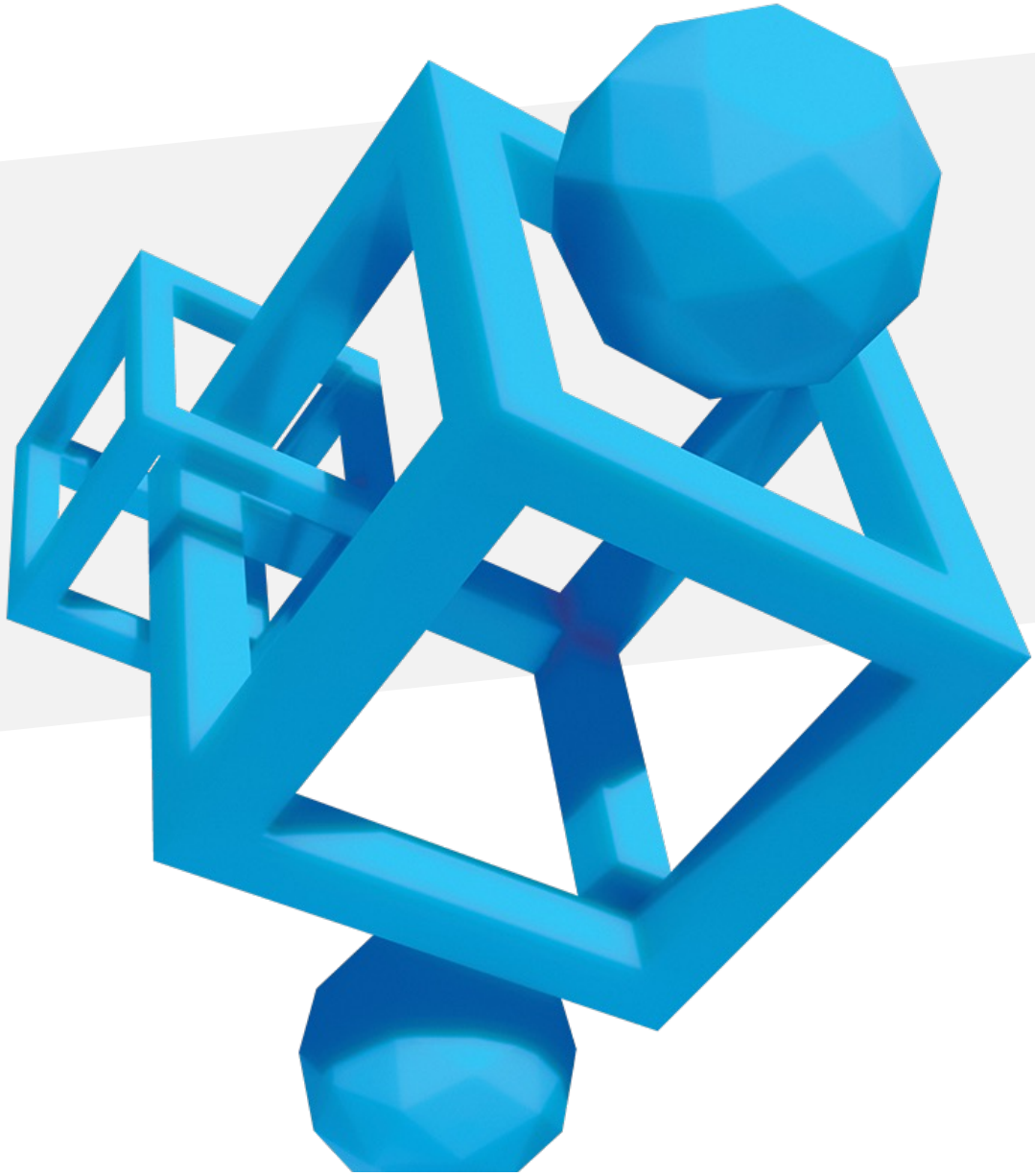
instantiations

The VAST AI Assistant

ESUG 2026



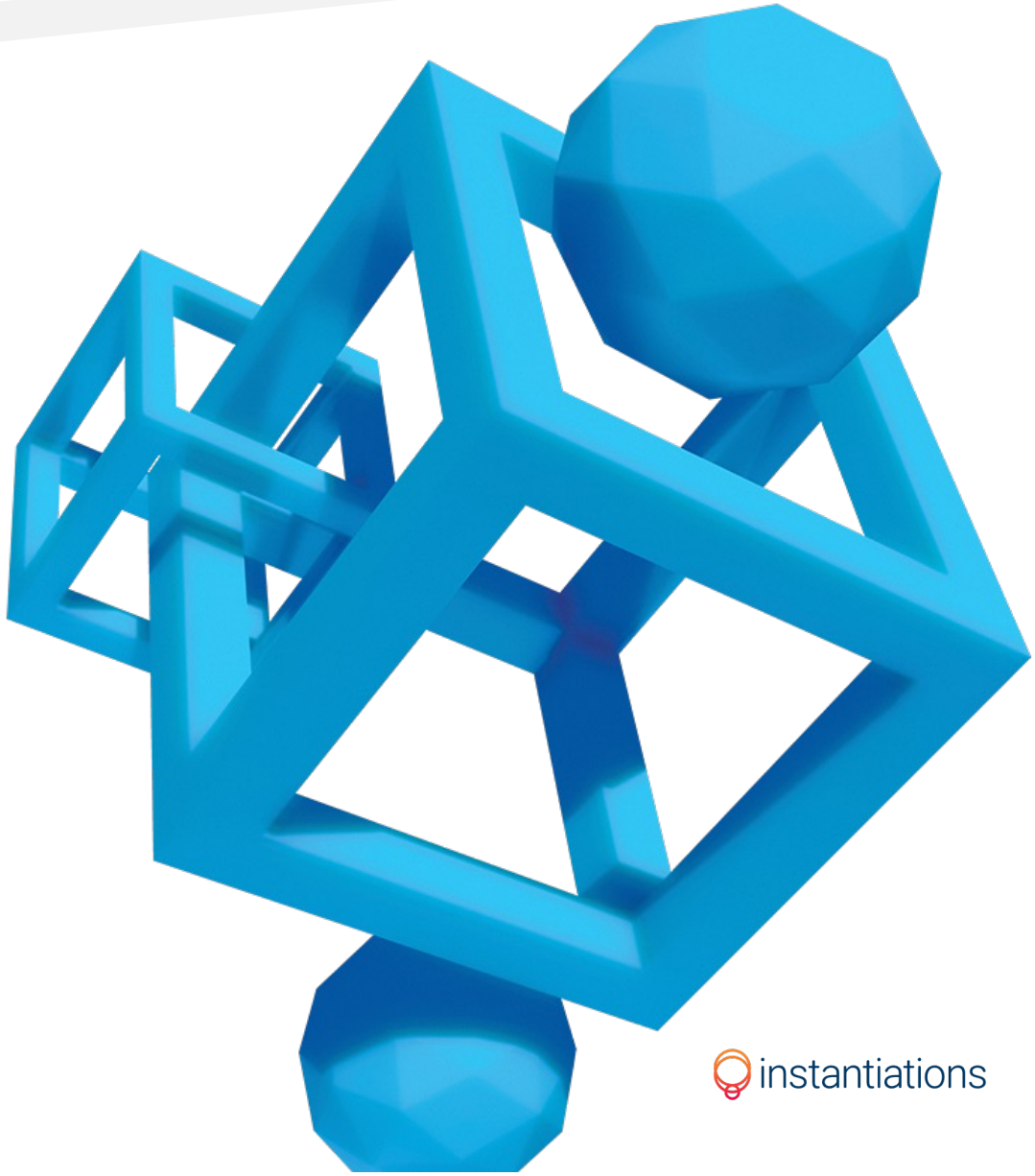
Johan Brichau and Kris Gybels

Senior Software Engineers

jbrichau@instantiations.com | kgybels@instantiations.com

Agenda

- VAST AI Assistant goals
- How to get started
- The Assistant
- IDE integration
  - Code changes browser
  - Debugger
  - Envy semantic diff
- Applications



instantiations

VAST AI Assistant Goals

instantiations

AI Development at Instantiations

- Embedding AI into real systems, workflows, and augmenting how people actually work
- Balancing innovation with stability, sustainability, and trust

