



De la conception à l'ingénierie du prompt

Maîtriser le langage naturel comme interface de programmation

Nicolas HERNANDEZ
IUT de Nantes - 2026

Un guide académique et technique structuré en 4 parties : Fondations, Anatomie, Ingénierie, Synthèse.

Avertissement

Diaporama généré à plus de 90% par Gemini
NotebookLM et ChatGPT - sept 2026

01

Introduction

« Avec un modèle de langage, la question n'est plus seulement quoi demander, mais comment le demander. »

Définitions Fondamentales

1. Prompt

Requête brute ou ensemble d'instructions adressé à un modèle de langage (LLM).

2. Prompt Design

Art de concevoir et structurer modulairement cette requête pour guider le modèle.

3. Prompt Engineering

Démarche systématique et itérative de conception, test, évaluation, débogage et amélioration.

Efficacité du *Prompt* = Contenu (l'intention) $\times$ Structure (l'architecture)

L'absence de structure dégrade le contenu le plus riche.

Le Problème : Le Prompt Naïf

L'Entrée Naïve

Fais-moi un voyage de 7j en Islande pour 2000€ avec 3 expériences insolites et le budget.



La Sortie Aléatoire

Voici un voyage e

Format imprévisible
(texte narratif long)

Jour 1 : Visite du Blue Lagoon.
C'est très b

Contenu générique
(Blue Lagoon n'est pas insolite)

Jour 2 : Cascades...

...

Contraintes ignorées
(Budget total souvent dépassé)

Constat : Un prompt naïf délègue la prise de décision architecturale au modèle, entraînant une perte de contrôle totale.

Contenu et structure

La qualité d'une requête dépend à la fois de ce qui est demandé et de la manière dont la demande est structurée.

VAGUE

Organise un voyage insolite en Islande

PRÉCIS

Organise-moi un voyage insolite en Islande pour 7 jours.

Budget : 2 000 €.

Je veux quelque chose d'original.

STRUCTURÉ ET OPERATIONNEL

TASK

Proposer un voyage insolite de 7 jours en Islande.

CONTEXT

Budget maximal : 2 000 € ; voyageur recherchant l'insolite.

OUTPUT

1 destination + 3 expériences + budget estimatif.

(Du peu au structuré) Même intention → ambiguïtés différentes.

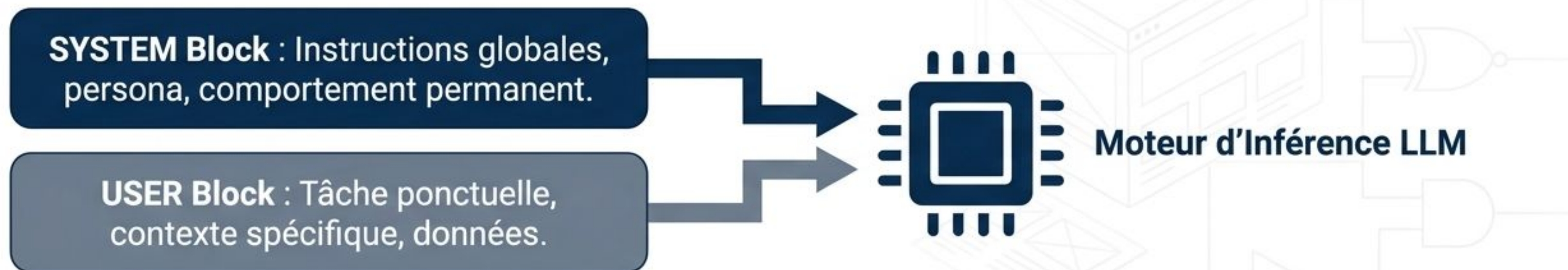
La structure ne garantit pas un bon résultat, mais elle rend les attentes plus explicites et testables.

02

Anatomie du prompt

De la requête libre à une structure explicite.

Cadres de Conception et Architecture de Base



Framework	Acronyme	Cas d'usage idéal
CO-STAR	Context, Objective, Style, Tone, Audience, Response	Marketing et création de contenu
RISEN	Role, Instruction, Steps, End goal, Narrowing	Tâches complexes à étapes multiples
RICE	Role, Instructions, Context, Examples	Usage général rapide et fiable

Source : AIUnpacking

Malgré la variation des acronymes, l'anatomie d'un prompt repose toujours sur un ensemble fini de composants fonctionnels universels.

Frameworks : des repères, pas des standards universels

Les sources proposent des terminologies et des structures qui se recouvrent partiellement.

