



LAURENT DÉSÉCHALLIERS

CONTEXTE

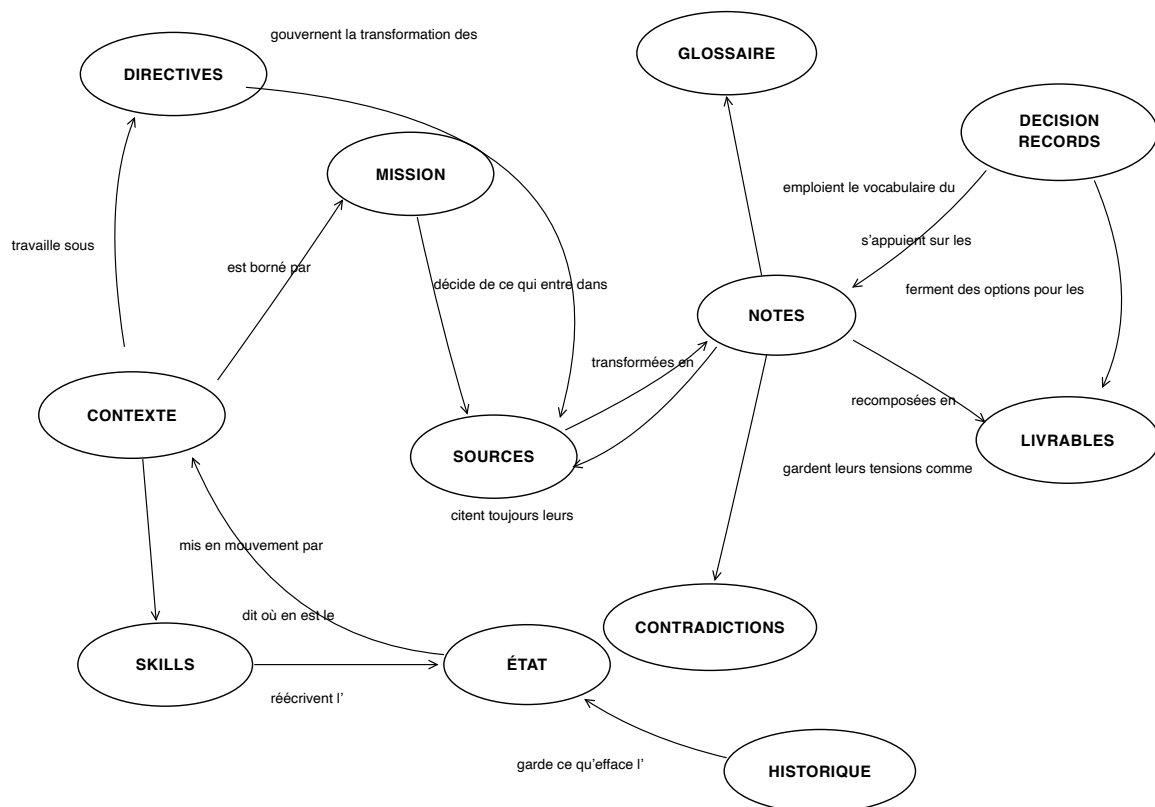
Créez un contexte qui permet à l'IA
de comprendre ce qui compte,
de relier vos connaissances
et de les mettre au service de vos buts.

Le Contexte

Comment construire un contexte qui permet à l'IA de comprendre ce qui compte, relier vos connaissances et capitaliser ce que vous apprenez au service de vos buts.

Définition du Contexte

Un contexte est un système borné par une mission humaine, dans lequel une IA organise et mobilise ce qu'on sait, d'où on le sait, comment les éléments se relient et où en est le travail, afin d'aider l'humain à atteindre son but sans repartir de zéro. Ce faisant, le système constitue et fait fructifier un capital informationnel durable : la connaissance produite reste disponible, s'enrichit dans le temps et peut être réutilisée pour d'autres objectifs.



Introduction

Nous n'avons jamais eu autant d'informations à notre disposition. Pourtant, nous consacrons encore une grande partie de notre énergie à les retrouver, à les remettre en ordre et à reconstruire ce que nous avons compris quelques jours plus tôt.

Une idée utile arrive rarement sous une forme propre. Elle naît d'un échange, se précise dans un document, rencontre une objection en réunion, produit une décision, puis devient un article, une spécification ou un projet. Nos outils conservent chacune de ces traces. Ils peinent à garder le fil qui les relie.

L'intelligence artificielle change une partie du problème. Elle sait lire, rapprocher, reformuler et transformer une matière très différente. Mais une conversation avec une IA reste un moment fragile : le sujet avance, la discussion s'allonge, puis il faut rappeler le but, les décisions prises, les sources utilisées et ce qui reste à faire.

C'est pour répondre à cette difficulté que j'ai construit le Contexte.

Un Contexte est un espace de travail consacré à un sujet. Un humain et une IA y conservent le but poursuivi, les

sources, les idées, les règles et l'état du travail, afin de produire des résultats sans devoir tout reconstruire à chaque séance.

L'IA peut alors travailler depuis une histoire réelle. Elle retrouve l'origine d'une idée, rappelle un choix, signale une contradiction, relève une preuve absente, reprend un brouillon là où il s'était arrêté. **Elle aide à organiser et à confronter la pensée. L'humain garde la responsabilité du sens et des décisions.**

Ce système est né dans mon travail quotidien, d'un seul fichier et d'une demande très simple adressée à une IA : organiser ce que j'allais lui dire. Il s'est développé par essais successifs, chaque difficulté rendant visible un manque. Il continue d'évoluer : certaines solutions présentées ici ont tenu dans plusieurs usages, d'autres restent des réponses provisoires à des problèmes que je rencontre encore.

Ce livre ne part donc pas d'une théorie achevée. Il part de problèmes vécus : cinq cents mails devenus impossibles à suivre, des idées dispersées entre plusieurs outils, des documents clients auxquels il fallait répondre sans perdre le travail déjà accompli, des textes repris après interruption.

Il s'adresse aux personnes qui transforment de l'information en compréhension, en décisions et en livrables : product managers, consultants, rédacteurs, chercheurs, analystes et, plus largement, travailleurs du savoir. Il propose une manière de faire de l'IA un partenaire de travail dans la durée, tout en conservant la maîtrise de sa matière.

Le livre est organisé en deux livres complémentaires.

Livre I — Comprendre le contexte raconte la naissance du système et ce qu'il change dans la relation avec l'IA.

Livre II — Construire et faire vivre le contexte explique comment le construire et l'utiliser dans la durée.

Ils peuvent être lus à la suite ou séparément. Le chapitre

Dans quel ordre lire ce livre présente les différents parcours possibles.

Note sur les exemples

Les réflexions et les exemples de cet ouvrage se nourrissent de plus de trente ans de pratique dans les métiers de l'informatique. Ils viennent de situations rencontrées dans des projets et des environnements professionnels variés, et de nombreux échanges avec des product managers, CTO, engineering managers... d'autres entreprises : plusieurs organisations, plusieurs périodes de mon parcours.

Pour servir le propos du livre, j'ai modifié, généralisé ou recomposé ces situations. Cas d'usage, données chiffrées et métriques illustrent des problèmes récurrents et les enseignements que j'en ai tirés. Ce sont des constructions : aucune ne décrit un client, un produit ou une organisation réels, ni ne reflète les données, les pratiques ou les positions de mon employeur.

Les exemples tirés de ma vie personnelle — la copropriété, mes articles et mes propres textes — reposent sur des situations que j'ai réellement vécues. Certains détails susceptibles d'identifier des personnes ont toutefois été modifiés.

Les analyses, méthodes et opinions exprimées dans cet ouvrage sont personnelles et n'engagent que moi.

À propos de l'auteur

Je travaille dans l'édition de logiciels depuis plusieurs décennies et j'exerce aujourd'hui comme product manager. Mon parcours m'a conduit du développement et de l'architecture à l'organisation des équipes, puis au produit, à ses opérations et à son marketing.

Ces changements de métier ont déplacé mon rapport à l'information. Il ne suffisait plus de connaître une solution technique ou de retrouver un document. Il fallait réunir des signaux dispersés, comprendre ce qu'ils disaient ensemble, arbitrer entre plusieurs directions et produire des textes adaptés à des lecteurs très différents. Mon intérêt pour l'intelligence artificielle et les systèmes distribués est ancien. Il rencontre aujourd'hui une question très concrète : **comment travailler avec une IA sans lui abandonner la maîtrise de ses informations, de ses décisions et de sa pensée ?**

J'ai contribué à la presse spécialisée et travaillé comme relecteur technique pour M2 Édition et Eyrolles. J'écris aujourd'hui sur [**malorean.net**](https://malorean.net), autour du product management, de l'IA, des méthodes de travail et des systèmes d'information.

Je contribue également à **Proposition Marketing**, un magazine indépendant consacré au marketing stratégique et opérationnel. Écrire pour des lecteurs venus du marketing et de la direction d'entreprise m'oblige à expliquer les questions posées sans supposer un vocabulaire commun. Cette exigence est aussi celle de ce livre.

J'aurais beaucoup de plaisir à échanger avec d'autres créateurs et créatrices de systèmes proches, différents ou complémentaires : le sujet mérite d'être discuté, confronté, enrichi par d'autres pratiques. Avec des universitaires travaillant sur des questions voisines aussi — j'ai passé quelques années dans deux centres de recherche CNRS, et renouer avec ce milieu serait un plaisir.

Laurent Deséchalliers

- **laurent@malorean.net**
- **<https://malorean.net>**

Remerciements

Mes premiers remerciements vont à vous.

Vous qui avez eu l'idée un peu folle d'ouvrir ce livre — un livre légèrement plus épais que prévu, qui contient sans doute quelques redites et porte sur un sujet abscons.

J'ai jusqu'ici travaillé sur le Contexte de manière assez solitaire. J'espère pourtant que cette solitude ne durera pas : que le sujet donnera envie à d'autres de l'explorer, de le discuter, de le critiquer et de le faire évoluer. Peut-être que les prochains articles publiés sur **malorean.net** seront alors l'occasion de remercier de nombreux contributeurs et contributrices.

Je remercie surtout ma femme, Christine.

Elle a vu les soirées et les week-ends se prolonger devant ce livre. Elle m'a soutenu tout au long de son écriture, tout en sachant me rappeler qu'il était tard et qu'il était vraiment temps de fermer ce foutu ordinateur.

Promis, ce n'était qu'une parenthèse.

Et, bien au-delà de ce livre, merci de m'accompagner depuis toutes ces années dans cette belle existence que nous partageons tous les deux.

Dans quel ordre lire ce livre

Le livre suit l'ordre dans lequel les choses me sont arrivées : un problème vécu, une solution bricolée, un système enfin. La description complète du Contexte n'arrive donc qu'au Livre II.

Le parcours « comprendre » est cet ordre-là. Tu lis de bout en bout, et chaque pièce du système arrive au moment où le manque qui l'a produite vient d'être montré.

Le parcours « construire » s'adresse à qui a déjà vécu le problème et veut la mécanique, pas le récit.

1. **Définition du Contexte**, quelques lignes pour savoir de quoi on parle.
2. **La partie 5 — Sous le capot d'un Contexte**, dans l'ordre de ses six chapitres : la carte, ouvrir, alimenter, transformer, produire et décider, sauvegarder.
3. **La partie 6 — Le Contexte dans son écosystème**, pour relier ce que tu viens de construire au reste de ton système d'information.
4. **Le Livre I**, quand un choix de conception te paraît arbitraire : tu y trouveras le problème qui l'a provoqué.

Ce raccourci coûte quelque chose. Le Livre II a été écrit pour un lecteur qui a déjà les situations du Livre I en tête, et il y fait référence. Tu croieras donc des exemples qui te manqueront un peu de contexte — c'est le prix, et il reste payant si tu sais déjà pourquoi tu es là.

Essayer avant de lire

Ce livre décrit une méthode, et une méthode se raconte mal. Tu peux en voir une tourner avant d'en lire la théorie, ou pendant — c'est souvent le meilleur moment pour comprendre à quoi sert une pièce.

Trois dépôts sont ouverts sur [**github.com/laurent-malorean**](https://github.com/laurent-malorean).

Contexte Nano Notion est le plus rapide à installer. Il ne demande ni environnement de développement ni compétence technique : un ChatGPT et un Notion suffisent. ChatGPT porte l'intelligence du Contexte, Notion lui sert de mémoire — c'est là que les sources, les notes, les décisions et l'état s'écrivent et se relisent d'une conversation à l'autre. En quelques minutes, tu as un Contexte qui tient.

Contexte V2 est l'implémentation complète, celle que j'utilise tous les jours et qui a servi à écrire ce livre. Elle s'installe avec Claude Code ou avec Codex, l'un ou l'autre au choix, et demande un peu plus de mise en place que la précédente.

LLM Wiki n'est pas un Contexte. C'est une bibliothèque de connaissances construite à partir de documents — livres,

normes, articles, cours — dont l'IA tire des fiches courtes et reliées entre elles, que l'on interroge ensuite en langage courant. Le Contexte garde le fil d'une mission ; le LLM Wiki garde massivement ce qu'on a lu. La partie 6 explique comment les deux travaillent ensemble.

Rien n'oblige à commencer par là. Mais si à un moment la lecture te paraît abstraite, c'est le chemin le plus court pour que ça redevienne concret.

Du même auteur



Malorean Corp — Hier, il fallait parfois des semaines pour transformer une idée en prototype. Aujourd’hui, une personne aidée par l’intelligence artificielle peut y parvenir en quelques jours. Ce changement est spectaculaire. Mais le véritable bouleversement pourrait bien se jouer ailleurs : dans la façon dont nous travaillons ensemble.

Sortie prévue : fin novembre 2026.

Plus d’informations sur [**https://malorean.net**](https://malorean.net).

Sommaire

LIVRE I

Comprendre le Contexte

<u>Partie 1 — La naissance d'un besoin</u>	<u>21</u>
<u>1.1 Cinq cents mails et une histoire qui disparaît</u>	<u>22</u>
<u>1.2 Une première définition</u>	<u>33</u>
<u>Partie 2 — Ce que le Contexte change</u>	<u>43</u>
<u>2.1 De la conversation au travail reprenable</u>	<u>44</u>
<u>2.2 Deux clients, un livre blanc et une contradiction à</u>	
<u>conserver</u>	<u>53</u>
<u>Partie 3 — Trois usages concrets</u>	<u>65</u>
<u>3.1 Répondre à un cahier des charges sans perdre le travail</u>	
<u>accompli</u>	<u>66</u>
<u>3.2 Écrire un article à partir d'une pensée encore informe</u>	
<u>74</u>	
<u>3.3 Faire mûrir un sujet produit au milieu du chaos</u>	<u>84</u>
<u>Partie 4 — Ce que les livrables laissent derrière eux</u>	<u>95</u>
<u>4.1 La réponse tient dans une cellule, pas le raisonnement</u>	
<u>96</u>	
<u>4.2 Six mois plus tard, la réponse ne suffit plus</u>	<u>100</u>

LIVRE II**Construire et faire vivre le Contexte****Partie 5 — Sous le capot d'un Contexte** **107****5.1 La carte complète** **108****5.2 Ouvrir : retrouver la mission et le présent** **120****5.3 Alimenter : faire entrer la matière** **132****5.4 Transformer : passer des sources aux idées** **139****5.5 Produire et décider** **150****5.6 Sauvegarder et reprendre** **161****Partie 6 — Le Contexte dans son écosystème** **171****6.1 Interroger les connaissances structurées** **172****6.2 Interroger le produit réel** **180****6.3 Le jour où j'ai eu trois Contextes** **191****6.4 Lire chez le voisin sans déménager chez lui** **198****6.5 Ce que la circulation coûte** **207****Avant de refermer****Clôture** **215****7.1 Épilogue** **216****7.2 Parentés et voisinages** **219**

LIVRE I

Comprendre le Contexte

- 1 La naissance d'un besoin
- 2 Ce que le Contexte change
- 3 Trois usages concrets
- 4 Ce que les livrables laissent derrière eux

Partie 1

La naissance d'un besoin

Le Contexte n'est pas né d'une idée sur l'intelligence artificielle. Il est né d'une boîte mail devenue impraticable, et d'un constat qui a mis du temps à se former : le problème n'était pas le nombre de messages.

Cette partie raconte cette origine, puis en tire une première définition — assez simple pour être retenue, assez complète pour servir de repère au reste du livre.

1.1

Cinq cents mails et une histoire qui disparaît

La première fois que j'ai créé un Contexte, je ne cherchais pas à inventer une méthode de travail avec l'intelligence artificielle. J'essayais simplement de ne plus perdre le fil de ma copropriété.

Je siège au conseil syndical. À ce titre, je reçois les messages du syndic, des copropriétaires, des locataires et des prestataires :

- un mail peut annoncer un nouveau problème,
- le suivant précise un devis,
- un troisième reporte une intervention,
- une semaine plus tard, quelqu'un répond dans le même fil pour parler d'un autre bâtiment.

Entre-temps, trois autres sujets ont avancé.

Le volume n'était pas le seul problème. Environ cinq cents mails se répartissaient entre une quinzaine de sujets :

- travaux,
- finances,
- assurances,

- assemblées générales,
- débiteurs,
- appels d'offres,
- incidents et demandes diverses
- ...

Chacun avançait à son propre rythme, parfois pendant plusieurs mois.

La personne qui suit un chantier au quotidien garde naturellement son histoire en tête. Elle sait quel prestataire est intervenu, pourquoi le calendrier a changé et quelle décision attend encore une réponse. Lorsque j'ouvrais un nouveau message, je ne disposais pas toujours de cette continuité. Je devais remonter le fil, retrouver les échanges précédents et reconstituer la situation avant même de pouvoir comprendre l'information nouvelle.

Je voulais pouvoir poser des questions simples :

- où en est la toiture ?
- quel prestataire a envoyé le dernier devis pour les gouttières ?
- qu'avons-nous décidé à propos de cette infiltration ?
- quelle action reste attendue du syndic ?

Ma boîte mail contenait les réponses. Elle ne me donnait pas la vision d'ensemble.

Un fichier et une phrase

J'ai commencé avec le dispositif le plus simple possible : un fichier et une consigne donnée à une IA.

| *Agrège les informations et découpe-les en thématiques.*

Je lui transmettais les mails. Elle les associait à des sujets et maintenait un résumé pour chacun.

- Les informations sur les travaux se retrouvaient ensemble.
- Les questions financières formaient un autre ensemble.
- Les prestataires, les assurances et les assemblées générales trouvaient progressivement leur place.

Ce premier essai apportait déjà quelque chose. Je pouvais demander où en était un sujet sans relire tous les messages qui le concernaient. L'IA ne cherchait plus seulement une occurrence dans la boîte mail : elle me présentait ce qu'elle avait compris de la situation.

Le fichier se réorganisait à mesure que la matière arrivait.

- Une rubrique apparaissait lorsqu'un sujet nouveau prenait de l'importance.

- Deux rubriques se séparaient lorsqu'elles commençaient à mélanger des réalités différentes.

L'information ne faisait plus que s'accumuler. Elle prenait une forme qui permettait de travailler avec elle.

Puis ce premier système a produit une réponse exacte et pourtant insuffisante.

Le chantier était terminé, mais son histoire avait disparu

Lorsqu'un chantier arrivait à son terme, son résumé indiquait : « chantier terminé, réception conforme ».

C'était bien son état présent. C'était aussi une perte.

- Quand le chantier avait-il commencé ?
- À quel moment avait-il été retardé ?
- Pourquoi le calendrier avait-il changé ?
- Quand l'avenant avait-il été signé ?

Toutes ces étapes avaient compté. La dernière mise à jour les avait effacées au profit de la situation finale.

J'avais demandé un résumé. L'IA l'avait tenu à jour en remplaçant l'ancien état par le nouveau. Elle avait suivi la consigne, mais la consigne ne protégeait pas l'histoire.

Il me fallait donc deux vues du même sujet.

- La première devait répondre rapidement à la question « où en sommes-nous ? ».
- La seconde devait conserver les événements dans leur ordre : début du chantier, retard, nouvel accord, intervention, réception.
- L'état présent permettait d'agir.
- L'histoire permettait de comprendre comment cet état avait été atteint et de vérifier un point en cas de doute.

Cette distinction est née d'un défaut très concret. Je la prenais pour une amélioration de rangement. Elle est devenue l'un des principes les plus importants de mon travail avec l'IA : **ce qui est vrai maintenant et ce qui s'est passé ne remplissent pas la même fonction.**

L'IA ne pouvait pas deviner mes règles

Un autre problème est apparu avec les noms :

- deux prestataires portaient le même nom de famille.
 - Pour les distinguer, il fallait comprendre le sujet du mail.
- une copropriétaire apparaissait tantôt sous son nom de naissance, tantôt sous son nom marital.

Sans indication particulière, l'IA pouvait créer deux personnes là où il n'y en avait qu'une.

D'autres différences tenaient au métier.

- Les débiteurs de moins de cent euros n'étaient pas suivis comme les autres.
- Cette limite ne venait d'aucune règle universelle que l'IA aurait pu connaître.
- Elle venait de notre manière de travailler.

J'ai commencé à formuler ces connaissances comme des règles durables, écrites en toutes lettres dans le fichier :

```
# Règles de traitement

- Martin cité à propos de la toiture désigne le couvreur ;
  Martin cité à propos des
  espaces verts désigne le paysagiste.
- Dupont-Lemoine et Lemoine désignent la même
  copropriétaire, lot 14.
- Présenter les débiteurs de moins de cent euros dans une
  liste séparée.
```

Rien de plus sophistiqué que cela. Trois phrases que j'aurais dû redire à chaque conversation, et que je n'ai plus eu à répéter une seule fois.

Chaque règle venait d'une erreur observée ou d'une ambiguïté réelle. Elle capturait une connaissance que je portais dans ma tête et que l'IA ne pouvait pas reconstruire avec certitude.

Le système commençait ainsi à conserver deux matières différentes :

- les informations sur la copropriété,

- la manière dont je voulais qu'elles soient traitées.

1

Le fichier qui aidait a commencé à gêner

Au fil des semaines, le fichier a grandi. Il a dépassé cinq cents lignes, puis huit cents, puis mille.

Cette croissance témoignait de son utilité :

- les sujets étaient mieux suivis,
- les prestataires identifiés,
- les décisions retrouvées.

Mais ce qui avait rendu le système possible commençait à limiter son usage.

- Trop d'informations différentes partageaient le même espace.
- Les règles se mêlaient aux faits.
- Les finances côtoyaient les travaux, les personnes, les assurances et les appels d'offres.

Retrouver la bonne matière demandait à nouveau un effort.

J'ai d'abord séparé les règles de traitement. Elles ne décrivaient pas la copropriété ; elles indiquaient comment travailler avec les informations qui la concernaient. Les isoler permettait de les retrouver et de les appliquer plus facilement.

J'ai ensuite réparti la matière entre plusieurs thèmes. Les travaux ont obtenu leur propre espace, comme les finances, les prestataires, les décisions d'assemblée générale ou les assurances. Une vue d'ensemble indiquait ce que contenait chacun. L'IA pouvait alors ouvrir le sujet utile à la question posée, au lieu de parcourir toute la matière à chaque fois.

Enfin, j'ai regroupé le vocabulaire particulier : termes juridiques, acronymes et expressions propres à la copropriété. Le but n'était pas de construire une encyclopédie. Il était de donner un sens stable aux mots dont dépendait la compréhension des échanges.

Je n'avais toujours pas de grande architecture en tête. J'ajoutais une pièce lorsqu'un problème rendait cette pièce nécessaire.

Faire confiance sans perdre la source

L'IA commettait parfois des erreurs. Elle pouvait rapprocher deux informations qui ne parlaient pas exactement du même sujet ou choisir le mauvais prestataire entre deux homonymes. Le système devait donc accélérer la compréhension sans devenir la seule version disponible des faits.

J'ai conservé tous les mails d'origine. Lorsque la réponse demandait une vérification ou que l'enjeu devenait important, je pouvais revenir à la matière reçue. Le résumé donnait un accès rapide. La source permettait de contrôler.

Ce choix rendait l'usage possible. Je pouvais accepter une approximation occasionnelle dans l'organisation des mails, parce que la matière d'origine restait accessible. Le même niveau de confiance ne conviendrait pas à un document bancaire, à une obligation réglementaire ou à une décision irréversible. Le risque dépend toujours de ce que l'on confie au système et de la possibilité de vérifier.

Le Contexte ne remplaçait donc pas ma boîte mail. Il transformait son contenu en une mémoire de travail, tout en me laissant une porte de retour vers les messages originaux.

Ce qui était apparu

En quelques semaines, le point de départ — un fichier et une phrase — était devenu un ensemble capable de répondre à plusieurs besoins : montrer l'état présent, conserver l'histoire, appliquer des règles propres à la

copropriété, distinguer les sujets, stabiliser le vocabulaire et revenir aux sources.

Quand un nouveau mail arrivait, je pouvais m'intéresser à ce qu'il changeait au lieu de reconstruire tout ce qui l'avait précédé. Quand une question surgissait, je pouvais la poser dans les termes du problème plutôt que chercher le bon message dans le bon fil. Quand l'IA se trompait, je pouvais corriger une règle ou revenir au texte d'origine.

Je n'avais pas conçu le Contexte d'un seul bloc. Je l'avais vu apparaître à travers une succession d'insuffisances : le résumé effaçait l'histoire, les noms se confondaient, les règles se perdaient, le fichier devenait trop grand.

Chaque correction rendait le travail plus durable et plus facile à reprendre.

Cette origine compte. Le Contexte n'est pas né d'une volonté de classer davantage. Il est né du besoin de comprendre une matière qui change, de conserver ce qui permet de la vérifier et de poursuivre un travail sans repartir de zéro.

