Embedding NetBSD: VOIP Applications

Fabio Balzano
fabiodive@gmail.com

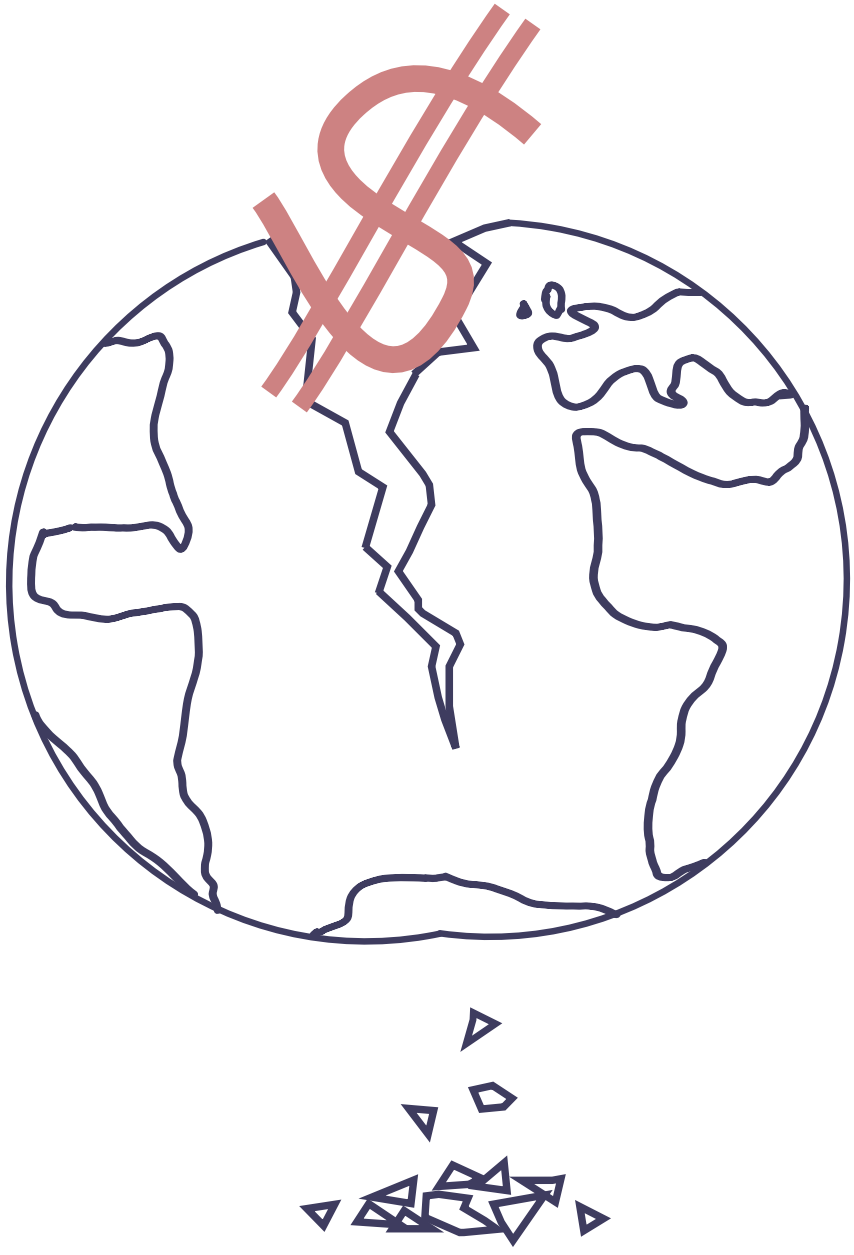
University of Applied Sciences,
Vienna, Austria
May 5, 2012

Talking Points:

- VOIP:..Cool!
- Alix Board: Platform introduction
- NetBSD: The workflow for embedding and configuring
- VOIP: Freeswitch configuration and usage examples

Follow the configuration on GitHUB:
`https://github.com/fabiodive/BSDDay-2012-Wien`

VOIP is cheap!



