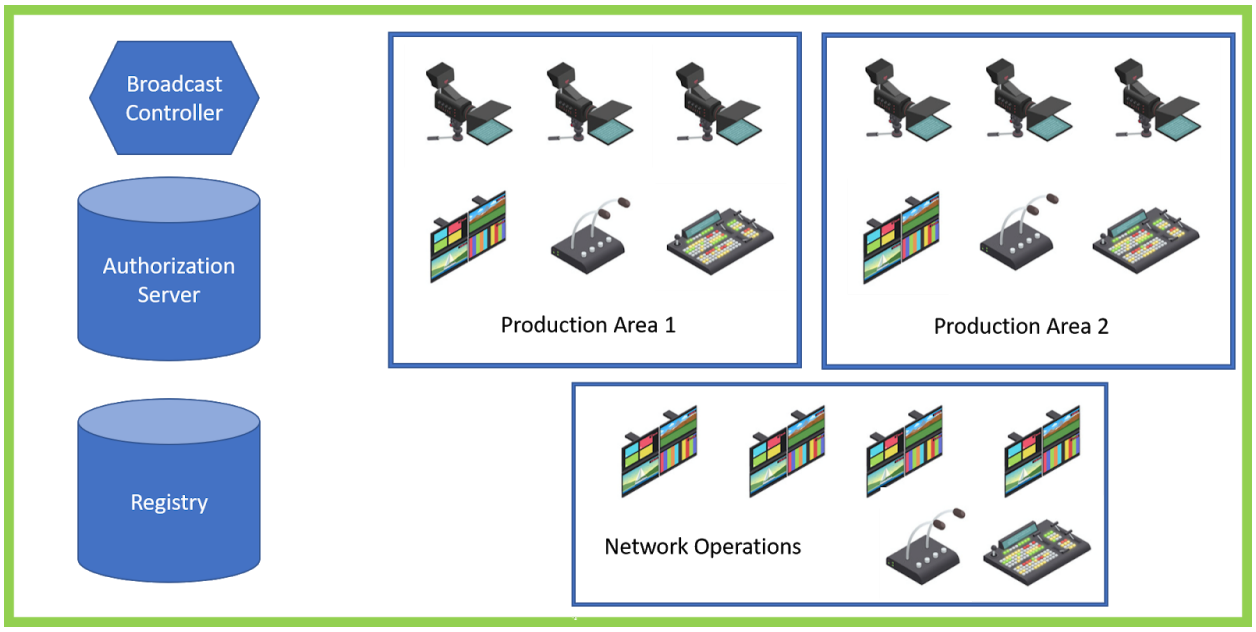
BCP-003-02 Architectures

The location(s) of IS-04 Registries and IS-10 Authorization Servers in a deployment can place additional requirements upon clients such as broadcast controllers, and may adversely impact the capabilities of the system. The following examples aim to explain the problems, and why use of the JWT ‘aud’ claim may provide the best solution in most cases.

