



ORGANISME DE FORMATION AUX TECHNOLOGIES ET METIERS DE L'INFORMATIQUE

Formation Mach : les nouvelles architectures agiles et composables

Vers une architecture en amélioration continue sur base de Microservices, API, Cloud, Headless

N° ACTIVITÉ : 11 92 18558 92

TÉLÉPHONE : 01 85 77 07 07

E-MAIL : inscription@hubformation.com

Objectifs

- | Expliquer les nouveaux patterns de l'architecture composable et évolutive
- | Expliquer les patterns d'architectures à base de Microservices, API, Cloud, Headless
- | Identifier le nouveau rôle de l'architecte dans une organisation agile, ses enjeux d'alignement et de communication
- | Expliquer l'architecture agile au travers d'exemples concrets chez des grands comptes et pure players

Public

- | Consultants, architectes, chefs de projets, développeurs

Prérequis

- | Aucun

Programme de la formation

Enjeux de la transformation digitale

- | Volatility, Uncertainty, Complexity and Ambiguity (VUCA)
- | Risque d'Uberisation par les pure players
- | L'expérience utilisateur devenue essentielle, le « Design Thinking »
- | "Customer obsession" et omnicanalité (multiplication des devices)
- | Lean Startup, autre manière de gérer un projet
- | Interopérabilité (via API)
- | Scalabilité (cloud-ready)
- | Design for failure (scalabilité, résilience)
- | Maîtrise de sa codebase
- | La pérennisation du Legacy

Principes d'architecture agile

- | Rappel : la matrice d'architecture en 4 couches
- | Patterns classiques du Système d'Information : Royaume-Émissaire, Échanges, Référentiels, SSO, ...
- | Patterns challengés : Architectures multi-tiers et Server Side Rendering (SSR)
- | Les frameworks de développement
- | Assurer le Time-to-market (Cloud first, Software second, Dev third)
- | Mode produit : l'architecture est en évolution constante, recomposition, refactoring
- | Le découplage (SOA, API, Headless)
- | La composition, évolutivité, flexibilité (pattern Lego)
- | L'Atomic Design
- | Les principes DRY (Don't Repeat Yourself), KISS (Keep It Simple, Stupid), YAGNI (You Ain't Gonna Need It)
- | Le Software Craftmanship et GenAI

Les nouveaux fournisseurs de technologie

- | Déclin des fournisseurs de serveurs (Sun, HP, IBM..)
- | Déclin des middleware complexes (BEA, Sun, IBM, Oracle...)
- | L'open source et le rôle centrale GitHub

Référence	MAC24
Durée	2 jours (14h)
Tarif	2 197 €HT

SESSIONS PROGRAMMÉES

PARIS

du 5 au 6 décembre 2024

[VOIR TOUTES LES DATES](#)

| Montée en puissance des outils des GAFAM (Kubernetes, React, Flutter...)

Les microservice

- | MonolithFirst : du monolithe aux micro-services, sans créer d'entropie
- | Modulith versus Monolith
- | Granularité, composition, orchestration
- | Découplage technique : spécialisation, liberté technologique
- | Découplage organisationnel : feature teams autonomes
- | Domain Driven Design (DDD)
- | Data management et persistance indépendante (persistance polygotte, théorème de CAP)
- | Command and Query Responsibility Segregation (CQRS)
- | Continuous Integration/Delivery (CI/CD) par service
- | Scalabilité et résilience : auto-scaling et self-healing avec Kubernetes

API first

- | Les principes et modèles d'API
- | API management (authentification, traçabilité, disponibilité)
- | Technologie Restless plutôt que SOAP
- | API as a Product (Stripe, YouSign, DeepL, etc.)
- | GraphQL
- | Backend for Frontend (BFF)
- | Des API pour pérenniser le Legacy

L'architecture Cloud-native

- | Scalabilité by design (hardware, software, réseau, systèmes de cache)
- | Services managés en best of breed
- | Distribution des traitements (load balancing, containers, machines virtuelles, FaaS)
- | Architecture synchrone versus event driven
- | Asynchronisme en "just enough"
- | Distribution du stockage (objets, sql, nosql, no-update)
- | Multi-tenancy et feature-flipping
- | Infrastructure as code et test driven architecture
- | Les outils des hyperscalers (AWS, Azure, GCP)
- | Exposition au risque Cyber (exploits, Bug Bounty)
- | Sécurité en contexte Cloud : isolation, zero trust, authentification multifacteur
- | Un Cloud sous contrainte de coûts, vendor lock-in, souveraineté

Architecture Headless

- | User Experience/Interface (UX/UI), Design Thinking, Design System
- | Pattern mobile first
- | Omnicanalité et reprise de sessions entre devices
- | Contenu marketing en Static Site Generation (SSG)
- | Contenu applicatif en Single page application (SPA)
- | WebApps : responsive (HTML5) et PWA (Progressive Web App)
- | Apps : natives (iOS, Android) et hybrides (React native, Flutter, Electron)
- | Frameworks web (vue.js, React, Angular)
- | Frameworks natifs (Dart)
- | Sécurité avec les JSON Web Token (JWT)
- | Obsolescence rapide des technologies Front

Impact organisationnel des architectures agiles

- | Utiliser les patterns des pure players chez les grands comptes
- | Collaboration architecte et tech lead
- | Architecture Decision Record (ADR), anti-pattern du schéma directeur
- | Roadmap agile et refactoring d'architecture
- | Cartographie dynamique du SI
- | Exemple d'une startup devenue scale up

Méthode pédagogique

Chaque participant travaille sur un poste informatique qui lui est dédié. Un support de cours lui est remis soit en début soit en fin de cours. La théorie est complétée par des cas pratiques ou exercices corrigés et discutés avec le formateur. Le formateur projette une présentation pour animer la formation et reste disponible pour répondre à toutes les questions.

Méthode d'évaluation

Tout au long de la formation, les exercices et mises en situation permettent de valider et contrôler les acquis du stagiaire. En fin de formation, le stagiaire complète un QCM d'auto-évaluation.

Accessibilité



Les sessions de formation se déroulent sur des sites différents selon les villes ou les dates, merci de nous contacter pour vérifier l'accessibilité aux personnes à mobilité réduite.

Pour tout besoin spécifique (vue, audition...), veuillez nous contacter au 01 85 77 07 07.