Vers la *Software-Defined Assurance*

Helsing white paper

17 décembre 2024

Sommaire

Synthèse

L’assurance logicielle joue un rôle clé dans l’introduction de nouvelles capacités de défense au rythme du champ de bataille. Avec la montée en puissance du logiciel et de l’intelligence artificielle au cœur des chaînes d’engagement, notre approche en matière d’assurance doit évoluer selon un nouveau modèle :

- L’assurance devient un processus continu favorisant l’adaptabilité des capacités.
- La réutilisation de composants logiciels sur plusieurs plateformes matérielles devient la norme.
- Les systèmes sont conçus pour être assurés dans le cadre de *composable forces* en perpétuelle transformation.
- Des approches et des techniques spécifiques sont nécessaires pour garantir la fiabilité des systèmes à base d’intelligence artificielle.

Nous qualifions ce modèle de ***software-defined assurance*** et préconisons des investissements stratégiques dans les infrastructures logicielles, les actifs numériques et de nouvelles normes pour en accélérer l’adoption. Une mise en œuvre réussie de ce modèle offrirait un avantage stratégique majeur dans la conception et l'évolution des capacités de défense, au rythme du champ de bataille.

01 À mesure que la défense devient *Software-Defined*, notre approche de l’assurance logicielle doit évoluer

Comme la guerre en Ukraine l’a montré, le logiciel et l’intelligence artificielle prennent une place centrale dans les capacités opérationnelles et systèmes d’armes modernes. La “masse” est désormais générée grâce à des solutions *software defined*, en intégrant intelligence et autonomie à du matériel sur étagère, à faible coût et au rythme du champ de bataille. La supériorité décisionnelle repose sur une compréhension tactique en quasi temps réel, rendue possible par la collecte, la fusion et l’analyse des données issues des capteurs. Les mises à jour logicielles permettent aux forces de s’adapter rapidement à des environnements opérationnels en constante évolution et à des menaces émergentes.

La couche logicielle s’affirme comme le moteur principal de capacités militaires performantes et évolutives, véritable ciment des chaînes d’engagement - dépassant largement le statut de simple composant.

Si le terme *software assurance*, en anglais, est parfaitement consacré, le terme “d’assurance logicielle”, en français, l’est beaucoup moins. Il est souvent réduit à la notion “d’assurance de conception logicielle”, ou Design Assurance, base de la DO-178C.

Dans ce document, le terme "assurance” désignera l’ensemble des démarches visant à garantir la confiance dans l’aptitude d’un système à son déploiement opérationnel. Elle englobe des processus tels que la qualification et la certification, reposant sur des activités spécifiques comme l’assurance de conception ; la validation et la vérification (V&V) ; l’assurance qualité ; la sécurité et la sûreté de fonctionnement.

Pour garantir que ces capacités soient efficaces et sûres, il est impératif de les “assurer”. L’assurance consiste à établir la confiance dans un système en démontrant qu’il répond aux exigences de performance et de sécurité. Sans processus d’assurance efficace, les forces ne peuvent pas compter sur ces systèmes en conditions opérationnelles. Les processus d’assurance actuels, élaborés au fil des décennies, s’appuient sur l’expérience accumulée dans la conception et la production des systèmes. Ils sont encadrés par des normes strictes, dont le respect est exigé par les acheteurs et garanti par les fournisseurs.

Cependant, à mesure que la prépondérance du logiciel dans nos capacités militaires progresse, nos processus d’assurance doivent aussi évoluer. Les approches d’assurance actuelles ne sont adaptées ni au tempo des opérations, ni aux possibilités du logiciel. Cela bride l’innovation, le développement et l’évolution des capacités militaires, et dilue les investissements. Des processus d’assurance mieux adaptés sont nécessaires pour concevoir des systèmes plus performants, plus rapidement.

