blucat

Joseph Paul Cohen
http://blucat.sf.net
The wireless future is here now!

Overview

- Streams (w/jokes)
- blucat inline netcat replacements
- blucat as Bluetooth nmap
- rfcomm and l2cap basics
- look at some devices
- how to prototype
- blucat architecture
- JSR-82 Basics

Get started!

Get blucat source:

git clone https://github.com/ieee8023/blucat

or to just run blucat on mac:

brew install blucat

Get demo android app:

https://github.com/ieee8023/blucat-android-remote

USB Adaptors that work great!

AirCable Host XR3

http://www.aircable.net/products/host-xr3.php

Plugable BT4LE

http://plugable.com/products/usb-bt4le

plugable
Dual Mode
Bluetooth® USB Adapter
Bluetooth 4.0 LE (Low Energy) and Classic Bluetooth Compatible
plugable
• Low Power Consumption • Small Size • Compatible

Questions for you

How many of you have:

Used a Bluetooth API?

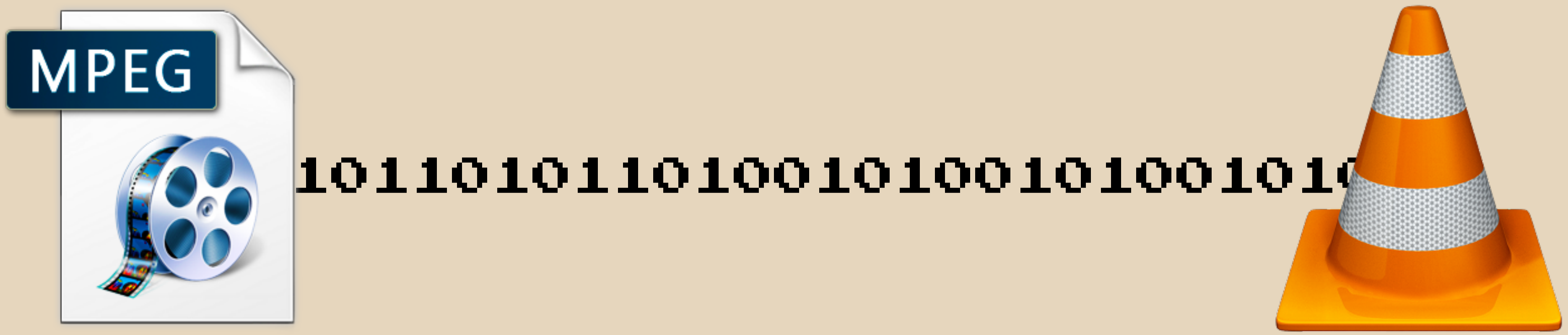
Used netcat to talk to a webserver?

Created outrageously complex Bash scripts that involved piping?

STREAMS==AWESOME

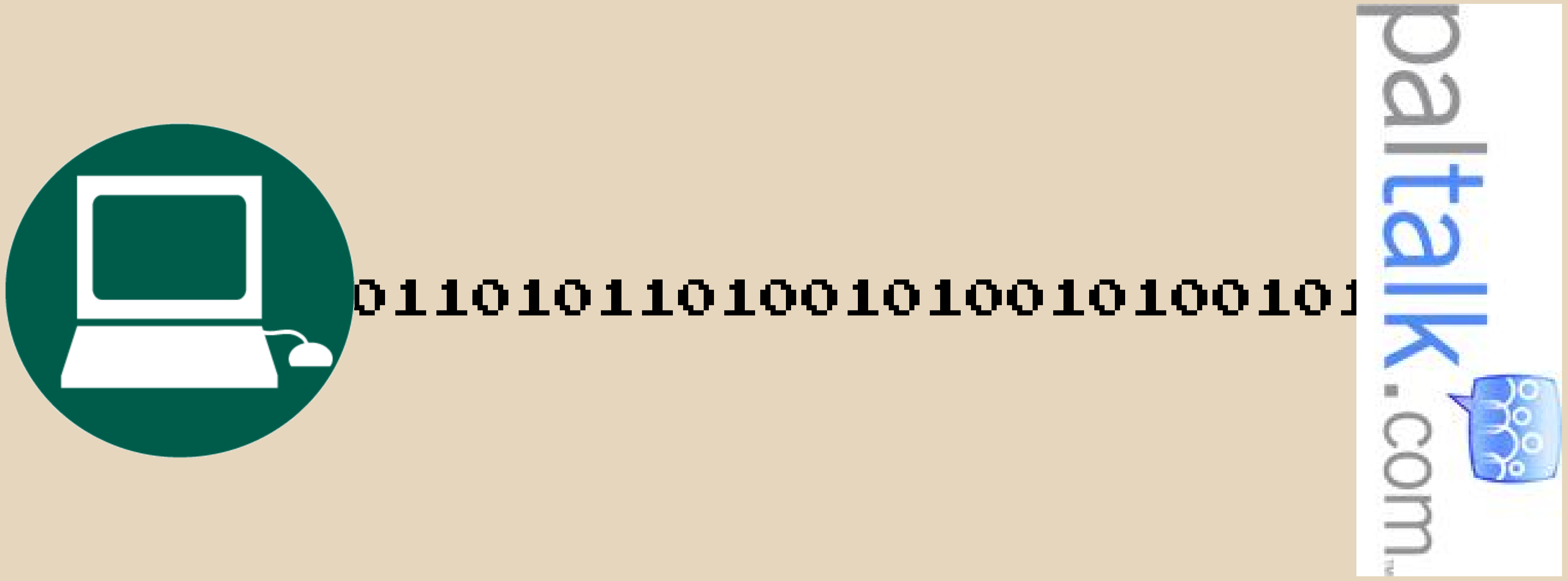
1001011010110100100100101

STREAMS==AWESOME



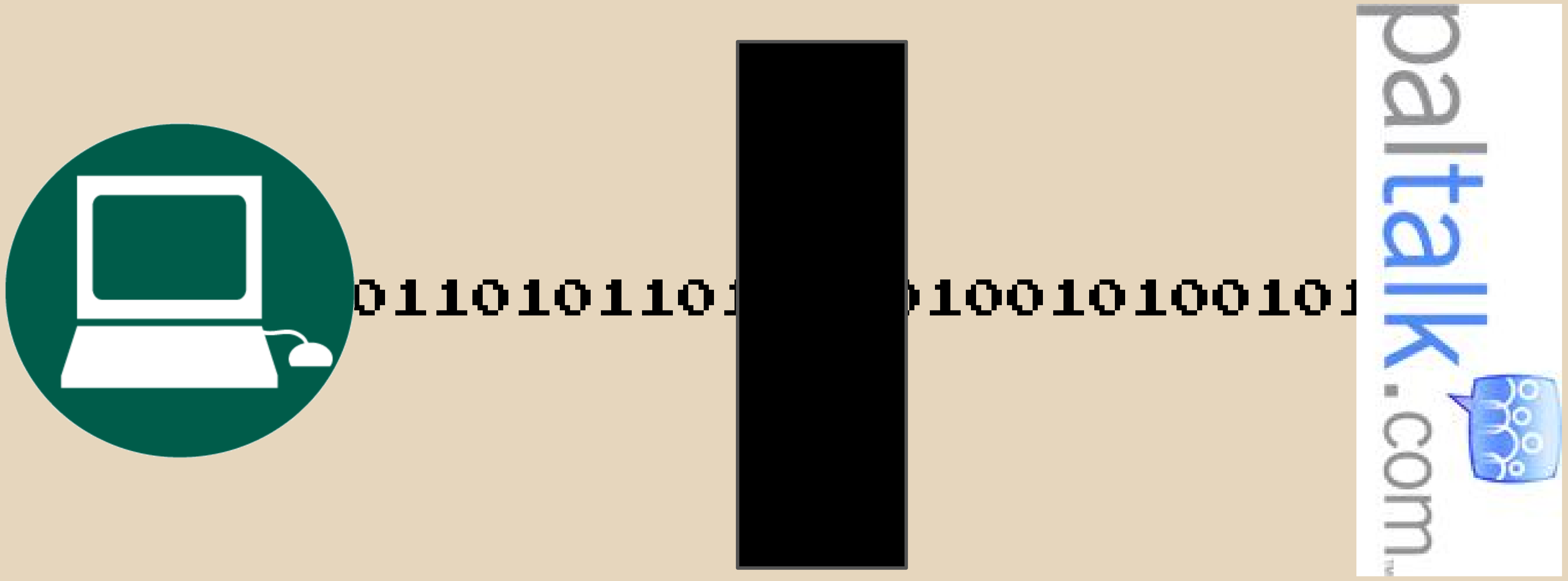
You can send files or data

STREAMS==AWESOME



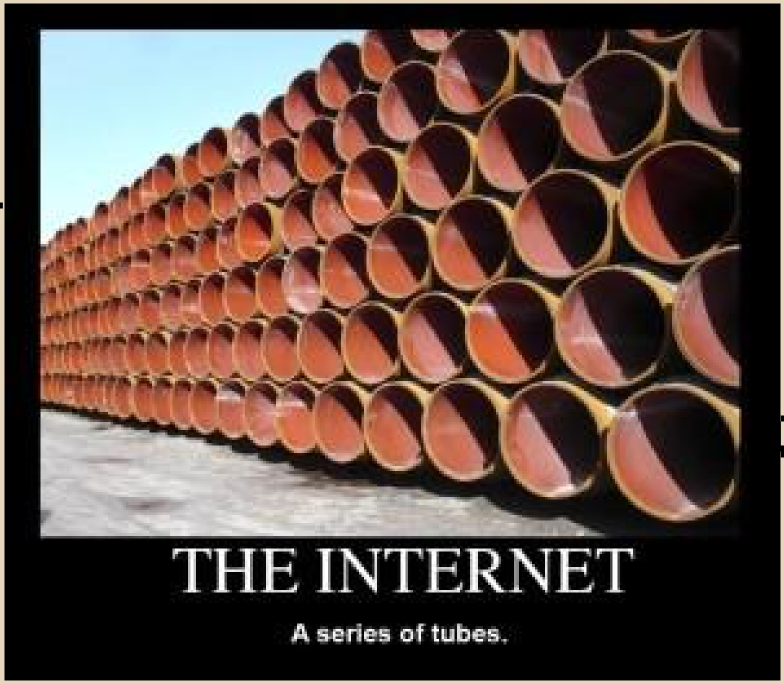
They even connect us all to PalTalk!

STREAMS==AWESOME



And it's all abstracted so each side just sees bits

01001011000 1



0001010010101

You can abstract a really complicated process this way

010010110001011

1000101001010101

And then ignore how complicated and dysfunctional they are

This works great for the TCP/IP

Why?