The Approach: Responsible Evolution™

instantiations

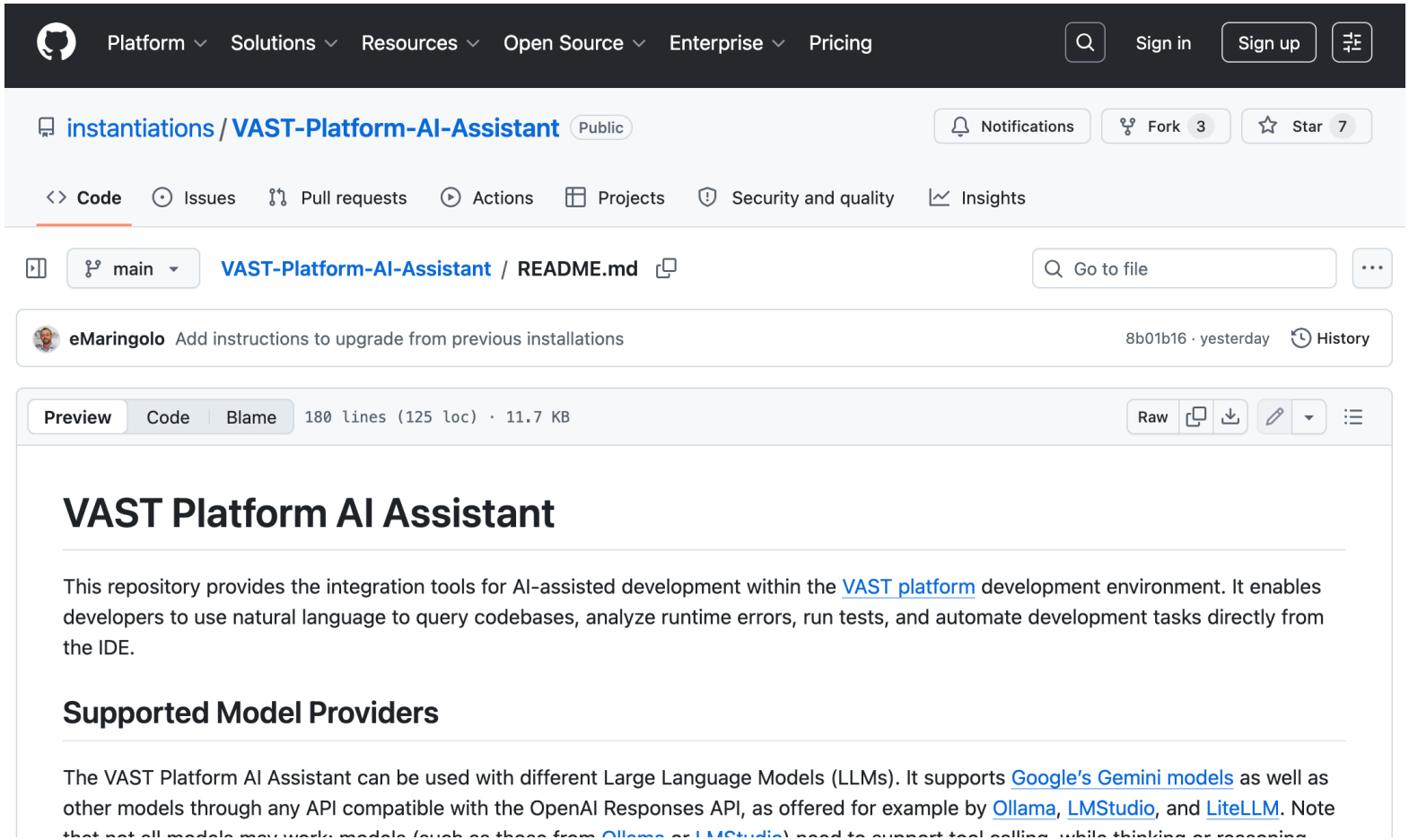
VAST AI Assistant goals

- Improve developer productivity and daily workflows
- Seamless and deep integration of natural language and AI-assisted development in the VAST IDE
  - Autonomous exploration of the system architecture to gather context before attempting to explain code or suggest modifications
  - Fully integrated with the live environment and live coding workflow
- Focus on ”Human in the loop” AI-assisted software development
- Allow agentic workflows

