



LOG8430: Architecture logicielle et conception avancée

Hiver 2022

©M. Fokaefs et F. Guibault

Enseignant

- Marios Fokaefs – Professeur Adjoint
- <http://www.mariosfokaefs.com>
- marios.fokaefs@polymtl.ca
- Local M-4119
 - Virtuel Webex: <https://polymtl.webex.com/meet/marios.fokaefs>
- Recherche
 - Systèmes distribués : Infonuagique, services de web, internet des objets (IoT)
 - Évolution et maintenance du logiciel, systèmes auto-adaptatifs
 - Économie et gestion du logiciel, DevOps/BizOps, qualité et sécurité

Enseignant – Marios Fokaefs

- Diplôme en science informatique - Grèce
- Maîtrise en génie logiciel – Alberta
 - Identification des opportunités de refactoring « Extraction de Classes » dans les systèmes OO
- Doctorat en génie logiciel – Alberta
 - Évolution des services du web
 - Gestion économique des services du web
- Associé de recherche (postdoctorat) – York
 - Systèmes auto-adaptatifs d'infonuagique
 - Gestion économique de l'infonuagique et des logiciels distribués
- En GIGL depuis septembre 2017



Mohammadreza Rasolroveicy - TA for Software Architecture

- PhD Student at Polytechnique
- Research interests: DevOps - AutoScaling - Blockchain - IoT Security - Cloud Computing
- Email : mohammadreza.rasolroveicy@polymtl.ca
- Office : 4218 - Department of Computer & Software Engineering , Polytechnique Montréal

Wejdene HAOUARI

chargée de travaux pratiques - Architecture logicielle
et conception avancée

- **Étudiante à la maîtrise** à Polytechnique Montréal
- **Domaines de recherche:**
Blockchain - DevOps - Cloud Computing - CyberSecurity
- **Email :** wejdene.haouari@polymtl.ca
- Bureau : 4218 - Département de génie informatique et logiciel , Polytechnique Montréal



Objectifs du cours

- Présenter les styles et les patrons d'architecture et de conception logicielle en mettant l'accent sur les systèmes distribués.
- Mesurer et maintenir la qualité de la conception et de l'architecture.
- Explorer et pratiquer la conception de logiciels distribués dans des cas réels : cadriciels, systèmes infonuagiques et systèmes d'analyse de données.



À la fin du cours, l'étudiant sera en mesure de

- Concevoir l'architecture d'un logiciel en choisissant et en justifiant les styles d'architecture et les patrons de conception selon les exigences.
- Concevoir les composantes de l'architecture en utilisant des modèles spéciaux et des nouvelles technologies (architectures évènementielles, orientées services, orientées objets, infonuagiques).
- Évaluer la qualité d'une architecture ou d'une conception logicielle, et la maintenir en appliquant des changements.

Philosophie du cours

- Il s'agit d'un cours de quatrième année.
 - Il ne vous sera pas demandé de concevoir ou de développer du code.
 - On suppose que vous connaissez déjà les principes de la conception logicielle (dont nous retiendrons ici).
- Pouvez-vous reconnaître, récupérer, évaluer une conception ou une architecture logicielle dans laquelle vous n'étiez pas impliqué ?
- Scénario : Vous venez d'être embauché par une entreprise, vous héritez d'un système existant et vous êtes intégré dans une équipe existante.
- On vous demande d'aider les autres à comprendre un système que vous n'avez pas conçu.
- Apprendre à communiquer et à présenter vos découvertes au personnel technique et non technique.

Situation du cours

- LOG8430 termine une chaîne des cours :
 - INF1005C – programmation procédurale
 - INF1010 – programmation orientée objet
 - LOG2410 – conception logicielle
 - LOG3210 - Éléments de langages et compilateurs
 - LOG3430 - Méthodes de test et de validation du logiciel

Situation du cours

- LOG8430 est aussi un cours gradué
- Être à l'aise pour lire (en anglais) et écrire.
- Faire de la recherche
 - Il n'existe pas toujours une seule réponse.
 - Justifiez et vérifiez vos réponses.
 - Pensée critique.
 - Collaboration!
- Bien travaillez en équipe.
- Concepts avancés sur
 - OO
 - SOA
 - Infonuagique
 - Systèmes distribués

