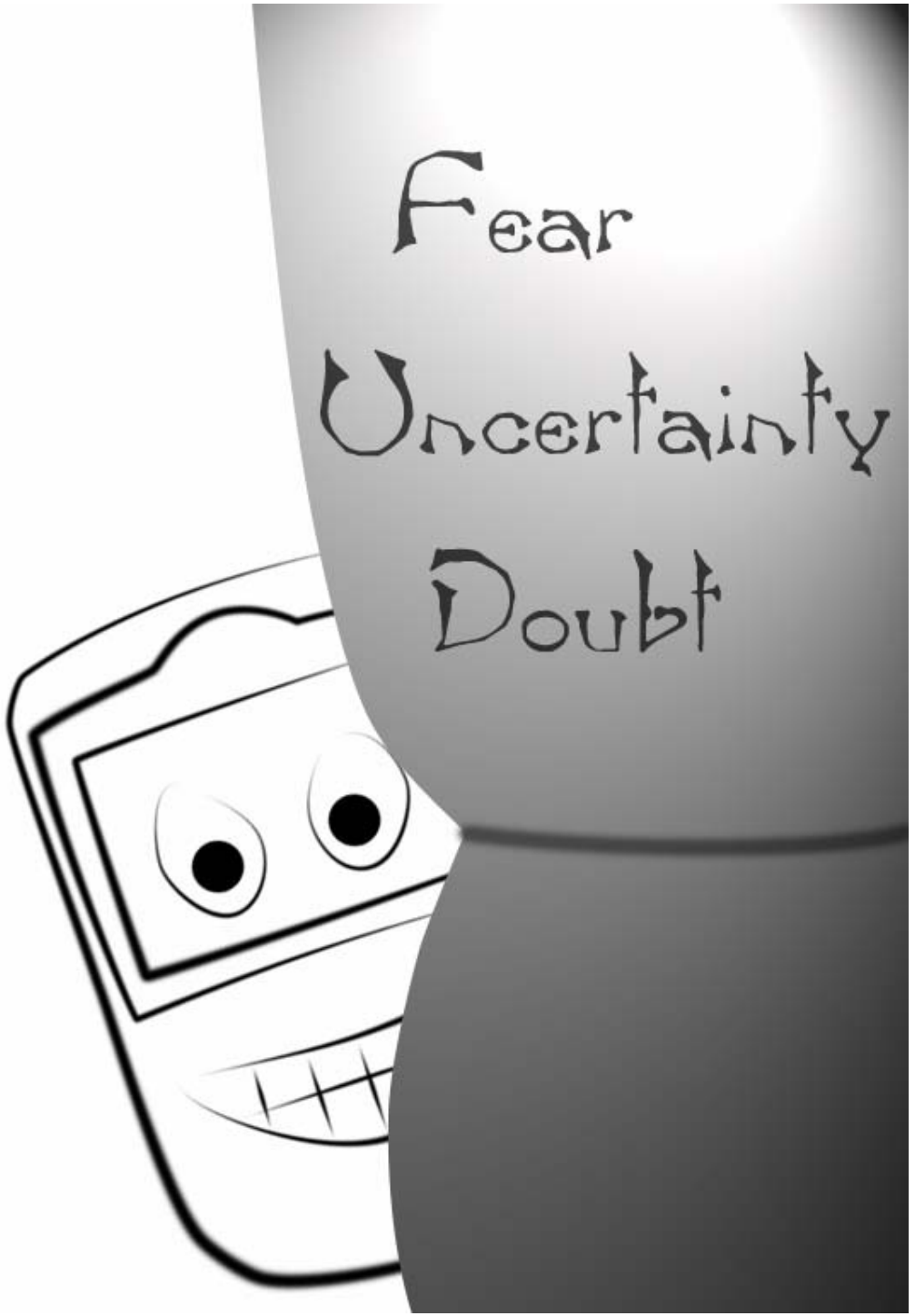
BlackBerry: Call to Arms, some provided

[Schwatzbärchen: Aufrufen von Waffen, manche vorhanden]

FtR & FX of Phenoelit



Preamble

This is the result of more than 500 hours of hard work as well as the support and trust from a few key people, who value this type of work and understand that it is neither quick nor easy.

This talk is dedicated to these people.

