POLITECNICO
MILANO 1863

Computing Infrastructures
-
System Dependability

Roberto Sala
roberto.sala@polimi.it
Contribution: Luca Cassano

Lecturer, web page & students appointments

Roberto Sala
roberto.sala@polimi.it

DEIB, 3rd floor, Building 22, Office 004

Send an email to fix an appointment.

What is dependability?

What is dependability?

A measure of how much we trust a system…

…from a microwave oven up to an airplane!

What is dependability?

The ability of a system to perform its functionality while exposing:

- Reliability
- Availability
- Maintainability
- Safety
- Security

What is dependability?

The ability of a system to perform its functionality while exposing:

- Reliability
- Availability
- Maintainability
- Safety
- Security

Continuity of correct service

What is dependability?

The ability of a system to perform its functionality while exposing:

- Reliability
- Availability
- Maintainability
- Safety
- Security

Readiness for correct service

What is dependability?

The ability of a system to perform its functionality while exposing:

- Reliability
- Availability
- Maintainability
- Safety
- Security

Ability for easy maintainance

What is dependability?

The ability of a system to perform its functionality while exposing:

- Reliability
- Availability
- Maintainability
- Safety
- Security

Absence of catastrophic consequences

What is dependability?

The ability of a system to perform its functionality while exposing:

- Reliability
- Availability
- Maintainability
- Safety
- Security

Confidentiality and integrity of data

Why dependability?

Why dependability?

A lot of effort is devoted to make sure the implementation

- matches specifications
- fulfills requirements
- meets constraints
- optimizes selected parameters (performance, energy, …)

Why dependability?

