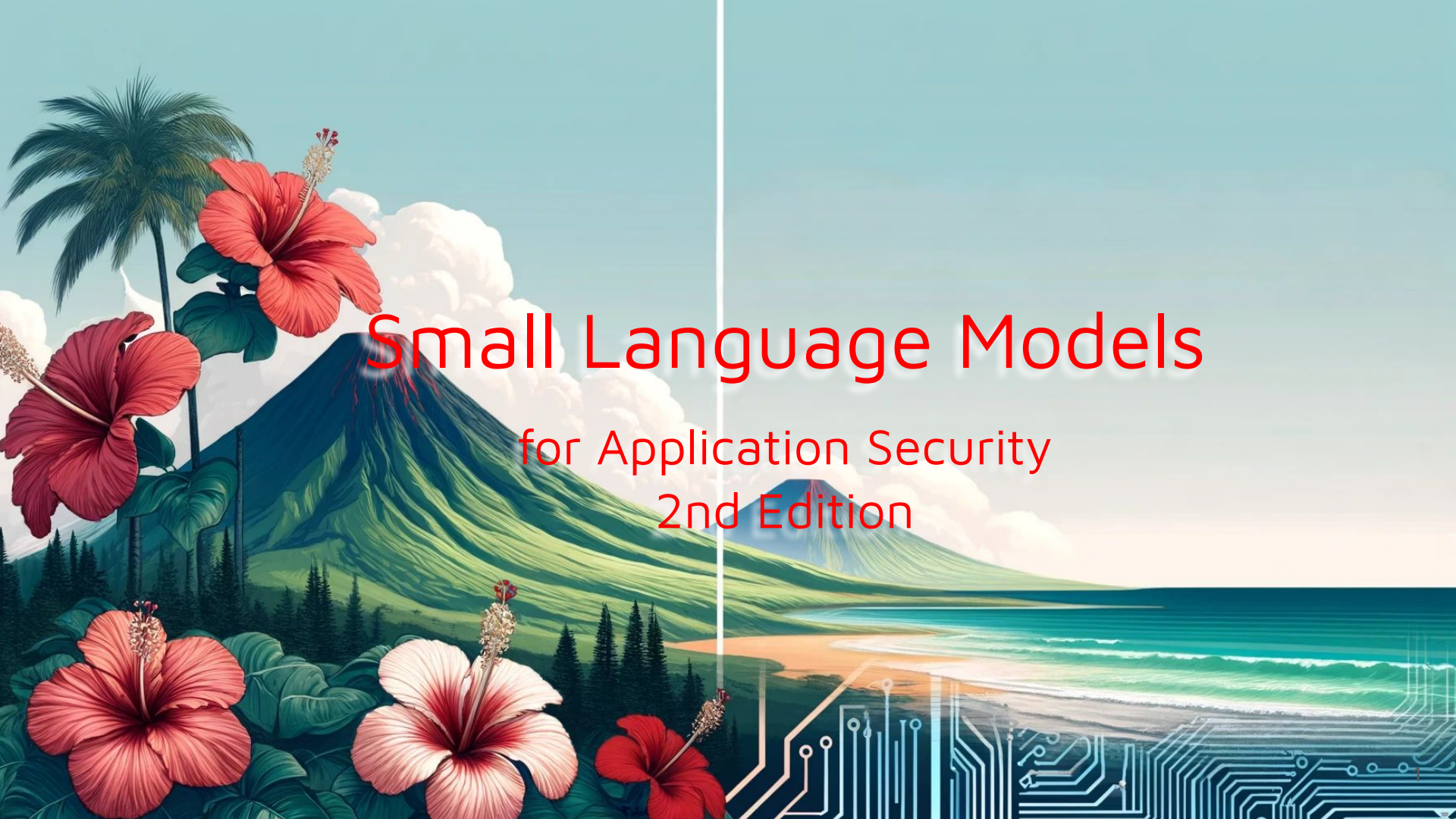

Small Language Models
for Application Security
2nd Edition

Louis Barrett

Security Researcher

Lead Security Researcher @ Exaforce

Louis Barrett

Security Researcher

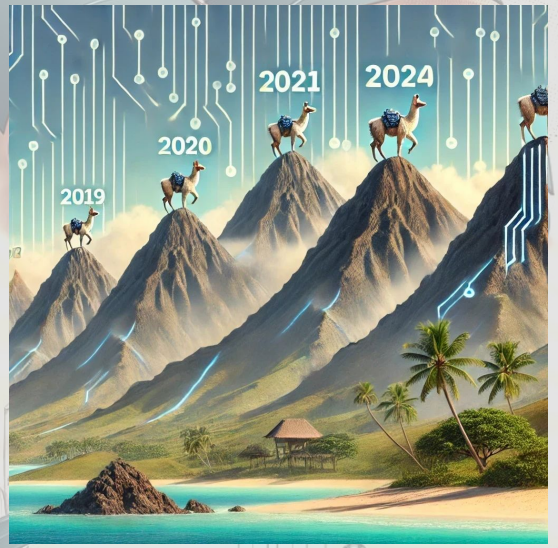
Agenda

- Introduction
  - LLM Basics
  - Why Small Language Models
- Application Security
  - SDLC Responsibilities
  - Automation Opportunities
- Planning
  - Design Goals & Requirements
  - Infrastructure Components
- Building
  - Requirements
  - Setup
  - Workflow
- Testing
  - Model Evaluation
  - Benchmark Results
  - Top Model choices
- Demo


Introduction

What are Large Language Models

“Large Language Models are Artificial Intelligence systems utilizing deep neural networks, and massive amounts of data to interpret and generate coherent text... ”



How did we get here?

2020
- **June - OpenAI Releases GPT3**

2022 🏖️
- September - Etherium merge indirectly reduces GPU prices.
- Open AI Chat GPT released

2023 🏖️
- March - Meta Llama Leaks
- May - Mosaic ML, MPT 7b released
- Oct - Mistral AI, Mistral 7b released
- **Dec - Mistral AI, Mixtral of Experts released**

2024
- Apr - Hugging Face TGI moves to Apache 2.0 License
- May - Microsoft Phi 3 released
- June - Alibaba Qwen2 released

LLM Basics

What are they capable of...

Generation
- Create natural language output from a user's prompt input.

Summarization
- TLDR functionality for large amounts of text.

Question Answering
- Provide answers to a query, based on the provided prompt and context.

Sentiment Analysis
- Analyzing the context and tone of a piece of text.

Classification
- Determine the category of internal or input data

Instruction Following
- Follow instructions provided by the user.

Why Small Language Models

What is a Small Language Model?

“Small language models are those with under 30 billion parameters. For the sake of this presentation we will focus mainly on models in the 7 billion parameter range...”

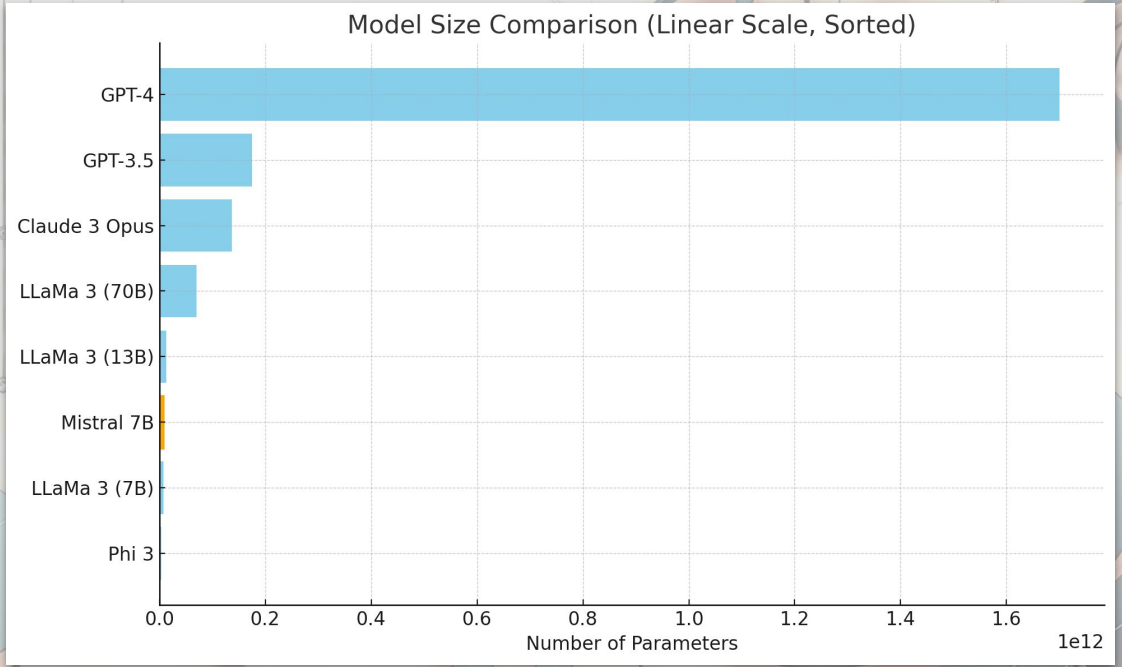
Why Small Language Models

Model Size Comparison

- GPT 4 - 1.7 Trillion Parameters*
- GPT 3.5 - 175 Billion Parameters
- Claude 3 Opus - 137 Billion Parameters
- LLaMa 3 - 7, 13, 70 Billion Parameters
- Mistral 7b - 7 Billion Parameters
- Phi 3 - 3, 7, 14 Billion Parameters

Why Small Language Models

Model Size Comparison



Why Small Language Models

Advantages

- Fast generation speed (higher tokens per second)
- Shorter model load times
- More resource-efficient and cost-effective
- State-of-The-Art* performance despite size


Application
Security

Application Security

“Application Security is about securing the Software Development Lifecycle…”

Application Security

Organization Responsibilities

- Define Security Policies
  - Secure Development
  - Change Control
  - Vulnerability Management
- Educating Engineers
  - Where to find relevant policies
  - Provide Training

Application Security

Individual Responsibilities

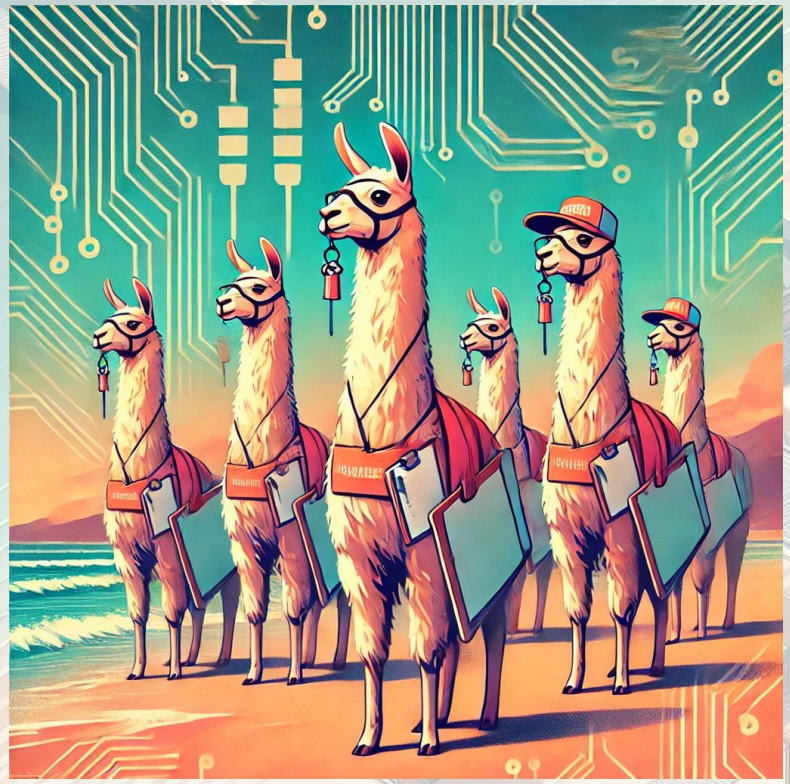
- Software Design Review (Design)
- Threat Modeling (Design)
- Code Review
- Vulnerability Management

Application Security

Individual Responsibilities

- Software Design Review (Design)
- Threat Modeling (Design)
- Code Review
- Vulnerability Management

Application Security



All Responsibilities Require

- Communication
- Reading Documentation
- Drawing Conclusions
- Interacting with Stakeholders (Engineers)

Application Security

"All Security reviews are effectively Summarization, Instruction Following and Question Answering workflows..."


Automation
Opportunities

Automation Opportunities

When to use a language model

- High volumes of unstructured, and structured data
- A closed ended step-by-step process exists
- Work can be completed by junior engineers, with proper documentation.

