mojaloop

Mojaloop Quality & Security Workstream

PI-10 Feedback Session – 22 July 2020

Presenters

Godfrey Kutumela, Program Manager – Crosslake Technologies
Lewis Daly, Lead Developer - Crosslake Technologies
Victor Akidiva, Security Architect - Modusbox
Max Gysi, Lead Consultant – Gysi Solutions

Mojaloop Quality & Security Workstream Overview

Objective:

- ❖ *Continuously improve the Trust (reliability, transparency, privacy, quality and security)* of the Mojaloop System.

Delivery Model:

- ❖ Supports both *functional and non-functional* requirements of the project, working alongside with other *workstreams & various governance committees* on a *shared responsibility Model.*

Approach:

- ❖ <u>Standard and Control Centric </u>– Define and maintain Mojaloop software quality and security standards/guidelines/controls.

- ❖ <u>Risk and Threat Centric </u>– Perform risk and threat modelling to identify, validate, classify & prioritize security requirements.

Key Milestones:

- ❖ PI 1 – 8 : Foundation Phase - Built-in confidentiality and Integrity as part of the Core Mojaloop Architecture.
  - ✓ Implemented Signatures, MTLS, PKI, encryption standards
  - ✓ Established a software quality and security framework - DevOps & CI/CD Tools automation, workflows & policies

- ❖ PI 9 – Current: Improvement Phase – Consolidate, optimize & improve.
  - ✓ Introduced a risk and threat driven approach
  - ✓ Baseline best practice standards
  - ✓ Focus on the data

PI 10 Objective

The key objective of the PI is to improve data protection measures:

- for handling of Personally Identifiable Information (PII) and payment sensitive data
- and baseline the Mojaloop platform against best practice standards

Objective Breakdown:

❖ **Security Standard Baselining**

Benchmark Mojaloop against best practice control frameworks mainly Payment Card Industry Data Security Standard (PCI DSS) Standard.

❖ **Data Privacy and Security**

Identify and protect PII and payment sensitive data - referencing General Data Protection Regulations (GDPR) and PCI DSS respectively

❖ **DevOps Security – Multi PI Operational Support**

DevOps security tool maintenance and ongoing vulnerability management operational support

❖ **Cryptographic Processing Function**

To begin to bring full transaction and data security into the Mojaloop Ecosystem,

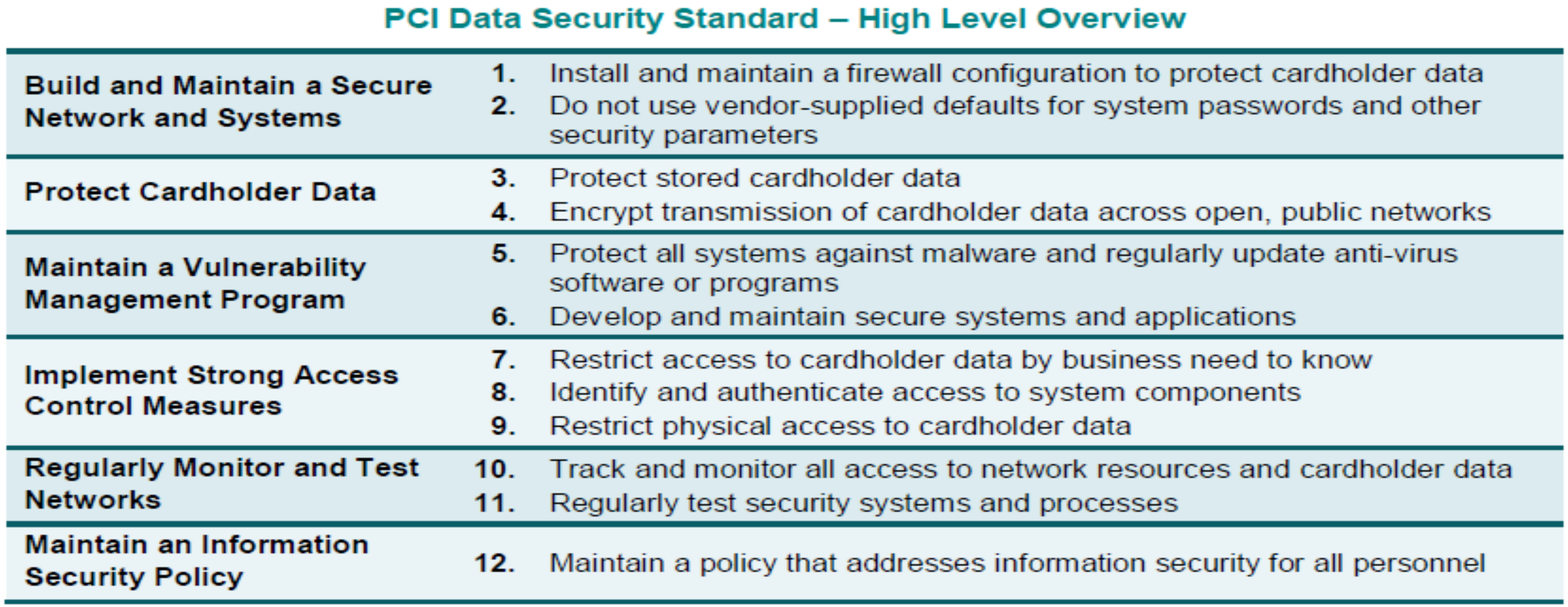
— Taking an inside-out approach, focusing on critical and sensitive data

