

CONTEXT

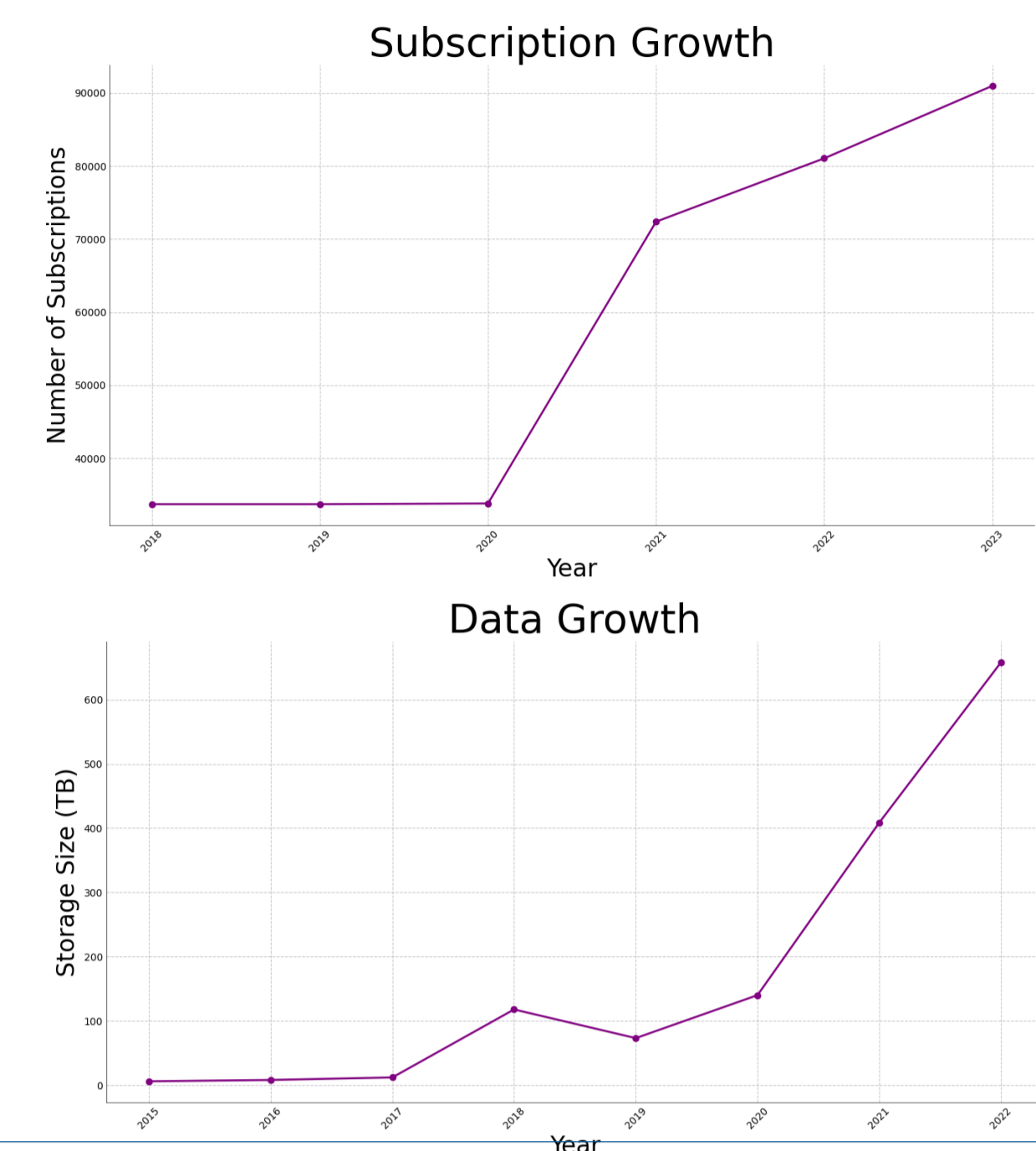
The CERN Accelerator Data Logging Service stores data generated by the accelerator infrastructure and beam related devices. This amounts to **3.5TB of data per day**, coming from more than 2.5 million signals from heterogeneous systems at various frequencies. Around 85% of this data (**0.75PB per year compressed**) is transmitted through the Controls Middleware Infrastructure. To reliably gather such volumes of data, the acquisition system must be highly available, resilient and robust. It also has to be highly efficient and easily scalable, as data rates and volumes increase, notably in view of the High Luminosity LHC.

