

Contents

Azure Vercel : Stratégie de Complémentarité pour Bénéteau	1
Table des matières	1
Positionnement stratégique	1
Vision : Complémentarité, pas remplacement	1
Cas d'usage idéaux	2
Différences techniques détaillées	2
1. Ce que Vercel NE PEUT PAS faire (Azure only)	2
Flux de données et architecture	3
Architecture hybride Azure + Vercel (Recommandé Bénéteau)	3

Azure Vercel : Stratégie de Complémentarité pour Bénéteau

Objectif : Utiliser le Manufacturing Portal comme cas d'usage pour démontrer comment Vercel peut compléter Azure chez Bénéteau, particulièrement pour les POCs, petites apps et déploiements rapides de test.

Table des matières

1. [Positionnement stratégique](#)
 2. [Différences techniques détaillées](#)
 3. [Flux de données et architecture](#)
 4. [Propriété et sécurité des données](#)
 5. [Stratégie de déploiement hybride](#)
 6. [Nouveautés Vercel 2024-2025](#)
 7. [Cas d'usage Manufacturing Portal](#)
-

Positionnement stratégique

Vision : Complémentarité, pas remplacement

ÉCOSYSTÈME BÉNÉTEAU

AZURE (Production)

- Apps critiques
- Infrastructure complexe
- Compliance stricte
- Long-running tasks
- VNet privé

VERCEL (Innovation)

- POCs rapides
- Prototypes
- Apps de test
- Démonstrations clients
- Expérimentations
- Dev/Preview URLs

- Managed Identity

Setup: 1-2 jours

- Itérations rapides

Setup: 30 min

Mêmes ressources :

- Azure APIM (IFS)
- Azure SQL DB
- Microsoft Entra ID
- Azure Storage

Cas d'usage idéaux

Critère	Azure	Vercel
App en production (>1000 users)	Recommandé	Possible
POC / Prototype (<50 users)	Overkill	Parfait
Démo client (temporaire)	Trop lent à setup	URL en 5 min
Tests A/B rapides	Complexe	Preview URLs
App critique 24/7	Recommandé	Cold starts
Long-running jobs (>60s)	Oui	Non (timeout)
VNet privé requis	Oui	Non
Dev rapide	CI/CD à configurer	Push = deploy

Différences techniques détaillées

1. Ce que Vercel NE PEUT PAS faire (Azure only)

A. Managed Identity - Authentification automatique **Azure Managed Identity** : Azure reconnaît automatiquement votre application et lui donne accès aux services (base de données, API) sans avoir besoin de mots de passe.

En pratique : - **Azure** : Pas de mots de passe à stocker, Azure gère l'authentification automatiquement - **Vercel** : Doit obligatoirement stocker les mots de passe et identifiants dans la configuration - **Sécurité Vercel** : Les mots de passe sont chiffrés, mais ils existent et peuvent être volés

Conséquence pour Bénéteau : - Sur Azure : Si un hacker accède à votre code, il ne trouve aucun mot de passe - Sur Vercel : Les mots de passe sont dans la configuration, risque supérieur

B. Synthèse des différences Azure vs Vercel

Fonctionnalité	Azure	Vercel	Impact Bénéteau
Réseau privé (VNet)	Isolé, communication interne sans Internet	Tout passe par Internet public	Azure plus sécurisé, Vercel nécessite firewall
Authentification base de données	Managed Identity (pas de mots de passe)	Mots de passe obligatoires dans config	Azure élimine le risque de vol de credentials
Connexion Azure SQL	Directe via VNet	Via firewall (liste IPs Vercel à maintenir)	Solution : API proxy sur Azure
Surveillance/Monitoring	Azure Monitor (tout centralisé)	Dispersé (logs 24h, outils externes)	Azure facilite le debugging
Authentification users	Easy Auth (config visuelle, 5 min)	NextAuth.js (~100 lignes code, 1-2h)	Azure plus rapide, Vercel plus flexible
Performance	Always-On (50ms constant)	Serverless (1ère visite 800-1500ms puis 50ms)	Vercel adapté aux POCs intermittents
Timeout API	Illimité	60 secondes maximum	Jobs longs doivent passer par Azure
Taille fichiers	Illimité	50 MB maximum	Upload direct vers Azure Blob Storage

