

Kombai : L'IA qui répare enfin le développement Frontend.

Au-delà du chatbot : l'ère de l'agent spécialisé qui comprend votre stack, votre navigateur et votre design system.



Le problème des LLM génériques : Le syndrome du "Copier-Coller".

LLM GÉNÉRIQUE (CLAUDE/GPT)

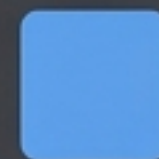
```
<button style="background:
blue; padding: 10px;
color: white;">
Click Me
</button>
```

Ignore le
Design System

RÉSULTAT DANS LE REPO

```
src/
├── components/
│   ├── legacy/
│   └── v2/
│       └── temp_button.js
├── styles.css
├── styles_v2.css
└── button_fix.css
```

Button



Button

Dette
technique
immédiate

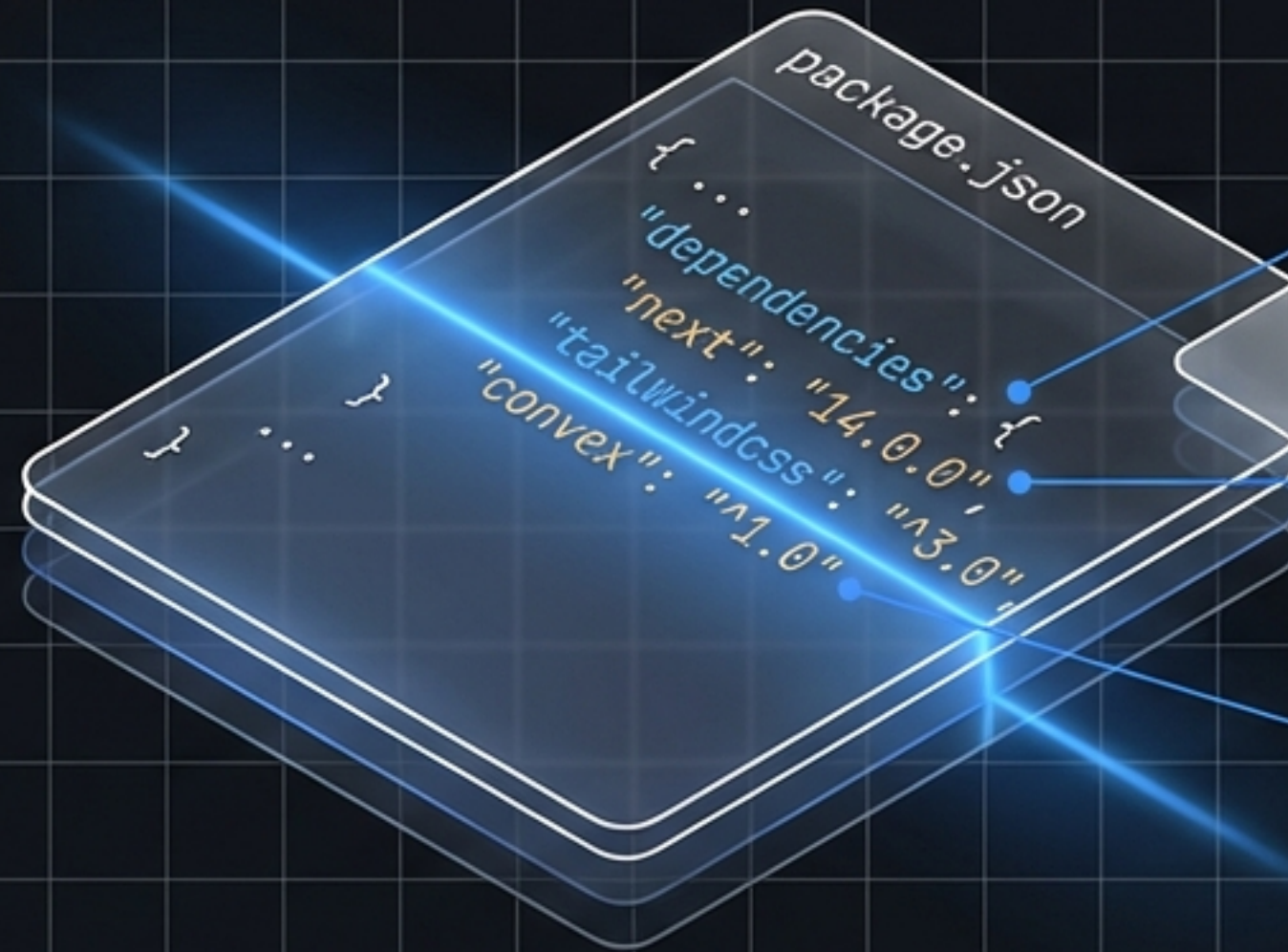
On a tous un problème avec l'IA : ces interfaces qui font trop IA,
ou qui cassent l'architecture existante.

