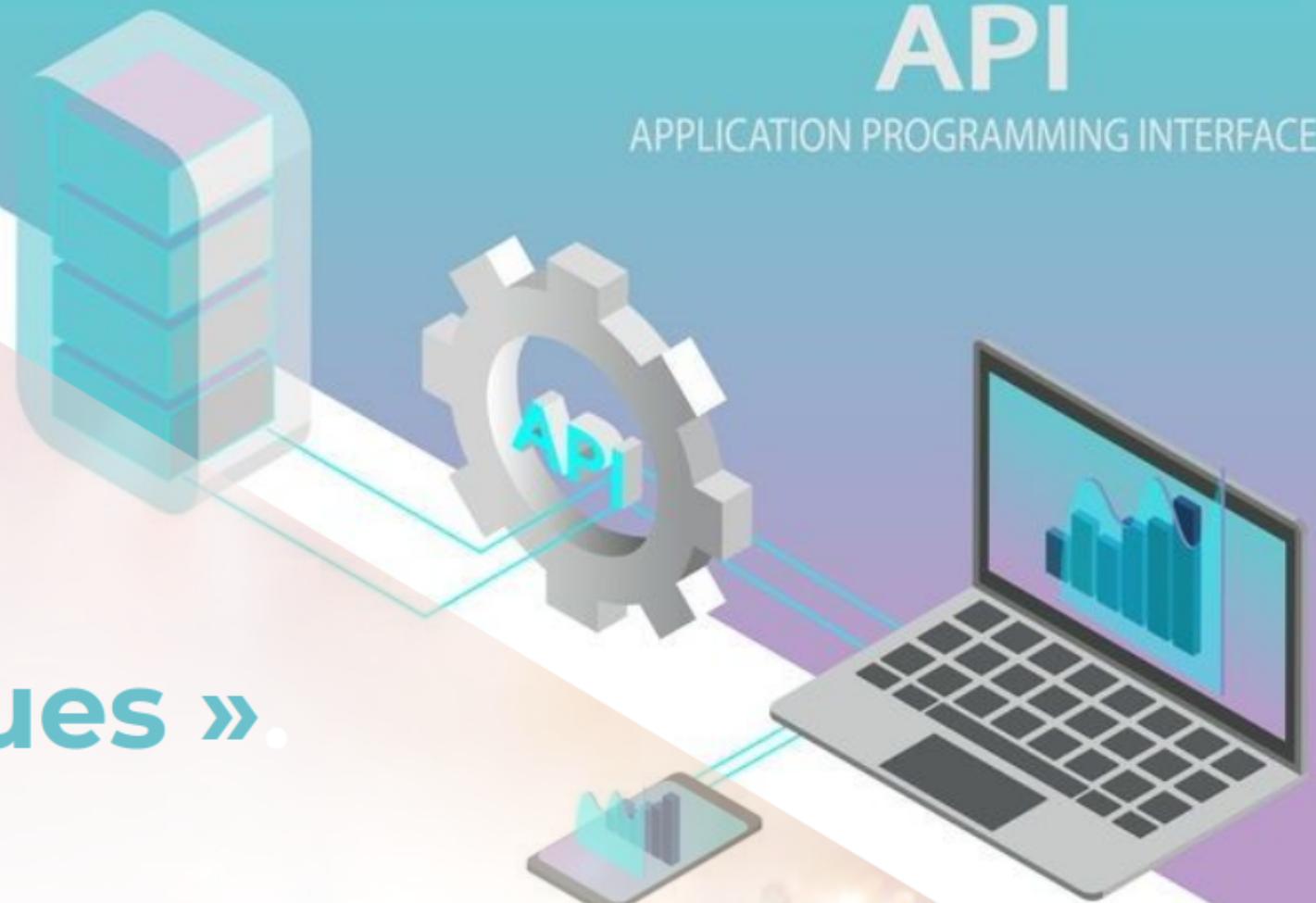


Epic Events



répondre aux besoins
des start-up voulant
organiser des « fêtes épiques ».

