EBU

OPERATING EUROVISION AND EURORADIO

R 159

PROCUREMENT OF INTEROPERABLE CONTENT DELIVERY NETWORKS

RECOMMENDATION

Geneva
June 2023

Document History

EBU Committee	TC	
Drafting Group	Broadband Distribution Architecture Group	
First published	June 2023	
Revised		

Publication Keywords: Distribution, Online, Content Delivery Network, CDN, Procurement, Invitation To Tender, ITT, Service Level Agreement, SLA, Site Acceptance Testing, SAT.

Contents

Procurement of interoperable Content Delivery Networks

Recommendation

The EBU, considering that:

- An increasing amount of online delivery capacity is required by EBU Members to reach audiences with their media products.
- EBU Members increasingly depend on private and public Content Delivery Networks (CDNs).
- Interoperability between CDN services is essential for the deployment of a sustainable online distribution operation.

And recognising that:

- EBU Members must be compliant with their company's procurement frameworks and regulatory requirements.
- Every tender is unique, as it combines specific demands with generic requirements.

Recommends that:

The following technical / operational requirements and process considerations are used when procuring CDN deployments[1].

a) When developing their CDN procurement documents, the technical requirements in **Annex A** are mapped to the specific demands of the procurer.

b) The Service Level Agreement example guidelines in **Annex B** be included in the procurement documentation in full or in part. They should be used in negotiations with CDN suppliers to reach agreement on the required support.

c) To use a multi-CDN instead of a single CDN deployment when entering into a multiyear contract.

d) For a multi-CDN setup, recurring 'mini-competitions' on price are organised to fix traffic commitments for a given period within the multiyear contract term.

e) After an initial selection process based on the tenders received, a proof-of-concept phase should be performed that needs to be successful before contracts are awarded.

Note *This recommendation is not a set of prescriptive steps to follow – questions and approach will change depending on the supplier and the procurement.*

Annexes A and B are integral to the recommendation. The following background sections are informative.

[1] CDNs can be Single- or multi-CDN, private- and/or public CDN or hybrid CDN deployments. For more information on CDN topologies used within the EBU Membership, see **TR068** "CDN Architectures Demystified", EBU 07-06-2022, https://tech.ebu.ch/publications/tr068.

Background

Interoperability

As current CDNs do not all offer exactly the same feature set, interoperability is not guaranteed. This leads to high implementation and change costs for CDN customers. This recommendation aims to capture a baseline set of media CDN requirements that are most relevant to EBU Members that can be used to select CDNs in procurement processes. This is both to aid market procurements and to inform the CDN industry of requirements that are essential to integrate technically when doing business with EBU Members.

Not all requirements in the annexes may be relevant to every Member, or every use-case. The goal is to provide them as a baseline, from which individual requirements can be scoped when preparing requirements to be inserted into a tender document. The more the CDN requirements are pushed through procurement processes, the more the market will conform around them and deliver interoperable services.

This document is intended to be used as a reference for procurement of CDN services. The content is primarily geared towards a single CDN provider but can be used for procurement of multi-CDN deployments with private and/or public setups. It covers streaming media use-cases only (e.g., DASH/HLS streams). Web, downloads, or application delivery use cases are out of scope although this document could be adapted for such purposes.

Included in **Annex B** is an example of a baseline media delivery Service Level Agreement (operational fulfilment of requirement during contract term) that may be used as an example of how to measure quality delivery over the lifetime of the contract (alignment of technical requirements with operational requirements).

Business needs

It is essential from both a commercial and operational perspective to forecast and communicate traffic capacity requirements to potential CDN suppliers as part of a tender. This allows each supplier to evaluate if their total service capacity meets the traffic requirements. This traffic estimation is also an essential input into the commercial negotiations.

Typically, the more traffic that can be offered to a CDN, the better the unit price they can offer. This can be a reason to tender for a single CDN instead of the more redundant multi-CDN setup. A multi-CDN setup nevertheless provides more flexibility and / or capacity.

