

**MulticoreWorld 2026:  
(February 16-20, 2026)**

## **System Software Solutions for FugakuNEXT and Beyond**

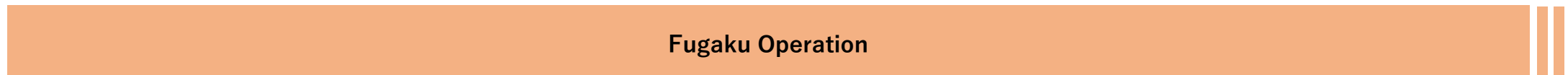
### **Kento Sato**

- Team Principal, High Performance Big Data Research Team, RIKEN R-CCS
- Unit leader, Data management platform development unit, AI for Science division, RIKEN R-CCS
- **Unit leader, Advanced HPC Technologies Development Unit, Next-Generation HPC Infrastructure Development Division, RIKEN R-CCS**
- Visiting professor, Kobe graduate university
- Visiting associate professor, Tohoku graduate university
- Adjunct lecturer, Osaka graduate university

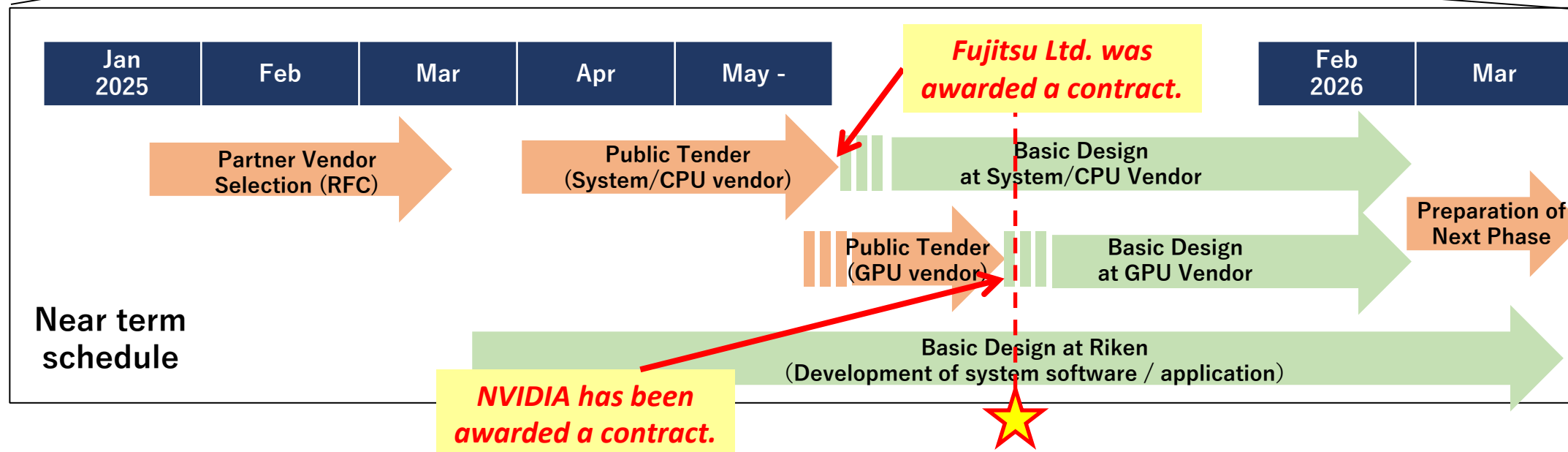
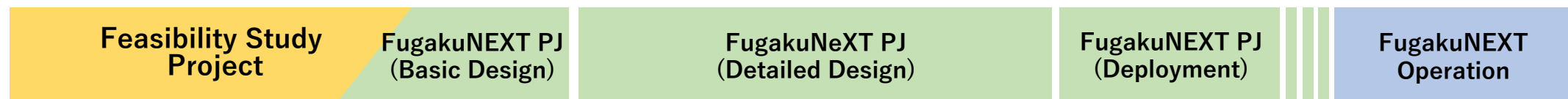
# Expected Timeline of Fugaku-NEXT R&D



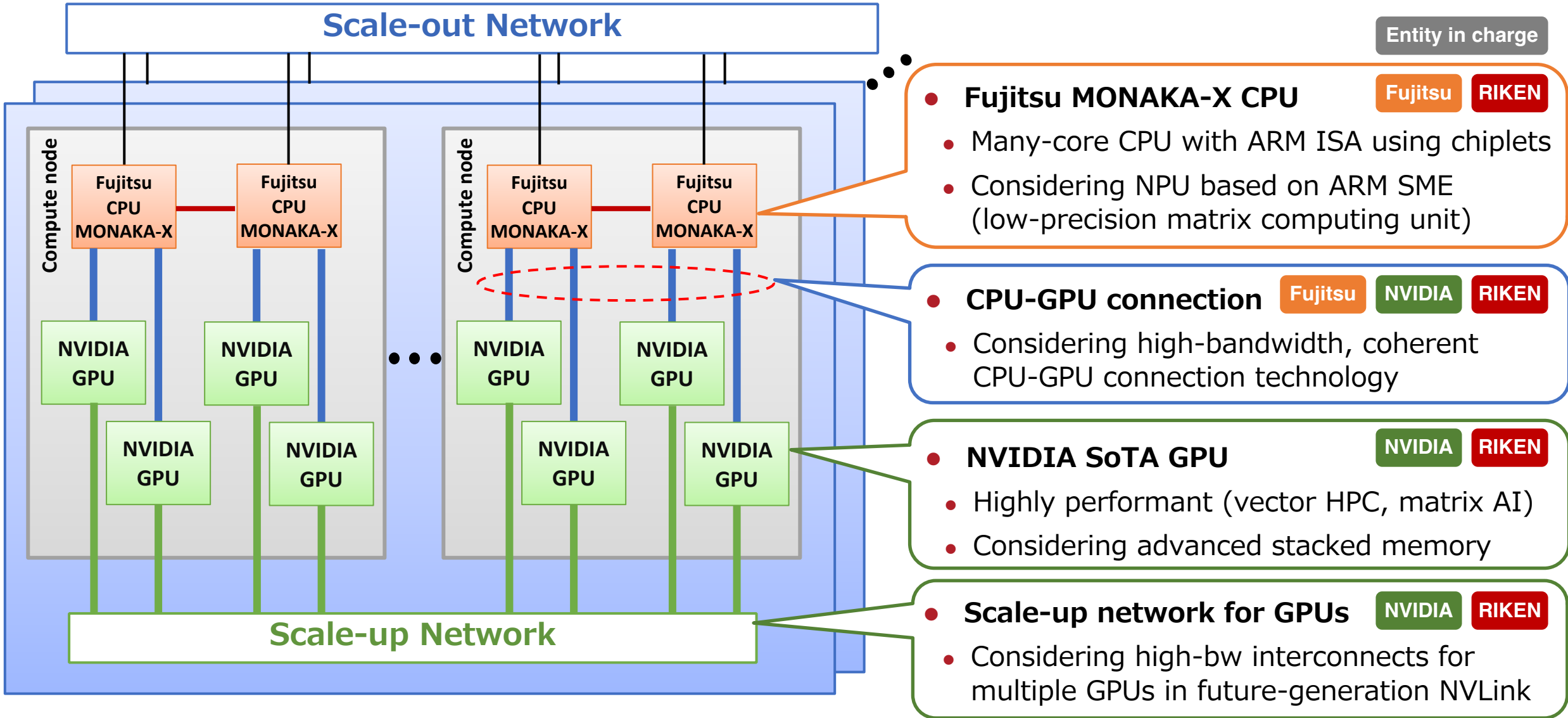
Fugaku



FugakuNEXT



# Overview of FugakuNEXT System Architecture



Detailed configurations and specs. will be determined in Basic design phase.

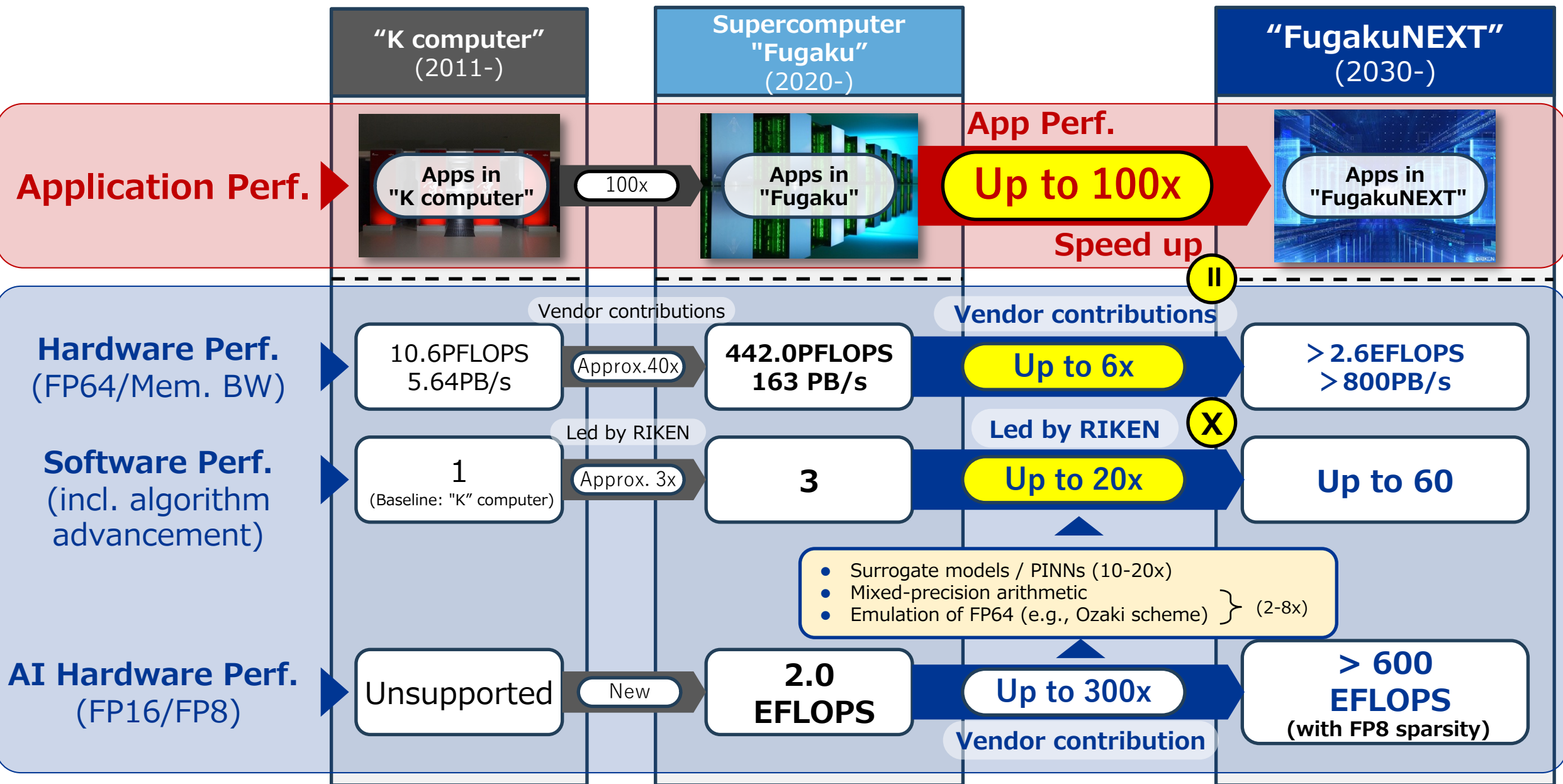
## Goal / Concept of FugakuNEXT

- **5 to 10x** higher effective performance than Fugaku for existing HPC applications
- **Effective performance of 50 EFLOPS or higher for AI processing** with **Zetta-FLOPS-scale** peak performance in mind
- **Target 10-100x improvement** for apps by combining simulation and AI

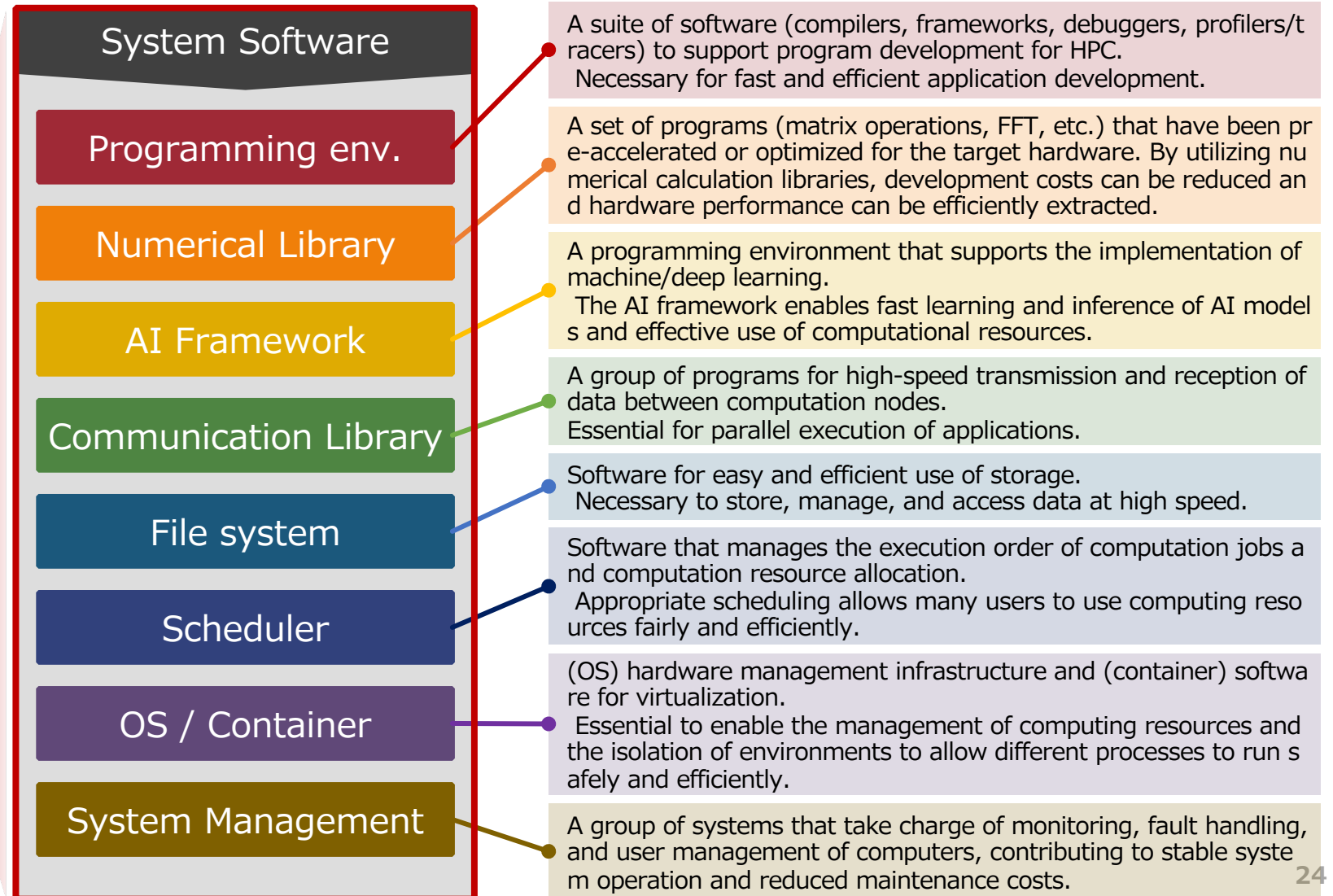
- **"Application First" philosophy** that prioritizes application performance – **"co-design"**
- **Heterogeneous architecture** combining power-efficient CPUs and bandwidth-oriented GPUs to achieve the goal under power constraint
- CPUs that **can utilize the assets such as application software** developed for "Fugaku"

