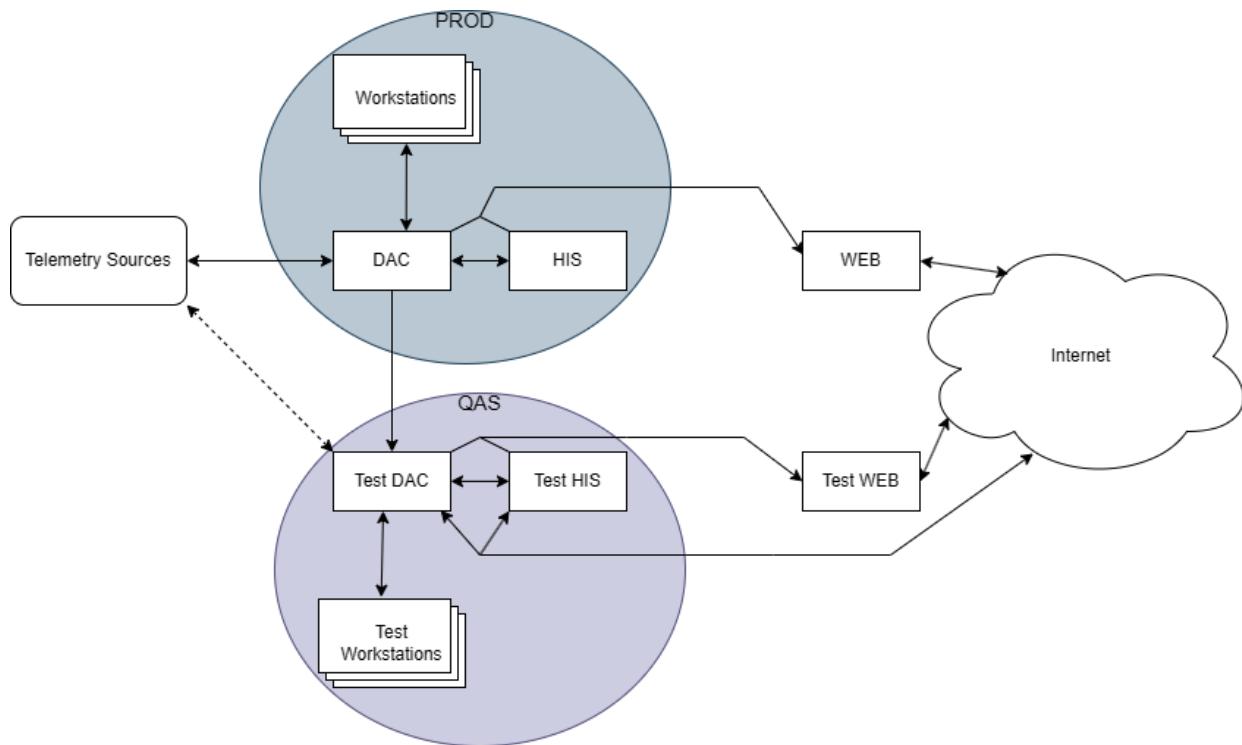


Threat Modeling a Distributed System

System Definition

A distributed monitoring and control system collecting telemetry from remote devices, aggregating data centrally to Data Acquisition & Control (**DAC**) servers, and providing operator access through internal **Workstations** via internal applications, and view-only access to customers via internet browsers hosted from **WEB** servers. Historical data is archived on local **HIS** servers. A replicated Quality Assurance System (**QAS**) is configured in parallel to allow for verification of changes before completing rollouts into Production (**PROD**).

Architecture Diagram



Trust Boundaries & Assets

Telemetry data is tied to Production DAC nodes, and controls may only be facilitated through connected DAC nodes. Customers connecting through WEB servers are unable to send controls, and have data visibility limited to authorized subsets syncing from Production DAC servers. Production DAC, HIS, and Workstation nodes lack internet connectivity to isolate the network from potential external threats.

Direct access to DAC, HIS, and Web-Facing servers is limited to system administrators who have access to service accounts running internal application software. Credentials are set to expire on a bi-annual basis to prevent stagnation. Operator access is limited to designated workstations, which have connections to DAC servers via internal applications. Controls can be sent by operators to Telemetry Sources via internal applications.

QAS servers receive one-way syncing of internal application files and telemetry data from Production DAC servers to QAS DAC servers, which are then propagated to QAS HIS and QAS WEB nodes. Connections cannot be made to Production from QAS servers. Operators are able to complete similar functions on QAS Workstations as Production Workstations, with the ability to connect to testing Telemetry Sources to recreate functionality.

Active Risks

Key areas of risk for proposed system design include:

1. All Telemetry Sources are on the same network.
2. QAS environment servers all possess active internet connections.
3. Admins access nodes and complete activities using shared service accounts.

Due to all Telemetry Sources being on the same network, it is possible for QAS servers to make connections to remote Telemetry Sources actively sending data to PROD DAC servers. This would result in QAS DAC servers taking away connections from PROD DAC's to Telemetry Sources, resulting in the data failing to reach PROD DAC's, as well as possibly allowing for controls to be sent to remote Telemetry Sources from QAS DAC servers. These issues are further exemplified when considering the 2nd finding; that all QAS servers are connected to the internet.

