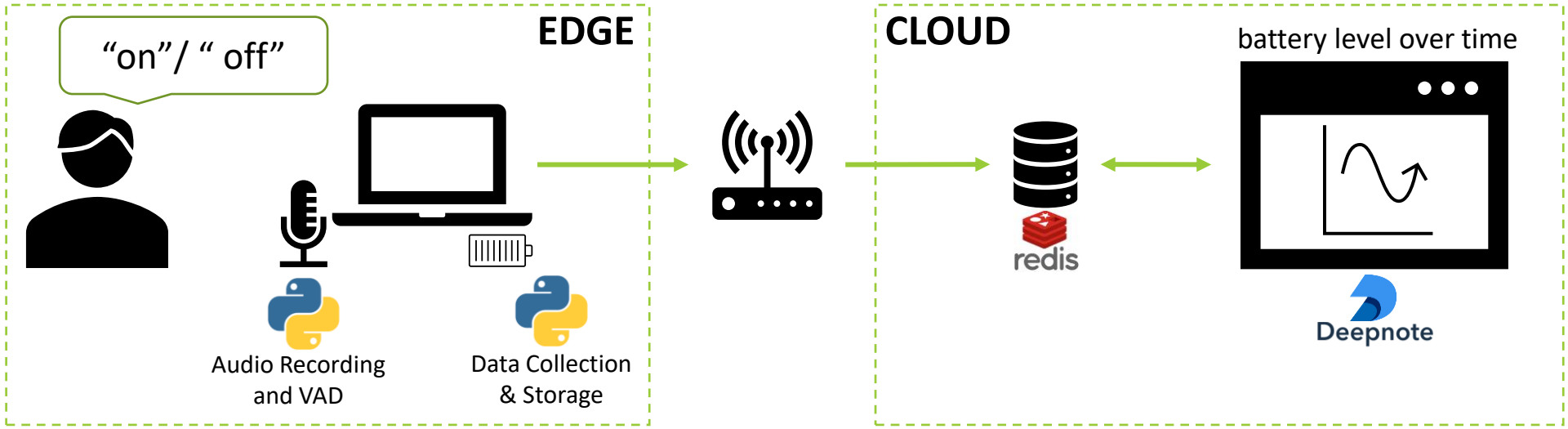
Machine Learning for IoT

LAB2: Pre-processing

LAB1-2: Smart Battery Monitoring (Simplified)



LAB2 Content

- Timeseries Processing:
  - Compression
  - Retention
  - Aggregation
- Audio Processing:
  - Resampling
  - Discrete Fourier Transform
  - Short-Time Fourier Transform
  - Mel Spectrogram
  - Mel-Frequency Cepstral Coefficients

LAB2 Content

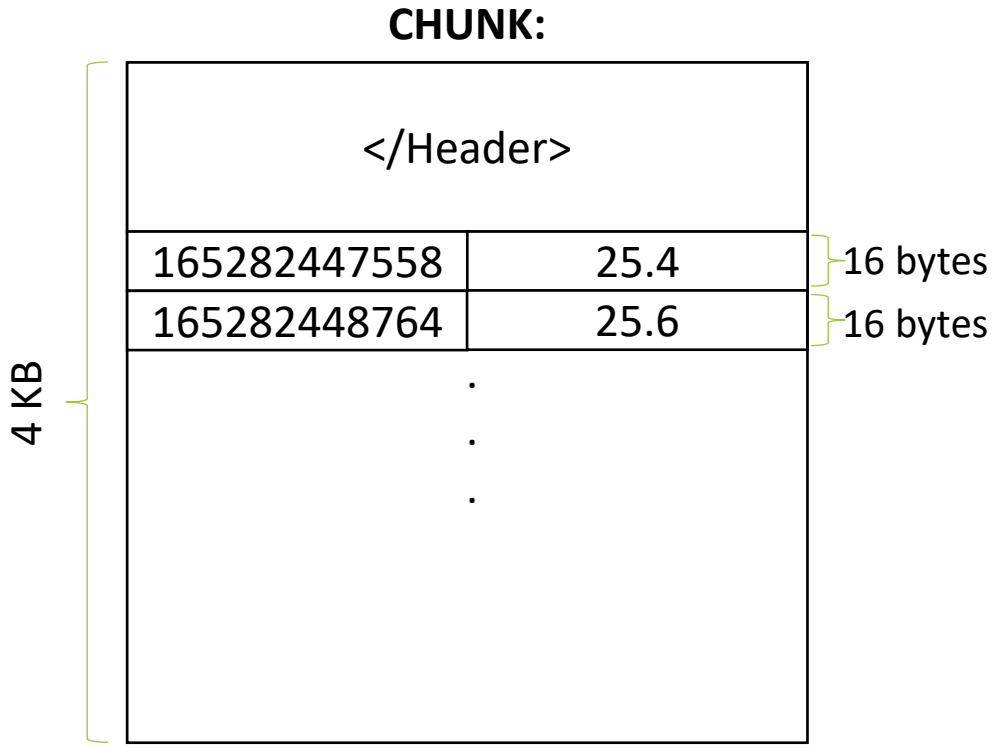
- **Timeseries Processing:**
  - **Compression**
  - **Retention**
  - **Aggregation**
- Audio Processing & Feature Extraction:
  - Resampling
  - Padding
  - Normalization
  - Fourier Transform
  - Short-Time Fourier Transform
  - Log-Mel Spectrogram
  - Mel-Frequency Cepstral Coefficients

