TV 3.0 Audience Measurement Management: Architecture, Lifecycle and APIs

Marcelo F. Moreno

Eduardo Barrére

CITE THIS ARTICLE

Marcelo F. Moreno and Eduardo Barrére; 2025. TV 3.0 Audience Measurement Management: Architecture, Lifecycle and APIs. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432, doi: 10.18580/setijbe.2025.5 web link: https://dx.doi.org/10.18580/setijbe.2025.5

SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING – SET IJBE V.11, 2025, Article 5 | 9 p
© 2025 SET – Brazilian Society of Television Engineering | ISSN (Print): 2446-9246 | ISSN (Online): 2446-9432

TV 3.0 Audience Measurement Management: Architecture, Lifecycle, and APIs

Marcelo F. Moreno, Associate Professor, UFJF and Eduardo Barrére, Associate Professor, UFJF

***Abstract*— Audience measurement in digital broadcast environments requires mechanisms capable of capturing granular, context-aware viewer behavior across interactive and hybrid broadcast–broadband services. Traditional household-level measurement techniques remain essential for statistical estimation but lack visibility into application-level interactions and platform dynamics. This paper presents the complete specification of the Audience Measurement Manager (AMM), a strategic middleware component of the Brazilian TV 3.0 Application-oriented Platform. AMM enables standardized, auditable, and privacy-by-design audience data collection, explicitly conditioned on viewer agreement and contextualized by service and viewer profile. The specification supports both declarative and API-driven session initiation, a formally defined session lifecycle, structured reporting with progressive delivery options, and trusted data delivery using digital signatures. Compared to existing solutions and related standards, AMM introduces middleware-enforced control, integrity assurance, and interoperability, advancing audience measurement from ad hoc application-level tracking to a deployable, system-level standard.**

***Index Terms*— Audience Measurement, Broadcast Technology, Data Collection, Software Architecture**

I. INTRODUCTION

[1]AUDIENCE measurement has long played a central role in the economic and operational organization of broadcast television, supporting content valuation, advertising planning, and regulatory oversight. Historically, this function has been fulfilled primarily through household-level methodologies, such as panel-based meters and viewer diaries [1], which provide statistically robust estimates of audience size and composition over time. While these techniques remain essential for longitudinal analysis and market comparability [2], they were conceived for largely linear, passive viewing scenarios and are increasingly strained by the complexity of contemporary digital broadcast ecosystems.

Modern broadcast environments are characterized by hybrid architectures that integrate traditional over-the-air delivery with broadband connectivity, interactive applications, and personalized user experiences. Viewers now engage with content through multiple layers of interaction, including application navigation, on-demand media selection, multimodal input commands, and service-specific features that blur the boundaries between broadcasting and internet-based platforms. In this context, audience behavior can no longer be fully described using aggregate household metrics alone. There is a growing need for more granular, context-aware telemetry capable of capturing how services and applications are actually consumed at the platform level, while remaining compatible with established measurement practices.

Existing solutions for collecting such detailed data typically rely on application-level mechanisms, including embedded scripts, tracking beacons, or proprietary software components [3][4]. Although widely deployed in smart TV platforms and hybrid broadcast standards, these approaches are often implemented in an ad hoc manner, lack interoperability, and offer limited transparency with respect to data governance and privacy controls. Moreover, by placing responsibility for data collection primarily at the application layer, they complicate auditing, weaken consistency across devices, and hinder the systematic enforcement of regulatory and ethical requirements related to personal data processing.

The Brazilian TV 3.0 system introduced a distinct opportunity to address these challenges through a middleware-centric audience measurement architecture. TV 3.0 embeds measurement capabilities directly within the Application-oriented Platform (AoP) [5], enabling standardized control over data collection, session management, and delivery, independent of individual application implementations. This architectural choice allows broadcasters to obtain operational insights into service usage and viewer interaction while preserving a clear separation of responsibilities between platform, application, and external measurement entities. Importantly, this model is designed to complement, rather than replace, existing audience measurement systems, enabling traditional measurement institutes to enrich their aggregated estimations with contextual data generated at the receiver level.

This paper presents the specification of the Audience Measurement Manager (AMM), a strategic component of the TV 3.0 AoP responsible for orchestrating audience measurement in TV 3.0 receivers. Particular emphasis is placed on the integration of privacy-by-design principles, including the enforcement of viewer's agreement with data processing, contextual scoping of data collection, and compliance with data protection regulations such as Brazil's Lei Geral de Proteção de Dados (LGPD) [6].