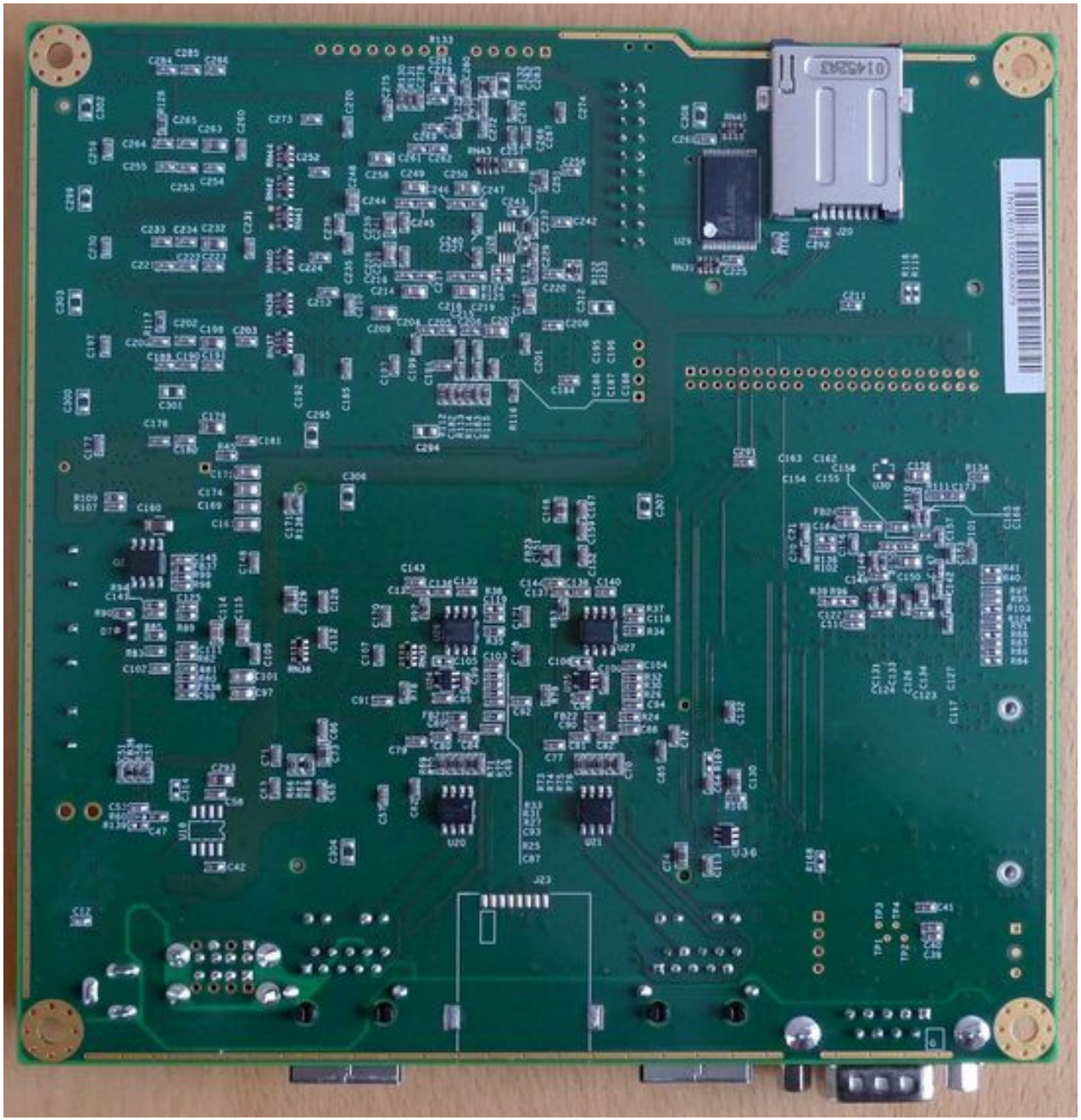
- No pay for distance
- No pay for call
- Build your own service

VOIP rocks!

- No GEO barriers
- Bring the services with you
- **INVENT YOUR COMMUNICATION!**

Alix6E2: Front

Alix6E2: B-Side



CENTRAL EUROPEAN BSD-DAY 2012 VIENNA, AUSTRIA

Alix6E2: Specifications

- AMD Geode LX800 CPU @ 500Mhz with 256MB DRAM
- One CF card slot
- One 44 pin IDE header to connect a 2.5" drive
- 2x 100Mbps Ethernet with separate Via VT6105M controller
- 2x USB 2.0 ports + 1 internal header
- One DB9 serial port, one internal RS232 header
- I2C and LPC bus headers
- One MiniPCI slot
- SIM card slot
- Optional RTC battery
- 3x software controllable green LEDs on the front panel
- One recessed push button switch
- Optional RTC battery
- 7 - 18V DC power input

Alix6E2: Killer Features

- **PEAK 6W POWER CONSUMPTION**
  without miniPCI cards and USB devices
- AMD Geode LX800 CPU
- Hardware Crypto Engine aes-128-cbc

NetBSD: Flash Memories

Pros:

- **LOW POWER**
- Low latency
- No mechanical failures
- Silence

Cons:

- **LIMITED LIFE DUE TO WRITE CYCLES**
  SLC NAND media: 100.000 program-erase cycles [P/E]
- No swap partition
- Actually more expensive than HDs
- Actually size smaller than Hds

CENTRAL EUROPEAN BSD-DAY 2012 VIENNA, AUSTRIA

NetBSD: The new CHFS

Introduced in November 2011

Born in the Department of Software Engineering,
University of Szeged, Hungary

- Log structured flash filesystem
- It can manage bad-blocks
- First specific Flash Filesystem for NetBSD

project page: `http://chewiefs.sed.hu`

Why NetBSD?

