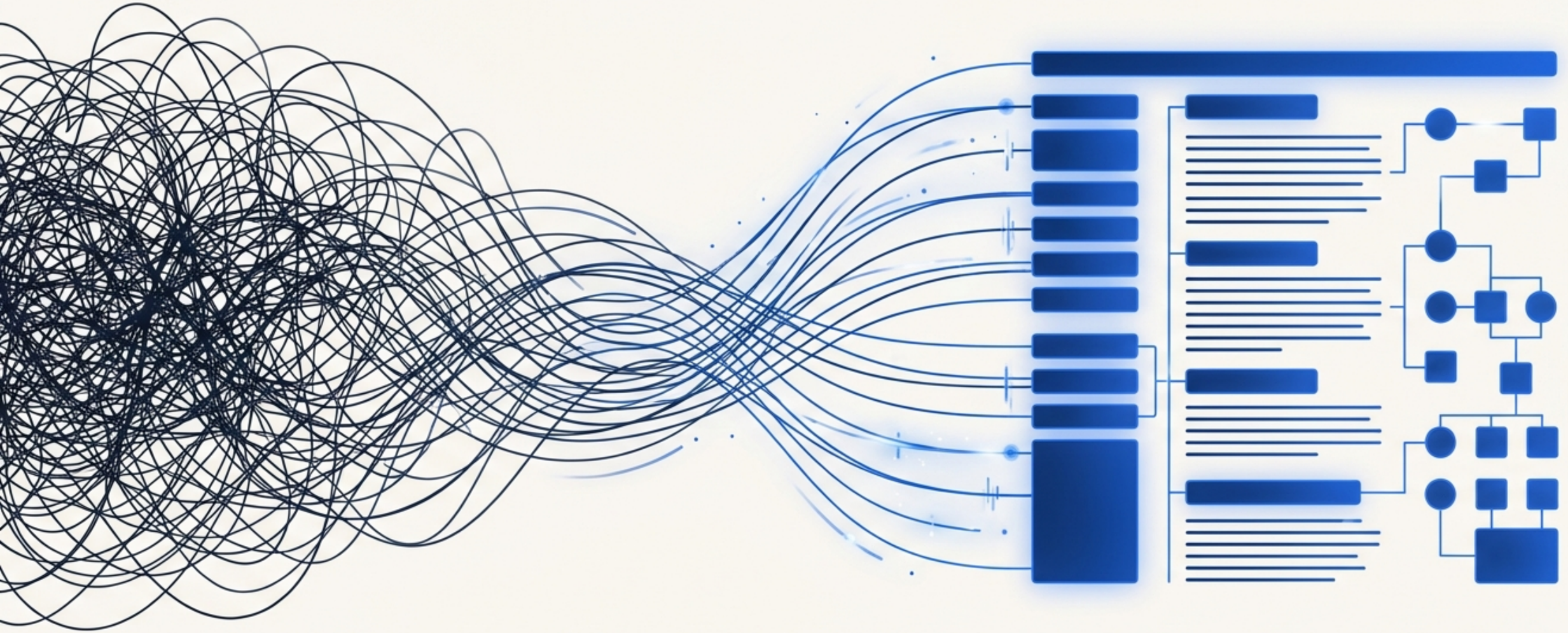
Code Wiki



La documentation qui comprend enfin votre code.

Le plus grand obstacle au développement logiciel n'est pas d'écrire du code.

« Lire du code est lent et coûteux. »

- 1. La plupart des équipes de développement perdent des heures à déchiffrer des modules existants ("legacy").
- 2. La documentation est le plus souvent obsolète, trompeuse ou inexistante.
- 3. Le coût de cette friction est énorme : productivité réduite, intégration des nouveaux membres ralentie, et risque d'erreurs accru.