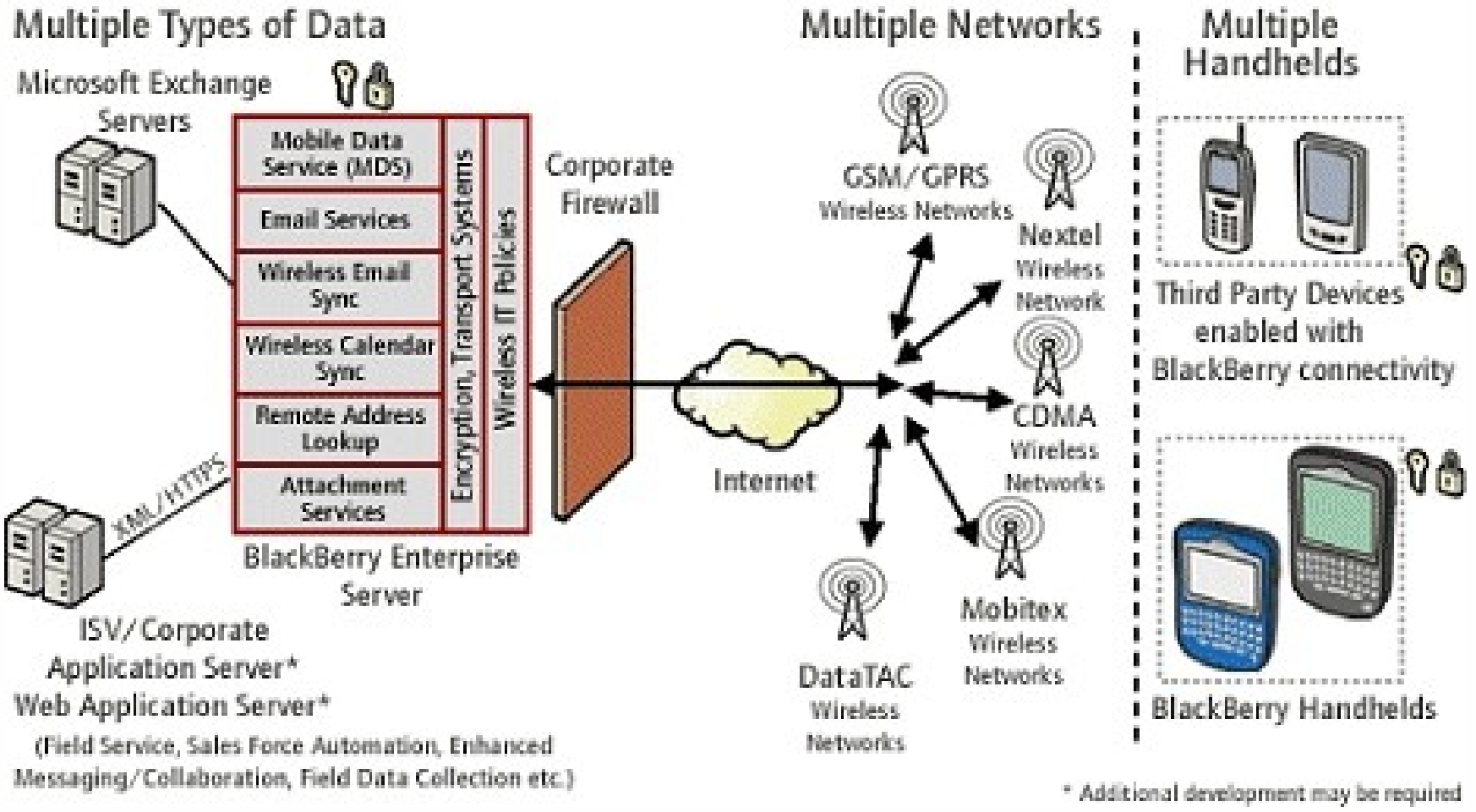
Purpose

- The purpose of this talk is to provide a starting point for interested hackers
- Nobody claims completeness, correctness or anything else ending on –ness
- We will try to balance freedom of information and risks to RIM customers

Summary:

It’s up to you where you take it from here.

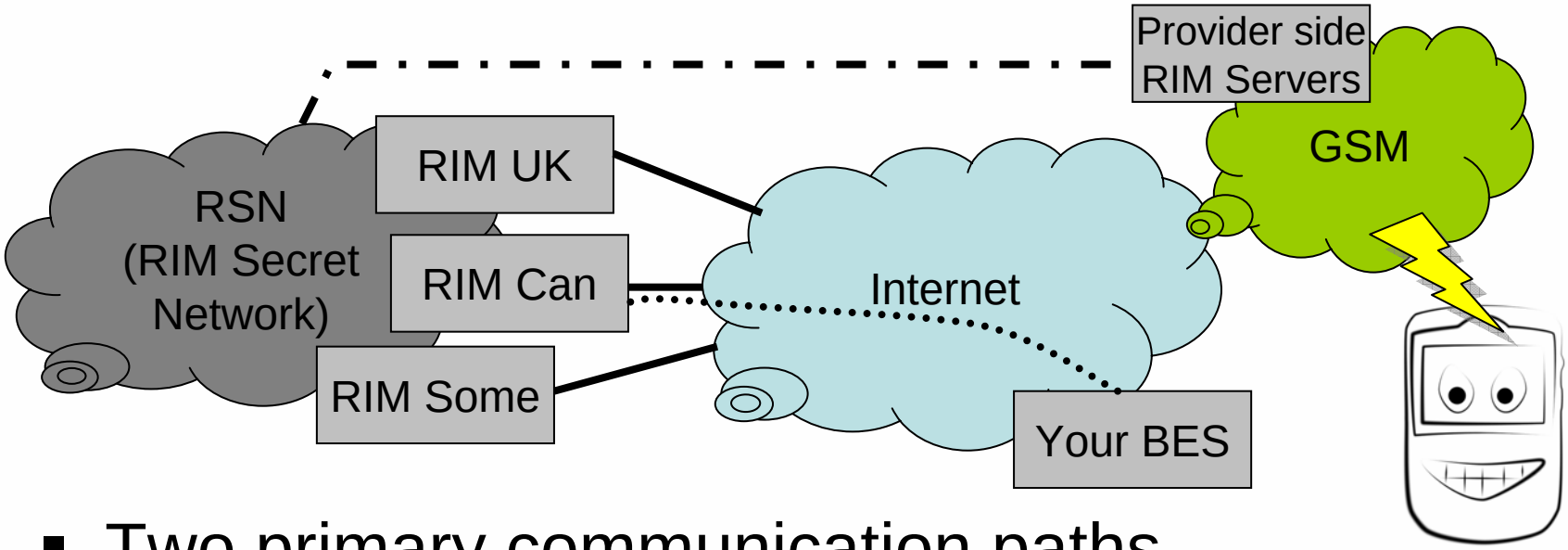
Overview – RIM's version



The Network Protocols

- Network architecture
- SRP
- GME
- Application data
- OTA

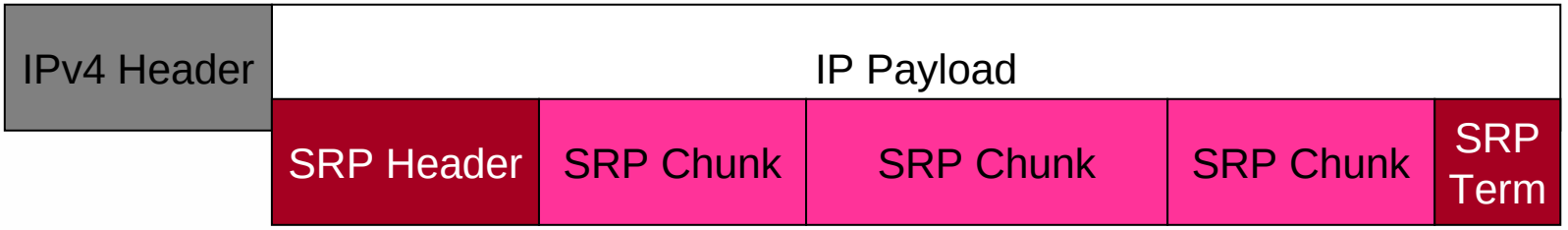
Network Architecture



- Two primary communication paths
  - Wired network (corporate network to RIM)
  - Over the Air network (RIM, GSM provider, Handheld)

Server Relay Protocol

- Encapsulation protocol inside IPv4
  - Simple header
  - Multiple string or integer payload chunks in TLV (type, length, value) format



Server Relay Protocol

Header

Byte	Meaning
1	Protocol Version
2	Function
3-6	Length of the entire message

Chunk Format

Data type	Byte	Value/Meaning
String	1	0x53 / type identifier
	2-5	/ length of the string
	6-x	/ content
Integer	1	0x49 / type identifier
	2-5	/ value

