MÉTHODE SAKITI · ARCHITECTE DU SUCCÈS NUMÉRIQUE

CURSUS 12 · AXE D — SÉCURITÉ & GOUVERNANCE

DU CI/CD AU DEVSECOPS

De l'ingénieur CI/CD à l'architecte Zero Trust : quatre jalons, une passerelle précoce, la reconversion du génie logiciel vers la cybersécurité par la chaîne logicielle

23 → 62 → 74 → 63 · F2 → F4 → F4 → F4

MONOGRAPHIE APPROFONDIE · CURSUS & RECONVERSION

ANCRAGE UEMOA · CHAÎNE LOGICIELLE · CYBERSÉCURITÉ

SOMMAIRE

01 Le cursus en un coup d'œil

02 Pourquoi cette voie — le contexte UEMOA

03 Jalon 1 · Ingénieur CI/CD (23)

04 La passerelle F2 → F4 — la reconversion vers la sécurité

05 Jalon 2 · Ingénieur DevSecOps (62)

06 Jalon 3 · Ingénieur AppSec (74)

07 Jalon 4 · Architecte Zero Trust (63)

08 Plan de mise en œuvre sur 30 mois

09 Étude de cas — la trajectoire de Bineta

10 Activer ce cursus — lecture RH & DSI

11 Conclusion

A Annexe A — récapitulatif des quatre jalons

B Annexe B — compétences & certifications

C Annexe C — glossaire & repères

D Annexe D — variantes, bifurcations & cursus voisins

E Annexe E — indicateurs de progression par jalon

F Annexe F — ressources de montée en compétences

G Annexe G — anti-modèles à éviter dans ce cursus

H Annexe H — le « shift-left » : sécuriser au plus tôt

I Annexe I — questions fréquentes

Cette monographie approfondit un unique cursus — le douzième de la collection — jalon par jalon, de l'entrée au sommet, avec son plan de mise en œuvre et son étude de cas.

RÉSUMÉ EXÉCUTIF

Quatre jalons prouvés, une passerelle précoce : comment un ingénieur CI/CD se reconvertit dans la cybersécurité et atteint, en quelques années, l'architecture de sécurité.

Le cursus **« Du CI/CD au DevSecOps »** conduit un ingénieur CI/CD jusqu'au poste d'Architecte Zero Trust. Sa particularité : la passerelle survient tôt et change la nature du parcours. Dès le deuxième jalon, l'ingénieur franchit de la Famille 2 (Génie Logiciel) à la Famille 4 (Cybersécurité) — une reconversion latérale entre deux familles techniques, et non une promotion vers le management. Le reste du cursus se déroule dans la sécurité : DevSecOps, AppSec, puis architecture Zero Trust.

Son enjeu se concentre sur ce franchissement précoce : la reconversion d'un métier technique vers un autre. L'ingénieur n'abandonne pas ce qu'il sait — il le transpose. La compétence-pont est la maîtrise de la chaîne logicielle : intégration et livraison continues, automatisation, infrastructure-as-code. Or le DevSecOps consiste précisément à sécuriser cette chaîne. Ce qui change, c'est la posture : passer de « livrer vite » à « livrer sûr », en pensant comme un adversaire.

Pour les organisations de la zone UEMOA, où la sécurité de la chaîne logicielle devient critique et les profils cyber rares, ce cursus est un instrument de reconversion et de résilience : transformer des ingénieurs CI/CD en spécialistes puis en architectes de la sécurité, à partir d'un vivier qui maîtrise déjà la chaîne à protéger. Ce document en détaille chaque jalon, le plan de mise en œuvre sur trente mois, une étude de cas et la lecture RH & DSI pour l'activer.

LE FIL CONDUCTEUR

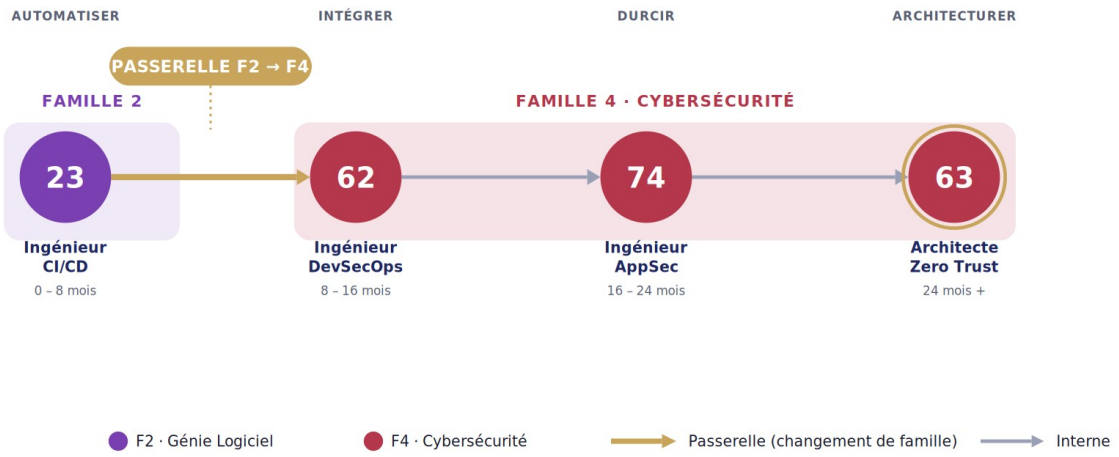
Un principe traverse ce cursus : **décaler la sécurité vers la gauche** — l'intégrer au plus tôt dans la chaîne logicielle, plutôt que de la plaquer à la fin. L'ingénieur CI/CD, qui maîtrise cette chaîne de bout en bout, est idéalement placé pour y intégrer la sécurité. C'est ce « shift-left » qui relie le CI/CD au DevSecOps, puis à l'architecture de sécurité.

01 LE CURSUS EN UN COUP D'ŒIL

De l'automatisation de la chaîne logicielle à l'architecture de sécurité : quatre jalons, une passerelle précoce, la voie qui reconvertit un ingénieur CI/CD vers la cybersécurité.

Le cursus **« Du CI/CD au DevSecOps »** occupe une place singulière dans la collection. Il part d'un poste d'**Ingénieur CI/CD** et conduit jusqu'au poste d'**Architecte Zero Trust**. Sa particularité : la passerelle survient **tôt**. Dès le deuxième jalon, l'ingénieur franchit de la Famille 2 (Génie Logiciel) à la Famille 4 (Cybersécurité). Ce n'est pas une promotion vers le management, mais une **reconversion latérale entre deux familles techniques** — un changement de domaine, pas de posture hiérarchique.

Le reste du cursus se déroule dans la sécurité : DevSecOps, AppSec, puis architecture Zero Trust. L'aboutissement n'est pas un poste de direction, mais un rôle d'architecte — un sommet d'expertise qui synthétise la maîtrise de la chaîne logicielle et la vision de la sécurité. C'est un cursus de reconversion, là où d'autres mènent au management ou à la pure expertise intra-famille.



