# LOG430 – Architecture logicielle

# Groupe 01

# < Sécurité > – Justification (en classe)

Une personne tente d’utiliser une carte qui ne lui appartient pas. Elle tente de faire plusieurs fois le nip pour accéder au compte via l’interface de l’ATM. Le système de sécurité de l’ATM n’autorise que trois mauvaises tentatives après quoi il garde la carte et envoye une notification à la banque. La banque offre ce service de sécurité afin de protéger ses clients dans le cas où il perdrait leur carte et qu’une personne mal intentionné tenterait de faire plusieurs tentatives à un ATM pour accéder au compte.

# <Sécurité> – Scénario 1 (en classe)

**Source**: Personne mal intentionné qui se serait emparé de la carte

**Stimuli**: La personne essai d’accéder au service de retrait d’argent

**Artéfact**: Service d’authentification de l’ATM

**Environnement**: L’ATM est connecté au serveur de la banque et est pleinement opérationnelle

**Réponse**: Le système empêche l’accès sans autorisation au service et aux données et il enregistre le nombre de tentative incorrect afin de limiter le nombre de tentative

**Mesure de la réponse**: 3 tentatives avec un nip incorrect avant que l’attaque soit détecté. L’utilisateur à 3/10000 de faire le bon nip

# <Sécurité> – Tactique 1 (en classe)

**Description**: Resist attacks – Authenticate Actors

**Justification**: Afin de s’assurer que la personne qui détient la carte est belle et bien le propriétaire de la carte elle doit saisir un nip à 4 chiffres

# <Sécurité> – Tactique 2 (en classe)

**Description** React to attacks – Revoke acces

**Justification**: Lors de la saisie de trois nip incorrect à l’ATM la carte sera conservée par l’ATM.

# <Sécurité> – Tactique 3 (en classe)

**Description** Recover from attacks

**Justification**: Contacter le propriétaire de la carte pour s’assurer de la légitimité des accès et lui renvoyer une carte en cas de besoin

# <Sécurité> – Vue Architecturale (en classe)

# 

# Description du diagramme

L’ATM lie la carte puis affiche l’interface de saisie du nip à l’utilisateur. L’utilisateur saisie un nip. Le module d’authentification de l’ATM va être notifié et initié une communication avec le serveur de la banque via le module de communication afin de valider le nip. Le module d’authentification sur le serveur reçoit la requête et vérifie dans la base de données que le nip correspond à la carte.

# Table de description des éléments du diagramme

|  |  |
| --- | --- |
| Lecteur de carte | Périphérique physique qui permet à l’utilisateur d’insérer une carte dans l’ATM |
| Saisie du nip | Interface utilisateur qui permet à l’utilisateur de saisir un nip |
| Authentification | Module qui gère l’authentification d’un utilisateur |
| Communication | Module qui gère les communications entre l’ATM et le serveur |
| Authentification | Module qui valide l’authentification d’un utilisateur |
| Audit | Module qui gère les logs |
| Notification | Module qui gère les notifications du système |
| Base de données | Logiciel qui s’occupe de la gestion des données |

# Tactique 1

L’interface de saisie du nip et le module d’authentification corresponde à la tactique « Authenticate Actors » afin de s’assurer que la personne qui utilise la carte est bien le propriétaire de la carte.

# Tactique 2

Le lecteur de carte permet l’application de la tactique « Revoke access ». Dans le cas où un utilisateur saisi trois nip incorrect le lecteur de carte conserve la carte.

# Tactique 3

Le module de notification correspond à la tactique « «Recover from attacks » qui permet de notifier la banque qui notifie à son tour le propriétaire de la carte lorsqu’un lecteur de carte conserve une carte à cause de trois nip incorrects.

# Equipe 02

LOG430 – Architecture logicielle

Scénarios de qualité, tactiques et justifications des attributs produits par les équipes en classe

# <Sécurité> – Justification

La banque veut prévenir les intrusions dans ses systèmes pour protéger l’intégrité et la confidentialité de ses données. Pour que ses clients soient satisfaits et lui fasse confiance, elle veut aussi mettre en place des mesures pour les cas où la prévention n’a pas été suffisante et les données ont quand même été compromises.

# <Sécurité> – Scénario

**Source :** Fraudeur ou client

**Stimuli :** La carte de crédit est utilisée de manière anormale selon les habitudes du client. La carte de crédit est utilisée dans un délais anormalement court contenu de la position géographie de chaque transaction.