Baseline against Payment Card Industry Data Security Standard (PCI DSS v3) Standard - Report Out

By Godfrey Kutumela

Overview of the PCI DSS

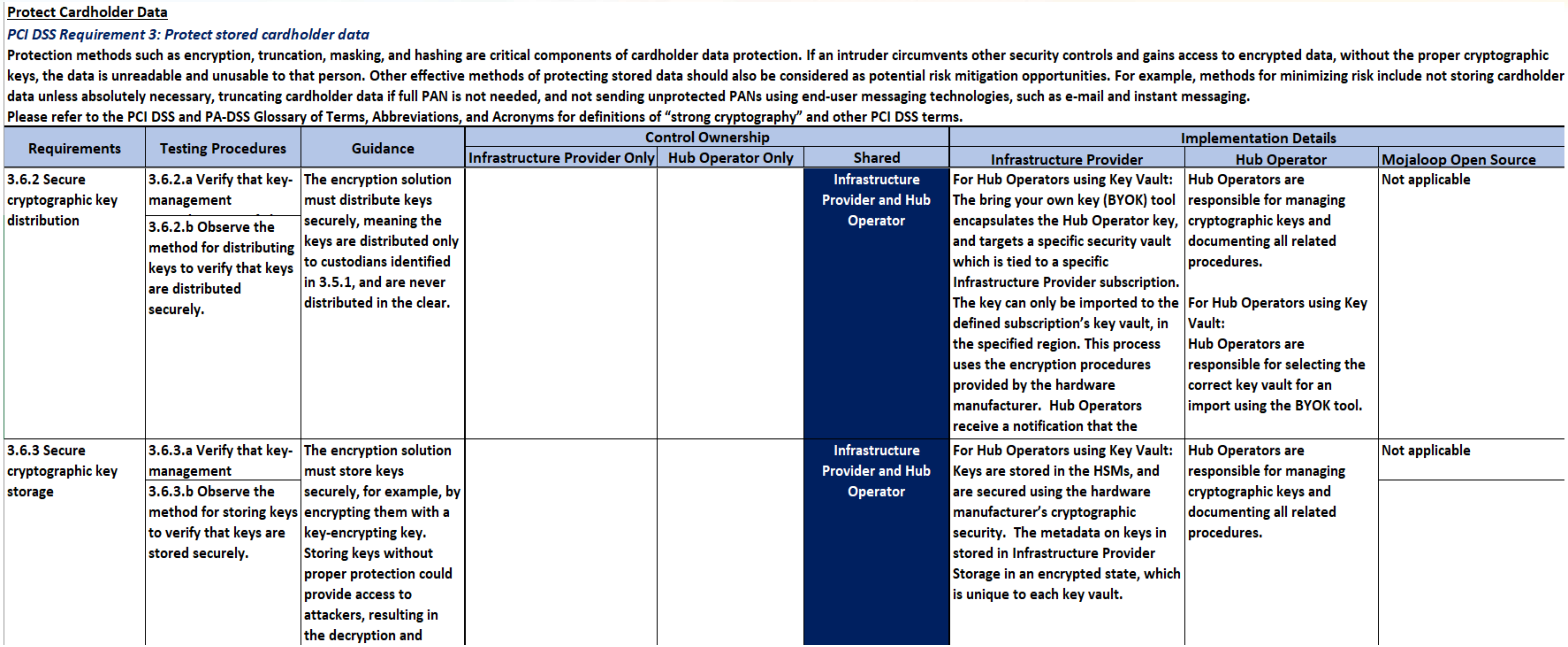
The PCI DSS is the global data security standard that any business of any size must adhere to in order to accept payment cards. It presents common sense steps that mirror best security practices. All sections are applicable to Mojaloop with exception of section 2 – Protech Cardholder Data.



