Microservice Communication Resource Scheduling for Distributed AI Model

<draft-yang-dmsc-distributed-model-04>

Hui Yang, **Tiankuo Yu,** Qiuyan Yao, Zepeng Zhang

Sate Key Laboratory of Information Photonics and Optical Communications,

Beijing University of Posts and Telecommunications, China

E - mail: yutiankuo@bupt.edu.cn

IETF 123, Madrid, July 2025

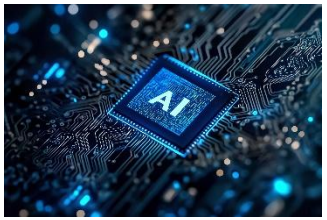
Outline

Outline

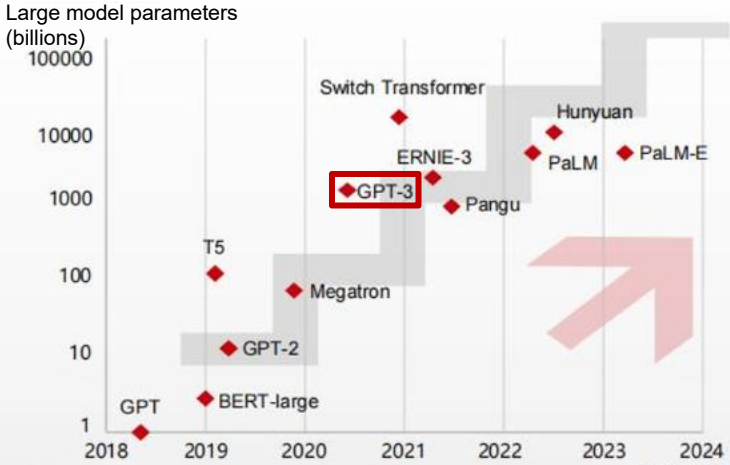
Growing Demand for Computing Power and Communication in Large Model Training

- Urgent requirements :
  - Operation of large models like GPT requires stronger computational support
  - The growing demand for real-time model training in fields like AI and IoT creates new challenges for distributed model training

Artificial intelligence



Digital economy



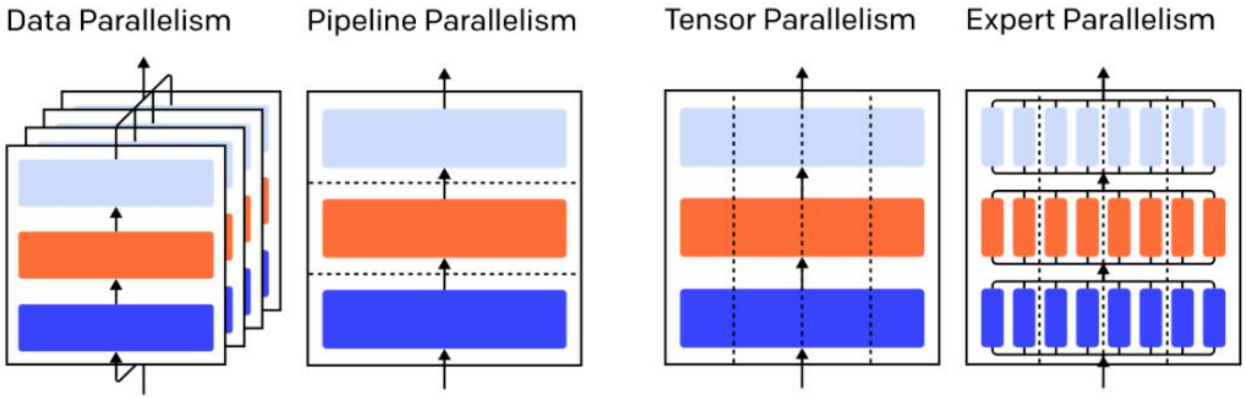
The large model parameters vary with release time

