



**Macoun**



# Zeroconf

## Automatische Erkennung & Konfiguration

DR. MICHAEL LAUER  
INFORMATION TECHNOLOGY



# Ein Fallbeispiel

# IP-Kamera im Heimnetz



Kaufen & Anschließen

# IP-Kamera im Heimnetz



IP-Kamera App öffnen – hier: Surveillance Pro

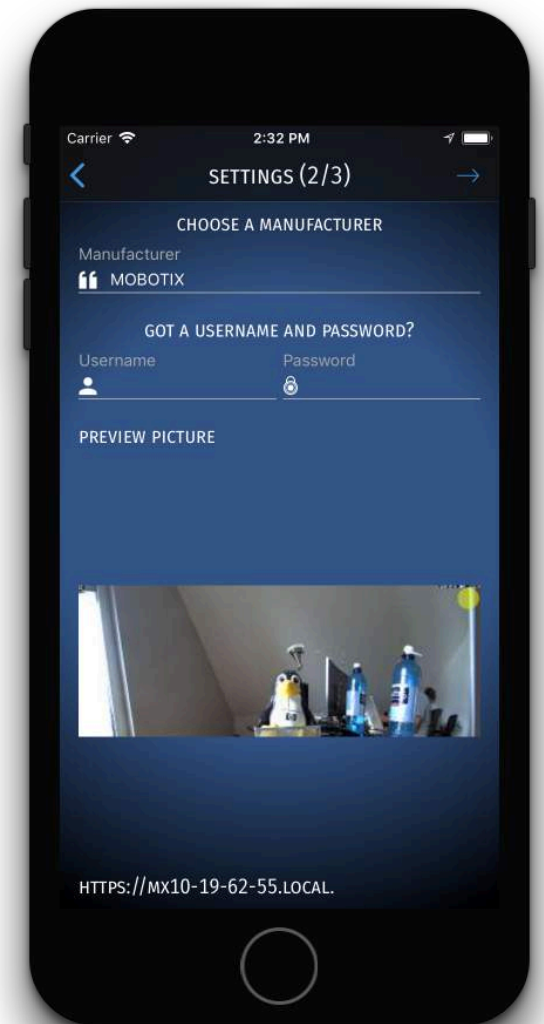


# IP-Kamera im Heimnetz



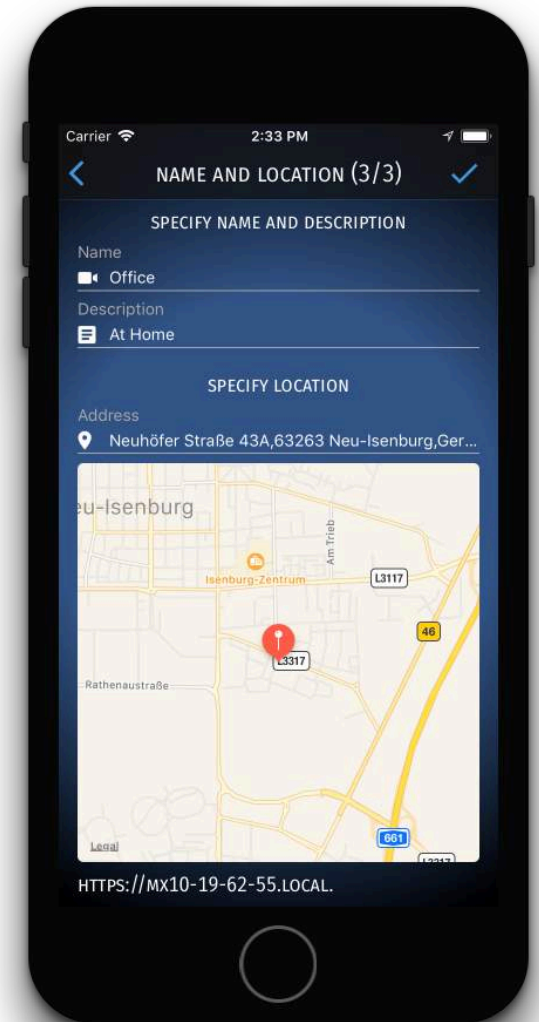
Kamera wird gefunden

# IP-Kamera im Heimnetz



Hersteller richtig erkannt

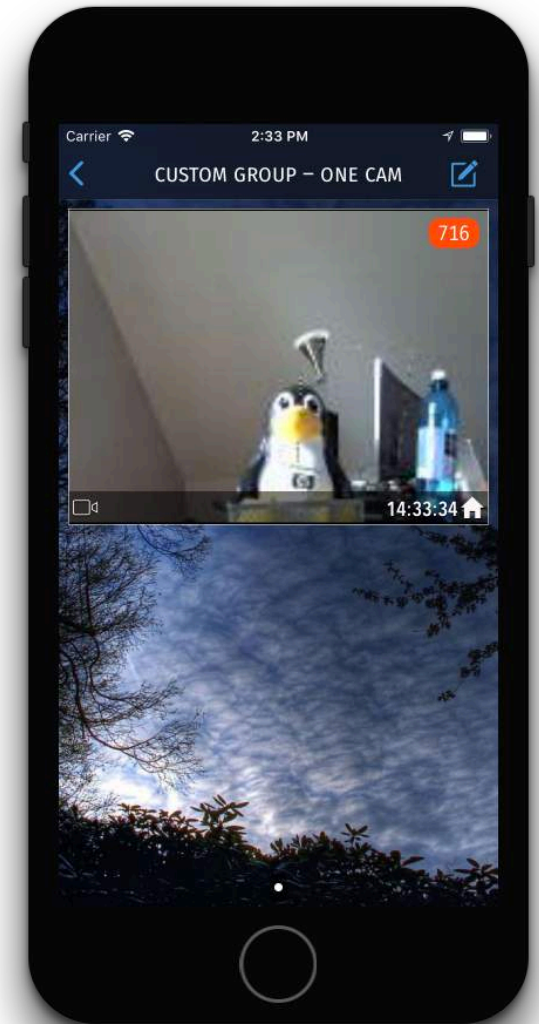
# IP-Kamera im Heimnetz



Metadaten anpassen



# IP-Kamera im Heimnetz



Fertig zur Benutzung

Terminologie  
Einordnung  
Abgrenzung

# Zeroconf

- „*Protokolle zur selbständigen Konfiguration von Rechnern in (spontan gebildeten) lokalen IP-basierten Rechnernetzen ohne Benutzung zentraler Komponenten*“
- Adressieren, Auffinden, Analysieren von Diensten
- IETF von 1999-2004, aufgelöst *mangels Konsens*
- RFC 3927 „Dynamic Configuration of IPv4 Link-Local Addresses“
- RFC 6763 „DNS-based Service Discovery“
- macOS, iOS, tvOS: Bonjour ([bɔ̃.ʒuʁ] bzw. [gu:'də])