Bien que l’approche générale d’ingénierie système fondée sur la preuve reste valide, les besoins de *l'économie de guerre* - essentiellement tempo et masse - nous enjoignent d'exploiter les avantages du logiciel. En particulier, notre approche de l’assurance doit permettre :

01 **La rapidité :** Le développement logiciel suit des cycles nettement plus courts, avec des itérations fréquentes couvrant l’ensemble du processus, des exigences initiales à la validation finale. Les mises à jour strictement logicielles, sans nécessiter de modifications matérielles, peuvent avoir des impacts déterminants sur le champ de bataille. Le processus d’assurance doit donc permettre de livrer ces améliorations rapidement, tout en garantissant le bon niveau de confiance.

02 **La transférabilité :** Les composants logiciels peuvent généralement être exploités sur différents équipements. Une même capacité *software-defined* peut être déployée sur de nombreux types (et générations) de matériels, même conçus par différents industriels, maximisant ainsi le retour sur investissement. Le processus d’assurance doit reconnaître cette relation 1:N entre logiciel et matériel.

03 **L’autonomie :** Le niveau d’autonomie des systèmes augmente dans tous les domaines, porté notamment par les avancées en intelligence artificielle. L’IA pose un nouveau défi pour l’assurance, nécessitant des approches techniques différentes et plus robustes de l’assurance logicielle “traditionnelle”.

04 **La “composabilité” :** Historiquement, nous avons compté sur des opérateurs humains pour connecter les systèmes entre eux, avec des interfaces numériques limitées. À mesure que les systèmes se connectent, la notion de chaînes d’engagements dynamiques, créées dans l’immédiateté du champ de bataille, devient réalité. Au delà d’une approche “système de systèmes” déjà dépassée, chaque objet est en fait un élément dont l’action peut être combinée dans une infinité de configurations au sein d’une *composable force*. C’est le coeur de la *Mosaïc Warfare* américaine, qui justifie à elle seule une petite révolution de l’assurance logicielle.

Ces défis appellent une refonte des processus d’assurance adaptés à un champ de bataille *software-defined*. Nous l'appelons la **software-defined assurance**. Elle ouvre la voie à un développement capacitaire plus performant, moins coûteux et plus rapide.

02 Comment définir la *Software-Defined Assurance*?

Le modèle de *software-defined assurance* repose sur quatre principes fondamentaux qui le distinguent des approches d'assurance traditionnelles dans le domaine de la défense.

Nous dépeignons ici une situation de manière exagérément binaire. Dans certains cas, des initiatives locales ont déjà conduit à des approches qui ressemblent davantage à la *software-defined assurance*. Nous ne rejetons pas ces percées, bien au contraire, nous les soutenons. Nous souhaitons qu'elles soient amplifiées et généralisées, dans le cadre d'une transition globale du domaine de la défense vers un modèle de *software-defined assurance*. En identifiant les besoins fondamentaux qui sous-tendent ces évolutions et en décrivant les tendances émergentes, nous espérons attirer l'attention sur cette opportunité et accélérer cette transition.

Approche traditionnelle	***Software-Defined Assurance***
Assure un système de manière ponctuelle pour approuver son déploiement ou sa rénovation à mi-vie	Considère l'assurance comme un processus continu, incluant les opérations et les mises à jour régulières
Définit et évalue un système combiné matériel + logiciel	Définit et évalue un système logiciel destiné à être utilisé sur de nombreuses plateformes (relation 1:N)
Assure un système de manière isolée	Assure un système comme une partie intégrante d'une *composable force*
Part du principe que tous les logiciels sont conçus à base de code	Définit des voies d'assurance spécifiques pour l'intelligence artificielle

Vers un modèle d’assurance continue

Les processus d’assurance actuels ont pour objectif d'évaluer l’aptitude d’un système à être déployé, une fois pour toutes (ou au moins pour les 10 à 15 ans précédant sa rénovation à mi-vie). Cela ne tient pas compte des opportunités offertes par la nature plus dynamique du logiciel, qui peut être mis à jour à un rythme arbitraire.