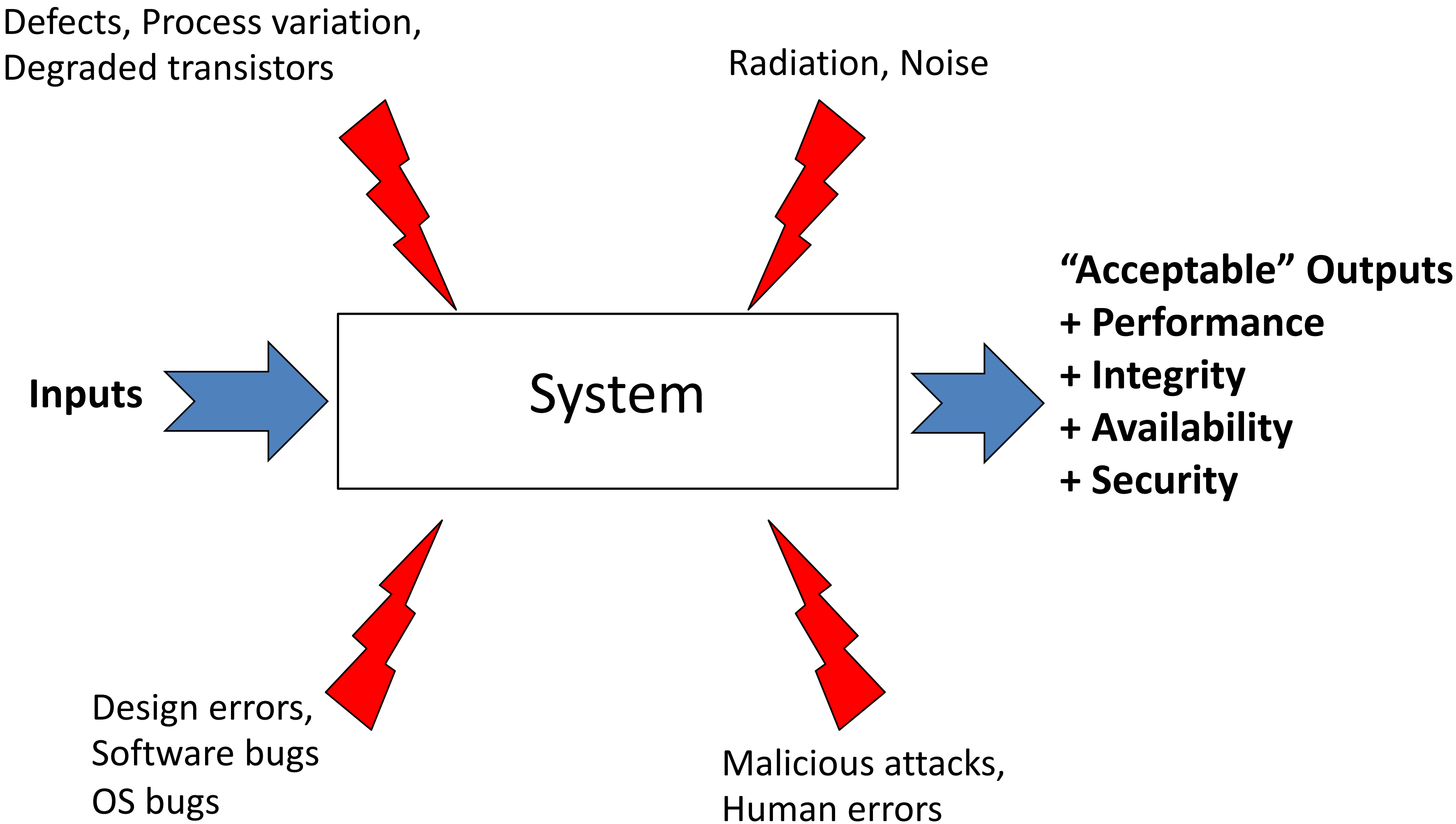
Functional Verification

A lot of effort is devoted to make sure the implementation
- matches specifications
- fulfills requirements
- meets constraints
- optimizes selected parameters (performance, energy, …)

Nevertheless, even if all above aspects are satisfied … things may go wrong
- systems fail

systems fail … because something broke

Why dependability?



Failure effects

A single system failure may affect a large number of people

Failure effects

A failure may have high costs if it impacts economic losses or physical damage

Failure effects

Systems that are not dependable are likely not be used or adopted

Failure effects

Undependable systems may cause information loss with a high consequent recovery cost

Why dependability?

Industrial standards require it:

- ISO 26262 for automotive
- CENELEC 50128 (SW) and 50129 (HW) for railways
- RTCA DO-178C (SW) and DO-254 (HW) for airborne
- ESA ECSS-E-ST-40C (SW) and ECSS-Q-ST-60-02C (HW) for space
- ….

When to think about dependability?

When to think about dependability?

Both at design-time and at runtime

Always!!!

When to think about dependability?

Both at $\boxed{\text{design-time}}$ and at runtime

- Analyse the system under design
- Measure dependability properties
- Modify the design if required

When to think about dependability?

Both at design-time and at runtime

- Detect malfunctions
- Understand causes
- React

When to think about dependability?

Failures occur in development & operation

- Failures in development *should* be avoided
- Failures in operation *cannot* be avoided (things break), they must be dealt with

When to think about dependability?

Failures occur in development & operation

- Failures in development *should* be avoided
- Failures in operation *cannot* be avoided (things break), they must be dealt with

Design should take failures into account and guarantee that control and safety are achieved when failures occur

When to think about dependability?

Failures occur in development & operation

- Failures in development *should* be avoided
- Failures in operation *cannot* be avoided (things break), they must be dealt with

Design should take failures into account and guarantee that control and safety are achieved when failures occur

Effects of such failures should be predictable and deterministic … not catastrophic

Where to apply dependability?

Where to apply dependability?