Industrial Internet

Health care

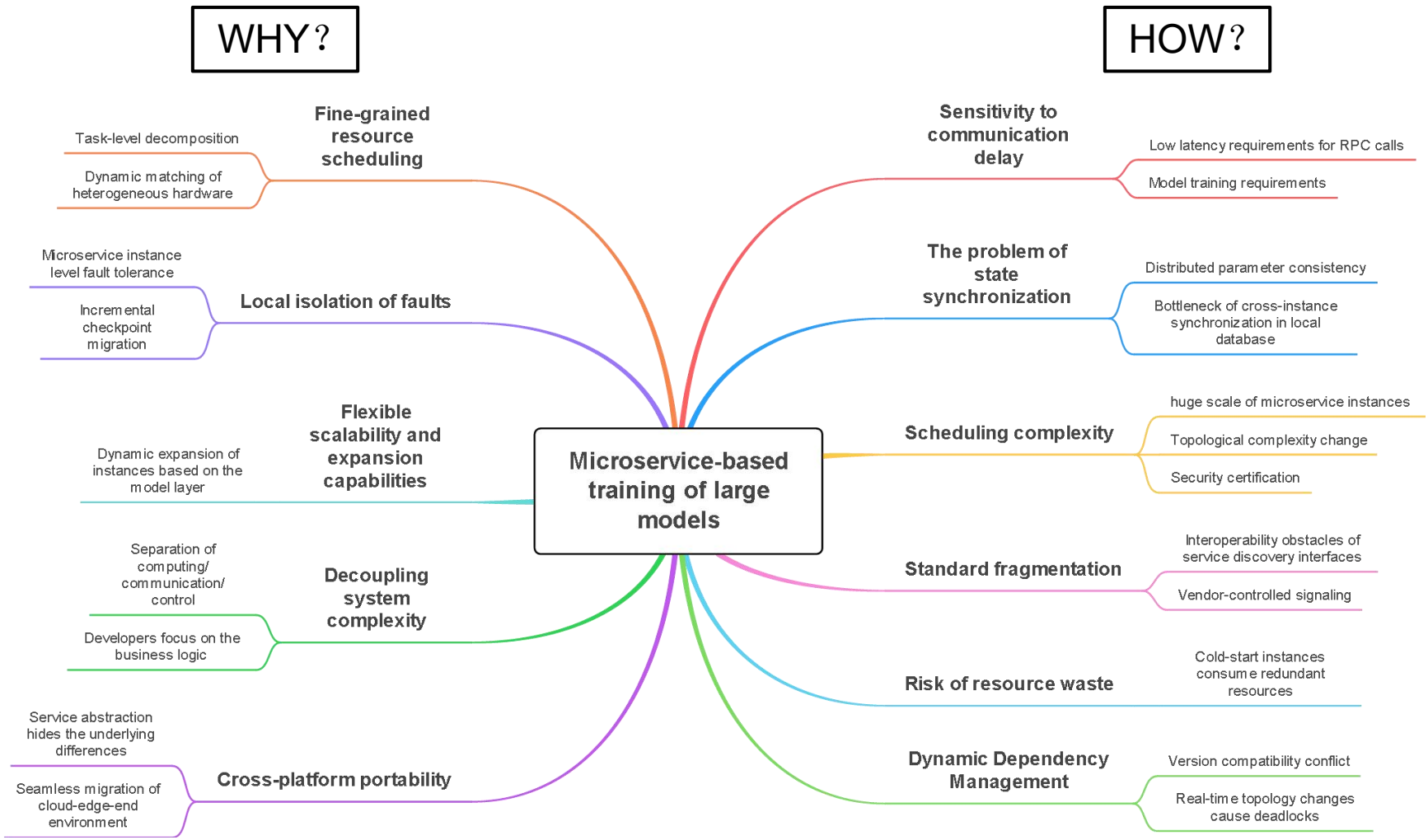
Distributed training mode analysis of large models

ChatGPT deepseek Grok



Parallel training method	Communication requirements	Delay requirements
Tensor Parallelism	Inside the server	Ultra-low latency
Pipeline Parallelism	Across servers	Delay-sensitive, capable of partial concealment
Data Parallelism	Across servers	delay can largely masked
Expert Parallelism	Across servers	Delay-sensitive