Let's look at HTTP

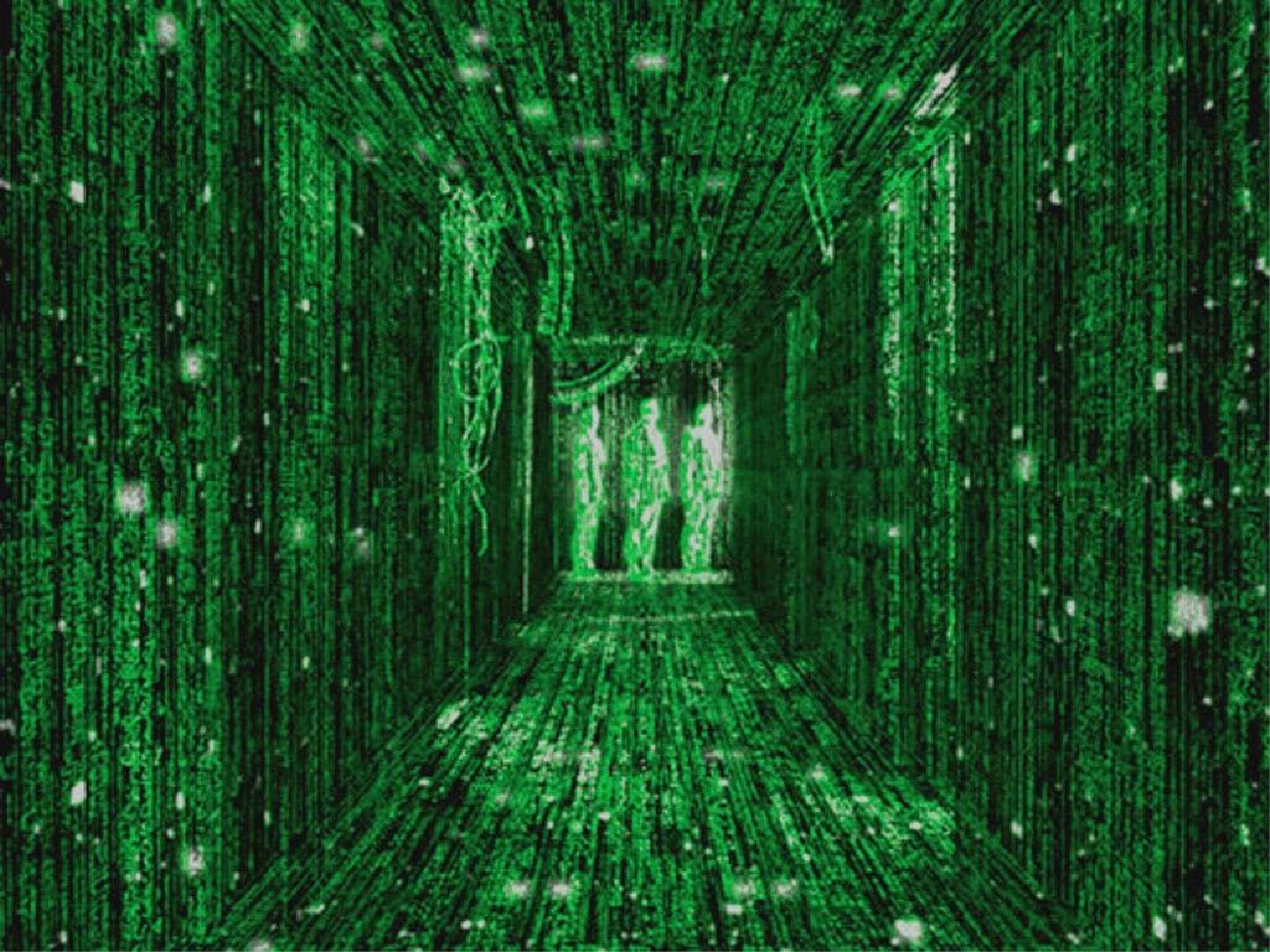
- It's so simple
- It's human readable
- Documentation isn't really necessary
- Debugging is easy
- You can encapsulate it
- You can customize it

```
$ nc -v mit.edu 80
Connection to mit.edu port 80 [tcp/http] succeed
GET / HTTP/1.1
Host: mit.edu

HTTP/1.1 302 Moved Temporarily
Server: AkamaiGHost
Content-Length: 0
Location: http://web.mit.edu/
Date: Mon, 09 Mar 2015 19:43:03 GMT
Connection: keep-alive
```

Sent

Received



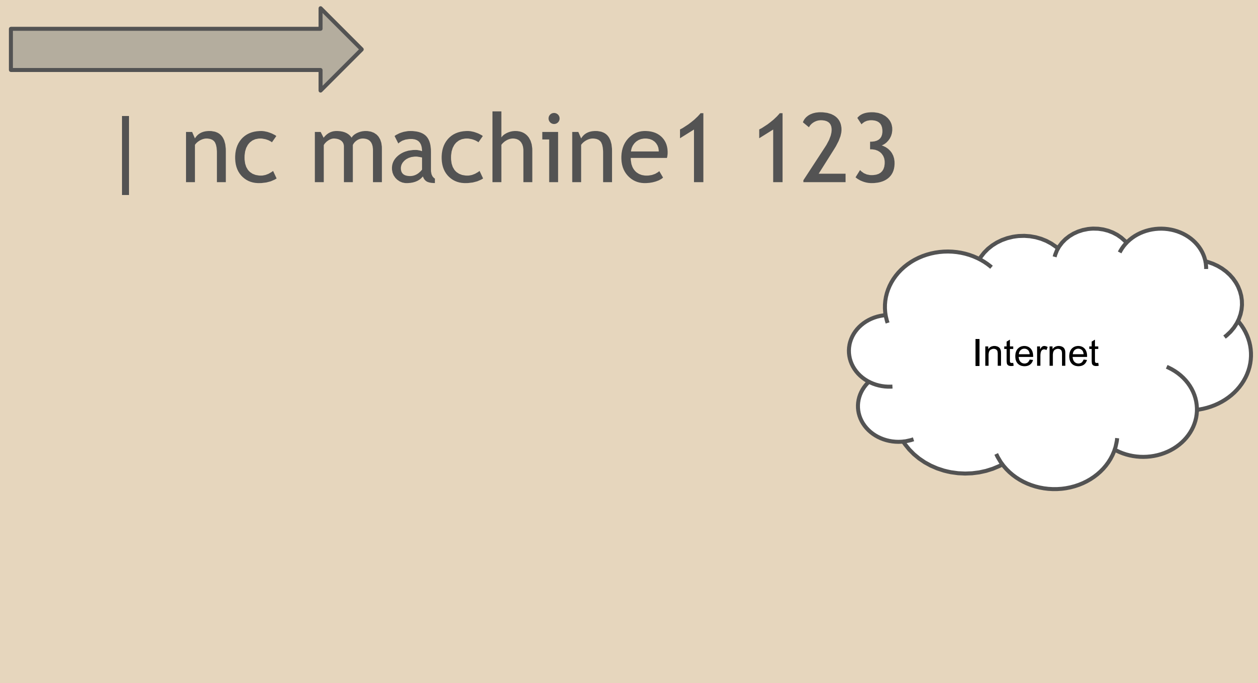
What is Blucat?

1. debugging tool for bluetooth applications
   a. connect to service for testing/emulation

2. device exploration tool
   a. reverse engineer existing services
   b. record nearby devices using scripts

3. a component in building other applications
   a. build applications on top of Blucat

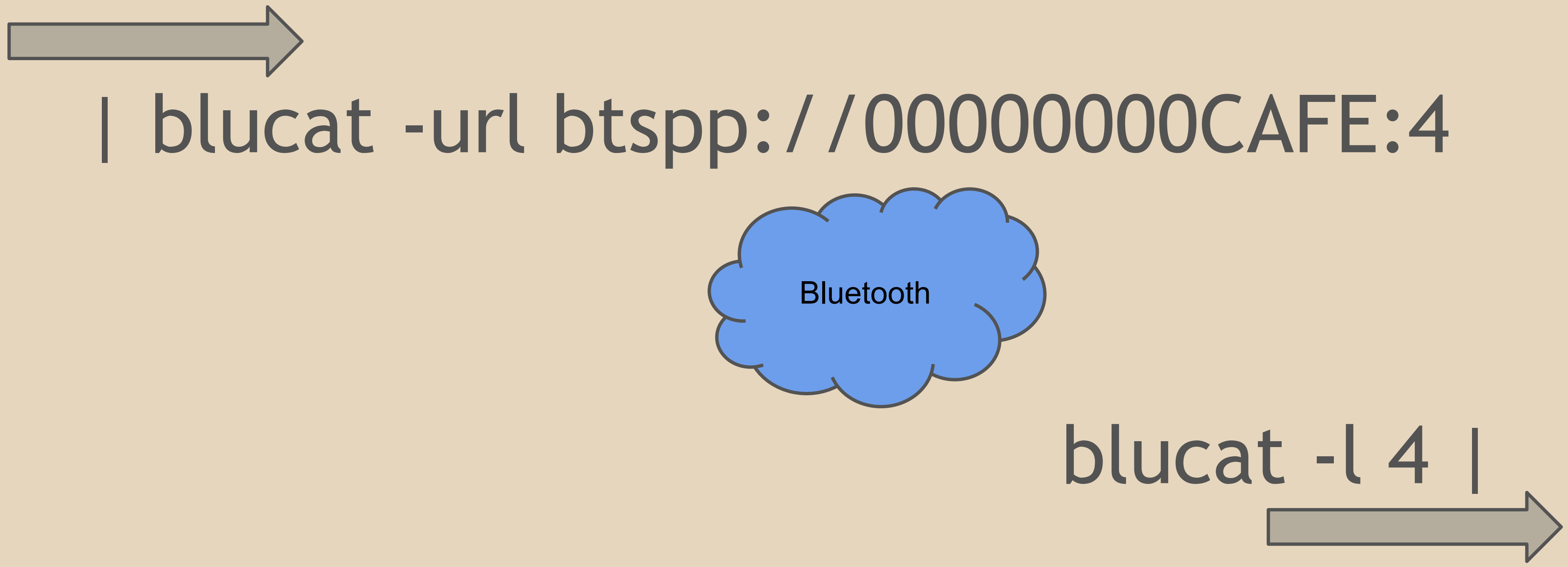
with netcat



```
| nc machine1 123
```

```
nc -l 123 |
```

with blucat



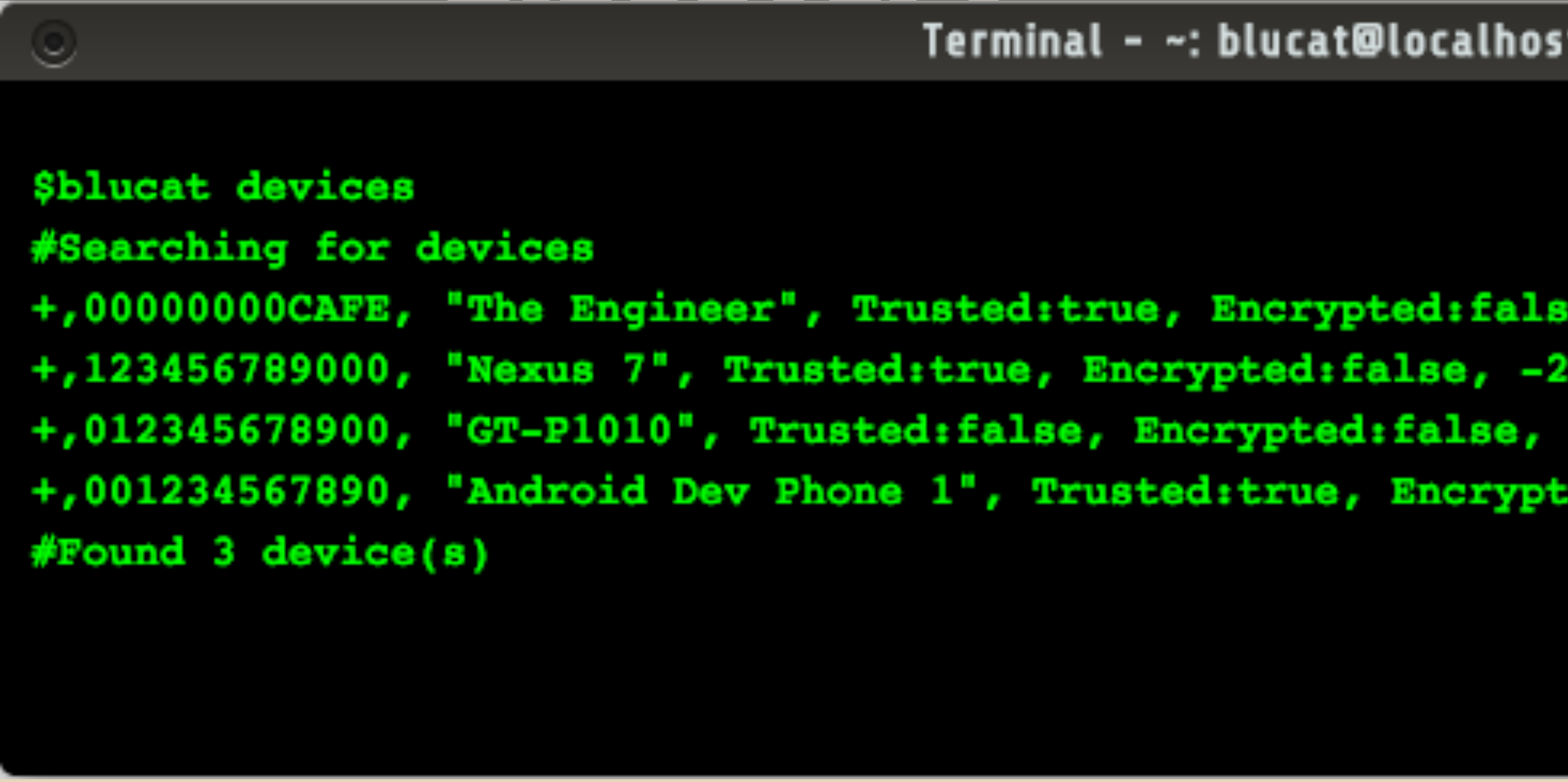
```
| blucat -url btspp://00000000CAFE:4
```

```
blucat -l 4 |
```

