Broadcast Control System (BCS)

NRK 2023-1264

SSA-T Appendix 1 – Customer requirements specification

[Contractor name]

[Contractor logo]

Norwegian Broadcasting Corporation Ltd. (NRK)
Bjørnstjerne Bjørnsons plass 1
N-0340 OSLO, Norway
Switchboard: +47 23 04 70 00
Website: www.nrk.no
Norwegian Business Registration NO976 390 512

Table of Contents

1 INTRODUCTION 4
2 BACKGROUND AND PURPOSE 4
3 NON-DISCRIMINATORY SOLUTION 5
4 TECHNICAL BCS SCOPE (CLAUSE 1.1) 5
4.1 BCS functional interfaces 6
4.2 Scope boundary between BCS and PMCS 7
4.3 Staged scope 7
4.3.1 v0.5 – Initial scope 7
4.3.2 v1.0 - Live production and post production workflows needed for new Trondheim 9
4.3.3 v2.0 - Preparations for the move to the new Oslo headquarter in Ensjø 11
5 TECHNICAL REQUIREMENTS 12
5.1 Security 12
5.2 Standards 13
5.3 Configuration 13
5.4 Federation 14
5.5 Resilience 16
5.6 Networks and orchestration 16
5.6.1 Feature areas 17
5.6.2 Target state 17
5.6.3 Requirements – Level 1 18
5.6.4 Requirements – Level 2 18
5.6.5 Requirements – Level 3 18
5.6.6 Requirements – Vendor-specific requirements 19
5.7 Logic engine 19
5.7.1 Control panel logic 19
5.7.2 Background processes (always on) 19
5.7.3 Customer defined logic 20
5.7.4 Requirements 20
5.8 Packaging 20
5.8.1 Federation 20
5.8.2 Control Planes 21
5.8.3 Source Package 21
5.8.4 Destination Package 22
5.8.5 Audio Package 22
5.8.6 Virtual Package 23
5.8.7 Package Lifecycle 23
5.8.8 Requirements 23
5.9 Device control 23
5.9.1 Requirements 23
5.10 Intercom 24
5.10.1 Ports 24
5.10.2 Routing 25
5.10.3 Conferences 25
5.10.4 IFBs 25
5.10.5 Groups 26
5.10.6 Functions 26
5.10.7 Intercom Panels 26
5.10.8 GPIO 27
5.10.9 User Interactions 27
5.10.10 Mixing 27
5.11 User Interfaces 27
5.12 Resource Scheduling 28
5.12.1 Deliverables 28
5.12.2 Time based connection management and event execution 29

5.12.3 Real time connection management and event execution ... 29
5.12.4 Scheduled connection management and event execution ... 29
5.12.5 Connection conflict detection ... 29
5.12.6 Resource scheduling user interface ... 29
5.12.7 Graphical display example (based on current workflow) ... 30
5.12.8 Requirements ... 31
5.13 PLATFORM AND OPERATIONAL ENVIRONMENT ... 31
5.14 LEGACY TRANSITION SUPPORT ... 32
6 INTERACTION WITH EQUIPMENT AND OTHER SOFTWARE (CLAUSE 2.3.2) ... 33
7 IMPLEMENTATION METHOD (CLAUSE 2.3.3) ... 33
8 DOCUMENTATION (CLAUSE 2.3.6) ... 33
9 TRAINING (CAUSE 2.3.7) ... 34
10 CONVERSION (CLAUSE 2.3.8) ... 34
11 PERFORMANCE LEVEL (CLAUSE 4.2) ... 34
12 GENERAL EXTERNAL LEGAL REQUIREMENTS AND MEASURES (CLAUSE 9.2) ... 34
13 PERSONAL DATA (CLAUSE 9.2) ... 34
14 INFORMATION SECURITY (CLAUSE 9.3) ... 34
15 SECURITY FOR ACCESS TO SOURCE CODE, ETC. (CLAUSE 10.2.2) ... 35
16 THE TOOLS AND METHODOLOGICAL BASIS OF THE CONTRACTOR (CLAUSE 10.6) ... 35

Attachments

Attachment 1: NRK Device ST2110 ST2059 NMOS Conformance

Attachment 2: NRK Supplier Security Requirements

1 INTRODUCTION

This Appendix describes the background, scope of the tender process, and the Customer's objectives, needs and requirements for a new Broadcast Control System (BCS) solution. The needs and requirements must be answered in subsequent appendices as indicated, where an overall description of the BCS solution must be provided as well as a detailed description of how the Contractor's objectives, needs and requirements will be fulfilled. It is important to emphasize that the specified needs and requirements in this Appendix must be understood based on descriptions provided as part of the Tender Provisions document and underlying appendices. By answering all requirements and describing how the requirements are met in the offered BCS solution, the Contractor is expected to present its best proposed solution based on the described needs that fulfils the objectives in a best possible manner.

The Contractor's response shall be submitted in SSA-T Appendix 2.

Note: the remaining descriptions of this Appendix will be clarified during the negotiation phase.

2 BACKGROUND AND PURPOSE

NRK is currently in the process of performing a business transformation from SDI and AES3 broadcast technologies to native AV over IP services (AVoIP). The AVoIP model will form the standard model for real-time broadcast activities at all sites.

This business transformation is being co-ordinated by a company-wide programme, referred to as MPP – Modernisation of the Production Platform (MPP). In addition to the transformation to AVoIP, the MPP programme aims to centralise production processing resources in external datacentres, and to lay the groundwork for integrating public cloud services into broadcast workflows.

The transformation was initiated by the decision to establish new headquarters in Oslo at Ensjø, due to outdated facilities at the current Oslo location at Marienlyst. In addition, several large projects have been added to the programme, including establishing a new WAN, and moving NRK's second largest facility in Trondheim to a new premises, making this the largest undertaking in NRK's history.

The objective of this RFP and the subsequent dialogue is to provide NRK with a Broadcast Control System (BCS) which is capable of meeting NRK's long-term strategic goals.

The BCS is a central component in a broadcast production platform. It is important to NRK to find both the right product and a long-term partner, with a product roadmap that aligns to the whole scope outlined as part of this Appendix 1. The BCS specification communicates the following NRK requirements to inform an RFP process:

- A vendor partner to supply, evolve and support a BCS product
- Specific foundational functionality for the first release of the NRK Modernised Production Platform (MPP v0.5)
- A BCS product with a roadmap that meets long term strategic objectives

The BCS requirements specification (this Appendix) should be read in the context of associated tender documentation and should give potential BCS suppliers a clear understanding of the product and partnership requirements to guide their RFP responses.

Please refer to Tender Provisions document for more information about tender background and purpose.

3 NON-DISCRIMINATORY SOLUTION

NRK is committed to "prevent exclusion and dropout from work". Complex and poorly designed computer applications may cause employees to be excluded from work. NRK is required by law, the Norwegian Public Procurement act, to consider life cycle costs, universal design, and environmental consequences of the procurement.

Universal design means "design or adaptation of the main solution in physical conditions", including information and communication technology (ICT), so that the general function of the enterprise can be used by "as many as possible" according to the Discrimination and Accessibility Act.

The Contractor should elaborate on how basic technical and perceptual universal design of relevant parts of the BCS solution is incorporated and enabled.

4 TECHNICAL BCS SCOPE (CLAUSE 1.1)

The main function of the Broadcast Control System (BCS) is controlling live media streams in the production platform. These streams will mainly be based on SMPTE ST-2110 but will also employ other standards such as NDI and SRT. Migrating to the new BCS is likely to require control of some legacy systems that include standards such as SDI and AES3.

The BCS solution has to be seen within the context of other systems. Many integrations are required, however responsibility for individual systems is likely to be as follows.

In scope for BCS solution:

- Routing control layer (end point discovery and connection management)
- Logical routing (packaging/bundling)
- Device parameter control
- Intercom integration and control
- Training
- Ongoing support

Out of scope for BCS solution:

- Resource management system including external resource bookings
- Resource provisioning, allocation of technical resources and building software defined systems
- The Software Defined Network (SDN) controller
- Monitoring and alarms
- Network, compute, operating systems on which the BCS runs
- The deployment system for infrastructure changes

Whilst the above are out of scope for BCS solution, the BCS delivery will require agreed interfaces to these systems.

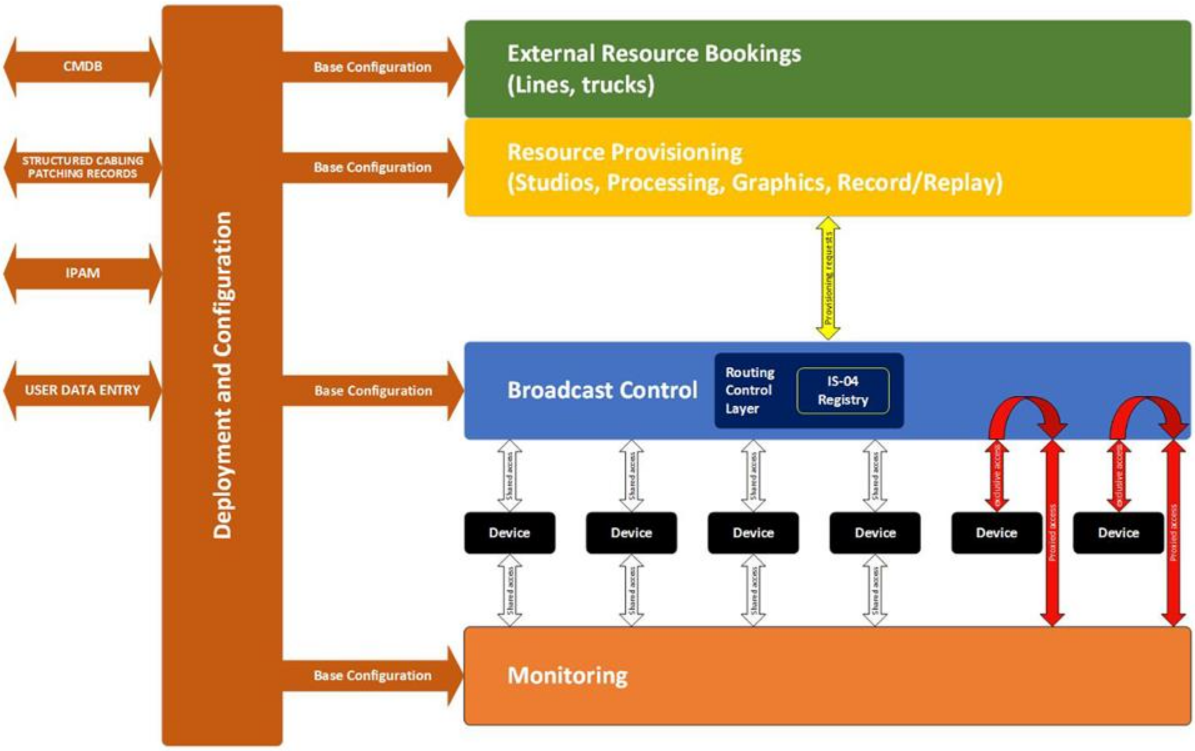


Figure 1 - BCS solution

The BCS is likely to control the following logical system blocks. These may well be added in an evolutionary process over some years across multiple sites and BCS instances.

