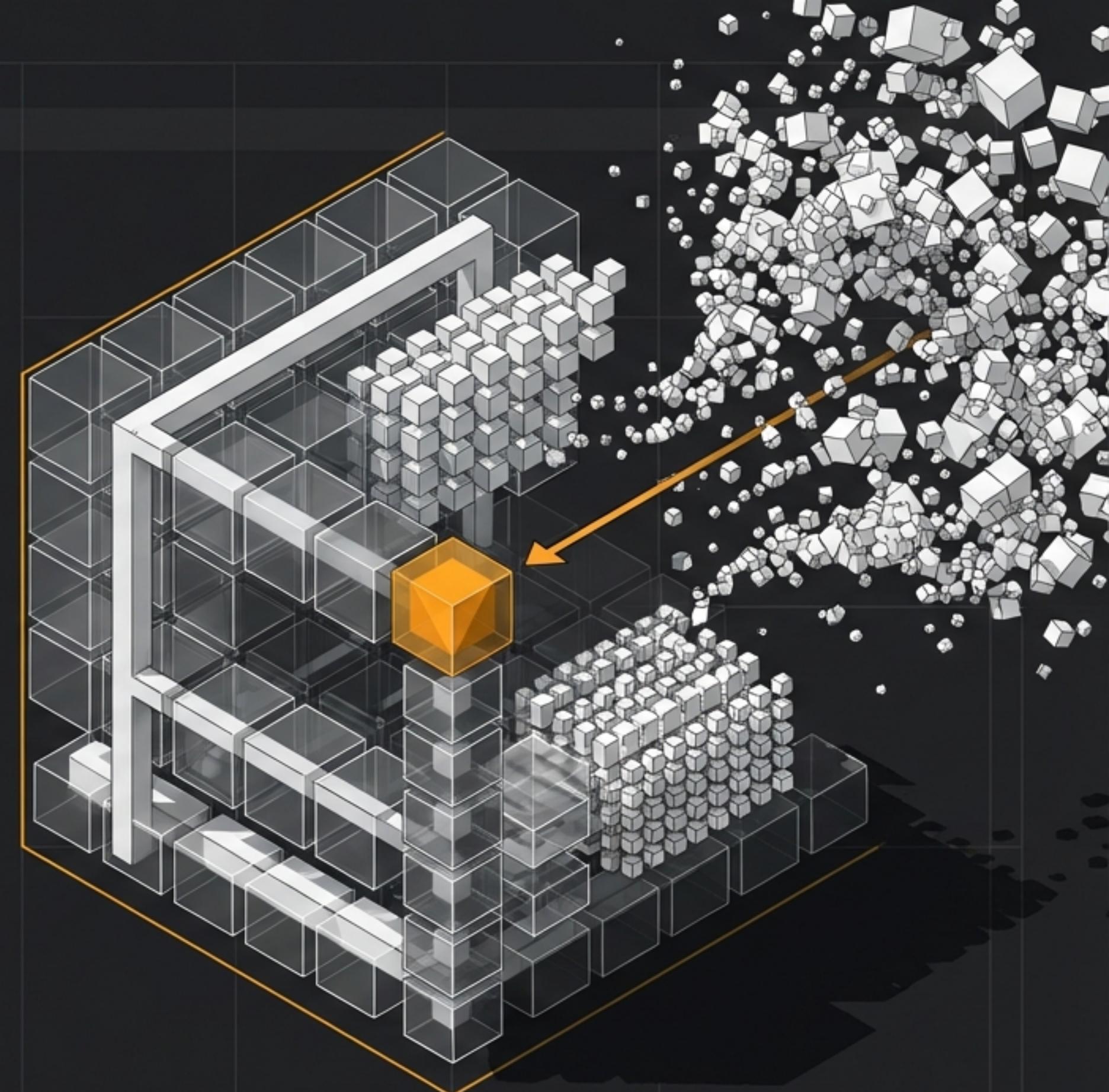


Fra Prompting til Engineering

Konfiguration af GitHub
Copilot i VS Code: Rules,
Skills og Agents 

Internal Team Presentation



Udfordringen: Fra Ikke-deterministisk til Semi-deterministisk

RAW CONTEXT (KAOS)

```
stan.gåBaet
    Uden instruktioner gætter AI'en baseret på træningsdata.
    Den mangler jeres specifikke kontekst,
    hvilket fører til hallucinationer.
    intetion.besvarende = "Generativ, og kontekstbaseret svar"
    return besvarende
}

if (context.get("x") === undefined) {
    stan.østrøg = "Generativ, og kontekstbaseret svar"
}

const contextGet = context.get("x");
const cache = {
    if (cache.get("x")) {
        stan.østrøg = "Generativ, og kontekstbaseret svar"
    }
}

fix.ingetnebysteder = "Generativ, og kontekstbaseret svar"
const contextGet = context.get("x");
const cache = {
    if (cache.get("x")) {