Items	CPUs	GPUs	Fugaku	Improvements
Total node count	≥ 3,400		158,976	
Theoretical peak perf. of FP64 vector	≥ 48 PFLOPS	≥ 3.0 EFLOPS	488-537 PFLOPS	x5.7~
Theoretical peak perf. of FP16/BF16 matrix	≥ 1.5 EFLOPS	≥ 150 EFLOPS	1.95-2.15 EFLOPS	x70.5~
Theoretical peak perf. of FP8 matrix	≥ 3.0 EFLOPS	≥ 300 EFLOP	—	
(Sparsity considered FP8 matrix)	—	≥ 600 EFLOPS	—	
Main memory capacity	≥ 10 PiB	≥ 10 PiB	4.85 PiB	x4.1~
Main memory bandwidth	≥ 7 PB/s	≥ 800 PB/s	163 PB/s	x4.9~
System power consumption	40 MW (compute node and storage)		about 30 MW	

# Road to Achieving up to 100x Application Performance

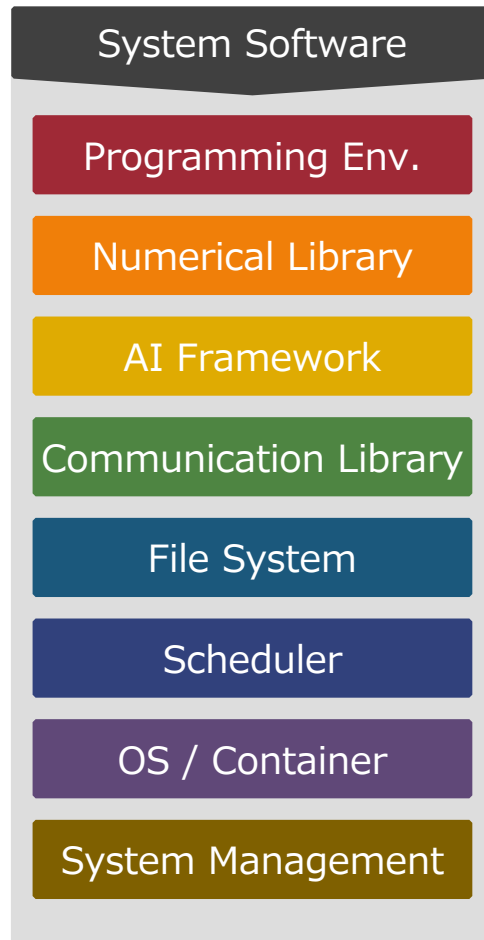


- System software refers to a set of software that maximizes the performance of computer hardware, streamlines the development of applications such as scientific simulation and AI, and also ease the operation of computers



# Strategy for System Software Development in FugakuNEXT

We adopt OSS as much as possible for sustainable maintenance. We also release software developed in this project as OSS and contribute to the community. In addition, if necessary, we adopt commercial/proprietary software that are expected to have sustainable development/improvement



- **High compatibility and continuous software improvement**

- Based on lessons learned from Fugaku, we develop system software that can be continuously developed and improved through OSS
- Build a software ecosystem by utilizing OSS in collaboration with domestic and international communities
- Enables seamless transition from "Fugaku" to "FugakuNEXT" for application users via Fugaku OnDemand ("Cloudified" Fugaku)

- **Deployment to cloud and domestic/international computing platforms**

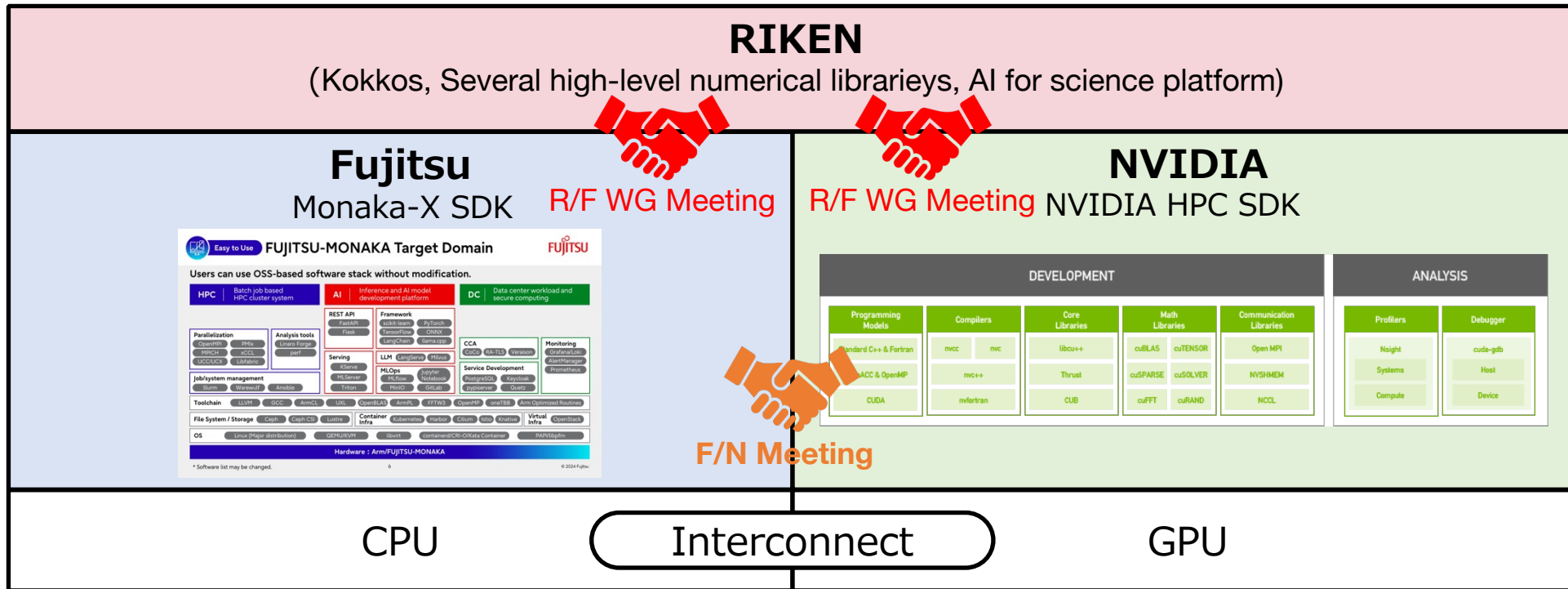
- Provide software environment by packaging HPC environment via Spack/Container
- Expand Virtual Fugaku ("Fugaku-fied" Cloud) environment deploy a software environment on the cloud that provide equivalent environment to the FugakuNEXT system

## Fugaku Open OnDemand

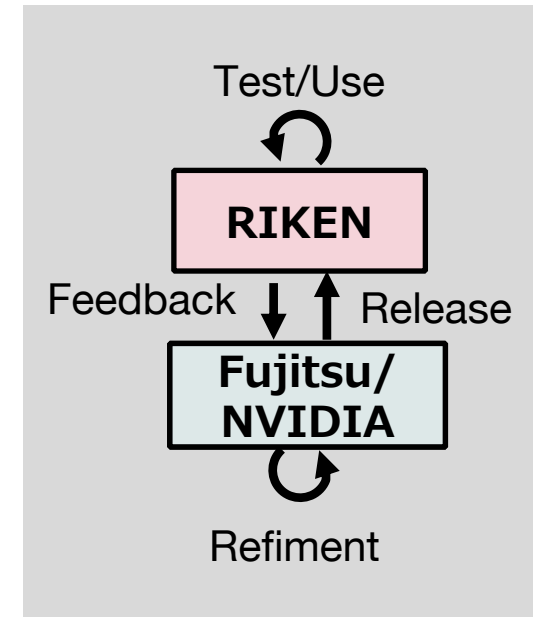


# Collaborations in System Software Development

- Fujitsu/NVIDIA will develop SDK for Monaka-X/SToA GPU
- RIKEN will develop system software on top of Monaka-X/NVIDIA HPC SDK
  - Kokkos, Several high-level numerical libraries, Scale-up/out performance analysis, AI for science platform (e.g., Multi-agent system, vibe coding platform)



## Collaboration loops

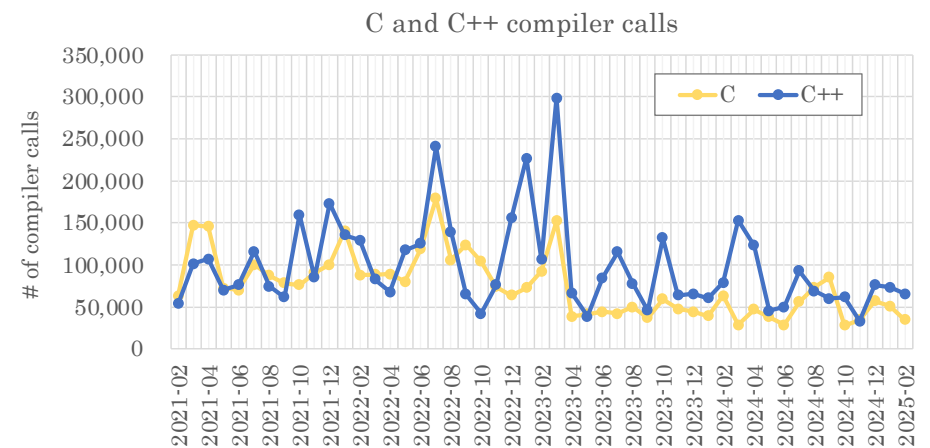
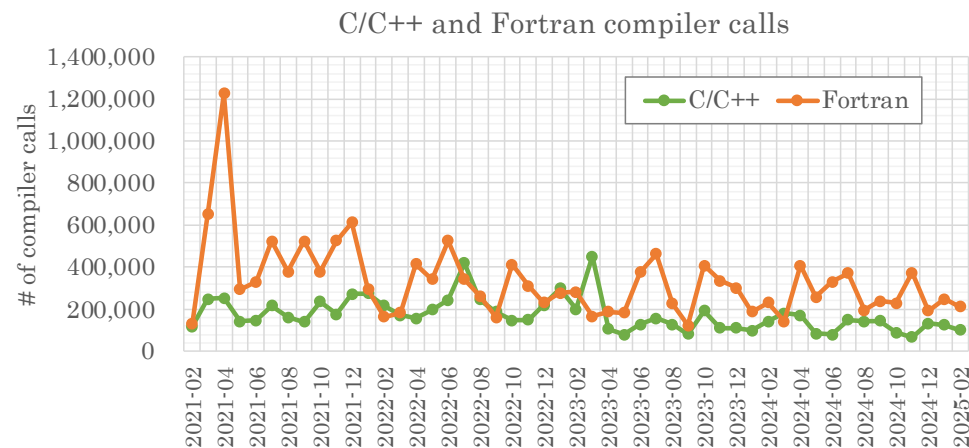


# FugakuNEXT System Software Development Goal: Programming Environment WG

## Programming Environment WG

- Support the majority of existing HPC applications in Fugaku and programs implemented in the following languages and framework environments
  - Languages: C/C++, Fortran, Python (and others)
  - Frameworks: OpenACC, OpenMP
- Provide better development tool environment (e.g., debugger, profiler) than Fugaku
- Support frameworks to efficiently work across different CPUs/GPUs with high compatibility
  - Kokkos, StdPar, SYCL, etc.

### Compiler Usage in Fugaku



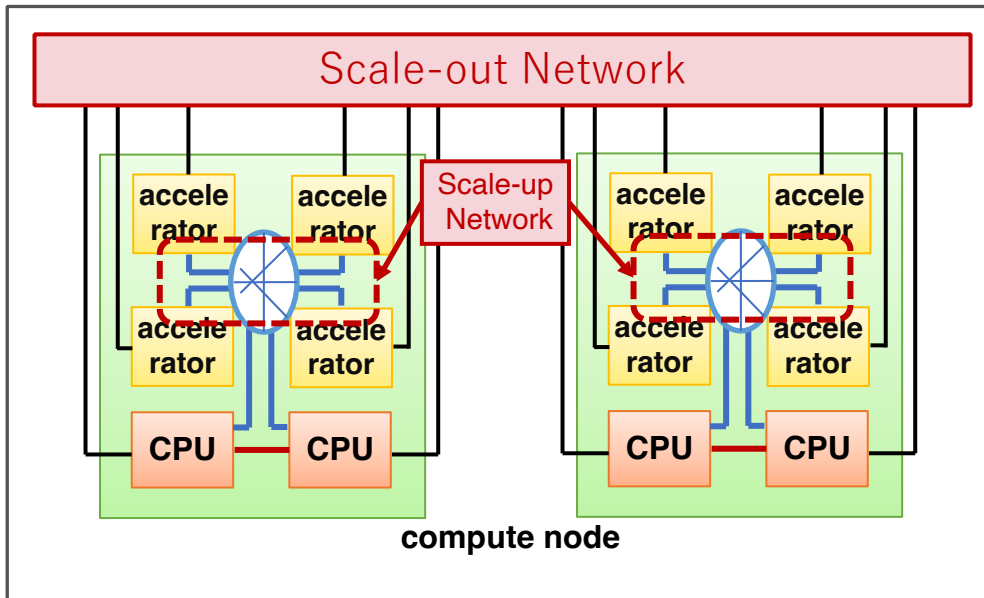
## Numerical Computing Library and Middleware WG

- Develop numerical libraries that efficiently use hardware and accelerate applications
  - Accelerate applications requiring high precision by utilizing low precision arithmetic units based on emulation techniques (e.g., OZAKI-Scheme).
  - If possible, we consider to apply emulation techniques to other cases other than matrix operations
- Middleware supported by Fugaku should also be supported by FugakuNEXT depending on the needs/seeds

Items	CPUs	Accelerators	Fugaku	Improvement
Total node count	≥ 3,400		158,976	
Theoretical peak perf. of FP64 vector	≥ 48 PFLOPS	≥ 3.0 EFLOPS	488-537 PFLOPS	x5.7~
Theoretical peak perf. of FP16/BF16 matrix	≥ 1.5 EFLOPS	≥ 150 EFLOPS	1.95-2.15 EFLOPS	x70.5~
Theoretical peak perf. of FP8 matrix	≥ 3.0 EFLOPS	≥ 300 EFLOP	—	
(Sparsity considered FP8 matrix)	—	≥ 600 EFLOPS	—	
Main memory capacity	≥ 10 PiB	≥ 10 PiB	4.85 PiB	x4.1~
Main memory bandwidth	≥ 7 PB/s	≥ 800 PB/s	163 PB/s	x4.9~
System power consumption	≤ 40 MW (compute node and storage)		about 30 MW	

### Communication Library WG

- Support communication libraries deployed on Fugaku and also following libraries
  - MPI, UCX/UCC, etc.
- Develop system software to efficiently use hierarchical networks with scale-up (e.g., UALink, NVLink) and scale-out (e.g., Ultra Ethernet & InfiniBand) networks so that existing frameworks and applications can efficiently run in FugakuNEXT



