

# Research Assistant Positions

## *Cloud Infrastructure Confidential Computing and Trustworthy AI*

School of Internet of Things XJTLU

We are recruiting Research Assistants (RAs) across four complementary technical tracks at the School of Internet of Things, Xi'an Jiaotong-Liverpool University (XJTLU). The work supports the development of the Trustworthy AI and Cyber Resilience Center and a university sandbox for industry co-creation, supervised on-campus project work and research validation.

Professor Dijiang Huang (Principal Investigator) and Dr. Yuji Dong (Co-Principal Investigator) lead the research team. RAs work with academic supervisors and technical mentors, and collaborate with industry engineers on agreed projects.

### **The work**

Our priority is to build usable systems. The XJTLU sandbox will combine cloud-native infrastructure, governed data and model services, hardware-backed confidential computing, and AI applications with repeatable security evaluation. It provides a controlled environment for development and testing before any partner-authorized production transition.

### **Choose your primary track**

- A Cloud Native Architecture and Sandbox Engineering
- B Confidential Computing Hardware and Software Integration
- C AI Multi Agent Applications and Model Post Training
- D Trustworthy AI and Cyber Defense Engineering

Apply for one primary track and, optionally, a secondary track. We value depth and demonstrated implementation ability in your chosen area. No applicant is expected to master all four tracks or every example technology listed below. Track A requires prior hands-on infrastructure deployment experience; the other tracks accept strong relevant projects as evidence of readiness.

### **Shared requirements**

- Demonstrate solid programming and debugging skills through coursework, research, open-source contributions or engineering projects. Understand and explain your own code.
- Use Git, automated tests and technical documentation. Read English technical material and communicate progress, design choices and blockers clearly.
- Work responsibly with confidential information. Use only approved data, credentials, computing resources and security test environments.
- Take ownership of a scoped component, seek review for consequential decisions and collaborate across infrastructure, application and security boundaries.

# **Track A Cloud Native Architecture and Sandbox Engineering**

Design and implement the XJTLU private-cloud sandbox, with responsibility for a scoped infrastructure stack under senior technical review. This is a hands-on systems role requiring deployment and troubleshooting ability, rather than architecture diagrams alone.

## **Responsibilities**

- Translate research and partner requirements into a modular architecture covering compute, networking, storage, identity and application deployment. Document alternatives and operational trade-offs.
- Provision Linux servers or virtual machines and a Kubernetes environment using repeatable configuration and infrastructure-as-code. Integrate GPU scheduling where the assigned hardware supports it.
- Implement project namespaces, role-based access, network policies, quotas, secrets integration and approved dataset/model storage. Connect to existing university services where appropriate.
- Build container-image and deployment workflows with versioned configuration, monitoring, logs, backup/restore and rollback. Troubleshoot networking, storage and resource bottlenecks.
- Provide a usable onboarding route for researchers, interns and industry collaborators. Prepare deployment packages and bounded portability tests for agreed private- or public-cloud targets.

## **Essential skills and experience**

- Practical Linux administration and TCP/IP networking, plus deployment and debugging of a containerized application on Kubernetes.
- Working knowledge of storage, virtualization, container images, identity/access control and service exposure. Ability to automate tasks in Python, Go or shell.
- Evidence of building and maintaining an environment beyond a guided tutorial, including diagnosing a failure and documenting recovery.

## **Desirable experience**

Helm, Terraform or OpenTofu, Ansible, GitOps, GPU scheduling, Prometheus/Grafana, OpenStack, or AWS/GCP/Alibaba Cloud. Experience in one relevant stack is sufficient.

## **Expected deliverables**

A reproducible sandbox deployment repository, architecture and configuration records, access/onboarding instructions, and test evidence for isolation, backup/restore and rollback. Initial acceptance should include deployment and use of one agreed sample application.

## **Scope and responsibility**

The RA contributes technical requirements to campus IT and facilities teams. Electrical work, cooling design, building works, procurement approval and university-wide IT authority remain with the responsible professionals.

