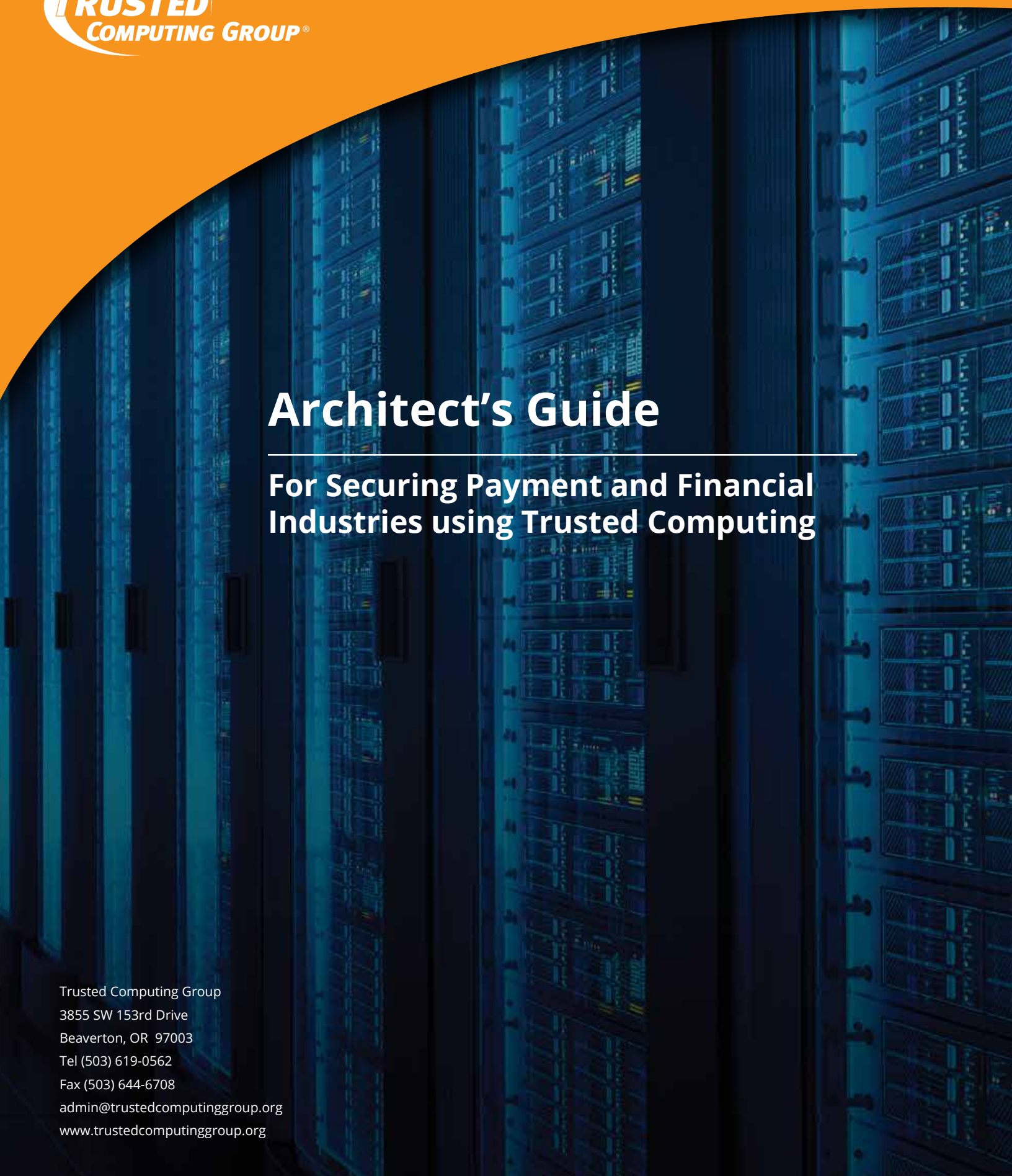
JUNE 2018

TRUSTED COMPUTING GROUP®



Architect's Guide

For Securing Payment and Financial Industries using Trusted Computing

Trusted Computing Group
3855 SW 153rd Drive
Beaverton, OR 97003
Tel (503) 619-0562
Fax (503) 644-6708
admin@trustedcomputinggroup.org
www.trustedcomputinggroup.org

EXECUTIVE SUMMARY AND ACTION ITEMS

This architects guide focuses on reducing fraud in the payment industry. IoT devices are slowly emerging as new payment instruments. Smart watches, in particular, are currently used for NFC tap and pay payments. The majority of these devices rely on a paired mobile device such as a smart phone for some functionality, which may include provisioning and authentication.