        stan.østrøg = "Generativ, og kontekstbaseret svar"
    }
}

try {
    const response = await fetch("https://api.openai.com/v1/completions", {
        method: "POST",
        headers: {
            "Content-Type": "application/json",
            "Authorization": `Bearer ${process.env.OPENAI_API_KEY}`
        },
        body: JSON.stringify({
            model: "text-davinci-003",
            prompt: `${contextGet} ${fix.ingetnebysteder}`,
            max_tokens: 100
        })
    });
    const data = await response.json();
    stan.østrøg = data.choices[0].text;
} catch (error) {
    stan.østrøg = "Generativ, og kontekstbaseret svar"
}

return stan.østrøg;
```

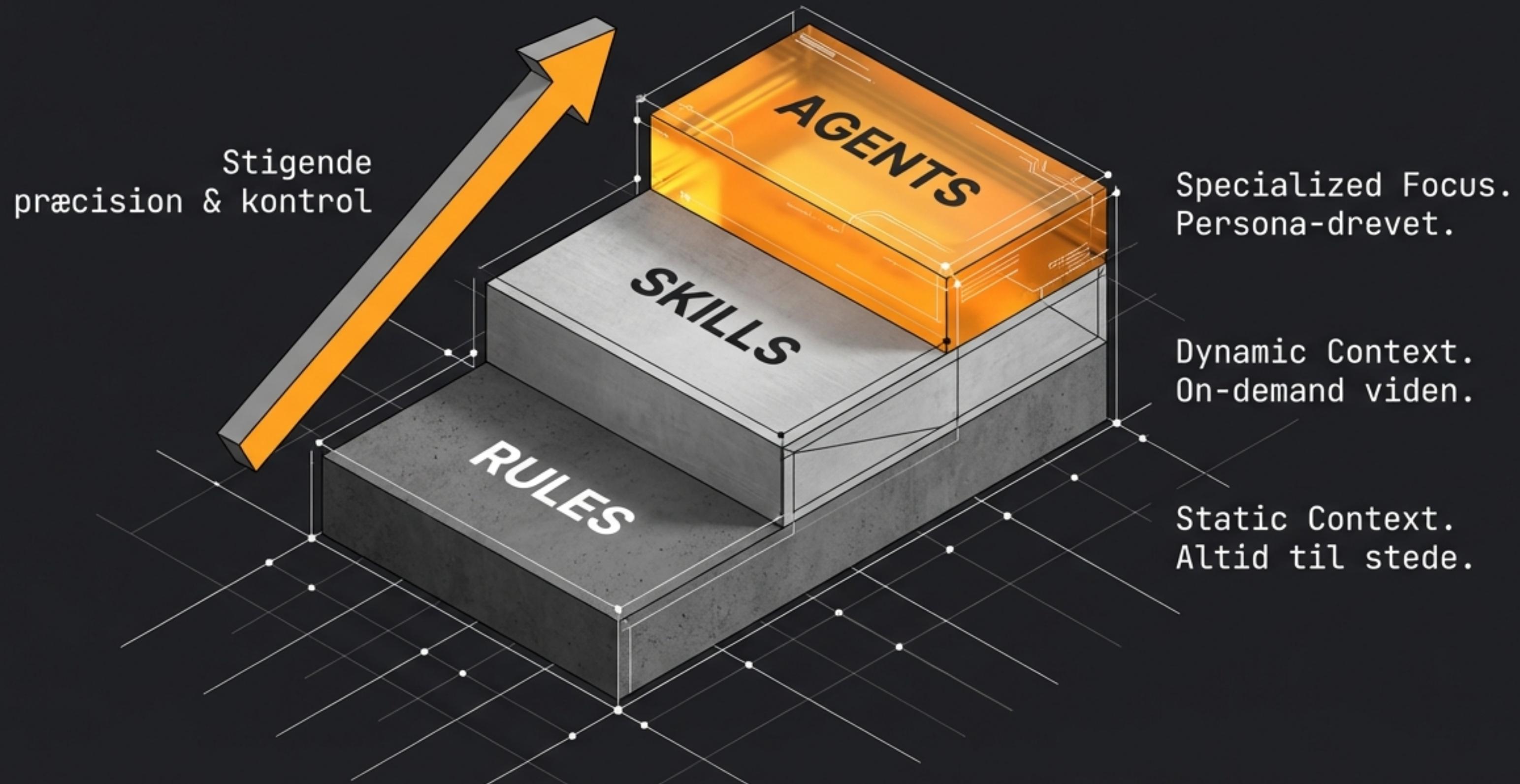
Uden instruktioner gætter AI'en baseret på træningsdata. Den mangler jeres specifikke kontekst, hvilket fører til hallucinationer og generiske svar.

