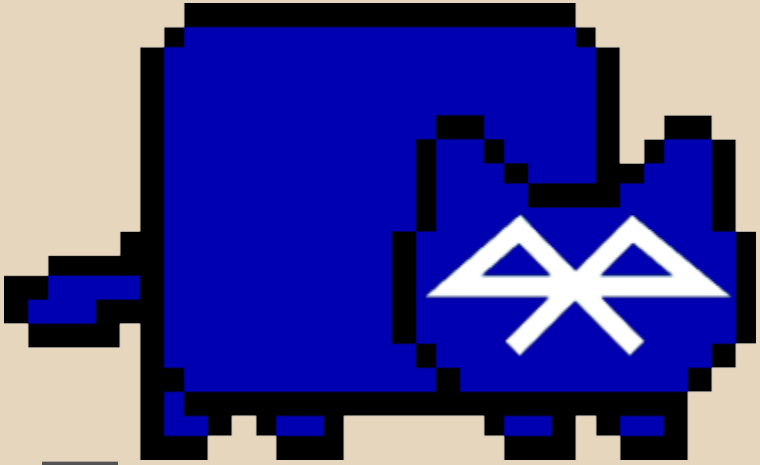


blucat

Joseph Paul Cohen
Defcon 21
http://blucat.sf.net

Abstract

TCP/IP has tools such as nmap and netcat to explore devices and create socket connections. Bluetooth has sockets but doesn't have the same tools. Blucat fills this need for the Bluetooth realm. Blucat can be thought of as a:

1. debugging tool for bluetooth applications
2. device exploration tool
3. a component in building other applications

Blucat is designed to run on many different platforms (including Raspberry Pi) by abstracting core logic from native code using the Bluecove library to interact with a variety of Bluetooth stacks. This talk will go over the objectives, designs, and current results of the project. More information is at http://blucat.sourceforge.net/

Bio

Joseph Paul Cohen (ieee8023)

Joseph is a Ph.D. student at the University of Massachusetts Boston. He has worked for large finance, IT consulting, and startup software companies. He now focuses on computer science research in areas of machine learning and cyber security education.

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?

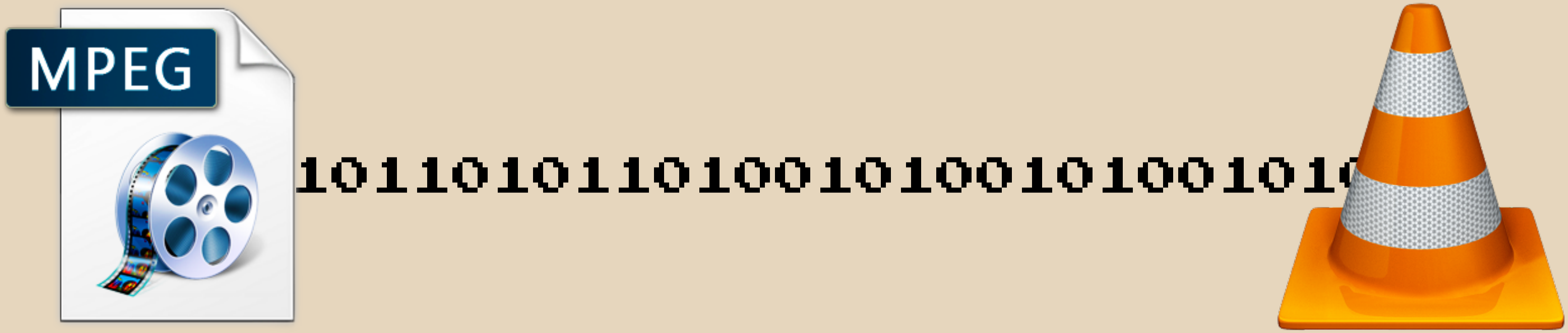
Overview

- Streams
- blucat inline netcat replacements
- blucat as Bluetooth nmap
- rfcomm and l2cap basics
- look at some devices
- how to prototype
- scanning stats
- blucat architecture

STREAMS==AWESOME

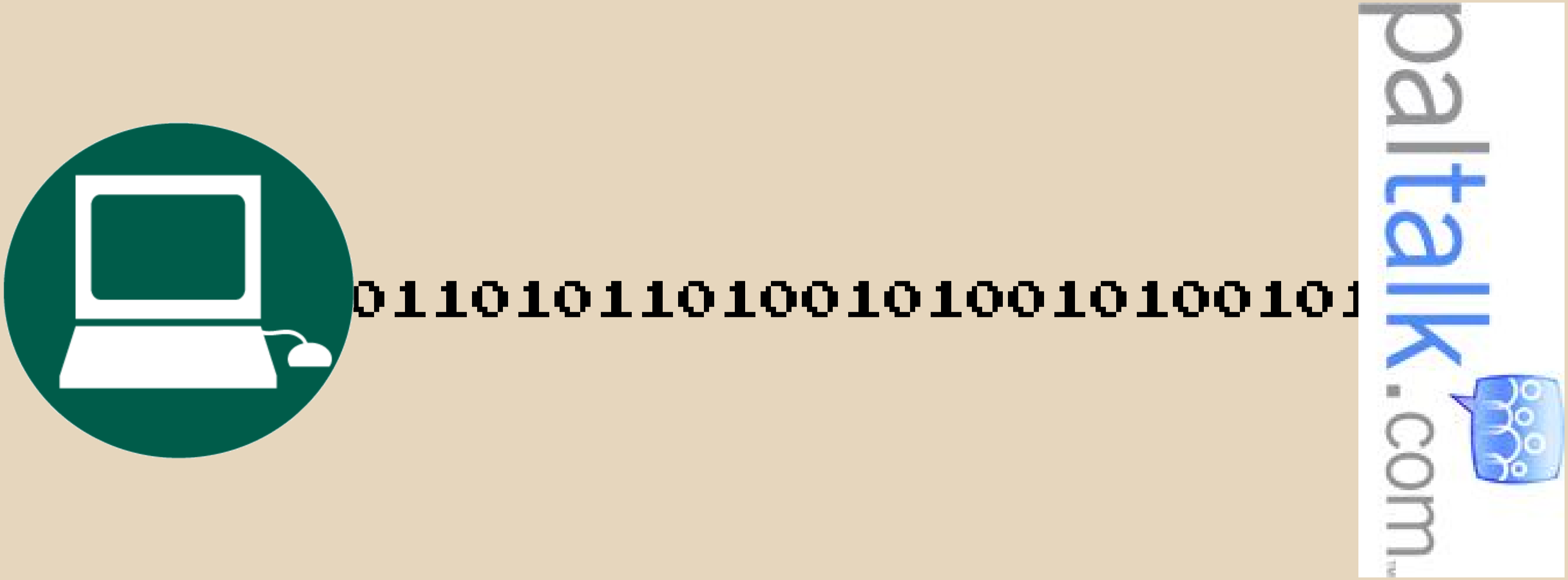
1001011010110100100100101

STREAMS===AWESOME



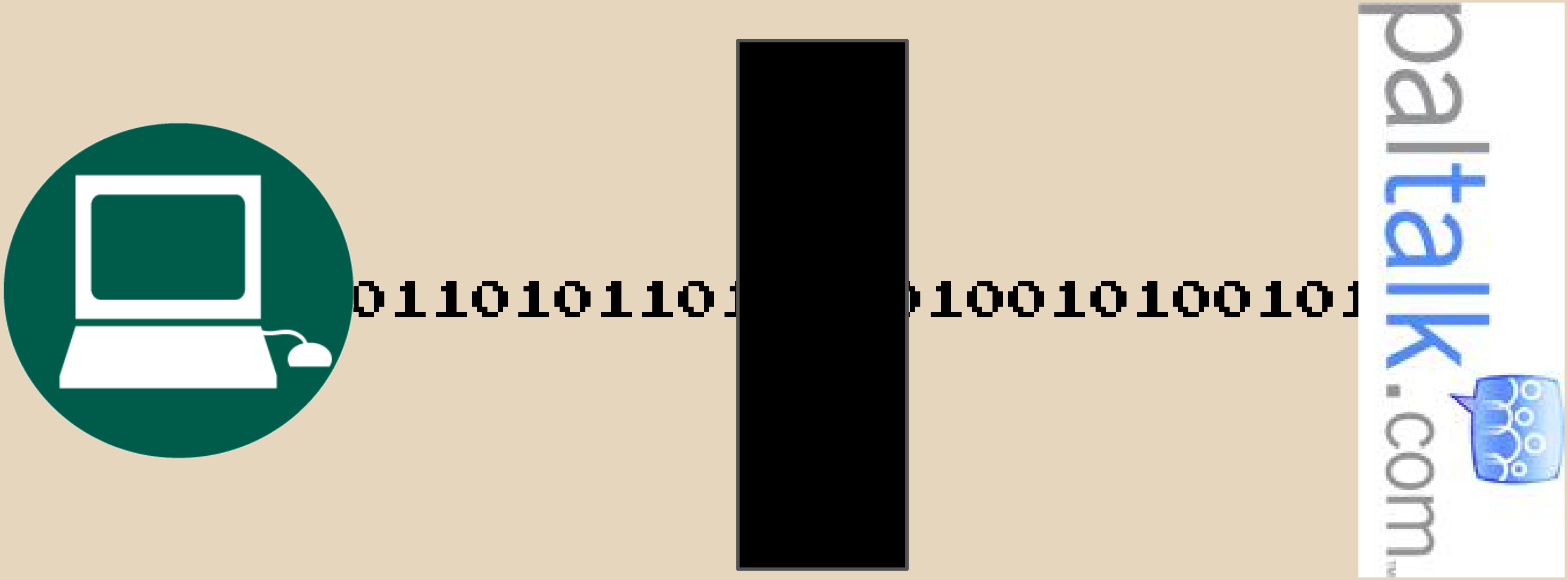
You can send files or data

STREAMS===AWESOME



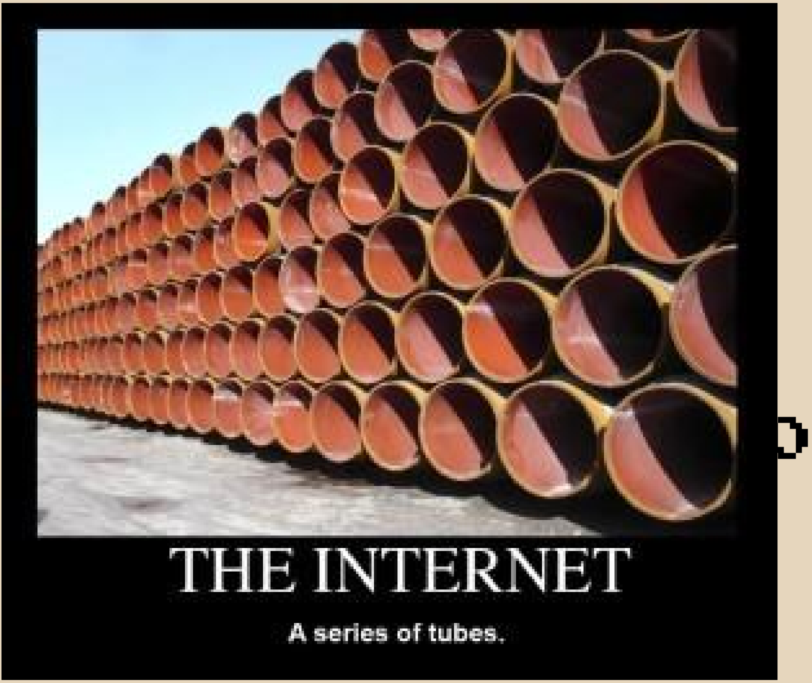
They even connect us all to PalTalk!

STREAMS===AWESOME



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

010010110001



0001010010101

You can abstract a really complicated process this way

01001011000101

10001010010101

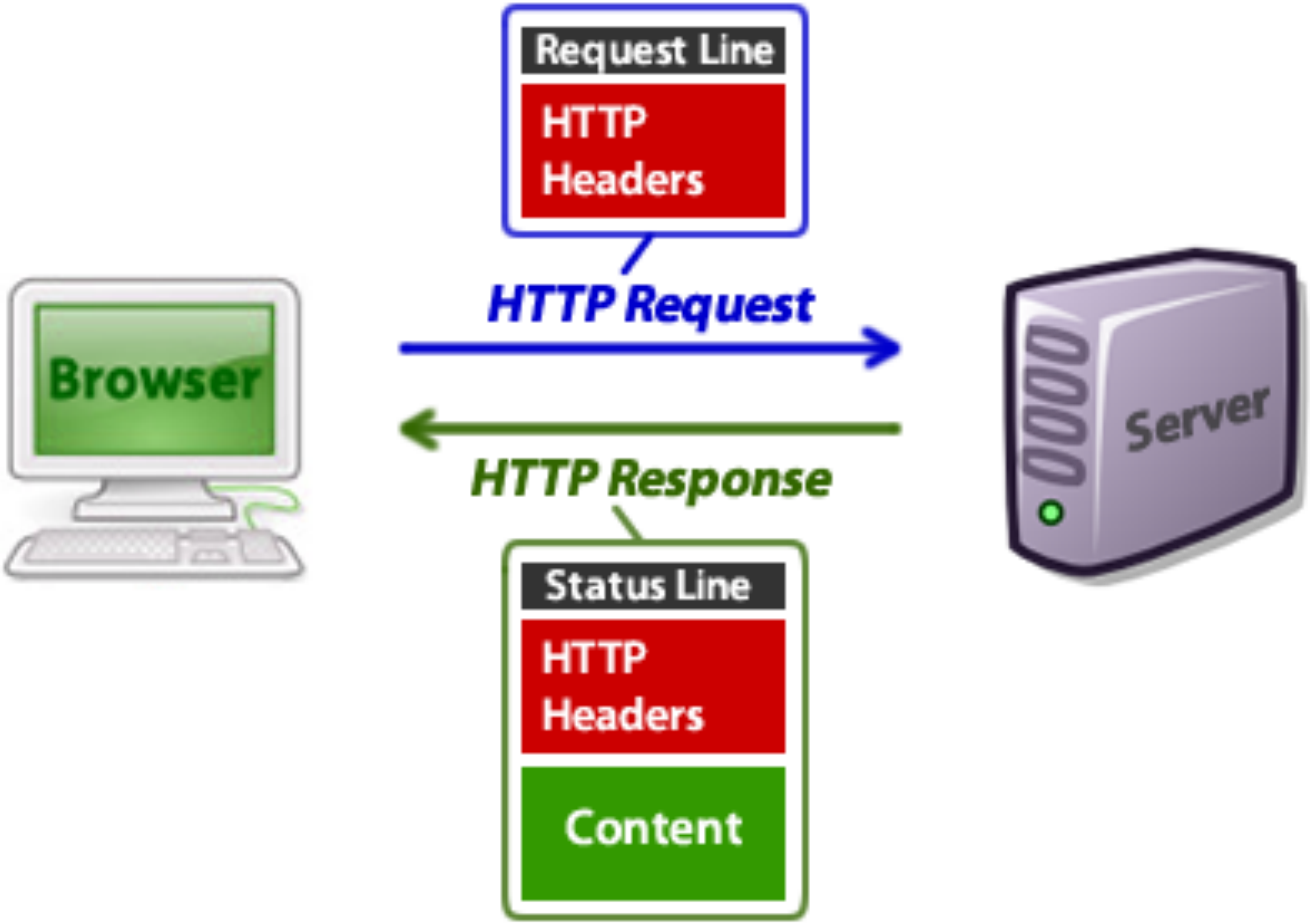
And then ignore how complicated and dysfunctional they are

