

Scraping Web Sites Using Scrapy Spiders



Janani Ravi

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Defining crawling tasks in classes using Spiders

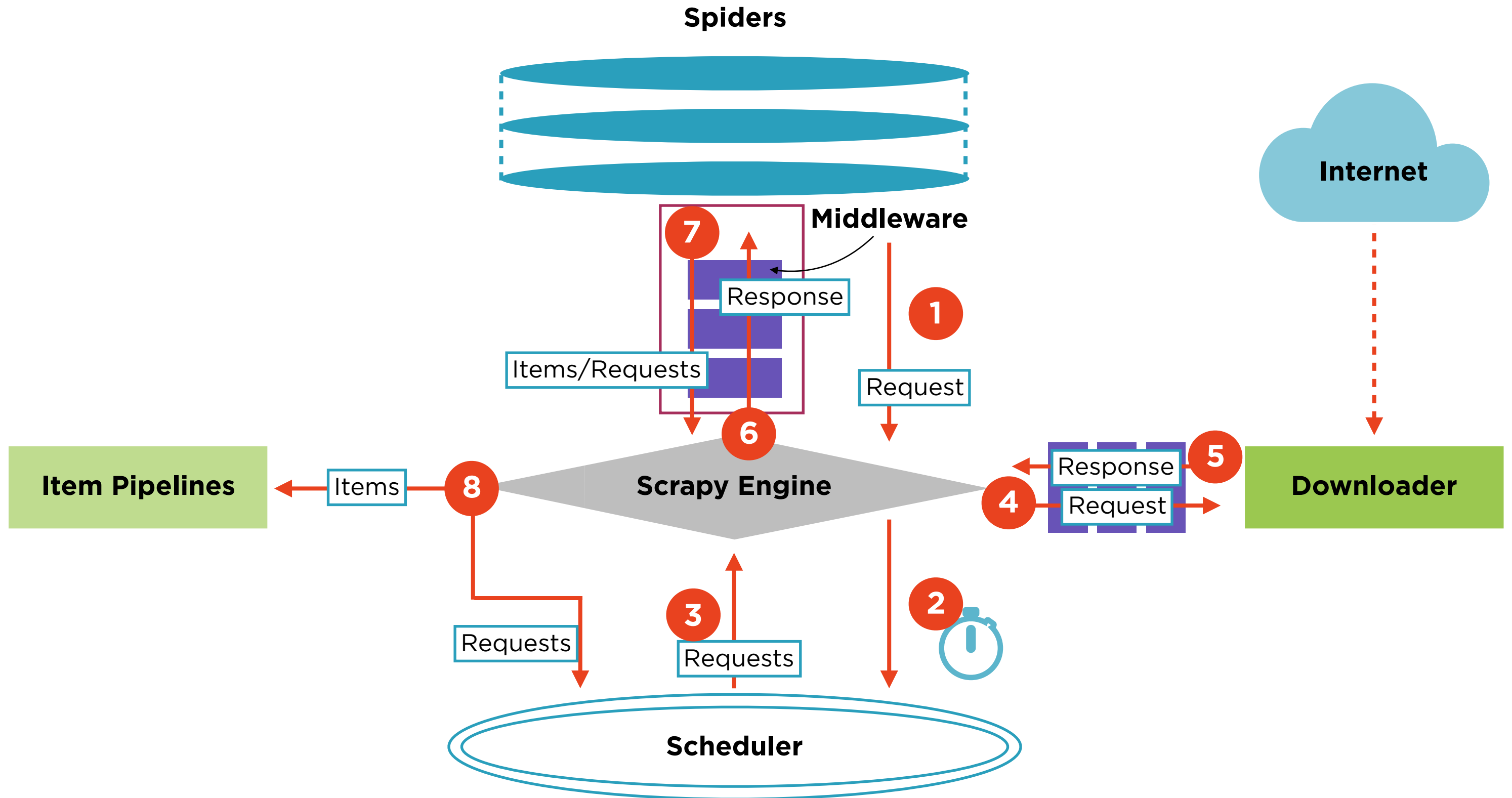
Extracting logical information using items and processors