It is common for content providers to experience seasonal variation in relation to their delivered traffic volumes over the course of a calendar year, for example, with lower traffic in the summer and higher traffic in winter. It is important to accommodate this within the tender to be clear to suppliers about how they should expect to supply your required capacity and to incorporate this into their own capacity planning. The average & peak capacity requirements may need to be specified differently throughout the year, including:

- Average daily expected peak (e.g., the max Gbit/s to deliver the expected daily viewing peak).
- Maximum expected peak (e.g., max Tbit/s to deliver important live events).

These traffic requirements have an impact on costs. For example, one needs to avoid having to pay more because minimum commitments have not been met or maximum commitments have been exceeded in a certain month. To prevent this, one can either have an accurate prediction (and ability

to transfer traffic commitments over to other periods) or have a large bucket that covers a longer term.

Other factors that may be important to identify are markets or geographic regions that have specific demand profiles, as the presence of infrastructure in those markets may be required to deliver the traffic to your expected performance criteria. For example, one can agree on a flat rate for traffic, combined with a regional usage pattern stating fixing commitments. Normally, EBU Members deliver their major traffic nationally, but some traffic is also delivered internationally.

For a multi-CDN setup, recurring 'mini-competitions' on price are organised to fix traffic commitments for a given period within the multiyear contract term. In such a multi-year contract one needs to detail how commitments can vary between each mini-tender per supplier and how this is decided.

Process Considerations

The typical tendering process consists of the following stages:

- Requirements gathering & documenting.
- Brief market assessment of known providers.
- Writing a formal Invitation to Tender (ITT) document.
- Publication of tender.
- Evaluation of tender responses.
- Request for Proposal (RFP) invitations.
- Evaluation and scoring of RFP responses.
- Selection and contract negotiation.
- Proof-of-concept, if applicable.
- Contract award.
- Implementation.

The technical requirement list in **Annex A** can be copied in whole or in part, depending on how it aligns with your needs. In the procurement documentation the requirements need to be categorised with a priority level that will define the importance of each requirement to your operation. The requirements indicated with a high priority are considered to be essential[2]. The detail of the requirements is general enough so as not to unnecessarily complicate the terms between a CDN provider and the Client. Adding use cases can improve the conversation with providers to understand your needs.

A formal ITT document should describe the contract organisation and selection process. It should be clear from the start if it is the aim to have a multi-CDN solution with recurring mini-competitions fixing commitments to pre-selected suppliers. All EBU Members have their own procurement frameworks they must comply with. In general, transparent conditions are published in a tender portal to invite and inform bidders. It is advisable to put SLA requirements in the ITT and clarify SLA conditions (on the basis of the example SLA in **Annex B**) in the negotiation process.

After initial agreement, a Proof-of-Concept phase is strongly advised, preferably as part of the selection process, before finalising contracts. This allows testing of the solutions, capabilities and support of the selected vendor(s). Important topics to test are request collapsing and connecting to

[2] For example, Low Latency is not supported by all CDNs. By categorizing as essential will automatically reduce the companies that can reply to the ones who already support it. This is not an issue if it is essential to your operation, however if it is not, it's inclusion could increase the price of the tendered solution.

origin, amongst others. Agreement on a migration/implementation plan including SAT (Site Acceptance Testing) is also advised as part of the selection process.

Requirements categories

Functional requirements are listed in **Annex A**. To aid the grouping of related requirements, the following categories have been identified:

Media

Being specific about the kind of media to be distributed and the packaging formats that are required. The Origin is not part of the CDN requirements and in this document the Origin is assumed to be controlled by the content provider. It is also assumed that the Origin can handle both live and on-demand content as part of supporting required media delivery formats.

- Audio, Audiovisual and static images. .
- MPEG-DASH / HLS-TS / CMAF / Smooth Streaming support.
- Generic traffic, i.e., picking up objects and deliver them over Unicast.
- Compatibility with Low Latency approaches, e.g., WebRTC.
- Multicast capability.

Resilience

Different approaches to streaming resilience can be taken. There is typically a set of trade-offs to be made between resilience, latency, and cost, depending on what you need as a content provider. The ability of a CDN to configure particular parameters and behaviours may be important to achieving a particular resilient architecture.

