# Enabling Micro-Synchrophasor Data Analytics

**Omid Ardakanian**
University of California, Berkeley

NASPI March 2016

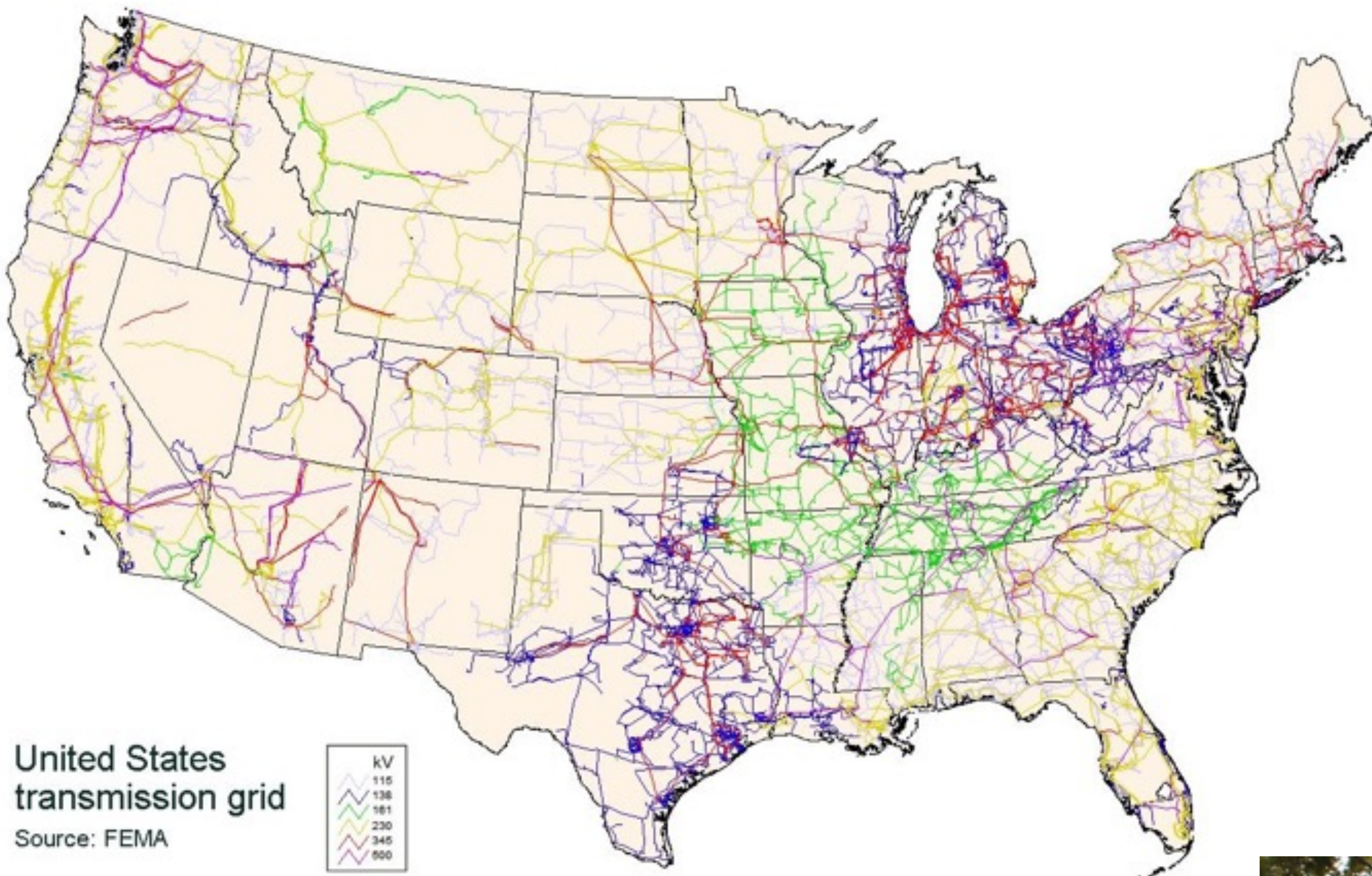
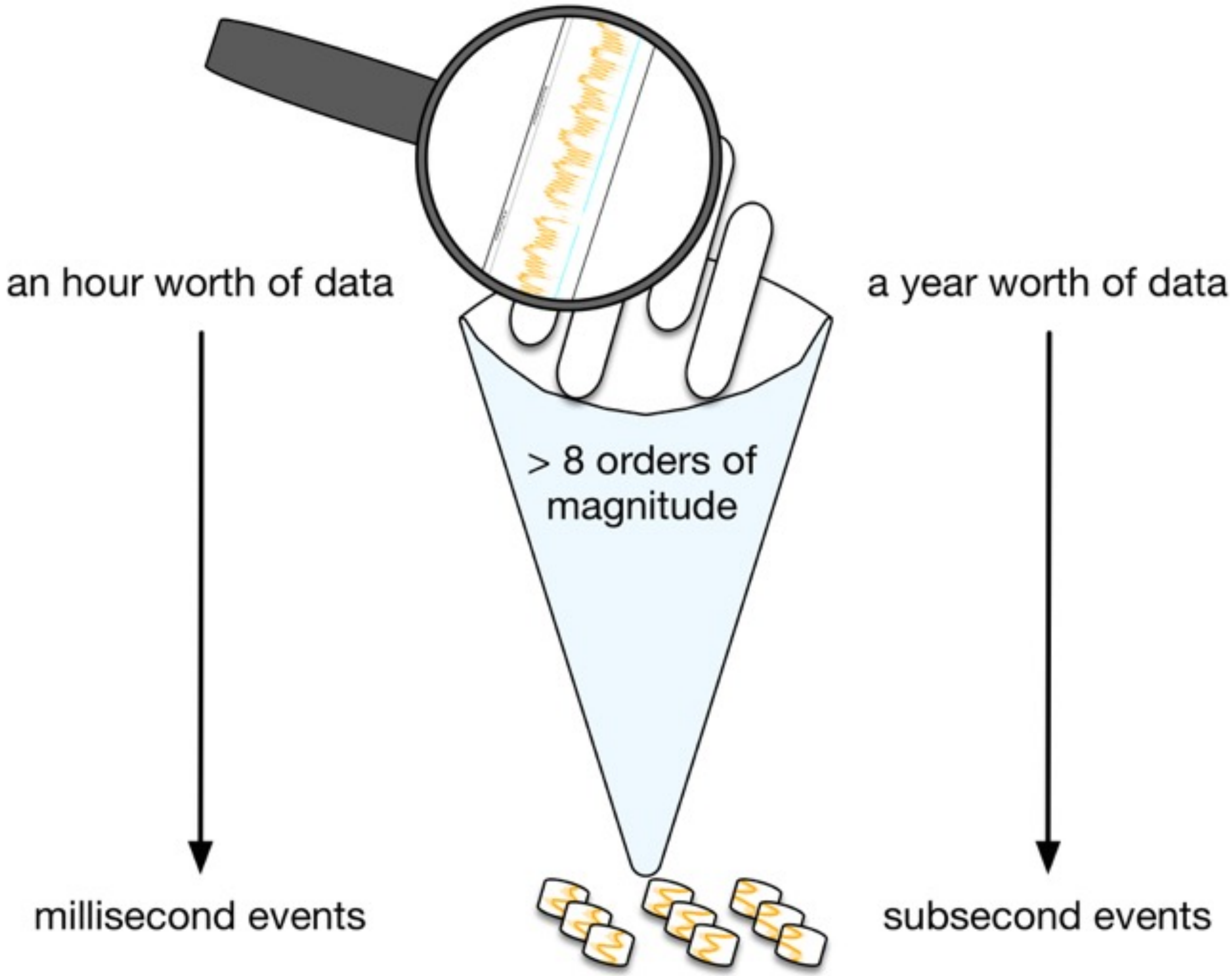



