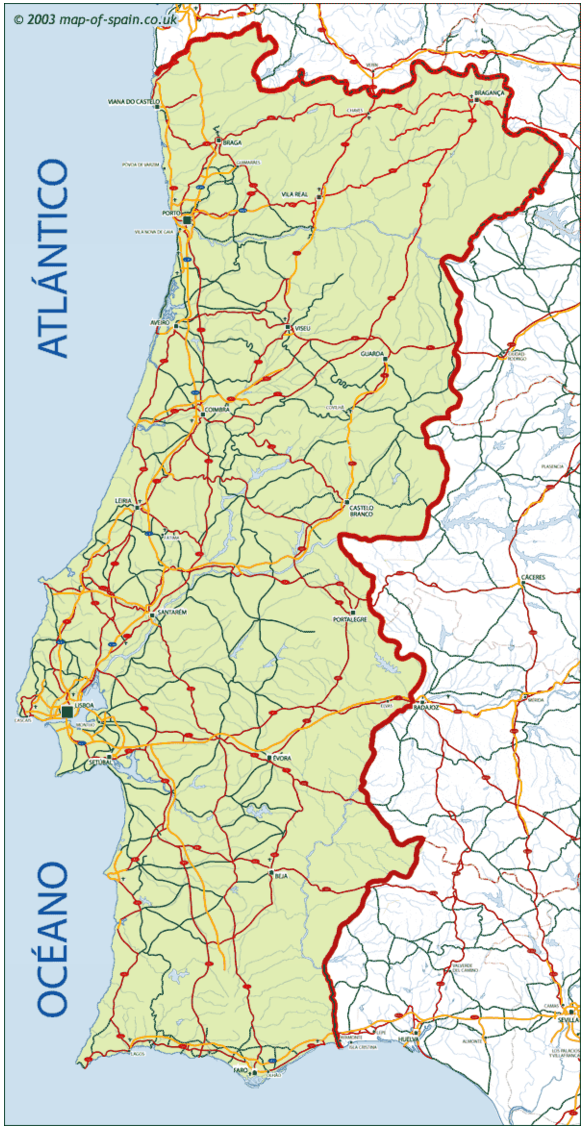
Internet of Things
A Process Calculus Approach

