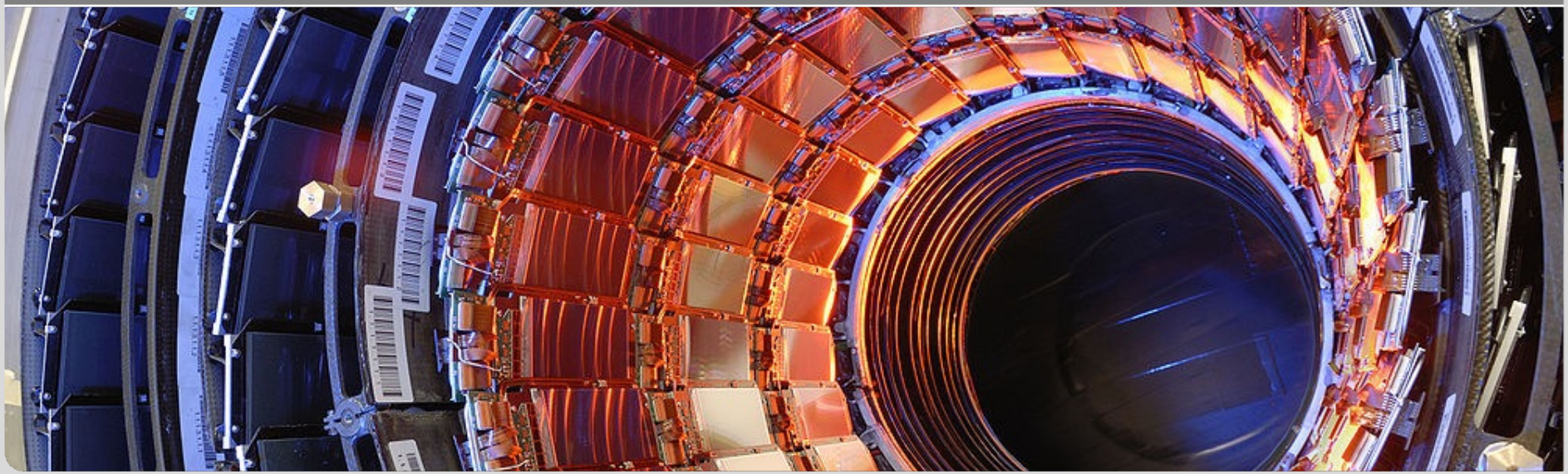
Karlsruhe Institute of Technology

A scalable architecture for online anomaly detection of WLCG batch jobs

Eileen Kuehn, Max Fischer, Manuel Giffels, Christopher Jung, Andreas Petzold

Steinbuch Centre for Computing, GridKa



Payload Monitoring Tool

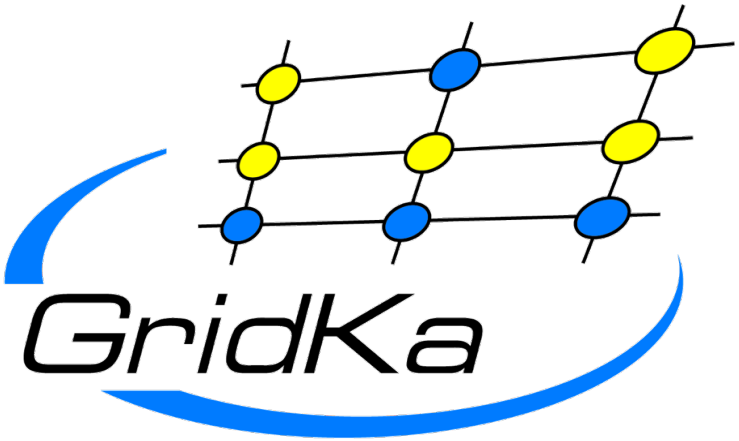
- Process and network traffic monitoring tool
  - Monitoring process trees of batch system jobs
  - Network monitoring based on libpcap
  - Logging of network traffic on packet level
  - Association of process and network data
- Internal preprocessing
  - Categorisation into internal and external network traffic
  - Grouping of traffic by connection

- Stable operation for several months
- Already proven to be useful
  - For example: Detected a class of misbehaving jobs producing loads of external traffic because of a software bug
    - Firewall was heavily loaded and GridKa not reachable
    - Due to fine-grained monitoring, fast identification and reaction on issue

Operation on GridKa Cluster

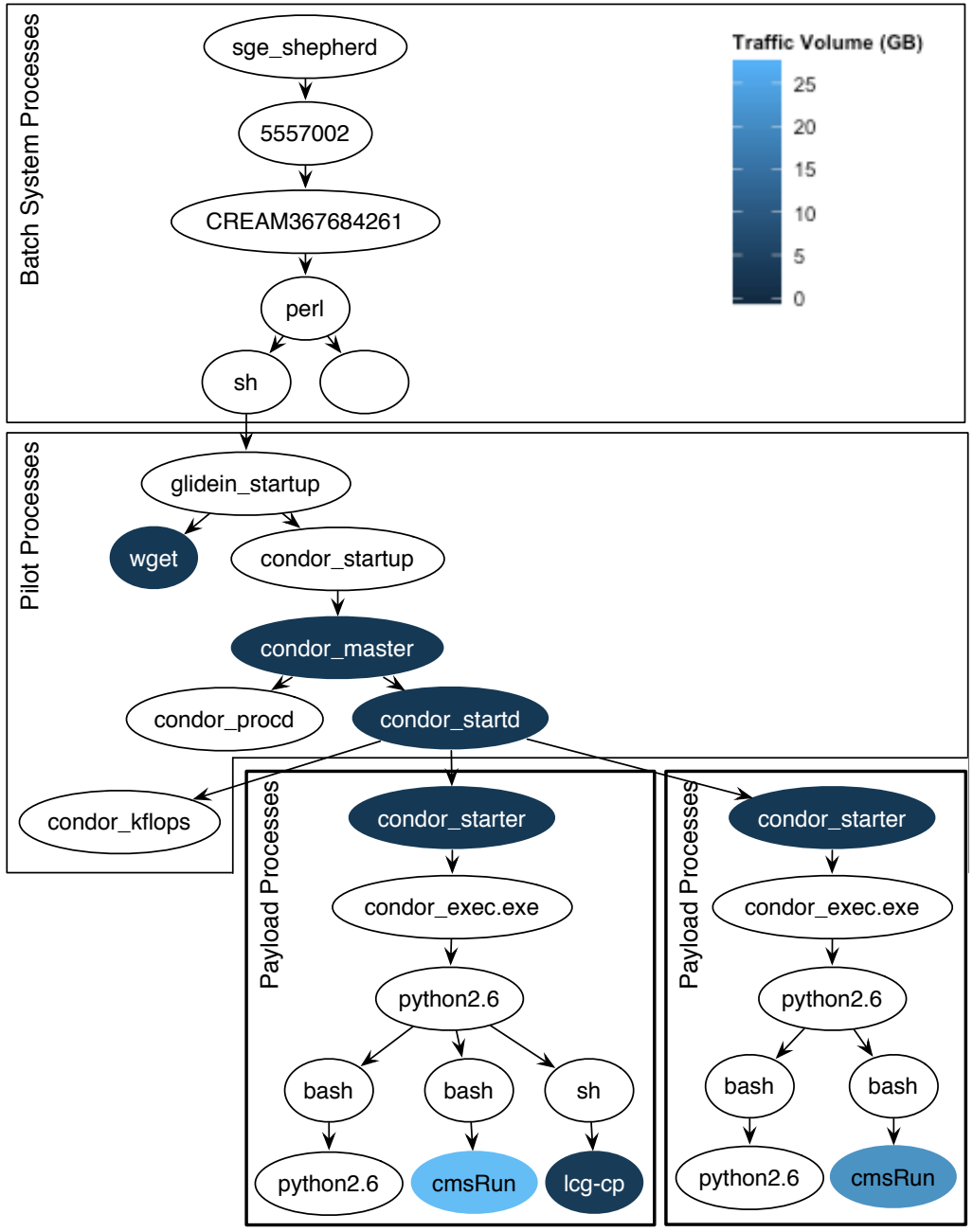
- Tier 1 computing centre
  - Supports all four major CERN LHC collaborations as well as AUGER, BELLE2, COMPASS, and others
  - Operation of ~600 worker nodes
  - Provision of ~13,000 job slots
  - Batch system: Univa Grid Engine

- Prototyp monitoring on 2 racks
  - 2x16 worker nodes, ~800 job slots
  - Stable operation since July ‘14