Source	Composants / techniques	Intention pédagogique
OpenAI	instructions claires · contexte · exemples · format	clarifier puis préparer la sortie
Anthropic	instructions · contexte · exemples · balises XML · rôle	séparer les blocs et désambiguïser
Google AI	zero/few-shot · structure · contraintes · étapes	guider le comportement attendu
Microsoft	instructions · contexte · exemples · décomposition · format	adapter la technique au scénario

Correspondance pédagogique retenue ici : TASK · ROLE · CONTEXT · OUTPUT_FORMAT · FEW-SHOT · raisonnement structuré.

Une même idée, plusieurs noms

Les composants se recouvrent ; la terminologie n’est pas normalisée.

Concept	Terminologies possibles	Question à se poser
Tâche	TASK · ACTION · INSTRUCTION	ce que le modèle doit faire
Rôle	ROLE · PERSONA	perspective / expertise mobilisée
Contexte	CONTEXT · BACKGROUND · INPUT	informations nécessaires
Sortie	OUTPUT FORMAT · RESPONSE FORMAT	forme de la réponse
Exemples	FEW-SHOT · EXAMPLES · DEMONSTRATIONS	comportement montré par des cas
Décomposition	STEPS · DECOMPOSITION · WORKFLOW	problème divisé en sous-tâches

Anatomie 1 : TASK (La Tâche)

Le Composant

- **Composant** : TASK
- **Synonymes** : Objective, Instruction
- **Position** : USER (Obligatoire)
- **Rôle** : Composant central de la requête. Doit utiliser des **verbes d'action précis** (créer, extraire, synthétiser) pour définir l'attente principale.

```
<user>
```

```
  <task>
```

```
    Propose un itinéraire de voyage de 7  
    jours en Islande. Présente une destination  
    principale, trois expériences originales et  
    une estimation du budget total.
```

```
  </task>
```

```
</user>
```

Anatomie 2 : ROLE (L'Expertise)

Le Composant

- **Composant** : ROLE
- **Synonymes** : Persona, Style, Tone
- **Position** : SYSTEM (Recommandé)
- **Rôle** : Définit l'expertise et l'intention (purpose). Force le modèle à piocher dans un espace latent sémantique spécifique (ex: expert vs amateur).

```
1 <system>
2   <role>
3     Tu es un agent d'une agence de voyage spécialisé
4     dans les voyages insolites et hors des sentiers
5     battus. Ton but est de fuir le tourisme de masse.
6   </role>
7 </system>
8
9 <user>
10   <task>
11     Propose un itinéraire de voyage de 7 jours en
12     Islande. Présente une destination principale, trois
13     expériences originales et une estimation du budget total.
14   </task>
15 </user>
```

ROLE : cadrer une perspective

Un rôle peut préciser fonction, expertise et expérience ; il ne remplace pas les données nécessaires.

SYSTEM

ROLE

Tu es un agent de voyage spécialisé dans les séjours insolites en Islande.

SYSTEM RULE

Privilégie des propositions réalisables et explicites sur leurs hypothèses.

À retenir

Le rôle cadre la réponse.

La règle système cadre le comportement général.

Les contraintes du cas restent à préciser dans la demande.

Anatomie 3 : CONTEXT (L'Ancrage)

Le Composant

- **Composant** : CONTEXT
- **Synonymes** : Grounding, Background
- **Position** : USER (Parfois SYSTEM)
- **Rôle** : Informations externes et contraintes absolues. Évite les hallucinations et délimite l'espace des possibles. Essentiel pour la fiabilité.

```
1 <system>
2   <role>Tu es un agent d'une agence de voyage spécialisé
3   dans les voyages insolites...</role>
4 </system>
5
6 <user>
7   <context>
8     - Le budget total maximal du client est strictement de
9     2 000 €.
10    - Le client souhaite éviter les lieux trop touristiques
11    comme le Cercle d'Or ou le Blue Lagoon.
12  </context>
13
14  <task>
15    Propose un itinéraire de voyage de 7 jours en Islande...
16  </task>
17 </user>
```

CONTEXT : distinguer instruction et données

Une instruction dit **quoi faire** ; le contexte fournit **ce qu'il faut utiliser ou savoir pour le faire**.

INSTRUCTION

Propose un voyage insolite de 7 jours.

CONTEXTE

Budget : 2 000 €

Voyageur : niveau de randonnée modéré

Période : octobre

Priorité : nature et expériences locales

Bon réflexe : identifier explicitement quelles informations sont stables, variables ou fournies par l'utilisateur.

