Mobile and Ubiquitous Computing

IN5020/9020 Autumn 2023

Lecturer: Amir Taherkordi

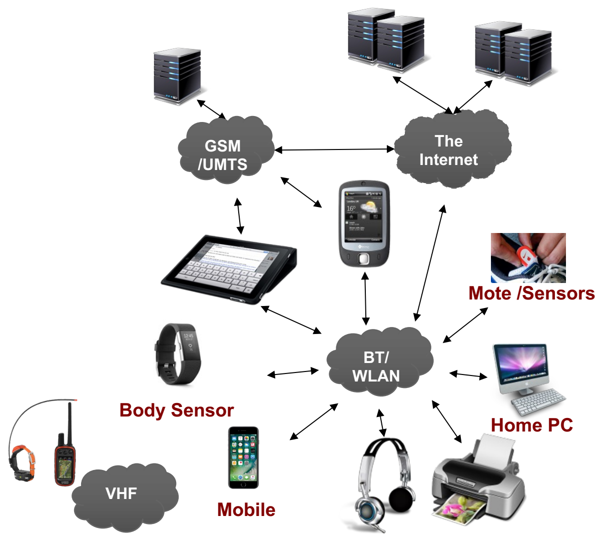
October 16, 2023

UiO : University of Oslo

Outline

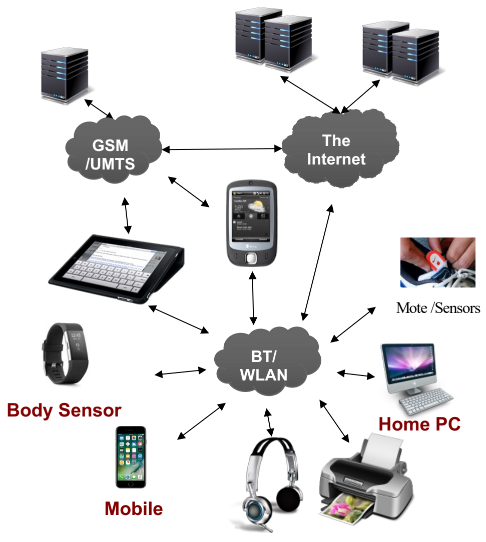
- Introduction
- Volatile Systems
- Integration with the Physical World
- Future Mobile Communications – 5G
- Summary

Motivation



- **Mobile computing:** exploiting the connectedness of portable devices
- **Ubiquitous computing:** exploiting the increasing integration of services and (small/tiny) computing devices in our everyday physical world
- **Mobile and ubiquitous computing:** requires particular solutions in many areas caused by dynamically changing computing environment: users, devices, and software components

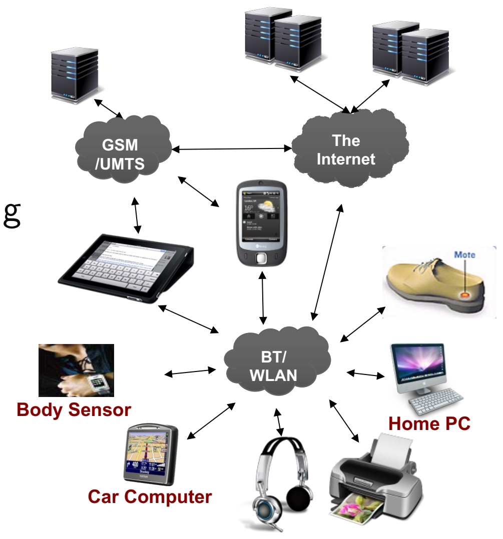
Some Open Questions



- How can software **components associate and interoperate** with one another while devices move, fail or spontaneously appear?
- How can systems become **integrated with the physical** world?
- How to adapt to small **devices' lack of computation and I/O resources**?
- How to handle **security in volatile**, physically integrated systems?