Pour le logiciel, l'état de l’art de l'assurance est mieux décrit par une boucle continue, avec un nombre arbitraire de redéploiements, plutôt que comme un processus linéaire avec un début et une fin. Ce changement de perspective fondamental ouvre deux opportunités immédiates :

En premier lieu, l’assurance dépasse la mise en service pour se prolonger tout au long de la vie opérationnelle. Le logiciel peut enregistrer et rendre compte d'événements opérationnels, alimentant ainsi le processus de mise à jour. En plus d’améliorer les capacités opérationnelles, cela permet d’identifier où l’assurance a été insuffisante ou dans quelles conditions non prévues les capacités ont été sollicitées, initiant une boucle d’assurance complémentaire.

Ensuite, chaque itération de la boucle peut être adaptée au contexte opérationnel *ad hoc*. Cela permet de déployer une primo-capacité plus rapidement, puis de couvrir d’autres contextes opérationnels, cas d’usages ou modes d’engagement au fur et à mesure qu’ils deviennent pertinents. Cette approche diffère radicalement des processus d’assurance traditionnels, qui visent à couvrir une grande variété de scénarios opérationnels lors d’une évaluation unique, excluant *de facto* les autres. Cette approche peut être vue comme une extension de la maintenance évolutive, dans laquelle c’est le périmètre et la définition même de la capacité opérationnelle de l’objet qui évolue.

1:N - Assurer un logiciel destiné à plusieurs plateformes

Les processus d’assurance actuels considèrent l’évaluation d’un système global composé à la fois de matériel et de logiciel, liant ainsi le développement et l’assurance du logiciel à ceux du matériel. Cette approche repose sur l’idée que c’est le système dans son ensemble qui est livré et qui doit être conçu, développé et évalué de manière intégrée. Dans ce cadre, le logiciel est perçu comme un simple composant du système majoritairement matériel. Cette logique entraîne une approche en cascade (*waterfall*) du développement et de l’assurance, avec des méthodes et des délais alignés sur ceux du matériel. En plus de brider la réutilisation des composants logiciels, cette approche est une des causes racines du problème de rétention des talents du logiciel dans l’industrie de défense, qui les éloigne de l'état de l’art du développement logiciel et nuit alors à leur employabilité.

Nous devons libérer les processus de développement et d’assurance logicielle des contraintes imposées par les approches et les délais liés au matériel. Chaque système devrait être conçu comme étant constitué de deux parties distinctes et égales : le matériel et le logiciel. En définissant précisément l’interface et les exigences entre ces deux composantes, nous pourrons itérer indépendamment sur le logiciel en adoptant une approche dédiée. Lorsque nécessaire, il sera possible de tester le logiciel avec le matériel pour évaluer l’ensemble du système. Le Next-Generation Acquisition Model de l’US Air Force illustre clairement cette orientation.

Les performances et la conformité du logiciel, notamment avec les spécifications de l'interface matérielle, peuvent être testées de manière largement automatisée, à grande échelle, en s'appuyant sur un mélange de données réelles, enrichies et simulées. Cela permet une évaluation robuste des performances et favorise des améliorations fondées sur les données. L'échelle et la fréquence de ces tests sont uniquement limitées par les coûts de calcul, et l'automatisation permet une réévaluation rapide du système logiciel.

L'assurance logicielle permet aussi de valider de manière rapide et automatisée des upgrades matérielles, et donc de favoriser l'insertion technologique incrémentale et de démultiplier le retour sur investissement logiciel. C'est là tout le potentiel de la défense définie par logiciel, soutenue par la *software-defined assurance*.

Assurer les *composable forces*

Les approches traditionnelles d'assurance se limitent à vérifier si un système singulier répond aux exigences qui lui ont été données, en se focalisant sur le système isolé. Si certaines de ces exigences concernent les interactions avec le contexte opérationnel environnant, en spécifiant des interfaces à respecter (par ex : L16), en comparaison, peu d'efforts sont consentis sur l'évaluation de la manière dont les systèmes fonctionnent ensemble.