In the foreseeable future, it's expected that these IoT payment instruments will no longer need a companion-assisting device. Backend fraud and risk engines currently rely on payment parameters to reach an authorize or reject decision on incoming transaction requests. To reduce fraud risk, the backend needs to collect some signals, authenticate the user, and identify the device as well as the POS (point of sale terminal, which can also be another IoT instrument.)

Trusted computing mechanisms make the act of payment easier for users, by allowing devices to provide a range of levels of protection depending on the type of payment. Backends can use trusted computing mechanisms to determine the level of protection during the current payment. Backends can demand the highest level of protection for riskier payments, even though this involves more complex and onerous user activity. Backends can accept lower levels of protection for less risky payments, enabling simpler user interaction, or even no user interaction. Also, some suggestions are proposed to expand the use cases and relief the customer of some friction.

INTRODUCTION

Some IoT devices, such as smart watches, are currently used in the payment industry for NFC tap and pay. Some devices rely on a secure element and others rely on TrustZone™ / SGX or a separation kernel. To enable mobile payment or IoT device payments, a PAN (Primary Account Number) needs to be provisioned on the device. This PAN is generally digitized or tokenized resulting in a DPAN.

Additionally, NFC transactions require the use of cryptograms that are verified and authorized at the back end to grant a payment transaction, reject it, or go through an additional step-up authentication. The back end needs to identify the device and authenticate the user.

There are currently some weakness and challenges in payment instruments not using hardware security, such as a secure element. Additionally, payment instruments that use technologies such as secure world/normal world separation have some areas that can be hardened. This separation provides adequate data at rest protection. Use of a secure element provides adequate data in use protection, as well.

But some implementations of payment frameworks built on secure world/normal world separation platforms do no provide sufficient data-in-use protection. For example, when an NFC payment transaction is conducted at a POS, the IoT device will extract the payment keys from the secure storage area to calculate a cryptogram. When this happens in normal world, payment credentials and other sensitive information could be exposed to malicious malware components.

SOLUTIONS OVERVIEW

The payment and financial industries depend on several parameters to ensure transactions are authentic. A typical fraud and risk engine at a financial institution backend gathers credentials and signals from the originating instrument to authorize or reject a given transaction. Authorization decisions have time constraints as well as latency and throughput requirements.

For example, an NFC transaction must be authorized or declined within 500 milliseconds, and the backend needs to be able to handle tens of thousands of simultaneous transactions per second. So, any realistic security solution must adhere to timing requirements of the given type of transaction. EMVco, GP, for example, have further information on the topic.

Payment networks may have additional constraints. The following sections list some of the more salient parameters needed for an IoT payment instrument. Since it's impossible to exhaust all weaknesses and hardening mitigating controls for each—as in a full threat modeling approach—*Appendix A* lists some of the more relevant security principles that the architect needs to keep in mind.

IN SCOPE

This guide covers direct security controls that affect the payment and financial industry from a payment network perspective. Other security requirements that may have an indirect impact on the financial institution, such as susceptibility of the device to be owned by a botnet, are out of scope.

DEVICE IDENTIFICATION AND AUTHENTICATION

Device identification by a challenger, a backend server, or another legitimate entity is often necessary to either enable some use cases or to reduce fraud and illegitimate access to services provided by the service provider. Device authentication is a control separate to device identification. The following two subsections give an overview of some authentication guidelines. The next section introduces multi-entity authentication, which might be used alongside multi-factor authentication.

MULTI-ENTITY AUTHENTICATION

Two entities may require authentication: the user and the IoT device. If the service requires both entities to be authenticated before it grants a service, then this is obviously two-entity authentication – not two-factor authentication. When this is the case, it's usually more flexible to leave the task of mapping the device to the user at the control of the service provider. Artificially coupling device parameters with user parameters may be useful in some situations, but this is a rather restrictive architectural design.