FAULT TOLERANCE

RESILIENCE

LOAD BALANCING

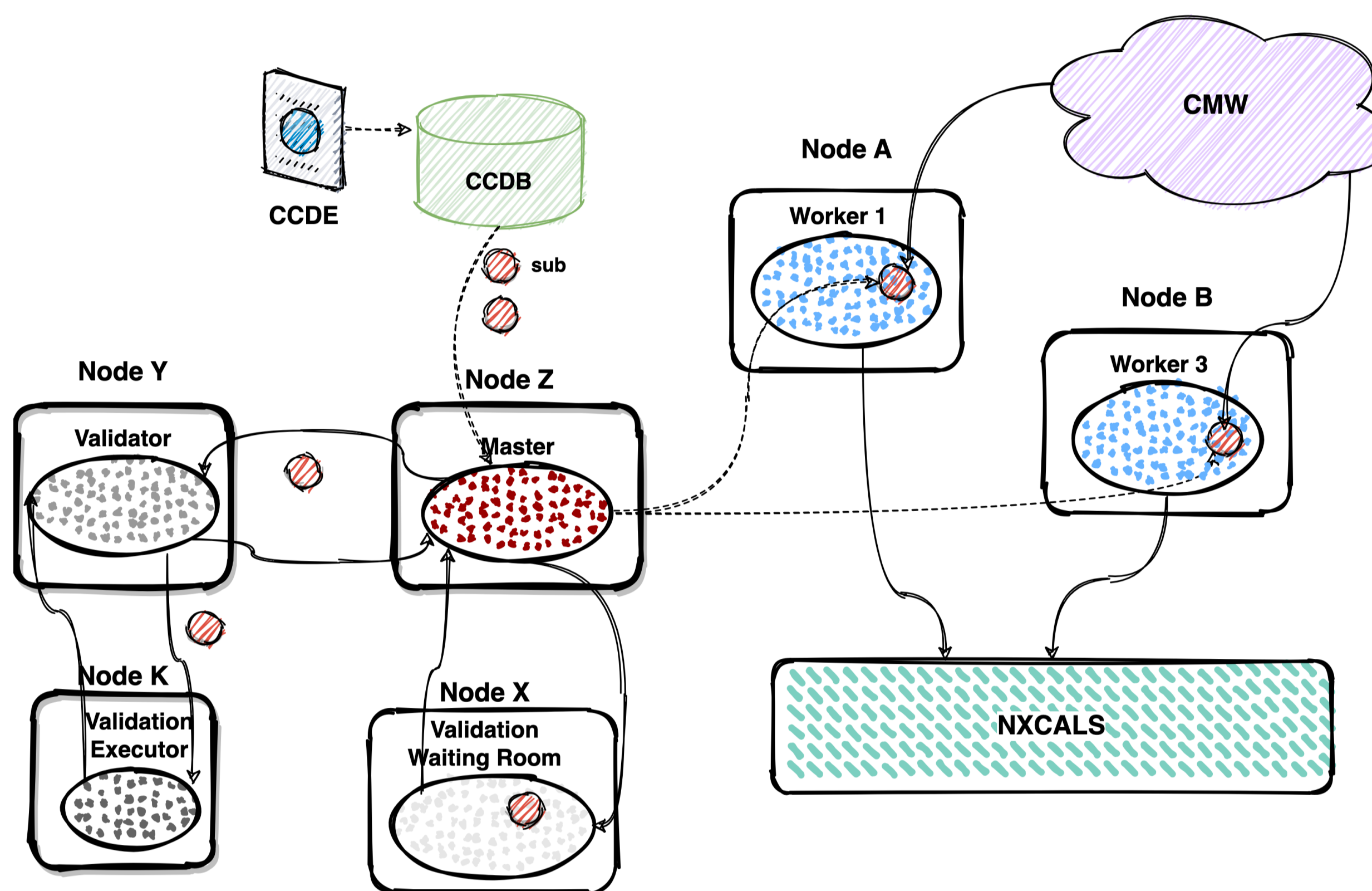
SCALABILITY



ARCHITECTURE

APPROACH

- Master-Worker **clustered architecture**.
- Master coordinates the Workers and distributes subscriptions among them.
- Workers handle subscriptions.
- Subscriptions are validated by the Validator and its executors.
- Based on Akka Framework and leveraging its Actor Model and Cluster features.

