



BeiGene

ACIRA: Empowering Clinical Data Analysis with AI-Driven ADaM Coding

Tingting Zeng (BeiGene)

May 2025

DISCLAIMER

The content, ideas, and conclusions presented in this presentation represent the personal views of the author(s) and do not necessarily reflect the views or positions of BeiGene.

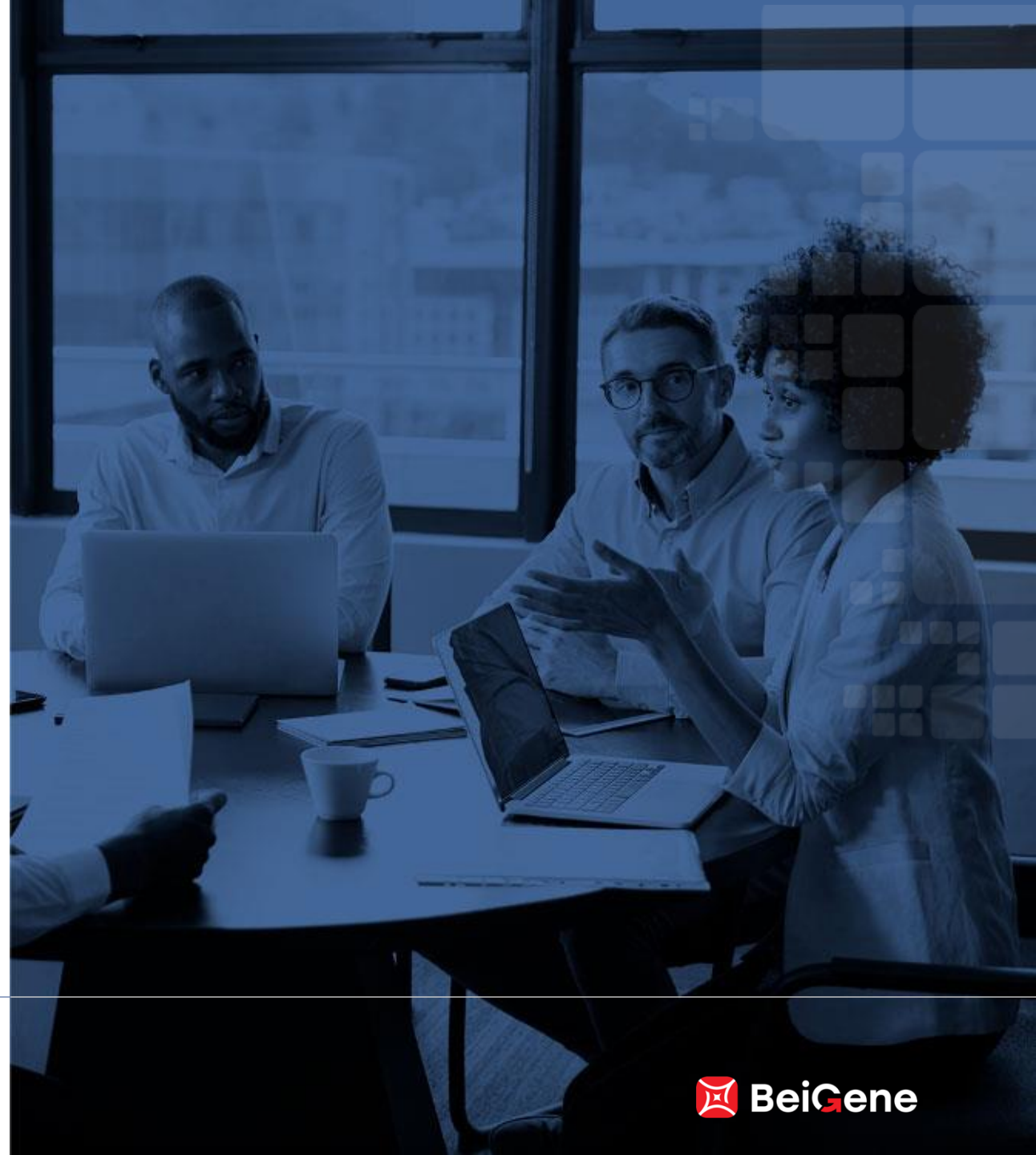


TABLE OF CONTENTS

- > **Key Challenges for ADaM Coding**
- > **Product Positioning and Strategic Goals**
- > **ACIRA Flowchart and Key Features**
- > **Product Showcase**