This work represents the first academic publication to present AMM specification as finalized during Phase 3 of the TV 3.0 Project [7]. This phase consolidated the results of a multi-year research and development effort focused on defining the application coding layer of the TV 3.0 system [8], culminating in the publication of national standards and reference specifications [5]. Earlier publications reported intermediate concepts and partial results of this effort, particularly with respect to application execution and platform architecture. By contrast, the present paper documents the finalized audience measurement architecture, operational models, session lifecycle, reporting framework, and trust mechanisms, thereby advancing the proposal from R&D and conceptual design to a standardized solution.

[1] This work was supported in part by Brazilian Ministry of Communications (MCom), SBTVD Forum and the National Research and Education Network (RNP), under TV 3.0 Project - Phase 3.

https://set.org.br/ijbe/ doi:10.18580/setijbe.2025.5 Web Link: https://dx.doi.org/10.18580/setijbe.2025.5

The remainder of this paper is organized as follows. Section II reviews traditional and digital audience measurement approaches and related standards, positioning the proposed solution within existing practice. Section III provides an architectural overview of the AMM within the TV 3.0 AoP. Section IV describes the supported models for initiating audience measurement sessions, including declarative and API-driven mechanisms. Section V details the standardized APIs for session control and operational management. Section VI presents the formal session lifecycle and finite state machine governing data collection. Section VII describes the data collection scope and the structured report schema used for audience measurement delivery. Section VIII discusses the trusted delivery mechanisms that ensure data integrity and authenticity. Finally, Section IX concludes the paper and outlines directions for future work.

II. BACKGROUND AND RELATED WORK

Audience measurement has traditionally been grounded in household-level methodologies, such as panel-based meters and diary surveys, which provide statistically robust estimates of audience size, composition, and viewing patterns over time [1]. These techniques have long served as the primary currency for advertising markets and programming decisions, enabling longitudinal comparability and market-wide inference [2]. Despite their continued relevance, such methods were conceived for predominantly linear and passive viewing scenarios and offer limited insight into the interactive and context-dependent behaviors increasingly observed in digital broadcast environments.

As television systems evolved toward software-driven, network-connected platforms, both academia and industry began exploring data-centric approaches capable of capturing large-scale, fine-grained usage information directly from receivers. Return-path data collected from set-top boxes and connected televisions has been widely discussed as a means to approximate census-level measurement and to complement traditional panels [9][10]. These approaches promise greater temporal resolution and coverage but also introduce new challenges related to data interpretation, representativeness, and governance.

More recently, content identification technologies, most notably Automatic Content Recognition (ACR) [3], have emerged as a prominent mechanism for inferring content exposure and engagement on smart TV platforms. ACR systems operate by matching locally observed audio or video fingerprints against reference databases, enabling platforms and third parties to detect what content is being presented on the screen regardless of delivery path. Empirical studies have shown that ACR-based tracking is widely deployed in commercial smart TVs and operates as a continuous telemetry mechanism with implications for transparency and user privacy [3][4][11].

Within hybrid broadcast environments, particularly those based on HbbTV [12], audience and engagement measurement largely follows the same architectural paradigm as web and streaming applications. Data collection is typically implemented through application-embedded scripts, tracking beacons, cookies, or similar client-side mechanisms [13]. This model mirrors established practices in web analytics and over-the-top streaming platforms, where each application defines its own tracking logic, session semantics, and reporting formats. While such approaches offer flexibility and rapid deployment, they also inherit shortcomings of web-based measurement ecosystems. Session boundaries are application-defined and inconsistent, continuity across application transitions is not guaranteed, and implementations vary significantly across services and receiver platforms. Furthermore, because measurement logic resides primarily at the application layer, privacy controls and transparency mechanisms depend heavily on individual application design choices, resulting in heterogeneous enforcement and limited auditability [4] [13].