Format du cours (en présentiel)

- 2 heures de cours magistraux – 1 heure de pratique en classe
 - M-1510
 - Exercices sur le cours du jour.
- 3 heures de labo par 2 semaines
 - Première séances de labo
 - mercredi 19 janvier, groupe 02I (B2) (L-3818)
 - lundi 24 janvier, groupe 01I (B1) (L-3818)
 - Outils
 - Exemples de systèmes
 - Travaux Pratiques
- Mesures de sécurité
 - Gardez une distance physique (asseyez-vous avec deux sièges entre vous et vous pouvez enlever vos masques, ou avec un siège et essayez de garder vos masques)
 - Ne vous approchez pas du podium ! Comme je n'approcherai pas le public. Si vous avez une question, vous pouvez la poser depuis votre siège, même après la fin du cours.
 - Un canal Slack sera maintenu, pour faciliter davantage la collaboration. Apportez vos ordinateurs portables ou appareils portables.

Format virtuel du cours

- Cours magistraux préenregistrés sur YouTube.
 - Vous êtes invités de regarder les vidéos avant le cours de chaque semaine et préparer vos questions pour discussion.
 - Les vidéos ne seront pas synchronisés avec la matière actuelle du cours, mais ils peuvent également fournir une très bonne approximation de la matière.
- An espace Slack sera maintenu pour l'interaction directe.
 - Il y aura des canaux pour les séances, les labos, les TPs et les examens.
 - Vous pouvez poser vos questions là à tout temps, mais l'espace sera monitoré plus attentivement pendant les heures du cours.
- Des sessions interactives en direct seront organisées sur WebEx et leur enregistrement sera disponible à Moodle après.

Programme du cours

Cours	Contenu	Date	Date limite/Quiz
01	Introduction Concepts de base Principes SOLID Patrons de conception	14 janvier	
02	Architectures distribuées Patrons et styles d'architecture Cadrage Écosystèmes	21 janvier	
03	Qualité de la conception Attributs de qualité Métriques de qualité	28 janvier	
04	Mauvaise Conception Antipatrons / design smells Refactoring, refactoring to patterns	4 février	



Programme du cours

Cours	Contenu	Date	Date limite/Quiz
05	Architectures logicielles pour les mégadonnées: Insertion et analytique	11 février	Quiz 1
06	Architectures logicielles pour les mégadonnées: Bases des données	18 février	TP1 (14 février)
07	Blockchain	25 février	
--	Semaine de relâche	4 mars	

