UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

Distributed Agents

SummerSOC 2026, June 29, 2026

Prof. Dr. Stefan Fischer – Institut für Telematik

Agenda

- Motivation and Example
- Architectures for Distributed Agent Systems
- Technical Foundations: MCP and A2A
- Broader Application Context in Edge-Cloud Architectures
  - physMCP
  - SDC-MCP Gateway

Motivation

What is Agentic AI?

- autonomous, goal-driven entities (“agents”)
- Agents can plan, decide, act, and learn
- Often built on top of Large Language Models (LLMs)
- Designed for complex, dynamic environments

Agentic AI = LLM-driven agents that plan in natural language, call tools/APIs, use memory/evaluation, and adapt to messy, open-world tasks (docs, web, tickets)

Agentic AI workflow = a process where an LLM-baed app executes multiple steps to complete a task

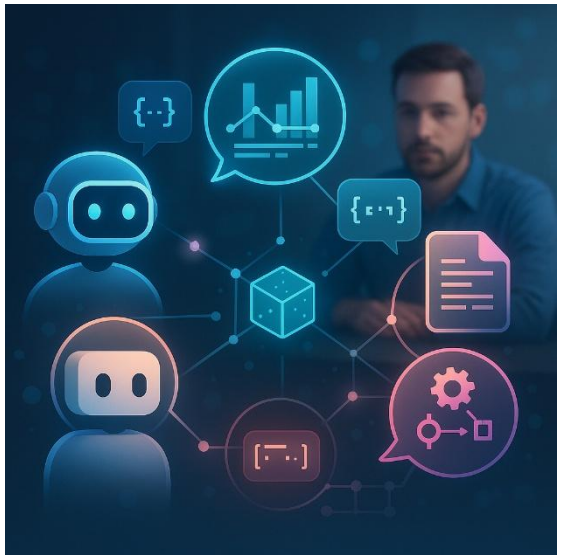
Distribution: Why and How?

Motivation

- Scale & parallelism: Split complex workflows across concurrent cooperating agents: higher throughput, shorter time-to-result.
- Latency & data locality: Place agents near users/data to cut round-trips and meet data-sovereignty/compliance needs.
- Specialization & composability: Each agent focuses on a capability/model/toolset; simpler to test, reuse, and evolve than monoliths.
- Robustness & governance: Isolate failures; enforce least-privilege tool access, auditing, budgets, and safe rollbacks.

Dimensions

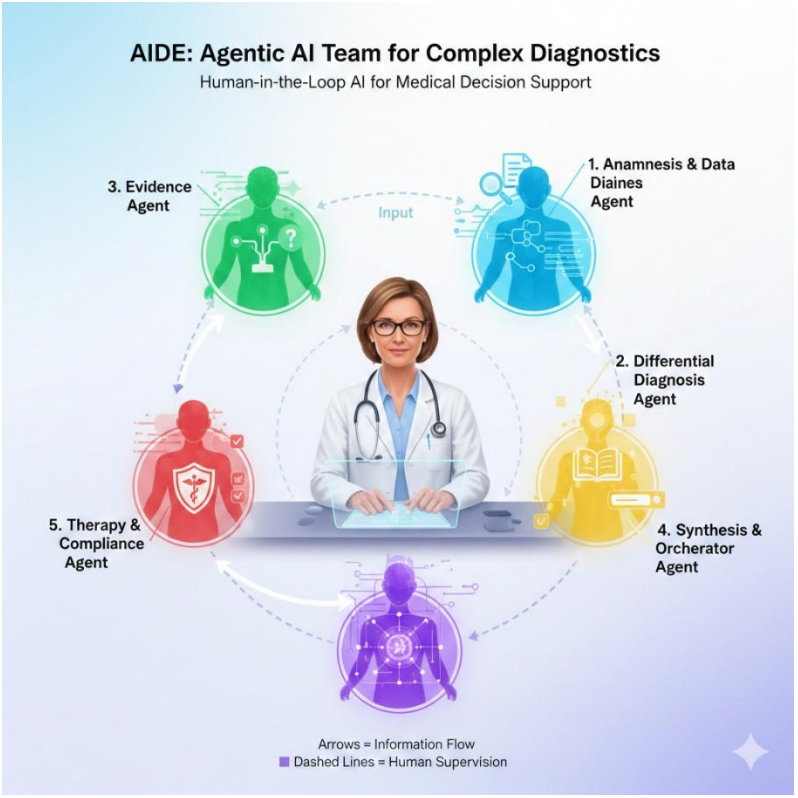
- Geographical – Edge, Cloud, On-Device
- Functional – specialized roles
- Data and knowledge – multiple sources and formats



UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

Example: a Team of Medical Agents

- Goal: Enhance diagnostic quality and efficiency in medicine
- Approach: Replace monolithic "Black-Box" models with a Multi-Agent Collaboration System
- Human-Centric: Strict "Human-in-the-Loop" architecture. The human physician retains final authority, supervision, and therapeutic decision
- Innovation: A framework for a legally compliant and transparent diagnostic process



The Team

Agent Role	Primary Function	Key Output
Anamnesis & Data Agent	Collects, cleans, and standardizes all patient data (e.g., to FHIR format).	Structured, consistent patient dossier.
Differential Diagnosis Agent	Generates weighted hypotheses for potential diseases based on the Dossier.	List of 3-5 prioritized hypotheses.
Evidence Agent	Validates hypotheses using up-to-date, external medical knowledge.	RAG-based evidence and source citations.
Synthesis & Orchestrator Agent	Resolves conflicts, integrates findings, and manages the entire agent workflow.	Final, justified diagnostic recommendation.
Therapy & Compliance Agent	Derives a personalized treatment plan and performs legal/ethical checks.	Final report including treatment and compliance verification.

