GRID PROTECTION ALLIANCE

GPA Synchrophasor Products Failover & Minimizing Downtime

Tools for redundant configuration

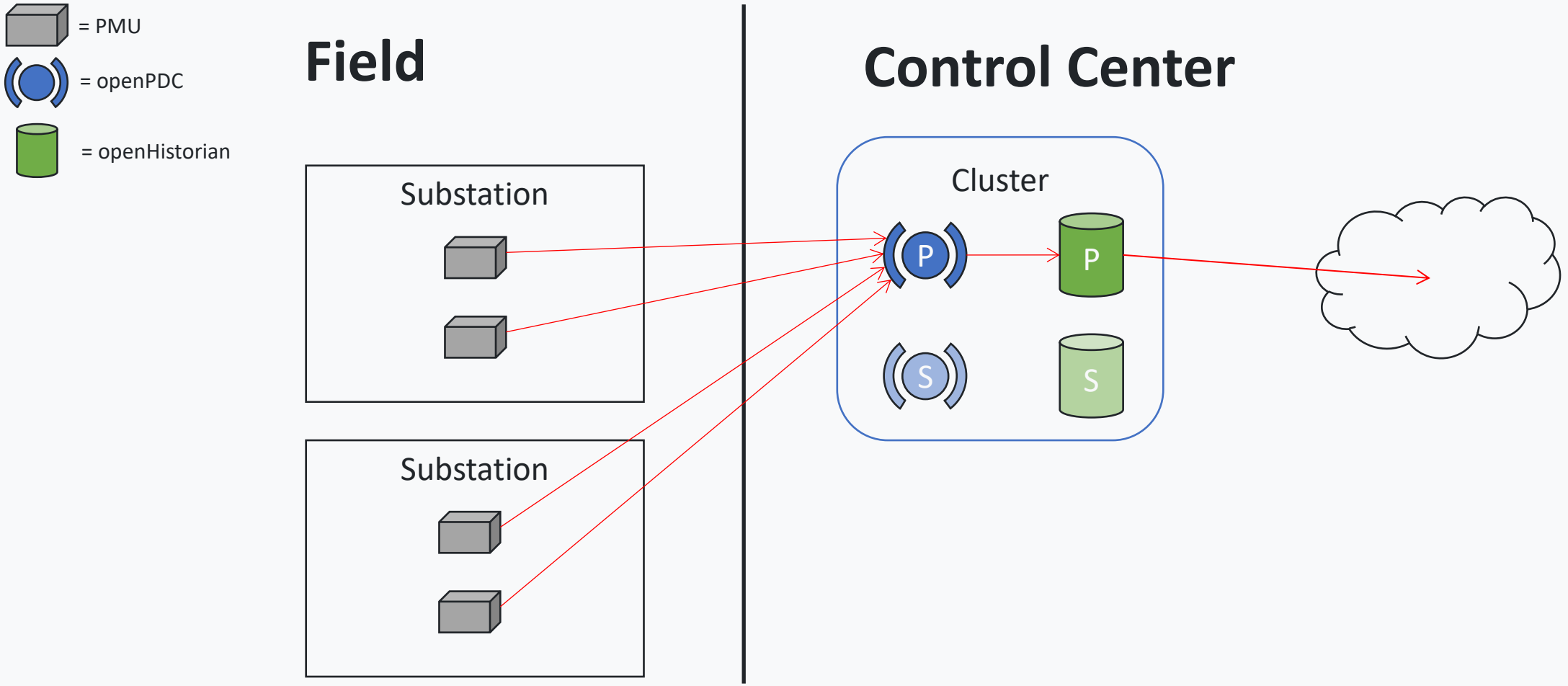
- Windows failover clustering
- openHistorian automatic failover
- Active/active configuration
- Data gap recovery
- Database mirroring

