Extracting Structured Data from the Web Using Scrapy

GETTING STARTED SCRAPING WEBSITES USING SCRAPY

Janani Ravi

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Scrapy is an application framework for crawling websites

Allows data extraction in a structured format

The Scrapy shell is an interactive shell to quickly test data extraction

Selectors allow you to specify XPath and CSS classes to scrape information

Prerequisites and Course Outline

Prerequisite Courses

Python: Getting Started

Python Fundamentals

Advanced Python

Software and Skills

Be very comfortable programming in Python (Python 3)

Understand some basics of HTML and CSS

Course Outline

Scraping websites

- Scrapy shell, XPath and CSS selectors

Spiders

- Spiders, Items, Item Loaders, Item Pipelines

Built-in services

- Logging, email notifications
- Debugging using the telnet console
- Broad crawls for parallel scraping
- Auto throttling crawls

Crawlers on the Scrapy Cloud

- Deploying a Scrapy project on scrapinghub.com
- Scraping on the cloud using Portia

Introducing Scrapy

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

Originally built for web scraping
but now used for web crawling

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

Scraping vs. Crawling

Web Scraping	Web Crawling
Extract data directly from web sites	Download and index web sites
Data analysis and somewhat unsavory reputation	Performed by all search engines and associated with legitimate use
Specific - “scrape prices from Amazon”	General - “crawl sites linked off Amazon”
Small scale, results in specialized dataset	Large scale, results in document corpus

Originally built for web scraping but now used for web crawling

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

Framework vs. library: inversion of control

https://doc.scrapy.org/en/latest/intro/overview.html

Library vs. Framework

Library	Framework
You call library functions	Framework calls you
You write the application and invoke library for specific portions	Framework defines the application and invokes your code for specific portions

Hollywood Principle: Don’t call us, we’ll call you

This is a defining characteristic of frameworks

Framework vs. library: inversion of control

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

You must know what you are looking for - tied to HTML format

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

Inherently somewhat fragile, like regular expressions and other related tools

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

Specific HTML elements are selected for processing using Selectors

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

Scrapy supports selectors specified in CSS and XPath

Scrapy

Scrapy is an application framework for crawling web sites and extracting structured data

https://doc.scrapy.org/en/latest/intro/overview.html

Benefits of Scrapy

Asynchronous Callbacks

Requests and callbacks are scheduled and processed asynchronously

Granular Control

Settings to govern politeness of crawl, error handling etc.

