LOG430 – Architecture logicielle

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

[La vitesse de réponse du serveur est diminuée à cause d’un grand nombre de transactions simultanées provenant de guichets.](#_30j0zll)

[Le temps requis pour authentifier le client avec sa carte et son code NIP dépasse les seuils acceptables.](#_1fob9te)

[Validation des transactions avec le serveur](#_4d34og8)

[Communication avec le serveur](#_17dp8vu)

[Accéder à la liste des comptes d’un utilisateur](#_35nkun2)

[Servir plusieurs ATM en parallèle](#_1ksv4uv)

[Latence de la communication avec le serveur dû l’encryptage des messages trop lourd](#_3j2qqm3)

[Surcharge de transactions sur le serveur](#_1y810tw)

[Temps de calcul pour le serveur trop long pour une opération](#_qsh70q)

[Latence entre la machine et la réponse du serveur](#_49x2ik5)

Performance – Justification

La raison d’être d’un système de guichet automatique est de simplifier et d’accélérer les opérations bancaires. À cet effet, la performance est un facteur de succès important pour plusieurs intervenants.

Pour les clients des banques, il s’agit d’effectuer les transactions rapidement. L’utilisation d’un guichet automatique est routinière, les pertes de temps induites par la lenteur du système s’accumuleraient donc rapidement et seraient source de frustrations. Le client veut également obtenir une réponse immédiate: une rétroaction retardée peut causer un malaise, notamment lorsqu’il est question d’argent.

Pour l’administration des banques, la rapidité du guichet automatique est directement proportionnel au volume de transactions qu’il est possible de traiter. Un système performant pourra gérer plusieurs transaction concurrentes. En somme, un volume important signifie une augmentation des revenus des frais d’opérations. De plus, la performance du système a un impact direct sur la satisfaction de la clientèle, un facteur important qui s’inscrit dans un processus de fidélisation.

Performance – Scénario 1

# La vitesse de réponse du serveur est diminuée à cause d’un grand nombre de transactions simultanées provenant de guichets.

| Scénario | La vitesse de réponse du serveur est diminuée à cause d’un grand nombre de transactions simultanées provenant de guichets. | |
| --- | --- | --- |
|  |
| Objectifs d'affaires | Temps de réponse moyen inférieur à 3 secondes | |
| Attribut de qualité | Performance | |
| Source | Externe : les délais sont causés par des une grande nombre de transactions issues d’autres guichets. | |
|  |  |  |
| Stimulus | Stochastique : les périodes de pointe se déroulent à des moments assez prévisibles. | |
| Artéfact | Guichet automatique; | |
|  | Serveur bancaire | |
| Environnement | Opération en période de pointe | |
| Réponse | Les transactions sont traitées, mais après un certain délai. | |
| Mesure de la | Latence moyenne de 2 secondes pour la réponse du serveur bancaire; | |
| réponse | Marge de 1 seconde sur le retour de la réponse | |
| Questions | 1. Quand ont lieu les périodes de pointe? Combien de transactions sont envoyées durant ces périodes? | |
|  |
|  | 2. | Combien de transactions simultanées peut traiter une instance de serveur? |
|  | 3. | Combien d’instances dupliquées de serveur sont nécessaires? |

Performance – Scénario 2

# Le temps requis pour authentifier le client avec sa carte et son code NIP dépasse les seuils acceptables.

| Scénario | Le temps requis pour authentifier le client avec sa carte et son code NIP dépasse les seuils acceptables. |
| --- | --- |
|  |
| Objectifs d'affaires | Le temps entre l’insertion de la carte et l’authentification du client doit être inférieur à 5 secondes  . |
|  |
| Attribut de qualité | Performance |
| Source | Interne: Le délai de déchiffrement des informations bancaires est causé par le module dédié sur le guichet. |
|  |

|  | Externe : Le délai d’authentification est aussi causé par une période de pointe sur le réseau bancaire. | |
| --- | --- | --- |
|  |
| Stimulus | Arrivée d’une transaction (périodique et fréquent) | |
| Artéfact | Sous ­système responsable du chiffrement des transactions; | |
|  | Serveur bancaire | |
| Environnement | Opération en période de surcharge | |
| Réponse | Le client est authentifié, mais après un certain délai. | |
| Mesure de la | Délai butoir de 500 millisecondes pour le déchiffrement des informations de la carte bancaire; | |
| réponse | Marge de 250 millisecondes pour le déchiffrement des informations de la carte bancaire; | |
|  | Latence moyenne de 2 secondes pour la réponse du serveur bancaire; | |
|  | Marge de 1 seconde sur le retour de la réponse | |
| Questions | 1. | Quel algorithme de chiffrement sera choisi? |
|  | 2. | Quel compromis sommes ­nous prêt à prendre en ce qui a trait à la performance et la sécurité dans le choix de l’algorithme? |
|  |  |
|  | 3. | Combien de transactions simultanées peut traiter une instance de serveur? |
|  | 4. | Combien d’instances dupliquées de serveur sont nécessaires? |

Performance – Tactiques pour le scénario 1

**Description**: ​*Prioritize events*

**Justification**: En période de pointe, un très grand nombre de guichets automatiques envoient des requêtes de façon quasi simultanée aux serveurs de la banque. Il serait donc pertinent de contrôler la demande des ressources serveur en priorisant les requêtes tout en introduisant un mécanisme de traitement de ces dernières selon leur priorité. Par exemple, une requête qui représente un retrait serait plus important qu’une requête de dépôt, considérant que le dépôt nécessite une certaine validation. De plus, une requête d’interrogation de solde est moins critique qu’un dépôt ou un retrait.

