Cahier des Charges pour le Développement du Produit QuantumEyes

1. Introduction

Le présent document constitue le cahier des charges pour le développement du produit QuantumEyes. Ce projet, initialement connu sous le nom de Vigilianz, vise à créer une solution de cybersécurité avancée, spécifiquement conçue pour répondre aux besoins des petites et moyennes entreprises (PME) ainsi que des collectivités. QuantumEyes se positionne comme une plateforme de Network Detection and Response (NDR) intégrant des technologies de pointe, notamment l'intelligence artificielle (IA) et le Machine Learning Quantique (QML), afin d'offrir une protection robuste et proactive contre les cybermenaces en constante évolution. L'objectif est de fournir une solution souveraine, hébergée en France, et alignée sur les standards européens, tout en étant accessible et facile à gérer pour des organisations ne disposant pas nécessairement d'équipes de cybersécurité étendues. Ce cahier des charges détaille les attentes fonctionnelles, techniques, et les objectifs du produit, en s'appuyant sur les informations fournies dans les documents de conception et de vision produit. Le développement de la solution est envisagé sur la plateforme Replit, ce qui impliquera des considérations spécifiques en termes d'architecture et de déploiement.

2. Contexte et Objectifs du Projet

Le paysage de la cybersécurité est en perpétuelle mutation, avec des menaces de plus en plus sophistiquées ciblant des organisations de toutes tailles. Les PME et les collectivités, en particulier, se retrouvent souvent démunies face à ces risques, manquant de ressources spécialisées et de budgets conséquents pour mettre en œuvre des défenses adéquates. QuantumEyes a pour ambition de combler cette lacune en proposant une solution de cybersécurité complète, accessible et souveraine. L'objectif principal est de fournir une plateforme NDR (Network Detection and Response) qui non seulement détecte les menaces connues et inconnues en temps réel, mais qui accompagne également les organisations dans l'évaluation de leur maturité cyber, la protection de leurs infrastructures et la garantie d'une continuité d'activité. Le projet met un accent particulier sur l'innovation, notamment par l'intégration d'algorithmes de Machine Learning (ML) et l'exploration du potentiel du Quantum Machine Learning (QML) pour une détection d'anomalies et une anticipation des menaces futures,

positionnant ainsi QuantumEyes comme une solution "quantum-ready". La solution doit être développée en France, opérée au sein de l'Union Européenne, et viser une conformité avec les standards et réglementations en vigueur, notamment ceux édictés par l'ANSSI.

3. Périmètre Fonctionnel

La solution QuantumEyes doit offrir un ensemble de fonctionnalités couvrant l'ensemble du cycle de vie de la gestion de la cybersécurité, articulé autour de trois piliers principaux : Évaluer, Protéger et Garantir.

3.1. Évaluation de la Maturité Cyber

Cette fonctionnalité a pour but d'aider les organisations à comprendre leur niveau de préparation face aux cyber-risques. Elle doit comprendre plusieurs aspects. Premièrement, un module d'"assessment" déclaratif permettra aux utilisateurs de s'auto-évaluer par rapport à des référentiels de conformité reconnus tels que CIS Controls, ISO 27001/27002, et d'autres cadres pertinents. Cette évaluation doit générer un rapport détaillé indiquant les points forts et les axes d'amélioration. Deuxièmement, la plateforme intégrera des capacités de scan de vulnérabilités périodiques et automatisés. Ces scans devront identifier les failles de sécurité potentielles au sein de l'infrastructure de l'utilisateur. Les résultats de ces scans devront être présentés de manière claire, avec une priorisation des vulnérabilités en fonction de leur criticité. Un tableau de bord dédié à l'évaluation permettra de visualiser l'évolution de la maturité cyber, de suivre le traitement des vulnérabilités et de présenter un indicateur global de sécurité. Ce tableau de bord devra être compréhensible tant par des profils techniques que par des dirigeants. Enfin, la solution doit proposer une "roadmap cyber" personnalisée, guidant l'utilisateur dans sa démarche d'amélioration continue de sa posture de sécurité.