Open internet connections from QAS servers allow for the possibility of malware and bad actors to gain access to these nodes. This is a known risk with designated WEB servers, but is mitigated by:

1. WEB nodes possess a limited subset of data from PROD DAC and HIS servers.
2. WEB servers are unable to edit data or send controls to Production servers/Telemetry Sources.

With the present network configuration of the Telemetry Sources and the QAS servers, it is possible for bad actors to gain access to and send controls out to the Telemetry Sources. This could potentially result in the damage or loss of assets, and negative impacts to customers.

Tracing of bad configuration, or even the presence of bad actors on the servers, is further hindered by the shared use of services accounts for all activities. Without further methods of tracking which users are on the accounts and at specific times, bad configurations or changes being made become more difficult to track and address as needed. This methodology further raises risk with the possibility of bad actors being able to access the QAS servers, and potentially gain access and use these same credentials. Time required to identify and resolve these critical incidents could be further impacted, and the risk for damages is potentially increased exponentially.

Mitigation Strategy

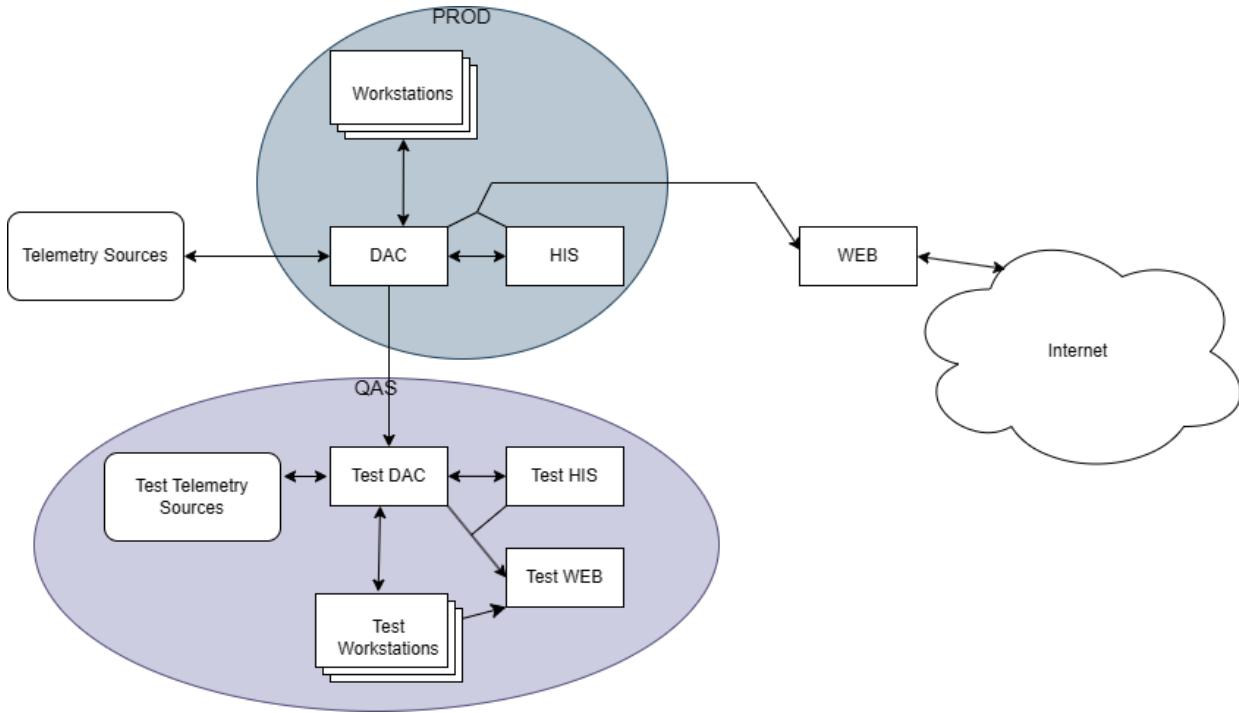
Mitigation 1: Isolating QAS Network

This mitigation is comprised of 2 primary changes:

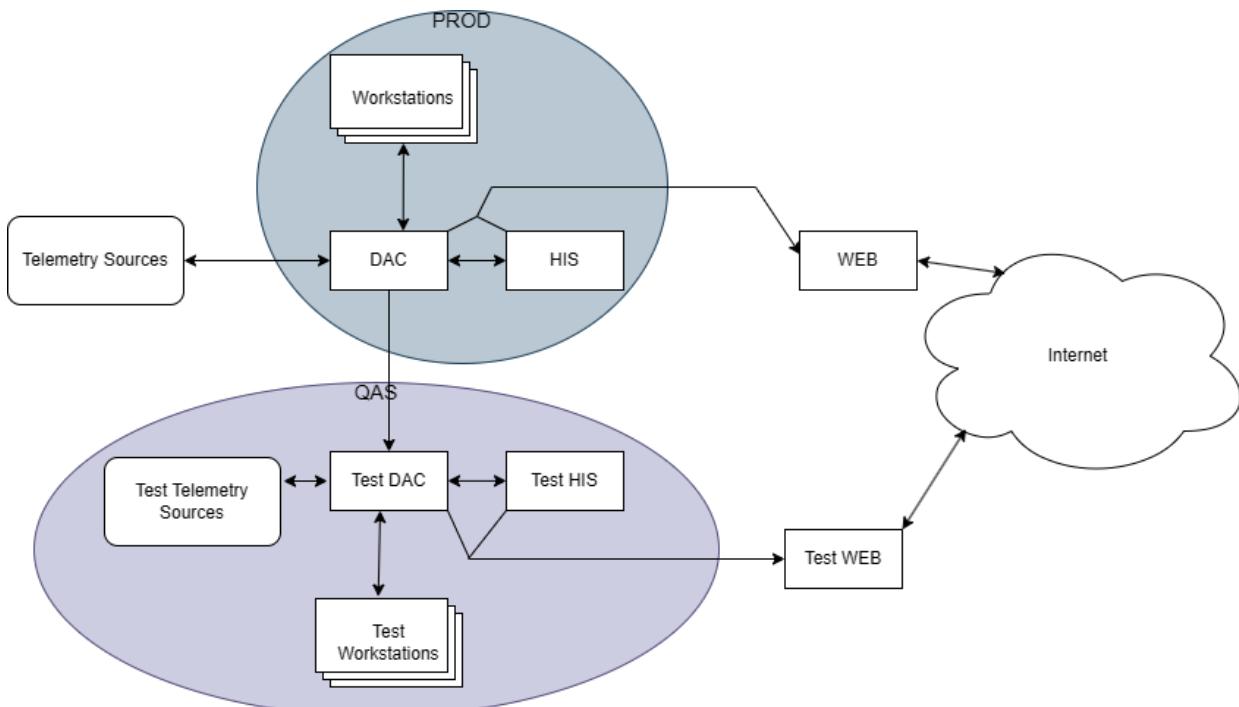
1. Closing the connection from the QAS DAC servers to the Production Telemetry Source network.
2. Removing internet internet access from QAS servers.

The first recommended change prevents any further attempts for connection to be made between the QAS DAC servers and the Production Telemetry Sources, resulting in test Telemetry Sources being configured on the local QAS network. This removes the ability to fully recreate the connections between QAS DAC servers and Telemetry Sources as they would be in the field, but still allows the test environment to verify configuration settings before being deployed into Production.

Removing the ability for any QAS server to connect to the internet prevents the possibility of bad actors to gain access to the system. To retain functionality and testing of the WEB servers, the QAS Workstations are able to connect to the WEB interface using browsers. The consequence is that external users would no longer be able to connect to the QAS WEB servers for testing purposes. The proposed System Architecture design would change to the following:



In the case of a requirement for internet-based users to still be able to access the QAS WEB server for testing, it would be possible to retain a similar configuration to PROD. In this instance the QAS Workstations would lose their connection to the QAS WEB node, but the QAS WEB server would still be able to be seen from the internet. The alternative, albeit less secure, design would be as follows:



Mitigation 2: Separating Accounts

The second proposed mitigation would be to create new local user accounts for each of the individual system administrators on each of the servers. Doing so allows for greater tracking and feasibility of audits on the servers as changes are applied and issues are troubleshooted. As each of these are local accounts, additional oversight would be required to maintain the credentials of the accounts as their individual passwords expire over time. A potential mitigation for this side-effect could involve the implementation of an Active Directory network to maintain the accounts and credentials, but that would be tracked in separate conversations.

An additional consideration in the case of Windows servers would be to configure the service accounts so to prevent the possibility of logging into the systems with them. This would result in the requirement of administrators logging into the servers with their own credentials, improving tracking and monitoring of system access and changes.

Conclusion

In production environments, system risk often arises from boundary erosion and configuration drift rather than single vulnerabilities. Growth of distributed systems also inevitably leads to the expansion of their attack surfaces as new areas and nodes are implemented. Operational convenience, testing requirements, and administrative shortcuts can lead to unintentional erosion of isolation boundaries originally designed to protect critical assets.

This case study highlights exposure pathways created by insufficient segmentation between Production and QAS environments, internet accessible QAS nodes, and the use of shared administrative service accounts. While each configuration decision may have been defensible for operational efficiency, the combined effect increased system risk of lateral movement, privilege misuse, and unauthorized control of telemetry assets. In an architecture capable of sending controls, such exposure extends beyond data confidentiality into potential operational disruption and operational asset impact.

The recommended mitigations of network isolation, reduction of unnecessary internet connectivity, and segregation of administrative accounts re-establishes trust boundaries and improves containment capabilities, traceability, and incident response effectiveness. These controls align with core security principles including least privilege, segmentation, and accountability, and reinforces that security posture is primarily determined by architectural discipline in distributed systems.