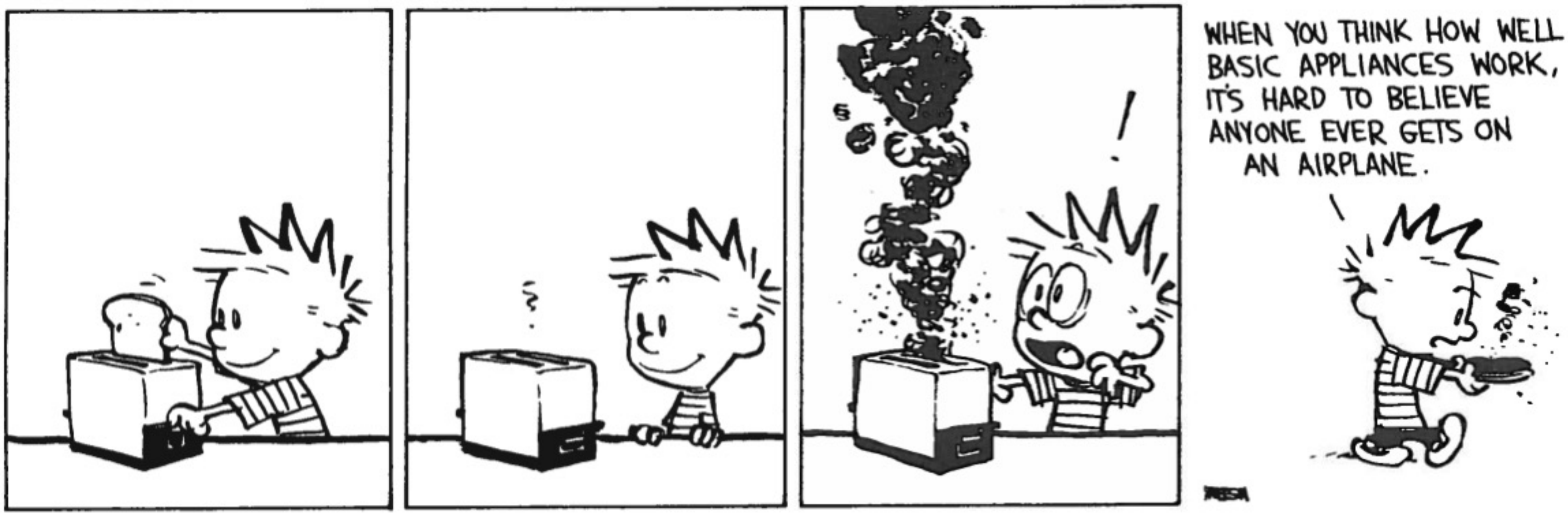
Once upon a time …

…dependability has been a <u>relevant aspect</u> only for safety-critical and mission-critical application environments

- Space
- Nuclear
- Avionics

Huge costs, acceptable only when mandatory …

However …



THE DAYS ARE JUST PACKED
A Calvin and Hobbes Collection by Bill Watterson

"When you think how well basic appliances work, it's hard to believe anyone ever gets on an airplane."

Non-critical critical systems

Non-critical critical systems: a failure during operation can have economic and reputation effects

- Consumer products

Mission-critical systems

Mission-critical systems: a failure during operation can have serious or irreversible effects on the mission the system is carrying out

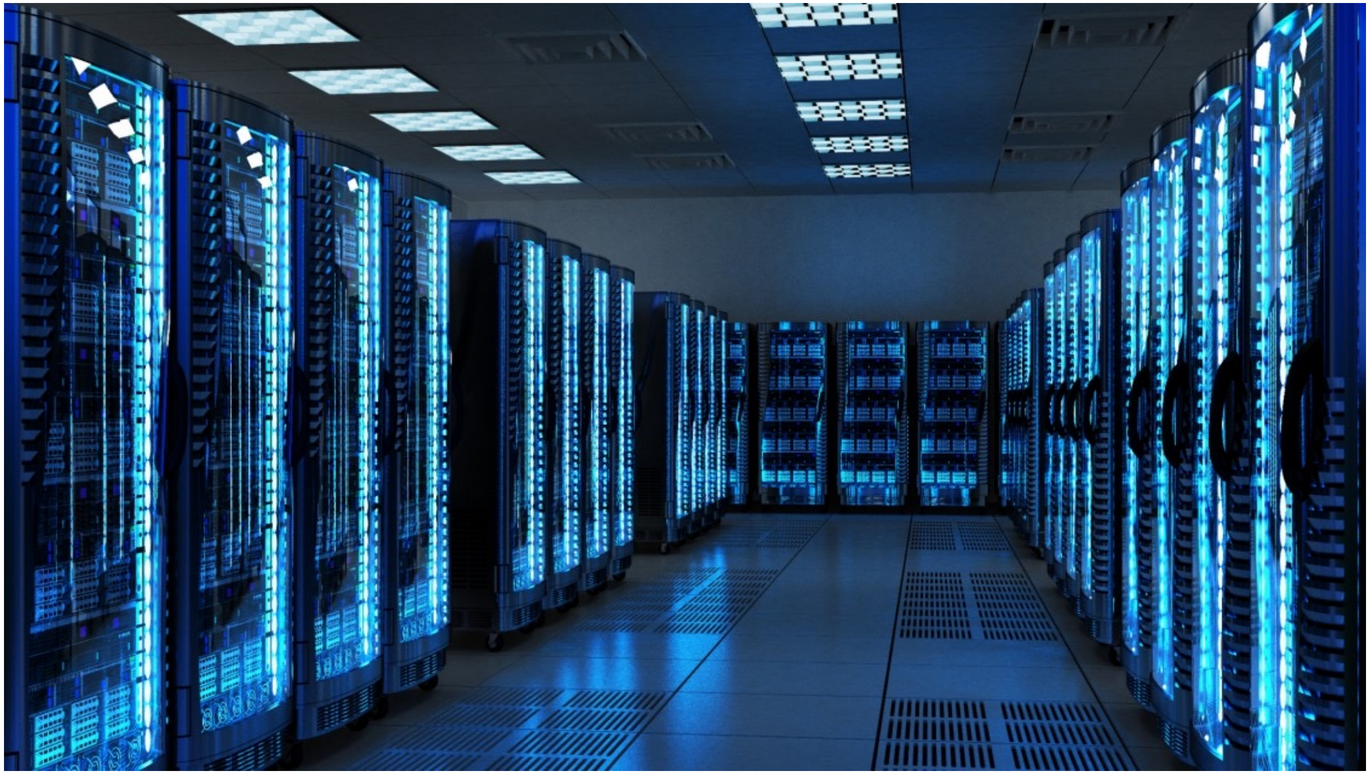
- Satellites
- Automatic Wheather Stations
- Surveillance drones
- Unmanned vehicles