with nmap

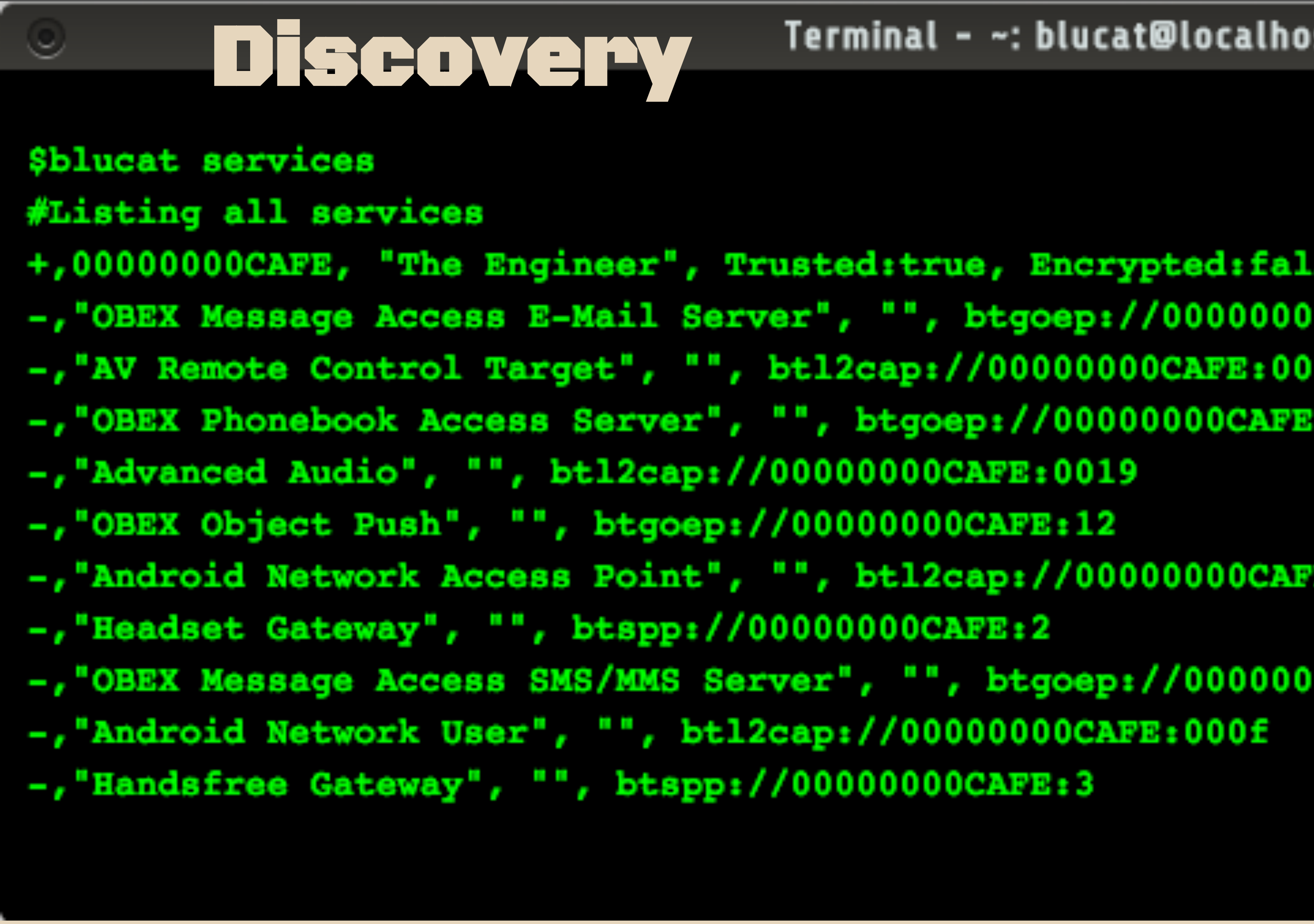
```
$nmap somehost
Starting Nmap 5.21 ( http://nmap.org )
Nmap scan report
Not shown: 846 closed ports, 152 filtered
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Discovery


Terminal - ~: blucat@localhos

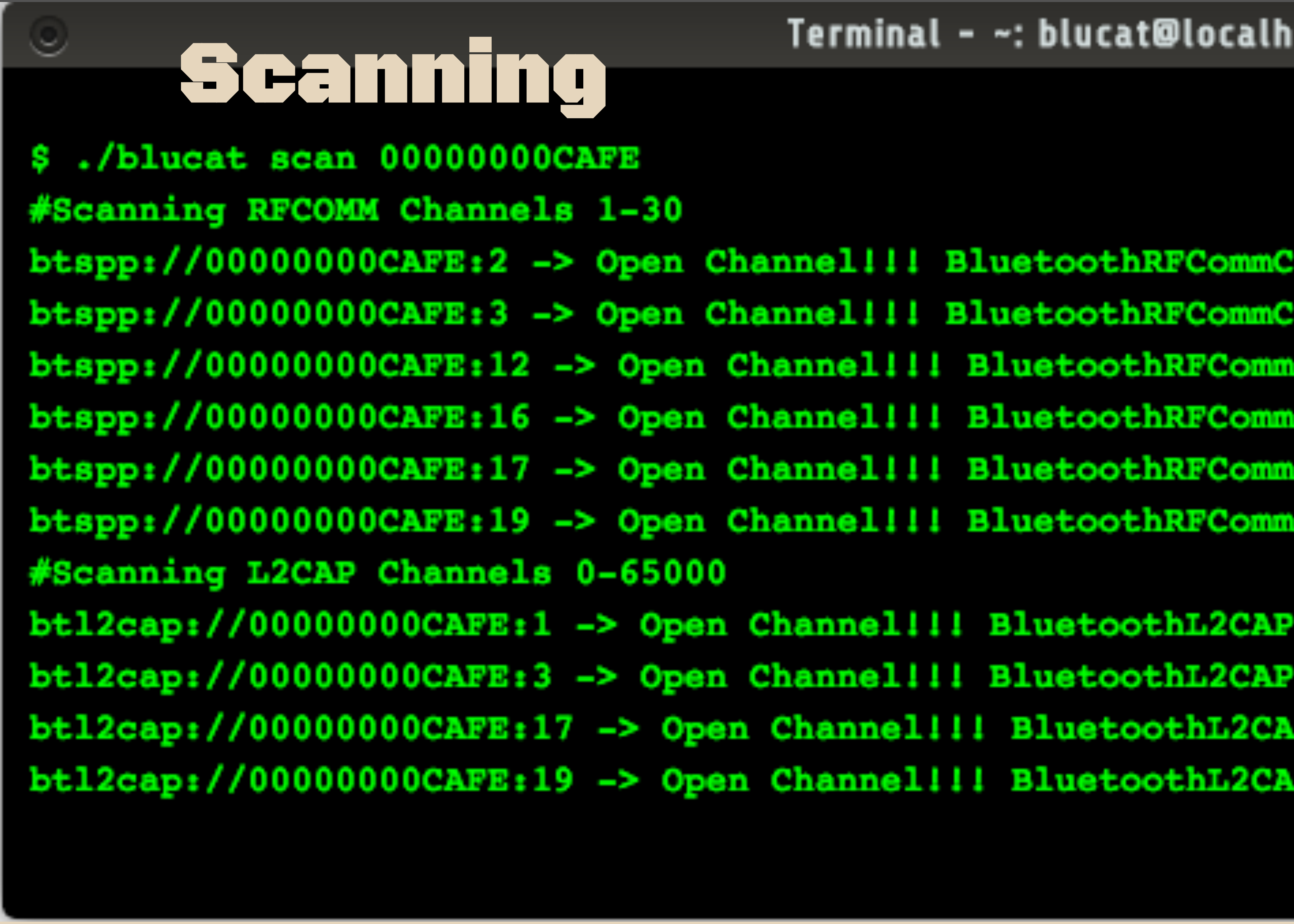
```
$blucat devices
#Searching for devices
+,00000000CAFE, "The Engineer", Trusted:true, Encrypted:fals
+,123456789000, "Nexus 7", Trusted:true, Encrypted:false, -2
+,012345678900, "GT-P1010", Trusted:false, Encrypted:false,
+,001234567890, "Android Dev Phone 1", Trusted:true, Encrypt
#Found 3 device(s)
```

Discovery



```
$blucat services
#Listing all services
+,00000000CAFE, "The Engineer", Trusted:true, Encrypted:fal
-,"OBEX Message Access E-Mail Server", "", btgoep://0000000
-,"AV Remote Control Target", "", btl2cap://00000000CAFE:00
-,"OBEX Phonebook Access Server", "", btgoep://00000000CAFE
-,"Advanced Audio", "", btl2cap://00000000CAFE:0019
-,"OBEX Object Push", "", btgoep://00000000CAFE:12
-,"Android Network Access Point", "", btl2cap://00000000CAF
-,"Headset Gateway", "", btspp://00000000CAFE:2
-,"OBEX Message Access SMS/MMS Server", "", btgoep://000000
-,"Android Network User", "", btl2cap://00000000CAFE:000f
-,"Handsfree Gateway", "", btspp://00000000CAFE:3
```

Scanning