# Terminologie

- Link-Lokale Adresse: Adresse, die **nicht** *gerouted* wird: 169.254.0.0/16
- .local: Spezielle Top-Level Domain für Zeroconf
- DNS: Domain Name System (Unicast)
- mDNS: Multicast DNS für die .local top-level domain
- dns-sd: DNS Service-Discovery
- SRV, PTR, TXT: {Service Resource | Pointer Resource | Text}-Record

# {icmp|tcp|http(s)} - Scan

- „Brute Force“ (Achtung: Firewalls, Intrusion Detection, DOS)
- Netzbereich pingen
- MAC-Adressen analysieren
- *Well-Known-Ports* abklopfen
- http(s)-Antworten analysieren
- Datenbanken

# Funktechnologien: btle & nfc

- Bluetooth Low Energy (aka BLE, Bluetooth 4.0)
  - ISM @ 2.4 GHz
  - macOS/ iOS: `CoreBluetooth.framework`
- Near Field Communication (aka NFC)
  - (HF)RFID @ 13.56 MHz
  - macOS / iOS: `CoreNFC.framework`

Worum es **nicht** gehen wird

HotPlug

Worum es noch **nicht** gehen wird

(Universal)  
Plug'n'Play



Worum es außerdem **nicht** gehen wird

DHCP, DNS

NETBIOS

AppleTalk

Bonjour en détail

# Bonjour en detail: Veröffentlichen



# Bonjour en detail: Veröffentlichen



Hat hier jemand  
die IP 169.254.0.42?



