
Embedding Governed AI into Statistical Programming Workflows

From SAS Coding Assistance to Compliance-Oriented Execution
In ISPP CoreFlow

SPEAKER

Yaohui Zhu
BeOne Medicines

EVENT

PHUSE Beijing SDE 2026

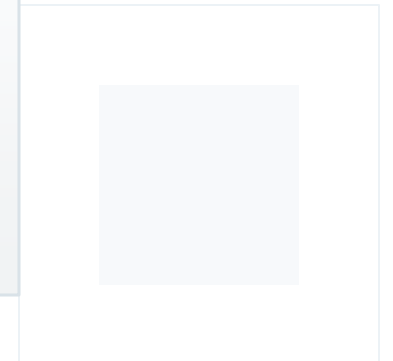


IMPORTANT NOTICE

Author Viewpoint 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 BeOne Medicines.



Session Agenda

From product landscape to governed execution in five blocks.

01-02



Opening and boundary

Cover, disclaimer, and the framing of governed AI inside regulated statistical programming.

03-07



CoreFlow Introduction

CoreFlow as the workflow shell, distributed AI surfaces, and why ISPP Copilot is needed.

08-10



CodeGenius workbench

The coding surface, native sidebar, bridge controls, and the coding-side demo sequence.

11-16



Governed execution

In-app capability, routing logic, safe path, human approval, and the CoreFlow demo arc.

17-18



Synthesis and close

Why both AI surfaces matter together, the final takeaways, and the closing thank-you.

TFL

Expand Filters

Reset Filters

Select All (All Pages)

Select Page

Run

Check

Details

Export

Dir Mode

Trk Editor



☐	#	Index 1	Index 2	Index 3	Index 4	Alert	Issue #	Dtime	Vtime	Log	Vlog	Vlst	PDF	RTF	Action
☐	1	1	1	1	1	4		2026-04-19 19:55:30		log			pdf	rtf	RunCheckDetails
☐	2	1	1	1	1	2		2026-04-11 10:54:16		log			pdf	rtf	RunCheckDetails
☐	3	1	1	1	1	2		2026-04-11 10:54:15		log			pdf	rtf	RunCheckDetails
☐	4	1	1	1	1	2		2026-04-11 10:54:17		log			pdf	rtf	RunCheckDetails
☐	5	1	1	1	1										RunCheckDetails
☐	6	1	1	1	1										RunCheckDetails
☐	7	1	1	1	1	0		2026-04-11 10:54:45		log			pdf	rtf	RunCheckDetails
☐	8	1	1	1	1	0		2026-04-11 10:54:25		log			pdf	rtf	RunCheckDetails
☐	9	1	1	1	1	0		2026-04-11 10:54:29		log			pdf	rtf	RunCheckDetails
☐	10	1	1	1	1	0		2026-04-11 10:54:31		log			pdf	rtf	RunCheckDetails



DESIGN IDEA ISPP Copilot is distributed across different work surfaces. For coding in CodeGenius and for workflow execution in CoreFlow pages.

CoreFlow pages

Home, Dir Mode, Tracker Mode,
Tracker Editor, Help, and AI Chat

The product context already exists
before AI is invoked.

CodeGenius

AI workbench for coding tasks
Native left sidebar ISPP copilot

Editor-to-chat and chat-to-editor
interaction loop

ISPP Copilot

Bottom-right in-app widget
Route-aware workflow assistant

Can bridge to governed execution
through the agent backend

AI is a distributed capability inside CoreFlow, not a single isolated page.

KEY INSIGHT

As workflows span pages and actions, users need a governed intent bridge, not just one more static help page.

What ISPP Copilot Does

Understands natural-language requests in context

Links page context to task interpretation

Routes requests toward the right execution path

Introduces approval before consequential actions



What Users Gain

Faster task completion across modules

Lower learning curve for new team members

Intent-driven workflows, not button-driven

Confidence in governed, auditable workflows

ISPP Copilot is the governed intent bridge across CoreFlow modules and actions.

SERVICE MODEL

The backend is a multi-agent service layer that converts product context into intent routing, approval, execution, and safe fallback behavior.

**Coding Agent**

Edit, review and debug SAS programs

BatchAgent

Batch execution and log-oriented operations

DataAgent

Query and analysis tools with bounded execution

GeneralAgent

Documentation, Q and A, and safer fallback support

This is a scoped orchestration layer inside the product, not a generic chatbot wrapper.

A browser-based, AI-powered SAS IDE that brings the VS Code experience to statistical programmers



CodeGenius

AI-Powered SAS Editor

CodeGenius is a **production-ready**, browser-native IDE built on **Monaco Editor** — the same engine powering VS Code. It provides **LSP support**, integrated AI assistance, and seamless file management for SAS statistical programming workflows.