À RETENIR — DEUX VUES DU MÊME SUJET

L'état répond à « où en sommes-nous ? ». L'histoire répond à « comment y sommes-nous arrivés ? ». Un résumé tenu à jour en écrasant le précédent conserve la première et perd la seconde. Les deux sont nécessaires : l'une pour agir, l'autre pour comprendre et vérifier.

Le système né de ces cinq cents mails peut maintenant recevoir une première définition.

Un Contexte est un espace de travail consacré à un sujet, où un humain et une IA conservent ce qui leur permet de comprendre, de décider et de produire dans la durée.

Cette définition reste volontairement simple. Le Contexte décrit dans ce livre possède une organisation précise, que la seconde moitié détaillera. Pour comprendre ce qu'il apporte, six repères suffisent au départ :

- un but,
- une matière,
- des idées,
- un état du travail,
- des règles propres au travail,
- des résultats à produire.

Un sujet et un but

Le Contexte de la copropriété ne cherchait pas à contenir tout ce que je savais. Il poursuivait un but : comprendre les sujets en cours, conserver leur histoire et retrouver

les informations nécessaires pour agir au conseil syndical.

Ce but formait une frontière. Un mail sur la toiture avait sa place. Une lecture intéressante sans rapport avec la copropriété n'en avait pas. Cette limite évitait au système de devenir une réserve où toute information jugée intéressante viendrait s'accumuler.

Un Contexte commence donc par une intention. Il peut servir à :

- écrire un livre,
- explorer un problème produit,
- préparer une décision,
- répondre à des cahiers des charges,
- suivre une entité métier,
- ...

Une matière dont on garde l'origine

La matière entre dans un Contexte par deux portes.

La première reçoit ce qui existe déjà avant la séance de travail :

- un mail,
- un article,
- un livre,
- une page web,

- la transcription d'un entretien
- ...

Ce sont les **sources documentaires**. Elles arrivent comme des objets déjà constitués et restent stables.

Dans le cas de la copropriété, le mail annonçant le report d'une intervention appartient à cette première catégorie.

La seconde reçoit ce qui se formule pendant le travail avec l'IA. Lorsque j'explique une situation, corrige une interprétation, précise une nuance ou formule une conviction issue de mon expérience, cette parole devient une **source de session**. Elle se construit progressivement au cours de l'échange. Lorsque j'explique que deux noms désignent la même copropriétaire, mon propos entre d'abord par cette seconde porte.

La différence tient donc au moment et à la manière dont la matière apparaît.

- Le document existe avant la séance et y est apporté.
- La source de session naît pendant la séance et grandit avec elle.

Ces deux sources remplissent ensuite la même fonction : conserver la matière avant son interprétation.

- Elles ne garantissent pas que tout ce qu'elles contiennent est vrai.
- Elles permettent de retrouver ce qui a été reçu ou formulé, puis d'en apprécier la fiabilité.

Le Contexte peut ensuite tirer de cette matière une idée, une règle ou un état du travail. Cette connaissance transformée occupe une autre place. Si une date semble fausse, si deux personnes ont été confondues ou si une interprétation devient contestable, on peut ainsi revenir à son origine.

Des idées rendues utilisables

Accumuler les sources ne suffit pas. Cinq cents mails bien conservés restent cinq cents mails à relire.

Le Contexte transforme donc progressivement cette matière. Il fait apparaître :

- des sujets,
- des faits importants,
- des règles,
- des distinctions,
- des questions ouvertes,
- des relations.

Un échange isolé rejoint l'histoire du chantier auquel il appartient. Plusieurs mentions d'un même prestataire

enrichissent une connaissance commune. Une correction empêche une confusion de se reproduire.

Cette étape intermédiaire compte parce qu'elle rend l'information utilisable.

- La matière reçue garde la trace.
- La connaissance organisée permet de comprendre et d'agir.

Dans des travaux plus complexes, certaines idées pourront servir plusieurs fois : une même connaissance produit nourrit une décision, une spécification et une documentation ; un même fait validé aide à répondre à plusieurs questionnaires clients. Le résultat change, tandis que l'idée et sa provenance restent disponibles.

Un état qui permet de continuer

Un travail vivant possède toujours un présent : ce que l'on sait, ce qui reste incertain, ce qui a été décidé et ce qui doit arriver ensuite.

Dans la copropriété, l'état répondait à une question immédiate : où en est ce chantier aujourd'hui ? Son histoire répondait à une autre question : comment en sommes-nous arrivés là ? Le système avait besoin des deux pour permettre une reprise réelle.

Cette distinction dépasse le suivi d'un chantier. Lorsque j'interromps l'écriture d'un texte ou l'exploration d'un problème produit, je veux retrouver plus qu'une collection de documents. Je veux savoir :

- ce qui est acquis,
- quel point reste fragile,
- quelle action peut relancer le travail.

Le Contexte conserve cette continuité en dehors de la conversation.

- La séance avec l'IA peut s'arrêter.
- Le travail garde un état auquel la séance suivante pourra revenir.

Des règles propres au travail

L'IA dispose de connaissances générales. Elle ignore en revanche une grande partie de ce qui rend une situation particulière :

- les deux prestataires qui portent le même nom,
- la manière dont une équipe qualifie un signal client,
- le niveau de preuve attendu avant de publier une affirmation
- le ton choisi pour un livre,
- ...

Le Contexte garde ces règles avec le travail qu'elles gouvernent. Elles peuvent être corrigées lorsque l'expérience montre leur limite. Elles s'appliquent ensuite sans que l'humain doive les reformuler dans chaque demande.

Ces règles ne donnent pas à l'IA la responsabilité du travail. Elles rendent explicite ce que l'humain attend d'elle. L'humain fixe le but, apporte ou valide la matière, corrige les interprétations et prend les décisions. L'IA absorbe, rapproche, organise, questionne, reformule et prépare des résultats à partir de ce cadre.

Des résultats adaptés à un besoin

Le Contexte reste orienté vers ce qu'il doit permettre de faire.

Dans la copropriété, le résultat pouvait être :

- une réponse à une question,
- un état des travaux,
- la préparation d'une discussion.

Dans d'autres usages, ce sera :

- un article,
- un cahier des charges,
- une documentation.

Le même espace peut produire plusieurs résultats parce qu'il conserve la matière avant sa mise en forme finale.

Les sources ne sont pas enfermées dans le premier document qui les utilise. Les idées restent disponibles pour une autre question ou un autre public.

Cette capacité distingue le `Contexte` d'une simple archive.

- Une archive permet de conserver.
- Le `Contexte` organise la reprise et la transformation de ce qui a été conservé.

Une forme qui peut commencer très simplement

Dans ma pratique, un `Contexte` prend la forme de fichiers que je peux lire, déplacer et conserver indépendamment de l'outil qui m'aide à les travailler. Cette forme répond à mon besoin de maîtrise et de durée. D'autres mises en œuvre sont possibles si elles préservent les mêmes fonctions.

Le premier `Contexte` ne possédait pourtant rien de sophistiqué. Il tenait dans un fichier. Il a grandi lorsque ses limites sont devenues visibles. Cette progression reste importante : on n'a pas besoin de construire tout le système décrit dans ce livre pour commencer à en éprouver l'idée.

Un sujet, un but et une première matière suffisent.

- L'IA aide à les organiser.
- L'humain observe ce qui manque, ajoute une règle, conserve une source ou sépare deux informations qui n'ont pas le même rôle.
- L'architecture apparaît au rythme des problèmes rencontrés.

Le `Contexte` désigne donc moins une arborescence particulière qu'une relation de travail durable entre un humain, une IA et une matière orientée vers un but.

La partie suivante va examiner ce que cette relation change. Une conversation avec une IA peut déjà être féconde. Le passage au `Contexte` commence lorsque cette conversation s'inscrit dans un environnement capable de garder son but, son histoire et son état au-delà du moment présent.

À RETENIR — LES SIX REPÈRES D'UN `CONTEXT`

Un but qui trace une frontière, une matière dont on garde l'origine, des idées rendues utilisables, un état qui permet de continuer, des règles propres au travail, des résultats à produire. Ces six repères suffisent pour commencer.

L'architecture vient ensuite, quand un problème rencontré la rend nécessaire.

Partie 2

Ce que le Contexte change

Une conversation avec une IA produit déjà beaucoup. Elle explique, reformule, propose, et donne souvent une réponse meilleure que celle qu'on aurait trouvée seul.

Cette partie cherche donc ce qu'un Contexte ajoute à cela. La réponse ne tient pas à la qualité des réponses obtenues, mais à ce qui survit entre deux séances : un travail qui reprend où il s'était arrêté, et des documents qui gardent leurs désaccords au lieu de les lisser.

De la conversation au travail reprenable

Une conversation avec une IA peut commencer très simplement.

Je lui donne une idée encore confuse. Elle la reformule. Je demande un plan, puis un autre. Elle me propose une objection à laquelle je n'avais pas pensé. Je précise mon intention, je refuse une piste, j'en approfondis une autre. En quelques minutes, la discussion fait avancer ce qui serait peut-être resté longtemps devant une page blanche.

C'est l'une des grandes forces du chat : une question suffit pour commencer. Cette forme convient très bien à une tâche brève. Je demande une synthèse, une explication, quelques possibilités. La réponse arrive, je l'utilise, puis la conversation peut s'arrêter.

Le rapport change lorsque le travail dure.

Cinq phrases avant de pouvoir écrire

Je rouvre une conversation vieille de six jours pour reprendre un texte commencé la semaine précédente. Avant de pouvoir écrire une seule ligne, je tape ceci :

Reprends le texte sur la qualité des comptes rendus. Il s'adresse à des responsables d'exploitation, pas à des développeurs. Nous avons écarté l'idée de raccourcir le formulaire, il est trop tôt pour trancher. Le chiffre sur les audits vient d'un seul entretien, traite-le comme une hypothèse. Nous en étions à la troisième section.

2

Cinq phrases pour remettre l'IA dans un travail qu'elle avait déjà devant elle six jours plus tôt.

Tout ce que je viens de taper existait avant que je le tape : l'angle, le public, la décision, la fragilité et la position dans le texte se trouvaient dans la conversation précédente. Je les ai simplement extraits de ma propre mémoire, parce que la conversation les avait rangés dans l'ordre où ils avaient été dits.

Le travail, lui, suit une autre organisation.

- Une consigne donnée au début doit encore s'appliquer à la fin.
- Une source ajoutée hier doit pouvoir soutenir un paragraphe écrit demain.
- Une décision ancienne doit empêcher de rouvrir la même discussion.
- Un point en attente doit réapparaître au bon moment.

Plus la conversation grandit, plus l'humain doit maintenir cette organisation dans sa tête. Lorsqu'une nouvelle séance commence, il reconstruit la situation pour l'IA : voici le but, voici ce qui a changé, voici le ton, voici les décisions, voici où nous en étions.

Le chat reste utile. La charge de continuité repose pourtant sur celui qui l'utilise.

Donner un lieu à ce qui doit durer

Un Contexte change la relation en donnant au travail un espace distinct de la conversation.

La discussion reste le lieu de l'échange. Elle permet de questionner, d'explorer, de préciser et de corriger. Ce qui doit durer rejoint l'espace de travail :

- le but,
- les sources,
- les règles,
- les idées stabilisées,
- les décisions,
- les textes,
- l'état d'avancement.

La différence peut sembler mince. Elle est comparable à celle qui sépare une réunion du projet dont elle parle.

Une bonne réunion peut faire progresser le projet, mais

le projet ne tient pas tout entier dans la mémoire de la réunion. Les documents, les décisions et les actions continuent d'exister après sa fin.

Conserver la transcription complète du chat ne produirait pas le même effet. Elle est précieuse lorsqu'on veut retrouver exactement ce qui a été dit, mais elle suit la forme de la conversation : une question, une réponse, une correction, une digression, un retour au sujet. Pour reprendre le travail, il faudrait encore relire cette suite et décider ce qui compte maintenant.

Le Contexte, lui, transforme ce qui s'est dit.

- Une correction rejoint l'endroit où elle pourra empêcher le retour de l'erreur.
- Une décision reste visible comme décision.
- Une source demeure liée à ce qu'elle soutient.
- Un texte existe dans sa dernière version complète.

La mémoire devient ainsi orientée vers l'action. Elle ne cherche pas seulement à conserver le passé. Elle prépare la prochaine intervention.

Une page pour retrouver trente-deux textes

J'ai mesuré ce que cela change en reprenant le travail sur ce livre.

Le manuscrit comptait alors trente-deux textes. Certaines parties étaient rédigées, d'autres encore vides. Une section avait été divisée parce qu'elle occupait une place disproportionnée. Deux passages attendaient ma relecture. Une affirmation ne reposait sur aucun cas vécu et devait être reprise.

Pris séparément, aucun de ces éléments n'est difficile à comprendre. Leur ensemble forme pourtant une situation qu'il faut reconstruire avant de pouvoir écrire une seule ligne utile.

Je pourrais ouvrir les trente-deux textes, relire le plan, parcourir mes anciennes conversations et chercher les remarques laissées en attente. Ce travail d'archéologie prendrait du temps et laisserait subsister un doute : ai-je oublié le point qui devait précisément empêcher une erreur ?

À la place, j'ouvre le Contexte du livre. Il me tend un fichier de reprise, qui disait ce matin-là à peu près ceci :

Reprise

Où on en est

Trente-deux textes existent, du plan à la conclusion.

Les parties techniques sont rédigées ; les deux premières restent trop théoriques.

Ce qu'il faut savoir avant de continuer

Le lecteur visé découvre le sujet : il perd pied dès qu'un concept arrive avant un cas vécu.

L'affirmation sur l'échec des systèmes documentaires ne repose encore sur aucune expérience personnelle.

La prochaine action

Réécrire l'ouverture autour du cas des cinq cents mails de copropriété.

Ce qui est resté ouvert

Deux passages attendent ma relecture.

Plusieurs sources sont entrées sans avoir été transformées en notes.

2

Une page, lue en moins d'une minute. Je sais quoi faire, et je n'ai ouvert aucun des trente-deux textes.

Continuer n'est pas reprendre

Continuer consiste à connaître l'action suivante.

Reprendre consiste à retrouver le sens du travail avant d'effectuer cette action.

Une liste de tâches peut m'indiquer qu'une source doit être traitée. Elle ne m'explique pas pourquoi cette source est prioritaire, quelle partie du livre elle doit éclairer ni quel risque elle doit permettre de corriger.

Pour reprendre, j'ai besoin d'un ensemble plus riche :

- le but que poursuit le travail ;
- l'endroit où il est réellement arrivé ;
- les décisions qui orientent encore la suite ;
- les fragilités qui ne doivent pas être oubliées ;
- une prochaine action assez précise pour recommencer.

Le fichier de reprise se lit d'ailleurs dans cet ordre.

« Trente-deux textes » donne une mesure. « Il perd pied dès qu'un concept arrive avant un cas vécu » donne une direction. La prochaine action découle des deux, et nomme le cas précis qui doit ouvrir le livre.

La reprise a reconstitué un raisonnement, pas seulement une position dans une liste.

Elle ne cherche pas pour autant à recréer dans ma tête chaque détail de la séance précédente. Un résumé trop court efface la raison d'un choix. Un rapport trop long repousse le moment où le travail recommence.

Une interruption ne choisit pas son moment

Au début, j'imaginai la reprise comme la conséquence d'une fermeture propre. Une séance se terminerait, son état serait soigneusement résumé et la prochaine action serait notée.

La pratique suit rarement ce scénario.

- Une réunion commence.

- La batterie se vide.
- La journée s'arrête plus vite que prévu.
- Une autre urgence prend la place du sujet.

Le moment où la mémoire de reprise devient la plus utile est justement celui où l'on n'a pas préparé la fin.

Le Contexte entretient donc sa capacité de reprise pendant le travail. Les décisions trouvent leur place lorsqu'elles sont prises, les tâches lorsqu'elles sont formulées, les sources lorsqu'elles entrent.

La reprise devient une propriété du système plutôt qu'une discipline héroïque de l'utilisateur.

L'humain garde pourtant un rôle décisif. Il peut juger que la prochaine action proposée n'est plus prioritaire, qu'une ancienne décision doit être rouverte ou qu'un fait nouveau change le but. La mémoire ne fige pas la trajectoire. Elle rend visible la trajectoire depuis laquelle on décide de la modifier.

Les cinq phrases que je retapais en rouvrant ma conversation, je ne les retape plus. Elles sont écrites quelque part, et quelqu'un d'autre que moi les relit à ma place au début de chaque séance.

À RETENIR — LE CHAT, L'ENVIRONNEMENT ET LA REPRISE

Le chat est le lieu de l'échange ; le Contexte est le lieu de ce qui doit durer. Une conversation range ses informations dans l'ordre où elles ont été dites ; un travail en demande un autre. Continuer, c'est connaître l'action suivante ; reprendre, c'est retrouver le sens du travail avant de l'effectuer. Une interruption ne prévient pas : la capacité de reprise s'entretient pendant le travail, pas au moment de le fermer.

Deux clients, un livre blanc et une contradiction à conserver

Jusqu'ici, j'ai décrit le Contexte et plusieurs effets qu'il produit. Mais à quoi ressemble-t-il réellement lorsqu'on l'ouvre ? Que voit-on dans les fichiers ? Comment plusieurs sources font-elles apparaître des idées qui peuvent être justes séparément et pourtant difficiles à concilier ?

Le sujet de ce Contexte fictif est simple : comprendre pourquoi certains techniciens produisent des comptes rendus d'intervention incomplets, avant de décider si le produit doit évoluer.

Sa mission tient en quelques lignes :

Comprendre ce qui dégrade la qualité des comptes rendus d'intervention. Distinguer les difficultés de saisie, les contraintes d'organisation et les besoins d'exploitation des données. Ne pas choisir de solution avant d'avoir confronté plusieurs sources.

Cette mission donne une direction au travail. Elle demande de comprendre un problème, pas encore de concevoir une fonctionnalité.

2

Un premier retour fait apparaître une idée

Le premier document versé au Contexte est la transcription d'un entretien avec une responsable d'exploitation.

Elle explique :

« Les techniciens remplissent souvent leurs comptes rendus le vendredi. Sur le terrain, le formulaire leur paraît trop long. À la fin de la semaine, ils ont oublié une partie des détails. »

L'IA conserve cet entretien comme **une source**. Elle en extrait ensuite une idée dans une fiche séparée, ce que j'appelle une **note atomique** :

Une saisie trop contraignante peut conduire certains techniciens à différer leur compte rendu et à perdre des détails.

Cette formulation ne dit pas que tous les techniciens agissent ainsi ni que la longueur du formulaire constitue l'unique cause. Elle garde le centre du témoignage : dans l'organisation décrite, la saisie est reportée et une partie

de l'information disparaît entre l'intervention et sa rédaction.

La source reste disponible. La note ne la remplace pas ; elle rend son idée principale plus facile à retrouver et à confronter à la suite.

Le deuxième retour enrichit la même note

Quelques semaines plus tard, un deuxième entretien entre dans le Contexte.

Cette fois, un responsable d'exploitation raconte :

« Nous avons rendu douze champs obligatoires pour améliorer la qualité des données. Maintenant, certains techniciens écrivent "RAS" ou mettent un tiret uniquement pour pouvoir valider. »

Ce témoignage ne décrit pas exactement le même comportement. Les techniciens du premier entretien reportent leur saisie. Ceux du second remplissent le formulaire, mais contournent son exigence.

Les deux sources portent néanmoins sur la même idée centrale : une contrainte destinée à obtenir davantage d'informations peut finir par réduire la qualité réelle de ce qui est saisi.

Le second entretien rejoint donc la première **note atomique**. Il ne remplace pas la première source et ne

crée pas une idée concurrente. Il enrichit la note en faisant apparaître une autre forme du même problème : après la saisie différée, la complétude de façade.

2

La note peut maintenant affirmer :

Une saisie trop contraignante peut conduire certains techniciens à différer ou contourner le compte rendu, au détriment de l'information recueillie.

Deux sources sont référencées. Chacune garde son rôle et ses limites. La note devient plus solide sans prétendre que deux témoignages suffisent à mesurer l'ampleur du phénomène.

Le livre blanc fait apparaître une autre idée

J'ajoute ensuite un livre blanc consacré à l'exploitation des données d'intervention.

Le document défend une autre exigence. Pour comprendre les pannes récurrentes, comparer les équipements et préparer certains audits, une entreprise a besoin de comptes rendus suffisamment structurés. Si chaque technicien écrit ce qu'il veut dans une zone libre — ou ne renseigne rien —, la saisie devient peut-être plus facile, mais les données perdent une grande partie de leur utilité collective.

Cette **source** ne rend pas les témoignages précédents moins vrais. Les techniciens décrits continuent de reporter ou de contourner la saisie. Le PDF fait apparaître une autre réalité : supprimer toute contrainte aurait également un coût.

L'IA pourrait être tentée de fusionner ces éléments dans la première note et de la réécrire sous la forme d'un compromis général. Ce serait une erreur. Le livre blanc ne nuance pas le fait observé dans les entretiens ; il porte une autre proposition.

Il produit donc une deuxième **note atomique** :

Des comptes rendus structurés sont nécessaires pour comparer les interventions et exploiter les données d'intervention.

Le Contexte contient désormais deux idées distinctes :

- la contrainte de saisie peut dégrader l'information réellement recueillie ;
- l'absence de structure peut dégrader l'exploitation ultérieure de cette information.

Les deux notes peuvent être vraies en même temps. Elles tirent pourtant la décision produit dans des directions opposées.

L'opposition doit rester visible entre les notes

Le rôle du Contexte n'est pas de modifier la première note jusqu'à ce que la contradiction disparaisse. Il doit conserver les deux propositions et rendre leur relation explicite.

Une **note thématique** peut alors formuler ce que leur rapprochement permet de comprendre :

La qualité d'un compte rendu dépend d'une tension entre l'effort demandé au moment de la saisie et la structure nécessaire à son exploitation.

Cette troisième note n'est pas une moyenne diplomatique. Elle ne remplace aucune des deux notes atomiques. Elle montre le problème nouveau qui apparaît lorsqu'on les lit ensemble.

La contradiction rejoint également l'état du travail : comment obtenir une information structurée sans provoquer le report ou le contournement de la saisie ? Aucune des trois sources ne permet encore de répondre.

Voici ce que l'on voit à l'écran

Le Contexte utilisé pour cet exemple existe sous la forme d'un petit dossier de fichiers. Sa structure n'est pas le concept lui-même : elle est ma manière actuelle de lui donner une forme concrète.

```
contexte-qualite-comptes-rendus/
├─ 00-frame/
│   ├── mission.md
│   └── directives.md
├─ 01-state/
│   ├── state.md
│   └── contradictions.md
├─ 03-sources/
│   ├── SRC00001-entretien-exploitation.md
│   ├── SRC00002-entretien-exploitation.md
│   └── SRC00003-livre-blanc-donnees-intervention.md
├─ 04-notes/
│   ├── atomic/
│   │   ├── AT000001-la-contrainte-de-saisie.md
│   │   └── AT000002-des-comptes-rendus-structures.md
│   └── thematic/
│       └── THM00001-la-qualite-depend-de-la-saisie.md
└─ 05-outputs/
    └── questions-prochain-entretien.md
```

Vous pouvez maintenant rattacher les mots employés depuis le début du livre à des objets visibles.

- La **mission** explique pourquoi le dossier existe.
- Les **sources** conservent ce qui a été reçu avant interprétation.
- Les **notes atomiques** stabilisent des idées distinctes.
- La **note thématique** rend leur tension intelligible.
- L'**état** dit ce que cette tension laisse ouvert.
- Le **livrable** prépare l'action suivante.

Rien de tout cela n'exige un logiciel mystérieux. VS Code (mais ça pourrait être l'application Codex ou Claude Cowork) est ici la fenêtre par laquelle j'accède au

Contexte. Le Contexte, lui, réside dans les fichiers, leurs fonctions et les relations conservées entre eux.

2

Le retour devient situé

Quelques jours plus tard, je reprends le travail et propose à l'IA :

« Nous devrions peut-être rendre tous les champs facultatifs. »

Dans une conversation isolée, l'IA pourrait défendre cette idée, proposer une interface plus légère ou dresser la liste de ses avantages. Dans le Contexte, elle rencontre les deux notes atomiques et la tension qui les relie.

Elle peut alors répondre :

Cette proposition réduirait probablement une partie de la friction décrite dans les entretiens. Elle entre toutefois en opposition avec le besoin de données structurées relevé dans le livre blanc. Les sources montrent deux contraintes réelles ; elles ne permettent pas encore de choisir entre simplification, saisie progressive, dictée vocale ou adaptation des champs au type d'intervention.

Ce retour ne vient pas d'une connaissance générale du product management. Il est situé dans le travail en cours.

L'IA sait quelle question nous cherchons à comprendre, quelles idées sont soutenues par quelles sources et quelle contradiction reste ouverte.

Elle ne décide pas à ma place. Elle empêche simplement ma nouvelle intuition d'effacer l'une des réalités déjà établies.

Le premier livrable prépare la source suivante

L'état du Contexte ne dit pas : « concevoir un nouveau formulaire ». Il sépare ce qui paraît acquis de ce qui reste incertain.

Ce qui paraît acquis Des contraintes de saisie peuvent provoquer report et contournement. Une structure suffisante reste nécessaire pour exploiter les comptes rendus.

Ce qui reste ouvert Nous ignorons encore quelle friction pèse le plus et quelles informations doivent absolument rester structurées.

Prochaine action Préparer un entretien qui distingue les causes du problème au lieu de présenter immédiatement une solution.

