

N O R M E

Architecture de référence

*Comment un outil de cette suite est construit — la norme dont
chaque architecture d'outil est dérivée.*

v0.4 — brouillon pour correction · état 2026-08-14 · treize principes et dix interdictions,
tous avec identifiant et ancrage déclaré

Kaspar Brönnimann

1 · But, portée, délimitation

Cette norme dit comment un outil de la suite doit être construit. Elle est l'entrée de l'agent d'architecture : d'elle — avec les manifestes, le catalogue de capacités et la norme d'utilisation — est dérivée l'architecture d'un outil concret.

Elle est moissonnée, non inventée. Chaque principe provient soit d'une phrase du manifeste, soit d'une décision qui a déjà été prise et éprouvée dans un produit en cours. Là où un principe n'a pas une telle origine, cela est déclaré.

CE QUE CETTE NORME RÈGLE

La construction d'un outil : couches, objets-noyaux, états, séparation des locataires, preuves.

Où les décisions peuvent tomber et où non.

Quels blocs sont utilisés et comment ils interagissent.

Quelles preuves un outil doit produire.

CE QUE CETTE NORME NE RÈGLE PAS

Ce qu'un outil fait sur le fond. Cela figure dans la norme de méthode du produit.

À quoi ressemble la surface. Cela figure dans l'identité visuelle et l'utilisation.

Ce que les blocs savent faire en détail. Cela figure dans le catalogue de capacités.

Comment on examine. Cela figure dans le concept de test.

2 · Les principes, leurs identifiants et leur ancrage

Treize principes portent l'architecture. Chacun a un identifiant fixe pour que des documents ultérieurs puissent y renvoyer au lieu de le répéter — un constat de conformité écrit « satisfait à RA-09 » et non plus « pas de contact direct avec le modèle ».

L'origine d'un principe se décompose en trois indications qui, jusqu'ici, tenaient dans une seule colonne et faisaient des choses différentes. Dans cette version elles sont tenues séparées ; ce que les trois signifient est dit à la section 2.2.

IDENTIFIANT	PRINCIPE	ANCRAGE	DÉRIVATION	ÉPROUVÉ SUR
RA-01	Le noyau décide, le modèle propose, l'humain assume	PH-03, appuyé sur PH-01	—	Technology Radar (E4)
RA-02	Pas de faits autoritatifs issus du modèle	PH-10, précisé par WA-02	concept d'architecture, frontière de fait	test d'invariants PIA
RA-03	Chaque objet-noyau porte son gouvernance-mixin	PH-02	concept de test	—
RA-04	Brouillon et confirmation sont des champs séparés	PH-03	—	protocole de divergence (E9), arguments et non valeurs (E14)
RA-05	L'approbation est une porte ; un changement l'invalidé	PH-03	concept de test ADR-T04, cycle de vie du savoir	—
RA-06	Les changements sont des transitions, non des ruptures	PH-08 (indirect)	concept de test	évaluations versionnées (E3, E9)
RA-07	Configuration avant programmation	aucun, règle technique	concept de test ADR-T05	coûts d'exploitation et de maintenance par branche
RA-08	La séparation des locataires est structure, non filtre	en suspens — voir 2.2	—	—
RA-09	Pas de contact direct avec le	PH-03, exécuté	cadre, portabilité	—

	modèle	par SC-01, SC-02	de modèle	
RA-10	Chaque résultat porte sa provenance	PH-02, précisé par WA-02	—	—
RA-11	L'incertitude est un champ, non une formulation	PH-04, appuyé sur WA-03	—	degrés de sûreté (E16)
RA-12	Vocabulaire contrôlé	aucun, règle technique	RA-05, RA-06	prolifération de catégories (E7, E12)
RA-13	Un outil produit des preuves, non seulement des résultats	PH-02	concept de test	—

2.1 · La règle pour les identifiants

Un identifiant n'est jamais requalifié.