```
$ ./blucat scan 00000000CAFE
#Scanning RFCOMM Channels 1-30
btspp://00000000CAFE:2 -> Open Channel!!! BluetoothRFCommC
btspp://00000000CAFE:3 -> Open Channel!!! BluetoothRFCommC
btspp://00000000CAFE:12 -> Open Channel!!! BluetoothRFComm
btspp://00000000CAFE:16 -> Open Channel!!! BluetoothRFComm
btspp://00000000CAFE:17 -> Open Channel!!! BluetoothRFComm
btspp://00000000CAFE:19 -> Open Channel!!! BluetoothRFComm
#Scanning L2CAP Channels 0-65000
btl2cap://00000000CAFE:1 -> Open Channel!!! BluetoothL2CAP
btl2cap://00000000CAFE:3 -> Open Channel!!! BluetoothL2CAP
btl2cap://00000000CAFE:17 -> Open Channel!!! BluetoothL2CA
btl2cap://00000000CAFE:19 -> Open Channel!!! BluetoothL2CA
```

Bluetooth URI Monikers

ex: btspp://10643FC98386:17

Bluetooth URI Monikers

btspp -

Bluetooth serial port profile RFCOMM

btl2cap -

Logical link control and adaptation protocol

btgoep -

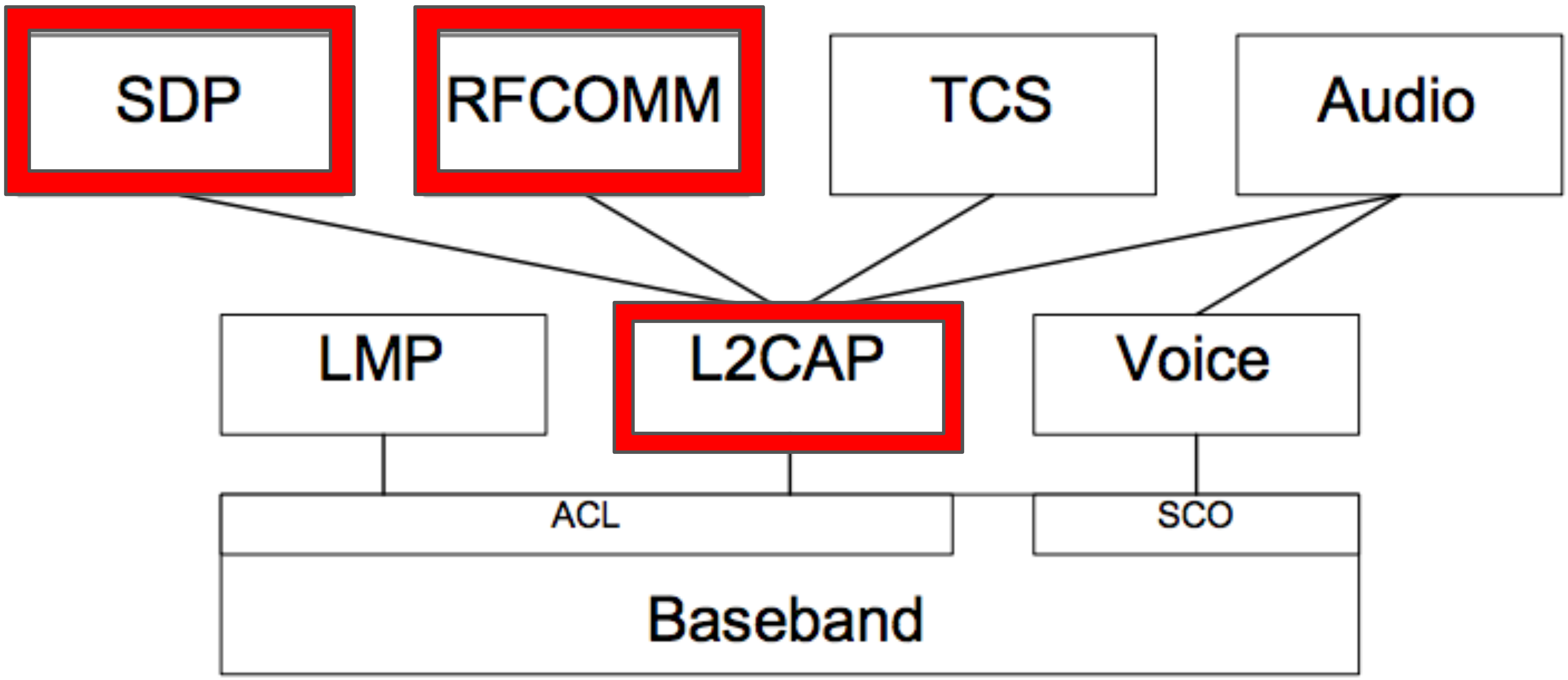
OBEX Generic Object Exchange profile

serial port profile (SPP)

- designed to emulate RS-232 serial ports
- same major attributes of TCP sockets
  - in order, retry,
- only allows ~30 ports
  - depends on stack
  - assigned dynamically like portmap (TCP/111)

link layer common access protocol (L2CAP)

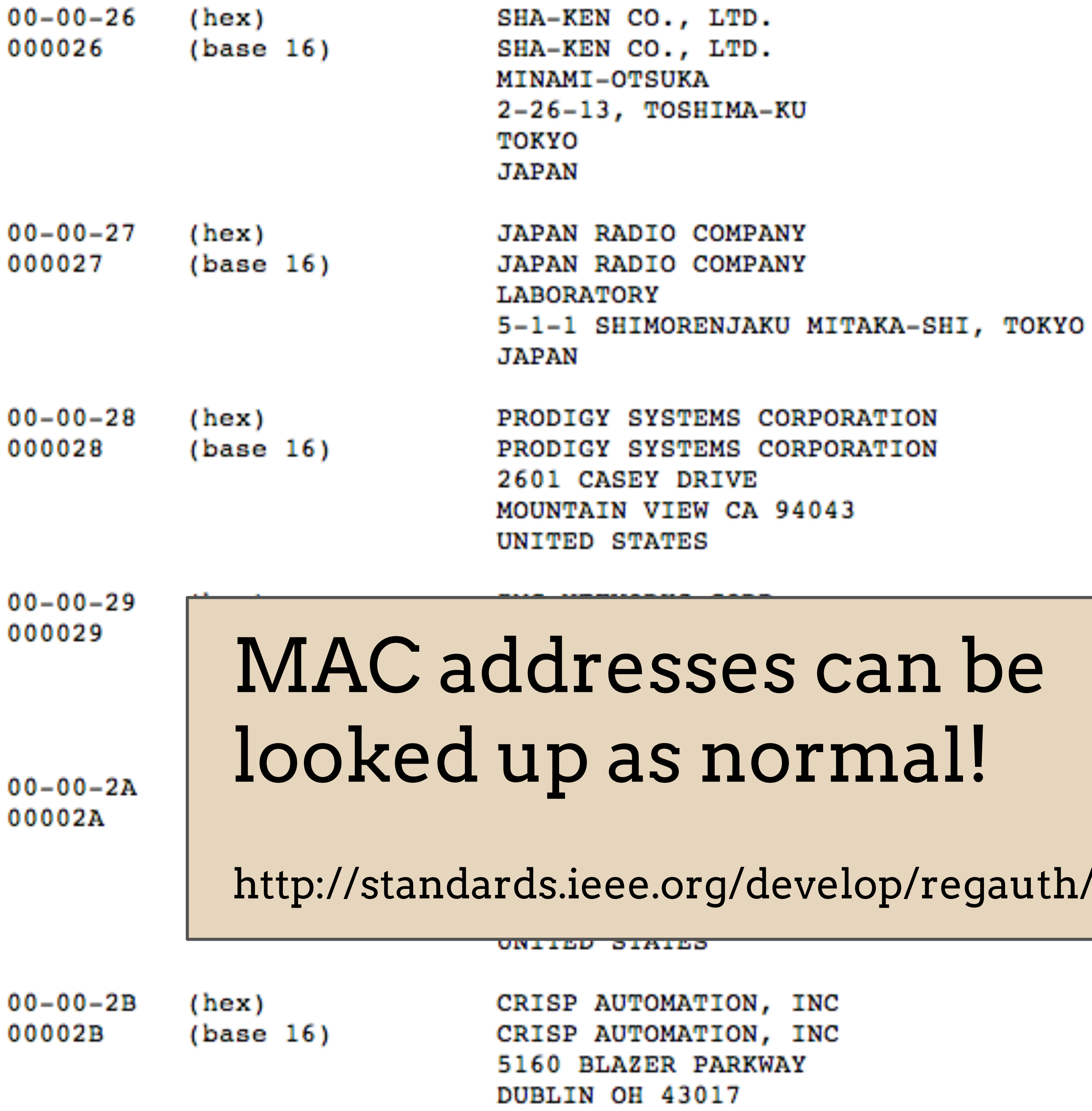
- can make unreliable similar to UDP
- default maximum packet size is 672 bytes
- RFCOMM uses L2CAP as a transport
    - connects over L2CAP PSM #3
- more port numbers
    - aka PSM (Protocol Service Multiplexer) number



L2CAP in Bluetooth Protocol Architecture

I want data in the form of a table!

protocol	terminology	reserved/ well-known ports	dynamically assigned ports
TCP	port	1-1024	1025-65535
UDP	port	1-1024	1025-65535
RFCOMM	**channel**	**none**	**1-30**
L2CAP	**PSM**	**odd numbered 1-4095**	**odd numbered 4097 - 32765**



```
00-00-26    (hex)            SHA-KEN CO., LTD.
000026      (base 16)        SHA-KEN CO., LTD.
                             MINAMI-OTSUKA
                             2-26-13, TOSHIMA-KU
                             TOKYO
                             JAPAN

00-00-27    (hex)            JAPAN RADIO COMPANY
000027      (base 16)        JAPAN RADIO COMPANY
                             LABORATORY
                             5-1-1 SHIMORENJAKU MITAKA-SHI, TOKYO
                             JAPAN

00-00-28    (hex)            PRODIGY SYSTEMS CORPORATION
000028      (base 16)        PRODIGY SYSTEMS CORPORATION
                             2601 CASEY DRIVE
                             MOUNTAIN VIEW CA 94043
                             UNITED STATES

