AssetOpsBench

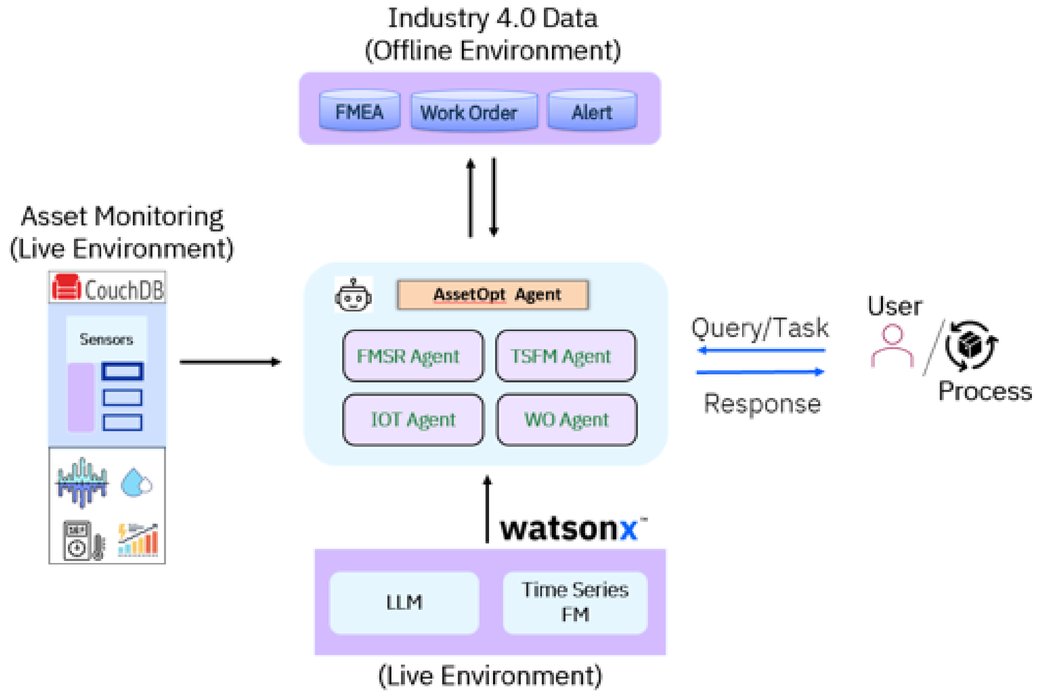
Contact: Dhaval Patel (pateldha@us.ibm.com)

https://arxiv.org/abs/2506.03828

AssetOpsBench – Opensource Benchmark for Industry 4.0 Automation

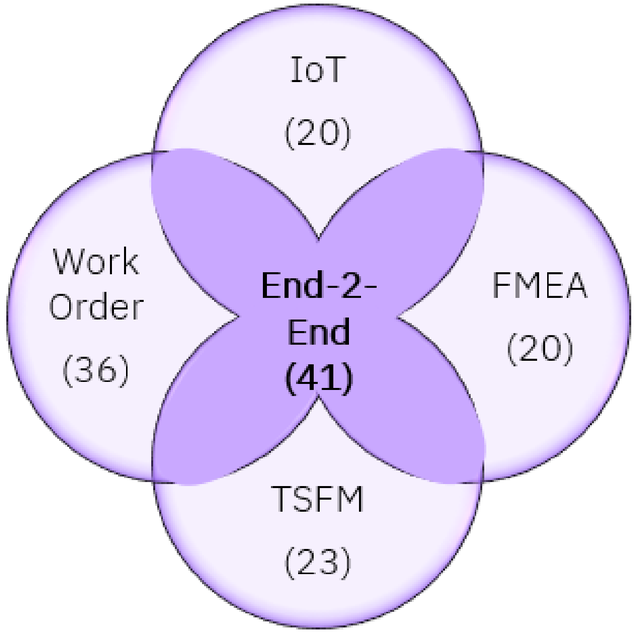
AssetOpsBench: Benchmarking AI Agents for Task Automation in Industrial Asset Operations and Maintenance