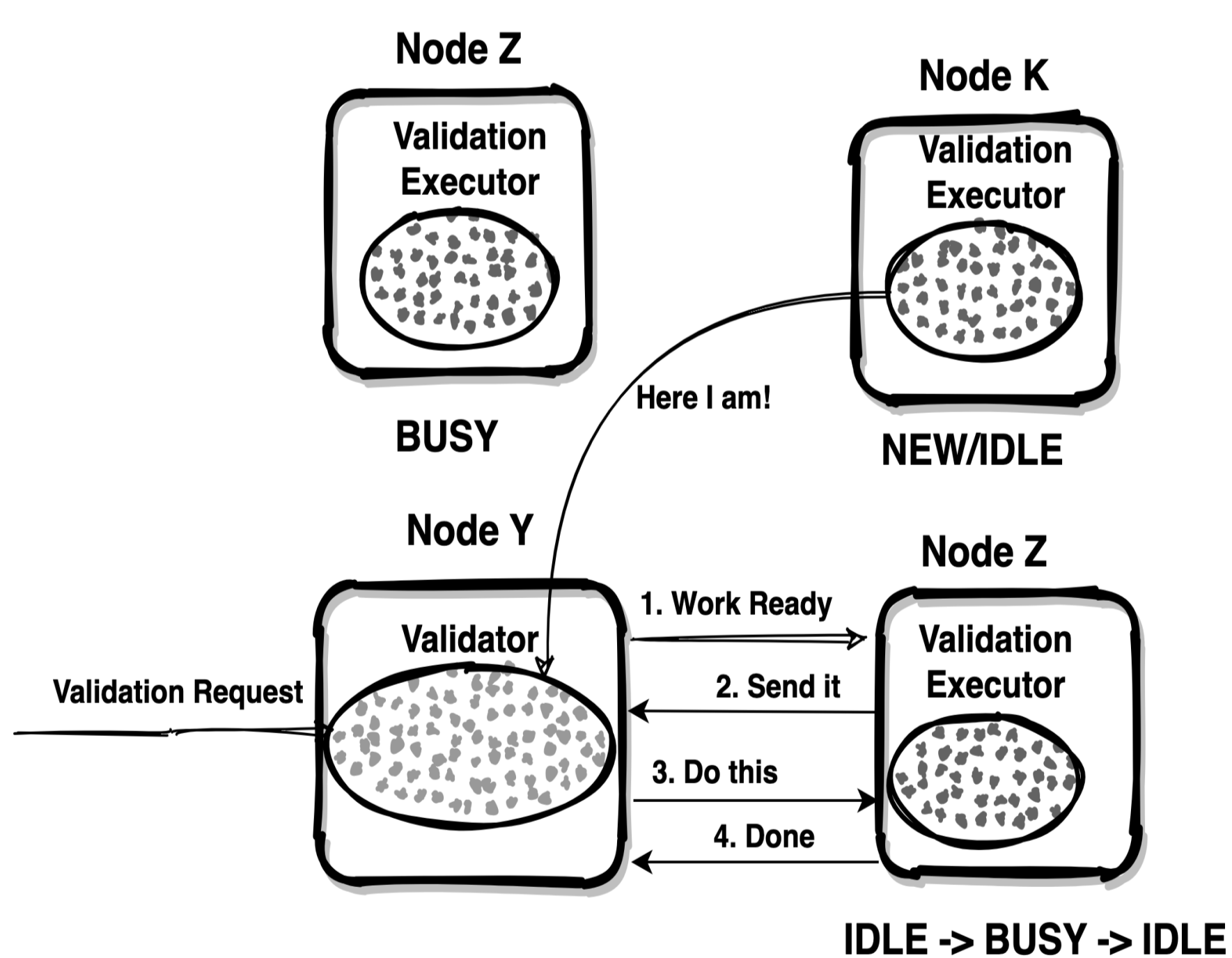


INSTANTIATION

- Handling 90k+ subscriptions.
- Processing 100k+ records per second.
- Cluster → 3 Masters, 36 Workers, 20 Validation Executors.
- Running on 8 bare-metal machines (3TB RAM + 448 virtual CPU).

