

# SAAS

Le système nerveux logiciel des entreprises modernes

---

Software as a Service | Cloud | Subscription | Product Architecture

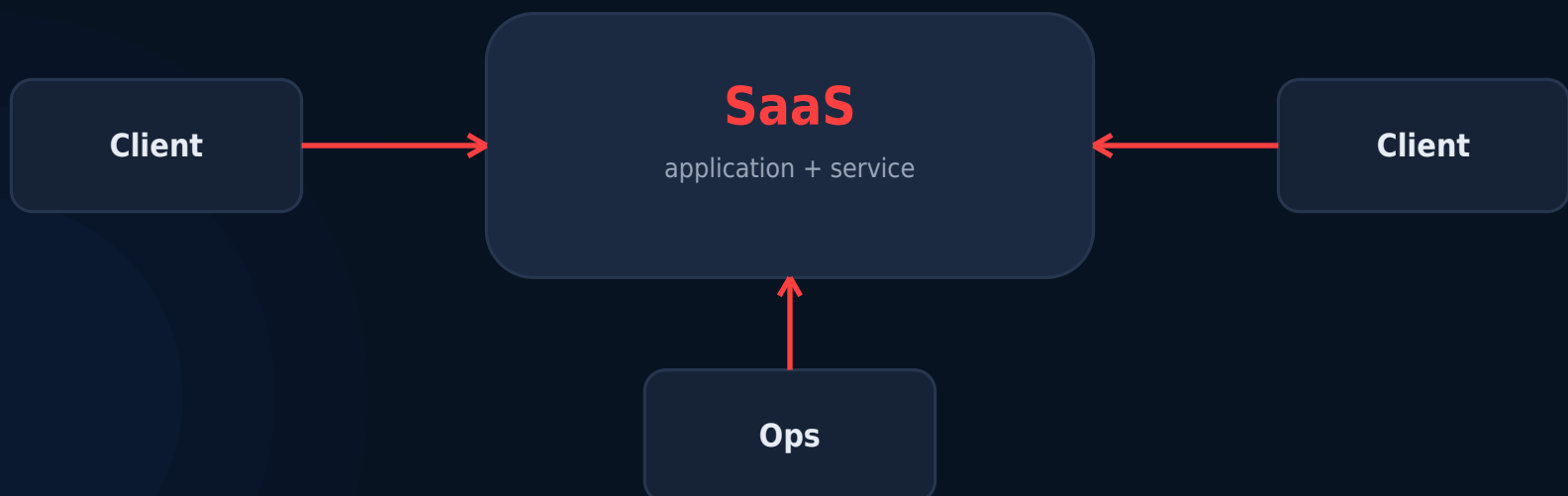
# C'EST QUOI LE SAAS ?

## Définition simple

Une application SaaS est un logiciel livré par Internet, généralement via navigateur, sans installation locale lourde pour le client. Le fournisseur opère l'infrastructure, les mises à jour, la sécurité et la disponibilité.

## Philosophie unique

Le logiciel cesse d'être un produit installé une fois. Il devient un service vivant : abonnement, amélioration continue, données centralisées, support intégré, disponibilité mesurable.



# POURQUOI LE SAAS A TOUT CHANGÉ

# 0

## SERVEUR LOCAL À MAINTENIR CÔTÉ SERVICE CONTINU

Le client consomme une capacité applicative, pas une pile technique complète.

# 24/7

La valeur est disponible partout, mesurable, supportée et mise à jour en continu.

# 1

## VERSION VIVANTE

Le produit évolue en permanence, sans cycles lourds de déploiement client.

### **Le basculement mental**

Avant : vendre une licence. Après : exploiter un service, fidéliser un client, améliorer le produit chaque semaine et prouver la valeur dans l'usage réel.

# AVANT / APRÈS

Ancien monde	Monde SaaS
Licence + installation	Abonnement + accès web
Mise à jour annuelle	Déploiement continu
Infrastructure client	Infrastructure fournisseur
Support réactif	Service observé en temps réel
Coût initial élevé	Coût étalé et ajustable

# LES 5 MARQUEURS DU SAAS

01

## Accès réseau

L'application est disponible par Internet : navigateur, mobile, API.

02

## Service géré

Le fournisseur exploite la plateforme, pas le client.

