOT-GERÄTEN

BACHELORARBEIT | KOLLOQUIUM

MALTE JOSTEN | 3066184 | MALTE.JOSTEN@STUD.UNI-DUE.DE

```
$ ssh raspberrypi@196.114.2.168
```

🌐 196.114.2.168:12345/information

hohe Kompatibilität

intuitiv

AUGMENTED REALITY

mobil

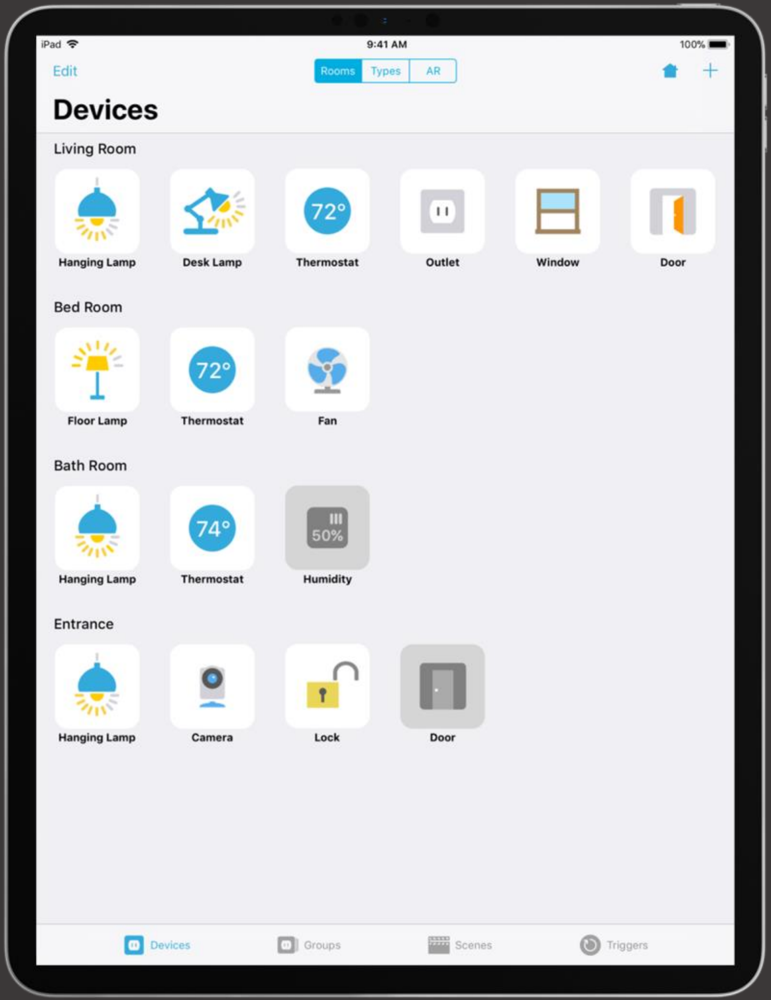
schnell/
effizient

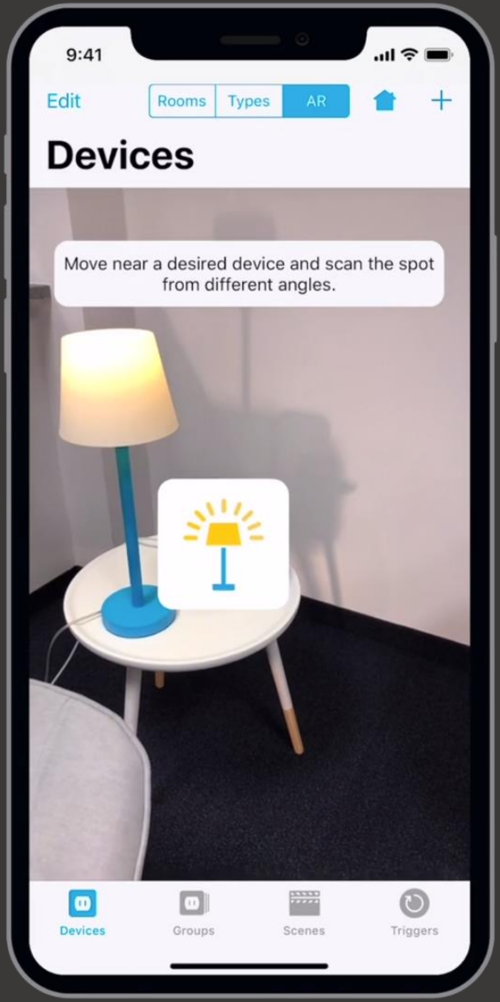
großes Potenzial

immersiv