Anatomie 4 : OUTPUT_FORMAT (La Structure)

Le Composant

- **Composant** : OUTPUT_FORMAT
- **Synonymes** : Response, End goal
- **Position** : USER (Recommandé)
- **Rôle** : Contraint la structure de sortie (Markdown, XML, JSON) pour permettre une lecture humaine claire ou l'intégration par une machine.

```
1  [...]
2  <user>
3    <context>[...]</context>
4    <task>[...]</task>
5
6    <output_format>
7      Réponds UNIQUEMENT avec un objet JSON valide suivant
8      cette structure exacte :
9      {
10        \"destination\": \"\",
11        \"experiences_insolites\": [],
12        \"total_cost_euros\": 0
13      }
14    </output_format>
15  </user>
```

OUTPUT_FORMAT : réduire l'ambiguïté

Plus la sortie est destinée à être consommée ou comparée, plus la structure explicite devient utile.

1 libre

Réponds avec une proposition de voyage.

2 texte

Présente :
destination,
expériences,
budget.

3 liste

- Destination
- 3 expériences
- Budget

4 tableau

Étape	Lieu	Coût
...	...	...

5 Markdown/JSON

```
### Itinéraire
- ...
### Budget
| ... | ... |
```

Le bon format dépend du destinataire et du système qui exploitera la sortie.

Anatomie 5 : FEW-SHOT SAMPLES (L'Exemple)

Le Composant

- **Composant** : FEW-SHOT SAMPLES
- **Position** : USER
- **Rôle** : Apprentissage en contexte (In-Context Learning). Permet de transférer un format ou logique complexe par l'exemple de manière déterministe.



Zero-shot



Few-shot

```
1 [...]
2 <user>
3   <output_format>[...]</output_format>
4
5   <examples>
6     Input: Voyage insolite 3j en Écosse, 500€.
7     Output: {
8       "destination": "Île de Jura",
9       "experiences_insolites": [
10        "Dégustation whisky distillerie fermée",
11        "Nuit en Bothy"
12      ],
13     }, "total_cost_euros": 480
14   </examples>
15 </user>
```

Zero-shot → one-shot → few-shot

Les exemples montrent le comportement attendu ; ils doivent être proches, cohérents et suffisamment variés.

ZERO-SHOT

Aucun exemple

Propose 3 expériences insolites en Islande.

ONE-SHOT

1 exemple

INPUT : 2 jours
OUTPUT : lieu + activité + coût

FEW-SHOT

Plusieurs exemples

INPUT → OUTPUT
cas court · cas long · cas limite

INPUT

OUTPUT

PATTERN

Anatomie 6 : STRUCTURED REASONING (La Logique)

Le Composant

- **Composant** : STRUCTURED REASONING
- **Synonymes** : Steps, Chain of Thought
- **Position** : USER
- **Rôle** : Pousser le modèle à décomposer son calcul. Critique pour le respect des budgets et la résolution de problèmes mathématiques ou logiques.

```
1  [...]
2  <user>
3    <examples>[...]</examples>
4    <reasoning_steps>
5      Avant de générer le JSON, affiche ton raisonnement :
6      Étape 1 : Sélectionne la région insolite.
7      Étape 2 : Chiffre chaque expérience individuellement.
8      Étape 3 : Additionne et valide que la somme ne dépasse
9        pas 2000 €.
10   </reasoning_steps>
11 </user>
```

Raisonnement : procédure, pas exposition automatique

L'enjeu pédagogique est d'organiser les étapes utiles sans transformer le prompt en demande d'exposition exhaustive d'un raisonnement interne.

À UTILISER AVEC PRUDENCE

DEMANDER

« Décris ton raisonnement détaillé étape par étape »

ALTERNATIVE

STRUCTURER

« 1. Extrais les contraintes.
2. Vérifie le budget.
3. Génère puis contrôle 3 itinéraires. »

Pour le voyage, on privilégie des étapes vérifiables et une justification concise lorsque nécessaire.

Choisir une technique selon la tâche

Ce sont des heuristiques : elles orientent les essais, sans constituer des règles absolues.