SRP Opcodes

- 01 - RETURN
- 02 - DISCONNECT
- 03 - RECEIVE
- 04 - STATUS
- 05 - SEND
- 06 - CONNECT
- 07 - REGISTER
- 08 - DATA
- 09 - PAUSE
- 0A - RESEND
- 13 - CANCEL
- 14 - STATUS_ACK
- 15 - SUBMITTED
- 18 - DATA_ACK
- 19 - RESUME
- 21 - STATE
- F0 - RESET
- F1 - INFO
- F2 - CONFIG
- FC - PING
- FD - PONG
- FE - SRP Error

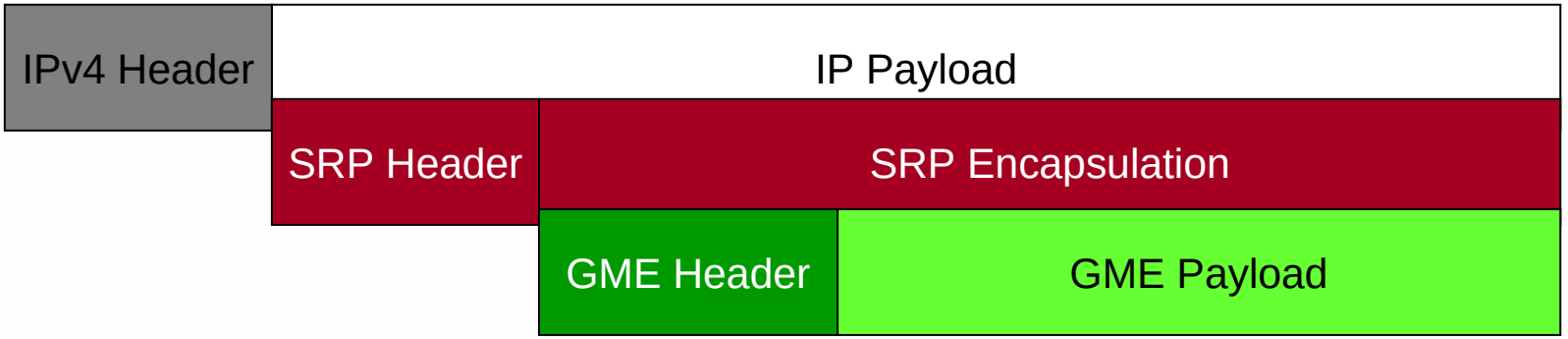
Session Setup

1. Client → Server: System ID
2. Server → Client: Server challenge
   - Server Random seed + Random value + Ctime
3. Client → Server: Client challenge
   - Client Random seed + Random value + Service string
4. Server → Client: HMAC_SHA1 (Client challenge)
   - Transformed SRP Key used for HMAC_SHA1
5. Client → Server: HMAC_SHA1 (Server challenge)
6. Server → Client: init request
7. Client → Server: init data

Successfully implemented a Server and a Client in Perl

Gateway Message Envelope

- Encapsulation protocol for messaging
- Routing Information of the message
  - Source (Server Identifier or PIN)
  - Destination (Server Identifier or PIN)
  - Message ID
- Comparable to information in Email headers



Generic Message Encapsulation(?)

GME Format

Field	Format
Protocol version	1 byte
Source	Type = 1 byte [0x10] Length = 1 byte Value
Destination	Type = 1 byte [0x20] Length = 1 byte Value
Terminator	1byte = [0x00]
Message ID	4 byte
Application Identifier	Type = 1 byte [0x50] Length = 1 byte Value
GME command	1 byte
Content length	Variable length integer
Terminator	1byte = [0x00]

Application Layer

- Application layer identifier in clear text
  - CMIME = message
  - CICAL = calendar updates
  - ITADMIN = key updates, IT policies, etc.
- Email, calendar and others encrypted
- PIN messages in clear text
  - Documented behavior, but very hard to find

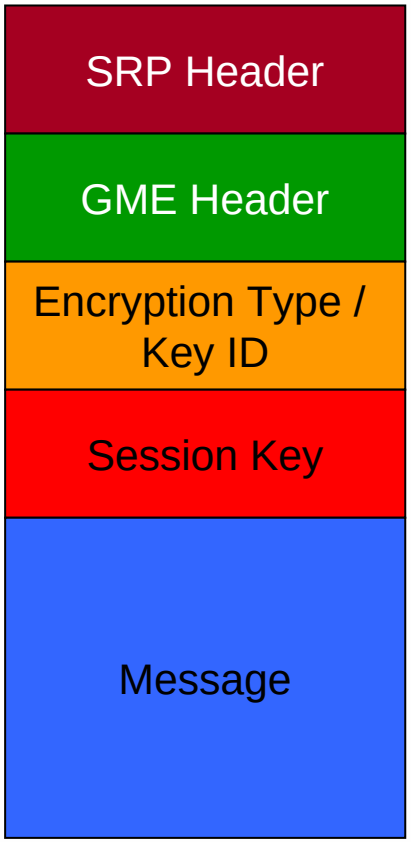
Application Layer

CMIME Format

Field	Format
Encryption Type	1 byte
Key ID	
Terminator	1 byte [0x00]
Session Key	32 Byte
Terminator	1 byte [0x00]
Message identifier	1 byte [0x19]
Message	

Application Layer Payload

- AES or DES encryption
- Key ID in clear text
- Session Key encrypted with device key
- Message compressed and encrypted with session key
- Successfully implemented packet dump message decryption script with given key in Perl



A word about the crypto

- Crypto library is FIPS certified
- Phe-no-crypto-people
- Implementation looks good in the disassembly
- No obvious key leak problems when activating devices via USB
- Crypto may be re-Weis-ed (as in Rüdi)

Decoding Dumps

```
0000000: 0208 0000 0083 4900 0002 f953  ......I....S
000000c: 0000 006f 2010 0954 3636 3632  ...o ..T6662
0000018: 3334 3236 2008 3233 3233 3233  3426 .232323
0000024: 3233 0000 000c 3850 0543 4d49  23....8P.CMI
0000030: 4d45 0340 4a00 0230 2b47 2b62  ME.@J..0+G+b
000003c: 001f 5131 9943 34ba e60e f8e4  ..Q1.C4.....
0000048: 1b9e 94e5 62c7 38ac 91dc c88a  ....b.8.....
0000054: ba93 6edf 1e32 6732 b800 19e7  ..n..2g2....
0000060: 1d40 d58b 0fbc eca3 0395 168c  .@..........
000006c: ddb8 b66e 501a 1f08 9d5e 93b7  ...nP....^..
0000078: 3d07 475c 4115 6149 0000 0000  =.G\A.aI....
0000084: 4900 0000 0300 00              I.....
```

SRP | GME | Encrypt Hdr | Key | Message

Traffic analysis

- Traffic analysis based on header possible
  - Sender PIN known
  - Recipient PIN known
  - Message content type known
  - Timing known
- In combination with (il)legal interception of SMTP email traffic
  - Email address to PIN mapping

