# A Report on the NSF NAIRR Software Workshop

Held at Argonne National Laboratory
December 3 - 4, 2024

## Executive Summary

On behalf of the National Science Foundation's *Office of Advanced Cyberinfrastructure* (OAC) and the Department of Energy's *Advanced Scientific Research Computing* (ASCR) program, a National Artificial Intelligence Research Resource (NAIRR) software workshop was held at Argonne National Laboratory on December 3–4, 2024.

The workshop focused on identifying a feasible AI software stack (or set of stacks)—comprising computer programs, training and inference frameworks, libraries, user interfaces, data management, debuggers, and performance tools—to be made available to the broadest possible community. A key goal was to determine the feasibility, necessary components, and future research and development required to support a long-term NAIRR effort. This effort, anticipated to launch in 1 to 1.5 years and target a 5-year horizon, will rely on robust software solutions that address the evolving needs of High-Performance Computing (HPC) and AI communities.

The workshop was organized by a technical steering committee composed of members from universities, national laboratories, and industry. More than 120 researchers, engineers, and educators from diverse organizations participated. Over the course of two days, attendees participated in a keynote talk, multiple invited talks, several breakout sessions, and a concluding panel discussion.

Some of the main conclusions of the workshop are as follows:

1. The NAIRR Pilot's software stack will leverage both existing and emerging software solutions, catering to a range of users (from novices to experts) across diverse hardware platforms and accelerators, including those used for education. Notably, the HPC community views these solutions as a layered ***software stack*** optimized for performance and scalability. In contrast, the AI community often refers to a broader ***software ecosystem*** that tightly integrates data, user support, and training frameworks. The NAIRR effort must bridge these perspectives to serve all stakeholders effectively.
2. The stack must respond to the evolving needs of the scientific and AI communities, including real-time data analysis, privacy and security challenges, and portability across emerging AI hardware. Data management, which encompasses cleaning, curation, and annotation, to ensure that researchers can fully leverage the growing volume of diverse datasets.
3. The following software components for the NAIRR stack must remain flexible and extensible to accommodate future technology advances:
   - Operating systems,
   - Middleware solutions for communication and resource management,
   - Languages and compiler support (with emphasis on Python, Julia, C, C++, and Fortran),
   - Workflow managers and AI-related libraries/models/frameworks.
4. Embracing open-source development and ensuring support for new hardware will be essential for keeping the NAIRR stack at the forefront of technological advances.

5. The stack should offer easy-to-use interfaces (e.g., Jupyter Notebooks, web-based platforms) to lower the barrier for newcomers to AI. At the same time, training and continuous user support cannot be separated from the software itself, underscoring the need for educational resources and dedicated guidance to help new users navigate this technology.
6. The stack must address near-term user-support needs; for instance, funding small supplements for current grantees during the NAIRR Pilot was proposed. Attendees also recommended creating intuitive *chatbot* interfaces to help users interact with the software stack, further reducing barriers to adoption and ensuring efficient troubleshooting and assistance.

# 1 Introduction

The National Artificial Intelligence Research Resource (NAIRR) Task Force identified the need to democratize access to artificial intelligence (AI) resources traditionally limited to large organizations. The Task Force's 2023 report, ***Strengthening and Democratizing the U.S. Artificial Intelligence Innovation Ecosystem: An Implementation Plan for a National Artificial Intelligence Research Resource***, underscored the importance of creating an inclusive AI ecosystem that fosters innovation, enhances diversity, and ensures equitable access to AI capabilities. NAIRR pursues four key goals:

- spurring innovation,
- increasing the diversity of talent,
- improving capacity, and
- advancing trustworthy AI.

To achieve these goals, at the request of the National Science Foundation's *Office of Advanced Cyberinfrastructure* (OAC) and the Department of Energy's *Advanced Scientific Research Computing* (ASCR) program, we organized a NAIRR software workshop.

The workshop aimed to discuss an AI software stack or a set of AI software stacks, including computer programs, training and inference frameworks, libraries, user interfaces, data management, curation, debuggers, and performance tools, to be available to the broadest possible audience. The NAIRR software stack should strive to reach across scientific research domains, scales, and users, leveraging current software stacks used in academia, national laboratories, and industry.

This workshop report presents the NAIRR Pilot's immediate needs and long-term goals, considers the composition of the NAIRR software stack over two to five years, and aims to address the evolving needs of the scientific community. These needs include real-time analysis of sensor/experimental data and decision-making using AI, privacy and security requirements in AI-based scientific applications, foundation models, mixed precision libraries, operating systems, programming environments, toolchains, storage needs for the ever-growing volume of training data, and portability of software across emerging novel AI hardware platforms.

The NAIRR Pilot's software stack will consist of existing software catering to diverse users, from beginners to experts, spanning multiple communities, including education, diverse hardware platforms, and current and emerging accelerators. The report also aims to address immediate user support needs and propose funding small supplements for current grantees during the NAIRR Pilot.

One essential purpose of the workshop was to determine the feasibility and needs of the required software stack and related research and development to sustain a long-term NAIRR effort, which is expected to start in 1 to 1.5 years and target a 5-year horizon. The workshop also explored procedures for adding newly developed software to the NAIRR software stack and future funding needs specifically for NAIRR software. This inquiry was separate from funding fundamental research in AI or AI applications in science, for which federal funding agencies have existing solicitations.

The workshop report also focuses on creating ethical, transparent (explainable), and trustworthy AI. AI for science, the goal of NAIRR, differs from AI for industry, necessitating a tailored software stack. Furthermore, the workshop included users who were not fluent in high-performance computing or simulations, such as experimentalists managing a deluge. The final product of the workshop is a cohesive report that ensures the seamless integration of inputs from all subgroups involved in the workshop.

The NAIRR software workshop aligned with the vision outlined in the NAIRR Task Force's final report and lays the foundation for a robust, democratized AI research infrastructure that empowers diverse users and drives innovation across the U.S. AI ecosystem.

The NAIRR Task Force identified the critical need to democratize access to AI resources, which have traditionally been accessible only to large organizations with substantial funds. The NAIRR Task Force's 2023 report emphasizes creating an inclusive AI ecosystem that fosters innovation, enhances diversity, and ensures equitable access to AI capabilities. To achieve these goals, NAIRR aims to spur innovation, increase the diversity of talent, improve capacity, and advance trustworthy AI. The NAIRR software workshop was essential for defining an AI software stack that can be broadly accessible, addressing both the immediate needs and long-term goals of the NAIRR Pilot. The NAIRR software workshop was crucial for determining how to leverage existing software stacks used in academia, national laboratories, and industry to create a comprehensive AI software stack that can support diverse users across various scientific research domains.

Furthermore, it addressed the evolving needs of the scientific community, such as real-time data analysis, privacy and security in AI applications, and software portability across emerging AI hardware platforms. By focusing on the feasibility and needs of a longer-term NAIRR, this workshop explored procedures for incorporating new software developments and future funding requirements, ensuring that AI tools and frameworks remain cutting-edge, user-friendly, and adaptable to various research environments. Ultimately, the workshop helped establish a robust, democratized AI research infrastructure that empowers diverse users and drives innovation across the U.S. AI ecosystem.

