WAVESTONE

GS Days
LES JOURNÉES FRANCOPHONES DE LA SÉCURITÉ

La sécurité des API Web

… ou la recette du bon miel

28 mars 2017

Gérôme BILLOIS
gerome.billois@wavestone.com
@gbillois

Bertrand CARLIER
bertrand.carlier@wavestone.com
@bertrandcarlier

Dans un monde où la capacité à se transformer est la clé du succès, nous éclairons et guidons nos clients dans leurs décisions les plus stratégiques

Des clients leaders dans leur secteur

2,500 collaborateurs sur 4 continents

Parmi les leaders du conseil indépendant en Europe, n°1 en France

Paris | Londres | New York | Hong Kong | Singapour* | Dubaï*
Bruxelles | Luxembourg | Genève | Casablanca
Lyon | Marseille | Nantes

* Partenariats

Réussir sa **transformation numérique** grâce à la **confiance numérique**

400+
Consultants & Experts

1,000+
Missions par an dans plus de **20** pays

Nos clients
COMEX, Métier, CDO, CIO, CISO, BCM

UNE EXPERTISE EPROUVEE

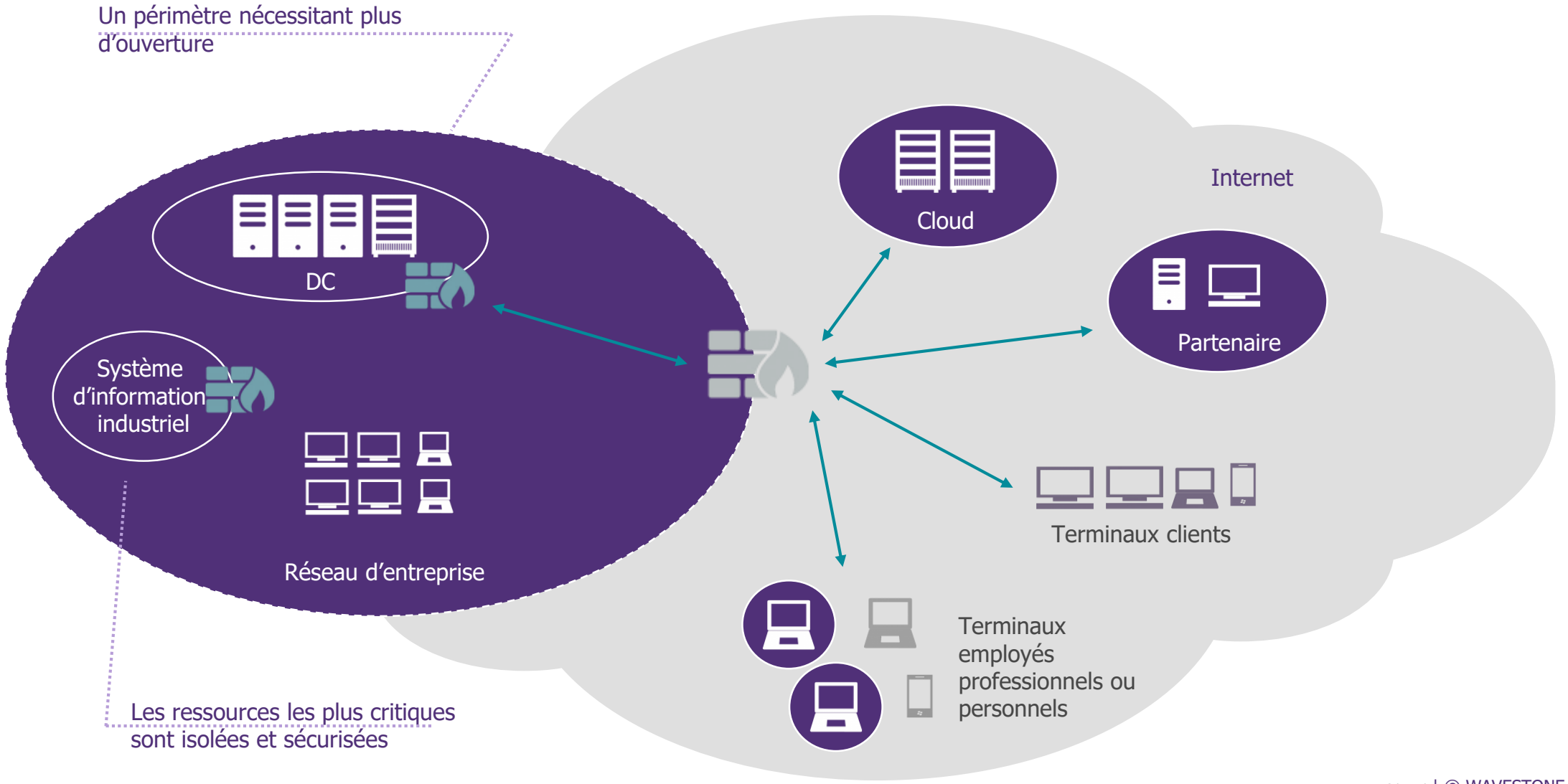
- Stratégie et Conformité
- Transformation métier sécurisée
- Architecture et programme sécurité
- Identité, Fraude et Services de Confiance
- Tests d'intrusion & Réponse à incident
- Continuité d'Activité & Résilience
- SI Industriel

NOS DIFFERENCIATEURS

- Connaissance des risques métier
- Méthodologie AMT pour les schémas directeurs
- Radars Innovation et Start-ups
- CERT-W
- Bug Bounty by Wavestone

Mais pourquoi tout le monde parle-t-il d'API Web ?