Type de tâche	Technique	Objectif
Question simple	Zero-shot	formuler directement la demande
Format précis	OUTPUT_FORMAT	contraindre la structure
Classification	Few-shot	montrer les catégories attendues
Transformation	Exemples	montrer la transformation
Tâche complexe	Décomposition	diviser le problème
Données fournies	CONTEXT	apporter les informations nécessaires

Synthèse : Matrice de Conception

Type de Tâche	TASK	ROLE	CONTEXT	FORMAT	FEW-SHOT	REASONING
Création de contenu (ex: Article)	Obligatoire	Obligatoire	Recommandé	Recommandé	— Facultatif	— Facultatif
Extraction de données (ex: Parsing)	Obligatoire	— Facultatif	Recommandé	Obligatoire	Obligatoire	— Facultatif
Résolution complexe (ex: Calcul)	Obligatoire	Recommandé	Obligatoire	Recommandé	Obligatoire	Obligatoire

Règle d'ingénierie

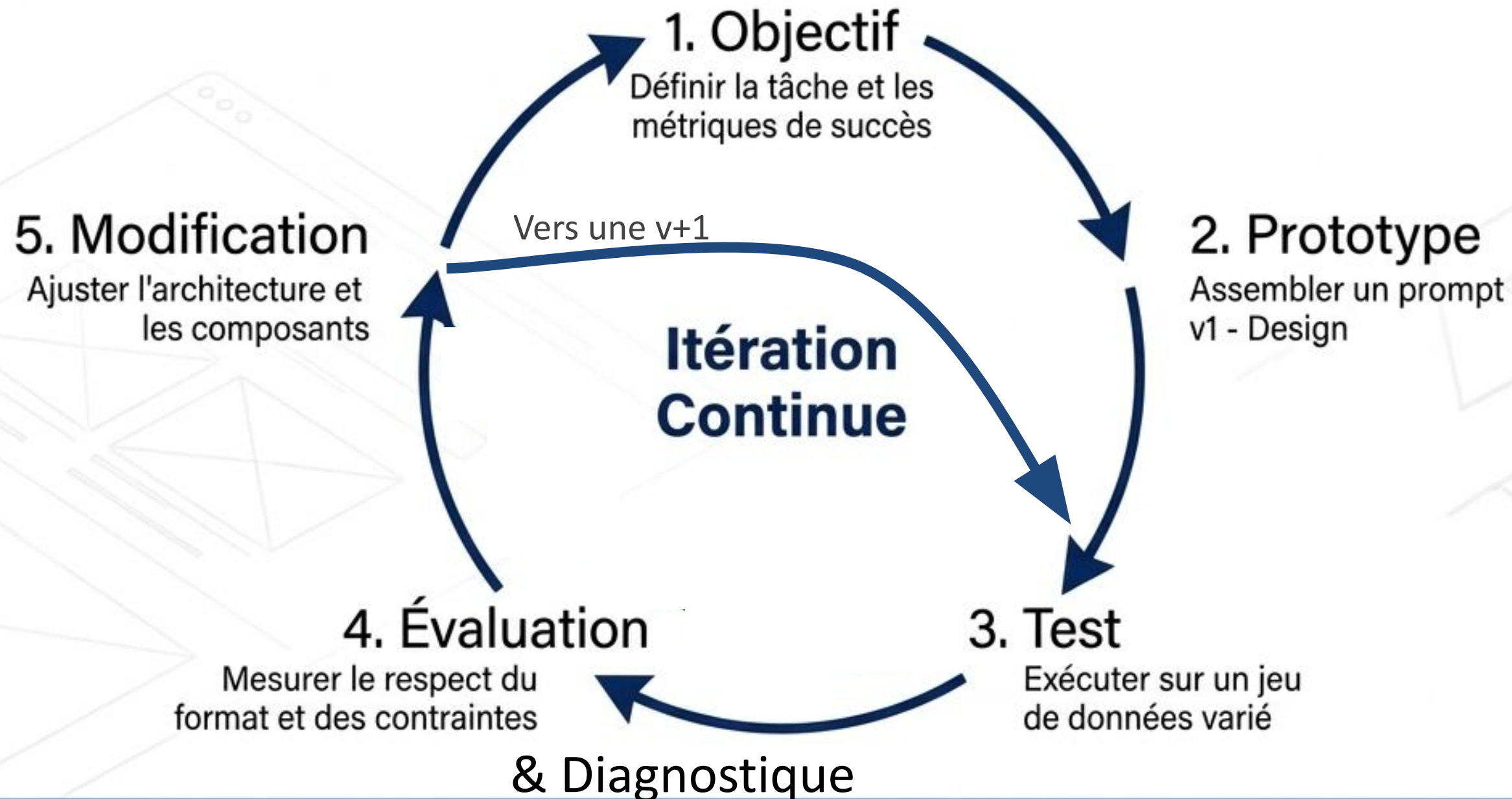
Le Prompt Design consiste à sélectionner et moduler ces blocs selon la **complexité algorithmique** de la tâche cible.

