Système de veille technologique et sécurité - Application TaskFlow

1. Introduction

Ce document décrit le système de veille technologique mis en place pour suivre les évolutions, les mises à jour de sécurité et les bonnes pratiques liées au déploiement de l'application TaskFlow développée avec Symfony 7.3.4.

Date de dernière mise à jour : 17 octobre 2025

Responsable de la veille : [NOM DU RESPONSABLE]

2. Objectifs de la veille

2.1 Objectifs principaux

- Assurer la sécurité de l'application en identifiant rapidement les vulnérabilités
- Maintenir la stack technique à jour avec les dernières versions stables
- Anticiper les problèmes de compatibilité
- Améliorer continuellement les processus de déploiement
- Rester informé des bonnes pratiques DevOps
- Optimiser les performances de l'application

2.2 Périmètre de la veille

Technologies surveillées :

- Symfony et ses composants
- PHP et ses extensions
- MySQL
- Apache/Nginx
- Composer et dépendances PHP
- AssetMapper
- Doctrine ORM
- Outils DevOps et CI/CD

3. Organisation de la veille

3.1 Fréquence de consultation

Type de veille	Fréquence	Responsable	Durée estimée
Alertes sécurité critiques	Quotidienne	Responsable technique	15 min
Mises à jour Symfony	Hebdomadaire	Développeur senior	30 min
Mises à jour PHP	Hebdomadaire	Développeur senior	15 min
Dépendances Composer	Hebdomadaire	Développeur	20 min
Veille DevOps	Mensuelle	DevOps/SysAdmin	1h
Veille générale tech	Mensuelle	Toute l'équipe	2h

3.2 Responsabilités

Responsable technique :

- Coordination générale de la veille
- Décision sur les mises à jour critiques
- Validation des actions correctives

Développeurs :

- Surveillance des dépendances
- Tests des nouvelles versions
- Implémentation des correctifs

Administrateur système :

- Surveillance infrastructure
- Mises à jour système
- Monitoring des performances

3.3 Processus de remontée d'information

```
Détection → Évaluation → Priorisation → Action → Documentation
```

1. **Détection :** Identification d'une information pertinente
2. **Évaluation :** Analyse de l'impact potentiel sur TaskFlow
3. **Priorisation :** Classification selon le niveau de criticité
4. **Action :** Planification et mise en œuvre
5. **Documentation :** Traçabilité dans un registre

4. Sources de veille Symfony

4.1 Sources officielles

Site officiel Symfony

- URL : https://symfony.com/blog
- Fréquence de consultation : Hebdomadaire
- Contenu : Annonces officielles, nouvelles versions, articles techniques

Documentation Symfony

- URL : https://symfony.com/doc/current/
- Fréquence de consultation : Lors de chaque mise à jour majeure
- Contenu : Guide de migration, changelog, nouveautés

Symfony Releases

- URL : https://symfony.com/releases
- Fréquence de consultation : Hebdomadaire
- Contenu : Calendrier de sortie des versions, roadmap

Security Advisories Symfony

- URL : https://symfony.com/security
- Fréquence de consultation : Quotidienne
- Contenu : Alertes de sécurité, CVE, correctifs
- **CRITIQUE :** Mise en place d'alertes email automatiques

4.2 Outils automatisés Symfony

Symfony Security Checker (intégré à Composer)

```
# Vérification des vulnérabilités
composer audit

# À exécuter lors de chaque déploiement et hebdomadairement
```

Commande d'analyse

```
# Vérifier les mises à jour disponibles
composer outdated

# Afficher les dépendances obsolètes
composer show --outdated
```

4.3 Communauté Symfony

SymfonyCasts

- URL : https://symfonycasts.com
- Contenu : Tutoriels, screencasts, formations

Symfony GitHub

- URL : https://github.com/symfony/symfony
- Fréquence : Hebdomadaire
- Contenu : Issues, pull requests, discussions

Forum Symfony

- URL : https://symfony.com/community
- Contenu : Questions, problèmes courants, solutions

5. Sources de veille PHP

5.1 Sources officielles PHP

PHP.net

- URL : https://www.php.net
- Sections importantes :
  - Releases : https://www.php.net/releases/
  - Security : https://www.php.net/security/
  - Migration guides : https://www.php.net/manual/en/appendices.php

PHP Watch

- URL : https://php.watch
- Contenu : Analyses détaillées des nouvelles versions PHP
- Fréquence : Mensuelle

5.2 Cycle de vie PHP

Versions supportées (octobre 2025) :

- PHP 8.3 : Support actif jusqu'au 23 novembre 2026
- PHP 8.2 : Support de sécurité jusqu'au 8 décembre 2026
- PHP 8.1 : Support de sécurité jusqu'au 25 novembre 2025

Action requise :
Planifier la migration vers PHP 8.4 avant fin 2026.

