



DEDICATED TO CLIENT SUCCESS
FOR A HEALTHIER WORLD
www.K3-Innovations.com

ML23: Harnessing LLM-Augmented Automation for ADaM Dataset Generation with Admiral

Authors:



Ashwin Kumar



Anil Mogha



Gourav Garg



DEDICATION

TRANSPARENCY

HONESTY

AGENDA



1

Limitations of current metadata and AI approaches

2

Solution : HYBRID AI

3

SPARC AI: A Practical Implementation

4

Architecture

5

Pilot Study results

6

Conclusion

LIMITATIONS OF CURRENT APPROACHES



Metadata Only Systems

- Rigid architectures
- High manual rework when protocols/SAPs change

General Purpose LLMs

- Lack CDISC context
- Inconsistent outputs
- Insufficient traceability for regulatory use
- Hallucinations

HYBRID AI: BRIDGING THE GAP



Why Hybrid?

- Neither rules nor LLMs alone are sufficient for regulated workflows

Hybrid AI combines

- Metadata-driven automation
- (CDISC, deterministic, traceable)
- Constrained LLM Augmentation
- (context-aware, flexible, semantic)

Outcome

- Flexibility with control
- Intelligence with auditability

METADATA-FIRST ARCHITECTURE



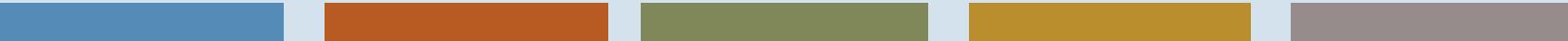
Variable-level
metadata:
source,
derivation intent,
terminology