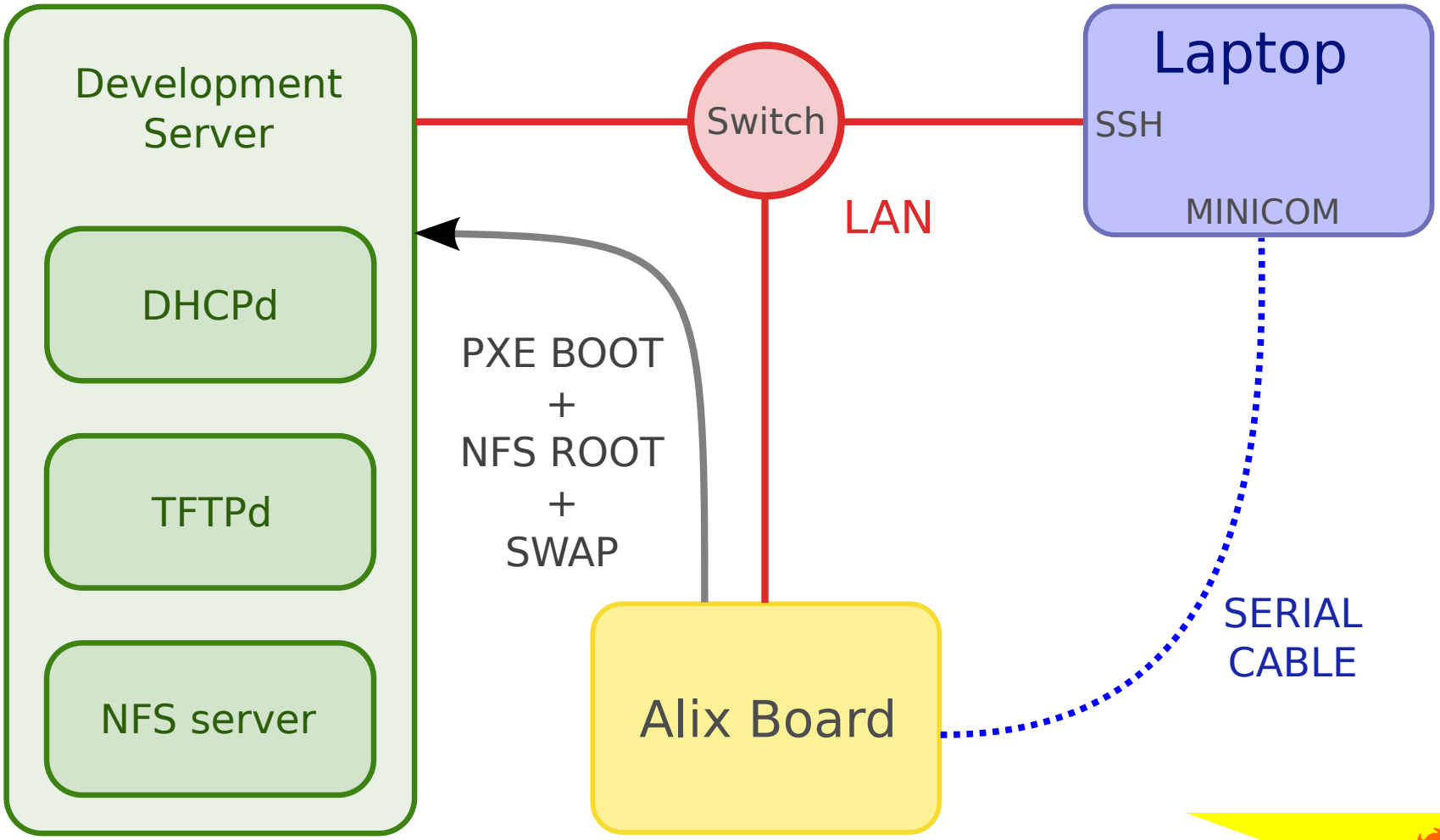
NetBSD: Why

- High Portable
- Compact
- Clean
- Security
- Kernel built like a rock
- Really Unixoid
- Tools for embedding

NetBSD: The Players

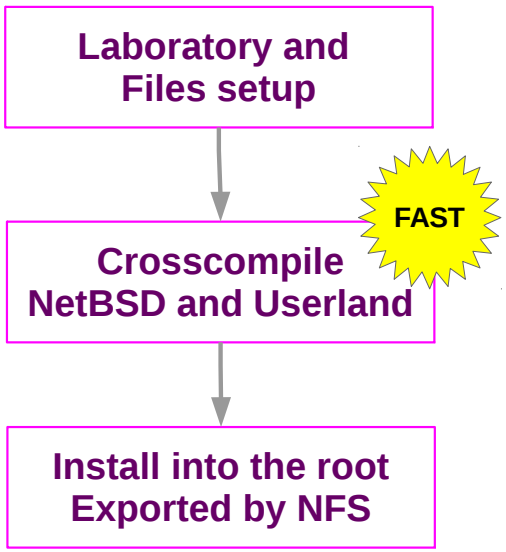
- Cross-compilation
  build.sh script
- DHCPd
- TFTPd
- PXE boot
- NFS