Standardization efforts within ATSC 3.0 introduce a more structured but still limited approach to usage reporting. The ATSC A/333 specification [14] defines a standardized Service Usage Reporting framework, including a JSON-based Consumption Data Message (CDM) format for reporting intervals of broadcast service usage. These reports may include service identifiers, timestamps delimiting usage intervals, delivery path metadata, and optional information related to application activity. A/333 thus establishes a common data representation for what usage information may be reported by capable receivers. However, the scope of A/333 is intentionally constrained to usage reporting and does not extend to a comprehensive audience measurement session management model. The specification does not define a normative session lifecycle with explicit states such as initialization, pause, resume, or termination, nor does it mandate persistence or recovery behavior across context changes. Usage data is reported as discrete intervals rather than as elements of a managed session governed by the receiver middleware. In addition, A/333 does not incorporate explicit binding to viewer profiles, and mechanisms for enforcing privacy constraints, authenticity, or integrity protection are not mandated at the platform level.

TV 3.0 AMM as presented in this paper addresses these limitations by shifting key measurement responsibilities from applications and optional reporting mechanisms into the middleware layer of the AoP. Rather than treating audience measurement as an auxiliary or application-specific function, AMM formalizes it as a managed platform capability, with standardized mechanisms for session creation, execution, suspension, and termination. Each measurement session is explicitly contextualized by the broadcast service and the active viewer profile, enabling consistent scoping and continuity even in the presence of application transitions or hybrid service interactions.

By operating at the middleware level, AMM also enables systematic integration of governance and privacy considerations into the measurement lifecycle. Platform-level coordination allows data collection to proceed only under an appropriate viewer agreement with data processing and within clearly defined contextual boundaries, consistent with data protection regulations such as Brazil's Lei Geral de Proteção de Dados [6]. In addition, AMM introduces a standardized reporting structure and optional cryptographic protections that support traceability, integrity verification, and independent auditing of delivered audience data [15] [16]. These characteristics distinguish the TV 3.0 approach from existing techniques in both academia and industry, which often prioritize scale and granularity at the expense of uniform session semantics, interoperability, and enforceable governance. Table I summarizes relevant technical differences between HbbTV (application-centric measurement approaches), ATSC 3.0 (service usage reporting), and TV 3.0's middleware-managed audience measurement model.

TABLE
COMPARISON OF AUDIENCE MEASUREMENT APPROACHES IN HYBRID TV ECOSYSTEMS

Technical criterion	HbbTV (ETSI TS 102 796 [12], Europe)	ATSC 3.0 (A/333 [14], USA)	TV 3.0 (ABNT NBR 25608 [5], Brazil)
Implementation model	Delegated to applications; scripts, beacons, and web-style trackers	Optional service usage reporting capability defined by A/333	Integrated and standardized at the receiver middleware level
Measurement scope	Application-defined tracking events	Interval-based service usage records (CDM)	Structured audience measurement sessions
Session modeling	Not standardized; application-specific	No normative session lifecycle; usage intervals only	Standardized finite state machine governing session lifecycle
Contextual binding	No enforced binding to service or viewer profile	Binding to broadcast service; no viewer profile association	Explicit binding to broadcast service and active viewer profile
Continuity across apps/ services	Interrupted by application transitions	No standardized pause/resume or persistence semantics	Automatic pause and resume based on context changes
Privacy governance	Application-dependent; heterogeneous controls	No platform-level enforcement mechanisms specified	Platform-level mechanisms supporting privacy-by-design and lawful processing
Report structure	Proprietary and heterogeneous	Standardized JSON schema for usage data (CDM)	Standardized JSON schema (header, initial status, events), with progressive delivery
Data authenticity	Not addressed	No mandatory integrity or signature mechanisms	Optional ECDSA/SHA-256 digital signatures
Auditability	Limited; application logs and vendor--specific tools	Standardized usage format with limited traceability	Improved traceability and auditability through session-based reporting

III. TV 3.0 AMM ARCHITECTURAL OVERVIEW

The Audience Measurement Manager (AMM) is designed as a strategic middleware component of the TV 3.0 Application-oriented Platform (AoP), responsible for orchestrating audience measurement at the receiver level. Its architectural role is to decouple audience data collection from individual applications and content, shifting this responsibility to the platform itself. By doing so, AMM eliminates the need for broadcasters to embed proprietary triggers, cookies, or tracking scripts within applications, enabling a consistent and standardized measurement approach across all TV 3.0 receivers. Figure 1 highlights AMM within TV 3.0 AoP architecture, where it is positioned as a component of TV 3.0 Ginga Common Core [5].