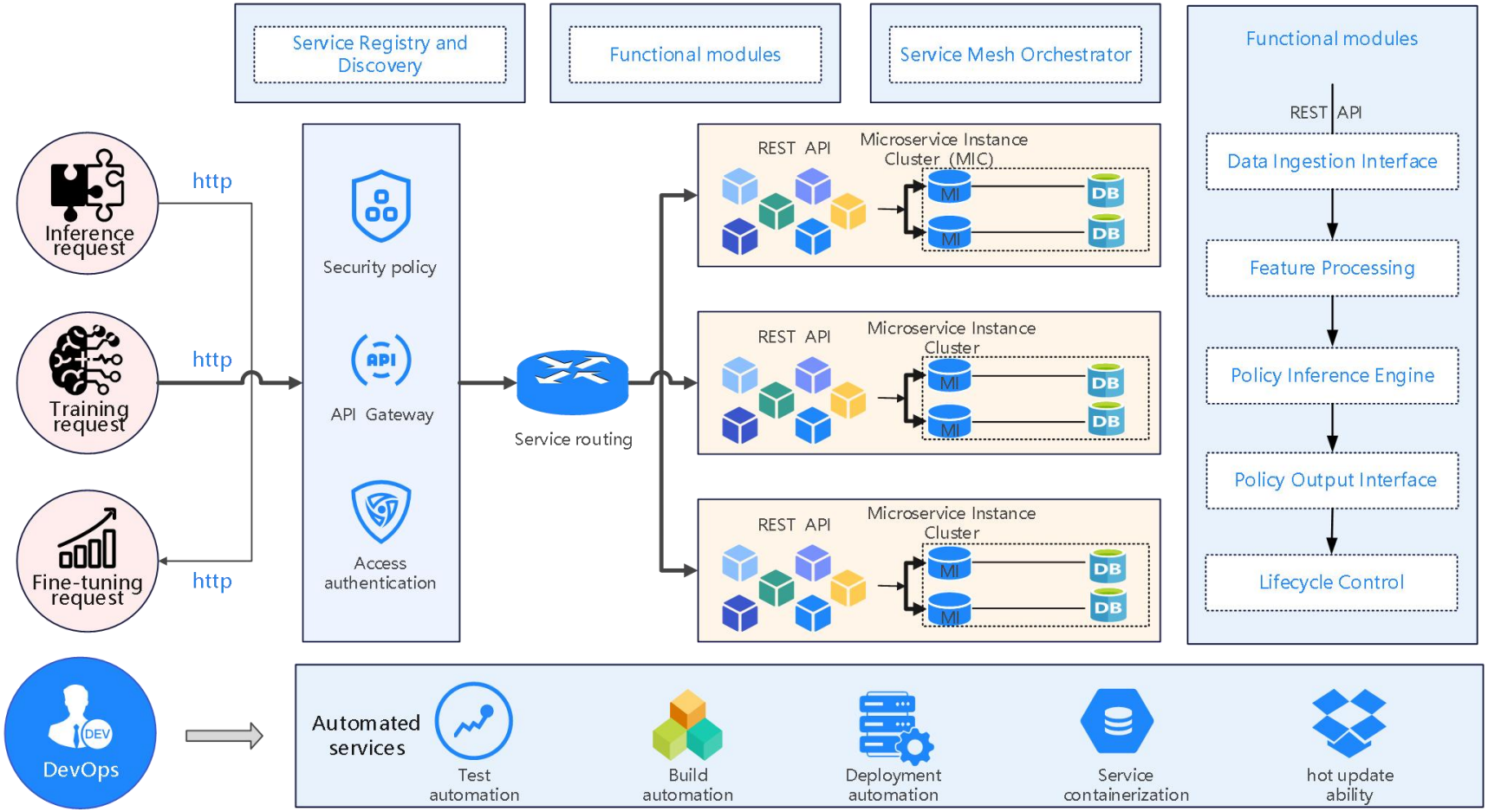
The Microservice Dilemma in Large Model Training: Advantages vs Challenges



Outline

Overview of the DMSC-LMT Architecture

- Distributed Microservice Communication Architecture for Large Model Training (DMSC-LMT)