Key Challenges for ADaM Coding



Complexity & Time-consuming

- Integrate multiple information
- Complex derivation logics
- Manual checks are inefficient and tedious



Frequent Changes & Lack of Version Control

- Continuous code updates due to spec/ other source changes
- Heavy communication and risk for misalignment
- Difficult to trace historical versions for both specs & codes



Limited Reusability

- Study-specific demands
- Diverse coding styles
- Lack of clear linkage between specs and codes



Product Positioning and Strategic Goals



**ADaM Code Intelligence
for Rapid Analysis**

*Our Orchestral Ecosystem:
Harmonizing ACIRA, SpecMaster* and Beyond.*

01

**AI and Digital-Driven
Efficiency**

- Establishes a centralized, highly reusable digital library.
- Develops intuitive platform to foster human-AI collaboration for code generation, maintenance and quality assurance.

02

**End-to-End Solution to
Expedite Data Analysis**

- Connects SpecMaster* to cover the entire ADaM lifecycle.
- Improves efficiency, quality and ensures regulatory compliance.

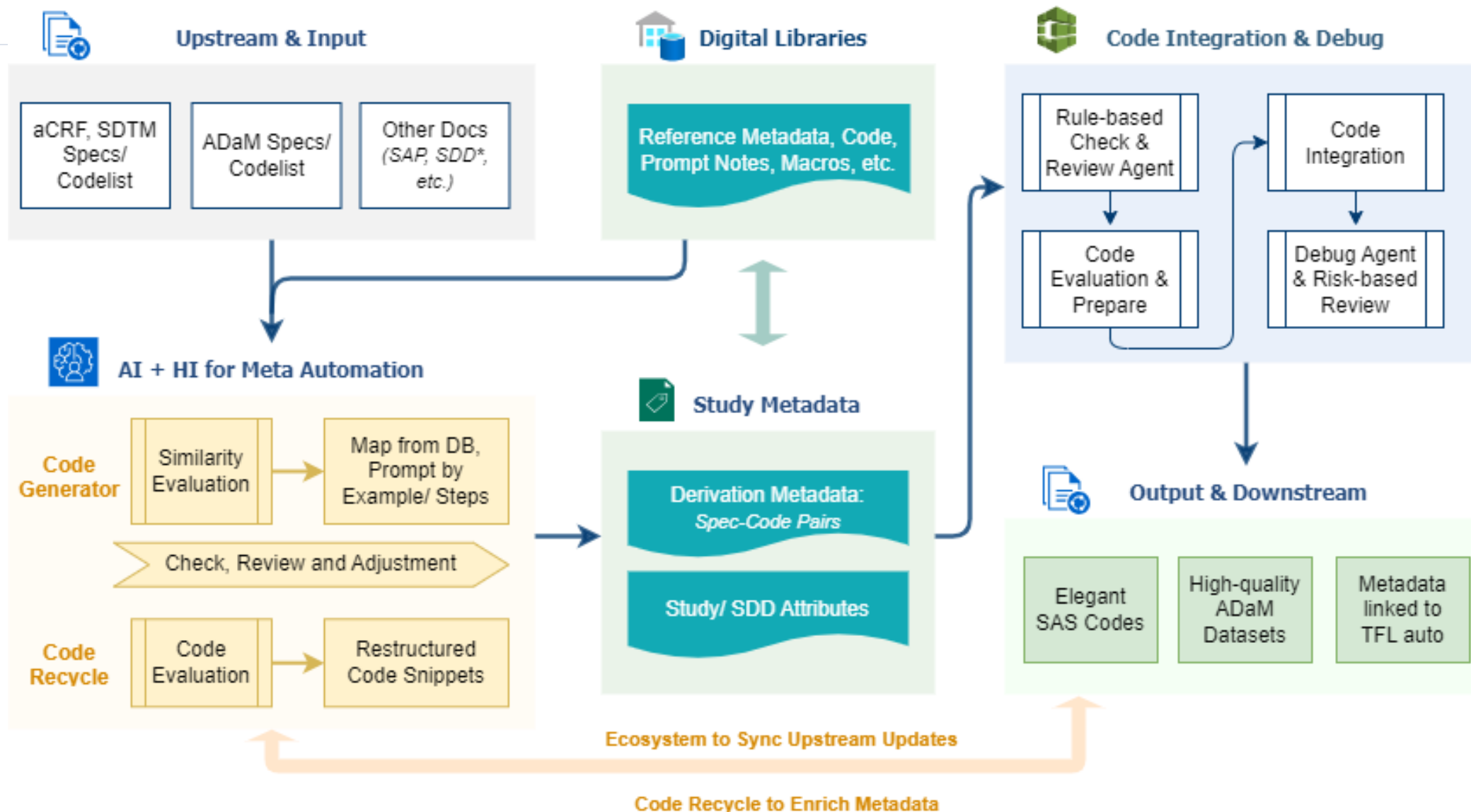
03

**Interconnected Data
Ecosystem**

- Integrates workflow across SDTM, ADaM, TLFs, and submission.
- Establishes data ecosystem through version control and change alert.