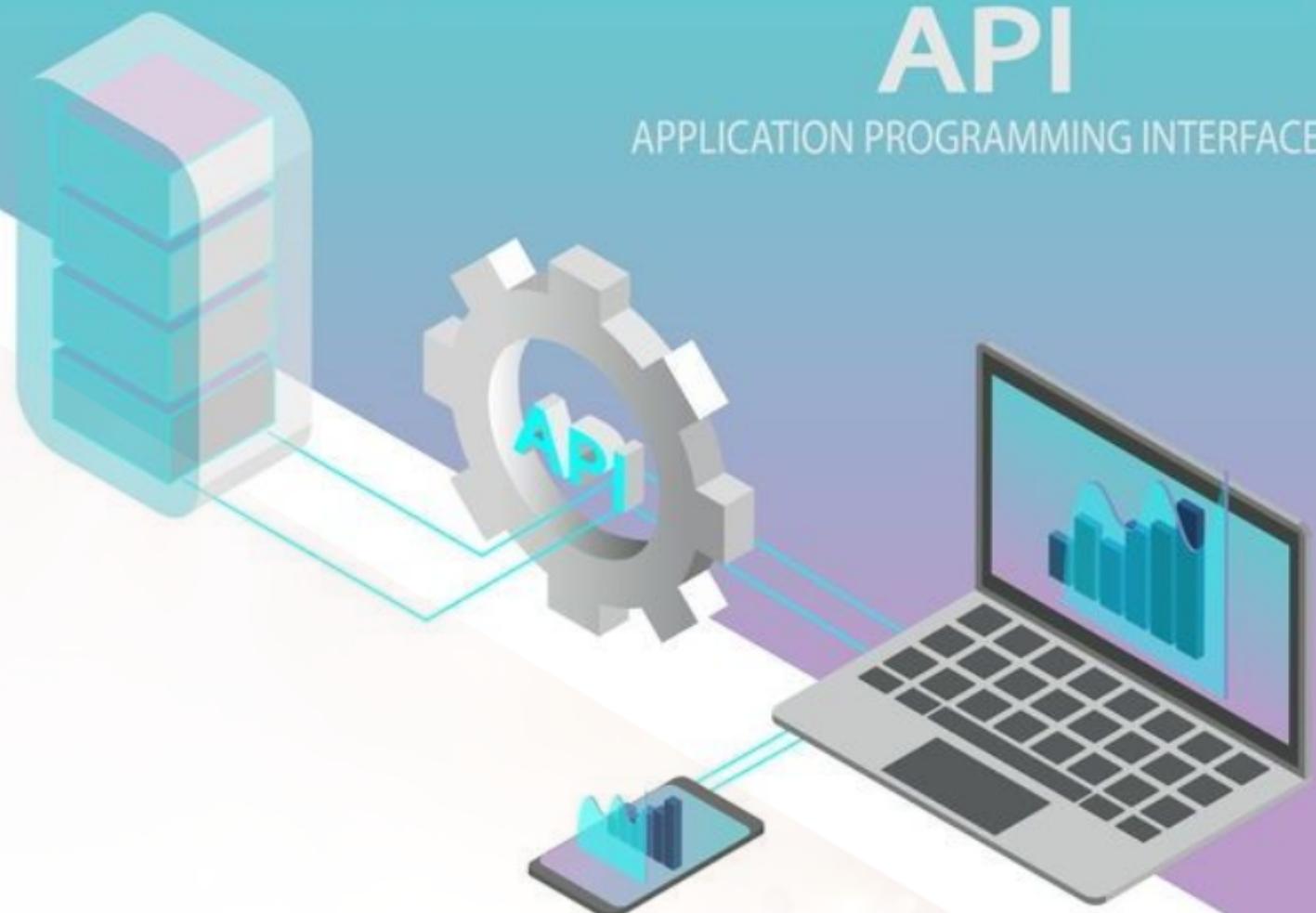


L'objectif

Élaborer un système CRM sécurisé interne à l'entreprise qui soit accessible via une API et une admin.