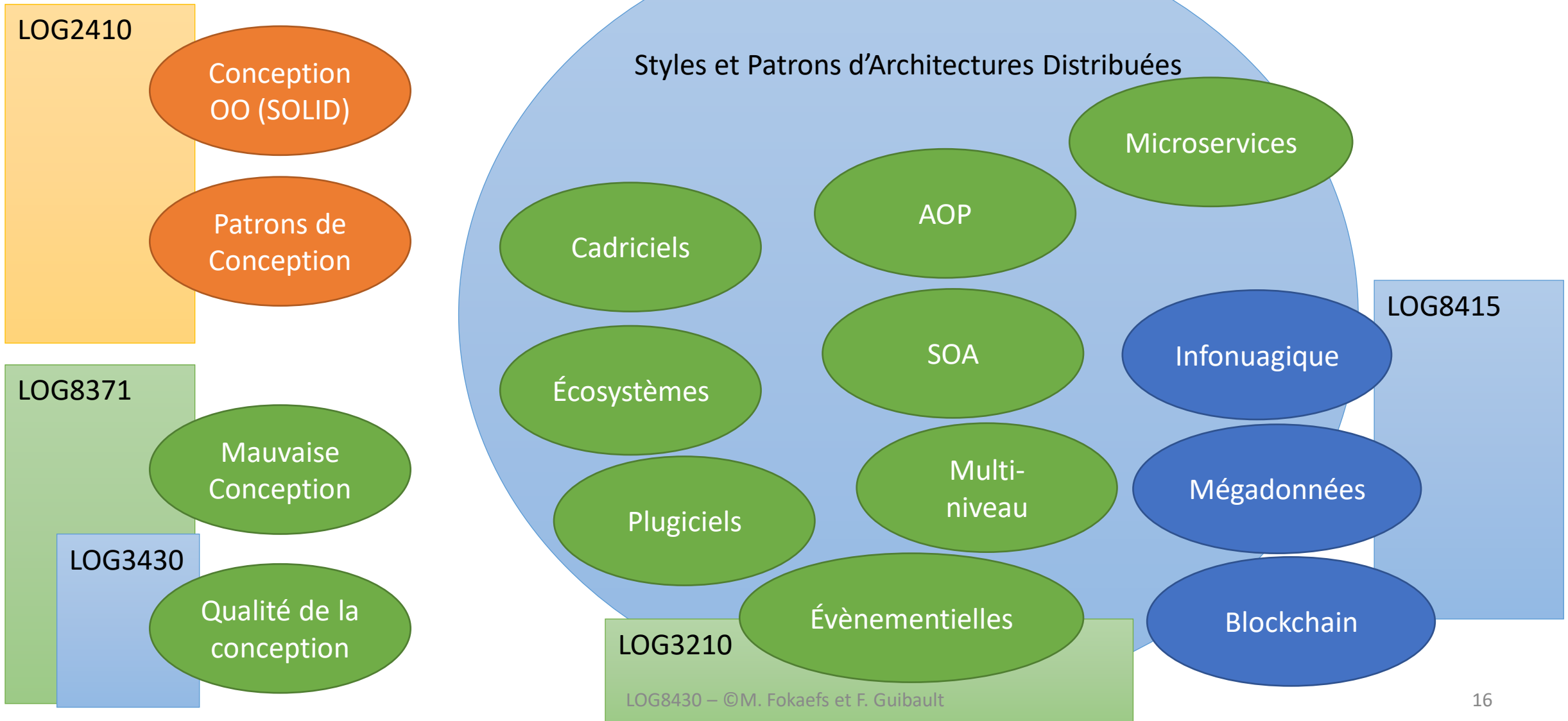


Programme du cours

Cours	Contenu	Date	Date limite/Quiz
08	Architectures événementielles Patrons de concurrence	11 mars	Quiz 2
09	Architectures orientées services (SOA)	18 mars	TP2 (21 mars)
10	Architectures microservices	25 mars	
11	Programmation orientée aspect (AOP)	1 avril	
12	Rappel – Présentations de TP3	8 avril	Quiz 3
13	Présentations de TP3	19 avril	TP3 (présentations: 18 avril, articles: 21 avril)



Carte du cours



Ressources

- Pas de manuel principal, mais plusieurs ressources pour étudier et approfondir la matière :
 - Liens
 - Tutoriels
 - Manuels
 - Articles
 - Logiciels
- Disponibles sur Moodle pour chaque séance.

Outils

- IDE
 - Android Studio - <https://developer.android.com/studio>
 - Eclipse - <https://www.eclipse.org/>
 - IntelliJ IDEA - <https://www.jetbrains.com/idea/>
- Git
 - GitHub - <https://github.com/>
 - Bitbucket - <https://bitbucket.org/>
- Conception et modélisation
 - Papyrus - <https://www.eclipse.org/papyrus/>
 - ArgoUML - <http://argouml.tigris.org/>
 - Enterprise Architect - <https://sparxsystems.com/products/ea/>
 - Understand - <https://scitools.com/features/>
 - PlantUML Integration - <https://plugins.jetbrains.com/plugin/7017-plantuml-integration>
 - PlantUML Diagram Generator - <https://plugins.jetbrains.com/plugin/15991-plantuml-diagram-generator>
 - Sketch It! - <https://plugins.jetbrains.com/plugin/10387-sketch-it->
- Analyse statique
 - JDeodorant - <https://marketplace.eclipse.org/content/jdeodorant>
 - Ptidej - <http://www.ptidej.net/>
 - SonarCloud - <https://sonarcloud.io/>
 - Android Studio
- Métriques
 - Understand
 - Eclipse Metrics 3 - <https://github.com/qxo/eclipse-metrics-plugin>
 - MetricsReloaded - <https://plugins.jetbrains.com/plugin/93-metricsreloaded>
 - MetricsTree - <https://plugins.jetbrains.com/plugin/13959-metricstree>