Conclusion pour Manufacturing Portal : - Opérations < 5 secondes : Aucun problème sur Vercel - POCs avec trafic intermittent : Démarrage lent acceptable (1-2s première fois) - Données sensibles : API proxy Azure recommandée pour isoler la base de données

Flux de données et architecture

Architecture hybride Azure + Vercel (Recommandé Bénéteau)

graph TB

```

    subgraph DEV[" Développement"]
        DEVELOPER[Developer]
    end

```

```

    subgraph SOURCE[" Code Source - UN SEUL REPO"]
        REPO[GitHub Repository<br/>morepudding/manufacturingPortal]
        MAIN[Branch: main<br/>Production]
        FEATURE[Branch: feature/*<br/>poc/* develop]
    end

```

```

    subgraph AZURE_PIPELINE[" CI/CD Azure"]
        AZURE_CI[GitHub Actions<br/>Workflow: azure-deploy.yml<br/>Trigger: push on main<br/>Build: azure-pipelines.yml]
        AZURE_BUILD[Build & Tests<br/>npm install<br/>npm run build<br/>npm test]
    end

```

```

    subgraph VERCEL_PIPELINE[" CI/CD Vercel"]

```

```

    VERCEL_CI[Vercel Auto-Deploy<br/>Trigger: push on feature/*<br/>Build: 2 min]
end

subgraph PRODUCTION[" PRODUCTION"]
    AZURE_APP[ Azure App Service<br/>manufacturing-portal<br/>.azurewebsites.net<br/><br/>]
end

subgraph INNOVATION[" POC / TESTS"]
    VERCEL_APP[ Vercel Deployment<br/>manufacturing-portal-poc<br/>.vercel.app<br/><br/>&l
end

subgraph SHARED[" RESSOURCES PARTAGÉES"]
    subgraph AUTH_SHARED[" Authentification"]
        AAD_SHARED[ Microsoft Entra ID<br/>App Registration unique<br/>Redirect URIs:<br/>]
    end

    subgraph DATA_SHARED[" Base de Données"]
        SQL_SHARED[( Azure SQL Database<br/>Firewall Rules:<br/>Azure Services<br/>Vercel
    end

    subgraph GATEWAY_SHARED[" API Gateway"]
        APIM_SHARED[ Azure APIM<br/>Public endpoint<br/>Subscription keys<br/>Partagé Azur
    end
end

subgraph EXTERNAL[" Externe"]
    IFS[ IFS Cloud<br/>API OData]
end

subgraph MONITORING[" Observabilité"]
    AZURE_MON[ Azure Monitor<br/>Prod metrics<br/>SQL queries<br/>APIM logs]
    VERCEL_MON[ Vercel Analytics<br/>POC metrics<br/>Web Vitals<br/>Logs 24h]
end

DEVELOPER -->|git push origin main| REPO
DEVELOPER -->|git push origin feature/xyz| REPO

REPO --> MAIN
REPO --> FEATURE

MAIN -->|Webhook trigger| AZURE_CI
AZURE_CI --> AZURE_BUILD
AZURE_BUILD -->|deploy| AZURE_APP

FEATURE -->|Auto-detect| VERCEL_CI
VERCEL_CI -->|deploy| VERCEL_APP

AZURE_APP -->|authentifie| AAD_SHARED