Specifically, the workshop delved into the various components that should comprise the NAIRR software stack. The key elements under consideration included operating systems like Unix and Linux, which form the backbone of many AI applications. The discussion also encompassed middleware solutions for communication and resource management. Language support and compilers were another focal point, with special attention given to widely used languages such as Python, Julia, C, C++, and Fortran, which are crucial for developing AI models and performing computational tasks. Additionally, the workshop explored workflow managers and AI-related libraries, including machine learning and deep learning libraries, models, and frameworks, which are integral to developing and deploying AI applications.

The goal was to identify various software choices that cater to users' varied needs without prescribing a single vendor. Emphasis was placed on open-source options to ensure broad accessibility and adaptability. We also conducted a pre-workshop user needs survey to understand better the software and libraries required for AI research.

Further discussions addressed immediate user-support needs for the NAIRR Pilot software stack, established priorities for the comprehensive NAIRR software stack, and identified future investment needs. These discussions focused on open-source development and support for emerging hardware, ensuring that the NAIRR software stack remains cutting-edge and relevant to evolving technological advancements.

## 2 Case Studies

At the outset, six domain-focused case studies illustrated how AI can accelerate discovery, facilitate novel analyses, and open new avenues for scientific exploration. These studies spanned x-ray science for real-time materials characterization, advanced biology for protein design, and the integration of AI in science education. They also addressed AI-driven methods for cosmology, regulatory challenges in medical imaging, and the application of machine learning architectures in weather forecasting. Collectively, these examples demonstrate both the promise of AI workflows in transforming scientific research and the persistent gaps that must be bridged to realize that promise at scale. The following section details each case study, examining the opportunities for innovation as well as the specific challenges—ranging from data collection and foundational model development to infrastructure requirements and formal uncertainty quantification—that emerged in these various domains.

End-to-end **x-ray science** powered by HPC and AI will unlock new scientific capabilities from existing instruments used in materials characterization. For example, at the Advanced Photon Source, a large-scale experimental user facility, AI at the edge supports real-time analysis of Gb/s data streams, producing often more accurate results 100 times faster. It also facilitates self-driving experiments and instruments to maximize information gain in minimal time and learns material physics directly from measurements, thereby expanding the knowledge base. **Gaps:** There are challenges associated with AI-aided real-time data analysis, foundation models, and the curation of data and models. A key issue is the need for an enhanced Pareto front for model performance and physics-aware deep learning, which includes both complex-valued operations and hardware-efficient, differentiable scientific operations. As we work towards a comprehensive scientific AI assistant for experiment planning, guidance, and operation, we face gaps such as the need for multimodal, scientific context-aware foundation models, standard interfaces for tool usage, agentic AI capabilities, and seamless machine learning operations (MLOps) for the evaluation and deployment of foundation models.

In **biology**, programmable protein design entails a framework that allows users to prescribe programmable design constraints via a natural language interface for ease and flexibility. A critical challenge in realizing such a framework is a lack of comprehensive multimodal protein design datasets that integrate text, protein/gene sequence, and structure/conformational modalities to build aligned representations for protein sequencefunction mapping. Curating such a dataset requires LLM-assisted workflows to create rich narratives that can resolve potential issues like mode collapse. Additionally, we lack workflows effectively designed to integrate experimental observables with foundation models seamlessly. Moreover, many of these observations are qualitative and not always quantitative, and there is a lack of sufficient experimentally labeled datasets. **Gaps:** To train such multimodal foundation models, we require high-performant software stacks that are easy to customize. For instance, existing software stacks for text-vision models are not easily transferrable to protein multimodal models. The current stacks require significant development time to incorporate custom changes. There is also a need for libraries that have ease of use while retaining scalability and performance. Finally, in the broader context of automated scientific discovery, there are software gaps in automated test beds that link foundation model outputs with self-driving laboratories. We require agentic frameworks to evaluate HPC resources and select the top-performing candidates for self-driving laboratory experiments. The framework must be customizable to implement agents with the sophistication to plan and execute fine-grained steps of robotic pipelines to realize the self-driving experiments and record experimental outputs to provide feedback for the foundation models.

As a data-rich science, **cosmology** is an excellent application domain for AI/ML methods. A confluence of data-intensive and high-performance computing swim lanes will accelerate adoption. AI methods have solved and will solve problems that could not be approached otherwise. There are many places where these methods can and must be applied due to the size and complexity of data sets. However, the status of formal uncertainty quantification (UQ) applications in these areas is still relatively crude, mainly because the problem is very complex. The presence of bias due to a number of problems with measurements and modeling uncertainties and assumptions remains a key issue, as techniques such as discrepancy modeling are not yet mature enough. A combination of physical/modeling input into purely data-based methods is needed, as is widely recognized, but current approaches only represent a starting point. Large-scale models combined with massive computing resources can open new avenues. **Gaps:** Historically, there was a gap between HPC and AI hardware; now, they are essentially the same, the consequences of which are still unknown. AI applications (LLMs primarily) are driving hardware evolution away from double precision, and modeling and simulation software tools must address the challenge of dealing with mixed precision, an argument to have a somewhat unified toolchain for AI and modeling and simulation. The diversity of the AI for Science (AI4S) application space is daunting; it is significantly more complex than the space that HPC professionals are used to. Nevertheless, the AI application ecosystem has a robust framework for dealing with this, including DL frameworks, ML libraries, NLP tools, notebooks, APIs, and more. However, the optimal strategies for bringing together HPC and AI approaches, such as modular design and workflow management systems, remain uncertain. A hybrid approach to integrating HPC and AI software stacks is probably best, and best determined by pilot projects rather than a top-down approach. HPC facilities will need to become more cloud-like to support the AI toolchain (containerization, orchestration, elastic scheduling, support for hybrid workflows, etc.).

**Weather forecasting** is an excellent testbed for newly developed machine learning architectures, as the extensive observational data necessary to make accurate predictions push the limits of current hardware and software. ERA5, the fifth generation of climate reanalysis produced by the European Centre for Medium-Range Weather Forecasts (ECMWF), offers hourly data on various atmospheric, land-surface, and ocean-state parameters, along with uncertainty estimates. Tasks such as predicting the three-dimensional (3D) atmosphere for medium-range weather forecasting (up to 14-day lead time), creating emulators for climate research, and downscaling images for local-scale impacts of weather and climate, rely on datasets of hundreds of terabytes to petabytes of data. The advent of scalable machine learning architectures (e.g., transformers), the availability of high-quality data, and access to numerous GPUs/TPUs is driving a paradigm shift in weather forecasting. Current software is primarily designed for traditional vision-based tasks, encompassing everything from data loading to readily available architectures. This limitation affects academic researchers and helps explain why most of these models are created by large technology companies. **Gaps:** Due to the large image sizes (721 x 1440) and the number of channels (potentially hundreds to thousands), substantial GPU memory is necessary for the activations alone. I/O is typically the limiting factor for training, so a significant challenge is adapting the current generation of hardware and software to these datasets. Custom I/O for training with node-local storage and improved caching/prefetching compared to native PyTorch Lightning has enhanced training time by 20-30 percent (DALI and DLIO for benchmarking and profiling); however, future generations will need petabytes of training data, and the model architecture along with deep learning packages are not optimized for models with O(100) channels. Nontrivial model parallelisms are also necessary to fit these large models into memory from gradient checkpointing, sharding model parameters, and tensor parallels. Moreover, PyTorch requires considerable customizations to minimize pre-processing and any nontrivial CPU-based pipelines, such as asynchronous operations.