00-00-29
000029



00-00-2A
00002A



                             UNITED STATES

00-00-2B    (hex)            CRISP AUTOMATION, INC
00002B      (base 16)        CRISP AUTOMATION, INC
                             5160 BLAZER PARKWAY
                             DUBLIN OH 43017
```

MAC addresses can be looked up as normal!

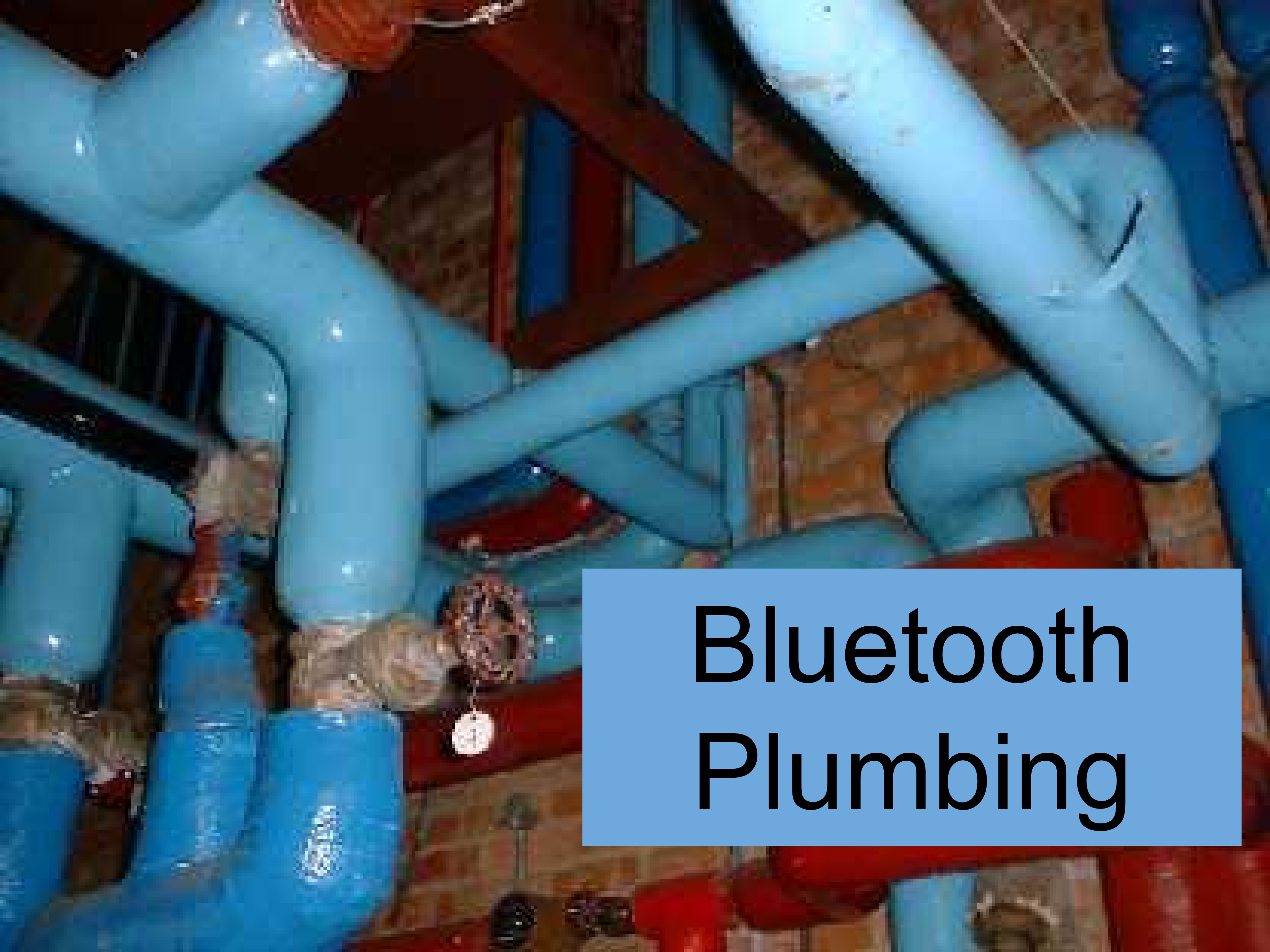
http://standards.ieee.org/develop/regauth/oui/oui.txt

On connect execution!

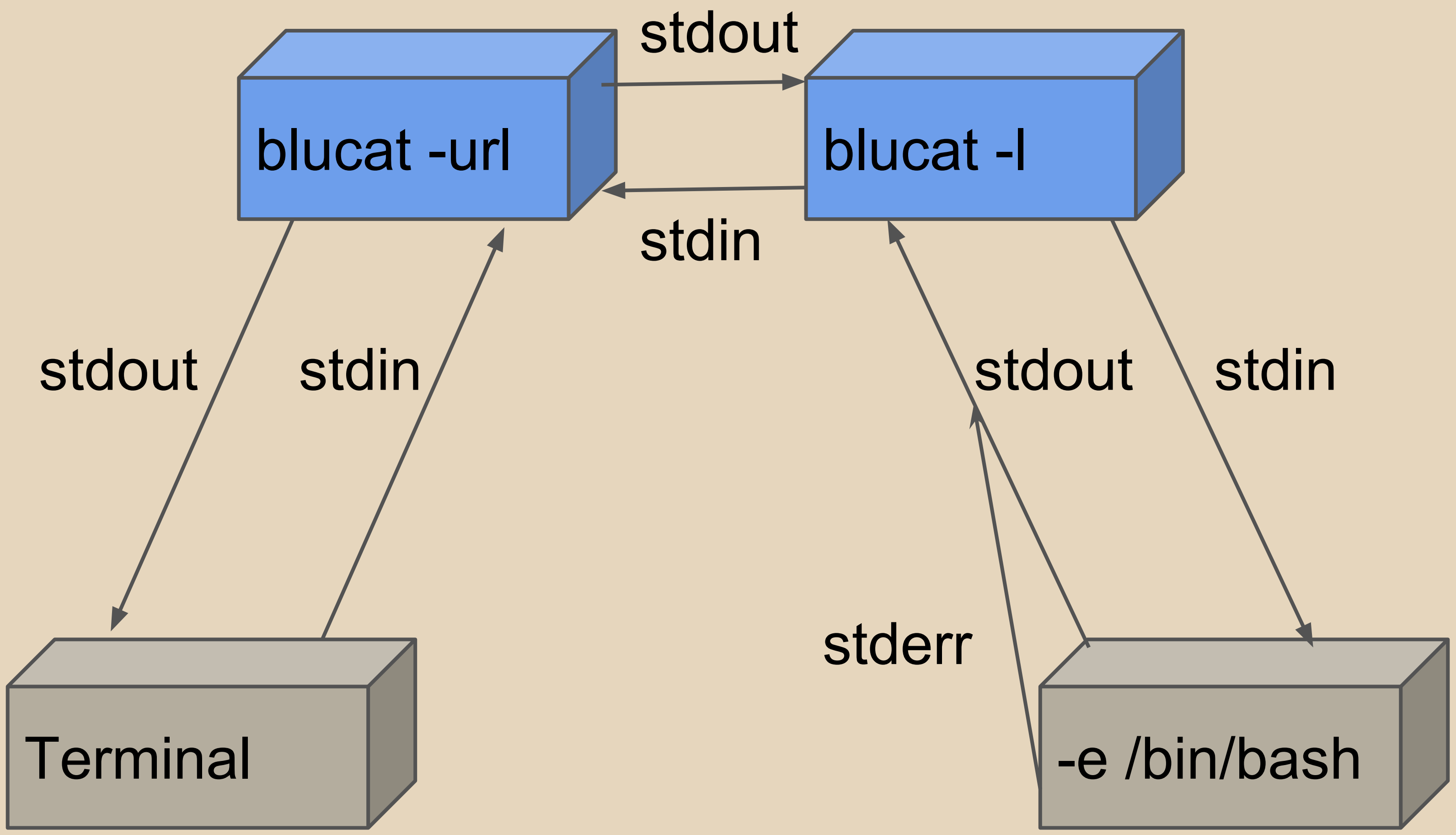
```
$./blucat -v -l -e /bin/bash
#Listening at btspp://002608AAAAAA:4
```

```
$./blucat services
"BlueCatPipe","",btspp://002608AAAAAA:4
```

```
$./blucat -url btspp://002608AAAAAA:4 -v
#Connected
Hi
/bin/bash: line 1: Hi: command not found
```



Bluetooth Plumbing



Bluetooth pipefitting for -e

Inspecting devices

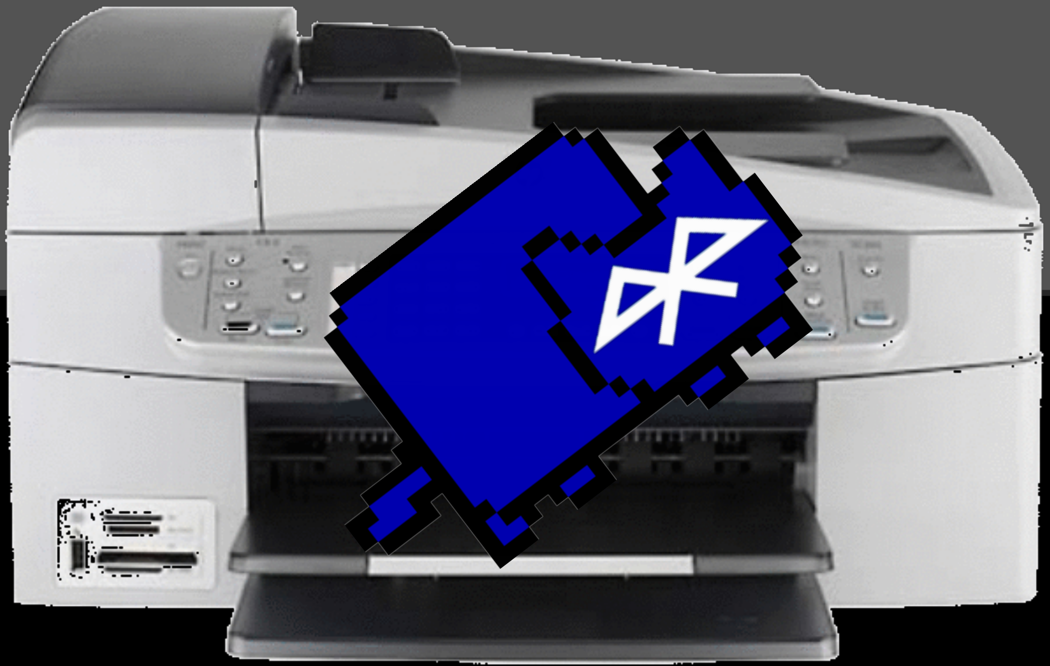
Bluetooth has “profiles”

Identified by UUID and device class

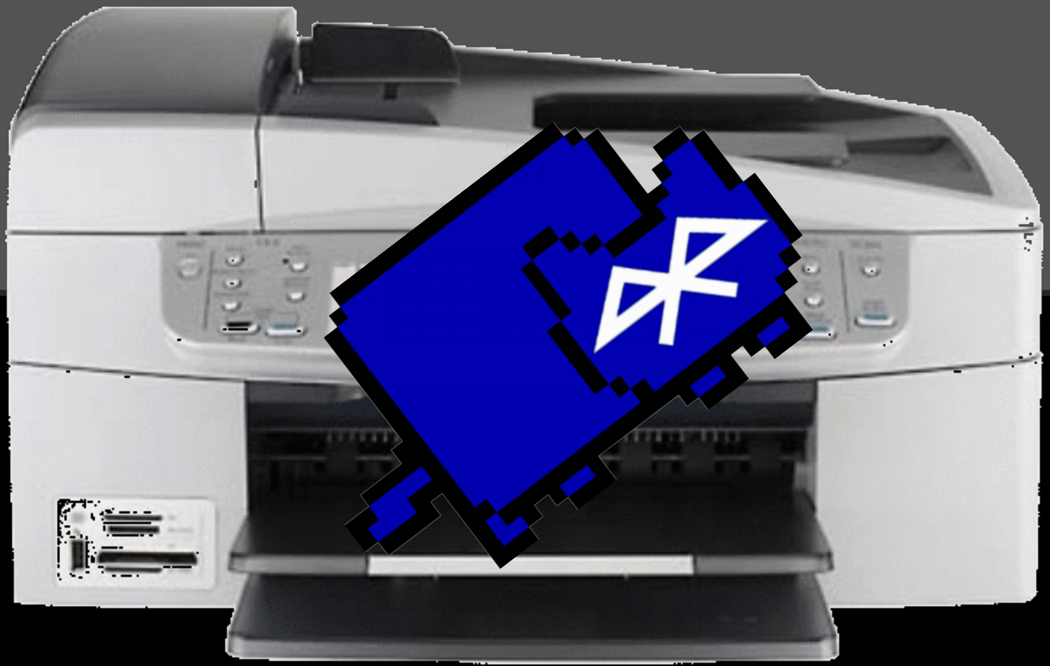
Implemented by one or more services which may be RFCOMM or L2CAP

000C55F8FBEE, "Officejet 6300 series"