**Artéfact :** Les transactions (*data produced by the system*)

**Environnement :** Complètement opérationnel

**Réponse :** Propriétaire de la carte est appelé afin de vérifier s'il a utilisé la carte aux deux endroits

**Mesure de la réponse :** Nombre d’heures écoulées entre la détection de la tentative d’intrusion et la résolution du problème. La résolution du problème peut soit être que la personne retrouve l’utilisation de son compte après l’intrusion ou que ce n’était finalement pas une fraude.

# <Sécurité> – Tactique 1

**Description :** *[Limit access]* – Limiter l’accès à la carte

**Justification :**  En attendant d’être en contact avec le client, le système bloque la carte de façon à limiter l'accès à ses ressources, l’argent dans ses différents comptes dans notre cas. Les ressources du client sont donc protégés contre une attaque et pourra se justifier s’il est l’acteur des transactions. Il pourra aussi choisir les paramètres de sécurité le temps et la distance nécessaire pour bloquer son compte.

# <Sécurité> – Tactique 2

**Description :** *[Detect Intrusion]* – Détecter l’intrusion en utilisant un algorithme qui dépend de la distance et du temps entre les retraits

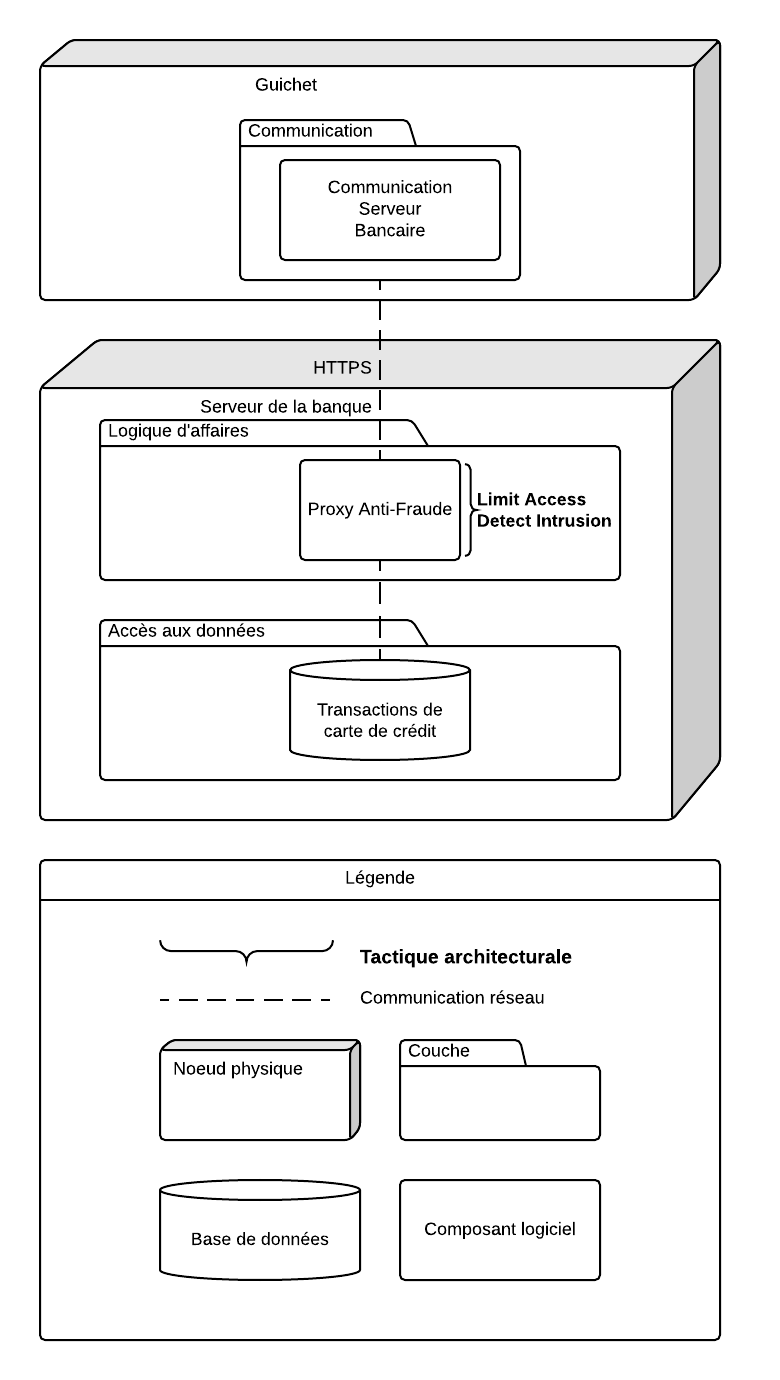
**Justification :** Il faut être capable de détecter efficacement les tentatives des fraude pour permettre de prévenir des dommages considérables aux finances des clients de la banque. Il est préférable d’avoir des *false positives* parfois pour augmenter le taux de tentatives de fraude résolues.