5.3 Extensions PHP

Surveillance des extensions utilisées par TaskFlow :

- pdo_mysql
- intl
- opcache
- mbstring
- xml
- ctype
- iconv

Vérification mensuelle : `php -m  php -v`

6. Sources de veille sécurité

6.1 Bases de données de vulnérabilités

CVE (Common Vulnerabilities and Exposures)

- URL : https://cve.mitre.org
- Recherche par mot-clé : "Symfony", "PHP", "MySQL", "Apache"
- Fréquence : Hebdomadaire

NVD (National Vulnerability Database)

- URL : https://nvd.nist.gov
- Alertes configurées pour : PHP, Symfony, MySQL
- Consultation automatique via flux RSS

Snyk Vulnerability Database

- URL : https://snyk.io/vuln
- Contenu : Vulnérabilités dans les packages Composer
- Intégration possible avec GitHub/GitLab

6.2 Outils de sécurité

Composer Audit (intégré)

```
# Analyse des vulnérabilités dans les dépendances
composer audit

# À exécuter quotidiennement en automatique
```

GitHub Dependabot

- Activation sur le dépôt TaskFlow
- Alertes automatiques sur les dépendances vulnérables
- Pull requests automatiques de mise à jour

Snyk (optionnel)

- Analyse continue des dépendances
- Intégration CI/CD
- Alertes en temps reel

6.3 OWASP

OWASP Top 10

- URL : https://owasp.org/www-project-top-ten/
- Consultation : Annuelle
- Application : Vérification de conformité de TaskFlow

OWASP Cheat Sheets

- URL : https://cheatsheetseries.owasp.org
- Sections pertinentes :
    - SQL Injection Prevention
    - Cross Site Scripting Prevention
    - Session Management
    - Authentication

7. Sources de veille DevOps

7.1 Outils de déploiement

Deployer

- URL : https://deployer.org
- Fréquence : Trimestrielle
- Contenu : Nouvelles fonctionnalités, recettes de déploiement

Ansible

- URL : https://www.ansible.com
- Si utilisation future pour l'automatisation

Docker

- URL : https://www.docker.com/blog
- Si conteneurisation envisagée

7.2 CI/CD

GitLab CI/CD

- URL : https://docs.gitlab.com/ee/ci/
- Fréquence : Mensuelle
- Contenu : Nouvelles fonctionnalités, optimisations

GitHub Actions

- URL : https://github.com/features/actions
- Si migration envisagée

7.3 Monitoring et observabilité

Sentry

- URL : https://sentry.io/blog
- Contenu : Bonnes pratiques de monitoring d'erreurs

New Relic / Datadog

- Si outils de monitoring APM utilisés

8. Sources de veille base de données

8.1 MySQL

MySQL Official Blog

- URL : https://blogs.oracle.com/mysql/
- Fréquence : Mensuelle
- Contenu : Nouvelles versions, optimisations, sécurité

MySQL Release Notes

- URL : https://dev.mysql.com/doc/relnotes/mysql/8.0/en/
- Fréquence : Lors de chaque nouvelle version

Percona Blog

- URL : https://www.percona.com/blog
- Contenu : Optimisations, bonnes pratiques MySQL

8.2 Doctrine ORM

Doctrine Project

- URL : https://www.doctrine-project.org/projects.html
- Fréquence : Trimestrielle
- Contenu : Mises à jour Doctrine ORM, DBAL

Doctrine GitHub

- URL : https://github.com/doctrine
- Surveillance des releases

9. Newsletters et agrégateurs

9.1 Newsletters recommandées

Symfony Station

- URL : https://www.symfonystation.com
- Fréquence : Hebdomadaire
- Contenu : Actualités Symfony, articles, packages

PHP Weekly

- URL : https://www.phpweekly.com
- Fréquence : Hebdomadaire
- Contenu : Articles PHP, outils, tutoriels