Within **regulatory science**, research across numerous program areas treats AI and machine learning as major focal points for developing lifesaving medical devices. Tools are designed to help innovators assess the safety and effectiveness of emerging technologies at every stage of device development. Over the past decade, advancements in AI image processing have led to a marked increase in clearances and approvals across diverse product areas. However, the availability of and access to medical imaging data remain a persistent challenge. To address this, a recent collaborative initiative was launched to establish a medical imaging data exchange platform equipped with built-in tools for creating algorithms that meet regulatory standards, ultimately benefiting the broader imaging ecosystem. **Gaps:** Current generative AI research centers on formulating a case-agnostic approach to assessing factual accuracy, employing performance assessment strategies such as benchmarking, expert evaluation, and model-based evaluation. Key hurdles in medical AI, from a regulatory standpoint, include the limited availability of accessible, sustainable data platforms that meet stringent requirements, the need for advanced evaluation platforms incorporating new methodologies and performance metrics, and the complexities involved in determining the quality of synthetic data.

Recognizing the expanding **role of AI in education** across diverse disciplines, a large environmental science course integrated AI-driven tools to enhance programming support and tackle complex analytical problems. This integration opened new opportunities for students to explore large-scale geospatial and observational data, apply advanced machine learning techniques to ecosystem modeling, and conduct real-time analyses of climate variables. Although these capabilities significantly extended the scientific scope of the course, they also surfaced multiple areas in need of attention. **Gaps:** The course implementation revealed a requirement for specialized hardware and local storage to run large-scale AI applications, as well as the limitations of the initial configuration in handling complex tasks. A persistent challenge lay in developing an adaptable, scalable infrastructure capable of accommodating evolving AI tools. Another gap was the absence of streamlined mechanisms for rapid deployment and testing across diverse AI models. Furthermore, linking Jupyter-AI with the course's tools necessitated custom configurations for an online environment, highlighting the complexity of such setups. Ultimately, this experience underscored that truly transformative AI in educational settings requires robust technical foundations, seamless model integrations, and carefully preconfigured AI environments to enable broad-scale access and flexibility.

# 3 State of the Community

## 3.1 Current Software, Tools, and Gaps

A wealth of HPC and AI resources exists across training/inference, data management, models/datasets, accessibility/usability, and security/privacy. Yet, *integration* across these focus areas remains a persistent hurdle. Researchers continue to seek end-to-end workflows—from initial data ingestion and curation to final model deployment—that are robust, reproducible, and secure. This section explores the community's growing *demand* for better incentives, privacy-aware frameworks, seamless hybrid HPC–Cloud–Edge infrastructures, and comprehensive education initiatives that lower the barrier to entry for domain experts. Finally, we discuss the critical role of a national-scale resource like **NAIRR** in complementing, rather than duplicating, existing industry-driven and HPC solutions while prioritizing the *unique needs of science*.

### 3.1.1 Training & Inference (Table: 1 & 2)

Training and inference remain central pillars in AI workflows. The research community, spanning academia, government labs, and industry, predominantly relies on well-known deep learning frameworks like **PyTorch** and **TensorFlow**. For large-scale model training, frameworks such as **DeepSpeed** and **Megatron-LM** enable efficient distributed training, especially for Large Language Models (LLMs). Meanwhile, tools like **vLLM** and **TensorRT-LLM** optimize inference performance for production deployments.

Beyond the core frameworks, there is an increasing need for containerization (e.g., via **Docker** or **Singularity/Apptainer**) and for reproducible environments, especially in the HPC ecosystem. Researchers also see value in **JAX** for high-performance ML and differentiable programming, while **Hugging Face** Spaces and model repositories ease experimentation and community sharing.

Despite the rich ecosystem, multiple **gaps** remain. Users require streamlined pipeline management and better environment version control to avoid the pitfalls of inconsistent or brittle deployments. As large and multimodal datasets become more prevalent, resource reservation and job scheduling complexities (e.g., across distributed systems) can become bottlenecks. Another critical gap emerges around bridging HPC and emerging AI stacks: how to efficiently port PyTorch or other frameworks to new hardware (Cerebras, Graphcore, AMD, etc.) while retaining reproducibility and performance.

Several participants emphasized the importance of privacy-preserving and secure infrastructure for training on proprietary or sensitive data and the imperative to integrate *non-deep-learning* AI methods, domain-specific simulations, and trustworthy AI frameworks into mainstream training/inference pipelines. Finally, reproducibility—particularly the ability to checkpoint, convert, and retrain models—remains a significant challenge at scale.

| **Tool/Software** | **Description/Use Case** |
|---|---|
| **PyTorch & TensorFlow** | Core deep learning frameworks for training & inference across multiple domains. |
| **DeepSpeed & Megatron-LM** | Distributed training of large-scale models (LLMs). |
| **JAX** | Differentiable programming, high-performance ML, often used in research. |
| **Hugging Face Repos/Spaces** | Hosting & sharing of pre-trained models; quick demos for domain use cases. |
| **vLLM** | Optimized large language model inference with low latency. |
| **TensorRT-LLM** | NVIDIA's high-performance inference engine for LLMs. |
| **Containers (Docker, Singularity)** | Portable, reproducible environments for AI workflows. |
| **Scikit-Learn** | Classic machine learning, widely used for simpler training & inference tasks. |
| **QisKit, PennyLane, TorchQuantum** | Quantum/ML frameworks (used in specialized research). |
| **Llama, llama.cpp, ollama** | Foundation model & inference frameworks for LLM experimentation. |
| **Lightning AI** | Wrappers for PyTorch/TF enabling easier training & scaling. |

**Table 1:** Commonly used tools and software for training and inference.

| **Gap** | **Description/Need** |
|---|---|
| **Pipeline Management & Monitoring** | End-to-end training/inference pipelines are complex; need integrated workflow & provenance tools. |
| **Resource Reservation & Usage** | Allocating and scaling HPC/cloud resources remains challenging for end-users. |
| **Software Version & Environment Control** | Tools like Spack/E4S exist, but consistent environment management is still burdensome. |
| **Conversion of Training Checkpoints** | Converting checkpoints for cross-framework inference or edge deployment is error-prone. |
| **AI on Private/Proprietary Data** | Confidentiality concerns require robust privacy-preserving training & inference methods. |
| **Performance Portability & Scaling** | Users want their code to seamlessly run on GPUs, specialized HW (Cerebras, Graphcore), HPC clusters, etc. |
| **Incentives & Reproducibility** | Lack of clear incentives to share reproducible pipelines; reproducibility remains a cultural & technical gap. |

**Table 2:** Top gaps in training and inference workflows.

### 3.1.2 Data Management & Storage (Table: 3 & 4)

Effective data management and storage solutions underpin the success of AI projects, yet much of the existing storage stack (e.g., parallel filesystems such as **Lustre**, **DAOS**, **Spectrum Scale**) was initially engineered for HPC workloads with large sequential I/O. AI training often involves many small, random reads, dynamic data augmentation, and the need to quickly load large batches of unstructured data (e.g., images, text, videos).