# <Sécurité> – Tactique 3

**Description :** *[Inform Actors]* –Suite a la détection d’une fraude, l’institution financière informe le client d’une brèche potentielle.

**Justification :** Afin d’assurer la satisfaction du client et de protéger les biens de celui-ci et de la banque, le client est informé du problème.

# <Sécurité> – Vue architecturale



La vue architecturale montre que le guichet bancaire communique, via le protocole HTTPS, à un proxy situé sur le serveur bancaire. Ce proxy détermine quand les transactions sont autorisées ou pas en vérifiant les règles incluant la géolocalisation et l’horodatage des transactions liées à la carte de crédit utilisée par le client.

## Éléments du diagramme

|  |  |
| --- | --- |
| **Élément** | **Description** |
| Guichet | Noeud physique représentant les guichets automatiques qui communiquent avec les serveurs bancaires. |
| Guichet / Communication | Couche du guichet permettant d’accéder au serveur bancaire via réseau (HTTPS). |
| Guichet / Communication Serveur Bancaire | Module situé sur le guichet qui gère les connexions et les échanges de données avec le proxy situé sur le serveur bancaire. |
| Serveur de la banque | Noeud physique représentant les serveurs bancaires qui sont utilisés pour traiter les transactions. |
| Lien Guichet – Serveur HTTPS | Le protocole HTTPS est utilisé pour communiquer entre le guichet et le serveur. |
| Serveur / Logique d’affaires | Couche du serveur bancaire contenant toutes les règles d’affaires propres au domaine bancaire. |
| Serveur / Accès aux données | Couche du serveur bancaire comprenant les bases de données et les classes d’accès aux bases de données. |
| Serveur / Proxy anti-fraude | Le proxy anti-fraude, situé dans la couche logique d’affaires, valide certaines conditions avant de donner accès à la base de données transactionnelle dans le but de prévenir la fraude. |
| Lien Proxy – Transactions | Un protocole quelconque, dépendant de l’implémentation choisie pour la base de données, est utilisé par le proxy pour communiquer avec la base de données. |
| Serveur / Transactions de carte de crédit | Base de données contenant les transactions de cartes de crédits. Les données qui y sont contenues sont utilisées pour l’analyse du niveau de menace. |

L’élément *Proxy anti-fraude* est directement lié aux tactiques 1 et 2. Le proxy est donc responsable de vérifier les transactions pour déterminer quand une intrusion est arrivée et de limiter l’accès subséquent aux cartes de crédits qui ont été déterminées comme frauduleuse. La tactique 3 n’est pas présente car elle a une caractéristique plus humaine du fait que quelqu’un devra appeler le client victime de fraude pour lui annoncer la mauvaise nouvelle et valider que c’est bien une fraude.

Les trois tactiques combinées forment une bonne stratégie de sécurité pour répondre à la justification initiale de la banque qui souhaite protéger ses usagers et réagir rapidement en cas de fraude. Il est nécessaire de détecter l’intrusion, limiter l’accès en cas d’intrusion et de notifier les parties prenantes quand une intrusion est détectée.

# Equipe 04

# Scénarios de qualité, tactiques et justifications des attributs produits par les équipes en classe

# < Sécurité > – Justification (en classe)

Un client désire effectuer une transaction bancaire en utilisant une machine ATM et la transaction ne se fait pas comme prévue. La banque détecte une attaque spécifique qui a modifié le message de transaction.