Key Issues

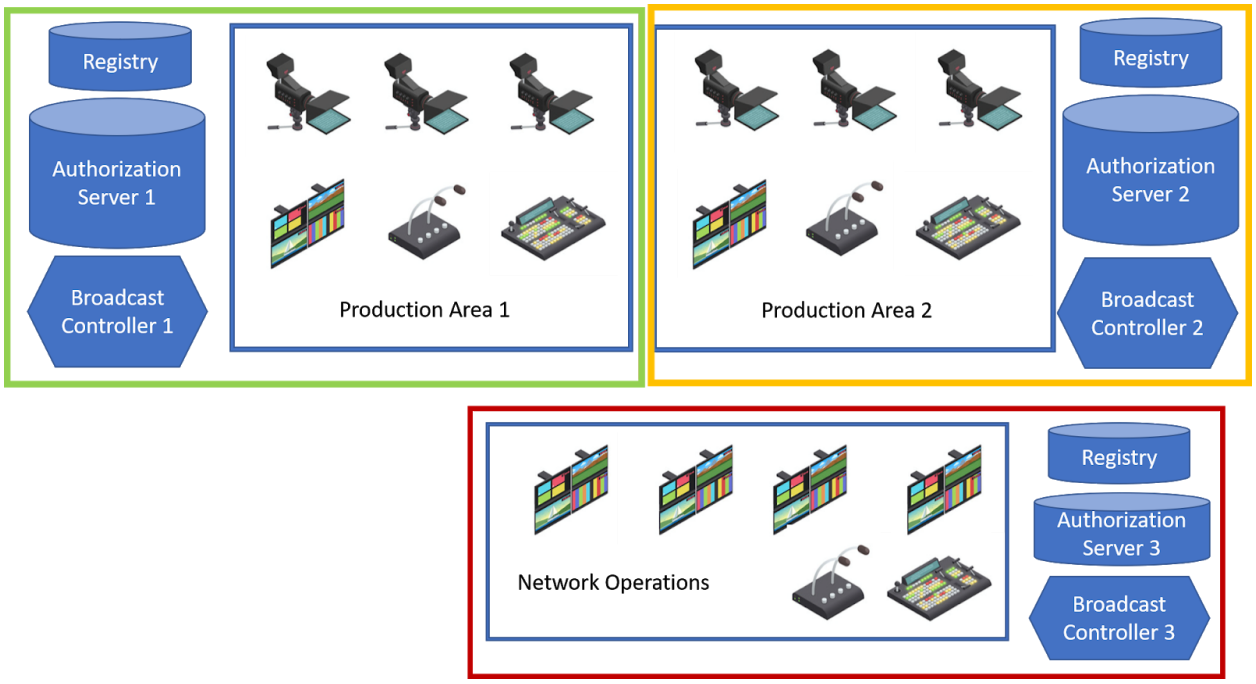
- Sharing access to Nodes, or parts of Nodes in normal operation
- Differences in expected behaviour between permanent physical deployments and temporary logical additions
- Enabling API access for a diverse range of controllers simultaneously

Flat Model



In this scenario, the broadcast controller is the gateway to all operations in the system. It holds a token which can do anything in the system, but imposes different access rights upon users. Multiple broadcast controllers may exist, but each of them will have the rights to do anything. If the broadcast controllers maintain their own maps of what users can and cannot do they will likely become out of sync.

“Physically” Segmented Model



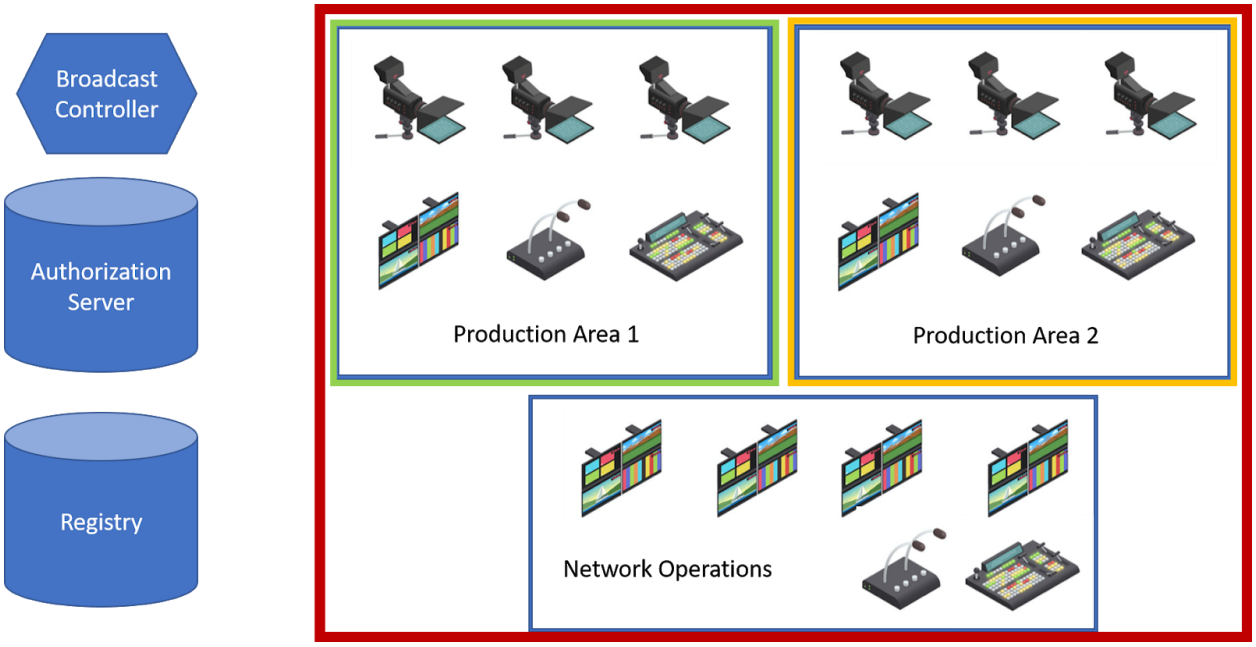
In this scenario, the logical production areas have been separated by hard boundaries in the system design. This may be fine if the areas are permanently separated, but in reality it is likely that at the very least the ‘network operations’ area would need to be able to interchange streams with the production areas.