- Live Production Studios (Vision and Audio)
- Master Control
- Media Ingest
- Edit Suites
- Voice Over Booths
- Post-production “finishing”
- “Quality Check” & Compliance
- Television Playout (but not scheduling/automation)
- Radio transmission routing
- Connection management for distribution (radio, television, and internet)

4.1 BCS functional interfaces

The BCS is likely to control the following equipment categories:

- AVoIP endpoints
- Software Defined Network “controller”
- Traditional crosspoint routers
- Multiviewers
- Vision mixers (BCS tile hosted within mixer UI)
- Audio mixers (BCS tile hosted within mixer UI)
- Intercom system
- Contribution devices (codecs and similar)
- WAN Media Gateways
- Cameras
- Video and audio record/playback servers
- Graphics devices
- System “glue” and processing devices

- Building management systems (house lights, blinds etc.)
- Digital signage and transmission lights

The scope and complexity of each of these interfaces will be determined during the detailed design phase.

4.2 Scope boundary between BCS and PMCS

The BCS is not intended to perform the role of a Platform Monitoring & Control System (PMCS) for all production systems.

For the production platform, a dedicated and thorough PMCS is required based on industry standard tools. In general, the PMCS should make any alarms relevant to the BCS via data made available to (and exchanged with) the BCS.

It is possible that, for a minority of broadcast systems, limitations of the hardware mean that only the BCS can communicate with the equipment. For that equipment the BCS is required to gather alarms as well as control the device and report alarm status to the PMCS.

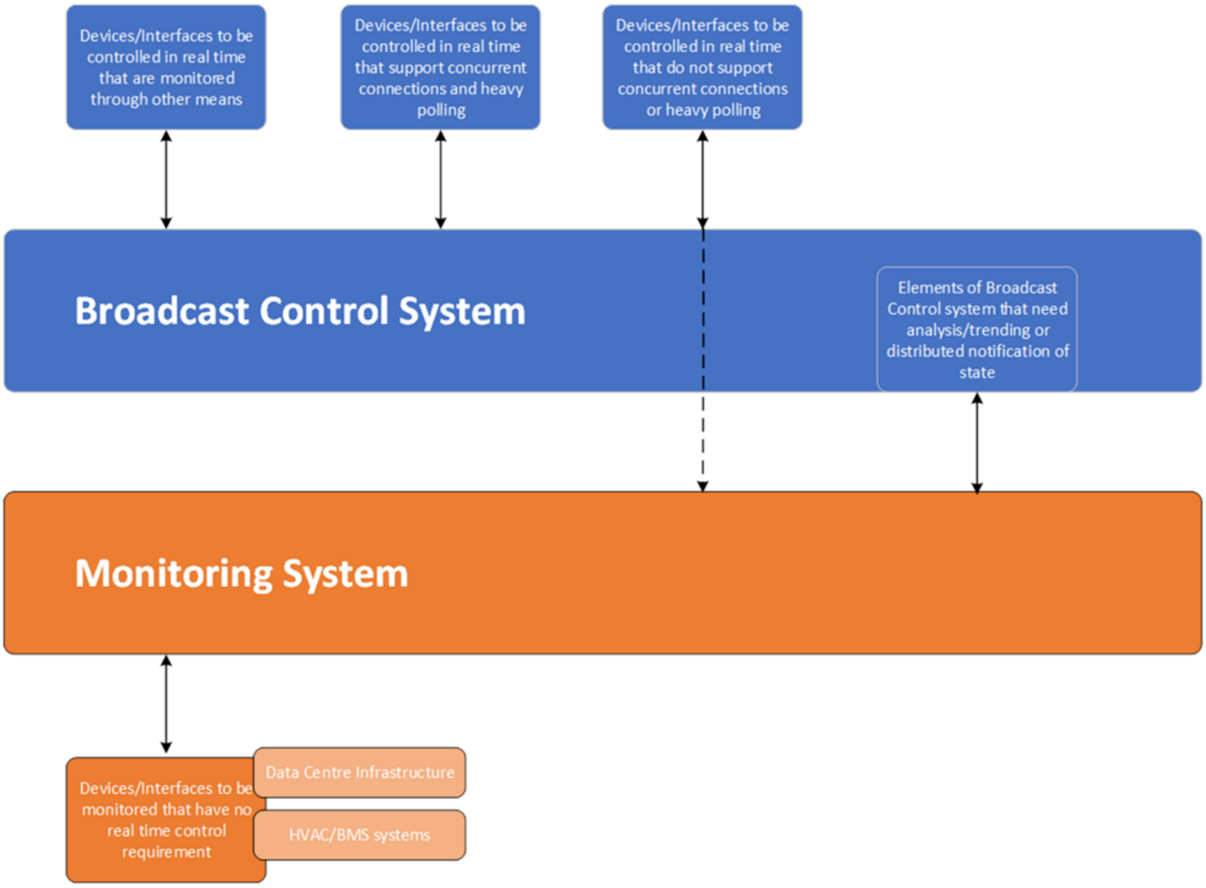

Devices/Interfaces to be controlled in real time that are monitored through other means
Devices/Interfaces to be controlled in real time that support concurrent connections and heavy polling
Devices/Interfaces to be controlled in real time that do not support concurrent connections or heavy polling
Broadcast Control System
Elements of Broadcast Control system that need analysis/trending or distributed notification of state
Monitoring System
Devices/Interfaces to be monitored that have no real time control requirement
Data Centre Infrastructure
HVAC/BMS systems

Figure 2 - BCS and PMCS

4.3 Staged scope

4.3.1 v0.5 – Initial scope

The programme stage leading to release v0.5 is in progress and concludes in Q4 2024.

This stage establishes foundation functionality for use at a minimum viable scale for live services operated from a new MOC workspace:

- Integration with the Network Controller (Multicast SDN)
- Federation (package sharing and flow exchange between BCS instances), see sub chapter for more details
- Logical Packaging, see sub chapter for details
- Intercom Integration
- Legacy (VSM) Integration

Product integrations introduced for this phase include (product choices not decided):

- Network Controller
- Intercom
- Media Gateway Connection Management
- Codecs (Video and Audio)
- Assignable Signal Processing (Video and Audio)
- Multiviewer
- Under Monitor Displays
- Audio Monitoring Unit
- Signal Analyser
- Test Signal Generator
- GPI Interface

4.3.1.1 Scope for v0.5

BCS scope within this stage is highlighted below. Items in grey exist but are out of scope for the BCS project in this phase.

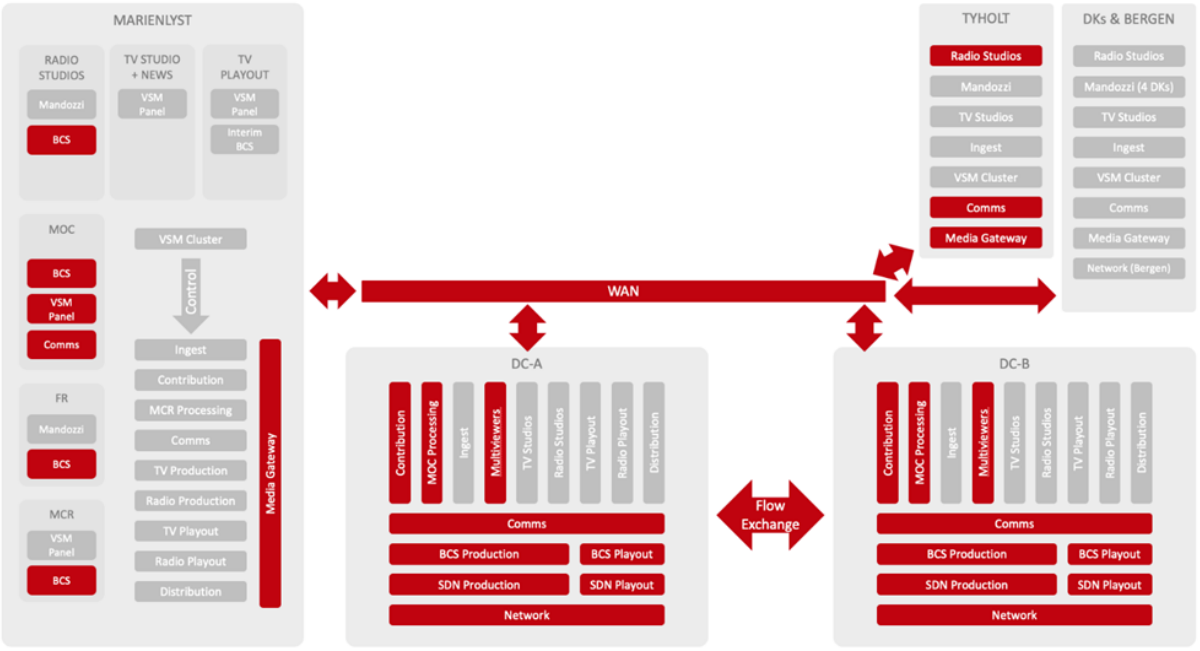


Figure 3 - BCS scope for v0.5

4.3.1.2 Scale for v0.5

Scaling tables are provided as an indicative, pre-design stage, summary for the purpose of price comparison. All BCS system instances, including live and non-live are included. UI rows give an indication of user count across different types of user interface. Talkback port is used to indicate a panel or '4-wire' audio connection. Multiviewer head represents one mosaic source generated in a datacentre. Media Gateway (Flow) indicates the quantity of adapted or compressed flows originating through media gateway devices in each location.

Scope area	Scope item	Datacentres	Marienlyst	Tyholt
BCS Instance	Lab/Test	2	1	1
BCS Instance	Preproduction / QA	2		
BCS Instance	Production	2		
BCS Instance	Playout Preproduction / QA	2		
BCS Instance	Playout Production	2		
UI	MOC Client	2	2	2
UI	MOC Hardware	2	2	2
UI	Legacy MCR (TV and Radio) Client		2	2
UI	Radio Tile		5	5
UI	TV Playout Client			
UI	TV Studio Client			
UI	MFR Client			
UI	Ingest Client			
Device	Talkback Port	24		
Device	Multiviewer Head	12		
Flow	Media Gateway services across WAN (1 x ST 2110-22, 4 x ST2110-30, 1 x ST2110-40 per service)	40	30	10
Flow	WAN Audio Flows (Radio)	40	20	20
Device	Video Codecs Devices	1+1		
Device	Audio Codec Devices	2+2		
Device	SDI <> ST 2110 Adaptation	4+4	8	2
Device	Audio IO & Processing	2+2	4	1
Device	Video Processing	1+1		
Device	Misc (e.g., AMU, IP WFR)	4+4	6	6

4.3.1.3 Functional goals

The v0.5 release is expected to provide the following operational functions:

- Accept a video contribution into the platform, normalise it through processing to house standards (video, audio, metadata), and route it for ingest or live broadcast
- Host a radio channel on the platform and perform source switching and news insertion
- At least one operational Media Operations Centre (MOC) position

4.3.2 v1.0 - Live production and post production workflows needed for new Trondheim

Following v0.5 release in Q4 2024, this stage runs for one year to support the new Trondheim facility launch (v1.0) planned for Q4 2025 in phased increments. v1.0 widens the scope to include all live production and post production workflows needed for new Trondheim. The new NRK Trondheim facility will launch using this functionality, which will evolve to inform Ensjø. All remaining product integrations are introduced for this stage, including:

- Media Gateway parameter control
- Vision Mixer

- Audio Mixer
- Workflow Provisioning
- Resource Management

4.3.2.1 Scope for v1.0

BCS project scope within this stage is highlighted below. Items in grey exist but are out of scope for the BCS project in this stage.