Consequently, frameworks like **Globus** for data transfer and **MinIO** for object-based storage are increasingly important, complemented by standard scientific formats such as **HDF5**. Multiple workshop participants highlighted the importance of **FAIR principles** for data (Findable, Accessible, Interoperable, Reusable) as well as **FAIR4ML** considerations. Tools that integrate metadata (including domain ontologies) are needed to handle the diversity and complexity of emerging AI datasets.

Key **gaps** include the lack of user-friendly data lifecycle management (especially for large, ever-growing datasets), the difficulty of integrating domain-specific metadata and provenance, and bridging HPC systems' node-local storage with shared parallel filesystems in a way that is easy for non-expert users. The community is also asking for a "data commons" approach allowing for shared, curated, domain-relevant datasets, provided privacy and licensing constraints are respected.

| Tool/Software | Description/Use Case |
|---|---|
| **HDF5 & NetCDF** | Common scientific data formats for array-based data. |
| **Globus** | Data access, sharing, & transfers across different sites. |
| **MinIO** | Distributed object storage frequently used for AI data. |
| **Parallel Filesystems (Lustre, DAOS, Spectrum Scale)** | HPC-grade storage often used for large-scale training. |
| **Data Version Control (DVC)** | Versioning of data & experiment tracking. |
| **Metadata & Ontologies (HPC-FAIR)** | Tools/ontologies for describing data, e.g. HPC-FAIR, domain-specific ontologies. |

**Table 3:** Common tools for data management and storage in AI.

| Gap | Description/Need |
|---|---|
| **AI-ready Metadata Layer** | Tools to systematically capture domain-specific metadata, provenance, and semantics. |
| **Unified Ontologies & Standards** | Lack of consistent data schemas across scientific domains hinders reusability. |
| **Multi-tiered Storage Integration** | Need seamless bridging of node-local and shared filesystems (burst buffers, HPC, cloud). |
| **Lifecycle & Provenance Management** | End-to-end policies for data creation, curation, archival, and potential unlearning/removal. |
| **Data Privacy & Confidentiality** | Mechanisms for secure storage & controlled access, especially in regulated domains. |
| **Scalability of Data Movement** | Transferring multi-terabyte datasets from distributed locations is expensive & time-consuming. |

**Table 4:** Top gaps in data management and storage.

### 3.1.3 Current Models, Datasets, and Gaps (Table: 5 & 6)

The ecosystem of AI **models** and **datasets** is increasingly diverse. Public model repositories (e.g., **Hugging Face**, **OpenMined**, domain-specific repositories) have enabled a proliferation of pre-trained models—particularly large language models and foundational vision models. Domain researchers in agriculture, climate science, health/medicine, and materials science have begun adopting these models, often requiring specialized datasets such as **PlantVillage**, **Phenobench**, or curated medical datasets subject to HIPAA compliance.

Crucial **gaps** include the need for better data curation (e.g., removing or correcting "bad" data), robust annotation tools, and the ability to *update* or *untrain* models without retraining from scratch. Synthetic data generation is gaining traction to address proprietary or sparse datasets, but best practices for verifying data authenticity and fidelity are still developing. Researchers desire more explicit **FAIR4ML schemas** to describe models and data, enabling more consistent, transparent sharing across different platforms.

| Model/Dataset/Software | Description/Use Case |
|---|---|
| **Hugging Face Repositories** | Hosting & serving pre-trained models (transformers, LLMs, etc.). |
| **PlantVillage, Weed Detection, CropAndWeed, Fruits-360** | Public agriculture datasets used for plant disease/weed detection tasks. |
| **Corn/Soybean Disease, Growth Stages (private)** | Proprietary agriculture datasets used in industry or specialized research. |
| **All of Us (NIH)** | Confidential medical dataset, subject to stringent privacy & IRB rules. |
| **Kaggle, Data.gov, NASA/CDF** | Wide variety of open datasets for AI & data science competitions. |
| **Domain-Specific Repositories** | E.g., `Phenobench` in crop research, `Flatiron` multimodal cosmology data. |

**Table 5:** Common models and datasets in use across various domains.

| Gap | Description/Need |
|---|---|
| **FAIR Model Schemas (FAIR4ML)** | Standardized ways to describe, discover, and reuse models & their training data. |
| **Model Unlearning & Continual Training** | Efficient removal of problematic data or incremental updates without full retraining. |
| **Synthetic Data Generation** | Tools for generating domain-specific synthetic data while preserving statistical fidelity. |
| **Benchmarking & Rigorous Evaluation** | Need standardized metrics for comparing models across tasks and domains. |
| **Data Mobility & Federated Learning** | "Moving compute to the data" to address connectivity or privacy constraints. |
| **Multi-modal Integration** | Merging images, text, sensor data, and simulation outputs into cohesive models. |

**Table 6:** Top gaps in models and datasets for AI.

### 3.1.4 Current Accessibility & Usability and Gaps (Table: 7 & 8)

A recurring theme is that *accessibility* in AI is about more than raw computing capacity. Researchers and practitioners seek frictionless environments—commonly, **Jupyter Notebooks** and web-based platforms—that minimize overhead for domain scientists who may not be HPC experts. Tools such as **Open OnDemand**, **Globus Compute**, or discipline-specific **Science Gateways** have made strides. Yet, the complexity of large-scale AI remains a barrier to new entrants.

