Data Pipeline Design

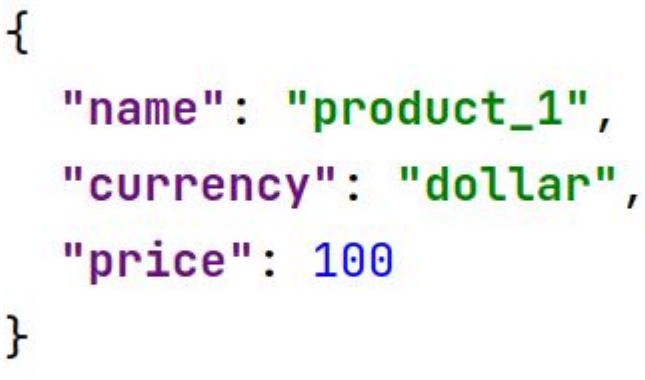
Advanced Python Programming

Valerio Velardo - The Sound of AI

What do we want to build?

A data pipeline that reads product records from json files, processes them, and stores the processed data in a DB.

Example product json file



```
{
  "name": "product_1",
  "currency": "dollar",
  "price": 100
}
```

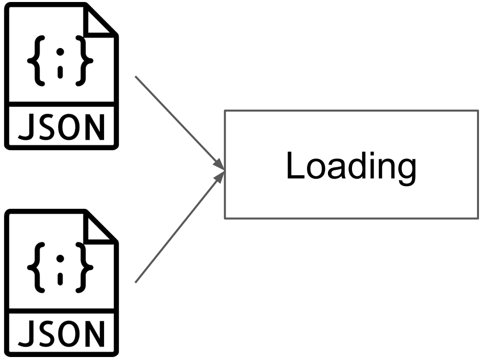
Programme workflow

Programme workflow

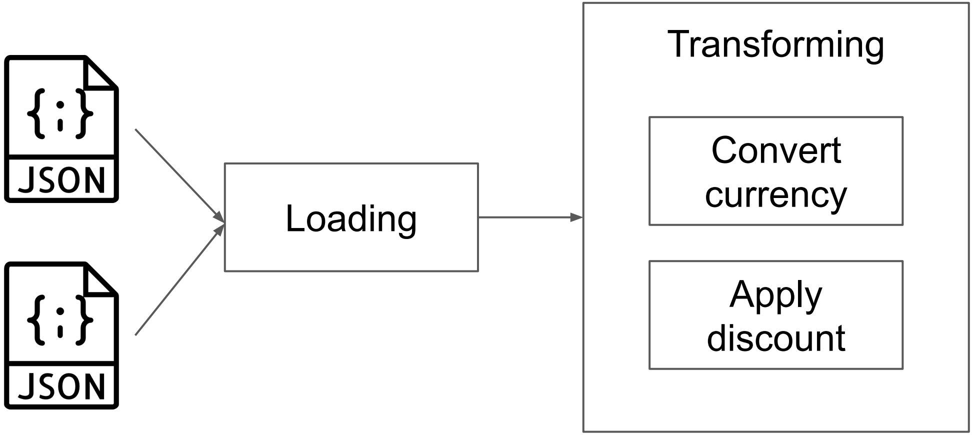
JSON

JSON

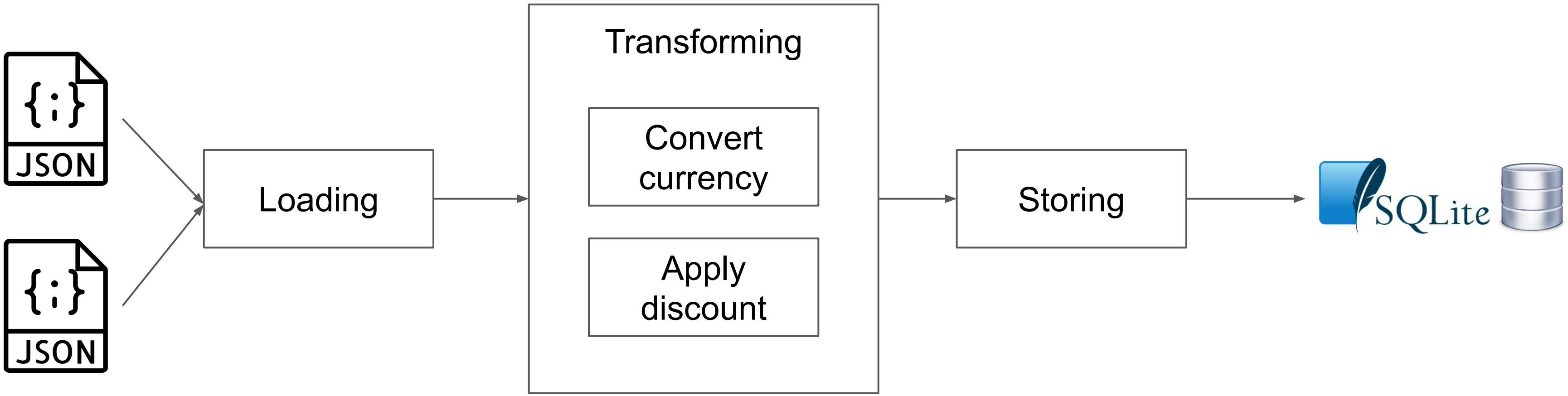
Programme workflow



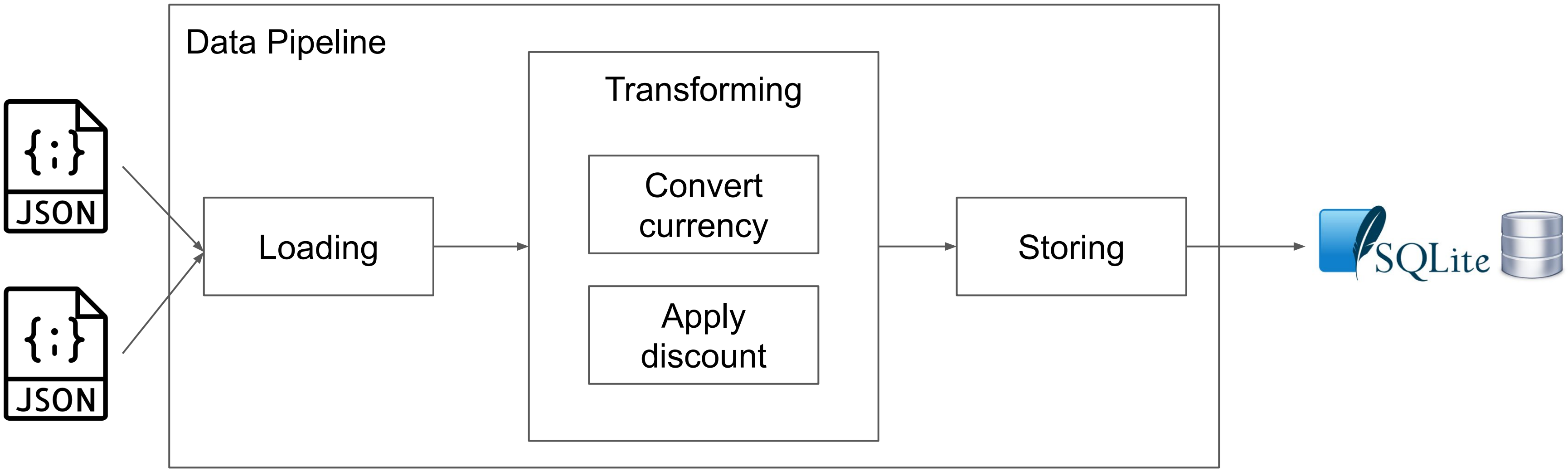
Programme workflow



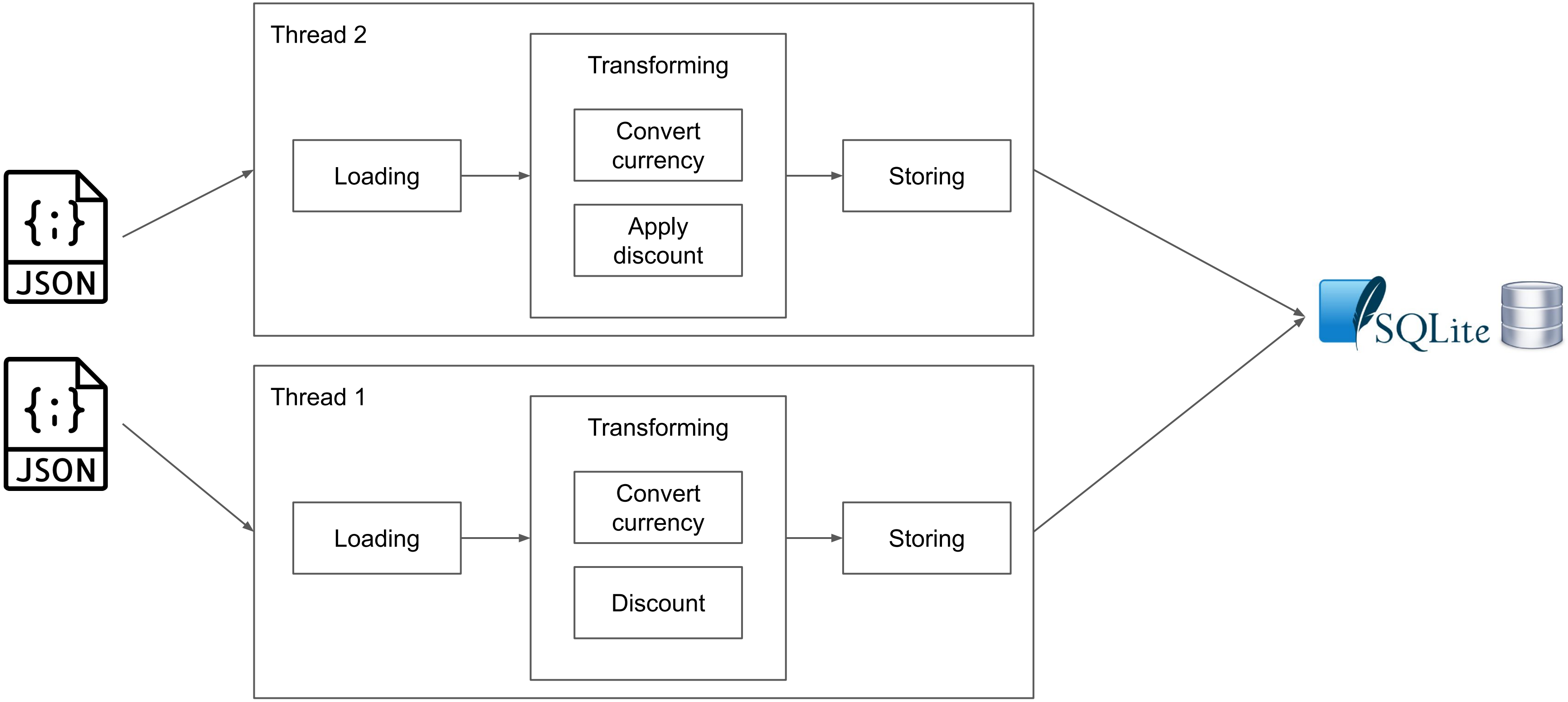
Programme workflow



Programme workflow



Programme workflow with threading


Thread 2

Loading

Transforming

Convert currency

Apply discount

Storing

Thread 1

Loading

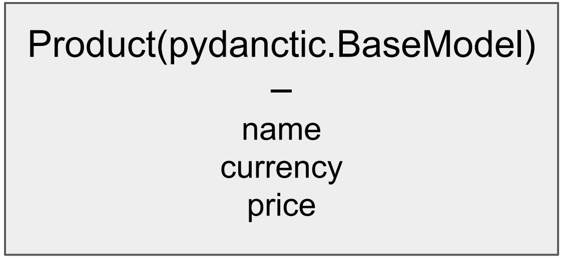
Transforming

Convert currency

Discount

Storing

Product DTO



Product(pydanctic.BaseModel)

