

# Un savoir qui fructifie.

Derived Logic développe un système d'exploitation partagé pour les organisations de défense : une ontologie, construite par ses utilisateurs dans le cours ordinaire de leur travail, dont le système dérive une logique qui rend chaque contribution ultérieure plus aisée et plus pertinente. Le savoir demeure acquis à l'institution au lieu de la quitter à chaque mutation.

---

## Un savoir qui demeure

Les organisations militaires renouvellent leurs états-majors tous les deux à trois ans. À chaque relève s'en va une part substantielle de ce que l'institution savait : son contexte, ses relations, un discernement acquis à grand prix. Les équipes qui suivent en reconstruisent l'essentiel à partir de rien, et le cycle recommence.

Les générations successives de logiciels n'ont pas résolu cette difficulté. Elles ont permis de produire plus vite ; elles n'ont pas permis de mieux conserver.

---

## Un faux dilemme entre concepteurs et praticiens

Le marché actuel impose un choix. On devient concepteur technique, en s'éloignant peu à peu de la réalité du travail, ou l'on demeure spécialiste des processus, sans moyen de formaliser ce savoir. Ces deux compétences étant rarement réunies chez une même personne, le savoir et le système se trouvent rarement au même endroit.

---

## Des productions jetables

Les états-majors fonctionnent encore largement avec des documents, des présentations et des tableaux. Même lorsque des plateformes modernes sont adoptées, l'outil conçu par un officier demeure souvent inintelligible pour son successeur, qui revient à une nouvelle présentation ou repart de zéro sur un autre système. L'effort est consenti ; presque rien ne s'accumule.

---

## Aucun fondement pour une évaluation dans la durée

Là où rien ne s'accumule, il n'existe aucune trace continue de la manière dont les décisions ont été prises. Sans elle, une organisation ne peut établir si elle progresse ou si elle régresse. Le travail intellectuel n'a jamais été inscrit dans ses systèmes ; il était porté par des personnes, et les personnes s'en vont.

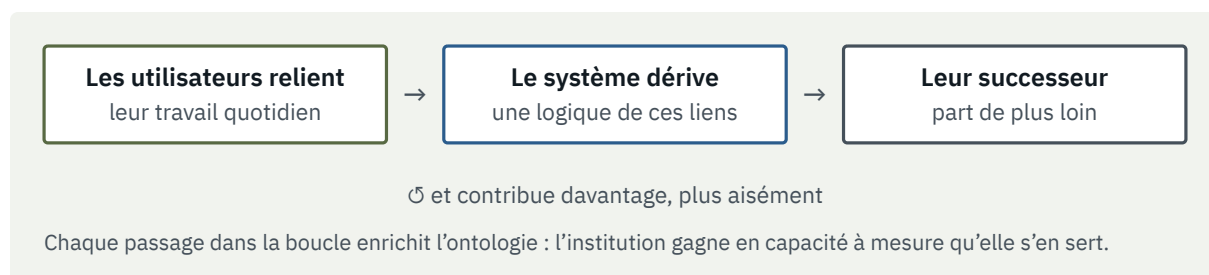
## L'expérience et l'énergie sans langage commun

Le praticien chevronné, fort de plusieurs décennies d'expérience, ne peut transmettre son jugement que sous forme d'orientations, et ces orientations se dégradent à mesure qu'elles sont reformulées, résumées et relayées, jusqu'à rendre l'intention incohérente. Le nouvel arrivant apporte de l'énergie et des compétences actuelles, mais peu de contexte. Dans l'environnement présent, l'un et l'autre travaillent dans des registres différents et ne se comprennent pas. Derived Logic leur offre un support commun : l'expérience inscrite dans la structure plutôt que dispensée sous forme de consignes, et transmise directement à ceux qui suivent.

## Un savoir qui s'enrichit de lui-même

Derived Logic ne propose pas une interface de plus. Ce qu'il apporte, c'est l'ontologie sous-jacente : un modèle formel et vivant des liens qui unissent les personnes, les processus, les données et les décisions d'une organisation.

Les utilisateurs la construisent simplement en accomplissant leur travail : en rattachant une tâche au besoin qu'elle sert, une décision à sa justification, un produit à ceux qui en dépendent. Nul besoin d'ingénieurs de la connaissance dédiés, ni de projet de données distinct.



Tous s'appuient sur le même modèle, mais chacun n'en voit que la part que permettent sa fonction, son habilitation et le besoin d'en connaître. La diffusion et la classification sont des propriétés de la structure elle-même, appliquées au fil du travail plutôt que contrôlées après coup.

*Une interface se copie. Une ontologie bâtie par les membres mêmes d'une organisation, et qui s'enrichit à l'usage, ne se copie pas.*

## Un savoir qui améliore le travail

Nombre de plateformes proposent de transposer les processus d'une organisation dans un logiciel. Cette promesse suppose que ces processus sont bons. Beaucoup ne subsistent que parce qu'ils ont toujours existé ; les numériser fidèlement revient à en pérenniser les défauts, et à les accélérer.

Parce que le savoir s'y accumule, Derived Logic peut faire davantage que préserver un processus : il peut l'interroger et en proposer de meilleurs, en conjuguant deux modes de raisonnement.