Les quatre jalons

JALON	RÔLE (N°)	FAMILLE	NIVEAU	DURÉE
1	Ingénieur CI/CD (23)	F2 · Génie Logiciel	Automatiser	0 – 8 mois
2	Ingénieur DevSecOps (62)	F4 · Cybersécurité	Intégrer	8 – 16 mois
3	Ingénieur AppSec (74)	F4 · Cybersécurité	Durcir	16 – 24 mois
4	Architecte Zero Trust (63)	F4 · Cybersécurité	Architecturer	24 mois +

Profil d'entrée, profil de sortie

À l'entrée, un ingénieur qui automatise la chaîne de livraison logicielle. À la sortie, un architecte qui conçoit la sécurité d'ensemble du système. Entre les deux, un changement de famille précoce puis une montée continue — **de l'automatiser à l'intégrer, puis au durcir, puis à l'architecturer**.

> **UNE PASSERELLE LATÉRALE ET PRÉCOCE**
>
> La passerelle **F2 → F4** (jalon 1 → jalon 2) n'est pas un franchissement vers le management, mais un **franchissement latéral entre deux familles techniques** : du génie logiciel à la cybersécurité. Elle survient tôt et s'appuie sur une compétence-pont solide : la maîtrise de la chaîne logicielle, que le DevSecOps a précisément pour mission de

sécuriser. La section 04 lui est entièrement consacrée.

02 POURQUOI CETTE VOIE — LE CONTEXTE UEMOA

La sécurité de la chaîne logicielle devient critique, et les attaques la visent de plus en plus. Reconvertir des ingénieurs CI/CD vers la sécurité est souvent plus rapide que de chercher des spécialistes rares.

Dans la zone UEMOA, la modernisation des services financiers s'appuie sur des chaînes de livraison logicielle de plus en plus automatisées. Cette accélération a un revers : la chaîne logicielle elle-même devient une cible — dépendances compromises, secrets exposés, déploiements détournés. Sécuriser cette chaîne exige une connaissance fine de son fonctionnement. Or les profils capables de le faire — qui réunissent maîtrise du logiciel et culture de la sécurité — sont rares. Il existe pourtant un vivier idéal : les ingénieurs CI/CD, qui maîtrisent déjà la chaîne à protéger.

Ce cursus répond à trois besoins structurels du marché régional. D'abord la **sécurité de la chaîne logicielle** : protéger l'intégration, la livraison et le déploiement contre les attaques. Ensuite la **reconversion de talents techniques** : transformer des ingénieurs CI/CD en spécialistes de la sécurité, plutôt que de chercher l'introuvable. Enfin la **sécurité par conception** : intégrer la sécurité au plus tôt, plutôt que de la plaquer à la fin.

Trois moteurs de la demande

MOTEUR	CE QU'IL IMPOSE	RÔLE DU CURSUS
Attaques sur la chaîne logicielle	Sécuriser dépendances, secrets, déploiements.	Reconvertit l'ingénieur CI/CD vers le DevSecOps.
Pénurie de profils cyber	Trouver autrement que sur un marché tendu.	Forme en interne, à partir d'un vivier d'ingénieurs CI/CD.
Sécurité par conception	Intégrer la sécurité dès la chaîne, pas après coup.	Mène jusqu'à l'architecture de sécurité (Zero Trust).

Pour une DSI ou une institution financière de la zone, ce cursus est un instrument de reconversion et de résilience. Le profil qui en résulte est rare et précieux : un spécialiste — puis un architecte — de la sécurité qui comprend la chaîne logicielle parce qu'il l'a construite. Quand il sécurise un pipeline, il sait exactement où sont les failles, parce qu'il l'a automatisé. Cette profondeur logicielle au service de la sécurité est un avantage décisif.

À qui ce cursus profite dans la zone

EMPLOYEUR	POSTE-CIBLE	POURQUOI CE CURSUS
Banque / établissement financier	Architecte sécurité, ingénieur DevSecOps	Sécuriser une chaîne logicielle qui se modernise.
FinTech / éditeur de logiciels	DevSecOps, AppSec	Protéger des applications et des déploiements fréquents.
Opérateur mobile money	AppSec, architecte Zero Trust	Sécuriser des applications critiques à grande échelle.
ESN / intégrateur	DevSecOps, architecte sécurité	Reconvertir des profils CI/CD vers un domaine porteur.

FIL ROUGE — LA BANQUE WAARIPAY

Tout au long de ce document, nous illustrerons le cursus par le cas d'une banque régionale fictive mais représentative, **WaariPay**, qui automatise la livraison de son système de paiement et choisit de reconvertir ses meilleurs ingénieurs CI/CD vers la sécurité — jusqu'à l'architecture.

LE DIFFÉRENCIATEUR DU PROFIL

Un architecte issu de ce cursus possède un atout que peu partagent : il a **automatisé une chaîne logicielle, intégré la sécurité dans le pipeline, durci des applications** avant de concevoir l'architecture. Là où d'autres architectes

EMPLOYEUR	POSTE-CIBLE	POURQUOI CE CURSUS
raisonnent sur une chaîne qu'ils ne connaissent que de loin, lui sait où frappent les attaques, parce qu'il a construit le pipeline. Sur des systèmes à livraison continue, cette crédibilité logicielle est rare et décisive.		

03 JALON 1 · INGÉNIEUR CI/CD (23)

Famille : F2 · Génie Logiciel **Mouvement :** Départ **Niveau :** Automatiser **Durée :** 0 – 8 mois

Le point d'entrée du cursus : maîtriser la chaîne de livraison logicielle — le terrain même que le DevSecOps devra sécuriser.

Le rôle

L'Ingénieur CI/CD construit et exploite la chaîne de livraison logicielle : intégration continue (build, tests automatisés), livraison et déploiement continus, automatisation, infrastructure-as-code, gestion des artefacts et des dépendances, fiabilité et rapidité de la chaîne. Il est le garant que le code passe du commit à la production de façon rapide, fiable et reproductible. C'est un rôle d'« automatiser » : il bâtit la chaîne — et apprend, en la construisant, comment elle fonctionne, où circulent les secrets, comment elle peut être détournée.

Ce jalon n'est pas propre à la sécurité : c'est un poste de génie logiciel classique. Mais c'est précisément ce qui en fait la force pour ce cursus. Les compétences qui s'y acquièrent — **maîtrise du pipeline, automatisation, gestion des dépendances et des secrets, infrastructure-as-code** — sont exactement celles qui feront un bon ingénieur DevSecOps. On ne sécurise bien une chaîne logicielle que si on sait comment elle est construite. L'ingénieur connaît déjà le terrain ; il lui restera à apprendre à le défendre.

Missions principales

MISSION	EN QUOI ELLE CONSISTE
Construire le pipeline	Build, tests automatisés, intégration continue.
Livrer en continu	Déploiement automatisé, reproductible, fréquent.
Gérer artefacts & dépendances	Versions, registres, traçabilité.
Automatiser l'infrastructure	Infrastructure-as-code, environnements reproductibles.
Fiabiliser la chaîne	Rapidité, stabilité, retours en arrière maîtrisés.

Compétences à acquérir

- Intégration et livraison continues : pipelines, build, tests.
- Automatisation et scripting.
- Infrastructure-as-code et environnements reproductibles.
- Gestion des dépendances, des artefacts et des secrets.
- Conteneurs et déploiement.
- Fiabilité de la chaîne et bonnes pratiques d'ingénierie.

> **LA PREUVE ATTENDUE**
>
> Construire et exploiter une chaîne CI/CD fiable : du commit à la production, automatisée, reproductible et rapide. Le critère : la chaîne livre vite et bien — et l'ingénieur en comprend les rouages assez bien pour, demain, savoir où elle peut être attaquée (dépendances, secrets, déploiements).