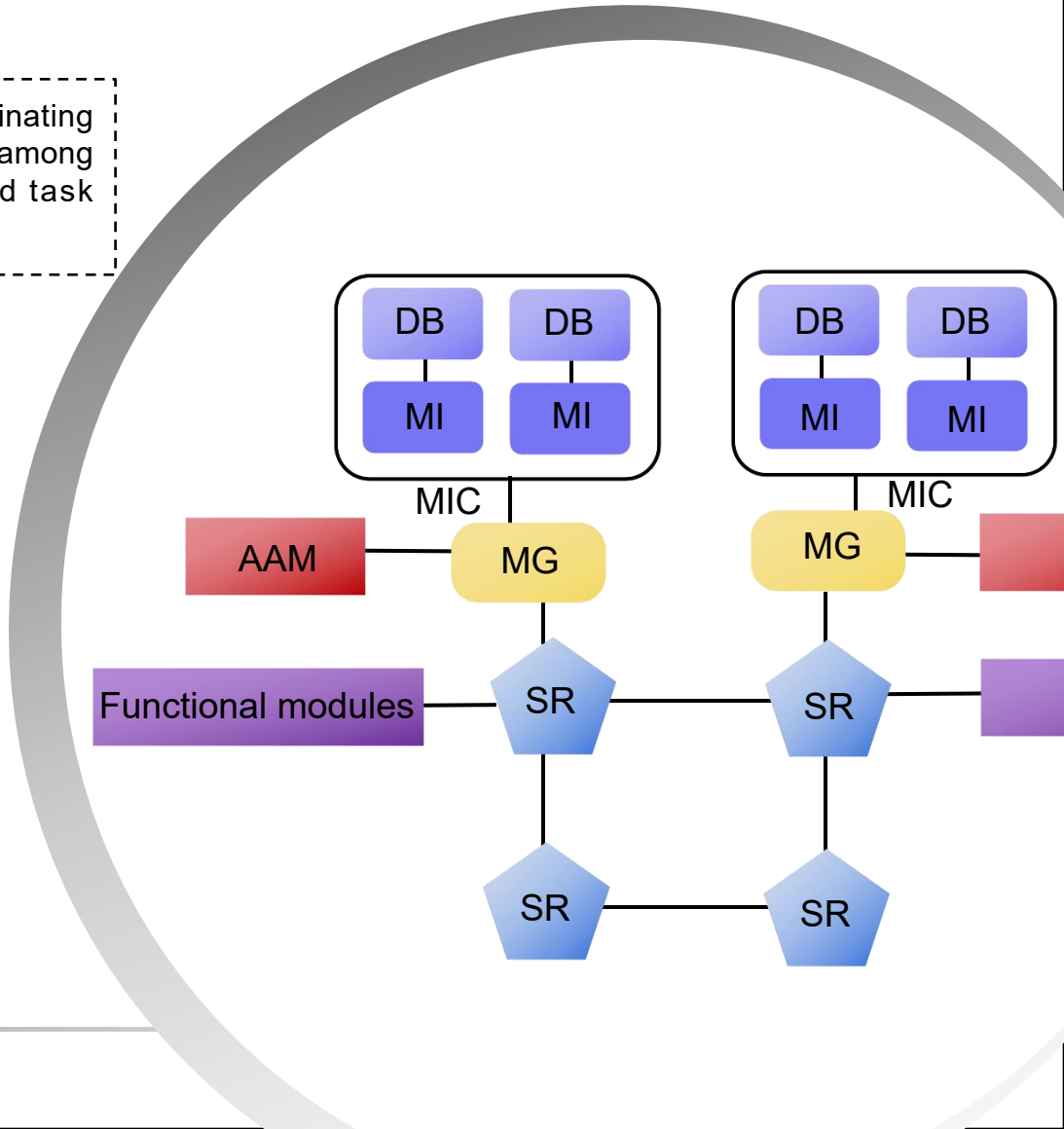


Overview of the DMSC-LMT Architecture

- Distributed Microservice Communication Architecture for Large Model Training (DMSC-LMT)

Service Mesh Orchestrator：Responsible for coordinating and optimizing communication and task scheduling among microservices, dynamically adjusting traffic and task allocation to ensure efficient collaboration.

Service Registry and Discovery: Manages microservice registration and discovery, providing instance information to SR and MG for dynamic service location and traffic routing.



Component Overview of the DMSC-LMT Architecture

- Distributed Microservice Communication Architecture for Large Model Training (DMSC-LMT)

