POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG8430 : Attributs de qualité, partie 3

Attributs de qualité – partie 3

Définition des attributs de qualité

1. Disponibilité
2. Interopérabilité
3. Modifiabilité
4. **Performance**
5. **Sécurité**
6. Testabilité
7. Utilisabilité
8. Autres attributs de qualité

Attribut de qualité 4: Performance

La performance d'un système réfère à sa capacité à fournir des résultats à l'intérieur de délais prescrits.

Lorsqu'un évènement se produit – interruption, message, requête de l'usager ou d'un autre système, évènement d'horloge marquant le passage du temps – le système ou l'un de ses éléments doit y répondre à temps.

Caractériser les évènements qui peuvent se produire et la réponse du système ou de certains de ses éléments en terme de temps est l'essence d'une discussion sur la performance.

Attribut de qualité 4: Performance

Différents systèmes, différents types d'évènements, différentes contraintes de performance:

- Sur un système web, les évènements arrivent sous la forme de requêtes des utilisateurs via leurs fureteurs,
- Sur un moteur à combustion interne, les évènements sont liés aux actions de l'opérateur et au passage du temps. Le système doit contrôler l'allumage et l'admission du mélange des gaz pour maximiser la puissance et minimiser la pollution.

Et différentes réponses:

- La performance d'un site web se mesure en nombre de transactions traitées par minute,
- La performance du moteur à explosion se mesure dans la variation permise sur le temps d'allumage et la synchronisation du cycle de combustion.

Attribut de qualité 4: Performance

Très longtemps, la performance a été l'élément moteur guidant les architectes logiciels, ce qui a fréquemment compromis tous les autres attributs de qualité.

Le rapport prix/performance du matériel poursuit sa chute et les coûts de développement continuent à augmenter, faisant émerger d'autres attributs de qualité comme des compétiteurs sérieux à la performance.

Tous les systèmes ont des requis de performance et la performance reste donc un attribut de qualité très important pour tout logiciel.

Attribut de qualité 4: Performance

La performance est souvent liée à l'évolutivité, particulièrement pour l'augmentation de la capacité de traitement.

Techniquement, l'évolutivité est une forme de changement. Elle est donc liée à la modifiabilité du système.

Scénario général de performance

Un scénario de performance débute par l'arrivée d'un évènement sur le système.

Répondre correctement à l'évènement requiert des ressources, incluant du temps.

Pendant que le système répond à l'évènement, il peut aussi devoir traiter d'autres évènements simultanément.

Scénario général de performance

Différents schémas d'arrivée des évènements sur un système:

- Évènements périodiques: les évènements arrivent à intervalles réguliers. Typique des systèmes temps-réels.
- Stochastiques: les évènements arrivent selon une distribution de probabilité.
- Sporadiques: les évènements arrivent selon un schéma ni périodique ni stochastique. Peuvent quand même être caractérisés, p. ex. nombre maximum d'évènements par intervalle de temps, délais minimum entre les évènements.

Scénario général de performance

Différentes mesures de la réponse du système:

- Latence: Le délais entre l'arrivée d'un évènement et la réponse du système.
- Temps limite: Contrainte dure sur le temps maximal avant lequel une réponse doit être produite pour être valide.
- Débit: Généralement mesuré par le nombre de transactions que le système peut traiter dans un laps de temps donné.
- Gigue: Variation permise dans la latence.
- Nombre d'évènements non traités: taux de perte des requêtes qui arrive sur le système. Généralement parce que le système est surchargé.

Scénario général de performance

Source du stimulus

Interne ou externe au système.

Stimulus

Arrivée d'un évènement périodique, sporadique ou stochastique.

Artéfact

Le système ou une ou plusieurs composantes du système

Environnement

Le mode d'opération du système: normal, en urgence, en pointe de charge ou en surcharge.

Réponse

Traiter les évènements, changer le mode d'opération.

Mesure de la réponse

Latence, temps limite, débit, gigue, taux de rejet.

Exemple de scénario concret de performance

Source du stimulus

Utilisateurs

Stimulus
Initiation d'une transaction

Artéfact
Système

Environnement
Mode d'opération normal

Réponse
Les transactions sont traitées

Mesure de la réponse
Latence moyenne de 2 secondes.

Attribut de qualité 4: Performance Concurrence

La concurrence est la possibilité/capacité d'exécuter plusieurs tâches simultanément – en parallèle.

Liée à la création de fils d'exécution distincts:

→ Les fils d'exécution sont des séquences de contrôle indépendantes,

→ À la base des systèmes multi-tâches et multi-usagers,

→ Permettent d'exploiter les architectures matérielles multiprocesseurs et multicœurs.

Plusieurs types d'infrastructures logiciels reposent sur l'utilisation de la concurrence:

- Architecture maître-esclaves (ex. map-reduce),
- Bases de données NoSQL, etc.

Attribut de qualité 4: Performance Concurrence

La concurrence permet d'améliorer la performance d'un système en lui permettant d'exécuter plusieurs tâches en parallèle:

- Permet d'exploiter les périodes de mise en attente d'un processus pour en faire progresser un autre,
- Les situations potentielles de compétition (*race conditions*) doivent être analysées et gérées correctement: gestion d'un état partagé,
- Utilisation de verrous ou partage/distribution de l'état entre les processus.