Le premier livrable est donc modeste : un guide de questions pour les nouveaux entretiens. Il doit permettre de comprendre le moment de la saisie, les contraintes du

terrain, l'utilité réelle des différents champs et les formes de contournement.

Le livrable ne tranche pas la contradiction. Il prépare la matière qui permettra peut-être de le faire plus tard.

2

Un objet de travail, pas une machine à tout fusionner

Cet exemple montre aussi une erreur que l'IA peut commettre : vouloir faire entrer une nouvelle source dans une note existante parce que les sujets se ressemblent.

Une **note atomique** peut grandir lorsque plusieurs sources portent sur la même proposition. C'est ce qui se passe avec les deux entretiens. Mais le livre blanc ne constitue pas une nouvelle preuve de cette proposition. Il fait apparaître une idée différente qui mérite sa propre note.

La distinction compte. Si chaque opposition réécrit la note précédente, « **le monde des idées** » finit par lisser les tensions. La connaissance paraît cohérente, mais elle ne correspond plus à ce que les sources ont réellement apporté.

Le Contexte ne garantit pas que l'IA choisira toujours la bonne frontière. L'humain doit pouvoir refuser une fusion, demander une nouvelle note et vérifier la relation créée

entre elles. La structure rend cette correction possible et durable ; elle ne la décide pas automatiquement.

Vous avez maintenant vu l'objet :

- des **sources** qui restent intactes,
- des **notes atomiques** qui gardent chacune leur centre,
- une **note thématique** qui rapproche sans fusionner,
- une **contradiction** qui demeure ouverte et un **livrable** qui prépare la suite.

Les chapitres suivants pourront revenir aux usages du Contexte, puis ouvrir plus précisément son architecture à partir de ces éléments devenus visibles.

À RETENIR — UNE CONTRADICTION SE CONSERVE

Deux sources peuvent être vraies ensemble et tirer la décision dans des directions opposées. Une note atomique grandit quand une source renforce la même proposition ; une proposition différente mérite sa propre note. Fusionner pour faire disparaître la tension produit une connaissance cohérente qui ne correspond plus aux sources.

Partie 3

Trois usages concrets

Les deux parties précédentes ont dit ce que le Contexte change. Cette partie le montre au travail, sur trois usages que je pratique réellement.

Répondre à un cahier des charges, écrire un article à partir d'une pensée encore informe, faire mûrir un sujet produit au milieu d'une journée fragmentée. Trois durées, trois matières, trois destinataires — et le même mouvement à chaque fois : retrouver ce qui a déjà été compris, le transformer, puis produire sans épuiser ce qui a servi.

3.1

Répondre à un cahier des charges sans perdre le travail accompli

3

Le premier cahier des charges que j'ai traité avec ce système comptait quarante-trois questions.

Il fallait répondre sur l'architecture du produit, sa sécurité, les sauvegardes, la continuité de service et la protection des données. Une partie des informations se trouvait dans notre documentation. D'autres étaient enfouies dans des fichiers de configuration. Pour certaines, il fallait solliciter l'équipe qui exploitait le service.

Nous avons dû rassembler sept sources techniques et faire un aller-retour avec l'équipe. Ce travail venait s'ajouter à la rédaction proprement dite : comprendre ce que le prospect demandait, retrouver l'information, vérifier qu'elle était encore vraie, décider ce que nous pouvions communiquer, puis remettre la réponse dans le format attendu.

Ce n'était pas une situation exceptionnelle. Chaque nouveau prospect envoyait son propre document. Une fois, c'était un tableur ; une autre fois, un fichier Word ou

un PDF. Les référentiels changeaient, les phrases aussi, mais les mêmes sujets revenaient.

Pourtant, nous avions chaque fois l'impression de recommencer. Répondre prenait entre trois et six heures et mobilisait plusieurs personnes, sans garantie que le prospect devienne client. Puis le document partait et le travail restait enfermé dans le fichier envoyé.

Mon premier objectif était donc simple : lorsque le questionnaire suivant arriverait, je ne voulais plus repartir de zéro.

Une réponse vraisemblable, mais fausse

La solution la plus évidente consistait à conserver les anciens questionnaires et à réutiliser leurs réponses.

Nous avons rapidement découvert pourquoi cela ne suffisait pas.

Certains documents reçus contenaient déjà des réponses. Elles pouvaient venir d'un modèle, d'un autre fournisseur ou d'un échange antérieur. Dans l'un d'eux, une réponse indiquait que la certification ISO 9001 datait de 2017. La phrase paraissait parfaitement exploitable. Elle était précise, déjà rédigée et placée en face de la bonne question.

La date correcte était 2018.

L'erreur ne tenait qu'à une année, mais elle révélait un problème beaucoup plus important. Une ancienne réponse ressemble à une connaissance. Elle possède une forme affirmative et rassurante. Lorsqu'une IA la rencontre dans un document, elle peut facilement la reprendre. Pourtant, sa présence ne dit ni qui l'a écrite, ni sur quoi elle repose, ni si elle décrit réellement notre produit.

J'ai alors compris que je ne devais pas construire une bibliothèque de réponses à copier. Il fallait construire une mémoire de ce que nous savions réellement répondre.

Le questionnaire reçu conservait une valeur : il montrait ce que le client voulait savoir. Mais il ne pouvait pas, à lui seul, prouver la réponse. Cette distinction est devenue une règle du Contexte : le document du client conserve la demande ; la réponse doit rester reliée à une information que nous avons vérifiée.

Conserver les faits plutôt que les formulaires

Nous avons commencé à extraire des cahiers des charges ce qui pouvait servir au suivant.

Les questions n'étaient plus rangées seulement par client ou par numéro. Nous les rapprochions par sujet :

sauvegardes, authentification, chiffrement, sous-traitants, hébergement, continuité de service. Une question appelée « 14.6.l » chez un prospect pouvait ainsi rejoindre une question appelée « CL-072 » chez un autre si toutes deux portaient sur le même sujet.

Pour chacun de ces sujets, le Contexte conservait les faits validés et leur origine. La réponse pouvait s'appuyer sur une documentation officielle, une configuration vérifiée, un test ou la confirmation de l'équipe compétente. Nous gardions également la date de validation et les limites de ce que nous pouvions annoncer.

Ce travail ne produisait pas toujours une réponse complète. Il faisait aussi apparaître les trous.

Lorsqu'une information manquait, le système devait le dire. Il préparait une question précise pour l'équipe concernée et indiquait quelles lignes du questionnaire cette réponse permettrait de compléter. Une absence d'information cessait ainsi de se cacher derrière une phrase plausible. Elle devenait une action identifiable.

Nous séparions enfin ce qui servait à vérifier une réponse de ce qui pouvait être envoyé au client. Le détail d'une configuration pouvait être indispensable au relecteur interne sans avoir sa place dans le document public. Le Contexte préparait donc la formulation destinée au

prospect tout en conservant, à côté, les preuves et les réserves nécessaires à sa validation.

Peu à peu, le but a changé. Il ne s'agissait plus seulement de terminer le questionnaire présent. **Chaque réponse vérifiée devait réduire le travail nécessaire pour les suivants.**

Le quatrième questionnaire

Le quatrième cahier des charges comptait cent quinze questions, soit bien plus que le premier.

Cette fois, nous ne partions plus d'un ensemble de fichiers épars. Le Contexte connaissait déjà les sujets récurrents et savait où retrouver les faits qui leur correspondaient. Quatre-vingt-seize réponses complètes ont pu être produites directement à partir de la mémoire constituée et des sources du produit, sans nouvelle collecte.

Le système a également classé le reste du document. Certaines réponses étaient solides, mais demandaient une dernière vérification. D'autres restaient incertaines. Quelques-unes ne pouvaient pas être données avec les informations disponibles.

Surtout, les manques avaient changé de nature. Nous ne passions plus des heures à chercher au hasard qui savait

quoi. Les points encore ouverts correspondaient à des décisions commerciales — par exemple la durée d'un engagement — ou à des confirmations techniques ponctuelles. Le travail restant était visible, attribuable et limité.

Ce résultat m'a permis de comprendre ce que signifiait réellement « répondre automatiquement » à un cahier des charges. Il ne s'agit pas de demander à une IA de remplir toutes les cases avec assurance. Il s'agit de préparer immédiatement ce qui peut l'être, de montrer le degré de confiance de chaque réponse et d'isoler clairement ce qui exige encore un jugement humain.

Dix minutes ne signifient pas cent pour cent de certitude

Dans mon usage, la préparation d'un cahier des charges est passée de trois à six heures à environ dix minutes pour produire 80 à 90 % du document.

Ce chiffre pourrait donner une impression trompeuse. Il ne signifie pas que 80 à 90 % des réponses deviennent instantanément certaines et prêtes à engager l'entreprise. Il signifie que le document est largement préparé : les réponses connues sont placées, les sources sont retrouvées, les points fragiles sont signalés et les

informations manquantes sont transformées en demandes précises.

La relecture humaine reste indispensable. Elle porte toutefois sur les bons endroits. Au lieu de consacrer son temps à fouiller d'anciens fichiers et à solliciter une nouvelle fois les mêmes personnes, le relecteur peut vérifier les engagements, arbitrer ce qui sera divulgué et se concentrer sur les réponses nouvelles ou incertaines. Le gain le plus important n'est donc pas seulement le passage de plusieurs heures à quelques minutes. C'est le changement de nature du travail.

Le premier questionnaire avait exigé une enquête. Les suivants ont bénéficié de tout ce que cette enquête avait permis d'apprendre. Une erreur de date est devenue une règle de prudence. Une confirmation obtenue auprès d'une équipe est devenue un fait réutilisable. Une question sans réponse est devenue un manque connu plutôt qu'un oubli.

Le prospect reçoit toujours son document.

L'entreprise, elle, ne perd plus ce qu'elle a appris en le préparant.

À RETENIR — UNE RÉPONSE VAUT PAR SA PREUVE

Une ancienne réponse ressemble à une connaissance : elle est affirmative, bien formée, et reste muette sur son origine et sur sa date. Conserver les faits vérifiés et leur provenance, plutôt que les formulaires déjà remplis. Un manque explicite vaut mieux qu'une phrase plausible : il devient une question adressée à quelqu'un.

3.2

Écrire un article à partir d'une pensée encore informe

3

Les articles que je publie sur mon blog commencent rarement devant mon ordinateur, face à une page blanche soigneusement préparée.

Une idée apparaît après une difficulté rencontrée, un outil essayé, une discussion ou un changement d'avis. Elle arrive parfois en marchant, dans les transports en commun ou sur mon canapé. Je sens qu'elle mérite un texte, mais je ne connais encore ni son angle exact ni l'ordre dans lequel elle devra être racontée.

À ce moment-là, je prends souvent mon téléphone.

Je ne l'utilise pas simplement pour enregistrer une note vocale ou dicter un texte que je recopierai plus tard.

J'engage une conversation avec l'IA. Elle m'écoute, me répond et me relance pendant que je réfléchis. Cette conversation rejoint le même environnement de travail que celui que je retrouverai sur mon ordinateur. Je peux donc commencer loin de mon bureau sans créer une note isolée qu'il faudra ensuite transférer et réinterpréter.

Décharger une pensée en train de se former

Je crée un Contexte avec un but simple : écrire l'article.

Puis je lui donne tout ce que j'ai en tête.

Pendant parfois deux heures, je parle d'un premier point, puis d'un autre, avant de revenir au premier. Je raconte une expérience, saute vers une conclusion, ajoute un détail, ouvre une nouvelle piste et abandonne momentanément la précédente. Je peux répéter quatre fois la même idée parce qu'elle me revient sous des formes différentes. Je peux aussi affirmer une chose, puis dire le contraire une heure plus tard parce que ma pensée a évolué pendant la conversation.

Ce désordre n'est pas un accident qu'il faudrait éviter avant de commencer. C'est souvent la forme réelle d'une idée lorsqu'elle est encore en train de naître.

Le dialogue avec l'IA me permet de continuer sans imposer trop tôt une structure. Une réponse me fait préciser un souvenir. Une question révèle que je viens de sauter une étape. Une reformulation me montre que ce que je croyais évident ne l'est pas encore. Je corrige, je nuance et je repars dans une autre direction.

Le Contexte conserve cette parole comme une source de session. Il ne retient pas seulement les phrases les plus propres. Il garde aussi les reprises, les hésitations et les

contradictions qui permettront de comprendre comment la pensée s'est déplacée.

L'IA rapproche ensuite les quatre formulations d'une même idée au lieu de les traiter comme quatre idées différentes. Elle signale deux affirmations incompatibles, retrouve la nuance arrivée une heure plus tard et réunit un exemple avec le principe qu'il éclaire. **Elle ne fait pas disparaître le désordre : elle le rend travaillable.**

Cette première étape me libère d'un choix prématuré. Je n'ai pas encore à décider quelle phrase ouvrira l'article, si telle idée mérite une section ou dans quel ordre raconter les événements. Je cherche d'abord à vider ce que j'ai en tête, sans laisser filer les idées qui ne font que la traverser.

Le Contexte sait cependant pourquoi cette matière est là. Il ne reçoit pas une conversation sans destination. Il reçoit la matière encore désordonnée d'un article à construire.

Utiliser l'IA comme partenaire de maturation

Une fois cette première matière déposée, je demande à l'IA ce qui manque.

Elle relève une affirmation trop rapide, un présupposé jamais expliqué, un passage qui ne tient que si le lecteur

connaît déjà mon expérience. Elle demande un exemple. Elle rapproche deux idées formulées à des moments différents. Elle propose parfois une piste qui paraît plausible et que je refuse parce qu'elle ne correspond pas à ce que j'ai vécu.

Je réponds, je précise, je corrige. Cette conversation ne produit pas encore l'article. Elle densifie sa matière.

Le rôle de l'IA est important, mais sa place reste claire. Elle dispose de la totalité de ce que j'ai versé et y cherche des tensions plus vite que moi. Elle ignore en revanche ce que je veux réellement défendre tant que je ne l'ai pas formulé. Elle ouvre la question ; je décide si cette question mérite de déplacer le texte.

Ce dialogue raccourcit le temps nécessaire pour rencontrer une faiblesse. Seul, j'aurais découvert deux jours plus tard qu'un exemple manquait ou que ma conclusion reposait sur une étape jamais expliquée. L'IA provoque cette découverte pendant que l'idée est encore facile à reprendre.

Ajouter des sources lorsque l'expérience ne suffit plus

Mes articles partent souvent de ce que j'ai vécu. La matière extérieure garde pourtant son rôle :

- une référence donne un nom à ce que j'ai observé ;

- une documentation vérifie un fonctionnement ;
- un autre auteur défend une position voisine, ou montre une limite que je n'avais pas rencontrée ;
- une recherche complémentaire révèle que mon expérience ne couvre qu'un cas particulier.

3

Ces documents rejoignent le *Contexte* comme **sources documentaires**. Leur rôle diffère de celui de ma parole initiale. La **source de session** conserve ce que j'affirme et ce que je comprends de mon expérience. Les **sources documentaires** apportent une matière que je peux confronter à cette expérience.

L'article peut alors distinguer ce que j'ai observé, ce qu'une source établit et ce qui reste une interprétation. Cette distinction protège le texte contre une tentation fréquente : présenter une expérience personnelle comme une règle générale.

Transformer les sources en idées réutilisables

À ce stade, le *Contexte* ne passe pas directement des sources au plan de l'article. Il transforme d'abord la matière brute en **notes atomiques**.

Chaque note stabilise une idée autonome issue de la conversation ou d'un document :

- une affirmation que je veux défendre ;

- un exemple qui lui donne une forme concrète ;
- une nuance qui en limite la portée ;
- une objection à laquelle l'article devra répondre ;
- un fait extérieur qui confirme ou contredit mon expérience.

La note conserve son origine. Je peux ainsi comprendre l'idée sans relire toute la conversation, puis revenir à la source si je dois vérifier une formulation ou retrouver son contexte.

Cette transformation change la manière de préparer l'article.

- La conversation suivait le mouvement de ma pensée ; les notes regroupent désormais ce qui parle de la même idée, même lorsque ces éléments sont apparus à une heure d'intervalle ou dans deux sources différentes.
- Une note centrale peut faire apparaître l'angle de l'article.
- D'autres lui apportent des exemples, des preuves, des objections ou des conséquences.

Ces notes ne sont encore ni des paragraphes ni un plan préfabriqué. **Elles restent neutres face au livrable : la**

même idée pourra nourrir cet article, une présentation ou un autre travail.

Passer par le plan avant de rédiger

Lorsque la matière devient suffisante, je demande un plan.

3

Cette étape m'empêche de passer directement d'une conversation riche à un texte qui en suivrait simplement l'ordre. Une discussion avance par retours, corrections et associations. Un article doit conduire un lecteur qui ne connaît ni la discussion ni son histoire.

Le plan fixe cette progression. Il indique ce qui doit être compris d'abord, où placer l'exemple principal, quelle objection traiter et ce que la conclusion peut réellement affirmer.

L'IA propose une structure à partir de la matière disponible. Je la déplace, je la resserre, je refuse les parties qui donnent trop d'importance à une piste secondaire. Deux ou trois échanges suffisent souvent pour obtenir une direction claire.

La rédaction vient ensuite.

À ce stade, l'IA ne crée pas le texte à partir d'une consigne vague. Elle travaille avec un corpus d'idées et de thématiques extraites de mes expériences racontées,

des réponses aux questions, des sources et un plan validé. Elle regroupe les répétitions, replace les exemples et transforme une parole sinueuse en prose lisible.

Je relis le résultat, en cherchant trois écarts précis :

- une nuance qui a disparu ;
- un passage devenu trop affirmatif ;
- une formulation qui trahit l'idée.

Je les corrige, puis je décide de la version publiée.

Ce que le Contexte apporte à un travail court

Un article peut être écrit en un ou deux jours. Il ne semble pas avoir besoin de la même mémoire qu'un produit suivi pendant plusieurs années ou qu'un livre de plusieurs dizaines de chapitres.

Ce cas léger permet justement de voir que la durée n'est pas le seul enjeu.

Le Contexte garde ensemble la parole initiale, les sources ajoutées, les questions posées, le plan choisi et le texte complet. Si le travail s'interrompt, je retrouve son état. Si une phrase devient contestable, je peux revenir à son origine. Si une idée apparue pendant l'écriture dépasse l'article, elle peut être stabilisée séparément et nourrir un travail futur.

Tout ce matériau ne mérite pas nécessairement d'être conservé indéfiniment. Un petit Contexte peut atteindre son but, produire son article, puis être archivé. Sa valeur tient à la continuité qu'il a donnée au travail pendant sa formation et à la trace qu'il laisse si l'on doit comprendre plus tard comment le texte a pris cette direction.

La publication elle-même peut déclencher d'autres tâches : préparer un résumé, adapter le texte à plusieurs langues ou produire les éléments nécessaires à sa mise en ligne. Ces opérations prolongent le même résultat, sous des règles déjà connues du Contexte.

Le mouvement complet reste simple : une expérience devient une parole, cette parole rencontre des questions et des sources, la matière devient un plan, puis le plan devient un texte.

L'IA accélère chaque transformation, et le Contexte garde le fil qui mène de l'une à l'autre.

Rien de ce mouvement n'appartient en propre à l'écriture longue. Il commence dès qu'une idée mérite mieux qu'une note vocale oubliée dans un téléphone.

À RETENIR — LE DÉSORDRE EST UNE MATIÈRE

Une pensée qui naît se répète, se contredit et revient en arrière : c'est sa forme normale, et non un défaut à corriger avant de commencer. Le Contexte garde cette parole comme source de session, puis la rend travaillable. Le plan s'intercale entre la conversation et le texte, parce qu'un lecteur ne suit pas l'ordre d'une discussion.

3.3

Faire mûrir un sujet produit au milieu du chaos

3

Les deux chapitres précédents ont suivi un seul sujet à la fois, de bout en bout. Ce chapitre décrit la réalité d'une journée type de product manager : plusieurs sujets, demandes et temporalités avancent en même temps.

Je suis le seul product manager de mon entreprise. Cela signifie qu'une grande partie des informations liées au produit finit par passer devant moi.

Dans une même journée, un client prononce une phrase qui me semble importante.

- Elle tient en quelques mots.
- Je ne sais pas encore si elle révèle un problème partagé par d'autres utilisateurs ou seulement une difficulté particulière, mais je ne veux pas la perdre.

Un peu plus tard, je récupère un livre blanc consacré à l'évolution d'un enjeu métier.

- Cette fois, la matière ne tient plus en une phrase, mais dans un PDF de plusieurs dizaines de pages.
- Je ne sais pas encore ce qui me sera utile ni si ce document confirmera ce que j'observe sur le terrain.

Puis les développeurs me parlent d'une fonctionnalité en cours de réalisation.

- Il faut répondre à une question, préciser un comportement ou revenir sur une hypothèse prise quelques semaines plus tôt.

Pendant ce temps, le CPO souhaite faire le point sur deux paris liés à une autre partie du produit.

Une conversation avec le support révèle enfin qu'il manque une documentation fonctionnelle.

- Une partie de cette documentation n'existe encore que dans ma tête.

À la fin de la journée, je suis passé sur trois ou quatre sujets différents (voir plus). J'ai reçu une citation de quelques mots, un document entier, une question sur un développement en cours, une demande d'arbitrage et un besoin de documentation.

Ce n'est pas une journée particulièrement mauvaise : c'est une journée ordinaire.

Des entrées qui n'ont ni la même taille ni la même destination

Le chaos du product management ne vient pas seulement de la quantité d'informations. Il vient de leur différence de nature.

Chacune de ces cinq entrées regarde un moment différent de la vie du produit :

- une phrase de client est un signal faible, dont je ne sais pas encore s'il en annonce d'autres ;
- un PDF donne une vision générale d'un métier ;
- une discussion avec les développeurs porte sur ce que nous construisons maintenant ;
- une demande du support concerne ce qui existe déjà et doit être expliqué ;
- une question du CPO oblige à comparer plusieurs futurs possibles.

Ces éléments n'ont pas vocation à entrer dans un document unique.

- Ils ne parlent pas tous du même sujet et ne se situent pas au même moment de la vie du produit.

Si je les rassemble simplement dans une liste de notes, je conserve les informations, mais je reporte le travail de compréhension au jour où j'en aurai besoin.

Les Contextes me permettent de maintenir plusieurs fils en parallèle.

- La citation rejoint le sujet auquel elle semble appartenir, même si sa signification reste incertaine.

- Le PDF est conservé avec son origine et peut être analysé pour en extraire les quelques idées (**notes atomiques**) réellement utiles.
- La conversation avec les développeurs rejoint le travail sur la fonctionnalité en cours.
- Le manque signalé par le support devient un travail à produire au lieu de rester une remarque entendue entre deux réunions.

Tout n'est pas immédiatement transformé en connaissance :

- la phrase du client restera peut-être ambiguë,
- le livre blanc peu pertinent,
- l'intuition contredite la semaine suivante.

Le Contexte ne donne pas de l'intelligence à ce qui entre. Il lui donne un endroit, une origine et une chance d'être retrouvé au bon moment.

Le Contexte contient le chaos sans prétendre l'abolir.

Pendant que les informations entrent, les demandes de sortie continuent

La journée ne se résume pas à recevoir et à classer de la matière : il faut aussi produire.

Le support me demande, par exemple, une documentation fonctionnelle sur une fonctionnalité que nous sommes en train de travailler.

- Il faut expliquer son fonctionnement, préciser les cas importants et donner aux personnes qui accompagnent les clients de quoi répondre correctement.

Le même jour, je peux vouloir terminer un article pour le blog de l'entreprise.

- Cet article ne parle pas nécessairement de la même fonctionnalité.
- Il peut toutefois porter sur le même sujet fonctionnel : le métier auquel répond le produit, une difficulté récurrente rencontrée par les utilisateurs ou une évolution de leurs pratiques.

Les deux documents puisent alors dans une partie de la même connaissance sans raconter la même chose.

- La documentation destinée au support explique le produit tel qu'il fonctionne.
 - Elle doit être précise, opérationnelle et adaptée aux questions des utilisateurs.
- L'article développe une expérience, un problème métier ou une réflexion.

- Il peut mobiliser certaines des mêmes idées, mais il les organise pour un autre lecteur et dans un autre but.

Je n'ai pas besoin de reconstruire deux fois tout ce que je sais du sujet.

- Le Contexte conserve une matière intermédiaire (les **notes atomiques** et **notes thématiques**) qui n'est encore ni la documentation ni l'article.
- À partir de cette matière, l'IA peut préparer plusieurs assemblages : l'un centré sur l'usage du produit, l'autre sur l'enjeu fonctionnel plus large.

Je relis évidemment les deux résultats.

- Une documentation support engage la compréhension du produit.
- Un article engage mon point de vue.

Mais je consacre mon temps à vérifier le fond et l'orientation, plutôt qu'à retrouver les mêmes éléments et à les rédiger une nouvelle fois depuis le début.

Ce mouvement peut avoir lieu dans la même journée.

Pendant que de nouvelles informations arrivent, une connaissance déjà organisée permet de produire les documents dont l'entreprise ou moi-même avons besoin.

Une petite phrase peut devenir un sujet produit

La citation entendue le matin ne déclenche pas nécessairement une décision.

- Prise seule, elle peut ne rien prouver.
- Un autre client exprimera peut-être une difficulté proche quelques semaines plus tard.
- Une interview apportera une nuance.
- Le livre blanc récupéré ce jour-là donnera peut-être un nom à l'évolution observée.
- Une donnée du support montrera que le problème revient plus souvent que nous le pensions.

Ces éléments arrivent à des moments différents et sous des formes différentes.