Handing out arms [1]

- SRP Session setup with someone else's key and SRP ID
  - Legitimate key owner disconnected when modifying data in the session startup
  - New connection from either source results in the other one begin dropped

  ➔After 5 reconnects in less than a minute, the key is locked out. No BlackBerry service until RIM resolves the issue.
- RIM Authentication keys are not viewed as secrets by most companies
  - Slides and screenshots with keys can be found by your favorite search engine

Handing out arms [2]

- SRP String Type length field
  - Integer overflow leads to Access Violation when initially decoding packets
  - Negative value -5 causes infinite decoding loop
  - Affects at least router and enterprise server

```
.text:0042B11B          OR      eax, edx
   ; EAX is length field (now in Host Byte Order) after \x53
.text:0042B11D          LEA     edi, [eax+ecx]
   ; ECX is current position pointer in packet
.text:0042B120          CMP     edi, ebx
  ; position + length > overall_length ?
.text:0042B122          JG      short loc_42B19F
  ; jump to failure handling code if position + length points
  ; past the packet
```

Spam anyone?

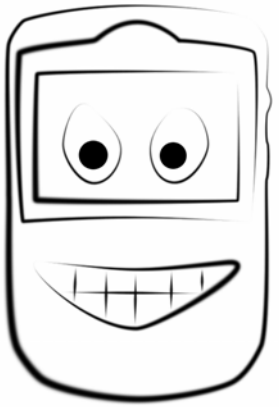
- PIN messages not encrypted
  - Therefore, no crypto code needed
- SRP authentication key can be used to PIN message anybody, not only your users
  - Any legitimate or stolen SRP key can be used
- Simple Perl script sufficient to send messages to any PIN
  - Sequentially sending it to all PINs from 00000000 to 99999999 ?
  - Spoofing sender might be possible (no evidence that it is not)

Over the Air protocol

- Fairly complex protocol for OTA activation and delivery of content
- Not our home turf (sorry :/)
- Should contain
  - GME transport
  - Wireless network independent protocol layer
  - A bunch of other interesting things
- Left as an exercise to the audience (see presentation title)

The Devices

- History
- Hardware
- OS Architecture
- Java Virtual Machine
- Policies
- Code Signing
- Curing the coffee addiction
- In memory of Siemens Phones
- Things to try at home



History

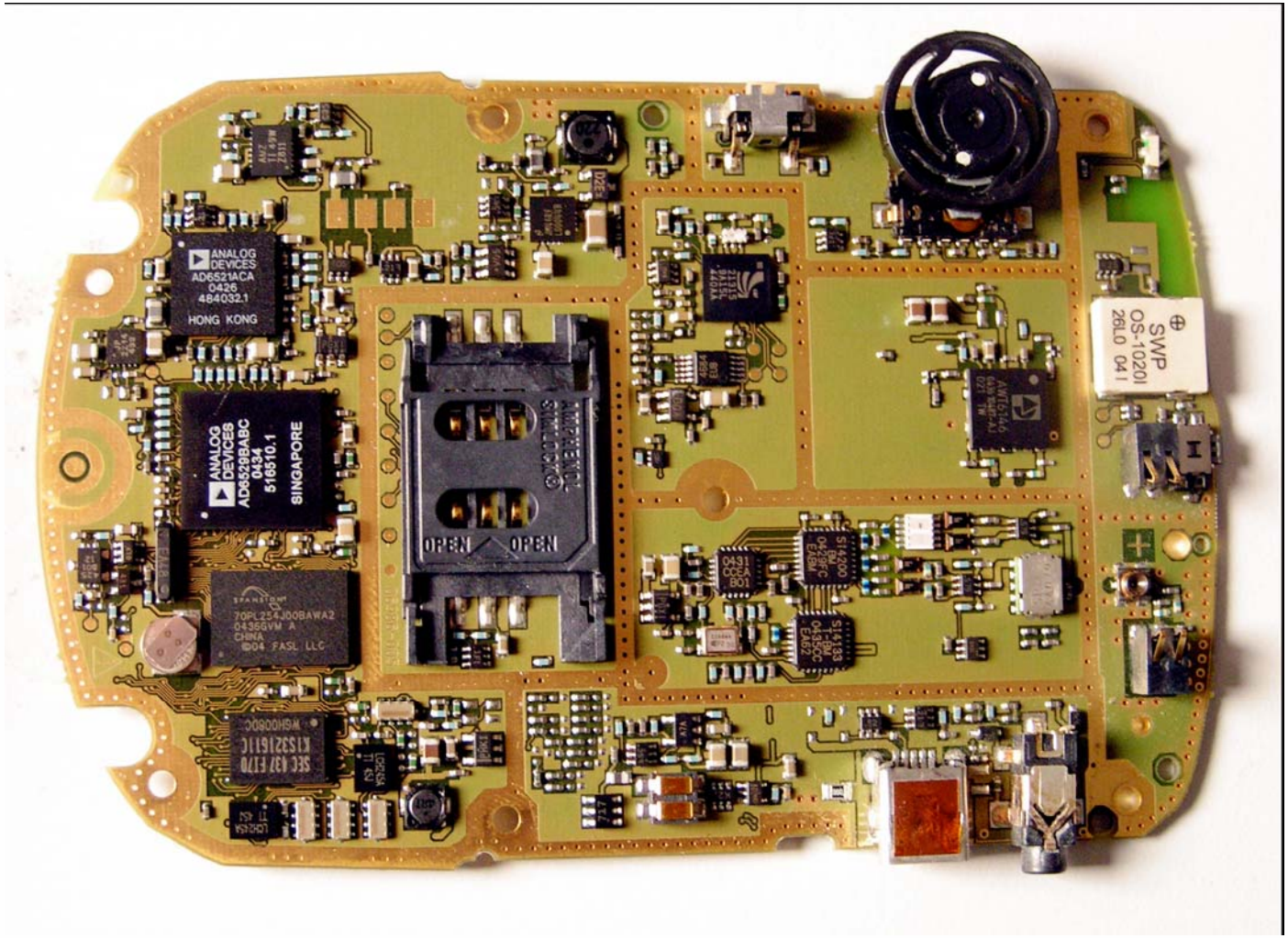
- First introduced as pager service for Mobitex and DataTAC
- Devices 8xx and 9xx were running on Intel© 386 CPU in a flat memory model without any (known) protections
- Real-Time OS with support for third party binary modules

