Prime Video Playback Metric Dictionary

Version	v12
Last Updated	10/16/2023
Changed By	schafand@

Prime Video Playback Metric Dictionary

1. Introduction

> This purpose of this document is to define the Quality of Experience (QoE) metrics and engagement metrics for Prime Video. This is a living document that provides a comprehensive list of metrics supported and maintained by Playback Analytics. The goal for this document is to be the single source-of-truth for metric definitions and calculation and help stakeholders within and outside Prime Video understand our metrics.

2. Anatomy of the Metric Dictionary

In order to drive a deep understanding of our metrics, we define the following key attributes for every metric.
1) Metric definition: clarifies the purpose of the metric ideally in terms of capturing the customer's experience. We strive to create a crisp definition of the metric that is easily understood by all stakeholders.
2) Use Case: Clarifies how this metric should be interpreted and used.
3) Session Calculation: Captures how the metric is calculated for every session when applicable.
4) Aggregation: Clarifies the metric calculation when it is aggregated across sessions for a selected time period. Aggregation metrics are critical to understand issues across sessions and spot trends in QoE degradation.
5) Adjustments and Sanitization: Calls out important inclusions and exclusions in the metric calculation so that the metric represents the ground truth of customer experience and is not skewed by outliers or exceptions.

Metrics definition change log:
https://w.amazon.com/bin/view/Amazon_Video/PV_Playback/Docs/Analytics/Metric_Dictionary/Changelog

3. Metric Types

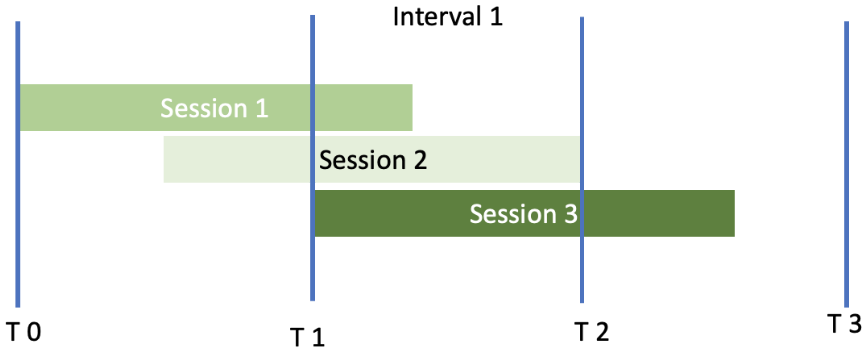
We classify our metrics into 3 types:

Audience Metrics: These metrics help provide the volumetric measure of how customers are interacting with the service. Examples include customers, devices, plays, concurrent plays etc.
Quality of Experience Metrics: Quality of Experience (QoE) is a measure of the delight or annoyance of a customer's experiences with a service. Example include *Rebuffer,* and *Time-to-First-Frame*. This is different than Quality of service (QoS) which is the description or measurement of the overall performance of a service. To quantitatively measure quality of service, several related aspects of the network service are often considered, such as packet loss, jitter, throughput and latency.
Engagement Metrics: These metrics help us understand the level of customer engagement. Examples include minutes viewed, minutes viewed/ unique, average session duration and average percentage complete.

3.1 METRIC CALCULATION TERMINOLOGY

In order to understand the metric calculations, we define a few important terms that determine when and how a metric value is calculated. To interpret a metric value, we consider two important elements 1) the filter that we apply to the metric example Live, VoD, Player, CDN etc 2) The time range for which we are evaluating the metric i.e. minute, hour, day, week etc. During aggregation, we classify metrics calculations into 1) interval metrics and 2) lifetime metrics.

Interval metrics calculate the value of the metric for all sessions in a selected time interval and includes both ongoing sessions as well as ended sessions in that interval. These metrics are extremely important to diagnose degradation in QoE issues since we do not want to wait for a session to end to understand a degradation. Examples include *Percentage Time Spent Buffering (PTSB)* and Average Bitrate. In the example below, Interval 1 aggregates metrics for Sessions 1,2 and 3 where Session 1 ends in interval 1, session 2 is ongoing and session 3 starts in interval 1. Further, it aggregates the value of the metric for each session from T1 to T2 only.



Lifetime metrics include the metrics for all sessions that have ended and includes the complete metric value for the entire session length. In the above example, Lifetime metrics for time interval T1 to T2 can only be computed for sessions 1 and 2 since they both have ended and we can determine the value of the metric for the entire session. An example of such a metric is ZBR (Zero Buffer Rate). A session is considered towards zero buffer, if it does not buffer for the lifetime of the session. Frequently through the metric definitions, we will also refer to metrics as ended metrics. An ended metric is calculated when a play ends in the selected time interval and includes the lifetime value of that session in the interval that it ended.

4. Audience Metrics

The key audience metrics are as follows:

SESSIONS

Availability : Available
Definition: Counts all distinct streaming segments. For lifecycle data, this is every streaming segment that ended in the time window. For interval data, this is equal to the number of distinct streaming segments that appear in the window, whether they started in the window or before it.
Use Case: This metric is useful for understanding the number of streaming segments that were active at some point in the window. This is especially useful as a denominator for understanding the customer impact.
Session calculation: 1
Aggregation : Distinct count of all streaming segments.
Adjustments and Sanitization: None
Sample Query:

```
SELECT COUNT(DISTINCT streaming_segment_id) as sessions
FROM default.fact_streaming_segment/default.fact_interval_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

ATTEMPTED PLAYS

Availability : Available

Definition: Counts all attempts to play a video which are initiated when a viewer clicks play or a video auto-plays. An attempt can result in a successful play, or an early termination due to a Video Start Failure (VSF) or an Exits Before Video Start (EBVS).

Use Case: Understanding this metric provides insight into how many customers started to play a video, and when compared with successful plays, allows us to understand issues with the start-up funnel of the playback experience.

Session Calculation: Not applicable

Aggregation: Sum of all attempts for the selected time interval. Note: *Attempted Plays* = *Successful Play*s+ *Video Start Failures*+ *Exits Before Video Start.*

Adjustments and Sanitization: None

Sample Query:

```
SELECT COUNT(DISTINCT streaming_segment_id) AS attemptedPlays
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

SUCCESSFUL PLAYS

Availability : Available

Definition: A play is counted when the viewer sees the first frame of video and does not include unsuccessful attempts to play. This metric shows the number of plays that started during a selected time period. As a percentage, this metric shows the percentage of attempts that resulted in plays in the selected interval. Successful Play is counted in the interval in which the play started.