Architectures

Architectural Approaches – Overview

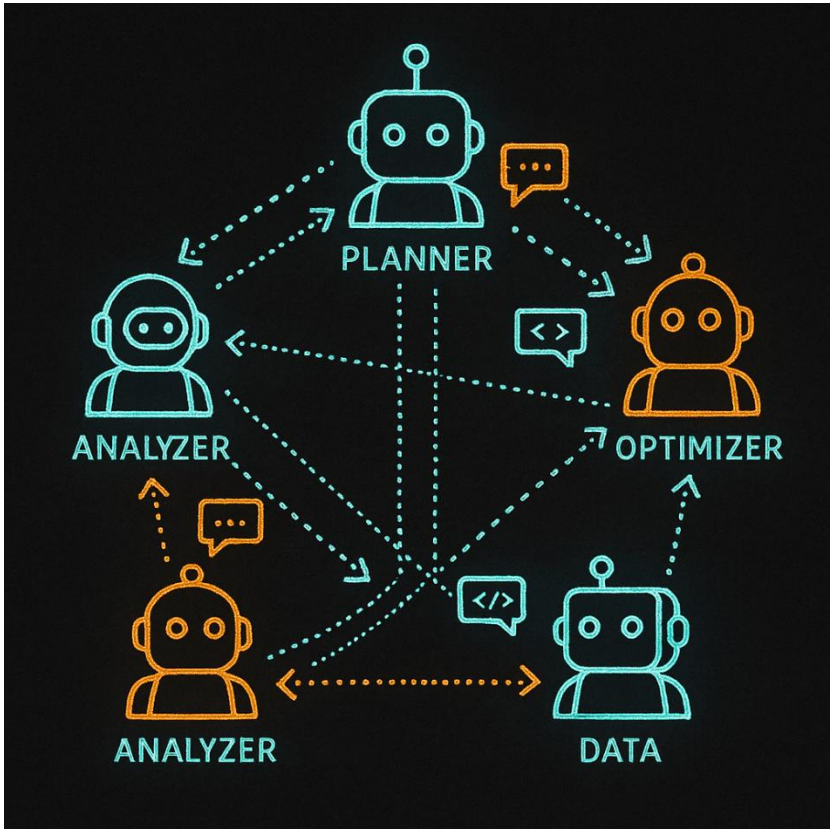
- Blackboard architecture – central knowledge store
- Peer-to-Peer – direct agent-to-agent communication
- Hierarchical – coordinator with sub-agents
- Service-oriented / API-based

Blackboard Architecture

- Central store ('blackboard') as coordination point
- Agents read/write shared information
- Pro:
  - Good for coordinated problem solving
- Con:
  - Asynchronous communication may delay the production of results

Peer-to-Peer Architecture

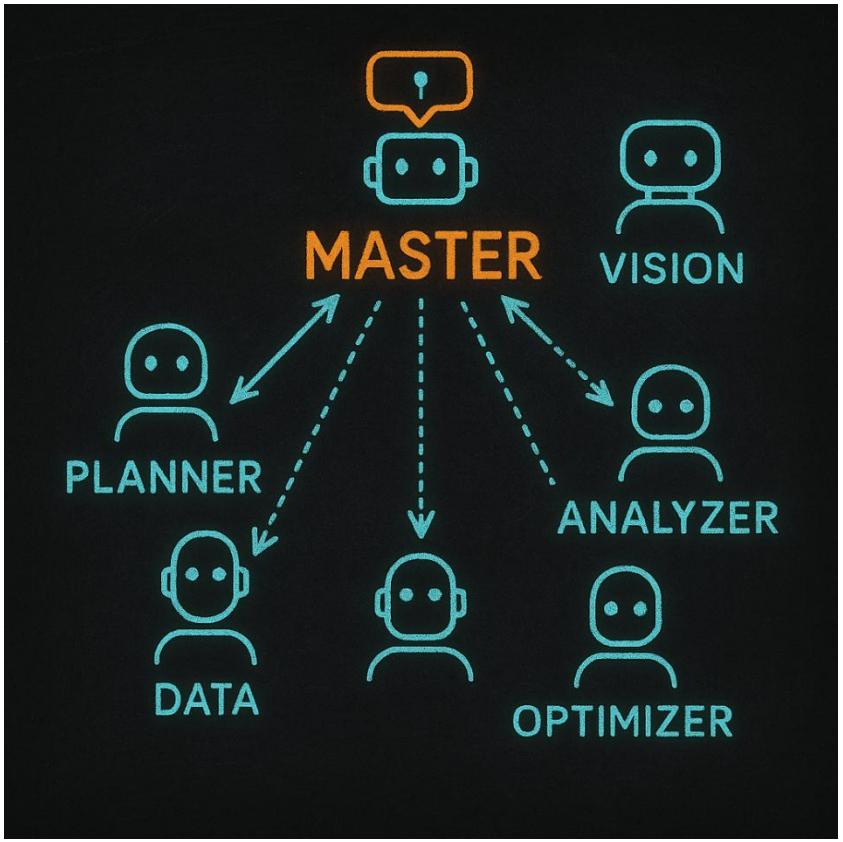
- Agents communicate directly with each other
- Pros:
  - High fault tolerance
  - Enables real-time decision making (synchronous communication!)
- Con:
  - Requires complex routing and synchronization



UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

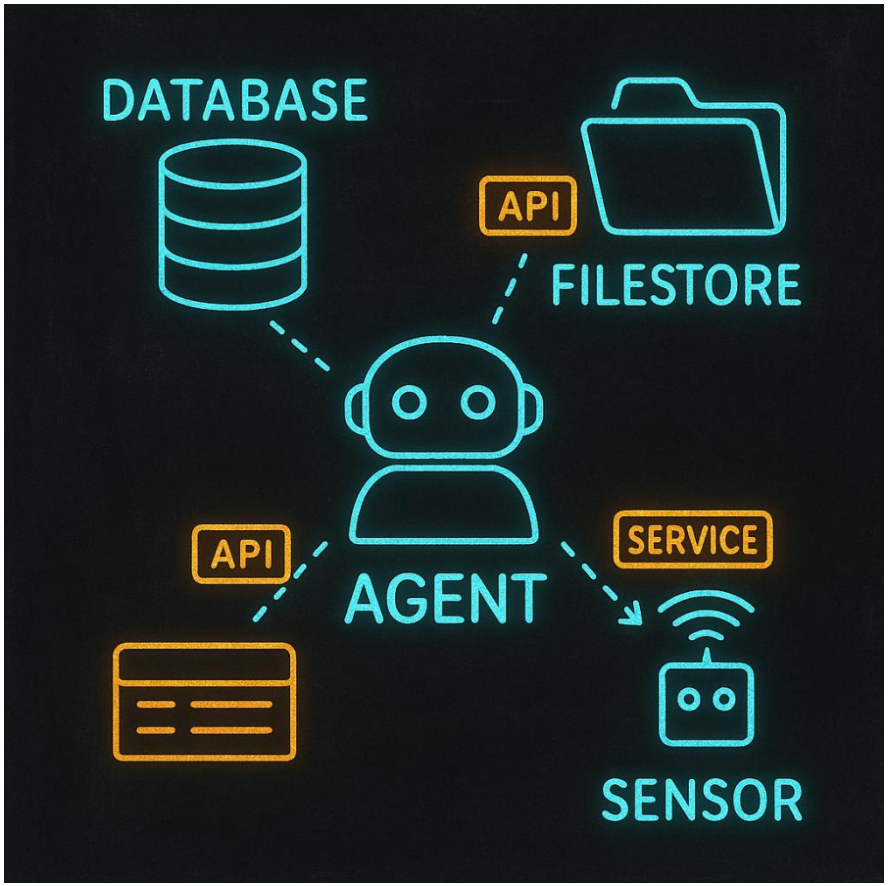
Hierarchical Architecture

- Central coordinator distributes tasks
- Collects results, generates the final answer
- Pro:
  - Fast decision-making, easier control
- Con:
  - Risk of single point of failure



Service-Oriented Architecture

- Access to additional services offered by the environment, such as documents, databases, or sensor measurements
- Also: agents as services with well-defined APIs
- Easy integration into existing systems
- Uses HTTP, gRPC, WebSockets or dedicated protocols like MCP (see later)



Hybrid Architectures

- Hybrid approach can make sense, depending on the application scenario
- Guidelines:
  - Blackboard for shared knowledge
  - P2P for real-time communication
  - API services for external tool integration

UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

Model Context Protocol MCP

Model Context Protocol – Motivation

- LLMs, as ANNs, have been trained at a specific point in time and are then used

→ As soon as they are finalized and deployed, they are aging and do not „know“ the newest information (after their „birth“).