The Hardware Today

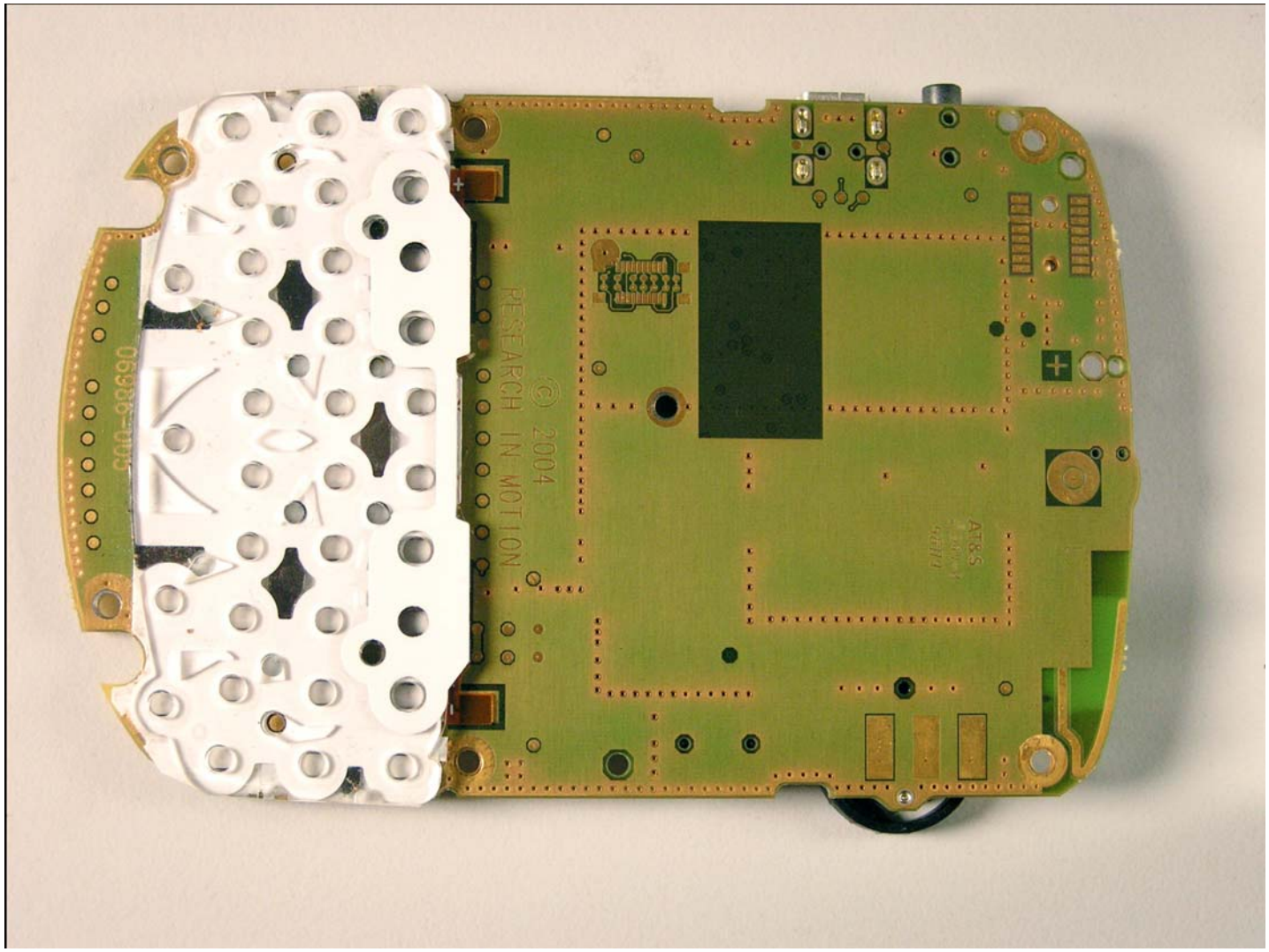
- ARM 7 TDMI main CPU
  - Partially Thumb code
- Build in DSP
- Separate Codec chip
- 64 MB Flash
- 32 MB RAM
- 4MB SRAM

Thanks to Frank for stripping the baby.

7290 naked


ANALOG DEVICES
AD6521ACA
0426
484032.1
HONG KONG
ANALOG DEVICES
AD6529BABC
0434
516510.1
SINGAPORE
70PL254J00BAWA2
04366VM A
CHINA
©04 FASL LLC
OPEN
OPEN
SWP
OS-1020I
26L0 041

7290 naked



OS Architecture

- Real-Time OS
  - 7290 with KADAK AMX 4 Kernel, others supposedly run RIM proprietary kernel/OS
  - GPRS, CDMA, Nextel, Mobitex and DataTAC code residing in binary code space
  - Distributed as large binary image
- Binary Modules
  - Still supported and used
  - PE/COFF Dynamic Link Libraries
  - Linked against RIM binary modules (Imports)

JVM

- Java Virtual Machine loaded as largest binary module (jvm.dll)
  - CDLC 1.1, MIDP 2.0
  - Java Vendor is RIM
- Limited set of J2ME classes
  - Reflection API missing ☹
- Device control via RIM classes
  - Java applications are almost useless without RIM class support

IT Policies

- Actually, a pretty good feature
- Centrally managed policies for handhelds
  - Prevent PIN Messages
  - Prevent third party application download
  - Require encrypted storage
  - Etc.
- Very important to implement when rolling out BlackBerry devices

Code Signing

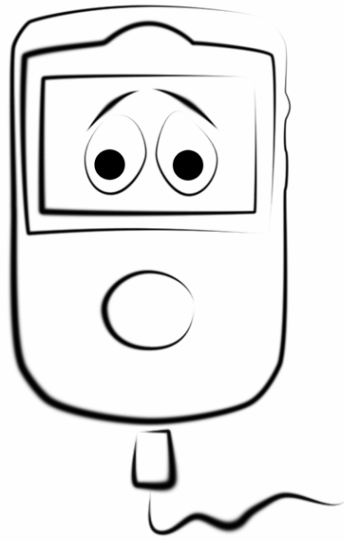
- Java Application signature
  - To use RIM classes
  - Signs a
  - $100 to
  - Suspicio
  - News Fla
  - Replacing
- Firmware i
  - Checked in Loader (see your debugger ☺)
  - Something is checked while device is loading ☹

```
Shift+F12
Alt+T
signature ENTER
ENTER
X
ENTER
```

