ENGR5720 – Mobile and Pervasive Computing

Smart Devices

Lecture notes based on Ubiquitous Computing: Smart Devices, Environments and Interactions (Chapter 3) &

The Landscape of Pervasive Computing Standards (Chapters 2 & 4) & Personal Notes

OntarioTech UNIVERSITY

Topics Covered

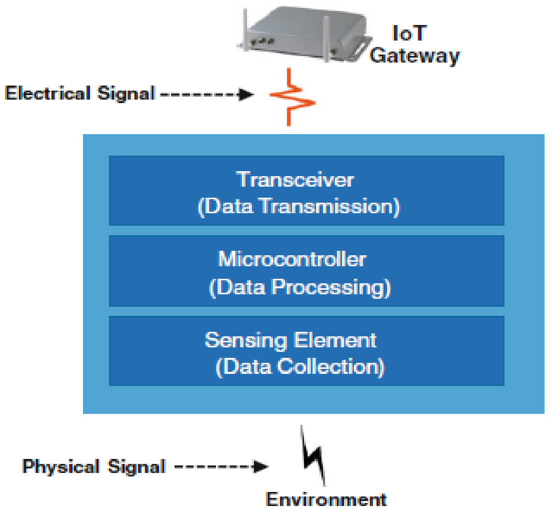
- Distributed Systems Viewpoint
- Sensing Elements
- Smart DEI Model

Sensing Element

This is the foundation of what the IoT is about.

IoT Sensors

- In the IoT most sensors are either “simple” or “smart” sensors.
- Simple
  - Generate data with no ability to filter
- Smart
  - Support some filtering capability, so require some form of programming capability.
- Many types of sensors exist in the market and this is very challenging but fortunately many sensors communicate through a few standard interfaces.
  - I2C, AD, Digital I/O
  - Most though are expected to communicate via IP.
- This implies that the sensors will have to support some form of network communications:
  - > cost, > processing time.



OntarioTech UNIVERSITY

General Purpose Input/Output (GPIO)

- Applications that use microcontrollers are rapidly growing as cost of production goes down and performance of embedded systems increase.
- A General Purpose Input/output (GPIO) is an interface available on most modern microcontrollers (MCU) used to connect microcontrollers to other electronic devices.
- These pins are available on a processor and can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.

General Purpose Input/Output

- The variable methods of data handling implemented in these pins, such as ADC conversion and interrupt handling, provide alternative uses that are ideal for multi-input applications.

Ontario Tech University

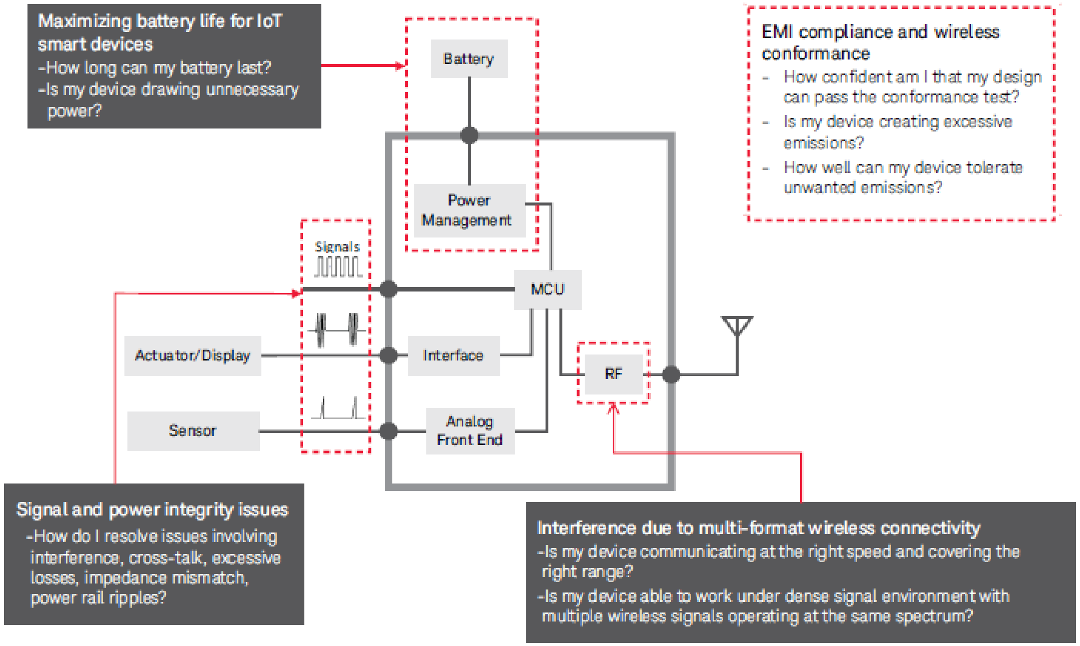
General Purpose Input / Output

- The pins can be programmed as input, where data from some external source is being fed into the system to be manipulated at a desired time and location
- Output can also be performed on GPIOs, where formatted data can be transmitted efficiently to outside devices, this provides a simple mechanism to program and retransmit data depending on user desires through a single port interface

OntarioTech UNIVERSITY

Sensor Challenges

- There are many challenges for sensor design that are more related to electronic design that we will not cover in this course but can affect software design.



Solving Test Challenges for Internet of Things-Enabled Consumer Electronics Devices, Keysight Technologies Application Note, June 2016.