3.2. Protection et Détection des Menaces (NDR)

Le cœur de QuantumEyes réside dans ses capacités de Network Detection and Response (NDR). Cette fonctionnalité doit assurer une surveillance continue du réseau de l'utilisateur pour détecter et répondre aux menaces en temps réel. La solution doit capturer et analyser les trames réseau pour offrir une visibilité complète sur les différentes couches OSI. L'approche de détection sera hybride. D'une part, une détection basée sur les signatures (signature-based) s'appuiera sur des bases de données de menaces connues (Cyber Threat Intelligence - CTI), notamment via une intégration avec des plateformes collaboratives comme OpenCTI. D'autre part, une détection comportementale (behavior-based) utilisera des algorithmes de Machine Learning (ML)

pour identifier les comportements anormaux et les menaces inconnues, y compris les attaques "zero-day". Une attention particulière sera portée à la minimisation des faux positifs. L'architecture devra être "quantum-ready", c'est-à-dire conçue pour intégrer à terme des algorithmes de Quantum Machine Learning (QML) afin d'améliorer les performances de détection et d'anticiper les futures capacités de calcul. La réponse aux menaces devra être automatisée autant que possible, avec des mécanismes de remédiation configurables. Un tableau de bord de protection offrira une vue d'ensemble des menaces détectées, des alertes en cours, et de l'état de santé du réseau.

3.3. Garantie et Continuité d'Activité

Au-delà de la détection et de la protection, QuantumEyes vise à aider les organisations à garantir la continuité de leurs activités et à faciliter la gestion des cyber-assurances. La plateforme devra permettre de mettre en place et de gérer des plans de continuité d'activité (PCA) et des plans de reprise d'activité (PRA) standardisés et adaptés aux besoins des PME. Un aspect crucial sera la création d'un référentiel commun et d'un indicateur de maturité cyber spécifiquement conçu pour les cyber-assureurs. Cela permettra de faire le lien entre le niveau de sécurité réel de l'entreprise, mesuré par QuantumEyes, et les offres d'assurance. La solution pourrait intégrer une marketplace d'offres de cyber-assurance de partenaires, proposant des conditions adaptées et potentiellement négociées en fonction du profil de risque de l'utilisateur. Cette fonctionnalité vise à rendre la cyber-assurance plus accessible et plus pertinente pour les PME.

3.4. Interface Utilisateur et Expérience (UI/UX)

L'interface utilisateur (UI) de QuantumEyes doit être intuitive, claire et facile à prendre en main, même pour des utilisateurs n'ayant pas une expertise approfondie en cybersécurité. Les tableaux de bord doivent présenter les informations de manière visuelle et synthétique, permettant une compréhension rapide de la situation. La navigation au sein de la plateforme doit être fluide et logique. Des efforts particuliers seront consacrés à l'expérience utilisateur (UX) pour s'assurer que la solution est non seulement puissante mais aussi agréable à utiliser. Des éléments de design inspirés des solutions existantes (comme les diagrammes radar pour l'évaluation) pourront être envisagés, tout en conservant une identité propre à QuantumEyes. La solution devra être accessible via un navigateur web et pensée pour une utilisation en mode SaaS.

4. Spécifications Techniques

Les spécifications techniques de QuantumEyes doivent garantir la performance, la scalabilité, la sécurité et la maintenabilité de la solution.

4.1. Architecture Générale

La solution QuantumEyes sera développée comme une application SaaS (Software as a Service), accessible via un navigateur web. L'architecture devra être modulaire pour faciliter les évolutions et l'intégration de nouvelles fonctionnalités. Elle comprendra typiquement un front-end pour l'interface utilisateur, un back-end pour la logique métier et le traitement des données, et une base de données pour le stockage des informations. Une attention particulière sera portée à l'architecture du pipeline de données pour la collecte, le traitement et l'analyse des flux réseau en temps réel. Des machines virtuelles (VM) légères, ou sondes, seront déployées chez les clients pour collecter les données réseau. Ces sondes communiqueront de manière sécurisée avec la plateforme centrale hébergée dans des datacenters en France. L'architecture doit être pensée pour être "quantum-ready", ce qui signifie qu'elle doit pouvoir intégrer des modules basés sur des algorithmes de QML lorsque la technologie sera mature, sans nécessiter une refonte majeure. Le développement sur Replit impliquera l'utilisation des services et des infrastructures fournis par cette plateforme, ce qui devra être pris en compte dans la conception architecturale (par exemple, pour le déploiement, la gestion des bases de données, et la scalabilité).