Safety-critical systems

Safety-critical systems: a failure during operation can present a direct threat to human life

- aircraft control systems
- medical instrumentation
- railway signaling
- nuclear reactor control systems

Today and tomorrow



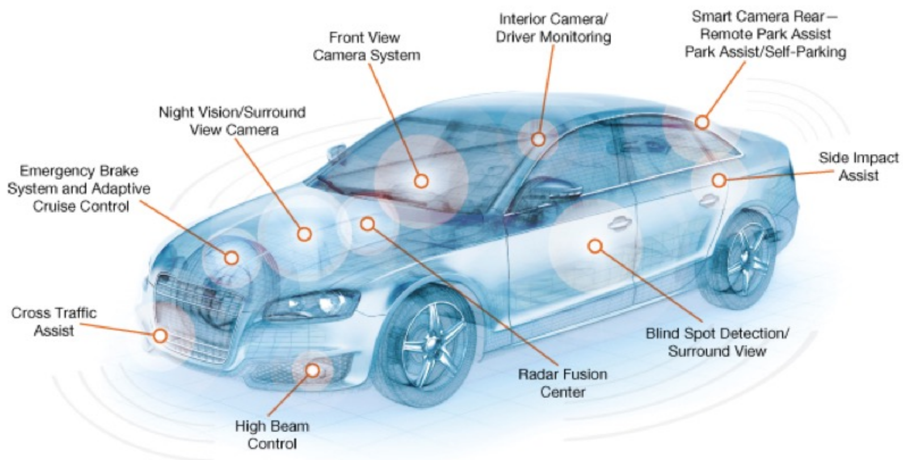
Downtime is the enemy of every data center.

Aberdeen Research reports the following downtimes and incidents:

- “Average” performing facilities, 60 minutes with 2.3 incidents per year.
- Best-in-class organizations, 6 minutes with 0.3 incidents per year.

