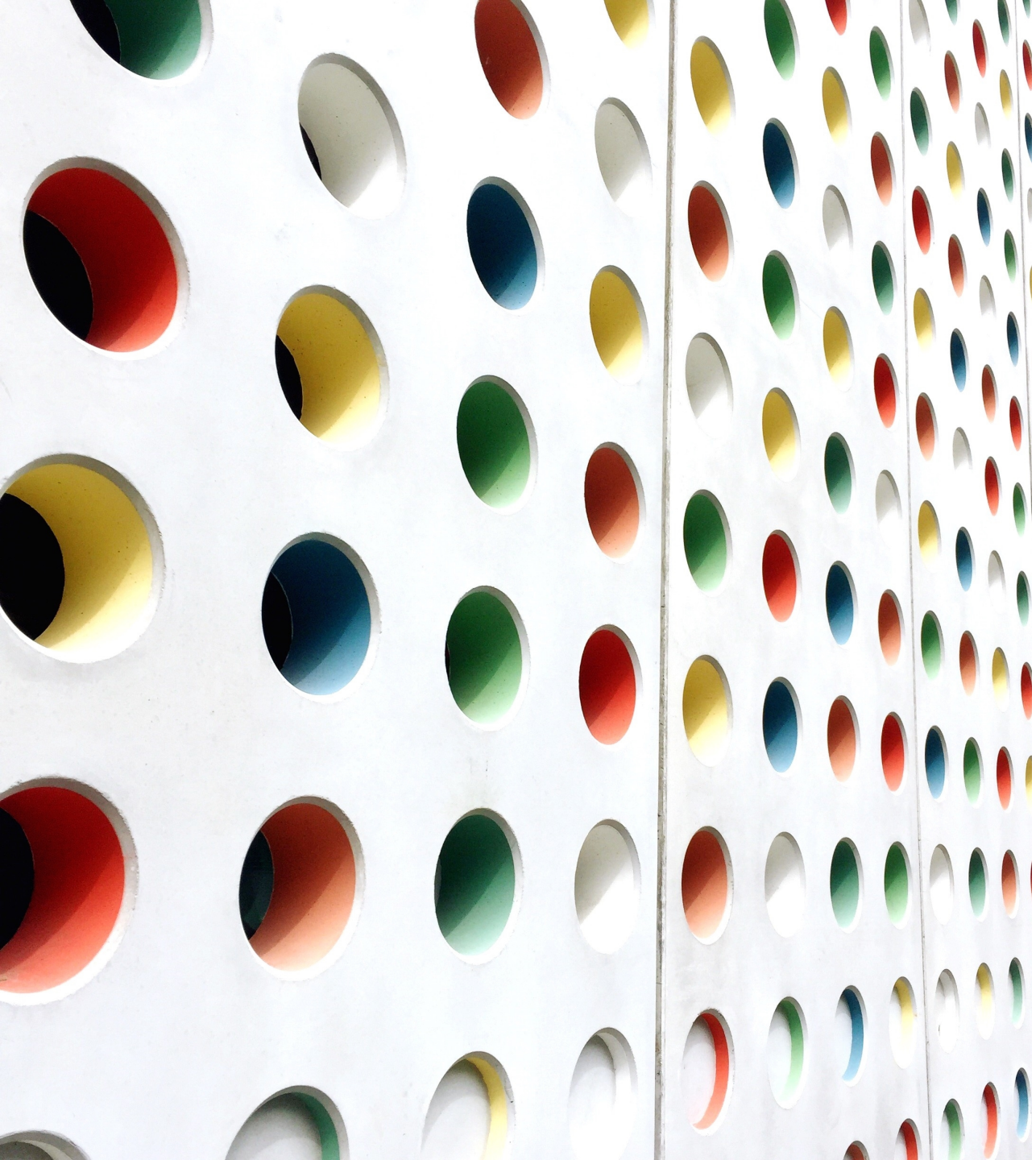


User Experience design for IoT

MS.T.KAVITHA
ASSISTANT PROFESSOR
CSE, KITS

What's Different About User Experience Design for the Internet of Things?

- **Many of our interactions now take place on mobile phones, tablets, e-readers, and smart TVs. And it's common to use one service across multiple devices with different form factors**

BBC iPlayer can be used on connected TVs, smartphones, tablets , PCs, game consoles, and set-top boxes

What's Different About User Experience Design for the Internet of Things?

- But our experience of computing has changed radically in the past 10–15 years
- User experience (UX) design and human–computer interaction (HCI) emerged in a world of desktop computers.
- A variety of scales, from tiny wrist-tops, to smartphones, to TV user interfaces (UIs) viewed from 10 feet away.
- The "Internet of Things" (IoT) refers to the growing range of everyday objects acquiring connectivity, sensing abilities, and increased computing power.