Example Pilot

- Up to 4 million nodes
- Up to 100 payloads
- Avg. duration 12 h
- Depth: 17
- Process <10 s cut



Issues

- Memory demands
  - Anomaly detection based on trees
  - Anomaly detection based on traffic time-series
- Algorithmic complexity
- Learning of prototypes

Unique encoding of edges

	Data Type	Size (B)
pid	`int`	4
ppid	`int`	4
gpid	`int`	4
uid	`int`	4
tme	`real`	4
exit_tme	`real`	4
exit_code	`int`	4
name	`string`	l_1
cmdline	`string`	l_2
		$28 + l_1 + l_2$

Memory requirements of single node (B)

Issues

- Memory demands
  - Anomaly detection based on trees
  - Anomaly detection based on traffic time-series
- Algorithmic complexity
- Learning of prototypes

Unique encoding of edges

Tree with 4 million nodes requires around 160 MB for values

	Data Type	Size (B)
pid	`int`	4
ppid	`int`	4
gpid	`int`	4
uid	`int`	4
tme	`real`	4
exit_tme	`real`	4
exit_code	`int`	4
name	`string`	l_1
cmdline	`string`	l_2
		$28 + l_1 + l_2$

Memory requirements of single node (B)

Issues

- Memory demands
  - Anomaly detection based on trees
  - Anomaly detection based on traffic time-series
- Algorithmic complexity
- Learning of prototypes

Unique encoding of edges

Tree with 4 million nodes requires around 160 MB for values

24 on each worker node: 3.75 GB

	Data Type	Size (B)
pid	`int`	4
ppid	`int`	4
gpid	`int`	4
uid	`int`	4
tme	`real`	4
exit_tme	`real`	4
exit_code	`int`	4
name	`string`	l_1
cmdline	`string`	l_2
		$28 + l_1 + l_2$

Memory requirements of single node (B)

Issues

- Memory demands
  - Anomaly detection based on trees
  - Anomaly detection based on traffic time-series
- Algorithmic complexity
- Learning of prototypes

Unique encoding of edges

Tree with 4 million nodes requires around 160 MB for values

24 on each worker node: 3.75 GB

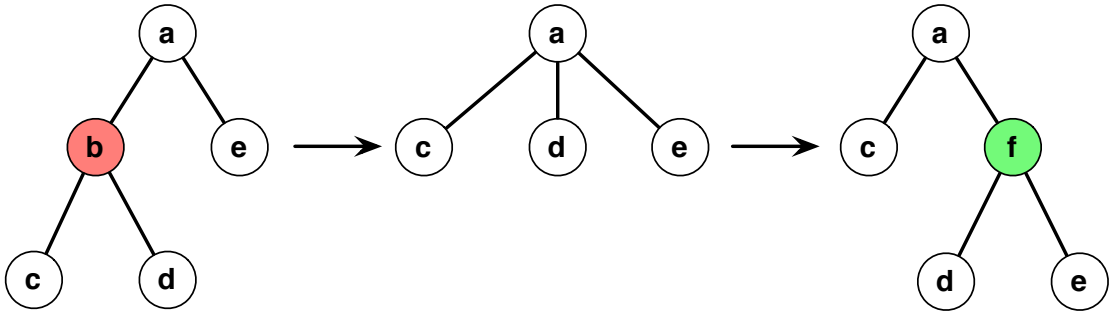
Additional memory for prototypes

	Data Type	Size (B)
pid	`int`	4
ppid	`int`	4
gpid	`int`	4
uid	`int`	4
tme	`real`	4
exit_tme	`real`	4
exit_code	`int`	4
name	`string`	l_1
cmdline	`string`	l_2
		$28 + l_1 + l_2$

Memory requirements of single node (B)

Issues

- Memory demands
- Algorithmic complexity
  - Similarity of trees
  - Handling of noise (e.g. repetition of processes, different software versions)
  - Correlation with network saturation (e.g. runtime of processes)
- Learning of prototypes



Example: Tree edit distance (e.g. $O(n^3)$ time and $O(mn)$ space complexity*)

* M.Pawlik and N.Augsten. RTED: A Robust Algorithm for the Tree Edit Distance. PVLDB. 2011.

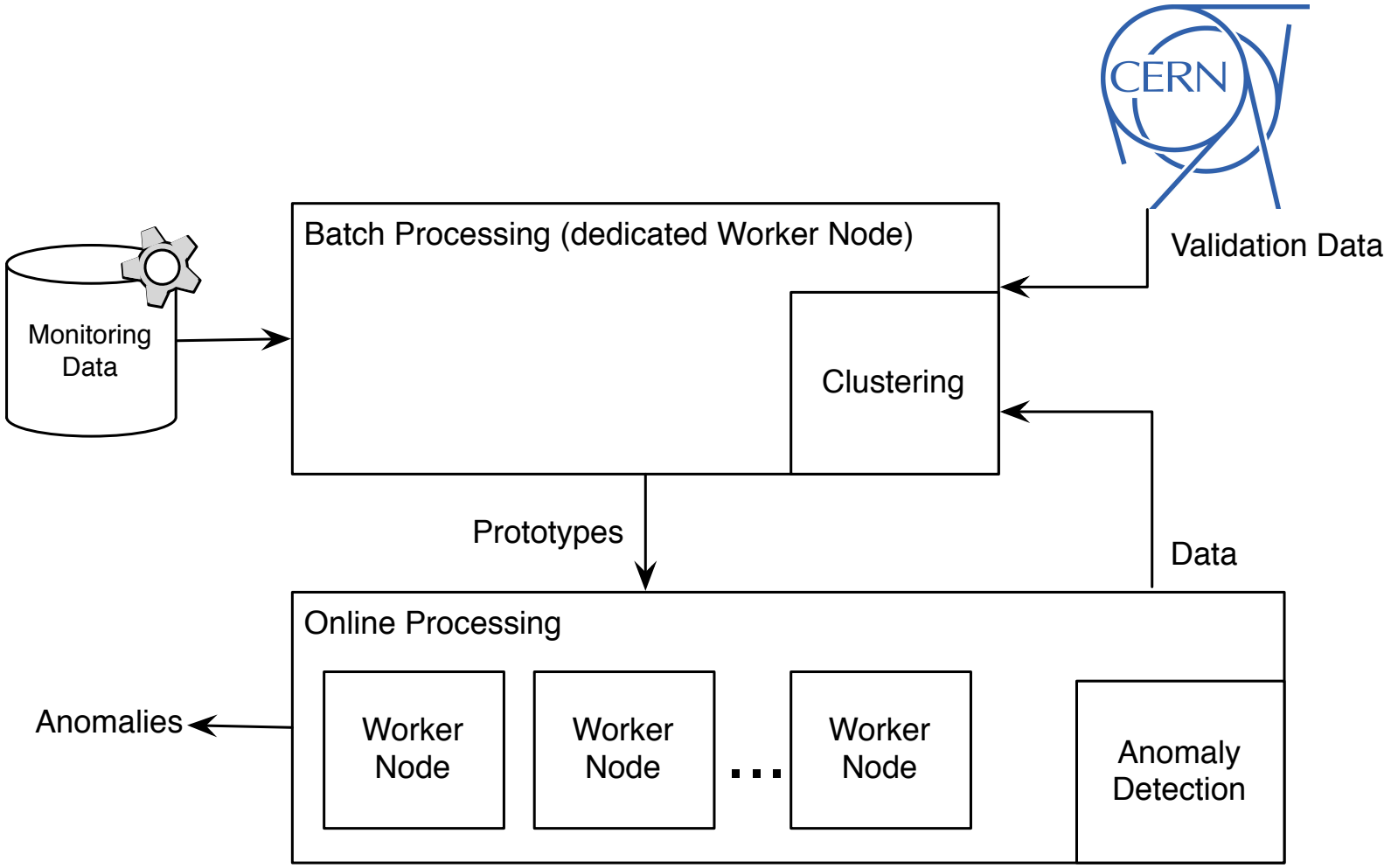
Issues

- Memory demands
- Algorithmic complexity
- Learning of prototypes
  - Centralised vs distributed learning
  - Evolution of batch jobs

Requirements Online Anomaly Detection

- Uninterrupted operation of batch system
  - Low communication overhead
  - Low memory consumption and CPU load on worker nodes
- Fast and accurate near real-time anomaly detection
  - Focus on reduction of false positive notifications
- Robust approach
- Unsupervised approach
- Scalability