API
APPLICATION PROGRAMMING INTERFACE

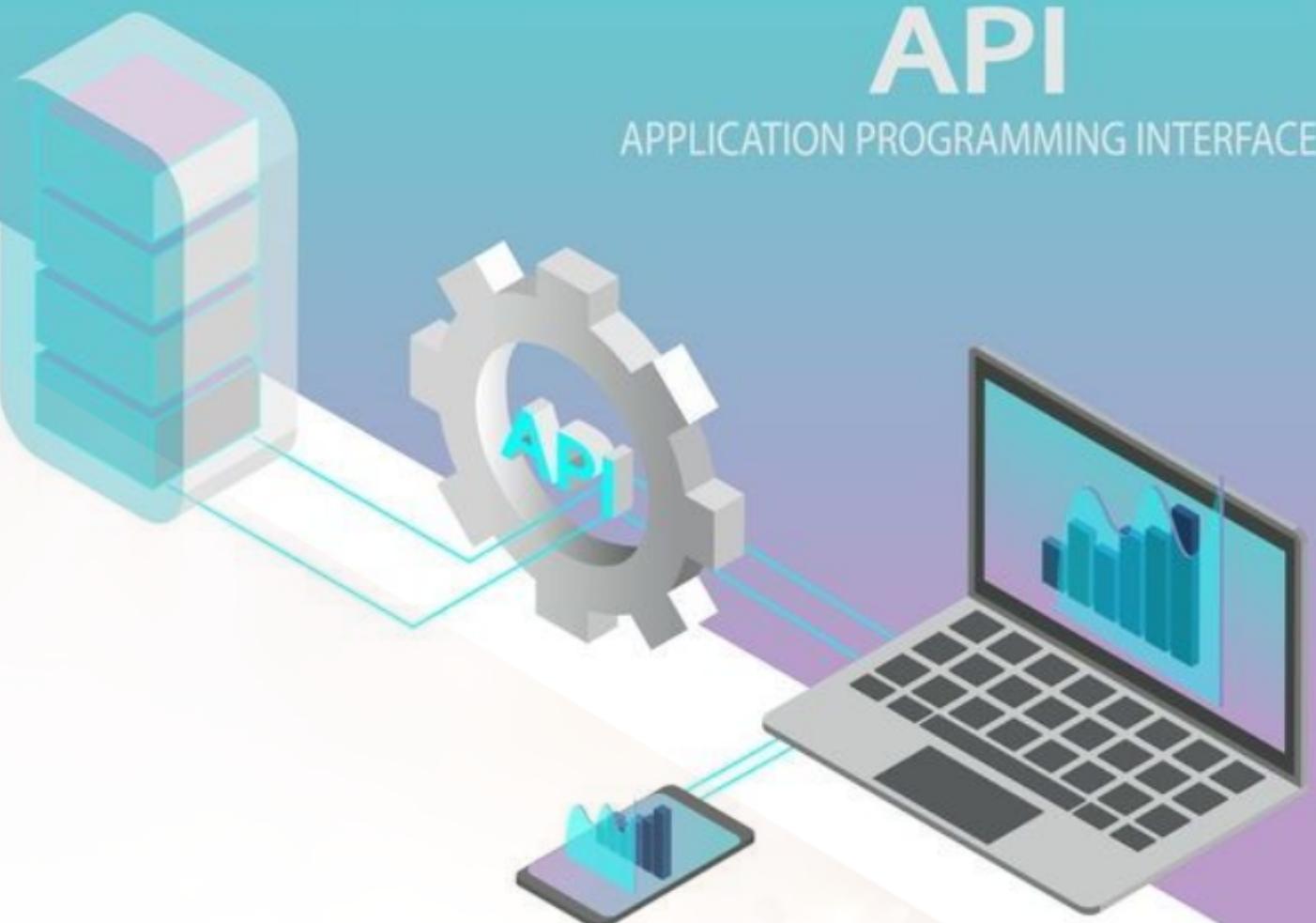


Les contraintes



- utiliser le Domain Driven Design pour établir un diagramme entité / relation (ERD) posant les bases
- identifier et créer les endpoints d'API correspondants
- mettre en place les opérations CRUD adaptées à chaque point de terminaison et verbe HTTP de l'API
- ajouter des filtres à l'API pour faciliter le travail des équipes
- permettre un accès via l'admin à l'équipe de gestion

