



Master informatique Accréditation 2024-2029

Structure générale

Master Informatique (environ 900H pour 120 ECTS)

Parcours ARIAS

Applications Réparties,
Intelligence Artificielle et Sécurité

Parcours GPEX

Graduate Program of Excellence Model

**Voie
professionnelle**

**Voie recherche
(sélective)**

- 70-80 % du parcours ARIAS
- 20-30 % modules multi-disciplinaires
- 20-30 % modules spécifiques

Stage obligatoire S4 (4-6 mois)

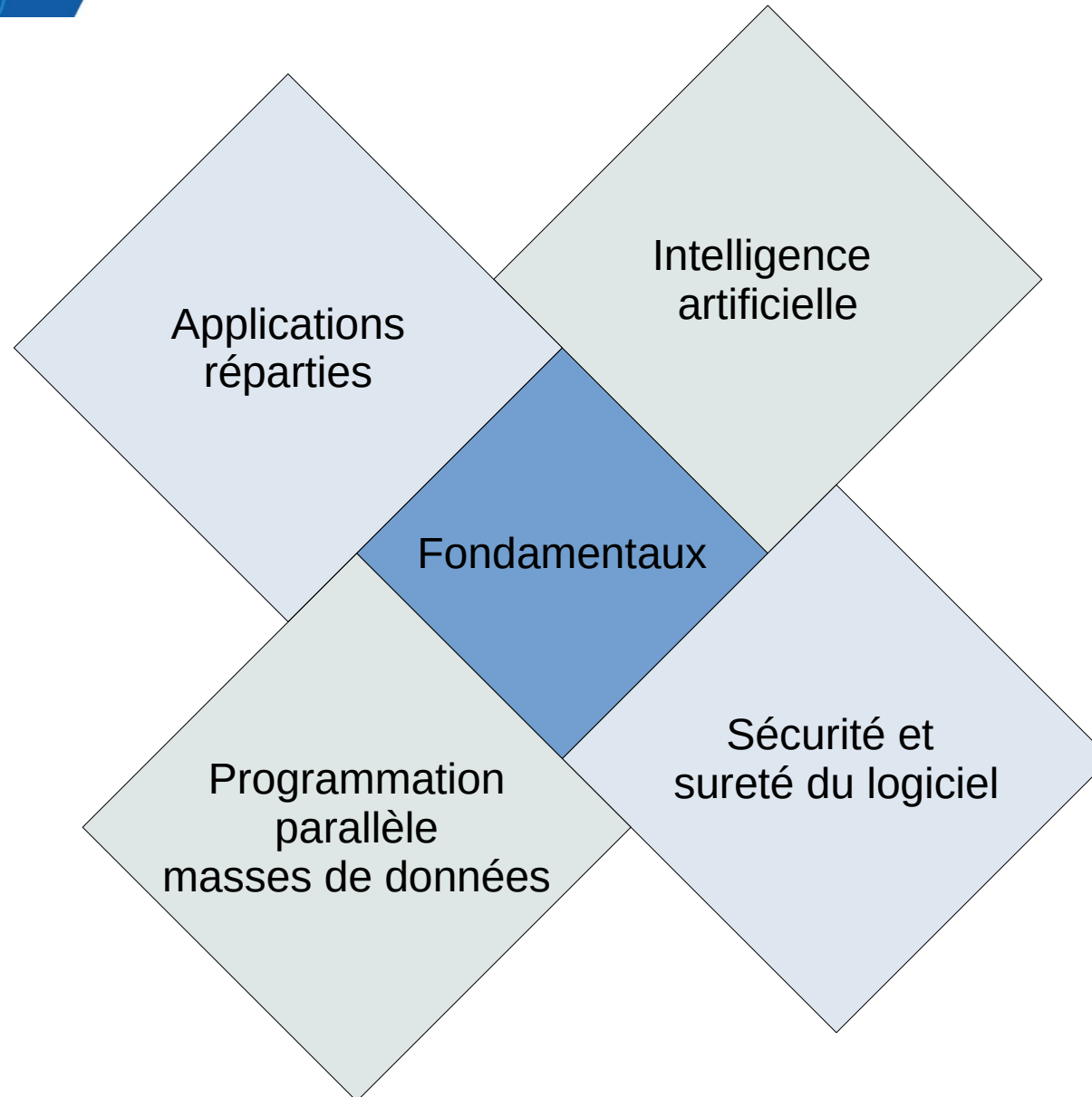
stages optionnels

immersion en laboratoire
(substitution de modules)

- bourse en Master 1
- contrat d'apprentissage en Master 2
 - immersion progressive en laboratoire

possibilité d'effectuer des semestres à l'international (ERASMUS)

Thématiques



Applications réparties

Architectures applicatives réparties

Approfondissement de la notion de composants logiciels : déclaration et injection de composants, interfaçage vers des données relationnelles via un ORM, mise en place d'un contrôleur (MVC et REST). Mise en place de tests. Déploiement dans un environnement microservices. Compréhension des principes techniques et conceptuels sous-jacents.

- Java
- Spring
- JPA
- ORM
- micro services

Applications réparties

Web services

« Un service web est un protocole d'interface informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués » [Wikipedia]

- Qu'est ce qu'un protocole web?
- Qu'est-ce que je peux échanger et comment ?