Une semaine-type

ACTIVITÉ	PART DU TEMPS
Construction et maintenance du pipeline	≈ 40 %
Automatisation et infrastructure-as-code	≈ 25 %
Gestion des dépendances, artefacts, secrets	≈ 20 %

ACTIVITÉ	PART DU TEMPS
Fiabilité, documentation, collaboration	≈ 15 %

Le mouvement

C'est le **départ** du cursus — et le métier que le professionnel va transposer, non abandonner. Le socle qu'il capitalise ici — pipeline, automatisation, dépendances, secrets — est précisément la compétence-pont qui rendra possible la reconversion vers la sécurité. La qualité de ce socle conditionne la suite : un ingénieur qui comprend en profondeur sa chaîne deviendra un excellent ingénieur DevSecOps ; un ingénieur superficiel peinera.

> **ANCRAGE UEMOA — WAARIPAY**
>
> Chez WaariPay, ce poste automatise la livraison du système de paiement, du commit à la production. C'est le vivier du cursus : la banque y repère les ingénieurs qui comprennent en profondeur leur chaîne, curieux de la sécurité, et leur propose la reconversion.

Pièges à éviter

- Automatiser sans comprendre : assembler des outils sans saisir la chaîne.
- Négliger les secrets et les dépendances : laisser des portes ouvertes dans le pipeline.
- Privilégier la vitesse à tout prix : livrer vite sans jamais penser à la sûreté.

04 LA PASSERELLE F2 → F4 — LA RECONVERSION VERS LA SÉCURITÉ

Le moment où l'ingénieur de la chaîne logicielle devient un professionnel de la sécurité. Une passerelle latérale, précoce, entre deux familles techniques voisines.

Dans la plupart des cursus de la collection, la passerelle est un franchissement vers le management (la Famille 6), placé tard, qui couronne une longue expertise. Celle-ci est différente, comme dans les cursus de reconversion. Elle est **latérale** : elle relie deux familles techniques (Génie Logiciel et Cybersécurité), sans changer de posture — on reste un technicien. Elle est **précoce** : elle survient dès le deuxième jalon. Et elle est une **reconversion** : un changement de domaine, pas une promotion. C'est le sujet même du cursus.

La compétence-pont

Une passerelle ne se franchit jamais à vide : elle s'appuie sur une compétence commune aux deux familles. Ici, cette compétence-pont est **la maîtrise de la chaîne logicielle** : pipeline, automatisation, dépendances, secrets, infrastructure-as-code. Or le DevSecOps **est** cette chaîne, vue sous l'angle de la menace : sécuriser le pipeline, vérifier les dépendances, protéger les secrets, durcir les déploiements. L'ingénieur qui construit déjà cette chaîne possède donc l'essentiel de ce qu'exige sa sécurisation. Il ne repart pas de zéro : il transpose.

Ce qui se transfère, ce qui s'apprend

DIMENSION	CE QUE L'INGÉNIEUR APPORTE	CE QU'IL DOIT APPRENDRE
Le savoir-faire	Pipeline, automatisation, dépendances, secrets	SAST/DAST, modélisation de la menace, durcissement
Le terrain	La chaîne logicielle	La même, vue par l'attaquant
La posture	Livrer vite et bien	Penser en adversaire, livrer sûr
La discipline	Bonnes pratiques d'ingénierie	Sécurité applicative, conformité, secure design

Ce qui se transfère est l'essentiel du métier technique ; ce qui s'apprend est la **posture de sécurité**. L'ingénieur ne change pas de nature — il reste un expert de la chaîne logicielle — mais il apprend à la regarder par les yeux d'un attaquant : où s'introduit une dépendance malveillante, comment fuit un secret, comment un déploiement est détourné. C'est un changement de regard, exigeant mais accessible, qui s'opère d'autant plus vite que le socle logiciel est solide.

Conditions de réussite

- Avoir un socle CI/CD solide : la passerelle s'appuie dessus, elle ne le remplace pas.
- Adopter la posture d'adversaire : penser comme un attaquant de la chaîne.
- Apprendre la discipline de sécurité : modélisation de la menace, secure design.
- Garder l'humilité du débutant en sécurité, tout en valorisant son socle logiciel.

> **LE RISQUE MAJEUR**
>
> Le piège le plus fréquent de cette reconversion est de **croire que le DevSecOps, c'est seulement ajouter des outils au pipeline**. L'ingénieur, sûr de sa maîtrise de l'automatisation, peut réduire la sécurité à l'insertion de scanners dans la chaîne, sans comprendre ce qu'ils détectent ni la dimension adverse : la modélisation de la menace, la logique de l'attaquant, la conception sécurisée. Réussir la passerelle, c'est ajouter à la maîtrise de la chaîne la pensée de l'adversaire — comprendre qu'un pipeline outillé peut encore livrer du code vulnérable.

Pourquoi une passerelle précoce ?

Contrairement aux franchissements vers le management, cette reconversion n'exige pas une longue expertise préalable. Un bon socle d'ingénieur CI/CD suffit. La passerelle survient donc tôt, dès le deuxième jalon, et l'essentiel du

cursus — le DevSecOps, l'AppSec, l'architecture — se déroule dans la nouvelle famille. C'est une logique d'immersion : on bascule vite, puis on approfondit longuement.

05 JALON 2 · INGÉNIEUR DEVSECOPS (62)

Famille : F4 · Cybersécurité **Mouvement :** Passerelle franchie (F2 → F4) **Niveau :** Intégrer **Durée :** 8 – 16 mois

Le premier jalon dans la sécurité : transposer sa maîtrise de la chaîne logicielle à sa défense — intégrer la sécurité dans le pipeline, au plus tôt.

Le rôle

L'Ingénieur DevSecOps intègre la sécurité dans la chaîne de livraison : analyse de sécurité automatisée dans le pipeline (SAST, DAST, SCA), gestion des secrets, sécurité de l'infrastructure-as-code, sécurité de la chaîne d'approvisionnement (dépendances, signatures), politiques de sécurité automatisées. C'est le premier poste de l'ingénieur dans la sécurité — et celui où sa compétence-pont s'exprime pleinement : sécuriser le pipeline, c'est avant tout connaître le pipeline. C'est un rôle d'« intégrer » : il décale la sécurité vers la gauche, au cœur de la chaîne qu'il maîtrise déjà.

Ce jalon est le premier test de la reconversion. L'ingénieur retrouve un terrain familier — pipeline, automatisation, dépendances — mais avec un regard nouveau : celui de l'**adversaire**. Il apprend que chaque dépendance est un risque, que chaque secret mal géré est une porte, que chaque étape du pipeline peut être détournée. Sa connaissance de la chaîne devient un atout de défense : il sait où intégrer les contrôles, parce qu'il sait comment la chaîne est construite. C'est l'incarnation du « shift-left ».

Missions principales

MISSION	EN QUOI ELLE CONSISTE
Intégrer l'analyse de sécurité	SAST, DAST, SCA automatisés dans le pipeline.
Gérer les secrets	Stockage, rotation, accès au moindre privilège.
Sécuriser l'infrastructure-as-code	Vérifier les configurations avant déploiement.
Protéger la supply chain	Vérifier dépendances, signer les artefacts.
Automatiser les politiques	Faire respecter la sécurité sans freiner la livraison.