Ivan Lanese
Computer Science Department
University of Bologna/INRIA
Italy

Joint work with
Luca Bedogni and Marco Di Felice
University of Bologna, Italy

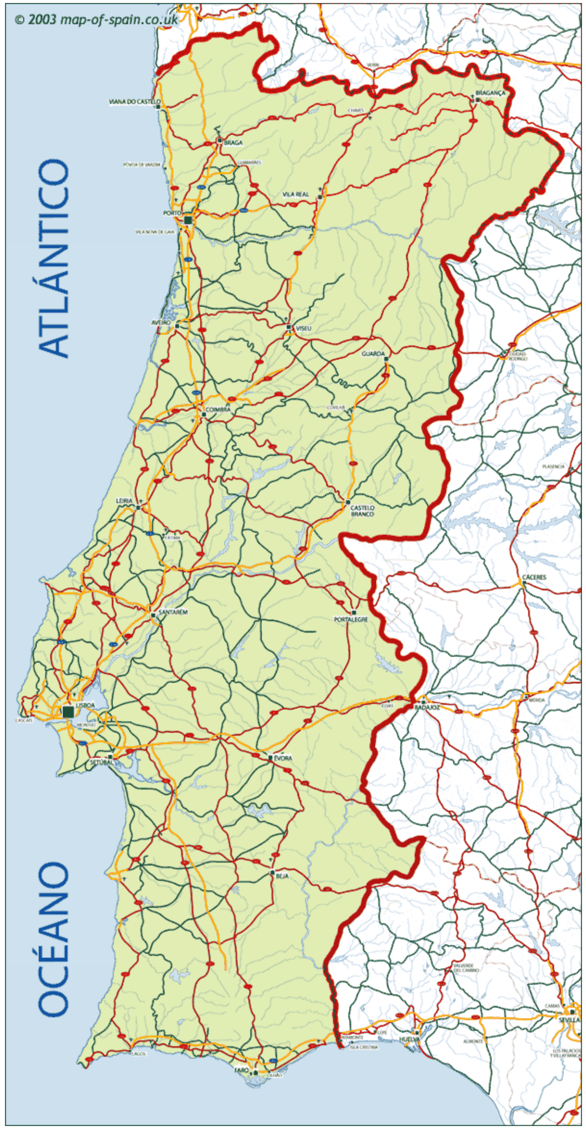
Map of the talk

- Motivation
- Modeling Internet of Things
- System equivalences
- Conclusions



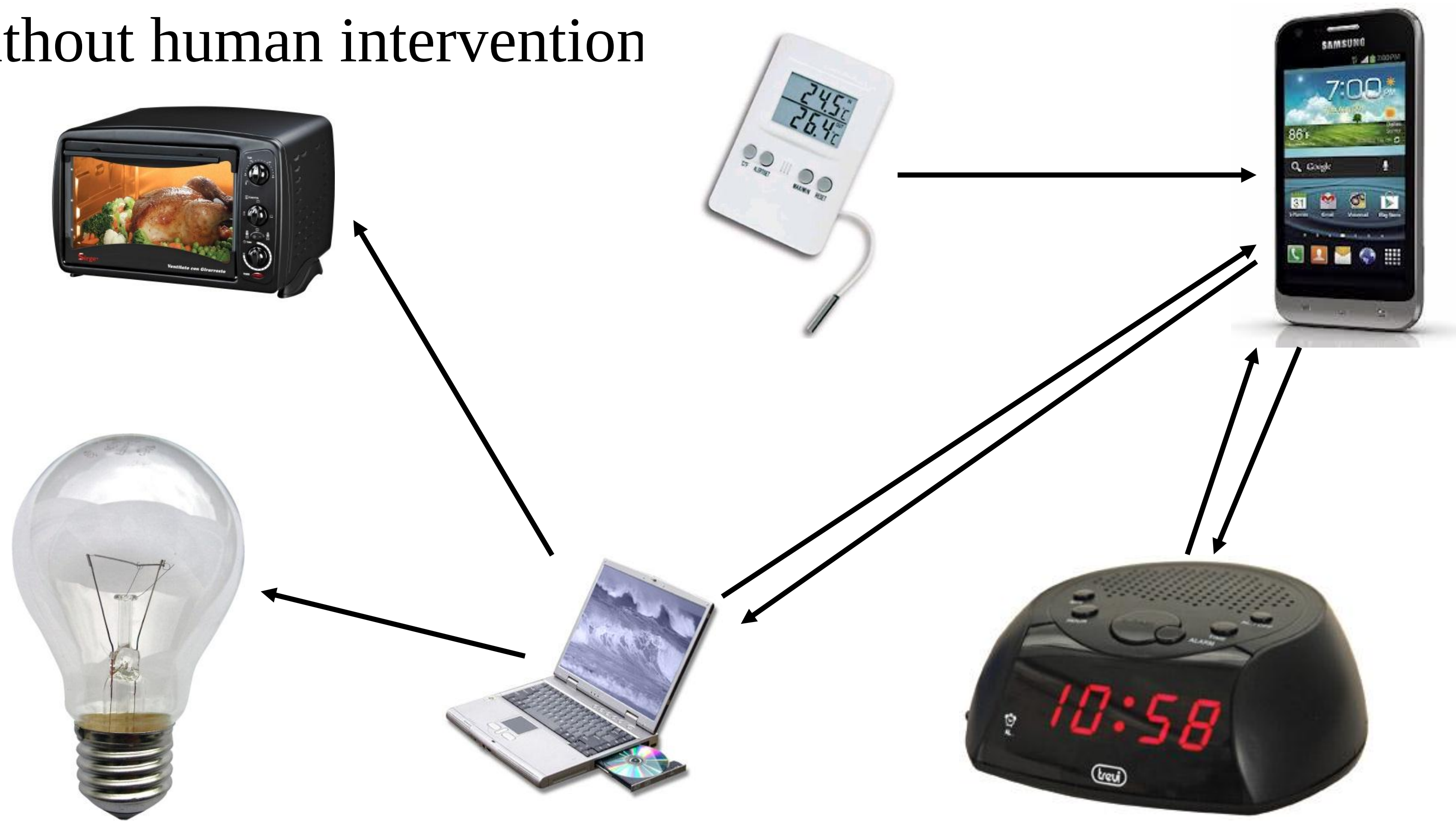
Map of the talk

- Motivation
- Modeling Internet of Things
- System equivalences
- Conclusions



Internet of Things

- An emerging paradigm allowing different things to interact without human intervention



Internet of Things

- What is a thing?
  - Oven, alarm clock, coffee machine, ...
  - Equipped with suitable computing and communication capabilities
  - To provide composite services to human users
- Sample scenario:
  - the alarm clock sets itself according to information on traffic on the road home-work from the net
  - the micro oven turns on automatically 5 minutes later to warm your breakfast

Internet of Things: properties

- No real agreement on what is an Internet of Things, even less on what it is not
- Some relevant features shared by most definitions
  - Distribution
  - Wireless communications (normally short range)
  - Heterogeneity
  - Effects on real world
  - Partial communication
  - Dynamicity
  - Self-healing

Process calculi

- Algebraic abstractions of interacting systems
- Allow to:
  - Clearly specify the behavior of those systems
    - » Allow to prove equivalences
  - Equipped with tools to help reasoning
    - » Behavioral equivalences, type systems
- Provide formal ground to programming languages and analysis tools

General calculi and paradigm-specific calculi

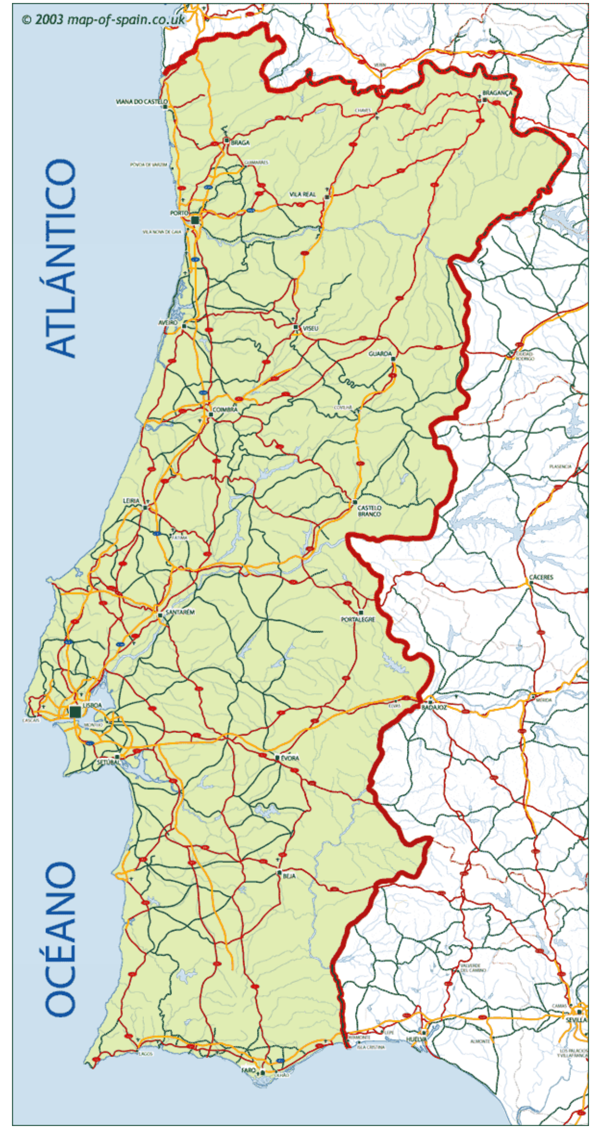
- General purpose calculi: CCS, π-calculus,…
- Calculi targeting a specific paradigm
  - For object-oriented languages, for service-oriented computing, for wireless sensor networks
- We propose a calculus targeting Internet of Things

Why (yet another) calculus?

