Standards & Interop in ICT ETPs
Panel 4 Convergence: The ‘must haves’

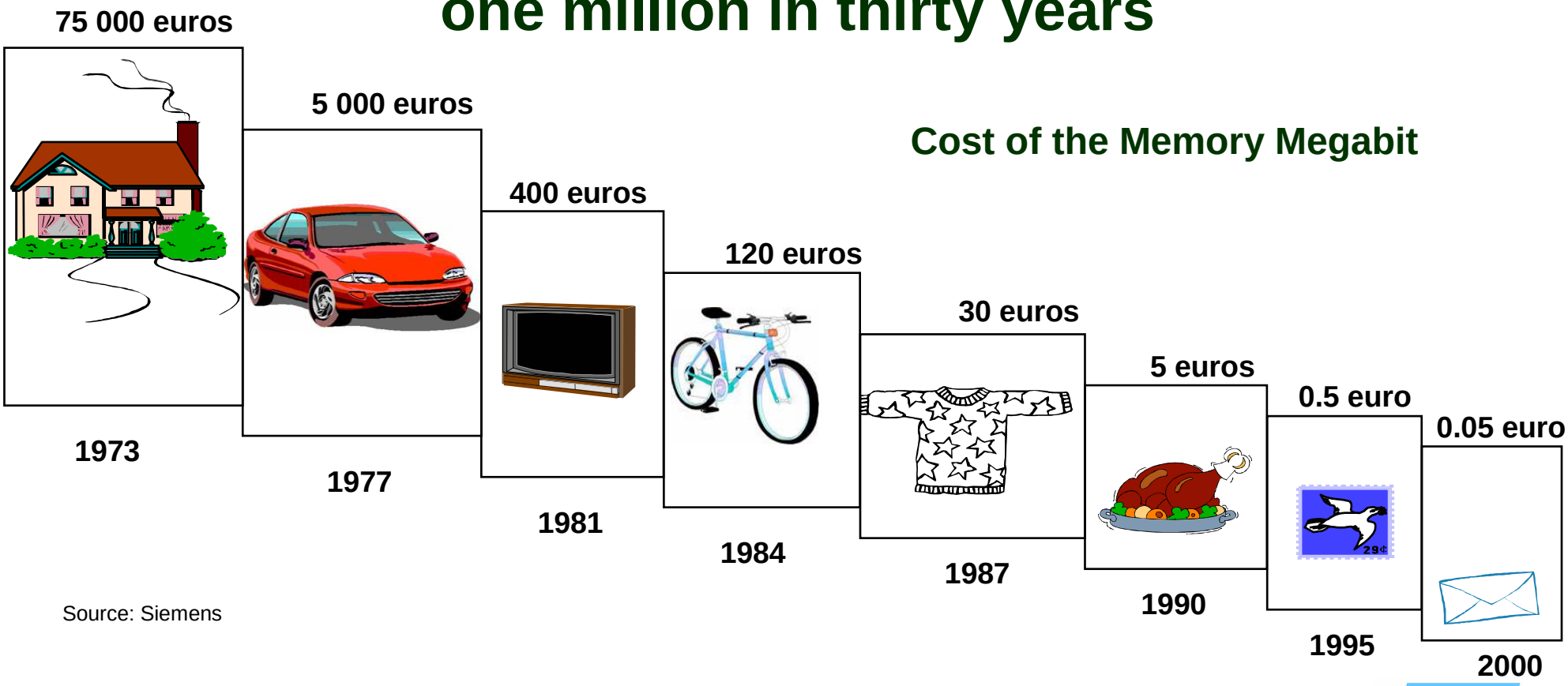
Software interoperability, profiles, building blocks, architectures and middleware

Eric Schutz
VP, External Technology Coordination

STMicroelectronics

Moore's law
Integration doubles every 18 months

The cost of a transistor has been divided by one million in thirty years



Source: Siemens

Complexity : Managing the productivity gap

- Convergence
  - Moore’s law enables the dream of access to people and content
    - anytime, anyplace, with anyone
  - Multimedia everywhere : PC, Home, Office, Car, Plane, Outdoor, …
  - Isolated worlds of connected devices
  - Transistor -> IC -> µP -> SoC : **System = HW + SW**
- Hardware
  - The complexity challenge -> IP reuse, raise abstraction level
  - The power challenge -> multi-processor/multi-core -> additional SW challenge
- Software
  - Proliferation of : target devices, OS, languages, applications, …
  - Challenges : Integration, portability, real-time, reliability, power efficiency, …

Hardware : Open SoC Design Platform

- SPRINT : Standards for interoperable and reusable IP
  - IP-tool interoperability, e.g. SPIRIT API
  - IP-IP interoperability, e.g. VSI, SPIRIT, SystemC,
  - Increase abstraction level to address complexity gap
    - From RTL (Register Transfer Level) to TLM (Transaction Level Modeling)
- Benefits of the approach
  - Cheaper and faster SoC integration
    - IP reuse, automation of IP integration and SoC verification
  - Stimulation of SoC design ecosystem
    - Effective use of innovation capacity, increase of innovation speed
    - Low entry barrier and sizeable market for SMEs
  - Enabler to focus on innovation and differentiation
    - Share costs & risks of developing non-differentiating IP
    - Avoid duplication in development effort -> « HW Open Source »

Software : Open Platform

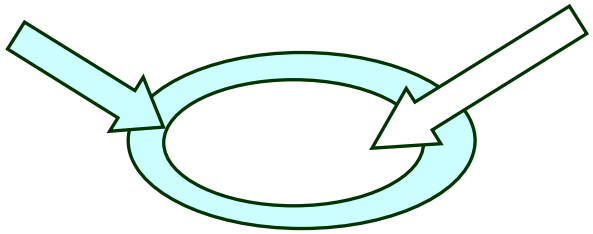
- At System-level: **Components**
  - Enable faster System Integration
  - Must Have:
    - Programming Model for Heterogeneous Multi-Processing
    - Good granularity

CORBA

OpenMAX

Nomadik

Kernel



- At Processor-level: **CLI**
  (Common Language Infrastructure)
  - Goal: Unified tools framework
  - Must Have:
    - Avoid tools fragmentation without loosing performance
    - Portability at no performance penalty

DotGNU Portable.NET

Microsoft .net

ISO International Organization for Standardization

ISO/IEC 23271

- "Pure" Software Entity
  - Enable fast system integration
    - Expectable results through component programming model and run-time support
    - Portable and efficient through advanced CLI compilation technology

Software : Focus on innovation and differentiation

- Create software infrastructures that allow “pure software” differentation
  - Open CLI standards to enable innovation
    - Increased coverage of language
    - Effective interoperability between components using different languages
  - Share the costs & risks of developing non-differentiating SW
    - Through Component reuse
- The importance of Open source for innovation
  - Open Source is a promising answer to foster innovation
    - Open community that generates good quality results
      - Components and Tools:
    - Structured community (ECLIPSE, GNU(*) Portable.Net, ObjectWeb, …)
      - Provide SW debugging, Traceability and Roadmap

(*)GNU: GNU not Unix