```
00-0C-55   (hex)               Microlink Communications Inc.
000C55     (base 16)           Microlink Communications Inc.
                               8F, 31, Hsintai Road
                               Chupei City
                               Hsinchu  302
                               TAIWAN, PROVINCE OF CHINA
```

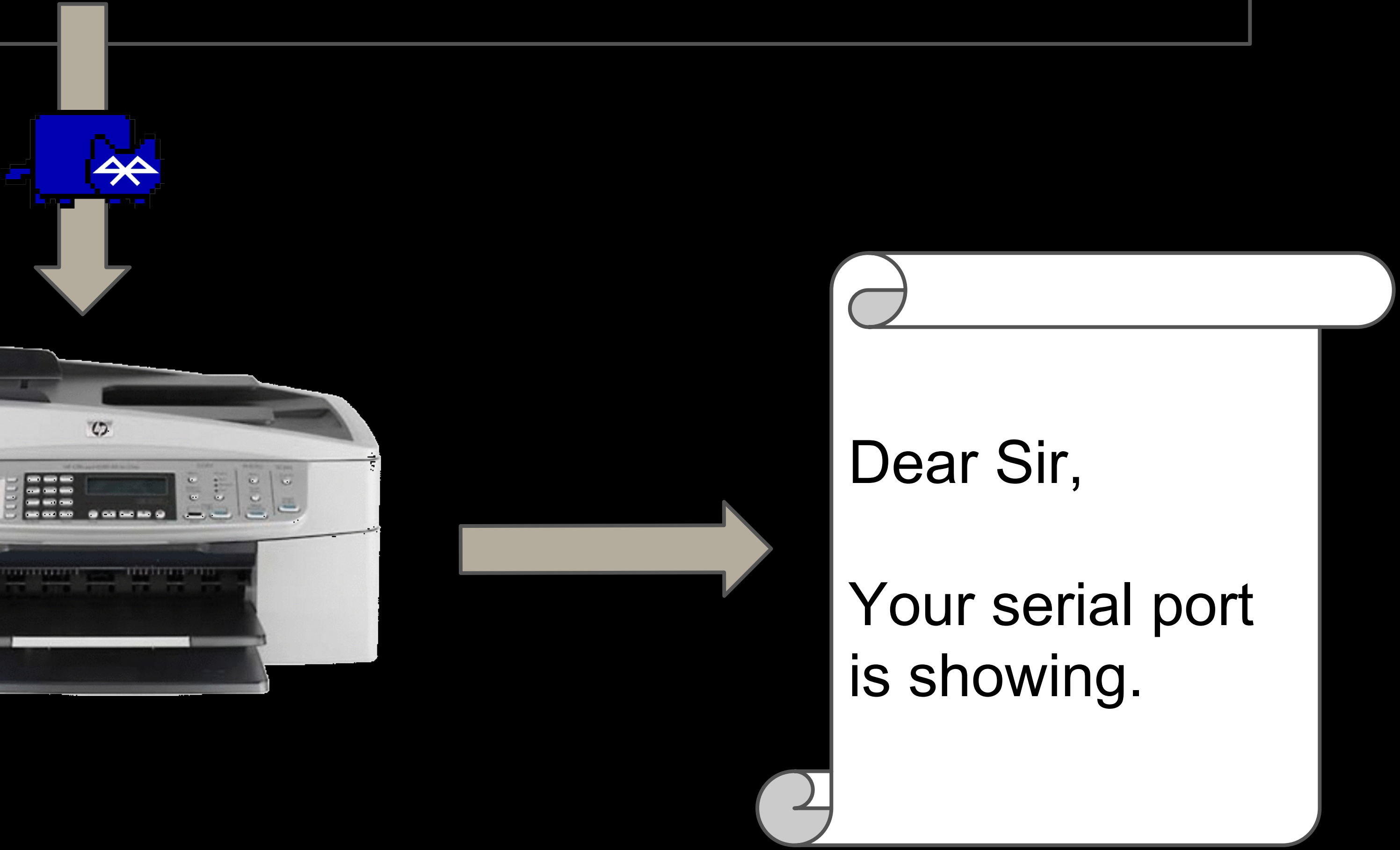


```
30F306AAAAAA, "Officejet 6300 series", Trusted:false, ...
"OBEX Object Push", "", btgoep://30F306598203:2
"Serial Port", "", btspp://30F306598203:1
"Basic Printing", "", btgoep://30F306598203:4
"Basic Imaging", "", btgoep://30F306598203:3
```



```
$./blucat -url btspp://30F306598203:1
```

```
$./blucat -v -url btspp://30F306598203:1
# Connected
Dear Sir, ...
```

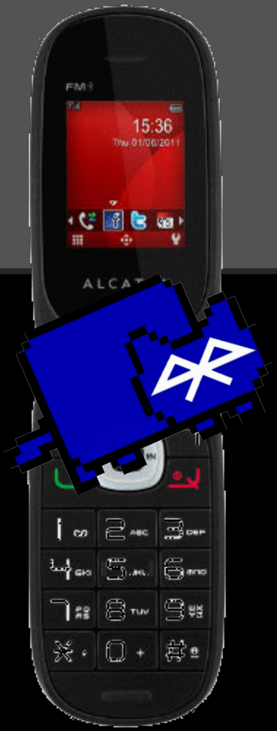


Dear Sir,

Your serial port is showing.

FM
15:36
Thu 01/06/2011
ALCATEL

Alcatel one touch 665A



```
"Serial Port0", "", btspp://9471ACDBACAD:11
```

```
9471ACAAAAAA, "Alcatel one touch 665A", ...
"AUDIO Gateway", "", btspp://9471ACDBACAD:1
"OBEX Object Push", "", btgoep://9471ACDBACAD:4
"Serial Port0", "", btspp://9471ACDBACAD:11
"Dial-up Networking", "", btspp://9471ACDBACAD:9
"Voice gateway", "", btspp://9471ACDBACAD:2
```

```
$ ./blucat -url btspp://9471ACAAAAAA:11
AT+CGMI          Typed
+CGMI: Alcatel
OK

                 Typed
AT+CGMM
+CGMM: one touch 665A
OK

                 Typed
AT+CGMR
+CGMR: Alcatel 010 04, 2012/03/05 14:56
OK
```

More AT Hayes Commands?

https://github.com/boos/bluesnarfer/blob/master/src/bluesnarfer.c

http://www.forensicswiki.org/wiki/AT_Commands

http://www.anotherurl.com/library/at_test.htm

http://gatling.ikk.sztaki.hu/~kissg/gsm/at+c.html

```
$ blucat services
#Listing all services
+,001B7A2879AA, "Nintendo RVL-CNT-01", Trust
Encrypted:false, NA
-,"", "", null
-,"Nintendo RVL-CNT-01", "", btl2cap://001B7A287
-,"", "", null
```

```
$ blucat scan 001B7A2879AA
#Scanning RFCOMM Channels 1-30
#Scanning L2CAP Channels 0-65000
btl2cap://001B7A2879AA:1 -> Open Cha
btl2cap://001B7A2879AA:11 -> Open Ch
btl2cap://001B7A2879AA:13 -> Open Ch
```

nexus
LG
FCC ID: ZNFE960 IC: 2703C-E960
MODEL LG-E960 MADE IN KOREA
CE 0168
4:20
Google
Maps
Play Store
YouTube
Google

```
$ ./blucat services
#Listing all services
+,00000000CAFE, "The Engineer", Trusted:true, Encrypte
-,"OBEX Message Access SMS/MMS Server", "", btgoep://000
-,"OBEX Phonebook Access Server", "", btgoep://00000000
-,"OBEX Object Push", "", btgoep://00000000CAFE:12
-,"Headset Gateway", "", btspp://00000000CAFE:2
-,"OBEX Message Access E-Mail Server", "", btgoep://0000
-,"Handsfree Gateway", "", btspp://00000000CAFE:3
```

"Handsfree Gateway", btspp://00000000CAFE:3

```
$ ./blucat -url btspp://00000000CAFE:3 -v
#Waiting for connection
#Connected
AT
AT+

ERROR
AT*

#Error: Connection is closed
```

Hands-Free Profile

AT+BLDN	Redials the previously dialed number.
AT+BRSF	Retrieves the supported features.
AT+BVRA	Enables or disables voice recognition in the AG.
AT+CCWA	Enables call waiting notification in the AG.
AT+CHUP	Rejects an incoming call.
AT+CIND?	Reads the current status of the AG indicators.
AT+CIND=?	Retrieves the indicator mappings for the AG.
AT+CLIP	Enables the call line identification.
AT+CMER	Registers or unregisters status updates.
AT+VGM=<gain>	Notifies the AG service when the microphone volume on the headset is changed to the specified gain value.
AT+VGS=<gain>	Notifies the AG service when the speaker volume on the headset is changed to the specified gain value.
AT+VTS	Transmits DTMF codes to the network.
ATA	Receives an incoming call.
ATD>nnn	Dials a number in memory.
ATDdd...dd	Dials a number.

What works

```
AT+CNUM
"16175555555",129,,4

AT+CIND=?
("call",(0,1)),("callsetup",(0-3)),("service",(0-
1)),("signal",(0-5)),("roam",(0,1)),("battchg",(0-
5)),("callheld",(0-2))

ATD5555555555     // dials number
```

Rapid prototyping with Blucat

How to prototype

Current presentation is using blucat (maybe)

- Android app creates service
  - sends strings to whoever connects
  - “f” and “b” are wired to buttons
- Laptop runs blucat and pipes it into script
- Script dispatches “f” and “b” to press left and right keys
- https://github.com/ieee8023/blucat-android-remote

Launch blucat and pipe to dispatcher

```
blucat -k -v -url btspp://00000000CAFE:4
-e "/bin/bash $(pwd)/dispatcher.sh"
```

Dispatcher reads input