Browser Native

Monaco Editor

LSP Powered

AI Copilot

SAS Focused

Stack: TypeScript · Vue 3 · React 18 · Vite · Tailwind CSS · Socket.IO



Monaco Editor & File Workspace

Full-featured Monaco editor with SAS syntax highlighting, IntelliSense, and an analysis-scoped virtual file system backed by REST APIs.



VS Code-like Layout & Navigation

Identical workbench chrome — Activity Bar, Explorer, tabs, status bar, and command palette — delivered entirely in the browser.



Native Left Sidebar AI Copilot

Chainlit-powered chat panel embedded in the sidebar. Context-aware code generation, review, and documentation — no context switching.



SAS-Focused Development Workflow

Dedicated SAS Language Server (pygls), batch program execution with job tracking, log checker, git SCM, and output panel for PDFs / datasets.



Interactive SAS Execution

Real-time interactive session panel with ODS viewer, DataFrame preview, scratch pad, and work file manager — run and inspect results instantly.

KEY POINT

The workbench identity matters because AI is only useful if it stays attached to coding context, file context, and human review.

Workspace model

Monaco editor

Workbench layout

File and task context

AI Copilot

Native left sidebar view

Conversation inside the same shell

Human review before insertion

Practical value

Less context switching

Visible suggestions and edits

Better foundation for governance

This is an AI-native coding surface, not an editor plus a detached chat widget.

Send code

Selection goes directly
from editor to chat

Get selection

Backend can request current
editor context

Insert code

Accepted output goes back
into the editor

Approval

High-risk actions
stay explicit

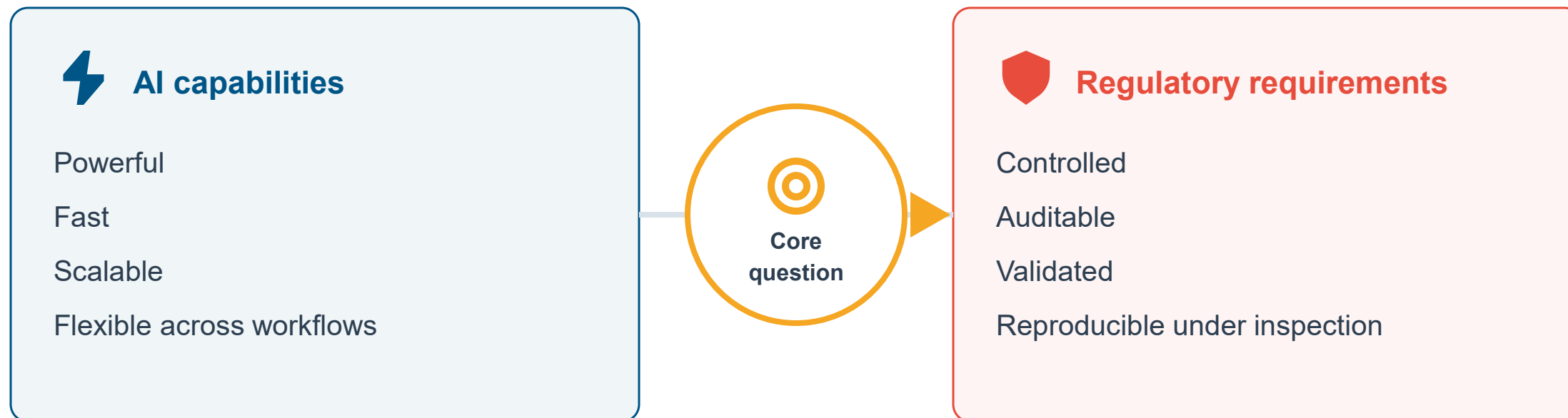
Design implication

CodeGenius is governed because suggestions remain visible, reviewable, and connected to concrete editing operations.