- Services Restfull et SOAP : principes et mise en oeuvre en java
- Pré-requis : Java : langage et IDEs standards
- Objectifs : Etre capable de développer et d'utiliser des web services dans le respect des conventions usuelles

Applications réparties

Développement multiplateforme

Le développement multiplateforme (ou cross-platform en anglais) fait référence à la conception et à la création d'applications logicielles qui peuvent être déployées et exécutées sur plusieurs plates-formes ou systèmes d'exploitation, tels que différents types d'appareils, de systèmes d'exploitation, ou même de navigateurs web. L'objectif principal est de maximiser le partage de code et de ressources, ce qui permet de réduire les efforts de développement, de maintenance et de coûts associés à la création d'applications spécifiques à chaque plateforme.

- Multiplateforme,
- Cross-platform,
- Android, iOS,
- MacOS, Windows, Linux,
- Flutter, Xamarin

Applications réparties

Développement nomade natif

Le développement nomade natif se réfère à la création d'applications mobiles spécifiquement conçues pour une plateforme particulière, utilisant les langages et les outils de développement natifs de cette plateforme. Par exemple, pour iOS, on utilise Swift ou SwiftUI avec Xcode, tandis que pour Android, on utilise Java ou Kotlin avec Android Studio. Cette approche offre un accès direct aux fonctionnalités et performances spécifiques à chaque plateforme, garantissant une expérience utilisateur optimale. Le développement nomade natif est souvent privilégié pour des applications nécessitant des performances élevées, une intégration profonde avec le système d'exploitation, ou des fonctionnalités spécifiques à une plateforme.

- Android,
- Kotlin,
- iOS,
- Swift,
- SwiftUI

Fondements

Modélisation, graphes et algorithmes

Cet enseignement complète la culture algorithmique acquise en licence. On y étudie des problèmes non triviaux, avec des approches pointues.

- résolution de problèmes par des approches algorithmiques
- modélisation de problèmes par des graphes
- problèmes de graphes (planarité, coloration, cycles eulériens et hamiltoniens, flots)
- algorithmes d'approximation
- algorithmes probabilistes

Calculabilité et complexité

En utilisant la culture acquise au cours des années de licence, on cherchera à préciser des fondements de l'informatique : "calculer", c'est quoi ? Peut-on en faire plus dans un langage de programmation (ou modèle de calcul) que dans un autre ? Quelles questions ont une réponse qui peut être déterminée par un ordinateur ? Toutes interrogations qui nous porteront à réfléchir à ce qu'est l'informatique !

- Problème décidable/indécidable
- Langage récursif, récursivement énumérable, non récursif
- Machine de Turing, arrêt, réduction
- Complexité d'un problème, non déterminisme
- Classes P et NP, complétude

Programmation quantique

La “programmation quantique” repose sur la manipulation de bits quantiques (qubits) à travers des portes et circuits quantiques. Concevoir un algorithme équivaut à élaborer le circuit logique pour résoudre le problème, en prenant en compte les caractéristiques propres aux qubits, telles la superposition (leur capacité d’être une combinaison de 0 et 1), les interférences et l’intrication. Cette approche permet de proposer des algorithmes surprenants, parfois exponentiellement plus efficaces que les algorithmes classiques. Nous aborderons également les aspects plus culturels, de l’avantage quantique espéré, aux limites des ordinateurs quantiques actuels.

- comparatif calcul classique/calcul quantique
- avantage quantique et limites des ordinateurs quantiques actuels
- qubit, superposition, intrication, interférence
- portes et circuits quantiques
- premiers algorithmes quantiques (mise en œuvre dans une boîte à outil open source)
- cryptographie quantique
- algorithme de Grover, algorithme de Shor
- intrication et conséquences

Fondements

Description formelle de langages

Ce module est consacré à l'analyse sémantique d'un programme. Partant de l'arbre syntaxique abstrait produit par l'analyse syntaxique, cette étape essentielle précède la production de code qui vise à obtenir une version exécutable du programme analysé. Elle travaille à décorer l'arbre abstrait par des informations sémantiques. On y aborde les définitions dirigées par la syntaxe qui sont des spécifications de haut niveau pour les traductions ainsi que les schémas de traduction qui permettent quant à eux de faire apparaître certains détails d'implantation. Une articulation est ensuite faite pour lier cette analyse sémantique à l'étape de production de code, qui fait partie du cycle d'un compilateur.

- syntaxe abstraite,
- sémantique statique et dynamique,
- grammaire attribuée,
- environnement d'exécution

Fondements

Compilation

Le module de compilation vise d'abord à approfondir sa connaissance de l'ordinateur et des langages de programmation. On cherchera à comprendre tout au long du semestre comment un programme écrit dans un langage lisible et concevable par un humain peut être exécuté physiquement par un ordinateur. Pour ce faire, il faudra explorer ce qui constitue un langage, et ce qui constitue une machine d'exécution, ainsi que toutes les étapes intermédiaires.

- Langages de programmation
- Assembleur,
- Syntaxe abstraite,
- Règles sémantique,
- Gestion de la mémoire,
- Traduction de programmes.

Intelligence artificielle

Intelligence artificielle

L'objectif général de ce module est de permettre aux étudiants d'acquérir les notions de bases de l'intelligence artificielle. Une présentation de l'historique de l'IA sera donnée, de ses débuts dans les années 50 à nos jours, avec des différentes facettes de l'IA. La notion d'agents intelligents sera introduite. La modélisation de problème et la recherche de solutions seront abordées, avec différentes techniques de recherche de solution (recherche dans un graphe d'états, recherche informée, recherche locale, recherche de solution dans un jeu à deux joueurs, etc.). Une introduction à l'apprentissage automatique sera présentée avec des méthodes de l'apprentissage supervisé et l'apprentissage non supervisé.