DevOps Weekly

- URL : https://www.devopsweekly.com
- Fréquence : Hebdomadaire
- Contenu : Actualités DevOps, outils, articles

9.2 Flux RSS

Configuration d'un agrégateur RSS (Feedly, Inoreader) avec les flux suivants :

- https://symfony.com/blog.rss
- https://www.php.net/feed.atom
- https://feeds.feedburner.com/symfony/blog

10. Critères de criticité

10.1 Classification des alertes

CRITIQUE - Action immédiate (< 24h)

- Faille de sécurité exploitable en production
- Vulnérabilité CVE avec score CVSS ≥ 9.0
- Bug critique bloquant des fonctionnalités principales
- Perte de données potentielle

URGENT - Action rapide (< 1 semaine)

- Vulnérabilité CVE avec score CVSS entre 7.0 et 8.9
- Mise à jour de sécurité importante
- Bug majeur impactant les utilisateurs
- Fin de support annoncée d'une technologie utilisée

IMPORTANT - Planification (< 1 mois)

- Mise à jour mineure de Symfony
- Vulnérabilité CVE avec score CVSS entre 4.0 et 6.9
- Optimisation de performance significative
- Nouvelle fonctionnalité majeure

NORMAL - Planification souple (< 3 mois)

- Mises à jour mineures de dépendances
- Améliorations de fonctionnalités
- Optimisations non critiques
- Nouvelles bonnes pratiques

INFORMATIF - Veille passive

- Articles de blog techniques
- Annonces de roadmap
- Retours d'expérience
- Tendances du marché

10.2 Matrice de décision

Criticité	Délai d'action	Validation requise	Tests nécessaires
CRITIQUE	< 24h	Responsable technique	Tests minimaux
URGENT	< 1 semaine	Responsable technique	Tests complets
IMPORTANT	< 1 mois	Équipe dev	Tests complets + staging
NORMAL	< 3 mois	Planning sprint	Tests complets + staging
INFORMATIF	Aucun	N/A	N/A

11. Procédure de traitement des informations

11.1 Workflow de traitement

```
Détection → Analyse → Évaluation → Décision → Action → Suivi
```

Étape 1 : Détection

- Identification d'une information pertinente
- Source et date de publication
- Périmètre concerné (Symfony, PHP, sécurité, etc.)

Étape 2 : Analyse

- Lecture et compréhension de l'information
- Vérification de la fiabilité de la source
- Recherche d'informations complémentaires

Étape 3 : Évaluation

- Impact potentiel sur TaskFlow
- Classification selon la criticité
- Estimation de l'effort de mise en œuvre

Étape 4 : Décision

- Go/No-Go pour l'implémentation
- Planification dans le backlog
- Attribution à un responsable

Étape 5 : Action

- Mise en œuvre du changement
- Tests en environnement de staging
- Déploiement en production

Étape 6 : Suivi

- Documentation de l'action effectuée
- Vérification de l'efficacité
- Mise à jour du registre de veille

11.2 Registre de veille

Un fichier `REGISTRE_VEILLE.md` doit être maintenu avec le format suivant :

```
# Registre de veille technologique

## [2025-10-17] Vulnérabilité Symfony CVE-2025-XXXX

**Source :** https://symfony.com/security/...
**Criticité :** CRITIQUE
**Impact :** Injection SQL possible dans FOSUserBundle
**Action :** Mise à jour immédiate vers Symfony 7.3.5
**Responsable :** Jean Dupont
**Statut :** Déployé en production le 2025-10-17
**Durée :** 4 heures
```

11.3 Template de fiche de veille

```
# Fiche de veille - [TITRE]

**Date de détection :** JJ/MM/AAAA
**Source :** [URL]
**Catégorie :** Symfony / PHP / Sécurité / DevOps / Autre
**Criticité :** CRITIQUE / URGENT / IMPORTANT / NORMAL / INFORMATIF

## Description
[Description concise de l'information]

## Impact sur TaskFlow
[Analyse de l'impact potentiel]

## Action recommandée
[Ce qui devrait être fait]

## Décision
- [ ] Mise en œuvre immédiate
- [ ] Planification court terme (< 1 mois)
- [ ] Planification long terme (> 1 mois)
- [ ] Aucune action nécessaire

## Responsable
[Nom de la personne en charge]

## Date limite
[Date cible de mise en œuvre]

## Suivi
[État d'avancement et notes]
```