Asynchronous Callbacks

Speed

Parallelism

Fault-tolerance

Granular Control

Download delays between requests

Limit on concurrent connections

- Per IP
- Per domain

Auto-throttling extension

Demo

Install and set up Scrapy on your local machine

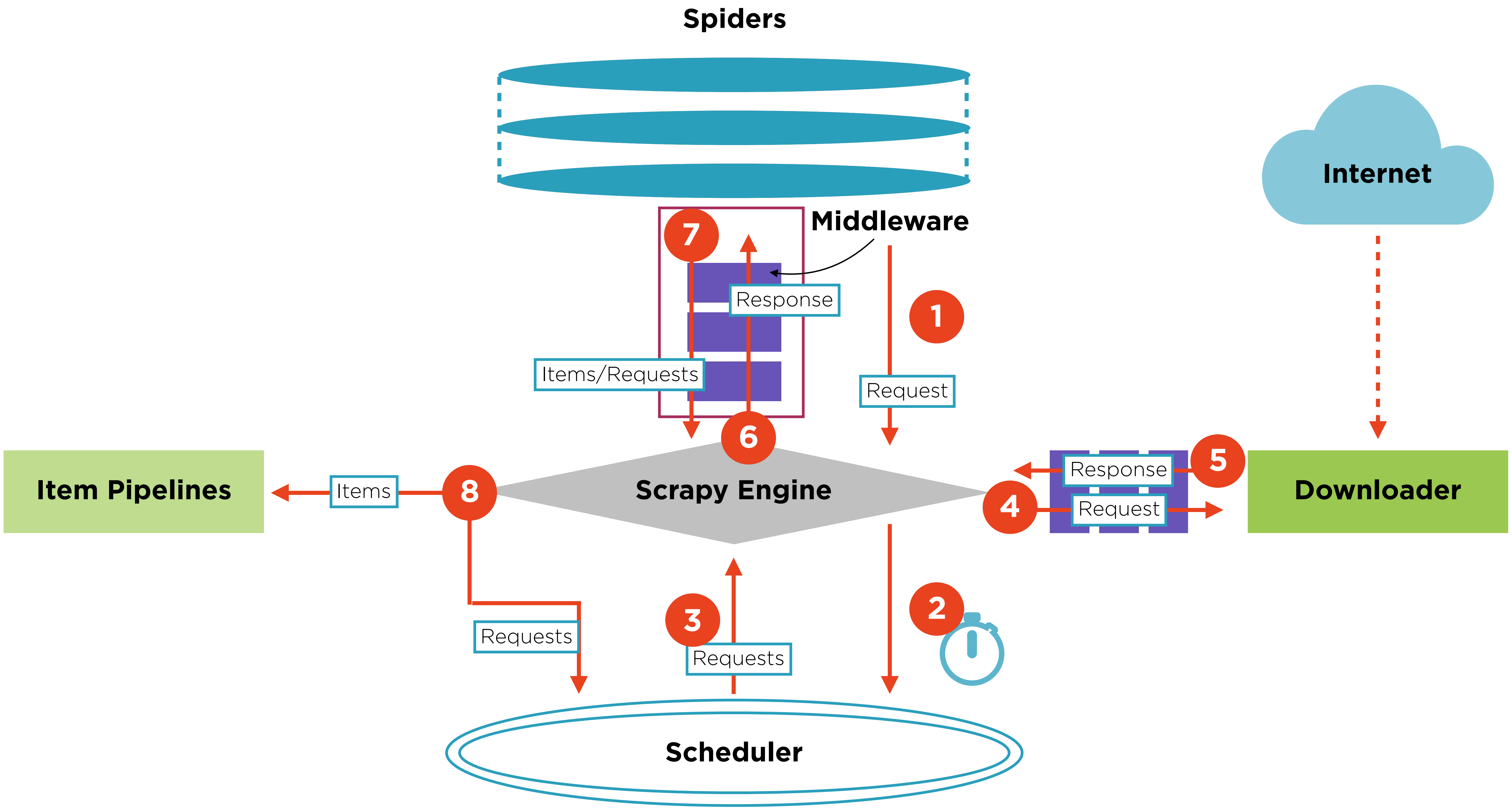
Basic introduction to Scrapy components

Demo

Introducing the Scrapy shell

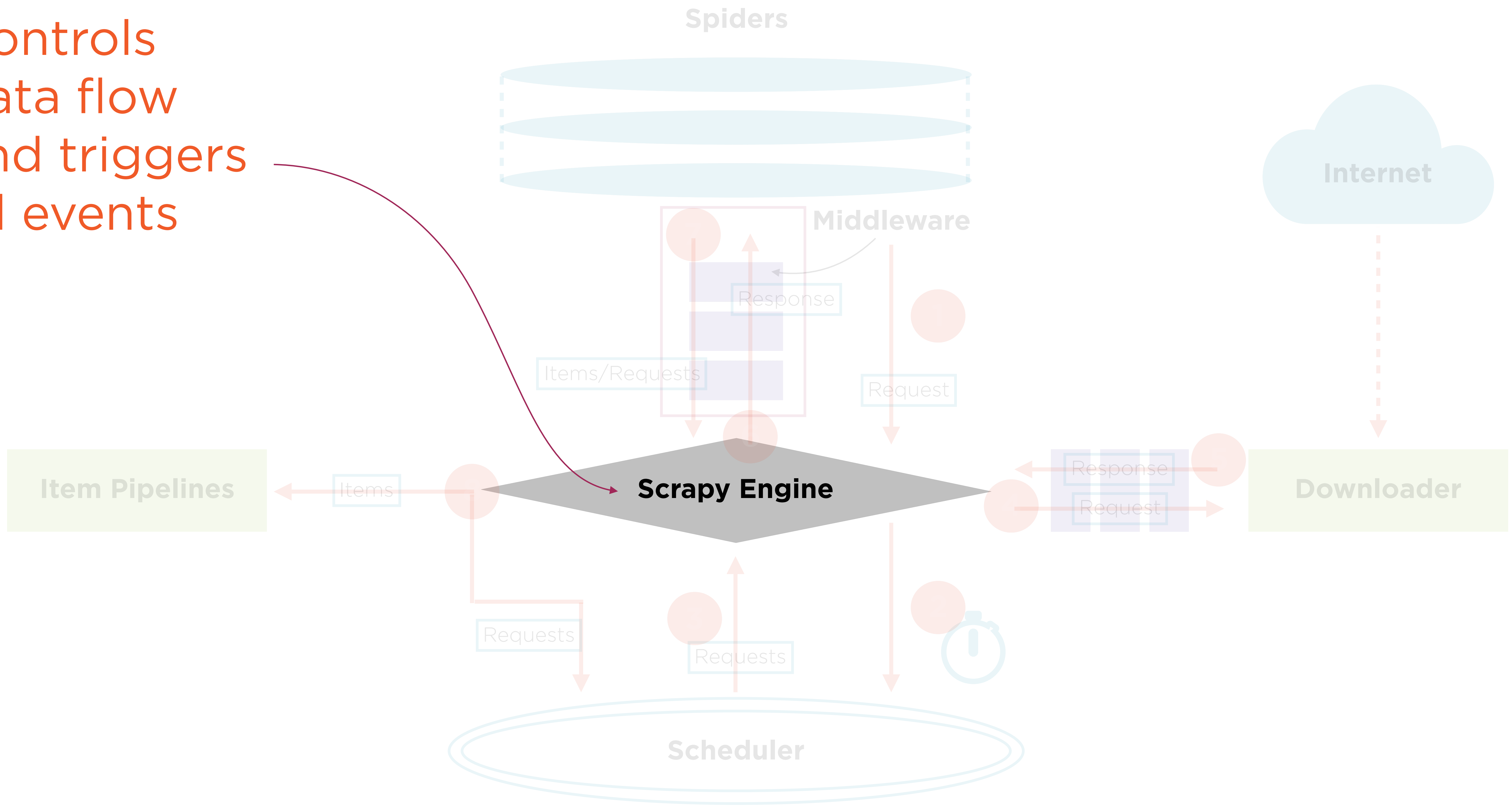
How Scrapy Works

How Scrapy Works



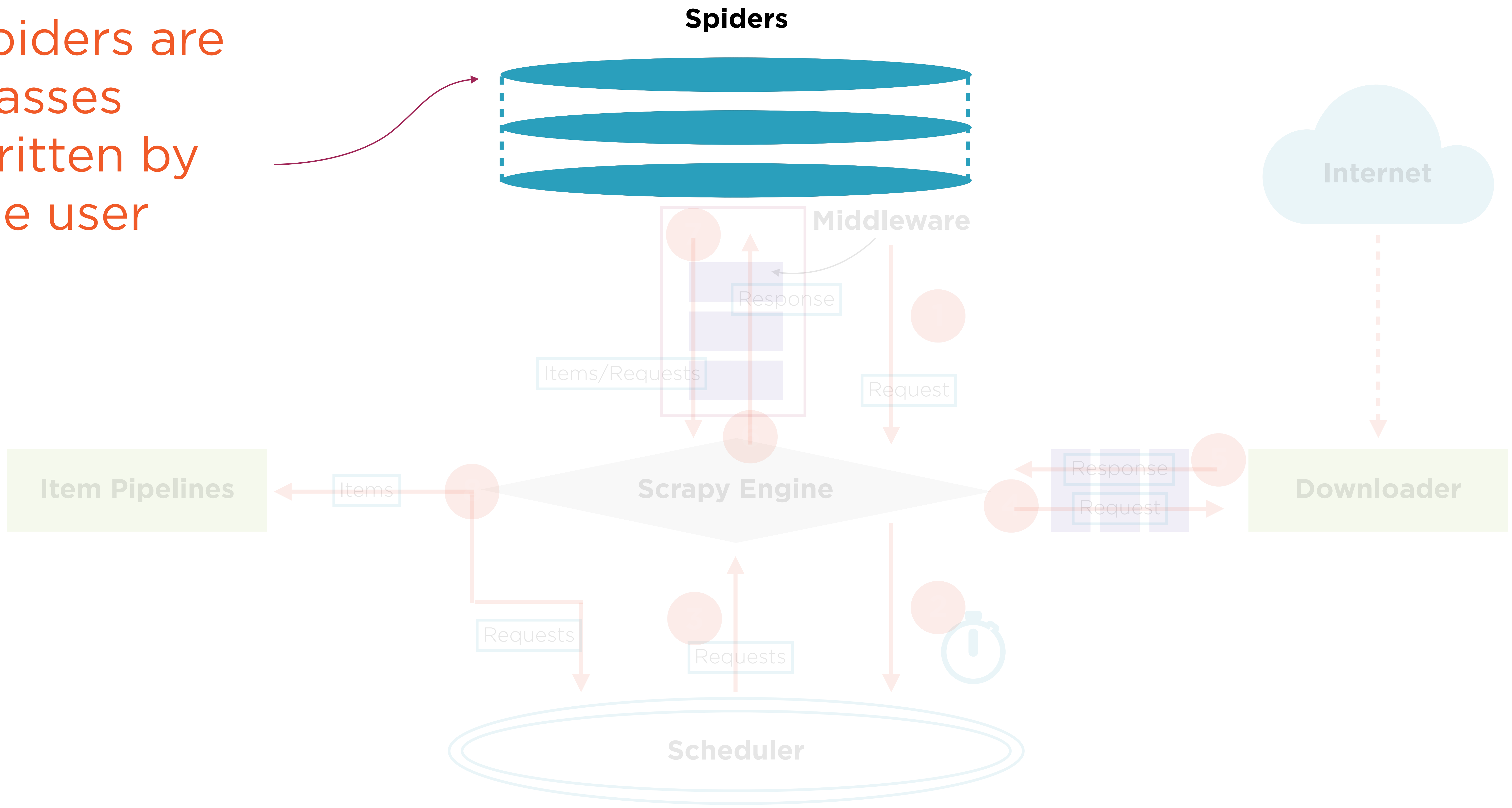
