

AI-Optimized AESI Flagging: A Metadata-Driven Architecture for Performance and Compliance

PHUSE US Connect 2026 | SM05

March 24, 2026

Hang Zhong, Rashad Rasual, Jiaqi Hu | Bristol Myers Squibb

Disclaimer

This presentation shares only the author's experience and opinions, and not necessarily those of Bristol Myers Squibb (BMS). BMS does not guarantee the accuracy or reliability of the information provided in the slides.

Acknowledgements

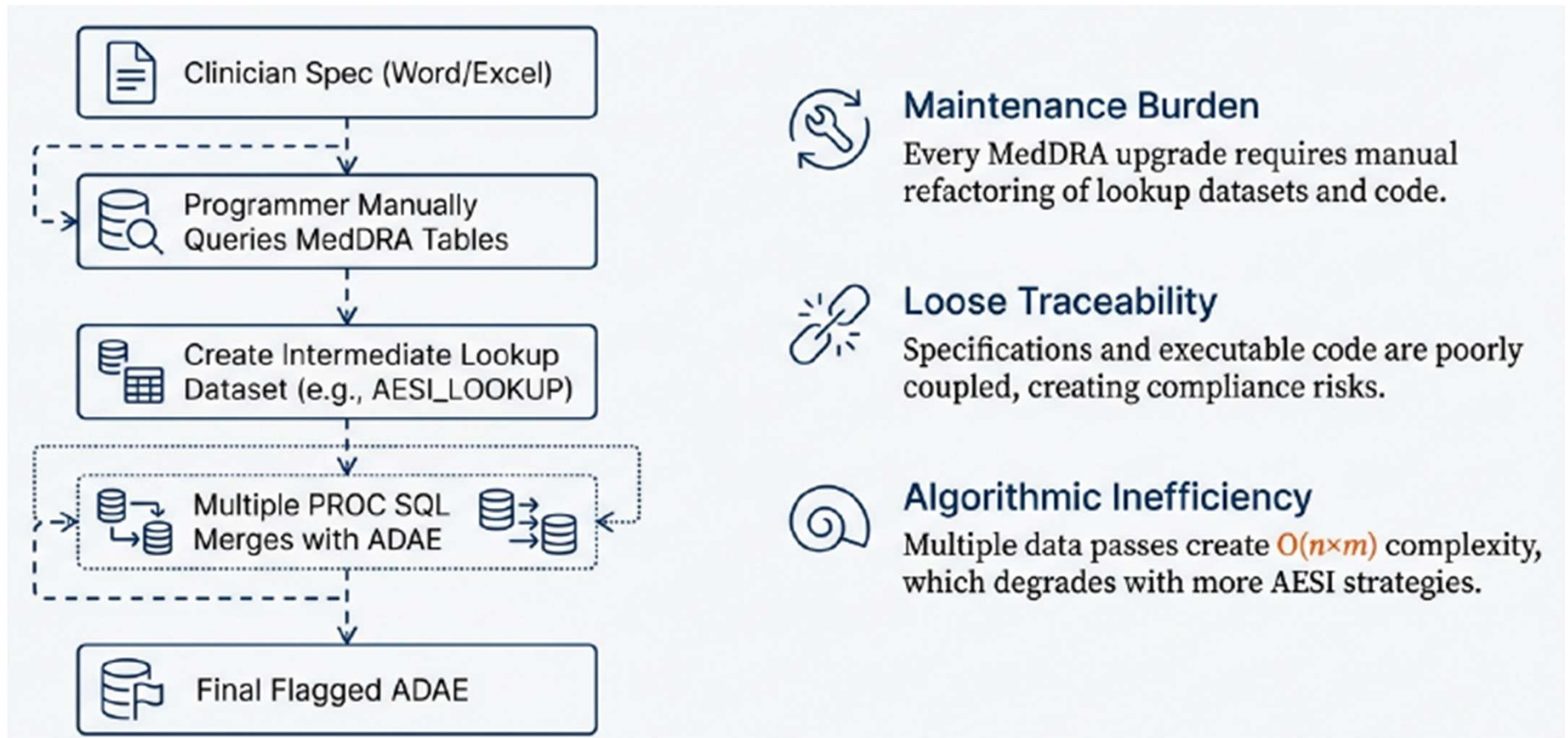
- **Special Thanks**

- Rashad Rasul - Technical guidance, methodology refinement and invaluable feedback throughout the development
- Jiaqi Hu - Specification alignment and cross-project consistency
- Samar Noor - Publication governance and oversight