- Agent intelligent
- modélisation et résolution de problèmes
- apprentissage automatique

Intelligence artificielle

Programmation par contraintes

La programmation par contraintes est un paradigme de programmation déclarative permettant de résoudre efficacement des problèmes combinatoires de grande taille. Les problèmes sont modélisés sous forme d'un problème CSP (Constraint Satisfaction Problem) à l'aide des variables de décision et des contraintes, ou d'un problème COP (Constraint Optimization Problem) avec une fonction objectif. Un solveur de programmation par contraintes cherche une solution, toutes les solutions ou la meilleure solution en exploitant les techniques d'inférence et de recherche. Ce module présente les différents aspects de la programmation par contraintes, de la modélisation à la résolution de problèmes CSP/COP.

- Contraintes
- problème de satisfaction de contraintes CSP
- problème d'optimisation sous contraintes COP
- solveur de programmation par contraintes
- propagation de contraintes
- stratégie de recherche

Intelligence artificielle

Machine learning

Ce module offre une introduction au Machine Learning, explorant les principaux concepts, techniques et applications dans le domaine. Les étudiants acquerront des compétences théoriques et pratiques pour développer et mettre en œuvre des modèles d'apprentissage automatique.

- Machine Learning
- Apprentissage automatique
- classification supervisée et non supervisée,
- régression
- réseaux de neurones
- algorithmes d'apprentissage



Intelligence artificielle

Machine learning, données séquentielles

Ce module explore l'application de l'apprentissage automatique aux données séquentielles, avec un accent particulier sur les données textuelles et les flux de données. Les étudiants découvriront comment modéliser et analyser des séquences, ouvrant la voie à la compréhension des applications telles que le traitement du langage naturel et l'analyse en temps réel des flux de données.

- Machine Learning
- Apprentissage automatique
- Données séquentielles
- Traitement du langage naturel
- Flux de données (Data streams)
- apprentissage en ligne.

Intelligence artificielle

Deep learning

Ce module fournit un aperçu complet des réseaux de neurones et des techniques d'apprentissage profond. Les étudiants posséderont les compétences nécessaires pour concevoir et entraîner des réseaux de neurones convolutifs (ConvNets) et mettre en œuvre des architectures modernes. Ils seront capables d'appliquer des techniques essentiels tels que les fonctions d'activation et la normalisation des lots, le dropout et l'augmentation des données. Les réseaux de neurones récurrents (RNN), les LSTM, les mécanismes d'attention et les transformers seront aussi présentés. En mettant l'accent sur les applications pratiques, les étudiants acquerront des compétences les tâches clés de vision par ordinateur telles que la détection, la segmentation à l'aide de modèles Faster R-CNN, YOLO et U-Net.

- Deep Learning
- Apprentissage profond
- apprentissage automatique
- réseaux de neurones.

Intelligence artificielle

Web mining et réseaux sociaux

L'évolution rapide du paysage numérique a conduit à une prolifération de données sur le web, offrant un terrain fertile pour l'exploration et l'extraction de connaissances. Ce module se plongera dans quatre dimensions essentielles du Data Mining appliqué aux applications web. Nous explorerons la structure des réseaux sociaux, plongerons dans le monde sémantique du web, comprendrons les mécanismes des systèmes de recommandation et de l'analyse de sentiments, et enfin, nous plongerons dans l'univers des données ouvertes et des bases de données NoSQL.

- Web mining structurel
- Web sémantiques
- Systèmes de recommandation
- Analyse de sentiments
- NoSQL

Sécurité et sûreté du logiciel

Test et validation du logiciel

L'objectif de ce module est de sensibiliser les étudiants à l'utilité des méthodes formelles pour effectuer des tests de logiciel tant sur le code source que par rapport au comportement du logiciel. Le test est un processus qui vise à établir qu'un système vérifie les propriétés exigées par sa spécification, ou à détecter des différences entre les résultats engendrés par le système et ceux attendus par la spécification

- test fonctionnel,
- test non-fonctionnel,
- scénario,
- jeux de données,
- spécifications algébriques,
- méthodes formelles

Vérification de programmes

Ce cours présente les notions de base de la preuve déductive de programme et leur mise en pratique à l'aide d'outils automatique ou semi-automatique. Cette technique permet de vérifier formellement la correction d'une programme vis à vis d'une spécification. Les concepts abordés seront mis en pratique à l'aide des outils Why3 et Coq.

- Méthodes formelles
- Fiabilité du logiciel
- Vérification de programme
- Preuve déductive
- Logique de Hoare
- Solveurs SMT
- Assistants de preuves.

Sécurité des applications

Le module se concentre sur les vulnérabilités mémoire des applications. Plusieurs types de vulnérabilités sont décrites, ainsi que les moyens qui sont utilisés en pratique pour exploiter ces vulnérabilités pour détourner des données ou prendre le contrôle de la machine qui exécute l'application ciblée. Des protections contre ces attaques sont également présentés, ainsi que des méthodes formelles permettant de garantir l'absence de vulnérabilités mémoire. Une dernière partie est un aperçu plus général de la littérature scientifique récente dans le domaine de la sécurité des applications.

- Vulnérabilités des programmes
- exploitation de vulnérabilités
- Défenses
- buffer overflow
- return oriented programming
- méthodes formelles

Concepts avancés des langages de programmation

Ce cours se concentre sur les principes et la mise en œuvre des concepts avancés des langages de programmation. La thématique principale est celle du typage, essentiellement statique, de langages de programmation récents avec pour objectifs de permettre d'assurer des propriétés de sécurité mémoire (par exemple avec Rust), d'exprimer et d'assurer des propriétés sur les programmes (par exemple avec Idris), de gérer la modularité des programmes.