What's Different About User Experience Design for the Internet of Things?

In consumer terms, some common categories currently include: •

- Connected home technology (such as thermostats, lighting, and energy monitoring) •
- Wearables (such as activity/fitness trackers and "smart" watches) •
- Medical/wellness devices (such as bathroom scales and blood pressure monitors) •
- Connected cars (which may provide access to smartphone apps via dashboard controls, engine diagnostics, and automatic alerting of authorities in case of a crash) •
- Urban systems (such as air quality sensors, city rental bikes, and parking meters/sensors)

User Experience Design

we mean by “UX” and “user experience design.”

UX is a holistic term referring to a wide range of design disciplines involved in creating systems that are useful, usable, and pleasurable to use.

How Is UX Different for IoT?

Designing for IoT comes with a bunch of challenges that will be new to designers accustomed to pure digital services.

How tricky these challenges prove will depend on:

- **The maturity of the technology you're working with**
- **The context of use or expectations your users have of the system**
- **The complexity of your service (e.g., how many devices the user has to interact with)**

Functionality can be Distributed Across Multiple Devices with Different Capabilities

IoT devices come in a wide variety of form factors with varying input and output capabilities.

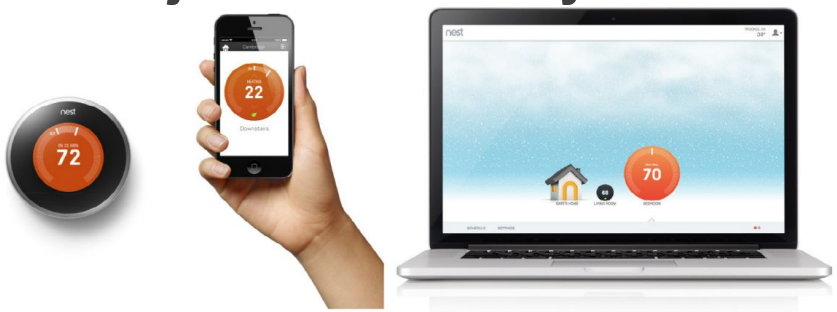
- Some may have screens, such as heating controllers or washing machines.
- So ways of communicating with us EDs or sounds.

The Honeywell evohome connected radiator valve has a basic LCD screen

The GlowCaps connected pill bottle lid uses light and sound notifications to remind the user to take medication

Functionality can be Distributed Across Multiple Devices with Different Capabilities

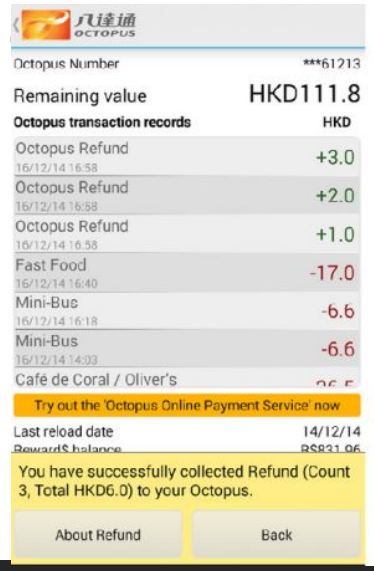
- Some may have no input or output capabilities at all and are unable to tell us directly what they are doing. Interactions may be handled by web or smartphone apps.
- Despite the differences in form factors, users need to feel as if they are using a coherent service rather than a set of disjointed UIs.
- It's important to consider not just the usability of individual UIs but interusability : distributed user experie



The Nest Learning Thermostat can be controlled by the on-device UI, a smartphone app, or a web app

The Focus Of The User Experience May Be In The Service

- The service around a connected device is often just as critical in delivering the user experience, if not more so, than the device itself.
- For example, the smart travelcards such as the London Oyster and Hong Kong Octopus are often thought of as the focus of the payment service. But the services can be used without a card at all via an NFC(Near Field ard.



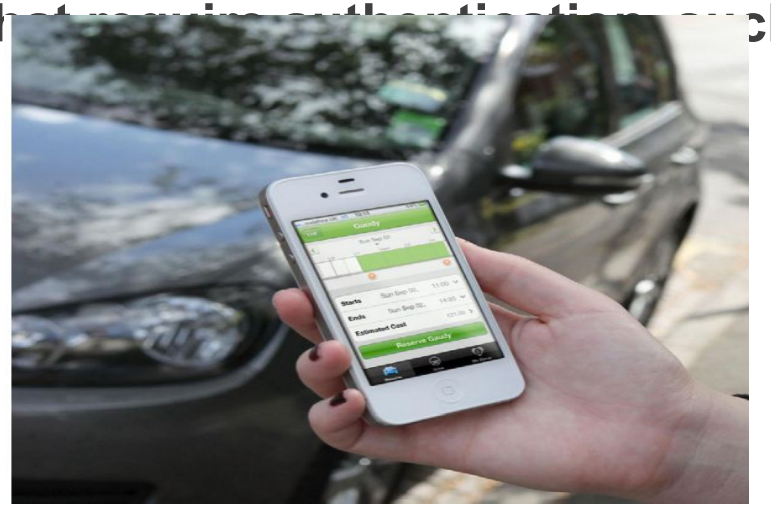
Hong Kong's Octopus payment service can be used with an NFC phone as well as a smart card