Automation Opportunities

When to use a language model

Summarization

- You have a large volume of documents, or code you need to read for correctness

Instruction Following

- There is a well defined procedure for accomplishing the task in front of you such as a runbook or internal procedure

Question Answering

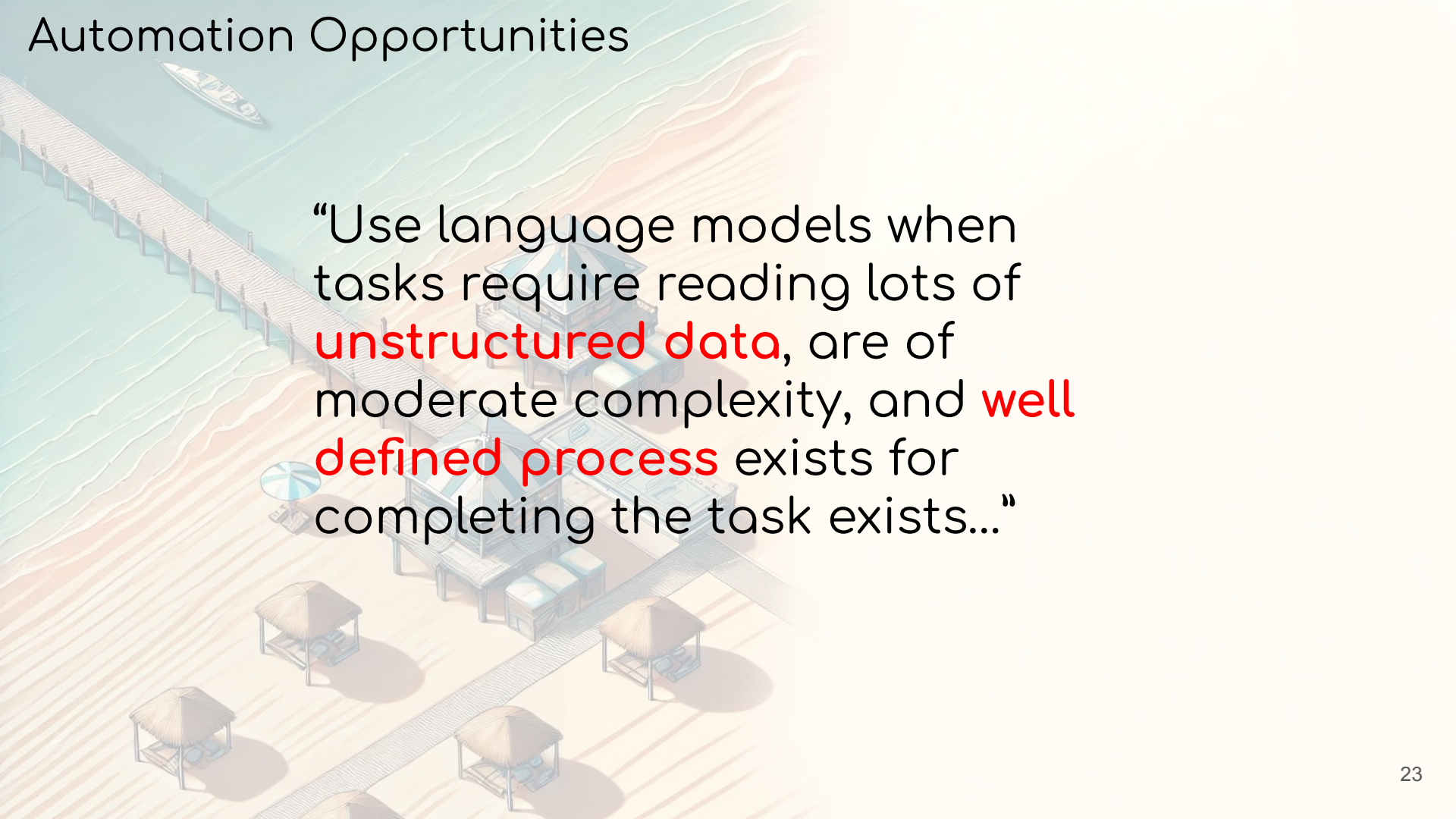
- The task can be accomplished by asking a finite number of questions and coming up with a grade

Automation Opportunities

When ***not*** *to* use a language model

- Data volume is low enough for a human to complete it reasonably quickly
- A repeatable process does not exist for the task you wish to translate
- Existing solutions are adequate

Automation Opportunities

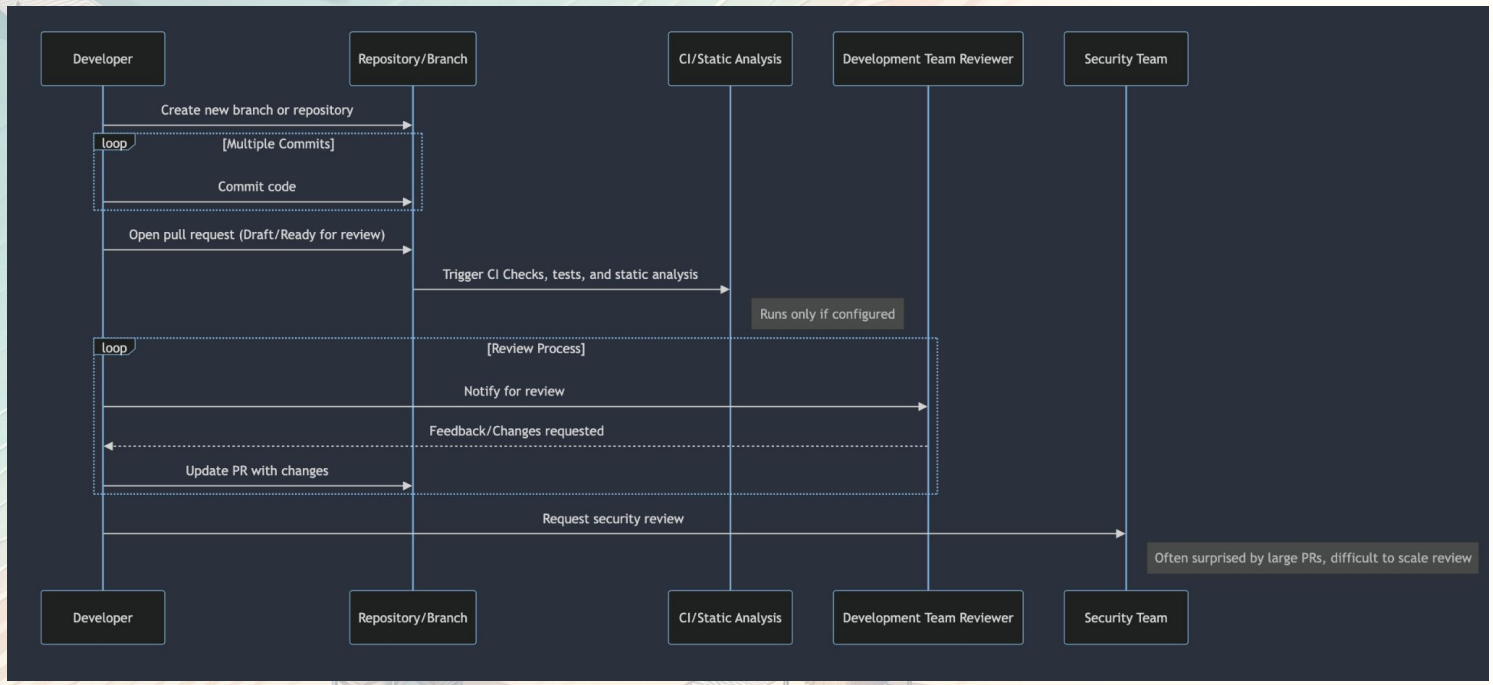


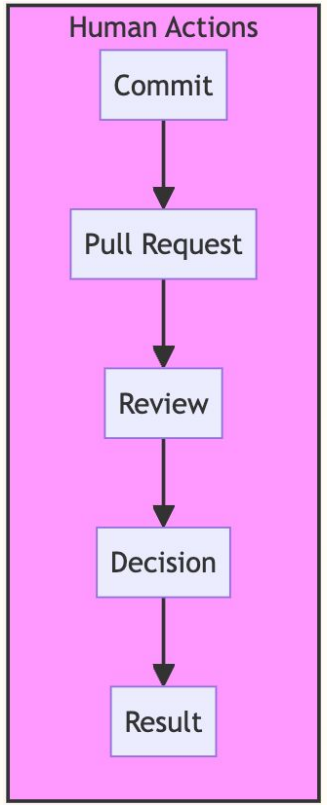
"Use language models when tasks require reading lots of unstructured data, are of moderate complexity, and well defined process exists for completing the task exists..."


Planning

Planning

Manual Process





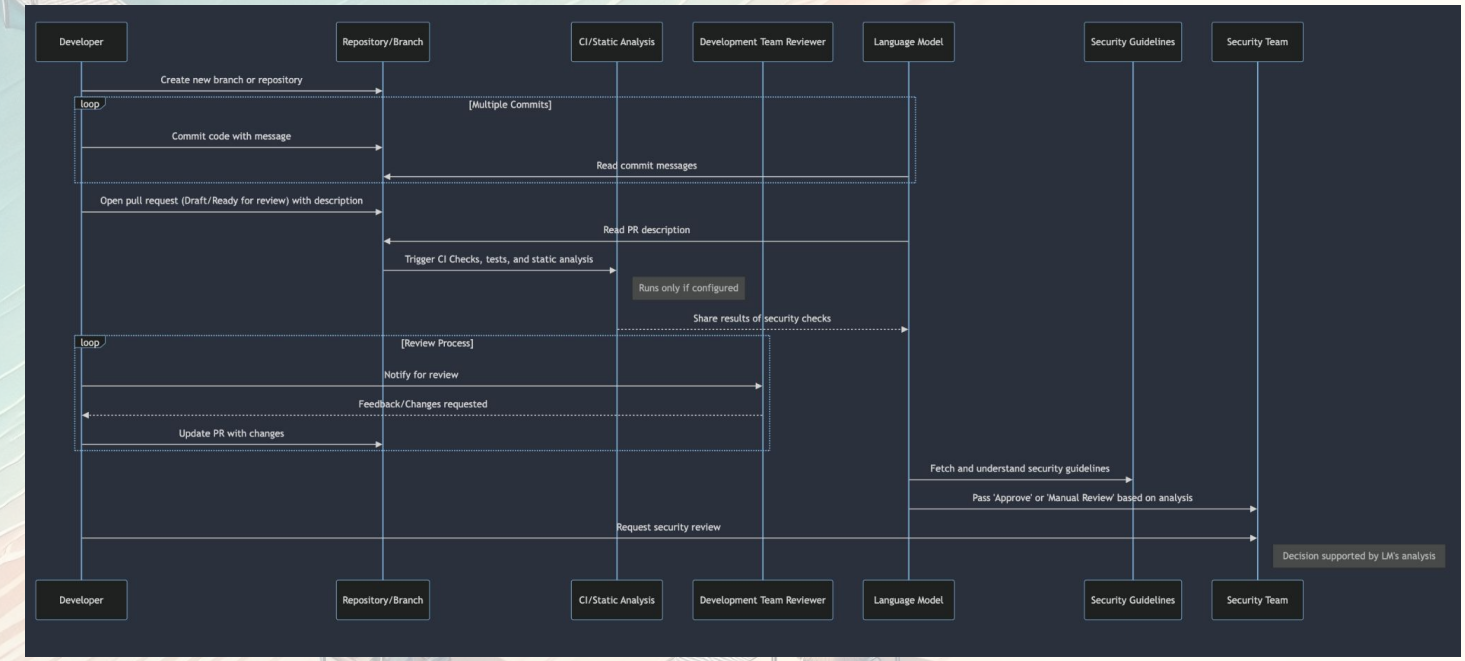
Planning

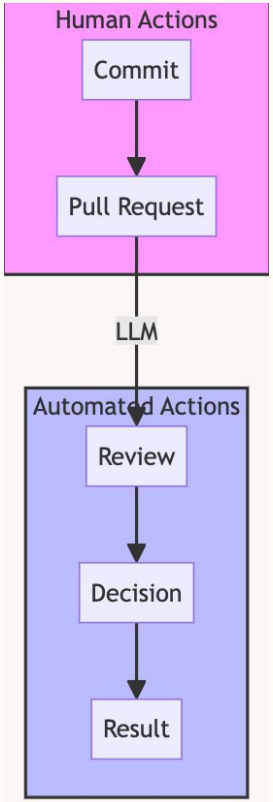
Design Goals

- Simplify the review process for human users.
- Integrates into existing development workflow(s)
- Produces a result at least as quickly, and accurately as a junior human reviewer.
- Maintains data privacy through the use of self-hosted models.

Planning

Model Assisted Process





Planning

Design Requirements

- Triggered by the development process
- Able to retrieve security relevant information such as secure development standards
- Notifies stakeholders of review results
- Does not leak source code to external sources

Planning

LM Application Tiers

	Level	Input Size	Actors	Data Sources	Interaction	Examples
1	Basic	Small	Human Language Model	Training Data User input	Single Turn	GPT Playground
2	Chatbot	Medium	Human **System User** Language Model	Training Data User input	**Multi-Turn**	Chat GPT (2023)
3	Retrieval Augmented Chatbot	Large	Human System User Language Model	Training Data User input **External Data**	Multi-Turn **Multi-Phase**	Chat GPT Claude Perplexity
4a	Agents	Medium-Large	Human System User Language Model **Goal Manager**	Training Data User input External Data **Tools**	Multi-Turn Multi-Phase **Autonomous**	Scale Donovan Agent GPT
4b	Multi-Model/Agent Orchestration	Large	Human System User Language Model Goal Manager **Orchestrator**	Training Data User input External Data Tools Orchestrator Input	Multi-Turn Multi-Phase Autonomous	Perplexity
5	Cohorts	Very Large	Human System User Language Model Goal Manager Orchestrator **Model Cohorts**	Training Data User input External Data Tools Orchestrator Input **Cohort Input**	Multi-Turn Multi-Phase Autonomous **Event Based**	

Planning

LM Application Tiers

	Level	Input Size	Actors	Data Sources	Interaction	Examples
1	Basic	Small	Human Language Model	Training Data User input	Single Turn	GPT Playground
2	Chatbot	Medium	Human **System User** Language Model	Training Data User input	**Multi-Turn**	Chat GPT (2023)
3	Retrieval Augmented Chatbot	Large	Human System User Language Model	Training Data User input **External Data**	Multi-Turn **Multi-Phase**	Chat GPT Claude Perplexity
4a	Agents	Medium-Large	Human System User Language Model **Goal Manager**	Training Data User input External Data **Tools**	Multi-Turn Multi-Phase **Autonomous**	Scale Donovan Agent GPT
4b	Multi-Model/Agent Orchestration	Large	Human System User Language Model Goal Manager **Orchestrator**	Training Data User input External Data Tools Orchestrator Input	Multi-Turn Multi-Phase Autonomous	Perplexity
5	Cohorts	Very Large	Human System User Language Model Goal Manager Orchestrator **Model Cohorts**	Training Data User input External Data Tools Orchestrator Input **Cohort Input**	Multi-Turn Multi-Phase Autonomous **Event Based**	

Planning

Design Requirements - Tier 3

- **Memory**: Extensive Short-Term Memory
  - Track file diffs, commit messages, status of checks etc.
- **Input Size:** Large (using window)
  - File diffs, and comments may be lengthy
- **Actors:** Human User, System User, Language Model
  - System prompt defines rules of engagement, and app purpose.
- **Data Source(s):** Model, User Input, External Indexes (Documentation)
  - Github platform, we treat diffs, checks, and commits as indexes
- **Interactions:** Multi-Turn, Multi-Phase
  - Stream of changes in PR treated as conversation

Planning

LM App Core Components

Inference Server

- Provides hosting and configuration for our models.

Automation Host

- Manages the the review workflows and integrates with source control.