Les identifiants ne valent quelque chose que si « satisfait à RA-09 » signifie dans cinq ans la même chose qu'aujourd'hui. C'est pourquoi trois règles s'appliquent :

- Les nouveaux principes sont ajoutés en fin et reçoivent le prochain numéro libre. Ils ne sont pas intercalés thématiquement, même si leur contenu appartiendrait ailleurs.
- Si un principe est retiré, son numéro reste occupé et est tenu comme retiré. Il n'est pas réattribué.
- Si le contenu d'un principe change substantiellement, c'est un nouveau principe avec un nouveau numéro — non le même numéro avec un autre contenu. L'ancien renvoie au nouveau.

Les mêmes trois règles s'appliquent aux identifiants de l'espace négatif (NR-01 ..., chapitre 9).

Cette version numérote pour la première fois ; l'ordre est donc encore choisi thématiquement. Dès maintenant il est figé. C'est la même discipline qu'avec les invariants du modèle de qualité et avec les décisions d'architecture du concept de test.

2.2 · Ce que signifie ancrage — et ce qu'il ne signifie pas

Trois indications qui font des choses différentes et n'appartiennent donc pas à une seule colonne :

- **Ancrage** — une phrase du manifeste. Elle fonde pourquoi la règle vaut aussi pour un outil qui n'existe pas encore. Seul le niveau supérieur vaut pour tout ; si un principe repose exclusivement sur un manifeste de domaine, il ne vaut à strictement parler que dans ce domaine.
- **Dérivation** — une autre norme de la suite. Elle montre de quoi la règle découle et porte à l'intérieur de l'ordre, non vers l'extérieur.
- **Éprouvé sur** — une décision qui a été prise et testée dans un produit. C'est une preuve, non un fondement, et peut être vide : une case vide signifie « nulle part encore éprouvé » et est une information, non une gêne.

Une preuve n'est pas un ancrage.

Elle ne répond pas à la question de savoir pourquoi une règle vaut pour un outil qui n'était pas construit à l'époque. Si le fondement d'un principe se tenait uniquement dans cette colonne, il n'était pas une norme de la famille, mais une expérience singulière généralisée — juste peut-être, mais non défendable devant un client.

L'ancrage n'est pas une déduction.

Ce qui est requis est un motif porteur, non une dérivation formelle-logique — la même règle qu'avec les identifiants du manifeste : un identifiant est nommé quand il porte, non quand il convient simplement. La proximité diffère donc ; les ancrages indirects sont marqués comme tels dans le tableau.

Une stratégie d'implémentation n'est pas une dérivation.

La question n'est pas de savoir si une valeur convient à la règle, mais si la règle en découle nécessairement ou n'est qu'un chemin possible vers elle. La configurabilité favorise l'autodétermination, mais l'autodétermination pourrait aussi être produite autrement ; c'est pourquoi RA-07 est tenu comme règle technique et non ancré.

La chaîne n'est pas de un à un. Une phrase du manifeste peut porter plusieurs principes, un principe peut honorer plusieurs valeurs, et une règle technique peut être nécessaire sans posséder de teneur morale. Un corpus de règles où sans exception chaque règle découle d'une valeur est construit après coup et n'est pas propre.

Deux principes ne portent donc expressément aucun ancrage : RA-07 et RA-12. Un ne le porte pas encore : RA-08 — la phrase du manifeste correspondante est formulée mais différée jusqu'à ce qu'il soit réglé qui peut destiner un savoir au partage et ce qu'il advient, en cas de révocation, du savoir qui en est dérivé.

3 · Les treize principes

RA-01 *Le noyau décide, le modèle propose, l'humain assume*

Trois niveaux qui ne coïncident jamais. Le modèle produit des propositions. Le noyau déterministe décide ce qui est valide — états, comptages, résultats d'examen, capacité d'approbation. L'humain décide ce qui devient contraignant.

Ce qui est garanti est imposé de manière déterministe, non demandé au niveau du prompt.

Un engagement qui ne tient que dans le prompt est suivi en étant reformulé. C'est pourquoi le noyau rétrograde une approbation lorsque les conditions manquent, et la langue ne fait que l'expliquer. Les rétrogradations sont unilatérales : un jugement plus strict n'est jamais adouci.

