iot for all

LEVEREGE

IoT 101

An Introduction to the Internet of Things

First Edition, © 2018 Leverege LLC

Table of Contents

Foreword

There's quite a bit of confusion around the Internet of Things (IoT).

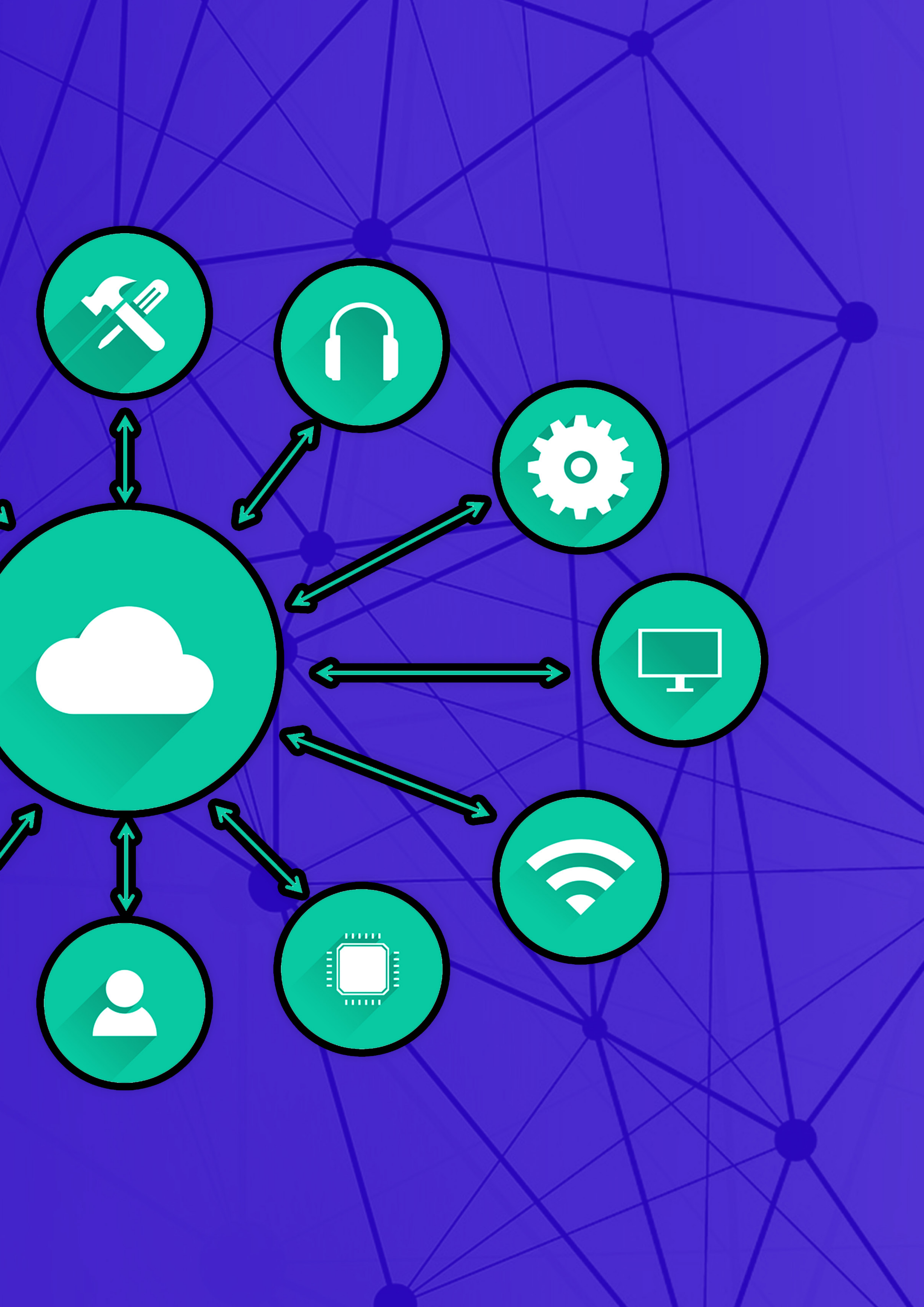
What is it exactly? Is it something that my business or organization needs to use? If so, how? What are the use cases? The risks? How do I get started?

Because IoT is such a large concept and there has been so much noise associated with it in the past few years, it's easy to dismiss it all as hype. But make no mistake, IoT is a powerful, long-term approach that every business will need to leverage to succeed in the future.

We won't be able to answer all the questions posed above, but this ebook is intended to equip you with a solid foundation in the Internet of Things and its accompanying concepts, components, and the technologies that make it all possible. You'll have a much better handle on what IoT means and how you might be able to use it to build a new organization, to launch a new business line within your existing organization, or to simply improve your internal processes and operations.

We firmly believe that everyone is made better by openly sharing knowledge, so we've created this guide for you. We've seen firsthand that many organizations don't know enough to even begin asking the right questions, and so we hope to narrow that education gap.

So if you still have questions after reading this eBook, please don't hesitate to ask us. And without further ado, let's get into it!



1 Introduction

"What is IoT?"

When you Google "what is IoT," many of the answers are unnecessarily technical. Case in point:

> ***"The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction."***
>
> – An unnecessarily technical explanation of IoT

If you just read that and thought, **"ok, so what?"**, you're not alone. Most people don't want to nor need to dive into the nitty-gritty of IoT. In this chapter, we'll provide you with a simple explanation of the Internet of Things and what it means for you.

Before we jump in, note that "The Internet of Things" and "IoT" can and will be used interchangeably. And a quick tip to sound knowledgeable: avoid saying "the IoT".

A Simple, Non-Technical Explanation of IoT

How are you reading this ebook right now? It might be on desktop, on mobile, maybe a tablet, but whatever device you're using, it's most likely connected to the internet.

An internet connection is a wonderful thing, it give us all sorts of benefits that just weren't possible before. If you're old enough, think of your cell phone before it was a smartphone. You could call and you could text, sure, but now you can read any book, watch any movie, or listen to any song all in the palm of your hand.

The point is that connecting things to the internet yields many amazing benefits. We've all seen these benefits with our smartphones, laptops, and tablets, but this is true for everything else too. And yes, we do mean everything.

The Internet of Things is actually a pretty simple concept. It means taking all the physical places and things in the world and connecting them to the internet.

Confusion arises not because the concept is so narrow and tightly defined, but rather because it's so broad and loosely defined. It can be hard to nail down the concept in your head when there are so many examples and possibilities in IoT.

To help clarify, it's important to understand the benefits of connecting things to the internet. Why would we even want to connect everything to the internet?

Why IoT Matters

When something is connected to the internet, that means that it can send information or receive information, or both. This ability to send and/or receive information makes things "smart."

Let's use **smart**phones again as an example. Right now you can listen to just about any song in the world, but it's not because your phone actually has every song in the world stored on it. It's because every song in the world is stored somewhere else, but your phone can send information (asking for that song) and then receive information (streaming that song on your phone).

To be smart, a thing doesn't need to have super storage or a super computer inside of it - it just needs access to it. In the Internet of Things, all the things that are being connected to the internet can be put into three categories:

1. Things that collect information and then send it
2. Things that receive information and then act on it
3. Things that do both

And all three of these have enormous benefits that compound on each other.

1. Collecting and Sending Information

Sensors could be temperature sensors, motion sensors, moisture sensors, air quality sensors, light sensors, you name it. These sensors, along with a connection, allow us to automatically collect information from the environment which, in turn, allows us to make more intelligent decisions.

On a farm, automatically getting information about the soil moisture can tell farmers exactly when their crops need to be watered. Instead of watering too much (which can be an expensive over-use of irrigation systems) or watering too little (which can be an expensive loss of crops), the farmer can ensure that crops get exactly the right amount of water. This enables farmers to increase their crop yield while decreasing their associated expenses.

Just as our sight, hearing, smell, touch, and taste allow us, humans, to make sense of the world, sensors allow machines (and the humans monitoring the machines) to make sense of the world.

2. Receiving and Acting on Information

We're all very familiar with machines getting information and then acting. Your printer receives a document and it prints it. Your car receives a signal from your car keys and the doors open. The examples are endless. Whether it's as simple as sending the command "turn on" or as complex as sending a 3D model to a 3D printer, we know that we can tell machines what to do from far away. So what?

The real power of the Internet of Things arises when things can do both of the above. Things that collect information and send it, but also receive information and act on it.

3. Doing Both: The Goal of an IoT System

Let's quickly go back to the farming example. The sensors can collect information about the soil moisture to tell the farmer how much to water the crops, but you don't actually need the farmer. Instead, the irrigation system can automatically turn on as needed based on how much moisture is in the soil.

You can take it a step further. If the irrigation system receives information about the weather from its internet connection, it can also know when it's going to rain and decide not to water the crops today because they'll be watered by the rain anyways.

And it doesn't stop there! All this information about the soil moisture, how much the irrigation system is watering the crops, and how well the crops actually grow can be collected and sent to supercomputers that run amazing algorithms that can make sense of all this information.

And that's just one kind of sensor. Add in other sensors like air quality and temperature, and these algorithms can learn much, much more. With thousands of farms all collecting this information, these algorithms can create incredible insights into how to make crops grow the best, helping to feed the world.

The Power of IoT: Examples & Applications

The Internet of Things (IoT) promises to bring immense value to every organization. By continuing to connect all our things, people, and environments, we'll unlock tremendous organizational value and achieve feats that will truly seem like magic. But because IoT is so broad and far-reaching of a concept, we've found that many are confused about what the potential applications for IoT are exactly. How can my business actually implement IoT solutions? How should my city think about creating value for residents using IoT? We'll give you some Internet of Things examples and applications to clear things up.

We should first make the distinction between consumer IoT and enterprise IoT. Consumer IoT refers to things like wearables, smart home devices, etc., all of which are marketed directly to consumers. In contrast, enterprise IoT refers the use of IoT in improving an organization's existing systems and processes and enabling organizations to increase operational efficiency or unlock entirely new value (e.g. by launching new business lines or products).

We'll be focusing on enterprise IoT in this ebook because we believe that this is where the most value can be created, even if it's not "sexy". Plus, this is the area in which we have deep experience to share with you. Now let's explore some examples and applications of IoT.

It's helpful to think of IoT as doing one (or more) of the following: increasing efficiency, improving health/safety, or creating better experiences.

Increasing Efficiency

> ***"This years' series of Internet of Things (IoT) and Industrial Internet of Things (IIoT) forecasts reflect a growing focus on driving results using sensor-based data and creating analytically rich data sets... solving complex logistics, manufacturing, services, and supply chain problems."***
>
> — Louis Columbus, Roundup of Internet of Things Forecasts and Market Estimates, 2016

Increasing efficiency means more output with the same input or the same output with less input. Inputs could include time, energy, money, or resources. Output could be units produced or tasks accomplished.

Efficiency is particularly important for industrial applications, because more production at less cost means greater profit, but efficiency gains can be realized in just about any organization. Below are some examples:

Manufacturing Efficiency

Sensors embedded in manufacturing equipment and placed throughout a factory can help identify bottlenecks in the manufacturing process. By addressing bottlenecks, manufacturing time and waste is reduced.

Rather than standard preventative maintenance, which means performing maintenance on machines before they break, "predictive maintenance" means using advanced sensing and analytics to predict exactly when machines will need maintenance. Because predictive maintenance means only servicing machines when they need it, this cuts total costs and the time machines spend idle.

Asset Tracking

Whether the assets are big or small, fixed or mobile, attaching sensors to them allows organizations to track real-time location, monitor performance, improve workflows, and optimize utilization.

For example, the smart boating solution we built for Siren Marine enables boat owners to check in on their boat(s) from afar and make sure all systems are functioning correctly. And the car tracking solution we built for Manheim allows personnel at the auction locations to quickly locate the vehicle(s) they're looking for, rather than manually search through thousands of parked cars.

Energy Efficiency

People and organizations can achieve significant decreases in their energy usage with IoT. Sensors monitor things like lighting, temperature, energy usage, etc. and that data is processed by intelligent algorithms to micromanage activities in real-time. This is how Google cut 15% of its energy expenditure in its data centers.