Using item loaders to automate data extraction

Chaining data transformations using pipelines

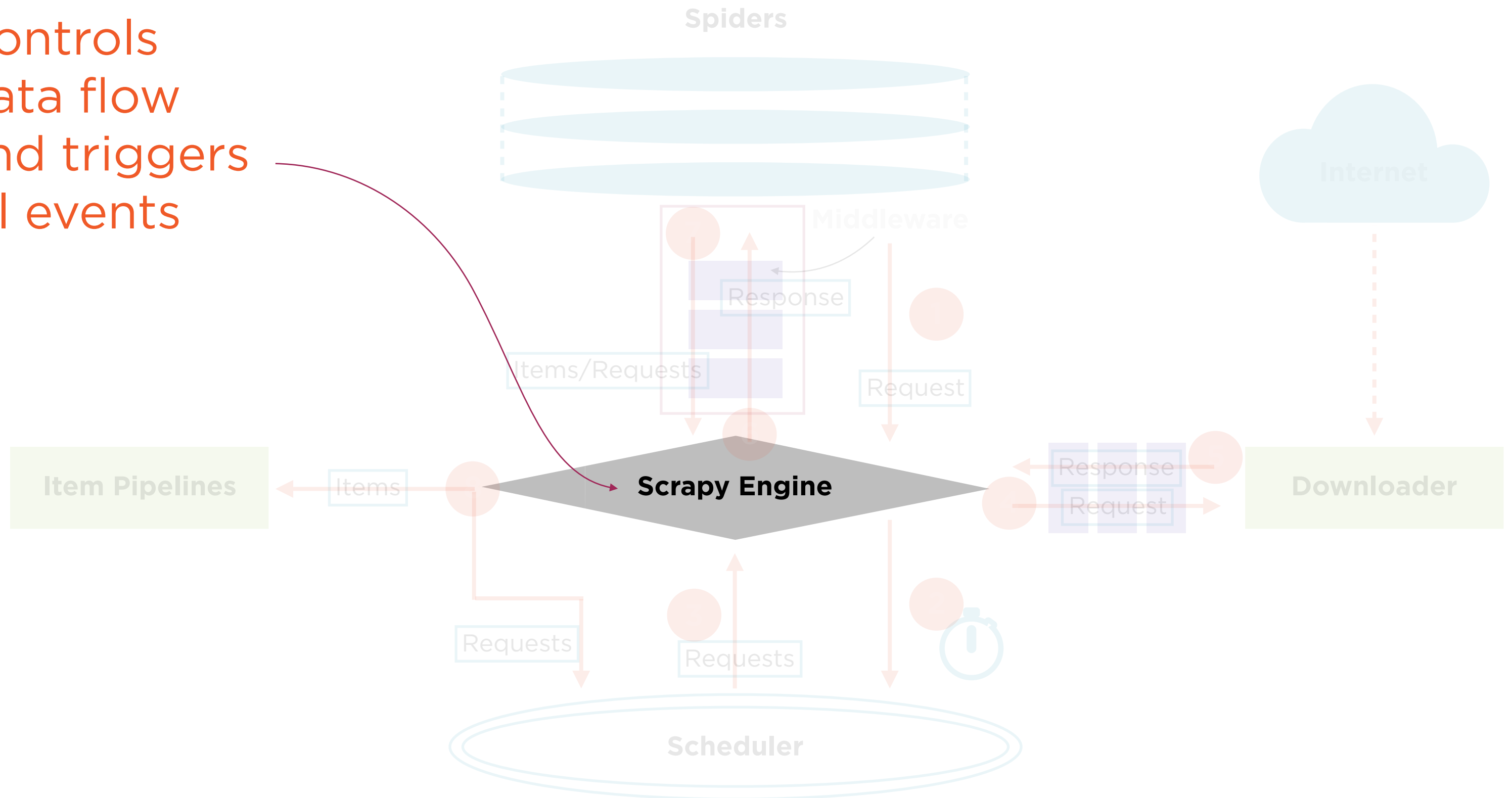
How Scrappy Works

How Scrapy Works



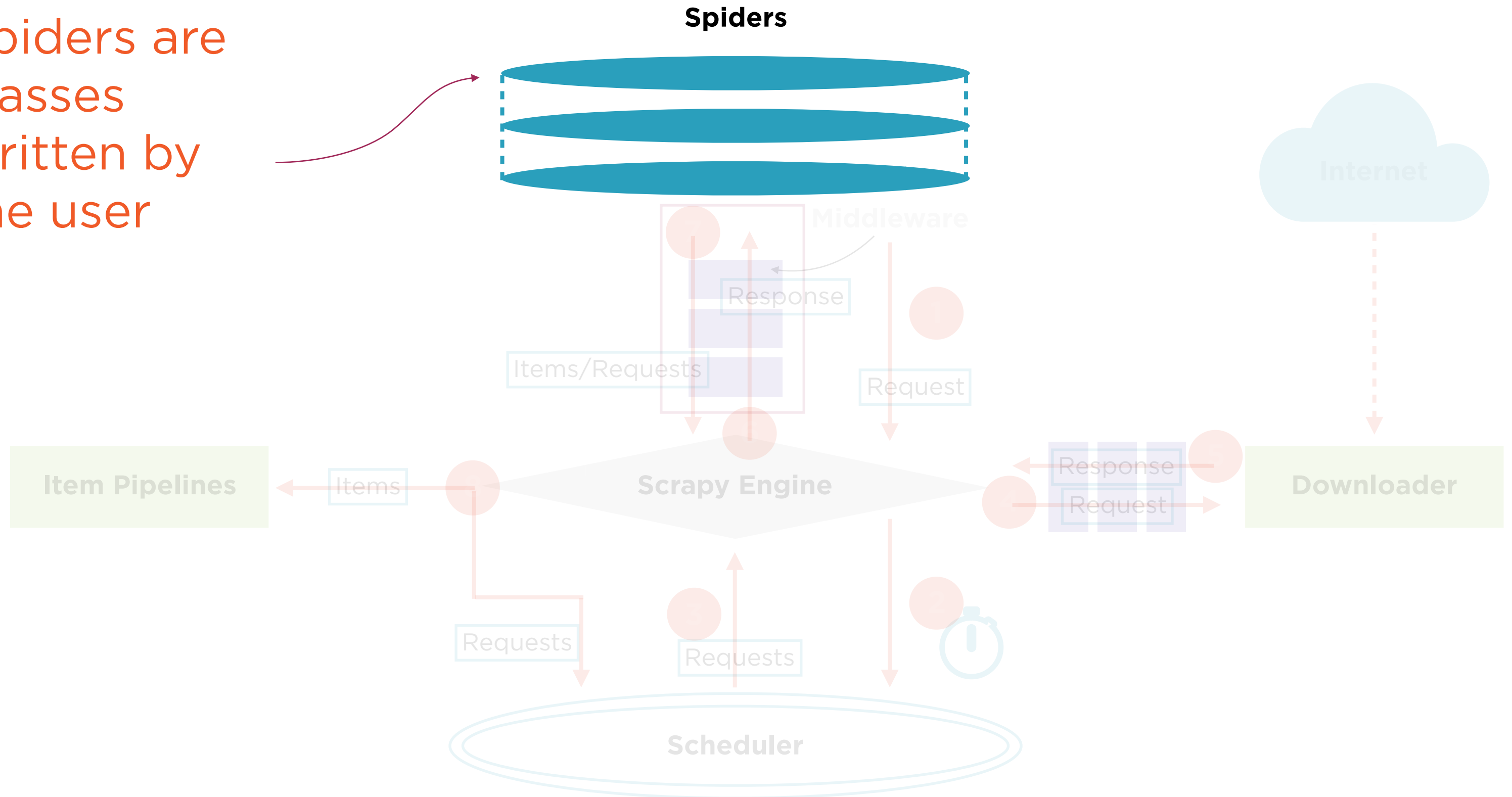
How Scrapy Works

Controls
data flow
and triggers