* *SpecMaster is a comprehensive platform that intelligently manages ADaM specifications across standard, compound, and study levels, offering robust features for collaboration and communication.*

Flowchart for ACIRA



* SDD: company/ compound standard definitions and derivation rules

ACIRA Key Features

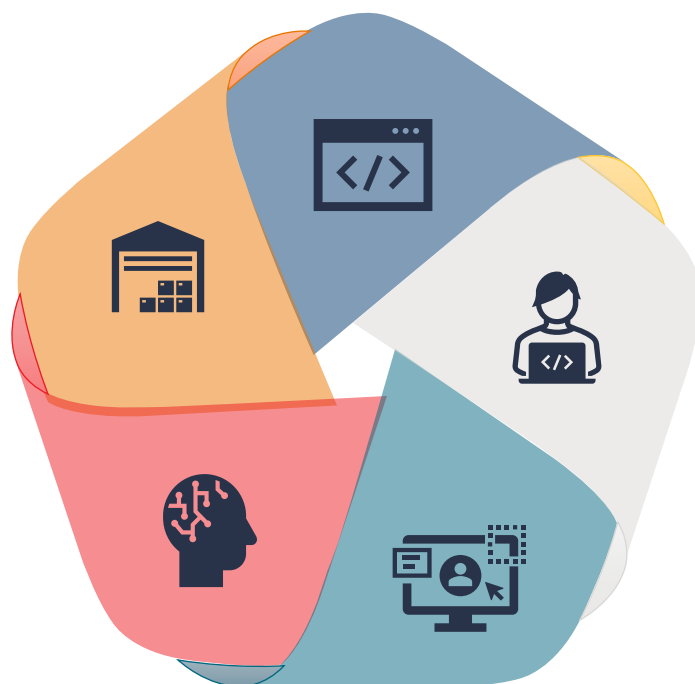


Digital Libraries for IntelliCode

Well-designed & Growing Repository
Version Control & Linkage across Modules
Knowledge Base to Enhance AI Adaptability

AI Application and Evaluation

AI-Powered Code Generation
Review & Debug Agents
Code Recycle for Lib. Enrichment



Well-structured and Elegant Code

Maintainable and Debuggable
Submission-Ready

Efficient Developer Console

Facilitate Human-AI Collaboration
Real-Time Metadata Monitoring
Ecosystem for up/downstream collaboration

Intuitive UI for Risk-Based Review

Risk Assessment and Mitigation
Cross-Check and Adjustment
Change Tracking and Comparison

Digital Libraries for Intelligent Code

Derivation Metadata Library

Method-Code pairs for variables & parameters
Build-in macros for ADaM derivation
Code designs to handle diverse scenarios

Prompt Note Library

RICCE framework* for AI prompts
Dynamic notes per meta/ code designs or
feedbacks to improve accuracy