Agricultural Efficiency

For outdoor agriculture, an example could be sensing soil moisture and taking weather into account so that smart irrigation systems only water crops when needed, reducing the amount of water usage.

For indoor agriculture, IoT allows monitoring and management of micro-climate conditions (humidity, temperature, light, etc.) to maximize production.

Inventory Management

By placing tags on individual products, the exact location of single items in a large warehouse can be shared, thus saving search time and lowering labor costs.

Another example is in a retail setting. By knowing exactly what's in-stock and what isn't, the store can order new products only when needed. This reduces the cost of keeping extra inventory in the back. Also, smart inventory management eliminates the need to manually check what's on the shelves, reducing labor costs.

Improved Health and Safety

IoT enables heightened surveillance, monitoring, and detection, which all combine to improve health and increase safety. This is particularly interesting for organizations like local or city governments, which need to ensure the health and safety of their residents, but also extends to large businesses supporting their employees.

Disaster Warning

Sensors can collect critical information about the environment, allowing for early detection of environmental disasters like earthquakes, tsunamis, etc., thus saving lives.

Law Enforcement

Better surveillance and tracking tools will allow authorities to detect when crime has occurred and respond much faster, keeping citizens safer. Also, law enforcement will even be able to predict crime, stopping it from happening in the first place.

Caregiving

Patient surveillance can be life-saving; automatically detecting when someone falls down or when they begin to experience a heart attack so that emergency care can be sent immediately.

Environmental Quality

Sensors can detect radiation, pathogens, and air quality so that dangerous concentrations can be identified early, allowing people to evacuate.

Better Experience

The Internet of Things will allow our world to increasingly shape itself to our needs and our wants, creating a better experience. Rather than just passively providing information and reacting to our inputs, much of the value of IoT will come from anticipating and addressing needs automatically.

For example, if your building(s) are equipped with smart building management systems, they can adjust temperature in real-time in response to occupancy (how many people are in which areas of the building), environmental factors (certain rooms might be getting more sunlight and in need of further cooling), and contextual factors (today might be a holiday so no one will be in the office).

While you might group the above example under energy efficiency because it would in fact be saving energy, this would undervalue the better experience provided to all your employees to make sure everyone has the optimal temperature for comfort and performance.

Hopefully you're beginning to grasp the potential of IoT, and some ways that you can apply it to or within your own organization. But how does an IoT system work exactly?

> **"We are stuck with technology when what we really want is just stuff that works."**
>
> — *Douglas Adams, The Salmon of Doubt*

Background image credit: Vecteezy

How an IoT System Actually Works

As you saw in the previous chapter, the applications for IoT extend across a broad variety of use cases and verticals. However, all complete IoT systems are the same in that they represent the integration of four distinct components: sensors/devices, connectivity, data processing, and a user interface.

We'll outline what each one means in the sections below and how they come together to form a complete IoT system. Each of these sections will also serve as the organizational structure of the rest of this ebook, and we'll dive into these components more deeply in the chapters to follow.

1. Sensors/Devices

First, sensors or devices collect data from their environment. This data could be as simple as a temperature reading or as complex as a full video feed.

We use "sensors/devices," because multiple sensors can be bundled together or sensors can be part of a device that does more than just sense things. For example, your phone is a device that has multiple sensors (camera, accelerometer, GPS, etc), but your phone is not just a sensor since it can perform many actions.

However, whether it's a standalone sensor or a full device, in this first step data is being collected from the environment by something.

2. Connectivity

Next, that data is sent to the cloud, but it needs a way to get there!

The sensors/devices can be connected to the cloud through a variety of methods including: cellular, satellite, WiFi, Bluetooth, low-power wide-area networks (LPWAN), connecting via a gateway/router or connecting directly to the internet via ethernet.

Each option has tradeoffs between power consumption, range, and bandwidth. Choosing which connectivity option is best comes down to the specific IoT application, but they all accomplish the same task: getting data to the cloud.

3. Data Processing

Once the data gets to the cloud (we'll cover what the cloud means in a later section,) software performs some kind of processing on it.

This could be very simple, such as checking that the temperature reading is within an acceptable range. Or it could also be very complex, such as using computer vision on video to identify objects (such as intruders on a property).

But what happens when the temperature is too high or if there is an intruder on property? That's where the user comes in.

4. User Interface

Next, the information is made useful to the end-user in some way. This could be via an alert to the user (email, text, notification, etc). For example, a text alert when the temperature is too high in the company's cold storage.

A user might have an interface that allows them to proactively check in on the system. For example, a user might want to check the video feeds on various properties via a phone app or a web browser.

However, it's not always a one-way street. Depending on the IoT application, the user may also be able to perform an action and affect the system. For example, the user might remotely adjust the temperature in the cold storage via an app on their phone.

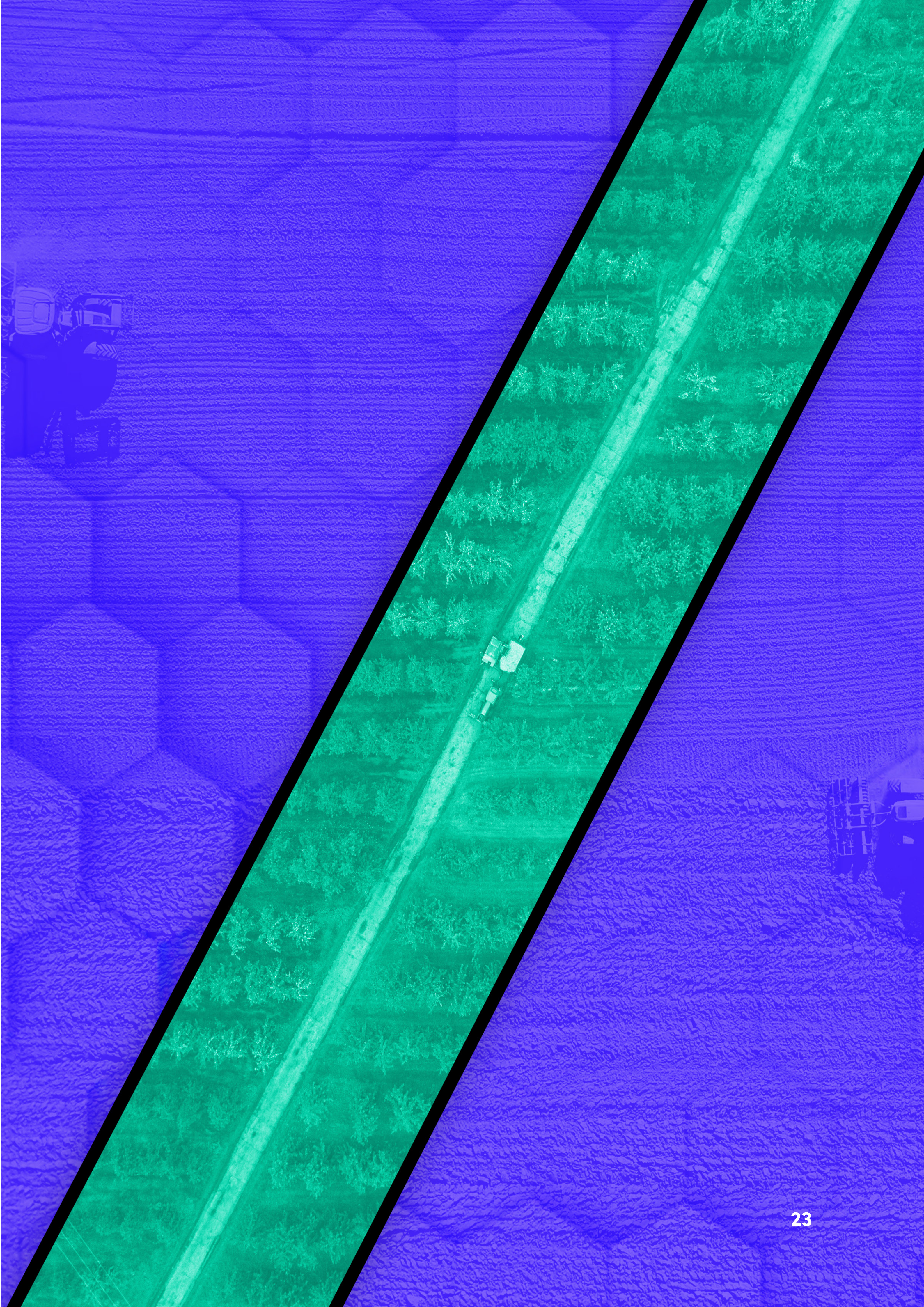
And some actions are performed automatically. Rather than waiting for you to adjust the temperature, the system could do it automatically via predefined rules. Rather than just call you to alert you of an intruder, the IoT system could also automatically notify security teams or relevant authorities.

In Summary

An IoT system consists of sensors/devices which "talk" to the cloud through some kind of connectivity. Once the data gets to the cloud, software processes it and then might decide to perform an action, such as sending an alert or automatically adjusting the sensors/devices without the need for the user.

But if user input is needed or if the user simply wants to check in on the system, a user interface allows them to do so. Any adjustments or actions that the user makes are then sent in the opposite direction through the system: from the user interface, to the cloud, and back to the sensors/devices to make some kind of change.

That's how an IoT system works at a high level. Now we'll take a deeper dive into each of these components to explain what they are, how they work, and important considerations for your organization as you consider building and/or implementing IoT solutions.



2

Sensors & Devices

Hardware Capabilities

As we established in the previous chapter, sensors/devices are a critical piece of the Internet of Things, serving as a system's "senses" and as the means of interacting with the world. Although no complete IoT solution can be built without some kind of hardware, the sensors/devices are often an underappreciated aspect of the system. Choices about the hardware affect everything downstream, from the connectivity you choose, to the analytics you're able to provide, and to the interactions and interfaces that you enable for end-users.

There are too many sensors/devices to possibly provide an exhaustive list here. Ultimately, the choices you make on hardware stem directly from the needs the specific application your organization is interested in pursuing. So instead of exploring all the possible sensors/devices there are out there for you to use, we'll be providing some important factors you need to consider.

Battery Considerations

One of the first considerations is whether or not the sensors/devices that you'll be using will have power available. If you're designing, say, a smart agriculture application with hundreds of sensors spread across broad, rural areas, you're going to have to rely on battery. If your application takes place in a building and only involves a few devices, power may not be an issue.

The reason this is one of the first considerations is because this directly translates into how much you can do with the sensor/device. If your hardware is battery-powered, you want it to last on battery power for a long time - hopefully years (as we'll cover in the next section, replacing and managing batteries at scale is a huge operational burden).

But to get to multi-month or multi-year battery life, the device can't be constantly active. Power hungry operations on the device, such as using GPS or sending and receiving messages over the network, must be used judiciously.

To get a GPS fix, a device must "listen" for signals from at least 4 GPS satellites that are in orbit (here's how GPS works). Depending on the terrain (which can block line-of-site to satellites) and the position of the satellite constellation, this can take 30 seconds, 60 seconds, or more. All of that time is time that the GPS unit is draining precious battery.

The same goes for connectivity. To receive messages over the network, the sensors/devices must be in "listening" mode, and that means battery drain. So when does your device listen? Are there situations where you need to push a critical message down to the device, such as triggering an alarm? Well that's going to significantly impact your battery life. And if not, how often do you want the device to check in?

A "heartbeat" message is a periodic message from the device where it essentially tells you that it's still alive and functioning. If you make its heartbeat once-per-week, you're significantly reducing battery drain, but that also means a device might die and be offline for a week or two before you notice. If it's every 10 minutes, that's much closer to real-time but also 1000x the battery drain relative to the once-per-week.

All of these battery considerations also influence the kind of connectivity you choose, which we'll cover next in the connectivity section. **The point is that hardware decisions are extremely important and there are many considerations to take into account, which all stem directly from your specific use case.**

Over-the-Air (OTA) Firmware Updates

However, regardless of application, having support for over-the-air (OTA) firmware updates is essential. At a high level, "firmware" is the program that's put on the hardware of the device, basically telling it how to function and perform. As the name implies, it's between hardware (which you can't change once it's been manufactured) and software (which you can update with relative ease).