How Scrapy Works

Controls
data flow
and triggers
all events



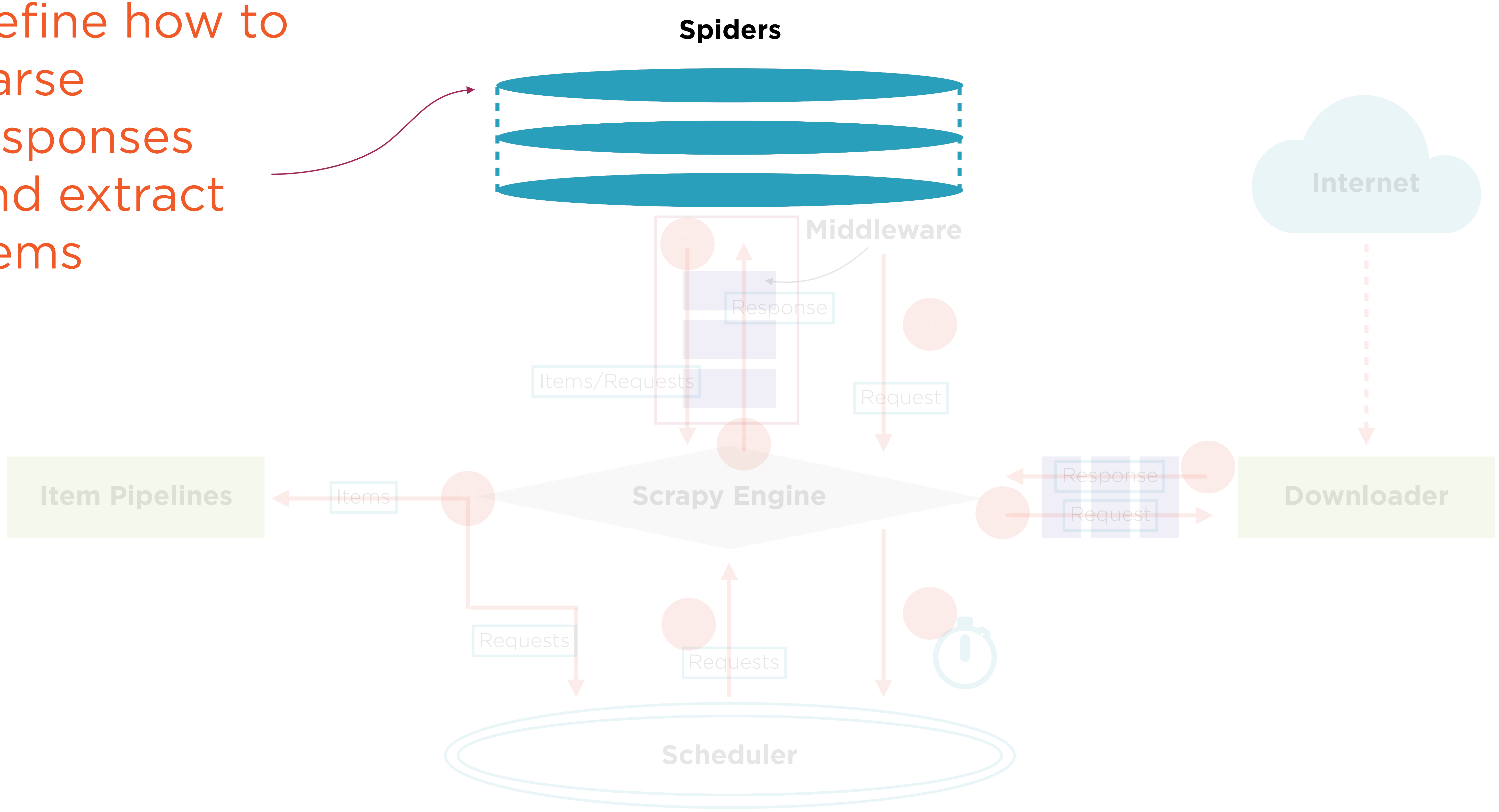
How Scrapy Works

Spiders are classes written by the user



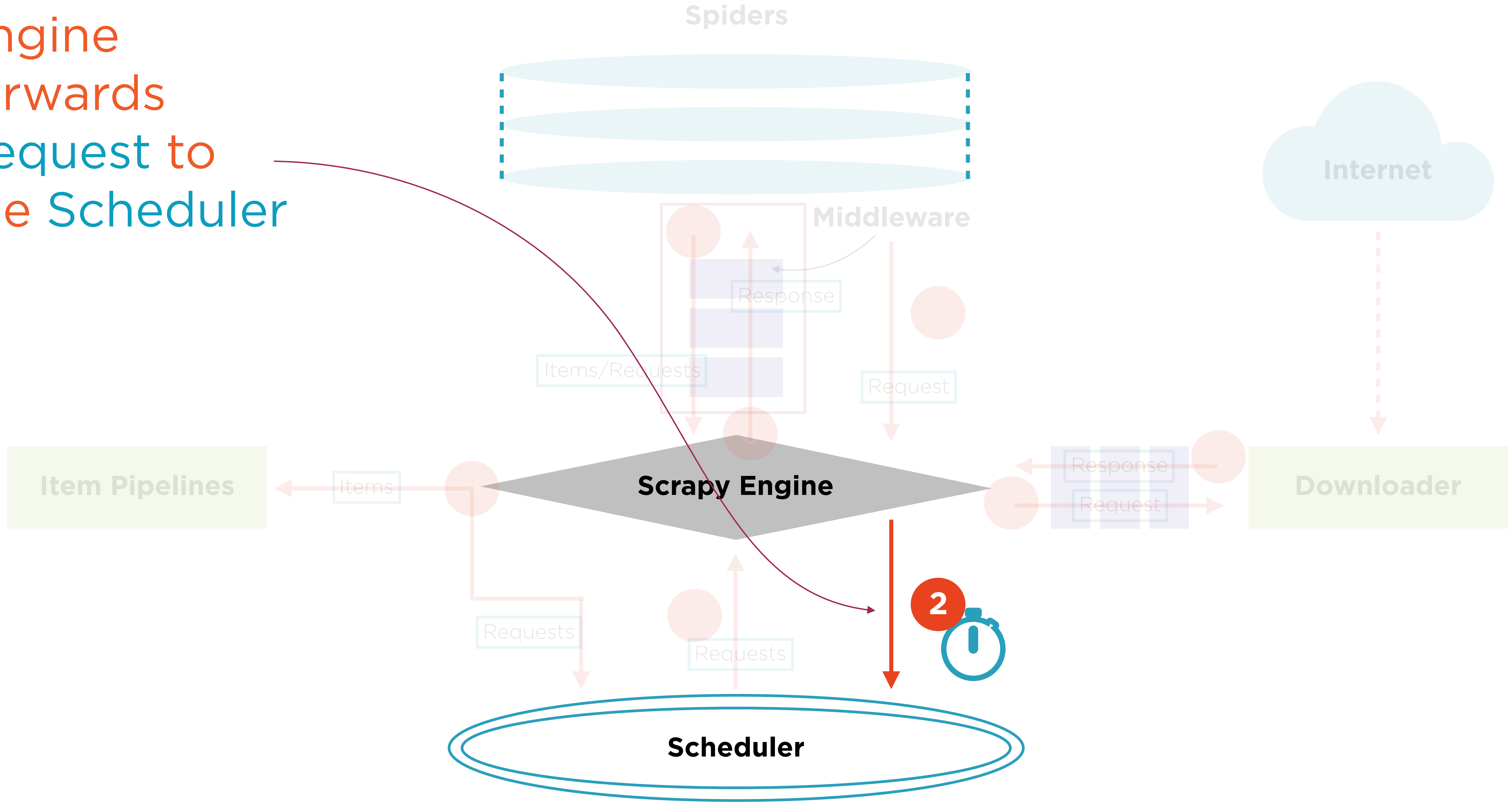
How Scrapy Works

Define how to parse responses and extract items



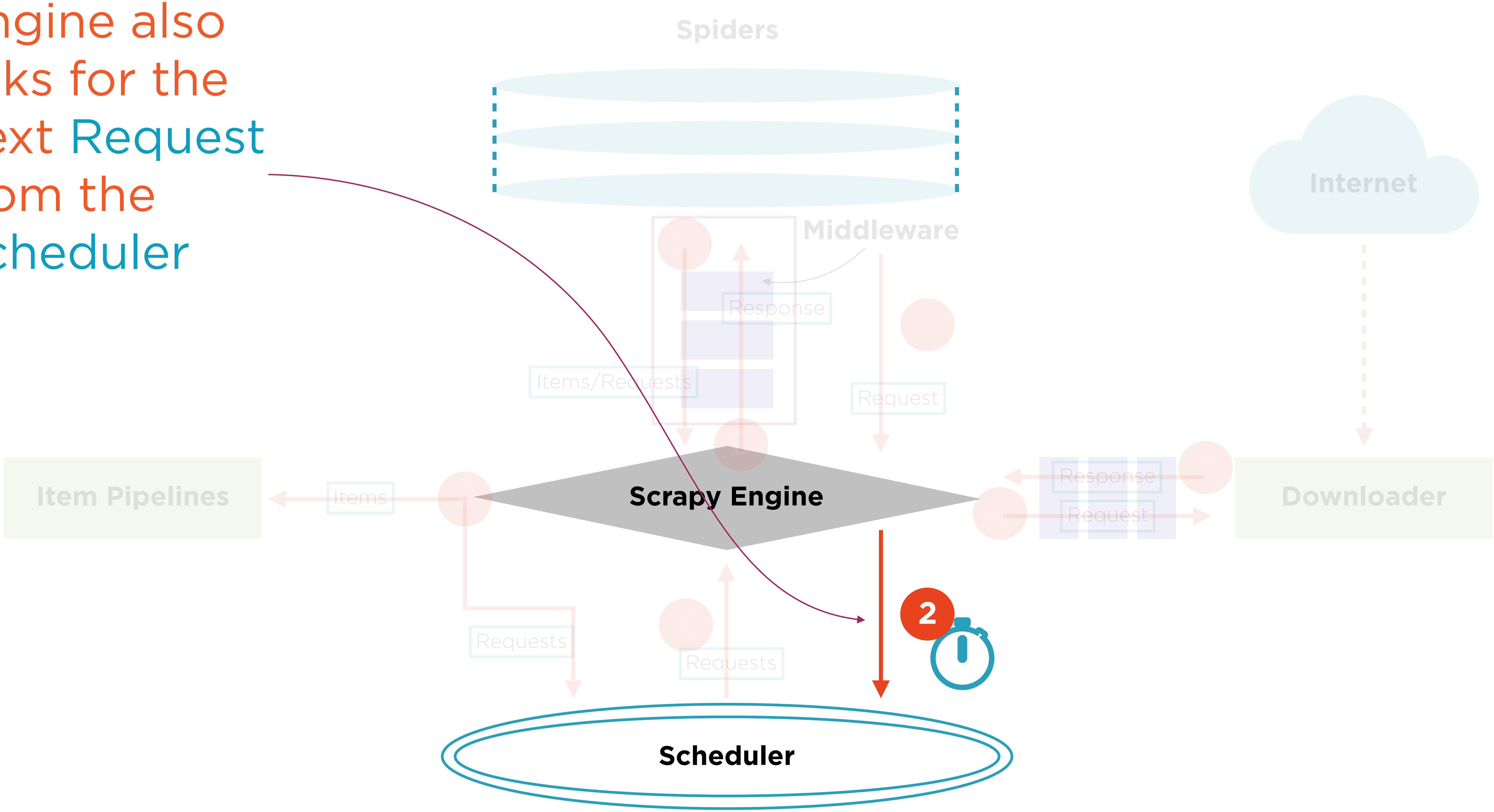
How Scrapy Works

Engine forwards Request to the Scheduler



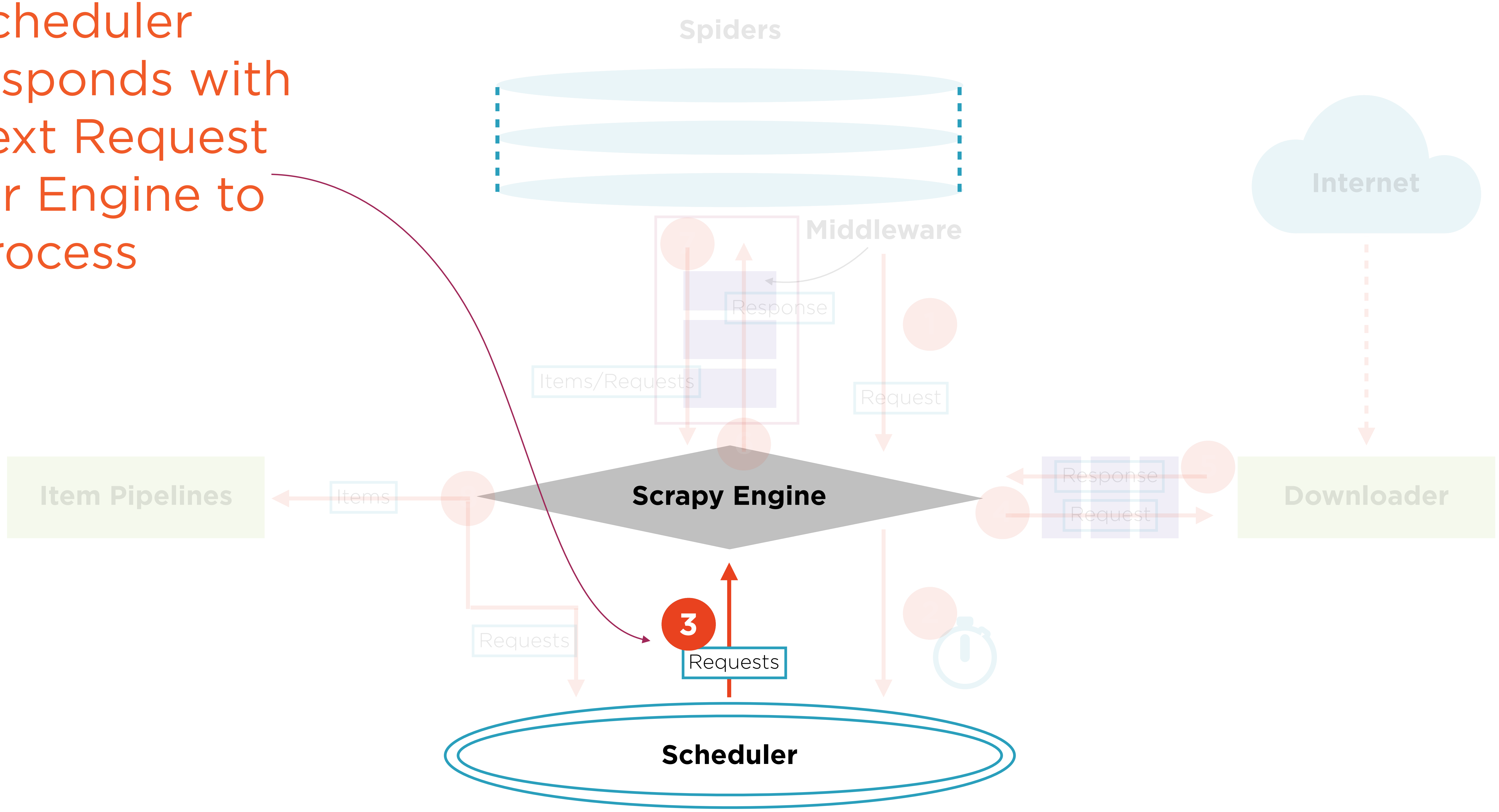
How Scrapy Works