Use Case: Understand if we are losing customers in the start-up funnel. Diagnose impact of start-up failures and exits before video starts.

Session Calculation: NA

Aggregation: Sum of all the sessions that had the first frame of video rendered in a time interval.

Adjustments and Sanitization: None

Sample Query:

```
SELECT COUNT(DISTINCT streaming_segment_id) AS successfulPlays
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
    AND content_duration > 0
```

PEAK CONCURRENT PLAYS (PCP)

Availability : Available

Definition: The peak concurrent play is the maximum number of simultaneously active plays for the selected time interval. The Concurrent Plays metric only counts active sessions where the video has started playing before or during the interval, including sessions with re-buffering and other trick modes that do not exceed a defined inactively limit (see sanitization).

Use Case: Capacity planning and scaling of the service components. Peak audience engagement, especially for Live events.

Session Calculation: NA

Aggregation: Max (active plays in a time interval)

Adjustments and Sanitization: Any session that buffers continuously for 30 seconds is terminated with a fatal error. Further, when pause duration of a session (user initiated or backgrounded sessions) exceeds an hour then the session is automatically terminated by the backend and no longer counts towards this metric.

Sample Query:

```
WITH minutes AS (
    SELECT
```

```
        CAST(date_column AS timestamp) AS minute
    FROM
        (VALUES
            (SEQUENCE(timestamp '2021-08-01 00:00:00',
                      timestamp '2021-08-02 00:00:00',
                      INTERVAL '1' MINUTE)
            )
        ) AS t1(date_array)
    CROSS JOIN
        UNNEST(date_array) AS t2(date_column)
), streams AS (
     SELECT
          streaming_segment_id,
          MAX(from_unixtime(start_time/1000)) AS startTime,
          MAX(from_unixtime(stop_time/1000)) AS endTime
      FROM default.fact_streaming_segment
      WHERE publish_timestamp BETWEEN timestamp '2021-08-01 00:00:00' AND timestamp '2
      GROUP BY 1
), aggregates AS (
    SELECT
          minute,
          SUM(CASE WHEN minute BETWEEN startTime AND endTime THEN 1 ELSE 0 END) AS str
      FROM streams
          INNER JOIN minutes
              ON minute BETWEEN startTime AND endTime
      GROUP BY 1
)

SELECT MAX(streamCount) AS peakConcurrentPlays
FROM aggregates
```

CUSTOMERS (UNIQUE)

Availability : Available
Definition: The total number of unique customers defined by PV customer id over a selected time interval.
Use Case: Understand how may unique customers tuned into the service for a selected period of time. De-dupe retries by the same customer and multiple sessions played by 1 customer.
Session Calculation: NA
Aggregation:
Adjustments and Sanitization:
Sample Query:

```
SELECT COUNT(DISTINCT encrypted_customer_id) AS uniqueCustomers
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

5. Quality of Experience (QoE) Metrics

The QoE metrics are listed in the order of the play funnel. The video playback funnel can be classified into two distinct sections 1) The start-up funnel: This represents the playback funnel from when a customer attempts to play video by clicking “Play” or by auto-play until the first frame of video is rendered which constitutes a successful play 2) play funnel: This represent the customer journey from when the video starts playing to end of playback. The merits of classifying the metrics into the 2 funnels is as follows 1) enables actionable intelligence to understand drop-off in viewers 2) distinguishes metrics that are unique to video start-up vs video playback and enables easier diagnostics, and 3) creates an understanding of the inherent trade-offs associated with video playback for optimization.

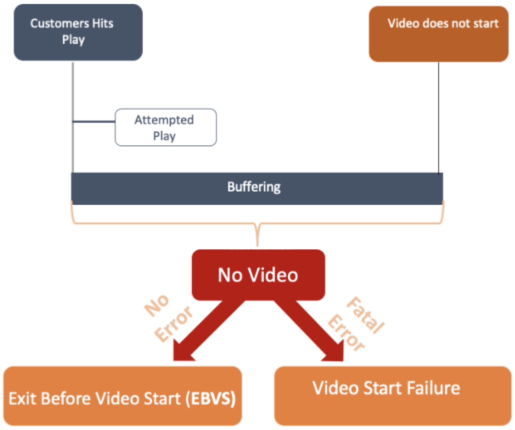
VIDEO START FAILURES (VSF)

Availability : Available
Definition: A Video Start Failure (VSF) is defined as an "*Attempted Play*" that ended in a fatal error before the first-frame of video is rendered. A fatal-error occurs whenever playback stops unexpectedly without the ability to resume and the player emits an error
Use Case: Video Start Failure metric provides an understanding of the *Attempted Plays* lost in the start-up funnel. Error codes associated with Video Start Failure help diagnose issues at the start of playback such as DRM failures, Manifest failures etc. It is useful to break this metric out separately since it captures unique sets of errors that occur during playback start.
Session Calculation: At the session level, VSF is denoted as a boolean along with the error code that caused the session to fail.
Aggregation: VSF is calculated as the number of Attempts that ended in a fatal error in that time period. In percentage format, it denotes the percentage of attempts that failed in the selected time interval.
Type: Float up to **1** decimal places
Adjustments and Sanitization: VSF does not include Attempts that failed with a fatal error due to business reasons for e.g. Geo-block errors, Exceeding number of concurrent streams etc. These errors are deemed as expected errors rather than technical delivery related errors. **Note:** Today this metric when computed may include expected errors since this is based on players choosing to report business logic errors as fatal.
Sample Query:

```
SELECT COUNT(DISTINCT streaming_segment_id) AS videoStartFailures
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
    AND content_duration = 0
    AND fatal_count > 0
```

EXITS BEFORE VIDEO START (EBVS)

Availability : Available
Definition: EBVS measures the attempts that terminated before the first frame of video was rendered—without a reported fatal error. If a fatal error is reported when the video terminates before starting, the video termination is counted as a VSF instead of EBVS.



Use Case: The most common reason for EBVS is that content is taking too long to start playing. Viewers abandon the video attempt when it takes too long to start. There are many reasons that could cause this problem and drilling down from EBVS is

an extremely efficient way for engineering teams to diagnose long start-up times.
Session Calculation: At session level, EBVS is denoted as a boolean
Aggregation: EBVS is calculated as the number of Attempts that did not join and render the first frame of video for the selected time period. In percentage, it is the percentage of attempts for which the first frame of video was never rendered and never received a fatal error in the selected time interval.
Type: Float up to 1 decimal place
Adjustments and Sanitization: EBVS for a session is calculated when we receive an explicit exit from the video player. Sometimes we may not get an explicit exit from the player. In such cases, any video that does not join within 30 seconds will be treated as a EBVS. **Note:** Viewers may leave for reasons other than long startup times, for example deciding that they clicked the wrong video or clicking on a related video. Before becoming concerned with a player's specific percentage, we must attempt to improve Time to First Frame (TTFF) and see how that impacts Exits Before Video Start.
Sample Query:

```
SELECT COUNT(DISTINCT streaming_segment_id) AS exitsBeforeVideoStart
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
    AND content_duration = 0
    AND fatal_count = 0
```

WAIT TIME BEFORE EXIT

Availability : Available
Definition: This metric is calculated as the time from when a viewer Attempts to play a video to when the video exited before the first frame of video was rendered **without a fatal error.**
Use Case: The wait time before exit metric helps understand viewer tolerance levels for waiting for the video to start.
Session Calculation: Calculated as the time from when viewer clicks Play to when the video exited without the first frame of video being rendered.
Aggregation: The metric is calculated as a distribution with x -axis being Wait time before Exit and Y-axis as the number of Attempts. Attempts with less than 1 second wait time exits can be deemed as intentional exits by customers.
Adjustments and Sanitization: None
Sample Query:

```
SELECT approx_percentile((stop_time - start_time), 0.50) AS waitTimeBeforeExit
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
    AND content_duration = 0
    AND fatal_count = 0
    AND (stop_time - start_time) >= 1000
```

TIME TO FIRST FRAME (TTFF)

Availability : Available
Definition: TTFF is the number of seconds between when the user clicks "play "or a video auto-plays to when the first frame of video is rendered. The first frame of a video can be the first frame of feature content or ad content.
Use Case: A long time to start playback causes viewers to abandon playback. This metric provides an understanding of the customer tolerance for start-up time of the video and can be used to improve it.
Session Calculation: NA
Aggregation: TTFF is calculated as an average and percentile value

- Median (50th Percentile) – Helps understand a typical experience (half are better than this number, half are worse). This is the 50th percentile of TTFF across all session for a given time interval.
- 95th Percentile – Helps understand what a poorer experience is like on a platform, while excluding outliers and frequent

enough (1 in 20 views) to always be worth attention. This is the 95th percentile of TTFF across all session for a given time interval.

PMET Calculation: As with other PMET based metrics the logic is the same but there are some important caveats to call out. The system that generates PMET data is inherently stateless, meaning it can only calculate metrics when all events that are used as input to them occur in the same batch of data (60 second window). For thinks like seeks and rebuffers, these are often short lived and this is not a problem; the outliers are just noise. For TTFF, however, this presents an interesting challenge. In general this solutions works fine here as well, but as TTFF begins to increase to 10, 20, or 30 seconds in some worse performing cohorts the odds of all the events being in the same batch decreases. This means those streams wont count their TTFF in PMET, leaving the data to be calculated as a sample from the remaining values. In extreme cases where TTFF has severely regressed this can lead to what actually looks like TTFF improvement in PMET, when in reality it's gotten so bad all of the bad ones are being discarded and we're only sampling a bias of the few remaining that did perform well. These regressions when they happen will initially show up in Sauron, but over time as it keeps getting worse eventually the metric will hit a breaking point where it looks like it has improved when in reality it has continued to regress. As such TTFF in PMET should only be used to monitor sudden shifts in magnitude, and Sauron should be referred to for long term performance analysis.

Adjustments and Sanitization: If during start-up, the video is paused by the user or by the system due to needing to enter a PIN, then this pause time is not calculated towards TTFF.

Sample Query:

```
SELECT approx_percentile(ttff, 0.50) AS timeToFirstFrame
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

REBUFFERS

Availability : Available

Definition: Rebuffering occurs when the video stalls during playback and the viewer must wait for the video to resume playing. An unexpected re-buffer occurs whenever playback stalls for a limited amount of time and it is not a result of an explicit customer action such as pause, seek, change audio track etc. During this period, a spinner generally appears on the screen. A stall during the start of playback (before the first frame is rendered) is not considered an unexpected re-buffer. This is considered initial loading time and is captured as *Time To First Frame.* Frame Drops aren't counted as re-buffers. If a customer exits playback while in a rebuffering state, a rebuffer is still counted (see: *Buffering Abandoned Plays*).

Use Case: Frequent rebuffering is a major source of poor quality of experience and often leads to customer abandonment.

Session Calculation: NA

Aggregation: NA

Adjustments and Sanitization: The following are **excluded** from being counted towards a *Rebuffers* 1) Short buffer: Any buffering that is less than 500ms, 2) Seek buffering: Any buffering that occurs after a user interaction such as pause, seek, rewind, fast-forward that is less than 30 seconds (see *Video Restart Time* metric below), 3) Ad Transition: A transition from ad to content or ad to ad that is less than 500ms is excluded, 4) Continuous play: A measure of how long it takes to transition from one feature content to the next within the same playback experience, 5) Audio track changes triggering buffering, 6) Casting from one device to another.

Sample Query:

```
SELECT SUM(rebuffer_count) AS rebuffers
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

ZERO BUFFER RATE (ZBR)

Availability : Available

Definition: ZBR is defined as the percentage of sessions that experience zero unexpected rebuffers. This metric is calculated for sessions that ended in a given interval.

Use Case: Understand and improve percentage of customers with no rebuffers. While this metric indicates the percentage of sessions that never experienced a buffer, it does not explain the magnitude of buffering for buffering impacted plays. PTSB (see below) is better suited for understanding and diagnosing issues with buffering. ZBR is best suited for long-term performance analysis of the system at daily, weekly, monthly cadences.
Session Calculation: NA
Aggregation: Calculated as percentage of sessions with zero rebuffers to total ended sessions for the selected interval.
PMET Calculation: It is important to note that while the logic of sessions without rebuffers / total sessions is the same in PMET as in Sauron and reports, the data source is very different. We are used to an aggregated “Lifecycle” view where 1 session = 1 row, and in this world that single row either had a rebuffer or it did not. This is what gives us our normal business ZBR baseline of ~95%, which means over all the minutes of a session on average 95% of our customer never see a rebuffer. PMET data, however, is inherently “Interval” based, similar to the “Interval” view in Sauron. In this data set every session has a new entry every minute, and in each of those minutes it can rebuffer or not. Given our ASD of 18 minutes, this means there are roughly 18x more records in this data set than the Lifecycle one. More often than not, however, even when there is a rebuffer there is only a single one in a single minute. Using some example math then, if we had 100 sessions and 5 rebuffered that would be a Lifecycle ZBR of 95% ((100-5)/100 = 0.95). If we took those same sessions from an interval perspective, however, we’d do (((100 * 18)-5)/(100*18) = 0.997). This means if we have a Lifecycle 95% ZBR, we’d have an Interval based 99.7% ZBR. This is important to remember when setting alarm thresholds, as doing them on something like 90% would mean they would only go off if over 10% of all customers streaming saw a rebuffer in the exact same minute, which almost never happens, even during major events and outages.
Adjustments and Sanitization: None
Sample Query:

```
SELECT (CAST(SUM(CASE WHEN rebuffer_count = 0 THEN 1 ELSE 0 END) AS double) / CAST(COU
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

LIVE ZERO BUFFER RATE (L-ZBR)

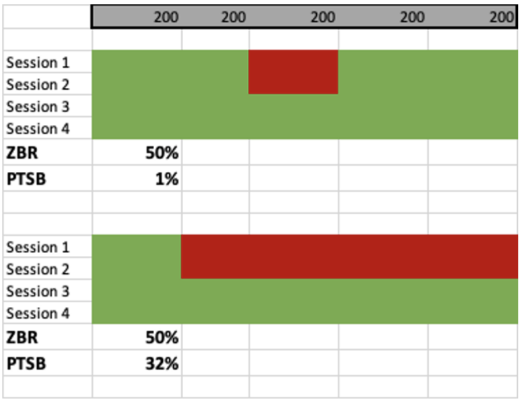
Availability : Available (in Sauron goals dashboard by request ONLY)
Definition: Live Zero Buffer Rate (L-ZBR) is the percentage of live streams that experienced zero, or a maximum of one re-buffer lasting less than 3 seconds in duration, including those from buffer abandoned plays. This metric is calculated for sessions that ended in a given interval.
Use Case: Understand and improve percentage of Live Event sessions with minimal rebuffering. L-ZBR was created as a more forgiving variant of ZBR specifically for Live Event customers for the purposes of setting Live goals and tracking against them.
Session Calculation: n/a
Aggregation: Calculated as percentage of sessions with zero rebuffers, or a single rebuffer that lasted less than three seconds, as compared to ended sessions for the selected interval.
Type: Float up to 2 decimal places
Adjustments and Sanitization: TBD
Sample Query:

```
WITH pretable AS (
SELECT streaming_segment_id,
 case when rebuffer_count=1 and rebuffer_duration<3000 then 0 else rebuffer_count end a
FROM restricted.fact_streaming_segment
WHERE publish_timestamp BETWEEN timestamp '2023-07-25 23:00:00' AND timestamp '2023-07

select
(CAST(count(distinct CASE WHEN rebuffer_new_count = 0 THEN streaming_segment_id ELSE nu
from pretable
```

PERCENTAGE TIME SPENT BUFFERING (PTSB)

Availability : Available
Definition: PTSB is defined as the percentage of time spent buffering compared to **Video viewing time**. Video viewing time is the sum of all the time spent watching the video inclusive of buffering. PTSB does not include the initial buffering before the first frame of video is rendered. This initial buffering is calculated as **TTFF**.
Use Case: This metric is used to understand the **magnitude of buffering** experienced by the customer. Unlike ZBR this metric is used to measure the level of impact of buffering on customer. In the example below, both cases have a ZBR of 50% however case 2 indicates a serious problem that needs investigation.



	200	200	200	200	200
Session 1	🟩	🟩	🟥	🟩	🟩
Session 2	🟩	🟩	🟥	🟩	🟩
Session 3	🟩	🟩	🟩	🟩	🟩
Session 4	🟩	🟩	🟩	🟩	🟩
ZBR	**50%**				
PTSB	**1%**				
Session 1	🟩	🟥	🟥	🟥	🟥
Session 2	🟩	🟥	🟥	🟥	🟥
Session 3	🟩	🟩	🟩	🟩	🟩
Session 4	🟩	🟩	🟩	🟩	🟩
ZBR	**50%**				
PTSB	**32%**				

Higher buffering leads to viewer abandonment and is a leading indicator. Use this metric to diagnose degradation in system performance and to diagnose issues.

- P50 values helps understand a typical experience (half are better than this number, half are worse).
- 95th Percentile – Helps understand what a poorer experience is while excluding outliers.

Session Calculation: Calculated as the percentage of time spent buffering against total *Video viewing time f*or that session.
Time Spent buffering (TSB) is the time in seconds spent buffering for a session. This is a lifetime value when the session has ended.
Aggregation: Calculated as the percentage of total time spent buffering to the total *Video viewing time* for all the sessions in the selected time interval. This metric is aggregated as an interval metric. A **distribution of TSB** on x-axis to Video viewing time on Y axis provides an understanding of customer tolerance and drop of points once buffering exceeds a certain threshold. This threshold can be different based on geography, content types, device and connection types and helps inform optimization techniques.
Adjustments and Sanitization: See *Rebuffers* metric sanitization and adjustments
Sample Query:

```
-- PTSB as Defined in Metrics Dictionary
SELECT (CAST(SUM(rebuffer_duration) AS double) / CAST(SUM(rebuffer_duration + content_
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'

/*
-- P50 PTSB for Streams with Rebuffers
SELECT (approx_percentile((CAST(rebuffer_duration AS double) / CAST((rebuffer_duratior
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
    AND rebuffer_duration > 0
*/
```

BUFFERING ABANDONED PLAYS (BAP)

Availability : Available
Definition: A Buffering Abandoned Play (BAP) occurs when a session ends with the last known player state is 'buffering'. The metric is defined as the percentage of all active sessions that ended with a BAP. Where active sessions refers to any session that started with a Successful Play.
Use Case: Understand viewer tolerance to buffering and percentage of successful plays that were abandoned due to buffering.
Session Calculation: At session level, this is denoted by a boolean.
Aggregation: BAP is calculated as the sum of sessions that ended in a BAP divided by the number of sessions that were active at some point during the window. Active indicates that a session started with a Successful Play
Adjustments and Sanitization: Since the player terminates sessions with continuous buffering of 45 seconds as "unrecoverable", this metric captures sessions that exited between 1 second and 45 seconds of buffering, including both customer-initiated exits and player-automated terminations.
Sample Queries:

```
-- interval
SELECT
    (CAST(SUM(CASE WHEN bufferingAbandonedPlayCount  > 0 THEN 1 ELSE 0 END) AS double)
FROM default.interval_streaming_segment
WHERE publish_timestamp = timestamp '2021-10-21 00:00:00'

-- lifecycle
/*
SELECT
    (CAST(SUM(CASE WHEN abandon_count > 0 THEN 1 ELSE 0 END) AS double) / CAST(COUNT(I
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-10-21 00:00:00'
    AND content_duration > 0
*/
```