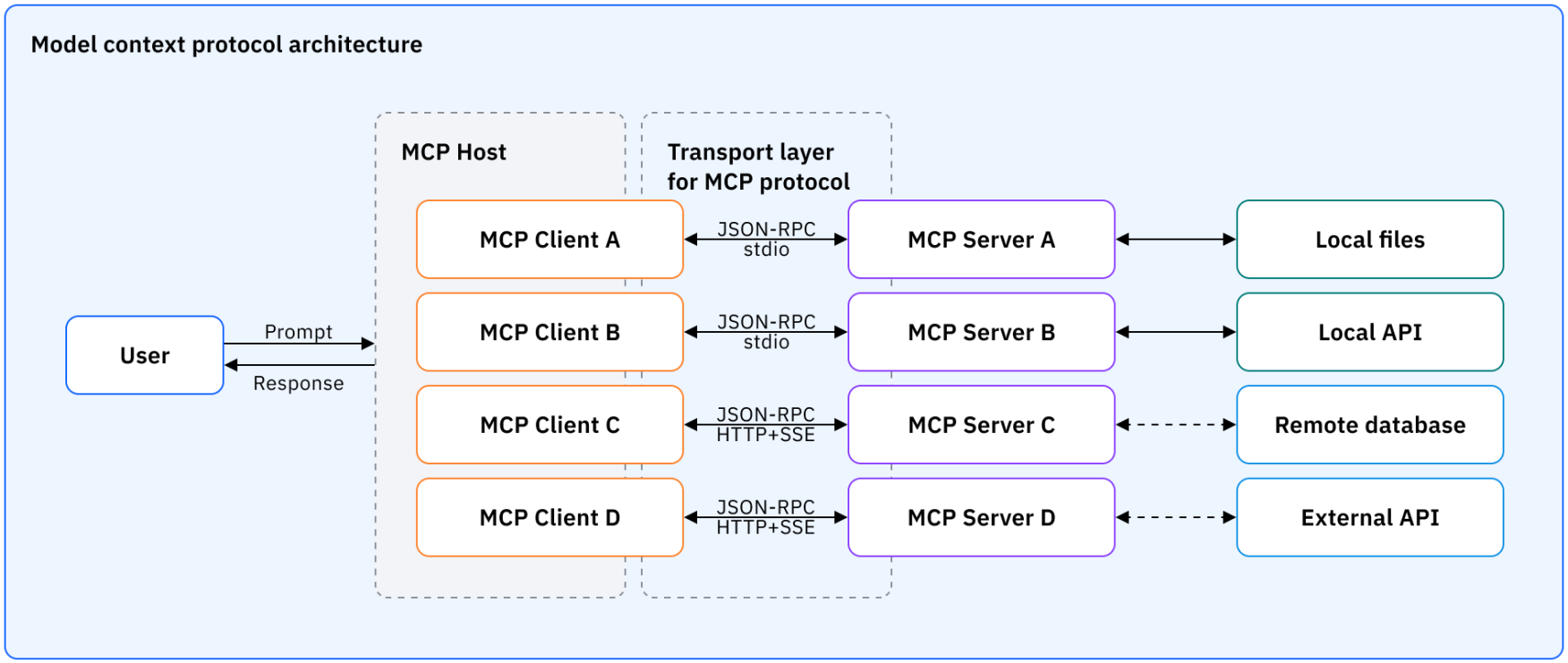
- Still, obviously, today‘s chat and other agents typically „know“ all the things that happened just recently – how to they do that?
- There has to be an additional mechanism that makes newer knowledge available to the agent.

MCP Characteristics and Functionality

- newer standard (2024/2025, initiated by Anthropic and now with growing ecosystem) that defines a **client-server protocol for exposing tools/services to models**
- Instead of each developer wiring in functions manually, tools are made available through MCP servers, and the LLM (or its hosting environment) can dynamically **discover and access** them.
- → tools = services of an MCP server
- Advantage:
  - Tools are portable and reusable (you don't reinvent your "search_flights" function every time).
  - Central discovery and authentication.
  - Works across multiple model providers.
- Limitation:
  - More overhead; needs infrastructure (MCP servers + clients)
  - still early in adoption.

UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

MCP Architecture



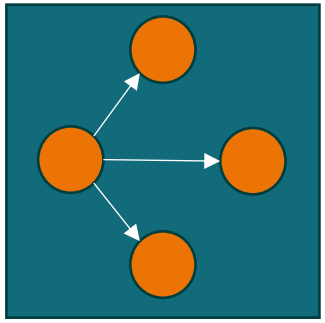
© https://www.ibm.com/think/topics/model-context-protocol#1509394338

UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

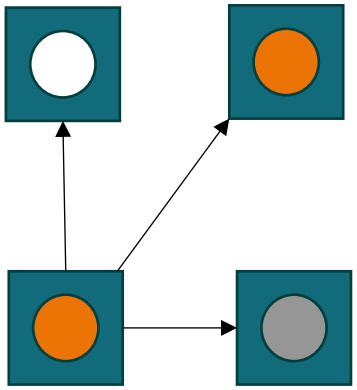
Agent-to-Agent Protocol A2A

Agent Scenarios

Two basic scenarios:



1. Homogeneous, non distributed
   - All agents of a team are operating in the same ecosystem, on the same platform.
   - a number of well-known and powerful platforms and programming environments, such as crewAI or LangChain
2. Heterogeneous, distributed
   - Some of the agents of a team are located in remote servers and operate in a different ecosystem
   - Even allows scenarios with agents not in a team, but with differing goals (negotiations etc.)
   - Need for a protocol to allow inter-agent communication → A2A



A2A: Introduction

- communication protocol for AI agents
- initially introduced by Google in April 2025, now hosted by the Linux Foundation as open source project.
- Designed for multi-agent systems, allowing for interoperability between AI agents from varied providers or built using different AI agent frameworks.
- Approaches such as crewAI and LangChain automate multi-agent workflows within their own ecosystems

  ←→

  the A2A protocol acts as a messaging tier that lets these agents “talk” to each other despite their distinct agentic architectures.

