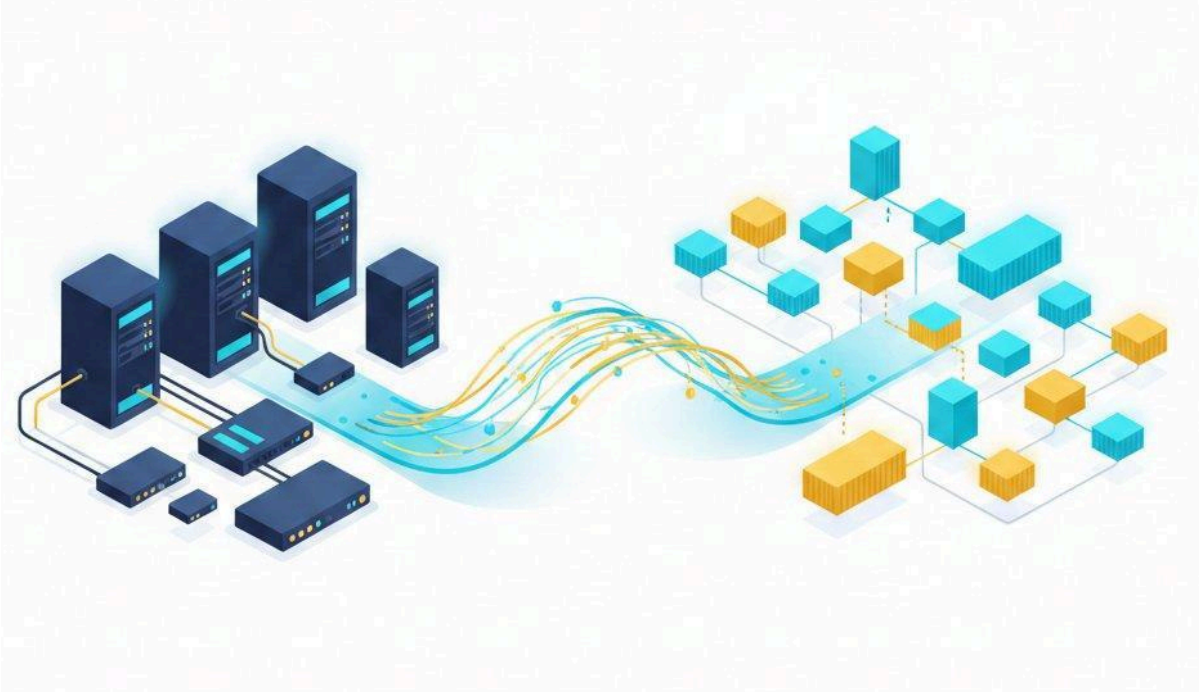
Sensor Factory

LIVRE BLANC

De la supervision à l'observabilité

Construire une stratégie d'observabilité pragmatique
pour les organisations en transformation



matthieu.noirbusson@sensorfactory.eu

Février 2026

Ce document a pour vocation d'accompagner la réflexion stratégique des équipes d'exploitation et des DSI dans leur transition vers l'observabilité. Il présente une vision pragmatique, indépendante de tout éditeur, fondée sur notre expérience de terrain

Résumé exécutif

Les systèmes d'information se transforment : infrastructures hybrides, conteneurisation, microservices, multi-cloud. Les outils de supervision traditionnels, indispensables pour les couches basses, ne suffisent plus à garantir la visibilité sur les couches applicatives. L'observabilité, fondée sur trois piliers : métriques, logs et traces, apporte les réponses manquantes. Mais cette transition soulève des défis concrets que ce document analyse en profondeur.

Au-delà de la visibilité, l'observabilité pose les fondations de l'automatisation, de la détection prédictive et de l'optimisation continue du SI.

Architecture	Organisation	Volumétrie	Collecte	Corrélation	Progressivité
Deux pôles complémentaires, standards ouverts	Qui fait quoi, self-service, gouvernance	Compression, downsampling, rétention adaptée	Agent unifié, gestion de flotte, cohabitation	Cross-linking natif métriques / logs / traces	Métriques → Logs → Traces, valeur à chaque étape

Aucun outil ne couvre tout. Le marché reste fragmenté. Les outils de supervision excellent sur l'infrastructure mais peinent sur l'applicatif. Les plateformes d'observabilité font l'inverse. Les solutions « tout-en-un » SaaS posent des questions de coût, de souveraineté et de verrouillage éditeur.

Notre recommandation : une approche bicéphale. Deux pôles complémentaires, supervision d'infrastructure et observabilité applicative, interconnectés par des standards ouverts (OpenTelemetry) et fédérés par une couche de visualisation commune (Grafana).

La volumétrie est un enjeu stratégique. L'observabilité génère des volumes considérables, de quelques centaines de gigaoctets à plusieurs téraoctets par mois. Le choix du backend de stockage conditionne la viabilité économique du projet.

La collecte ne s'improvise pas. Déployer et maintenir une flotte d'agents à l'échelle d'un parc est un projet opérationnel à part entière. La convergence vers un agent unifié basé sur OpenTelemetry réduit cette complexité.

L'organisation des équipes conditionne l'adoption. Administrateurs réseau, DevOps, développeurs, DSI : chaque profil a des besoins et des formats distincts. La gouvernance, le self-service et la conduite du changement doivent être anticipés.

La corrélation est la vraie promesse. Collecter et stocker des données normalisées n'est qu'une première étape. La valeur se réalise quand métriques, logs et traces sont corrélés automatiquement.

