



Quelques travaux autour des « smart objects » et des flottes de drones

Damien Sauveron

13 juin 2016

<http://damien.sauveron.fr/>

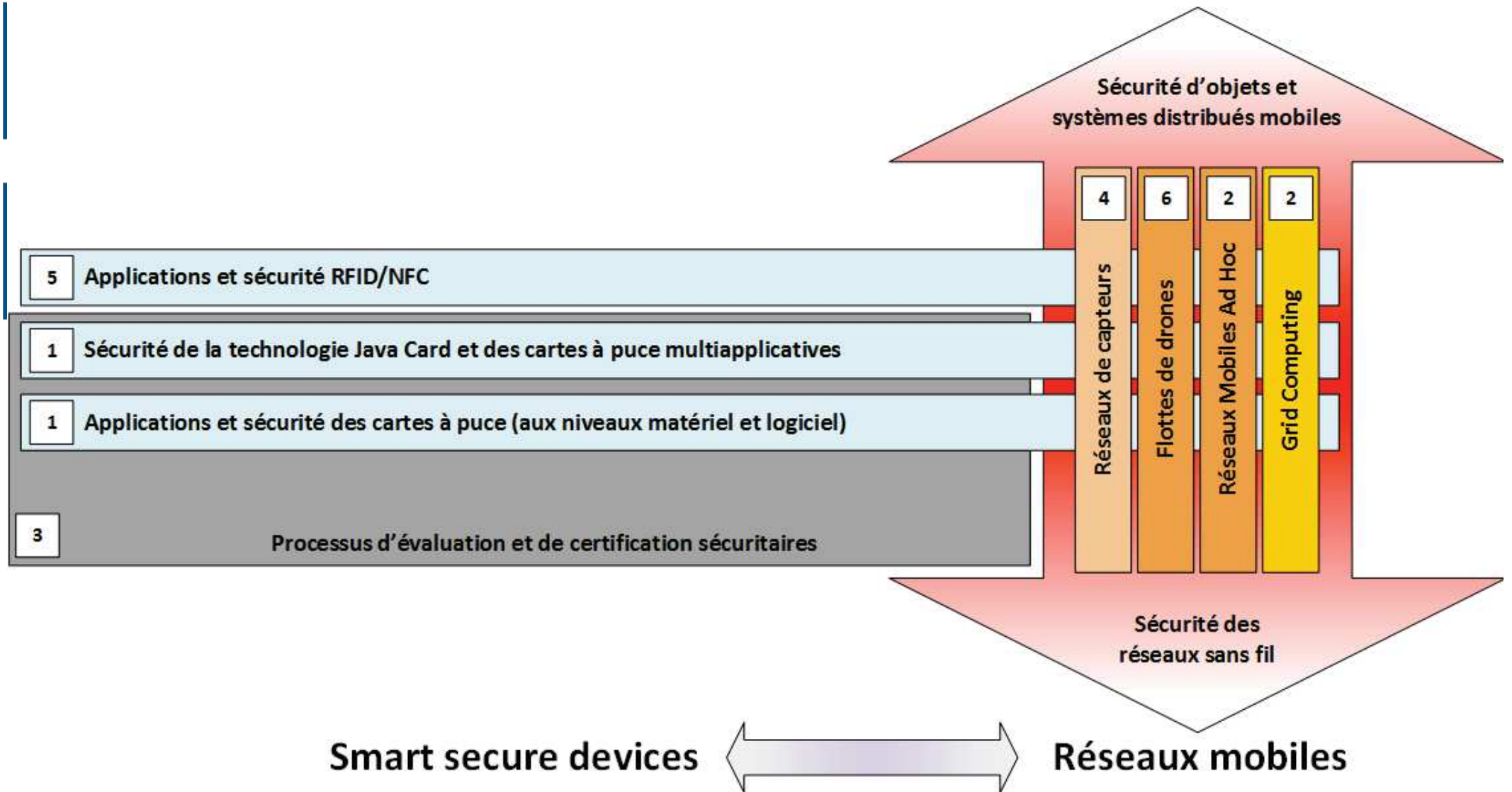
Plan de la présentation

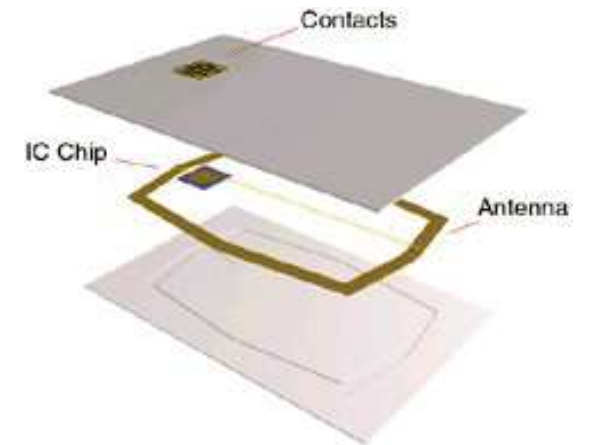
- **Mes centres d'intérêts recherche**
- **« Smart objects »**
 - Java Card
 - Vers des solutions plus ouvertes ... user-centric ?
- **Flottes de drones**
 - Sécurisation du drone et des communications
 - Optimisation des communications
 - Communications « Smart Objects » et Drones
- **Conclusion et perspectives**

Bio

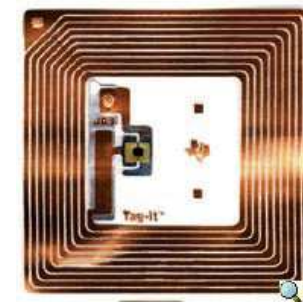
- **Maître de conférences à l'Université de Limoges (09/2006 – ...)**
 - **Doctorat en 2004 à Bordeaux et Post-doc en 2006 au Royal Holloway**
 - **HDR en 2014**
 - **2 thèses en cours :**
 - **Sécurité des RFID actifs**
 - **Solutions de sécurité pour l'IoT**
- **Enseignements de M2**
 - **à Limoges**
 - **Carte à puce et applications**
 - **Certification et développement sécurisé**
 - **à Bordeaux**
 - **Carte à puce et Java Card**
 - **à Clermont-Ferrand**
 - **Sécurité des systèmes embarqués**
- **Président de IFIP WG 11.2 Pervasive Systems Security (depuis 2014 et vice-président depuis 2007)**