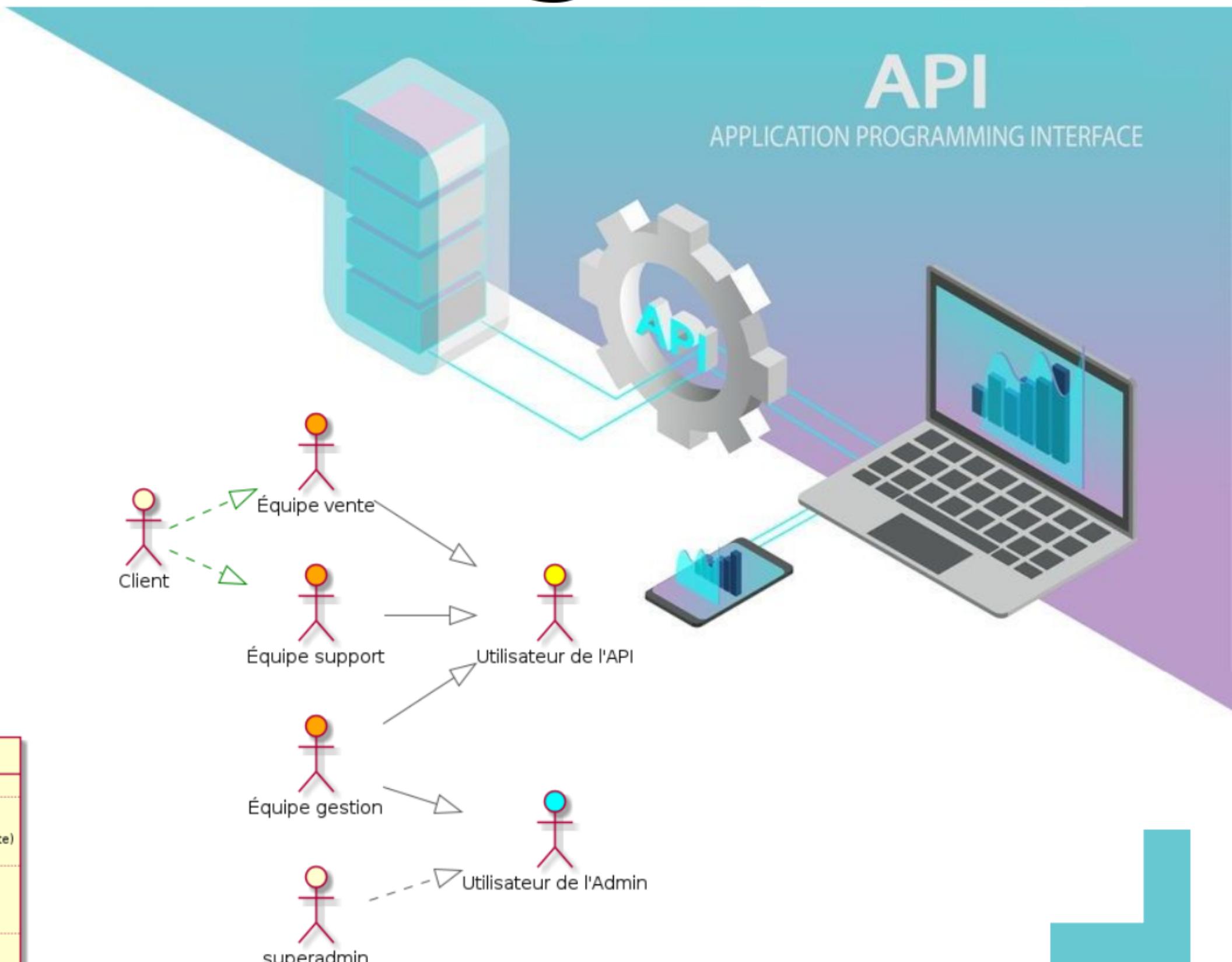
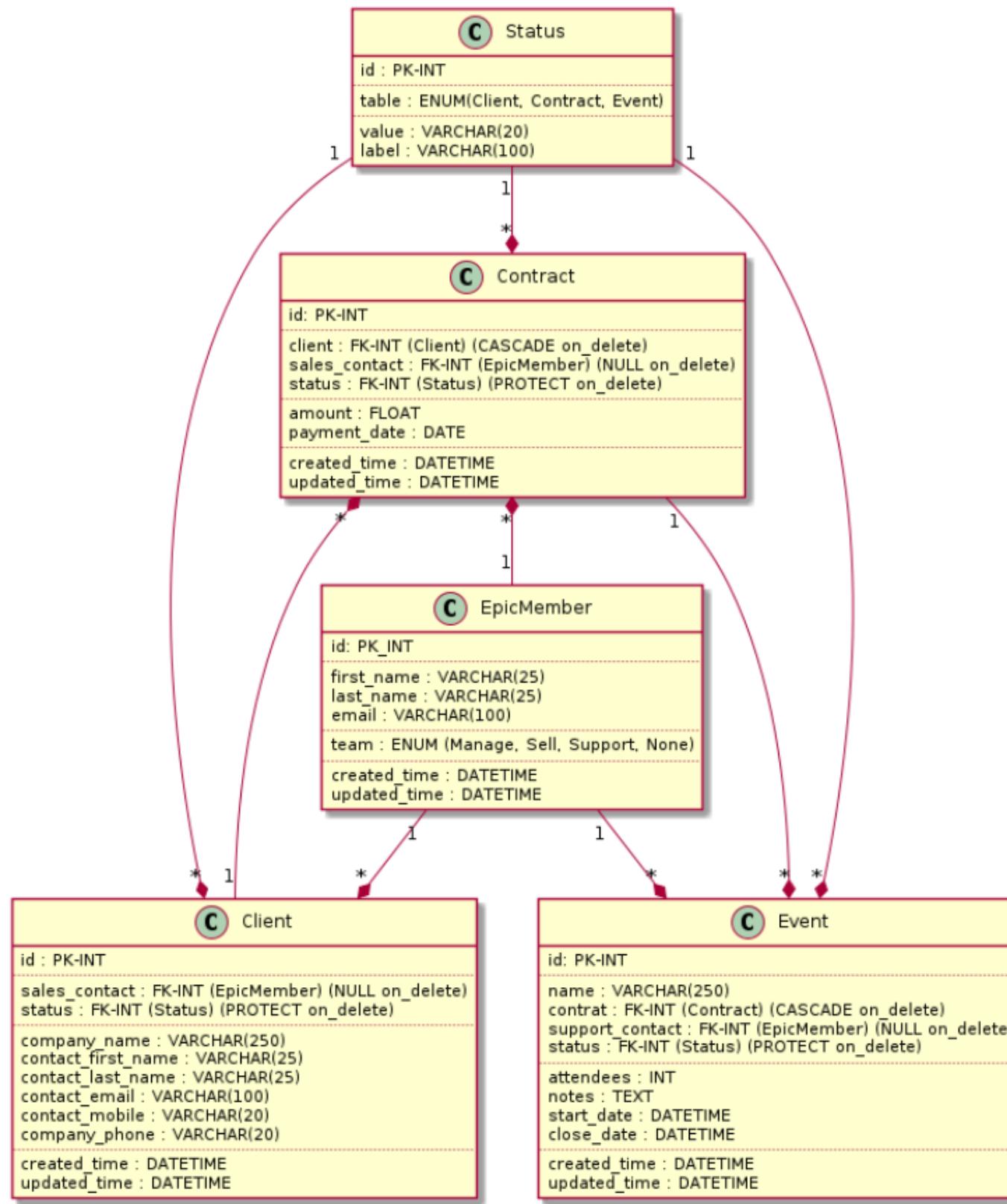
- utiliser Django (3.0+) & Django rest frameworks
- utiliser Postegre SQL (12.0+)
- incorporer les recommandation de sécurité de l'OWASP & RGPD
- respecter la PEP 8
- tester les endpoints d'API de l'application avec Postman