Overview Online Anomaly Detection

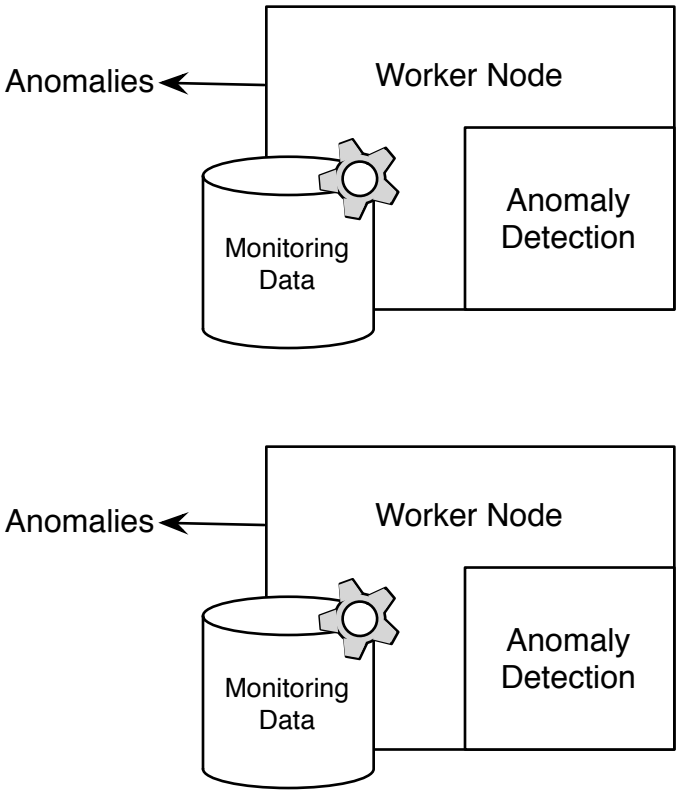


Local Anomaly Detection

- Each worker node does the whole detection workflow

- Advantages
  - No communication overhead

- Disadvantages
  - High CPU load on worker nodes
  - Context is ignored
  - Prototype learning
    - Different models on each node
    - Central node for model calculation

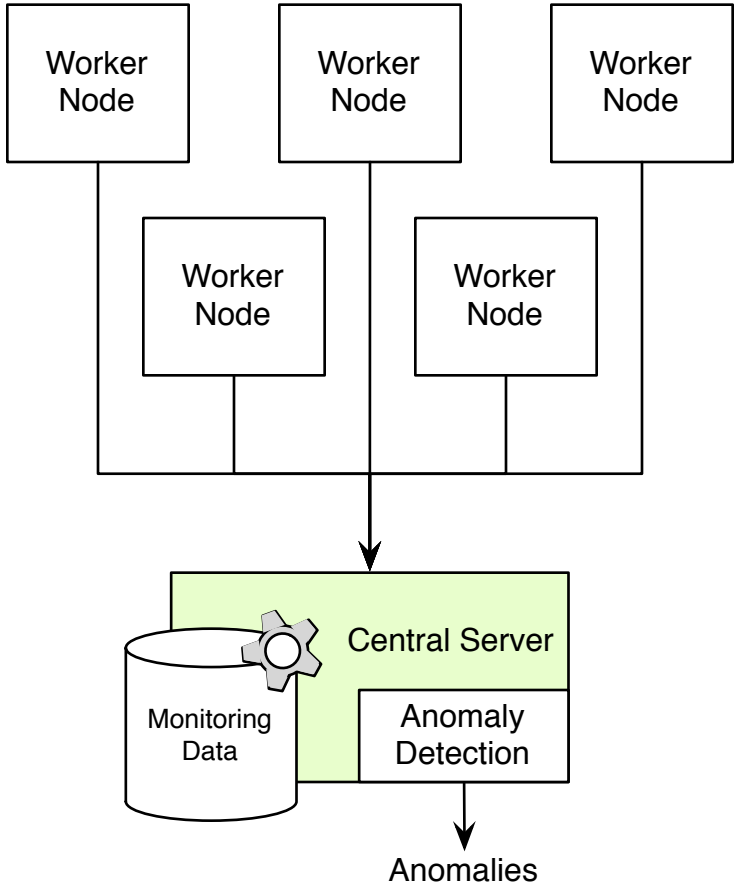


Centralised Systems

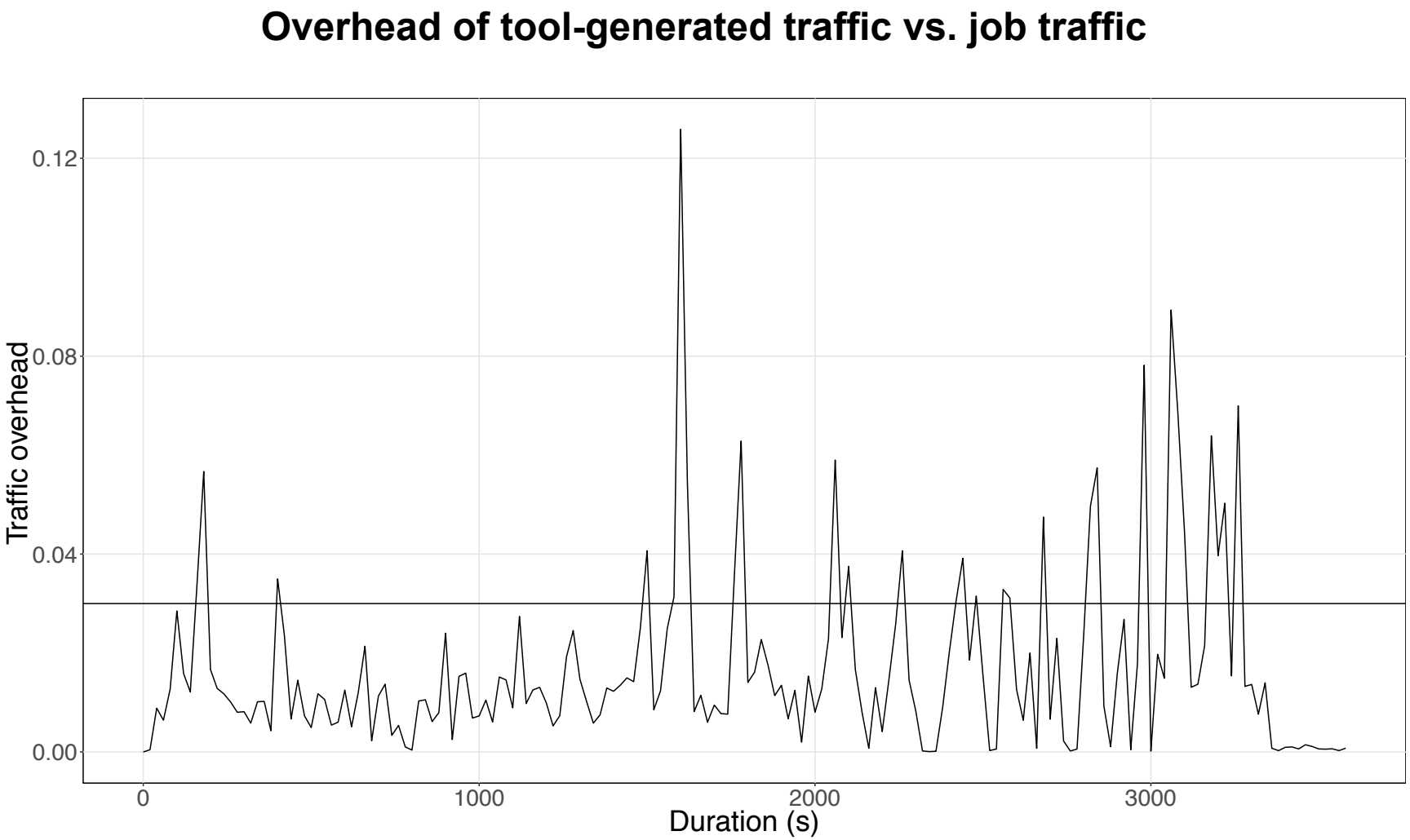
- Anomaly detection centralised on single server
- Worker nodes send monitoring information

- Advantages
  - Easy to manage
  - Data consistency and coherence
  - Centralised prototype model

