PHILIPS

The Video Clip Player:
Philips Nexperia™ PNX0106 and Linux based platform

Armin Gerritsen & Marcin Klecha
Advanced Systems Laboratory

08-04-2006

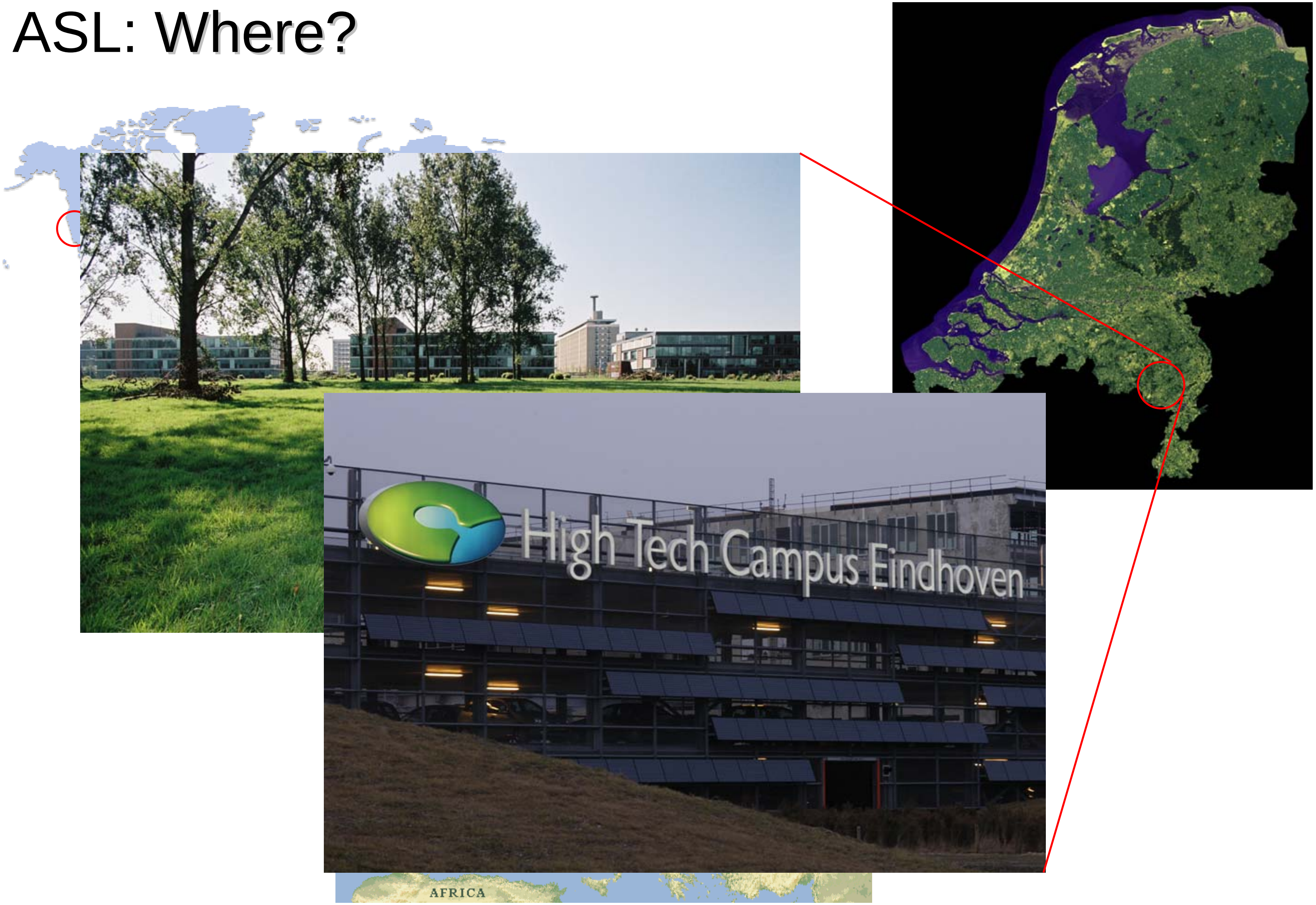
Advanced Systems Laboratory

- Vision
  - A world where everyone can always connect to information, entertainment and services.
- Mission
  - To prototype advanced applications, one or two “Moore’s Law” generations ahead, using Nexperia™ platforms.
  - To gain insight on future requirements, identifying disruptive market and technology trends.
- Strategy
  - Show PS’ strengths in “Connected Consumer” through prototype demonstrators built on Nexperia™.
  - Develop market understanding through strategic customer engagement together with Business Clusters and Marketing and Sales.
  - Link program to business direction through Business Relevance and Technology Program Boards.

ASL: Where?


Arctic Ocean
North Cape
URAL MTS.
Norwegian Sea
EUROPE
NORTHERN EUROPE
North Atlantic Ocean
North Sea
Baltic Sea
Volga
EASTERN EUROPE
English Channel
Rhine
Odra
Don
Dnieper
WESTERN EUROPE
Loire
CARPATHIAN MTS.
CAUCASUS MTS.
El'brus
ALPS
PYRENEES
Drava
Cabo da Roca
Black Sea
ASIA
Danube
MIDDLE EAST
Mediterranean Sea
AFRICA

ASL: Where?



PHILIPS

Outline

- Video Clip Player
  - Introduction
  - System Overview and Use Cases
- Linux & Software architecture
  - Video: Framebuffer Implementation
  - Audio: EPICS DSP
  - UPnP and DTCP-IP protection
  - Linphone: VoIP application
- Linux and prototyping
  - History
  - Measurements: powermanagement
  - Next steps

Video Clip Player: Introduction

The Video Clip Player (VCP) shows Linux-based applications and is based around the Nexperia™ PNX0106 System-on-Chip solution.

The VCP is hard-drive based reference design featuring a colour QCIF+ LCD and provides features such as FM radio, audio playback, photo viewing and video playback on LCD or TV.

The VCP has multiple connectivity options like USB 2.0, USB OTG, UART, SPI, BlueTooth, NFC and Ethernet.

The VCP is using the latest technologies like UPnP, VoIP and DTCP-IP.

PHILIPS

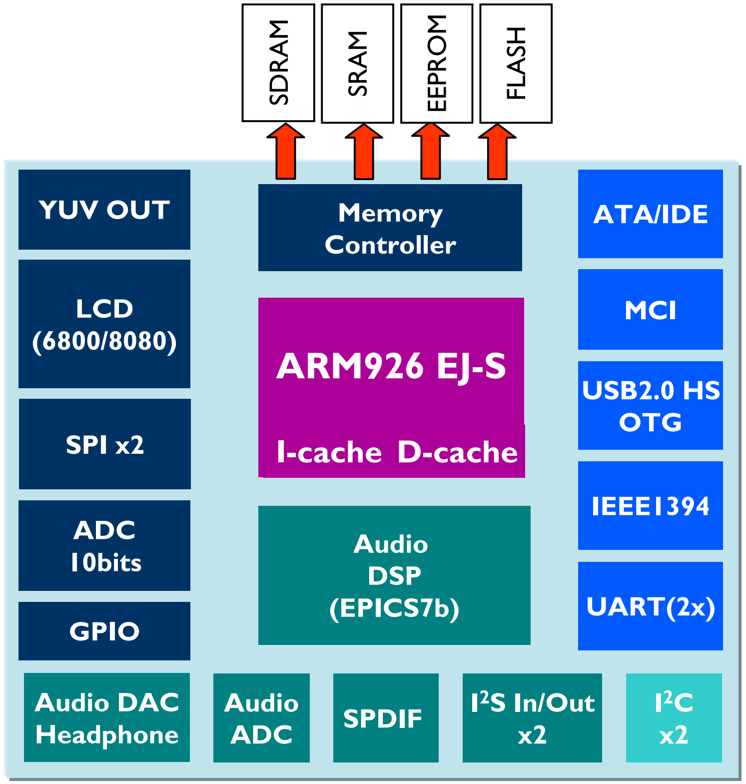
Video Clip Player: Introduction

And what about Linux?

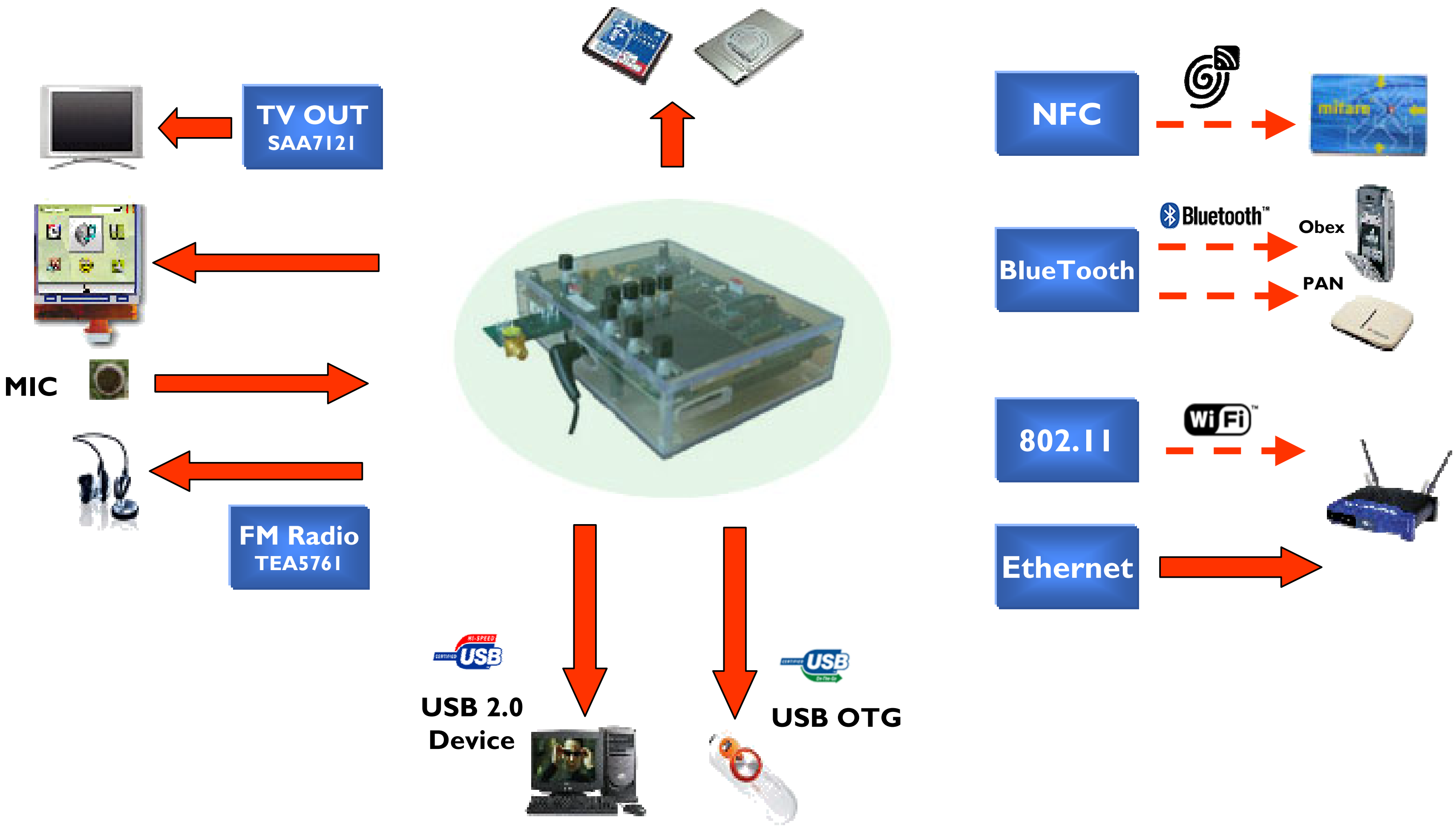
Linux played an essential role in the development and usage of the VCP in three ways

- The VCP runs Linux. Linux enabled Philips Semiconductors to quickly have a demonstrator platform to promote its Nexperia™ PNX0106 IC.
- The VCP can be easily used as a test-platform for Philips departments, to test the latest ideas and technologies.
- The VCP was prototyped using Linux, allowing IP to be validated, actual use-cases to be tested and performance measurements to be done, all before the IC itself was available.