03

## Abonnement

Paiement récurrent lié au temps, à l'usage, aux sièges ou à la valeur.

04

## Scalabilité

Capacité à absorber plus de clients, données et transactions.

05

## Mesure

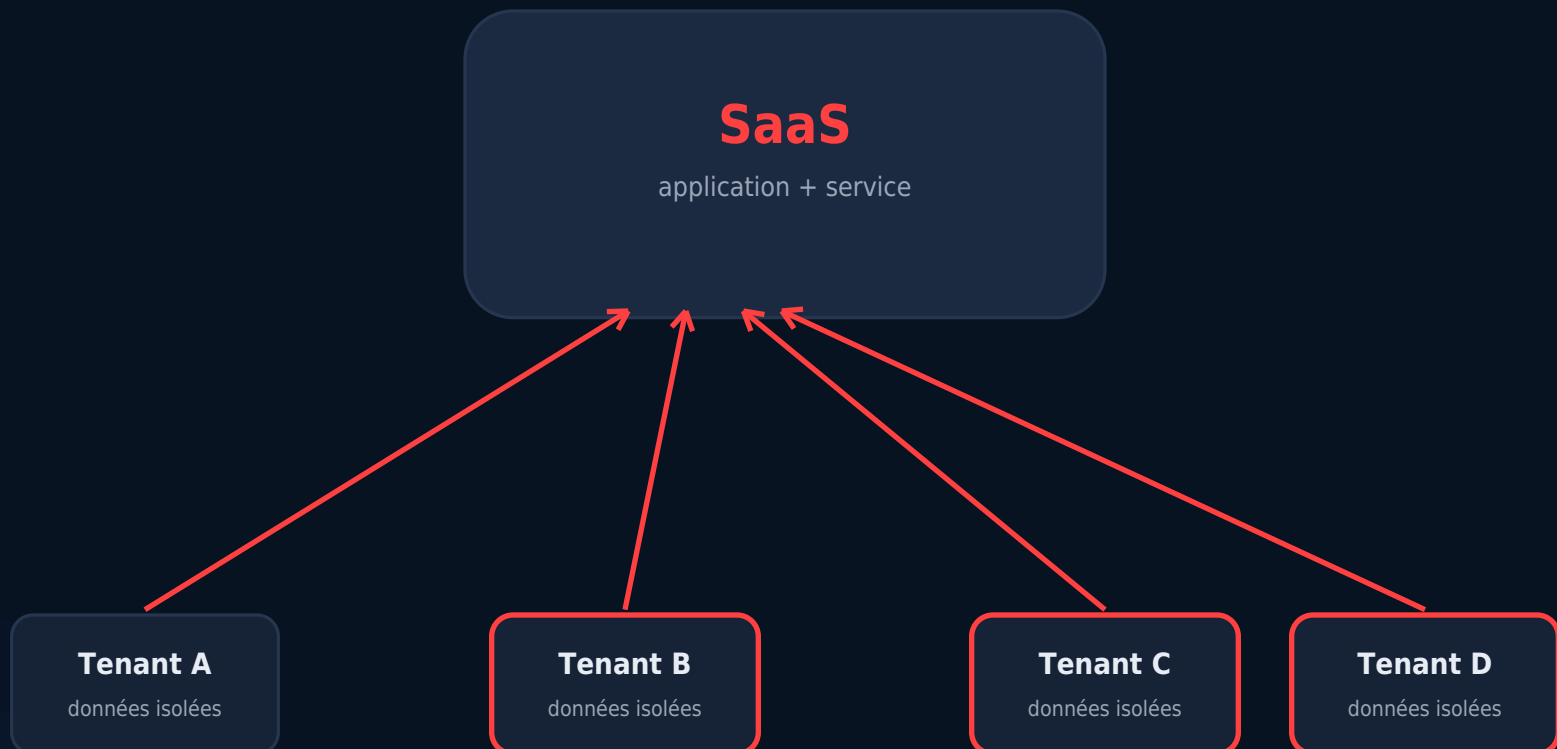
Usage, performance, coûts, disponibilité : tout devient observable.