Une transition progressive. Métriques d'abord, puis logs structurés, puis traces distribuées, pour démontrer la valeur à chaque étape et éviter l'effet tunnel.

Sommaire

1. Introduction : pourquoi ce document ?

Surveiller un système d'information aujourd'hui, c'est observer un organisme vivant : hybride, distribué, en mutation permanente. Chaque conteneur déployé, chaque API exposée, chaque cloud ajouté élargit le périmètre, sans que les exigences de disponibilité, elles, ne laissent la moindre marge.

Dans ce contexte, les outils de supervision traditionnels continuent de remplir leur rôle sur les couches basses du SI, mais ils ne suffisent plus à répondre aux enjeux posés par les couches applicatives modernes. Un nouveau paradigme s'est imposé : l'observabilité.

Ce document de cadrage a pour objectif d'aider les équipes d'exploitation, les architectes et les DSI à structurer leur réflexion autour de trois questions fondamentales :

- **Quels sont les enjeux réels** derrière la transition vers l'observabilité ?
- **Quelle stratégie adopter** face à un marché fragmenté où aucun outil ne couvre tout ?
- **Comment construire une feuille de route** pragmatique et progressive ?

2. Supervision, observabilité : de quoi parle-t-on ?

2.1 La supervision : surveiller ce que l'on connaît

La supervision consiste à collecter des indicateurs prédéfinis pour vérifier que les composants d'un système fonctionnent conformément aux attentes. Elle répond à une question simple : est-ce que ça marche ?

Son terrain de jeu naturel est la couche infrastructure, réseau, serveurs, stockage, hyperviseurs, bases de données, où les outils spécialisés excellent depuis des années, qu'ils opèrent à distance via SNMP ou WMI, ou s'appuient sur des agents dédiés.

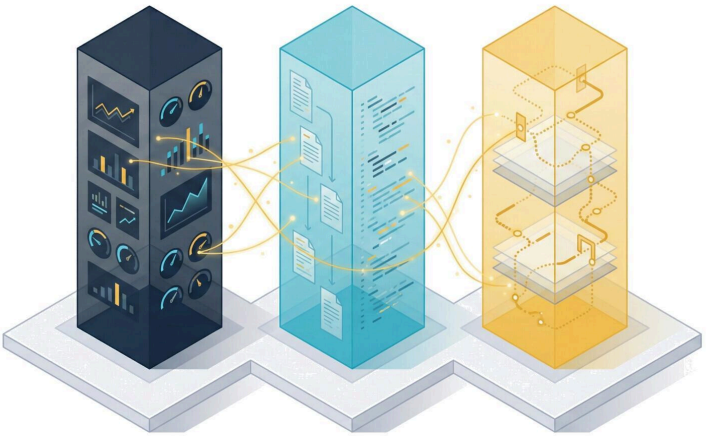
Un changement notable est en cours : l'intégration des logs. Longtemps laissés à des solutions séparées et souvent déconnectées, les journaux système et événements réseau rejoignent désormais les métriques dans une interface unifiée. C'est une avancée significative, elle permet de parler, pour la première fois, d'une forme d'observabilité d'infrastructure légère.

2.2 L'observabilité : comprendre ce que l'on ne connaît pas encore

L'observabilité va plus loin. Elle permet de comprendre l'état interne d'un système à partir de ses sorties externes. Elle répond à la question : « pourquoi est-ce que ça ne marche pas ? » et même « qu'est-ce qui va bientôt poser problème ? ».

L'observabilité repose sur trois piliers fondamentaux :

Pilier	Définition	Usage principal
Métriques	Mesures quantitatives collectées à intervalles réguliers (CPU, latence, taux d'erreur…)	Détecter les anomalies, déclencher des alertes, analyser les tendances
Logs	Enregistrements textuels horodatés des événements survenant dans un système	Contextualiser un incident, reconstituer la chronologie, auditer
Traces	Représentations du parcours d'une requête à travers un système distribué	Identifier les goulots d'étranglement, visualiser les dépendances



Les trois piliers de l'observabilité : métriques, logs et traces, reliés par la corrélation.

À retenir

La vraie puissance de l'observabilité réside dans la corrélation de ces trois types de données. Un pic de latence (métrique) peut être relié à une erreur dans un log, elle-même localisée grâce à une trace montrant quel microservice est en cause.

2.3 Synthèse comparative

Critère	Supervision	Observabilité
Question clé	« Est-ce que ça marche ? »	« Pourquoi ça ne marche pas ? »
Approche	Réactive : alertes sur seuils	Proactive : exploration et corrélation
Données	Métriques + logs d'infrastructure	Métriques + Logs + Traces corrélés
Périmètre	Réseau, système, stockage, hyperviseurs	Applications, microservices, API, Cloud
Utilisateurs	Admins système et réseau	DevOps, SRE, équipes applicatives
Outils	PRTG, Zabbix, Centreon, Nagios	VictoriaMetrics, Grafana Stack, Datadog

3. Le constat : aucun outil ne couvre tout

C'est la réalité du marché en 2025-2026 : malgré les promesses marketing, aucun outil unique ne couvre de manière satisfaisante l'ensemble du spectre, de la supervision réseau la plus basse à la traçabilité applicative la plus fine.