- Types
- possession et emprunt
- types dépendants
- modularité

Cryptographie et sécurité

Le but de ce module est de sensibiliser les étudiants sur les notions de sécurité par la cryptographie. Au-delà de la sensibilisation, nous allons pratiquer et exploiter quelques biais d'utilisations des technologies actuelles comme historiques. Le but est également de comprendre en quoi les technologies cryptographiques nous permettent de garantir des propriétés de sécurité comme la confidentialité, l'intégrité ou encore l'authentification par les mécanismes de signatures ou de calculs d'empreintes numériques.

- Chiffrements symétriques
- chiffrements asymétriques
- Stéganographie
- chiffrements par blocs
- Signatures
- intégrité

Sécurité des données et des réseaux

Politiques et bonnes pratiques de sécurité ; sécurité des systèmes d'exploitation ; sécurité réseau ; authentification ; sécurisation des serveurs d'applications.

- Concepts de base sur l'administration Unix / Windows
- Modèles de sécurité
- Sécurité d'un système Linux
- Sécurité des systèmes Windows NT
- Systèmes de détection d'intrusion

Programmation parallèle

Dans de nombreux domaines scientifiques, l'étude des phénomènes étudiés nécessite soit du calcul intensif pour réaliser des simulations soit des traitements de gros volumes de données pour leur analyse. Être capable d'exploiter des super calculateurs ou des machines multi-processeurs est indispensable dans ce contexte pour développer des codes performants adaptés à ces architectures. Pour cela il est nécessaire d'avoir des connaissances sur les différentes machines parallèles, comment programmer des codes exécutables sur ces machines mais également comment paralléliser des algorithmes efficacement.

- Optimisation sur un processeur : Vectorisation (SIMD SSE/AVX)
 - Architecture à mémoire distribuée :
Paradigme de programmation par passage de messages (bibliothèque MPI)
 - Architecture à mémoire partagée :
Paradigme de programmation par directives (OpenMP)

Parallélisme, masses de données

Programmation haute performance

- *Les cartes graphiques* sont désormais utilisées pour accélérer de nombreux algorithmes. En particulier les algorithmes de Deep Learning ont les propriétés nécessaires à des parallélisations efficaces sur GPU. Ce module apporte les connaissances permettant de paralléliser un algorithme pour une architecture GPU.
- *La visualisation scientifique* est utilisée dans de nombreux domaines scientifiques. Elle a pour objectif de transformer des données en des images adaptées aux besoins de compréhension des utilisateurs.
- La programmation haute performance est utilisée dans de nombreux domaines pour répondre à des besoins d'utilisateurs qui ne sont pas des informaticiens. Ils existent de nombreuses approches qui permettent soit de démocratiser la programmation parallèle soit de traiter efficacement un type d'applications.

- API CUDA et OpenCL
 - API VTK
- Démocratisation de la programmation parallèle

Big data

Ce module aborde les aspects fondamentaux et technologiques des systèmes modernes de traitement de données massives (mégadonnées, ou Big data). Ces données sont habituellement caractérisées selon différentes dimensions principales que sont : Volume, Variété, Véracité et Vélocité. Le module est découpé en différentes parties, correspondant à l'étude des différentes dimensions caractérisant les big data.

- systèmes polystores (PostgreSQL, ...)
- contrôle de la qualité des données, de l'analyse (data profiling)
- transformation et au nettoyage (data cleaning) des données de mauvaise qualité.
- modèles de qualité des données et approches méthodologiques de contrôle de la qualité

Anglais

Semestre 7

Ce module est une préparation et un entraînement systématique à la certification TOEIC® (niveau B2 visé en fin de Master). Il s'agira d'acquérir du vocabulaire, d'affiner ses connaissances grammaticales et syntaxiques et de développer des stratégies efficaces pour réussir le meilleur score possible au TOEIC®. Une session de test TOEIC® officielle sera proposée à l'issue de ce semestre et au moins 2 fois dans l'année universitaire.

TOEIC – certification en langues - niveau B2

Semestre 8

S8 : Ce module vise à développer les compétences en expression orale des étudiants dans un contexte professionnel. Il s'agira principalement de s'entraîner à interagir oralement entre pairs. Quelques notions théoriques sur la communication orale en situation professionnelle seront également abordées.

Communication orale – conversation - anglais professionnel

Anglais

Semestre 9

Le monde de l'informatique dans les années à venir : quels nouveaux enjeux technologiques et humains ? quelles implications pour les futurs professionnels ?

enjeux – Informatique - avenir

Semestre 10

L'entrée dans le monde du travail dans le domaine informatique : mon parcours, mon avenir.

insertion professionnelle - monde du travail - entrepreneuriat – inclusivité

Parcours ARIAS

Semestre 1	ECTS	CM	TD	TP
Cours communs	5	-	-	-
Anglais	2		16	
Initiation à la recherche	3	24,5		
Cours substituable	25	-	-	-
Programmation parallèle	5	15	6	14
Intelligence artificielle	4	12	14	4
Test et validation du logiciel	4	12	9	8
Description formelle de langages	4	12	10	6
Modélisation, graphes et algorithmes	4	12	18	-
Développement multiplateformes	4	12	-	18