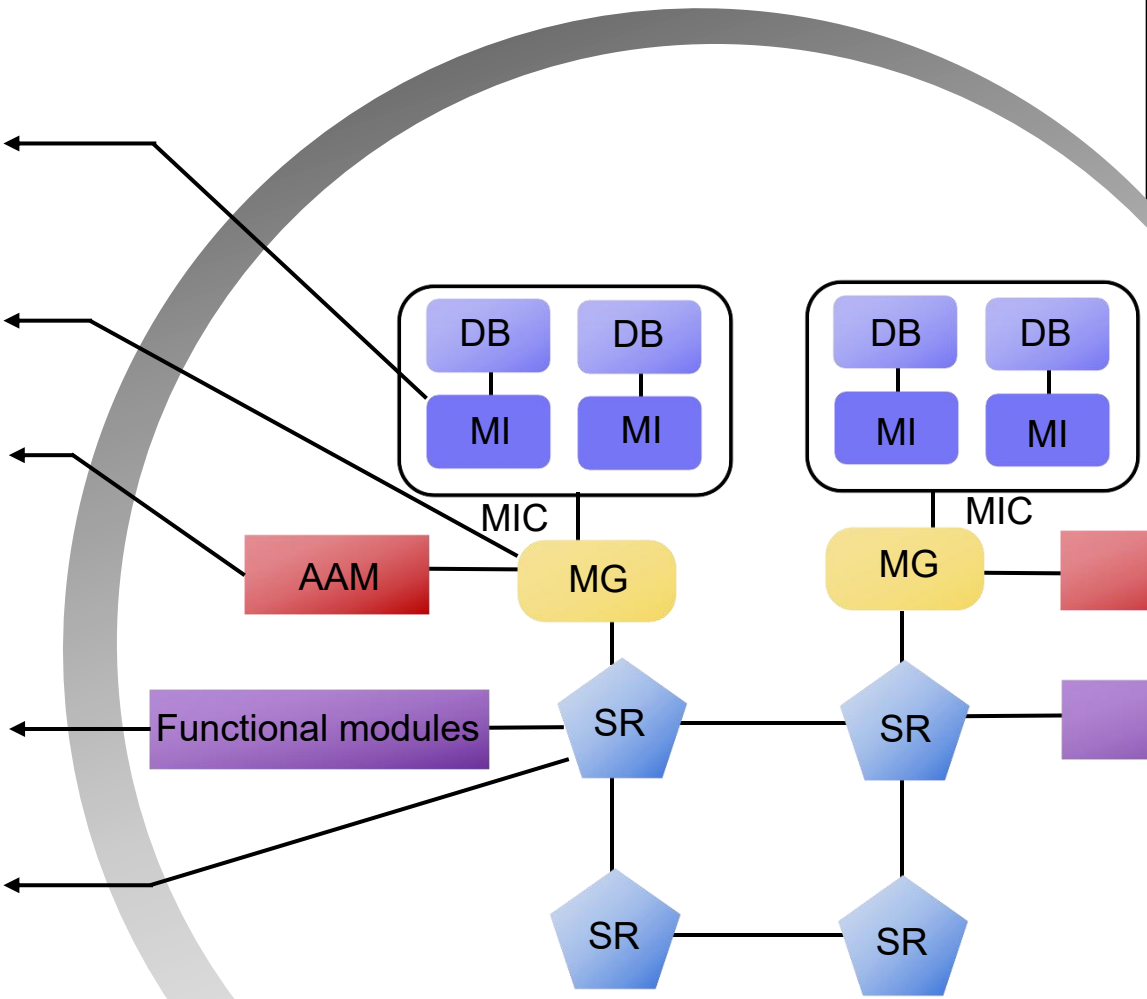
Microservice Instance: A runtime entity identified by a unique Service ID (SID), deployed in a Microservice Instance Cluster (MIC).

Microservice gateway: A core component that handles external requests, including routing, load balancing, authentication, and API aggregation for efficient and stable service dispatch.

Authentication and Authorization Module: A core component in MG that handles identity authentication and authorization, ensuring incoming requests are authenticated and authorized to prevent unauthorized access.

loosely coupled components used to perform intelligent policy analysis, path optimization, and predictive reasoning.

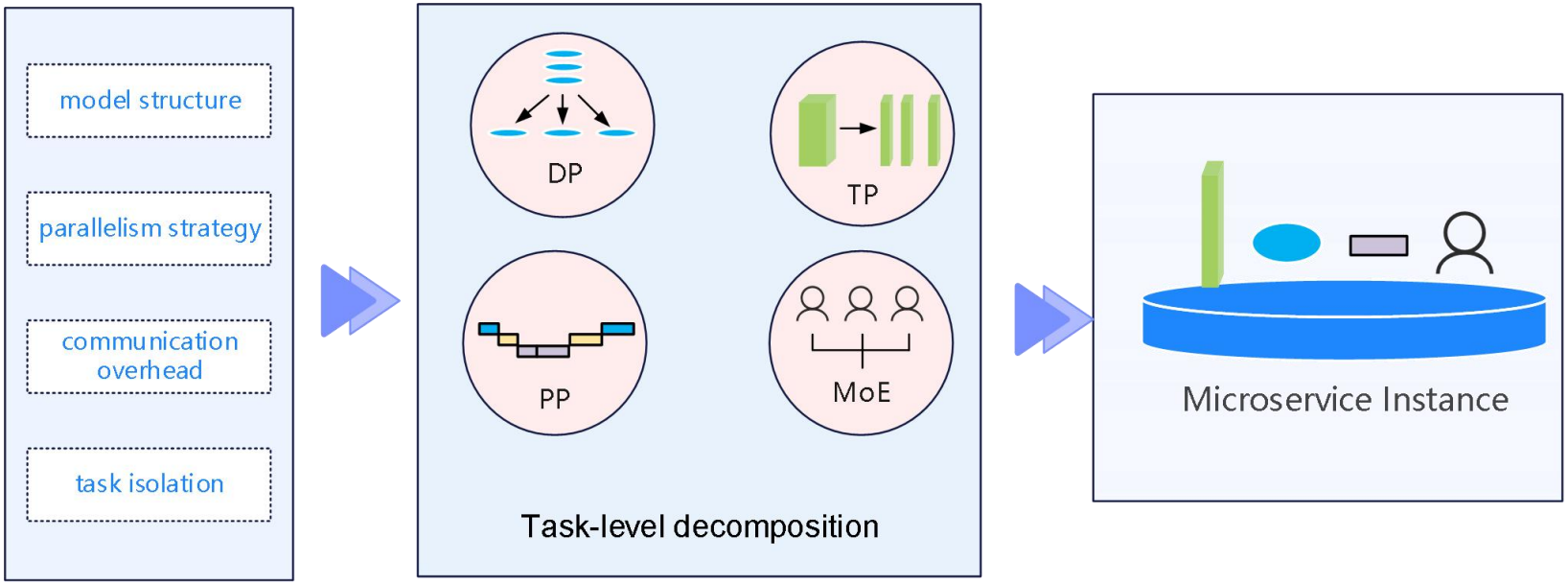
Service Router: Manages internal communication, dynamically selects service instances, and routes traffic with load distribution to ensure efficient service communication.