12. Automatisation de la veille

12.1 Scripts d'automatisation

Script de vérification quotidienne des vulnérabilités

```
#!/bin/bash
# check_vulnerabilities.sh
# À exécuter quotidiennement via cron

PROJECT_DIR="/var/www/taskflow/current"
LOG_FILE="/var/log/taskflow/security_check.log"
EMAIL_ALERT="tech@taskflow.app"

cd $PROJECT_DIR

echo "[$(date)] Vérification des vulnérabilités" >> $LOG_FILE

composer audit --format=json > /tmp/audit_result.json

if [ $? -ne 0 ]; then
    echo "ALERTE: Vulnérabilités détectées!" >> $LOG_FILE
    cat /tmp/audit_result.json >> $LOG_FILE

    mail -s "ALERTE SECURITE TaskFlow" $EMAIL_ALERT < /tmp/audit_result.json
fi

rm /tmp/audit_result.json
```

Configuration cron

```
# Vérification quotidienne à 8h00
0 8 * * * /usr/local/bin/check_vulnerabilities.sh
```

Script de vérification des mises à jour

```
#!/bin/bash
# check_updates.sh
# À exécuter hebdomadairement

PROJECT_DIR="/var/www/taskflow/current"
REPORT_FILE="/var/log/taskflow/updates_report_$(date +%Y%m%d).txt"

cd $PROJECT_DIR

echo "=== Rapport de mise à jour - $(date) ===" > $REPORT_FILE
echo "" >> $REPORT_FILE

echo "Mises à jour Composer disponibles:" >> $REPORT_FILE
composer outdated >> $REPORT_FILE

echo "" >> $REPORT_FILE
echo "Version PHP actuelle:" >> $REPORT_FILE
php -v >> $REPORT_FILE
```

```
echo "" >> $REPORT_FILE
echo "Version Symfony actuelle:" >> $REPORT_FILE
php bin/console --version >> $REPORT_FILE

mail -s "Rapport hebdomadaire - Mises à jour TaskFlow" tech@taskflow.app < $REPORT_FILE
```

12.2 Intégration GitHub/GitLab

Configuration Dependabot (GitHub)

Fichier `.github/dependabot.yml` :

```yaml
version: 2
updates:
  - package-ecosystem: "composer"
    directory: "/"
    schedule:
      interval: "weekly"
      day: "monday"
      time: "09:00"
    open-pull-requests-limit: 10
    reviewers:
      - "responsable-technique"
    assignees:
      - "dev-senior"
    labels:
      - "dependencies"
      - "security"
    commit-message:
      prefix: "deps"
      include: "scope"
```

Configuration GitLab Dependency Scanning

Fichier `.gitlab-ci.yml` (extrait) :

```yaml
include:
  - template: Security/Dependency-Scanning.gitlab-ci.yml
  - template: Security/License-Scanning.gitlab-ci.yml

dependency_scanning:
  variables:
    DS_EXCLUDED_PATHS: "vendor/"
```

12.3 Alertes automatiques

Configuration d'alertes Google

- Mots-clés : "Symfony security", "PHP vulnerability", "CVE PHP"
- Fréquence : Au fur et à mesure
- Destination : tech@taskflow.app

Flux RSS configurés

- Symfony Blog : https://symfony.com/blog.rss
- PHP News : https://www.php.net/feed.atom
- GitHub Symfony : https://github.com/symfony/symfony/releases.atom

13. Veille spécifique DevOps

13.1 Pratiques de déploiement continu

Sources de veille CI/CD :

- GitLab CI/CD Documentation
- GitHub Actions Marketplace
- Jenkins Blog

Métriques à surveiller :

- Temps de déploiement moyen
- Taux de succès des déploiements
- Fréquence des rollbacks
- Temps de détection des erreurs

13.2 Conteneurisation et orchestration

Si passage à Docker prévu :

- Docker Blog : https://www.docker.com/blog
- Docker Hub : images officielles PHP et Nginx
- Best practices Docker pour Symfony

Si passage à Kubernetes prévu :

- Kubernetes Blog
- Helm Charts pour Symfony
- Operators Kubernetes

13.3 Infrastructure as Code

Outils à surveiller :