Vérifiable : Il n'existe aucun chemin par lequel un résultat de modèle devient contraignant sans examen par le noyau.

RA-02 *Pas de faits autoritatifs issus du modèle*

Le modèle peut classer, synthétiser, formuler, aiguiller et interpréter. Il ne peut pas produire de références, de chiffres, d'énoncés juridiques ou de preuves. Ceux-ci proviennent de sources vérifiables ou d'outils.

Vérifiable : Toute valeur ayant caractère de preuve porte une source déposée ; à défaut, elle n'est pas restituable. Les comptages dans le résultat viennent du noyau, non du texte.

Depuis le 14 août 2026 ce principe est ancré au niveau supérieur : PH-10 — l'observé et le supposé sont deux choses distinctes ; ce qui apparaît comme fait nomme sa provenance, et provenance signifie d'où quelque chose vient, non qui l'a dit. Auparavant, RA-02 reposait

seulement sur une phrase de manifeste liée à un domaine et ne valait donc, à strictement parler, que là où du nouveau est ramené de l'extérieur.

Remarque : sur cette formulation il existe un besoin de clarification ouvert vis-à-vis du concept de test (chapitre 13, constat 1). Cette norme utilise la formulation plus large, parce que la formulation étroite exclurait l'examen substantiel par un modèle.

RA-03 *Chaque objet-noyau porte son gouvernance-mixin*

Les objets-noyaux sont les choses qu'un outil conduit et assume — un mandat, une entrée de radar, une évaluation, un protocole, un modèle de processus. Chacun porte les mêmes champs.

created_at · updated_at · version · created_by · status

Ainsi chaque objet est datable, versionnable et attribuable à un auteur — humain ou modèle. L'historique naît en sous-produit : qui stocke des évaluations versionnées reçoit gratuitement la question « où l'appréciation s'est-elle déplacée ? ».

Vérifiable : Aucun objet-noyau sans mixin complet ; created_by distingue humain et modèle.

RA-04 *Brouillon et confirmation sont des champs séparés*

Non seulement des états séparés : des champs séparés. Le brouillon du modèle est conservé à côté de la version finale humaine.

Fig. 1 — Structure d'un objet-noyau. Le brouillon ne disparaît pas.

Deux raisons. Premièrement, la différence entre brouillon et version finale est le protocole de divergence — on y apprend sa propre grille d'évaluation et l'on affine l'instruction.

Deuxièmement, un humain examine une motivation, non un nombre ; c'est pourquoi le brouillon contient des arguments et non seulement des valeurs.

Vérifiable : Le champ de confirmation pour l'humain examinateur est obligatoire ; sans lui, l'objet n'est pas contraignant. Le brouillon n'est jamais écrasé.

RA-05 *L'approbation est une porte, et un changement l'invalidé*

Fig. 2 — Un motif qui revient dans chaque produit.

Ce motif apparaît dans la suite à quatre endroits, et il doit partout fonctionner de la même manière : à l'approbation du savoir, à l'approbation d'un mandat, à la porte entre validation interne et test client, à l'extension d'un vocabulaire contrôlé.

Les approbations connaissent trois formes, non deux : approuvé · approuvé sous condition · non approuvable. La forme intermédiaire est pratiquement la plus importante — elle permet d'avancer sans perdre un point ouvert, parce que la condition est documentée avec responsabilité et échéance.

Vérifiable : Un changement à un objet approuvé remet le statut à zéro ; il n'y a aucun moyen de conserver une approbation et de changer le contenu.

RA-06 *Les changements sont des transitions, non des ruptures*

Les autres principes décrivent à quoi ressemble un outil à un instant donné. Celui-ci décrit comment il a le droit de changer dans le temps. Sans lui, les douze autres ont une date de péremption.

Qui change une norme, un vocabulaire ou un schéma change la signification des données déjà stockées.

Sans transitions consignées, l'apprentissage n'est pas possible — ni pour une organisation ni pour un humain. Qui change silencieusement la signification de valeurs stockées n'a plus d'historique, mais seulement un stock qui ressemble à un historique. C'est la manière la plus silencieuse de détruire le savoir : aucun test ne se déclenche, et personne ne le remarque.