Analysis intent
metadata for
TLFs

Metadata reused
across specs,
datasets, and
outputs

ADaM
specifications as
structured inputs

VARIABLE-GROUP DECOMPOSITION: A PRACTICAL TECHNIQUE



Common
variables

Copied-from-
source

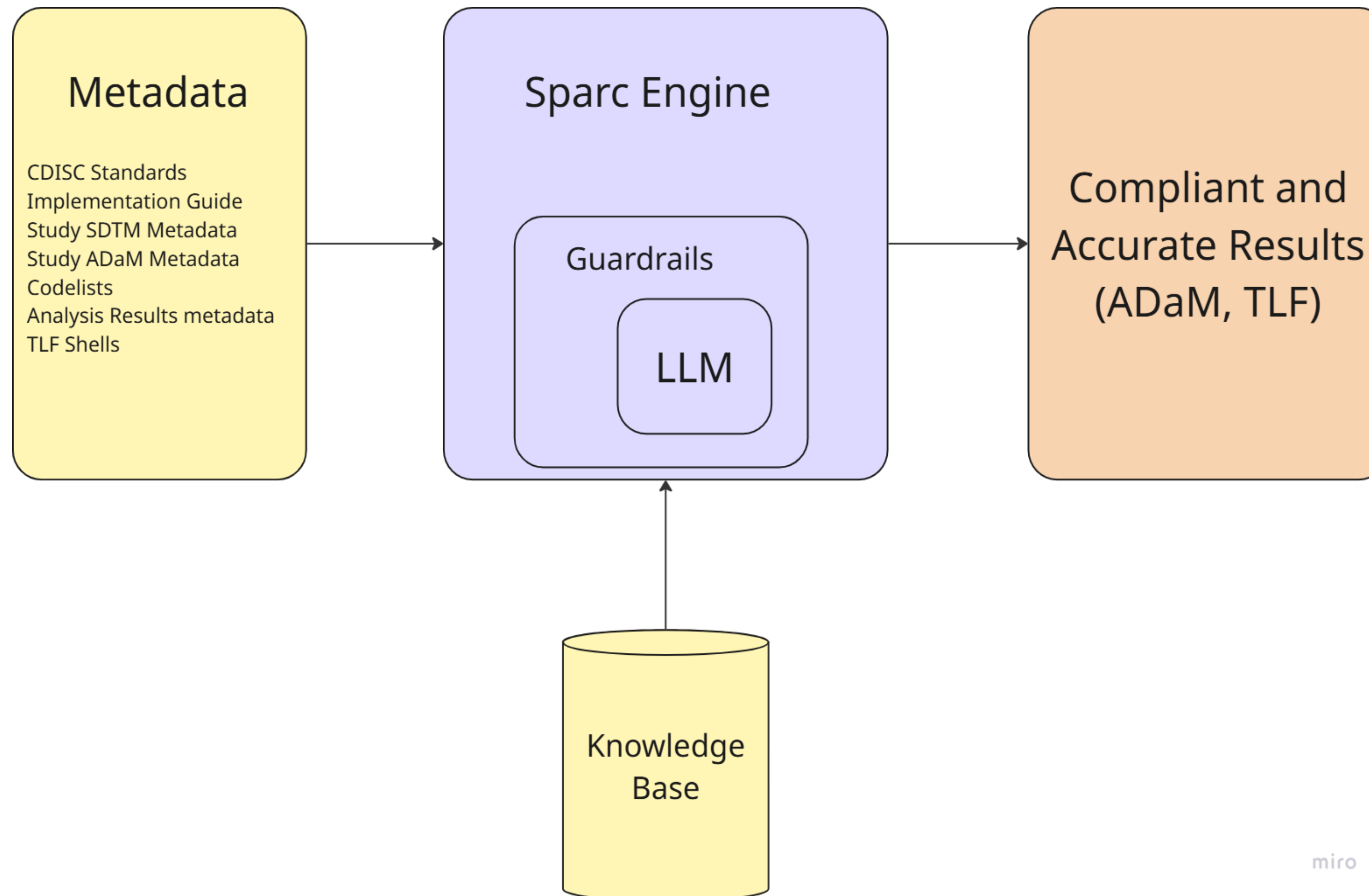
Dates and timing

Parameterized
variables

Flags/populations

Reduces dependency errors and improves validation

HOW IT WORKS



EXAMPLE SCREEN

```
1 # Generated on 2026-02-06 11:22:53 UTC
2
3 # =====
4 # Group: VS Origin Variables (Part 1 of 3)
5 # Variables: ABLFL, ADT, ADY, ATPT, AVAL, AVISIT, BASE, BASETYPE, PARAM, PARAMCD, VISIT, VISITNUM, VSSEQ
6 # =====
7 # Load required libraries and initialize data
8 library(admiral)
9 library(haven)
10 library(dplyr)
11 vs <- read_sas("SDTMDatasets/Outputs/vs.sas7bdat")
12 adsl <- read_sas("AdamDatasets/Outputs/adsl.sas7bdat")
13 tryCatch(
14   advs <- vs %>%
15     mutate(
16       PARAMCD = VSTESTCD,
17       PARAM = VSTEST,
18       ADT = as.Date(VSDTC),
19       ADY = VSDY,
20       ATPT = VSTPT,
21       AVAL = VSSTRESN,
22       BASE = ifelse(VSBLFL == "Y", VSSTRESN, NA),
23       BASETYPE = VSTPT,
24       VISITNUM = VISITNUM,
25       VISIT = VISIT,
26       VSSEQ = VSSEQ,
27       ABLFL = VSBLFL
28     ),
29   error = function(e) {
30     message("Error in processing VS Origin variables: ", e)
31   }
32 )
33
34 # =====
35 # Group: ADSL Origin Variables (Part 2 of 3)
36 # Variables: AGE, AGEGR1, AGEGR1N, RACE, RACEN, SAFFL, SEX, SITEID, STUDYID, TRTA, TRTAN, TRTEDT, TRTP, TRTPN, TRTSDT, USUBJID
37 # =====
38 tryCatch(
39   advs <- advs %>%
40     derive_vars_merged(
41       dataset_add = adsl,
42       by_vars = exprs(STUDYID, USUBJID),
43       new_vars = exprs(
44         AGE = AGE,
45         AGEGR1 = AGEGR1,
46         AGEGR1N = AGEGR1N,
47         RACE = RACE,
48         RACEN = RACEN,
49         SAFFL = SAFFL,
50         SEX = SEX,
51         SITEID = SITEID,
52         TRTA = TRT01A,
53         TRTAN = TRT01AN,
54         TRTEDT = TRTEDT,
55         TRTP = TRT01P,
56         TRTPN = TRT01PN,
57         TRTSDT = TRTSDT
58       )
59     ),
```