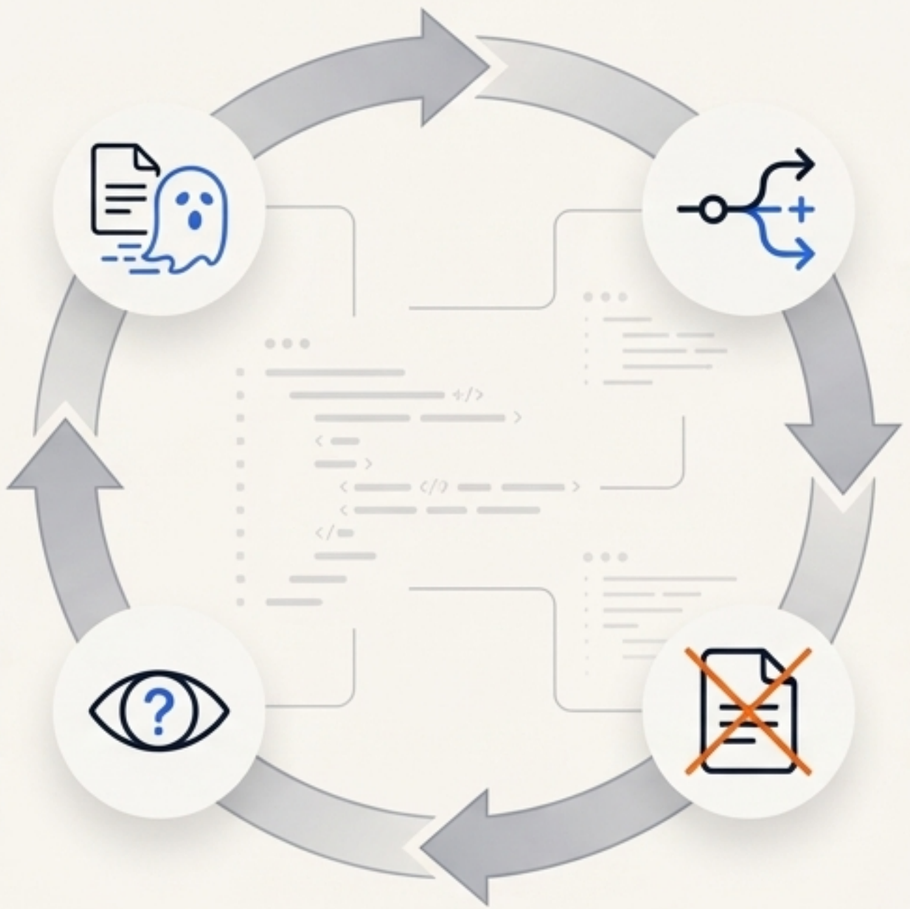
Le cycle infernal de la documentation manuelle

1. Le code évolue : De nouvelles fonctionnalités sont ajoutées, le code est refactorisé.

2. La doc n'est pas mise à jour : Les fichiers `README.md` et les diagrammes statiques sont oubliés.

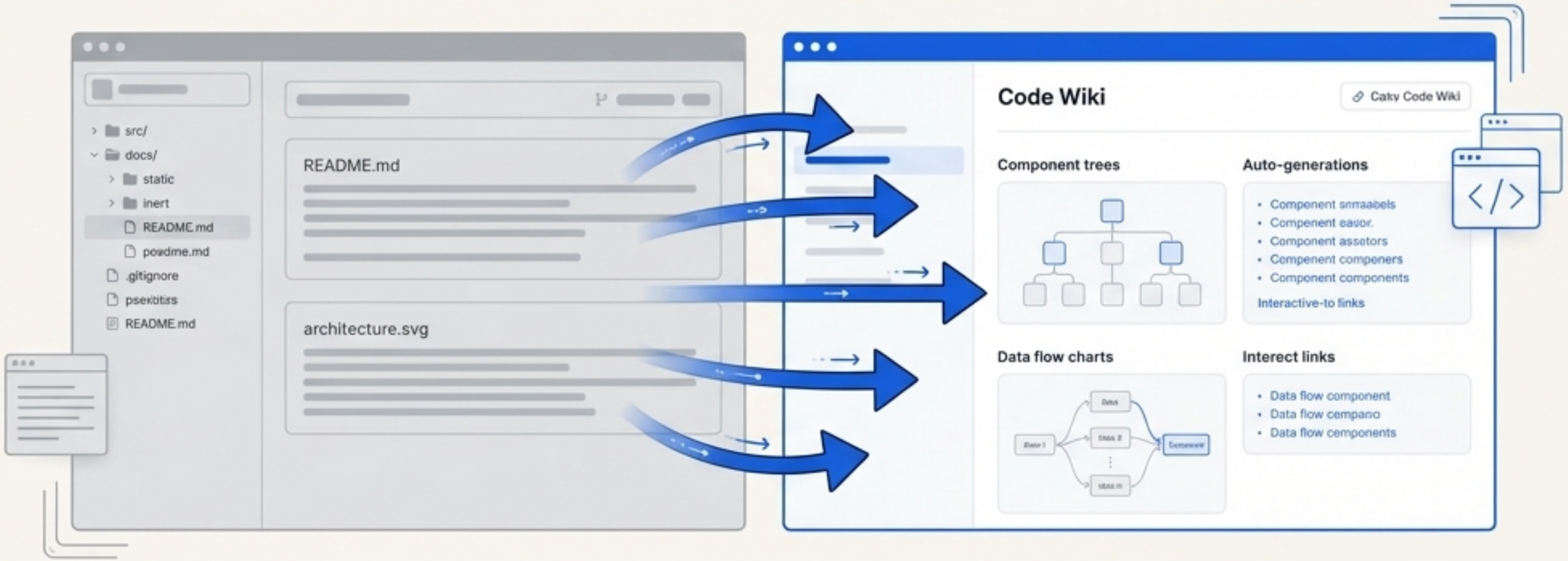
3. Perte de confiance : Les développeurs apprennent à se méfier de la documentation, car elle est fausse.

