



# Duo: AI for GitLab

LC User Meeting  
March 13<sup>th</sup>, 2025

**John Consolati**

Prepared by LLNL under Contract DE-AC52-07NA27344.

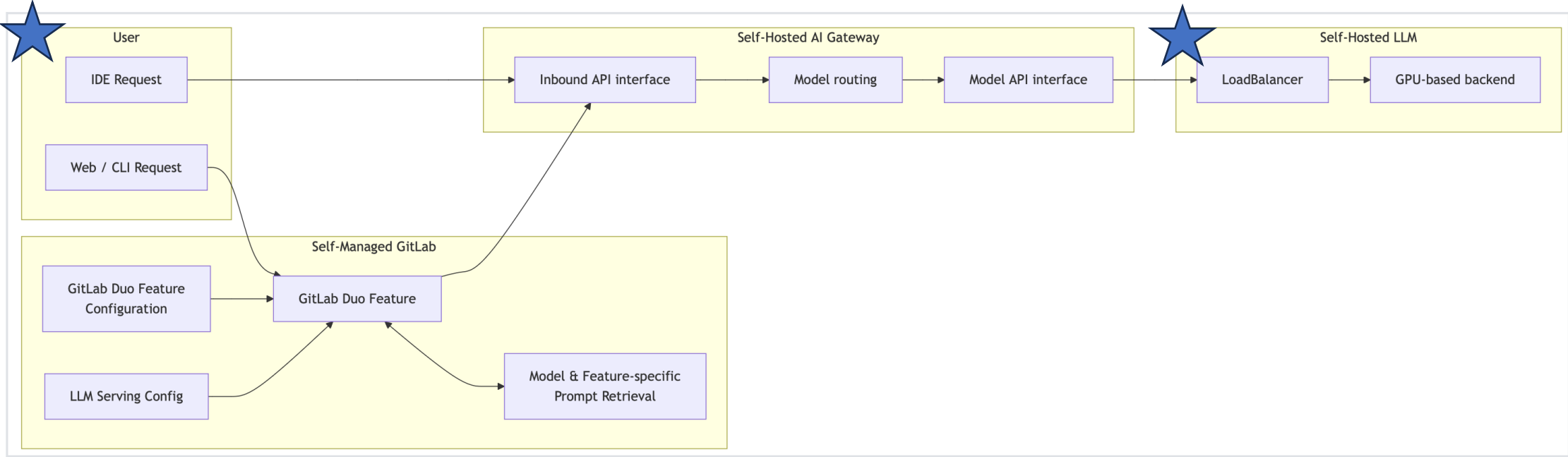
# What is it?

- New GitLab functionality that lets GitLab talk to an LLNL-managed large language model
- AI inference capabilities are accessible across the GitLab web UI, GitLab web IDE, and various desktop IDE plugins
- GitLab context-aware AI enables [many features](#):
  - Today: chat, code suggestions, code explanation, code refactoring, code fixes, test code generation
  - Future: Summarize discussions in issues/MRs, generate MRs automatically, root cause analysis for CI pipeline failures, code vulnerability analysis

# Where is it?

- Deployed as Early Access to CZ GitLab late January 2025, announced via [Tech Bulletin #576](#)
- LC purchased 100 seats for the pilot, and there are seats remaining for interested users. Contact [lc-gitlab-admin@llnl.gov](mailto:lc-gitlab-admin@llnl.gov)
- Exploring possibilities for RZ deployment (soon) and SCF (later). Details in subsequent slides.