NetBSD: The BattleField

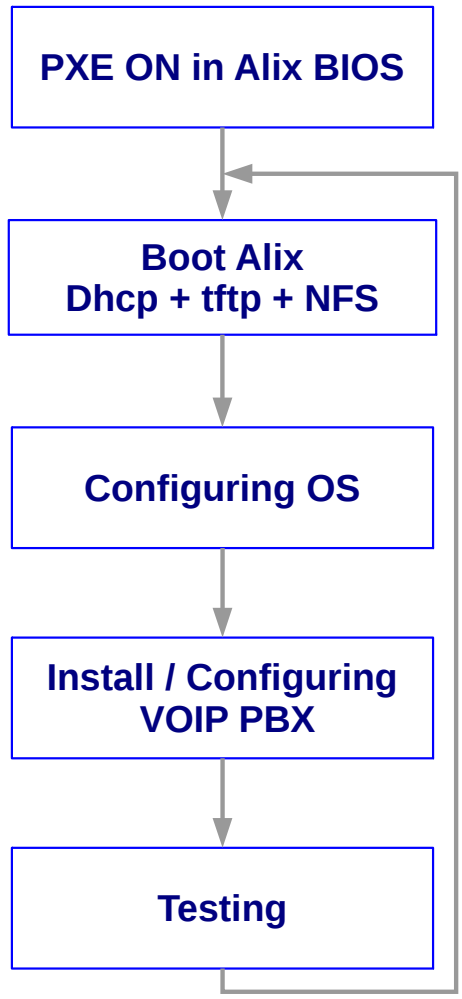


NetBSD: The Game

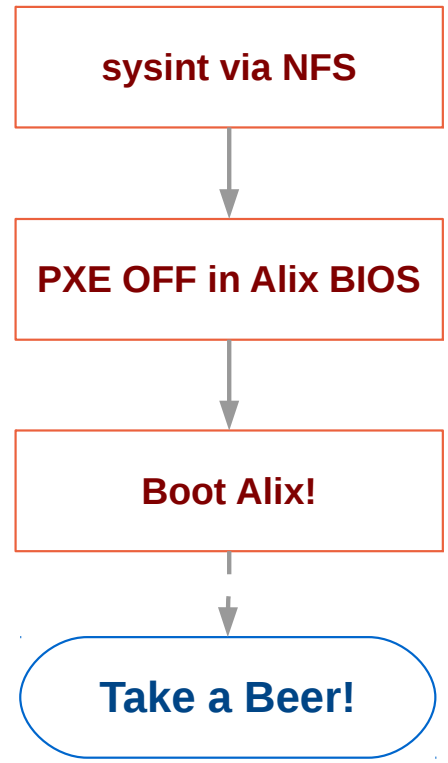
Level 1



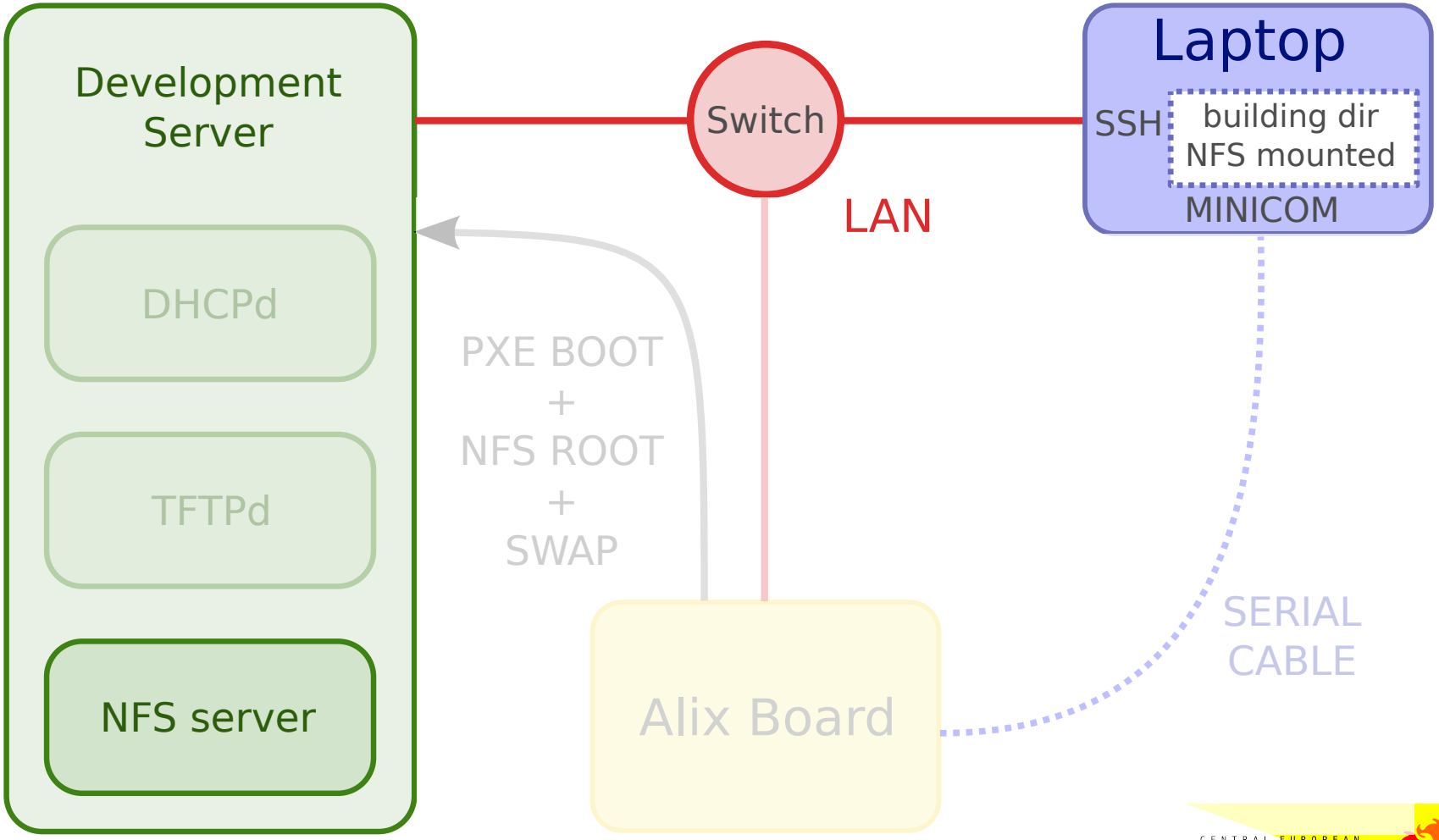
Level 2



Game Over



NetBSD: Crosscompiling



NetBSD: NO binaries?

Compiling benefits:

- Optimization
- Custom make options
- Load our drivers, enable features
- Smaller, faster kernel
- Security and Reliability

binaries
test boots!

Crosscompiling benefits:

- **FASTER!**
  example: a multiprocessor laptop
- The storage is a Flash (CompactFlash Card)
  ..limited number of write cycles
- Easy files management and versioning

NetBSD: configuring an embedded KERNEL

What to do:

- begin with GENERIC_TINY and GENERIC
- compare with 'dmesg' command output
- use 'modstat', 'pcictl' commands
- hardware specifications help!

Remember to enable:

- CS5536 chip GEODE drivers
- VIA Network drivers
- glxsb PCI cryptographic device
- disable all the modules you don't need
- **HZ=1000 KERNEL SPEED**

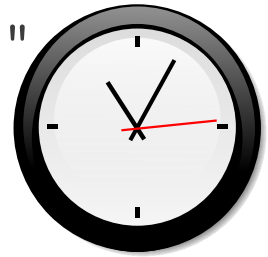
NetBSD: Preparing the files

CVS:

- Easy to keep updated the sources
- It fetches only the new files

To get the sources by CVS:

```
$ export CVS_RSH="ssh"
$ export CVSROOT="anoncvs@anoncvs.NetBSD.org:/cvsroot"
$ cd <project directory>/usr
$ cvs checkout -r netbsd-6 -P src
```



To update:

```
$ cvs update -Pd
```

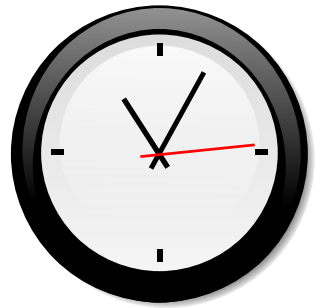
NetBSD: Crosscompiling

Hello build.sh!