# IAAS, PAAS, SAAS : QUI GÈRE QUOI ?

	On-prem	IaaS	PaaS	SaaS
Application	Client	Client	Client	Fournisseur
Données	Client	Client	Client	Fournisseur
Runtime	Client	Client	Fournisseur	Fournisseur
OS	Client	Client	Fournisseur	Fournisseur
Virtualisation	Client	Fournisseur	Fournisseur	Fournisseur
Serveurs	Client	Fournisseur	Fournisseur	Fournisseur
Stockage	Client	Fournisseur	Fournisseur	Fournisseur
Réseau	Client	Fournisseur	Fournisseur	Fournisseur

Dans un SaaS, le client ne gère presque plus la pile technique. Il gère surtout l'usage, les droits, la gouvernance des données et le contrat de service.

# LA MULTI-TENANCE



## Idée clé

Plusieurs clients partagent une plateforme commune, mais leurs données, configurations, droits et limites restent strictement séparés.

# ISOLATION : LE MUR INVISIBLE

## Isolation logique

Un seul code, une seule plateforme, mais chaque requête est liée à un `tenant_id` et contrôlée à toutes les couches.

## Isolation des données

Schéma partagé, schéma dédié ou base dédiée selon niveau de risque, coût, performance et conformité.

## Isolation opérationnelle

Quota, rate limit, monitoring par client, alerting ciblé, support par tenant.