–

name
currency
price

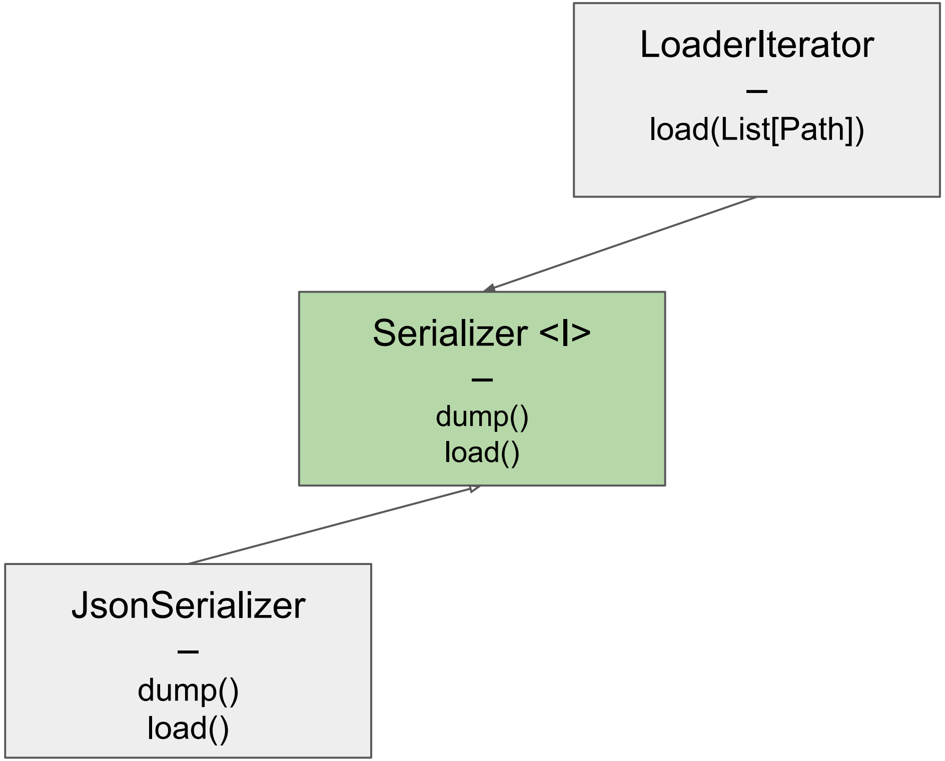
Loading component

LoaderIterator

–

load(List[Path])

Loading component

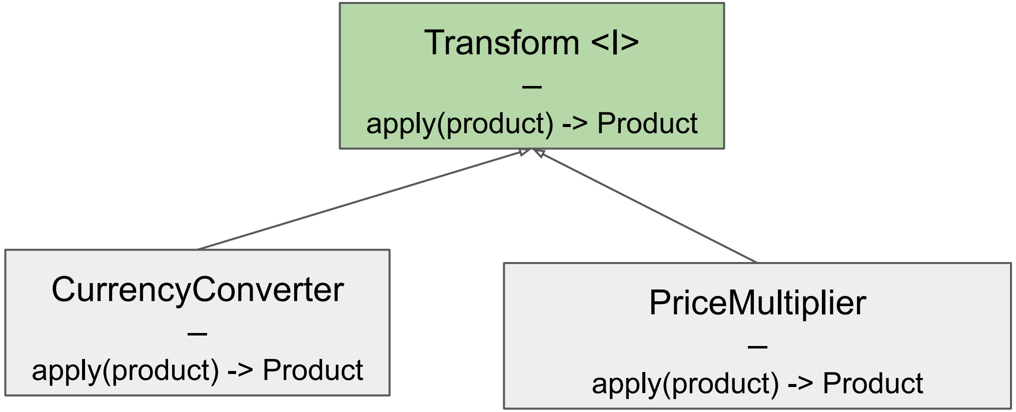


LoaderIterator
–
load(List[Path])