SELECTED WORKFLOWS

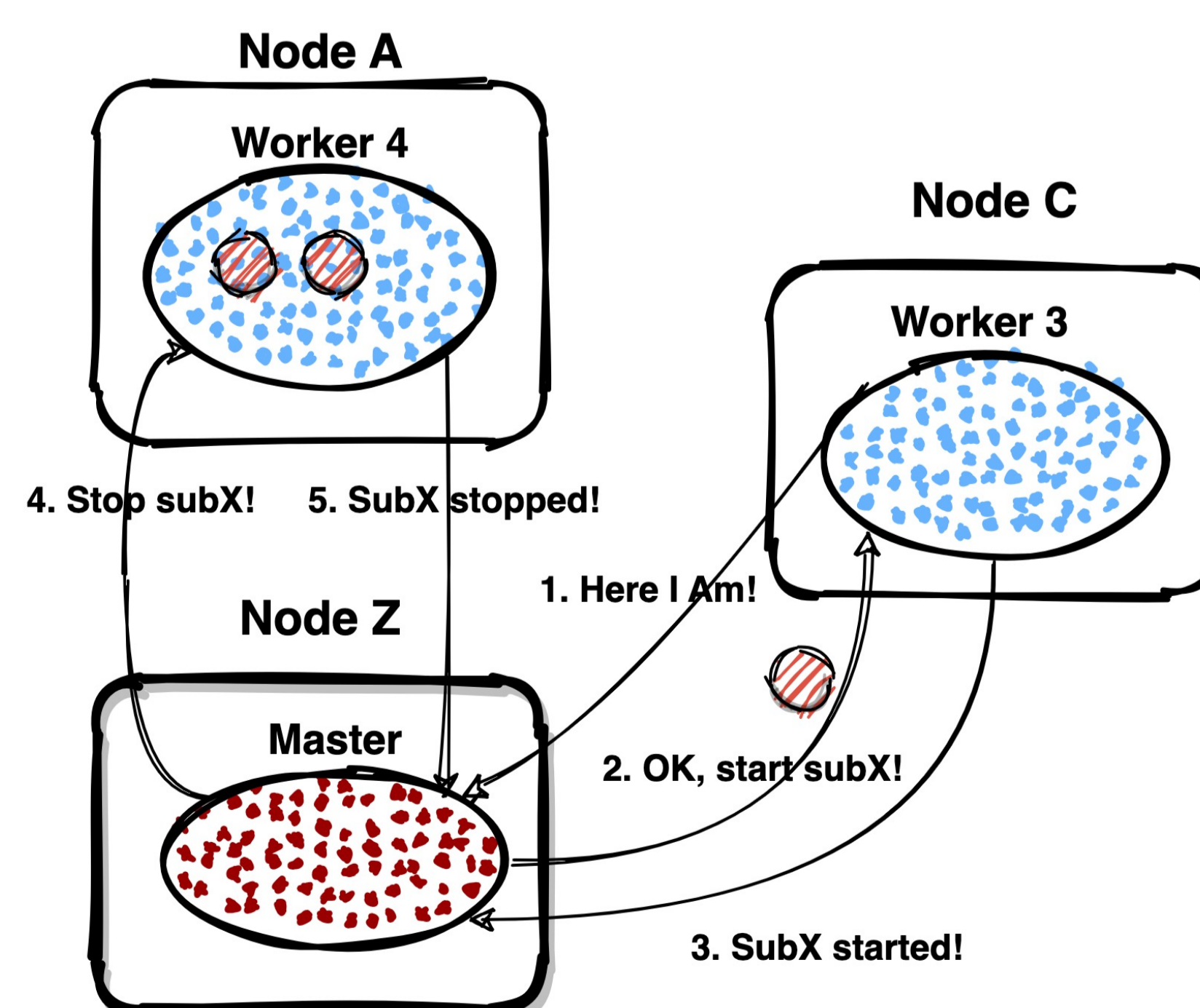
VALIDATION PROTOCOL



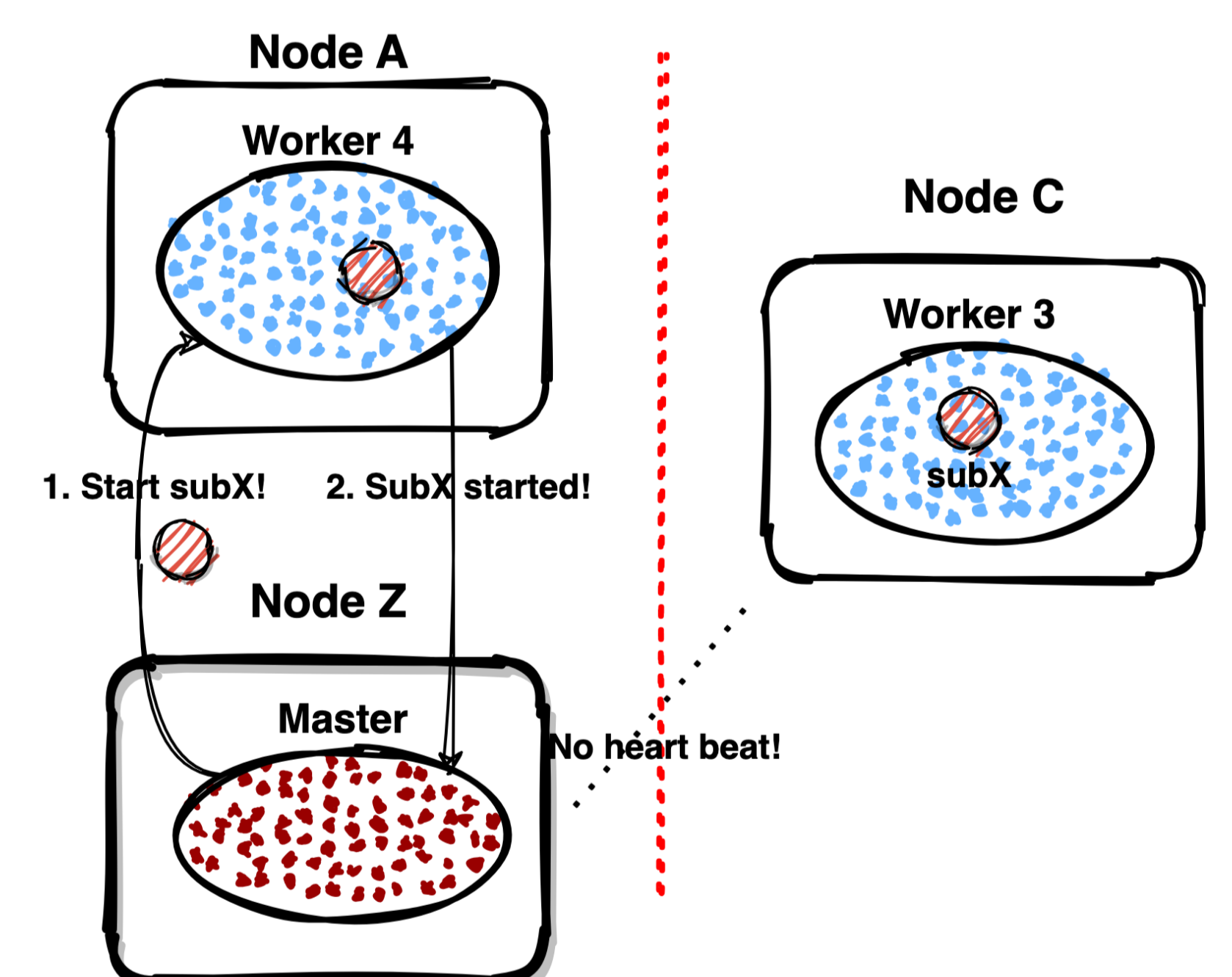
- Inversion of responsibility – validation executor pulls work from dispatcher rather than dispatcher pushing work to the executor.
- Validation executors are not considered members of the cluster therefore, if they fail while validating, the cluster is not affected.
- Inspired by “Balancing Workload Across Nodes with Akka2” by Derek Wyatt.

ADDING NEW WORKER

1. Worker 3 registers with Master.
2. Master sends *subX* to be started on the new Worker 3.
3. Worker 3 confirms *subX* has been started.
4. Master asks Worker 4 to stop *subX*.
5. Worker 4 confirms *subX* has been stopped.



WORKER FAILURE



- Master assumes responsibility of monitoring the availability of all active workers.
- The monitoring is performed using Akka “watching” mechanism.
- If master detects that a worker has become unavailable it reassigns all its subscriptions to other workers that are operational.