all events



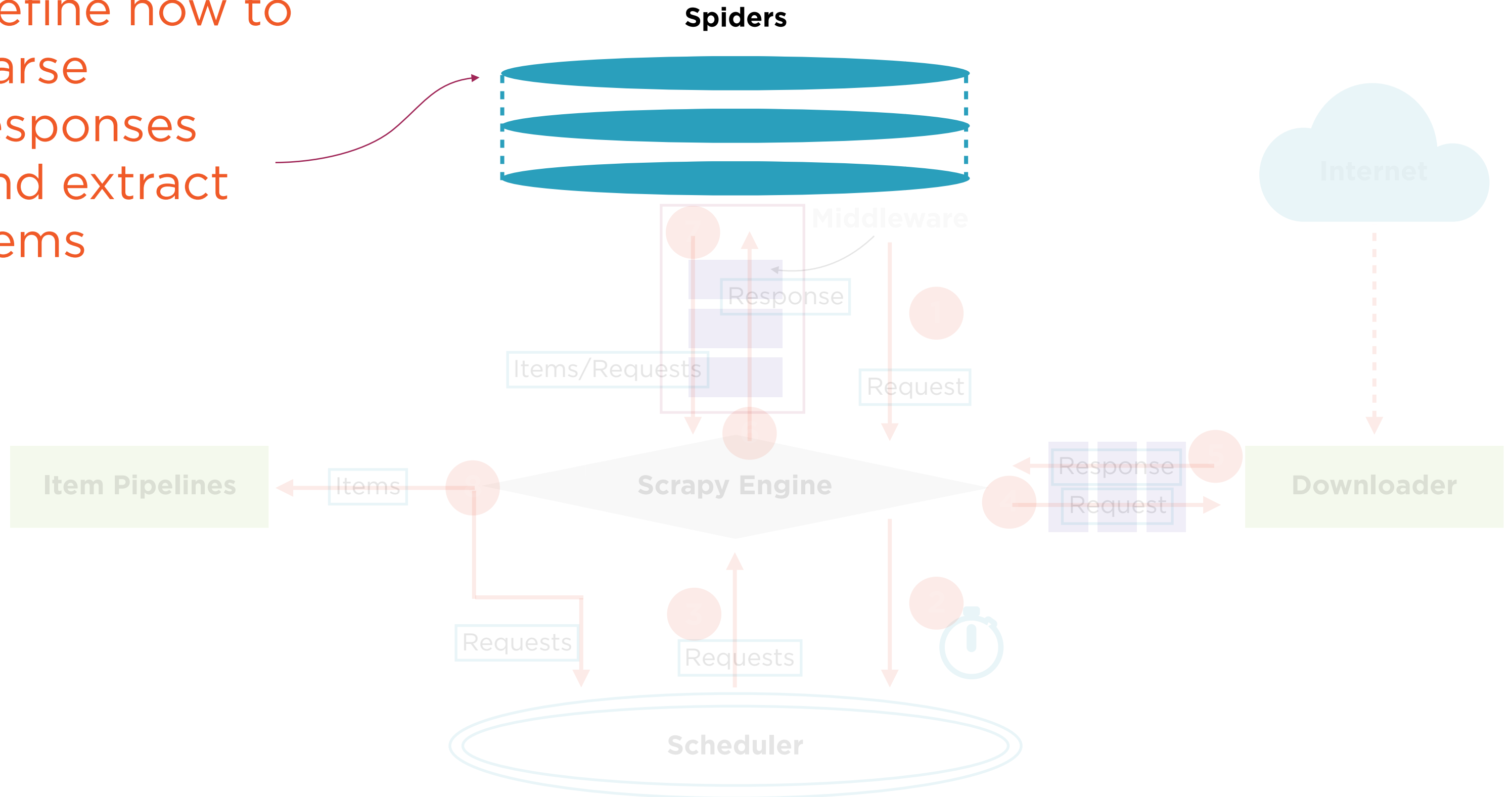
How Scrapy Works

Spiders are classes written by the user



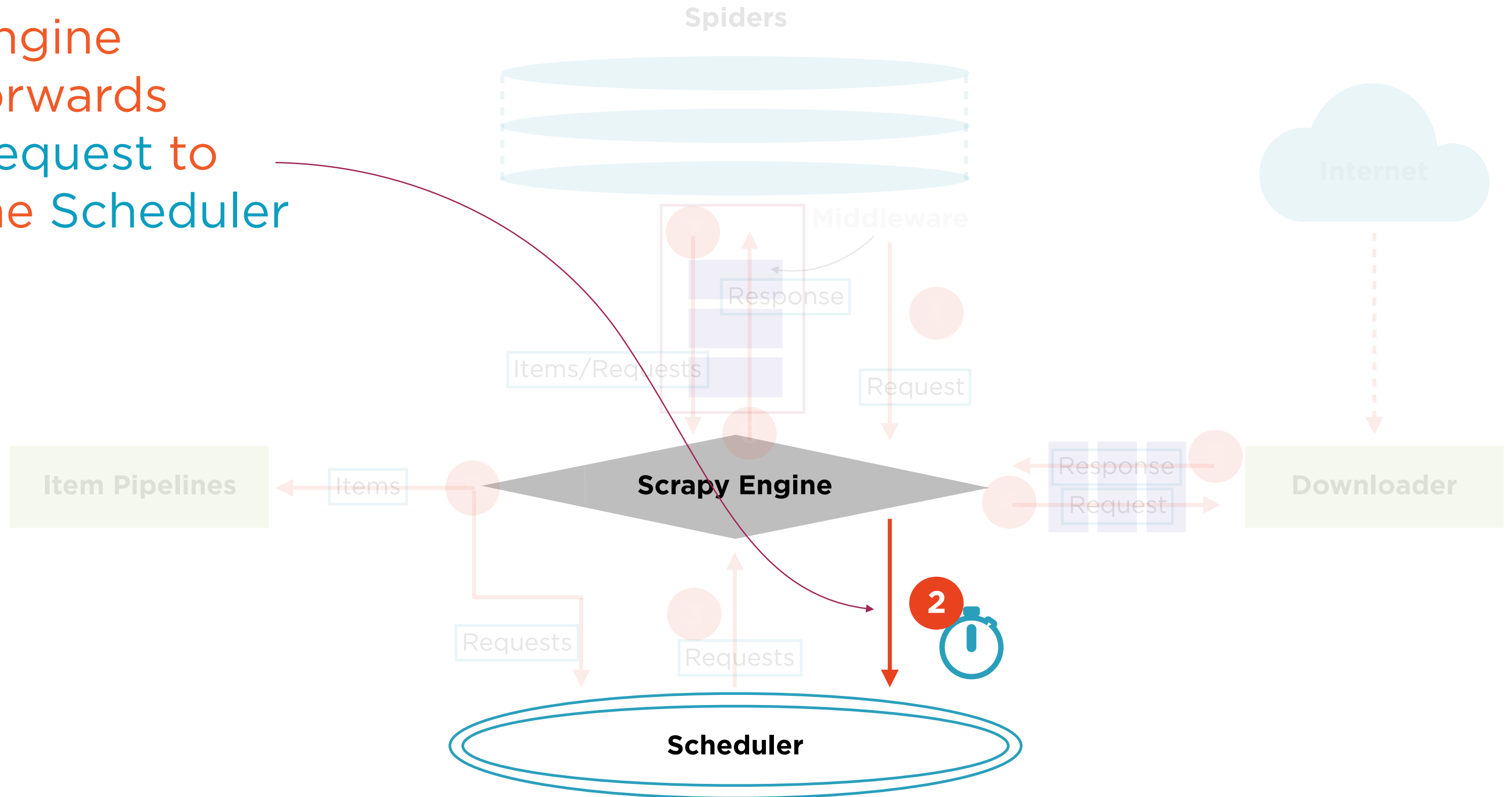
How Scrapy Works

Define how to
parse
responses
and extract
items



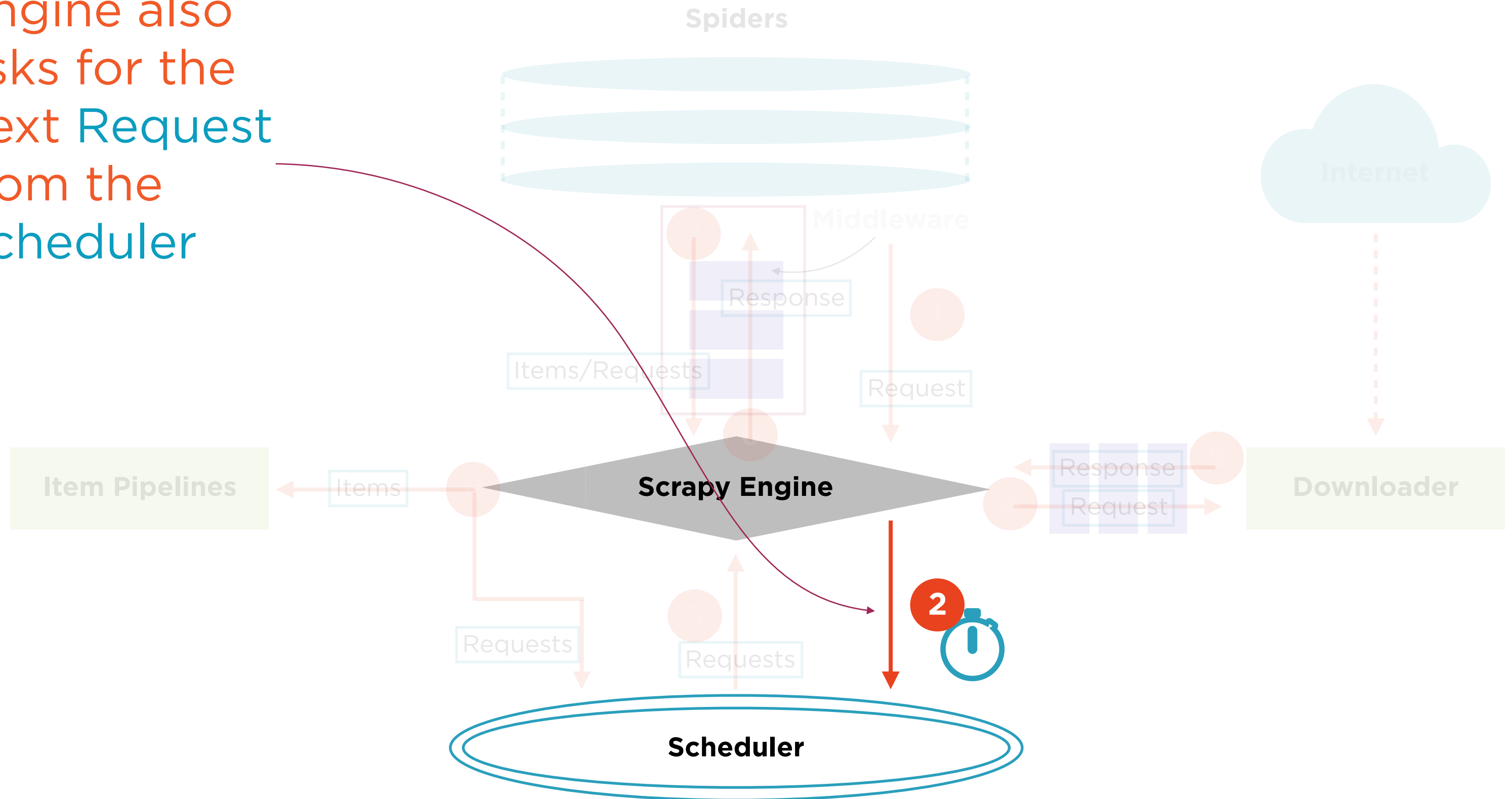