# Component Architecture



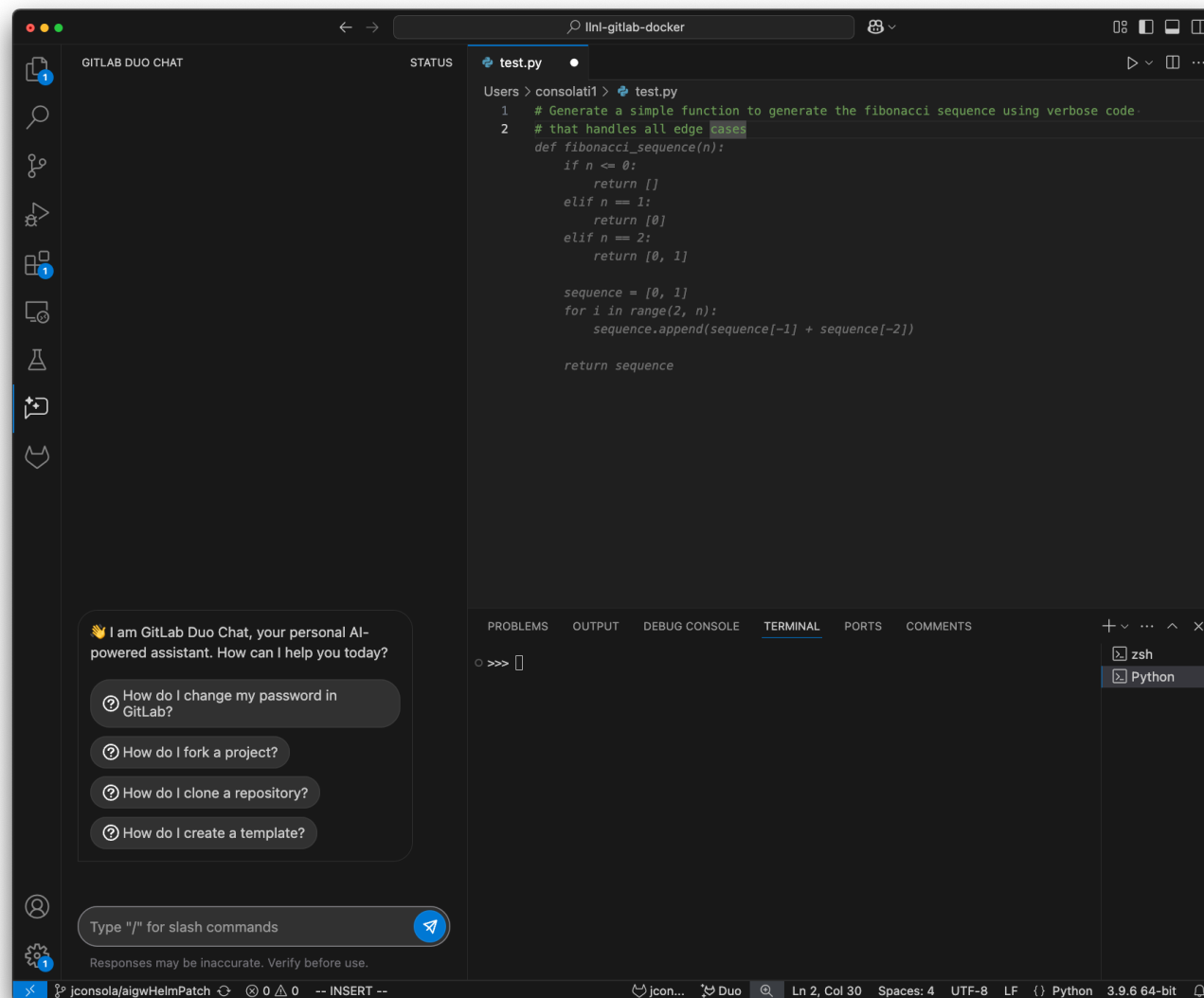
Source: [gitlab.com](https://gitlab.com)

- Duo introduces several new components, but this quick primer will focus on the end-user experience (starred). Where can I use it, and what models are backing it?