Attribut de qualité 4: Tactiques pour la performance

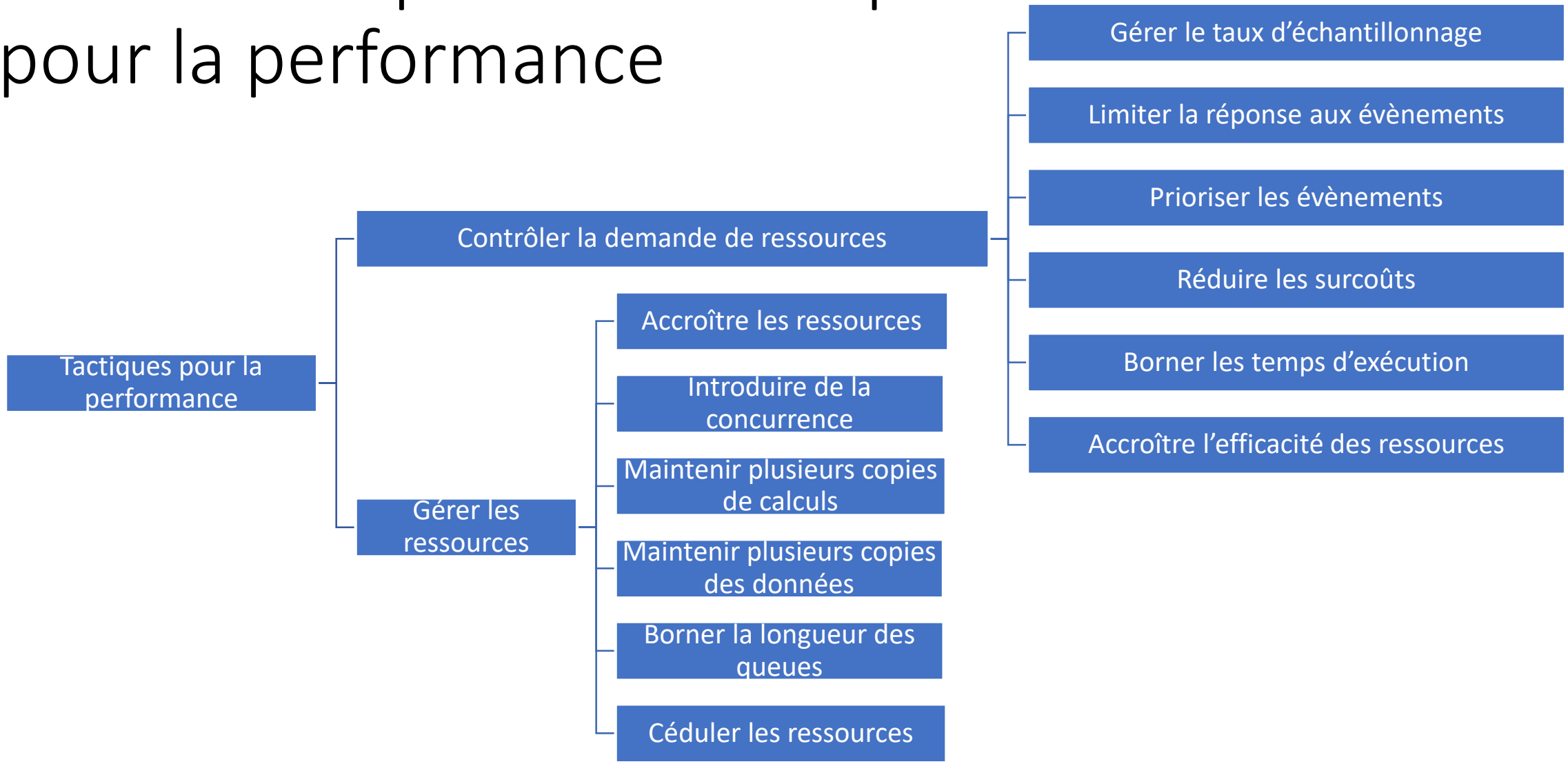
Un système est performant s'il est en mesure de traiter les évènements qui lui arrivent dans les délais spécifiés.

Attribut de qualité 4: Tactiques pour la performance

Entre le moment où un évènement arrive sur le système et le moment où la réponse est complétée, le système peut être soit:

1. En train de s'exécuter: le temps pendant lequel des ressources sont consommées pour répondre à l'évènement. Ressources matérielles: mémoire, processeur, bande passante, entrepôt de données. Ressources logicielles: composantes et entités du système, queues, tampons, sections critiques.
2. Bloqué en attente: le temps passé à attendre que des ressources deviennent disponibles:
   - Contention sur les ressources, p. ex. ressources qui ne peuvent servir qu'un seul client à la fois.
   - Indisponibilité de ressources, p. ex. en cas de faute d'une composante.
   - Dépendance à d'autres calculs, p. ex. nécessité de synchroniser des résultats avant de continuer.