3.1 Les outils de supervision évoluent… mais restent limités

Les éditeurs d'outils d'infrastructure font des efforts considérables pour intégrer des capacités d'observabilité : meilleure gestion des logs, intégrations avec des sources de traces, tableaux de bord enrichis. Ces évolutions sont réelles et utiles.

Cependant, ces outils restent fondamentalement conçus pour la supervision d'infrastructure. Leur modèle de données, leur architecture et leurs agents sont optimisés pour interroger des équipements en SNMP, WMI, ou via des capteurs spécifiques. Ils ne sont pas nativement adaptés aux paradigmes de l'observabilité applicative (instrumentation OpenTelemetry, cardinalité élevée, traçage distribué).

3.2 Les plateformes d'observabilité ne remplacent pas la supervision

Inversement, les plateformes d'observabilité excellent dans le traitement des métriques applicatives, des logs structurés et des traces distribuées. Mais elles ne sont pas conçues pour remplacer un outil de supervision d'infrastructure :

- Découverte automatique d'équipements réseau (switches, routeurs, firewalls)
- Supervision SNMP fine avec des milliers de capteurs préconfigurés
- Cartographie et dépendances d'infrastructure physique
- Gestion des SLA et des rapports pour les équipes réseau

3.3 Le piège du « tout-en-un »

Certaines solutions SaaS (Datadog, New Relic, Dynatrace) promettent de tout couvrir. Si ces plateformes sont puissantes, elles posent d'autres questions :

- **Coût :** tarification basée sur le volume de données ingérées, potentiellement très élevée à l'échelle
- **Verrouillage éditeur :** dépendance à un écosystème propriétaire qui complique toute évolution
- **Souveraineté des données :** externalisation de toute la télémétrie vers un cloud tiers
- **Couverture réelle :** couverture insuffisante de la supervision bas niveau réseau et stockage on-premise

3.4 L'enjeu critique de la volumétrie

Lorsqu'on parle d'observabilité, on parle de volumes de données sans commune mesure avec la supervision traditionnelle. Un outil de supervision classique collecte quelques milliers de métriques à intervalles espacés. Une stack d'observabilité ingère en continu des millions de séries temporelles, des flux de logs de l'ensemble des composants applicatifs, et des traces couvrant chaque requête.

À l'échelle d'une infrastructure de taille intermédiaire, les volumes se comptent rapidement en centaines de gigaoctets. Pour les environnements plus conséquents, microservices, Kubernetes, multi-cloud, on atteint facilement le téraoctet, voire la dizaine de téraoctets sur des rétentions de quelques mois. Le choix d'un backend d'observabilité ne peut pas se faire uniquement sur les fonctionnalités : l'efficacité du stockage est un critère stratégique qui conditionne la viabilité économique du projet.

4. Notre vision : une approche bicéphale

Face à ce constat, nous défendons une approche « bicéphale » : deux pôles complémentaires, chacun optimisé pour son périmètre, interconnectés par des standards ouverts.

4.1 Schéma de principe

Deux pôles complémentaires reliés par une couche d'intégration unifiée.

STRATÉGIE D'OBSERVABILITÉ UNIFIÉE	
PÔLE INFRASTRUCTURE *« De la couche réseau au système »* • Réseau, Stockage, Hyperviseurs • Serveurs physiques et VM • Cloud (Azure, AWS) SNMP, WMI, API • PRTG, Zabbix, Centreon	**PÔLE OBSERVABILITÉ** *« Du système aux applications »* • Applications et microservices • Conteneurs et K8s • Expérience utilisateur OpenTelemetry (OTLP) • VictoriaMetrics, Grafana
INTÉGRATION : Grafana (dashboards unifiés) | Alerting centralisé | Corrélation cross-layer	

4.2 Pourquoi cette approche ?

- **Chaque outil excelle dans son domaine :** on capitalise sur les forces de chacun
- **Transition progressive :** on conserve l'existant qui fonctionne et on ajoute la brique d'observabilité
- **Standards ouverts :** OpenTelemetry comme langage commun, pas de verrouillage
- **Visibilité unifiée :** Grafana fédère les données des deux pôles
- **Maîtrise des coûts :** contrôle total, sans surprise liée aux volumes

4.3 La zone de recouvrement et la convergence des logs

Les deux pôles ne sont pas hermétiques. Il existe une zone de recouvrement légitime autour de la couche système. Cette redondance garantit la continuité de la visibilité.

La convergence la plus significative se joue sur les logs. Historiquement, la gestion des logs d'infrastructure était complexe : les outils de supervision ne les intégraient pas, et les solutions de log management (syslog, ELK) fonctionnaient en silo. Aujourd'hui, les deux pôles savent traiter les logs, créant une passerelle naturelle entre supervision et observabilité.

5. Qui fait quoi ? L'organisation des équipes

5.1 Des besoins très différents selon les profils

Derrière la question des outils se cache une question plus fondamentale : qui consomme quelle information, sous quelle forme, et pour quel usage ?

Chaque profil accède à ses propres vues et données, adaptées à son métier.