Compétences à acquérir

- Analyse de sécurité automatisée : SAST, DAST, SCA.
- Gestion des secrets et sécurité de l'infrastructure-as-code.
- Sécurité de la chaîne d'approvisionnement logicielle.
- Politiques de sécurité automatisées (policy-as-code).
- Interprétation des résultats : trier le vrai risque du bruit.
- La posture d'adversaire appliquée à la chaîne logicielle.

LA PREUVE ATTENDUE

Intégrer la sécurité dans une chaîne de livraison en production — par exemple des contrôles automatisés (SAST/DAST/SCA), une gestion saine des secrets et une vérification de la supply chain. Le critère : les vulnérabilités sont détectées et traitées au plus tôt, sans freiner la livraison — la marque d'un profil qui sécurise en connaisseur de la chaîne.

Une semaine-type

ACTIVITÉ	PART DU TEMPS
Intégration des contrôles de sécurité dans le pipeline	≈ 35 %
Gestion des secrets, sécurité de l'IaC	≈ 25 %

ACTIVITÉ	PART DU TEMPS
Sécurité de la supply chain, tri des résultats	≈ 20 %
Politiques automatisées, immersion sécurité	≈ 20 %

Le mouvement

Premier jalon après la **passerelle F2 → F4** (62, dans la nouvelle famille). L'ingénieur confirme sa reconversion en transposant sa maîtrise de la chaîne à sa défense. Ce jalon est décisif : c'est ici qu'il valide que sa compétence-pont — la chaîne logicielle — tient dans le nouveau domaine. La maîtrise acquise prépare le jalon suivant, où il passera de la sécurité de la chaîne à la sécurité des applications elles-mêmes.

> **ANCRAGE UEMOA — WAARIPAY**
>
> Chez WaariPay, l'ingénieur reconverti intègre des contrôles de sécurité dans la chaîne qui livre le système de paiement, et sécurise les secrets et les dépendances. Il sait où placer les garde-fous, parce qu'il a construit cette chaîne. Il devient un ingénieur DevSecOps à part entière, fort de son double socle logiciel et sécurité.

Pièges à éviter

- Réduire le DevSecOps à des outils : empiler des scanners sans comprendre leurs résultats.
- Noyer les équipes d'alertes : générer du bruit sans prioriser le vrai risque.
- Freiner la livraison : une sécurité si lourde qu'elle est désactivée.

06 JALON 3 · INGÉNIEUR APPSEC (74)

Famille : F4 · Cybersécurité **Mouvement :** Trajectoire interne (F4 → F4) **Niveau :** Durcir **Durée :** 16 – 24 mois

De la sécurité de la chaîne à la sécurité des applications : durcir le logiciel lui-même — modéliser la menace, traquer les vulnérabilités, concevoir sûr.

Le rôle

L'Ingénieur AppSec (sécurité applicative) protège les applications elles-mêmes : modélisation de la menace, conception sécurisée (secure design), revue de code de sécurité, gestion des vulnérabilités, sensibilisation des développeurs, référentiels (par exemple OWASP). Là où le DevSecOps sécurise la chaîne, lui sécurise ce qu'elle produit. C'est un rôle de « durcir » : il s'assure que le logiciel livré résiste aux attaques. C'est aussi le jalon où le profil approfondit la pensée de l'adversaire, au-delà de l'outillage.

Ce jalon marque un approfondissement de la posture de sécurité. L'ingénieur AppSec ne se contente plus d'intégrer des contrôles : il **comprend les vulnérabilités** — comment elles naissent, comment on les exploite, comment on les évite par conception. Il modélise la menace pesant sur une application, revoit le code avec un œil offensif, et accompagne les développeurs vers un code plus sûr. C'est ici que l'ancien ingénieur de la chaîne devient un véritable expert de la sécurité du logiciel.

Missions principales

MISSION	EN QUOI ELLE CONSISTE
Modéliser la menace	Identifier ce qui peut être attaqué, et comment.
Concevoir sûr	Intégrer la sécurité dans la conception des applications.
Revoir le code	Repérer les vulnérabilités à l'œil et avec l'outillage.
Gérer les vulnérabilités	Prioriser, corriger, suivre la remédiation.
Accompagner les développeurs	Diffuser une culture de code sécurisé.

Compétences à acquérir

- Modélisation de la menace (threat modeling).
- Conception sécurisée (secure design) et revue de code de sécurité.
- Vulnérabilités applicatives et référentiels (OWASP).
- Gestion et priorisation des vulnérabilités.
- Sensibilisation et accompagnement des développeurs.
- La pensée de l'adversaire appliquée au logiciel.

> **LA PREUVE ATTENDUE**
>
> Élever le niveau de sécurité des applications : une modélisation de la menace conduite, des vulnérabilités majeures traitées, des développeurs sensibilisés et un code plus sûr par conception. Le critère : les applications livrées résistent mieux aux attaques, et la sécurité est comprise au-delà des seuls outils — comme une discipline de conception.

Une semaine-type

ACTIVITÉ	PART DU TEMPS
Modélisation de la menace, conception sécurisée	≈ 30 %
Revue de code de sécurité, gestion des vulnérabilités	≈ 30 %

ACTIVITÉ	PART DU TEMPS
Accompagnement et sensibilisation des développeurs	≈ 20 %
Veille, référentiels, outillage	≈ 20 %

Le mouvement

Trajectoire interne **F4 → F4** (62 → 74). De la sécurité de la chaîne à la sécurité des applications. L'ingénieur DevSecOps approfondit en devenant expert de la sécurité du logiciel lui-même. Ce jalon est décisif pour la suite : c'est en modélisant la menace et en concevant sûr que le professionnel acquiert la **maîtrise de la conception sécurisée** qui, élargie à tout le système, fera l'architecte. On n'architecture bien la sécurité que ce que l'on a su durcir.

> **ANCRAGE UEMOA — WAARIPAY**
> Chez WaariPay, l'ingénieur AppSec modélise la menace sur les applications de paiement, traque leurs vulnérabilités et accompagne les développeurs. Fort de sa connaissance de la chaîne, il relie la sécurité du logiciel à celle de sa livraison — et se prépare à concevoir la sécurité d'ensemble.

Pièges à éviter

- Se fier aux seuls outils : ignorer la modélisation de la menace et la revue humaine.
- Accabler les développeurs : imposer la sécurité sans les accompagner.
- Traiter les vulnérabilités en vrac : corriger sans prioriser par le risque.

07 JALON 4 · ARCHITECTE ZERO TRUST (63)

Famille : F4 · Cybersécurité **Mouvement :** Trajectoire interne (F4 → F4) · Aboutissement **Niveau :** Architecturer **Durée :** 24 mois +

L'aboutissement du cursus : concevoir l'architecture de sécurité d'ensemble — où l'on ne fait jamais confiance par défaut — avec la profondeur logicielle d'un ancien ingénieur CI/CD.

Le rôle

L'Architecte Zero Trust conçoit l'architecture de sécurité de l'organisation, fondée sur un principe simple et exigeant : **ne jamais faire confiance par défaut, toujours vérifier**. Gestion des identités et des accès, moindre privilège, micro-segmentation, vérification continue, chiffrement, sécurité des flux — et l'intégration de tout cela dans la chaîne logicielle et les applications. Là où l'ingénieur AppSec durcit le logiciel, lui conçoit la sécurité de tout le système. C'est un rôle d'« architecturer » — l'aboutissement de la reconversion, et un sommet d'expertise de la Famille 4.