How Scrapy Works

Engine
forwards
Request to
the Scheduler



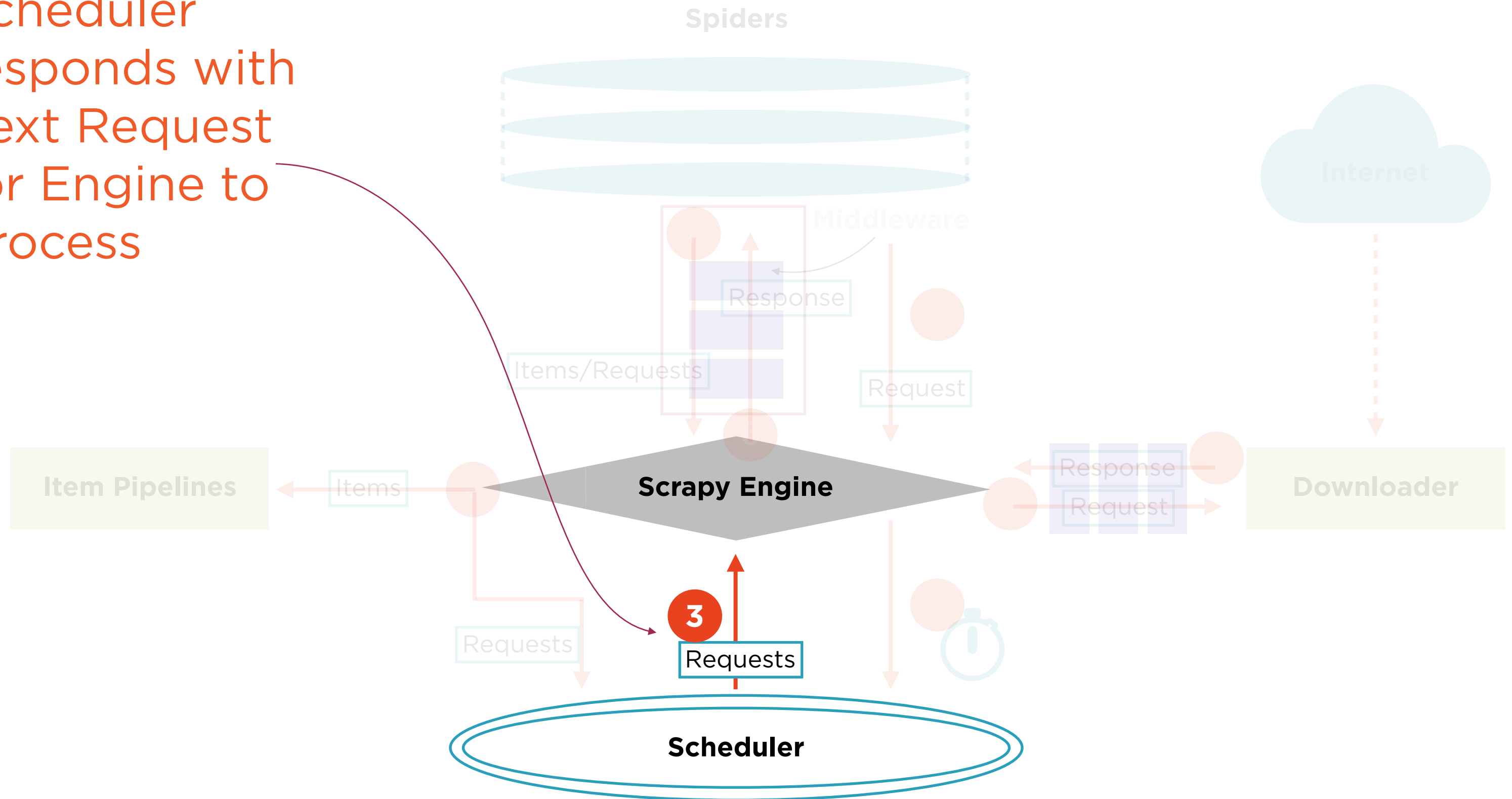
How Scrapy Works

Engine also asks for the next Request from the Scheduler



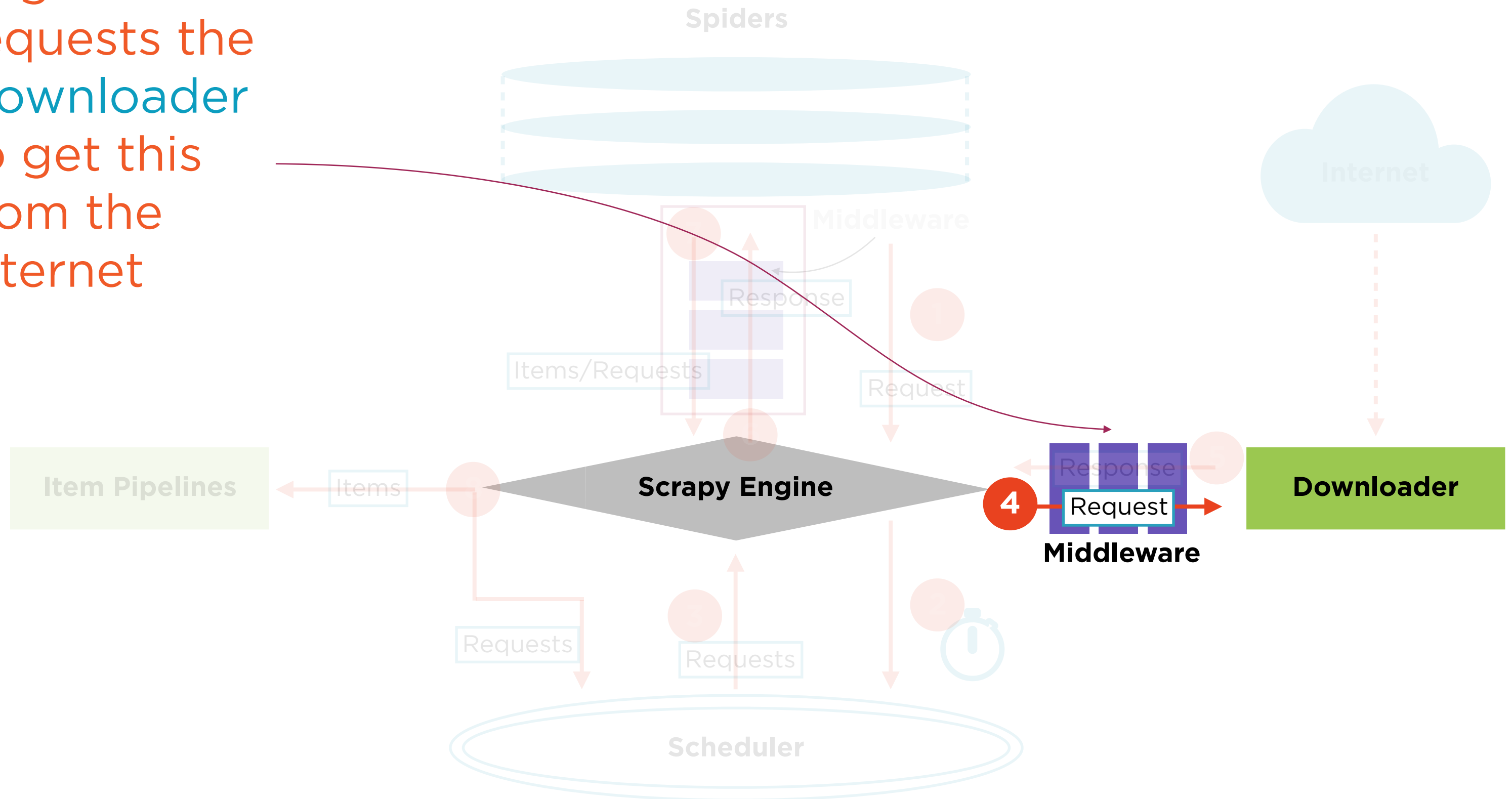
How Scrapy Works

Scheduler
responds with
next Request