While firmware can be updated (and should be), it's a non-trivial process. Having devices that can receive updates to their firmware over-the-air (meaning that that you can update them over the network, rather than needing to have them physically in your hands) is critical, especially in applications where you have devices spread over large areas.

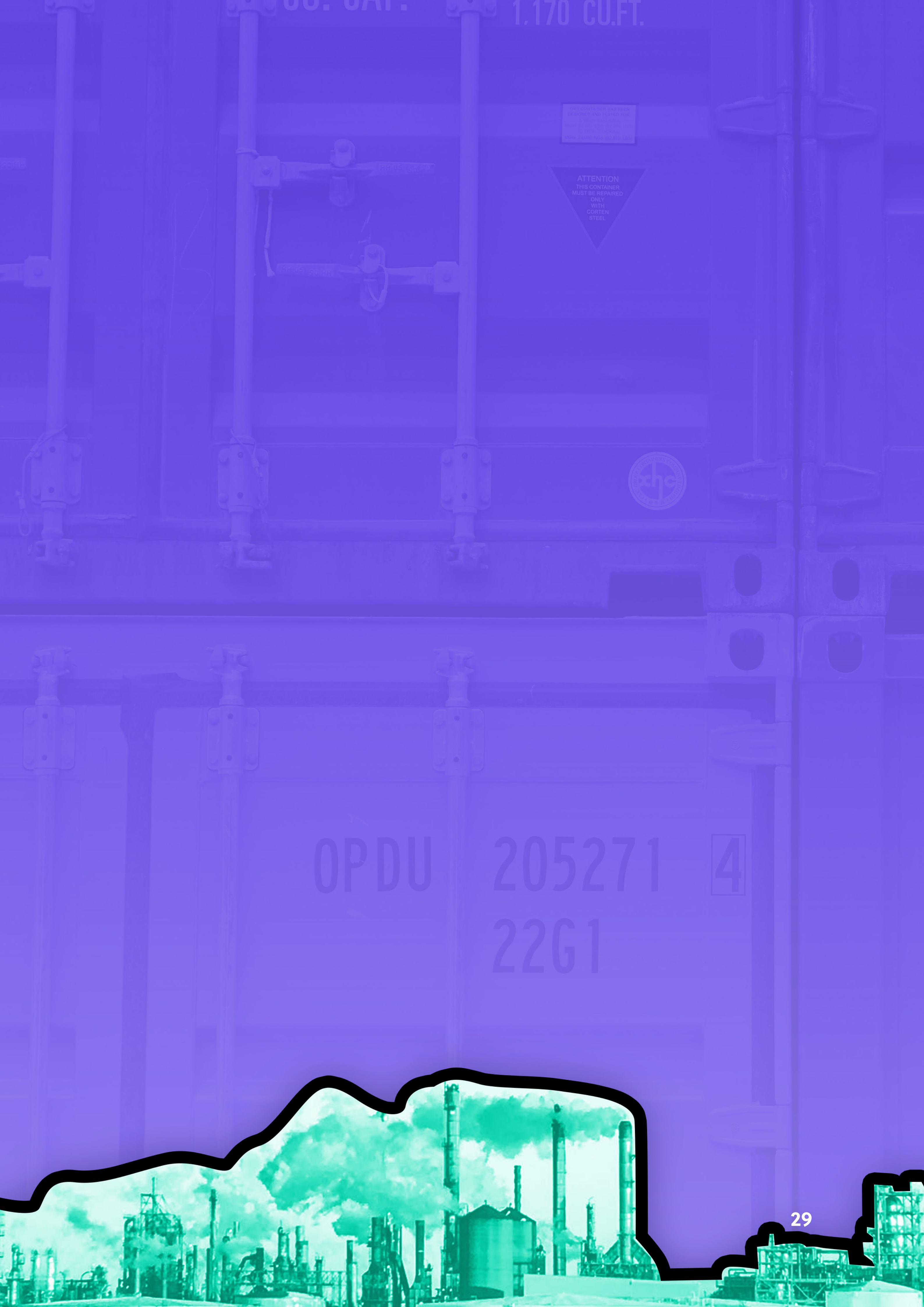
OTA firmware updates are a critical tool in addressing issues that may come up as you learn and refine, but they can also cause major issues. If you're considering pushing a firmware update to all of your sensors/devices, make sure that you've tested extensively on a small subset that are actually out in the field. We've had experiences where firmware updates caused unforeseen issues and drained the batteries of hundreds of devices before a new firmware update to fix it could be sent OTA.

Key Takeaways

We've given a few examples of important hardware considerations above, but IoT applications vary so widely in their requirements that a comprehensive exploration of all considerations isn't within the scope of this ebook.

One of the key takeaways from our experiences is that no matter how much you think through your application, inevitably you're going to run into things you didn't foresee. So if your application has been done before, you may be able to find hardware that has been purpose-built directly for your application, which you can and should leverage.

If you're pursuing something new, make sure to work with an experienced hardware partner and listen closely to their expertise. Every little decision matters, and there are likely many considerations you may not have thought of.


1,170 CU.FT.
ATTENTION
THIS CONTAINER
MUST BE REPAIRED
ONLY
WITH
CORTEN
STEEL
OPDU 205271 4
22G1

Scaling & Operations

Like the considerations around the capabilities of hardware, which we just explored in the previous chapter, another underappreciated aspect of a large-scale IoT solution is the operational component.

It's one thing to deal with a prototype that only has a few pieces of connected hardware, or a pilot with just a few hundred sensors/devices at one location. It's quite another when you scale up to a full production system with potentially millions of sensors/devices, and there are critical considerations purely on the operations side that you need to address.

More Battery Considerations

In the last chapter, we covered how the needs of the application influence hardware decisions like whether or not you use battery power in your sensors/devices. If you go with battery power, what happens when you need to replace the batteries? First, you need to figure out how you're going to access of collect the sensors/devices.

Are they spread over miles of terrain? That means you'll need to know where they all are, because if they run out of battery power they won't be using GPS.

Or maybe they don't even have GPS because they're just soil moisture sensors and don't need it. Do you go get them one at a time as the batteries die? Or do you wait and do it in batches?

How do you replace the batteries? If you're replacing, don't forget the cost of all the additional batteries you'll need!

Or are they rechargeable? This means you don’t need to replace batteries, which is nice, but now you need to figure out how they’re recharging. If they can charge wirelessly that’s great, you can just put them on a wireless charging matt and decrease hassle. But if they need to be plugged in, that means either dedicated charging stations for multiple sensors/devices at once, or a ton of wires. Or maybe the devices are cheap enough that you just replace them completely; in that case, make sure to consider how you’re going to dispose of them!

Sensor/Device Association

Say you’re doing an asset tracking application. Great. That means you have some way of tracking the location of the asset (probably GPS if outdoors and over large areas, probably bluetooth if indoors and within relatively confined areas) by putting a device on it. But how do you know that a specific device is associated with a specific asset?

Sensor/device association (for example, device A12B3 goes with this car) can be a big hassle. It’s critical, because otherwise you just know the location of the sensors/devices but not the specific assets they’re attached to. This means that, at some point in the process, you need to make that association. Does someone have to manually enter the information? Is there a barcode on the asset that you can scan? For example, all cars have a VIN (vehicle identification number) which can be scanned. That’s awesome, but now you need to sync the information you already have in your system (the VINs of your cars) with the new location data you’re getting. And that’s going to require some integration.

Sensor/Device Errors

As much as we all hope for things to work perfectly, inevitably there are going to be errors. These could stem from a defect in manufacturing for a specific sensor/device or could be due to a bug in the firmware.

Regardless, you need a process for how you handle when a sensor/device has an error. How do you 1) identify the error in the first place and 2) address the issue once you find out?

The ability to predict and proactively address errors is absolutely critical. Dealing with errors in sensors/devices after they happen can be a huge operational burden. Just imagine trying to find a single sensor/device over square miles of parking lot when the error means that you can't get GPS location anymore. Not fun.

Key Takeaways

The above considerations may not matter for many applications, whether that's because you don't need that many devices or because you're not relying on battery power. Again, we bring up these examples simply to get you thinking and realizing that there are many purely operational factors, beyond technology, to consider when you go from prototype to pilot to full scale.

This is why many IoT deployments fail. Not because the underlying technology isn't good enough, but because the sheer operational burden makes the return-on-investment not worth the effort.

This is also why it's so critical that you have a clear, measurable impact you're trying to achieve with your solution.

Manufacturing & Shipping

During the prototype and pilot phases, you can get by with using hardware that's been hacked together from off-the-shelf parts. But when you go to deploy a full IoT system, you're likely going to need to work with a manufacturer.

Manufacturing could be an entire ebook in and of itself (and maybe it will be, let us know if you're interested!), but here we'll give you some of the important considerations for manufacturing production-grade sensors/devices. Some of the points below won't apply when you're using sensors/devices that have already been built and manufactured, and whether you need to build entirely new sensors/devices will (as always) depend heavily on your specific use case.

Manufacturing: It's Going to Take a While

The most important consideration is that it's going to take a while. Do not expect to go from a few prototypes to multiple thousands of production units within a couple months. We've done it in four or five months, but that's like trying to sprint a marathon. It's possible, but it's more than likely you'll burn out and fail.

Expect several months to a year to go through the entire process. And scale also factors in here; a manufacturer will be much more willing to move quickly if they know they'll be producing hundreds of thousands of units than they will to produce just a few thousand.

This process takes a considerable amount of time for a number of reasons.

With software, if there are bugs or areas of improvement, you can update the system after it's live. And with the connectivity layer, you might be using proven standards and existing infrastructure that carry relatively low risk (like WiFi or Cellular). Even if you're using a relatively new standard, making updates to the network can also be performed after-the-fact.

With hardware, you are not making changes to your sensors/devices once they've been produced. This is why having a small scale pilot is absolutely critical, it allows you to rigorously test your prototypes to identify any bugs or weaknesses in the hardware so that the final units can as purpose-built for the specific use case as possible. And manufacturing usually involves building injection molds, which help reduce per-unit costs but have massive upfront fixed-costs, meaning that it's extremely costly and time-intensive to make changes.

Also, for any sensor/device that's communicating wirelessly, you'll need to get FCC certifications (if you're in the US that is; for any other country you'd get certifications from the relevant governing body). Wireless means that the device is communicating using electromagnetic waves over a certain spectrum, so these certifications are to ensure that your device isn't harmful to people nor infringing on licensed bands.

Another reason that the manufacturing process can take a while, is that you need to source all the materials. For a given sensor/device, there may be dozens or into the hundreds of individual components necessary to build the full device.

The manufacturer will need to set up production lines too. This involves setting up all the equipment and assembly lines to actually manufacture the sensors/devices, as well as setting up the testing processes for each important stage.

You will want your sensors/devices to be tested throughout the entire process to ensure that you don’t produce an entire batch of units with defects.

Shipping: Unexpected Issues

Finally, once the units have been produced, they’ll need to be shipped to wherever they’re needed (perhaps directly to you or to the location where they will be deployed). If you’re manufacturing in China, which is likely, this means that the devices will first need to pass customs because they’re being imported from abroad. This process can be several days or weeks and doesn’t have a set time, so make sure there’s a bit of cushion in your delivery schedule.

Shipping itself is an important consideration. If units have lithium batteries, there are regulations that may prevent them from being transported on airplanes (due to possibility of combustion).

Also, it’s critical to consider device behavior (which ties into our earlier chapter on hardware capabilities). Do your devices “know” they are being shipped? Do they stay in sleep mode? We once made the mistake of shipping units that, because of movement, “woke up” and began trying to find a network. Since they were in transit, they couldn’t find a network and ended up completely draining their batteries trying to connect. Most were dead when they arrived on location.

Key Takeaways

It’s no accident that hardware has “hard” in its name. If you do not specialize in hardware, find hardware partners with extremely deep experience and high aptitude to complete your project. We recommend that you listen closely to the words of any hardware partners you work with.

Gateways

As mentioned previously, our goal in this section of the ebook isn't to go into detail about specific sensor/device types. However, nearly every IoT system needs some way to connect its sensors/devices to the cloud so that data can be sent back-and-forth between them. So in this chapter we'll be exploring a particular type of IoT hardware called the gateway, which makes that connection to the cloud possible.

