PROJET FOXTROT

Plateforme de suivi des consommations Cloud

Pôle SISR

Infrastructure & Sécurité

Jâsir & Adrien

Pôle SLAM

Développement & Data

Raphaël & Ilyes

5 MARS 2026

Contexte et Enjeux

La Problématique

Le campus IT "DCS Games" devait moderniser son infrastructure pour fournir un suivi précis et sécurisé de la consommation des ressources (Stockage, Réseau, CPU) de ses applications internes.

Sécurité Maximale

Haute Disponibilité

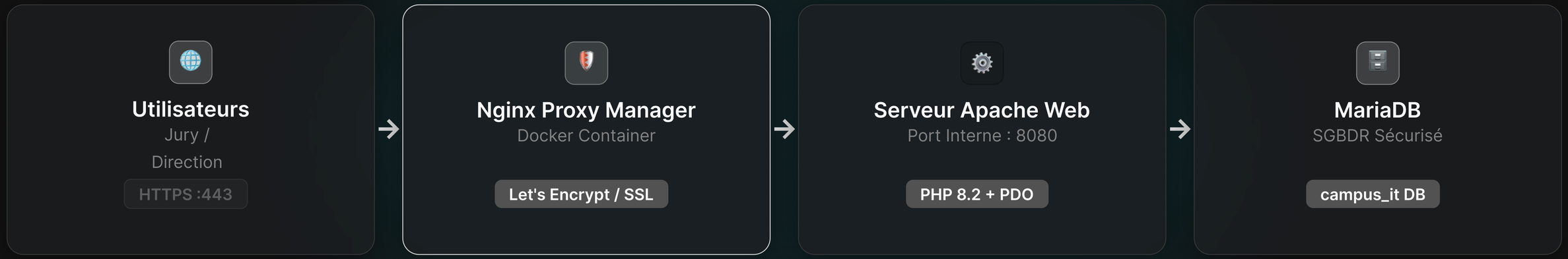
Data Visualisation

Nos Objectifs Communs

- **Déployer un environnement de production** stable sous Linux (Debian 12).
- **Sécuriser les accès** (Web et distant) contre les attaques.
- **Concevoir une base de données** relationnelle robuste.
- **Développer un Dashboard dynamique** avec des indicateurs clés (KPIs).
- **Assurer une synergie** parfaite entre les pôles SISR et SLAM.

VISION GLOBALE

Architecture de la Solution



Sécurité Périphérique (SISR)

UFW Firewall · Fail2Ban · WireGuard VPN · Clés SSH

Traitement Data (SLAM)