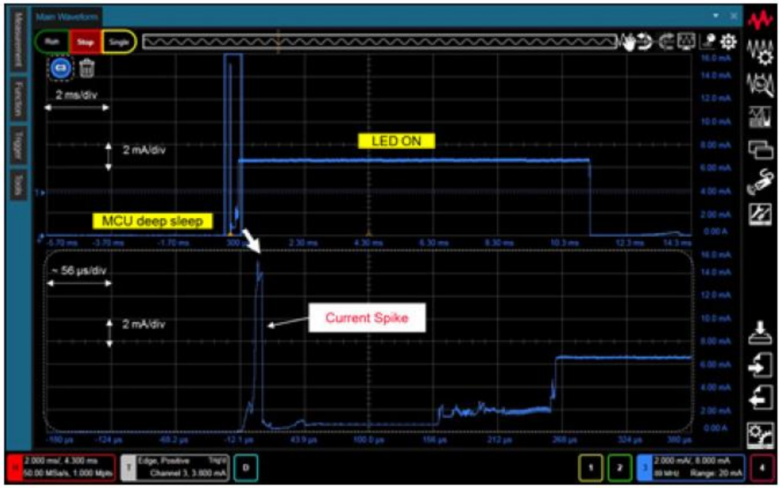
Wireless Conformance

- Wireless conformance ensures that a wireless device (e.g., Wi-Fi, Bluetooth, Zigbee, LTE) adheres to specific standards for signal quality, frequency use, transmission power, and protocol behavior.
- **Why it matters:**
  - Ensures the device communicates reliably and does not interfere with other wireless systems.
  - Guarantees interoperability between products from different manufacturers (e.g., a Bluetooth speaker working with all Bluetooth phones).
  - Required to legally market and sell wireless devices in many countries.
- **Examples of wireless conformance concerns:**
  - A device using non-standard transmission power might interfere with neighboring devices.
  - A Wi-Fi chip not properly implementing the protocol might cause dropped connections.

OntarioTech UNIVERSITY

Power Challenges

- Periodic low level current wave forms, sharp current spikes, and current spikes in deep sleep mode that can cause power draws.
  - Can be caused by hardware or software.



OntarioTech UNIVERSITY

IoT Device Current Drain Characteristics

- The device occasionally wakes up and briefly enters an active state to process data or communicate.
  - Extremely low duty cycle, often much less than one percent
  - Wide dynamic range (up to 600,000:1 current ratio between operating and sleep modes)
- Although the sleep/standby power consumption is very low, these low-power modes consume much of the battery's capacity.

Device Design Tips

1. Carefully consider the architecture of your device and its peripherals.
2. Take advantage of memory, clock, timer, and low-power state options.
3. Write your firmware with an eye toward reducing power consumption.
4. Measure power consumption with instruments of sufficient bandwidth and seamless ranging.
5. Frequently generate automated current profiles to evaluate firmware changes.

Architecture and Peripherals

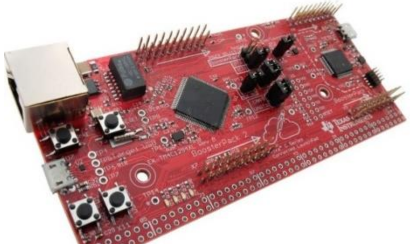
- The MCU hardware architecture that you choose will have a major impact on power consumption.
  - efficient DC-DC buck converters with a variety of input voltages can save power.
  - Math accelerators that are optimized to perform integer and floating point arithmetic for cyclic redundancy checks, cryptography, and data analysis.
  - some MCUs integrate RF radios, sensors, and other peripherals.

OntarioTech UNIVERSITY

Microcontrollers

Texas Instruments LaunchPad

	MSP-EXP430G2	MSP430F5529	Tiva C Series TMC4C1294
Microcontroller	MSP430	MSP430	TM4C1294NCPDT ARM Cortex-M4
Flash Memory	16 KB	128KB	1 MB
Clock Speed	16 Mhz	25 MHz	120 Mhz
RAM	512B	8KB	256 KB
Price (approx, USD)	$9.99	$12.99	$19.99
Other			Ethernet



https://www.ti.com/sitesearch/en-us/docs/universalsearch.tsp?langPref=en-US&nr=1359&searchTerm=MSP-EXP430G2#q=MSP-EXP430G2

Memory, Clock, Timer, and Low-power States

- Memory usage and memory type affects power consumption.
  - Unused memory uses power, disable it.
- Types of clocks and timers on your MCU module will also affect battery runtime.
  - Master, real-time, and and low-power clocks might be available for use.
- Understand the different low power modes of the MCU.
  - idle, snooze, sleep, hibernation, and so on.
  - Sensors in a network can never really go to sleep unless their communications is synchronized.
    - They work in a low power listening mode in case a message is sent to them.

Common Low Power Modes in MCUs

Mode	Power Consumption	What Stays On	Use Case
Active Mode	High	CPU, peripherals, clocks	Normal operation, processing
Idle Mode	Moderate	Peripherals active, CPU halted	Waits for interrupt to resume processing
Sleep Mode	Low	RAM and timers may stay on, CPU off	Short-term standby
Snooze Mode	Lower than sleep	Selective wake-up sources like UART, ADC	Waits for specific peripheral triggers
Deep Sleep	Very low	RTC, low-power timer, some memory	Long sleep between periodic tasks
Hibernation	Ultra low	MCU state stored in non-volatile memory	Shutdown-like state with long wake time
Shutdown	Minimum	No clock, MCU fully off	System fully off, cold start required