Engine also asks for the next Request from the Scheduler



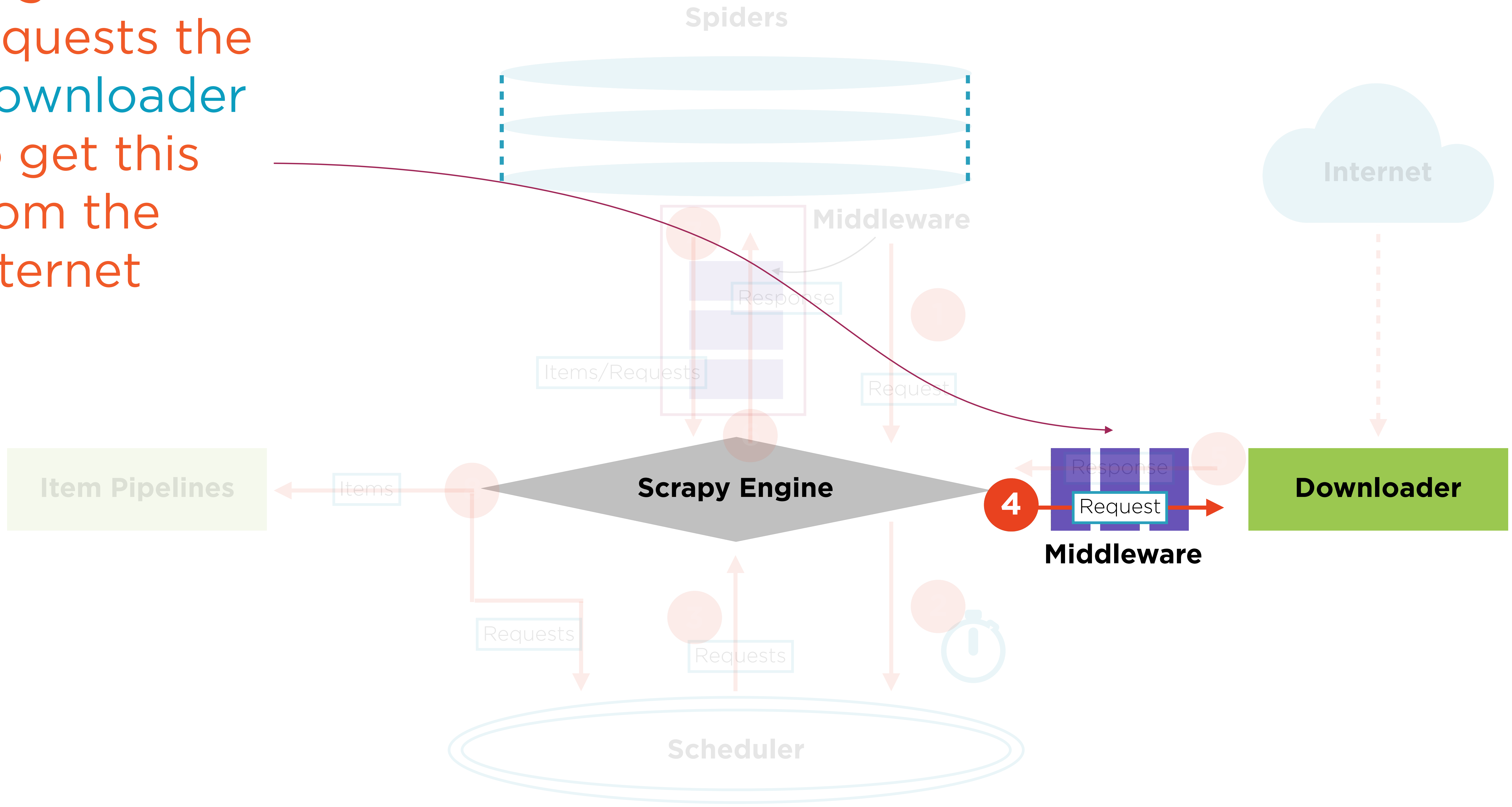
How Scrapy Works

Scheduler responds with next Request for Engine to process



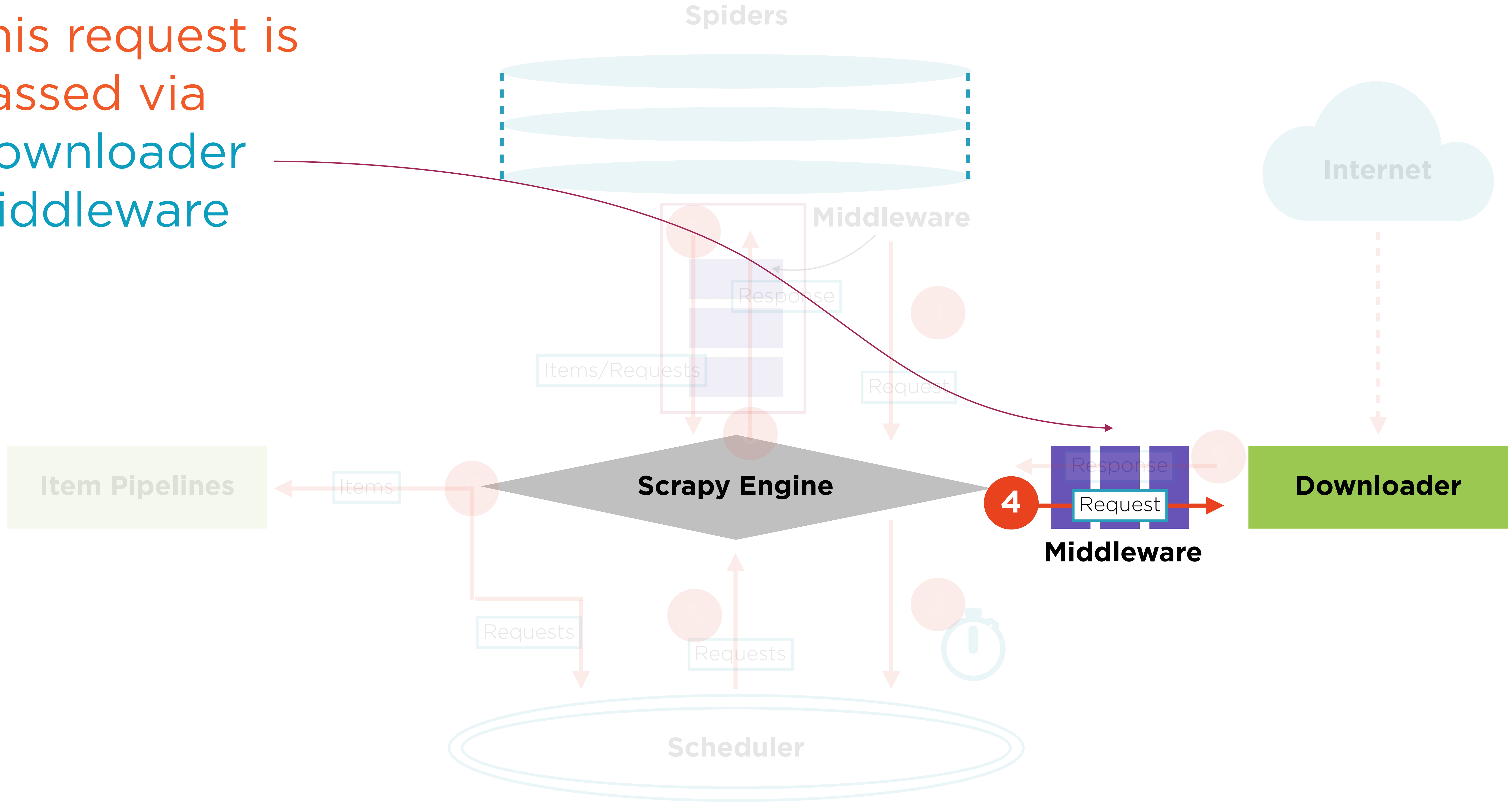
How Scrapy Works

Engine requests the Downloader to get this from the internet



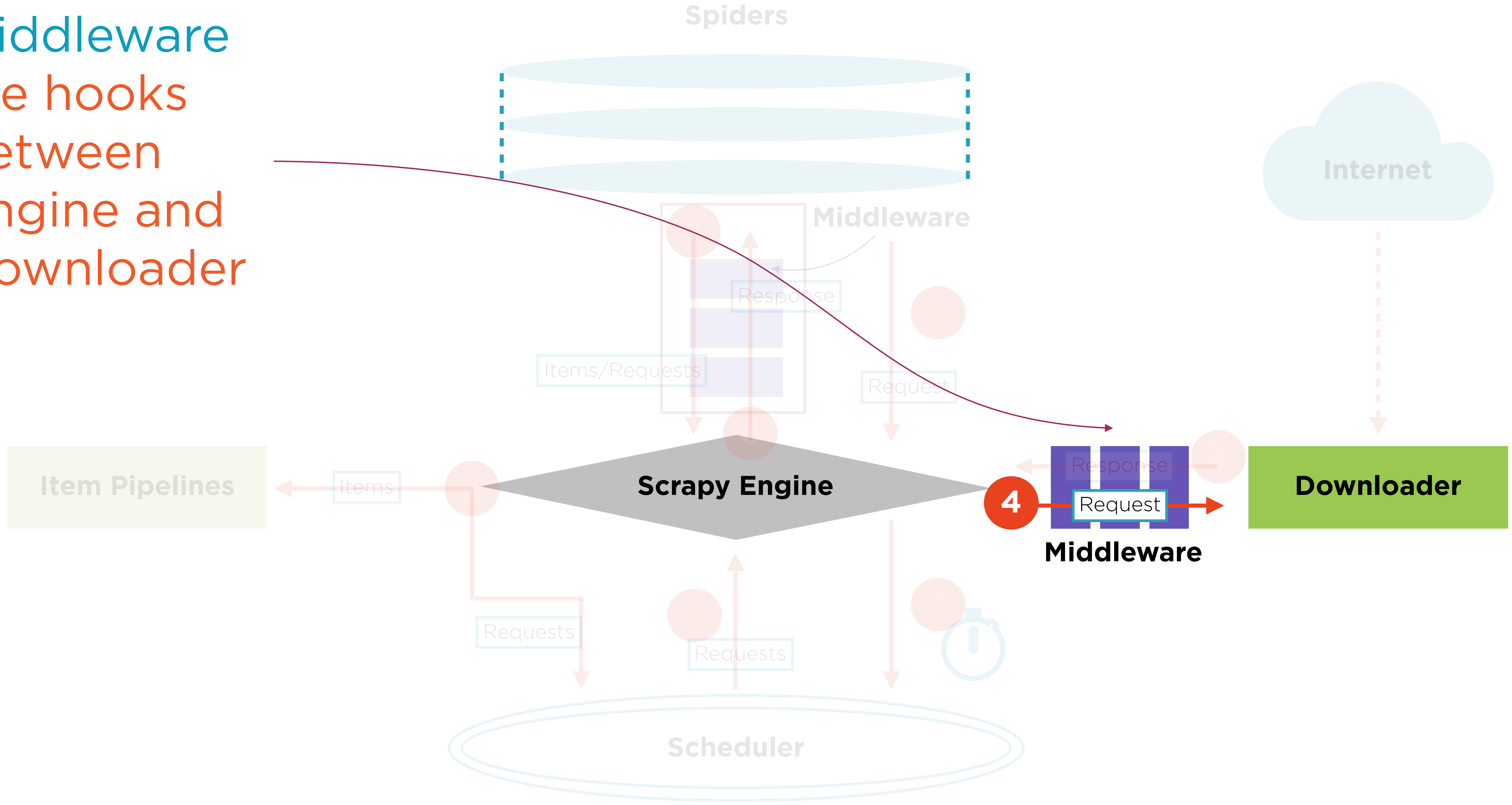
How Scrapy Works

This request is passed via Downloader Middleware



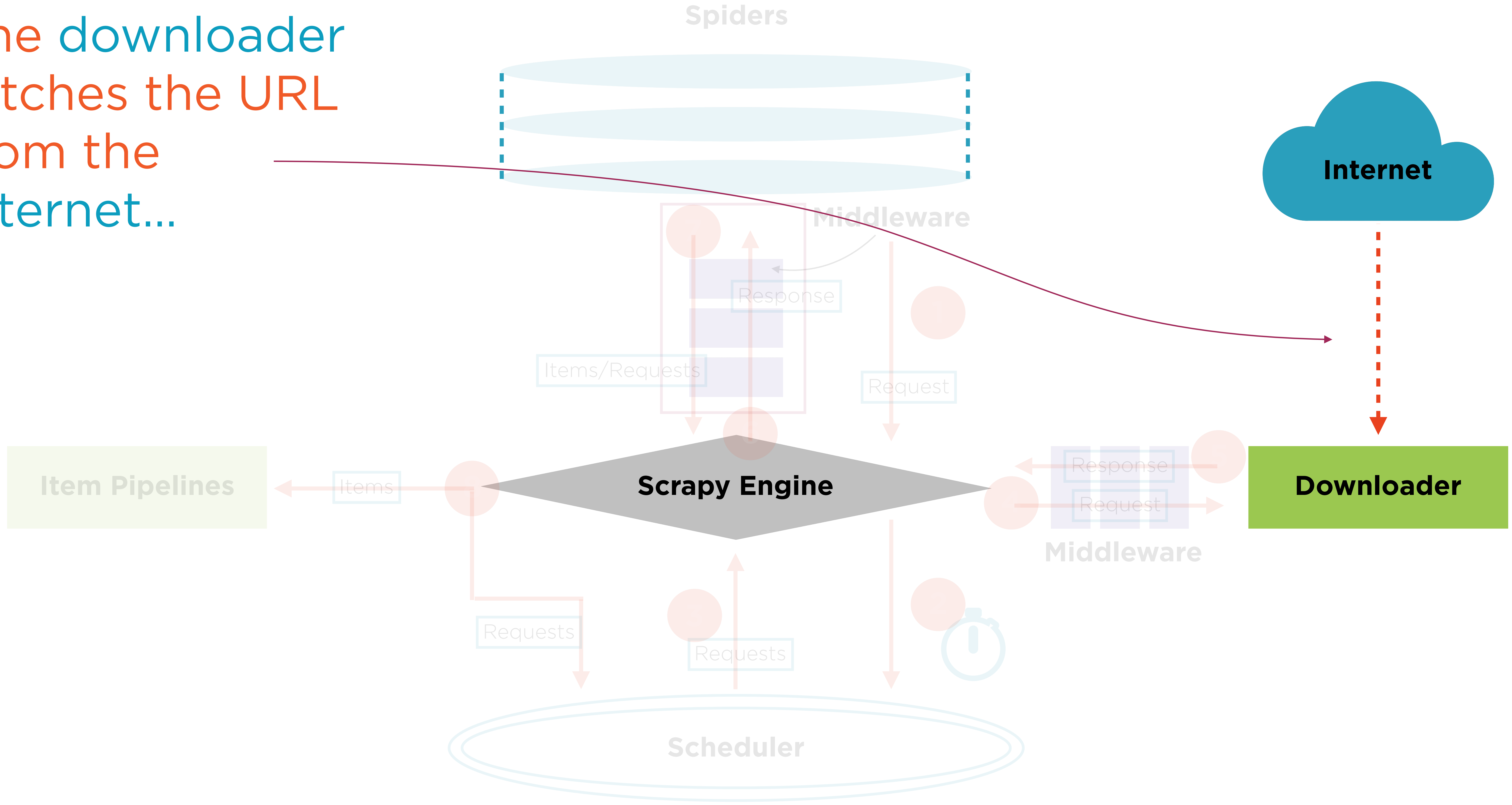
How Scrapy Works

