oxylabs®

Real-Time Price Monitoring System Architecture

An action chain for eCommerce businesses

Price intelligence at a glance

Price is a crucial factor behind a customer's buying decisions. Is it too high (from the buyer’s perspective), or is it too low (from the seller's perspective)? Maybe there is a better deal at a different retailer, or perhaps a similar, more modestly priced product could satisfy the needs just as well.

Retailers and brands are compelled to be ultra-competitive with prices to maintain their products or services in the spotlight. Prices can change every minute or second to avoid falling under the wrong value and disrupting sales of an otherwise excellent offering.

A correct price must provide a healthy profit margin and return on investment that keeps the business running. Beyond the necessities, prices determine a brand's reputation – the perception of superiority or inferiority compared to rivals determined by the premium (luxury), mid-range, and budget segments.

Prices must generate profit and act as a catalyst to sales. This is where price intelligence comes into play.

What is Price Intelligence?

Price intelligence is the process of tracking, monitoring, and analyzing competition and market data to make informed and well-motivated pricing decisions.

Businesses apply competitive price intelligence to understand the market, optimize pricing strategies, preserve margins, and increase profits. It is achieved by monitoring pricing data to keep track of the ever-changing marketplace and stay ahead of the competition.

Price intelligence consists of three principal steps:

1. Identifying the competition
2. Acquiring pricing data
3. Analyzing the data

Price intelligence depends on automation and public web data extraction to complete these steps. The second step, in particular, comprises the technical aspects of price monitoring – locating, extracting, and structuring large real-time data sets.

Price monitoring explained

Price monitoring is aimed at creating awareness for trends and patterns in competitors' pricing strategies. Such knowledge should help create tactics directed towards making offerings more desirable.

Price monitoring helps detect, understand, and conduct:

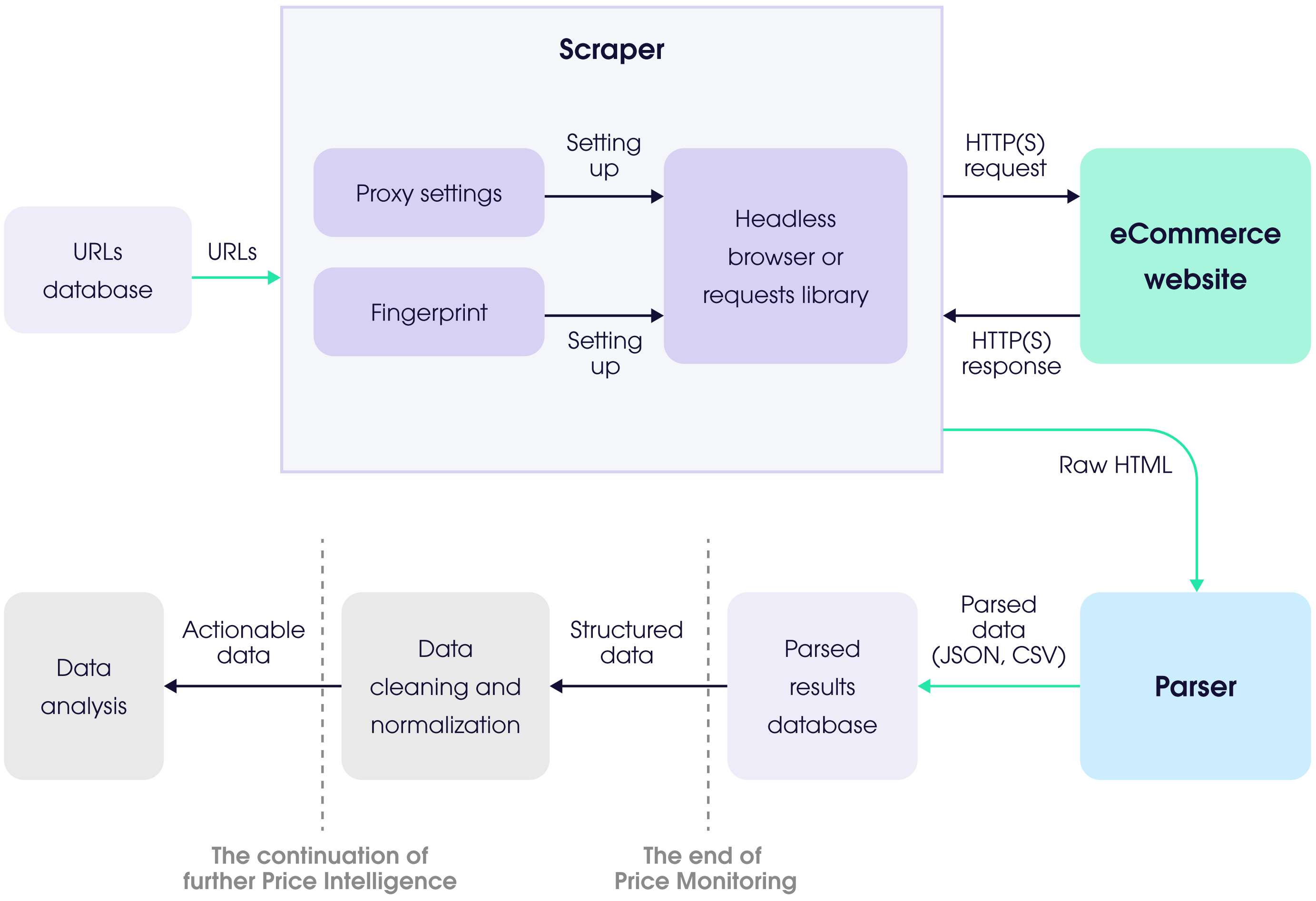
- **Price changes** – triggering alerts.
- **Price comparison** – knowing where a company stands compared to the competitors.
- **Pricing analytics** – analyzing the retrospective of changes.