OntarioTech UNIVERSITY

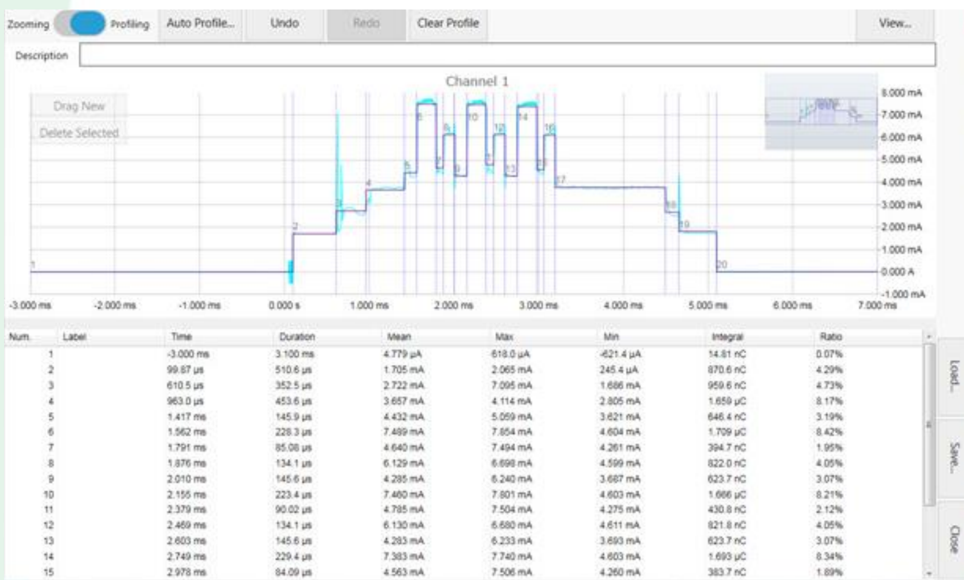
Firmware Choices

- Optimize the MCU clock speed.
  - A MCU current consumption is typically specified in µA / MHz, a slowly-clocked processor consumes less current than one with a fast clock.
- Configure the device display update rate, the frequency at which the MCU receives data from sensors, and the blink rates and duration for LEDs.
- Adjust the frequency at which the MCU turns on the device's radio to transmit data.
- Make sure that sensors and other peripherals are on only when needed, but be sure to remember to allow for sensor power-on stabilization time to avoid inaccurate measurements.

OntarioTech UNIVERSITY

Measuring Current Draws

- It is not clear that we have the right equipment to measure these current draws in the right precision.
- One requires a scope with sufficient bandwidth to measure the device’s highly dynamic signals, and enough dynamic range to measure current from sleep mode (nanoamps) to radio transmission (milliamps).



Num.	Label	Time	Duration	Mean	Max	Min	Integral	Ratio
1		-3.000 ms	3.100 ms	4.779 µA	618.0 µA	-621.4 µA	14.81 nC	0.07%
2		99.87 µs	510.6 µs	1.705 mA	2.065 mA	245.4 µA	870.6 nC	4.29%
3		610.5 µs	352.5 µs	2.722 mA	7.095 mA	1.686 mA	959.6 nC	4.73%
4		963.0 µs	453.6 µs	3.657 mA	4.114 mA	2.805 mA	1.659 µC	8.17%
5		1.417 ms	145.9 µs	4.432 mA	5.059 mA	3.621 mA	646.4 nC	3.19%
6		1.562 ms	228.3 µs	7.489 mA	7.854 mA	4.604 mA	1.709 µC	8.42%
7		1.791 ms	85.08 µs	4.640 mA	7.494 mA	4.261 mA	394.7 nC	1.95%
8		1.876 ms	134.1 µs	6.129 mA	6.698 mA	4.599 mA	822.0 nC	4.05%
9		2.010 ms	145.6 µs	4.285 mA	6.240 mA	3.687 mA	623.7 nC	3.07%
10		2.155 ms	223.4 µs	7.460 mA	7.801 mA	4.603 mA	1.666 µC	8.21%
11		2.379 ms	90.02 µs	4.785 mA	7.504 mA	4.275 mA	430.8 nC	2.12%
12		2.469 ms	134.1 µs	6.130 mA	6.680 mA	4.611 mA	821.8 nC	4.05%
13		2.603 ms	145.6 µs	4.283 mA	6.233 mA	3.693 mA	623.7 nC	3.07%
14		2.749 ms	229.4 µs	7.383 mA	7.740 mA	4.603 mA	1.693 µC	8.34%
15		2.978 ms	84.09 µs	4.563 mA	7.506 mA	4.260 mA	383.7 nC	1.89%

OntarioTech UNIVERSITY

Power Budget

- The IoT architect must build a power budget for the edge device, which includes:
  - Active sensor power
  - Frequency of data collection
  - Wireless radio communication strength and power
  - Frequency of communication
  - Microprocessor or microcontroller power as a function of core frequency
  - Passive component power
  - Energy loss from leakage or power supply inefficiency
  - Power reserve for actuators and motors

Power Budget

- The budget simply reflects the sum of these power contributors subtracted from the source of power (battery).
- Batteries also do not have a linear power behavior over time.
  - As the battery loses energy capacity while discharging, the amount of voltage will drop curvilinearly.
- If the battery drops below a minimum voltage, a radio or microprocessor will not reach the threshold voltage required to transmit / receive.

TI CC-2650 Example

- These are the SensorTag that is used in the labs.
  - Standby mode current: 0.24 mA
  - Running with all sensors disabled (only powering LEDs): 0.33 mA
  - Running with all sensors on at 100 ms/sample data rate and broadcasting: 12.08 mA
    - BLE(Bluetooth Low Energy): 5.5 mA
    - Temperature sensor: 0.84 mA
    - Light sensor: 0.56 mA
    - Accelerometer and gyros: 4.68 mA
    - Barometric sensor: 0.5 mA

OntarioTech UNIVERSITY

TI CC-2650 Example

- The TI SensorTag uses a standard CR2032 coin cell battery rated at 240 mAh.
  - Therefore, the maximum life is expected to be about 44 hours.
- Many power management practices are employed, such as:
  - clock gating components not being used in silicon,
  - reducing the clock rates of processors or microcontrollers,
  - adjusting the sensing frequency and broadcast frequency,
  - backoff strategies to reduce communication strength,
  - and various levels of sleep modes.