Requêtes SQL Avancées · Chart.js · UI/UX Glassmorphism

L'Infrastructure Serveur

Hyperviseur & Matériel

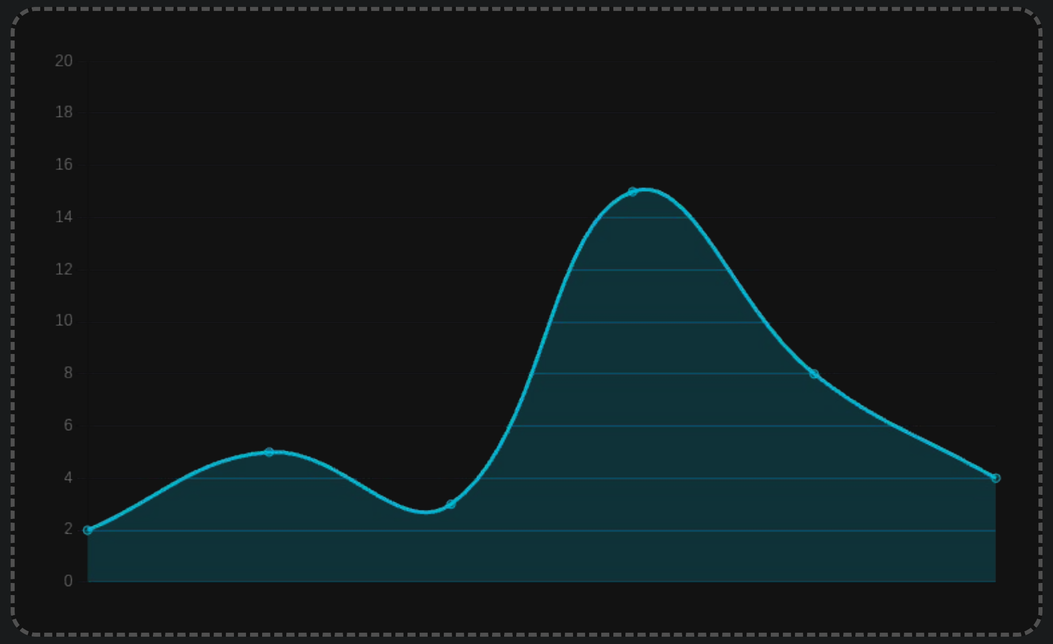
- **Matériel hôte :** Mini PC Intel N150 (TDP 6W), 16 Go RAM, SSD 512 Go + HDD Ext 2 To
- **Hyperviseur :** Proxmox VE (Architecture On-Premise)
- **Ressources VM allouées :** 2 vCPU, 4 Go RAM, 30 Go SSD NVMe (q35)

Système d'Exploitation

Choix délibéré de **Debian 12 (Bookworm)** en version "Netinst" (sans interface graphique) pour :

- Optimisation des performances (faible consommation RAM/CPU)
- Stabilité reconnue en environnement de production
- Réduction de la surface d'attaque

Monitoring & Métriques (Proxmox)



CPU Movenne
4.2%

RAM Utilisée
845 Mo

Sécurité Réseau Active

Pare-feu UFW

Mise en place d'une politique de sécurité stricte : tout fermer par défaut, n'ouvrir que l'essentiel.

```
# Politique par défaut
ufw default deny incoming
ufw default allow outgoing

# Ports autorisés
ufw allow 51820/udp # Wireguard
ufw allow 80/tcp    # HTTP (NPM)
ufw allow 443/tcp   # HTTPS (NPM)
```

Fail2Ban

Protection dynamique contre les attaques par force brute (Brute-force) sur les services exposés.

- Analyse continue des logs systèmes en temps réel
- Bannissement automatique des adresses IP suspectes
- Intégré au pare-feu iptables/UFW

Durcissement SSH

Le port 22 n'est pas exposé sur Internet. L'accès root est totalement banni.

```
# /etc/ssh/sshd_config
PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
```

Accès via Clés Ed25519 uniquement

Tunnels Privés et Travail d'Équipe

Pourquoi WireGuard ?

Pour permettre à l'équipe SLAM de déployer son code en toute sécurité, sans exposer le port SSH et SFTP sur Internet, nous avons configuré un VPN **WireGuard**.

- **Cryptage moderne :** Très performant et ultra-sécurisé.
- **Invisibilité :** Le port 51820 UDP est "furtif" de l'extérieur.
- **Adresse IP privées :** Le pôle SLAM accède au serveur via une IP locale (ex: 10.10.10.1).

Flux de déploiement sécurisé

1. Connexion VPN
Le dev active son client WireGuard et rejoint le réseau virtuel.

2. Transfert SFTP
Transfert des fichiers PHP/CSS dans `/var/www/html` avec le compte utilisateur restreint.

3. Base de Données
Connexion au SGBD local via le couple application / compte de service.

Reverse Proxy et Chiffrement SSL

Architecture Web

Pour répondre à l'exigence du HTTPS, nous avons découplé le serveur web du gestionnaire de certificats.

- **Apache (Backend) :** Configuré sur le port interne 8080. Il exécute uniquement le code PHP du Dashboard.
- **Nginx Proxy Manager / NPM (Frontend) :** Déployé via Docker. Il écoute publiquement sur le port 80/443.
- **Nom de Domaine :** Redirection DNS de dcsgames.jasir.fr gérée chez Infomaniak.

