SMART INDIA HACKATHON 2025

TITLE PAGE

- **Problem Statement ID –** SIH25179
- **Problem Statement Title-** Quantum Secure Email Client Application
- **Theme-** Blockchain & Cybersecurity
- **PS Category-** Software
- **Team ID-**
- **Team Name (Registered on portal) :** Caffeinated Stumblers

Caffeinated Stumblers

IDEA TITLE

SMART INDIA HACKATHON 2025

To create a next-generation email client that delivers unconditional, future-proof security by integrating Quantum Key Distribution (QKD) with existing email infrastructure. "QuMail" will protect confidential communication from all known and future threats, including quantum computers.

Core Ideas:

- **Secure Key Management:** Interfaces with a Key Manager (KM) to retrieve and manage quantum keys using the ETSI GS QKD 014 REST-based API protocol, ensuring interoperability.

- **Multi-Level Security:**

Level 1 (Quantum Secure): Uses One-Time Pad (OTP) with a QKD-generated key for provably unbreakable security.

Level 2 (Quantum-aided AES): Uses a QKD key as the seed for a robust AES-256 cipher, balancing security with performance.

Level 3 (Hybrid PQC): Provides an option to use a Post-Quantum Cryptography (PQC) algorithm for secure key exchange.

Level 4 (No Quantum Security): Standard encryption (TLS/SSL) for interoperability with non-QuMail users.

Key Misuse Detection: Tracks and alerts users if a key is reused, preventing a critical security violation.

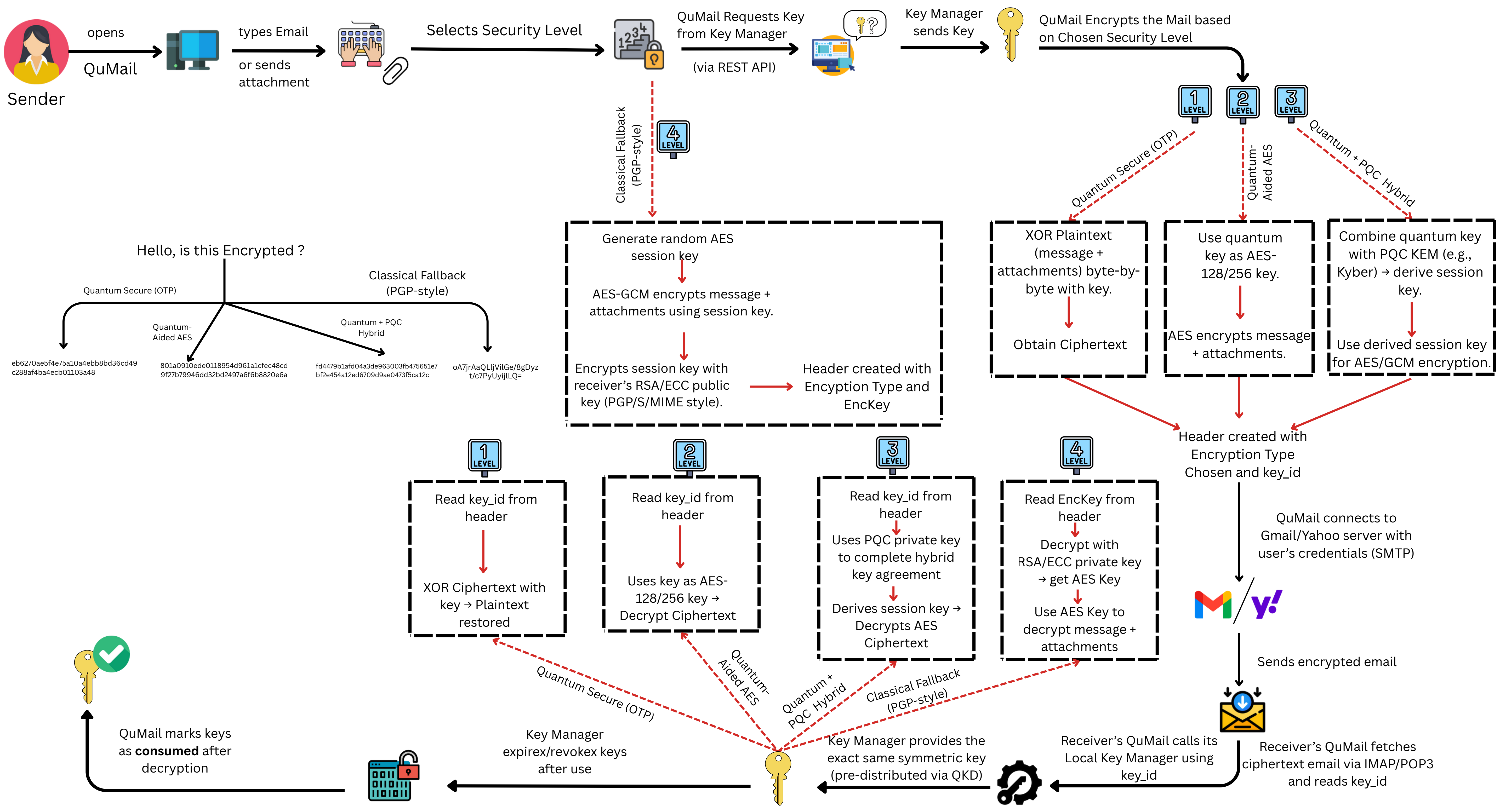
Interoperability Mode: Allows export of encrypted messages as .qmail files, which can be decrypted with a standalone tool and a key, enabling secure communication with non-QuMail users.