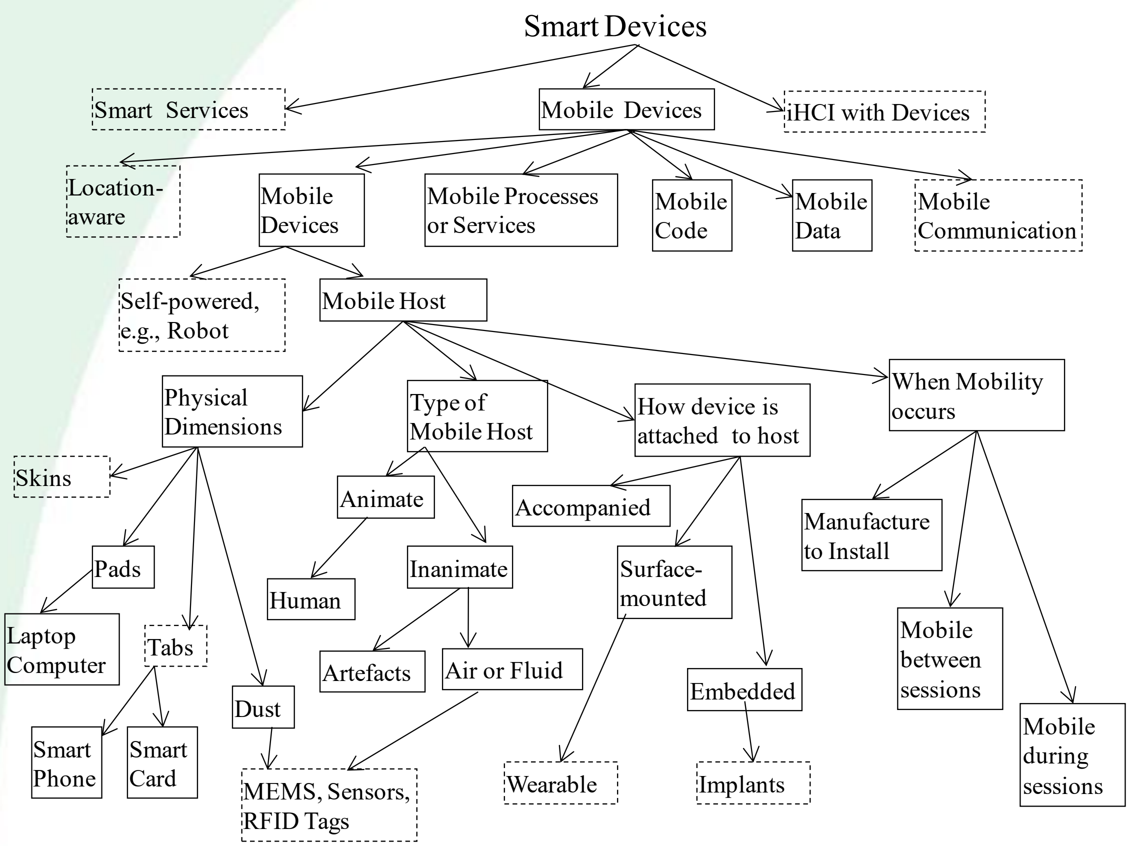
Summary of Sensor Element

- The key concern with sensory devices is that they need to be configured to the specific application and domain.
  - Some OS can abstract these APIs but it is key to understand the operation of a sensor in order to optimize it for the domain of operation.

Smart Device

What constitutes a smart device?