image source: power standards lab

# Big Data Analytics




120 samples per second -> **3.8 billion samples** per stream per year -> **30 billion bytes** per stream per year

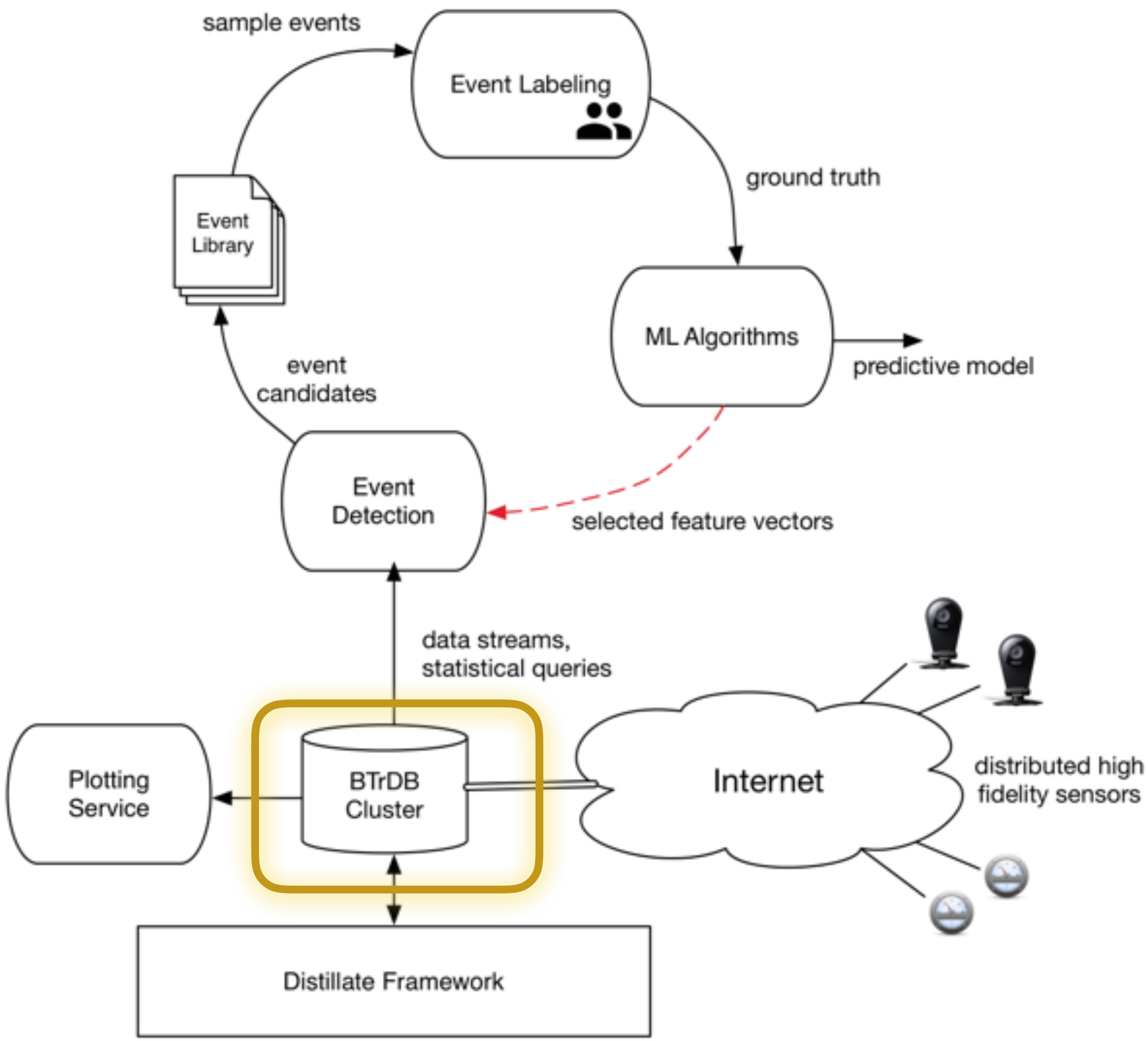
# Introduction

- High-precision high-sample-rate data from distributed high fidelity sensors
  - ✳ many sensors, a wide range of temporal scales, *rare* events
- Finding anomalies in these systems is the holy grail
  - ✳ failing to identify and react to critical events in a timely manner may cost millions of dollars
- Energy data analytics (both real-time and historical) is critical yet computationally expensive
  - ✳ the ability to detect, analyze, and control with a limited time budget

# Goals

- Detect: identify rare events
  - using an efficient search algorithm that is logarithmic in the size of the data set and linear in the number of events that are found
