

# FICHE DE POSTE

## Apprenti chercheur – Apprentie chercheuse

### en Sciences du Numérique

STRUCTURE D'ACCUEIL DE L'APPRENTI.E	
Nom du laboratoire d'accueil	Laboratoire d'Informatique, Signaux et Systèmes de Sophia-Antipolis (I3S)
Nom de l'équipe de recherche d'accueil	Kairos
Prénom et Nom du maître ou de la maîtresse d'apprentissage	Julien Deantoni (en collaboration avec Gérard Rocher)
E-mail de contact du maître ou de la maîtresse d'apprentissage	Julien.DEANTONI@univ-cotedazur.fr
Localisation géographique de l'équipe de recherche d'accueil et du bureau de l'apprenti.e	INRIA Sophia Antipolis, Bâtiment Lagrange

ENCADREMENT DE L'APPRENTI.E
<p><b>Modalités d'encadrement de l'apprenti.e pendant la durée du contrat</b></p> <p>L'apprenti.e sera encadré.e par un maître d'apprentissage chercheur au sein de l'équipe Kairos, assurant un suivi hebdomadaire des avancées techniques et scientifiques. Des réunions d'équipe régulières permettront de confronter les travaux à un regard collectif et d'ajuster les orientations en fonction des résultats obtenus. Un point mensuel formel sera organisé pour évaluer la progression par rapport aux objectifs définis, identifier les éventuels blocages et planifier les étapes suivantes.</p> <p>L'apprenti.e sera également encouragé.e à participer aux séminaires et journées scientifiques du laboratoire, afin de développer sa posture de chercheur.se et d'élargir sa culture scientifique au-delà du périmètre strict du sujet. En lien avec l'école d'ingénieur, un tuteur pédagogique assurera le suivi académique et le lien entre les exigences de la formation et les activités de recherche conduites en laboratoire.</p> <p><b>Niveau de connaissance de la formation suivie par l'apprenti.e</b></p> <p>L'apprenti.e est étudiant.e en cinquième année d'école d'ingénieur, spécialité</p>

informatique. Il ou elle dispose de bases solides en algorithmique, réseaux et architecture logicielle, ainsi que d'une première culture en apprentissage automatique et en développement logiciel (génie logiciel, tests, intégration continue). Le sujet s'appuie sur ces acquis tout en les étendant vers des domaines plus spécialisés (informatique distribuée, systèmes cyber-physiques, IA embarquée) que l'apprenti.e approfondira progressivement dans le cadre de ses travaux de recherche, en lien avec les enseignements de sa dernière année de formation.

### SUJET DE RECHERCHE (1 page maximum)

#### **Vers des jumeaux numériques distribués et hiérarchiques : architectures context-aware et gestion de modèles à toutes les strates du continuum edge-to-cloud**

##### **Résumé du sujet**

Les jumeaux numériques (Digital Twins, DT) ont émergé comme répliques virtuelles de systèmes physiques destinées à la simulation et à la prédiction, dans des architectures souvent déployées dans le Cloud. L'évolution des usages vers la supervision active et le contrôle-commande en temps réel impose aujourd'hui de repenser cette centralisation. En effet, les architectures centralisées peinent à intégrer l'hétérogénéité des données, à respecter les contraintes de latence du contrôle temps-réel, et à capturer les dynamiques locales fines de systèmes complexes.

Ce sujet propose de faire évoluer le DT vers une architecture *distribuée* et *hiérarchique*, déployée à toutes les strates du continuum capteurs/actionneurs, Edge, Fog, Cloud. Cette décomposition s'inspire du passage des applications monolithiques aux microservices : de même que le Domain-Driven Design (DDD) identifie des *bounded contexts* pour découper un système logiciel, l'enjeu ici est de délimiter des sous-modèles cohérents, couplés à leur contexte opérationnel local, et déployables au plus proche des processus physiques représentés.