How to get started

instantiations

Open Source

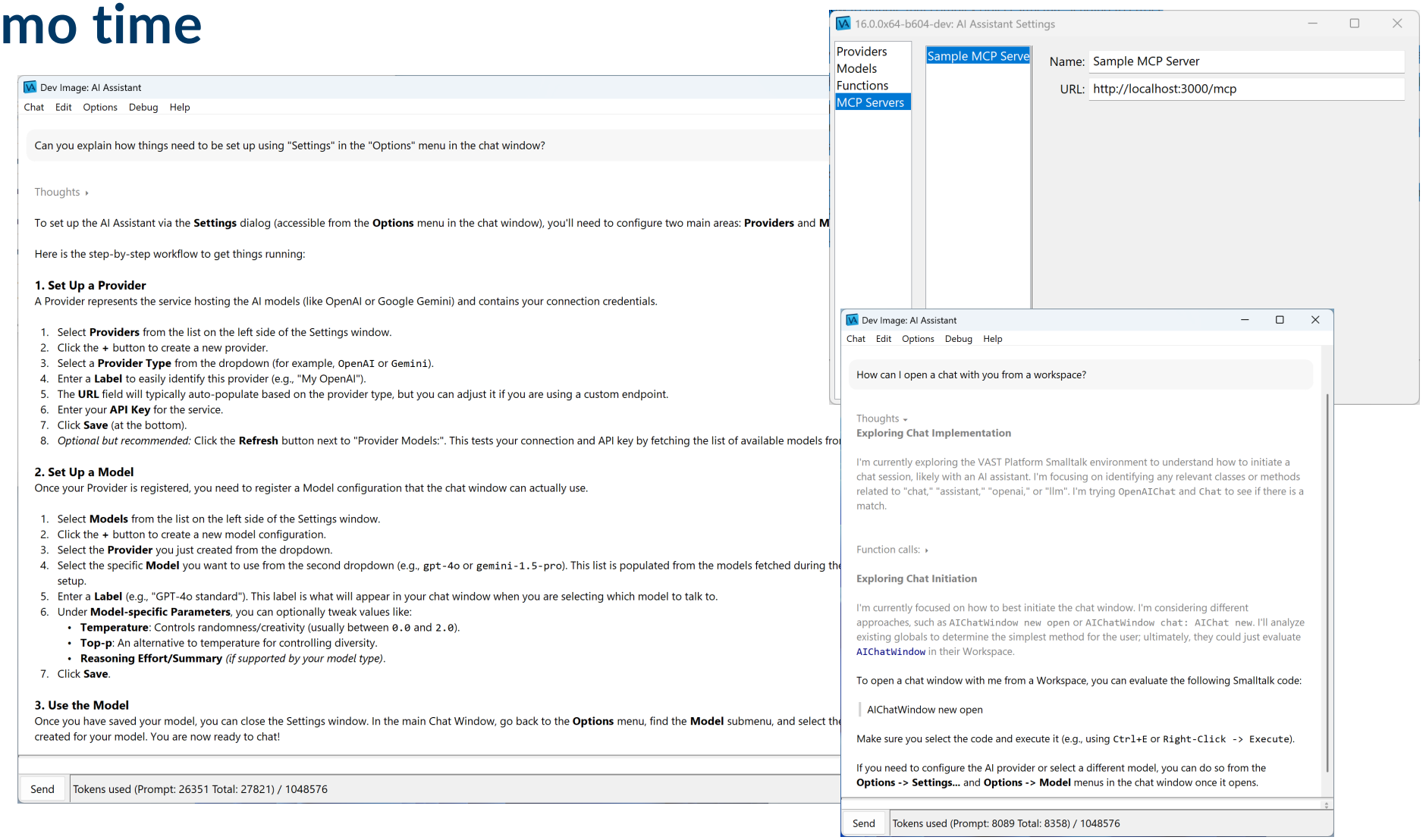


instantiations

The Assistant in practice

instantiations

Demo time

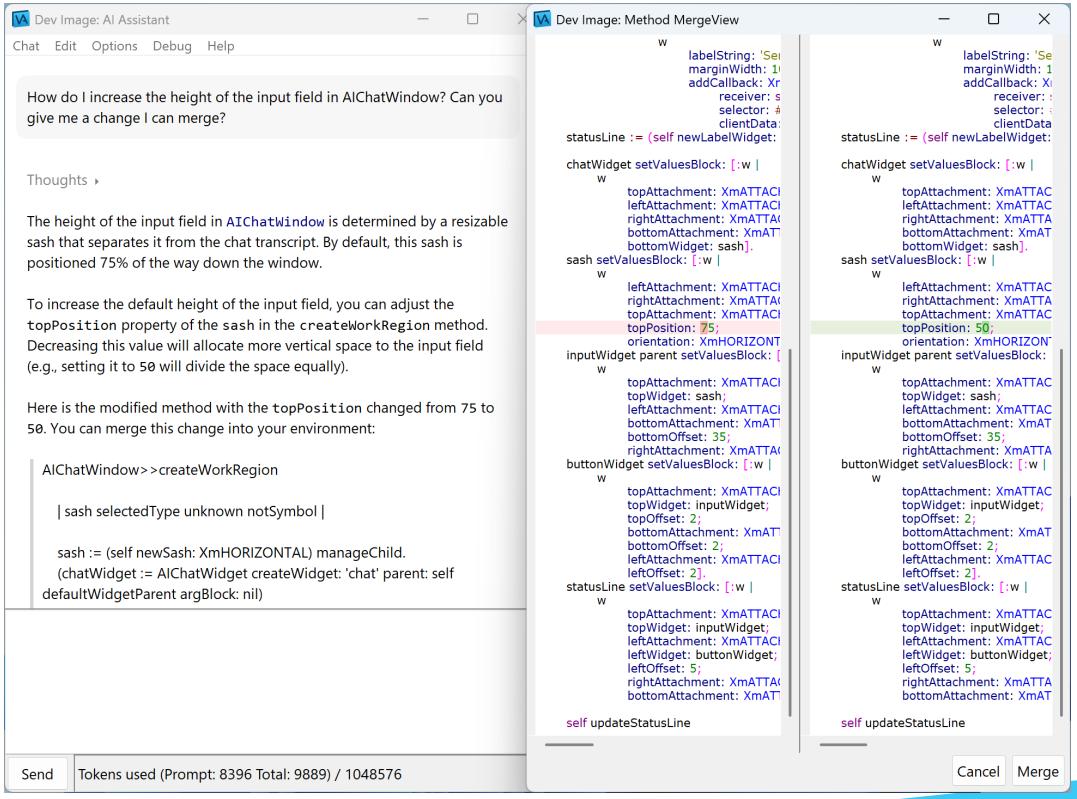

16.0.0x64-b604-dev: AI Assistant Settings
Providers
Models
Functions
MCP Servers
Sample MCP Serve
Name: Sample MCP Server
URL: http://localhost:3000/mcp
Dev Image: AI Assistant
Chat Edit Options Debug Help
Can you explain how things need to be set up using "Settings" in the "Options" menu in the chat window?
Thoughts
To set up the AI Assistant via the Settings dialog (accessible from the Options menu in the chat window), you'll need to configure two main areas: Providers and M
Here is the step-by-step workflow to get things running:
1. Set Up a Provider
A Provider represents the service hosting the AI models (like OpenAI or Google Gemini) and contains your connection credentials.
1. Select Providers from the list on the left side of the Settings window.
2. Click the + button to create a new provider.
3. Select a Provider Type from the dropdown (for example, OpenAI or Gemini).
4. Enter a Label to easily identify this provider (e.g., "My OpenAI").
5. The URL field will typically auto-populate based on the provider type, but you can adjust it if you are using a custom endpoint.
6. Enter your API Key for the service.
7. Click Save (at the bottom).
8. Optional but recommended: Click the Refresh button next to "Provider Models:". This tests your connection and API key by fetching the list of available models fro
2. Set Up a Model
Once your Provider is registered, you need to register a Model configuration that the chat window can actually use.
1. Select Models from the list on the left side of the Settings window.
2. Click the + button to create a new model configuration.
3. Select the Provider you just created from the dropdown.
4. Select the specific Model you want to use from the second dropdown (e.g., gpt-4o or gemini-1.5-pro). This list is populated from the models fetched during the setup.