Poslad's View of Smart Devices

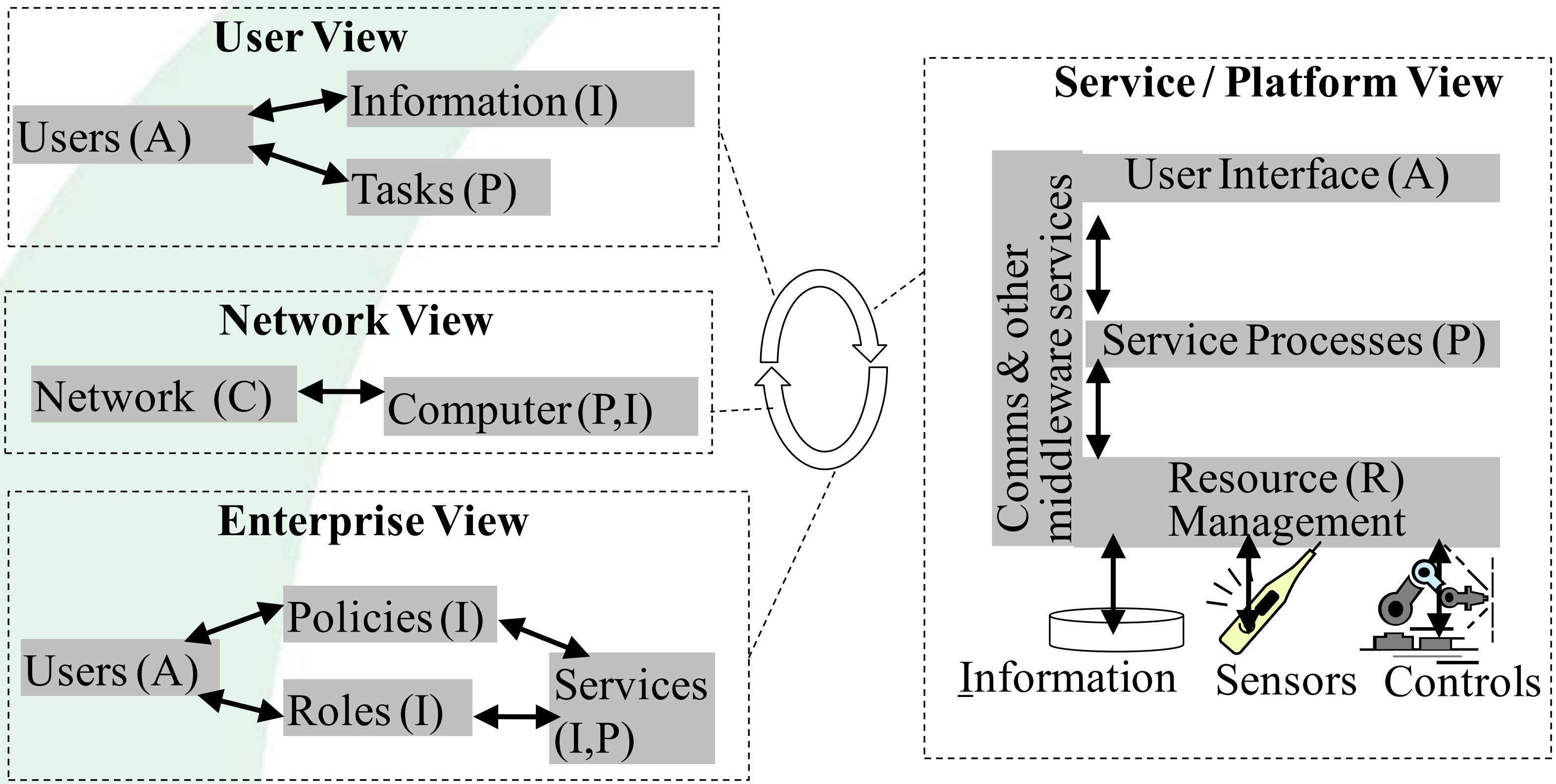


Poslad primarily focuses on Human Computer Smart Devices

Distributed Mobile Services

- Smart devices are inherently distributed in nature and embody user access to distributed system components via an ICT service.
- A distributed mobile system can have different viewpoints based on:
  - Network infrastructure providers,
  - Computer devices,
  - Service infrastructure providers,
  - Individual users and enterprise.

Distributed System Viewpoints



A = Access/presentation, I = Info./data, P = Processing/computation, C=Comms/networking

Abstraction vs Virtualization

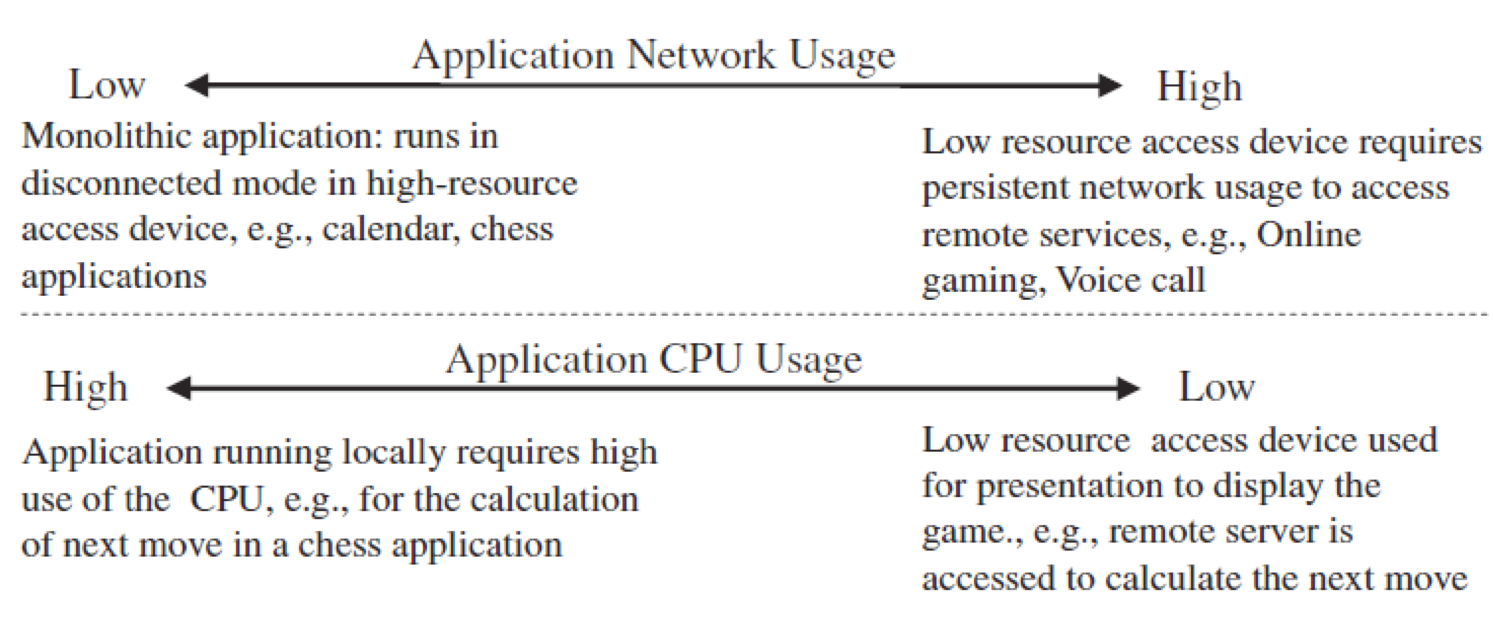
- Abstractions defines those things that are important in a system and to hide or make transparent those properties that are not.
  - *Access transparency* specifies resources that can be accessed from anywhere.
  - *Concurrency transparency* facilitates multiple users or (application) programs operating on shared data without interference between users.
  - *Failure transparency* (Fault Tolerance) enables systems to mask partial failures of a system and availability increases.
  - *Replication transparency* enables users or programs to be unaware that a system uses multiple instances of resources to increase reliability or performance.
  - *Migration transparency* permits resources to move during use.
  - *Scaling transparency* facilitates dynamic resource supply so that it can expand (or contract) to meet demand.