**Description**: ​*Maintain multiple copies of computation*

**Justification**: En période de pointe, le serveur bancaire constitue un goulot d’étranglement. Ainsi, afin de mieux gérer les ressources serveur, il serait pertinent de répliquer les instances de serveur et d’introduire un répartiteur de charge afin de limiter la charge individuelle de chacun des serveurs répliqués tout en permettant un traitement efficace de la masse de requêtes. Le mécanisme utilisé par le répartiteur distribuerait ainsi les requêtes entrantes au serveur avec le moins de charge.

Performance – Tactiques pour le scénario 2

**Description**: ​*Reduce overhead*

**Justification**: Il serait pertinent de réduire au minimum le nombre d’intermédiaires utilisés lors du chiffrement et du déchiffrement des informations contenues sur la carte bancaire du client. En effet, ce module devrait être particulièrement efficace puisque l’authentification et par conséquent le déchiffrement des informations bancaires est une opération sensible au temps. En regroupant les ressources logicielles nécessaires pour effectuer le travail de déchiffrement sous le même composant, on peut empêcher d’avoir à faire un appel coûteux à un autre composant.

**Description**: ​*Increase resource efficiency*

**Justification**: L’algorithme de chiffrement employé par la carte bancaire et implémenté dans la logique de déchiffrement devrait être le plus efficace possible, compte tenu du fait que l’opération de déchiffrement est sensible au temps. Cependant, il ne faut pas non plus compromettre la sécurité pour la performance dans le choix de l’algorithme.

**Description**: ​*Prioritize events*

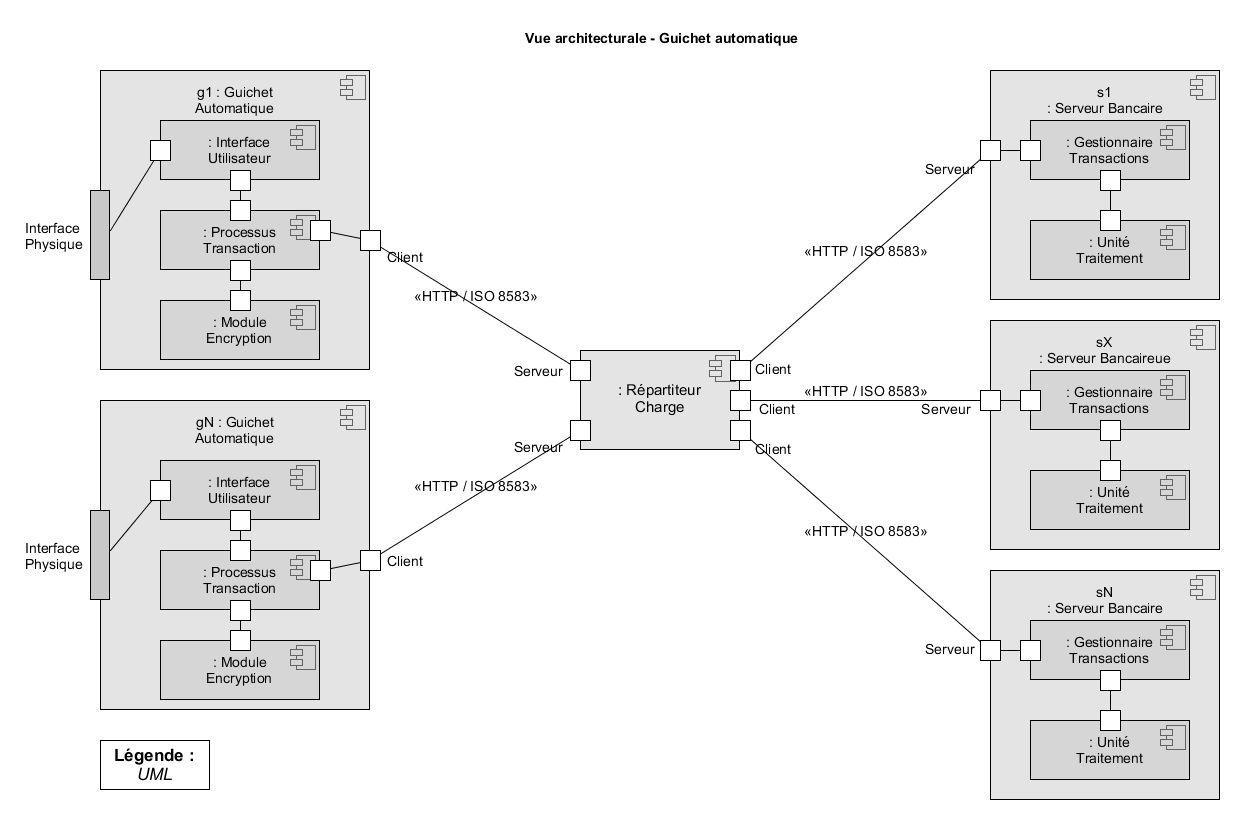
**Justification**:En période de pointe, un grand nombre de transactions et de requêtes provenant des différents guichets sont acheminées vers les serveurs bancaires. Il est donc pertinent de prioriser ces requêtes afin de traiter plus rapidement celles qui sont les plus critiques et ainsi diminuer la latence pour ces requêtes. Le mécanisme de priorisation préconisé est basé sur le type de transaction et de requête.