Windows Failover Clustering

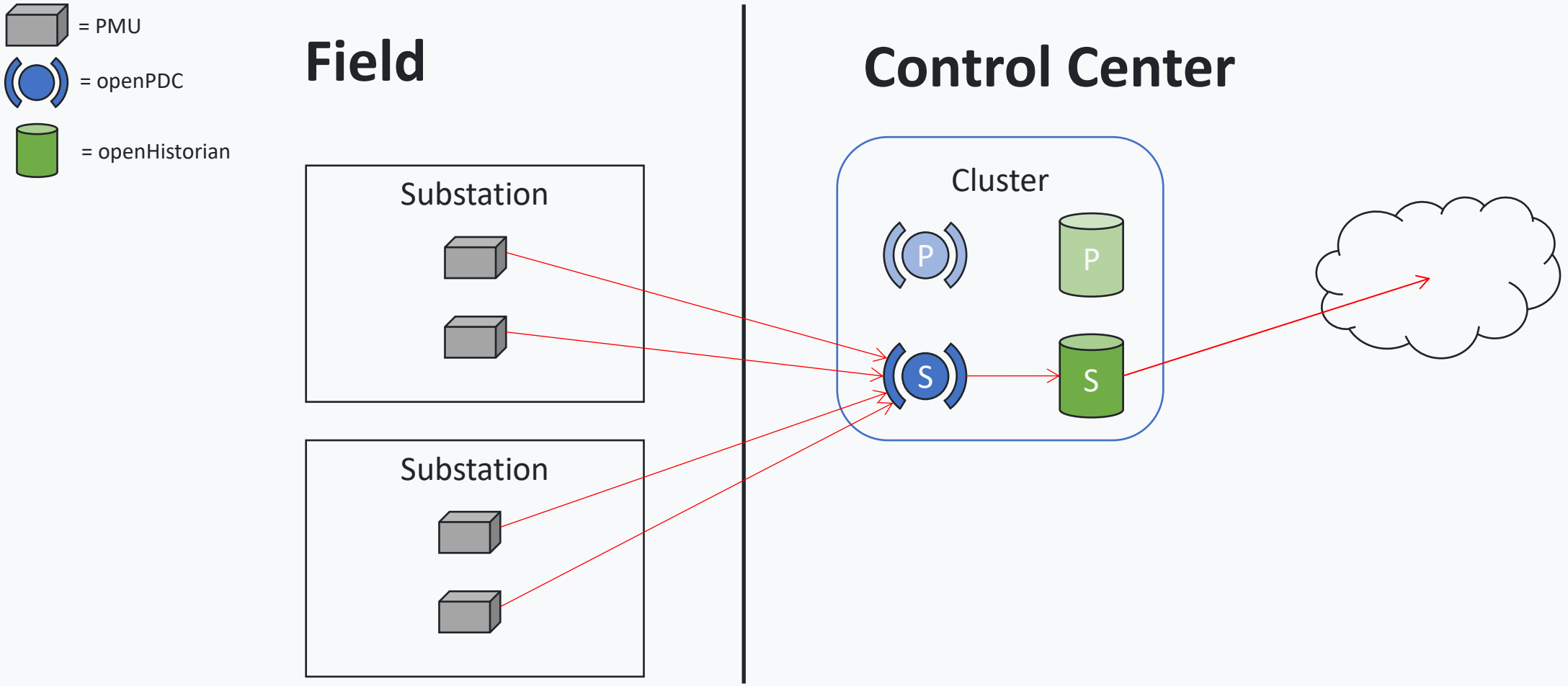
- Simplified cluster management
  - openPDC/openHistorian services can be easily failed over to another server
  - Failover can be performed automatically to keep the system running without user intervention
- Can include SQL Server replication/failover
  - SQL Server instances can be hosted on the same servers as openPDC/openHistorian

- Additional configuration and complexity
- Requires shared storage (cluster witness)
- Momentary downtime when switching from primary to secondary