Fields and Subfields

- Mobile computing
- Ubiquitous computing
- Wearable computing
- Context-aware computing



Outline

- Introduction
- **Volatile Systems**
- Integration with the Physical World
- Future Mobile Communications - 5G
- Summary

Volatile Systems

- Common **system model** for mobile and ubiquitous computing (and their subfields)
- **Changes** (or **failures**) are considered **common** rather than exceptional (in contrast to other types of systems where changes or failures are considered to be exceptions)
- Forms of volatility
  - **failures** of devices and communication links
  - **changes** in the characteristics of communication such as bandwidth
  - the **creation and destruction of associations** – logical communication relations – between software components resident on the devices
- Mobile and ubiquitous computing exhibit **all** of the above forms of volatility

Modeling Elements

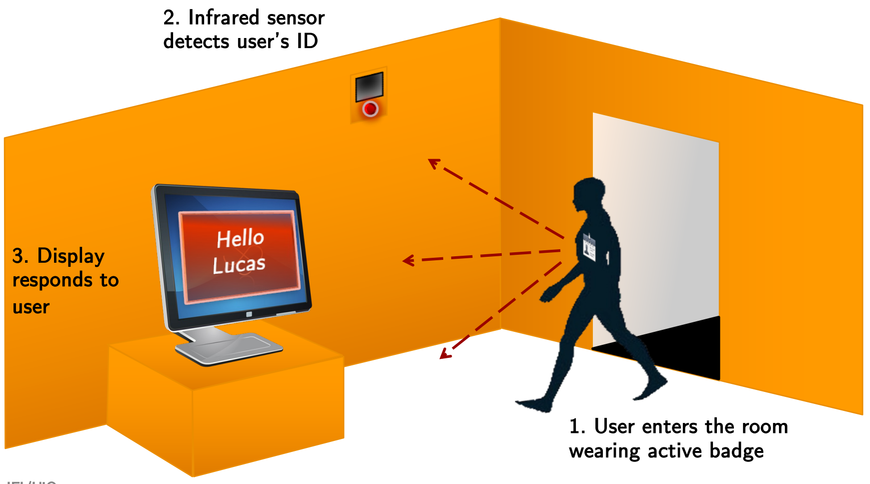
- Smart spaces
- Device model
- Volatile connectivity
- Spontaneous interoperation

Smart Spaces

- **Environments** within which volatile systems subsist
- A physical place/room with **embedded services**
  - The services are provided only or principally within that space
- Movements or "**appearance** or **disappearance**" in a smart space:
  - Physical mobility
  - Logical mobility
  - Service/device appearence
  - Service/device disappearance
- Example: Smart home systems

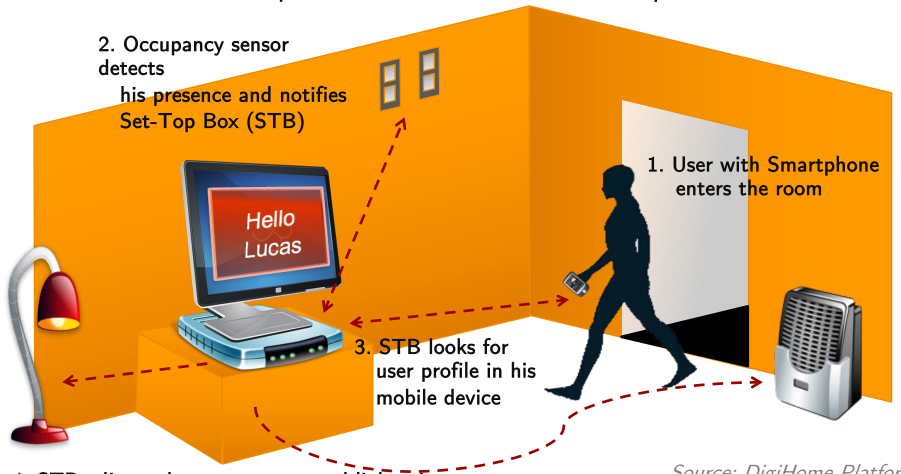
Smart Space: Example 1

- **Smart rooms**: responding to a user with an active badge


2. Infrared sensor detects user's ID
3. Display responds to user
Hello Lucas
1. User enters the room wearing active badge

Smart Space: Example 2

- **Smart homes:** respond to a user with a smartphone


2. Occupancy sensor detects
his presence and notifies Set-Top Box (STB)
Hello Lucas
1. User with Smartphone enters the room
3. STB looks for user profile in his mobile device
4. STB adjusts the temperature and lightening level
according to user's preferences

