

Exercice Scientifique et technique

Conception des APIs : un enjeu stratégique

Li Yohan

1. Bpifrance et son Système d'Information

Bpifrance – Banque Publique d'Investissement

- Finance et accompagne les entreprises, des startups aux ETI
- Acteur central de la transformation numérique de l'économie française

Un SI étendu et hétérogène

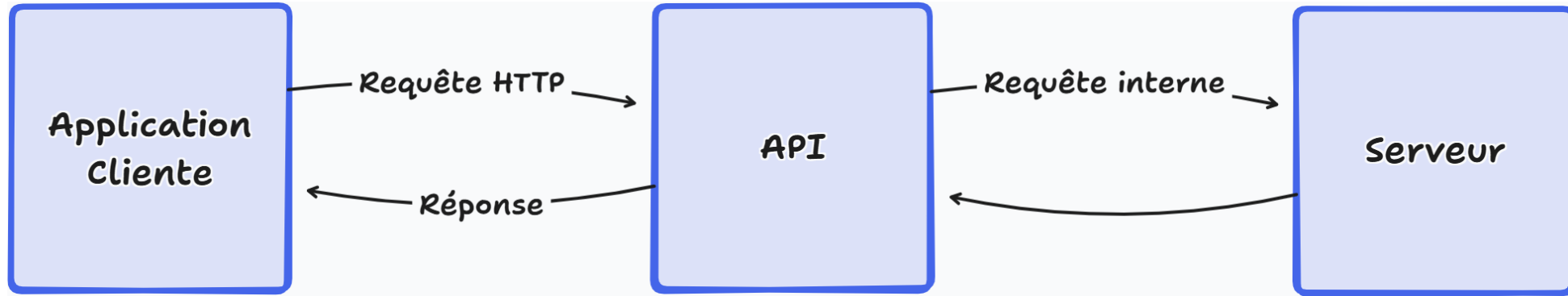
- Applications internes (chargés d'affaires), portails clients digitaux, interfaces partenaires (Fintechs, banques, réglementaires...)
- Toutes ces briques doivent communiquer en permanence

La stratégie API de Bpifrance

- Objectif : aucune application ne communique directement avec une autre sans passer par une couche d'API
- Trois niveaux d'exposition : APIs privées, partenaires et publiques (Open API)

→ **Que se passe-t-il quand ces APIs sont mal conçues ?**

2. Définition d'une API ?



Qu'est-ce qu'une API ?

- Une API est une interface permettant à deux systèmes logiciels de communiquer

Standard dominant : REST

- Verbes HTTP : GET / POST / PUT / DELETE

L'API est une façade

- Elle masque la complexité interne et n'expose que ce qu'elle choisit.
- Ce rôle de contrat rend sa conception essentielle.

3. Problématique

Question centrale de cette étude

- La qualité de conception d'une API a-t-elle des conséquences qui dépassent le code ?
- Impact sur les développeurs ?
- Impact sur les équipes métier ?
- Impact sur la performance globale de l'entreprise ?

4. Les APIs : un élément critique du SI

Un élément critique du SI

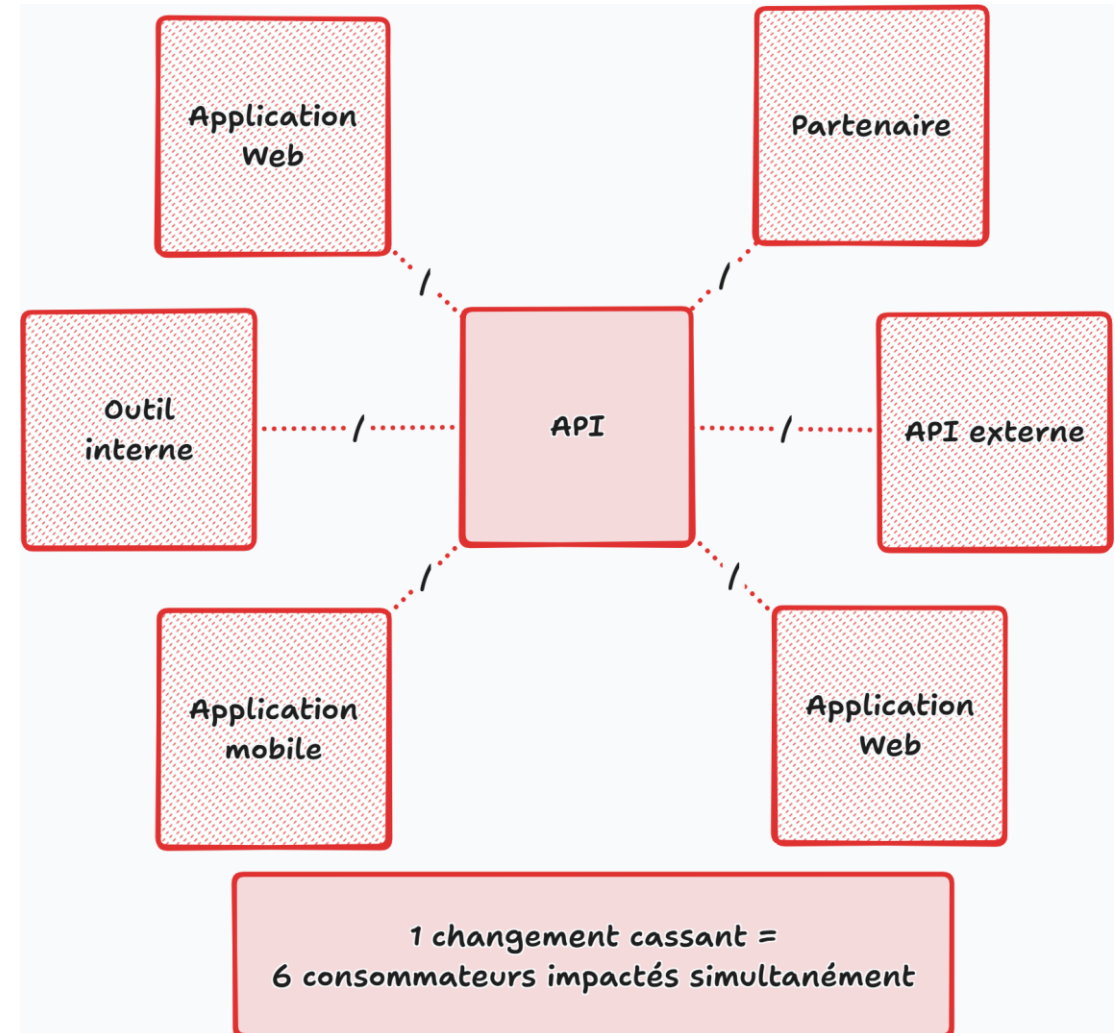
- Une API est appelée par de nombreux consommateurs simultanément.