# Bonjour en detail: Veröffentlichen



IP

Niemand? Ok, dann  
nehme ich sie jetzt.



169.254.0.42

# Bonjour en detail: Veröffentlichen

mdnsd UDP:5353



IP

Niemand? Ok, dann  
nehme ich sie jetzt.



169.254.0.42

# Bonjour en detail: Veröffentlichen

mdnsd UDP:5353



IP

Hat jemand den Namen  
Tintenmeister.local?



169.254.0.42

# Bonjour en detail: Veröffentlichen

mdnsd UDP:5353



Nein? Na gut, dann  
nehme ich den jetzt.



IP  
Name

169.254.0.42

Tintenmeister.local



# Bonjour en detail: Veröffentlichen

printing TCP:1010

mdnsd UDP:5353



Nein? Na gut, dann  
nehme ich den jetzt.



IP  
Name

169.254.0.42

Tintenmeister.local

# Bonjour en detail: Veröffentlichen

printing TCP:1010

mdnsd UDP:5353



Jemand mit Druckerservice  
Konkret-Krasser-Drucker?



IP  
Name

169.254.0.42

Tintenmeister.local

# Bonjour en detail: Veröffentlichen

printing TCP:1010

mdnsd UDP:5353



Stille im Wald? Ok,  
mein Dienst dann.



IP 169.254.0.42  
Name Tintenmeister.local  
Dienst Konkret-Krasser-Drucker

# Bonjour en detail: Veröffentlichen

printing TCP:1010

mdnsd UDP:5353



Stille im Wald? Ok,  
mein Dienst dann.



SRV Konkret-Krasser-Drucker.\_printing.\_tcp.local. = Tintenmeister.local:1010

PTR \_printing.\_tcp.local. = Konkret-Krasser-Drucker.\_printing.\_tcp.local.

# Bonjour en detail: Auffinden

printing TCP:1010

mdnsd UDP:5353



Jemand mit  
Druckerdienst  
am Start?

←  
\_printing.\_tcp?



# Bonjour en detail: Auffinden

printing TCP:1010

mdnsd UDP:5353



Ich hab' hier

Konkret-Krasser-Drucker!

# Bonjour en detail: Auffinden

printing TCP:1010

mdnsd UDP:5353



Ok, wo läuft dieser  
Konkret-  
Krasser-  
Drucker?



# Bonjour en detail: Auffinden

printing TCP:1010

mdnsd UDP:5353



Läuft auf

Tintenmeister.local:1010!



# Bonjour en detail: Auffinden

printing TCP:1010

mdnsd UDP:5353



Welche Adresse hat  
Tintenmeister.local?



# Bonjour en detail: Auffinden

printing TCP:1010

mdnsd UDP:5353



162.254.0.42



# Bonjour en detail: Speichern

printing TCP:1010

mdnsd UDP:5353



Anwendungsspezifische  
Kommunikation  
gemäß Definition

~~\_printing.\_tcp~~

IP	<del>169.254.0.42</del>
Name	<del>Tintenmeister.local</del>
Dienst	Konkret-Krasser-Drucker

# Bonjour mit Foundation.framework

```
_browser = [[NSNetServiceBrowser alloc] init];
_browser.delegate = self;
[_browser searchForServicesOfType:@"_printing" inDomain:@"local"]

-(void)netServiceBrowser:(NSNetServiceBrowser *)browser didFindService:(nonnull
NSNetService *)service moreComing:(BOOL)moreComing
{
    NSUInteger row = _services.count;
    [_services addObject:service];
    NSIndexPath* ip = [NSIndexPath indexPathForRow:row inSection:0];
    [_tableView insertRowsAtIndexPaths:@[ip] withRowAnimation:Automatic];

    service.delegate = self;
    [service resolveWithTimeout:1.0];
}
```

# Bonjour mit Foundation.framework

```
-(void)netServiceDidResolveAddress:(NSNetService *)service
{
    if ( service.hostName.length )
    {
        NSArray<NSData*>* addresses = service.addresses;
        NSString* a = [service.formattedAddresses componentsJoinedByString:@" "];
        NSString* s = [NSString stringWithFormat:@"\n%@ \n%@", service.hostName, a];
        NSDictionary<NSString*,NSData*>* dict =
            [NSNetService dictionaryWithFromTXTRecordData:service.TXTRecordData];
    }
}
```

```
% dns-sd -B _services._dns-sd._udp
```

```
Browsing for _services._dns-sd._udp
```

```
DATE: ---Tue 03 Oct 2017---
```

```
17:18:29.099 ...STARTING...
```