```

```
VERCEL_APP -->|authentifie| AAD_SHARED
```

```
AZURE_APP -->|requêtes SQL| SQL_SHARED
```

```
VERCEL_APP -->|requêtes SQL| SQL_SHARED
```

```
AZURE_APP -->|appels API| APIM_SHARED
```

```
VERCEL_APP -->|appels API| APIM_SHARED
```

```
APIM_SHARED -->|OAuth2| IFS
```

```
AZURE_APP -.->|logs| AZURE_MON
```

```
VERCEL_APP -.->|analytics| VERCEL_MON
```

```
SQL_SHARED -.->|metrics| AZURE_MON
```

```
APIM_SHARED -.->|logs| AZURE_MON
```

```
classDef azureStyle fill:#0078D4,stroke:#003366,stroke-width:3px,color:#fff
classDef vercelStyle fill:#000000,stroke:#333,stroke-width:3px,color:#fff
classDef sharedStyle fill:#10B981,stroke:#065F46,stroke-width:3px,color:#fff
classDef externalStyle fill:#FF6B6B,stroke:#C92A2A,stroke-width:2px,color:#fff
classDef monitorStyle fill:#8B5CF6,stroke:#5B21B6,stroke-width:2px,color:#fff
classDef sourceStyle fill:#F59E0B,stroke:#D97706,stroke-width:2px,color:#fff
```

```
class AZURE_APP,AZURE_CI,AZURE_BUILD,AZURE_PIPELINE azureStyle
```

```
class VERCEL_APP,VERCEL_CI,VERCEL_PIPELINE vercelStyle
```

```
class AAD_SHARED,SQL_SHARED,APIM_SHARED sharedStyle
```

```
class IFS externalStyle
```

```
class AZURE_MON,VERCEL_MON monitorStyle
```

```
class REPO,MAIN,FEATURE sourceStyle
```

Avantages de cette architecture : - **UN SEUL REPO** : Code partagé, pas de duplication - **Déploiement automatique** : Pas d'intervention manuelle selon la branche - **Ressources partagées** : Azure AD, SQL, APIM utilisés par les deux - **Preview URLs automatiques** : Chaque branche feature = URL de test unique - **CI/CD Azure robuste** : Tests automatiques avant production - **Flexibilité** : Production stable (Azure) + Innovation rapide (Vercel)

Je continue avec la partie suivante sur la propriété des données et la sécurité ?

Contents

Vercel : Propriété, Sécurité et Flux des Données	1
Table des matières	1
Propriété du code et des données	1
Votre code source	1
Vos données applicatives	2
Flux des données	3
Exemple : Affichage d'une page	3
Ce que Vercel voit et stocke	4
Tableau de synthèse	4
Contenu des logs Vercel	5
Sécurité et conformité	5
Chiffrement	5
Certifications	5
Localisation géographique	5
Comparaison avec Azure	6
Architecture réseau	6
Tableau comparatif	6
Recommandations (optionnel)	7
Architecture sécurisée avec Gateway Azure	7
Gestion des secrets	8
Plan de sortie	8
Résumé	8
Documentation complémentaire	8

Vercel : Propriété, Sécurité et Flux des Données

Document technique expliquant où vont les données, qui en est propriétaire, et comment les flux de données fonctionnent avec Vercel

Table des matières

1. [Propriété du code et des données](#)
 2. [Flux des données](#)
 3. [Ce que Vercel voit et stocke](#)
 4. [Sécurité et conformité](#)
 5. [Comparaison avec Azure](#)
 6. [Recommandations \(optionnel\)](#)
-

Propriété du code et des données

Votre code source

Question : Est-ce que je reste propriétaire de mon code ?

Réponse : OUI, complètement

```
graph LR
    subgraph "Votre propriété"
        GITHUB[GitHub Repository<br/>morepudding/manufacturingPortal<br/><br/>PROPRIÉTAIRE: Bér]
    end

    subgraph "Vercel (accès lecture seule)"
        VERCEL_READ[Vercel lit le code<br/>pour le build<br/><br/>PAS DE PROPRIÉTÉ]
    end

    subgraph "Build artifacts"
        BUILD[Code compilé<br/>temporaire<br/><br/>Supprimé après déploiement]
    end

    GITHUB -->|clone| VERCEL_READ
    VERCEL_READ -->|compile| BUILD
    BUILD -->|deploy puis supprime| VERCEL_READ

    classDef ownerStyle fill:#10B981,stroke:#065F46,stroke-width:3px,color:#fff
    classDef tempStyle fill:#F59E0B,stroke:#D97706,stroke-width:2px,color:#fff
    classDef readStyle fill:#6B7280,stroke:#374151,stroke-width:2px,color:#fff

    class GITHUB ownerStyle
    class BUILD tempStyle
    class VERCEL_READ readStyle
```

Process Vercel : 1. **Lecture** : Se connecte à GitHub (permission lecture seule) 2. **Build** : Compile le code temporairement 3. **Déploiement** : Copie les fichiers compilés (HTML, JS, CSS) vers CDN 4. **Suppression** : Le code source n'est PAS conservé

Propriété intellectuelle : - Bénéteau garde 100% des droits - Vercel ne peut pas utiliser, vendre ou partager le code - Possibilité de quitter Vercel à tout moment

Source : [Vercel Terms of Service - Section 6](#) - "You retain all rights to your content and code"

Vos données applicatives

Question : Où sont stockées mes données (base de données, fichiers) ?