We Don't Expect Internet-like Glitches From The Real World

- It's frustrating when a web page is slow to download or a Skype call fails. But we accept that these irritations are just part of using the Internet.
- By contrast, real-world objects respond to us immediately and reliably. When we interact with a physical device over the Internet, that interaction is subject to the same latency and reliability issues as any other
- Internet communication. So there's the potential for delays in response and for our requests and commands to go missing altogether. This could make the real world start to feel very broken. Imagine if you turned your lights on and they took two minutes to respond, or failed to come on at all.

We Don't Expect Internet-like Glitches From The Real World

- In theory, there could be other unexpected consequences of things adopting Internet-like behaviors.
- In the Warren Ellis story "The Lich House"4a woman is unable to shoot an intruder in her home: her gun cannot contact the Internet for the authentication that would allow her to fire it. This might seem far-fetched, but we already have objects th[illegible]ch as Zipcars



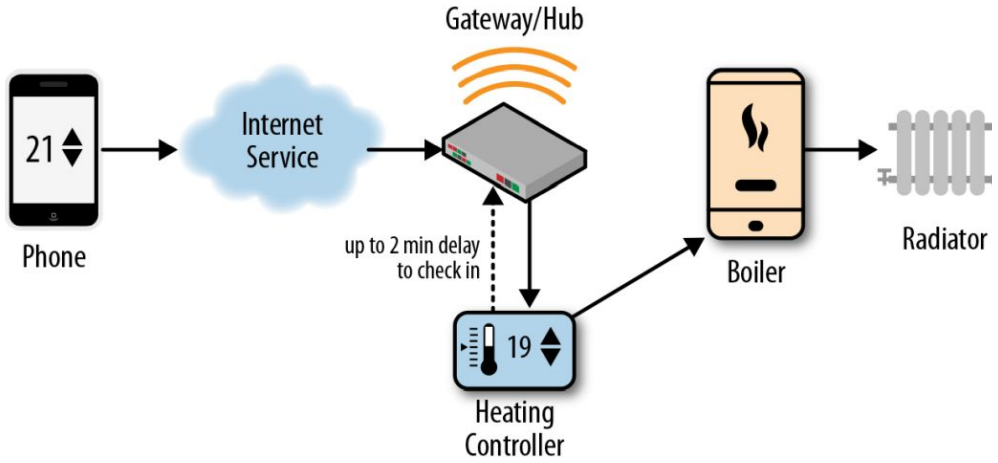
When you book a Zipcar online, the service sends details of the reservation to the car; swiping a smart card authenticates you as the person who made the booking

IoT is Largely Asynchronous

- When we design for desktops, mobiles, and tablets, we tend to assume that they will have constant connectivity. Well-designed mobile apps handle network outages gracefully, but tend to treat them as exceptions to normal functioning.
- We assume that the flow of interactions will be reasonably smooth, even across devices. If we make a change on one device it will quickly propagate across any other devices we use with the same service.
- That will not always happen in IoT systems. Many connected devices run on batteries, and need to conserve electricity. Maintaining network connections uses a lot of power, so they only connect intermittently. This means that parts of the system can be out of sync with one another, creating discontinuities in the user experience.

IoT is Largely Asynchronous

- If your heating is set to 19° C, and you use the heating app on your phone to turn it up to 21° C, it will take a couple of minutes for your battery-powered heating controller to go online to check for new instructions. During this time, the phone says 21° C, and the controller say 19° C.

