Java Card for the Internet of Things

Building The Foundation Of End-to-End IoT Security

November 2018

AUTHORS

Sara Ellinger, NXP
Francois Ennesser, Gemalto
Volker Gerstenberger, Giesecke+Devrient Mobile Security
Mireille Pauliac, Gemalto
Ly-Thanh Phan, Gemalto
Florian Tournier, Oracle
Patrick Van Haver, Oracle

CONTACT

Karen Brindley, Operations & Marketing Secretariat, Java Card Forum
karen.brindley@javacardforum.org
www.javacardforum.org

CONTENTS

INTRODUCTION

Connected devices volumes are expected to increase exponentially in the upcoming years. Analyst data seem to converge and indicate that the number of things connected to the internet should exceed the number of humans by a factor of 8 come 2020.

While this growth brings transformative effects to several industries and to people's daily lives, it also induces an additional level of system complexity to the infrastructure that will handle device data. IoT systems need to be cognisant of a vast number of device types, interfaces, communication protocols, or device lifecycles.

In parallel, there is a strong imperative to be able to trust the data that gets acquired and acted on by IoT solutions. It is a well-known fact that Big Data is only as good, or as secure, as the small data that it is built on. The effects of corrupted devices or data on systems that make instant, analytics-based decisions can have a severe financial, material and human cost.

As a result, there is an increasing need for solutions that secure the source of data at the edge, creating end-to-end security up to the cloud, and help IoT customers simplify the overall security equation.

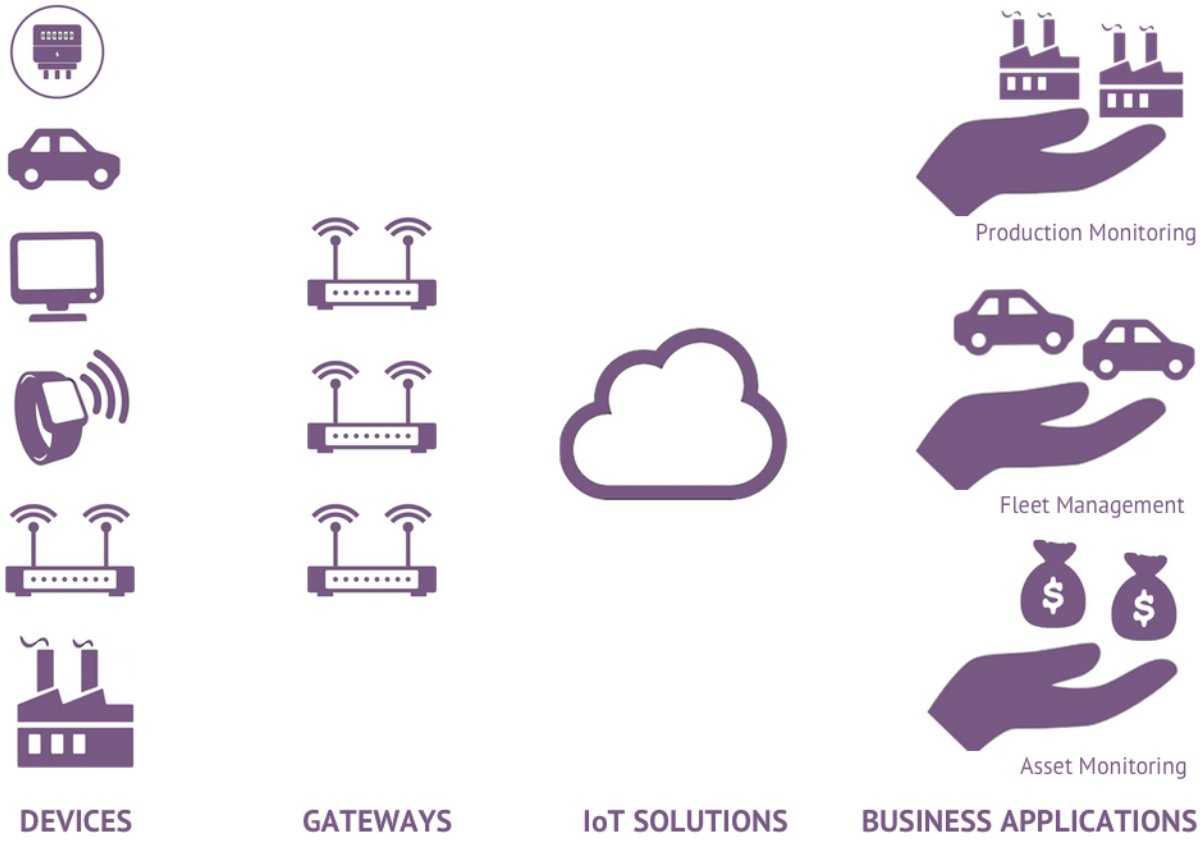
Fortunately, there is an extensive body of work already available detailing the security risks associated with Internet of Things systems. This document does not intend to replicate this work. Instead we will focus specifically on the security of IoT devices. We will make an effort to categorise some of the key challenges faced by IoT vendors at the edge of the network. We will also go over hardware or software solutions that can abstract a good portion of the IoT complexity from the customer. We will discuss what new use cases can be brought about by open, secure application platforms such as Java Card™ technology.

SECURITY CHALLENGES IN IoT

In this section we will enumerate some of the unique security challenges of IoT systems, with a goal to highlight how stronger edge security can contribute to their remediation. This list does not intend to be exhaustive, but offers a representative view of the changes induced by IoT and the related security threats.