4. Abandon : Personne ne veut plus investir de temps dans une documentation que personne ne lit ou ne croit.



Un système défaillant par nature.

Et si votre documentation pouvait vivre et évoluer avec votre code ?



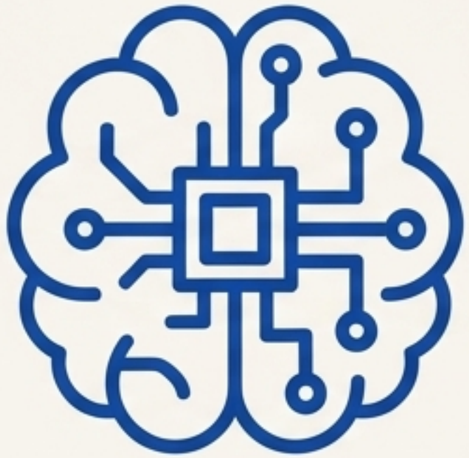
Voici Code Wiki.

Fini les fichiers Markdown statiques. Fini les diagrammes désuets. Code Wiki transforme la documentation statique en une ressource vivante et toujours précise.

Une approche basée sur trois principes fondamentaux.

Automatisée et Toujours à Jour

La documentation est regénérée à chaque modification du code.



Intelligente et Contextuelle

Une IA Gemini qui comprend la logique de votre dépôt de A à Z.

Intégrée et Actionnable

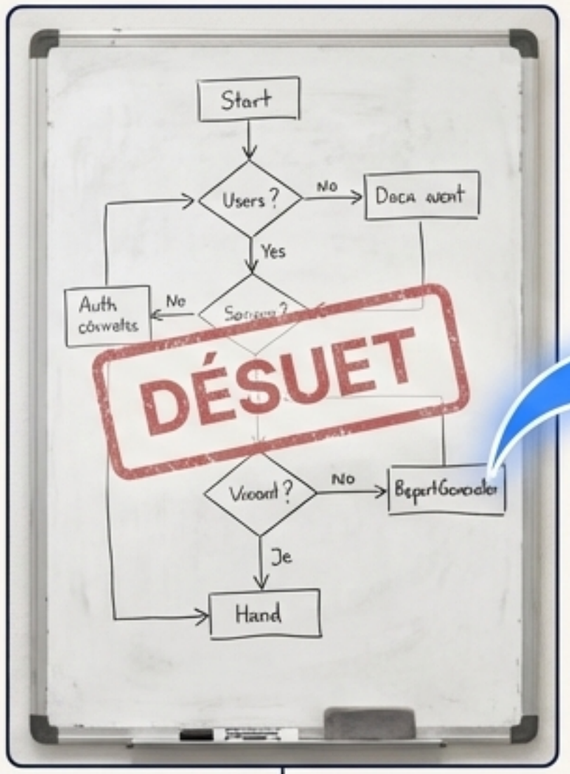
Des hyperliens profonds qui connectent directement la doc au code.

1. Une documentation qui ne ment jamais.

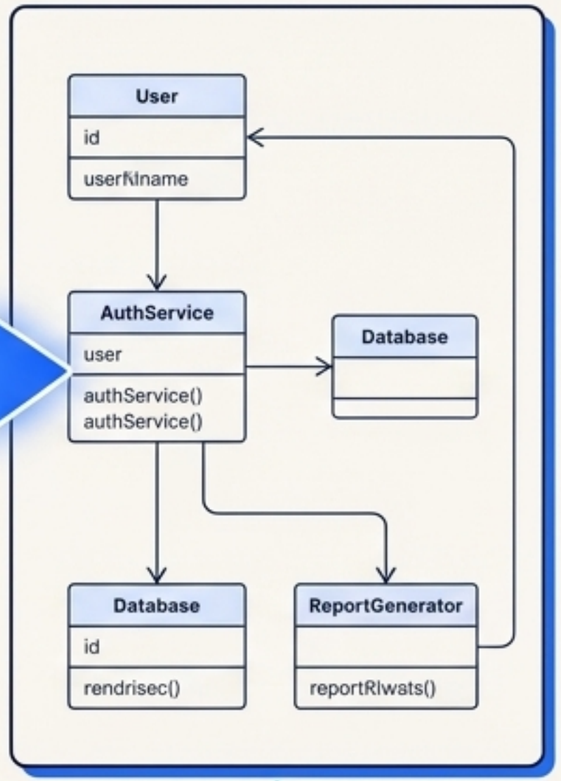
- **Analyse complète :** Code Wiki scanne l'intégralité de votre base de code.
- **Régénération continue :** À chaque `commit`, la documentation est reconstruite pour refléter précisément les changements.
- **Diagrammes vivants :** Génère automatiquement des diagrammes d'architecture, de classe et de séquence qui représentent l'état *actuel* du code.