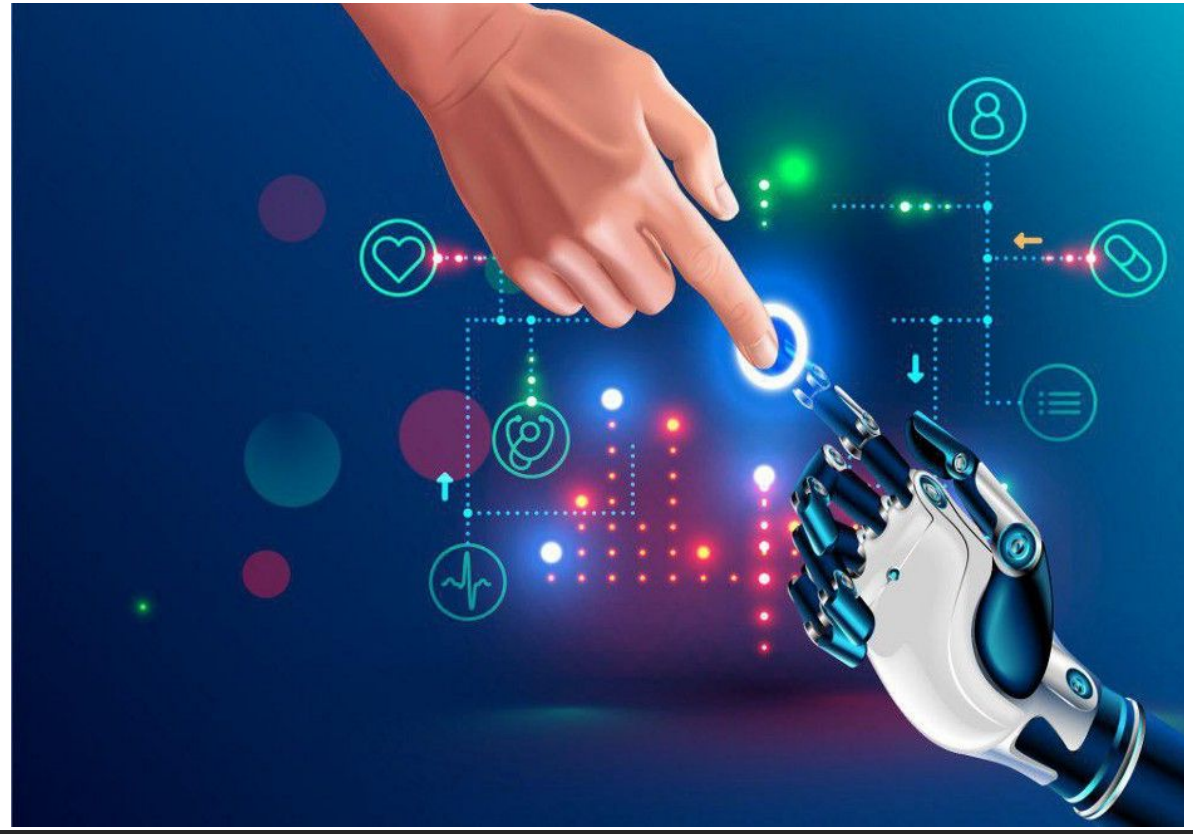


Schematic of heating system with app and controller giving different status information

- These discontinuities won't always be noticed: sometimes the delays will be very short, and sometimes users won't be around to notice them

Video on IoT and ML

https://www.youtube.com/watch?v=mlE03Fj2T9s

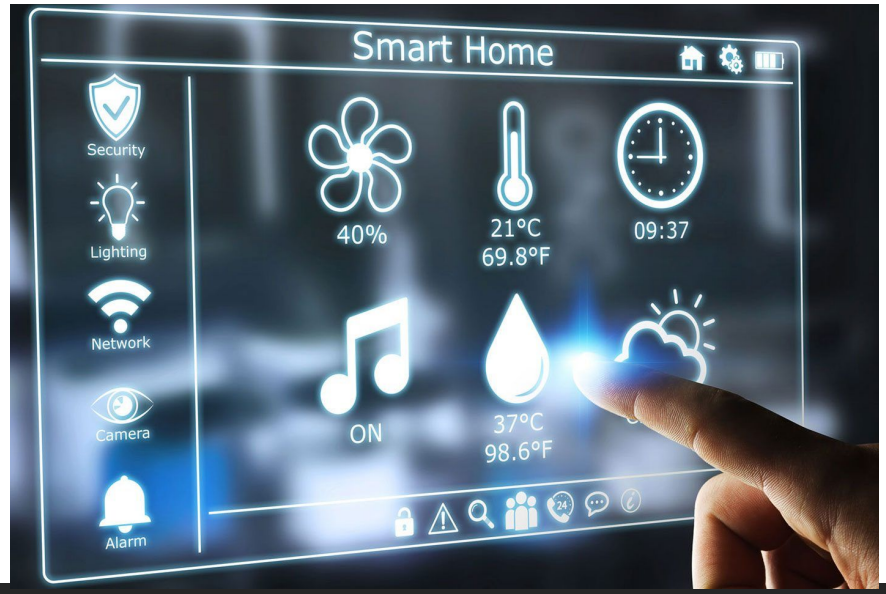


Code Can Run in Many More Places

- The configuration of devices and code that makes a system work is called the system model. In an ideal world, users should not have to care about this.
- We don't need to understand the technical architecture of a conventional Internet service, like Amazon, in order to use it successfully.
- Instead, we form a conceptual model of what Amazon does and how it works that's good enough to help us understand what to do. We know we can search or browse products, that we need to add them to a basket, set up or log into a user account, pay, and then we'll get a delivery.
- We don't need to understand the different machines involved in making this system function. But as a consumer of an IoT service right now, you can't always get away from some of this technical detail.