for Engine to
process



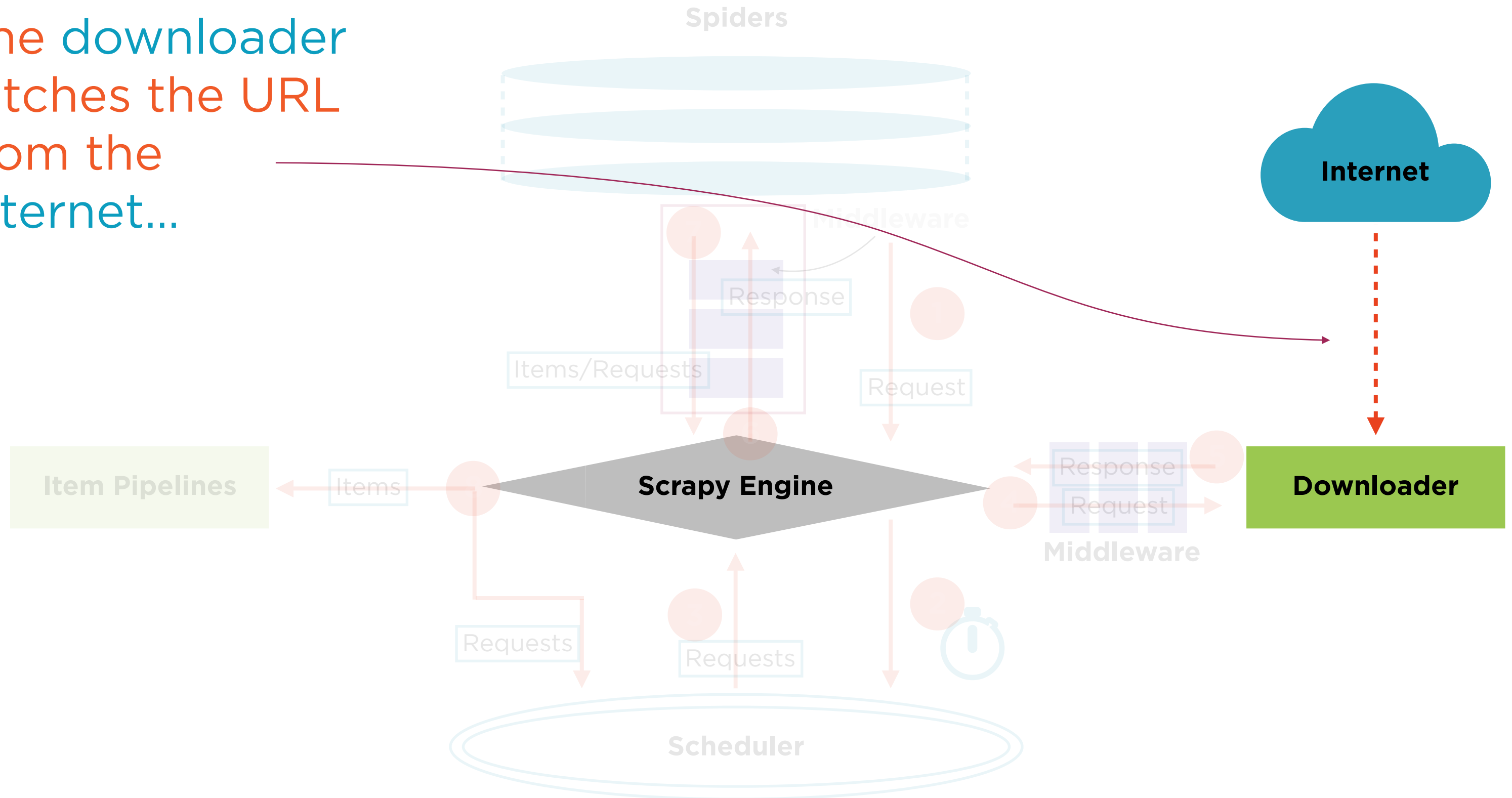
How Scrapy Works

Engine
requests the
Downloader
to get this
from the
internet



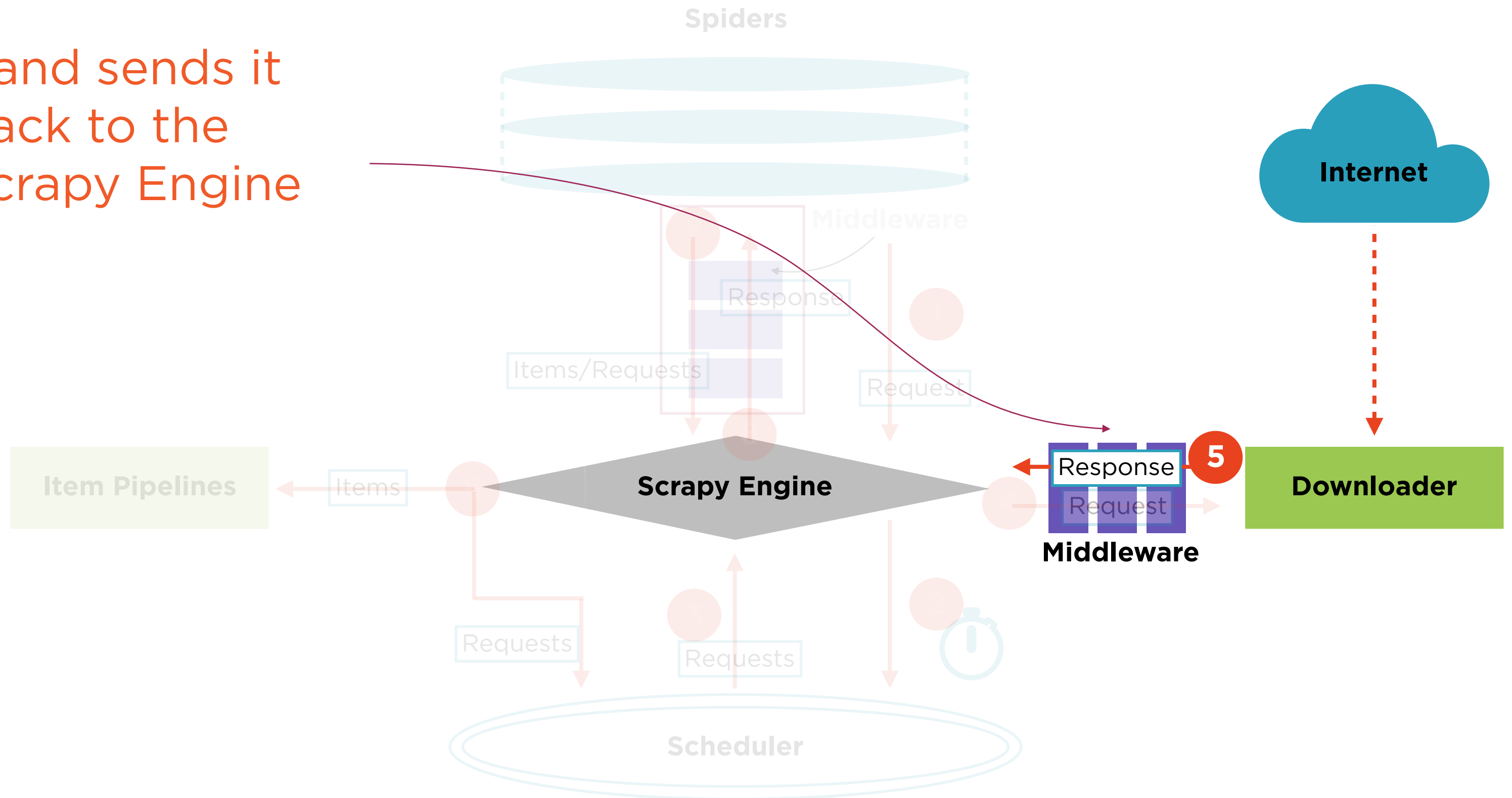
How Scrapy Works

The downloader
fetches the URL
from the
internet...



