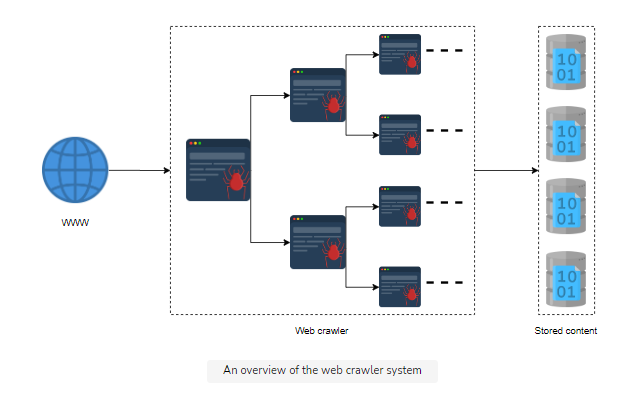
**Web Crawler**

* A web crawler is an Internet bot that systematically scours the world wide web (WWW) for content, starting its operation from a pool of seed URLs. This process of acquiring content from the WWW is called crawling.
* It further saves the crawled content in the data stores. The process of efficiently saving data for subsequent use is called storing.
* This is the first step that’s performed by search engines; the stored data is used for indexing and ranking purposes.
* The output of the crawling process is the data that’s the input for the subsequent processing phases - data cleaning, indexing, page relevance using algorithms like page ranks, and analytics.
* Additional Utilities of Web crawler includes Web pages testing, Web pages monitoring, Copyright infringement check, Site mirroring.



* Functional Requirements:

1. Crawling: The system should scour the WWW, spanning from a queue of seed URLs provided initially by the system administrator.
   1. From where do we get seed URLs from? Manually create them, Scan IP addresses for the presence of web servers.
   2. How do we select seed URLs for crawling? Location-based, Category-based, Popularity-based.
2. Storing: The system should be able to extract and store the content of a URL in a blob store. This makes that URL and its content processable by the search engines for indexing and ranking purpose
3. Scheduling: Since crawling is a process that’s repeated, the system should have regular scheduling to update its blob stores’ records.

* Non-Functional Requirements:
  1. Scalability
  2. Extensibility
  3. Consistency
  4. Performance: The system should be smart enough to limit its crawling to a domain, either by time spent or by the count of the visited URLs of that domain. This process is called **self-throttling**. The URLs crawled per second and the throughput of the content crawled should be optimal.
  5. Improved user interface-customized scheduling: Besides the default recrawling, which is a functional requirement, the system should also support the functionality to perform non-routine customized crawling on the system administrator’s demands.
* Resource Estimations
  1. Storage Requirements
  2. Traversal Time
  3. Number of Servers
  4. Bandwidth Estimations