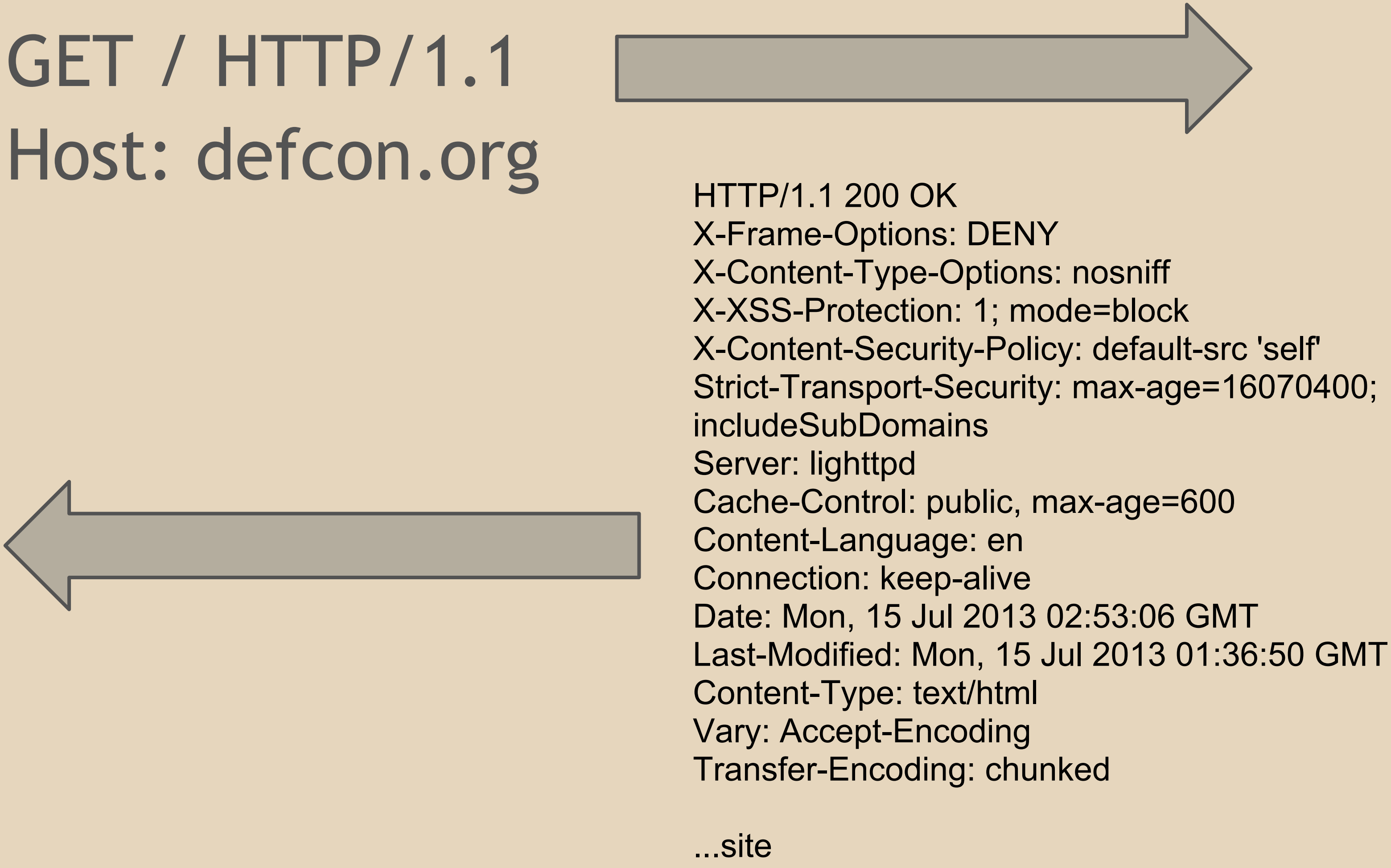
This works great for the TCP/IP

Why?

Let's look at HTTP

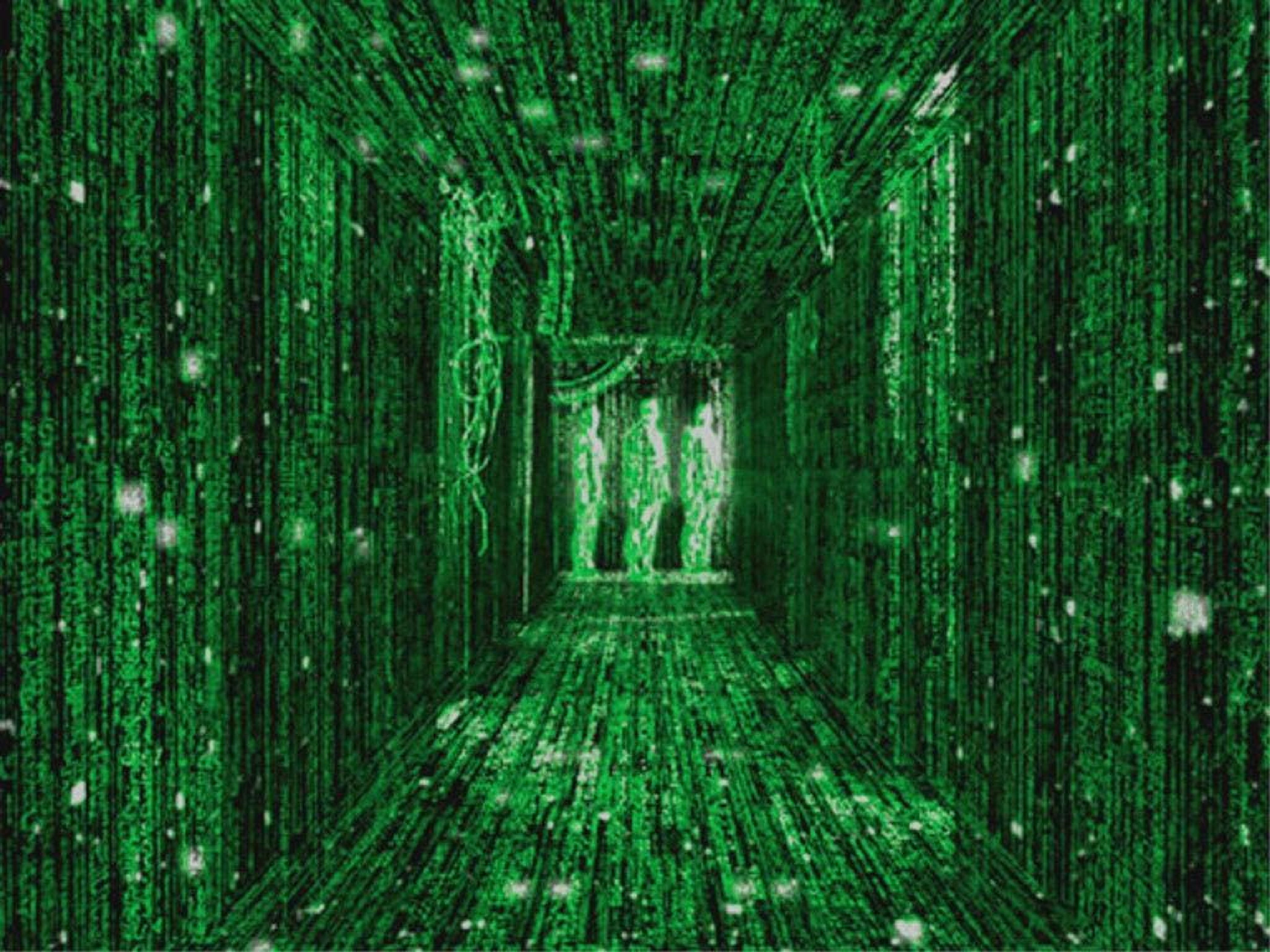
- It's so simple
- It's human readable
- Documentation isn't really necessary
- Debugging is easy
- You can encapsulate it
- You can customize it


Request Line
HTTP Headers
HTTP Request
Browser
Server
HTTP Response
Status Line
HTTP Headers
Content



```
GET / HTTP/1.1
Host: defcon.org
```

```
HTTP/1.1 200 OK
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Content-Security-Policy: default-src 'self'
Strict-Transport-Security: max-age=16070400;
includeSubDomains
Server: lighttpd
Cache-Control: public, max-age=600
Content-Language: en
Connection: keep-alive
Date: Mon, 15 Jul 2013 02:53:06 GMT
Last-Modified: Mon, 15 Jul 2013 01:36:50 GMT
Content-Type: text/html
Vary: Accept-Encoding
Transfer-Encoding: chunked

...site
```



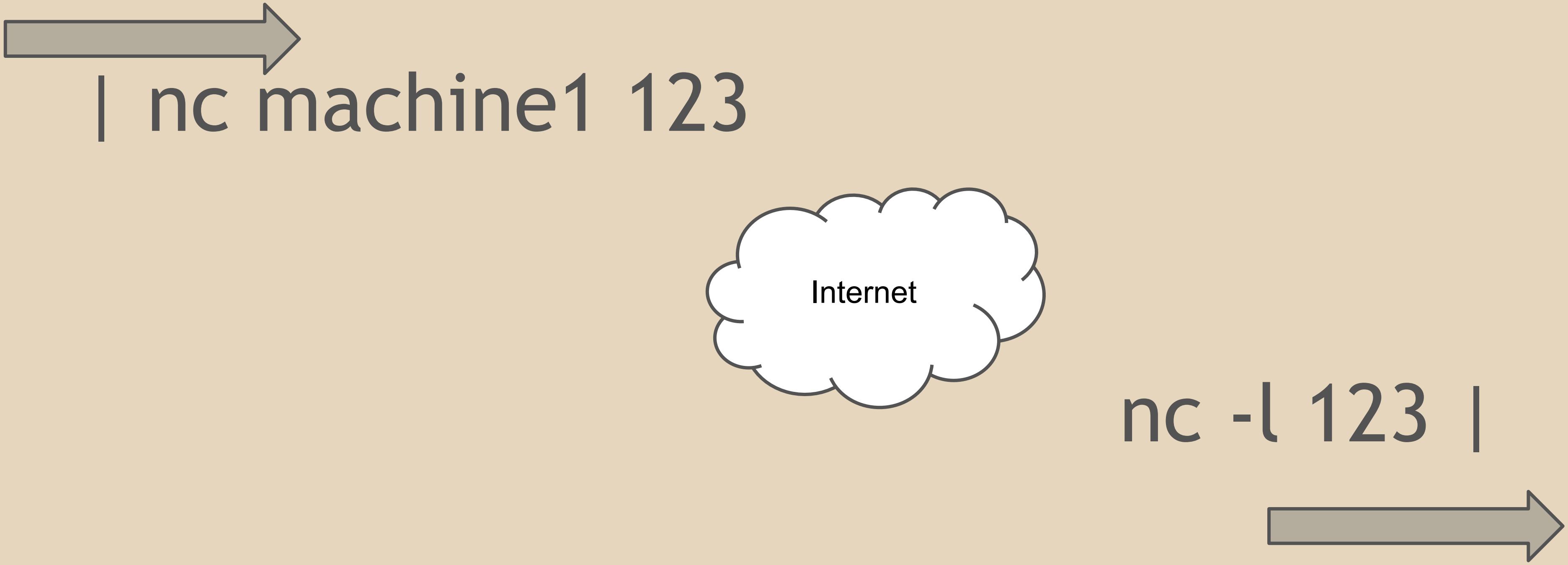
What is Blucat?