L'effet domino

- Un changement cassant = tous les consommateurs brisés en même temps.
- Plus le SI est grand, plus l'impact est amplifié.

Chez Bpifrance

- Des dizaines de consommateurs différents d'une seule API
- Une mauvaise conception peut bloquer des integrations critiques.



5. Mauvaise conception : exemples concrets

Bonne vs Mauvaise conception : exemples Bpifrance

X Mauvaise conception

Nommage incoherent

GET /getEntreprise

GET /entreprise/chercher

GET /entreprises

=> 3 routes, meme action !

Pas de versioning

GET /api/entreprises (modifie sans preavis)

=> Tous les clients casses du jour au lendemain

URL = action (pas ressource)

POST /creerContrat

GET /supprimerContrat?id=1

VS

OK Bonne conception (Bpifrance)

Nommage coherent & pluriel

GET /v1/entreprises

GET /v1/entreprises?nom=X

POST /v1/entreprises

=> 1 ressource, verbes HTTP = actions

Versioning dans l'URL

GET /v1/entreprises (stable, maintenu)

GET /v2/entreprises (nouvelle version)

Preavis 3 a 6 mois avant suppression

URL = ressource, verbe = action

POST /v1/contrats

DELETE /v1/contrats/{id}

6. Impact côté développeurs

TEMPS

Comprendre une api mal documentée, déboguer, contourner les incohérences

Des heures non investies dans de vraies features

DETTE

Pour faire fonctionner une mauvaise API, des devs ajoutent des couches de code correctif.

Le système se fragilise à chaque version

MORAL

Une API est censée simplifier le travail. Quand c'est l'inverse, la qualité globale du travail se dégrade

Impact sur la productivité des équipes

Le coût invisible d'une mauvaise API se mesure en heures de développement perdues et en dette technique.

7. Impact côté métier et entreprise

TIME-TO-MARKET

- Chaque intégration difficile retarde les projets
- Des offres innovantes en retard = perte de compétitivité

DONNEES

- Une API incohérente = tableau métier incorrect = mauvaises décisions
- Conséquences sérieuses dans une banque

PARTENARIATS

- Fintechs, startups réglementaires consomment des API pour s'intégrer
- API difficile = intégration longue = partenaires qui abandonnent

Ce qui se passe dans le code impact directement les décisions business et l'image de l'entreprise.

8. Bonnes pratiques de conception

DESIGN FIRST

- Concevoir l'API avant de coder
- Penser aux consommateurs pas à l'implémentation interne

NOMMAGE & STRUCTURE

- Pluriel : /v1/entreprises
- Tirets : /v1/evenements-marquants
- Verbes HTTP = actions
- GET /v1/entreprises
- POST /v1/entreprises
- DELETE /v1/contrats/{id}

VERSIONING

- Version dans l'URL : /v1/ /v2/
- Max 2 versions simultanément
- Prévenir 3 à 6 mois avant la suppression d'une version
=> Aucun client cassé par surprise

DOCUMENTATION OpenAPI

- Standard Swagger / OpenAPI 3.0
- Documentation auto-générée et toujours à jour

Conclusion

Une API n'est pas qu'un détail d'implémentation

- C'est un contrat entre équipes, entre systèmes, et parfois entre organisations

Trois dimensions d'impact

- Technique : dette, bugs, maintenabilité
- Organisationnelle : productivité des équipes, time-to-market
- Stratégique : partenariats, image, compétitivité

Références

- Formation API Bpifrance — OCTO Academy, janvier 2025 (document interne, non public)
- Documentation interne Bpifrance — Norme API (document interne, non public)