

Use case

Development Data
Transformation (DX):
Leveraging the power of AI
with structured data

Whitepaper by ASAM e.V.

Apr 07, 2026

Table of Contents

1. Executive Summary	3
2. Benefits	3
3. How it works	4
4. Decisions that need to be taken	5
5. System Components.....	5
6. Requirements.....	6
7. Conclusion	6

Disclaimer:

This document is the copyrighted property of ASAM e.V. This publication may not be reproduced, distributed, or transmitted in any form or by any means without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. In case of any discrepancy in the technical content, the reader should always refer to the official specification documents for both standards as the definitive source of information.

For permission requests, please contact: info@asam.net.

1. Executive Summary

Modern development environments require constant real-time measurement, evaluation, and analysis of data. Data can come in different ways and formats and must be made available without delay. Engineers must handle numerical data as well as unstructured information such as requirement documents, specifications, and test reports, at the same time. AI-powered development workflows, from natural language query to automated summarization and anomaly detection, are increasingly seen as solutions for these challenges.

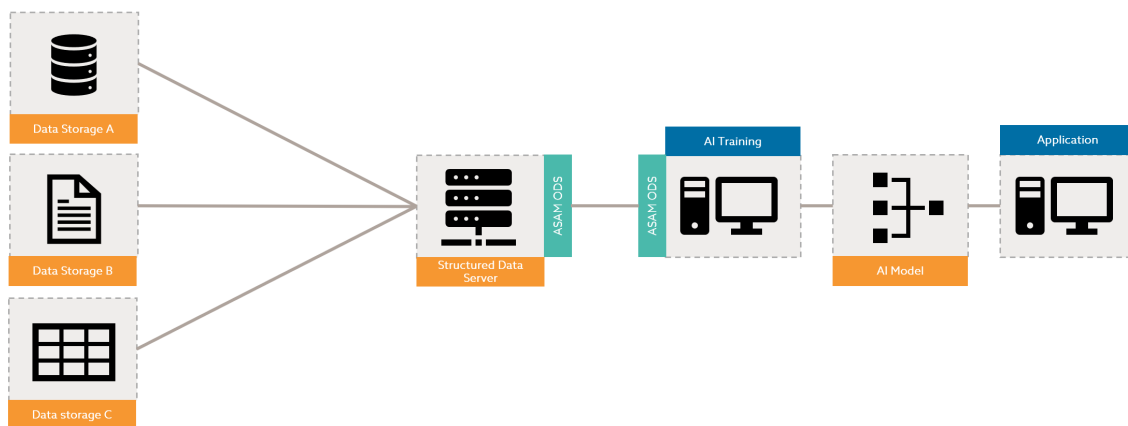


Figure 1: Training AI models efficiently by structuring unstructured data sources

But AI tools can only provide reliable support if they can access consistent, well-structured, and traceable data (

Figure 1). If the underlying data is incomplete or fragmented across systems, this directly limits the usefulness of any AI-based analysis or assistant.

Building on the ASAM ODS data foundation described in our paper "Data management for compliance purposes", this document explains how the same standardized and traceable measurement data can be made available to AI tools for development support. By systematizing measurement data in ASAM ODS and linking it with relevant documents, AI-based tools can answer questions, highlight patterns, and propose next steps, while engineers maintain full transparency over the data behind each result.

This document examines the use of AI on top of an ASAM ODS-based test data management system, focusing on data access and analysis rather than model development.

2. Benefits

- Systematizing measurement data and metadata in ASAM ODS creates a structured environment that is well-suited for AI-based access, search, and analysis.
- Centralized access control and metadata management make it easier to define which data AI tools can access. Documenting the scope of access reduces the risk of unintended gaps or unwanted exposure.

- Since data is managed in a standardized way, AI tools can access the same ODS data as other analysis applications, and engineers can drill down from AI responses to the underlying measurements using any ASAM ODS-compatible client.
- As data accumulates on the ASAM ODS server, AI assistants can compare current results with past tests, highlight trends, and point out deviations without manual searching across multiple systems.
- By combining ODS metadata with associated documents (requirements, procedures, reports), AI tools can summarize context, extract key information, or propose metadata/tag suggestions, significantly reducing manual work for engineers and analysts.

3. How it works

Store measurement data and related documents in an ASAM ODS server

- Measurement data and its metadata are stored on the ASAM ODS server, typically after being collected in ASAM MDF format as described in the compliance paper.
- Where necessary, references to related documents—such as requirement documents, test procedures, and test reports—are stored on the same server. The ASAM ODS model includes mechanisms for referencing external data, allowing both the structured metadata in ODS and the unstructured documents it references to be presented together as context for AI tools.

Connect AI tools to the ODS-based data foundation

- An integration layer (for example, an MCP-based connector or other middleware) is used to expose a controlled set of ODS queries and operations to AI systems.
- Instead of accessing databases directly, AI tools call well-defined functions such as “search tests by attribute”, “retrieve measurement channels”, or “get documents linked to this test object”, which internally use the ASAM ODS APIs.
- This approach ensures that all AI access respects existing ODS permissions and governance, and that changes to backend systems or ODS products do not require changes in the AI tools themselves.