# **Track B Confidential Computing Hardware and Software Integration**

Implement and evaluate confidential-computing workflows across hardware capabilities, firmware configuration, virtualization, runtime services and applications. The goal is an evidenced protection chain for approved data and model workloads.

## **Responsibilities**

- Assess the assigned CPU/GPU platform, firmware, drivers and runtime prerequisites. Record supported configurations and their security limitations before deployment.
- Deploy confidential virtual machines or containers on supported hardware. Capture and verify remote attestation evidence against defined workload and platform policies.
- Integrate workload identity and attestation decisions with key-release services. Implement established encryption, credential handling and protected artifact workflows without inventing cryptographic algorithms.
- Trace sensitive inputs, keys, intermediate outputs and logs across the application lifecycle. Identify where plaintext exists and which components belong to the trusted computing base.
- Automate positive and negative tests for attestation, policy rejection, key release, logging and recovery. Compare representative workload performance against a controlled baseline.

## **Essential skills and experience**

- Strong understanding of operating systems, virtualization, memory protection and basic networking, with practical Linux debugging.
- Programming ability in Python, C/C++, Go or Rust, and familiarity with encryption, digital signatures, certificates and key-management fundamentals.
- A substantial systems or security implementation project. Prior TEE deployment is advantageous, but strong fundamentals and demonstrated ability to reproduce technical experiments are acceptable.

## **Desirable experience**

Experience with Intel TDX/SGX, AMD SEV-SNP, supported GPU confidential computing, Confidential Containers, TPMs or key-management services. Depth in one platform is sufficient.

## **Expected deliverables**

A supported-configuration record, reproducible deployment scripts, an attestation-to-key-release demonstration, documented trust boundaries, negative-test evidence and a measured performance report for an agreed workload.

## **Scope and responsibility**

Hardware-to-software work means configuration, integration and verification, not chip design. CPU and GPU trust must be verified separately where relevant. Confidential computing does not by itself prevent malicious application logic or leakage through permitted outputs.

# Track C AI Multi Agent Applications and Model Post Training

Build task-oriented applications using existing large language models (LLMs), and adapt selected models where evidence supports fine-tuning. This role covers engineering beyond an initial chat demo, including reliable tool use, evaluation and deployment.

## Responsibilities

- Turn an agreed research or industry workflow into an application with clear inputs, outputs and success criteria. Implement APIs, backend services and a usable interface where required.
- Develop agent workflows with tool calling, shared state, task routing, recovery and human approval for consequential actions. Use multiple agents only when they improve on a simpler baseline.
- Build retrieval-augmented generation (RAG) and data connectors with access-aware retrieval, source attribution and controlled handling of project documents.
- Prepare approved training data and implement supervised fine-tuning or parameter-efficient adaptation such as LoRA/QLoRA on selected models. Track dataset, model and experiment versions.
- Evaluate task success, factual grounding, tool-use errors, latency and resource cost. Compare base, retrieval-based and fine-tuned approaches using held-out data and prevent evaluation leakage.
- Package applications and model-serving components for the sandbox, with permissions, secrets handling, traces, regression tests and rollback to a known version.

## Essential skills and experience

- Strong Python programming, API integration and software debugging, with an implemented AI/ML application or a reproducible training project.
- Practical understanding of neural-network training and inference, datasets and evaluation. Ability to use PyTorch or an equivalent framework.
- Ability to explain why a workflow needs retrieval, tool use, multiple agents or fine-tuning, and to test the chosen design rather than rely on a successful demo.

## Desirable experience

Hugging Face Transformers/PEFT, model-serving frameworks, vector search, LangGraph or another agent framework, supervised fine-tuning, preference optimization, or GPU memory optimization. Equivalent tools are welcome.

## Expected deliverables

One scoped application with reproducible deployment, an evaluation dataset and report, recorded baseline comparisons, and operational documentation. Where fine-tuning is justified, also deliver a versioned adaptation pipeline and model/adaptor evaluation.

## Scope and responsibility

The focus is post-training and application development on existing models, not pretraining a foundation model from scratch. RAG does not update model weights. Multi-agent execution alone is not evidence of improved accuracy or reliability.