Downloader Middleware are hooks between Engine and Downloader



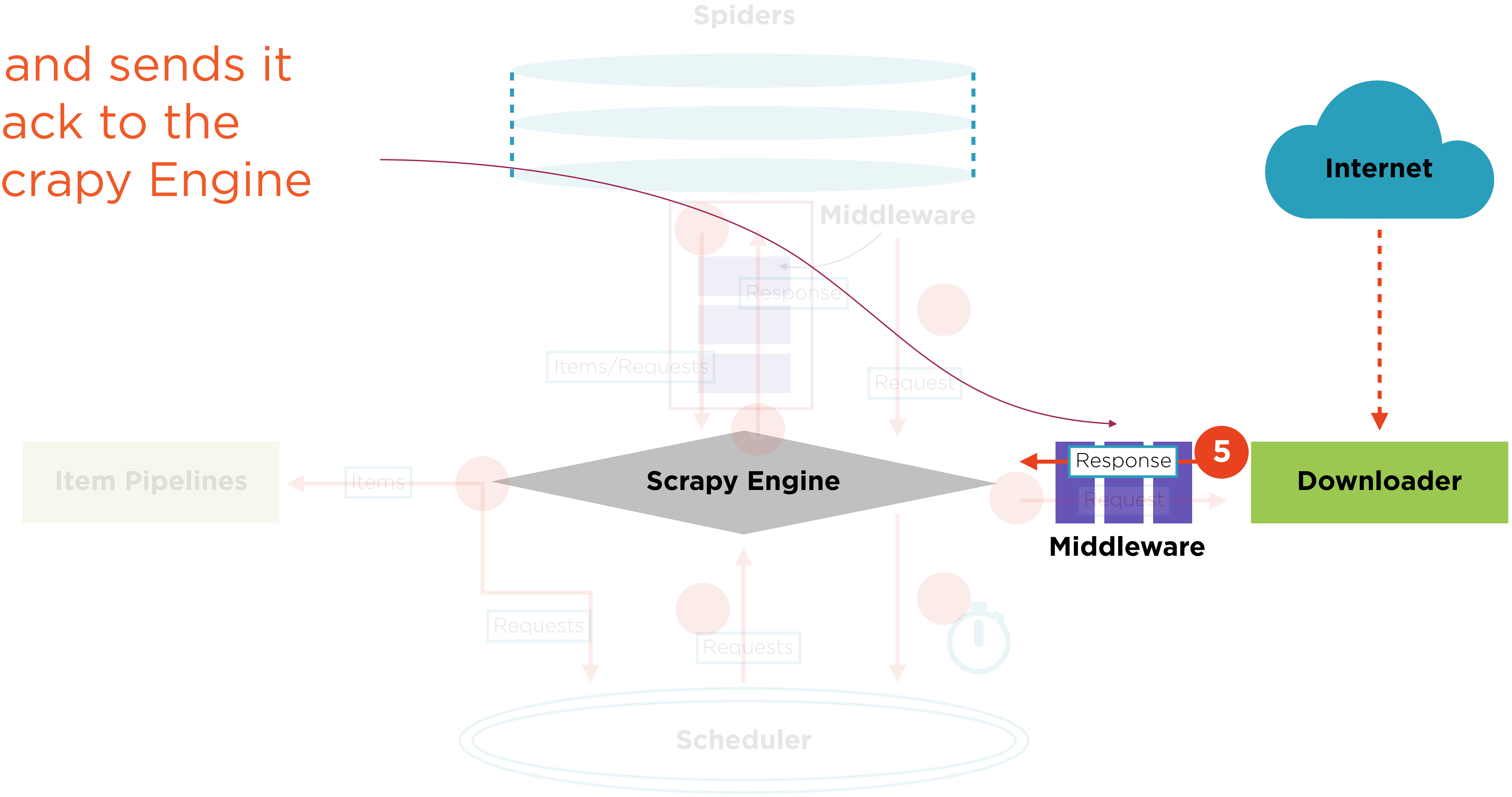
How Scrapy Works

The downloader fetches the URL from the internet…



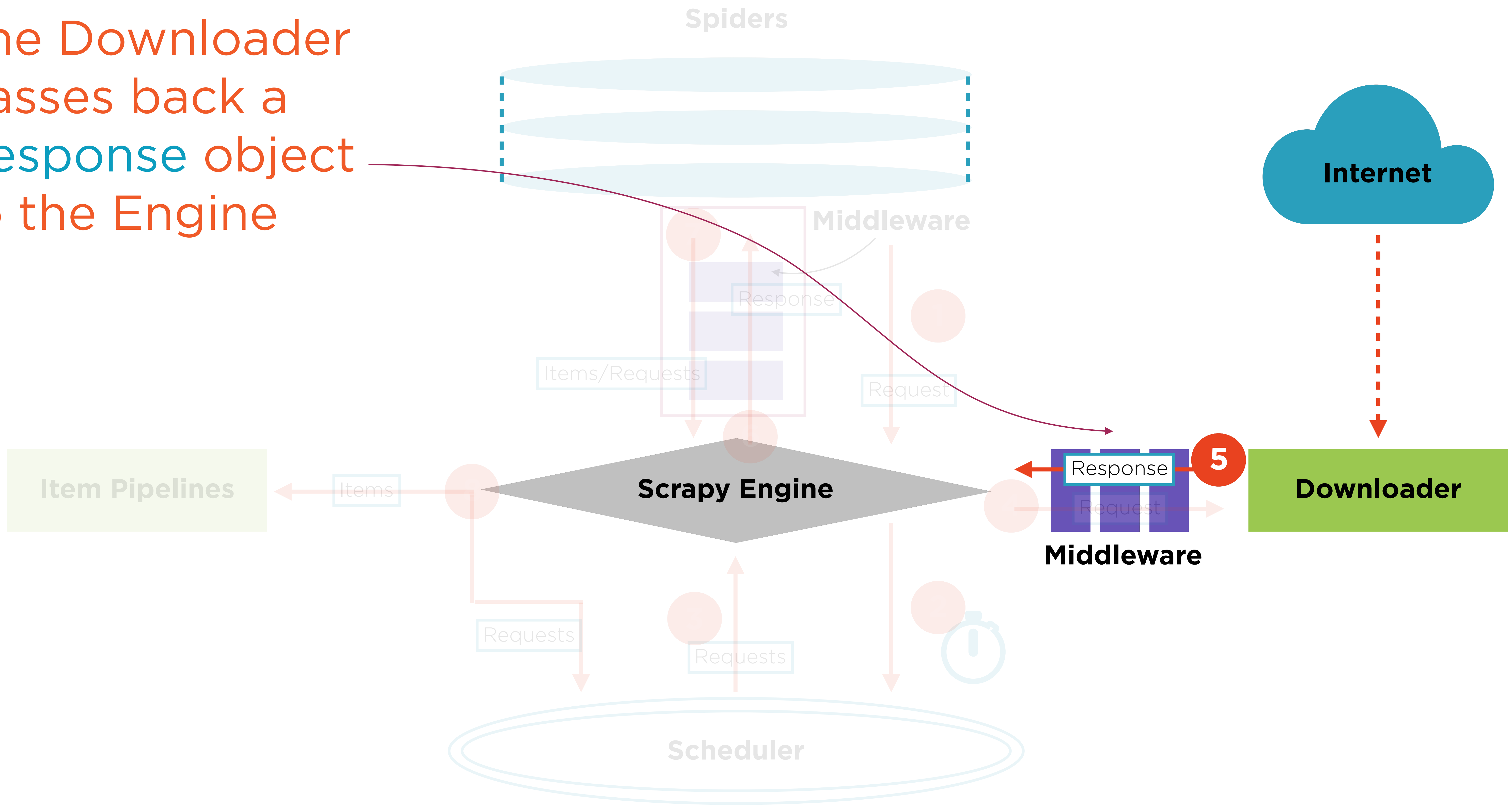
How Scrapy Works

...and sends it back to the Scrapy Engine



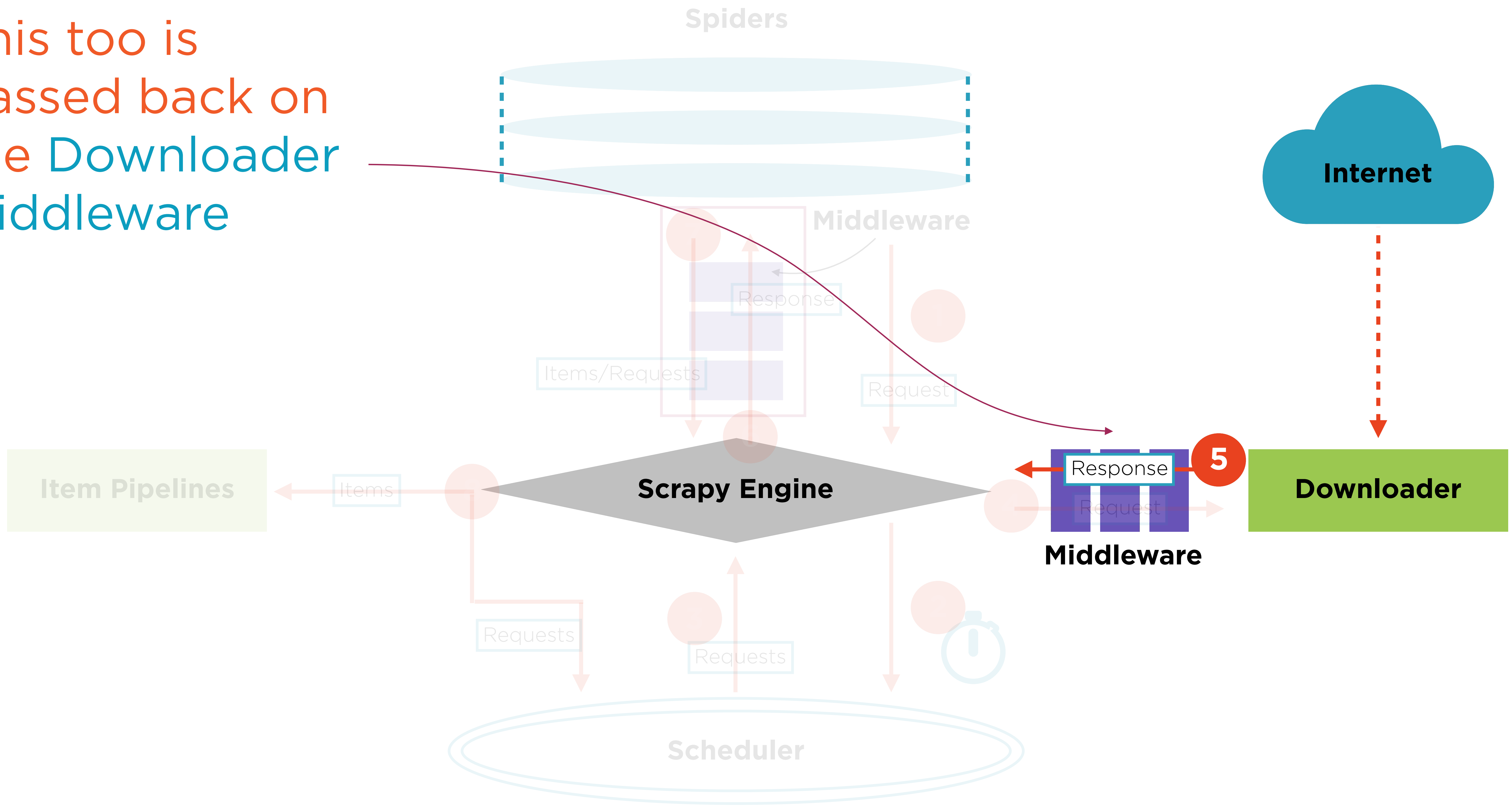
How Scrapy Works

The Downloader passes back a Response object to the Engine



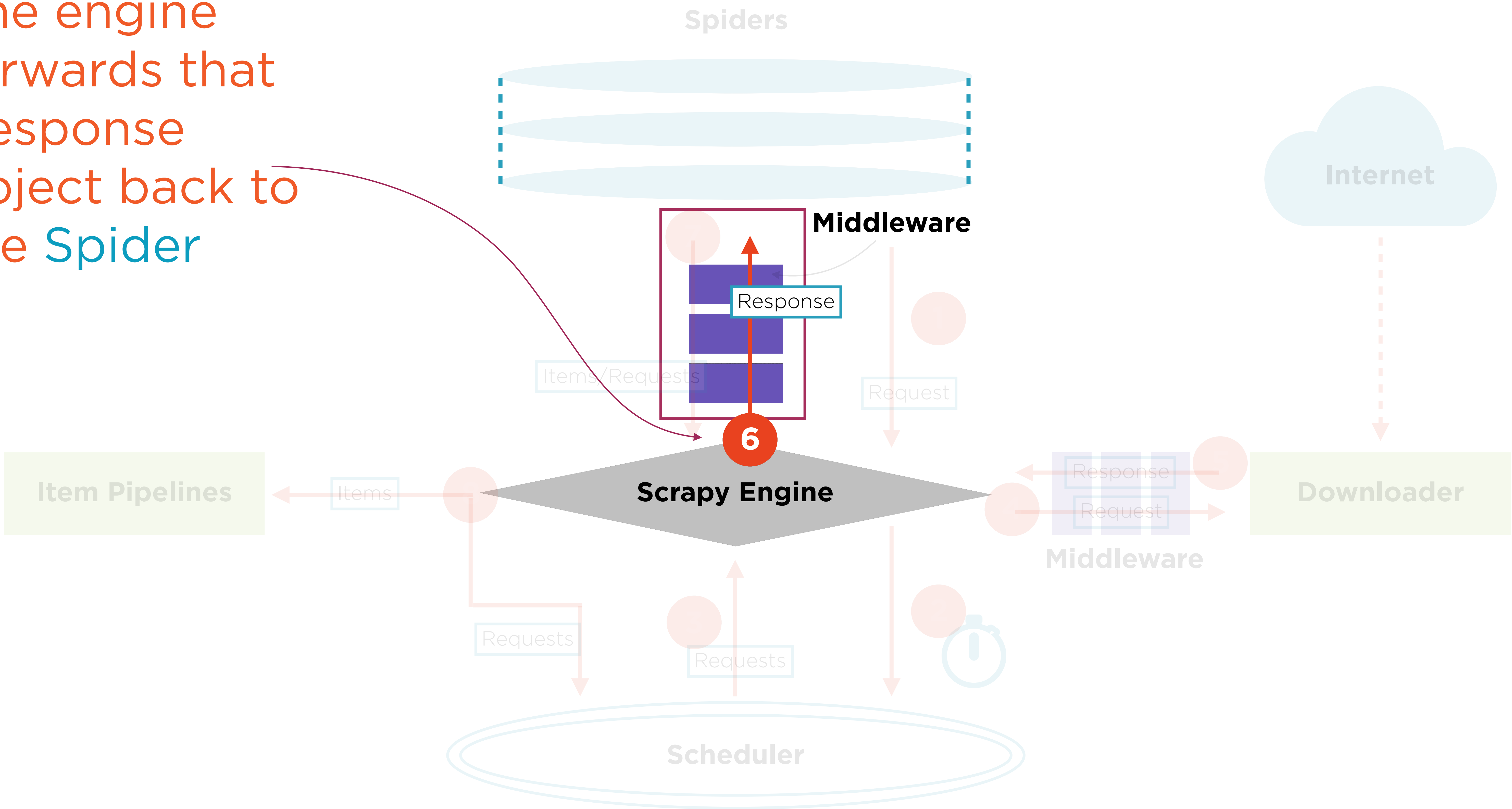
How Scrapy Works