MANAGED CONTEXT (STYRING)

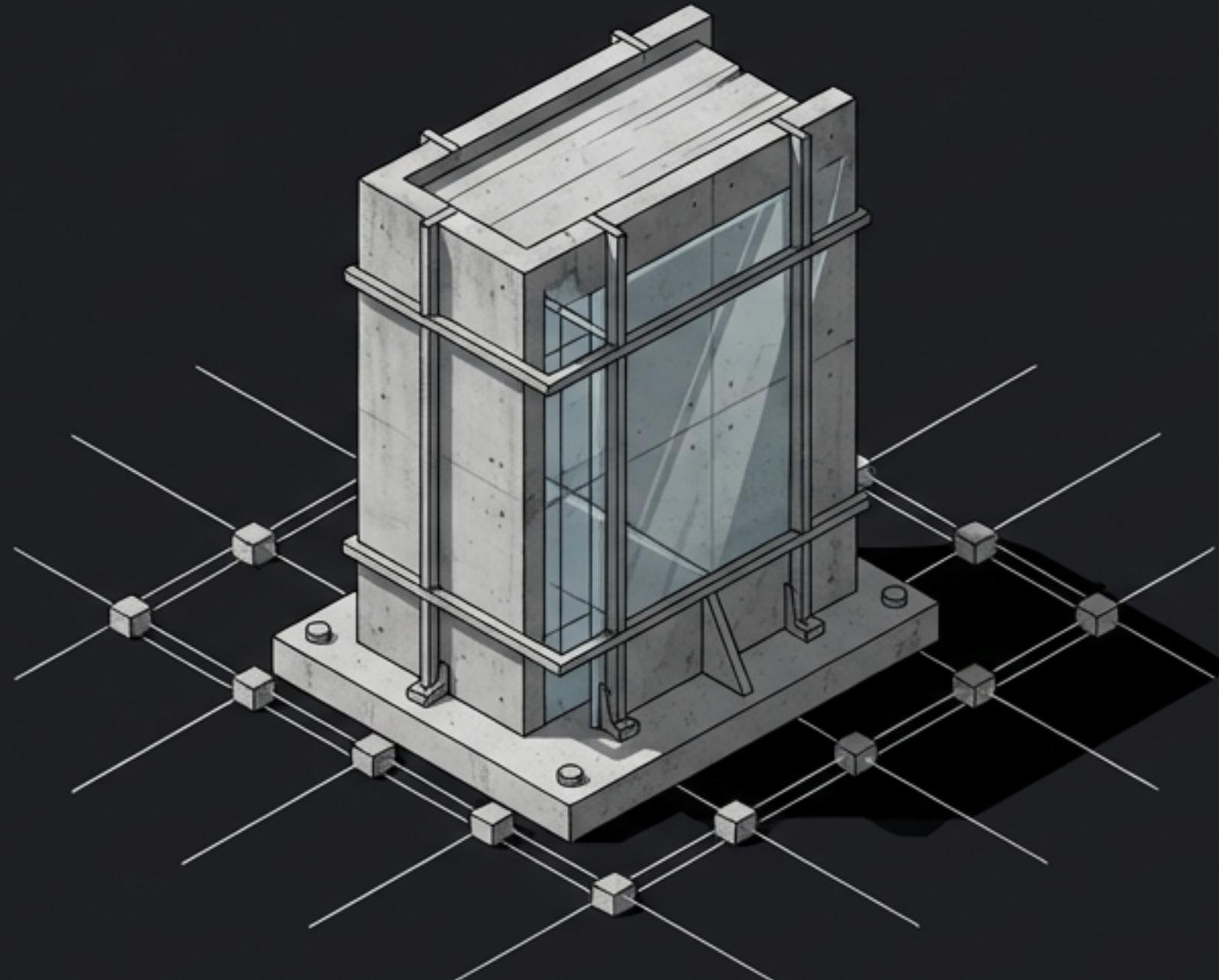


Ved at låse konteksten fast gennem konfiguration, tvinger vi AI'en ned i et 'semi-deterministisk' spor. Vi går fra at håbe på det rigtige svar til at designe det.

Evolutionsstigen for Context Management



Niveau 1: Regler (Static Context)



Regler er "husreglerne", der gælder for hver eneste prompt. Det er her, vi definerer vores kodestandarder, navngivningskonventioner og arkitektoniske principper.

Forhindrer "Context Drift" ved at sikre, at AI'en aldrig glemmer, hvordan vi skriver kode her.

"Den tavse kollega, der altid husker standarde."

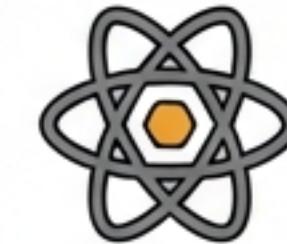
Implementering af Instructions



GLOBAL CONTEXT

.github/copilot-instructions.md

Placeres i roden. Gælder automatisk for ALLE chat requests.



SCOPED CONTEXT

*.instructions.md

Fil-specifikke regler styret af glob patterns.

```
--- applyTo: '**/*.tsx' ---
```

React Guidelines

- Use functional components...

Best Practice for Regelfiler

- **Hold det rent**

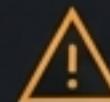
Brug regler til generelle standarder (naming conventions, error handling), ikke til specifikke opgaveløsninger.

- **Modularitet**

Brug flere .instructions.md filer med specifikke scopes frem for én gigantisk fil.

- **Living Artifact**

Opdater løbende. Fejl i PR? Tilføj rettelsen til instruktionsfilen.



Undgå at overfylde Context Window med irrelevant information.