- Le Contexte permet de les rapprocher sans effacer leur origine.
- Il peut relever
 - qu'une nouvelle remarque ressemble à un signal déjà conservé,
 - faire apparaître une contradiction,
 - demander ce qui permettrait de départager deux interprétations.

Peu à peu, une phrase isolée peut devenir un sujet digne d'être travaillé. À ce moment-là, je dispose déjà d'une

partie de son histoire : les premiers signaux, les documents qui l'éclairent, ce qui les confirme, ce qui les contredit et les questions encore ouvertes.

Le Contexte aide cette matière à mûrir.

- Il ne décide pas que le problème est important et ne choisit pas la solution à ma place.

Je reste responsable du pari produit.

- En revanche, je peux prendre cette décision avec autre chose qu'un souvenir vague ou une série de documents dispersés.

Le même système accompagne ensuite le travail :

- ajouter des entretiens,
- confronter une hypothèse aux faits,
- préparer des scénarios,
- produire un argumentaire pour le CPO,
- réaliser des spécifications,
- ...

Un petit signal capté au milieu d'une journée peut ainsi continuer à vivre assez longtemps pour devenir une décision réelle.

Une idée abandonnée peut continuer à mûrir

Toutes les explorations ne vont pas jusqu'à la mise en production.

Il m'est arrivé de travailler sur des usages de l'intelligence artificielle qui semblaient intéressants, mais que la technologie disponible ne permettait pas encore de réaliser correctement.

- L'idée n'était pas nécessairement mauvaise.
- Le moment l'était.

Dans un fonctionnement classique, l'exploration aurait produit quelques documents, puis aurait disparu dans un dossier après la décision d'abandon.

- Un an ou deux plus tard, il aurait fallu retrouver ces fichiers, se rappeler les hypothèses de départ et comprendre pourquoi nous nous étions arrêtés.

Avec un `Contexte`, l'exploration peut être archivée avec sa démarche : les signaux initiaux, les scénarios imaginés, les essais, les limites rencontrées et la raison de l'arrêt.

- Lorsque la technologie évolue, je peux reprendre ce travail sans transformer un ancien résultat négatif en idée définitivement mauvaise.

Je retrouve ce que nous voulions faire, mais aussi pourquoi cela ne fonctionnait pas. Je peux alors vérifier si l'obstacle existe encore, conserver ce qui reste valable et reprendre la réflexion à partir de là.

La maturation d'un sujet produit n'est donc pas toujours un trajet continu.

- Elle peut commencer par une phrase, s'enrichir pendant plusieurs mois, s'interrompre, puis reprendre lorsque le produit, le marché ou la technologie a changé.

Faire émerger de l'ordre sans ralentir la journée

Ma journée reste fragmentée.

- Les clients ne coordonnent pas leurs remarques avec les demandes du support.
- Les développeurs ne choisissent pas le moment où je viens de finir la lecture d'un PDF.
- Les décisions à préparer ne s'arrêtent pas pendant que j'écris une documentation ou un article.

Le Contexte ne transforme pas cette réalité en succession parfaite de tâches.

Il permet à chaque fragment de conserver une continuité.

- Une petite phrase et un long document peuvent entrer sans recevoir le même traitement.
- Plusieurs sujets peuvent avancer sans se confondre.
- Une connaissance déjà travaillée peut servir à plusieurs livrables sans être enfermée dans le premier.

- Une idée interrompue peut attendre sans perdre toute son histoire.

C'est ainsi que je l'utilise en product management : non pour rendre ma journée moins mouvante, mais pour éviter que son mouvement disperse ce que j'apprends.

Le chaos continue d'entrer.

Il ne repart plus sous forme de perte.

Ces trois usages ont un point commun que je n'ai pas encore nommé : chacun se termine par quelque chose qui part — une réponse envoyée, un article publié, une décision prise.

La partie suivante regarde ce qui reste une fois que le livrable est parti.

À RETENIR — PLUSIEURS FILS EN PARALLÈLE

Une phrase de client et un PDF de cinquante pages appellent un traitement et une destination différents. Le Contexte donne à chaque fragment un endroit, une origine et une chance d'être retrouvé au bon moment. Une exploration abandonnée garde ses raisons d'arrêt : elle redevient reprenable quand la technologie ou le marché change.

Partie 4

Ce que les livrables laissent derrière eux

Les trois usages précédents se terminent chacun par un livrable remis. Cette partie regarde ce qui reste une fois qu'il est parti.

Une même question, posée deux fois à six mois d'intervalle, sert de fil. La première fois, la réponse tient dans une cellule de tableur. La seconde, cette réponse existe encore mais ne suffit plus : ce qui manque est le raisonnement qui l'avait produite, et les relations entre les documents qui le soutenaient.

4.1

La réponse tient dans une cellule, pas le raisonnement

4

Revenons à l'un des cahiers des charges de la partie précédente.

La question arrive dans la cellule D47 d'un tableur :

| *Vos sauvegardes sont-elles chiffrées au repos ?*

La réponse finale tiendra en trois lignes. Pour les écrire, je dois pourtant réunir plusieurs éléments :

- notre documentation décrit le fonctionnement général des sauvegardes ;
- un fichier de configuration indique le réglage réellement actif ;
- l'équipe Ops confirme la manière dont ce réglage s'applique en production ;
- le juridique précise ce que nous pouvons communiquer à un prospect.

Je passe du tableur à la documentation, de la documentation à la configuration, puis d'un échange avec l'équipe Ops à un arbitrage juridique. Je reviens enfin dans la cellule D47 et j'y écris la réponse.

Le document part.

Trois lignes au bout d'une chaîne

La cellule contient le résultat du travail. Elle ne montre pas le chemin qui l'a produit.

Elle ne dit pas :

- quelle documentation j'ai consultée ;
- quel réglage j'ai vérifié ;
- qui a confirmé son application ;
- quelle formulation a été autorisée ;
- quelles réserves ont été écartées du texte envoyé au prospect.

Ces éléments existent pourtant. Ils ont déterminé chacune des trois lignes, et pendant que je répons, leur relation est claire dans ma tête.

Une fois le document envoyé, cette unité disparaît. Le tableur conserve la réponse, mais pas le raisonnement qui permettait de lui faire confiance.

Chaque outil a fait son travail

Le problème ne vient pas d'un mauvais rangement.

- La documentation se trouve bien dans l'espace documentaire.
- La configuration reste dans le produit.

- La réponse de l'équipe demeure dans son fil de discussion.
- L'arbitrage juridique possède son propre support.
- La cellule D47 est à sa place dans le document du prospect.

4

Chaque outil a conservé l'objet qu'on lui avait confié. Aucun n'avait pour fonction de garder l'ensemble du raisonnement.

Pour le reconstruire, il faut savoir quels éléments chercher, où les chercher et quel rôle chacun a joué. Cette connaissance reste portée par la personne qui a préparé la réponse.

Le livrable aplatit ce qui l'a produit

Un livrable doit présenter un résultat adapté à son destinataire.

- Le prospect attend une réponse claire, pas le journal complet de l'enquête menée pour l'obtenir.

Cette simplification est nécessaire. Elle devient coûteuse lorsque le livrable reste la seule trace du travail.

Le résultat final est toujours plus court que le chemin qui y mène — la réponse envoyée comme l'article publié ou la décision produit. Si seul le résultat demeure, le travail suivant doit recommencer une partie de ce chemin.

Donner une adresse au raisonnement

Le Contexte ne remplace ni le tableur, ni la documentation, ni la configuration, ni les échanges avec les équipes.

- Il donne une adresse commune au travail qui les relie.

La question D47, les éléments consultés, les validations obtenues et la réponse produite appartiennent au même sujet. Ils peuvent garder leur forme et leur emplacement d'origine tout en restant reliés dans le Contexte.

La cellule continue de remplir sa fonction : répondre au prospect. Le Contexte en remplit une autre : conserver ce qui permettra de comprendre, de vérifier et de reprendre cette réponse.

Six mois plus tard, cette différence devient décisive.

À RETENIR — LE LIVRABLE NE CONSERVE PAS LE RAISONNEMENT

Une réponse peut tenir en trois lignes tout en dépendant de plusieurs sources, validations et réserves. Les outils conservent chacun leur objet ; l'humain porte encore les relations entre eux. Le Contexte donne une adresse au travail qui relie ces éléments sans remplacer leurs outils d'origine.

4.2

Six mois plus tard, la réponse ne suffit plus

4

Six mois après la cellule D47, un nouveau prospect pose la même question sur le chiffrement des sauvegardes.

- Je retrouve l'ancienne réponse en dix secondes.
- Elle est claire, bien écrite et prête à être recopiée.

Je reste pourtant devant elle sans pouvoir m'en servir.

Trois questions suffisent à bloquer la reprise :

- d'où venait cette réponse : de la documentation, du réglage observé ou de ce que l'équipe Ops m'avait expliqué ce jour-là ?
- le juridique avait-il validé cette formulation exacte ou une version plus prudente ?
- le réglage est-il encore le même aujourd'hui ?

La réponse a survécu ; ce qui permettait de lui faire confiance a disparu.

Retrouver n'est pas encore reprendre

L'ancienne cellule me rend un texte. Elle ne me rend :

- ni son niveau de preuve,
- ni ses limites,

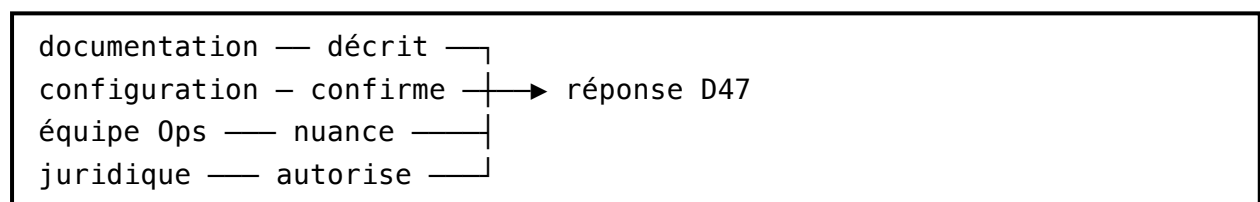
- ni les vérifications nécessaires avant de l'utiliser à nouveau.

Sans autre mémoire, je dois refaire toute l'enquête du chapitre précédent, dans l'ordre inverse. Je ne repars pas tout à fait de zéro, puisque je possède déjà une réponse. Je repars néanmoins sans savoir quelle part du travail reste valable.

C'est l'erreur de la certification datée de 2017 au lieu de 2018, vue cette fois du côté de nos propres réponses : la phrase est nette, elle vient de nous, et sa preuve est devenue invisible.

Retrouver le rôle de chaque élément

Dans le Contexte, l'ancienne réponse ne reste pas seule. Elle conserve les relations utiles au raisonnement qui l'a produite :



Les verbes changent ce que je peux faire de chaque élément.

- La documentation décrit le fonctionnement général ; elle ne prouve pas à elle seule le réglage actif.

- La configuration confirme ce réglage à un moment donné ; elle ne décide pas de ce qui peut être communiqué.
- L'équipe Ops apporte une nuance issue de l'exploitation réelle.
- Le juridique autorise une formulation destinée au prospect.

4

Je ne retrouve plus seulement quatre objets proches d'une réponse. Je retrouve la fonction que chacun remplissait.

Cette précision empêche une source de prouver davantage que ce qu'elle établit réellement. Elle montre aussi quelle vérification doit être recommencée lorsque le temps a passé.

Vérifier ce qui peut avoir changé

Le Contexte ne transforme pas l'ancienne réponse en vérité permanente : il prépare sa vérification.

- La documentation peut être encore valable tandis que la configuration a changé.
- La configuration peut rester identique tandis que la formulation autorisée doit être revue.
- Une nuance apportée par l'équipe Ops peut avoir disparu après une évolution du produit.

Les relations rendent ces différences visibles. Je peux contrôler les éléments susceptibles d'avoir changé sans reprendre toute l'enquête au hasard.

La réponse finale sera peut-être identique à celle d'il y a six mois. La différence tient à ce que je sais pourquoi je peux encore l'utiliser.

Conserver les relations qui servent le travail

Il ne s'agit pas de relier tous les documents entre eux.

- Deux fichiers peuvent partager un mot sans participer au même raisonnement.

Une relation mérite d'être conservée lorsqu'elle change la manière de comprendre, de vérifier, de décider ou de produire.

- Dans le cas de D47, quatre relations suffisent : décrire, confirmer, nuancer et autoriser.

Elles permettent de répondre à des questions concrètes :

- sur quoi cette réponse repose-t-elle ?
- quelle partie doit être vérifiée aujourd'hui ?
- qui avait validé ce que nous pouvions communiquer ?
- qu'est-ce qui rendrait cette réponse caduque ?

Le Contexte ne cherche donc pas à fabriquer une carte totale de tout ce que je sais. Il conserve le tissu nécessaire à un travail donné.

Ouvrir la machine

Le Livre I a montré comment le `Contexte` est né, ce qu'il change dans la relation avec l'IA et comment il accompagne plusieurs usages. Le cas D47 fait apparaître ce qui reste à comprendre.

- Comment conserver la mission du travail sans la répéter à chaque séance ?
- Où placer les sources, les idées, les décisions et les livrables pour qu'ils gardent des rôles distincts ?
- Comment charger ce qui est utile maintenant sans faire entrer toute la matière dans la conversation ?

Le Livre II va ouvrir cette machine. Il suivra le parcours complet d'une matière, depuis son arrivée dans un `Contexte` jusqu'à sa transformation, sa production et sa reprise.

À RETENIR — RETROUVER CE QUI PERMET DE FAIRE CONFIANCE

Retrouver une réponse ne suffit pas : il faut retrouver ce qui permet encore de l'utiliser. Une relation nommée indique le rôle précis d'une source, d'une validation ou d'une réserve.

Le `Contexte` prépare la vérification en montrant ce qui soutient la réponse et ce qui peut avoir changé.

LIVRE II

Construire et faire vivre le Contexte

- 5 Sous le capot d'un Contexte
- 6 Le Contexte dans son écosystème

Partie 5

Sous le capot d'un Contexte

Jusqu'ici, le Contexte a été décrit par ce qu'il permet. Cette partie l'ouvre.

Elle présente d'abord son arborescence complète, puis suit une séance de travail du début à la fin : ouvrir et retrouver le présent, faire entrer de la matière, la transformer en idées, produire un livrable, arbitrer, puis sauvegarder pour que la séance suivante reparte sans rien reconstruire.

L'exemple qui traverse ces six chapitres est entièrement fictif : comprendre pourquoi certains techniciens d'intervention produisent des comptes rendus incomplets, avant de décider si le produit doit évoluer.

5.1

La carte complète

J'ai laissé dans la partie 2 le Contexte consacré à la qualité des comptes rendus d'intervention :

- trois sources,
- deux notes atomiques,
- une note thématique,
- un guide pour le prochain entretien.

Je regardais alors ce qu'il permettait de comprendre.

J'ouvre maintenant le dossier lui-même.

Voici l'arborescence complète de la version que j'utilise dans ce livre :

```
contexte-qualite-comptes-rendus/
├── README.md           navigation générale, générée
├── offline.md          la boîte aux lettres hors séance
├──
├── 00-frame/           ce qui donne une direction durable
│   ├── README.md       navigation de la zone, générée
│   ├── mission.md      le but, la question, les limites
│   ├── directives.md   les règles propres à la mission
│   └── loading.md      ce qui sera lu à l'ouverture
├──
├── 01-state/           le présent du travail
│   ├── README.md       navigation de la zone, générée
│   ├── state.md        acquis, incertitudes, dettes
│   ├── synthesis.md    ce qui est devenu durable
│   ├── todo.md         les actions à accomplir
│   ├── decisions.md    les choix qui structurent le Contexte
│   ├── contradictions.md ce qui reste en tension
│   ├── handoff.md      le mode d'emploi de la reprise
│   └── checkpoints.md  la suite compacte des états
```

└─ health.md	le résultat du dernier contrôle
└─ 02-history/	la trace factuelle et datée
└─ └─ README.md	le rôle de cette trace, généré
└─ └─ YYYY-MM-DD.md	les événements de la journée
└─ 03-sources/	ce qui a été reçu ou formulé
└─ └─ README.md	la liste des sources, générée
└─ └─ ingestion-log.md	ce qui a été lu, ce qu'il a produit
└─ └─ documents/	
└─ └─ └─ README.md	la liste des documents, générée
└─ └─ └─ raw/	les fichiers originaux
└─ └─ └─ fulltext/	leur texte intégral exploitable
└─ └─ └─ SRC00001-....md	identité, provenance, état
└─ └─ sessions/	
└─ └─ └─ README.md	la liste des séances, générée
└─ └─ └─ SESSION-....md	la parole versée pendant le travail
└─ 04-notes/	la connaissance tirée des sources
└─ └─ README.md	la carte des notes, générée
└─ └─ atomic/	une idée par note – AT000001-....md
└─ └─ thematic/	un angle qui relie – THM00001-....md
└─ └─ glossary/	un terme stabilisé – GL000001-....md
└─ └─ decision-records/	un arbitrage – DRE00001-....md
└─ 05-outputs/	ce qui est produit pour un usage
└─ └─ README.md	la liste des résultats, générée
└─ └─ _draft/	les livrables en cours
└─ └─ questions-entretien.md	un guide pour poursuivre
└─ └─ note-de-cadrage.md	une synthèse pour décider
└─ └─ presentation-equipe.md	les mêmes acquis, reformulés

Les numéros imposent un ordre de lecture, pas une hiérarchie de valeur :

- le cadre et l'état orientent le travail ;
- l'histoire permet d'en contrôler le déroulement ;
- les sources gardent la matière, les notes la transforment en connaissance ;
- les livrables lui donnent une forme pour un destinataire.

Six zones, six fonctions

00-frame/ répond à la question « pourquoi suis-je ici, et sous quelles règles ? ». Cette zone change rarement, et pour cause : modifier la mission transformerait ma façon de juger toutes les sources, toutes les tâches et tous les résultats.

```

contexte-qualite-comptes-rendus/
├── 00-frame/                ce qui donne une direction durable
│   ├── README.md           navigation de la zone, générée
│   ├── mission.md          le but, la question, les limites
│   ├── directives.md       les règles propres à la mission
│   └── loading.md          ce qui sera lu à l'ouverture

```

01-state/ répond à « où en suis-je maintenant ? ». Cette zone bouge avec le travail, et me permet de distinguer un acquis d'une hypothèse, une action future d'une décision déjà prise, une difficulté passagère d'une contradiction qui mérite de rester visible.

```

contexte-qualite-comptes-rendus/
├── 01-state/                le présent du travail
│   ├── README.md           navigation de la zone, générée
│   ├── state.md            acquis, incertitudes, dettes
│   ├── synthesis.md        ce qui est devenu durable
│   ├── todo.md             les actions à accomplir
│   ├── decisions.md        les choix qui structurent le Contexte
│   ├── contradictions.md   ce qui reste en tension
│   ├── handoff.md          le mode d'emploi de la reprise
│   ├── checkpoints.md      la suite compacte des états
│   └── health.md           le résultat du dernier contrôle

```

02-history/ répond à « que s'est-il réellement passé ? ». Cette zone ne raconte pas joliment la séance : elle en

garde les faits utiles, une source ajoutée, une piste abandonnée, une décision prise, une note créée. Je la consulte rarement, mais elle devient décisive le jour où l'état présent me paraît incohérent.

```

contexte-qualite-comptes-rendus/
├── 02-history/                la trace factuelle et datée
│   ├── README.md             le rôle de cette trace, généré
│   └── YYYY-MM-DD.md         les événements de la journée

```

03-sources/ répond à « quelle matière ai-je reçue ? ». Un document extérieur y entre figé, ma propre parole y entre au fil des séances. Dans les deux cas, la source reste séparée de ce que le Contexte en comprendra.

```

contexte-qualite-comptes-rendus/
├── 03-sources/               ce qui a été reçu ou formulé
│   ├── README.md             la liste des sources, générée
│   ├── ingestion-log.md      ce qui a été lu, ce qu'il a produit
│   ├── documents/
│   │   ├── README.md         la liste des documents, générée
│   │   ├── raw/              les fichiers originaux
│   │   ├── fulltext/         leur texte intégral exploitable
│   │   └── SRC00001-....md   identité, provenance, état
│   └── sessions/
│       ├── README.md         la liste des séances, générée
│       └── SESSION-....md    la parole versée pendant le travail

```

La fiche SRC00001-....md ne contient ni le document ni son résumé. Elle répond à tout ce que j'ai besoin de savoir à son sujet :

- d'où il vient et pourquoi je l'ai ajouté ;
- où retrouver l'original et son texte intégral ;

- si l'extraction a réussi ;
- quel niveau de confiance lui accorder ;
- si sa lecture a déjà produit des notes.

Elle rend donc l'état de la source visible sans toucher à la matière elle-même.

04-notes/ répond à « qu'ai-je appris ? ».

- Une note atomique stabilise une idée.
- Une note thématique relie plusieurs idées.
- Le glossaire précise un mot.
- Un decision record garde les raisons d'un arbitrage.

```
contexte-qualite-comptes-rendus/
```

```
|— 04-notes/                la connaissance tirée des sources
|   |— README.md           la carte des notes, générée
|   |— atomic/             une idée par note – AT000001-....md
|   |— thematic/          un angle qui relie – THM00001-....md
|   |— glossary/          un terme stabilisé – GL000001-....md
|   |— decision-records/   un arbitrage – DRE00001-....md
```

05-outputs/ répond à « que vais-je faire de cette connaissance ? ». Les brouillons et les résultats y vivent, sous la forme qu'attend leur public :

- un guide d'entretien ;
- un article ;
- une spécification ;
- une présentation ;
- une décision, parmi d'autres formes possibles.

```

contexte-qualite-comptes-rendus/
└─ 05-outputs/                ce qui est produit pour un usage
    ├── README.md             la liste des résultats, générée
    ├── _draft/               les livrables en cours
    ├── questions-entretien.md un guide pour poursuivre
    ├── note-de-cadrage.md     une synthèse pour décider
    └─ presentation-equipe.md les mêmes acquis, reformulés

```

Le monde des idées : des briques pour plusieurs constructions

La séparation entre 04-notes/ et 05-outputs/ porte l'une des forces principales du Contexte. Les notes et le glossaire y forment un **monde des idées** : un ensemble de briques de connaissance, réutilisables à l'infini, qui ne sont plus la matière brute et pas encore un résultat destiné à quelqu'un.

```

03-sources/
  |
  ▼
04-notes/                le monde des idées
├─ AT000001             une idée stabilisée
├─ AT000002             une autre idée stabilisée
└─ THM00001             une lecture de leur relation
    |
    ├──> questions-entretien.md
    ├──> note-de-cadrage.md
    └─> presentation-equipe.md

```

Dans mon exemple, la friction de saisie et le besoin de données structurées ont été stabilisés avant que je sache sous quelle forme ils seraient présentés.

- le guide d'entretien en tire des questions,

- la note de cadrage en tire des enjeux,
- la présentation en tire un récit destiné à l'équipe.

Ces trois livrables ne possèdent pas la connaissance qu'ils utilisent.

- Ils en sélectionnent et en recomposent une partie pour un besoin, un public et un format particuliers.
- Un livrable n'épuise donc pas les notes dont il est fait : les briques restent disponibles pour une autre construction.

5

Sans ce monde intermédiaire, chaque nouveau résultat m'obligerait à rouvrir les sources, à refaire les rapprochements et à reformuler les mêmes acquis.

Le contexte ne conserve pas seulement ce qui a déjà été produit : il construit une capacité durable à produire de nouveau.

Tous les fichiers n'évoluent pas de la même manière

L'arbre ne devient vraiment lisible que lorsque je sais comment chacun de ses éléments évolue.

Certains fichiers sont réécrits pour montrer la meilleure situation disponible maintenant :

- `state.md` ;
- `handoff.md` ;
- les brouillons de `05-outputs/_draft/`.

D'autres ne font que s'allonger afin de conserver le passé :

- les sources de session ;
- l'histoire datée ;
- la synthèse ;
- la série des checkpoints.

Les documents reçus, eux, restent figés, et une note remplacée par une autre demeure consultable. Les fichiers `README.md`, enfin, sont générés à partir de ce qui existe : les corriger à la main créerait une deuxième version de la réalité, qui finirait par diverger.

Cette diversité explique pourquoi un seul grand fichier ne me suffisait plus. Il aurait fallu demander au même texte :

- de montrer le présent,
- de conserver le passé,
- de recevoir du brut,
- de stabiliser des idées,
- de porter un résultat fini.

Un squelette plutôt qu'une recette à recopier

Ce squelette n'est pas une documentation posée à côté du Contexte : c'est un vrai dossier, que je copie.

- Chaque fichier y arrive avec les sections qu'il doit contenir et quelques indications, si bien que deux

Contextes créés à plusieurs mois d'intervalle partagent les mêmes repères.

La structure reste adaptable pour autant. Un Contexte peut recevoir :

- une directive particulière,
- un livrable propre à son métier,
- une source d'un type inédit.

5

Le socle commun conserve surtout les trois frontières dont mon travail a besoin :

- le stable et le présent,
- le brut et l'interprété,
- la connaissance et sa mise en forme.

Parler au Contexte

Jusqu'ici, j'ai montré des dossiers et des fichiers. On pourrait en déduire que je dois ouvrir chacun d'eux, savoir où écrire et déplacer moi-même chaque information. Ce n'est pas ainsi que je travaille.

Je parle à l'IA :

Ouvre le Contexte consacré à la qualité des comptes rendus.

Ajoute ce document comme source.

Transforme cette source en notes.

| Sauvegarde le travail.