Substitution

Pro et recherche

1 module d'ouverture

Voie Recherche

Equivalent ECTS de
2 modules du
tronc commun

Stage optionnel

Parcours ARIAS

Semestre 2	ECTS	CM	TD	TP
Cours communs	2	-	-	-
Anglais	2		16	
Cours substituable	28	-	-	-
Programmation par contraintes	4	14	8	6
Machine learning	4	16	8	6
Calculabilité et complexité	4	14	20	-
Compilation	4	14	6	10
Cryptographie et sécurité	4	14	8	8
Vérification de programmes	4	14	4	12
Web services	4	14	-	14

Substitution

Pro et recherche

1 module d'ouverture

Voie Recherche

Équivalent ECTS de
2 ou 3 modules du
tronc commun

Stage optionnel

Parcours ARIAS

Semestre 3	ECTS	CM	TD	TP
Cours communs	18	-	-	-
Anglais	2	-	16	-
DILL 1 (ou recherche avancée pour la voie recherche)	3	-	-	-
Architectures applicatives réparties	5	16	24	-
Sécurité des applications	4	14	14	-
Programmation quantique	4	10	10	10
Cours substituable	12	-	-	-
Sécurité des données et des réseaux	4	14	16	-
Concepts avancés des langages de programmation	4	16	6	10
Deep Learning	4	6	8	16
Web Mining et réseaux sociaux	4	14	16	-
Développement nomade natif	4	12	-	18

Substitution

Pro et recherche

1 module d'ouverture

Voie Recherche

Équivalent ECTS de
2 ou 3 modules du
tronc commun

Stage optionnel

Parcours ARIAS

Semestre 4 Voie pro	ECTS	CM	TD	TP
Cours communs	18	-	-	-
Anglais	2	-	16	-
DILL 2	3	-	-	-
Sécurité des applications	4	14	14	-
Cours substituable	12	-	-	-
Programmation haute performance	4	14	16	-
Big Data	4	16	6	10
Machine learning	4	6	8	16

Semestre 4 Voie recherche	ECTS
Stage	30

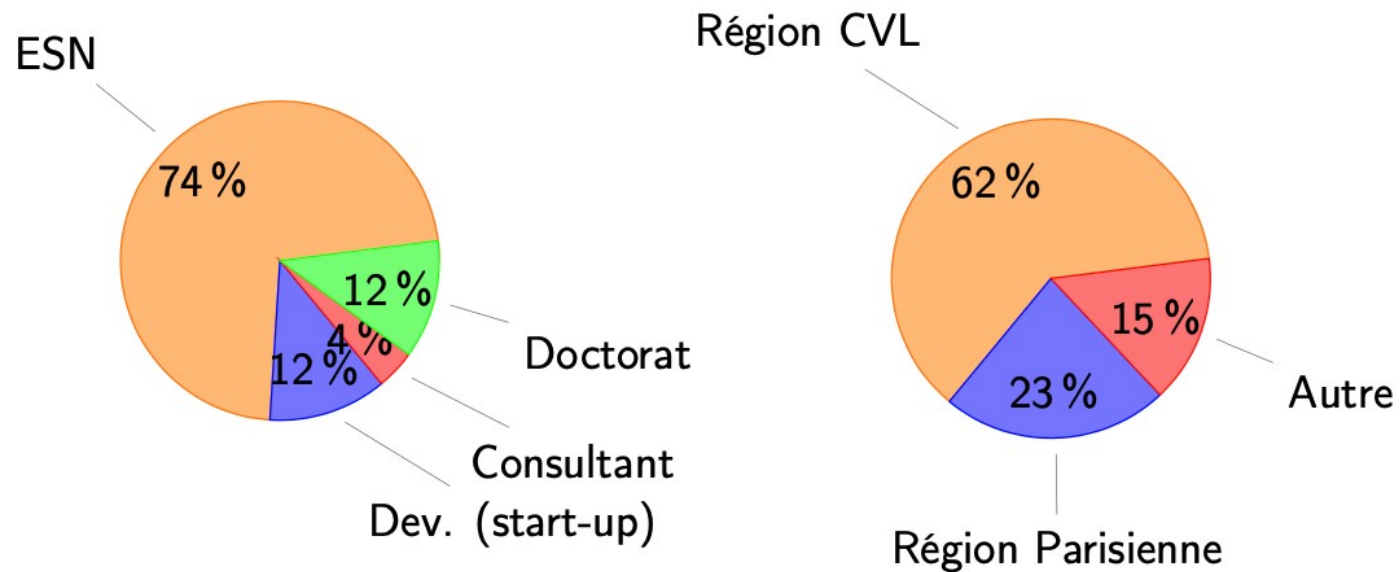


Professionalisation

- **Conférences par des professionnels extérieurs**
 - Pour votre culture
 - Permet de se faire une idée des métiers et d'échanger avec les intervenants
 - Exemples : Développement agile, DevOps
- **Interventions de professionnels extérieurs dans les modules**
 - Exemples : Développement d'applications multiplateformes, Développement nomade natif, Test et validation du logiciel, Sécurité des applications, Sécurité des données et des réseaux
- **Projet Dill**
 - Sujets Proposés par des entreprises ou des laboratoires
 - Travail en équipe multi-disciplinaires sur deux semestres en Master 2
 - Trois événements annuels

Débouchés

- Débouchés



- Salaires annuels bruts (hors doctorat)

- De 28 000 euro à 39 000 euro
- En moyenne 32500 euro

Semestres à l'étranger

- Possibilité de suivre un (ou deux) semestres à l'étranger
- Généralement au second semestre. A préparer dès maintenant
- Partenariats : Cluj en Roumanie, Varsovie en Pologne
- Contact : le service VICI (Valorisation, international et communication) de l'UFR