DEVICES – CONTROL FOR HOMEKIT

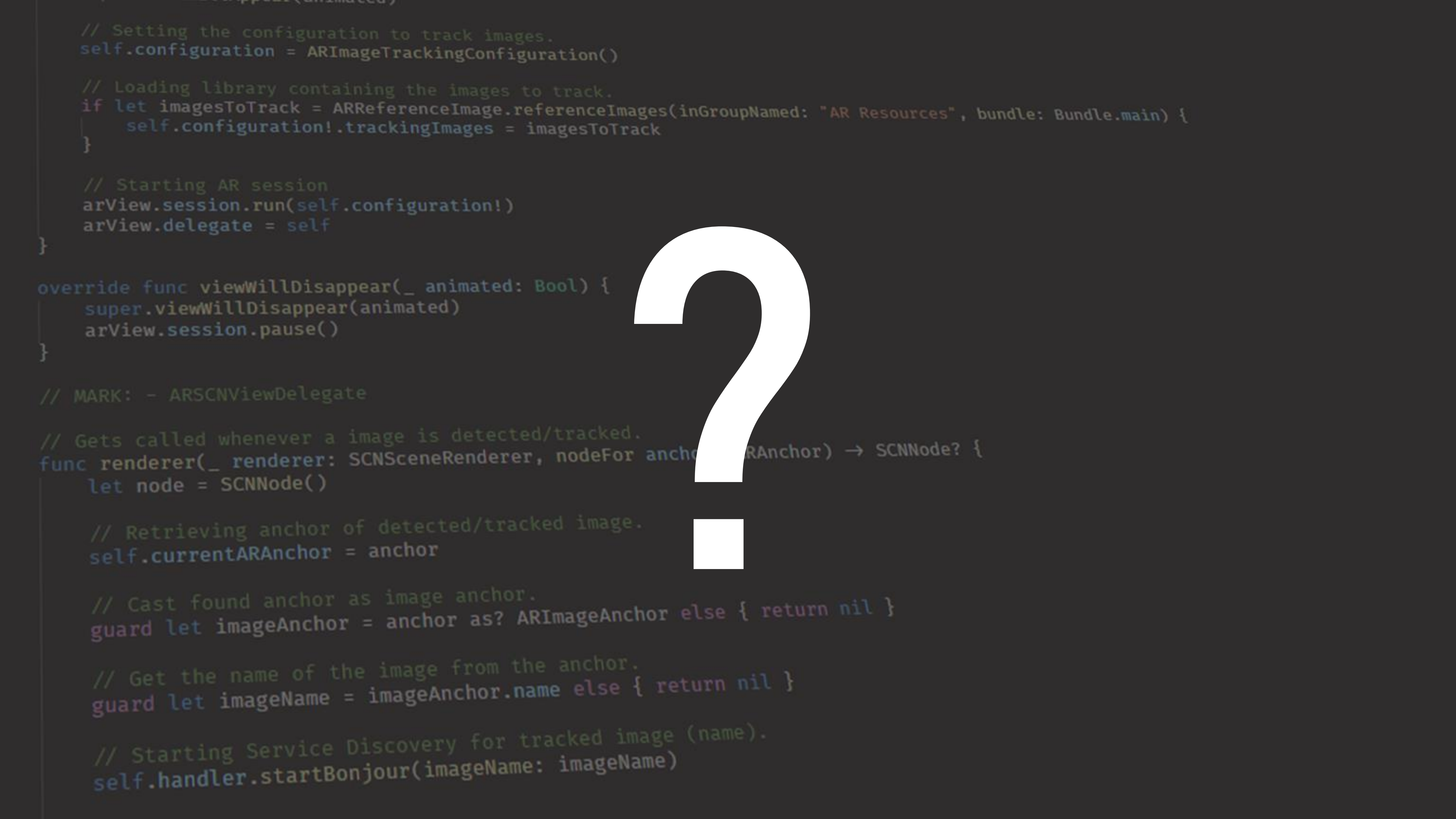
LINKDESK





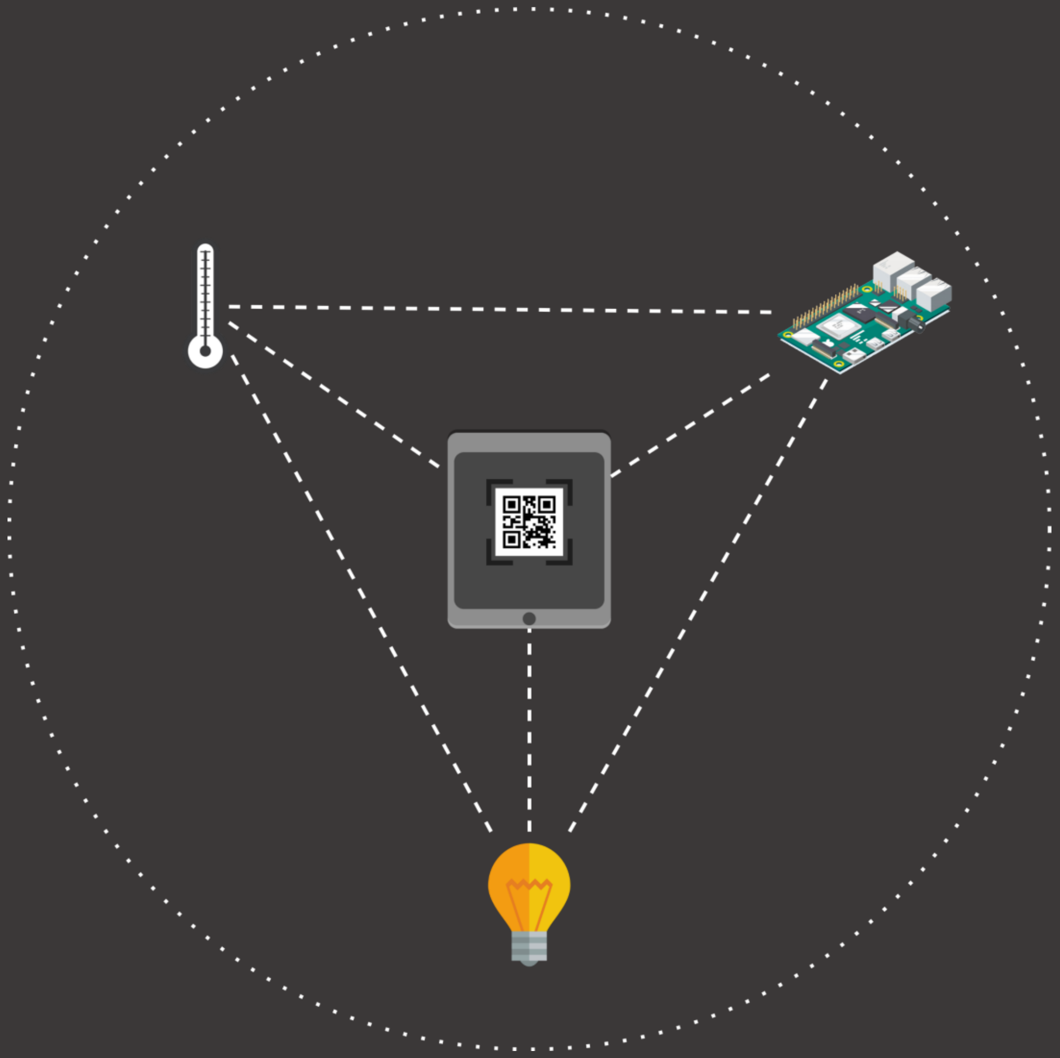
Move near a desired device and scan the spot from different angles.