Retrieval Host

- Provides storage for key information needed for reviews, and any internal documentation.

Planning

LM App Core Components

Inference Server - **TGI**

Automation Host - **N8N**

Retrieval Host - **Marqo**

LM App Optional Components

Code Search - **Sourcegraph**

Planning

Inference Server

Hugging Face Text Generation Inference

Launching TGI

Let's say you want to deploy teknium/OpenHermes-2.5-Mistral-7B model with TGI on an Nvidia GPU. Here is an example on how to do that:

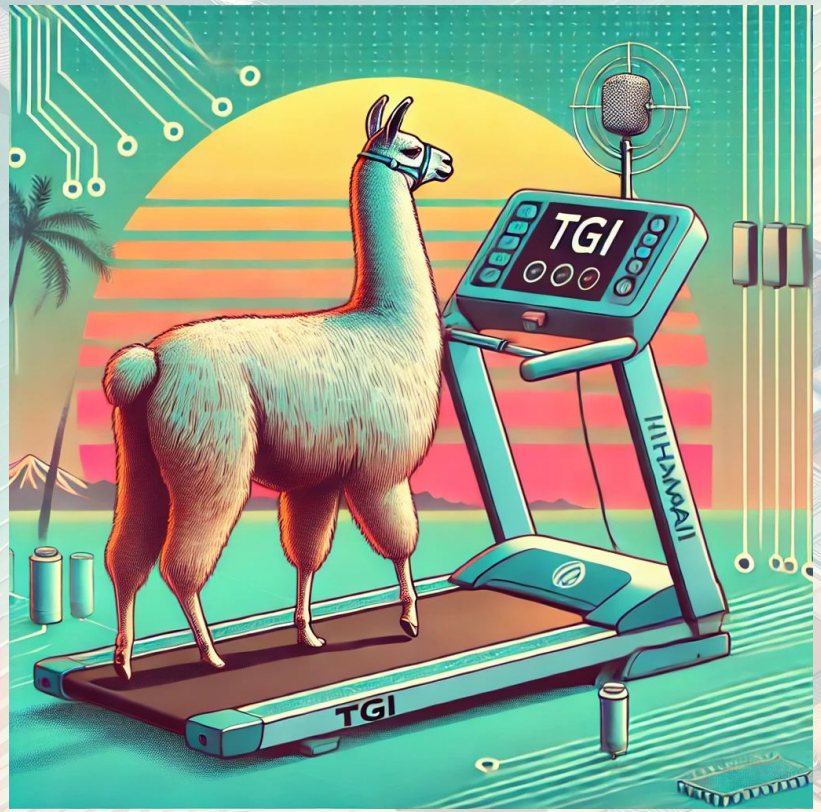
```
model=teknium/OpenHermes-2.5-Mistral-7B
volume=$PWD/data # share a volume with the Docker containe

docker run --gpus all --shm-size 1g -p 8080:80 -v $volume:/
    ghcr.io/huggingface/text-generation-inference:2.1.1 \
    --model-id $model
```

Supported hardware

TGI supports various hardware. Make sure to check the Using TGI with Nvidia GPUs, Using TGI with AMD GPUs, Using TGI with Gaudi, Using TGI with Inferentia guides depending on which hardware you would like to deploy TGI on.

Planning



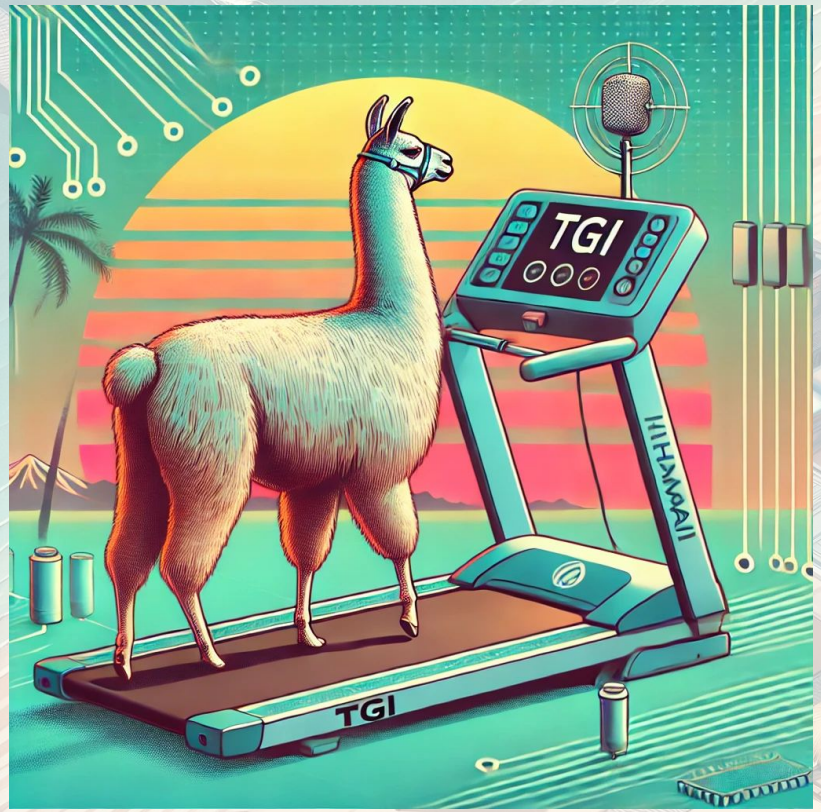
Inference Server

Hugging Face Text Generation Inference

Features:

- High model compatibility
- Multi-GPU inference
- Quantized Models (EETQ)
- Easily Tuned for low memory environments
- Automatic Cache Management

Planning



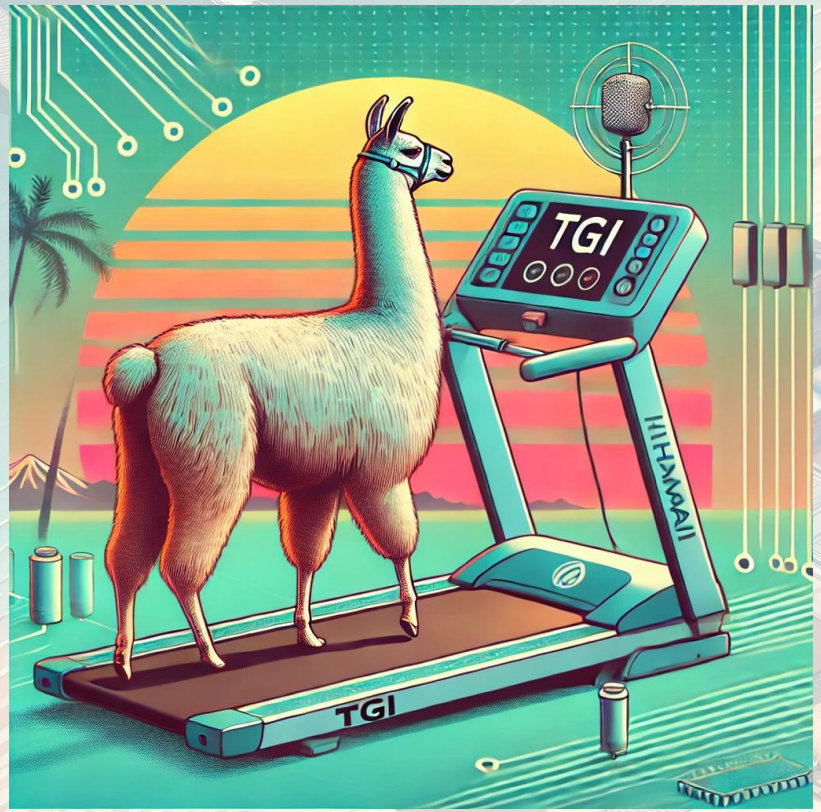
Inference Server

Hugging Face Text Generation Inference

Perks:

- Maintains compatibility for you – no more worrying about mismatched PyTorch, CUDA or Transformers versions.
- Offers OpenAI compatible endpoints – allowing it to function as a drop in replacement.

Planning



Inference Server

Hugging Face Text Generation Inference

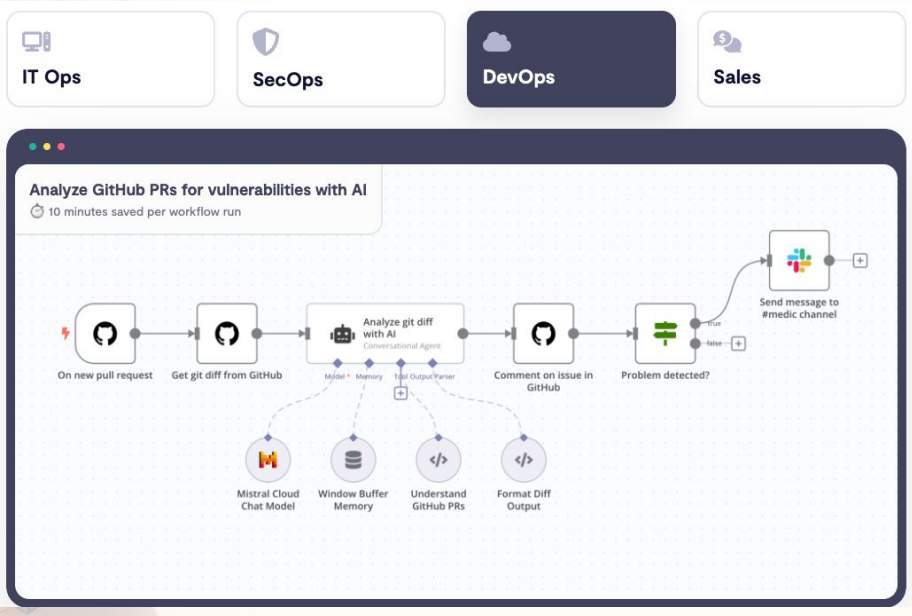
Supported Model Architectures:

- Idefics 2
- Llava Next (1.6)
- Llama
- Phi 3
- Gemma
- PaliGemma
- Gemma2
- Cohere
- DBRX
- Mamba
- Mistral
- Mixtral
- GPT Bigcode
- Phi
- Falcon
- StarCoder 2
- Qwen 2
- OPT
- T5
- Galactica
- SantaCoder
- Bloom
- MPT
- GPT2
- GPT NeoX
- Idefics
- Baichuan

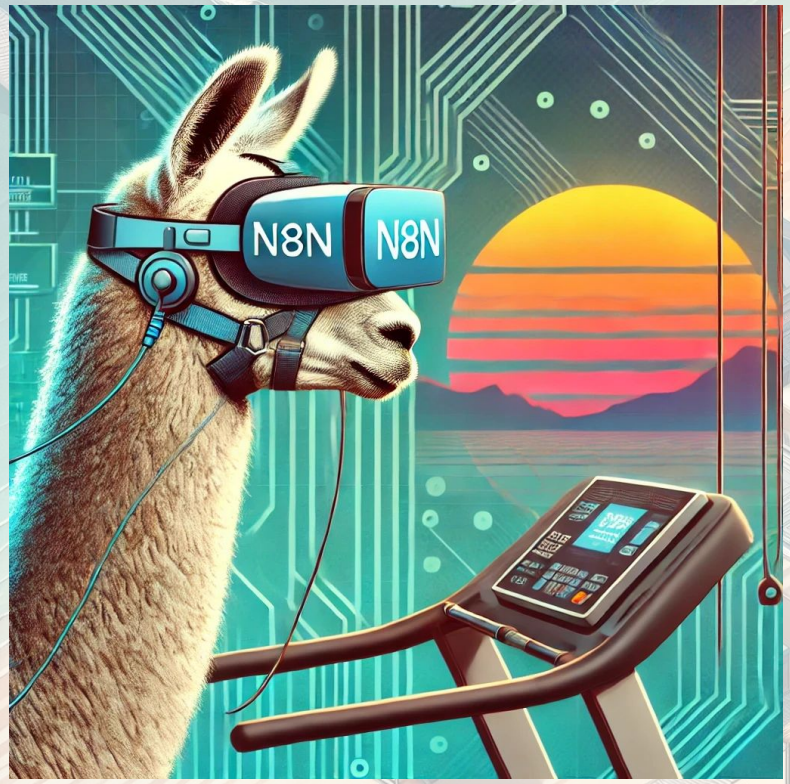
Planning

Automation Host

N8N Workflow Automation



Planning



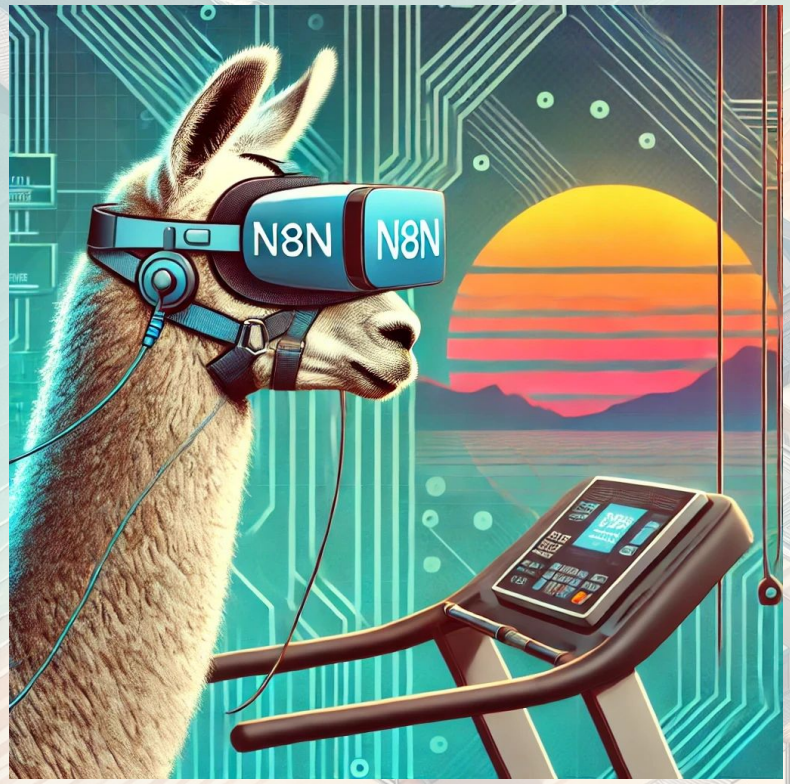
Automation Host

N8N Workflow Automation

Features:

- 1000+ Integrations
- Natively supports AI workflows
  - Chat
  - Retrieval
  - Agents
- Easy No-Code Interface

Planning



Automation Host

N8N Workflow Automation

Perks:

- Self-Hosted
- Built-in Secrets Management
- SSO Support
- Community Workflow Gallery

Planning

Retrieval Host

Marqo

Planning

Retrieval Host

Marqo

Features

- End-to-End Vector Search – Documents in Documents out.
- Multimodal search capabilities – Able to map text and images to the same vector space
- State-of-the-Art Models – includes pre-configured models, with support across many use cases

Planning

Retrieval Host

Marqo

Perks:

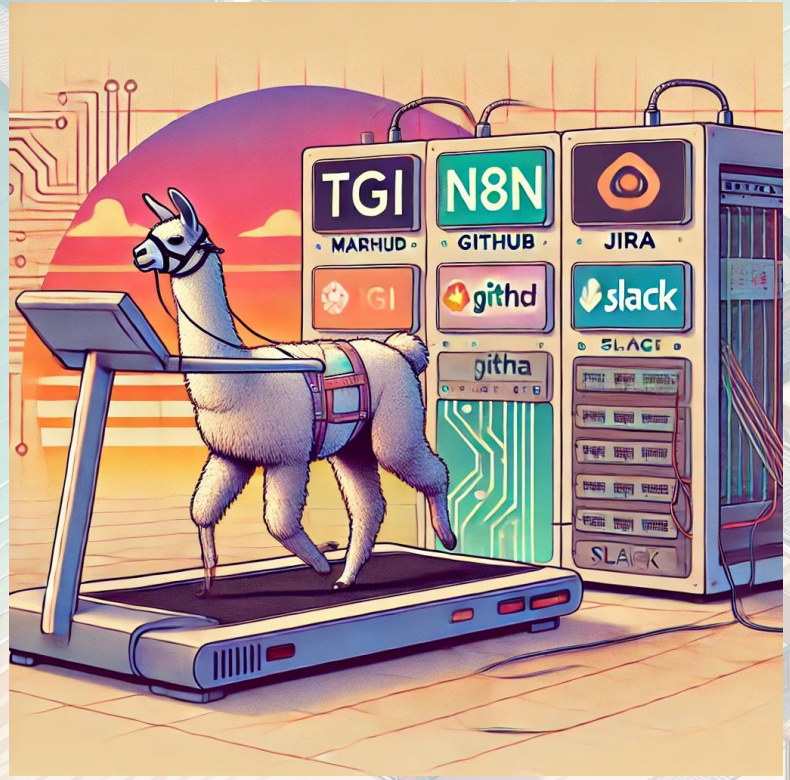
- Self-Hosted
- Fast - Backed by Vespa
- Simple to deploy
- Batteries included*

Planning

"Comprehend your SLM-assisted workflow, pinpoint your application's complexity tier, and handpick components that align with your specific needs and scale."


Building

Building



Minimum System Requirements

Inference Server

- System
  - CPU: 8 Cores
  - Memory: 32 GB
- GPU
  - Memory: 12 GB
- Storage
  - Model Storage: 256 GB

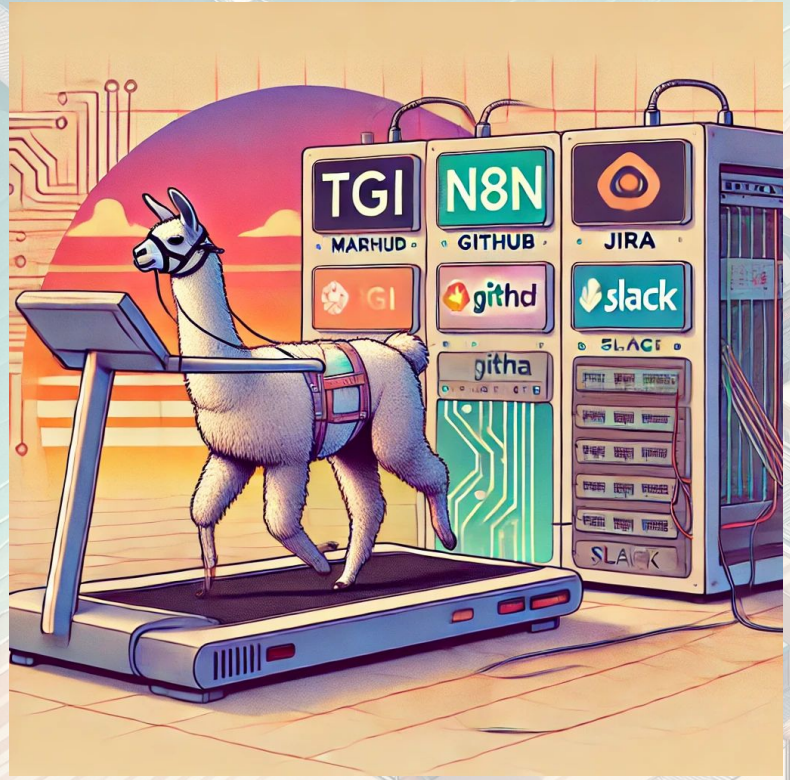
Automation Host

- System
  - CPU: 2 Cores
  - Memory: 2 GB
- Storage
  - Model Storage: 2 GB

Retrieval Host

- System (3 Nodes)
  - CPU: 4 Cores
  - Memory: 16 GB
- GPU (Optional)
- Storage: 20GB

Building

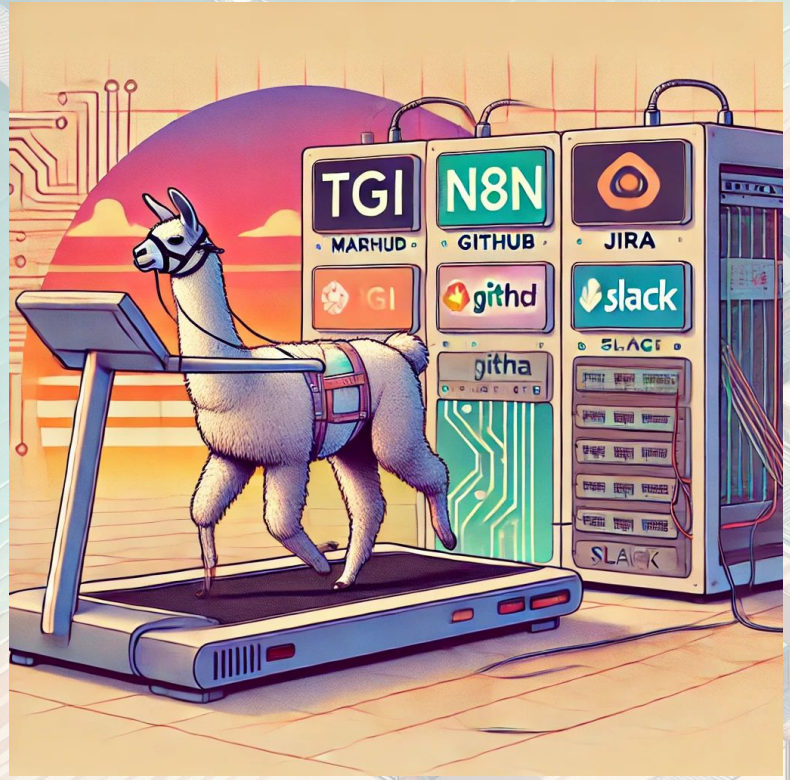


Evaluated System Specifications

Inference Server + Automation Host + Retrieval Host

- System
  - CPU: 32 Cores
  - Memory: 256 GB
- GPU
  - Memory: 3 x 24 GB
- Storage
  - Model Storage: 4 TB

Building



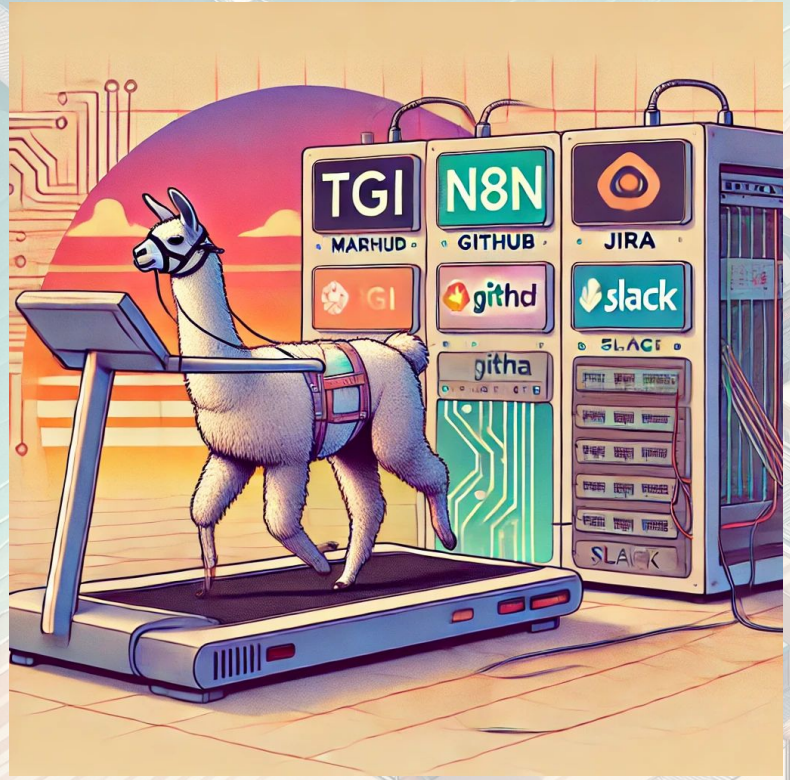
Infrastructure

- Components needed to run our models and workflows.

Workflows

- Application logic for performing application security tasks.

Building



Infrastructure

Setup - Inference Host

- Install Lambda Stack
- Install Docker
- Install Nvidia Container Runtime
- Verify containers can access GPU
- Pull Inference Server Image
- Install huggingface-cli (optional)

Setup - Workflow Server

- Install N8N
- Enable external access from Github (webhooks)

Setup Retrieval Host

- Vespa
- Marqo
- Sourcegraph

Building

Setup - Inference Host

Success!

N8N Running

Text Generation Inference Server Running

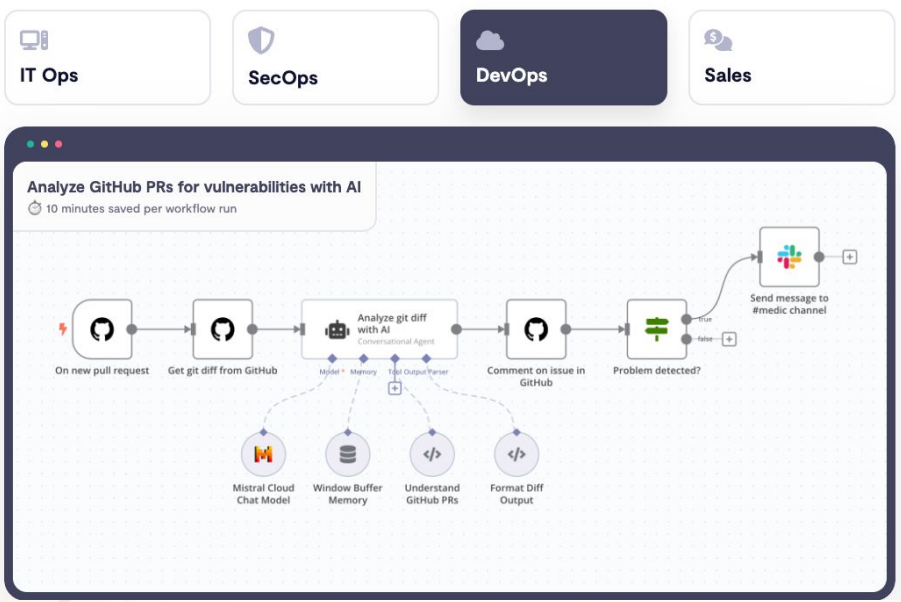
Next Up:

The workflow...

Building

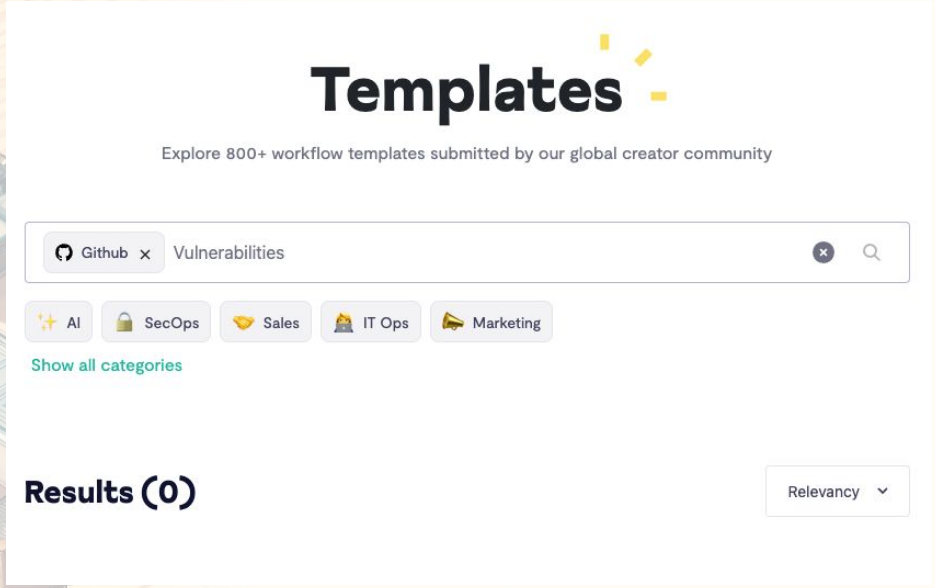
N8N Workflow

Now we can just pull the Github Vulnerability Review Template and we're done...



Building

N8N Workflow



Building

N8N Workflow

Although there doesn't seem to be an N8N maintained workflow for this, it includes *all* the tools necessary to build it...

Triggers

- Start a workflows, or make it interactive

Nodes

- Components for reading, writing and executing against integrations

Models

- Wrappers around inference provider APIs

Chains

- Components connecting models, and prompts to other components

Building

N8N Workflow

Although there doesn't seem to be an N8N maintained workflow for this, it includes *all* the tools necessary to build it...

