DÉVELOPPER DES APPLICATIONS WEB SÉCURISÉES…

…et après ?

3 avril 2012

Etat des lieux en 2012

- **Augmentation de la fréquence et de la complexité des attaques WEB**

- **Des intrusions parfois difficile à détecter**
  - APT: Plusieurs exemples d'intrusion et de maintient durant plusieurs jours / semaines…
  - AET: Qui en parle ?

- **De nombreuses entreprises ne maitrisent pas leur exposition sur le Web**
  - Les métiers qui contractualisent en direct
  - Architectures distribuées
  - Externalisation de services

- **Des exemples réguliers dans la presse**
  - Cyber-Espionnage
  - Cyber-Crime
  - Cyber-Activisme

Etat des lieux en 2012

- **Augmentation des interventions forensic / post-mortem en 2011**
  - E-Commerce: Chantages à l'injection SQL
  - Jeux vidéo en Ligne: Vols massif de données joueur
  - Collectivité: Défiguration de site institutionnel

- **Les applications restent vulnérables**
  - Aux attaques par injection
  - Au Cross Site Scripting
  - Aux faiblesses dans l'authentification
  - Aux faiblesses dans la gestion des sessions
  - …

OWASP Top 10 Web Application Security Risks for 2010

A1: Injection
A2: Cross-Site Scripting (XSS)
A3: Broken Authentication and Session Management
A4: Insecure Direct Object References
A5: Cross-Site Request Forgery (CSRF)
A6: Security Misconfiguration
A7: Insecure Cryptographic Storage
A8: Failure to Restrict URL Access
A9: Insufficient Transport Layer Protection
A10: Unvalidated Redirects and Forwards

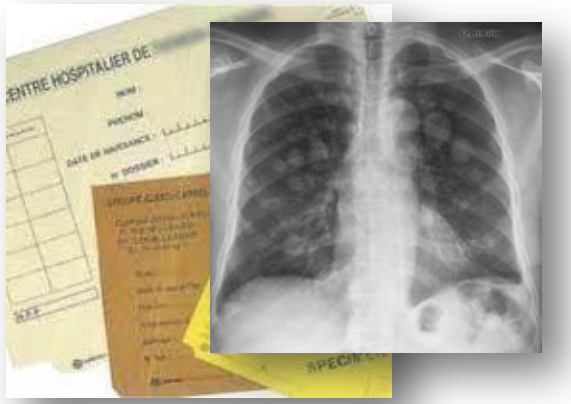
Le niveau de sécurité des applications Web ne progresse pas.

Plus de 85% des applications Web auditées par Advens en 2011 présentent des failles dans le top 5 de l'OWASP.

Morceaux choisis…

- **Banque: Test d'intrusion interne**
  - Réseau mal segmenté
  - Découverte d'une interface Web
    - Aucune vulnérabilité identifiée
  - Authentification : admin / admin
  - Possibilité de reprogrammer les paniers du DAB

- **Tests d'intrusion sur une application médicale (plusieurs hôpitaux concernés)**
  - Cross Site Scripting, difficilement exploitable
  - Prise de contrôle sur la base de données
  - Accès aux dossiers patients



Morceaux choisis…

- **Collectivité: Test d'intrusion sur applicatif métier**
  - **Attaque par Injection SQL**
    - Dump de la base des utilisateurs
  - **Attaque sur les mots de passe**
    - Découverte du mot de passe administrateur
    - Accès aux données personnelles des citoyens
  - **Exploitation d'une faille PHP**
    - Compromission totale du serveur

Nous avions déjà audité une application de cet éditeur. Le mot de passe découvert est identique à celui trouvé lors d'un précédent test d'intrusion

Recherche Google: Plusieurs collectivités possèdent cette application.