Abstraction vs Virtualization

- Virtualization to the Rescue
  - Supports the ability to map components in one interface at a given level of abstraction into different interfaces and different resources at different levels of abstraction
  - does not necessarily aim to hide and simplify all the details of accessing services,

Service Architecture Models

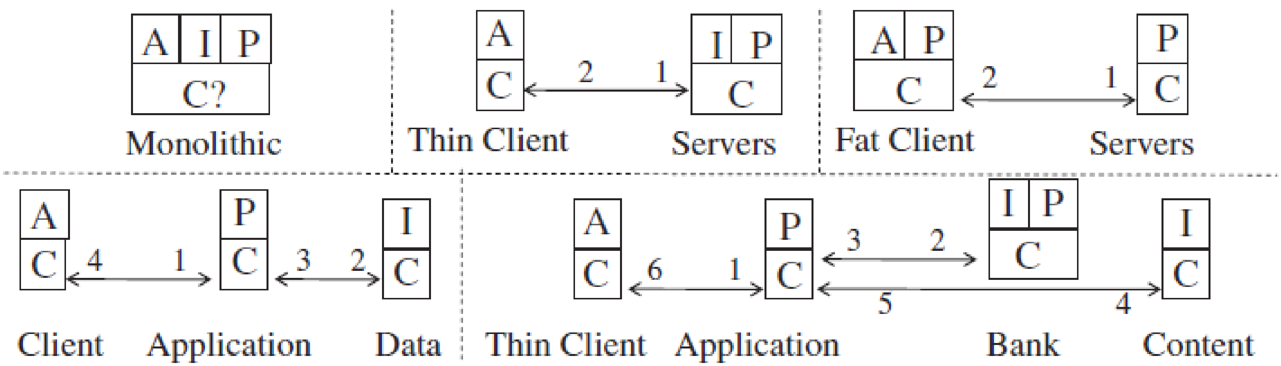
- Balancing local processing and communications
  - balance may need to shift dynamically



OntarioTech UNIVERSITY

Multi-tier Client Service Models

- Different designs for partitioning and distributing Information (I), Processing (P) and Service Access (A) using communication (C).
  - Note: The numbers on the arrow indicate the ordering of the interaction.



OntarioTech UNIVERSITY

Multi-tier Client Service Models

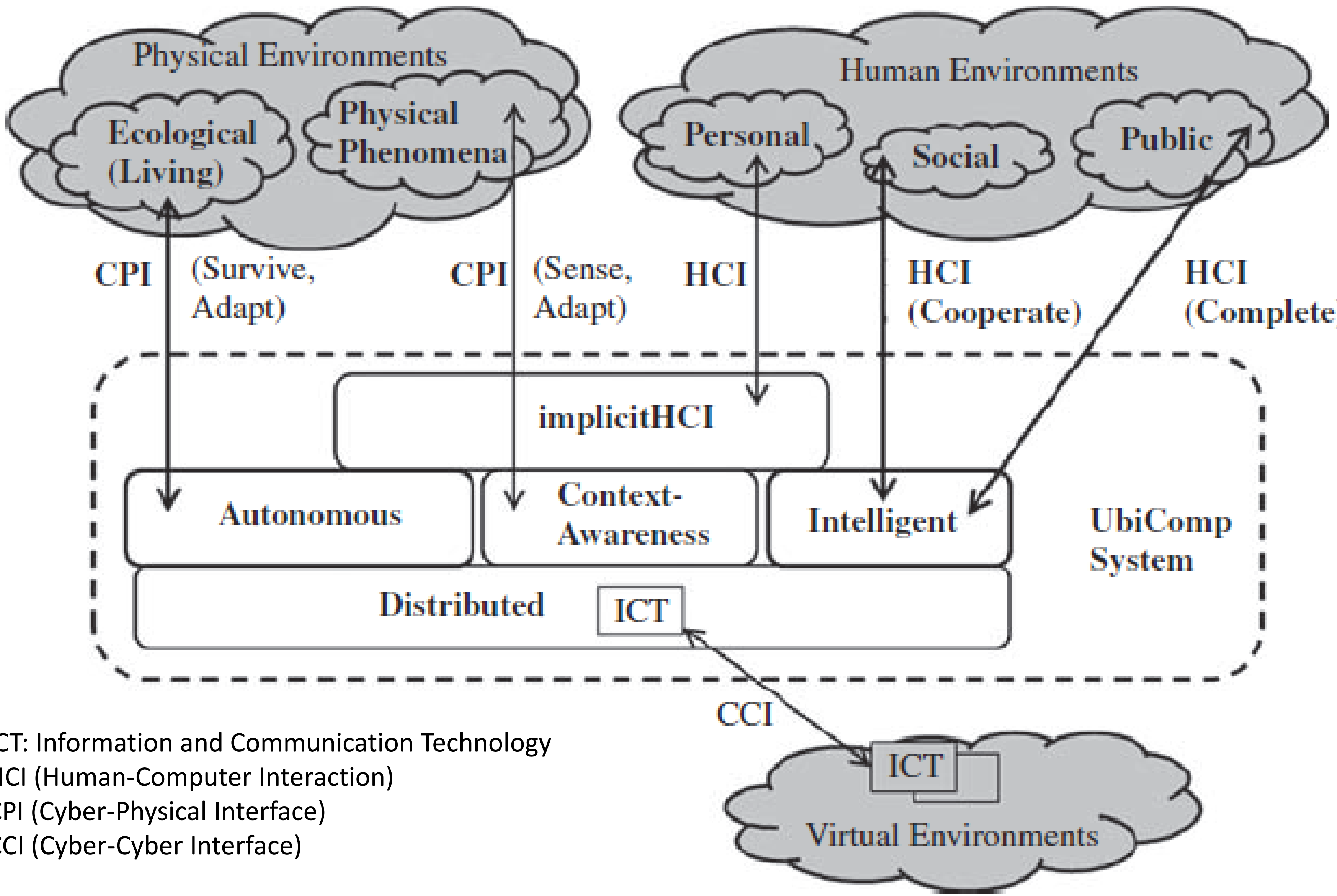
- Three-tier systems
  - The use of single intermediate nodes, are designed that decouple services access from service provision via the use of discovery services
- Four-tier systems
  - application processing and application data are put on separate nodes
- Application
- Services can also be designed to be distributed over five or more tiers depending on the application.