03

Du prompt design au prompt engineering

Évaluer, diagnostiquer, itérer — puis intégrer.

Prompt Engineering : Le Cycle de Vie (PromptOps)



Passer de « rédacteur de texte » à « développeur de prompt » signifie accepter que le premier jet n'est qu'un prototype qui doit être éprouvé et compilé.

Construire une première version

Une première version n'est pas un produit fini : elle doit être testée.

ROLE	CONTEXT	TASK	CONSTRAINTS	OUTPUT
agent de voyage	7 jours · 2 000 €	3 itinéraires	≤ 2 000 €	tableau Markdown

V1

ROLE : agent de voyage spécialisé dans les séjours insolites en Islande.
CONTEXT : 7 jours · budget maximal 2 000 € · randonnée modérée.
TASK : propose 3 itinéraires insolites.
CONSTRAINTS : aucun itinéraire ne dépasse 2 000 € ; durée totale = 7 jours.
OUTPUT : tableau Markdown + total par itinéraire.

Un cas de test est une hypothèse vérifiable

Tester un prompt demande des entrées représentatives et des critères explicites.

ID	Entrée	Critère	Attendu
T1	7 jours · 2 000 €	format OK + budget OK	tableau + total
T2	7 jours · 1 200 €	budget respecté	aucune option > 1 200 €
T3	2 jours · 400 €	durée respectée	ne pas extrapoler à 7 jours
T4	profil non-randonneur	adaptation au public	éviter activités exigeantes
T5	cas limite : budget très faible	gestion de contrainte	expliquer les compromis

Un bon jeu de tests couvre le cas nominal, les variantes d'entrée et les cas limites.

Évaluer : quels critères ?

Une évaluation utile transforme une intuition (« c'est mieux ») en observation comparable.

Adéquation

La tâche demandée est-elle réellement effectuée ?

Complétude

Toutes les informations attendues sont-elles présentes ?

Contraintes

Les limites explicites sont-elles respectées ?

Format

La structure demandée est-elle respectée ?

Robustesse

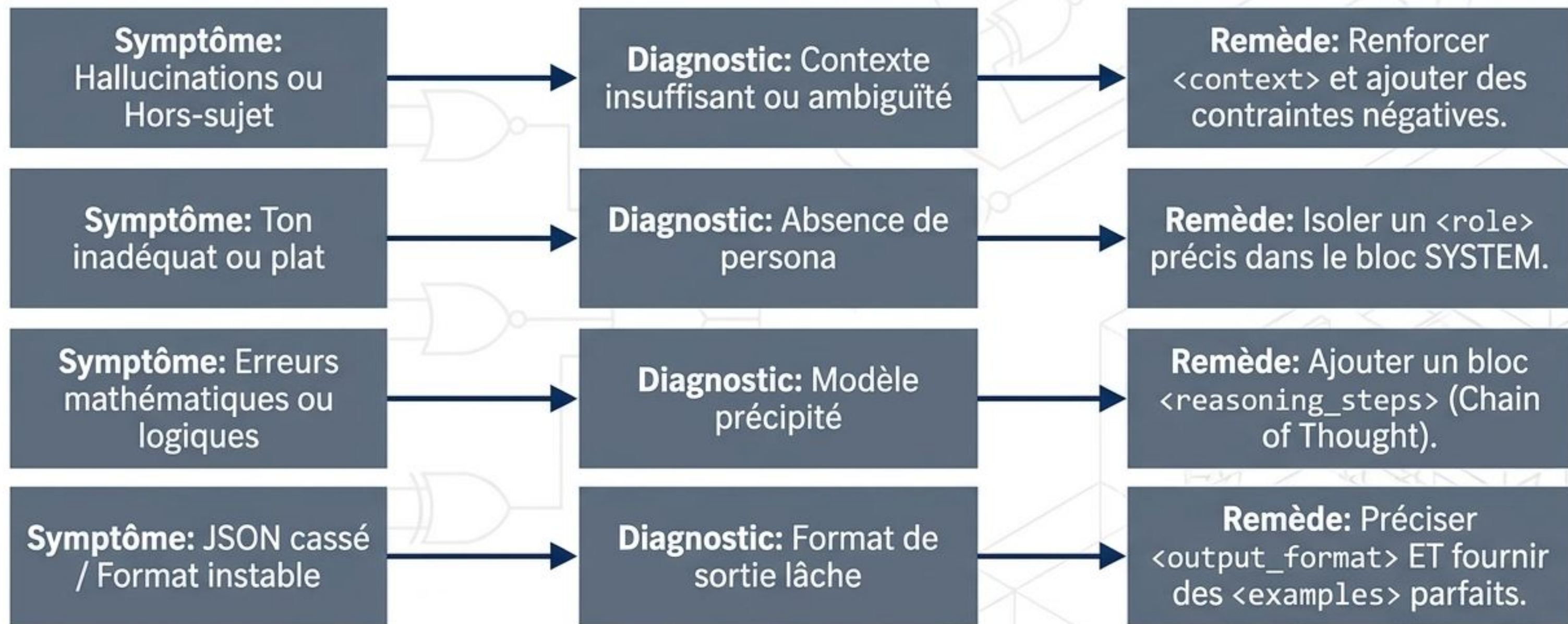
Le comportement tient-il sur plusieurs variantes ?

