Use UUID 4:

- Focus is on uniqueness

- No need for chronological order or device identification

Use MongoDB ObjectId:

- If the data has time-series requirements

- Want the ID to be more compact and contain temporal information

Avoid UUID 1:

- Avoid using uuid1 if concerned about exposing device information or security (e.g., MAC address disclosure).

Use SnowFlake:

- Ideal for distributed systems, especially for tasks that require temporal ordering and ultra-high generation speed

- Distributed Uniqueness: Snowflake IDs are composed of timestamps, data center IDs, machine IDs, and serial numbers to ensure uniqueness in a distributed environment.

- Time-ordered: Generated based on timestamps, IDs are generated in increasing order for easy sorting and querying.

- Efficient Generation: Generation speed is very fast, suitable for high concurrency scenarios.

- Compactness: Snowflake IDs are 64-bit integers (19-bit numbers), which are shorter than UUIDs, saving storage space.

- No Dependency: No need to rely on external databases or services, it is completely done by the generation algorithm and local machine.h as database master keys or large-scale data processing

- Easy to parse (metadata): Timestamp, data center, and machine information can be extracted from the ID for debugging or tracing.