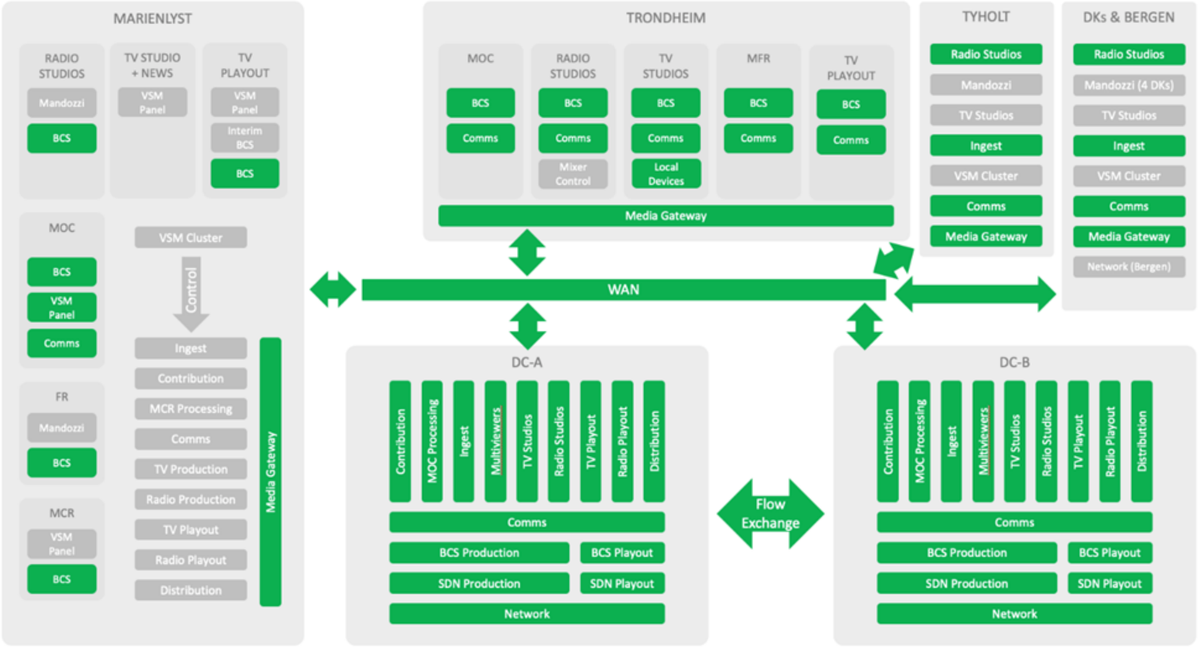


Figure 4 - BCS scope for v1.0

4.3.2.2 Scale for v1.0

Scope area	Scope item	Datacentres	Marienlyst	Tyholt	Trondheim	DKs
BCS Instance	Production	2				
UI	Playout	2				
UI	MOC Client	2	8	2	8	
UI	MOC Hardware	2	8	2	8	
UI	MCR/FR Client		2	2		
UI	Radio Tile		100	20	20	
UI	TV Playout Client		10		2	
UI	TV Studio Client				5	
UI	MFR Client				32	
Device	Ingest Client				5	
Device	Talkback Port	116				
Device	Multiviewer Head (assume 16 x Inputs per head)	28				

Flow	Media Gateway services across WAN
	(1 x ST 2110-22,
	4 x ST2110-30,
	1 x ST2110-40 per service)
		190(V)
		760(A)
		190(D)
			30(V)
			120(A)
			30(D)
				10(V)
				40(A)
				10(D)
					60(V)
					240(A)
					60(D)
						130(V)
						520(A)
						130(D)

Flow	WAN Audio Flows (Radio)	750				
Device	Video Codecs Devices	30+30				
Device	Audio Codec Devices	30+30				
Device	SDI<>ST2110 Adaptation	4+4	8		2	
Device	Audio IO & Processing	9+9	4	1	25	
Device	Video Processing	4+4				
Device	Misc. (e.g. AMU, IP WFR etc.)	5+5	16	6	30	
Device	Live Ingest	20+20				
Device	Playout & Distribution	20+20				
Device	TV Vision Mixing	1+1				
Device	TV Audio Mixing	1+1				

4.3.3 v2.0 - Preparations for the move to the new Oslo headquarter in Ensjø

This stage starts in Q1 2026 and prepares to support the new Ensjø launch (v2.0) and closure of Marienlyst. v2.0 is planned for Q2 2028 and increases platform scale to service Ensjø, using integration and workflows proven through v0.5 and v1.0. Between v1.0 and v2.0 there will be a gradual scaling up of the platform as end-of-life facilities located at Marienlyst and the District Offices are upgraded to work on the MPP.

4.3.3.1 Scope for v2.0

BCS project scope within this stage is highlighted below. Items in grey exist but are out of scope for the BCS project in this stage.

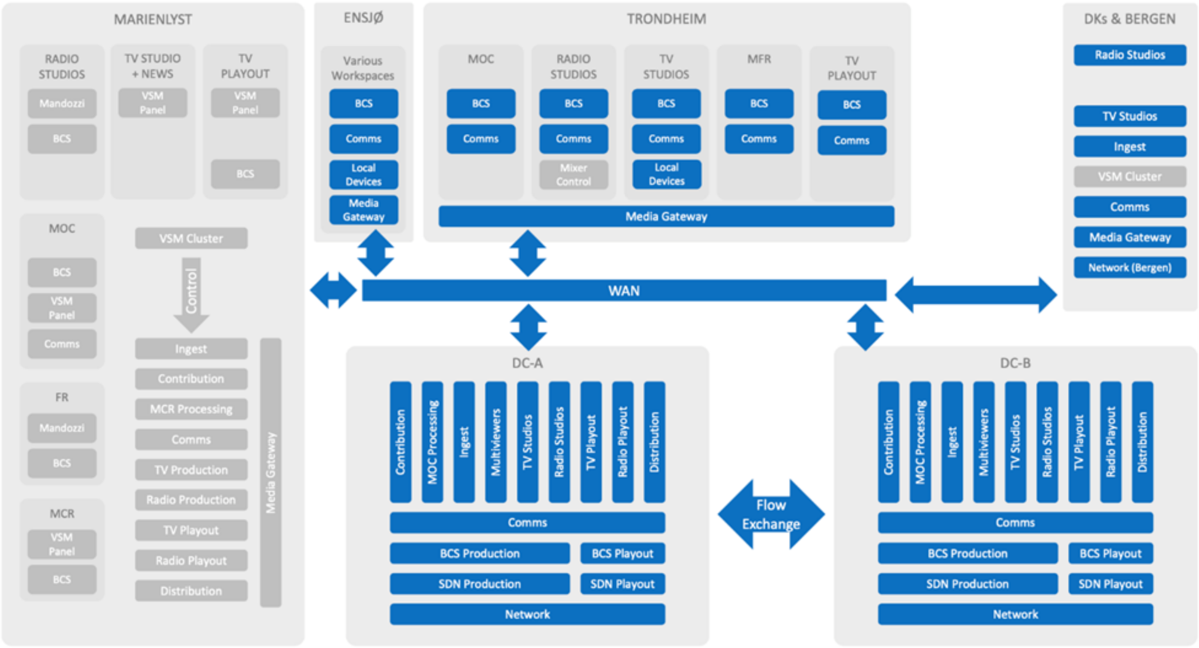


Figure 5 - BCS scope for v2.0

4.3.3.2 Scale for v2.0

Scope area	Scope item	Datacentres	Trondheim	DKs	Ensjø
BCS Instance	Production	2			

BCS Instance	Playout	2			
UI	MOC Client	2	8		8
UI	MOC Hardware	2	8		8
UI	MCR/FR Client				
UI	Radio Tile		20	50	100
UI	TV Playout Client			2	10
UI	TV Studio Client		5	10	50
UI	MFR Client		32	80	150
UI	Ingest Client		5	10	20
Device	Talkback Port	678			
Device	Multiviewer Head	164			

Flow	Media Gateway services across WAN
	(1 x ST 2110-22,
	4 x ST2110-30,
	1 x ST2110-40 per service)
		610(V)
		2440(A)
		610(D)
			60(V)
			240(A)
			60(D)
				130(V)
				520(A)
				130(D)
					420(V)
					1680(A)
					420(D)

Flow	WAN Audio Flows (Radio)	750			
Device	Video Codecs Devices	30+30			
Device	Audio Codec Devices	20+20			
Device	SDI<>ST2110 Adaptation	4+4	2		8
Device	Audio IO & Processing	9+9	25	75	150
Device	Video Processing	4+4			
Device	Misc. (e.g. AMU, IP WFR etc.)	10+10	30		220
Device	Live Ingest	20+20			
Device	Playout & Distribution	20+20			
Device	TV Vision Mixing	8+8			1
Device	TV Audio Mixing	8+8			1
Device	Graphics	4+4			
Device	Replay	8+8			

5 TECHNICAL REQUIREMENTS

This section contains the Customers' requirements for the BCS solution.

5.1 Security

High public trust is essential for NRK, and safeguarding content production and delivery is vital. Therefore, NRK should always follow security best practices. Going forward, NRK expects vendors to adhere to security best practice, such as EBU Cybersecurity Recommendations for Media Vendors' Systems, Software and Systems (EBU R 143). The BCS is a critical production component and must employ adequate protection and security measures. The Contractor should comply with NRK's Supplier Security Requirements as specified in Attachment 2.