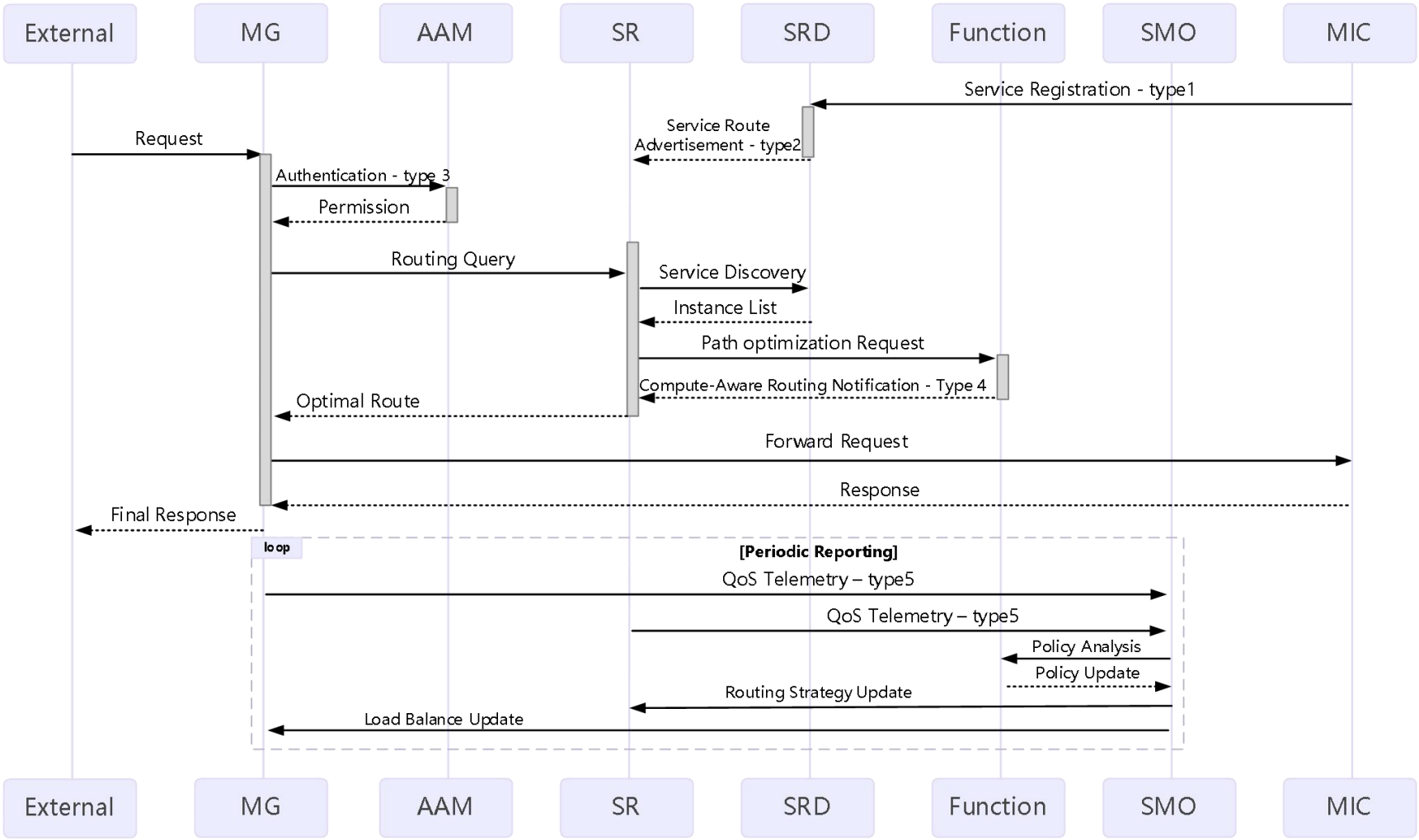
Outline

Task-level decomposition

- **Core Considerations:**
  - Decouple the large-scale training workflow and split it into computing sub-tasks
  - The determination of the granularity level is based on the model structure, parallel strategies, as well as the trade-off between communication overhead and task isolation.
  - Achieve an effective balance between atomic execution units and manageable coordination complexity