Une approche Agentique : Trois piliers de spécialisation



Kombai ne devine pas. Il scanne, planifie et exécute en utilisant les outils que vous avez déjà installés.

Intelligence Contextuelle : Il lit votre `package.json` avant de coder.



STATUS: CONFIGURATION AUTOMATIQUE DE LA STACK

Déecté : Next.js

Déecté : Tailwind CSS

Déecté : Convex (Pas de Prisma I)

“

"Il arrive à analyser la tech stack et surtout, il vient télécharger tous les composants."

Le principe DRY : Il n'invente pas, il réutilise.

LLM CLASSIQUE

```
return <button  
  className="bg-blue-500 text-  
white p-4 rounded-md  
hover:bg-blue-600">Click me  
</button>
```

Code Jetable (Doublon)

KOMBAI

```
import { Button } from  
"@/components/ui/button"  
  
return  
  <Button variant="primary"  
    size="lg">Click me</Button>
```

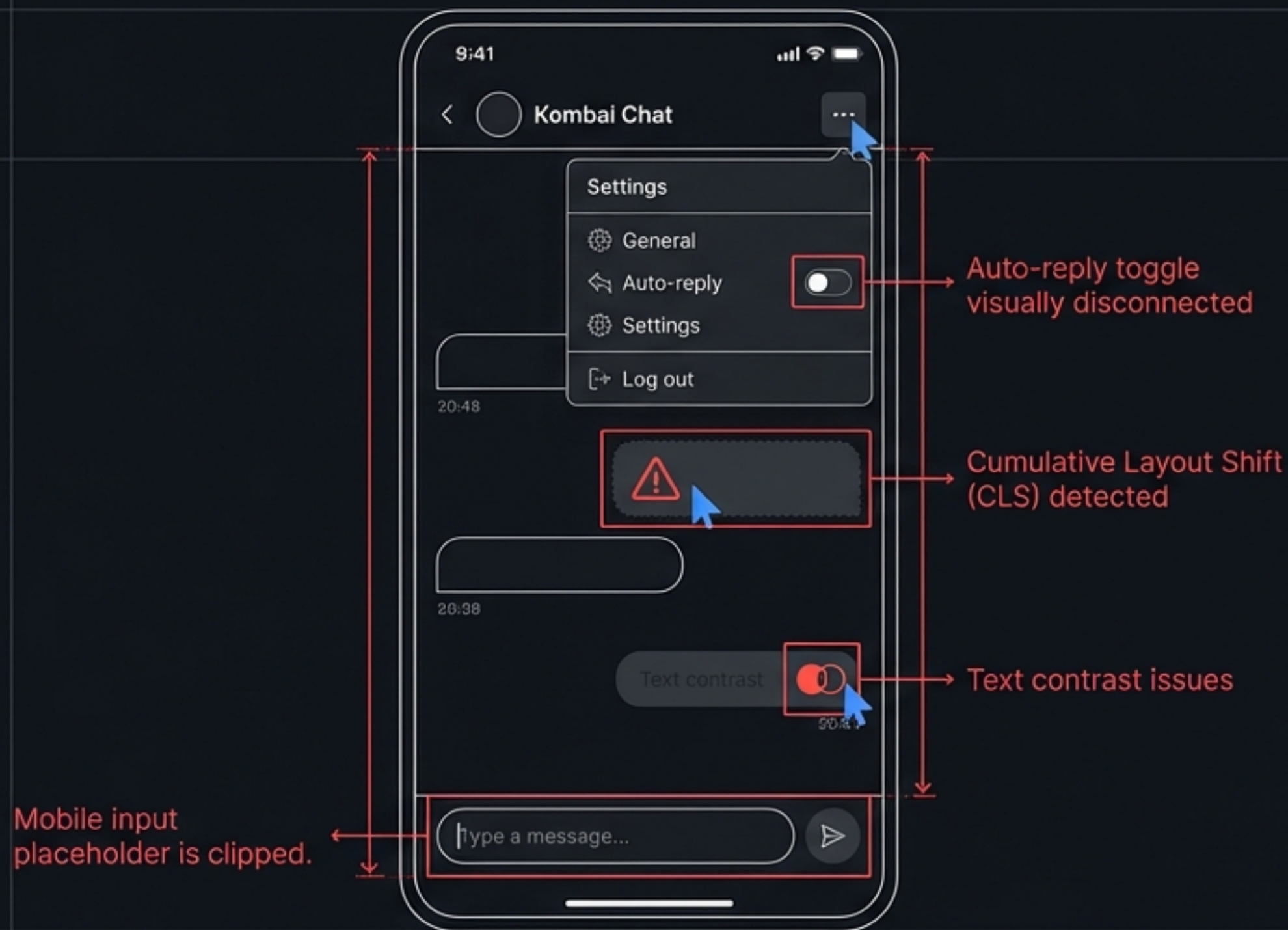
Intégration Design System

INDEXED COMPONENTS

- Alert.tsx
- Dialog.tsx
- AspectRatio.tsx
- AudioPlayer.tsx

Kombai indexe votre dossier `/components`. Il comprend vos props et vos variantes pour générer un code qui s'insère nativement.

L'Audit UI/UX : Un 'Reviewer' visuel dans votre équipe.



RAPPORT D'AUDIT

- [!] Alignement
- [!] Responsivité
- [!] Logique Visuelle

Kombai voit ce que l'utilisateur voit. Il analyse le rendu (rendering) pour détecter les bugs que le code seul ne révèle pas.

'Vibe Coding' : Manipulez le DOM en langage naturel.



Intégration bidirectionnelle. Modifiez l'interface visuellement dans le navigateur intégré, et Kombai répercute les changements proprement dans le code source.

Planification Autonome : 442 secondes de réflexion.

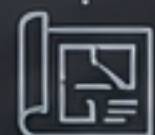
PLANNING PHASE: 442s



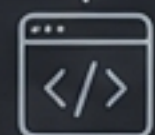
1. Analyse de la demande (Prompt)



2. Lecture des guidelines (Tailwind, Shadcn)



3. Planification de l'architecture



4. Écriture du code (Multi-fichiers)

Pour les tâches complexes, Kombai prend le temps de réfléchir. Il structure son approche avant d'agir, garantissant un résultat "One-Shot" de haute qualité.

Du Prompt au "Award Winning Website".