Régression

Une amélioration casse-t-elle un cas déjà correct ?

Versionner les prompts rend le cycle reproductible.

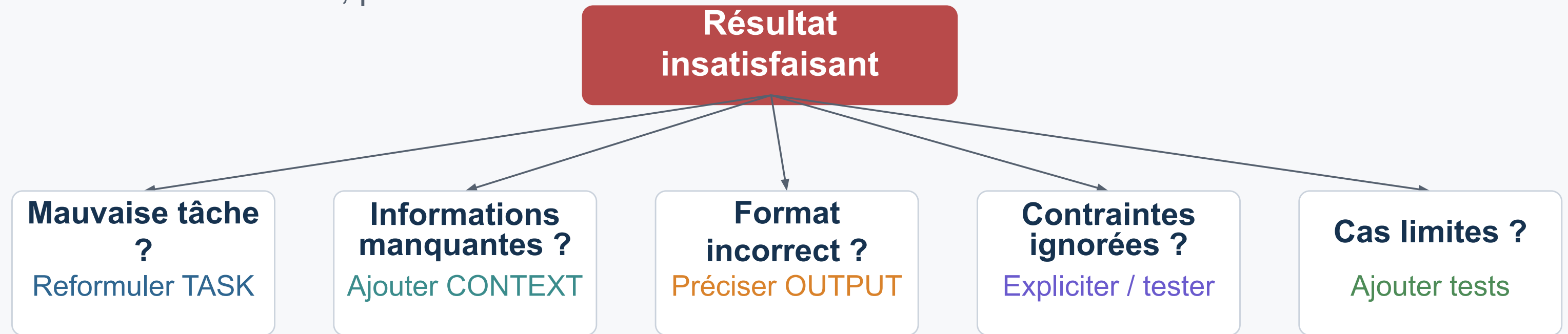
Diagnostic et Résolution (Troubleshooting)



L'ingénierie du prompt consiste à diagnostiquer méthodiquement l'échec de la sortie pour corriger chirurgicalement l'entrée.

Déboguer un prompt : arbre de diagnostic

Partir de l'écart observé, puis tester une cause à la fois.



Autres causes : exemples incohérents · contexte trop long · instructions contradictoires · dépendance à une formulation particulière.

Anti-patterns : écrire plus ne résout pas tout

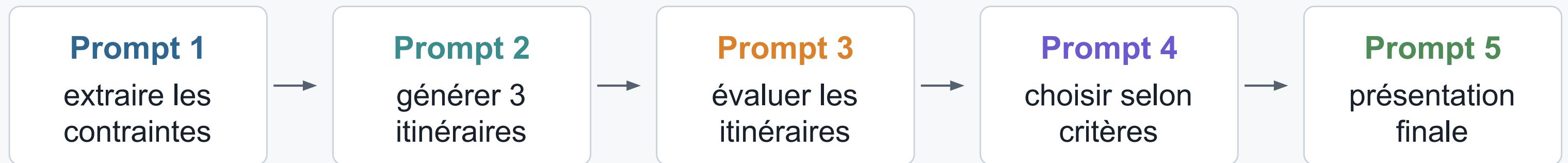
Chaque défaut doit être relié à une conséquence observable puis à une correction.

Problème	Conséquence possible	Correction
Instruction vague	sortie imprévisible	rendre le résultat observable
Prompt très long sans structure	signal noyé	segmenter en blocs
Instructions contradictoires	priorités ambiguës	supprimer / hiérarchiser
Contexte insuffisant	réponse générique	fournir les données utiles
Contexte inutilement volumineux	bruit / coût	réduire aux informations pertinentes

Question à poser : « Quel défaut précis de sortie ce changement est-il censé corriger ? »

Décomposer une tâche complexe : chaînage de prompts

Pour les problématiques algorithmiques complexes, il est préférable de chaîner plusieurs prompts simples et spécialisés plutôt que de concevoir un méga-prompt mégalithique et fragile



Le chaînage peut rendre le système testable ; il augmente aussi le nombre de points à surveiller.