### Tenant A

identité + données + quotas

### Tenant B

identité + données + quotas

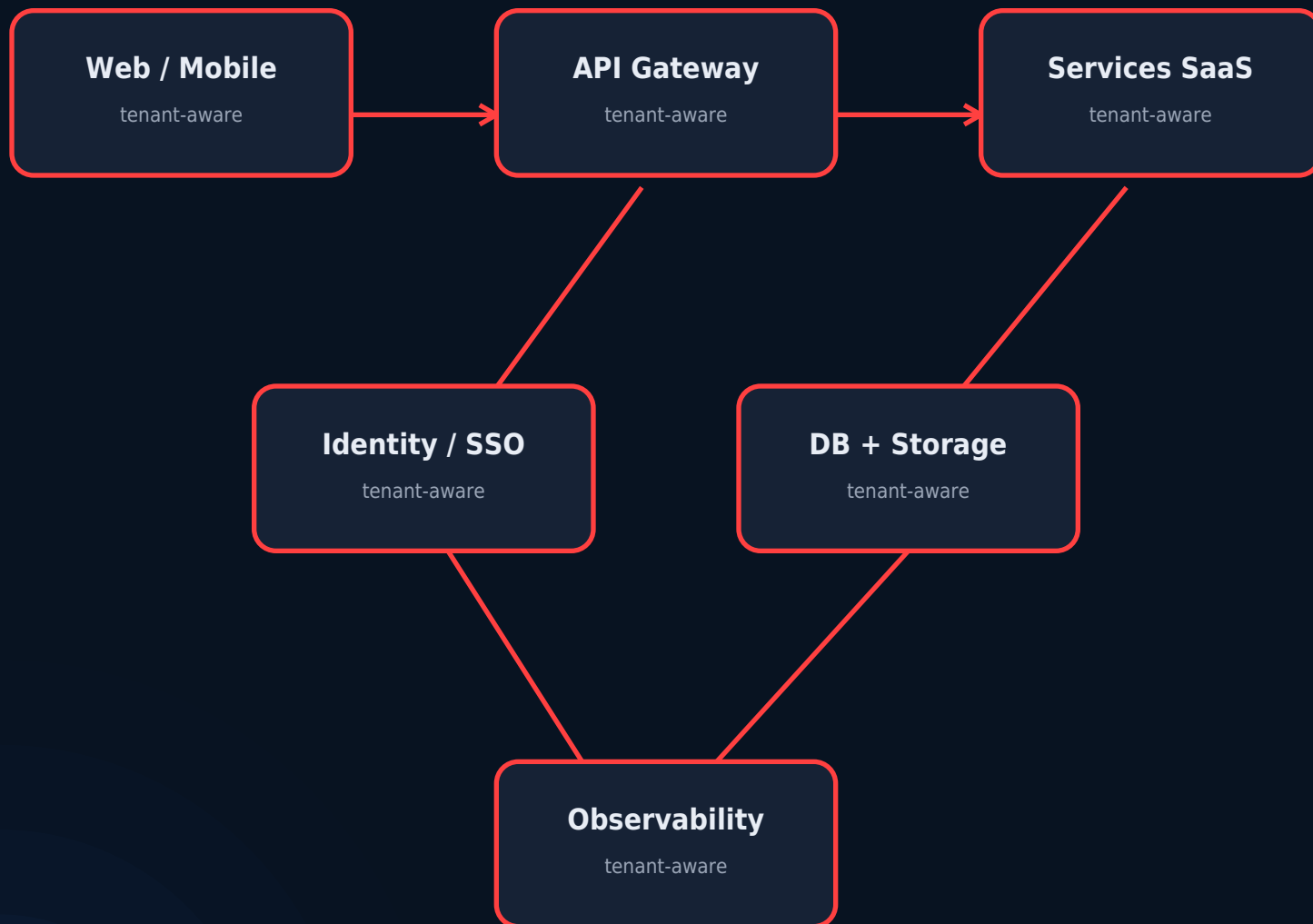
### Tenant C

identité + données + quotas

# PATTERNS DE DONNÉES

Pattern	Avantage	Risque
Base partagée	Coût bas, simplicité DevOps	Isolation applicative critique
Schéma par tenant	Bon compromis isolation/coût	Migrations plus complexes
Base par tenant	Isolation forte, conformité	Coût et orchestration élevés
Hybride	Segmentation par taille/risque	Gouvernance plus exigeante

# ARCHITECTURE DE RÉFÉRENCE



# LE MODÈLE ÉCONOMIQUE

## Revenu récurrent

Le SaaS remplace le revenu ponctuel par un flux : MRR, ARR, renouvellement, expansion, rétention.

## Valeur dans la durée

Le client reste si le produit devient indispensable, fiable et facile à adopter.

## Coût marginal

Une même plateforme sert de nombreux clients : l'économie d'échelle devient centrale.

# ARR

ANNUAL RECURRING REVENUE

# NRR

NET REVENUE RETENTION

# LTV

LIFETIME VALUE

# PRICING : VENDRE LA VALEUR

P1

## Par utilisateur

Simple, lisible, adapté aux outils collaboratifs.

P2

## Par usage

Aligné sur consommation : API calls, stockage, documents, temps CPU.

P3

## Par palier

Good / Better / Best : segmentation claire du marché.

P4

## Par valeur métier

Prix lié au résultat : volume traité, économies, revenu généré.

# LES MÉTRIQUES VITALES

## MRR

revenu mensuel récurrent

## ARR

$MRR \times 12$

## Churn

clients ou revenu perdus

## NRR

réention + expansion

## CAC

coût d'acquisition

## LTV

valeur vie client

# UNIT ECONOMICS : L'ÉQUATION

$$\text{LTV} > 3 \times \text{CAC}$$

sinon la croissance peut détruire de la valeur

## Ce qui améliore LTV

Onboarding, adoption, expansion, support, fiabilité, intégrations, stickiness produit.

## Ce qui réduit CAC

Positionnement clair, product-led growth, contenu, referral, partenaires, cycle de vente court.

Dans le SaaS, la stratégie produit, la stratégie commerciale et l'architecture technique sont liées par les mêmes métriques.

# GTM : PRODUCT-LED VS SALES-LED

## **Product-Led Growth**

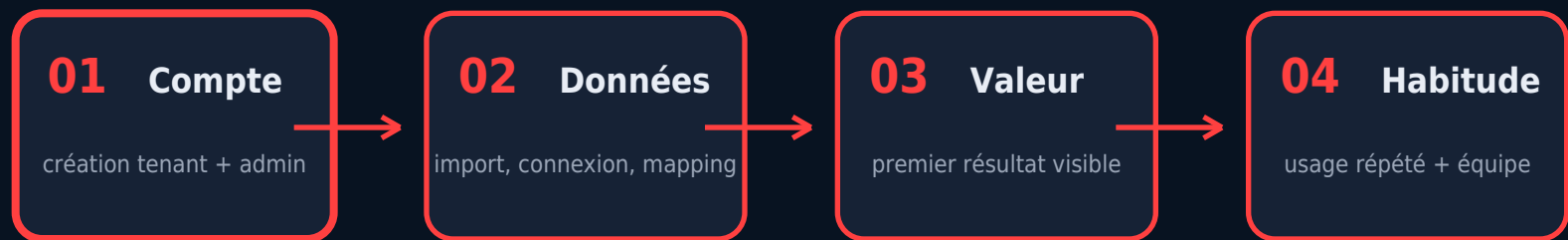
Le produit vend le produit : essai gratuit, self-service, activation rapide, adoption organique dans les équipes.

