GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Towards a Trusted Computing Base in Industrial Systems

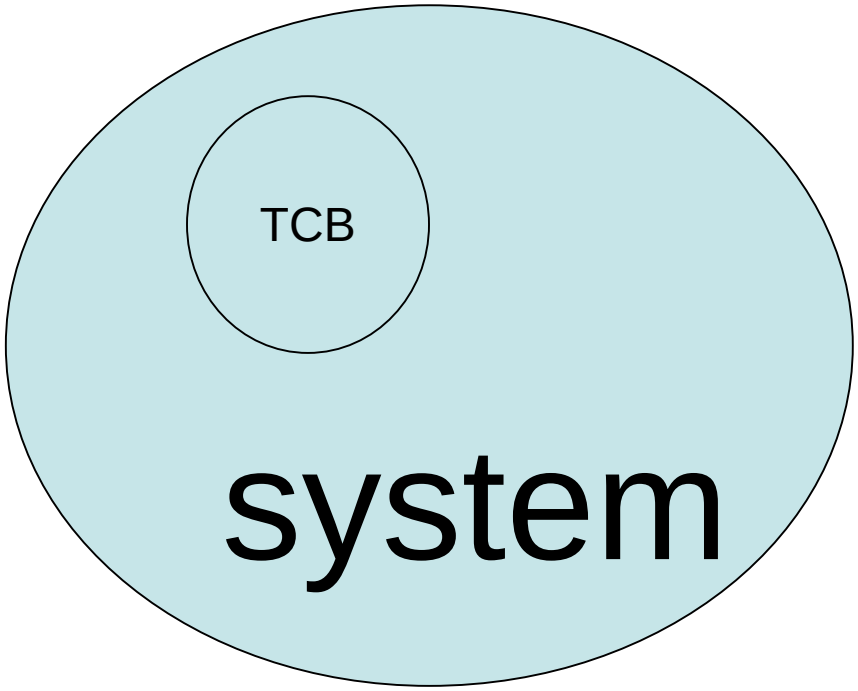
W. Paul
Universität Saarbrücken and DFKI
wiss. Gesamtprojektleiter
bmb+f Projekt Verisoft

Verisoft

Motivation: in the near future...

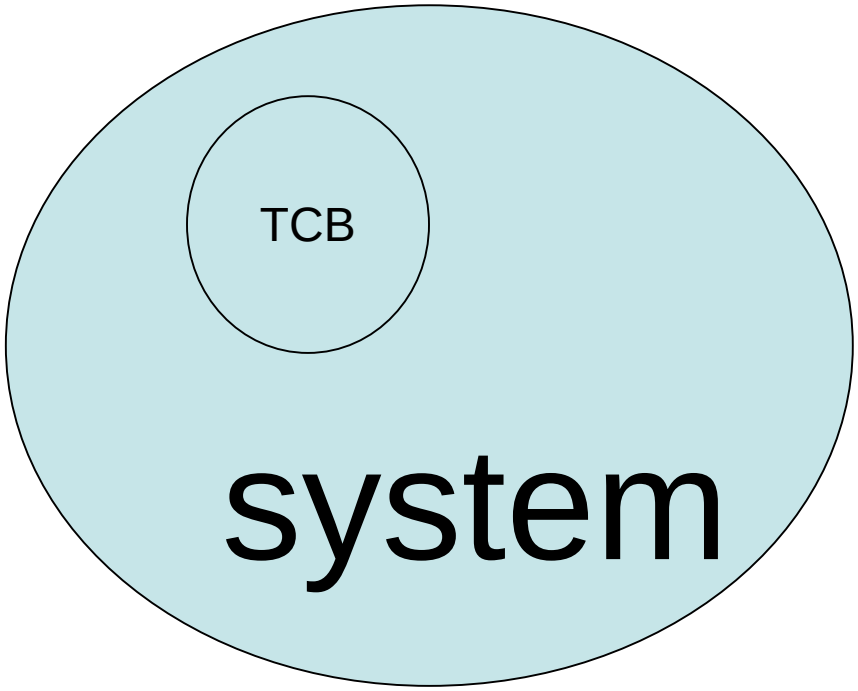
- cars and fighter airplanes will form ad hoc networks
  - error in authentication ....
  - ...spurious maneuvers
  - ...hacking the OS
  - ...taking over part of vehicle with laptop
- on home PC want
  - comfort of commodity OS
  - security of military system

Trusted Computing Base (TCB)



- tiny portion of large system
- makes critical operations of large system secure

Trusted Computing Base (TCB)



- tiny portion of large system
- makes critical operations of large system secure
- for this talk...

  ‚trusted‘ = no design errors

Testing System Design

- how do you know that zero needles are left ?

Measure it !

- how do you know that zero needles are left ?
- Formal Verification
  - correctness proof
  - measure number of gaps in proof by CAV system
  - syntax check

Measure it !