**Description**: ​*Maintain multiple copies of computations*

**Justification**: En période de pointe, le serveur bancaire constitue un goulot d’étranglement puisque plusieurs guichets lui acheminent des transactions et des requêtes. Il est donc pertinent d’introduire des répliquesde serveurs bancaires identiques en fonctionnement et un répartiteur de charge pour acheminer les requêtes vers les serveurs répliqués. Ce répartiteur acheminera ainsi les requêtes vers le serveur le moins occupé.

Performance – Vue architecturale

Diagramme



Description des éléments

| Élément | Description |
| --- | --- |
|  |  |
| Guichet automatique | Composant représentant le terminal avec lequel les clients interagissent pour compléter des transaction bancaires avec leur carte bancaire. Chacune des instances de guichet est identique au niveau du fonctionnement. |
|  |
|  |
|  |  |
| Interface utilisateur | Composant interne du guichet représentant les périphériques avec lesquels l’utilisateur fera interface, soit le clavier numérique, l’écran et le lecteur de carte. |
|  |
|  |
|  |  |
| Processus de | Composant interne du guichet responsable de traiter les entrées du client et de créer des transactions qui seront acheminées vers les serveurs bancaires. |
| transactions |
|  |  |
| Module de chiffrement | Composant internet du guichet responsable de déchiffrer |
|  |  |
| Répartiteur de charge | Composant représentant un module qui gère la répartition des transactions provenant de différents guichets automatiques vers les serveurs bancaires répliqués. |
|  |
|  |
|  |  |
| Serveur bancaire | Composant représentant un serveur qui traite des transactions bancaires provenant du répartiteur de charge. Chacune des instances de serveur est identique au point de vue du fonctionnement et leur charge est gérée par le répartiteur. |
|  |
|  |
|  |
|  |  |
| Gestionnaire de | Composant interne du serveur responsable de trier les transactions en |
| transactions | fonction de leur priorité (soit le type de transaction). Une fois triées, les |
|  | transactions sont envoyées une après l’autre à l’unité de traitement. |
|  |  |
| Unité de traitement | Composant interne du serveur responsable de compléter les transactions. Il reçoit ces transactions du gestionnaire de transactions, les complète et renvoie une réponse correspondante au gestionnaire, pour que ce dernier l’achemine vers le guichet en remontant suivant le parcours inverse de la requête. |
|  |
|  |
|  |
|  |
|  |  |
| Connecteur réseau | Connecteur établissant un lien entre les clients et les serveurs sur le réseau reliant les guichets automatiques et les serveurs bancaires. Ce connecteur emploie le protocole HTTP pour la connexion réseau et le protocole ISO 8583 pour la forme des messages de transaction bancaire. |
| HTTP / ISO 8583 |
|  |
|  |
|  |  |

Relation entre les éléments et les tactiques

*Prioritize events*

Le processus de transactions du guichet automatique génère des transactions suite à l’interaction avec le client et celles­-ci sont acheminées vers les serveurs bancaires. Ces transactions respectent la forme décrite par la norme ISO 8583 (type de transaction, destinateur, destinataire, montant, etc). C’est le type de transaction qui sera utilisé plus tard pour déterminer la priorité des transactions.

Le gestionnaire de transactions ordonne les transactions adressées à l’instance de serveur duquel il fait partie en fonction de leur type. Il maintient une file de transactions en attente de traitement et achemine à l’unité de traitement les prochaines transactions dans la file. L’unité de traitement s’occupe de traiter les transactions que lui achemine le gestionnaire de transactions et de renvoyer une réponse suite à leur complétion.

*Maintain multiple copies of computation*

Le répartiteur de charge a comme responsabilité de gérer la charge des transactions qui est acheminée à chacune des instances des serveurs bancaires. Pour s’y faire, il surveille le nombre de transactions que chaque serveur a présentement dans sa file avant d’acheminer la prochaine transaction au serveur le moins occupé.

Les serveurs bancaires sont des instances dupliquées d’un serveur afin d’augmenter le débit de transaction pouvant être traité simultanément. Ils sont identiques en ce qui a trait au fonctionnement.

*Reduce overhead*

Le module de chiffrement est le composant touché par cette tactique. Il est conçu spécialement pour diminuer le plus possible les intermédiaires qu’il utilise afin d’accomplir sa tâche de chiffrement et de déchiffrement.

*Increase resource efficiency*

Le module de chiffrement est le seul composant touché par cette tactique. Il est conçu spécialement afin de limiter sa complexité et améliorer sa performance, sans toutefois compromettre la sécurité.

Relation entre les tactiques et les scénarios

Scénario 1

La tactique de priorisation des évènements permet de répondre au scénario de qualité 1 en s’assurant que les transactions les plus critiques telles que les retraits et les demandes d’authentification soient traitées en premier lors des périodes de charge. La tactique de maintenance de copies multiples des unités de traitement, quant à elle, répond au scénario 1 en s’assurant que la charge causée par un grand nombre de transactions soit répartie sur plusieurs serveurs bancaires. Ces deux tactiques visent à réduire la latence au niveau de la transaction.

Scénario 2

La tactique de réduction des coûts généraux (ou ​*overhead​*)répond au scénario 2 en diminuant le nombre d’intermédiaires du module de chiffrement et ainsi améliorer son efficacité. La tactique d’augmentation de l’efficacité d’une ressource répond au scénario 2 puisqu’il augmente l’efficacité du module de chiffrement en diminuant sa complexité. Ces deux tactiques visent à réduire la latence au niveau de la lecture et le chiffrement des informations bancaires des clients avant leur envoi aux serveurs bancaires pour authentification. Les tactiques de priorisation des évènements et de maintenance de copies multiples des unités de traitement répondent au scénario 2 puisqu’il visent à réduire la latence causée par la communication réseau avec les serveurs bancaires et leur traitement des transactions.