API

APPLICATION PROGRAMMING INTERFACE

Domain Driven Design



Démo front-end



127.0.0.1:8000

Postman / curl / browser

**demo-login : manage_user
sales_user
support_user
noteam_user**

demo-password : demopass



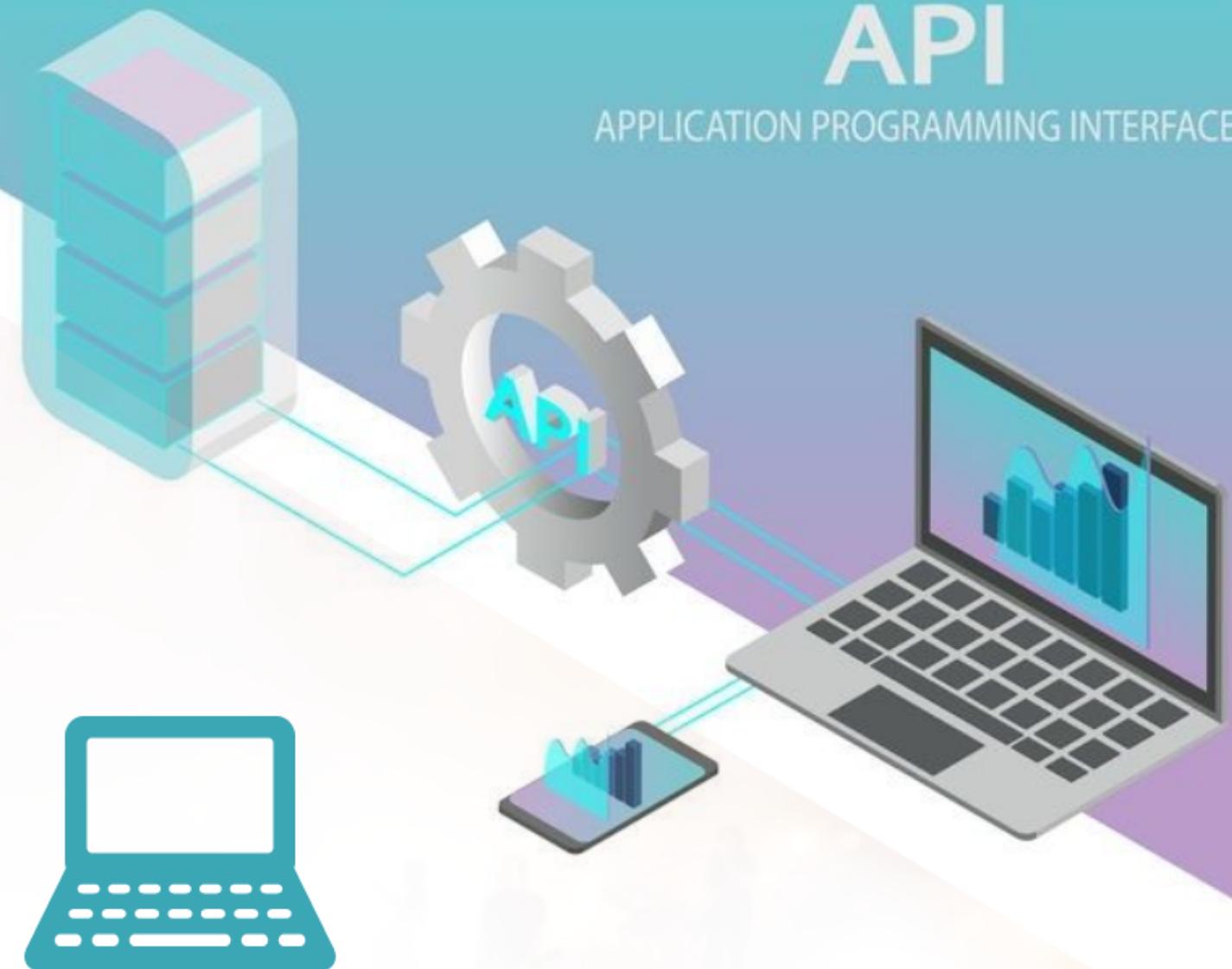
Démo back-end



127.0.0.1:8000/zadmin

demo-login : `manage_user`
`sales_user`
`support_user`
`noteam_user`
`epicadmin`

demo-password : `demopass`



- minimiser les informations demandées aux utilisateurs

- traiter les vulnérabilités regroupées selon le processus AAA (Authentication, Authorization, Accounting)

- Authentification > identifier l'utilisateur
on utilise JWT pour confirmer l'identité sans surcharger la DB

- Authorization > donner les accès CRUD selon la vue ou l'objet
on utilise les classes de permissions pour définir les accès aux vues