TYPOLOGY OF IoT SYSTEMS

As a starting point, it is useful to define the typical components of an IoT system. A basic IoT architecture can be represented by the diagram below:



The **IoT solution** (typically a cloud service, but it could also be on-premise) connects to a variety of **vertical specific devices** that are commercialised by manufacturers. Its role is to manage devices' lifecycle, communication, aggregation of data, provide storage and analytics means for back-end applications. In most cases, the provider of the IoT solution is different to the provider of the device, although there is a trend toward vendor integration. In most cases, IoT solution vendors provide open source software components that need to be embedded in the device software stack and facilitate the connection to their service.

Devices may be connected directly to the IoT solution, or through a **gateway**. The role of a gateway is usually to act as a proxy to the IoT service for endpoint devices that may not have WAN communication capabilities or may have limited security and processing properties. Gateways and/or devices communicate to the IoT solution through a variety of protocols (HTTP, CoAP, MQtt), through APIs and interfaces that are generally vendor specific or proprietary.

Once device data is ingested by an IoT solution, it is analysed and processed, and either acted upon (a command or response is sent to the device) or passed on to **business applications** that will consume it and make operational sense of the information.

Although many IoT solutions adopt the same architecture, there is still a **lot of potential variance**. For example many industrial IoT applications (e.g. control of power substations in smart grids, assisted or autonomous driving, and industrial process control) involve real-time constraints, which may not be compatible with centralised application architectures that distinguish between a "field (device) domain" and an "application (server) domain". In such situations, the information needs to be processed along the communication chain between sensors and actuators devices, with information typically filtered at the source and processed at destination. Industrial IoT applications then end up being distributed and split into subcomponents developed to run on heterogeneous computing platforms : embedded devices, mobile apps, gateways or mainframe servers. We will see that this makes the concept of a Virtual Machine architecture capable to abstract the huge heterogeneity of embedded hardware platforms particularly attractive to IoT application developers.

Furthermore, **standardisation is only in its infancy**. Each vendor defines and implements their own security policies and procedures. This is an aggravating factor when trying to address some of the intrinsic security challenges of IoT systems.

OVERALL SYSTEM COMPLEXITY INCREASES ATTACK SURFACE AND THREATS

The most obvious security challenge facing IoT systems is **scale**. Those solutions differ from traditional IT environments, because of the drastically increased attack surface induced by billions of connected devices. As IoT systems connect devices to gateways to clouds to business applications, we see attack vectors emerge that were not featured in traditional systems.

For example:

- Physical attacks on device credentials
- Attacks on the device low-level software
- Attacks on the local link between devices and gateways
- Attacks on the communication between device and IoT service
- And much more…

This risk is further heightened by the **high motivation of attackers**: connected devices can represent high-value targets (e.g. automobiles, factories), or provide significant publicity. Unlike traditional IT systems, IoT applications typically involve actuators affecting the physical world, hence creating a link between Information Technology and so-called Operational Technology. This means that people's safety can be directly affected by attacking an Information System.

They may also be an easy target. IoT stakeholders may not have the **security maturity** of their counterparts in the IT space, or operational priorities may take precedence over security considerations. In other terms, the bridges IoT is building between Operational Technology and Information Technology are not yet solid enough to fully address the challenges brought forward by the multiplication of devices and data sources.

DEVICE CAPABILITIES AND COST CONSTRAINTS LIMIT CHOICES FOR SECURITY SOLUTIONS

The **perceived cost of implementing device security** may also frustrate attempts to strengthen the security of IoT systems.

Many solutions are available to secure the IoT Edge, both at the hardware and software level. They all come with an associated cost. Although that cost can be easily trivialised compared to the financial risk induced by a vulnerability, short term thinking may push users to forego or defer the acquisition of security technology and focus on the reducing the device's bill of materials, without considering the cost over the entire device lifecycle.

Incentive gaps also exist between the device provider and the owner of the solution responsible for the security.

Furthermore, **not all security hardening options may be available on all connected objects**. On very basic devices, any security option may be technically unviable or uneconomical. Repurposed "brownfield" devices can also typically not be upgraded with new security technology.

Overall, security is often considered as a premium option in a device bill of materials. As a result, edge security techniques tend to be applied in priority to limit or fringe usage scenarios. This situation weakens the security of the entire IoT system.

DEVICE HETEROGENEITY AND FRAGMENTATION CREATE SECURITY INCONSISTENCIES

Fragmentation of the device landscape is another risk factor. Even within a manufacturer's product range, **devices may implement different security properties** (different credential storage mechanisms, different software versions, different security hardware). This makes it challenging to implement consistent end-to-end security policies. It renders a holistic, one-size fits all approach to edge security inapplicable to this space.

Furthermore, IoT deployments at scale typically require **integration with existing devices** and networking infrastructures. Even if new devices and sensors are being rolled out, they are often based on hardware and software that were designed for smaller scale, closed reporting systems. Connecting devices that were not initially designed to be part of a wide, open network significantly increases the security and safety risks.

LIMITED UPGRADE CAPABILITY INCREASES THE RISK DURING DEVICES' LIFESPAN

Another unique characteristic of an IoT system is the **long lifetime of a device**. It may range from a couple of years (for example phones and wearables) to a couple of decades (for example utility meters). This has strong security implications: even if a security model can be designed to span heterogeneous hardware, it will have to factor time as another dimension, and feature a mechanism to keep software, security properties and policies up to date.