Source: DigiHome Platform

It is here

Local

Internet

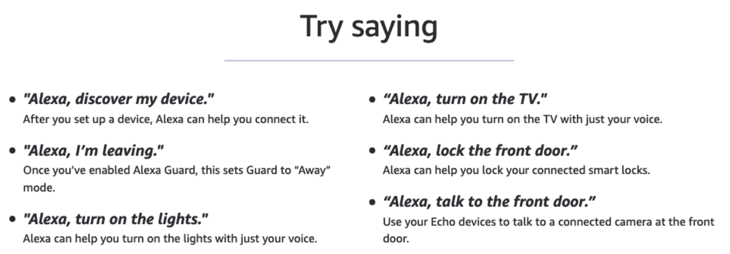
Microsoft Flow

zapier

IFTTT Platform

Amazon Alexa as a Smart Home Solution

- A virtual assistant AI technology developed by Amazon
  - first used in Amazon smart speakers
- capable of voice interaction, music playback, making to-do lists, setting alarms, etc.
- It can also control several smart devices using itself as a home automation system
  - extending the Alexa capabilities by installing "skills"



Try saying

- ***"Alexa, discover my device."***
  After you set up a device, Alexa can help you connect it.
- ***"Alexa, I'm leaving."***
  Once you've enabled Alexa Guard, this sets Guard to "Away" mode.
- ***"Alexa, turn on the lights."***
  Alexa can help you turn on the lights with just your voice.
- ***"Alexa, turn on the TV."***
  Alexa can help you turn on the TV with just your voice.
- ***"Alexa, lock the front door."***
  Alexa can help you lock your connected smart locks.
- ***"Alexa, talk to the front door."***
  Use your Echo devices to talk to a connected camera at the front door.

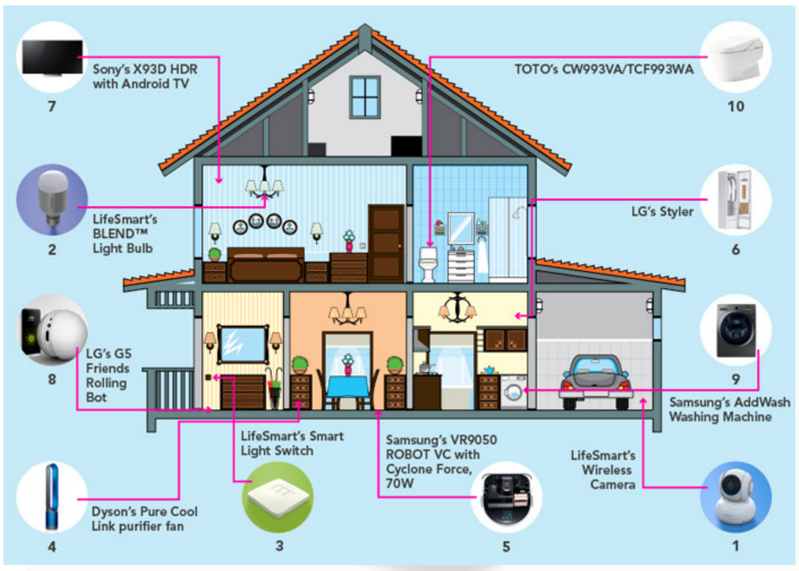
Modeling Elements

- Smart spaces
- Device model
- Volatile connectivity
- Spontaneous interoperation

Device Model

- Limited energy
  - typically use battery
  - energy-preserving algorithms
- Resource constraints
  - relative resource poor (CPU, memory, ...)
  - algorithmic challenge
- Sensors and actuators
  - to make a device context-aware
- Examples
  - Motes
  - Smart phones
  - ....

Real, Virtual and Digital Home


Sony's X93D HDR with Android TV
7
TOTO's CW993VA/TCF993WA
10
LifeSmart's BLEND™ Light Bulb
2
LG's Styler
6
LG's G5 Friends Rolling Bot
8
9
Samsung's AddWash Washing Machine
LifeSmart's Smart Light Switch
Samsung's VR9050 ROBOT VC with Cyclone Force, 70W
LifeSmart's Wireless Camera
Dyson's Pure Cool Link purifier fan
4
3
5
1

Modeling Elements

- Smart spaces
- Device model
- Volatile connectivity
- Spontaneous interoperation