Éprouvé au Technology Radar : lorsque la grille d'évaluation y est durcie sans que la transition ne soit consignée, le même échelon signifie quelque chose de différent avant et après le changement. L'historique — la vraie valeur du stock — est alors silencieusement falsifié. Il paraît toujours bon et n'est plus juste.

Quatre règles gardent la ligne du temps propre :

- Chaque objet stocké sait sous quelle version de la norme, du vocabulaire et de la grille d'évaluation il est né. C'est l'extension du triplet de versions des méthodes aux données.
- Chaque changement d'une norme livre une transition avec lui : les données de stock continuent de valoir, ou il y a une mise en correspondance avec la nouvelle version, ou elles sont expressément marquées comme historiques. Une réinterprétation silencieuse n'est aucune des trois possibilités.
- La grille d'évaluation elle-même est un objet versionné, non un savoir dans la tête. Son historique appartient à la preuve.
- Une preuve délivrée est immuable. Une correction est une nouvelle version, non une retouche de l'ancienne.

Une rupture reste permise — tout ne peut pas être tenu compatible dans la durée. Mais elle est expliquée, datée et appliquée aux données de stock. Est seule interdite la rupture inaperçue.

***Vérifiable :** Pour chaque objet stocké, la version sous laquelle il est né peut être déterminée. Aucun changement d'un vocabulaire contrôlé sans règle de transition pour les données de stock. Une preuve délivrée ne peut être modifiée, seulement remplacée.*

Rapport à RA-05 : les deux protègent quelque chose du changement inaperçu — RA-05 l'état d'un objet, RA-06 la signification de ses valeurs. Et RA-06 est la condition pour que RA-12 ne devienne pas un piège : un vocabulaire contrôlé étendu sans transition réinterprète son propre passé.

RA-07 *Configuration avant programmation*

Ce qui diffère par locataire, par produit ou par cas n'est pas programmé mais configuré — comme fichier versionné à côté du code, non dans une base de données et non dans un système tiers.

CE QUI EST CONFIGURÉ	POURQUOI
Méthodes et procédures	Le COMMENT métier change plus vite que le code.
Catalogues et vocabulaires contrôlés	L'extension est un acte conscient, pas un effet de bord.
Règles d'examen et tolérances	Elles sont l'objet de discussion métier, non technique.
Cas de test	Outil partagé, cas propres au produit.
Identité visuelle par locataire	Logo et couleurs sont configuration, non construction sur mesure.
Affectations : fonction à rôle, taux de coût	Ils diffèrent par organisation.

***Vérifiable :** Une demande d'un locataire issue de cette liste n'exige aucun changement de code. Tous les fichiers de configuration sont versionnés avec le code et diffables.*

RA-08 *La séparation des locataires est structure, non filtre*

Fig. 3 — Trois niveaux ; l'héritage se fait vers le bas, jamais en travers.

Une séparation qui repose sur un filtre ne tient qu'aussi longtemps que personne n'oublie le filtre. C'est pourquoi chaque objet lié à un locataire porte structurellement son appartenance, et ce qui est partagé est expressément marqué — non reconnaissable au fait qu'une indication manque.

Trois niveaux : la base partagée sans locataire, le locataire avec ses deltas, l'instance ou le projet avec ses valeurs. L'héritage se fait vers le bas ; un accès en travers des frontières de locataires n'existe pas — ni comme chemin, ni comme habilitation.

***Vérifiable :** Un accès sans référence de locataire ne livre exclusivement que des stocks partagés. Il n'existe pas de requête qui voie deux locataires à la fois.*

Ancrage en suspens. La phrase du manifeste correspondant à ce principe est formulée — le savoir confié reste séparé ; n'est partagé que ce qui est expressément destiné au partage —, mais différée. Tant qu'il n'est pas réglé qui peut destiner quelque chose au partage et ce qu'il advient, lors d'une révocation, du savoir qui en est dérivé, une phrase du manifeste que l'exploitation en cours briserait. Jusque-là, RA-08 est le seul principe sans ancrage, et il est déclaré comme tel.

