Code Quality & Security Workstream

PI-17 Workstream Feedback Session – 26 January 2022

Presenter

Godfrey Kutumela

Team Members - Aime Bukasa (Security Developer and Engineer) Kim Walters, Victor Akidiva, Pedro Barreto, Michael Richards, Simeon Oriko, Miguel de Barros, Sam Kummary, Lewis & Tom Daly

Workstream Overview

The problem today:

- Evolving Cybersecurity threat landscape and regulatory compliance mandates.

Objective:

- *Continuously improve the Trust (reliability, transparency, privacy, compliance, quality and security)* of the Mojaloop Platform and transform our approach to quality and security in line with L1P principle on data privacy and emerging technological trends.

Delivery Model:

- Supports *functional and non-functional* requirements working with other *workstreams & governance committees o*n a *shared responsibility Model.*

Approach:

- Standards and Control Centric – Define and maintain Mojaloop software quality and security standards and guidelines – In certain areas we provide reference implementation whereas for other areas we require certain policies or standards to be adhered to.
- Risk and Threat Centric – Perform risk assessments and threat modelling to identify, validate, classify & prioritize security requirements.

Milestones:

- PI 1 – 8 : Foundation Phase - Built-in foundational confidentiality and Integrity as part of the Core Mojaloop Architecture.
- PI 9 – Current: Phase 5 (One Loop for all) – Assess residual risk, close critical gaps & move securely to production.
  - ✓ Introduced a risk and threat driven approach
  - ✓ Embedding Security into the reference architecture, new functionality additions and DevOps processes

✓ **When this is done :** Delivery of a trustworthy platform that meets market demands *by embedding security into the core platform* (V1 and V2) and *alignment with best practice industry standards* to minimize the risk of a data breach and prevent potential reputational risks.

PI 16 Objectives

1. Reference architecture security domain bounded contexts identification, design and implementation

1. Fraud and risk management system security review and validation

2. Continuously vulnerability management activities and DevSecOps process/tool enhancements

3. Provide quarterly Open-Source security (OSS) reports – Tracking our commitment to make the platform completely free to use!

4. Engagement with the implementation teams – Support for implementors and learning from their experiences

Reference architecture security domain

The primary goal is to embed and decouple security functions (AuthZ, AuthN, Logging, Auditing and Crypto) into the reference architecture platform core.

Completed to date – PI 14- 16:

1. Documentation of the security bounded contexts
   - *Aligns well with good security architecture development practices*
2. High-level architecture definition and documentation
3. Low-level architecture definition and development common artifacts
4. Started with the Implementation of the Alpha version

Planned for the next 2 quarters – PI 17 & 18:

1. Completion of Alpha and Beta
2. Security testing and validation
3. Move to production

Mojaloop security architecture implementation progress

Reference Architecture Security BC Implementation progress – 58%

1. Crypto
2. KMS
3. AIM
4. AuthN
5. AuthZ
6. JWT generation
7. JWT verification
8. Token refresh (TBD)
9. Security BC Bootstrap
10. Security BC Startup
11. API calls with JWT

Ref: reference-architecture-doc/index.md at patch-1 · bukasaaime/reference-architecture-doc (github.com)

Low Level Design – 70%

1. Reference Architecture Security BC's – On going
2. Infrastructure as code (Serverless implementation platform on AWS for delivering security specific components only). Completed.
3. K8 implementation. Ongoing

Overall Build Progress - 35%

1. Infrastructure as code (Serverless implementation) for registration, login and JWT token verifications APIs available. Local libraries.
2. K8 implementation. Ongoing.
3. Crypto adapter. Planned
4. AIM adapter. Planned

Mojaloop security architecture implementation is also open for community contributions so if interested please reach to me or Simeon!

Fraud and Risk Management System security implementation review and validation

The objective is to review and validate security implementation on all new major functionality additions to ensure alignment with mojaloop security standards and policies.

Completed to date – PI 14- 16:

1. Defined the security requirements/principles for FRMS
2. Performed an Open-source scan on the current codebase

Planned for the next 2 quarters – PI 17 & 18:

1. Review detailed design and plans
2. Security testing and validation

Fraud and Risk Management System Open-source security (OSS) scan results overview

OSS scan objectives

- Open-Source License Assessment: Open-source utilization is appropriately controlled to minimize intellectual property risks.
- Open-Source Security Assessment: Open-source versions used are not vulnerable and up to date, minimizing security risks for using older versions of open-source software.