Gateways act as bridges between sensors/devices and the cloud. Many sensors/devices will "talk" to a gateway and the gateway will then take all that information and "talk" to the cloud.

But you may be wondering, what benefit is there to taking that extra step between the sensors/devices and the cloud? There are several benefits:

Battery life

As you may be noticing, battery life tends to be a critical consideration for many IoT systems. For example, take an IoT solution that operates in a remote area. To get data from sensors/devices to the cloud, there will need to be a long-range connection, usually provided by satellite. As will be explained in greater depth in the connectivity section, longer range typically means increased power consumption (and costs); this can be a problem for small sensors/devices with limited battery life.

If you're doing Smart Agriculture, you want your field sensors to last years, not months or weeks. By using an elevated gateway installed near the top of an outbuilding or grain silo, the sensors/devices only have to send data a relatively short distance to the gateway and the gateway can then send the data to the cloud through a single higher bandwidth connection like satellite. **Gateways allow sensors/devices to communicate over shorter distances, boosting battery life.**

Varying Protocols

A complete IoT application might involve many different kinds of sensors and devices. Using Smart Agriculture again as an example, you might want sensors for temperature, moisture, and sunlight and devices such as automated irrigation and fertilizer systems.

All of the different sensors and devices can use varying transmission protocols (basically, the rules and format for the information being transmitted). Protocols include LPWAN, Wi-Fi, Bluetooth, and Zigbee, among many others.

Gateways can communicate with sensors/devices over varying protocols and then translate that data into a standard protocol such as MQTT to be sent to the cloud.

Unfiltered Data

Sometimes, sensors/devices can generate so much data that it’s overwhelming to the system or extremely costly to transmit and store. Often in such cases, only a small fraction of the data is actually valuable. For example, a security camera doesn’t need to send video data of an empty hallway.

Gateways can pre-process and filter the data being generated by sensors/devices to decrease transmission, processing, and storage requirements.

There are also techniques that can be employed on the sensor processor itself (if there is enough processing power) to limit the amount of unfiltered data sent to the gateway or directly across the network.

High Latency

Time can be critical for certain IoT applications; the sensors/devices can't afford to transmit data to the cloud and wait to get a response before taking action. This is true for life-or-death situations in the medical realm or for fast-moving objects like cars.

Higher latency can be avoided by processing the data on the gateway or on the sensor itself and giving commands locally. However, many sensors/devices in IoT applications are too small and too power constrained to do the processing themselves.

Gateways can reduce latency in time-critical applications by performing processing on the gateway itself rather than in the cloud.

Security

Every sensor/device that is connected to the internet becomes vulnerable to being hacked. Hacked sensors/devices are bad. Not just for the owner, but for everyone else too.

Gateways reduce the number of sensors/devices connected to the internet because the sensors/devices are only connected to the gateway. However, this makes gateways themselves targets and also the first line of defense. This is why security needs to be a priority for any gateway.

Key Takeaways

Not all IoT applications will need a gateway, but they're an important class of hardware that's often a requirement for certain use cases because they're needed to provide the connectivity to the sensors/devices.

In the next section we'll explore connectivity as a whole for IoT as well as specific connectivity and network standards that you may need to consider.

3 Connectivity

An Introduction to Connectivity

When it comes to connecting the Internet of Things, there are a seemingly overwhelming number of options. Cellular, satellite, WiFi, Bluetooth, RFID, NFC, LPWAN, and Ethernet are just some of the possible ways to connect a sensor/device to the internet. And within each of these options there can be different providers (e.g. for cellular there's T-Mobile, Verizon, AT&T, Sprint, etc.). Connectivity is a huge facet of IoT, so it's important to understand the options so your project runs smoothly at the lowest expense.

Trade Off Between Power Consumption, Range, and Bandwidth

The perfect connectivity option would consume extremely little power, have huge range, and would be able to transmit large amounts of data (high bandwidth). Unfortunately, this perfect connectivity doesn't exist.

Each connectivity option represents a tradeoff between power consumption, range, and bandwidth. This allows us to segment the various connectivity options into three major groups, which you'll find below. However, these groups should serve more as a framework for thinking about connectivity than a definitive classification, as there can be connectivity standards that sit more on the borders of these groups.

1. High Power Consumption, High Range, High Bandwidth

To wirelessly send a lot of data over a great distance, it takes a lot of power. A great example of this is your smartphone. Your phone can receive and transmit large amounts of data (e.g. video) over great distances, but you need to charge it every 1–2 days. Connectivity options in this group include cellular and satellite.

Cellular is used when the sensor/device is within coverage of cell towers. For sensors/devices that are, say, in the middle of the ocean, satellite becomes necessary.

2. Low Power Consumption, Low Range, High Bandwidth

To decrease power consumption and still send a lot of data, you have to decrease the range. Connectivity options in this group include WiFi, Bluetooth, and Ethernet.

Ethernet is a hard-wired connection, so the range is short because it's only as far as the wire length. WiFi and Bluetooth are both wireless connections with high bandwidth and lower power consumption than cellular and satellite. However, as I'm sure you've experienced just walking around your home, the range is limited.

3. Low Power Consumption, High Range, Low Bandwidth

To increase range while maintaining low power consumption, you have to decrease the amount of data that you're sending. Connectivity options in this group are called Low-Power Wide-Area Networks (LPWANs).

LPWANs send small amounts of data which allows them to operate at very low power with ranges in miles rather than feet. For example, a moisture sensor for agricultural purposes doesn't need to send a lot of data, perhaps just a single number (the moisture level) every few hours. You also don't want this sensor to consume a lot of power because it needs to run on battery (plugging it into an outlet in the middle of a field just isn't realistic). And since agriculture covers a wide area, WiFi and Bluetooth lack the range.

LPWANs are extremely useful for many IoT applications. They allow tons of sensors/devices to collect and send data over broad areas while lasting years on battery life. Although they can't send much data, most sensors don't need to. However, these kinds of application often need IoT gateways to work, which we explored in the previous chapter.

When to Skip Connectivity

The Internet of Things is made up of connected sensors/devices, so by definition an IoT system needs some kind of connectivity, especially if it uses the cloud.

However, there are certain cases where the data processing or the interaction with the sensor/device through the user interface can take place without any data first being transferred over an external network.

Why Skip the Connectivity?

One reason is latency. Latency refers to how long it takes for a packet of data to get from the start point to the end point. Although latency doesn't matter in the vast majority cases, for some IoT applications latency is critical.

Imagine you're in a self-driving car and suddenly somebody loses control of their car in front of you. Would you want to wait for the self-driving car to send data to the cloud, have that data processed, then have instructions for what to do sent back to the car? No! Those milliseconds could mean life or death.

Even if you're the one driving the car, you want the user interface (i.e. the steering wheel) directly hooked up to the device (i.e the car) rather than waiting for your input to be transmitted externally, processed, and then sent back.

Another reason is that sending lots of data can become really expensive. Some IoT applications collect a ton of data but only a small fraction is actually important. Local algorithms can restrict what gets sent thus lowering costs.

A good example is a security camera. Streaming video takes a lot of data, but the vast majority of the footage might be of an empty hallway.

So How Do You Skip the Connectivity?

Rather than send data over a network for it to be processed in the cloud, an alternative approach is to process the data on a gateway or on the sensor/ device itself.

This is called either fog computing or edge computing (because you're bringing the cloud "closer to the ground" and the computing is taking place at the edges of the IoT system rather than the center).

For the security camera, it could use machine vision to "watch" for anything abnormal and only then send that footage to the cloud.

For the self-driving car, the data processing all takes place in the onboard computer which allows for faster decision-making.

Key Takeaway

Every IoT system combines the four components we've outlined and begun to detail: Sensors/Devices, Connectivity, Data Processing, and User Interface. However, an IoT system can combine these components in different ways and can use very different forms of connectivity. It all comes down your specific application and organizational need.

In the coming chapters we'll take a deeper dive into some of the connectivity options we've mentioned here.

LPWAN

As the name implies, Low-Power Wide-Area Networks (LPWANs) allow for low power consumption over a wide area, aka long range. So how is this accomplished?

Messages sent over LPWAN must be small and simple. Because of their simplicity, these messages can be communicated over the distance without a large power source. For machines, decreasing the amount of data sent (the bandwidth) means lower energy at range.

This is what LPWANs do, they send and receive small packets of information at infrequent intervals. Sensor/devices can send data over miles of range instead of feet and can last for years on battery instead of weeks or months.

However, LPWANs aren't without downsides. Messages that are transmitted over LPwan sometimes aren't received by the gateway (called packet loss). This can usually be overcome by sending multiple messages or by adding additional gateways to the network, but these solutions have power and financial costs respectively.

Despite certain disadvantages, LPWANs play an essential role in the Internet of Things.

Key Takeaways

IoT applications can vary greatly, but many applications need tons of sensors spread over big areas. There are many ways for these sensors/devices to communicate, each with varying pros and cons. When you have thousands of sensors spread over a big area, you need wireless communication with long range and low power consumption. After all, and as we covered previously, it’s a tremendous operational burden to replace the batteries in thousands of sensors on a frequent basis. It also costs money to send messages and connectivity options like cellular are expensive. Imagine having to pay your phone bill not just for one device, *but for thousands.* Yikes.

LPWAN technology thus plays a crucial role in enabling the Internet of Things. These networks make it possible to have many thousands of sensors/devices collecting and sending data at lower cost, over longer range, and with better battery life than other connectivity options. Some use cases among many include:

1. **A car auction lot or dealership** — sensors are placed on vehicles to track their location and status in real-time.
2. **A parking garage** — sensors detect when spots are open, sending a simple Yes or No message only when that value changes.
3. **A school building** — battery-powered locks can be remotely activated or deactivated, helping with general security and crisis situations.
4. **A city** — waste containers throughout a city can send alerts when they’re close to being full, allowing for more efficient garbage collection.

It’s important to note that LPWAN is a general term, and there are many different competing standards and technologies under that umbrella. The competing LPWAN standards and technologies include but are not limited to: LoRa, SIGFOX, Ingenu, Weightless, and SymphonyLink. For the purposes of this ebook we won’t go into detail on these specific technologies, but we have hands-on experience with most of them which we’re always happy to share if you’re weighing them for an application.

Cellular

Cellular networks provide the backbone for much of what we know and love, allowing us to access the internet, send messages, and connect with friends. In addition to the personal benefits we're all familiar with, cellular networks also serve a critical role in many Internet of Things applications.

As we've discussed in previous chapters, there will always be a tradeoff between power consumption, range, and bandwidth. Cellular connectivity has been focused on range and bandwidth at the expense of power consumption, meaning that it can send lots of data over long distance but drains battery rather quickly.

This is fine for devices that are connected to an electricity source or that can be recharged often (i.e. your phone), but a no-go when it comes to IoT applications that require remote sensors/devices to last months or years on battery.

As such, cellular connectivity is usually reserved for backhaul (i.e. a gateway might use LPWAN to talk to all the sensors/devices but use cellular to connect to the cloud and pass along that data) or for sensors/devices that need to send a lot of data and/or don't have concerns about battery life.

However, that's not the full story when it comes to cellular. You've probably heard names like 2G, 3G, and 4G (which refer to different generations of cellular networks), but new cellular technologies like Cat-1, LTE-M, and NB-IoT are aimed specifically at IoT applications. As with LPWANs, we won't go into the details of these cellular technologies, but the main takeaway is that they are either upgrades to existing networks or new infrastructure entirely, and all aimed specifically at reducing data costs per sensor/device and power requirements.

Some of these cellular technologies are currently available, and others are promised but yet to come. This also includes 5G, which will also have significant implications for IoT applications and enable high-bandwidth, high-speed applications like Ultra-HD (4K) streaming, self-driving car connectivity, or VR/AR applications.

There's also discussion around supporting IoT devices with 5G-IoT networks. **However, all these are just speculations** as 3GPP (the standards organization for cellular technologies) will finalize the specifications in 2019. The commercial rollout target year is 2020.

Key Takeaway

It's important to understand that these different options do not have to be mutually exclusive. This extends to other connectivity options as well, like the LPWANs.

