

COMPUTER ARCHITECTURES



BLACK BOX

ZEN

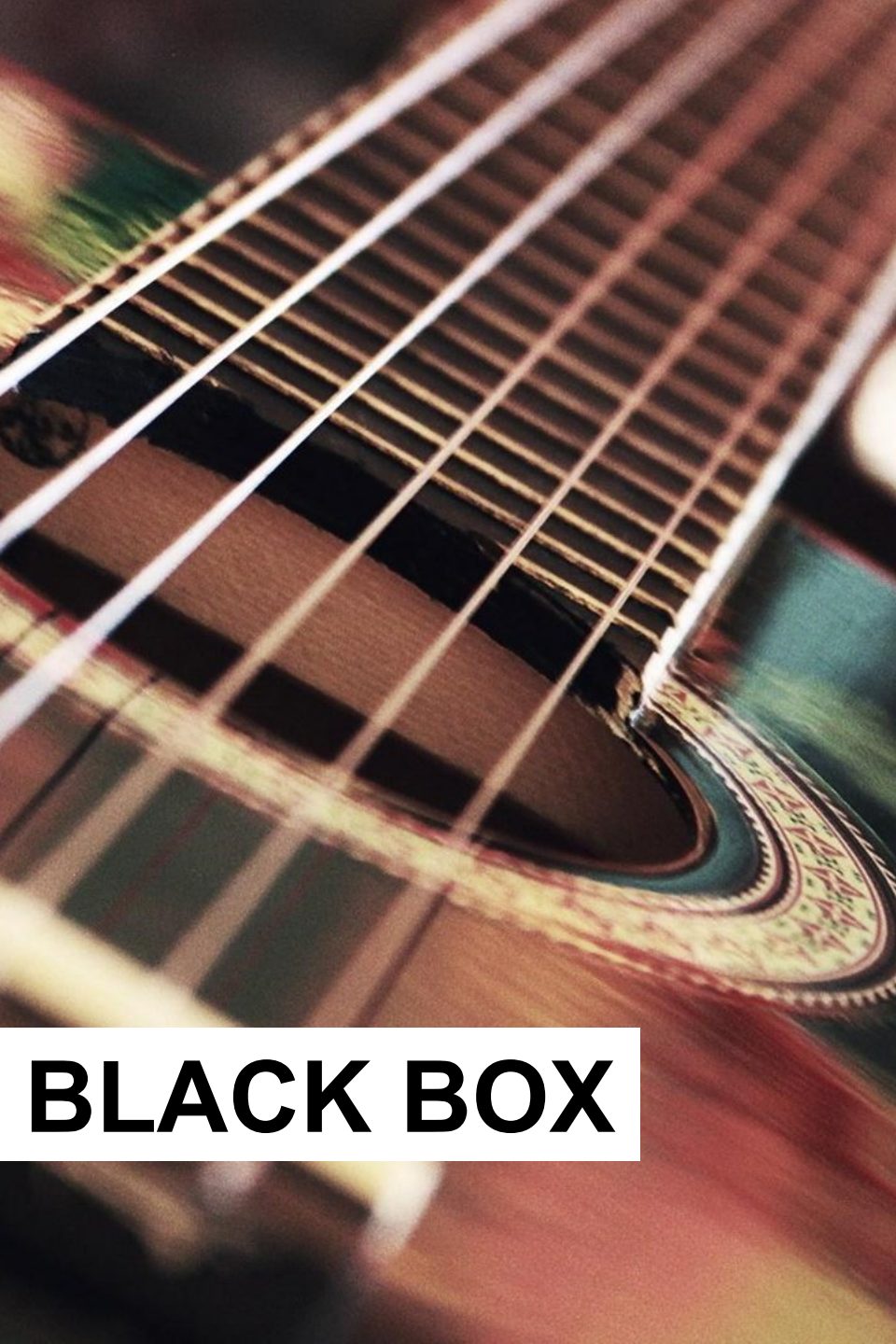


AND THE
ART OF
MOTORCYCLE
MAINTENANCE

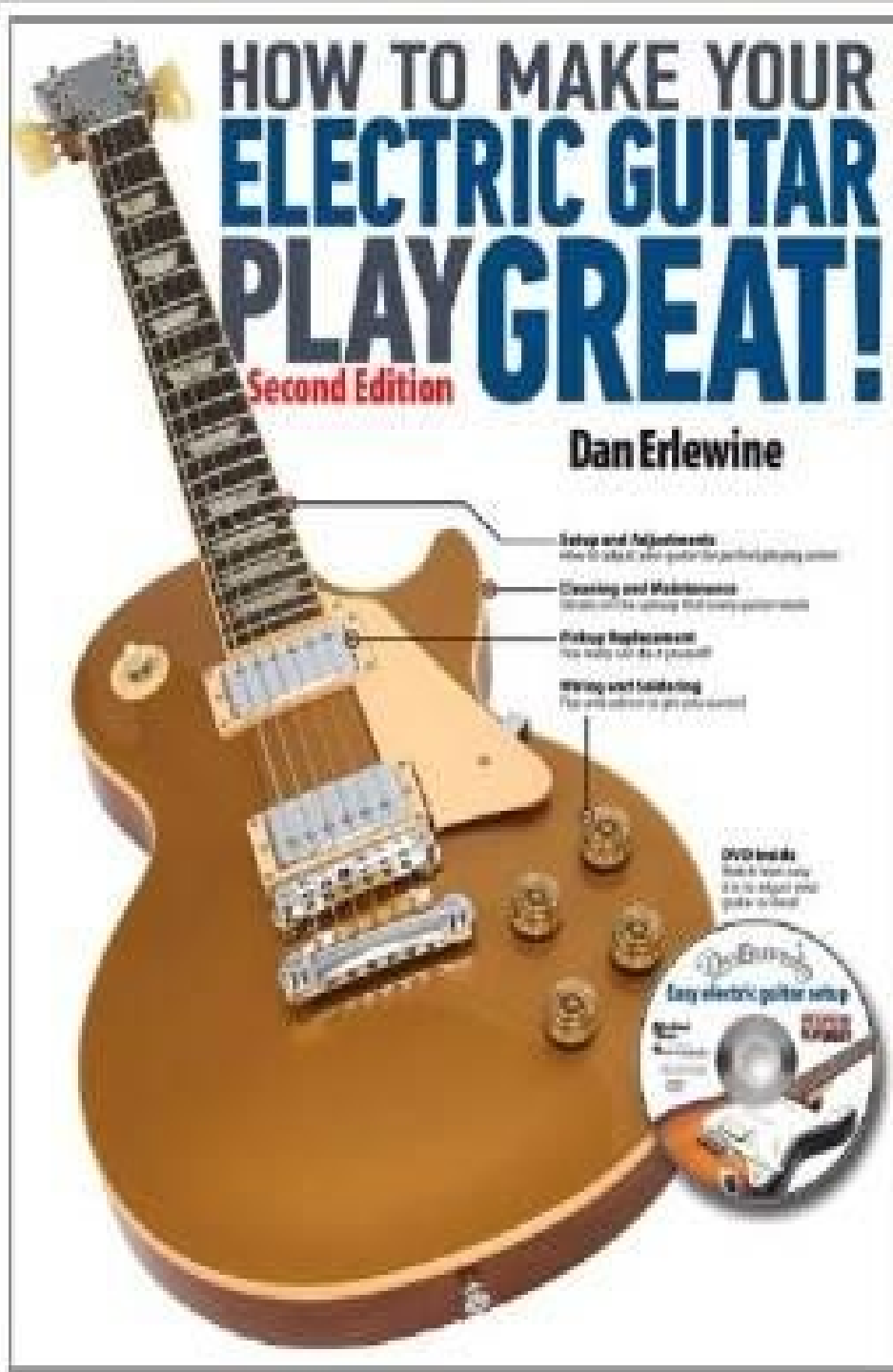
An Inquiry into Values

With a New Introduction by the Author

ROBERT M. PIRSIG



BLACK BOX



COMPILER

EMBEDDED SYSTEMS