PCI Data Security Standard – High Level Overview

Build and Maintain a Secure Network and Systems	**1.**	Install and maintain a firewall configuration to protect cardholder data
	2.	Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	**3.**	Protect stored cardholder data
	4.	Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	**5.**	Protect all systems against malware and regularly update anti-virus software or programs
	6.	Develop and maintain secure systems and applications
Implement Strong Access Control Measures	**7.**	Restrict access to cardholder data by business need to know
	8.	Identify and authenticate access to system components
	9.	Restrict physical access to cardholder data
Regularly Monitor and Test Networks	**10.**	Track and monitor all access to network resources and cardholder data
	11.	Regularly test security systems and processes
Maintain an Information Security Policy	**12.**	Maintain a policy that addresses information security for all personnel

Standard Baseline Method



Protect Cardholder Data

PCI DSS Requirement 3: Protect stored cardholder data

Protection methods such as encryption, truncation, masking, and hashing are critical components of cardholder data protection. If an intruder circumvents other security controls and gains access to encrypted data, without the proper cryptographic keys, the data is unreadable and unusable to that person. Other effective methods of protecting stored data should also be considered as potential risk mitigation opportunities. For example, methods for minimizing risk include not storing cardholder data unless absolutely necessary, truncating cardholder data if full PAN is not needed, and not sending unprotected PANs using end-user messaging technologies, such as e-mail and instant messaging.

Please refer to the PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms for definitions of “strong cryptography” and other PCI DSS terms.

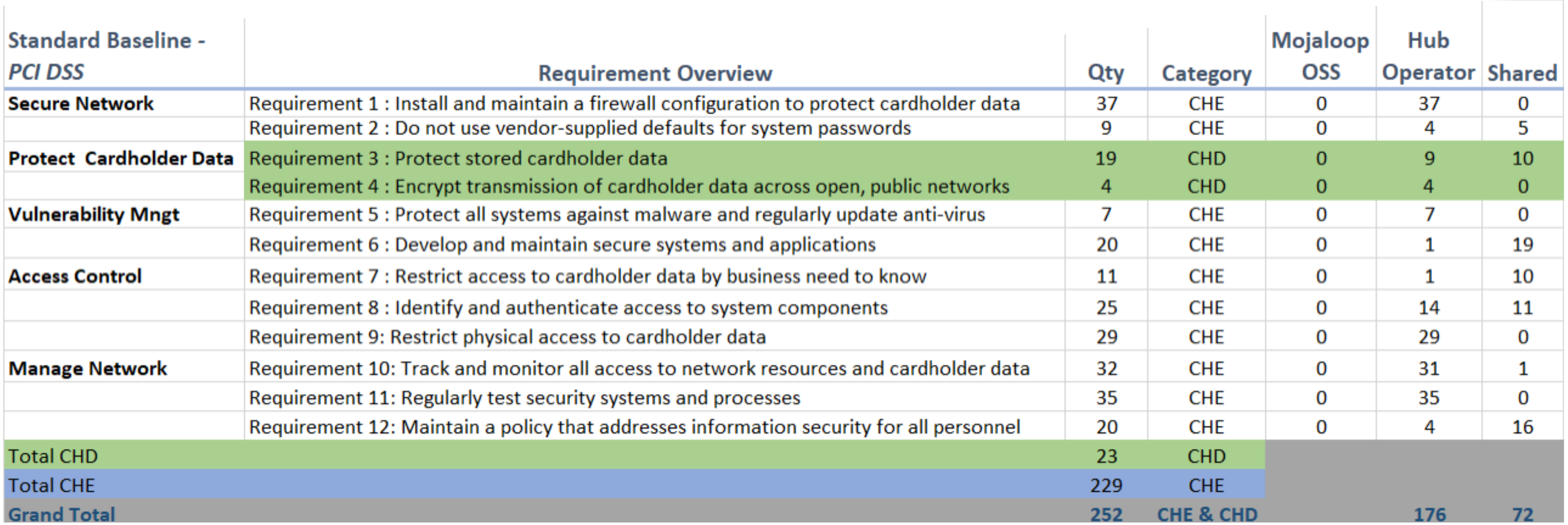
Requirements	Testing Procedures	Guidance	Control Ownership			Implementation Details		
			Infrastructure Provider Only	Hub Operator Only	Shared	Infrastructure Provider	Hub Operator	Mojaloop Open Source
3.6.2 Secure cryptographic key distribution	3.6.2.a Verify that key-management 3.6.2.b Observe the method for distributing keys to verify that keys are distributed securely.	The encryption solution must distribute keys securely, meaning the keys are distributed only to custodians identified in 3.5.1, and are never distributed in the clear.			Infrastructure Provider and Hub Operator	For Hub Operators using Key Vault: The bring your own key (BYOK) tool encapsulates the Hub Operator key, and targets a specific security vault which is tied to a specific Infrastructure Provider subscription. The key can only be imported to the defined subscription’s key vault, in the specified region. This process uses the encryption procedures provided by the hardware manufacturer. Hub Operators receive a notification that the	Hub Operators are responsible for managing cryptographic keys and documenting all related procedures. For Hub Operators using Key Vault: Hub Operators are responsible for selecting the correct key vault for an import using the BYOK tool.	Not applicable
3.6.3 Secure cryptographic key storage	3.6.3.a Verify that key-management 3.6.3.b Observe the method for storing keys to verify that keys are stored securely.	The encryption solution must store keys securely, for example, by encrypting them with a key-encrypting key. Storing keys without proper protection could provide access to attackers, resulting in the decryption and			Infrastructure Provider and Hub Operator	For Hub Operators using Key Vault: Keys are stored in the HSMs, and are secured using the hardware manufacturer’s cryptographic security. The metadata on keys in stored in Infrastructure Provider Storage in an encrypted state, which is unique to each key vault.	Hub Operators are responsible for managing cryptographic keys and documenting all related procedures.	Not applicable