DMSC-LMT: Dynamic Signaling Flow for Intelligent Orchestration



Communication Path Configuration and Task Execution Process

Service Registration and Communication Path Configuration

Data Exchange During Model Training

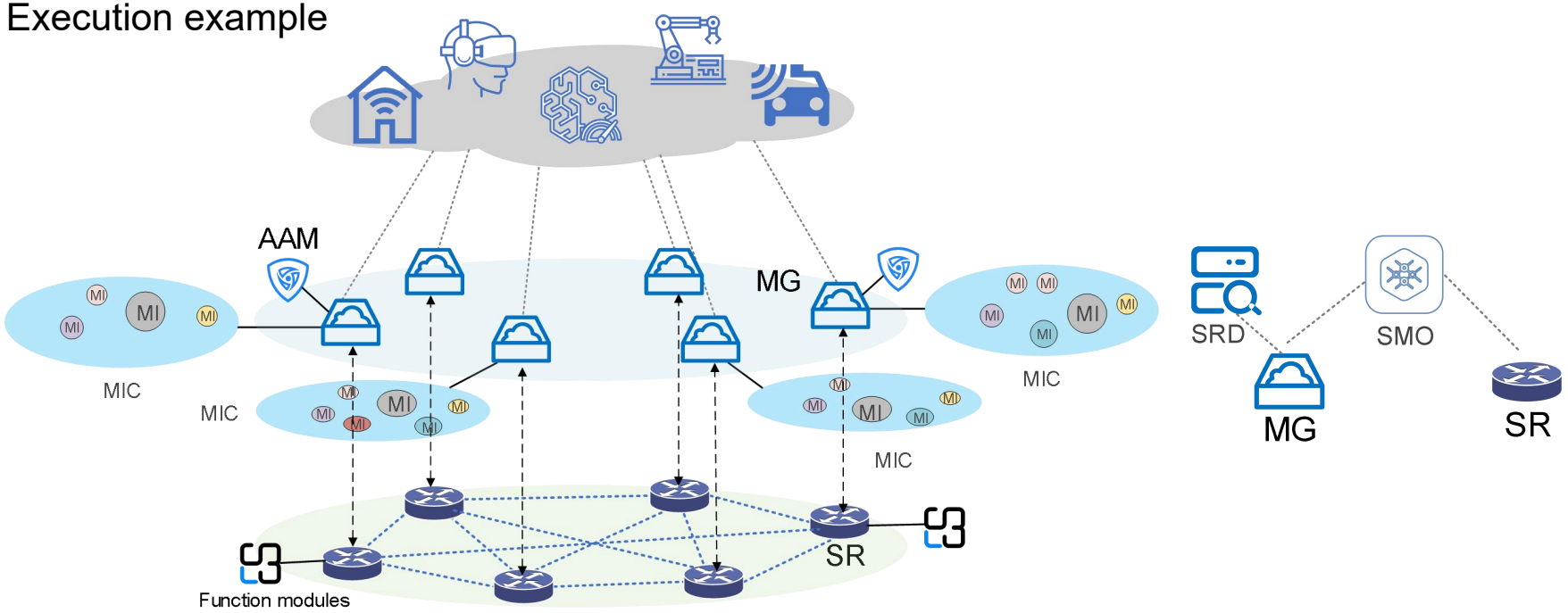
Telemetry of Communication Quality and Policy Feedback