BUFFERED SESSIONS RATE (BSR)

Availability: **Not Yet Available**
Definition: This metric is defined as the percentage of successful plays that buffered in a given time interval.
Use Case: Alert on increase in plays with buffers.
Session Calculation NA
Aggregation: Percentage of successful plays that had a rebuffer greater than 500ms in the chosen interval
Adjustments and Sanitization: Any session that has a buffer less than 500ms does not count towards this metric
Sample Query:

VIDEO PLAYBACK FAILURES (VPF)

Availability : Available
Definition: A VPF occurs when video playback reports a fatal error. Playback failure occurs whenever playback stops without the ability to resume and the player emits a fatal error. Three conditions must be met for a play to be deemed as a VPF 1) A fatal error must be received from the player, 2) The session must have successfully started playback, and 3) Playtime after the fatal error must be less than or equal to the buffer length. We observe from our data that in some cases plays with fatal errors continue to play. The condition of tracking whether playtime after the fatal is greater than buffer length ensures that any playtime seen after a fatal error is not because of play from the local buffer, otherwise it is not counted in VPF.
Use Case: Understand plays that ended unexpectedly due to fatal errors. A breakdown of VPF plays by error codes helps diagnose the top fatal errors impacting the playback experience.

Session Calculation: At session level, a VPF is a boolean value of True/False with the associated error code.
Aggregation: VPF is calculated as the sum/percentage of plays that ended in a fatal error to the total number of active sessions in a given minute.
Adjustments and Sanitization: Any session that has a continuous buffer of 30 seconds forces an error for customers. This is to ensure that customers do not experience infinite spinners in the service. Such plays count towards VPF metric. This metric does not count playback fatals caused due to business reasons such as geo-block errors.**Note:** Today this metric when computed may include expected errors since this is based on player reporting or not reporting business logic errors as fatal.
Sample Query:

```
SELECT (CAST(COUNT(DISTINCT case when fatal_count > 0 and content_duration > 0 then st
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

FATAL ERROR RATE (FER)

Availability : Available
Definition: This metric is defined as all the sessions that ended in fatal error. A fatal error occurs whenever playback stops unexpectedly without the ability to resume and the player emits an error. In other words, this metric is the sum of *Attempted Plays* that ended in VSF and Successful *Plays* that ended in a VPF.
Use Case: Understand the total impact of fatal errors on the service. A breakdown of Total Error Rate by error codes helps diagnose the top fatal errors impacting the playback experience.
Session Calculation: At session level, this is indicated as a VSF or a VPF with a boolean value of True/False with the associated error code.
Aggregation: FER is calculated as the percentage of all sessions that reported a fatal in a given minute.
Adjustments and Sanitization: Any session that has a continuous buffer of 30 seconds forces an error for customers. This is to ensure that customers do not experience infinite spinners in the service. Such plays count towards VPF metric. This metric does not include expected fatal errors such as geo-block errors, exceeding concurrents session limit etc. **Note:** Today this metric when computed may include expected errors since it is based on players reporting or not reporting business logic errors as fatal.
Sample Query:

```
SELECT (CAST(SUM(fatal_count) AS double) / CAST(COUNT(DISTINCT streaming_segment_id)
FROM default.interval_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

ZERO ERROR RATE (ZER)

Availability : Available
Definition: This metric is defined as sessions that had no fatal errors. This metric is calculated for sessions that ended in a given interval.
Use Case: This is a success metric and indicated how well the system is performing with regards to fatal error rates. A drop in this metric indicates increase in fatal errors in the service.
Session Calculation : NA
Aggregation: ZER is calculated as the sum/percentage of all the sessions that had no fatal error to the total number of ended sessions in the selected time interval.
PMET Calculation: It is important to note that while the logic of sessions without fatal errors/ total sessions is the same in PMET as in Sauron and reports, the data source is very different. We are used to an aggregated “Lifecycle” view where 1 session = 1 row, and in this world that single row either had a rebuffer or it did not. This is what gives us our normal business ZER baseline of ~99%, which means over all the minutes of a session on average 99% of our customer never see a fatal error. PMET data, however, is inherently “Interval” based, similar to the “Interval” view in Sauron. In this data set every session has a new entry every minute, and in each of those minutes it can rebuffer or not. Given our ASD of 18 minutes, this means there are roughly 18x more records in this data set than the Lifecycle one. More often than not, however, even when there is a

rebuffer there is only a single one in a single minute. Using some example math then, if we had 100 sessions and 1 fatal errored that would be a Lifecycle ZER of 99% ((100-1)/100 = 0.99). If we took those same sessions from an interval perspective, however, we'd do (((100 * 18)-1)/(100*18) = 0.999). This means if we have a Lifecycle 99% ZER, we'd have an Interval based 99.9% ZER. This is important to remember when setting alarm thresholds, as doing them on something like 98% would mean they would only go off if over 2% of all customers streaming saw a fatal error in the exact same minute, which almost never happens, even during major events and outages.

Adjustments and Sanitization: Any session that has a continuous buffer of 30 seconds forces an error for customers. This is to ensure that customers do not experience infinite spinners in the service. Such plays count towards VPF metric.

Sample Query:

```
SELECT (CAST(SUM(CASE WHEN fatal_count = 0 THEN 1 ELSE 0 END) AS double) / CAST(COUNT
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

ZERO INTERRUPTION RATE (ZIR)

Availability : Available

Definition: This metric is defined as sessions that had no fatal errors or session that had no buffering, in other words an interruption-free experience. This metric is calculated for sessions that ended in a given interval.

Use Case: This is a success metric and indicates how well the system is performing with regards to delivering an interruption-free service. Use this metric to understand and improve long-term performance of the service.

Session Calculation : NA

Aggregation: ZIR is calculated as the total number of streams that never experienced a rebuffer OR an error over the total count of streams that ended in a time interval.

Adjustments and Sanitization: Any session that has a continuous buffer of 30 seconds forces an error for customers. This is to ensure that customers do not experience infinite spinners in the service. Such plays count towards VPF metric.

Sample Query:

```
SELECT (CAST(SUM(CASE WHEN fatal_count = 0 AND rebuffer_count = 0 THEN 1 ELSE 0 END) A
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

PERCENTAGE HIGH DEFINITION DELIVERED (PHD)

Availability : Available

Definition: PHD represents the total duration of fragments delivered in HD quality (720p or 1080p) for HD eligible streams/devices divided by the total duration of fragments delivered at all quality levels (240p, 360p, 480p, 540p, 720p, 1080p, etc) for HD eligible streams/devices.