Attribut de qualité 4: Tactiques pour la performance



Attribut de qualité 4: Tactiques pour la performance: **Contrôler la demande de ressources**

Gérer le taux d'échantillonnage

- Réduire la fréquence d'échantillonnage à laquelle un flux de données est capturé
  - Gérer le niveau de fidélité de la réponse.
  - P.ex. différents codecs avec différents taux d'échantillonnage et différents formats de données.
  - Décider si un taux d'échantillonnage plus bas, donc une fidélité plus basse, est avantageux pour assurer plus de stabilité et diminuer la latence ou le taux de perte de paquets.

Attribut de qualité 4: Tactiques pour la performance: **Contrôler la demande de ressources**

Limiter la réponse aux évènements

- Mise en queue des évènements discrets qui arrivent sur le système
  - →Si les évènements sont discrets, non désirable de réduire le taux d'échantillonnage,
  - →Si les évènements s'accumulent trop dans la queue, le temps de traitement devient moins prédictible,
- Décider s'il est acceptable de laisser tomber des évènements:
  - →Si non, prévoir des queues suffisamment grande pour les pires scénarios,
  - →Si oui, quel est la politique choisie: quels évènements ne sont pas traités? Sont-ils enregistrés ? Faut-il avertir quelqu'un: usager, administrateur, un autre système ?

Attribut de qualité 4: Tactiques pour la performance: **Contrôler la demande de ressources**

Prioriser les évènements

- Si tous les évènements n'ont pas la même importance, il est possible d'établir des priorités:
  - Définir des niveaux de priorité,
  - Quels évènements doivent absolument être traités,
  - Quels évènements peuvent être ignorés.

Attribut de qualité 4: Tactiques pour la performance: **Contrôler la demande de ressources**

Réduire les surcoûts

- Si des intermédiaires ont été introduits pour accroître la modifiabilité du système, cela peut augmenter la latence,
- La séparation des responsabilités peut créer une longue chaîne de composantes impliquées dans le traitement d'un évènement,
- Réduire les intermédiaires et raccourcir les chaines de traitement améliore la latence.
- S'assurer que des composantes logicielles s'exécutent sur un même nœud de calcul réduit les coûts de communication.
- Intégrer des composantes dans un même exécutable réduit les coûts de communication inter-processus.
- Certaines ressources doivent être nettoyées ou réinitialisées périodiquement pour améliorer leur performance (tables de hachage, pagination virtuelle).

Attribut de qualité 4: Tactiques pour la performance: **Contrôler la demande de ressources**

Borner les temps d'exécution

- Mettre une limite sur le temps de calcul qui peut être utilisé pour traiter un évènement.
- Limiter le nombre d'itérations dans un algorithme itératif.
- Limiter la précision des résultats. Déterminer quel niveau de fidélité est nécessaire ou acceptable.

Attribut de qualité 4: Tactiques pour la performance: **Contrôler la demande de ressources**

Accroître l'efficacité d'utilisation des ressources

- Améliorer le algorithmes et optimiser les structures de données.
- Connaitre et utiliser les options d'optimisation du code et de gestion des ressources.
- Identifier les sections critiques:
  - Instrumenter le code,
  - Analyser les traces d'exécution,
  - Réimplémenter certaines fonctions dans un autre langage.
- Utiliser les versions les plus récentes des bibliothèques, composantes et services.

Attribut de qualité 4: Tactiques pour la performance: **Gérer les ressources**

Accroître les ressources

- Plus de processeurs, des processeurs plus rapides, plus de mémoire, un réseau plus rapide peuvent tous réduire la latence.
- Équilibre entre coût et performance.
- Souvent la façon la plus rapide et la moins coûteuse d'accroître la performance.

Attribut de qualité 4: Tactiques pour la performance: **Gérer les ressources**

Introduire de la concurrence

- Le traitement en parallèle des requêtes permet de réduire la latence en réduisant le temps de blocage.
- Des flux d'évènements distincts peuvent être traités par des processus distincts ou des fils séparés peuvent être utilisés pour différents types d'activités.
- En présence de concurrence, différentes politiques d'ordonnancement peuvent être établies: maximiser l'équité, maximiser le débit, etc.

Attribut de qualité 4: Tactiques pour la performance: **Gérer les ressources**

Maintenir plusieurs copies de calculs

- Plusieurs serveurs dans une architecture client-serveur sont des serveurs répliqués ou des répliques.
- Le but des répliques est de réduire les contentions qui peuvent se produire sur un serveur unique.
- Un équilibreur de charge est un algorithme qui permet de répartir les tâches entre les différents serveurs. Différents algorithmes possibles:
  - Round-robin : répartition séquentielle à tour de rôle.
  - Assignation selon la charge courante des serveurs.

Attribut de qualité 4: Tactiques pour la performance: **Gérer les ressources**

Maintenir plusieurs copies des données

- Les caches: permettent de conserver des copies multiples (souvent partielles) de données sur des dispositifs ayant différentes vitesse d'accès.
- La réplication de données: implique de garder plusieurs copies des données pour réduire les contentions d'accès dues, p.ex., au réseau:
- Conserver des copies cohérentes et synchronisées devient une responsabilité du système.
- Choisir quelles données sont répliquées ou mise en cache est aussi une responsabilité du système:
  - Conserver ce qui a été demandé récemment
  - Tenter de prévoir ce qui va être demandé prochainement

Attribut de qualité 4: Tactiques pour la performance: **Gérer les ressources**

Borner la longueur des queues

- Mécanisme de contrôle du nombre maximum d'évènements qui peuvent arriver sur le système.
- Nécessaire d'établir une politique concernant la réaction du système en cas de débordement. Est-il acceptable d'ignorer certains évènements ?

Attribut de qualité 4: Tactiques pour la performance: **Gérer les ressources**

Céduler les ressources

- Dès qu'il y a contention pour une ressource, il faut introduire un mécanisme de planification.
- Les processeurs, les tampons, les réseaux sont contrôlés par des mécanismes de planification.
- Analyser les caractéristiques de la ressource pour choisir la politique appropriée.

Attribut de qualité 4: Tactiques pour la performance: Politiques de planification

Une politique de planification (*scheduling*) a 2 parties:

1. L'assignation de priorités,
2. L'assignation (*dispatching*).

Toute planification doit établir des priorités, simples ou complexes.

Plusieurs critères concurrents doivent être considérés pour établir une politique de planification: utilisation optimale des ressources, importance des requêtes, minimisation de la latence, maximisation du débit, éviter l'épuisement, assurer l'équiter, etc.

Attribut de qualité 4: Tactiques pour la performance: Politiques de planification

Une politique de planification a 2 parties:

1. L'assignation de priorités,
2. L'assignation (*dispatching*).

Pour permettre d'assigner un flux d'évènement à haute priorité, il peut être nécessaire de préempter une ressource.

Plusieurs options possible pour effectuer la préemption:

- À tout moment,
- À certains points d'exécution spécifiques,
- La préemption peut être interdite par certains processus.

Attribut de qualité 4: Tactiques pour la performance: Politiques de planification

Quelques politiques de planification courantes:

- Premier arrivé/premier servi (FIFO),
- Planification à priorité fixe,
- Importance sémantique,
- Échéances monotones,
- Débit monotone,
- Priorité dynamique,
- Round-robin,
- Échéance la plus proche d'abord,
- Marge minimum d'abord.

Liste de vérification pour la performance
Allocation des responsabilités

- Déterminer quelles responsabilités du système seront associées à une charge importante, auront des requis critiques de réponse en temps, seront utilisées intensivement ou impact une portion du système où une charge importante ou une réponse critique se produit.
- Pour chacune de ces responsabilités, identifier les requis de traitement pour chaque responsabilité et déterminer si elles peuvent engendrer des goulots d'étranglement.
- Pour les responsabilités et ressources identifiées, s'assurer que les requis de réponse en performance peuvent être rencontrés. P.ex. à l'aide d'un modèle de performance.

Liste de vérification pour la performance
Allocation des responsabilités

- Identifier des responsabilités additionnelles pour reconnaitre et traiter les requêtes de façon efficace:
  - Des responsabilités résultant d'un fil de contrôle qui gère plusieurs processus ou des processus sur plusieurs processeurs,
  - Des responsabilités pour gérer les fils d'exécution, allocation, désallocation, pool de fils,
  - Des responsabilités pour gérer des ressources partagées ou des artéfacts liés à la performance, comme des queues, des tampons et des caches.

Liste de vérification pour la performance
Modèle de coordination

- Déterminer les éléments du système qui doivent se coordonner entre eux, directement ou non, et choisir des mécanismes de coordination et de communication afin de:
  - Supporter au besoin, la concurrence, la priorisation d'évènements ou des stratégies de planification,
  - S'assurer que la réponse en performance peut être fournie,
  - Capturer les arrivées d'évènements périodiques, sporadiques et stochastiques,
  - Fournir les propriétés appropriées pour les mécanismes de communication: avec ou sans état, synchrone ou asynchrone, livraison garantie, débit, latence.

Liste de vérification pour la performance
Modèle de données

- Déterminer les portions du modèle de données qui sont fortement chargées, nécessitent une réponse critique en temps, sont très utilisées ou ont un impact sur des portions du système où une forte charge ou des évènements critiques en temps se produisent.
- Pour ces abstractions de données, déterminer:
  - Si maintenir plusieurs copies de données clés améliorerait la performance,
  - Si partitionner les données améliorerait la performance,
  - S'il est possible de réduire les requis de traitement pour la création, l'initialisation, la persistance, la manipulation, la traduction ou la destruction de ces données,
  - S'il est possible d'ajouter des ressources pour réduire les goulots d'étranglement pour la création, l'initialisation, la persistance, la manipulation, la traduction ou la destruction de ces données.

Liste de vérification pour la performance
Correspondance entre les éléments architecturaux

- Là où une forte charge réseau se produit, déterminer si de placer sur un même nœud de calcul certaines composantes peut réduire la charge et améliorer l'efficacité globale.
- S'assurer que les composantes avec les requis de charge les plus importants soient assignés aux processeurs avec la plus grande capacité de traitement.
- Déterminer les endroits où introduire de la concurrence est faisable et a un impact positif significatif sur la performance.
- Déterminer l'introduction de fils de contrôle et les responsabilités associées introduisent des goulots d'étranglement.

Liste de vérification pour la performance
Gestion des ressources

- Déterminer quelles ressources du système sont critiques pour la performance. Pour ces ressources, s'assurer qu'elles sont monitorées et gérées dans des conditions d'opération normales et en surcharge:
  - Les éléments du système qui doivent être au courant et gérer le temps et d'autres ressources critiques pour la performance,
  - Les modèles de processus et de fils d'exécution,
  - La priorisation des ressources et l'accès aux ressources,
  - Les stratégies de planification et de verrouillage,
  - Le déploiement de ressources additionnelles sur demande afin de répondre à l'augmentation de la charge.

Liste de vérification pour la performance
Choix du moment pour effectuer une liaison

- Pour chaque élément qui sera lié après la compilation, déterminer:
  - Le temps nécessaire pour effectuer la liaison,
  - Le surcoût supplémentaire introduit par l'utilisation d'un mécanisme de liaison tardif.
- S'assurer que ces valeurs n'imposent pas des pénalités de performance inacceptables.

Liste de vérification pour la performance
Choix de technologie

- Est-ce que vos choix de technologies vont vous permettre d'établir et de respecter des temps limites durs? Connaissez-vous leurs caractéristiques et leurs limites en cas de charge importante?
- Vos choix technologiques vous permettent-ils d'introduire:
  - Une politique de planification,
  - Des priorités,
  - Des politiques pour réduire la demande,
  - Une allocation de portions de technologies à des processeurs,
  - D'autres paramètres liés à la performance.
- Vos choix technologiques introduisent-ils des goulots d'étranglement inacceptables en cas de forte charge?

Attribut de qualité 5: Sécurité

La sécurité est la capacité d'un système à protéger les données et l'information contre les accès non-autorisés tout en fournissant l'accès aux personnes et systèmes autorisés.

Une action prise contre un système pour lui causer des dommages est appelée une attaque, et les attaques peuvent prendre plusieurs formes:

- Tentative non-autorisée d'accès à des données ou service,
- Tentative de modification de données,
- Tentative d'empêcher des utilisateurs légitimes d'avoir accès à un service.

Attribut de qualité 5: Sécurité

La façon la plus simple de caractériser la sécurité comprend trois caractéristiques (CID):

- Confidentialité: la propriété qu'ont des données ou services d'être protégés contre des accès non-autorisés.
- Intégrité: la propriété qu'ont des données ou services d'être protégés des contre des modifications non-autorisés.
- Disponibilité: la propriété qu'a un système d'être disponible pour les utilisateurs légitimes.

Attribut de qualité 5: Sécurité

Trois autres caractéristiques qui sont utilisées pour supporter les CID:

- Authentification: vérifie l'identité des parties intervenant dans une transaction et s'assure qu'elles sont bien qui elle prétendent être.
- Non-répudiation: garantit que l'envoyeur d'un message ne peut pas, plus tard, nier avoir envoyé le message et que le récepteur du message ne puisse pas nier l'avoir reçu.
- Autorisation: fournit à un utilisateur les privilèges nécessaires pour effectuer une tâche.

Les objets qui sont protégés contre les attaques sont les données au repos, les données en transit et les processus de calcul.

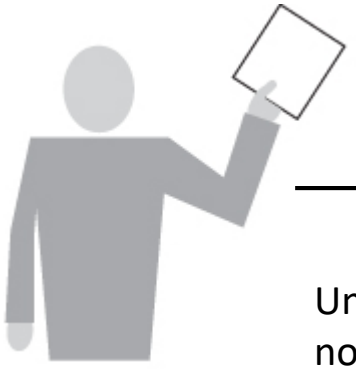
Attribut de qualité 5: Sécurité Modélisation d'attaques

Une approche utilisée dans le domaine de la sécurité consiste à modéliser les attaques à l'aide d'arbres d'attaques, similaire à l'analyse de fautes.

La racine de l'arbre est une attaque réussie. Les nœuds de l'arbre sont les causes directes possibles pour le succès de l'attaque. Chaque nœud et décomposé récursivement pour identifier les causes précises.

Une attaque tente de briser les CID. Les feuilles dans un arbre d'attaque sont les stimuli d'un scénario d'attaque. La réponse à une attaque vise à préserver les CID ou à repousser les attaquants par monitoring de leurs activités.

Scénario général de sécurité



Source du stimulus

Humain ou un autre système qui peut avoir été préalablement identifié, correctement ou non, ou qui peut être inconnu. Un humain peut être interne ou externe à l'organisation.

Stimulus

Une tentative non autorisée est faite pour: -
- afficher, modifier ou effacer des données,
- accéder à un service,
- changer le comportement du système ou réduire sa disponibilité.

Artéfact

Services du système, données contenues dans le système, une composante ou ressource du système, données produites ou consommées par le système

Environnement

Le système est soit en ligne ou hors ligne, connecté au réseau ou déconnecté, derrière un pare-feu ou ouvert sur le réseau, complètement, partiellement ou non opérationnel.

Réponse

Les transactions sont effectuées de façon à:
- Protéger les données ou services contre les accès non autorisés,
- Empêcher que les données ou services soient manipulés de façon non autorisée,
- Identifier de façon certaine les parties dans une transaction,
- Les parties ne peuvent pas répudier une transaction,
- Les données, ressources et services seront disponibles pour les utilisations légitimes.

Le système suit les activités en:
- Enregistrant les accès, les modifications, les tentatives d'accès aux données, ressources ou services,
- Avertissant les entités appropriées lorsqu'une attaque semble se produire.