- Crosscompiling the tools
- Crosscompiling the kernel
- Crosscompiling the userland

To compile the tool-chain:

```
$ cd ./usr/src
$ ./build.sh -j2 -m i386 -O ../../build/obj.i386 \
> -T ../../build/tools.i386 tools
```



To update without clean:

```
$ ./build.sh -j2 -m i386 -O ../../build/obj.i386 \
> -u -T ../../build/tools.i386 tools
```

NetBSD: Crosscompiling Userland and Kernel

To crosscompile the kernel:

```
$ cd ./usr/src
$ ./build.sh -j2 -m i386 -T ../../build/tools.i386 \
> kernel=ALIXKERN
```

To crosscompile the kernel and userland:

```
$ ./build.sh -j2 -m i386 -M ../../build/obj.i386 \
> -T ../..//build/tools.i386 \
> -D ../../build/destdir.i386 -U \
> kernel=ALIXKERN release releasekernel=ALIXKERN
```

NetBSD guide:

www.netbsd.org/docs/guide/en/chap-build.html

NetBSD: What we built in 'source' directory:

```
— source
    ├— build
    |    ├— destdir.i386
    |    ├— obj.i386
    |    └— tools.i386
    └— usr
         └— src
```

Read the final output of build.sh

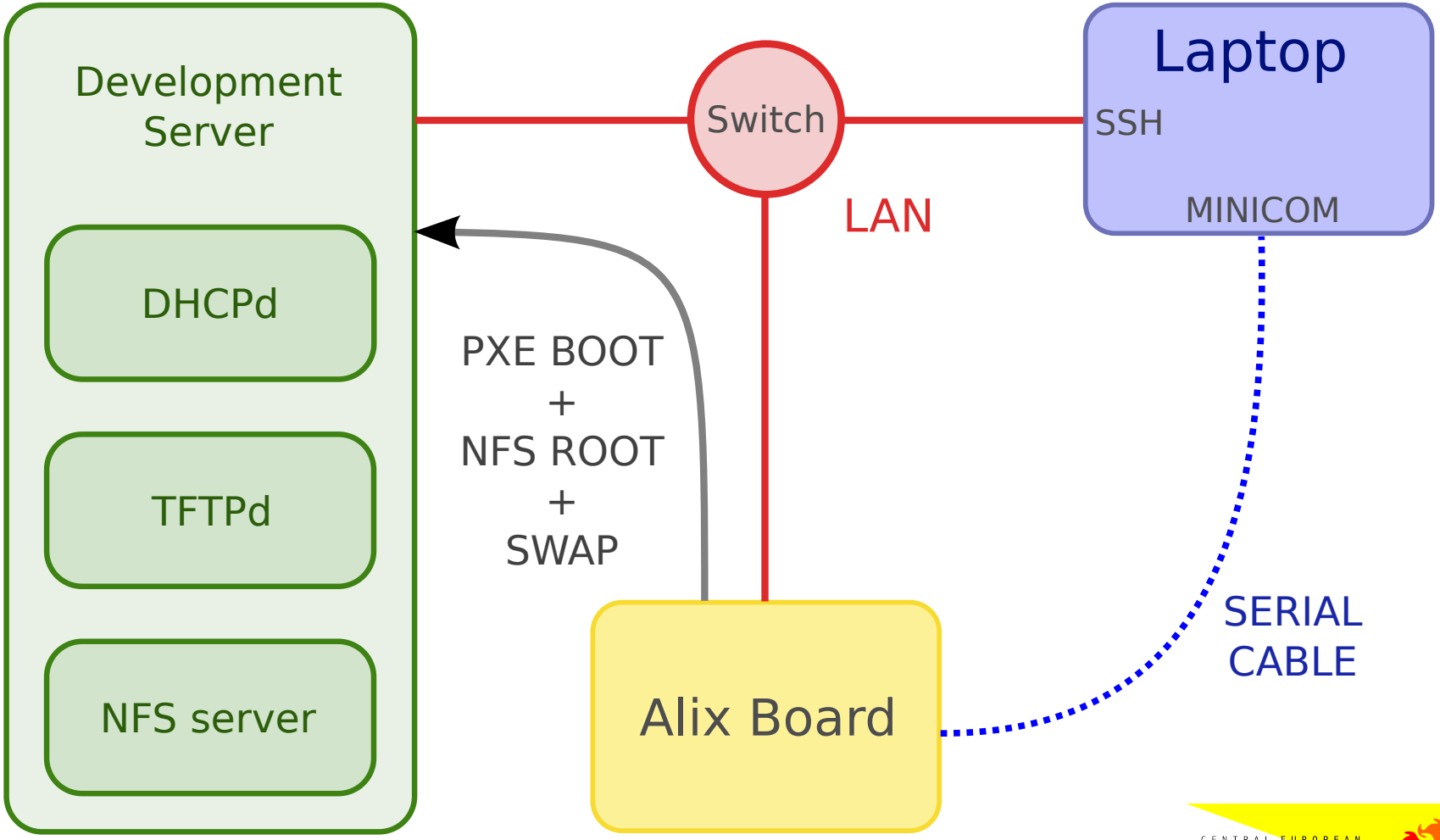
Copy the new relased kernel to the TFTP directory

New DHCPd server options:

- option root-path "DESTDIR"
- option filename "Kernel copy"

Edit /etc/exports for NFS

NetBSD: Booting the custom NetBSD via PXE and NFS



NetBSD: The 'pxeboot' file

1. Download pxeboot_ia32.bin from FTP

   ```
   ftp://ftp.de.netbsd.org/pub/NetBSD/NetBSD-6.0_BETA/
   i386/installation/misc/
   ```

2. Enable the serial console:

   ```
   $ installboot -e -o console=com0 -o speed=0 -m i386 \
   > ./pxeboot_ia32.bin
   ```

