Standards-based Streaming Analytics and its Visualization

Stefan Pham
Mariana Avelino
Daniel Silhavy
Truong-Sinh An
Stefan Arbanowski
stefan.pham@fokus.fraunhofer.de
mariana.avelino@fokus.fraunhofer.de
daniel.silhavy@fokus.fraunhofer.de
truong-sinh.an@fokus.fraunhofer.de
stefan.arbanowski@fokus.fraunhofer.de
Fraunhofer FOKUS
Berlin, Germany

ABSTRACT

As over-the-top (OTT) media streaming and underlying technologies have matured, streaming analytics has become more important, especially in a heterogeneous device ecosystem, where new devices or software updates can potentially cause streaming issues. In this paper we consider Server and Network Assisted DASH (SAND), Common Media Client Data (CMCD) and CTA-2066 Streaming Quality of Experience Events, Properties and Metrics as standards to enable interoperable, standard-based streaming analytics for the predominant streaming formats MPEG-DASH and HLS. We focus on the visualization aspect of streaming metrics in user interface (UI) dashboards.

CCS CONCEPTS

• **Information systems** → Multimedia streaming; • **General and reference** → Metrics.

KEYWORDS

SAND, CMCD, CTA-2066, DASH, HLS, streaming analytics

ACM Reference Format:

Stefan Pham, Mariana Avelino, Daniel Silhavy, Truong-Sinh An, and Stefan Arbanowski. 2021. Standards-based Streaming Analytics and its Visualization. In *12th ACM Multimedia Systems Conference (MMSys 21), September 28–October 1, 2021, Istanbul, Turkey.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3458305.3478438

1 INTRODUCTION

Adaptive bitrate streaming over the open Internet is a best-effort approach and can therefore lead to various streaming performance issues. Streaming metrics and events, which are collected by media players and underlying frameworks, help better understand these issues. The metrics can be difficult to make sense of, however, given the complexity of today's streaming ecosystem. Problems can occur for different reasons, ranging from incompatible device updates to unstable bandwidth. An unified, interoperable streaming analytics platform is therefore an ideal solution. It enables a comprehensive overview of streaming performance, as well as the early detection and troubleshooting of problems. A platform such as this furthermore allows different devices and streaming formats (e.g. DASH [7] and HLS [6]) to be compared, as it leverages streaming metrics standards.

In this paper, we first outline the standards used in our streaming analytics solution. Then we describe how streaming metrics are collected from players based on the example of dash.js [5]. Finally, we focus on how streaming metrics are visualized for the purposes of analysis and debugging, which is the main focus of the demo.

2 STANDARDS

This section outlines the standards that our streaming analytics solution is based on.

2.1 SAND

Server and Network Assisted DASH (SAND) is a standard specified in ISO/IEC 23009-5 [1] which defines how clients (e.g. video players on websites), servers (e.g. CDNs) and networks (e.g. the Internet and its components) should communicate with each other. Per SAND specifications, clients, servers and networks should exchange real-time status information, such as network metrics (e.g. bandwidth) and video-player metrics (e.g. buffer level). In doing so, the different network components can become aware of current network conditions and adjust their behaviour to improve bandwidth utilization and streaming experience.

2.2 CMCD

Common Media Client Data (CMCD; CTA-5004) [3] is a specification published by the Consumer Technology Association (CTA) in the Web Application Video Ecosystem (WAVE) project. The main idea behind CMCD is to enhance requests from a media player to the CDN with information that can be useful in log analysis, quality of service monitoring and content delivery optimization. For example, "bitrate, buffer and segment signaling allow CDNs to fine-tune and optimize their midgress traffic by intelligently reacting

MMSys 21, September 28–October 1, 2021, Istanbul, Turkey

to the time constraints implicit in each request" [3]. By improving the quality of service offered by CDNs, the streaming experience enjoyed by viewers is improved as well.

2.3 CTA-2066

Streaming Quality of Experience Events, Properties and Metrics (CTA-2066) [2] aims to standardize how streaming quality is measured and in this way makes comparisons of performance across media players and analytics solutions more objective. It prescribes standardized terminology and a minimum set of quality of experience events to be reported by media players, as well as a minimum set of quality of experience metrics to be computed by analytics solutions. Examples include average initial startup time, average playback bitrate, and exits before video start percentage.

3 PLAYER INTEGRATION

This section deals with the implementation of CMCD and SAND metric reporting in media players. In this paper, we focus on the implementation in dash.js [5].