Libraries

Datasets

Source Variables

Merged Variables

```
54   TRTEDT = TRTEDT,
55   TRTP = TRT01P,
56   TRTPN = TRT01PN,
57   TRTSDT = TRTSDT
58 )
59 },
60 error = function(e) {
61   message("Error in processing ADSL origin variables: ", e)
62 }
63 )
64
65 # =====
66 # Group: Other variables (Part 3 of 3)
67 # Variables: ANL01FL, ATPTN, AVISITN, CHG, PARAMN, PCHG
68 # =====
69 tryCatch(
70   advs <- advs %>%
71     mutate(
72       PARAMN = case_when(
73         PARAMCD == "SYSBP" ~ 1,
74         PARAMCD == "DIABP" ~ 2,
75         PARAMCD == "PULSE" ~ 3,
76         TRUE ~ NA_real_
77       ),
78       ATPTN = case_when(
79         ATPT == "PRE-DOSE" ~ 1,
80         ATPT == "POST-DOSE" ~ 2,
81         TRUE ~ NA_real_
82       ),
83       AVISITN = case_when(
84         AVISIT == "SCREENING" ~ 0,
85         AVISIT == "BASELINE" ~ 1,
86         AVISIT == "WEEK 1" ~ 2,
87         AVISIT == "WEEK 2" ~ 3,
88         TRUE ~ NA_real_
89       ),
90       CHG = AVAL - BASE,
91       PCHG = ifelse(!is.na(BASE) & BASE != 0, 100 * (CHG / BASE), NA_real_),
92       ANL01FL = ifelse(
93         !grepl("SCREENING|UNSCHEDULED|AMBUL ECG REMOVAL", AVISIT),
94         "Y",
95         NA_character_
96       )
97     ),
98   error = function(e) {
99     message("Error in processing Other variables: ", e)
100   }
101 )
102 # Export the dataset
103 write_xpt(
104   advs,
105   "AdamDatasets/Outputs/advs.xpt",
106   version = 5
107 )
108 print("Dataset advs has been saved to: AdamDatasets/Outputs/advs.xpt")
```

Variable Groups
Section

Export format

DEPENDENCIES



Software Versions & Release Info

- tidyverse – v2.0.0 (CRAN)
- haven – v2.5.5 (CRAN)
- admiral – v1.4.1 (CRAN)
- GPT-4o – March 26, 2025

HYBRID CODE GENERATION



- Metadata defines the contract
- LLMs operate *within* that contract

Enriched metadata includes

- Library & execution metadata
 - admiral, dplyr, haven (fixed dependencies)
- Dataset-type metadata
 - ADSL, BDS, OCCDS (drives structure & patterns)
- Source dataset metadata
 - VS, DM, EX (constrains available inputs)
- Merge & join metadata
 - ADSL joins on STUDYID / USUBJID
- Variable grouping metadata
 - SDTM-origin, ADSL-merged, derived variables
 - Enables bounded, reviewable code blocks
- Output & export metadata
 - XPT format, versioning, naming conventions
- ADaM specification metadata
 - Predecessor, computation, label, datatype

TRACEABILITY AND AUDITABILITY

- Metadata-driven lineage
- Semi Deterministic regeneration of datasets across runs
- Audit-ready outputs

CDISC_PILOT_STU...

Study

SDTM

ADAM

TLFs

MDR

SAP

Reference Codes

Workspaces

Dataset	Variable	Label	Type	Key Sequence	Origin
ADSL	DSRAEFL	Discontinued due to AE?	Char	0	Derived
ADSL	DTHFL	Subject Died?	Char	0	Derived
ADSL	BMIBL	Baseline BMI (kg/m^2)	Char	0	Derived
ADSL	BMIBLGR1	Pooled Baseline BMI Group 1	Char	0	Derived
ADSL	HEIGHTBL	Baseline Height (cm)	Char	0	Derived
ADSL	WEIGHTBL	Baseline Weight (kg)	Char	0	Derived
ADSL	EDUCLVL	Years of Education	Char	0	Derived
ADSL	DISONSDT	Date of Onset of Disease	Char	0	Derived
ADSL	DURDIS	Duration of Disease (Months)	Char	0	Derived
ADSL		Pooled Disease			

© 2025 K3-Innovations, Inc. All rights reserved.