Completed all 252 PCI DSS requirements – See https://github.com/mojaloop/project/issues/1449

Standard Baseline – Overview of Findings

Key Highlights:

- Hub Operator/Mojaloop OSS partnership addresses all of 229 Card Holder Environments (CHE) Requirements
- Hub Operator/Mojaloop OSS shared responsibility model for continuity and consistency
- Overall, the PCI DSS standard fits very well with both Traditional and Cloud Native/Microservices Architectures



Standard Baseline - *PCI DSS*	Requirement Overview	Qty	Category	Mojaloop OSS	Hub Operator	Shared
Secure Network	Requirement 1 : Install and maintain a firewall configuration to protect cardholder data	37	CHE	0	37	0
	Requirement 2 : Do not use vendor-supplied defaults for system passwords	9	CHE	0	4	5
Protect Cardholder Data	Requirement 3 : Protect stored cardholder data	19	CHD	0	9	10
	Requirement 4 : Encrypt transmission of cardholder data across open, public networks	4	CHD	0	4	0
Vulnerability Mngt	Requirement 5 : Protect all systems against malware and regularly update anti-virus	7	CHE	0	7	0
	Requirement 6 : Develop and maintain secure systems and applications	20	CHE	0	1	19
Access Control	Requirement 7 : Restrict access to cardholder data by business need to know	11	CHE	0	1	10
	Requirement 8 : Identify and authenticate access to system components	25	CHE	0	14	11
	Requirement 9: Restrict physical access to cardholder data	29	CHE	0	29	0
Manage Network	Requirement 10: Track and monitor all access to network resources and cardholder data	32	CHE	0	31	1
	Requirement 11: Regularly test security systems and processes	35	CHE	0	35	0
	Requirement 12: Maintain a policy that addresses information security for all personnel	20	CHE	0	4	16
Total CHD		23	CHD			
Total CHE		229	CHE			
Grand Total		**252**	**CHE & CHD**		**176**	**72**

Next Steps:

- Detailed analysis findings to identify Mojaloop addressable gaps
- Validation through risk assessment and threat modelling
- Architecture alignment and Improvement

Standard Baseline Outcomes

Alignment to Best Practice as part of our control standard & centric security approach

Key outcomes and its benefits to Mojaloop OSS and Hub Operators:

Standard Baseline Report	**Responsibility Matrix**	**Input into the CPM Design**	**Regulatory Reporting**	**Security Baseline Library**
Baselined Mojaloop against mainly *the Cardholder Environmental (CDE) requirements (**229**) of the PCI DSS standard.*	Developed *a responsibility matrix* to clearly segregate Mojaloop OSS and Hub Operator in-scope items	Factored the all the cryptographic requirements of the PCI DSS into the Cryptographic Processing Module Design	Provided a standard-based security baseline matrix ready to be shared with authorities and regulators	Provided templates to be leveraged by PCI DSS complying Hub Operators to fast tract their PCI DSS compliance or baselining programs

All PCI DSS Standards Requirements(252) maps 100% to both NIST 800-53 and ISO 27000 Series

Mojaloop API Touch Points

Area	Description	Applicability	Components	Source
Secure Network	Secure network and system configuration	Mojaloop Hub	Infrastructure & Connectivity	Best Practice/ PCI DSS
Access Control	Access restriction measures on a least access model	Mojaloop Hub	Mojaloop Core & Infrastructure	Best Practice/ PCI DSS
Card Holder Data	Payment sensitive data	All Mojaloop API's	Mojaloop Core, LPS & CPM	Best Practice PCI DSS
Vulnerability Mgmt.	Secure Development & Malicious Code Protection	All Mojaloop API's	All Hub Components	Best Practice/ PCI DSS

Data Privacy and Security - Report Out

By Victor Akidiva

Data Privacy and Security Objective

1. Identify PII data (At Rest and In-Transit)
2. Explore Data Protection Standard for Data at Rest
3. Explore Data Protection Standard for Data In-Transit
4. Document Logging & Forensic Data Controls

PII Data Broad Classification (GDPR and PCI-DSS)

In GDPR PII is any information that can be used to distinguish or trace an individual’s identity. It is divided into broad areas:

1. Linked data - Directly attributed to a natural person such as name, social security number, Identification number, passport number, date and place of birth, mother’s maiden name, or biometric records.
2. Linkable data - any other information that is linkable to an individual, such as medical, educational, financial, and employment information.
3. Others - Additional data which when used with other data about a natural person may provide additional information but is useless on its own e.g. cookies, IP address, device ID.

PCI DSS define PII as:

1. Cardholder data such as the cardholder’s names, the primary account number, and the card’s expiration date and security code.
2. Sensitive authentication data, including magnetic-stripe data, the equivalent data contained on a chip, and PINs.

Our approach looked at data protection from a holistic perspective with generic recommendations that can be made specific at implementation.

— If we guard our toothbrushes and diamonds with equal zeal, we will lose fewer toothbrushes and more diamonds

Key Findings - Data Protection Data at Rest / Motion

1. PII Data is stored in Mojaloop database and Application logs. Some examples:
   a) MSISDN
   b) FirstName
   c) MiddleName
   d) Lastname
   e) Date of Birth
   f) Nationality

2. Credentials are stored within application logs (especially application initialization logs)

3. KAFKA topics, producers, clients and streams use plain text information exchange

4. Kafka Audit logs not enabled by default - Kafka does not provide a mechanism for maintaining a record of authorization decisions out of the box.

5. Audit logging not enabled within Kubernetes clusters. We noted normal system logs and default security details are available.

6. PII data within Mojaloop databases is not encrypted. The storage (Azure) is encrypted.

Data Protection Standard - Logging

Best practice security recommends that any application land/or service logs contain adequate amount of information to identify, investigate and resolve adverse security events (what, where, who, when, outcome):

1. Audit and Accountability - disabling logging, deletion of logs
2. Security Operations - application shutdown, communication errors.
3. Security Administration - enable/disable security policies, change in user rights, log settings, certification services etc
4. Authentication - Login attempts (success and failure), account lockouts, remote access
5. Authorization - actions performed by privileged accounts
6. System Administration - Application component installation and changes e.g. module installation / deletion
7. API calls - Successful and unsuccessful API requests

Objective - obtain a chronological record of activities to provide an independently verifiable trail that permits reconstruction, review and examination to determine the original sequence of attributable transactions.

Data Protection Standard - Log & Forensic Data

Mojaloop application logs are captured via the event framework and logged into EFK. The following are recommendations for logging security within Mojaloop:

1. Remove sensitive information from application logs e.g. passwords.

2. Develop standard processes for log management - This will define what is being logged, where its being logged and in what format. This is specific to security logs.
   a) This includes log generation, transmission, storage, analysis, and disposal.
   b) Log Management processes (multiple sources with unstructured log formats)

3. Define standardised security logging format for application logs.

4. Create and maintain log management infrastructure - Log management infrastructures typically perform several functions that support the analysis and security of log data.