5. Enter a Label (e.g., "GPT-4o standard"). This label is what will appear in your chat window when you are selecting which model to talk to.
6. Under Model-specific Parameters, you can optionally tweak values like:
- Temperature: Controls randomness/creativity (usually between 0.0 and 2.0).
- Top-p: An alternative to temperature for controlling diversity.
- Reasoning Effort/Summary (if supported by your model type).
7. Click Save.
3. Use the Model
Once you have saved your model, you can close the Settings window. In the main Chat Window, go back to the Options menu, find the Model submenu, and select the created for your model. You are now ready to chat!
Send
Tokens used (Prompt: 26351 Total: 27821) / 1048576
Dev Image: AI Assistant
Chat Edit Options Debug Help
How can I open a chat with you from a workspace?
Thoughts
Exploring Chat Implementation
I'm currently exploring the VAST Platform Smalltalk environment to understand how to initiate a chat session, likely with an AI assistant. I'm focusing on identifying any relevant classes or methods related to "chat," "assistant," "openai," or "llm". I'm trying OpenAIChat and Chat to see if there is a match.
Function calls:
Exploring Chat Initiation
I'm currently focused on how to best initiate the chat window. I'm considering different approaches, such as AIChatWindow new open or AIChatWindow chat: AIChat new. I'll analyze existing globals to determine the simplest method for the user; ultimately, they could just evaluate AIChatWindow in their Workspace.
To open a chat window with me from a Workspace, you can evaluate the following Smalltalk code:
AIChatWindow new open
Make sure you select the code and execute it (e.g., using Ctrl+E or Right-Click -> Execute).
If you need to configure the AI provider or select a different model, you can do so from the Options -> Settings... and Options -> Model menus in the chat window once it opens.
Send
Tokens used (Prompt: 8089 Total: 8358) / 1048576