- Analyze: run compute intensive tasks on smaller chunks of data
- Control: take corrective/preventive actions (in real-time applications)

# System Architecture



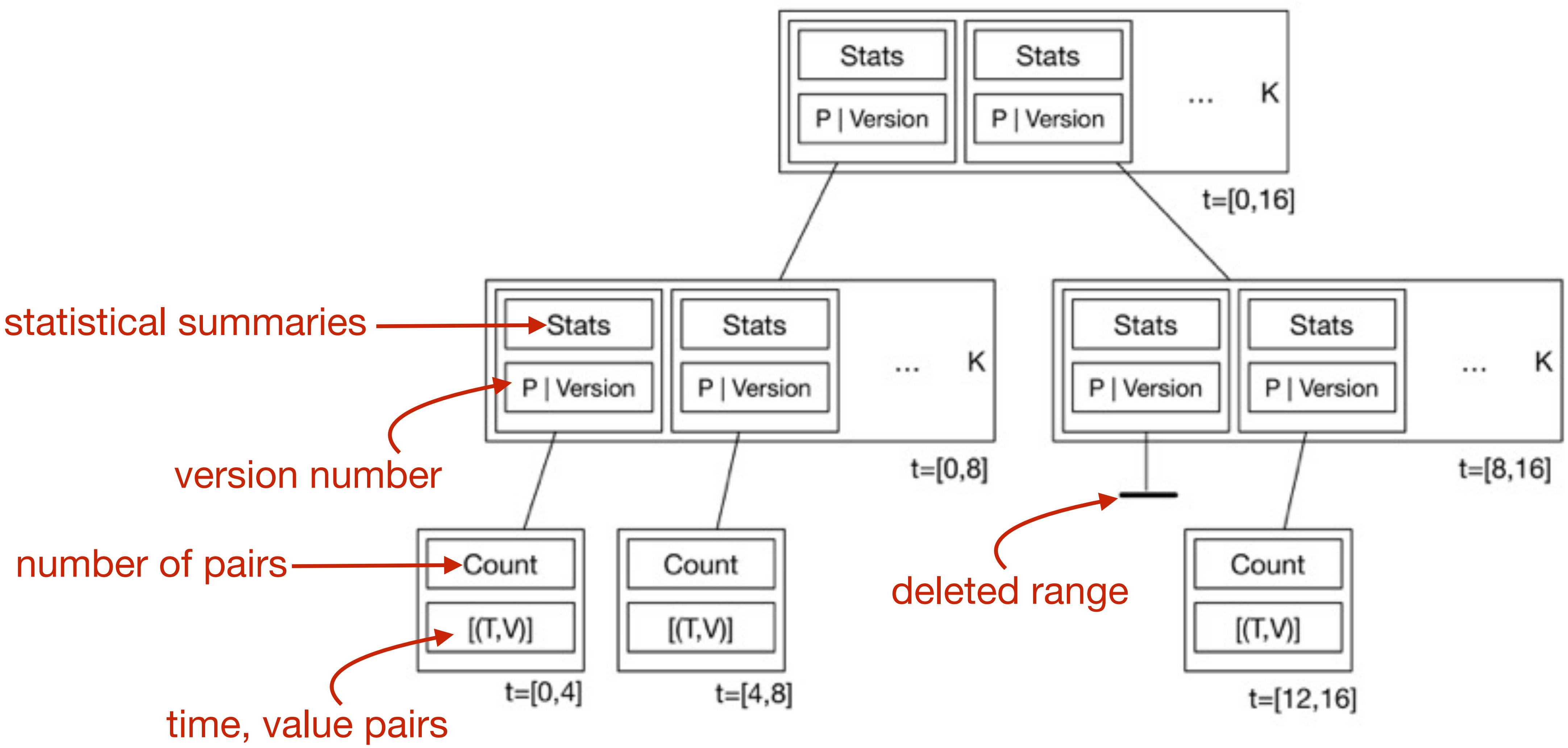
# BTrDB Timeseries Database

- High throughput, fixed response-time timeseries store running on a four-node cluster
  - 53 million inserted values per second
  - 119 million queried values per second
- Provides nanosecond timestamp precision
- Supports out-of-order arrivals

References:
[1] Michael Andersen, Sam Kumar, Connor Brooks, Alexandra von Meier, David Culler, “DISTIL: Design and Implementation of a Scalable Synchrophasor Data Processing System”, IEEE SmartGridComm, 2015.
[2] Michael P Andersen and David E. Culler, “BTrDB: Optimizing Storage System Design for Timeseries Processing”, USENIX Conference on File and Storage Technologies, 2016.