In this context, **devices need to be securely upgraded** or patched after issuance. The network topology also needs to be fluid to allow for device addition or replacement. Not all device and IoT infrastructures available today can provide this flexibility.

NON-ACCESSIBLE EDGE DEVICES LIMIT THE ABILITY TO DISCOVER A PHYSICAL ATTACK

Finally, a unique property of IoT devices compared to IT systems is the fact that **not all devices may be connected all the time.** Some devices may check-in occasionally and not be accessible for extensive periods. As a result, IoT deployments cannot rely on always-on connectivity to detect tampering. A device may be attacked or compromised while offline. Some processing capability and security measures are required at the edge, to detect issues and react accordingly.

METAMORPHIC ATTACK SURFACE

The security challenges listed in this section are only the tip of the iceberg. Other vectors of attack already exist, and new ones will be found before the devices deployed today finish their mission. Evidence of the security risks associated with IoT systems have been made visible through a recent string of well publicised attacks: the 2017 Mirai attack turning thousands of edge devices into botnets, or its 2018 Okiru variant for example. The reality is that vendors of connected devices and solutions have been **late in tackling security issues**, in part because of the complexity of framing the attack vectors in IoT systems, and in part because of cost considerations.

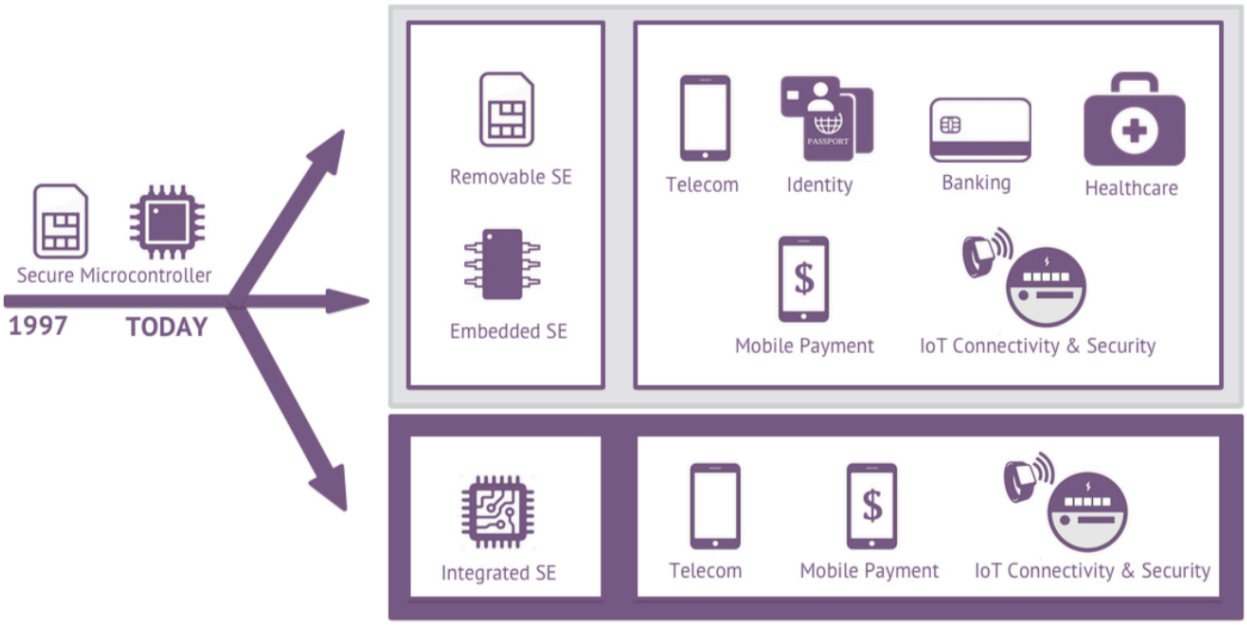
However, this situation is improving. Customers are increasingly quoting security as the leading barrier to IoT adoption and indicate that they are willing to pay a premium for device security. A recent study suggests that customers may be willing to pay an average 22% more to secure a device (source: Bain 2018). As a result, device and solution vendors look at ways to differentiate through security. They are increasingly partnering with security specialists, in order to offer their customers dedicated security solutions at the device level.

Overall, **security is not necessarily a losing battle**. In order to reduce the complexity of IoT security, and find the right cost / risk tradeoffs, it makes sense to delegate some security tasks at the very edge to a software environment that is easy to update and allows implementation of consistent security policies across a wide range of devices. This is a key driver for using Java Card technology in the Internet of Things.

SECURING THE IOT WITH JAVA CARD

SECURE ELEMENTS IN IoT

We call the “secure element” (SE) a tamper-resistant hardware environment capable of securely hosting applications and their confidential and cryptographic data. The most common form of secure element is a one-chip secure microcontroller, found in smart cards and other removable cryptographic tokens. New form factors have started to emerge, from embedded SEs (a non-removable secure microcontroller soldered onto a device board) to new security designs embedded into general purpose chips.
For any of those, a set of common critical requirements can be identified:



- Security: secure element applications must factor in security attributes, such as transaction atomicity, cryptography support, signing an authentication of applications, application isolation and firewall.
- Certifiability: customers require high-level security certifications according to Common Criteria and FIPS, as well as domain-specific certifications, for example from government entities or payment organisations.
- Compactness: secure elements are typically resource-constrained devices (CPU, memory, bandwidth). In particular, memory configurations rarely exceed 1MB of overall available memory, and RAM specifications

can be as low as 2KB. Security in general is more about reducing the scope and factorizing dedicated resources. This is the most viable strategy to reach high certification levels for instance.
- Standards-based manageability: secure element applications and stored credentials must be securely managed and updated, according to open industry standards.