LOG430 ­ Architecture logiciel

**Performance**

Justification:

* Plus de transactions possibles
* Fidéliser le client
* Satisfaction du client
* Transactions concurrentes (nous)
* Latence réseau
* ­Performance serveur

Scénario 1

# Validation des transactions avec le serveur

| Scénario | Validation des transactions avec le serveur |
| --- | --- |
|  |  |
| Objectifs d’affaires | Fidéliser le client en s’assurant que les transactions sont |
|  | validées en moins de 5 secondes |
|  |  |
| Attribut de qualité | Performance |
|  |  |
| Source | Utilisateur du guichet |
|  |  |
| Stimulus | Début d’une transaction |
|  |  |
| Artéfact | Serveur |
|  |  |
| Environnement | Opération normale |
|  |  |
| Réponse | La transaction a été validée. |
|  |  |
| Mesure de la réponse | Latence pour le traitement |
|  |  |
| Questions | ● Quel matériel utiliser (CPU, RAM, etc.)? |
|  | ● Redondance? |
|  |  |

Scénario 2

# Communication avec le serveur

| Scénario | Communication avec le serveur | |
| --- | --- | --- |
|  |  | |
| Objectifs d’affaires | Fidéliser le client en s’assurant que les transactions sont valides | |
|  | en moins de 5 secondes | |
|  |  | |
| Attribut de qualité | Performance | |
|  |  | |
| Source | Utilisateur du guichet | |
|  |  | |
| Stimulus | Début d’une transaction | |
|  |  | |
| Artéfact | Réseau | |
|  |  | |
| Environnement | Opération normale | |
|  |  | |
| Réponse | La transaction a été effectuée. | |
|  |  | |
| Mesure de la réponse | Latence pour la communication | |
|  |  | |
| Questions | ● Quel matériel utiliser? (Fibre optique, câble coaxial, | |
|  |  | torsadé, téléphonique etc.) |
|  | ● | Quelles données envoyer |
|  | ● | Doit­ on compresser? |
|  | ● |  |
|  |  |  |

Tactique 1

**Description**:​Increase resource efficiency

**Justification**:​En améliorant les algorithmes et la gestion des ressources et des données, ilest possible de limiter la quantité d’information a traiter ce qui réduit le temps global d’analyse des requêtes.

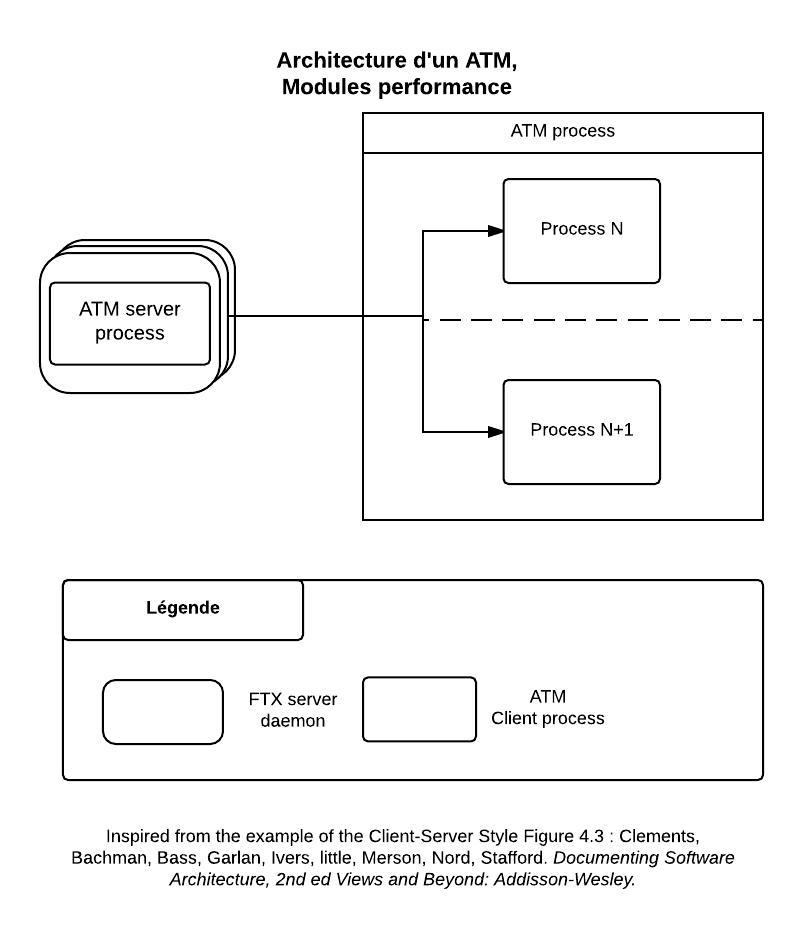
Tactique 2

**Description**:​Introduce concurrency

**Justification**:​Au lieu de traiter les requêtes des clients de façon bloquante en les traitantpremier arriver premier servi, ajouter la gestion de concurrence permet de traiter de multiples requêtes simultanément et ainsi réduire la latence entre l’envoi d’une requête et le traitement de celle-­ci.

Vue architecturale

Diagramme et légende



Description du diagramme

Le ATM server process représente le système global de traitement de requête, il traite en interne les requêtes faites par les différents ATM, les ATM process représentent les différents threads qui exécutent en parallèle les requêtes.