Windows Failover Clustering



Windows Failover Clustering



openHistorian automatic failover

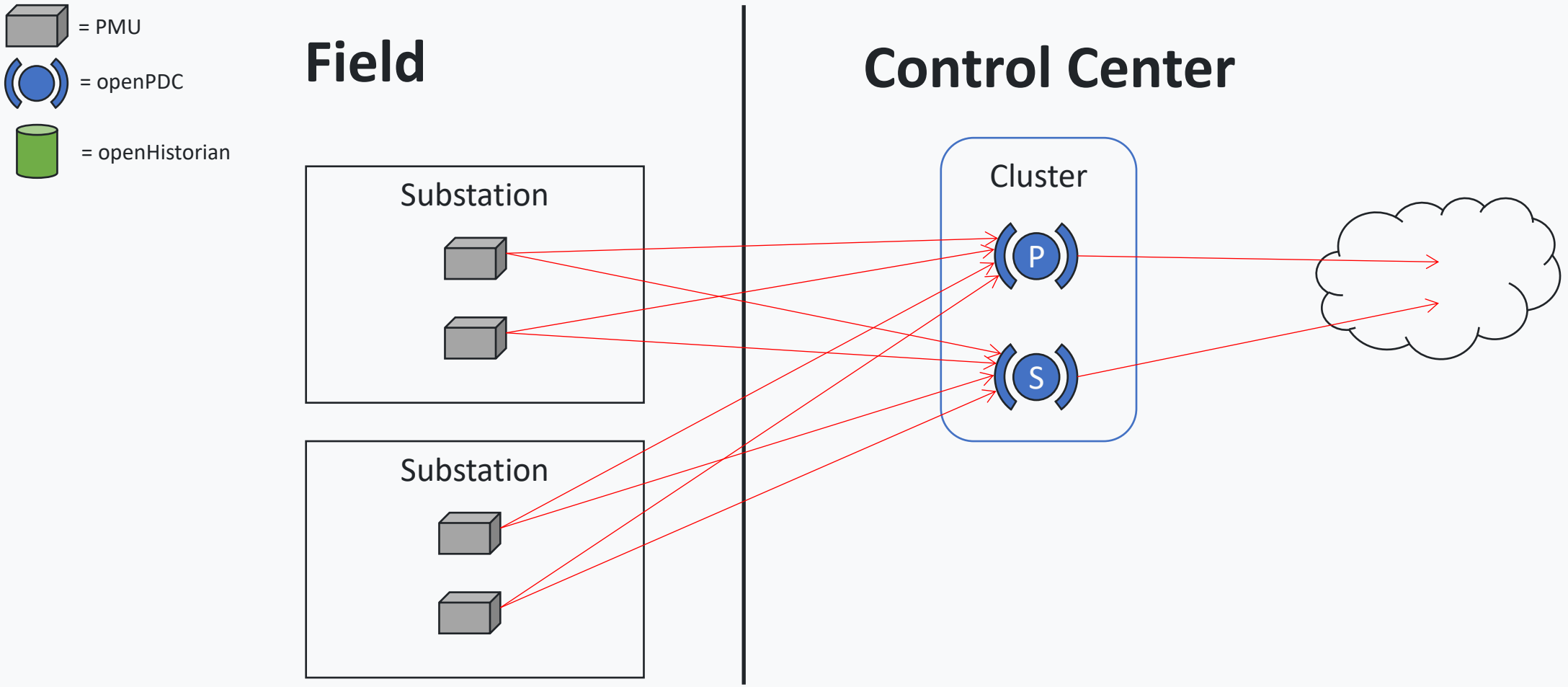
- Each node has three configuration parameters
  - Node type (primary/secondary)
  - Restart delay (seconds)
  - Endpoint
- At startup, the primary node attempts to force the secondary node to fail over
- The secondary node periodically checks the status of the primary node and will start itself up if the primary node is not running
- The endpoint URL is the web interface of the other node in the cluster, used for monitoring status and executing forced failover

Active/active configuration

- No cluster management
- No downtime when bringing nodes offline

- Requires additional equipment (stream splitter) if PMUs do not support multiple simultaneous connections
- Two copies of the real-time data
  - Requires more bandwidth and CPU resources downstream
  - Requires data concentrator for downstream analytics
- This approach is not entirely applicable to openHistorian which requires exclusive access to the location where it writes archive files

Active/active configuration



Data gap recovery

- Upstream short-term archive
- Downstream outage detection
- Stream missing data from the upstream archive to the downstream system after an outage
- Downstream applications that rely on historical data analysis can benefit from having a complete archive despite momentary outages due to failover

Database mirroring

- The SQL database contains configuration used by downstream applications, such as device IDs and measurement IDs, that may need to be mirrored to avoid partitioning data when the system fails over
- Options for mirroring:
  - SQL Server replication
  - Sharing the same database
  - STTP metadata exchange

Degradation (SQL database)

- openPDC and openHistorian operate using a runtime configuration copied from the SQL database into application memory
- Runtime configuration is cached to the local filesystem and can be loaded back into application memory in the event that the database is unreachable
- Changes to runtime configuration must be pushed from the database into application memory
- Loss of the SQL database results in partial degradation, where the real-time data stream is still operational, but no changes can be made to the system until the database comes back online

