In the context of an API gateway in a microservices architecture, rate limiting and throttling are mechanisms used to control the usage of APIs and protect the services from being overwhelmed. Although they are often used interchangeably, they have distinct differences.

**Rate Limiting**

**Definition:** Rate limiting is the process of restricting the number of API requests a client can make within a specified time period.

**Purpose:**

* Prevents misuse and abuse of APIs.
* Ensures fair usage among multiple clients.
* Protects backend services from being overloaded by too many requests in a short span of time.

**Example:**

* A client can make up to 100 requests per minute. Any requests beyond this limit will be denied until the time window resets.

**Implementation:**

* Rate limits are typically defined in terms of a maximum number of requests per unit of time (e.g., 1000 requests per hour).
* Responses to requests that exceed the limit usually return an HTTP status code like 429 (Too Many Requests).

**Throttling**

* **Definition:** Throttling is the process of controlling the rate of API requests a client can make over time by slowing down the rate at which requests are processed or by delaying the requests.

**Purpose:**

* Ensures the system remains responsive under high load by managing the flow of incoming requests.
* Prevents a service from being overwhelmed by sudden spikes in traffic.

**Example:**

* A client is allowed to make 10 requests per second. If they exceed this rate, subsequent requests may be delayed or queued to be processed at a slower rate.

**Implementation:**

* Throttling controls the flow of requests by introducing delays or pauses.
* It can be more flexible than rate limiting, allowing a certain burst of traffic but then slowing down subsequent requests.