# <Sécurité> – Scénario 1 (en classe)

**Source:** Inconnu (Hacker)

**Stimuli:** Change or delete data : Une personne pas autorisé a fait des changement sur les données de transaction.

**Artéfact:** Data within the system : La somme d’argent déposée par le client se dépose dans un autre compte.

**Environnement:** Le système de transaction est sécurisé par un firewall, est online et connecté a une base de données.

**Réponse:** Data are protected from unauthorized access.

**Mesure de la réponse**: Aucune données devraient être potentiellement vulnérable à être modifiée par une personne non authorizée.

# <Sécurité> – Tactique 1 (en classe)

**Description:** Detect intrusion

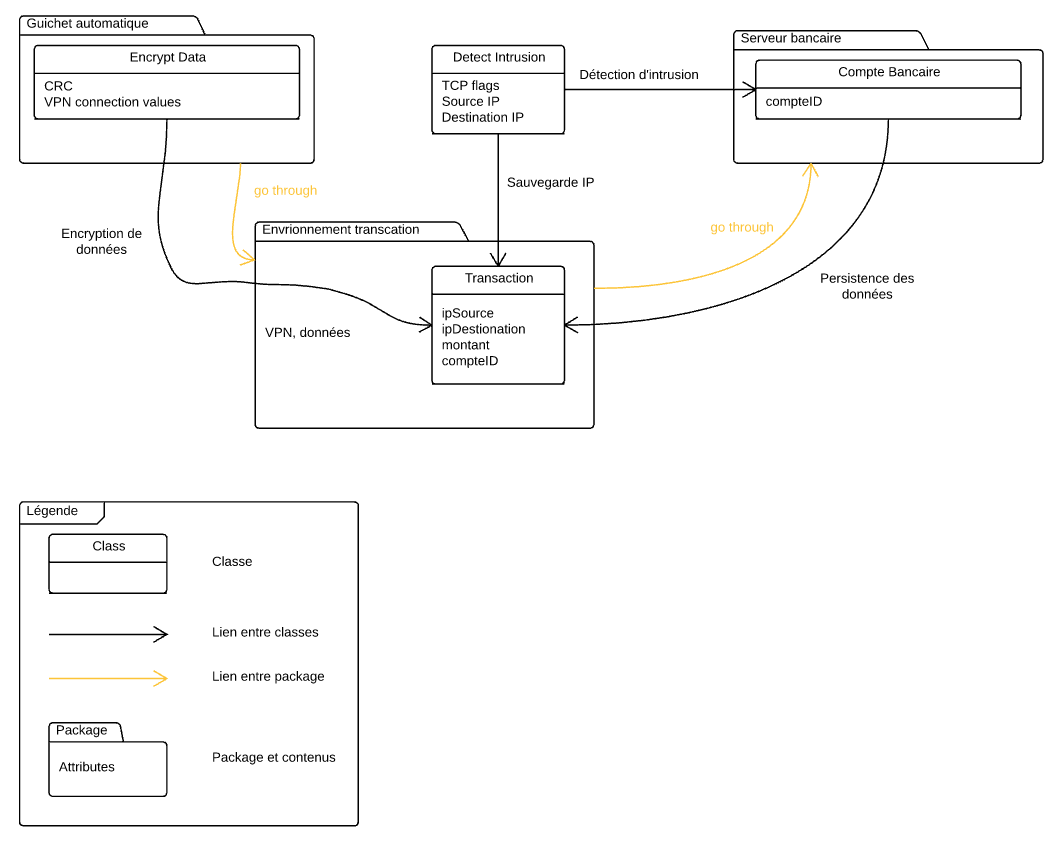
**Justification:** Le système de protection doit identifier l’intégrité de chaque paquet de données afin d’assurer une sécurité à l’aide des identifiants spécifiques des couches du protocole du modèle OSI et d’identifiant de source et destination.

# <Sécurité> – Tactique 1 (en classe)

**Description:** Encrypt data

**Justification:** Pour toute transaction bancaire, un système de protection de donnée doit prévenir et empêcher l’accès non-autorisé aux informations de transaction. L’encryptions protège l’intégrité des informations de chaque paquet.