Mes centres d'intérêts recherche



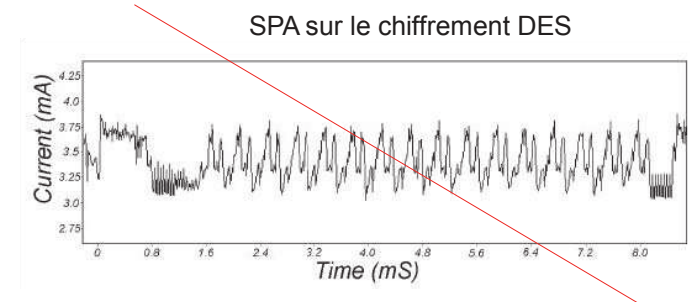
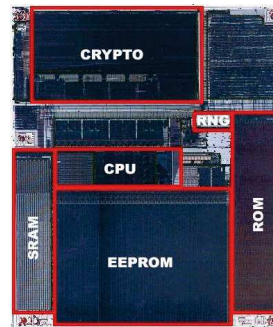
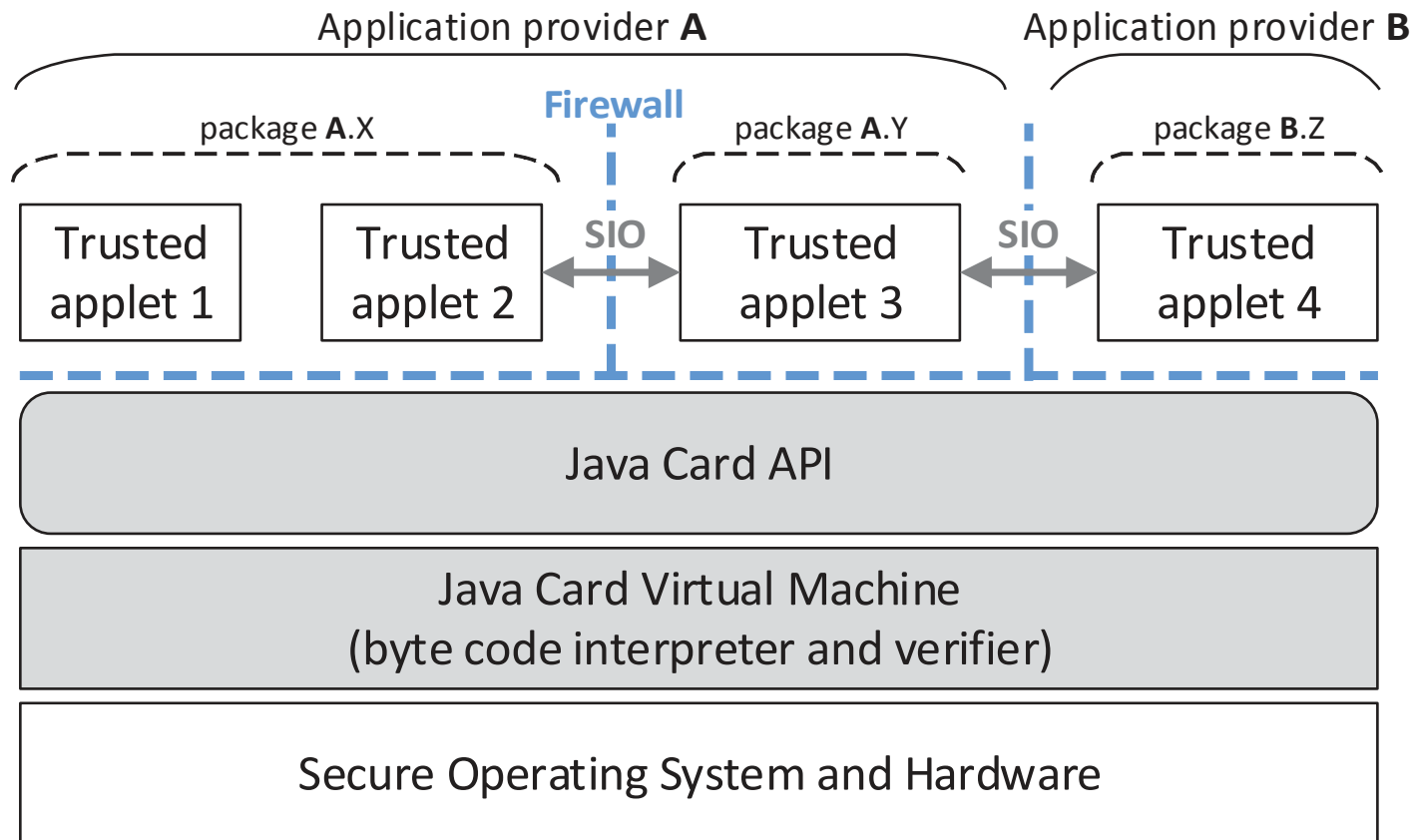


« Smart objects »



Java Card

- Une plate-forme multi-applicative sécurisée pour cartes à puce



Principales contributions

- Identification de bugs sémantiques dans les spécifications Java Card
- Identification des **nouvelles problématiques** liées à la multiapplication sur cartes à puce.
- Définition d'une méthodologie et d'outils efficaces pour attaquer la sécurité Java Card à bas niveau
- Définition d'un **nouveau modèle d'utilisation et de gestion du contenu des cartes à puce multiapplicatives**.
- **Développement de l'écosystème pour les cartes suivant le modèle centré utilisateur** avec Raja Naeem Akram et Konstantinos Markantonakis



Konstantinos
Markantonakis



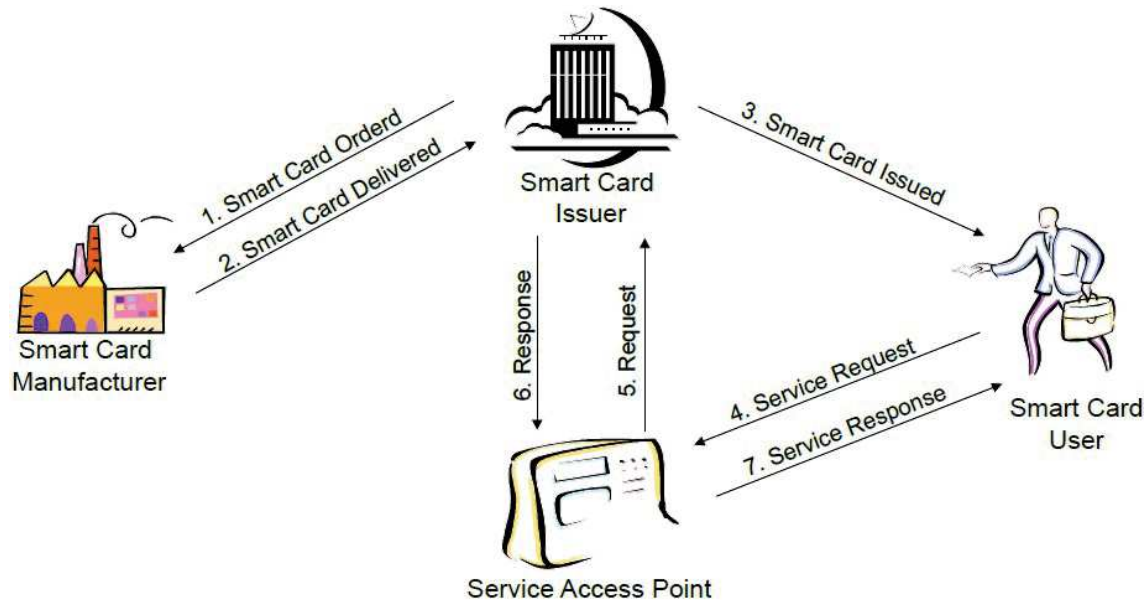
Raja Naeem Akram



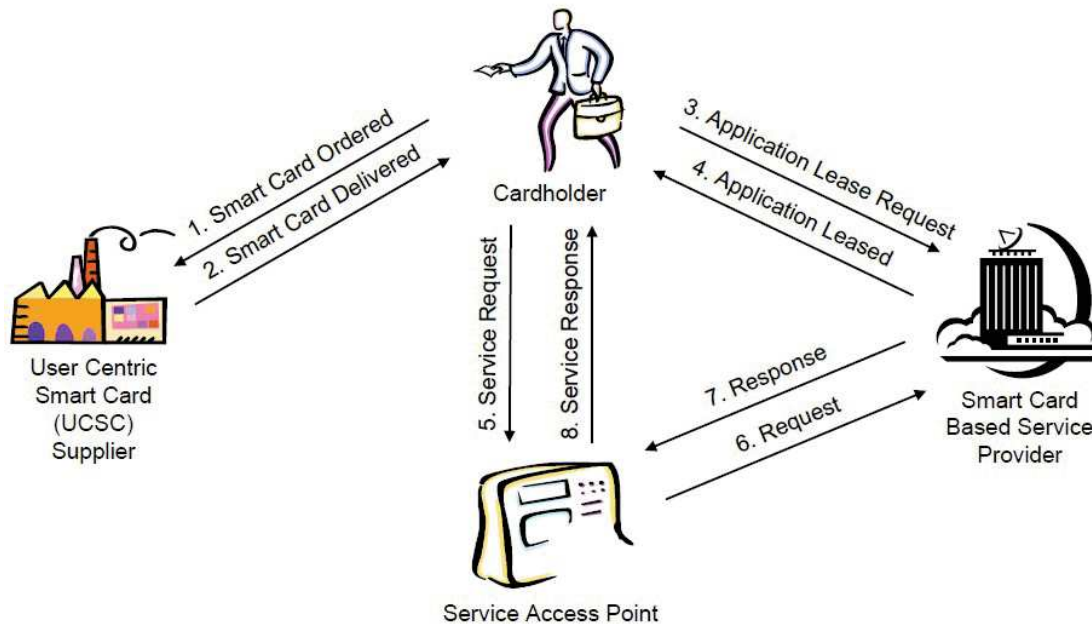


Quel modèle pour les cartes ?