Study/ SDD Attributes

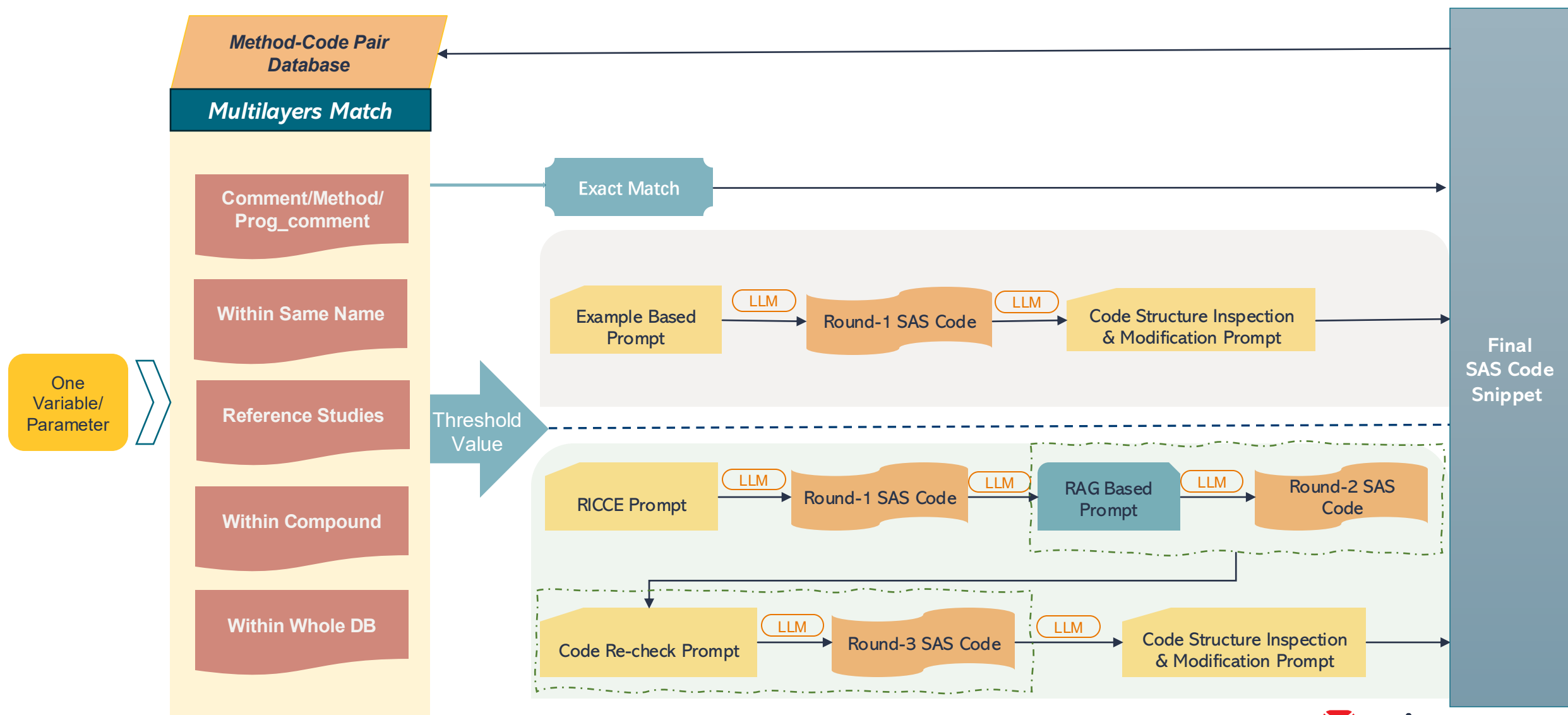
Code snippets per different **study designs**
and **data collection scenarios**
SDD attributes (e.g. for ADTTE PFS, OS, etc.)

Cross-module Meta Integration

**Integrated metadata across SDTM, ADaM
and TFL** with solid linkage and version control
Ecosystem with **real-time change alerts**

* *RICCE framework: a mnemonic device for the five key elements (role, instruction, context, constraints, examples)
to consider when crafting prompts*

AI Workflow for Code Generation



Overview of AI Agents*



- **Automated Code Analysis** to evaluate code snippets.
- **Implement Customizable Rules** to ensure flexibility and adaptability.
- **Instant and Actionable Feedbacks** to facilitate quick iterations and improvements.



- **Comprehensive Debugging** to identify and resolve issues in integrated codes.
- **Adaptive Error Learning** to enhance error detection capability.
- **Dynamic Workflow** to optimize problem-solving efficiency.