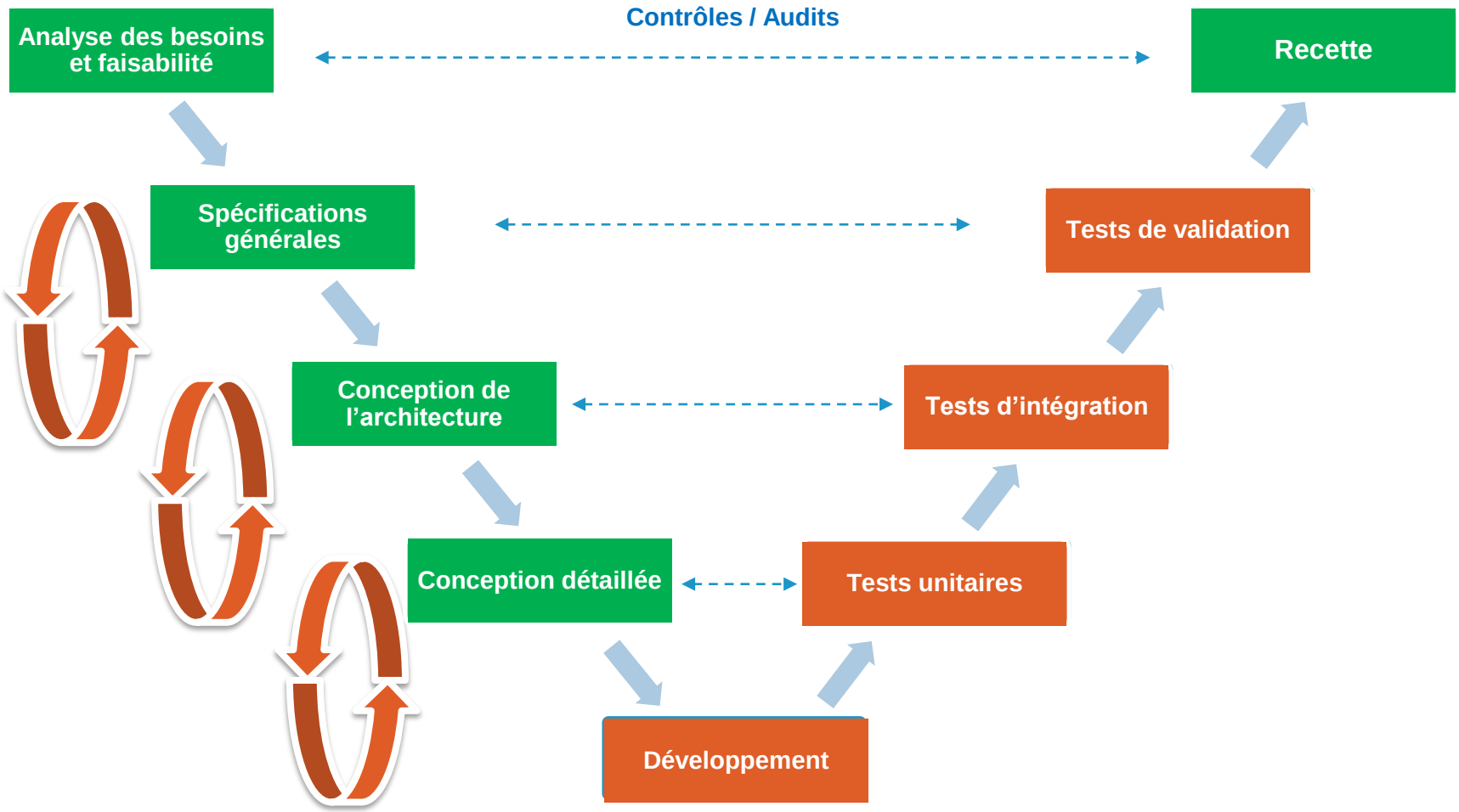
Après contact et vérification: Les installations possèdent toutes le même mot de passe…

