LM-Eval and Guardrails

Architecture

TrustyAI

LM-Eval

Experience overview

- Benchmark and evaluate language model performance on a variety of tasks
    - Using https://github.com/EleutherAI/lm-evaluation-harness
- Deploy LM-Eval Job CR
    - Specifies model and specific tasks for eval
- Metrics/benchmarks saved in specified storage

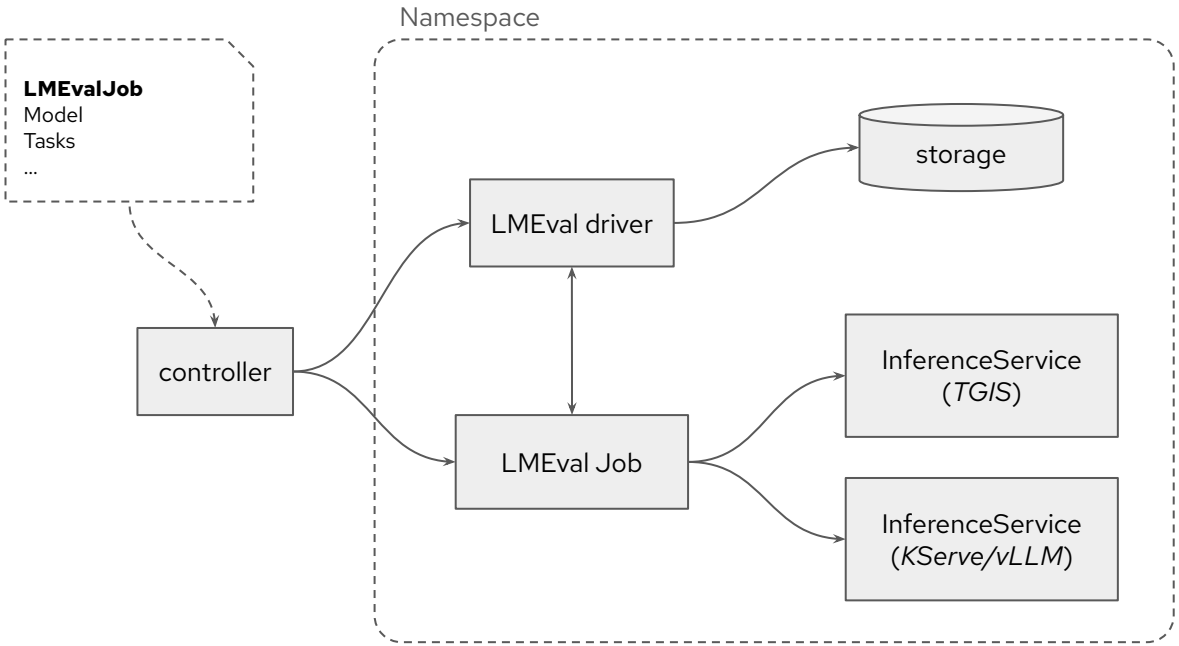
Sample of the 111 currently available tasks

logiqa	Logical reasoning tasks requiring advanced inference and deduction.
anli	Adversarial natural language inference tasks designed to test model robustness.
asdiv	Tasks involving arithmetic and mathematical reasoning challenges.
realtoxicityprompts	Tasks to evaluate language models for generating text with potential toxicity.
medqa	Multiple choice question answering based on the United States Medical License Exams
eq_bench	Tasks focused on equality and ethics in question answering and decision-making.
crows_pairs	Tasks designed to test model biases in various sociodemographic groups.