Ontario Tech University

Architectural Design for UbiCom Systems: Smart DEI Model

Ontario Tech University

A Ubiquitous System Model



ICT: Information and Communication Technology
HCI (Human-Computer Interaction)
CPI (Cyber-Physical Interface)
CCI (Cyber-Cyber Interface)

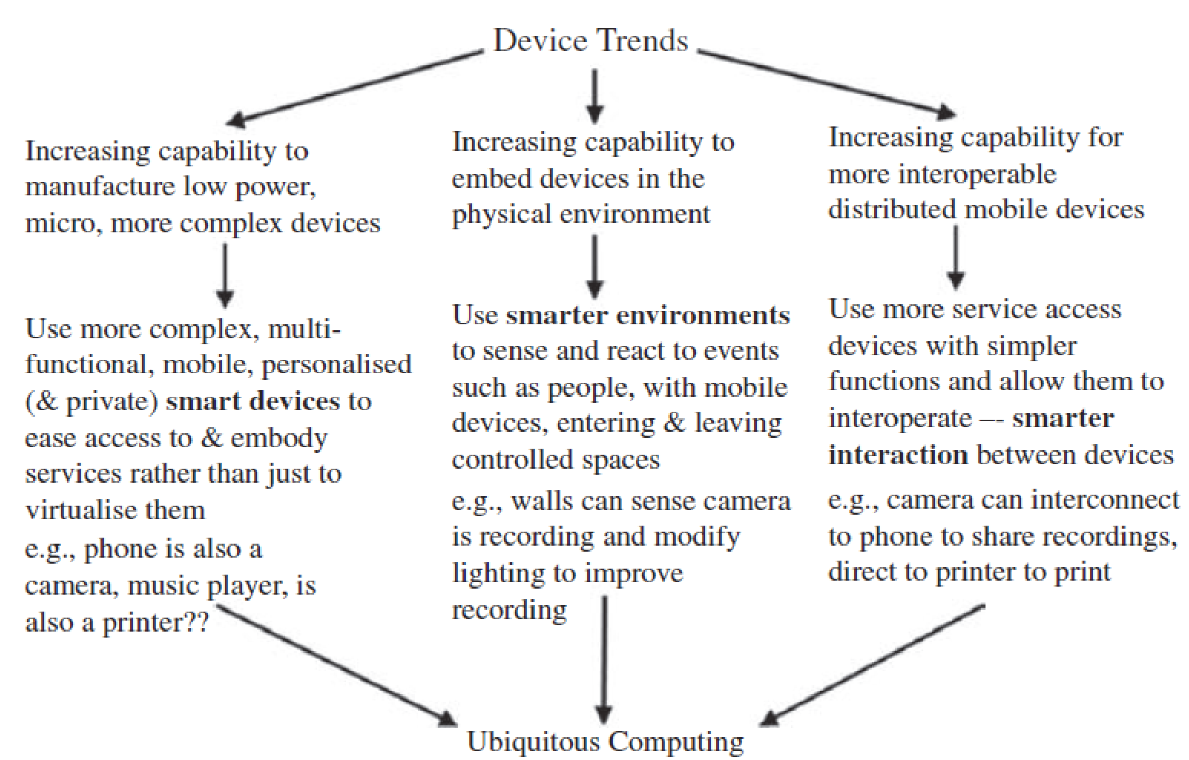
DEI Model

- DEI stands for Smart **Devises**, **Environnements** and **I**nteractions model.
- Smart means that the entity is active, digital, networked, can operate to some extent autonomously, is reconfigurable and has local control of the resources it needs such as energy, data storage, etc.