Ce jalon couronne le cursus en faisant la synthèse de tout le parcours. L'architecte issu de cette voie est rare : il possède la **maîtrise de la chaîne logicielle** d'un ancien ingénieur CI/CD, la **culture DevSecOps** du « shift-left », et la **profondeur applicative** d'un ingénieur AppSec. Quand il conçoit une architecture Zero Trust, il sait comment le logiciel est construit, livré et attaqué. Cette continuité — du commit à l'architecture — fait de lui un architecte particulièrement crédible sur les systèmes modernes à livraison continue.

Notons-le : l'aboutissement de ce cursus est un rôle d'**expertise et de conception**, au sein de la Famille 4 — non un poste de management. L'architecte oriente, conçoit, influence — mais ne dirige pas une fonction au sens hiérarchique. C'est un sommet d'expertise, distinct du pilotage (RSSI) et de la gouvernance. Pour qui souhaite ensuite piloter ou gouverner, la voie reste ouverte (voir la conclusion) ; mais le cursus, lui, s'accomplit dans l'architecture.

Le principe directeur : la sécurité par conception

Un principe guide l'architecte : **la sécurité se conçoit, elle ne se rajoute pas**. Une architecture Zero Trust bien pensée rend les attaques structurellement plus difficiles — une compromission isolée ne se propage pas. Pour l'ancien ingénieur de la chaîne, habitué à concevoir des systèmes automatisés, cette logique de sécurité native, intégrée dès la conception, est un prolongement naturel de la pensée « shift-left ».

Missions principales

MISSION	EN QUOI ELLE CONSISTE
Concevoir l'architecture	Identités, accès, segmentation, vérification continue.
Appliquer le moindre privilège	À l'échelle du système, pas d'un seul périmètre.
Intégrer chaîne & applications	Relier sécurité de la livraison, du logiciel et du système.
Définir les standards	Cadres et patterns de sécurité réutilisables.
Orienter les équipes	Guider la mise en œuvre, sans autorité hiérarchique.

Compétences à acquérir

- Architecture Zero Trust : identités, vérification continue, segmentation.
- Conception de sécurité d'ensemble (secure by design).
- Standards et patterns de sécurité réutilisables.
- Intégration de la sécurité de la chaîne, du logiciel et du système.
- Influence et orientation des équipes sans lien hiérarchique.
- Mise au service de l'architecture de tout le socle technique acquis.

LA PREUVE ATTENDUE

Concevoir une architecture de sécurité qui réduit structurellement le risque — par exemple un modèle Zero Trust intégrant identités, segmentation et sécurité de la chaîne logicielle. Le critère : une compromission isolée ne se propage pas, la sécurité est native plutôt que rajoutée, et l'architecture est adoptée par les équipes avec une crédibilité que seul un ancien bâtisseur de la chaîne peut avoir.

Une semaine-type

ACTIVITÉ	PART DU TEMPS
Conception de l'architecture de sécurité	≈ 35 %
Standards, patterns, intégration chaîne / applications	≈ 25 %
Orientation des équipes et revue d'architecture	≈ 20 %
Veille, modélisation de la menace, alignement	≈ 20 %

Le mouvement

Trajectoire interne **F4 → F4** (74 → 63), et **aboutissement du cursus**. La reconversion est complète : l'ingénieur CI/CD est devenu architecte de la sécurité, riche d'un socle technique que peu d'architectes possèdent. Ce sommet d'expertise valorise pleinement le chemin parcouru. Pour qui le souhaite, il peut ensuite ouvrir vers le pilotage (RSSI) puis la gouvernance — précisément les trajectoires tracées par les cursus voisins de l'axe sécurité.

ANCRAGE UEMOA — WAARIPAY

L'Architecte Zero Trust de WaariPay, ancien ingénieur CI/CD reconverti, conçoit la sécurité d'un système de paiement à livraison continue qu'il connaît de bout en bout. Quand il définit un standard, les équipes le suivent — car il a construit la chaîne qu'il sécurise. La banque dispose d'un architecte sécurité rare, capable de relier crédibilité logicielle et vision de la sécurité.

Pièges à éviter

- Rester ingénieur dans sa tête : vouloir tout outiller au lieu de concevoir.
- L'architecture d'ivoire : concevoir sans tenir compte de la réalité des équipes.
- Le Zero Trust de façade : le vocabulaire sans la vérification réelle.

08 PLAN DE MISE EN ŒUVRE SUR 30 MOIS

Un jalon à la fois, une preuve à chaque étape : la cadence qui sécurise la reconversion de l'ingénieur CI/CD à l'architecte Zero Trust.

Le cursus se déploie sur une trentaine de mois, de l'automatisation de la chaîne logicielle à l'architecture de sécurité. La règle est constante : chaque jalon mobilise plusieurs mois, le temps de produire une preuve déployable avant d'ouvrir le suivant. La particularité tient à la passerelle : elle survient tôt, dès le passage du jalon 1 au jalon 2, et l'essentiel du cursus se déroule ensuite dans la sécurité.

PHASE	MOIS	OBJECTIF	PREUVE ATTENDUE
Jalon 1	M1 – M8	Maîtrise de la chaîne CI/CD	Pipeline fiable, automatisé, reproductible
Passerelle	M6 – M9	Reconversion vers la sécurité	Immersion sécurité, posture d'adversaire
Jalon 2	M9 – M16	Sécurité de la chaîne (DevSecOps)	Contrôles intégrés, secrets et supply chain protégés
Jalon 3	M17 – M24	Sécurité des applications (AppSec)	Menace modélisée, vulnérabilités traitées
Jalon 4	M25 – M30+	Architecture de sécurité (Zero Trust)	Architecture native adoptée, risque réduit

Le détail par grandes étapes

PÉRIODE	FOCUS	LIVRABLE DE LA PÉRIODE
M1 – M8	Pipeline, automatisation, dépendances, secrets	Chaîne CI/CD fiable
M6 – M9	Immersion sécurité : posture d'adversaire, menace	Bascule réussie vers la nouvelle famille
M9 – M16	SAST/DAST/SCA, secrets, supply chain, IaC	Chaîne logicielle sécurisée
M17 – M24	Modélisation de la menace, secure design, vulnérabilités	Applications durcies
M25 – M30+	Architecture Zero Trust, standards, intégration	Architecture de sécurité native

GOUVERNANCE DU PARCOURS

Chaque jalon est encadré par un **sponsor** (un dirigeant qui porte le cursus) et un **mentor** (un pair plus avancé). La preuve de fin de jalon est revue formellement avant d'ouvrir le suivant. La passerelle, précoce, mérite un accompagnement particulier : c'est le moment où l'ingénieur adopte la posture de sécurité.

09 ÉTUDE DE CAS — LA TRAJECTOIRE DE BINETA

Comment WaariPay reconvertit en interne, en trente mois, une ingénieure CI/CD en architecte de la sécurité de son système de paiement — sans recrutement externe.

Le point de départ

Bineta est ingénieure CI/CD chez WaariPay : cinq ans d'expérience, une maîtrise solide de la chaîne qui livre le système de paiement, et une curiosité grandissante pour la sécurité — elle a remarqué des dépendances non vérifiées et des secrets mal gérés, et veut agir. La banque, qui veut sécuriser sa chaîne logicielle et peine à recruter des spécialistes, identifie en elle un potentiel et lui propose le cursus « Du CI/CD au DevSecOps ».