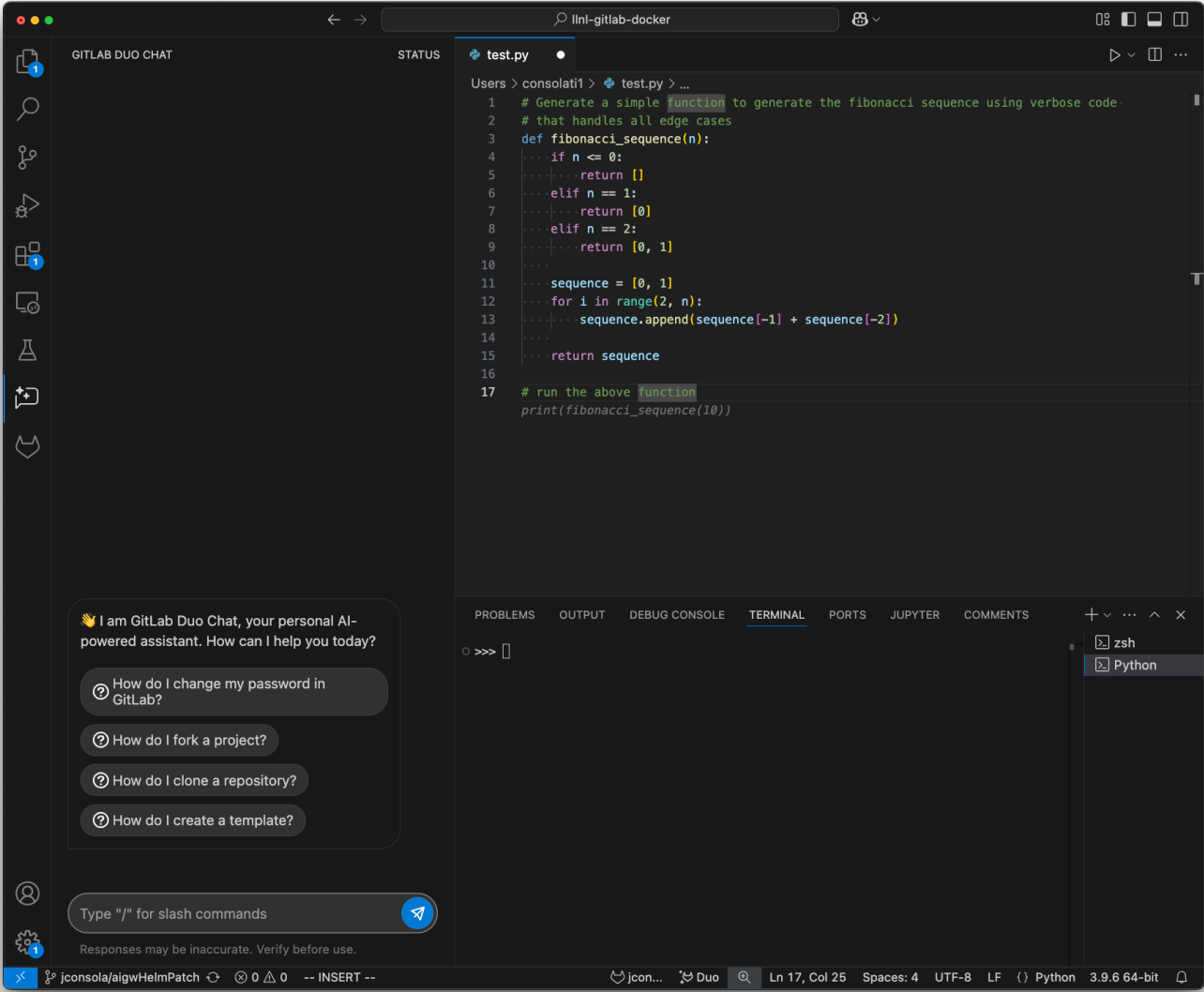
# User Desktop IDE Integration

- GitLab [provides plugins](#) for various desktop IDEs such as VSCode, JetBrains, Visual Studio
- Plugins authenticate to CZ GitLab using a Personal Access Token (PAT) over 443/HTTPS
  - Same PAT as today; subject to 30-day expiration