Workspace
Dev

Admin

A

Summary

Target Variable
Dataset: ADSL
Variable: BMIBL
ID: var_33

Lineage Graph

SDTM
VSTESTCD
VS

SDTM
VISITNUM
VS

ADaM
HEIGHTBL
ADSL

ADaM
WEIGHTBL
ADSL

ADaM
BMIBL
ADSL
Target

+

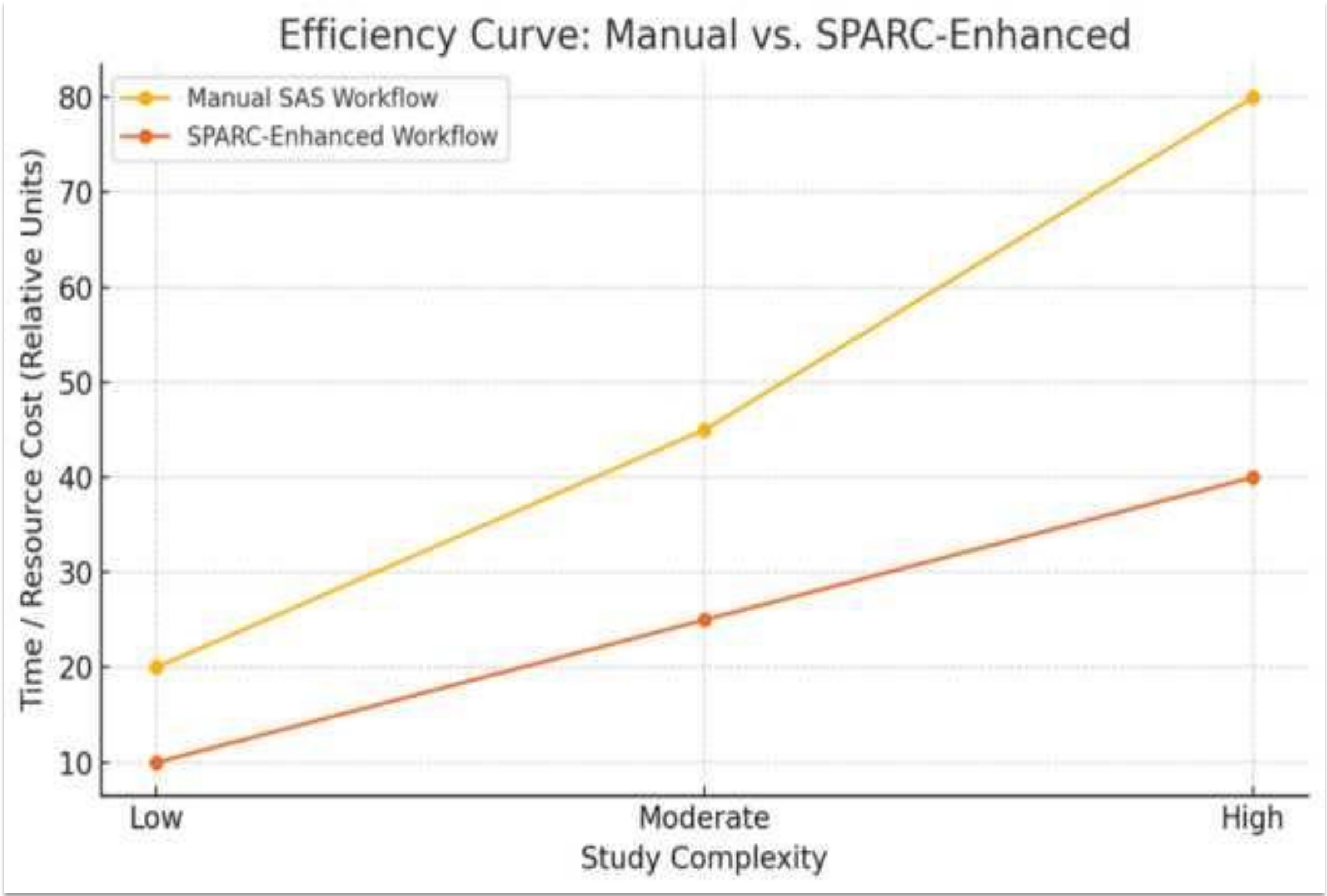
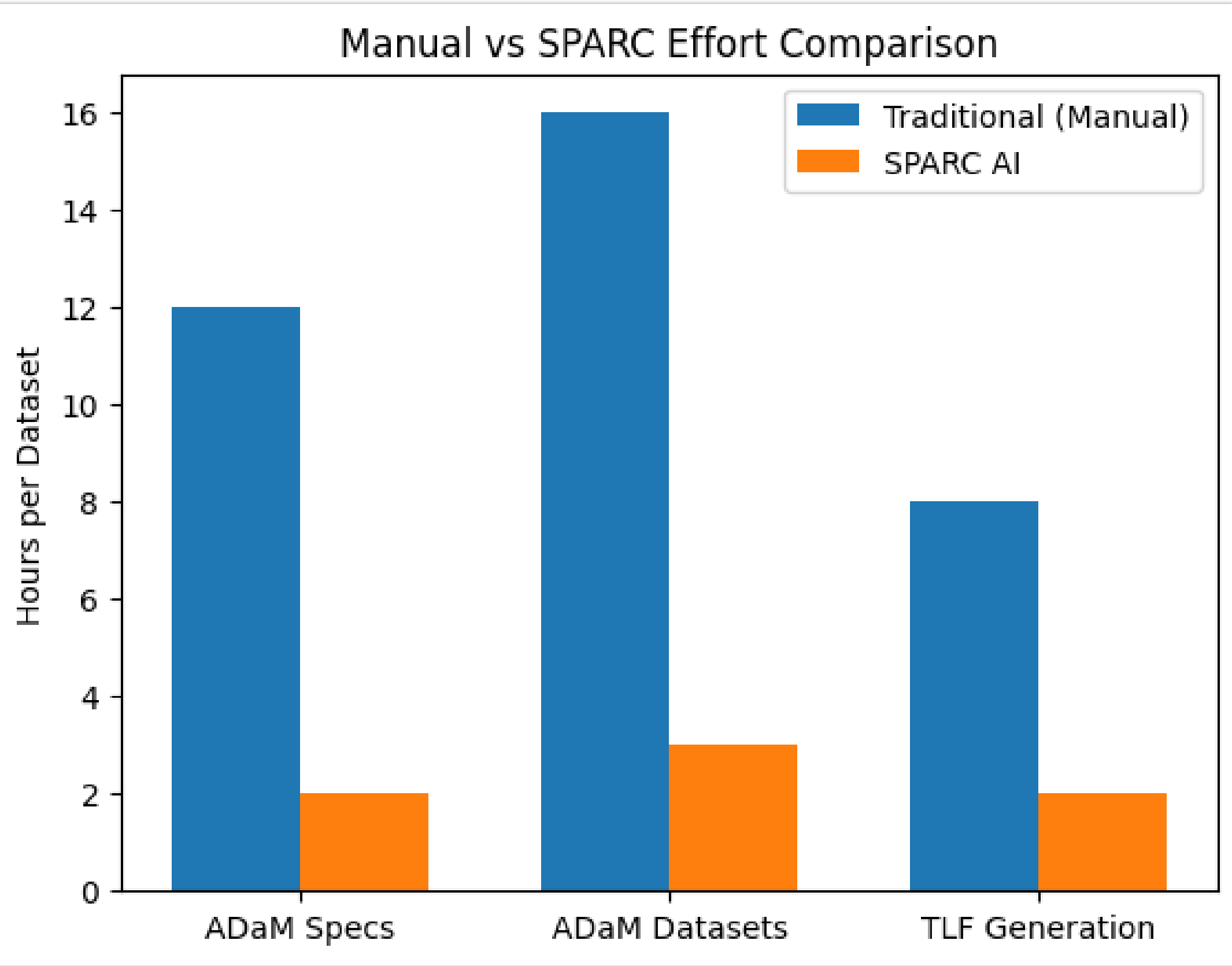
-

↺

↻

🗑

KPIs: MEASURING SUCCESS



Conclusion



- ✓ AI must be **metadata-native** to be regulator-ready
- ✓ Hybrid architectures balance flexibility with control
- ✓ Deterministic automation and LLMs are complementary
- ✓ Platforms like SPARC demonstrate a practical path forward

References

<https://k3-innovations.com/>
<https://sparcgen.com/>
<https://sparcgen.com/demo>



THANK YOU!!

CONTACT US

To know more on our solutions, visit : info@k3-innovations.com

Please contact +1-732-645-8366 101, or <https://www.k3-innovations.com/>

101 Morgan In, Suite 206, Plainsboro, New Jersey 08536