Timestamp	A/R	Flags	if	Domain	Service Type	Instance Name
17:18:29.099	Add	2	6	.	_tcp.local.	_apple-mobdev2
17:18:29.198	Add	3	6	.	_udp.local.	_sleep-proxy
17:18:29.198	Add	3	6	.	_tcp.local.	_airport
17:18:29.198	Add	3	6	.	_tcp.local.	_afpovertcp
17:18:29.198	Add	3	6	.	_tcp.local.	_smb
17:18:29.198	Add	3	6	.	_tcp.local.	_adisk
17:18:29.198	Add	3	6	.	_tcp.local.	_http
17:18:29.198	Add	3	6	.	_tcp.local.	_https
17:18:29.198	Add	3	6	.	_udp.local.	_ntp
17:18:29.198	Add	3	22	.	_tcp.local.	_apple-mobdev2
17:18:29.198	Add	3	6	.	_tcp.local.	_axis-video
17:18:29.198	Add	3	6	.	_tcp.local.	_rtsp
17:18:29.198	Add	3	6	.	_tcp.local.	_touch-able
17:18:29.198	Add	3	6	.	_tcp.local.	_airplay
17:18:29.198	Add	3	6	.	_tcp.local.	_raop
17:18:29.198	Add	3	6	.	_tcp.local.	_mediaremotetv
17:18:29.198	Add	2	6	.	_tcp.local.	_homekit

```
% dns-sd -B _http._tcp
```

```
Browsing for _http._tcp
```

```
DATE: ---Tue 03 Oct 2017---
```

```
17:20:18.915 ...STARTING...
```

```
Timestamp      A/R      Flags   if Domain
```

```
17:20:18.915  Add           3    6 local.
```

```
17:20:18.916  Add           2    6 local.
```

```
Service Type
```

```
_http._tcp.
```

```
_http._tcp.
```

```
Instance Name
```

```
mx10-19-62-55
```

```
AXIS M1054
```

```
% dns-sd -L "mx10-19-62-55" _http._tcp
```

```
Lookup mx10-19-62-55._http._tcp.local
```

```
DATE: ---Tue 03 Oct 2017---
```

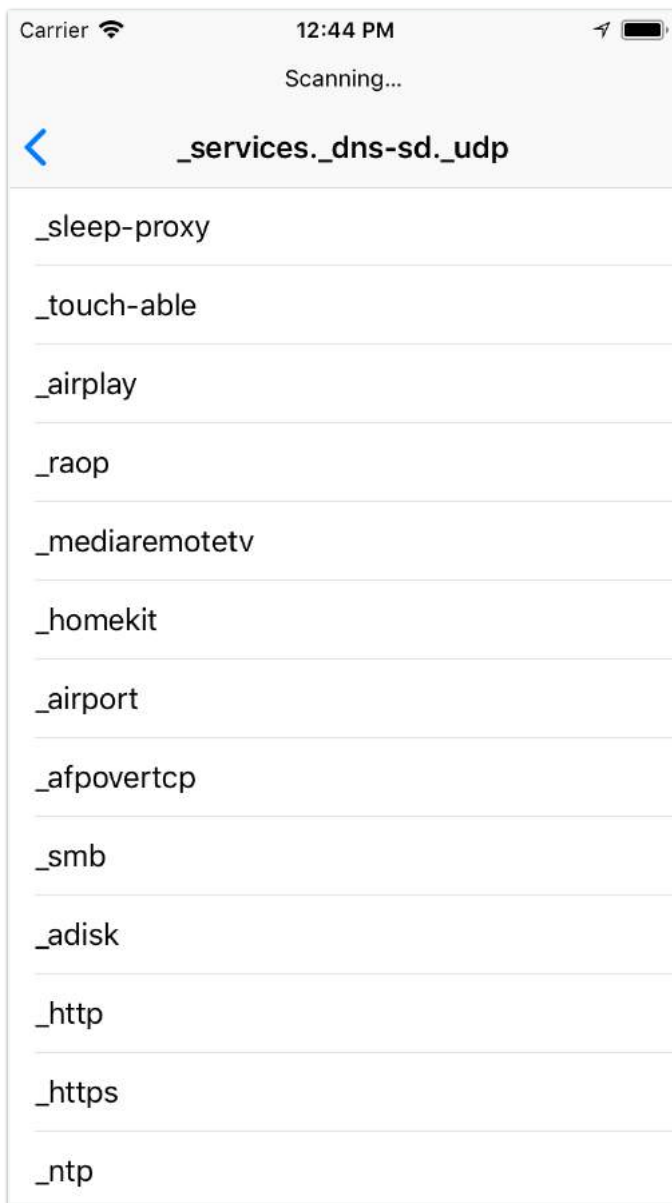
```
17:20:30.082 ...STARTING...
```

```
17:20:30.267 mx10-19-62-55._http._tcp.local. can be reached at mx10-19-62-55.local.:80  
(interface 6)
```