A typical IoT service is composed of:

- One or more embedded devices (the things in IoT)
- An Internet service • Perhaps a gateway device (a separate device needed to connect some embedded devices to the Internet)
- One or more mobile or web apps for the user to interact with the service via a mobile, tablet, or PC



Example – Smart Home

Example – IoT System Failure

- Imagine you have a connected lighting system in your home. It has controllable bulbs or fittings, perhaps a gateway that these connect to, an Internet service, and a smartphone app to control them all.
- You have an automated rule set up to turn on some of your lights at dusk if there's no one home. If your home Internet connection goes down, does that rule still work? If the rule runs on the Internet service or your smartphone, it won't. If it runs on the gateway, it will.
- As a user, you want to know whether your security lights are running or not. You need to understand a little about the system model to understand which devices are re ality, and how the system may fail.

The Philips Hue system consists of connected bulbs, a gateway, an Internet service, and a smartphone app

Devices are Distributed in the Real World

- The shift from desktop to mobile computing means that we now use computers in a wide variety of situations. Hence, mobile design requires a far greater emphasis on understanding the user's needs in a particular context of use. IoT pushes this even further: computing power and networking is embedded in more and more of the objects and environments around us.
- For example, a connected security system can track not just whether the home is occupied, but who is inside, and potentially video record them. Hence, the social and physical contexts in w ces and services can be used are even more complex and va

Remote Control and Automation are Programming-like Activities

- In 1982, the HCI researcher Ben Shneiderman defined the concept of direct manipulation. *User interfaces based on direct manipulation "depend on visual representation of the objects and actions of interest*, physical actions or pointing instead of complex syntax, and rapid incremental reversible operations whose effect on the object of interest is immediately visible.
- This strategy can lead to user interfaces that are *comprehensible, predictable and controllable*. Ever since, this has been the prevailing trend in consumer UX design.
- Direct manipulation is successful because interface actions are aligned with the user's understanding of the task. They receive immediate feedback on the consequences of their actions, which can be undone.
- Unlocking the door remotely is an easier action to comprehend. But we are distanced from the consequences of our actions, and this poses other challenges. Can we be sure the door was locked again once the parcel had been left? A good system should send a confirmation, *but if our smartphone (or the lock) lost connectivity, we might not receive this(disadvantage).*

Complex Services Can Have Many Users, Multiple UIS, Many Devices, Many Rules And Applications

- A simple IoT service might serve only one or two devices (e.g., a couple of connected lights). You could control these with a very simple app. But as you add more devices, there are more ways for them to coordinate with one another.
- If you add a security system with motion sensors and a camera, you may wish to turn on one of your lights when the alarm goes off. So the light effectively belongs to two functions or services: security and lighting.
- Then add in a connected heating system that uses information from the security system to know when the house is empty. And assume that there are several people in the house with slightly different access privileges to each system.

Complex Services Can Have Many Users, Multiple UIS, Many Devices, Many Rules And Applications

- For example, some can change the heating schedule, some can only adjust the current temperature. Some have admin rights to the security system, some can only set and unset the alarm. What started out as a straightforward system has become a complex web of interrelationships.
- For a user, understanding how this system works will become more challenging as more devices and services are added. It will also become more time consuming to manage.

Many Differing Technical Standards Make Interoperability Hard

---

As we begin to connect these devices together, this lack of common technology standards is causing headaches. Just getting devices talking to one another is a big enough challenge, as there are *many different network standards*. Being able to get them to coordinate in sensible ways is vastly more complicated still.

Many Differing Technical Standards Make Interoperability Hard

The consumer experience right now is of a selection of mostly closed, manufacturer-specific ecosystems. Devices within the same manufacturer's ecosystem, such as Withings, will work together. But this is the only given.