Volatile Connectivity

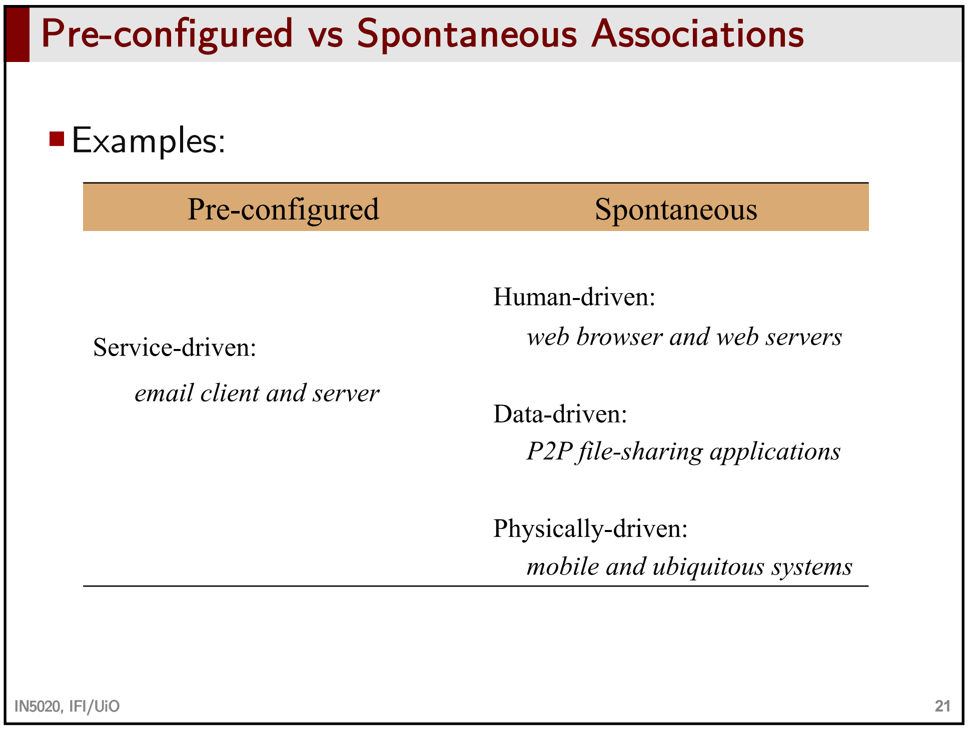
- Variation between different technologies (Bluetooth, WiFi, 3G, 4G, 5G, etc.)
  - Bandwidth, latency
  - Energy costs
  - Financial costs to communicate
- Disconnection
  - More likely in wireless networks
  - Multi-hop routing
- Variable bandwidth and latency
  - **Packet loss** due to weak signal
  - **Signal strength** varies
  - Difficult to determine **timeout values** in higher layer protocols due to varying conditions

Modeling Elements

- Smart spaces
- Device model
- Volatile connectivity
- Spontaneous interoperation

Spontaneous Interoperation

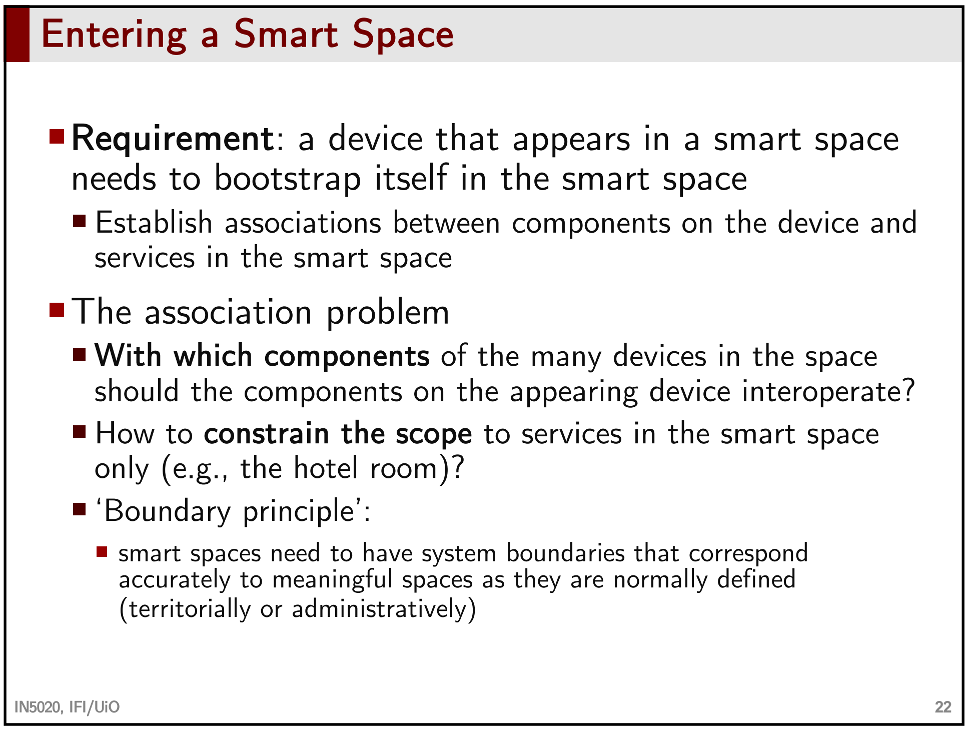
- In volatile systems, components routinely **change** the set of **components** they **communicate with**
  - take advantage of possibility to communicate with **local** components in a smart space, or
  - a device may want to **offer services** to clients in its local environment
- **Association**: a logical relationship formed when at least one of a given pair of components communicates with the other over some well-defined period of time
- **Interoperation:** interaction during an association
- **Spontaneous interoperation**: interoperation that is not planned or designed in!