Use AI-assisted analysis in development workflows

Typical examples of how engineers can use AI on top of ODS include:

- Asking natural-language questions such as “Show me durability tests for this model with temperature issues” and having AI translate them into ASAM ODS queries and present the relevant results.
- Requesting summaries like “Compare the latest emissions test results to the previous campaign” and receiving a structured overview with links back to the underlying ODS data and documents.
- Letting AI highlight anomalies or unusual patterns in measurement data retrieved from ASAM ODS and then investigating those directly in existing analysis tools.

In all cases, the source of information remains the ODS-based test data foundation, and engineers can trace AI responses back to concrete measurements and documents.

4. Decisions that need to be taken

Decide which data AI tools are allowed to access

It is essential to clearly define which ODS data and documents can be exposed to AI tools and for which user groups. This includes determining whether specific domains, projects, or historical data should be excluded due to business, confidentiality, or compliance reasons.

Define how AI results are used and verified

Human users must determine how AI-generated suggestions, summaries, or analyses are used in development. AI responses are not guaranteed to be correct, so an appropriate level of review and validation must be defined for each use case.

Determine the level of detail returned to AI and to users

For some scenarios, it may be sufficient to provide aggregated, anonymized, or filtered data to AI tools, while detailed raw data remains accessible only via specialized analysis applications. These rules should be defined in line with internal policies and regulations.

Clarify how to handle historical and legacy data.

Older data and documents may be less relevant to current products or may follow different standards. It is therefore necessary to decide which historical information should be considered by AI tools and how it should be distinguished from current development data.

5. System Components

ODS server + file server/document management server > AI > Front-end software

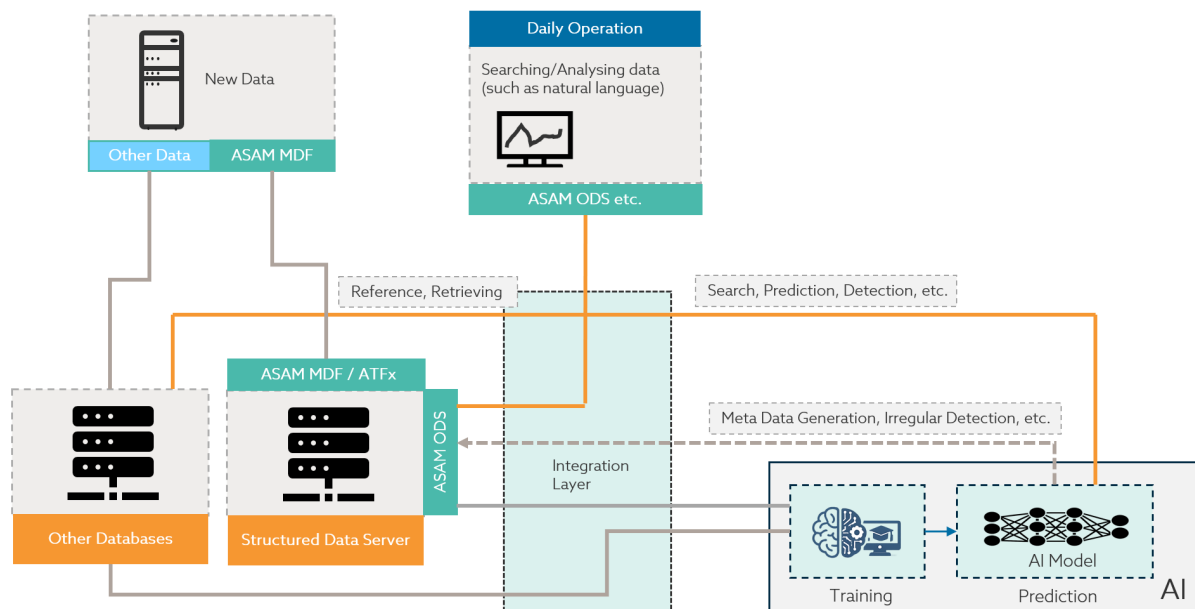


Figure 2: System components in a layout for ODS-enhanced AI tools

Building on the ASAM ODS-based test data foundation described in the compliance paper, an integration layer and AI components are added. AI tools never bypass ODS; instead, they use the same standardized APIs and permissions as other applications, ensuring consistent governance and traceability.

6. Requirements

When using AI on top of an ODS-based test data management system, it is important to:

Define and document the AI integration architecture

Decide which ODS functions are exposed to AI tools via the integration layer, how authentication and authorization are handled, and how AI access is logged.

Establish governance for the use of AI and data exposure

Assign responsibilities for deciding which data and documents AI tools may access, how prompts and responses are monitored, and how issues (e.g., inappropriate suggestions) are handled.

Ensure users understand the role and limitations of AI

Developers, testers, and analysts must be trained and informed that AI tools provide assistance based on ASAM ODS data, but do not replace engineering judgment. They should know how to drill down to the underlying ASAM ODS data when needed and how to report questionable AI outputs.

7. Conclusion

By leveraging AI tools on top of a standardized ASAM ODS data foundation, organizations can significantly reduce the effort required for data search, comparison, and basic analysis, and can provide engineers with faster, more intuitive access to relevant information. At the same time, because AI access is routed through the ODS-based test data management system, existing compliance, traceability, and access control mechanisms remain in effect.

The systematization of measurement data and metadata using ASAM ODS — together with the ASAM MDF and ASAM ODS-based integrity measures described in our paper "Data management for compliance purposes" — greatly improves the quality, traceability, and explainability of the data that AI tools work with. As a result, companies can increase confidence in AI-assisted development decisions while maintaining a consistent, vendor-independent foundation for transparent and traceable test data.