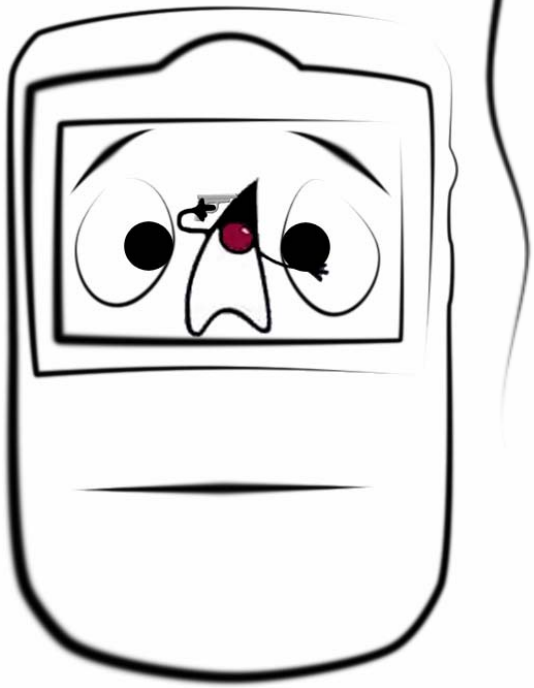
Curing the Coffee addiction

- The shipped application loader still supports a lot more features, one being /nojvm
- Loading of non-Java OS images works

  … and looks like a 90's Nokia phone
- Building a DLL *should* work like this:
  - Creating DLLs with stub functions for the used RIM C-API calls (documented)
  - Making a ARM DLL using MSEVC
  - Creating a .ali description file
  - Load the DLL onto the Device



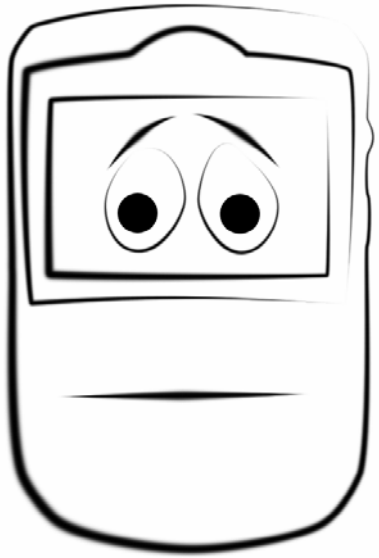
It’s not a Siemens, but …



- Browser Issue when parsing .jad Files:
  long name for
  MIDlet-Name or -Vendor
  - Exception thrown by the dialog
  - Uncaught, modal dialog left over
  - Browser toast, everything else still works
  - Soft- or Hard-Reset don’t work (solution: denial all power to the device)
- RIM says it’s fixed in 4.0.2

Things to try at home

- Find the JTAG connectors
- Bluetooth on BlackBerry
- JVM bugs
- Reversing Images
- Figuring out checksums
- Loader.exe should be able to read memory contents from the device as well (credit: mark@vulndev.org)



Call to Arms

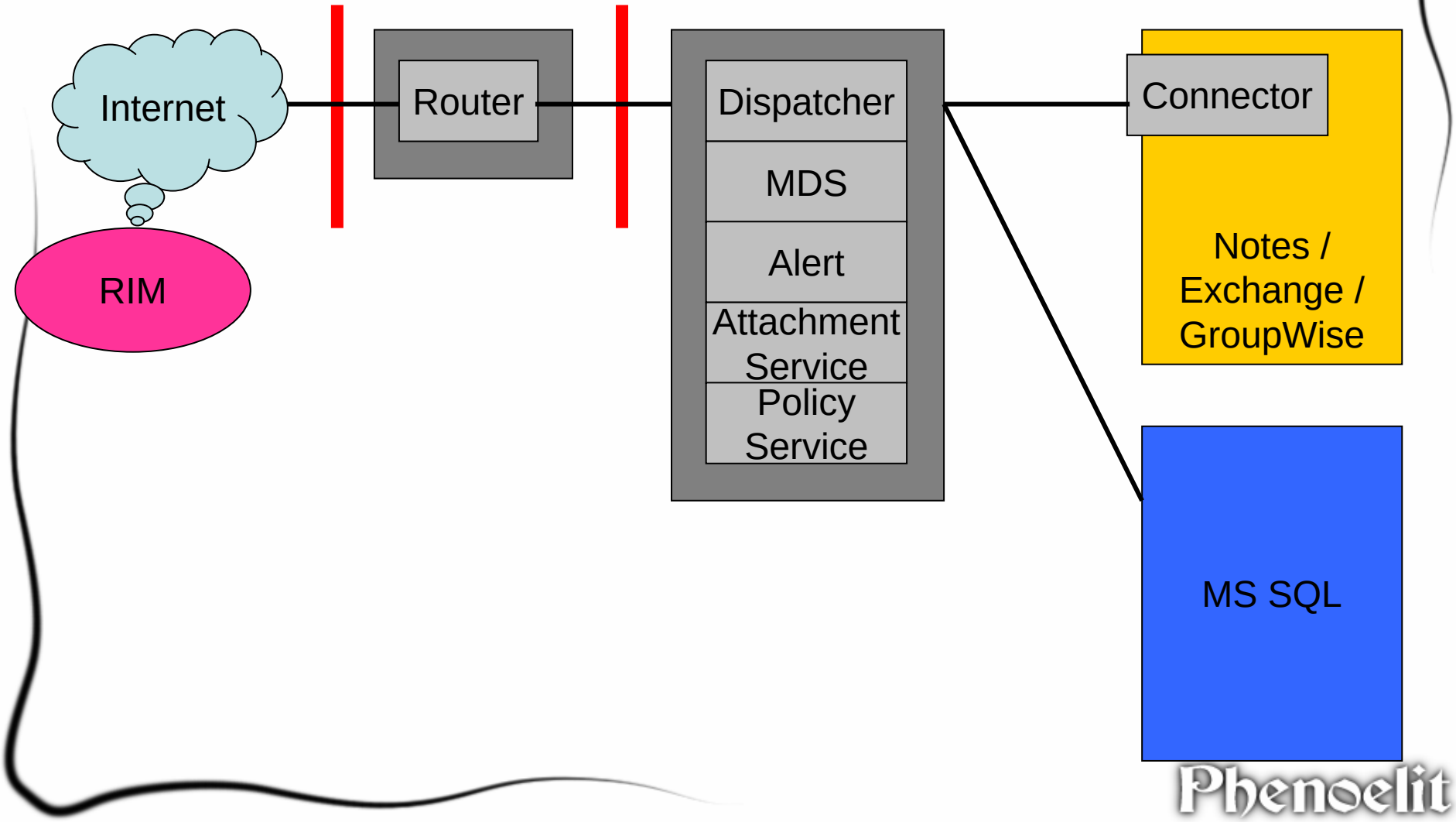
BlackBerry Enterprise Server

- BES Architecture
- SQL Database
- The beauty of updates
- Code style and quality
- Interesting libraries
- Attachment Service Special