Today and tomorrow

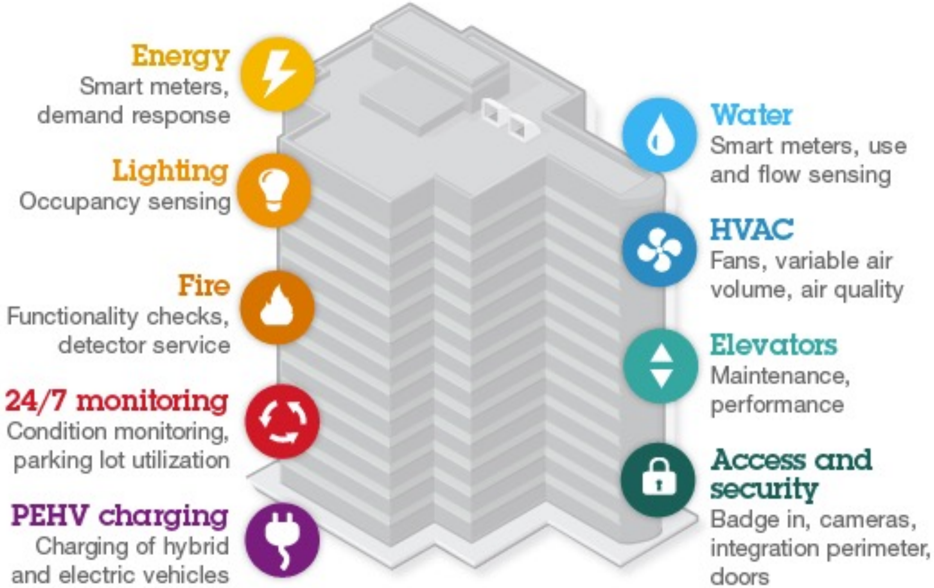
Automotive

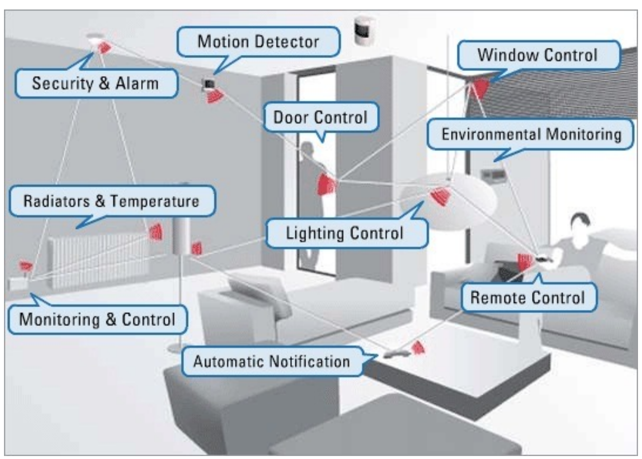


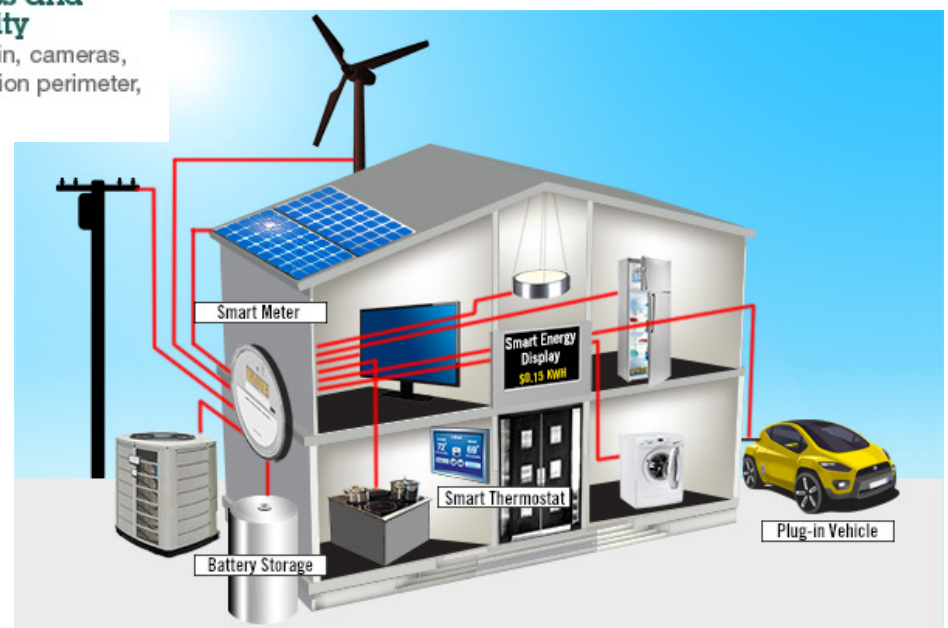
Today and tomorrow

Telecommunications