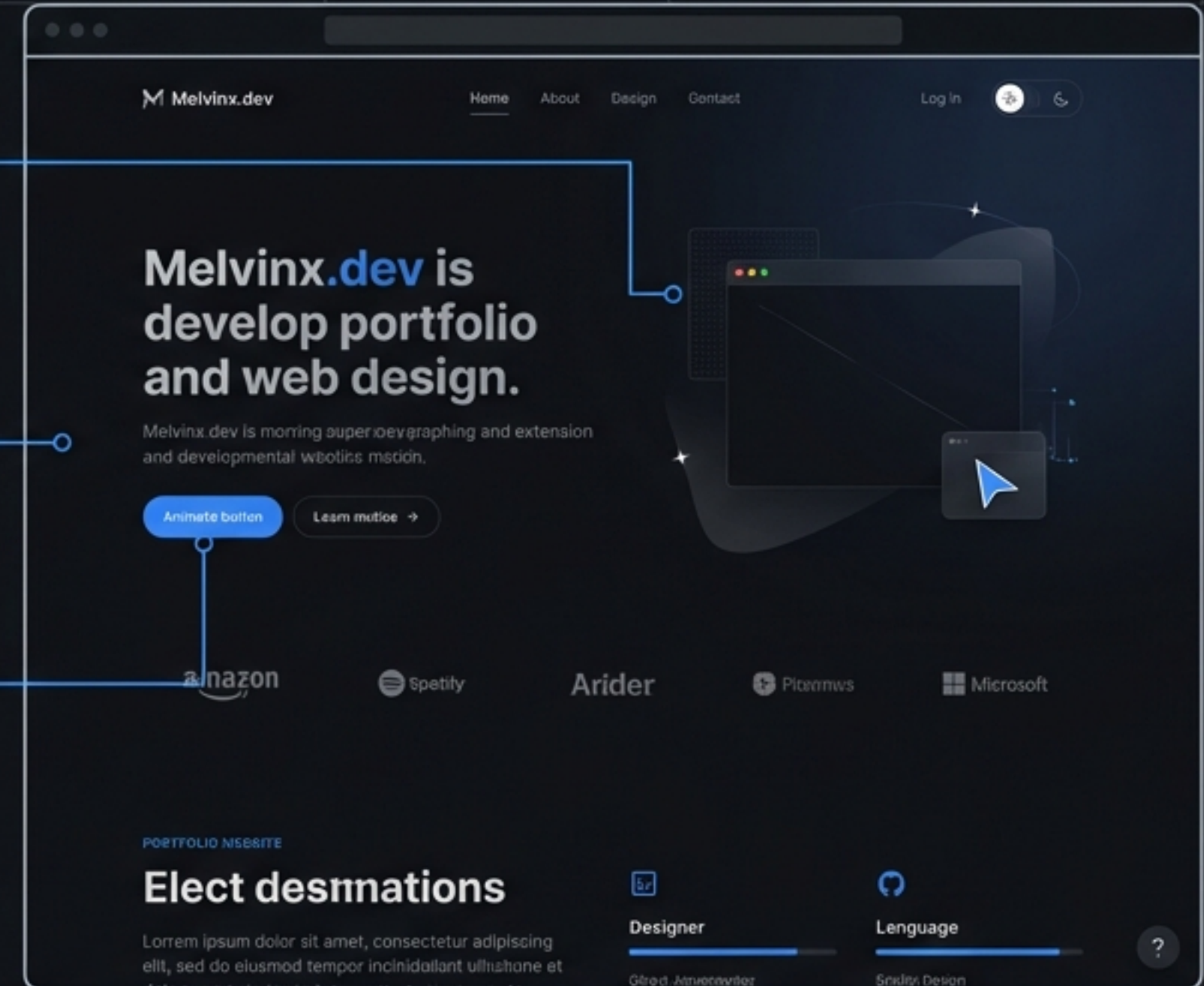
“C’est pas mal pour un one-shot sans rien demander. Le frontend et la partie UI sont 100% réalisés par leur service.”

Génération d'interfaces modernes sans hallucination de styles CSS.