4.2. Technologies Envisagées

Le choix des technologies devra privilégier la performance, la sécurité, et l'écosystème disponible sur Replit. Pour le front-end, des frameworks JavaScript modernes comme React, Vue.js ou Angular pourraient être envisagés, en s'appuyant sur des bibliothèques de visualisation de données (par exemple, D3.js, Chart.js ou des composants Kibana si pertinent et intégrable). Pour le back-end, des langages comme Python (avec des frameworks comme Flask ou Django) ou Node.js (avec Express.js) sont des options courantes, notamment pour leur écosystème riche en bibliothèques pour le traitement de données, le machine learning et les opérations réseau. Des outils comme Wireshark (pour l'analyse de paquets) et ElasticSearch (pour l'indexation et la recherche de logs et d'événements) sont mentionnés comme des hypothèses et leur intégration devra être étudiée. Pour les bases de données, le choix dépendra des besoins (relationnel type PostgreSQL, NoSQL type MongoDB, ou des solutions managées sur Replit). L'utilisation de technologies Open Source sera privilégiée pour favoriser la souveraineté et la flexibilité. Pour la partie Machine Learning, des bibliothèques comme Scikit-learn, TensorFlow ou PyTorch seront utilisées. L'exploration du QML nécessitera de se familiariser avec les SDK quantiques disponibles (par exemple, Qiskit, Cirq, Pennylane) et d'anticiper leur intégration.

4.3. Traitement des Données et Performance

La solution doit être capable de traiter des flux de données réseau en temps réel et à haut débit. Cela inclut le décryptage en ligne (si applicable et légalement autorisé), le décodage de protocoles, et le réassemblage complet des flux pour une analyse de contenu intégrale. La télémétrie devra être entièrement programmable. Les algorithmes de Machine Learning, classiques et quantiques, devront être optimisés pour une inférence rapide et un apprentissage continu. La plateforme doit être scalable pour s'adapter à un nombre croissant d'utilisateurs et à des volumes de données variables. Des mécanismes de gestion de la charge et d'optimisation des requêtes seront essentiels. La latence dans la détection et la réponse aux menaces doit être minimisée.

4.4. Sécurité de la Plateforme

La sécurité de la plateforme QuantumEyes elle-même est primordiale. Toutes les communications entre les sondes chez le client et la plateforme centrale devront être chiffrées. L'accès à la plateforme sera protégé par des mécanismes d'authentification forte (MFA recommandé). Les données clients devront être stockées de manière sécurisée, avec des chiffrements au repos et en transit, et dans le respect strict des réglementations sur la protection des données (RGPD). Des audits de sécurité réguliers de la plateforme devront être planifiés. La gestion des vulnérabilités de la plateforme elle-même devra être proactive. L'architecture devra être conçue pour résister aux attaques courantes (injection SQL, XSS, DoS, etc.).

4.5. Intégrations

La solution devra pouvoir s'intégrer avec d'autres outils et services. Une API robuste pourrait être développée pour permettre des intégrations tierces. L'intégration avec des plateformes de CTI comme OpenCTI est un requis. Des connecteurs vers des systèmes de gestion d'incidents (SIEM, SOAR) pourraient être envisagés à terme. L'intégration avec les offres des cyber-assureurs sera un point clé de la fonctionnalité "Garantir".

5. Exigences Non Fonctionnelles

Ces exigences définissent les qualités et les contraintes de la solution.

5.1. Souveraineté et Conformité

QuantumEyes doit être une solution souveraine française, opérant en Union Européenne. Les données doivent être hébergées en France. La solution doit être conforme aux réglementations européennes et françaises, notamment le RGPD. Une

attention particulière sera portée à l'absence de dépendance à des solutions tierces situées en dehors de la zone de conformité de l'UE, notamment américaines, sauf si des garanties de conformité strictes peuvent être apportées. La solution visera une normalisation ANSSI ou, a minima, suivra ses recommandations.

5.2. Scalabilité et Disponibilité

La plateforme doit être capable de monter en charge pour supporter un nombre croissant de clients (PME et collectivités de 100 à 1500 utilisateurs) et un volume de données croissant. Elle doit offrir un haut niveau de disponibilité (par exemple, 99.9%) avec des mécanismes de redondance et de basculement.

5.3. Maintenabilité et Évolutivité

Le code doit être bien documenté, modulaire et suivre les bonnes pratiques de développement pour faciliter la maintenance et les évolutions futures. L'architecture doit permettre l'ajout de nouvelles fonctionnalités et l'intégration de nouvelles technologies (notamment QML) sans refonte majeure.