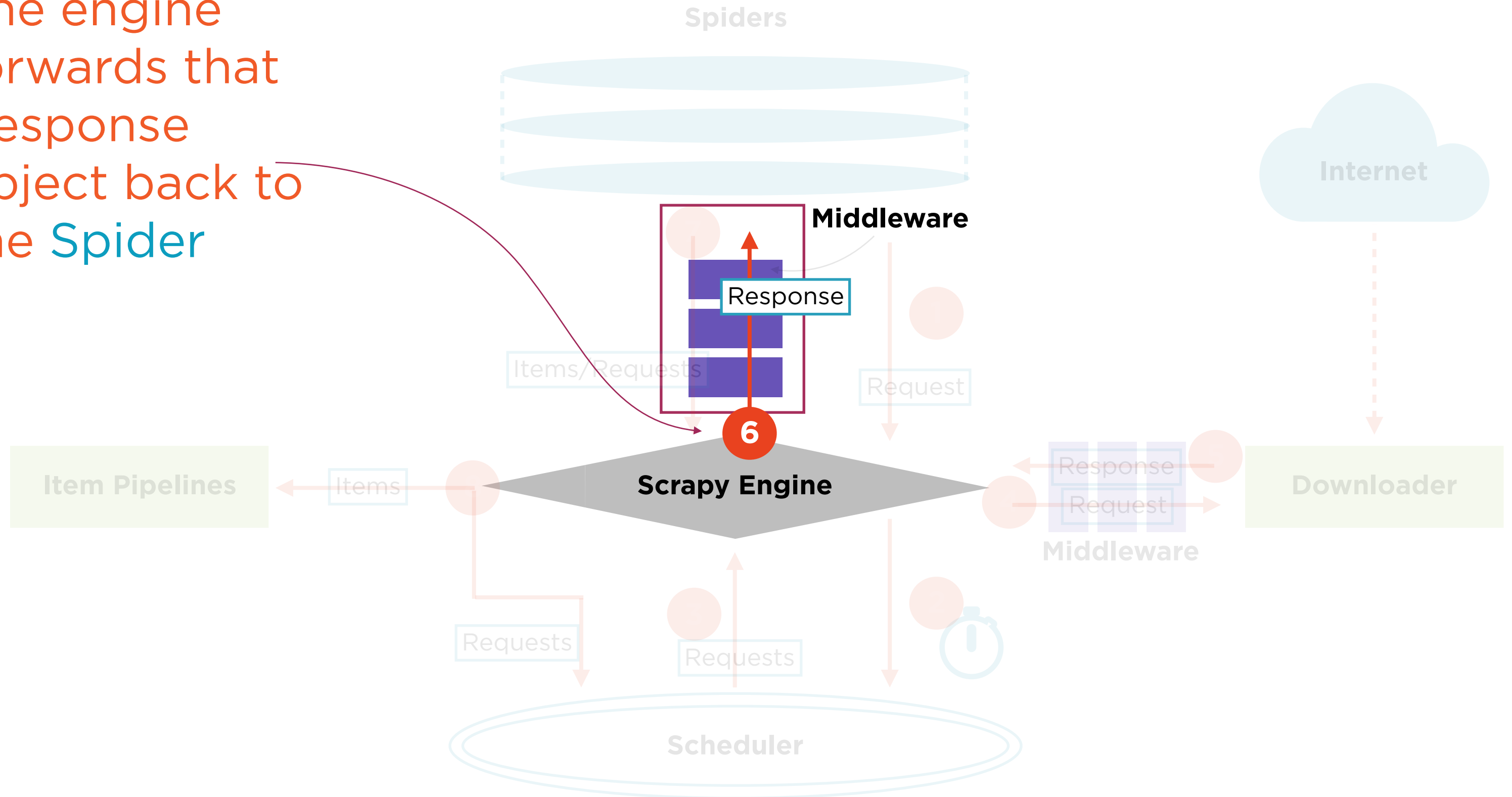
How Scrapy Works

...and sends it
back to the
Scrapy Engine



How Scrapy Works

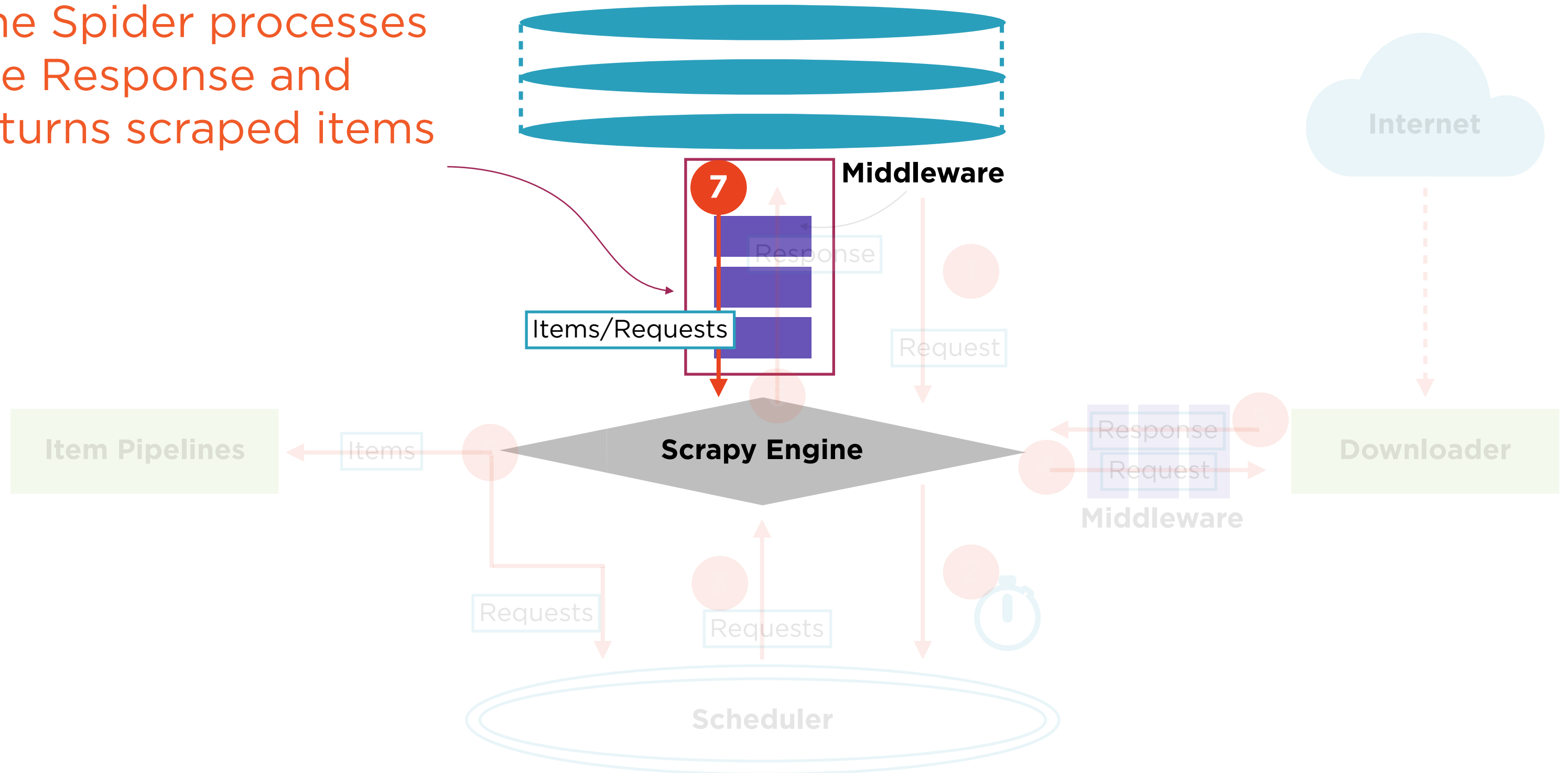
The engine forwards that Response object back to the Spider