Modèle centré sur l'émetteur vs modèle centré sur l'utilisateur



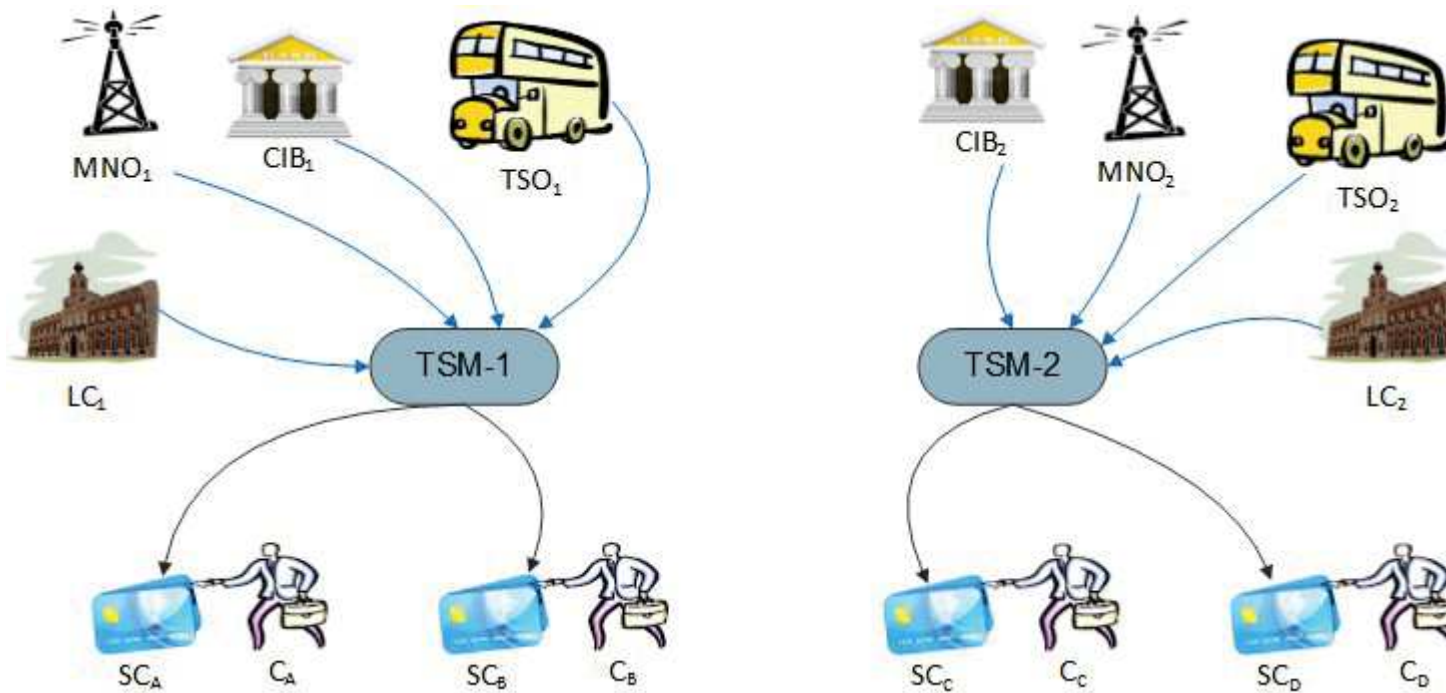
ICOM
ou l'échec des cartes
multiapplicatives ?



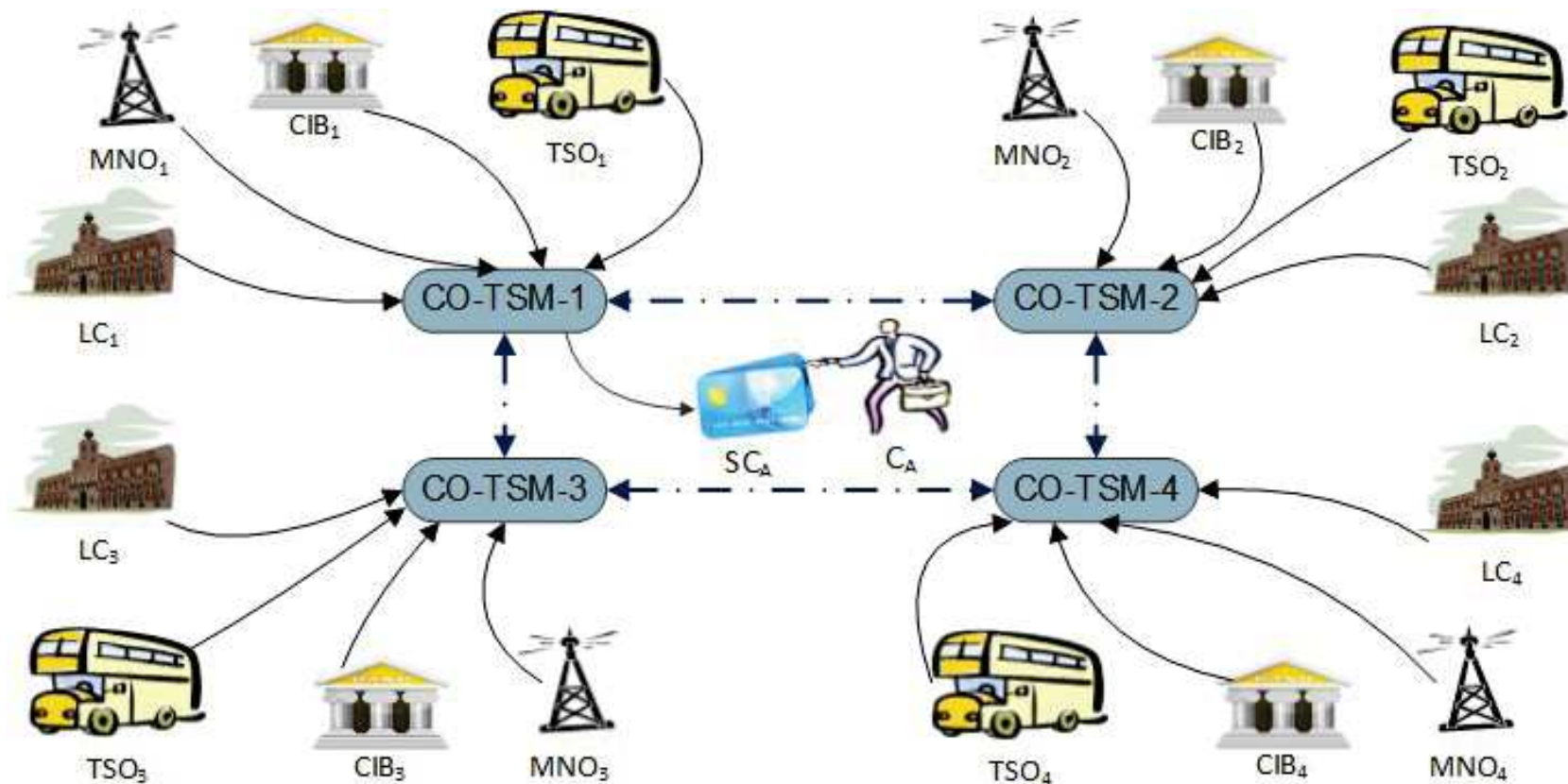
UCOM
Un modèle réaliste ?
Voir le white paper GP-CCM

Le Trusted Service Manager

- NFC, le sauveur de la multiapplication ?



