Référentiel de certification

Exploiter la blockchain dans le développement d'applications

Référentiels d'activités et de compétences

REFERENTIEL D'ACTIVITES	RÉFÉRENTIEL DE CERTIFICATION	
COMPÉTENCES	MODALITES D'EVALUATION	CRITÈRES D'ÉVALUATION
C1. Réaliser la conception technique d'un *smart-contract** en formalisant, dans le respect des bonnes pratiques d'éco-conception, les modélisations et les spécifications techniques associées afin de répondre au besoin d'intégration de fonctionnalités blockchain* dans une application web.	**E1. Mise en situation professionnelle (C1, C2, C3, C4, C5)** Le projet évalué a pour but la conception et le développement d'un *smart-contract* puis de son intégration, et du développement des fonctionnalités associées, dans une applications web existante. Livrable : rapport professionnel individuel Évaluation : - Correction du rapport professionnel - Soutenance orale individuelle	Le périmètre des spécifications techniques est complet, il couvre l'ensemble des moyens techniques à mettre en œuvre pour le développement du smart-contract : rappel de l'objet et faisabilité, la structure de données, les variables d'état du *smart-contract*, les contrôles d'accès au contrat, l'optimisation du gaz*, le traitement des erreurs, le caractère évolutif du contrat, l'émission des événements. L'objectif et la fonctionnalité du contrat sont définis clairement. Son champ d'application est clairement rappelé. Les structures de données sont appropriées pour représenter les informations dans le contrat.

		Elles sont modélisées dans le respect d'un standard choisi (UML ou Merise par exemple). Les variables choisies pour représenter les différents états du contrat sont rappelées et définies dans les spécifications. Les implications des variables d'état sur les coûts du gaz* sont rappelées et évaluées dans les spécifications techniques. Des mécanismes de contrôle d'accès sont prévus et définis dans les spécifications pour limiter les interactions avec le contrat et exécuter des fonctions sensibles (autorisations d'accès basées sur la définition de rôles par exemple). Des mécanismes robustes de gestion des erreurs sont anticipés et définis dans les spécifications. La stratégie de test et les caractéristiques de l'environnement de test sont rappelés dans les spécifications techniques du contrat. Les spécifications techniques précisent si le contrat doit être évolutif dans le temps, puis la complexité et les considérations de sécurité associées sont rappelées et définies.

		Les événements des actions contractuelles à émettre pour les développeurs et les utilisateurs sont rappelés et définis.
C2. Développer un *smart-contract* conforme aux modélisations et aux spécifications techniques validées en respectant les standards de sécurité informatique (SCSVS*) en vigueur afin d'assurer son bon fonctionnement dans la blockchain		L'environnement de développement du *smart-contract* est configuré en accord avec les spécifications techniques. Toutes les fonctionnalités du smart-contract spécifiées sont implémentées. Les développements respectent les principes de sécurisation du SCSVS*. Le code du contrat est optimisé afin de minimiser les coûts de gaz* pour les utilisateurs. - Des *modifiers** pour réduire la duplication du code sont utilisées quand c'est possible. - Des instructions "*require*"*, pour valider les entrées et l'état du *smart-contract*, sont utilisées quand c'est possible. Les émissions des événements prévus par les spécifications sont implémentés. Les événements sont correctement émis, dans le respect des spécifications techniques. La procédure d'installation de l'environnement

		de développement est documentée. La procédure d'exécution du smart-contract est documentée. L'architecture du *smart-contract* et les règles implémentées sont documentées. Un outil de *linting** est configuré et opérationnel en environnement de développement. Le build* du smart contract* s'exécute sans erreur.
C3. Livrer un *smart-contrat* sur une blockchain cible, en définissant une procédure automatisée ou manuelle de déploiement, afin de rendre le programme disponible aux utilisateurs et aux applications tierces.		Les outils de déploiement (*plugin*, IDE, scripts...) du *smart-contract* sont configurés dans le respect des spécifications techniques. Le coût en gaz* pour le déploiement est connu et validé en amont. La procédure de déploiement est documentée et permet de reproduire la livraison en environnement de test. Les secrets (clés, *tokens*, mots de passe...) sont gérés comme tels dans la procédure, ne sont ni versionnés ni partagés manuellement ou automatiquement.