IDE Integration

instantiations

AI Code Generation – first approach

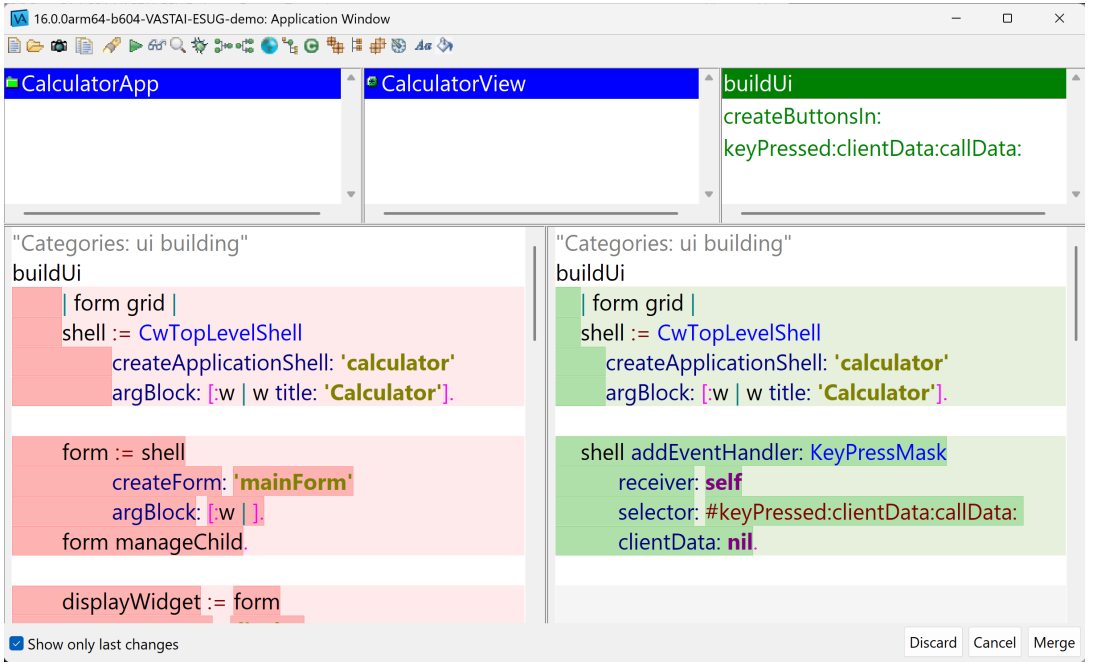
- Show **code snippets** in conversation
- Open diff view to apply code change
- Disadvantages:
  - Often produced invalid code (i.e. LLM did not validate the code first)
  - The syntax for method definitions was often not followed by the LLM
  - Not all changes are possible (e.g. remove and rename)



instantiations

AI Code Generation – new approach

- Offer **functions/tools for code modification**
- Open a merge browser on the suggested changes
- Advantages:
  - Code is always validated by the functions
  - No formatting rules required, AI will more likely follow function descriptions than the system instructions
  - Dedicated Merge Browser visualization
  - Allows agentic coding