# <Sécurité> – Vue Architecturale (en classe)



* **Texte de description du diagramme**
* La classe Guichet automatique contient la classe d’encryption de données.
* Le package Environnement Transaction contient la classe transaction.
* La classe Detect Intrusion valide la cohérence valide la source et la destination de la transaction avec les données contenues dans la classe de la banque.
* **Table de description des éléments du diagramme**

|  |  |
| --- | --- |
| Encrypt Data CLASS | Classe qui ajoute à la classe Transaction une encryption de donnée avant l’envoie de transaction. |
| Transaction CLASS | S’occupe d’encapsuler les données d’une transaction spécifique et de l’envoyer dans l’environnement du serveur bancaire afin que les données soit appliqués aux compte bancaires spécifiques. |
| Detect Intrusion CLASS | Cette classe permet de valider la source et destination des données. |
| Compte Bancaire CLASS | Est exprimée comme étant une base de données de l’ensemble des comptes bancaire. |
| Guichet Automatique PACKAGE | Package représentant l’environnement du ATM. |
| Environnement Transaction PACKAGE | Représente l’environnement dans lequel le paquet passera pour aller du ATM au serveur bancaire. |
| Serveur bancaire PACKAGE | Représente l’environnement de la banque, du côté serveur. |
| Encryption de données LINK | Encrypte les données avant qu’ils soit envoyés vers le serveur par le GAB. |
| Sauvegarde Ip LINK | Identifie et sauvegarde les addresses IP au niveau de l’environnement de la transaction. |
| Détection d’intrusion LINK | Accède aux comptes bancaires pour vérifier l’intégrité de la transaction |
| Persistence des données | Accède aux données de la transaction pour la persistence. |

* **Texte décrivant la relation entre les éléments et la tactique**

La classe DetectIntrusion communique avec la classe Transaction et la classe Compte bancaire afin de valide la cohérence entre le compte source et destination.

La classe Encrypt Data communique avec la classe transaction afin d’ajouter une protection au message de transaction avant sa transmission.

* **texte décrivant comment la tactique répond a chacun des scénarios de qualité.**

Les technique Detect Intrusion ainsi que Encrypt Data répond a chacun des scénarios de qualité puisque : La tactique Detect Intrusion detecte la provenance ou les transaction sont faites afin de faciliter la detection de problèmatique. Comme par exemple, dans le cas spécifique, on est en mesure de déterminer que la somme d’argent déposé par le client est déposé dans un autre compte que celui que le client désire. Pour ce qui est de la tactique Encrypt, elle est une tactique fondamentale pour protéger les données des personnes non-authorizé comme il est arrivé dans le cas spécifique (hacker).

# Equipe 05

Scénarios de qualité, tactiques et justifications des attributs produits par les équipes en classe

< Sécurité > – Justification (en classe)

Au bout de 3 mauvaises tentatives de NIP le guichet ne retourne pas la carte au client et l'invite à aller la récupérer au comptoir de la banque.

<Sécurité> – Scénario 1 (en classe)

**Source**: Un utilisateur qui tente d'utiliser le GAB avec un mauvais NIP.

**Stimuli**: Trop de tentative de NIP de la part de l'utilisateur

**Artéfact**: Le service d'accès aux comptes bancaires

**Environnement**: Pleinement opérationnel

**Réponse**: Le GAB garde la carte pour protéger les informations du client

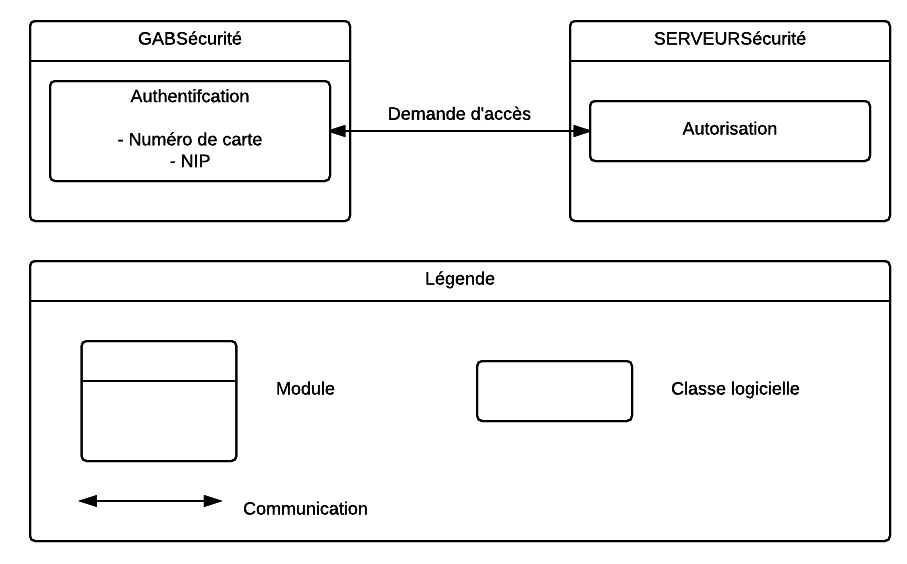
**Mesure de la réponse**: Un message d'erreur est affiché pour inviter l'utilisateur à récupérer la carte au comptoir de la banque.