Même en tentant de capturer les interactions spécifiques entre les systèmes dans les exigences, il serait impossible de couvrir de manière exhaustive ou dynamique la diversité des interactions, en particulier avec l'augmentation du niveau d'autonomie de chaque système. En effet, le problème de spécification croît de manière exponentielle avec le nombre de capacités impliquées, et une définition exhaustive des exigences et des tests devient alors irréalisable.

Avec la *software-defined assurance*, il est possible de tirer parti des environnements de test riches et des logiciels pour modéliser le comportement d'un système en développement lorsqu'il interagit avec d'autres systèmes. Cette approche doit être axée sur les données. Il s'agit de créer des environnements d'évaluation où les systèmes sont confrontés à une diversité de situations opérationnelles pour analyser leurs comportements. Cela permet de dépasser les limites des approches "manuelles" de spécifications, en s'appuyant sur ce que les données et les capacités de calcul peuvent offrir.

Assurer l’intelligence artificielle (IA)

Les approches actuelles d’assurance des logiciels sont issues de celles développées pour les systèmes électroniques, les logiciels étant initialement perçus comme une extension de ces derniers. Au début, l’assurance des logiciels suscitait des doutes : les méthodes courantes, souvent statistiques (par ex : AMDEC), ne permettaient pas d’analyser efficacement les erreurs de conception logicielle. Dans le domaine de l’aéronautique, cela a conduit à la création de la norme DO-178, ancêtre de la DO-178C / ED-12C utilisée aujourd’hui, avec un accent novateur sur l’assurance de la conception et le développement de techniques de vérification spécifiques au logiciel.

Aujourd’hui, l’intelligence artificielle (IA) se trouve dans une situation similaire. Bon nombre des méthodes bien établies pour assurer la fiabilité des logiciels sont difficiles, voire impossibles, à appliquer aux logiciels basés IA, où les modèles sont construits à partir de données et non programmés manuellement par des développeurs. Cette situation crée des frictions dans la conception et le développement des logiciels d’IA, car les critères d’assurance manquent de clarté. Le risque maximal serait que les systèmes impliquant l’IA soient soumis à un cumul d’exigences inadaptées et de nouvelles exigences encore mal définies.

La *software-defined assurance* reconnaît que les méthodes de création logicielle évoluent, et que cela influence la manière dont ils doivent être assurés. Elle distingue clairement les concepts indépendants de la méthode de mise en œuvre (comme les exigences de haut niveau) de ceux qui y sont liés (comme la couverture du code). Comme les approches utilisées pour créer de l’IA évolueront sans aucun doute dans les années à venir, cette structure doit permettre des mises à jour régulières et l’intégration de nouvelles techniques.

03 Lignes d'effort pour une software-defined assurance

Pour réussir la transition vers une *software-defined assurance*, des actions concrètes et des investissements ciblés sont nécessaires dans plusieurs domaines clés :

01 **L'infrastructure :** Une infrastructure logicielle capable de soutenir le développement, l'assurance, le déploiement, la réassurance rapide, les mises à jour et le suivi à grande échelle sur des milliers de systèmes.

02 **Les actifs numériques :** Des actifs numériques détenus par les clients — données, environnements, modèles — développés de manière continue et réutilisables pour l'assurance dans différents programmes.

03 **Les normes :** Une refonte des normes de défense pour le développement des logiciels, afin de permettre une assurance qui prenne en compte les spécificités des logiciels.

Une stratégie claire doit guider ces efforts, avec des investissements adaptés pour initier le processus de changement. La pleine exploitation, à l'échelle, de la *software-defined assurance* nécessitera aussi de repenser les processus d'acquisition.

L'infrastructure

L'atout principal de l'assurance logicielle réside dans le fait que le processus peut lui-même être intégré dans un logiciel. Historiquement, les processus d'assurance sont décrits dans des normes, appliqués à travers des procédures, et vérifiés par des audits et des contrôles. Ces approches sont souvent chronophages, sujettes aux erreurs et exigent une gestion complexe des compétences et des connaissances.