RA-09 *Pas de contact direct avec le modèle*

Aucun produit ne parle directement avec un fournisseur de modèle. Tous les appels — y compris les plongements — passent par la capability appel IA, qui tient un adaptateur par fournisseur et intercale la pseudonymisation en amont.

Un appel sans pseudonymisation n'est pas techniquement possible, non simplement interdit.

Il en résulte deux propriétés pour le prix d'une : la couche de protection n'est pas contournable, parce qu'il n'y a pas de second chemin. Et un changement de modèle devient une question d'adaptateur au lieu d'un redéveloppement.

***Vérifiable :** Dans le code il n'existe qu'un seul endroit avec contact fournisseur. La preuve d'appel retient fournisseur, modèle et version de modèle.*

RA-10 *Chaque résultat porte sa provenance*

La provenance est un sous-produit de l'exécution, non une prouesse de mémoire du modèle. Qui fait rédiger la provenance à la fin obtient un récit plausible au lieu d'une preuve.

Par section d'un résultat sont tenus : les types de sources utilisés, les énoncés-clés avec leur provenance, les questions de retour posées avec celles qui ont été refusées, les hypothèses faites et ce qui est resté ouvert.

Vérifiable : La provenance peut être reconstituée à partir des données d'exécution sans interroger un modèle. Toute valeur affichée est traçable jusqu'à son origine.

RA-11 *L'incertitude est un champ, non une formulation*

Un outil qui habille son incertitude en mots la rend inexploitable. Le degré de sûreté et l'âge d'un énoncé sont des champs structurés — confirmé, probable, non confirmé — et apparaissent dans l'interface, non seulement dans l'enregistrement.

Vérifiable : Les énoncés à caractère d'appréciation portent un degré de sûreté ; les évaluations portent leur date et sont marquées comme échues quand elles vieillissent.

RA-12 *Vocabulaire contrôlé*

Le classement se fait exclusivement dans des listes existantes. Un modèle range ; il n'étend pas. De nouveaux termes exigent une approbation consciente selon le motif de RA-05 et une transition pour les données de stock selon RA-06.

Sans cette règle apparaissent en quelques mois des dizaines de catégories et l'exploitabilité est perdue — les données ont l'air complètes et ne portent plus rien.

Vérifiable : Une valeur hors de la liste est rejetée, non créée en silence.

RA-13 *Un outil produit des preuves, non seulement des résultats*

À la question « est-ce examiné ? » ne répond aucun mot, mais un document. Cinq preuves surviennent dans cette suite ; lesquelles un outil tient dépend de ce qu'il fait.

PREUVE	CE QU'ELLE CONSIGNE	QUI LA PRODUIT
Provenance	Origine des énoncés d'un résultat.	le noyau, à l'exécution
Protocole d'examen	Résultat d'un examen substantiel d'un résultat.	la méthode qui examine
Protocole de test	Quels types de tests ont tourné avec quel résultat.	le test-runner
Preuve d'appel	Fournisseur, modèle et version par appel de modèle.	la capability appel IA
Protocole de divergence	Différence entre brouillon du modèle et version finale humaine.	le noyau, à la confirmation

Vérifiable : *Chaque preuve porte le governance-mixin et est lisible sans l'outil qui la produit.*

4 · L'architecture en couches

Fig. 4 — Cinq couches et une règle d'appel.

COUCHE	CE QUI VIT ICI	CE QUI NE DOIT PAS ÊTRE ICI
Surface	Présentation et interaction selon la norme d'utilisation.	Règles métier, décisions, appels directs à des capacités.
Méthodes et procédures	Le COMMENT métier : procédures, rubriques, catalogues, vocabulaires — configurés.	Blocs techniques, gestion d'état, comptages.
Noyau	Modèle de domaine, états, approbations, examen des règles, comptages, provenance — et l'exécution des méthodes.	Jugement métier sans norme ; code propre à un fournisseur.
Capabilités	Aptitudes techniques selon contrat de catalogue.	Savoir métier de quelque nature que ce soit. Une capability sait COMMENT, non QUOI est bon.
Persistance	Stockage avec séparation structurelle des locataires.	Logique métier, transitions d'état.