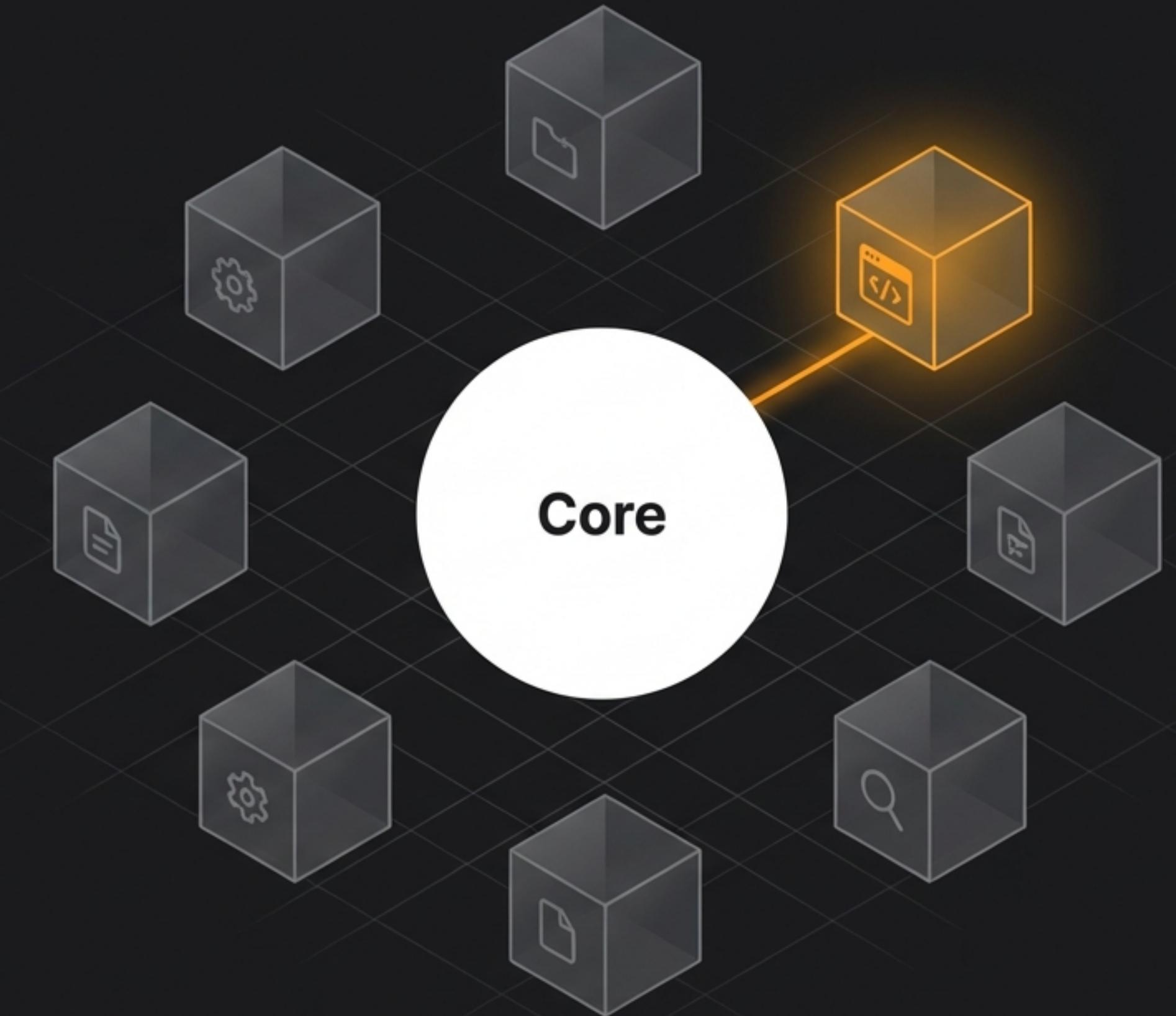
Niveau 2: Skills (Dynamic Context)

Løsningen på “Context Bloat”

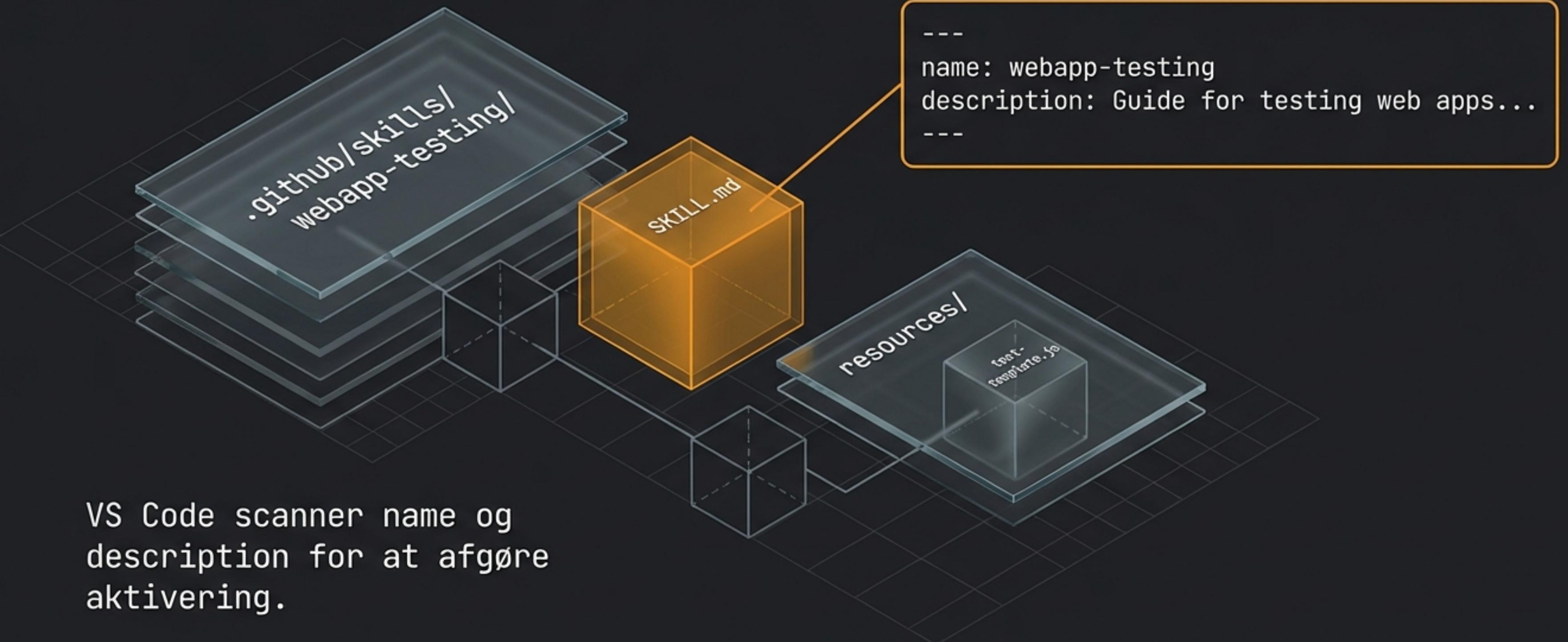
Vi kan ikke lære AI'en alt på én gang. Skills er mapper med instruktioner og scripts, der KUN indlæses, når de er relevante.