Réponse : Chez Bénéteau (Azure/IFS uniquement)

```
graph TB
    subgraph VERCEL["Vercel (Frontend uniquement)"]
        APP[Application Next.js<br/>Code compilé<br/>AUCUNE donnée métier]
    end

    subgraph BENETEAU["Infrastructure Bénéteau"]
        SQL[(Azure SQL Database<br/>Shop Orders<br/>Configurations<br/><br/>PROPRIÉTAIRE: Bénéte)]
    end
```

```

    APIM[Azure APIM<br/>Gateway IFS<br/><br/>PROPRIÉTAIRE: Bénéteau]

    AAD[Microsoft Entra ID<br/>Utilisateurs<br/><br/>PROPRIÉTAIRE: Bénéteau]

    STORAGE[Azure Blob Storage<br/>Fichiers, PDFs<br/><br/>PROPRIÉTAIRE: Bénéteau]
end

subgraph EXTERNAL["Services externes"]
    IFS[IFS Cloud<br/>Données ERP<br/><br/>PROPRIÉTAIRE: Bénéteau]
end

APP -->|Requêtes SQL| SQL
APP -->|Appels API| APIM
APP -->|Auth| AAD
APP -->|Upload/Download| STORAGE
APIM -->|OAuth2| IFS

classDef vercelStyle fill:#000000,stroke:#333,stroke-width:2px,color:#fff
classDef beneteauStyle fill:#10B981,stroke:#065F46,stroke-width:3px,color:#fff

class APP vercelStyle
class SQL,APIM,AAD,STORAGE,IFS beneteauStyle

```

État actuel : - Vercel héberge uniquement le code frontend compilé (HTML/JS/CSS) ¹ - AUCUNE donnée métier stockée sur Vercel (architecture serverless sans persistance) ² - Toutes les données restent sur Azure SQL, IFS Cloud et Azure Storage ³

Sources : - [Vercel Edge Network](#) - “Static assets and server-rendered pages are cached and served from our global Edge Network” - [Vercel Functions](#) - “Functions are stateless and ephemeral by design. They do not persist data between invocations” - [Vercel Storage Guide](#) - “Vercel does not include a built-in database. Connect to external databases like PostgreSQL, MySQL, or MongoDB hosted elsewhere”

Flux des données

Exemple : Affichage d’une page

```

sequenceDiagram
    actor User as Utilisateur
    participant Vercel as Vercel
    participant Azure as Azure SQL
    participant APIM as Azure APIM
    participant IFS as IFS Cloud

    User->>Vercel: 1. GET /boat-configuration
    Vercel->>User: 2. Page HTML + JS

```



```

User->>Vercel: 3. GET /api/shop-orders
Vercel->>Azure: 4. SELECT * FROM ShopOrders
Azure->>Vercel: 5. Résultats SQL
Vercel->>User: 6. JSON response

User->>Vercel: 7. GET /api/ifs/serial-numbers
Vercel->>APIM: 8. GET /ShopOrderHandling.svc
APIM->>IFS: 9. OAuth2 + OData
IFS->>APIM: 10. Données IFS
APIM->>Vercel: 11. JSON
Vercel->>User: 12. JSON response

```

Note over User,IFS: Transit uniquement - Aucun stockage sur Vercel

Comportement : - Frontend servi depuis le CDN Vercel (Edge Network) - API calls exécutés par Serverless Functions Vercel (compute sans état) - Données transitent en mémoire uniquement (durée d'exécution de la fonction) - Aucun stockage permanent sur Vercel (pas de base de données intégrée par défaut) - Toutes les données restent sur Azure/IFS (connexions externes via HTTPS)

3

Sources : - [Edge Network Caching](#) - "Content is distributed across 100+ global edge locations" - [Serverless Functions](#) - "Functions execute on-demand without managing servers" - [Function Lifecycle](#) - "Each function invocation is isolated and short-lived. Memory state is not shared between invocations" - [File System](#) - "The filesystem is read-only except for /tmp which is cleared between invocations"