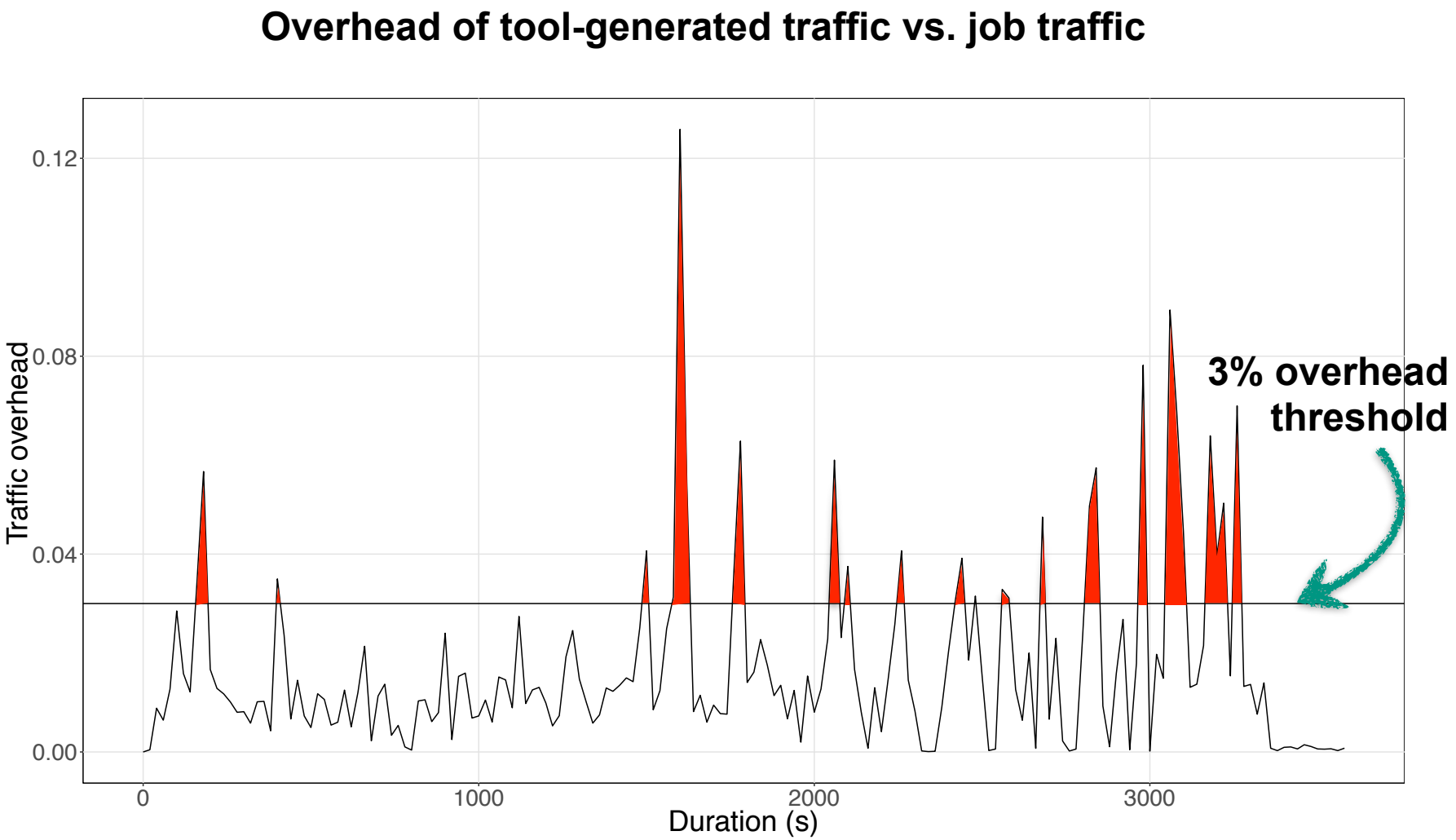
- Disadvantages
  - High CPU load on server
  - High memory demands on server
  - Communication overhead
  - Single point of failure
  - Server is bottleneck to scalability and performance