Video Clip Player: Nexperia™ PNX0106

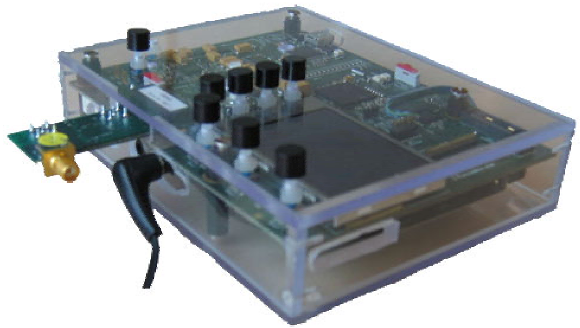


Video Clip Player: System overview



Video Clip Player: Use Cases

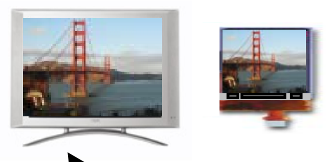
- Audio/Video Player (showing MP3 and MPEG4)
- Audio Recording
- FM Radio
- Pictures Slideshow (LCD and TV)
- USB OTG (Pictures Download from a camera)
- USB 2.0 Device – PC Connection
- VoIP (Voice-over-IP)
- Send/Receive pictures via BlueTooth
- iRadio – audio streaming over TCP/IP
- Games
- UPnP Server with DTCP-IP link protection
- NFC mifare card reader



Video Clip Player: Audio/Video/Pictures

Main window of the GUI

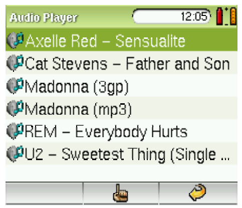
Video Player
Audio Player
Audio Applications
Pictures
Games
Settings



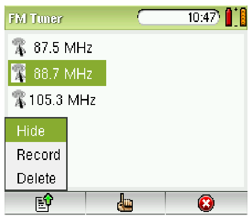
JPEG slideshow
to a TV or LCD

Audio/Video playback to LCD/TV
-.3gp, *.mp4*
-MPEG-4 SP + AMR Audio

Audio MP3 playback
from a CF/HDD via a headphone or speakers



Audio Recording and playback of the recording
From a MIC on the board



FM Tuner - Radio

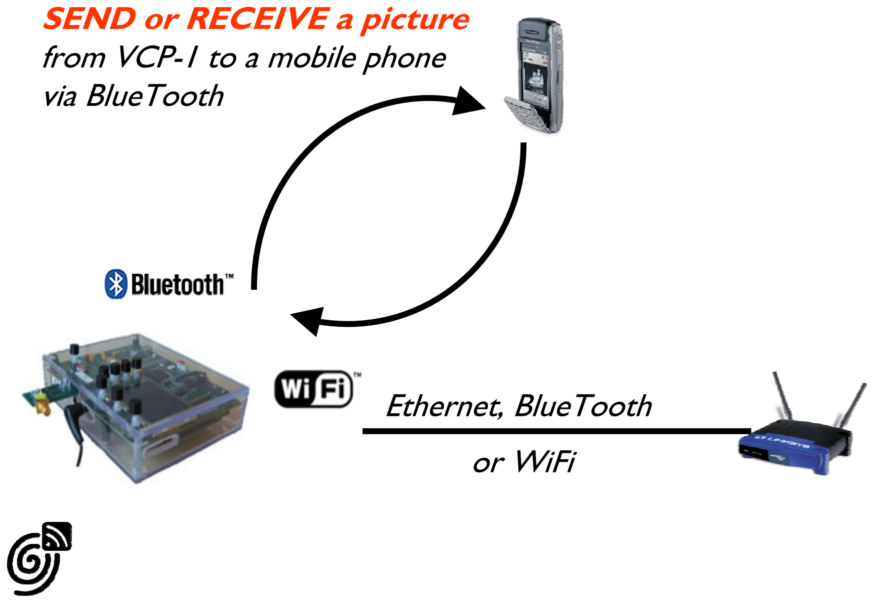
Video Clip Player: Connectivity and Networking

PHILIPS

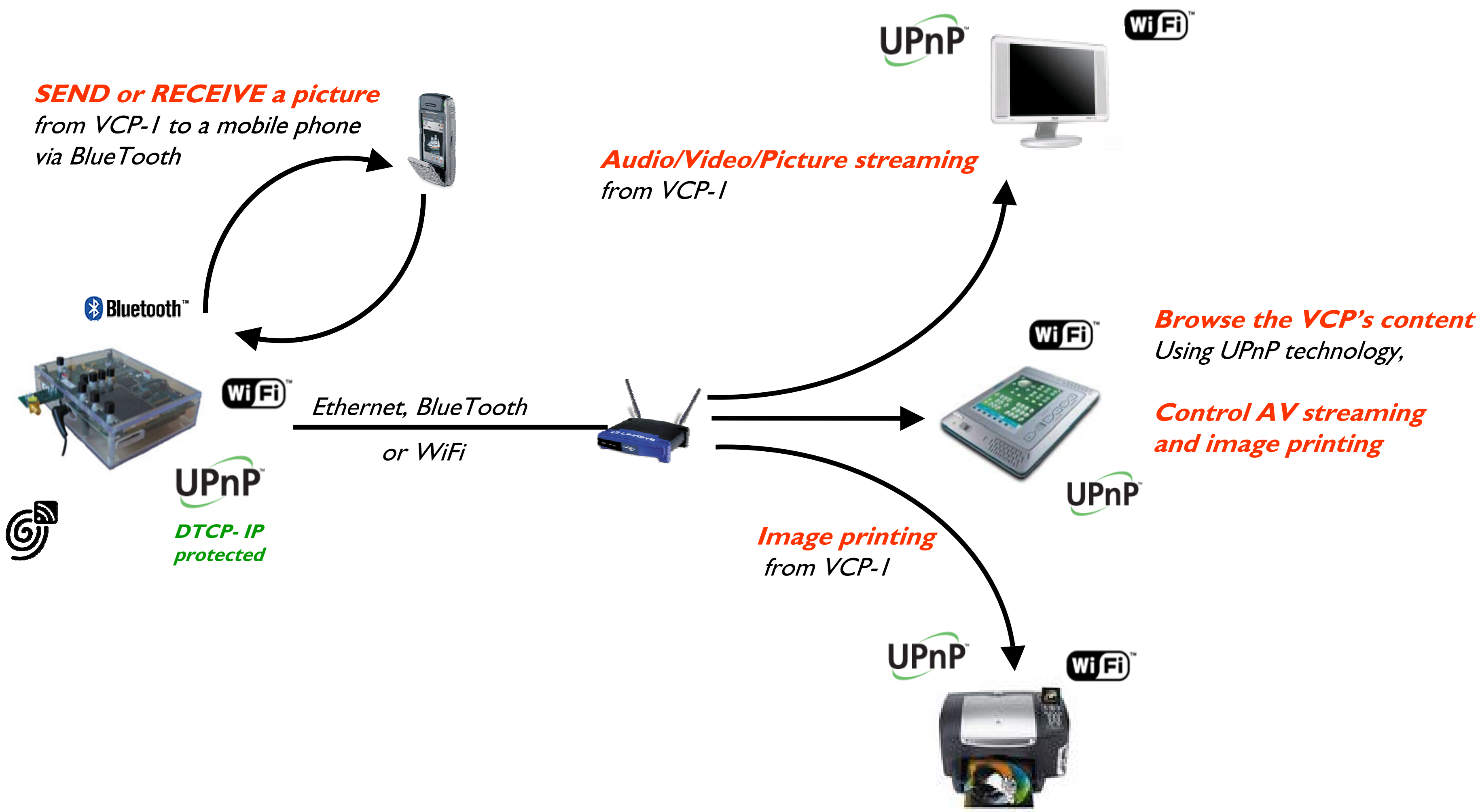
Video Clip Player: Connectivity and Networking

SEND or RECEIVE a picture
from VCP-1 to a mobile phone
via BlueTooth

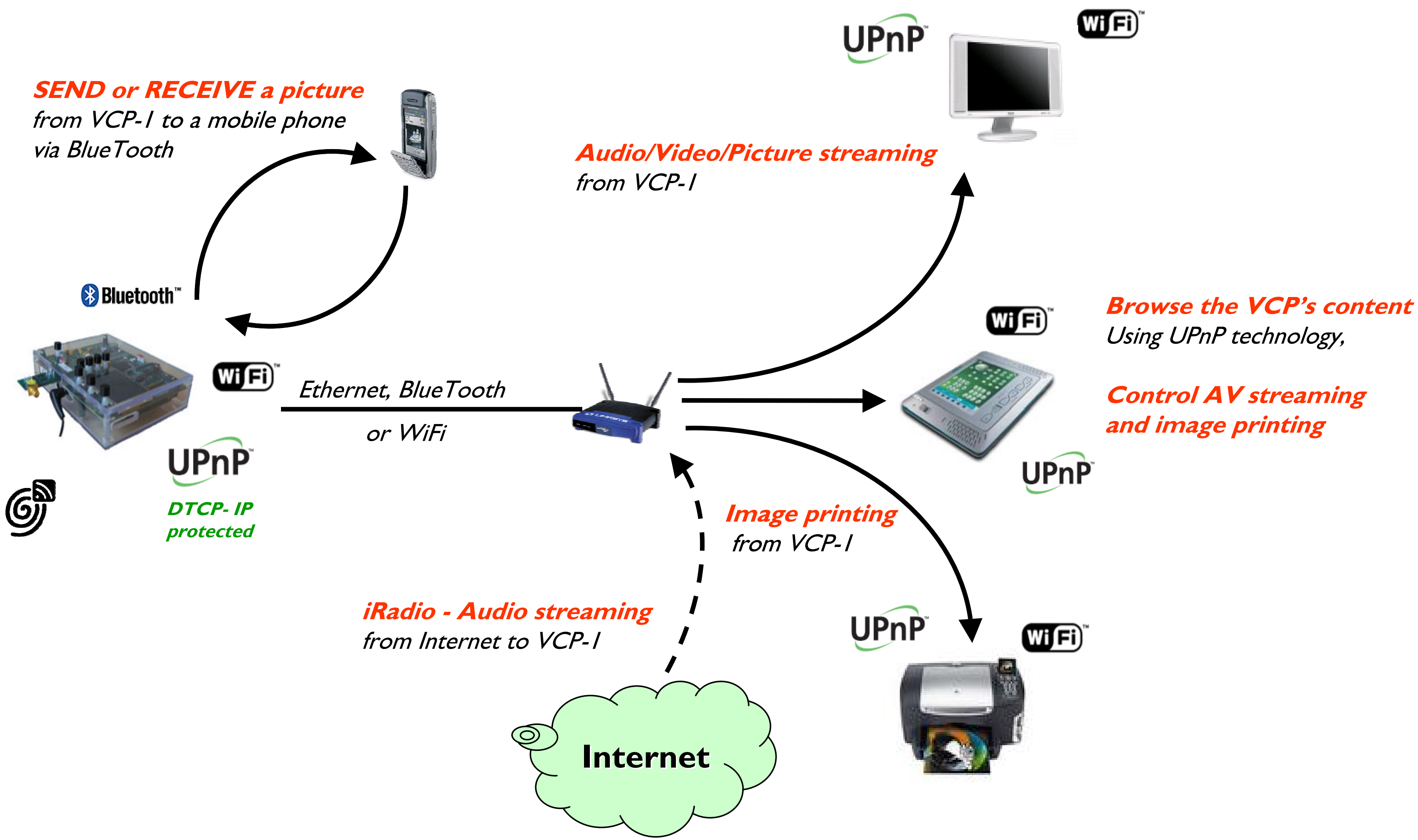
Video Clip Player: Connectivity and Networking