In order to share access to any Node in this environment, a broadcast controller must be manually configured to know the location of an external registry and authorization server (or browse multiple DNS subdomains). This is a rather specialist requirement which is unlikely to be available in all controllers. What is more, by segmenting the registries it is impossible to look at one of them and develop a full map of the system. Again, this can only be achieved by specialised controllers.

A further complication in this model exists where Nodes do not neatly fit into single logical areas. In many cases Nodes have large processing capabilities which may need to be subdivided. In this case any such Nodes will impose further challenges for controllers which may need to track multiple tokens, registries and authorization servers when managing single productions.

A shared registry could overcome some of these issues, but that registry would then need to discover and trust tokens from multiple independent authorization servers in order to handle Nodes’ registrations and broadcast controllers’ queries.

“Logically” Segmented Model



By pairing a flat architecture with restricted public ‘aud’ or additional private ‘x-nmos-api’ claims which may be issued to given clients and/or users, we can achieve the desired segmentation at a higher layer. This balances the need for minimising potential attack surfaces with the flexibility provided by a single registry and authorization server. Care should of course be taken to avoid granting highly wildcarded tokens or registering such clients unless absolutely necessary, and if this is necessary these entities must be more rigorously tested and highly protected via other design considerations.

If desired, a suitably equipped registry could still offer filtered Query API instances into each production area for consumption by specific sets of clients and/or users (with a matching ‘aud’ in their tokens), but this should not be necessary in all cases.

Inter-Facility Operation

Consideration should also be given to cases where two facilities which typically operate independently need to share some resources. This differs from the logically segmented model as the facilities cannot share a registry or trust domains afforded by their authorization servers. The solution to this may depend upon exactly what is being shared.

1) If the intention is to temporarily expand a facility and provide full access across the boundaries, the simplest solution may be to consolidate the equipment into a single registry with a single authorization server. This would require re-provisioning of

certificates and tokens, along with DNS re-configuration at the very least, but keeps the problem simple. This would of course also depend upon the IP addressing schemes and similar matching up across the boundary, but this is beyond the scope of this work.

2) Where a limited set of resources are to be shared, or potentially just access to a set of streams, the simplest solution may be to use a gateway Node. This would register into both systems and proxy resources into each of them. It could expose a proxy version of any NMOS API and pass on the requests as necessary, dealing with any IP address and token translation. This Node (or Nodes) may need to exist across the boundary of a demilitarised zone (DMZ).

3) A final option would be to add capabilities to controllers to manage tokens from multiple trust domains (authorization servers), and handle data from multiple registries. This carries with it the complexity of the physically segmented model, but may be more manageable in the specific gateway/WAN cases where this is necessary.