Des causes multiples

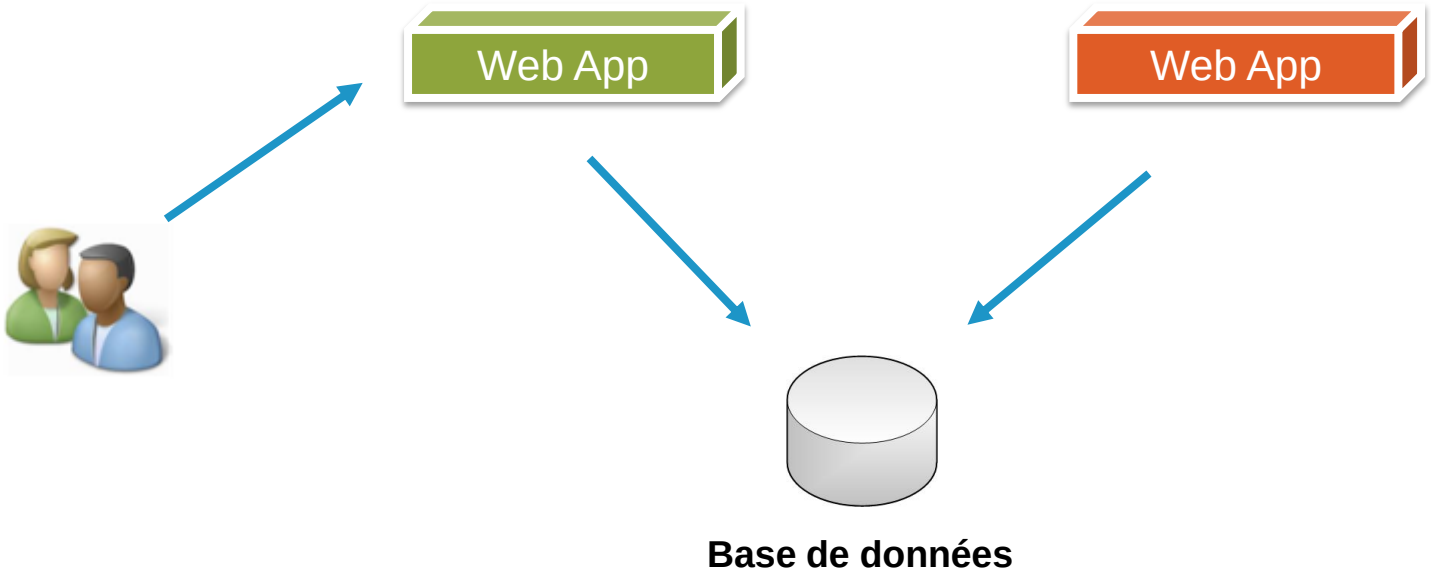
- Manque de sécurité dans les cahiers des charges
- Manque de sensibilisation des développeurs et des chefs de projets
- Manque de communication entre les développeurs / les architectes / le réseau
- Time to Market: Pression du métier
- Contrôles de sécurité insuffisants ou inadaptés
- Un comportement inacceptable de certains éditeurs sur les logiciels de niche
- Un manque de maitrise dans les techniques de sécurisation
- Une sécurité « périphérique » (ex: focus sur le frontal Web)