How Scrapy Works

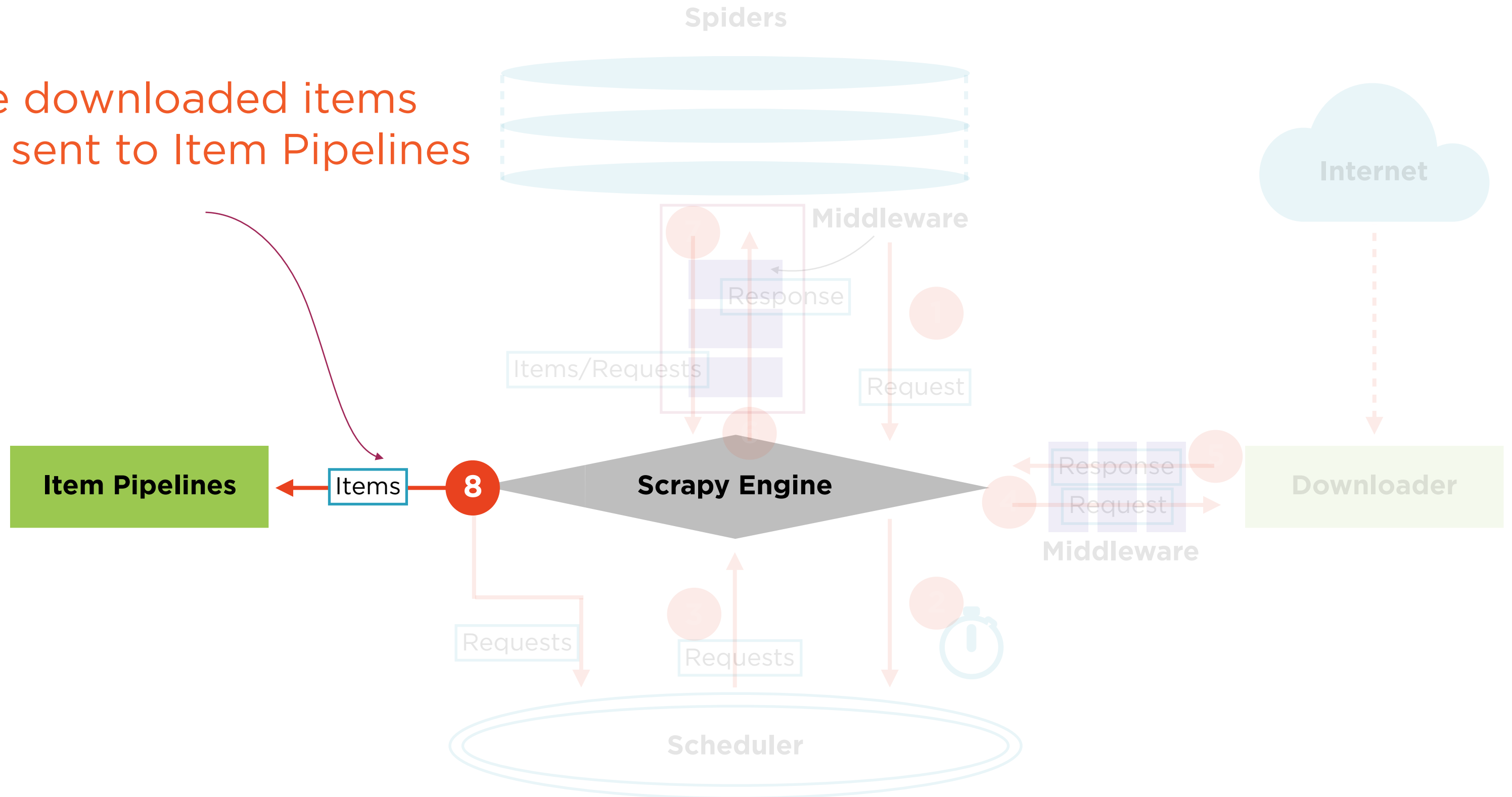
Spiders

The Spider processes the Response and returns scraped items



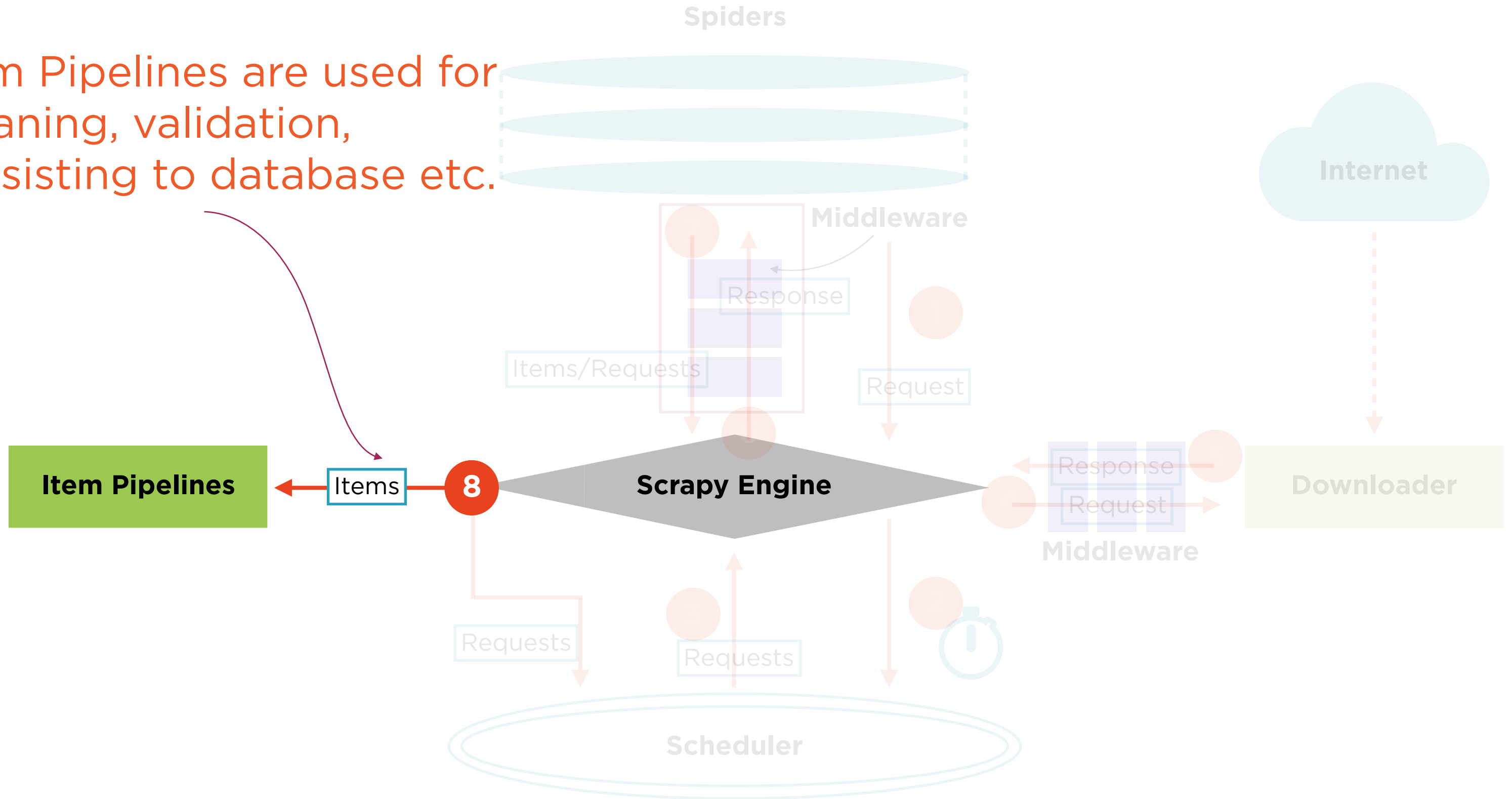
How Scrapy Works

The downloaded items are sent to Item Pipelines



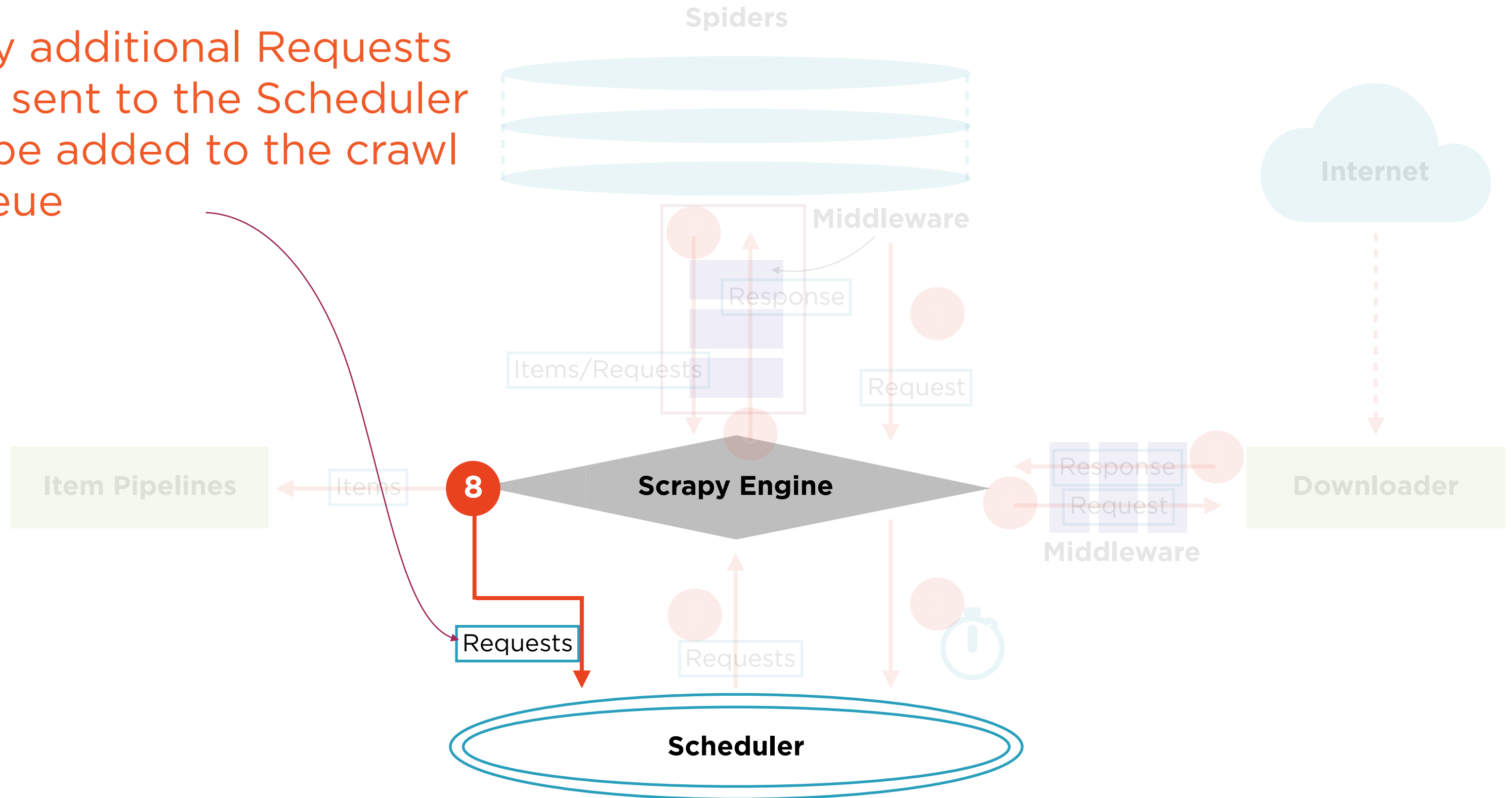
How Scrapy Works

Item Pipelines are used for cleaning, validation, persisting to database etc.



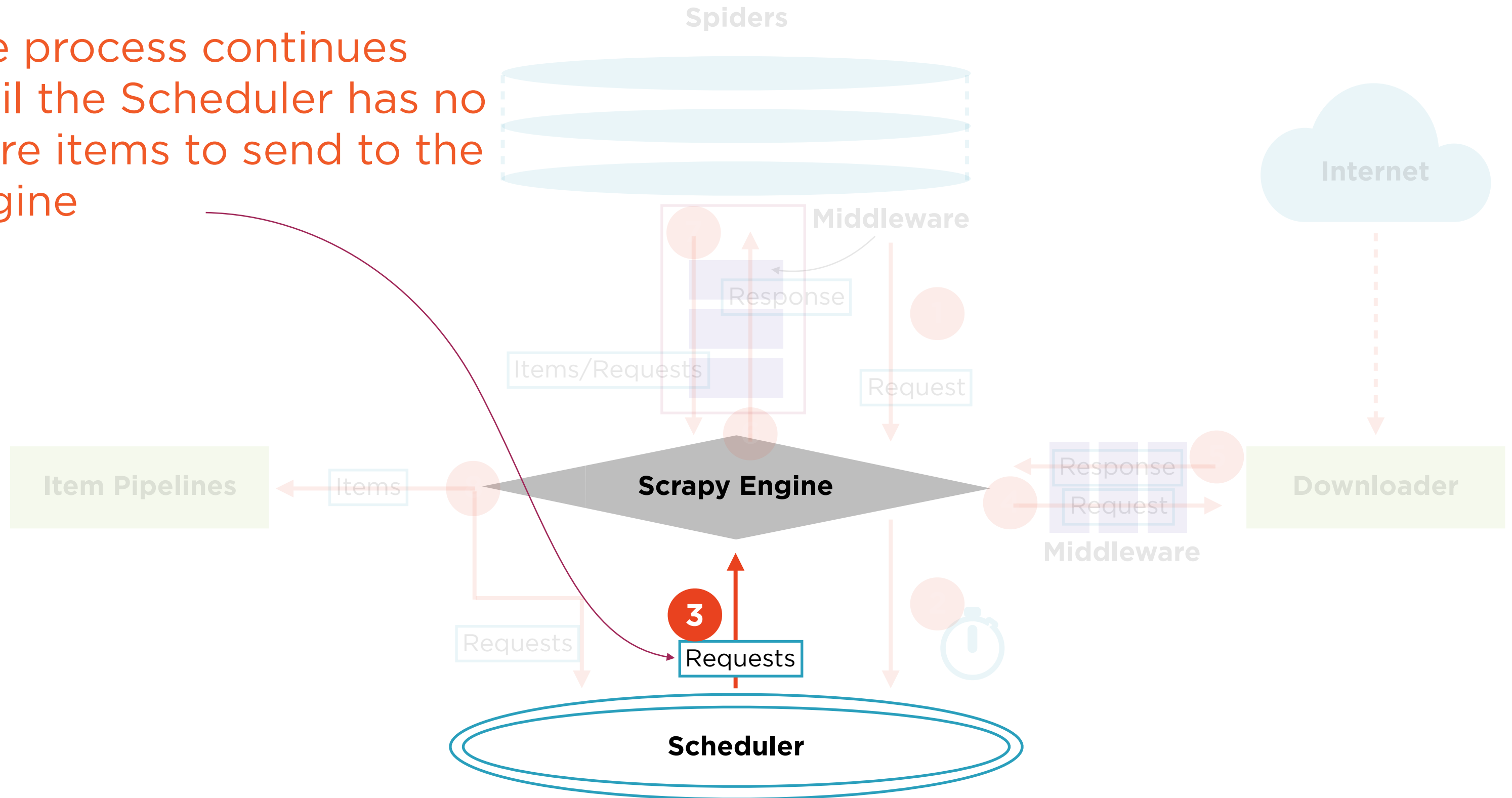
How Scrapy Works

Any additional Requests are sent to the Scheduler to be added to the crawl queue



How Scrapy Works

The process continues until the Scheduler has no more items to send to the Engine



Spiders

Custom classes where you define custom behavior for crawling and parsing pages from a site or group of sites

Implementing Spiders

What to crawl

URLs to start with are in
the `start_requests()`
method

How to crawl