2005-2015 : une ouverture sous contrôle du système d'information



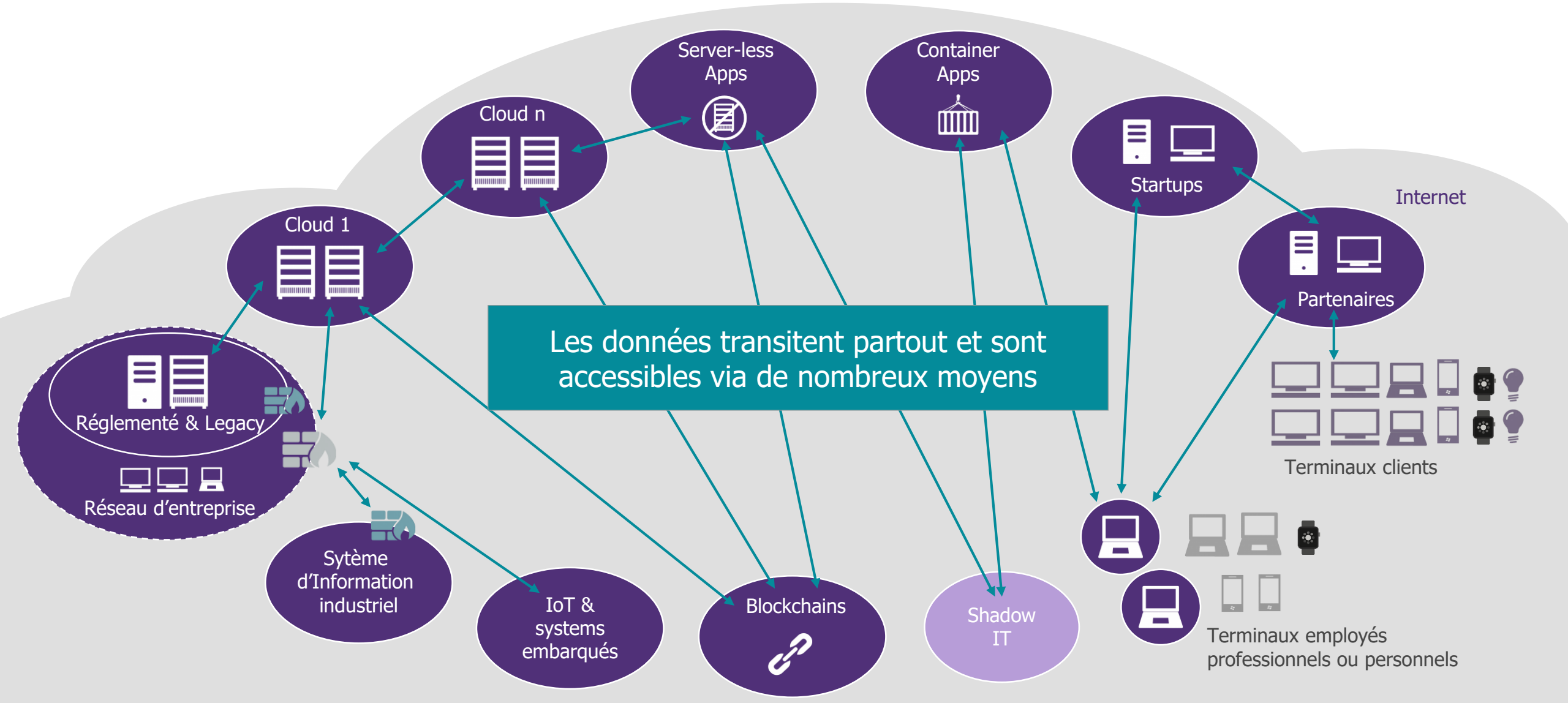
En 2020

Le Cloud est une réalité, y compris pour des applications critiques métier

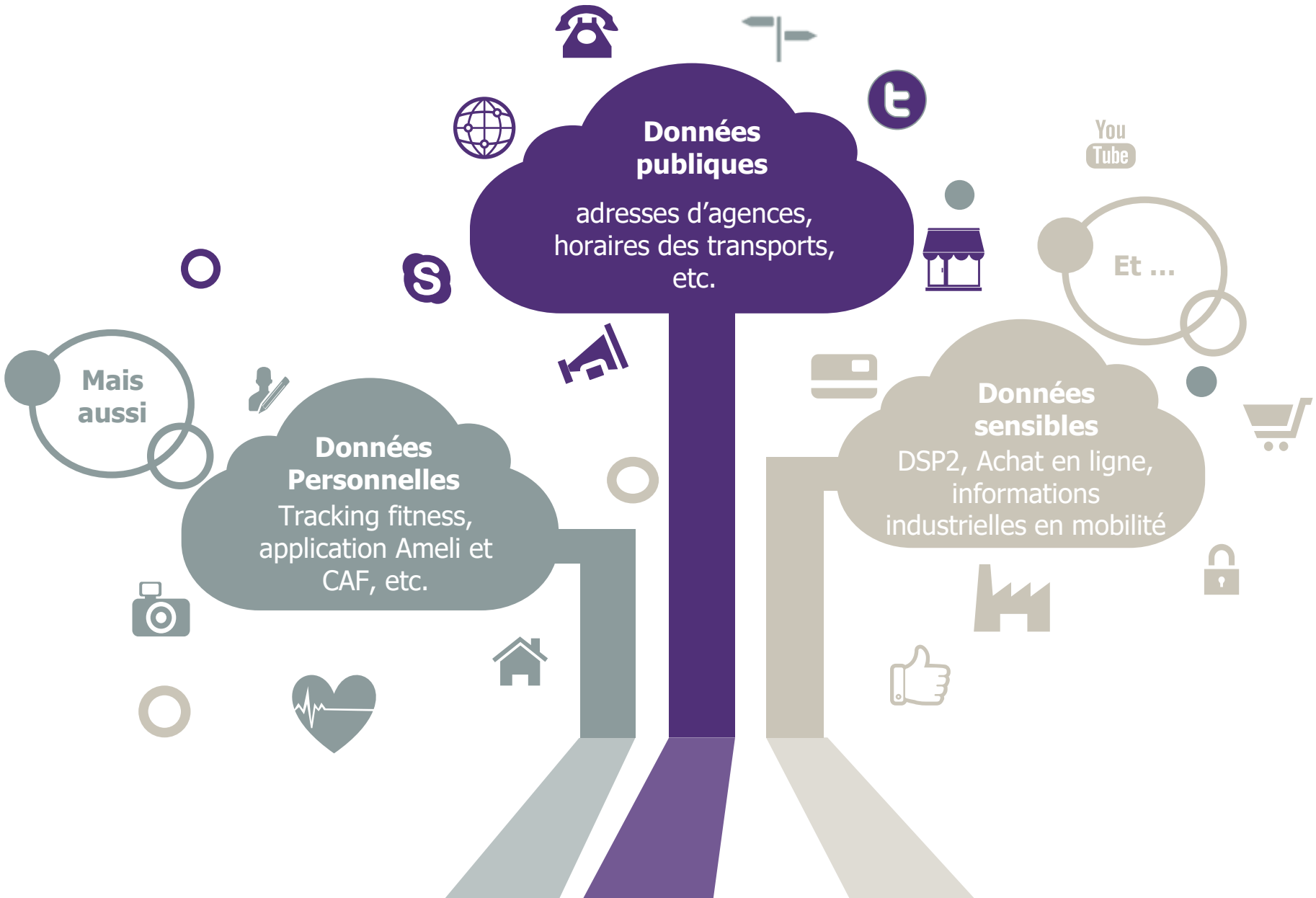
Les méthodes Agile et DevOps sont d'ores et déjà répandues dans l'entreprise

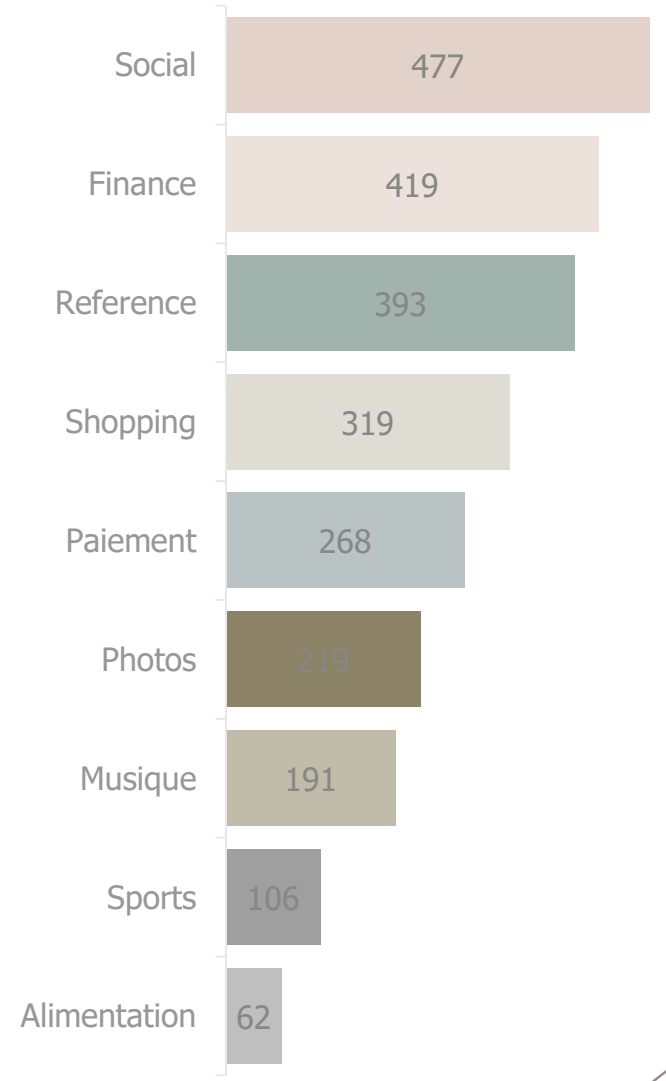
Les méthodes et pratiques de sécurité actuelles ne suivent plus le rythme de ce nouveau modèle opérationnel

2020 : Un système d'information décentralisé



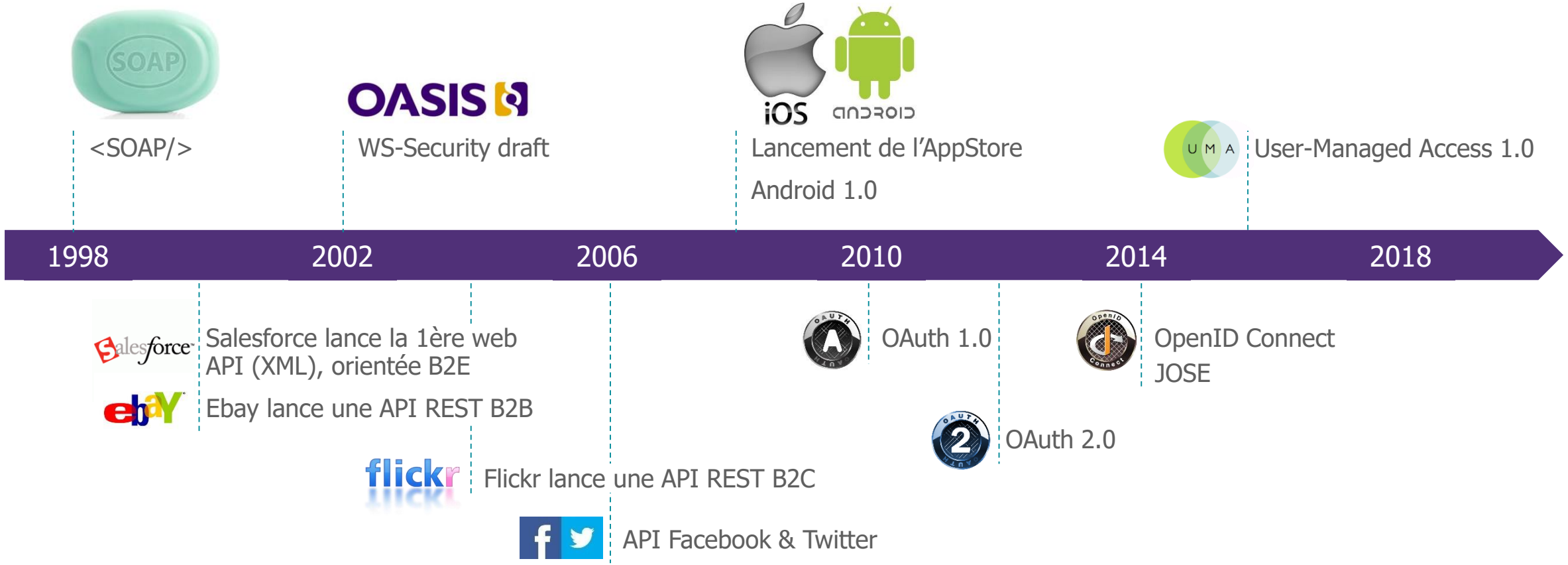
Les APIs Aujourd'hui



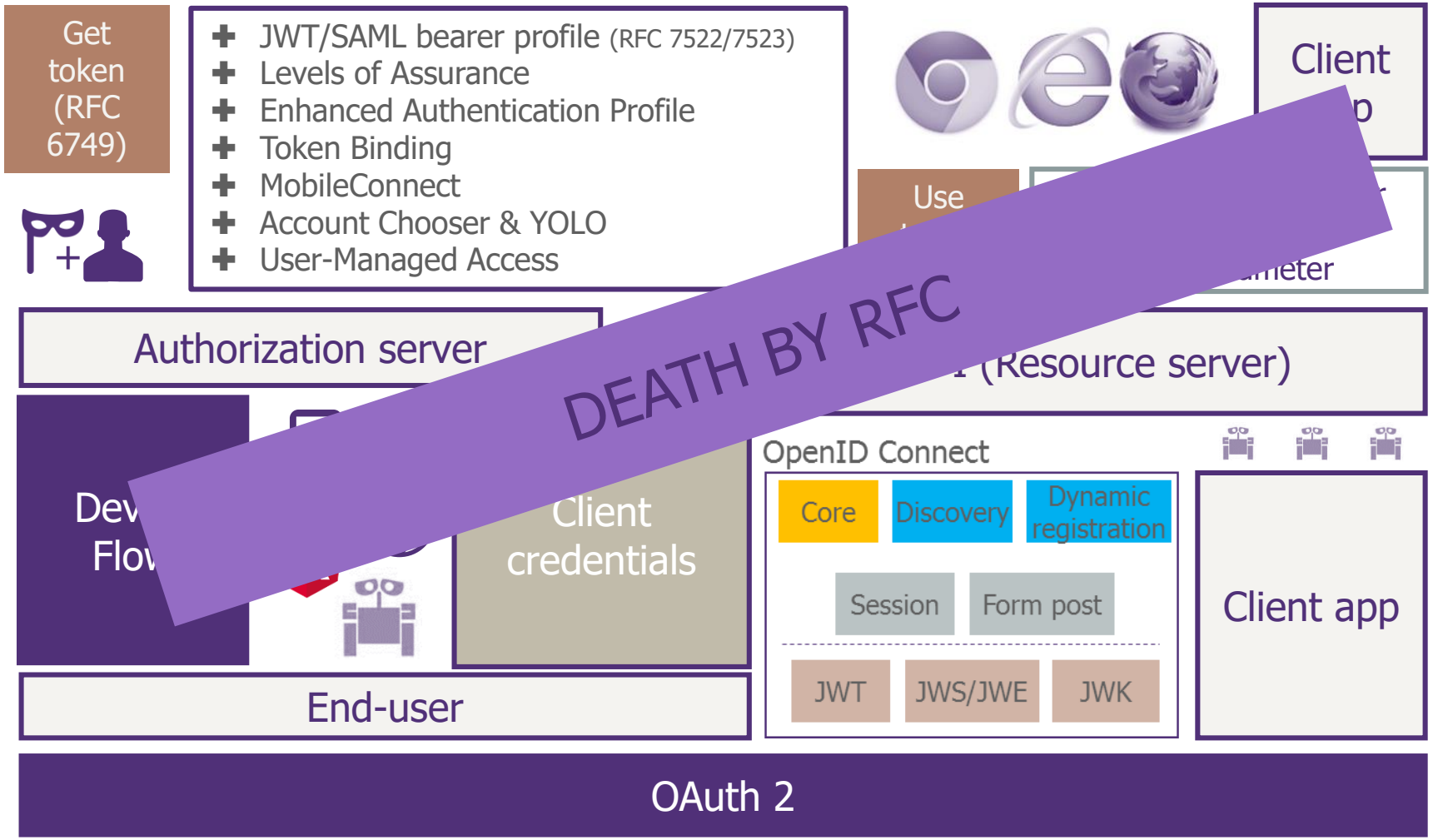


APIs par catégories sur theprogrammableweb.com

API Web, 20 ans déjà!