#	Requirement	Stage
1	The BCS should support single sign on using oauth2/openid and/or SAML.	v0.5

2	The BCS can use either Active Directory or Azure Active Directory for users and groups. If the BCS has its own internal user and group database, the BCS should support provisioning users and groups using SCIM.	v0.5
3	The BCS should support role-based access control.	v0.5
4	Communication between the users and the BCS system should be encrypted using TLS1.3 or higher.	v0.5
5	Security logs from the BCS should include the following information: • Log in information • Audit logs for changes done by users • Audit logs for changes done by administrators • Audit logs for changes done via API • User and group synchronization	v0.5
6	The BCS should be able to send security logs to a remote SIEM (Security Incident and Event Management) system. Currently NRK uses Azure Sentinel.	v0.5
7	The BCS should support API access using supplier specific access tokens or OAuth2.	v1.0
8	The BCS should support RBAC for API access.	v1.0
9	The provider should provide a SBOM for the BCS.	v1.0
10	The provider has a responsible disclosure policy.	v1.0
11	The provider should have security.txt implemented.	v0.5
12	The provider should have a defined patch strategy with specific time frames for when patches is provided, differentiated by severity levels of the security issue.	v0.5

5.2 Standards

NRK endeavours to be a driving force in the utilization of a standards driven IT-centric production platform. To support this strategy the BCS should prioritise standards over vendor specific solutions.

#	Requirement	Stage
1	The BCS should support NMOS IS-04 for device registration.	v0.5
2	The BCS should be able to act as a NMOS registry.	v0.5
3	The BCS should support NMOS IS-05 for connection management.	v0.5
4	The BCS should support NMOS IS-07.	v0.5
5	The BCS should support NMOS IS-08.	v0.5
6	The BCS should support NMOS IS-10.	v1.0

5.3 Configuration

The BCS should be configurable using industry standards and APIs.

#	Requirement	Stage
1	The BCS should have a complete REST or GraphQL API for configuration of the BCS providing at a minimum: • Adding, removing, and updating devices • Adding, configuring/updating, and removing user interfaces • Configuring multicast orchestration integrations	v0.5
2	The BCS should have support for configuration using Ansible or Terraform with a Ansible module or Terraform provider supported by the vendor.	v0.5

3	The BCS should support administrator-configurable webhooks to interact with other generic APIs (e.g., on status updates).	v0.5
4	The BCS should support sending and receiving events on a service bus using AMQP.	v1.0

5.4 Federation

The BCS high-level deployment model diagram below provides an overview of the MPP AVoIP federation architecture.

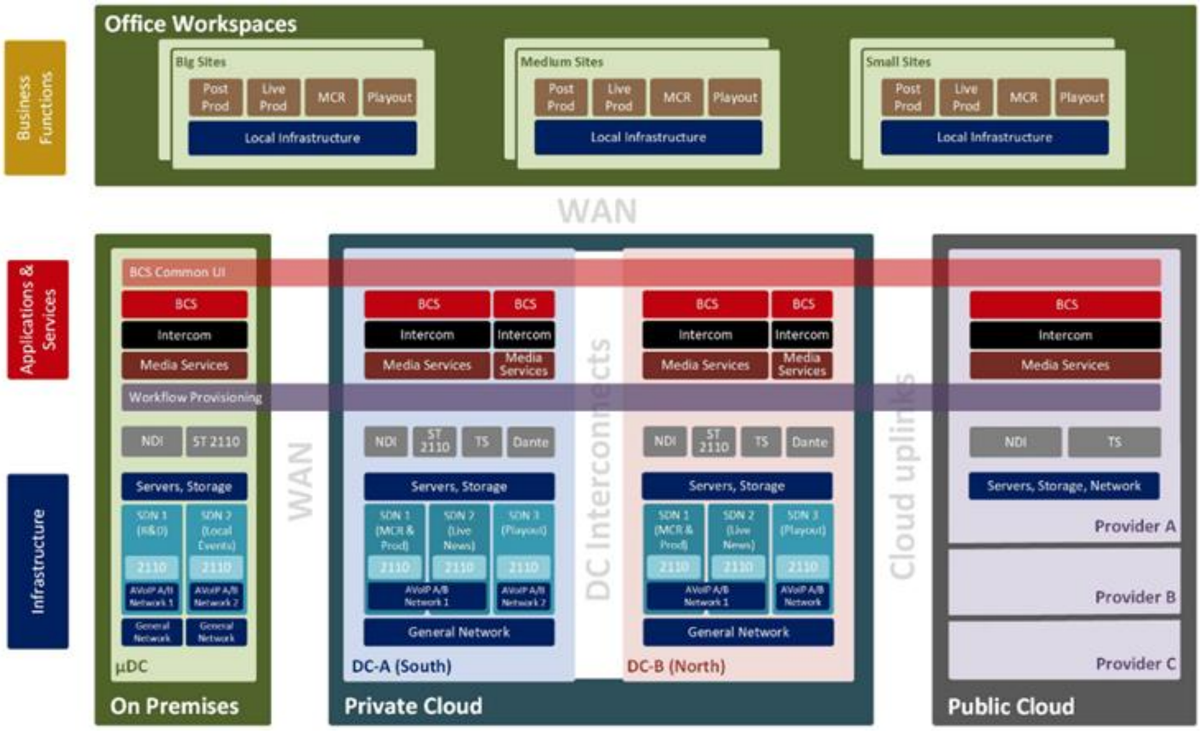


Figure 6 - The BCS high-level deployment model diagram

To preserve security of the operation, reliability of the platform and segment the system into manageable chunks, interconnected BCS instances will exist across NRKs locations and in places, across a campus or site.

Within each DC there will be two BCS instances. One assigned to Production and Master Control, the other to Playout and Distribution. Each BCS instance will communicate with associated Production and Playout SDN instances. Underneath the Production and Playout BCS /SDN instances with be a common, mirrored spine/leaf AVoIP network fabric. BCS instances must communicate with each other and their associated SDN instances to exchange ST 2110 multicast flows between Senders and Receivers. Flow exchange may be between systems within a location or between systems in different locations.

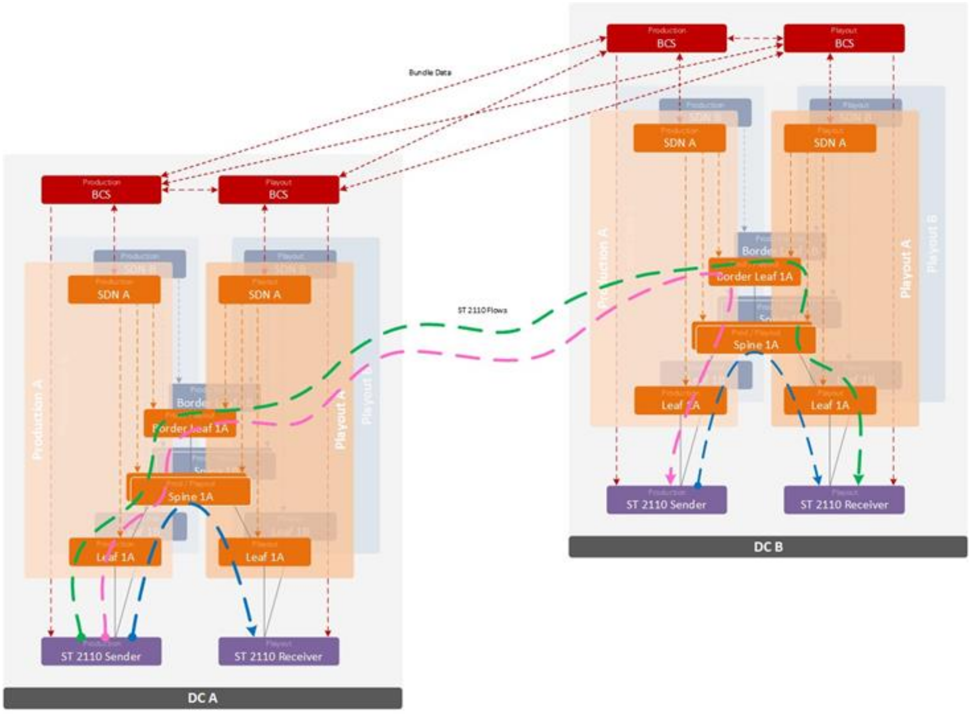


Figure 7 - DC A and DC B

Various flow exchange scenarios must be supported between the DCs. In summary:

1. Prod BCS/SDN DC A <--> Prod BCS/SDN DC B
2. Prod BCS/SDN DC A --> Playout BCS/SDN DC A
3. Prod BCS/SDN DC B --> Playout BCS/SDN DC B
4. Prod BCS/SDN DC A --> Playout BCS/SDN DC B
5. Playout BCS/SDN DC A --> Playout BCS/SDN DC A
6. Playout BCS/SDN DC B --> Playout BCS/SDN DC B
7. Playout BCS/SDN DC A <--> Playout BCS/SDN DC B

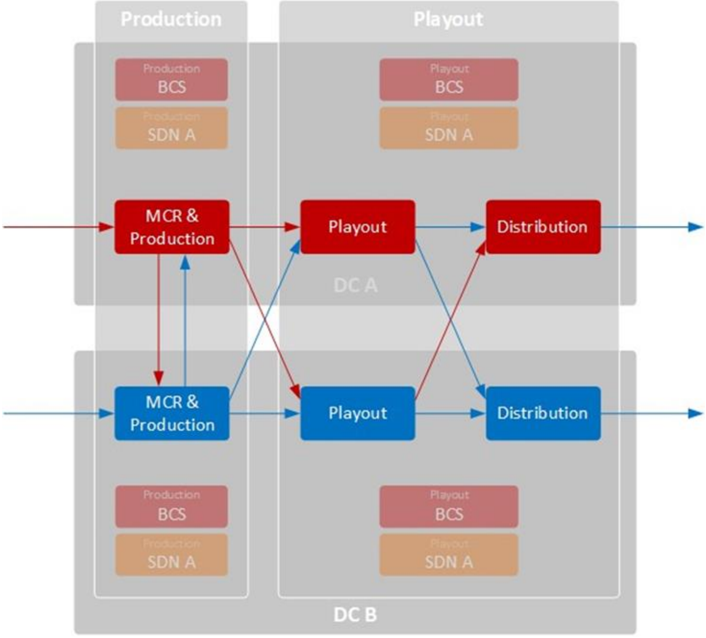


Figure 8 - Flow exchange between DC A and DC B

Added to this, BCS instances within the DCs must control equipment located within offices. For example, ST 2110 Receivers with control rooms (e.g., MCR, Playout, Production Control) which feed monitors.

To streamline the operation, the BCS must provide a mechanism to enable a single federated name space to work across NRKs facilities at our different locations.

The possibility exists that NRK could run their existing BCS system (VSM) alongside the MPP BCS system for a transition period to simplify migration. For example, VSM might be retained for radio production in the near term, whilst the MPP BCS project focuses on delivering core platform features for MCR as well as features for Playout and TV Studios. The two systems would hand-off audio flows via logical tielines and a media edge device.

#	Requirement	Stage
1	The BCS system should support a federated model with more BCS instances as described above.	v0.5
2	Each of these BCS instances should communicate with each other to share resources (e.g., source and destination bundles) with human or system users with appropriate access privileges.	v0.5

5.5 Resilience

NRK is obligated to ensure that information from the government reaches the population during times of emergency or war. Being a critical component in NRK's live production infrastructure, the BCS must be fully resilient and fault tolerant. Each BCS instance must also be able to operate autonomously, so that for instance if one data center is cut off, NRK can still produce and distribute content using the remaining data centers and locations.