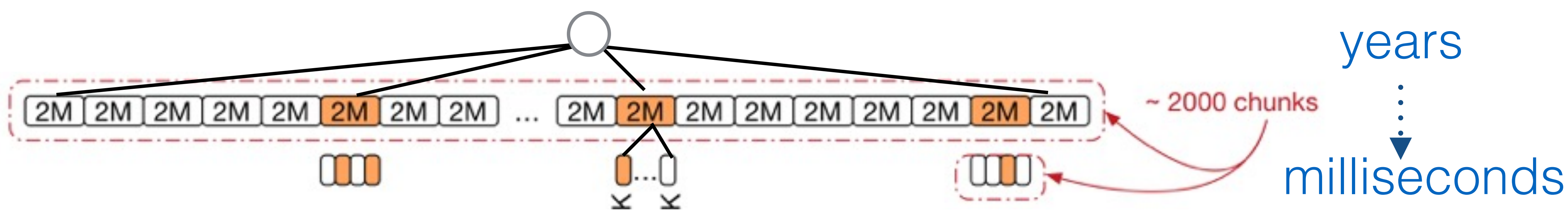
# Abstraction for Timeseries Data

- Time-partitioning version-annotated copy-on-write K-ary tree

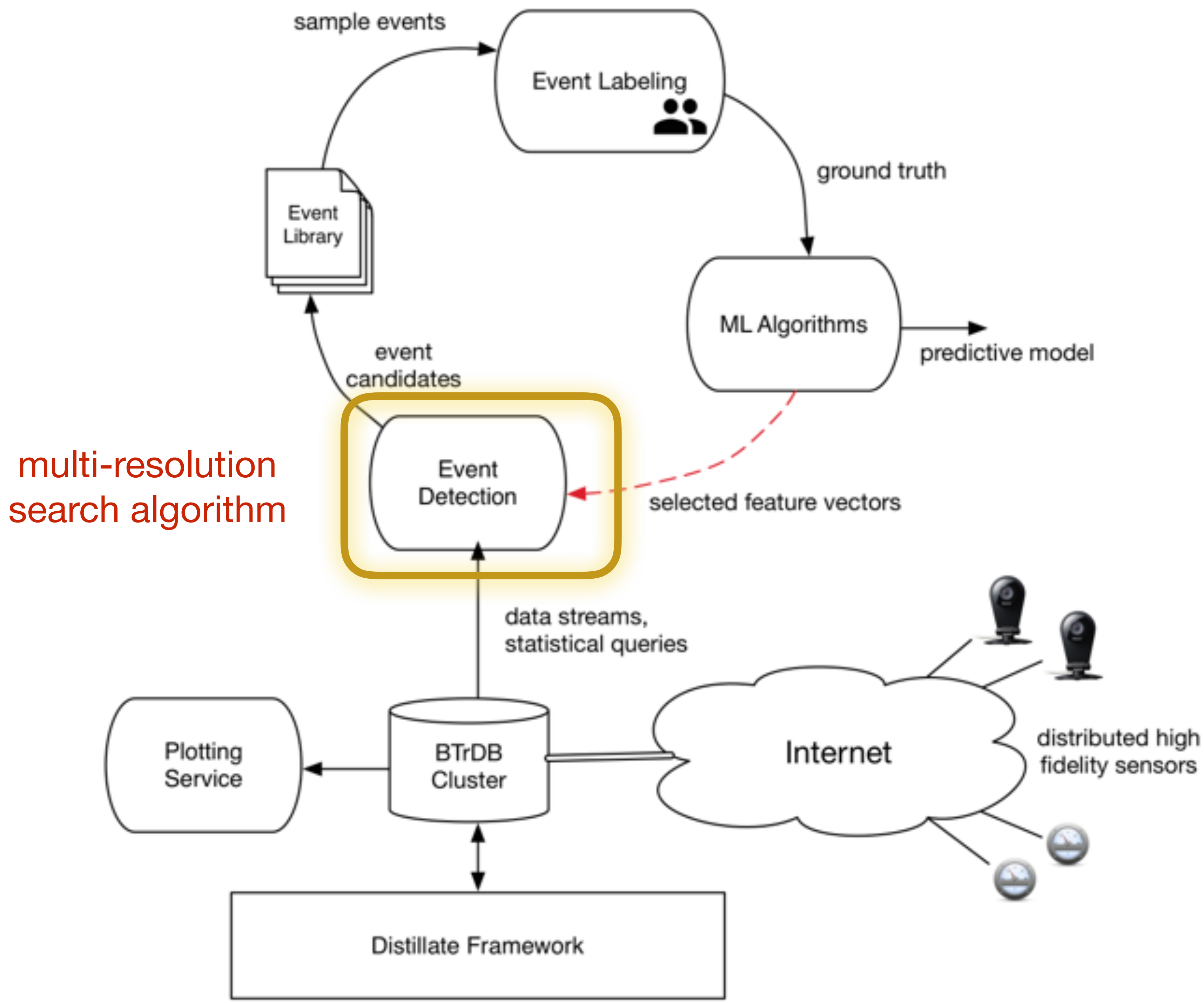


# Statistical Summaries

- statistical summaries (max, min, average, and count) are stored at different temporal resolutions



# System Architecture



# Example Query

Find 5-second intervals that contain at least one value greater than a threshold

# Example Query

Find 5-second intervals that contain a value greater than a threshold

- Query **max** at the given temporal resolution
- Dive down if $\max_{resolution}$ > threshold
- Repeat for the next temporal resolution until the desired resolution is reached