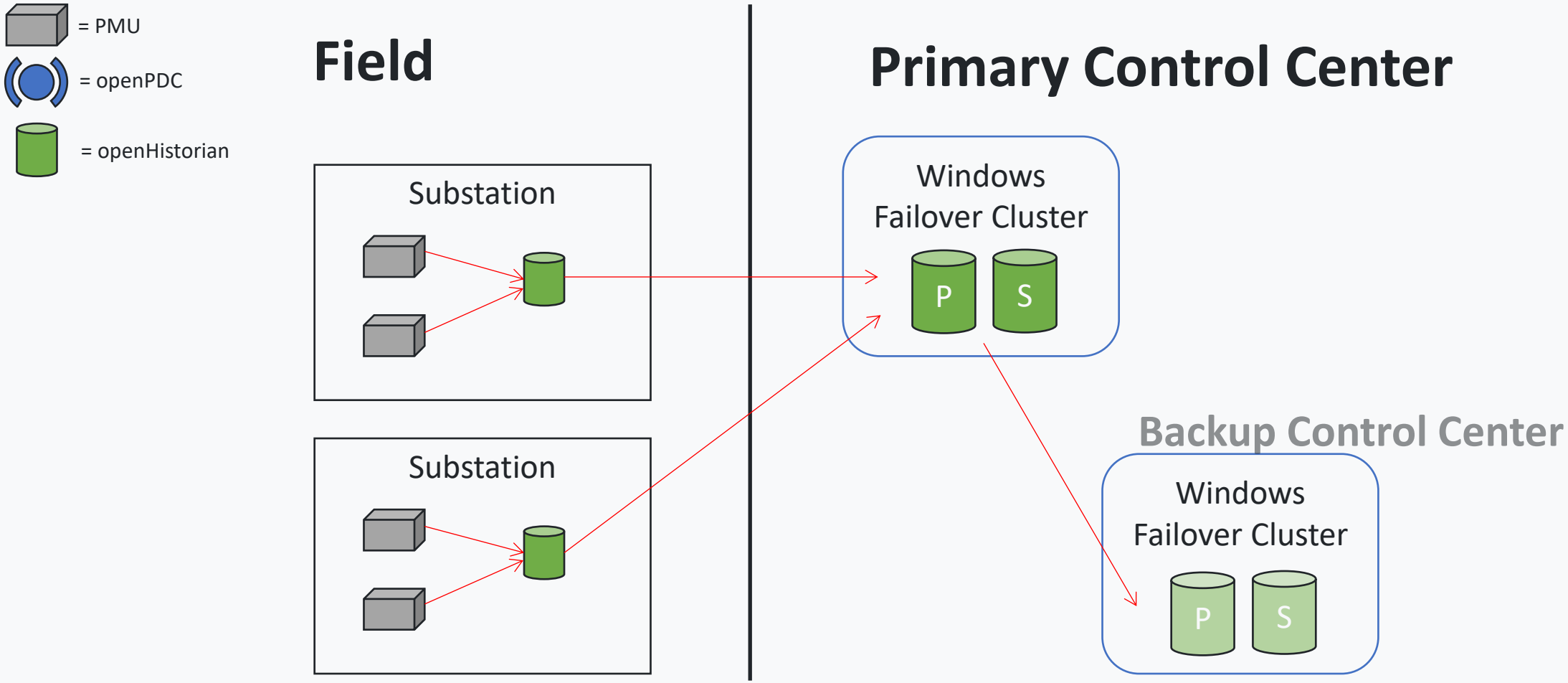
Degradation (archive storage)

- openHistorian writes real-time data to archive files in stages
  - Stage 0 (in-memory)
  - Stage 1 (one file every 10 seconds)
  - Stage 2 (~70 MB)
  - Stage 3 (~2 GB)
- Stages 1 and 2 are both stored in the “working directory”
  - If the working directory is unavailable, data will build up in memory
- Stage 3 is stored in the “archive directory”
  - If the archive directory is unavailable, data will build up in the working directory
- If the working directory is local to each server, then the loss of one server would cause up to 2 GB of data in the working directory to become unavailable for queries