Core Components

- **User**: The end user or automated service initiating requests
- **A2A Client**: Application or agent that makes requests to remote agents on behalf of users
- **A2A Server**: AI agent or agent system implementing A2A protocol HTTP endpoints

UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

A2A Workflow: Communication

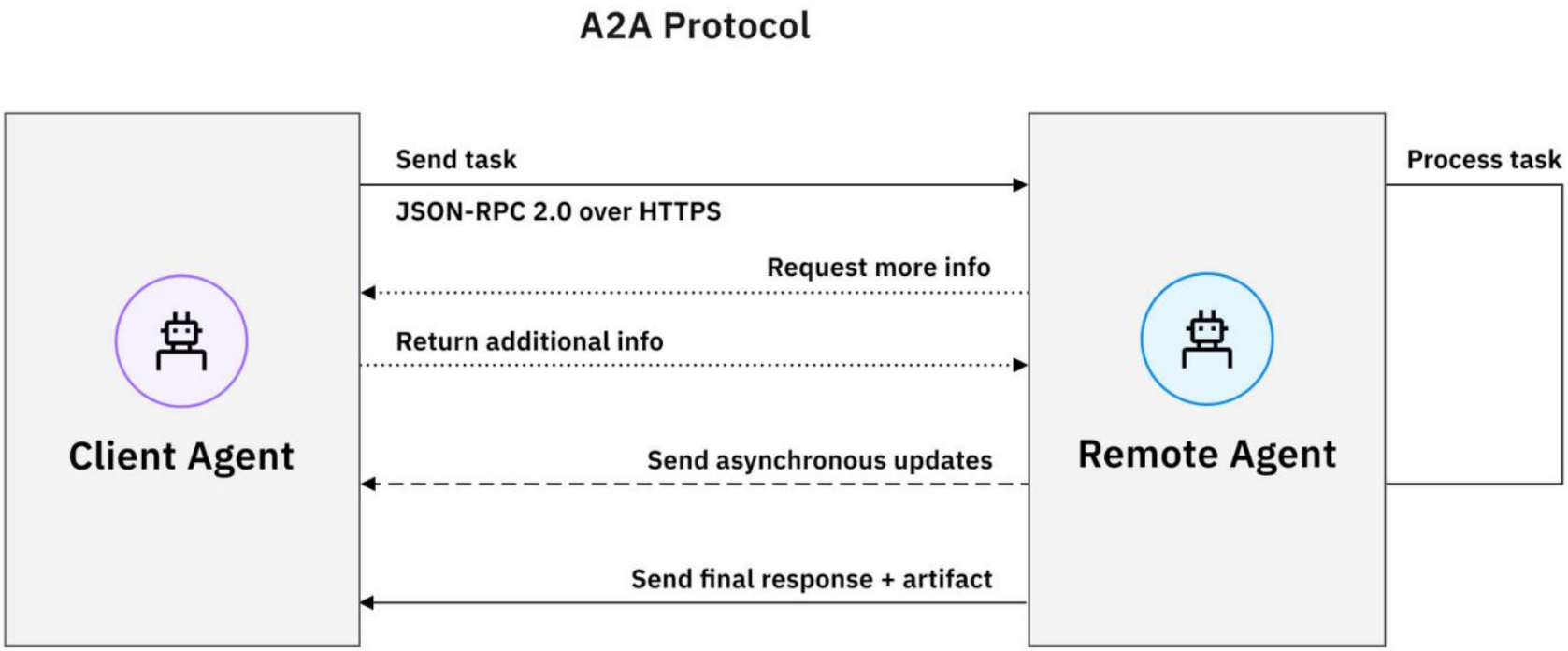


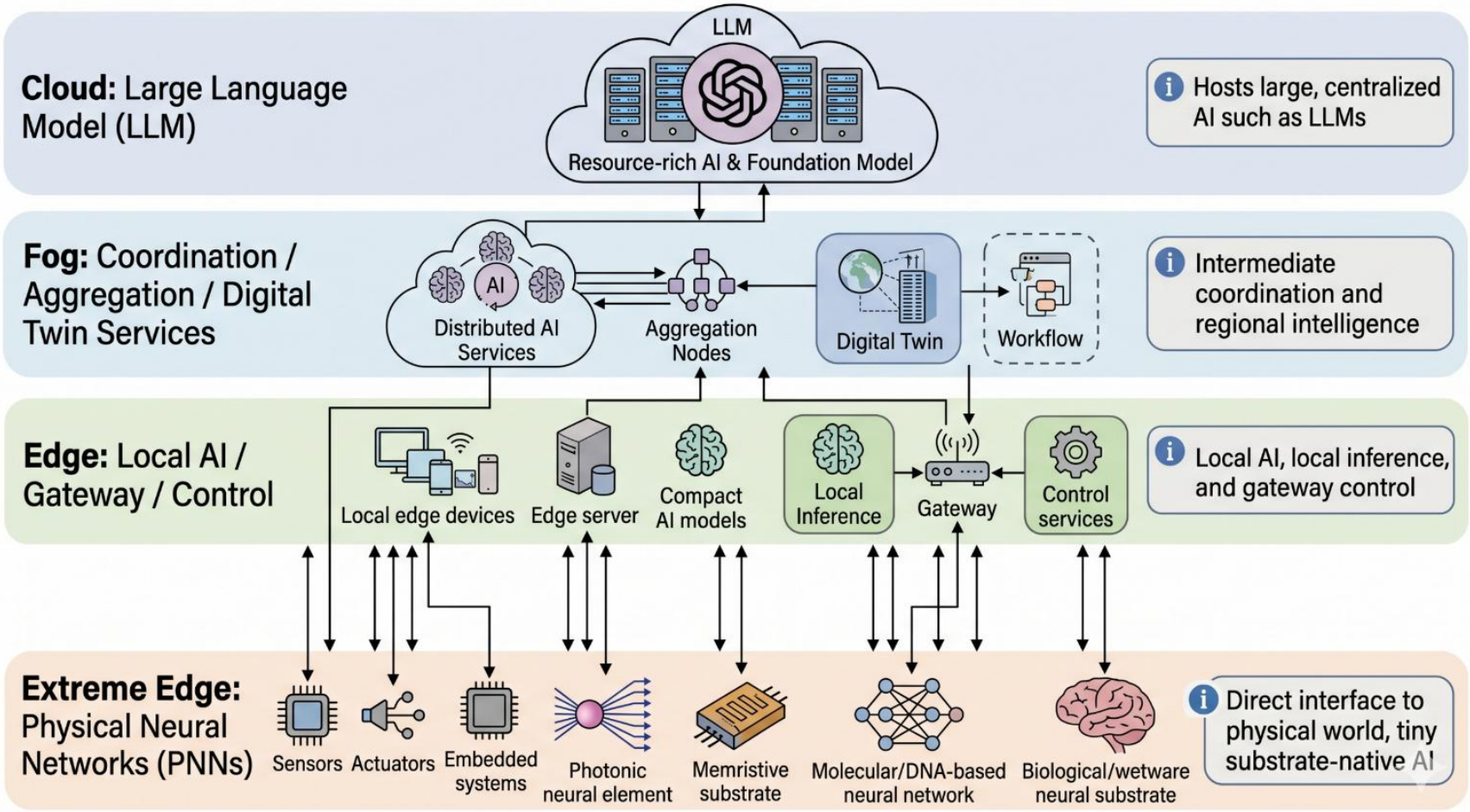
Image source: https://www.ibm.com/think/topics/agent2agent-protocol

UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

Broader Application Context in Edge-Cloud Architectures

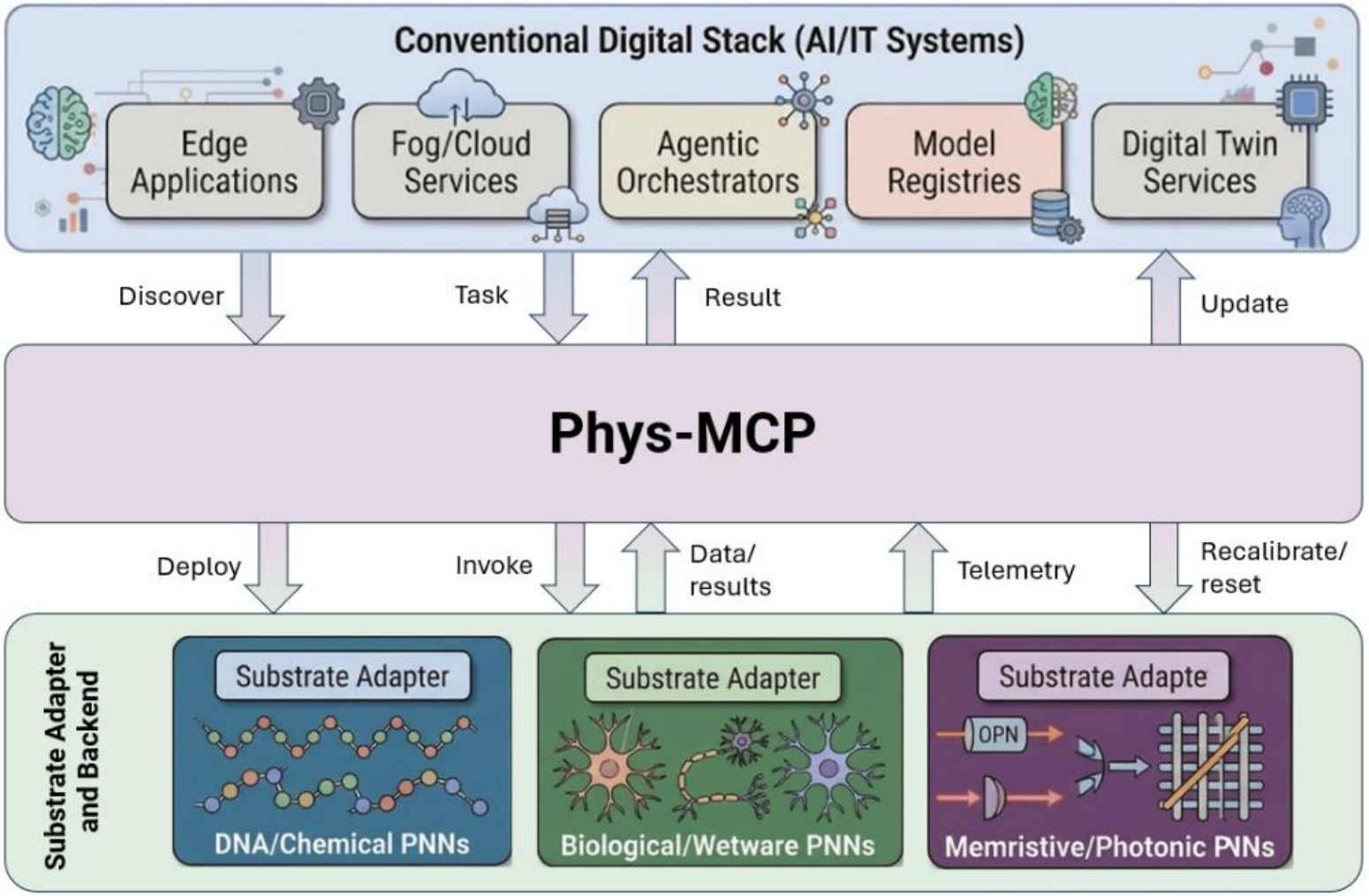
UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