3. Move the file to the TFTP directory

4. Edit dhcpd.conf

Follow the configuration on GitHUB:
`https://github.com/fabiodive/BSDDay-2012-Wien`

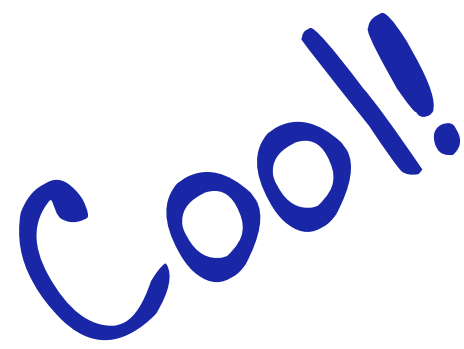
NetBSD: Enabling PXE in Alix BIOS

```
640 KB Base Memory
261120 KB Extended Memory

01F0 Master 044A CF 1GB
Phys C/H/S 1966/16/63 Log C/H/S 983/32/63

BIOS setup:

(9) 9600 baud (2) 19200 baud *3* 38400 baud (5) 57600 baud (1) 115200 baud
*C* CHS mode (L) LBA mode (W) HDD wait (V) HDD slave (U) UDMA enable
(M) MFGPT workaround
(P) late PCI init
*R* Serial console enable
*E* PXE boot enable
(X) Xmodem upload
(Q) Quit
```

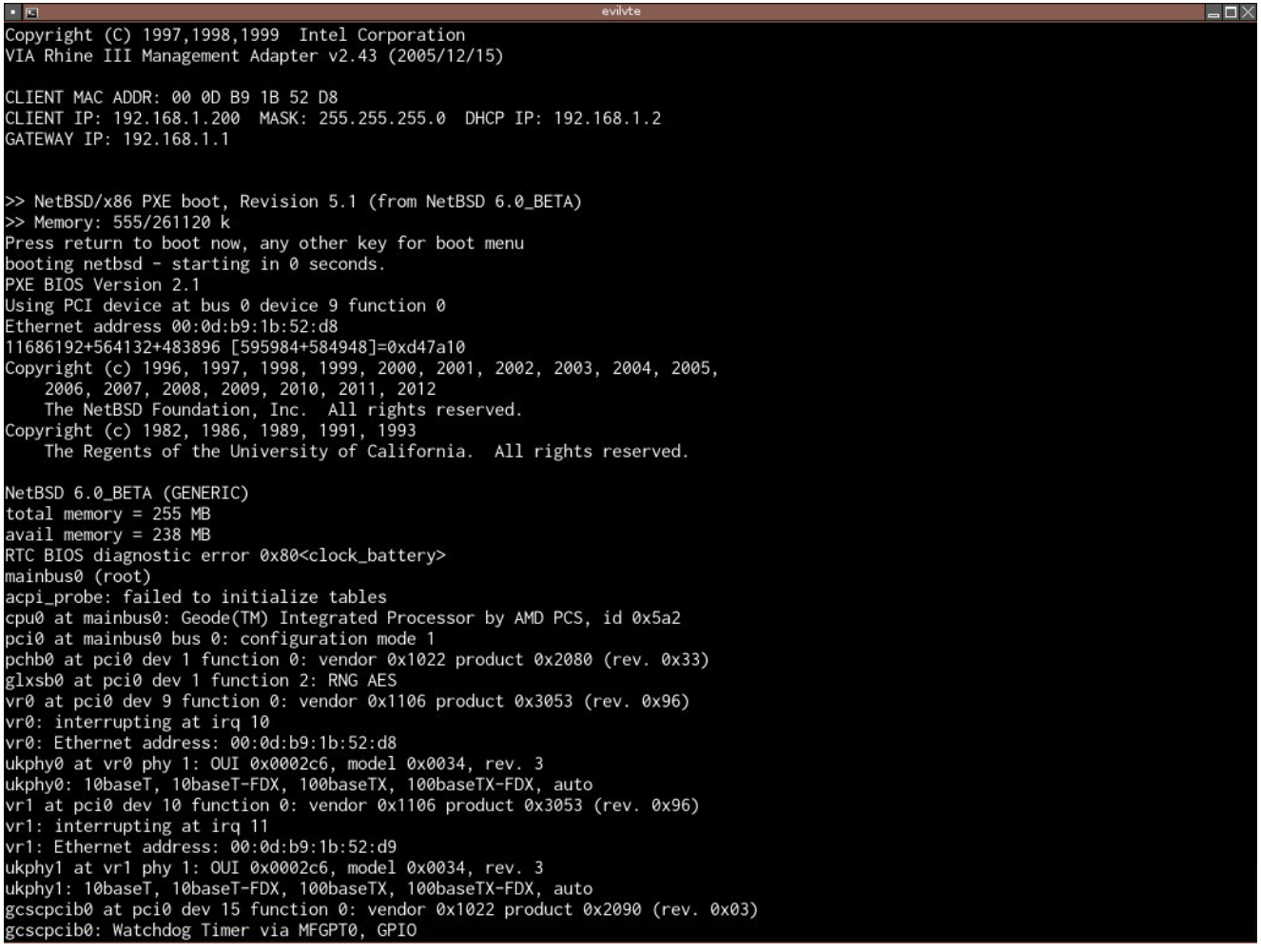
NetBSD: Diskless testing boot

Ready to boot the device for a test.

DISKLESS: PXE boot + NFS root



NetBSD: Booting the custom NetBSD via PXE and NFS



Compare these lines with a GENERIC kernel bootstrap

NetBSD: Something wrong?

Relax..