```
while read input
do
    if [[ "$input" == *"f"* ]]; then
        echo "Forward"
        sh key-mac.sh 124
    fi
...
```

Details

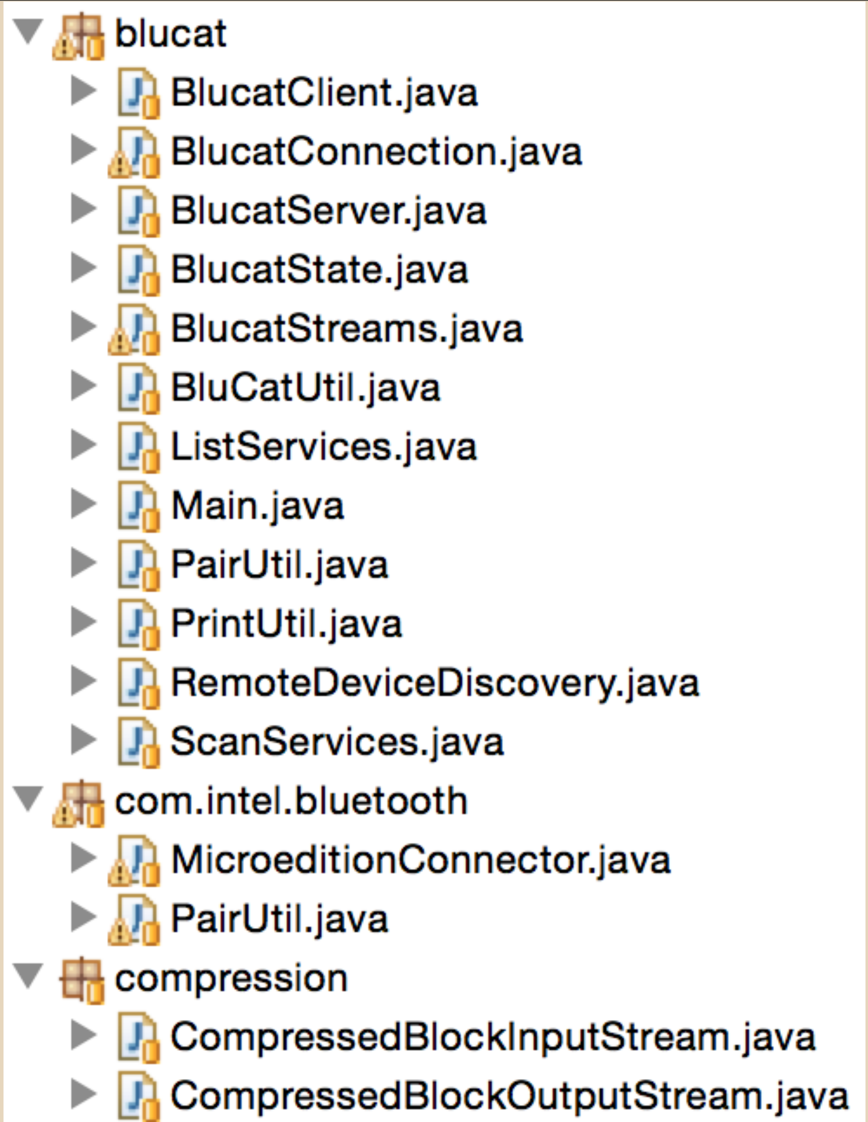
Java based

Uses BlueCove Java Libraries

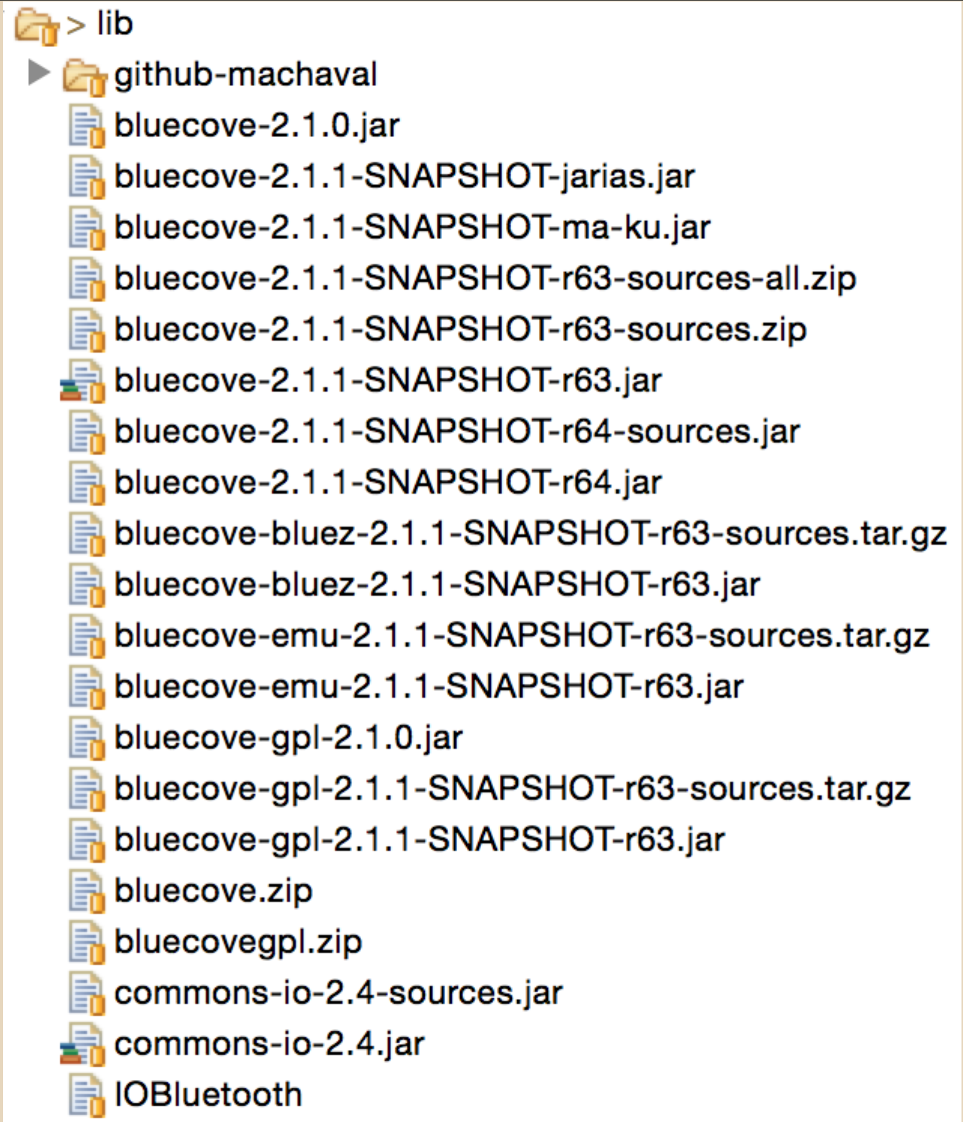
Works on Mac and Linux

State of the code

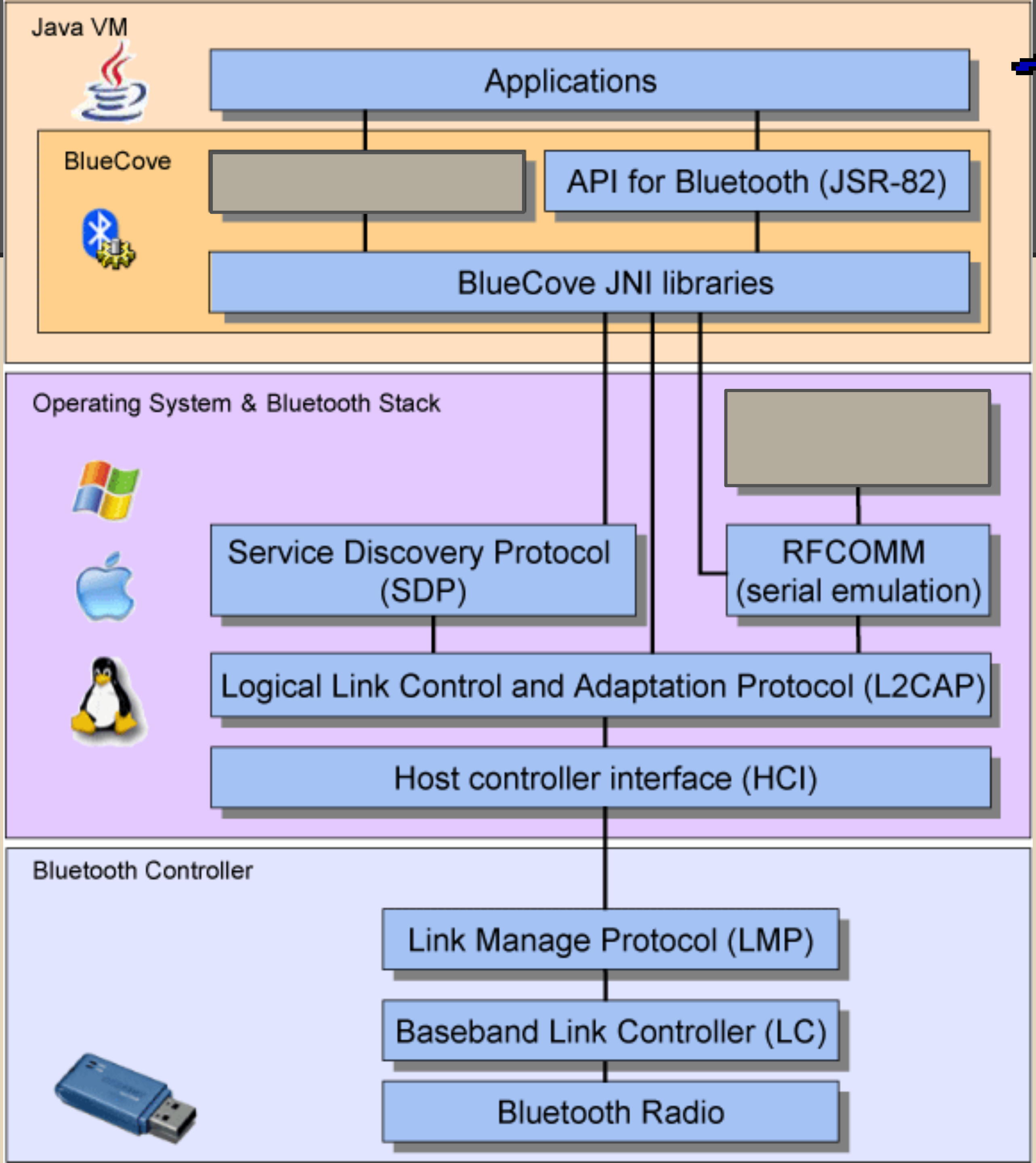
https://github.com/ieee8023/blucat

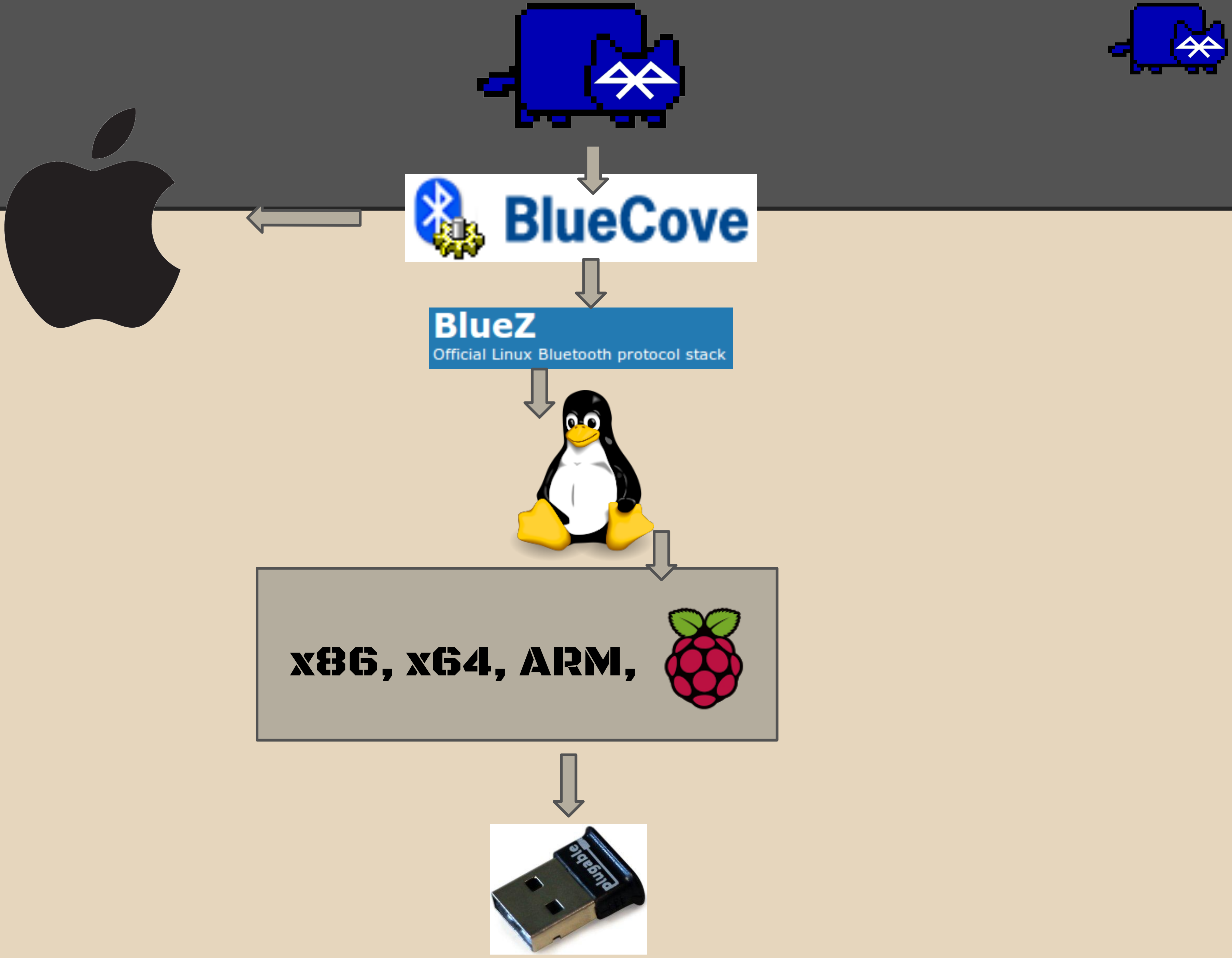