Centralised Systems

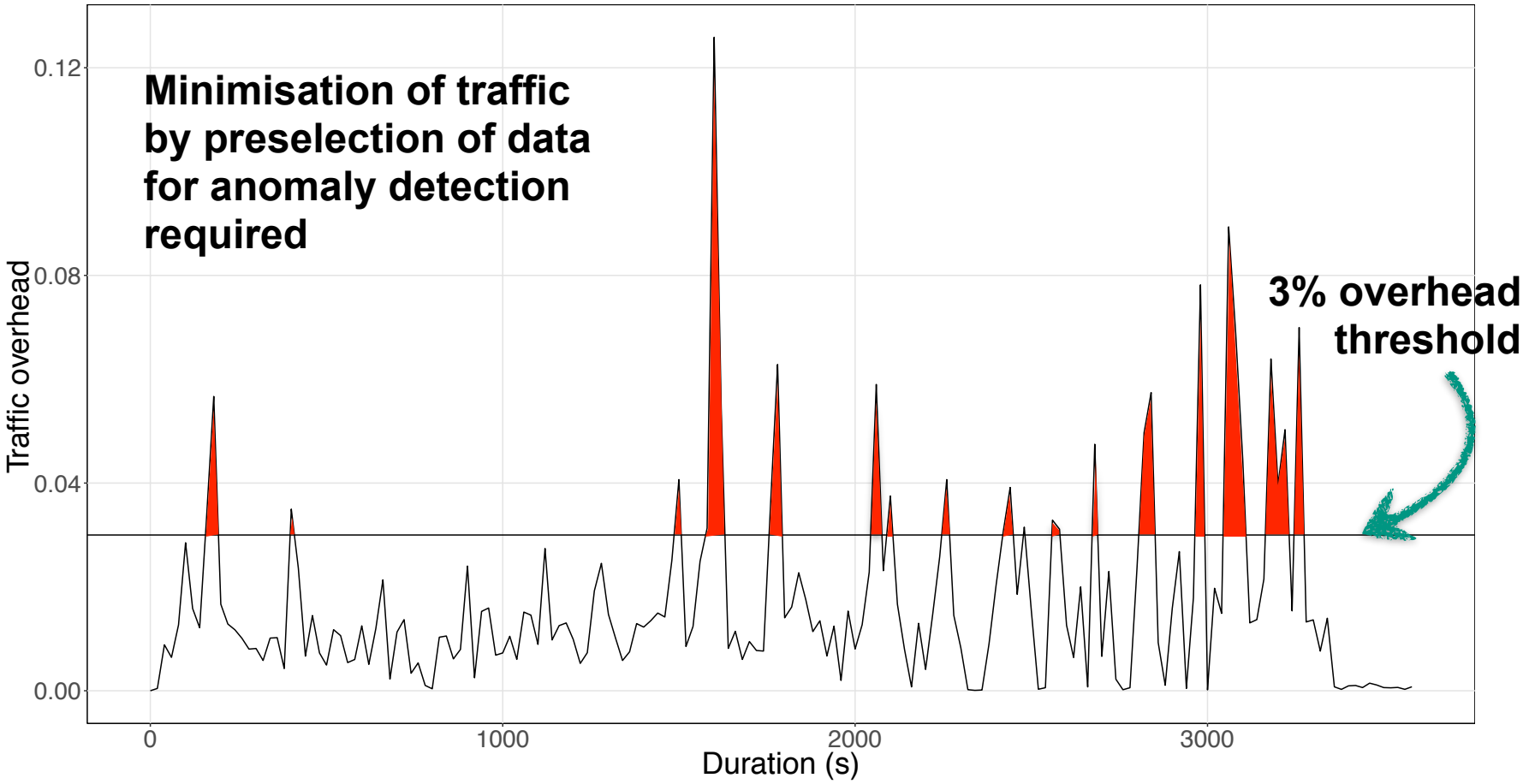


Centralised Systems



Centralised Systems

Overhead of tool-generated traffic vs. job traffic

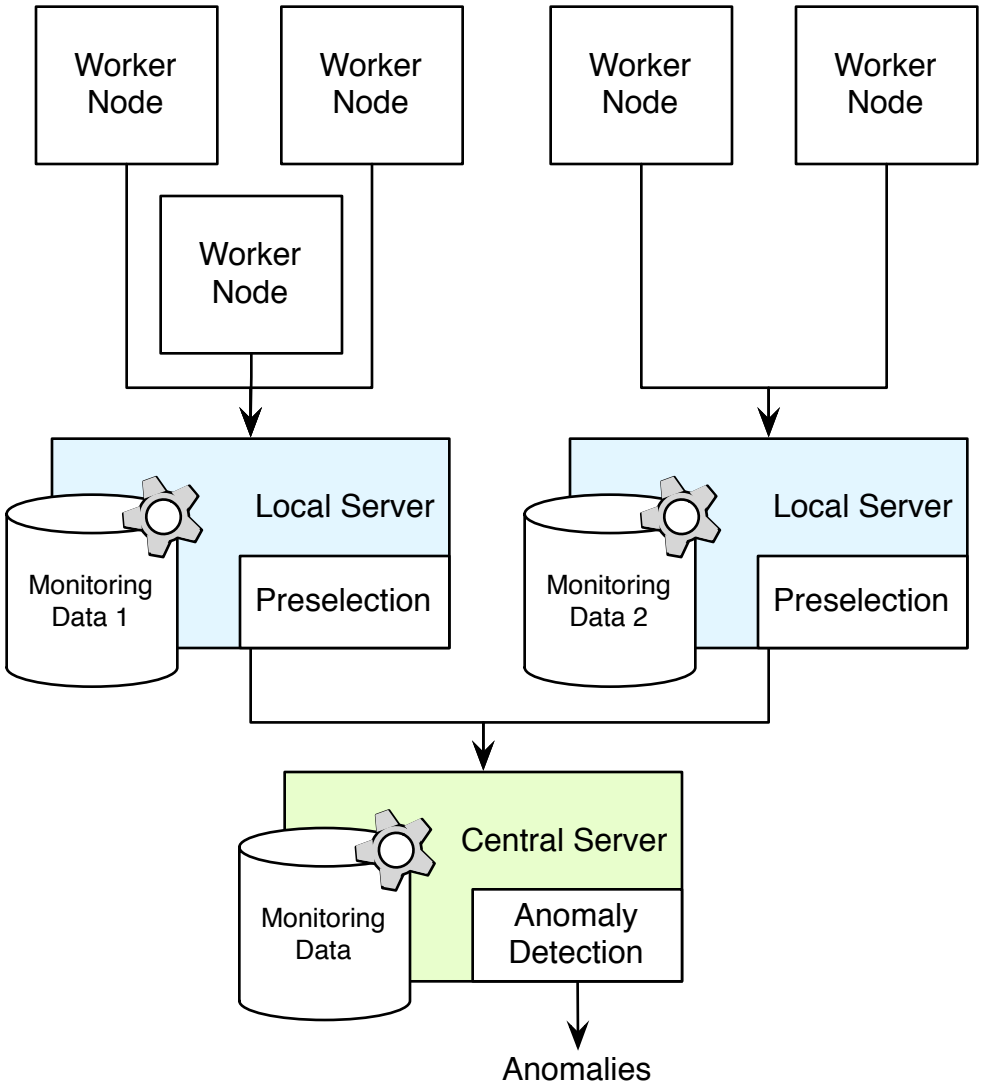


Hierarchical Systems

- Allocation of a group of worker nodes to several servers
- Logical hierarchy of processing

- Advantages
  - Scalability
  - Data consistency and coherence
  - Centralised prototype model

- Disadvantages
  - Hardware requirements for servers
  - Communication overhead
  - Single groups may fail

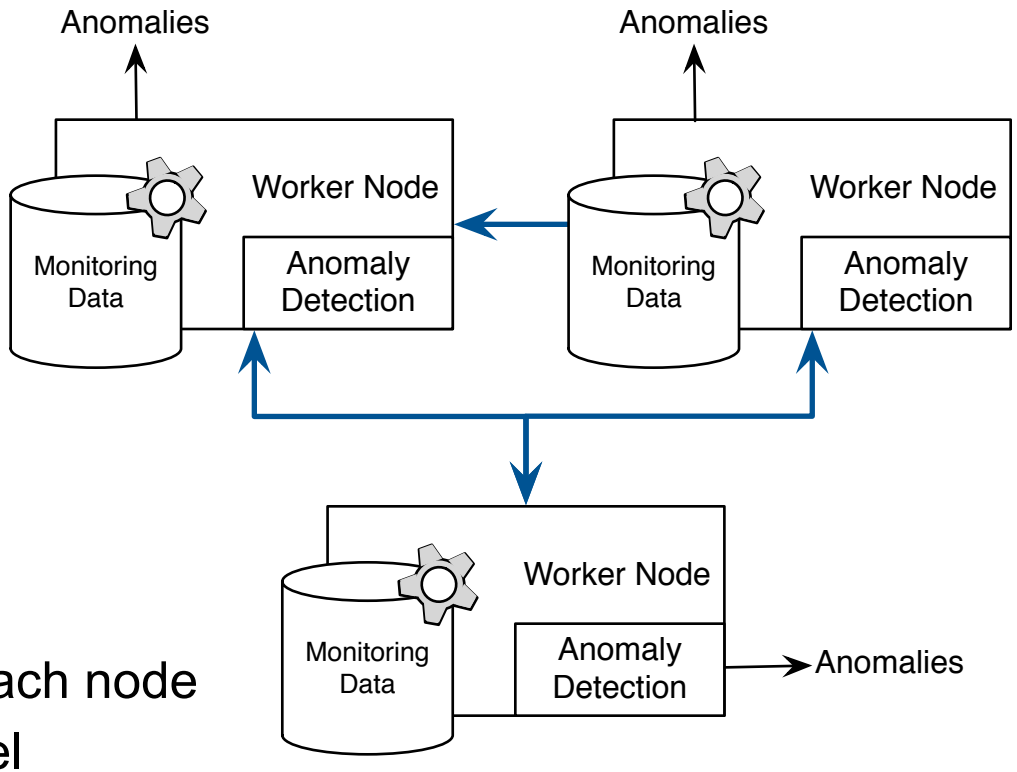


Peer-to-Peer Systems

- Any node can exchange data with any other node

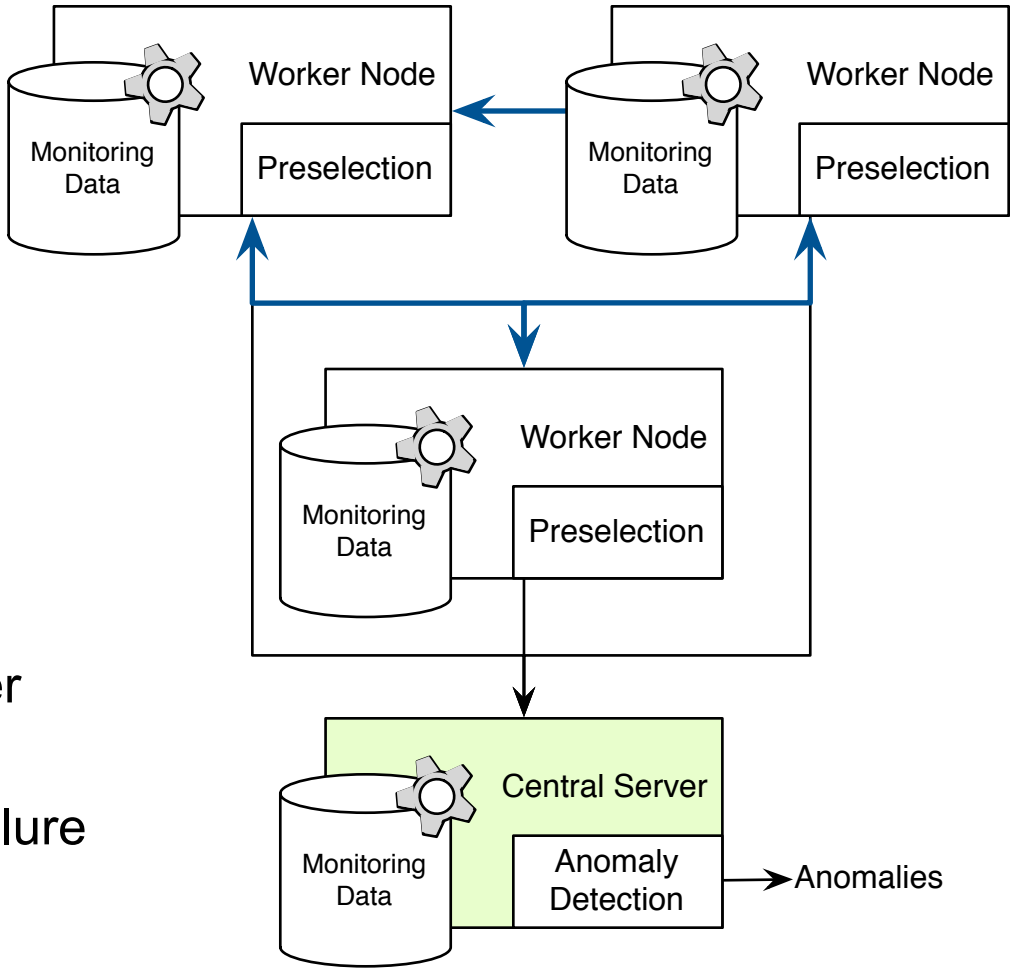
- Advantages
  - Fault tolerance to failure/shutdown
  - Horizontal scalability