..go back,

review the kernel, fix,
recompile and reboot

NetBSD: NetBSD configuration

- sysinst installation via NFS
- Switch off unneeded services
- Disable ttys minus the one on serial COM0
- 'noatime' for root in fstab
- Create RAM disks for /var and /tmp
- Delete unused files
- Enable SSH
- Write a simpler 'init' file
- Root file system read-only

NetBSD: Alternative ways

Creating crunched binaries with crunchgen:

- no big statically linked binaries
- no dynamic linking infrastructure overhead

It combines all binaries and their libraries in one "super-binary", it use hard links (filesystem's alias for a program) to point the linked binary

makefs to wrap the filesystem in an image file

In Memory file-system:

- Embed root filesystem into the kernel
- increase the memory usage
- fast diskless device
- some MEMORY_DISK_ kernel options to enable

NetBSD: Alternative ways

Writing the OS in a CompactFlash card

- Tools: fdisk, disklabel, newfs, mount, build.sh
- Working on the development server
- We need a CompactFlash cards reader
- Slower, because the build-copy-reboot-test cycle
- Multiple copies can reduce the life of the flash card!

VOIP: Opensource VOIP software

- 

- 

VOIP: Installing FreeSwitch

- ## Download from the latest Git tree

```
$ git clone git://git.freeswitch.org/freeswitch.git
$ cd freeswitch
$ ./bootstrap
```

- ## Edit modules.conf

Disable all 'languages/mod_spidermonkey*'

Disable all the modules we don't need

- ## Configure

```
$ CFLAGS=-L/usr/pkg/lib ./configure
```

- ## Compile and Install

```
$ gmake
$ gmake install
$ gmake hd-sounds-install
$ gmake hd-moh-install
```

VOIP: Configuring FreeSwitch

XML configuration files in conf directory:

- SIP accounts
  /opt/freeswitch/conf/directory/*
- VoIP Gateways (Trunks)
  /opt/freeswitch/conf/sip_profiles/external
- Dialplan
  /opt/freeswitch/conf/dialplan/default/*

Configure some VOIP phones as well

VOIP: Configuring FreeSwitch

Example extension XML file:

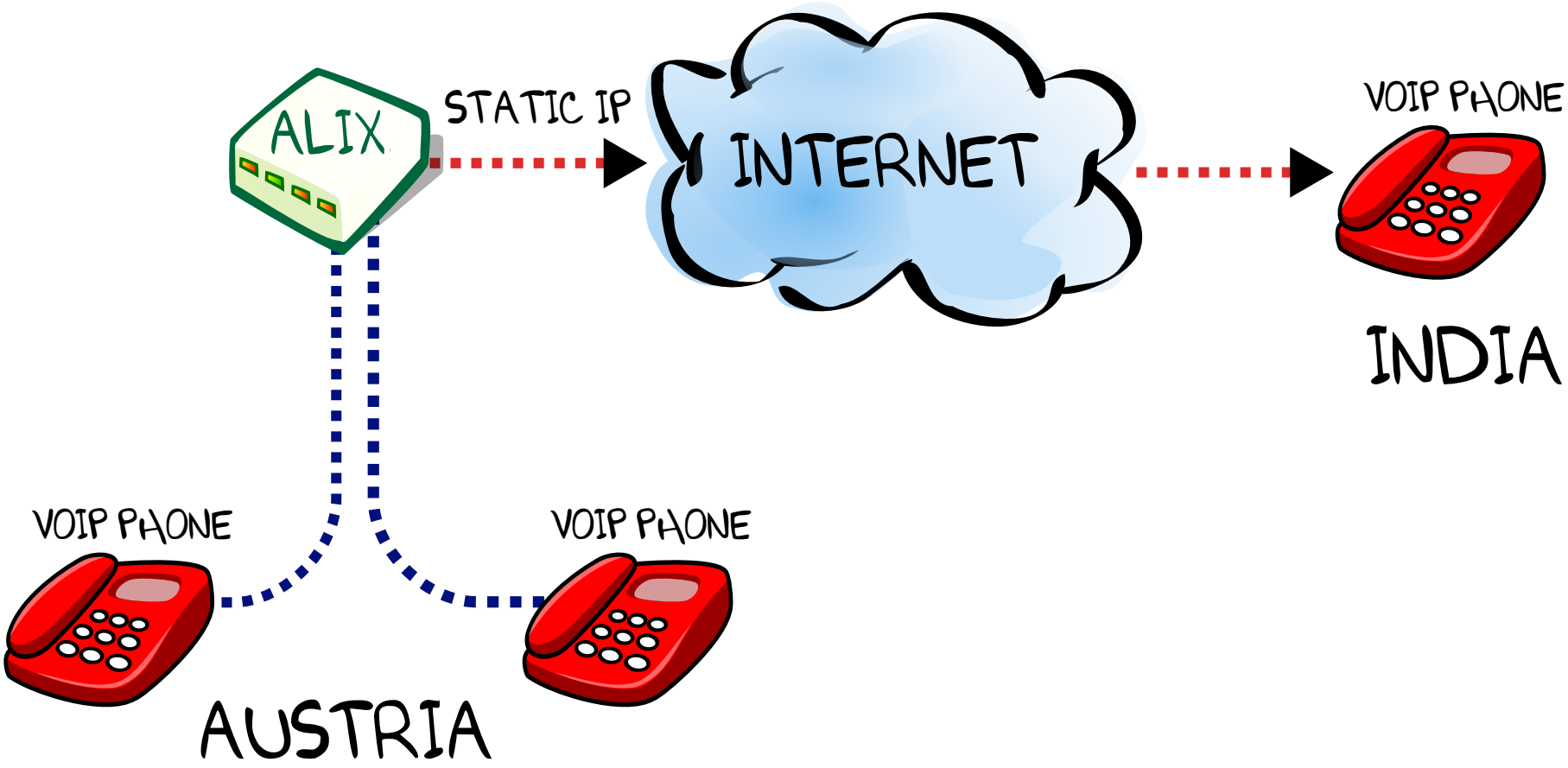
```
<include>
  <user id="1111" mailbox="1111">
    <params>
      <param name="password" value="1234"/>
      <param name="vm-password" value="1000"/>
    </params>
    <variables>
      <variable name="toll_allow" value="domestic,international,local"/>
      <variable name="accountcode" value="1111"/>
      <variable name="user_context" value="default"/>
      <variable name="effective_caller_id_name" value="Extension 1111"/>
      <variable name="effective_caller_id_number" value="1111"/>
      <variable name="outbound_caller_id_name" value="$${outbound_caller_name}"
      <variable name="outbound_caller_id_number" value="$${outbound_caller_id}"
      <variable name="callgroup" value="techsupport"/>
    </variables>
  </user>
</include>
```

VOIP: No Limits with pure VOIP calls!