A key benefit of using secure elements in IoT systems, is that they allow **isolation of the security functions in a device**: the secure element becomes the focus for sensitive operations and leverages hardened and tamperproof hardware and software to ensure those operations are conducted securely. This helps reduce the attack surface for security-related components to a minimal perimeter. The secure element can then offer security and cryptography services to the rest of the devices. Sensitive operations such as software updates can be hardened against attacks.

We will discuss in the rest of this document the use cases unlocked by secure elements running a programmable, secure application environment such as Java Card.

INTRODUCING JAVA CARD

Java Card is the most pervasive application platform in the world. We estimate that close to 6 billion Java Card-based secure elements and chips have been deployed in 2017 alone. The majority of those are smart cards (SIM cards, banking cards and ID cards). However, in recent years, we have witnessed a strong growth of Java Card embedded in mobile and IoT devices. New types of secure elements are being introduced in edge devices for Internet of Things (IoT) networks.

Java Card is at its core a very minimal subset of Java, enriched with features catering to the security needs of secure elements and tamperproof hardware designs. It addresses those requirements, while retaining the openness and code portability brought forward by Java. Several attributes of Java Card technology make it extremely well suited to provide services to internet of things devices, and address some of the risks exposed earlier:

Designed for Security and Highly Constrained Environments	• Dedicated runtime to isolate security related functions and credentials • Optimized footprint for very constrained devices (few KB, low power consumption) • Robust Security Model (language and VM, firewall, security countermeasures, certification schemes) • API (cryptography, secure storage and handling of credentials : keys, certificates, pin codes, biometric data, …)
Multi-application and multi-tenant	• Java Card can host several security applications from different providers and guarantee isolation • Adding new services is memory efficient thanks to the use of a Virtual Machine and platform API common to all applications, and the ability to use shared libraries to avoid code duplication

Open, Extensible and Manageable platform	• Java Card Platform and development tools available from multiple vendors • Platform can be extended with API and applications for specific vertical markets • Platform supports remote application management and allows for new applications being securely deployed, removed or updated after issuance. • Application Management offers various deployment models matching with major business models (single issuer, multiple stakeholders with or without direct business relationships, delegation model, …)
A solution to security hardware fragmentation	• Unified Security Model to address Hardware and Firmware fragmentation 1. Java Card provides a unified framework for applications on different security hardware and firmware 2. It also offers a migration path across different hardware solutions • Consistent management interface – different security form factors can be addressed through standardized interfaces and protocols that offer a consistent way to manage different devices during their entire lifecycle

These attributes make Java Card uniquely suited to the IoT markets and unlock a wealth of new use case. Some of those are detailed in the next few paragraphs.

IMPLEMENTING IoT EDGE SECURITY WITH JAVA CARD

As we have established, there are multiple facets to IoT Security, and an ever-expanding range of attack vectors for IoT systems. There is no one-size-fits-all solution. When it comes to edge security, it is essential to select components and technologies that can evolve according to changing threats. In this context, we will look at how Java Card's inherent secure design, flexibility and openness can help address some of the most critical security functions in an IoT system.

Preventing Tampering of Device Software and Credentials

Java Card has been designed from its inception for the management of sensitive assets. The introduction of IoT connected devices gives it a new chance to shine and provide integrity and trust services to the rest of the system. With Java Card, the secure element can provide a root of trust to the device, or serve as a basis for device attestation. Developers can add new security services in the form of Java Card applications to protect the device and its credentials, while retaining high-levels of certification.

Preventing Tampering of Device Software and Credentials

Sample Device Integrity Services

Protect the integrity and authenticity of the device software	• Secure Boot and Root of Trust • Measurements and Remote attestations to provide reliable evidence of software or configuration data to a remote entity • Use of signed code to verify the application code being loaded before its execution
Protect the storage of keys, root certificates and sensitive data (data at rest)	• Enforce the integrity and confidentiality of these sensitive data (keys, security policies, access control rules and permissions, configuration, …) • Detect attacks on the storage, including rollback or replay attacks
Secure application activation and upgrade	• Control issuers identities, verifies integrity and authenticity • Manage authorizations and activation rights

Securing Authentication and Communication to Cloud Services

The last couple years have seen the introduction of a large numbers of IoT platforms looking to connect devices, analyze data and dispatch processed information to business applications. Unlike the previous generation of M2M systems, most of those IoT platforms are cloud- based, and have indirect control or oversight over the devices themselves. IoT solution vendors generally rely on partners to connect objects to the cloud and to make sure that edge components are secure.

A secure element running Java Card can play a critical role to ensure trust between the cloud and connected device. It can be leveraged by the device to delegate the provisioning of device identity and to manage the initial on-boarding process. It can further secure the cloud authentication and authorisation process and store the related credentials securely.