- Disadvantages
  - Manageability
  - Network flooding
  - High CPU load
  - Prototype learning
    - Different models on each node
    - Central node for model calculation



Hybrid Peer-to-Peer Systems

- Additional central server to gather required information
- Advantages
  - Data consistency and coherence
  - Centralised prototype model
- Disadvantages
  - CPU load on server
  - Memory demands on server
  - Communication overhead
  - Server as single point of failure and bottleneck

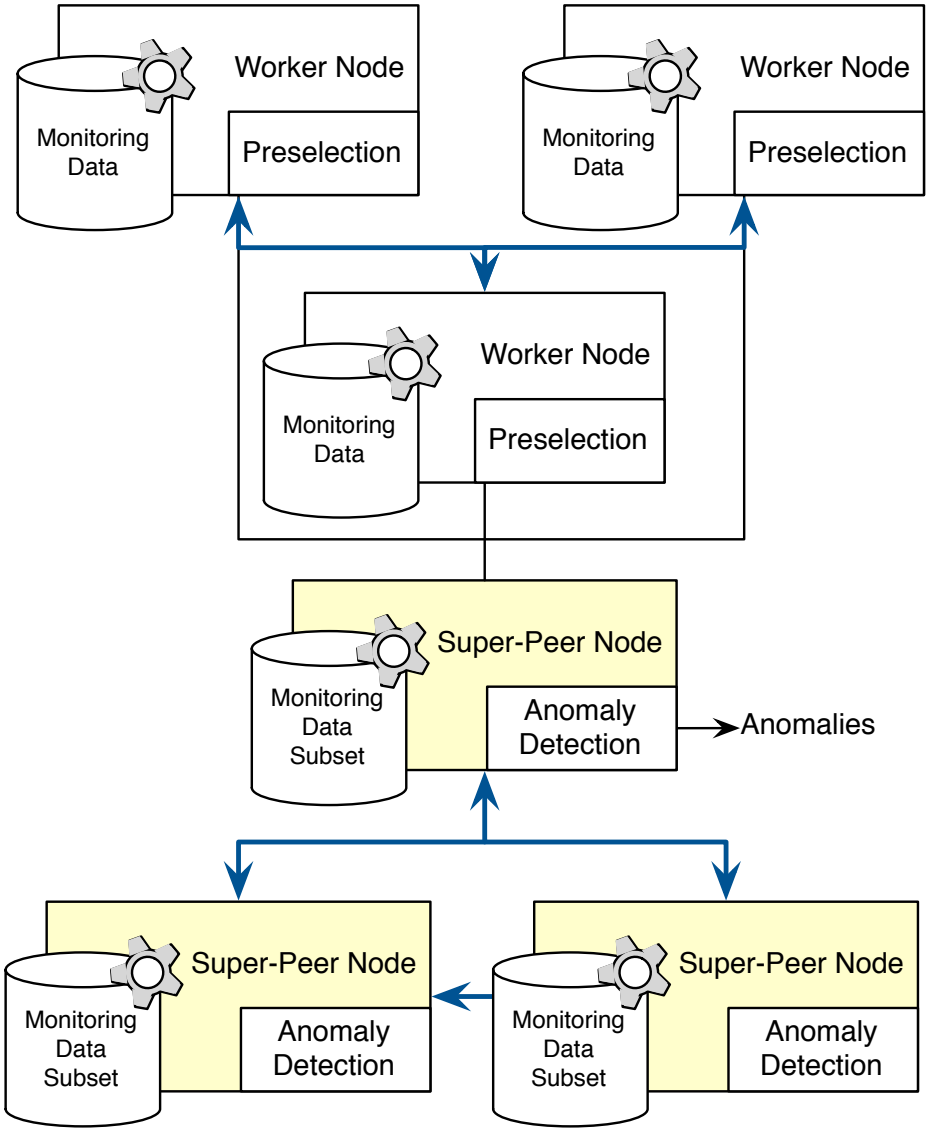


Super-Peer Systems

- Super-peers act as centralised servers to a subset of worker nodes
- Super-peers are connected to each other as peers
- Dynamic assignment of responsibilities

- Advantages
  - Flexible resource utilisation
  - Autonomous units

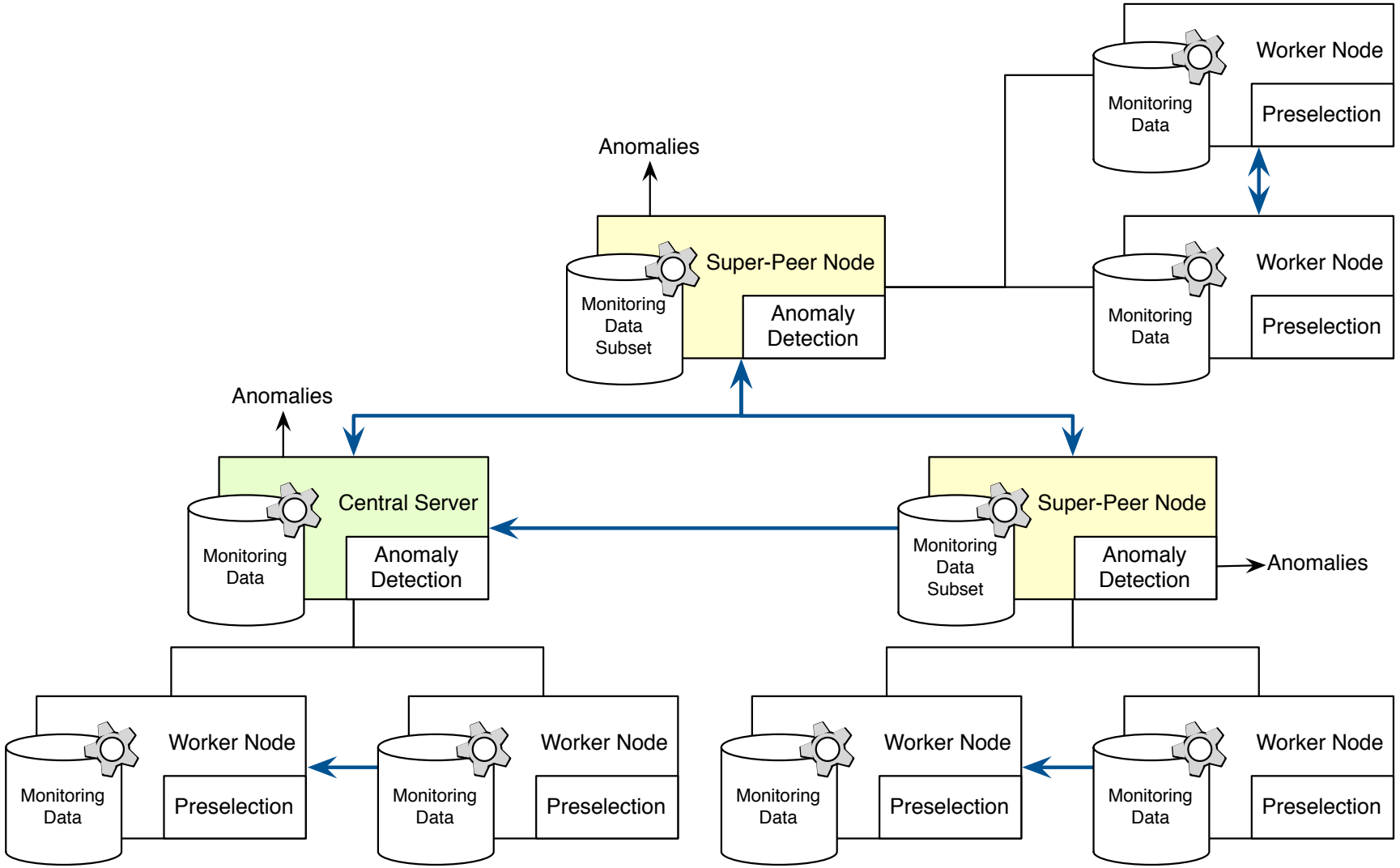
- Disadvantages
  - Prototype learning
  - Communication overhead