IoT covers a broad spectrum of applications. Sometimes you need high bandwidth, like with real-time surveillance. For asset tracking, data throughput is small, but there are inevitably many handovers as objects move. Smart meters and many smart city use-cases require small data transfer once or twice a day. This means that no one technology (even 5G) may fit the specific needs of your particular IoT solution and may use a combination.

Satellite

As the name implies, this form of connectivity uses satellites to connect sensors/devices to the cloud. The first artificial communications satellite was launched in 1960, and served merely as a giant reflector for signals beaming between different places on the earth's surface. Today's communications satellites are much more robust and featured.

The importance of satellite connectivity for the Internet of Things comes from its incredible coverage. A single network of satellites is capable of providing coverage to effectively the entire planet. This means that a single device moving around the world can stay on a single network and use only a single connectivity type.

Satellite's incredible range give it an advantage in remote areas that other communication types such as cellular or Wi-Fi cannot reach, and in places that have underdeveloped infrastructure or none at all, such as the middle of the ocean.

Satellite connectivity has two major configurations with respect to connectivity: direct and backhaul.

Direct

The first major type of configuration, direct, is broken down into the two sub-categories: dual mode and satellite only.

Dual mode satellite connectivity is connectivity that uses cellular data as much as possible and uses satellite when necessary. This gives a best-of-both-worlds connectivity option that leverages the lower cost and higher bandwidth of cellular when possible, but makes use of satellite connectivity's greater coverage to fill in spaces where cellular data connections are sparse or unreliable.

The best example of this connectivity is container ships, which use cellular when in port or near coastlines, but make use of satellite when on the open ocean.

Satellite-only connectivity is exactly what is sounds like, a data connection that uses purely satellite connectivity to transmit data. This is typically for large, immobile resources like oil and gas equipment, that are sending large amounts of data from locations that have no cellular or other connectivity options.

Backhaul

The second major type of configuration, backhaul, uses a main tower that connects directly to a satellite and then a different kind of connectivity (e.g. an LPWAN) to connect with the sensors/devices in the area. This connectivity option is typically used when you have many low bandwidth sensors/devices in remote areas.

Satellite requires high power usage, and can require larger pieces of equipment such as dishes for connectivity. This raises the cost for individual sensors/devices, and can make direct connection infeasible for groups of sensors/devices that don't use much data.

One example of this is a farm that uses a set of moisture sensors to collect soil data. All of those sensors may use an LPWAN to connect to a main tower that then transmit the data over a satellite connection. This saves on battery life and lowers the overall cost of the sensors.

Key Takeaways

Satellite has excellent coverage, but with it comes larger equipment and higher battery usage than other connectivity options. Satellite also has good bandwidth, but can be expensive at scale. As such, it fills a niche where a single tower can be used to service a group of sensors/devices, sensors/devices are larger and higher costs are acceptable, or sensors/devices are so remote that satellite is the only means of transmitting data to the cloud.

In these instances, satellite is an excellent connectivity option because a single network can encompass the entire globe, and connectivity can be reliable in places no other options can reach, even in the middle of the ocean.

WiFi

WiFi has a few notable differences from other wireless technologies. For example, WiFi transmits at frequencies of 2.4 GHz or 5 GHz. These frequencies are much higher than the frequencies used for cellular transmission. Higher frequency means that signals can carry more data.

However, as you now know well, all forms of wireless communication represent a tradeoff between power consumption, range, and bandwidth. So in exchange for high data rates, WiFi consumes a lot of power and doesn’t have a lot of range.

The longest range WiFi has ever transmitted data is 260 miles. The Swedish Space Agency transmitted data to an overhead stratospheric balloon 260 miles away, but they used non-standard WiFi equipment and 6 watt amplifiers to achieve this.

For your average WiFi router, ranges are much, much shorter and depend on a number of factors. Range can depend on the antenna, reflection and refraction, and radio power output. A range of about 100 ft is common, so if you have thousand of sensors out in a field, WiFi isn’t a great option.

WiFi can be good for IoT applications that don’t have to worry about power drain (e.g. devices that are plugged into an outlet), that need to send a lot of data (e.g. video), and that don’t need high range. A good example would be a home security system.

Types of WiFi

Like LPWANs and cellular connectivity, there are several versions of WiFi including, 802.11a, 802.11b, 802.11g, 802.11n, and 802.11ac.

Each of these standards comes with various pros/cons related to data speed, signal interference from outside sources, and cost. Cost is a factor because different hardware is needed for different standards, though newer versions are made to be backwards compatible with older versions.

So, while WiFi currently isn’t great for many IoT applications, there are two WiFi standards that have been developed, or are being developed, specifically for IoT; WiFi HaLow (802.11ah) and HEW (802.11ax).

WiFi HaLow was ratified in 2016 and aimed at addressing range and power concerns for IoT applications. HEW (High Efficiency Wireless) is an upcoming standard that builds on HaLow to add additional IoT-friendly features.

Key Takeaway

As always, it all comes down to your specific application. One of the advantages of WiFi is that it’s a proven and standardized technology that’s already present in many buildings and public areas. However, current instantiations of WiFi lack the necessary range and consume too much power for many IoT applications.

Bluetooth

Invented by Ericsson in 1994, Bluetooth was intended to enable wireless headsets. Bluetooth has since expanded into a broad variety of applications including Bluetooth headsets, speakers, printers, video game controllers, and much more.

Bluetooth is also important for the rapidly growing Internet of Things, including smart homes and industrial applications. It is a low power, low range, high bandwidth connectivity option. When Bluetooth devices connect to each other (for example, your phone and your wireless speaker), it follows the parent-child model, meaning that one device is the parent and other devices are the children. The parents transmits information to the child and the child listens for information from the parent.

A Bluetooth parent can have up to 7 children, which is why your computer can be connected via Bluetooth to multiple devices at the same time. When devices are connected together via Bluetooth, it's called a "piconet".

Not only can a device be a parent in one piconet and a child in a different piconet at the same time, but the parent-child relationship can also switch. When you put your Bluetooth device in pairing mode to connect it, it's temporarily becoming the parent so that it can establish a connection and proceeds to connect as the child.

In contrast to WiFi, which we explored in the previous chapter, Bluetooth was meant for portable equipment and related applications therefore excels when you need to connect two devices with minimal configuration. Also, because Bluetooth uses weak signals, there's limited interference and devices can communicate in "noisy" environments.

In the Industrial Internet of Things, machines often need to send short bursts of data in extremely noisy environments. With potentially hundreds of sensors and devices sending data, WiFi poses too much hassle to set up.

A drawback of Bluetooth is lower bandwidth, but for many industrial applications this higher bandwidth simply isn't needed.

Bluetooth is also useful in a smart home setting. Again, many devices in the smart home don't need high bandwidth connections and it's much easier to set up Bluetooth.

Furthermore, newer versions of Bluetooth can create a self-healing mesh network which means that individual devices can still communicate even if one device runs out of power or is disconnected. If your door locks, HVAC system, washer, dryer, fridge, and lights are all connected, you certainly wouldn't want them all to fail just because one goes down.

Bluetooth Version 5

The Bluetooth Special Interest Group officially adopted Bluetooth 5 as the latest version of Bluetooth back in December 2016.

> ***"With Bluetooth 5, Bluetooth continues to revolutionize how people experience the IoT. Bluetooth continues to embrace technological advancements and push the unlimited potential of the IoT."***
>
> —Bluetooth 5 Now Available

As is clear from Bluetooth SIG's announcement, Bluetooth 5 is specifically aimed at the Internet of Things. It boasts quadruple the range, double the speed, and boosts broadcast messaging capacity by 800%. It also introduces the mesh networking capability mentioned above.

Bluetooth 5 is backwards-compatible with previous versions of Bluetooth, but new hardware is required to take advantage of the new benefits listed above. So it might be awhile until we see all the benefits that Bluetooth 5 has to offer, but it's an exciting development as the Internet of Things continue to gain traction.

Key Takeaway

In addition to the capabilities explored above, Bluetooth can also provide indoor asset tracking by using multiple Bluetooth beacons and using their relative signal strengths to triangulate position. GPS is great for outdoor applications but has inherent accuracy limitations and fails indoors when sensors/devices can't receive the signal from the GPS satellites.

Together with the advantages in noisy environments and the ease of setup, Bluetooth is therefore a strong option for many indoor Internet of Things applications.

4

Data Processing

Introduction to the Cloud

So far we've covered the sensors/devices that are out in the world collecting data, and the connectivity technologies that enable those sensors/devices to pass that data up to the cloud for processing. But what is the cloud? And what happens when that data is received?

Back in the 1970s, it was popular for businesses to rent time using big, mainframe computer systems. These systems were extremely large and expensive, so it didn't make sense financially for businesses to own the computing power themselves. Instead, they were owned by large corporations, government agencies, and universities.

Microprocessor technology allowed for great reductions in size and expense, leading to the advent of the personal computer, which exploded in popularity in the 1980s. Suddenly, businesses could (and did) bring computation in-house.

However, as high-speed connections have become widespread, the trend has reversed: businesses are once again renting computing power from other organizations. But why is that?

Instead of buying expensive hardware for storage and processing in-house, it’s easy to rent it for cheap in the cloud. The cloud is a huge, interconnected network of powerful servers that performs services for businesses and for people. The largest cloud providers are Amazon, Google, and Microsoft, who have huge farms of servers that they rent to businesses as part of their cloud services.

For businesses that have variable needs (most of the time they don’t need much computing, but every now and then they need a lot), this is cost effective because they can simply pay as-needed.

When it comes to people, we use these cloud services all of the time. You might store your files in Google Drive instead of on your personal computer. Google Drive, of course, uses Google’s cloud services. You might also listen to songs on Spotify instead of downloading the songs to your computer or phone. Spotify uses Amazon’s cloud services.

Generally, something that happens “in the Cloud” is any activity that takes place over an internet connection instead of on the device itself.

The Internet of Things and the Cloud

Because activities like storage and data processing take place in the cloud rather than on the device itself, this has had significant implications for IoT. Many IoT systems make use of large numbers of sensors to collect data and then make intelligent decisions.

Using the cloud is important for aggregating data and drawing insights from that data. For instance, a smart agriculture company would be able to compare soil moisture sensors from Kansas and Colorado after planting the same seeds. Without the cloud, comparing data across wider areas is much more difficult.

Using the cloud also allows for high scalability. When you have hundreds, thousands, or even millions of sensors/devices, putting large amounts of computational power on each sensor/device would be extremely expensive and energy intensive. Instead, data can be passed to the cloud from all these sensors and processed there in aggregate.

For much of IoT, the head (or rather, the brain) of the system is in the cloud. Sensors/devices collect data and perform actions, but the processing/commanding/analytics (aka the "smart" stuff), typically happens in the cloud.

So Is the Cloud Necessary for IoT?

Technically, the answer is no. The data processing and commanding could take place locally rather than in the cloud via an internet connection. Known as "fog computing" or "edge computing", this actually makes a lot of sense for some IoT applications. However, there are substantial benefits to be had using the cloud for many IoT applications including:

- Decreased costs, both upfront and infrastructure
- Pay-as-needed for storage/computing
- High system scalability and availability
- Increased lifespan of battery-powered sensors/devices
- Ability to aggregate large amounts of data
- Anything with an internet connection can become "smart"

There are legitimate concerns with cloud usage though:

→ **Data ownership:** When you store data in a company's cloud service, do you own the data or does the cloud provider? This can be hugely important for IoT applications involving personal data such as healthcare or smart homes.

→ **Potential crashes:** If connection is interrupted or the cloud service itself crashes, the IoT application won't work. Short-term inoperability might not be a big deal for certain IoT applications, like smart agriculture, but it could be devastating for others. You don't want applications involving health or safety crashing for even a few seconds, let alone a few hours.

→ **Latency:** It takes time for data to be sent to the cloud and commands to return to the device. In certain IoT applications, these milliseconds can be critical such as in health and safety. A good example is autonomous vehicles. If a crash is imminent, you don't want to have to wait for the car to talk to the cloud before making a decision to swerve out of the way.

The Internet of Things is a broad field and includes an incredible variety of applications. There is no one-size-fits-all solution so you need to consider your organization's specific application when deciding whether the cloud makes sense.