La couche d'exécution appartient au noyau.

Ce qui gère les boucles de retour, les états, les handoffs, les critères d'abandon et la provenance ne fait pas partie d'une méthode, mais l'exécute. Il en découle la règle empirique pour la répartition : les critères vérifiables de manière déterministe appartiennent au noyau, ceux qui sont à apprécier sur le fond à la méthode. Cette couche est tenue dans le catalogue de capacités comme workflow et n'est pas encore construite — c'est l'endroit ouvert le plus important de cette architecture.

5 · Objets-noyaux et états

Un outil conduit un nombre maîtrisable d'objets-noyaux. Ils portent le governance-mixin, traversent le cycle d'approbation de P5 et sont les ancres auxquelles la provenance et les preuves sont accrochées.

Deux motifs ont fait leurs preuves et sont contraignants :

- La chose suivie et ses preuves sont séparées. Une entrée a sa propre identité ; des observations datées viennent s'y amarrer. Ce n'est que de cette manière que le mouvement dans le temps peut être montré.
- L'évaluation est un objet propre, versionné, avec horodatage — non un champ sur la chose évaluée. L'historique naît ainsi gratuitement.

***Vérifiable** : Pour chaque objet-noyau on peut répondre à qui l'a mis dans quel état quand, et sur quoi il s'appuie.*

6 · Surface : ce que la norme d'utilisation exige de l'architecture

La norme d'utilisation tient dans un document propre. Pour l'architecture, il est décisif que plusieurs de ses principes ne puissent être honorés que si le modèle de données les porte. Ils sont donc listés ici comme exigences et non laissés à la surface.

PRINCIPE D'UTILISATION

L'humain décide, le système propose

Aucun énoncé sans source

CE QUE L'ARCHITECTURE DOIT TENIR POUR CELA

Champs de brouillon et de confirmation séparés (RA-04) ; un état qui distingue le brouillon du contraignant.

Provenance par valeur récupérable (RA-10) ; indication de source comme champ obligatoire, non comme convention textuelle.

Rendre l'incertitude visible

Degré de sûreté et date comme champs (RA-11)
; échéance calculable.

Identique à travers la famille

Même vocabulaire de statut dans tous les
produits — donc dans le noyau, non par surface
(RA-03, RA-12).

Confiance et dignité

Identifiants pseudonymes au lieu de noms en
clair dans l'affichage ; divulgation comme pas
conscient selon RA-05.

Réciproquement : les valeurs d'examen concrètes de la norme d'utilisation — tailles
minimales, contrastes, la couleur jamais comme seul signal — appartiennent à l'examen des
règles et non au soin du développeur.

7 · Configuration et extension par les locataires

La suite ne gagne pas son argent en construisant pour chaque client quelque chose d'autre, mais du fait que le même peut être configuré différemment. C'est une exigence d'architecture, non une question commerciale.

NIVEAU	CE QUI EST ICI	QUI L'ENTRETIENT
Base	Méthodes, catalogues, vocabulaires, stocks de référence, règles d'examen.	responsabilité produit
Locataire	Deltas à la base : procédures propres, affectations, taux de coût, identité visuelle, règles d'examen propres.	conseil conjoint avec le locataire
Instance	Valeurs d'un cas concret.	utilisateur

Règles : un delta de locataire écrase la base, il ne la remplace pas. Un delta sans affectation à son homologue dans la base est invalide. Et un delta d'un locataire n'est pas visible pour d'autres locataires — pas même comme modèle.

8 · Exploitation et environnements

Quatre environnements avec une disposition différente : développement avec interfaces mockées, environnement de test interne, environnement d'intégration comme image de la production avec interfaces réelles, et exploitation productive sans exploitation de test.

Pour l'architecture il en découle avant tout ceci : un outil doit pouvoir tourner sans changement de code contre des systèmes environnants mockés ou réels. Le mode d'interface est configuration et est consigné dans la preuve, pour qu'aucune fausse sécurité ne naisse.

