IoT Design Principles

Executive Briefing

An executive briefing offered by Ardexa to help general business executives improve their understanding of IoT implementations and operations

Authors

George Cora – CEO and Director of Ardexa Pty. Ltd.

David Mohr – CTO and Director of Ardexa Pty. Ltd.

Property of Ardexa Pty. Ltd.

Released 11th April 2016

Summary

It's time. Let's stop rushing into IoT implementations and instead approach IoT design holistically and professionally. A robust design must but built on a set of rock solid fundamentals to ensure the necessary security and scalability characteristics are achieved, while being flexible enough to facilitate continuous improvement.

Ardexa has identified 7 key design principles when dealing with *Things* and the *Internet*. These are:

- ***Things require software to manage sensors and handle asynchronous communications***
- ***Things must always initiate connections to the Internet and must not accept inbound connections***
- ***Each and every Thing must be identified and authenticated, in a secure, reliable and scalable manner***
- ***Things must provide the ability to be controlled remotely, including the ability to transfer files***
- ***Metadata and flexible data formats are fundamental to growing and scaling IoT systems***
- ***Cloud services provide cost-effective scalability and always on reliability for IoT services***
- ***Cloud services must integrate seamlessly with existing IT systems***

Implementing these principles requires a deep technical understanding of security, communications, development and operations (DevOps).

Software Development Kits (SDKs) targeting IoT developers only cover a small portion of an end-to-end IoT solution and still require time and experienced staff to implement.

The Ardexa team has focused their efforts on building the leading end-to-end IoT solution allowing you to focus on what matters most: your device and the insights it can provide.

Definitions

Definitions of the terminology used in this whitepaper are as follows:

1. ***Gateway/Device***: A small form, single board computer similar to a Raspberry Pi. These are mostly ARM or x86 based processors, with on-board storage, RAM and peripherals. They are low cost and low power, yet run various forms of Linux or Windows.
2. ***Sensor/Actuators***: A simple or complex electronic component designed to carry out a well defined task. Actuators or sensors may contain a micro controller, and some may be controlled by a gateway. A sensor usually simply collects information, whereas an actuator can control equipment such as opening/closing valves, relays, lights or related equipment.
3. ***Thing***: A Thing is defined as the software and hardware that encompasses both the gateway and sensor.
4. ***Metadata***: Metadata is "data about data". It may be a stream of data that describes elements such as the source of the data, the units used in the data stream (eg; Celsius, Fahrenheit, Newtons, etc), the author or device producing the data and perhaps the structure of the data (eg; data separated by commas, tabs, etc).

Data format: Some data may be a single integer describing something like a temperature reading. Another piece of data may be 1 or more strings. Other again may be values separated by commas. Data formats can vary widely depending on who or what has produced the data.

Introduction

IoT implementations may be as simple as a thermometer being connected to the Internet or as complex as monitoring every facet of a multi-story building. In the early years of the Internet of Things (IoT) movement, developers focused tightly on specific outcomes or use cases, without too much regard for how those use cases might evolve in the future.

This is not a problem if the lessons learned are carried into the design of the next project, however all too often, the problems brought about by earlier decisions are compounded by persisting with a faulty design. Future capabilities are then entirely dependent on a design and implementation which likely never considered these newer scenarios. More serious consequences will see a marked increase in the number of IoT security incidents, particularly as more Things come online. These newer things run the risk increasing the potential reach of an attacker beyond simply bits and bytes and into the physical world.

For those not well versed in information security, protecting the IoT and associated cloud services can be a confronting issue. Design decisions made across many businesses today are setting up future success, or failure, based on the quality of IoT infrastructure being selected and implemented. Today's executive needs to be very sure that the team and resources dedicated to such issues truly represent the experience and maturity of the information security industry.

Designing IoT implementations should take the same path that the automotive industry took with safety and environmental design. Over many decades, the automotive industry began to adopt the fundamental principle that wherever possible, designs should strengthen safety and environmental credentials. Somebody designing a chassis had to reduce the weight for fuel

efficiency, while simultaneously increasing strength and shock absorption to enhance its performance in crash tests. This resulted in a continuous improvement in safety and environmental performance, underpinned by a mindset change for all those involved in the design. Many thousands of elements were changed over the past few decades to enable such an improvement, and it became ingrained in every activity associated with automotive design. The result is a vehicle today that is many times safer than those our parents drove.

Similarly, IoT implementations must now be designed in an equally holistic manner. A disciplined approach to system design, with the best professionals involved, is required. A process encouraging reviews and iterations, leveraging sub-system providers to enhance designs, and more are required. Only such a holistic approach can bring about the changes required to meet the rigorous security standards required when dealing with customer data.

The IoT industry is converging on a set of accepted principles to enhance overall security, functionality and cost-benefit. Many of these principles have been derived from the broader technology industry, however the practicalities of the IoT means that some principles must be reassessed.

The fundamentals presented below help make an overall design robust through practice, not just on paper. Compromising any of these fundamental principles will result in, at best, a substandard implementation, and at worst catastrophic failure.