Les travaux s'appuieront sur la maquette fischertechnik Training Factory Industry 4.0 – 24V comme physical twin de référence, instanciant nativement les strates Edge/Fog/Cloud via PLC, gateway IoT et communications synchrones et asynchrones.

##### **État de l'art**

Les travaux fondateurs de Grieves (2002) et la formalisation par Tao et al. (2018) ont établi le DT comme un système biparti (physical twin et digital twin). Les architectures actuelles, largement centralisées dans le Cloud, s'appuient sur des plateformes industrielles (e.g. Azure Digital Twins, AWS IoT TwinMaker, Siemens MindSphere). L'évolution vers la distribution est documentée dans plusieurs directions complémentaires :

- **Continuum edge-fog-cloud** : les travaux autour du fog computing (Bonomi et al., 2012) et des architectures hiérarchiques IIoT posent les bases d'un déploiement multi-strates, sans pour autant être appliqués aux DT.
- **DT distribués et fédérés** : des propositions récentes (Cimino et al., 2019 ; Lim et al., 2020) esquissent des architectures de DT composites, sans traiter la question du partitionnement des modèles ni de leur synchronisation sans centralisation des données.
- **Federated Learning (FL) appliqué aux DT** : les approches FL (McMahan et al., 2017) offrent un cadre prometteur pour la synchronisation de modèles distribués, mais leur intégration dans des architectures DT hiérarchiques reste un problème ouvert.
- **Interopérabilité sémantique** : les ontologies SSN/SOSA (W3C) et SAREF (ETSI) fournissent des briques pour l'harmonisation des flux hétérogènes, sans couvrir la

gestion dynamique et cohérente des modèles.

### Verrous scientifiques

- Définition d'une architecture de référence pour le DT distribué, permettant de délimiter des sous-modèles cohérents à partir d'un jumeau numérique centralisé existant. Le parallèle DDD/microservices trouve ici ses limites: les frontières de contextes opérationnels sur des modèles hétérogènes ne se déduisent pas des seules frontières métier logicielles.
- Partitionnement optimal des modèles en fonction des capacités hétérogènes des nœuds de calcul distribués.
- Maintien de la cohérence globale d'un DT distribué en présence de connectivité intermittente ou de nœuds défaillants.
- Interopérabilité sémantique entre des modèles de nature différente (physiques, statistiques, basés sur l'apprentissage, hybrides).
- Passage à l'échelle de l'architecture face à un nombre croissant de physical twins supervisés.

## ACTIVITES CONFIEES A L'APPRENTIE

L'apprenti chercheur- apprentie chercheuse approfondira les savoirs et savoir-faire suivants :

- **Compréhension du contexte de recherche et méthodologie scientifique.** Appréhender les enjeux scientifiques et industriels des systèmes cyber-physiques et des jumeaux numériques, situer les travaux dans le champ de la recherche en informatique distribuée et en IA embarquée, et s'initier à l'éthique de la recherche et à l'intégrité scientifique.
- **Acquisition de connaissances, analyse de l'état de l'art et production scientifique.** Conduire une revue de littérature sur les architectures de jumeaux numériques distribués, les approches de décomposition inspirées du Domain-Driven Design, et les mécanismes de synchronisation de modèles (federated learning, transfer learning). Compiler et maintenir une bibliographie structurée, rédiger des livrables scientifiques (rapport intermédiaire, article de conférence), et présenter les avancées devant un public académique et industriel.
- **Conception, implémentation et évaluation d'une architecture distribuée.** Participer à la définition de l'architecture de référence du DT distribué, implémenter des composants logiciels déployables sur le continuum edge-fog-cloud, et évaluer expérimentalement les solutions proposées sur la maquette fischertechnik Training Factory Industry 4.0 - 24V. Les expériences porteront notamment sur l'ingestion de flux hétérogènes, le partitionnement des modèles et la synchronisation distribuée.
- **Environnement scientifique, collaborations et interactions académiques-industrie.** S'intégrer dans une équipe de recherche, participer aux réunions de suivi et aux séminaires internes, et interagir avec les partenaires industriels le cas échéant. Prendre connaissance des dynamiques de financement de la recherche (programmes Horizon Europe, France 2030) et des modes de valorisation des résultats (publications, dépôts logiciels, transfert technologique).