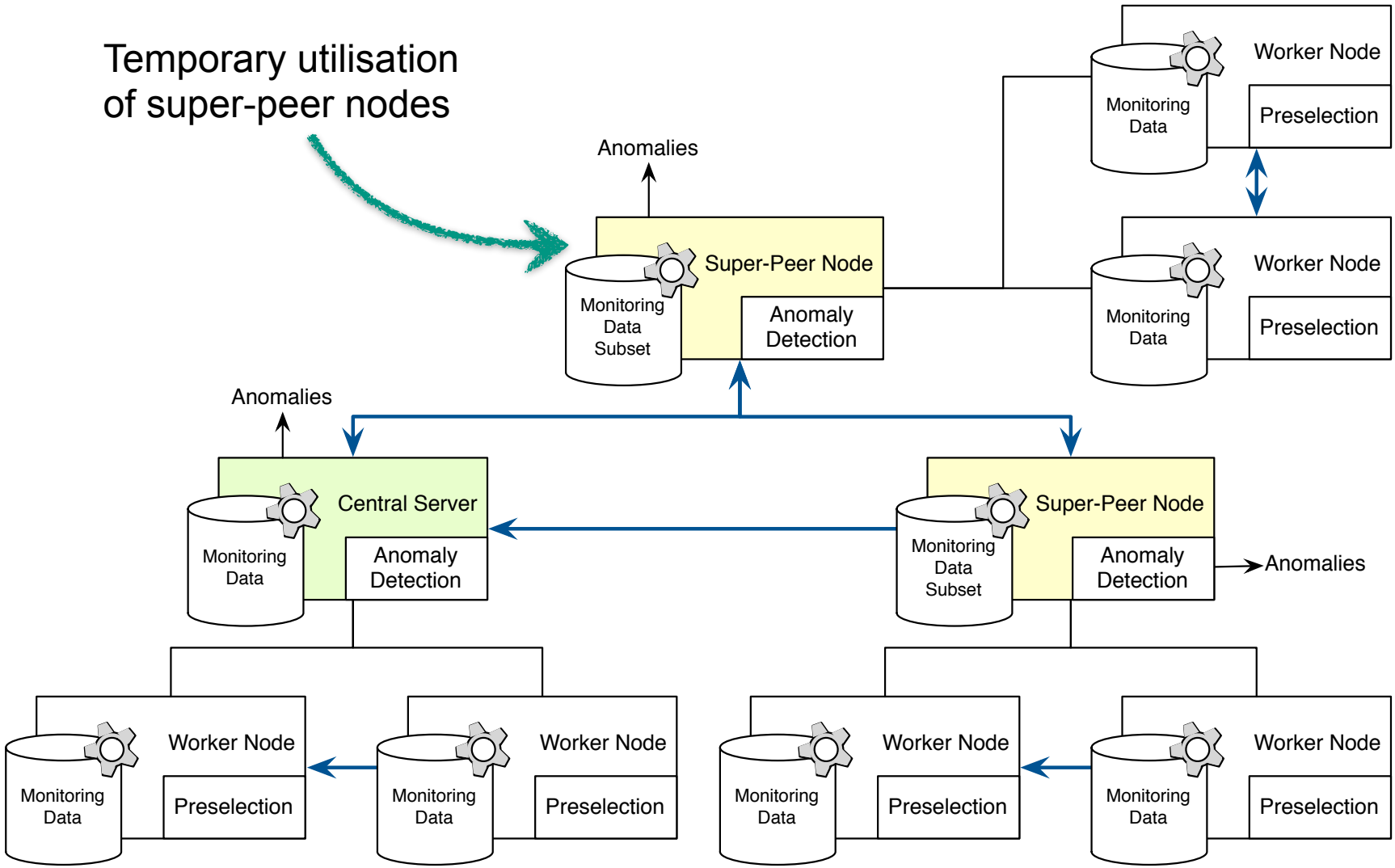
Concept of Scalable Architecture

- High computational cost for distance measurement but also memory demands require distributed approach
- High flexibility and heterogeneity of batch system require fault tolerant autonomous units/cluster
- Monitoring allows detection of unsaturated worker nodes
- To reduce communication overhead algorithm has modular design