Attachment Integrity Check: Adds a digital signature/HMAC to attachments using a key from the KM to verify file integrity and detect any tampering.

Caffeinated Stumblers

TECHNICAL APPROACH

SMART INDIA HACKATHON 2025



Sender

opens

QuMail

types Email

or sends attachment

Selects Security Level

QuMail Requests Key from Key Manager

(via REST API)

Key Manager sends Key

QuMail Encrypts the Mail based on Chosen Security Level

1 LEVEL — Quantum Secure (OTP)

XOR Plaintext (message + attachments) byte-by-byte with key.

Obtain Ciphertext

2 LEVEL — Quantum-Aided AES

Use quantum key as AES-128/256 key.

AES encrypts message + attachments.

3 LEVEL — Quantum + PQC Hybrid

Combine quantum key with PQC KEM (e.g., Kyber) → derive session key.

Use derived session key for AES/GCM encryption.

Header created with Encryption Type Chosen and key_id

QuMail connects to Gmail/Yahoo server with user's credentials (SMTP)

Sends encrypted email

Receiver's QuMail fetches ciphertext email via IMAP/POP3 and reads key_id

Receiver's QuMail calls its Local Key Manager using key_id

Key Manager provides the exact same symmetric key (pre-distributed via QKD)

Quantum Secure (OTP)

1 LEVEL

Read key_id from header

XOR Ciphertext with key → Plaintext restored

Quantum-Aided AES

2 LEVEL

Read key_id from header

Uses key as AES-128/256 key → Decrypt Ciphertext

Quantum + PQC Hybrid

3 LEVEL

Read key_id from header

Uses PQC private key to complete hybrid key agreement

Derives session key → Decrypts AES Ciphertext

Classical Fallback (PGP-style)

4 LEVEL

Read EncKey from header

Decrypt with RSA/ECC private key → get AES Key

Use AES Key to decrypt message + attachments

Key Manager expirex/revokex keys after use

QuMail marks keys as **consumed** after decryption

Classical Fallback (PGP-style)

4 LEVEL

Generate random AES session key

AES-GCM encrypts message + attachments using session key.

Encrypts session key with receiver's RSA/ECC public key (PGP/S/MIME style).

Header created with Encyption Type and EncKey

Hello, is this Encrypted ?

Quantum Secure (OTP)

eb6270ae5f4e75a10a4ebb8bd36cd49 c288af4ba4ecb01103a48

Quantum-Aided AES

801a0910ede0118954d961a1cfec48cd 9f27b79946dd32bd2497a6f6b8820e6a

Quantum + PQC Hybrid

fd4479b1afd04a3de963003fb475651e7 bf2e454a12ed6709d9ae0473f5ca12c

Classical Fallback (PGP-style)

