Analytics Vidhya

Node Parsers

Instructors

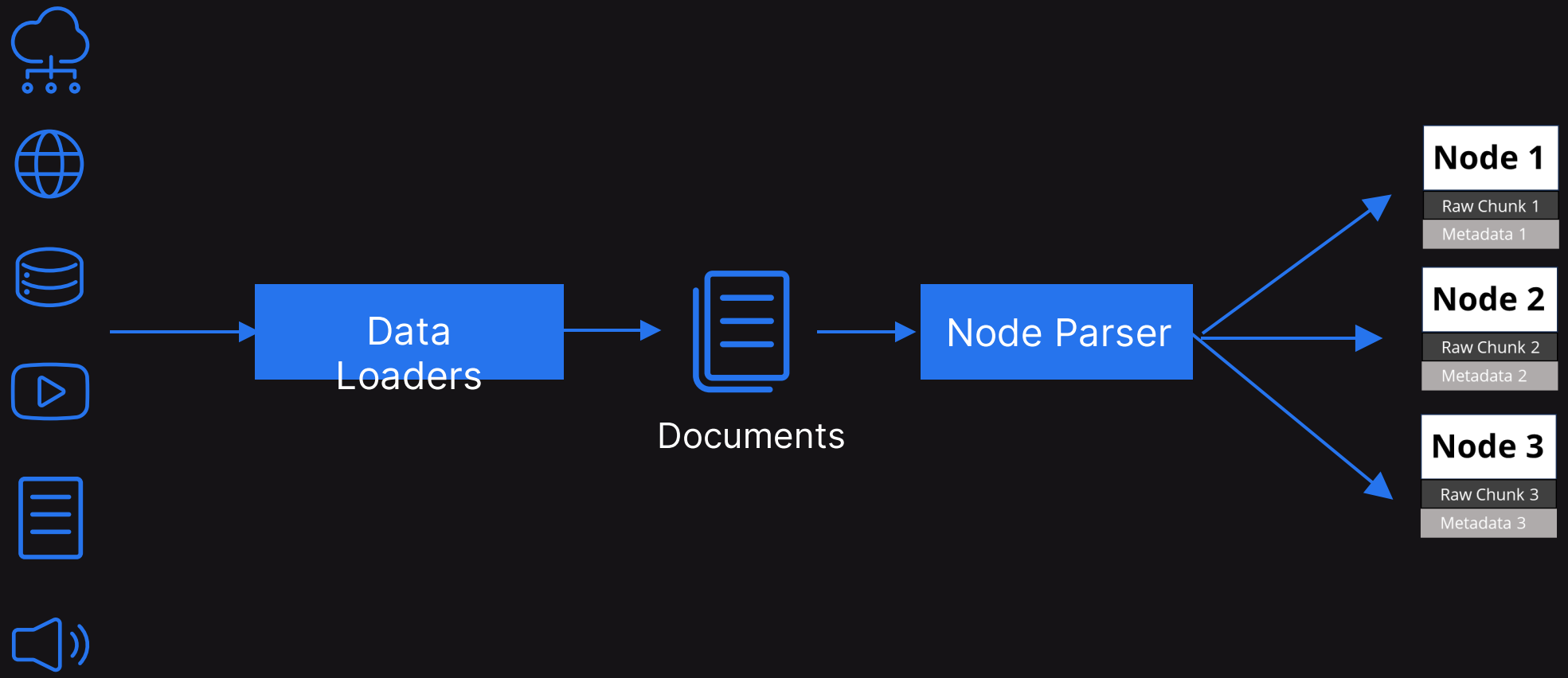
Prashant Sahu

Manager - Data Science, Analytics Vidhya

Ravi Theja

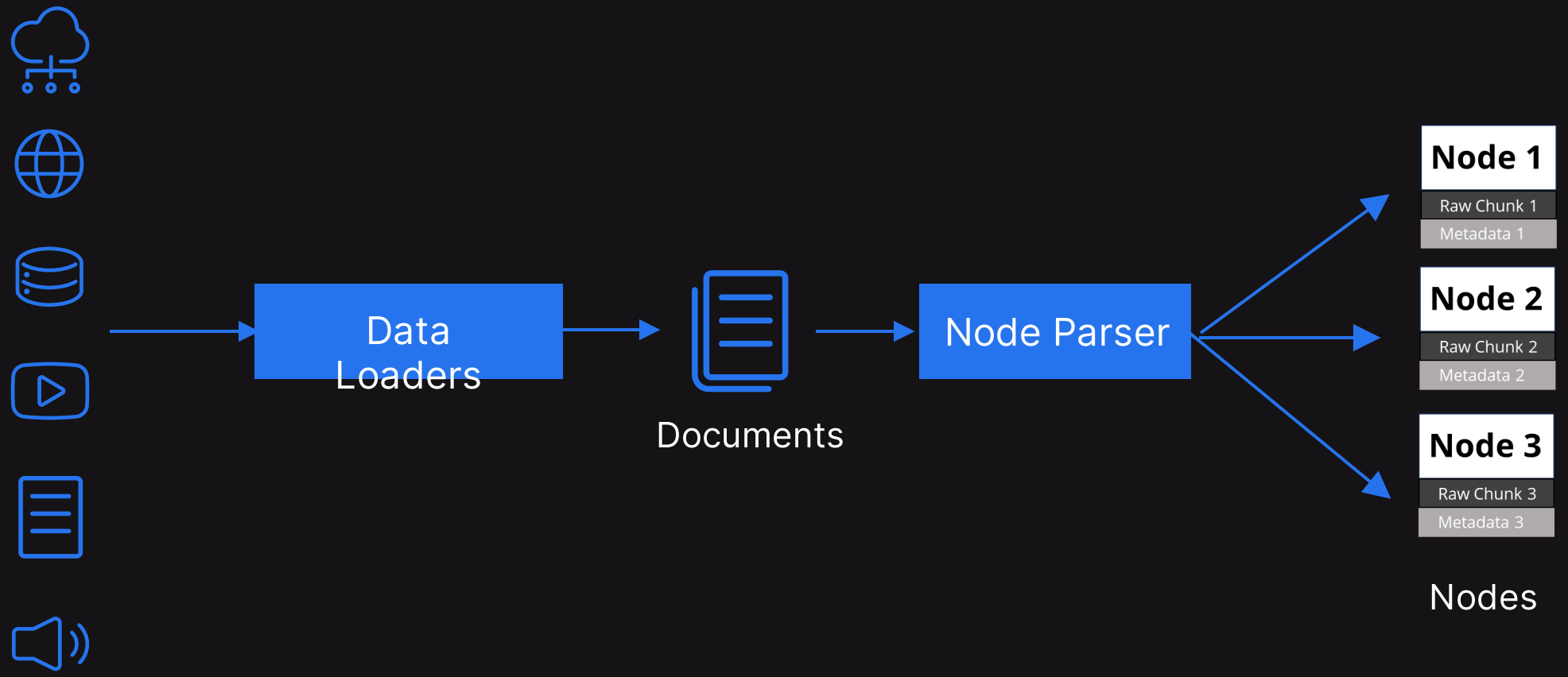
Developer Advocate Engineer, LlamaIndex

Documents in LlamaIndex



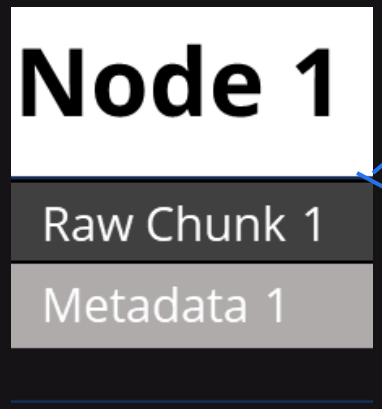
Analytics Vidhya

Nodes in LlamaIndex



Analytics Vidhya

Node




```
nodes[0].text
```

```
'In this part, we\nwill discuss the corresponding configurations for four major\nparts of the Transformer, includin
g normalization, position\nembeddings, activation functions, and attention and bias.\nTo make this survey more self