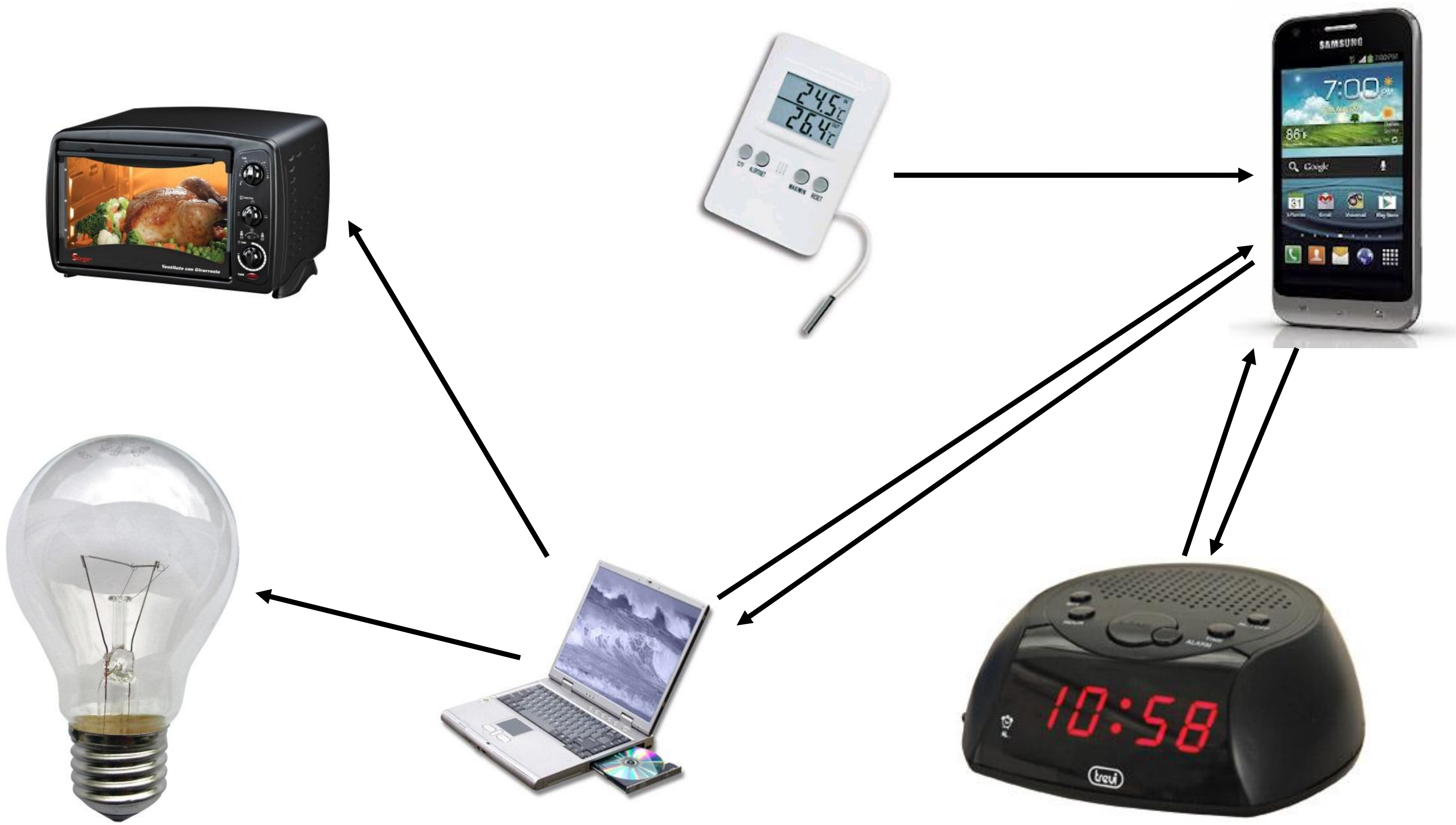
- No calculi for Internet of Things in the literature
- Capture in a clear algebraic setting the peculiar aspects of IoT
- Have the relevant entities as first-class in the calculus
- Needed to speak about them, e.g. when developing a logic
  - All the sensors are reachable from the smartphone T1
  - Difficult to express if sensors and smartphones are not explicitly represented

Map of the talk

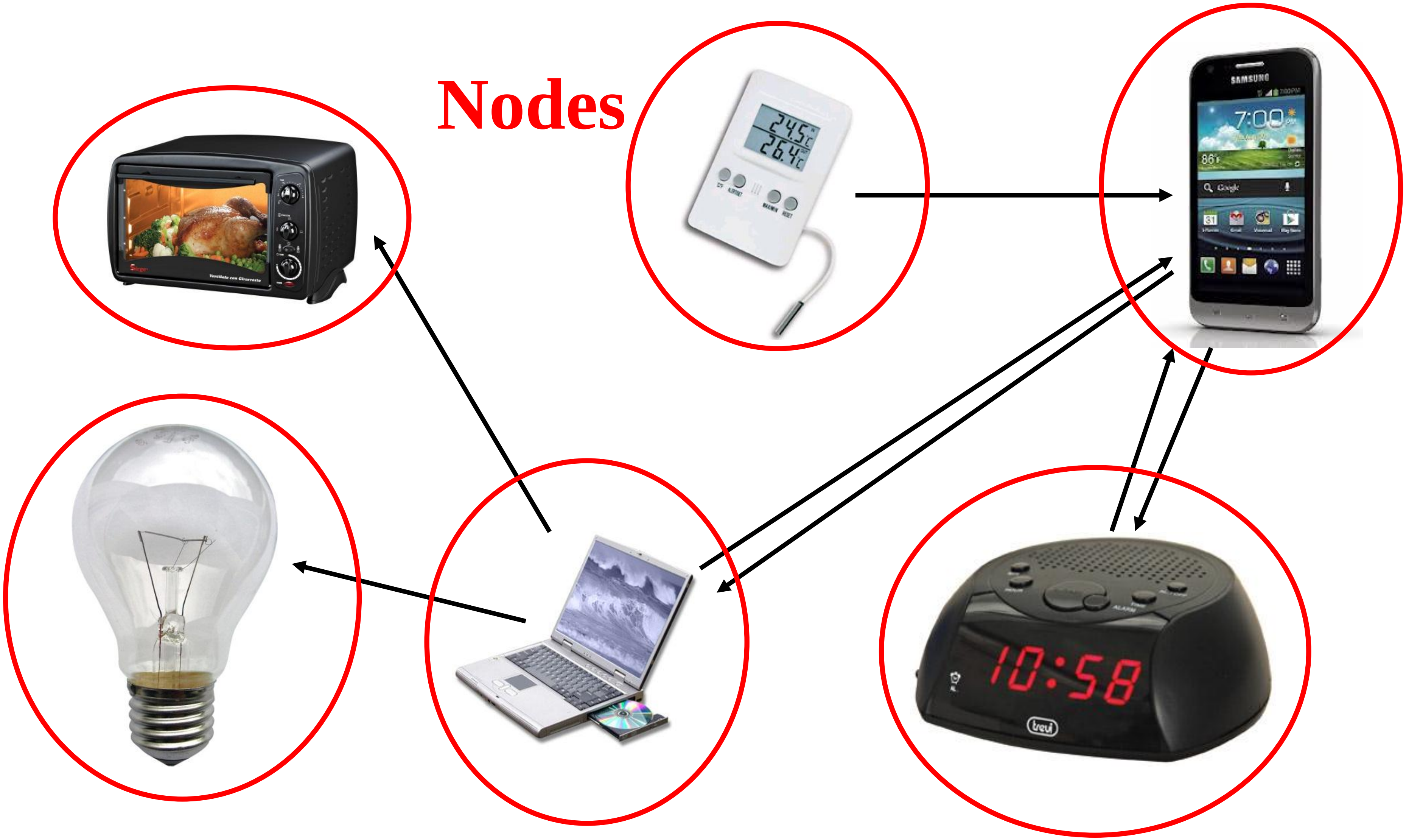
- Motivation
- Modeling Internet of Things
- System equivalences
- Conclusions