Derrière chacune de ces demandes se trouve une procédure que j'ai écrite une fois et que l'IA peut reprendre à chaque séance.

- Dans Claude Code et Codex, ce type de procédure réutilisable porte un nom : un **skill**.

Le skill d'ouverture indique par exemple :

- ce qu'il faut absorber avant de commencer ;
- quels fichiers lire systématiquement ;
- ce qui doit rester sur disque jusqu'à ce que la tâche l'exige ;
- comment signaler un problème ;
- quelle information me rendre pour que je puisse agir immédiatement.

Je n'ai pas besoin de redonner ces étapes chaque fois.

- Je demande simplement d'ouvrir le Contexte.

La conversation porte mon intention du moment.

- Le skill porte la manière fiable d'exécuter les gestes qui reviennent.

Mon système possède ainsi quelques verbes stables :

- créer un Contexte ;
- l'ouvrir et retrouver le travail en cours ;

- ajouter une source ou conserver une séance ;
- transformer les sources en notes ;
- sauvegarder l'état du travail ;
- contrôler sa cohérence ;
- l'archiver lorsqu'il a atteint son but.

Tous mes échanges avec l'IA ne passent pas par un skill. Je continue à réfléchir, questionner, corriger et arbitrer en langage courant.

5

Les skills encadrent seulement les opérations qui demandent de respecter toujours les mêmes étapes.

- Ils empêchent, par exemple, qu'une source soit transformée sans garder son origine ou qu'une séance se termine sans préparer la reprise.

La procédure reste stable, mais le jugement ne devient pas automatique. Deux modèles peuvent formuler une note différemment. Ils doivent néanmoins consulter les mêmes éléments, respecter les mêmes frontières et effectuer les mêmes contrôles.

Partager les skills ne donne donc pas à quelqu'un mon Contexte. Cela lui donne la même manière de créer et de faire vivre le sien.

Je n'en dirai pas plus sur leur mécanique.

Les skills ne sont ni une invention ni une particularité du Contexte. Claude Code et Codex les proposent précisément pour décrire des procédures réutilisables, et leur documentation explique mieux que je ne le ferais ici comment les créer et les utiliser.

Ce qui appartient au Contexte, c'est le choix des compétences qu'on lui donne : créer, ouvrir, alimenter, transformer, sauvegarder, contrôler et archiver. C'est aussi l'ordre dans lequel elles s'enchaînent et les frontières que chacune doit respecter.

Les fichiers conservent ce que le Contexte sait. Les skills organisent les gestes par lesquels il évolue.

La carte est ouverte. La visite peut maintenant commencer comme commence une vraie séance : en ouvrant le Contexte.

À RETENIR — LE SQUELETTE D'UN CONTEXTE

Un Contexte neuf arrive déjà meublé : six zones, dont chacune répond à une question différente. Les numéros imposent un ordre de lecture, pas une hiérarchie de valeur. Ce qui rend l'arbre lisible n'est pas sa forme, mais le fait de savoir comment chaque fichier évolue. Claude Code et Codex fournissent le mécanisme des skills ; le Contexte définit les gestes récurrents, leur ordre et leurs limites.

Ouvrir : retrouver la mission et le présent

Deux semaines ont passé depuis ma dernière séance sur les comptes rendus d'intervention — ces documents dans lesquels un technicien consigne ce qu'il a constaté et réalisé après une intervention sur un équipement.

- Les entretiens et les notes sont toujours là, à leur place.

Mais si je les rouvre au hasard, je vais passer une demi-heure à reconstituer ce que je savais déjà, et je le reconstituerais mal.

- L'ouverture commence donc par un noyau beaucoup plus petit : le cadre, l'état et la reprise.

Je n'ai pas à demander tout cela. J'appelle le skill d'ouverture, et il déroule sa procédure.

Je retrouve d'abord pourquoi je suis là

Le premier fichier que l'IA lit (via son skill d'ouverture de contexte) est `00-frame/mission.md`. Son modèle pose cinq questions :

```
# Mission

## Le but
Que doit permettre ce `Contexte` ?

## La question centrale
Quelle question guide le tri et l'exploration ?

## Ce qui est dans le sujet
Quelle matière peut entrer ?

## Ce qui est hors sujet
Que refuse-t-on, même si cela paraît intéressant ?

## Le livrable attendu
Que veut-on produire, si on le sait déjà ?
```

Dans le mien, le début est rempli ainsi :

```
## Le but
Comprendre ce qui dégrade la qualité des comptes rendus
d'intervention.

## La question centrale
Comment recueillir une information exploitable sans rendre
sa saisie dissuasive ?

## Ce qui est hors sujet
- Choisir une interface avant d'avoir confronté plusieurs
sources.
- Généraliser à partir d'un seul entretien.

## Le livrable attendu
Préparer les décisions produit et les prochains entretiens.
```

Une mission qui dirait seulement « améliorer les comptes rendus » ne m'orienterait pas beaucoup. La question centrale, elle, rend déjà certains documents plus pertinents que d'autres, et le hors-sujet m'empêche de

laisser une idée séduisante transformer silencieusement mon exploration en projet de solution.

- Le livrable, lui, peut rester inconnu.
- J'écris cette incertitude plutôt que d'inventer une sortie prématurée : le Contexte sait alors qu'il explore encore.

Les règles que j'avais posées m'attendent

Deux missions proches peuvent demander des précautions différentes. `directives.md` garde ces règles avec le travail :

- confidentialité,
- niveau de preuve,
- vocabulaire,
- validation ou forme attendue
- ...

Le modèle est volontairement simple : une règle, une date, une formulation impérative.

```
# Directives
```

- (2026-09-11) Anonymiser chaque entretien avant son intégration : remplacer le nom de l'intervenant par sa fonction, mais conserver le nom de l'entreprise et le pays.
- (2026-09-11) Distinguer ce que les sources montrent de ce que nous en déduisons.
- (2026-09-11) Ne proposer aucune solution avant d'avoir documenté les usages réels.

Une directive ne vaut que si elle peut guider une action future.

- « Être prudent » me laisse toute l'interprétation à refaire ;
- « distinguer l'observation de la déduction » me donne un contrôle précis au moment d'écrire une note ou un livrable.

Mais une directive n'est pas seulement une consigne que je retrouve en ouvrant un fichier. C'est une règle persistante que le Contexte doit appliquer chaque fois qu'une action entre dans son champ.

J'ajoute par exemple la transcription d'un nouvel entretien. Je demande simplement au Contexte de l'intégrer (via son skill dédié), sans penser à rappeler qu'elle contient le nom de la personne interrogée. La directive d'anonymisation intervient pourtant pendant l'ajout : avant d'enregistrer la transcription comme source, le skill remplace ce nom par la fonction de l'intervenant, tout en conservant le nom de l'entreprise et le pays.

Je n'ai pas à me souvenir de la règle au moment précis où elle devient nécessaire. Le Contexte s'en souvient pour moi.

Les directives ne cessent donc pas d’agir une fois l’ouverture terminée. Elles restent applicables pendant toute la séance et accompagnent les différents gestes du Contexte :

- lors de l’ajout d’une source, la règle d’anonymisation protège l’identité de l’intervenant ;
- lors de la création d’une note, la règle sur le niveau de preuve sépare ce que la source montre de ce que nous en déduisons ;
- lors de la préparation d’un livrable, la dernière directive empêche de présenter trop tôt une solution comme acquise.

Le skill qui exécute une action doit commencer par identifier les directives qui la concernent, puis vérifier que son résultat les respecte.

```
demande
  ↓
directives applicables
  ↓
action du skill
  ↓
contrôle du résultat
```

Les directives entourent ainsi l’action : elles la contraignent avant son exécution et servent à en contrôler le résultat après celle-ci. Elles n’agissent pas

seules et ne savent pas tout. Ce sont les skills qui les rendent opérantes au bon moment.

La mission indique où aller. Les directives accompagnent chaque geste du Contexte, afin que je n'aie pas à me rappeler, à chaque étape, les précautions déjà décidées.

Tout ne mérite pas d'être ouvert

Mon dossier contient des dizaines de sources et de notes. Les charger toutes à chaque ouverture noierait les priorités dans la masse. `loading.md` choisit donc un noyau stable :

```
## Toujours

- `00-frame/mission.md`
- `00-frame/directives.md`
- `01-state/state.md`
- `01-state/todo.md`
- `01-state/handoff.md`

## À la demande

- les sources nécessaires à une vérification
- les notes liées au sujet travaillé
- le brouillon actif
```

Tout reste accessible, mais tout n'est pas chargé dans le contexte de l'IA. `loading.md` désigne les éléments indispensables pour comprendre la mission, interpréter une demande et agir sans perdre les règles déjà établies.

- Le reste demeure sur disque et n'est consulté qu'en cas de besoin.
- Une vieille source peut être indispensable pour contrôler une affirmation sans occuper de place dans le contexte actif pendant toutes les séances.

Le contenu de `loading.md` dépend du Contexte.

- Un livre chargera son plan à chaque ouverture, une exploration produit sa note thématique centrale.
- Le principe reste le même :
 - commencer avec assez d'informations pour agir,
 - puis aller chercher le reste quand la tâche le justifie.

Où j'en étais vraiment

La mission reste stable ; `01-state/state.md`, lui, est fait pour changer. Il tient en trois blocs :

```
# État

## Ce qu'on sait
- La saisie différée entraîne une perte de détails dans le premier entretien.
- Certains champs obligatoires sont contournés dans le second.

## Ce qui reste incertain
- Nous ignorons quels champs sont réellement réutilisés après la saisie.

## Les dettes
- Le livre blanc a été ajouté mais son niveau de fiabilité reste à qualifier.
```

L'état ne raconte pas toute l'histoire et ne résume pas toutes les notes : il expose la meilleure compréhension dont je dispose du travail lui-même.

`state.md` s'adresse à la fois à l'IA et à l'humain.

Pour l'IA, il constitue une mémoire de travail persistante.

- Chargé à chaque ouverture, il lui indique les acquis qu'elle peut mobiliser, les incertitudes qu'elle doit préserver et les dettes qu'elle ne doit pas oublier.
- Sans lui, elle devrait reconstruire cet état à partir des sources et des notes à chaque nouvelle séance.

Pour moi, il joue le rôle d'un tableau de bord.

- Je peux vérifier en quelques lignes ce que le `Contexte` tient pour établi, ce qu'il considère encore comme incertain et les travaux qui restent incomplets.
- Je peux alors corriger cet état s'il ne correspond pas à ma compréhension.

Quelques dizaines de lignes suffisent ainsi à l'IA pour reprendre le travail sans confondre les acquis et les hypothèses, et à moi pour contrôler la compréhension sur laquelle elle s'appuie.

`todo.md` en extrait les actions :

```
## En cours
- T-001 – Préparer un entretien sur l’usage réel des champs
obligatoires.

## À faire
- T-002 – Vérifier quelles données servent aux audits.

## Fait
- T-003 – Comparer les deux premiers entretiens. – terminé
le 2026-09-10
```

Je n’ouvre pas nécessairement `todo.md` pour le modifier moi-même. Je peux simplement dire :

Ajoute une tâche pour vérifier quelles données servent aux audits.

Le Contexte crée une ligne, lui attribue un identifiant et la place dans la bonne section.

Une tâche peut aussi apparaître pendant le travail.

- Si l’analyse d’une source fait émerger une vérification nécessaire, l’IA me propose de l’ajouter plutôt que de la laisser disparaître dans la conversation.
- Elle ne transforme pas silencieusement chaque idée en nouvelle obligation : lorsqu’une tâche modifie les priorités ou engage un travail supplémentaire, je reste responsable de l’arbitrage.

Lorsque le travail est accompli, je peux demander :

Clôture la tâche T-002 (ou une description approximative de la tâche et l'IA va retrouver la bonne référence).

Le Contexte ne supprime pas la ligne. Il la déplace dans Fait, ajoute sa date de clôture et, lorsque c'est utile, indique le résultat ou l'élément qui permet de la considérer comme terminée.

- Une tâche terminée ne disparaît pas.
- Sa présence m'évite de refaire le même travail et m'explique pourquoi une piste a cessé d'être active.

todo.md n'est donc pas une liste écrite une fois pour toutes. C'est la file des actions du Contexte : l'IA la consulte pour savoir ce qui reste à faire, la met à jour lorsque le travail avance et conserve les tâches terminées pour éviter de les recommencer.

Par quoi je recommence

handoff.md est le dernier fichier du noyau. Il est réécrit à chaque sauvegarde pour répondre à une situation très concrète : quelqu'un arrive maintenant, une personne ou une IA, sans rien de la conversation précédente. Que doit-il savoir pour repartir sans se tromper ?

Reprise

Où on en est

Trois sources ont fait apparaître une tension entre facilité de saisie et exploitation.

Ce qu'il faut savoir avant de continuer

Les deux entretiens décrivent des situations particulières, tandis que le livre blanc défend un point de vue général. Ce corpus fait apparaître une tension, mais ne permet pas encore d'en mesurer l'ampleur ni de conclure qu'elle existe dans toutes les organisations.

La prochaine action

Préparer les questions qui permettront d'identifier les champs réellement utilisés.

Ce qui est resté ouvert

Le niveau de structure nécessaire aux audits reste inconnu.

5

La prochaine action est unique et immédiatement exécutable. Une liste de dix possibilités me rendrait le travail de priorisation que ce fichier existe précisément pour m'épargner.

La commande d'ouverture absorbe donc ce qui a été déposé pendant le sommeil du Contexte, lit ce noyau, puis m'annonce cette action.

- Il retrouve son but avant de retrouver ses détails, et mes deux semaines d'absence ne coûtent plus qu'une minute de lecture.
- La séance peut maintenant recevoir de la matière nouvelle sans perdre sa direction.

À RETENIR — LE NOYAU D'OUVERTURE

Rouvrir un Contexte ne consiste pas à relire son contenu, mais à charger un petit noyau : la mission, les directives, l'état et la reprise. La mission dit où aller, les directives déterminent les conditions du trajet. Une reprise utile désigne une seule action, immédiatement exécutable.

5.3

Alimenter : faire entrer la matière

La séance commence, et trois choses arrivent en même temps.

- Un responsable d'exploitation m'envoie le compte rendu d'un entretien,
- une idée me vient pendant l'échange,
- une note que j'avais laissée la veille attend dans une InBox (gestionnaire de note, email...).

Ces trois matières n'entrent pas dans le Contexte de la même manière, et chacune a son skill.

Un document arrive comme un objet fini

Un document externe est conservé avec son fichier d'origine, son texte exploitable et une fiche qui permet de le suivre.

```
03-sources/documents/  
├─ SRC00001-entretien-exploitation.md  
├─ raw/  
│   └─ SRC00001-entretien-exploitation.pdf  
└─ fulltext/  
    └─ SRC00001-entretien-exploitation.md
```

La fiche SRC00001-entretien-exploitation.md peut prendre cette forme :

```
---
added: 2026-09-11
kind: pdf
intention: entretien
raw: raw/SRC00001-entretien-exploitation.pdf
fulltext: fulltext/SRC00001-entretien-exploitation.md
extraction: ok
reliability: medium
---

# Entretien avec le responsable d'exploitation

## Pourquoi elle est ici

Comprendre ce qui empêche les techniciens de produire des
  comptes rendus exploitables.

## Interprétation – non analysée

Le responsable décrit une opposition entre la vitesse de
  saisie sur le terrain et le besoin de données structurées
  pour l'analyse.

## Couverture

- Saisie mobile
- Champs obligatoires
- Réutilisation des données

## Ce qu'elle a produit

- À compléter après analyse
```

Cette fiche ne résume pas forcément le document. Elle répond d'abord à des questions de gestion :

- d'où vient-il,
- pourquoi l'ai-je ajouté,
- puis-je lire son contenu,
- quelle confiance lui accorder ?

La section Interprétation – non analysée met en quarantaine une première lecture. Elle évite qu’une impression formulée au moment de l’import soit immédiatement traitée comme une connaissance établie. Lorsqu’une source arrive sous la forme d’un PDF, d’une page web ou d’un autre format difficile à exploiter directement, l’extraction en produit une version textuelle lisible par l’humain et par l’IA.

5

- Cette opération ne produit encore aucune connaissance : elle rend seulement la matière accessible sans modifier le document original.

La transformation vient ensuite.

- Le texte extrait est analysé pour faire apparaître des notes atomiques, des notes thématiques, des termes de glossaire ou des arbitrages.
- Ces notes alimentent le **monde des idées**, où la matière des sources devient un ensemble de briques de connaissance réutilisables.

Le niveau de fiabilité ne juge pas les sources ajoutées.

- Il indique simplement comment utiliser cette matière : témoignage isolé, donnée vérifiée, document contractuel ou hypothèse à confirmer.

Toute la matière utile n'existe pas encore dans un document. Une partie se trouve dans mon expérience, mes souvenirs, mes observations ou les distinctions que je formule en travaillant.

Lorsque je les exprime pendant une séance d'échanges avec l'IA, elles entrent elles aussi dans le Contexte.

- Elles ne deviennent pas automatiquement des faits établis
- Elles gardent leur origine et pourront être confrontées à d'autres sources.

Ajouter de la matière consiste donc autant à intégrer un document qu'à conserver ce que la personne qui possède le Contexte apporte à la conversation.

La connaissance de la personne entre par la séance

Une session n'est pas un document livré une fois pour toutes. Elle se construit au fil de l'échange. Elle est enregistrée dans `03-sources/sessions/SESSION-2026-09-12.md`, qui peut prendre cette forme :

Session du 12 septembre 2026

09:41 – IA

Les champs obligatoires améliorent-ils réellement la qualité, ou déplacent-ils seulement l'effort vers le technicien ?

09:42 – Auteur

Dans les entretiens, les champs obligatoires garantissent surtout qu'une valeur a été saisie.

Ils ne garantissent ni sa précision ni son utilité.

Lorsque le technicien ne comprend pas l'usage futur d'un champ, il le remplit au plus vite pour poursuivre.

La contrainte améliore donc la complétude apparente, mais peut déplacer l'effort sans améliorer l'information réellement exploitable.

...

5

Tout ce que dit l'IA n'a pas besoin d'être conservé. En revanche, une question, une objection ou une reformulation qui change le travail mérite de l'être.

- Ma parole et celle de l'IA restent séparées.
- Je peux ainsi retrouver l'origine d'une idée et distinguer mon intention d'une proposition du système.

Une session peut être désordonnée.

- Son rôle est de capturer le mouvement réel de la pensée, pas de présenter une synthèse parfaite.

La boîte aux lettres parle au Contexte lorsqu'il est fermé

Une idée me vient hors séance, dans le train ou au milieu d'une réunion et sans accès à Internet.

- Elle entre alors dans la boîte aux lettres.
- Cette boîte aux lettres est le fichier `offline.md`, placé à la racine du Contexte.
- Il me suffit d'écrire dans ce fichier Markdown.

Il peut prendre cette forme :

Demander si les techniciens contournent les champs obligatoires avec des valeurs génériques.

À la prochaine ouverture du contexte, cette note est relue, rapprochée du travail en cours puis absorbée au bon endroit. Elle peut devenir une tâche, une question d'entretien, une note ou disparaître si elle n'est plus pertinente.

Ajouter n'est pas encore comprendre

Une fois cette matière entrée, mon arbre contient plusieurs sources et la trace de la séance en cours.

```
03-sources/  
├─ documents/  
│   ├── SRC00001-entretien-exploitation.md  
│   ├── SRC00002-entretien-exploitation.md  
│   └── SRC00003-livre-blanc-donnees-intervention.md  
└─ sessions/  
    └── SESSION-2026-09-11.md
```

Ces fichiers prouvent que la matière est disponible. Ils ne prouvent pas qu'elle a été comprise.

Le fichier `03-sources/ingestion-log.md` distingue les deux opérations. Pour chaque source, il indique si l'analyse est complète, partielle ou à revoir, avec une empreinte du contenu et quelques notes de traitement.

Cette distinction devient essentielle quand un document est remplacé, qu'une extraction change ou qu'une séance s'interrompt. Le Contexte peut alors reprendre l'analyse sans faire semblant de tout savoir.

5

La matière est entrée. Il me reste à la transformer en idées assez précises pour être reliées, discutées et réutilisées.

À RETENIR — LA SOURCE

Un document, une parole de séance et une note hors séance entrent par trois portes différentes et gardent chacune leur forme d'origine. Une source conserve la matière et la preuve, jamais l'interprétation qu'on en tirera. Ajouter n'est pas comprendre : le suivi d'ingestion distingue ce qui est présent de ce qui est assimilé.

5.4

Transformer : passer des sources aux idées

J'ai maintenant trois sources devant moi.

- Deux entretiens décrivent la friction provoquée par les formulaires,
- Un livre blanc défend la nécessité de données structurées.

La tentation serait de trancher tout de suite et d'écrire que les champs obligatoires nuisent à la qualité. Ce serait aller trop vite : le Contexte doit préserver ces apports sans les fondre dans une conclusion unique.

J'appelle donc le skill qui crée et met à jour les notes : il lit ce qui n'a pas encore été traité et en tire des idées.

J'extrais une idée qui tient debout toute seule

Une source contient rarement une idée déjà prête à être réutilisée.

- Un entretien mêle des faits, des exemples, des opinions, des hésitations et des détails propres à la situation racontée.
- Le transformer ne consiste pas à tout résumer, mais à en extraire les idées.

Le chemin tient en trois étapes :

Témoignage	« Quand le formulaire est trop long, certains techniciens mettent une valeur générique pour pouvoir terminer. »
	↓
Observation	Des valeurs génériques sont parfois utilisées pour accélérer la saisie.
	↓
Idée stabilisée	Une saisie trop contraignante peut appauvrir les comptes rendus.

5

Une idée forte n'est pas une formule spectaculaire ni une vérité définitive.

- Elle possède un centre de gravité clair et fort,
- Peut être comprise hors du texte qui l'a fait naître
- Reste assez délimitée pour être confirmée, nuancée ou contredite.
- Elle conserve également son origine et ses limites afin de ne pas transformer un témoignage particulier en règle générale.

L'IA lit la source à la lumière de la mission et des directives, puis la confronte aux notes déjà présentes. Elle crée une nouvelle note lorsque l'idée est nouvelle ; sinon, elle enrichit l'existante avec la nuance ou la limite supplémentaire.

Elle effectue ce travail sans me demander de valider chaque formulation. Je peux ensuite relire et corriger les notes.

- Lorsqu'une ambiguïté apparaît pendant la séance, elle m'interroge ;
- Si elle découvre plus tard une contradiction qu'elle ne peut résoudre, elle la conserve au lieu de trancher à ma place.

```
04-notes/  
└─ atomic/  
   └─ AT000001-la-contrainte-de-saisie-peut-appauvrir-les-comptes-rendus.md
```

```
---
id: AT000001
status: active
sources: [SRC00001, SRC00002]
---

# Une saisie trop contraignante peut appauvrir les comptes
rendus

## Idée principale

Lorsque le formulaire ralentit l'intervention, les
techniciens peuvent saisir des valeurs génériques ou
réduire leur description au minimum attendu.

## Pourquoi c'est important

Une donnée formellement structurée peut sembler propre tout
en décrivant moins bien la réalité du terrain.

## Nuances et limites

Les deux entretiens ne permettent pas de mesurer la
fréquence du phénomène.

## Origine / preuve

- SRC00001 : exemples de valeurs génériques utilisées pour
terminer plus vite.
- SRC00002 : baisse de précision observée après l'ajout de
champs obligatoires.

## Liens

- Contraste avec AT000002.
```

Le titre porte une proposition, pas seulement un thème. La note reste compréhensible sans que j'aie à rouvrir les documents dont elle provient, tout en conservant les liens qui me permettent d'en vérifier l'origine.

Une note atomique n'est pas figée au premier jet.

- Elle peut gagner une nuance, une nouvelle preuve ou une relation sans perdre son identité.

Le livre blanc produit une seconde note : les données structurées sont nécessaires pour comparer, agréger et automatiser.

- Cette idée reste séparée de la première parce qu'elle affirme autre chose.

Je refuse de trancher trop tôt

Les deux idées paraissent s'opposer, et je n'ai pas encore assez de matière pour les arbitrer. J'ouvre donc une contradiction.

C-001 – Structurer sans dégrader la saisie

- D'un côté : les champs imposés peuvent appauvrir les descriptions terrain.
- De l'autre : l'exploitation à l'échelle exige des données comparables.
- Enjeu : déterminer quelles informations doivent être structurées au moment de la saisie et lesquelles peuvent l'être ensuite.
- Ouverte le : 2026-09-11

La contradiction m'évite deux erreurs symétriques : choisir trop tôt un camp, ou lisser le désaccord dans une formulation vague qui ne gêne personne.

- Elle reste ouverte jusqu'à ce qu'une décision, une nouvelle source ou une reformulation la résolve.

Toute incertitude ne mérite pas ce traitement.

- Je le réserve aux tensions qui pèsent réellement sur la suite du raisonnement ou sur la qualité d'une décision.

Je relie ce que ces idées disent ensemble

Pendant l'ingestion, l'IA ne crée pas seulement des notes atomiques.

- Elle examine aussi les relations entre les idées extraites et les notes déjà présentes.
- Lorsque plusieurs d'entre elles forment un ensemble qui ne peut pas être réduit à une seule proposition, elle crée dans la même opération une note thématique et pose les liens qui les relient.

Je n'ai pas à lancer une seconde commande.