PRINCIPLE 1: *Things require software to manage sensors and handle asynchronous communications*

Sensors and actuators are simple devices designed to do one thing. They are not designed to talk to the cloud, handle security, manage certificates or cache events. The Internet is not a one-way flow of information. To create a true *Internet* of things, sensors and actuators will require a gateway to handle their communications.

These gateways are likely to be in the form of single board computers, such as the Raspberry Pi, running a lightweight piece of software commonly known as an agent. These tiny machines only use a few watts of power, but are identical (functionality wise) to an early desktop computer. The gateways offer the functionality required to collect data from a variety of sources (e.g. radio signals, serial connections, I/O ports), manage metadata, securely transmit data over the Internet, cache data when the network is unavailable and provide systems to troubleshoot any issues that may arise.

Requests and replies will be sent to and from the gateway. Combined with the fact that the gateway may have an inconsistent or unreliable connection to the Internet, this means that the agent, and consequently the associated cloud service, must be able to handle asynchronous operations and communications in order to achieve the resilience required to manage a fleet of Things reliably.

Asynchronous communication essentially means that a device does not need to be online in order to communicate with it. Just like emails sitting on a server waiting to be downloaded, asynchronous requests sent to a Thing will wait on the cloud server, ready for retrieval and

processing next time the Thing is online.

PRINCIPLE 2: *Things must always initiate connections to the Internet and must not accept inbound connections*

There are two very good reasons why you should *not* attempt to contact your Things directly:

1. It is a very poor security practice to unnecessarily expose services to the Internet.
2. You need to know the address of your Thing before you can even attempt a connection.

Security

Traditional Internet facing services, such as email or websites, require constant upkeep and surveillance to ensure their protection and continued operation. Remote IoT gateways will have to operate reliably under somewhat adverse conditions, often with a high level of neglect. They are likely to have software that is outdated and/or configurations that have not been adjusted for long periods of time. Allowing incoming connections from the Internet brings with it an element of risk from mild (the device is taken offline) to extreme (the device is used as a launchpad to reach inside your private network). There is very little middle ground, and subtle mis-configuration can render not just one device, but your entire infrastructure insecure.

Secure tunnels provide a robust and well-respected alternative. This means that communications can be established without the need for a hole to be 'punched' though the device's firewall.

Locating your device

Anytime a machine connects to a network, it does so using an IP address. These addresses can be assigned either statically, so that a machine has the same address every time it boots up, or dynamically, so that machines may come and go and share a pool of addresses. Static addresses are not always an option when connecting to the Internet and managing dynamic addresses by having the gateway update a registry somewhere requires almost as much work as setting up a secure tunnel.

A secure tunnel is configured by the Thing ***after*** it has connected to your Internet service. Therefore, when addressing your Thing, you need only know its identity, which is either conveniently mapped to an established connection if your Thing is online, or requests are queued ready for the next time the Thing logs in.

PRINCIPLE 3: *Each and every Thing must be identified and authenticated, in a secure, reliable and scalable manner*

To properly manage your IoT deployment, one must be able to address each Thing individually. Consequently, every Thing must have a unique identity and a means of proving it.

While intuitive at first blush, this deceptively simple principle can be difficult to get right. There are a number of different systems for generating unique identities and most will suffice. However, proving this identity, and preventing duplicate IDs, is where things get interesting.

Authentication

If any Thing is allowed to connect to your Internet service and simply announce its identity, there is a very high risk that fraudulent devices will connect to your service. Therefore, some kind of authentication is required: either a shared secret, such as a password, or certificate-based Public Key Infrastructure (PKI) authentication. Whichever authentication system is used, always ensure that the communication channel is encrypted to mitigate the risk of interception of either the data stream or, where shared secrets are in use, the authentication process.

Authentication doesn't apply to Things in the same way as humans. For example, authenticating to an email service requires a user to remember their password, or carry a hardware token, and enter it as required. For a Thing to "remember" a password, it must be written down in a retrievable format making it a prime target for attack. And the risk only grows as the number of Things grows, especially if some of those things exist outside any physical security monitoring you have in place.

Passwords, pre-shared keys (PSK) and API keys are examples of shared secrets. One mitigation strategy for dealing with the aforementioned risk of loss is to never share the same secret between multiple devices. Otherwise, if one Thing is compromised, the entire fleet of Things will require updating, especially if you employ a predictable identity system, such as incrementing a number.

Digital certificates are essentially identity, authentication and encryption all rolled into one convenient package. For additional protection, there are hardware devices that can securely store your digital certificate for on-demand use. While full PKI systems take some work to initially set up, the major advantage over shared secrets is trust. The details of PKI trust is beyond the scope of this document, however the benefits include: digitally signing updates, authentication across multiple services using the same certificate, removing the need to add new accounts to your services and revoking Things when they become compromised.