		La procédure de déploiement du *smart-contract* est validée en testant une livraison sur l'environnement de test.
C4. Interfacer une application web existante à une blockchain et un *smart-contract* à l'aide de bibliothèques spécialisées*, dans le respect des standards de sécurité (OWASP*), de gestion des données personnelles (RGPD) et d'accessibilité numérique en vigueur (RGAA*), afin de répondre au besoin fonctionnel et technique modélisé et spécifié.		L'application web servant comme point de départ au projet est correctement installée en environnement de développement et fonctionne. Les dépendances, outils et bibliothèques logicielles pour le développement de l'interaction avec le *smart-contract* sont installés dans l'environnement de développement de l'application. La connexion à la blockchain est effective depuis l'application. L'ensemble des points d'interaction avec le *smart-contract* sont intégrés à l'application, en utilisant une bibliothèque dédiée. L'intégration du *smart-contract* à l'application est faite dans le respect des spécifications fonctionnelles et techniques. Les nouveaux composants d'interface ou les mises à jour des composants existants sont intégrés dans le respect des spécifications

		données. Les développements réalisés sur les composants d'interface respectent les exigences d'accessibilité définies dans les spécifications. Dans le cas où l'application traite des données personnelles : - Le registre des traitements de données personnelles intègre l'ensemble des traitements de données personnelles impliqués dans le projet. - Les procédures de tri de données personnelles pour la mise en conformité de l'application avec le RGPD sont rédigées. - Les procédures de tri détaillent les traitements de conformité (automatisés ou non) à appliquer ainsi que leur fréquence d'exécution.
C5. Implémenter les tests du *smart-contract* et les tests fonctionnels associés à l'interaction entre la blockchain et l'application, à l'aide de bibliothèques de tests, afin de détecter et traiter les dysfonctionnements		Tous les critères d'acceptation définis dans le *backlog** (ou cahier des charges) du projet sont couverts par des tests automatiques. Un environnement de test du *smart-contract* est proposé, il est configuré et en état de

techniques et logiques relevés par leur exécution.		marche. L'environnement de test est connecté à une blockchain de test. Des tests automatiques sont implémentés. L'exécution des tests se fait sans erreur. Les rapports de tests sont interprétés correctement. La procédure d'installation de l'environnement de test est documentée. La procédure d'exécution des tests est documentée.

Glossaire

- **Bibliothèques spécialisées :** une bibliothèque correspond à une brique logicielle préconçue facilitant l'implémentation de fonctionnalités dans un logiciel.

- **Blockchain** : c'est un registre numérique, distribué et sécurisé, de toutes transactions effectuées depuis le démarrage du système.

- **Build :** La compilation consiste à transformer le code source en code objet. L'édition de liens ("linking") est l'action de combiner le code objet avec les bibliothèques en un exécutable brut. La construction (ou le "build") est la séquence composée de la compilation et du "linking", avec éventuellement d'autres tâches telles que la création d'un programme d'installation. "Un build" correspond à l'objet en sortie de la phase de "build".

- **Gaz :** Toutes les transactions sur la blockchain sont associées à des frais de gaz appelés "gas fees" en anglais. Ces frais sont prévus pour éviter le gaspillage de la puissance de calcul des acteurs du réseau et rémunérer les mineurs.

- **Lint / Linting / Linter** : "Lint" est le terme informatique désignant un outil d'analyse statique du code utilisé pour repérer les erreurs de programmation, les bogues, les erreurs de style et les constructions suspectes. Le terme provient d'un utilitaire Unix qui examinait le code source du langage C. Un programme qui remplit cette fonction est également connu sous le nom de "linter".

- **Modifiers :** Le comportement d'une fonction peut être modifié à l'aide de modifiers (en anglais) ou "modificateurs" de fonction. Les "modificateurs" de fonction peuvent être utilisés pour vérifier automatiquement une condition avant d'exécuter la fonction. Ils peuvent être créés pour de nombreux cas d'utilisation différents : vérification d'autorisation d'accès, validation de données fournies par un utilisateur, etc.

- **OWASP :** Open Worldwide Application Security Project est une communauté en ligne qui produit des articles, des méthodologies, de la documentation, des outils et des technologies librement accessibles dans le domaine de la sécurité

des applications web.

- **RGAA :** Référentiel général d'amélioration de l'accessibilité, définit une méthode technique et propose un cadre opérationnel de vérification de la conformité aux exigences d'accessibilité.

- **SCSVS :** Smart Contract Security Verification Standard correspond à un standard de sécurité créé pour normaliser la sécurité des contrats intelligents pour les développeurs, les architectes et les auditeurs sécurité.

- **Smart-contract :** Il s'agit d'un contrat reposant sur un code informatique, dont l'exécution répond à des conditions prédéfinies. Hébergé sur un réseau décentralisé comme une blockchain, il est autonome et accessible à tous.