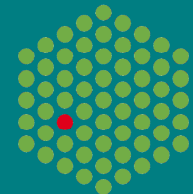


AN EMBEDDED DISTRIBUTED SYSTEM BASED ON TINE & WINDOWS CE

PCAPAC 2008 – Andrés Pazos, Uwe Ristau
(EMBL), Philip Duval (DESY)

EMBL



Outlook

1. Windows CE OS
3. TINE Control System
4. TINE + Windows CE
5. Applications
6. Test System
7. Conclusions

Windows CE



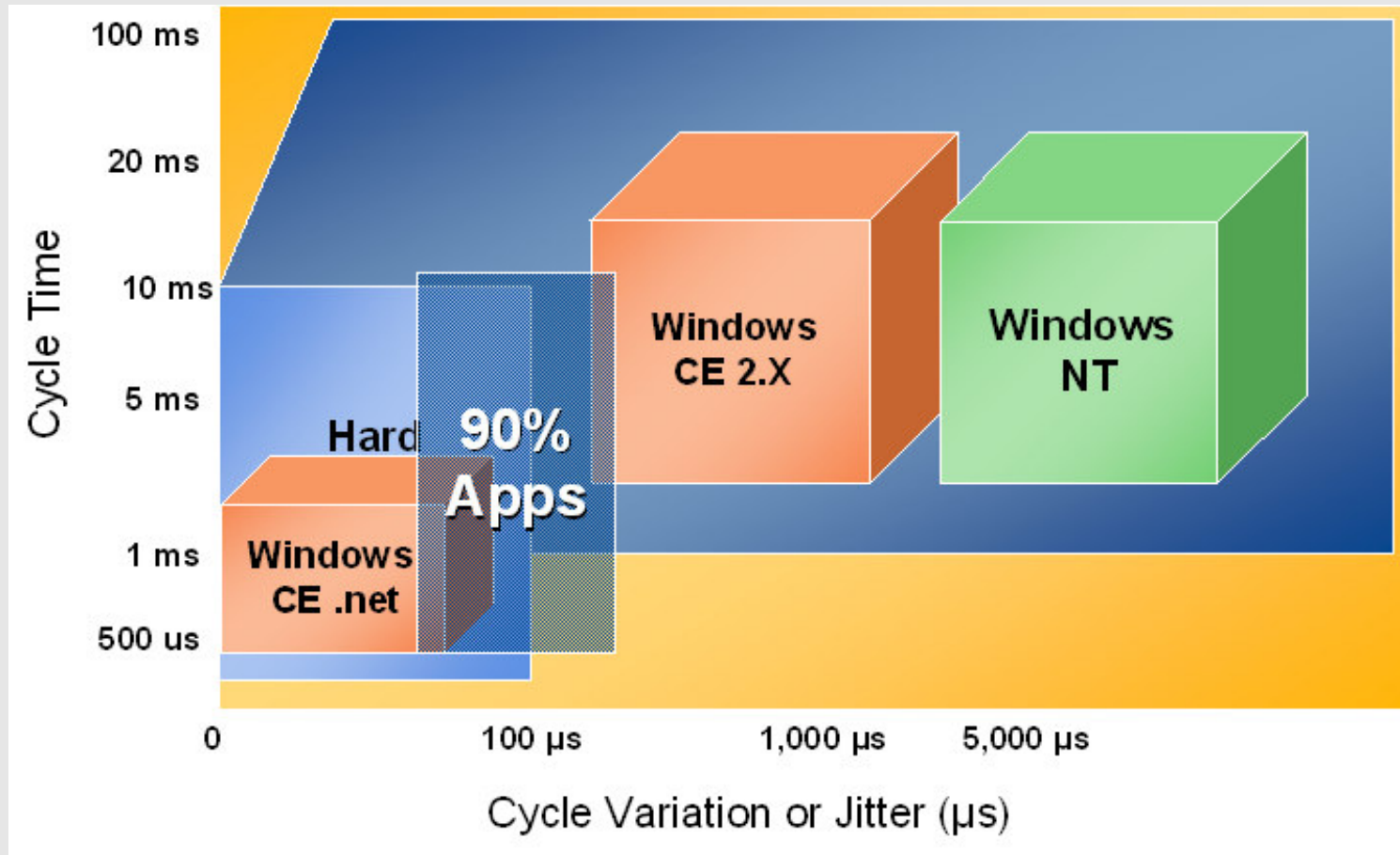
Windows CE OS

- New 32-bit architecture (not Desktop Windows!)
- Multitasking, multithread, scalable
- Component-based architecture
- Possible to recompile the kernel
- Minimum kernel size: 300 Kb
- Open Source code (with the shared source license)
- Actual release Windows CE 6.0 (September 2007)
 - From 32 to 32000 processes
 - From 32MB to 2 GB Virtual Memory
 - Better integration in Visual Studio 2005/2008
- Just to clarify:

Windows CE is an OS and for example Windows Mobile is a platform based on Windows CE and other applications

Windows CE OS

WCE032 – Windows CE Real-time Architecture (WinHEC 2006)



From OMAC (represents Industrial Automation Community)

TINE Control System

...you have already heard about

but why together with Windows CE:

Not dependent on third-party developments

Flexible to incorporate new developments

Very low size

Is made in C, “easy” to recompile for other platform

Source code available: <http://tine.desy.de/downloads>

TINE + Windows CE

- Cross-Compilation of TINE for Windows CE 6.0
- Migration of the code
 - Adaptation to the requirements of the Windows CE libraries
 - Some functions used in TINE were not supported by WinCE (is a thin OS)
- Cross-compilation was straight forward but not trivial
 - Needs time to understand the system and the tools
 - ... sometimes google is your best friend