Callback function inputs
web page and outputs
Items, Requests etc.

How to parse

Selectors which
determine which parts of
web page are processed

Demo

Creating a simple custom spider

Demo

Exploring items using the Scrapy shell

Demo

Working with items to store scraped data

Demo

Using item loaders to extract scraped content

Using input and output processors to process content in Scrapy item fields

Demo

Using pipelines to transform scraped data

Summary

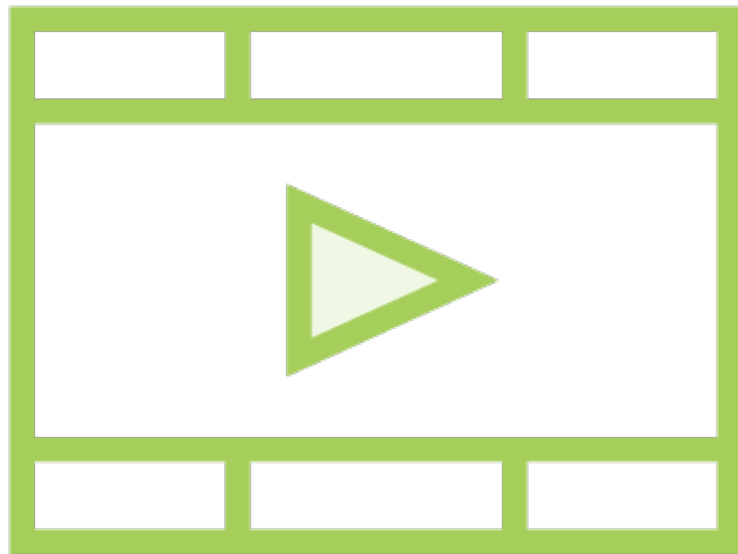
Defining crawling tasks in classes using Spiders

Extracting logical information using items and processors

Using item loaders to automate data extraction

Chaining data transformations using pipelines

Related Courses



Web Scraping: Python Data Playbook

**Extracting Data from HTML with
Beautiful Soup**

**Extracting Structured Data from the
Web Using Scrapy**