- La création des notes atomiques, leur rapprochement et l'apparition éventuelle d'une note thématique font partie d'une même ingestion.

```
---
id: THM00001
status: active
sources: [AT000001, AT000002]
---

# La qualité dépend de la saisie et de l'exploitation

## Angle

La qualité d'un compte rendu ne peut pas être évaluée
uniquement par le nombre de champs remplis.

## Synthèse

Le système doit obtenir assez de structure pour exploiter
les données sans rendre la saisie artificielle ou
coûteuse.

## Notes reliées

- AT000001 – La contrainte de saisie peut appauvrir les
comptes rendus.
- AT000002 – Des données structurées sont nécessaires à
l'exploitation.

## Tensions

- C-001 – Structurer sans dégrader la saisie.
```

La note thématique ne juxtapose pas seulement des résumés. Elle propose une lecture de leurs relations et montre ce qui reste ouvert.

Je m'aperçois qu'un mot n'a pas le même sens pour tout le monde

Une partie du désaccord vient parfois du vocabulaire. Le glossaire fixe alors le sens retenu dans ce Contexte.

Donnée structurée

Information enregistrée selon une forme explicite qui permet sa comparaison ou son traitement automatique.

À ne pas confondre avec : champ obligatoirement rempli par l'utilisateur au moment de l'intervention.

Exemples : type de panne, durée d'arrêt, équipement concerné.

Cette définition locale n'a pas vocation à devenir universelle.

- Elle empêche simplement deux notes, ou deux personnes, d'employer le même mot pour parler de choses différentes.

Des briques neutres face au livrable

Une note appartient au **monde des idées** tant qu'elle cherche à stabiliser une connaissance plutôt qu'à produire un effet sur un public.

- Elle ne doit donc pas devenir trop tôt une accroche d'article, une slide, un argument commercial ou un morceau de spécification.

Cette neutralité ne signifie pas que la note reste vague.

- Au contraire, son titre affirme une proposition précise, son contenu en expose les nuances et ses liens permettent d'en contrôler les fondations.

- Ce qui reste neutre, c'est sa forme : elle n'est pas organisée pour servir un seul résultat.

AT000001 pourra ainsi devenir une question sur les contournements dans le guide d'entretien, un risque dans la note de cadrage ou un exemple dans la présentation d'équipe. AT000002 pourra soutenir une question sur les usages de la donnée, justifier une exigence produit ou expliquer à l'équipe pourquoi le texte libre ne suffit pas toujours.

La note thématique ne fusionne pas ces briques dans un texte définitif.

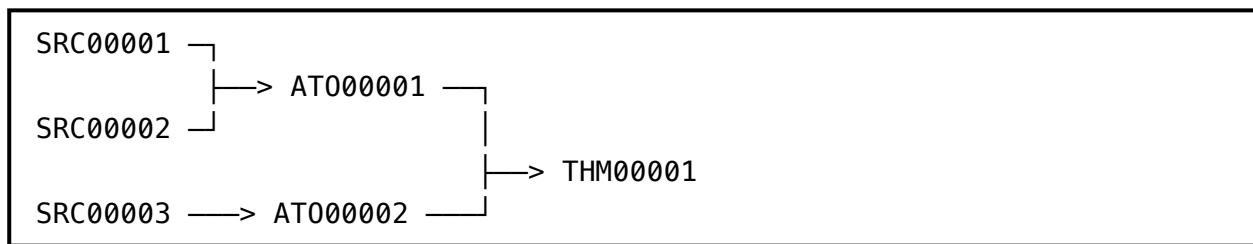
- Elle indique ce qu'elles permettent de comprendre ensemble et conserve la tension qui devra être traitée différemment selon le livrable.

Cette couche intermédiaire évite de recommencer l'analyse pour chaque production.

- Les sources restent les preuves,
- les notes deviennent des connaissances mobilisables,
- chaque livrable compose avec elles selon son propre besoin.

Je peux remonter de chaque idée à sa preuve

À ce stade, les relations se représentent ainsi :



Les liens nommés me permettent de distinguer une source qui confirme, nuance ou contredit une idée.

- Un niveau de confiance rend visible ce qui repose sur plusieurs observations et ce qui reste hypothétique.

5

La couverture d'ingestion montre quelles sources ont réellement contribué aux notes. Elle m'empêche de considérer comme assimilé un document simplement présent dans le dossier.

Ces notes ne sont pas écrites pour un livrable particulier.

- Elles forment une connaissance commune qui pourra alimenter plusieurs résultats sans être recopiée ni enfermée dans l'un d'eux.

****Le monde des idées** est précisément ce capital réutilisable : un ensemble de briques suffisamment stables pour être retrouvées et suffisamment indépendantes pour être recombinaées.**

À RETENIR — LA NOTE ATOMIQUE

Une idée stabilisée porte une proposition et non un thème, et se comprend sans rouvrir le document dont elle provient.

Une contradiction se conserve tant qu'elle pèse sur le raisonnement, plutôt que d'être tranchée trop tôt ou lissée.

Chaque note garde le lien vers ses sources, ce qui permet de remonter de l'idée à sa preuve.

5.5

Produire et décider

Ma note thématique a fait apparaître une question précise : quelles informations faut-il renseigner à la fin de l'intervention, et lesquelles peuvent être complétées ensuite ?

5

Trois personnes attendent maintenant quelque chose de moi.

- Un technicien que je dois interroger,
- un comité qui doit décider,
- une équipe à qui je dois expliquer.

Je ne vais pas mener trois enquêtes : je vais puiser trois fois dans les mêmes notes.

C'est le seul moment de la séance où aucun skill ne m'aide : produire un livrable est un travail d'écriture, pas un geste qui se répète.

Je puise trois fois dans le même stock

Le passage des notes aux livrables n'est pas une copie. Chaque résultat sélectionne les briques qui lui sont utiles, les ordonne et les reformule en fonction de son destinataire.

Brique commune	Guide d'entretien	Note de cadrage	Présentation d'équipe
AT000001 — Une saisie trop contraignante peut appauvrir les comptes rendus	Demander quels champs sont contournés et comment	Décrire le risque de perte de qualité	Montrer un exemple de friction terrain
AT000002 — Des données structurées sont nécessaires à l'exploitation	Demander quelles données sont réellement réutilisées	Établir le besoin de comparaison et d'automatisation	Expliquer la valeur attendue pour l'équipe
THM00001 — La qualité dépend de la saisie et de l'exploitation	Organiser l'entretien autour des deux exigences	Formuler l'enjeu et les critères d'arbitrage	Construire le fil du récit
C-001 — Structurer sans	Transformer la tension en	La conserver comme	La rendre visible sans annoncer

Brique commune	Guide d'entretien	Note de cadrage	Présentation d'équipe
dégrader la saisie	questions à vérifier	problème à résoudre	une solution prématurée

Le même stock produit ainsi trois assemblages, sans qu'aucun livrable ne reprenne tout ni ne remplace les notes qui l'alimentent.

5

Cette séparation me permet surtout de corriger la connaissance une seule fois.

- Si une nouvelle source nuance AT000001, mes prochains livrables repartiront de cette brique enrichie, au lieu de reproduire une ancienne formulation dispersée dans plusieurs documents.

Je commence par le plus urgent : le prochain entretien

Mon premier besoin est de préparer l'entretien avec un technicien. Le livrable commence dans l'espace des brouillons.

```
05-outputs/
├── _draft/
│   └── questions-prochain-entretien.md
```

Questions pour le prochain entretien

1. Quels champs vous ralentissent le plus pendant une intervention ?
2. Que saisissez-vous quand l'information demandée n'est pas disponible ?
3. Quelles informations pourriez-vous ajouter après l'intervention ?
4. Quelles données doivent être exactes immédiatement pour être utiles à l'exploitation ?

Les notes cherchent à conserver une idée réutilisable. Le livrable, lui, choisit un destinataire, une forme et une action attendue.

Une fois relu et validé, le fichier quitte `_draft/` pour rejoindre directement `05-outputs/`. Le déplacement indique un niveau de maturité, pas une différence de genre documentaire.

Jusqu'ici, je n'ai fait que produire : j'ai mis en forme ce que je savais déjà, pour trois publics. La suite de ce chapitre relève d'un tout autre geste. À un moment, il faut arrêter de restituer et trancher — et ce que je tranche, il faut pouvoir le retrouver dans six mois.

Le decision record garde le choix qui façonne le résultat

« Interroger un technicien avant de rédiger une spécification » n'est pas un arbitrage sur ce qui sera livré.

- Si cet entretien est simplement la prochaine action, il appartient à `01-state/todo.md`.
- Si tous les travaux de découverte doivent respecter cette règle, elle devient une directive dans `00-frame/directives.md`.

Dans les deux cas, elle ne justifie pas un decision record. Le decision record apparaît lorsque le travail fait émerger plusieurs directions possibles et qu'il faut choisir celle qui structurera le produit, le texte, la méthode ou un autre résultat du Contexte. Il conserve alors ce que l'on a décidé de concevoir ou de livrer, les alternatives écartées, les raisons du choix et les conséquences acceptées.

Dans mon exemple, l'arbitrage ne porte donc pas sur l'organisation des entretiens. Il porte sur une décision fonctionnelle :

- imposer toutes les informations à la fin de l'intervention,
- conserver uniquement du texte libre,
- ou permettre que le compte rendu soit complété dans un second temps.

Un decision record utile ne délègue pas le cœur du choix à la personne qui le mettra en œuvre. Trois personnes qui le lisent doivent comprendre le même périmètre, les

mêmes comportements attendus et les mêmes limites. Les détails techniques non structurants peuvent rester ouverts, mais aucun d’eux ne doit pouvoir modifier silencieusement la décision.

```
04-notes/  
└─ decision-records/  
    └─ DRE00001-limiter-la-cloture-a-quatre-informations.md
```

Voici ce que ça donne une fois écrit. Le modèle est long parce qu’il conserve tout le raisonnement qui rend la décision compréhensible et révisable — un lecteur déjà familier des decision records peut sauter directement au commentaire qui le suit.

5

```
---  
uuid: 0d5efbe2-1eac-4ad7-94c1-29f68adcf001  
id: DRE00001  
created: 2026-09-11  
updated: 2026-09-11  
status: accepted  
decided_by: équipe produit  
decided_on: 2026-09-11  
supersedes: null  
superseded_by: null  
sources: [AT000001, AT000002, THM00001]  
---  
  
# Limiter la clôture d’une intervention à quatre  
  informations obligatoires  
  
## Décision  
  
Pour qu’une intervention soit considérée comme clôturée,  
  exactement quatre informations doivent être présentes :
```

1. L'équipement concerné, déjà indiqué sur la fiche d'intervention ou renseigné au moment de la clôture.
2. La date et l'heure réelles de fin de l'intervention.
3. L'état de l'équipement à la fin de l'intervention : remis en service, maintenu en fonctionnement dégradé ou laissé hors service.
4. Un résumé non vide des travaux réalisés ou de la raison pour laquelle l'équipement n'a pas pu être remis en service.

Aucune autre information ne conditionne la clôture.

Après la clôture, le compte rendu peut être complété avec la nature de la défaillance, sa cause identifiée, la nature des travaux réalisés, les pièces remplacées et la durée d'indisponibilité de l'équipement, sans remettre en cause la clôture de l'intervention.

Le technicien peut renseigner ces informations complémentaires, mais le responsable d'exploitation doit les compléter ou les valider dans les vingt-quatre heures.

Un compte rendu encore incomplet reste consultable.

Il est signalé comme restant à compléter, puis le responsable d'exploitation est alerté après vingt-quatre heures.

Contexte

Les techniciens doivent pouvoir clôturer leur travail sur le terrain sans parcourir un formulaire d'analyse complet.

Le service d'exploitation a néanmoins besoin d'informations comparables pour analyser les défaillances, leurs causes, les travaux réalisés, les pièces remplacées et les durées d'indisponibilité.

Hypothèses

- La fiche d'intervention identifie normalement l'équipement avant le début des travaux.
- Le technicien connaît l'état de l'équipement et peut décrire brièvement les travaux réalisés au moment de la clôture.

- Le responsable d'exploitation dispose du temps et des informations nécessaires pour valider le compte rendu complet dans les vingt-quatre heures.

Alternatives

- Exiger les neuf informations au moment de la clôture.
- Ne conserver qu'un résumé libre et en extraire automatiquement les informations d'intervention après coup.
- Rendre toutes les informations facultatives et contrôler uniquement la qualité a posteriori.

Raisons

Les quatre informations immédiates suffisent pour identifier l'intervention, dater sa fin, connaître l'état de l'équipement et comprendre les travaux réalisés.

Les cinq informations d'analyse restent nécessaires, mais elles ne doivent pas empêcher le technicien de terminer son travail sur le terrain.

Le second temps attribue explicitement au responsable d'exploitation la responsabilité de compléter ou de valider le compte rendu.

Base

- AT000001 décrit les contournements liés à une saisie trop contrainte.
- AT000002 établit le besoin de données comparables.
- THM00001 relie les deux exigences.

Conséquences

Positives

- La clôture ne demande que quatre informations connues immédiatement.
- Les informations nécessaires aux analyses des interventions restent disponibles après que le compte rendu a été complété.
- La responsabilité des données manquantes est visible.

Arbitrages

- Les indicateurs d'intervention peuvent rester incomplets pendant les vingt-quatre heures suivant une intervention.
- Le responsable d'exploitation reçoit une charge de validation supplémentaire.

Risques

- Des comptes rendus incomplets peuvent s'accumuler.
- Des informations ajoutées après coup peuvent refléter moins fidèlement la situation constatée sur le terrain.

Critères de conformité

- Une intervention ne peut pas être clôturée si l'une des quatre informations immédiates manque.
- Une intervention peut être clôturée sans aucune des cinq informations complémentaires.
- Les informations complémentaires peuvent être modifiées sans changer l'état de clôture de l'intervention.
- Le compte rendu est présenté comme « restant à compléter » ou « complet et validé » dans tous les usages qui en sont faits.
- Le responsable d'exploitation est alerté vingt-quatre heures après la clôture si le compte rendu n'est pas complet et validé.

Décisions liées

- Résout C-001 – Structurer sans dégrader la saisie.

Sources

- SRC00001
- SRC00002
- SRC00003

Le statut accepted indique ici que l'arbitrage est validé et suffisamment précis pour guider l'implémentation.

- Tant que les informations attendues, les responsabilités ou les critères de conformité restent

indéterminés, le decision record doit conserver le statut `proposed`.

- Une fois accepté, il est figé comme trace historique et ne change plus silencieusement.

Si une nouvelle décision le remplace, l'ancien decision record reste consultable et les deux fichiers sont reliés par `supersedes` et `superseded_by`.

Cette continuité conserve la chronologie complète des arbitrages : ce qui avait été décidé, sur quelles informations, puis quel élément a conduit à choisir autrement.

Six mois plus tard, je ne retrouve donc pas seulement la règle actuelle. Je peux comprendre pourquoi une première direction avait paru juste, ce qui l'a rendue insuffisante et comment la décision suivante s'est construite.

Sans cette chronologie, une succession de choix finit par ressembler à une incohérence. Avec elle, elle redevient l'histoire compréhensible d'un produit qui évolue.

- Les alternatives montrent ce qui a été écarté.
- Les arbitrages rendent visible ce qui a été sacrifié.
- La base relie la décision aux connaissances et aux preuves qui la soutiennent.

Produire n'épuise pas ce que je sais

Le guide d'entretien, la note thématique et le decision record traitent du même problème sans faire doublon, parce qu'ils remplissent trois fonctions différentes.

- La note conserve ce que le Contexte croit savoir.
- La décision conserve un choix et ses raisons
- Le guide transforme cette connaissance en action auprès d'un destinataire précis.

5

Un autre livrable pourra reprendre la même connaissance commune sous une forme différente. **Produire ne vide donc pas le Contexte de sa substance : cela révèle ce que cette substance permet de faire.**

Il me reste à enregistrer l'état atteint, pour que la prochaine séance reprenne sans avoir à reconstituer le chemin parcouru.

À RETENIR — LE DECISION RECORD

Les mêmes notes alimentent plusieurs livrables, et aucun livrable ne consomme la connaissance qu'il utilise. Un decision record garde un arbitrage qui façonne ce qui sera conçu ou livré, pas une action à mener. Il conserve ce qui a été écarté et ce qui a été sacrifié, pour que le choix reste révisable.

5.6

Sauvegarder et reprendre

Le decision record vient d'être accepté et une réunion commence dans trois minutes. Je n'ai fini ni l'étude, ni même une étape qui ressemblerait à une fin.

C'est le cas ordinaire : une autre tâche réclame mon attention, la machine doit redémarrer, je change d'ordinateur, ou la fenêtre de contexte de l'IA devient trop chargée.

- Sauvegarder ne répond donc pas à un besoin du travail lui-même, du genre « l'étude est terminée ».
- C'est une garantie technique : l'état du Contexte doit survivre à la session dans laquelle il a été produit.

Je dois pouvoir m'interrompre maintenant, rouvrir plus tard et retrouver le travail exactement là où je l'ai laissé.

Un skill s'en charge, et il doit tenir en quelques secondes : une sauvegarde qui coûte cher ne sera pas lancée au moment où l'on est pressé, c'est-à-dire au moment où elle sert le plus.

L'IA fige ce que l'on sait à cet instant

Le fichier `01-state/state.md` résume ce qui est suffisamment stable pour orienter la suite.

```
contexte-qualite-comptes-rendus/  
└─ 01-state/  
   └─ state.md
```

À l'instant de la sauvegarde, il peut prendre cette forme :

```
# État courant  
  
## Connaissance stabilisée  
  
- Une saisie trop contraignante peut appauvrir les comptes  
  rendus.  
- L'exploitation à l'échelle exige néanmoins des données  
  structurées.  
  
## Décision actée  
  
- La clôture exige l'équipement concerné, la date et  
  l'heure de fin, l'état de l'équipement et un résumé des  
  travaux réalisés.  
- La nature et la cause de la défaillance, la nature des  
  travaux, les pièces remplacées et la durée  
  d'indisponibilité sont complétées après la clôture et  
  validées sous vingt-quatre heures par le responsable  
  d'exploitation.  
  
## Incertitude principale  
  
- Mesurer si cette séparation réduit réellement le temps de  
  clôture sans dégrader la qualité des données.  
  
## Dette de connaissance  
  
- Aucun pilote n'a encore mesuré le temps de clôture ni la  
  proportion de comptes rendus complets après vingt-quatre  
  heures.
```

- Une tâche terminée peut disparaître de la todo.
- Une contradiction non résolue reste visible dans l'état.

La synthèse ne contient que ce que le travail a suffisamment stabilisé. L'état est réécrit pour décrire le présent, tandis que la synthèse s'enrichit progressivement pour conserver la compréhension acquise.

Je prépare une seule action pour mon futur moi

Le fichier de reprise ne raconte pas toute l'histoire : il prépare le prochain geste utile.

```
contexte-qualite-comptes-rendus/  
└─ 01-state/  
   └─ handoff.md
```

Le fichier `01-state/handoff.md` peut prendre cette forme :

Reprise

État actuel

DRE00001 est accepté et C-001 est résolue.

Les quatre informations de clôture et les cinq informations complémentaires sont fixées.

Point de vigilance

Ne pas ajouter une information obligatoire à la clôture sans ouvrir un nouveau decision record qui remplace DRE00001.

Prochaine action

Démarrer la mise en œuvre de DRE00001.

Ouvert

- Définir les libellés présentés aux techniciens pour les trois états possibles de l'équipement après l'intervention.
- Préparer la mesure du temps de clôture et de la proportion de comptes rendus complets après vingt-quatre heures.

5

Une reprise utile désigne une seule prochaine action.

Une liste de dix possibilités obligerait la séance suivante à refaire le travail de décision que je viens justement de faire.

Je garde une trace pour comparer mes reprises

Le fichier 01-state/checkpoints.md conserve une suite de photographies compactes de la situation.

```
contexte-qualite-comptes-rendus/  
└─ 01-state/  
   └─ checkpoints.md
```

Le fichier peut prendre cette forme :

```
## 2026-09-11  
  
- Sources : 3  
- Notes atomiques : 2  
- Notes thématiques : 1  
- Contradictions ouvertes : 0  
- Decision records acceptés : 1  
- Prochaine action : démarrer la mise en œuvre de DRE00001.  
  
## 2026-09-04  
  
- Sources : 1  
- Notes atomiques : 1  
- Notes thématiques : 0  
- Contradictions ouvertes : 0  
- Prochaine action : obtenir le point de vue de  
  l'exploitation.
```

5

Ces photographies me montrent l'évolution du Contexte sans m'obliger à comparer tous les fichiers. Elles me font aussi repérer une stagnation, ou une dette qui grandit sans que je m'en rende compte.

L'IA note ce qui s'est passé, pas ce qu'elle en pense

L'historique enregistre les événements importants sans chercher à reconstituer chaque échange. Dans mon exemple, les événements de la journée sont ajoutés à 02-history/2026-09-11.md.

```
contexte-qualite-comptes-rendus/  
└─ 02-history/  
   └─ 2026-09-11.md
```

Le bloc ajouté peut prendre cette forme :

```
## 2026-09-11  
  
- Ajout de SRC00003-livre-blanc-donnees-intervention.  
- Création de AT000002 et THM00001.  
- Ouverture de C-001.  
- Validation de `questions-prochain-entretien.md`.  
- Acceptation de DRE00001 et résolution de C-001.
```

5

- L'état dit où l'on en est.
- L'historique dit ce qui s'est passé.
- La reprise dit quoi faire ensuite.

Je laisse la machine vérifier ce que je ne verrais pas

Au moment de sauvegarder, quelques contrôles mécaniques réduisent les oublis. Leur résultat est écrit dans 01-state/health.md.

```
contexte-qualite-comptes-rendus/  
└─ 01-state/  
   └─ health.md
```

- Tous les fichiers référencés existent-ils ?
- Les identifiants sont-ils uniques ?
- Chaque note indique-t-elle ses sources ?
- Une source ajoutée mais non analysée est-elle signalée ?

- Les décisions remplacées sont-elles reliées à celles qui les remplacent ?
- La prochaine action est-elle explicite ?

Ces vérifications ne remplacent pas mon jugement. Elles empêchent simplement une erreur de structure de se faire passer pour un problème de pensée.

Le cycle complet, et ce qu'il autorise

Le mouvement général devient maintenant visible.

5

```
ouverture
  ↓
lecture de la boîte aux lettres
  ↓
chargement de la mission, de l'état et de la reprise
  ↓
ajout d'une source
  ↓
création ou mise à jour des notes
  ↓
production d'un brouillon
  ↓
contrôle, validation ou décision
  ↓
mise à jour de l'état, de la synthèse et de la reprise
  ↓
checkpoint et historique
  ↓
contrôle de sauvegarde
  ↓
prochaine ouverture
```

Je n'attends pas la fin de cette chaîne pour sauvegarder.

- Je le fais après l'ajout d'une source, au milieu de la création d'une note, avant une décision ou pendant la

préparation d'un livrable, quand je vois ma fenêtre de contexte exploser...

C'est bien là le rôle de la sauvegarde : **rendre chacun de ces états reprenable, indépendamment de la conversation en cours.**

Ma réunion peut donc commencer.

Le Contexte n'est pas seulement une arborescence de fichiers : c'est un système pour penser dans la durée, qui fait entrer la matière, préserve les tensions, stabilise des idées, produit plusieurs résultats et prépare la reprise.

Sa forme reste volontairement simple et portable — des fichiers lisibles suffisent à rendre le travail observable, vérifiable et transmissible.

Une fois ce moteur compris, une autre question apparaît : comment le Contexte dialogue-t-il avec les outils, les bases de connaissances et le produit réel qui existent hors de lui ?

À RETENIR — LA SAUVEGARDE

Sauvegarder ne marque pas la fin d'une étape : c'est la garantie que l'état survit à la session qui l'a produit. L'état dit où l'on en est, l'historique ce qui s'est passé, la reprise quoi faire ensuite. Le cycle peut être interrompu à n'importe quel point, et chacun de ces points reste reprenable.

Partie 6

Le Contexte dans son écosystème

Un Contexte bien tenu donne envie d'y mettre toute la connaissance disponible.

Cette partie regarde ce qu'il laisse dehors, et pourquoi il l'y laisse. Il reste petit pour rester capable de penser, et ce qu'il garderait par précaution vieillirait sans prévenir pendant que l'original continue de changer ailleurs. Il travaille donc avec des voisins : des outils dont le métier est de conserver, de mesurer ou d'exécuter, et d'autres Contextes qui instruisent d'autres questions que la sienne.

6.1

Interroger les connaissances structurées

Les comptes rendus d'intervention attendront : ce matin, je travaille sur un autre sujet du même produit, les personnes qui abandonnent en cours d'inscription. Une agence vient de me livrer son étude et je l'ajoute au

Contexte : plusieurs participants hésitent devant les règles du mot de passe.

Cette observation me rappelle une idée déjà rencontrée : demander un effort important avant que l'utilisateur ait perçu la valeur du produit peut le faire partir.

- Je suis presque certain d'avoir de la matière sur le sujet.
- Je ne sais simplement plus où elle se trouve ni quels mots j'avais employés.

Ai-je parlé d'effort initial, de friction, d'activation ou d'onboarding ? L'idée a peut-être

- été extraite d'un livre,
- développée dans une note de product management,
- reliée à un autre parcours utilisateur,
- ...

Je pourrais essayer plusieurs recherches par mots-clés et ouvrir les résultats un à un. À la place, je demande au Contexte :

Qu'avons-nous déjà appris sur les efforts demandés avant que l'utilisateur perçoive la valeur du produit ?

Le Contexte interroge une mémoire plus large

La réponse n'est pas dans le Contexte actif. Elle peut en revanche se trouver dans mon LLM Wiki, une bibliothèque commune où sont déjà structurées les milliers de notes tirées de livres, d'articles, de normes et d'autres documents.

Le LLM Wiki et le Contexte n'ont pas la même nature.