To take advantage of the processes above and collect localized real-time data from public eCommerce sources, an expansive array of actions and mechanisms needs to be set up and executed.

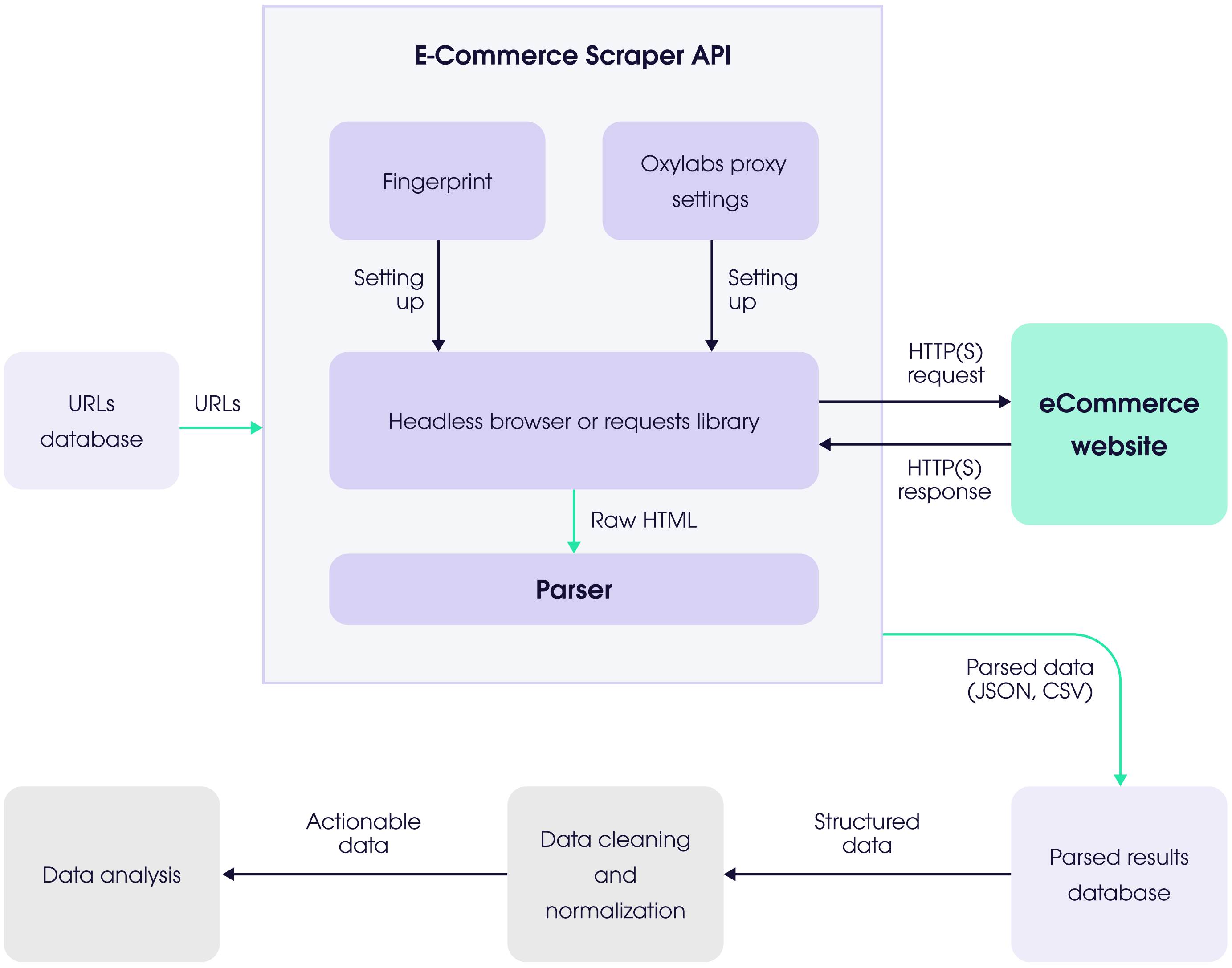
Price monitoring architecture

Price monitoring technical architecture consists of four main actions. Each action has multiple subactions and options to choose from according to a particular task or available tools at hand:

1. Collecting target URLs.
2. Web scraping.
3. Data parsing.
4. Data cleaning and normalization (optional).



The architecture is highly customizable, allowing a purely hands-on approach or completely automated solutions as well as a combination of both.



The price monitoring architecture supported by the Oxylabs' solutions, namely E-Commerce Scraper API and our wide selection of Residential and Datacenter proxies, automates and facilitates a large portion of the underlying processes.

Collecting target URLs

First and foremost, after a thorough analysis of the competitive landscape, target websites and their URLs are to be selected. The next step is to set up a database for URLs through either manual collection or automated solutions such as web crawling – systematic browsing of the internet based on certain keywords to index the most viable candidates for eCommerce monitoring.

After establishing a URL database, the next step is to configure a data extraction tool – a web scraper.

Web scraping

Configuring a web scraper involves three steps:

1. Setting up proxies
2. Creating a fingerprint
3. Sending an HTTP request

Setting up proxies

First, setting up a desirable virtual location requires connection to proxies with a more limited location coverage (Datacenter Proxies) or configuration of proxy settings to establish custom location (Residential Proxies). The largest eCommerce sites usually provide their prices and offerings based on the visitor's location. If it is not an international site version, country and city-level targeting opportunities from a proxy provider will cover almost every possible location worldwide. The location should be determined by a targeted eCommerce website.

Next, the differentiation between the two proxy types, residential and datacenter, adheres to a specific scraping scenario. Residential proxies are less prone to being flagged, blacklisted, or banned because they originate from household devices in use, and their IPs are assigned by internet service providers.

As a result, residential IPs resemble organic traffic, more efficiently concealing web scraping operations. Residential proxies can potentially cover every corner of the world, offering a greater range of locations as data centers are mainly clustered in and around more densely populated areas.

If the priorities revolve around high speed and large data volumes, however, datacenter proxies are more capable of handling heavy loads, granted that a target website is not blocking multiple traffic instances originating from a singular subnet.

Performance and circumvention of anti-scraping measures are the main diverging characteristics between the datacenter and residential proxies. The final choice depends on whether top-notch legitimacy or general performance takes precedence for the task at hand.

Lastly, toggling settings, such as timing sessions, setting up multiple scraping tasks in parallel, choosing protocols (SOCKS5, HTTP), adjusting request limits, and configuring proxy rotation tools are essential considerations that will add up to the chances of success for the whole process.

NOTE: the exact details of how often proxies need to be configured, which type should be used, and similar issues are highly dependent on scraping targets, frequency of data extraction, and other factors. Proxy personalization is paramount.