Un TSM collaboratif et flexible

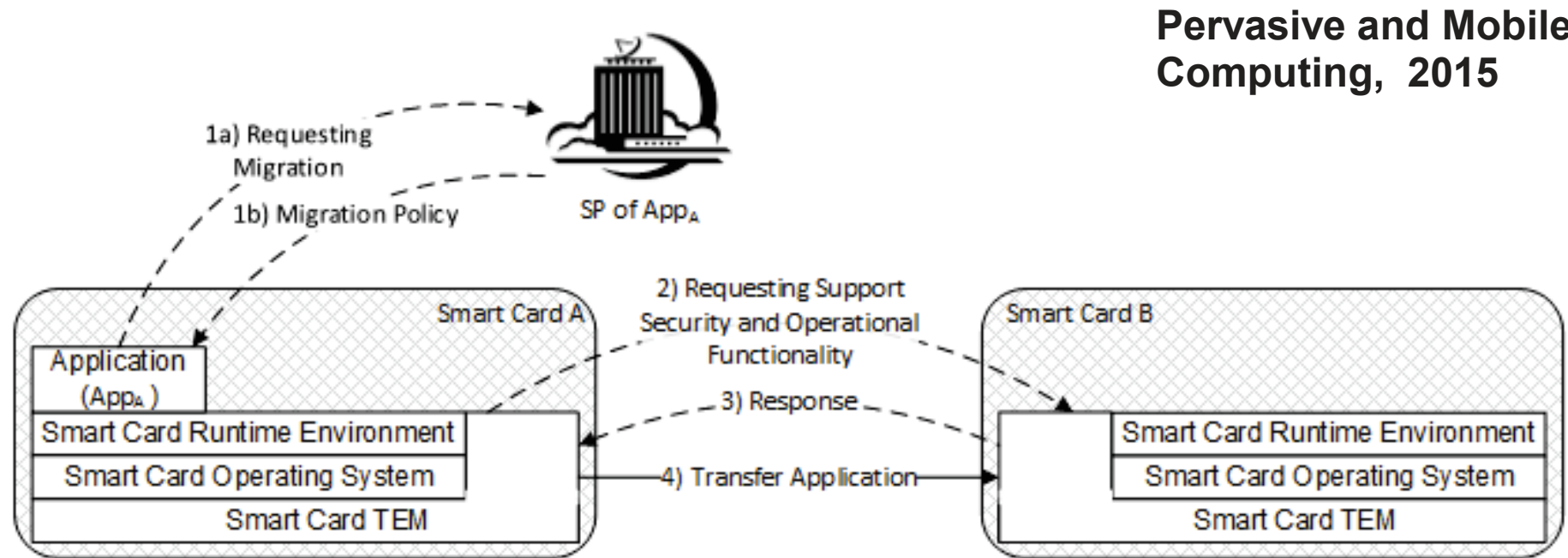


- Adaptable à ICOM, UCOM et GP-CCM

TrustCom 2014

Des points critiques pour l'écosystème User-centric

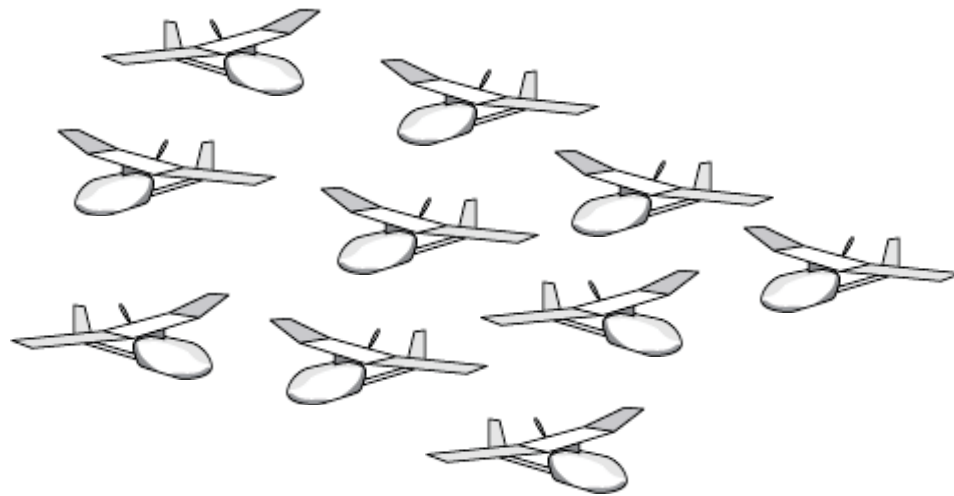
- Sauvegarde et migration d'applications



- Un framework pour minimiser les conséquences des FIP (Feature Interaction Problem)

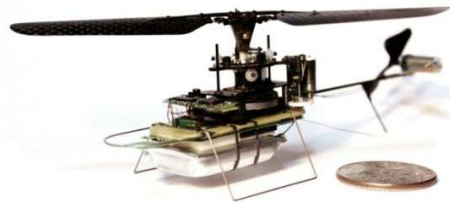
TrustCom 2015

Flottes de drones



Pourquoi des flottes de drones ?

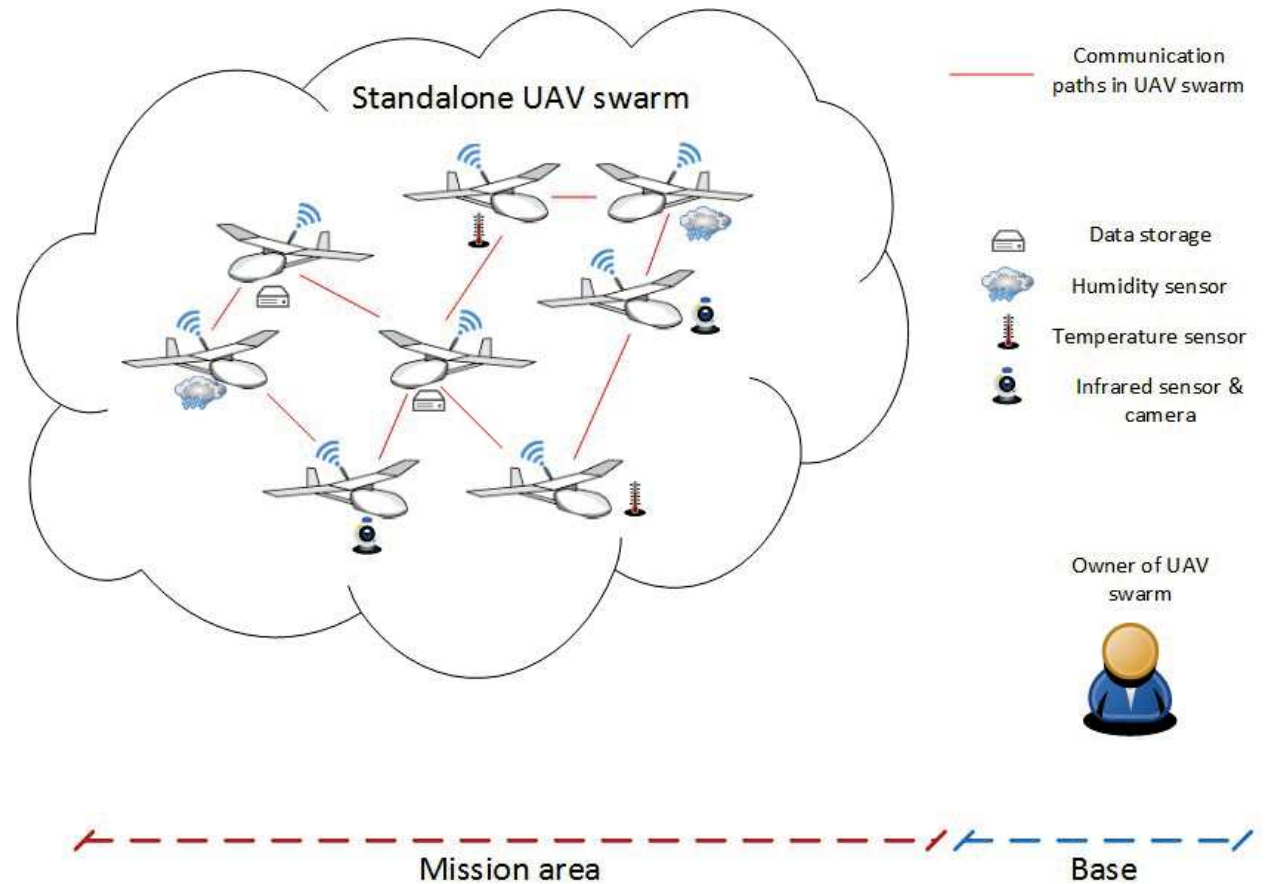
- Elles peuvent être plus efficaces et moins chères qu'un simple drones