Workshop participants also cited the importance of **facilitators** (akin to XSEDE's ECSS or "AI research facilitators") who provide hands-on support. Standardizing data collection and annotation practices would help ensure that domain experts can more easily *contribute* as well as *consume* AI resources. The "digital divide" extends to AI in many domains, with limited connectivity in rural or resource-poor regions, making data ingestion and model inference difficult.

| Tool/Software | Description/Use Case |
|---|---|
| **JupyterHub, Jupyter Notebooks** | Interactive computing environment for prototyping, teaching, & collaboration. |
| **Open OnDemand, Globus Compute** | Simplified web portals for HPC/AI resource access & job management. |
| **Science Gateways** | Domain-focused portals (GUI-based) for specialized AI/ML tasks. |
| **TAPIS, DIAMOND, TACC interfaces** | HPC/AI abstraction layers and workflow engines. |

**Table 7:** Common accessibility and usability tools for AI.

| Gap | Description/Need |
|---|---|
| **Ease of Environment Setup** | Containerization & environment mismatch hamper broad adoption; simpler solutions needed. |
| **Onboarding & Education** | Many new AI users lack HPC experience; guided learning paths or "AI facilitators" would help. |
| **Data Movement & Management** | Users struggle with multi-step workflows to ingest large data for training or analysis. |
| **Computational Resource Heterogeneity** | Each cluster has different scheduling, container, or library constraints. |
| **Low-code/No-code Interfaces** | Domain experts want GUIs or chat-like interfaces for model exploration without heavy coding. |
| **Digital Divide & Connectivity** | Rural or under-resourced communities have limited network bandwidth for data transfer. |

**Table 8:** Top gaps in accessibility and usability for AI workflows.

### 3.1.5 Security & Privacy (Table: 9 & 10)

Security and privacy considerations loom as AI permeates sensitive domains (e.g., healthcare, finance, agriculture with proprietary genetics data). A variety of **Privacy Enhancing Technologies (PETs)** exist, including **Homomorphic Encryption**, **Differential Privacy**, **Secure Multi-Party Computation**, and **Federated Learning**, but many remain complex to implement at scale.

Workshop participants also pointed to the growing need for adversarial robustness and **red-teaming** tools (e.g., IBM's Adversarial Robustness Toolbox and Microsoft's Counterfit). However, *practical* best practices around model security (model theft, data leakage, etc.) are still lacking. End-to-end security (from data ingestion to model deployment) rarely has a single blueprint, especially in interdisciplinary, multi-institution collaborations. Education, reproducibility, and the notion of *trustworthy AI* are further cross-cutting challenges.

| Tool/Software | Description/Use Case |
|---|---|
| **Privacy Enhancing Crypto (PECs)** | Homomorphic encryption (HE/FHE/PHE), ZKPs, secure MPC, differential privacy. |
| **Adversarial AI Evaluation (ART, Counterfit)** | Toolkits to test AI models against adversarial attacks. |
| **Trusted Execution Environments (TEEs)** | Hardware-based enclaves (Intel SGX, AMD SEV) for secure computation. |
| **Federated Learning Frameworks** | Often built atop PyTorch or TensorFlow to train models without centralizing data. |
| **Confidential Computing (NVIDIA, Intel TDX)** | Industry solutions to protect data and models in hardware-based secure enclaves. |

**Table 9:** Common security and privacy tools for AI.

| Gap | Description/Need |
|---|---|
| **Practical PET Integration** | Many privacy-enhancing cryptographic methods remain difficult to deploy & scale. |
| **Standardized Security/Privacy Blueprints** | Researchers lack reference architectures for end-to-end secure AI. |
| **Data Governance & Ownership** | Clear policies for who owns the data/models, especially in multi-institution consortia. |
| **Adversarial Robustness Testing** | Tools exist but remain underused; best practices for systematic red-teaming are lacking. |
| **Federated Identity & Access Management** | Need robust solutions for cross-institution authentication (e.g., InCommon, NIH login). |
| **Regulatory & Ethical Gaps** | AI regulations, fairness, bias, and explainability remain unaddressed in many domains. |
| **Model Unlearning & Data Removal** | Mechanisms to remove or anonymize data points after model training are still nascent. |

**Table 10:** Top gaps in security and privacy for AI.

### 3.1.6 Summary

Across all five focus areas—training/inference, data management, models/datasets, accessibility/usability, and security/privacy—the community has developed a substantial toolkit. However, **integration** remains a recurring challenge: researchers consistently request end-to-end workflows, from data ingestion and curation to final model deployment, that remain robust, reproducible, and secure.

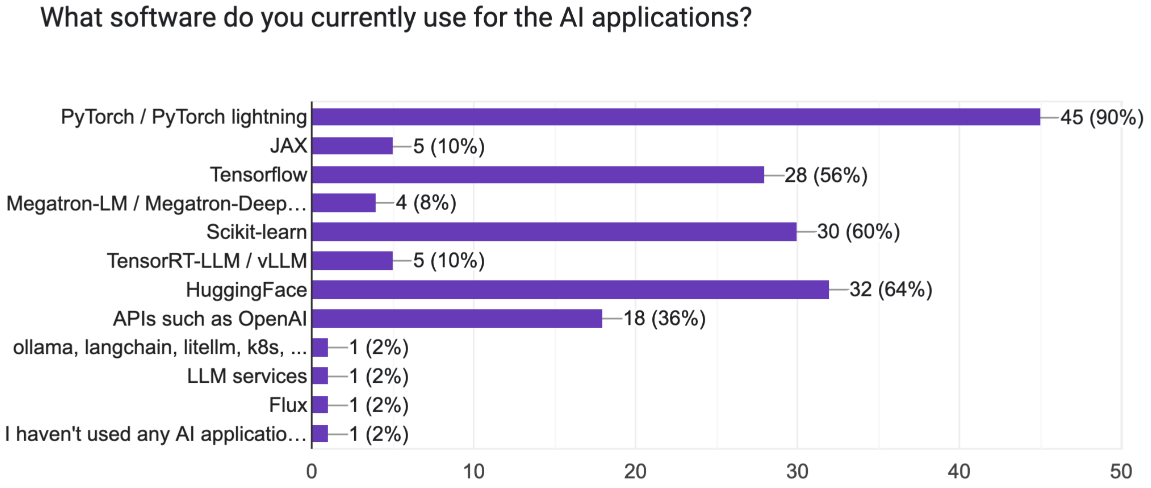
There is a growing **demand** for:

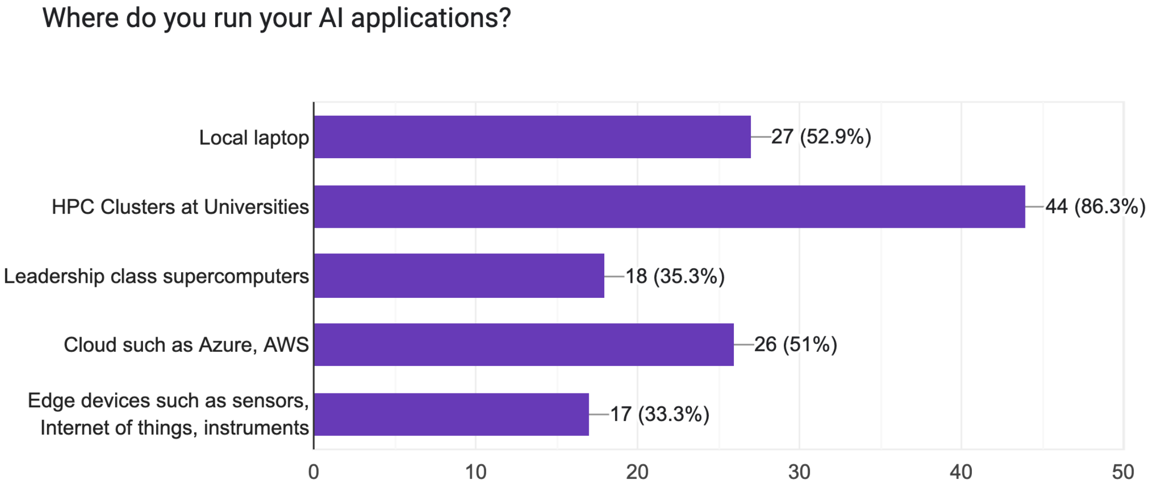
- **Better incentives** to share, document, and maintain reproducible solutions.
- **Robust frameworks** for privacy-aware and secure AI, including unlearning and continuous model updates.
- **Hybrid HPC–Cloud–Edge workflows** that meet domain-specific needs (e.g., agriculture, medical imaging).
- **Comprehensive education & facilitation** that lowers barriers to entry for domain experts.