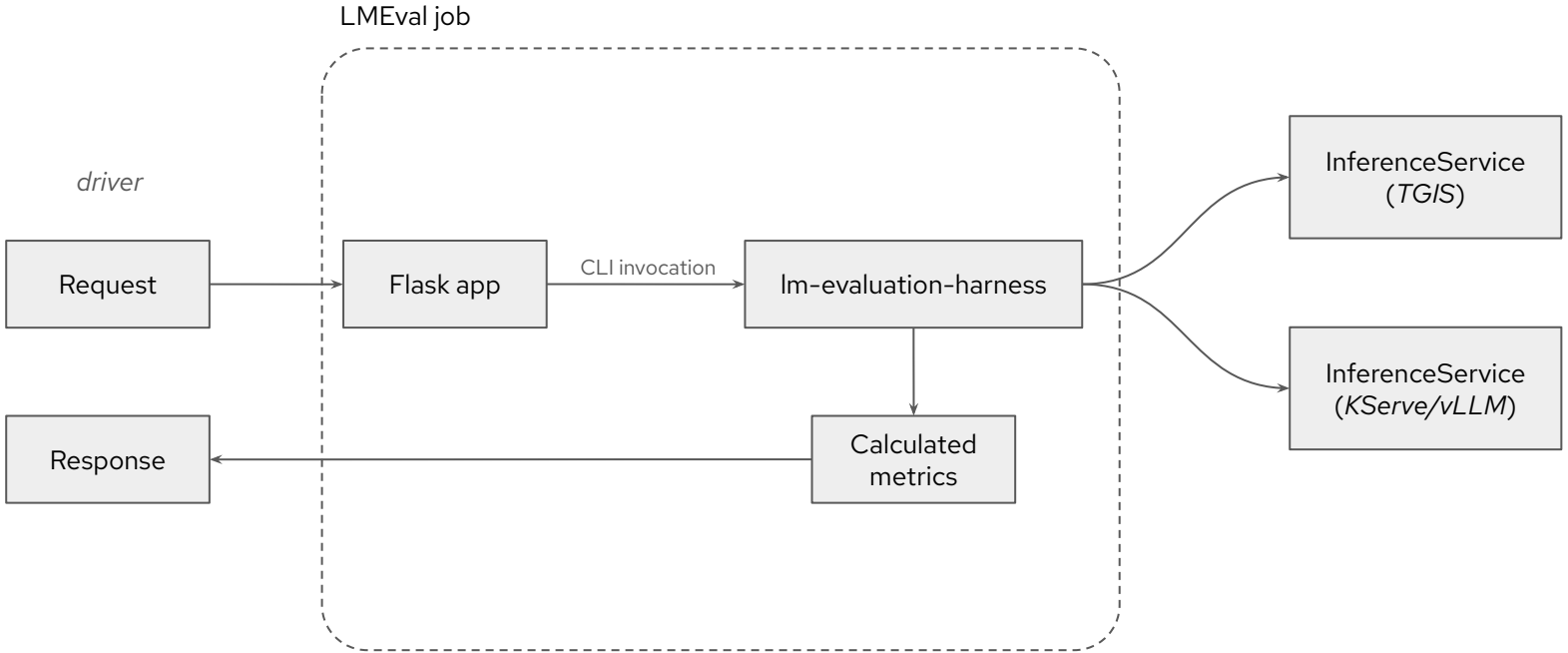
Components

- **Controller**: k8s controller to manage deployment of everything else
- **Job**: encapsulates LM-evaluation-harness, provides HTTP REST API for harness tasks
- **Driver**: translates tasks to Job's REST API, manages evaluation result storage (log stream, PVC, etc)