Introduction to IoT Platforms

Whether you're new to IoT or a seasoned veteran, you've probably heard the term "IoT Platform" before. After all, there were over 300 IoT platforms as of 2016 and this number continues to quickly grow (I've heard there are now over 700). The IoT platform market is growing at a compound annual growth rate (CAGR) of 33% and is expected to reach a $1.6 billion market size in 2021.

IoT platforms are a critical component of the IoT ecosystem, but we've found that for many people, it's not clear what an IoT platform exactly is or what the differences are between them.

So What Is an IoT Platform Exactly?

IoT platforms are the support software that connects everything in an IoT system. An IoT platform facilitates communication, data flow, device management, and the functionality of applications. With all the varying kinds of hardware and the different connectivity options that you just read about in the previous section, there needs to be a way of making everything work together and that's what IoT platforms do.

IoT platforms help:

- Connect hardware;
- Handle different communication protocols;
- Provide security and authentication for devices and users;
- Collect, visualize, and analyze data; and,
- Integrate with other web services.

When Should Your Organization Use an IoT Platform?

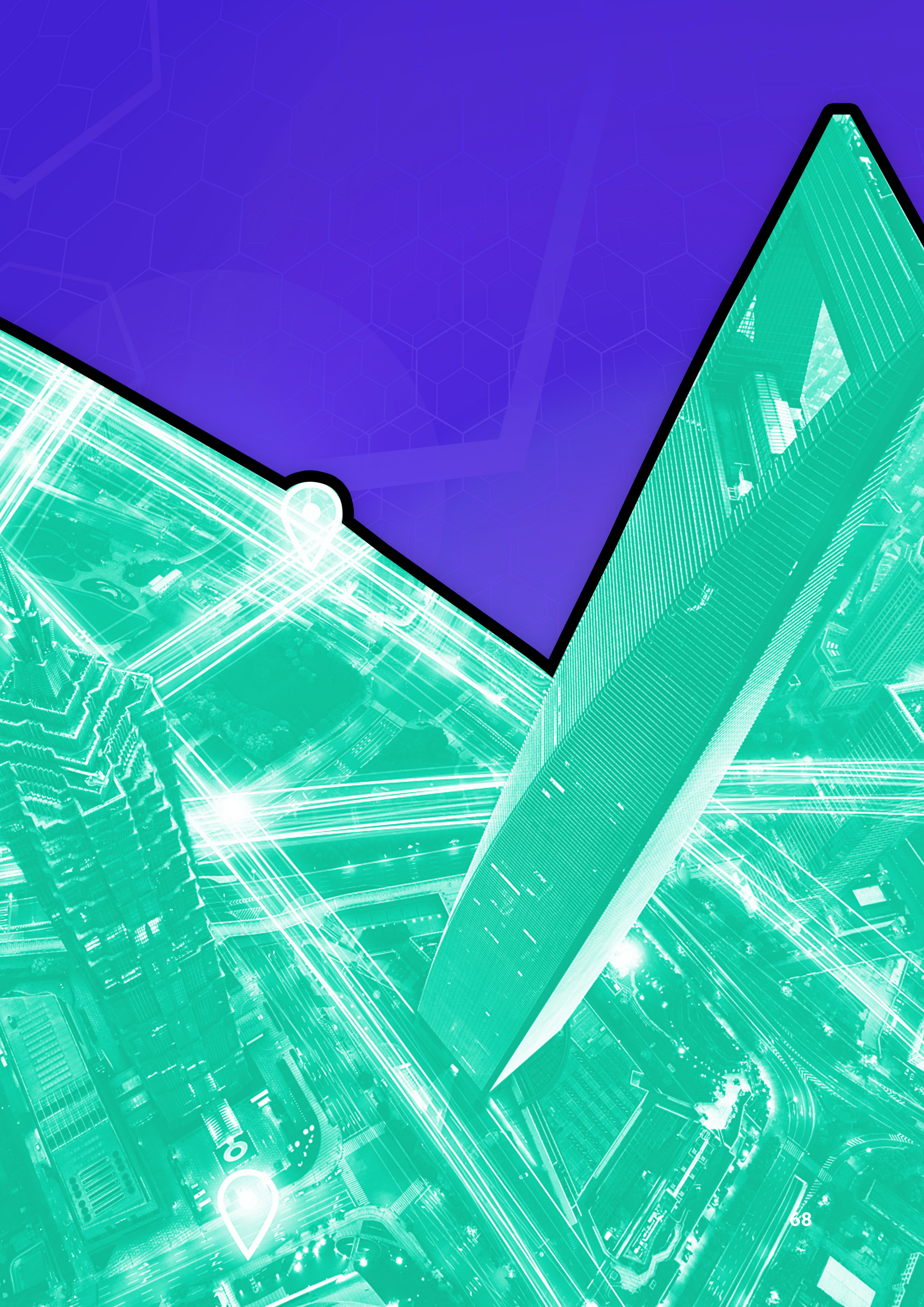
Because IoT is a system of systems, rare is the organization that has expertise across all the relevant domains. IoT platforms exist to help businesses overcome technical challenges without the need to figure it all out in-house.

For example, your organization might be really good at building hardware and decide that you want to make your hardware “smart”. **Instead of the expensive and time-intensive process of hiring software developers to build everything in-house, you can instead use an IoT platform to get up and running quickly and more cost-effectively.**

However, there is a tradeoff. IoT platforms that save you time may cost more in the long run depending on how they’re priced. This is because they charge use-based and/or subscription fees that can add up over time. But you still get the benefit of significantly lower up-front costs (no CapEx).

IoT platforms that are inexpensive up front will likely cost you time. This comes back to the same point in bold above, the less you spend the more work you’ll have to do on your own, which takes time.

In the next chapter we’ll explore how to choose an IoT platform.



Choosing an IoT Platform

How do you know which IoT platform is the best? As much as we'd like to give you a simple answer, as with most things, it depends. O'Reilly Media puts it nicely:

> ***"Each industry vertical — healthcare, manufacturing, energy, and banking, to name a few — will present its IT and OT specialists with particular conditions and problems to solve. Municipal police and fire departments, for example, will depend on a platform that ensures communication between field operations and command centers. Energy and transportation companies will search for ruggedized solutions that will protect field assets from harsh environmental conditions. Banking IoT platforms will demonstrate robust encryption and security features that protect internal and consumer communications and transfers."***
>
> — Evaluating and Choosing an IoT Platform

That being said, there's a key distinction between enterprise IoT platforms and consumer IoT platforms. Consider: is your application enterprise (such as oil and gas, manufacturing, or asset management) or meant for consumers (such as smart home applications or wearables)?

Enterprise IoT platforms and consumer IoT platforms can differ significantly due to their different needs. For enterprise IoT platforms, a failure in the system can be extremely high-stakes, perhaps costing millions of dollars or even lives. For consumer IoT platforms, a failure might simply be an inconvenience to the end-user.

And even within enterprise or consumer segments, applications can have very different platform needs. However, despite the great variance in IoT applications, there are some common elements that are critical to consider when evaluating the best IoT Platform for your application:

1. The Stability of the Platform

With so many platforms out in the market, it's likely that some will fail. It's important to choose a platform that's likely to be around for several years, otherwise your investment might go to waste if the platform provider folds.

Ask about current and past customers. If they don't have any, that's probably not a good sign.

2. The Scalability and Flexibility of the Platform

Your needs are going to change with time. Make sure that the platform works when you're small and just beginning, but will also work when you're (hopefully) large and growing fast.

In addition to being scalable, the platform should be flexible enough to keep up with rapidly changing technologies, protocols, or features. Flexible platforms are often those that are built on open standards and that commit to keeping pace with evolving IoT protocols, standards, and technologies, as well as offering third-party integrations and robust APIs (APIS are covered in the next chapter).

It's also important that the platform is network agnostic. This means that it can integrate and work with all major tech systems out there, rather than be locked into one vendor.

3. The Past Work of the Platform Provider

As mentioned above, IoT applications can vary greatly. If the platform provider has done previous work that is similar to your application, that's a good indicator that they can meet your specific needs.

However, note that it need not be an exact match. If you're building a smart agriculture application, for example, you might look for a use case with similar characteristics. That would be an application that also involves hundreds or thousands of sensors/devices generating data, a similar connectivity (such as LPWAN), and applied data analytics to create useful insights.

4. The Pricing Model and Your Business Case

Make sure the platform provider is transparent in their pricing; some will show an introductory rate and then hike that up significantly when you actually go to sign up.

Also, how are you going to be selling? If you're doing a subscription model, then it makes sense to pay a subscription for the IoT platform service, since you can wrap the costs into the pricing. However, if you're selling hardware, it might make more sense to pursue a platform option with an upfront license so you can wrap that into the development costs of the hardware product.

5. How Does the Platform Provider Handle Security?

Security is absolutely critical to any IoT system and an IoT platform must have security built into every layer.

When you ask about security, be on the lookout for: device-to-cloud network security, user app-to-wireless network security, cloud security, device security (including authentication and up-to-date certificates), application authentication, data encryption, data protection (at rest, in transit, and in the cloud), secure session initiation, and concrete plans for updating security, including via over-the-air (OTA) communications.

6. Time to Market

In the previous chapter, we saw that one of the biggest advantages of using an IoT platform is that it speeds up the time to market. Ask for a realistic estimate of how long it will take to get to market and how the platform provider intends to support you during that journey.

7. Data Analytics and Data Ownership

The value of the Internet of Things is in the data. Data can provide actionable insights into operations or simple day-to-day activities to reduce inefficiencies or improve experiences. You should look for basic descriptive analytics, visualization, diagnostics, predictive analytics, and perhaps even machine learning tools. We'll cover analytics and machine learning in the coming chapters.

Also, make sure to ask who owns the data. If the answer isn't a simple, "you own the data generated by your products", this is a big red flag because, again, the value of the Internet of Things is in the data.

8. Does the IoT Platform Provider Care About You?

In addition to all of these questions, you should be asking, also take note of the questions that they ask you. Do they ask about your budget, timeline, expectations, use cases, etc.? Do they seem like genuinely nice, caring people?

This is one of the most critical considerations. A platform provider that cares about you and your success will go the extra mile and make up for any areas in which their platform might be lacking.

APIs

If there’s one thing you’ve learned so far, it should be that a complete IoT system requires many different components all working closely together. We’ve explored the hardware that collects data, the connectivity that sends that data, and now the cloud and IoT platforms that ingest that data to make it useful.

However, even at just the cloud level there is a need for systems to communicate and work together and that’s what Application Program Interfaces (APIs) make possible. This is especially important for programs because they can be written in different languages, so APIs provide a means for different programs to overcome the “language barrier”.

In addition, APIs mean that users of your system don’t need to leave your system to use another organization’s application. For example, by using weather.com’s API, you can request current weather data and display it on your site or app for users. That way users can get weather information without having to leave your site or app and go to weather.com.

Also, APIs reduce complexity. When you use an API to request something from an application, many complex processes occur behind the scenes that you don’t have to worry about. You just get whatever it is that you requested in return.

A great example of this is the Alexa Voice Service API. Individual developers can’t build Natural Language Processing like Alexa, but instead they can use Alexa’s API to make tools based on it. So rather than needing to figure out how to take speech and understand the meaning (which is really hard), developers can focus on cool new applications that involve voice control.

Key Takeaway

APIs are hugely important to the Internet of Things. APIs allow companies to focus on their own expertise, plugging in the tools and programs of other companies as needed to create an IoT product/service that's greater than the sum of its parts.

And this also means that you can build a business by creating an API that's extremely valuable for other organizations to use. If you're a city, you may have data on foot traffic or vehicle traffic that could help local businesses or create new services to provide value to residents.

Data Analytics vs. Machine Learning

With all the hype around machine learning, many organizations are asking if there should be machine learning applications in their business somehow.

In the vast majority of cases, the answer is a resounding no.

As you learned a few chapters ago, one of the major benefits of the cloud is that it enables you to leverage virtually infinite storage and processing power to gain critical insights from the data your sensors/devices will be collecting. Both data analytics and machine learning can be powerful tools in doing so, but there's often confusion on what they actually mean and when is best to use one or the other.