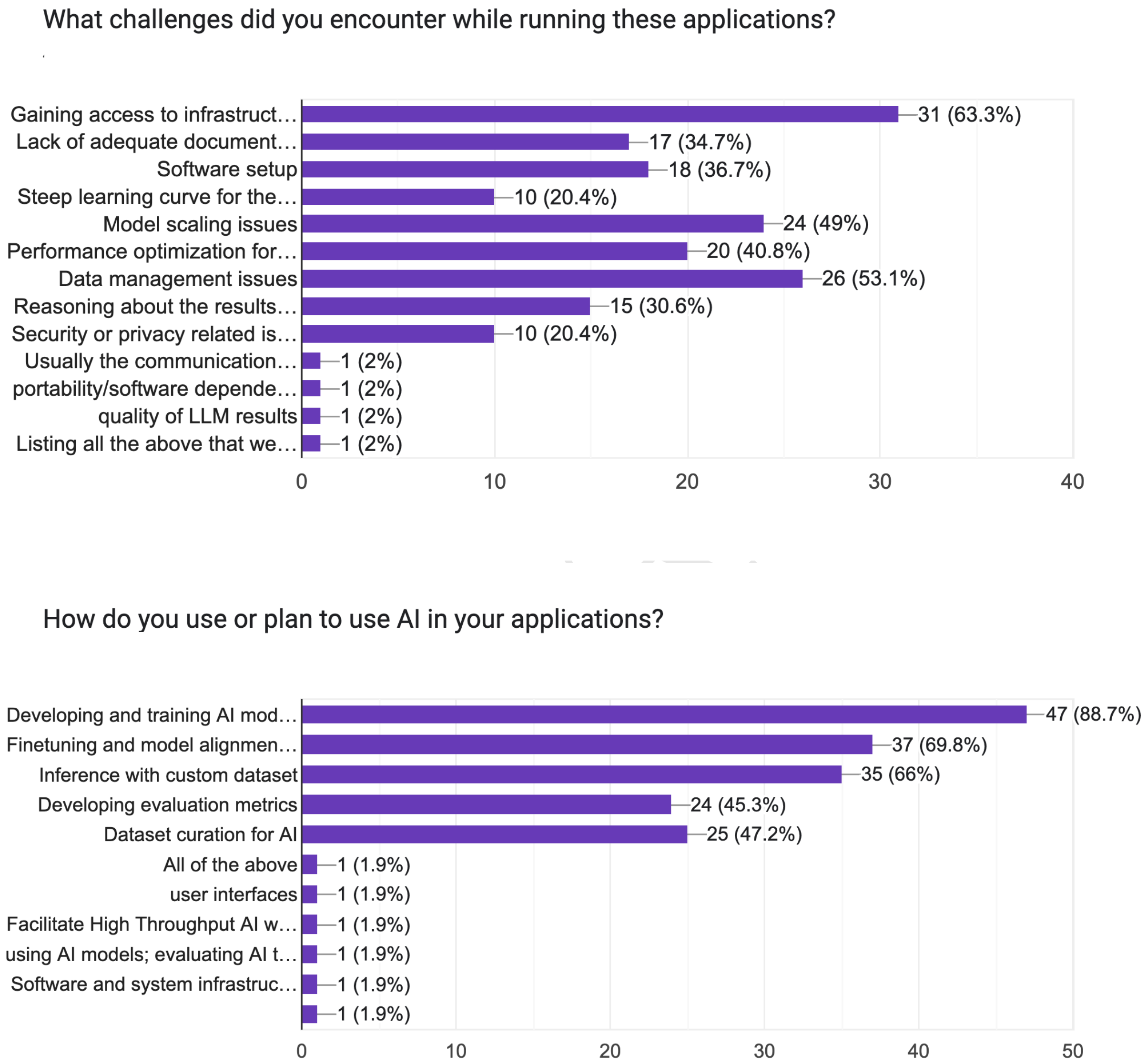
Finally, participants frequently highlighted that **NAIRR** (or any national-scale resource) should not *reinvent the wheel.* Instead, it should capitalize on existing industry-driven software and HPC solutions while focusing on the ***unique needs of science***, such as long-term data curation, specialized domain workflows, or novel compute architectures for emerging AI paradigms.

## 3.2 Workshop Survey

To gain insights into the attendees' perspectives on the workshop, a survey was distributed to capture key focus areas and guide meaningful discussions. This feedback also informed the design of the break-out sessions. The survey covered the AI software ecosystem, applications, deployment strategies, data challenges, infrastructure, and associated hurdles. It included a combination of multiple-choice, Likert-scale, and open-ended questions to gather quantitative and qualitative insights. A detailed summary of the survey results is provided in the Appendix (graphs to be added). Key observations from the survey highlighted the dominant use of AI across various scientific domains, including tasks such as model training, fine-tuning on custom datasets, inference, model alignment, and reasoning. The primary compute resources utilized included NVIDIA GPU-based accelerators and CPUs on university HPC clusters, leadership-class supercomputers, and cloud platforms like AWS and Azure. Popular AI software tools reported by attendees included PyTorch, TensorFlow, Scikit-learn, Hugging Face libraries, and APIs such as OpenAI. There was significant interest in understanding the performance of AI software with appropriate tooling support. Finally, responses to a question about attendees' expectations for the workshop provided several constructive suggestions. Key themes included exploring the landscape of open-source AI software, addressing deployment challenges on various systems, ensuring access to scalable infrastructure, promoting reproducible and trustworthy AI practices, and increasing training opportunities.







What challenges did you encounter while running these applications?

How do you use or plan to use AI in your applications?

Figure 1: NAIRR workshop attendee survey insights

# 4 Insights and Emerging Ideas from Workshop

The workshop concluded by bringing together leading experts in artificial intelligence (AI), high-performance computing (HPC), and data science to explore software-related strategies and governance considerations for the National Artificial Intelligence Research Resource (NAIRR). The panel included specialists with backgrounds in security and governance, data lifecycle and accessibility, AI-driven interfaces, and HPC-AI integration. Their collective goal was to determine best practices, key priorities, and actionable steps for advancing NAIRR's software ecosystem.