Important

- Toute la matière présentée pendant les séances de cours ou de labo peut faire l'objet d'une évaluation.
- Le contenu du cours est dynamique!
- La matière est déjà disponible sur Moodle en français et en anglais.
 - Des enregistrements des séances existent des anciennes sessions.
 - Les vidéos ne seront pas maintenus en sync avec les diapos, mais ils sont une bonne source pour comprendre la matière.
- Des annonces précéderont les mises à jour du matériel et du cours.
- Les travaux remis en retard seront pénalisés de 10% par jour de retard.
- Toute demande d'examen différé doit passer par le bureau des affaires académiques.
- La collaboration est permise pour les TP, mais les règlements relatifs au plagiat s'appliquent en tout temps.

Évaluation du cours

- Travaux Pratiques 60%
 - TP1 15%
 - TP2 15%
 - TP de Discussion 30%
- Quizz (3) 18% (6% chacun)
- Examen Final 22%
- Au moins 40% à l'examen final pour passer le cours.

Travaux Pratiques

- En équipe (3-4 max)
 - Utilisez le lien « Choix de groupe » sur Moodle avant le 24 janvier.
- Objectifs
 - Maîtriser la théorie
 - Compréhension et analyse
 - Conception et décision
- 2 travaux pratiques + 1 TP de discussion (écriture d'un article).
- 2 séances de labo par TP : la première session va démontrer des tutoriels, la deuxième va répondre aux questions et résoudre des problèmes.
- Travaux pratiques : Analyse d'un système (récupérer l'architecture, identifier des problèmes de conceptions, etc.)
 - Préparer un rapport sur l'analyse.
- TP de discussion : Comparer des solutions alternatives sur un problème de conception/architecture, exécuter des systèmes pour les évaluer.
 - Préparer un article scientifique pour décrire le concept de base, la comparaison des alternatives et les expérimentations de l'évaluation.

Examen final et Quizz

- Évaluation individuelle
- 3 Quizz pendant le trimestre
 - Dates possibles : 11 février, 11 mars, 8 avril
 - Chacun 6% de la note totale du cours)
- Examen final : 5 questions
 - 22% du total du cours
 - Vous avez besoin au moins 40% afin que les TP et les quizz comptent.
- Si vous manquez un quiz, vous devez m'informer avant le quiz avec une excuse d'absence valide.
 - La note du quiz manqué sera factorisée dans la note de l'examen final. ATTENTION! L'examen final sera plus difficile que le quiz.
 - Vous ne pouvez manquer qu'un seul quiz pendant la session.
- Théorie et pratique
 - Comme les exercices en classe.
- Exemples de systèmes discutés en classe, au labo ou dans les TPs.
- Dépendant de la situation
 - Virtuel : Les quizz et l'examen final seront effectués comme des Moodle Quiz en ligne.
 - En présentiel : En cahier dans la salle (pendant la séance) ou organisé par le registrariat.

Sessions Interactives

- Exercices et quizz sur le sujet de la semaine.
- Apprentissage par la mise en pratique.
- Améliorer et évaluer votre capacité de prendre des décisions, les expliquer et les justifier.
- Il n'y aura pas de note pour ces exercices.
- MAIS! Des exemples de ces exercices pourraient être utilisés à l'examen et aux quizz.