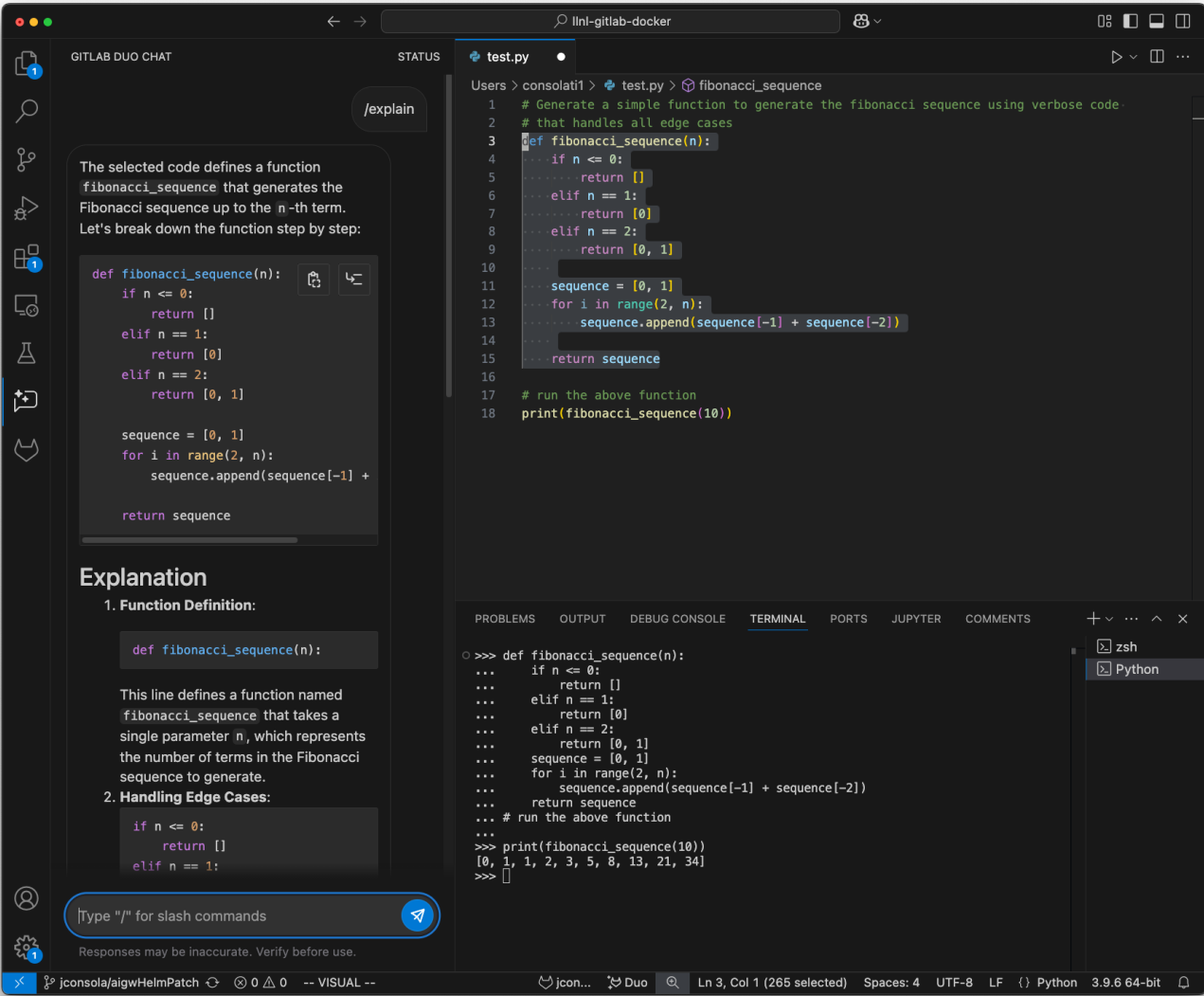
# Desktop IDE – Example Workflow



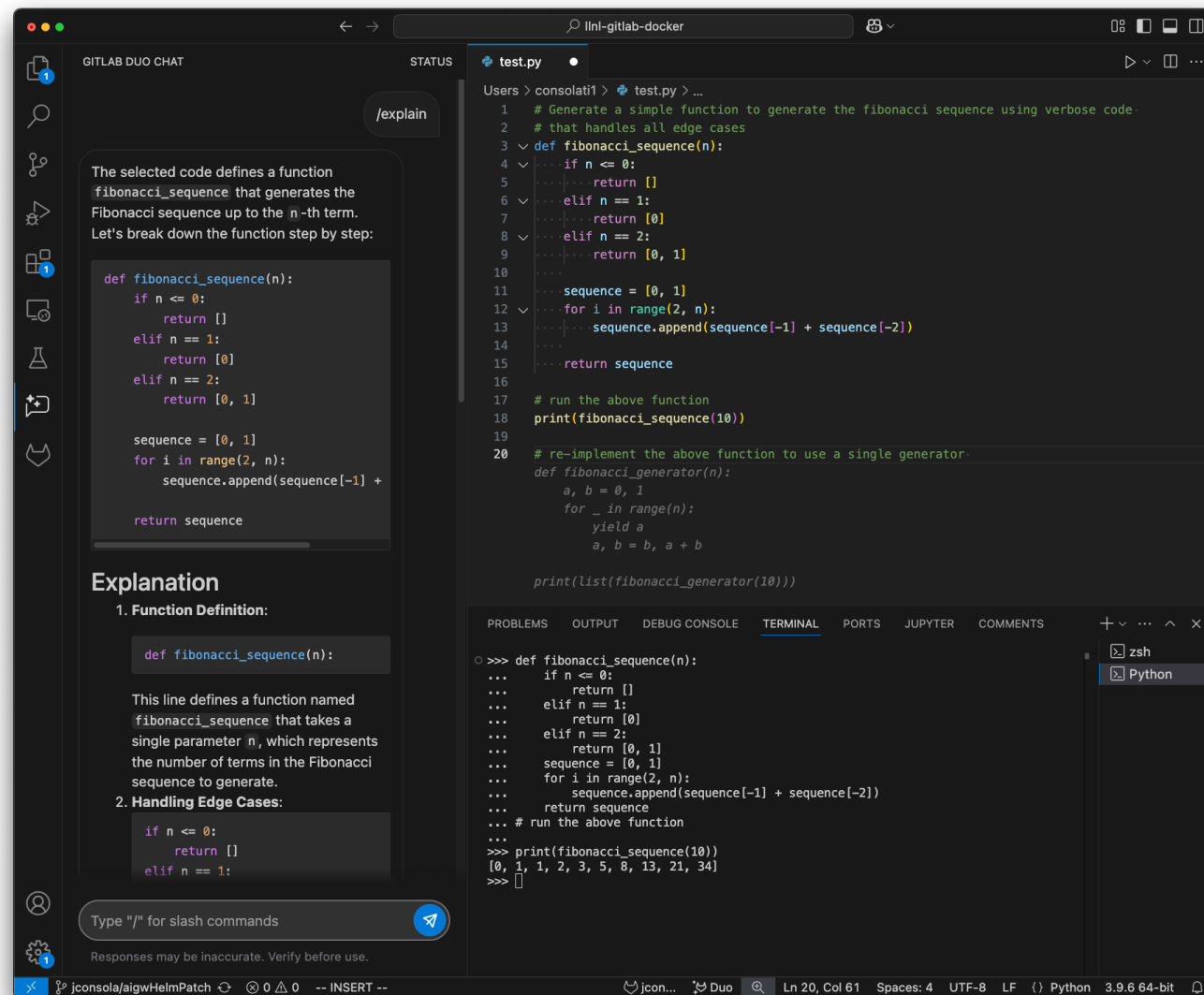
# Desktop IDE – Example Workflow



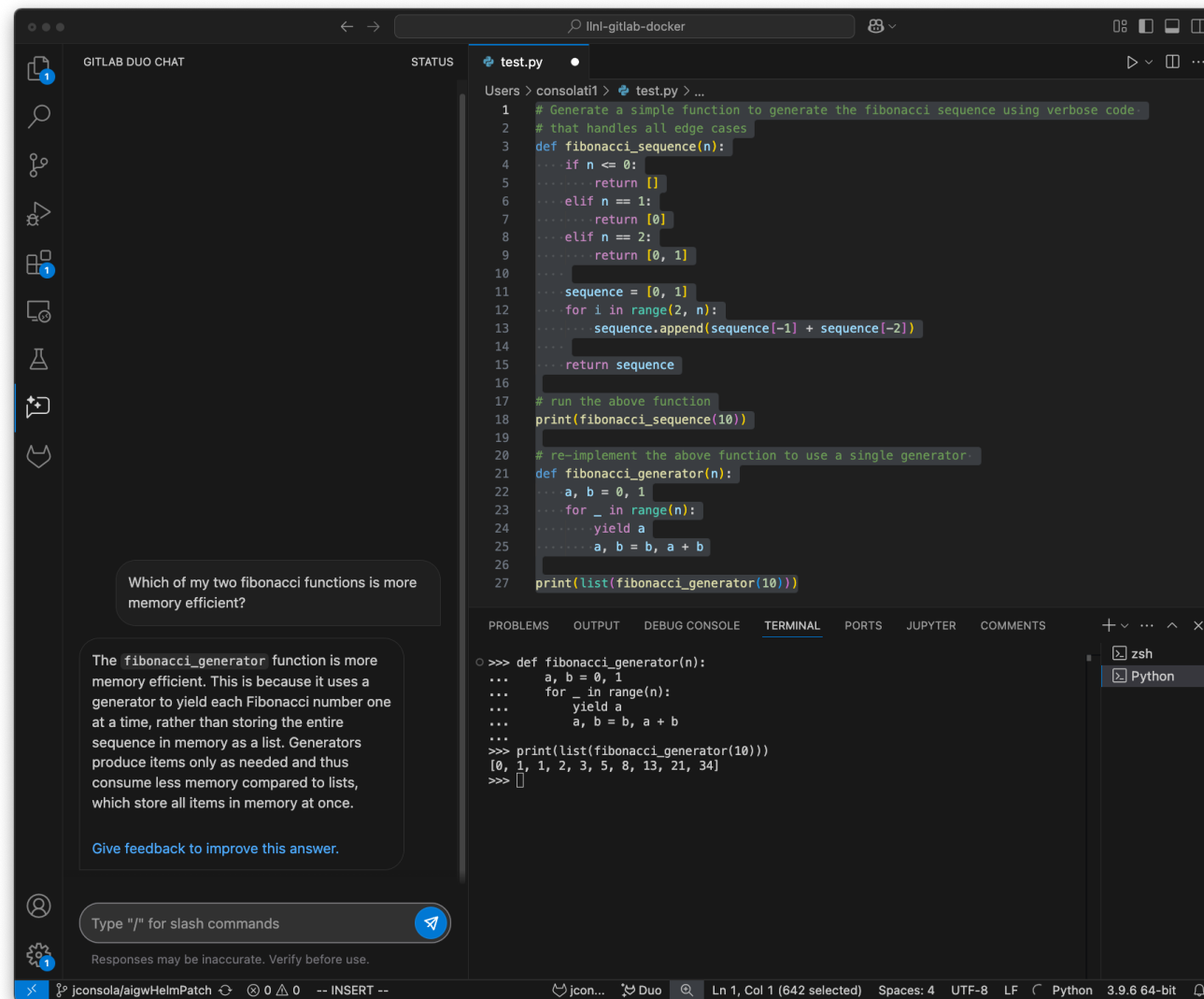
# Desktop IDE – Example Workflow



# Desktop IDE – Example Workflow



# Desktop IDE – Example Workflow



# Web UI / IDE Integration

- Duo chat also available on the GitLab web UI with the same capabilities
  - Chat sessions will carry over between the web and your IDE
- The web IDE within GitLab (VSCode) also supports Duo in a similar fashion

# Web UI - Example

Collaboration Zone (CZ) [LC Gitlab CI Info] [AI Features Early Access] [Connecting With SSH] [List C]