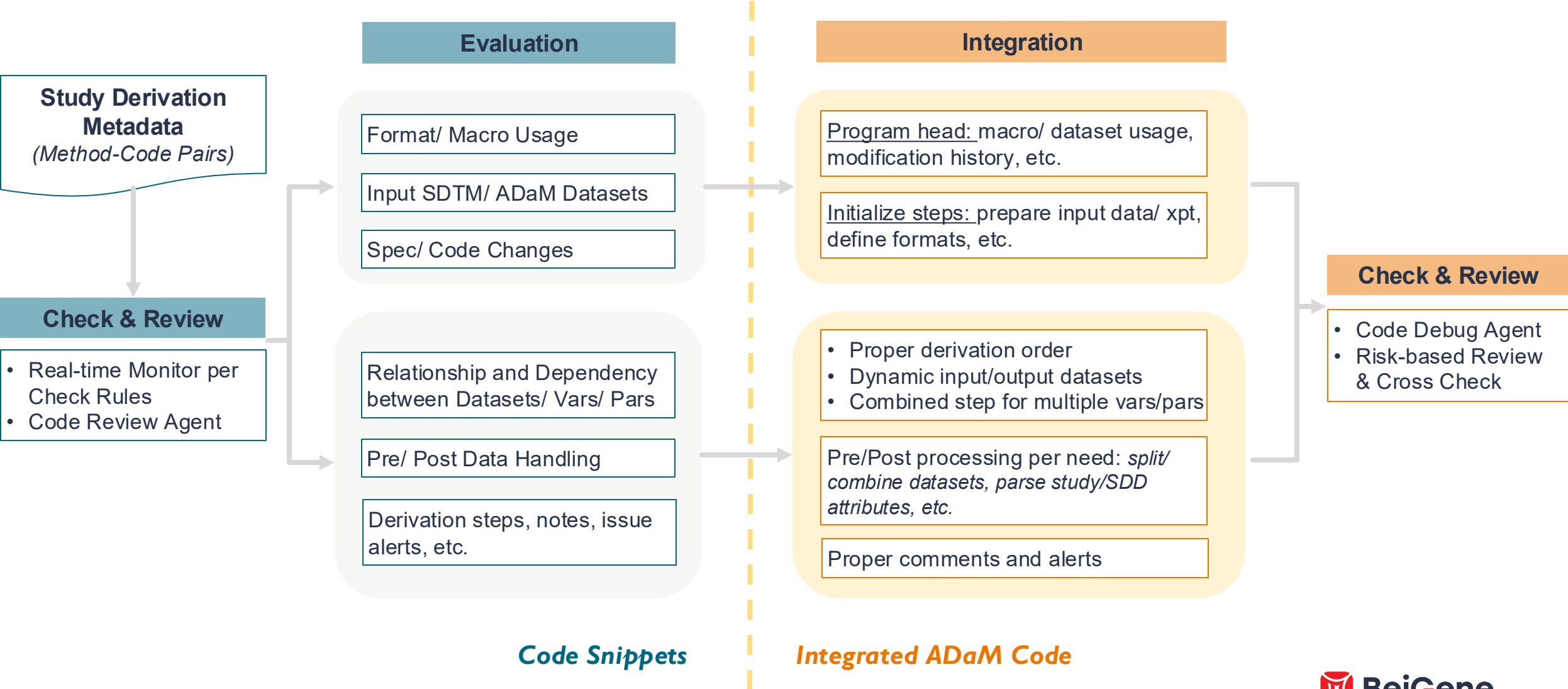


- **Change Monitoring** to track and analyze manual updates in final codes.
- **Real-time System Updates** to ensure codebase up-to-date and standardized.
- **Prompt Generation Optimization** to enhance AI accuracy.

* Agents under plan or development.

Code Evaluation and Integration

Aim to generate well-structured, unified-style codes which are easy to maintain and submission-ready.



Code Evaluation and Integration – *One Simple Example*

```
set lb_all; ①
length ABLFL $1 AVISIT $200; ②
format ADT date9. ADTM datetime20.;

ABLFL=LBBFL;

*** Derive ADT, ATM, ADTM, ADY;
%m_adams_dtc2dt_v2(vardtc=LBDC, vardt=ADT, vartm=ATM, vardtm=ADTM, ady_startdt1=TRTSDT,

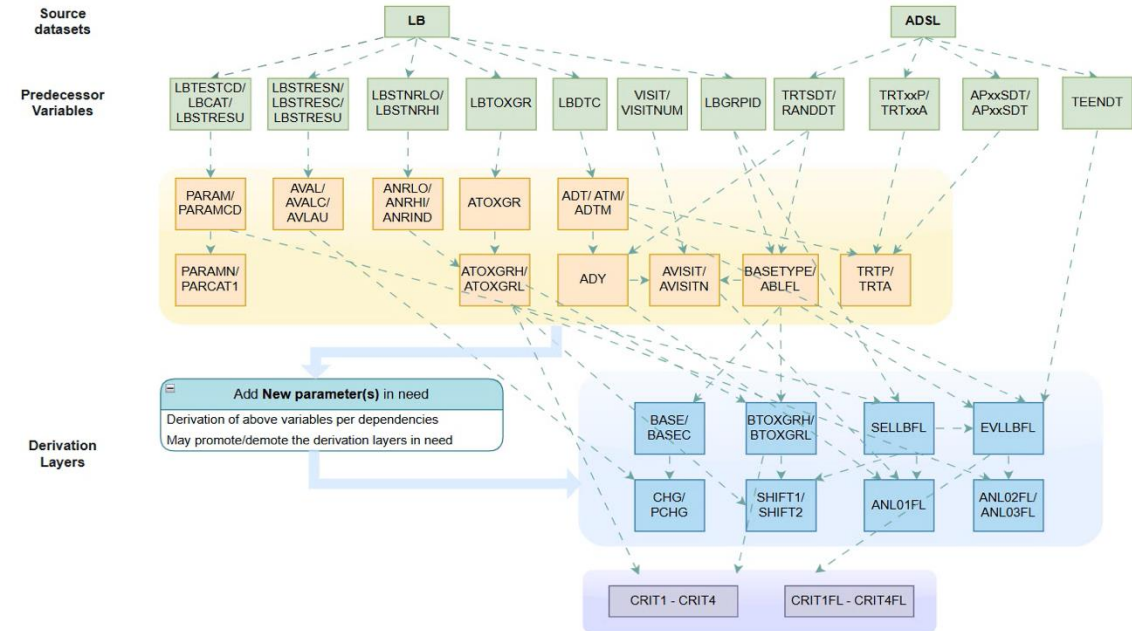
if ABLFL='Y' then AVISIT='BASELINE';
else if ADY>=1 and find(visit, 'UNSCHEDULED')=0 then AVISIT=VISIT;

if not missing(AVISIT) then AVISITN=VISITNUM;
if ABLFL='Y' then AVISITN=0; ③
```