Le wiki est une base de connaissances.

- Il conserve des notes structurées (dans mon cas j'ai réutilisé le concept de notes atomiques) à partir de ce que d'autres ont écrit : une définition, un principe, un résultat d'étude, une exigence de norme.
- Chaque note dit ce qui est su, et d'où ça vient.
- Il peut en contenir des dizaines de milliers, accumulées sur des années et sur des sujets sans rapport entre eux.

Aucun Contexte ne tiendra jamais cette quantité, et ce n'est pas son rôle.

Mais le wiki s'arrête là.

- Il ne dit pas pourquoi cette connaissance compte dans le travail que je mène aujourd'hui, ni ce que j'en ai fait.
- Il ne relie pas une idée à une décision, ni une décision à ce qu'elle m'a coûté.

Le LLM Wiki n'est pas un système de pensée : il conserve massivement mais il ne raisonne pas.

Le raisonnement vit dans le Contexte. C'est lui qui garde le pourquoi du comment :

6

- quelle est la mission,
- quelles sont les directives et décisions actées,
- pourquoi cette idée a été retenue pour cette mission,
- contre quoi j'ai tranché,
- ce que j'ai accepté de perdre,
- quelle contradiction reste ouverte.

Il n'y parvient que parce qu'il reste petit — quelques dizaines de notes qu'un humain peut relire.

Chacun paie le prix de sa force.

- **Le wiki gagne l'échelle et perd le fil.**
- **Le Contexte garde le fil et renonce à l'échelle.**

C'est précisément pour ça qu'ils doivent se parler plutôt que fusionner.

Je décris l'idée que je cherche

Je ne copie pas tout le wiki dans le Contexte consacré à l'inscription. Je ne lance pas non plus un RAG, puis une recherche dans un graphe.

Je pose simplement ma question au Contexte :

Qu'avons-nous déjà appris sur les efforts demandés avant que l'utilisateur perçoive la valeur du produit ?

Je la formule avec mes propres mots.

- Je n'ai besoin de connaître ni le titre d'une note, ni son emplacement, ni le vocabulaire employé au moment où elle a été écrite.

Le Contexte transmet cette idée au LLM Wiki par son API. Tout le reste se passe à l'intérieur du wiki.

Le wiki cherche par le sens, puis autour du résultat

Le LLM Wiki commence par retrouver les notes dont le sens se rapproche de ma question.

- Une note peut parler de friction initiale, d'activation ou de délai avant la première valeur sans employer le mot « effort ».

C'est ce que son RAG lui permet de faire : rapprocher mon idée des connaissances déjà structurées sans exiger le même vocabulaire.

Le wiki regarde ensuite autour des meilleurs résultats. Son graphe sémantique suit les relations déjà posées entre les notes et ajoute les voisines qui peuvent éclairer le sujet :

- une définition,
- une limite,
- une contradiction,
- une idée complémentaire.

Une de ces notes voisines n'aurait peut-être jamais répondu directement à ma question. Elle devient pourtant précieuse parce qu'elle montre dans quelles conditions l'idée s'applique, ou à quel endroit elle cesse d'être vraie.

```
graph TD; A[une idée formulée avec mes mots] --> B[le wiki retrouve les notes proches par le sens]; B --> C[il ajoute leur voisinage utile]; C --> D[quelques références sourcées reviennent]; D --> E[le Contexte retient ce qui sert sa mission];
```

une idée formulée avec mes mots

↓

le wiki retrouve les notes proches par le sens

↓

il ajoute leur voisinage utile

↓

quelques références sourcées reviennent

↓

le Contexte retient ce qui sert sa mission

Le RAG retrouve ce que je n'aurais pas su nommer.
Le graphe fait apparaître ce que je n'aurais pas
pensé à chercher.

Pour moi, ces deux opérations n'en forment qu'une. Je demande des références qui répondent à un problème ; le LLM Wiki cherche l'idée et son voisinage, puis me rend une courte liste.

Le Contexte reçoit des références, pas une réponse toute faite

Le wiki renvoie des notes existantes avec leur contenu, leur origine et leurs relations. Il ne fabrique pas une synthèse qui mélangerait silencieusement plusieurs résultats.

Le Contexte peut donc lire chaque proposition, la comparer aux observations sur l'inscription et revenir à sa source lorsqu'une preuve doit être contrôlée.

- Certaines notes concernent bien un effort imposé avant l'accès au produit.
- D'autres parlent d'un effort demandé plus tard, lorsque l'utilisateur en connaît déjà la valeur.

Leur proximité avec ma question ne suffit pas à les rendre équivalentes.

Le Contexte fait le tri à partir de sa mission. Il peut :

- retenir une idée qui éclaire l'abandon pendant l'inscription,
- conserver une contradiction,

- écarter une note trop générale.

Une note retenue entre dans le `Contexte` avec son origine et les sources sur lesquelles elle repose. Le reste demeure dans le wiki.

Je bénéficie ainsi d'une mémoire bien plus vaste sans transformer chaque `Contexte` en copie de toute ma bibliothèque.

- Je n'ai pas besoin d'apprendre comment cette mémoire est indexée pour m'en servir : je décris l'idée que je cherche, puis je vérifie les références qu'elle me rapporte.

Cette chaîne donne accès à la connaissance déjà structurée.

- Elle ne dit pas encore ce que le produit fait réellement aujourd'hui.

Pour répondre à cette autre question, le `Contexte` doit quitter la bibliothèque et interroger directement le code, les tests et le produit en fonctionnement.

À RETENIR — DEUX SYSTÈMES, UNE PORTE

Le LLM Wiki gagne l'échelle et perd le fil ; le Contexte garde le fil et renonce à l'échelle. Je décris une idée avec mes propres mots ; le wiki retrouve les notes proches par le sens et ajoute leur voisinage utile. Le Contexte reçoit quelques références sourcées et ne retient que ce qui sert sa mission.

6.2

Interroger le produit réel

Je suis en train de relire l'étude sur les abandons pendant l'inscription lorsqu'une question arrive :

Un utilisateur invité doit-il créer un mot de passe comme les autres ?

La question paraît minuscule. La réponse est pourtant attendue par une personne du support qui doit répondre à un client.

Je pourrais :

- interrompre ma lecture,
- ouvrir un environnement de test,
- préparer deux comptes,
- rejouer les parcours,
- comparer ce qui se passe.

C'est ce que je faisais auparavant.

- Une demi-heure plus tard, j'aurais probablement la réponse.
- J'aurais aussi perdu le fil de mon travail.

Cette fois, je pose la question au Contexte.

Le Contexte va voir le produit

Le Contexte travaille avec les notes issues de l'étude utilisateur et conserve cette étude comme source de leurs affirmations.

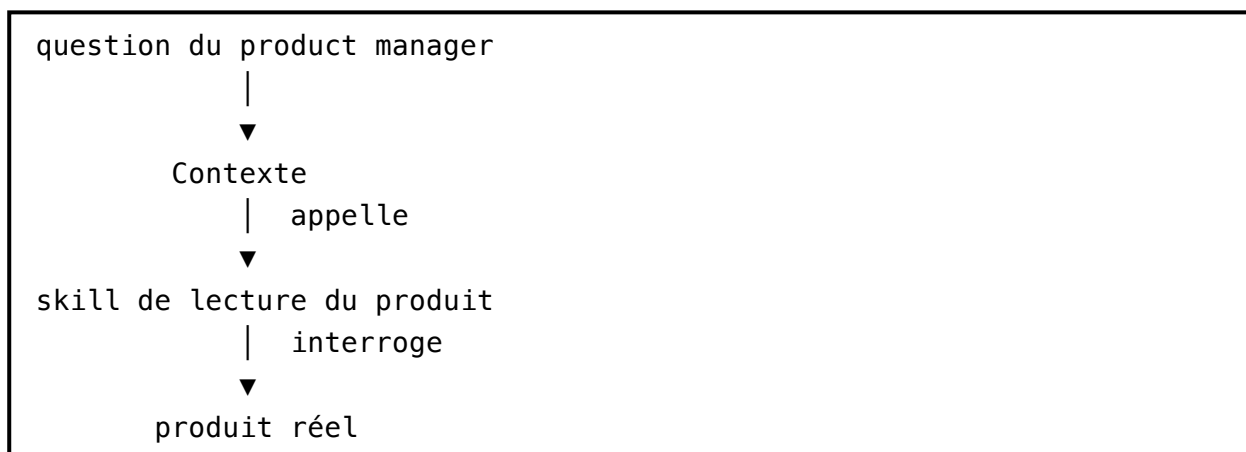
- Il porte aussi la mission en cours.
- Il comprend pourquoi la question compte, mais ces éléments ne lui permettent pas de connaître le comportement actuel du produit.

Je ne maintiens plus de spécification chargée de décrire durablement ce comportement.

- Le produit évolue trop souvent pour qu'un document écrit à côté reste sa copie fidèle.

La seule vérité à jour est ce qui est livré et qui tourne en production : le code source de l'application.

Dans mon système, des skills extérieurs au Contexte savent interroger le code source et produire de la rétro-documentation. Le Contexte appelle celui qui répond aux questions produit et lui transmet ma demande.



Le skill commence par retrouver le parcours d'invitation dans le code. Il suit ensuite ce parcours jusqu'à la règle de création du compte.

6

Le code montre le comportement programmé. Un test confirme qu'un utilisateur invité peut terminer son inscription grâce au lien reçu, sans créer immédiatement de mot de passe. La configuration indique qu'aucune variante propre à un client n'est active pour ce parcours. Le skill me rend une réponse fonctionnelle :

Un utilisateur invité peut accéder à son compte depuis son lien d'invitation sans créer de mot de passe pendant cette étape. S'il choisit ensuite d'en créer un, les règles ordinaires s'appliquent. Les tests confirment le parcours d'invitation et la création volontaire du mot de passe. Le parcours de récupération d'un mot de passe oublié n'a pas été examiné.

Je peux transmettre les deux premières phrases au support. Je peux aussi ouvrir les références fournies par le skill si je veux contrôler son analyse avec un développeur.

Je peux enfin verser cette réponse dans le Contexte, avec ses preuves, la date de la vérification et la limite signalée.

- Répondre au support règle un cas et s'arrête là.
- Capitaliser dans le Contexte fait entrer un fait vérifié sur le produit dans le **monde des idées** de la mission : la prochaine question sur le parcours d'invitation ne repartira pas de zéro, et la note issue de l'étude gagne ce que l'observation seule ne donnait pas — le moment exact où chaque parcours demande le mot de passe.

Une réponse vaut par les preuves qui l'accompagnent

Le skill ne s'est pas contenté de produire une réponse plausible. Pour chaque affirmation, il indique ce qu'il a consulté :

- la règle présente dans le code,
- le test qui vérifie le comportement,
- la configuration qui détermine la variante active.

Cette trace protège contre une réponse assurée mais fausse. Elle permet aussi de voir immédiatement ce qui manque.

Ici, le skill n'a pas vérifié la récupération d'un mot de passe oublié. Il le dit au lieu d'étendre sa conclusion à un parcours qu'il n'a pas lu.

Selon la question, d'autres éléments peuvent devenir nécessaires.

- Le code contient les règles possibles, mais un feature flag indique laquelle est active.
- La structure de la base montre ce que le produit peut enregistrer, tandis que les données autorisées permettent de comprendre la situation d'un compte précis.
- L'historique Git et les Pull Requests aident à retrouver quand une règle a changé et les explications conservées avec ce changement.
- ...

Le skill rassemble seulement ce dont il a besoin pour répondre à la question posée. Il n'aspire pas tout le dépôt et ne transforme pas chaque fichier en source du Contexte.

La réponse reste lisible par le support, les sales ou un product manager. Les preuves restent disponibles pour la personne qui doit la vérifier ou poursuivre l'enquête.

Trouver la règle ne décide pas de la changer

Je reviens à mon étude sur l'inscription. Elle m'avait déjà appris que les deux parcours ne présentent pas la création du mot de passe au même moment.

Le skill ajoute ce que l'étude ne pouvait pas voir.

L'utilisateur invité ne contourne pas la règle du mot de passe : il la rencontre plus tard, et elle est alors exactement la même.

La friction n'a donc pas disparu pour lui, elle a changé de place. Cette précision déplace ma question. Je ne cherche plus à savoir qui subit la règle, mais à quel moment chacun la rencontre : avant d'avoir perçu la valeur du produit, ou après.

Le code m'a appris ce que fait le produit.

- Il ne me dit pas si la règle est bonne, si elle protège réellement les comptes ou si son coût pour les utilisateurs est acceptable.

Pour décider, j'ai encore besoin des autres éléments du

Contexte :

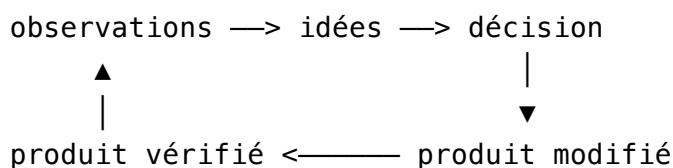
- les observations de l'étude et leur périmètre ;

- les objectifs de sécurité ;
- les raisons qui avaient conduit à retenir cette règle ;
- les conséquences attendues d'un changement.

Le skill établit un fait sur le produit. Le Contexte relie ce fait aux notes et aux décisions déjà présentes, tout en conservant les sources qui permettent d'en contrôler l'origine. **Je reste responsable de l'arbitrage.**

Si je décide de modifier la création du mot de passe, le decision record conserve le choix, son périmètre et les raisons qui le soutiennent. Après l'implémentation, le skill peut revenir au produit et vérifier que le nouveau comportement correspond bien à la décision.

Le mouvement fonctionne donc dans les deux sens. Le produit éclaire la réflexion, puis la décision transforme le produit.



Garder le pourquoi, reconstruire le quoi

L'APPROCHE CODE-CENTRIC

Le code n'est plus seulement le résultat final du développement. Il devient la source à partir de laquelle je peux reconstruire ce qui décrit le comportement actuel du produit. Le Contexte conserve ce que le code ne peut pas expliquer seul : les sources, le raisonnement, les décisions, les alternatives écartées et les raisons du choix.

Cette séparation change la place de la documentation. Je conserve ce qui explique pourquoi le produit a pris une direction : les entretiens, les contraintes, les hypothèses et les arbitrages. Je reconstruis depuis le produit ce qui décrit son comportement actuel :

- une réponse au support,
- une documentation fonctionnelle,
- un journal des changements,
- ...

Je n'entretiens plus manuellement deux versions du même comportement, l'une dans le code et l'autre dans un document qui commence à vieillir dès sa publication (et même bien avant).

Tout ne vient pourtant pas du code.

- Une étude utilisateur ne s’y trouve pas.
- Une obligation réglementaire garde sa propre source.
- Une décision ne se résume pas au comportement qui en résulte.

Le code porte ce qui a été implémenté. Le Contexte garde le pourquoi et permet de le discuter lorsque la situation change.

Revenir six mois plus tard

La réponse donnée au support reste attachée à la version du produit examinée aujourd’hui. Elle ne décrit pas automatiquement les versions suivantes.

- Six mois plus tard, le parcours d’invitation peut avoir changé.
- Un feature flag peut activer une autre variante ou un nouveau mode de connexion peut être apparu.

Le Contexte conserve la question, la réponse fonctionnelle, la date de la vérification, la version examinée, les preuves citées et la limite signalée. Je retrouve ainsi ce que nous savions lorsque nous avons raisonné.

Pour connaître le comportement présent, je demande simplement au skill de rejouer la vérification.

- Il repart du produit réel au lieu de recopier l’ancienne réponse.

Je peux alors distinguer deux questions qui se confondent souvent :

- pourquoi avons-nous pris cette décision à l'époque ?
- que fait le produit aujourd'hui ?

Le Contexte répond à la première en conservant le raisonnement. Le skill répond à la seconde en retournant à la source de vérité.

Le produit ne remplace donc pas le Contexte, et le Contexte ne prétend pas contenir le produit.

- L'un fournit une réalité à vérifier ;
- l'autre garde les connaissances nécessaires pour lui donner du sens et la faire évoluer.

Cette circulation ne s'arrête pas aux outils extérieurs.

Lorsque plusieurs Contextes travaillent sur des aspects différents d'un même produit, leurs connaissances doivent elles aussi pouvoir se retrouver sans être fondues dans un dossier unique.

À RETENIR — L'APPROCHE CODE-CENTRIC

Aucune spécification n'est maintenue comme description durable du produit : ce qui décrit son comportement se reconstruit depuis le code. Une réponse ne vaut que par les preuves qui l'accompagnent et par les limites qu'elle signale. Le code porte le quoi implémenté ; le Contexte garde le pourquoi et permet de le rediscuter.

6.3

Le jour où j'ai eu trois Contextes

Un technicien me raconte qu'il a dû rappeler un client pour une information qui manquait dans un compte rendu d'intervention.

Je note la phrase. Puis je m'arrête, parce que je ne sais pas où la mettre.

- J'ai un Contexte sur la qualité des comptes rendus — celui qui m'a occupé pendant toute la deuxième moitié de ce livre.
- J'en ai un deuxième sur l'intervention elle-même : ce qu'elle contient, les règles qui la gouvernent, ce qui a déjà été tranché à son sujet.
- Et j'en ai un troisième, plus large, ouvert quelques semaines plus tôt : comment amener les opérateurs à prendre en charge les interventions de premier niveau.

La phrase du technicien parle aux trois :

- elle documente un compte rendu défaillant,
- elle dit quelque chose de l'intervention,
- elle éclaire la question des opérateurs.

C'est le moment où le problème change de nature.

- Tant que j'avais un seul Contexte, la question était : est-ce que ça entre, ou est-ce hors sujet ?
- Avec trois, la question devient : dans lequel ?

Les deux façons de se tromper

La première tentation est de tout réunir.

- Un seul grand espace, toute la connaissance au même endroit, plus de question de rangement.

C'est exactement ce que je fuyais. Un espace unique qui contient tout redevient ce que les systèmes documentaires ont toujours été : un lieu où

- l'information est présente et où le tri recommence à chaque consultation,
- les buts se mélangent,
- les sources s'accumulent,
- plus aucune note n'a de propriétaire clair.

J'ai raconté cette impasse au début du livre ; la recréer avec un vocabulaire neuf ne la rendrait pas meilleure.

La seconde tentation est l'inverse : que chaque Contexte reste chez lui. Chacun est propre, borné, cohérent — et aveugle.

Le résultat se voit vite. Mon Contexte sur les comptes rendus a passé trois séances à stabiliser une règle que celui sur l'intervention portait déjà. Ce n'était pas un

doublon visible : c'était du travail refait, sans que personne puisse le remarquer.

Il faut donc une troisième voie, et elle commence par savoir ce que chaque Contexte porte.

Un chiffre devant le nom

Mes Contextes s'appellent 01-saisie-compte-rendu, 02-intervention, 03-autonomie-operateurs, 04-exploitation-catch-all.

Le chiffre n'est pas un rang, c'est une convention de nommage. Il dit ce que le Contexte porte, avant même que je l'ouvre.

- **02- une entité du produit.** Un morceau du logiciel qui tourne en production et qui a ses règles : 02-facturation dans un ERP, 02-intervention dans le mien. Ce qui s'y trouve est établi. Ce Contexte ne se referme pas.
- **01- une hypothèse sur une entité.** 01-saisie-compte-rendu : je pense que la saisie devrait changer, je ne sais pas encore comment. Rien de ce qu'il contient n'est acquis, je crée donc un Contexte 01- pour bien différencier ce qui est stable (en production) d'une hypothèse.
- **03- un problème.** 03-autonomie-operateurs : amener les opérateurs à prendre en charge les interventions de

premier niveau. Ce n'est pas une fonctionnalité, ça traverse plusieurs entités et ça fait naître des 01-.

- **04- le catch-all.** Une phrase entendue, un PDF, une intuition : de la matière dont le sujet n'a encore aucun Contexte. Sans lui, il faudrait en créer un par bribe d'idée — j'ai commencé comme cela et me suis retrouvé avec plus de 50 contextes sans consistance en quelques semaines.

Un 01- finit toujours par sortir du travail actif

Le jour de la mise en production, la nouvelle saisie cesse d'être une hypothèse : elle devient une partie de l'intervention.

Ce qui décrit désormais le produit est proposé à 02-intervention. Le reste demeure dans le Contexte exploratoire, qui est ensuite archivé. Il n'y a plus deux endroits actifs où chercher ce que fait un compte rendu. Une hypothèse qui n'aboutit pas est archivée avec ce qui l'a arrêtée. Ce n'est pas un échec à effacer : c'est ce qui évitera de refaire le même chemin dans deux ans.

Le 04-, lui, se vide dans l'autre sens. Le jour où un problème émerge, je crée le Contexte qui lui correspond, puis je retourne voir ce que le catch-all avait gardé à son sujet. Ce qui ne sert à rien est supprimé.

Tout ça règle une seule question — où je mets ça — en séparant ce que j'instruis de ce qui est acquis.

- Sans cette séparation, chaque hypothèse abandonnée laisse ses traces dans la description du produit.
- Et on n'ose plus rien supposer dans un endroit où tout est censé être vrai.

L'erreur que j'ai mise des mois à voir

J'ai commencé par ouvrir 03-tarifification. Ça me paraissait évident : c'est un gros sujet, il concerne plusieurs entités, il dure.

Six mois plus tard, il contenait :

- la grille de prix,
- deux benchmarks concurrents,
- une discussion sur les remises,
- un ticket de support sur une facture fausse,
- une note sur la TVA belge,
- trois captures d'écran d'un concurrent,
- ...

Rien de tout ça n'était hors sujet : tout parlait bien de tarification.

C'est exactement le problème. Pour refuser une matière, il faut pouvoir dire « ça ne m'aide pas ». Or ce Contexte ne

cherchait rien — **il n'avait pas de question**, seulement une étiquette.

03-autonomie-opérateurs, lui, en a une : comment amener les opérateurs à prendre en charge les interventions de premier niveau.

Un 03- n'est donc pas un gros sujet. C'est une question. La tarification, elle, n'a pas disparu pour autant : elle traverse mes Contextes — 02-facturation en porte les règles, un 01- peut en dépendre — sans en commander aucun.

6

Ma convention n'a donc pas de case pour les domaines, ni pour bien d'autres choses. Elle sert à ranger ce qui a une place évidente, et rien de plus. Le reste attend dans le catch-all, et se retrouve en décrivant ce qu'on cherche plutôt qu'en se rappelant où on l'avait rangé.

J'en ai d'ailleurs deux, appris en me trompant. Un seul mélangeait ce qui venait de mon travail produit et ce qui venait du domaine de l'exploitation industrielle — deux vocabulaires, deux publics, deux façons de nommer les mêmes objets. Il est devenu illisible plus vite qu'il ne se vidait.

- 04-exploitation-catch-all reçoit le fonctionnel de mes clients, l'exploitation ;

- 04-productops-catch-all reçoit ce qui touche à mon métier et à son organisation : product OPS.

Chacun reste lisible, donc chacun se vide.

La phrase du technicien, elle, n'a pas eu besoin du catch-all. Elle est allée dans 02-intervention, parce que c'est là que vit ce qui est vrai du produit indépendamment de la question que je me pose en ce moment.

Reste à savoir comment mes trois Contextes s'en aperçoivent.

À RETENIR — QUATRE PRÉFIXES, UNE SEULE QUESTION

Le chiffre devant le nom dit ce que le Contexte porte : 02- une entité qui tourne, 01- une hypothèse sur cette entité, 03- un problème qui en traverse plusieurs, 04- le sas où l'on dépose ce qu'on ne sait pas encore placer. Un 01- existe pour proposer à son 02- ce qui décrit désormais le produit, puis pour être archivé. S'il n'aboutit pas, il est archivé avec ce qui l'a arrêté. Un domaine n'est pas un Contexte : il en traverse plusieurs sans en commander aucun. Un Contexte sans question ne saura jamais ce qui est hors sujet.

6.4

Lire chez le voisin sans déménager chez lui

6

Je reprends 01-saisie-compte-rendu, le Contexte consacré à la qualité des comptes rendus d'intervention. Pour avancer, j'ai besoin de connaître les règles de l'intervention elle-même : ce qu'un technicien a le droit de saisir, ce qui est obligatoire et ce qui a déjà été tranché.

Ces règles ne vivent pas dans mon Contexte. Elles sont stabilisées dans 02-intervention, le Contexte voisin qui décrit ce qui existe réellement dans le produit.

Je pourrais les recopier pour les avoir sous la main. Six mois plus tard, j'aurais probablement deux versions de la même règle et aucun moyen simple de savoir laquelle fait foi.

Je commence donc par un geste plus léger : mon Contexte déclare qu'il lit 02-intervention.

Lire laisse la connaissance chez son propriétaire

Cette relation est écrite dans le cadre stable de mon Contexte, à côté de sa mission et de ses directives. Elle

ouvre un droit précis : consulter ce que `02-intervention` sait déjà.

La matière reste chez son propriétaire. Elle continue d'y évoluer, et mon exploration s'appuie dessus sans prétendre la posséder.

Je peux ainsi vérifier la règle actuelle sur les champs obligatoires sans la dupliquer dans chaque étude qui concerne un compte rendu. Je peux également revenir à ses sources et aux décisions qui l'ont construite.

Lire n'autorise pas à modifier. Si une note me paraît incomplète ou fausse, mon `Contexte` peut formuler une proposition, mais il ne corrige jamais directement son voisin.

Cette première relation répond à un besoin courant : utiliser une connaissance extérieure tout en la laissant vivre à l'endroit qui en est responsable.

Elle ne répond pas à tous les cas.

Absorber rend la connaissance autonome

Parfois, une simple consultation ne suffit pas.