Les API c'est très simple…



Toute ressemblance avec des personnes ou des situations existantes ou ayant existé

…par exemple lors d'une réunion avec des architectes…

ne saurait être que fortuite.

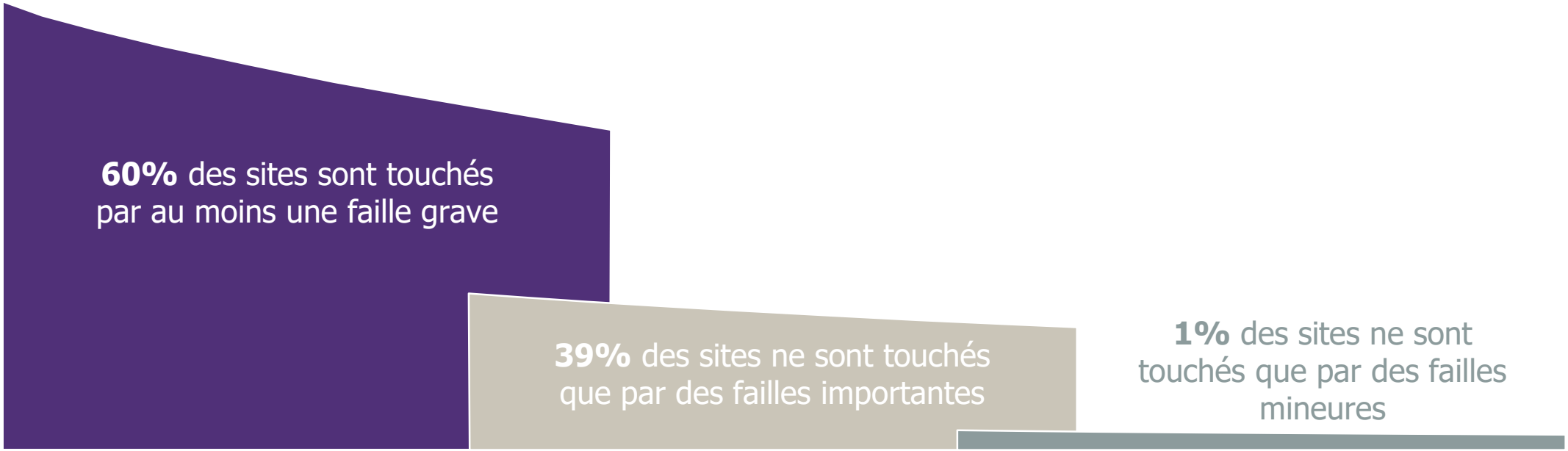
Quelle recette pour sécuriser ses API ?

La recette pour des APIs sécurisées

Une base de *Security as usual*

API & Applications web – *Security as usual*

Benchmark Wavestone : sur 128 sites audités en un an, de nombreuses failles…
… une situation très (très) similaire pour les API



Faille grave
Permet d'accéder à l'ensemble du contenu du site et/ou de compromettre les serveurs

Accès à l'ensemble des données du site, exécution de code par le serveur, utilisateur A ayant accès aux données de B, etc.

Faille importante
Permet d'accéder aux informations d'autres utilisateurs mais en nombre limité ou de manière complexe

Vol de session d'un utilisateur, faiblesses dans le chiffrement, possibilité de faire réaliser des actions à l'insu de l'utilisateur, etc.

Faille mineure
Permet principalement d'obtenir des informations pour continuer l'attaque

Messages techniques superflus, absence de sécurisation des cookies, déconnexion utilisateur non efficace, etc.

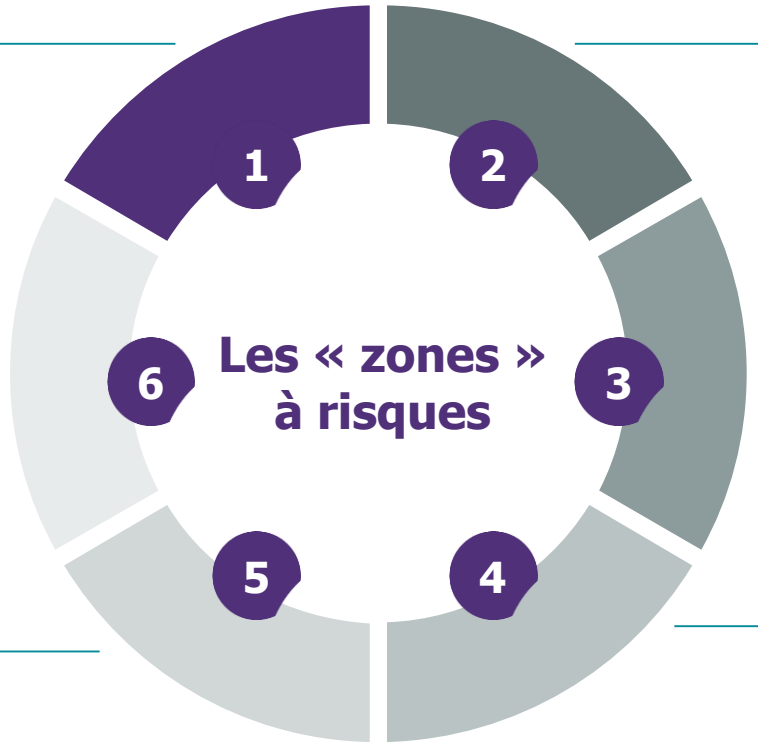
API & Applications web – *Security as usual*

N’oublions pas les recommandations de bases de la sécurité web…

Gestion des sessions
Authentification & Maintien de session
Côté client vs côté serveur
Identifiant de sessions non devinables
Ré-authentifier pour des actions critiques

Contrôle d’accès
Gestion des profils & privilèges
Situation de concurrence
Séparation des espaces utilisateurs

Gestion des entrées/sorties
Traitement des entrées
Accès aux ressources et traitement
Injection de données côté client
Encoder les données avant réponse



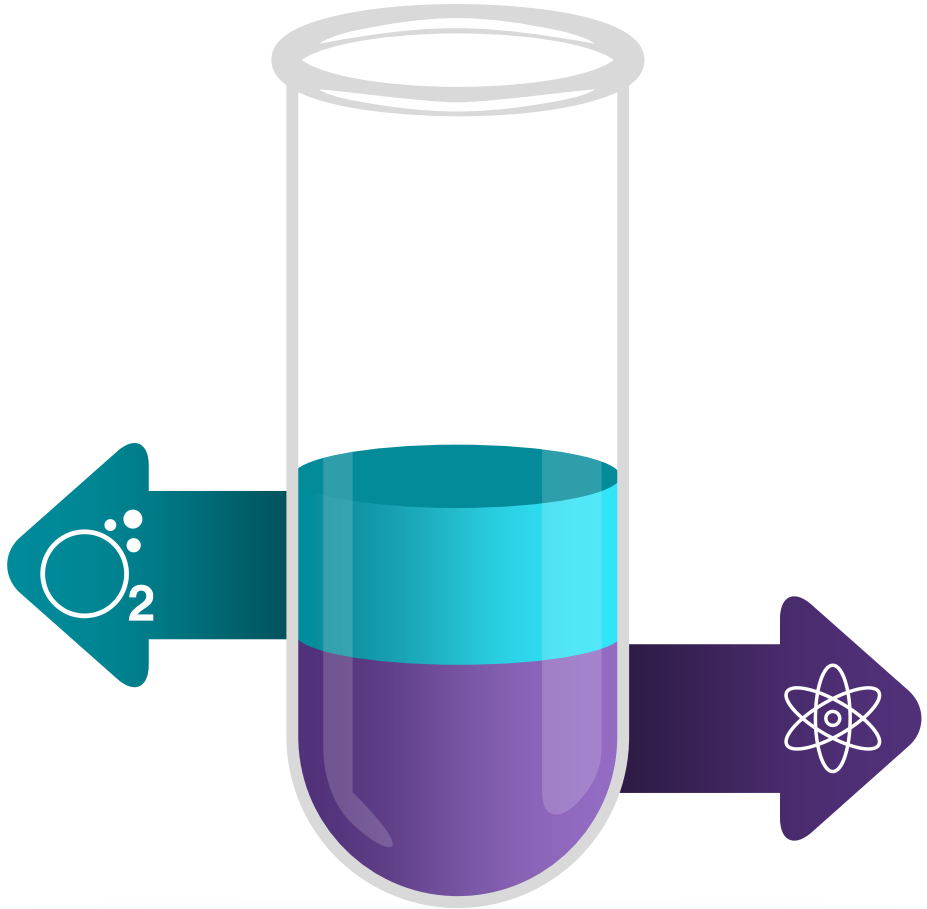
Données sensibles
Séparation des environnements
Stockage et gestion des secrets
Faire usage de mécanismes de sécurité éprouvés

Gestion des exceptions
Gestion des erreurs
Journalisation
Intercepter toutes les erreurs et les traiter

Gestion de la mémoire
Allocation de la mémoire
Initialisation des objets et des variables
Supervision consommation mémoire