Gør det muligt at pakke komplekse workflows (f.eks. testing, PRD-skrevning) ned i moduler.

⚠️ Undgå at overfylde Context Window med irrelevant information.



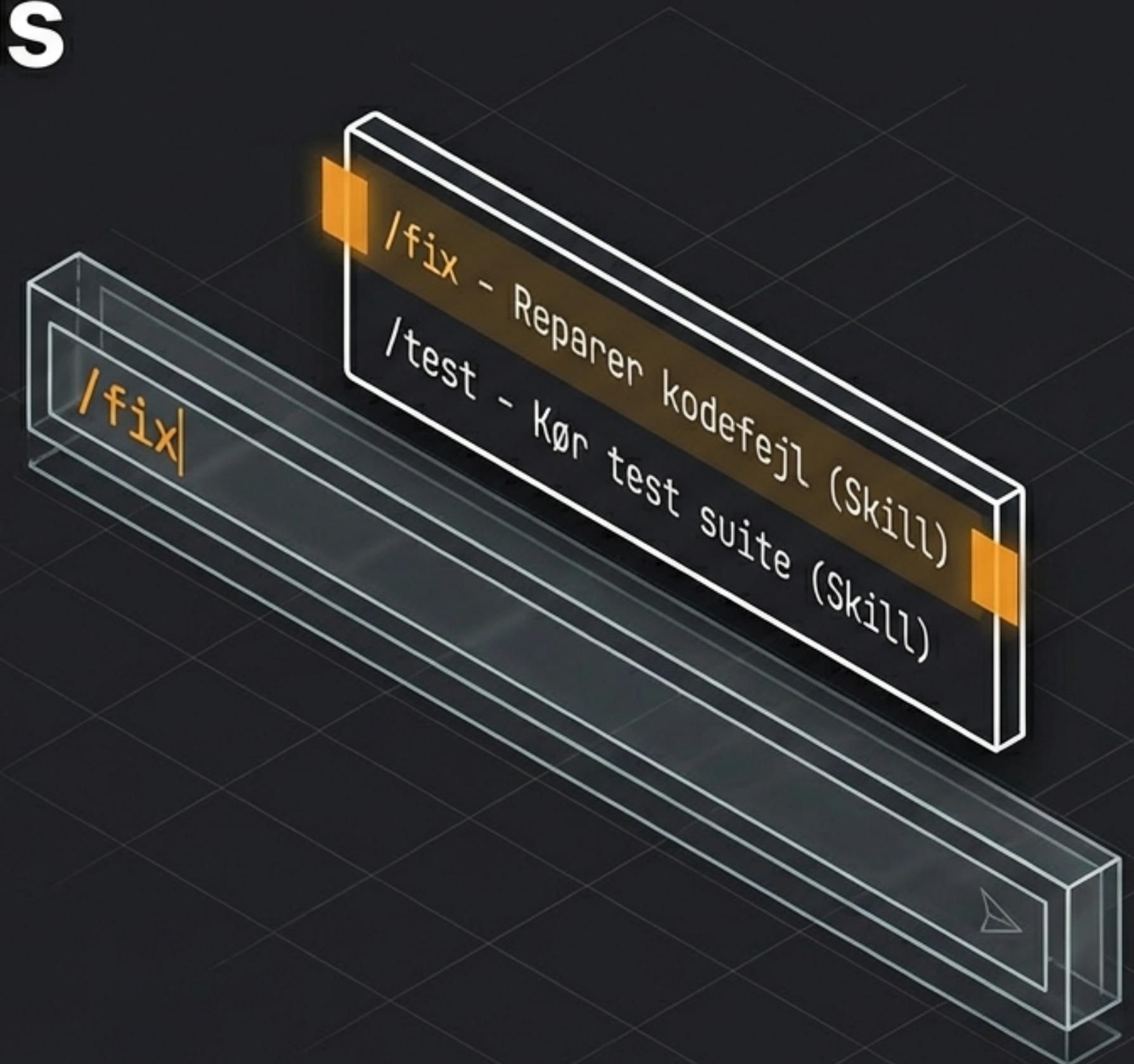
Anatomi af en Skill



Slash Commands & Workflows

Slash commands er genvejen til gentagne workflows. Pak 'Perfect Prompts' ind i en simpel kommando.

Deling af workflows på tværs af teamet via git.



Niveau 3: Custom Agents (Specialized Focus)

At skifte “hat”. En agent er en specialiseret persona med begrænsede værktøjer.