- How should failover work? For example, specification of failover rules to a number of resilient content origins.
- What specification of time outs are needed? Client-side & server-side and / or at each CDN tier?
- Anycast support.
- DNS TTLs.

Delivery Capacity

Processes that increase efficiency of traffic delivery.

- Compression techniques dynamically delivered to the clients.
- Transfer handling for example with live content, packages should be delivered to end users before an object has been fully fetched from origin.

Observability

Knowing how well a particular CDN service is performing is crucial to understanding its value for money and the quality of service being provided. Different approaches can be taken that can have different cost implications or require purchasing of separate products.

- Monitoring & metrics – real-time & historical.
- Log delivery, e.g., to cloud object stores for later processing.
- Native reporting functionality.

Origin Protection

Content origins must be protected from 'thundering herd' traffic storms, especially for live streaming. Also, requests to origin can be costly if not managed to a minimum. CDNs have different capabilities depending on their underlying architectures.

- Primary/secondary origin failover.
- Request collapsing.
- Request shielding. e.g., 'shield POPs' to further increase request collapsing efficiency.

Cache Control

The ability to set & manipulate default HTTP Cache-Control headers may be important for certain streaming use-cases, also the ability to purge objects from the CDN that need to be made unavailable.

- Content Purging amongst others.
- Cache TTL manipulation.

Custom Behaviours

It can be important to be able to perform detailed manipulation of both HTTP request & response headers, either from client to CDN edge or from CDN to content origin.

- Header manipulation amongst others.
- Native support within CDN distribution configuration or use of edge/near-edge compute functionality.

Content Protection

It may be important to be able to protect content from unauthorized access and playback.

- DRM (centrally or at the edge, JIT).
- Anti-deep linking measures.
- Token authentication – in-URL or cookie-based carrier (JWTs).

Control Plane

This provides the ability to implement and make changes to new and existing CDN distribution configurations. There is a spectrum of capabilities required (depending on levels of internal expertise and staffing), from total self-service to managed changes implemented by the providers themselves, e.g., under a professional services arrangement.

- API access.
- Web portal.
- Config-as-code, e.g., Terraform provider support.
- PS Hours.

Change Control

The ability to manage changes to CDN distribution configurations either programmatically or via a remote process.

- Versioned changes that can be rolled back.
- Non-live environments, e.g., 'staging'.
- Change windows.

Security

- Keeping content, configurations and data secure, also protecting content origins. User & API account security.
- RBAC permissions scope.
- DDOS protection.
- WAF functionality.

Sustainability

- References to compliance with sustainability ISOs and other best practice standards.

Service Level (see Annex B)

- Support (response times to questions, what is an answer, real time support for live events).
- Delivery KPIs.
- Payback schemes, e.g., Service credits when not delivered.

Annex A: Functional Requirements

The following is a table of CDN functionality that we consider to be important as they relate to streaming media. It is strongly recommended that EBU Members prioritise functional requirements in line with their specific delivery needs (e.g., indicating MoSCoW or High, Medium, Low prioritisation).

Functionality	Category	Description/Example	Importance
MPEG-DASH / HLS-TS / CMAF	***Media***	Support of latest version of most common media transport formats used for online delivery, where MPEG-DASH and CMAF are open specifications and HLS-TS is proprietary but widely supported.	
Origin Failover	***Resilience***	Automatic failover to a secondary origin if the primary origin is unavailable. Timeouts and conditions must be configurable. Failure conditions can include a set of response codes (404, 500, etc.) as well as connection failures.	
Custom timeouts for connect / TTFB / TBB / total transfer	***Resilience***	We must be able to ensure that the CDN can “fail fast”, rather than keeping clients waiting indefinitely if an origin is slow.	
Quick Retry	***Resilience***	Fetching of content from a secondary origin within the same HTTP request/response flow as the original request to the primary origin.	
Origin / client keep-alive	***Resilience***	HTTP keep-alive must be supported to both the origin and the client, to avoid connection churn.	
416 responses for out-of-range requests	***Resilience***	Some clients “guess” the size of a file and send range requests beyond the end of the file. Endpoints should respond with a HTTP 416 response rather than a 200, as a 200 generally indicates to said clients that range requests aren’t supported	
Origin Health Checks	***Resilience***	Periodic health checks to determine the status of customers' origin servers, automatically failing over if a given origin is detected as being unhealthy or unreachable.	
Redirects should be avoided	***Resilience***	HTTP redirects as part of the client request flow (e.g., to direct users to other cache nodes) should be avoided if possible as this can cause performance problems for some clients	
stale-while-revalidate / stale-if-error	***Resilience, Origin Protection***	Content should continue to be served “stale” (if available) while revalidating against the origin.	
Gzip Compression	***Delivery Capacity***	Compression of content where the request headers have stated support for gzip compression. Support enabling/disabling this functionality on a per content-type basis as device support can vary.	