Profil	Périmètre	Informations clés	Format attendu
Admin réseau	Switches, routeurs, firewalls, WAN	Disponibilité, bande passante, latence, logs équipements	Cartographies, dashboards temps réel, alertes SNMP
Admin système	Serveurs, VM, hyperviseurs, stockage	CPU, mémoire, disque, processus, logs système	Dashboards infra, alertes seuils, rapports capacité
DBA	Bases de données, réplication	Connexions, temps de requête, verrous, slow queries	Dashboards spécialisés, alertes performance
DevOps / SRE	Apps, microservices, CI/CD, K8s	Latence applicative, taux d'erreur, traces, logs applicatifs	Exploration ad-hoc, corrélation 3 signaux

Développeur	Code applicatif, API, dépendances	Traces de ses services, logs de debug, perf endpoints	Traces détaillées, accès self-service
DSI	Vision transverse, SLA	Synthèse disponibilité, tendances, coûts	Rapports consolidés, vues métier
Équipe sécurité	Événements sécurité, conformité	Logs authentification, flux anormaux, alertes sécurité	SIEM, corrélation d'événements

5.2 Ce que ce tableau révèle

La séparation infra/applicatif se retrouve dans les équipes. Les administrateurs réseau et système travaillent dans le monde de la supervision. Les DevOps, SRE et développeurs travaillent dans le monde de l'observabilité. L'approche bicéphale reflète une réalité organisationnelle.

Les formats attendus diffèrent radicalement. Un admin réseau a besoin de cartographies et d'alertes SNMP. Un DevOps a besoin d'explorer des traces. Un DSI a besoin d'un rapport de synthèse. Forcer tous ces profils dans le même outil, c'est ne satisfaire personne.

Le self-service est un enjeu clé. Les équipes applicatives ont besoin d'un accès en libre-service avec la capacité de créer leurs propres dashboards et alertes. C'est un changement culturel majeur.

La vision transverse est le rôle de la couche d'intégration. Grafana fédère les vues : chaque profil accède à ses dashboards, le responsable d'exploitation dispose d'une vue consolidée.

5.3 Implications pour le projet

- Identifier les populations cibles dès le départ et leurs besoins réels
- Définir les périmètres de responsabilité : qui configure les alertes, crée les dashboards, gère les agents ?
- Prévoir la gestion des droits d'accès (par équipe, par namespace, par source)
- Planifier la conduite du changement et la formation

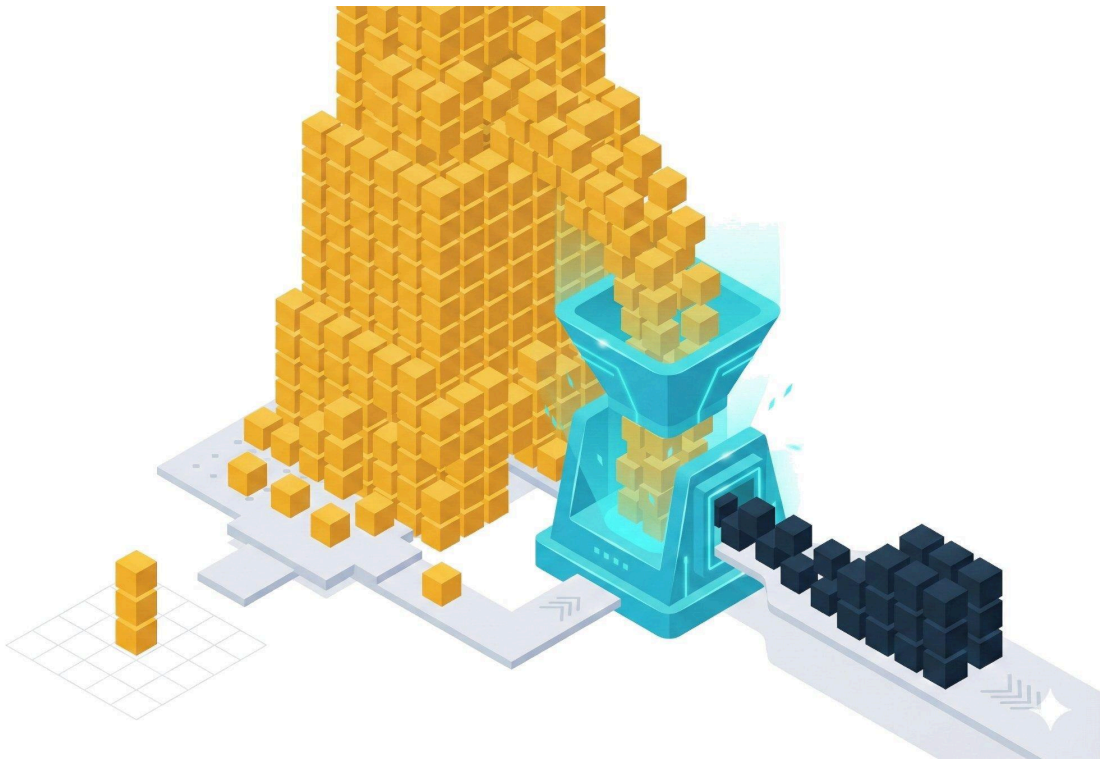
6. Le défi de la volumétrie et du stockage

6.1 Des volumes qui changent d'échelle

Passer de la supervision à l'observabilité, c'est passer d'un flux de données maîtrisé à un torrent :

Périmètre	Métriques	Logs	Traces	Stockage / mois
PME (50 serveurs)	~10 000 séries	~10 Go/jour	Minimes	50 à 150 Go
ETI (500 serveurs)	~500 000 séries	~100 Go/jour	Significatives	500 Go à 2 To
Grand compte (2000+)	Millions de séries	500 Go–1 To/j	Massives	5 à 20+ To

Ces volumes croissent naturellement avec le SI. Une stratégie qui ne prend pas en compte cette volumétrie dès le départ est vouée à l'échec économique.