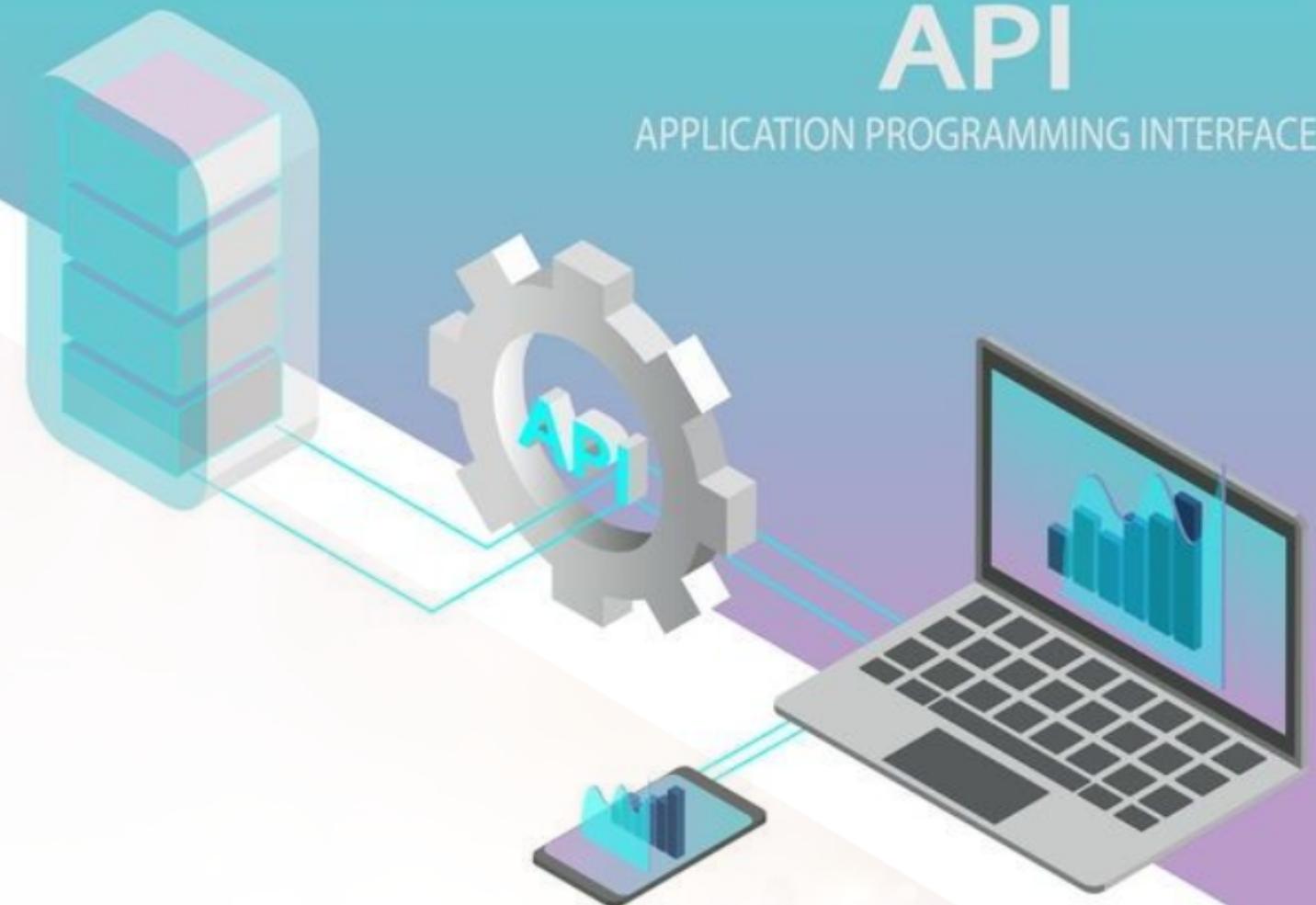
- Accounting > donner les accès CRUD selon les permissions liées à l'utilisateur
on utilise un paramètre lié au compte utilisateur pour définir ce qu'il peut faire