Functionality	Category	Description/Example	Importance
Brotli Compression	***Delivery Capacity***	Compression of content where the request headers have stated support for Brotli compression. Support enabling/disabling this functionality on a per content-type basis as device support can vary.	
Streaming Transfers	***Delivery Capacity***	Delivery of content to end users before an object has been fully fetched from origin.	
(Near-)real-time throughput / request rate per distribution (host)	***Observability***	Metrics that will allow customer to determine the health of a given CDN to perform operational actions.	
Access log availability	***Observability***	Availability of access log files via a method of the customer's choice in a timely manner. This could be either: • Shipping of log files to an object store, e.g., AWS S3 • Making log files available for download from a data store, e.g., via FTP or object store API Logs should be shipped in a format that matches customer's requirements.	
Diagnostic headers	***Observability***	Ability to see variables like cache status (hit/miss), POP/node used etc as a client via response headers, perhaps by sending a "debug" request header or similar.	
User Agent Filtering	***Origin Protection***	Filtering of requests from known malicious, or otherwise misbehaving user agents, at the CDN edge.	
Mid-Tier Cache	***Origin Protection***	Often known as an 'Origin Shield', a mid-tier cache will allow customers to scale against a predictable set of known hosts that will be used to connect directly to our origins, rather than receiving requests from every CDN edge server or every POP.	
Origin DNS Re-resolution	***Origin Protection***	Allow customer to change DNS records for an origin, and have these changes automatically propagate to the CDN without manually re-loading cached DNS responses on the CDN platform.	
Request Collapsing	***Origin Protection***	Multiple (almost) simultaneous requests for the same uncached object will not generate multiple requests to origin. Requests will wait for the first response from the origin then the cached content will be sent to all end users.	
Protected Origin Connectivity	***Origin Protection***	All origin connectivity can be protected with Mutual TLS rather than shared-secret or unauthenticated HTTPS/HTTP.	

Functionality	Category	Description/Example	Importance
Setting different default TTLs based on certain criteria, e.g., file extension or origin response code	***Cache Control***	Certain 4xx Responses need to be cached within the CDN, but with a much shorter TTL than is specified on the origin response, in order to facilitate quicker retrying of the object. **Example:** Manifest files or media chunks should be retried after a 404 origin-response quicker in order to recover more quickly from stream errors.	
Cache Invalidation	***Cache Control***	Manually clearing the cache for an object that has been changed, via an API call or control panel. **Example:** A corrupt or missing cache entry may need to be cleared in order to either delete the content from the CDN completely (for compliance reasons) or to recover from stream errors.	
HTTP and HTTPS requests share the same Cache Key	***Cache Control***	The HTTP scheme in the cache key can be ignored so that incoming HTTP and HTTPS requests share the same cache. **Example:** Manifests & media chunks are the same content regardless of the surety of the underlying transport. Clients may transfer using HTTP or HTTPS, but if an origin is available only over HTTPs, the requests should be considered equivalent for cache fill purposes.	
RFC-compliant caching	***Cache Control***	Ensuring content is cached (or not) in accordance with expected standards (RFC 7234 / RFC 9111).	
Default caching	***Cache Control***	Default caching rules, defined by the customer, for any origin responses that don't have explicit cache-control headers. **Example:** Not every origin may be configured to express default caching rules.	
Range normalization	***Cache Control***	Consistent (and cacheable) responses to HTTP RANGE requests	
Normalising HTTP Paths	***Cache Control***	Support for normalising relative http paths (handling of ../../, etc.)	
Query String Stripping	***Custom Behaviour***	Removal of (part of the) query strings from a request path: Only a specified list of parameters should be used in the cache key and forwarded to the origin, preferably in an order-independent way. This is to prevent cache-miss behaviour caused by query string parameter manipulation, and overwhelming of the origin as a result. **Example:** Specific URL query strings need to be allowed through to origin requests, but otherwise blocked and discarded from any cache key.	