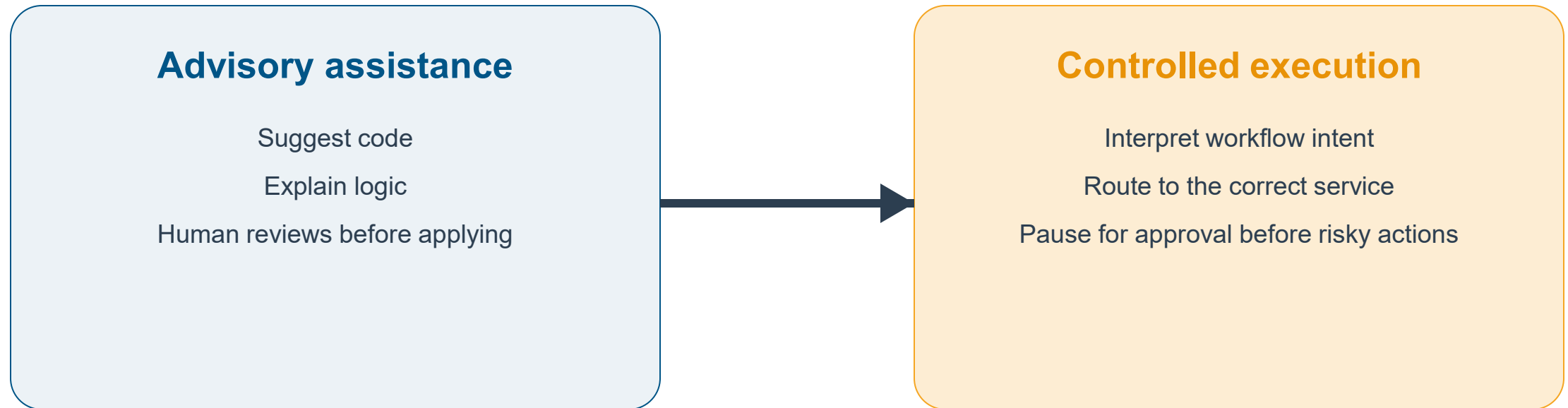
The product differentiator is the interaction bridge between code, chat, and approval.

KEY INSIGHT

A governance framework is the mechanism that lets capability survive contact with compliance requirements.



How do we keep AI helpful without sacrificing human authority, auditability, or reproducibility?



KEY INSIGHT

Governance is not a blocker. It is the operating system that turns AI capability into a trustworthy workflow.



Transparency

- Structured logs
- Traceable operations
- Visible AI decisions
- Confidence scoring

Every AI action becomes reviewable.



Human Authority

- Human-in-the-loop checkpoints
- Approval before high-risk execution
- Closed-loop control
- Explicit override capability

AI proposes; humans remain accountable.



Policy Routing

- Intent recognition
- Risk-based path selection
- Safe fallbacks for uncertainty

Control adapts to request risk.

Governance is an enabler: without it, AI remains a fragile experiment.

KEY INSIGHT

The system should automate preparation and evidence capture, but the decision to proceed still belongs to a person.

**Why this matters**

Clear ownership for consequential actions
Visible decision record for auditors

**Approval path**

Explicit confirmation
Resume execution from a known state

**Rejection path**

Prevent unintended execution
Trigger a safe cancel event

Human-in-the-loop is not friction. It is how high-risk automation becomes safe enough to use.

15 Each agent owns a clear scope with controls proportional to risk

KEY INSIGHT

Single responsibility makes governance practical: risk becomes manageable when each agent's boundary is explicit.

BatchAgent/CodingAgent

Run SAS programs
Analyze execution logs
Coordinate batch operations

Risk: high

Control: human approval required

DataAgent

Query data sources
Run statistical or Python analysis
Return sandboxed computation results

Risk: medium

Control: sandbox and bounded tools

GeneralAgent

Document help and code review
Explain architecture and respond to Q&A
Provide direct guidance without execution

Risk: low

Control: direct response with clear boundaries

Clear boundaries are what let risk controls scale without turning every interaction into friction.

KEY INSIGHT Capability without governance increases risk; governance converts capability into something the organization can actually rely on.

$$\text{Trust} = \text{AI capability} \times \text{Governance}$$

Without governance

Powerful but risky
Fast but uncontrolled
Capable but hard to defend in audit
Useful as a demo, fragile as an operating model

With governance

Powerful and trustworthy
Fast and controlled
Capable and explainable under review
Ready for adoption in standardized workflows

KEY INSIGHT

The strongest adoption message is operational: governance makes AI safe enough for real work, not just interesting enough for pilots.

1. Governance enables adoption

Without governance, AI remains a high-risk experiment.

With governance, it becomes a tool teams can trust in production workflows.

2. Human authority is preserved

AI can accelerate work, but people must still control consequential decisions.

Approval points keep accountability where regulators expect it.

3. Transparency builds trust

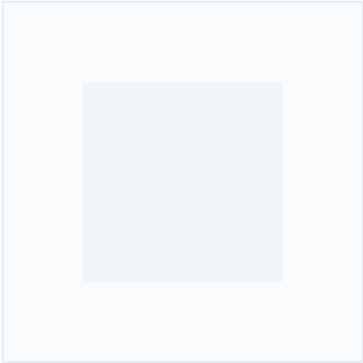
Operations must be logged, attributable, and reviewable.

No regulated environment can rely on black-box execution.

4. Architecture determines scalability

Good boundaries support audit, testing, and controlled change.

Bad architecture makes governance costly and fragile.



THANK YOU

Questions and Discussion



Embedding governed AI into real statistical programming practice

Yaohui Zhu

BeOne Medicines | PHUSE 2026

yaohui.zhu@beonemed.com

