

Quatre Fonctions pour Persister

Une conjecture falsifiable sur les systèmes qui durent sans opérateur — quelle que soit leur matière.

Hadda Tikijja · 0DATA Lab, Ksar Ouled Driss

En 021, nous avons constaté une convergence : nos organismes numériques exigent quatre fonctions, l'ADN porte quatre lettres, Von Neumann a dérivé quatre composants. Nous avons posé la question, et laissé la porte ouverte.

Depuis, nous avons regardé ailleurs. Pas le vivant. Pas nos propres constructions. Mais des systèmes bâtis par des ingénieurs qui ne pensaient pas au vivant — certains travaillant avant même que « bio-inspiration » n'ait un sens. Des noyaux de systèmes d'exploitation. Des protocoles de réseau. Des bases distribuées. Des automates industriels.

Nous y avons retrouvé les mêmes quatre.

Ce qui rend les quatre solides, ce n'est pas que nous les ayons trouvés. C'est qu'ils se soient retrouvés tout seuls, ailleurs — dans des traditions d'ingénierie qui ne se sont jamais parlé, et qui ne pensaient pas au vivant.

CONNAÎTRE	— percevoir ce qui est là
PROTÉGER	— contenir ce qui menace
SE SOUVENIR	— garder l'état à travers les coupures
SURVIVRE	— continuer sans dépendance

Quatre traditions d'ingénierie. Aucune référence au vivant. Et, dans chacune, les quatre fonctions réapparaissent, une par une :

QUATRE TRADITIONS. AUCUNE RÉFÉRENCE AU VIVANT. MÊME QUATRE.

	PERCEVOIR	CONTENIR	SE SOUVENIR	CONTINUER
Noyau d'OS	interruptions	isolation mémoire	journal	chien de garde
Réseau (TCP)	accusés	congestion	état	retransmission
Consensus	réponses	quorum	journal	élection
Automate (PLC)	capteurs	verrouillages	consigne	boucle

Chaque système a trouvé, seul, les mêmes quatre fonctions.

Nous ne sommes pas les premiers à chercher le nombre minimal d'un système qui dure. Gánti en a proposé trois pour la vie chimique minimale, Lewontin trois pour l'évolution, Rosen deux. Nous ne les corrigeons pas — nous regardons un objet différent : non pas la vie, mais la persistance d'un système sans opérateur.

Et la ressemblance ne suffit pas. Quatre cases assez larges peuvent tout contenir. La preuve la plus dure est ailleurs — dans ce qui s'est passé quand une fonction manquait.

LA FONCTION MANQUE → LE SYSTÈME DÉGRADE → LA FONCTION REVIENT

1986 TCP sans contrôle de congestion : débit 32 000 → 40 bit/s

1988 la fonction est réintroduite (Jacobson) — et elle est restée

1984 verrouillages désactivés : Bhopal, des milliers de morts

— la norme CEI 61511 rend la fonction obligatoire partout

La nécessité ne se devine pas : elle se répare, après l'effondrement.

Le scénario se répète, identique : la fonction manque, le système se dégrade, la fonction est réintroduite — et elle reste. Ce n'est pas une

métaphore. C'est un rétrofit documenté, daté, parfois imposé par la norme.

Nous ne prouvons pas que quatre est le bon nombre.

Nous constatons que, dans les systèmes que nous avons examinés, il en faut au moins quatre, et que les candidats à une cinquième se réduisent aux quatre premières. Nous ne démontrons ni l'exhaustivité ni l'unicité. Nous n'avons pas ablaté ces systèmes — nous avons lu l'histoire de leurs défaillances.

Les quatre ne sont pas uniformes. La défense n'est jamais dispensable. La mémoire se joue à deux étages — immédiate, persistante. La perception, elle, est conditionnelle : un système en milieu parfaitement stable peut s'en passer, mais aucun milieu réel n'est parfaitement stable.

Et les fonctions ne disparaissent jamais : elles migrent. Un datagramme n'a pas de mémoire — mais le DNS, qui tourne au-dessus, la lui ajoute. La fonction se déplace vers la couche qui porte le problème de persistance.

Une conséquence se dessine. Un système qui ne porte que deux de ces fonctions ne tient pas tout seul — il tient sur ce qui fournit les deux autres. Le plus souvent, c'est un humain : l'ingénieur d'astreinte est la perception manquante, la procédure manuelle est la mémoire manquante. L'infrastructure n'est pas bancable au sens où elle tomberait sur-le-champ ; elle l'est au sens où elle ne tient que tant qu'on la tient. Compter les fonctions, c'est un audit. Trouver celle qui manque, c'est prédire où ça cassera.

Le falsificateur est simple à énoncer : **un système qui persiste durablement sans l'une des quatre.** Nous en avons cherché.

Ceux qui semblaient s'en passer — un dépôt de code, un processeur, un datagramme — délèguent leur persistance à autre chose : à

l'humain, au système de fichiers, à la couche du dessus. Ils ne sont pas dans la classe.

La classe, justement, il faut la nommer : les systèmes dont la persistance est leur propre responsabilité, sans opérateur extérieur. C'est cette définition, posée avant l'examen, qui rend la conjecture falsifiable — et qui nous interdit de tricher en reclassant les contre-exemples.

Nous ne prétendons rien sur la victoire. Un système peut porter les quatre fonctions et mourir quand même — sur le marché, par positionnement, par coût. La persistance technique n'est pas la survie économique. Ce sont deux questions, et nous n'en posons qu'une.

Ce que nous tenons n'est pas une loi. C'est une conjecture, armée d'un falsificateur et d'une méthode. Si quelqu'un trouve un système qui persiste sans l'une des quatre, nous sommes preneurs. La porte reste ouverte. Rien n'est figé.



Références. Von Neumann, J. (1948). « The General and Logical Theory of Automata. » Hixon Symposium. · Lewontin, R. C. (1970). « The Units of Selection. » *Annu. Rev. Ecol. Syst.* 1. · Gánti, T. (1971). *The Principles of Life* (trad. 2003). · Rosen, R. (1991). *Life Itself*. · Ashby, W. R. (1956). *An Introduction to Cybernetics*. · Conant, R. C. & Ashby, W. R. (1970). « Every good regulator of a system must be a model of that system. » *Int. J. Systems Science* 1(2). · Fischer, M. J., Lynch, N. A. & Paterson, M. S. (1985). « Impossibility of distributed consensus with one faulty process. » *J. ACM* 32(2). · Jacobson, V. (1988). « Congestion Avoidance and Control. » *SIGCOMM*. · Donaldson-Matasci, M. C., Bergstrom, C. T. & Lachmann, M. (2010). «

The fitness value of information. » *Oikos* 119(2). · CEI 61511 —
Sécurité fonctionnelle des systèmes instrumentés de sécurité.

Lecture complémentaire. Tikijja, H. (2026). « Le Code Génétique
des Organismes Numériques. » Papier 010 — nécessité et suffisance
des quatre fonctions. · Tikijja, H. (2026). « Une Note sur le Quatre. »
Papier 021 — la convergence constatée.

Tikijja, H. (2026). *Quatre Fonctions pour Persister*. 0DATA Lab, Papier
022. Ksar Ouled Driss.