TRACK B — EVM Engineer (Solidity/Foundry)

Détail exécutable (PDF séparé)

RBK 2.0

Contents

1 Résumé exécutif

Positionnement. Ce track vise un **EVM Engineer** capable de livrer des smart contracts Solidity **production-ready** avec une discipline de tests **professionnelle** (unit, fuzz, invariants), une approche **security-first**, la maîtrise des standards (ERC) et des patterns industriels (déploiement, verification, upgrades, monitoring).

Promesse mesurable.

- Capacité à produire des contracts avec **Foundry pipeline complet** (unit → fuzz → invariants → coverage).
- Capacité à livrer un projet intégrant **verification on-chain**, scripts de déploiement reproductibles et **plan d'upgrade** (UUPS).
- Capacité à rédiger un **mini audit report** et à corriger des findings.

Non négociables (track B)

- Tests fuzz et invariants sur les composants critiques.
- Threat model + checklist sécurité.
- Scripts de déploiement et verification reproductibles.

2 Objectifs mesurables (preuves)

3 Programme (12 semaines : modules 1→6)

Domaine	Compétence	Preuve + outil (seuil)
Solidity core	Storage/memory/calldata, errors, events, libs	Repo “Vault/Escrow” + tests + doc API
Testing pro	unit + fuzz + invariants	Foundry pipeline + rapports ; invariants sur modules critiques
Standards	ERC-20/721 (+extensions)	Contrat standard + RBAC + scripts deploy/verify
Upgrades	UUPS + gouvernance d’upgrade	Plan d’upgrade + tests + “upgrade safety checklist”
Sécurité	vuln classes + mitigations	Mini audit report (min 8 findings) + correctifs testés
Production	deploy manifest + monitoring plan	address book + runbook + events/metrics map

Table 1: Objectifs mesurables Track B

Semaine	Module	Objectifs	Lab / livrable	DoD
S9	M1	Solidity deep dive + state machine	Vault v0	unit tests + doc
S10	M1	permissions + erreurs + events	Escrow	tests négatifs + invariants
S11	M2	Foundry env pro	Pipeline CI + coverage	CI verte + badges
S12	M2	fuzz/invariants	Fuzz harness sur vault	invariants + report
S13	M3	ERC standards	ERC-20/721 + RBAC	deploy scripts + verify
S14	M3	composabilité	module composable	tests + docs
S15	M4	dApp integration	front minimal + tx UX	demo + error taxonomy
S16	M4	events + indexing	indexer simple	logs + docs
S17	M5	L2/patterns + gas	gas budget + optim	bench + justification
S18	M5	upgrades UUPS	UUPS + upgrade plan	tests upgrades
S19	M6	security hardening	audit checklist + fix	mini audit report
S20	M6	capstone release	capstone final	PRR + verification + demo

Table 2: Programme Track B (12 semaines)

4 Labs détaillés (extraits studio-grade)

Lab Vault (Module 1) — Spec & invariants

Fonctionnalité	Invariants	Tests attendus
deposit	balance augmente, event émis	unit tests + fuzz sur montants
withdraw	ne dépasse pas balance	tests négatifs + invariant “never negative”
roles	seuls rôles autorisés	tests access control

Table 3: Vault spec (résumé)

Lab UUPS Upgrade (Module 5) — Safety gate

Axe	Poids	Critères
Sécurité (threat model + findings)	30	vuln classes couvertes + correctifs testés
Tests (unit+fuzz+invariants)	25	invariants critiques + coverage utile
Standards + composabilité	15	ERC correct + RBAC + scripts
Upgrades + production	15	UUPS safe + PRR + deploy manifest
Docs + demo	15	README + API + demo reproductible

Table 4: Rubrique Track B

5 Rubrique standard audit EVM (100 points)

6 Stack Foundry + outils

Catégorie	Outils	Artefact attendu
Dev	Foundry (forge/cast/anvil)	scripts + pipeline tests
Qualité	lint/format + CI	badges + rapports
Sécurité	analyse statique (option) + checklist	mini audit report
Deploy	scripts + verification	address book + verify links
Monitoring	events map + runbook	plan monitoring + alerting

Table 5: Stack Track B

7 Tables/Figures indispensables

Table — Foundry pipeline

Étape	Objectif / preuve
Unit tests	logique nominale + edge cases
Fuzz tests	robustesse sur large espace d'entrées
Invariant tests	propriétés globales (safety)
Coverage	visibilité sur chemins critiques
Report	rapport CI + conclusions

Table 6: Foundry pipeline : unit → fuzz → invariants → coverage

Table — Vuln classes → tests → mitigations

Figure — UUPS upgrade flow

Figure — From code to mainnet (verification & monitoring)

Classe	Test	Mitigation
Auth flaws	tests rôles + négatifs	least privilege + modifiers
Reentrancy	scénario appel externe	checks-effects-interactions / guards
DoS	tests limites + gas	bounded loops + pull pattern
Oracle misuse	tests data stale	validations + fallback
Upgrade risks	tests migration	storage discipline + governance

Table 7: Vuln classes → tests → mitigations (défensif)

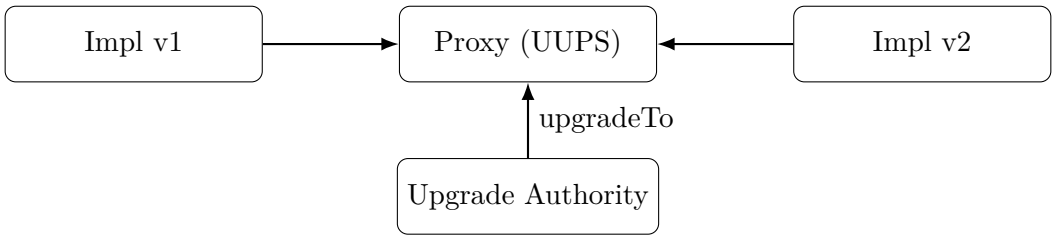

Figure 1: UUPS upgrade flow (vue simplifiée)

Figure 2: Chaîne industrielle : code → tests → deploy → verify → monitoring