« Les visuels ne mentent jamais. »

Avant



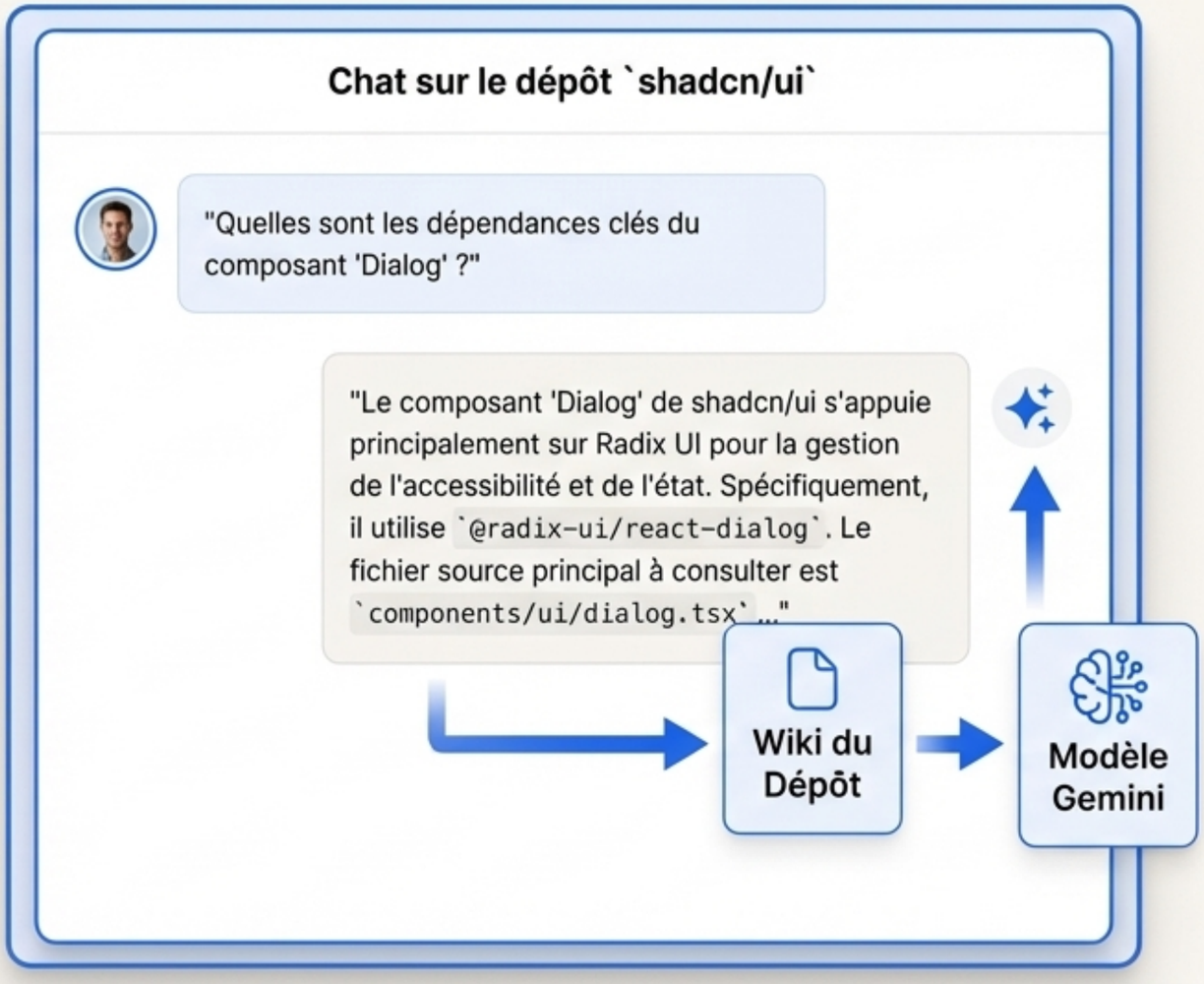
Après

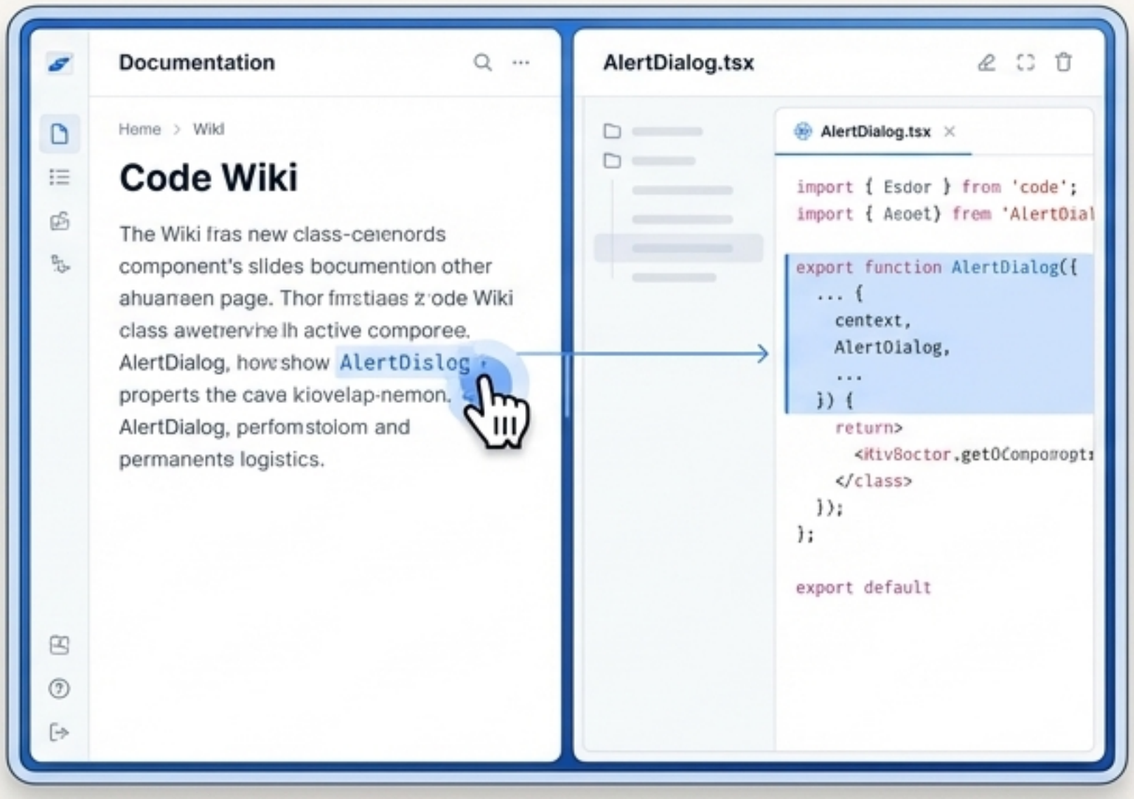


2. Un expert IA intégré, entraîné sur votre code.

Le wiki devient le 'cerveau' du chat Gemini.

1. Lorsque vous posez une question, vous ne parlez pas à une IA généraliste.
2. Vous discutez avec un modèle dont l'unique source de connaissance est le wiki à jour de votre dépôt.
3. **Résultat** : Des réponses pertinentes, précises et contextuelles.