- **BMS Collaborators**

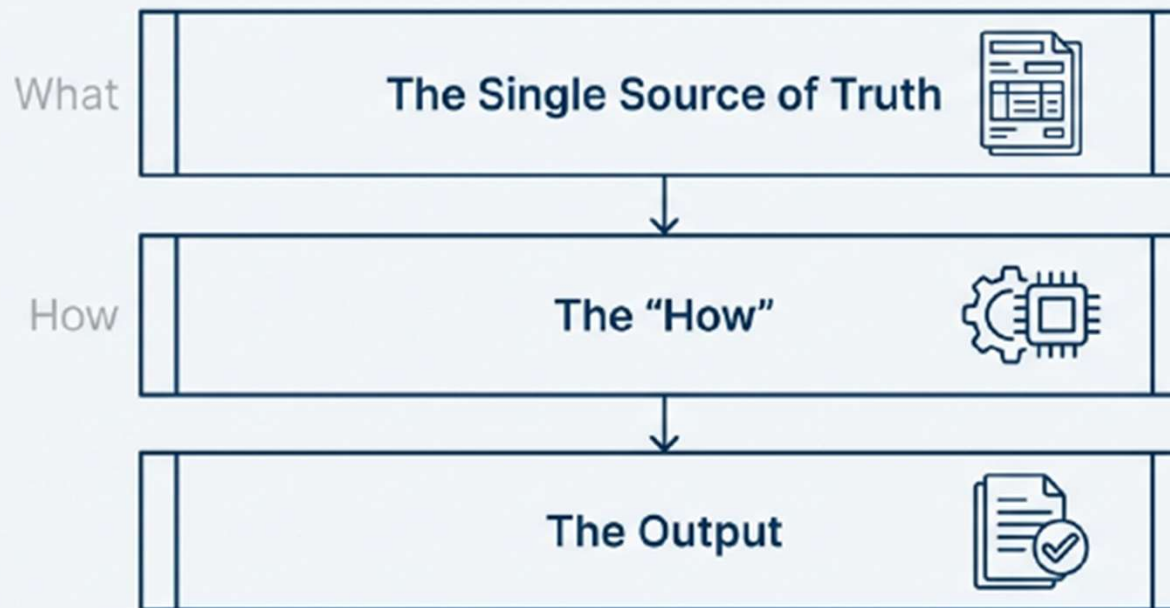
- Xiaohan Zou - Methodology and GBDS best-practice alignment
- Zhouzhou (Joe) Xi - Cross-functional review
- Lei Zhang - Programming strategy

The AESI Challenge



Our Solution: A Three-Tier, Metadata-Driven Architecture

We implemented a three-tier architecture separating specifications ('What') from implementation logic ('How').



This ensures 100% concordance between submission documentation (Define.xml) and the executed code, a critical regulatory requirement.