Résolution HTTP(S)

```
# docker-compose.yml de NPM
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    ports:
      - '80:80'
      - '443:443'
      - '81:81' # Admin UI
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
```

Génération automatisée des certificats Let's Encrypt + Force SSL

Base de données & Sauvegardes

Principe de Moindre Privilège

L'application web (PHP) ne se connecte **jamais** en tant que root à MariaDB.

```
CREATE DATABASE campus_it;
CREATE USER 'app_campus'@'localhost'
  IDENTIFIED BY '********';
GRANT ALL PRIVILEGES ON campus_it.*
  TO 'app_campus'@'localhost';
FLUSH PRIVILEGES;
```

Politique de Sauvegarde

Mise en place d'une double stratégie de backup (OS + Data).

1. Backup OS (Proxmox) : Snapshots Hebdomadaires

2. Backup BDD (CRON) :

```
# mysqldump tous les jours à 23h00
0 23 * * * mysqldump -u root -p*** campus_it >
  /var/backups/mysql/campus_it_$(date +\%F).sql
```

Analyse Comparative Financière (On-Premise vs Cloud)

Réponse au cahier des charges : Le choix de l'infrastructure impacte l'investissement initial (CAPEX) et les coûts de fonctionnement réguliers (OPEX).

CRITÈRE D'ANALYSE	**SOLUTION ON-PREMISE (RETENUE)** MINI PC INTEL N150 (AUTO-HÉBERGEMENT)	**SOLUTION CLOUD VIRTUELLE** VPS / INSTANCE OVH OU AWS
Type de coût	CAPEX (Investissement matériel initial)	OPEX (Abonnement mensuel + Engagement)
Coût Matériel	~200€ (N150, 16Go RAM, NVMe 512Go + HDD 2To)	0 € (Inclus dans la location distante)
Frais Opérationnels	~0 € / mois (Box internet du foyer, conso très faible ~6W)	~10 € à 15 € / mois
Maintenance & Énergie	À la charge exclusive de notre équipe SISR	Gérée et garantie (SLA) par le fournisseur
Sécurité & Données	Maîtrise totale en interne	Données hébergées chez un tiers (Confiance requise)
VERDICT	**Solution Retenue (Amortissement ultra-rapide)**	**Rejetée (Coût récurrent sur le long terme)**

Conclusion : Notre Mini PC Intel N150 offre une solution d'auto-hébergement écologique, totalement maîtrisée et gratuite à l'usage.

Architecture de la Base de Données

Modèle Relationnel `campus_it`

La base de données MySQL/MariaDB a été conçue pour normaliser le stockage des métriques des départements.

- **Table `application` :** Liste des 8 services critiques (Portail, Helpdesk, Intranet...).
- **Table `ressource` :** Types de métriques suivies (Stockage Go, Réseau Go, CPU vCPU).
- **Table `consommation` :** Table de faits contenant 336 enregistrements, croisant les applications, ressources, volumes et la date d'enregistrement (mois).

```
application (app_id PK, nom)

ressource (res_id PK, nom, unite)

consommation (
  conso_id PK,
  app_id FK,
  res_id FK,
  mois DATE,
  volume DECIMAL
)
```

Extraction de la Donnée (Requêtes SQL)

Pour alimenter dynamiquement le Dashboard, nous avons rédigé des requêtes SQL avancées utilisant des jointures, des groupements (GROUP BY) et des fonctions d'agrégation conditionnelles (CASE WHEN).