3. Un flux de travail continu entre la lecture et l'exploration.

Chaque section est truffée d'hyperliens profonds.

- Cliquez sur une classe, une fonction ou un fichier mentionné dans la documentation pour y accéder instantanément dans le code.
- Fini la recherche manuelle dans l’arborescence des fichiers.
- Lire la documentation et explorer le code deviennent une seule et même action.

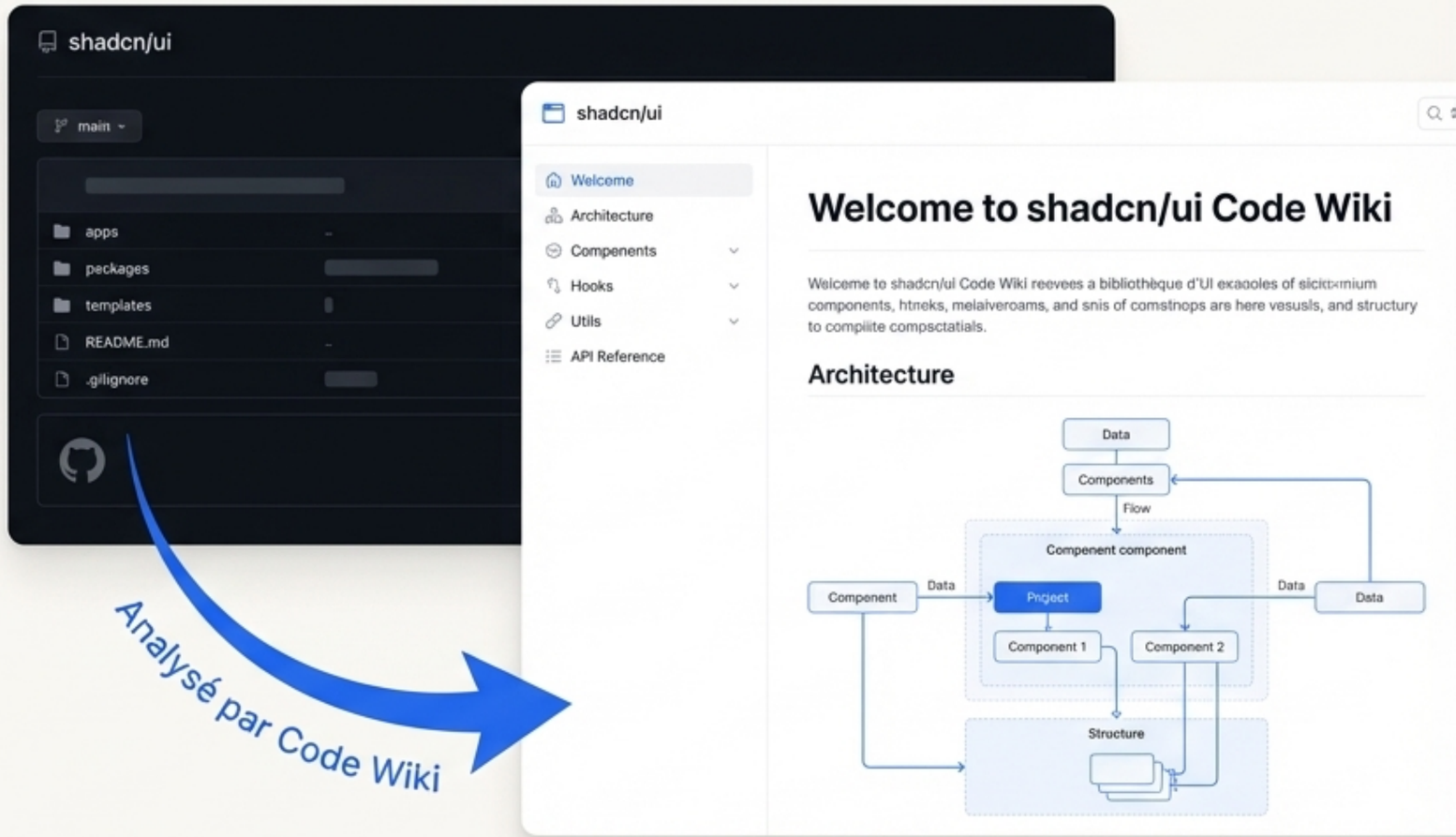
Assez de théorie. Voyons Code Wiki en action.