La recette pour des APIs sécurisées

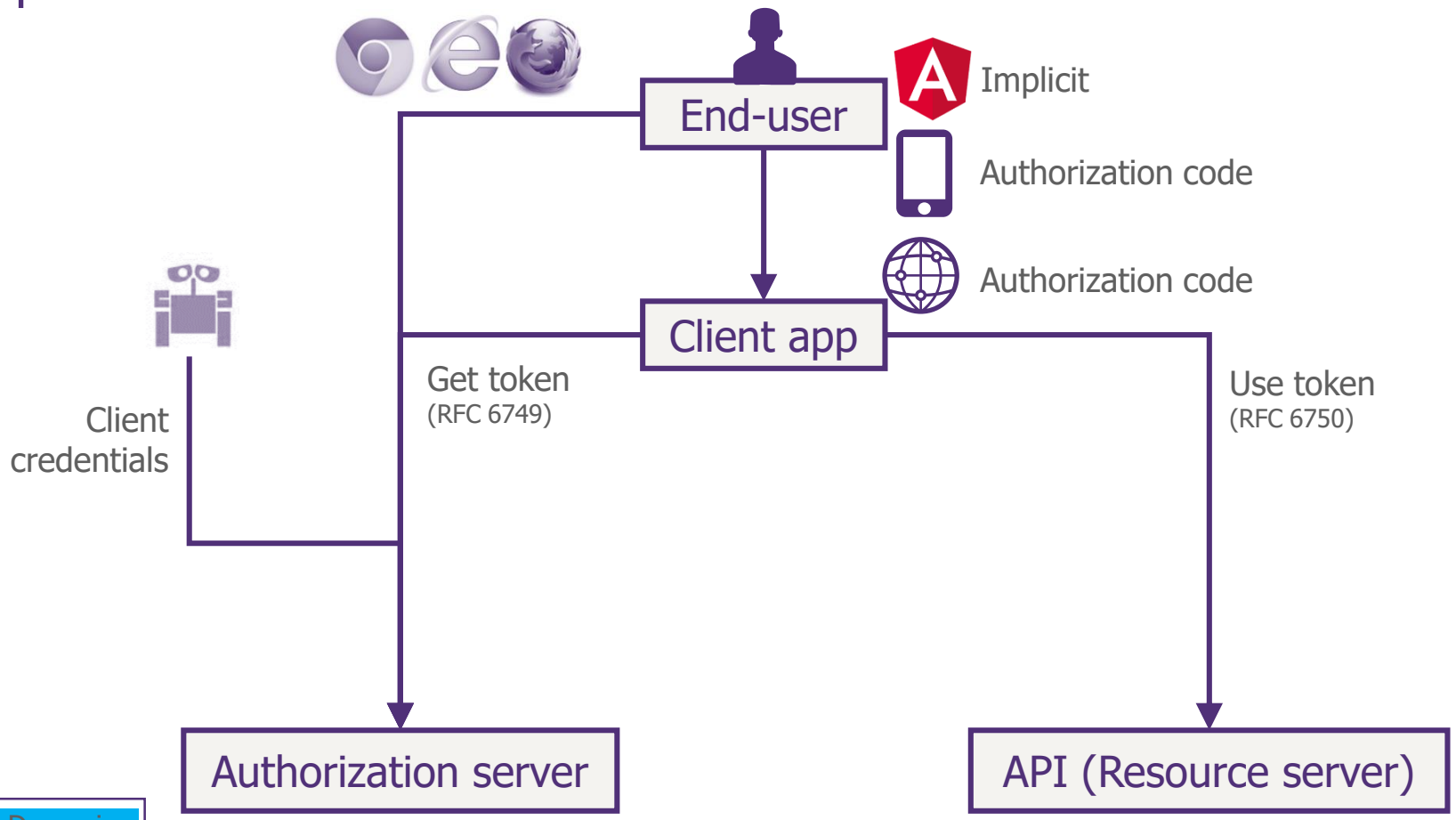
Une pincée d'OAuth



Une base de *Security as usual*

Une API est une application web

Commencer par l'essentiel d'OAuth2

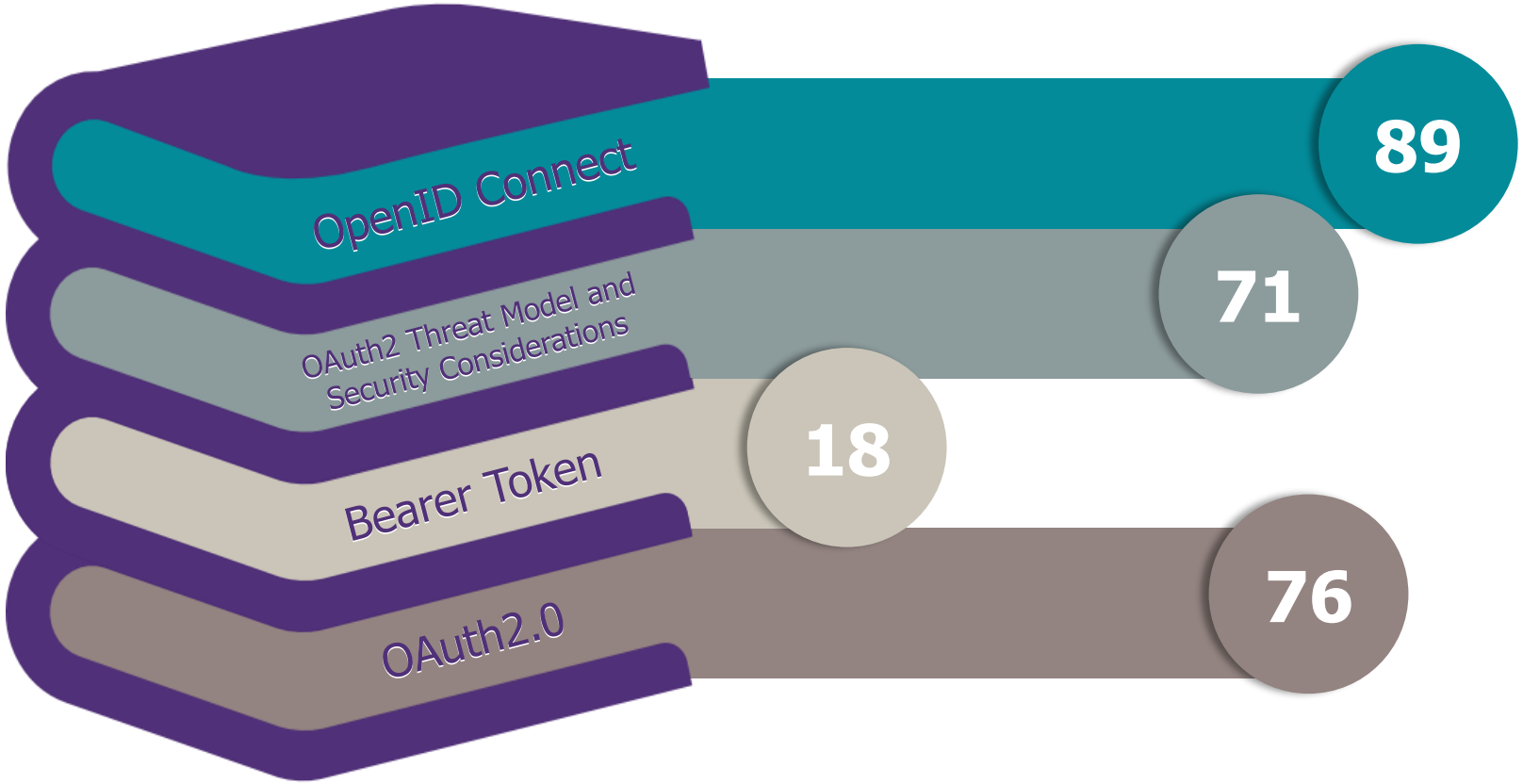


OpenID Connect

Core | Discovery | Dynamic registration

OAuth2.0 : un corpus documentaire très touffu

Lors de l'implémentation d'Oauth ou l'utilisation de framework, une erreur est vite arrivée…



Les failles les plus courantes

- Usurpation d'identité d'une application
- Accès aux données personnelles d'un utilisateur tiers
- Vol de cookie Facebook/Google lors d'un *social login*
- Compromission de compte utilisateur

Les six recommandations essentielles

1 **Secret local**

Ne pas mettre de secret dans l'application mobile ou le considérer compromis

2 **Redirect URI**

Valider strictement les URLs de redirection, sans wildcard

3 **Implicit**

Éviter le *Implicit grant* dans la mesure du possible (*proxy pattern*)

4 **Authz code**

Valider strictement les authorization code et clients associés

5 **state & PKCE**

À utiliser pour garantir l'intégrité d'une cinématique complète

6 **Authz ≠ Authn**

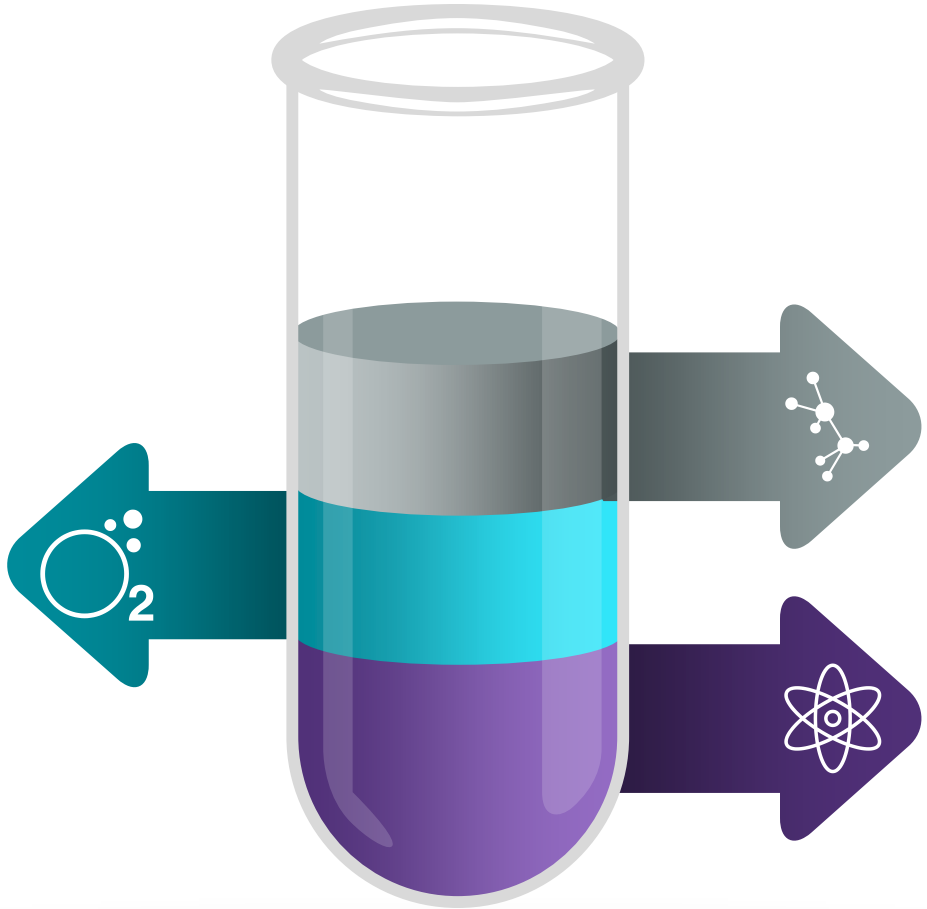
Utiliser OpenID Connect pour authentifier, OAuth pour déléguer

La recette pour des APIs sécurisées

Une pincée d'OAuth

Les pièges à éviter

Les recommandations essentielles



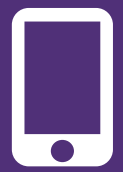
Limitez les additifs

Une base de *Security as usual*

Une API est une application web

Les besoins additionnels rencontrés chez nos clients

Des sous-standards, du plus mature au plus expérimental, à utiliser avec parcimonie !
Et en évitant les solutions « fait-maison »…



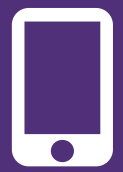
Single Sign-On Mobile

Authentification contextuelle

Propagation de l'identité

Protection contre le vol de jeton

Les besoins additionnels rencontrés chez nos clients



Single Sign-On Mobile

Navigation sans couture entre application natives mobiles et vers ou depuis le navigateur du terminal

Authentification contextuelle

Ajustement du niveau d'authentification en fonction de la sensibilité de l'API consommée

Propagation de l'identité

Transmission de l'identité de l'utilisateur aux APIs backend et validation de la chaîne d'appels