Croissance des volumes et mécanisme de compression : l'efficacité du stockage est un enjeu stratégique.

6.2 Les caractéristiques d'un stockage adapté

Compression avancée. Les meilleurs moteurs atteignent des ratios de 7x à 10x. Sur 500 Go/jour d'ingestion, cela signifie 50 à 70 Go/jour de stockage réel, un facteur décisif sur le coût annuel.

Downsampling natif. Haute résolution sur fenêtre courte, agrégation automatique pour le stockage longue durée. Granularité maximale pour l'investigation, coût maîtrisé sur l'historique.

Ingestion performante sous pression. Lors d'un incident majeur, le backend doit absorber les pics sans dégradation ni perte.

Efficacité des requêtes sur grands volumes. Interroger des mois d'historique en quelques secondes change fondamentalement l'expérience d'investigation.

Politiques de rétention granulaires. Longue durée pour métriques critiques, courte pour logs debug verbose.

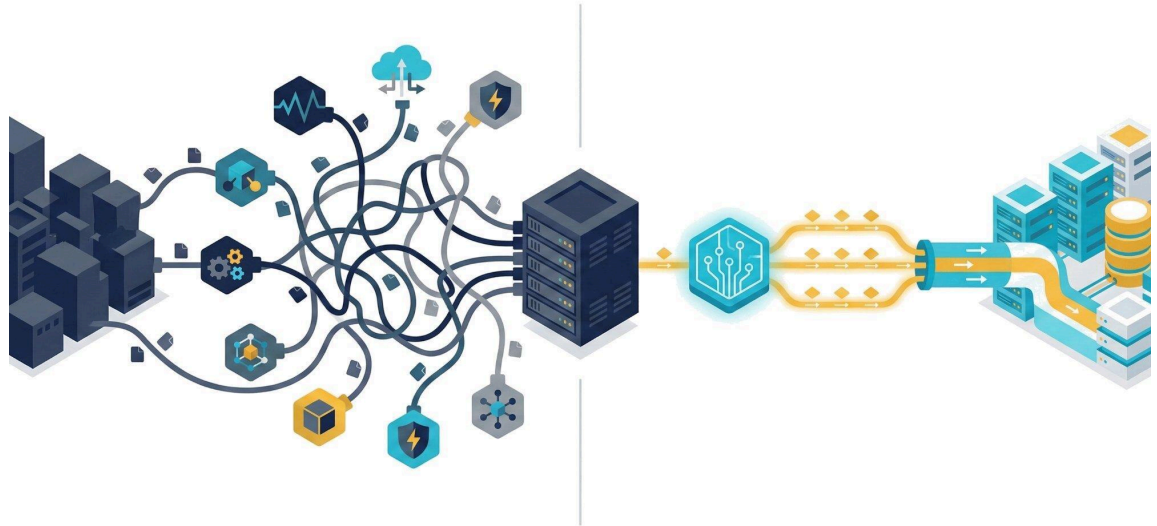
7. La collecte : des agents au cœur du dispositif

7.1 Le problème de la prolifération des agents

Chaque source de télémétrie nécessite un agent ou un mécanisme de collecte. Historiquement, cela signifiait multiplier les agents spécialisés : un exporteur pour les métriques, un agent de logs, un agent de traçage, sans compter les agents de supervision. Sur un parc de quelques centaines de serveurs, on se retrouvait avec trois à cinq agents distincts par machine.

7.2 Vers un agent unifié : OpenTelemetry Collector

L'OpenTelemetry Collector est l'incarnation la plus aboutie de l'agent unifié. Son architecture en pipeline (receivers → processors → exporters) collecte simultanément métriques, logs et traces, les transforme et les route vers un ou plusieurs backends, dans un seul binaire.



Avant : prolifération d'agents hétérogènes. Après : agent unifié et pipeline organisé.

AVANT : paysage hétérogène	APRÈS : architecture clarifiée
Supervision agentless (PRTG via SNMP/WMI)	Supervision agentless (inchangée)
Agent Zabbix	Centreon CMA (basé OTLP)

Agent Centreon (legacy) Exporteur Prometheus Agent Filebeat (logs) Agent Fluentd (logs) Agent Jaeger (traces) → **Jusqu'à 6-7 composants/machine**	**OpenTelemetry Collector** (métriques + logs + traces) → **2 à 3 composants, périmètres clairs**

7.3 La cohabitation avec la supervision existante

Le déploiement ne se fait pas sur un terrain vierge. PRTG fonctionne en mode agentless (SNMP, WMI, API REST), Zabbix et SolarWinds s'appuient sur des agents dédiés, et Centreon propose le Centreon Monitoring Agent (CMA), en GA depuis juillet 2025, basé sur le protocole OpenTelemetry (OTLP), illustrant la convergence du monde de la supervision vers les standards de l'observabilité.

- **Empreinte cumulée.** L'inventaire des composants existants et le calcul de l'empreinte cumulée doivent faire partie de l'audit initial.
- **Recouvrements de collecte.** Éviter le stockage en double inutile.
- **Convergence progressive.** Des éditeurs comme Centreon adoptent OTLP. Certains receivers OTel ingèrent des sources historiques.
- **Ports et flux réseau.** L'adoption d'OTLP impose de planifier les cohabitations.

7.4 La gestion de flotte à l'échelle

Déploiement automatisé (Ansible, Puppet, opérateur K8s), rolling updates avec retour arrière.