Description des éléments du diagramme

| ATM Server Process | Système de traitement de requête |
| --- | --- |
|  |  |
| ATM Process | Ensemble des différents threads |
|  |  |
| Process N/N+1 | Processus |
|  |  |

Description des relations

Les ATM main process représentent l’implémentation de la tactique de concurrence. Le Server process gère la tactique des ressources efficace.

Tactiques répond aux scénarios

Pour réduire le temps de traitement de chaque requête, une tactique d’usage efficace des ressources nous aide à résoudre notre problème en s’assurant que le nombre de données que doit traiter le serveur est limité au strict nécessaire et organisé de façon efficace pour augmenter la vitesse générale de validation des requêtes.

De plus, implémenter une gestion de concurrence nous permet d’éviter d’avoir à traiter les possibles milliers de requêtes de façon séquentielle et de plutôt les traiter en parallèle. Ceci nous permet de réduire au maximum le temps écoulé avant que la réponse soit envoyée.

**LOG430 – Architecture logicielle**

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

**Performance – Justification**

La performance du système sert à éviter de créer des files devant les guichets et améliorer l’expérience usager.

**Performance – Scénario 1**

# Accéder à la liste des comptes d’un utilisateur

| **Scénario** | Accéder à la liste des comptes d’un utilisateur | |
| --- | --- | --- |
| **Objectifs d'affaires** | Le temps d’attente lors d’une transaction doit être réduit au maximum pour ne pas créer de bouchons | |
|  |
| **Attribut de qulité** | Performance | |
| **Source** | Le guichet | |
| **Stimulus** | Lance une requête de la liste des comptes d’un usager | |
| **Artéfact** | Centrale des serveurs bancaires | |
| **Environnement** | Environnement de production en opération normale | |
| **Réponse** | La liste des comptes de l’usager | |
| **Mesure de la** | Le temps d’attente entre la l’envoi de la requête et la réception de la réponse. | |
| **réponse** |  |  |
| **Questions** | 1. | Quel est le temps d’attente acceptable maximal? |
|  | 2. | Combien de requêtes peuvent être effectuées par minutes? |
|  | 3. | Est-ce que les tables des comptes devraient être accessibles sur plusieurs serveurs pour réduire la quantité de requêtes sur un seul serveur? |
|  |  |
|  | 4. | Est-ce que plusieurs centrales seraient nécessaires au travers du pays pour réduire la distance avec les guichets? |
|  |  |

**Performance – Scénario 2**

# Servir plusieurs ATM en parallèle

| **Scénario** | Servir plusieurs ATM en parallèle |
| --- | --- |
| **Objectifs d'affaires** | Permettre à plusieurs ATM de se connecter à un même serveur simultanément. |
| **Attribut de qualité** | Performance |
| **Source** | Le guichet |
| **Stimulus** | Nouvelle transaction |
| **Artéfact** | Serveur |
|  | Le guichet |
|  | Log de la transaction |
| **Environnement** | Opération normale |
| **Réponse** | Transaction effectuer |
| **Mesure de la** | Le temps d’attente entre l’envoi de la requête et la réception de la réponse. |
| **réponse** |  |
| **Questions** | 1. Combien d’ATM vont se connecter en même temps à un serveur? |
|  | 2. Quelles opérations nécessitent l’intervention du serveur? |
|  | 3. Quelles données sont susceptibles d’être modifiées en parallèle? |

**Performance – Tactique 1**

**Description**: Maintain multiple copies of data

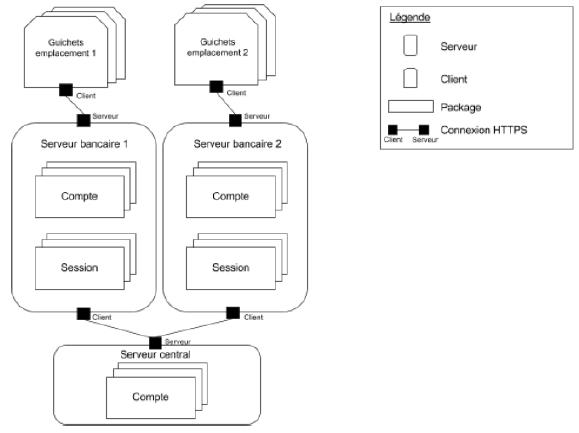
**Justification**: La tactique utilisée dans le cadre du scénario 1 est *Maintain multiple copies of data*, car plusieurs requêtes d’obtention de la liste des comptes pourraient survenir et causer des délais si la donnée n’est enregistrée que sur un serveur. C’est pourquoi il faudrait dupliquer les bases de données des comptes sur plusieurs serveurs au travers du pays, réduisant du fait même la latence.

**Performance – Tactique 2**

**Description**: La tactique employée pour le scénario 2 est *Introduce concurrency*, de la catégorie *Manage Resources*. Pour supporter cette tactique, le serveur devra être capable de gérer plusieurs sessions simultanément. Plusieurs ATM pourront donc se connecter à un même serveur, permettant une meilleure utilisant des ressources disponibles.

**Justification**: La plupart des serveurs supportent la concurrence par défaut. Il est important de s’assurer que nos serveurs la supporte et qu’ils le font de manière appropriée. Permettre à plusieurs ATM de se connecter en parallèle à un serveur nous permettra de mieux utiliser les ressources des machines.

**Performance – Vue architecturale (à l’extérieur de la classe)**



**Description du diagramme**

Pour créer la vue architecturale, nous avons opté pour le style client-serveur. Dans le diagramme, plusieurs éléments sont schématisés et, en les analysant, on peut voir que ce sont des guichets à des emplacements différents qui communiquent à l’aide du protocole HTTPS avec des serveurs bancaires différents.