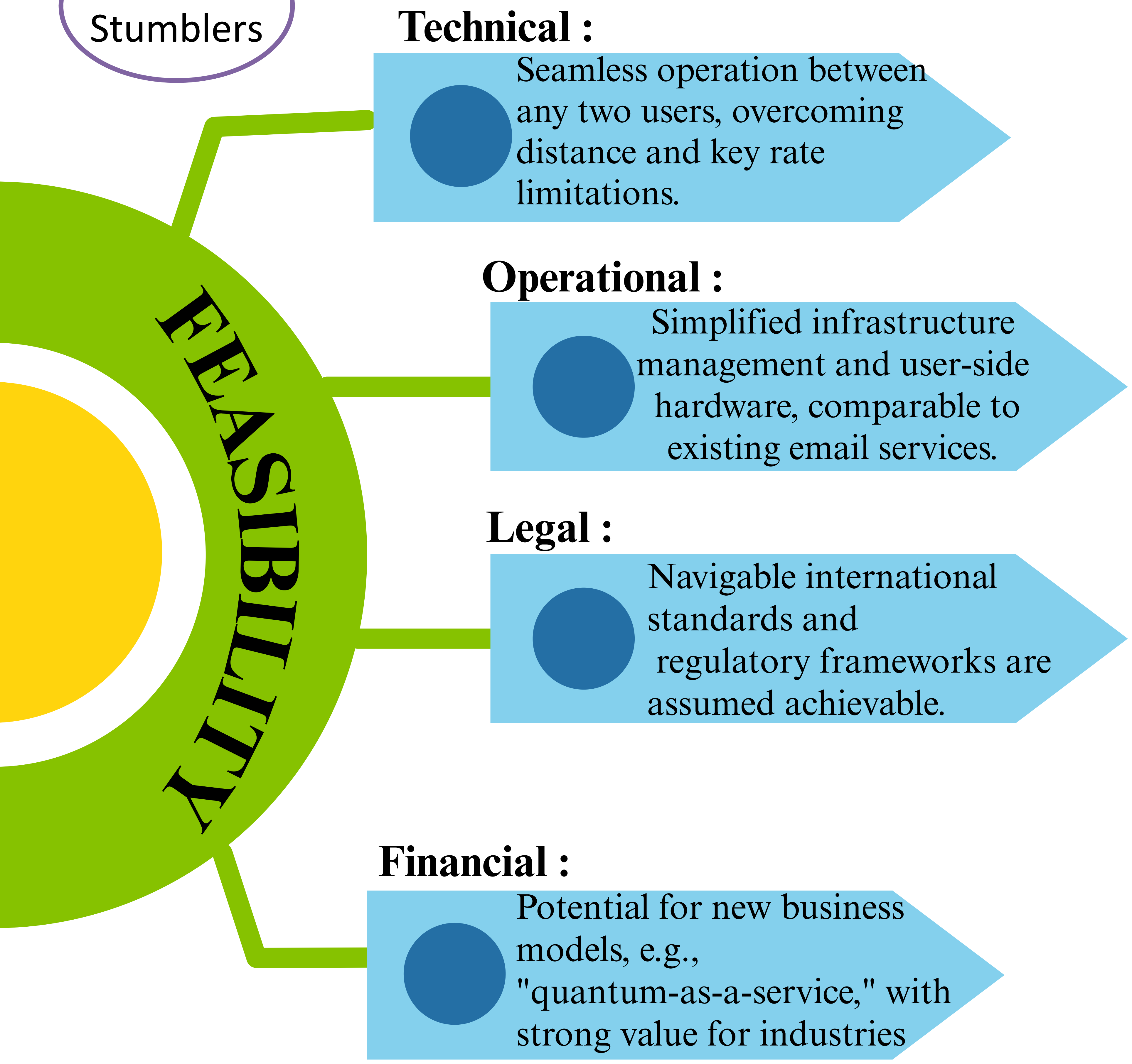
oA7jrAaQLljVilGe/8gDyz t/c7PyUyijlLQ=

- **Frontend :** Electron.js (Node.js runtime inside) OR native Windows (.NET WPF)
- **Local backend** / service (can be embedded in Electron or separate): Node.js (Express/Fastify) or Python (FastAPI) — this implements KM client, key reservation, encryption, DB access, zeroization helpers.
- **KME (Key Manager Emulator)**: Local REST service (FastAPI / Express) that mimics ETSI GS QKD 014 REST interface.
- **Email transport:** SMTP for sending (authenticated), IMAP for receiving. Use libraries that support TLS and OAuth where needed.
- **Local storage**: SQLite for mail metadata, key usage ledger, offsets.
- **Crypto stacks**: production-grade crypto libs + OQS (Open Quantum Safe) for PQC.

Caffeinated Stumblers

FEASIBILITY AND VIABILITY

SMART INDIA HACKATHON 2025



Technical :

Seamless operation between any two users, overcoming distance and key rate limitations.

Operational :

Simplified infrastructure management and user-side hardware, comparable to existing email services.

Legal :

Navigable international standards and regulatory frameworks are assumed achievable.

Financial :

Potential for new business models, e.g., "quantum-as-a-service," with strong value for industries

Challenges:

Technical

Distance and key generation rates remain critical barriers without ideal repeaters and satellites.

Legal

Conflicts with national laws that require government access to communications. Need for standardized QKD regulations.

Operational

Complex infrastructure and user hardware management in non-ideal scenarios.

