spec-to-code-compliance

spec-to-code-compliance

> *Catalogue généré le 2026-05-11*

En une phrase

Compare un document de spécification (le « cahier des charges » d'un projet, type whitepaper) au code réellement écrit, pour vérifier que le code fait exactement ce que la doc promet — ni plus, ni moins.

Quand l'utiliser

- Tu audites un smart contract et tu disposes du whitepaper (le document fondateur d'un projet blockchain qui décrit comment il est censé fonctionner).
- Tu veux trouver les « trous » entre ce qui est promis sur le papier et ce qui est codé.
- Tu cherches du code qui fait des choses non documentées (potentiellement un backdoor ou une fonctionnalité oubliée).
- Tu veux valider qu'une implémentation respecte un protocole standard.

Comment l'invoquer

- **Slash command** : `/spec-to-code-compliance` (ou `/spec-compliance`).
- **Phrases déclencheurs (texte)** : "does this code match the spec?", "what's missing from the implementation?", "compliance check", "compare to whitepaper".
- **Auto-invocation** : Sur demande explicite.

Description détaillée

Dans le monde de la blockchain et des protocoles décentralisés, un projet commence presque toujours par un « whitepaper » : un document qui explique en détail ce que le système est censé faire (les règles, les calculs, les garanties de sécurité, les invariants à respecter). Puis des développeurs traduisent ce document en code. Le problème, c'est que la traduction n'est jamais parfaite : il y a des oublis, des interprétations, des raccourcis. Ce skill traque ces écarts.

Il fonctionne en plusieurs phases. D'abord, il identifie tous les documents qui font office de spécification (whitepaper PDF, Markdown, notes de design). Ensuite, il extrait de manière exhaustive chaque exigence ou affirmation de la doc. Puis il cherche dans le code la portion qui implémente cette exigence. Pour chaque correspondance, il attribue un type (match complet, match partiel, divergence, exigence non implémentée) et un niveau de confiance entre 0 et 1. Il signale aussi le sens inverse : du code qui existe mais qui ne correspond à aucune ligne de la spec — c'est souvent là que se cachent les vulnérabilités les plus retorses.

La règle d'or du skill : ne jamais inventer. Chaque conclusion doit citer une preuve précise (section du document, numéro de ligne du code). Pas de supposition, pas de « probablement ». Quand quelque chose est ambigu, c'est classé comme tel plutôt que tranché à la légère.

Pour aller plus loin

Pour les détails techniques, exemples et patterns spécifiques, voir le SKILL.md original.

Source

- **Plugin** : `trailofbits/spec-to-code-compliance`
- **Nom interne** : `spec-to-code-compliance`
- **Fichier** : `/home/thymon/.claude/plugins/cache/trailofbits/spec-to-code-compliance/1.1.0/skills/spec-to-code-compliance/SKILL.md`

Revision #2
Created 2026-05-11 21:19:27 UTC by thymon
Updated 2026-05-11 21:37:04 UTC by thymon