A central theme of the session was the importance of **user-centric design**. Panelists emphasized that NAIRR's software stack must lower barriers to AI adoption for diverse communities, particularly those without extensive technical backgrounds. Several participants proposed the rapid development of a ***NAIRR-GPT***—a chatbot interface leveraging large language model (LLM) technology to guide new users. This interface would enable researchers to articulate scientific problems in plain language and receive tailored recommendations for relevant models, data sources, and computing resources.

However, panelists cautioned against a "one-size-fits-all" approach. Given AI's rapid evolution, the NAIRR must remain flexible and agile, allowing domain experts to integrate emerging tools without overhauling the entire infrastructure. Participants agreed that the NAIRR should initially focus on a smaller, well-defined scope—such as educational pilots and foundational user support—before scaling to more complex features or broader user bases.

**Governance and security** emerged as paramount considerations. One panelist highlighted the need to define clear roles, responsibilities, and oversight mechanisms, particularly regarding software development pipelines, hardware/software supply chains, and data usage agreements. From the beginning, NAIRR must incorporate robust security measures to protect software integrity, user data, and the research process.

The panel also explored the **intersection of HPC and AI**, noting that traditional HPC workloads differ significantly from AI inference and training tasks. Experts underscored that software systems must accommodate both domains, potentially requiring new instrumentation, monitoring, and resource management layers. Attendees further discussed **community-driven innovation**, pointing to hackathons, targeted workshops, and pilot programs as effective ways to foster collaboration among HPC experts, AI researchers, and domain scientists.

Throughout the discussion, panelists stressed the value of **leveraging existing software tools** rather than reinventing them. One contributor recommended identifying tried-and-true solutions in data management, model training, and collaborative analytics and refining these tools for NAIRR's unique needs. In the process, NAIRR could fill gaps by supporting new features or ensuring compatibility across a broad spectrum of user requirements.

The panel converged on several core recommendations. First, **prioritize user accessibility and rapid onboarding** through initiatives like NAIRR-GPT. Second, **structure governance and security** from the ground up, clarifying vendor responsibilities and ensuring robust software pipelines. Third, **focus on a manageable initial scope**, then scale the resource as AI capabilities and community needs evolve. Finally, **promote continuous community engagement**—through hackathons, surveys, and cross-domain collaborations—to keep pace with the fast-changing AI ecosystem. These guiding principles will be the foundation for an inclusive, sustainable, and forward-looking NAIRR software environment.

# 5 Conclusions and Recommendations

Some of the main conclusions and recommendations of the workshop (in no particular order) are as follows:

1. The NAIRR Pilot's software stack will leverage both existing and emerging software solutions, catering to a range of users (from novices to experts) across diverse hardware platforms and accelerators, including those used for education. Notably, the HPC community views these solutions as a layered ***software stack*** optimized for performance and scalability. In contrast, the AI community often refers to a broader ***software ecosystem*** that tightly integrates data, user support, and training frameworks. The NAIRR effort must bridge these perspectives to serve all stakeholders effectively.

2. The stack must respond to the evolving needs of the scientific and AI communities, including real-time data analysis, privacy and security challenges, and portability across emerging AI hardware. Data management, which encompasses cleaning, curation, and annotation, to ensure that researchers can fully leverage the growing volume of diverse datasets.

3. The following software components for the NAIRR stack, must remain flexible and extensible to accommodate future technology advances:
    - Operating systems,
    - Middleware solutions for communication and resource management,
    - Languages and compiler support (with emphasis on Python, Julia, C, C++, and Fortran),
    - Workflow managers and AI-related libraries/models/frameworks.
4. Embracing open-source development and ensuring support for new hardware will be essential for keeping the NAIRR stack at the forefront of technological advances.
5. The stack should offer easy-to-use interfaces (e.g., Jupyter Notebooks, web-based platforms) to lower the barrier for newcomers to AI. At the same time, training and continuous user support cannot be separated from the software itself, underscoring the need for educational resources and dedicated guidance to help new users navigate this technology.
6. The stack must address near-term user-support needs; for instance, funding small supplements for current grantees during the NAIRR Pilot was proposed. Attendees also recommended creating intuitive *chatbot* interfaces to help users interact with the software stack, further reducing barriers to adoption and ensuring efficient troubleshooting and assistance.

## Acknowledgments

# A Workshop Participants

Names in **bold** are **technical committee members**, session leads$^{*}$, speakers$^{+}$ and session scribes$^{\#}$.