La promotion se fait pas à pas et n'est jamais sautée. Chaque produit occupe une plage de ports propre sur l'hôte partagé ; les outils partagent la pile d'exploitation, non leurs données.

***Vérifiable :** Un changement de mode d'interface n'exige pas de nouvelle livraison.*

Aucun produit n'accède aux stocks de données d'un autre.

9. Ce qu'un outil ne doit pas faire

L'espace négatif est la partie la plus vérifiable d'une architecture, parce que les infractions y sont concrètes. Chaque interdiction porte un identifiant afin qu'un constat puisse s'y référer, au lieu de parler généralement de « principes fondamentaux ». Pour ces identifiants s'applique la même règle que pour les principes : ils ne sont jamais requalifiés.

IDENTIFIANT	UN OUTIL NE DOIT PAS ...	PRINCIPE ASSOCIÉ
NR-01	prendre une décision qui revient à un humain — pas non plus par valeur par défaut, automatisme silencieux ou vérité pré-remplie.	RA-01
NR-02	produire une référence, un chiffre ou une preuve qui ne provienne pas d'une source.	RA-02
NR-03	s'adresser directement à un modèle.	RA-09
NR-04	franchir une frontière de locataires — en lecture, en écriture ou comme modèle.	RA-08
NR-05	conserver un état approuvé pendant que le contenu change.	RA-05

NR-06	étendre de sa propre autorité un vocabulaire contrôlé.	RA-12
NR-07	émettre un résultat sans provenance.	RA-10
NR-08	garantir une assurance uniquement par instruction à un modèle.	RA-01
NR-09	changer la signification de valeurs stockées sans consigner la transition.	RA-06
NR-10	retoucher ultérieurement une preuve délivrée.	RA-06, RA-13

Un constat qui invoque l'espace négatif nommé l'identifiant.

Sans identifiant c'est une affirmation. L'invocation de l'espace négatif pèse lourd dans plusieurs procédures de la suite — elle fait d'une réserve un constat impératif —, et ce qui pèse si lourd doit être adressable.

10 · Conformance

Une architecture d'outil est conforme si, pour chacun des treize principes, elle expose comment elle l'honore — ou fonde pourquoi il n'est pas applicable. Le lieu de cette exposition est le constat de conformité issu du cadre.

Parce que les principes portent des identifiants, la liste 1 du constat est un court tableau au lieu d'un essai — par ligne un identifiant, la mise en œuvre et la référence. La liste 2 tient les écarts avec motivation, la liste 3 les endroits où cette norme se tait ; la troisième liste est le retour vers ce document.

konformitaet:

```
- { prinzip: RA-01, erfuellt: ja, fundstelle:
"Noyau/Approbation, section 4.2" }
- { prinzip: RA-06, erfuellt: ja, fundstelle: "Version de schéma
par enregistrement" }
- { prinzip: RA-08, erfuellt: nein, begruendung: "mono-locataire,
délibéré", liste: 2 }
- { prinzip: RA-12, erfuellt: offen, hinweis: "vocabulaire non
encore fermé", liste: 3 }
```

Ainsi le constat est vérifiable par machine : un vérificateur de conformité peut établir si, pour chaque identifiant, une ligne existe — bien avant que quelqu'un n'en lise le contenu.

Même norme ne veut pas dire même comportement — l'approbation vaut par modèle et par surface. Une mise en œuvre n'est approuvée que lorsqu'elle passe les invariants obligatoires et la vérité de référence sur le modèle déployé. La preuve retient modèle et version.

11 · Comment un outil naît de cette norme

1. Consigner but et délimitation de l'outil — ce qu'il fait et ce qu'expressément il ne fait pas.
2. Nommer les objets-noyaux : que conduit l'outil, qu'est-ce qui n'est qu'attribut ?