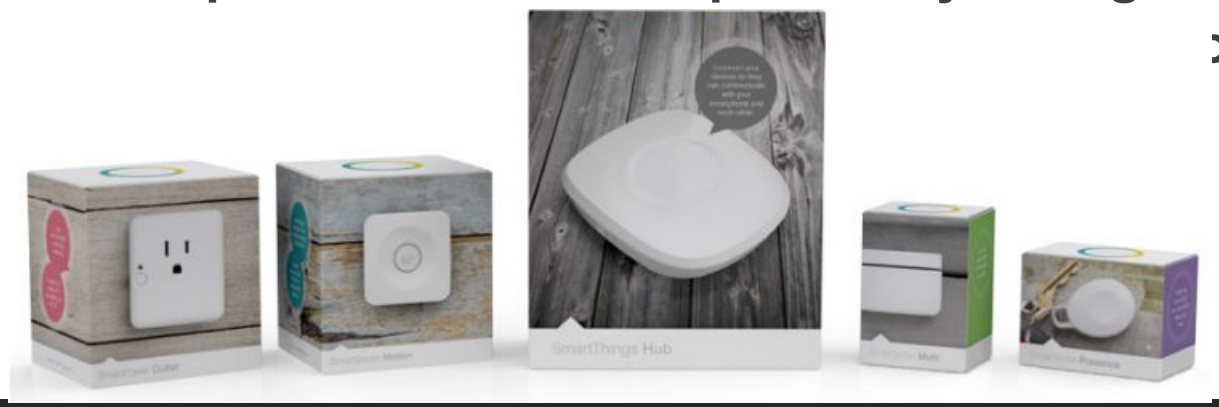
In the case of Withings, this means that devices share data with a common Internet service, which the user accesses via a smartphone app. Apple's Airplay is an example of a proprietary ecosystem in which devices talk directly to one another.

The Withings ecosystem of devices

Many Differing Technical Standards Make Interoperability Hard

There are also some emerging platforms that seek to aggregate devices from a number of manufacturers and enable them to interoperate. The connected home platform SmartThings supports a range of network types and devices from manufacturers such as Schlage and Kwikset (door locks); GE and Honeywell (lighting and power sockets); Sonos (home audio); and Philips Hue, Belkin, and Withings (connected home products);

But the platform has been specifically configured to work with each of these. You ork well with SmartThings.