Ce que Vercel voit et stocke

Tableau de synthèse

Type de donnée	Vercel voit ?	Vercel stocke ?	Durée conservation
Code source	Oui (pendant build)	Non	Supprimé après build
Code compilé	Oui	Oui	Permanent (tant que déployé)
Variables d'env	Oui (chiffrées)	Oui (chiffrées)	Permanent
Logs d'exécution	Oui	Oui	24 heures (Pro) / 1 heure (Hobby)
Métriques performance	Oui	Oui	30 jours
Données SQL	Non (transit)	Non	Jamais
Données IFS	Non (transit)	Non	Jamais
Fichiers uploadés	Non	Non	Jamais
Données utilisateurs	Non (transit)	Non	Jamais

Contenu des logs Vercel

Exemple :

```
2025-11-13 14:23:45 | GET /api/shop-orders | 200 | 234ms
2025-11-13 14:23:46 | GET /api/ifs/serial-numbers?orderId=563 | 200 | 1823ms
2025-11-13 14:23:50 | POST /api/print | 200 | 456ms
```

Vercel voit : URL, méthode HTTP, statut, temps d'exécution, IP utilisateur

Vercel NE voit PAS : Contenu SQL, données API, mots de passe, tokens, fichiers

Sources : - [Runtime Logs](#) - "Logs include request URL, method, status code, duration, and IP address" - [Security & Compliance](#) - "We do not log request or response bodies. Environment variables are encrypted at rest (AES-256-GCM)"

Sécurité et conformité

Chiffrement

En transit :

Utilisateur [HTTPS/TLS 1.3] > Vercel [HTTPS/TLS 1.3] > Azure SQL
[HTTPS/TLS 1.3] > Azure APIM

- Chiffrement bout-en-bout
- Certificats SSL automatiques (Let's Encrypt)
- TLS 1.3

Au repos : - Variables d'environnement : AES-256 - Code compilé : Chiffré sur CDN - Logs : AES-256

Source : [Vercel Security Whitepaper](#) - "TLS 1.3 for data in transit, AES-256 for data at rest"

Certifications

Certification	Status
SOC 2 Type II	Certifié
ISO 27001	Certifié
GDPR	Conforme
CCPA	Conforme

Sources : - [Vercel Trust Center](#) - Certifications SOC 2 & ISO 27001 actives et vérifiables - [Privacy Policy](#) - "We are compliant with GDPR and process data according to EU regulations"

Localisation géographique

Vercel Functions : - Régions disponibles : US East/West, Europe (Amsterdam, Dublin, Frankfurt), Asia Pacific - Configuration possible : Forcer Europe uniquement

Azure (inchangé) : - France Central / West Europe - Pas de transfert hors UE - Conformité RGPD garantie

Comparaison avec Azure

Architecture réseau

graph TB

```
subgraph AZURE["Azure uniquement"]
    A_APP[App Service<br/>France Central]
    A_SQL[(SQL Database<br/>France Central)]
    A_STORAGE[Blob Storage<br/>France Central]
    A_APIM[APIM<br/>France Central]

    A_APP -.->|VNet privé| A_SQL
    A_APP -.->|VNet privé| A_STORAGE
    A_APP -.->|VNet privé| A_APIM
end

subgraph HYBRID["Hybride Azure + Vercel"]
    V_APP[Vercel Function<br/>Amsterdam/Paris]
    H_SQL[(SQL Database<br/>France Central)]
    H_STORAGE[Blob Storage<br/>France Central]
    H_APIM[APIM<br/>France Central]

    V_APP -->|Internet public<br/>HTTPS| H_SQL
    V_APP -->|Internet public<br/>HTTPS| H_STORAGE
    V_APP -->|Internet public<br/>HTTPS| H_APIM
end

classDef azureStyle fill:#0078D4,stroke:#003366,stroke-width:2px,color:#fff
classDef vercelStyle fill:#000000,stroke:#333,stroke-width:2px,color:#fff

class A_APP,A_SQL,A_STORAGE,A_APIM azureStyle
class V_APP vercelStyle
class H_SQL,H_STORAGE,H_APIM azureStyle
```