#	Requirement	Stage
1	No single point of failure in system core.	v0.5
2	No single instance of a database.	v0.5
3	The BCS should be able to merge changes which will be applied without disrupting existing configurations, streams, and ongoing productions.	v0.5
4	Driver architecture which ensures that large amounts of repetitive data from one, or many sources do not cause the system to fail to control inputs or to log data from other devices.	v0.5
5	Driver architecture which supports seamless fail over between resilient driver instances.	v0.5
6	Support for in-service partial and full system backup.	v0.5
7	Support export of the system backup in order to support off-site backup.	v0.5
8	Support roll-back to selected snapshot (partial backup) or full backup, with restoring the configuration, but keeping the live settings in the system such as routing of transport streams, etc.	v0.5
9	Support backup and restore with no overall impact to system availability.	v0.5
10	Each BCS instance should be able to function autonomously.	v1.0
11	Support setup of SMTPE 2022-7 redundant streams across two networks fabrics.	v0.5
12	Support re-establishing network connections after network failure.	v0.5

5.6 Networks and orchestration

To fulfil the needs of an IP-first BCS, there is a need for the BCS to be the point of co-ordination between Media Endpoint devices, the network fabric and its multicast orchestration system, a Configuration Registry (often referred to as a Source of Truth).

In NRK, the Configuration Registry is provided by Netbox. In addition to handling IP address management, a lot of other resource management throughout NRK is co-ordinated through this tool. There is a high-level goal to ensure that information in this tool is kept as up to date as possible.

Because of this, there is a need for an integration between the BCS and the network fabric control systems, as well as the Network Registry. These integrations may take the form of native integrations but can also be achieved by configuring a generic "web-hooks"-based event system.

In the cases where the BCS expects to poll, or otherwise receive information from external systems in order to display status, it is important that this is done in such a way that does not pose a significant toll on network equipment in terms of CPU usage.

5.6.1 Feature areas

There are three main areas of BCS to network integration, in decreasing order of criticality:

1. Integration with Multicast Orchestrator in a federated environment
2. Network Registry and Dynamic Multicast Allocation
3. NAT handling and "IP Tielines"

5.6.2 Target state

In NRK's envisioned IP-based production platform, the BCS will do the following:

Provisioning multicast flows

- The Multicast Group (G) address of the multicast flow will be selected from a pre-determined multicast range, relevant to the site from which the flow originates, and the class of stream.
  - The multicast range may be allocated to the BCS in the Network Registry.
- The BCS can configure the media endpoint to start the flow towards the desired G address.
- The BCS can provide information about both the Source and the Group address of each flow it aims to establish, in addition to Label metadata, to help identify the origin of the flow.
  - To a target Multicast Orchestrator (e.g., Arista MCS or Cisco IPFM).
  - To a "generic" system for further accountancy (message queue or webhook).
- The BCS can populate an external multicast registry with:
  - The Origin Device Name.
  - Flow information (Source and Group address, and flow label).
- The BCS will receive a callback from the Multicast Orchestrator when the Orchestrator has confirmed a successful route establishment or failure; and make this information available in the BCS.

Subscribing endpoints to multicast flows

- The G address of the multicast flow, as well as further metadata about the device subscribing to the flow will be communicated to the endpoint, which will subscribe to the multicast flow.
- If the endpoint device provides a status callback, this information will be made available in the BCS.

Provisioning devices

- Maintaining Name Relationships
  - Devices that are configured in the BCS will have their name populated with the device FQDN as assigned in the Network Registry; or, failing

this, update the Network Registry with a new IP address assigned to the same device.

- Sub-devices and interfaces that have their own IP addresses will likewise have the same update mechanics.

5.6.3 Requirements – Level 1

The BCS must integrate with a Multicast Orchestrator.

#	Requirement	Stage
1	The integration must make the Multicast Orchestrator aware of the desired flows between senders and receivers.	v0.5
2	The integration must receive callbacks from the Multicast Orchestrator regarding the successful establishment of flows.	v0.5
3	The BCS must be able to integrate with multiple Multicast Orchestrator instances.	v0.5
4	The BCS must be able to exchange flows between Multicast Orchestrator instances on the same physical network fabric.	v0.5
5	The BCS must be able to exchange flows between Multicast Orchestrator instances on different physical network fabrics.	v0.5
6	The BCS must be able to exchange flows between a locally controlled Multicast Orchestrator instance on a local physical network fabric and a remote Multicast Orchestrator instance on a remote network fabric, under the control of another BCS instance.	v0.5
7	The BCS must be able to exchange flows between an endpoint attached to a network under Multicast Orchestrator control and an endpoint attached to a non-orchestrated network.	v0.5
8	The BCS must be able to exchange flows between an endpoint attached to a network under Multicast Orchestrator control and an endpoint attached to a non-orchestrated network, where the one of the endpoints has an affinity to another BCS instance.	v0.5

5.6.4 Requirements – Level 2

The BCS should integrate with a Configuration Registry (Single Source of Truth), to enable dynamic multicast allocation. NRK has currently chosen Netbox as this single source of truth.

#	Requirement	Stage
1	When a sender is to be configured from the BCS, the integration should generate a query, containing parameters about the class of sender, and type of stream to a REST API, which will return the next vacant multicast address.	v1.0
2	The BCS should configure the sender to use the multicast group address provided by the network registry.	v1.0
3	The BCS should perform a callback to the network registry, updating the state of the multicast address allocation with relevant metadata, including but not limited to: the device ID, the device name, the sender interface, the flow group (e.g. if it belongs to a device-originated group), and the flow index.	v1.0

5.6.5 Requirements – Level 3

When bridging disparate external IP networks, for instance from 3rd party contribution or occasional networks, NRK needs to be able to exchange multicast flows. It is not a given that NRK is able to define the addressing scheme of the 3rd party network – and as such, a multicast address translation is required (M-NAT). There are two primary approaches to this:

- M-NAT by means of Media Gateway

- M-NAT by means of generic IT switch, directly configured or via a Multicast Orchestrator

The BCS should be able to dynamically allocate multicast addresses for these use cases and configure the translation device to achieve flow exchange.

#	Requirement	Stage
1	The BCS can configure multicast NAT, using broadcast-specific devices, e.g., “translation gateways”.	v1.0
2	The BCS can configure multicast NAT by using the network vendors orchestrator via APIs.	v2.0

5.6.6 Requirements – Vendor-specific requirements

NRK has not yet chosen the vendor for media network infrastructure, so the BCS should support the two major players for media networks, that is Arista and Cisco.

#	Requirement	Stage
1	The BCS should support establishing paths through with Cisco NBM-based fabrics via Cisco IPFM APIs.	v0.5
2	The BCS should support handling events returned from Cisco IPFM.	v0.5
3	The BCS should support establishing multicast paths through Arista fabrics, via Arista MCS APIs.	v0.5
4	The BCS should support handling events returned from Arista MCS.	v0.5
5	API calls made to multicast orchestration systems should be logged as events.	v0.5
6	Received events / callbacks from multicast orchestration systems can be logged as events.	v0.5

5.7 Logic engine

There are many occasions when it is necessary to apply business logic within the BCS using a built-in logic engine. These can be divided into two broad categories:

5.7.1 Control panel logic

Typical examples:

Destination list
On a routing panel, a control that produces an on-screen report that shows all the destinations to which the currently selected source is routed.

Monitor follow
On a routing panel, a control that watches for local operator's selection of source and routes that directly to a local monitoring destination, or it receives a destination index, and routes the source that is currently routed to that index to a local monitoring destination.

5.7.2 Background processes (always on)

Often, there is a requirement for logic to operate continuously, typical examples are:

Under monitor displays (UMD) and Tallies
A major example of background logic is the handling of UMDs and tallies. An automatic process watches the Package Routing, and any underlying XY routing, along with the tally states of all Vision Mixers and forwards UMD strings and tally states to devices that require them

(Multiviewers, Physical UMDs, GPIs etc.) and continuously re-calculates values according to state and routing changes.

Camera control/Shading Touchdown Logic
As a camera control operator/shader presses down on a Camera RCP, the appropriate CCU output is routed to a local monitor. When the RCP is released, the monitor routing reverts to a preselected 'flyback' source. Multiple RCPs will result in a 'last press wins' and a logic that works out the appropriate releasing any RCP will result in the latest routing command to be executed.

5.7.3 Customer defined logic

The BCS solution should provide a mechanism for the customer to define and write scripts or applications to perform logical operations. This should be capable of processing large data sets in a timely manner, in which case compiled applications may be more appropriate than a scripted approach.

5.7.4 Requirements

#	Requirement	Stage
1	The BCS should support the control panel logic.	v0.5
2	The BCS should support the background processes to automatically update GPIO and UMD.	v0.5
3	The BCS solution should provide a mechanism for the customer to define and write scripts or applications to perform logical operations. This should be capable of processing large data sets in a timely manner, in which case compiled applications may be more appropriate than a scripted approach.	v0.5
4	The BCS should support a language that gives us the ability to develop custom logic scripts.	v1.0
5	The BCS logic engine should be able to make actions based on all the states in the BCS.	v0.5

5.8 Packaging

NRK will employ a “Package” routing concept, where senders and receivers, both real and virtual, on different routing planes are packaged together to form “virtually associated sources” or “virtually associated destinations”. Routing of packages will be a process of making a series of connections between the items contained within a Source package with those in a destination package across one or more routing planes.

Packages may also contain pointers to logical constructs within other systems under control, such as the intercom.

For the purposes of commissioning, testing and emergencies, it must still be possible to ‘make routes’ (i.e., subscribe receivers to senders) on each individual routing plane.

5.8.1 Federation

The precise definition of how federation will work is, to a large extent, informed by the topology of the IP fabric and endpoint arrangements. However, some broad principles are that

- all Source Packages are visible and available to all Destination Packages
- availability (and thus visibility) of specific levels in a Package route are dictated by two factors:
    - IP Network topology/tieline availability (e.g., if there is no direct network path from sender to receiver)

  - Logical restrictions (e.g., preventing a TV Studio Camera being available to Playout, as that is not an expected or desired workflow)

5.8.2 Control Planes

The graphic below describes the scope and overlap between Senders, Receivers, Packaging and Bundling.