-contained, we present the\ndetailed formulations for these configurations in Table 6.\nNormalization Methods. Trai
ning instability is a challeng-\ning issue for pre-training LLMs. To alleviate this issue,\nnormalization is a wide
ly adopted strategy to stabilize the\ntraining of neural networks. In the vanilla Transformer [22],\nLayerNorm [25
6] is employed. Recently, several advanced\nnormalization techniques have been proposed as alterna-\ntives to Layer
Norm, e.g., RMSNorm, and DeepNorm.\n•LayerNorm. In the early research, BatchNorm [265] is\na commonly used normaliz
ation method. However, it is\ndifficult to deal with sequence data of variable lengths and\nsmall-batch data. Thus,
LayerNorm [256] is introduced to\nconduct layerwise normalization. Specifically, the mean and\nvariance over all ac
```

Raw Chunk for Node

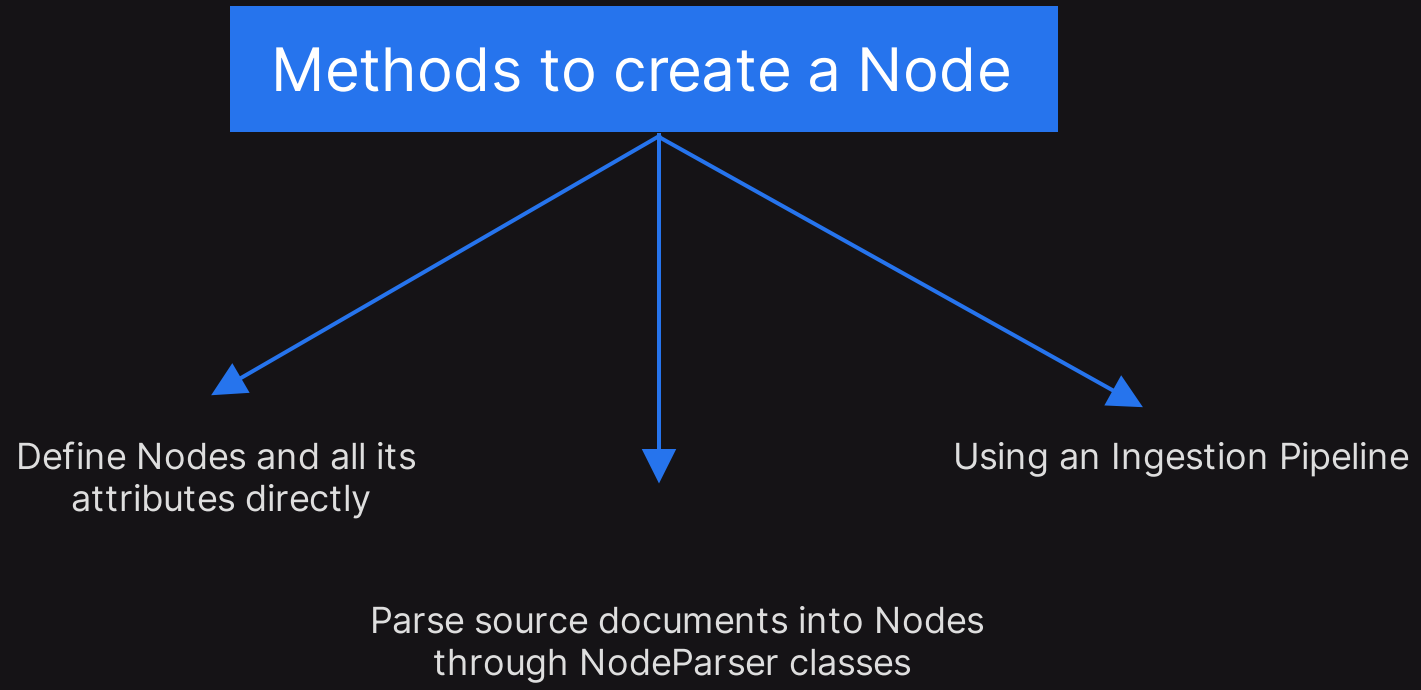
```
nodes[0].metadata
```

```
{'page_label': '23',
 'file_name': 'A Survey of Large Language Models.pdf',
 'file_path': 'A Survey of Large Language Models.pdf',
 'file_type': 'application/pdf',
 'file_size': 5664979,
 'creation_date': '2024-05-14',
 'last_modified_date': '2024-05-14'}
```