Later we'll explore the value of machine learning in greater depth, but at a high level, machine learning takes large amounts of data and generates useful insights that help the organization. That could mean improving processes, cutting costs, creating a better experience for the customer, or opening up new business models.

However, most organizations can get many of these benefits from traditional data analytics, without the need for more complicated machine learning applications.

Traditional data analysis is great at explaining data. You can generate reports or models of what happened in the past or of what's happening today, drawing useful insights to apply to the organization.

Data analytics can help quantify and track goals, enable smarter decision making, and then provide the means for measuring success over time.

So When Is Machine Learning Valuable?

The data models that are typical of traditional data analytics are often static and of limited use in addressing fast-changing and unstructured data. When it comes to IoT, it's often necessary to identify correlations between dozens of sensor inputs and external factors that are rapidly producing millions of data points.

While traditional data analysis would need a model built on past data and expert opinion to establish a relationship between the variables, machine learning starts with the outcome variables (e.g. saving energy) and then automatically looks for predictor variables and their interactions.

In general, machine learning is valuable when you know what you want but you don't know the important input variables to make that decision. So you give the machine learning algorithm the goal(s) and then it "learns" from the data which factors are important in achieving that goal.

A great example is Google's application of machine learning to its data centers last year. Data centers need to remain cool, so they require vast amounts of energy for their cooling systems to function properly. This represents a significant cost to Google, so the goal was to increase efficiency with machine learning.

With 120 variables affecting the cooling system (i.e. fans, pumps, speeds, windows, etc.), building a model with classic approaches would be a huge undertaking. Instead, Google applied machine learning and cut its overall energy consumption by 15%. That represents hundreds of millions of dollars in savings for Google in the coming years.

In addition, machine learning is also valuable for accurately predicting future events. Whereas the data models built using traditional data analytics are static, machine learning algorithms constantly improve over time as more data is captured and assimilated. This means that the machine learning algorithm can make predictions, see what actually happens, compare against its predictions, then adjust to become more accurate.

The predictive analytics made possible by machine learning are hugely valuable for many IoT applications. Let’s take a look at a few concrete examples.

Machine Learning Applications in IoT

Cost Savings in Industrial Applications

Predictive capabilities are extremely useful in an industrial setting. By drawing data from multiple sensors in or on machines, machine learning algorithms can “learn” what’s typical for the machine and then detect when something abnormal begins to occur.

A company called **Augury** does exactly this with vibration and ultrasonic sensors installed on equipment:

> ***"The collected data is sent to our servers, where it is compared with previous data collected from that machine, as well as data collected from similar machines. Our plattform can detect the slightest changes and warn you of developing malfunctions. This analysis is done in real-time and the results are displayed on the technician's smartphone within seconds."***

Predicting when a machine needs maintenance is incredibly valuable, translating into millions of dollars in saved costs. A great example is Goldcorp, a mining company that uses immense vehicles to haul away materials. When these hauling vehicles break down, it costs Goldcorp $2 million per day in lost productivity. Goldcorp is now using machine learning to predict with over 90% accuracy when machines will need maintenance, meaning huge cost savings.

Shaping Experiences to Individuals

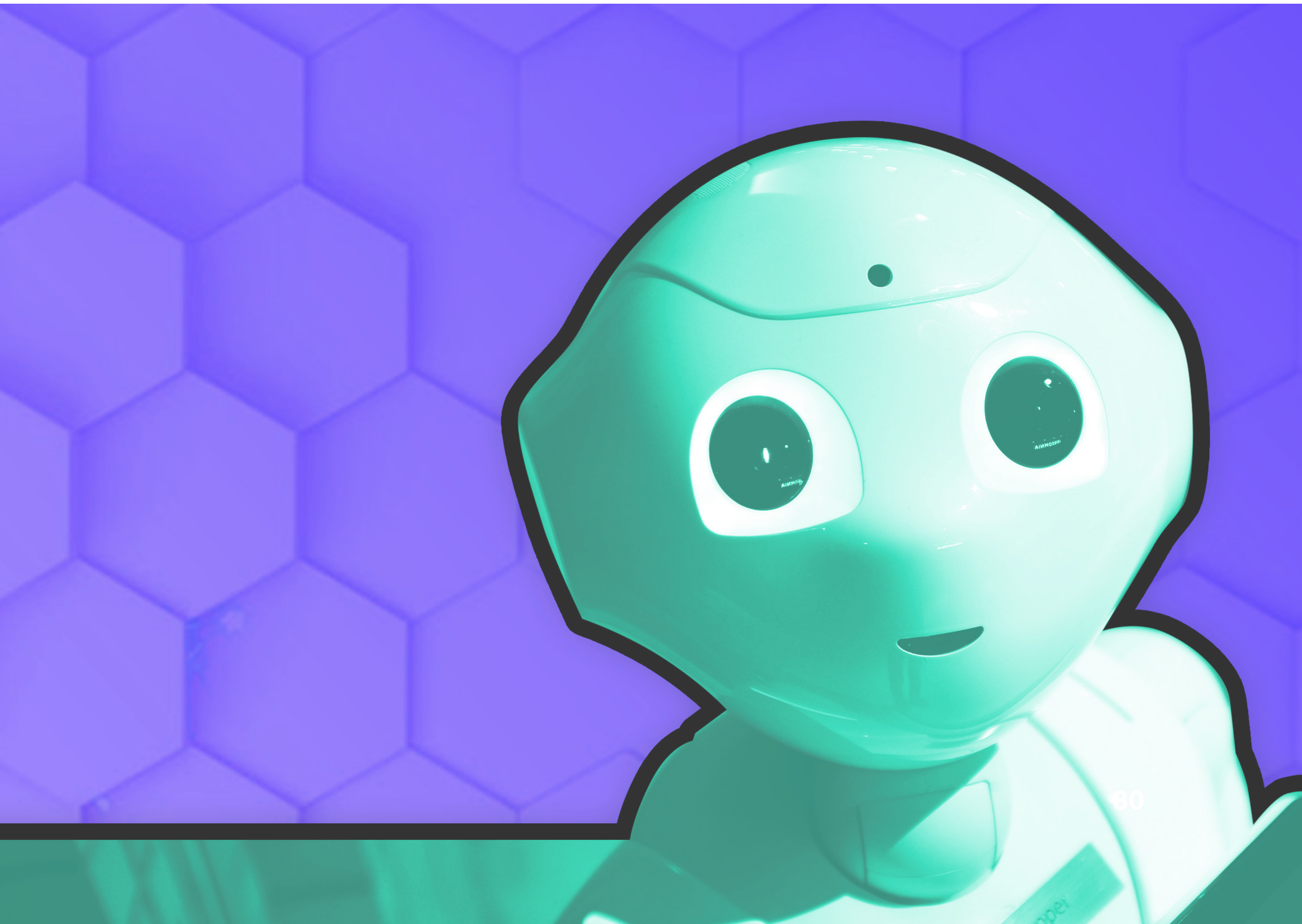
We're actually all familiar with machine learning applications in our everyday lives. Both Amazon and Netflix use machine learning to learn our preferences and provide a better experience for the user. That could mean suggesting products that you might like or providing relevant recommendations for movies and TV shows.

Similarly, in IoT machine learning can be extremely valuable in shaping our environment to our personal preferences. The Nest Thermostat is a great example: it uses machine learning to learn your preferences for heating and cooling, making sure that the house is the right temperature when you get home from work or when you wake up in the morning.

Key Takeaways

The use cases described above are just a few of the virtually infinite possibilities, but they're important because they're useful applications of machine learning in IoT that are happening right now. However, to reiterate, traditional data analytics are usually good enough for most IoT applications. Don't be fooled by an IoT platform selling you on its machine learning capabilities when you're just trying to look at trends over time to measure and improve your efficiency.

To make one final, critical point: with both traditional data analytics and machine learning, you need data. Gaining and maintaining large sets of clean, relevant data is an essential prerequisite to unlocking all the value that both data analytics and machine learning have to offer.



5 User Interface & User Experience in IoT

Introduction to UIs & UX for IoT

So far we've covered the sensors/devices that are out in the world collecting data and performing actions. We've covered the connectivity that enables those sensors/devices to send data to and receive data from the cloud. And in the previous section we saw how that data is ingested and transformed to provide valuable insights and automate processes.

But for any given IoT system, there needs to be a way of interacting with it. And just as we saw with sensors/devices, connectivity standards, and IoT platforms, there are many options you can pursue depending on your specific application and business needs.

> ***"We live in a time full of opportunity for imaginative individuals. In our lifetime, we will witness the emergence of more and varied forms of human-computer interaction than ever before."***
>
> \- Learning and Thinking with Things by O'Reilly Media

User Interface

Users need a way to view and understand the data captured by IoT. That's where the user interface comes in. In the simplest terms, a user interface (UI for short) is the means by which a user and a computer system interact. Many think of UIs as just software or apps on phones and computers, but a user interface could be anything from a smartwatch to voice-controlled Amazon Echo to the buttons on a smart tractor dashboard.

Apple pioneered the first graphical user interface (GUI) in 1983 with the introduction of Lisa. A graphical user interface is a visual way of interacting with a computer using items like buttons, windows, and icons. This meant that people didn't have to learn complex command languages to interact with computers and thus made the computer more accessible to everyday users. When they made the leap to touch interfaces and smartphone technology in 2009, Apple helped to further open up the door for new types of interfaces.

When it comes to mobile interfaces, there are a few important distinctions that you should be aware of:

Native Apps

Native apps are what most people think of when they think of mobile UIs. Native apps are applications that you download directly onto your phone. The advantage of native apps is that you have greater access to the phone's capabilities and can create a better overall user experience (we'll talk more about user experience below). The disadvantage is that they can take more time and resources to build, particularly because you need to build for both iOS and Android (iOS is the operating system created by Apple for iPhones and Android is an open-source operating system from Google) since they're not compatible with each other.

Web Apps

Like a website, a web app is accessed by going to a certain url (e.g. https://instagram.com). However, while websites are largely informational (like Wikipedia), web apps are built to have certain functionalities (like controlling a device remotely). The advantage of web apps is that they can work on both iOS and Android because you're just using a web browser instead of actually downloading something. Also, because you don't need to download anything, it can make it easier to get into the hands of users (just send them the url link). The disadvantage is that you have somewhat limited access to the phone's full capabilities (like the inability to send push notifications) and less control over the overall user experience.

Hybrid Apps

As the name implies, hybrid apps are between native apps and web apps. You still download something, like a web app, but when you open the app it is essentially opening a web page meaning that it can act like a web app. This can be a good option if you know you'll be creating native apps eventually, but you want to get a minimum viable product into the hands of users early, and can therefore benefit from the speedier development offered by web apps.