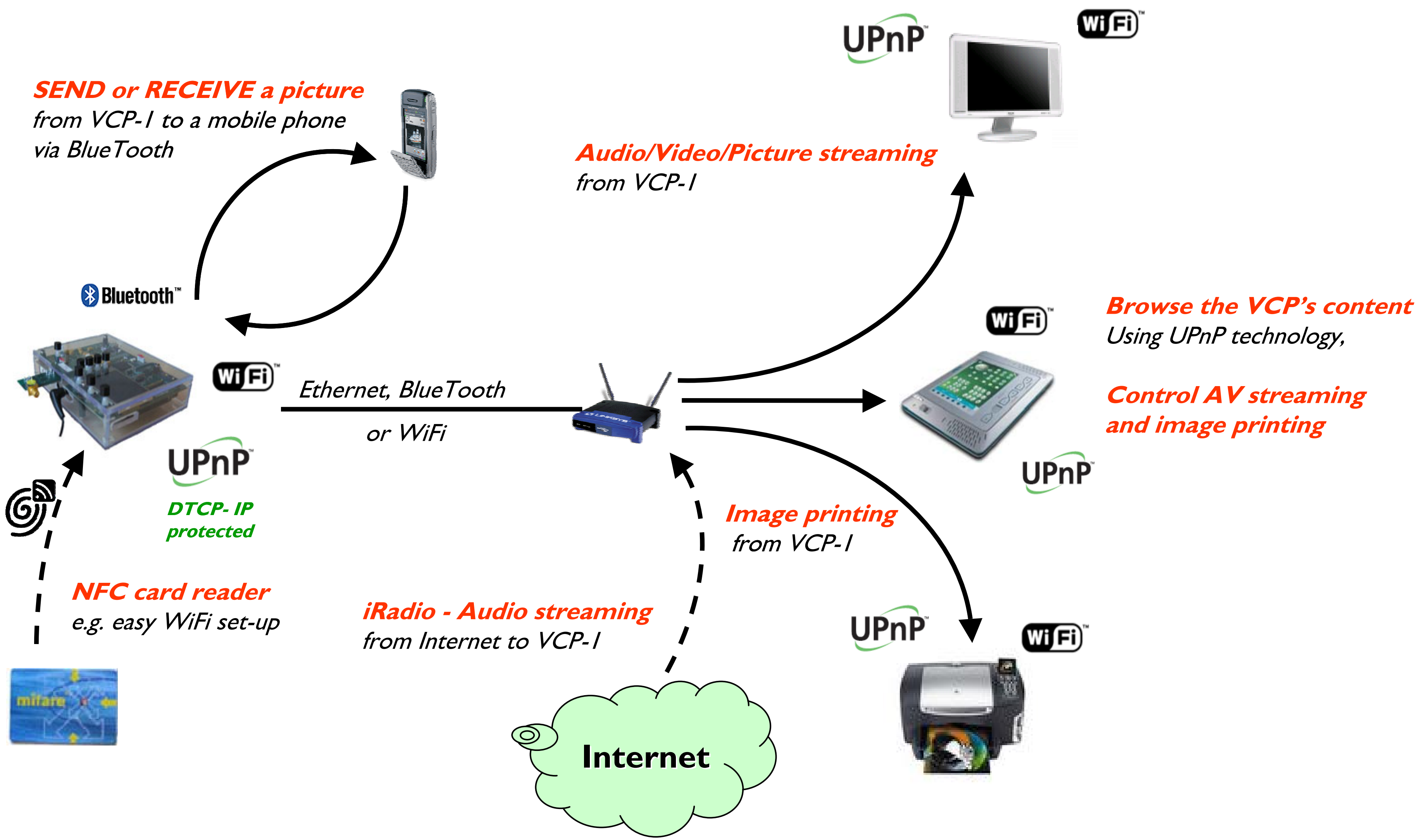
Video Clip Player: Connectivity and Networking



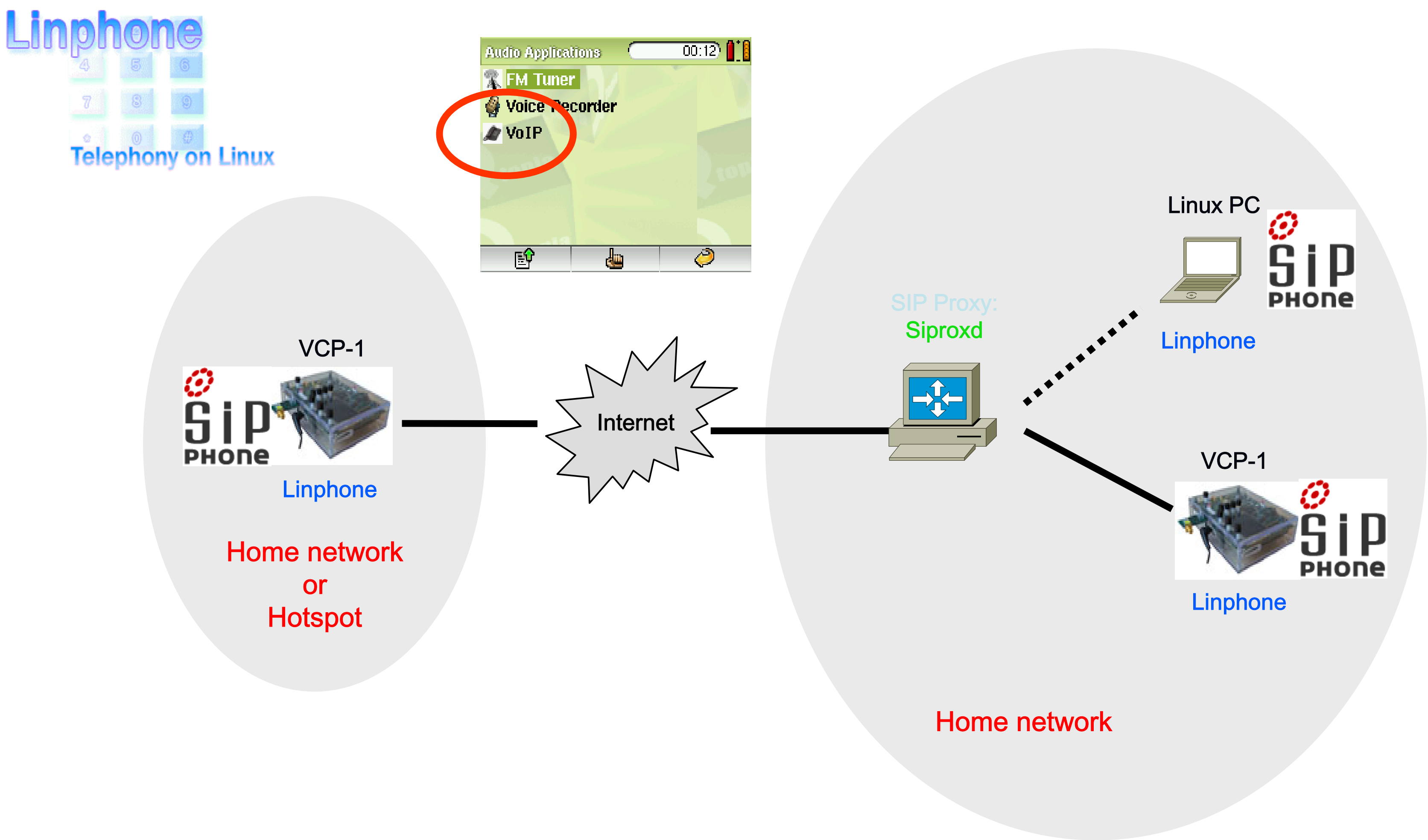
Video Clip Player: Connectivity and Networking



Video Clip Player: Connectivity and Networking



Video Clip Player: VoIP (SIP) telephony



PHILIPS

Outline

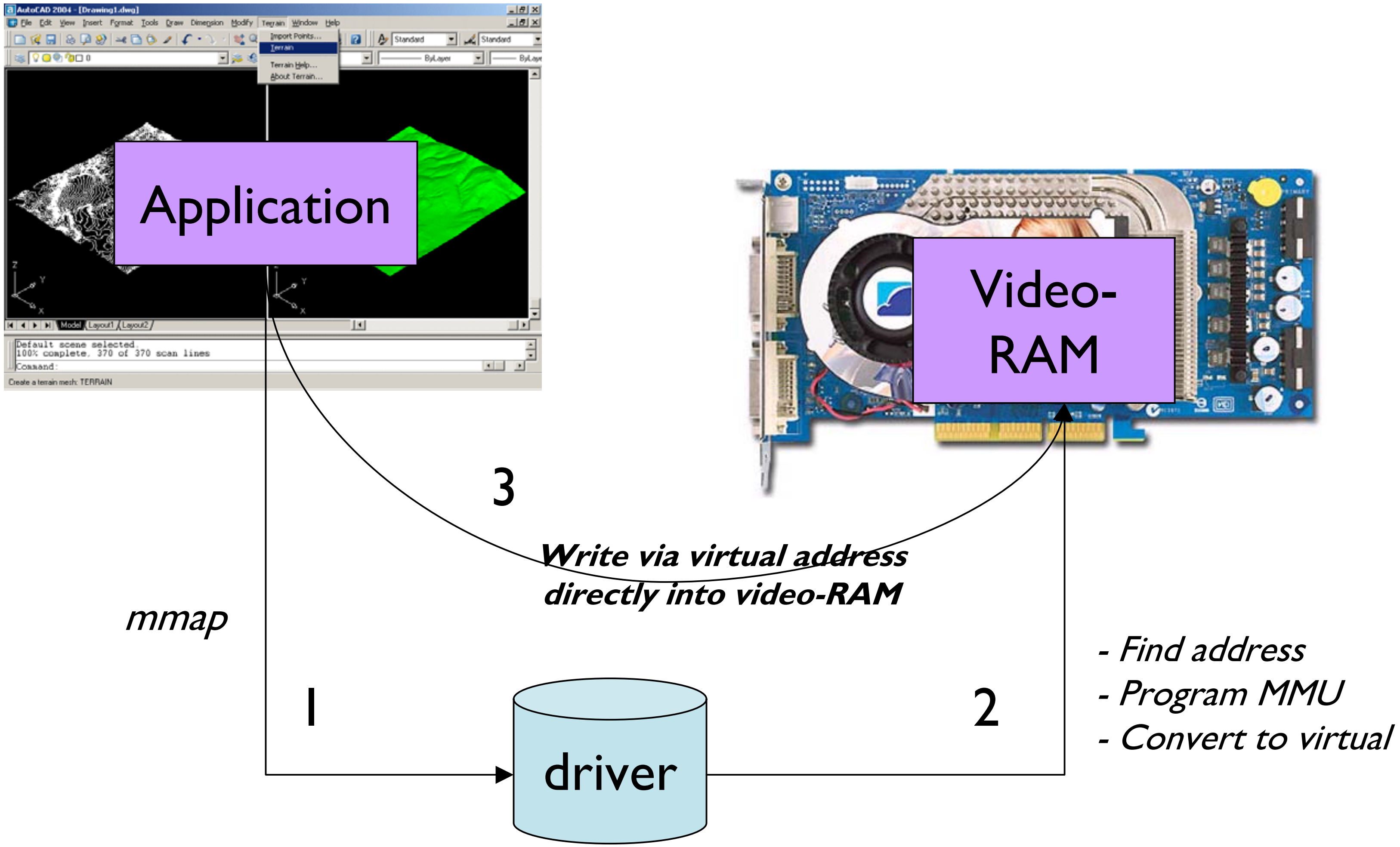
- Video Clip Player
  - Introduction
  - System Overview and Use Cases
- Linux & Software architecture
  - Video: Framebuffer Implementation
  - Audio: EPICS DSP
  - UPnP and DTCP-IP protection
  - Linphone: VoIP application
- Linux and prototyping
  - History
  - Measurements: powermanagement
  - Next steps