Proxy Dynamics PD-100 PRS Black Hornet Prototype

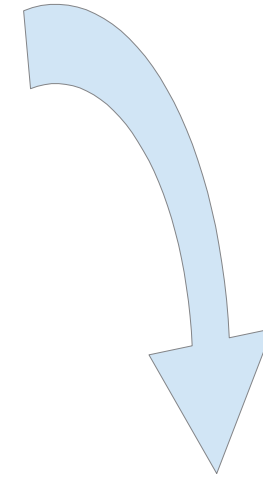
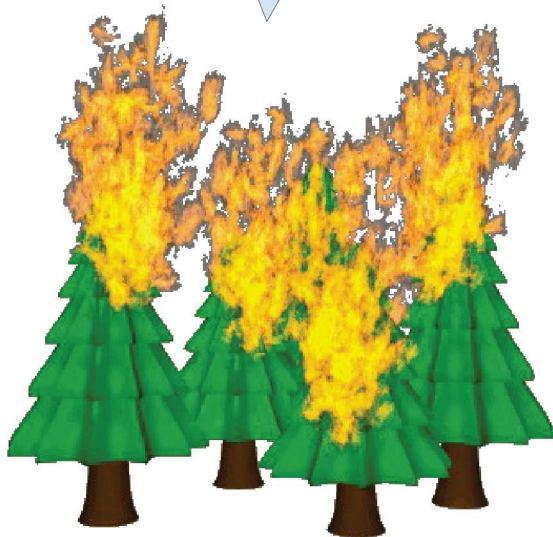
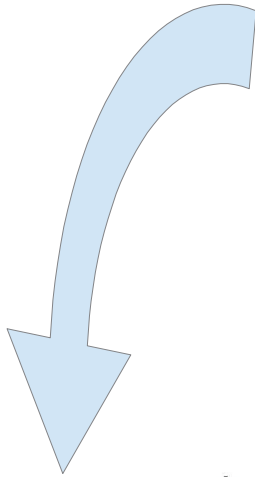
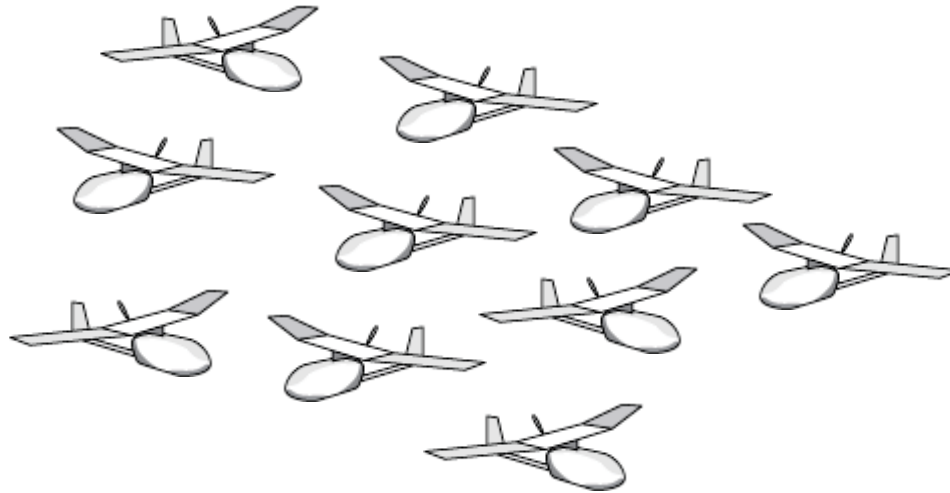