From an architectural perspective, AMM operates as an autonomous AoP service that is explicitly activated by broadcaster intent. Upon receiving a valid request, the AoP, through the AMM, assumes responsibility for collecting audience-related data, aggregating it into a standardized report, and delivering it securely to broadcaster-defined endpoints. Neither Bootstrap nor Broadcaster Applications directly participate in the data collection process. Instead, they act solely as initiators or controllers of measurement sessions, while the middleware ensures uniform behavior, persistence, and enforcement of platform policies.

A central design principle of AMM is the integration of privacy-by-design mechanisms at the architectural level. Audience measurement is only permitted when the viewer has previously provided agreement for data processing through the platform's privacy management framework [17], in accordance with data protection regulations such as Brazil's Lei Geral de Proteção de Dados (LGPD). In particular, the use of AMM APIs is conditioned on viewer agreement for service usage analytics purposes, as defined in the platform's privacy and regulatory reference data [5]. This agreement is verified by the AoP before anysession is established, ensuring that data collection cannot be initiated implicitly or without viewer awareness.

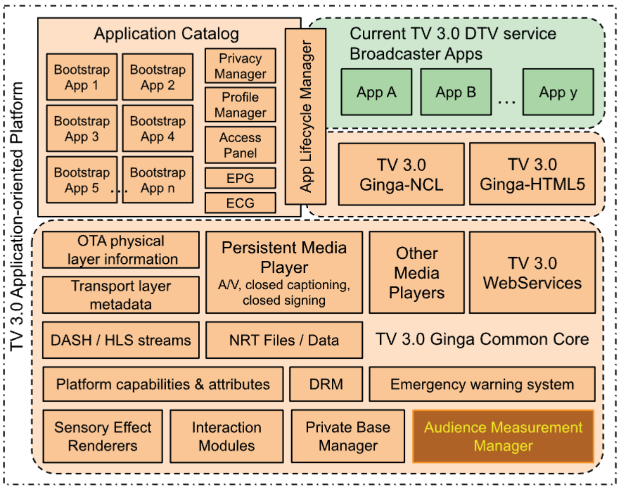


Fig. 1. Audience Measurement Manager (AMM) in the TV 3.0 Application-oriented Platform (AoP) architecture. Adapted from ABNT NBR 25608 [5]

To enforce strict contextual boundaries, AMM binds each audience measurement session to a well-defined collection context identified by the pair (service-globalServiceID, viewer-id). This context captures both the broadcast service that initiated the session and the active viewer profile that authorized data collection. All collected data is stored, processed, and transmitted exclusively within this context, preventing unintended aggregation across services or viewer profiles. Any change to either element, such as service zapping, a viewer profile switch, or repurposing of the receiver, constitutes a disruption in the collection context and triggers an automatic response by the middleware.

Audience measurement within AMM is organized around data collection sessions, which provide a structured framework for lifecycle management and auditability. Sessions can be initiated in two complementary ways: declaratively, through metadata defined in the Bootstrap

Application Manifest, or procedurally, via standardized TV 3.0 WebServices APIs exposed by the AoP (see Section IV). Regardless of the initiation mechanism, the resulting session is managed uniformly by AMM and assigned a unique identifier that enables monitoring, control, and correlation of collected data with its originating context.

Once established, a session remains active across application transitions within the same broadcast service, ensuring continuity of data collection without requiring application-level coordination. If a disruption in the collection context occurs, the middleware automatically transitions the session into a paused state. When the original context is restored, the session resumes transparently, preserving continuity as well as contextual correctness.

Each session is associated with a predefined validity period, after which data collection concludes, and the final report is prepared for delivery. AMM also supports progressive delivery mechanisms, allowing partial reports to be transmitted at defined checkpoints during an ongoing session. This capability enables timely access to audience data without waiting for session completion, while still preserving the integrity and traceability of the overall report.

From a data governance perspective, the AMM architecture clearly delineates roles and responsibilities. Audience measurement data collected through AMM is intended for the exclusive use of broadcasters, who act as independent data controllers and retain full authority over subsequent data processing decisions. This approach aligns the technical architecture with regulatory expectations and supports accountability across the measurement ecosystem.

Finally, the variety of possible session states, context changes, and error conditions encountered during audience measurement are formally modeled as a finite state machine (see Section VI). This model defines the permissible transitions throughout a session's lifecycle and provides a foundation for predictable behavior, recoverability, and external control. Using the session identifier, Broadcaster Applications may query or request modifications to the current session state at [18] through standardized APIs, while the middleware ensures that all transitions comply with contextual, privacy, and resource constraints.