Covered in lectures

Timeseries Processing

Redis TimeSeries Memory Model

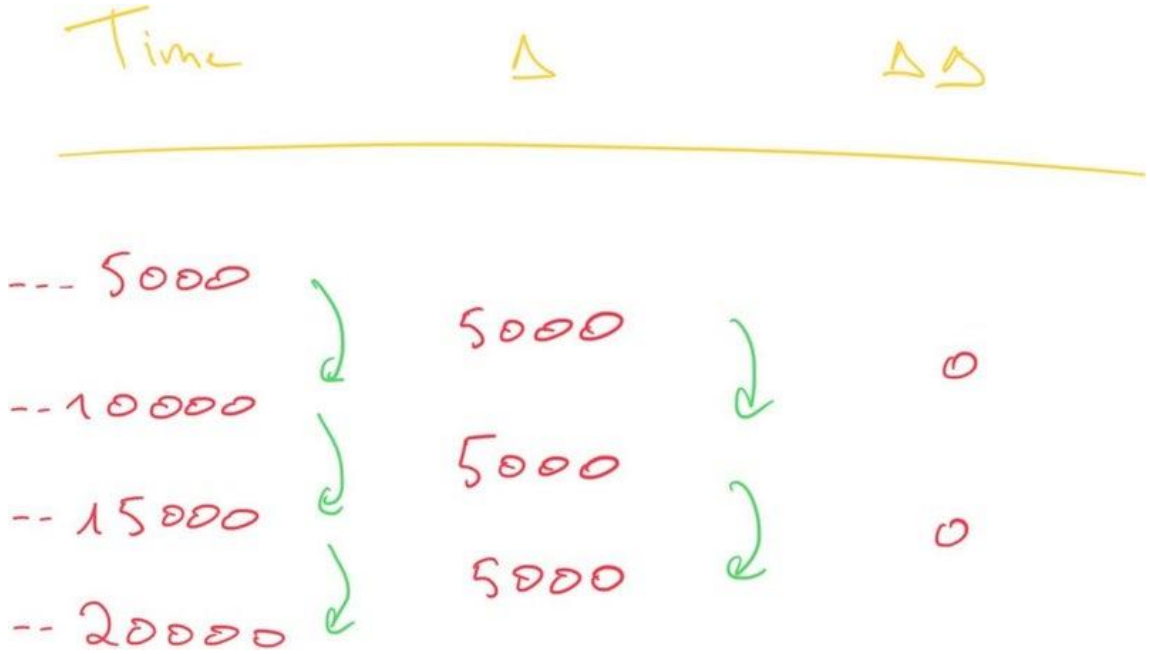
- A Redis TimeSeries consists of a list of linked chunks
- Each chunk contains
  - Header
    - Information needed by Redis to manage the data
  - A set of Records
    - Each record consists of:
      - Timestamp: 64-bit (8 bytes)
      - Value: 64-bit (8 bytes)
- Chunk size is set when creating the TimeSeries
  - Default: 4 KB
  - Smaller → Less Memory, Slower Read/Write
  - Larger → More Memory, Faster Read/Write



TimeSeries Compression

- Lossless compression
  - Gorilla algorithm

Timestamp Compression:



Value Compression:

TimeSeries Compression

- Lossless compression
  - Gorilla algorithm
- Memory savings:
  - Depends on data
    - Best-case: 98.4%
    - Worst-case: 113.3%
      - Memory increases! But rare.
    - Average-case: 90.0%
- Compression improves performance due to a lower number of memory accesses