Deep Dive: The Machine-Executable Specification



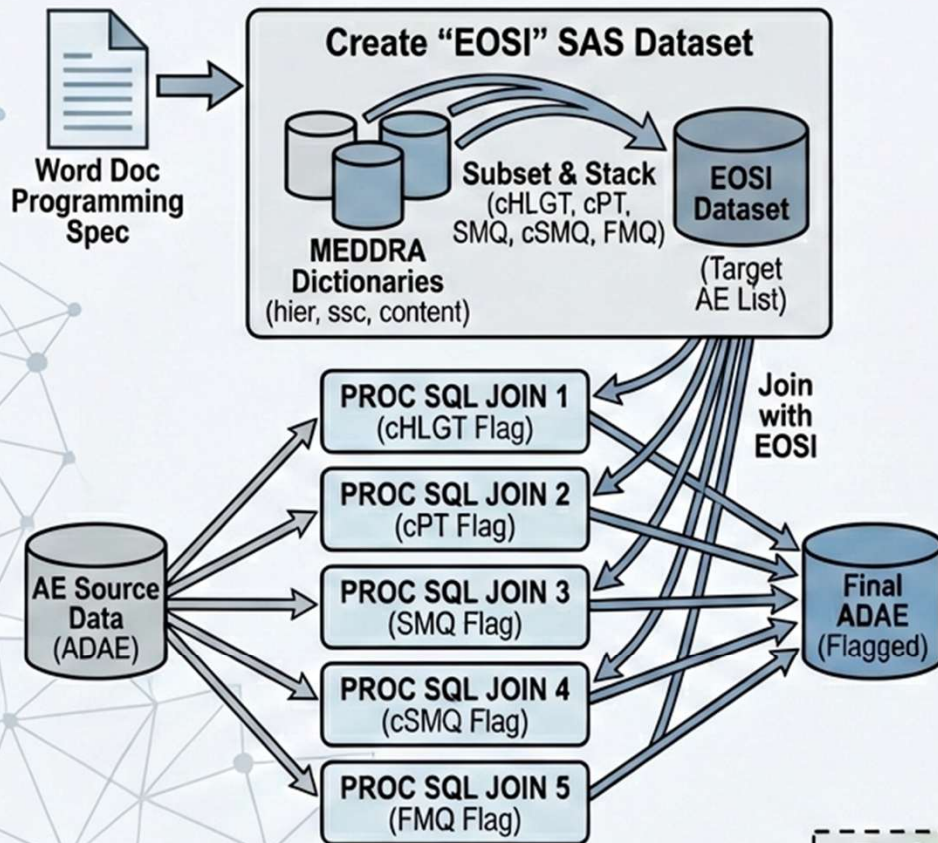
- An Excel/CSV file contains all AESI definitions, MedDRA strategies, and clinical criteria.
- Accessible to medical writers and clinicians, who can maintain definitions without SAS knowledge.
- Enables version control via Git, providing a full audit trail for regulatory submissions.
- When the spec is updated, programmers re-run the process with zero code changes.

Specification Structure Example

Variable	Description
AESI_DESC	Human-readable description ("Urinary Events")
AESI_VAR	Target variable name in ADAE (CQ01NAM)
STRATEGY	Lookup strategy (cHLGT, cPT, cSMQ, SMQ)
PROG_CODE	Primary MedDRA selection criteria
CONDITION	Additional WHERE clause filter
MEDDRA_VER	Numeric MedDRA version for audit trail (280)

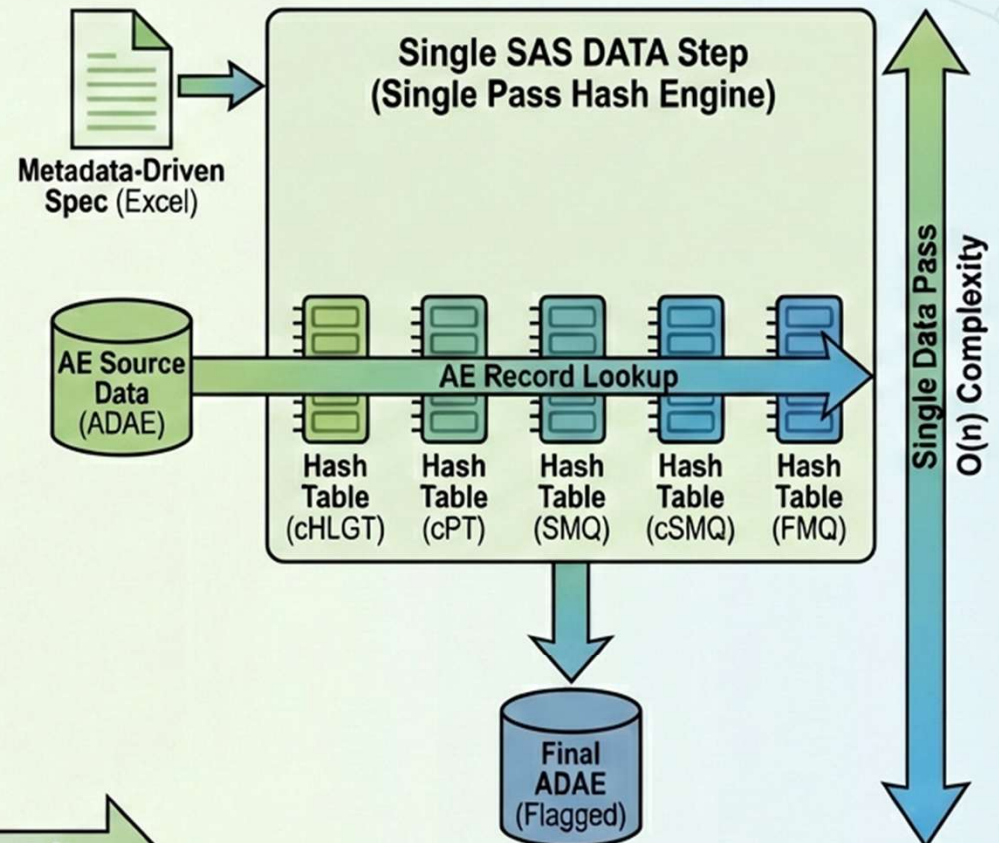
Hash vs. Merge Schematic: Traditional vs. New AESI Implementation

