



LOG430 Architecture logicielle

Objectif du laboratoire

L'objectif principal des 4 itérations de ce laboratoire est de vous permettre de maîtriser les concepts d'analyses et de conception d'une architecture logicielle d'un système complexe. Cette architecture sera développée sur la base d'une architecture de microservices. Le système fourni offre un service permettant de faire la comparaison de temps de trajet à partir de données sur les autobus de la STM et de services externe fournissant des données sur les automobiles.

4 itérations

Itération #1 (3 semaines)

- Intégration des mécaniques de télémétrie

Itération #2 (3 semaines)

- Réalisation de tactiques de disponibilité
- ChaosMonkey détruit aléatoirement les conteneurs de calcul

Itération #3 (3 semaines)

- Réalisation des tactiques de performance
- ChaosMonkey s'attaque aux connecteurs (message queue, bus, etc.)
- Chaosmonkey attaque à la limitation du nombre d'instruction par secondes
- ChaosMonkey s'attaque à la mémoire disponible pour les containers

Itération #4 (3 semaines)

- ChaosMonkey s'attaque au Data store (BD, cache, etc.)

Le contexte de l'organisation

Vous êtes nouvellement embauché par l'organisation LOG430STM pour améliorer l'architecture du comparateur de trajet. La réussite de ce projet n'est pas optionnelle. La carrière des étudiants peut

grandement être impactée s'ils échouent à ce cours. C'est pour cette raison que l'organisation a décidé de séparer les responsabilités selon les différentes [parties prenantes](#).

Parties prenantes du projet

Chargé de laboratoire (client)

- Effectuera l'évaluation de l'architecture de chaque équipe (Documentation, Intégration et Implémentation)
- Responsable de répondre aux questions des étudiants (durant les périodes de laboratoire seulement)
- Responsable d'aider les étudiants à maîtriser les concepts d'architecture
- Veux un rapport détaillé de l'architecture et des interfaces

Équipe (3 Étudiants)

- L'équipe doit conserver une traçabilité de quel étudiant est responsable de quelles tâches. Ceci correspond à une vue d'allocation à insérer dans votre rapport.
- L'équipe doit connaître en tout temps l'état d'une tâche assignée à un étudiant
- Le chargé de cours crée les équipes de laboratoires.
- Les équipes doivent utiliser le Kanban de Github pour planifier et réaliser le projet. Vos Kanban doivent avoir au minimum les colonnes suivantes:
 - **backlog**: idée générique de tâches à considérer, priorisée par l'équipe.
 - **todo**: ce qu'on fait à la prochaine itération, à faire, assigner explicitement ou non.
 - **in progress**: tâches sur lesquels vous travaillez présentement. Généralement, une seule tâche par étudiant
 - **review**: les tâches qui doivent être révisées par un autre étudiant de l'équipe. Idéalement sous forme de pull request.
 - **Done**: le pull request a été accepté et la tâche est terminée.

Représentant d'équipe (1 représentant par équipe)

- Le représentant d'équipe est responsable de communiquer avec le chargé de laboratoire pour les questions concernant l'équipe.
- Il participe aux réunions avec le chargé de laboratoire.

Contraintes de réalisation

Language¶

Nous n'imposons aucune contrainte au niveau du langage de développement utilisé à l'exception que celui-ci doit être de type **Orienté objet**.

Directive de déploiement

Vous pouvez déployer votre solution de l'itération #1 sur n'importe quel serveur.

Directive de Démonstration

- Vous n'aurez droit qu'à une seule démonstration pour l'intégration et/ou l'implémentation de chaque exigence.
- Chaque équipe disposera d'un maximum de 10 minutes par démonstration/exigence.
- Donc soyez bien préparé
 - Assurez-vous d'avoir testé votre architecture
 - Assurez-vous de ne pas faire des modifications de dernières minutes qui pourraient impacter votre démonstration
- À chaque démonstration, le chargé de laboratoire peut vous demander de créer des tâches que vous devrez avoir satisfaites lors de la démonstration subséquente. Le non-respect de cette directive pourrait entraîner des pertes de points.

Directive de remise

Vous devez mettre vos sources à jour dans la branche main, et ensuite vous générez un tag correspondant à l'itération ou vous faites votre remise. **Les remises doivent se faire avant minuit le jour de la séance de laboratoire correspondant à la semaine identifié dans le tableau suivant.**

Semaine	Tag
3	git tag itération1
6	git tag iteration2
9	git tag iteration3
12	git tag iteration4

- Il y aura des directives supplémentaires pour la remise de la documentation de chaque itération.

Parasites et mollasson

À la fin de chaque itération, les membres de l'équipe doivent réaliser leur propre évaluation par les pairs pour chacun des membres de l'équipe. Référez-vous à l'article Parasites et mollasson pour vous aider à faire l'évaluation des autres étudiants. Voici en référence un exemple de [fichiers Excel d'évaluation par les pairs](#) dans le répertoire DOC. Vous pouvez utiliser ce fichier pour faire vos évaluations par les pairs. Vous devez indiquer clairement dans votre rapport d'itération le résultats de cette évaluation et spécifie si nous appliquons celui-ci à la note de votre laboratoire.