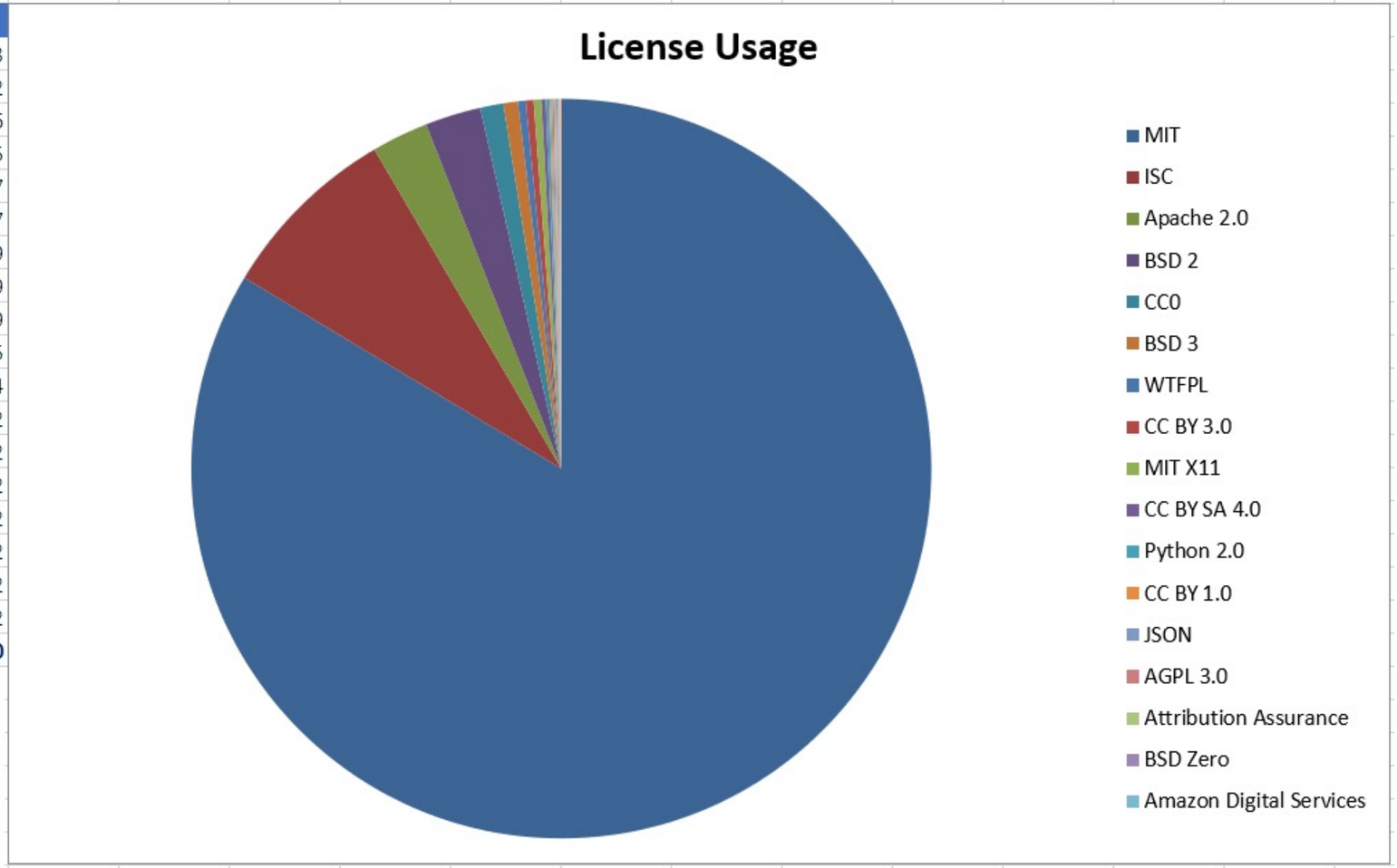
Upon Analysis:

- The very high- and high-risk libraries are dual licensed with permissive licenses, BSD and MIT.
- All libraries with high security vulnerable have a fix available via an upgrade.