MULTI-FACTOR AUTHENTICATION

In multi-factor authentication, classic factors are: something you know (username/password, PIN,) something you have (hardware token, smart card,) and something you are (biometrics such as fingerprint or retina scan or facial recognition).

Other parameters such as location information, time restrictions, etc. are authorization control parameters, not authentication control parameters. One important factor to consider is parameter-entity-affinity. Location information, for example has device affinity—not user affinity. It's important to be cognizant of this fact because if an architect depends on location information to mean user location, then their architecture is susceptible to some attack and fraud methods. The user of the device can be in control of an IoT device thousands of miles away, and the location of the device has nothing to do whatsoever with the location of the user. Therefore, control parameter-entity-affinity needs to be carefully considered.

SECURE CRYPTOGRAPHIC OPERATIONS

EXTENDED USAGE OF PLATFORM KEY HIERARCHY

TPM 2.0 has controls and a dedicated (platform) key hierarchy that enables platform hardware/firmware/software to use TPM facilities irrespective of how the user uses that TPM.

SIGNALS FIDELITY

In the mobile IoT payment industry it's sometimes essential to identify the source of the authorization request or the destination of a device where a credit card is provisioned for future mobile payments. Currently, the signals from the device are not deterministic enough to identify the device, let alone authenticate it.

There are exceptions to this and depend on close relationships between the network and the OEM or operating system provider. This is particularly true for HCE (host-based card emulation) type devices. Devices that use an embedded secure element do not suffer from this limitation as the network obtains an SEID (secure element ID.) Additionally, some OEMs and operating system providers or wallet platform providers are not willing to share identifying device information due to privacy policy restrictions.

TRUSTED PAIRING

In some situations where an IoT device provides functionality by pairing with a mobile device through a Bluetooth Low Energy protocol, to enable mobile payments from the IoT device, the pairing protocol and transport may need further hardening.

INTEGRITY METRICS – FOR DEVICE ROOTING AND JAIL-BREAKING

Device tamper resistance and detection is an important signal that aids fraud and risk engines at the service provider's end or the mobile wallet provider to aid in scoring the device risk based on the device tamper state. The network or backend needs to be able to obtain device tamper information as a separate signal or as part of an integrity metrics report.

GENUINE DEVICES

Another important signal the backend needs is information on authenticity of the device, in the sense that: is it a genuine device or is it a simulator running on a much more powerful and open computer. Integrity metrics may be utilized for this.

MEASURE BEFORE YOU HAND CONTROL

TCG enables measured boot and integrity metrics that can be used to enhance the security posture of the IoT device. An IoT device passes through several stages during the bootstrap process. Per TCG specifications, such as PC Client, each stage must measure and extend the next stage to the specified PCR before handing control to it. This functionality enables the backend to verify the integrity of an IoT device.

There is a subtle difference between secure boot and measured boot: Secure boot means digital signature verification of various bootstrap stages. If one stage fails signature verification, the boot process is halted and the system may typically revert to recovery mode.

In measured boot, there is no signature verification required. The previous stage measures and extends the next stage or range into one or more of the specified PCRs. The challenger checks the PCRs among other things to verify the reported state of the device (attestation.) Values in PCRs can also be used to seal data. Sealing data to PCRs means that this data cannot be decrypted or unsealed until the platform is in a certain state—as shown by the values in the PCRs.

FRAUD REDUCTION CONTROLS

Fraud protection includes protecting against device cloning attacks. Some insiders (as in OEMs) have access to tools that enable them to clone devices. This is mainly used for debugging and to aid in root cause analysis. If a fraudster—an insider or an outsider—gains access to these tools then they'll have the ability to clone a victim's device and conduct financial transaction on their behalf. Architects should consider incorporating PUF, or physically unclonable functionality, in their designs.

SECURE OTA CRYPTOGRAM ALGORITHM UPDATES

Architects also should utilize the interface between fTPM and TEE and utilization for the payment framework to perform sensitive calculations within the confines of TEE. When crypto-sensitive operations take place within a hostile execution environment such as a rich OS execution environment that runs in the normal world (where the rich operating system runs,) sensitive information and credentials as well as cryptographic keys are exposed, even if for a short period of time.