Concept of Scalable Architecture

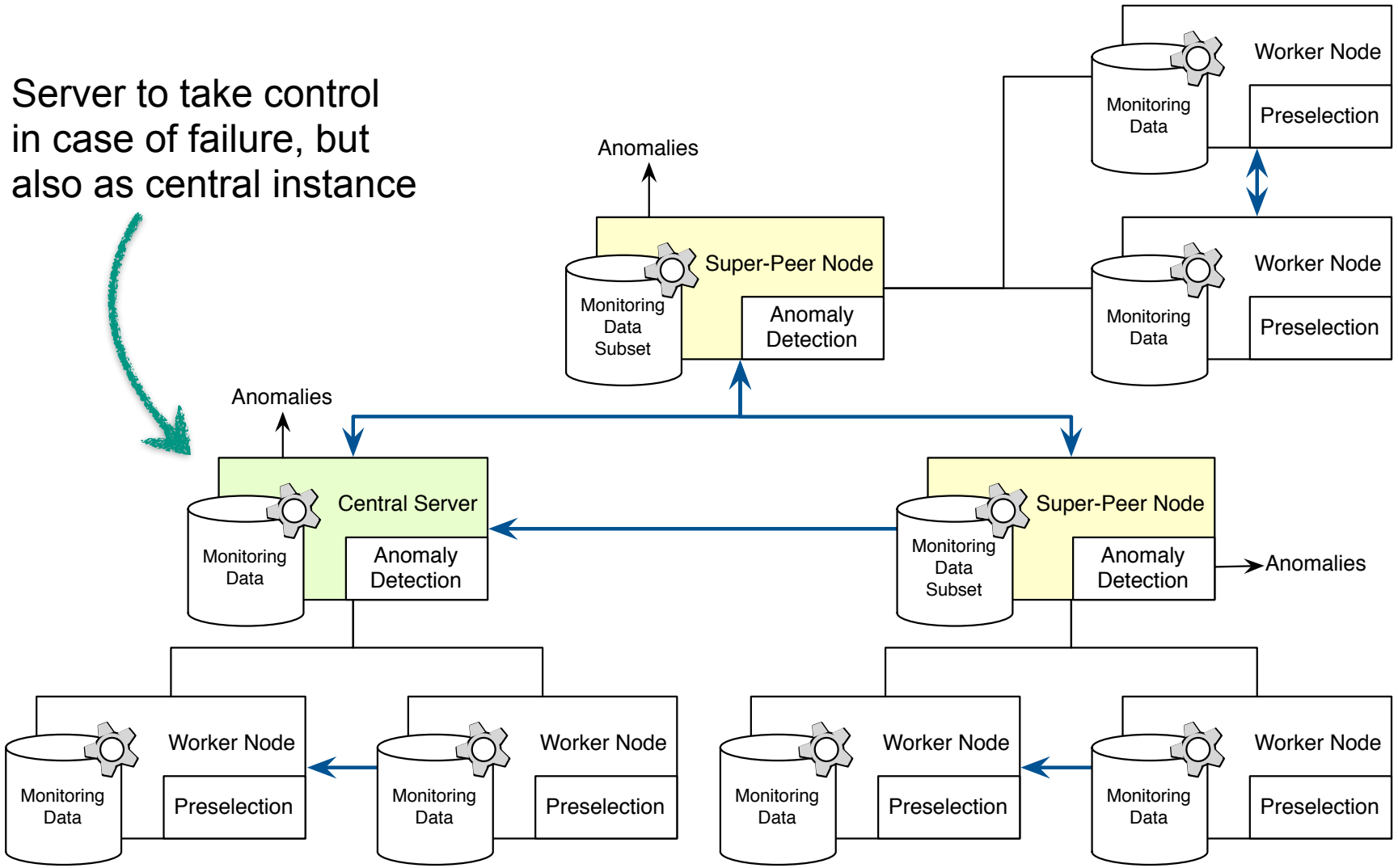


Concept of Scalable Architecture

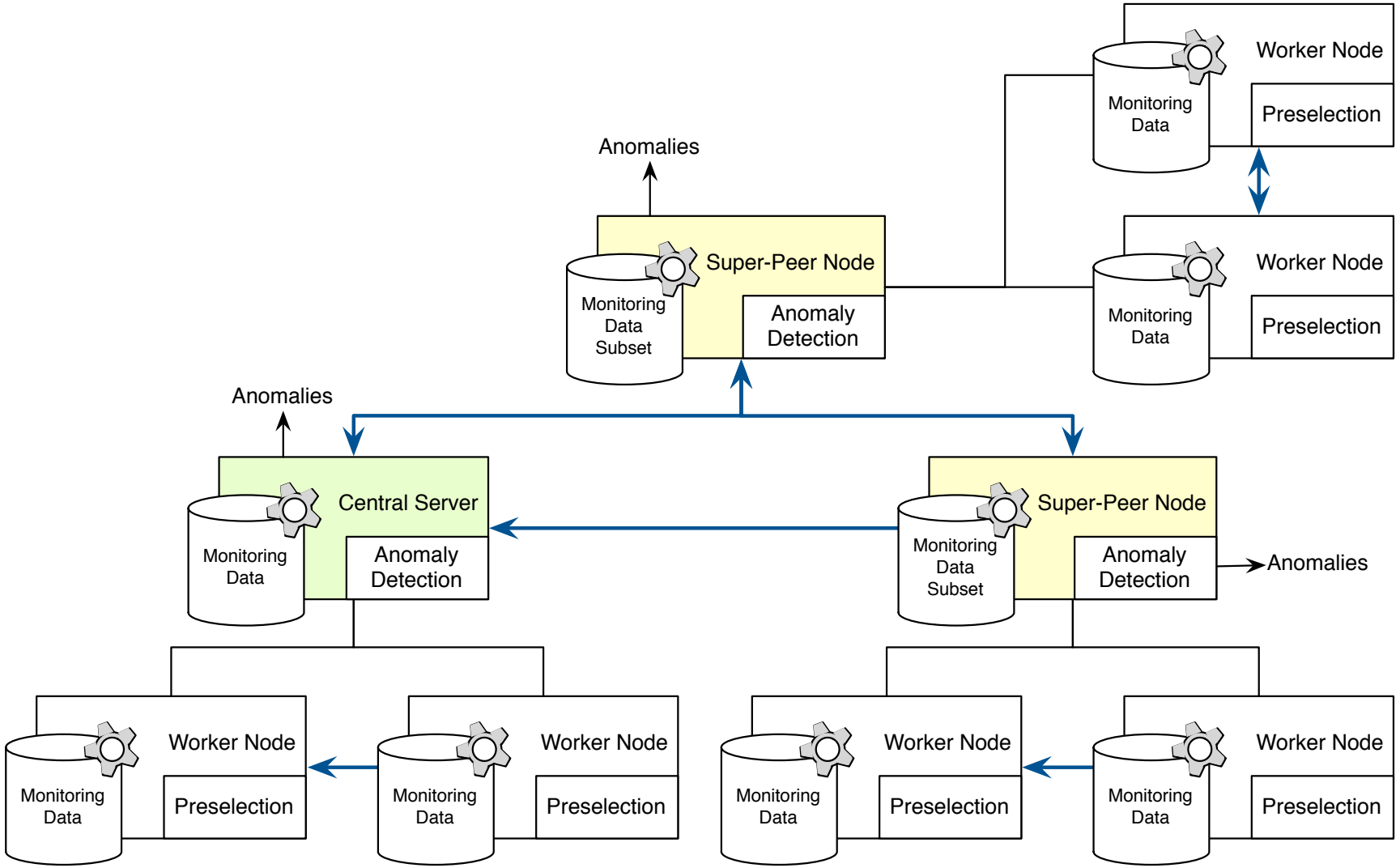


Concept of Scalable Architecture

Server to take control
in case of failure, but
also as central instance



Concept of Scalable Architecture



Summary and Outlook

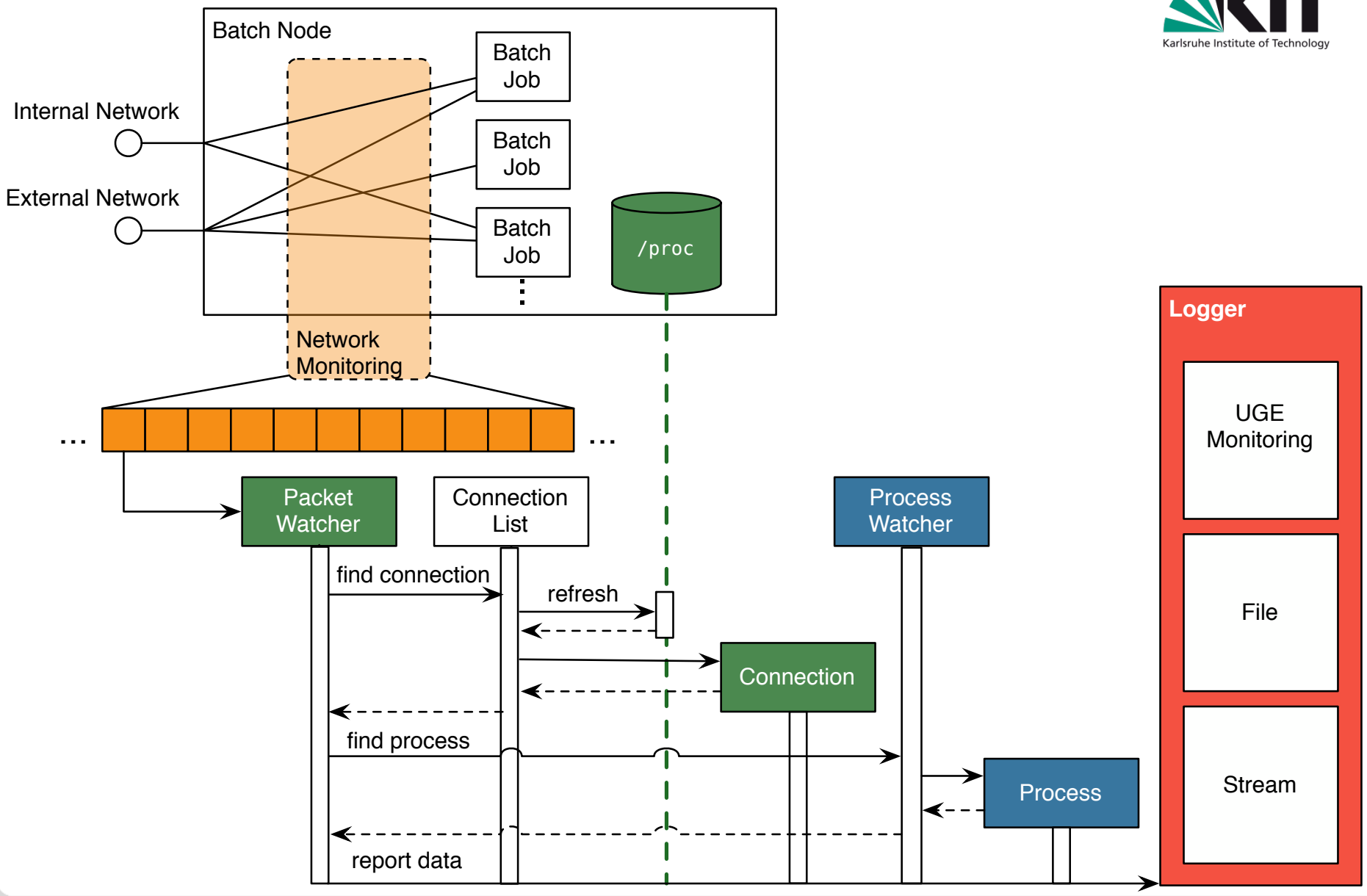
- Scalable and modular architecture for anomaly detection of WLCG batch jobs
  - Based on batch job-level monitoring on single worker nodes
  - Different contexts (e.g. same/different rack, or computing centre)
- Opportunistic resource usage
  - Unsaturated worker nodes used as super-peers
  - Additional server as a failover super-peer node

- Further improvements
  - Reduction of communication overhead
    - Compression of data formats
    - Improved preselection
  - Reduction of CPU load
    - Incremental distance measurement
  - Reduction of memory demands

Summary and Outlook

- Scalable and modular architecture for anomaly detection of WLCG batch jobs
  - Based on batch job-level monitoring on single worker nodes
  - Different contexts (e.g. same/different rack, or computing centre)
- Opportunistic resource usage
  - Unsaturated worker nodes used as super-peers
  - Additional server as a failover super-peer node

Questions?


Batch Node
Batch Job
Batch Job
Batch Job
Internal Network
External Network
/proc
Network Monitoring
Logger
UGE Monitoring
File
Stream
Packet Watcher
Connection List
Process Watcher
find connection
refresh
Connection
find process
Process
report data