Traditional AESI Implementation (Multiple Merges - Fragmented)



Before → After

Single-Pass Hash Implementation (Metadata-Driven - Unified)



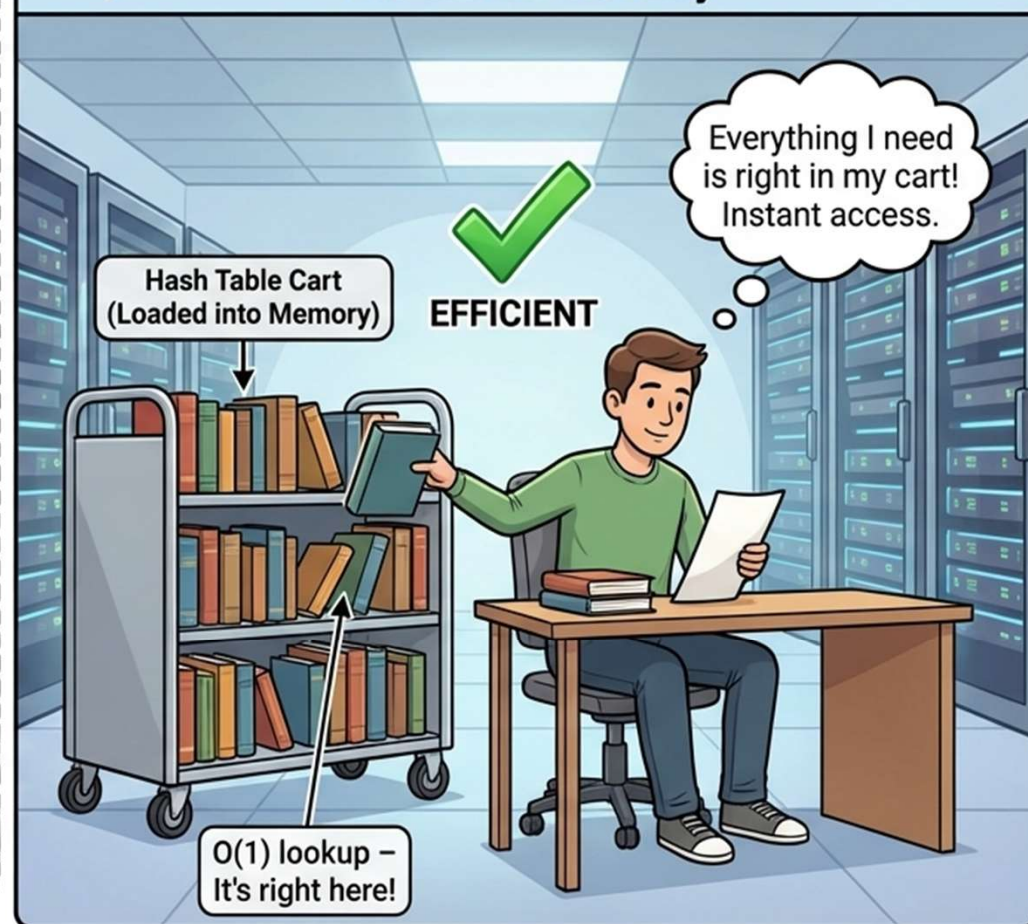
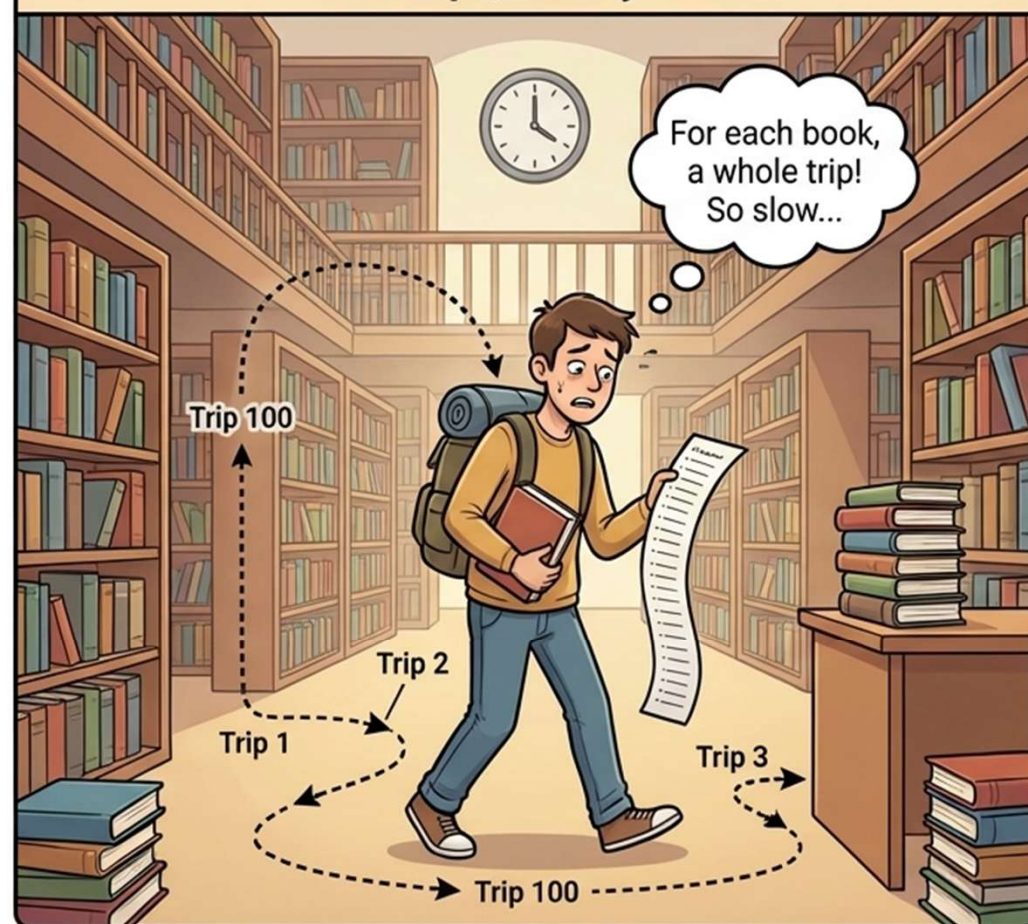
The “Library Cart” Hash Table Analogy