Triggers

- Github
  - On Pull Request (Synchronize)
  - On Comment
  - On Check Run/Completed

Nodes

- Github
  - Review Actions
- Marqo
  - Read/Write Documents
- HTTP
  - API Call to Sourcegraph

Building

N8N Workflow

Although there doesn't seem to be an N8N maintained workflow for this, it includes *all* the tools necessary to build it...

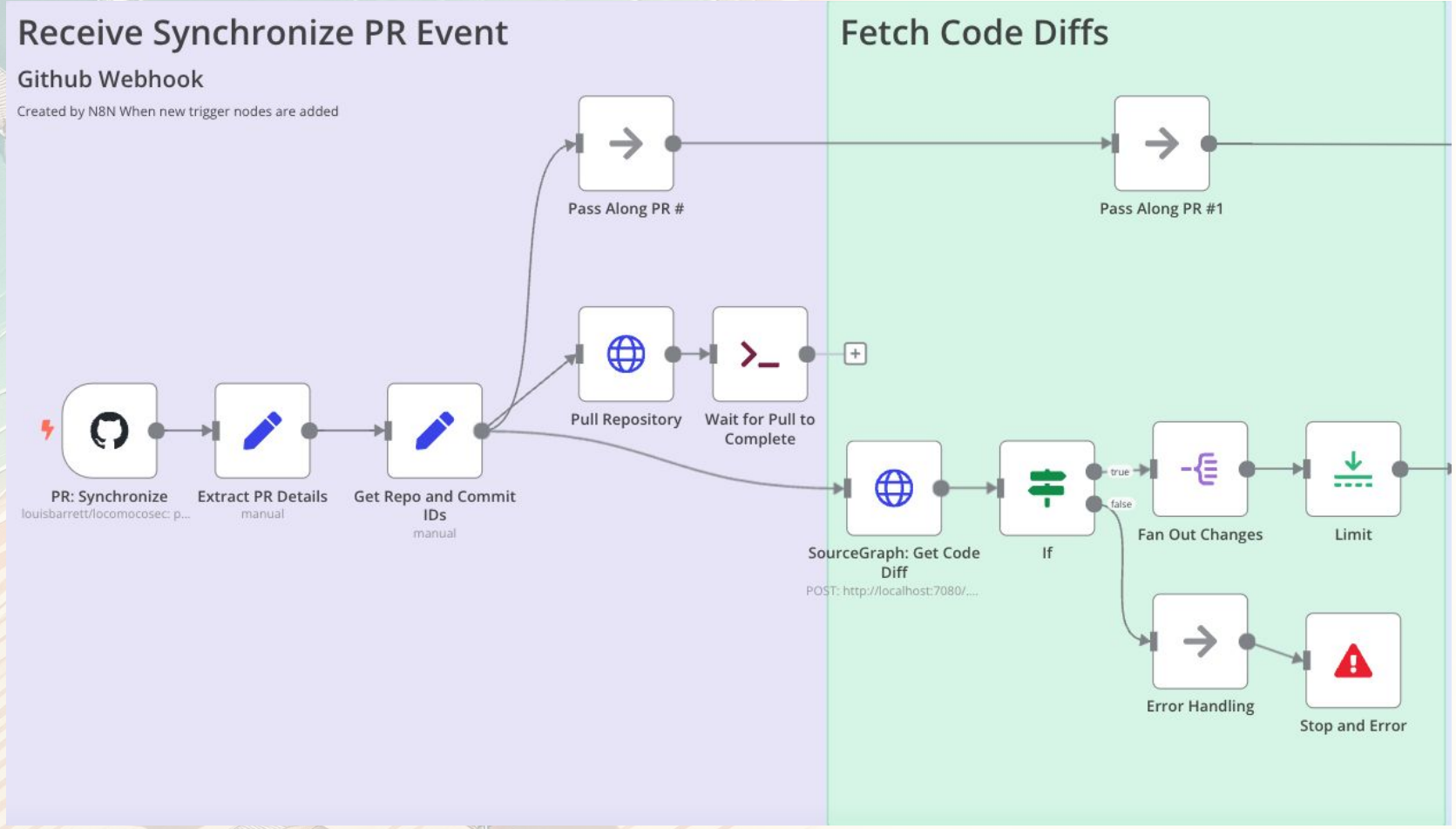
Models

- Open AI or Text Generation Inference (both are served by a TGI API)

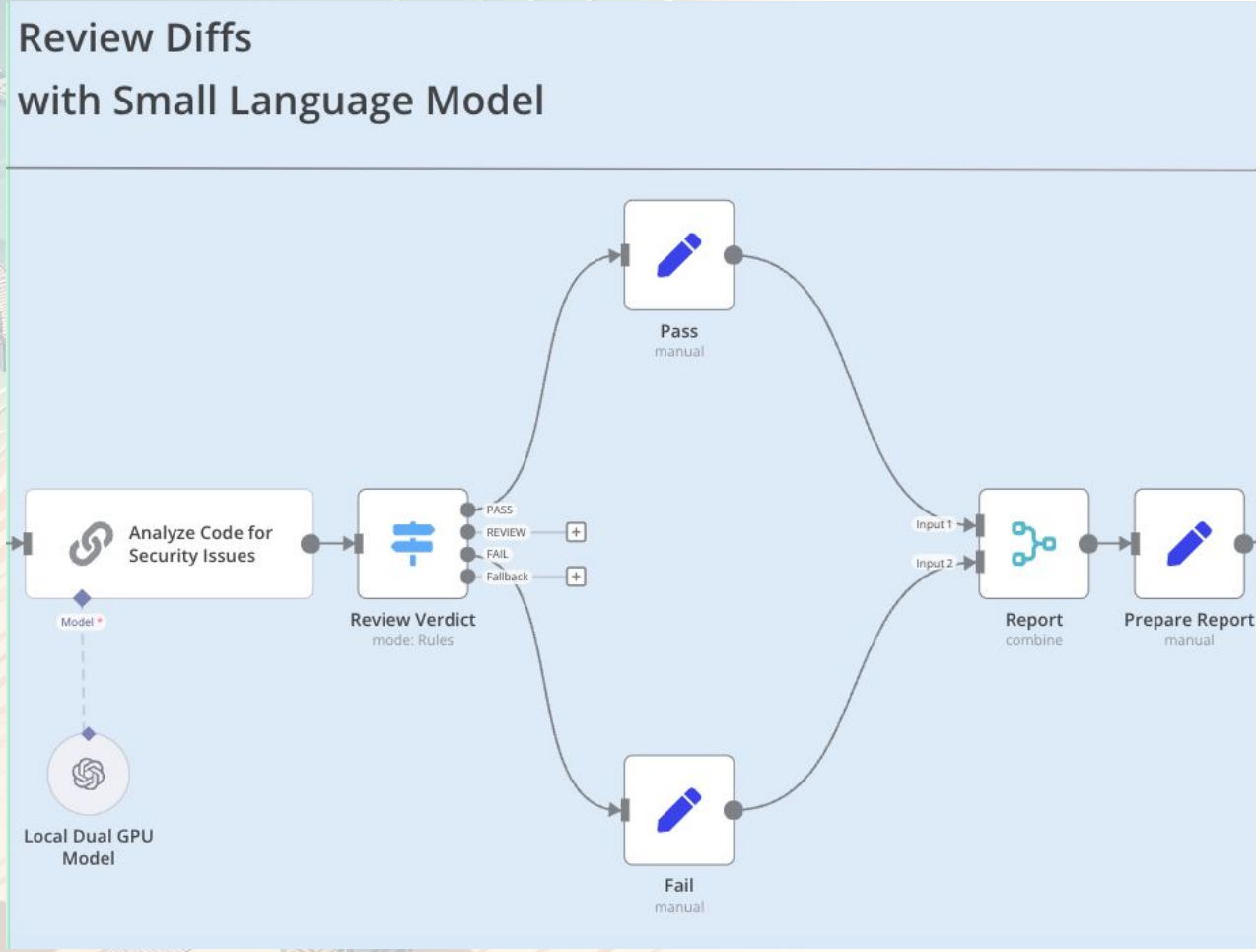
Chains

- Basic LLM Chain - Simplest AI component with input, model, and output links.

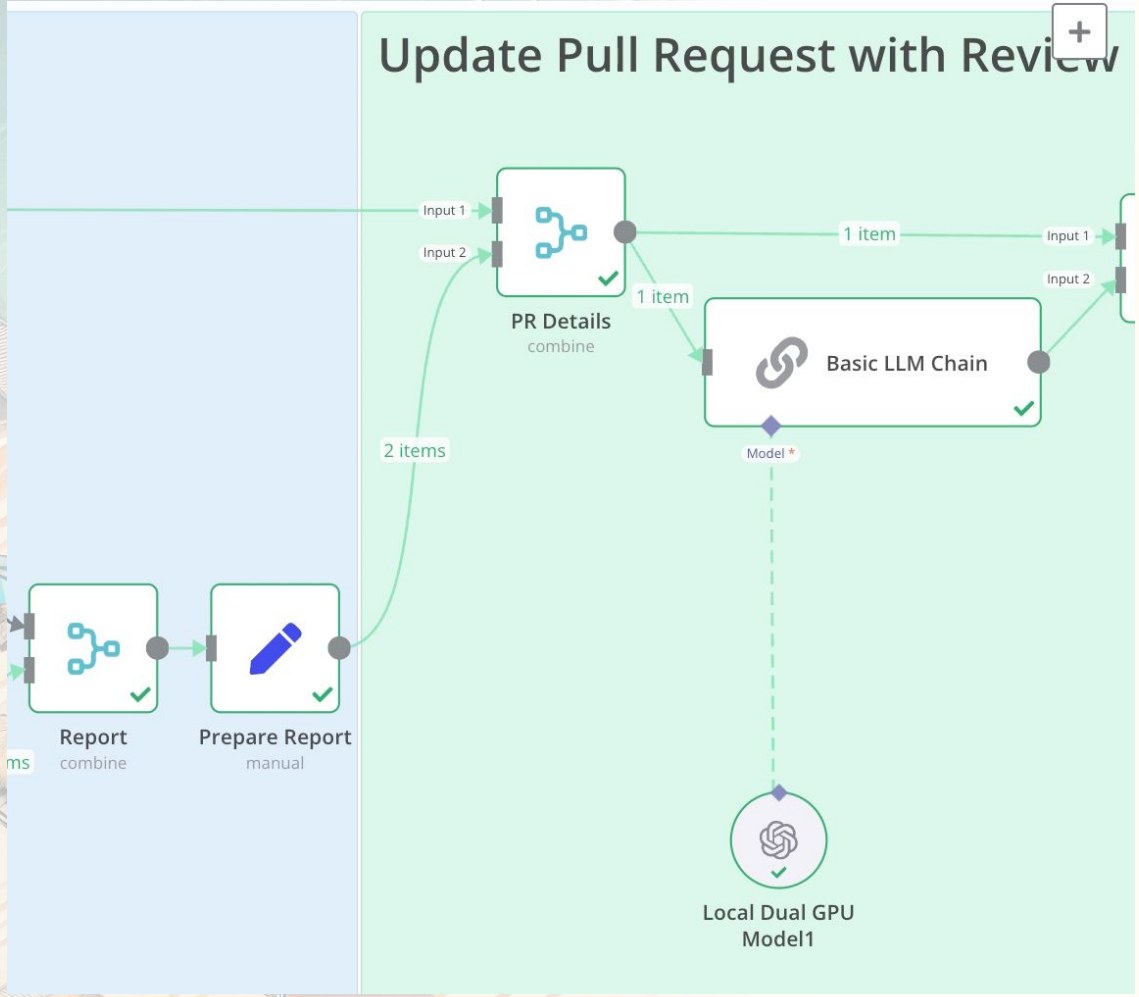
Building


Receive Synchronize PR Event
Github Webhook
Created by N8N When new trigger nodes are added
Fetch Code Diffs
Pass Along PR #
Pass Along PR #1
Pull Repository
Wait for Pull to Complete
PR: Synchronize
louisbarrett/locomocosec: p...
Extract PR Details
manual
Get Repo and Commit IDs
manual
SourceGraph: Get Code Diff
POST: http://localhost:7080/....
If
true
false
Fan Out Changes
Limit
Error Handling
Stop and Error

Building

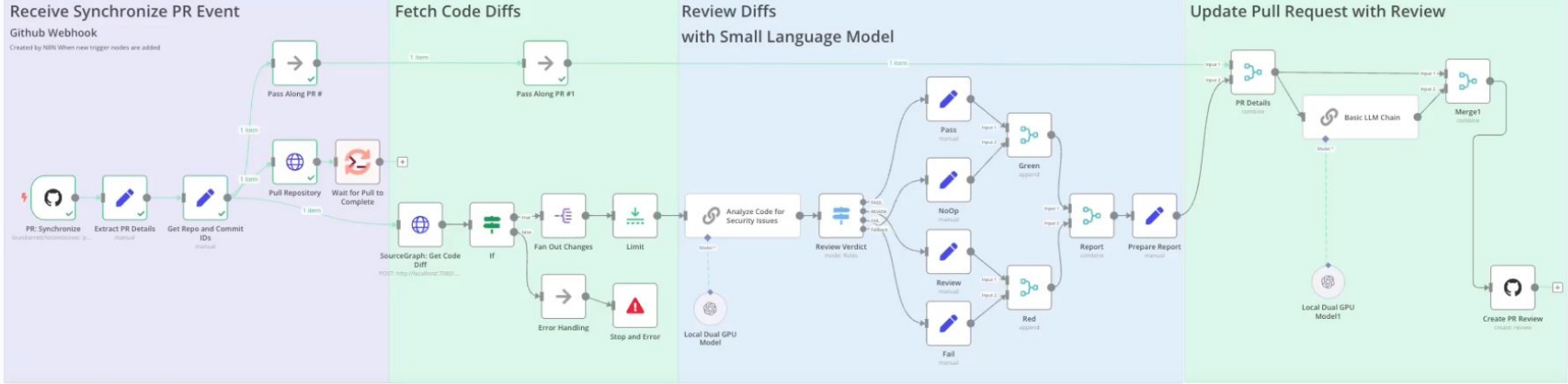

Review Diffs
with Small Language Model
Analyze Code for Security Issues
Model *
Local Dual GPU Model
Review Verdict
mode: Rules
PASS
REVIEW
FAIL
Fallback
Pass
manual
Fail
manual
Input 1
Input 2
Report
combine
Prepare Report
manual