3. Par objet-noyau, fixer le cycle d'approbation et déterminer les champs de confirmation.
4. Sélectionner les capacités depuis le catalogue ; déclarer ce qui manque comme demande au catalogue.
5. Tracer la frontière de configuration : qu'est-ce qui est base, qu'est-ce qui est delta locataire, qu'est-ce qui est valeur d'instance ?
6. Fixer les preuves : lesquelles des cinq cet outil produit-il ?
7. Fixer la ligne du temps : quelles versions chaque objet-noyau porte-t-il avec lui, et qu'advient-il des données de stock quand vocabulaire ou grille d'évaluation changent ?
8. Traduire les phrases de manifeste applicables ou les noter comme non applicables.
9. Établir le constat de conformité sur les treize identifiants.

12 · Points ouverts de cette norme

P O I N T

É T A T

La couche d'exécution n'est pas construite.

Le noyau exécute des méthodes — jusqu'ici chaque produit le fait à sa manière. Endroit ouvert le plus important.

L'état des lieux des produits en cours manque.

Cette norme décrit comment on construit. Dans quelle mesure les produits existants y correspondent n'est pas encore mesuré.

Le vocabulaire de statut n'est pas unifié.

RA-03 exige les mêmes états dans toute la famille ; les valeurs concrètes ont poussé produit par produit. L'unification est elle-même un cas pour RA-06.

Les données de stock ne connaissent pas encore leur version.

RA-06 exige que chaque objet stocké porte avec lui sa version de norme, de vocabulaire et de grille. Dans les produits en cours ce n'est que partiellement le cas — le rattrapage a lui-même besoin d'une transition.

Les autorisations ne sont pas réglées.

Cette version règle la séparation des locataires, non les rôles et droits à l'intérieur d'un locataire.

Rapport aux décisions d'architecture existantes.

Les décisions existantes sont ici intégrées, mais pas encore formellement ramenées à cette norme.

RA-08 ne porte pas encore d'ancrage.

La phrase du manifeste est formulée et différée jusqu'à ce que consentement au partage et révocation soient réglés. Jusque-là déclaré, non dissimulé.

La provenance n'est pas transitive pour un savoir dérivé.

RA-10 tient une indication par valeur. Si de A et B naît un C et de là un E, après une révocation de A il ne peut être établi que E en dépend. Conceptuellement, cela exige un graphe de provenance. Noté, non construit.

13 · Constats en dérivant

CONSTAT

CONSÉQUENCE

RA-08 n'a pas de phrase de manifeste.

Fait le 14/08/2026 : la phrase est formulée — le savoir confié reste séparé — et différée jusqu'à ce que consentement et révocation soient réglés. RA-08 est jusque-là tenu sans ancrage.

RA-06 n'a pas non plus de phrase de manifeste.

Fait : ancré dans PH-08 (le souvenir comme aptitude persistante), expressément marqué comme indirect. Le fondement a en même temps été formulé de manière neutre vis-à-vis des produits.

La frontière de fait est formulée deux fois différemment.

Cette norme utilise la formulation plus large. PH-10 ancre désormais le principe au niveau supérieur ; la contradiction avec le concept de test en reste indépendante, et RA-02 reste à cet égard sous réserve.

Trois notions de preuve n'étaient pas délimitées.

Le chapitre 3.12 les délimite. Elles restent séparées mais portent le même gouvernance-mixin.

La norme d'utilisation place des exigences sur le modèle de données.

Le chapitre 6 les rend explicites. Sans cette traduction, plusieurs principes d'utilisation ne seraient pas honorables, mais seulement affirmés.

La question de l'ancrage n'avait été posée que deux fois.

Lors de l'examen de tous les treize identifiants il est apparu : cinq ancrés, deux seulement liés à un domaine, six sans réponse. Après les décisions du 14/08/2026 ce sont dix ancrés, deux expressément techniques, un en suspens.

Quatre des cinq manifestes sont des manifestes produit.

Seul le niveau supérieur vaut pour tout. Tant que les manifestes de domaine sont taillés selon les outils, chaque principe qui s'y appuie hérite de leur frontière de domaine. Les formulations neutres vis-à-vis des produits gisent déjà dans le registre des identifiants, dans les documents

sources pas encore.