Linux & Software architecture: Framebuffer

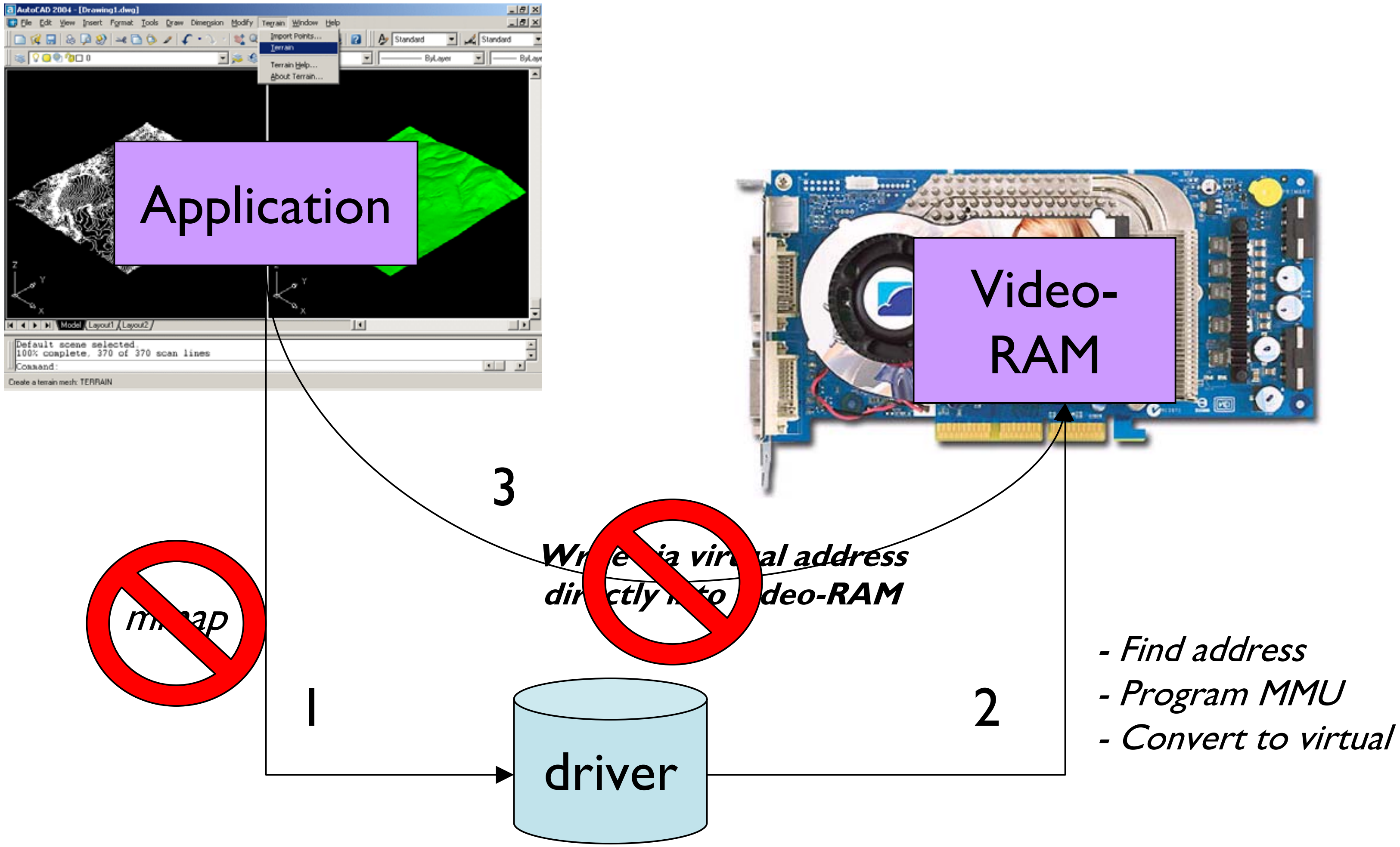
- VCP is using the Linux framebuffer architecture
  - Enables the usage of many of the Linux applications and graphic libraries (e.g. Qt(e)/Qtopia and DirectFB)

- 'Out of the box' the Linux framebuffer architecture doesn't support typical LCD's for CE devices
  - The absents of address lines prevents applications from directly accessing the memory of the LCD.
  - Therefore a virtual memory buffer is required, which needs to be synchronised with the real LCD memory.

Linux & Software architecture: Framebuffer



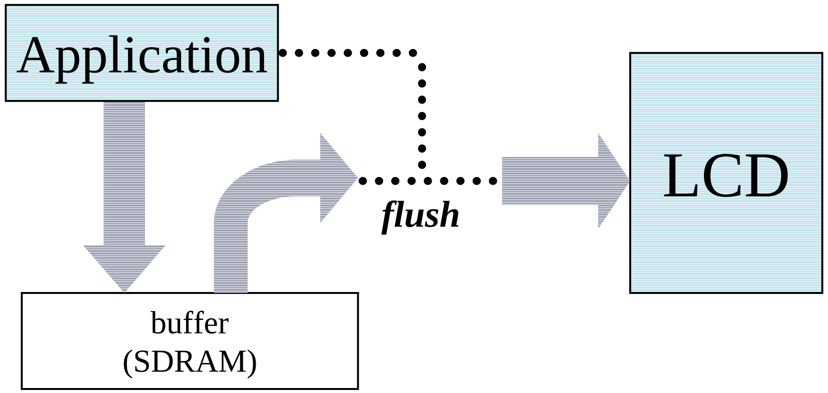
Linux & Software architecture: Framebuffer



PHILIPS

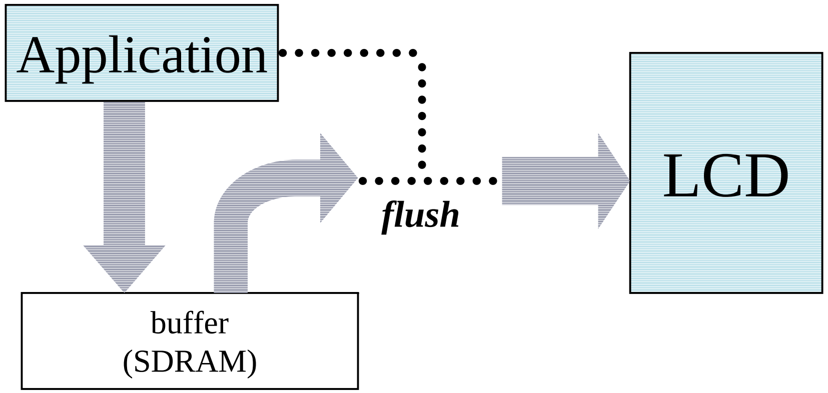
Linux & Software architecture: Framebuffer

- How to use the framebuffer architecture, if you cannot mmap its memory?



Linux & Software architecture: Framebuffer

- How to use the framebuffer architecture, if you cannot mmap its memory?



Solution: Use an intermediate buffer

=> *Very small application modification required*

Linux & Software architecture: Framebuffer

- The VCP implementation provides the following solutions
  - Manual or a timer based synchronisation
  - Using DMA to do synchronisation in the background

- *Making a virtue of necessity*: During synchronisation use hardware features in the LCD and/or PNX0106 to do accelerated video post processing.
  E.g.
  - flipping the screen
  - endianess conversion
  - colour space conversion
  - windowing
  - … *and more …*

Linux & Software architecture: Framebuffer

- The VCP implementation provides the following solutions
  - Manual or a timer based synchronisation
  - Using DMA to do synchronisation in the background

- *Making a virtue of necessity*: During synchronisation use hardware features in the LCD and/or PNX0106 to do accelerated video post processing.

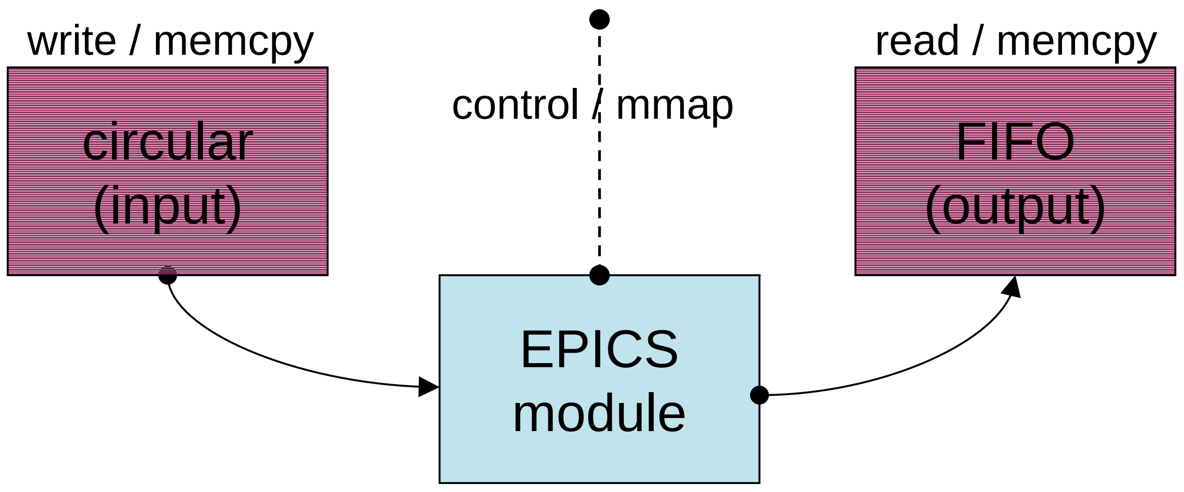
  E.g.
  - flipping the screen
  - endianess conversion
  - colour space conversion
  - windowing
  - … *and more …*

Extensions to the architecture, which don't break compatibility!

Linux & Software architecture: Audio

EPICS DSP

- The PNX0106 has a build in EPICS DSP
  - Firmware loadable via hotplug or udev
  - Statistics and control via *sysfs* and *ioctl*
  - /dev/epics/[wma/mp3/…]



Linux & Software architecture: Audio

EPICS DSP

- Example of loading using *udev*

```
mkdir /dev/epics
mknod /dev/epics/mp3 c 254 0
mknod /dev/epics/wma c 254 1
insmod e7b.ko &
sleep 1
echo 1 > /sys/class/firmware/epics0/loading
cat /usr/lib/firmware/epics_firmware > /sys/class/firmware/epics0/data
echo 0 > /sys/class/firmware/epics0/loading
```

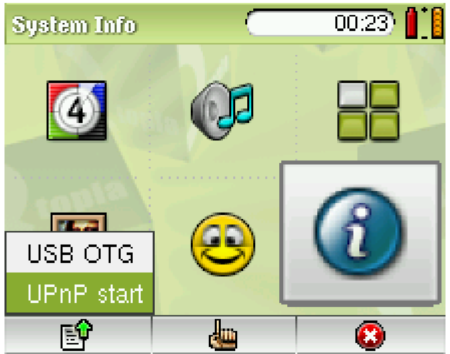
Outline

- Video Clip Player
  - Introduction
  - System Overview and Use Cases
- Linux & Software architecture
  - Video: Framebuffer Implementation
  - Audio: EPICS DSP
  - UPnP and DTCP-IP protection
  - Linphone: VoIP application
- Linux and prototyping
  - History
  - Measurements: powermanagement
  - Next steps

Linux & Software architecture: UPnP

- Universal Plug and Play (UPnP) is an architecture for peer-to-peer network connectivity of intelligent devices or appliances, particularly within the home.
  - UPnP uses internet standards and technologies, such as TCP/IP, HTTP, and XML
  - Allows devices to automatically connect and work together.