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

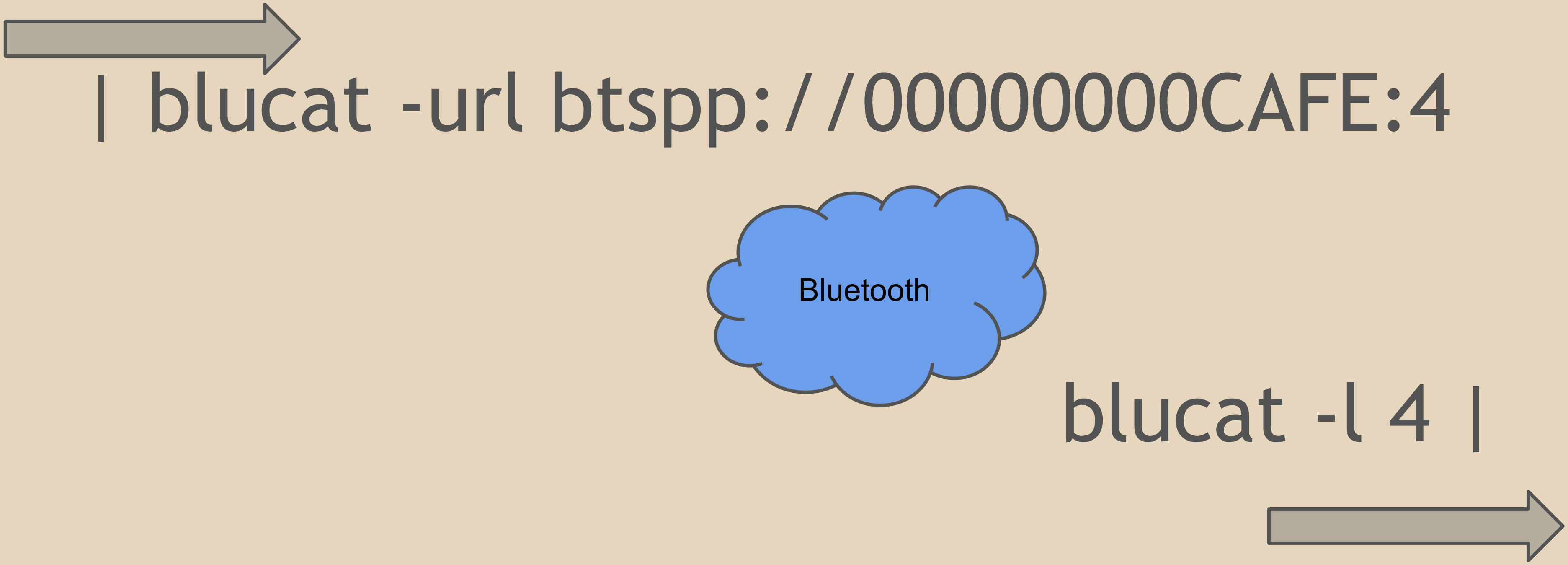
with netcat



```
| nc machine1 123
```

```
nc -l 123 |
```

with blucat



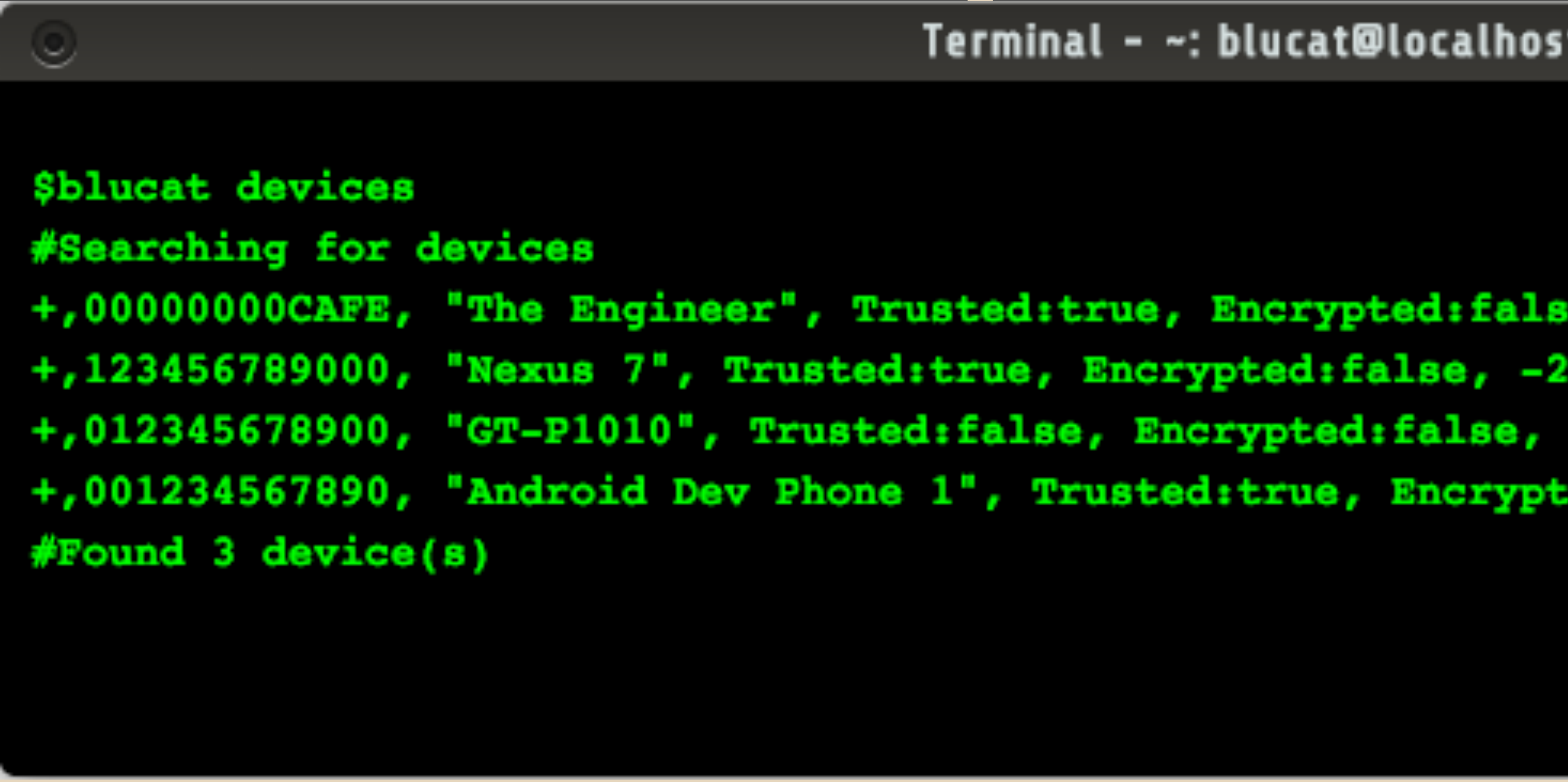
```
| blucat -url btspp://00000000CAFE:4
```

```
blucat -l 4 |
```

with nmap

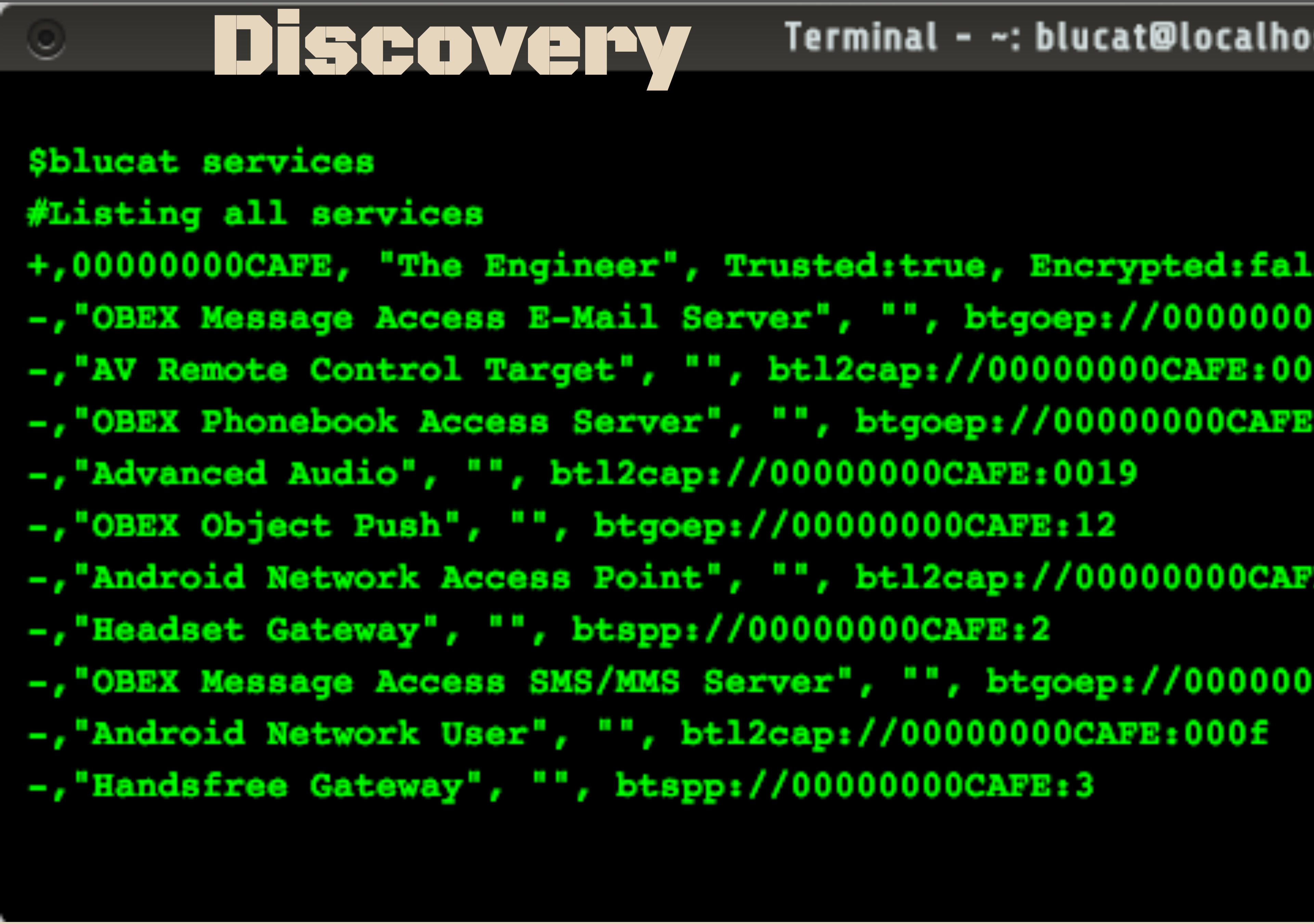
```
$nmap somehost
Starting Nmap 5.21 ( http://nmap.org )
Nmap scan report
Not shown: 846 closed ports, 152 filtered
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Discovery



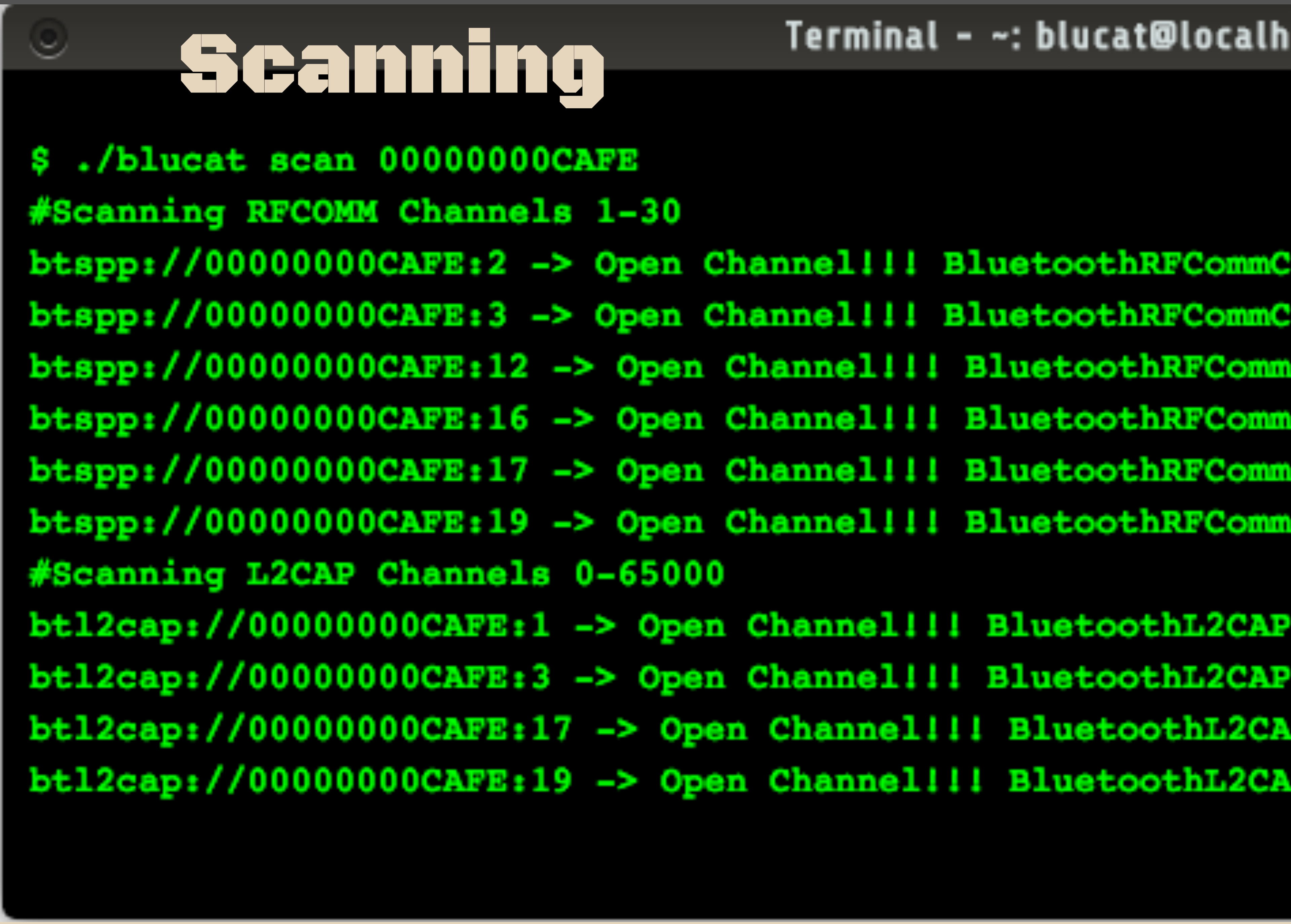
```
$blucat devices
#Searching for devices
+,00000000CAFE, "The Engineer", Trusted:true, Encrypted:fals
+,123456789000, "Nexus 7", Trusted:true, Encrypted:false, -2
+,012345678900, "GT-P1010", Trusted:false, Encrypted:false,
+,001234567890, "Android Dev Phone 1", Trusted:true, Encrypt
#Found 3 device(s)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