Begrænsning af værktøjer minimerer fejl og øger kvaliteten.



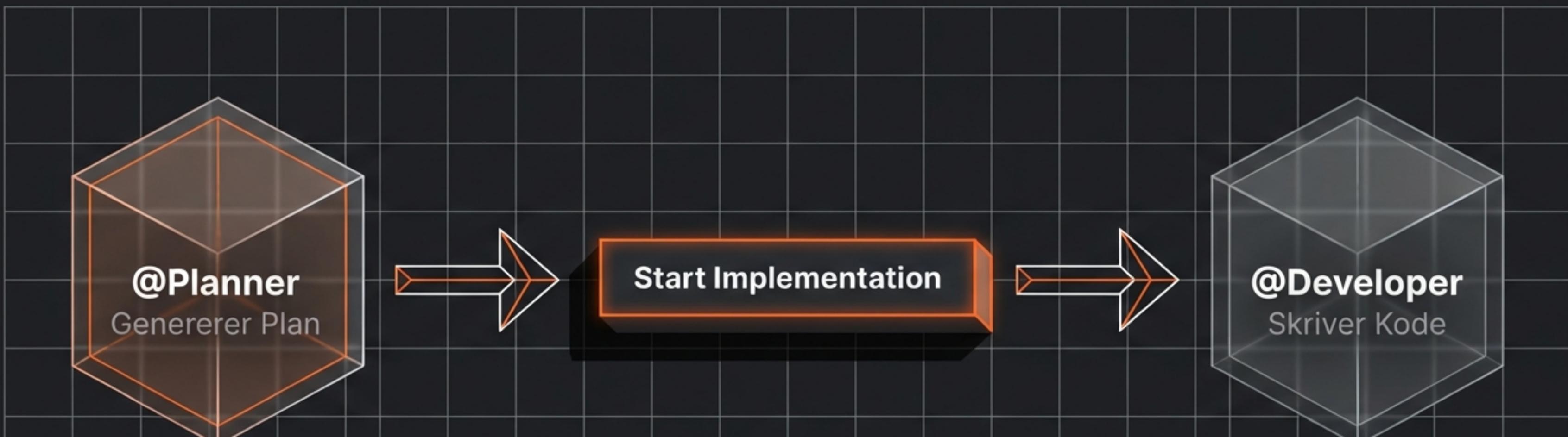
Konfiguration af Custom Agents

```
.github/agents/planner.agent.md  
  
---  
name: Planner  
description: Generate implementation plans.  
tools: ['fetch', 'search', 'githubRepo']  
model: Claude Sonnet 4  
---  
  
# Planning Instructions  
You are in planning mode. Do not write code.
```

NO WRITE ACCESS

Agents defineres i
.agent.md filer og dukker
op i @-menuen.