Securing Authentication and Communication to Cloud Services

Sample IoT Cloud Security Services

Secure the provisioning of credentials	• Provide a means to secure the provisioning of credentials at manufacturing and/or during device activation • Manage credentials' life-cycle (initialization, expiration, renewal, blacklisting…) and secure update (protect data in motion)
Manage device identity, activation and authentication	• Protect the credential used during the authentication process (protect data at rest and data in use) to defend against device impersonation • Authenticate remote servers, for example with Certificate chain validation and management of root certificates (initial set and renewed list) • Generate and sign authorization requests to enforce origin
Cloud Security	• HSM to securely manage credentials (isolation, tamper resistance, compliance with regulatory standards, …)

Managing and Enforcing Endpoint Security Policies

Related to the advent of cloud platforms is a growing need for the management of device lifecycle states and local enforcement of security policies.

In a market where connected objects are likely to outlive several generations of IoT systems, control over the lifecycle of a device is key to ensure that it can be retired or repurposed securely, and that it continues to align with security policies as they evolve.

Since many objects may be intermittently connected to the network, it also helps to have a secure local focal point for the administration and enforcement of security policies.

Java Card can support this use case by combining the flexibility and updatability of Java with the local enforcement capabilities of a secure element.

Managing and Enforcing Endpoint Security Policies

Sample Endpoint Policy Services

Protect Device Lifecycle management	• Control and enforce life-cycle changes (consistency, authorization, history, anti-replay) • Control administration operations against device life-cycle state
Deploy, manage and upgrade application security policy	• Verify software version according to the security policy • Manage Access Control policy and permissions • Control revocation lists, white listing, black listing…
Monitor device and perform behavioral analysis	• Monitor device and applications (access, resource consumption, …) • Manage individual or contextual thresholds and generate alerts based on history, geo-fencing or any sensor's values (temperature, vibrations, acceleration

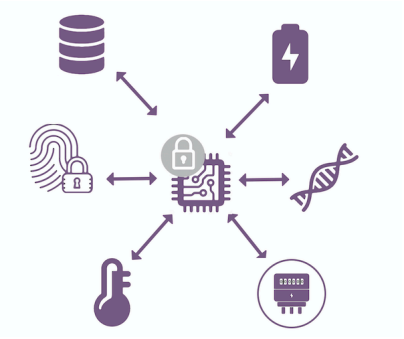
Controlling the Access and Securing the Use of Device Peripherals

Another exciting use for Java Card in IoT, is to secure the "last yard" between devices (or gateways) and attached peripherals. This is especially important for systems comprising actuator devices that can have a direct impact on safety. Securing data coming from a sensor (or sent to an actuator), can be a complex cost/technology equation in untrusted physical environments. It is a challenging problem to solve, as sensors are often very simple devices with severe price constraints. Having them perform cryptographic operations or coupling them with a dedicated security chip brings about some commercial hurdles, in particular in markets where sensor data can be highly sensitive, such as automotive.

The introduction of multi-application secure elements, powered by Java Card offers cost-effective options, as the price of the hardware may be shared by multiple services.

Firstly, the secure element can be granted direct access to peripherals. This can be used to make sure that there is no "in the clear" communication between a sensor and the cloud. One can also leverage the secure element to perform verification of biometric data against an expected value, avoiding side channel attacks. The secure element can also verify the operating conditions using built-in sensors and detect abnormal use. There are also scenarios where a secure chip could perform general purpose operations such as data averaging or initial filtering and even edge analytics, to alleviate the load or act as a substitute to a general purpose MCU.

Java Card is at the core of these use cases. The Java Card Forum and Oracle have standardised APIs and a new I/O model, to allow direct and secure access to peripherals from within Java Card applications. Those features are included in the Java Card 3.1 release, helping to simplify the implementation of Trusted Peripheral use cases and enabling trust and the exchange of sensitive data at the very edge.



Controlling the Access and Securing the Use of Device Peripherals

Sample Trusted Peripheral Use Cases

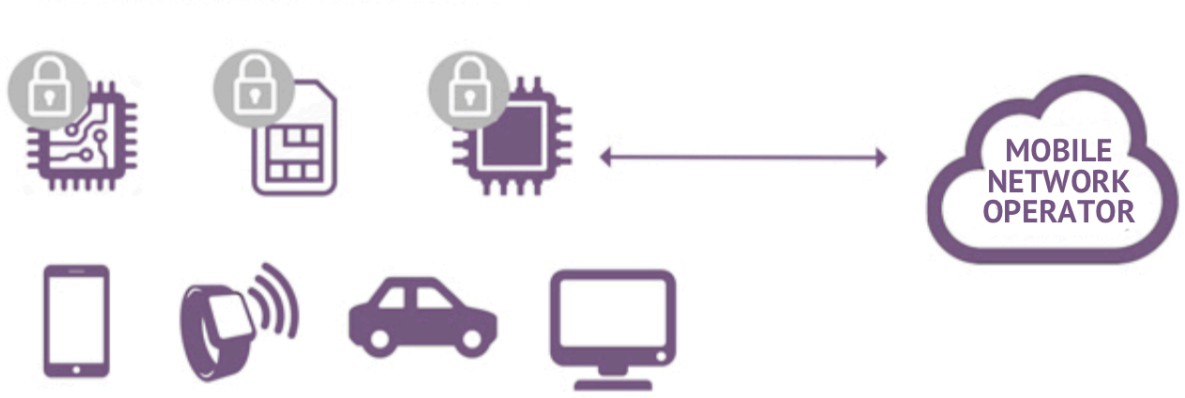
Secure Channel between peripherals and security chip	• Perform biometric authentication: access to fingerprint reader, data acquisition and verification with the secure element • Enable Encrypted data storage: access to local store • Perform Payment and transport transactions: access to NFC reader • Securely display sensitive data and capture input: trusted UI • Use sensors to detect abnormal use of a device with regards to a security policy or lifecycle state (motion sensor, temperature sensor, localization sensor, …)
Out-of-band communication between security chip and cloud	• Allow direct communication for sensitive data without transiting through the main processor (e.g. biometric information, root of trust credentials) • Remotely perform security operations (device location, data wipe, lock/unlock, audit, …)
Authenticated data sources at the edge	• Allow authentication of data coming from a sensor or commands to an actuator
Data processing by edge secure chip	• Enable the secure chip to perform pre-processing and edge analytics on local data • Alleviate workload from, or substitute to the main processor