Pre-configured vs Spontaneous Associations

- Examples:

Pre-configured	Spontaneous
	Human-driven: *web browser and web servers*
Service-driven: *email client and server*	Data-driven: *P2P file-sharing applications*
	Physically-driven: *mobile and ubiquitous systems*



Entering a Smart Space

- **Requirement**: a device that appears in a smart space needs to bootstrap itself in the smart space
  - Establish associations between components on the device and services in the smart space
- The association problem
  - **With which components** of the many devices in the space should the components on the appearing device interoperate?
  - How to **constrain the scope** to services in the smart space only (e.g., the hotel room)?
  - 'Boundary principle':
    - smart spaces need to have system boundaries that correspond accurately to meaningful spaces as they are normally defined (territorially or administratively)

Discovery Services

- A **directory service** that is used to register and look up services in a smart space
- Requirements to discovery services for a smart space
  - Service **attributes** is determined at runtime (hard!)
  - Service discovery must be possible in a smart space **without pre-existing infrastructure** to host a service discovery service
  - Registered services may **spontaneously disappear**
  - The protocols used for accessing the directory need to be sensitive to the **energy** and **bandwidth** they consume (cf. device model)

Interface to a Discovery Service

Methods for service de/registration	Explanation
lease := register(address, attributes)	Register the service at the given address with the given attributes; a lease is returned
refresh(lease)	Refresh the lease returned at registration
deregister(lease)	Remove the service record registered under the given lease
Method invoked to look up a service	
serviceSet := query(attributeSpecification)	Return a set of registered services whose attributes match the given specification

Discovery Services: Design Choices

- Directory server or serverless discovery
- **Directory server**: clients issue a multicast-request to locate the server (as in Jini)
  - Not all smart spaces have facilities for server implementations
- **Serverless** discovery: the participating devices collaborate to implement a distributed discovery service
    - **Push model**: servers multicast ('advertise') their descriptions regularly, and clients run their queries against them
    - **Pull model:** clients multicast their requests and devices providing matching services, respond
  - Both approaches are relatively resource demanding (battery, bandwidth) in their pure form

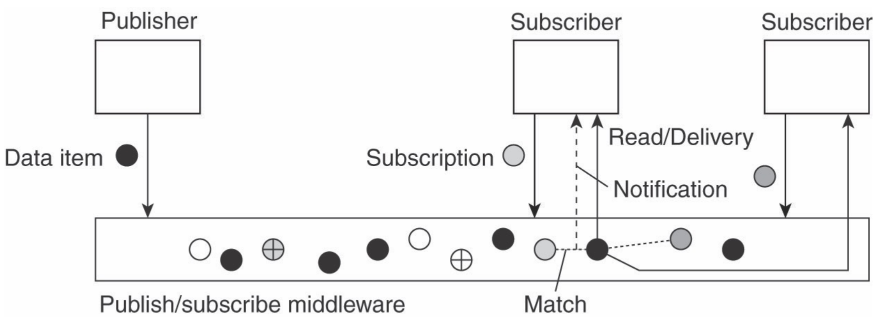
Interoperation

- How can components that want to associate determine **what protocol they can use** to communicate?
- Main problem is **incompatibility between software interfaces** (components need not have been designed together)
- Two approaches:
  - **Adapt interface** to each other (interface adaptation): difficult
  - Constrain interfaces to be **identical in syntax across as wide a class** of components as possible
    - Example: Unix pipes (read, write)
    - Example: The set of methods defined in HTTP (GET, POST, ...)
    - Such systems are called ***data oriented***
      - Require additional mechanisms to describe type and value of data exchanged (e.g. MIME types), as well as the processing semantics of the server (difficult!)