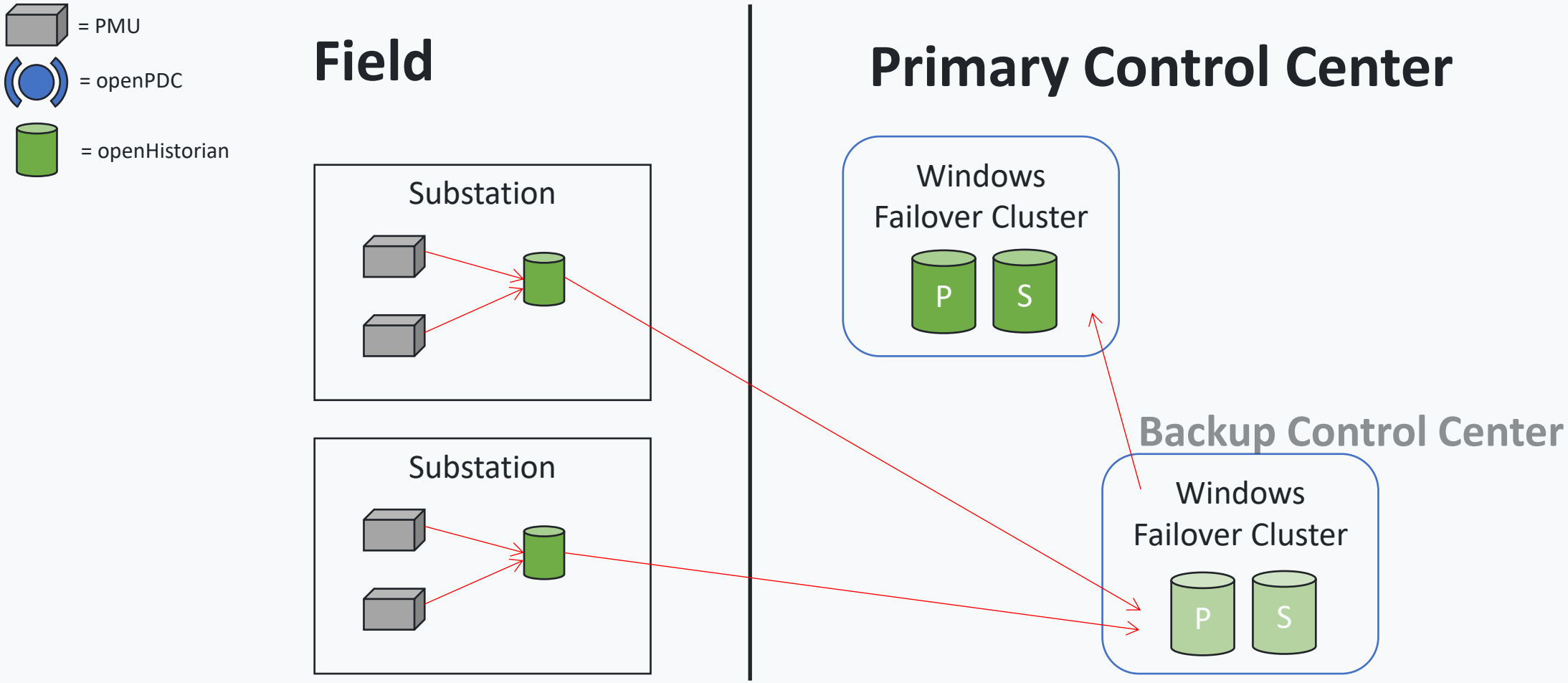
Real World Example 1

- Fully redundant control centers
- Mirrored SQL databases—each control center has a copy of the other's database
- To swap roles, simply switch to the other control center's database configuration
- Substation openHistorian hosts short-term archive for data gap recovery

Real World Example 1



Real World Example 1



Real World Example 2

- Active/active configuration
  - STTP used for sharing measurement IDs (database mirroring)
- openHistorian automatic failover
  - SQL configuration is shared between instances
  - openPDC hosts short-term archive for data gap recovery
- STTP metadata exchange used for mirroring between openPDC instances