- UPnP facilitates enabling simple and reliable connectivity between stand-alone devices, and easy configuration into a home network.
  - Without any need for user intervention, a new UPnP-enabled device is automatically registered and configured into the network.
  - This is including an announcement of the services and content the device has available.



Linux & Software architecture: UPnP

Bluetooth™

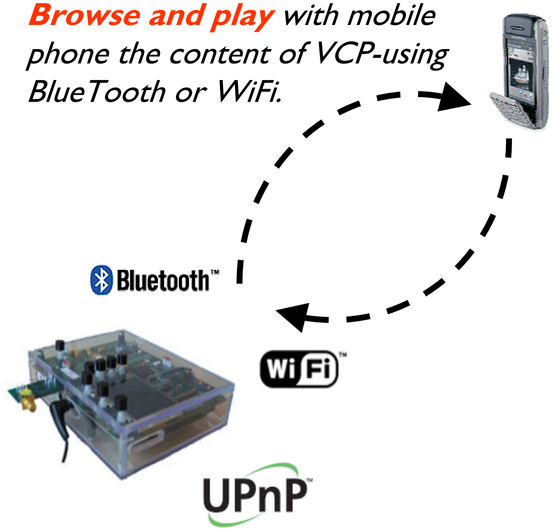
WiFi™

UPnP™

DTCP- IP protected

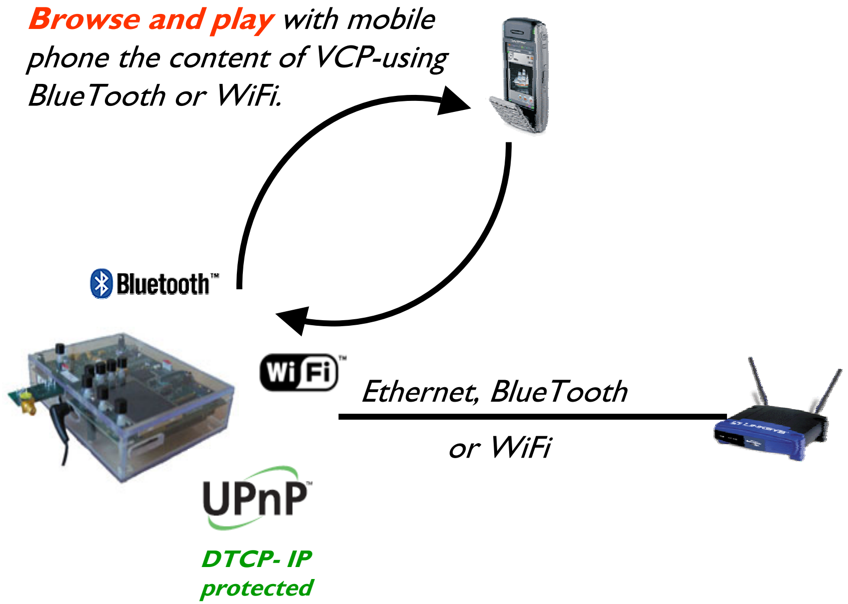
Linux & Software architecture: UPnP

Browse and play *with mobile phone the content of VCP-using BlueTooth or WiFi.*

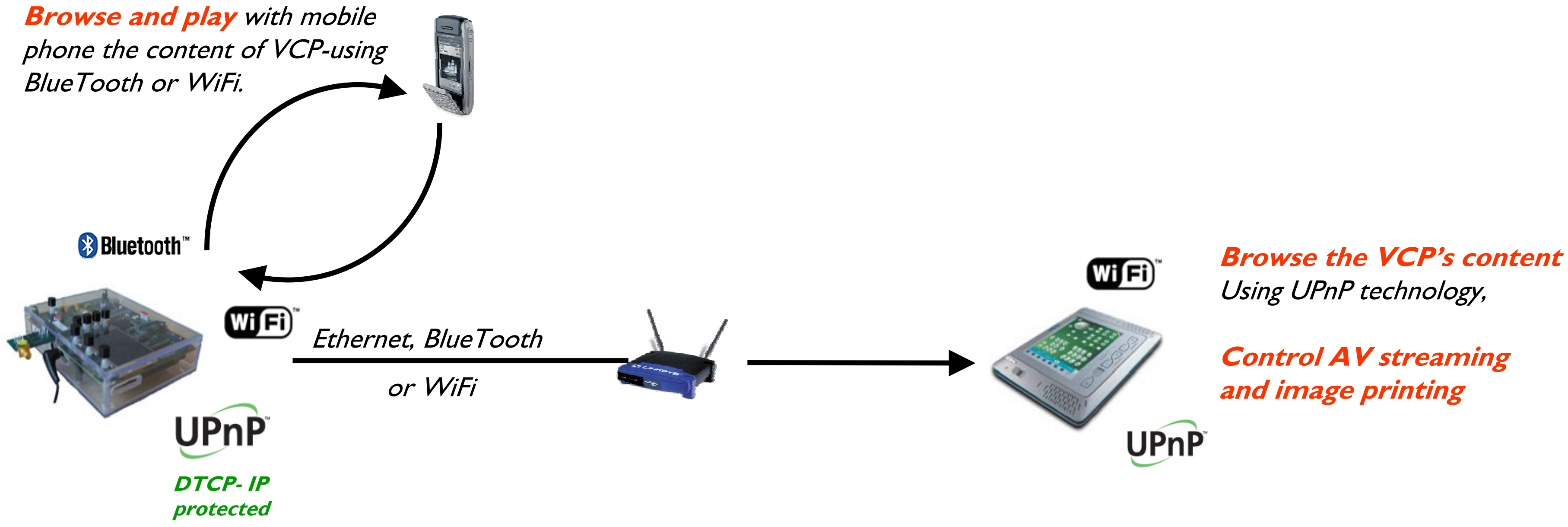


DTCP- IP protected

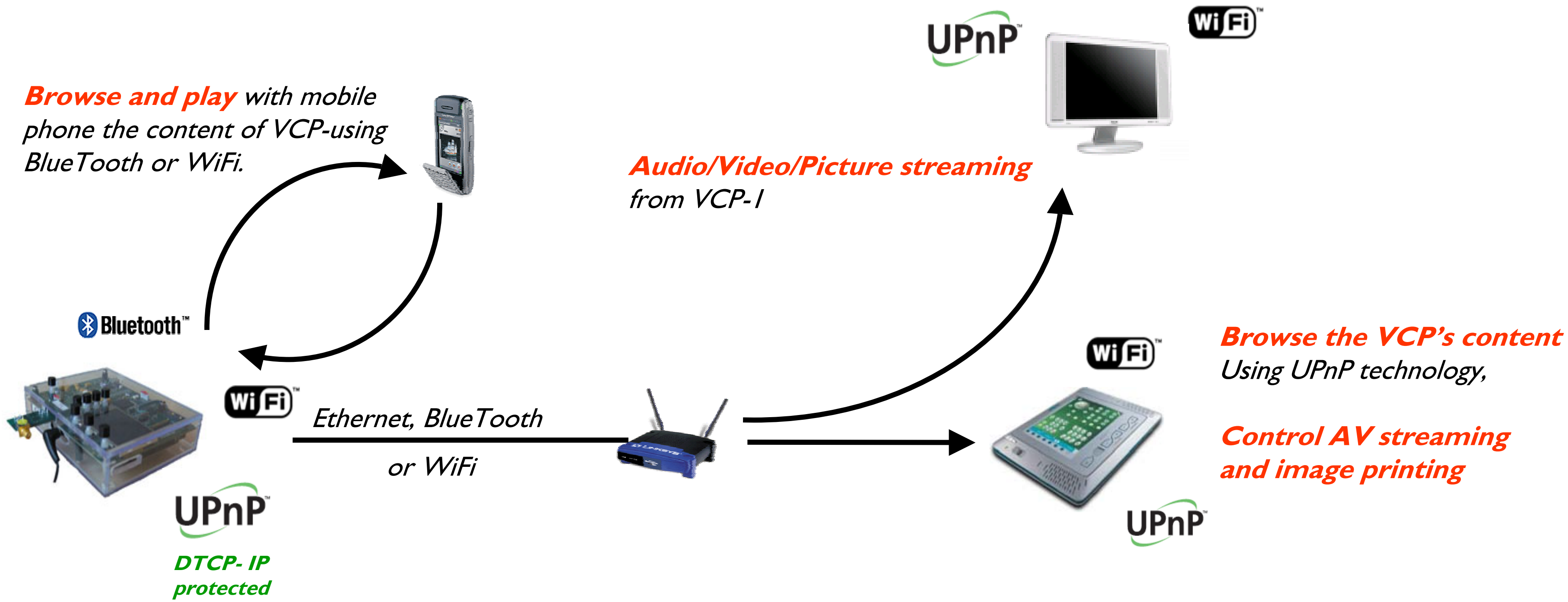
Linux & Software architecture: UPnP



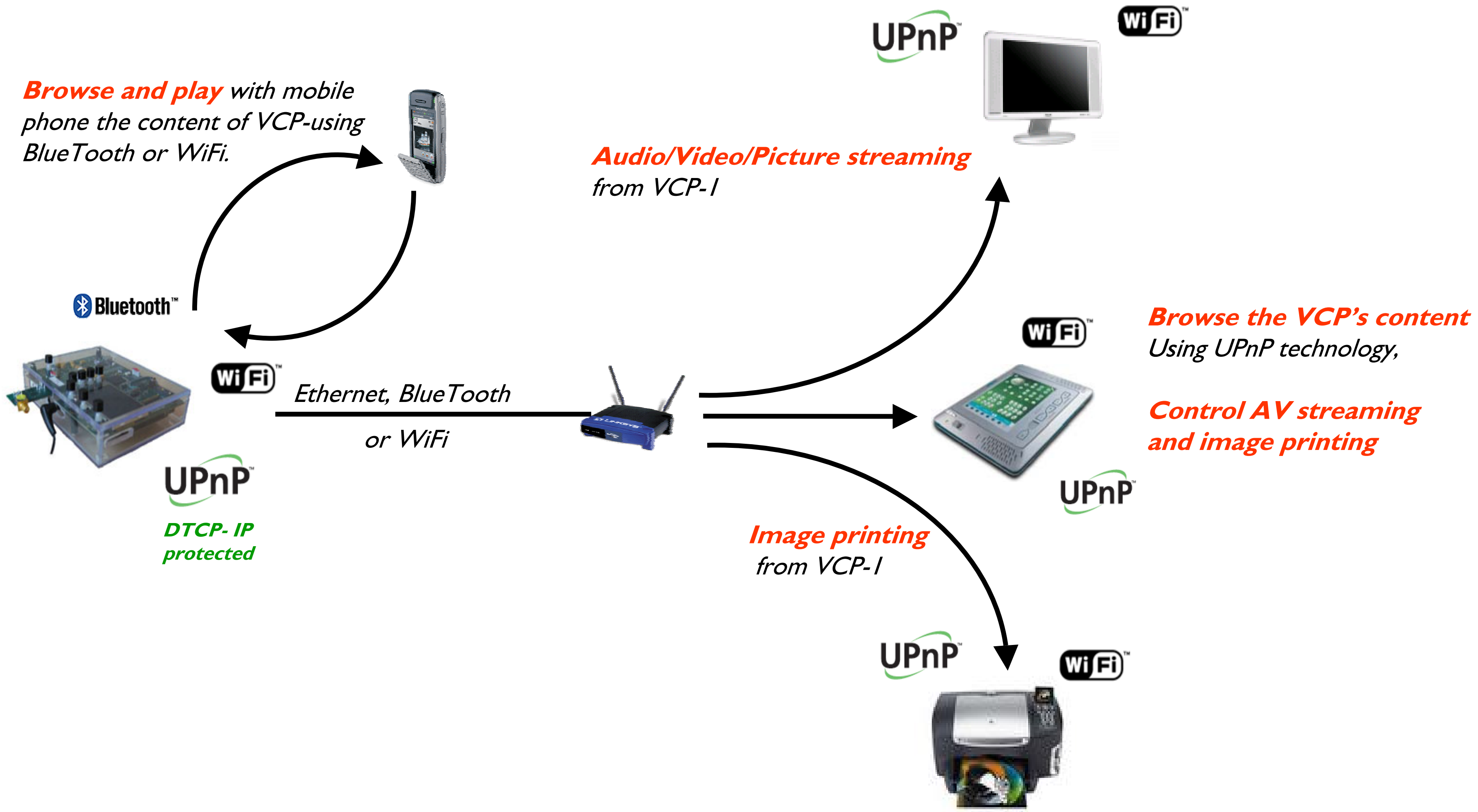
Linux & Software architecture: UPnP