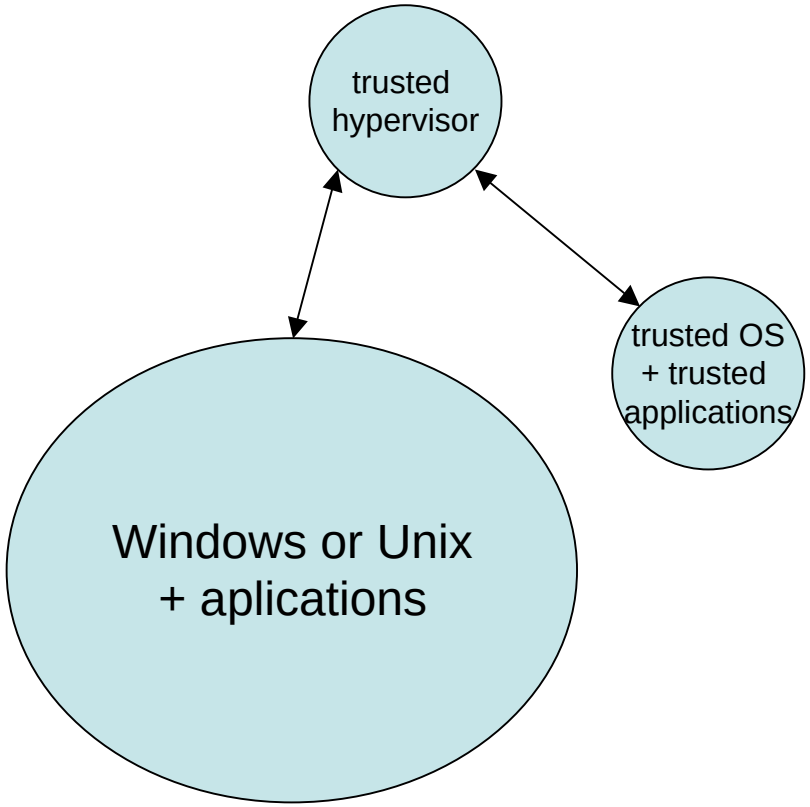
- how do you know that zero needles are left ?
- Formal Verification
  - correctness proof
  - measure number of gaps in proof by CAV system
  - syntax check
- CAV system = measuring instrument
  - nothing less
  - nothing more

TCB: ad hoc networks in vehicles

- ECU
  - microcontroller
  - I/O devices
  - bus interface
- Real Time OS kernel
  - process separation
  - inter process communication
  - device I/O
  - worst case execution time analysis (WCET)
- Application
  - Authentication

TCB: Home PC

- hypervisor
  - like kernel
  - users are operating systems and their user processes (partitions)
  - on high end microprocessors
- small OS
- critical applications



Key Manoevers for TCB (1): Consider Many Computational Models Together

- 1 Processor + I/O Devices
  - a distributed system
- OS Kernel Semantics
  - user processes: assembler machines
  - effect of kernel calls:C semantics
- In-Line Assembler Semantics
  - necessary for kernel implementation
  - C computation
  - assembler computation
  - related by allocation function of compiler
- Real Time Bus System
  - digital hardware model
  - multiple clock domains
  - at borders of domains: set up and hold times, metastability
- Worst Case Execution Time
  - assembler language
  - processor architecture at RTL level

Key Manoevers for TCB (1): Consider Many Computational Models Together

- 1 Processor + I/O Devices
  - a distributed system
- OS Kernel Semantics
  - user processes: assembler machines
  - effect of kernel calls:C semantics
- In-Line Assembler Semantics
  - necessary for kernel implementation
  - C computation
  - assembler computation
  - related by allocation function of compiler
- Real Time Bus System
  - digital hardware model
  - multiple clock domains
  - at borders of domains: set up and hold times, metastability
- Worst Case Execution Time
  - assembler language
  - processor architecture at RTL level
- ONE unified mathematical theory of system correctness!

Pure WCET above RTL level of processor

- is either by measurements
  - guarantees usually nothing
- or
  - like guaranteeing a speed of at least 4.07 km/h for this car
- because:
  - cache penalties can affect execution time of an ISA intruction by factor 100

Key Manoevers for TCB (2)
System Verification Environment

- System Verification Environment/Repository
  - development tools
  - formal proof tools
  - equivalence proofs for semantics
  - formal models
  - verified components (processor, compiler, kernel, OS,libraries)
- supporting collaborative development of proofs in unified theory

Key Manoevers for TCB (3 & 4)
Proof Tools

- Combination of Automatic and Interactive Provers
  - you never get stuck,
  - since Nov 2006: realistic benchmarks for automatic software verification tools with known proofs.
  - degree of automation quickly rising (now 30%-80%)
  - degree of automation in hardware verification:

    > 95 % (OneSpin Solutions)
- Semantics Hierarchy
  - Hoare Logic
    - high productivity
    - no interleaving of computations
  - Small Step Semantics
    - poor productivity
    - interleaving
  - ‚equivalence' theorems
  - high productivity with interleaving
- This works with devices

Verified Components (Jan 07)

- Formal Ver. Completed
  - Microcontroller with devices
  - C compiler (Pascal like subset)
  - Communication Protocols
  - Crypto Primitives
  - Signature Server
  - Disk Driver; uses in-line assembler
  - Theory of Linking
  - .....
- Putting Big Proofs Together
  - page fault handler
  - paging kernel
  - OSEKTime like real time kernel
  - FlexRay-like bus interface
  - ......
- use everything above

Possible Future Research Projects (2007-2010)

- Best effort of Complete Formal Verification of Industrial
  - ECU (automotive)
  - real time OS kernel (avionics)
  - biometry and authentication on chip cards (telecommunications)
  - hypervisor (commdodity software)
- Get degree of automation above 90 % in software verification

You (should) want this...

- Complete Formal Verification of Industrial
  - ECU (automotive)
  - real time OS kernel (automotive)
  - authentication on ECUs (automotive)
- Trusted Computing Base for ad hoc networks in cars