En lien avec la tactique 1, on peut voir que les comptes sont disponibles sur tous les serveurs bancaires, ce qui réduit le délai d’attente.

En ce qui a trait à la tactique 2 , on peut voir que des sessions sont ouvertes sur un seul serveur pour une panoplie de guichets qui se trouve à l’emplacement 1.

**Table de description des éléments du diagramme**

Voici un tableau qui décrit les caractéristiques des éléments des vues architecturales présentées à la page précédente.

| **Élément** | **Description** |
| --- | --- |
|  |  |
| Guichet emplacement | Représente un ensemble de guichets présent à un même emplacement physique. Chaque guichet sert d’interface usager physique. Ils envoient diverses données sur les transactions et les statuts des périphériques. |
|  |
|  |
|  |  |
| Serveur bancaire | Reçois les données de plusieurs guichets et les traites. Contiens une copie des comptes clients susceptible de se connecter via les guichets connectés à ce serveur. |
|  |
|  |
|  |  |
| Serveur central | Gère l’information des comptes clients |
|  |  |
| Compte | Représente les données associées à un compte utilisateur |
|  |  |
| Session | Représente les données associées à une session |
|  |  |
| Connexions client-serveur | Connecte via HTTPS les guichets aux serveurs de la banque ainsi que le |
|  | serveur central. |
|  |  |

LOG430 – Architecture logicielle

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

Performance – Justification (en classe)

Pour éviter des mécontentement du client a cause d’une trop longue attente.

La performance du serveur est important pour assuré la gestion des transaction de masse.

Performance – Scénario 1 (en classe)

# Latence de la communication avec le serveur dû l’encryptage des messages trop lourd

| **Scénario** | Latence de la communication avec le serveur dû l’encryptage des messages trop lourd | |
| --- | --- | --- |
|  |
| **objectifs d'affaires** | Satisfaction du client | |
| **Attribut de qualité** | Performance | |
| **Source** | Interne communication entre l’ATM et le serveur | |
| **Stimulus** | Périodique | |
| **Artéfact** | Le message encrypté | |
| **Environnement** | Normal |  |
| **Réponse** | Message au client : veuillez patienter! | |
|  | Log d’un temps anormale pour la lecture d’un message client. | |
| **Mesure de la** | Temps de latence, deadline, débit | |
| **réponse** |  |  |
| **Questions** | 1. | C’est quoi le type d’encryption utiliseé? |
|  | 2. | Quel doit être le débit acceptable ? |
|  | 3. |  |
|  | 4. |  |
|  | 5. |  |
|  |  |  |

Performance – Scénario 2 (en classe)

# Surcharge de transactions sur le serveur

| **Scénario** | Surcharge de transactions sur le serveur | |
| --- | --- | --- |
| **objectifs d'affaires** | Satisfaction du client | |
| **Attribut de qualité** | Performance | |
| **Source** | Interne entre l’ATM et le serveur | |
| **Stimulus** | Sporadique | |
| **Artéfact** | Serveur |  |
| **Environnement** | Overload | |
| **Réponse** | Changement de mode du serveur a overload. | |
| **Mesure de la** | Temps de latence et réponse, deadline. | |
| **réponse** |  |  |
| **Questions** | 1. | C’est quoi la capacité de traitement du serveur? |
|  | 2. |  |
|  | 3. |  |

Performance – Tactique 1 (en classe)

**Description**: Control​ Ressource Demand ­> Prioritize events

**Justification**:Il est important que les messages encryptés soient traités rapidement par le serveur. Pour ce faire la priorisation d’évènements permettrait de prioriser les messages plus lourds et qui demandent plus de ressources à être traité dans un temps acceptable sans trop ralentir le système.

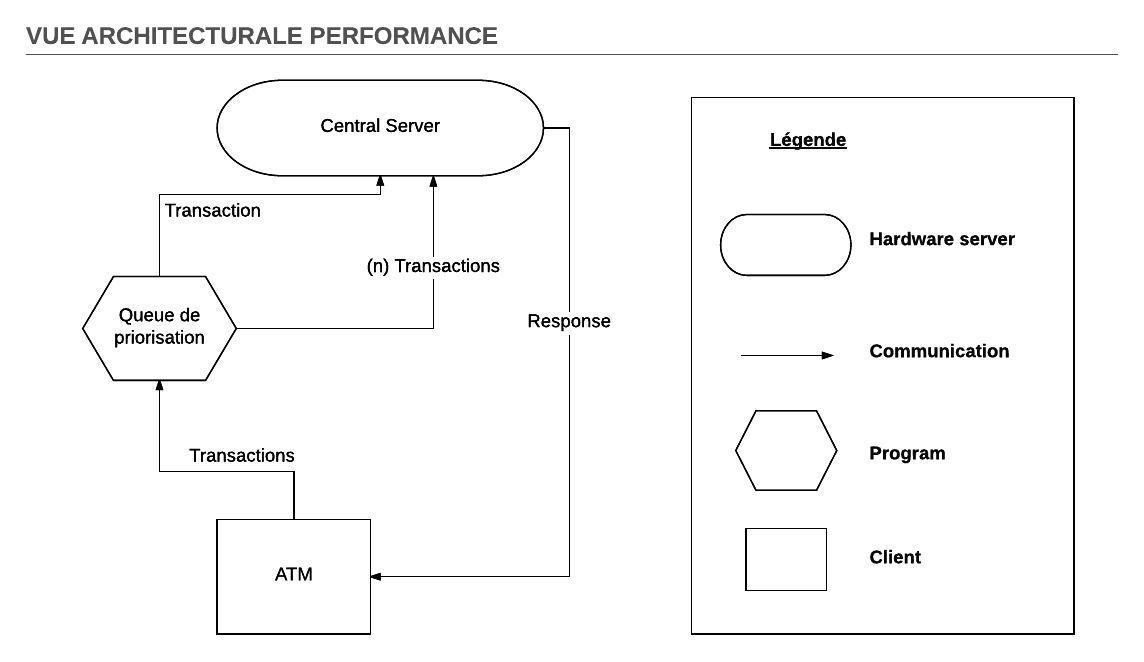
Performance – Tactique 2 (en classe)

