PRESENTED BY
INSIGHT BUILDER

TRACING LLM INFERENCE WITH PHOENIX

Part 1: Tracing the Calls to LLMs

Modules

PHOENIX

Retrieval
Augumented
Vector Stores

WHY PHOENIX & DSPY

Phoenix provides a robust and portable way of looking inside the modules and metrics of DSPy & many other apps

DSPy Modules & Metrics can make multiple calls to different APIs, during the development & deployment. Phoenix helps to observe & analyse.

Prompt optimisation with DSPy modules & Metrics requires reviewing the optimisation flow. Phoenix makes it easier with its persistence option

WHAT TO EXPECT

OPEN TELEMETRY & OPEN INFERENCE

WHERE PHOENIX FITS

PHOENIX SETUP

DSPY INSTRUMENTATION

METRIC DEVELOPMENT

EXECUTING EVALUATION

MAKING DATASETS

WHAT ELSE WITH OPEN TELEMETRY

OPEN TELEMETRY

Mechanism to make the system “observable” by instrumenting the application code

Purpose of Open Telemetry

- Collect
- Process
- Export

What is Telemetry

- Spans
- traces
- Logs

OpenInference is complimentary to OpenTelemetry to enable tracing of AI applications.

OPEN INFERENCE

PARTS OF TELEMETRY

1) traces: Records the path taken by requests as they propagate through multi-service architectures
   - it is made of one or more spans, starting with root span
   - root span represents request from start to finish
   - child spans below show sub requests sent out to complete root span

2) logs: a timestamped message emitted which is not necessarily associated with particular request

3) spans: Represents a unit of work. It tracks a specific operation that a request makes, and shows what happened during the time the operation was executed.
   - Contains structured logs, time data, along with metadata
   - Span Attributes are the metadata attached to span

4) metrics: A measurement captured at a runtime

5) baggage: Context info passed between signals

PHOENIX
IS THE COLLECTOR

Server with API endoints to which the spans and its attributes are written

INSTRUMENT	EXPORTER	COLLECTOR	OLTP FRAMEWORK
An application to emit traces for analysis, the application must be instrumented.	exporter takes the spans created via instrumentation and exports them to a collector	Phoenix starts receiving spans form any application(s) that is exporting spans to it.	OpenTelemetetry Protocol (or OTLP for short) is the means by which traces arrive from your application to the Phoenix collector.

SETTING UP PHOENIX

WITH PERSISTENCE

0
```
pip install arize-phoenix
```

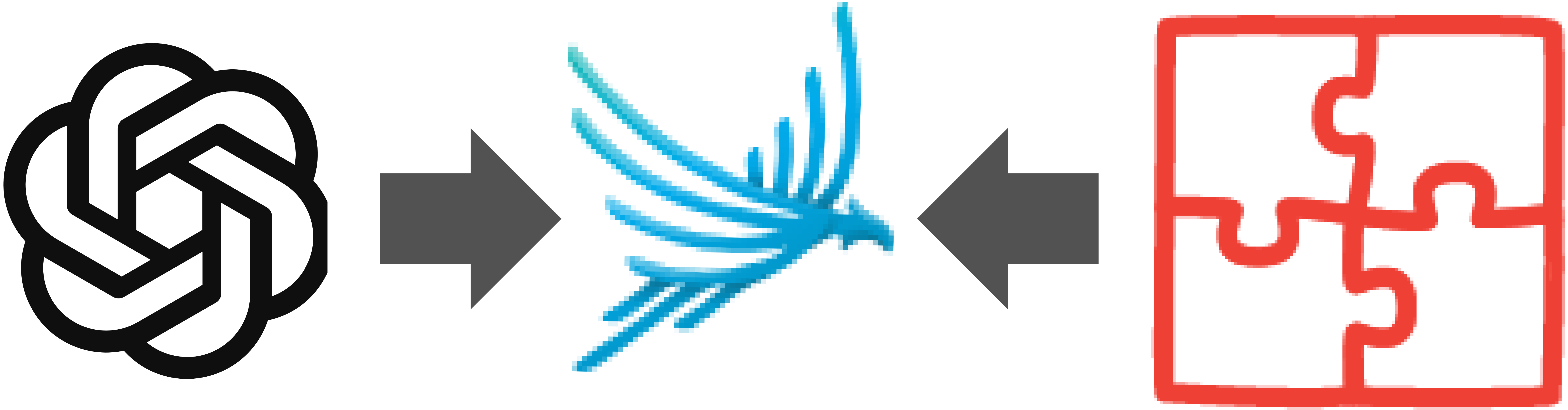
1
```
pip install psycopg2-binary asyncpg
```

2
```
export PHOENIX_SQL_DATABASE_URL
```

3
```
python -m phoenix.server.main serve
```

4 connect @ http//localhost:6006

INSTRUMENTING & COLLECTING



THANKS FOR WATCHING

♥ LIKE

SHARE

SUBSCRIBE

HTTPS://GITHUB.COM/INSIGHTBUILDER