# Multi-Resolution Search

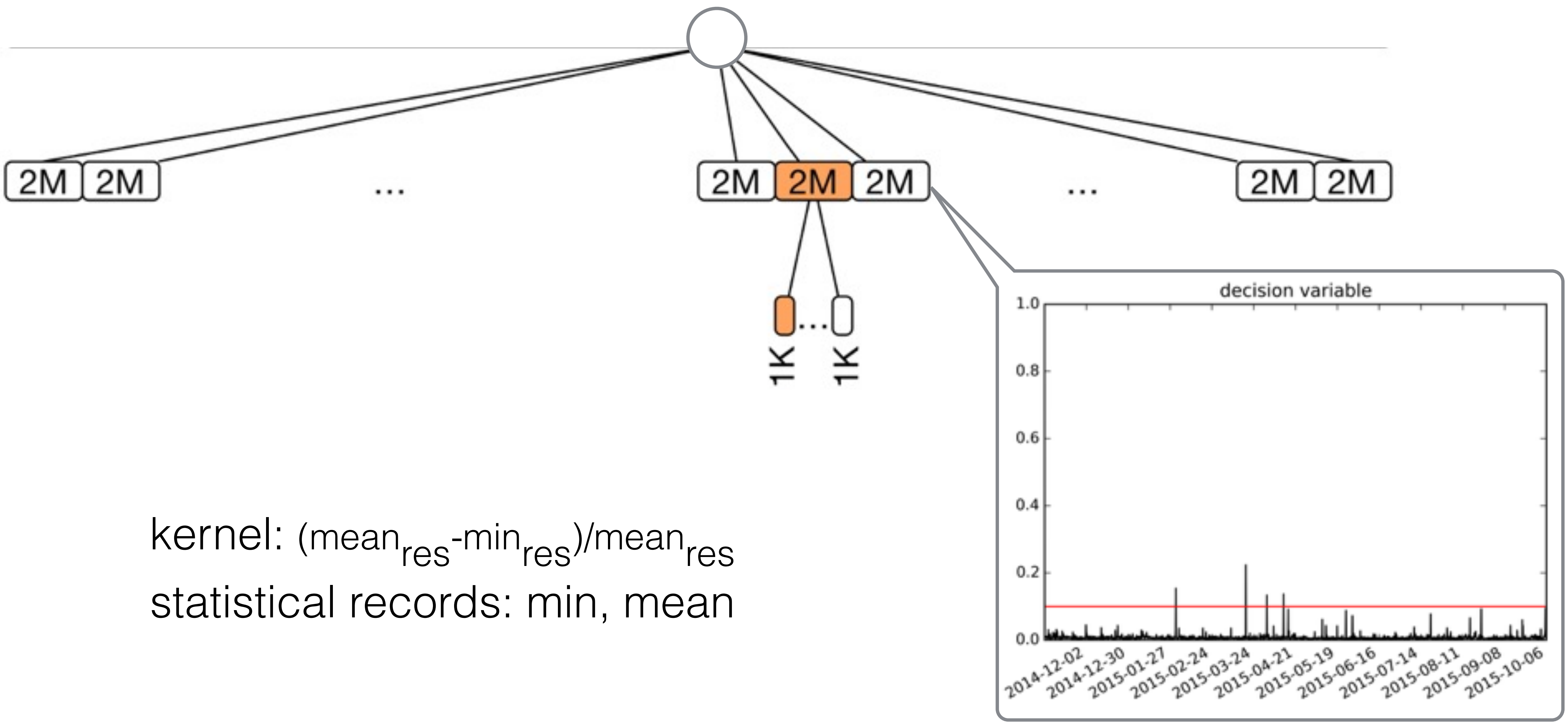
- Start with a definition of the event (search criteria)
- Query statistical summaries of data at a given temporal resolution
- Compare a function of these statistical summaries against a threshold
- Dive down if the condition is satisfied
- Query raw data when the desired resolution is reached and run your algorithm on a small chunk of data

# Interesting Events

- voltage sags
    - voltage magnitude stream
- tap changing events
    - angle difference stream
- reverse flows
    - real power or power factor stream
- switching events
- …

# Case Study: Voltage Sag Detection

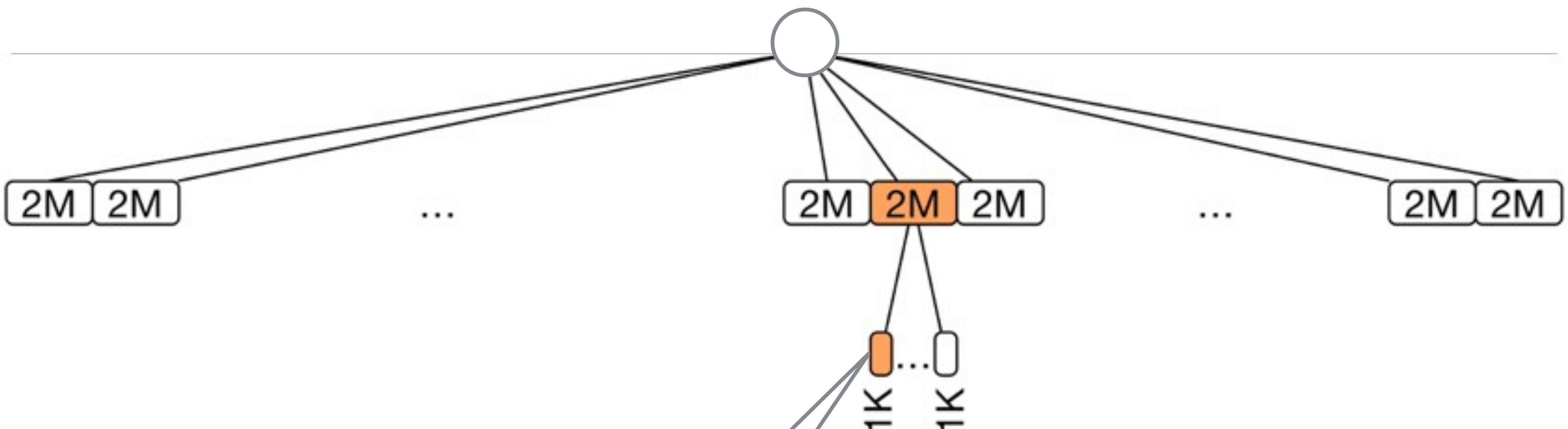
# Step 1: Querying Statistical Summaries at a Given Temporal Resolution