OP. SYSTEM

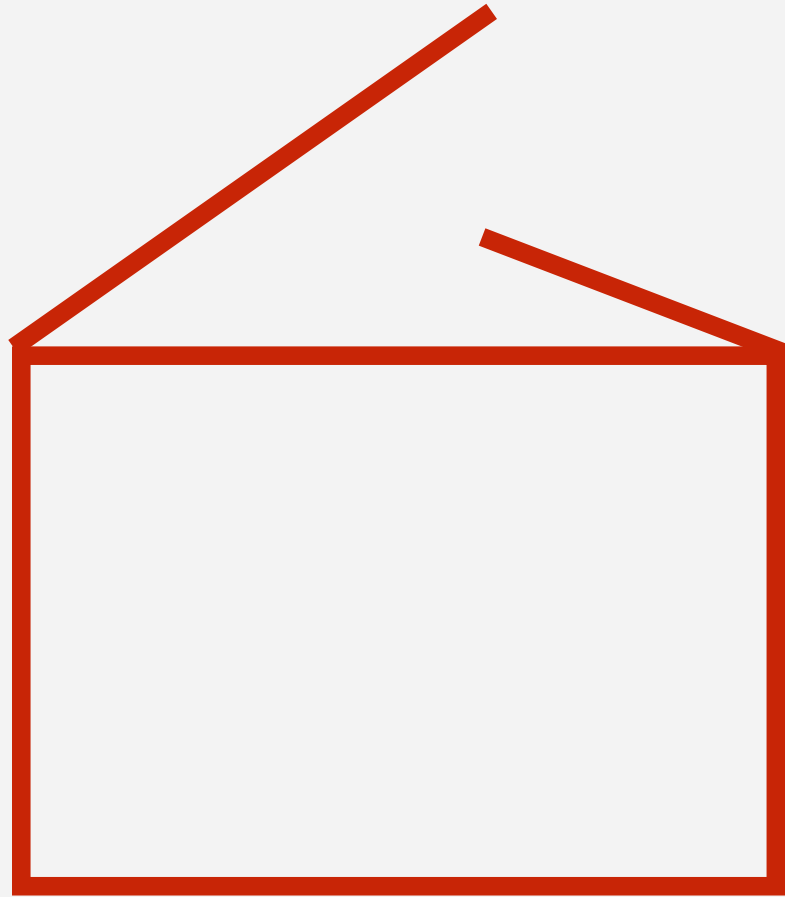
HARDWARE

Black
Boxes...

Black Boxes
Everywhere!

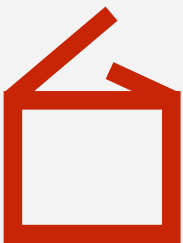


We. Open. Boxes.

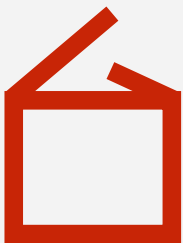


WHY OPENING THE BOXES?

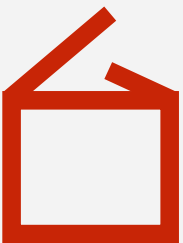
- curiosity
 - craving for knowledge
 - “hacking” philosophy
- we’re engineers!
 - *understanding*: knowing precisely what’s inside the box allows to make it work better, even without changing anything
 - *full stack engineering*: cutting-edge results can be obtained only by opening the box and getting your hands dirty