**Description**: Manage​ Resources ­> Introduce concurrency

**Justification**: Il est important d’être en mesure de traiter toutes les transactions reçues dans un temps acceptable. L’utilisation de la concurrence accélérait le traitement des transactions en traitant en parallèle plusieurs transactions.

Performance – Vue Architecturale (à l’extérieur de la classe)

L’architecture proposée permettra à l’ATM de se connecter au serveur central. La connexion entre l’ATM et le serveur central est faite par une queue de priorisation (passerelle). Celle-­ci permettra de faire le lien entre les transactions reçues de l’ATM et les serveurs. De cette façon, il sera possible de traiter plusieurs transactions simultanément sur plusieurs serveurs.



**Tableau des éléments**

| Éléments | Description |
| --- | --- |
|  |  |
| Central servers | Plusieurs serveurs physiques permettant de la redondance. |
|  |
|  |  |
| Queue de priorisation (Passerelle) | Dirige la transaction de façon transparente |
|  | entre l’ATM et un serveur central disponible. |
|  |  |
| ATM | Le guichet automatique. |
|  |  |
| Transactions (ATM vers Passerelle) | Flux de transactions qui entre vers les serveurs de façon transparente pour l’ATM |
|  |
|  |  |
| Transactions (Passerelle vers Serveurs) | Transaction dirigée vers le bon serveur |
|  |  |
| Response | Réponse du serveur pour l’ATM |
|  |  |

**Description de la relation entre les éléments et les tactiques**

L’élément “Queue de priorisation” a pour responsabilité de répartir les transactions dépendamment de la priorisation du message ainsi que de l’état des serveurs pour répondre au besoin de la tactique 1.

L’élément “Central server” représente plusieurs serveurs qui reçoivent des messages précédemment priorisés. De cette façon, l’architecture permet de répondre au besoin de la tactique 2.

LOG430 – Architecture logicielle

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

**Performance** – Justification (en classe)

L’utilisateur doit pouvoir accomplir les tâches qu’il désire dans le plus petit laps de temps possible, afin de pouvoir faire autre chose de sa journée. La banque à qui appartient le GAB veut accommoder le plus grand nombre de clients possibles, afin de maximiser les frais de transaction.

**Performance – Scénario 1 (en classe)**

# Temps de calcul pour le serveur trop long pour une opération

| **Scénario** | Temps de calcul pour le serveur trop long pour une opération |
| --- | --- |
| **objectifs d'affaires** | Réduire au minimum le temps de calcul nécessaire au guichet |
| **Attribut de qualité** | Performance |
| **Source** | Utilisateurs |
| **Stimulus** | L’utilisateur confirme une grosse opération qu’il souhaite faire effectuer. |
| **Artéfact** | Système |
| **Environnement** | Opération normale |
| **Réponse** | * Le système accomplit les tâches demandées en priorité et retarde l’exécution des processus non essentiels. * Le système affiche un message et une animation à l’utilisateur pour le faire patienter et pour lui indiquer que le guichet n’est pas gelé. * Le guichet démarre son application d’auto-diagnostic pour voir si une des pièces électroniques du guichet a un bris mécanique. |
| **Mesure de la réponse** | Temps d’exécution de moins de 30 secondes |
| **Questions** | 1. Est-ce que la mesure de la réponse est trop longue? 2. Quel genre d’opération peut prendre beaucoup de temps? 3. Quels facteurs peut-on contrôler? |

# 

**Performance – Scénario 2 (en classe)**

# Latence entre la machine et la réponse du serveur

| **Scénario** | Latence entre la machine et la réponse du serveur |
| --- | --- |
| **objectifs d'affaires** | Réduire le temps d’attente du client pour la réponse du serveur |
| **Attribut de qualité** | Performance |
| **Source** | Utilisateur |
| **Stimulus** | L’utilisateur veut faire une opération qui demande une confirmation ou une information du serveur |
| **Artéfact** | Système de communication du guichet |
| **Environnement** | Opération normale |
| **Réponse** | * Le serveur entame les opérations qu’on lui demande, si besoin est. * Le serveur redirige les demandes trop lourdes vers d’autres serveurs moins occupés. * Le serveur démarre une application d’auto-diagnostic pour la carte réseau et les autres systèmes de communication entre le serveur et le guichet. |
| **Mesure de la réponse** | Temps entre l’envoi de la demande et la réception de la réponse du serveur de moins de 30 secondes |
| **Questions** | 1. Qu’est-ce qui peut causer de la latence entre le serveur et le guichet? 2. Pour quels facteurs peut-on faire quelque chose?   2.  3.  4.  5. |

**Performance – Tactique 1 (Prioritize Events)**

**Description**: Prioriser les événements et ignorer ceux qui sont moins importants, en faveur de ceux avec une plus haute priorité.