Commonly attributed to Paul Dorfman, SAS expert

Traditional Lookup (MERGE/SQL JOIN):
100 Trips, One by One

The Shift to Efficiency

Hash Table Lookup ($O(1)$ Lookup):
The Cart in Memory



The Result: An 85% Reduction in Per-Study Programming Effort

85%

Programming Hours by Activity (Pilot Study Data)

Activity	Traditional (hours)	Metadata-Driven (hours)
Specification Creation	3.0	0.5
AESI Dataset Programming	4.0	0.0
QC/Validation	4.0	1.0
Documentation	2.0	0.5
Total Per Study	13.0	2.0

“This frees up 11 hours of programmer time per study to be redirected to higher-value safety analysis.”

Ensuring Correctness and Capturing Value



100% Concordance

We cross-validated the output from the hash-based approach against the traditional `PROC SQL` method on our pilot studies. The resulting AESI flags matched perfectly, confirming the new method's correctness while delivering superior performance.

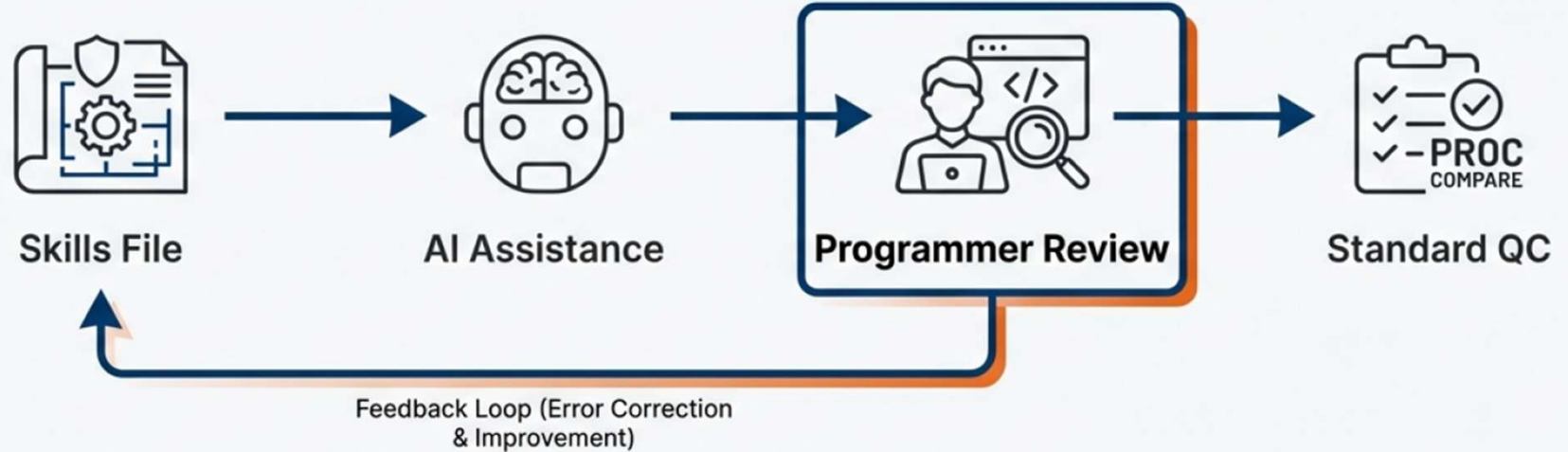


Eliminating Steps is More Powerful than Accelerating Steps

The true value was realized not just by automating coding, but by questioning the necessity of legacy processes like intermediate datasets. This architectural redesign delivered the majority of the efficiency gains.

AI Acceleration: Optional, Validated, Compliant

The architecture is implementable manually, but we accelerated it using AI tools guided by a curated “skills file.”



**“Programming remains the human responsibility;
AI augments generation, not decision-making.”**

Our Approach Aligns with EMA/FDA Principles for AI in Drug Development

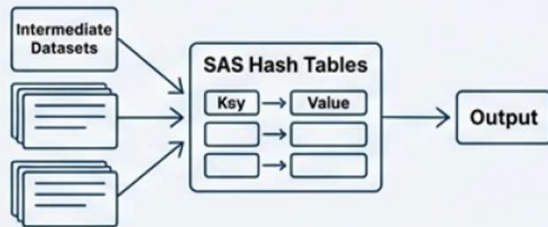