Handoffs: Orkestrering af Workflows



handoffs:

- label: Start Implementation
- agent: implementation
- prompt: Implement the plan outlined above.

Det Samlede Ecosystem i VS Code



Kombinationen skaber en konsistent, semi-deterministisk udviklingspartner.

Fra Teori til Praksis: Next Steps

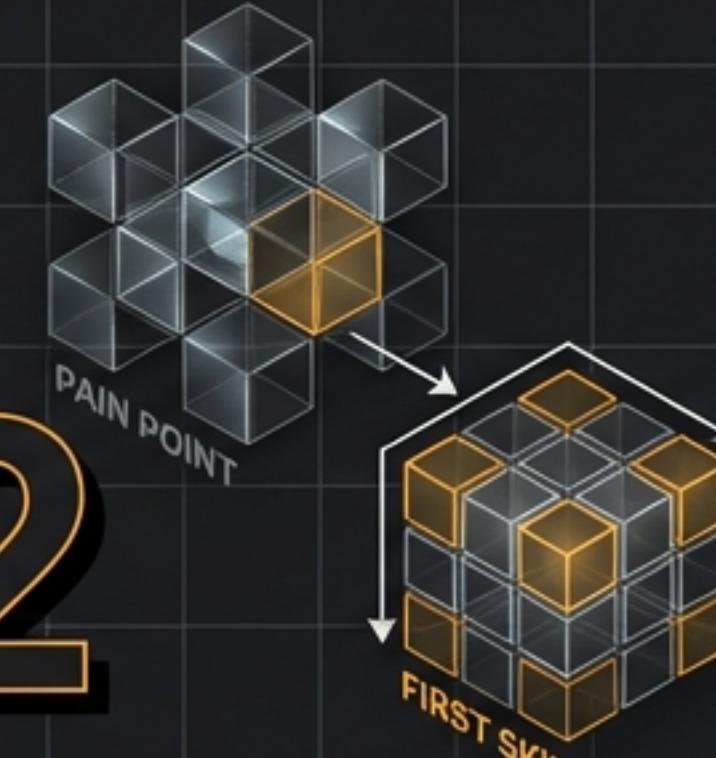


Inter Tight
Etabler Fundamentet

JetBrains Mono

Opret .github/copilot-instructions.md med core guidelines.

01

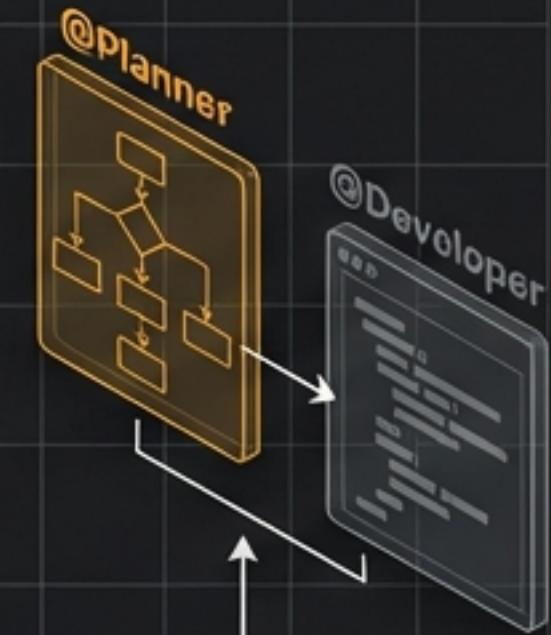


Inter Tight
Byg en Skill

JetBrains Mono

Identifier et pain point (f.eks. test-setup) og lav første Skill.

02



Separerer Arkitektur / Kodning

Inter Tight
Definer Roller

JetBrains Mono

Eksperimenter med en 'Planner' agent for at separere arkitektur fra kodning.

03

Semi-determinisme handler ikke om at begrænse AI'en, men om at give den de rigtige rammer for at lykkes.



Fra operatører til orkestratorer