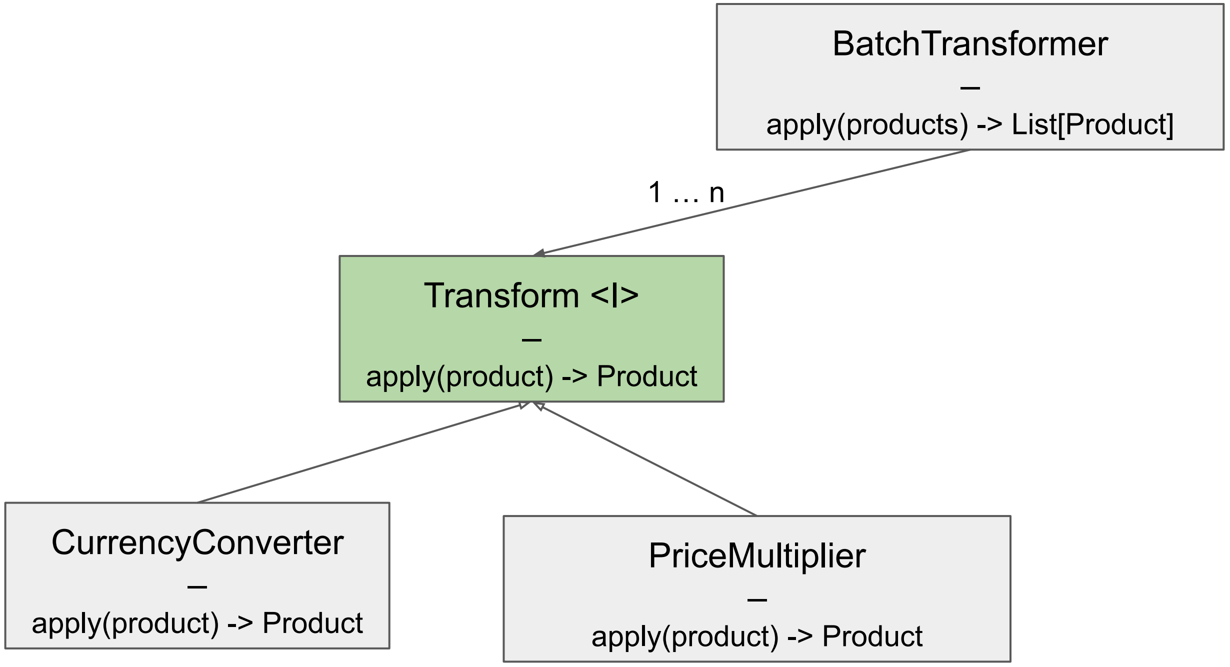
Serializer <I>
–
dump()
load()

JsonSerializer
–
dump()
load()

Transforming component


Transform <I>
–
apply(product) -> Product
CurrencyConverter
–
apply(product) -> Product
PriceMultiplier
–
apply(product) -> Product

Transforming component



BatchTransformer

–

apply(products) -> List[Product]