Tableau comparatif

Aspect	Azure uniquement	Hybride Azure + Vercel
Propriétaire données	Bénéteau	Bénéteau
Localisation données	France/Europe	France/Europe (Azure)
Réseau	VNet privé	Internet public (HTTPS)
Logs applicatifs	Azure Monitor (30-90j)	Vercel (24h) + Azure Monitor
Accès admin	Équipe IT Bénéteau	Équipe IT Bénéteau
Accès tiers	Aucun	Vercel (logs transit)

Aspect	Azure uniquement	Hybride Azure + Vercel
Conformité RGPD	Total	Total (données en EU)
Audit trail	Azure AD + Monitor	Azure AD + Vercel logs
Backup données	Azure Backup	Azure Backup (inchangé)

Constat : Avec Vercel, les données métier restent à 100% chez Bénéteau. Seul le frontend est externalisé.

Recommandations (optionnel)

Architecture sécurisée avec Gateway Azure

Si nécessaire, possibilité d'ajouter un proxy Azure pour isoler complètement les accès :

```
graph TB
    subgraph USERS["Utilisateurs"]
        USER[Employé Bénéteau]
    end

    subgraph VERCEL["Vercel"]
        V_APP[Next.js Application]
    end

    subgraph AZURE_GATEWAY["Azure Gateway"]
        GATEWAY[Azure Function<br/>Proxy sécurisé]
    end

    subgraph AZURE_DATA["Azure Données privées"]
        SQL[(SQL Database)]
        APIM[APIM]
        STORAGE[Blob Storage]
    end

    USER -->|HTTPS| V_APP
    V_APP -->|API calls| GATEWAY

    GATEWAY -->|VNet privé| SQL
    GATEWAY -->|VNet privé| APIM
    GATEWAY -->|VNet privé| STORAGE

    classDef userStyle fill:#6B7280,stroke:#374151,stroke-width:2px,color:#fff
    classDef vercelStyle fill:#000000,stroke:#333,stroke-width:2px,color:#fff
    classDef gatewayStyle fill:#F59E0B,stroke:#D97706,stroke-width:3px,color:#fff
    classDef dataStyle fill:#10B981,stroke:#065F46,stroke-width:3px,color:#fff

    class USER userStyle
    class V_APP vercelStyle
    class GATEWAY gatewayStyle
    class SQL dataStyle
    class APIM dataStyle
    class STORAGE dataStyle
```

```
class V_APP vercelStyle
class GATEWAY gatewayStyle
class SQL,APIM,STORAGE dataStyle
```

Avantages : - Base de données reste en VNet privé - Vercel ne connaît jamais les credentials Azure
- Contrôle total via le Gateway

Gestion des secrets

Configuration minimale :

```
// Dans Vercel
AZURE_GATEWAY_URL=https://beneteau-gateway.azurewebsites.net
AZURE_GATEWAY_KEY=<clé publique rotatable>
```

```
// Dans Azure Gateway
DATABASE_CONNECTION_STRING=<connection complète>
IFS_CLIENT_SECRET=<secret OAuth2>
APIM_SUBSCRIPTION_KEY=<clé APIM>
```

Plan de sortie

Si nécessaire de quitter Vercel :

1. < 1h : Supprimer déploiement + révoquer accès GitHub
2. 1-2 jours : Déployer sur Azure App Service + changer DNS
3. 1 semaine : Audit logs + suppression projet Vercel

Aucune donnée perdue (tout reste sur Azure).

Résumé

Propriétaire du code ? → Bénéteau 100%

Localisation des données ? → Azure France/Europe

Vercel voit les données ? → Non, transit uniquement

Conforme RGPD ? → Oui

Réversible ? → Oui, facilement

Verdict : Vercel adapté pour POCs/tests. Données métier restent chez Bénéteau.

Documentation complémentaire

Architecture Next.js sur Vercel : - [Next.js Deployment](#) - [Vercel Platform Overview](#)

Sécurité et conformité : - [Vercel Security Documentation](#) - [GDPR Compliance Guide](#)

Intégrations externes : - [Database Integration Guide](#) - [Environment Variables Best Practices](#)