En théorie: Sécuriser le cycle de développement

Suivi de l'application en production

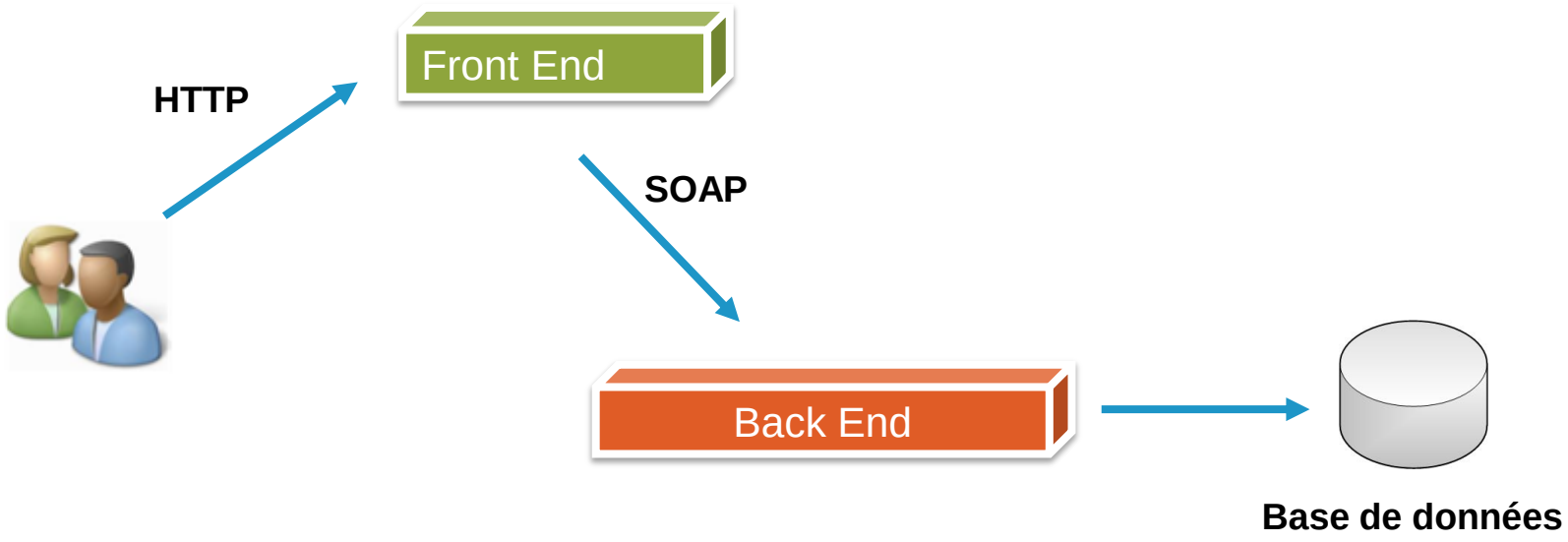


En pratique: Ne pas se limiter à l'application



Prendre en compte la sécurité de toute la chaine applicative

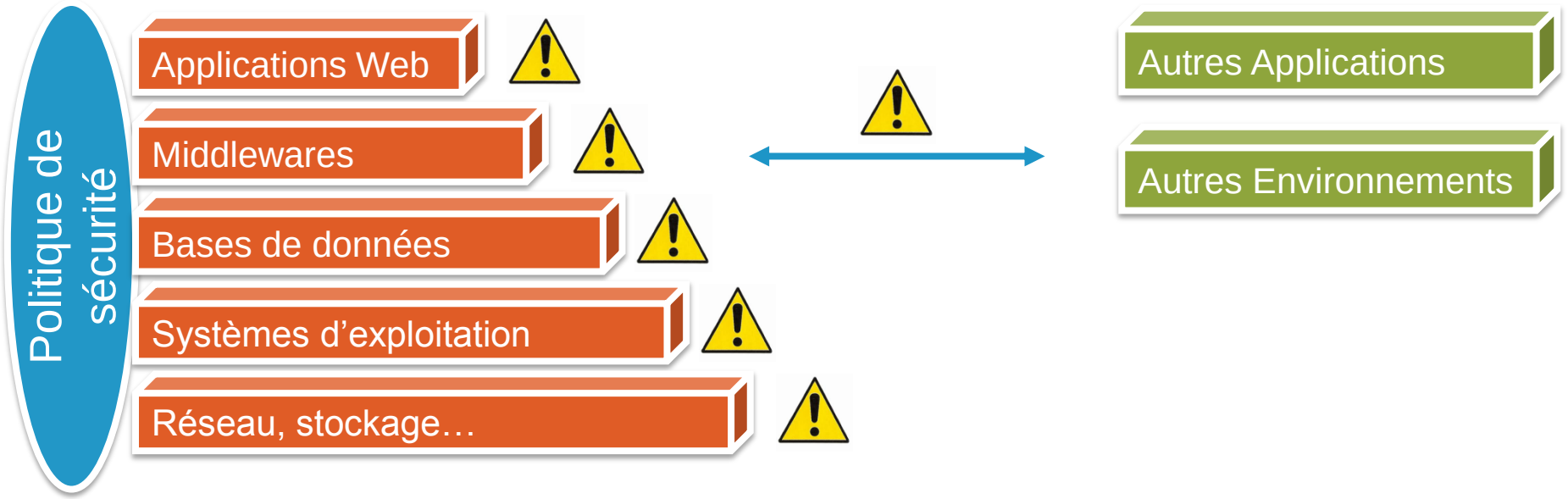
En pratique: Repenser les contrôles techniques



- **Les contrôles en boite-noire sont insuffisants et pourtant ce sont les plus utilisés**
  - Inadapté dans le cycle de développement
- **L'audit de code statique a ses limites également**
  - Inadapté dans le cycle de développement
- **Nécessité de suivre le « codeflow » pour analyser le comportement applicatif**
  - Apparition de nouveaux outils, à la portée des développeurs

Les vrais défis à relever

- **Les questions à se poser:**
  - Comment définir une politique de sécurité au niveau applicatif ?
  - Comment appliquer cette politique de manière « consistante » sur l'ensemble de la chaîne ?
  - Comment industrialiser les contrôles pour veiller au bon respect de la politique ?