Diplôme universitaire (DU)

- Graduate School Orléans Numérique
- Acquisition de compétences complémentaires
 - Droits de l'informatique, physique, ...
- Matérialisées par un DU
- Deux modules de 20H sur des créneaux banalisés

La recherche

Formation adossée au LIFO
(Laboratoire d'Informatique Fondamentale d'Orléans)
Université d'Orléans – INSA CVL

5 équipes

- CA (Contraintes et apprentissage)
- GAMOC (Graphes, Algorithmes et modèles de calcul)
- LMV (Langages, Modèles et Vérification)
- PAMDA
- SDS (Sécurité des systèmes distribués, localisée à Bourges)

La recherche

CA

- Contraintes
- Apprentissage automatique
- Traitement automatique des langues

GAMOC

- Algorithmique
- Combinatoire
- complexité et calculabilité des structures discrètes

LMV

- Sémantique des langages de programmation
- Preuve déductive de programmes
- Méthodes formelles pour la sûreté

PAMDA

- Parallélisme et Calcul Haute Performance
- Bases de Données

SDS

- Contrôle d'accès et d'usage
- Modèles d'attaque (données et algorithmes)
- Approche formelles (sécurité et vie privée)
- Applications pluridisciplinaires de la sécurité

La recherche

Poursuite en doctorat

- Durée : 3 ans
- Purement académique ou CIFRE (mi-temps en entreprise)
- Recherche sur un sujet de pointe
- Publication dans des conférences internationales
- Manuscript et soutenance pour obtenir le titre de Docteur
- Eventuellement : jusqu'à 64 heures d'enseignement par an

Débouchés

- Chercheur ou enseignant chercheur
- Ingénieur de recherche
- En milieu académique ou dans la section R&D d'une entreprise

Entre les deux

- POSTDOC
 - CDD Recherche
- ATER
 - Attaché temporaire d'enseignement et de recherche

La recherche

Substitution de modules

- Immersion dans une équipe de recherche
- Equivalent ECTS de
 - Deux modules au S7
 - Deux ou trois modules au S8
 - Deux ou trois modules au S9

GPEX

- Parcours d'excellence
- Ouverture multi-disciplinaire
- Immersion en laboratoire tout au long de la formation
- Bourse et contrat d'apprentissage

Module initiation à la recherche

Conférences animées par des membres du laboratoire (4,5H par équipe)

Stage recherche au S4

4 à 6 mois dans un laboratoire de recherche ou dans la section R&D d'une entreprise

Immersion totale en laboratoire

Graduate Program of Excellence Model : « GPEx » du Projet « **minerve** »



Objectif : promouvoir la formation **par et pour la “Recherche”**, telle que pratiquée au sein des thèmes **d’excellence** de l’UO : santé, multimatériaux, géosciences, data science, Intelligence artificielle, activités physiques-rééducation, micro-nanosciences, espace, cosmétique, génomique, environnement, énergie, etc.

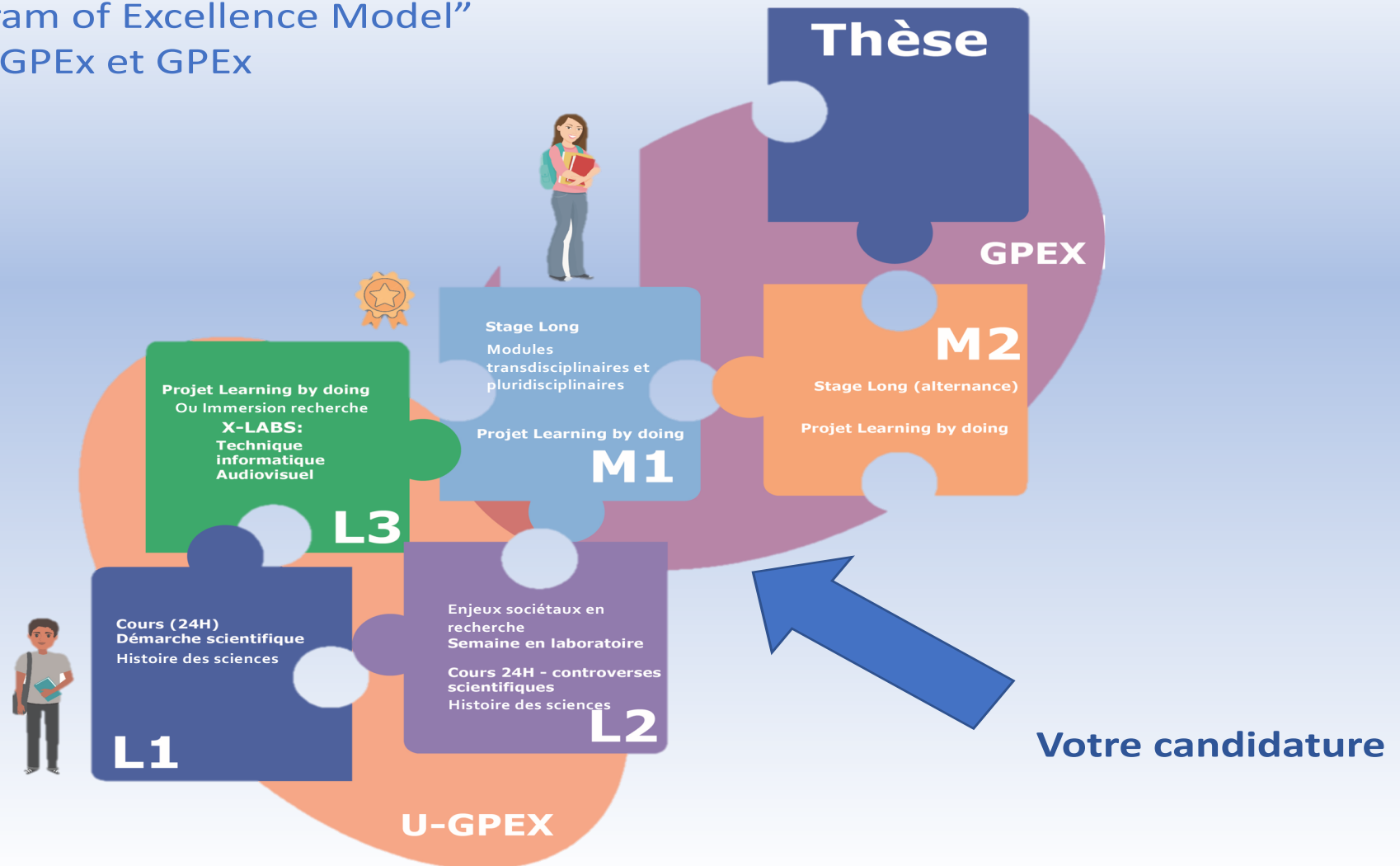
Ambition : Amener ces étudiants à cheminer vers **une carrière dans la recherche et l’innovation** en passant à terme par le **doctorat**.