Creating a fingerprint

Internet browsers provide information to websites upon every visit. This transaction allows a server to understand the nature of content destined for a user. The server determines language settings, layout preferences, and other parameters, as well as collects information about the browser, operating system, and device in use.

Browser fingerprinting is the process of tracking users based on the HTTP headers, GPU and CPU responses, and other data. It works by combining small data points into a larger picture to create a digital "fingerprint". In web scraping, the manual process of constructing a set of organic HTTP headers takes precedence. Upon a visit, a browser sends a set of HTTP headers to the receiving server. Combining User-Agents with associated headers assures that the browser is sending successful requests contributing to avoiding blocks when web scraping.

Sending an HTTP request

To make an HTTP request, a combination of URL, proxy settings, and headers are sent using a request library to an eCommerce site. Alternatively, requests can be sent via a vast number of headless browsers. Luckily, the most popular internet browsers are also available in headless mode.

Various programming languages have sprawling web scraping networks with multiple libraries that support HTTP requests out of the box. Here are the most popular request libraries for some of the most common web scraping languages:

- Python: Requests, aiohttp
- JavaScript: Request, Axios
- R: rvest
- Golang: Colly
- Ruby: HTTParty, Kimurai
- PHP: Goutte, Panther

NOTE: Selenium, an automation tool for browsers, works well with most of these languages.

Alternatively, a more automated approach, an API (automated programming interface), could be used to alleviate some of the more time-consuming steps and processes. The main benefits of dedicated Scraper APIs when compared to, for example, Selenium are:

- Automation of web scraping processes
- Ease of scalability
- Lack of extra coding
- 100% return rates per successful requests
- Built-in tools (proxy rotator, etc.)

In case of failure, response evaluation, error handling (rechecking settings and setups), and retrying take place. After successfully getting the HTML file, it is transferred to a parser.

Parsing

Extracted data in raw HTML is unstructured and difficult to read. A parser pulls the HTML file from the scraper and processes it to a more conveniently readable format, such as JSON and CSV. The structurization process should be tailored to parse the specific data elements by locating their HTML attributes from a particular eCommerce site. A parser determines what information from an HTML string is useful based on predefined rules.

A typical eCommerce retailer displays most of these data elements:

- Title
- Product code – stock keeping unit (SKU)
- Price
- Discounted price
- Shipping cost to a specific location
- Delivery time
- Product specification and description
- Related products
- Bundled products
- Review rating
- Stock availability
- Product URLs
- Images

The main challenge is adapting to frequent layout changes of eCommerce sites. Parsers require constant maintenance to stay on top of the changes to deliver consistent results.

Choosing the data elements above and configuring a parser accordingly will return the desired results in a structured format.

Data cleaning and normalization

For extra clarity, the structured data file can be passed through the final step of quality control to ensure the removal of inaccurate or corrupt records or to simply convert currencies and translate foreign language texts.

Finally, the actionable data can be passed on for further analysis that, in turn, could influence ongoing and upcoming pricing strategies.

Summary

The automation of price monitoring is the only way to stay on top of the price intelligence challenges. Either keeping up with price changes, uncovering competitor strategies, or observing historical pricing data, price monitoring reveals applicable business patterns.

Completing the actions of the price monitoring architecture will grant a price scraping software that efficiently collects structured eCommerce data. A good price monitoring tool is the one that pays for itself in saving time.

We hope the architecture provided in this paper will be a considerable help stepping into the eCommerce data collection. If still unsure of the actions, settings, or tools to use, we recommend referring to an expert who could assist in finding the best solution based on individual activities and budget.

oxylabs®

Want to Know More?

If you would like to know more about any of the topics mentioned in this white paper or **learn about our products**, please get in touch! Our team is ready to answer any of your questions and offer you the best solution for your business needs.

Get in touch with Oxylabs

Our Mission

Our mission is to share all the know-how that we collected over the years in the industry in order to create the future where big data is accessible to all businesses. We seek to create a healthy environment for everyone to grow and thrive in.

Our Values

As a leading company in the proxy and web scraping industry, we ensure that the highest standards of business ethics lead all our operations. Our core values guide us toward achieving our mission. **Learn more**