Linux & Software architecture: UPnP



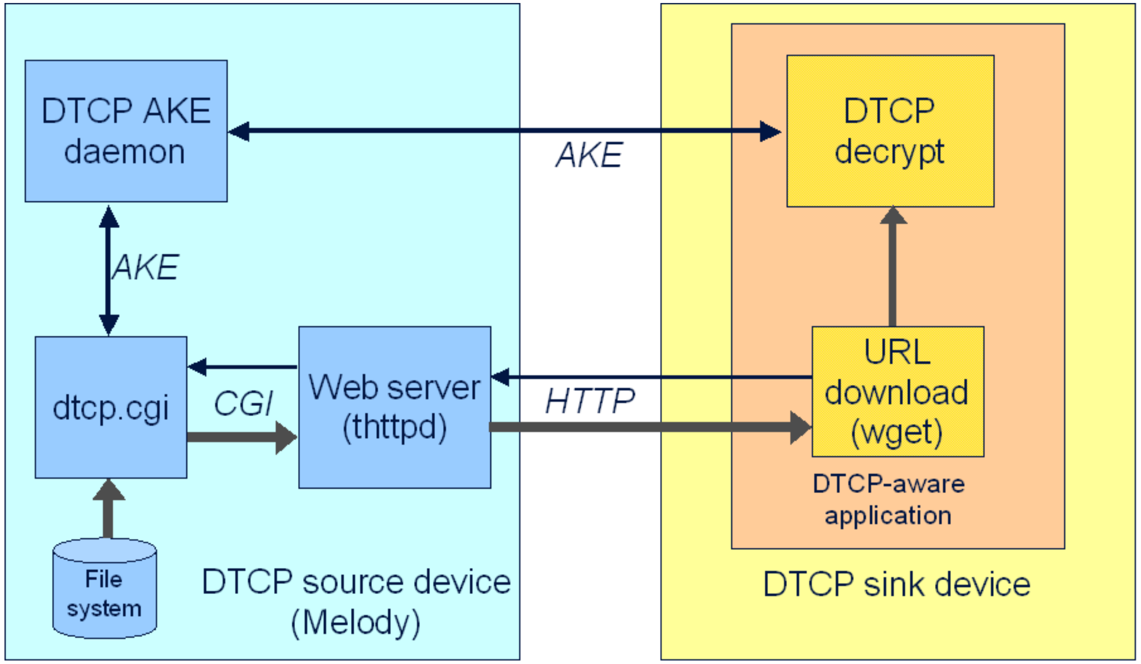
Linux & Software architecture: UPnP



Linux & Software architecture: DTCP-IP

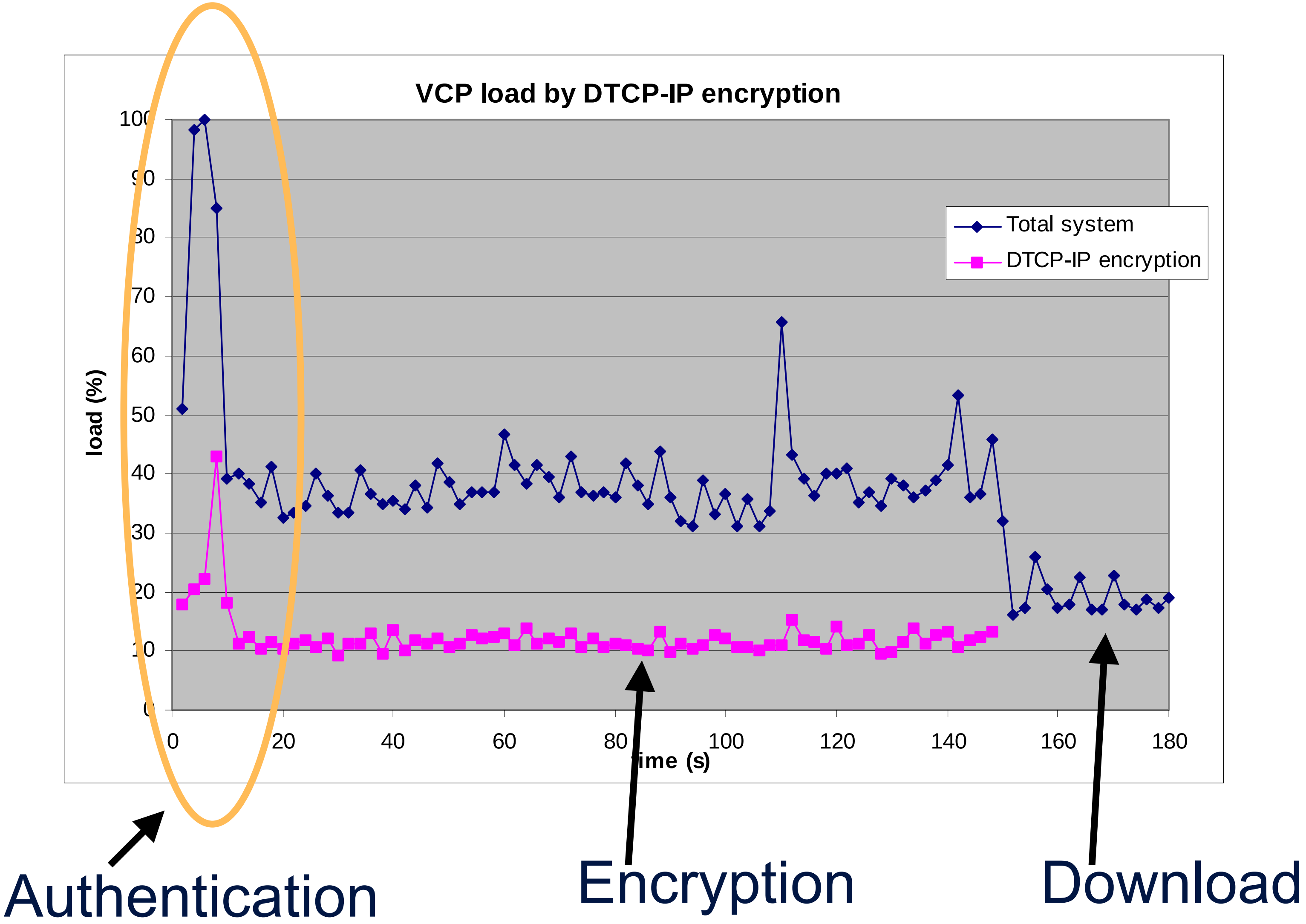
- The DTCP-IP protection is optional by DLNA

Digital Living Network Alliance (DLNA) aligns industry leaders in the CE, mobile, and PC industries. It provides a shared vision of a wired and wireless interoperable network devices in the home enabling a seamless environment for sharing and growing new digital media and content services.



DTCP-IP	: Digital Transmission Content Protection over Internet Protocol
CGI	: Common Gateway Interface
AKE	: Authentication and Key Exchange

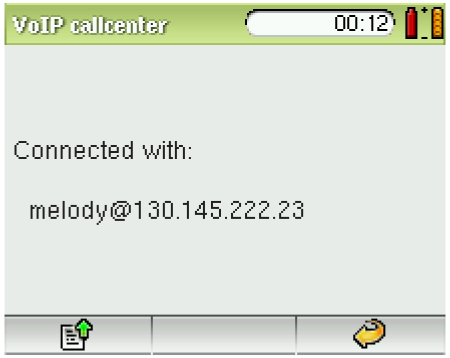
Linux & Software architecture: DTCP-IP

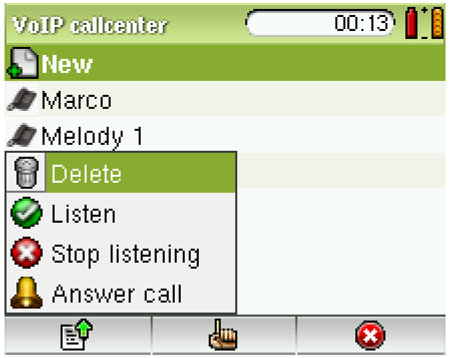


PHILIPS

Linux & Software architecture: Linphone (VoIP)