Currently there are some workaround mechanisms like WBC (white box cryptography) which are not considered to be a long-term solution. Cryptogram calculation algorithms change frequently, and the ability to securely update them over the air is a needed capability. The idea is to contain these algorithms within a secure execution environment such as TEE.

APPENDIX A – GENERIC HIGH-LEVEL ARCHITECTURE:

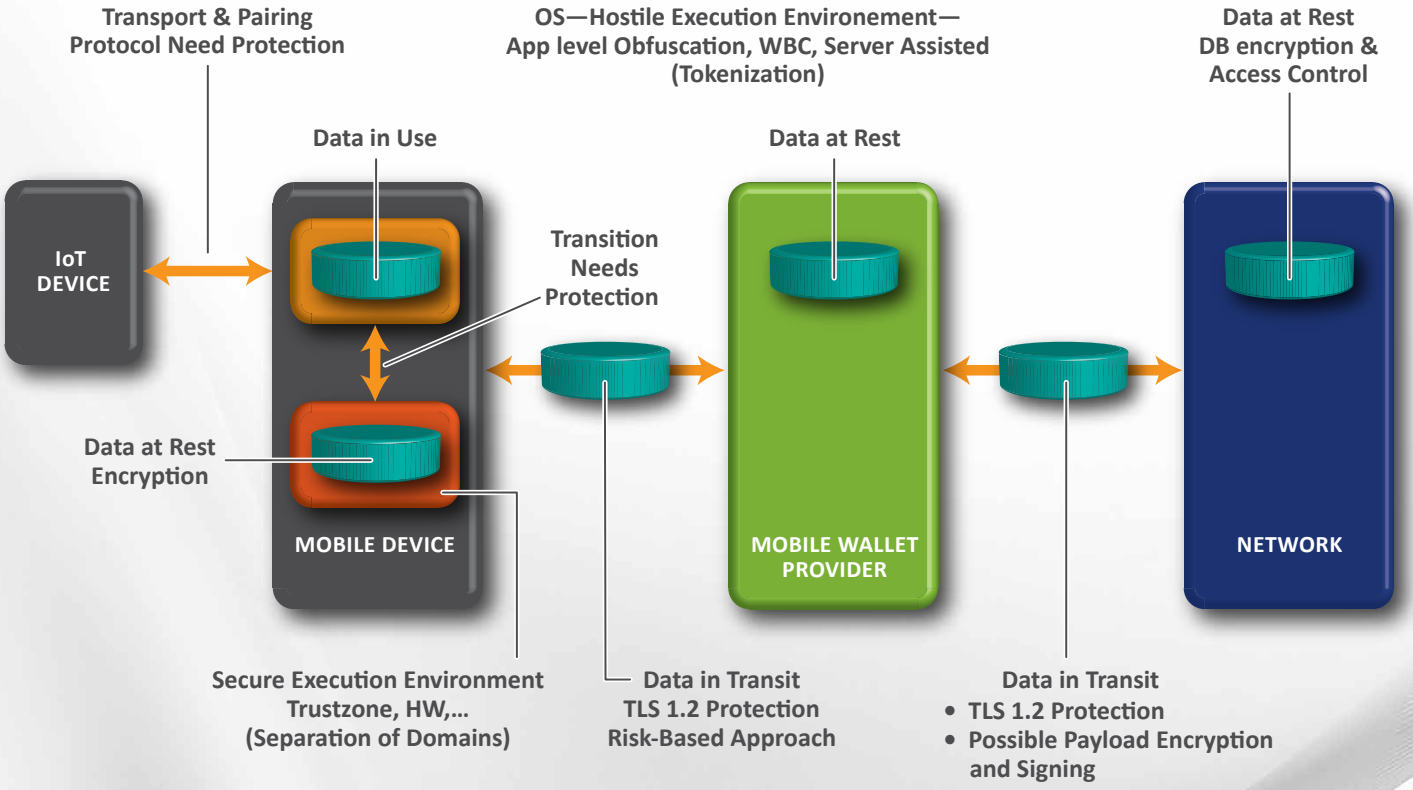


Figure 1. High-level deployment diagram of a generic IoT financial setup.