Défense en profondeur

Et les données ?

- **Des référentiels de plus en plus contraignants**
- **Peu de mesures de protection réellement en place**

Applications Web

Middlewares

- **Prise de conscience**

Bases de données

- **Qui applique les mises à jours ?**

Systèmes d'exploitation

- **Amélioration:** Patch management, durcissement, contrôles de conformité opérationnelle

Réseau, stockage…

- **Très hétérogène, mais globalement OK**

Quelles solutions ?

- **Formation des développeurs**
  - La sensibilisation à l'OWASP, la formation au pentest, aux dangers d'Internet… c'est bien
  - La fourniture de méthodologies, d'outils (ou de services…) c'est mieux

- **Un développeur ne sera jamais un expert en sécurité**
  - Il faut lui donner des outils adaptés
    - Self-assessment
    - Logiciels de contrôle de la qualité / sécurité du code intégrés dans le processus de développement
  - Intégrer l'équipe sécurité dans le process de développement

- **Mener des démarches de fond sur la sécurité applicative**
  - Mise en place de « normes » de développement
  - Mise en place de Frameworks standardisés et sécurisés
  - Mise en place de cellules d'homologation des logiciels
  - …

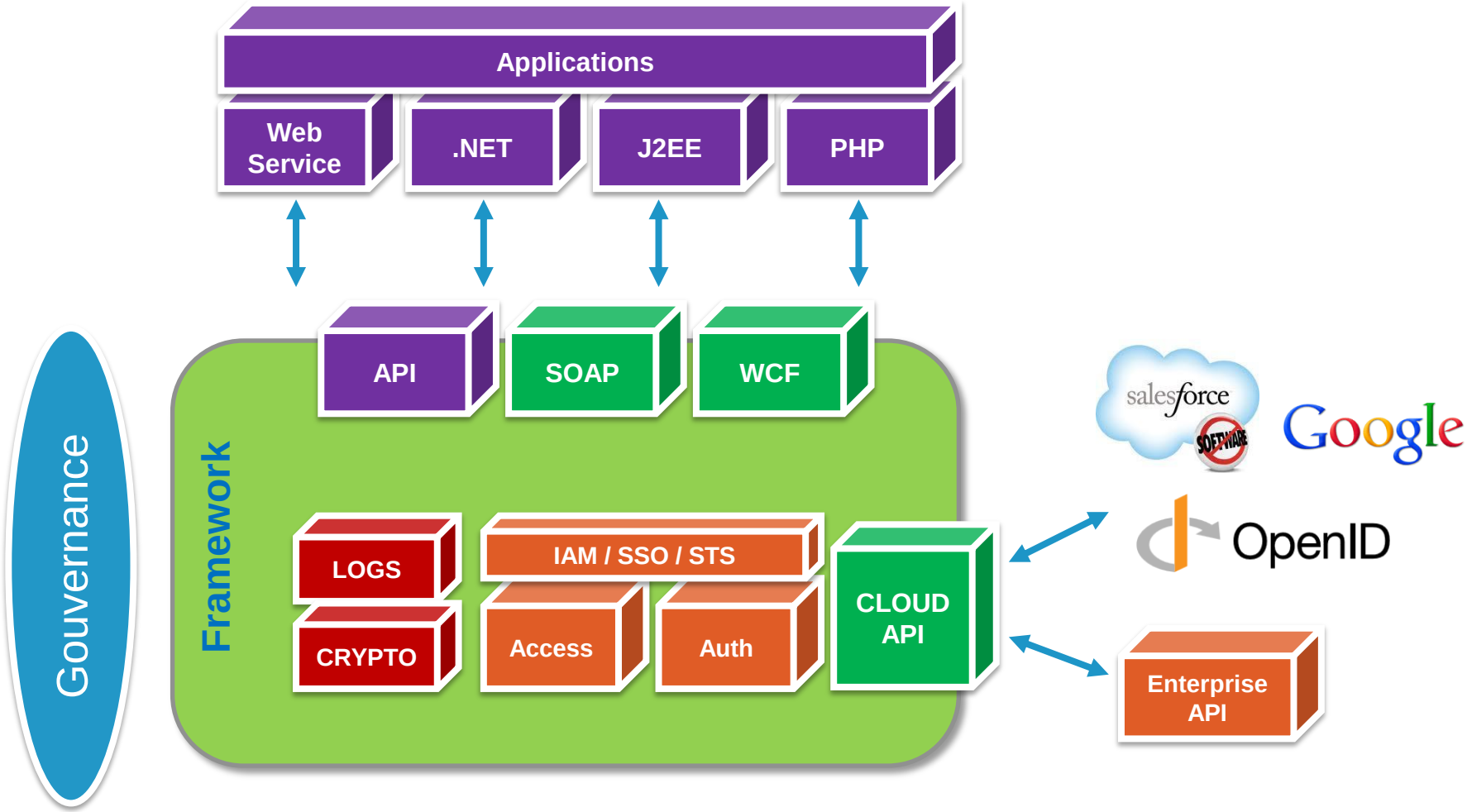
Quelles solutions ?