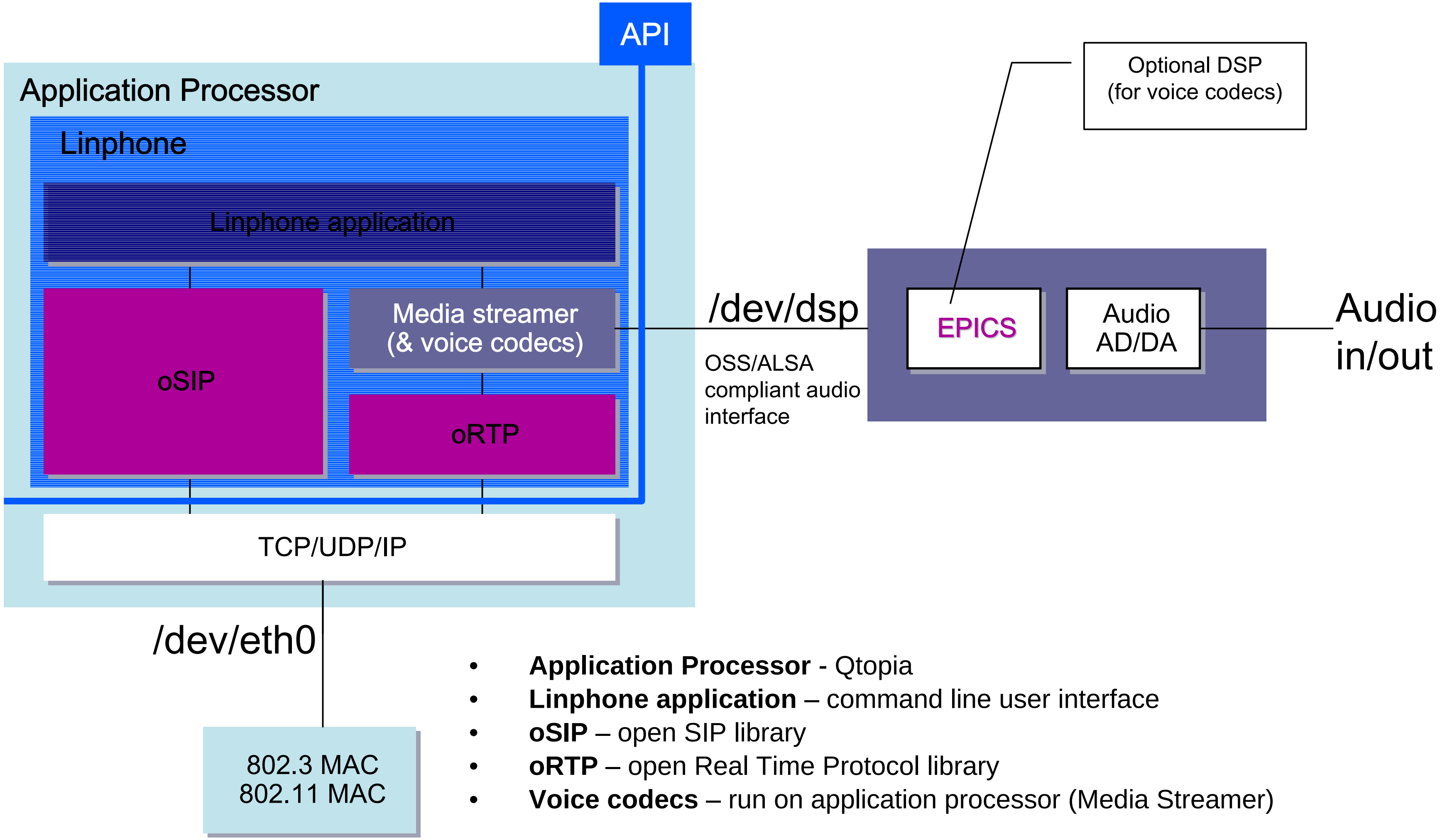
- GUI integration in Qtopia





- At default the VoIP-application waits for calls when started
- 'Waiting' is stopped when application exits

Linux & Software architecture: Linphone (VoIP)



- **Application Processor** - Qtopia
- **Linphone application** – command line user interface
- **oSIP** – open SIP library
- **oRTP** – open Real Time Protocol library
- **Voice codecs** – run on application processor (Media Streamer)

Linux & Software architecture: Linphone (VoIP)

Implementation details

- Protocols
  - TCP/UDP/IP
  - SIP, RTP protocols provided with the application (oSIP, oRTP)

- Codecs running on the application processor or EPICS DSP
  - A/μ-law PCM
  - GSM and Speex

- Interfaces
  - Ethernet and/or WiFi (/dev/ethX)
  - OSS compliant audio interface

Outline

- Video Clip Player
  - Introduction
  - System Overview and Use Cases
- Linux & Software architecture
  - Video: Framebuffer Implementation
  - Audio: EPICS DSP
  - UPnP and DTCP-IP protection
  - Linphone: VoIP application
- Linux and prototyping
  - History
  - Measurements: powermanagement
  - Next steps

Linux and prototyping: History

Proof of concept

Prototyping

Validation

Evaluation Board

Demo & Development

VCP-1

PNX0106

Reference design

VCP-2

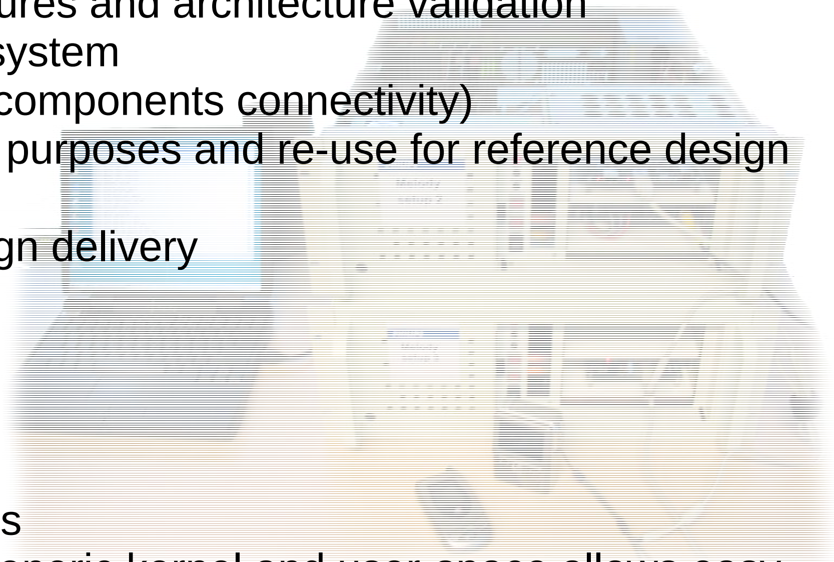
PNX0106

Linux enabled a smooth transition from a prototyping system to a very advanced and compact design with extended networking and connectivity capabilities.

PHILIPS

Linux and prototyping: History

- Why Prototyping?
  - – Proof of concept – verification of new features and architecture validation
  - – IP (subsystems) verification in the whole system
  - – A prior verification of a reference design (components connectivity)
  - – Development of prototyping SW for demo purposes and re-use for reference design evaluation
  - – Reduce time and risk for a reference design delivery



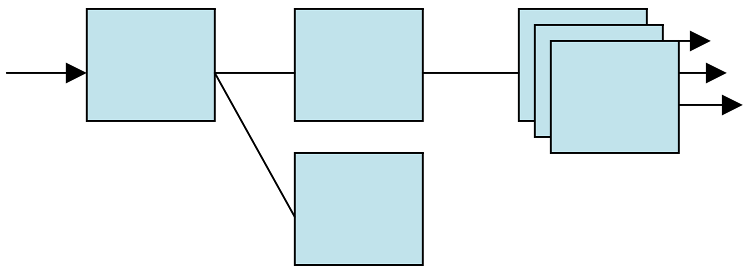
- Why use Linux?
  - – Prototype platforms often have it provided
  - – CE-OS'es often require expensive licences
  - – Linux clean separation between drivers, generic kernel and user-space allows easy transfers from prototype to validation to demonstration
  - – Therefore ***quick*** results!

Linux and prototyping: Dynamic Power Measurements

CGU driver

- The Clock Generation Unit (CGU) of the PNX0106 allows many different clocksettings:
  - base clocks
  - dividers
  - independent spread stages

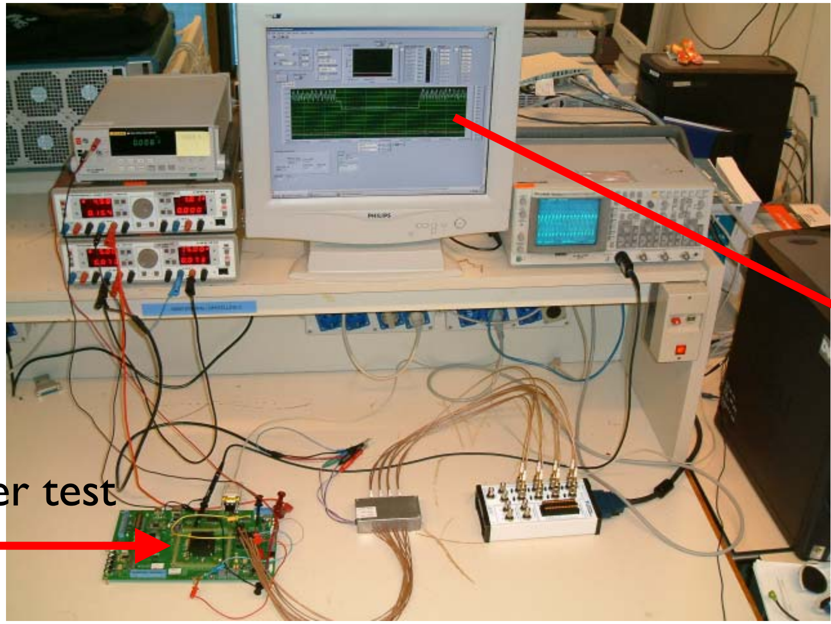
All of them are controllable during runtime!



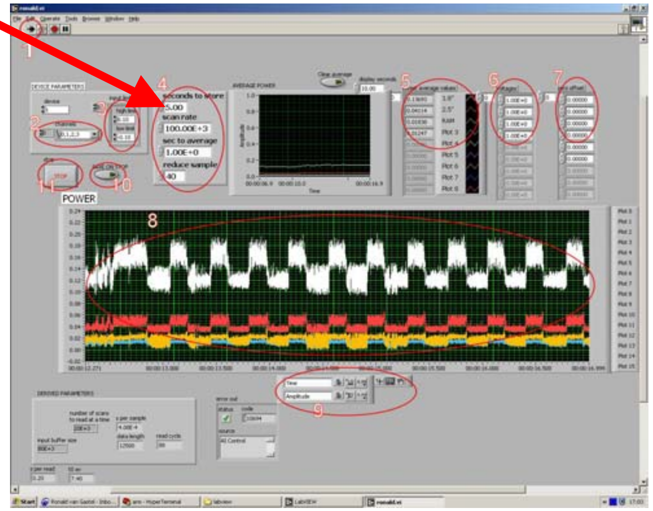
PHILIPS

Linux and prototyping: Dynamic Power Measurements

A board under test



LabView

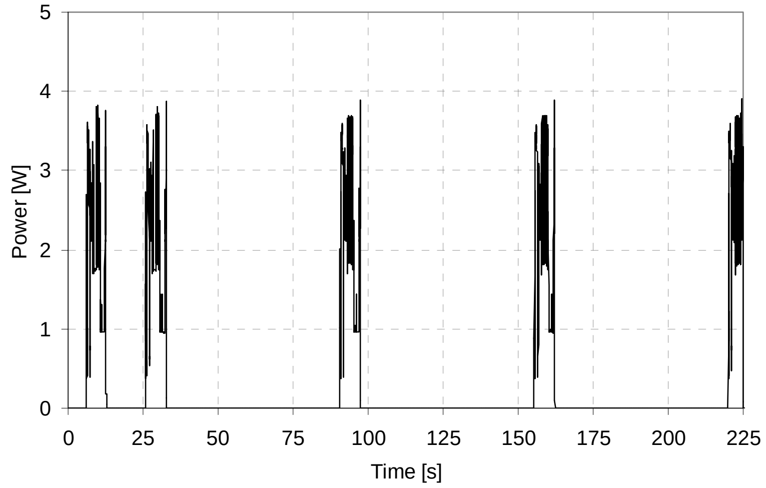


Actual real-world use-cases could be measured!

e.g. MP3 playback, effects on powerusage in various scenario's

Linux and prototyping: ABISS
Low Power HDD Scheduler

- **A**ctive **B**lock **I**/O **S**cheduling **S**ystem (ABISS)
  - The algorithm was developed by Philips Research.
  - It is a scheduling framework in kernel space.
  - Targeted at low latency and low power streaming.
  - Allows multiple streams; both reading and writing.
  - Simple and non-intrusive API
  - Obsoletes buffering in streaming applications and adds low power properties.