- Terraform (HashiCorp)
- Ansible
- Puppet/Chef

Bonnes pratiques :

- Versioning de l'infrastructure
- Tests d'infrastructure
- Documentation as Code

14. Formation et partage de connaissance

14.1 Réunions de veille

Réunion mensuelle de veille technologique

- Durée : 2 heures
- Participants : Toute l'équipe technique
- Ordre du jour :
    1. Tour de table des découvertes du mois
    2. Présentation des alertes critiques traitées
    3. Discussion sur les technologies émergentes
    4. Planification des actions pour le mois suivant

Format de présentation :

```
# Veille tech - Mois de [MOIS ANNEE]

## Alertes critiques
- [Alerte 1] → Action prise
- [Alerte 2] → En cours

## Mises à jour effectuées
- Symfony 7.3.4 → 7.3.5 (correctif sécurité)
- Doctrine ORM 3.0.1 → 3.0.2

## Découvertes intéressantes
- [Nouvelle fonctionnalité Symfony]
- [Outil DevOps à tester]

## Actions pour le mois prochain
- [ ] Tester la nouvelle version de PHP 8.4 en dev
- [ ] Évaluer l'outil X pour le monitoring
```

14.2 Documentation interne

Wiki technique interne

- Créer une section "Veille technologique"
- Documenter les décisions architecturales
- Partager les retours d'expérience

Base de connaissances

- Solutions aux problèmes rencontrés
- Tutoriels internes
- Procédures de mise à jour

14.3 Formation continue

Plan de formation annuel :

- Budget formation : [MONTANT]€/an/développeur
- Formations prioritaires :
    - Certification Symfony
    - Sécurité des applications web
    - DevOps et automatisation

Ressources de formation :

- SymfonyCasts (abonnement équipe)
- Udemy / Pluralsight
- Conférences (SymfonyCon, PHP Conference)

15. Outils de gestion de la veille

15.1 Agrégateurs RSS

Recommandations :

- Feedly (https://feedly.com) - Gratuit avec limite
- Inoreader (https://www.inoreader.com) - Plus de fonctionnalités
- NetNewsWire (macOS/iOS) - Open source

Configuration type :

```
Dossier "Symfony"
├── Symfony Blog
├── Symfony Security
└── SymfonyCasts

Dossier "PHP"
├── PHP.net News
├── PHP Watch
└── PHP Weekly

Dossier "Sécurité"
├── CVE Recent
├── OWASP
└── Snyk Blog

Dossier "DevOps"
├── GitLab Blog
├── Docker News
└── DevOps Weekly
```

15.2 Gestionnaire de tâches

Utilisation d'un outil de gestion de projet :

- Jira, Trello, ou équivalent
- Création d'un tableau "Veille technologique"
- Colonnes : Détecté / En analyse / Planifié / En cours / Terminé

Labels recommandés :

- `veille-securite`
- `veille-symfony`
- `veille-php`
- `veille-devops`
- `critique`

- `urgent`
- `amélioration`

15.3 Outil de communication

Slack/Teams/Discord :

- Canal dédié : #veille-tech
- Partage quotidien des découvertes
- Intégrations possibles :
    - GitHub/GitLab notifications
    - RSS feeds
    - Alertes Dependabot

16. Indicateurs de performance (KPI)

16.1 Métriques de suivi

Indicateur	Objectif	Fréquence mesure
Temps moyen de correction vulnérabilité critique	< 24h	Mensuelle
Nombre de vulnérabilités détectées	Tracking	Mensuelle
Taux de mises à jour appliquées	> 90%	Trimestrielle
Nombre d'articles de veille partagés	> 10/mois	Mensuelle
Temps de retard sur version Symfony LTS	< 6 mois	Trimestrielle
Score de sécurité Composer	0 vulnérabilité	Hebdomadaire

16.2 Rapports de veille

Rapport mensuel à produire :

```
# Rapport de veille - [MOIS ANNEE]

## Résumé exécutif
[2-3 phrases sur les points clés du mois]

## Chiffres clés
- Alertes traitées : X
- Mises à jour effectuées : X
- Temps moyen de traitement : Xh
- Vulnérabilités corrigées : X

## Actions majeures
1. [Action 1]
2. [Action 2]

## Tendances observées
[Analyse des tendances technologiques]

## Recommandations
[Suggestions pour le mois suivant]
```