Proxy Dynamics PD-100 PRS Stealthy Look

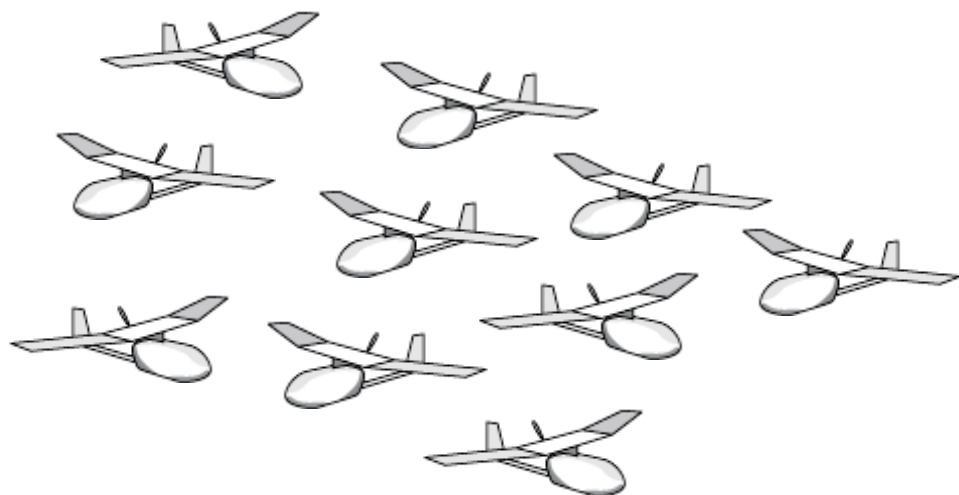


Domaines d'application

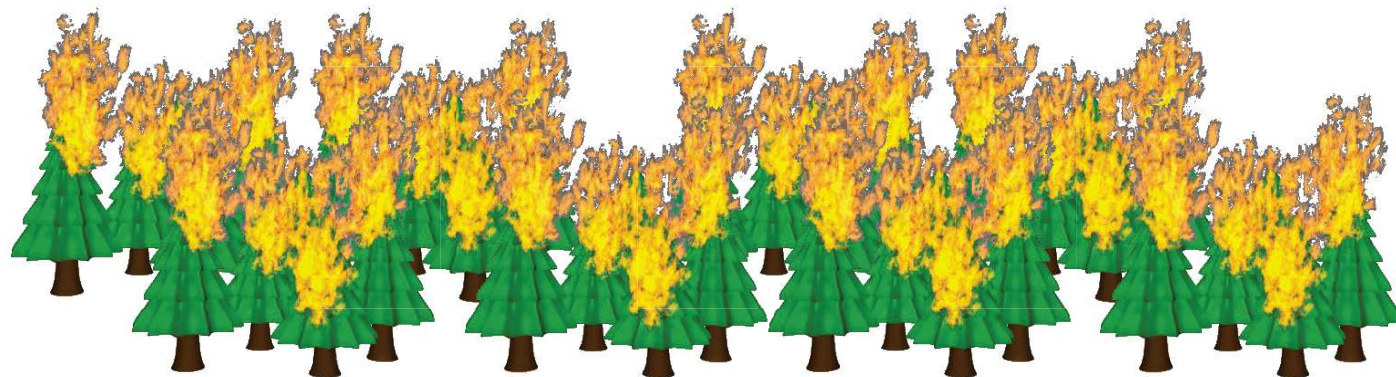
- Le civil
- Le militaire



Applications civiles



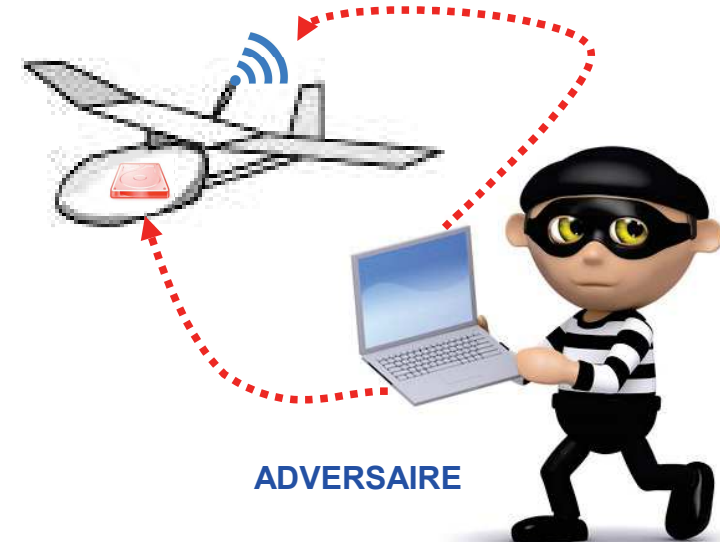
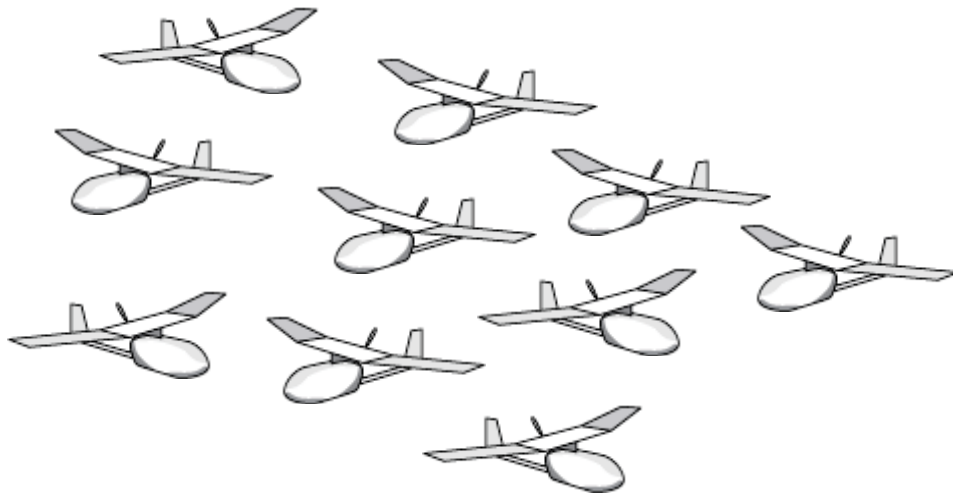
La sécurité peut
ne pas être un problème ?



Applications militaires

- Les drones peuvent stocker et échanger de nombreux **biens** :

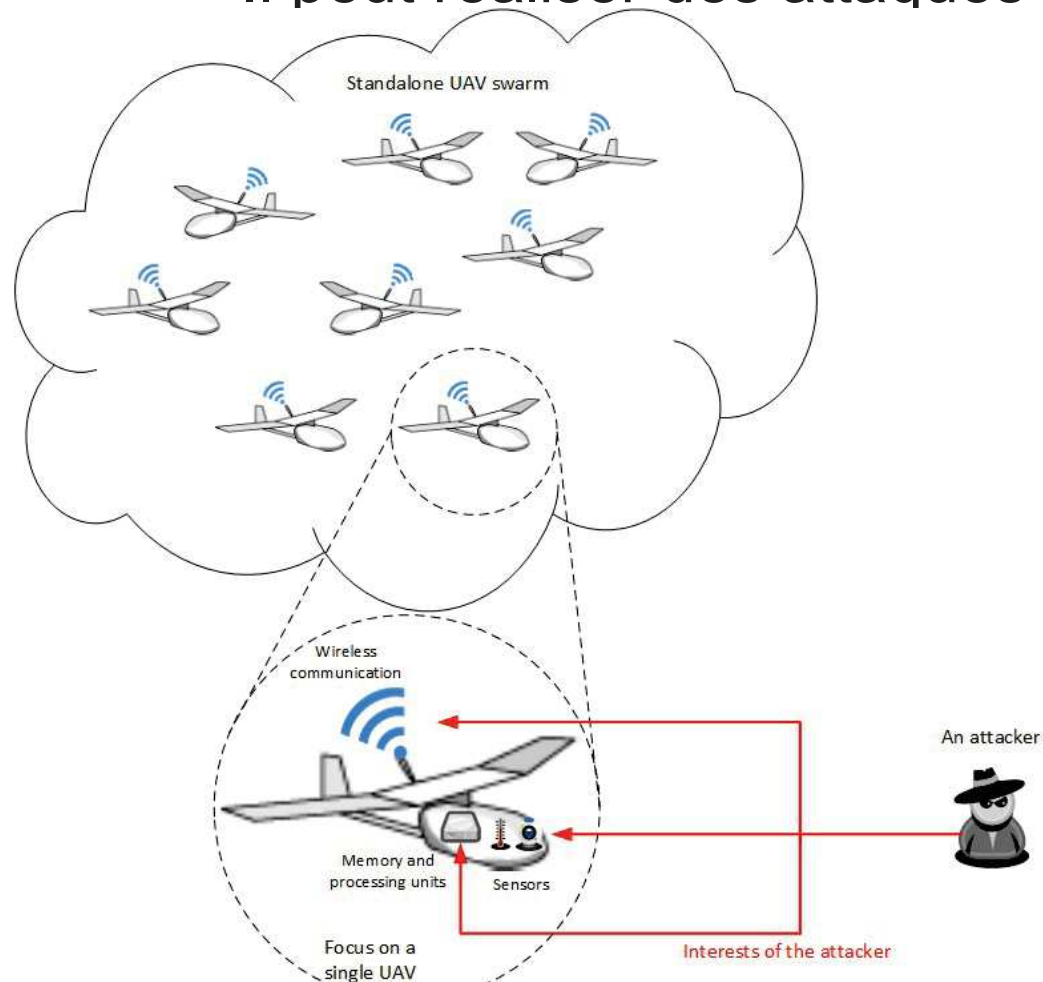
- Plan de vol de la mission
- Photos
- Coordonnées des points d'intérêts (positions des ennemis ou des alliés, ...)