- Computational node with CPU and acceleration unit connected
  - CPU compatible with "Fugaku" at the binary level
  - Accelerated part of the arch based on GPUs with a proven track record of utilization in large scale supercomputers

- **System network suitable for both strong and weak scaling**
  - **Combined Scale-up/Scale-out network**
  - **Direct network link connection between accelerators**

- Tens of thousands of accelerator sockets throughout the system

## AI Software WG

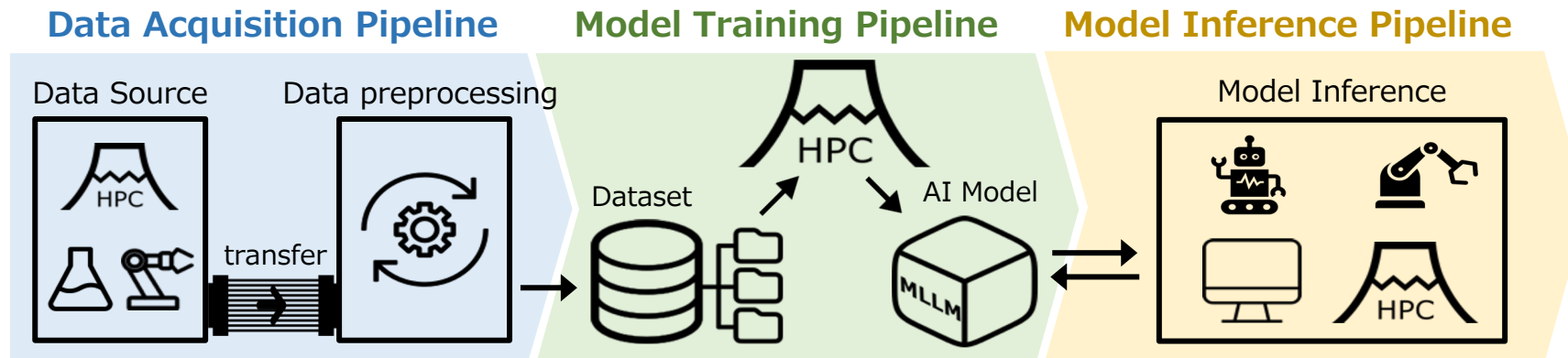
- Support individual AI software which is widely used in 2029-30 and tune to efficiently run on FugakuNEXT
  - Including SoTA collective communication libraries: namely xCCL, including NCCL, RCCL, HCCL, oneCCL, MSCCL, ACCL
- Develop AI and data processing pipelines/workflows that seamlessly integrate individual AI tools to automate a wide range of AI-for-Science research tasks.

- **Acceleration of AI for Science by FugakuNEXT**

- (1) **Data acquisition pipeline**: Fast generation of training data, fast transfer of various scientific data
- (2) **Model training pipeline**: Large-scale pre-training, post-training, and fine-tuning of AI models
- (3) **Model inference pipeline**: Automated workflow for scientific experiments by AI

- **"AI for Science" by integrating AI with simulations and scientific experiments**

- Automation of generation of simulation program code, experiments, data analysis of results and proposal of the next steps
- AI models are also self-reinforced to improve AI for Science research cycle



## Benchmark WG in App. Dev. Area

### App. Dev. Area

#### Application Community

- Application Development Area
- HPCI Consortium
- CPU/GPU Vendors
- HAIRDESC
- Other domestic and international universities and research institutions

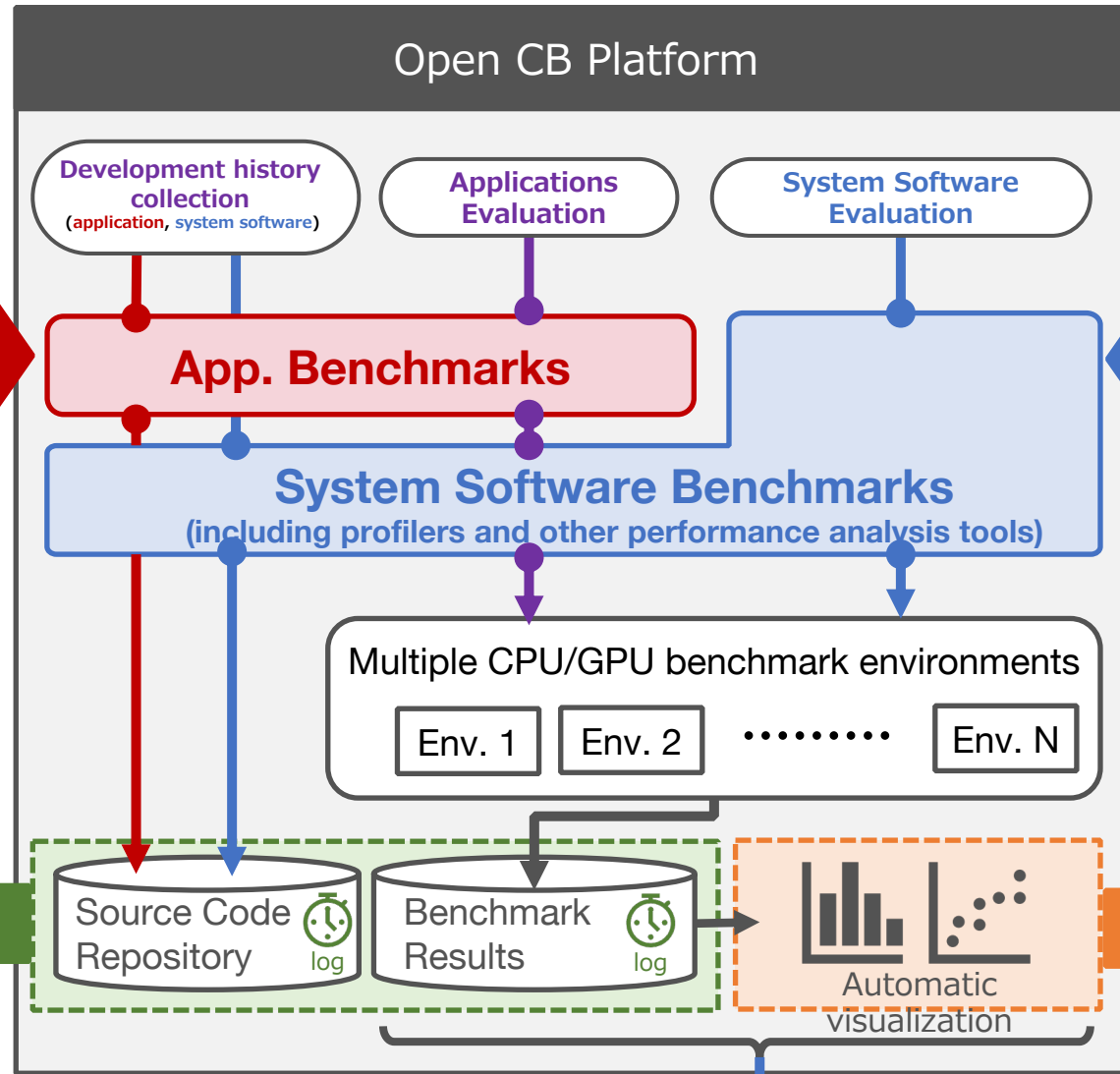
### AI for Science Div. or AI soft WG

#### AI (for Science) Factory

- **AI for general supports:** General application development and execution support with AI chatbots
- **Advanced development supports:** Program generation, code conversion, and auto-tuning by AI

Vibe coding Platform

Benchmarks



Benchmarks

### System Software & Operations Area

#### Systems software community

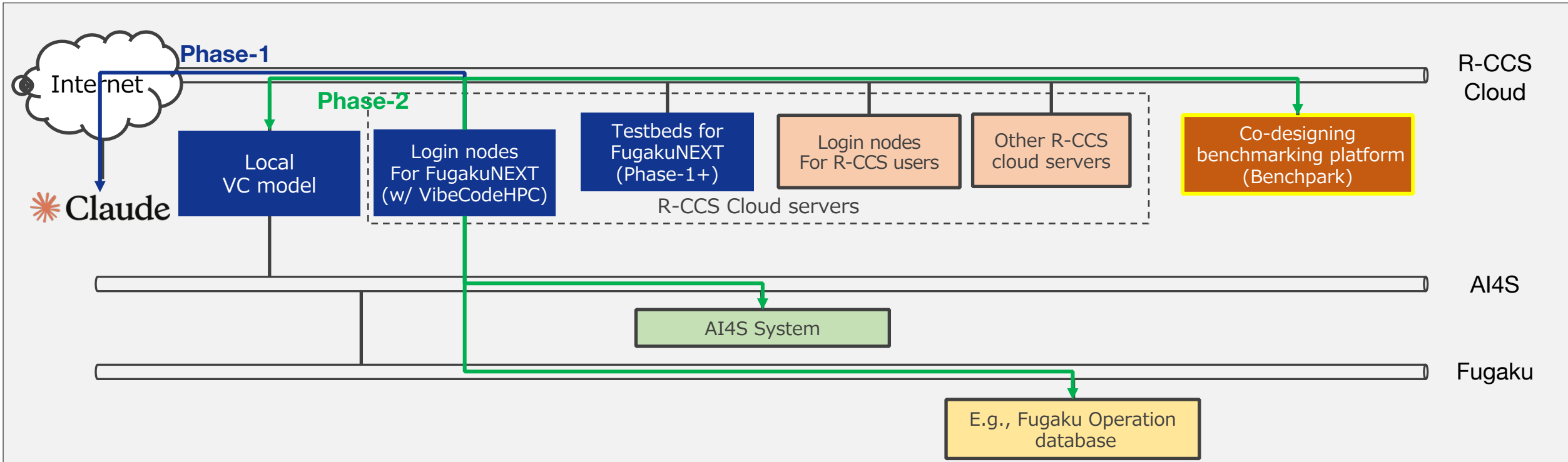
- System Software Area & Operation Area
- CPU/GPU Vendors
- AI for Science Platform Division
- Quantum HPC Collaboration Platform Division
- Other domestic and international universities and research institutions

#### Architecture Area

#### Open evaluation results in public

- $N = N_1 \times N_2 \times N_3 \dots$
- $N_1$  : CPU/GPU Environment
  - $N_2$  : Problem Size
  - $N_3$  : Configurations (compiler, library, version)
  - ...

**Feedback of evaluation results to the app/system software community and architecture**



- **Vibe Coding environment**

- **Phase-1 genAI: Claude Code**

- **Phase-2 genAI: Claude Code + Local Models**

- Open-source or self-developed Vibe coding (VC) AI models in the local environment (e.g., Kimi, ChatHPC, Fortran2CPP etc.)
- Integration to AI4S system
- Connect with MCP servers (e.g., Fugaku operation database, **Open CB platform**)

# AI-for-Science Supercomputer Overview [1]

## ■ Background and Purpose

- New computational infra. to advance "AI for Science" through fusion of AI & ModSim
- Accelerating scientific research through world-leading AI performance and interplay with Fugaku
- Dramatically accelerate research cycles by scientific foundation models

## ■ System Configuration

- **Nodes:** 400 nodes equipped with NVIDIA Grace Blackwell superchips (over 1,600 GPUs)
- **Interconnect:** NVIDIA InfiniBand XDR (up to 3.2 Tbps)
- **Performance:** Over 64.16 PFLOPS (FP64), Over 15.539 EFLOPS (FP8)
- **Rack:** Supermicro servers with hot-water cooling capability adopted, achieving both high performance and energy efficiency

## ■ Comparison with "Fugaku"

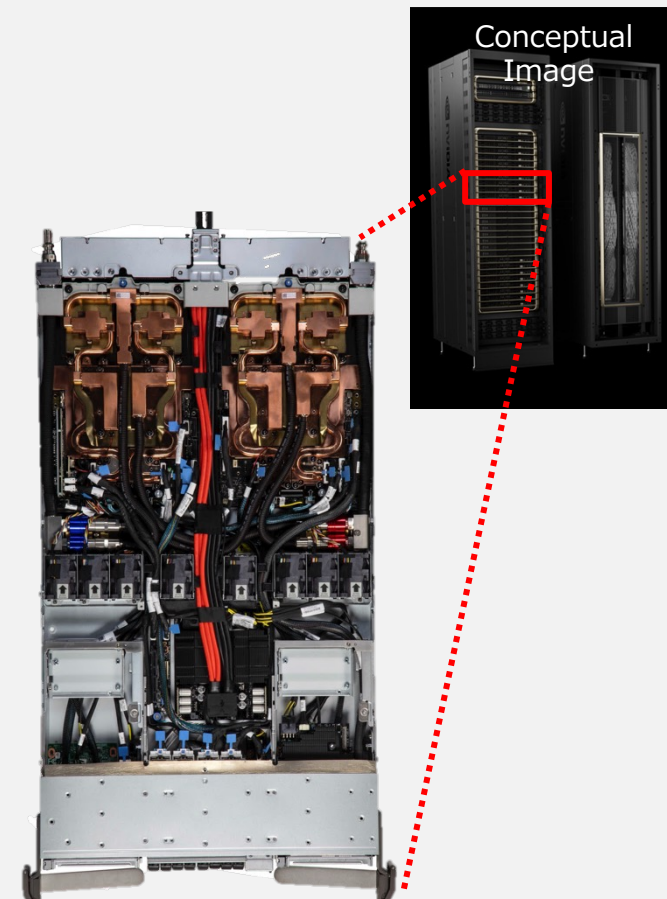
- FP8 computational performance is approximately 7.23 times that of Fugaku's low-precision (FP16) unit
- Collaboration with "Fugaku," which excels at FP64, enables AI×Science fusion research

## ■ Future Plans

- Installation by fiscal year 2025, full-scale operation starting early fiscal year 2026
- Expected to develop and accelerate collaboration with institutions like the U.S. Argonne National Laboratory (MOU signed in 2024) and others