Étude de Cas

`shadcn/ui`

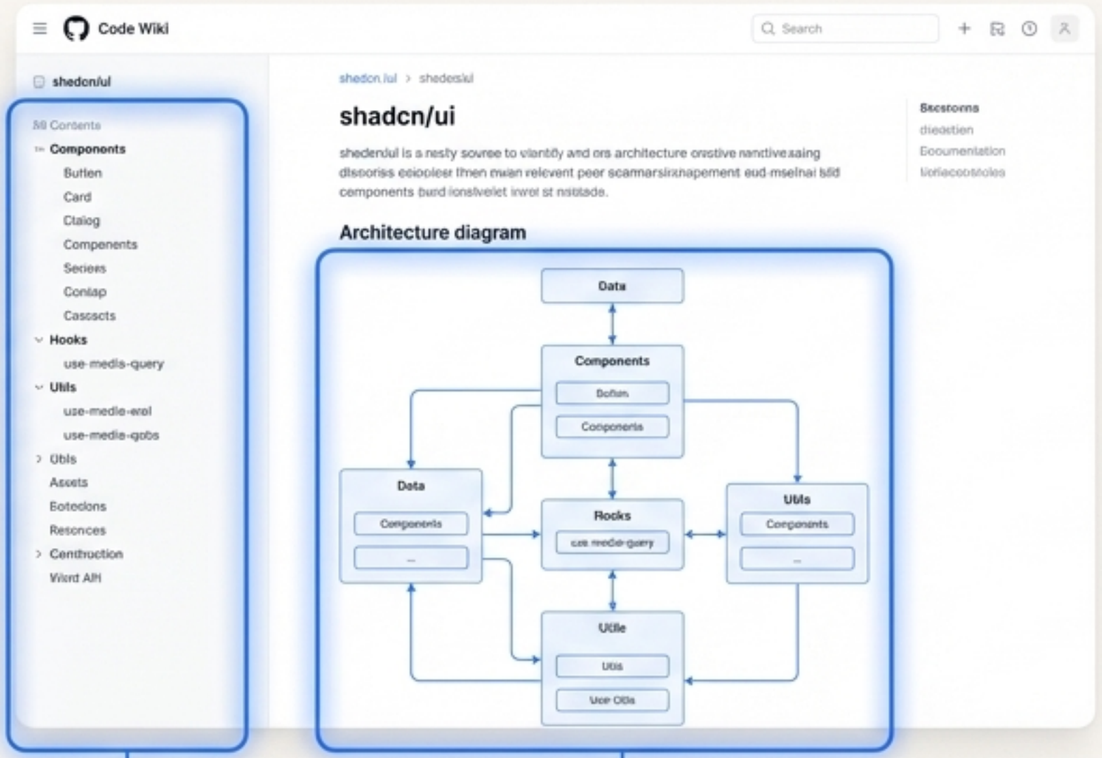
Pourquoi cet exemple ?