<Sécurité> – Tactique 1 (en classe)

**Description**: Authentification des acteurs

**Justification**: L'authentification par le NIP permet de s'assurer que l'utilisateur du GAB est bien le propriétaire de la carte bancaire.

<Sécurité> – Vue Architecturale (en classe)



Texte de description du diagramme

* Le diagramme précédent représente l’interaction qu’il y a entre un guichet automatique bancaire et un serveur du point de vu de la sécurité. On peut remarquer que le guichet bancaire s’occupe de lire le numéro de carte et le NIP entré par l'utilisateur qui est immédiatement envoyé au serveur pour vérification. Le serveur vérifie la correspondance entre les 2 informations rend son verdict sur l'autorisation d'accès de du client.
* Table de description des éléments du diagramme
* **GABSécurité**: Module de sécurité du GAB
* **SERVEURSécurité**: Module de sécurité du serveur bancaire
* **Authentification**: Classe logicielle récolte les informations du client
* **Autorisation**: Classe logicielle vérifie la concordance des données reçues.
* **Demande d'accès**: Lien de communication entre le GAB et le serveur.

# Équipe 7

**< Sécurité > – Justification (en classe)**

La sécurité est essentielle pour un GAB. Les clients désirent avoir confiance que leurs fonds ne seront pas disponibles pour une personne malicieuse. S'il ne se sentent pas en confiance, il n'utiliseront probablement pas le système.