IoT USE-CASE EXAMPLE SOLUTIONS WITH JAVA CARD

Having established some of the benefits of using Java Card in an IoT environment, it may be useful to go over some real-life illustrations of Java Card usage. The technology is already being used at scale to connect and secure devices. This section lists some the key examples of Java Card in IoT deployments.

SECURE NETWORK ACCESS AND COMMUNICATION

Flexible connectivity has become a critical success factor in a world full of connected devices. This is even more critical as more and more devices are joining the IoT, and massive-IoT is an expected outcome of 5G. IoT device manufacturers deploy embedded SIM (eSIM) modules for consumer and M2M use cases to provide a reliable, robust and trusted access to the IoT ecosystem.



Embedded during the device production, blank SIMs are deployed in a variety of different formats and can be updated with the eSIM profiles of local MNOs (Mobile Network Operators) over the air, in the field. This flexibility reduces the need for IoT device makers and their suppliers to stock many different versions of SIMs for use in multiple countries, eliminating a lot of headaches from the manufacturer's logistics chain.

Furthermore, the GSMA specifications[1] for remote subscription provisioning allows IoT service providers to select and download a subscription inside an embedded SIM for their devices, once they are actually deployed in the field. The remote subscription provisioning system also allows the switch from one subscription to another, which could for instance be triggered based on the device required quality of service or locally available access networks.

The latter aspect is part of a larger scope of power saving strategies required for IoT devices that need to be functioning autonomously for ten or twenty years, in low reachability locations. 3GPP has in this area enhanced the USIM and 3GPP devices specifications with features that allow IoT devices to deactivate or suspend the USIM for a long period of time, with the USIM being able to keep its internal status and thus optimize its wake-up time. These power saving mode features allow IoT devices to reduce battery consumption to a minimum level.

Java Card has traditionally been used in GSM, 3G, 4G and soon 5G networks to secure access to the cellular network. It is referenced in the 3GPP USIM and ISIM specifications[2], and follows the 3GPP authentication and key agreement protocols[3]. As a result, Java Card is being used in billions of SIM cards deployed each year worldwide. MNOs develop and deploy Java Card applications to host and manage customer subscription, or implement operator network and power optimisation strategies. The dedicated security features of Java Card provide the perfect environment to securely store credentials, govern the authentication to the communication network and manage MNO customer-specific profiles and applications.

SECURE GATEWAY AUTHENTICATION AND COMMUNICATION

The deployment of Gateways becomes more and more common in IoT and plays a central role, given the limited power resources and connectivity capabilities of a lot of endpoints, but also in response to real-time constraints, implying the need for aggregating control locally over a number of endpoints (unified interface to the user) and implementing automation locally (sensing & control) without constantly communicating with the Cloud.

Gateways can be found in Smart homes (for example to control lighting, to control appliances remotely, door locks, garage openers, collect data from motion or presence sensors, etc.). Other application areas include Industrial automation systems or street lights in smart cities.