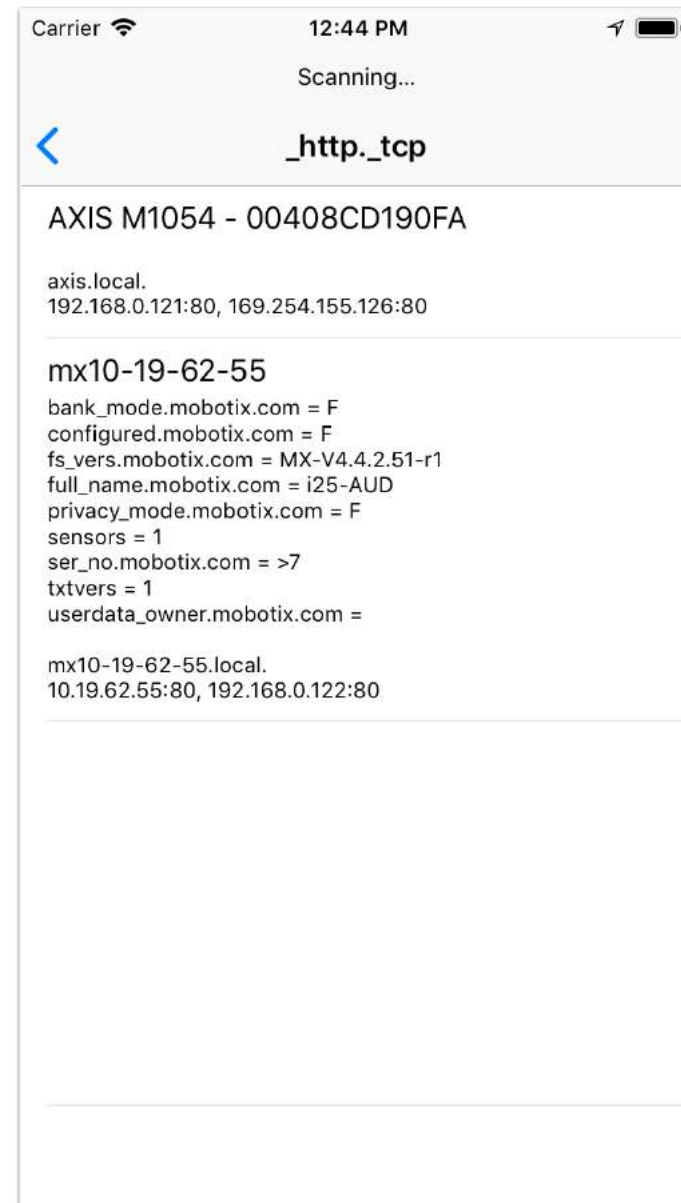
```
txtvers=1 ser_no.mobotix.com=\\x13\>7 full_name.mobotix.com=i25-AUD
```

```
fs_vers.mobotix.com=MX-V4.4.2.51-r1 bank_mode.mobotix.com=F privacy_mode.mobotix.com=F
```

```
userdata_owner.mobotix.com= configured.mobotix.com=F sensors=1
```



