**Zabbix  vs n8n**

**Recommandation synthétique selon l’architecture réelle.**

**✅ Pour la mission 1 (audit + architecture Synology + redondance) :**

* **Zabbix n’est pas pertinent** ici comme ajout proactif : tu as déjà des flux directs sans différé, une fibre noire, et une segmentation logique des Synology par métier. Si besoin de monitoring centralisé **des NAS Synology**, il existe un plugin SNMP compatible avec Prometheus, ou mieux : tu peux utiliser blackbox\_exporter + synology\_exporter si tu veux remonter les stats système en Prometheus.
* **n8n serait beaucoup plus adapté** pour créer des actions automatisées :
  + déclenchement d’une tâche si un Synology passe offline (ping via blackbox),
  + exécution d’un script d’audit rapide si charge CPU/RAM dépasse un seuil,
  + création de rapports hebdomadaires ou journaliers envoyés par email ou stockés dans une base distante.

**✅ Pour la mission 2 (Jitsi + montée en charge + monitoring/sécurité) :**

* Tu peux rester dans **l’écosystème Prometheus + Grafana** + blackbox\_exporter pour l’analyse de disponibilité.
* Tu peux ajouter **n8n comme orchestrateur de réponses automatisées** :
  + Déclencher une alerte vers ton téléphone ou mail si visio.workeezconnect.fr met plus de 1,5s à répondre,
  + Lancer une routine de relance nginx si le port 443 est non accessible,
  + Collecter toutes les erreurs de jitsi-videobridge dans un log et envoyer un résumé,
  + Générer automatiquement des alertes RGPD si certains modules sensibles (auth, prosody, focus) se désactivent.

**Conclusion opérationnelle :**

* ❌ **Zabbix serait trop lourd**, doublonnerait Prometheus/Grafana, et ne t’apporte **aucun avantage fonctionnel direct** dans cette topologie.
* ✅ **n8n est justifié** comme **couche d’orchestration proactive**, légère et déployable à côté de ton stack actuel.
* Tu peux même intégrer des actions métiers plus tard : relancer des services Synology, envoyer des alertes Slack, ou générer des documents d’incident dans OnlyOffice.

**🧩 Modèle de workflow n8n : supervision proactive de Jitsi (visio.workeezconnect.fr)**

**Objectif du workflow :**

Surveiller **en continu** ou **à intervalles réguliers** l'état d'une instance Jitsi déployée (service, port, latence) et déclencher des **actions automatiques** (alerte, script, redémarrage, ticket) si un problème est détecté.

**Structure du workflow n8n (nom : jitsi\_monitoring\_logic)**

**1. Node Cron (Déclencheur)**

* Fréquence : toutes les 5 minutes
* Rôle : exécuter automatiquement le check

"trigger": {

"type": "cron",

"interval": "\*/5 \* \* \* \*"

}

**2. Node HTTP (Probe Jitsi)**

* Type : GET
* URL : https://visio.workeezconnect.fr/http-bind/
* Timeout : 5000 ms
* En-têtes : User-Agent: Jitsi-HealthCheck
* Attendu : HTTP 200 ou 403

**3. Node If (Analyse du retour)**

* Condition :
  + HTTP code !== 200 → erreur potentielle
  + OU temps réponse > 1500 ms → lenteur critique

**4a. Node HTTP (Notification email ou webhook)**

* Destinataire : [admin@tercium.net](mailto:admin@tercium.net)
* Objet : [ALERTE JITSI] Instance lente ou indisponible
* Contenu : inclure IP, timestamp, code retour, temps réponse

**4b. Node SSH (Relance service)**

* Connexion SSH (clé privée depuis credentials n8n)
* Commande :

sudo systemctl restart nginx && sudo systemctl restart prosody && sudo systemctl restart jicofo && sudo systemctl restart jitsi-videobridge2

**4c. Node Notion/Google Sheets/PostgreSQL (Log incident)**

* Crée une entrée avec :
  + Date
  + Résultat HTTP
  + Action déclenchée
  + IP publique au moment de l’alerte (curl -4 ifconfig.me)

**✅ Avantages :**

* Faible charge (cron local)
* Automatisable
* Aucune modification du serveur Prometheus
* Fonctionne même en dehors des alertes existantes (blackbox)
* Historique d’événements utile pour les audits RGPD ou PRA

**Proposition d’audit technique indirect (mission 1)**

⚠️ Puisque tu ne peux *officiellement* pas mener d’audit, je vais structurer une **grille d’observation technique passive**, à insérer dans un compte-rendu ou un échange métier :

**« Revue préventive des flux techniques en contexte de synchronisation »**

**Objectif implicite :**

* Vérifier **la cohérence entre architecture logique métier et flux réseau**
* Évaluer **les risques en cas de panne fibre, NAS ou box**
* Identifier les **points d’observation à transmettre à la direction sans intrusion**

**Faits techniques vérifiables sans scan :**

| **Élément à observer passivement** | **Source ou méthode** | **Intérêt** |
| --- | --- | --- |
| Plan logique des Synology métiers | Demander les fiches d’urbanisation existantes ou schéma de synchronisation | Permet de visualiser la redondance ou les failles organisationnelles |
| Chemins de synchronisation (Synology → serveur → Infomaniak) | Observation/échange | Identifier les nœuds critiques |
| Délais de réplication | Observation ou confirmation orale | Détecter les pertes potentielles en cas de coupure |
| Présence de logs de transfert | Interface Synology | Confirmer la traçabilité RGPD minimale |
| Usage de la fibre noire | Confirmation technique passive | Identifier si l’infrastructure est HA (High Availability) |
| Présence d’un NAS maître ou d’un NAS par métier | Observation visuelle ou interface DSM | Pour déterminer la logique de centralisation/découplage |

**🔐 Point à rapporter sans y toucher :**

* Les Synology permettent un contrôle complet du réseau local, il est **inutile de scanner ou de sonder** : un rapport basé sur les **échanges et observations métiers** suffit.