## COMPETENCES REQUISES POUR REALISER LES ACTIVITES

### Compétences techniques indispensables

- Maîtrise de la programmation orientée objet et des langages courants en

développement logiciel et systèmes (Python, Java ou C++) ainsi qu'une expérience en développement de systèmes distribués (API, protocoles de communication, gestion de services).

- Connaissance des architectures logicielles et des principes de conception modulaire (séparation des responsabilités, couplage faible), constituant le socle nécessaire à la compréhension et à l'application des principes du Domain-Driven Design dans le contexte du sujet.
- Bases en apprentissage automatique et en traitement de données : manipulation de données structurées et temporelles, entraînement et évaluation de modèles, utilisation de frameworks standards (scikit-learn, PyTorch ou équivalent).
- Compétences en réseaux et protocoles de communication, avec idéalement une première exposition aux protocoles IoT industriels (MQTT) ou à défaut une capacité démontrée à s'approprier rapidement de nouveaux standards de communication.

#### Compétences transversales

- Capacité à lire, analyser et synthétiser des articles scientifiques en anglais.
- Autonomie dans la conduite d'un travail de recherche et d'expérimentation, rigueur dans la documentation des résultats et des choix de conception.
- Aptitude à travailler en équipe et à communiquer des résultats techniques.

#### Compétences souhaitables

- Première expérience avec des environnements de conteneurisation et d'orchestration (Docker, Kubernetes) ou des architectures microservices.
- Sensibilité aux problématiques de systèmes temps-réel ou embarqués.

### PERSPECTIVES APRES LA PERIODE D'APPRENTISSAGE

L'architecture de référence développée pendant cette période d'alternance constituera un socle de recherche pour l'équipe et le candidat, notamment dans une perspective de poursuite en thèse, en ouvrant les problématiques scientifiques suivantes:

- **Gestion de la fidélité des modèles et des exigences fonctionnelles et non-fonctionnelles.** À chaque strate, les modèles pourront être calibrés selon un compromis explicite entre fidélité de représentation (model fidelity) et contraintes fonctionnelles et non-fonctionnelles: consommation énergétique, empreinte mémoire, temps de réponse, résilience aux pannes. Des politiques de migration et de reconfiguration dynamique permettront d'adapter l'architecture aux évolutions du contexte opérationnel sans interruption de service.
- **Détection et explication des dérives opérationnelles (reality gap).** La distribution fine des modèles au plus proche de leurs processus cibles offrira une sensibilité accrue aux écarts locaux entre prédictions et observations. Des méthodes d'explication (XAI) associées à ces modèles permettront d'identifier et de caractériser les causes profondes des dérives (usure, changement de régime, défaillance de capteur) et d'en informer les boucles d'adaptation du DT de manière intelligible pour les opérateurs.
- **Sélection dynamique de modèles et adaptation au contexte.** L'architecture distribuée ainsi constituée ouvrira la voie à des mécanismes de commutation dynamique entre les modèles du catalogue : selon le contexte opérationnel courant (précision requise, ressources computationnelles disponibles, criticité de la décision) le système pourra mobiliser à la volée le modèle le plus approprié parmi des modèles de niveaux d'abstraction et de fidélité variables, activant par exemple un modèle physique détaillé lorsque la situation l'exige, ou se rabattant sur un modèle plus léger en cas de saturation des ressources. La constitution de ce catalogue pourra s'appuyer sur des

approches de transfer learning, permettant de spécialiser des modèles génériques (entraînés à l'échelle globale du système) aux contextes locaux de chaque strate ou équipement, sans repartir de zéro à chaque déploiement.

À noter enfin que ces recherches s'inscrivent dans le programme national d'ingénierie des jumeaux numériques ([www.edtlab.fr](http://www.edtlab.fr)) à laquelle l'équipe participe.