Documentation



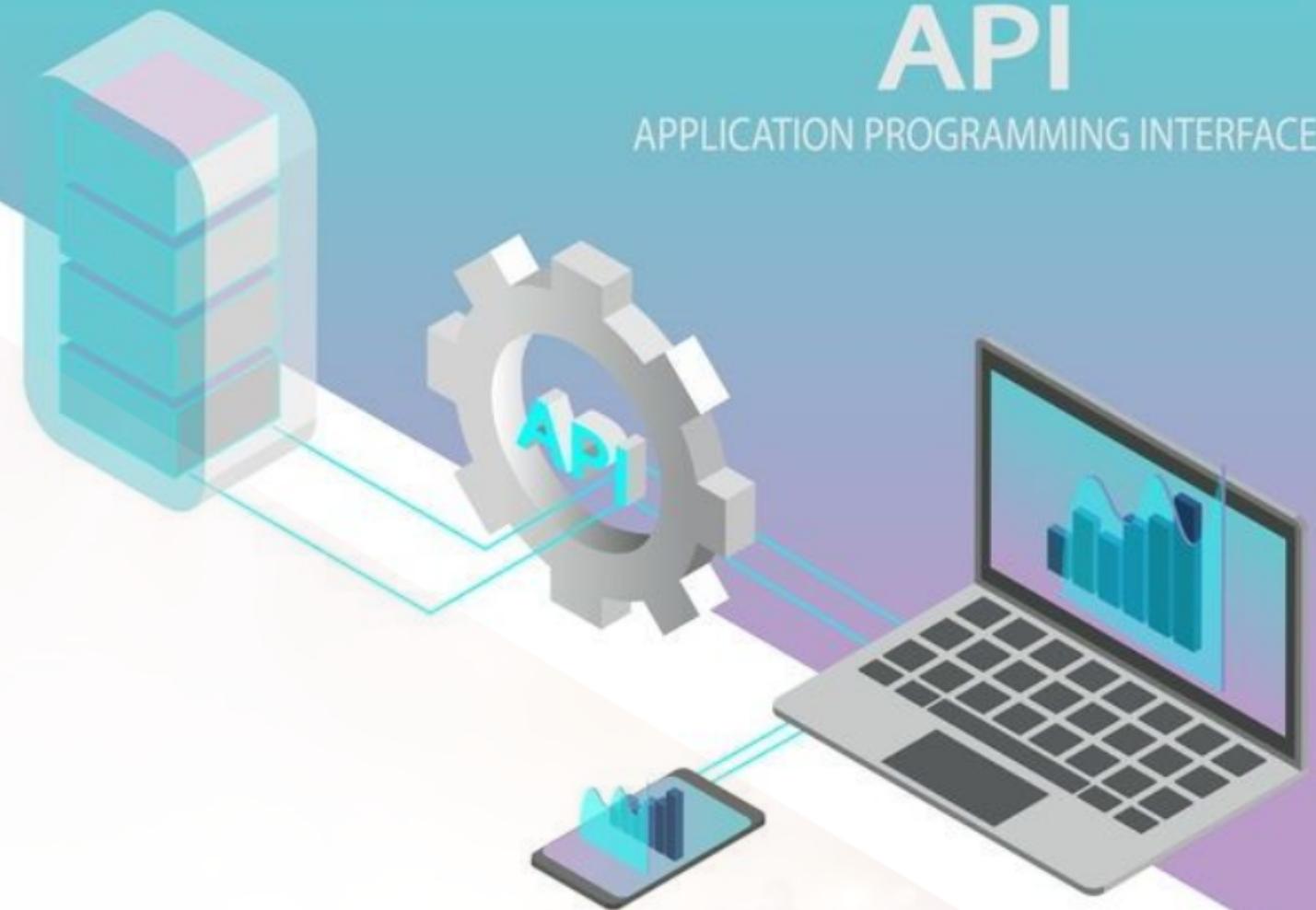
README.md
Postman API
documentation



Bonnes pratiques



- **docstrings** (admindoc Django)
- **PEP8** (black & vim-flake8)
- **README.md**
- **Documentation Postman**
- **Recommendations OWASP**
- **Tests** (Postman & unittest)
- **Coverage**



Merci

de m'avoir écouté,
évalué et conseillé



API

APPLICATION PROGRAMMING INTERFACE