Fraud and Risk Management System Open-source security (OSS) scan results overview

Overview of the FRMS license landscape

License	CC
MIT	2233
ISC	212
Apache 2.0	66
BSD 2	65
CC0	27
BSD 3	17
WTFPL	9
CC BY 3.0	9
MIT X11	9
CC BY SA 4.0	5
Python 2.0	4
CC BY 1.0	2
JSON	2
AGPL 3.0	2
Attribution Assurance	2
BSD Zero	2
Amazon Digital Services	2
GPL 3.0	2
Total	**2670**

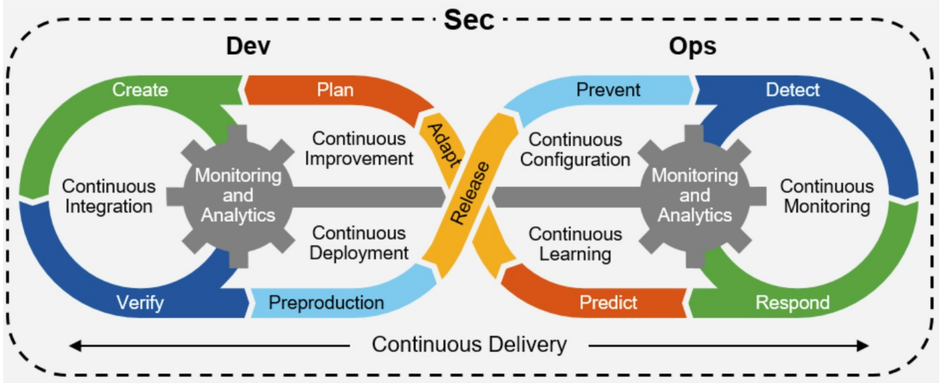


Continuously vulnerability management and DevSecOps process/tool enhancements

On going maintenance and enhancements of the DevSecOps processes, policies and tools.

<u>On going support Activities:</u>

1. Regular Security Patches + Updates
   - Addressing regular Dependabot and Snyk security alerts
   - Running `npm audit` on flagged repos
   - Improving CI/CD Workflows and adding new policies as needed
   - Exploring with automated releases
   - Benchmarked npm Audit against WhiteSource – once a quarter
2. Manage community vulnerability disclosure process
   - 6 vulnerabilities disclosed this PI
     - ✓ 1 on the core repos (exposing of credential) – Resolution is ongoing
     - ✓ 5 on the learning platform (OWASP related) – To be resolved in the new platform if they persist