Custom
Scroll
Effects

Dark Mode
Aesthetic

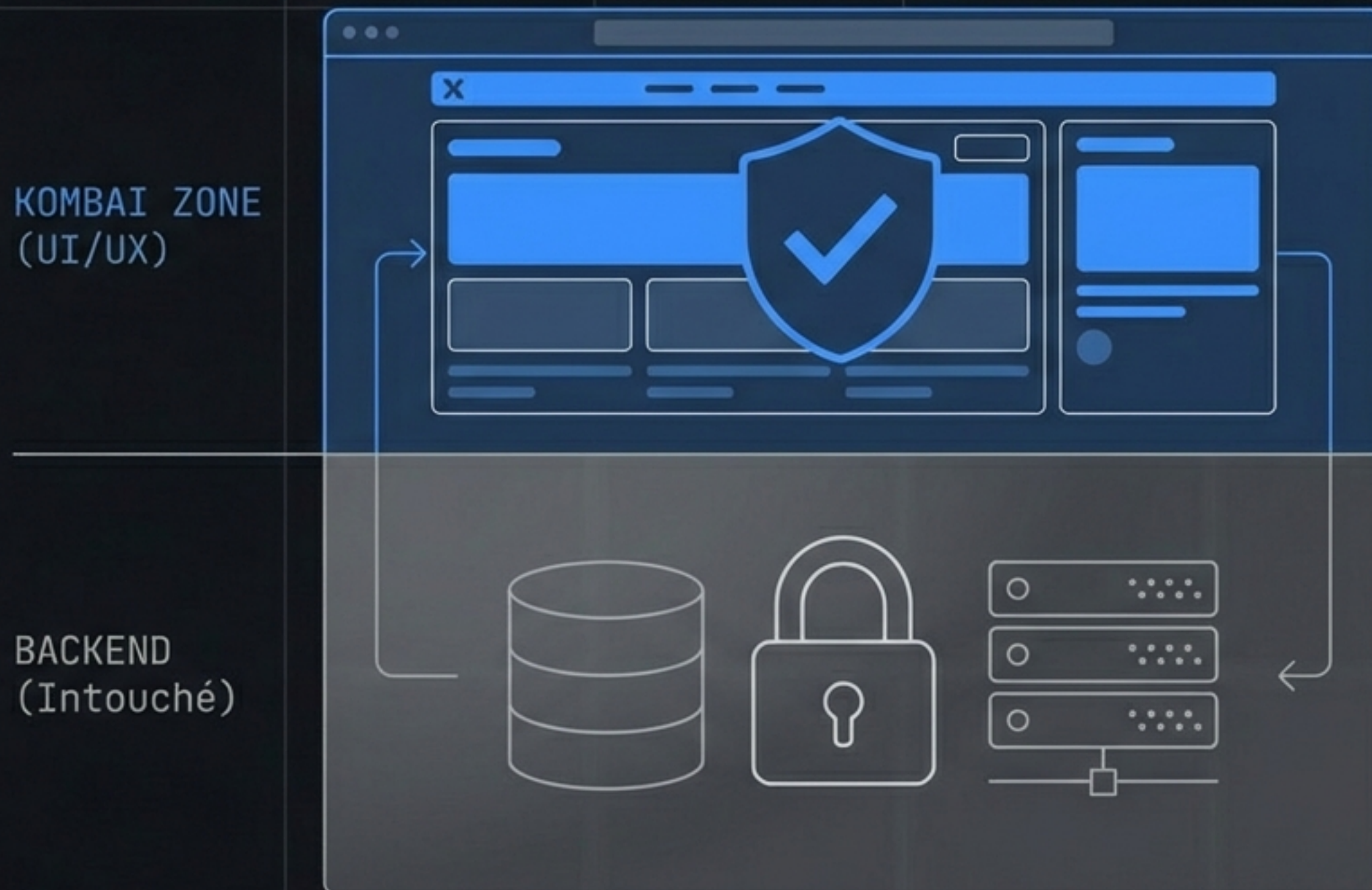
Framer Motion
Animations



Sécurité & Workflow : Isolé au Frontend.

- ✓ Fonctionne en local (Extension VS Code)
- ✓ Certifié SOC2
- ✓ Pas de risque pour la logique métier

Kombai se concentre uniquement sur la couche de présentation. Votre base de données et vos API restent sécurisées.



Ce que disent les développeurs.

Pas d'hallucination, pas de drame backend.
La qualité du code est utilisable
immédiatement.

— *Shark Numbers*

Pas juste 'bon pour une IA', mais réellement
bon.

— *Cess*

Il connaît votre stack vraiment, vraiment
bien.

— *The Dev Logger*

Rien ne s'approche de Kombai pour le
Figma-to-Code.

— *Astro K Joseph*

Mieux que Claude Code et Codex pour le Front.

Le Futur du Frontend est Agentique

Arrêtez de copier-coller du code générique. Commencez à manager un agent qui connaît votre codebase.



Installer l'extension VS Code

JetBrains Mono
Essai gratuit 300 crédits/mois

Un outil qui comprend enfin que le code ne vit pas dans le vide, mais dans un projet.