1. Top 5 des Applications Gourmandes

```
SELECT App.nom, SUM(Conso.volume) AS volume
FROM consommation Conso
JOIN application App ON Conso.app_id = App.app_id
GROUP BY App.nom
ORDER BY volume DESC
LIMIT 5
```

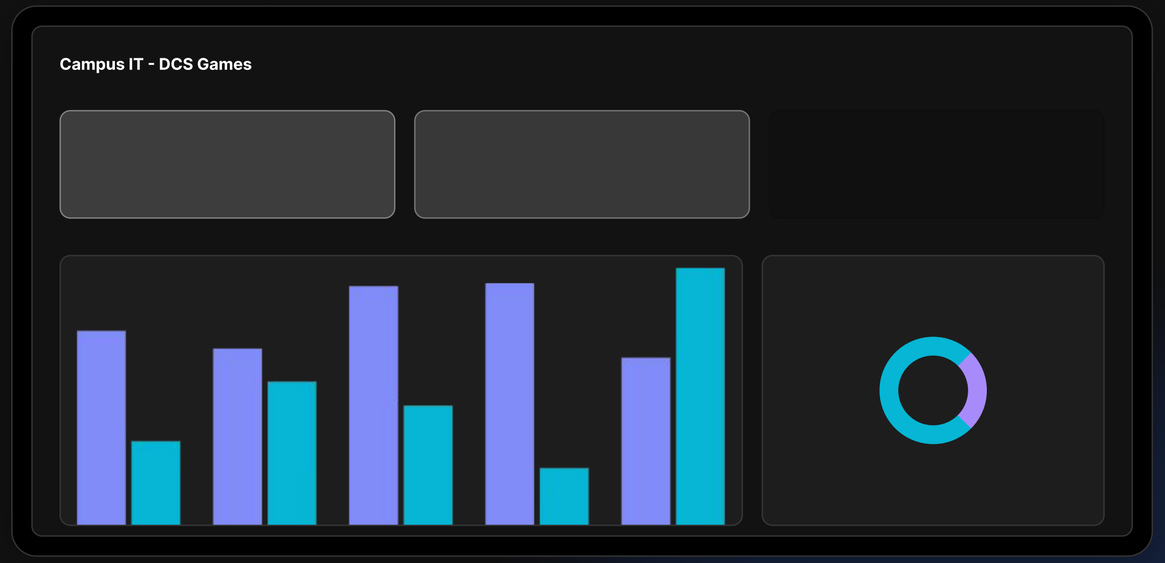
2. Comparaison Pivot : Stockage vs Réseau

```
SELECT mois,
  SUM(CASE WHEN res_id = 1 THEN volume ELSE 0 END)
    AS stockage,
  SUM(CASE WHEN res_id = 3 THEN volume ELSE 0 END)
    AS reseau
FROM consommation
GROUP BY mois ORDER BY mois
```

Design du Dashboard & Implémentation

Développement PHP & JavaScript

- **Connexion PDO sécurisée :** Utilisation d'une classe `DialogueBD` avec blocs Try/Catch.
- **Design System :** Thème sombre "Neon", effets Glassmorphism (cartes floutées), palettes HSL personnalisées (Cyan, Violet, Vert).
- **Intégration Chart.js & Highcharts :** Graphiques en aires (Tendance), barres (Comparaison) et Donut (Vue globale) injectés via JSON.
- **Micro-animations :** JS personnalisé pour l'effet d'incrémentation des chiffres, et le suivi de souris CSS (effets de lumière sur les bordures de cartes).



Synergie Inter-Pôles

La réussite de l'application DCS Games IT a reposé sur une collaboration technique fluide et sécurisée entre l'Infrastructure et le Développement.

Communication

Partage constant des informations réseaux (IP virtuelles, accès WireGuard) et des contraintes applicatives (Versions PHP).

Droits et Sécurité

Création de l'utilisateur MariaDB limité `app_campus` pour protéger l'OS des erreurs applicatives potentielles.

Déploiement Continu

Le pôle SISR a préparé le dossier `/var/www/html` (chown) pour un envoi des fichiers SFTP fluide par le pôle SLAM.

CONCLUSION

Mission Accomplie

L'infrastructure Proxmox est prête pour la production : sécurisée, performante et sauvegardée.

Le Dashboard fourni est dynamique, esthétiquement abouti et répond aux exigences de pilotage de la
Donnée du campus.

Merci de votre attention

Avez-vous des questions ?