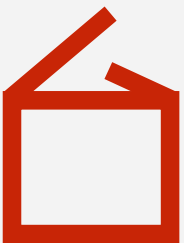
**EMBEDDED
SYSTEMS**



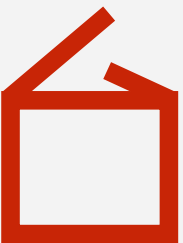
COMPILERS



OP. SYSTEMS



ARCHITECTURE



HARDWARE

CROSS-PROPERTY

POWER EFFICIENCY

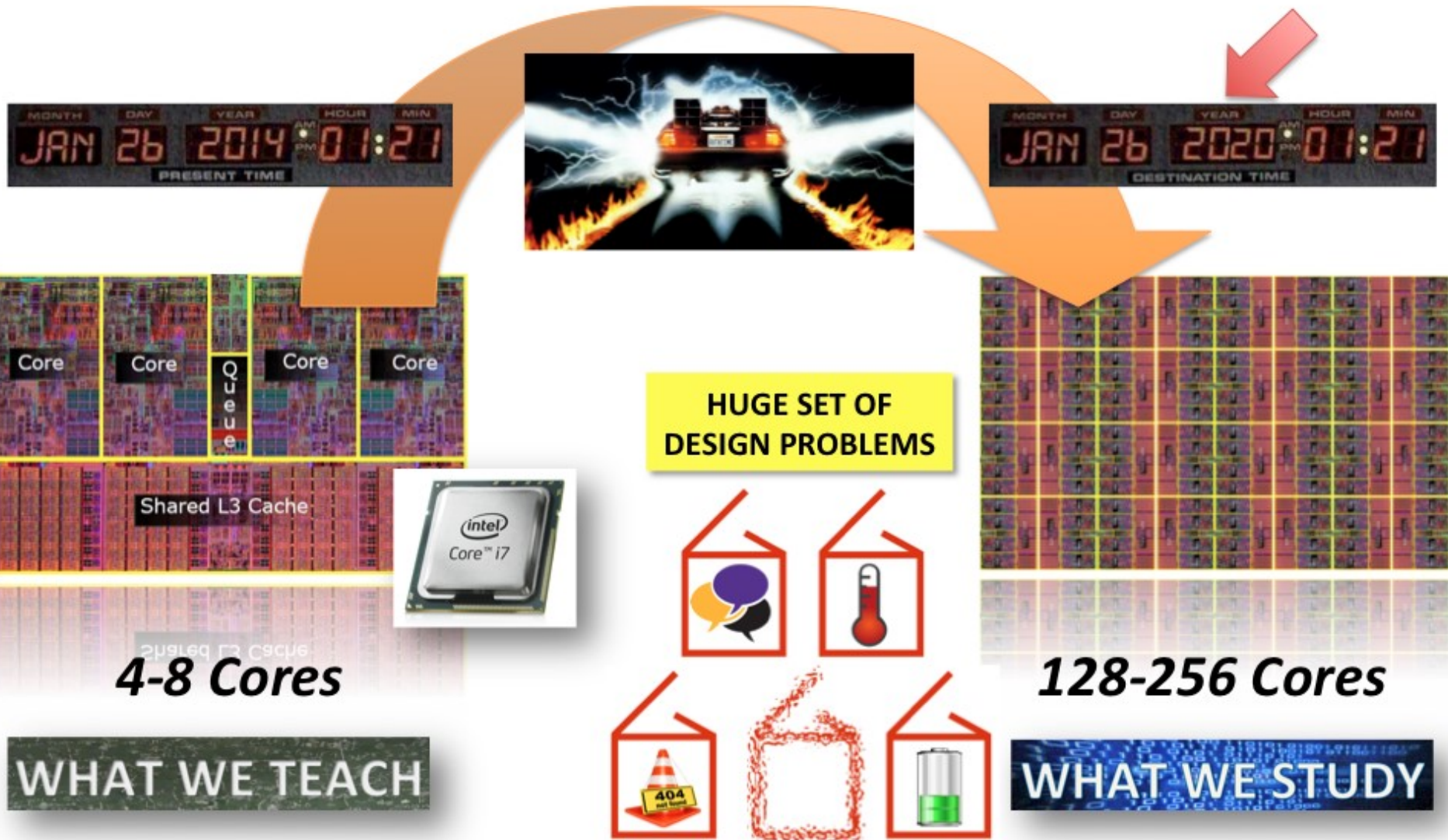
SECURITY

RELIABILITY

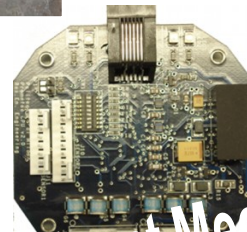
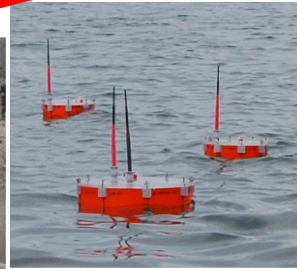
PERFORMANCE