Metadata for Node

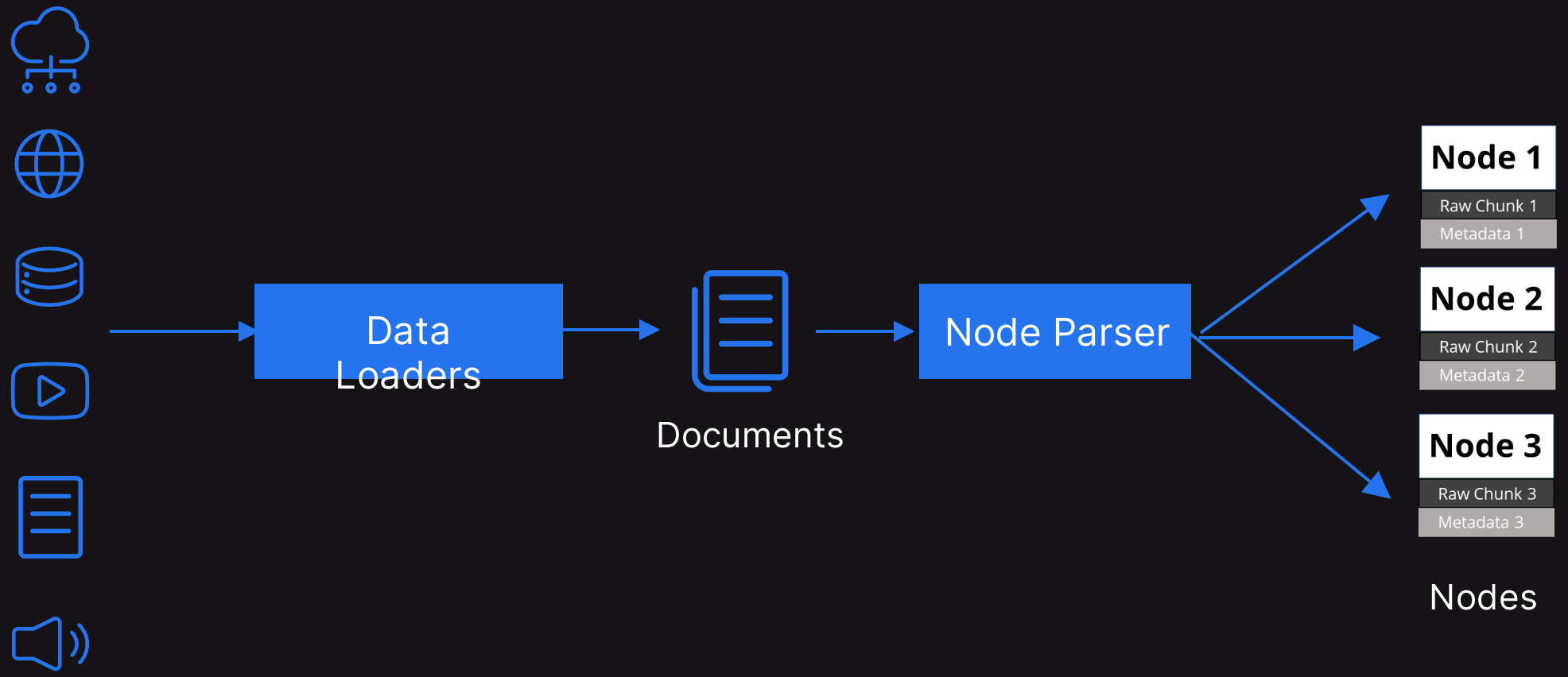
Analytics Vidhya

Methods to create Nodes



Analytics Vidhya

Introduction to Node Parsers



Analytics Vidhya

Node Parsers

Node Parsers break down data into manageable pieces (nodes) based on specific rules or formats, such as text, HTML, JSON, or CSV.

Analytics Vidhya

Node Parsers in Llama Index

- **TextNodeParser**
  For parsing plain text from documents and articles
- **HTMLNodeParser**
  For parsing HTML documents