17. Plan d'action annuel

17.1 Planification 2026

Q1 2026 (Janvier - Mars)

- Mise à jour vers Symfony 7.4 (si disponible)
- Audit de sécurité complet
- Formation équipe sur les nouveautés PHP 8.3

Q2 2026 (Avril - Juin)

- Évaluation migration PHP 8.4
- Mise en place monitoring avancé (APM)
- Optimisation performances base de données

Q3 2026 (Juillet - Septembre)

- Tests de charge et optimisation
- Refactoring code legacy
- Documentation technique complète

Q4 2026 (Octobre - Décembre)

- Préparation migration Symfony 8.x (si annoncé)
- Bilan annuel de la veille
- Planification 2027

17.2 Projets d'amélioration

Court terme (< 6 mois)

- Automatisation complète des tests de sécurité
- Mise en place de GitHub Dependabot
- Script automatisé de rapport de veille

Moyen terme (6-12 mois)

- Migration vers infrastructure conteneurisée (Docker)
- Mise en place d'un pipeline CI/CD complet
- Monitoring et observabilité avancés (Sentry, New Relic)

Long terme (> 12 mois)

- Architecture microservices (si pertinent)

- Mise en place de Kubernetes
- Infrastructure as Code complète (Terraform)

18. Gestion des incidents de sécurité

18.1 Procédure d'urgence

En cas de vulnérabilité critique détectée :

1. **Alerte immédiate** (< 15 min)
    - Notification de l'équipe technique
    - Évaluation rapide de l'exposition
2. **Mesures conservatoires** (< 1h)
    - Mise en place de WAF rules si applicable
    - Restriction d'accès temporaire si nécessaire
    - Monitoring renforcé des logs
3. **Correction** (< 24h)
    - Application du patch de sécurité
    - Tests en staging
    - Déploiement en production
4. **Vérification** (< 48h)
    - Scan de sécurité post-correction
    - Vérification d'absence d'exploitation
    - Documentation de l'incident
5. **Communication** (< 72h)
    - Rapport interne
    - Communication clients si nécessaire
    - Post-mortem

18.2 Contacts d'urgence sécurité

Rôle	Contact	Disponibilité
RSSI	[NOM - TEL - EMAIL]	24/7
Responsable technique	[NOM - TEL - EMAIL]	24/7
Équipe sécurité hébergeur	[CONTACT]	24/7
CERT France	cert-fr.cossi@ssi.gouv.fr	Bureau

19. Ressources complémentaires

19.1 Livres recommandés

- "Symfony 7: The Fast Track" - Fabien Potencier
- "Modern PHP" - Josh Lockhart
- "The DevOps Handbook" - Gene Kim
- "Web Application Security" - Andrew Hoffman

19.2 Podcasts

- "Voices of the ElePHPant"
- "DevOps and Docker Talk"
- "The Changelog"
- "Security Now"

19.3 Conférences à suivre

- SymfonyCon (annuelle)
- PHP UK Conference
- DevOps World
- Black Hat / DEF CON (sécurité)

20. Révision et amélioration continue

20.1 Révision du processus de veille

Fréquence : Semestrielle

Points à évaluer :

- Efficacité des sources de veille
- Pertinence des informations collectées
- Temps consacré vs valeur apportée
- Taux de mise en œuvre des recommandations

20.2 Amélioration du processus

Questions à se poser :

- Les sources sont-elles toujours pertinentes ?
- Y a-t-il des redondances dans la veille ?
- Les outils utilisés sont-ils efficaces ?
- La charge de travail est-elle raisonnable ?
- Les informations sont-elles bien partagées ?

20.3 Historique des révisions

Version	Date	Auteur	Modifications
1.0	2025-10-17	[NOM]	Création initiale

21. Contacts et responsabilités

Responsable de la veille technologique

- Nom : [NOM]
- Email : [EMAIL]
- Téléphone : [TEL]

Référents par domaine :

- Symfony : [NOM]
- PHP : [NOM]
- Sécurité : [NOM]
- DevOps : [NOM]
- Base de données : [NOM]

Escalade en cas de problème :

1. Référent technique du domaine
2. Responsable de la veille
3. Responsable technique / CTO
4. Direction

Prochaine révision prévue : Avril 2026