Alignment Table

EMA/FDA Principle	Our Implementation Mechanism
Human-Centric Design	All AI code undergoes independent programmer review and sign-off.
Risk-Based Approach	Tiered validation (Dev < Test < Prod) with concordance testing.
Standards Compliance	Skills file enforces PHUSE GPP and internal coding standards.
Transparent Documentation	Code is plain SAS with no hidden transformations; all outputs documented.
Life Cycle Management	Version control for spec/code; revalidation triggered by MedDRA updates.

- Based on the joint EMA/FDA ‘Guiding Principles of Good AI Practice in Drug Development’ (2026)

A New Standard for Scalable, Compliant Pharmacovigilance Programming

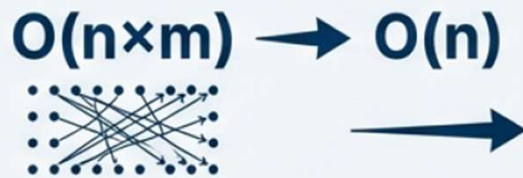
1



Architectural Simplification

SAS hash tables replace intermediate datasets, reducing logical fragmentation.

2



Performance Improvement

Algorithmic complexity reduced from $O(n \times m)$ to a single-pass $O(n)$.

Algorithmic complexity reduced and programming with > leverage automation.

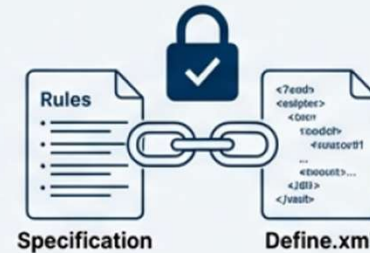
3



Dramatic Effort Reduction

An 85% decrease in per-study programming hours (13 down to 2).

4



Bulletproof Regulatory Traceability

100% specification-to-Define.xml concordance for end-to-end auditability.

Contact and Resources



Contact Information

Author: Hang Zhong
Company: Bristol Myers Squibb
Email: hang.zhong@bms.com



Related Resources

Main Paper: SM05
Supplementary Material: SM05-SUPP
Presentation materials available from
PHUSE US Connect 2026.

Your comments and questions are valued and encouraged.