- **JSONNodeParser**
  Processes JSON data structures
- **CSVNodeParser**
  Handles CSV files by extracting data from rows

Analytics Vidhya

Node Parsers in Llama Index

- **MarkdownNodeParser**

  Specifically designed for Markdown files

- **HierarchicalNodeParser**

  Creates a recursive hierarchy of nodes by splitting documents into multiple levels of nodes.

- **SimpleFileNodeParser**

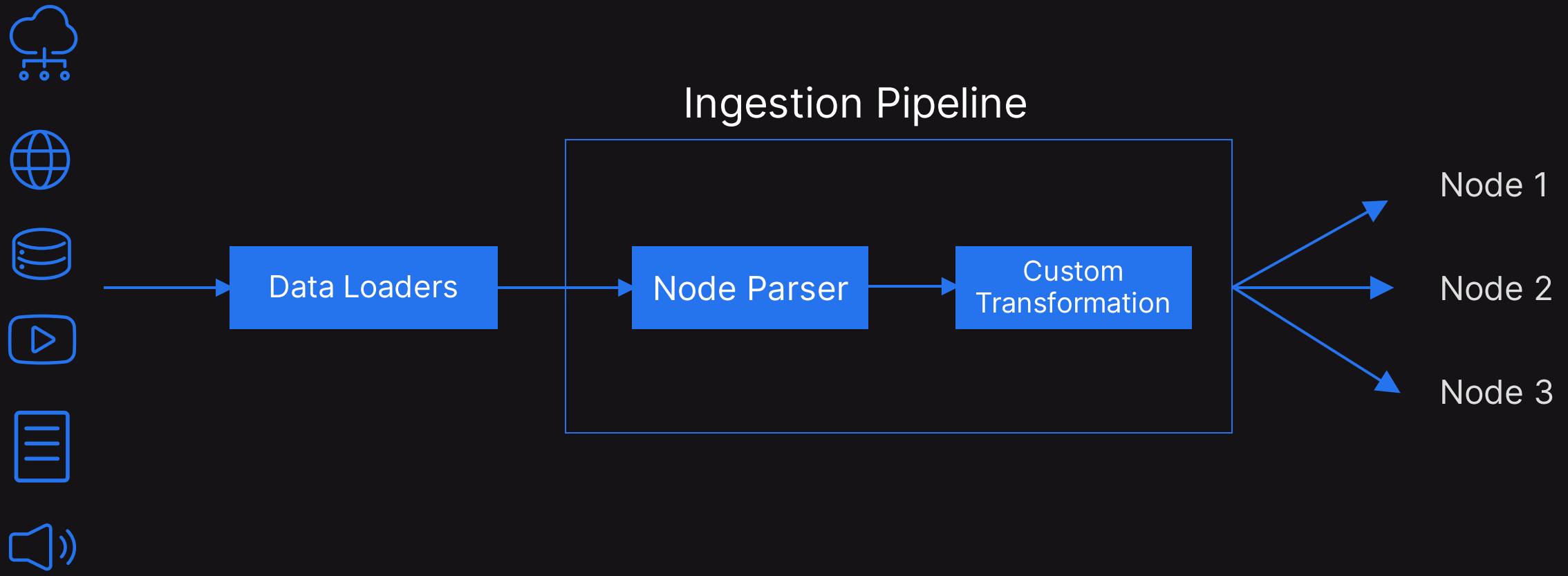
  Handling of different file formats within a single application.