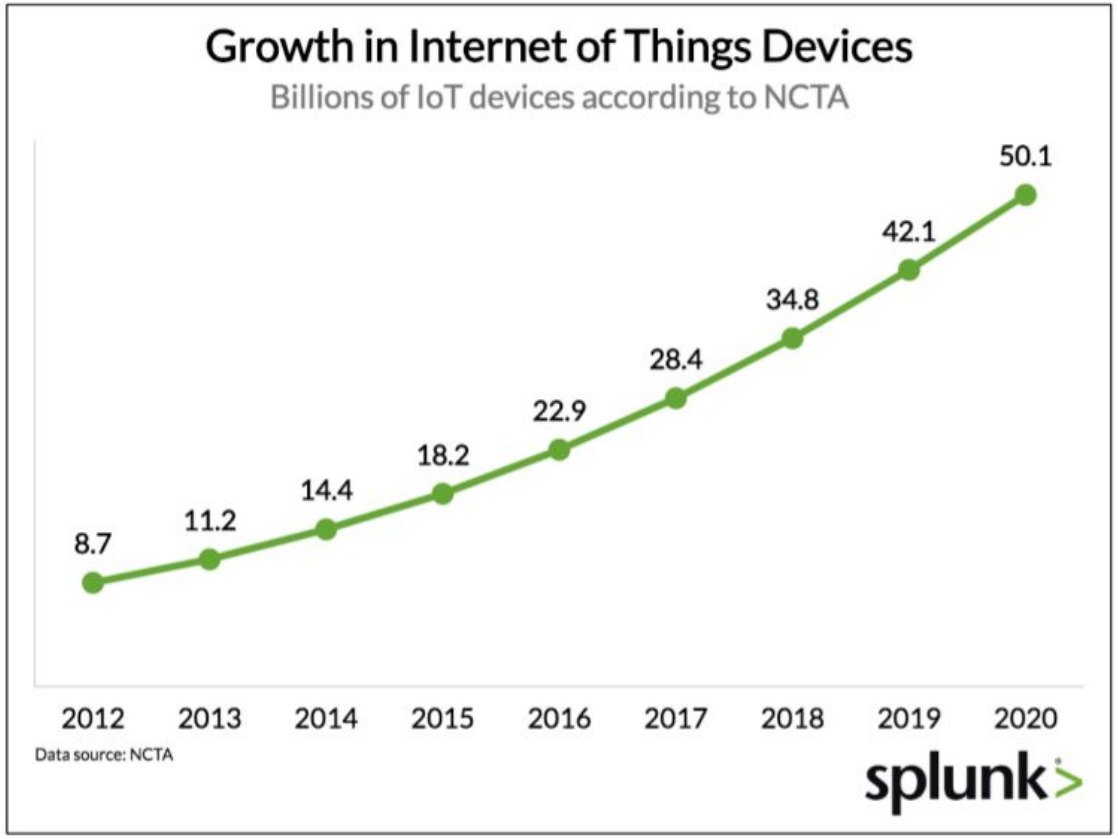


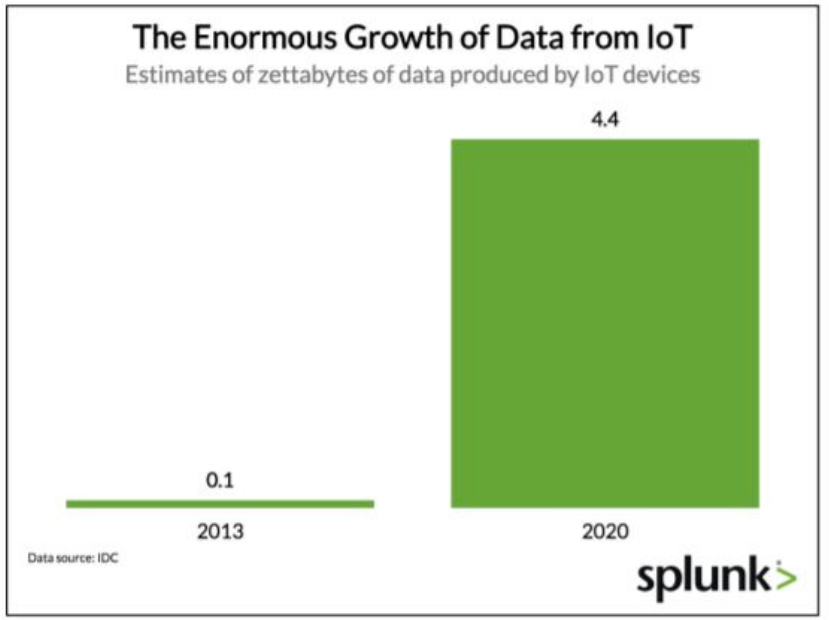
The SmartThings gateway and some compatible devices

IoT is All About Data

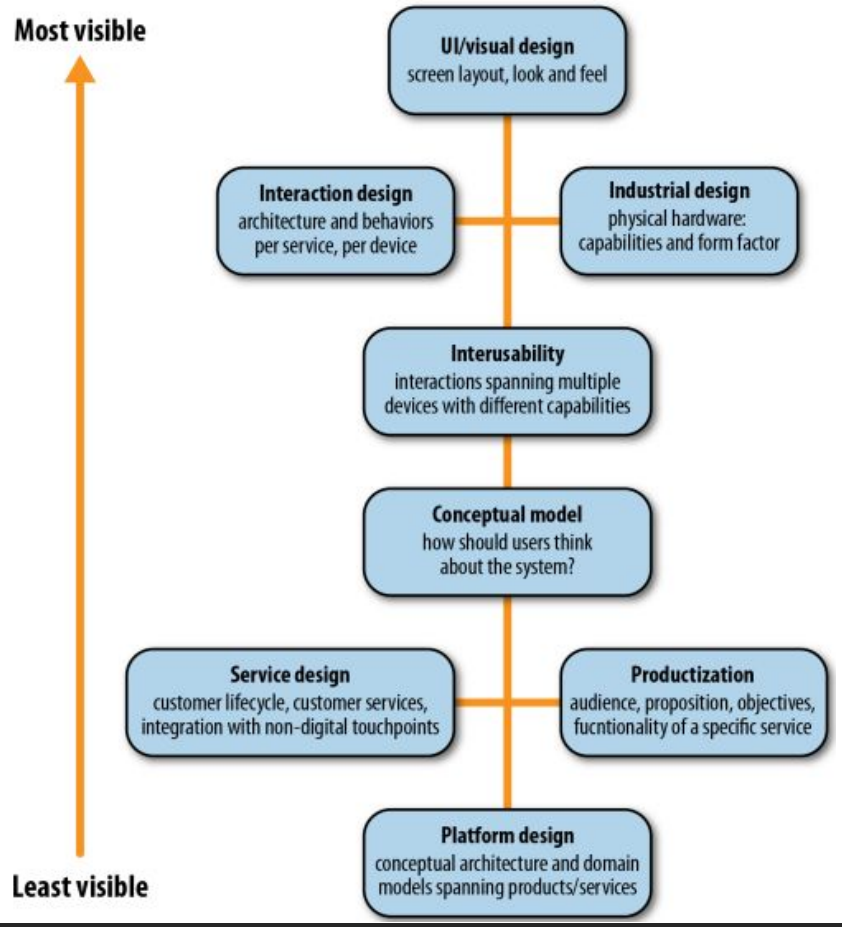
- Networked, embedded devices allow us to capture data from the world that we didn't have before, and use it to deliver better services to users. For example, drivers looking for parking spaces cause an estimated 30% of traffic congestion in US cities.
- Smart parking applications such as Streetline's Parker use sensors in parking spaces to track where spaces are free, for drivers to find via a mobile app.
- The software company Opower analyzes data from smart meters to suggest ways in which utility customers could save energy and money