Discovery

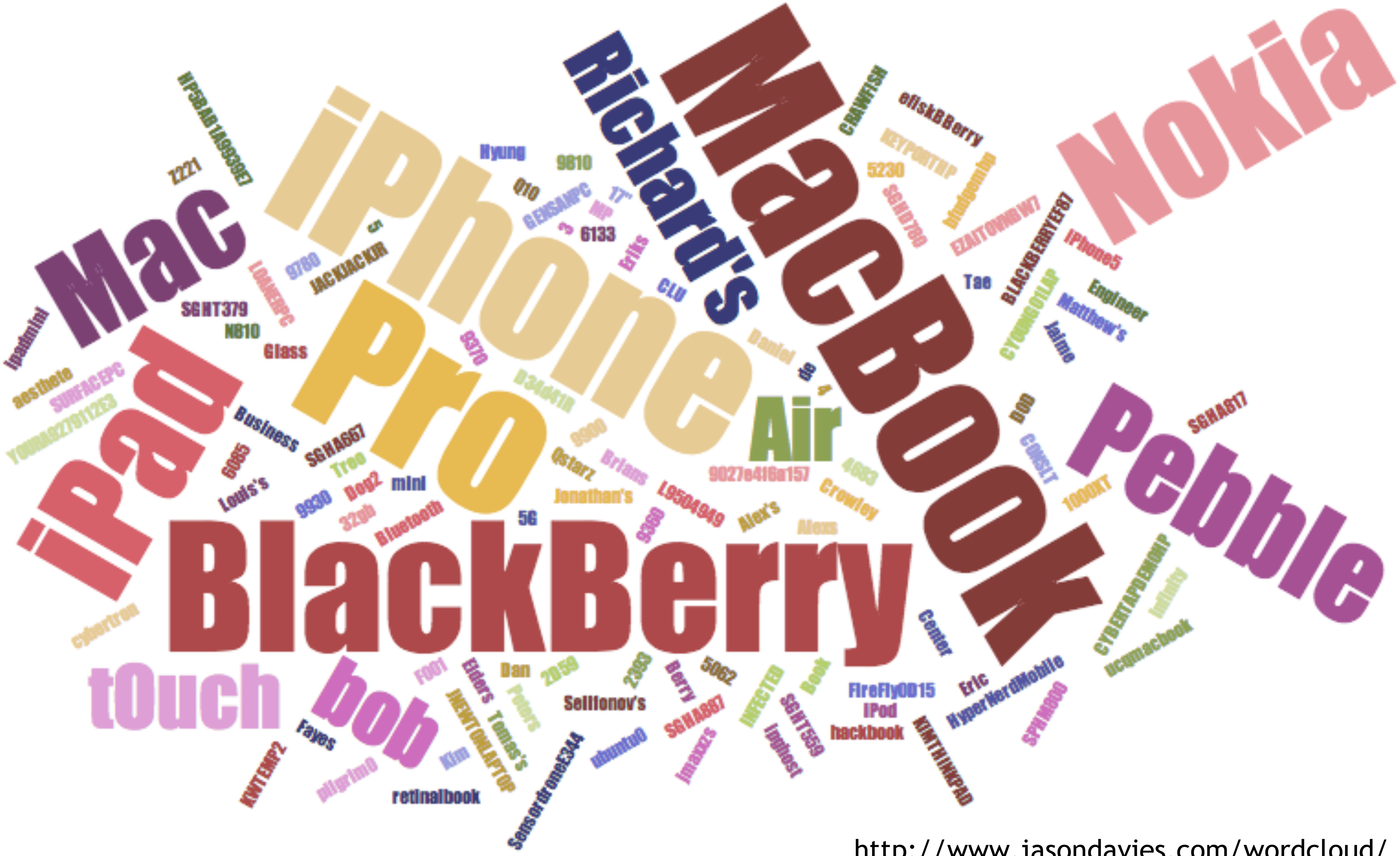


```
$blucat services
#Listing all services
+,00000000CAFE, "The Engineer", Trusted:true, Encrypted:fal
-,"OBEX Message Access E-Mail Server", "", btgoep://0000000
-,"AV Remote Control Target", "", btl2cap://00000000CAFE:00
-,"OBEX Phonebook Access Server", "", btgoep://00000000CAFE
-,"Advanced Audio", "", 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
```



http://www.jasondavies.com/wordcloud/

Defcon 2013 (Thursday-Saturday) visible bluetooth devices

Defcon 2013 Bluetooth Statistics

```
 $sort names | uniq | wc -l
      92

$ cat bdaddr | sort | uniq | wc -l
     126

$cat pairingrequests | wc -l
    1367
```

Best Bluetooth Device Names @DC13

hackbook

INFECTED

HyperNerd-Mobile

DOD

SensordroneE344

cybertron

tOuch-mE-5G

Bluetooth URI Monikers

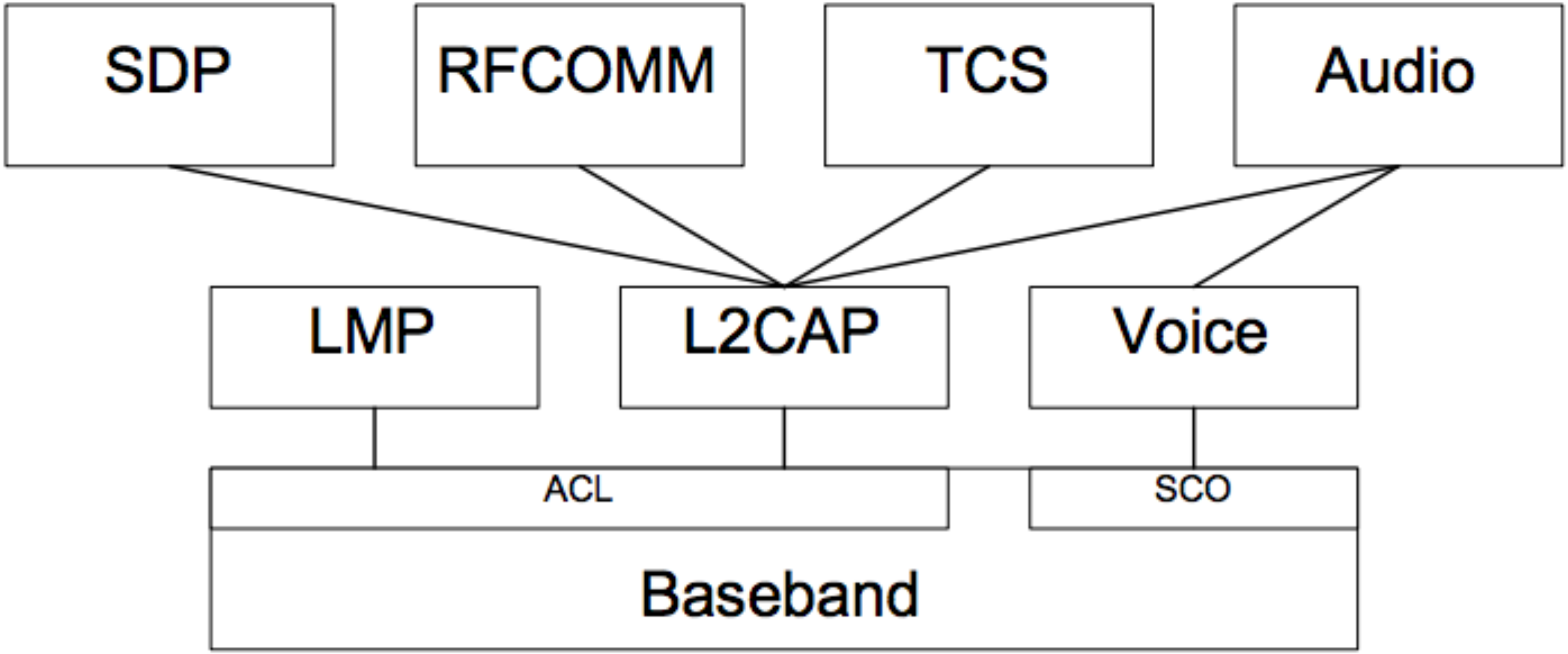
ex: btspp://10643FC98386:17

Bluetooth URI Monikers

btspp -
Bluetooth serial port profile RFCOMM

btl2cap -
Logical link control and adaptation protocol

btgoep -
OBEX Generic Object Exchange profile



L2CAP in Bluetooth Protocol Architecture

serial port profile (SPP)

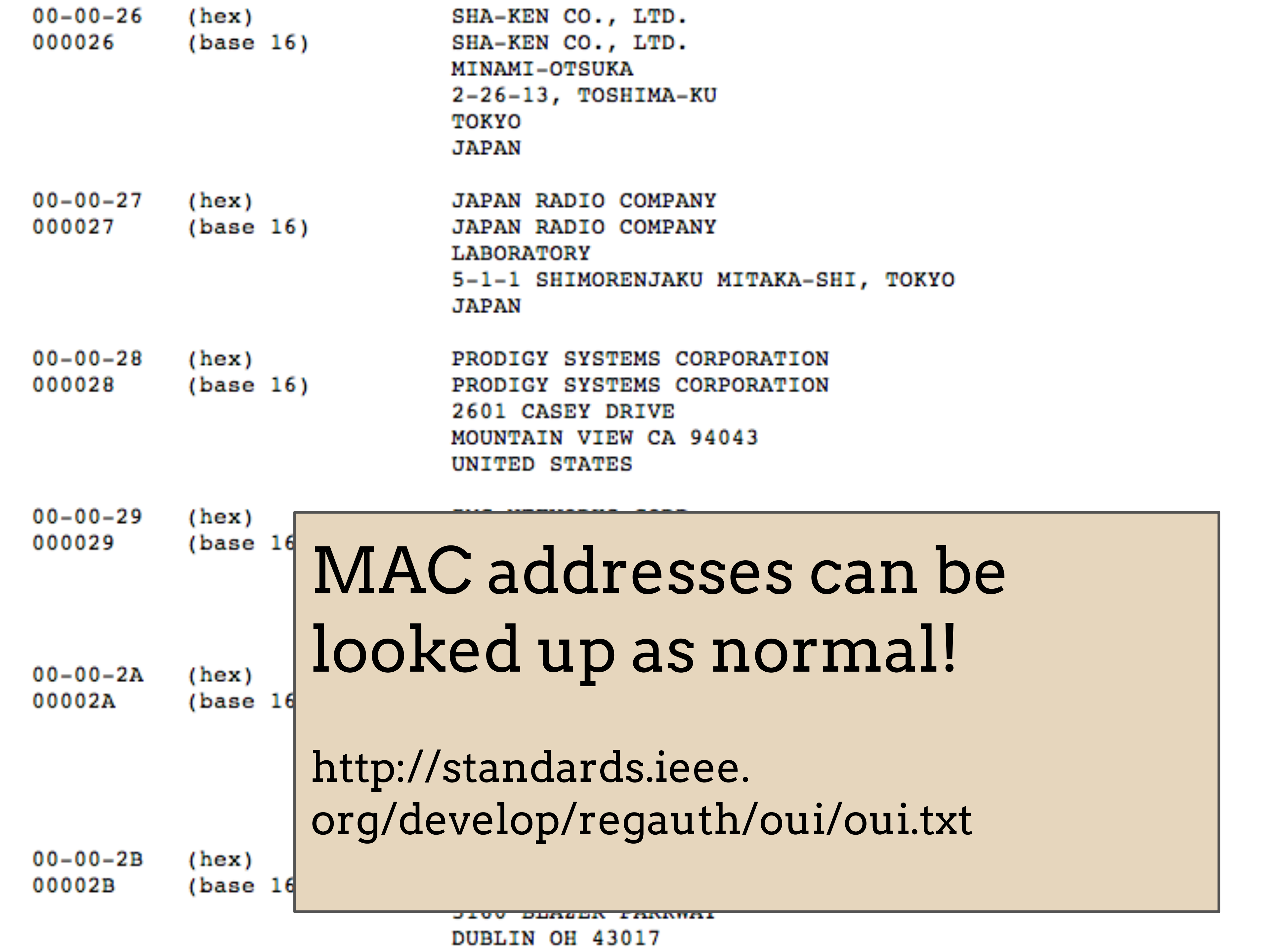
- designed to emulate RS-232 serial ports
- same major attributes of TCP sockets
    - in order, retry,
- only allows ~30 ports
    - depends on stack
    - assigned dynamically like portmap (TCP/111)

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

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   (hex)              IMC NETWORKS CORP.
000029     (base 16