- **SimpleNodeParser**

  Splits documents into text nodes using a sentence splitter by default.

Analytics Vidhya

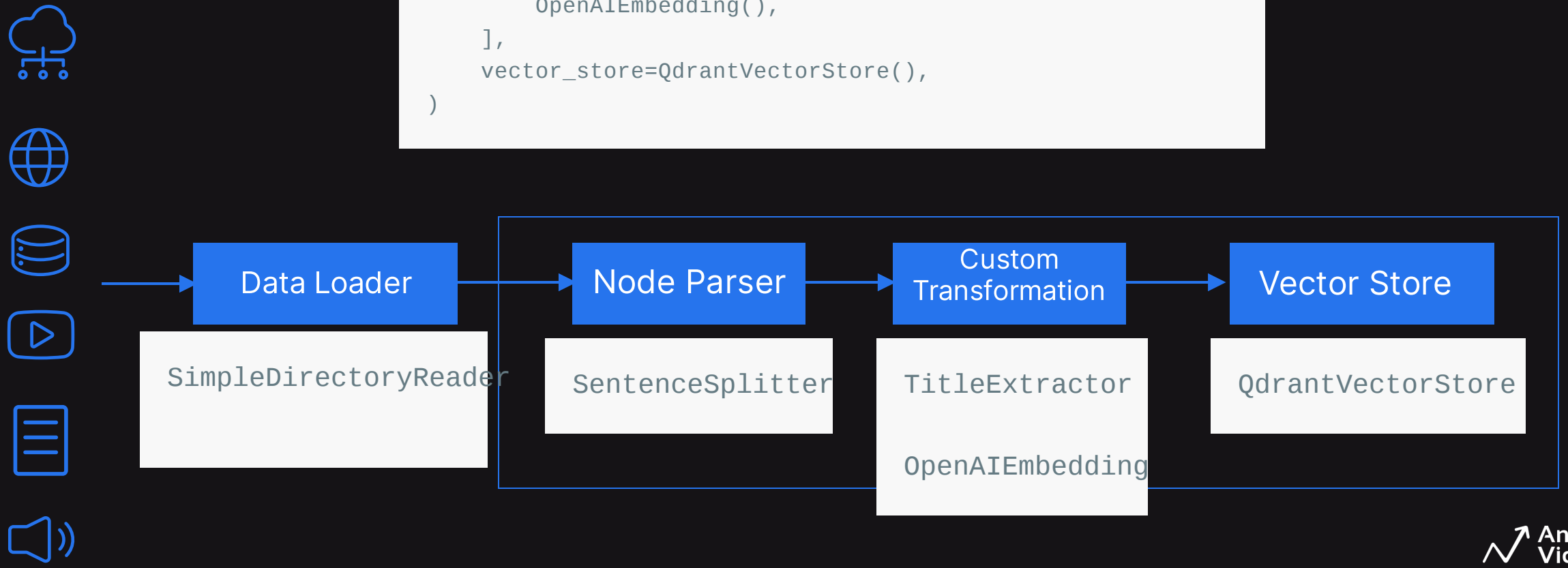
Ingestion Pipeline



Analytics Vidhya

Ingestion Pipeline: Example

```
pipeline = IngestionPipeline(
    transformations=[
        SentenceSplitter(chunk_size=500, chunk_overlap=25),
        TitleExtractor(),
        OpenAIEmbedding(),
    ],
    vector_store=QdrantVectorStore(),
)
```



Analytics Vidhya

Thank You

Analytics Vidhya