This too is passed back on the Downloader Middleware



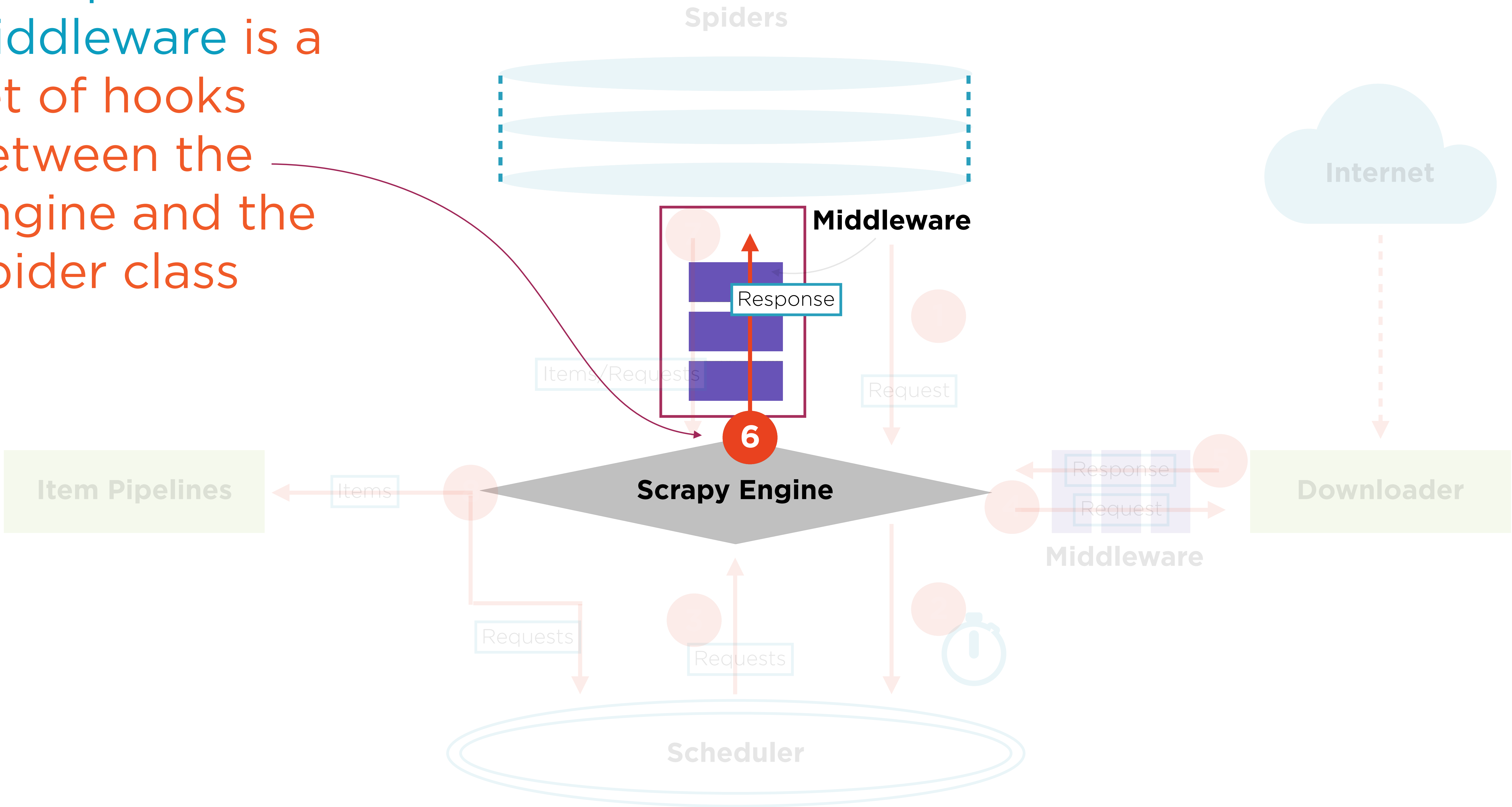
How Scrapy Works

The engine forwards that Response object back to the Spider



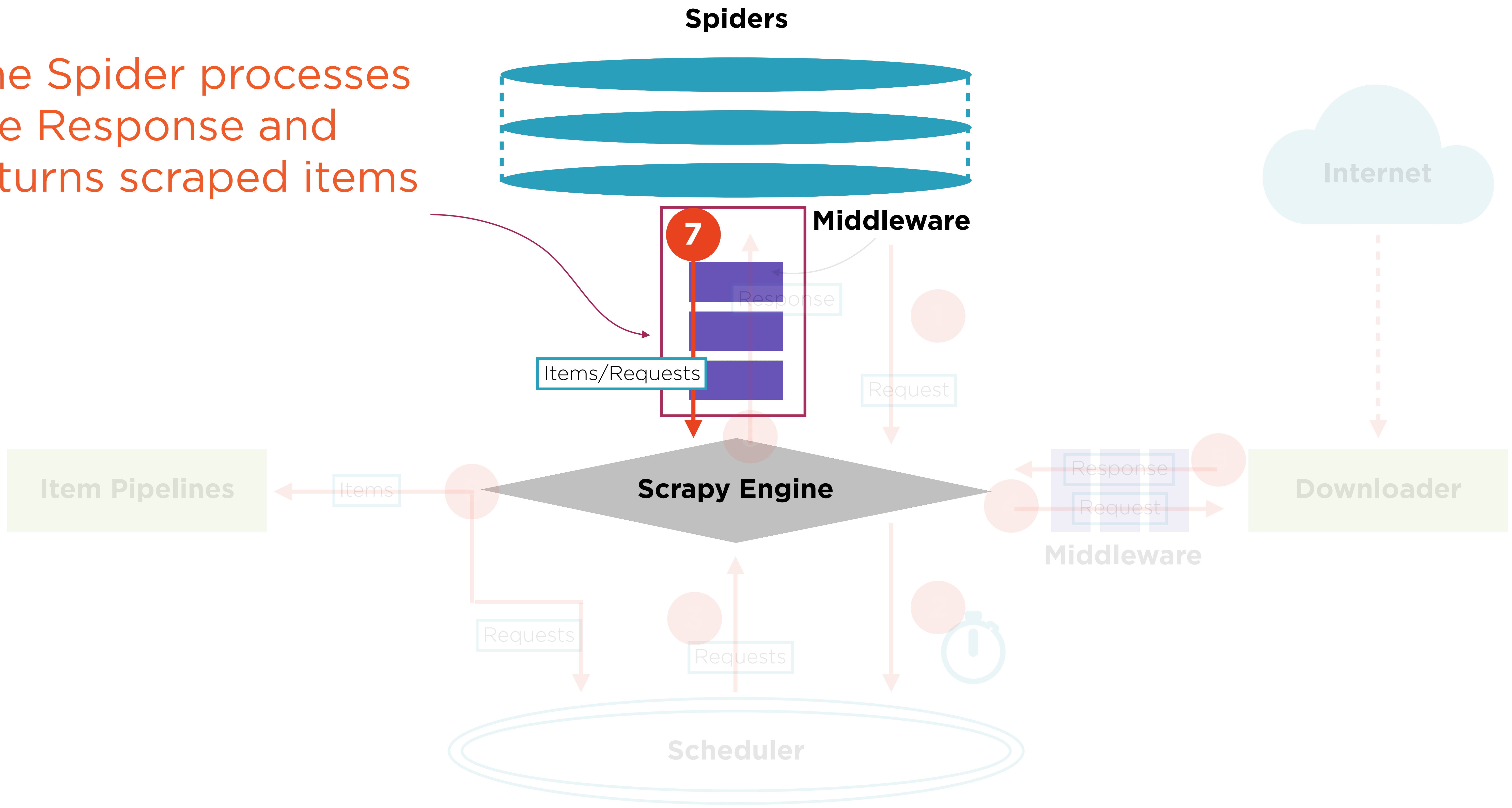
How Scrapy Works

The Spider Middleware is a set of hooks between the Engine and the Spider class



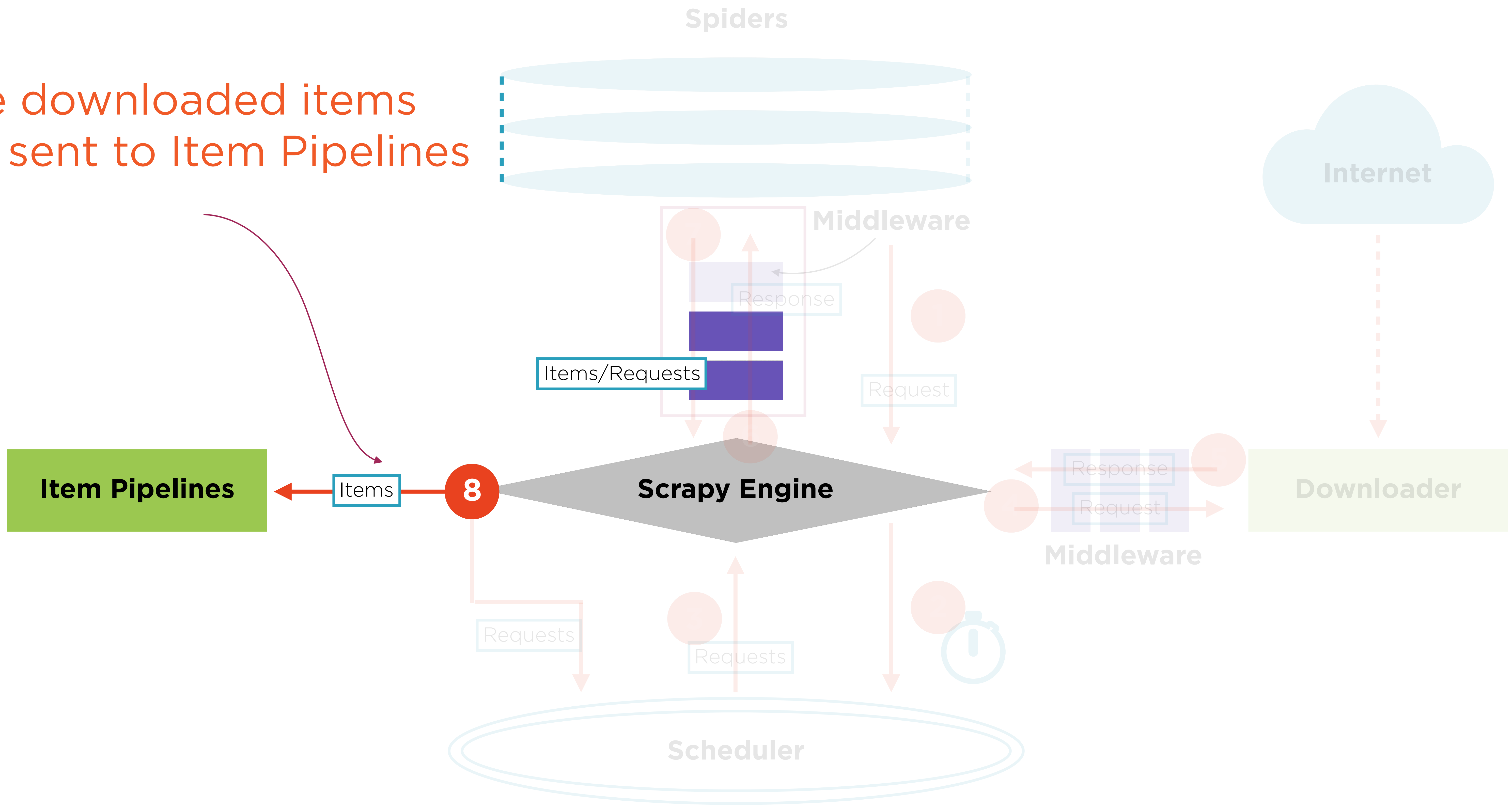
How Scrapy Works

The Spider processes the Response and returns scraped items



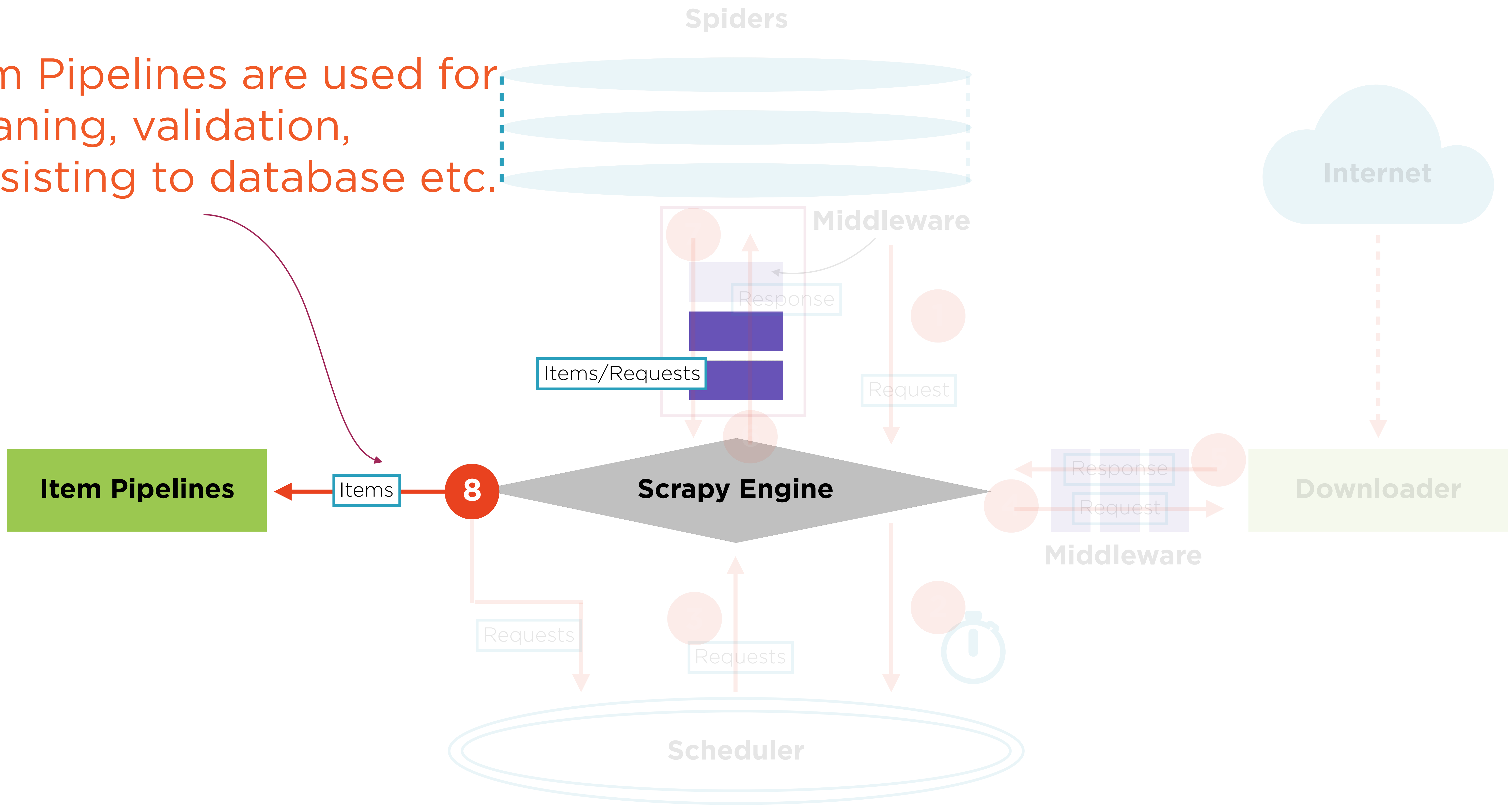
How Scrapy Works