Next Step: Edge-Cloud AI Systems



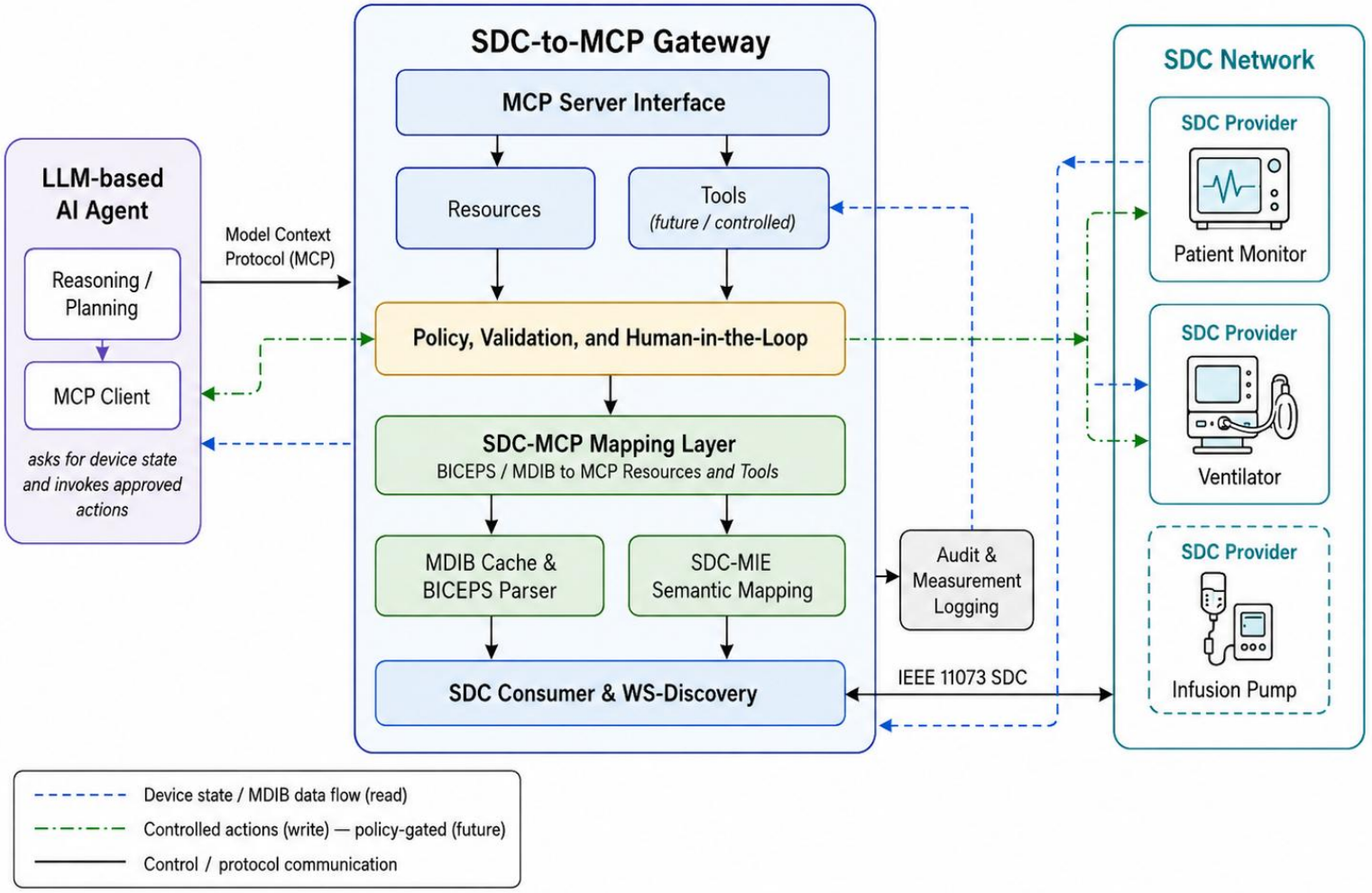
UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