Dhaval Patel[1*] Shuxin Lin[1*] James Rayfield[1*] Nianjun Zhou[1*]
Roman Vaculin[1] Natalia Martinez[1] Fearghal O'donncha[2] Jayant Kalagnanam[1]
[1]IBM Research - Yorktown [2]IBM Research - Ireland
pateldha@us.ibm.com, shuxin.lin@ibm.com, jtray@ibm.com, jzhou@us.ibm.com,
vaculin@us.ibm.com, Natalia.Martinez.Gil@ibm.com, feardonn@ie.ibm.com,
jayant@us.ibm.com
*Equal contribution



- Framework to assess Gen AI solutions' ability to solve I4.0 Automation *"Scenarios"*: **June GA**
- **Simulated** industrial environment, **9 multi-source data sets** (work orders, FMEAs, timeseries) and **4 agents** (IoT, data science, work order, failure mode to sensor mapping)
- **140+** human-authored natural language queries**,** grounded in **enterprise industrial scenarios**
- **Agent harness**: systematic procedure for automated discovery of emerging failure modes

Distribution of 140+ scenarios across 4 agents



https://github.com/IBM/AssetOpsBench

AssetOpsBench : A Multi-Agent System (MAS) is at the core

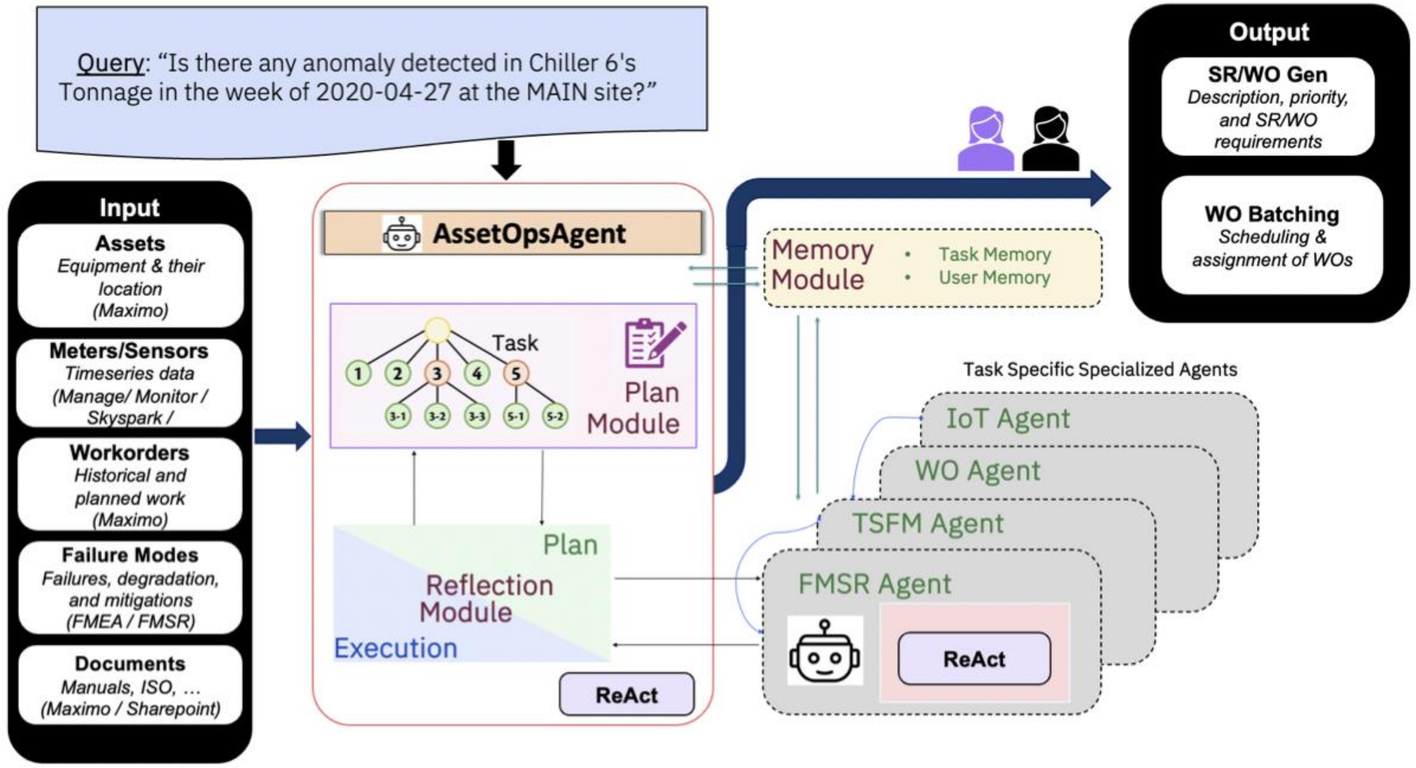
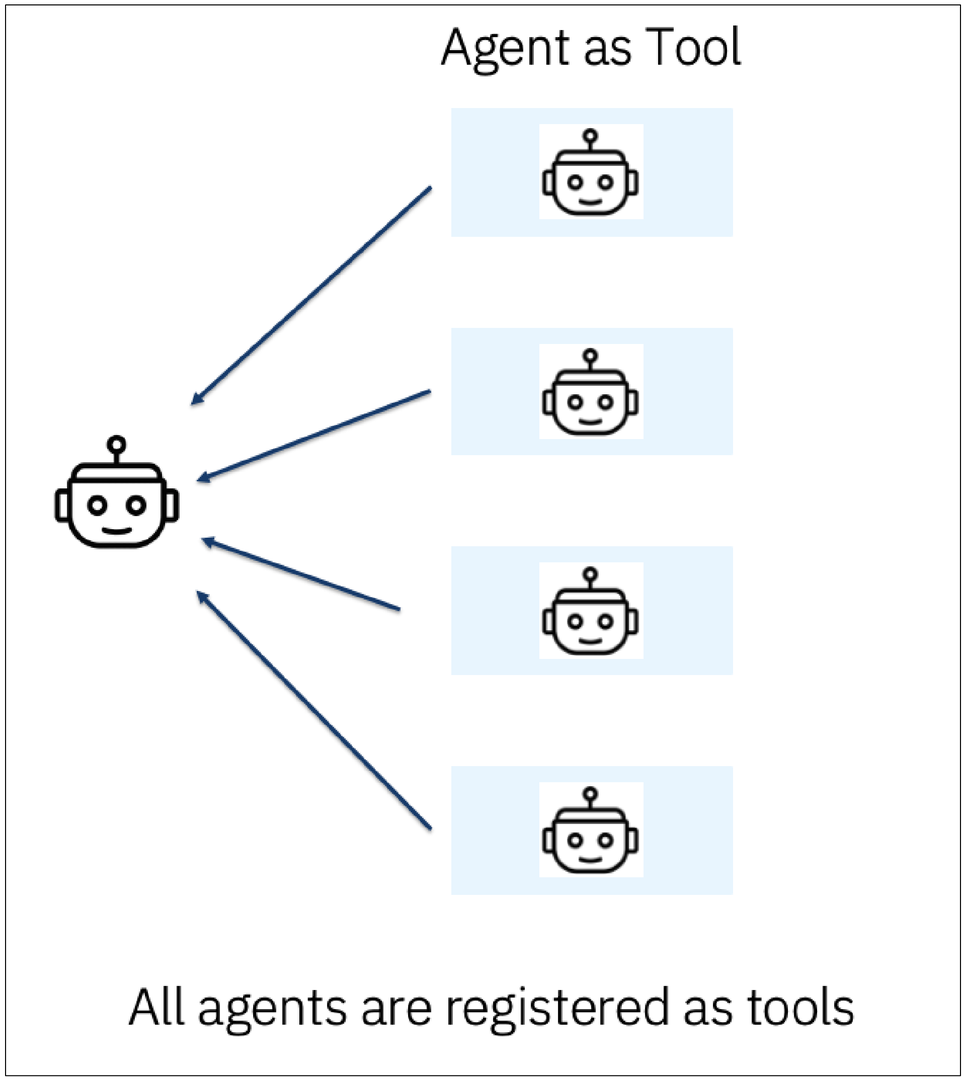
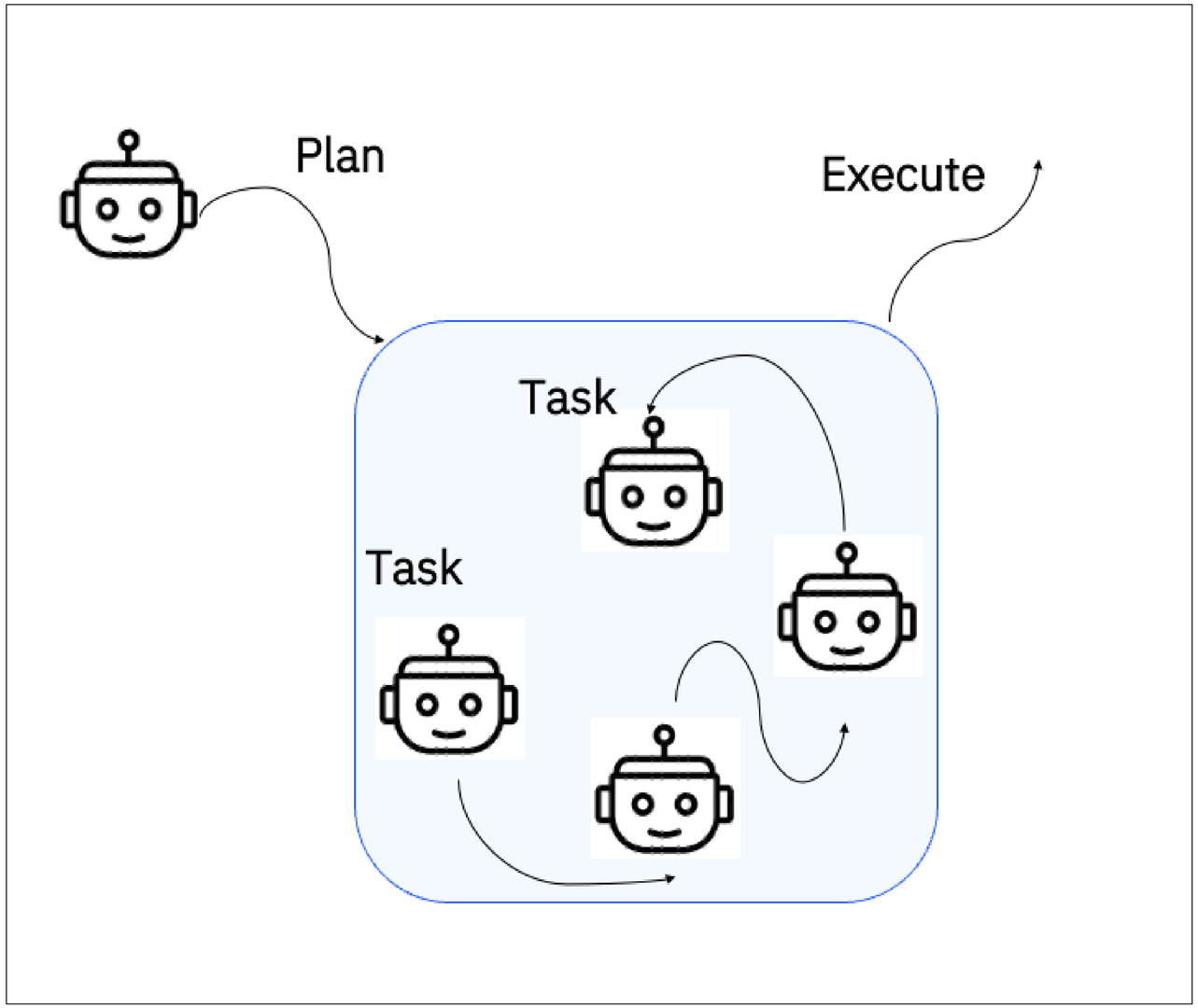