Protection contre le vol de jeton

Rendre inopérant le vol d'un jeton de sécurité et ne pas dépendre de la sécurité d'un terminal non maîtrisé ou d'un tiers

Single Sign-On mobile : le besoin

Agent de maintenance en tournée d'intervention

Plusieurs applications mobiles (gestion de tournée, rapports d'intervention, documentations techniques, etc.) spécifique & **applications collaboratives Microsoft**

Agent sur le terrain, en contact clientèle

Jusqu'à **14 applications utilisées quotidiennement**. Certaines sont globales à l'entreprise (100k salaries), d'autres spécifiques à son métier

Employés en déplacement international

Équipés de tablettes, besoins d'accès à des **applications mobiles et web avec authentification forte**

Un besoin récurrent, pour des populations employés ou clients

Depuis 2008, les techniques pour obtenir du SSO mobile ont varié au gré des possibilités des OS mobiles:

- Keychain iOS
- Paramètres URL
- Mobile Device Management

En 2015, Apple et Google convergent vers une solution commune : utiliser le navigateur système comme point d'ancrage d'une session SSO

Single Sign-On mobile : la solution

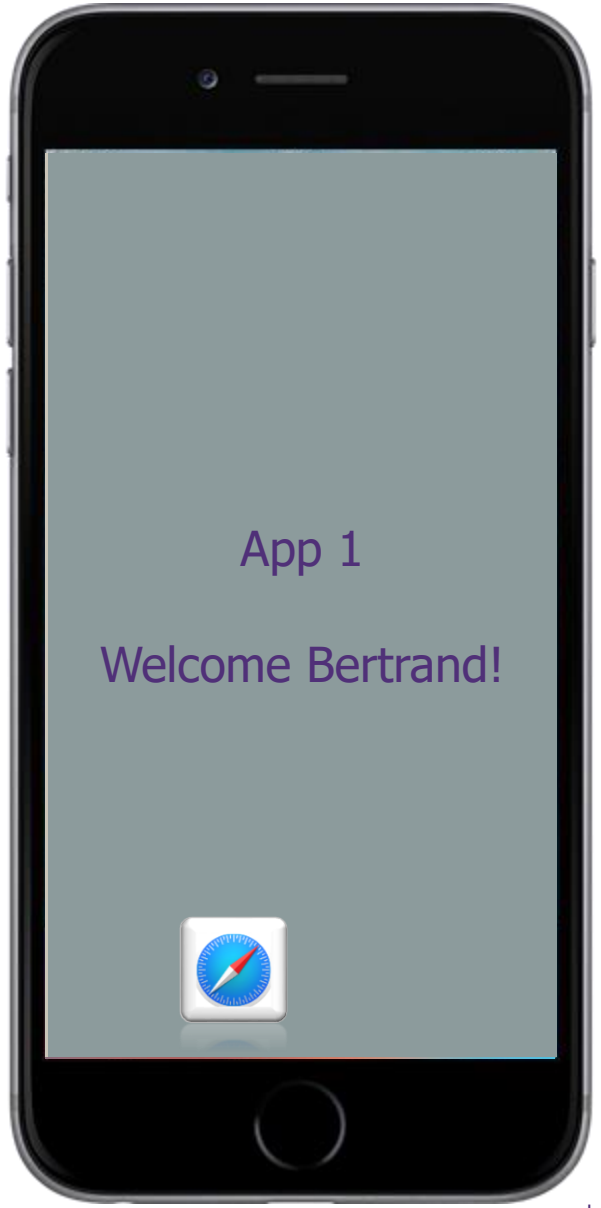
Best Current Practice : OAuth 2.0 for Native Applications

- Utilisation des composants iOS BrowserView & Chrome CustomTabs, liés au navigateur système
- Utilisation du flow authorization code de OAuth 2.0
- Permet un SSO app/app et app/web

Fonctionne entre applications maîtrisées et développées par des tiers

AppAuth

Un SDK disponible pour iOS et Android pour accélérer le développement



Les besoins additionnels rencontrés chez nos clients

Single Sign-On Mobile

Navigation sans couture entre application natives mobiles et vers ou depuis le navigateur du terminal

Authentification contextuelle

Ajustement du niveau d'authentification en fonction de la sensibilité de l'API consommée

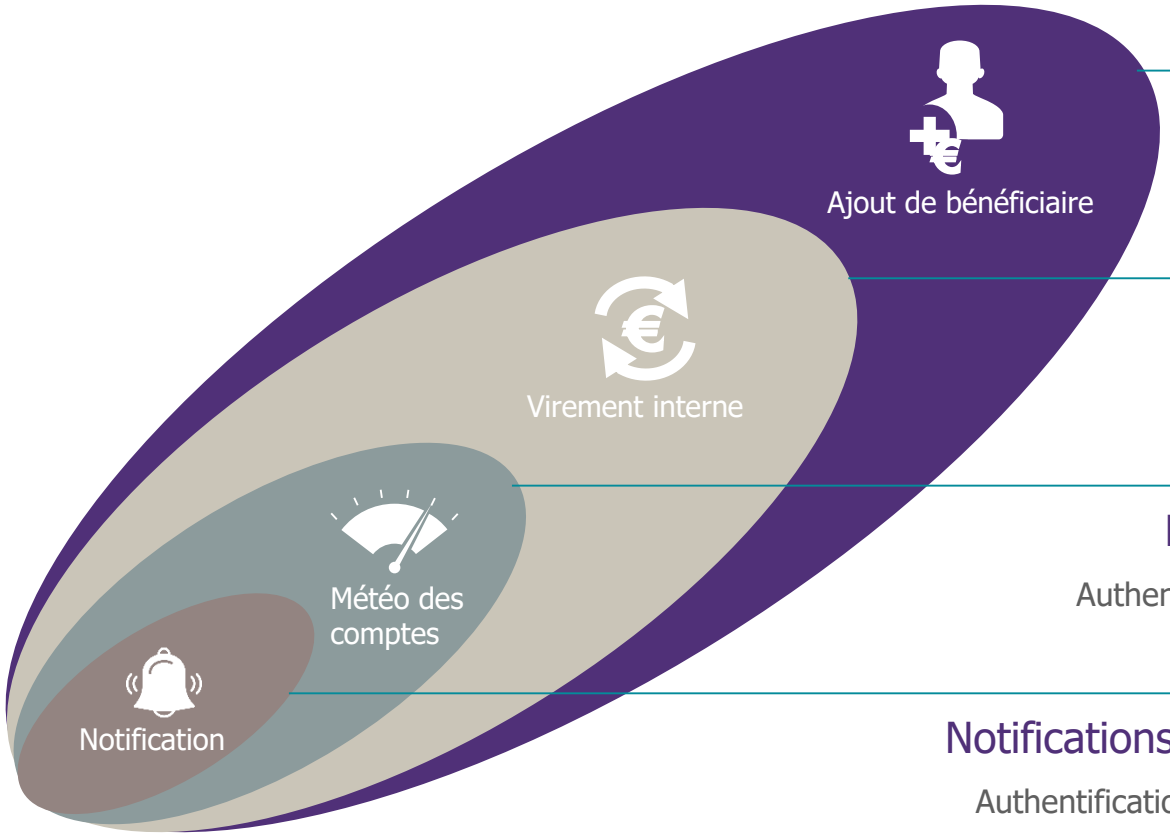
Propagation de l'identité

Transmission de l'identité de l'utilisateur aux APIs backend et validation de la chaîne d'appels

Protection contre le vol de jeton

Rendre inopérant le vol d'un jeton de sécurité et ne pas dépendre de la sécurité d'un terminal non maîtrisé ou d'un tiers

Authentification contextuelle : un exemple dans le secteur bancaire



Opération très sensible

Authentification forte / transaction

Opération sensible

Authentification simple / session

Données personnelles

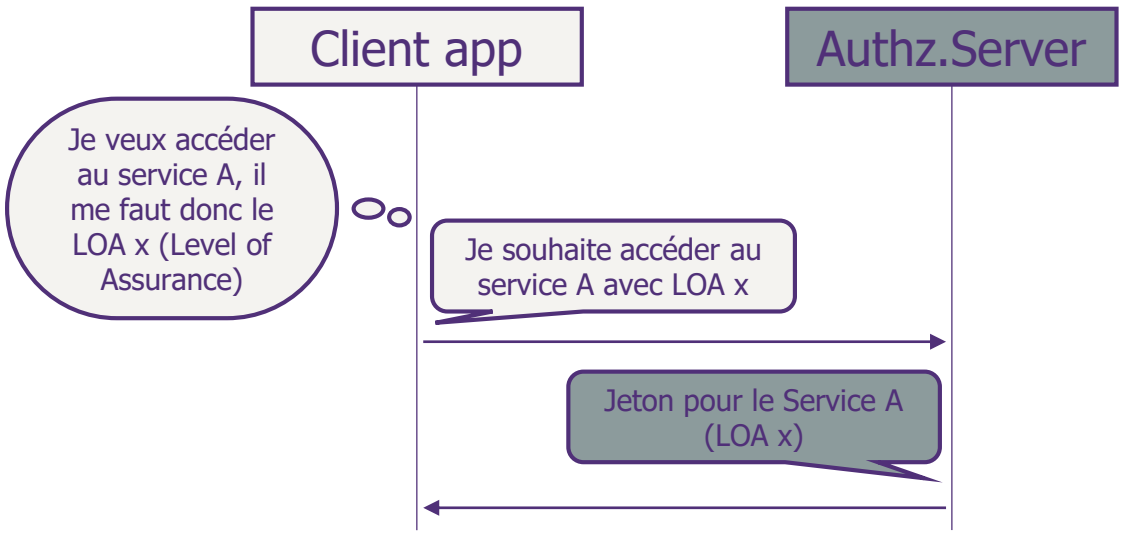
Authentification simple / 1 semaine

Notifications personnelles

Authentification simple / 1 mois

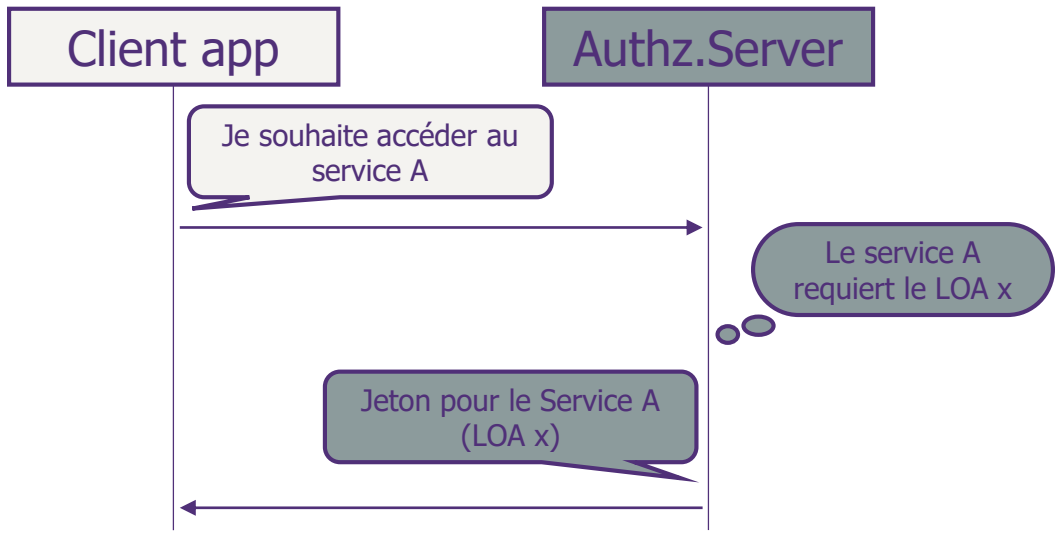
Authentification contextuelle : la solution

Aujourd'hui



- Les standards sont écrits dans cette logique initiée par le client
- Les solutions du marché fonctionnent comme ça
- Mais les besoins sur le terrain sont différents !