3.1 dash.js

dash.js, is a free, open source MPEG-DASH player that serves as a reference client for implementing production-grade DASH players [1]. The dash.js player is written in JavaScript and relies on the Media Source Extensions (MSE) and the Encrypted Media Extensions (EME) defined by World Wide Web Consortium (W3C).

3.1.1 CMCD. dash.js began supporting CMCD reporting in its 3.0.3 version [2] and became fully CMCD specification-compliant in version 3.2.1 [3]. Initial findings of this implementation were evaluated in [4]. A simplified illustration of the CMCD workflow in dash.js is depicted in Figure 1.

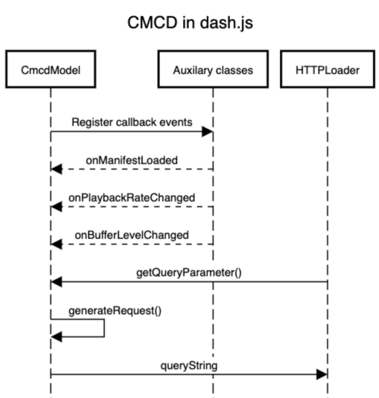

Figure 1: CMCD implementation in dash.js

The CmcdModel is the main class of the CMCD implementation in dash.js. It is responsible for collecting the CMCD-related information from different parts of the player, as well as generating the final payload to send alongside the network requests.

[1]https://github.com/Dash-Industry-Forum/dash.js
[2]https://github.com/Dash-Industry-Forum/dash.js/releases/tag/v3.0.3
[3]https://github.com/Dash-Industry-Forum/dash.js/releases/tag/v3.2.1

In the bootstrap process of the player, CmcdModel registers for certain callback events, like *manifestLoaded*, *playbackRateChanged* and *bufferLevelChanged*. The CmcdModel is a singleton class, and updates its internal state whenever it receives valid payload from one of these events. That way, the internal representation of the different parameters like playback rate and buffer level is always up-to-date. Before sending a media or manifest requests to the CDN, the HTTPLoader class will ask the CmcdModel for the CMCD-specific query parameters.

In order to build the query string, CmcdModel internally distinguishes between different types of requests, namely, MPD, media segment and init segment requests. This is mainly due to the fact that requests for media segments require additional parameters. As an example, the requests to media segments require details about the type of media segment (audio or video). Once the CMCD query string has been generated it is handed back to the HTTPLoader class and is incorporated in the final request URL.

3.1.2 SAND. While the CMCD implementation is directly embedded in dash.js, the logic for SAND metric reporting is kept in a separate library. The general interaction between dash.js and the SAND library is depicted in Figure 2.

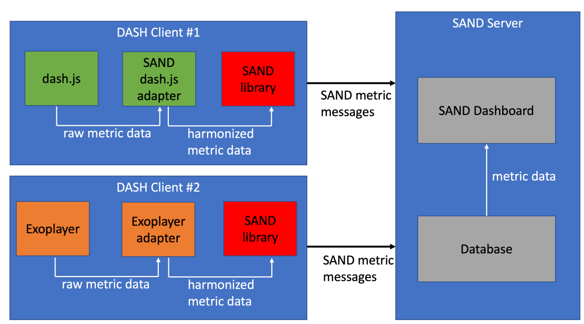

Figure 2: SAND metric reporting integration in dash.js and other player implementations.

The core logic of SAND metric reporting is encapsulated in the SAND library and is kept player agnostic. A specific adapter for each player implements player specific logic. In the case of dash.js, the SAND adapter for dash.js registers dash.js events and interacts with the dash.js API. That way the raw metric is collected and harmonized before being passed to the SAND library. The same principle applies for other players like Exoplayer and AVFoundation. The SAND library generates SAND-compliant metric messages and sends them via HTTP requests to a server (the SAND Server). This server stores the metric data in a database, which provides the required data for the SAND dashboard.

4 DEMO: STREAMING ANALYTICS VISUALIZATION

As described above, media players report SAND and CMCD metrics during streaming. These raw metrics, which include the quality of experience events specified in CTA-2066, get stored in a database, specifically, ElasticSearch [4]. The web application Grafana [5] queries the database and aggregates the metrics for visualization on dashboards. In this way, the dashboards unify CMCD and SAND data using CTA-2066 naming conventions and display most of the quality of experience metrics prescribed by CTA-2066.