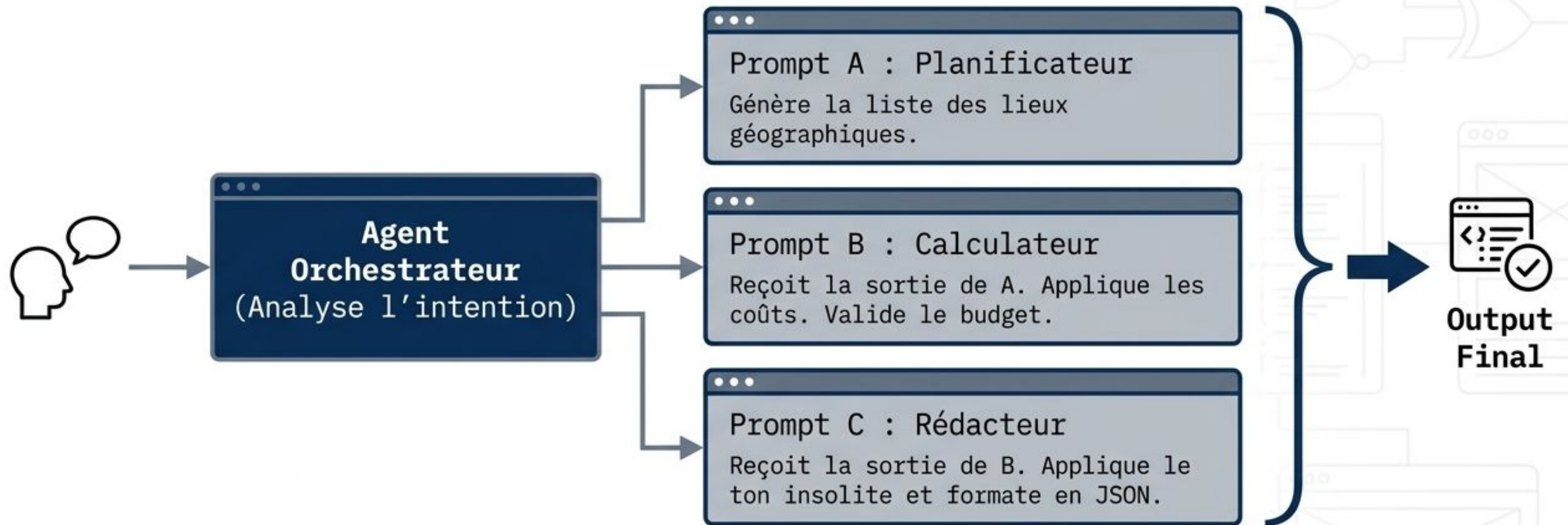
GAINS

testabilité · modularité

COÛTS

latence · orchestration ·
erreurs cumulées

Workflow agentique



Workflow agentique : plusieurs étapes, outils, modèles ou agents peuvent être orchestrés. Le choix dépend de la complexité du problème.

Tester et développer les prompts

L'outillage aide à comparer des versions, automatiser des contrôles et suivre les régressions.

Promptfoo

tester des prompts / modèles / cas ; assertions et comparaison

Opik

Agent optimization and Automated prompt engineering: Prompt and agent development, testing (LLM and agent evaluation, LLM-as-a-judge metrics), production (monitoring)

DSPy / optimisation

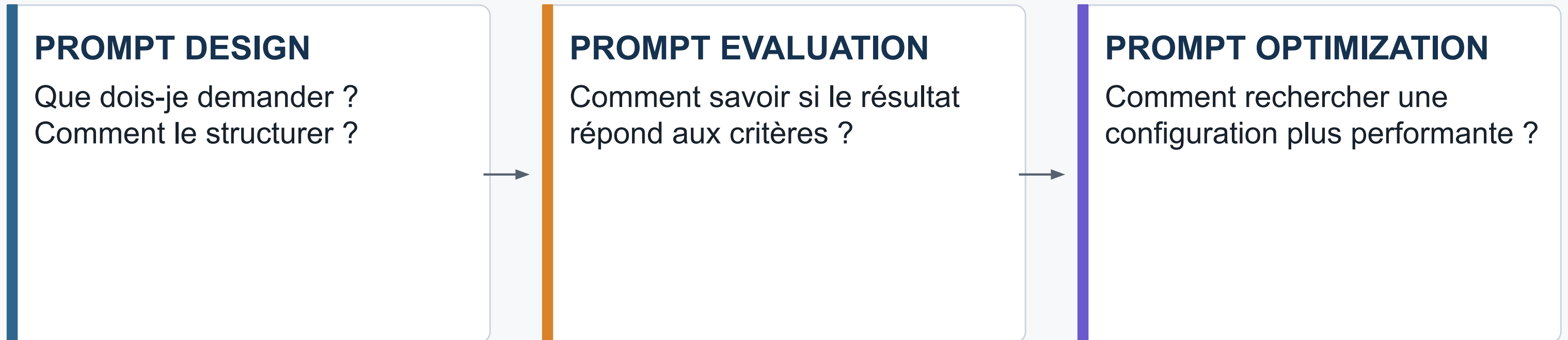
programmer des comportements LLM et expérimenter des optimisations