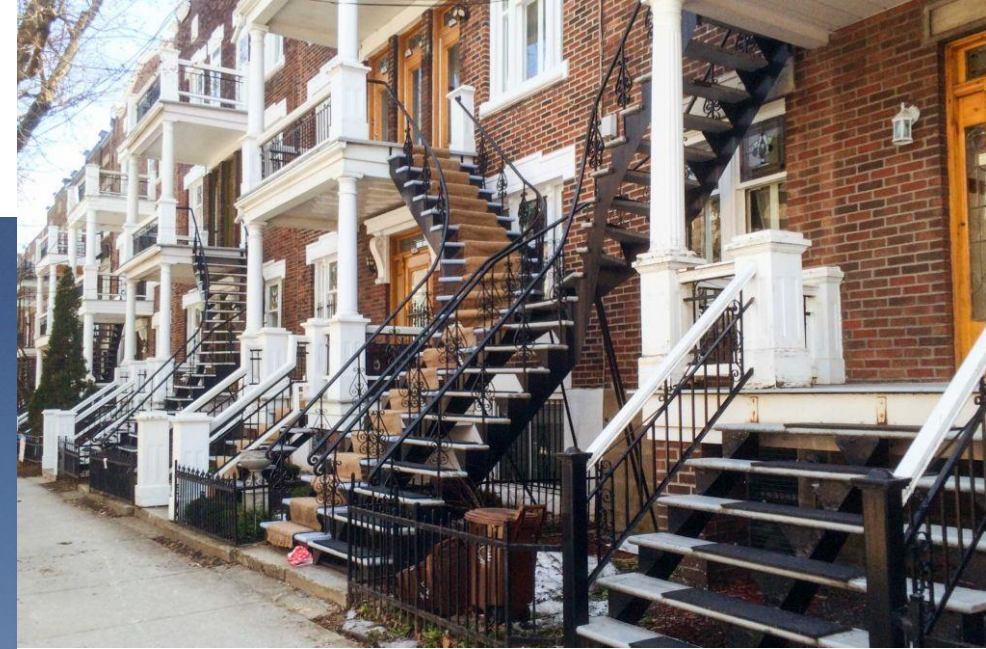
3 Questions sur l'architecture logicielle

1. Qu'est-ce qu'une architecture logicielle ?
2. Quels sont les critères qui doivent guider un architecte logiciel dans le choix d'un style architectural particulier ?
3. Quel impact a le choix d'un style architectural sur le développement d'un système logiciel ?

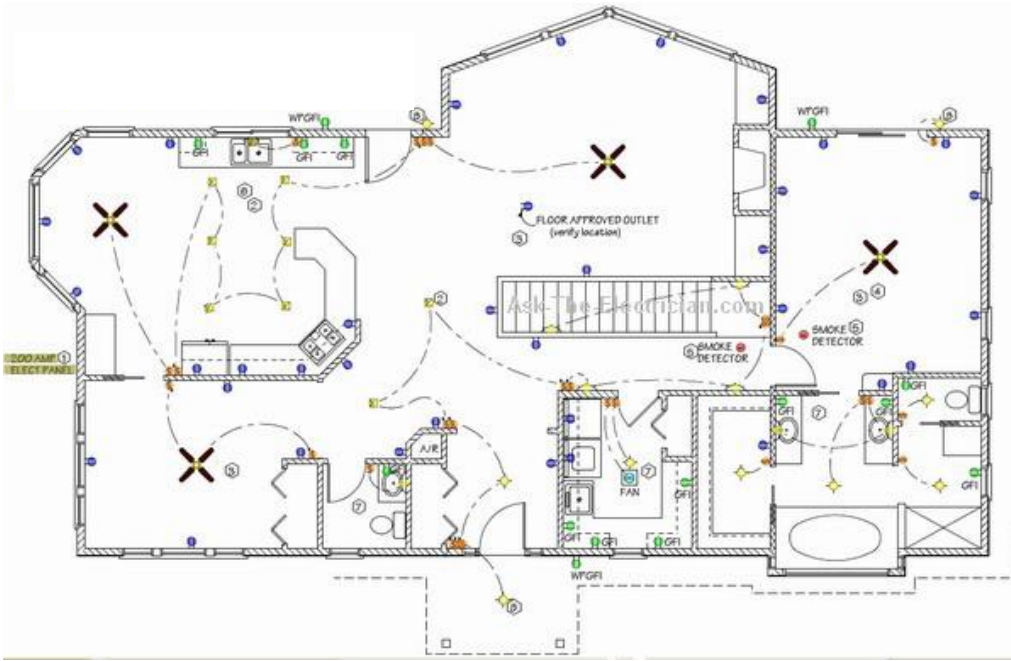
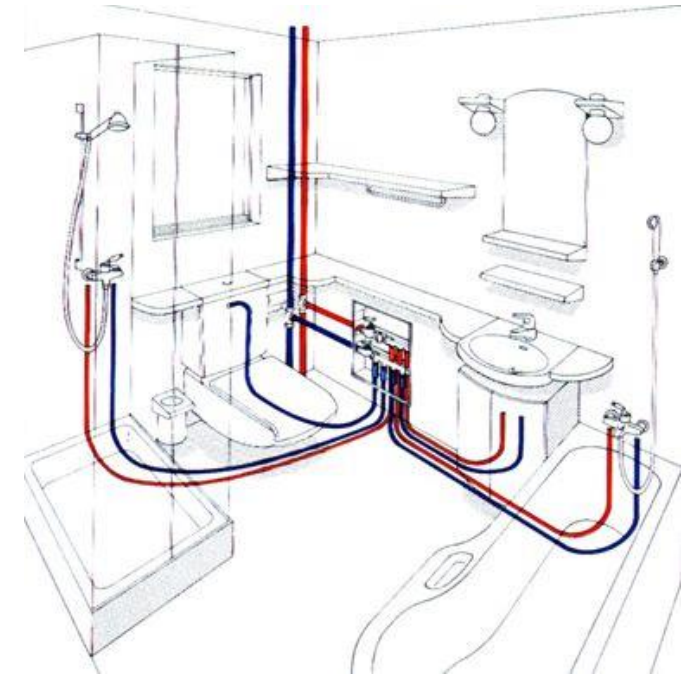
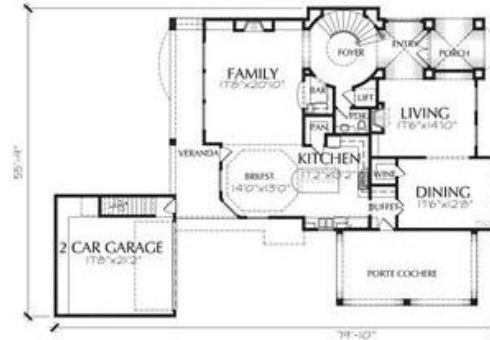
3 Questions sur la conception logicielle