Our demo conveys the value of standardized metrics for the monitoring and analysis of streaming services. Different media players (dash.js, ExoPlayer and AVFoundation) running test streams in DASH and HLS format will be made available. The audience is invited to watch these streams via their personal devices and observe how CMCD and SAND metrics get aggregated and visualized in real-time on the Reporting, Errors Overview and Session Details dashboards of our analytics solution. These three dashboards are outlined below.

4.1 Reporting Dashboard

The Reporting Dashboard is for big-picture insights. Its panels give a global overview of streaming service usage and performance for the defined time interval:

As Figure 3 shows, the dashboard reports the overall number of streams (1), clients (2), and errors (3)(4). It also plots the average number of streaming sessions and clients over time (5). Additionally, it breaks down the platforms (6), browsers (8), and manifest URLs (10) used to stream. Accompanying this data are time-series graphs of the average number of active platforms (7), browsers (9) and manifest URLs (11).

Dashboard panels are interactive, meaning that they contain even more information than initially visible. Elements can be hovered over, clicked on, zoomed into and much more. The interactivity gives the user the ability to explore the data.

4.2 Errors Overview Dashboard

The Errors Overview Dashboard, depicted in Figure 4, focuses on being a debugging and error analysis tool. As a result, it offers an overview of all errors that happened within a given time range. It is divided into five sections: General, DRM License Server Errors, DRM Client-Side Errors, XHR Errors and Media Errors.

The General section shows all errors that occurred. In this section, the total number of reported errors (1) are displayed, as well as a doughnut chart (2) and time-series (3) visualization of all errors. Two additional doughnut charts filter for serious errors (DRM, XHR and Media errors), revealing the frequency with which the errors occurred (4) and how many streaming sessions were affected by them (5). Lastly, a table lists all the streaming sessions affected by DRM, XHR and Media errors.

The remaining four sections (DRM License Server Errors, DRM Client-Side Errors, XHR Errors and Media Errors) drill even deeper into these serious errors. As these sections are structured identically, only the DRM License Server Errors section will be introduced.

As illustrated in the lower half of Figure 4, the DRM License Server Errors section visualizes the total number of DRM license server errors (1) thrown and, adjacently, the specific types of DRM license server errors (2) reported. It also traces the number of sessions affected over time (3). Two doughnut charts break down the affected devices (4) and browsers (5), and a table lists the specific errors together with the impacted platform and browser (6).

To summarize, the Errors Overview Dashboard tells the user what went wrong, as well as where and when it went wrong. It reports key stats (e.g. the total number of errors), key groups (e.g. the different types of XHR errors that were thrown), and observable trends (e.g. whether specific errors are reoccurring on specific platforms).

4.3 Session Details

The Session Details Dashboard takes a lens to a single, specific streaming session. It displays all the metrics collected from a session and puts the user in control of exploring the data. Fine-grained metrics can be interactively examined via the different panels.

As seen in Figure 5, the dashboard begins with a table containing key data about the streaming session (1). A second table contains device-specific information (2). A third table specifies the startup durations (3). Beside these tables, a timeline (4) pin-points the events that happened during the stream. Below, time-series graphs plot the average bitrate (5), buffer level (6), live-edge delay (7), and memory usage (9) of the session, as well as the errors that occurred (8). A table lists the errors that occurred, accompanied by relevant information (e.g. the MPD URL) (10). The final three tables display all the raw data reported by the client (11), all the DRM requests sent by the client (12), and all the HTTP transactions made between client and server (13).

5 CONCLUSION

Standards such as SAND, CMCD and CTA-2066 enable a common understanding of streaming metrics, from which there are many insights to be gained. Analytics solutions which leverage these standards are effective tools for streaming analysis and debugging. By collecting, processing and visualizing streaming metrics with our solution, we are capable of revealing both global and detailed insights about media streaming performance.