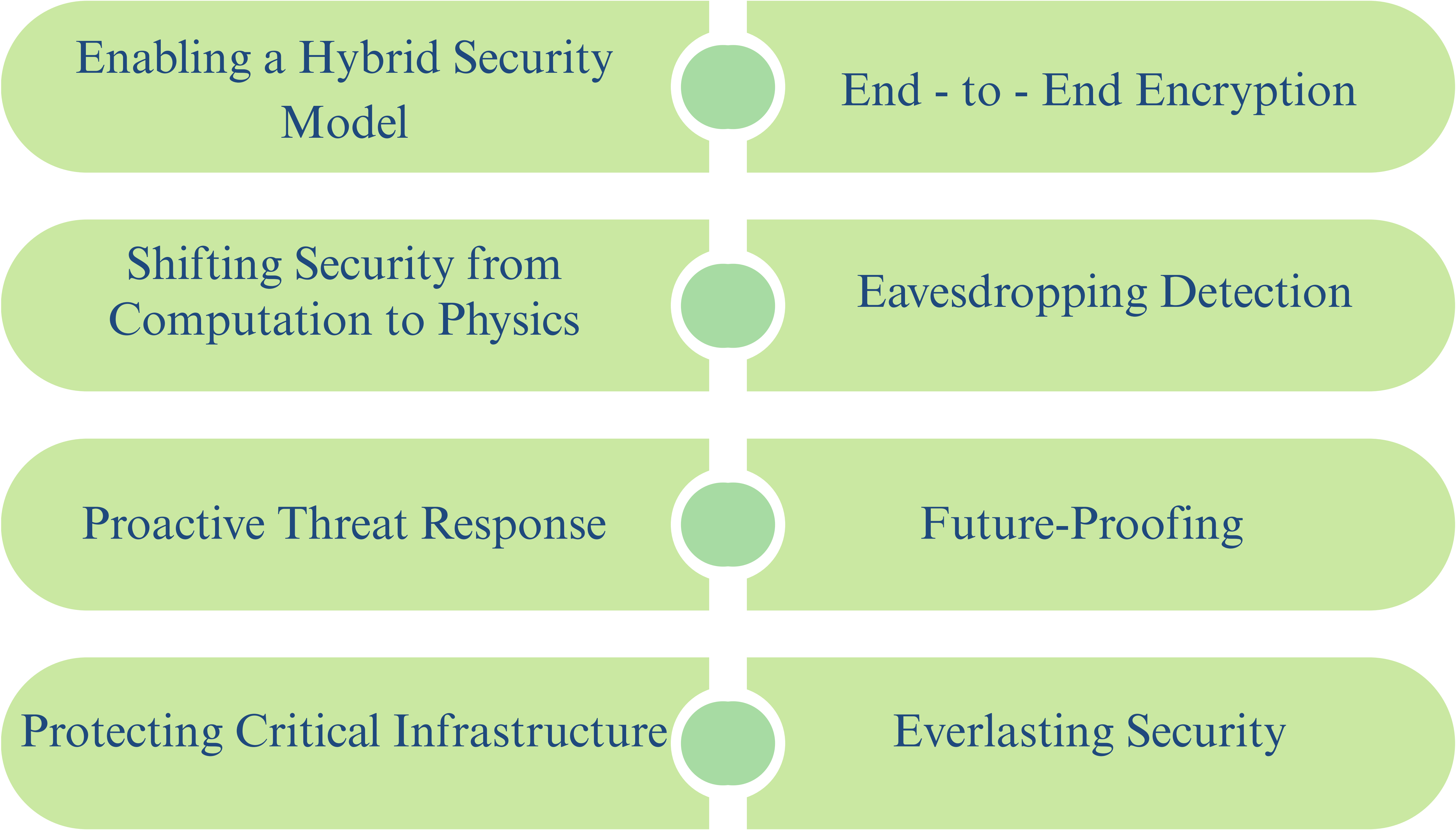
Financial

High current cost of QKD hardware and infrastructure in the present, non-ideal scenario.

Caffeinated Stumblers

IMPACT AND BENEFITS

SMART INDIA HACKATHON 2025



IMPACT	BENEFITS
Enabling a Hybrid Security Model	End - to - End Encryption
Shifting Security from Computation to Physics	Eavesdropping Detection
Proactive Threat Response	Future-Proofing
Protecting Critical Infrastructure	Everlasting Security

Caffeinated Stumblers

RESEARCH AND REFERENCES

SMART INDIA HACKATHON 2025

- Report on Post Quantum Computing - This Report was Compiled as part of an Internship accounting as proof of work for Domain Expertise.

- https://www.ibm.com/docs/en/app-connect/11.0.0?topic=applications-processing-email-messages

- ISSN: 2073-607X,2076-0930 - Quantum secured Email

- E-ISSN: :2073-607X - QKD for Robust Encryption

- https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards