

## Exercice ATAM sur le laboratoire

1. Présentation du processus ATAM par moi-même
  1. Séparer la classe en 2 groupes distinct
    - i. L'équipe qui partage son projet
    - ii. Toute les autres équipes d'évaluation qui prennent des notes
    - iii. Les parties prenantes (Tout le monde)
  2. Présentation des déclencheurs d'activité commerciales (par moi-même)
    - OA-1. Rendre le laboratoire intéressant pour les étudiants.
    - OA-2. Permettre aux étudiants de constater l'importance de l'architecture logicielle.
    - OA-3. Faciliter le recrutement des nouveaux chargés de laboratoire.
    - OA-4. Faciliter le processus de correction des laboratoires.
    - OA-5. Arrimer le laboratoire (pratique) et le cours (théorique)
    - OA-6. Faire expérimenter le travail d'architecte logiciel aux étudiants.
    - OA-7. Améliorer la qualité de vie des citoyens de la ville de Montréal

### 3. Fonctionnalité importantes

**CU01.** Veut comparer les temps de trajet

**CU02.** Veut pouvoir mettre le chaos dans les services en

**CU04.** Veut pouvoir perturber le trafic à l'aide d'une interface usager pour simuler un accident ou une période de pointe

**CU05.** Simulation donnée GPS

**CU06.** Veut simuler l'envoi simultané des coordonnées GPS de 100 appareils mobiles pendant au moins une minutes pour s'assurer du bon fonctionnement du système même lorsqu'il est soumis au chaos.

**CU07.** Service externe permettant de récupérer le temps de trajet entre deux intersections. Chaque équipe doit implémenter au moins un service externe

### Attributs de qualité dominants:

**AQD1.** Vous devez implémenter un mécanisme de monitoring pour tous les microservices.

**AQD2.** Chaque microservice doit supporter le mode écho (echo).

**AQD3.** Chaque microservice doit supporter le mode heartbeat.

**AQD4.** Chaque équipe est responsable d'implémenter une redondance passive de l'un de ses microservice

**ADQ5.** Chaque équipe est responsable d'implémenter une redondance active sur l'un de ses microservice.

**AQM1.** Vous devez être en mesure de modifier les intersections utilisées (ajouter ou retrait) seulement en modifiant un fichier de configuration ou à l'aide de l'interface usager. L'application des changements à votre solution ne devrait pas prendre plus de 15 secondes une fois le fichier de configuration complété.

**AQM2.** Vous devez être en mesure d'adapter votre solution pour utiliser les microservices d'une autre équipe en moins d'une heure.

**AQM3.** Vous devriez être en mesure de généraliser la redondance passive pour n'importe quel composant en moins d'une heure.

**AQM4.** Vous devriez être en mesure de généraliser la redondance active pour n'importe quel composant en moins de 3 heures.

**AQT1.** Ajoutez la tactique "Interface spécialisé" pour permettre de modifier dynamiquement la latence des microservices de votre architecture.

**AQT2.** Utilisez la tactique "Abstraire la source de données" pour spécifier la source de donnée.

**AQT3.** Utilisez la tactique "Limiter opération non déterministe" pour faciliter l'exécution de vos test.

**AQT4.** Utilisez la tactique Record/playback pour pouvoir tester à nouveau votre système avec des données déjà reçu.

**AQT5.** Utilisez la tactique "Interface spécialisé" pour suivre à la trace tous les messages entrant dans les microservices par l'association d'un numéro unique à chaque message entrant. Ceci devrait permettre de faire la trace de tous les microservices actifs ayant été utilisés pour traiter un message.

**AQU1.** Utilisez des interfaces usager au lieu de fichier de configuration pour faciliter la configuration de vos microservices.

**AQU2.** Utilisez la tactique "Aggregation" pour applique des commandes ou configuration à de multiple microservice simultanément.

**AQP1.** Vous devez être en mesure de configurer votre système pour utiliser les microservices ayant les plus faible latences pour réaliser le calcul du temps de trajet. Sont inclus les microservices développés par toutes les équipes de votre sous-groupe.

**AQP2.** Proposer un scénario de performance pour que votre système soit en mesure de supporter 1000 clients simultanément.

Démontrez mathématiquement que la solution proposée est réalisable.

**AQI1.** Vous devez pouvoir utiliser les services fournis par les autres équipes du laboratoire uniquement en modifiant dynamiquement la configuration de votre système.

**AQI2.** Votre système devrait pour récupérer la liste des microservices qu'il a besoin pour s'acquitter de sa tâche et d'utiliser ceux-ci dynamiquement.

**AQS1.** Vous devez mettre en place un mécanisme pour que votre service ne soit accessible que pour les équipes autorisées.

**AQS2.** Appliquez la tactique "Maintien audit trail" dans un service centralisé pour faciliter la détection de tentative non autorisée de connexion. Vous devriez enregistrer l'adresse IP de l'appelant pour chaque tentative.

#### 4. Présentation de l'architecture (Par l'équipe d'architecture)

#### 5. Identification des approches architecturales (par l'équipe d'évaluation)

##### 1. Tactiques;

**AQD1.** Vous devez implémenter un mécanisme de monitoring pour tous les microservices.

Tactiques: Ping + heartbeat, Service discovery fait le ping tous les autres font le echo.

Service discovery reçoit les heartbeat, tous les microservices envoient les heartbeat. Vue C&C

AQT2. Utilisez la tactique "Abstraire la source de données" pour spécifier la source de donnée. ServiceSwitch implémente la tactique, adresse provient de serviceDiscovery. On utilise un UI pour configure la switch et choisir la source de données. Vue module
AQM2. Vous devez être en mesure d'adapter votre solution pour utiliser les microservices d'une autre équipe en moins d'une heure. ????
AQT1. Ajoutez la tactique "Interface spécialisé" pour permettre de modifier dynamiquement la latence des microservices de votre architecture. ChaosMonkey -> modifié la latence dans service discovery, vue module pour montrer l'interface
AQD2. Chaque microservice doit supporter le mode écho (echo). Voir AQD1
AQP2. Proposer un scénario de performance pour que votre système soit en mesure de supporter 1000 clients simultanément. ??
AQD3. Chaque microservice doit supporter le mode heartbeat. Voir AQD1
AQI1. Vous devez pouvoir utiliser les services fournis par les autres équipes du laboratoire uniquement en modifiant dynamiquement la configuration de votre système. Définition des interface dans des vues de type module, et partage d'information entre les équipe

2. Heuristiques;
3. Styles architecturaux
  - i. Module;
    1. Utilise
    - 2.
  - ii. C&C;
    1. SOA
    2. Client/Serveur
  - iii. Allocation.
    1. Déploiement
    2. Assignment de tâches/ équipe vs microservice / élément de microservice vs étudiant
    3. Instalation?
4. Autre.
6. Construction de l'arbre d'utilité des attributs de qualité (équipe d'évaluation)  
Stimulus, environnement, réponse, mesure de la réponse.  
  
Demander aux étudiants de compléter l'arbre d'utilité avec la mesure et la priorité.  
Priorité 1 Importance pour le projet  
Priorité 2 Difficulté de réalisation
7. Analyse des approches architecturales (équipes d'évaluation)

Motivée par les feuilles de l'arbre qui ont des priorités élevées

- demander à l'équipe qui présente de démontrer si leur architecture satisfait ces scénarios.. (2-3 scénarios sinon on manque de temps)

Identification des risques, non-risques, points sensibles et points de compromis

Scénario	Risque	Non risque	sensibilité	Compromis
AQD1	Heartbeat: 30s – ne pas détecter un microservice en panne assez rapidement Ping/echo - >10sec		Sur la disponibilité si la fréquence est trop basse.	Peut avoir un impact sur la performance si fréquence trop élevée Fréquence du ping/echo
AQD2		Interface commune à toutes les équipes		
AQD3				
AQT1				
AQT2				
AQM2		Interface commune à toutes les équipes		
AQP2	Service externe ne soit pas disponible			(n) Multiple copie of computation performamnce vs disponibilité
AQI1		Interface commune à toutes les équipes		

Risques

Non Risques

Points de sensibilité

Compromis

## 8. Remue-méninges et assignation de priorités aux scénarios (par votes 30%)

Chaque personne propose un scénario (actuel, futur et exploratoire) et on les priorise tous ensemble

Priorité 1 : Importance pour le succès du projet

Priorité 2: Difficulté d'implémentation demander aux étudiants High, Medium, Low

Utiliser les objectifs d'affaire pour vous inspirer

- OA-1. Rendre le laboratoire intéressant pour les étudiants.

Formation des connections entre les microservices sur Portainer par les chargés de laboratoire

- OA-2. Permettre aux étudiants de constater l'importance de l'architecture logicielle.

Majorité du temps passé sur les détails d'implémentation et setup

+ d'exemple d'adventure builder

Plus tot dans la session (mslite, tmx)

- OA-3. Faciliter le recrutement des nouveaux chargés de laboratoire.

- OA-4. Faciliter le processus de correction des laboratoires.

Gabarit de rapport détaillé

Enlever l'aspect de compétition entre les équipes et conserver celle entre les sous-groupes (entraide)

- OA-5. Arrimer le laboratoire (pratique) et le cours (théorique)

Demander aux équipes de réaliser leur propres scénario, faire arbre d'utilité par sous groupe et priorisé

Socrative: Question anonymes

- 

- OA-6. Faire expérimenter le travail d'architecte logiciel aux étudiants.

1er itération chaque équipe propose une architecture à l'équipe de coordonnateur

Réutiliser des services existant?

- OA-7. Améliorer la qualité de vie des citoyens de la ville de Montréal

## 9. Ré-exploration des approches architecturales

Motivée par les feuilles de l'arbre qui ont des priorités élevées

- demander à l'équipe qui présente de démontrer si leur architectures satisfait ces scénarios..

Identification des risques, non-risques, points sensibles et points de compromis

Scénario	Risque	Non risque	sensibilité	Compromis
----------	--------	------------	-------------	-----------


10. Présentation finale / rapport  
Voir ce document