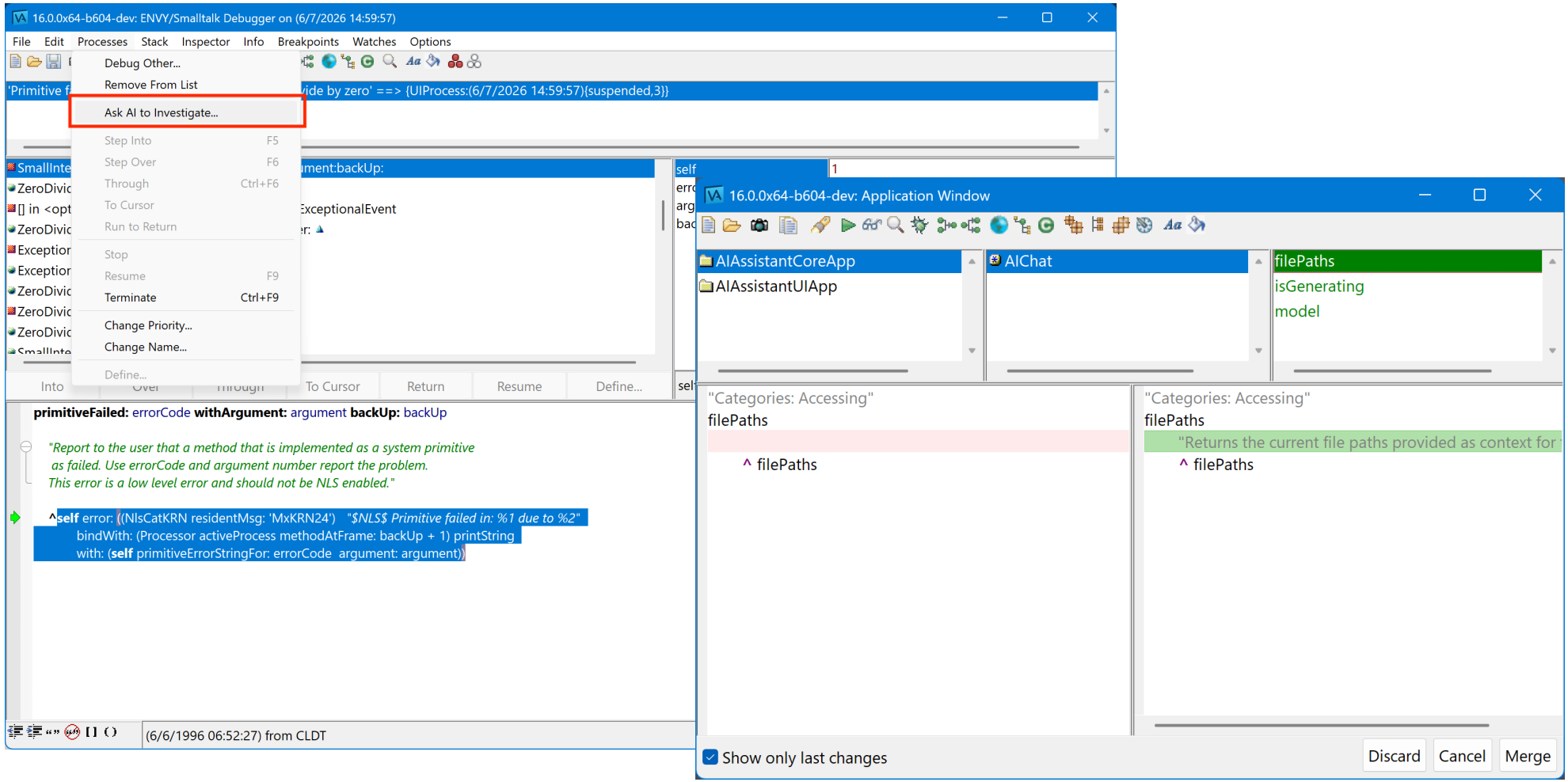


instantiations

AI-assisted debugging

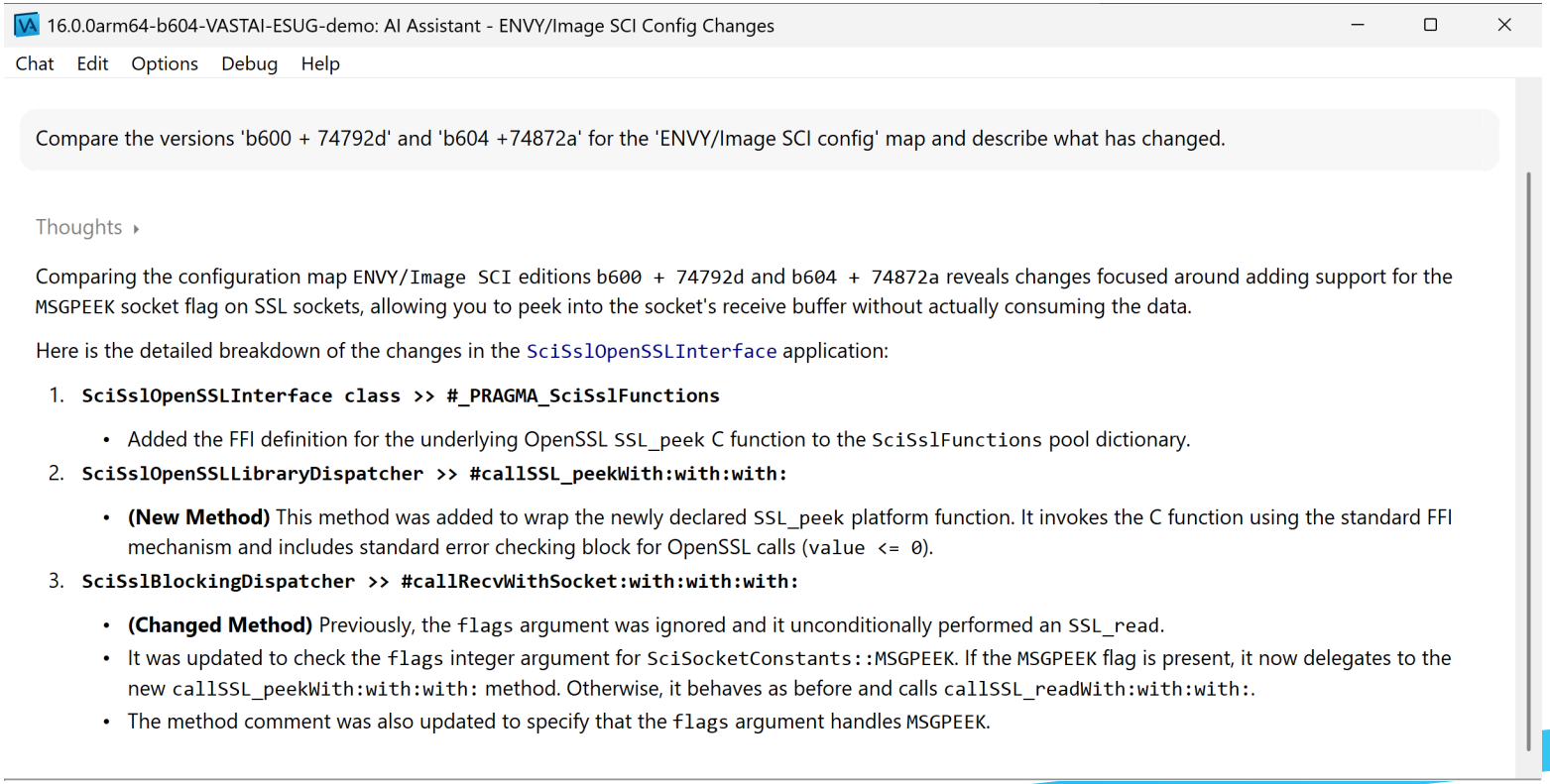
- Smalltalk is a live coding environment
  - Development in the debugger
- Trigger the AI from the debugger
  - New chat: investigate a stand-alone error
  - Select existing chat: report error in its development context
- Run and debug test cases
- More ‘live’ features in the future:
  - Inspect objects
  - Navigate the object memory

Demo time 2


16.0.0x64-b604-dev: ENVY/Smalltalk Debugger on (6/7/2026 14:59:57)
File Edit Processes Stack Inspector Info Breakpoints Watches Options
Debug Other...
Remove From List
Ask AI to Investigate...
Step Into F5
Step Over F6
Through Ctrl+F6
To Cursor
Run to Return
Stop
Resume F9
Terminate Ctrl+F9
Change Priority...
Change Name...
Define...
Into
To Cursor
Return
Resume
Define...
primitiveFailed: errorCode withArgument: argument backUp: backUp
"Report to the user that a method that is implemented as a system primitive
as failed. Use errorCode and argument number report the problem.
This error is a low level error and should not be NLS enabled."
^self error: ((NlsCatKRN residentMsg: 'MxKRN24') "NLS Primitive failed in: %1 due to %2"
bindWith: (Processor activeProcess methodAtFrame: backUp + 1) printString
with: (self primitiveErrorStringFor: errorCode argument: argument))