Figure 1 depicts an example of a generic architecture of an IoT device communicating through a mobile phone and Figure 2 depicts a standalone IoT configuration. The diagram shows assets (data and credentials) at various states and some needed controls of protection. In the diagram, going from left to right, the participating entities are in the following table:

ENTITY	PURPOSE
Network	Authorize transaction
Mobile Wallet Provider	
IoT Device	Present payment credentials Calculate cryptogram
POS—Point of Sale Terminal	Communicate with IoT device to conduct transaction
IoT Payment Gateway	Entry point to Network

All entities need to protect data in its three states: Data at rest, Data in Transit, and Data in Use.

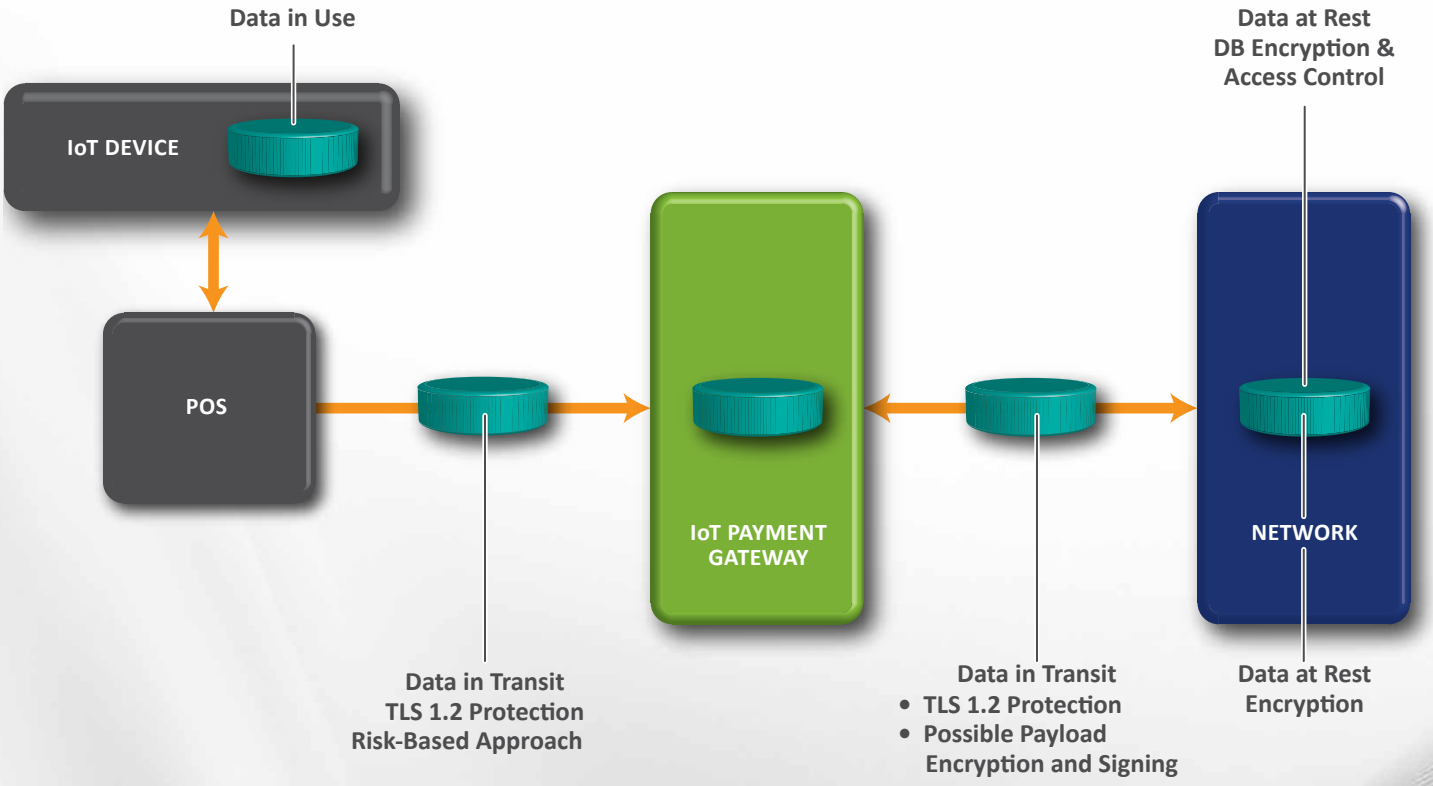


Figure 2. Generic IoT setup.