# Demo





{icmp|tcp|http(s)} - Scan

# {icmp|tcp|http(s)} - Scan

- Nicht alle Endgeräte implementieren Zeroconf
- Alternative: Subnetz durchsuchen – leider nicht mit Bordmitteln
  - MMLanScan – <https://github.com/mavris/MMLanScan>
- MAC-Adressen analysieren
  - Eigene MAC nicht mehr seit iOS 7
  - Keine MAC seit iOS 10

# Well-known-ports abklopfen

- POSIX

```
int connect(int socket, const struct sockaddr *address, socklen_t  
address_len);
```

- CoreFoundation

```
CFStreamCreatePairWithSocketToCFHost(CFAllocatorRef alloc,  
CFHostRef host, SInt32 port, CFReadStreamRef _Nullable  
*readStream, CFWriteStreamRef _Nullable *writeStream);
```

- Foundation

```
-(NSURLSessionStreamTask *)streamTaskWithHostName:  
(NSString *)hostname port:(NSInteger)port;
```

# http(s)-Antworten analysieren

```
% http HEAD https://www.heise.de
```

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 9
Cache-Control: public, max-age=32
Connection: keep-alive
Content-Encoding: gzip
Content-Length: 72502
Content-Type: text/html; charset=UTF-8
Date: Tue, 03 Oct 2017 16:02:28 GMT
Expires: Tue, 03 Oct 2017 16:02:52 GMT
Last-Modified: Tue, 03 Oct 2017 16:02:20 GMT
Server: nginx
Vary: Accept-Encoding,X-Forwarded-Proto,User-Agent,X-Export-Format,X-Export-Agent
```

# Demo

Telekom.de 15:38 46 %	
Idle	
<	Subnet Scan
mphone-7	192.168.0.221
switch.office.drlauer-research.com	192.168.0.1
apollo	192.168.0.3
mf8450.office.drlauer-research.com	192.168.0.102
ipmi.office.drlauer-research.com	192.168.0.103
dx600.office.drlauer-research.com	192.168.0.104
helios.office.drlauer-research.com	192.168.0.110
mobotix.office.drlauer-research.com	192.168.0.122
axis.office.drlauer-research.com	192.168.0.121
Unknown	192.168.0.155

**BTLE & NFC**

# Bluetooth 4.0 (BLE, Bluetooth Smart)

- Drahtloser Kommunikationsstandard für spontan vernetzte Systeme
- Reichweite ~10m, Datendurchsatz 2kbit - 8kbit
- 2006 entwickelt von Nokia als „Wibree“
- 2010 assimiliert von der BTSIG als „Bluetooth 4.0“
- 2011 erstmals eingesetzt in Volumenprodukt (iPhone 4S)
- Schnelle Verbindung, langsame Datenübertragung
- Batterieeffizient, flexibel und interoperabel

# Bluetooth 4.0 Kernkonzepte

- Verbindungsorientiert: Initiator („Central“) und Rezipient („Peripheral“)
- Geräte implementieren beliebig viele Dienste („Service“)
- Services enthalten beliebig viele Werte („Characteristic“)
- Ein Peripheral schickt „Beacons“ mit „Advertising Data“  
Auch hier wieder *Well-Known-Services*



# CBCentralManager, <CBCentralManagerDelegate>

```
_central = [[CBCentralManager alloc] initWithDelegate:self queue:nil];
[_central scanForPeripheralsWithServices:nil options:nil];

-(void)centralManager:(CBCentralManager *)central didDiscoverPeripheral:
(CBPeripheral *)peripheral advertisementData:(NSDictionary<NSString *, id>
*)advertisementData RSSI:(NSNumber *)RSSI
{
    [peripheral.delegate = self];
    [_central connectPeripheral:peripheral options:nil];
}

-(void)centralManager:(CBCentralManager *)central didConnectPeripheral:(CBPeripheral
*)peripheral
{
    [peripheral discoverServices:nil];
}
```

# CBPeripheral, <CBPeripheralDelegate>

```
-(void)peripheral:(CBPeripheral *)peripheral didDiscoverServices:(nullable NSError *)error
{
    for ( CBService* service in peripheral.services )
        [peripheral discoverCharacteristics:nil forService:service];
}