(6/6/1996 06:52:27) from CLDT
16.0.0x64-b604-dev: Application Window
AIAssistantCoreApp
AIAssistantUIApp
AIChat
filePaths
isGenerating
model
"Categories: Accessing"
filePaths
^ filePaths
"Categories: Accessing"
filePaths
"Returns the current file paths provided as context for
^ filePaths
Show only last changes
Discard
Cancel
Merge

ENVY version history tools

Compare the versions 'b600 + 74792d' and 'b604 +74872a' for the 'ENVY/Image SCI' config map and describe what has changed.



instantiations

VAST team uses the VAST AI

- Rapid prototyping
  - Code merge view; diff algorithm; user interface;...
- Test case generation
  - Repetitive work, sometimes a lot of boilerplate
- Assist in exploring bugs
- Fixed a couple of edge case bugs in the VM
- Generate platform function definitions for external libraries
  - Provide C header files and documentation to product a library wrapper
- Development of the AI Assistant itself
  - Rapid prototyping, exploration, ...

instantiations

Thank you for attending!

VAST AI Assistant

Questions?

Kris Gybels and Johan Brichau
Senior Software Engineers

kgybels@instantiations.com | jbrichau@instantiations.com

Contact

General Inquiry
info@instantiations.com

Sales
sales@instantiations.com

VAST Support Portal
vast-support.instantiations.com

North America, Toll Free
855 476 2558

International
+1 503 263 0058