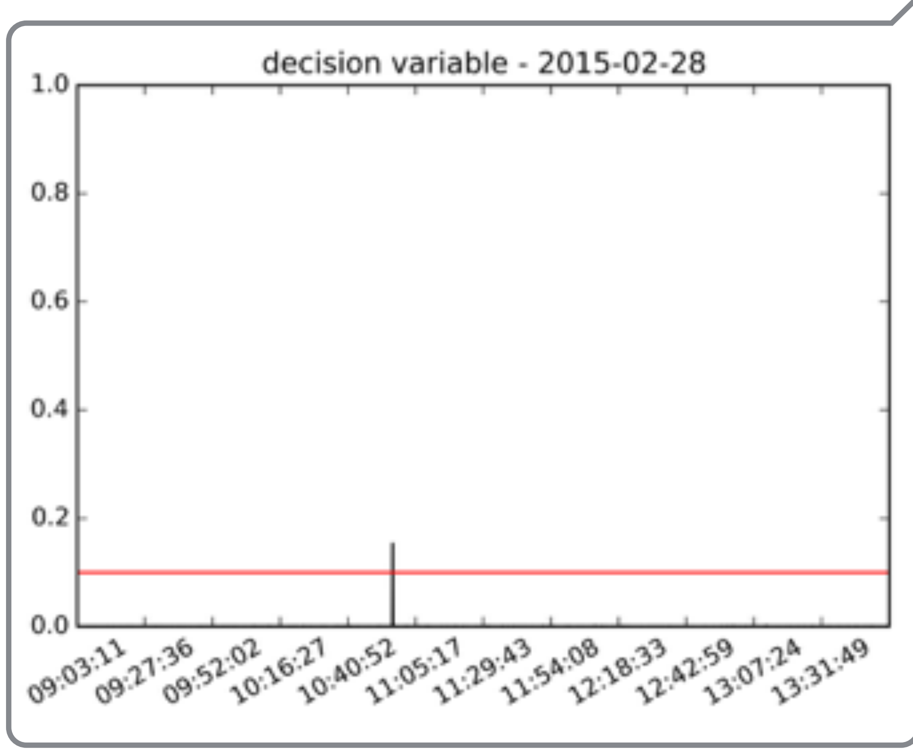



kernel: $(\text{mean}_{res}-\text{min}_{res})/\text{mean}_{res}$

statistical records: min, mean

# Step 2: Diving Down

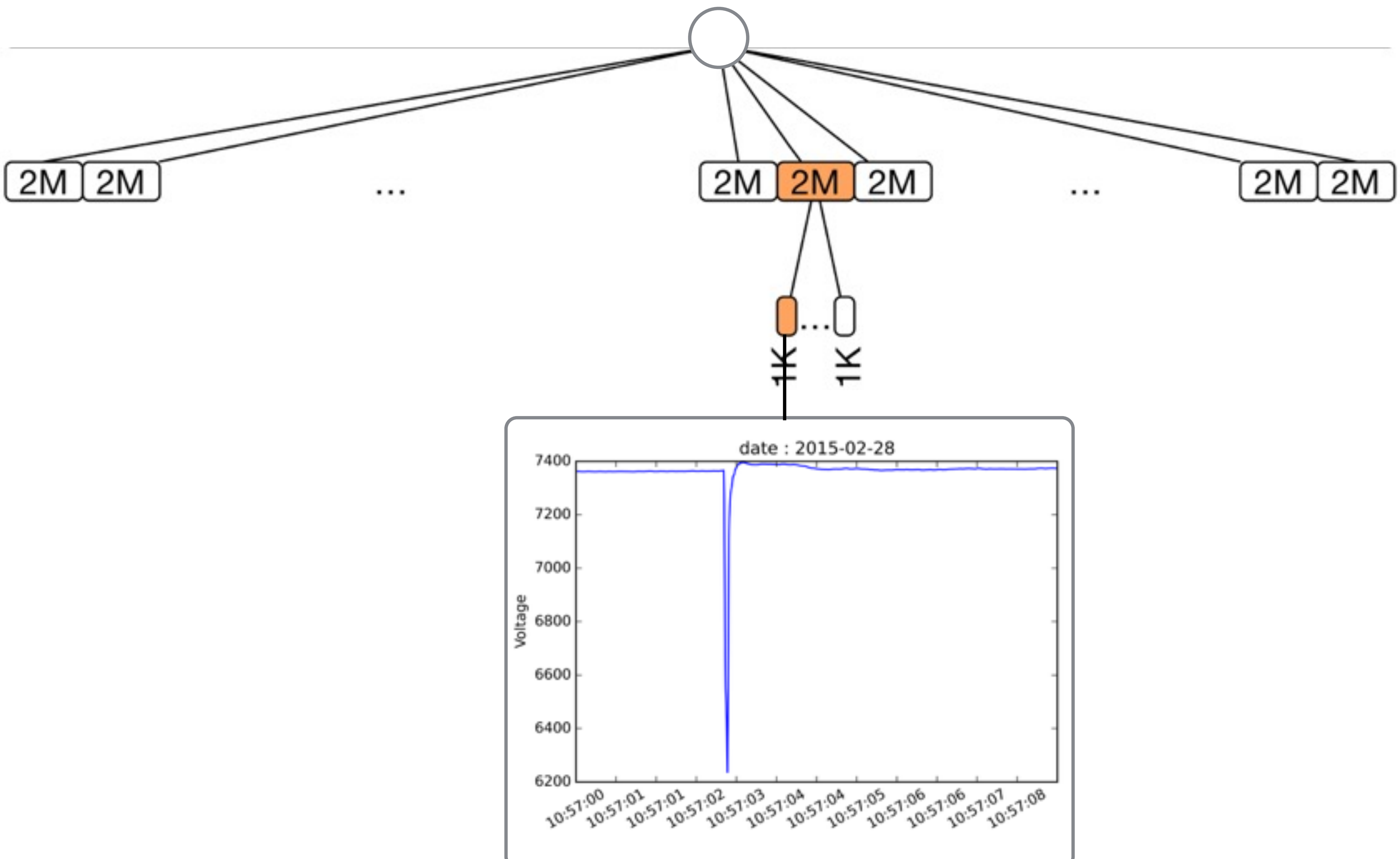


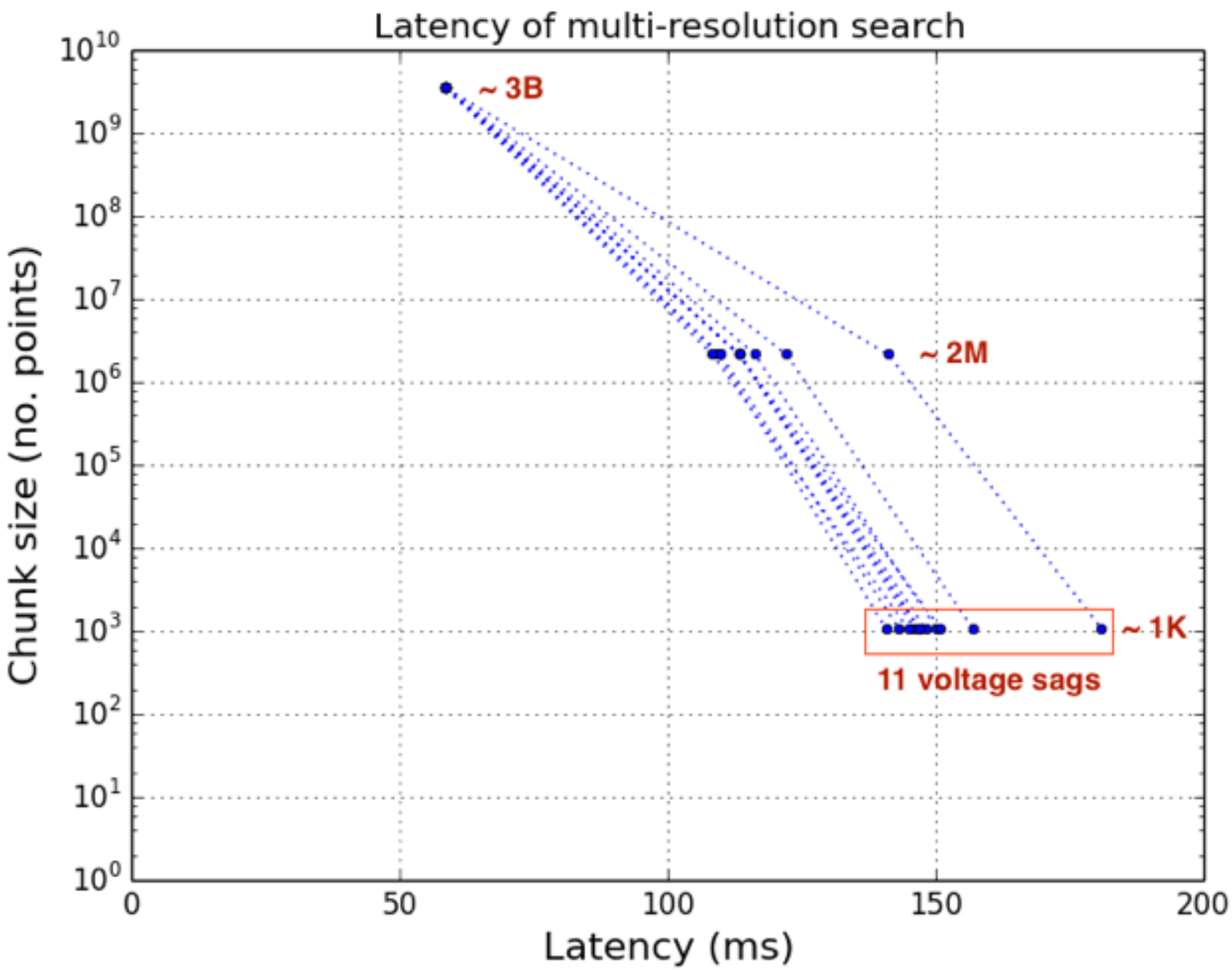

kernel: $(mean_{res}-min_{res})/mean_{res}$

statistical records: min, mean

# Step 3: Querying Raw Data



# Evaluation



Latency of multi-resolution search
~ 3B
~ 2M
~ 1K
11 voltage sags
Chunk size (no. points)
Latency (ms)
$10^{10}$
$10^{9}$
$10^{8}$
$10^{7}$
$10^{6}$
$10^{5}$
$10^{4}$
$10^{3}$
$10^{2}$
$10^{1}$
$10^{0}$
0
50
100
150
200