Functionality	Category	Description/Example	Importance
Hostname to Origin Mapping	***Custom Behaviour***	Support for multiple edge hostnames, mapping these to different origins as required. **Example:** The requirement that several edge hostnames be managed by the same configuration or distribution within the CDN, and for the hostnames to be available to be mapped to distinct content origins as required.	
HTTP Header Manipulation	***Custom Behaviour***	Addition, modification, or removal of HTTP headers at the CDN layer. **Example:** The addition, modification and removal of specific HTTP headers & their values from both edge responses and origin requests & responses, e.g., Cache-Control & Access-Control-Allow-Origin (for implementation of CORS at the edge).	
TLS Configuration	***Custom Behaviour***	Allow customers to define own TLS configuration for each CDN distribution. Variables include: Ciphers, TLS Versions and Session Ticket support.	
Ability to configure HTTP versions used	***Custom Behaviour***	Ability to turn HTTP2 / QUIC / HTTP3 on/off for specific distributions.	
Ability to turn IPv6 support on/off	***Custom Behaviour***	Ability to turn IPv6 on/off for specific distributions.	
Low Latency	***Custom Behaviour***	Provide support for Low Latency variants of MPEG-DASH, HLS and CMAF.	
Geoblocking / VPN Blocking	***Content Protection***	Important for enforcement of content rights agreements. May be different for on-demand vs. Live streaming. **Example:** Content must be made accessible/inaccessible by clients on a per-country basis based on the geolocation of the client request. VPN exit nodes must also be blocked within countries that would be otherwise accessible, in order to comply with content rights agreements.	
Token authentication	***Content Protection***	The ability allow/deny edge requests based on the presence of a valid token, present either on the URL or passed by the client with a HTTP request header, e.g., a JWT token. **Example:** It is more difficult for end-users to share links to locations where content can be retrieved.	

Functionality	Category	Description/Example	Importance
Custom Geolocation Databases	***Content Protection***	The ability to integrate and use specific geolocation / proxy detection databases/providers. This can be important to fulfil certain content rights agreements. **Example:** The contract is with Geolocation service X, and all content needs to be protected at the CDN edge with the same database rather than the vendor default, so that compliance can be maintained with specific rights agreements that state that provider X must be used to protect the content.	
Geolocation overrides	***Content Protection***	The ability to configure exceptions to the default geolocation or geoblocking behaviour. **Example:** The need to allow certain fixed IP address ranges to access content that would otherwise be blocked due to their geolocation, i.e., foreign offices, education establishments, or partner organisations.	
GeoIP blocking	***Content Protection***	The ability to block certain IP addresses or address ranges to override the default behaviour. **Example:** The need to block certain fixed IP addresses or ranges from accessing content in my country that would otherwise be allowed due to default geolocation.	
CDN-managed TLS certificate support	***Content Protection***	Allow the provision and automatic management of X.509 TLS certificates for CDN distributions. CDN to be responsible for certificate issuance. Auto-renewal and deployment of CDN-managed certificates without customer intervention.	
Customer-supplied TLS certificate support	***Content Protection***	Allow the supply and configuration of customer-provided TLS certificates for CDN distributions. Customer to be responsible for certificate issuance and supply.	
Signed Requests	***Content Protection***	Implementation of a signed request format according to an agreed standard.	
Webportal and API	***Control Plane***	Ability to implement and make changes to new and existing CDN distribution configurations either/and through webportal or integrated via API-calls.	
Pre-live environment available for testing	***Change Control***	Ability to test changes in a pre-live environment before promoting to live.	
Configuration as Code	***Change Control***	Ability to deploy configuration changes using a CI/CD based process, including progressing changes through non-live environments.	
Change Management	***Change Control***	Ability for customers to make changes to the CDN configuration themselves rather than require CDN support or professional services teams. **Example:** The ability to create, deploy and modify a CDN configuration via declarative Infrastructure-as-Code or in a configuration language, e.g., Terraform.	

