guidelines-advisor

guidelines-advisor

> *Catalogue généré le 2026-05-11*

En une phrase

Conseille un projet de smart contracts (surtout Solidity) selon les "Development Guidelines" Trail of Bits : génère la doc, revoit l'architecture, audite les patterns d'upgrade et proxy, évalue les tests et les dépendances.

Quand l'utiliser

- En **début de projet** pour cadrer l'architecture avant d'avoir trop de dette technique.
- Quand tu hérites d'un codebase blockchain et que tu veux une vue d'ensemble structurée.
- Pour générer la documentation manquante : description en français clair, diagrammes d'architecture, NatSpec (le format de commentaires standardisé Solidity).
- Quand le projet utilise des **proxies upgradeables** (un pattern complexe et dangereux où le contrat peut être modifié après déploiement) et que tu veux faire vérifier la conformité.
- Pour challenger les choix on-chain vs off-chain (= "qu'est-ce qui doit vraiment vivre sur la blockchain et qu'est-ce qu'on peut faire ailleurs ?").

Comment l'invoquer

- **Slash command** : `/guidelines-advisor`
- **Phrases déclencheurs (texte)** : "review my smart contract architecture" / "check best practices" / "audit upgradeability"

- **Auto-invocation** : Sur demande explicite (souvent au début d'un projet ou lors d'une refonte).

Description détaillée

Trail of Bits maintient un guide "Building Secure Contracts — Development Guidelines" qui condense des années d'audits en bonnes pratiques. Ce skill applique ce guide à ton projet de façon interactive et te livre des recommandations actionnables.

Il couvre 11 zones d'analyse : 1) **Documentation & Specifications** (descriptions plain-English, diagrammes, NatSpec), 2) **On-Chain vs Off-Chain Computation** (quoi mettre où), 3) **Upgradeability** (faut-il vraiment de l'upgradeable ? migration vs upgrade, séparation données/logique), 4) **Delegatecall Proxy Pattern** (cohérence du storage layout, init, function shadowing — sources classiques de bugs critiques), 5) **Function Composition** (taille, clarté, modularité), 6) **Inheritance** (profondeur, diamond problem), 7) **Events Logging**, 8) **Common Pitfalls**, 9) **Dependencies**, 10) **Testing Coverage** et 11) techniques de test avancées.

Workflow en 5 phases : Discovery → Documentation Generation (génère ce qui manque) → Architecture Analysis → Implementation Review → Recommandations priorisées. Pour Solidity, il s'appuie sur les "Slither printers" (les générateurs de diagrammes intégrés à Slither, l'analyseur statique Trail of Bits).

C'est un skill **stratégique** plutôt qu'un scanner : il ne te dit pas "ligne 42, bug X", il te dit "votre architecture a tel risque structurel, voici comment y remédier".

Pour aller plus loin

Pour la liste détaillée des critères et les checks par zone, consulter le SKILL.md original à `/home/thymon/.claude/plugins/cache/trailofbits/building-secure-contracts/1.0.1/skills/guidelines-advisor/SKILL.md` et la ressource `resources/ASSESSMENT_AREAS.md`.

Source

- **Plugin** : `trailofbits/building-secure-contracts`
- **Nom interne** : `guidelines-advisor`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/building-secure-contracts/1.0.1/skills/guidelines-advisor/SKILL.md`

Revision #2
Created 2026-05-11 21:19:35 UTC by thymon
Updated 2026-05-11 21:37:12 UTC by thymon