SW Engineering Principles and Django Q&A
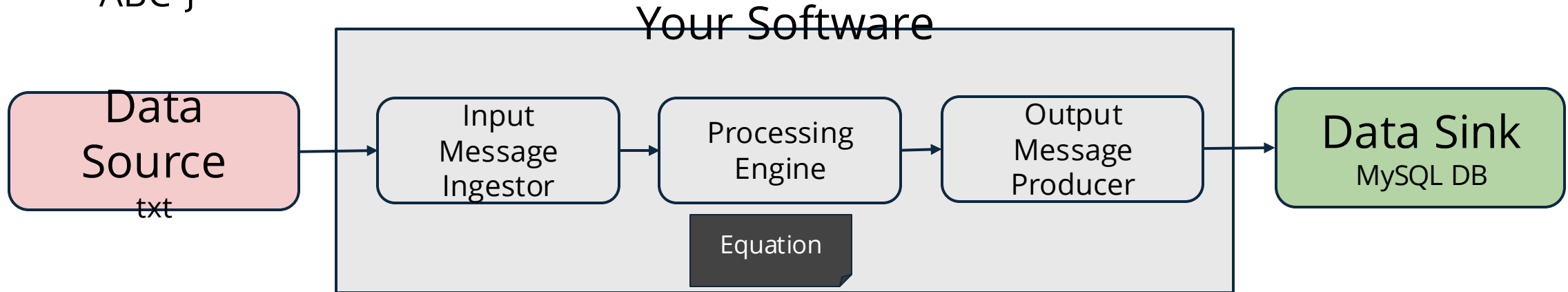

Esraa Sayed

Software Development Center, Giza Systems
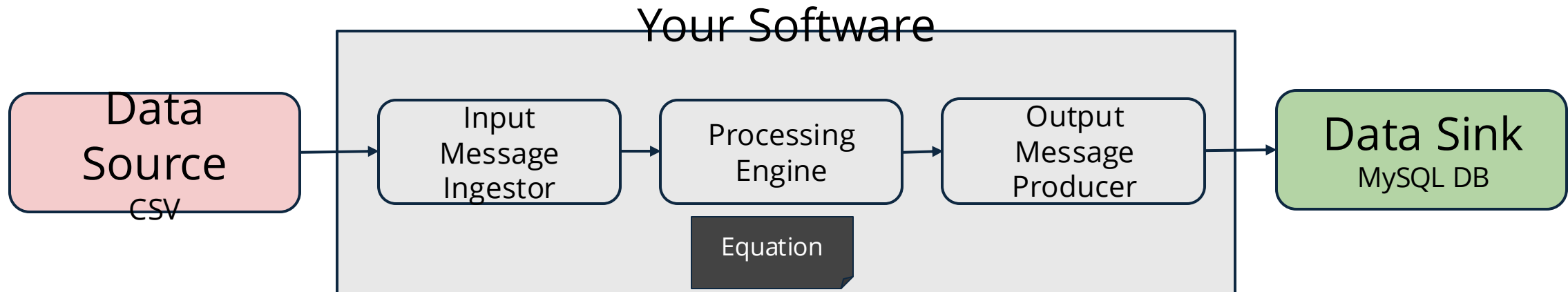
Task Description:

1. Suppose you have a txt file of records representing messages to be sent from multiple sensors at a frequency of 1 msg / 5 seconds (Data Ingestor will read the messages at this frequency).

2. The message will be in the following JSON format:
   {"asset_id": "123", "attribute_id": "#", "timestamp": "2022-07-31T23:28:37Z[UTC]", "value": "ABC"}

Task Description:

3.	Your processing engine will read an equation from a configuration file (Equation) representing the equation to be done on the ingested messages.

4.	Then, it will interpret the equation (Follow the blogs from 1 to 8 HERE to implement your own interpreter)

5.	The equation includes arithmetic operators (*, /, +, -, ^), and regex operator (Regex) to match the message value with a regular expression.

   a.	Examples of the format of the equation will be as follows:

   - ATTR+50*(ATTR/10) → calculate value of the received message using that arithmetic equation.

   - Regex(ATTR, "^dog") → matches the received message values if it starts with "dog", the produced value is "True" and "False" otherwise.

