Secure hotswapping : un réel problème pour carte à puce

Agnès NOUBISSI

Jean-louis Lanet & Julien Iguchi-Cartigny

Université de Limoges

Xlim institut de recherche

Equipe SSD


Java Card

Sommaire

- Pourquoi la mise à jour en ligne (DSU) pour carte à puce?
- Objectifs de la DSU
- État de l'art pour DSU
- Présentation du champ d'étude : La Java Card 3.0 *connected edition*
- Approche de mise à jour proposée
- Processus de DSU en on-card
- Conclusion

I- Pourquoi la DSU pour carte à puce?

Un constat

- Passeport électronique
- Carte bancaire
- etc

La durée de vie de plus en plus élevée des cartes

Difficile pour la carte de résister aux nouvelles Attaques
(physiques, logiques, cryptographiques)

Certaines applications Possèdent une durée de vie identique à celle de la carte

DSU des applications et Composants systèmes embarqués

II- Objectifs de la DSU

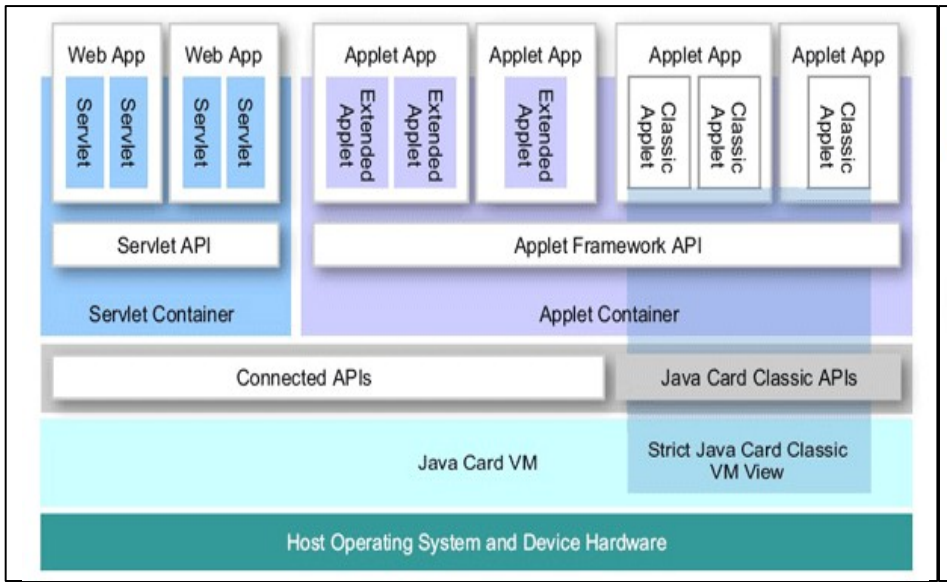
- Fixer des bugs dans les algorithmes cryptographiques
- Améliorer ou ajouter des fonctionnalités au systèmes et aux applications embarquées
- Supprimer des fonctionnalités obsolètes.

III- Etat de l'art : DSU sécurisée

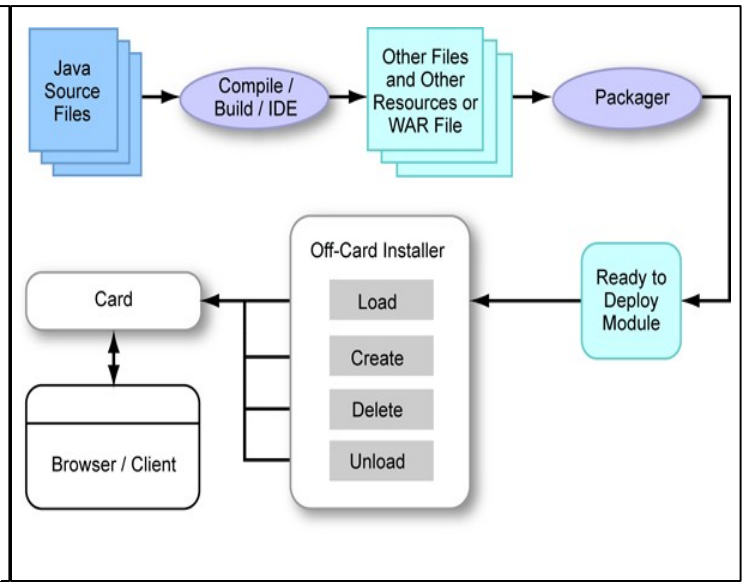
- De nombreuses systèmes de DSU existent
  - Jvolve, Jdrums, DVM, etc
- Systèmes d'exploitations actuels au niveau des cartes à puce
  - JAVACARD, MULTOS , CAMILLE, SMARTCARD.NET
- Limites actuelles
  - Au niveau des applications: Ajout – Retrait & Au niveau des modules systèmes : Extensions
  - Réduction de la flexibilité de l'utilisation de la carte à puce
- Apports futurs
  - DSU des applications embarquées sur la carte
  - DSU des composants systèmes (code natif) de la carte
  - Proposer des solutions aux nouvelles problématiques sécuritaires qu'apporte la DSU