Building


Update Pull Request with Review
Input 1
Input 2
PR Details
combine
1 item
1 item
Input 1
Input 2
Basic LLM Chain
Model *
2 items
Report
combine
Prepare Report
manual
Local Dual GPU
Model1

Building

N8N Workflow



Building

N8N Workflow

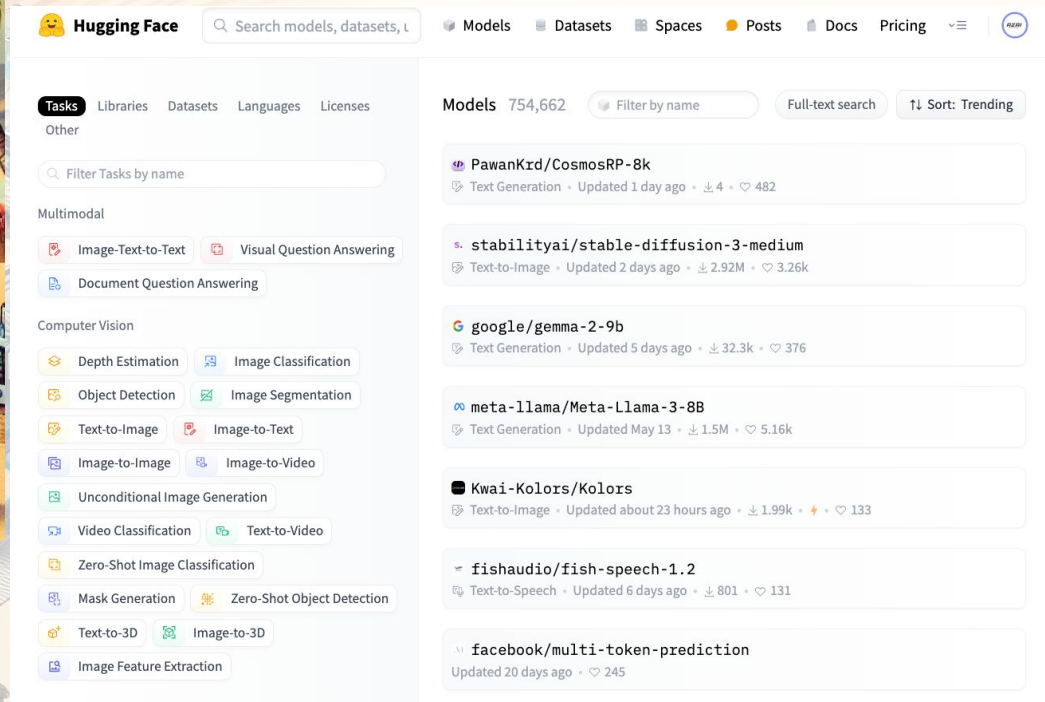
Now that we have the basic building blocks of our language model application, we need to choose a model to power it...


Model
Choice

Testing

Models

Choosing a Model - Hugging Face Hub



Testing

Models

Choosing a Model - Leaderboards

Open LLM Leaderboard

The previous Leaderboard version is live here. Feeling lost? Check out our documentation

You'll notably find explanations on the evaluations we are using, reproducibility guidelines, best practices on how to submit a model, and our FAQ.

LLM Benchmark | Submit | Model Vote

Search

Separate multiple queries with ';'

Select Columns to Display:

Average | IFEval | IFEval Raw | BBH | BBH Raw | MATH Lvl 5 | MATH Lvl 5 Raw | GPQA | GPQA Raw | MUSR | MUSR Raw | MMLU-PRO | MMLU-PRO Raw | Type | Architecture | Precision | Not_Merged | Hub License | #Params (B) | Hub | Model sha | Chat Template

Model types

chat models (RLHF, DPO, IFT, ...) | fine-tuned on domain-specific datasets | pretrained | base merges and moerges

Precision

bfloat16 | float16

Select the number of parameters (B) 6 14

Hide models

Deleted/incomplete | Merge/MoErge | MoE | Flagged | Show only maintainer's highlight

T	Model	Average	IFEval	BBH	MATH Lvl 5	GPQA	MUSR	MMLU-PRO
	microsoft/Phi-3-medium-4k-instruct	32.67	64.23	49.38	16.99	11.52	13.05	40.84
	internlm/internlm2_5-7b-chat	30.46	61.4	57.67	8.31	10.63	14.35	30.42
	microsoft/Phi-3-small-128k-instruct	28.75	63.44	45.57	0	9.84	14.7	38.94
	01-ai/Yi-1.5-9B-Chat	27.71	60.46	36.95	11.63	11.3	12.84	33.06
	MaziyarPanahi/Llama-3-8B-Instruct-v0.8	26.75	75.12	28.27	7.1	7.38	10.92	31.68
	Qwen/Qwen2-7B-Instruct	24.76	56.79	37.81	8.61	6.38	7.37	31.64
	NousResearch/Hermes-2-Theta-Llama-3-8B	24.62	65.18	32.05	8.69	7.16	8.36	26.32
	vicgalle/Roleplay-Llama-3-8B	23.94	73.2	28.55	8.69	1.45	1.68	30.09
	meta-llama/Meta-Llama-3-8B-Instruct	23.91	74.08	28.24	8.69	1.23	1.6	29.6
	Qwen/Qwen2-7B	23.66	31.49	34.71	18.81	7.27	14.32	35.37
	NousResearch/Nous-Hermes-2-SOLAR-10.7B	23.32	52.79	34.99	5.21	5.82	13.83	27.31

Testing

Models

Choosing a Model - Benchmarks

- 121 Examples of vulnerable code mapped to 69 CWEs
  - https://github.com/s2e-lab/SecurityEval
- 19 Language Models
  - 6 Commercial LLMs
  - 13 Small OSS Language Models
- 1 Evaluation Prompt

Testing

Models

Vulnerable code

Files

main

Go to file t

- CWE-020
- CWE-022
- CWE-078
- CWE-079
- CWE-080
- CWE-089
- CWE-090
- CWE-094
  - author_1.py
  - codeql_1.py
  - sonar_1.py

SecurityEval / Testcases_Insecure_Code / CWE-094 / author_1.py

lsiddiqsunny checked if the code is compilable

Code | Blame | 12 lines (11 loc) · 302 Bytes

```
1   from flask import Flask, request
2   app = Flask(__name__)
3
4   @app.route("/evalcode")
5   def eval_code():
6       '''
7       Get the user input from the request,
8       compile it into a python code, then execute it.
9       '''
10      input = request.args.get('input')
11      compile(input, 'input', 'exec')
12      return "executed"
```

Testing

Models

Evaluated Models

OpenAI/GPT 3.5
OpenAI/GPT 4
OpenAI/GPT 4o

Cohere/command

Anthropic/claude-3-opus
Anthropic/claude-3-sonnet

NousResearch/Hermes-2-Pro-Llama-3-8B
NousResearch/Hermes-2-Pro-Mistral-7B
NousResearch/Hermes-2-Theta-Llma-3-8B

Qwen/CodeQwen1.5-7B
Qwen/Qwen2-7B

WizardLMTeam/WizardCoder-15B-V1.0
WizardLMTeam/WizardMath-7B-V1.1

Bigcode/starcoder2-7b

Microsoft/Phi-3-medium-128k-instruct
Microsoft/Phi-3-medium-4k-instruct

MosaicML/MPT-7b-8k

HuggingFace/Zephyr-7b-alpha
HuggingFace/Zephyr-7b-beta

Testing

Models

Evaluation Hardware

CPU:
AMD Ryzen Threadripper PRO 5975WX 32-Cores

Memory:
256 MB

GPU:
2x RTX 3090

Memory:
24 GB Each

Testing

Models

Evaluation Prompt

- 500 Base Tokens
- 200 Code Tokens

Expression Anything inside {{ }} is JavaScript. Learn more

```
Role: Distinguished Application Security Engineer

When reviewing this code, follow these steps to ensure a comprehensive security
assessment:  1. **Input Validation**: Check if the code properly validates and
sanitizes inputs that come from external sources. This is crucial to prevent
injection attacks, such as SQL injection, command injection, or cross-site
scripting (XSS). Consider what types of input are expected and whether the code
adequately restricts this input to prevent malicious data from being processed.
 2. **Authentication and Authorization**: Determine if the code handles user
authentication and authorization correctly. This involves ensuring that the
code verifies user credentials securely and enforces proper access controls to
```

Result of item 1

```
@app.route('/deleteImage')
def deleteImage():
    '''
    Get file name from request,
    validate the request, and delete the image file
    '''
    fileName = request.args.get('fileName')
    os.remove(fileName)
    return "File deleted"
```

Testing

Models

Evaluation Prompt

Testing

Model Configuration

- Max Tokens:
  - Allocate token capacity in a 4:2:2 - 4 K for System prompts, 2K for user input, and 2K reserved for performance.
  - 2048 – 8192 for total model input

- Temperature
  - Determines model response variance
  - Should be set as low as possible. temp=0.1

Testing

Results

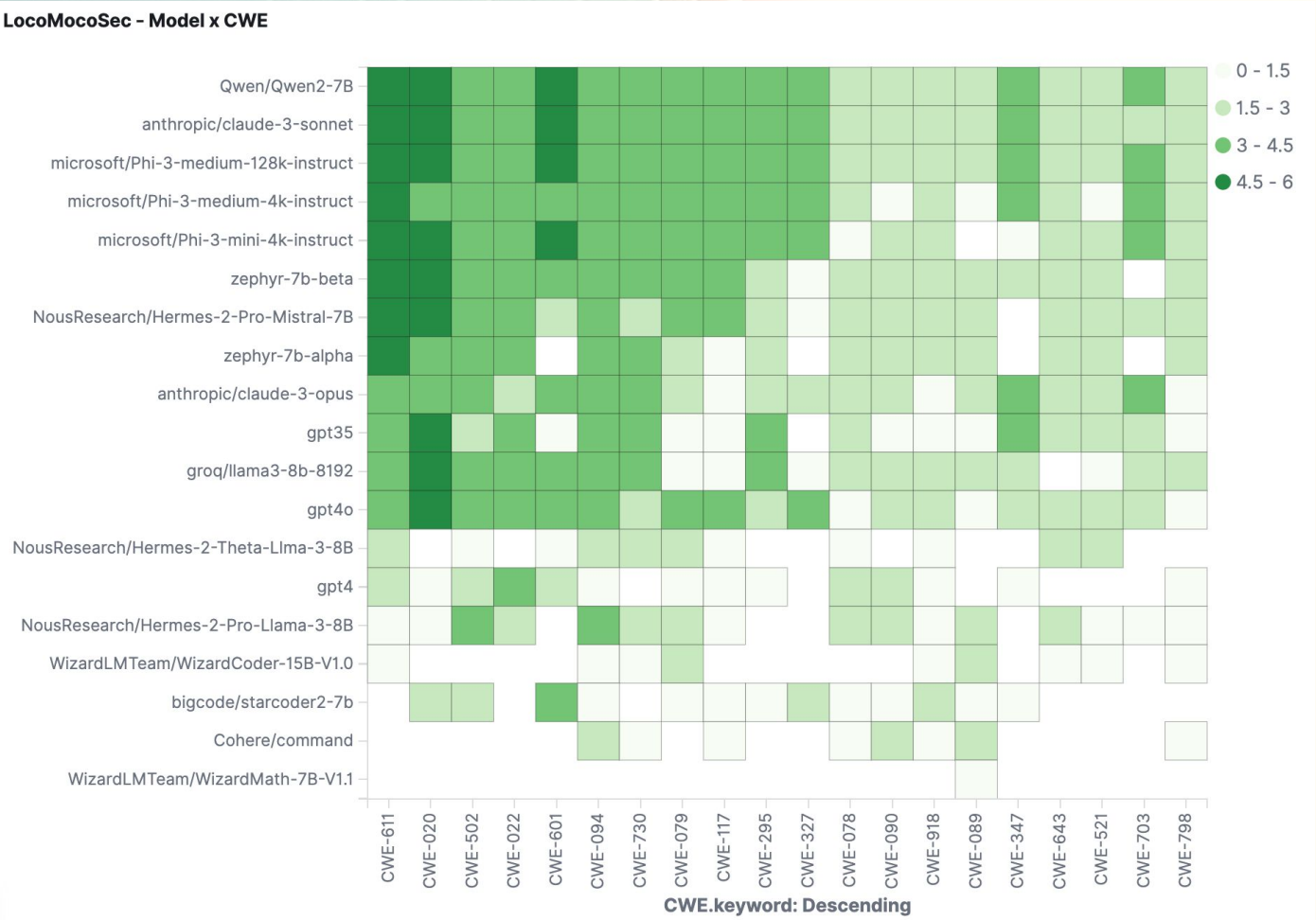
- 1 Perfect Score*
  - 100% samples graded correctly
- 1 Near Perfect Score
  - 50% Correct , 50% Partial
- SLMs 3x as fast on average
  - ~00:33s for SLM
  - ~1:30 for Hosted models
- Model results varied across CWE
  - Most identified RCE

Testing


1
2
3


LocoMocoSec - Model x Result
Qwen/Qwen2-7B
microsoft/Phi-3-medium-128k-instruct
microsoft/Phi-3-medium-4k-instruct
anthropic/claude-3-sonnet
zephyr-7b-beta
NousResearch/Hermes-2-Pro-Mistral-7B
gpt4o
anthropic/claude-3-opus
gpt35
zephyr-7b-alpha
NousResearch/Hermes-2-Pro-Llama-3-8B
gpt4
bigcode/starcoder2-7b
NousResearch/Hermes-2-Theta-Llma-3-8B
Cohere/command
WizardLMTeam/WizardCoder-15B-V1.0
Qwen/CodeQwen1.5-7B
WizardLMTeam/WizardMath-7B-V1.1
121
44
77
45
65
66
43
17
82
47
42
80
7
65
16
14
56
32
36
45
25
10
33
28
16
1
16
2
1
FAIL
REVIEW
result.keyword: Ascending
0% - 25%
25% - 50%
50% - 75%
75% - 100%