**<Sécurité> – Scénario 1 (en classe)**

**Source**: Un groupe d'individus qui a en main les informations bancaires de clients du GAB

**Stimuli**: Tentative de retrait d'argent dans des endroits éloignés dans un court laps de temps

**Artéfact**: Données bancaires des clients

**Environnement**: Runtime, opération normale du GAB

**Réponse**: Bloquer l'accès aux fonds, bloquer la carte du client et avertir la banque d'un possible clonage de carte

**Mesure de la réponse**: Le nombre d'attaques en dehors du comportement normal d'opération

**<Sécurité> – Tactique 1 (en classe)**

**Description** : *DetectIntrusion*

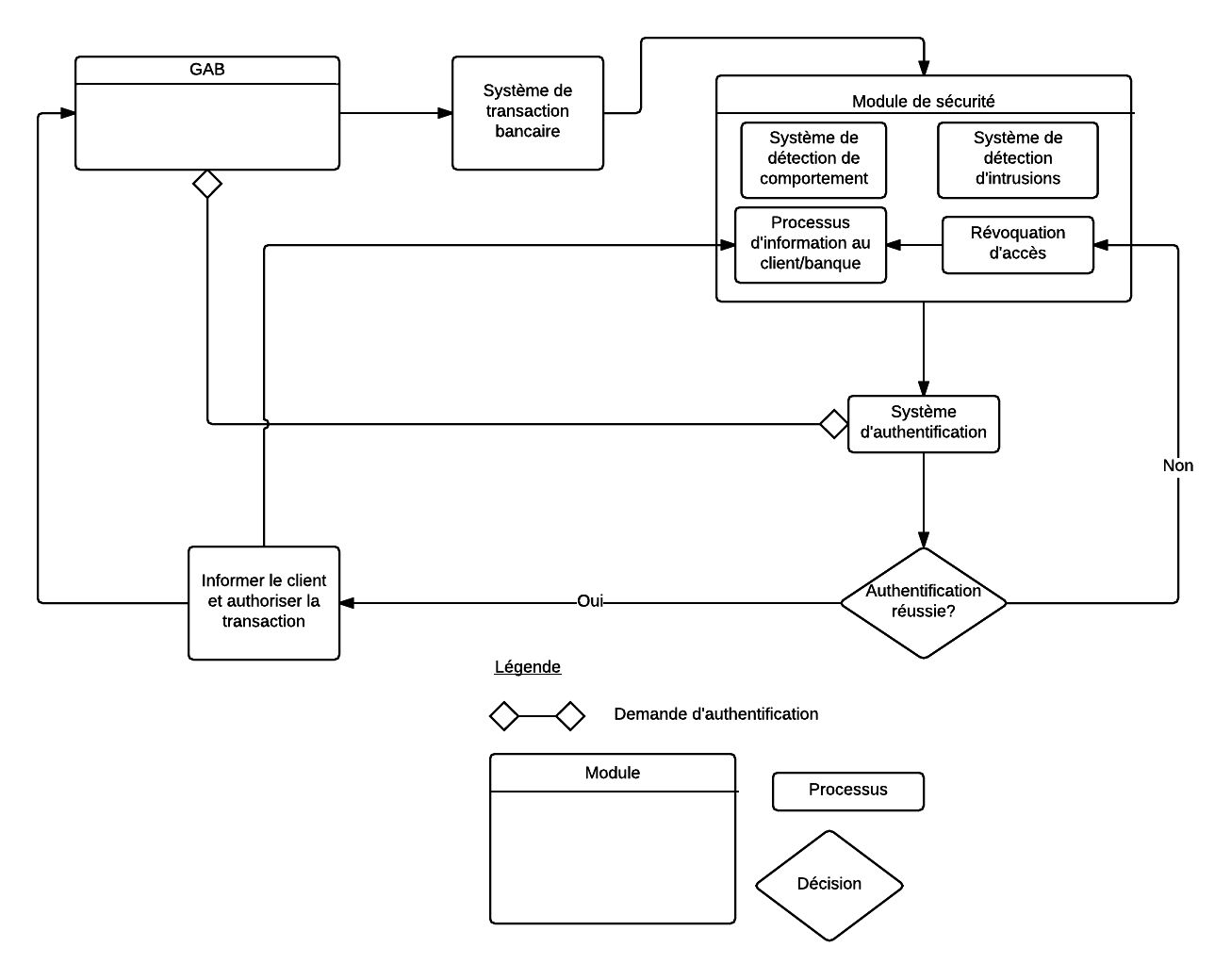
**Justification**: On détecte un comportement anormal en vérifiant si les transactions de retrait sont hors normes (transactions dans des endroits éloignées, gros montants, beaucoup de transactions rapides, etc.).

**<Sécurité> – Tactique 2 (en classe)**

**Description**: *Identify actors, Revoke Access*

**Justification**: Lorsqu'une attaque a été détectée, le système utilisera des tactiques pour vérifier l'identité du client (2FA, date de naissance, etc.). Si la personne malveillante ou l'utilisateur ne réussit pas à s'authentifier correctement, le compte sera bloqué jusqu'à ce que la banque contacte le client et règle le problème

**<Sécurité> – Vue Architecturale (en classe)**



Le diagramme représente le flux de travail d’un GAB qui tente de prévenir les intrusions en communiquant avec le serveur bancaire pour vérifier si le comportement de l’utilisateur est hors des normes. Si une intrusion est détectée, le GAB tentera d’authentifier l’utilisateur avant de procéder aux transactions bancaires. Sinon, le compte client sera bloqué, et le client ainsi que la banque en seront notifiés.

**Table de description des éléments du diagramme**

|  |  |
| --- | --- |
| **Élément** | **Description** |
| Système de transaction bancaire | Système qui effectue les transactions bancaires usuelles |
| GAB | Solution logicielle implémentée dans le guichet automatique bancaire |
| Module de sécurité | Module qui détermine si une intrusion est possible selon le comportement des utilisateurs |
| Système d’authentification | Système qui déclenche une authentification sur le GAB en cas de comportement anormal. |

Le module de sécurité analyse le comportement de l’utilisateur dans le but de détecter les intrusions. Le système d’authentification, lui, sert à effectuer le processus d’autorisation avec le GAB.

En gardant une trace de l’utilisation normale que les clients font du GAB dans le module de sécurité, il est possible de vérifier si l’utilisation du GAB qui en présentement faîte est dans les normes pour ce client en particulier. Ceci augmente la sécurité en donnant une barrière d’authentification si des comportements anormaux sont détectés.