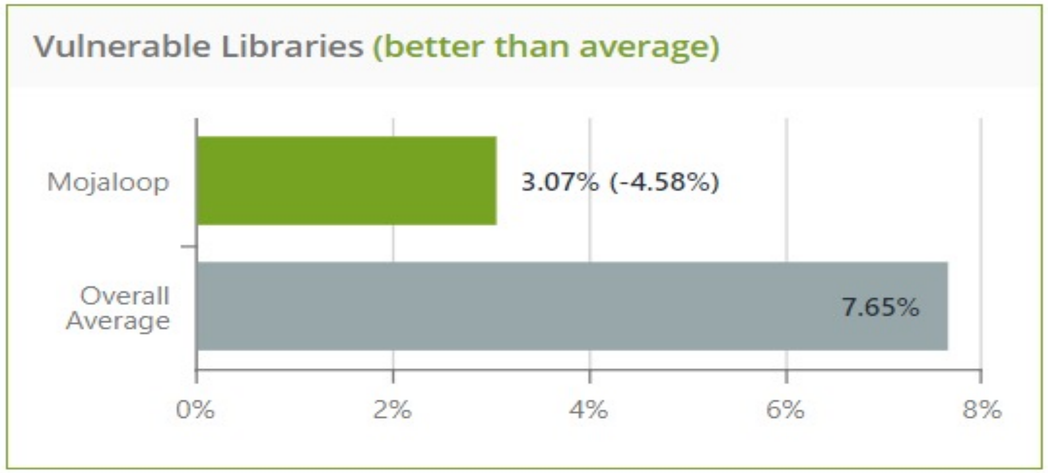
<u>Planned Enhancements for PI 17</u>

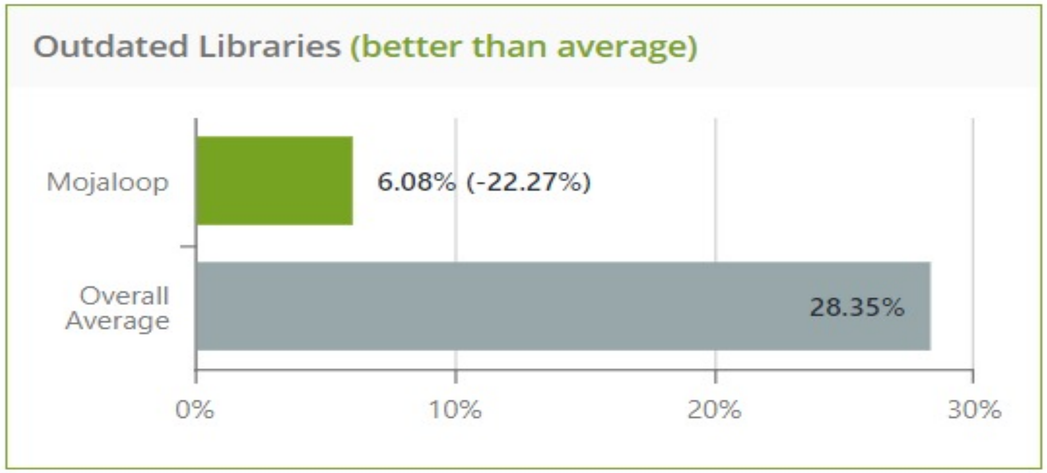
1. Establish code security metrics and enforce using a static code analyser - https://github.com/mojaloop/project/issues/2633
2. Review and improve code signing measures https://github.com/mojaloop/project/issues/2634

Quarterly Open-Source usage security reports on the core Mojaloop modules

According to WhiteSource Benchmark database, Mojaloop is better than average in vulnerable libraries and in outdated libraries, potentially indicating that Mojaloop have an effective security vulnerability scanning capability and version management controls.

- The scan was performed on 104 code repos and detected about 7055 unique libraries:
  - 6625 libraries are up to date – Excellent version management!
  - 430 libraries are outdated and should be reviewed for upgrades
  - 130 libraries with multiple version and should be upgraded to the most updated version.
  - 217 libraries are vulnerable – 113 High Risk and 103 Medium Risk - All should be reviewed and mitigated (Mostly NPM transitive dependencies)





Quarterly Open-Source usage security reports on the core Mojaloop modules

Upon Analysis:

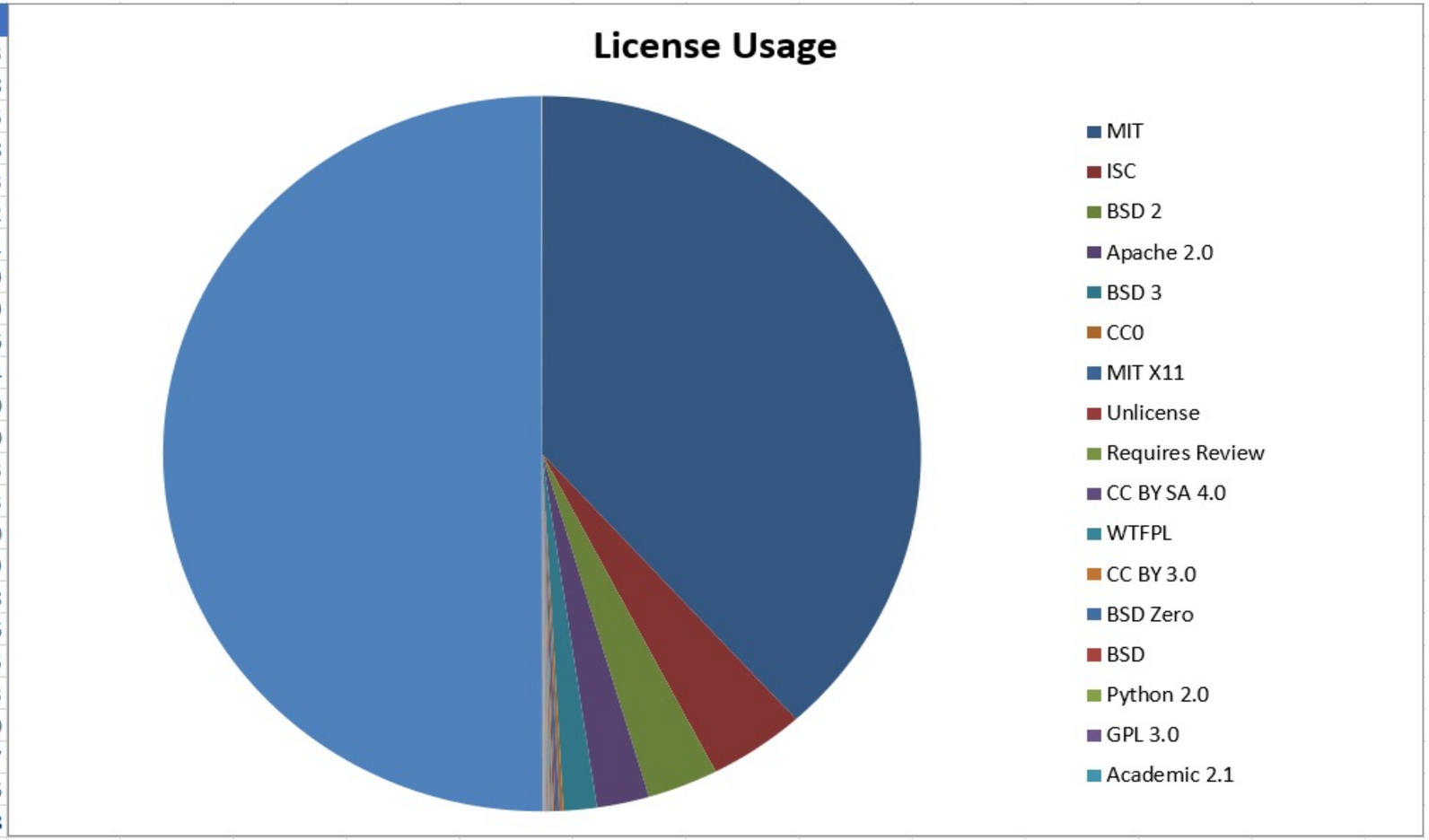
- Out of 14 high-risk license libraries:
  - ❖ 6 libraries are dual licensed with permissive and apache compatible licenses, MIT and BSD.
  - ❖ 8 libraries (fuzzball, cbor, commons, docker-hub-api, jxon, qt-everywhere, types &utils) have only GPL licenses available and should be replaced to comply with the Mojaloop policy, Apache 2.0 compatible.

- Out of 116 high-risk security libraries:
  - ❖ 6 libraries does not have fixes available and should be reviewed and mitigated.
  - ❖ The rest of the libraries have a fix available and should be upgraded.

Quarterly Open-Source usage security reports on the core Mojaloop modules

Overview of Mojaloop core modules license landscape

License types	
MIT	29963
ISC	3218
BSD 2	2365
Apache 2.0	1748
BSD 3	1113
CC0	132
MIT X11	131
Unlicense	69
Requires Review	49
CC BY SA 4.0	46
WTFPL	34
CC BY 3.0	30
BSD Zero	30
BSD	23
Python 2.0	23
GPL 3.0	20
Academic 2.1	19
CC BY 4.0	18
GPL 2.0	16
Eclipse 1.0	15
Mozilla 2.0	13
ODC Open Database	10
bzip2	7
Suspected Unspecified License	6
Total	**39098**