## **Sales-Led Growth**

Vente structurée : démonstration, preuve de valeur, sécurité, procurement, contrat, accompagnement entreprise.

*Le meilleur SaaS sait souvent combiner les deux sans les mélanger.*

# ONBOARDING : LE MOMENT DE VÉRITÉ



## Règle d'or

Un SaaS qui ne donne pas une première valeur rapidement augmente mécaniquement son churn. L'onboarding doit être conçu comme un produit dans le produit.

# SÉCURITÉ : RESPONSABILITÉ PARTAGÉE

Responsabilité client	Responsabilité fournisseur
Gestion des utilisateurs, MFA, rôles	Infrastructure, patching, durcissement
Classification et usage des données	Chiffrement, sauvegardes, disponibilité
Contrôle des accès internes	Journalisation, monitoring, réponse incident
Conformité métier et contrats	Contrôles techniques et preuves d'audit

# IDENTITÉ : LA PORTE D'ENTRÉE

## **SSO / SAML / OIDC**

Connexion centralisée avec l'identité entreprise, réduction des mots de passe faibles.

## **RBAC / ABAC**

Rôles, permissions, attributs, séparation stricte par tenant et par équipe.

## **Audit trail**

Qui a fait quoi, quand, sur quelle donnée : indispensable pour entreprise.

# No identity, no SaaS.

# CONFORMITÉ : PROUVER LA CONFIANCE

- RGPD : base légale, minimisation, droits, DPA, localisation.
- SOC 2 / ISO 27001 : contrôles, procédures, preuves et audits.
- Données sensibles : chiffrement, segmentation, accès privilégiés.
- Logs et rétention : utiles, mais jamais sans gouvernance.

## Point stratégique

La conformité n'est pas seulement juridique. Elle influence l'architecture, la vente entreprise, le support, les backups, le monitoring et la roadmap produit.

# | SLA, SLO, SLI

# 99.9%

ce n'est pas une promesse marketing : c'est une discipline opérationnelle

## **SLI**

Indicateur mesuré :  
latence, taux d'erreur,  
disponibilité, saturation.

## **SLO**

Objectif interne  
mesurable qui guide les  
priorités techniques.

## **SLA**

Engagement  
contractuel envers le  
client, souvent avec  
compensation.

# OBSERVABILITÉ SAAS

## Logs

Événements applicatifs et sécurité, corrélés par tenant.

## Metrics

Taux d'erreur, temps de réponse, usage, quotas, coûts.