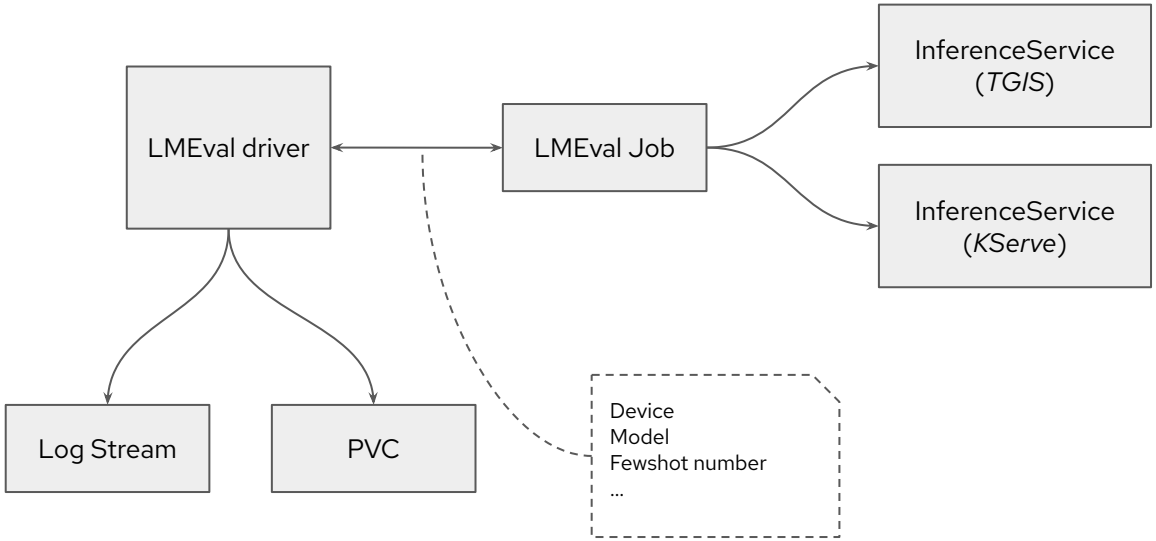
LM-Eval


Namespace
LMEvalJob
Model
Tasks
...
storage
LMEval driver
controller
InferenceService
(TGIS)
LMEval Job
InferenceService
(KServe/vLLM)

LM-Eval


LMEval job
driver
Request
Flask app
CLI invocation
lm-evaluation-harness
InferenceService
(TGIS)
InferenceService
(KServe/vLLM)
Response
Calculated
metrics

LM-Eval


LMEval driver
LMEval Job
InferenceService
(TGIS)
InferenceService
(KServe)
Log Stream
PVC
Device
Model
Fewshot number
...

Guardrails

Experience overview

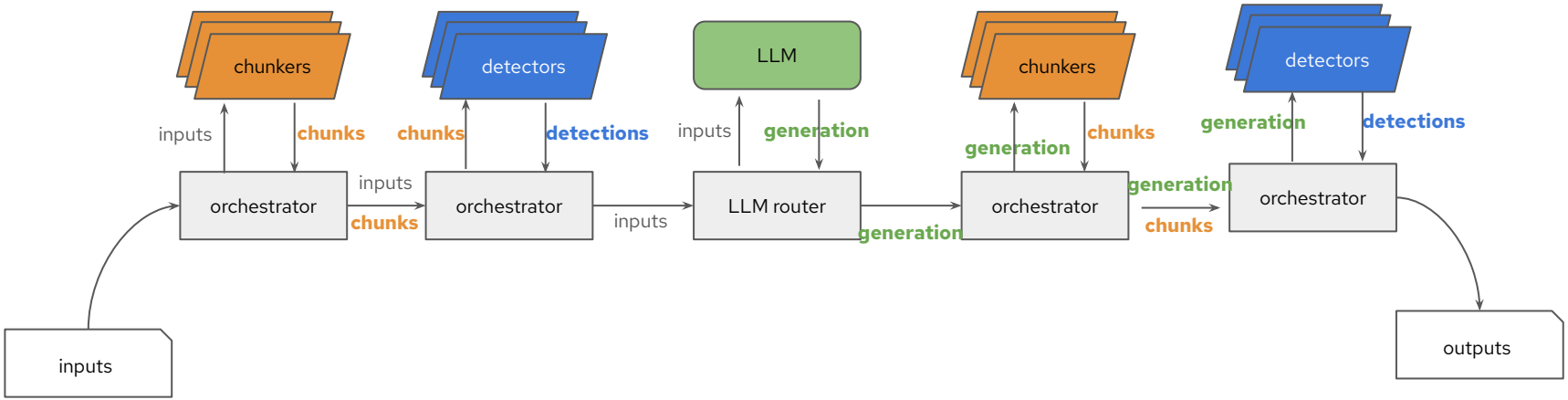
- Receive User Input
- Validate that user input is acceptable
- Pass validated input to generative model
- Validate that generated output is acceptable
- Pass validated output back to user
    - Include violation metrics