- **Les plans d'action des audits techniques sont-ils bien adaptés ?**
  - Appliquer un patch n'est pas toujours possible
  - Chiffrer un mot de passe en base ce n'est pas toujours possible
  - Implémenter un mécanisme d'authentification robuste ce n'est pas toujours possible
  - …
- **Proposer des solutions en terme d'architecture applicative**
  - Ne pas se limiter à l'exploitation technique de la vulnérabilité
- **Et si, au lieu de rechercher les failles, on partait du principe qu'il y en avait ?**
  - Evaluer les impacts en cas d'exploitation
  - Mettre en place les mesures de sécurité minimalistes et ne pas transiger dessus
  - Mettre en place les contrôles adaptés

Quelles solutions ?

- **Revenir au bon sens et aux fondamentaux…**
  - Isoler ce qu'on ne peut contrôler ou sécuriser au premier abord
  - Contrôler les accès
  - Traçabilité
  - Politique du moindre privilège
  - Mesures de protection ciblées: Chiffrement, signature…
- **Capitaliser sur les développements et casser la logique de « silo »**
  - Ne pas réinventer la roue à chaque projet
- **… et appréhender la sécurité dans la globalité de l'architecture applicative**

Quelles solutions ?



Retour d'expérience: Attaques ciblées (1/2)

- Attaques ciblées contre un groupe international
  - Surface d'attaque gigantesque
  - Nombreuses intrusions avérées
    - Vols de données, sabotages
- Détection
  - Les IPS n'ont rien vu (aucune alerte… et pas de logs!)
  - Plusieurs applications métiers sont inutilisables
- Réaction
  - Coupure immédiate de tous les accès réseaux vers les applications
  - Plutôt bien maitrisé, réaction dans les heures qui suivent les attaques
- Analyse
  - Plusieurs jours nécessaires pour la remise en état des bases de données
  - Scans des vulnérabilités applicatives en urgence
    - Défauts de mise à jour, SQL Injection, XSS, inclusion de fichiers, configurations par défaut, Interfaces de management accessibles…

Plusieurs mois / homme de travail pour sécuriser les applications métier

Retour d'expérience: Attaques ciblées (2/2)

- Mise en place de firewalls applicatifs sur plusieurs sites
  - Protection des applications métiers stratégiques pour le groupe
  - Mise en place de politiques très strictes
  - Tuning de la configuration **avec les responsables applicatifs**
    - Monitoring des WAF
    - Tests fonctionnels sur l'intégralité des processus applicatifs
    - Ajustement des politiques / élimination des faux-positifs

- Bilan après plusieurs mois de service
  - Surveillance continue des WAF (supervision temps réel de l'activité)
    - 100% des scans / bruit de fond Internet bloqué
    - Détection de plusieurs attaques de type « découverte »
  - Contrôle d'intégrité temps réel sur les interfaces publiques / authentification
  - Scans applicatifs réguliers
  - Initialisation d'un programme de sécurisation des applications