5. Secure logs with appropriate access protocols

Enabling detailed security logging needs to be tested to ascertain performance implications.

Data Protection Standard - Kafka/Zookeeper

Kafka supports cluster encryption and authentication, including a mix of authenticated and unauthenticated, and encrypted and non-encrypted clients.The following are some of our recommendations:

1) **Encryption of Data In-Flight Using SSL/TLS** - It keeps data encrypted between our producers and Kafka, as well as our consumers and Kafka. This will involve configuring applications to always use encryption when reading and writing data to and from Kafka.

2) **Authentication Using SSL or SASL** - To authenticate our Kafka Cluster, SSL and SASL allow our producers and our consumers to verify their identity.You can define that only specific applications are allowed to connect to your Kafka cluster.

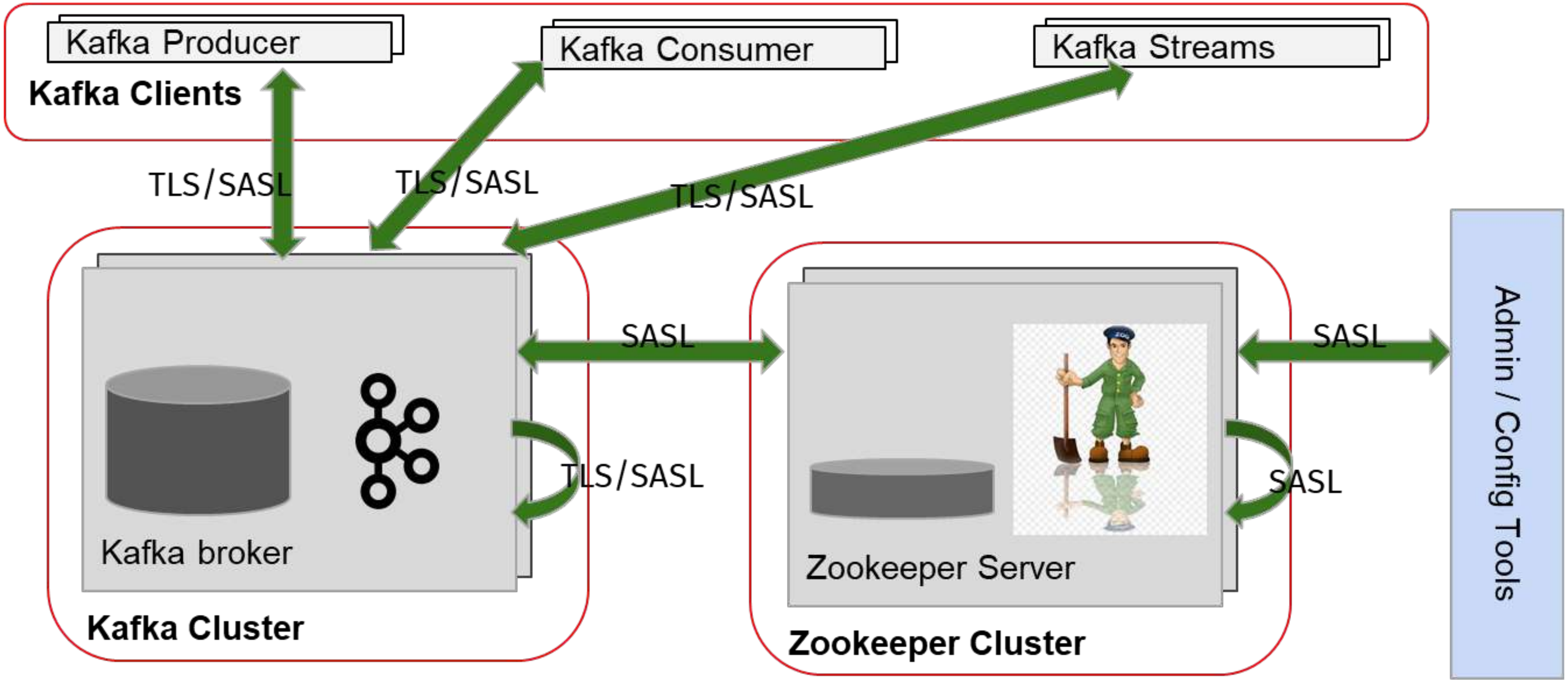
3) **Authorization Using ACLs** - enable client authorization of read and write operations by your applications. For example, you can define that only specific applications are allowed to read from a Kafka topic.