1. Quelle est la différence entre l'architecture et la conception logicielle ?
2. Les décisions architecturales et de conception sont-elles indépendantes ?
3. Comment peut-on évaluer la qualité d'une architecture et la qualité d'une conception ?

L'architecture et la conception dans le domaine immobilier



L'architecture et la conception dans le domaine immobilier



3 Questions sur l'architecture logicielle

1. Qu'est-ce qu'une architecture logicielle ?
2. Quels sont les critères qui doivent guider un architecte logiciel dans le choix d'un style architectural particulier ?
3. Quel impact a le choix d'un style architectural sur le développement d'un système logiciel ?

3 Questions sur la conception logicielle

1. Quelle est la différence entre l'architecture et la conception logicielle ?
2. Les décisions architecturales et de conception sont-elles indépendantes ?
3. Comment peut-on évaluer la qualité d'une architecture et la qualité d'une conception ?

Qu'est-ce que l'architecture?
Qu'est-ce que la conception?



Architecture logicielle (ANSI/IEEE Std 1471-2000)

L'architecture d'un système logiciel est définie comme l'**organisation** fondamentale du système, qui s'incarne dans ses **composantes**, leurs **relations** entre elles et avec leur **environnement**, et les principes qui guident sa **conception** et son **évolution**.

Architecture logicielle (définition SEI)

L'architecture d'un système logiciel est l'ensemble des **structures** nécessaires pour **raisonner** à propos du système, qui comprennent les **éléments** logiciels, les **relations** entre eux et les **propriétés** des chaque.



Architecture logicielle (définition Sangwan, 2015)

L'architecture d'un système logiciel se préoccupe fondamentalement de l'organisation d'un système en ses constituants et leurs interrelations afin de réaliser un objectif donné.

Architecture

- Elle répond à « quoi? ».
- Au niveau modèle
 - Paquetages, composantes et relations
- Niveau de décision plus abstrait.
 - Les décisions qui doivent être prises en avance.
 - Les décisions qui sont très coûteuses à changer pendant le développement.
- Pertinente pour les objectifs opérationnels et les exigences non-fonctionnelles.
- Elle inclut:
 - Modules
 - Bases/technologies de données
 - Environnement d'exécution et de déploiement

Conception

- Elle répond à « comment? ».
 - Comment on va implémenter l'architecture.
- Éléments plus concrets:
 - Classes
 - Méthodes/Fonctions
 - Tables de données
- Pertinent pour les exigences fonctionnelles.
- Éléments qui peuvent être changés pendant le développement (la phase de maintenance)
 - Réingénierie (*refactoring*)
 - Évolution
 - Adaptation