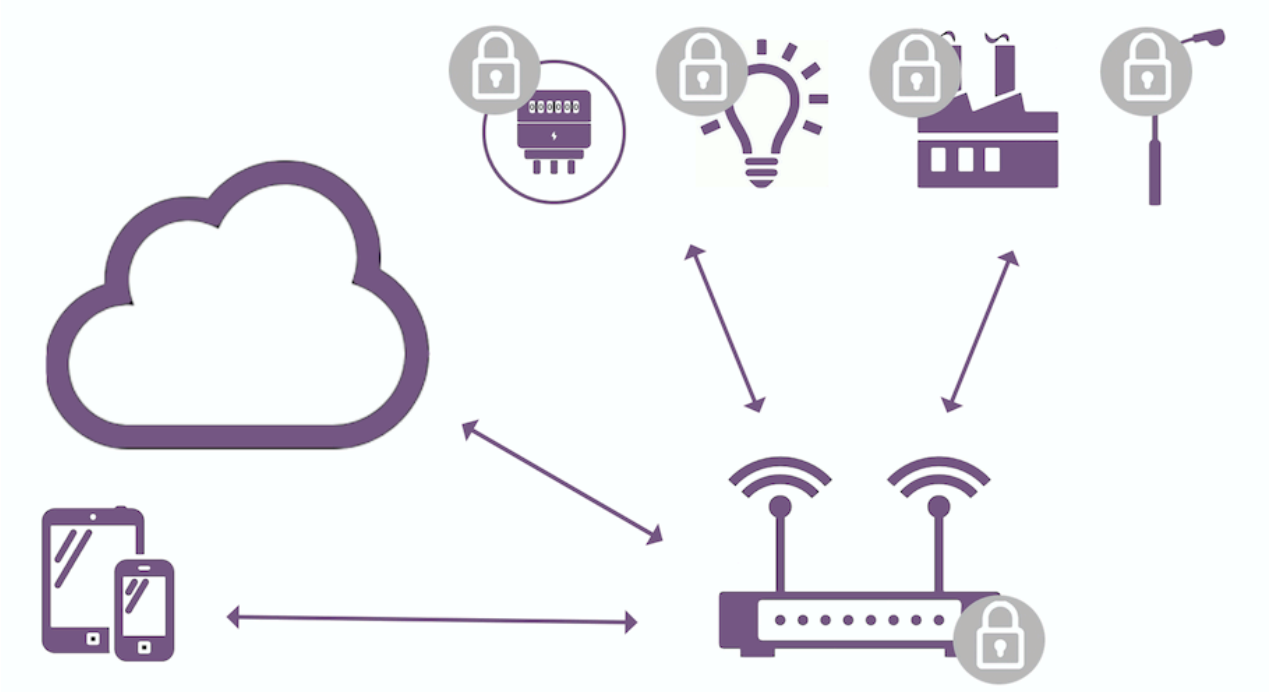
1 http://www.gsma.com/esim/

2 3GPP TS 31.102 for USIM, and 3GPP TS 31.103 for ISIM

3 3GPP TS 33.102, TS 33.401 and TS 33.501 specifications for 3G, 4G and 5G security architecture respectively

These gateways have three types of interfaces:

- Interface to the Cloud on one hand (typically through WiFi or cellular)
- Interface to local devices (typically using communications standards like ZigBee, Thread, BLE)
- Interface to users' phones and tablets (typically through WiFi or BLE)



The interface to the Cloud is crucial, as it is the interface towards the external world. Therefore, authentication of the remote server it connects to is critical, to make sure a rogue server does not access the gateway with its Private data and functionality, and to insure a door is not opened for an attacker to load a malware on the gateway. But also, authentication of the gateway itself to the Cloud and the service(s) running in the Cloud is essential to protect the Cloud infrastructure, protect access to the services and the associated monetisation, and protect the integrity of data uploaded to the Cloud. As a result, mutual authentication gateway-Cloud is key to secure the IoT applications.

The interface to local devices must be encrypted and authenticated too, to avoid an injection of fake messages, the inclusion of unauthorised devices into the private local network set up by the gateway which could disrupt local automation mechanisms, or the leakage of data that could be captured by sensing tools tens of meters away from the gateway. A challenge is the secure sharing of the network key(s) with new devices to be on-boarded onto the network.

Finally, as user devices like phones are used to pair devices with the gateway or to control it, the gateway will need to implement access control mechanisms to these devices and the apps running on it.

Java Card-based secure elements are a prime choice for implementing gateway security; Java Card being the leading application platform for secure elements, many products are available today for integration into gateway hardware. Java Card based applications can be loaded on the secure element to ensure mutual device / cloud authentication and provide security services to attached devices. Since a Java Card based secure element can be updated with new applications, a service provider can ensure that authentication schemes supported in the device hardware, continue to follow policies set by cloud platforms. This is a unique advantage compared to static hardware security options, which run the risk of being quickly obsoleted, or need to rely on code loaded in unsecured device software to adapt to new security policies.

Additional use cases for Java Card in home and gateway environments can easily be found. oneM2M TR-0038 “Developer Guide: Implementing Security” describes a complete application scenario in a smart home environments, where some components could be advantageously developed using Java Card technology:

- A Door Lock applications implemented on top of Secure Components supporting a Java Card Virtual Machine, to ease portability across hardware from different suppliers
- A Mobile Phone application relying on a Java Card applet running inside the SIM card.
- New examples of Java Card usage are being rolled up every day, both in home and industrial applications.

SMART METERING & SMART GRID

Smart metering is certainly one of the largest deployment of IoT devices, with ongoing roll-out in several regions including the US, Japan and China, or about to start, as in several European countries or India.

Each deployment has its own characteristics in terms of system architecture, but usually three types or class of architectures can be identified:

- Open, de-centralised architecture in which meters of different types are connected to a residential gateway, itself accessed by different service providers (energy retailers but also potentially others)
- Closed, centralised architecture where meters communicate with a residential hub connected to a central back-end system managed by a single entity. The back-end routes the data to/from different energy retailers
- Closed, centralised architecture where meters connect to data concentrators (typically located on the streets), collecting and aggregating data from typically hundreds of meters (usually thru power-line communications, cellular or Sub-GHz radio technology) and communicating consolidated data to the back-end

In terms of security, the first architecture will require, in particular, the implementation of strong access control mechanisms to insure Privacy of data, as well as selective access to different data sets. This will require strong authentication to support differentiated role-based access control (multi-tenant platform), as well as platform ownership management and administration. The (SW) integrity of the gateways will be essential to prevent any data leakage or loss of control.

In the second architecture, the centralised collection of data in the Cloud from a large collection of endpoints makes the protection of such a data centre very important, leading to the need for end-to-end secure connections (meter to back-end), as well as strong authentication to the back-end system with secure distribution and management of hub and meter credentials.