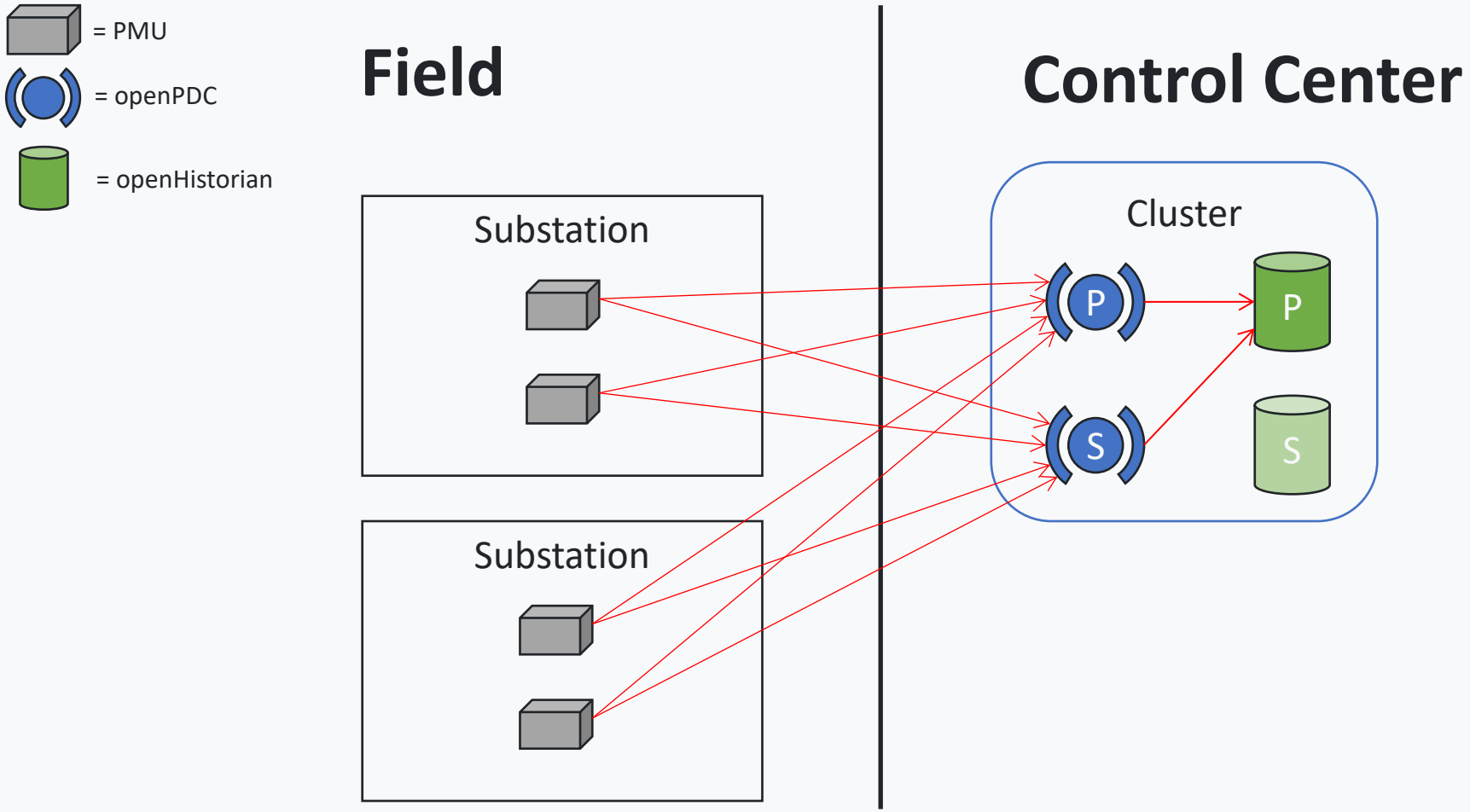
Real World Example 2

- Additional considerations for the SQL database
  - The SQL database instance is hosted on the secondary openHistorian server. If this server becomes unavailable, the other nodes will be dependent on their locally cached configuration to continue operation. This results in a partial degradation of the system where the user is unable to edit configuration until the secondary openHistorian server comes back online.
  - It is important that the SQL database is not hosted on the primary openHistorian server because the secondary openHistorian is normally inactive and therefore may not have up-to-date configuration cached locally.