QoS Policy Update and Traffic Adjustment

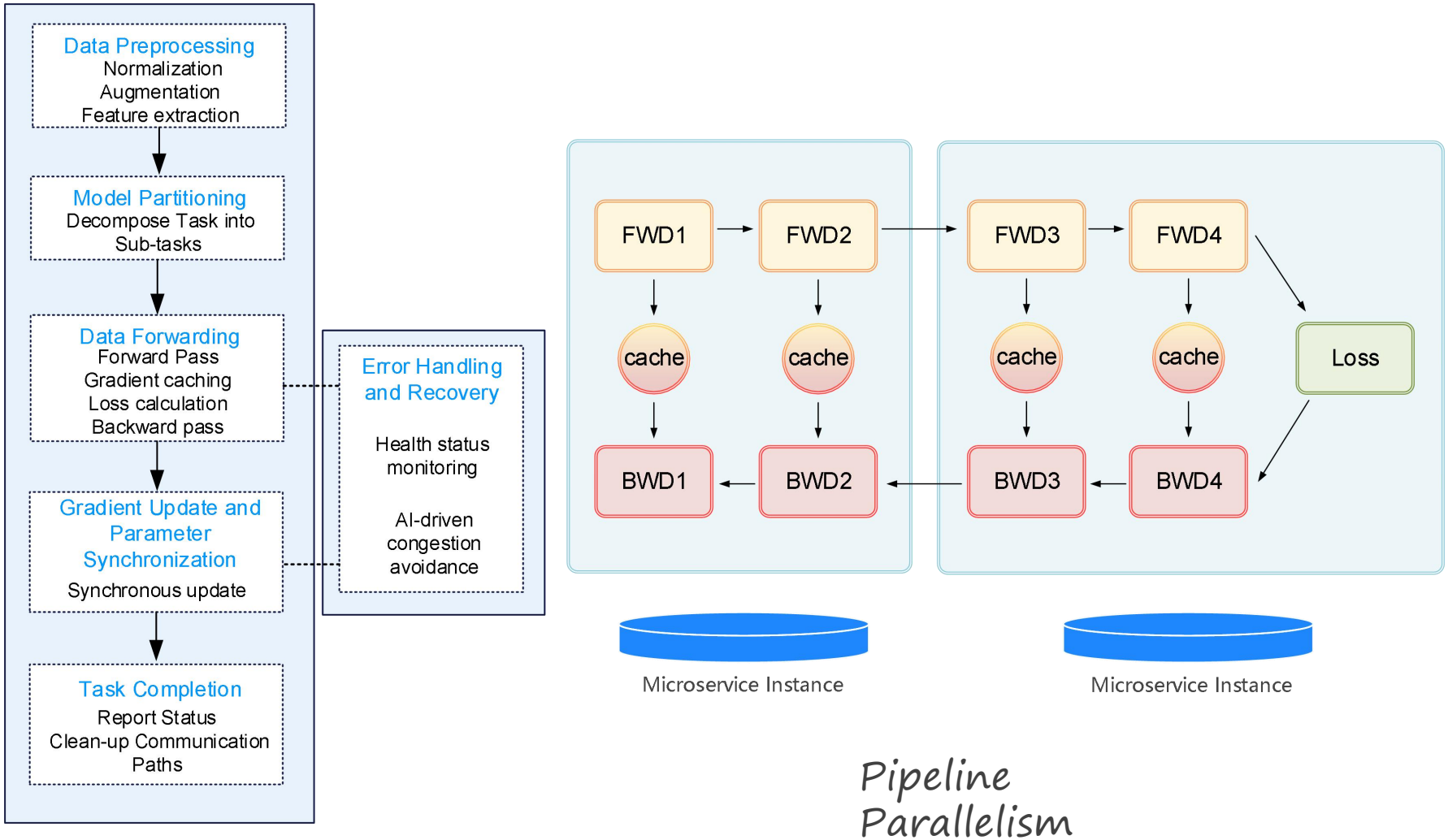
Exception Handling and Fault Recovery

Communication Termination and Resource Cleanup

- Execution example



Task-Level Decomposition and Communication Path Flow



Outline

Conclusion

- Advantages of the DMSC-LMT Architecture:
  - **Efficient task scheduling:** Decompose tasks into independent sub-tasks for optimized computing and resource use.
  - **Flexible communication path configuration:** Use SMO and SRD for dynamic service management and optimized data transmission.
  - **Intelligent Decision Support:** Leverage functional modules for traffic prediction and real-time optimization based on system load and network conditions.

- Future Outlook and Challenges
  - **Intelligent enhancement:** Further introduction of more self-learning and self-adaptive algorithms enables the system to become more intelligent and automated during the process of task decomposition and scheduling.
  - **Cross-disciplinary application:** The DMSC-LMT architecture is not only applicable to large-scale AI model training, but can also be extended and applied to other fields, such as data science and industrial internet.

Thank You !

Comments are always appreciated.

IETF：Microservice Communication Resource Scheduling for Distributed AI Model

http://www.ietf.org/archive/id/draft-yang-dmsc-distributed-model-04.txt

Hui Yang, **Tiankuo Yu**, Qiuyan Yao, Zepeng Zhang

Sate Key Laboratory of Information Photonics and Optical Communications,
Beijing University of Posts and Telecommunications, China

E - mail: yutiankuo@bupt.edu.cn