4) **Enforce quotas and throttling** where applicable depending on business use cases.

5) **Restrict access to Zookeeper via network segmentation.**

Enabling Kafka security needs to be tested to ascertain performance implications.

Secure Kafka/Zookeeper Architecture Overview



Mojaloop API Touch Points

Area	Description	Applicability	Components	Source
PII Data	Identification PII data at rest & in-transit	All Mojaloop API except PISP	All DB's and Kafka	Best Practice /GDPR
Log Data	Secure & immutable transaction logs	All data stores and repositories	Mojaloop Core, All DB's and Kafka	Best Practice/ Data Protection
Transaction data	Full call and data flow tracing	All Mojaloop API's except PISP	Mojaloop Core, All DB's and Kafka	Best Practice/ Data Protection
3rd Party Data	Secure 3rd Party data exchange	Party Lookups, Pathfinder, DFSP	Oracle, Mongo DB and Kafka	Best Practice/ Data Protection

Next Steps

1. Explore Kafka + Zookeeper security enhanced architecture
2. Test impact of enabling Kafka security on performance (design architecture + implementation)
3. Investigate PKI architecture for Kafka+Zookeeper security (HSM will help here)
4. Investigate enabling auditing information in Kubernetes, Kafka and MySQL
5. Database and impact on performance.
6. Investigate disabling / removal of sensitive information in application logs.
7. Document a standard Mojaloop security logging framework around existing Event Framework

mojaloop

DevOps & CI/CD Code Quality & Security Integration Support - Report Out

By Lewis Daly

DevOps & CI/CD Code Quality & Security Support

Objective

Ensure that all deployed tools are maintained and updated in line with the changing threat landscape and provide vulnerability management services (identify, analyze, prioritize and mitigate issues):

- DevOps tools maintenance
- Update of vulnerability databases
- Set, maintain and respond to GitHub security vulnerabilities alerts
- Manage the overall vulnerability management process – Vulnerability detection, Analysis and Mitigation

Key Tasks Completed this PI

- Monthly Github security alerts review – analysis and resolution
- Update npm audit checker to scan production modules only
- npm audit resolved/ignored where applicable
- Open followup tickets for more in-depth dependency reviews where needed

Mojaloop API Touch Points

Area	Description	Applicability	Components	Source
GitHub Security Alerts	Management of GitHub Security Alerts	All Mojaloop API’s	All GitHub Repositories	DevSecOps Best Practice
Refinement of Security Digests	Refinement of the security digest to reduce noise	All Mojaloop API’s	All GitHub Repositories	DevSecOps Best Practice
NPM Audit	Regular cleanups of npm audits	All Mojaloop API’s	All GitHub Repositories	DevSecOps Best Practice
Security Patches	Manage all security patches and updates	All Mojaloop API’s	All supported versions	DevSecOps Best Practice

Cryptographic Processing Module Design Overview – Report Out

By Max Gysi

Cryptographic Processing – Objective & Progress

Overview:

To begin to bring full transaction and data security into the Mojaloop Ecosystem, including the management of *keys and signing/encryption of data* approportionate to risk (For OSS) and/or as mandated by compliance requirements (For Hub Operators).

What has been archived so far in this PI:

1. Key agreement reached is that, wherever possible international standards will be used
   a) The KMIP, standard for connecting to Key Management systems will be used for connections to security systems
   b) PKCS11 for the online security processing where possible