ADAPTIVENESS

#OPENTHEBOX TO DESIGN NEXT GENERATION COMPUTER ARCHITECTURES



#OPENTHEBOX - (INTELLIGENT) EMBEDDED SYSTEMS



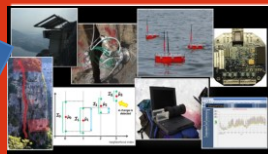
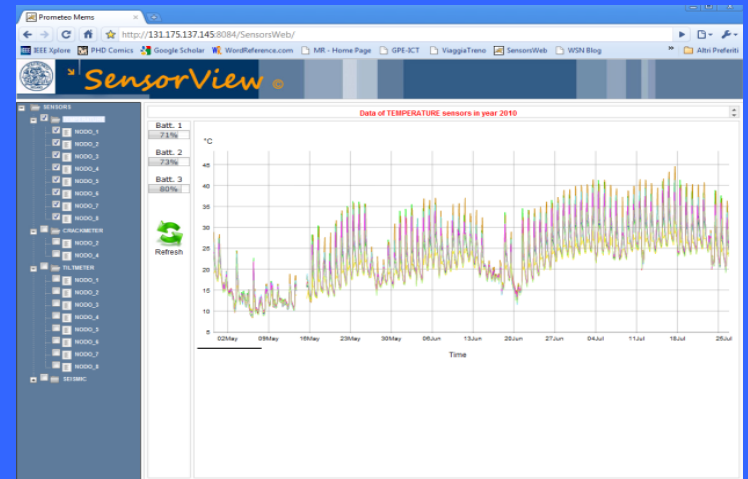
Intelligent Mechanisms

#OPENTHEBOX - INTELLIGENT EMBEDDED SYSTEMS

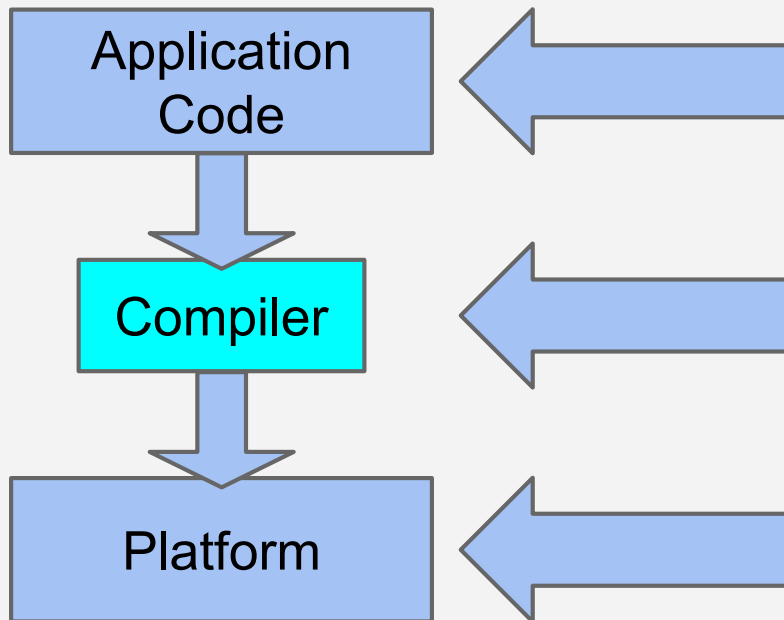
Intelligent embedded systems and multimedia sensor networks



Adaptive intelligent systems in nonstationary environments



COMPILER CONSTRUCTION



Defines:

- Application behavior

Black box approach

- Write program
- Call compiler
- Run executable on any platform

Offer:

- Performance
- Security
- (Low) Energy
- Form factor

Problems: how do we achieve goals (performance, security) if we don't know about the platform?

#openthebox: inside the compiler

- Support for multiple platforms
- Transformations for
 - Performance
 - Security
 - Code size
 - Low Energy

How?

Research:

- *Performance Portability:* how can we squeeze more performance from code, regardless of the platform it is run on?
- *Security:* how can we ensure secret information cannot be extracted, if the hardware does not provide specialized support?



HARDWARE



ARCHITECTURE



OP. SYSTEMS



COMPILERS



**INTELLIGENT
EMBEDDED
SYSTEMS**

POWER EFFICIENCY

SECURITY

RELIABILITY

PERFORMANCE

ADAPTIVENESS

CROSS-AREA

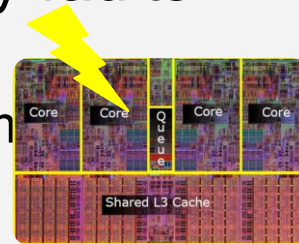
Reliability

Digital systems may be affected by faults.