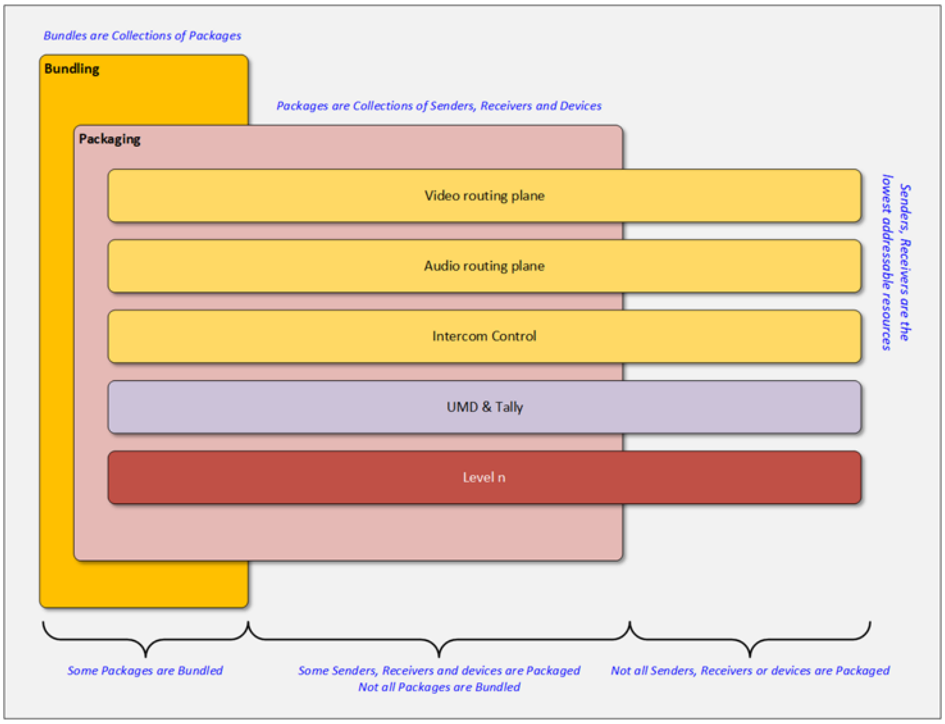

Bundles are Collections of Packages
Bundling
Packages are Collections of Senders, Receivers and Devices
Packaging
Video routing plane
Audio routing plane
Intercom Control
UMD & Tally
Level n
Senders, Receivers are the lowest addressable resources
Some Packages are Bundled
Some Senders, Receivers and devices are Packaged
Not all Packages are Bundled
Not all Senders, Receivers or devices are Packaged

Figure 9 - Scope and overlap between Senders, Receivers, Packaging and Bundling

Senders and Receivers are the lowest level of granularity in the routing system, Packages contain Senders and Receivers, and Bundles contain one or more Packages.

Most routing is carried out at the Package Level, facilitating connecting multiple levels, in both directions in one atomic action (although this is by no means mandated - it must be possible to connect a subset of levels (e.g., Video or Video + Audio - without, say, intercom)

Operationally there are reasons why certain senders and receivers may be excluded from Packaging (e.g., unequipped senders or receivers in a studio that are treated as tielines with no predetermined use).

The package routing system within the facility is envisaged to have “levels” as listed in the following sections (not an exhaustive list).

5.8.3 Source Package

- Video Senders (none, one or several)
  - These may be the same content, via Main/Backup paths OR
  - Same content, different Video format (e.g., UHD 2160p50/HDRPQ & 1080p50 SDR)
- Video Receivers (none or one)
  - Typically used to feed return/cue vision to contributors
- Audio Senders (none, one or many in combination)
  - Content may be split across sender streams (e.g., International Sound on one stream, commentary on another)
  - Multiple channels of content may be encapsulated in a single stream (e.g., 16 ch, including multiple Stereo and/or Surround/Atmos channels)

- IFBs (none, one or two)
    - A pointer to an IFB construct within the intercom system and/or ports on the intercom system to which return audios to contributors/commentators are connected
    - Mix-minus signals from receiving control rooms will need to be routed according to a set of rules. This is likely to involve combinations of FIFO queues, simple routes, and additive mixing, depending on the circumstance
- Conferences [Party Lines] (None, one or several)
    - A pointer to a conference construct within the intercom system and/or ports on the intercom system to which conference audio is connected
- Metadata (multiple fields)
    - Additional operational data such as booking number, start/end times, contact names/numbers, production info
    - Video Standards/Resolutions/HDR Transfer Characteristics (implied or gathered from Video Sender(s)
    - Audio language, usage (Stereo/5.1/Mix/International Sound) tags which may be used to automate audio routing
    - UMD Text (none, one or several) – names which will be displayed on target equipment – e.g., whilst a multiviewer might handle >16 characters, an intercom panel might be limited to six characters, a Vision Mixer to four characters
    - GPIO
- Tally receivers
    - For devices such as CCUs that accept tally signals

5.8.4 Destination Package

- Audio Receivers (none, one or many in combination)
    - Content may be split across sender streams (e.g., International Sound on one stream, commentary on another)
    - Multiple channels of content may be encapsulated in a single stream (e.g., 16 ch, including multiple Stereo and/or Surround/Atmos channels
- Audio Senders [Mix-Minus] (none, one or two)
    - A sender that carries mix-minus from an audio console for onward routing to contributors via the IFB and return audio path defined in a Source Bundle
- Metadata (multiple fields)
    - Video Standards/Resolutions/HDR Transfer Characteristics (implied or gathered from Video Sender(s)
    - Audio language, usage (Stereo/5.1/Mix/International Sound) tags which may be used to automate audio routing
    - GPIO
- Tally senders
    - For devices such as Vision Mixers that generate tally signals. It must also be possible to generate a tally state from logical routing e.g., when a Source Bundle is routed to a Destination Bundle that represents a specific outgoing line.

5.8.5 Audio Package

Depending on the complexity of the eventual requirements, it is feasible that some levels (e.g., Audio Senders, IFB) are defined in separate Audio Package which themselves become levels in a Source Package.

5.8.6 Virtual Package

Virtual Package exist as both Source and Destination Package. Anything routed to a Virtual Destination Package is available (and maintained) as a Source Package. This simplifies the operational experience and allows resource substitution to take place, without the need for manual rerouting of multiple levels.

Both a Virtual Source Package and a Virtual Destination Package can contain pointers to real senders and receivers, which permits operational flexibility – the ability to override / tap off individual levels at points a routing chain.

5.8.7 Package Lifecycle

The majority of sources and destinations will be permanently represented by "Static" Packages. For day-to-day bookings, Packages will be populated and 'cleared down' as necessary.

As changes to a Source Package are committed, they will ripple through the system. Consider a faulty IRD which is swapped out for a working one. Any routes from the original IRD to the new one will be automatically replaced by the Packaging logic. The "ripple" effect must take account of all Virtual Packages in the routing chain.

5.8.8 Requirements

#	Requirement	Stage
1	The BCS should support the bundle and package concept as described above.	v0.5
2	The BCS should support bundle and packages between federated instances.	v0.5

5.9 Device control

BCS must communicate with a wide range of broadcast devices, such as baseband routers, multiviewers, processing devices and video/audio mixers.

Whilst the preferred (and predominant) connectivity for control and monitoring will be TCP/IP, it is anticipated that there will be a limited amount of serial and GPI control required.

Any devices requiring serial communications will use a Serial Device Server to keep legacy cabling to a minimum and allow fast replacement of controlling PCs.

There may also be a requirement to collect and originate physical GPIO signals within the NRK systems (e.g., CCU tally inputs). To facilitate this, the BCS will control and manage a distributed GPIO to IP interface devices, which will ideally present an NMOS IS-07 interface.

Devices that have both control and media (ST 2110 or AES67) Ethernet connections are anticipated to have multiple NICs with different NICs used for each purpose.

Control and monitoring communication will not be conveyed to a device by the media exclusive (AVoIP) network, even if the communication involves routing of signals to/from that device, unless there is no alternative.

5.9.1 Requirements

A list of key device integrations for each stage. Please list relevant integrations.

#	Requirement	Stage
1	Network Controller (Multicast SDN)	v0.5

2	Intercom (Talkback, Comms)	v0.5
3	Media Gateway Connection Management	v0.5
4	Codecs (Video and Audio)	v0.5
5	Assignable Signal Processing (Video and Audio)	v0.5
6	Multiviewer	v0.5
7	Under Monitor displays (Protocols for Label and Tally and User Levels)	v0.5
8	Audio Monitoring Unit	v0.5
9	Signal Analyser	v0.5
10	Test Signal Generator	v0.5
11	GPI Interface	v0.5
12	Media Gateway Parameter Control	v1.0
13	Vision Mixer (BCS tile hosted within mixer UI)	v1.0
14	Audio Mixer (BCS tile hosted within mixer UI)	v1.0
15	Workflow Provisioning	v1.0
16	Resource Management	v1.0

5.10 Intercom

Integration between the BCS and the intercom & talkback (comms) system is fundamental to the live production workflow. The BCS shall provide close integration with the comms system, both locally and to any remote systems linked through trunking arrangements.

It must be able to perform the following functions, assuming the vendor specific API provides a suitable interface allows such actions and/or previously developed methods within the BCS.

The intercom vendor is yet to be chosen, please provide responses for your capabilities for the intercom systems that you currently support in any capacity.

Overall goals are:

- Automate the creation of intercom configuration items based on event requirements.
- Empower end users to make production required changes through a clear easy to use interface.

Clarifications/notes:

- Where a requirement states the need to work on local and remote systems, this infers that the intercoms Trunking system cannot manage the required functionality, therefore the BCS is required to manage it.
- Where a requirement is not possible directly via the API it is requested that responders detail any internal BCS logic that makes the requirements possible.

5.10.1 Ports

Requirements are for all port types; 4-wire & panel. Where a requirement is specific to 4-wire only it will be detailed in (brackets). Fill out one answer per provider you support.

#	Requirement	Stage
1	If more than one intercom system is supported, please specify in the Compliance column like shown here: Vendor A: Y	

	Vendor B: P Vendor C: N	
2	Should be able to set Longname of ports.	v0.5
3	Should be able to set Shortname of ports.	v0.5
4	Should be able to set Aliases of ports.	v0.5
5	Should be able to set Subtitle of ports.	v0.5
6	Should be able to set Input gain control.	v0.5
7	Should be able to set Output gain control.	v0.5
8	Should be able to do Calculation of port usage - crosspoint routes, IFB, Group & Conference.	v0.5
9	Should be able to set Vox on/off Threshold.	v0.5
10	Should be able to do Virtual Key control.	v0.5

5.10.2 Routing

Fill out one answer per provider you support.

#	Requirement	Stage
1	Should support crosspoint routing on local system.	v0.5
2	Should support crosspoint routing on a remote system.	v0.5
3	Should support crosspoint routing between federated instances.	v0.5
4	Should support crosspoint Gain control.	v0.5

5.10.3 Conferences

Fill out one answer per provider you support.

#	Requirement	Stage
1	Should be capable of setting labels of conferences.	v0.5
2	Should be capable of setting aliases of conferences (on local and remote systems).	v0.5
3	Should be capable of setting subtitle of conferences.	v0.5
4	Should be able to add 4wire ports as Talker (always/vox).	v0.5
5	Should be able to add 4wire port as Listener (always/vox).	v0.5
6	Should be able to add 4wire port as Talker/Listener (always/vox).	v0.5
7	Should be capable of setting priority of conference members individually.	v0.5

5.10.4 IFBs

Fill out one answer per provider you support.