Le besoin réel



- Le besoin est de définir en un point central les politiques de sécurité
- Ce fonctionnement de l'authorization server est permis mais pas décrit
- Encore beaucoup de déploiements spécifiques

Les besoins additionnels rencontrés chez nos clients

Single Sign-On Mobile

Navigation sans couture entre application natives mobiles et vers ou depuis le navigateur du terminal

Authentification contextuelle

Ajustement du niveau d'authentification en fonction de la sensibilité de l'API consommée

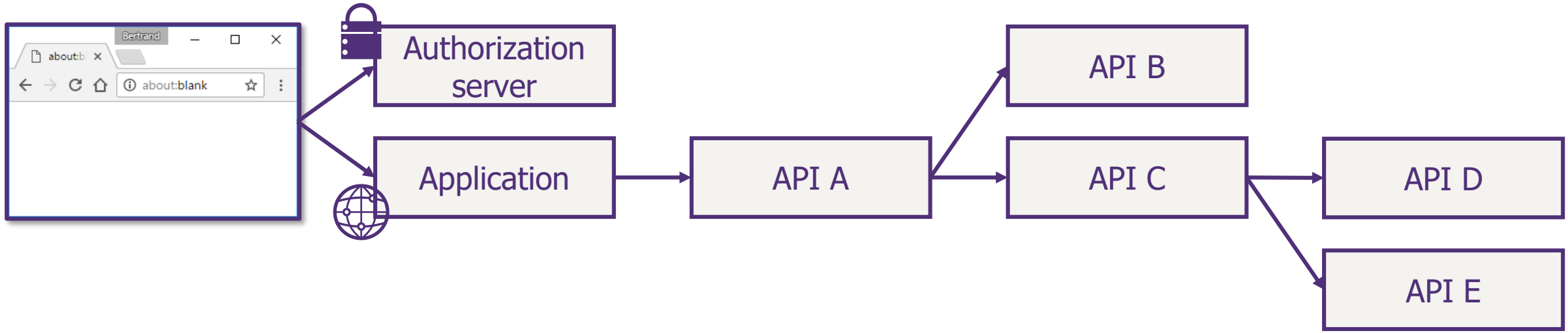
Propagation de l'identité

Transmission de l'identité de l'utilisateur aux APIs backend et validation de la chaîne d'appels

Protection contre le vol de jeton

Rendre inopérant le vol d'un jeton de sécurité et ne pas dépendre de la sécurité d'un terminal non maîtrisé ou d'un tiers

Propagation de l'identité : le besoin



Transmission du jeton

Le jeton initial portant l'identité de l'utilisateur est trop puissant

Fraude interne très facile

Authentifier l'appelant

Un composant compromis dans la chaîne peut usurper l'identité de n'importe quel utilisateur et compromet le reste de la chaîne

Transmettre deux jetons

L'intégrité de la combinaison utilisateur/API n'est pas assurée

La chaîne n'est pas vérifiable

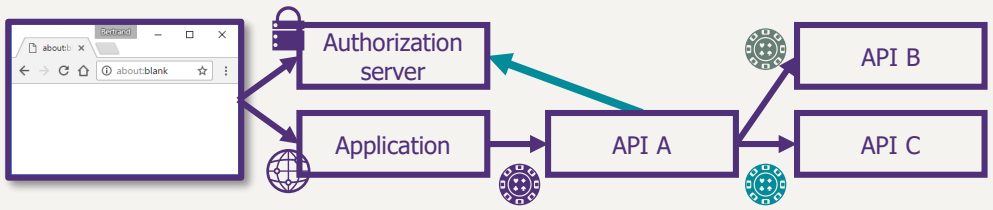
Propagation de l'identité : la solution

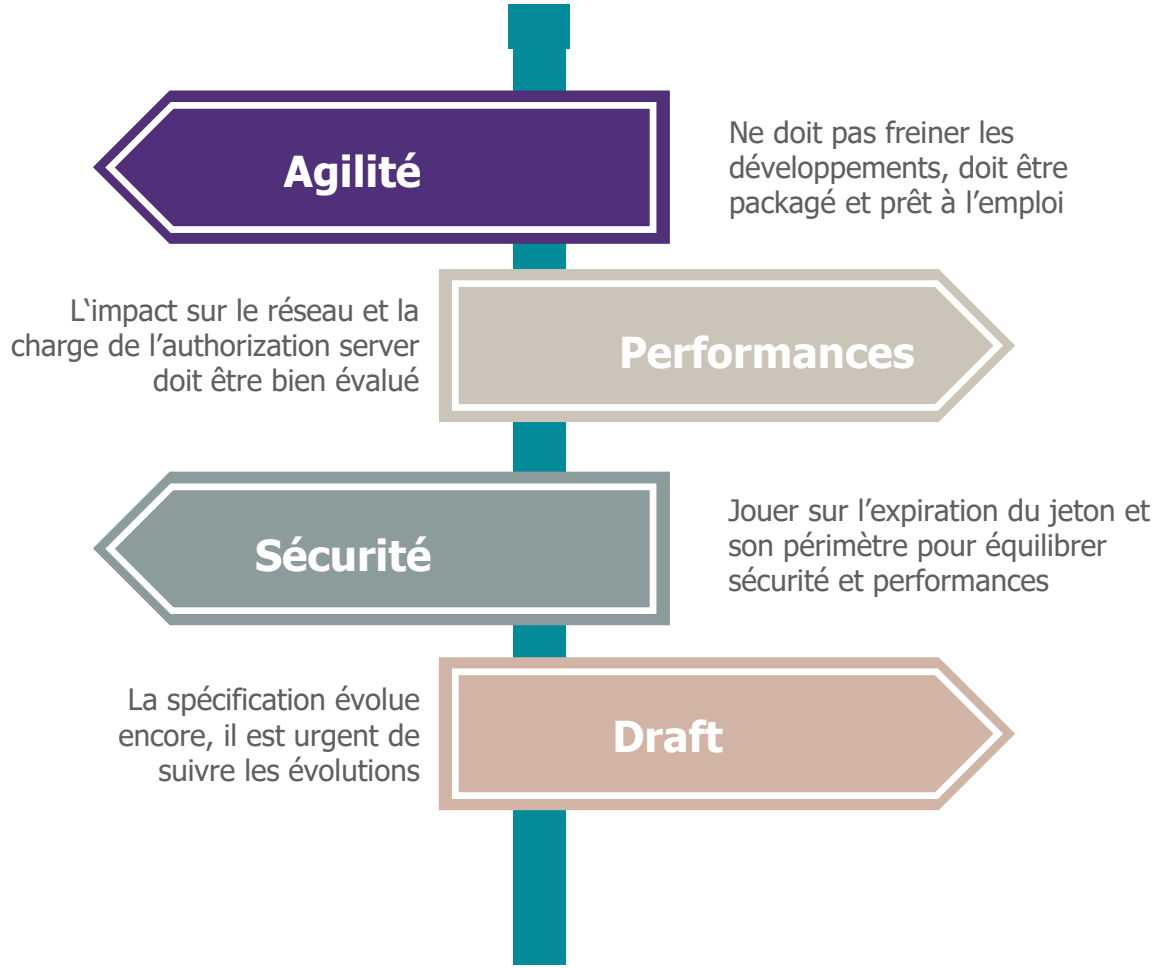
Token Exchange est un nouveau *grant type* qui définit :

1. Un mécanisme de demande de jeton intermédiaire
2. Un jeton composite permettant le contrôle de
   - l'identité utilisateur,
   - l'identité applicative
   - la chaîne d'appel

Une fois mis en œuvre, permet la centralisation :

- de la politique d'appels entre micro-services
- de l'application de cette politique
- de la traçabilité des appels entre micro-services





Les besoins additionnels rencontrés chez nos clients

Single Sign-On Mobile

Navigation sans couture entre application natives mobiles et vers ou depuis le navigateur du terminal

Authentification contextuelle

Ajustement du niveau d'authentification en fonction de la sensibilité de l'API consommée

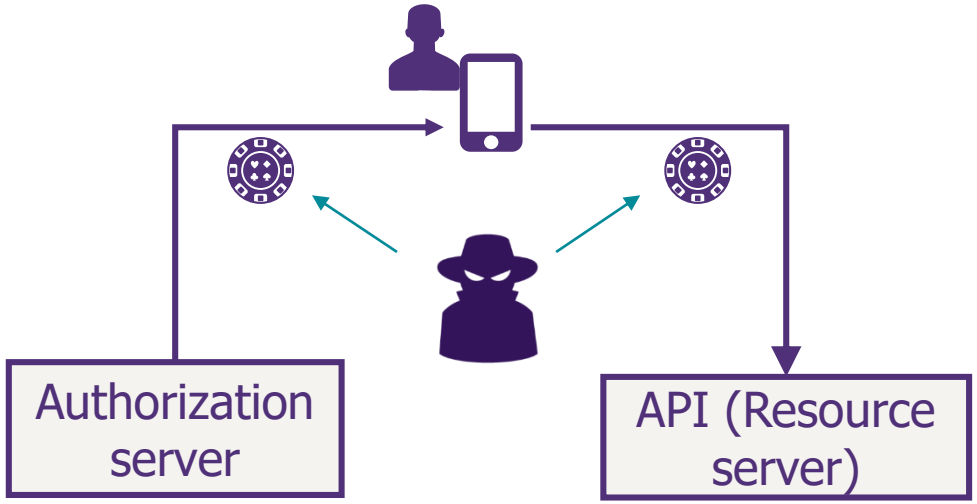
Propagation de l'identité

Transmission de l'identité de l'utilisateur aux APIs backend et validation de la chaîne d'appels

Protection contre le vol de jeton

Rendre inopérant le vol d'un jeton de sécurité et ne pas dépendre de la sécurité d'un terminal non maîtrisé ou d'un tiers