Data Oriented Programming Models

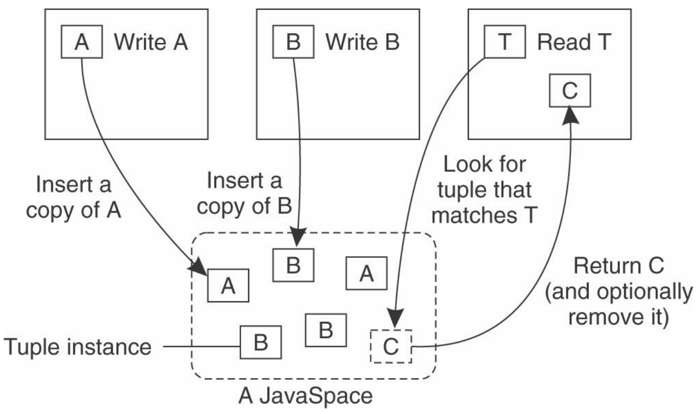
- **Data oriented programming models that have been used for volatile systems**:
  - Event-systems (pub/sub)
  - Tuple spaces
  - Direct device interoperation (devices brought into direct association)

Principle of publish-subscribe (event system)



Example of Tuple Space: JavaSpaces

- Event model is asynchronous, tuple spaces model is synchronous.
  - Problem in volatile systems?



Outline

- Introduction
- Volatile Systems
- **Integration with the Physical World**
- Future Mobile Communications – 5G
- Summary

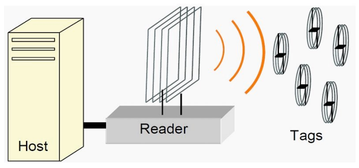
Mobile and Ubiquitous Computing Systems

How can such systems be integrated with the physical world?

- Key requirements:
  - Identification
  - Sensing
  - Processing
  - Communication
- Technological Development

Radio-Frequency Identification (RFID) → Wireless Sensor Networks (WSNs) → Internet of Things (IoT)

RFID Systems



- Composed of one or more reader(s) and several RFID tags
- Automatic identification of anything they are attached to (acting as an electronic barcode)
- RFID tag:
  - small microchip attached to an antenna, used for both receiving the reader signal and transmitting the tag ID)
  - has a unique identifier
  - *passive* RFID tags
  - *active* RFID tags
- Readers trigger the tag transmission by generating an appropriate signal
  - Signal: like a query for the possible presence of tags in the surrounding area and for the reception of their IDs.
- Applications: from logistics to e-health and security

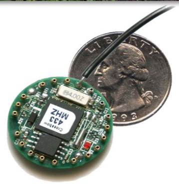
Wireless Sensor Networks (WSNs)

- Based on the need to:

  integrate systems with the **physical world** through **sensing** and **context** awareness

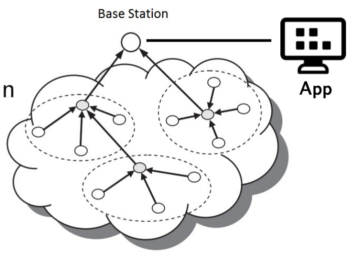
  - **Sensing**: use sensors to collect data about the environment
  - **Context** of an entity (person, place or thing): an aspect of its physical circumstances of relevance to system behaviour
  - **Context Awareness:** can respond to its (sensed) physical environments (location, heat, presence of a device, etc.) and the context can determine its (further) behaviour
- Sensors
  - Combination of hardware and software
  - Sensors are the basis for determining contextual values
    - Location, velocity, orientation, ...
    - Temperature, light intensity, noise, ...
    - Presence of persons or things (e.g., based on RFID)

Wireless Sensor Networks (WSNs) – cont'd



- Networks consisting of a (typically high) number of small, low-cost units or nodes
  - Nodes are more or less arbitrary arranged (e.g., "thrown out" in high numbers in a certain geographical area)
- Sensor nodes:
  - have **sensing** and processing capacity
  - can **communicate** wirelessly with a limited range (save energy)
  - act as **routers** for each other
- WSNs are volatile systems:
  - nodes can fail (battery exhaustion or otherwise destroyed (e.g., fire))
  - connectivity can change due to node failures
- WSNs are self-organising (ad-hoc network)
  - functions independently of an infrastructure