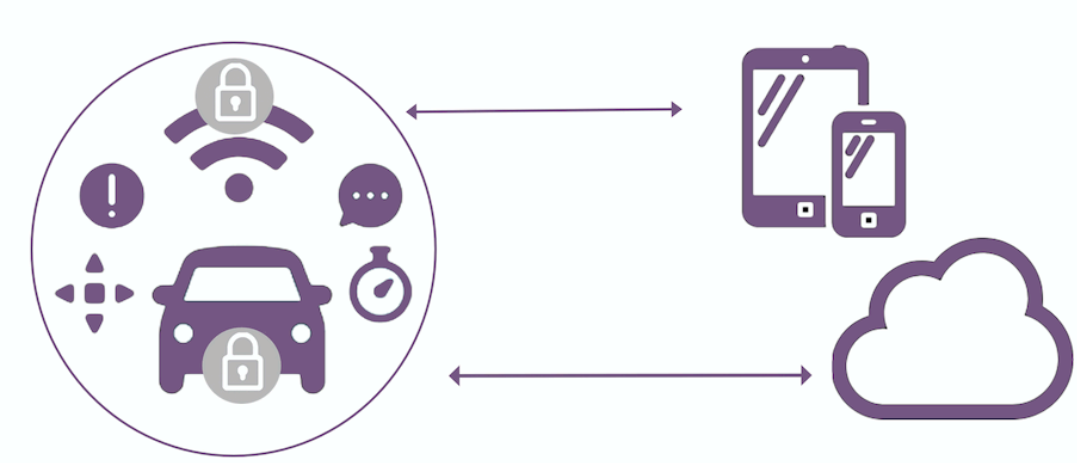
In the third case, the use of data concentrators in potentially exposed locations and controlling a significant number of endpoints, will require strong protection of credentials, as well as countermeasures to verify, preserve and maintain the integrity of software running on the concentrators.

In all cases, it is essential to set-up encrypted and authenticated communications with the meters (endpoints) with integrity protection of SW running on the smart meters, especially when critical remote control functions, like a switch on electrical meters or valve in gas meters, are implemented. Java Card products are widely used in those scenarios to provide authentication and integrity services. They can be adapted to multiple system architectures by virtue of the programmability. Running on secure and certified hardware, Java Card applications ensure that physically exposed metering devices can protect their security credentials.

VEHICLE COMMUNICATION SECURITY

In a world of connected or even autonomously driving cars, privacy, integrity and confidentiality of the data exchanged must be paramount:

- Data communication within the vehicle network
- Data communication from the vehicle to the cloud or roadside equipment (e.g. geo-location, vehicle specific data, traffic conditions,…)
- Data communication from the cloud or roadside equipment to the vehicle (e.g. remote diagnostics, SW updates, …)
- Data communication from the user (owner, driver, fleet manager,…) to the vehicle via the mobile device



Java Card based solutions can help to protect and manage the identities of vehicles, their components and their owners / users / drivers from the time of production over the entire lifecycle of the vehicle /device. Java Card can maintain the authenticity and integrity of the exchanged data, as well as the protection against cyber attacks from the Internet or from the infrastructure (e.g. traffic lights, traffic control systems,...). Thanks to the latest Java Card IoT functionality, it can securely handle sensitive data coming from vehicle peripherals and sensors to protect it against tampering and spoofing.

The biggest cyber security challenge for the Automotive industry is that with the connected car; all of a sudden a device gets connected that was not foreseen to be part of the Internet. Yet, the benefits for customers, as well as for producers, are expected to justify the efforts AND the risk:

- Consumers demand seamless mobile device integration for ubiquitous reachability
- Regulators mandate eCall and collision avoidance capabilities to save lives and make driving more secure
- Manufacturers aim to monetize on new opportunities (e.g. predictive maintenance, product improvements,..)

Four main challenges stem from this hunger for data communication:

- Gather the relevant data from the independent sensor endpoints
- Process the generated data in "real-time" - locally or in the cloud
- Ensure the integrity and authenticity of the generated / collected data
- Protect and manage the vehicle /sensor identities over the entire lifecycle of the vehicle (i.e. starting with production)

Java Card can be used to secure the vehicle communication in various ways:

- As the security vault for the credentials
- As the control gateway managing and controlling all connected endpoints
- As the security gateway protecting the in-vehicle communication streams
- As the security gateway protecting the communication streams to the cloud

Being already the preferred platform to deploy flexible connectivity Java Cards products are now considered by automotive OEMs to host secure connected car use cases as well.

CONCLUSION

In this white paper, we have covered some of the security challenges introduced by devices in IoT systems. We have discussed how Java Card can address security risk at the edge of the network, and detailed real-life examples illustrating how the technology is used today.

New solutions leveraging Java Card technology continue to emerge. To support the evolution of IoT security needs, the Java Card Forum and Oracle are bringing the Java Card 3.1 platform to market, with a wide range of functionality aimed at facilitating IoT security use cases. New features include an extensible I/O model for the support for IoT protocols and communication with trusted peripherals, cryptographic extensions and security services to ease the implementation of IoT Cloud Services authentication schemes, and a variety of usability enhancements to make the utilization of secure elements by IoT OEMs and developers more efficient. With the introduction of Java Card 3.1, we expect to see increased usage of Java Card and secure elements in the IoT space.

In the meantime there is a variety of additional resources available to learn more about Java Card :

- Java Card Forum : https://javacardforum.com/
- Oracle : http://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html