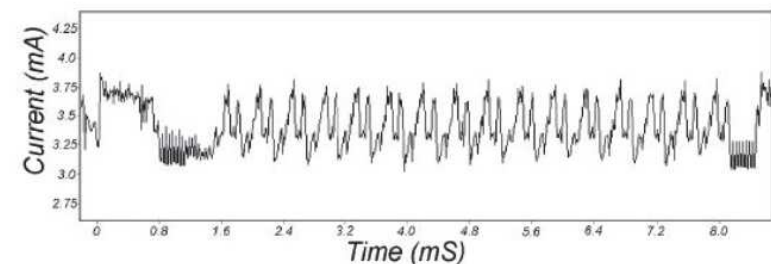
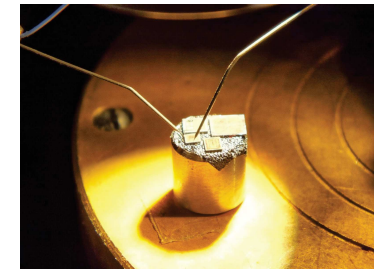
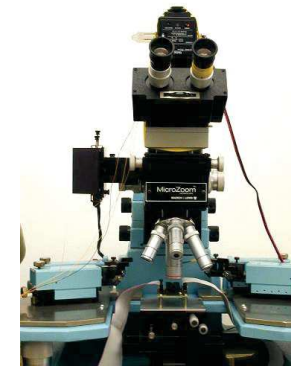
La sécurité **est** une préoccupation !

Notre modèle de l'attaquant

- Un adversaire avec un haut potentiel d'attaque :
 - Il a les capacités et les connaissances pour capturer un drone dans un état fonctionnel
 - Il peut réaliser des attaques en vol

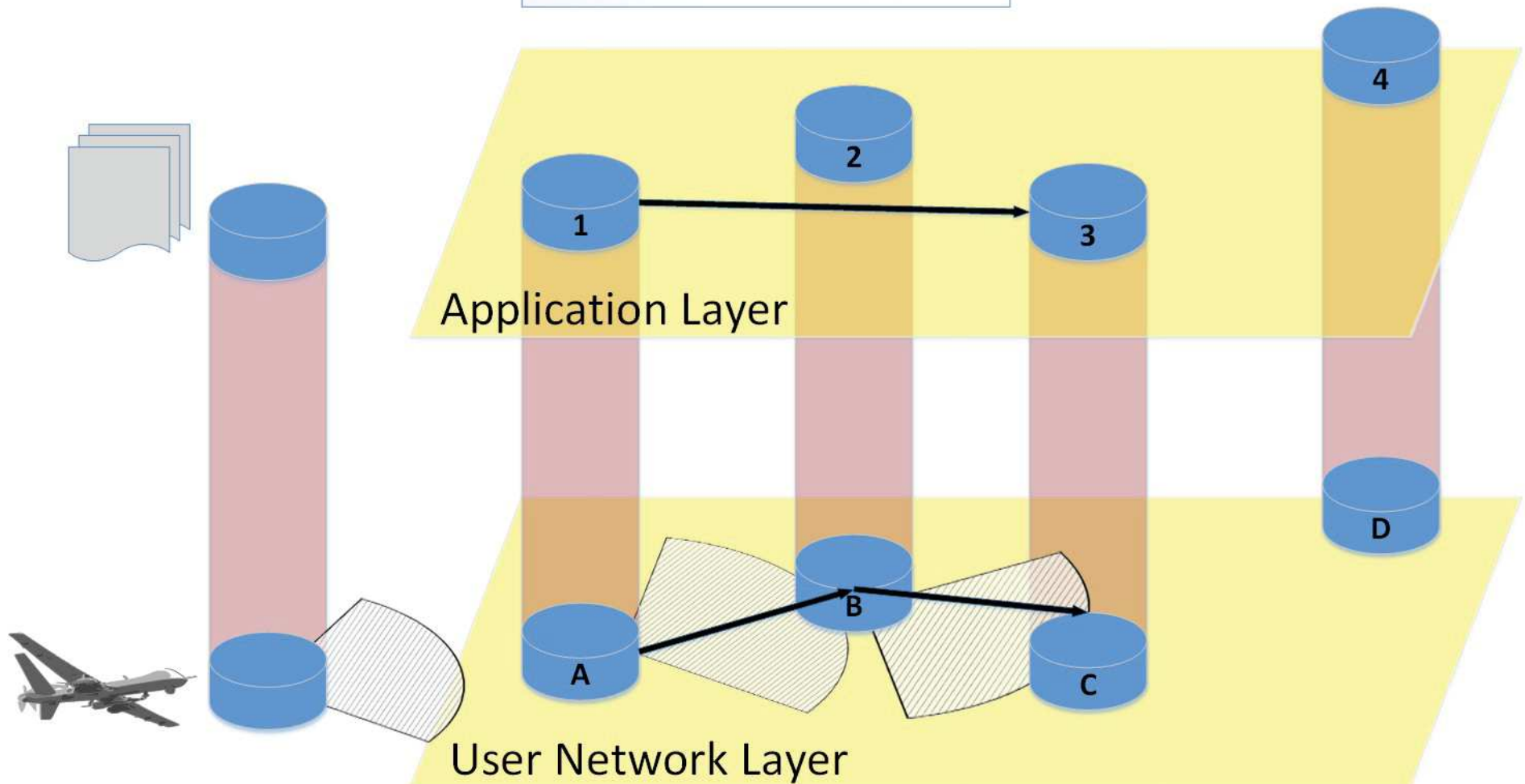


Attaques par canaux auxiliaires, injections de fautes, probing, ...



Flottes de drones

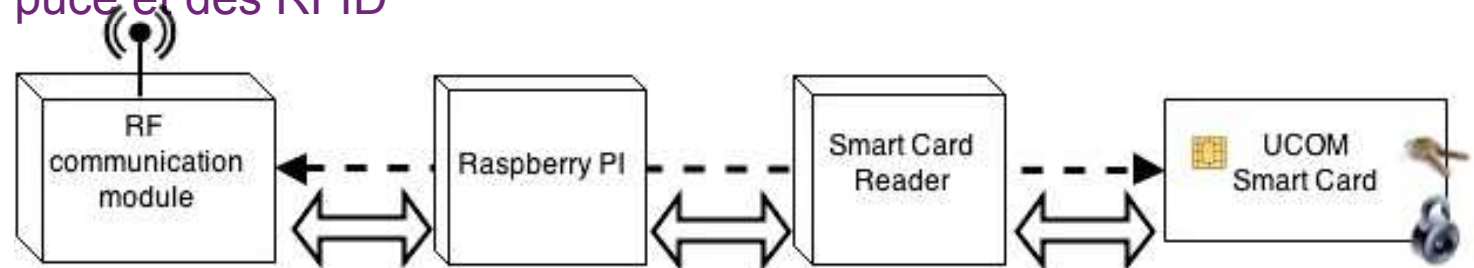
Application Identities: 1, 2, 3, 4
Network Identities: A, B, C, D