5.4. Facilité d'Utilisation et Accessibilité

Comme mentionné dans les exigences fonctionnelles, l'interface doit être intuitive et accessible à des utilisateurs non experts. La documentation utilisateur (guides, tutoriels) devra être claire et complète.

5.5. Performance

Les temps de réponse de l'interface utilisateur doivent être rapides. La détection des menaces et la génération des alertes doivent se faire en quasi-temps réel.

6. Contraintes du Projet

Plusieurs contraintes doivent être prises en compte pour le développement de QuantumEyes.

6.1. Développement sur Replit

Le choix de Replit comme plateforme de développement et potentiellement de déploiement impose certaines contraintes techniques et architecturales. Il faudra s'assurer que Replit offre les capacités nécessaires en termes de performance, de scalabilité, de sécurité et de gestion des données pour une solution de type NDR. Les

limitations éventuelles de la plateforme devront être identifiées et des solutions de contournement trouvées si nécessaire.

6.2. Budget et Délais

Bien que non spécifiés en détail dans les documents fournis pour cette phase, les contraintes budgétaires et les délais de mise sur le marché seront des facteurs importants. Un MVP (Minimum Viable Product) devra être défini pour permettre une commercialisation rapide, suivi d'itérations pour ajouter des fonctionnalités plus avancées. La roadmap technologique initiale mentionne un PoC NDR ML sur un cas d'usage identifié, avec une durée estimée d'au moins 3 mois après une phase de modélisation.

6.3. Expertise en QML

Le développement de la brique QML nécessitera une expertise pointue, potentiellement via des partenariats avec des laboratoires de recherche ou des experts du domaine. La maturité actuelle des ordinateurs quantiques est également une contrainte : la solution doit être "quantum-ready" mais fonctionner efficacement avec du ML classique en attendant.

6.4. Adoption par le Marché Cible

La solution doit être pensée pour les PME et les collectivités, ce qui implique des contraintes en termes de coût, de complexité d'installation et de gestion. L'approche "one-stop-shop" est un atout mais doit être implémentée de manière cohérente et simple pour l'utilisateur final.

7. Livrables Attendus

Les livrables attendus pour le projet de développement de QuantumEyes incluent, sans s'y limiter :

- La plateforme QuantumEyes fonctionnelle, accessible en mode SaaS, avec toutes les fonctionnalités décrites dans ce cahier des charges (selon la roadmap et les phases de développement).
- Les sondes logicielles (VM) à déployer chez les clients.
- La documentation technique complète (architecture, code, API, etc.).
- La documentation utilisateur (guides d'installation, manuels d'utilisation, FAQ).
- Les rapports de tests (unitaires, intégration, performance, sécurité).
- Le code source complet de la solution.
- Un plan de déploiement et de maintenance.

8. Stratégie de Développement et de Déploiement

Une approche agile est recommandée pour le développement de QuantumEyes, avec des cycles itératifs permettant de livrer de la valeur rapidement et d'adapter le produit en fonction des retours utilisateurs et de l'évolution des technologies. La définition d'un MVP est cruciale. Le développement sur Replit facilitera la collaboration et le déploiement continu. La stratégie de déploiement devra assurer une haute disponibilité et une scalabilité. Des tests rigoureux à chaque étape du développement sont indispensables. La constitution d'une équipe technique pluridisciplinaire (cybersécurité, réseau, data science, ML, QML, développement front-end et back-end) sera nécessaire.

9. Conclusion

Le projet QuantumEyes représente une opportunité significative de développer une solution de cybersécurité innovante et souveraine, adaptée aux besoins spécifiques des PME et des collectivités. L'intégration de technologies avancées comme le ML et le QML, combinée à une approche complète (Évaluer, Protéger, Garantir), positionne QuantumEyes comme un acteur potentiellement majeur sur le marché du NDR. Ce cahier des charges fournit les bases pour le développement de cette solution ambitieuse. Un suivi rigoureux, une collaboration étroite entre les équipes et une adaptation continue seront les clés du succès de ce projet.

10. Références

- Phase1_DesignProduit_Vigilianz_08112021.pdf
- Vigilianz - Vision Produit.pdf
- Rapport Diagnostic DeepTech - QuantumEyes.pdf