2. TPS Calculation Model – Customized for Mojaloop

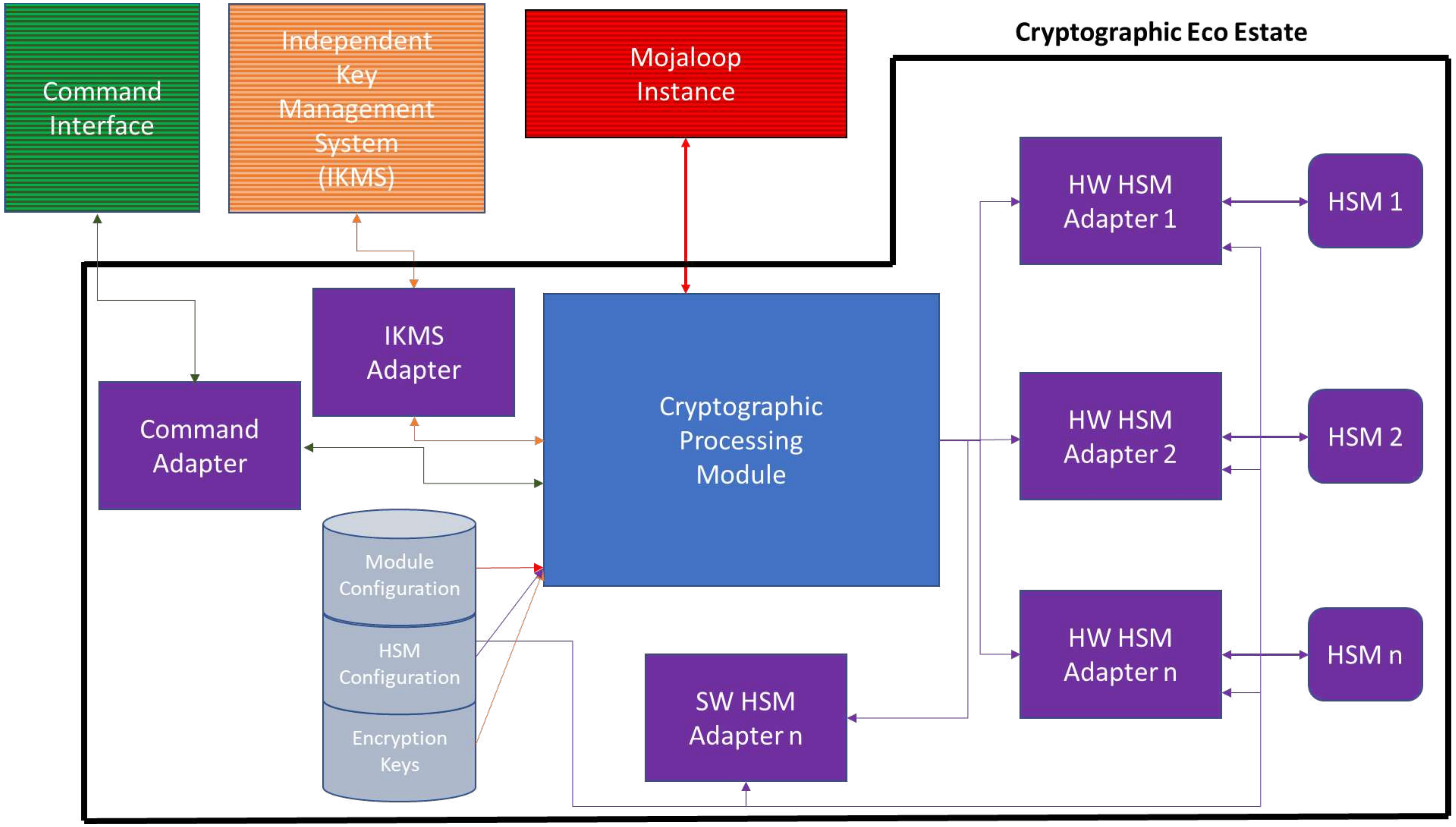
3. High level design – Under Review/QA

   a) The cryptographic side of the system (Cryptographic Processing Module – CPM) will reside outside the main Mojaloop system. This will minimize the impact on the current system

   b) The design must be modular in order to

      - Have the ability to bring in new HSM vendors or a new Key Management standard/system in quickly
      - Any new HSMs or standards can be introduced with no impact on the current implementation

Cryptographic Processing Module – Design Overview



CPM Mojaloop API Touch Points

Area	Description	Applicability	Components	Source
Cryptographic Processing Module (CPM)	Provide all cryptographic functions	All Mojaloop API's	Mojaloop Switch & Adaptors(HSM and IKMS)	Best Practice
HSM Adapters	Interface to the HSM crypto subsystem	CPM	HSM Systems	PKI Best Practice
IKMS Adaptors	Interface to the external IKMS systems	CPM	IKMS Systems	Best Practice/ PKI and PCI DSS

In Closing - Targets for PI 11

1. Refactor Documentation – Classify, Update and Publish on Mojaloop.io
2. Approval the CPM High Level Design and start with the low-level designs(LLD's)
3. Detailed analysis of the baseline standard findings and identify addressable scope for OSS
4. Development Data Privacy and Security Standards Implementation plans – HLD and LLD's
5. Continuous improvement and Support on DevOps & CI/CD Tool, Processes and Policies

muchas gracias

mojaloop

Thank you

Questions and Comments

Old Indian Proverb “ *Work is not done by a magnificent plan or strategy; work is done, when is done, and done by people.”*