Mesure de la réponse

Une ou plusieurs parmi:
- Combien du système est compromis lorsqu'une composante ou donnée est compromise.
- Combien de temps à passé avant que l'attaque soit détectée.
- Combien d'attaque ont été repoussées.
- Combien de temps est nécessaire pour récupérer après une attaque.
- Combien de données sont vulnérable à une attaque particulière.

Exemple de scénario concret de sécurité

Source du stimulus

Un employé insatisfait à partir d'un site distant

Stimulus
Tentative de modification du taux horaire de rémunération

Artéfact
Données contenues dans le système

Environnement
Mode d'opération normal

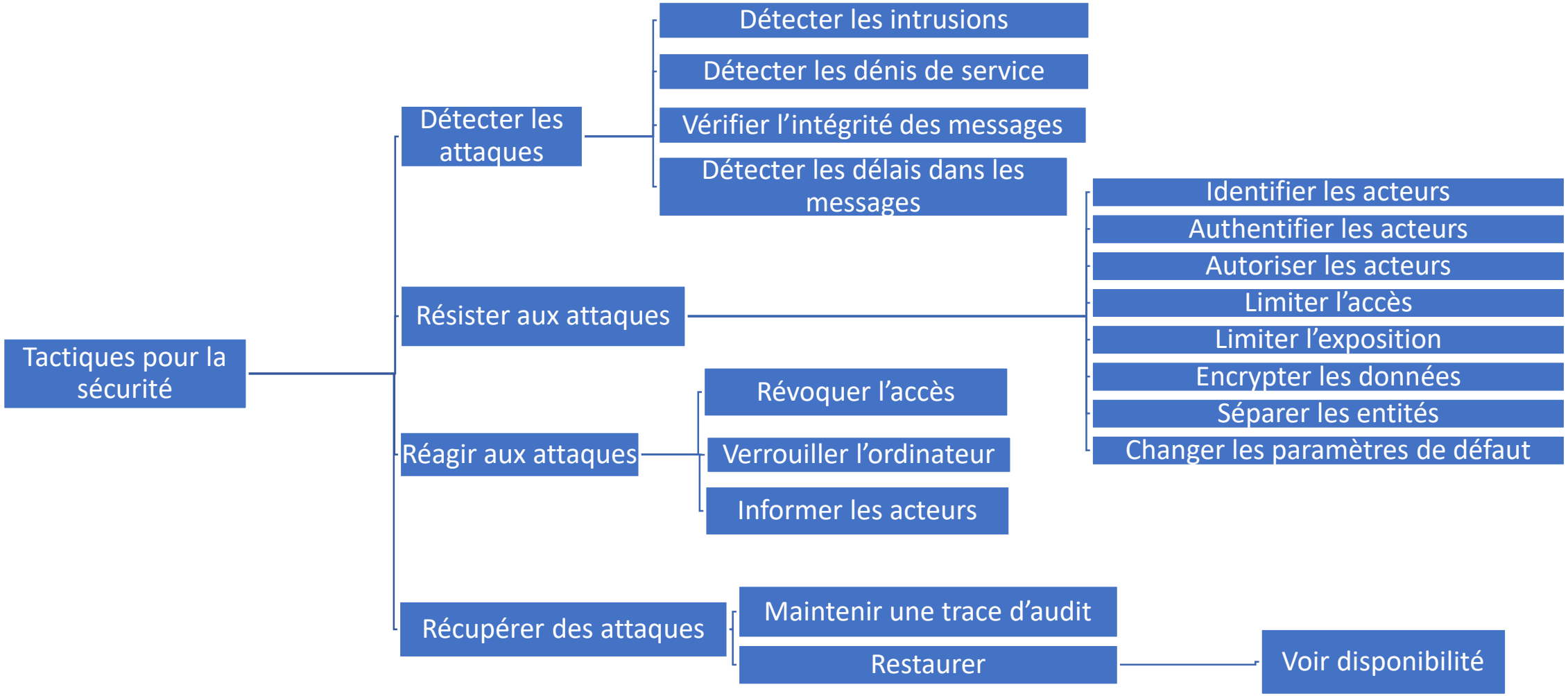
Réponse
Le système conserve une trace d'audit des transactions

Mesure de la réponse
Les données sont corrigées à l'intérieur d'une journée et la source de manipulation est identifiée.

Attribut de qualité 5: Tactiques pour la sécurité

Un système est sécure s'il est en mesure de conserver la confidentialité, l'intégrité et la disponibilité de ses données et de ses services.

Attribut de qualité 5: Tactiques pour la sécurité



Attribut de qualité 5: Tactiques pour la sécurité: **Détecter les attaques**

- Détecter les intrusions: comparaison du trafic réseau ou des patrons de requêtes de service **dans** le système avec un ensemble de signatures ou patrons connus de comportements malicieux stockés dans une base de données. Les signatures peuvent être basées sur le protocole, les indicateurs TPC, la taille de la charge utile, les applications, adresses sources ou destination ou les numéros de ports.
- Détecter les dénis de service: comparaison du trafic réseau ou des patrons de requêtes de service **qui arrive sur** le système avec un ensemble historique de profiles d'attaques de dénis de services connus.