IOT IS ALL ABOUT DATA





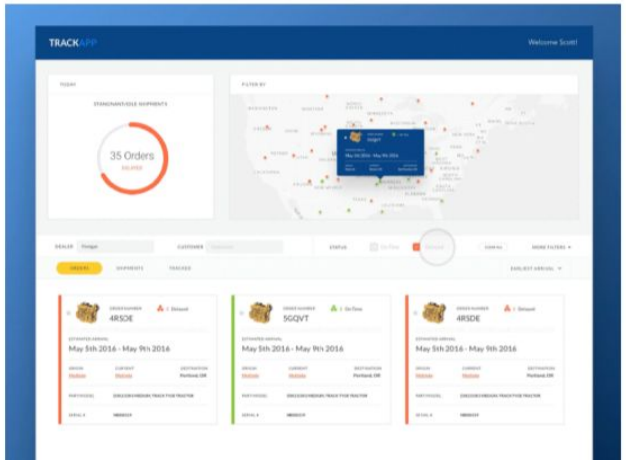
IoT is all About Data



Facets of design in IoT—a good product requires integrated thinking across all of these

UI/VISUAL DESIGN

UI/visual design refers to screen layout, visual styling, and look and feel on a device.



INTERACTION DESIGN

- Interaction design is the design of device behaviours.
- Interaction designers shape the sequences of actions between the user and the device needed to achieve particular goals or activity.
- They also determine how to organize the user-facing functions of the device.
- Interaction design is closely aligned to UI design in the sense that the two are usually done in tandem and often by the same people.
- Interaction design is primarily concerned with behaviours and actions, whereas UI/visual design is concerned with layout and aesthetics.

INTERUSABILITY

- Interusability refers to the additional considerations of designing interactions that span multiple devices.
- The goal is to make the overall experience feel like a coherent service, even when the devices involved may have quite different form factors and input/output capabilities.

INDUSTRIAL DESIGN

- **Industrial design refers to the aesthetic and functional design of the physical hardware in the service: the choice of form, materials, and capabilities it may have.**

SERVICE DESIGN

Service design is an emerging discipline that addresses this holistic view of user experience. It looks at the whole lifespan of a user's experience with a service, provides a view of all the components of the user experience, and specifies how these function together as a coherent whole.

It might include:

- **Customer support interactions**
- **Instructional guides**
- **Marketing or sales materials**
- **In-store experiences**
- **Email communications and notifications**
- **The UX of software updates and rolling out new functi**

CONCEPTUAL MODEL

The conceptual model is the understanding and expectations you want the user to have of the system.

- **Enables users to figure out how to interact with your service.**
- **You can create a clear conceptual model through careful system and interaction design and supporting documentation.**
- **They should feel confident that they will be able to use the system, even if they don’t understand all the details yet.**

PRODUCTIZATION

- Productization is the activity of defining a compelling product proposition.
- It addresses the audience, proposition, objectives, and overall functionality of service
- Does your product solve a real problem for a real audience?
- Is it presented so that they understand that?
- Does it appeal to them?

PLATFORM DESIGN

A platform is a software framework.

- **It takes care of low-level details to help developers build applications more easily.**
- **For example, a platform like Hue or Withings may provide standard ways to:**
  - **Discover new devices and applications**
  - **Add devices and applications onto the system**
  - **Manage devices and users**
  - **Manage how devices share data**
  - **But the system won't know that unless this kind of logic is encoded in the platform**

Summary

Embedded devices often save power by connecting only intermittently	... which means parts of the system can be out of sync, creating discontinuities in UX
Latency on the Internet is out of your control (and reliability is not 100%)	... which means although we expect physical things to respond immediately and reliably, this might not happen
Code can run in many more places	... which means users have to engage with the system model to predict how it will work if parts are offline
Devices are distributed in the real world	... which means social and physical context of use is complex and varied
Functionality can be distributed across multiple UIs	... which means designers need to consider not just usability but interusability
Much of the information processing happens in the Internet service	... which means the service experience is often equally or more important than the single device UX
Remote control and automation are programming-like activities	... which means IoT breaks direct manipulation—the basis of most successful consumer UXes
Many differing technical standards	... which means getting things to work together is hard
Complex services can have many users, many UIs, many devices, many rules and applications	... which means understanding and managing how they all interrelate can be extremely difficult. Users will turn off if admin becomes too onerous.
IoT enables us to capture and act on data we didn't have before	... which means designers need to understand how to use information as a design material

- ✔ Qualitative Assessment – 1
- ✔ Give your Idea presentation by presenting new solutions using IoT Technology