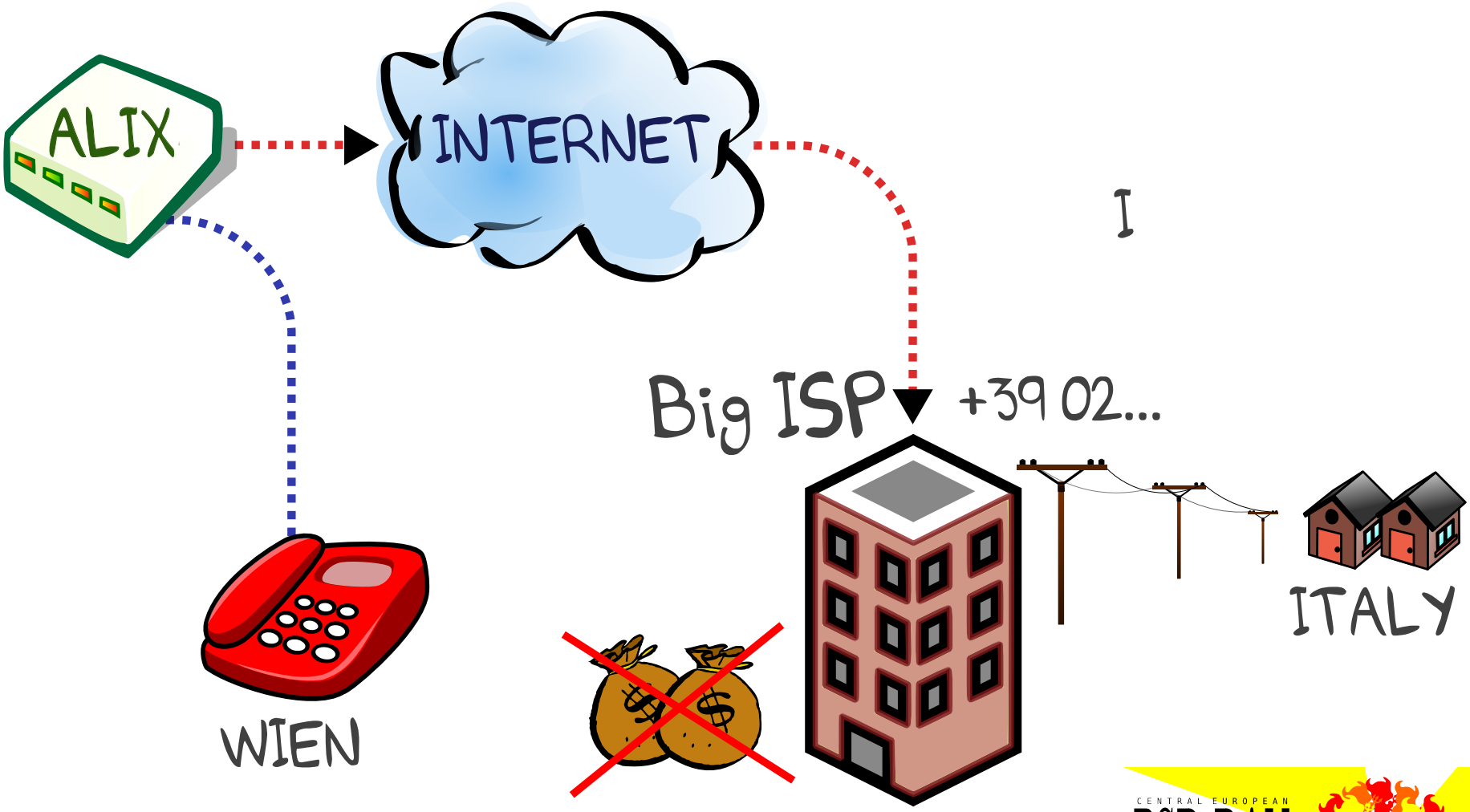
Everywhere

CENTRAL EUROPEAN
BSD-DAY
2012
VIENNA, AUSTRIA

VOIP: No Limits with pure VOIP calls!



VOIP: The Big Picture



VOIP: Some great ideas

Voicemail

- Leave messages in an email box
- mp3 or wav audio format
- Easy to organize and find
- Available everywhere
- Decentralized storage

VOIP: Some great ideas

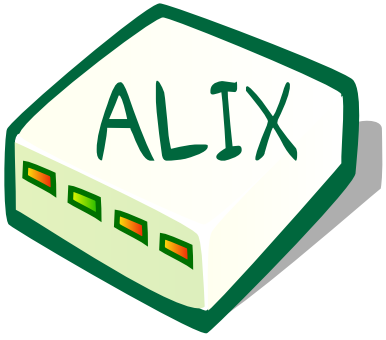
Your telephone number comes with you!

...everywhere:

- **NO PAY FOR LONG DISTANCES**
- No pay for extra services
- No needs to subscribe special contract options
- It works immediatly
- Outsurcing company sections abroad

Demo VOIP calls

The End: Enjoy your new mobile PBX



Thank you for your attention!

flood me questions :-)

fabiodive@gmail.com

`https://github.com/fabiodive/BSDDay-2012-Wien`