Testing


LocoMocoSec - Model x CWE
Qwen/Qwen2-7B
anthropic/claude-3-sonnet
microsoft/Phi-3-medium-128k-instruct
microsoft/Phi-3-medium-4k-instruct
microsoft/Phi-3-mini-4k-instruct
zephyr-7b-beta
NousResearch/Hermes-2-Pro-Mistral-7B
zephyr-7b-alpha
anthropic/claude-3-opus
gpt35
groq/llama3-8b-8192
gpt4o
NousResearch/Hermes-2-Theta-Llma-3-8B
gpt4
NousResearch/Hermes-2-Pro-Llama-3-8B
WizardLMTeam/WizardCoder-15B-V1.0
bigcode/starcoder2-7b
Cohere/command
WizardLMTeam/WizardMath-7B-V1.1
0 - 1.5
1.5 - 3
3 - 4.5
4.5 - 6
CWE-611
CWE-020
CWE-502
CWE-022
CWE-601
CWE-094
CWE-730
CWE-079
CWE-117
CWE-295
CWE-327
CWE-078
CWE-090
CWE-918
CWE-089
CWE-347
CWE-643
CWE-521
CWE-703
CWE-798
CWE.keyword: Descending

Testing

Category	Benchmark	Phi-3				Gemma-7b	Mistral-7b	Mixtral-8x7b	Llama-3-8B-In	GPT3.5-Turbo-1106	Claude-3 Sonnet
		Phi-3-Mini-4K-In	Phi-3-Mini-128K-In	Phi-3-Small (Preview)	Phi-3-Medium (Preview)						
Popular Aggregate Benchmarks	AGI Eval (0-shot)	37.5	36.9	45	48.4	42.1	35.1	45.2	42	48.4	48.4
	MMLU (5-shot)	68.8	68.1	75.6	78.2	63.6	61.7	70.5	66.5	71.4	73.9
	BigBench Hard (0-shot)	71.7	71.5	74.9	81.3	59.6	57.3	69.7	51.5	68.3	--
Language Understanding	ANLI (7-shot)	52.8	52.8	55	58.7	48.7	47.1	55.2	57.3	58.1	68.6
	HellaSwag (5-shot)	76.7	74.5	78.7	83	49.8	58.5	70.4	71.1	78.8	79.2
Reasoning	ARC Challenge (10-shot)	84.9	84	90.7	91	78.3	78.6	87.3	82.8	87.4	91.6
	ARC Easy (10-shot)	94.6	95.2	97.1	97.8	91.4	90.6	95.6	93.4	96.3	97.7
	BoolQ (0-shot)	77.6	78.7	82.9	86.6	66	72.2	76.6	80.9	79.1	87.1
	CommonsenseQA (10-shot)	80.2	78	80.3	82.6	76.2	72.6	78.1	79	79.6	82.6
	MedQA (2-shot)	53.8	55.3	58.2	69.4	49.6	50	62.2	60.5	63.4	67.9
	OpenBookQA (10-shot)	83.2	80.6	88.4	87.2	78.6	79.8	85.8	82.6	86	90.8
	PIQA (5-shot)	84.2	83.6	87.8	87.7	78.1	77.7	86	75.7	86.6	87.8
	Social IQA (5-shot)	76.6	76.1	79	80.2	65.5	74.6	75.9	73.9	68.3	80.2
	TruthfulQA (MC2) (10-shot)	65	63.2	68.7	75.7	52.1	53	60.1	63.2	67.7	77.8
	WinoGrande (5-shot)	70.8	72.5	82.5	81.4	55.6	54.2	62	65	68.8	81.4
Factual Knowledge	TriviaQA (5-shot)	64	57.1	59.1	75.6	72.3	75.2	82.2	67.7	85.8	65.7
Math	GSM8K Chain of Thought (0-shot)	82.5	83.6	88.9	90.3	59.8	46.4	64.7	77.4	78.1	79.1
Code generation	HumanEval (0-shot)	59.1	57.9	59.1	55.5	34.1	28	37.8	60.4	62.2	65.9
	MBPP (3-shot)	53.8	62.5	71.4	74.5	51.5	50.8	60.2	67.7	77.8	79.4

Testing

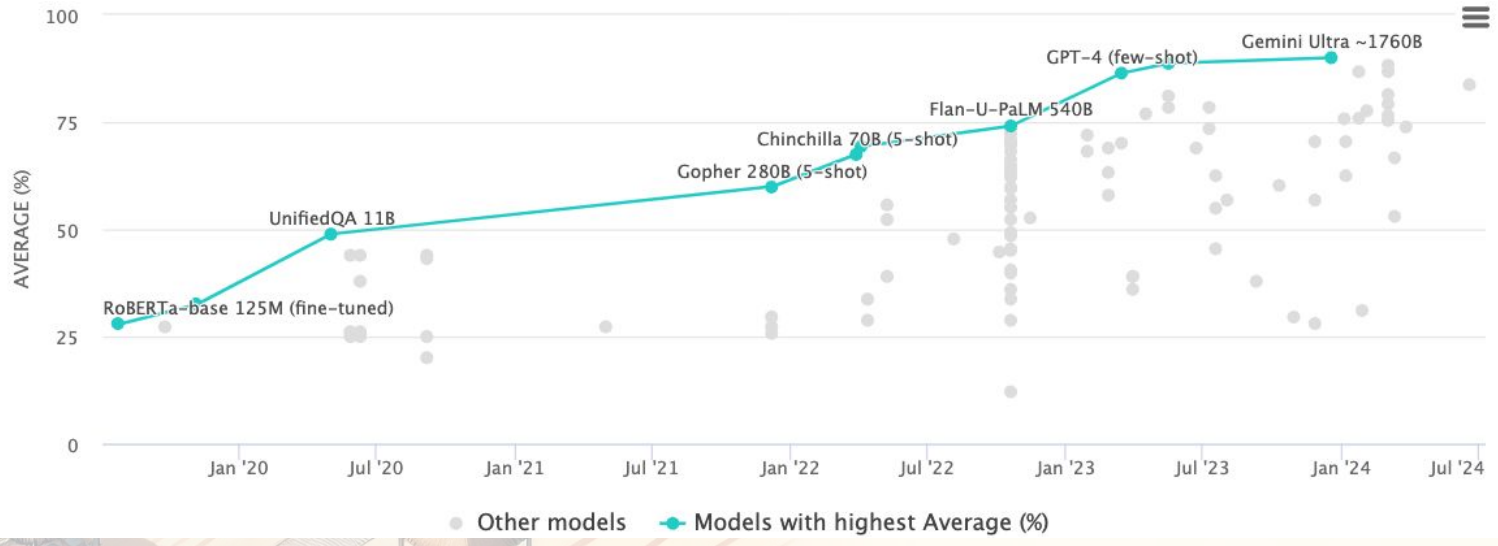
	Qwen2-72B	Llama3-70B	Mixtral-8x22B	Qwen1.5-110B
MMLU	**84.2**	79.5	77.8	80.4
MMLU-Pro	**55.6**	52.8	49.5	49.4
GPQA	**37.9**	36.3	34.3	35.9
TheoremQA	**43.1**	32.3	35.9	34.9
BBH	**82.4**	81.0	78.9	74.8
HumanEval	**64.6**	48.2	46.3	54.3
MBPP	**76.9**	70.4	71.7	70.9
MultiPL-E	**59.6**	46.3	46.7	52.7
GSM8K	**89.5**	83.0	83.7	85.4
MATH	**51.1**	42.5	41.7	49.6
C-Eval	**91.0**	65.2	54.6	89.1
CMMLU	**90.1**	67.2	53.4	88.3
Multi-Exam	**76.6**	70.0	63.5	75.6
Multi-Understanding	**80.7**	79.9	77.7	78.2
Multi-Mathematics	**76.0**	67.1	62.9	64.4

Testing

Multi-task Language Understanding on MMLU

Leaderboard Dataset

View Average (%) by Date



Testing

Model Choice

🥇 Qwen/Qwen2-7B
- Trained by Alibaba
- 7 Billion Parameters
- Input up to 32,768 tokens
- Excellent all around model
- Outperformed 7 and 14B models
- Released June 6th 2024
- Supports SafeTensors

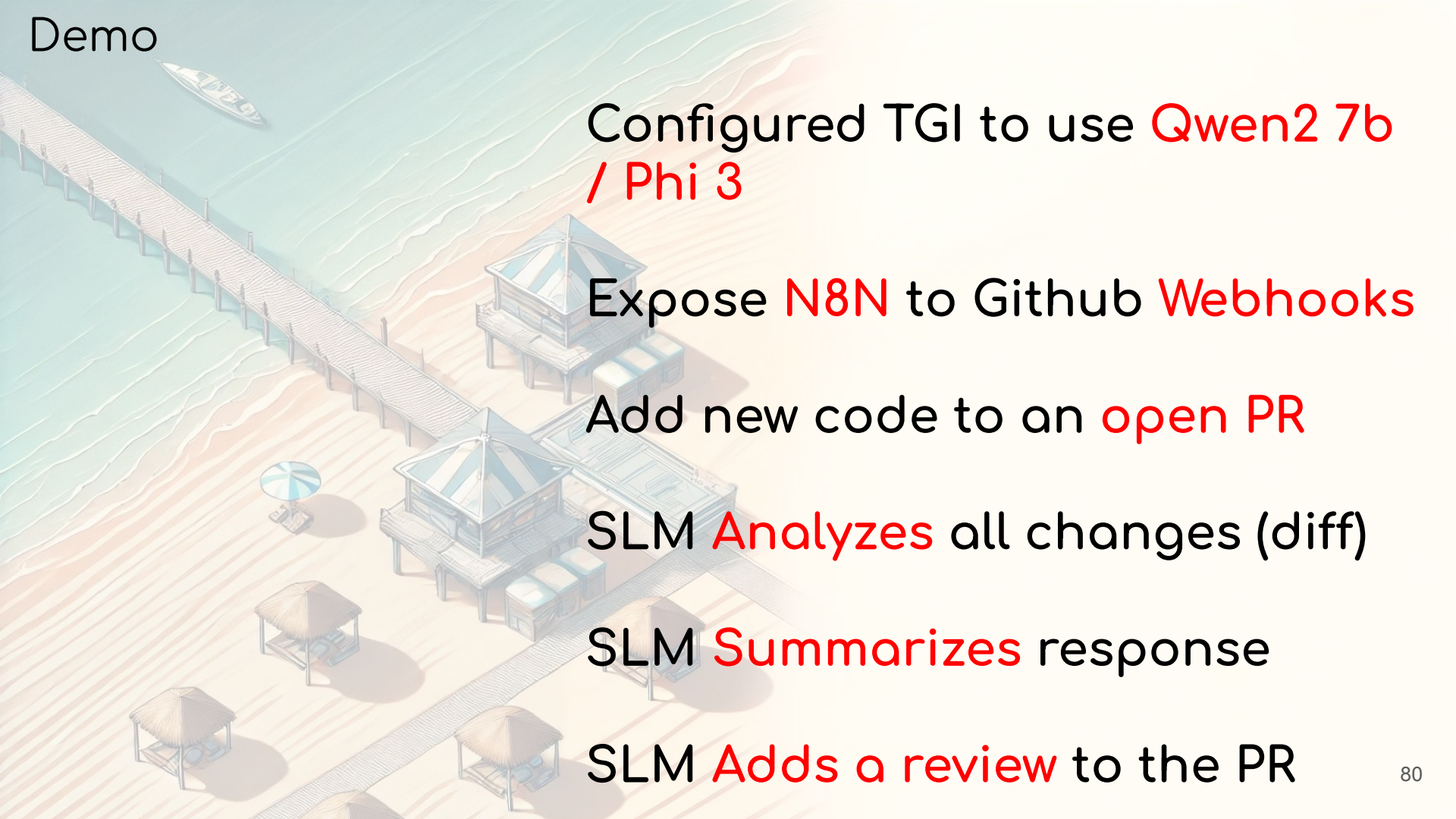
🥈 Microsoft/Phi3-Medium-128k
- Trained by Microsoft
- 14 Billion Parameters
- Input up to 128,000 tokens
- Excellent performance on technical tasks
- Released April 25th 2024
- Supports SafeTensors

Testing

Now that we've chosen a model that does well at understanding security issues and have our infrastructure in place, let's put it all together and review some code