- L'une des bibliothèques d'UI les plus populaires.
- Un excellent cas d'usage pour démontrer la capacité de Code Wiki à analyser et structurer un projet complexe.



Résultat : Une structure limpide, générée en quelques instants.

- Génération d'une documentation propre et parfaitement organisée pour chaque partie du projet.
- Décomposition logique du projet, section par section, rendant la navigation intuitive.
- Chaque section est accompagnée de son propre diagramme généré automatiquement, offrant une compréhension visuelle immédiate des interactions.



Le chatbot en action : Des réponses qui vont au-delà de la simple documentation.

"Comment fonctionne le composant Button ?"

Explication du code

Le composant `Button` est construit sur la primitive `Slot` pour une composition flexible. Il utilise `cva` (Class Variance Authority) pour gérer ses variantes (`variant`) et tailles (`size`). Le code source principal est <a href='#' class='text-blue-600 underline components/ui/button.tsx.

Exemples d'utilisation

```
// Default Button
<Button>Default</Button>

// Destructive Variant
<Button variant="destructive">Destructive</Button>

// With Icon
<Button><Mail className="mr-2 h-4 w-4" /> Login</Button>
```

Contexte d'intégration

Pour l'utiliser comme un lien avec Next.js, combinez-le avec le composant `<Link>` en utilisant la prop `asChild` pour une sémantique et une accessibilité correctes : `<Button asChild><Link href="/login">Login</Link></Button>`.

Editorial Tech

Allez plus loin : Utilisez Code Wiki pour coacher votre codeur IA.

Satoshi #1A202C

Scénario d'Usage: Générer un fichier de contexte pour des outils comme GitHub Copilot.

Résultat: Votre codeur IA comprend la structure, les règles et les conventions du projet, générant un code de bien meilleure qualité et plus cohérent.

Crée un fichier copilot-instructions.markdown qui résume les standards d'architecture et les patterns que Copilot doit respecter dans ce projet.