00-00-2A   (hex)
00002A     (base 16

00-00-2B   (hex)
00002B     (base 16
                              5100 BLAZER PARKWAY
                              DUBLIN OH 43017
```

MAC addresses can be looked up as normal!

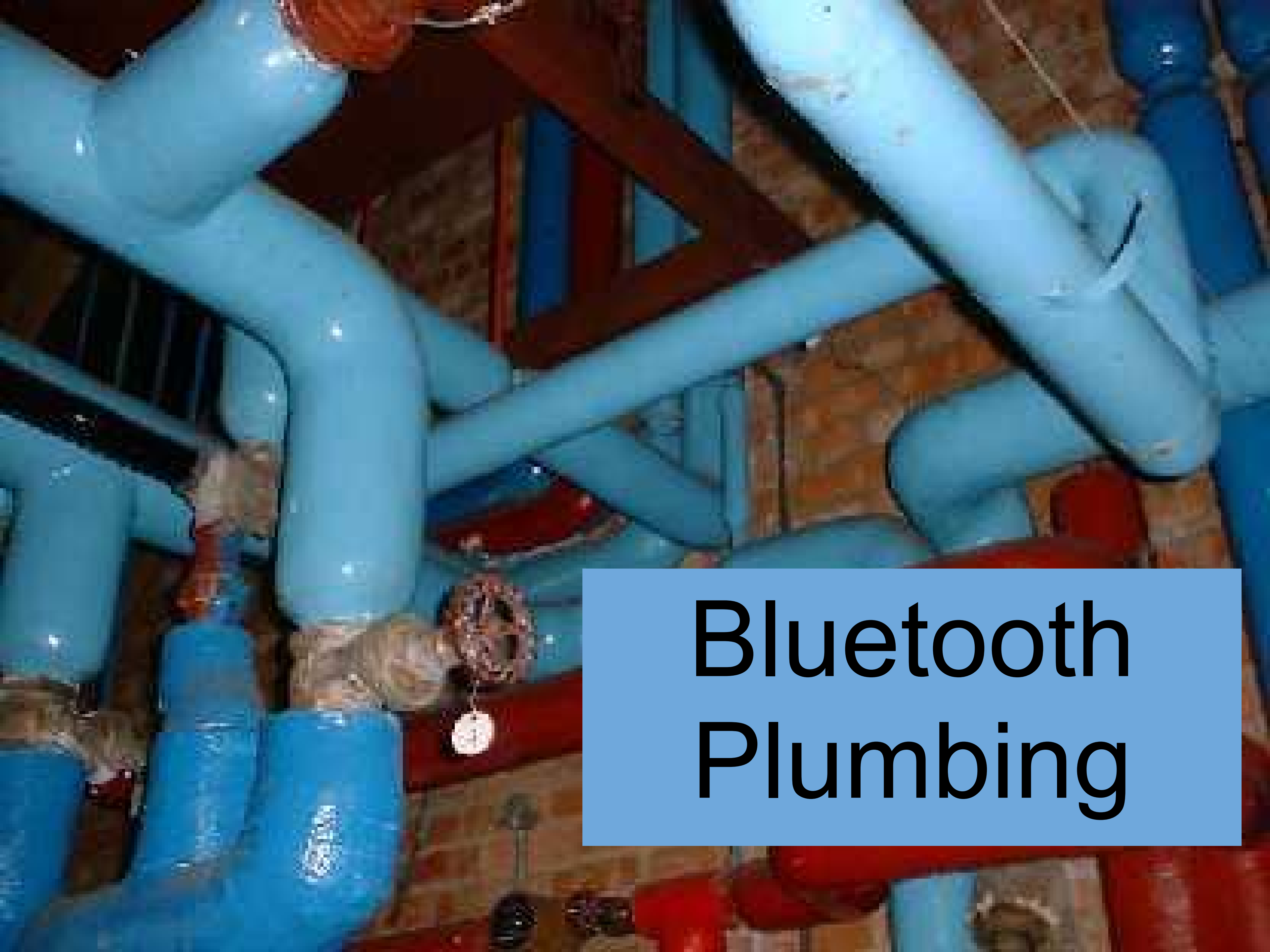
http://standards.ieee.org/develop/regauth/oui/oui.txt

On connect execution!

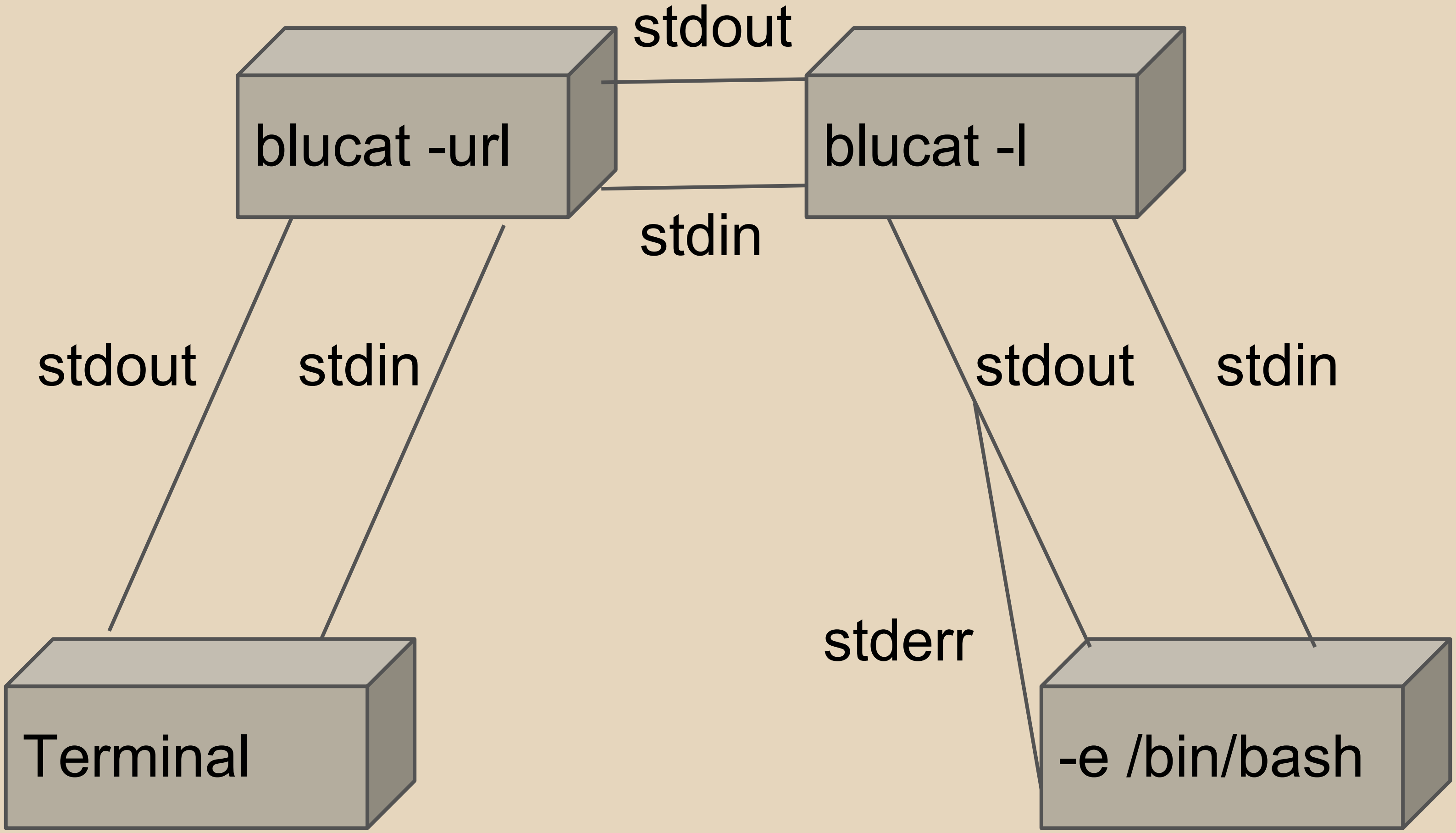
```
$./blucat -v -l -e /bin/bash
#Listening at btspp://002608AAAAAA:4
```

```
$./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

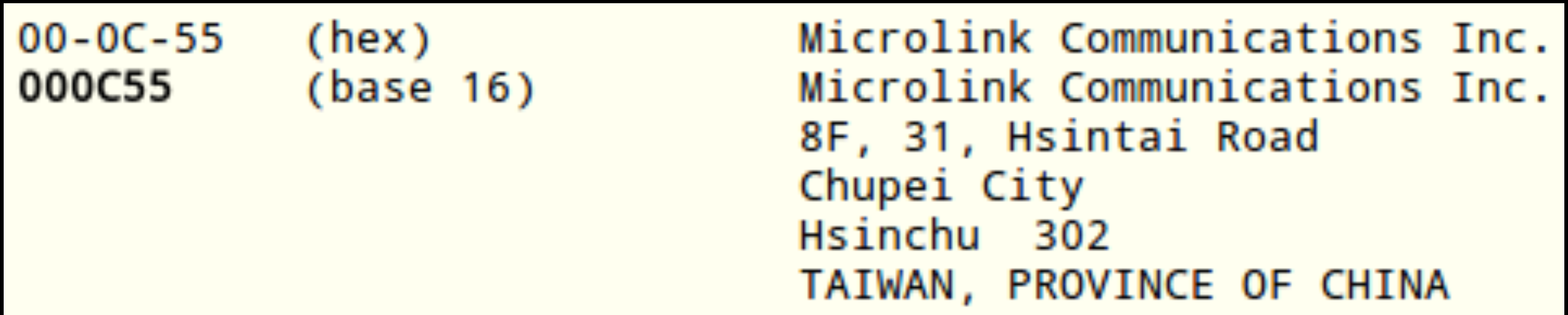
Inspecting devices

Bluetooth has “profiles”

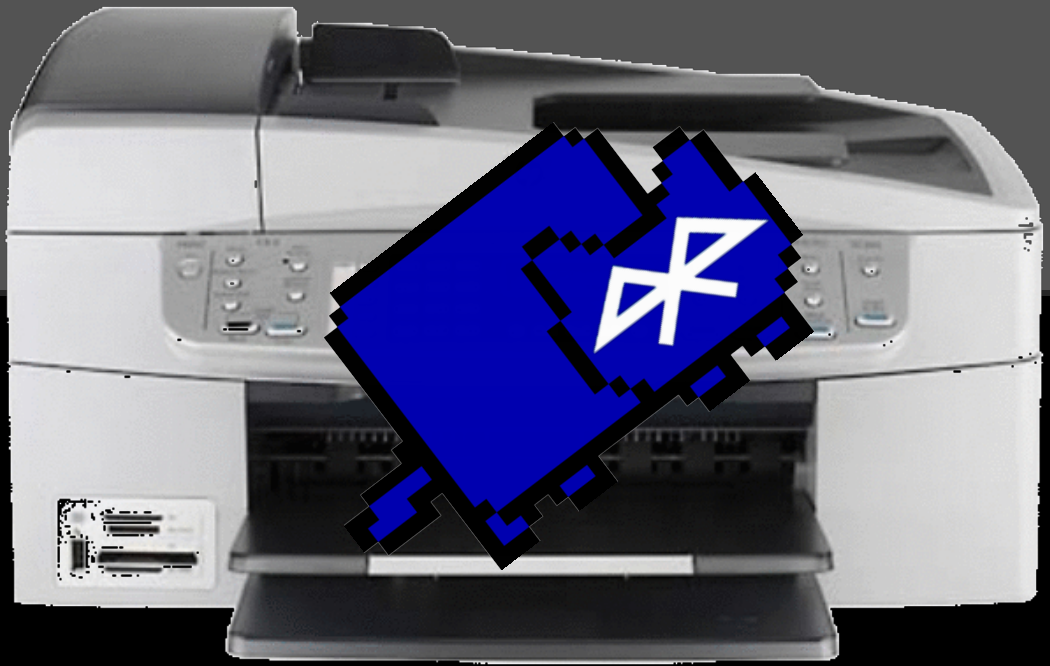
Identified by UUID and device class

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

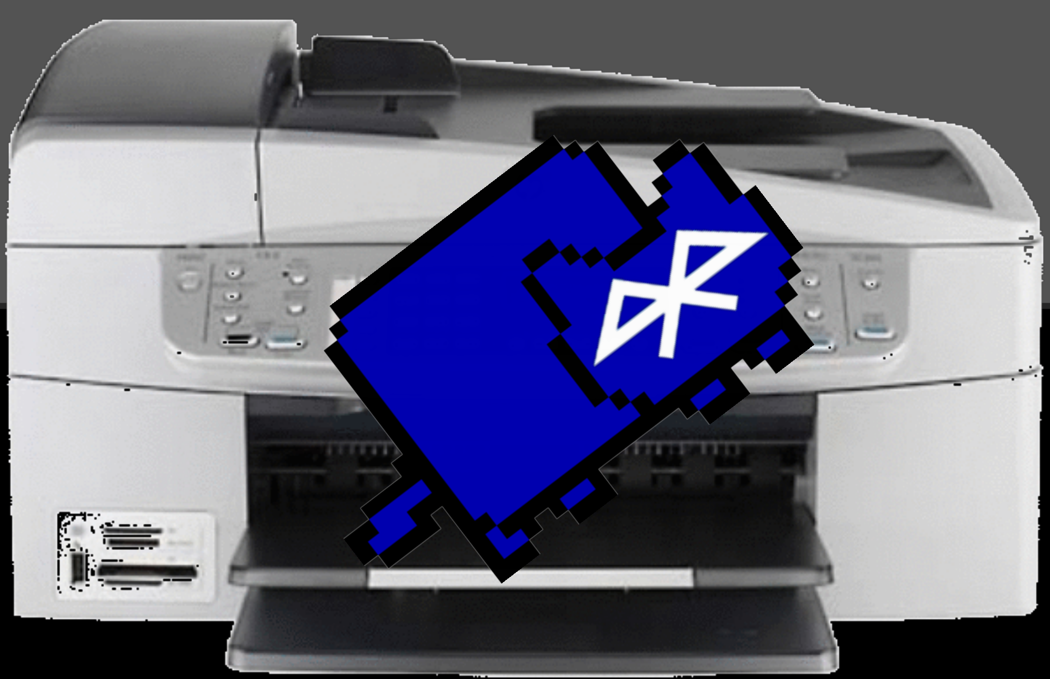
000C55F8FBEE, "Officejet 6300 series"



```
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
```

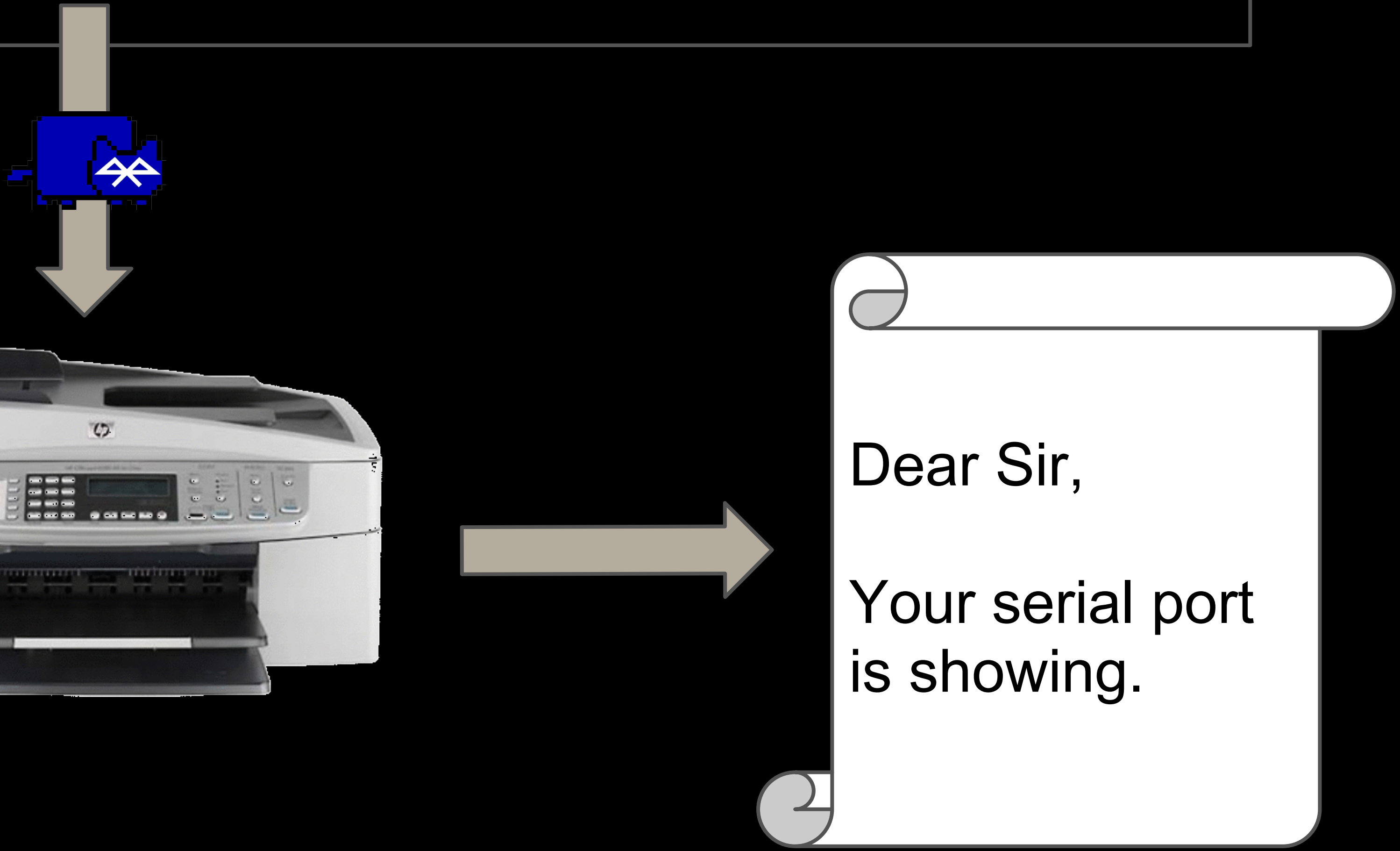


```
30F306AAAAAA, "Officejet 6300 series", Trusted:false, ...
"OBEX Object Push", "", btgoep://30F306598203:2
"Serial Port", "", btspp://30F306598203:1
"Basic Printing", "", btgoep://30F306598203:4
"Basic Imaging", "", btgoep://30F306598203:3
```



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

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

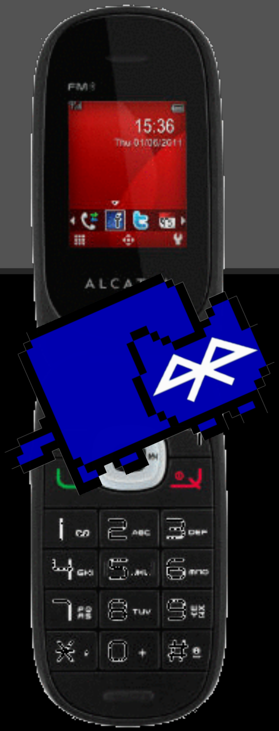


Dear Sir,

Your serial port is showing.