REFERENCES

[1] ISO/IEC 23009-5:2017. 2017. *Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 5: Server and network assisted DASH (SAND).* Standard. International Organization for Standardization, Geneva, CH.
[2] Consumer Technology Association. 2020. Streaming Quality of Experience Events, Properties and Metrics (CTA-2066). Retrieved June 10, 2021 from https://shop.cta.tech/products/streaming-quality-of-experience-events-properties-and-metrics
[3] Consumer Technology Association. 2020. Web Application Video Ecosystem - Common Media Client Data - CTA-5004. Retrieved June 7, 2021 from https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5004-final.pdf
[4] Abdelhak Bentaleb, May Lim, Mehmet N. Akcay, Ali C. Begen, and Roger Zimmermann. 2021. Common Media Client Data (CMCD): Initial Findings. In *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (Istanbul, Turkey) *(NOSSDAV '21)*. Association for Computing Machinery, New York, NY, USA, 25–33. https://doi.org/10.1145/3458306.3461444
[5] DASH Industry Forum (DASH-IF). 2021. dash.js. https://github.com/Dash-Industry-Forum/dash.js/.
[6] R. Pantos and W. May. 2017. *HTTP Live Streaming.* RFC 8216. RFC Editor.
[7] Thomas Stockhammer. 2011. Dynamic adaptive streaming over HTTP: standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems.* ACM, ACM, New York, NY, United States, 133–144.