Components

- **Controller**: k8s controller to manage deployment of everything else
- **Orchestrator**: manages and implements the data flow of Guardrails
- **Chunker**(s): splits up input text into partitions, where the size, format, and partitioning scheme are determined by the detector
- **Detector**(s): identifies whether inbound or outbound text contains whatever arbitrary property is being guardrailed

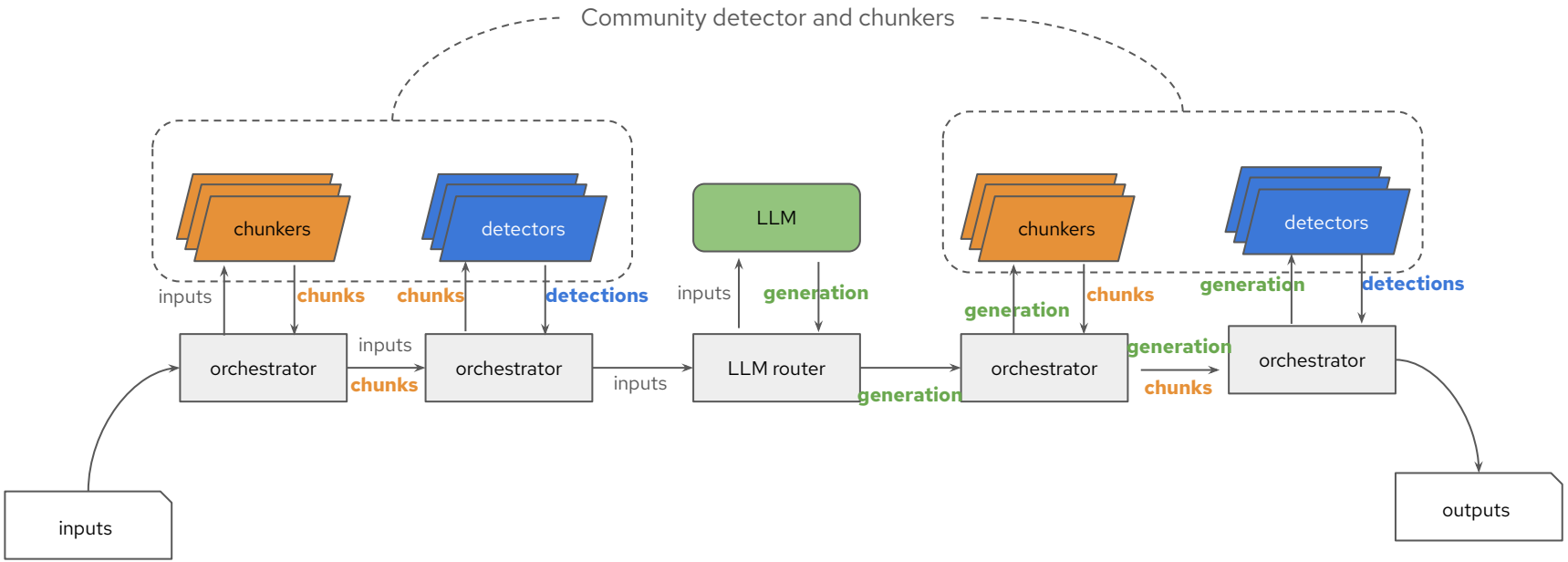
Guardrails

- Deploy **LLM**
- If **detectors** are ML models, deploy detector models into serving runtime
- If **chunkers** are ML models, deploy chunker models into serving runtime
- Apply Guardrails CR (specifies the desired detectors and chunkers)
- Use Orchestrator endpoints (instead of raw model server endpoints) to perform various guardrailed LLM tasks



TrustyAI

Guardrails


Community detector and chunkers
chunkers
detectors
LLM
chunkers
detectors
inputs
chunks
chunks
detections
inputs
generation
generation
chunks
generation
detections
orchestrator
inputs
chunks
orchestrator
inputs
LLM router
generation
orchestrator
generation
chunks
orchestrator
inputs
outputs

Resources

Resources

- LM-Eval development branch
  - https://github.com/trustyai-explainability/trustyai-service-operator/tree/dev/lm-eval
- Detectors / Chunkers
  - Coming soon to https://github.com/trustyai-explainability