ARIOT

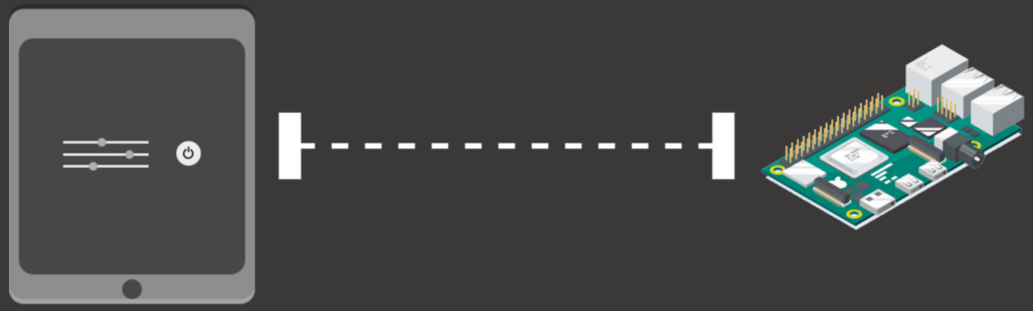


```
    // Setting the configuration to track images.
    self.configuration = ARImageTrackingConfiguration()

    // Loading library containing the images to track.
    if let imagesToTrack = ARReferenceImage.referenceImages(inGroupNamed: "AR Resources", bundle: Bundle.main) {
        self.configuration!.trackingImages = imagesToTrack
    }

    // Starting AR session
    arView.session.run(self.configuration!)
    arView.delegate = self
}

override func viewWillDisappear(_ animated: Bool) {
    super.viewWillDisappear(animated)
    arView.session.pause()
}

// MARK: - ARSCNViewDelegate

// Gets called whenever a image is detected/tracked.
func renderer(_ renderer: SCNSceneRenderer, nodeFor ancho RAnchor) → SCNNode? {
    let node = SCNNode()

    // Retrieving anchor of detected/tracked image.
    self.currentARAnchor = anchor

    // Cast found anchor as image anchor.
    guard let imageAnchor = anchor as? ARImageAnchor else { return nil }

    // Get the name of the image from the anchor.
    guard let imageName = imageAnchor.name else { return nil }

    // Starting Service Discovery for tracked image (name).
    self.handler.startBonjour(imageName: imageName)
```

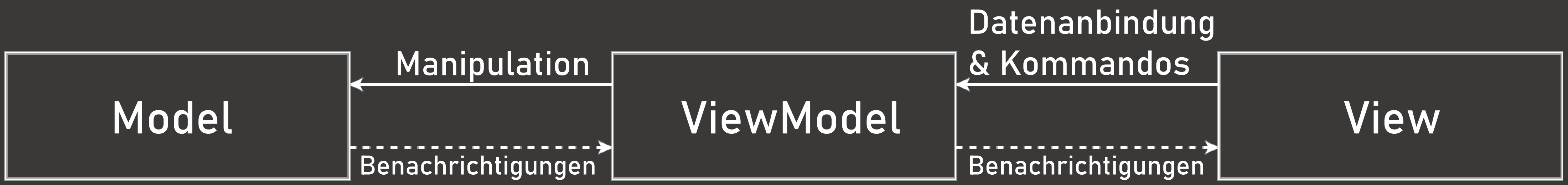
?