[4]https://www.elastic.co/elasticsearch/
[5]https://grafana.com

Figure 3: Reporting Dashboard

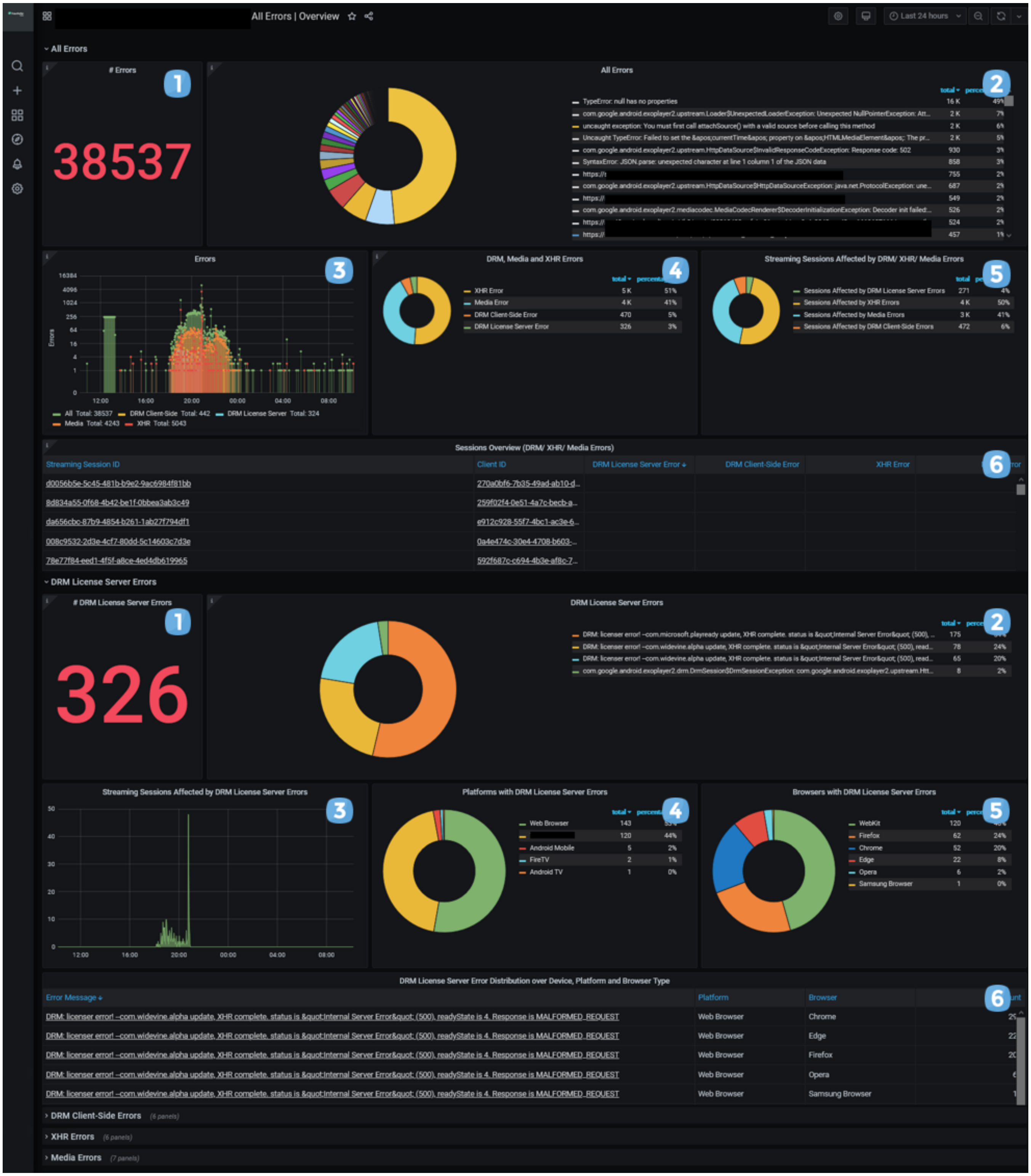

Figure 4: Errors Overview Dashboard

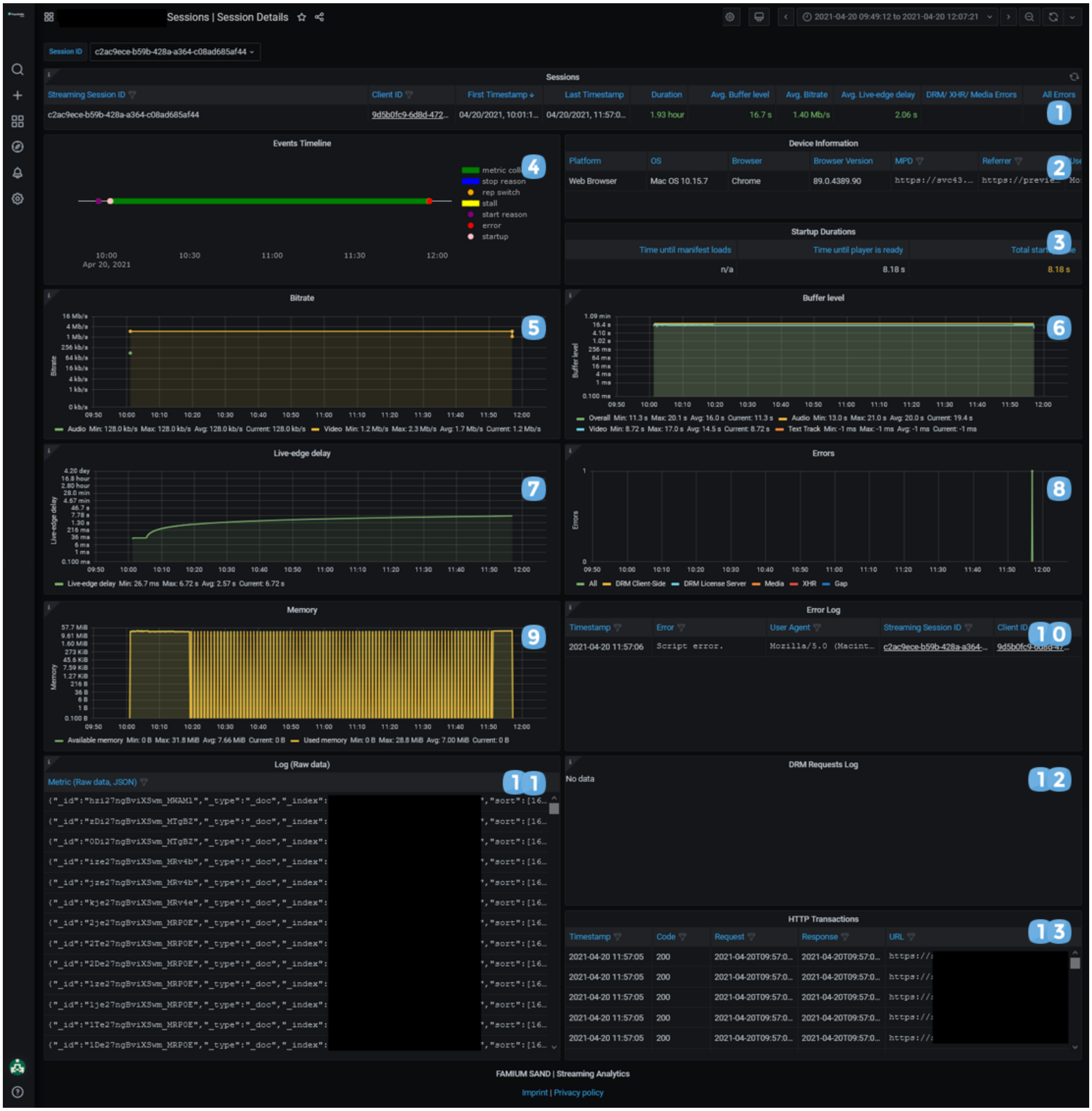

Figure 5: Session Details Dashboard