## AI Inference Performance

15.539 ExaFLOPS or higher (FP8 performance)





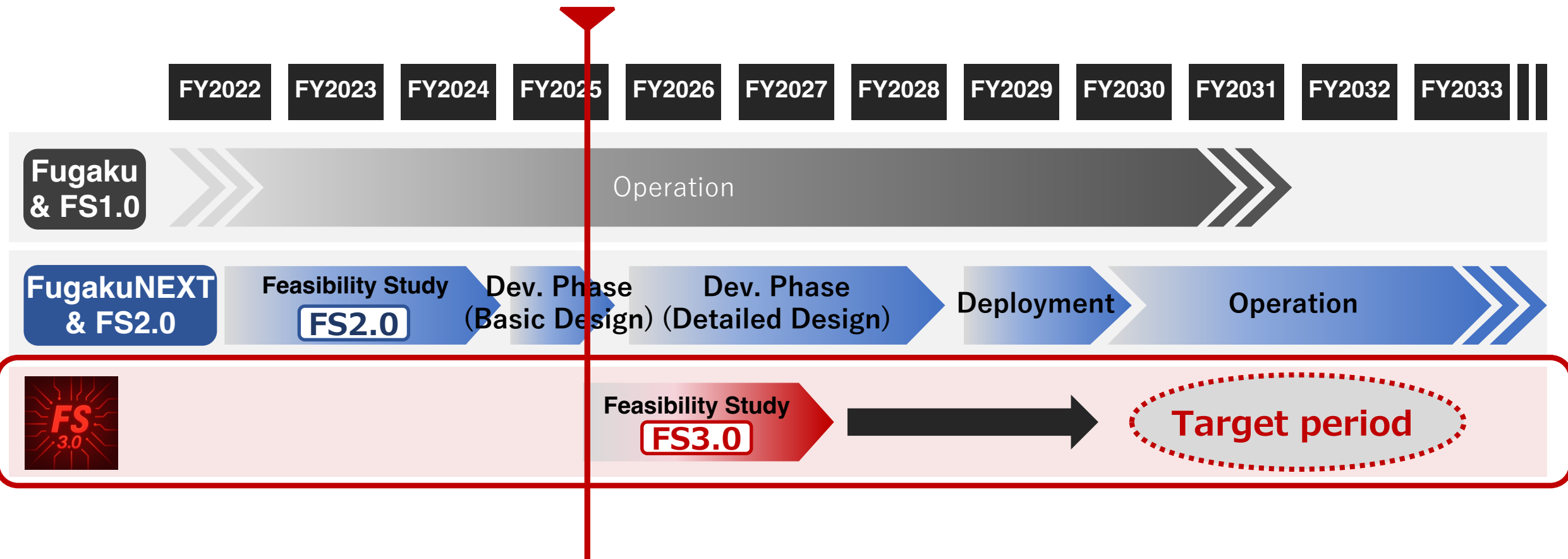
FS3.0 Web site

# **FS3.0: Feasibility Study on the future HPCI** (HPCI System Planning)

「HPCI整備計画調査研究」-- 運用システム整備計画調査研究

**PI: Kento Sato, RIKEN R-CCS**

**Co-PI: Kazuhiko Komatsu, Tohoku University**

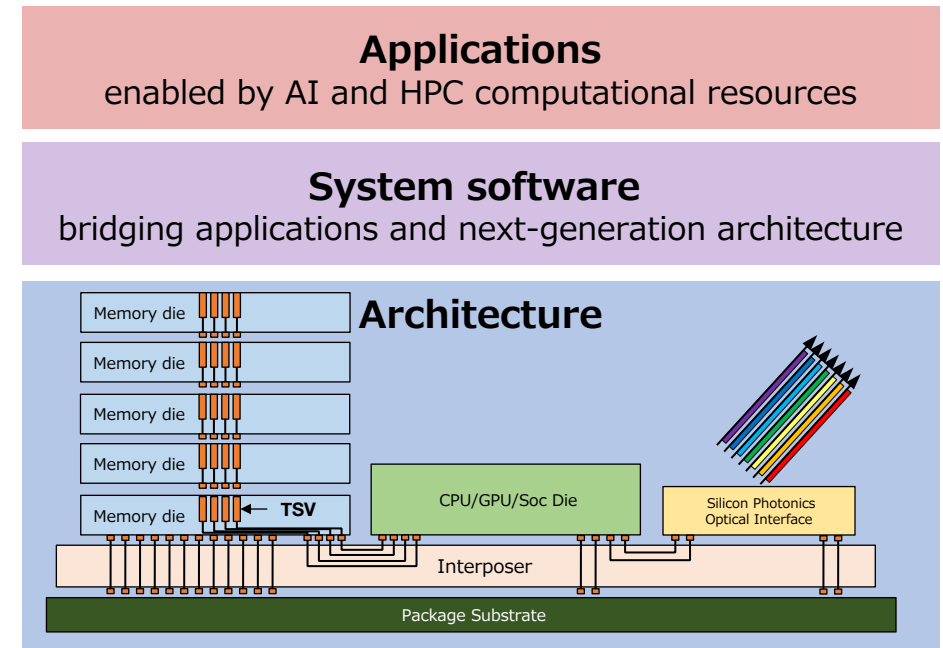


## Background

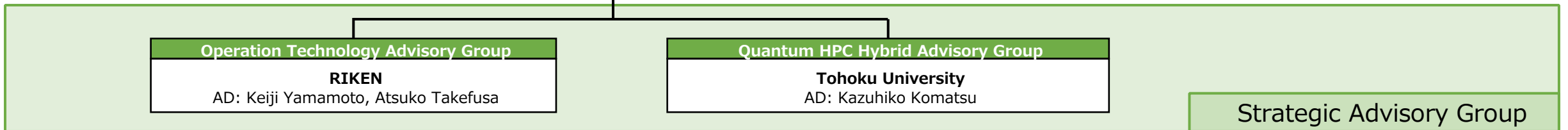
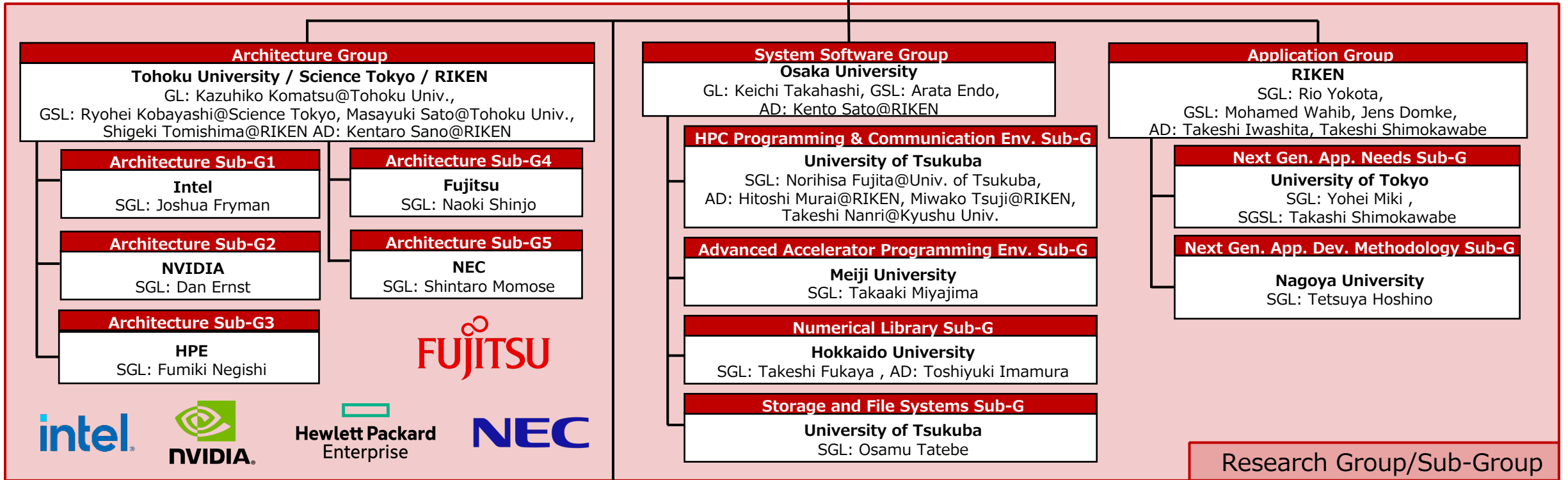
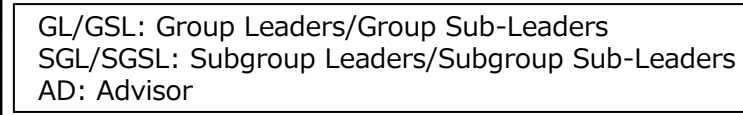
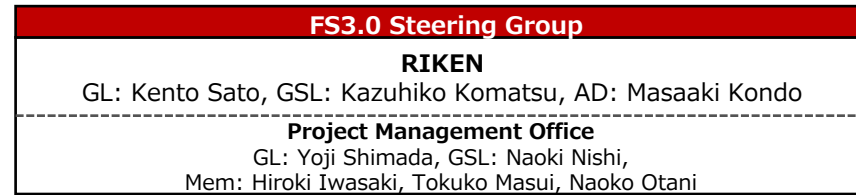
- The development of the next flagship system that will succeed the supercomputer Fugaku (hereinafter referred to as “FugakuNEXT”) and the enhancement of the HPCI (High Performance Computing Infrastructure for Innovation), which integrates diverse computational resources across Japan, have been progressing in parallel.
- In the FugakuNEXT era, **the rapid advancement of generative AI will significantly increase the importance of AI for Science, and the computational resources required to support such workloads are becoming increasingly diverse.**
- To meet these emerging requirements, the **FS3.0 project conducts feasibility studies on computer architecture, system software, and next-generation application development, and proposes procurement/development plans for HPCI systems in the FugakuNEXT era and beyond.** (FS1.0: Feasibility Study for Fugaku; FS2.0: Feasibility Study for FugakuNEXT; FS3.0: Feasibility Study for Systems Beyond FugakuNEXT.)

## Feasibility Study Areas

- **Architecture Feasibility Study**
  - In collaboration with participating vendors, we investigate advanced technologies such as **next-generation semiconductor technologies, 2.5D/3D/3.5D stacked packaging, and chip-to-chip optical interconnects**, and explore a range of possible system configurations for HPCI systems in the FugakuNEXT era.
- **System Software Feasibility Study**
  - Building on the software ecosystem inherited from Fugaku and FugakuNEXT, we will **define Japan’s system-software development positioning within the international community and examine system software that supports AI for Science** in addition to traditional large-scale simulations.
- **Application Feasibility Study**
  - We investigate trends in innovative application use and development enabled by generative AI—such as **automated computational science experiments and automatic code generation**—across diverse scientific domains, and **clarify the system requirements needed to enable such next-generation HPC applications.**



# FS3.0 Project Organization

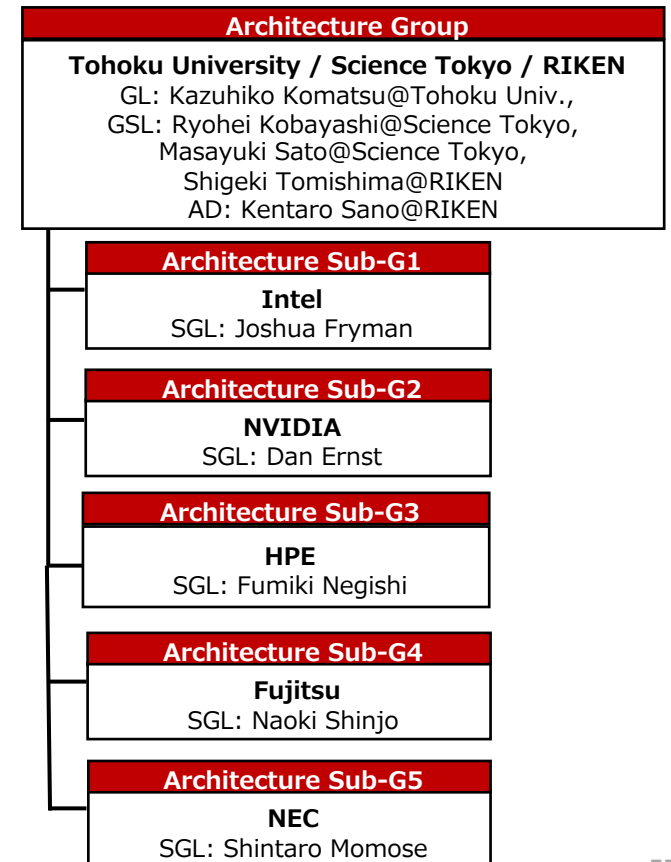


## Objectives

- Through a survey of technology and research trends related to architecture, the group will investigate and study the system requirements that the next-generation HPCI system suite should have for the development of Japan's computational infrastructure.

## Feasibility Study Areas

- Architecture Group (Tohoku University, Institute of Science Tokyo and RIKEN):** To conduct a comprehensive feasibility study on the next-gen. systems that meets the needs in HPC, **the architecture group collaborates with vendors and lead the investigation for elements of architectural technologies that are required for various applications such as AI for Science and large-scale scientific computation.**
- Architecture subgroup 1 (Intel Corporation):** Investigate and study various architectures based on current/future roadmaps for heterogeneous node configurations with Xeon CPUs and GPUs; evaluate chiplet configurations using advanced packaging technologies such as EMIB and Foveros; Study of CXL; Analyze cross-architecture programming environment based on oneAPI
- Architecture subgroup 2 (NVIDIA Corporation):** Provide our technical information on future architecture, system configuration, related software, etc. to maximize application and power performance, investigate elemental technologies, and study the possibility of utilizing domestic technologies
- Architecture subgroup 3 (HPE):** Study on system architecture (processor, memory, storage, I/O, etc.), OS, compilers, file systems, libraries, frameworks, development languages, and other related technical fields.
- Architecture subgroup 4 (Fujitsu Limited):** Study future available semiconductor manufacturing technologies (chiplet packaging technology, memory device technology, optical interconnect technology), study the roadmap of FUJITSU-MONAKA-X, and research on the target application domains and the system software for the target architectures and application domains
- Architecture subgroup 5 (NEC Corporation):** Study various architectural options and roadmaps, including heterogeneous node configurations with vector, scalar, accelerator, etc with the applicability to next-generation HPC, AI, quantum, etc. based on the characteristics of vector architecture (Trade-off among compute density, power saving, etc.). Study the direction of system architecture and system software development for efficient operation and utilization. Also, study key applications and benchmark performance targeted by HPCI systems in the FugakuNEXT era.

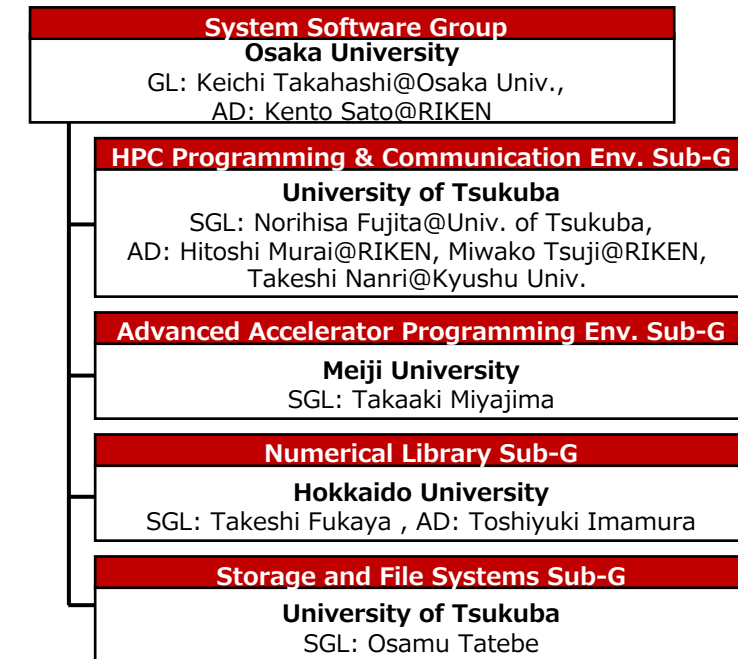


## Objectives

- We will investigate the system-software ecosystem that next-generation HPCI systems should provide and examine succession strategies for sustaining the software stack of Fugaku, FugakuNEXT, and systems beyond FugakuNEXT (ensuring **sustainable development**). In addition, based on technology trends in architectures and applications, we will study the software ecosystem required for the integrated operation of future HPCI systems across sites (enabling **cross-cutting deployment**).