-(void)peripheral:(CBPeripheral *)peripheral didDiscoverCharacteristicsForService:
(CBService *)service error:(nullable NSError *)error
{
    for ( CBCharacteristic* characteristic in service.characteristics )
    {
        [characteristic discoverDescriptorsForCharacteristic:characteristic];
        if ( characteristic.properties & CBCharacteristicPropertyRead )
            [peripheral readValueForCharacteristic:characteristic];
    }
}
```

# NFC (Near Field Communication)

- Drahtlose Kurzstrecken-Funktechnik (üblicherweise  $\leq 10\text{cm}$ )
- 13.556MHz auf ISO/IEC 1800-3 mit 106kbit/s - 424kbit/s
- 1983: Charles Walton patentiert RFID
- 1997: Erste kommerzielle Verwendung in StarWars-Action-Figuren
- 2004: Nokia, Philips & Sony etablieren NFC-Forum
- 2014: Apple baut NFC ein für Apple Pay
- 2015: Android baut NFC ein für Android Pay

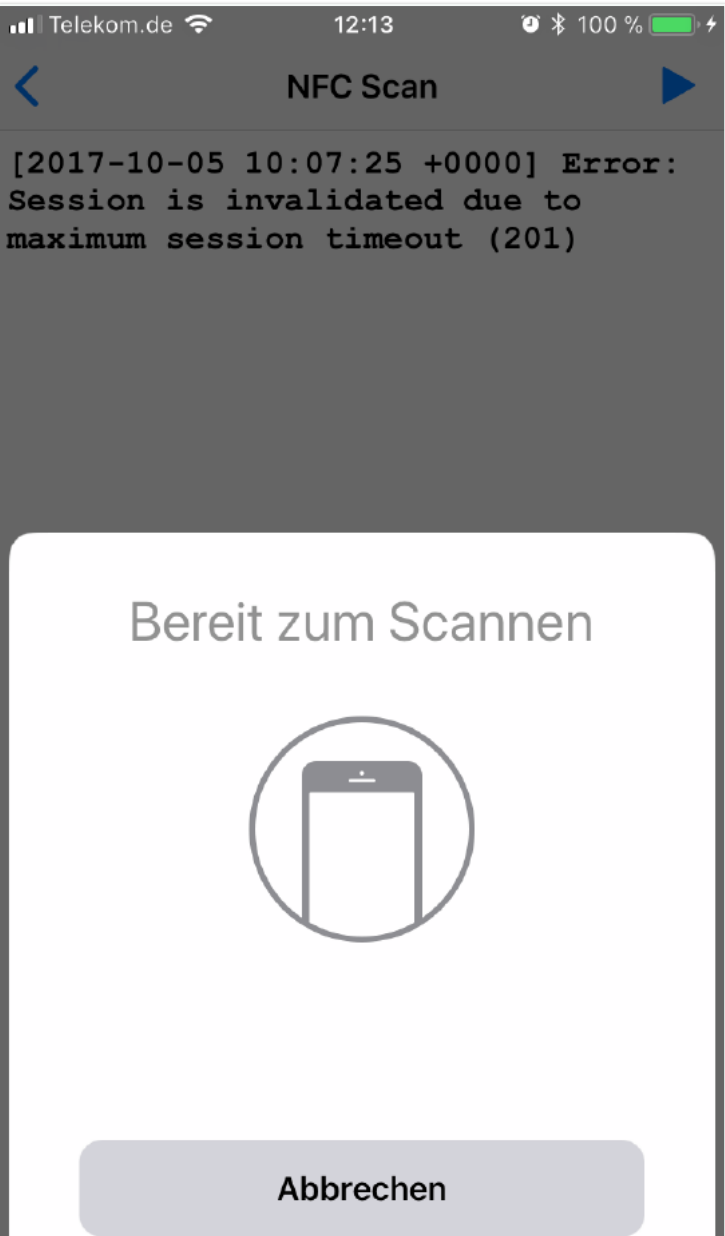
# NFC Kernkonzepte

- Aktive vs. Passive Tags
- Nur Lesbare vs. Wiederbeschreibbare Tags
- (Aktiver) Initiator baut Radiowellenfeld auf, um Target zu lesen
- Zwei aktive Tags könn(t)en bidirektional kommunizieren

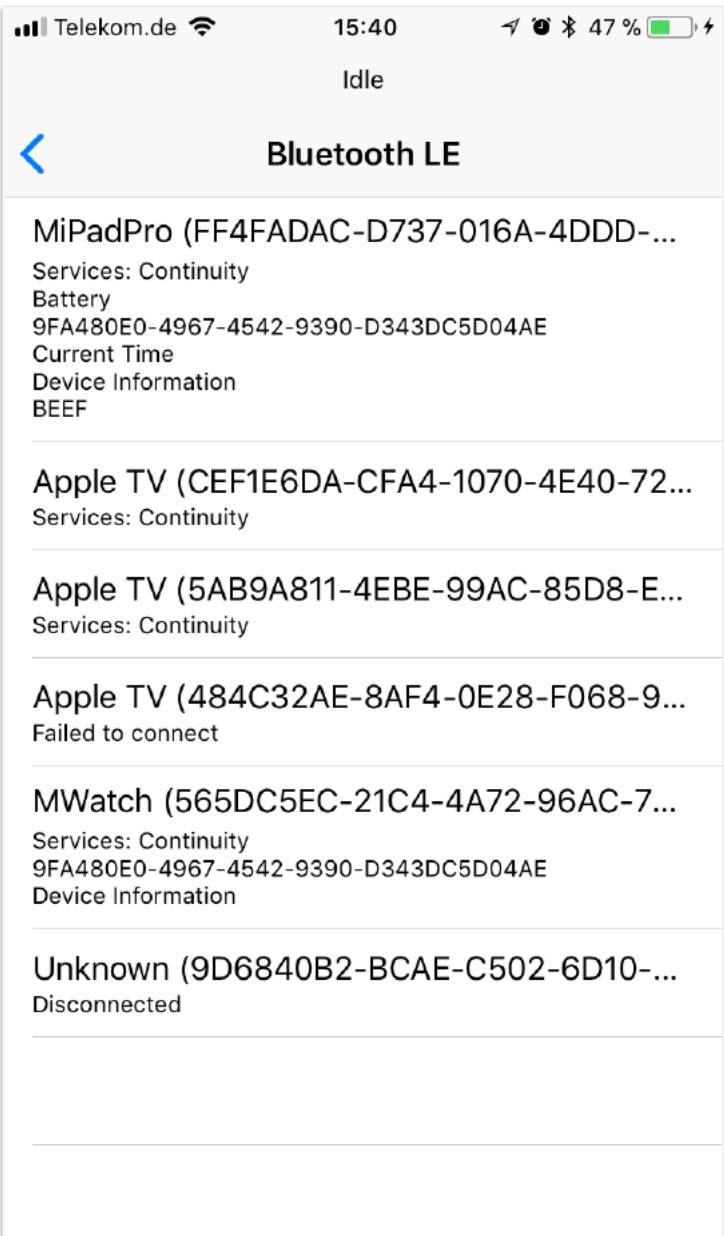
# NFCNDEFReaderSession, <NFCNDEFReaderSessionDelegate>