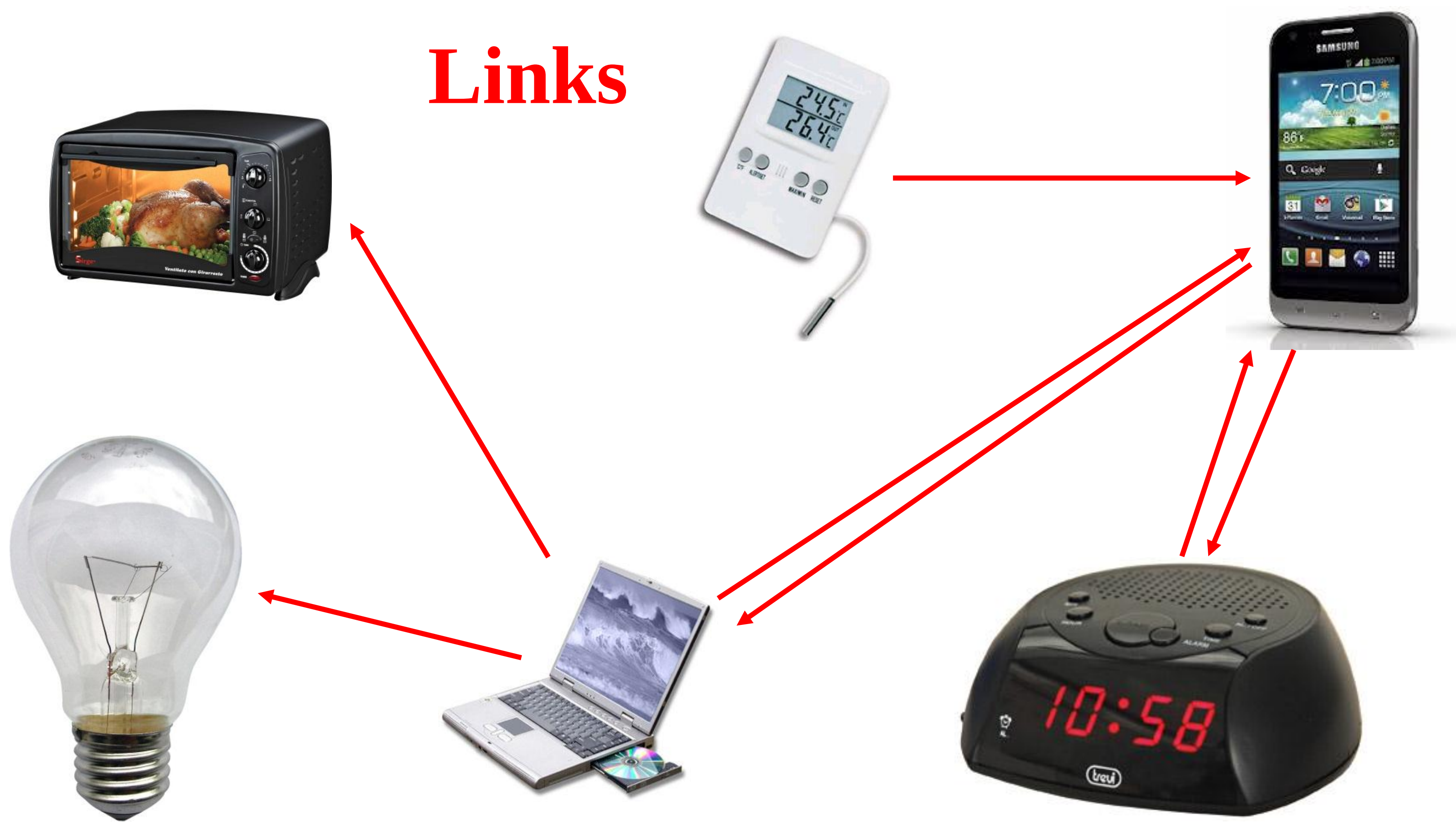
Internet of Things: relevant elements



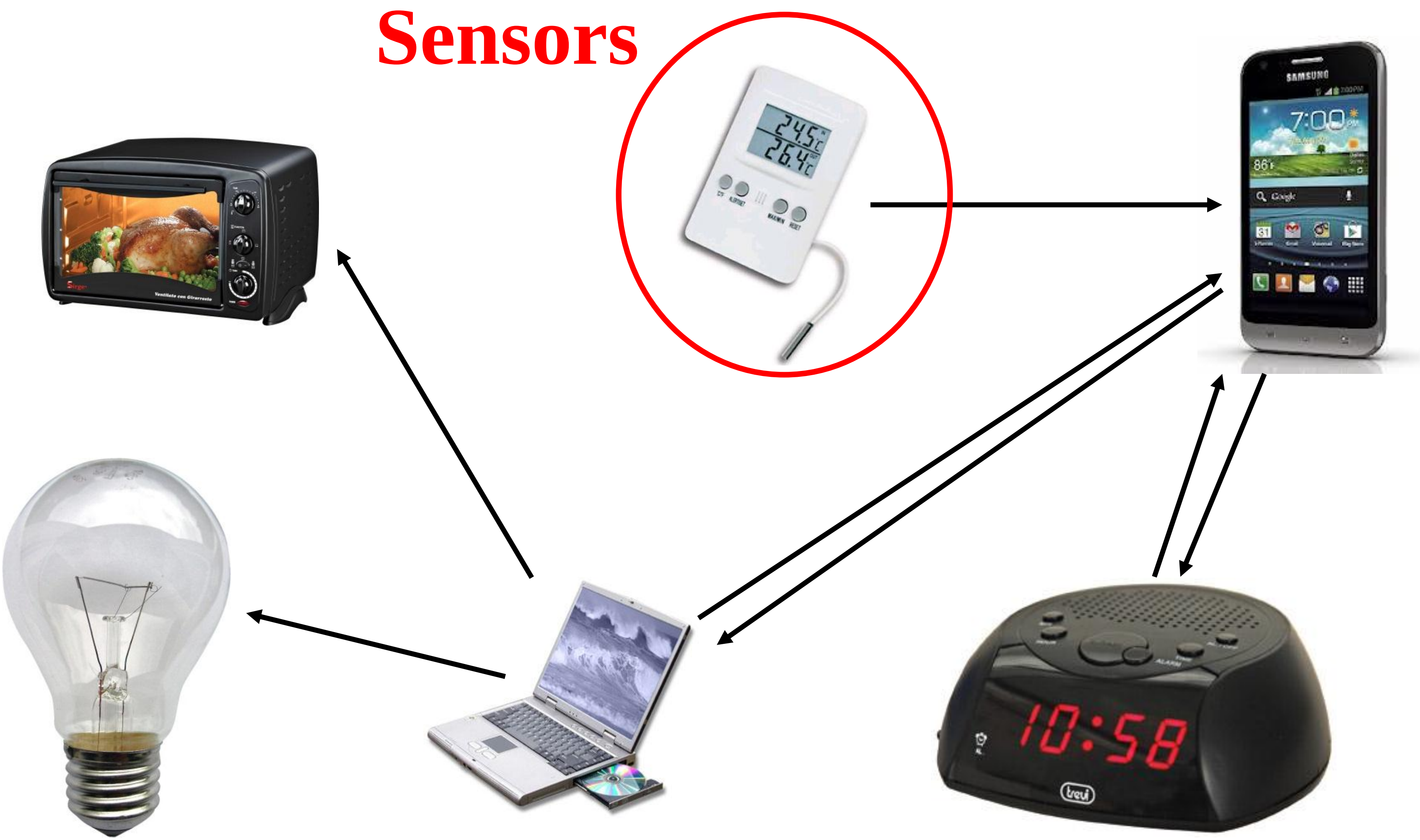
Internet of Things: relevant elements



Internet of Things: relevant elements

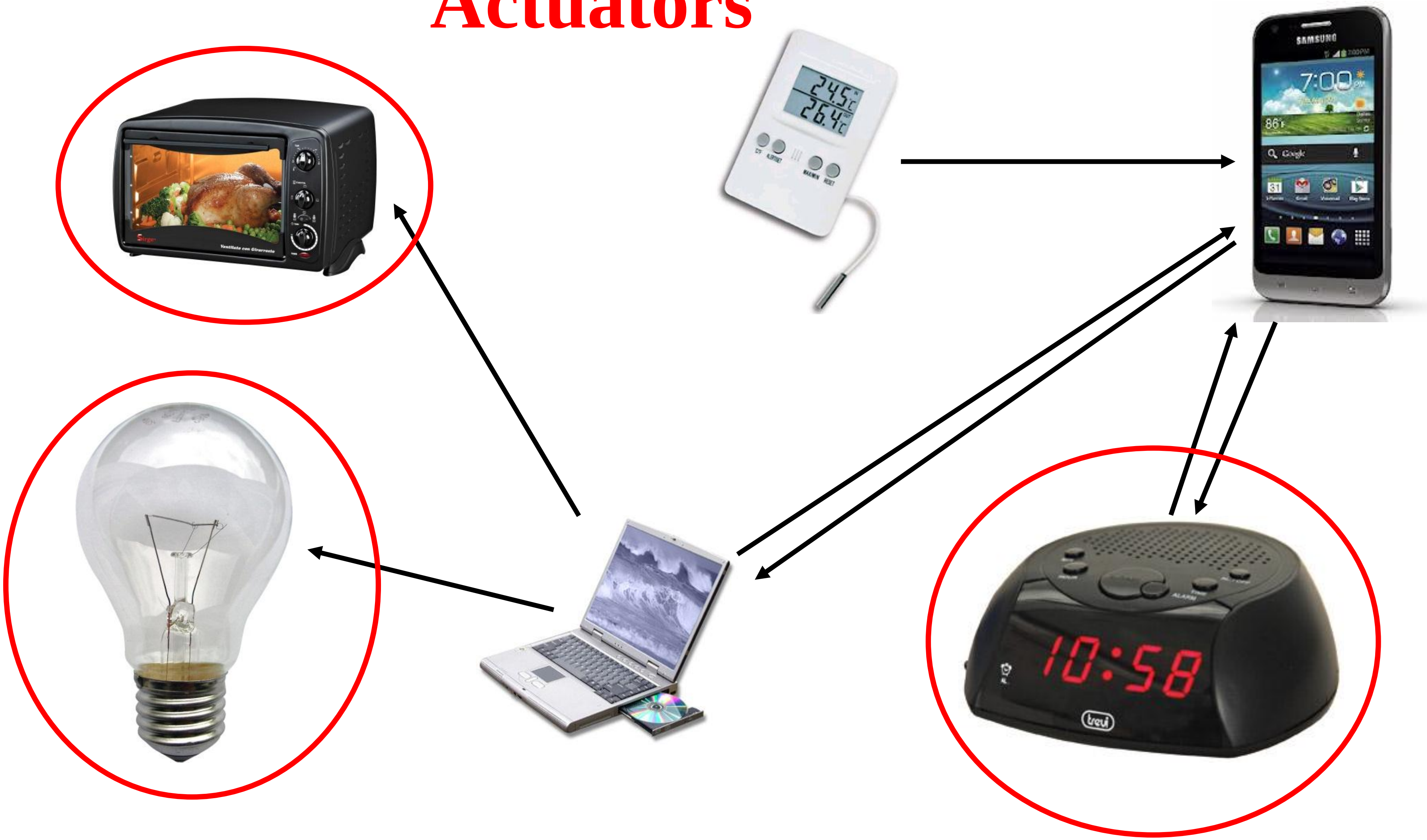


Internet of Things: relevant elements



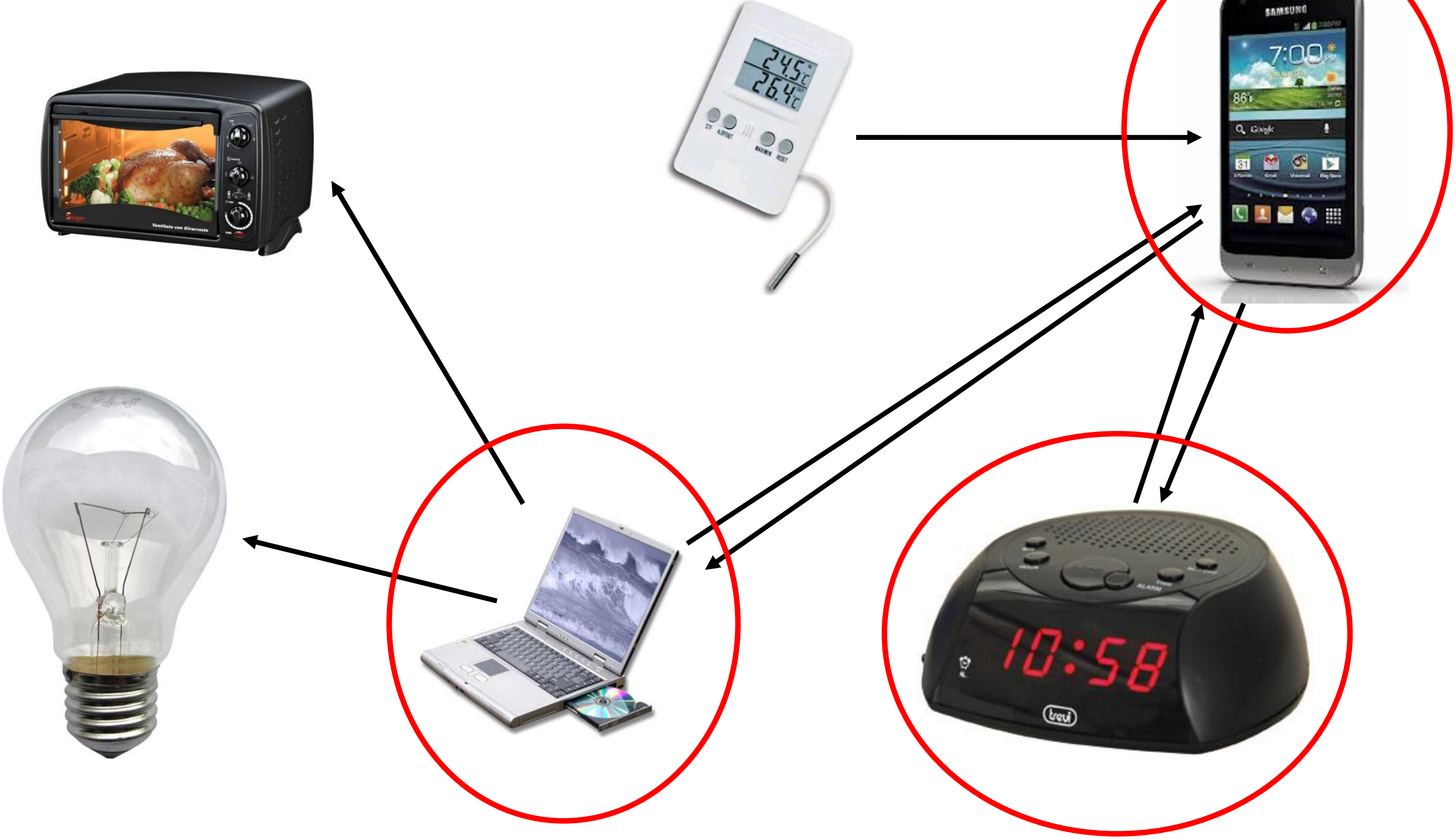
Internet of Things: relevant elements

Actuators



Internet of Things: relevant elements

Processes



Basic elements

- A network of nodes connected by edges
  - Nodes are things
  - Edges are communication links
- Nodes may include
  - Sensors
  - Actuators
  - Running processes
- Processes may
  - Communicate along links
  - Read sensors
  - Write to actuators

Syntax

$\mathcal{S} ::=$	$L \vdash N$	System
$L ::=$	$n \leftrightarrow m$	Link
	$L \mid L$	Parallel Composition
$N ::=$	$n[B]$	Node
	$N \mid N$	Parallel Composition