Pour l'assurance logicielle, y compris pour l'intelligence artificielle, ces processus peuvent être intégrés directement dans l'infrastructure utilisée tout au long du cycle de vie : développement, assurance, déploiement, mises à jour et surveillance. Cette infrastructure pourrait servir de système centralisé de gestion des capacités pour :

- Intégrer les exigences d'assurance dans les processus de développement (par exemple, la traçabilité des données pour l'IA).
- Gérer et intégrer les actifs numériques nécessaires au développement et à l'assurance (voir la section suivante).
- Fournir des outils pour définir et, dans la mesure du possible, automatiser le processus d'assurance.
- Vérifier que les exigences d'assurance sont respectées avant le déploiement des capacités logicielles.

Cette infrastructure n'a pas besoin d'être monolithique, mais elle doit être suffisamment intégrée pour garantir sa cohérence. Les organisations — qu'il s'agisse des États ou de l'industrie — auront des préférences pour certains outils spécifiques, mais ces derniers devront être réunis sous un système global qui gère l'ensemble du processus d'assurance logicielle. Ce système central devra orchestrer le processus tout en respectant les exigences strictes de sécurité et de classification, essentielles à la défense. Les ministères de la Défense auront un rôle clé à jouer dans le déploiement et l'adoption de cette infrastructure à l'échelle de l'écosystème.

Les actifs numériques

Les capacités d'assurance des logiciels et de l'intelligence artificielle se construisent progressivement, grâce à des investissements continus dans les infrastructures et les actifs. Les principales catégories d'actifs nécessaires sont :

- **Simulations :** Environnements de simulation avec des modèles précis. Les simulateurs permettent de modéliser et de tester les performances des systèmes dans une large gamme de scénarios, y compris des événements rares ou extrêmes difficiles à reproduire dans le monde réel. Cela accélère considérablement le processus de réassurance après les mises à jour.
- **Domaines opérationnels de conception (ODD) :** Définitions normalisées du contexte opérationnel attendu, qui établissent les limites dans lesquelles évaluer les performances d'un système. Une bonne normalisation des ODD favorise la réutilisation des composants logiciels et l'évaluation au niveau des systèmes intégrés.
- **Données :** Collections de données réelles, augmentées et artificielles. Une évaluation rigoureuse des systèmes logiciels et de l'IA nécessite des données suffisamment représentatives et couvrant les ODD.

Ces actifs doivent être détenus et contrôlés par l'État, afin de garantir que l'ensemble des partenaires industriels puissent développer des systèmes assurés. Ils peuvent parfaitement être développés et maintenus par des partenaires industriels, comme d'autres types d'équipements, mais leur administration ne doit pas leur en être déléguée.

Ils doivent s'inscrire dans une feuille de route à long terme, pilotée par l'État. Bien que des programmes spécifiques puissent en accélérer le développement, il est crucial d'avoir une vision cohérente des actifs qui pourront être utilisés à travers différents programmes. Des exemples émergent déjà, comme l'investissement de l'US Air Force dans les jumeaux numériques, mais davantage de rapidité et d'échelle sont nécessaires.

Les normes

Les normes actuelles pour l'assurance logicielle en défense proviennent essentiellement de deux sources :

01 Les normes de défense pour le matériel adaptées au logiciel (par ex : AQAP 2310 + 2210).

02 Les normes civiles pour l'assurance logicielle adaptées à la défense (par ex : DO-178C).

Dans le cas des *software-defined capabilities*, il est nécessaire de développer des approches standardisées qui reconnaissent deux principes fondamentaux : (a) le logiciel diffère du matériel et (b) les exigences militaires diffèrent de celles du monde civil. Les nouvelles normes doivent :

- Fournir des orientations pour le développement et la vérification de l'IA et des modèles d'apprentissage automatique.
- Trouver un équilibre entre l'innovation rapide et les risques liés à l'assurance itérative.
- Répondre aux besoins de réassurance rapide pour les logiciels mis à jour de manière incrémentale (par exemple, la mise à jour des modèles IA avec des données de mission).