```
dispatch_queue_t q = dispatch_queue_create( "nfc.reader.demo",
DISPATCH_QUEUE_CONCURRENT );
_session = [[NFCNDEFReaderSession alloc] initWithDelegate:self queue:q
invalidateAfterFirstRead:NO];
[_session beginSession];

-(void)readerSession:(nonnull NFCNDEFReaderSession *)session didDetectNDEFs:(nonnull
NSArray<NFCNDEFMessage *> *)messages
{
    for (NFCNDEFMessage *message in messages)
    {
        for (NFCNDEFPayload *payload in message.records)
        {
            NSLog(@"Payload: %@", payload);
        }
    }
}
```



# Demo



# Zusammenfassung

- Wann und Wofür
- Bonjour ist Zeroconf
- Subnet-Scan, wenn Zeroconf nicht reicht
- Weitergehende Analyse gefundener Geräte
- Funktechnologien als Ergänzungen oder Alternative

# Links

- <https://tools.ietf.org/html/rfc3927>
- <https://tools.ietf.org/html/rfc6763>
- <https://www.avahi.org>
- <https://developer.apple.com/bonjour/>
- <http://www.dns-sd.org/servicetypes.html>
- <https://github.com/mavris/MMLanScan>
-





# Zeroconf

## Automatische Erkennung & Konfiguration

DR. MICHAEL LAUER  
INFORMATION TECHNOLOGY





**Macoun**