```
▼ blucat
    ▶ BlucatClient.java
    ▶ BlucatConnection.java
    ▶ BlucatServer.java
    ▶ BlucatState.java
    ▶ BlucatStreams.java
    ▶ BluCatUtil.java
    ▶ ListServices.java
    ▶ Main.java
    ▶ PairUtil.java
    ▶ PrintUtil.java
    ▶ RemoteDeviceDiscovery.java
    ▶ ScanServices.java
▼ com.intel.bluetooth
    ▶ MicroeditionConnector.java
    ▶ PairUtil.java
▼ compression
    ▶ CompressedBlockInputStream.java
    ▶ CompressedBlockOutputStream.java
```



```
> lib
  ▶ github-machaval
    bluecove-2.1.0.jar
    bluecove-2.1.1-SNAPSHOT-jarias.jar
    bluecove-2.1.1-SNAPSHOT-ma-ku.jar
    bluecove-2.1.1-SNAPSHOT-r63-sources-all.zip
    bluecove-2.1.1-SNAPSHOT-r63-sources.zip
    bluecove-2.1.1-SNAPSHOT-r63.jar
    bluecove-2.1.1-SNAPSHOT-r64-sources.jar
    bluecove-2.1.1-SNAPSHOT-r64.jar
    bluecove-bluez-2.1.1-SNAPSHOT-r63-sources.tar.gz
    bluecove-bluez-2.1.1-SNAPSHOT-r63.jar
    bluecove-emu-2.1.1-SNAPSHOT-r63-sources.tar.gz
    bluecove-emu-2.1.1-SNAPSHOT-r63.jar
    bluecove-gpl-2.1.0.jar
    bluecove-gpl-2.1.1-SNAPSHOT-r63-sources.tar.gz
    bluecove-gpl-2.1.1-SNAPSHOT-r63.jar
    bluecove.zip
    bluecovegpl.zip
    commons-io-2.4-sources.jar
    commons-io-2.4.jar
    IOBluetooth
```


Java VM
Applications
BlueCove
API for Bluetooth (JSR-82)
BlueCove JNI libraries
Operating System & Bluetooth Stack
Service Discovery Protocol (SDP)
RFCOMM (serial emulation)
Logical Link Control and Adaptation Protocol (L2CAP)
Host controller interface (HCI)
Bluetooth Controller
Link Manage Protocol (LMP)
Baseband Link Controller (LC)
Bluetooth Radio

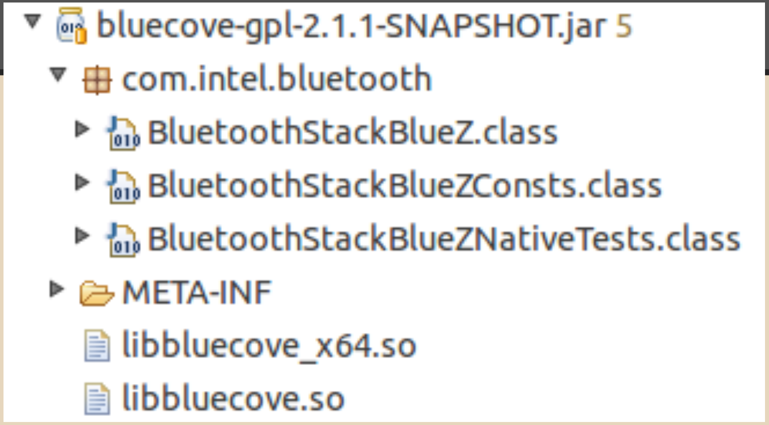

BlueCove
BlueZ
Official Linux Bluetooth protocol stack
x86, x64, ARM,
plugable

Running blucat

```
$./blucat
```

```
if [[ $OSTYPE == *darwin* ]]; then
   LIBS=build/blucat.jar:lib/bluecove-2.1.1-SNAPSHOT.jar
   ...
elif [[ $OSTYPE == *linux* ]]; then
   if [[ $MACH == *arm* ]]; then
       LIBS=$DIR/...
   else
       LIBS=$DIR/...
   fi
fi
java -cp $COMMONLIBS:$LIBS blucat.Main $@ 2> >(grep --
line-buffered -v NSAutoreleaseNoPool >&2)
```

Java Native Interface



bluecove-gpl-2.1.1-SNAPSHOT.jar 5
com.intel.bluetooth
BluetoothStackBlueZ.class
BluetoothStackBlueZConsts.class
BluetoothStackBlueZNativeTests.class
META-INF
libbluecove_x64.so
libbluecove.so

```
==Somewhere in the program:
System.loadLibrary("bluecove");
// Searched for file
// libbluecove.so
// in LD_LIBRARY_PATH

==BluetoothStackBlueZ.java:
private native
int rfServerGetChannelIDImpl(long handle) throws
IOException;

==Some C file
JNIEXPORT void JNICALL
Java_bluecove_rfServerGetChannelIDImpl(JNIEnv *env,
jobject obj, jlong handle){ … }
```

JSR-82 Basics

LocalDevice

Portal to adaptor

RemoteDevice

Represents paired, discovered, and connected devices

Connection

Access to Streams. "Notifier" version is server.

UUID

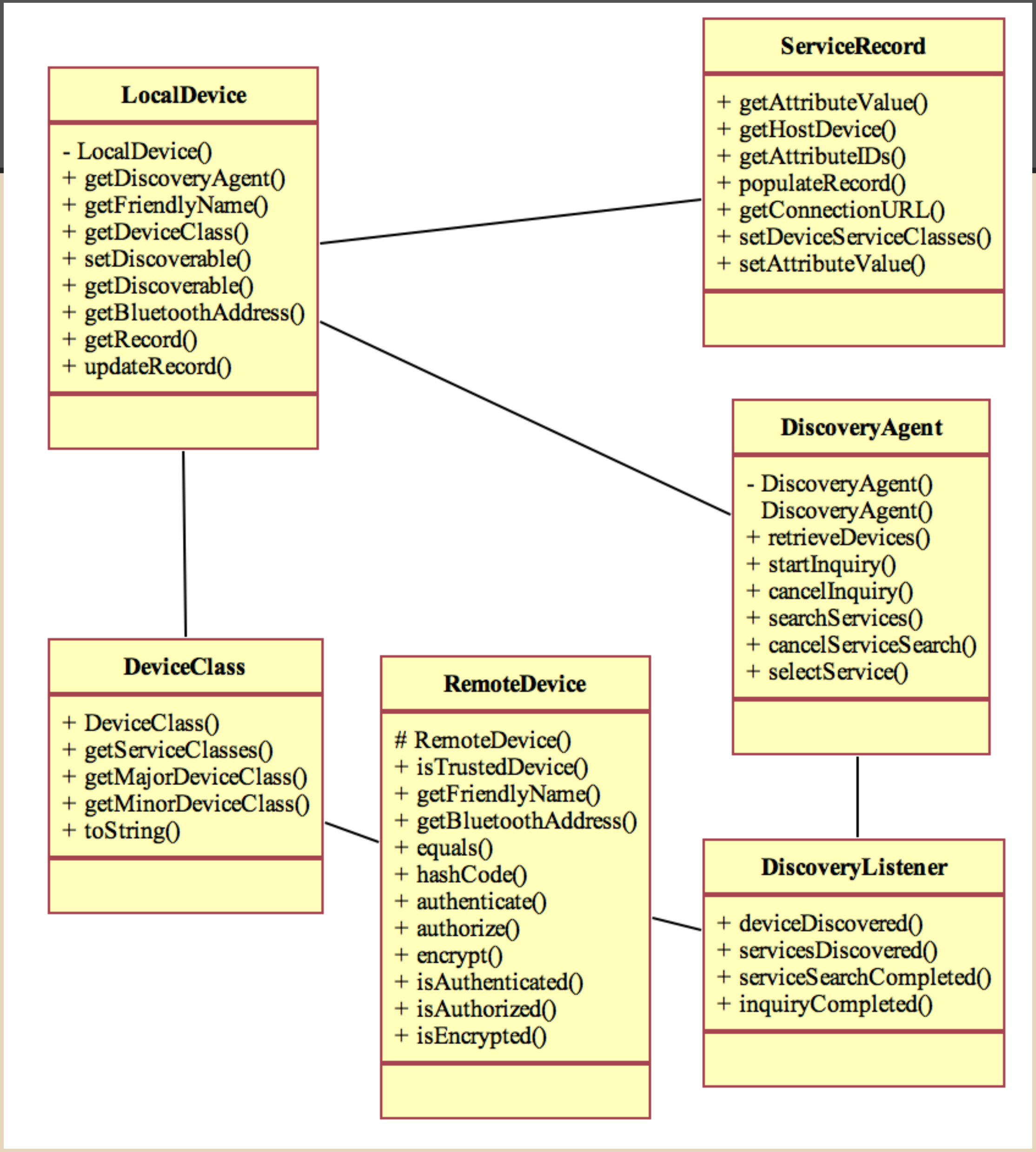
Identifies service and "profile."

ServiceDiscovery

Access to run SDP on nearby devices.

ServiceRecord

Advertisement of service. UUID, Channel, Name.



Example

1. Make a connection to a device in Java
2. Echo messages to console
3. Send messages to the device

Make a connection

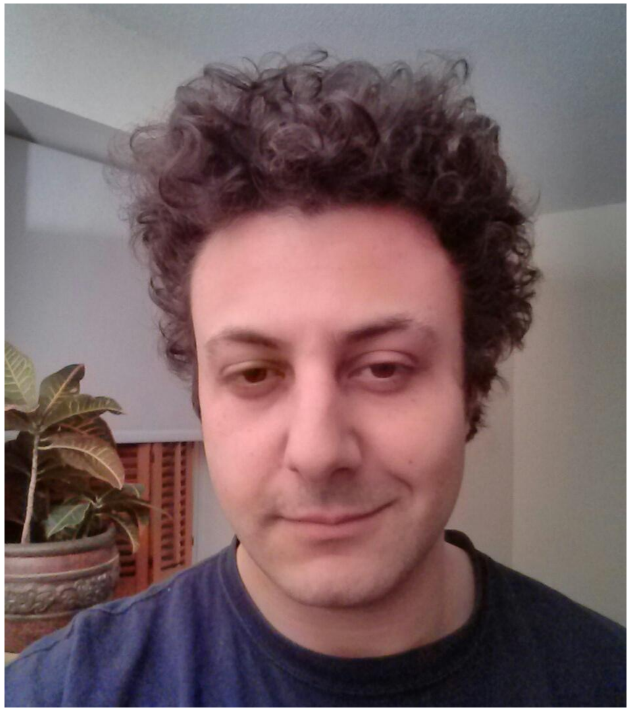
```
String url = "btspp://50B0FA2C2AAE:4";
StreamConnection con =
                Connector.open(url,
                Connector.READ_WRITE,
                true);
InputStream is =
             con.openDataInputStream();
OutputStream os =
             con.openDataOutputStream();
```

Echo to console

```
BufferedReader in =
            new BufferedReader(
            new InputStreamReader(is));

String line = null;
while((line = in.readLine()) != null) {
   System.out.println(line);
}
```

Speaker



Joseph Paul Cohen

Email: joecohen@cs.umb.edu

National Science Foundation Graduate Fellow

Ph.D Candidate - Computer Science

University of Massachusetts Boston

blucat

http://blucat.sf.net

https://github.com/ieee8023/blucat