

Objectifs de l'UE

Au terme de cette UE, les étudiants seront capables de :

- Utiliser les paradigmes de la programmation par composants et par services au travers des conteneurs logiciels (contraintes de la conteneurisation des services).
- Concevoir des applications à base de composant et de services.
- Appréhender la mise en œuvre des composants et des services dans les systèmes distribués en environnement conteneurisé.
- Maitriser les environnements d'exécution de composants et de services.
- Savoir utiliser, administrer les serveurs d'applications.
- Manipuler les intergiciels fournissant les services techniques pour le développement d'applications serveur légères et lourdes, découverte et compréhension des bus a messages, EAI, ESB.
- Concevoir des architectures orientées services, abord des architectures micro-service et Event Driven Architecture.
- Comprendre la notion de dynamisme d'application et de dynamisme architectural.
- Appréhender les problématiques de déploiements et d'adaptation des applications à chaud.
- Appréhender les problématiques de développement ServerLess (dit FaaS) en environnement conteneurisé, notion sur le Cold Start et Warm Start.
- S'appuyer sur et mettre en œuvre des paradigmes de communications synchrones et asynchrones au travers des bus a messages et des web services

Description des ECUE**Intergiciels et programmation par composants**

Principe du développement à base de composants et de services.

Ingénierie logiciel pour les composants et les services.

Phases de développement, abord des frameworks et outils de développements adapté a la programmation par composants et services.

Création de composants fiables, transactionnels, distribuables.

Intergiciels de persistance (bus MOM Kafka, EAI Mirth Connect, ESB Talend), d'instanciation et d'assemblages.

Déploiement de services distribués (Docker Swarm, modèle de partage des ressources).

Inversion de contrôle.

Principes des serveurs d'applications.

Architectures orientées services

Les principes des Architecture Orientées services, micro-services, Architecture EDA.

Création d'application dynamiques et adaptables en Java, utilisation des composants du framework SpringBoot pour la construction d'applications robustes.

Déploiement et assemblage à chaud des applications dans des conteneurs avec la solution docker, docker Swarm, portainerIO.

Conteneurisation et instanciation à la demande, notion de services en ServerLess (FaaS).

Passage à l'échelle des applications, scaabilité et distributivité des applications via des système

Notion de qualité et sécurité dans l'assemblage des applications.

Mise en œuvre dans le monde Java, FrameWork SpringBoot, composants cloud, découverte a la volée de services, etc...

Communication asynchrone et synchrone et files de messages au travers du bus a message Kafka et de web services REST.

Pré-requis

Maitrise du Java, connaitre Maven ou Graddle.

Des notions sur les principes producteurs /consommateurs, notion de processus synchrone et Asynchrone, Web Services REST.

Notions sur la sécurité des web services.

Connaissances sur les architectures tiers applicatives.

Bibliographie