#	Requirement	Stage
1	Should be able to set key label of IFB.	v0.5
2	Should be capable of setting subtitle of IFBs.	v0.5
3	Should be able to set input port or group of IFB from local & remote systems.	v0.5
4	Should be able to set mix minus port or group of IFB from local & remote systems.	v0.5

5	Should be able to set output port or group of IFB from local & remote systems.	v0.5
6	Should be able to set Dim level of mix minus.	v0.5

5.10.5 Groups

Fill out one answer per provider you support.

#	Requirement	Stage
1	Should be able to set Key label of Group.	v0.5
2	Should be able to set subtitle of Groups.	v0.5
3	Should be able to add 4wire port from local & remote systems.	v0.5
4	Should be able to add panel ports.	v0.5
5	Should be able to set balancing of audio levels of group members.	v0.5

5.10.6 Functions

Functions can be applied to intercom panel keys, virtual keys, and panel and 4wire virtual functions. This list represents all functions and parameters that are required to be applied by the BCS. Fill out one answer per provider you support.

#	Requirement	Stage
1	Should be able to set all to port.	v0.5
2	Should be able to listen to port.	v0.5
3	Should be able to control call to conference.	v0.5
4	Should be able to control call to IFB.	v0.5
5	Should be ablet to control call to group.	v0.5
6	Should support logic commands in the intercom system.	v0.5
7	Should be able to support remote key in the intercom system.	v0.5
8	Should be able to support send key string in the intercom system.	v0.5
9	Should be able to support control audio patch in the intercom system.	v0.5
10	Should be able to support route audio in the intercom system.	v0.5

5.10.7 Intercom Panels

Fill out one answer per provider you support.

#	Requirement	Stage
1	Should be able to set available functions on keys, virtual keys, virtual functions and GPIO.	v0.5
2	Should be able to set Key mode: Momentary, Latch, Auto.	v0.5
3	Should be able to set Dim mode: Dim, No Dim.	v0.5
4	Should be able to set key label.	v0.5
5	Should be able to set key subtitle.	v0.5
6	Should be able to set Monitoring state of individual keys.	v0.5
7	Should be able to set Icons on keys.	v0.5

8	Should be able to set group colours.	v0.5
9	Should be able to control internal audio matrix crosspoints.	v0.5
10	Should be able to set microphone parameters - type, gain.	v0.5
11	Should be able to set dynamics - compressor, limiter.	v0.5
12	Should be able to set stacking of functions on keys - up to the limit imposed by the intercom.	v0.5
13	Should be able to set lock Function - Ability to set functions to remain in the event that a key is cleared.	v0.5
14	Should be able to set Audio Input/Output - Where the intercom panel supports sendings and receiving audio as analog and/or AES67.	v0.5

5.10.8 GPIO

Fill out one answer per provider you support.

#	Requirement	Stage
1	Should be able to query GPI/O States.	v0.5
2	Should be able to set GPO State.	v0.5

5.10.9 User Interactions

Fill out one answer per provider you support.

#	Requirement	Stage
1	Panel Copy/Paste - enable the user to copy an entire panels configuration to single/multiple panels.	v0.5
2	Key Copy/Paste - enable the user to copy single or multiple keys to single/multiple panels.	v0.5

5.10.10 Mixing

Fill out one answer per provider you support.

#	Requirement	Stage
1	Should be able to present a mix minus group with multiple members as a mixer that allows for adjustment of audio level to all "listeners" of the group.	v0.5

5.11 User Interfaces

All functionality within the BCS must be available through the BCS user interfaces and give the users the ability to interact with the system based on their production role and their login privileges.

The user experience should be recognisable and consistent across user interfaces, e.g., across desktop interfaces, programmable hardware panels, touch screens. The industry best practice to achieve this is to build user interfaces using a design system – a shared design and code collection of reusable components, guided by clear standards. Origo (NRK's own design system for internal NRK systems) and Aksel (NAV's design system – the Norwegian abbreviation for the Norwegian Labour and Welfare Administration) are examples of well-established Norwegian design systems. An international (and best practice) example is Google's open-source design system Material Design.

The BCS interfaces should be possible to use by "as many as possible". BCS user interfaces and components in BCS design system should be built with accessibility in mind, trying to be as inclusive as possible, but meeting at least the minimum set of Web Content Accessibility Guidelines (WCAG).

In order for the designer to utilize the design system it should have a corresponding component library available in a design tool. Figma is an industry standard tool and is the design tool used by UX/UI designers at NRK.

#	Requirement	Stage
1	The BCS user interface must be accessible as a responsive web interface.	v0.5
2	The BCS user web interface should follow proper HTML semantics (e.g., not using table for visual layout).	V1.0
3	The BCS web interface should support the common browser versions from Edge, Chrome, Firefox, and Safari.	v0.5
4	The user interfaces should have full UTF-8 or UTF-16 support in order to support all characters required for Norwegian, North Sami, South Sami and Lule Sami.	v0.5
5	The BCS should have web components available to be used inside 3rd party solutions.	v0.5
6	The BCS user interface should support role-based access control.	v0.5
7	The users should have the ability to personalize the user interface into their needs in a flexible way based on their privileges.	v0.5
8	The users should have the ability to use hard panels if needed, for both routing and parameter control.	v0.5
9	All BCS user interfaces should be built using one common design system.	v0.5
10	The BCS design system should have a corresponding Figma component library, for design collaboration between Contractor and NRK.	v0.5
11	The BCS should support replacing vendor branding with NRK branding, i.e., replace or remove vender logo, use NRK colour scheme, etc.	v0.5
12	The BCS should meet the latest Web Content Accessibility Guidelines (WCAG) at levels A and AA. Currently, the latest version is WCAG 2.1.	v0.5
13	Contractor should use the check list W3Cs WCAG-EM or a similar Accessibility Conformance Report (ACR) to document how the BCS matches the WCAG 2.1 requirements on levels A and AA.	v0.5

5.12 Resource Scheduling

Resource Scheduling features may be an integral feature of the BCS or an external system with integration to the BCS using northbound APIs. This chapter describes the functionality desired for solving some of the workflows NRK have today.

5.12.1 Deliverables

Resource Scheduling for the BCS comprises four main deliverables:

- Time based connection management of pre-defined single endpoints or packaged connections
- Time based event execution
  - Device parameter control
  - Signalling
- Connection conflict detection
- Resource scheduling user interface

5.12.2 Time based connection management and event execution

Time based connection management is about being able to execute connection requests on a schedule and log what has happened.

Time based event execution is about doing scheduled device parameter control and signalling and log what has happened.

- All current and future schedules must have a start time and may have an end time
- All previous schedules must have a start time and an end time
- All future events must have an execution time
- All past events must have an actioned time

The scheduled connection event should update meta information in the destination about connection changes. This in order for an operator to see for instance when a destination was last used.

5.12.3 Real time connection management and event execution

Real time provisioning can be implemented by:

- user input from the BCS user interfaces
- external trigger from a third-party system (e.g., playout)

For endpoints or packages pre-defined as scheduled, manual connections, e.g., from a user interface, must be regarded as scheduled.

All real time connection management and event execution should be immediately represented in the user interfaces.

5.12.4 Scheduled connection management and event execution

Scheduled events can be entered by:

- input into a text-based list display
- a linear time-based graphical display
- external trigger from a third-party system

All scheduled connection management and event execution should be represented on the text and graphical displays in advance.

5.12.5 Connection conflict detection

The system should check and alert the users if there is a connection clash (conflict) whenever:

- a real time connection management event is requested
- a scheduled connection management event is requested
- a scheduled connection management event is edited

Scheduled connections have priority over real-time connections. By default, a real-time connection will be overridden by a scheduled connection.

When a connection conflict is detected, the BCS must present the user with a conflict resolution dialogue.

5.12.6 Resource scheduling user interface

The system should support both graphical and text-based display formats. In either format:

- previous, current, and new schedules should be represented and be visually distinct

- current and future schedules will appear as indefinite unless/until and end time is added, or the schedule is manually closed
- current schedules should be visually distinguishable from future schedules
- individual schedules in a concurrent series of the same connection must be distinguishable

The graphical display should be in the form of linear timelines relating to destination resources. The current time should be indicated on the timelines. It must be possible to:

- filter the visible destination resources by type (user defined)
- filter the visible destination resources by upcoming provisioning events
- zoom the visible time frame in/out
- scroll the visible time frame forward/back

The text display should be in the form of a sequential log of actual and scheduled provisioning events. The current time should be indicated in the text display. It must be possible to:

- filter the visible scheduled events by type and date (user defined)
- scroll the visible time frame forward/back
- add notes to a scheduled event in the log
- add notes to the log
- export the schedule log

5.12.7 Graphical display example (based on current workflow)

NRK's current scheduler UI used to perform and visualize:

- Connection/schedule connection/edit connections (set stop time on a current connection)
- Visualising scheduled items and the connected sources
- Destinations presented in a an "Outlook appointment planning" style visualisation showing scheduled items and the connected sources with adjustable timeline view.

Example of how a resource scheduling can be displayed:

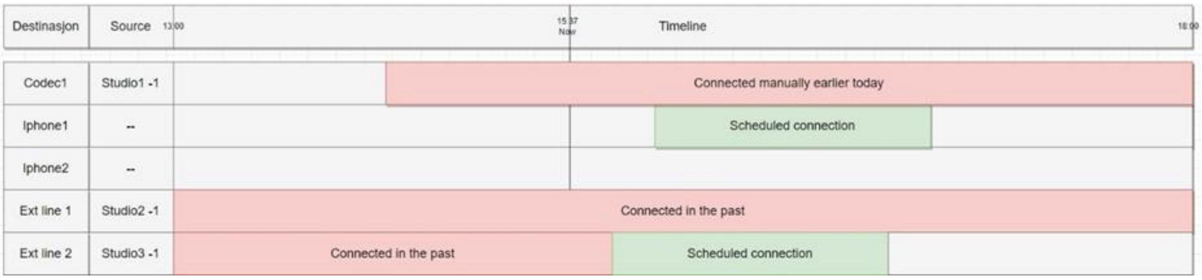
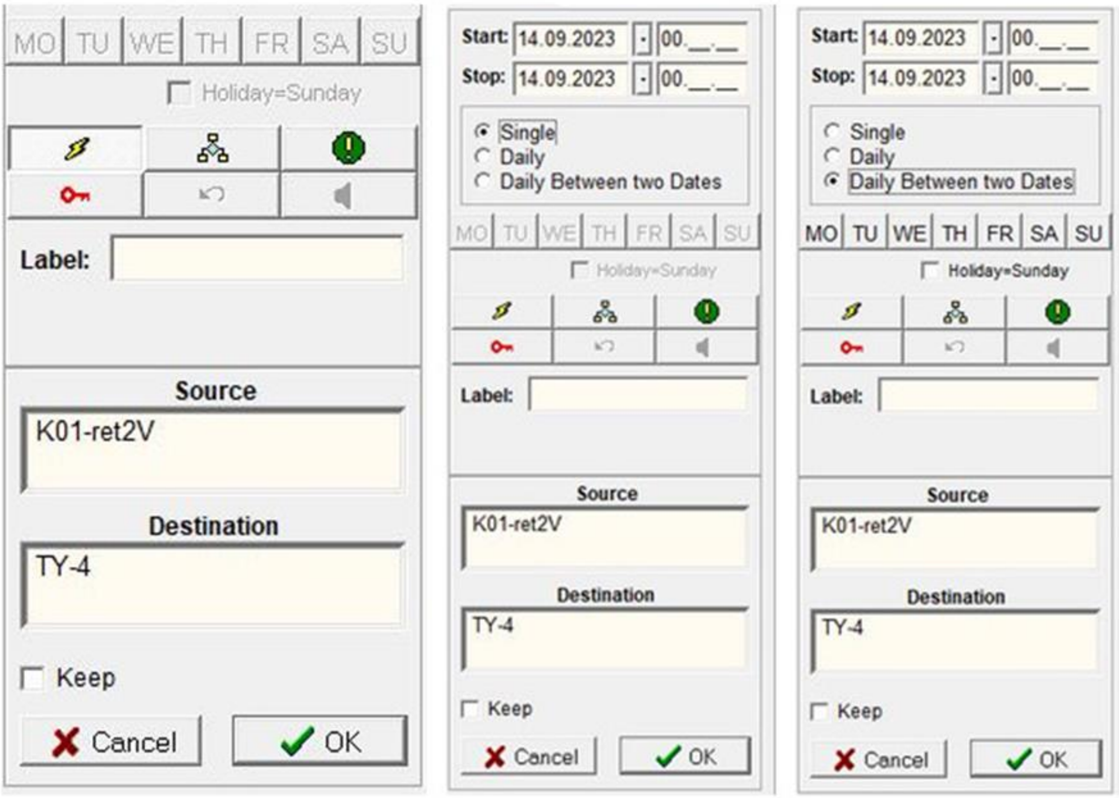


Figure 10 - Resource scheduling example

The scheduler can perform single/daily/Daily between dates connections. It can also suspend daily schedules for a period.

From the UI it is direct access and very easy and fast (few button clicks) to set up a scheduled connection.



□ Holiday=Sunday

Label:

Source: K01-ret2V

Destination: TY-4

□ Keep

Start: 14.09.2023 00._._

Stop: 14.09.2023 00._._

● Single
○ Daily
○ Daily Between two Dates

Start: 14.09.2023 00._._

Stop: 14.09.2023 00._._

○ Single
○ Daily
● Daily Between two Dates

Figure 11 - Resource scheduling example

There is similar functionality for scheduling to a channel (input on a summing amp) and scheduling events like GPIO setting and signalling information to end user UI. In addition, there is also a text based rundown view.

5.12.8 Requirements

#	Requirement	Stage
1	The BCS should comply to resource scheduling as described above.	v1.0
2	In case the BCS do not comply to the above description: Is there an existing integration to an external scheduling tool which comply to the description in combination with the BCS?	v1.0
3	If neither of the two requirements above comply to the description: Is there a preferred scheduling tool for developing an integration with the BCS?	v1.0

5.13 Platform and operational environment

As part of the transition to an IT and standard driven approach for media production, all the systems used for the media production, including the BCS must adhere to common operational best practices. NRK has for many years been at the forefront of this and have been using container-based approach for many of our own applications, first with Mesos and since 2017 with Kubernetes.

We see many benefits from deploying applications on containers and using a container orchestration and runtime environment such as Kubernetes and we would like to see also the BCS, and the rest of the media broadcast system move in the same direction.

While we understand that a container deployment of the BCS can take some time, an alternative approach is to run the BCS as a fully orchestrated and managed service using configuration management. NRK currently uses Saltstack for Linux configuration management, but also uses Ansible some places and are well versed in both.

#	Requirement	Stage
1	The BCS must support one or more of these runtime environments in preferred order: • Running as a container • Running as a Linux service • Running as a Windows service Running as a Windows Gui application is not an acceptable runtime environment	v0.5
2	If not running as a container, the BCS can run on a virtual machine.	v0.5
3	The BCS should be able to be installed by configuration management tools, such as Saltstack, Ansible or SCCM and does not require manual operations for installation.	v0.5
4	The BCS should support the following log configurations based on the runtime environment: • When running as a container, writes logs to standard out/standard error • When running as a Linux service, writes logs either to standard out/standard error or directly to journal • When running as a Windows service, writes logs to the Windows event log	v0.5
5	The BCS should support providing runtime metrics using Prometheus. If Prometheus is not supported, describe the supported runtime metrics system(s).	v0.5
6	The BCS should support OpenTelemetry.	v1.0
7	The BCS should support patching and upgrades without downtime when run in a HA configuration.	v1.0
8	The BCS provides a northbound API using REST or GraphQL providing at a minimum: • The Ability to Make, Park, Request/Poll Package Routes, including registering for Asynchronous callbacks • The ability query state of connections within the system • The ability to update monitoring status within the system from an external PMCS	v0.5

5.14 Legacy transition support

The migration in to NRK's new centralized platform covered in the MPP high level architecture plan, will be a step-by-step approach over many years. VSM will still play an active role in most of NRK's facilities, until the last facility is moved to the new platform, many years from now. In all our production facilities we do live ingest into our MAM system, which currently is Tedial. We use Tedial Capture manager, from now on Tedial CM, locally in each production facility, as an ingest tool. Tedial CM use probel swp08 to talk to VSM, setting up live streams (today SDI) into Quantel SQ servers.

Tedial CM will be replaced by Sofie (an internally developed playout automation system), as an ingest tool, sending route requests into VSM, from Trondheim into new BCS, setting up 2110 transport streams into Black Magic Hyper Decks. Replacement of Tedial CM and Quantel SQ servers with Black Magic Hyper Decks and Sofie on all our region offices are work in progress. How this will be solved in our new HQ at Ensjø is not yet decided, this is work in progress just started up.

#	Requirement	Stage
1	The BCS must support Northbound control of Package Routing using a full implementation of the probel swp08 protocol over IP. The BCS must provide asynchronous state values for Package Routing using the same interface.	v1.0
2	The BCS should support integration with the existing VSM BCS using Probel swp08, exchanging routing and labels.	v1.0

6 INTERACTION WITH EQUIPMENT AND OTHER SOFTWARE (CLAUSE 2.3.2)

Any requirements concerning interaction between the deliverables and software and equipment that the Customer has specified in SSA-T Appendix 3 shall be stated here.

If the Contractor is to be responsible for the integration of software that is delivered together with other software (that the Customer has described in SSA-T Appendix 3), the integration shall be performed pursuant to the requirements specified here.

Here in Appendix 1 the Customer shall specify which integrations the Contractor shall bear responsibility for in respect of their results and progress, and which, if any, shall be delivered as additional services (contribution obligation).

7 IMPLEMENTATION METHOD (CLAUSE 2.3.3)

Any special requirements on the part of the Customer as far as the methods, tools or environment are concerned shall be set out here.

The Customer shall stipulate requirements for how the detailed specification work shall be performed, including any requirements concerning which routines and guidelines shall be followed.

8 DOCUMENTATION (CLAUSE 2.3.6)

The BCS solution including customisations and configurations must be documented so that the Customer has the necessary and correct documentation available in all areas of the BCS solution. The documentation shall facilitate effective use, training, further development, and service of the Solution. The BCS solution shall be provided at least with the following documentation:

Detailed specification - design documentation

The BCS solution is provided with a detailed specification document that is developed during the specification phase. This design document is a description of how the processes and requirements are to be fulfilled in the new solution. The document describes how to solve the desired functionality and the guide to those who will configure and complete the solution for NRK's use.

Training material

The Contractor will provide training material. The training material must be adapted to the individual user group and must be written in Norwegian or English.

User documentation

The BCS solution is provided with NRK-specific user documentation that is:

- Provided in a common electronic format.
- Adapted to different user groups, including advanced users, such as project managers, superuser, system managers, and end-users who will use the BCS solution in a limited way.

Documentation of integrations

The BCS solution is provided with technical documentation in English describing public API, integrations, and APIs specific to NRK (interfaces, methods, etc.).

Functional system documentation

The BCS solution is provided with functional documentation in English describing configurations and setup, including functional design for any customizations.

Technical system documentation

The BCS solution comes with a technical system documentation in English, describing logical service and information architecture, security, and access architecture (ADFS, etc.), provisioning and orchestration functionality and procedures.

Installation and maintenance documentation

The BCS solution comes with installation and maintenance documentation in English. This requirement applies only to on-premises-solutions.

9 TRAINING (CAUSE 2.3.7)

The Contractor shall plan and develop a training program, including training material, and train NRK in the use of the BCS solution. The training shall ensure that NRK can perform qualified testing of the BCS solution and start using the BCS solution properly and efficiently.

All user groups shall receive training. The training shall be designed appropriately for each specific user group. Training methods shall be described.

10 CONVERSION (CLAUSE 2.3.8)

If the Contractor shall convert the Customer's data, this shall be specified here.

If the Contractor shall develop or deliver separate conversion utilities for extracting or inputting data, this must be stated here.

11 PERFORMANCE LEVEL (CLAUSE 4.2)

If the parties have not concluded a maintenance and service agreement, the requirements concerning the Contractor's performance in the warranty period shall be specified here.

12 GENERAL EXTERNAL LEGAL REQUIREMENTS AND MEASURES (CLAUSE 9.2)

The Customer shall identify, in Appendix 1, which legal requirements or statutory requirements applicable to the Contractor (party-specific requirements) are of relevance to the conclusion and implementation of this Agreement. The Customer shall be responsible for specifying, here in Appendix 1, any relevant functional and security requirements that are applicable to the deliverables.

13 PERSONAL DATA (CLAUSE 9.2)

Further provisions governing how personal data shall be processed, including relevant security measures and requirements for storage times and deletion, etc. shall be set out here.

14 INFORMATION SECURITY (CLAUSE 9.3)

Any functional requirements concerning the security of the solution and other security measures shall be stated here.

15 SECURITY FOR ACCESS TO SOURCE CODE, ETC. (CLAUSE 10.2.2)

If the Contractor shall offer an agreement concerning access to source code or some other solution (for example, a performance bond from its parent company or an associated company) that satisfactorily secures the Customer's interests should the Contractor go bankrupt or for some other reason be unable, or cease, to deliver its services pursuant to this Agreement or an associated maintenance agreement, the Customer shall specify such requirements here.

16 THE TOOLS AND METHODOLOGICAL BASIS OF THE CONTRACTOR (CLAUSE 10.6)

If the Contractor, subcontractors, and any other authorised entities shall be prohibited from using tools and methods that they only have rights to during the implementation of the deliverables, this shall be stated here.