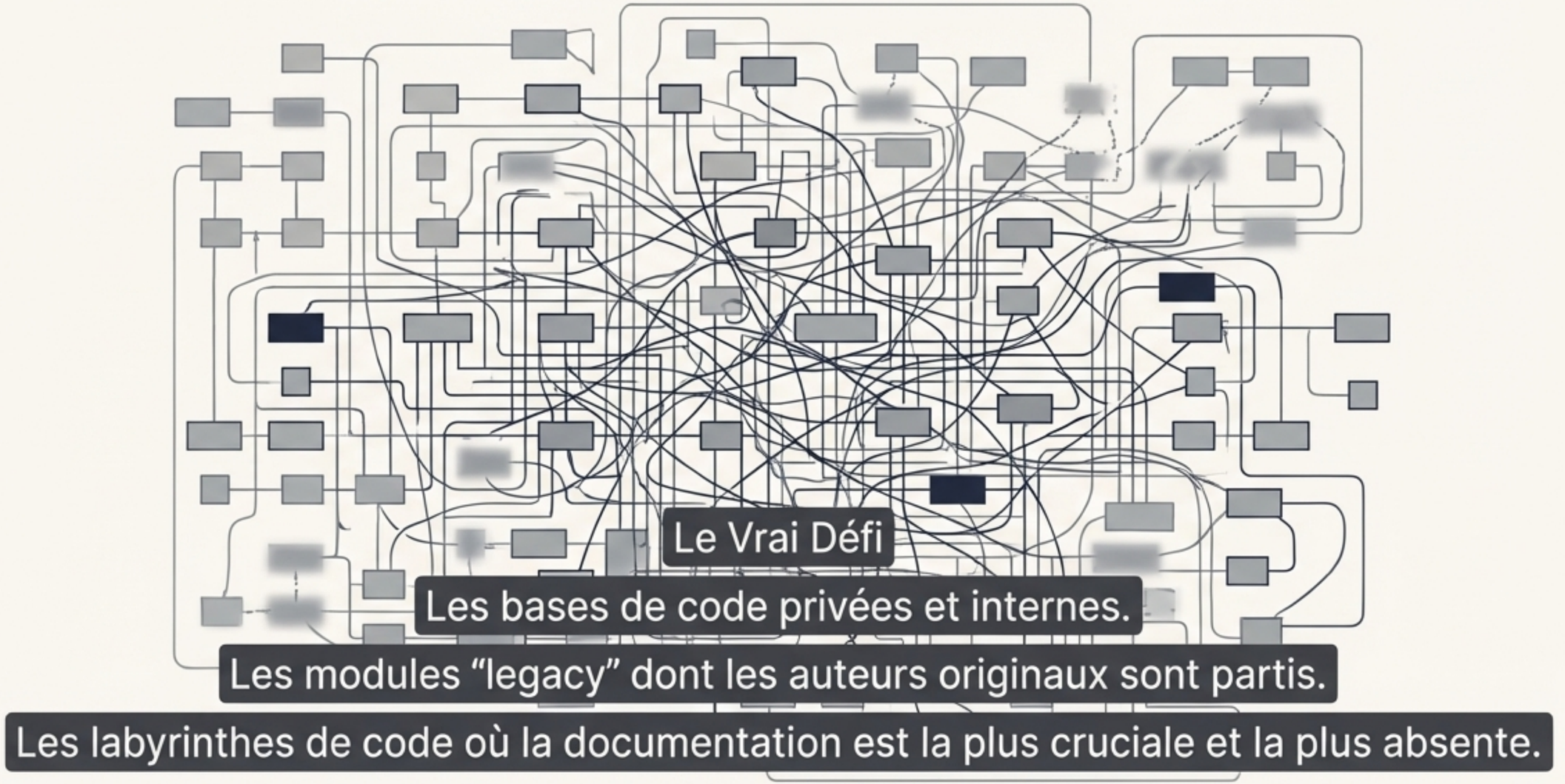
copilot-instructions.markdown

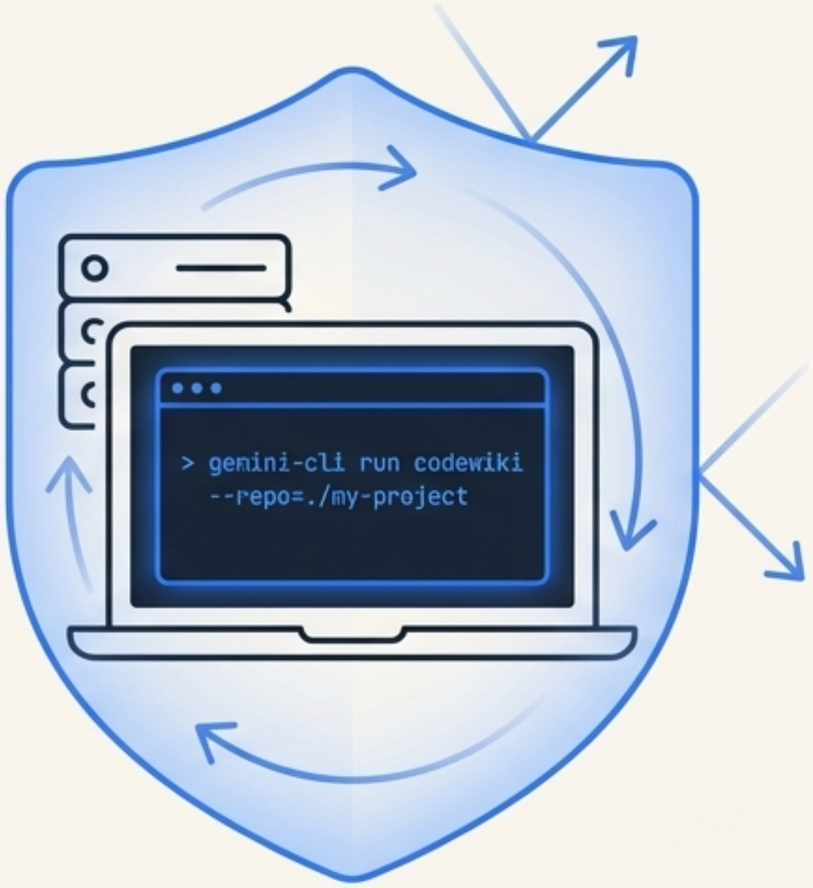
```
# GitHub Copilot Instructions for shadcn/ui

## Core Principles
- **Composition over configuration:** Components should be built using Radix UI primitives and be highly composable.
- **Accessibility First:* All components must follow WAI-ARIA standards.
- **Styling with Tailwind CSS:** Use utility classes for styling. Do not write custom CSS.

## Key Patterns
  - Use `cva` for all component variants.
  - Props should be passed down using the `...props` spread operator.
  - Always include support for the `asChild` prop for polymorphism.
```

Les dépôts publics sont un bon début.
Mais la vraie complexité se trouve ailleurs.





Prochainement : La puissance de Code Wiki, en local et en toute sécurité.

L'extension **Gemini CLI pour Code Wiki.**

Local : Exécutez Code Wiki directement sur vos machines, dans votre infrastructure.

Sécurisé : Votre code ne quitte jamais votre environnement. Aucune donnée n'est envoyée à l'extérieur.

Intégral : Apportez une véritable compréhension et une documentation vivante aux projets qui en ont le plus besoin.

« Un véritable 'game-changer' pour le développement en entreprise. »

Prêt à transformer votre documentation ?

Essayez-le Maintenant

Analysez vos dépôts publics dès aujourd'hui.

Visiter codewiki.dev

Préparez l'Avenir

Soyez le premier à accéder à l'outil qui va changer la donne pour votre équipe.

Rejoindre la liste d'attente pour le CLI