Phenoelit

Call to Arms

BES Architecture



Phenoelit

Call to Arms

BES Accounts

	Logon Locally	Logon as Service	Local Admin	Exchange RO Admin	Exchange MailStore Admin
Service Account	✓	✓	✓	✓	✓
Server Mgmt Account	✓	✓	✓	✓	✓
User Admin Account		✓	✓	✓	

Phenoelit

SQL Database

- MS SQL Server with user authentication

```
SRP Authentication Key format:
  abcd-affe-fefe-ffff-cafe-babe-0bad-f00d-dead-beef
  ➔ abcdaffefefeffffcafebabe0badf00ddeadbeef

Algorithm:
  i := 0 ;
  while ( i < 21 )
  begin
    b := inbuf [ i ] ;
    i := i + 1 ;
    outbuf := map[ outbuf | ( b & 0x0F )] ;
    outbuf << 4 ;
  end
```

The beauty of updates

- RIM updates the BES
  - Service Packs
  - HotFixes
  - Release and fix notes tend to be extremely entertaining
- Hackers should update BES
  - SABRE BinDiff
  - Free .pdb debug information files in some fixes. Many thanks to RIM.

Code style & quality

- Massive C++ code
  - By-the-book pattern implementations
  - Large classes
  - STL
  - Harder to reverse engineer
- Surprisingly good
  - STL helps a lot
  - “If in doubt, check again” approach
    - A.k.a. select, select, select, recv
  - But generally using signed integers, although mostly correct

Interesting Libraries - commercial

- Microsoft IStream classes
  - Parsing of Microsoft Office documents
- Microsoft MSHTML4 engine
  - Parsing of HTML documents
- MSXML SDK
  - Installed, no idea what for.
  - MSXML used for Sync server.
- Arizan parsing product
  - Central parsing engine
  - Parsing of PDF and Corel WordPerfect

Interesting Libraries – open source

- Zlib 1.2.1
  - ZIP attachment handling is copy & paste contrib/unzip.c (almost binary equal)
  - Known bugs ☺
    1.2.3 is current
- GraphicsMagick 1.1.3
  - ImageMagick spin-off
  - Fully linked, including debug code and

Interesting Libraries – open source

- Supported and compiled in file formats in GraphicsMagick:
  - ART, AVI, AVS, BMP, CGM, CMYK, CUR, CUT, DCM, DCX, DIB, DPX, EMF, EPDF, EPI, EPS, EPS2, EPS3, EPSF, EPSI, EPT, FAX, FIG, FITS, FPX, GIF, GPLT, GRAY, HPGL, HTML, ICO, JBIG, JNG, JP2, JPC, JPEG, MAN, MAT, MIFF, MONO, MNG, MPEG, M2V, MPC, MSL, MTV, MVG, OTB, P7, PALM, PBM, PCD, PCDS, PCL, PCX, PDB, PDF, PFA, PFB, PGM, PICON, PICT, PIX, PNG, PNM, PPM, PS, PS2, PS3, PSD, PTIF, PWP, RAD, RGB, RGBA, RLA, RLE, SCT, SFW, SGI, SHTML, SUN, SVG, TGA, TIFF, TIM, TTF, TXT, UIL, UYVY, VICAR, VIFF, WBMP, WMF, WPG, XBM, XCF, XPM, XWD, YUV

Interesting Libraries – open source

- GraphicsMagick ChangeLog:
  - “coders/avi.c, bmp.c, and dib.c: applied security patch from Cristy.”
  - “coders/tiff.c (TIFFErrors): Prevent possible stack overflow on error.”
  - “coders/psd.c (ReadPSDImage): Fix stack overflow vulnerability”
  - “coders/tiff.c (ReadTIFFImage): Fix overflow while computing colormap size.”
- Odd own format strings in arbitrary text fields of any image format
  - Expect image comment **`100%tonne`**
    to become 100C:\Windows\temp\bbaAA.tmponne

Open Source Arms [1]

- Heap overflow in TIFF parser
  - Integer overflow in image data memory requirement allocation
  - Allocation of small (0) memory block for image data

Open Source Arms [2]

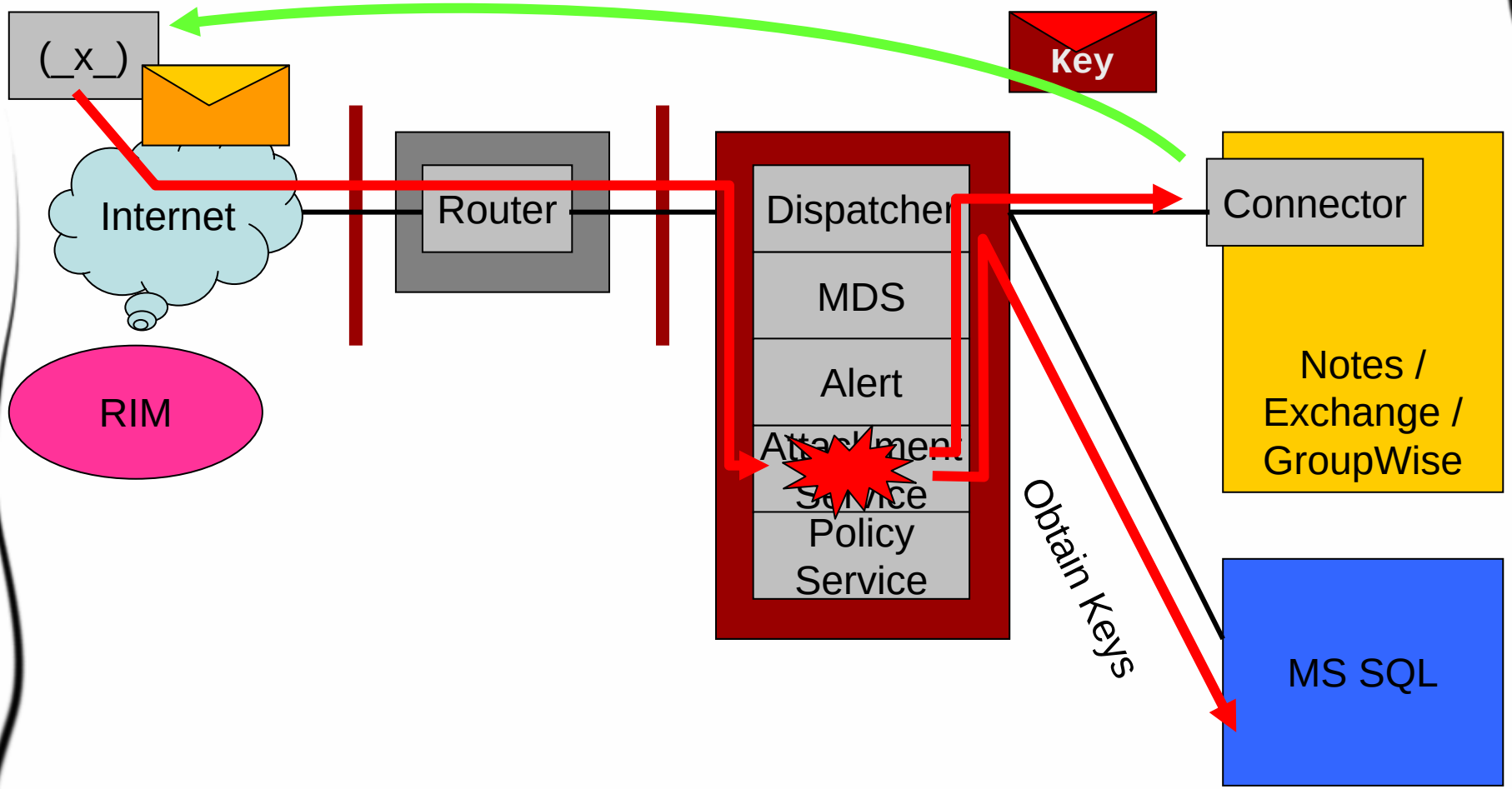
- Heap overflow in PNG parser
  - #define PNG_USER_WIDTH_MAX 1000000L does not prevent integer overflows
  - Overflow in memory allocation counter
  - Allocation of small (1MB) memory block for image data decompression