WSN Software Architecture



- WSN Applications:
  - normally operate on more abstract values than sensors can produce
- Sensor **abstractions**
  - to avoid application level concerns with the peculiarities of individual sensors
- Therefore common **to build a software architecture** for sensor data as hierarchies
  - Nodes at a low hierarchical level provide sensor data at a low level of abstraction (e.g. longitude/latitude of a device)
  - Nodes at higher hierarchical levels (closer to the root node) provide sensor data at higher levels of abstraction (e.g. device is in Frank's Cafe)
- Nodes at higher levels combine sensor data from lower levels both **to abstract and to increase reliability**

WSNs: Architectural Features

- Features: driven by requirements of **energy conservation** and **continuous operation**
- **In-network processing:** The nodes have processing capabilities because processing is much less costly in energy consumption than (wireless) communication. Can be exploited to reduce the need for communication (only communicate when there is a need for it)
- **Disruption-tolerant networking:** based on store-and-forward transfer of data (not end-to-end)
- **Data oriented programming of nodes:** since nodes can **fail**, we can not rely on programming techniques for sensor nodes that refer to single nodes

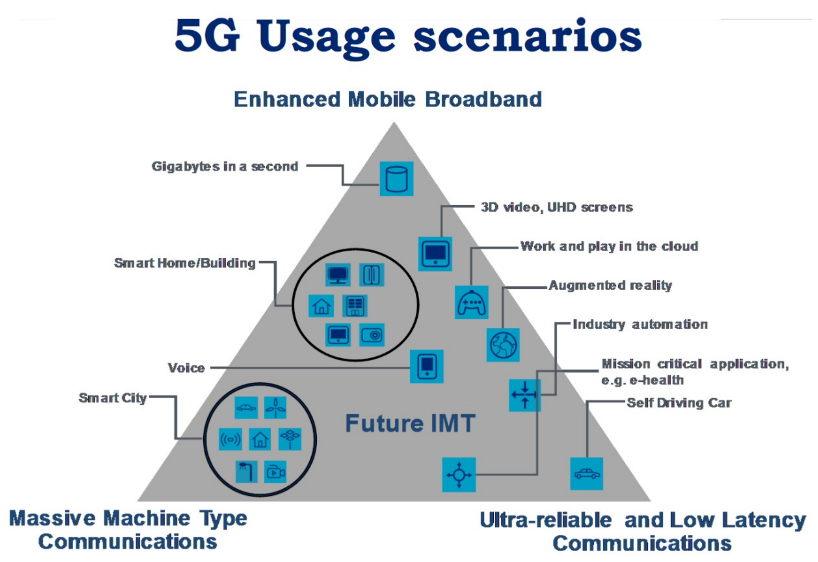
Outline

- Introduction
- Volatile Systems
- Integration with the Physical World
- **Today's Mobile Communications – 5G**
- Summary

5G Mobile Communications

- 5G: The recent generation of mobile standards being defined by the International Telecommunication Union (ITU)
- Capacities
  - Theoretical download speed 10Gbit/s
  - User experienced data rate up to 1 Gbit/s
  - 1 ms latency added in the radio network
  - Mobility up to 500 km/h
- Both macro-cell and small-cell
- 5G pilots were launched in 2018
- Deployments started from 2020

Usage Scenarios for 5G


5G Usage scenarios
Enhanced Mobile Broadband
Gigabytes in a second
3D video, UHD screens
Work and play in the cloud
Smart Home/Building
Augmented reality
Industry automation
Voice
Mission critical application, e.g. e-health
Smart City
Self Driving Car
Future IMT
Massive Machine Type Communications
Ultra-reliable and Low Latency Communications

Summary

- Most challenges to mobile and ubiquitous systems are caused by their **volatile** nature
- In such environments applications need to be context aware and adaptive
  - Integrated with the physical world through sensing and context awareness
  - Adapt to changes in the physical circumstances by changing behavior (e.g. component reconfiguration)
- From RFID to WSNS and today's IoT systems
- There are many challenges, but yet only few (comprehensive) solutions
- Future mobile communication may support mobile and ubiquitous computing applications which are not available today