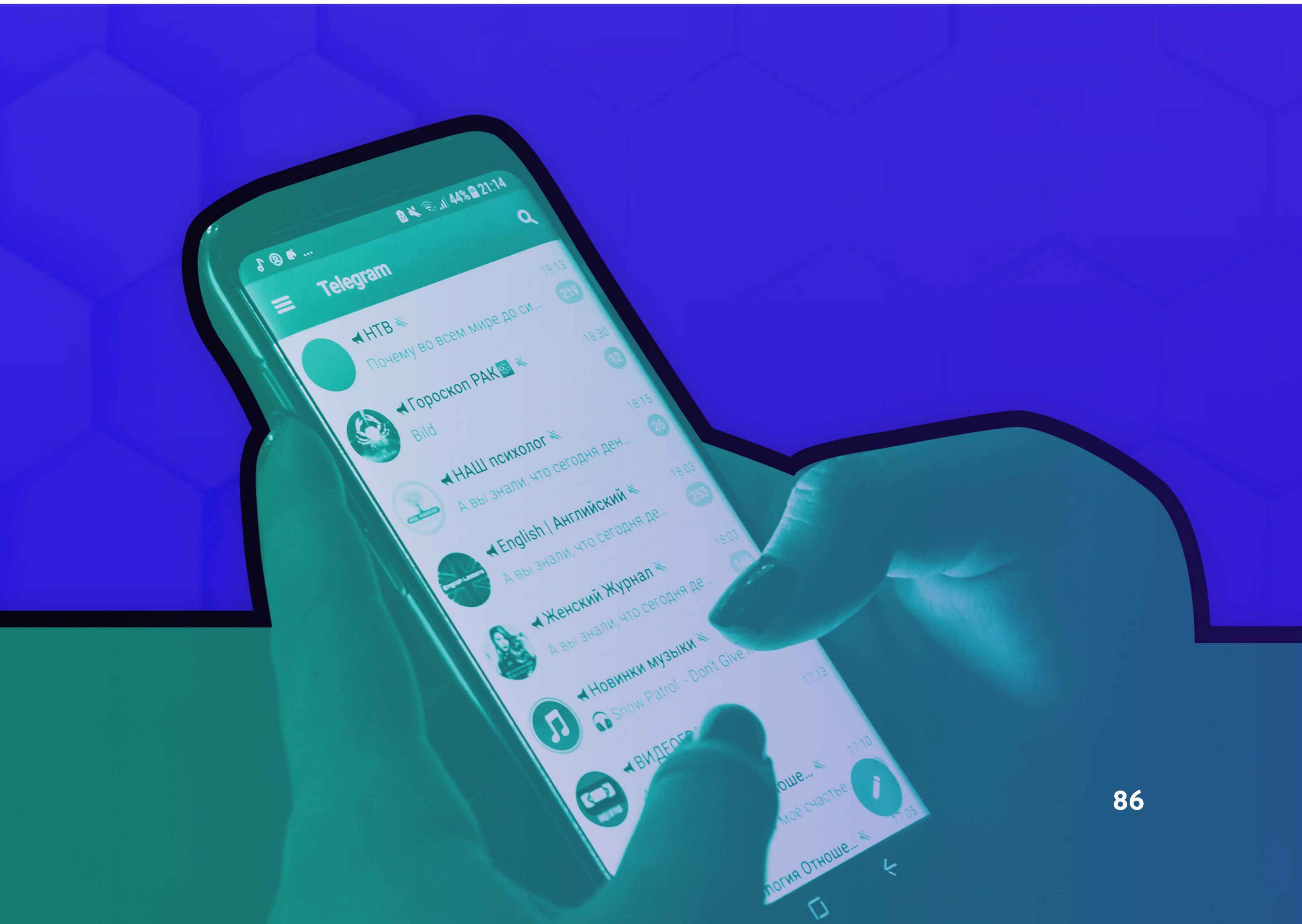
Beyond Mobile Apps

The above distinctions are for mobile apps, but as we mentioned above there are many different types of user interfaces beyond just mobile. Let's use Nest as an example to show how user interfaces have evolved past phones and computers.

In the picture on the left, you can see Nest's thermostat. It's controlled by touch, displays relevant data, and it's connected to the internet. It is therefore a user interface because it's a way in which a human interacts with a computer (i.e. setting the temperature).

On the right, we have Nest's App, which is a native app like we covered above. Both of these are user interfaces that must be designed to suit their specific medium and user's needs when interacting with that medium.

User interfaces are constantly evolving, with entirely new interfaces made possible by technology. In 2005, Blackberries were cutting edge technology. Now, we have voice assistant interfaces entering our homes with Google Home and Amazon Alexa. Technological change will continue to enable new ways of interacting with objects and IoT systems, but each of these ways will still be a user interface.



User Experience

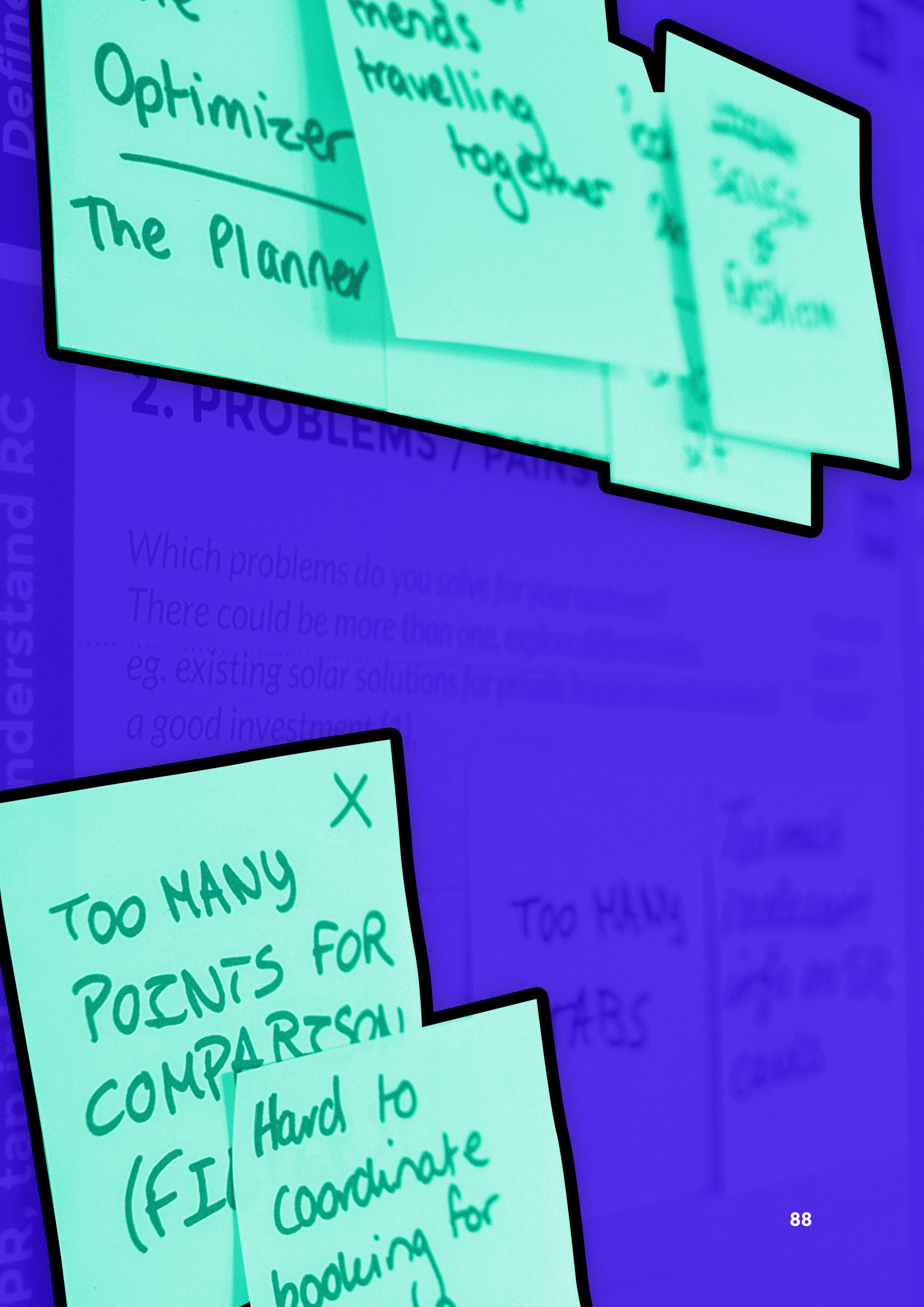
Returning to our Nest example, both of the interfaces pictured are only part of the overall user experience of Nest. Per designer and author Don Norman, user experience is "all aspects of the end-user's interaction with the company, its services, and its products."

For Nest, that's everything from their native app and website to the actual thermostat and the packaging it comes in. It's the user's journey from the moment they make an account, to buying the thermostat, to opening the package, to installing it, and to every other interaction they have with it after installation and until they get eventually get rid of it. This includes things like notifications and alerts, which are critical pieces of many IoT systems and count as part of the overall user experience.

The most universal example of a designed user experience is the Apple store. Every interaction, every product placement, every interior design choice, has been chosen to provide you with the best Apple experience. The staff are called geniuses, the wait area has the latest apple gadgets, and the most expensive products are front and center. All of this is by design.

The important takeaway here is that user interfaces do not exist in a vacuum. As you're pursuing your specific application, make sure to consider who your users are and every step of their overall interaction with your system.

In the next section we'll examine some best practices as you approach UIs in IoT.


Optimizer
The Planner
friends travelling together
2. PROBLEMS / PAINS
Which problems do you solve for your
There could be more than one
eg. existing solar solutions for
a good investment (1).
X
TOO MANY POINTS FOR COMPARISON (FI
Hard to coordinate for booking
Too MANY TABS
Understand RC
Define

Key Considerations for UIs

In the last chapter you learned what a user interface is and what user experience means. But just having a UI isn't enough. No matter how incredible your underlying technology (be it hardware, connectivity, or software), if humans can't easily access, control, and interact with your IoT solution, it will fail. In this chapter we cover some of the key considerations for the UIs and UX of your IoT solution.

Who Are Your Users?

Although this may seem like an easy question to answer, the implications are often more complex than you might realize. This is because you not only need to identify who the different users are, but then you need to determine how each user can and should be able to interact with your IoT solution.

For example, let's say that you have an asset tracking application that places trackers on thousands of vehicles on an auction lot so you know exactly where they are at all times. You want employees on the auction lot to be able to set geofences so they can get an alert if certain vehicles move outside of those areas. But do you want all of the employees to be able to set those geofences? It could get pretty confusing if dozens of employees are all setting their own geofences. So you might want to have an admin user who can take actions like setting geofences and adding contacts to get those alerts, and then a regular user who has more limited access to features.

However, as you put this IoT solution in place, you realize that you can add a lot of value if you open this up to your dealers and transporters. They may not be direct employees of the auction lot, but if they can find their cars directly, they won't need to bother your employees to do so. But to provide this to the dealers and transporters, you're going to need to make this information publicly available. Can your competitors use this information to gain an advantage? Could bad actors use this information to steal vehicles? Again, you need to consider how these public users should be able to interact with your IoT solution.

User stories are an extremely useful tool for gaining clarity here. A user story is a sentence that takes the form: "As a [role], I want [feature] so that [reason]". Though it may seem like unnecessary work, taking the time to write out as many detailed user stories as possible is well worth it.

Alerts & Notifications

Many IoT solutions provide the majority of their value passively rather than actively. As we saw in the example above, it's much more valuable to get notified when cars move where they shouldn't, rather than having to manually check each car to make sure it's where it's supposed to be.

Although alerts and notifications aren't necessarily a part of your UI (since the user might be getting an alert outside of your UI, like in a text or email,) they are still a crucial part of the overall user experience. When do users get alerts (one minute after a trigger event? One hour? Can the user configure this themselves?) How do users get alerts (email, text, push notification, phone call?) What happens if a user doesn't react to an alert in a certain period of time?

Much of these considerations circle back to who your users are and how they can and should be interacting with your overall IoT solution.

Responsiveness of the UI

Many IoT solutions use a dashboard to provide the user with information and insights. These dashboards need to be responsive, which means they work on whatever device, browser, or operating system that the user is using.

You'll want to do extensive testing across different browsers like Chrome, Safari, Firefox, and Internet Explorer; different tablets like the iPad and Microsoft Surface; and different phones including varied models of iPhones and Androids. By using a responsive design system, your dashboard should work flawlessly across all of these. In the mobile age, users should be able to access their information anytime, anywhere.

6 The Future of IoT

The Future of IoT

Congratulations! You've finished the **Introduction to IoT** eBook. At this point you should have a solid foundation in IoT, what it means, and how to pursue building and deploying real IoT solutions for your organization.

You might also be left feeling like you now have more questions than when you began, and that's a good thing. It's the pursuit of these questions that will lead you to build something that's never been built before, creating a better future for your organization and for all of us.

The future isn't set. The future is the culmination of all the choices we are collectively making today. While it's inevitable that we'll continue to connect our things, people, and environments to make them more intelligent, efficient, and user friendly, it's not inevitable which problems we focus on solving and who benefits from those solutions.

We're committed to amplifying human potential and we see IoT and related technologies as essential tools to make that happen. We strive to make those tools more accessible to everyone and to share the knowledge we've gained through our decades of experience, so that all businesses are empowered to unlock human value in and out of their organization.

Thank you for reading! If you still have a burning question, if you think there's an opportunity to work together, or if want to just say hi, please don't hesitate to reach out.

iot for all

LEVEREGE

Thanks for reading our eBook!

Like what you saw? Want to find out more?
Connect with the experts at Leverege!

Connect with Experts