TINE + Windows CE

- Tools used:
 - You develop from your desktop Windows PC
 - VisualStudio 2005 or evC4 (free download)
 - Windows CE SDK (free download)
 - Windows CE emulator (free download)
- Status
 - Release TINE 4.0 supported
 - Also the CDI (common device interface) has been ported
 - Allows semi-automatic generation of servers to access EtherCAT from WinCE
 - A real system has been tested
 - Used for x86 architecture

Applications

1. Client implementation*

- Development of GUI clients to embedded devices
- Application example: the beamline operator checks from a PDA the status of the beamline

*Like Mr.Takashi Kosuge showed on Monday

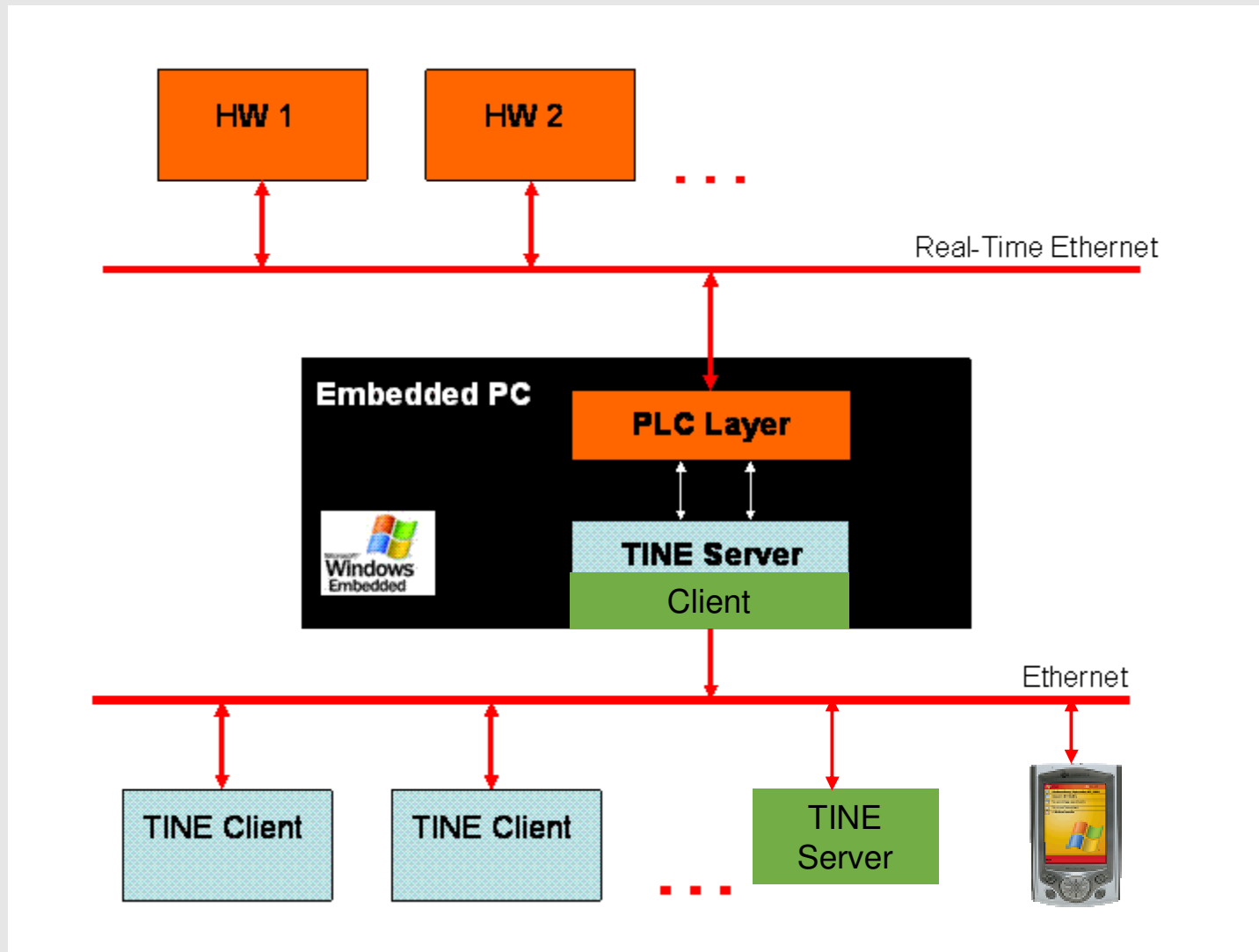
2. Server implementation

- Run low level device servers in Windows CE
- Application example: run your device servers inside a compact embedded-PC

3. Servers (but also clients)

- Run low level device servers
- The servers are clients of other servers running in the control network
- Build your high level software architecture inside the embedded device
- “Network of embedded devices”

Application System



Application Example

- Hardware used:
 - Stepper motors
 - Servo motors
 - Encoder counters
 - Analog IO
 - Digital IO
- Some data:
 - Embedded-PC has a 1,5 Ghz CPU and 1 GB flash memory
 - Low termal dissipation
 - Real-time Ethernet system used is EtherCAT
 - PLC cycle runs at 1 ms, with very low jitter

Conclusions

- Porting is not nice job but results straight forward after knowing the system. Some time needed for understading Windows CE system
- We have made the server independent of the network
- Low-cost final embedded solution
- New programming framework started
- Real time capabilities to explore in practical application
- Windows CE is a growing community, also for industrial automation applications

Acknowledgments

- EMBL Instrumentation group
 - Group Leader: Christoph Hermes
- PETRAIII Instrumentation group
 - Group Leader: Stephan Fiedler

Thank you!!

`andres.pazos@embl-hamburg.de`