Attribut de qualité 5: Tactiques pour la sécurité: **Détecter les attaques**

- Vérifier l'intégrité des messages: utiliser des techniques comme des sommes de contrôle ou des valeurs de hachage pour vérifier l'intégrité des message, fichiers de ressources, de déploiement ou de configuration. Les sommes de contrôle et les valeurs de hachage sont des données redondantes calculées par le système pour vérifier que des données n'ont pas été modifiées.
- Détecter les délais dans les messages: pour détecter les attaques de type « intermédiaire » (*man-in-the-middle*) où une tierce-partie malicieuse intercepte et peut modifier des messages. En vérifiant le temps pour livrer un message, il est possible de détecter des temps de livraisons suspects.

Attribut de qualité 5: Tactiques pour la sécurité: **Résister aux attaques**

- Identifier les acteurs: identifier la source de toute donnée d'entrée externe au système. Les utilisateurs sont généralement identifiés par des identificateurs. D'autres systèmes peuvent être identifiés par des codes d'accès, des adresse IP, des protocoles, des ports, etc.
- Authentifier les acteurs: authentifier signifie vérifier qu'un acteur est vraiment qui il prétend être. Les mots-de-passe, certificats, identification biométrique et autres fournissent des mécanismes d'authentification.
- Autoriser les acteurs: autoriser signifie de s'assurer qu'un acteur authentifié a les droits pour accéder ou modifier des données ou services. Implémenter par des mécanismes de contrôle d'accès dans le système, par un acteur ou une classe d'acteurs.

Attribut de qualité 5: Tactiques pour la sécurité: **Résister aux attaques**

- Limiter l'accès: contrôler quoi et qui peut accéder à quel partie du système. Ceci peut inclure limiter l'accès aux processeurs, à la mémoire et aux connexions réseau, ce qui peut être accompli en gérant les processus, en protégeant la mémoire, en bloquant un hôte, en fermant des ports ou en rejetant un protocole. P.ex. un mur pare-feu est un point unique d'accès à un système. Une zone démilitarisée est une sous portion de réseau entre l'internet et l'intranet, protégée par 2 pares-feux. Une zone démilitarisée est utilisée pour permettre à des utilisateurs externe d'avoir accès à des services qui devraient être disponible à l'extérieur de l'intranet.

Attribut de qualité 5: Tactiques pour la sécurité: **Résister aux attaques**

- Limiter l'exposition: réduire la probabilité de succès d'une attaque ou restreindre les dommages possibles si une attaque réussit. Peut être accompli en camouflant certains faits sur le système à protéger ou en divisant et distribuant les ressources critiques de façon à ce que l'exploitation d'une seule faiblesse ne compromette pas entièrement une ressource.
- Encrypter les données: les données doivent être protégées contre les accès non-autorisés. La confidentialité est généralement obtenue à l'aide d'une forme de cryptage des données et des communications. Au-delà de l'autorisation, le cryptage fournit une protection supplémentaire pour les données persistantes. Les communications sont souvent accessibles sans autorisation. Le cryptage est la seule protection pour transmettre des données sur un réseau ouvert. Implémentations possible avec des VPN ou des couches sécurisées SSL. Le cryptage peut être symétrique ou non.

Attribut de qualité 5: Tactiques pour la sécurité: **Résister aux attaques**

- Séparer les entités: la séparation des entités d'un système peut se faire par une séparation physique sur différents serveurs liés à des réseaux distincts; en utilisant des machines virtuelles; ou par un « espace d'air » i.e. en éliminant toute connexion entre différentes parties d'un système. Les données sensibles sont souvent séparées des données non sensibles pour réduire la possibilité d'une attaque par ceux qui ont accès aux données non sensibles.
- Changer les paramètres de défaut: forcer les utilisateurs à changer les paramètres de configuration d'un système pour empêcher des attaquants d'accéder au système en utilisant des paramètres qui sont connus publiquement.

Attribut de qualité 5: Tactiques pour la sécurité: **Réagir aux attaques**

- Révoquer l'accès: si le système ou un administrateur pense qu'une attaque est en cours, il peut limiter sévèrement l'accès aux données sensibles, même pour les utilisateurs et les utilisations légitimes.
- Verrouiller l'ordinateur: des tentatives infructueuses répétées peuvent indiquer qu'une tentative d'attaque est en cours. Beaucoup de systèmes limitent, temporairement, l'accès à un compte s'il détectent plusieurs échecs consécutifs d'accès à partir d'un ordinateur donné.
- Informer les acteurs: Durant une attaque, des actions peuvent être requises des opérateurs, d'autre personnel ou systèmes. Les acteurs impliqués dans le scénario d'attaque doivent être informés lorsque le système détecte une attaque.

Attribut de qualité 5: Tactiques pour la sécurité: **Récupérer des attaques**

Une fois qu'un système a détecté et tenter de résister à une attaque, il doit récupérer.

Une partie de la récupération consiste à restaurer les services. On peut voir une attaque réussie comme une sorte de faute, et dans ce cas, appliquer les stratégies de récupération en cas de faute liées à la disponibilité

Il faut aussi conserver une trace permettant d'effectuer un audit afin de possiblement identifier les actions et l'identité des attaquants.