La trajectoire, par grandes étapes

ÉTAPE	JALON	PREUVE PRODUITE
Phase 1	Ingénieure CI/CD (23)	Chaîne de livraison fiable et automatisée, du commit à la production.
Passerelle	Reconversion F2 → F4	Immersion réussie dans la sécurité, adoption de la posture d'adversaire.
Phase 2	Ingénieure DevSecOps (62)	Contrôles de sécurité intégrés au pipeline, secrets et supply chain protégés.
Phase 3	Ingénieure AppSec (74)	Menace modélisée, vulnérabilités majeures traitées, développeurs sensibilisés.
Phase 4	Architecte Zero Trust (63)	Architecture de sécurité native adoptée, propagation des intrusions limitée.

Le bilan

En une trentaine de mois, Bineta est passée d'un rôle d'ingénieure CI/CD à l'architecture de la sécurité. Chaque étape s'est validée par une preuve déployable, ce qui a permis à la banque de sécuriser sa reconversion sans pari. WaariPay a transformé une ingénieure de talent en une architecte sécurité rare — capable de concevoir la protection d'un système qu'elle comprend du commit à l'architecture.

Les trois décisions-clés de la direction

La réussite de Bineta ne tient pas au hasard mais à trois décisions de la direction. **D'abord, choisir le bon profil** : une ingénieure rigoureuse et curieuse de la sécurité, pas seulement la plus technique. **Ensuite, accompagner la passerelle** : un soin particulier à l'immersion dans la posture de sécurité, le moment le plus délicat. **Enfin, exiger la preuve** : aucune étape validée sans livrable déployable, ce qui a permis d'avancer sans pari.

Le calcul économique

	RECRUTEMENT EXTERNE	RECONVERSION INTERNE (BINETA)
Délai	6 – 12 mois de recherche, profil sécurité rare	Reconversion planifiée, à partir d'un vivier interne
Coût	Prime de rareté, package élevé, risque d'échec	Accompagnement (mentors, temps) bien moindre
Risque	Méconnaissance de la chaîne logicielle interne	Profil connu, double socle logiciel et sécurité
Crédibilité	Architecte coupé du terrain de livraison	Architecte qui a construit la chaîne qu'il sécurise

RECRUTEMENT EXTERNE	RECONVERSION INTERNE (BINETA)

CE QUE L'EXEMPLE DÉMONTRE

Le cursus n'est pas un plan de formation théorique : c'est un instrument de reconversion. Il transforme un vivier sous-exploité — les ingénieurs CI/CD — en spécialistes puis en architectes de la sécurité, au bénéfice conjoint du collaborateur et de l'organisation.

10 ACTIVER CE CURSUS — LECTURE RH & DSI

De l'intention au pilotage : comment une direction RH ou DSI met concrètement ce cursus de reconversion en mouvement.

Quatre usages pour une direction

USAGE	CE QUE LE CURSUS PERMET
Reconversion de talents CI/CD	Transformer des ingénieurs de la chaîne en spécialistes sécurité.
Réponse à la pénurie cyber	Former en interne plutôt que chercher l'introuvable.
Sécurité de la chaîne logicielle	Protéger la livraison par ceux qui la maîtrisent.
Sécurité par conception	Diffuser le « shift-left » jusqu'à l'architecture.

Une feuille de route en 90 jours

À **J+30**, repérer les ingénieurs CI/CD à fort potentiel (ceux qui ont un bon socle et une curiosité pour la sécurité) et le poste-cible (DevSecOps, puis architecte). À **J+60**, valider l'appétence des candidats pour la sécurité et désigner sponsor et mentors. À **J+90**, lancer le jalon 1 — ou la passerelle si le socle CI/CD est déjà solide — avec une preuve attendue claire et un accompagnement renforcé sur l'immersion sécurité.

Facteurs de réussite & risques

FACTEUR DE RÉUSSITE	RISQUE SYMÉTRIQUE À ÉVITER
Socle CI/CD solide avant la passerelle	Reconvertir un ingénieur au socle fragile
Accompagnement de l'immersion sécurité	Réduire le DevSecOps à l'ajout d'outils
Preuve déployable validée à chaque jalon	Valider la progression sur la seule bonne volonté
Posture d'adversaire acquise	Sécuriser sans penser comme un attaquant

RECOMMANDATION

Documenter ce cursus pour votre organisation, identifier un à trois ingénieurs CI/CD à fort potentiel et curieux de la sécurité, et leur proposer la reconversion. Sur un marché régional où la chaîne logicielle devient une cible et les spécialistes de sa sécurité sont rares, reconvertir en interne est l'option la plus rapide, la moins coûteuse et la plus fidélisante.

11 CONCLUSION

Une voie de reconversion vers la sécurité, à condition d'en respecter la discipline : la preuve à chaque jalon, et une immersion sérieuse dans la posture d'adversaire.

« Du CI/CD au DevSecOps » est, avec les autres cursus de reconversion de la collection, une voie de **franchissement latéral** d'une famille technique vers une autre : ni voie vers le management, ni pure voie d'expertise intra-famille, mais un changement de domaine. Il a pour lui la cohérence : un socle de chaîne logicielle qui se transpose à la sécurité, puis une montée continue jusqu'à l'architecture. Il a contre lui une exigence : la passerelle, bien que précoce et accessible, ne doit pas être prise à la légère. Croire que le DevSecOps, c'est seulement ajouter des outils au pipeline, c'est rater la reconversion.

Pour le professionnel, ce cursus offre une perspective précieuse : quitter un métier d'automatisation pour un domaine porteur et stratégique, sans repartir de zéro — en transposant son socle. Pour l'employeur, il offre un instrument de reconversion et de résilience : faire émerger des spécialistes puis des architectes de la sécurité à partir d'un vivier d'ingénieurs qui maîtrisent déjà la chaîne. Pour la zone UEMOA, il répond à un besoin structurel : sécuriser la chaîne logicielle des systèmes financiers, à partir des compétences techniques déjà présentes.

Trois principes à retenir

- La preuve, pas la promesse : chaque jalon se valide par un livrable déployable.
- Le socle se transpose : la maîtrise de la chaîne logicielle est la compétence-pont vers la sécurité.
- Décaler vers la gauche : la sécurité s'intègre au plus tôt, elle ne se plaque pas à la fin.

Pour aller plus loin

Ce cursus dialogue avec ses voisins. Il est le **pendant « par le dev »** du cursus « De l'infra à la sécurité », qui entre dans la cybersécurité par l'infrastructure : deux portes d'entrée — la chaîne logicielle et l'exploitation — vers la même famille. Et l'**Architecte Zero Trust (63)** peut prolonger vers le pilotage de la sécurité (RSSI) puis la gouvernance IT, comme le tracent les cursus « De l'infra à la sécurité » et « De l'incident à la gouvernance ». Trois cursus dessinent ainsi l'univers de la sécurité : deux portes d'entrée techniques, et une montée vers le pilotage et la gouvernance.

> **ENGAGEMENT**
>
> De l'ingénieur de la chaîne logicielle à l'architecte de la sécurité, un chemin de reconversion lisible et pilotable, prouvé à chaque étape. C'est ainsi qu'un bâtisseur de la chaîne devient un gardien de la sécurité — et que les organisations de la zone protègent leur logiciel à partir des talents qu'elles ont déjà.