Your Software

Data Source CSV	Input Message Ingestor	Processing Engine	Output Message Producer	Data Sink MySQL DB

Equation

Giza Systems

Task Description:

6.   Your message producer will construct a message in the following JSON format and stores it in a relational database.
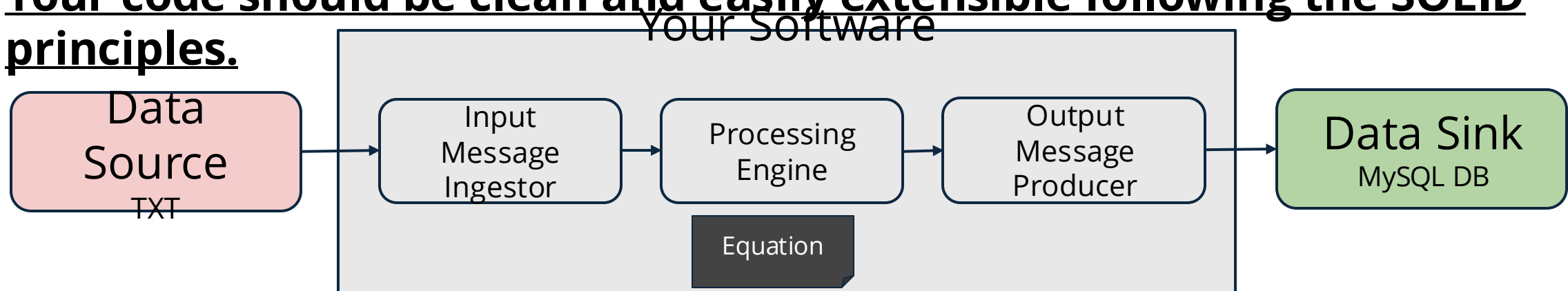
{"asset_id": "123", "attribute_id": "output_#", "timestamp": "2022-07-31T23:28:37Z[UTC]", "value": "XYZ"}.

Please note that this task will be modified and extended after later sessions.
Data Source, and Data Sink could be changed.
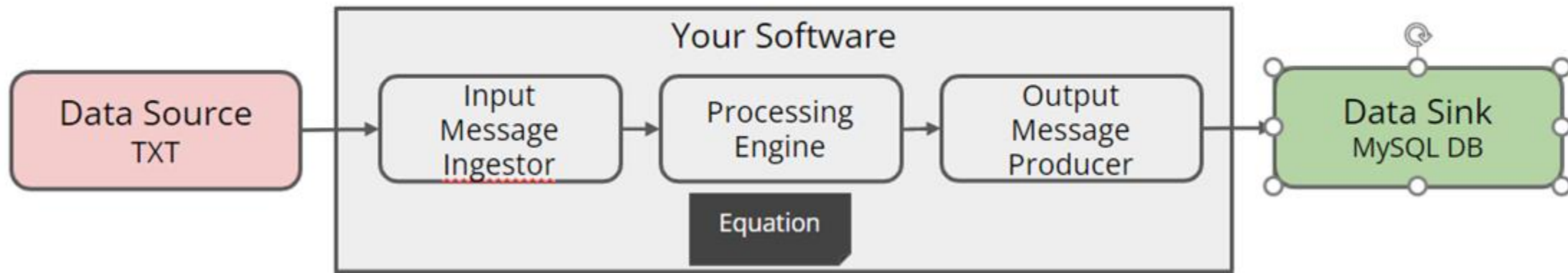**More operators could be supported other than arithmetic or regex matching.**
Your code should be clean and easily extensible following the SOLID principles.

Your Software

Data Source
TXT

Input Message Ingestor → Processing Engine → Output Message Producer

Equation

Data Sink
MySQL DB

Hands-On Task

Task Description:

7. Create a Django project that has KPI app
8. Create models to save KPI information : name , expression , description(optional)
9. Create model that link between the KPI table and asset_id
10. Create endpoint to list / create KPI
11. Create endpoint to link asset to a KPI

Note: Endpoints should be documented using swagger

code should be unit tested

- Create a repository for your tasks.
- Make sure to follow the DEV MOP Code Style, Testing, and Documentation Sections.
  - Make sure to apply OOP, SOLID principles as much as possible in addition to design patterns if needed.
- Deadline: will be conducted later when they receive their accounts

Your Software