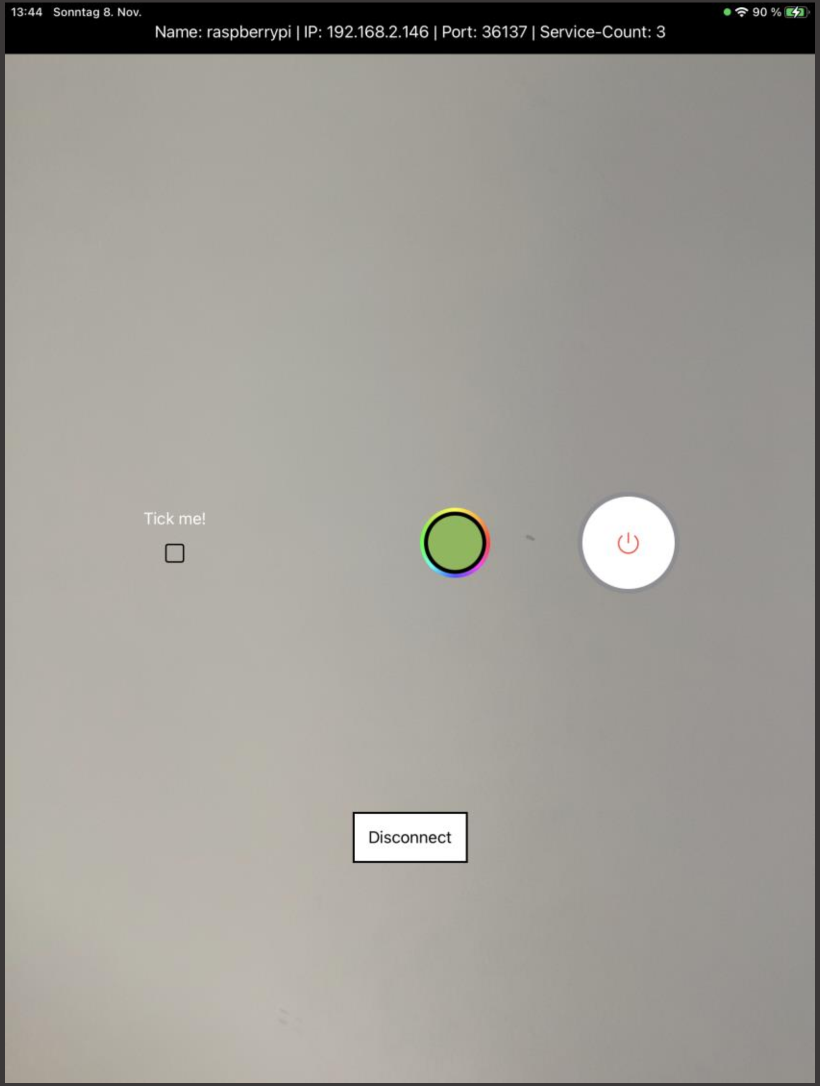
Service Discovery im lokalen Netzwerk



TCP-Verbindung zwischen Tablet und Raspberry Pi


Model
Manipulation
Benachrichtigungen
ViewModel
Datenanbindung & Kommandos
Benachrichtigungen
View

Bildschirm nach Starten der Anwendung



Name: raspberrypi | IP: 192.168.2.146 | Port: 36137 | Service-Count: 3

TCP-Verbindung zu IoT-Gerät aufgebaut

```xml
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name>raspberrypi</name>
  <service>
    <type>_http._tcp</type>
    <port>36137</port>
    <txt-record>running=true</txt-record>
    <txt-record>toggle=false</txt-record>
    <txt-record>colorpicker=00387A</txt-record>
  </service>
</service-group>
```

```
sudo java Main <Service-Datei Pfad> <Port> [<Inputs>]
```

0

…

Button

Checkbox

Colorpicker

Textfield

```
pi@raspberrypi $ sudo java dev.Main /etc/avahi/services/iot.service 0 -b -c -p
Using port 43691.
192.168.2.194 has connected.
[192.168.2.194]: Changed value of Service COLORPICKER to EC6C58 (old: FFFFFF).
[192.168.2.194]: Changed value of Service TOGGLE to true (old: false).
[192.168.2.194]: Changed value of Service TOGGLE to false (old: true).
[192.168.2.194]: Changed value of Service CHECKBOX to true (old: false).
[192.168.2.194]: Changed value of Service COLORPICKER to EB512E (old: EC6C58).
[192.168.2.194]: Changed value of Service COLORPICKER to 9CA435 (old: EB512E).
[192.168.2.194]: Changed value of Service COLORPICKER to 591D77 (old: 9CA435).
192.168.2.194 has disconnected.
```

KEY (Elementyp) = Value (Wert)

```
TOGGLE=true   TOGGLE:true
```

KEY ∈ {verfügbare Typen}

```
TOGGLE         LIGHT
```

{<,>} ∉ Value

```
TOGGLE=true</txt-record>
<service>
<type> tls. tcp</type>
<port>43152</port>
</service>
```

```
TOGGLE=true</txt-record>
<txt-record>foo=bar</txt-record>
```

Elementtyp **T**	valider Wert **W**
TOGGLE	$W \in \{"true", "false"\}$
CHECKBOX	$W \in \{"true", "false"\}$
COLORPICKER	Länge von $W == 6 \wedge W$ ist eine Hex-Zahl
TEXTFIELD	Keine Einschränkungen

!