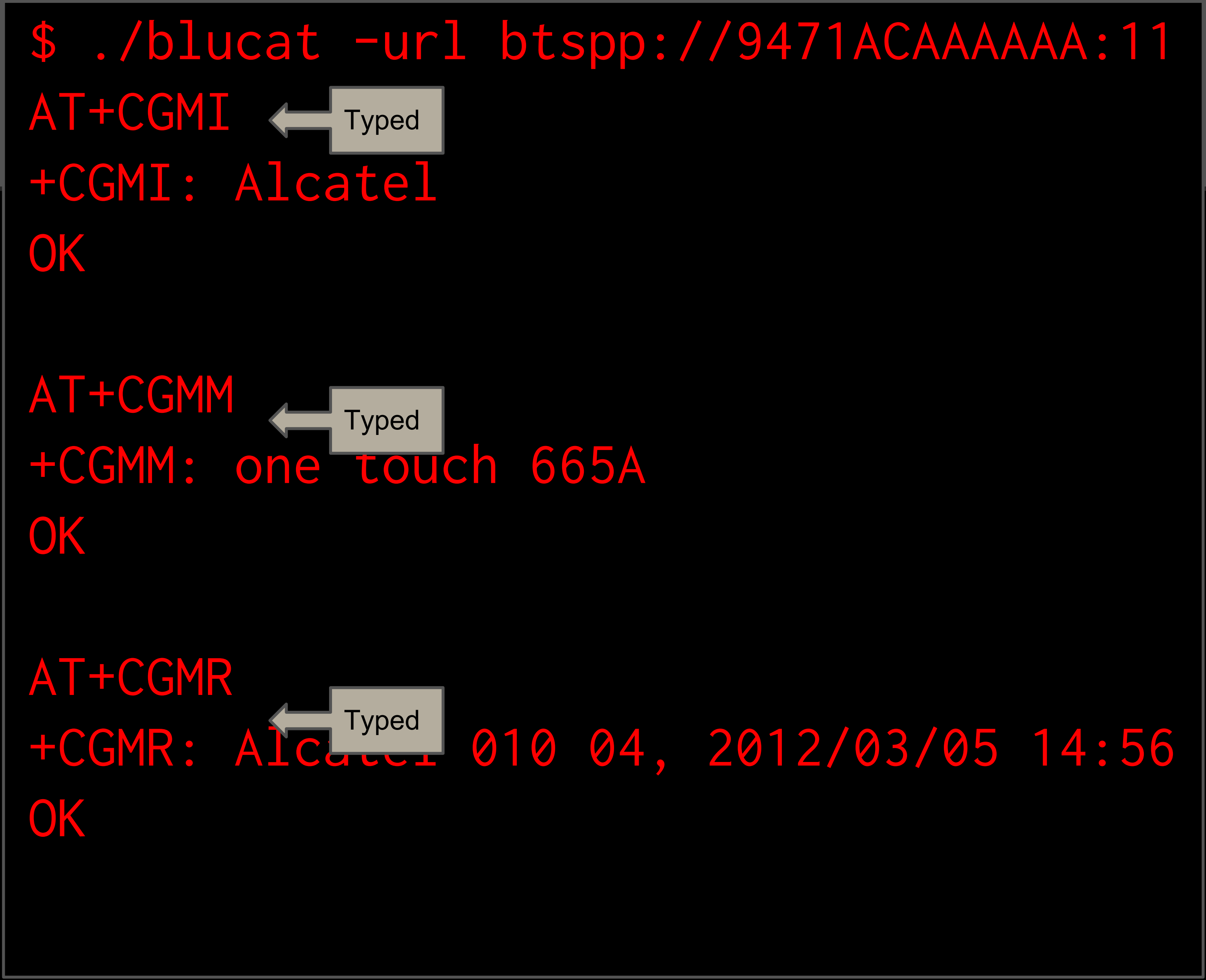
FM
15:36
Thu 01/06/2011
ALCATEL

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
+CGMI: Alcatel
OK

AT+CGMM
+CGMM: one touch 665A
OK

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

Typed

Typed

Typed

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
CE0168
4:20
Google
You Tube
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))
```

IhPone iAP service

iPod Accessory Protocol

-,"Wireless iAP", "", btspp://34C059AAAAAA:1

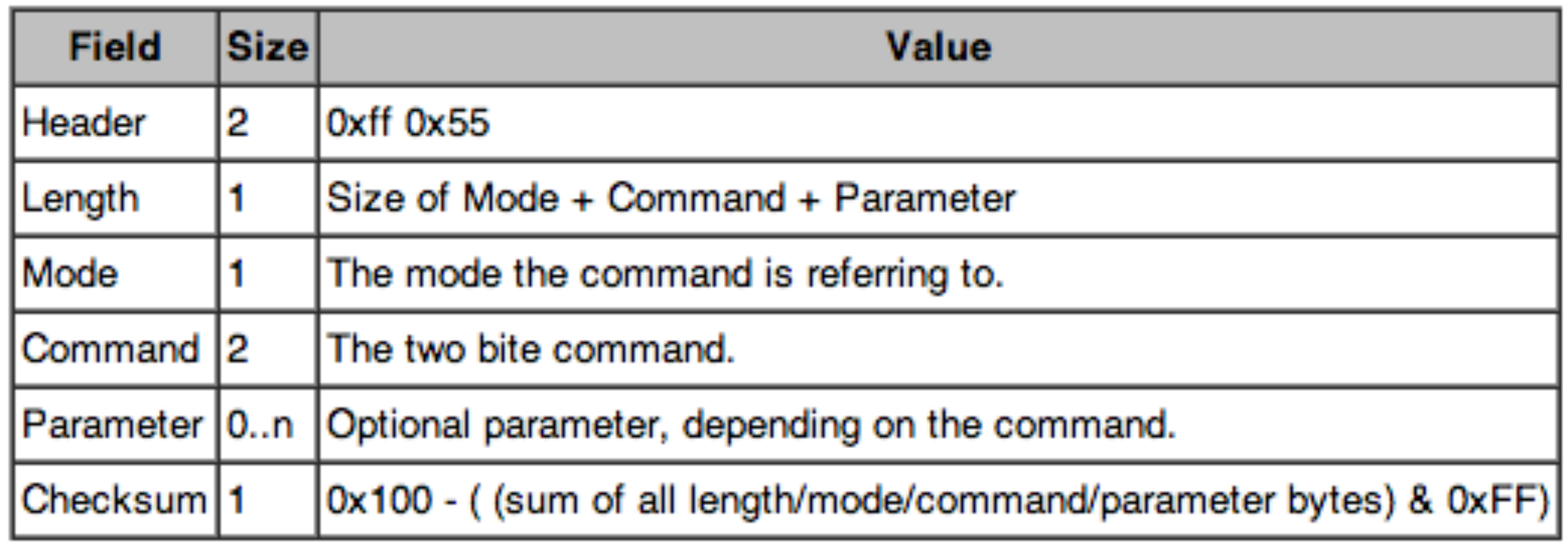
Explored with Alex Whittemore @DC13

Goal to play/stop/control audio and tracks

Should be the same as interacting with standard UART in wire Apple connector

IhPone iAP service

iPod Accessory Protocol



Field	Size	Value
Header	2	0xff 0x55
Length	1	Size of Mode + Command + Parameter
Mode	1	The mode the command is referring to.
Command	2	The two bite command.
Parameter	0..n	Optional parameter, depending on the command.
Checksum	1	0x100 - ((sum of all length/mode/command/parameter bytes) & 0xFF)

https://nuxx.net/wiki/Apple_Accessory_Protocol

IhPone iAP service

iPod Accessory Protocol

Speaking the protocol only made the ihpone say “This accessory is not supported”

“...establishing Bluetooth data connections with Apple devices requires a unique discovery/pairing sequence and negotiation with the Apple authentication co-processor”

Soo, this service requires a special chip from apple

Rapid prototyping with Blucat

How to prototype

Current presentation is using blucat

- 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

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
...
```

Data!

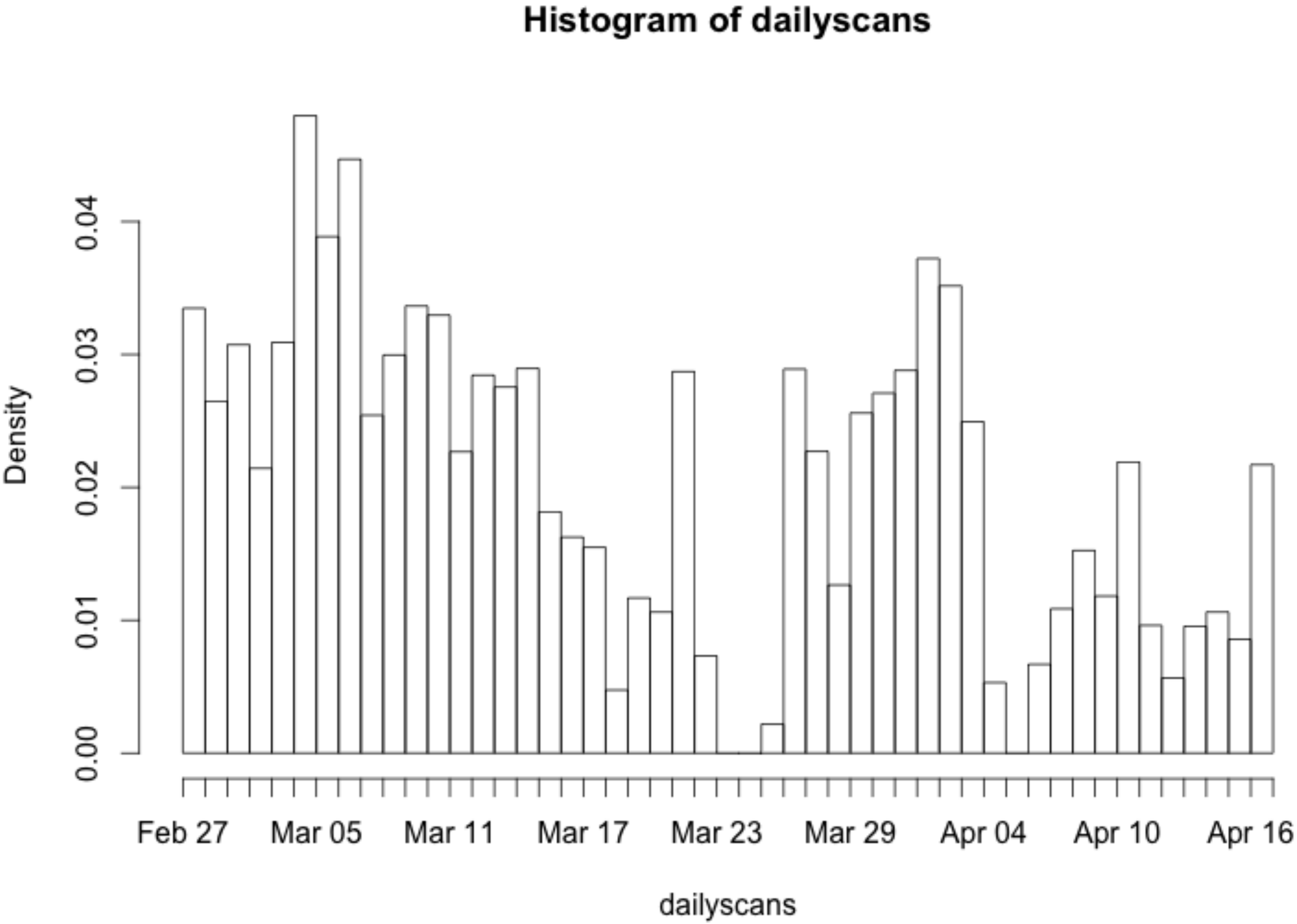
Scanning every 5 minutes from fixed location