MFEM / mfem / Commits / 8058cb08

linalg/ginkgo.cpp

```

...    @@ -408,13 +408,8 @@ const
408    408    {
409    409        // Add the logger object. See the different masks available in Ginkgo's
410    410        // documentation
411    411        ~ #if MFEM_GINKGO_VERSION < 10500
412    412            convergence_logger = gko::log::Convergence<>::create(
413    413            executor, gko::log::Logger::criterion_check_completed_mask);
414    414        ~ #else
415    415            convergence_logger = gko::log::Convergence<>::create(
416    416            gko::log::Logger::criterion_check_completed_mask);
417    417        ~ #endif
418    418            residual_logger = std::make_shared<ResidualLogger<>>(executor,
419    419            system_oper.get(),b);
420    420
...    @@ -571,11 +566,7 @@ GinkgoIterativeSolver::Mult(const Vector &x, Vector &y) const
571    566        combined_factory->add_logger(convergence_logger);
572    567
573    568        // Finally, apply the solver to x and get the solution in y.
574    574        ~ #if MFEM_GINKGO_VERSION < 10600
575    575            solver->apply(gko::lend(gko_x), gko::lend(gko_y));
576    576        ~ #else
577    569            solver->apply(gko_x, gko_y);
578    578        ~ #endif
579    579
580    571        // Get the number of iterations taken to converge to the solution.
581    572        final_iter = convergence_logger->get_num_iterations();
...    @@ -1113,11 +1104,7 @@ GinkgoPreconditioner::Mult(const Vector &x, Vector &y) const
1113    1104            gko_array<real_t>::view(executor,
1114    1105                y.Size(),
1115    1106                y.ReadWrite(on_device)), 1);
1116    1116        ~ #if MFEM_GINKGO_VERSION < 10600
1117    1117            generated_precond.get()->apply(gko::lend(gko_x), gko::lend(gko_y));
1118    1118        ~ #else
1119    1107            generated_precond.get()->apply(gko_x, gko_y);
1120    1120        ~ #endif
1121    1108    }
1122    1109
1123    1110        void GinkgoPreconditioner::SetOperator(const Operator &op)
...    @@ -1211,11 +1198,7 @@ ILUPreconditioner::ILUPreconditioner(
1211    1198            .with_skip_sorting(skip_sort)
1212    1199            .on(executor);
1213    1200            precondition_gen = gko::preconditioner::ilu<>::build()
1214    1214        ~ #if MFEM_GINKGO_VERSION < 10700
1215    1215            .with_factorization_factory(fact_factory)
1216    1216        ~ #else

```

CitLab Duo Chat

What does this commit do?

The commit titled "Increase minimum Ginkgo version to 1.9.0" updates the minimum required version of the Ginkgo library from 1.4.0 to 1.9.0. This change involves modifying the `INSTALL` file to reflect the new version requirement and updating the code to remove conditional compilation directives that were previously used to handle different versions of Ginkgo. The changes ensure that the codebase is compatible with Ginkgo version 1.9.0 and above, simplifying the code by removing version-specific conditions.

Give feedback to improve this answer.

Type /help to learn more

Responses may be inaccurate. Verify before use.

# Supported Models & Hosting Environments

| Hosting Platform                        | GitLab Supported Models                                                                                                                                                                | Notes                                                                                                                                                    |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fully on-prem (\$\$\$)                  | <ul style="list-style-type: none"> <li>Mistral series </li> <li>Mixtral series </li> <li>CodeGemma (future) </li> <li>CodeLlama (future) </li> <li>DeepSeek Coder (future) </li> </ul> | <ul style="list-style-type: none"> <li>On-prem involves procuring new hardware or reconfiguring HPC nodes</li> <li>DeepSeek not LLNL approved</li> </ul> |
| Amazon Bedrock (\$) (GovCloud instance) | <ul style="list-style-type: none"> <li>Mixtral series </li> <li>Claude 3.5 Sonnet </li> </ul>                                                                                          | <ul style="list-style-type: none"> <li>Usable in CZ only</li> </ul>                                                                                      |
| Azure OpenAI (\$) (Government instance) | <ul style="list-style-type: none"> <li>GPT series </li> <li>Llama 3 series </li> </ul>                                                                                                 | <ul style="list-style-type: none"> <li>Usable in CZ only</li> <li>Pilot currently using GPT-4o</li> </ul>                                                |

Icons represent degree of accuracy in model responses for GitLab use cases

Combinatorial problem of model support, model accuracy, cost, security, and hardware availability

# What's next?

- Launching pilot for RZ GitLab. Will initially run models hosted on rzvernal.
- Evaluate viability of the Mixtral 8x22B-it model
  - CZ pilot previously ran Mistral 7B-it which was insufficient
  - Expecting results better than Mistral 7B but worse than GPT. Good enough?
- Ensure stable infrastructure prior to general availability launch
- Feature parity of Duo across different instances

# Seeking Feedback

- Accuracy & speed of responses
- Ease of providing context to the model
- Does it enable you to do things you couldn't previously? What are the time savings?
- How can we make this better?
- If you use both CZ & RZ, do you notice differences between GPT and Mixtral responses?
- [lc-gitlab-admin@llnl.gov](mailto:lc-gitlab-admin@llnl.gov)