Use Case: Understand what percentage of viewing duration was spent in watching content at HD resolution. Improve player heuristics and encoding profiles to deliver higher percentage of stream duration in HD.

Session Calculation: At session level, for HD eligible streams, it is calculated as the total duration of fragments delivered in HD quality divided by the total duration of fragments delivered at all quality levels for that session.

Aggregation: The total duration of fragments delivered in HD quality (720p or 1080p) for HD eligible streams/devices divided by the total duration of fragments delivered at all quality levels (240p, 360p, 480p, 540p, 720p, 1080p, etc) for HD eligible streams/devices.

Adjustments and Sanitization: HD is calculated using the fragment reported resolution from players. The metric measures the fragment manifest value of the downloaded fragments and not the fragments that are played. This may result in this metric being skewed when the heuristics discards any fragments in cache in favor of a different fragment.

Sample Query:

```
SELECT
    stream_type,
    COUNT(*) AS streams,
    (
        CAST(SUM(CASE WHEN video_duration_eligible_4k IS NULL AND video_duration_eligi
        CAST(SUM(CASE WHEN video_duration_eligible_4k IS NULL AND video_duration_eligi
    ) * 100 AS PHD
FROM default.fact_streaming_segment
WHERE publish_timestamp BETWEEN timestamp '2022-07-13 00:00:00' AND timestamp '2022-07
GROUP BY 1
```

PERCENT ULTRA HIGH DEFINITION DELIVERED (PUHD)

Availability : Available
Definition: This metric is defined as the percentage of total duration of fragments delivered in UHD for UHD streams/devices to the total duration of fragments delivered at all quality levels.
Use Case: Understand what percentage of viewing duration was spent in watching content at HD resolution. Improve player heuristics and encoding profiles to deliver higher percentage of stream duration in HD.
Session Calculation: At session level, for UHD eligible streams, it is calculated as the total duration of fragments delivered in UHD quality divided by the total duration of fragments delivered at all quality levels for that session.
Aggregation: The total duration of fragments delivered in UHD quality (4K) for UHD eligible streams/devices divided by the total duration of fragments delivered at all quality levels (240p, 360p, 480p, 540p, 720, 1080p, etc) for HD eligible streams/devices.
Adjustments and Sanitization: UHD is calculated using the fragment reported resolution from players.The metric measures the fragment manifest value of the downloaded fragments and not the fragments that are played. This may result in this metric being skewed when the heuristics discards any fragments in cache in favor of a different fragment.
Sample Query:

```
SELECT
    stream_type,
    COUNT(*) AS streams,
    (
        CAST(SUM(LEAST(COALESCE(video_duration_4k,0),21600000)) AS double) /
        CAST(SUM(CASE WHEN video_duration_eligible_4k IS NOT NULL THEN LEAST(COALESCE
    ) * 100 AS PUHD
FROM default.fact_streaming_segment
WHERE publish_timestamp BETWEEN timestamp '2022-07-13 00:00:00' AND timestamp '2022-07
GROUP BY 1
```

PERCENT MAX RESOLUTION (PMR)

Availability : Available
Definition: This metric is defined as the percentage of total duration of fragments delivered at the maximum resolution available to the individual streams/devices to the total duration of fragments delivered at all quality levels.
Use Case: Understand what percentage of viewing duration was spent in watching content at maximum available resolution, normalized across all content. Improve player heuristics and encoding profiles to deliver higher percentage of stream duration at maximum available resolution.
Session Calculation: At session level it is calculated as the total duration of fragments delivered at maximum resolution for the stream divided by the total duration of fragments delivered at all quality levels for that session.
Aggregation: The total duration of fragments delivered at maximum available stream resolution divided by the total duration of fragments delivered at all quality levels (240p, 360p, 480p, 540p, 720, 1080p, etc).
Adjustments and Sanitization: PMR is calculated using the fragment reported resolution from players. The metric measures the fragment manifest value of the downloaded fragments and not the fragments that are played. This may result in this metric being skewed when the heuristics discards any fragments in cache in favor of a different fragment.
Sample Query:

```
SELECT
    (
        CAST(SUM(LEAST(COALESCE((
              CASE
                WHEN resolution = '480p' THEN video_duration_sd
                WHEN resolution = '720p' THEN video_duration_720p
                WHEN resolution = '1080p' THEN video_duration_1080p
                WHEN resolution = '2160p' THEN video_duration_4k
            END
        ),0),21600000)) AS double)  /
        CAST(SUM(LEAST(COALESCE(video_duration_total,0),21600000)) AS double)
     ) * 100 AS percentageMaxResolution
FROM default.fact_streaming_segment
WHERE publish_timestamp BETWEEN timestamp '2022-07-13 00:00:00' AND timestamp '2022-07
```

AVERAGE BIT-RATE (ABR)