Des initiatives comme l'**EICACS** (European Initiative for Collaborative Air Combat Standardisation) doivent intégrer ces évolutions sans être trop dépendantes des approches civiles ou orientées matériel. Des efforts similaires sont nécessaires dans tous les domaines — aérien, maritime, terrestre — pour garantir que les normes reflètent les exigences opérationnelles militaires.

Les États ont un rôle essentiel à jouer en montrant leur volonté d'innover, non seulement en matière de capacités, mais aussi en matière d'assurance. Les décideurs doivent recentrer leur attention sur les résultats souhaités — des systèmes fiables — plutôt que de se reposer aveuglément sur des normes ou des processus existants. Cela nécessitera des efforts pour trouver des solutions mutuellement acceptables, mais un ou deux programmes significatifs, dotés d'un mandat pour promouvoir l'innovation à la fois dans l'assurance et dans les capacités, pourraient agir comme des pionniers et servir de point de bascule pour l'assurance et le déploiement des capacités basées sur l'intelligence artificielle.

Des modèles d’acquisition transformés sur la base d’architectures ouvertes

Il est important de reconnaître que la modernisation des processus d’assurance, à elle seule, ne suffira pas. Les processus d’acquisition doivent également évoluer. Le *Next-Generation Acquisition Model* de l’US Air Force met en lumière certains des principaux sujets à aborder. Sans changements dans la manière dont les opérations d’armement sont conduites, les bénéfices apportés par une *software-defined assurance* risquent d’être limités par des formes diverses de dépendances aux fournisseurs, qu’elles soient techniques ou commerciales. Cette question dépasse largement le cadre de ce document, mais la nécessité stratégique des architectures ouvertes mérite d'être soulevée : pour exploiter pleinement l’opportunité 1:N offerte par les logiciels et l’IA — en particulier sur des plateformes développées par différents industriels — il est indispensable de définir et d’adopter des architectures ouvertes dans lesquelles ces logiciels et systèmes d’IA peuvent être intégrés.

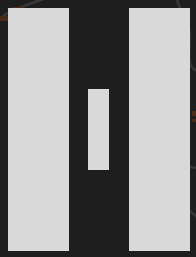
04 La *Software-Defined Assurance* : une nécessité stratégique

Le potentiel transformatif du logiciel et de l’intelligence artificielle dans les conflits modernes est, à juste titre, au centre de l’attention. Cependant, sans une adaptation de l'approche d’assurance dans la défense, ce potentiel restera latent. Le développement et le déploiement des capacités sera ralenti, et les investissements fragmentés. Sur le long terme, cela affectera considérablement nos capacités à faire évoluer nos systèmes d’armes et, plus généralement, à innover au rythme des mutations du champ de bataille.

Investir dans la *software-defined assurance* est essentiel pour exploiter pleinement les opportunités offertes par les mutations du logiciel de défense. Les nations capables de développer des systèmes d’assurance logicielle à grande échelle acquièrent en fait la capacité à transformer leur R&D en avantage décisif sur le champ de bataille.

Une telle transition nécessite une approche nouvelle, centrée sur le logiciel en tant que système distinct. Cela permettra de découpler l’assurance logicielle de celle du matériel, d’en faire un processus continu, et de relever les défis complexes posés par l’autonomie et les *composable forces*. Pour cela, nous devons mobiliser des ressources et des compétences autour de trois axes principaux : l’infrastructure d’assurance logicielle, les actifs numériques contrôlés par les clients, et des normes adaptées aux besoins militaires.

La *software-defined assurance* constitue une véritable boussole stratégique autour de laquelle nous pouvons mobiliser une expertise exceptionnelle, publique et privée, dans l’ensemble des pays de l’OTAN, afin d’accélérer le développement global de nos capacités militaires. Pour atteindre cet objectif, nous devons concentrer notre attention et nos ressources sur trois domaines essentiels : l’infrastructure d’assurance logicielle, les actifs numériques détenus par l'État et de nouvelles normes logicielles adaptées à la défense. Avec ces fondations en place, nous serons en mesure de développer et de mettre à jour des *software-defined capabilities* de manière rapide et fiable.



Helsing