Today and tomorrow

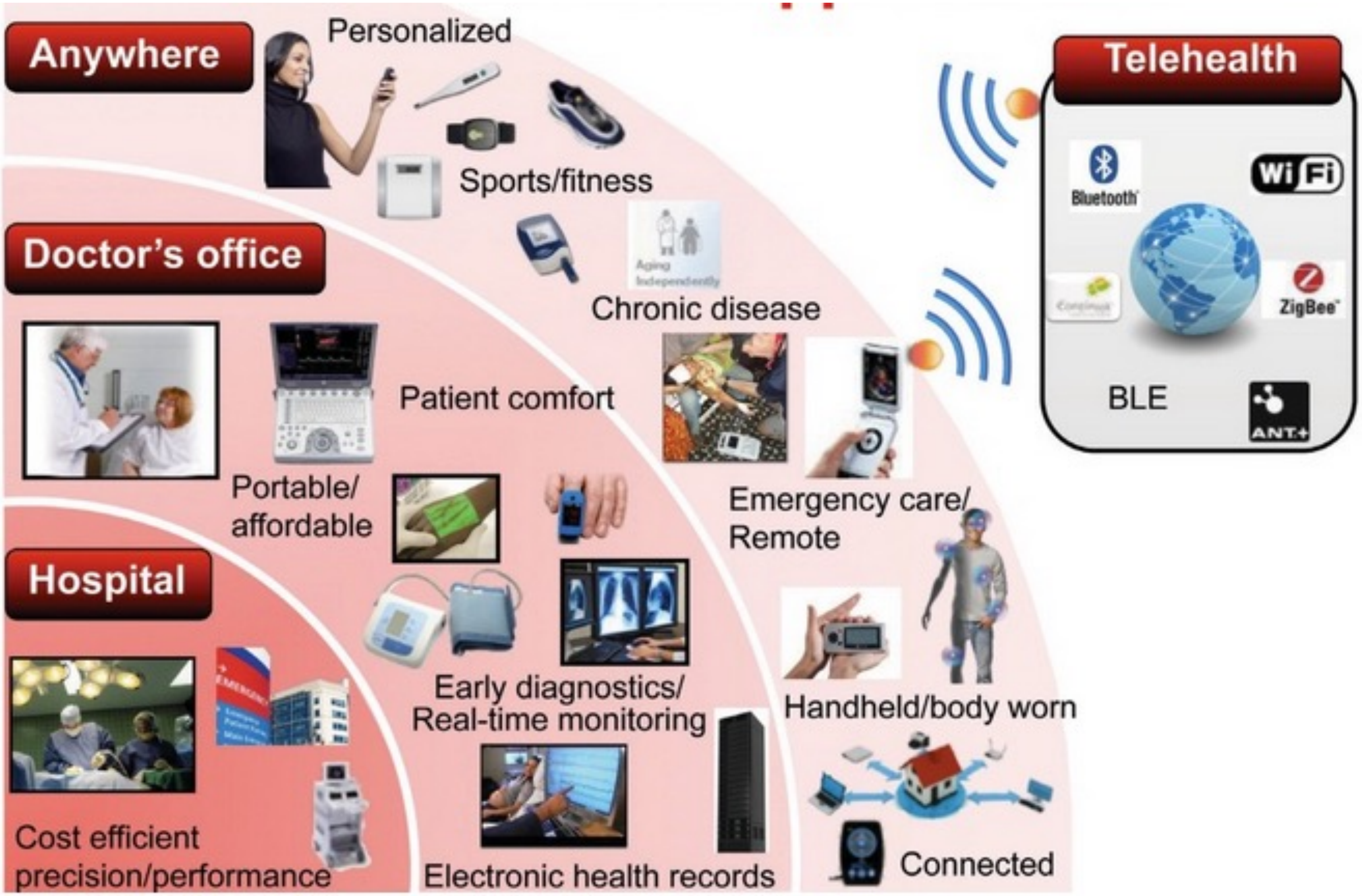






Smart spaces

Today and tomorrow



Creating solutions for health through technology innovation - Karthik Vasanth, Jonathan Sbert, Texas Instruments