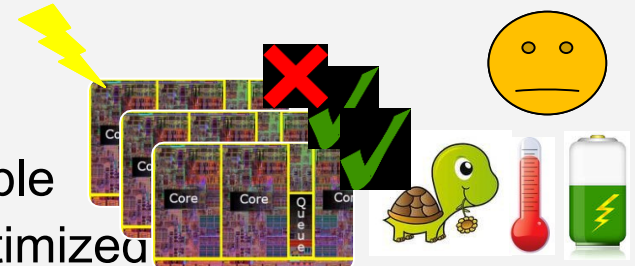
Causes:

external phenomena (radiations, electromagnetics, ...)

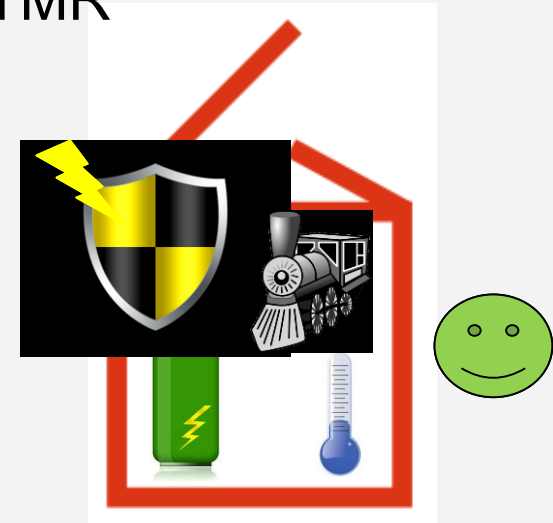
aging of digital systems



- Black-box approach
 - use of the nominal system -> NOT reliable
 - platform-agnostic hardening -> NOT optimized
- #openthebox
 - definition of new hardening strategies specific for the considered architecture/device
 - definition of methods and tools for the optimization of the hardening w.r.t various metrics (reliability, performance, area, power, ...)
 - definition of adaptive approaches for reliability