- Attention à la gestion du changement

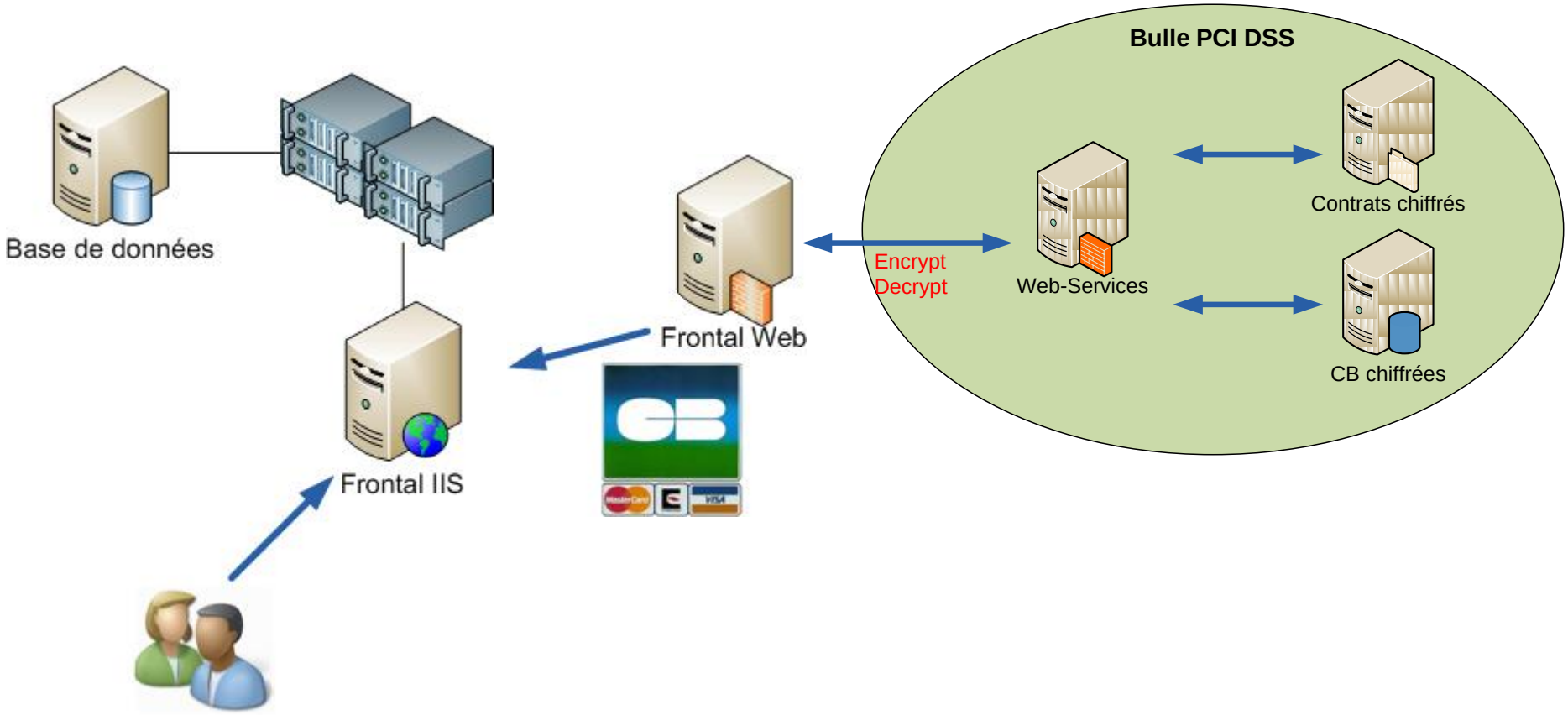
Retour d'expérience: Mise en conformité PCI DSS (1/2)

- **Société dans le secteur de l'assurance**
  - Plateforme applicative dédiée au métier de l'entreprise
  - Souscription de contrats par téléphone, via une plateforme d'appel
  - Collecte d'informations personnelles
  - Collecte de données de type cartes bancaires

- **Plateforme accessible à d'autres courtiers / assureurs**

- **L'un des assureurs exige une conformité PCI DSS de la plateforme**
  - Plateforme mutualisée
  - Numéros de carte bancaire en clair dans la base de données
  - Numéros de carte bancaire « en clair » dans certains documents

Retour d'expérience: Mise en conformité PCI DSS (2/2)



Questions ?

Jérémie Jourdin - jeremie.jourdin@advens.fr

Advens | www.advens.fr