## Track D Trustworthy AI and Cyber Defense Engineering

Develop practical protection and evaluation capabilities for AI applications and their supporting systems. Assess both model/data risks and conventional cyber risks, and turn findings into reproducible tests and defensible mitigation advice.

### Responsibilities

- Define threat models for the selected application, including users, model providers, data owners, agent tools and infrastructure administrators. State assets, trust boundaries and permissible tests.
- Evaluate AI risks such as prompt injection, unauthorized tool use, sensitive-output leakage, unreliable answers and relevant model attacks using approved datasets and controlled experiments.
- Assess infrastructure and application controls through code review, dependency/container checks, access tests and log analysis. Reproduce findings and verify narrowly scoped mitigations.
- Build detection and investigation workflows from approved telemetry, including event correlation and AI-assisted triage. Measure detection quality and false positives against known test cases.
- Exercise incident response and recovery in the sandbox, including containment, evidence preservation and restoration. Produce reports that distinguish observed findings, limitations and untested claims.

### Essential skills and experience

- Practical security knowledge in at least one area: application/cloud security, network defense, systems security or ML/LLM security.
- Ability to write Python or another suitable language, analyze logs, reproduce a failure and explain the evidence for a security conclusion.
- Understanding of authentication, authorization, common attack surfaces and experimental evaluation. Ability to work within explicit authorization and data boundaries.

### Desirable experience

Threat modeling, SIEM/detection engineering, incident response, software supply-chain security, adversarial ML, privacy evaluation, differential privacy, federated learning, or embedded/AIoT security.

### Expected deliverables

An application-specific threat model and test matrix, reproducible defensive test scripts, a findings/mitigation report and one validated detection or recovery demonstration. Report known limits rather than claiming complete security or certification.

### Scope and responsibility

Security testing is restricted to approved assets and scoped sandbox exercises. External scanning, partner production testing and access to sensitive datasets require separate authorization. The role does not carry 24/7 SOC or enterprise compliance ownership.

# Working arrangements and application

## Working with the team

- The team provides scientific direction and technical mentoring. RAs implement assigned components, participate in code review and demonstrate working results with reproducible evidence.
- Cloud and hardware integration require regular on-site work at XJTU's Taicang campus. Other work may be flexible by agreement, but applicants should state their availability and ability to attend on-site activities.
- Industry-linked work can include supervised on-campus development and internship activities where arrangements permit. Academic supervision, partner mentoring, data/IP rights and access permissions are agreed for each project.
- Responsibilities and initial milestones are agreed according to the selected track, experience and assigned resources. We do not expect one RA to independently architect, deploy and secure an entire production platform.

## Engineering and research practice

- Maintain reviewable source code, dependency/configuration records, tests and short operating instructions. Separate demonstrations, local tests and actual deployment evidence.
- AI coding tools may support the work, but candidates must understand, test and explain the code they submit. Do not upload partner data, credentials or unpublished code to unapproved services.
- Research outputs may include technical reports, demonstrators, open-source contributions, papers or intellectual property, subject to contribution, supervision and project agreements.

## How to apply

Send your application to both contacts:

Jie.Song@xjtlu.edu.cn

- CV with education, relevant coursework and projects. Specify your primary track A, B, C or D and any secondary interest.
- A short statement describing a system or application you personally built, the part you owned, a difficult problem you solved, and why you fit the selected track.
- A repository, code sample, deployment record or technical report where available. A sanitized sample or private walkthrough is acceptable when work is confidential.
- Your availability, possible start date and ability to work on site. Include relevant study or employment commitments.

Suggested subject: RA Application – Track [A/B/C/D] – [Name]

## Selection

Selection may include a project walkthrough and a scoped practical discussion or exercise appropriate to the chosen track. We will focus on implementation ability, debugging, technical judgment and clear explanation rather than familiarity with every named tool.

## Appointment details

Appointment category, number of openings, duration, working hours, remuneration and start date will be confirmed through the project and university appointment process. Do not assume a particular salary, contract period or guaranteed internship placement from this description.