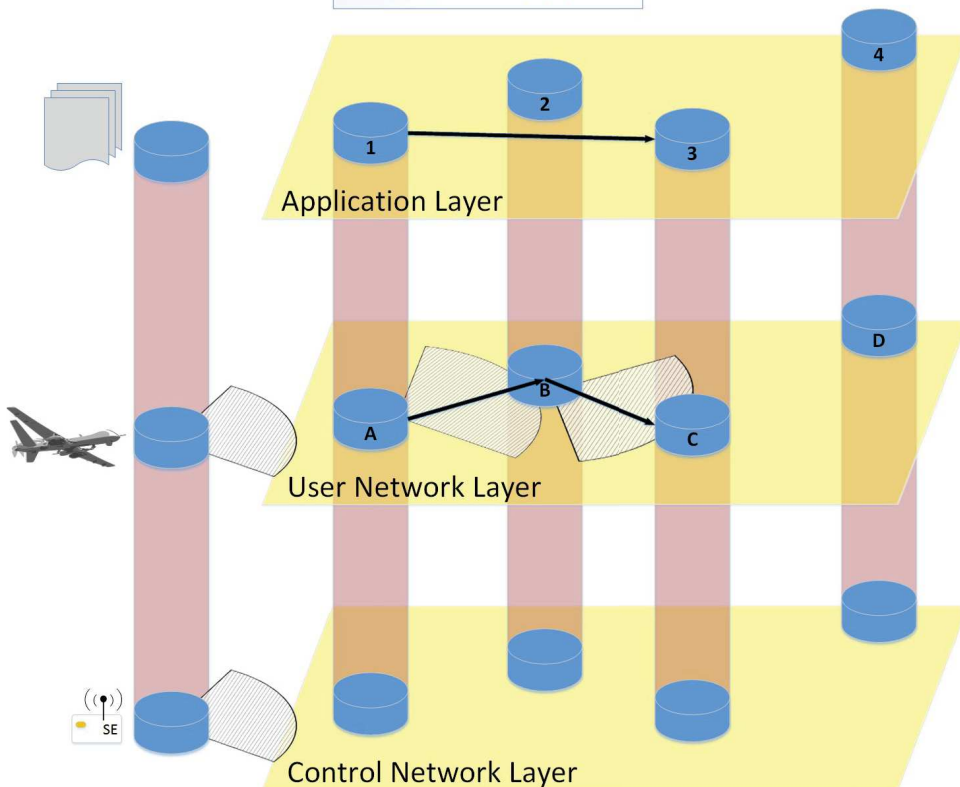
- Les solutions classiques issues des MANETs ne sont pas suffisantes contre **notre modèle de l'attaquant**.

Flottes de drones

- Définition d'un modèle d'adversaire et justification
- Proposition d'un plate-forme de confiance reposant sur un **élément de sécurité**, appelé « Active Radio Frequency Smart Secure Device » (ARFSSD) reprenant le meilleur de la carte à puce et des RFID



Application Identities: 1, 2, 3, 4
Network Identities: A, B, C, D



TrustCom 2016

Financé par la région Limousin
SFD: Security of Fleets of Drones

Communications « Smart Objects » et Drones

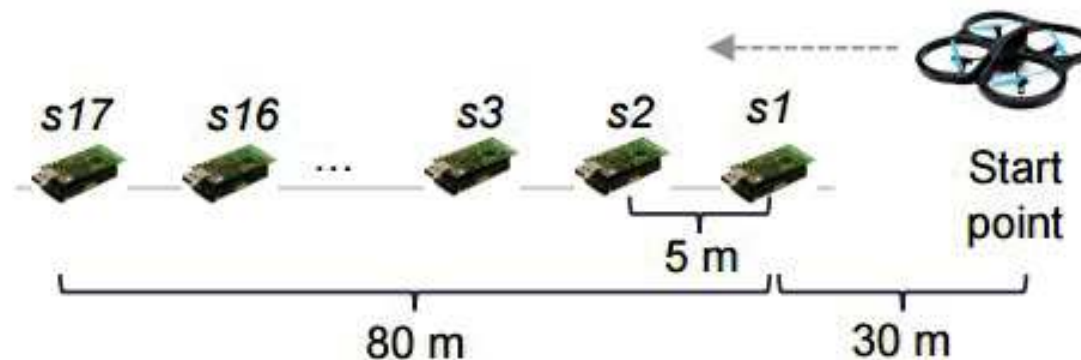
- Amélioration et implémentation en cours de :
 - *A Secure Communication Protocol for Drones and Smart Objects* (ASIACCS'15 – équipe de E. Bertino)



AR.Drone 2.0



TelosB



Principe de efficient Certificateless Signcryption Tag Key Encapsulation Mechanism (eCLSC-TKEM)

- Le KGC choisit une courbe, génère sa master key x et calcule sa clé publique $P_{pub} = xP$ et des fonctions de hachage (H_0, H_1, H_2, H_3)
 - Il publie tout (sauf x)
- Chaque entité calcule une partie de sa clé privée x_A et la clé publique correspondante $P_A = x_A P$
- Le KGC va calculer l'autre partie de clé privée d_A de chaque entité qui lui aura envoyé son ID_A , et son P_A valable pour une période de temps t_A selon :
 - Chooses $r_A \in \mathbb{Z}_q^*$ and computes $R_A = r_A P$.
 - Computes $d_A = r_A + xH_0(ID_A, R_A, P_A, t_A) \bmod q$.

The partial private key of the entity A is d_A . The entity can validate its private key by checking whether $d_A P = R_A + H_0(ID_A, R_A, P_A, t_A)P_{pub}$ holds.
- La clé privée complète de chaque entité est : $sk_A = (d_A, x_A)$
- La clé publique complète de chaque entité est : $pk_A = (P_A, R_A)$

Suite

- Le drone (B) envoie $M1 = \{ID_B, P_B, R_B, t_B\}$
- Le capteur (A) exécute l'algo de génération de la clé symétrique K

- Chooses $l_A, s_A \in \mathbb{Z}_q^*$ and computes $U = l_A P, V = s_A P$.
- Computes $Y = R_B + H_0(ID_B, R_B, P_B, t_B) \cdot P_{pub} + P_B$,
 $T = s_A \cdot Y (= s_A \cdot (H_0(ID_B, R_B, P_B, t_B) \cdot P_{pub} + R_B + P_B))$
and $K = H_1(Y, V, T, ID_B, P_B)$.
- Outputs K and the intermediate information $\omega = (l_A, s_A, U, V, T, ID_A, pk_A, sk_A, ID_B, pk_B, t_B)$.

- Le capteur chiffre le message avec K pour obtenir Γ et calcule

- Computes $H = H_2(U, \tau, T, ID_A, P_A, ID_B, P_B)$,
 $H' = H_3(U, \tau, T, ID_A, P_A, ID_B, P_B)$ and $W = d_A + l_A \cdot H + x_A \cdot H'$
- Outputs $\varphi = (U, V, W)$.

- Le drone reçoit : $M2 = \{ID_A, P_A, R_A, t_A, U, V, W, \tau\}$

- Computes $Y = (d_B + x_B) \cdot P (= (r_B + x H_0(ID_B, R_B, P_B, t_B) + x_B) \cdot P = R_B + H_0(ID_B, R_B, P_B, t_B) \cdot P_{pub} + P_B)$,
 $T = (d_B + x_B) \cdot V (= (d_B + x_B) \cdot s_A P = s_A \cdot Y)$.
- Computes $H = H_2(U, \tau, T, ID_A, P_A, ID_B, P_B)$ and $H' = H_3(U, \tau, T, ID_A, P_A, ID_B, P_B)$.

- If $W \cdot P = R_A + H_0(ID_A, R_A, P_A, t_A) \cdot P_{pub} + H \cdot U + H' \cdot P_A$, outputs $K = H_1(Y, V, T, ID_B, P_B)$.

Otherwise, outputs an invalid encapsulation error. The correctness of the above equation is as follows:

$$\begin{aligned}
 W \cdot P &= (d_A + l_A \cdot H + x_A \cdot H') \cdot P \\
 &= d_A \cdot P + l_A \cdot P \cdot H + x_A \cdot P \cdot H' \\
 &= (r_A + x H_0(ID_A, R_A, P_A, t_A)) \cdot P + U \cdot H + H' \cdot P_A \\
 &= R_A + H_0(ID_A, R_A, P_A, t_A) \cdot P_{pub} + H \cdot U + H' \cdot P_A
 \end{aligned}$$



Conclusion et perspectives

Conclusion et perspectives

- Cartes à puce
 - Développement de l'écosystème user-centric (TEM, FIP, etc.)
 - Utilisation pour les drones (application ~ mission)
- Flottes de drones
 - Utilisation du network coding
 - Utilisation de protocoles Serverless pour des communication entre drone « smart objects »
- Protocoles serverless
 - Développement de protocoles découlant de celui présenté et implémentation sur WISP (TinyECC) et sur un lecteur avec μ ECC





Questions ?