Méthode : Donner l’envie/former pour : • Repousser les limites de la connaissance • Développer des technologies • Comprendre des phénomènes • Expliquer des mécanismes • Imaginer l’utilisation des découvertes • Utiliser l’interdisciplinarité pour innover.

- Une base **disciplinaire** des formations de masters
« Physique fondamentale, Chimie, SV, Informatique, Math appliquées, STAPS »
- Une **immersion progressive** dans la Recherche/Développement/Innovation
- Une forte **modularité/personnalisation** des parcours
- Une approche « **apprendre en faisant** » interdisciplinaire
via des plateformes expérimentales ou numériques (FabLab, SimLab, WebLab...)
- Un **compagnonnage** scientifique et pédagogique
- Une **valorisation** continue de la progression des acquis

- Une appropriation des méthodologies **numériques**
- Une ouverture vers les sciences de l’homme et de la société-**SHS**
- Une sensibilisation aux **enjeux sociétaux** et à l’**entreprenariat**
- Une portée **internationale** active

“Graduate Program of Excellence Model” Parcours Under-GPEX et GPEX



Graduate Program of Excellence « GPEX »



Les formations qui ouvrent un parcours GPEX dès sept 2024 :

à l'UFR ST

- Master Chimie (5 parcours partiellement recherche)
- Master Physique fondamentale et applications (2 parcours recherche)
- Master Sciences du vivant (1 parcours recherche)
- Master Mathématiques (1 parcours partiel. Recherche)
- Master Informatique (1 parcours partiel. recherche)
- Master STAPS (1 parcours recherche: EOPS)

à l'OSUC

- Master Sciences de la Terre (3 parcours partiel. recherche)

à Polytech

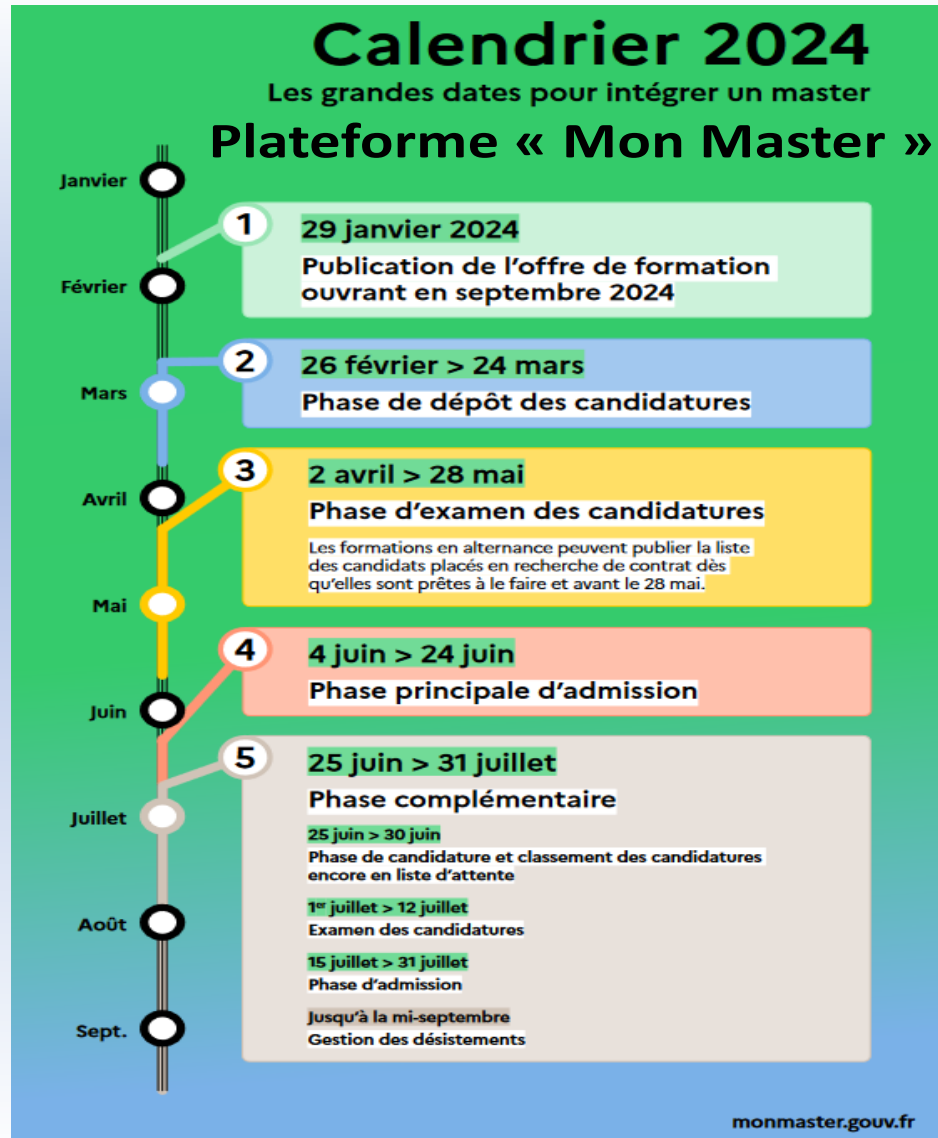
- Master Mécanique (2 parcours partiel. Recherche)