## Feasibility Study Areas

- System Software Research Group (Osaka University):** Survey latest system-software trends and, together with the Archi and App groups, define requirements for next-generation hardware, while considering reuse and modernization of existing software assets.
- HPC Programming and Communication Env. Subgroup (University of Tsukuba):** Survey programming and communication environments for future HPCI systems, focusing on accelerator-oriented methods and needs from HAIRDESC, including emerging environments such as Kokkos and Julia.
- Advanced Accelerator Programming Env. Subgroup (Meiji University):** Investigate applicability of advanced AI chip (e.g., Cerebras, SambaNova) to HPC, comparing programming environments and identifying performance characteristics using representative HPC applications.
- Numerical Library Subgroup (Hokkaido University):** Provide guidelines for future numerical library development by reviewing next-generation architectures, application needs, and new mathematical techniques, including trends in mixed-precision and randomized algorithms.
- Storage and File System Subgroup (University of Tsukuba):** Study storage and file-system technologies for HPCI systems in the FutureNEXT era, identifying trends, issues, and requirements, and exploring methods to meet the requirements

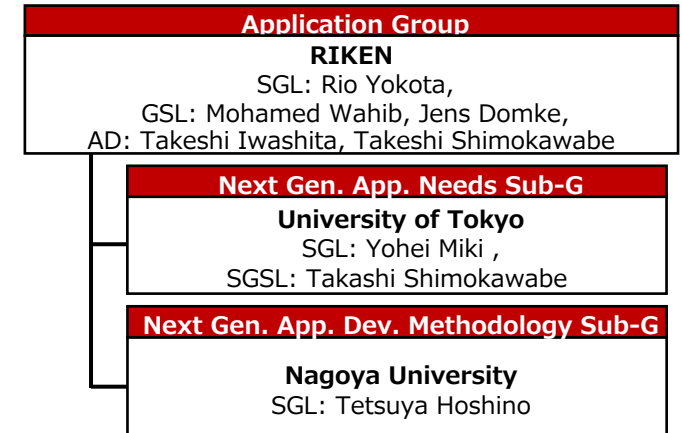


## Objectives

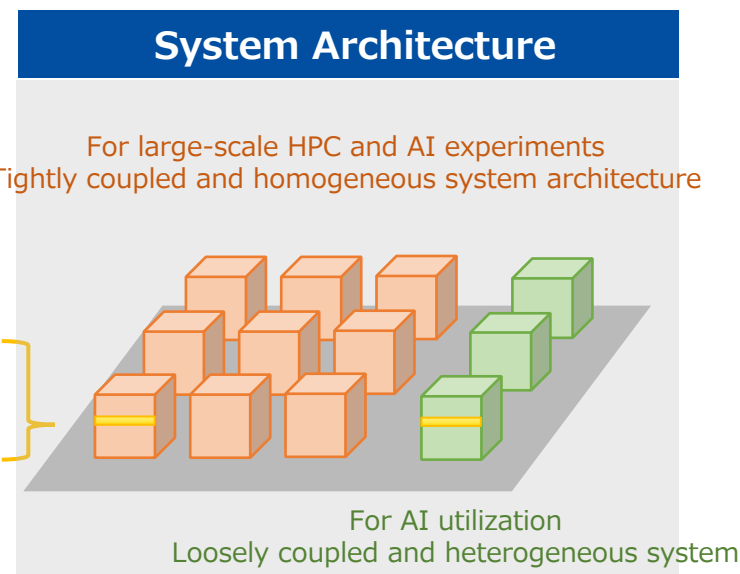
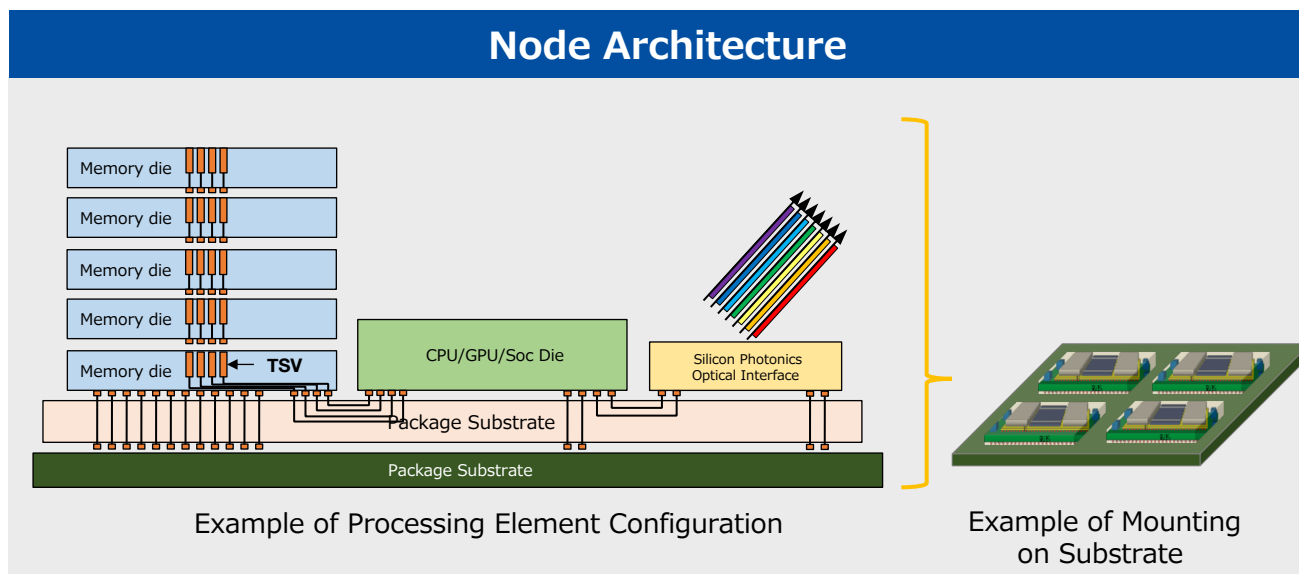
- We investigate trends in innovative applications that will drive future HPC utilization, aiming to clarify—from the application perspective—the functional and performance requirements needed for next-generation computing infrastructure. Looking toward 2030, we work with academic and industrial communities to define the vision for next-generation HPC applications.

## Feasibility Study Areas

- Application Research Group (RIKEN):** The group focuses on **four pillars: AI for Science, AI agents, automatic code generation, and automated computational experiments**. We study how advanced AI technologies can be integrated with scientific computing, including autonomous agent-based AI, automatic GPU-porting and code generation, and AI-driven experiment-automation workflows.
- Next-Generation Application Needs Subgroup (University of Tokyo):** The subgroup coordinates with HAIRDESC and the FugakuNEXT developers to **identify user needs for next-generation HPC applications**. These findings inform **system requirement definitions** for architecture and system software groups. To support broad adoption, the subgroup develops **representative benchmark codes** usable for performance evaluation and **as training material** for application developers and users, including NIS users. We analyze diverse programming approaches—from CUDA and OpenMP target to abstraction layers like SYCL and Kokkos—and explore **generative-AI-assisted development** in collaboration with the Next-gen. App. Dev. Methodology subgroup.
- Next-Generation Application Development Methodology Subgroup (Nagoya University):** We examine **LLM-based iterative development workflows**, where large language models propose, revise, and test code. Through this, we extract **system requirements needed to support efficient LLM-assisted development** on future HPCI systems. The subgroup evaluates both commercial LLMs (e.g., Claude Code) and **local deployment of domestic LLMs (e.g., Swallow)** for confidential applications. Evaluations use benchmark suites from the Next Gen. App. Needs Subgroup and key applications from HAIRDESC. We also assess **multi-agent LLM workflows**, clarifying the functions required for next-generation HPCI systems to effectively support such advanced development environments.



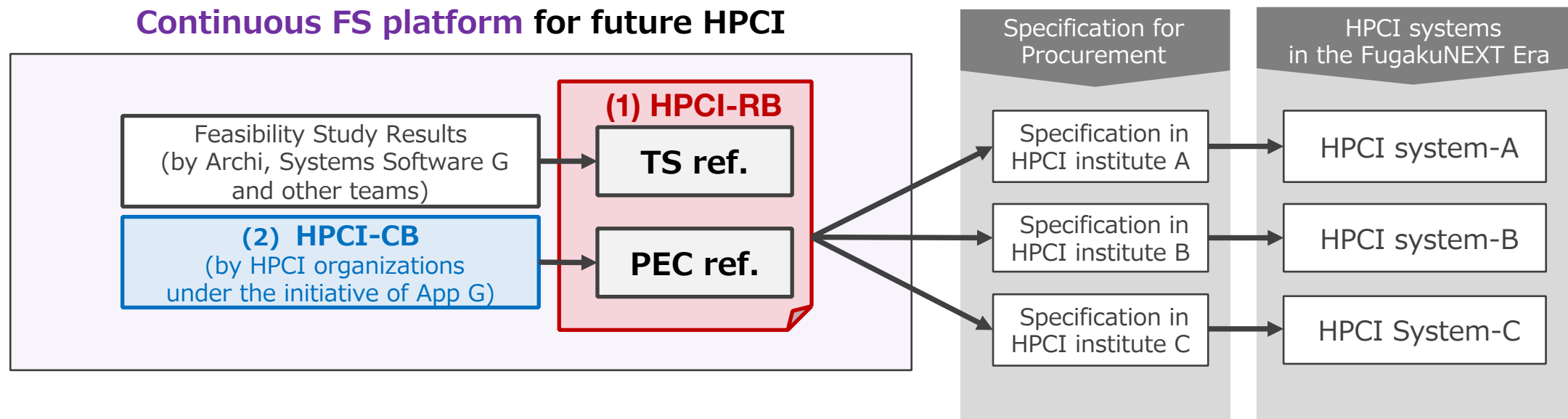
- Study of Architectures for Advancing Next-Generation HPC and AI Research
- Node Architecture
  - Heterogeneous systems: Explore heterogeneous node designs that combine domestic and international CPUs, GPUs, FPGAs, and AI accelerators, in addition to the FugakuNEXT architecture (domestic Arm CPU + GPU).
  - Next-generation memory hierarchies: Study configurations featuring large-capacity, high-bandwidth memory using HBM and CXL, along with hierarchical memory-access management.
  - Adoption of 2.5D/3D/3.5D packaging and chiplet technologies: Investigate the potential for high-density integration to alleviate data-movement bottlenecks and enable low-power operation (Collaborators: Tadahiro Kuroda, Atsutake Kosuge).
  - New computing principles: Assess the applicability of high-bandwidth, low-latency, and energy-efficient computing and communication technologies such as optical computing (e.g., silicon photonics) (Collaborator: Jun Shiomi).
- System Architecture
  - Large-scale HPC/AI systems: Supporting HPC simulations, AI processing, and their convergence at the scale of several thousand to ten thousand nodes, emphasizing high performance and efficiency through tightly coupled and homogeneous system architectures.
  - Small- to medium-scale AI-utilization systems: Consider flexible cluster or distributed environments capable of deploying large numbers of AI agents or distributed inference workloads, enabling diverse loosely coupled and heterogeneous system configurations, including cloud-like operational models.



- **Continuous FS Platform for future HPCI**

- Ensure long-term preservation and utilization of FS outcomes, including results and know-how from **HPCI-RB/HPCI-CB**
- **(1) HPCI-RB: HPCI Reference Blueprint**
  - Reference describing system requirements for future HPCI systems based on feasibility studies
  - **TS ref. (Tech. Spec. Reference)** : Provide references for architecture and system software roadmaps
  - **PEC ref. (Performance Evaluation Criteria Reference)**: Provide effective performance to be achieved by the target applications on systems proposed in TS ref.
- **(2) HPCI-CB: HPCI Continuous Benchmarking Env.**
  - Enables continuous performance evaluation of target applications and performance prediction in future HW

### Continuous FS platform for future HPCI





# R&D on AI for Science Platform

## Kento Sato

- **Team Principal, High Performance Big Data Research Team, RIKEN R-CCS**
- **Unit Leader, Data management platform development unit, AI for Science division, RIKEN R-CCS**
- Unit Leader, Advanced HPC Technologies Development Unit, Next-Generation HPC Infrastructure Development Division, RIKEN R-CCS
- Visiting Professor, Kobe graduate university
- Visiting Associate Professor, Tohoku graduate university
- Adjunct lecturer, Osaka graduate university

# Our mission: HPC, AI & Big Data Fusion

The mission of our team is to investigate state-of-the-art techniques for efficiently running large-scale applications on HPC systems. In addition to **fundamental R&D of system software in HPC**, our team conducts performance analysis and software development to accelerate deep learning and big data processing on HPC systems (**HPC for AI/BD**), while we also apply AI techniques to solve technical challenges in HPC (**AI/BD for HPC**)

## HPC for AI/BD

Research and software development for accelerating AI/Big data applications by techniques in HPC

## AI/BD for HPC

Research and software development for resolving technical challenges in HPC by AI/Big Data techniques

## Fundamental R&D for HPC

# AI Platform in the FugakuNEXT era and beyond: HPC/AI platform enabled by state-of-the-art system software technologies

- **AI-for-Science Platform**

- Develop AI and data processing pipelines that seamlessly integrate individual AI tools to automate a wide range of AI-for-Science research tasks

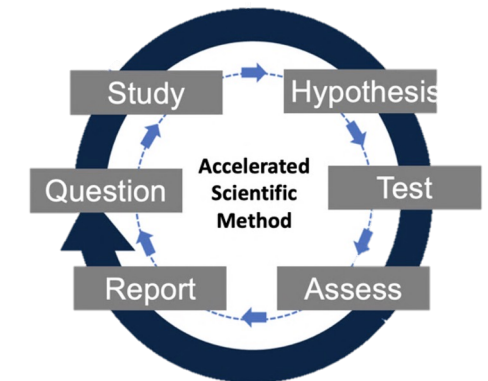
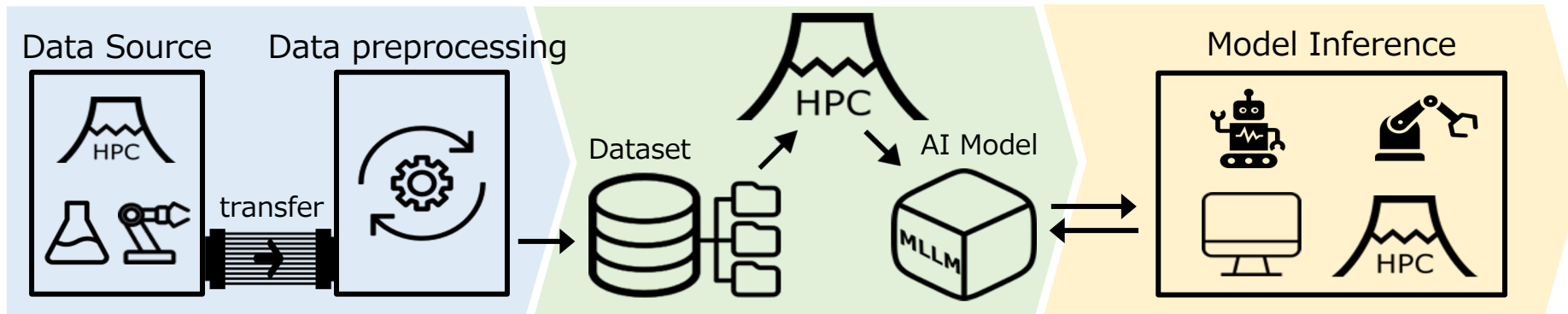
- **Acceleration of AI for Science**