$B ::=$	$B \mid B$	Parallel Composition
	$(s \leftarrow v)$	Sensor
	$(a \rightarrow v)$	Actuator
	P	Running Process
$P ::=$	0	Nil
	$\overline{c}\langle v\rangle.P$	Output
	$c(x).P$	Input
	$!c(x).P$	Replicated Input
	$(x) \leftarrow s.P$	Read from Sensor
	$v \rightarrow a.P$	Write to Actuator

Semantics: main ideas

- Communication follows the π-calculus semantics but
  - Partial communication: only processes in the same node or in connected nodes can communicate
  - Actuators may be written, sensors may be read
- Connections with the real world
  - Sensors may be written by the environment
  - Actuators may be read by the environment
  - Topology changes are decided by the environment
- These transitions are always enabled

Semantics of processes

$$(\textsc{Out})\ \overline{c}\langle v\rangle.P \xrightarrow{\overline{c}\langle v\rangle} P \qquad (\textsc{In})\ c(x).P \xrightarrow{c(v)} P\{^{v}/_{x}\}$$

$$(\textsc{RIn})\ !c(x).P \xrightarrow{c(v)} !c(x).P \mid P\{^{v}/_{x}\} \qquad (\textsc{Rd})\ (x) \leftarrow s.P \xrightarrow{v \leftarrow s} P\{^{v}/_{x}\}$$

$$(\textsc{CoRd})\ (s \leftarrow v) \xrightarrow{(s \leftarrow v)} (s \leftarrow v) \qquad (\textsc{Wt})\ v \rightarrow a.P \xrightarrow{v \rightarrow a} P$$

$$(\textsc{CoWt})\ (a \rightarrow v) \xrightarrow{(a \rightarrow v')} (a \rightarrow v') \qquad (\textsc{Sens})\ (s \leftarrow v) \xrightarrow{\tau : s \leftarrow v'} (s \leftarrow v')$$

$$(\textsc{Act})\ (a \rightarrow v) \xrightarrow{\tau : a \rightarrow v} (a \rightarrow v) \qquad (\textsc{Par})\ \frac{B \xrightarrow{\alpha} B'}{B \mid C \xrightarrow{\alpha} B' \mid C}$$

$$(\textsc{IntSynch})\ \frac{B \xrightarrow{\alpha} B' \quad C \xrightarrow{\overline{\alpha}} C'}{B \mid C \xrightarrow{\tau} B' \mid C'}$$