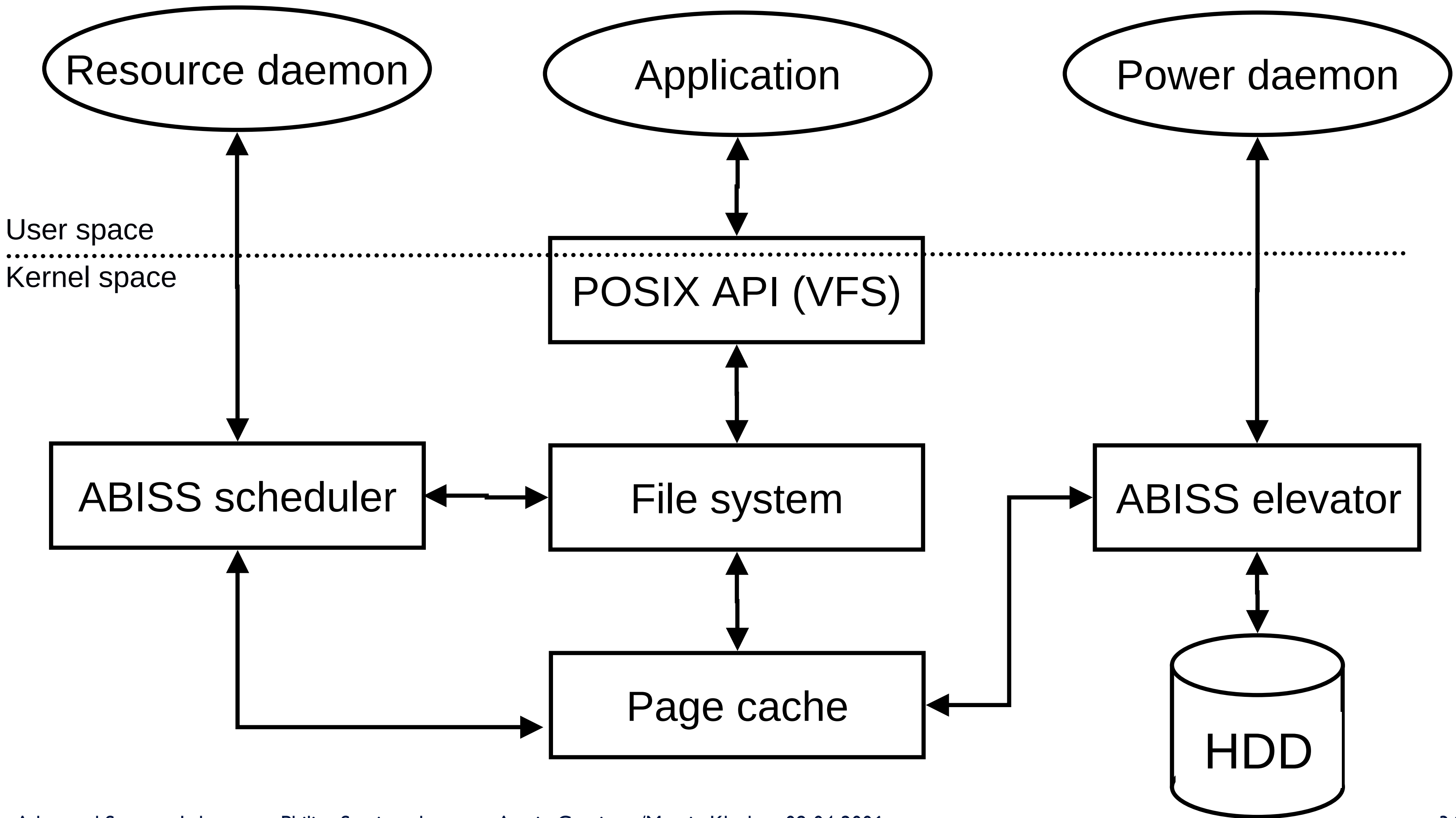
Result:

~200mW HDD down from ~1.7W for a 2Mbit/s stream

PHILIPS

Linux and prototyping: ABISS

VCP architecture



Linux and prototyping: Next steps

- Extending the Audio/Video capabilities of the PNX0106 with an external video co-processor
  - e.g. PNX0106 + PNX4103
  - Based on ARM926 and Philips TriMedia™ architecture

- Linux plays again an important role!

- More details see **CELF 2007** …

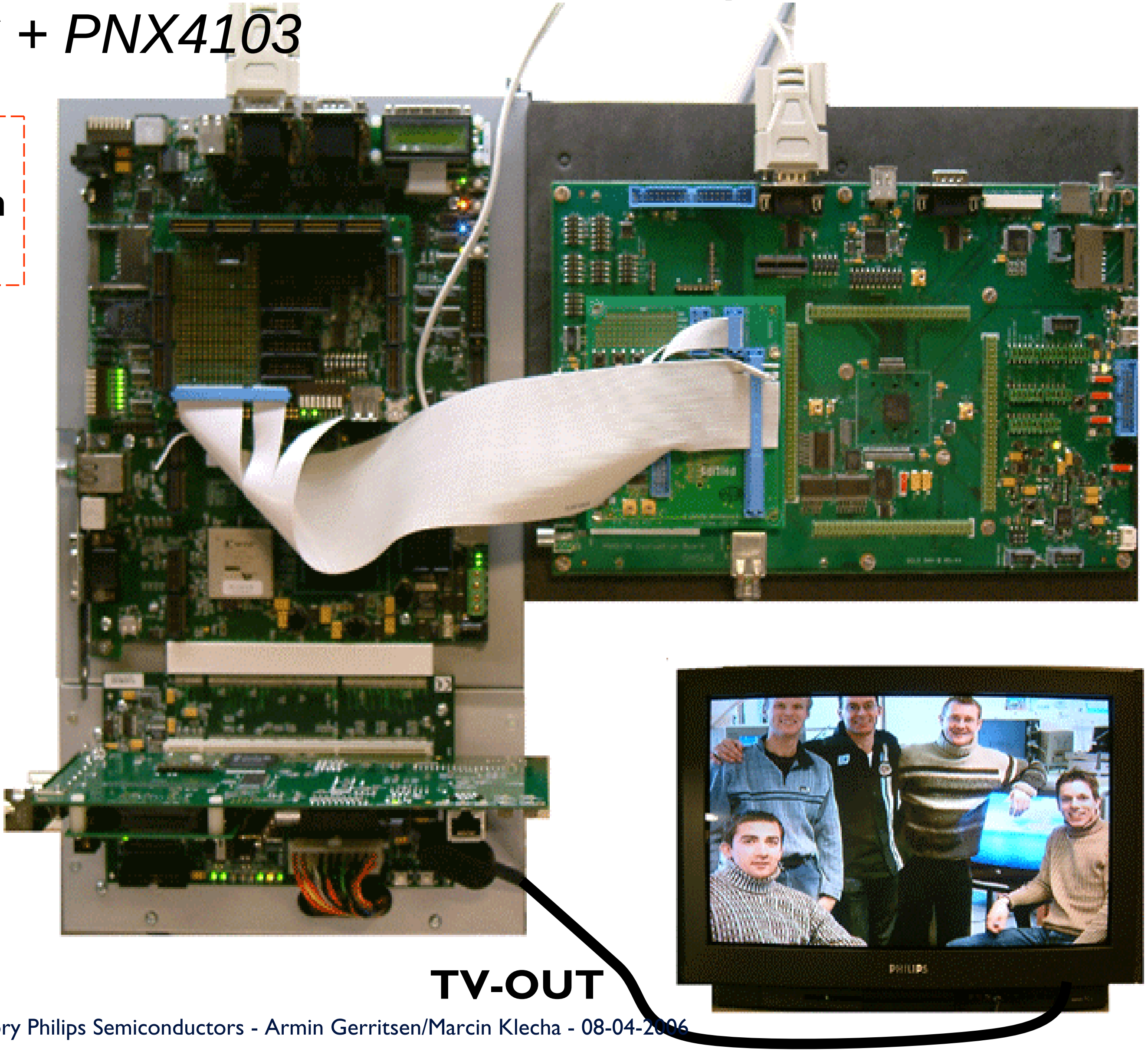
Linux and prototyping: Next steps

PNX0106 + PNX4103

PNX4103 Emulation Platform

ARM Versatile Platform

LCP1500 (PCI)



TV-OUT

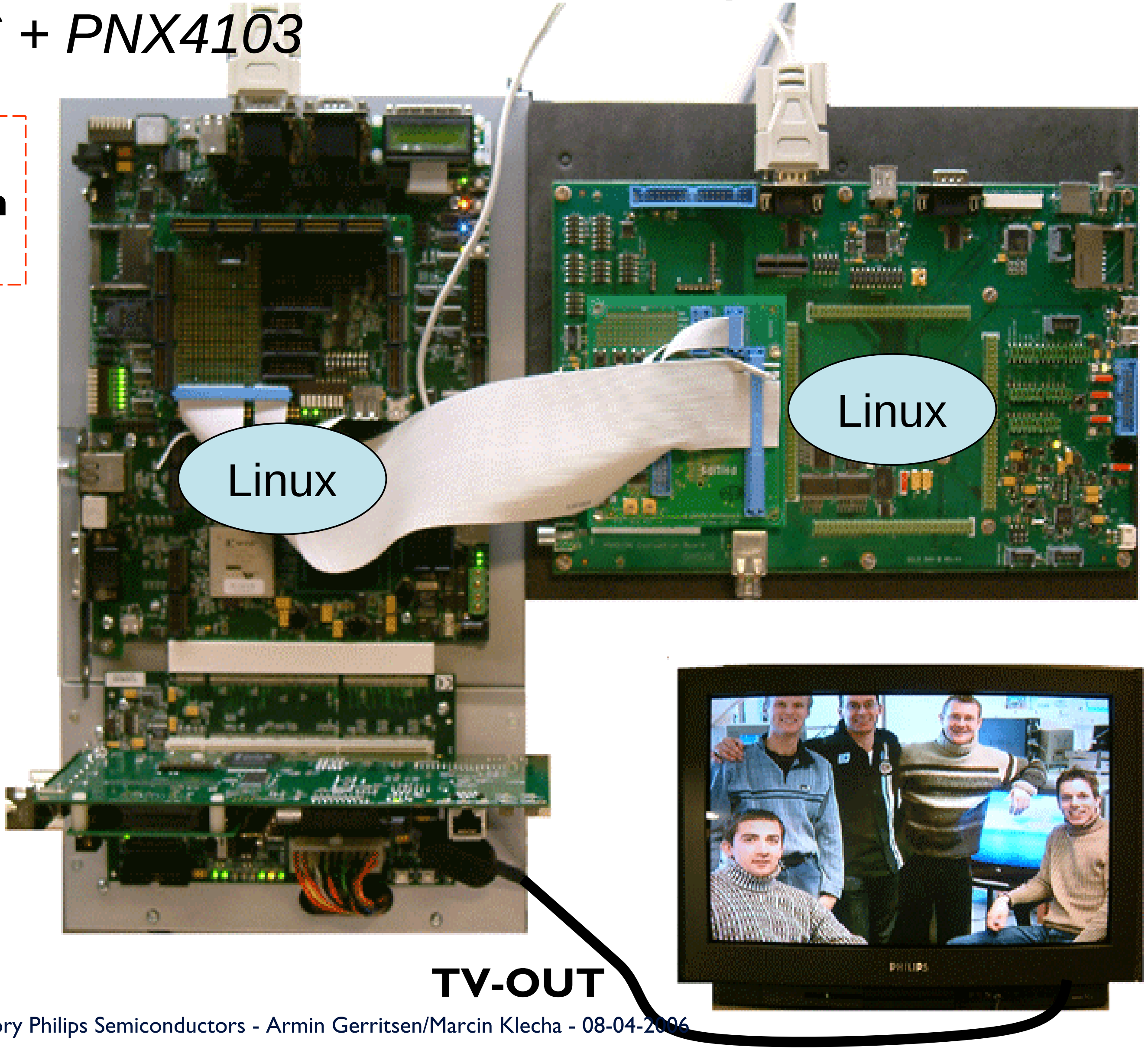
Linux and prototyping: Next steps

PNX0106 + PNX4103

PNX4103 Emulation Platform

ARM Versatile Platform

LCP1500 (PCI)



TV-OUT

PHILIPS