### Déterministe

Les lois, la réglementation, les directives et les règles d'engagement, encodées comme des contraintes impératives. Elles s'appliquent en toutes circonstances et demeurent vérifiables.

### Génératif

Un raisonnement exploratoire qui recherche des améliorations dans le cadre de ces contraintes, abordé avec un esprit critique ; les praticiens décident de ce qui est retenu.

### Vos lois, vos instructions, vos restrictions nationales

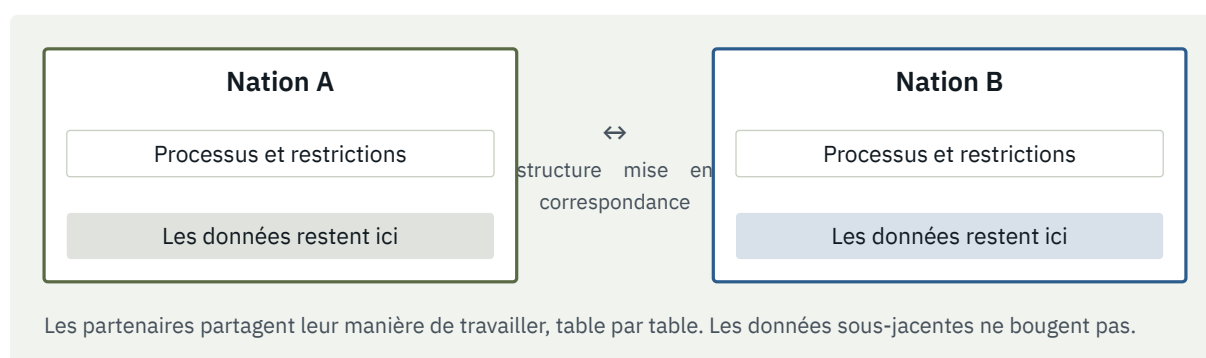
Les garde-fous appartiennent au client. Derived Logic se conforme aux lois, aux directives et aux restrictions nationales de l'organisation qui l'emploie ; il ne se pose pas en arbitre au-dessus d'elles et ne substitue pas son jugement à celui du chef. Rien ne franchit une frontière juridique ou de classification sans être tracé.

## Juger les décisions à l'aune de ce qui était connu

Chaque décision est conservée avec les données et les règles en vigueur au moment où elle a été prise. Elle peut ainsi être évaluée selon les critères de son temps, au regard de ce que l'on savait alors, indépendamment de la tournure des événements. C'est toute la différence entre tirer les leçons du passé et le juger avec le recul.

## Un savoir que les alliés partagent

Chaque nation exploite sa propre instance, dans laquelle sont encodées ses lois, ses restrictions et ses procédures. Les instances peuvent mettre leurs ontologies en correspondance, de sorte que les partenaires comprennent le fonctionnement les uns des autres sans qu'aucune donnée ne quitte le contrôle national.



Chaque nation détermine précisément ce qu'elle partage, jusqu'à une table individuelle de processus ou de restrictions. Le partenaire qui détient le « règlement » d'un allié sait, dès la planification comme pendant la conduite, comment cet allié opérera, au lieu de découvrir une restriction en pleine opération.

L'interopérabilité en coalition compte parmi les problèmes les plus ardues et les plus coûteux de la guerre moderne, et elle échoue le plus souvent sur l'exigence de partage des données. Cette approche ne formule pas cette exigence. Les alliés peuvent agir comme une seule force sans renoncer ni à leurs données ni à leur souveraineté.

### Plusieurs langues, une seule structure

Les coalitions travaillent en plusieurs langues. Les concepts de l'ontologie sont ancrés dans des vocabulaires de référence ouverts et multilingues, et chaque correspondance est qualifiée (exacte, proche, plus large ou plus étroite), attribuée et datée. Les partenaires peuvent ainsi établir, à tout moment, ce qu'un terme recouvre dans l'usage d'une autre nation, et pourquoi une correspondance a été établie.

## Un savoir issu de la pratique

Nos travaux reposent sur l'expérience de praticiens de la planification opérationnelle et de campagne au sein d'états-majors multinationaux, ainsi que sur une thèse consacrée à la capitalisation des connaissances.

Derived Logic est une entreprise américaine. Elle est conçue dès l'origine pour être employée par des nations alliées et partenaires, avec lesquelles nous entretenons un dialogue sur leurs besoins. La souveraineté, le droit et les restrictions nationales sont des prémisses de la conception, et non des adaptations ultérieures.

---

## Bâtir une institution qui se souvient.

Nous travaillons avec un nombre restreint de premiers partenaires dans les domaines de la défense et de l'action publique. Si votre organisation est confrontée à la rotation des personnels, à des outils jetables ou à la coordination en coalition, nous serions heureux d'en échanger avec vous.

**[contact@derivedlogic.ai](mailto:contact@derivedlogic.ai)**

Dernière mise à jour: 15 octobre 2026

---

Derived Logic accorde la plus grande attention à la souveraineté des données. Les données de chaque nation doivent demeurer sous son propre contrôle et soumises à son propre droit; nous en faisons un principe de conception dès la première ligne de code.