- **(1) Data acquisition pipeline:** Fast generation of training data, fast transfer of various scientific data measured and produced and management of the data
- **(2) Model training pipeline:** Large-scale pre-training, post-training, and fine-tuning of AI models
- **(3) Model inference pipeline:** Automated workflow for scientific experiments by AI

## Data Acquisition Pipeline

## Model training Pipeline

## Model Inference Pipeline

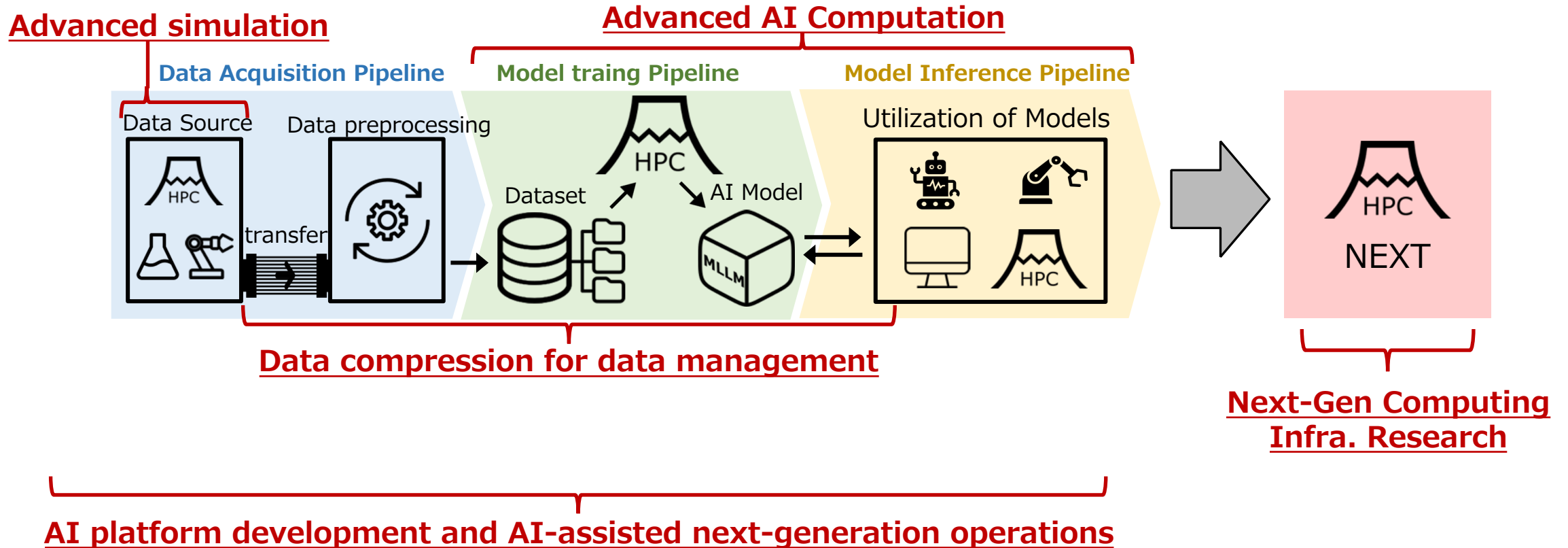


(Source: <https://doi.org/10.1038/s41524-022-00765-z>)

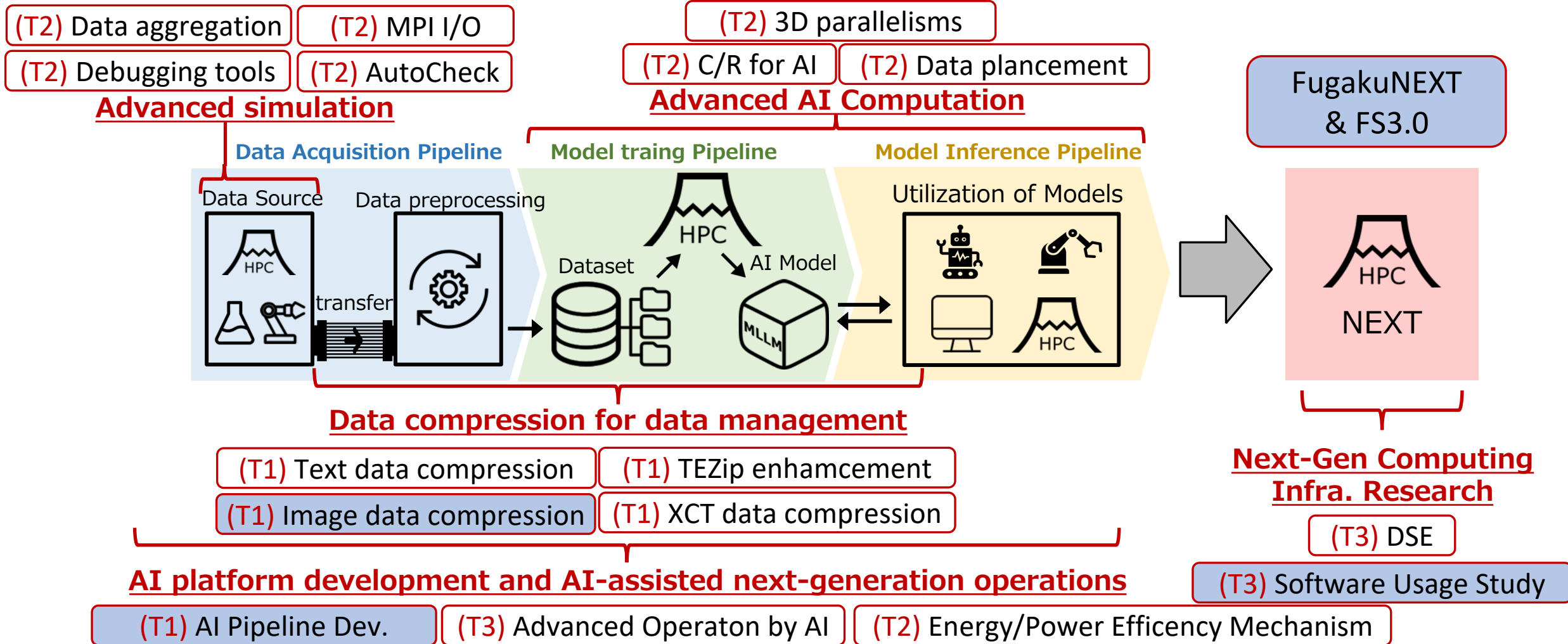
### Automate research cycles:

program code generation, execution of program, data generation, data analysis on experimental results, proposal for the next steps

# Research Plan and End-user software adoption strategy: Fundamental/core technology research for AI for Science Platform



# Research Plan and End-user software adoption strategy: Fundamental/core technology research for AI for Science Platform



We aim to develop “**default-on**” software incorporated into the platform so that end-users adopt our software without consciously

# Secured & Extendable AI Workflow Orchestration for AI4S

Masaru Nagaso<sup>†1</sup>, Jens Domke<sup>†1</sup>, Elliott Jacopin<sup>†2</sup>, Emmanuel Jeannot<sup>†1</sup>,  
Kento Sato<sup>†1</sup>, Kozo Nishida<sup>†2</sup>  
†1: RIKEN R-CCS, †2: RIKEN BDR

WIP

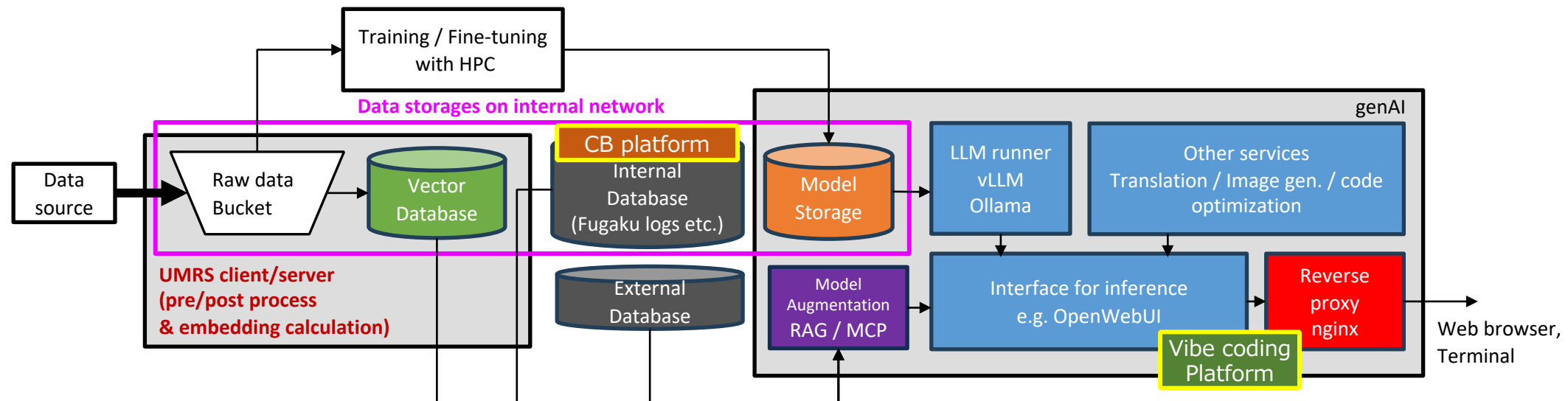


- **Background**

- Scientific AI workflows span **data transfer, data management, model training, inference, and customized data processing & analysis**, requiring tight integration.
- Manual handling slows research and increases risk of errors.
- **Security is critical** in some use cases (e.g., handling pre-published or sensitive scientific data).

- **Approach: In-house AI for Science Plat form**

- **Data pipeline:** Parallel, secure transfers with/without Globus.
- **Storage & automation:** Leverage **UMRS** (Emmanuel Jeannot) for pre/post-processing, embeddings, and data management.
- **Training & fine-tuning:** Managed execution with monitoring of progress and resources.
- **GenAI services** (Jens Domke): Network-isolation of models and data for security, supporting inference, RAG, MCP servers, etc.
- **Extendibility:** Users can deploy additional services into the workflow.
- **Centralized management layer** to coordinate data feeding, customized pre/post-processing, training status, and inference deployment.
- Built with **scalability, automation, and secure handling of sensitive datasets** as guiding principles.





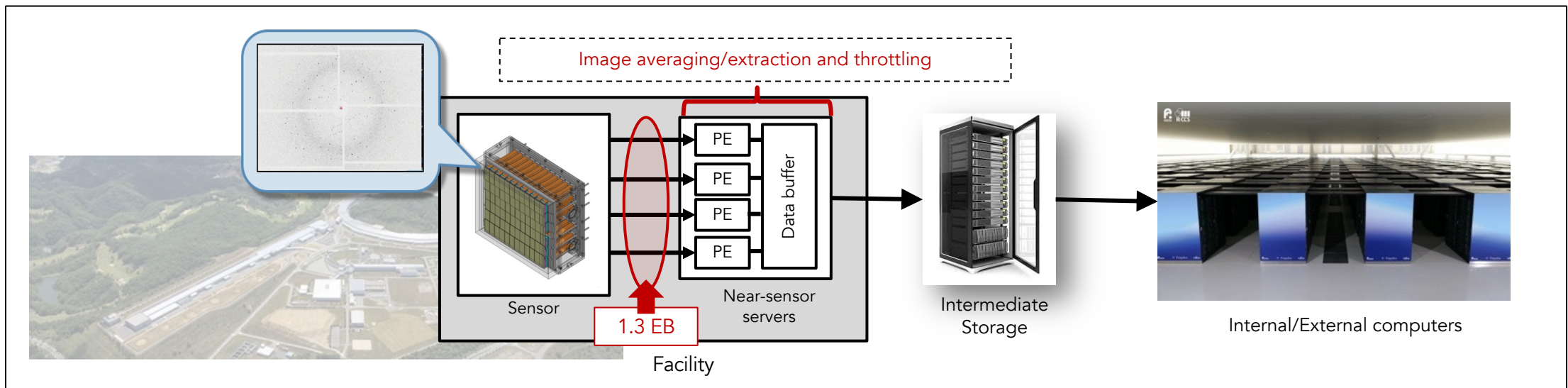
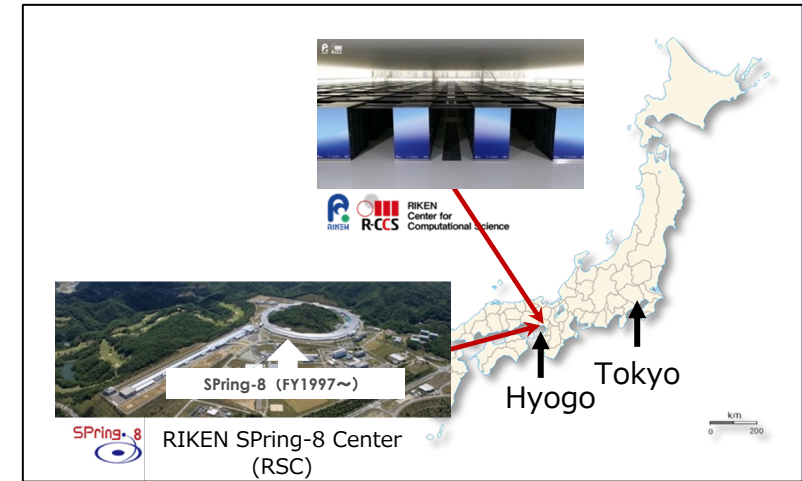
Rupak Roy<sup>†1</sup>, Kento Sato<sup>†2</sup>, Subhadeep Bhattacharya<sup>†1</sup>, Xingang Fang<sup>†1</sup>, Yasumasa Joti<sup>†3</sup>, Takaki Hatsui<sup>†4</sup>, Toshiyuki Hiraki<sup>†4</sup>, Jian Guo<sup>†2</sup> and Weikuan Yu<sup>†1</sup>

†1: Florida State University, †2: RIKEN R-CCS, †3: JASRI, †4: RIKEN RSC,



● **Background:**

- The new detector (CITIUS) in the SPring-8 Center (RSC) expected to generate about 100~400 PB per year
- To analyze and/or train an AI model with the data, data transfer from the sensors to a large-scale computer is necessary
- However, the transfer of large data becomes a performance bottleneck for AI pipeline
- X-ray CT data is one of popular data in RSC. Scientists periodically take snapshots of a target specimen while rotating it (→ Time evolutionary data)



[15] Rupak Roy, Kento Sato, Subhadeep Bhattacharya, Xingang Fang, Yasumasa Joti, Takaki Hatsui, Toshiyuki Hiraki, Jian Guo and Weikuan Yu, "Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks", In the proceedings of the 21 IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2021), May, 2021



Rupak Roy<sup>†1</sup>, Kento Sato<sup>†2</sup>, Subhadeep Bhattacharya<sup>†1</sup>, Xingang Fang<sup>†1</sup>, Yasumasa Joti<sup>†3</sup>, Takaki Hatsui<sup>†4</sup>, Toshiyuki Hiraki<sup>†4</sup>, Jian Guo<sup>†2</sup> and Weikuan Yu<sup>†1</sup>