Trois activités à distinguer :

- PROMPT DESIGN
- PROMPT EVALUATION
- PROMPT OPTIMIZATION

Design, évaluation, optimisation : trois questions différentes

Ne pas confondre « améliorer la formulation » et « prouver que la version est meilleure ».



Une optimisation utile repose sur une évaluation : sinon on ne sait pas si l'on progresse.

04

Conclusion

Même besoin, deux approches : puis le passage au système.

Manifeste de l'ingénieur de prompt

Le langage naturel constitue désormais une nouvelle couche de programmation.

Un prompt utilisé pour piloter un système IA doit être conçu avec une rigueur comparable à celle d'un code source.

Avant : APPROCHE NAÏVE

Organise un voyage insolite en Islande pour 7 jours et 2 000 €.

Après : APPROCHE D'INGÉNIERIE

ROLE

Agent de voyage spécialisé en Islande.

CONTEXT

7 jours · 2 000 € · randonnée modérée.

TASK

3 itinéraires insolites.

CONSTRAINTS

≤ 2 000 € ; 7 jours.

OUTPUT

Tableau Markdown + total.

La seconde n'est pas « parfaite » mais elle est explicite, testable et versionnable.

Piliers de l'ingénierie du prompt

Le prompt évolue ; la démarche d'évaluation s'ajoute ensuite et boucle sur le système.



écrire → structurer → essayer → observer → corriger → mesurer → intégrer

Le passage au workflow introduit des dépendances supplémentaires : données, outils, modèles, erreurs et observabilité.

Sources principales

Documentation officielle et outils d'évaluation consultés pour les techniques citées.

OpenAI [Best practices for prompt engineering with the OpenAI API](#)
<https://help.openai.com/en/articles/6654000-comprehensive-next-generation-prompting>

Anthropic [Prompting best practices](#)
<https://platform.claude.com/docs/claude/use-xml-tags>

Google AI [Prompt design strategies — Gemini API](#)
<https://ai.google.dev/gemini-api/docs/prompting-strategies>

Microsoft Learn [Techniques d'ingénierie rapide](#)
<https://learn.microsoft.com/fr-fr/azure/foundry/openai/concepts/prompt-engineering>

Opik [Framework de développement, test et de production de prompt et d'agents IA](#)
<https://www.comet.com/docs/opik/>

Promptfoo [Getting started — prompt testing & evaluation](#)
<https://www.promptfoo.dev/docs/getting-started/>

Les définitions de prompt design / engineering sont présentées comme un cadre pédagogique : les sources emploient des terminologies qui se recouvrent sans constituer un standard unique.

Références Gemini NotebookLM

<https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-prompting-best-practices>
<https://claude.com/blog/best-practices-for-prompt-engineering>

<https://ai.google.dev/gemini-api/docs/prompting-strategies?hl=fr>
<https://cloud.google.com/discover/what-is-prompt-engineering>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/introduction-prompt-design?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/prompt-design-strategies?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/introduction-prompt-design?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/prompt-design-strategies?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/clear-instructions?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/few-shot-examples?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/contextual-information?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/system-instructions?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/structure-prompts?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/explain-reasoning?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/break-down-prompts?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/adjust-parameter-values?hl=fr>
<https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/prompt-iteration?hl=fr>

<https://help.openai.com/en/articles/4936848-how-do-i-create-a-good-prompt-for-an-ai-model>
<https://help.openai.com/en/articles/10032626-prompt-engineering-best-practices-for-chatgpt>
<https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api>

<https://docs.mistral.ai/inference/prompting>
https://docs.mistral.ai/resources/cookbooks/mistral-prompting-prompting_capabilities
<https://docs.mistral.ai/vibe/work/get-started>