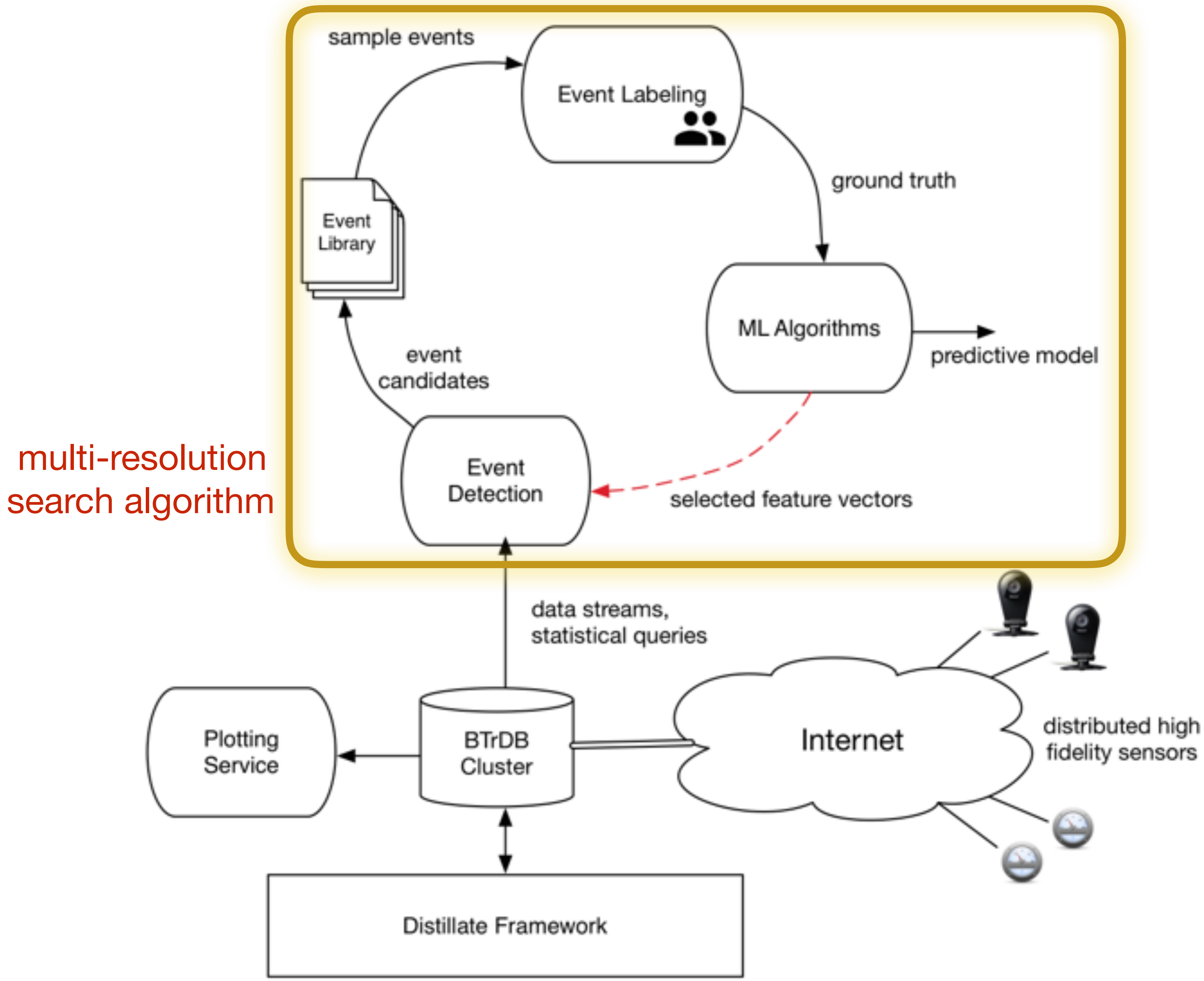
# Example Result

| | no. events (0.05) | runtime (ms) | no. events (0.1) | runtime (ms) | no. events (0.15) | runtime (ms) | no. events (0.2) | runtime (ms) | days |
|---|---|---|---|---|---|---|---|---|---|
| /clean/GP_BUS1/L1MAG | 9 | 431.77 | 4 | 237.13 | 0 | 76.78 | 0 | 88.41 | 135 |
| /clean/GP_BUS1/L2MAG | 10 | 394.39 | 4 | 226.85 | 1 | 115.30 | 0 | 70.55 | 135 |
| /clean/GP_BUS1/L3MAG | 5 | 309.07 | 2 | 163.25 | 1 | 118.95 | 0 | 77.08 | 135 |
| /clean/switch_a6/L1MAG | 14 | 666.59 | 6 | 273.01 | 3 | 194.95 | 1 | 132.75 | 330 |
| /clean/switch_a6/L2MAG | 21 | 947.24 | 11 | 523.78 | 4 | 235.44 | 3 | 190.83 | 330 |
| /clean/switch_a6/L3MAG | 11 | 608.94 | 4 | 318.44 | 2 | 213.57 | 0 | 90.06 | 330 |
| /clean/RPU/CE_CERT_Bld_1200/L1MAG | 8 | 312.53 | 2 | 68.41 | 1 | 64.93 | 1 | 66.55 | 86 |
| /clean/RPU/CE_CERT_Bld_1200/L2MAG | 12 | 379.19 | 4 | 163.71 | 3 | 119.51 | 2 | 112.95 | 86 |
| /clean/RPU/CE_CERT_Bld_1200/L3MAG | 12 | 627.72 | 4 | 228.18 | 2 | 111.41 | 2 | 133.00 | 86 |

10% drop

**logarithmic in the size of the data set and linear in the number of events that are found**

# Event Detection: A Data Driven Approach



# Takeaways

- Complexity of the search algorithm is O(nLog(L))
    - Locating and analyzing rare events among billions of time-value pairs is possible in a fraction of a second
- Defining a kernel function can be quite challenging for some detectors
- Machine learning techniques can be used to develop sophisticated detectors