Figure 2: Architecture of the Multi-Agent System: **T**ime **S**eries **F**oundation **M**odel (TSFM) Agent, **F**ailure **M**ode **S**ensor **R**elations (FMSR) Agent, **W**ork **O**rder (WO) Agent

AssetOpsBench : Multi-Agent Implementation Strategy

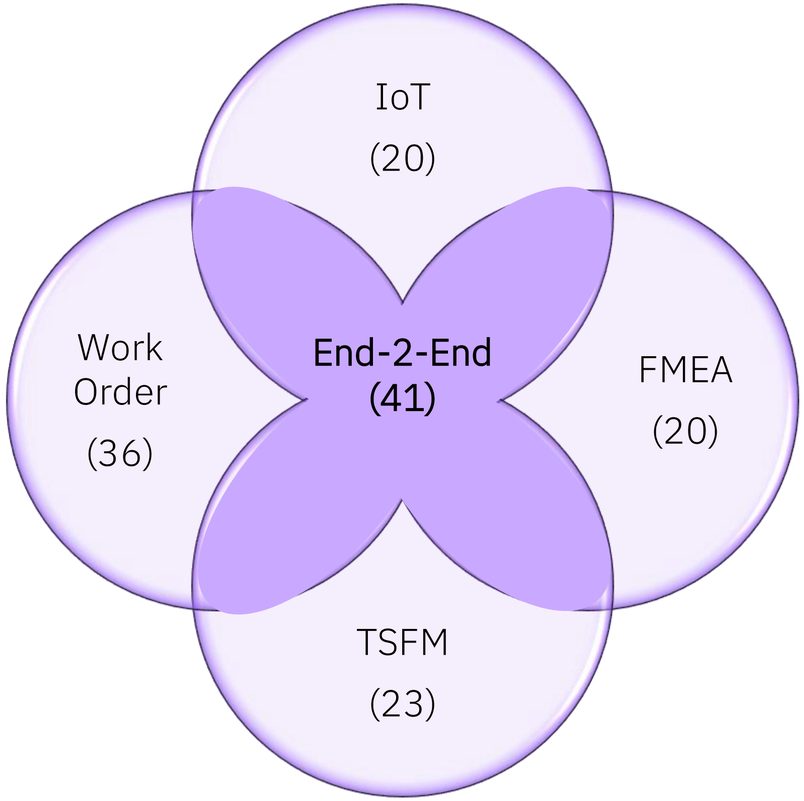


Agent-As-Tool Approach

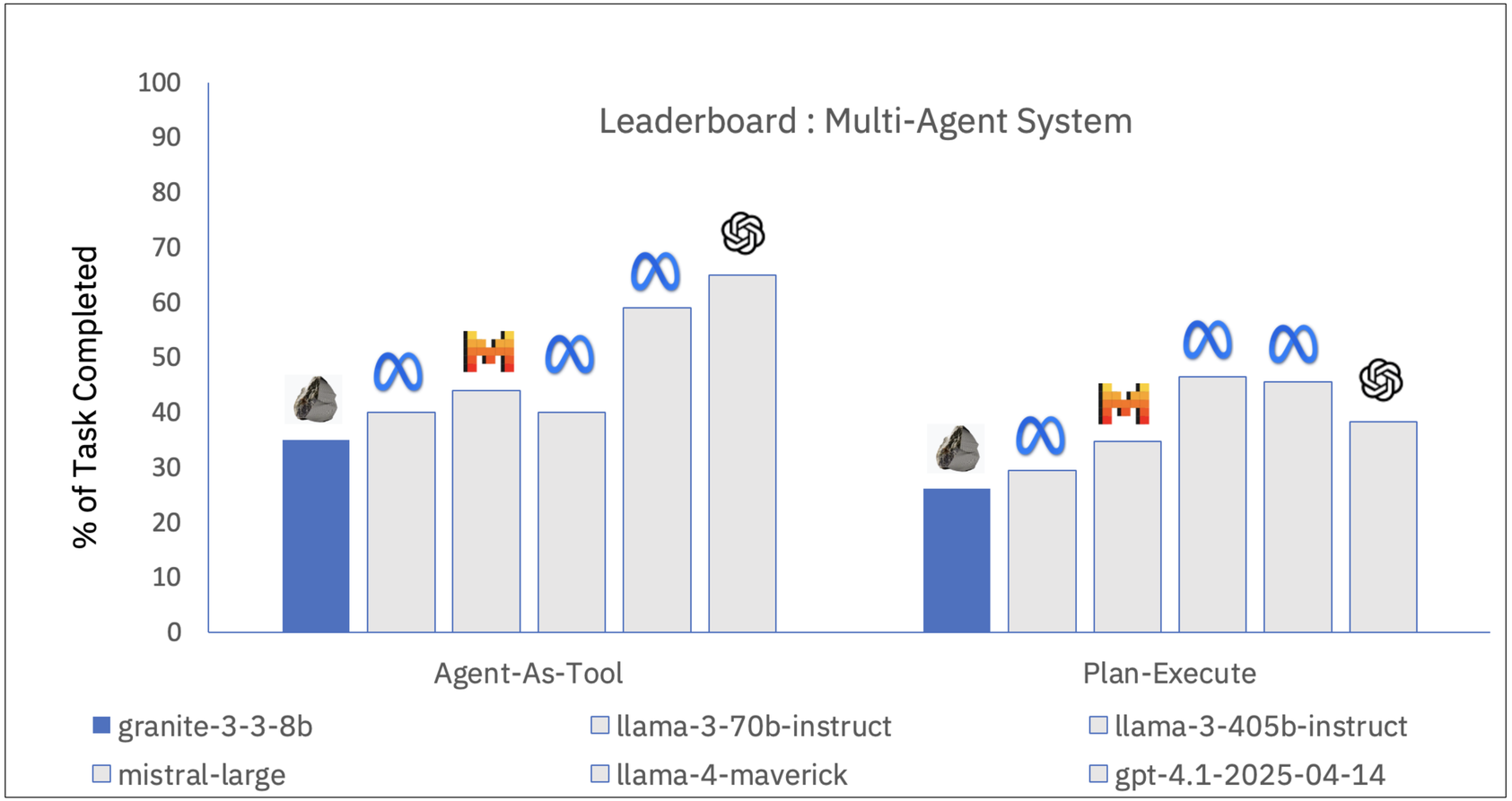


Plan-Execute Approach

AssetOpsBench : Open Source V1, June 2025



Distributions of 141 Scenarios across multiple agents



Extensive Evaluation of two different paradigm for Multi-Agent System

AssetOpsBench : Huggingface Dataset

```
from datasets import load_dataset

# Login using e.g. `huggingface-cli login` to access this dataset
ds = load_dataset("ibm-research/AssetOpsBench", "scenarios")
```

Datasets: ibm-research / **AssetOpsBench** | like 1 | Follow IBM Research 327 | Dataset card | Data Studio | Files and versions xet | Community 2 | Set

Subset (2)
scenarios · 141 rows

Split (1)
train · 141 rows

Search this dataset

id int64 1 – 622	type string · classes 4 values	text string · lengths 27 – 468	category string · classes 6 values	characteristic_form string · lengths 49 – 936 ø	deterministic bool 2 classes	note string · class 2 values
220	TSFM	Finetune a forecasting model for 'Chiller 9 Condenser Water Flow'…	Tuning Query	The finetuned forecasting model is saved in save_model_dir=tunedmodels with result stored in…	null	
221	TSFM	Finetune a forecasting model for 'Chiller 9 Condenser Water Flow'…	Tuning Query	The finetuned forecasting model is saved in save_model_dir=tunedmodels with result stored in…	null	
222	TSFM	I need to perform Time Series anomaly detection of 'Chiller 9…	Anomaly Detection Query	The anomaly detection results are stored in file data/tsfm_test_data/tsad_conformal.csv	null	
223	TSFM	Find and run several methods to analyze data sensor 'Chiller 9…	Complex Query	The forecasting results for 'Chiller 9 Condenser Water Flow' using data in…	null	
400	Workorder	Get the work order of equipment CWC04013 for year 2017.	null	There will be 33 records. The expected response should retrieve all work orders for equipment…	true	
401	Workorder	I would like to check the work order distribution for the equipment…	null	Work order with primary Code MT010 occurred 3 times and code MT013 occurred once. The expected respons…	true	