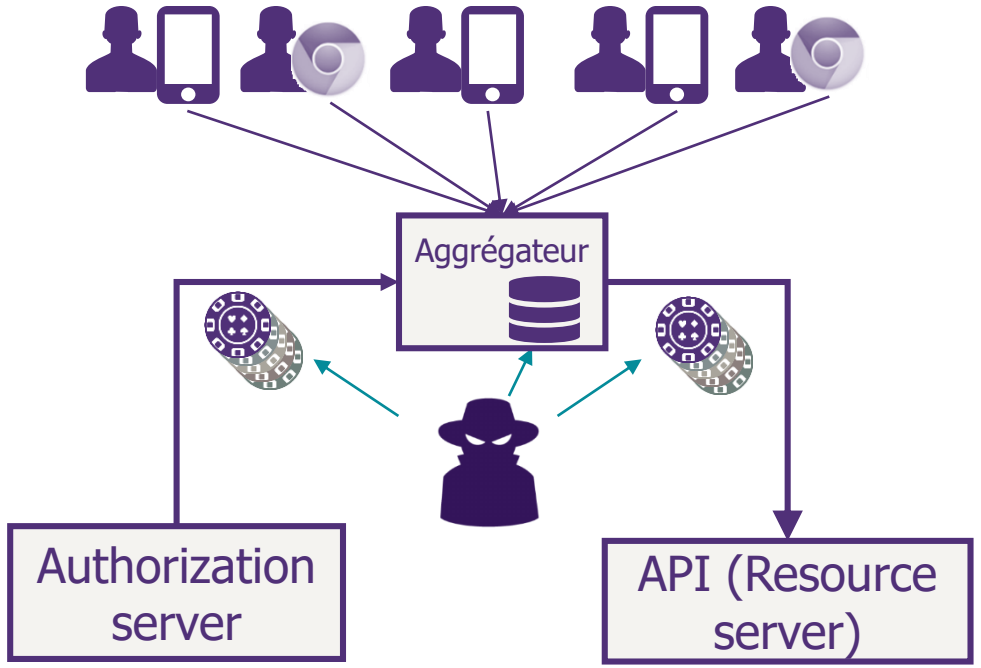
Protection contre le vol de jeton : le besoin



Vol de jeton

Le principe même de « bearer token » entraîne un risque important lors du vol de ce dernier.

La détection du vol étant très difficile, seule l'expiration du jeton est une mesure relativement efficace



Vol d'une base de jetons

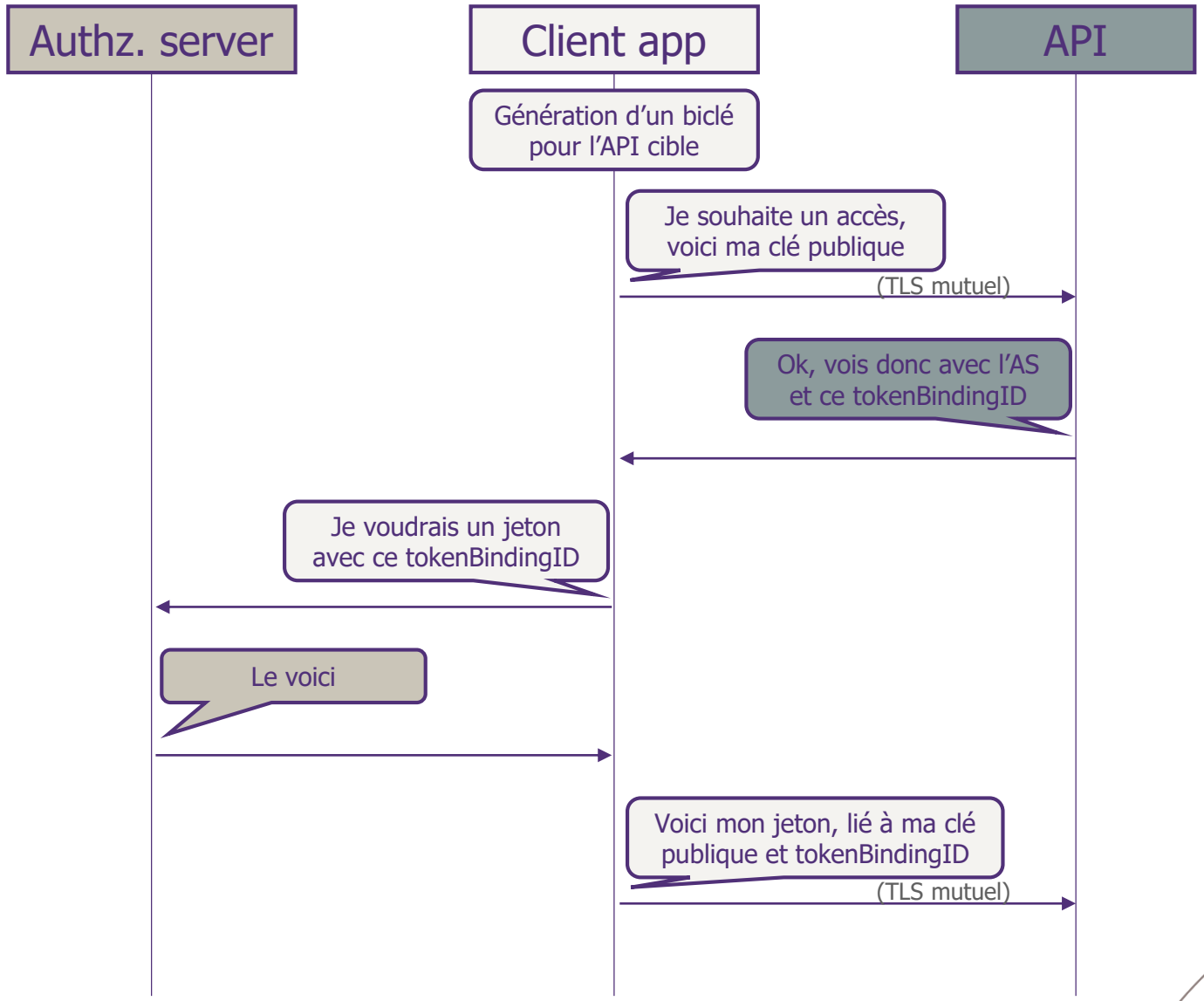
Dans un contexte d'intermédiation (eg. DSP2), un tiers peut se retrouver en possession de très nombreux jetons.

Le propriétaire de l'API est à la merci de ce tiers et de son niveau de sécurité

Protection contre le vol de jeton : la solution

Token Binding

- La négociation à deux (ou trois) composants permet de **lier un jeton** (ou un cookie) **à une clé publique et la clé privée associée**
- Le client doit prouver qu'il possède la clé privée correspondante en établissant une **connexion TLS mutuelle**
- Le jeton contient (un hash de) la clé publique du client, **distincte pour chaque serveur d'API**
- Si le jeton (ou cookie) est **intercepté**, il est **inutilisable**
- Requiert compatibilité du client et des serveurs
  - Edge, IE & Chrome disponibles en channels dev
  - Module Apache disponible