A ANNEXE A — RÉCAPITULATIF DES QUATRE JALONS

JALON	RÔLE (N°)	FAMILLE	MOUVEMENT	DURÉE
1	Ingénieur CI/CD (23)	F2	Départ	0 – 8 mois
2	Ingénieur DevSecOps (62)	F4	Passerelle F2 → F4	8 – 16 mois
3	Ingénieur AppSec (74)	F4	Interne	16 – 24 mois
4	Architecte Zero Trust (63)	F4	Interne · Aboutissement	24 mois +

Enchaînement complet : 23 → 62 → 74 → 63 • **Familles :** F2 → F4 → F4 → F4 • **Aboutissement :** Architecte Zero Trust (63).

LA SINGULARITÉ DU CURSUS

Cursus de reconversion latérale par le génie logiciel : sa passerelle relie deux familles techniques (F2 → F4), survient tôt (jalon 1 → 2) et aboutit à un sommet d'expertise (architecture), non au management. Avec le cursus 11 (porte par l'infra), il forme la paire des deux portes d'entrée techniques vers la cybersécurité.

B ANNEXE B — COMPÉTENCES & CERTIFICATIONS PAR JALON

JALON	COMPÉTENCES CLÉS	CERTIFICATION / PREUVE CIBLE
Ingénieur CI/CD (23)	Pipeline, automatisation, IaC, dépendances, secrets	Certifs CI/CD / DevOps
Ingénieur DevSecOps (62)	SAST/DAST/SCA, secrets, supply chain, policy-as-code	Certifs DevSecOps
Ingénieur AppSec (74)	Threat modeling, secure design, OWASP, vulnérabilités	Certifs sécurité applicative
Architecte Zero Trust (63)	Architecture Zero Trust, secure by design, standards	Certifs architecture sécurité

NOTE SUR LES CERTIFICATIONS

Les certifications matérialisent une compétence mais ne la remplacent pas. Aux niveaux techniques, la preuve réelle est la chaîne fiable puis la chaîne et les applications sécurisées. Au niveau architecte, elle se déplace vers une architecture native adoptée. La force du profil tient à la continuité : un architecte qui a tout fait, du pipeline à la conception sécurisée.

C ANNEXE C — GLOSSAIRE & REPÈRES

TERME	DÉFINITION
CI/CD	Intégration et livraison continues : automatisation du passage du code à la production.
Pipeline	Chaîne automatisée d'étapes, du commit au déploiement.
Shift-left	Intégrer la sécurité au plus tôt dans la chaîne, plutôt qu'à la fin.
DevSecOps	Intégration de la sécurité au cœur de la chaîne de développement et de livraison.
SAST / DAST / SCA	Analyses de sécurité : du code, de l'application en exécution, des dépendances.
AppSec	Sécurité applicative : protéger le logiciel lui-même contre les attaques.
Threat modeling	Modélisation de la menace : identifier ce qui peut être attaqué, et comment.
Supply chain logicielle	Chaîne de production du logiciel, de la dépendance au déploiement — une cible.
Zero Trust	Modèle où l'on ne fait jamais confiance par défaut : vérifier toujours.
Reconversion latérale	Changement de famille technique sans changement de posture (pas vers le management).

D ANNEXE D — VARIANTES, BIFURCATIONS & CURSUS VOISINS

« Du CI/CD au DevSecOps » n'est pas l'unique destin d'un ingénieur CI/CD. Plusieurs bifurcations s'ouvrent en chemin, et son aboutissement peut lui-même se prolonger.

Ce cursus décrit une voie de reconversion directe, mais le graphe des compétences en autorise d'autres. L'**Ingénieur DevSecOps (62)** comme l'**Ingénieur AppSec (74)** sont des **rôles-charnières** : depuis ces points, plusieurs trajectoires divergent. Connaître ces bifurcations aide une direction à orienter chaque talent.

Trois bifurcations fréquentes

BIFURCATION	POINT DE DÉPART	VERS	CURSUS VOISIN
Vers le pilotage sécurité	Architecte Zero Trust (63)	RSSI, puis gouvernance IT	Cursus 11 & 10 (sécurité)
Vers le pentest / l'offensif	Ingénieur AppSec (74)	Pentester, red team	Voie offensive de la sécurité
Retour vers le logiciel	Ingénieur DevSecOps (62)	Architecte logiciel	F4 → F2 (si la sécurité ne convient pas)

Deux portes vers la sécurité technique

Ce cursus est le jumeau du cursus « De l'infra à la sécurité » : tous deux sont des reconversions vers la cybersécurité, mais par deux portes différentes — la chaîne logicielle pour l'un (du dev), l'exploitation pour l'autre (de l'infra). L'architecte issu du dev excelle sur la sécurité de la chaîne et des applications ; celui issu de l'infra, sur la sécurité des plateformes et des conteneurs. Une organisation gagne à disposer des deux, complémentaires.

> **LECTURE**
> Présenter ces bifurcations dès l'entrée du cursus est un atout RH : le collaborateur comprend que la reconversion n'est pas un cul-de-sac, mais une porte ouvrant sur tout l'univers de la sécurité, jusqu'au pilotage et à la gouvernance. Cela renforce l'engagement.

E ANNEXE E — INDICATEURS DE PROGRESSION PAR JALON

Comment objectiver l'avancement : pour chaque jalon, un indicateur de maîtrise et sa cible, à revoir avant d'ouvrir le suivant.

JALON	INDICATEUR DE MAÎTRISE	CIBLE DE FIN DE JALON
1 · Ingénieur CI/CD (23)	Fiabilité de la chaîne	Pipeline stable, automatisé, reproductible
Passerelle F2 → F4	Réussite de l'immersion	Posture d'adversaire acquise
2 · DevSecOps (62)	Sécurité de la chaîne	Contrôles intégrés, secrets et supply chain protégés
3 · AppSec (74)	Sécurité des applications	Menace modélisée, vulnérabilités majeures traitées
4 · Architecte Zero Trust (63)	Architecture native	Propagation limitée, architecture adoptée

PRINCIPE

Un indicateur n'a de valeur que s'il déclenche une décision : ouvrir le jalon suivant, prolonger le jalon en cours, ou réorienter. La passerelle mérite une revue spécifique : a-t-on réellement adopté la posture d'adversaire, ou seulement ajouté des outils au pipeline ?

F ANNEXE F — RESSOURCES DE MONTÉE EN COMPÉTENCES

Pour chaque jalon, les modalités d'apprentissage les plus efficaces : la pratique d'abord, la preuve ensuite, le mentorat tout du long.

JALON	MODALITÉS DE MONTÉE EN COMPÉTENCES
Ingénieur CI/CD (23)	Construction de pipelines réels, automatisation, infrastructure-as-code ; mentorat d'un ingénieur CI/CD senior.
Passerelle F2 → F4	Immersion dans la posture de sécurité : modélisation de la menace, exercices offensifs sur la chaîne, accompagnement renforcé.
Ingénieur DevSecOps (62)	Intégration réelle de contrôles (SAST/DAST/SCA), sécurisation des secrets et de la supply chain ; mentorat d'un DevSecOps.
Ingénieur AppSec (74)	Modélisation de la menace, revue de code de sécurité, exercices OWASP ; mentorat d'un spécialiste AppSec.
Architecte Zero Trust (63)	Conception d'architecture, définition de standards, revue d'architecture ; mentorat d'un architecte sécurité.