Je peux avoir besoin de figer la règle exacte sur laquelle repose une décision. Je peux vouloir la reformuler pour le périmètre particulier de mon exploration. Je peux aussi

avoir besoin que cette connaissance reste disponible si le Contexte d'origine est archivé, déplacé ou réorganisé. Dans ce cas, je choisis de l'absorber.

Absorber fait entrer une connaissance dans mon propre Contexte. Je peux reprendre une note telle quelle ou la reformuler dans mon vocabulaire. Selon le besoin, je copie aussi les sources qui permettent de la vérifier ou je conserve des références vers elles.

La copie ne devient pas pour autant une idée sans passé. Elle conserve le Contexte dont elle vient, la date de l'absorption et ce qui a été reformulé.

La différence tient donc moins au geste technique qu'à la responsabilité qui en résulte :

- lorsque je lis, la connaissance continue d'appartenir au voisin ;
- lorsque j'absorbe, une version autonome entre chez moi et je deviens responsable de son usage.

Lire crée une dépendance. Absorber crée une nouvelle responsabilité.

Le mouvement peut repartir dans l'autre sens

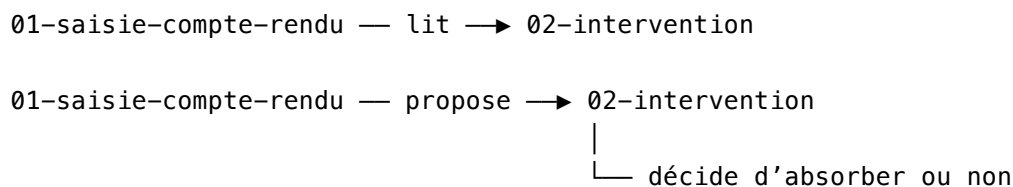
Mon exploration ne fait pas que consommer ce que 02-intervention sait déjà. Elle produit à son tour des

observations, des hypothèses et parfois une connaissance durable.

En étudiant la qualité des comptes rendus, je peux découvrir qu'une règle de saisie appartient en réalité à l'intervention elle-même. Cette connaissance ne devrait pas rester enfermée dans un Contexte exploratoire qui finira peut-être par être archivé.

Je ne l'écris pas directement dans 02-intervention. Je la lui propose.

Le sens de circulation vient de changer :



Celui qui produit une connaissance ne décide pas de sa place définitive. Le Contexte qui la reçoit l'évalue selon sa propre mission.

Il peut :

- reprendre la note telle quelle ;
- la reformuler dans son vocabulaire ;
- conserver une divergence lorsque les deux lectures restent valables ;
- refuser l'absorption lorsque ce qui est utile à l'exploration ne décrit pas l'intervention elle-même.

Le refus n'est pas un échec de la circulation. C'est ce qui empêche un résultat local de devenir silencieusement une vérité générale.

Quand l'exploration devient une réalité produit

Quelques semaines plus tard, la nouvelle saisie des comptes rendus est mise en production. 01-saisie-compte-rendu a rempli son rôle : ce qui n'était qu'une hypothèse est devenu une réalité du produit.

Je ne déplace pas pour autant tout son contenu dans 02-intervention. Je passe en revue ce que l'exploration a produit et je sépare deux ensembles.

Le premier contient ce qui décrit ou explique désormais le produit :

- les notes qui établissent les nouvelles règles de saisie ;
- le vocabulaire stabilisé pour les nommer ;
- les décisions structurantes et les raisons de ces choix ;
- les contradictions qui restent ouvertes après la mise en production ;
- les sources ou les références nécessaires pour vérifier cette connaissance.

Ces éléments sont proposés à 02-intervention. Le Contexte stable ne les accumule pas aveuglément : il enrichit ses notes existantes, reformule ce qui doit l'être et ne crée

une nouvelle note que lorsqu'une idée est réellement nouvelle.

Dans notre exemple, la règle qui limite la clôture à quatre informations rejoint *02-intervention* avec la décision qui l'explique. Elle décrit maintenant une réalité du produit. Le second ensemble raconte le chantier sans décrire le produit actuel :

- les hypothèses abandonnées ;
- les maquettes et les pistes écartées ;
- les brouillons intermédiaires ;
- les tâches propres à l'exploration ;
- les discussions qui permettent de comprendre le chemin parcouru.

Ces éléments restent dans *01-saisie-compte-rendu*. Ils ne sont pas devenus inutiles : ils expliquent ce qui a été essayé, pourquoi certaines directions ont été refusées et ce qu'il ne faudra pas recommencer sans raison nouvelle. Une fois cette sélection terminée, le Contexte exploratoire est archivé dans son ensemble. Il reste lisible et peut être repris plus tard, mais il ne prétend plus décrire le travail actif.

Si l'exploration n'avait pas abouti, elle aurait également été archivée, avec les raisons de son abandon, sans rien imposer à 02-intervention.

La mise en production ne fusionne donc pas deux Contextes. Elle déclenche un passage en revue : ce qui décrit désormais le produit rejoint le Contexte stable ; ce qui raconte l'exploration reste dans le Contexte archivé.

Cette sélection protège les deux côtés :

- l'exploration reste libre de se tromper ;
- le Contexte stable ne reçoit que ce qu'il choisit de reconnaître comme sien.

6

Les relations font partie du cadre

Lire, absorber et proposer ne doivent pas rester des gestes implicites.

Le cadre stable du Contexte indique :

- quels autres Contextes il peut consulter ;
- ce que cette consultation autorise ;
- selon quelles règles une connaissance peut être absorbée ;
- comment son origine et les éventuelles reformulations sont conservées.

Cette gouvernance ne demande pas une infrastructure lourde. Elle donne simplement une adresse aux relations

pour que la provenance, les responsabilités et les frontières restent visibles.

Le réseau ne se construit donc pas en connectant tout à tout. Il se construit relation par relation, chaque fois qu'un travail a réellement besoin d'un voisin.

- Lire chez le voisin évite de dupliquer ce qu'il sait.
- Absorber permet de rendre autonome ce qui doit vivre chez moi.
- Proposer permet enfin au voisin de décider s'il veut intégrer ce que mon exploration a produit.

Cette circulation paraît propre sur le papier. Elle a pourtant un prix : une référence peut vieillir sans prévenir, une copie peut perdre son origine et deux Contextes peuvent conserver des lectures différentes.

C'est ce que le chapitre suivant va examiner.

À RETENIR — TROIS GESTES, TROIS RESPONSABILITÉS

Lire consulte une connaissance qui reste chez son propriétaire ; absorber en crée une version autonome et sourcée dans le Contexte qui la reçoit. Une exploration propose ce qu'elle a appris, mais le propriétaire du Contexte cible décide de ce qui entre chez lui. La mise en production ne fusionne pas les dossiers : le Contexte stable absorbe ce qui décrit désormais le produit, tandis que l'exploration conserve le reste avant d'être archivée.

Ce que la circulation coûte

Six semaines ont passé.

Mon travail sur les comptes rendus d'intervention est prêt à partir en relecture, et quelqu'un relève un écart. La règle sur laquelle je m'appuie — celle qui dit quels champs sont obligatoires à la clôture — n'est plus celle du produit.

Elle a changé. Pendant que je travaillais, une autre exploration a abouti, et le Contexte de l'intervention l'a absorbée.

Personne ne me l'a dit. Je ne l'ai pas lu à nouveau, parce que je l'avais lu six semaines plus tôt et que rien n'a signalé qu'il fallait y retourner.

Le chapitre précédent pose une règle qui a l'air propre : on référence dans le sens de la lecture, on copie dans le sens de l'absorption. Je l'ai tenue pour un arbitrage résolu.

Elle ne supprime pas le risque. Elle le déplace, et il vaut mieux savoir de quel côté il tombe.

Une dépendance qu'on ne voit pas

Ce qui vient d'arriver a un nom : une dépendance aveugle.

Le lien est vivant, donc ce que je lis peut changer sans moi. Ce n'est pas le problème — c'est souvent exactement ce que je veux. Une règle qui évolue chez son propriétaire, c'est le système qui fonctionne.

Le problème est de ne pas savoir qu'elle a bougé.

Le manque est simple à énoncer : rien ne surveille ce que je lis. Mon Contexte sait de qui il dépend ; il ne sait pas quand ce dont il dépend a changé. C'est la moitié du mécanisme, et c'est celle qui coûte.

Une copie m'aurait protégé de cela. Le Contexte qui reçoit devient autonome : il ne dépend plus ni de l'existence ni de l'évolution de sa source.

Mais il perd autre chose en échange. Si rien ne dit d'où vient la copie, la provenance disparaît — et un an plus tard j'ai une note juste dont je ne sais plus sur quoi elle repose. C'est précisément l'état que toute la traçabilité de ce livre cherche à empêcher.

La conclusion n'est donc pas de choisir un camp. C'est d'écrire ce que chaque geste doit porter :

- une **référence** demande de savoir quand sa source change ;

- une **copie** demande de dire d'où elle vient — quel Contexte, à quelle date, et ce qui a été reformulé au passage.

ÉVOLUTION POSSIBLE

Une autre réponse, encore à venir, sera technologique : la capacité des Contextes à notifier leurs observateurs d'un changement de leur état, et plus particulièrement du périmètre observé par l'autre contexte. Le Contexte est un sujet vivant, où de nombreuses possibilités restent à explorer.

6

Les droits que la lecture ne donne pas

Un Contexte qu'on lit beaucoup finit par ressembler à un Contexte qu'on possède un peu. J'ai eu envie, plusieurs fois, de corriger directement une note chez le voisin — elle était incomplète, je savais quoi y mettre, et cela m'aurait pris trente secondes.

La règle ne bouge pas pour autant : ce que j'ai à lui dire est une proposition, et la décision appartient à celui qui reçoit.

Supprimer sans casser la traçabilité

La suppression demande une règle à part, et je l'ai apprise en vidant un bac.

Quand une note change de Contexte, les sources qui l'ont produite restent derrière elle. Elles peuvent se retrouver seules, sans plus aucune note qui s'y rattache.

Le réflexe est de faire le ménage. Il est à moitié bon.

Une source déjà lue et transformée en notes peut être supprimée : ce qu'elle portait vit ailleurs. Une source que personne n'a encore lue, non.

- Elle porte une matière que nul n'a extraite, et la jeter revient à perdre ce qu'on n'a même pas ouvert.

6

Le critère n'est donc ni l'âge du fichier ni la place qu'il occupe.

- C'est son état de lecture, croisé avec ce qui la cite encore.

Une condition va avec, et elle ne se négocie pas. Avant de supprimer, vérifier qu'aucune note, aucune entrée de vocabulaire, aucun brouillon et aucune ligne du journal ne cite plus cette source. Sans cette vérification, la règle casse la traçabilité qu'elle prétend servir.

Une divergence n'est pas une erreur

Reste le cas où deux Contextes ne disent pas la même chose du même sujet.

Le réflexe est d'y voir une incohérence, donc de trancher vite pour rétablir l'ordre. C'est souvent faux.

Une divergence peut être une différence de périmètre : chacun a raison chez lui. Elle peut être une hypothèse encore ouverte que l'un instruit et que l'autre ignore. Elle peut être deux lectures qui tiennent toutes les deux, parce qu'elles regardent le sujet depuis deux endroits. Elle mérite alors d'être conservée comme telle, et c'est l'humain qui en décide. Ce que le système doit garantir, c'est qu'on sache qu'elle existe.

Une divergence oubliée finit toujours par se rappeler au pire moment : quand quelqu'un s'appuie sur l'une des deux versions sans savoir que l'autre existe. C'est ce qui m'est arrivé sur la règle de clôture, à ceci près qu'il ne s'agissait pas de deux lectures mais d'une seule qui avait vieilli.

Le réseau tient donc à un prix, et je préfère le nommer plutôt que de le découvrir en relecture. Relier deux Contextes ne supprime pas le travail de vigilance : cela le rend possible, et cela le déplace sur l'humain. C'est, au fond, ce que ce livre raconte depuis le début.

À RETENIR — LE PRIX DU RÉSEAU

Une référence peut vieillir sans prévenir, une copie peut perdre sa provenance : chaque geste a une dette, et il faut écrire laquelle. Lire n'autorise pas à écrire ; ce qu'on a à dire au voisin est une proposition, jamais une correction. Une source déjà transformée en notes peut disparaître, une source jamais lue jamais — et rien ne se supprime sans vérifier qui la cite encore. Deux Contextes qui se contredisent n'ont pas forcément tort : ce qui est grave, c'est de l'ignorer.

Avant de refermer

Clôture

Clôture

Le livre s'arrête ici, mais le sujet ne s'arrête pas avec lui.

L'épilogue dit où en est ce travail, ce qu'il n'a fait qu'effleurer et où en paraîtra la suite. Les parentés situent le Contexte parmi les approches qui l'ont précédé ou qui l'entourent, pour que tu saches ce qu'il emprunte et ce qu'il ne reprend pas.

7.1 Épilogue

Merci d'être arrivé jusqu'ici.

J'ai essayé de rendre ce livre aussi lisible que possible, et je n'y suis pas complètement parvenu. Il reste des passages plus arides que je l'aurais voulu, des endroits où la mécanique prend le pas sur le récit. Tu as tenu quand même : c'est un cadeau que je ne prends pas à la légère.

7

Ce livre n'a fait qu'effleurer le sujet

Malgré son épaisseur, il n'aborde le Contexte que superficiellement.

Plusieurs parties restent légères sur le plan conceptuel, et je le sais en les publiant. Les notes atomiques, par exemple : selon la manière dont je les implémente, j'obtiens des résultats très inégaux. Ce qui fonctionne dans un Contexte demande des adaptations dès qu'on bascule vers un LLM Wiki, où la matière n'a ni le même volume ni le même usage.

Ces zones méritent d'être renforcées. Elles ne l'ont pas été ici parce qu'il faut bien qu'un livre s'arrête, pas parce que le sujet est clos.

Un travail qui restera un projet à côté

Le Contexte continuera d'avancer, mais toujours comme side project.

C'est une activité menée en marge de mon métier, et je lui ai déjà consacré beaucoup trop de soirées. J'y reviendrai — j'y reviens toujours — mais il est temps de lever le pied et de profiter de la vie.

Ce que je voudrais explorer ensuite

La recherche d'antériorité, je l'ai faite. Elle a remonté des concepts très proches du mien, parfois dans des domaines éloignés du produit et de la gestion de connaissance, souvent dans d'autres langues — le japonais et le chinois notamment, dont les travaux restent largement invisibles depuis le monde francophone et anglophone. J'ai écrit un article sur cette démarche ; il paraîtra sur le blog.

Ce qui manque, c'est la suite : analyser ces travaux.

Les noms sont identifiés, les articles sont là, et je ne les ai pas lus. Ce sont des dizaines d'années de recherche menée par des centaines de personnes sur des questions que j'ai abordées seul, le soir. Les dépouiller renforcerait ce cadre bien plus sûrement que de continuer à l'inventer.

La suite ne sera pas un tome II

Ce livre a servi à cristalliser le concept. C'était son rôle : rassembler ce qui était épars, le rendre transmissible, lui donner une forme assez stable pour qu'on puisse la discuter.

Pour un certain temps, les évolutions seront publiées sur `malorean.net`, et non dans un tome II ou dans une nouvelle version de cet ouvrage. Un article se corrige, se complète, se contredit. Un livre fige — et ce qui est décrit ici n'est pas encore assez mûr pour être figé deux fois. Si tu veux suivre la suite, c'est là qu'elle s'écrit.

7.2

Parentés et voisinages

Contexte n'est pas apparu dans un vide intellectuel.

La plupart des mécanismes qui le composent ont des parents, parfois très anciens, dans la gestion des connaissances, les méthodes de prise de notes, l'ingénierie logicielle, les systèmes d'aide à la décision, les sciences cognitives ou encore la mémoire organisationnelle.

Certaines de ces influences étaient conscientes dès le départ. D'autres n'ont été découvertes qu'après la conception de la méthode, au cours d'une recherche d'antériorité menée une fois le livre presque terminé. Il faut donc distinguer deux choses.

Une **influence** est une méthode ou une idée que je connaissais et dont j'ai consciemment repris ou adapté certains principes.

Une **parenté** ou un **voisinage** désigne un système antérieur qui résout un problème proche, parfois avec des mécanismes étonnamment similaires, sans pour autant avoir directement influencé Contexte.

Cette distinction est importante. Deux idées peuvent être proches sans que l'une dérive de l'autre.

Zettelkasten : la connaissance comme réseau d'unités réutilisables

La parenté avec le Zettelkasten est probablement la plus évidente.

Dans le système de Niklas Luhmann, une note n'est pas simplement rangée dans un dossier correspondant au projet du moment. Elle rejoint un système durable de notes reliées entre elles et peut être réutilisée dans de nouveaux raisonnements.

7

Contexte partage fortement cette idée : une connaissance produite pendant un travail ne devrait pas disparaître avec son livrable.

Les notes atomiques du Contexte appartiennent clairement à cette famille.

La différence principale tient à leur statut. Une note atomique doit conserver sa provenance et s'inscrit dans une architecture qui distingue explicitement sources, contradictions, décisions, état courant, synthèse et livrables.

Le Zettelkasten fournit donc une parenté très forte pour la **construction progressive d'un capital de connaissances**, mais pas pour l'ensemble du système.

PARA : organiser l'information en fonction de l'action

PARA, de Tiago Forte, fait partie des influences conscientes de Contexte.

PARA organise l'information selon quatre catégories : Projects, Areas, Resources et Archives. Une de ses intuitions importantes est qu'une organisation de l'information doit servir l'action plutôt qu'une classification abstraite du monde.

Contexte conserve quelque chose de cette intuition.

Un Contexte n'existe pas parce qu'un sujet existe. Il existe parce qu'un humain poursuit une mission.

La frontière n'est donc pas : « tout ce que je sais sur les bases de données ».

Elle pourrait être : « choisir une architecture de données pour le produit X ».

La différence est que les Contextes eux-mêmes deviennent des unités de travail autonomes, avec leur propre état, leurs sources, leurs connaissances, leurs décisions et leur cycle de vie.

Domain-Driven Design : la frontière avant le contenu

Le terme même de Contexte évoque naturellement le

Bounded Context du Domain-Driven Design.

La parenté tient surtout à une idée : une information ou un concept n'a de sens qu'à l'intérieur d'une certaine frontière.

Le même mot peut avoir une signification différente dans deux contextes différents sans qu'il soit nécessaire de les fusionner.

Contexte transpose cette intuition.

Mais au lieu de faire dépendre la frontière d'un modèle métier cohérent, il la fait dépendre d'une **mission humaine**.

Deux Contextes peuvent donc conserver des compréhensions différentes du même sujet parce qu'ils poursuivent des buts différents.

Git : divergence, autonomie et réintégration

Git est une influence plus directement technique.

Lorsqu'une exploration devient suffisamment importante, il peut être préférable de la faire évoluer dans un Contexte séparé plutôt que de perturber l'espace stable. Cette logique ressemble à une branche.

L'exploration peut diverger, produire ses propres sources et connaissances, puis certaines conclusions peuvent être réintégrées dans l'espace d'origine.

Mais, comme avec Git, intégrer ne signifie pas nécessairement tout fusionner.

Cette intuition a également influencé les relations entre Contextes : autonomie locale, divergence possible et intégration sélective.

Project Memory : conserver le pourquoi du projet

Les travaux sur la **mémoire de projet** sont une parenté que je n'avais pas identifiée lors de la conception de Contexte.

Le problème posé est presque exactement celui qui apparaît lorsqu'un travail dure plusieurs mois : les organisations conservent assez facilement les documents finaux, mais beaucoup moins bien le contexte qui explique comment et pourquoi le projet est arrivé à ce résultat.

Pourquoi cette décision ?

Qu'avons-nous essayé avant ?

Quelles contraintes existaient à ce moment-là ?

Qu'est-ce qu'une nouvelle personne devrait comprendre avant de continuer ?

Contexte répond aux mêmes questions avec plusieurs mémoires distinctes : l'état courant, l'histoire, les décisions, la synthèse et le handoff.

La proximité fonctionnelle est donc très forte.

Experience Factory : le livrable n'est pas le capital

L'Experience Factory, développée autour des travaux de Victor Basili et notamment expérimentée à la NASA, apporte une parenté encore plus importante.

Elle distingue l'organisation qui réalise un projet de celle qui transforme l'expérience du projet en connaissance réutilisable pour les projets suivants.

Cette idée rejoint directement une distinction fondamentale de Contexte :

| *le livrable n'est pas le capital.*

Un projet peut produire un excellent résultat tout en ne laissant presque rien derrière lui.

À l'inverse, son expérience peut être analysée, transformée et conservée de manière à augmenter la capacité à résoudre les problèmes suivants.

Contexte intègre cette capitalisation directement dans l'espace où le travail est effectué.

Knowledge-Centered Service : produire la connaissance en travaillant

Knowledge-Centered Service — KCS — est probablement l'un des voisins les plus intéressants de Contexte.

Son principe est que la documentation ne doit pas constituer une activité séparée réalisée après le travail.

Lorsqu'un problème est résolu, la connaissance existante est recherchée et réutilisée. Si elle est imparfaite, elle est améliorée. Si elle n'existe pas, elle est créée.

Le simple fait de travailler devient ainsi un mécanisme de création et d'entretien de la connaissance.

C'est exactement une des ambitions de Contexte :

*chaque utilisation du système devrait pouvoir
augmenter son capital informationnel.*

KCS est cependant principalement pensé pour des environnements de service et de support, là où Contexte cherche à fournir une structure générique pour de nombreuses missions.

HWMSE : humains, machines et connaissances dans un même espace

La parenté la plus surprenante découverte pendant la recherche vient probablement de Chine.

Le **Hall for Workshop of Meta-Synthetic Engineering**, ou HWMSE, est développé autour des travaux de Qian Xuesen à partir du début des années 1990.

Il est conçu pour traiter des problèmes complexes en combinant explicitement :

- des experts humains ;
- des systèmes machines ;
- des systèmes de connaissances.

Plus intéressant encore, les travaux sur HWMSE ne parlent pas seulement d'utiliser de la connaissance existante. Ils parlent aussi de **création de connaissance** et d'un système de connaissance propre au problème étudié qui s'enrichit continuellement pendant son utilisation.

La proximité avec *Contexte* est donc importante :

- une activité orientée vers un problème ;
- une collaboration humain-machine ;
- des ressources et modèles mobilisés pendant le travail ;
- de la connaissance produite par l'activité ;
- un capital qui s'accumule.

HWMSE ne possède pas pour autant la même grammaire opérationnelle que *Contexte*. On n'y retrouve pas, sous

cette forme, la séparation entre source, note atomique, état, synthèse, contradiction, handoff et livrable.

Mais parmi les systèmes étudiés, c'est probablement l'un des voisins architecturaux les plus proches.

ScholOnto et SWAN : une proposition n'est pas sa source

Une autre parenté forte apparaît dans les systèmes consacrés au discours scientifique.

ScholOnto et, plus tard, SWAN distinguent les documents scientifiques des **propositions produites à partir de ces documents**.

Une ressource peut servir d'origine à une affirmation. Cette affirmation conserve sa provenance.

Elle peut être soutenue, discutée ou contredite par d'autres éléments.

C'est extrêmement proche du principe qui sous-tend les notes atomiques du Contexte :

la source reste une source ; la connaissance construite à partir d'elle est un autre objet.

Ces systèmes montrent également qu'une proposition enregistrée dans un système de connaissances ne doit pas être automatiquement considérée comme vraie.

Elle possède une provenance et un statut dans un raisonnement.

C'est une parenté particulièrement forte avec l'architecture épistémique de *Contexte*.

Systemes fédérés : lire n'est pas copier

Enfin, certaines idées concernant les relations entre Contextes ont des parents étonnamment anciens dans les systèmes de bases de données fédérées.

Ces architectures doivent permettre à plusieurs bases autonomes de coopérer sans renoncer à leur autonomie.

Un système peut exposer certaines informations.

Un autre peut les consulter à distance ou décider d'en conserver une copie locale.

Et dès qu'une copie est créée apparaît le problème classique : que se passe-t-il lorsque l'original change ?

C'est exactement ce que résume dans *Contexte* la formule :

| *référence = dépendance ; copie = responsabilité.*

Les mécanismes informatiques existaient donc depuis longtemps.

Contexte les transpose en règles de gouvernance de la connaissance entre espaces humains-IA.

Une synthèse plutôt qu'une invention ex nihilo

Ces voisinages rendent difficile de présenter Contexte comme l'invention de mécanismes entièrement nouveaux.

- Ce n'est pas ce que montre l'histoire.
- La provenance existait.
- Les notes atomiques existaient.
- La mémoire des projets existait.
- La capitalisation inter-projets existait.
- La création de connaissance pendant le travail existait.
- Les environnements combinant humains, machines et systèmes de connaissances existaient.
- Les assertions sourcées et contestables existaient.
- Les architectures autonomes capables de partager ou copier de l'information existaient.

Ce qui semble davantage caractériser Contexte est la manière dont ces mécanismes sont **sélectionnés, combinés et normalisés dans une même méthode.**

La méthode leur donne une grammaire commune :

- sources ;
- notes atomiques ;
- contradictions ;
- décisions ;
- état ;

- histoire ;
- synthèse ;
- actions ;
- handoff ;
- livrables ;
- relations entre Contextes.

Et surtout, elle applique cet assemblage à un problème très actuel : maintenir une collaboration durable entre un humain et une IA sans transformer chaque conversation en épisode isolé et amnésique.

Contexte possède donc de nombreux parents.

7

Mais, comme souvent, connaître ses parents permet surtout de mieux comprendre ce qui lui est propre.