Pourquoi s'inscrire dans un Master GPEX « minerve »

- Parcours modulable et personnalisable
- Inter, pluri et transdisciplinarité
- Immersion continue dans la recherche
- Thématiques porteuses de l'université d'Orléans
 - Bourses d'excellence en M1
 - Contrat d'apprentissage en laboratoire en M2
 - Opportunités de thèse de doctorat

-Sélection des candidatures GPEX après candidature sur « Mon Master »

Graduate Program of Excellence « GPEX »



-Candidatures GPEX à l'UO : ecandidat

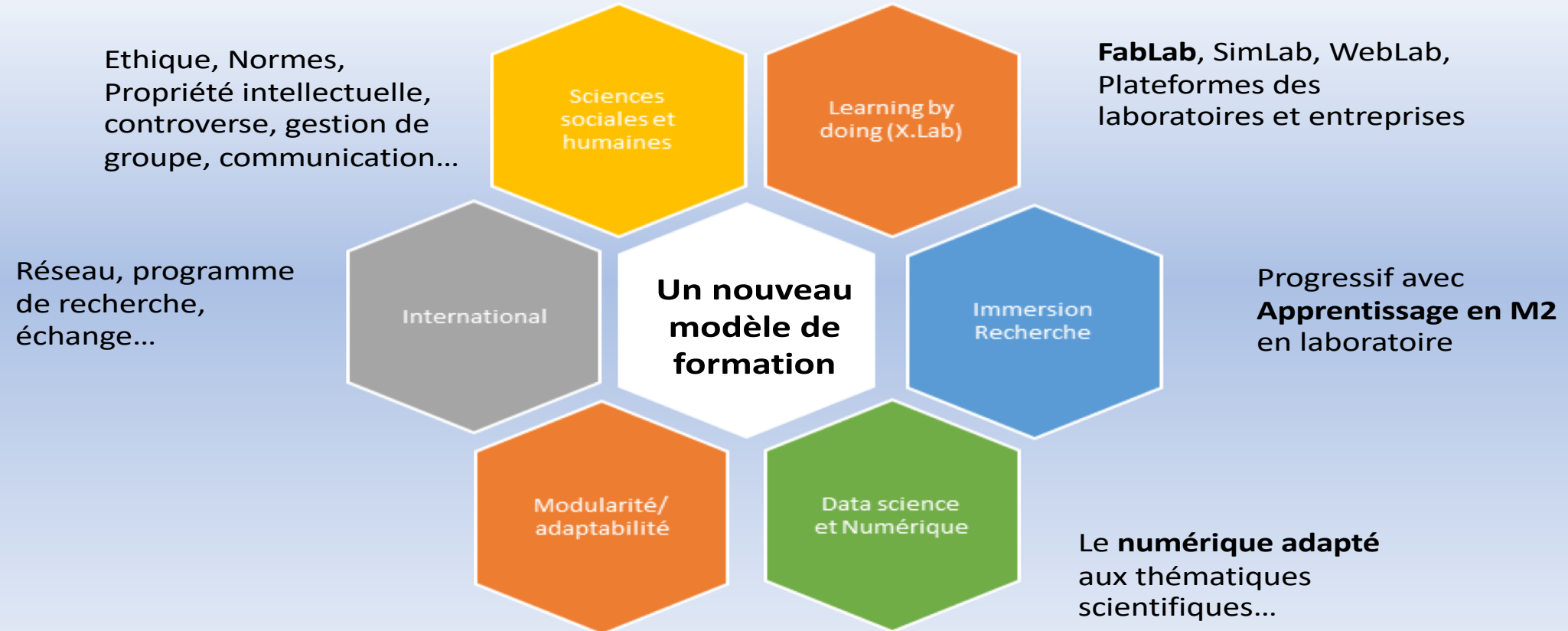


-Sélection des candidatures GPEX après examen des candidatures sur « Mon Master »



-Envoi des Résultats GPEX aux candidats fin mai

Le modèle d'excellence: GPEX en Master/Doctorat, UnderGpex en Licence



- Un **recrutement** des le 1ere année universitaire
- Une **sensibilisation** dès le lycée
- Une forte **personnalisation** des parcours

- Une base **disciplinaire** des formations existantes
- Un **compagnonnage** scientifique et pédagogique
- Des projets collaboratifs **pluridisciplinaires** issus des **partenaires**