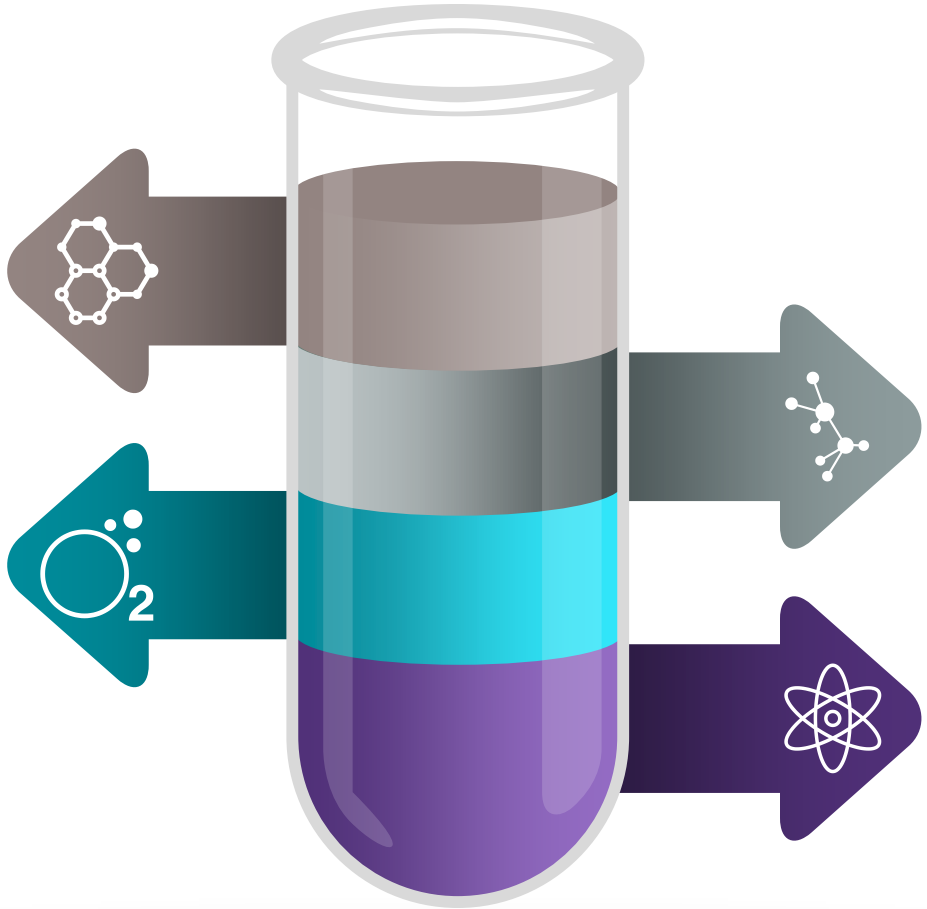
La recette pour des APIs sécurisées

Écrire la recette

Une pincée d'OAuth

Les pièges à éviter

Les recommandations essentielles



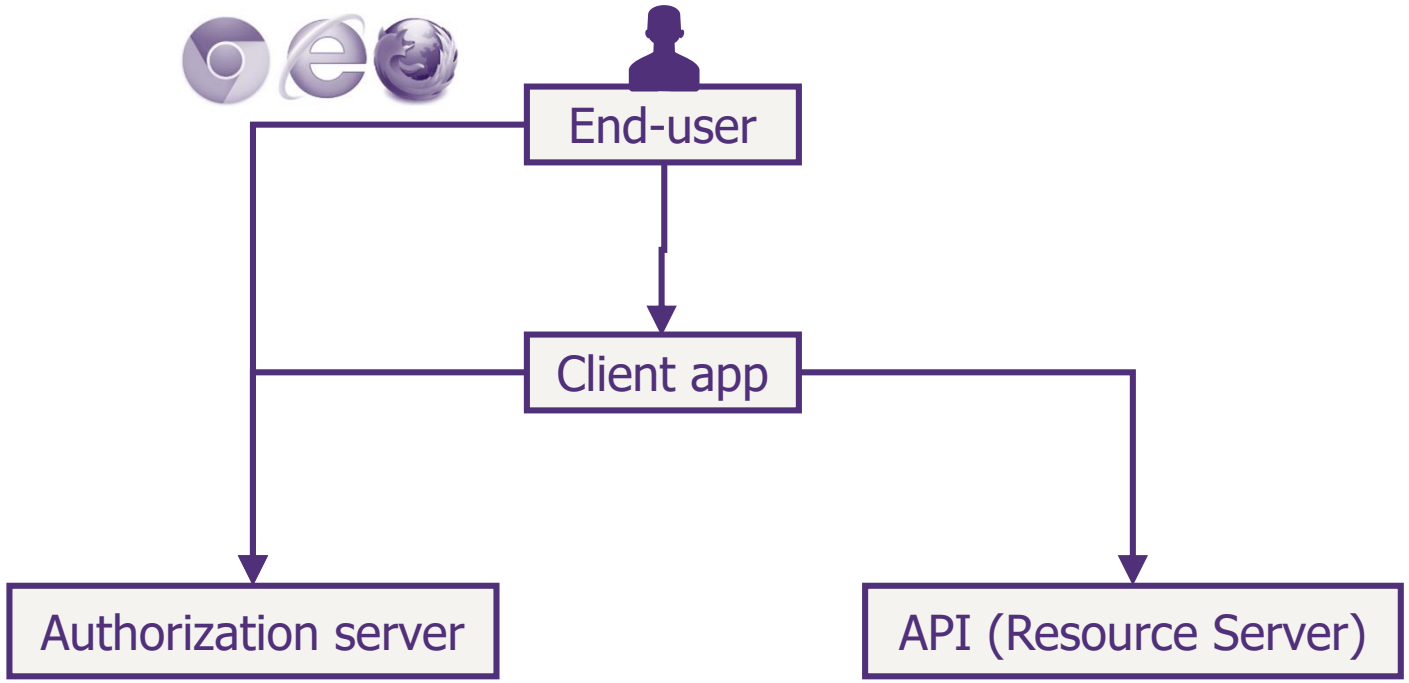
Limitez les additifs

Se poser la question des besoins

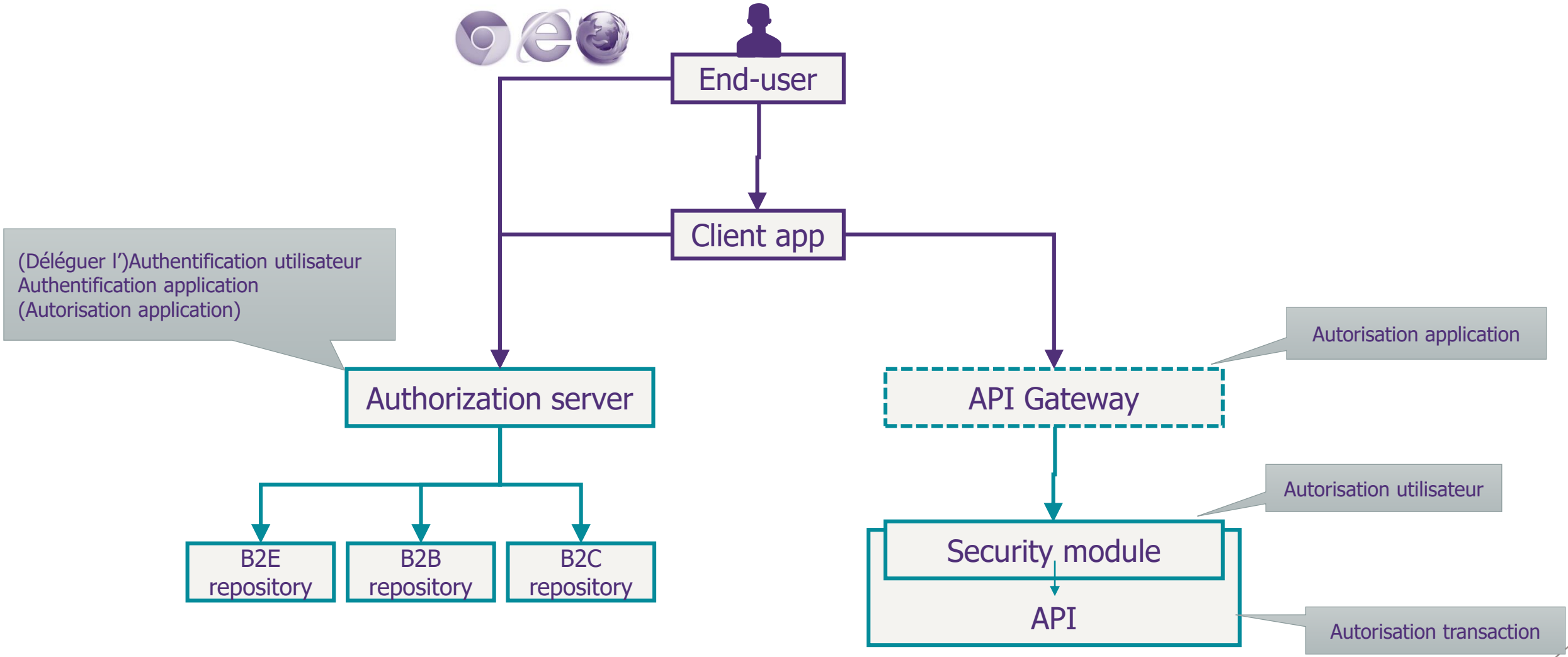
Security as usual

Une API est une application web

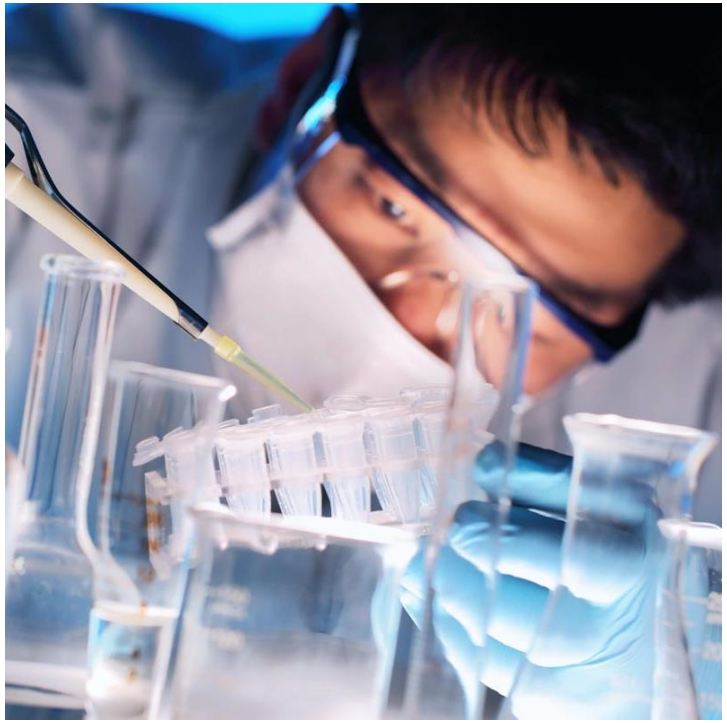
Une architecture de référence type OAuth



Une architecture de référence type OAuth à décliner dans votre contexte !



Définir le cadre d'utilisation des APIs



Définir les règles et les communiquer

- Les cinématiques autorisées et leur cadre d'application, les checklists sécurité, l'architecture de référence doivent être formalisées.

Former et outiller les développeurs

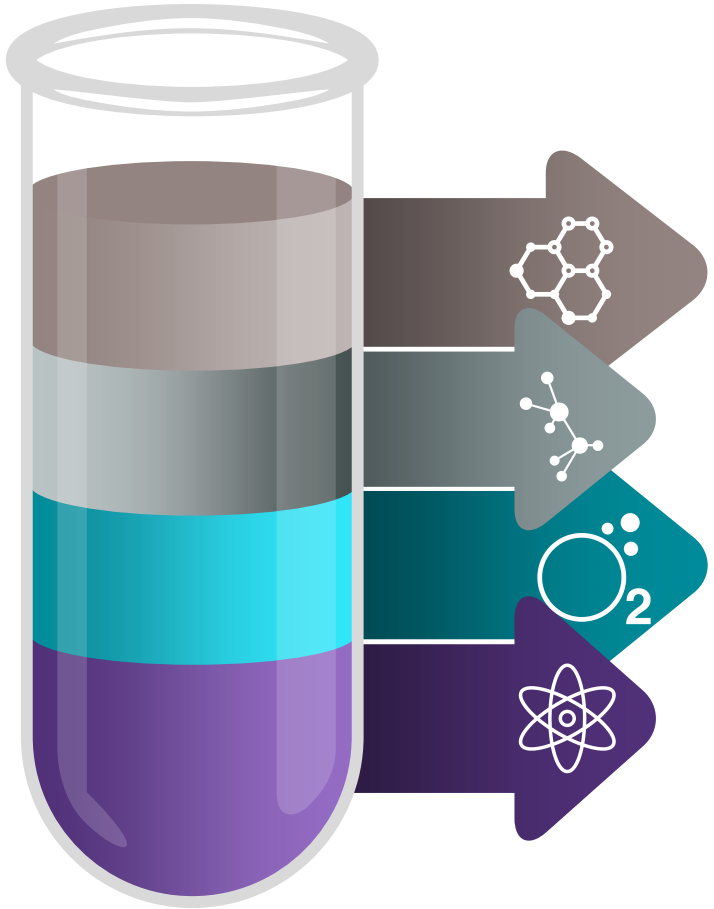
- Des sessions de formation et de présentations des principes adoptés doivent être organisées.
- Les équipes projets peuvent être rendues autonomes dans leur intégration au reste du SI.

Intégrer des ressources sécurité dans des sprints agiles

- Identifier des ressources agissant en tant que coach sécurité pour accompagner la conception applicative
- Apporter des solutions prêtes à l'emploi et être un accélérateur

Et pour finir…

La recette pour des APIs sécurisées



Ecrire la recette

Une architecture de référence et un cadre d'application

Limitez les additifs

Se poser la question des besoins réels vs standard

Une pincée d'OAuth

Sans tomber dans les pièges du standard

Une base de Security as usual

Une API est une application web

Les clés pour vivre sereinement la révolution API en cours !

WAVESTONE

Gérôme BILLOIS
Senior Manager
M +33 (0) 6 10 99 00 60
gerome.billois@wavestone.com

Bertrand CARLIER
Senior Manager
M +33 (0)6 18 64 42 52
bertrand.carlier@wavestone.com

riskinsight-wavestone.com
@Risk_Insight

securityinsider-solucom.fr
@SecuInsider

wavestone.com
@wavestone_

http://wavestone.com/privacy

Une étude exclusive auprès de 1500 citoyens de 6 pays avec les regards croisés d'un régulateur, d'un philosophe et d'experts métiers

PARIS

LONDRES

NEW YORK

HONG KONG

SINGAPORE *

DUBAI *

SAO PAULO *

LUXEMBOURG

MADRID *

MILAN *

BRUXELLES

GENEVE

CASABLANCA

ISTAMBUL *

LYON

MARSEILLE

NANTES

* Partenariats

WAVESTONE

OAuth 2, c'est très simple…