Bluetooth devices are set to visible

blucat outputs in csv format

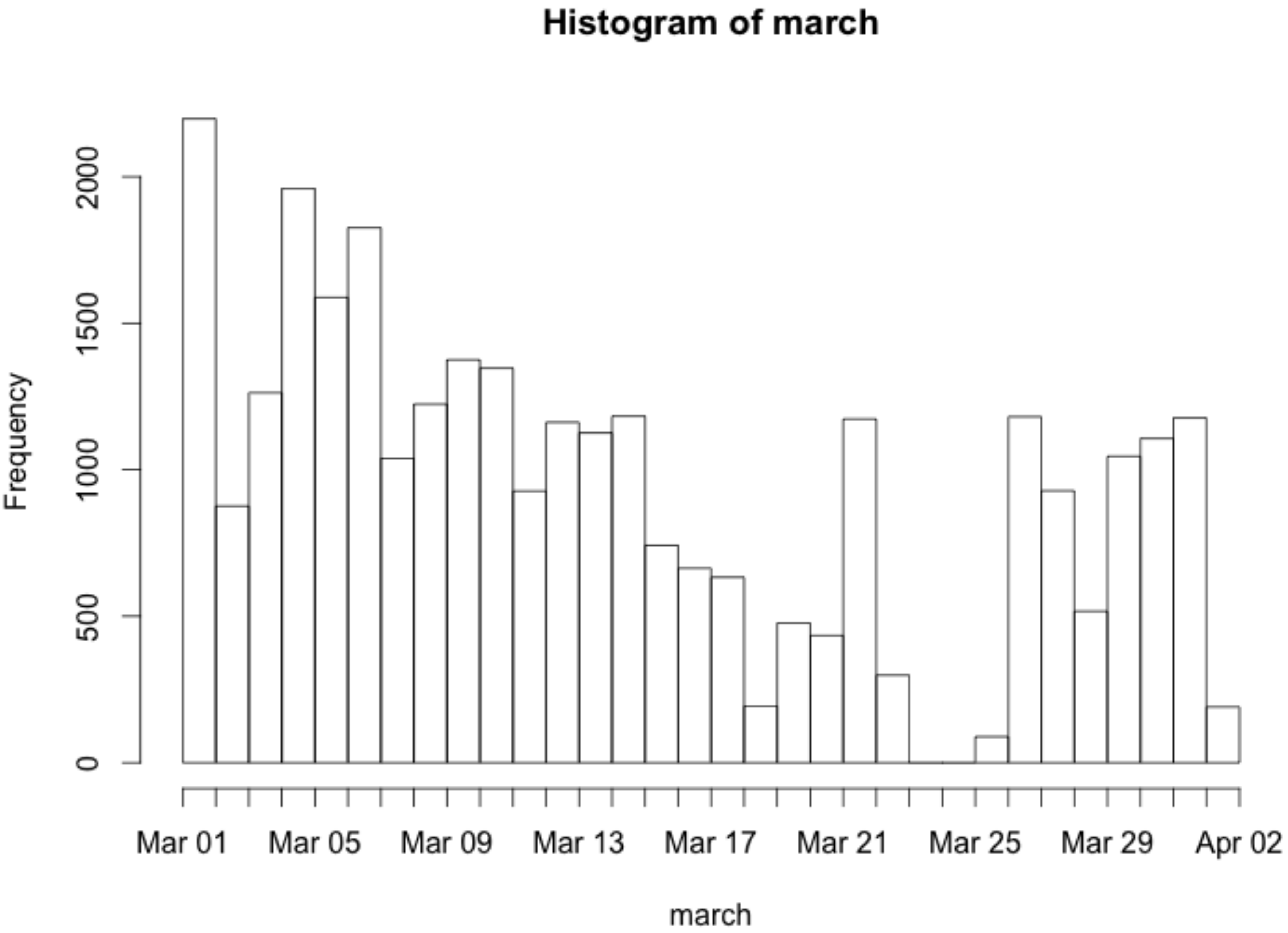
System-R

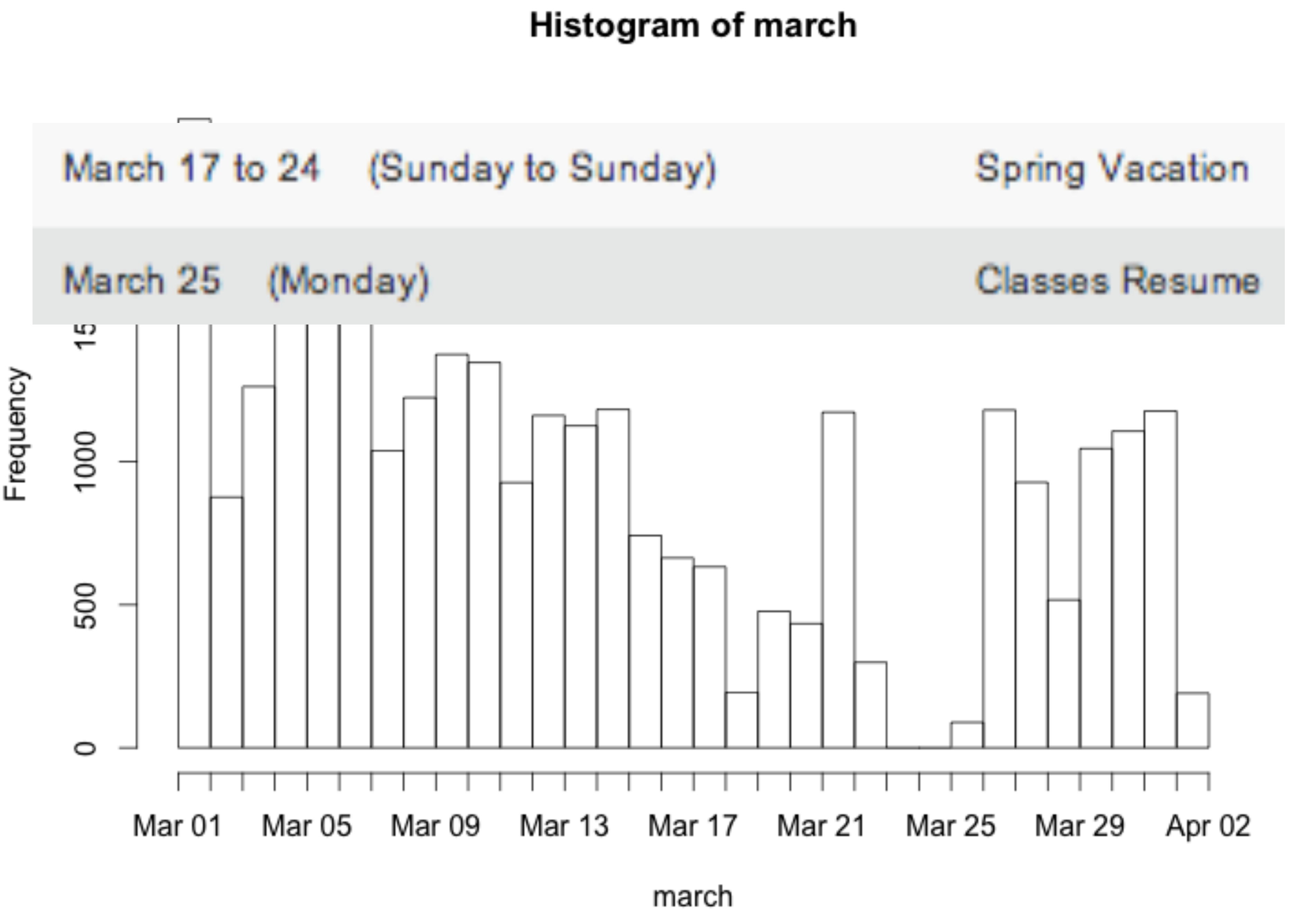
```
file = "logs.csv"
data = read.csv(file=file,header=T, row.names=NULL);

library(zoo)
dailyscans = as.Date(as.POSIXct(data[,2]/1000,
                                        origin="1970-01-01"))

hist(dailyscans,breaks=100,freq=T)
```


Histogram of dailyscans
Density
0.00
0.01
0.02
0.03
0.04
Feb 27
Mar 05
Mar 11
Mar 17
Mar 23
Mar 29
Apr 04
Apr 10
Apr 16
dailyscans

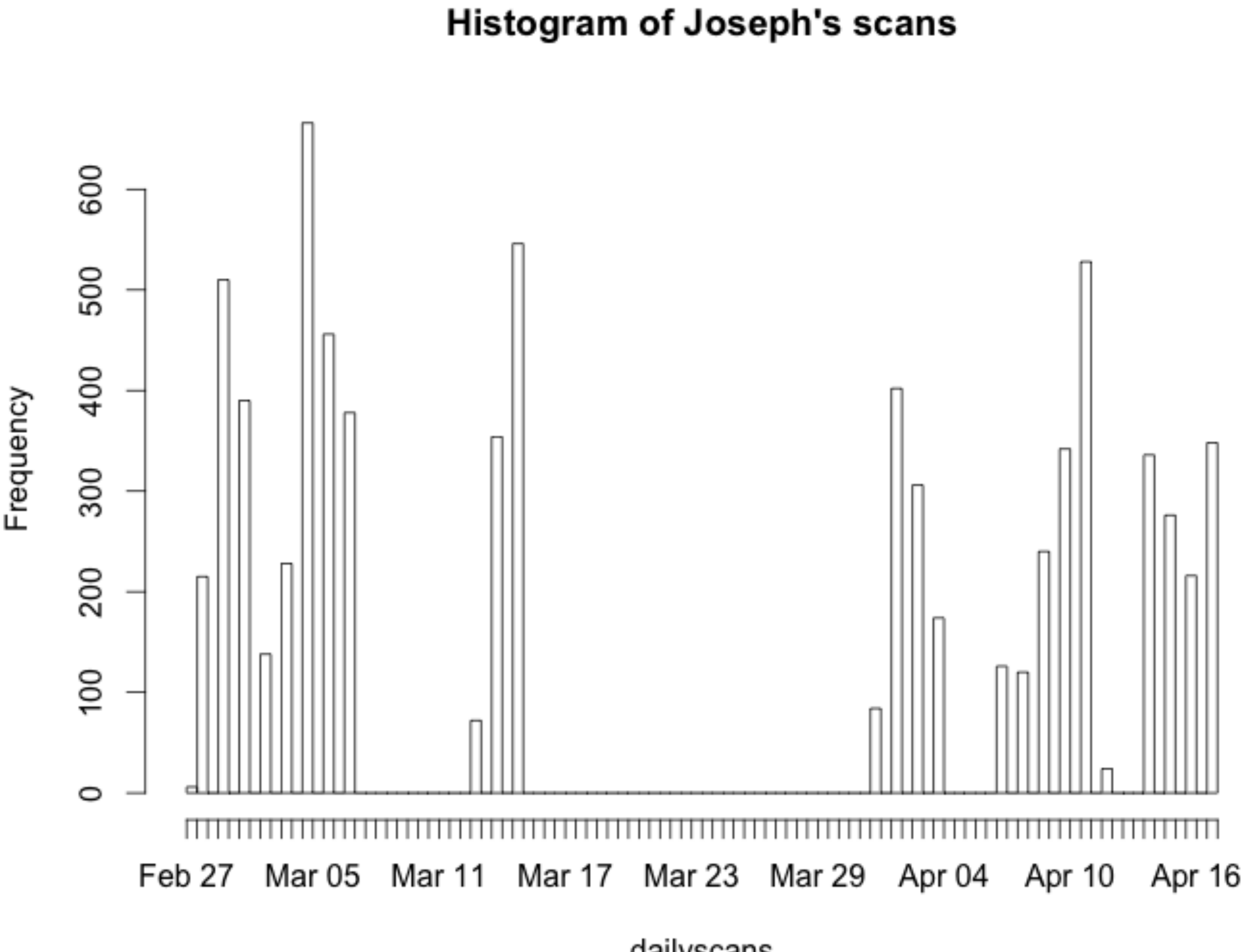
System-R

```
march = dailyscans
          [dailyscans>="2013-03-01"][dailyscans<"2013-4-01"]
```


Histogram of march
Frequency
0
500
1000
1500
2000
Mar 01
Mar 05
Mar 09
Mar 13
Mar 17
Mar 21
Mar 25
Mar 29
Apr 02
march



March 17 to 24	(Sunday to Sunday)	Spring Vacation
March 25	(Monday)	Classes Resume


Histogram of Joseph's scans
Frequency
0
100
200
300
400
500
600
Feb 27
Mar 05
Mar 11
Mar 17
Mar 23
Mar 29
Apr 04
Apr 10
Apr 16
dailyscans