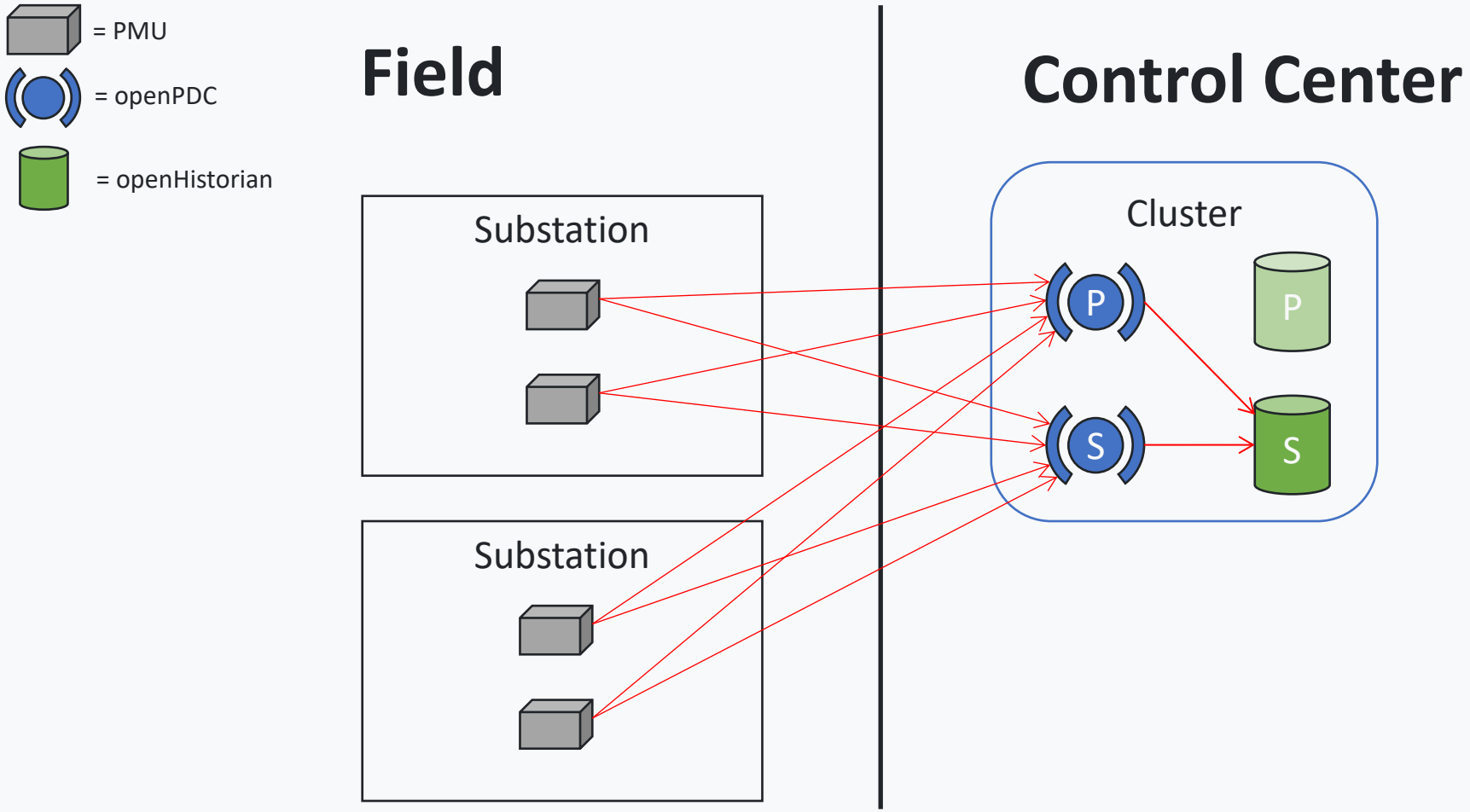
Real World Example 2

- Additional considerations for archive storage
  - Each server hosts two local drives for archive files
  - Working directory is in one of the two local drives
  - Multiple archive directories are configured using UNC paths so openHistorian can write to all four drives from either the primary or the secondary server
  - Loss of one server will result in the data stored on that server's local drives to become inaccessible for queries until the server is brought back online

Real World Example 2



Real World Example 2



Enterprise Edition Adapter Failover

- Intended for active/active configuration
- STTP stream between EE instances to deliver heartbeat signals
- The primary adapter publishes heartbeat measurements periodically to indicate that it is still active
- If the secondary adapter's EE instance does not receive the primary adapter's heartbeat measurement within a configured timeframe, the secondary adapter will be activated