Semantics of networks

$$(\textsc{Node})\ \frac{B \xrightarrow{\alpha} B' \quad \alpha \notin \{\tau, \tau : s \leftarrow v, \tau : a \rightarrow v\}}{n[B] \xrightarrow{n:\alpha} n[B']}$$

$$(\textsc{NodePass})\ \frac{B \xrightarrow{\alpha} B' \quad \alpha \in \{\tau, \tau : s \leftarrow v, \tau : a \rightarrow v\}}{n[B] \xrightarrow{\alpha} n[B']}$$

$$(\textsc{NodePar})\ \frac{N \xrightarrow{\alpha} N'}{N \mid M \xrightarrow{\alpha} N' \mid M} \qquad (\textsc{Synch})\ \frac{N \xrightarrow{n:\alpha} N' \quad M \xrightarrow{m:\overline{\alpha}} M' \quad n \leftrightarrow m}{N \mid M \xrightarrow{\tau} N' \mid M'}$$

$$(\textsc{Judg})\ \frac{N \xrightarrow{\alpha} N'}{L \vdash N \xrightarrow{\alpha} L \vdash N'} \qquad (\textsc{Conn})\ L \vdash N \xrightarrow{\tau:n\leftrightarrow m} L \mid n \leftrightarrow m \vdash N$$

$$(\textsc{Disc})\ L \mid n \leftrightarrow m \vdash N \xrightarrow{\tau:n\not\leftrightarrow m} L \vdash N$$

Sample system

$$n_s[(s_t \leftarrow 20)] \qquad n_a[(a_d \rightarrow 15)]$$

$$n_t[!disp().(x) \leftarrow s_t.x \rightarrow a_d]$$

Sample system

$$n_s[(s_t \leftarrow 20)] \qquad n_a[(a_d \rightarrow 15)]$$

$$n_t[!disp().(x) \leftarrow s_t.x \rightarrow a_d \mid \overline{disp}\langle\rangle]$$

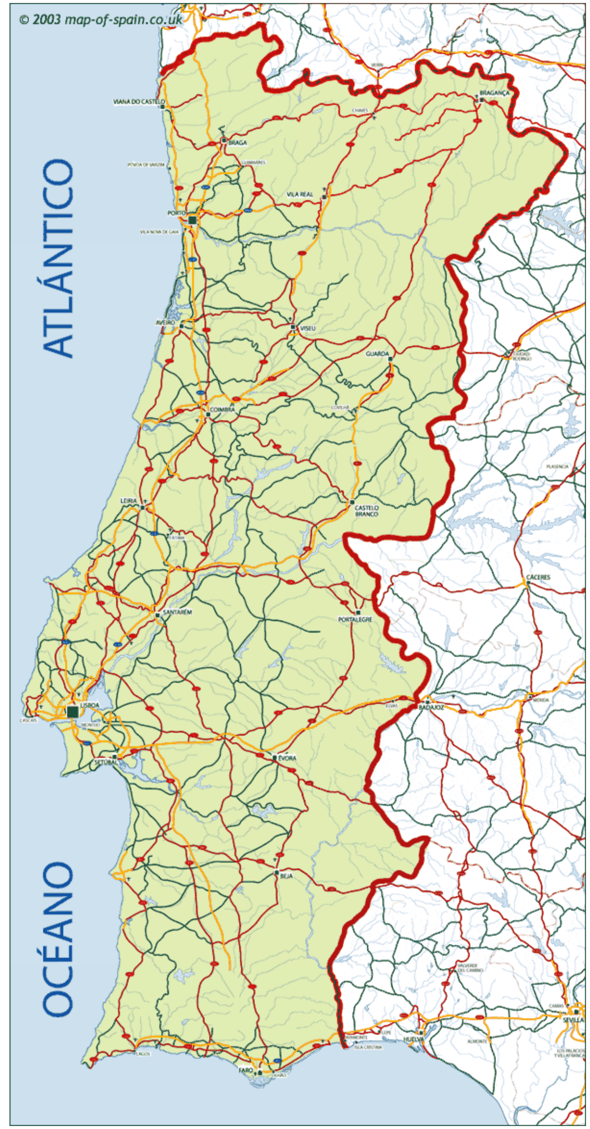
Sample system

$$n_s[(s_t \leftarrow 20)] \qquad n_a[(a_d \rightarrow 15)]$$

$$n_t[!disp().(x) \leftarrow s_t.x \rightarrow a_d \mid$$
$$!trig().\overline{disp}\langle\rangle.\overline{trig}\langle\rangle \mid \overline{trig}\langle\rangle]$$

Map of the talk

- Motivation
- Modeling Internet of Things
- System equivalences
- Conclusions

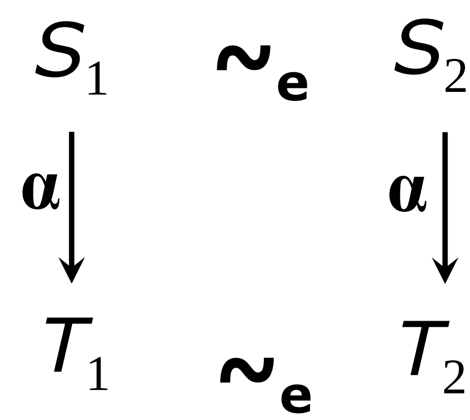


System equivalence

- Are two systems equivalent from the point of view of the end user?
- What can the end user observe?
  - Can see the values provided by the actuators
  - Can send values to sensors
  - Can move things thus creating and removing connections