Functionality	Category	Description/Example	Importance
SSO, API	***Security***	Request usage of your own SSO solution to gain access to CDN Provider's administrative platform for authentication.	
RBAC	***Security***	The access each user gets in the administrative platform must be controlled using role-based access control (RBAC) and should be able to use groups synced from procurer's user directory.	
API Access Tokens	***Security***	Support API access using supplier specific access tokens or OAuth2 (incl. RBAC).	
Access and Audit Logs	***Security***	In order for procurer to have control and tracking of who has done what in the platform, there should be audit logs for all operations done on procurer's configurations.	
DDOS, WAF, Bot Protection and Error Logging	***Security***	PSM has a community mission to be able to serve content to citizens, but at the same time their mission and journalistic coverage also makes them a highly valued target by many threat actors, from "script kiddies" to nation states. On this basis, it is important that the CDN solution is able to handle and mitigate DDoS and other typical web attacks, using solutions such as: • Support Web Application Firewall (WAF) including automatic updates to the Web Application Firewall ruleset. • Support for detecting and mitigating bot traffic. • Support for creating custom error pages when traffic is blocked due to DDoS mitigation, Web Application Firewall rules or bot traffic mitigation and insert them automatically in procurers monitoring tools.	
Sustainability targets	***Sustainability***	Request explanation how they operate sustainably, what their targets and plans are, and what changes they are making. This ideally includes footprint calculations, such as those using the GHG protocol3, reporting such as the Carbon Disclosure Project CDP4 & the Task Force on Climate Related Financial Disclosure TCFD5 targets e.g., Science Based Targets SBTi6, and any organizational memberships and performance accreditations such as ISO 50001.	

Annex B: Example CDN SLA

This is an example of a Service Level Agreement (SLA) framework applicable to the delivery of media streaming services through a third-party Content Delivery Network. It recommends some example qualities that are important to the delivery of media streaming from a customer point of view. The SLA should also include a recurring review mechanism to evaluate performance.

General Principles

It is advised that CDN SLA should clearly define:

- Service levels, metrics and the penalties/credits associated with being in breach of them.
- A measurement period – the length of time each SLA will be calculated for.
- A methodology for monitoring and reporting on the SLAs.
    - How each service will be tested – which tools, where from, how often.
    - How the performance of each SLA will be calculated from the data generated.
- A definition of 'success' and 'failure' of a test.
- A definition of a service 'outage'
- Any exceptions to either the measurement methodology or individual SLA definitions as a result of technical or practical constraints.
- Any obligations on the part of either the vendor or the customer in order for the SLA to be valid, e.g., Provision of test objects, minimum overall traffic levels, or minimum spend if not defined elsewhere.

Example Measurement Methodology

Test type	HTTP GET Requests
Test frequency	Every 5 minutes
Test source locations	3 geographic locations
Test targets	15 kbyte test object, located on a content origin owned by the customer
Test method	2x parallel requests: 1 to nearest CDN edge, 1 to content origin.
Measurement method	HTTP Response codes and response times from each target will be compared to measure each defined SLA metric.
Measurement period	Monthly

In this example document, there are four main SLA categories that are relevant to media streaming:

- Availability
- Throughput
- Rebuffering
- Support/Incident Management

Availability SLA

Metric Definition	Percentage of total number of requests for a Stream Type that resulted in successful HTTP responses from the CDN edge, during the Measurement Period. • "Successful" HTTP response codes: 2xx, 3xx, 4xx *(i.e., not server-side errors)* • "Unsuccessful" HTTP response codes: 5xx
"Minimum Volume"	The minimum delivered traffic volume for a Stream Type for the Service Level to apply. If the minimum volume is not reached within the Measurement Period, the SLA will not apply. *N.B.: This allows for variable traffic volumes for streams that may not always be active, or for Stream Types that may carry much smaller or greater volumes on average than others – e.g., 4K/UHD video content, or audio streams.*

Example

Stream Type	DASH + HLS Live HD Video	DASH + HLS HD VoD	HLS Live Audio
Minimum Volume	50 Tbyte	100 Tbyte	10 Tbyte
Service Level	99.99%	99.99%	99.99%

Throughput SLA

N.B.: Throughput SLAs may not be applicable for segments that contain audio only as they are too small, or you may want to have separate audio-only SLAs with smaller segment size definitions.

There can be different Service Levels applicable to different stream bitrates within an ABR ladder. For instance, a lower bitrate stream may be held to a higher threshold, whereas a higher bitrate stream may not always be reasonably expected to perform to the same threshold. Therefore, the SLA usually needs to accommodate these differences to be useful, as in the example below.

In practice, it is advised that thresholds should be set based on measuring real-world performance to establish what 'good' looks like and reflecting the business priorities of the overall media delivery proposition.

Other real-world variables may need to be accommodated as distinct SLAs, for example throughput in different geographic areas, or excluding specific source ASNs.

High Bitrate Metric Definition	Percentage of responses served with a throughput above a defined high bitrate threshold, compared to the sum of requests for the Stream Type, during the Measurement Period.
Minimum Bitrate Metric Definition	Percentage of responses served with a throughput above a defined minimum bitrate threshold, compared to the sum of requests for the Stream Type, during the Measurement Period.

Example

Stream Type	DASH + HLS Live HD Video	DASH + HLS HD VoD	HLS Live Audio
Minimum Volume	5 Tbyte	10 Tbyte	500 Gbyte
Service Level (min bitrate)	99.99% (1.5 Mbit/s)	99.99% (1 Mbit/s)	99.99% (128 kbit/s)
Service Level (high bitrate)	95% (5 Mbit/s)	97% (3.5 Mbit/s)	99.95 (384 kbit/s)

Rebuffering SLA

N.B.: From a client player POV, "rebuffering" occurs during stream playback when media segments of a certain time length are not downloaded from the CDN edge in enough time for the player to sufficiently fill and maintain its playback buffer, and the buffer is exhausted, resulting in the stream pausing, 'spinning wheel' behaviour, and possible CDN failover depending on how the client is configured to behave. This SLA defines the delivery quality necessary to maintain a sufficiently filled buffer relative to the AV segment length defined in the media packaging configuration.

Metric Definition	Percentage of media segments downloaded from the CDN below the defined Segment Length Threshold, compared against the sum of requests for media segments, for each Stream Type, during the Measurement Period.

Example

Stream Type	DASH + HLS Live HD Video	DASH + HLS HD VoD	DASH UHD VoD	HLS Live Audio
Minimum Volume	5 Tbyte	10 Tbyte	25 Tbyte	500 Gbyte
Service Level (segment length)	99.5% (5s)	99.5% (5s)	95% (5s)	99.5% (8s)

Support/Incident Management SLA

It is advised to agree on the following metrics:

- Initial response time, which could be a first line acknowledgement (e.g., 15 minutes.).
- Time to provide a meaningful response with a commitment to when a problem will be resolved. These are measured in mitigation and remedy times for any operational incidents raised (e.g., 1 hour mitigation, 8-hour remedy).
- Mean time to remediate (MTTR) is an example to measure overall problem resolving performance over a period of time (e.g., one year, contract term).

Examples of Penalties for SLA Breaches

It is suggested to define penalties in monetary terms which should be reimbursed as opposed to a service credits approach. This will ensure service underperformance is noticed in supplier's organisation. Elements to take into consideration:

- Breach days, as a pro-rata refund of the monthly fees for each day at least one SLA breach has occurred.
- Fixed penalty fee deducted from monthly fees.