Details

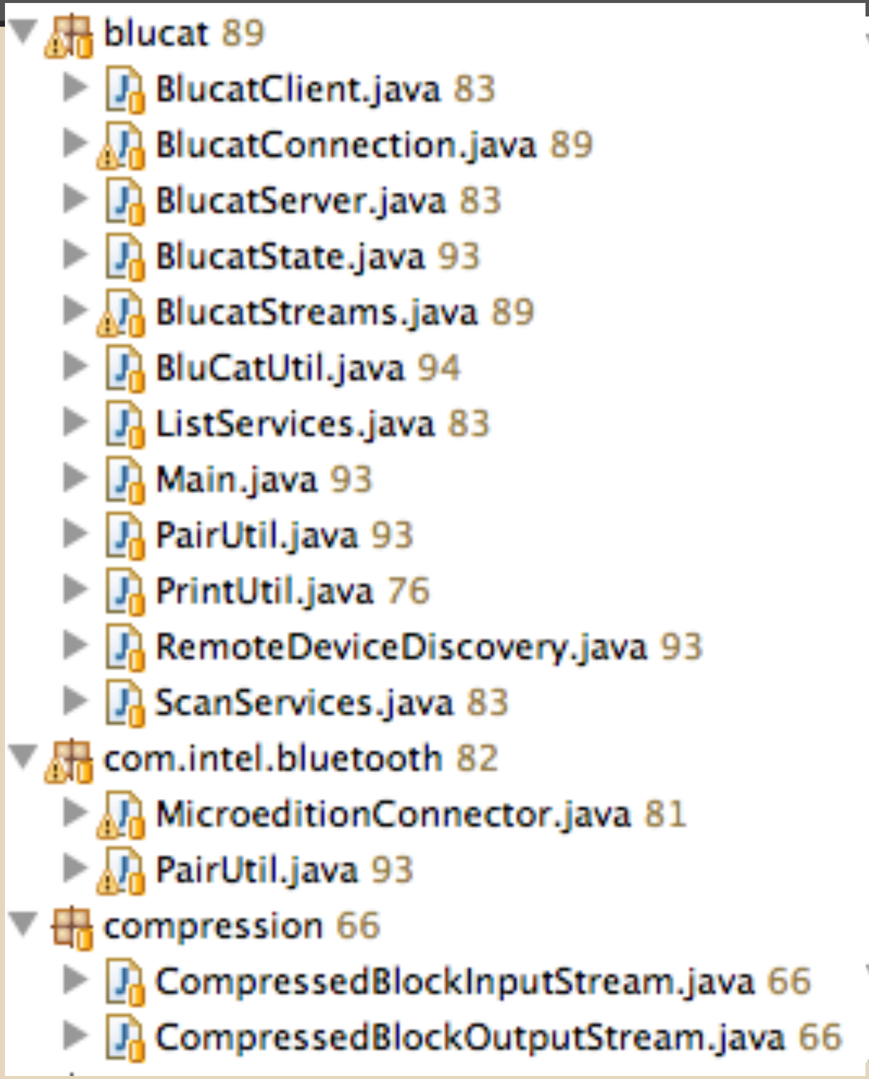
Java based

Uses BlueCove Java Libraries

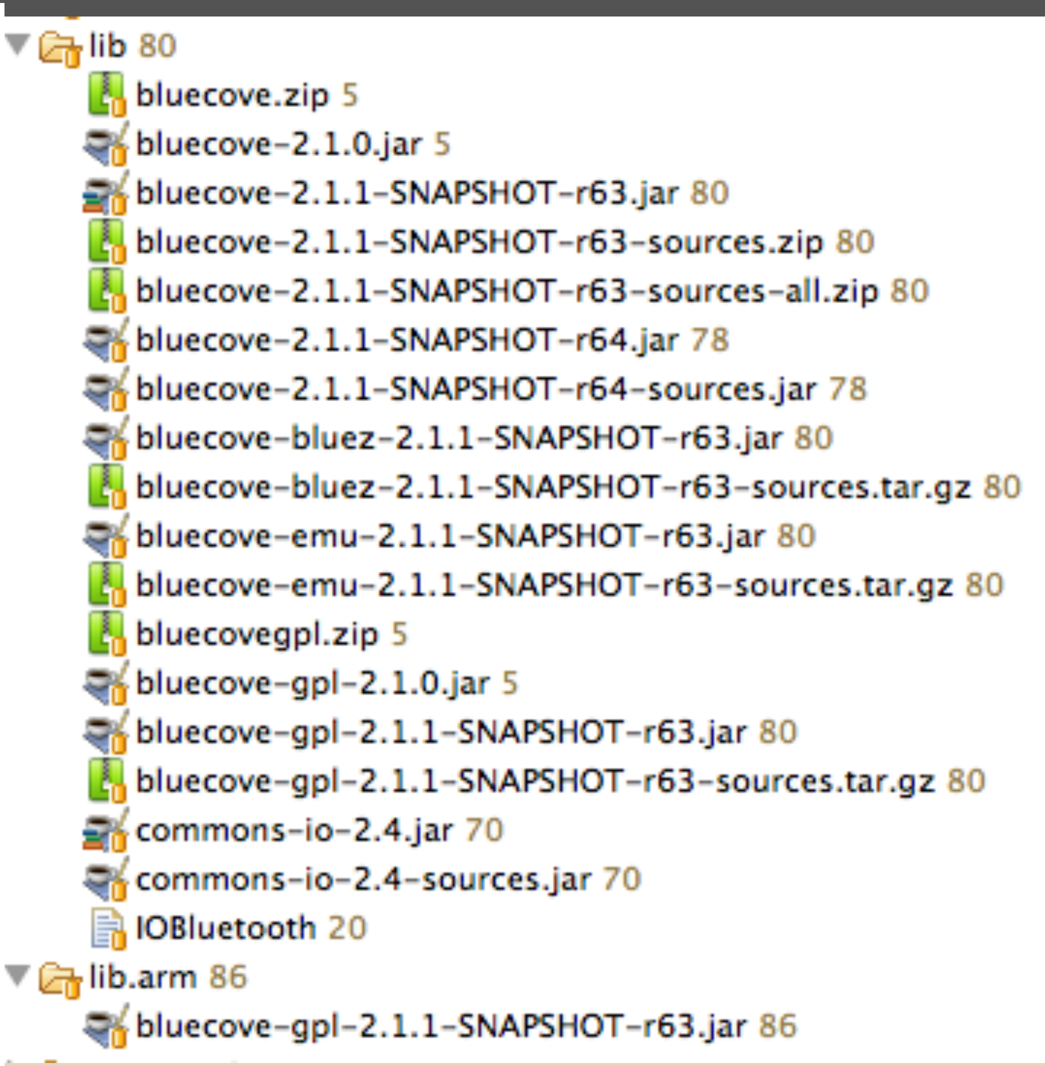
Tested on Mac and many Linux versions using Bluez

State of the code

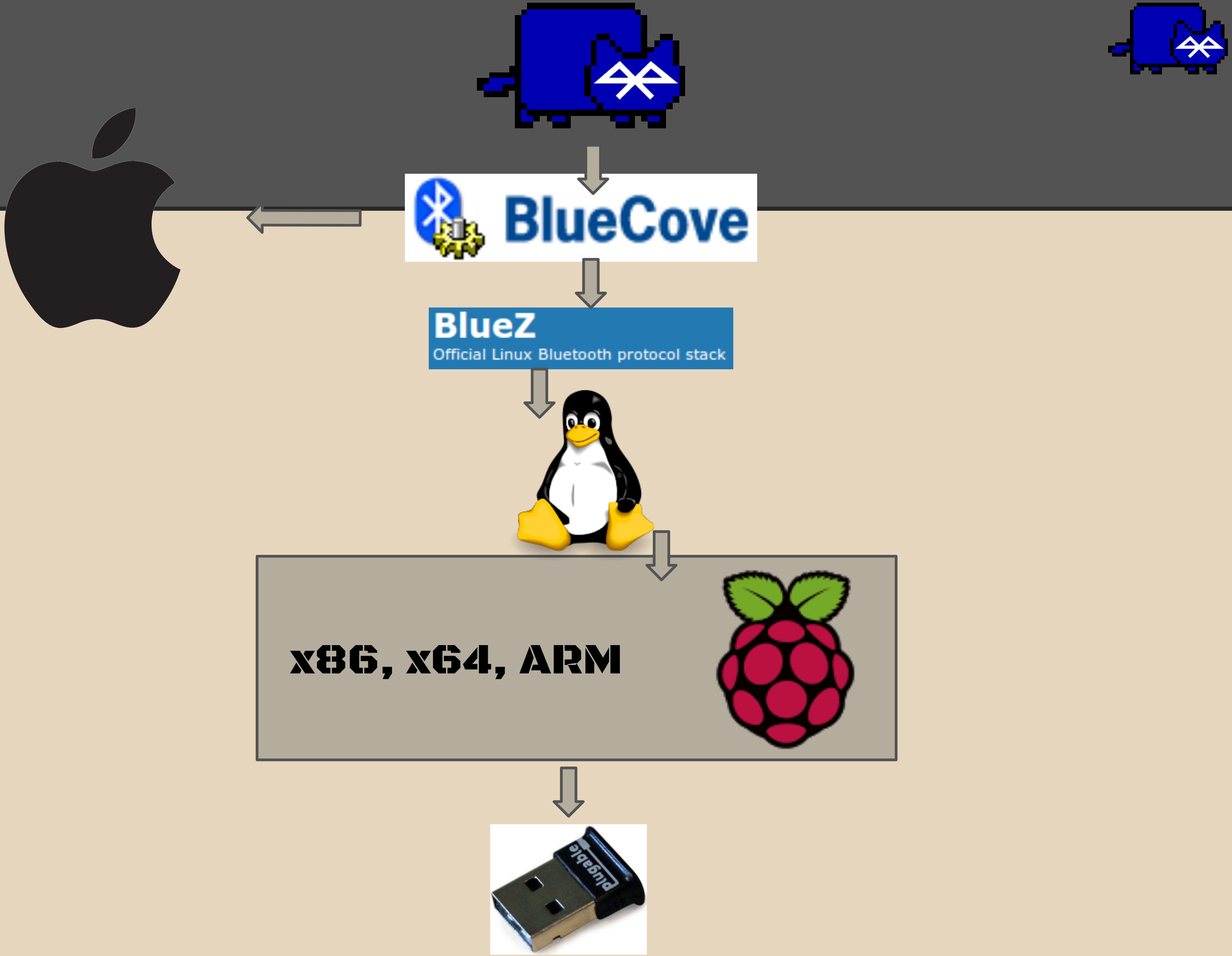
svn co http://svn.code.sf.net/p/blucat/code/trunk/blucat/



```
▼ blucat 89
    ▶ BlucatClient.java 83
    ▶ BlucatConnection.java 89
    ▶ BlucatServer.java 83
    ▶ BlucatState.java 93
    ▶ BlucatStreams.java 89
    ▶ BluCatUtil.java 94
    ▶ ListServices.java 83
    ▶ Main.java 93
    ▶ PairUtil.java 93
    ▶ PrintUtil.java 76
    ▶ RemoteDeviceDiscovery.java 93
    ▶ ScanServices.java 83
▼ com.intel.bluetooth 82
    ▶ MicroeditionConnector.java 81
    ▶ PairUtil.java 93
▼ compression 66
    ▶ CompressedBlockInputStream.java 66
    ▶ CompressedBlockOutputStream.java 66
```



```
▼ lib 80
      bluecove.zip 5
      bluecove-2.1.0.jar 5
      bluecove-2.1.1-SNAPSHOT-r63.jar 80
      bluecove-2.1.1-SNAPSHOT-r63-sources.zip 80
      bluecove-2.1.1-SNAPSHOT-r63-sources-all.zip 80
      bluecove-2.1.1-SNAPSHOT-r64.jar 78
      bluecove-2.1.1-SNAPSHOT-r64-sources.jar 78
      bluecove-bluez-2.1.1-SNAPSHOT-r63.jar 80
      bluecove-bluez-2.1.1-SNAPSHOT-r63-sources.tar.gz 80
      bluecove-emu-2.1.1-SNAPSHOT-r63.jar 80
      bluecove-emu-2.1.1-SNAPSHOT-r63-sources.tar.gz 80
      bluecovegpl.zip 5
      bluecove-gpl-2.1.0.jar 5
      bluecove-gpl-2.1.1-SNAPSHOT-r63.jar 80
      bluecove-gpl-2.1.1-SNAPSHOT-r63-sources.tar.gz 80
      commons-io-2.4.jar 70
      commons-io-2.4-sources.jar 70
      IOBluetooth 20
▼ lib.arm 86
      bluecove-gpl-2.1.1-SNAPSHOT-r63.jar 86
```



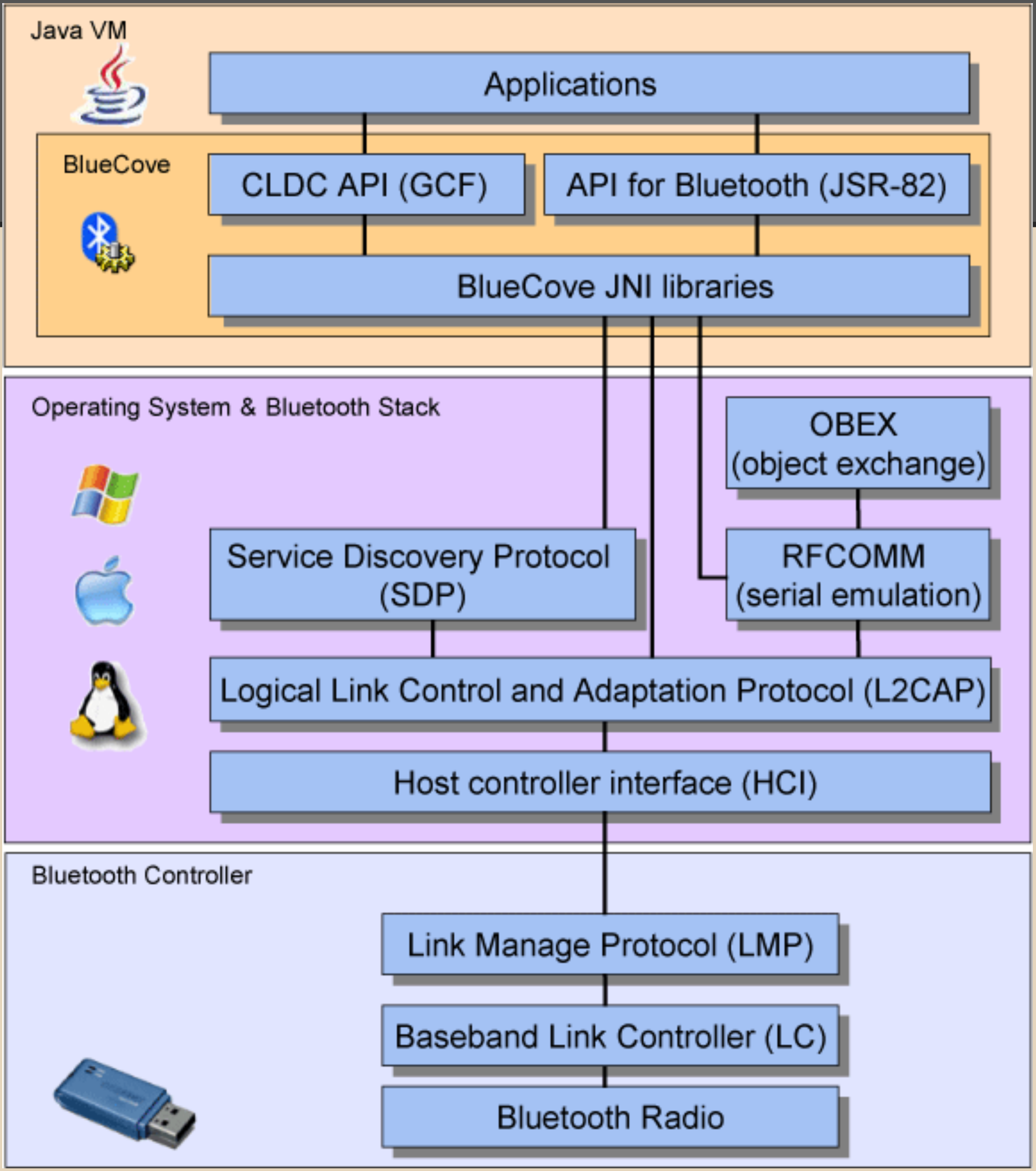
BlueCove

BlueZ

Official Linux Bluetooth protocol stack

x86, x64, ARM

BlueCove



Java VM

Applications

BlueCove

CLDC API (GCF)

API for Bluetooth (JSR-82)

BlueCove JNI libraries

Operating System & Bluetooth Stack

OBEX (object exchange)

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)

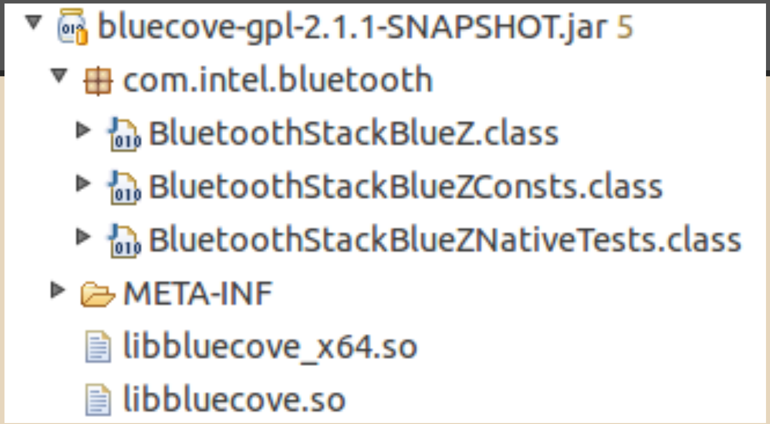
Bluetooth Radio

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){...}
```

Thanks!

http://josephpcohen.com
http://blucat.sf.net