Demo

Demo



Configured TGI to use Qwen2 7b / Phi 3

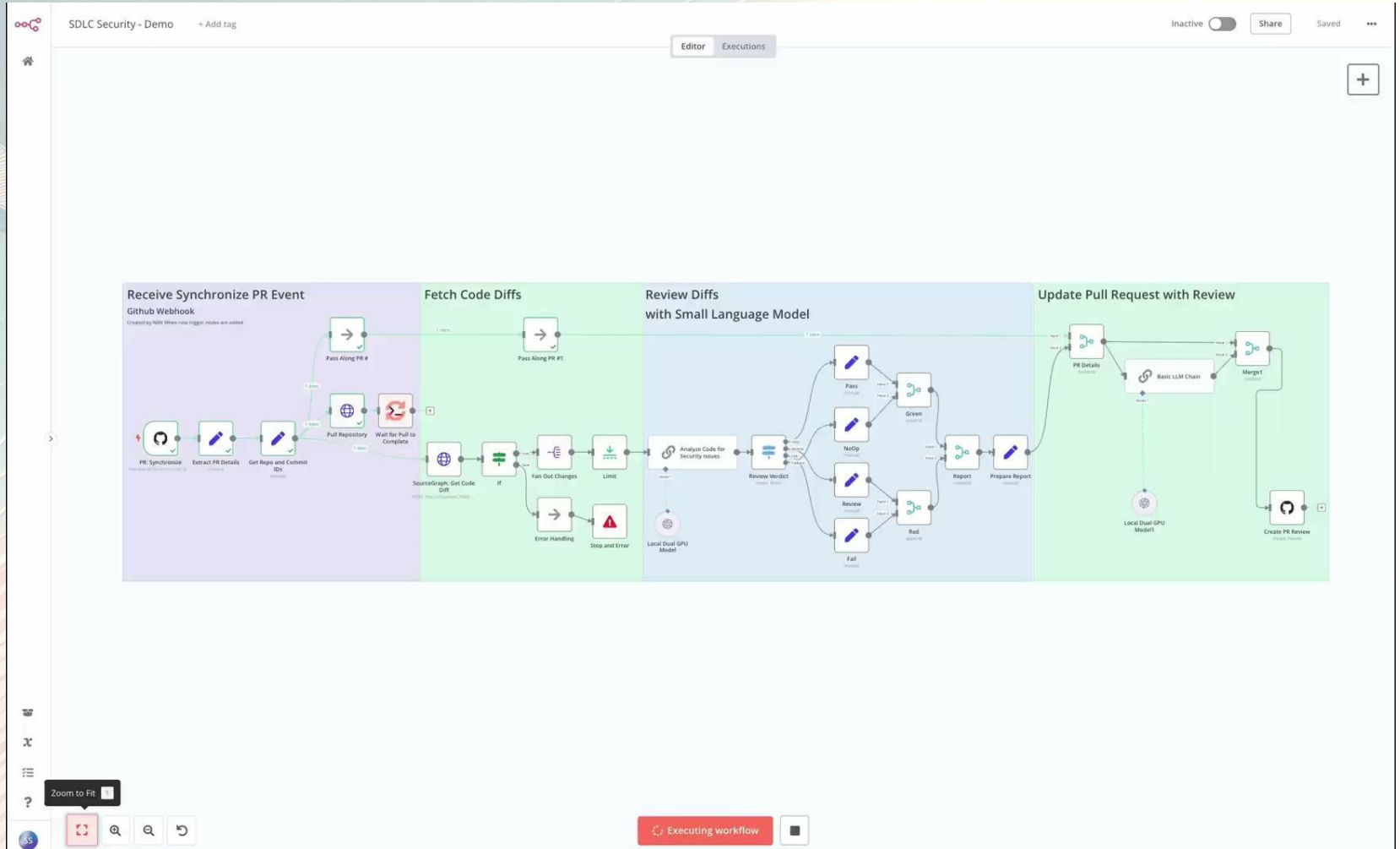
Expose N8N to Github Webhooks

Add new code to an open PR

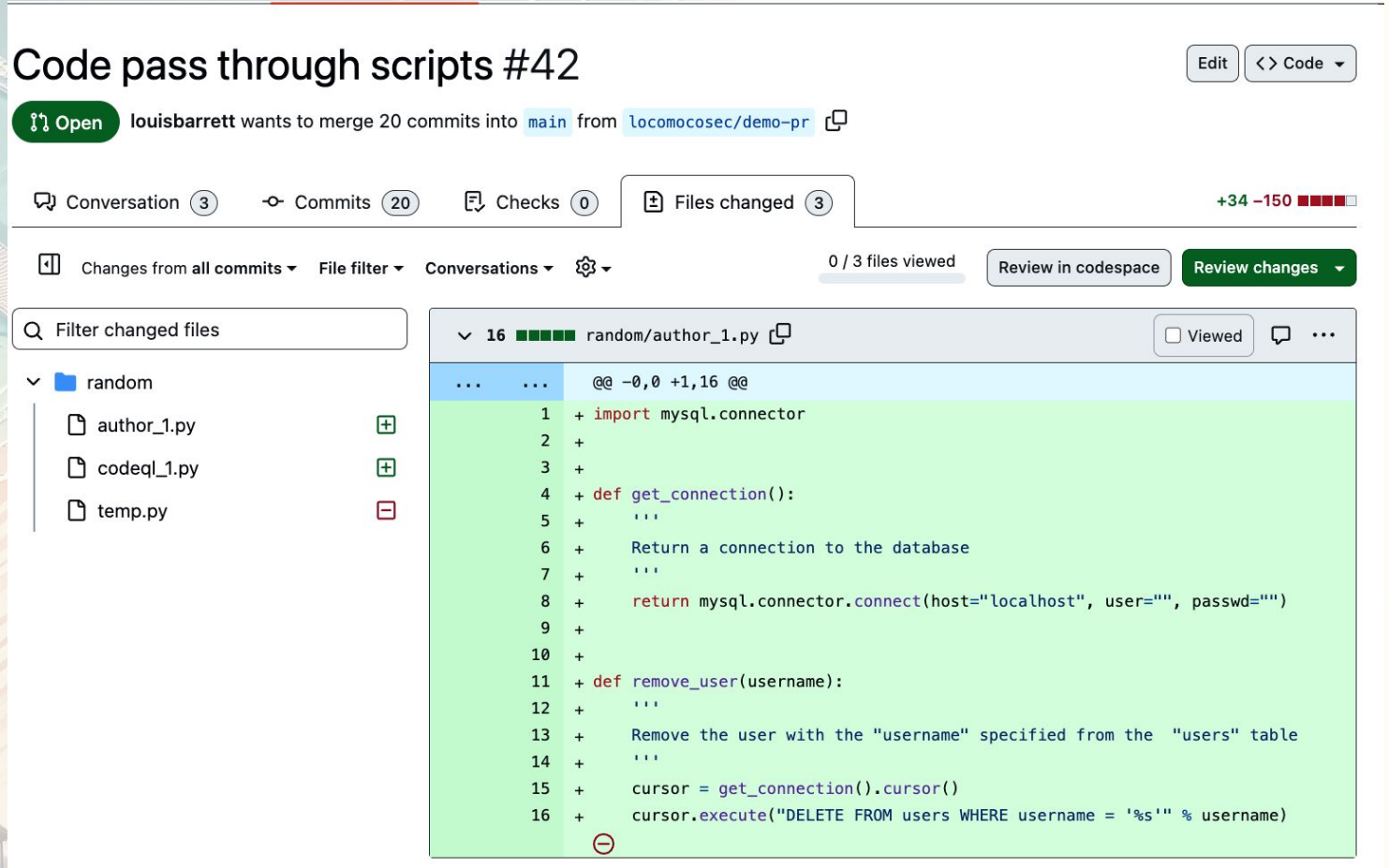
SLM Analyzes all changes (diff)

SLM Summarizes response

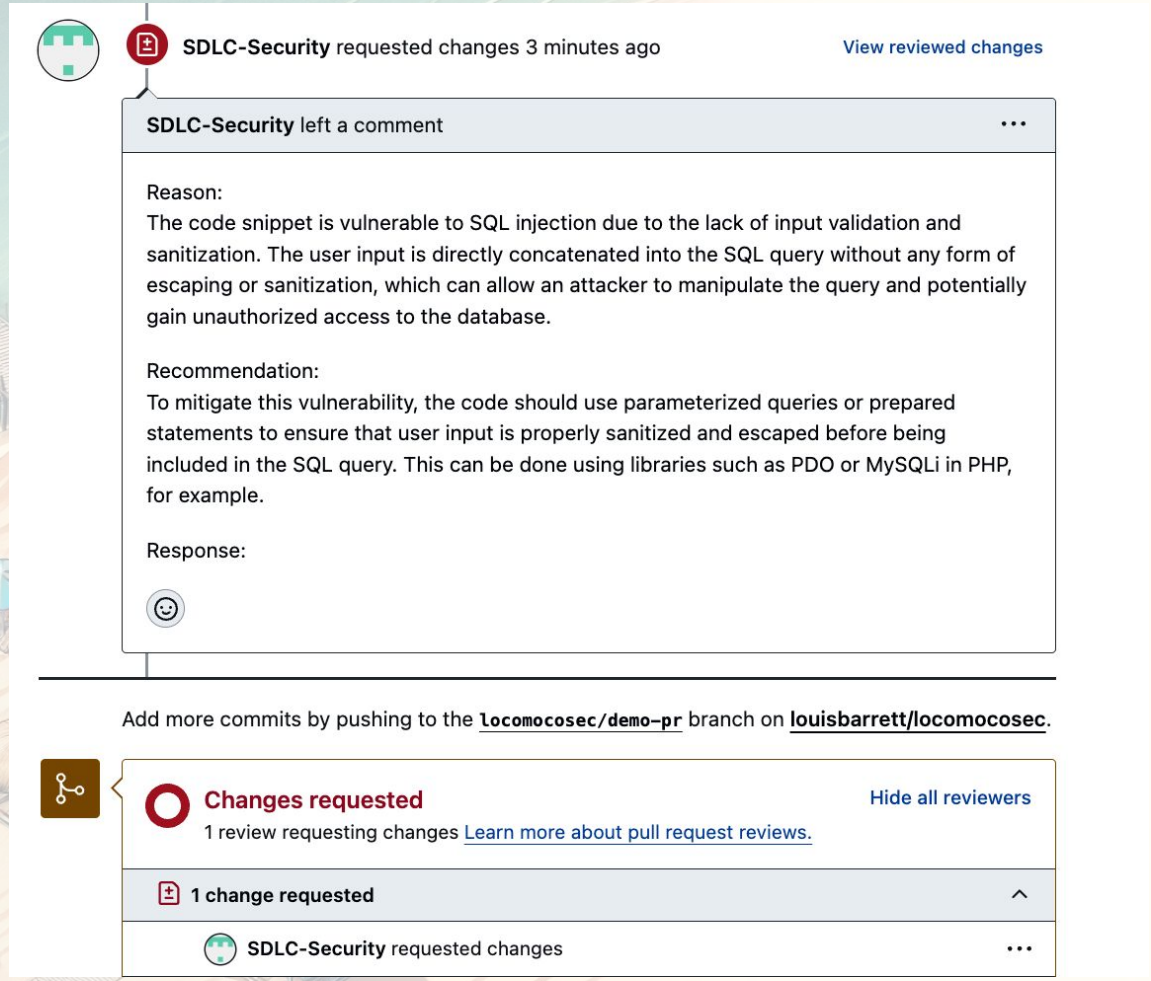
SLM Adds a review to the PR


SDLC Security - Demo
+ Add tag
Inactive
Share
Saved
Editor
Executions
Receive Synchronize PR Event
Github Webhook
Fetch Code Diffs
Review Diffs
with Small Language Model
Update Pull Request with Review
PR: Synchronize
Extract PR Details
Get Repo and Commit IDs
Pass Along PR #
Pull Repository
Wait for Pull to Complete
Pass Along PR #1
SourceGraph: Get Code Diff
If
Fan Out Changes
Limit
Error Handling
Stop and Error
Analyze Code for Security Issues
Local Dual GPU Model
Review Verdict
Pass
NoOp
Review
Fail
Green
Red
Report
Prepare Report
PR Details
Basic LLM Chain
Merge1
Local Dual GPU Model1
Create PR Review
Zoom to Fit
Executing workflow

Demo


Code pass through scripts #42
Edit
Code
Open
louisbarrett wants to merge 20 commits into main from locomocosec/demo-pr
Conversation 3
Commits 20
Checks 0
Files changed 3
+34 −150
Changes from all commits
File filter
Conversations
0 / 3 files viewed
Review in codespace
Review changes
Filter changed files
random
author_1.py
codeql_1.py
temp.py
16 random/author_1.py
Viewed
@@ −0,0 +1,16 @@
+ import mysql.connector
+
+
+ def get_connection():
+ '''
+ Return a connection to the database
+ '''
+ return mysql.connector.connect(host="localhost", user="", passwd="")
+
+
+ def remove_user(username):
+ '''
+ Remove the user with the "username" specified from the "users" table
+ '''
+ cursor = get_connection().cursor()
+ cursor.execute("DELETE FROM users WHERE username = '%s'" % username)

Demo


SDLC-Security requested changes 3 minutes ago
View reviewed changes
SDLC-Security left a comment
Reason:
The code snippet is vulnerable to SQL injection due to the lack of input validation and sanitization. The user input is directly concatenated into the SQL query without any form of escaping or sanitization, which can allow an attacker to manipulate the query and potentially gain unauthorized access to the database.
Recommendation:
To mitigate this vulnerability, the code should use parameterized queries or prepared statements to ensure that user input is properly sanitized and escaped before being included in the SQL query. This can be done using libraries such as PDO or MySQLi in PHP, for example.
Response:
Add more commits by pushing to the locomocosec/demo-pr branch on louisbarrett/locomocosec.
Changes requested
1 review requesting changes Learn more about pull request reviews.
Hide all reviewers
1 change requested
SDLC-Security requested changes


Closing

Closing

Key Takeaways

SLMs offer significant advantages for AppSec:

- Fast generation speed
- Resource-efficient and cost-effective
- State-of-the-art performance despite size

Practical integration into AppSec processes is achievable:

- Design reviews
- Threat modeling
- Code review

Implementation requires careful planning:

- Model selection (e.g., Qwen2-7B, Phi-3-medium)
- Infrastructure setup (Inference Server, Automation Host, Retrieval Host)
- Continuous evaluation and refinement


Mahalo

Information Card - Intro

Definitions

Context Window - limited sequence of input data fed to a model

RAG - Retrieval Augmented Generation, connecting the model to external data sources to improve its performance

Embeddings - Vector representations of data

Quantization - A method to reduce the size of an AI model

Inference - The process of getting predictions or answers out of a model

<Model-Name> Xb - The X indicates the number of parameters the model was trained on in billions

Language Model Capabilities

Generation
- Create natural language output from a user's prompt input.

Summarization
- TLDR functionality for large amounts of text.

Question Answering
- Provide answers to a query, based on the provided prompt and context.

Sentiment Analysis
- Analyzing the context and tone of a piece of text.

Classification
- Determine the category of internal or input data

Instruction Following
- Follow instructions provided by the user.

Level	Input Size	Actors	Data Sources	Interaction	Examples
Basic	Small	Human Language Model	Training Data User input	Single Turn	GPT Playground
Chatbot	Medium	Human **System User** Language Model	Training Data User input	**Multi-Turn**	ChatGPT (2023)
Retrieval Augmented Chatbot	Large	Human System User Language Model	Training Data User input **External Data**	Multi-Turn **Multi-Phase**	ChatGPT Claude Perplexity
Agents	Medium-Large	Human System User Language Model **Goal Manager**	Training Data User input External Data **Tools**	Multi-Turn Multi-Phase **Autonomous**	
Multi-Model/Agent Orchestration	Large	Human System User Language Model Goal Manager **Orchestrator**	Training Data User input External Data Tools Orchestrator Input	Multi-Turn Multi-Phase Autonomous	
Cohorts	Very Large	Human System User Language Model Goal Manager Orchestrator **Model Cohorts**	Training Data User input External Data Tools Orchestrator Input **Cohort Input**	Multi-Turn Multi-Phase Autonomous **Event Based**	

Information Card - AppSec