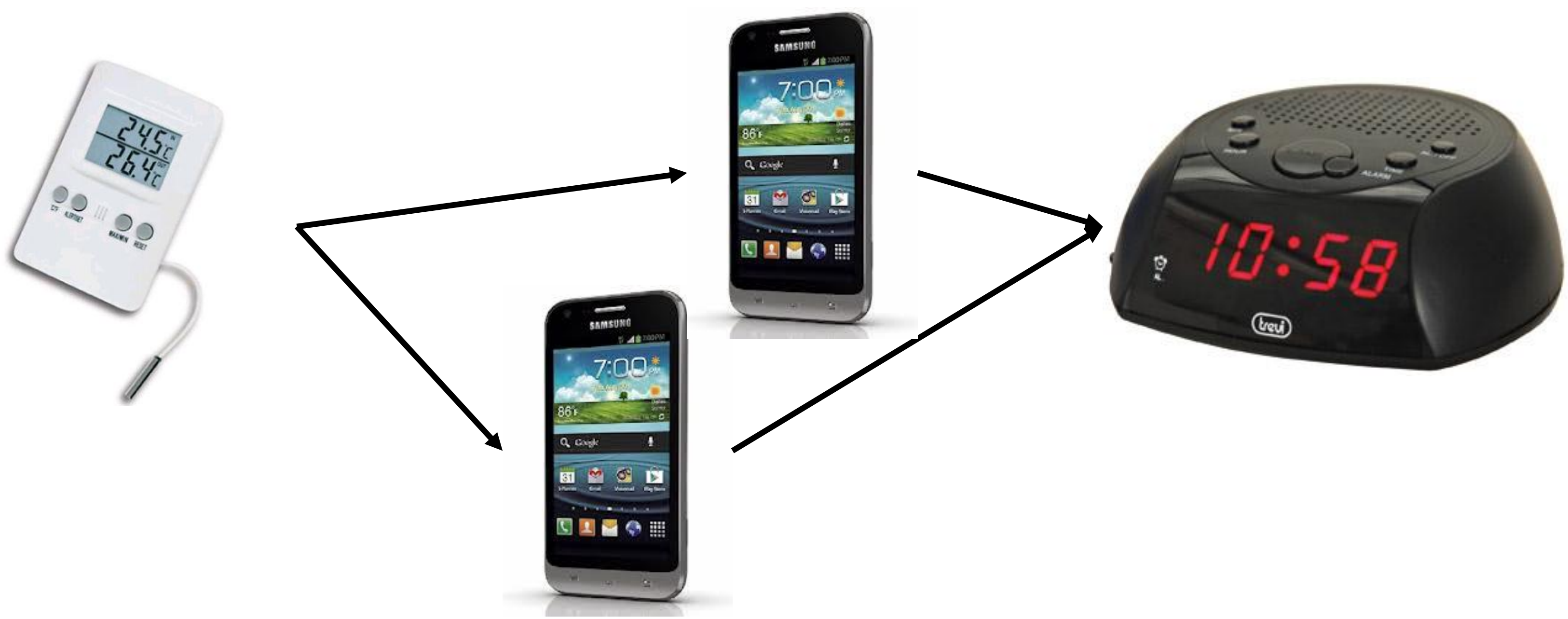
End-user bisimilarity

- We use a form of bisimilarity
  - Two systems are equivalent iff they can match each other actions and go to equivalent systems
  - Only notification actions (including τ) are valid challenges
- Strong and weak variants
  - Strong: $\sim_e$
  - Weak: $\approx_e$, needs not to match the number of τs



Strong end-user bisimilar systems

$\sim_e$



Weak end-user bisimilar systems

$\approx_e$

Compositionality issues

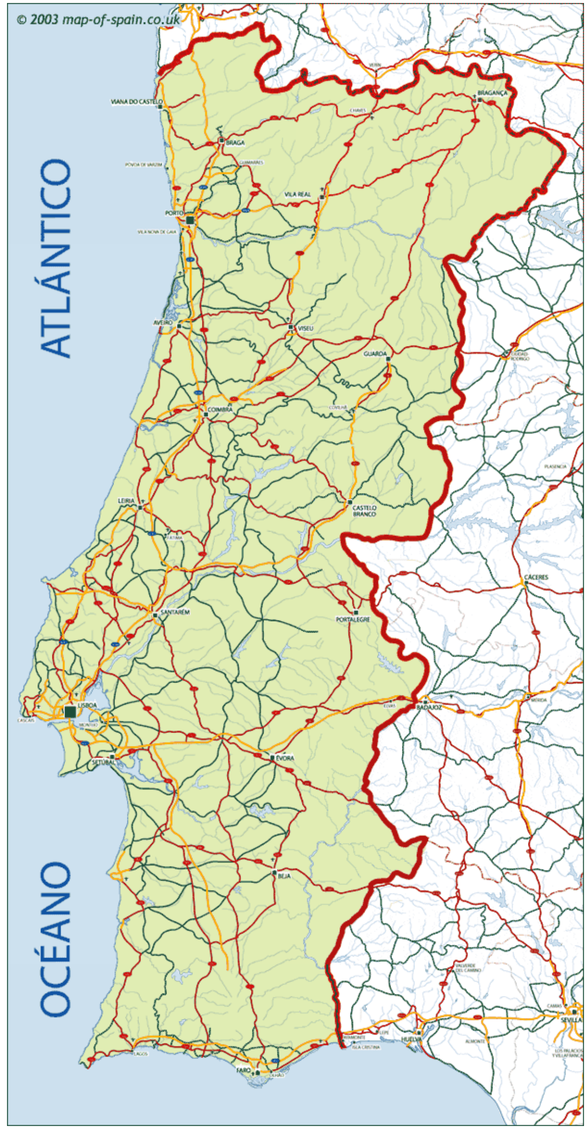
- End-user bisimilarity is not compositional
- Proof sketch:
  - Any two systems with the same sensors and nodes, and no actuators are weak bisimilar, regardless of their computation capabilities
  - Easy to find two such systems that are no more equivalent when adding the same actuator
- Difficult to prove end-user bisimilarity for large systems

A criterium for end-user bisimilarity

- We need an equivalence:
  - Compositional
  - Which implies end-user bisimilarity
- Standard bisimilarity satisfies the requirements
  - Essentially, all actions have to be matched, not only notification actions
  - Strong and weak variants

Map of the talk

- Motivation
- Modeling Internet of Things
- System equivalences
- Conclusions



Did we succeed?

- Which features did our model captures?
  - Distribution
  - Wireless communications (normally short range)
  - Heterogeneity
  - Effects on real world
  - Partial communication
  - Dynamicity
  - Self-healing

Future work

- Refining the calculus
  - Applying it to more complex case studies
- Controlling the cost of computation and communication
  - In terms of energy consumption
  - Looking for bounds on the energy needed for a given computation
- Controlling mobility
  - Not all mobility patterns are allowed
  - Impact on bisimulations

End of talk

Thanks!

Questions?