TMR



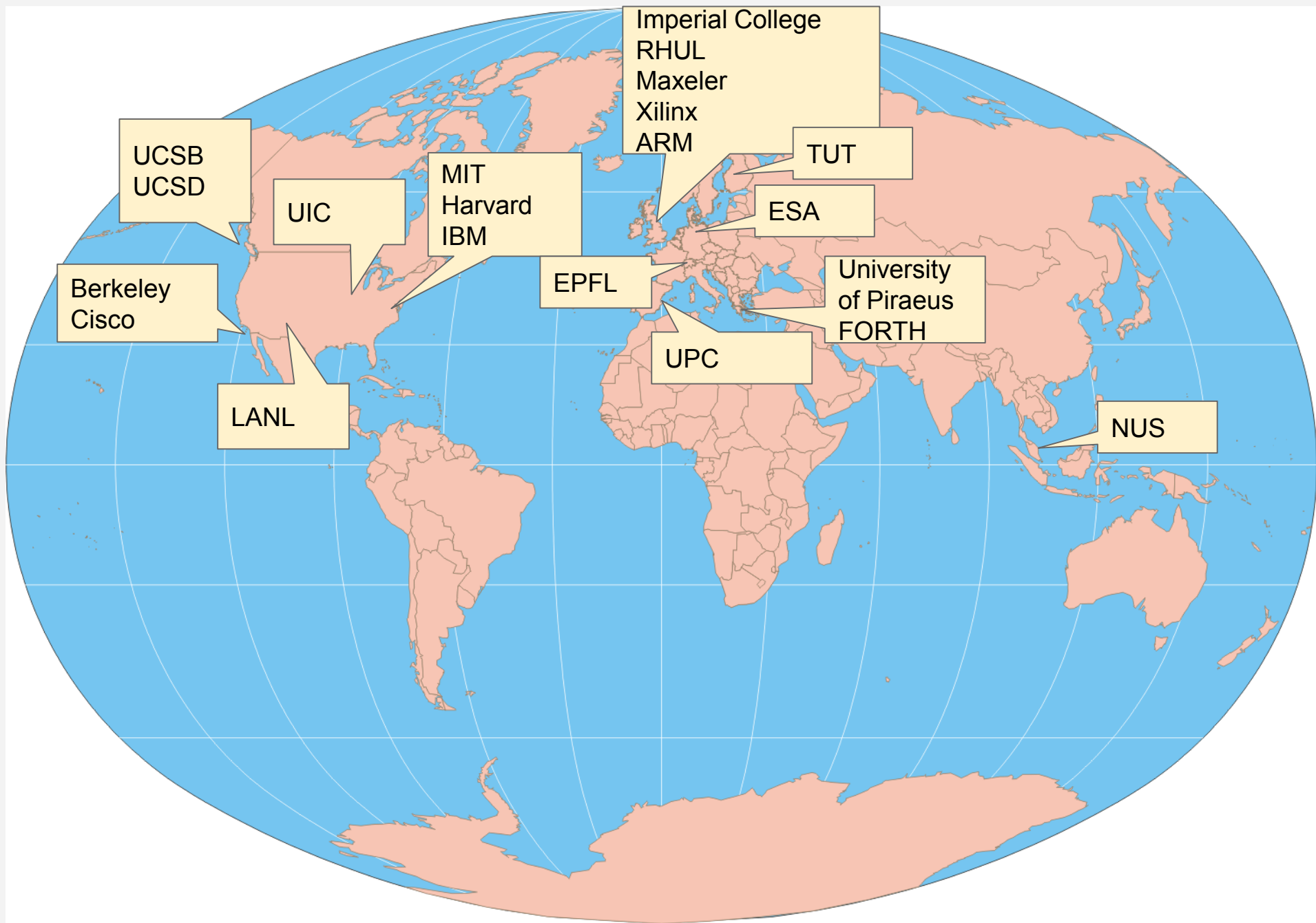
SECURITY

(Some doors should stay shut...)



SECURITY "My crime is that of curiosity."

- Hacker's manifesto, 1986
- Approach: what happens if I change this...?
- Which boxes?
 - web traffic, OS kernel, banking transactions
- Closed boxes leads to mediocre results
 - system security analysis
 - systematic discovery of vulnerabilities
- #openthebox leads to outstanding results!
 - learn from programming bugs to make systems secure
 - malware analysis (dynamic and static)
 - system modeling for anomaly detection






NECST
laboratory



<http://sagroup.ws.dei.polimi.it/>
<http://www.necst.it>

#openthebox

(EXTRA SLIDES)

ADAPTIVE HETEROGENEOUS COMPUTER ARCHITECTURES

- Problem
 - More computational horse-power than ever before
 - Computer architectures, not even in an embedded system, are any longer a single core but..
 - an ecosystems of heterogeneous (adaptable/configurable) cores
 - where it is Impossible to balance all constraints manually
- A problem turned into an opportunity
 - We need novel techniques/ideas to
 - approach the design of such architectures
 - properly manage their runtime
 - ...

Research activities

- Develop novel self-adaptive systems that can observe their runtime behavior, learn, and take actions to meet desired goals
- Design new CAD tools to support the definition of ever increasing complex computing architectures

OPERATING SYSTEM

- Problem
 - Computer complexity is skyrocketing but..
 - Key computer science abstractions have not changed since the 1960's
- A problem turned into an opportunity
 - We need to rethink operating systems from the ground up to

Research activities

- Design and implement techniques for management of heterogeneous computing resources
- Definition and development of autonomic components able to autonomously adapt at runtime
- ...

DESIGN SPACE EXPLORATION

Sizing of the architecture
for a specific application
domain



vs



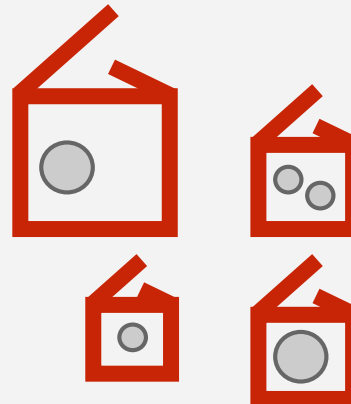
Studying tradeoffs of
parallelism vs
performance and power



vs



Optimal **task mapping**
and **run-time**
management



vs

