* Pipenv helps manage both pip and virtualenv
* Create an pip env interpreter in pycharm settings
* Previously, we had only langchain single package, it includes a log of agents or the 3rd party llm connections which you won’t even use. Now, they wanted to split it and make it more simpler.
* Then pipenv install langchain
* Pipenv install langchain-community. Example (Langchain-openai)
* pipenv install huggingface\_hub
* Pipenv install langchainhub – This is for the already contributed template of great prompts.
* Pipenv install black – code formatter.
* Pipenv install python-dotenv

A screenshot of a computer screen

AI-generated content may be incorrect.

A diagram of a software development process

AI-generated content may be incorrect.

React Agent Executor

A diagram of a tool

AI-generated content may be incorrect.

## Defining custom tools

* When constructing the new agent, you need to provide it with the list of tools that It can use.
* From langchain.agents import tool, decorate your function with tool, it acts as an utility function. It takes your function and creates a lang chain class.
* The description which we have provided inside the class, makes the LLM reasoning engine to decide what to do.

A screenshot of a computer

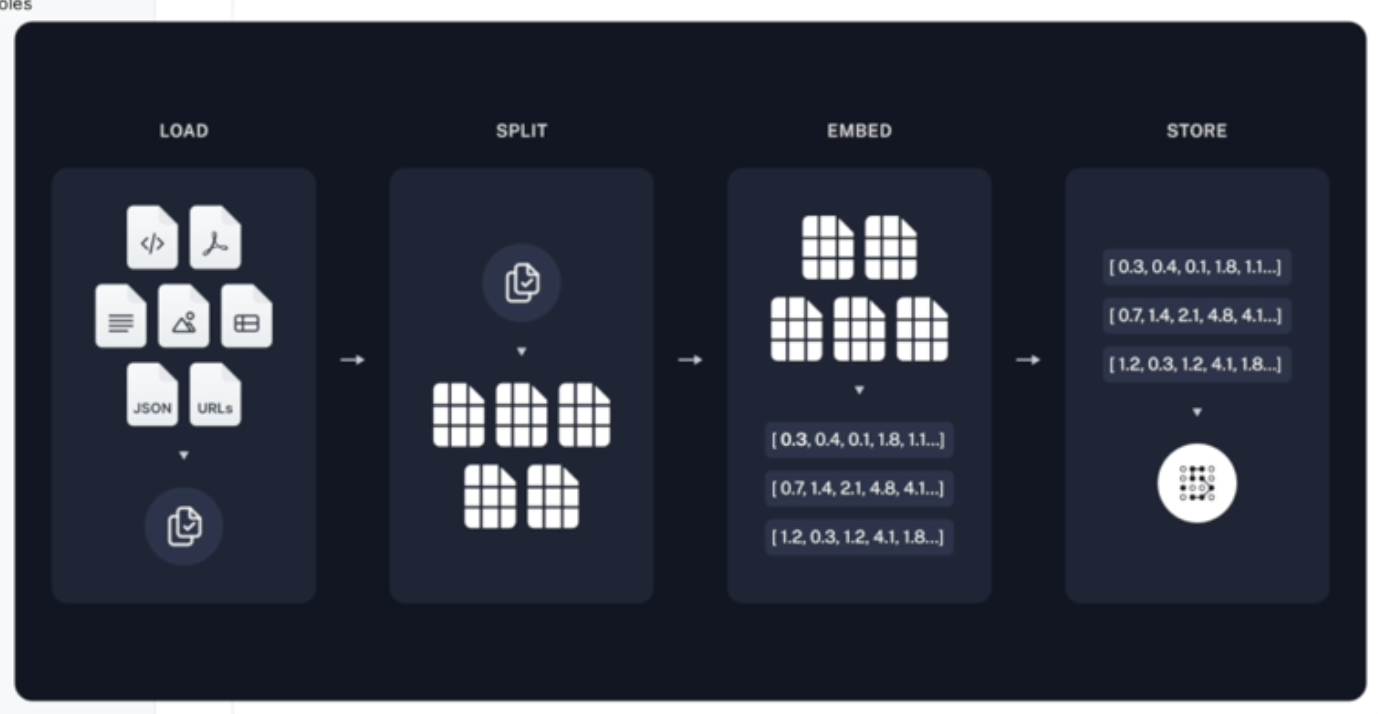
AI-generated content may be incorrect.

* It just becomes a structured tool.
* You can invoke the function, like how we do for every langchain tool has.
* Agent scratchpad is not used we’’ll do later.
* In LangChain, the partial() method in a template (like a PromptTemplate or ChatPromptTemplate) is used to "pre-fill" some of the variables in the template. This is useful when you want to reuse a prompt with some variables already filled in, while leaving others open for later.
* Stop=[“\nObservation”], it generates text even after that, but we are telling the model to stop here rather than hallucinating more.
* Prompt | llm -> The pipe thing is the langchain expression language (LCEL).

Inorder to keep the history of the chat, agent scratchpad is used.

* See the callback handler.

### Introduction Pinecone



* Open a pincone account.
* We need not to worry about availability, scalability, durability etc.
* While creating the index, we have to let it know what kind of metrics is used to find the distance.

A screenshot of a computer

AI-generated content may be incorrect.

* Vector Type – Dense & Sparse.
* We have to use langchain’s text loader and splitter packages. Splitter is quite complicated, there are lot of splitting strategies which has been used.