†1: Florida State University, †2: RIKEN R-CCS, †3: JASRI, †4: RIKEN RSC,

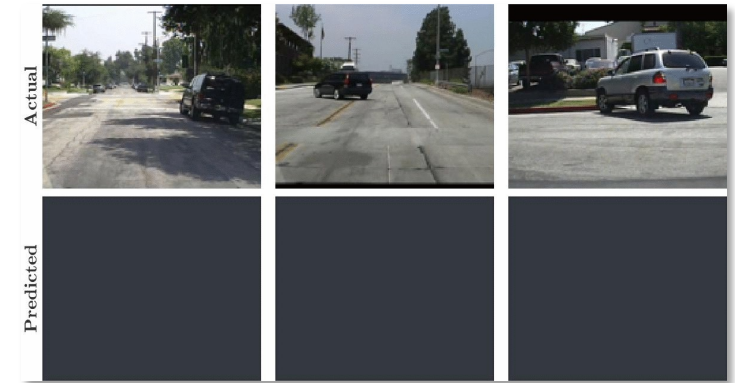


● **Key idea**

- Prediction is one of key aspects in data compression (**Fig.1**)
- With accurate prediction, target data can be converted data with sequence of zeros

● **Approach:**

- We has been developing and enhancing an AI-based data compression tool (TEZip)
- The AI model predicts or reconstruct target images and TEZip only store the delta values (**Fig.2**)
  - E.g. ) PredNet: 1<sup>st</sup> image fame –[predict]-> 2<sup>nd</sup>, 3<sup>rd</sup>,... N<sup>th</sup> image frame
  - B: Original image frames, P: Predicted image frames, D: Difference between B and P, C: Compressed image frames by a series of encoding
- Pre-processing (by AI Training): Train a NN to learn the pattern of the movement of the specimen (**Fig.2**)
- Data (De)compression (by AI Inference): Predict future images, compute delta and apply encoding/compressor (**Fig.2**)



PredNet (video from drive recorder)  
<https://coxlab.github.io/prednet/>

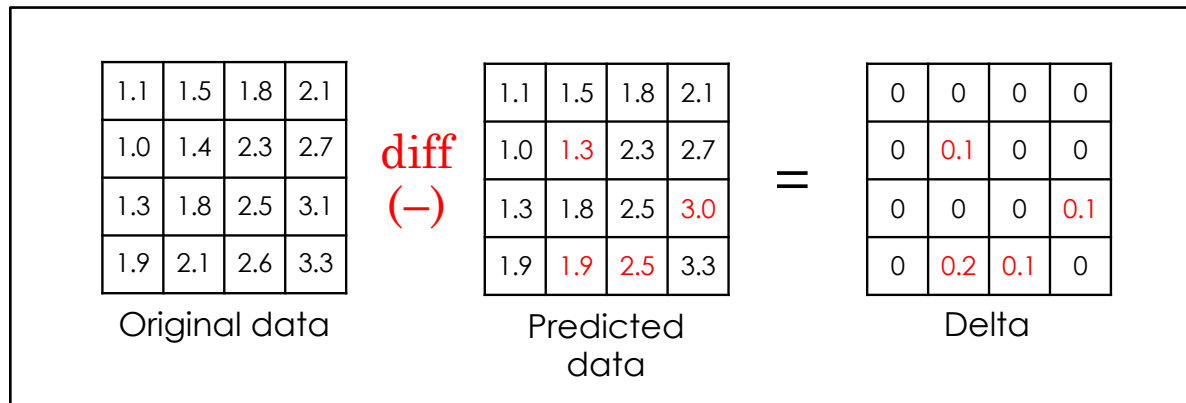


Figure 1

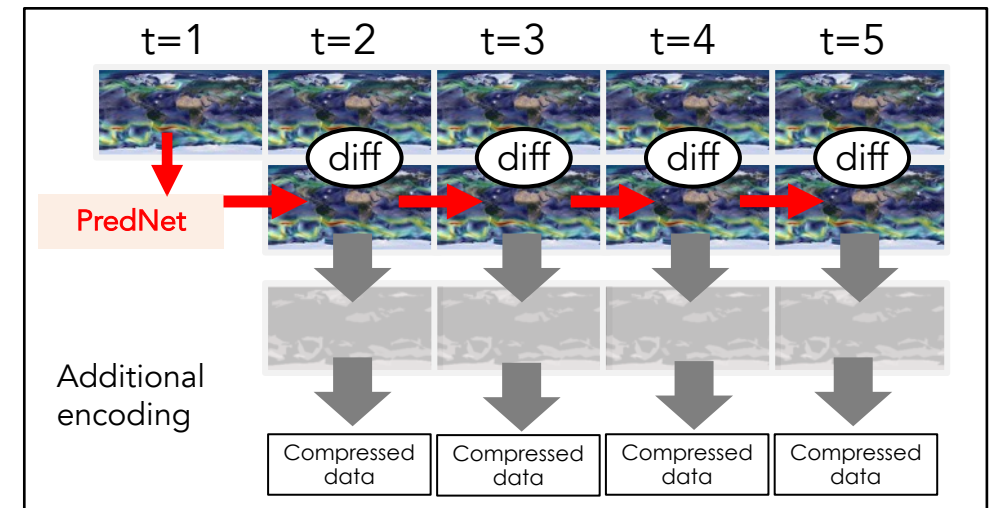


Figure 2



● **Results (Fig.8, 10):**

- TEZip gives higher compression ratio than major compression tools (e.g., X.265, SZ)
- Lossless mode: 9 to 15 / Lossy mode (w/ a few % of errors): 40 to 50

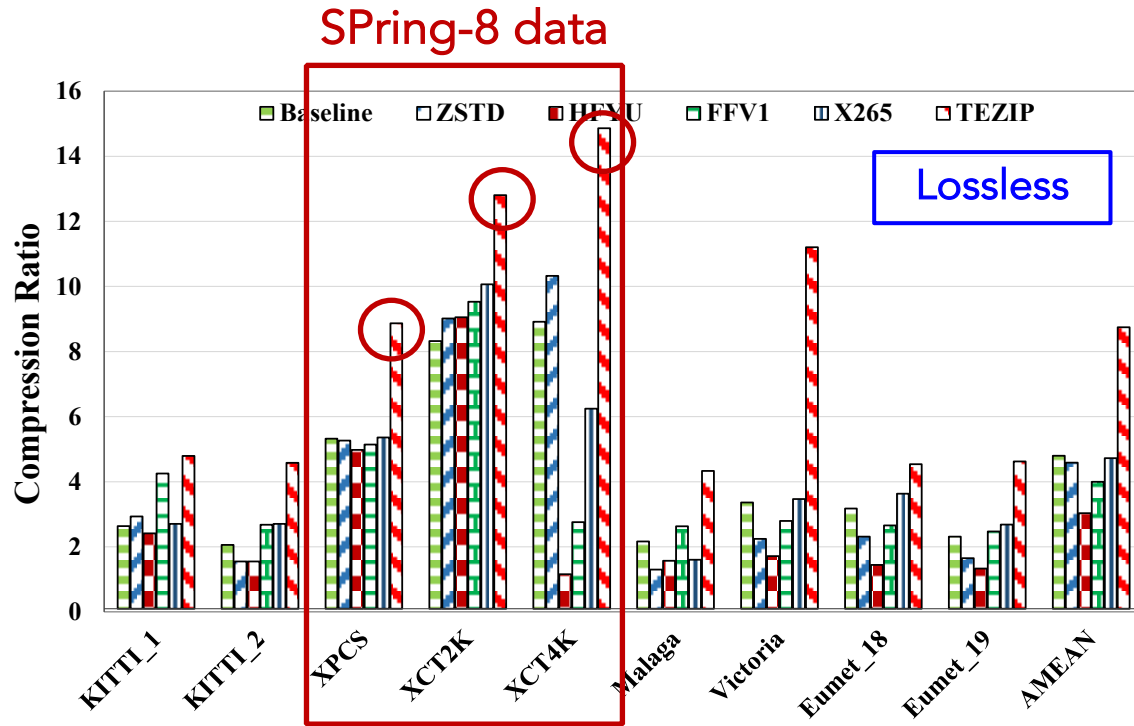


Fig. 8. Compression ratio with lossless compressors.

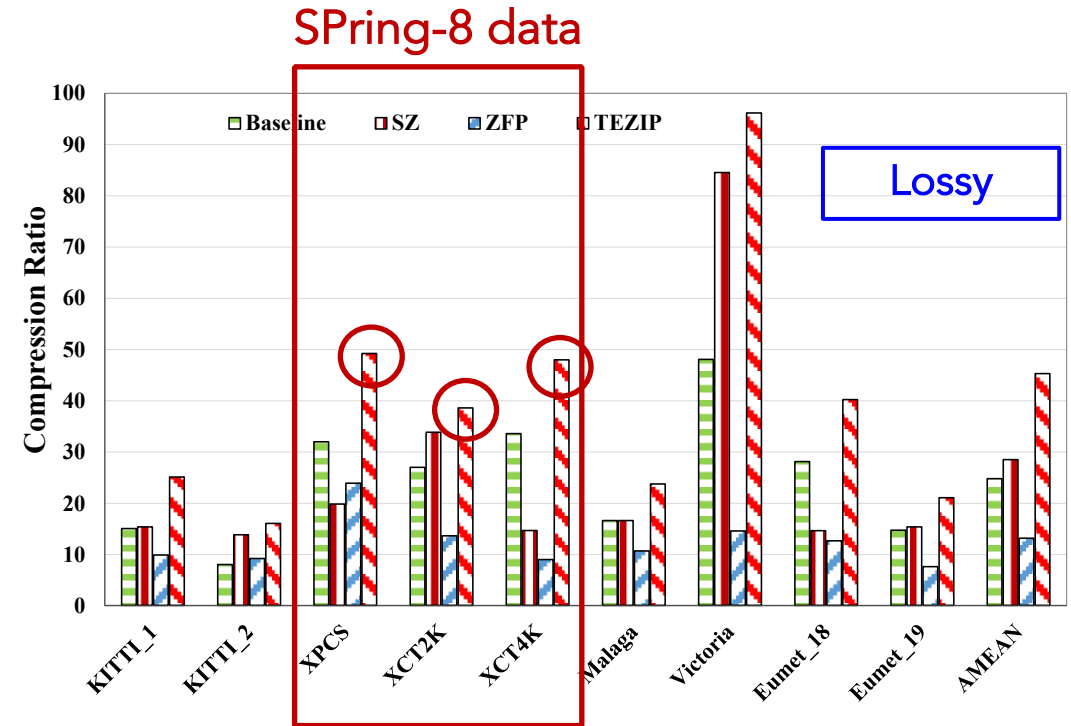


Fig. 10. Compression ratio with different lossy compressors

# System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])



Yosuke Asai<sup>†1</sup>, Kento Sato<sup>†1</sup>, Keiji Yamamoto<sup>†1</sup>, Hitoshi Murai<sup>†1</sup>, Kohei Yoshida<sup>†1,2</sup>  
†1: RIKEN R-CCS, †2: The University of Electro-Communications

## ● Background:

- RIKEN is currently developing Japan's next flagship supercomputer, the successor to Fugaku, named "FugakuNEXT".
- It is scheduled to begin operation around 2030, will continue to use Fujitsu MONAKA-X CPU and the NVIDIA's state-of-the-art GPU
- To prioritize system software development for FugakuNEXT, we need to understand usage of system software, library, compiler tool and other system software.

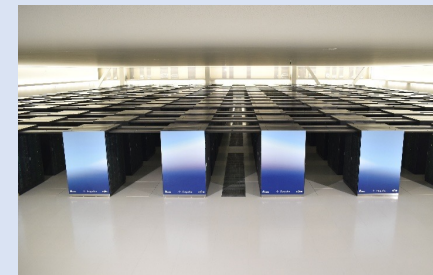
## ● Approach: Analyze usage of (i) shared libraries, (ii) Spack recipes and (iii) compilers

- (i) Shared libraries: [App log](#) produced when a user runs an MPI application
- (ii) Spack recipes: [Spack log](#) produced when a user loads Spack package. (Spack: A software package management tool developed for HPC environments)
- (iii) Compilers: [License server log](#) produced when a user invokes a Fujitsu compiler (C, C++, Fortran)

## Period

Log Types	Start date	End date	Number of logs
<a href="#">App log</a>	May 21, 2021	March 31, 2025	11,359,316 Jobs
<a href="#">Spack log</a>	July 19, 2022	March 31, 2025	5,807,006 Jobs
<a href="#">Compiler log</a>	Feb. 19, 2021	March 31, 2025	121,478 Calls

## System software development: Fugaku to FugakuNEXT



## Metrics and types of aggregated data

Log Types	Metrics	Aggregated
App log	Shared library, path of exec. app.	Usage count, node hour, num of user, num of user's group (For the entire period and for each month)
Spack log	Spack package	Usage count, node hour, num of user, num of user's group (For the entire period and for each month)
Compiler log	Language type, compilation options	Invocation count, invocation count as a linker (For the entire period and for each month)

# System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])



Yosuke Asai<sup>†1</sup>, Kento Sato<sup>†1</sup>, Keiji Yamamoto<sup>†1</sup>, Hitoshi Murai<sup>†1</sup>, Kohei Yoshida<sup>†1,2</sup>  
<sup>†1</sup>: RIKEN R-CCS, <sup>†2</sup>: The University of Electro-Communications

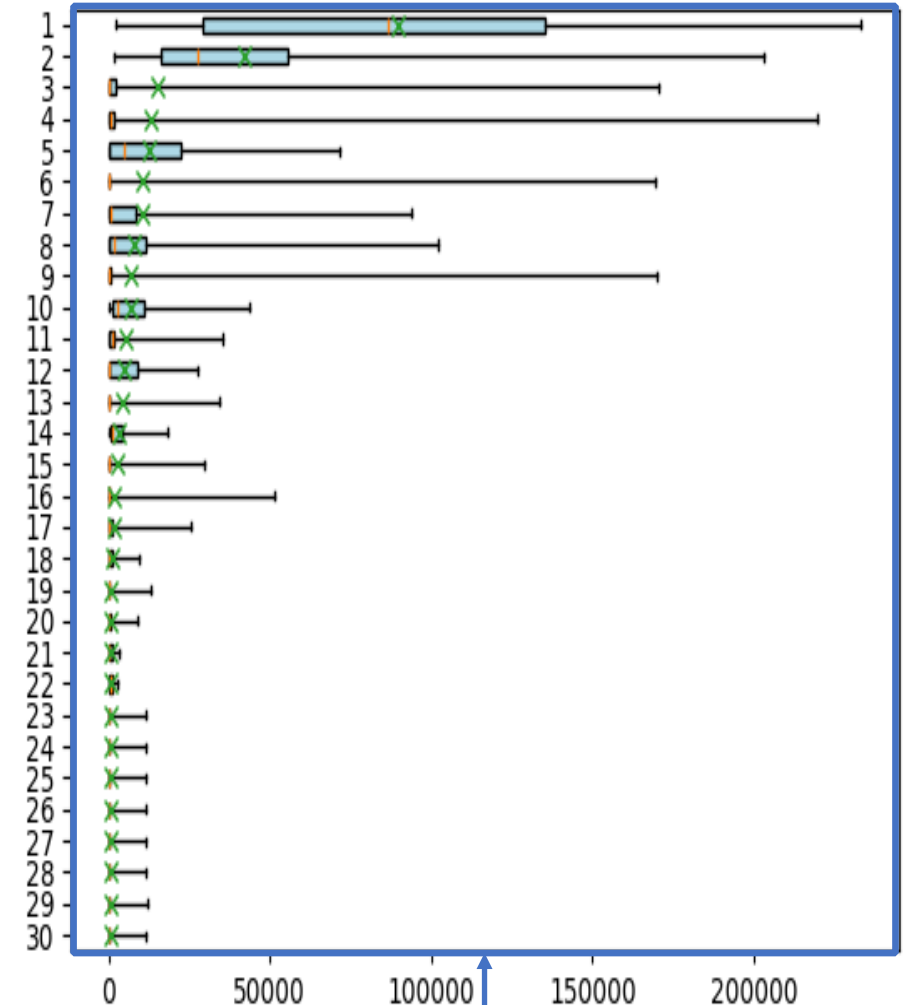
## Spack log – package usage count

- **netCDF (for Fortran) is most used via Spack**
  - netcdf-fortran (ranked 1st) is used more than twice as frequently as the next-ranked (2nd) package
- **Many of MD-related application/libraries are used via Spack**
  - GROMACS, ABINIT-MP, Quantum ESPRESSO, LAMMPS, pymatgen etc.
- **Spack users use GROMACS and netCDF**
  - The rest of packages ranked below third place are relatively low usage