Open Source Arms [3]

- Zlib museum in PNG parser
  - Paying attention?
    Version 1.2.1 used, inclusive decompression bug
  - PNG image data is zip compressed
  - Heap overflow when decompressing image data
  - Your arbitrary BugTraq example works
- Interestingly enough, known libPNG bugs are fixed

Call to Arms

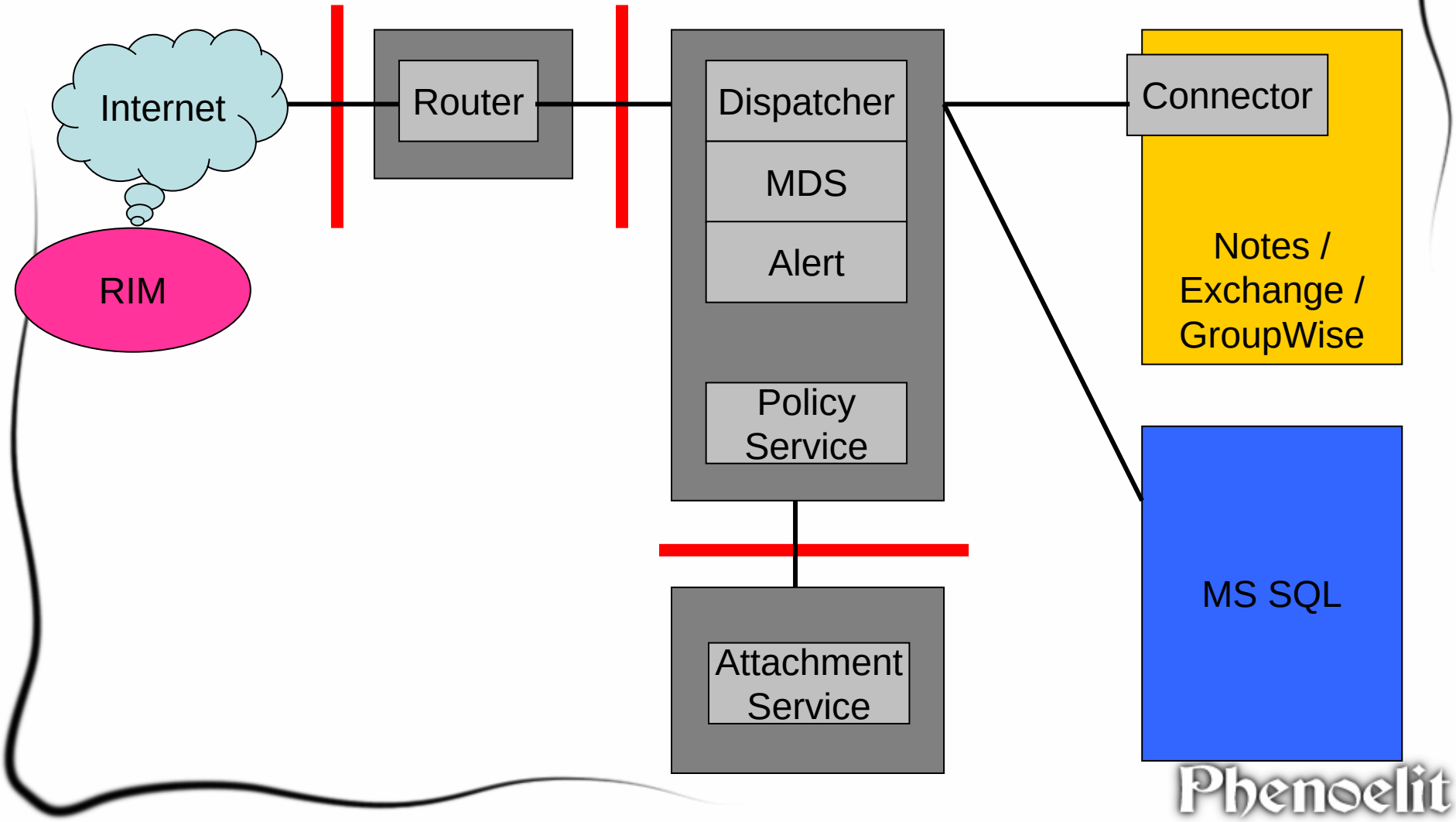
BES Architecture Attack



Phenoelit

Call to Arms

BES Architecture must be



Phenoelit

Separate Attachment Service issue

- Remote control
  - TCP port 1999
  - Unauthenticated XML
  - Query
    - Version
    - Statistics
    - Number of processes
  - Set number of processes
    - Recommended test values: 0, 20000

Administrativa: Fix Information

- Handheld browser JAD issue fixed in 4.0.2
- PNG issue(s) fixed in BES release 4.0.3
- Zlib updated
- TIFF issue fix pending
- SRP issue fix pending

Thanks & Greets

Greets fly to:
Phenoelit (here or on vacation), 13354, Halvar Flake & SABRE Security, THC, all@ph-neutral, hack.lu, Scusi, mark@vulndev.org, Frank Rieger, the Eschschloraque Rümpschrümp, mac, t3c0, trash, the darklab@darklab.org people, Dan Kam-in-Sky, and Ian Robertson from RIM