Configuration centralisée via OpAMP (Open Agent Management Protocol).

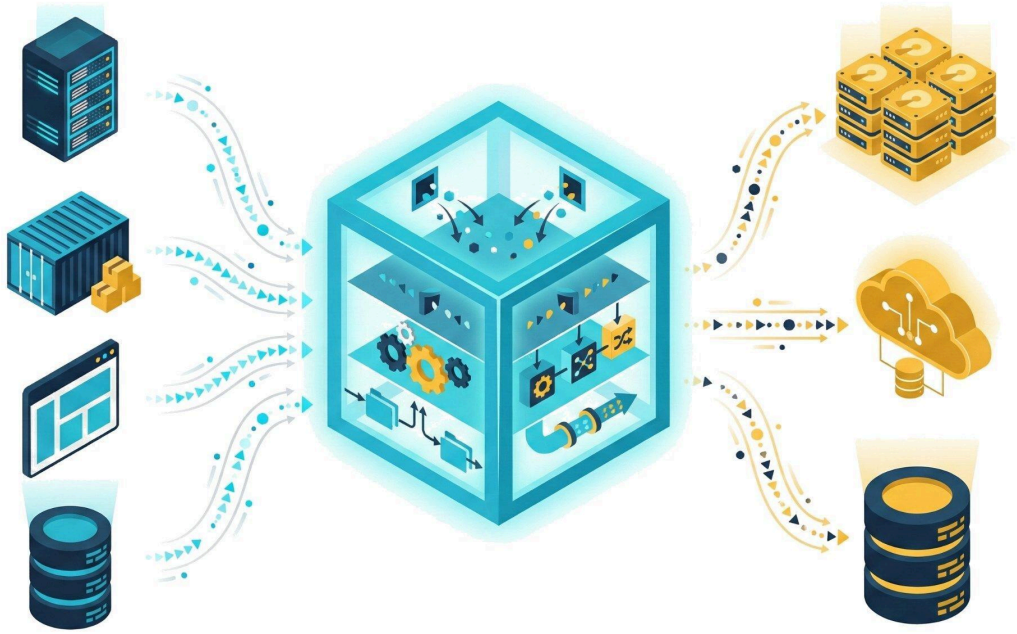
Empreinte minimale : quelques dizaines de Mo de RAM et une fraction de CPU.

Auto-découverte dans les environnements dynamiques (K8s, auto-scaling).

8. Focus : la stack d'observabilité open-source

8.1 OpenTelemetry : le standard fédérateur

OpenTelemetry (OTel) est le standard open-source de la CNCF pour la collecte, le traitement et l'export des données de télémétrie. Supporté par la quasi-totalité des éditeurs, il garantit une portabilité totale : en instrumentant avec OTel, vous pouvez changer de backend sans modifier le code.



L'architecture pipeline de l'OpenTelemetry Collector : receivers, processors, exporters.

8.2 VictoriaMetrics : la TSDB nouvelle génération

VictoriaMetrics est une base de séries temporelles open-source (Apache 2.0), conçue pour des performances élevées et une efficacité en coût remarquable. Son écosystème couvre les trois piliers :

Composant	Rôle	Points forts
VictoriaMetrics	Métriques	Compression 7x+, MetricsQL, support OTLP natif
VictoriaLogs	Logs	Multi-sources (Filebeat, OTel, Syslog), LogsQL, Grafana
VictoriaTraces	Traces	OTLP HTTP/gRPC, visualisation Grafana
VM Anomaly Detection	Détection anomalies	IA sur séries temporelles pour alerting proactif
vmagent	Collecte	Agent léger, plus performant que Prometheus Agent

Résultats documentés

Grammarly : coûts ÷ 10. Spotify : performances Grafana améliorées. Zomato : 2,2 milliards de séries actives. CERN : monitoring temps réel du détecteur CMS.

8.3 Grafana : la visualisation unifiée

Grafana fédère les données des deux pôles dans une interface unique. Ses datasources connectent VictoriaMetrics, VictoriaLogs, VictoriaTraces, mais aussi PRTG ou d'autres outils via des plugins dédiés.

8.4 De la normalisation à la corrélation : la vraie promesse

La normalisation via OpenTelemetry n'est qu'une première étape. La vraie promesse se réalise quand un opérateur peut, depuis une alerte sur une métrique, naviguer vers les logs correspondants puis remonter la trace, sans quitter son interface.

- Une métrique anormale reliée automatiquement aux logs du même service
- Un log d'erreur rattaché à la trace via un identifiant de corrélation (trace ID)
- Une trace enrichie des métriques du service concerné au moment de l'exécution