Usage counts for the entire period

Rank	Spack package	Count
1	netcdf-fortran	2,966,582
2	gromacs	1,397,102
3	py-pandas	496,051
4	py-pip	436,878
5	abinitmp	420,233
6	glew	356,763
7	fujitsu-mpi	350,796
8	quantum-espresso	266,086
9	lammps	229,913
10	py-numpy	226,151
11	fujitsu-fftw	175,070
12	py-pymatgen	164,821
13	libxc	142,440
14	hdf5	102,145
15	akaikkr	94,919
16	netlib-lapack	61,421
17	py-scipy	55,259
18	py-mpi4py	33,748
19	fftw	30,672
20	python	28,233
21	gcc	24,883
22	netcdf-c	24,856
23	py-pillow	24,540
24	py-python-dateutil	22,878
25	py-pytz	22,573
26	py-pyparsing	22,550
27	py-cycler	22,545
28	py-kiwisolver	22,542
29	openblas	22,428
30	paraview	20,049

Monthly usage counts



# System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])

Yosuke Asai<sup>†1</sup>, Kento Sato<sup>†1</sup>, Keiji Yamamoto<sup>†1</sup>, Hitoshi Murai<sup>†1</sup>, Kohei Yoshida<sup>†1,2</sup>

<sup>†1</sup>: RIKEN R-CCS, <sup>†2</sup>: The University of Electro-Communications

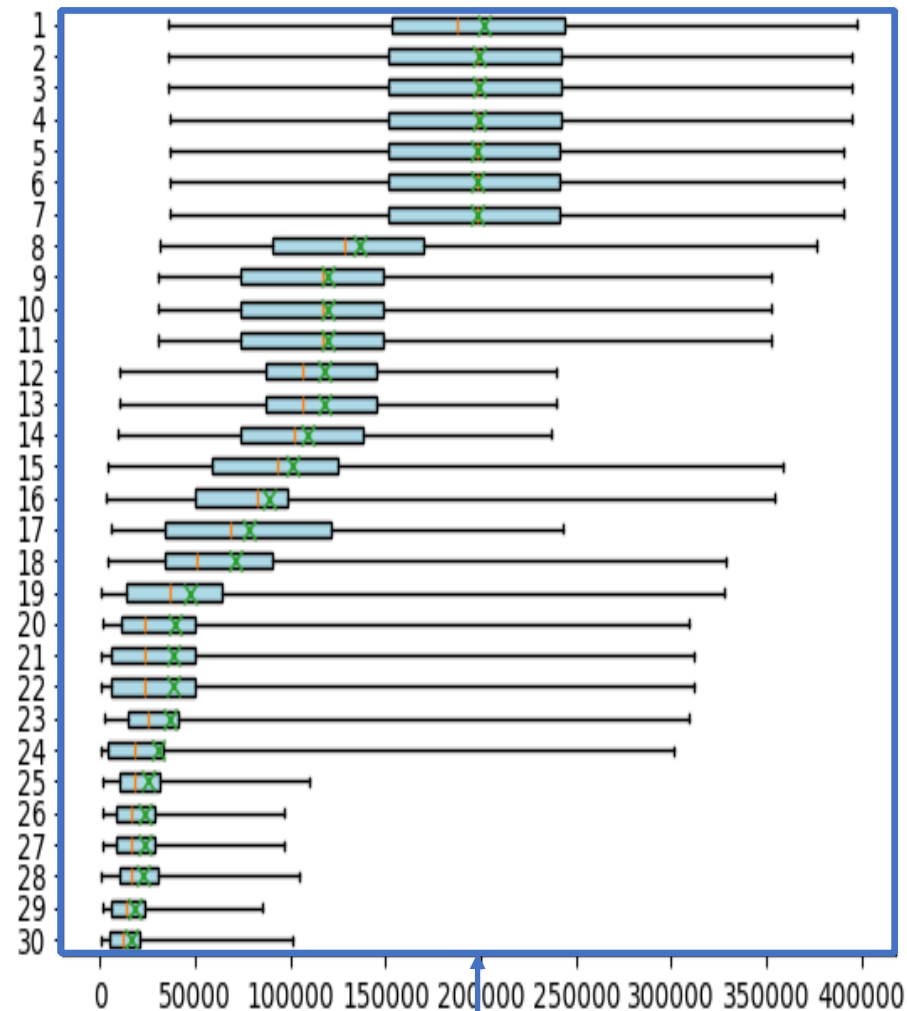
## App log – Shared library usage count

- **Basic shared libraries are majority**
  - Libraries related to MPI, OpenMP, Fortran, C++, profiler
  - Note:
    - libfj\*.so: Fujitsu libraries.
    - libmpg.so: handles large shared-memory pages.
    - libelf.so: provides access to files in the Executable and Linkable Format (ELF).
    - libmpi\*.so/lib\*mpi.so: MPI-related libs
- **Well-known libraries used many times**
  - LAPACK
  - GROMACS
  - FFTW
  - ...
- **Usage of those libraries are relatively consistent across different months**

## Usage counts for the entire period

Rank	Shared library	Count
1	libmpg.so	9,493,441
2	libfjprofcore.so	9,342,392
3	libfjprofomp.so	9,342,392
4	libelf.so	9,339,212
5	libfj90i.so	9,321,471
6	libfj90f.so	9,321,471
7	libfjsrcinfo.so	9,321,150
8	libatomic.so	6,421,803
9	libfjomp.so	5,633,687
10	libfjompshk.so	5,633,197
11	libfjompprt.so	5,633,107
12	libmpi_usempi_ignore_tkr.so	5,541,646
13	libmpi_usempif08.so	5,541,378
14	libfjprofmpif.so	5,142,711
15	libmpi_cxx.so	4,757,229
16	libfjprofmpi.so	4,154,245
17	libstdc++.so	3,679,499
18	libfjrt.so	3,331,647
19	libmpi_mt.so	2,238,543
20	<b>libfjlapacksve.so</b>	<b>1,856,063</b>
21	libc++.so	1,799,068
22	libc++abi.so	1,799,068
23	libfj90rt.so	1,713,729
24	<b>libgromacs_mpi.so</b>	<b>1,429,380</b>
25	libmpi.so.0	1,185,445
26	libfjc++.so	1,103,723
27	libfjc++abi.so	1,103,676
28	<b>libfftw3.so</b>	<b>1,051,486</b>
29	libfidemgl.so	836,751
30	<b>libfjlapackexsve.so</b>	<b>763,892</b>

## Monthly usage count



# System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])



Yosuke Asai<sup>†1</sup>, Kento Sato<sup>†1</sup>, Keiji Yamamoto<sup>†1</sup>, Hitoshi Murai<sup>†1</sup>, Kohei Yoshida<sup>†1,2</sup>  
†1: RIKEN R-CCS, †2: The University of Electro-Communications

- **C/C++ vs. Fortran compiler calls (Figure 1)**
  - Fortran compilers are used more than C/C++
  - Many of applications/programs in Fugaku are written in Fortran  
→ Continuous support for Fortran language/compier in HPC is necessary for FugakuNEXT and beyond
- **C vs. C++ compiler calls (Figure 2)**
  - C++ compilers are used more than C across different months  
→ Reasonable to consider deployment of C++-based frameworks such as Kokkos, StdPar and SYCL
- **Future work:** See other metrics, e.g., counting # of unique users (in total and in each research area)

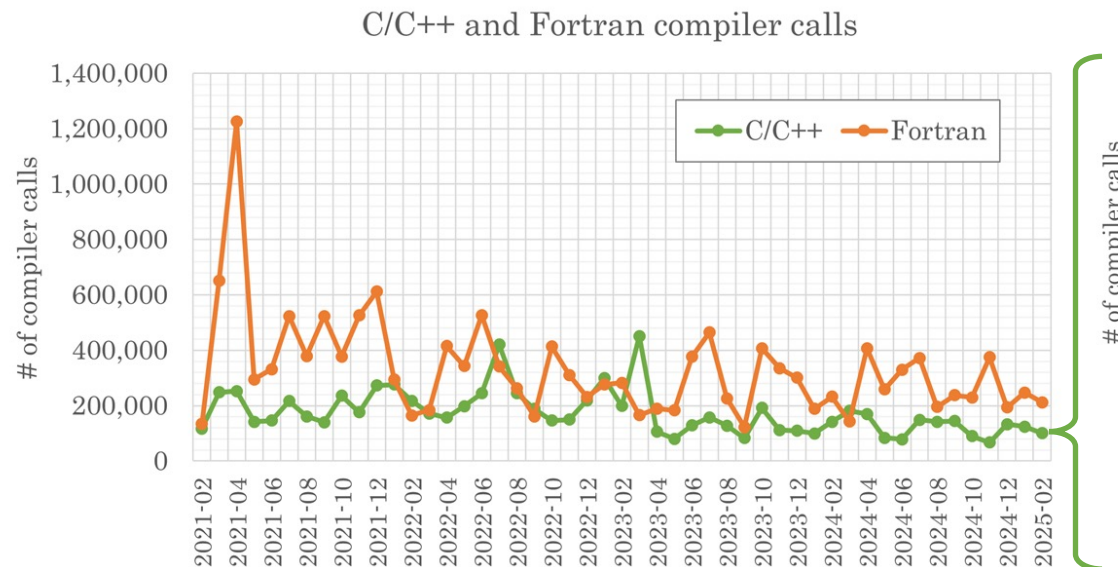


Figure 1

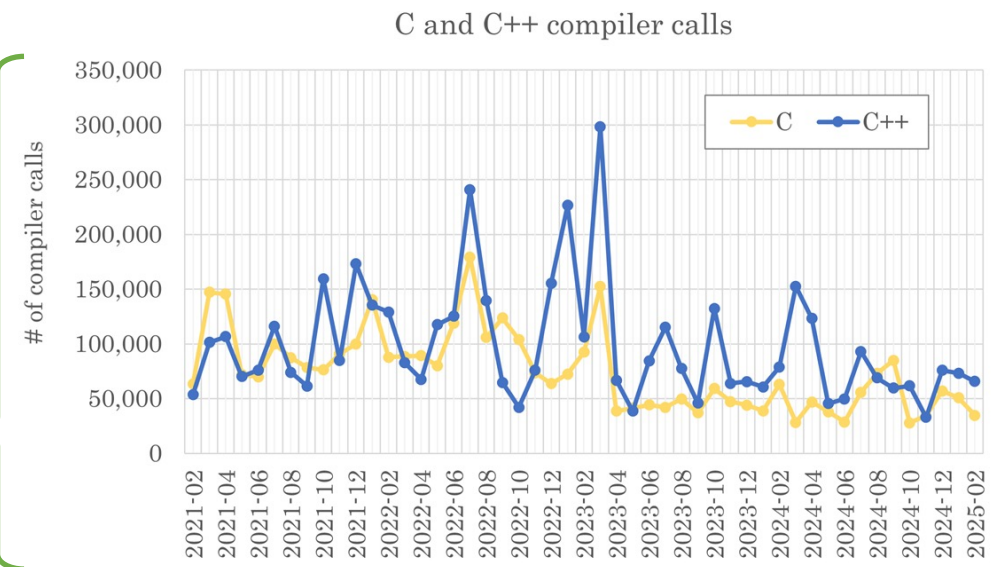


Figure 2

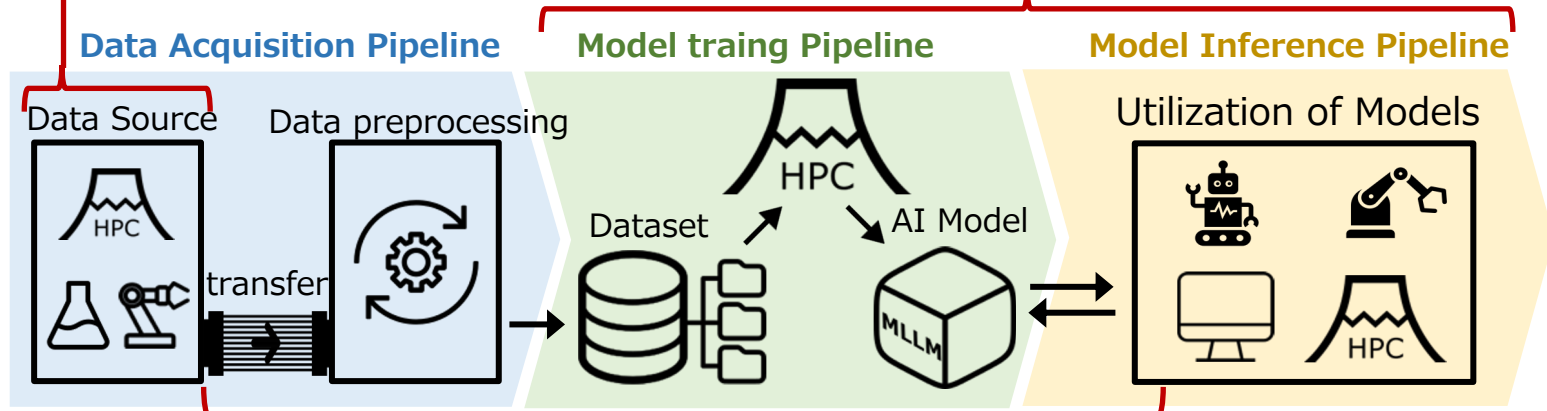
# Summary: AI for Science Platform towards FugakuNEXT

- (T2) Data aggregation
- (T2) MPI I/O
- (T2) Debugging tools
- (T2) AutoCheck

## Advanced simulation

- (T2) 3D parallelisms
- (T2) C/R for AI
- (T2) Data placement

## Advanced AI Computation



FugakuNEXT & FS3.0

HPC NEXT

## Data compression for data management

- (T1) Text data compression
- (T1) TEZip enhancement
- (T1) Image data compression
- (T1) XCT data compression

## Next-Gen Computing Infra. Research

(T3) DSE

## AI platform development and AI-assisted next-generation operations

(T3) Software Usage Study

- (T1) AI Pipeline Dev.
- (T3) Advanced Operaton by AI
- (T2) Energy/Power Efficiency Mechanism