- Ryan Adamson - Oak Ridge National Laboratory
- Marian Adly - U.S. Department of Veterans Affairs
- **Ilkay Altintas**$^{*}$ - San Diego Supercomputer Center
- Katie Antypas - National Science Foundation
- Troy Arcomano$^{+}$ - Argonne National Laboratory
- Aldo Badano$^{+}$ - FDA DNA HIVE
- David Balenson - USC/ISI
- Purushotham Bangalore - National Science Foundation
- Shivam Barwey$^{\#}$ - Argonne National Laboratory
- Nate Bastian - DARPA
- Karlo Berket - Lawrence Berkeley National Laboratory
- **Wahid Bhimji**$^{*}$ - Lawrence Berkeley National Laboratory
- Anoushka Bhutani - University of Michigan
- Brian Bockelman - Morgridge Institute for Research
- Nicolae Bogdan - Argonne National Laboratory
- Carl Boettiger$^{+}$ - University of California, Berkeley
- Zechun Cao - Texas A&M University – San Antonio
- Giuseppe Cerati - Fermi National Accelerator Laboratory
- Dhruv Chakravorty - Texas A&M University
- Kyle Chard - University of Chicago
- Vipin Chaudhary - Case Western Reserve University
- Haipeng Chen - William & Mary
- Yiran Chen - Duke University
- Matthew Cherukara$^{+}$ - Argonne National Laboratory
- Krishna Teja Chitty-Venkata$^{\#}$ - Argonne National Laboratory
- Sajal Dash - Oak Ridge Leadership Computing Facility, ORNL
- Daniel DeFreez - Southern Oregon University
- Gautham Dharuman$^{+}$ - Argonne National Laboratory
- **Murali Emani**$^{*}$ - Argonne National Laboratory
- **Nicola Ferrier**$^{*}$ - Argonne National Laboratory / Northwestern University
- Sam Foreman - Argonne National Laboratory
- Sheikh Ghafoor - National Science Foundation
- Josh Greenberg - Sloan Foundation
- Anju Gupta - University of Toledo
- Salman Habib$^{+}$ - Argonne National Laboratory
- Shawn Haag - University of Minnesota
- Ben Hawks - Fermi National Accelerator Laboratory
- Amr Hilal - Tennessee Technological University
- Khalid Hossain$^{\#}$ - Argonne National Laboratory
- Shu Hu - Purdue Polytechnic Institute
- Xiaolei Huang - University of Memphis
- Tanzima Islam - Texas State University
- Ali Jannesari - Iowa State University
- **Shantenu Jha** - Rutgers University
- Krishna Kant - Temple University
- Anuj Karpatne - Virginia Tech
- Ian Kash$^{*}$ - University of Illinois Chicago
- Daniel S. Katz - University of Illinois Urbana-Champaign
- Kristopher Keipert - NVIDIA
- Duckbong Kim - Tennessee Tech University
- Farinaz Koushanfar - University of California, San Diego
- Ho-Joon Lee - Yale School of Medicine
- Juan Li - National Science Foundation
- Frank Y. Liu - Old Dominion University
- Miron Livny - University of Wisconsin–Madison
- Raghu Machiraju - Ohio State University
- Mahnaz Maddah - Broad Institute
- Amit Majumdar - San Diego Supercomputer Center
- Manil Maskey - NASA
- Kenton McHenry - University of Illinois Urbana-Champaign
- Lois Curfman McInnes$^{*}$ - Argonne National Laboratory
- **Diana McSpadden**$^{*}$ - Jefferson National Laboratory
- Bill Miller - National Science Foundation
- Pratik Mukherjee - University of California, San Francisco
- **Anita Nikolich**$^{*}$ - University of Illinois Urbana-Champaign
- Nwamaka Okafor$^{\#}$ - Argonne National Laboratory
- **Dhabaleswar K. Panda**$^{*}$ - Ohio State University
- **Michael E. Papka**$^{*}$ - University of Illinois Chicago
- Jasmin Phua - Datavant
- Marlon Pierce - National Science Foundation
- Elena Pourmal - LifeBoat LLC
- Amina Qutub - University of Texas San Antonio
- David Rabson - Department of Energy
- Nick Rahimi - University of Southern Mississippi
- Subhashini Ramkumar - OpenMined
- Benedikt Riedel - University of Wisconsin
- Mike Ringenburg - Microsoft / Azure HPC & AI
- Jesse Roberts - Tennessee Tech University
- James Rondinelli - Northwestern University
- Varuni Sastry - Argonne National Laboratory
- Ranga Setlur - SUNY Buffalo
- Shilpika$^{\#}$ - Argonne National Laboratory
- Leah Silen - NumFOCUS
- Carol Song - Purdue University
- Biplav Srivastava - University of South Carolina
- Hari Subramoni - Ohio State University
- Al Suarez - National Science Foundation
- **Nathan Tallent**$^{*}$ - Pacific Northwest National Laboratory
- Alastair Thomson - ARPA-H
- Jiachuan Tian - ESnet
- Mikhail Titov - Brookhaven National Laboratory
- Karen Tomko - Ohio Supercomputer Center
- Wen-Wen Tung - National Science Foundation
- Archit Vasan - Argonne National Laboratory
- Alex Wadell - University of Michigan
- **Feiyi Wang**$^{*}$ - Oak Ridge National Laboratory
- Yingfeng Wang - University of Tennessee at Chattanooga
- Jim Willenbring - Sandia National Laboratories
- Chandi Witharana - University of Connecticut
- Jianjun Xu - Amazon (AWS)
- Shinjae Yoo - Brookhaven National Laboratory
- Zhao Zhang - Rutgers University
- Huihuo Zheng - Argonne National Laboratory
- Michael Zink - University of Massachusetts Amherst
- Houlong Zhuang - Arizona State University

# B Workshop Agenda

## Day 1: December 3, 2024

- **08:00 - 09:00** *Breakfast and Check-in (TCS Conference Center)*
- **09:00 - 09:30** *Workshop Overview and Objectives (Room 1416, TCS Conference Center)*
  - 09:00 - 09:05 Introductions (DK Panda and Michael Papka)
  - 09:05 - 09:10 NAIRR Software Workshop Goals and Outcomes (Sheikh Ghafoor)
  - 09:10 - 09:30 NAIRR Overview (Katie Antypas)
- **09:30 - 10:30** *Domain AI Talks: AI in Practice, AI What is Missing (Room 1416)*
  - 09:30 - 09:40 Climate (Troy Arcomano)
  - 09:42 - 09:52 Medical (Aldo Badano)
  - 09:54 - 10:04 Instruments/Experiments (Mathew Cherukara)
  - 10:06 - 10:16 Biology (Gautham Dharuman)
  - 10:18 - 10:28 Environment (Carl Boettiger)
- **10:30 - 10:50** *Surveys (Room 1416)*
  - 10:30 - 10:40 NERSC Survey Results (Wahid Bhimji)
  - 10:40 - 10:50 NAIRR Software Survey Results (Murali Emani)
- **10:50 - 11:15** *Break*
- **11:15 - 12:30** *Breakout Session 1: Software Needs for NAIRR Pilot (Rooms 1416, 1404, 1405, 1406, 1407)*
  - Current Software, Tools, and Gaps - Training and Inference
  - Current Software, Tools and Gaps - Data Management and Storage
  - Current Models, Datasets and Gaps
  - Current Accessibility & Usability and Gaps
  - Current Software, Tools, and Gaps - Security & Privacy
- **12:30 - 13:00** *Summary of Breakout 1 (Room 1416)*
- **13:00 - 14:00** *Lunch (Networking)*
- **14:00 - 15:00** *Cosmology Meets AI: Roadmapping the Final Frontier (Speaker: Salman Habib, Room 1416)*
- **15:00 - 16:00** *Parallel Breakout Session 2: Key Software Features (Rooms 1416, 1404, 1405, 1406, 1407)*
- **16:00 - 16:15** *Break*
- **16:15 - 16:45** *Summary of Breakout 2 (Room 1416)*
- **16:45 - 17:00** *Day 1 Recap and Discussion (Room 1416)*
- **17:00 - 18:00** *Return Shuttles to Crowne Plaza*

## Day 2: December 4, 2024

- **08:00 - 09:00** *Breakfast and Networking (TCS Conference Center)*
- **09:00 - 09:15** *Welcome Back and Recap of Day 1 (Room 1416)*
- **09:15 - 10:45** *Parallel Breakout Session 3: Software Adoption & Deployment Challenges (Rooms 1416, 1404, 1405, 1406, 1407)*
- **10:45 - 11:15** *Break*
- **11:15 - 11:45** *Summary of Breakout 3 (Room 1416)*
- **11:45 - 12:45** *Parallel Breakout Session 4: Addressing Short- and Long-Term Objectives (Rooms 1416, 1404, 1405, 1406, 1407)*
- **12:45 - 13:15** *Summary of Breakout 4 (Room 1416)*
- **13:15 - 14:15** *Lunch (Networking)*
- **14:15 - 15:30** *Action Plan for Post-Workshop Process (Room 1416)*
  Panelists:
  - Ilkay Altintas
  - Wahid Bhimji
  - Nicola Ferrier
  - Anita Nikolich
- **15:30 - 16:00** *Break*
- **16:00 - 16:30** *Final Thoughts and Next Steps (Room 1416)*
- **16:30 - 17:00** *Closing Remarks and Adjournment (Room 1416)*
- **17:00 - 18:00** *Return Shuttles to Crowne Plaza*