1 … n

Transform <I>

–

apply(product) -> Product

CurrencyConverter

–

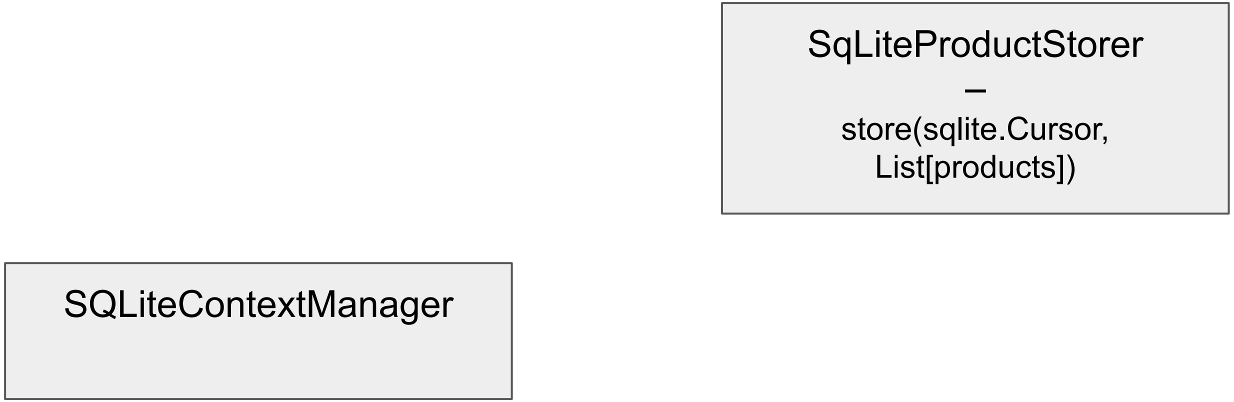
apply(product) -> Product

PriceMultiplier

–

apply(product) -> Product

Storing component



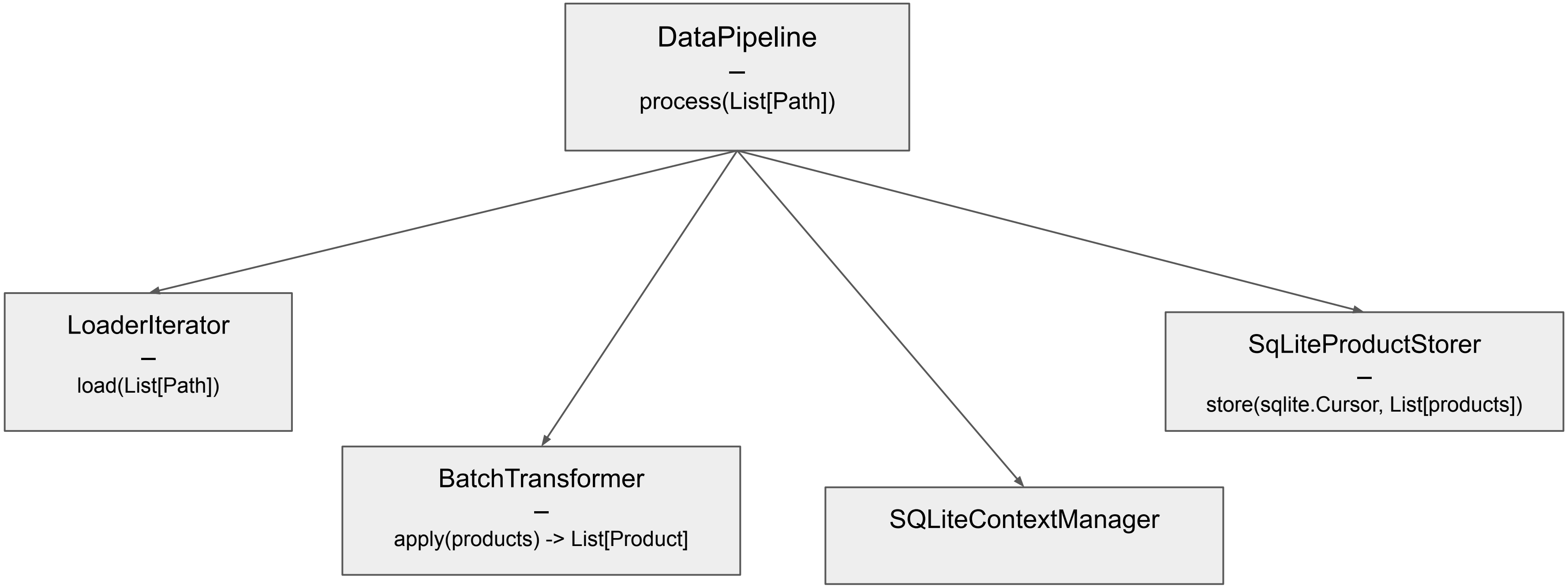
SqLiteProductStorer

–

store(sqlite.Cursor, List[products])

SQLiteContextManager

Data pipeline component



DataPipeline
–
process(List[Path])

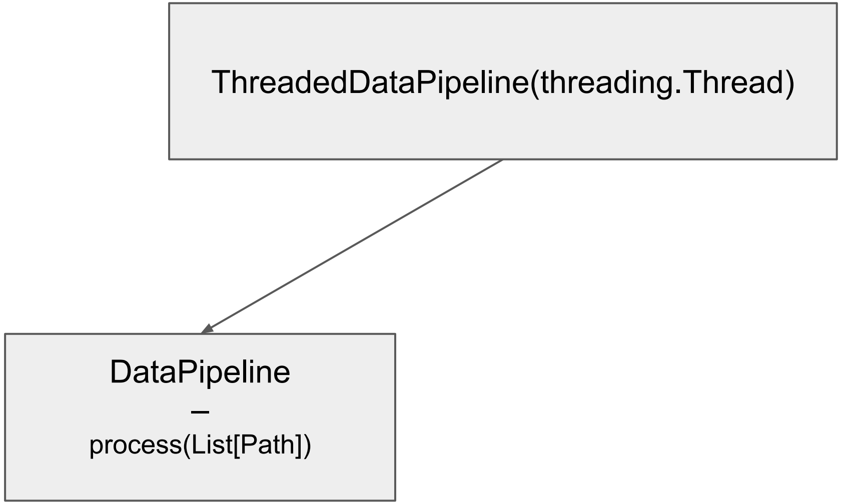
LoaderIterator
–
load(List[Path])

BatchTransformer
–
apply(products) -> List[Product]

SQLiteContextManager

SqLiteProductStorer
–
store(sqlite.Cursor, List[products])

Threading component



ThreadedDataPipeline(threading.Thread)

DataPipeline
–
process(List[Path])

What’s next?

1. Product DTO
2. Loading component
3. Transforming component
4. Storing component
5. Data pipeline
6. Multi threading
7. Running the programme