Integrating PNNs in such systems

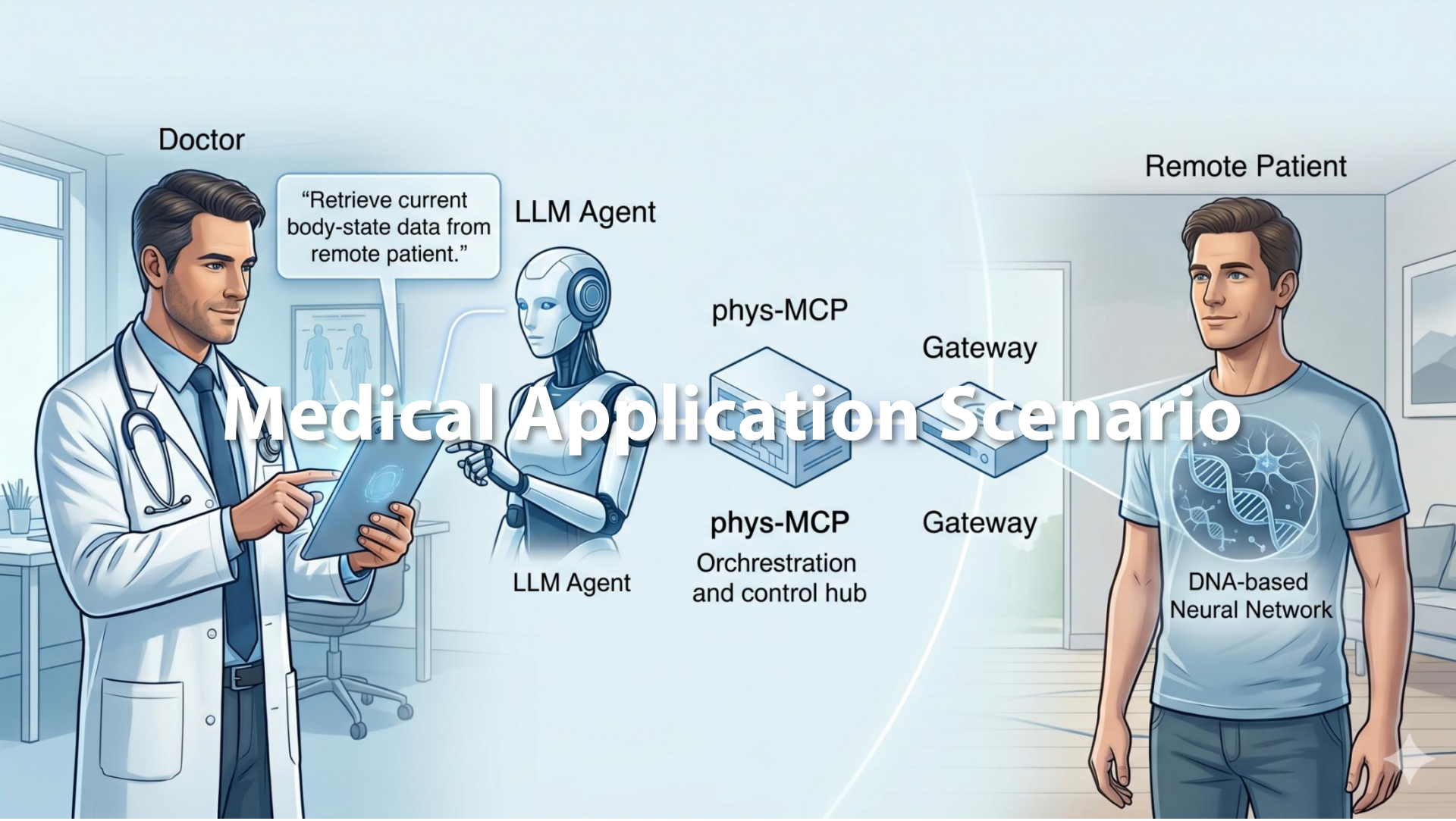


UNIVERSITÄT ZU LÜBECK
STIFTUNGSUNIVERSITÄT
SEIT 2015

An SDC-MCP Gateway



Medical Application Scenario



Kontakt

Prof. Dr. Stefan Fischer

University of Lübeck
Ratzeburger Allee 160
23562 Lübeck
Tel: +49 451 3101 - 6400 / 6406
Email: {stefan.fischer, f.lau}@uni-luebeck.de