The downloaded items are sent to Item Pipelines



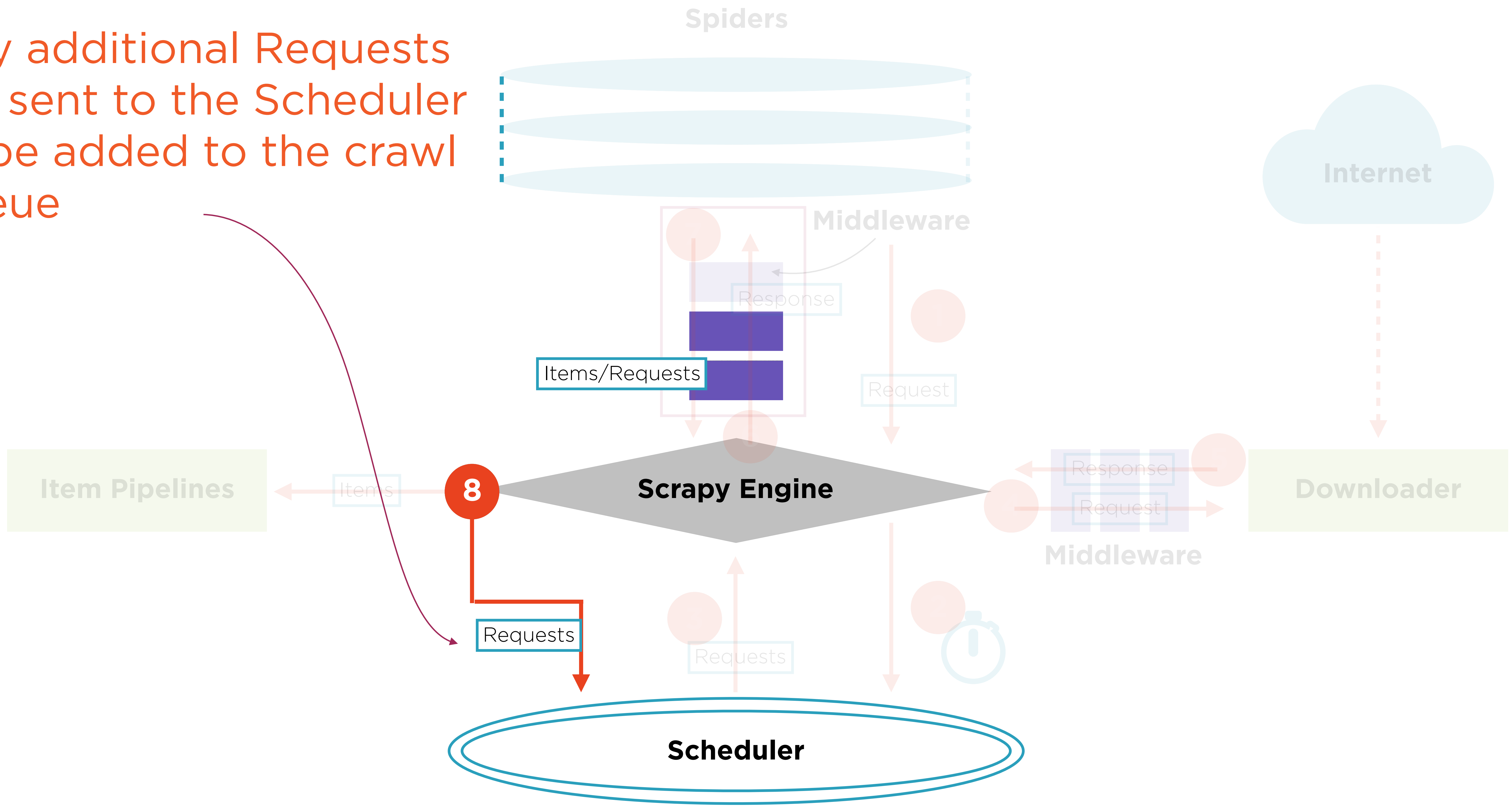
How Scrapy Works

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



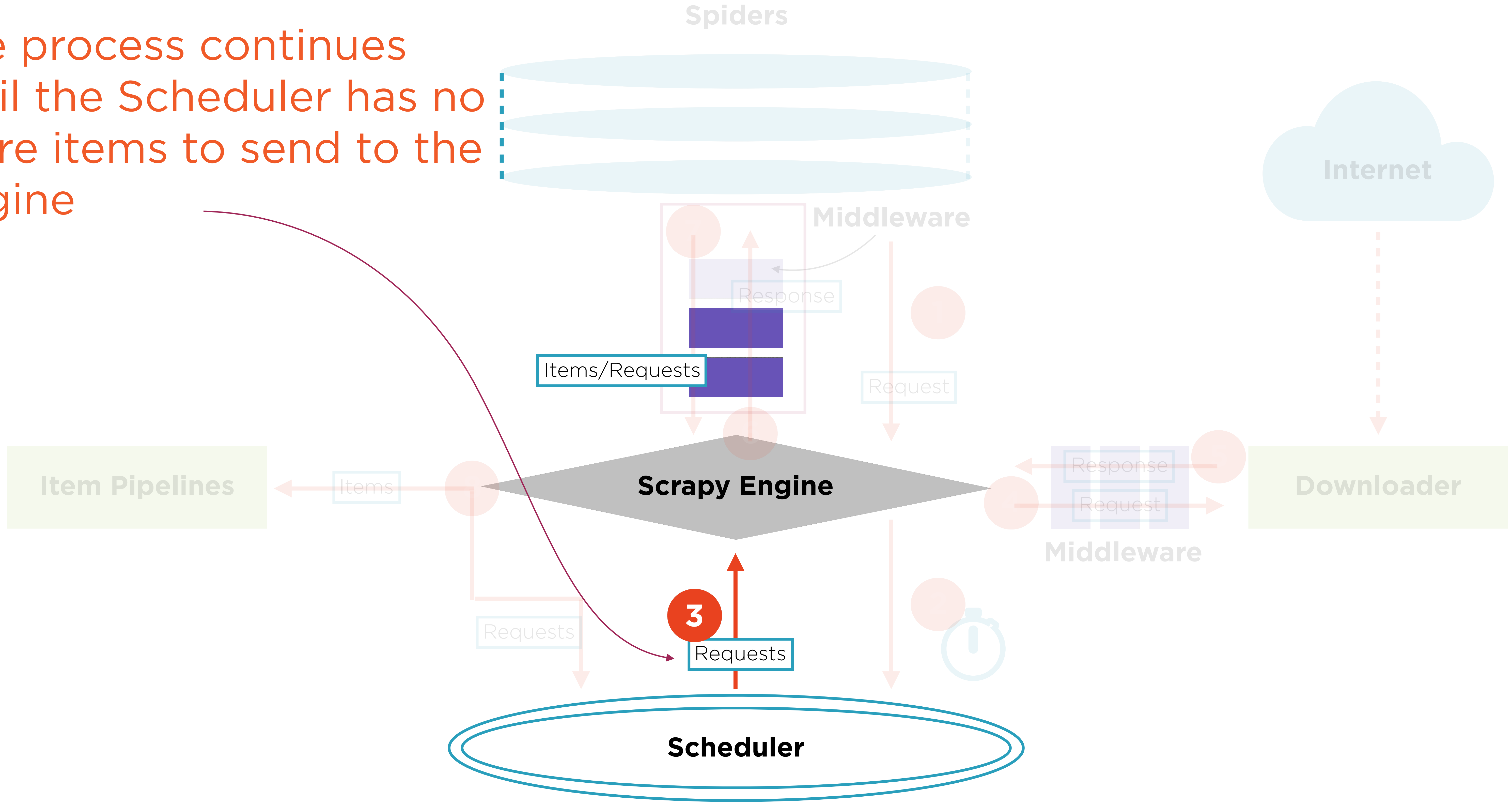
How Scrapy Works

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



Demo

Working with Selectors using XPath and CSS classes

Selector

Specification of what HTML elements ought to be selected for processing. Scrapy supports XPath and CSS selectors.

Scrapy Selectors

XPath

Select nodes in an XML (or HTML) document

CSS

Select HTML elements (usually to associate styles with them)

Scrapy selectors are built atop the lxml library

Demo

Using regular expressions with Selectors

Summary

Scrapy is an application framework for crawling websites to extract structured data

The Scrapy shell is an interactive shell to quickly test data extraction

Selectors allow you to specify XPath and CSS classes to scrape information