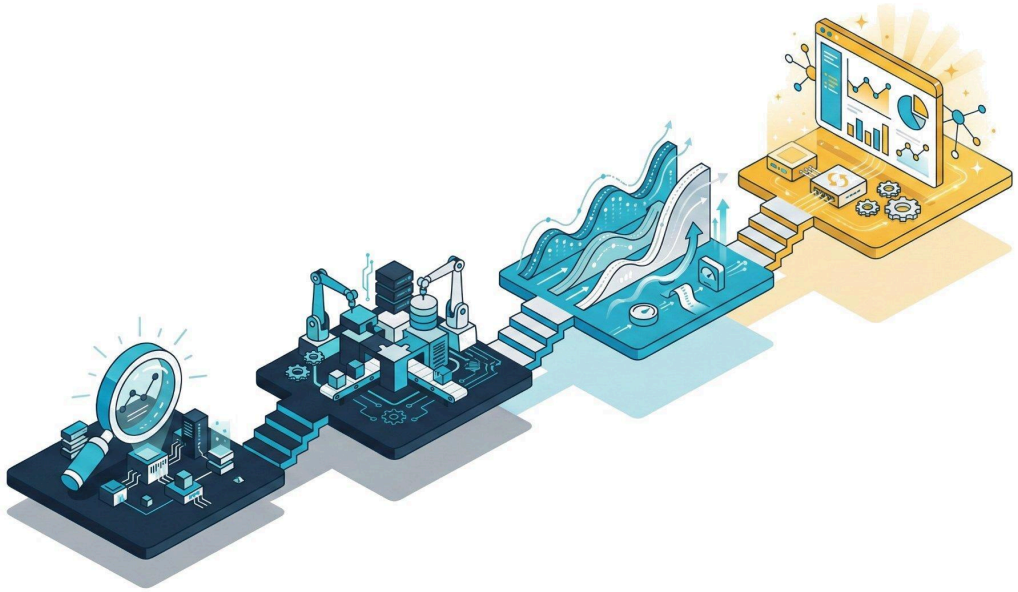
Sans ces capacités, les trois piliers restent trois silos. Le choix du backend doit intégrer ce critère.

8.5 Comparaison des approches

Critère	SaaS (Datadog...)	Grafana Stack	VM Stack
Modèle	Cloud propriétaire	Open-source assemblé	Open-source unifié
Coût	Élevé et variable	Infra uniquement	Infra, très optimisé
Complexité ops	Faible (managé)	Élevée (4+ composants)	Modérée
Souveraineté	Non (cloud tiers)	Oui	Oui
Vendor lock-in	Fort	Faible	Faible
Performance	Bonne	Variable	Excellente

9. Feuille de route suggérée

La transition ne se fait pas en un jour. Voici une feuille de route progressive en quatre phases.



Quatre phases progressives : audit, fondations, instrumentation, maturité.

Phase 1, Cadrage et audit

- Inventaire de l'existant : outils, agents, périmètres couverts, lacunes
- Cartographie des besoins par équipe
- Définition des objectifs et indicateurs de succès
- Arbitrage de la stratégie outillage

Phase 2, Fondations

- Consolidation du pôle supervision d'infrastructure
- Déploiement de la stack d'observabilité
- Mise en place de l'OpenTelemetry Collector
- Premiers dashboards Grafana fédérant les deux pôles

Phase 3, Instrumentation progressive

L'instrumentation se fait en trois vagues, chacune apportant de la valeur immédiate :

Vague 1 : Métriques	Vague 2 : Logs	Vague 3 : Traces
Effort : faible Endpoints Prometheus/OTLP **Valeur : dashboards, alerting**	Effort : modéré Format structuré + trace ID **Valeur : contexte, investigation**	Effort : significatif SDK OTel / auto-instrumentation **Valeur : cause racine**

- Mise en place de l'alerting unifié
- Formation des équipes et transfert de compétences

Phase 4, Optimisation et maturité

- Corrélation cross-layer (de l'infrastructure à l'application)
- Détection d'anomalies et alerting proactif
- Optimisation des coûts (rétention, downsampling, nettoyage)
- Amélioration continue des dashboards et de l'expérience utilisateur

10. Critères clés pour le choix d’une solution

Avant toute démonstration ou proof of concept, il est utile de clarifier ses propres exigences. Les critères suivants couvrent les dimensions techniques, organisationnelles et économiques d'un choix qui, mal anticipé, peut s'avérer coûteux à corriger.

Critère	Questions à se poser
Interopérabilité	Support natif OpenTelemetry ? Intégration avec l’existant sans tout remplacer ?
Efficacité stockage	Ratio de compression ? Downsampling natif ? Impact coût infra à 3 ans ?
Corrélation	Cross-linking natif métriques/logs/traces ? Navigation fluide entre signaux ?
Collecte / agents	Agent unifié ? Empreinte légère ? Gestion de flotte ? Auto-découverte ?
Organisation	Droits par équipe ? Self-service applicatif ? Vues métier ?
Coût total (TCO)	Coût réel à 3 ans ? Évolution linéaire avec le volume ?
Scalabilité	Millions de séries temporelles ? Absorption des pics ?
Souveraineté	Données sous contrôle ? Déploiement on-premise possible ?
Pérennité	Éditeur indépendant ? Modèle économique viable et transparent ?

11. Conclusion : les points critiques et les horizons

Au terme de cette analyse, un constat s'impose : la transition vers l'observabilité n'est ni un simple changement d'outil, ni un projet purement technique. C'est une transformation qui touche à l'architecture du SI, à l'organisation des équipes et à la culture opérationnelle de l'entreprise.

Deux chemins, une destination commune

Plutôt que de chercher l'outil unique qui n'existe pas, l'approche bicéphale que nous défendons reconnaît que les chemins vers la visibilité complète du SI diffèrent selon le point de départ.

Du côté de l'infrastructure, le chemin est celui de la consolidation. La plupart des organisations disposent déjà d'outils de supervision qui assurent la surveillance des couches basses. Ces outils ont fait leurs preuves. Le chemin consiste à les renforcer, à combler leurs angles morts, notamment sur les logs d'infrastructure, et à les ouvrir vers les standards modernes.