Top security and compliance concerns from implementation teams

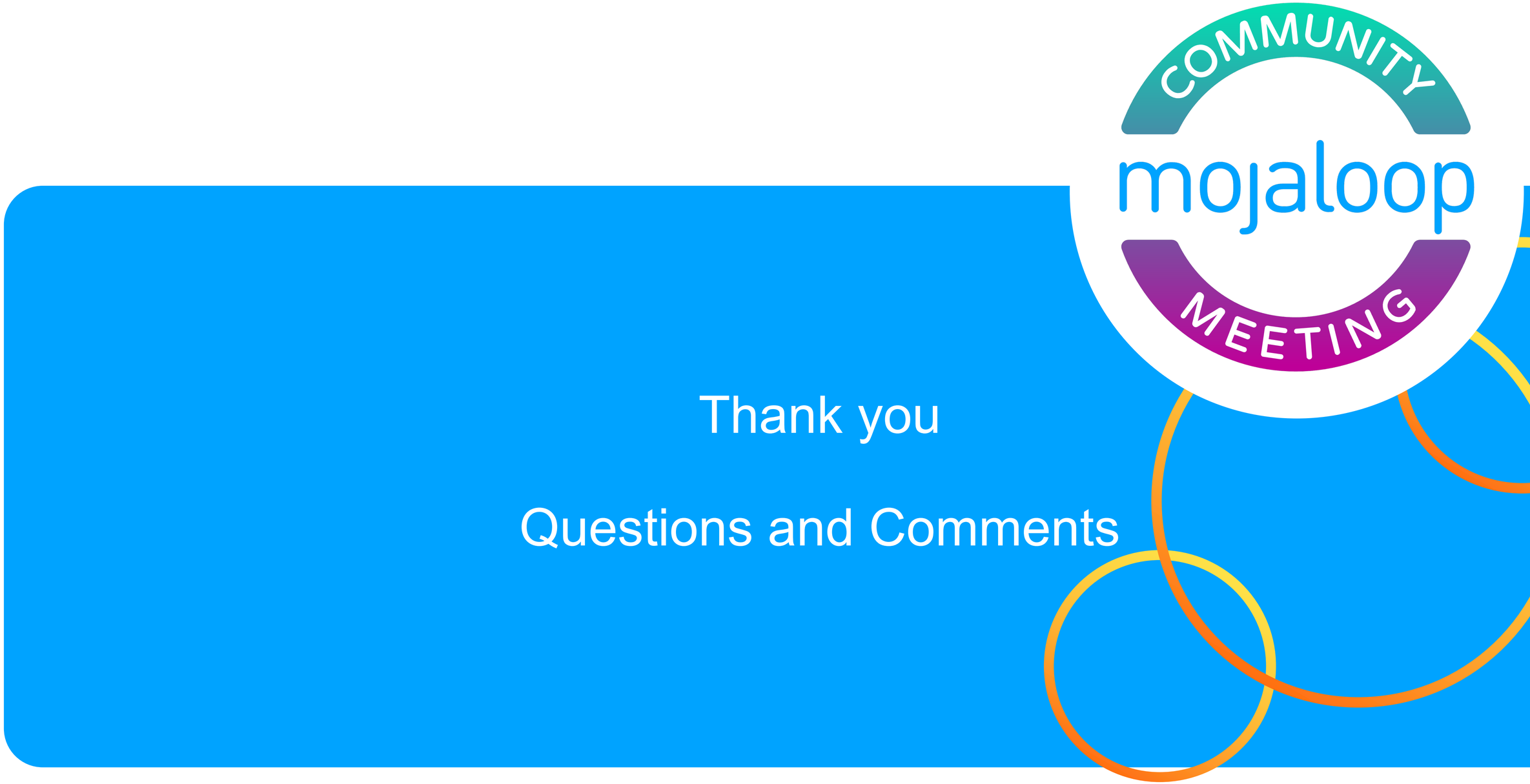
Business

- User activity auditing capabilities
- Separation of duties and role-based access control
- Directory services abuse and DOS attacks
- Dispute resolution measures and processes
- Refund management/payment reversals
- Independent security audits and platform certification
- Fraud prevention and monitoring including AML
- Sharing KYC information with participating DSFP’s
- Data privacy matters
- Continuity of business and support assurance

Technical

- Robust IAM controls including privileged user management
- Code security measures to prevent malicious intruders
- Payment grade cryptographic controls (HSM type solution desired)
- Security monitoring and incident response capabilities
- Provision of sufficient security information to fulfil third party audits from mainly settlement banking partners
- Business continuity and disaster recovery

muchas gracias



Thank you

Questions and Comments