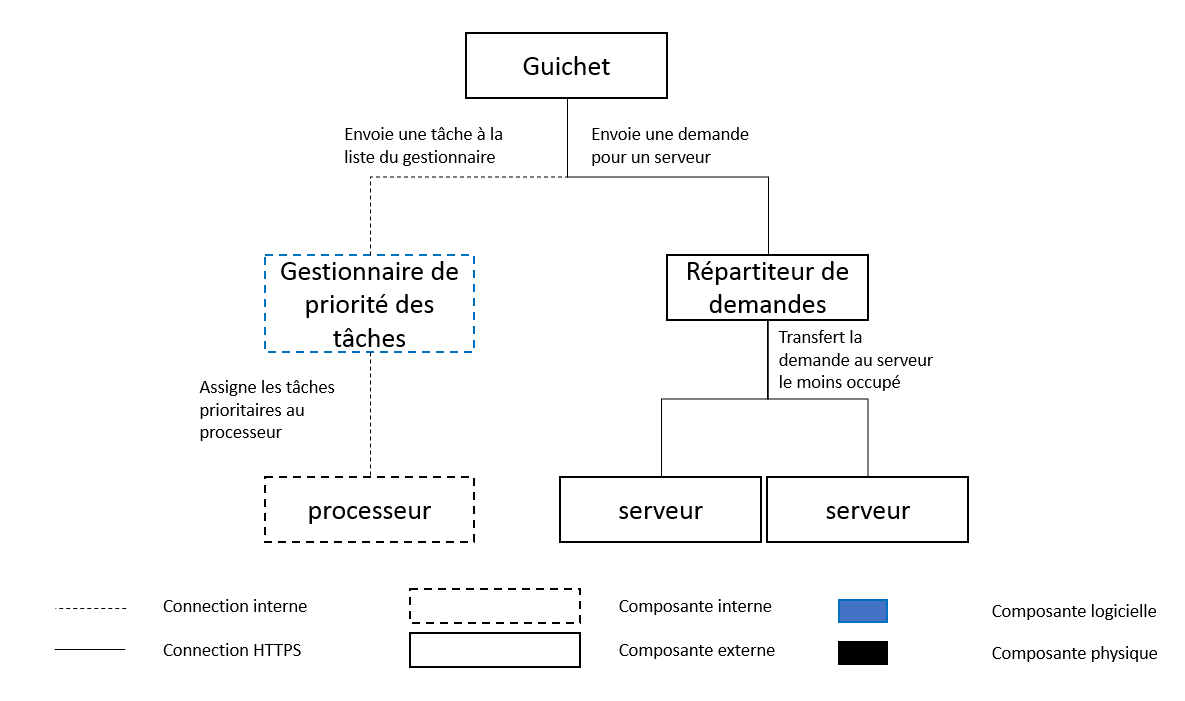
**Justification**: En ignorant certains des processus non-essentiels du système, on peut accélérer l’accomplissement des tâches qu’on demande au guichet d’accomplir.

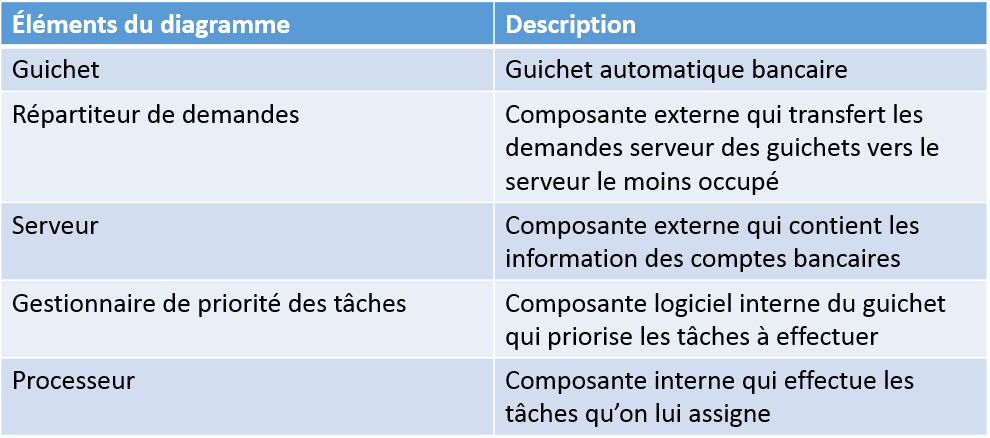
**Performance – Tactique 2 (Maintain Multiple Copies of Computations)**

**Description**: Répliquer les calculs à faire, en séparant les processus parmi les ressources disponibles.

**Justification**:En distribuant les demandes parmi les serveurs disponibles, le temps de latence peut être réduit.

**Performance – Vue Architecturale (à l’extérieur de la classe)**





Dans le cas d’une opération à l’interne, le guichet doit optimiser son rendement. Pour ce faire, il passe toute les tâches à effectuer à une composante logicielle appelée gestionnaire de tâches. Celui-ci affecte des priorités aux tâches selon les règles d’affaire qui lui ont été spécifiées. Les tâches à faire effectuer en priorité sont assignées en premier au processeur. Celui-ci, composante interne du guichet, exécute les tâches qui lui sont demandées et envoie le résultat au système qui le requiert.

Pour les opérations qui demande un appel à un serveur externe, la demande passera par un répartiteur de demandes. Ce dernier reçoit les demandes d’appel au serveur des guichets et les redistribue verss les moins occupés. Cette méthode permet de réduire le temps moyen d’attente de réponse des guichets, et donc le temps de latence.