Duplication

Provisioning will become a major step in any IoT system and the more automation that can be brought to bear, the better. However, there is always a risk of a duplicate identity appearing in the system. Duplicates accidentally generated when first assigning a device ID should be quickly recognized and discarded. Where a highly concurrent system of generating identities is required, an appropriate identity generation system, with a low chance of collisions, should be selected, e.g. UUID. Old devices being reactivated, copied configuration files and faulty device templates all run the risk of duplicate identities attempting to use the system. Procedures for detecting and dealing with these duplicates is an important step in avoiding unexpected consequences.

PRINCIPLE 4: *Things must provide the ability to be controlled remotely, including the ability to transfer files*

The Internet of Things would hardly be a revolution if people, or Things, could not communicate and *interact* with other Things. The benefits are many: manage home security on the go, be alerted to a possible vehicle maintenance issues that have been remotely diagnosed, allow thermometers and battery monitors to decide when to activate heating or cooling systems, and

so on.

Remote control also means that the "smarts", that ability to automatically decide what changes your Thing, can live in the cloud or on the device. The choice is yours. However, utilizing the cloud allows for simpler devices, more complex decision making tools (e.g. machine learning) and a single point of contact for controlling all the peripherals in a given network of Things.

Furthermore, the benefits of remote control extend beyond just the original purpose of the Thing and into management issues such as troubleshooting, configuration updates and software patches.

All of these scenarios, from activating a light to patching your software, call for an exchange of information and unfortunately, Things that only allow one-way communications will be abandoning a wealth of functionality. When designing an IoT system for remote control, it is crucial to include the ability to upload and download files. Continuous improvement of your Things will require software updates, thus ignoring the ability to transfer files will simply defer serious thinking and architectural changes that will most likely be required at some point in the future anyway, however delays will likely introduce additional effort and risk.

PRINCIPLE 5: *Metadata and flexible data formats are fundamental to growing and scaling IoT systems*

A measurement has no meaning without context. Metadata is critical in communicating the context regarding a measurement including information about the target object, the property being measured, the units being used and acceptable values. And what if this information were to change? What if another sensor was brought online to monitor some other property of the same target object? What if, instead of a number, this sensor wanted to send a photo?

It is neither desirable, nor possible to plan for every eventuality. Instead, designers should embrace change and create flexible systems that keep communication overhead to a minimum. Enabling your Thing to report metadata when required results in lower overall complexity since new data sources can be added in real time without costly software updates to the Thing or the cloud service, making the entire system more robust and adaptable.

For example, sending the metadata about a temperature reading does not need to happen every time a measurement is taken. Instead, updates should only happen when there is a change in circumstances, triggered by either the Thing (when it detects a change) or a request from the cloud service. However, if this sensor is attached to a moving object such as a refrigerated truck, then its location will be changing over time and should be recorded alongside the temperature.

PRINCIPLE 6: *Cloud services provide cost-effective scalability and always on reliability for IoT services*

Technology moves quickly and is getting cheaper. Tools and systems that were expensive and time consuming to manage are now cheap and available on-demand. IoT infrastructure builders must take advantages of these advances, in either public or private clouds, if they want to compete in this rapidly evolving market.

To build high speed, resilient services that deliver the results you need in the time frames you demand, there is a need to leverage open source and cost effective proprietary components that are impossible or impractical to replicate. Leveraging these solutions will enable to you to build an end-to-end platform for provisioning, managing and running your IoT system and ultimately deliver the results you're after.

PRINCIPLE 7: *Cloud services must integrate seamlessly with existing IT systems*

The level of integration required will vary from alerts being sent to internal ticketing system all the way through to full in-house platform hosting. The right mix of build and buy will support cost-effective, fit-for-purpose implementations that deliver project outcomes on time and in budget.

However, some IoT implementations will be dealing with sensitive client data or control access to private or confidential systems. These situations will call for even higher levels of security than normal and may need to bypass the public cloud in order to meet regulatory or compliance requirements.

In these cases, control of the Thing and any associated data must only be made available on corporate systems. The IoT for some companies will mean a big competitive advantage, and that kind of project requires that the data is managed in-house and not by third-party cloud providers.

Conclusion

To be successful, IoT implementations need to be guided by deep security experience, alongside the broader design team undertaking the project. Knowledge and experience leveraging from security implementations, data handling methods, communications, authentication, encryption, certificates and software component evaluations are critical inclusions.

At Ardexa, our perspective is that launching IoT initiatives does not necessarily mean a higher technology risk profile. There is a solid foundation of industry knowledge and experience in technology security to implement IoT initiatives successfully. The change the industry needs to make is approach the design in a holistic fashion, with a security mindset from the very start. Operations can be molded into the new world, with standards and methods to suit, and these 7 design principles will help light the road ahead.

The Ardexa team has been leading the security industry for many years

We want to take the hassle out of IoT connectivity.

We look for clients that are collaborative in nature and are aiming to lead the IoT world. We are ready to evaluate your specific needs or simply converse on what is possible. Our team members bring an unmatched depth of experience in security, connectivity, data collection and end-point management. This means we can deliver high quality work and advice in a timely manner. Please contact us to discuss your IoT initiatives.