IV- Présentation de la Java Card 3.0 *connected edition*

Architecture JC 3.0

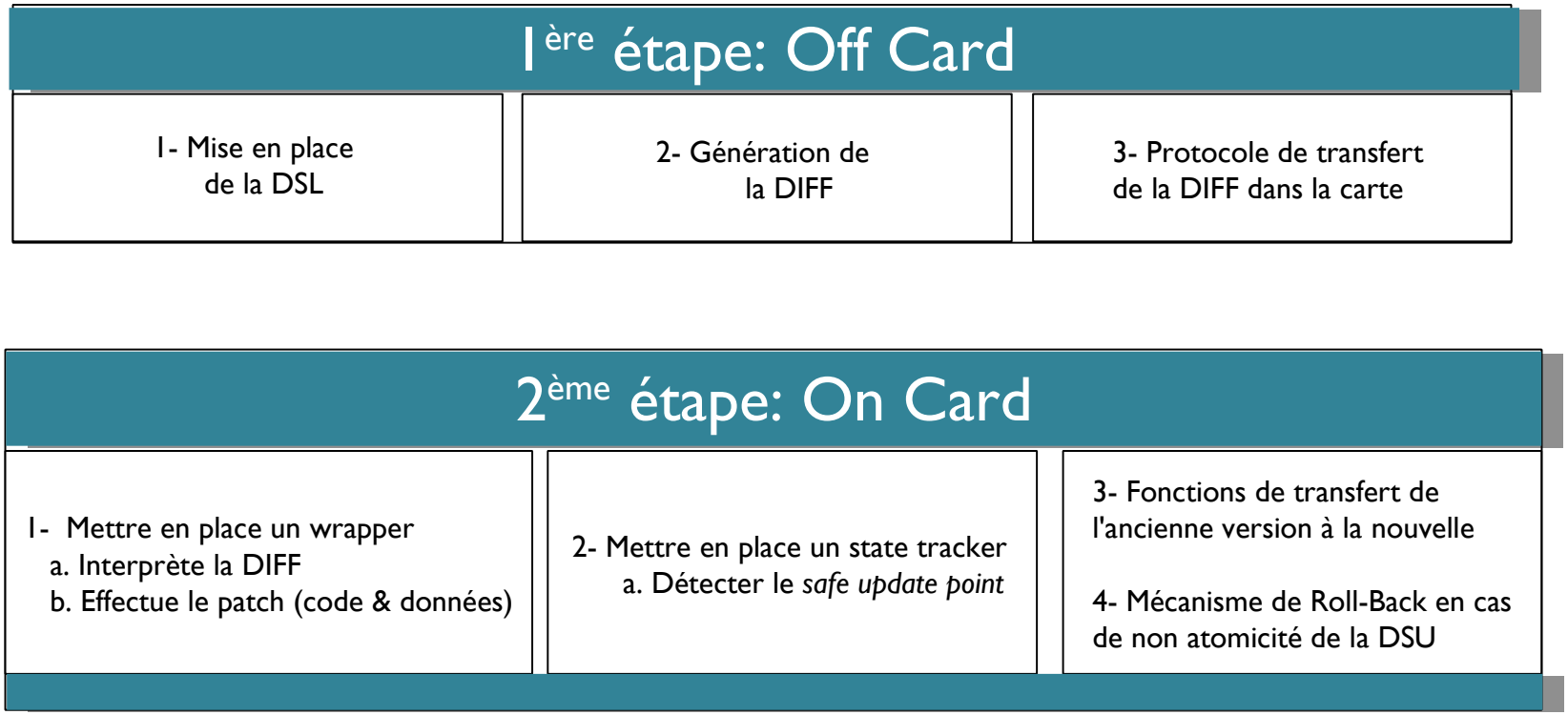


Processus de déploiement d'une application JC 3.0



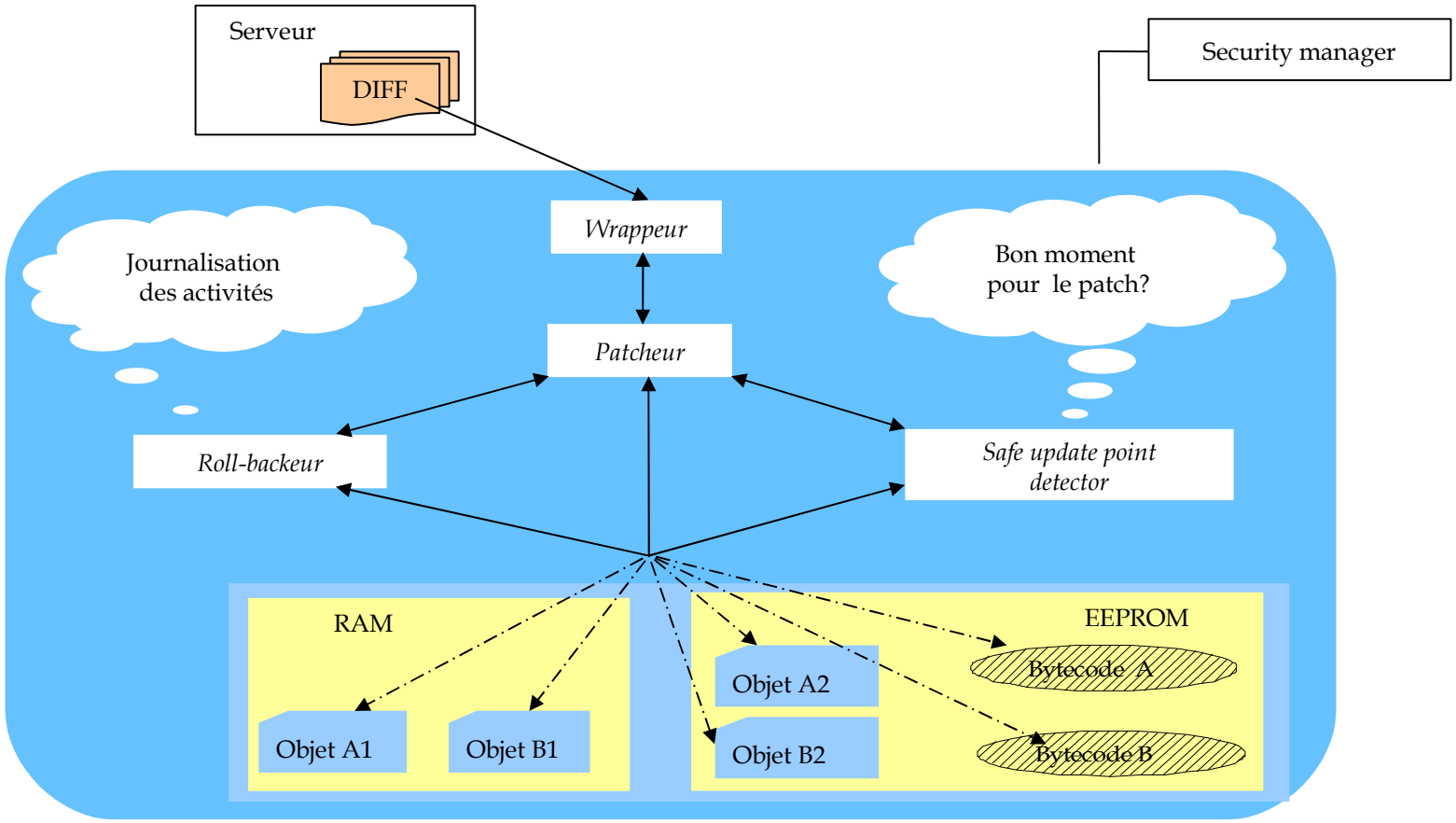
V- Approche de DSU proposée

- DSU sécurisée



V- Approche de DSU : Le module updateCardApp

- Communication entre les composants du module



VI- Processus de DSU en on card

- La modification de la définition de la classe
  - Mettre à jour le code de la classe
  - Mettre à jour les objets de la classe
- La modification des instances de la classe
  - Recherche de toutes les instances de la classe
  - Et mise à jour des ces instances
- La modification des classes dépendantes
  - Analyse du flot de donnée entre les classes
  - Mise à jour des classes dépendantes
  - Mise à jour des services publiés dans le service manager de la carte
- Et la sécurité lors de ces modifications.

Conclusion

- La DSU est un problème relativement complexe notamment dans le domaine des cartes à puces où les contraintes des ressources sont encore plus sévères

- L'utilisation d'un *safe update point detector* permet de repérer le moment idéal pour le patch afin de préserver la cohérence du système de type dans la JC.

- Cependant rester cohérent avec l'architecture Java Card est un point essentiel d'où l'intégration du module *updateAppCard* dans le module de gestion du cycle de vie des applications sur la carte.

- Toutefois, des nouvelles failles de sécurité sont à déterminer et des solutions à celles ci sont à proposer.

"C'est en faisant des erreurs qu'on apprend."

Merci pour votre attention !

Des questions