- **Note:** Compression is active by default

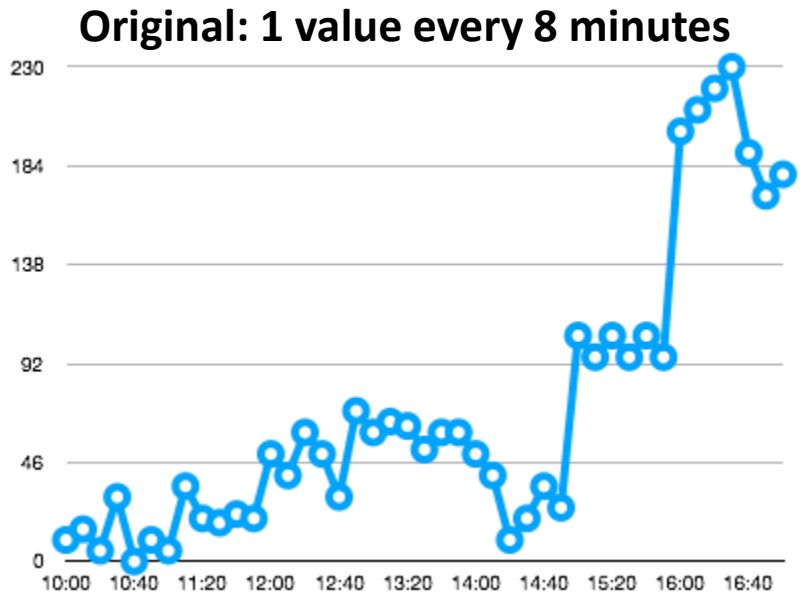
Example

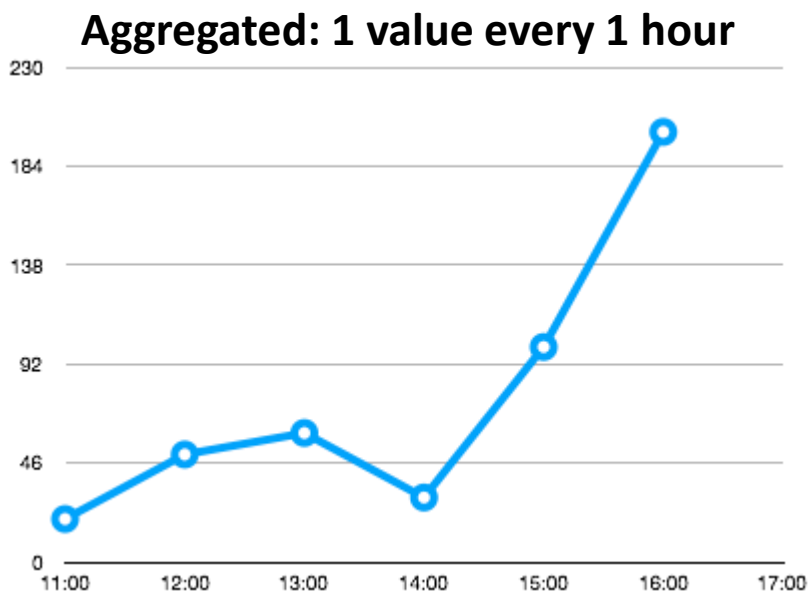
- Which is the memory usage to store temperature every 5 seconds after 1 month?
  - 1 month = 30 days * 24 hours * 60 minutes * 60 seconds = 2592000 seconds
  - # of records = 2592000 / 5 = 518400
  - Uncompressed Memory ≈ 518400 * 16 bytes = 8294400 bytes = 7.910 MB
  - Compressed Memory ≈ 7.910 MB - 90% = 0.791 MB

  - Approximations:
    - We neglected the header size
    - We neglected that the memory usage is always a multiple of the chunk size
    - We considered the average compression ratio

TimeSeries Aggregation

- Lossy Compression
- Aggregation Parameters:
  - Bucket Duration
  - Aggregation type: avg, sum, min, max, range, count, first, last.
- **Note:** Aggregation never changes the original timeseries but creates a new one





TimeSeries Retention

- You can prevent your timeseries growing indefinitely by setting a maximum age for samples compared to the last event time (in milliseconds).
- By default, retention is 0
  - i.e., the timeseries will be never trimmed