Du côté applicatif, le chemin est celui de la construction. Peu d'organisations disposent aujourd'hui d'une véritable stack d'observabilité. Il faut choisir un backend adapté aux volumes, déployer un agent unifié, instrumenter progressivement les applications critiques. C'est un chantier nouveau, avec ses propres compétences et ses propres rythmes.

L'enjeu stratégique réside à la jonction de ces deux chemins. C'est dans la zone de recouvrement, la couche système, les logs, les premiers signaux applicatifs, que les deux pôles se rejoignent. OpenTelemetry fournit le langage commun, Grafana offre l'interface fédératrice. Sans cette articulation, on ne fait que substituer deux silos à un seul.

Les points critiques de la mise en œuvre

Cette convergence se heurte à quatre défis concrets.

La volumétrie, d'abord, le point le plus sous-estimé. L'observabilité génère des volumes sans commune mesure avec la supervision traditionnelle : des centaines de gigaoctets à

plusieurs téraoctets par mois. Un backend inadapté transforme un investissement stratégique en gouffre financier.

La collecte, ensuite. Sur le pôle infrastructure, les mécanismes sont en place. Sur le pôle observabilité, tout reste à construire : déploiement d'agents OpenTelemetry, cohabitation avec l'existant, gestion de flotte à l'échelle. Des signaux encourageants existent, comme l'adoption d'OTLP par Centreon, mais la convergence ne s'improvise pas.

La corrélation, surtout, la vraie ligne de partage entre une accumulation de données et une capacité d'investigation réelle. La valeur se réalise quand un opérateur peut, depuis une alerte infrastructure, naviguer vers les logs applicatifs puis remonter la trace jusqu'à la cause racine. Sans cette corrélation cross-layer, les trois piliers restent trois silos.

Le facteur humain, enfin. Le pôle infrastructure est opéré par des administrateurs réseau et système ; le pôle observabilité sera porté par des DevOps et des développeurs, avec une culture très différente. Gouvernance, self-service, conduite du changement : ces questions organisationnelles sont aussi déterminantes que le choix technique.

La progressivité comme principe directeur

Quel que soit le point de départ, la progressivité est la clé. Sur le pôle infrastructure : consolider, puis ouvrir vers les standards. Sur le pôle observabilité : métriques d'abord, logs structurés ensuite, traces distribuées enfin. Chaque vague apporte une valeur immédiate, justifie la suivante, et rapproche les deux pôles d'une vision véritablement unifiée.

Au-delà de l'observabilité : les horizons

Une stack d'observabilité structurée pose les fondations de transformations plus profondes. L'automatisation de la réponse aux incidents, redémarrage, scaling préventif, tickets enrichis, devient possible dès que les signaux sont fiables et corrélés. L'AIOps permet de détecter une saturation trois jours avant l'incident ou de repérer une dégradation invisible à l'œil humain. L'Infrastructure as Code s'appuie sur les données de télémétrie pour piloter une infrastructure qui s'auto-régule. Et la dimension FinOps, corrélation performance/coûts, fournit les données factuelles indispensables à une gestion financière éclairée du SI.

Les horizons de l'observabilité : détection IA, automatisation, IaC, FinOps.

Sensor Factory vous accompagne

Partenaire officiel VictoriaMetrics, spécialiste intégrateur PRTG, praticiens de Grafana, Zabbix, Centreon et OpenTelemetry. Nous intervenons à chaque étape : audit, cadrage, déploiement, organisation, formation, exploitation.

N'hésitez pas à nous contacter pour échanger sur votre contexte.

12. Ressources complémentaires

OpenTelemetry

Documentation officielle, opentelemetry.io/docs

Découvrir OpenTelemetry, opentelemetry.io/docs/what-is-opentelemetry

Guide du Collector, opentelemetry.io/docs/collector

Application de démo OTel, opentelemetry.io/docs/demo

OpAMP, gestion d'agents à distance, opentelemetry.io/docs/specs/opamp

Awesome OpenTelemetry, github.com/magsther/awesome-opentelemetry

Formation gratuite Getting Started, opentelemetry.io/docs/getting-started

VictoriaMetrics

Documentation officielle, docs.victoriametrics.com

Guide OTel + VictoriaMetrics, docs.victoriametrics.com/guides/getting-started-with-opentelemetry

Best practices, docs.victoriametrics.com/BestPractices.html

Études de cas, victoriametrics.com/case-studies

Dashboards Grafana et règles d'alerte, victoriametrics.com/resources

Blog technique, victoriametrics.com/blog

Playground en ligne, play.victoriametrics.com

Grafana

Documentation officielle, grafana.com/docs

Grafana + OpenTelemetry, grafana.com/docs/opentelemetry

Tutoriels et Learning Paths, grafana.com/tutorials

CNCF

Livre blanc Observabilité, github.com/cncf/tag-observability/blob/main/whitepaper.md

CNCF Landscape, landscape.cncf.io

Observability Trends 2025, cncf.io/blog/2025/03/05/observability-trends-in-2025

Tutoriels en français

Xavki, VictoriaMetrics, xavki.blog/victoriametrics-tutoriels-pour-debuter-francais

Xavki, Grafana, xavki.blog/grafana-tutoriels-francais

Cocadmin, Monitoring, cocadmin.com/category/monitoring

Communauté et événements

PromCon, promcon.io

ObservabilityCON, grafana.com/events/observabilitycon

KubeCon + CloudNativeCon, events.linuxfoundation.org/kubecon-cloudnativecon-europe