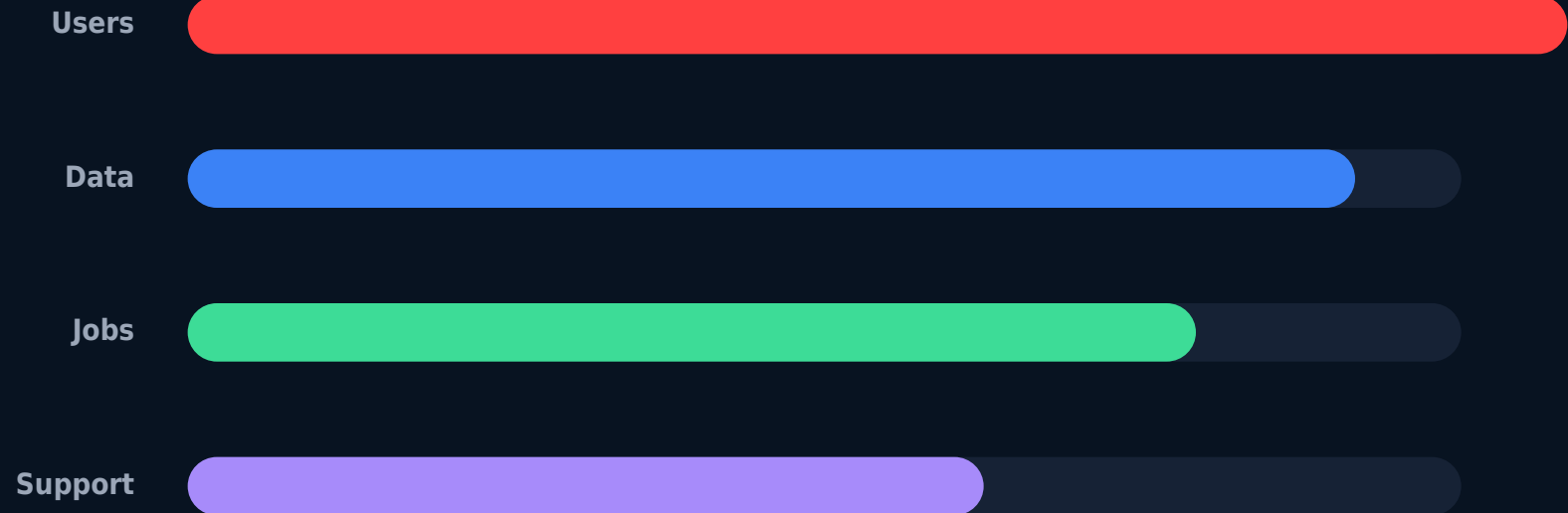
## Traces

Parcours d'une requête dans l'architecture distribuée.

## Business telemetry

Activation, adoption, churn risk, feature usage.

# SCALABILITÉ : GRANDIR SANS CASSER



## Le vrai défi

Scaler un SaaS, ce n'est pas seulement ajouter des serveurs. C'est scaler les migrations, les coûts, le support, les backups, les incidents, les intégrations et la gouvernance.