Code snippets in metadata

How does ACIRA evaluate and integrate above code snippets?

- ① Combine several snippets in one data step and define input dataset (LB merged ADSL).
- ② Add variable attributes per need.
- ③ Decide proper order based on variables dependency.



How does ACIRA create a complete ADaM code?

- Evaluates code snippets for all vars/pars (relationship/ dependency/ etc.)
- Designs overall code structure and dynamic derivation steps
- Performs necessary pre/post processing
- Creates final integrated code and ensure its accuracy and readability

ONE click does it all

Intuitive Interfaces for Code Generation and Review

Aim to foster seamless human-AI cooperation, enhancing efficiency and ensuring quality and oversight.



Developer Console

- One-click to get **AI generated code**
- Intuitive UI to **compare specs/ AI codes** vs. recommended **examples**
- Up/Downstream **change tracking and sync alert**
- Monitor **meta status and quality**, and one-click to **integrate codes**
- **Continually improve quality** via feedbacks to AI and regular review on examples

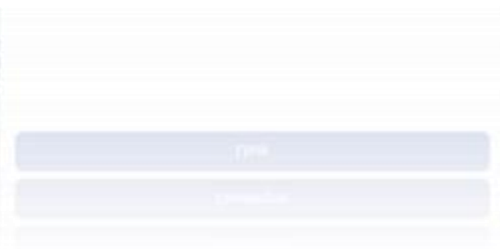
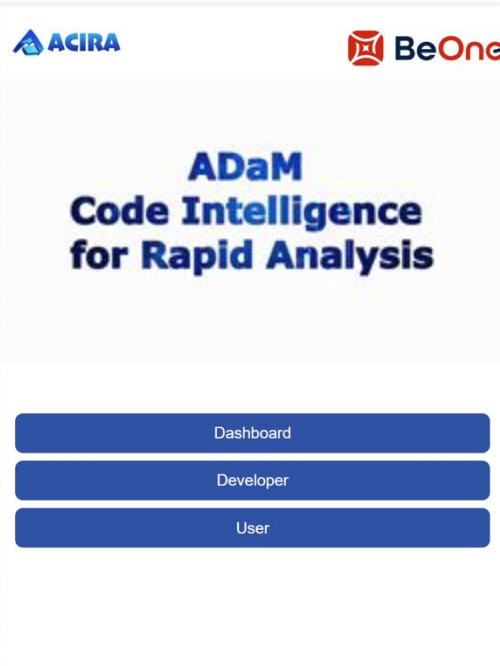


Risk-based Review

- **Risk assessment** to focus review efforts
- **Cross review** codes with spec methods, with **quick navigation** to meta workspace
- Leverage **CodeGenius*** features such as AI-driven code formatting, review, debug, etc.
- **Centralized dashboard** for tracking project progress, efficiency, changes and quality indicators

** CodeGenius is an AI-powered SAS copilot that boosts programming efficiency and code quality by integrating internal resource.*

Product Showcase



Summary of Key Features

System Demonstration

Achievements and Goals



THANK YOU!

For any questions, please reach out to:
tingting.zeng@beigene.com