Availability : Available
Definition: Average Bit-rate is the time weighted average of the indicated bitrates that a viewer experiences during a video stream. The weighted average is calculated from the amount of time spent at each bitrate while a video is played. The bitrate value is the indicated bitrate from the video manifest for the rendition that is used for each segment of playback.
For **example**, if during a view lasting 3 minutes a video plays for 1 minute at 1Mbps and 2 minutes at 2Mpbs, the Weighted Average Bitrate would be: [(1Mbps * 1min) + (2Mbps * 2min)] / 3min = 1.67Mbps.
Use Case: Measure and optimize the visual quality of the videos that viewers experience
Session Calculation: Calculated as the weighted average of time spent at each bit-rate divided by total duration spent at all bit-rates for that session.
Aggregation: Calculated as the weighted average of time spent at each bit-rate across all sessions divided by total duration spent at all bit-rates for all sessions for selected time interval.
Adjustments and Sanitization: The fragment bitrate used is the bitrate value for the fragment as represented in the manifest and not the actual bitrate of the fragment itself. The metric measures the fragment manifest value of the downloaded fragments and not the fragments that are played. This may result in this metric being skewed when the heuristics discards any fragments in cache in favor of a different fragment.
Sample Query:

```
SELECT (CAST(SUM(bitrate_avg * 0.000001 * content_duration) AS double) / CAST(SUM(cont
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

AVERAGE SHIFTS PER MINUTE (ASM)

Availability : Available (via query only)
Definition: Average Shifts per Minute (ASM) is defined as the average number of combined upshifts and downshifts in bitrate, per normalized minute of video fragment duration downloaded . A shift is any “just noticeable difference” in bitrate, which is synonymous which a change in rung of the bitrate ladder. The reason we used “downloaded video duration” instead of session duration is that time normalization is needed to compare sessions/cohorts of different durations, given that shifts can naturally increase as session duration increases.
Use Case: Quantify and compare the impact of perceivable bitrate shifts customers experience over a minute of video and make tradeoff decisions between optimizing quality vs limiting perceivable changes in bitrate given different network conditions. This metric was designed for video heuristics usage and is not meant for taking goals against directly as it could drive the wrong behavior.
Session Calculation: Calculated as the per-minute average of combined upshifts and downshifts over the duration of video downloaded.
Aggregation: Calculated as the average of combined upshifts and downshifts across all sessions divided by total duration (in

minutes) of video downloaded for selected sessions.
Type: Float up to 1 decimal place
Sample Query: coming soon

PERCENTAGE BITRATE CHANGE PER SHIFT (PBCS)

Availability : Available (via query only)
Definition: Percentage Bitrate Change per Shift (PBCS) is defined as the per-shift average difference in bitrate for all upshifts and downshifts. A shift is any “just noticeable difference” in bitrate, which is synonymous which a change in rung of the bitrate ladder. A shift can consist of a change of a single rung, or multiple rungs.
Use Case: Quantify and compare the magnitude of perceivable bitrate shifts customers experience the viewing experience. Because shifts can cross multiple ladder rungs not all shifts are created equal. This helps us put the number of bitrate shifts in context with how jarring those shifts are. This metric was designed for video heuristics usage and is not meant for taking goals against directly as it could drive the wrong behavior.
Session Calculation: Calculated as the weighted average of rungs crossed for all upshifts and downshifts over the course of a session.
Aggregation: Calculated as the weighted average of rungs crossed for all upshifts and downshifts across selected sessions.
Type: Float up to 1 decimal place
Sample Query: coming soon

AVERAGE BANDWIDTH (ABW)

Availability : Available
Definition: Average bandwidth is the time weighted average of the downloaded bytes and vs time to acquire them during a video stream. The weighted average is calculated from the total amount of bytes acquired over the total time spent acquiring bytes.
Use Case: Measure and optimize ability to deliver content to the customer.
Session Calculation: Calculated as the weighted average of total bytes acquired over the total time spent downloading.
Aggregation: Calculated as the weighted average of total bytes acquired over the total time spent downloading.
Adjustments and Sanitization: Average bandwidth accuracy can be impacted by two major things; 1) downloading multiple fragments in parallel can lead to under reporting, and 2) missing acquisition/bitstream events can cause the metric to effectively be “sampled” to just the subset of streams that do report them (in some extreme examples this sampling can be on less than 1% of data).
Sample Query:

```
SELECT (CAST(SUM((bandwidth_avg/125) * content_duration) AS double) / CAST(SUM(content
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2022-04-26 00:00:00'
```

VIDEO RESTART TIME (VRT)

Availability: **Not Yet Available**
Definition: Video Restart Time is the number of seconds after user-initiated interaction until video begins playing. Customer interaction occurs when a viewer scrubs the play bar, fast forwards, or rewinds the video, pauses and resumes the video or changes audio. In these cases, viewers expect some interruption in response to their action but expect it to be within reasonable bounds (see adjustments).
Use Case: Use the VRT metric to monitor unnecessary delays in video access after user-initiated actions. These delays often lead to session abandonment. A high restart time indicates issues with trick mode delivery or player heuristics when playhead

changes.
Session Calculation: Calculated as the sum of total time in seconds of buffering that exceeds 30 seconds after a customer-initiated action for the session.
Aggregation: Calculated as a percentile value

- Median (50th Percentile) – Helps understand a typical experience (half are better than this number, half are worse). This is the 50th percentile of VRT across all session for a given time interval.
- 95th Percentile – Helps understand the experience for 95% of customers with 5% of customers of having a poorer experience. This is the 95th percentile of VRT across all session for a given time interval.

Adjustments and Sanitization: If a playback does not resume within 30 seconds of a user-initiated action then the buffering time is counted towards Video Restart time. In the case of multiple seek interaction such as the continuous press of the forward button, this metric is measured as the time between the last seek event and time taken to resume playback. Sessions that remain in the pause state for more than 1 hour with no additional events sent by the player are automatically ended by the system. In this case, the pause may be user-initiated or forced pause such as app being backgrounded when a phone call is received or when a user browses X-ray while watching content causing the content to pause.
Sample Query:

TIME BEHIND LIVE (TBL)

Availability: **Not Yet Available**
Definition: Time Behind Live measures the delay in seconds between the live signal and the live playhead delivered for viewers. This metric determines how far behind playback is from the live action. This metric is relevant for live streams only and is especially important for sports where a large delay could lead to spoilers for the customer experience. Today, for live feeds, we cannot measure time behind live autonomously and we use a proxy in lieu of manual measurement. The proxy is the time difference between what the customer sees on screen to the packager output. We manually measure the latency between the packager output to the incoming feed or camera time and then add to the proxy measurement to provide the time behind live required for external reporting.
Use Case: Use this metric to understand delay behind live for different players and protocols such as Sye vs DASH and to diagnose issues contributing to delay.
Session Calculation: Time in seconds from packager output to customer playhead time.
Aggregation: This is the average time in seconds for time behind live values for all sessions in a given interval.
Adjustments and Sanitization: None
Sample Query:

VIDEO BUFFER FULLNESS

Availability : Available
Definition: The percentage of video content buffer that is available in milliseconds to the total device buffer capacity in milliseconds. The total device device buffer capacity is immutable. This is a session level metric.
Use Case: Use this metric to understand understand causes of buffering, the percentage is expected to be zero when a session buffers. Some players may initiate a buffer before the percentage hits 0.
Session Calculation: **To Be Defined**
Aggregation: NA
Adjustments and Sanitization: None
Sample Query: None

6. Engagement Metrics

Engagement metrics allow us to understand how much our customers engage with the service. The following section outlines the key metrics

TOTAL HOURS VIEWED (THV)

Availability : Available
Definition: This metric is defined as the total playtime time in hours. Playtime only includes time during which the video was playing and excludes any time buffering. This metric is inclusive of both ad and feature playtime.
Use Case: This metric is used to understand how long customers engage with the service. The higher the number, the more engaged the audience. Using this metric to understand the launch of a new geo, new player feature to understand if it is contributing to customers using the service more. For business metrics, this metric is converted to hours.
Session Calculation: Calculated as the total playing time in seconds for a session
Aggregation: Total minutes is the sum of all playtime in minutes for all the sessions that ended in the selected interval
Adjustments and Sanitization: Calculated as the total minutes of playtime for all sessions that ended in that time interval.
Sample Query:

```
SELECT SUM(CAST(content_duration AS double)/1000/60) AS totalMinutesViewed
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

AVERAGE SESSION DURATION (ASD) A.K.A. MINUTES VIEWED PER ENDED PLAY

Availability : Available
Definition: This metric is defined as the average playtime in minutes per ended play
Use Case: This metric helps understand if the customer engagement on a per session basis is high. A low number for a segment of customers on a given player or CDN usually indicates QoE issues that need to be addressed in order to improve engagement.
Session Calculation: NA
Aggregation: Calculated as the total playtime divided by the total number of plays that ended in the selected time interval.
Adjustments and Sanitization: None
Sample Query:

```
SELECT (SUM(CAST(content_duration AS double)/1000)/60) / CAST(COUNT(DISTINCT streaming
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

HOURS PER CUSTOMER (HPC) A.K.A. HOURS VIEWED PER CUSTOMER

Availability: Available
Definition: The playtime per unique customer metric is calculated by dividing the total played hours in a selected time range by number of unique customers in that time range.
Use Case: This metric indicates the level of engagement by unique customer. While both total playtime and playtime/ended play can be influenced by a subset of customers that watch a lot, this metric captures the engagement level of all customers.
Session Calculation: NA
Aggregation: Calculated by dividing the total played minutes in a selected time range by number of unique customers in that time range.
Adjustments and Sanitization: None
Sample Query:

```
SELECT (SUM(CAST(content_duration AS double)/1000)/60/60) / CAST(COUNT(DISTINCT encryp
FROM default.fact_streaming_segment
WHERE publish_timestamp = timestamp '2021-08-01 00:00:00'
```

AVERAGE PERCENTAGE COMPLETE (APC)

Availability: **Not Yet Available**
Definition: This metric defines the percentage of content viewed against the content length. A higher percentage complete is

indicative of both engaging content and high QoE.
Use Case: Use this metric to understand which content types and titles drive higher engagement rates. Lower percentage completes across across all titles for a specific players or CDN is generally indicative of QoE issues.
Session Calculation: Calculated as ratio of the total Video viewing time to content duration
Aggregation: This is aggregated as the average percentage complete across all sessions in the selected time interval.
Adjustments and Sanitization: Calculated as the average of completion rates across all plays.
Sample Query:

OVERALL REPORT QUERY

The following report should be used as the authoritative version of all metrics defined in this document. Alterations can be made to the group by or filter clauses if needed, but no changes should be made to the metric logic.

```
SELECT
    CASE WHEN video_type IN ('Full', 'Feature', 'Trailer') THEN 'VOD' ELSE video_type
    CASE WHEN device_platform = 'blast' THEN 'LRC' WHEN device_platform = 'android' TH
    player_name AS player,
    COUNT(DISTINCT streaming_segment_id) AS sessions,
    COUNT(DISTINCT streaming_segment_id) AS attemptedPlays,
    (CAST(SUM(CASE WHEN content_duration > 0 THEN 1 ELSE 0 END) AS double) / CAST(COUN
    COUNT(DISTINCT encrypted_customer_id) AS uniqueCustomers,
    (CAST(SUM(CASE WHEN content_duration = 0 AND fatal_count > 0 THEN 1 ELSE 0 END) AS
    (CAST(SUM(CASE WHEN content_duration = 0 AND fatal_count = 0 THEN 1 ELSE 0 END) AS
    approx_percentile(CASE WHEN content_duration = 0 AND fatal_count = 0 AND (stop_tim
    approx_percentile(ttff, 0.50) AS TTFF,
    SUM(rebuffer_count) AS rebuffers,
    (CAST(SUM(CASE WHEN rebuffer_count = 0 THEN 1 ELSE 0 END) AS double) / CAST(COUNT
    (CAST(SUM(rebuffer_duration) AS double) / CAST(SUM(rebuffer_duration + content_dur
    (CAST(COUNT(DISTINCT case when fatal_count > 0 and content_duration > 0 then strea
    (CAST(SUM(CASE WHEN fatal_count = 0 THEN 1 ELSE 0 END) AS double) / CAST(COUNT(DI
    (CAST(SUM(CASE WHEN fatal_count = 0 AND rebuffer_count = 0 THEN 1 ELSE 0 END) AS d
    (CAST(SUM(CASE WHEN content_quality = 'HD' THEN LEAST(COALESCE(video_duration_720p
    (CAST(SUM(CASE WHEN content_quality = 'UHD' THEN LEAST(COALESCE(video_duration_4k,
    (CAST(SUM(LEAST(COALESCE((CASE WHEN resolution = '480p' THEN video_duration_sd WH
    (CAST(SUM(bitrate_avg * 0.000001 * content_duration) AS double) / CAST(SUM(content
    SUM(CAST(content_duration AS double)/1000/60) AS totalMinutesViewed,
    (SUM(CAST(content_duration AS double)/1000)/60) / CAST(COUNT(DISTINCT streaming_s
    (SUM(CAST(content_duration AS double)/1000)/60/60) / CAST(COUNT(DISTINCT encrypted
FROM default.fact_streaming_segment
WHERE publish_timestamp BETWEEN timestamp '2021-08-31 00:00:00' AND timestamp '2021-08
    AND COALESCE(is_test_traffic, false) = false
    AND device_platform != '[UNKNOWN]' --optional, included to reduce noise
    AND COALESCE(player_name,'') <> '' --optional, included to reduce noise
    AND is_autoplay = false
GROUP BY 1, 2, 3
HAVING COUNT(DISTINCT streaming_segment_id) >= 1000  --optional, included to reduce no
ORDER BY 1, 2, 3, 4 DESC
```

`--wiki-syntax: "xwiki"--`

--wiki-tags: "Playback Metrics; Playback Quality of Service; Playback Quality of Experience"--

--wiki-sync: "Amazon_Video/PV_Playback/Docs/Analytics/quipversion/MetricDictionary"--