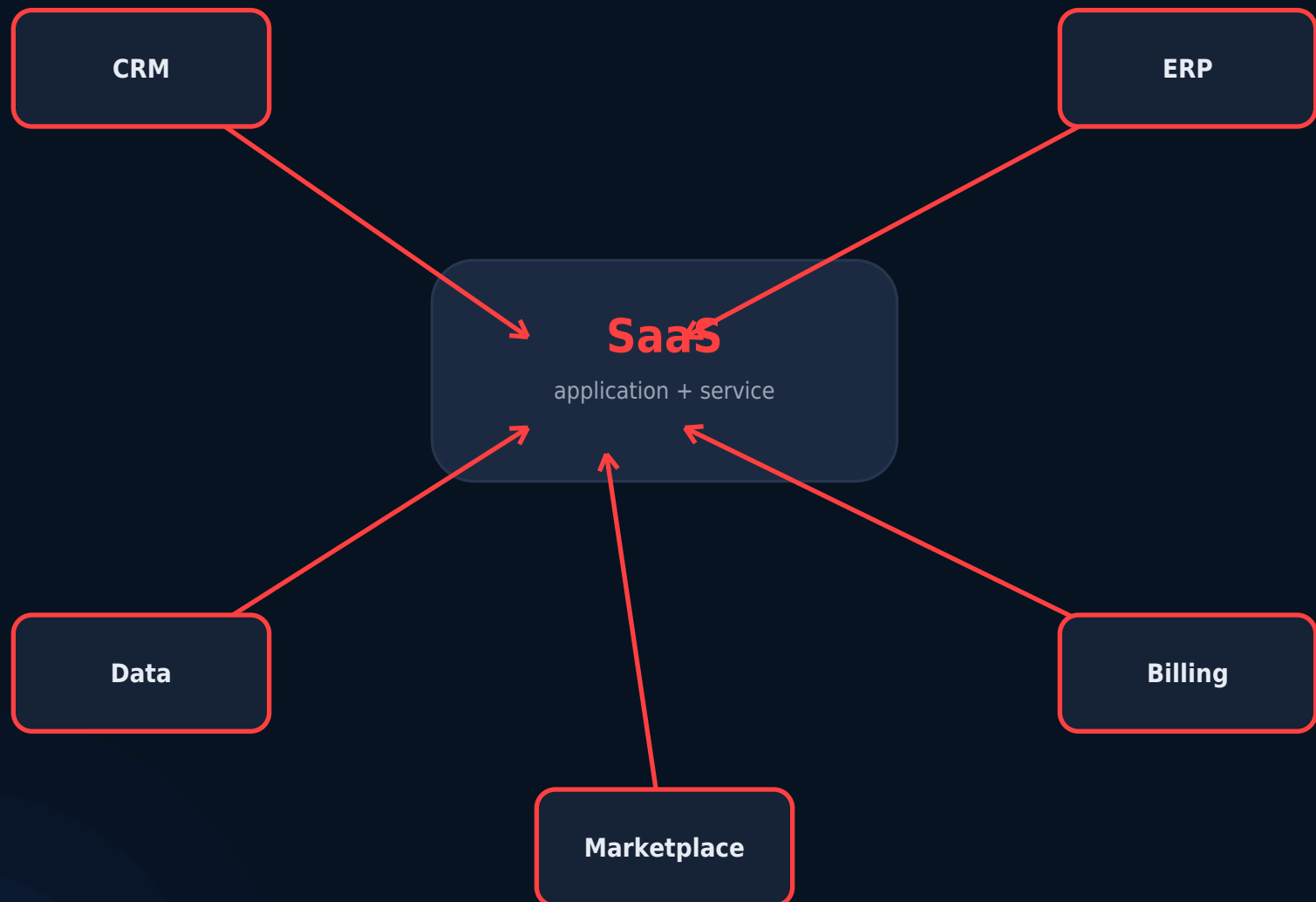
# DÉPLOIEMENT CONTINU



## **En SaaS, le déploiement est un muscle**

Feature flags, canary release, migration progressive, rollback, monitoring post-release : la plateforme doit évoluer sans interrompre le service.

# ÉCOSYSTÈME : APIS, WEBHOOKS, MARKETPLACE



Un SaaS isolé est fragile. Un SaaS intégré devient une plateforme.

# DONNÉES : SOUVERAINETÉ ET RÉSIDENCE

## Résidence

Où les données sont stockées et répliquées : région, pays, cloud, sauvegardes.

## Portabilité

Exporter, supprimer, migrer : le client doit pouvoir reprendre le contrôle.

## Gouvernance

Classification, rétention, purge, chiffrement, accès privilégiés.



# IA + SAAS : L'ÉTAPE SUIVANTE

## Copilots

Assistance contextuelle dans le workflow existant : recherche, résumé, recommandation.

## Agents

Automatisation de tâches complexes avec garde-fous, validations et traçabilité.

## Data advantage

La donnée d'usage, correctement gouvernée, devient un levier de personnalisation.

L'IA ne remplace pas forcément le SaaS : elle transforme l'interface, l'automatisation, le pricing et la valeur perçue.

# LES PIÈGES CLASSIQUES

!

**Fausse multi-tenance**  
copier/coller des instances par client

!

**Pricing illisible**  
trop de plans, trop d'options, trop de friction

!

**Dette sécurité**  
MFA, audit, RBAC repoussés trop tard

!

**Produit sans adoption**  
beaucoup de features, peu de valeur immédiate

!

**Coûts invisibles**  
cloud, support, logs, IA, stockage non pilotés

!

**Migration brutale**  
versions et schémas non compatibles

# CONSTRUIRE UN SAAS : MÉTHODE COURTE

## 1 Segment client

un problème précis

## 2 MVP utile

première valeur rapide

## 3 Tenant model

isolation dès le début

## 4 Billing

paiement + plan

## 5 Observabilité

usage + coût + incidents

## 6 Sécurité

MFA, audit, RBAC

# CHECKLIST DE LANCEMENT

- ✓ Tenant creation automated
- ✓ User roles and invite flow
- ✓ Billing and plan limits
- ✓ Backups and restore tested
- ✓ SLO dashboard ready
- ✓ Audit logs searchable
- ✓ Export/delete data process
- ✓ Incident playbook written
- ✓ Support channels defined
- ✓ Activation metric measured

# Questions ?

Merci pour votre attention

## Sources & inspiration

- NIST SP 800-145 - The Definition of Cloud Computing
- AWS - What is SaaS? / SaaS Architecture Fundamentals
- Microsoft Azure - What is SaaS? / SaaS Workload Documentation
- Salesforce - SaaS definition and business framing
- Redis guide fourni par l'utilisateur - inspiration visuelle et rythme de présentation