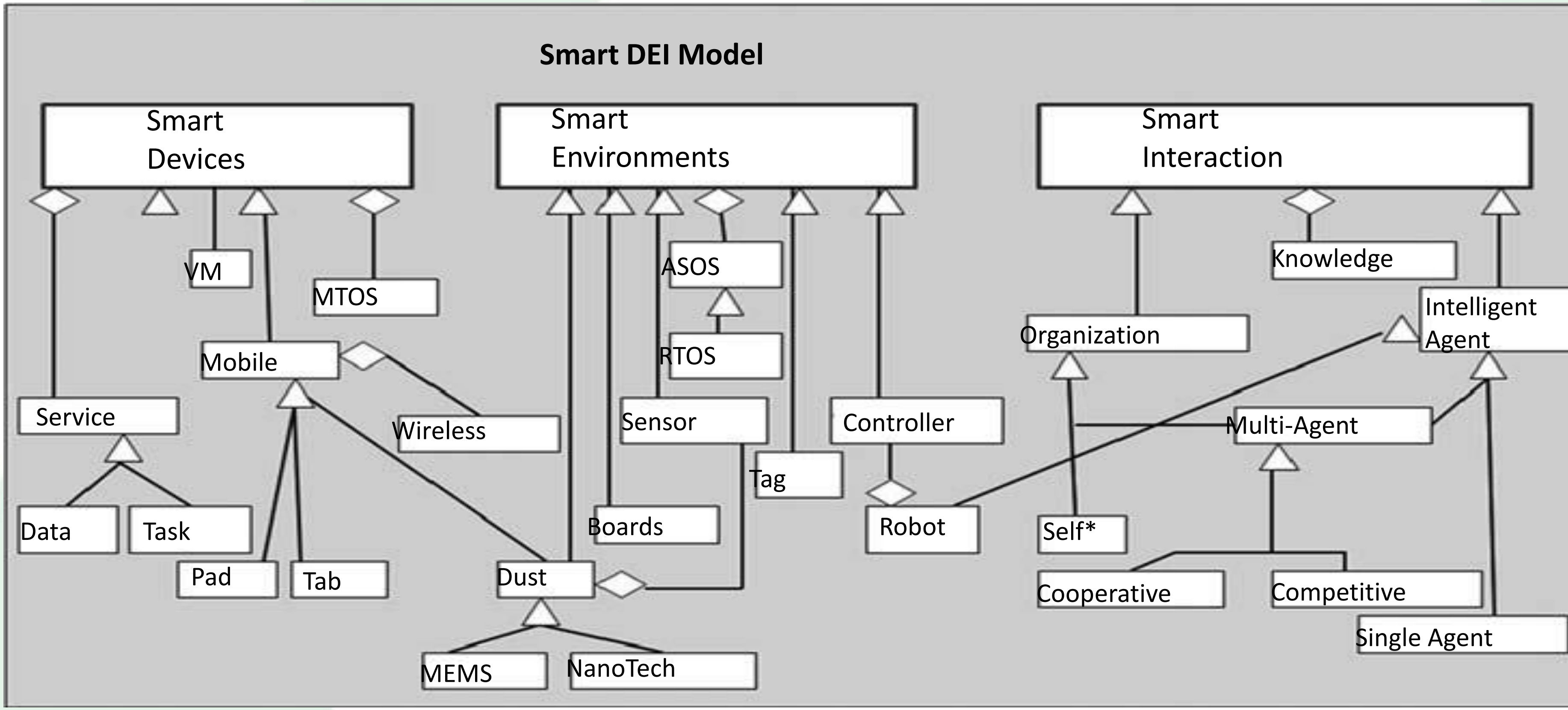
OntarioTech UNIVERSITY

DEI Architectural Design Types

- Basic architectural design patterns for ubiquitous ICT system.



Smart DEI Model



- MTOS: Multi-Tasking Operating System
- VM: Virtual Machine
- ASOS: Application Specific or embedded system OS
- RTOS:Real-Time OS
- MEMS: Micro ElectroMechanical System

Smart Devices

- Characteristics
  - mobility, dynamic service discovery and intermittent resource access (concurrency, upgrading, etc.).
  - Devices are often designed to be multi-functional because these ease access to, and simplify the interoperability of, multi-functions at run-time.
    - However, the trade-off is in a decreased openness of the system to maintain (upgrade) hardware components and to support more dynamic flexible run-time interoperability.

Smart Environments

- "A smart environment is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment." [Cook and Das 2007]
- Characteristics
  - consists of a set of networked devices that have some connection to the physical world, and usually execute a single predefined task.

Smart Interaction

- In smart interaction models, system components dynamically organize and interact to achieve shared goals.
  - Typical of the self* capabilities
- Basic Interaction: Sequencing of messages known in advance. Can be synchronous or asynchronous in nature.
- Smart Interaction: Coordinated, policy and convention-based, dynamic organizational, and semantic and linguistic.

Comparison of smart device, smart environment and smart interaction

Type	Smart Device	Smart Environment	Smart Interaction
Characteristics	Active multi-function devices based in a virtual computing environment	Active single function devices embedded or scattered in a physical environment	Individual components that must cooperate or compete to achieve their goals
System environment interaction	Weak CPI, strong H2C, weaker C2H and strong C2C	Strong C2P and C2H	Rich H2H, P2P models that apply to HCI and CPI
Dynamic services	Dynamic ICT service, resource discovery	Dynamic physical resource discovery	Dynamic composition of entities and services
Context-awareness	Low-medium	High	Low-medium

OntarioTech UNIVERSITY

Comparison of smart device, smart environment and smart interaction

Type	Smart Device	Smart Environment	Smart Interaction
HCI: locus of control	Localized in ICT device	Localized in part(s) of Physical World	Distributed in physical and virtual world
Autonomy	Autonomous control of local ICT resources, less autonomous control of remote services	Autonomous control of local ICT resources	High autonomy of actions and interaction
Intelligence	Low to medium individual rational intelligence	Low to medium individual rational intelligence	High collective intelligence: semantic sharing, social cooperation and competition

Alternative DEI Model