LA RÈGLE CONSTANTE

À tous les jalons, l'ordre est le même : **pratiquer, prouver, puis formaliser**. La passerelle fait exception par l'importance de l'immersion : on n'adopte pas la posture d'adversaire dans un manuel, mais en confrontant sa chaîne logicielle à la menace réelle.

G ANNEXE G — ANTI-MODÈLES À ÉVITER DANS CE CURSUS

Les erreurs récurrentes qui font dérailler la reconversion — et l'antidote correspondant, à garder en tête à chaque jalon.

ANTI-MODÈLE	POURQUOI IL NUIT	ANTIDOTE
Le socle fragile	Reconvertir un ingénieur médiocre échoue : la passerelle s'appuie dessus.	Consolider le socle CI/CD avant de basculer.
Le DevSecOps-outils	Réduire la sécurité à l'ajout de scanners oublie l'adversaire.	Comprendre les résultats, modéliser la menace.
Le bruit d'alertes	Noyer les équipes sous des alertes non priorisées les fait ignorer.	Trier par le risque, ne remonter que l'essentiel.
L'AppSec qui accable	Imposer la sécurité aux développeurs sans les accompagner.	Sensibiliser, outiller, accompagner.
L'architecture d'ivoire	Concevoir hors-sol, sans tenir compte des équipes.	Concevoir avec le terrain, faire adopter.
L'ingénieur resté outilleur	Vouloir tout automatiser au lieu de concevoir dénature le rôle.	Assumer le passage de l'outillage à l'architecture.

À RETENIR

La plupart des échecs de ce cursus tiennent à deux causes : un socle CI/CD insuffisant au départ, ou une sécurité réduite à l'outillage, sans posture d'adversaire. Nommer ces anti-modèles, c'est déjà commencer à les éviter.

H ANNEXE H — LE « SHIFT-LEFT » : SÉCURISER AU PLUS TÔT

Un primer sur le principe qui fonde ce cursus : intégrer la sécurité au plus tôt dans la chaîne logicielle, là où elle coûte le moins et protège le plus.

Le « **shift-left** » est l'idée directrice du DevSecOps : déplacer la sécurité vers la gauche de la chaîne — c'est-à-dire vers les premières étapes du développement et de la livraison — plutôt que de la traiter à la fin, juste avant la mise en production. Plus une vulnérabilité est détectée tôt, moins elle coûte à corriger, et moins elle a de chances d'atteindre la production. C'est précisément ce qui fait de l'ingénieur CI/CD le profil idéal pour cette reconversion : il maîtrise déjà la chaîne où la sécurité doit s'intégrer.

ÉTAPE DE LA CHAÎNE	CE QUE LE SHIFT-LEFT Y APPORTE
Conception	Modélisation de la menace, sécurité par conception.
Développement	Revue de code de sécurité, bonnes pratiques outillées.
Intégration (build)	Analyse des dépendances (SCA), analyse statique (SAST).
Test	Analyse dynamique (DAST), vérification des configurations.
Livraison / déploiement	Gestion des secrets, signatures, durcissement.
Production	Détection, supervision — mais l'essentiel est déjà traité en amont.

Pourquoi le shift-left change tout

Trois raisons font du shift-left un principe décisif. Le **coût** : une vulnérabilité corrigée à la conception coûte une fraction de ce qu'elle coûterait en production. La **couverture** : intégrée à la chaîne, la sécurité s'applique à chaque livraison, automatiquement, sans dépendre d'un audit ponctuel. La **culture** : en rapprochant la sécurité des développeurs, le shift-left en fait l'affaire de tous, et non d'une équipe isolée. Le cursus tout entier est une montée en maîtrise de ce principe — du pipeline (jalon 1) à l'architecture qui l'incarne (jalon 4).

> **LECTURE**
> Comprendre le shift-left éclaire pourquoi le cursus part du CI/CD : c'est dans la chaîne logicielle que la sécurité moderne se joue d'abord. L'ingénieur qui maîtrise cette chaîne tient le levier le plus puissant de la sécurité — l'intégrer au plus tôt, partout, automatiquement.

I ANNEXE I — QUESTIONS FRÉQUENTES

Les interrogations les plus courantes des directions et des candidats au moment d'engager ce cursus.

Pourquoi la passerelle survient-elle si tôt ?

Parce que cette passerelle est une reconversion latérale, non un franchissement vers le management. Elle n'exige pas une longue expertise préalable : un bon socle d'ingénieur CI/CD suffit. On bascule donc tôt vers la sécurité, dès le deuxième jalon, puis on approfondit longuement dans le nouveau domaine. C'est une logique d'immersion précoce.

Un ingénieur CI/CD peut-il vraiment devenir architecte sécurité ?

Oui, et il y excelle souvent sur les systèmes modernes. L'architecte issu de ce cursus possède une maîtrise de la chaîne logicielle que peu d'architectes ont : il comprend comment le logiciel est construit, livré et attaqué. À condition d'assumer le passage de l'outillage à la conception — de « faire » à « concevoir » —, c'est un profil rare et crédible.

En quoi ce cursus diffère-t-il de « De l'infra à la sécurité » ?

Les deux sont des reconversions vers la cybersécurité, mais par deux portes différentes : la chaîne logicielle pour celui-ci (du dev), l'exploitation pour l'autre (de l'infra). L'architecte issu du dev excelle sur la sécurité de la chaîne et des applications ; celui issu de l'infra, sur la sécurité des plateformes et des conteneurs. Ce sont deux voies complémentaires vers la même famille.

Faut-il abandonner l'automatisation ?

L'automatisation n'est pas abandonnée, elle est transposée puis dépassée. Aux jalons 2 et 3, le profil automatise encore beaucoup — il intègre des contrôles, outille la sécurité. Au jalon 4 (architecte), il conçoit davantage qu'il n'outille, mais sa maîtrise de l'automatisation reste un atout permanent. Le cursus ne renie pas l'ingénieur CI/CD : il l'élève vers un rôle où sa maîtrise de la chaîne sert la sécurité.

Et si la sécurité ne convient finalement pas ?

La reconversion n'est pas irréversible. Un ingénieur qui découvre que la sécurité ne lui convient pas peut revenir au génie logiciel : son socle est intact, et il aura gagné une culture de la sécurité précieuse partout. La passerelle précoce limite d'ailleurs le risque : on teste la sécurité tôt, avant d'y avoir tout investi.

Ce cursus convient-il aux petites structures ?

Oui, et particulièrement. Dans une petite structure, reconvertir un ingénieur CI/CD interne vers la sécurité est souvent la voie la plus réaliste face à la pénurie de spécialistes. La logique des jalons et des preuves reste la même ; seules la taille des équipes et la séparation des rôles s'adaptent aux moyens.

> **EN RÉSUMÉ**
>
> Ce cursus n'est ni automatique ni universel : c'est une voie de reconversion, à activer pour des ingénieurs CI/CD solides et curieux de la sécurité, dans des organisations où la protection de la chaîne logicielle est un enjeu stratégique.

*Ce document constitue l'approfondissement du **Cursus 12 · Du CI/CD au DevSecOps**, douzième des treize cursus de la collection « Les cursus intra & inter familles » de la Méthode SAKITI.*