Liste de vérification pour la sécurité
Allocation des responsabilités

- Déterminer quelles responsabilités du système doivent être sécures. Pour chaque responsabilité, s'assurer que des responsabilités additionnelles ont été allouées pour:
  - Identifier les acteurs,
  - Authentifier les acteurs,
  - Autoriser les acteurs,
  - Accorder ou refuser l'accès à des données ou services,
  - Enregistrer les tentatives d'accès ou de modification des données,
  - Encrypter les données,
  - Reconnaitre une disponibilité réduite aux ressources ou services et avertir le personnel approprié et restreindre l'accès,
  - Récupérer d'une attaque,
  - Vérifier les sommes de contrôle et les valeurs de hachage.

Liste de vérification pour la sécurité
Modèle de coordination

- Déterminer les mécanismes requis pour communiquer avec et coordonner d'autres systèmes ou individus. Pour ces communications, s'assurer que sont en place:
  - Des mécanismes d'authentification et d'autorisation des acteurs ou système
  - Du cryptage des données,
  - Des mécanismes de monitoring et de détection de demande élevée de ressources ou de services,
  - Des mécanismes pour restreindre ou fermer une connexion.

Liste de vérification pour la sécurité
Modèle de données

- Déterminer les données sensibles dans différent champs de données. Pour chaque abstraction de données, s'assurer que:
  - Les données avec différents niveau de sensibilité sont séparés,
  - Les données avec différents niveau de sensibilité ont différents droits d'accès et que les droits sont vérifiés avec de fournir l'accès,
  - L'accès aux données sensibles sont tracées et que les traces sont correctement protégées,
  - Les données puissent être restaurées si elles ont été modifiées de façon inappropriée.

Liste de vérification pour la sécurité
Correspondance entre les éléments architecturaux

- Déterminer comment les différentes variations possibles de correspondance entre les éléments architecturaux peuvent changer comment un individu ou un système peut:
  - Lire, écrire, ou modifier des données,
  - Accéder à des services ou ressources du système,
  - Réduire la disponibilité à des services ou ressources du système.
- Déterminer comme ces différentes variations peuvent affecter l'enregistrement des accès aux données, services ou ressources et la détection d'une demande plus élevée qu'attendue de ressources.

Liste de vérification pour la sécurité
Correspondance entre les éléments architecturaux

- Pour chaque variante de correspondance, s'assurer que des responsabilités ont été prévues pour:
  - Identifier les acteurs,
  - Authentifier les acteurs,
  - Autoriser les acteurs,
  - Accorder ou refuser l'accès à des données ou services,
  - Enregistrer les tentatives d'accès ou de modification des données,
  - Encrypter les données,
  - Reconnaitre une disponibilité réduite aux ressources ou services et avertir le personnel approprié et restreindre l'accès,
  - Récupérer d'une attaque.

Liste de vérification pour la sécurité
Gestion des ressources

- Déterminer les ressources système nécessaires pour identifier et monitorer un système ou un individu qui est interne ou externe, autorisé ou non, et qui a accès à des ressources spécifiques ou à toutes les ressources.
- Déterminer les ressources nécessaires pour:
  - authentifier un acteur,
  - accorder ou refuser l'accès à des données ou ressources,
  - avertir les entités appropriées,
  - enregistrer les tentatives d'accès à des données ou ressources,
  - encrypter les données,
  - reconnaitre une demande extraordinairement élevée de ressources,
  - informer les utilisateurs ou systèmes, et
  - restreindre l'accès.

Liste de vérification pour la sécurité
Gestion des ressources

- Pour ces ressources, évaluer si une entité externe peut accéder à des ressources critiques ou épuiser une ressource critique; comment monitorer la ressource; comment gérer l'utilisation de la ressource; comme tracer l'utilisation de la ressource; et s'assurer qu'il y a des ressources suffisantes pour effectuer les opérations de sécurité nécessaires.
- S'assurer qu'un élément contaminé peut être empêché de contaminer d'autres éléments.
- S'assurer que des ressources partagées ne peuvent pas être utilisées pour passer des données sensible d'un acteur ayant accès les droits d'accès aux données à un acteur n'ayant pas les droits d'accès à ces données.

Liste de vérification pour la sécurité
Choix du moment pour effectuer une liaison

- Déterminer les cas où une instance de composante liée tardivement peut ne pas être fiable. Dans ces cas, s'assurer que les composantes liées tardivement peut être qualifiées:
  - Si des certificats de propriété pour des composantes liées tardivement sont requis, qu'il y a des mécanismes appropriés pour les gérer et les valider;
  - Que l'accès à des données et services liés tardivement peut être gérés;
  - Que les accès par des composantes liés tardivement à données et services peuvent être bloqués;
  - Que des mécanismes pour enregistrer l'accès, la modification ou les tentatives d'accès à des données ou services par des composantes liées tardivement sont en place; et
  - Que les données systèmes sont encryptées avec les clés intentionnellement cachées aux composantes liées tardivement.

Liste de vérification pour la sécurité
Choix de technologie

- Déterminer quels technologies sont disponibles pour:
  - Authentifier les usagers,
  - Gérer les droits d'accès,
  - Protéger les ressources, et
  - Encrypter les données.
- S'assurer que les technologies choisies supportent les tactiques pertinentes à vos besoins de sécurité.