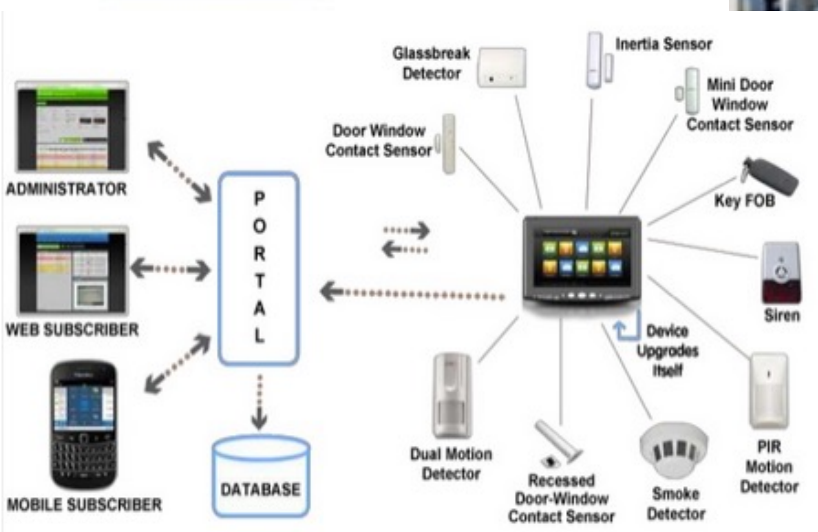
Today and tomorrow

Logistics

Creating solutions for health through technology innovation - Karthik Vasanth, Jonathan Sbert, Texas Instruments

Today and tomorrow

Security & surveillance



Creating solutions for health through technology innovation - Karthik Vasanth, Jonathan Sbert, Texas Instruments

Anatomy of the scenarios

the nodes

- computing systems
- sensors and actuators

the communication

- network

the cloud

- data storage
- data manipulation

Everything has to work properly for the overall system to be working

How to provide dependability?

Failure avoidance paradigm

Conservative design

Design validation

Detailed test

- Hardware
- Software

Infant mortality screen

Error avoidance

Failure tolerance paradigm

robust (computing) systems

Error detection / error masking during system operation

On-line monitoring

Diagnostics

Self-recovery & self-repair

Key elements

Dependable systems can be achieved by means of a

- Robust design (error-free)
  - Processes and design practices
- Robust operation (fault tolerant)
  - Monitoring, detection and mitigation

Safety-critical systems

They include all of the components that work together to achieve the safety-critical mission

- may include input sensors, digital data devices, hardware,
- peripherals, drivers, actuators, the controlling software, and
- other interfaces

Their development requires rigorous analysis and comprehensive design and test

Where to work

technological level

- design and manufacture by employing reliable/robust components
    - Highest dependability
    - High cost
    - Bad performance (generally devices from old generation)

Where to work

architectural level

– integrate normal components using solutions that allow to manage the occurrence of failures

- High dependability
- High cost
- Reduced performance

Depending on the adopted solution

Where to work

software/application level

– develop solutions in the algorithms or in the operating systems that mask and recover from the occurrence of failures

- High dependability
- High cost
- Reduced performance

Depending on the adopted solution

Where to work

What do all solutions have in common?

Where to work

What do all solutions have in common?

- Cost
- Reduced performance

You have to pay for dependability

Challenges

Design robust systems from unreliable cost Commercial Off-The-Shelf (COTS) components

Integrate COTS components to get a complex functionality

Tackle new problems introduced by technological advances

- Process variations
- Stressed working conditions
- With small geometries, several failure mechanisms, largely benign in the past, are becoming visible at the system-level

Challenges

Find the best tradeoff between dependability and costs depending on:

- **Application field**
  - Is there a specific design standard?
  - Which degree of dependability is actually required?
  - Will failures cause human losses?
  - Which would be the monetary cost of a failure?
  - Would a failure have a “reputation cost”?
  - ….

Challenges

Find the best tradeoff between dependability and costs depending on:

- **Working scenario**
  - Are there sources of faults (radiation, ageing, heat, vibration…)?
  - Which are the nominal working conditions (and the extreme ones) for the system?
  - Are there systems connected to my system?
  - ….

Challenges

Find the best tradeoff between dependability and costs depending on:

- **Employed technologies**
  - Are the cpu, memory, interfaces free from sources of failures?
  - Are the cpu, memory, interfaces tolerant to failures?
  - Which are the components most susceptible to failures?
  - …

Challenges

Find the best tradeoff between dependability and costs depending on:

- **Algorithms and applications**
  - Are the input of the application free of inexactness?
  - Is the algorithm tolerant to a certain degree of inexactness?
  - Can the application tolerate a certain “down-time”?
  - …

how to work

based on the application scenario:

- 100% dependability
  costs and overheads are relevant but are "justified"

- dependability is a trade-off with performance and power consumption