https://huggingface.co/datasets/ibm-research/AssetOpsBench?library=datasets

AssetOpsBench : AI Agentic Challenge

ASSETOPSBENCH

1,00,000 INR

Edit | Participants | Submissions | Dumps | Migrate

ORGANIZED BY: CODS-2025-AI-Agent-Challenge
(pateldha@us.ibm.com)
CURRENT PHASE ENDS: 1 November 2025 At 08:00 GMT-4
CURRENT SERVER TIME: 2 October 2025 At 23:07 GMT-4

Docker image: quay.io/assetopsbench/assetopsbench-basic
Secret url: https://www.codabench.org/competitions/10206/?secret_key=1e1a39c4-cf61-4cb3-8854-d5fde008f4bd
Competition Report: https://arxiv.org/pdf/2506.03828

Sep 2025 | Oct 2025 | Nov 2025

Get Started | Phases | My Submissions | Results | Forum

- Introduction
- Resources
- Track 1: Task Planning
- Track 2: Task Execution
- Team Registration
- Terms

🚀 Understanding the AssetOpsBench

Here are a few key resources to help you dive deeper into **AssetOpsBench** ⬇️

📂 GitHub Repository

🔗 IBM/AssetOpsBench
Main codebase, datasets, and benchmark setup.

🔗 IBM/ReActXen

Get Started | Phases | My Submissions | Results | Forum

- Introduction
- Resources
- Track 1: Task Planning
- Track 2: Task Execution
- Team Registration
- Terms
- Files

🚀 Track 1: Planning-Oriented Multi-Agent Orchestration

🎯 Motivation

In **single-agent** settings, reasoning styles such as **ReAct**, **Reflect**, or **Chain-of-Thought** can effectively solve
Participant can learn about single agent system Here.

However, in **multi-agent** settings, the problem becomes more complex:

- 🤖 Which set of agents should be selected?
- 🔗 In what sequence should they be called?
- 📝 How do we organize communication and dependencies?
- ❓ What questions should be asked of each agent?

Get Started | Phases | My Submissions | Results | Forum

- Introduction
- Resources
- Track 1: Task Planning
- Track 2: Task Execution
- Team Registration
- Terms
- Files

🚀 Track 2: Execution-Oriented Dynamic Multi-Agent Workflow

🎯 Motivation

Current challenge code uses a **SequentialWorkflow**, where tasks are executed strictly in order. While this works for pipelines, it has several limitations in multi-agent settings:

- 🤖 Only **one agent per task** is supported.
- 🔗 Execution is **linear**, unable to adapt dynamically to task outputs.
- 📝 Context handling is rigid — cannot easily combine information from multiple previous tasks.
- ⚠️ Bottlenecks occur if one task fails or is delayed.

The goal of Track 2 is to **introduce helper agents** and design a **DynamicWorkflow** that:

- Allows tasks to execute **non-linearly** (parallel or conditionally).
- Supports **multiple agents per task** for collaboration or fallback.
- Dynamically manages context across tasks to improve reasoning and efficiency.

Think of it as **moving from a sequential assembly line to a flexible, adaptive multi-agent factory floor.**

🛠️ Baseline Workflow

We provide a **baseline SequentialWorkflow** as a starting point:

👉 `SequentialWorkflow` Implementation

This baseline executes tasks in order and supports different **context types** (`DISABLED`, `ALL`, `PREVIOUS`, `SELECTED`
It serves as a **scaffold**, not a fully flexible solution.

https://www.codabench.org/competitions/10206/