IV. SESSION INITIATION MODELS

TV 3.0 AMM supports two complementary models for initiating audience measurement sessions: a declarative model, based on metadata in the Bootstrap Application Manifest (BAM), and a procedural model, based on standardized TV 3.0 WebServices APIs. Both models expose a common set of configuration parameters and result in the creation of a managed session governed by the same lifecycle and state machine. The distinction lies in how and when the session is requested, allowing broadcasters to choose the most appropriate approach depending on operational requirements.

A. Declarative Initiation via Bootstrap Application Manifest

In the declarative model, audience measurement is initiated automatically when a Bootstrap Application is launched, based on configuration metadata included in the Bootstrap Application Manifest (BAM) [5], which is carried in the BAM table (BAMT) as defined by the TV 3.0 transport layer specification [18]. The audienceMeasurementSession element allows broadcasters to declaratively associate an audience measurement campaign with the lifecycle of the Bootstrap Application, thereby establishing a collection context that may be reused by subsequent Broadcaster Applications and associated content, without requiring any explicit runtime interaction with APIs. This approach is particularly suitable for long-running or service-wide measurement campaigns that should start as soon as the broadcast service becomes active on the receiver. Listing 1 illustrates a BAMT instance in which a BAM element declares an audience measurement session.

The campaignId attribute provides a broadcaster-defined identifier for the measurement campaign, enabling aggregation, correlation, and analysis of audience data across receivers and reporting intervals. This identifier is preserved throughout the session lifecycle and included in all delivered reports, serving as a stable reference for downstream processing and analytics systems.

Delivery configuration is defined through the deliveryURLs attribute, which specifies one or more endpoints to which audience measurement reports are transmitted. These endpoints are fixed at session creation time and are used by AMM to deliver either final or partial reports using secure, standardized HTTPS mechanisms. By defining delivery targets declaratively, broadcasters ensure that reporting behavior remains consistent and independent of application-level logic.

Temporal characteristics of the session are controlled through the sessionPeriod attribute, optionally combined with scheduledStart. Together, these attributes define the validity window of the measurement session, either by specifying a fixed duration or a scheduled finish time, and optionally delaying session start until a predefined instant. This allows broadcasters to align measurement precisely with editorial schedules, service availability, or regulatory constraints, while ensuring that data collection remains bounded and predictable.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<BAMT xmlns=
  "tag:sbtvd.org.br,2025:XMLSchemas/TV30/Delivery/BAMT/1.0/">
  <BAM globalServiceId=
        "urn:br:com:broadcaster:tv30:service:main"
     appVersion="1" appName="Broadcaster" >
   <!-- Other BAM elements may appear here -->
   <audienceMeasurementSession
    campaignId="fifa-worldcup26-BRA-MAR"
    deliveryURLs=
      "https://tv30amm.broadcaster.com.br/collect
       https://api.mediaresearch.com/tv30/YnJvYWQ6bWFpbg"
    sessionPeriod="PT2H"
    scheduledStart="2026-06-13T22:00:00Z"
    progressiveDelivery="PT10M" filterConfig="all"
    signatureRetrievalURL=
      "https://api.mediaresearch.com/sig/YnJvYWQ6bWFpbg"/>
  </BAM>
</BAMT>
```

List. 1. Example of a BAMT instance with declarative audience measurement session configuration.

The optional filterConfig attribute allows broadcasters to constrain the scope of collected data by specifying which categories of audience measurement information should be included in the report. When set to the value "all", all available data categories defined by the platform are collected; otherwise, only the explicitly listed categories are included. This mechanism supports data minimization strategies and enables broadcasters to tailor reports to specific analytical needs.

Trusted delivery of audience measurement reports may be enabled through the optional signatureRetrievalURL attribute. When present, this attribute specifies a service from which the platform retrieves digital credentials used to digitally sign the generated reports prior to delivery. This allows broadcasters and authorized third parties to verify the integrity and origin of the collected data without relying on application-level security mechanisms. The use of trusted delivery is optional at session creation time and applies uniformly to both final and progressively delivered reports; its cryptographic aspects are detailed in Section VIII.

Finally, the progressiveDelivery attribute enables interim delivery of partial reports during the session's execution. When configured with a duration, partial reports are delivered periodically at the specified interval; alternatively, delivery may be triggered asynchronously upon changes in the session's state. If this attribute is not specified, the middleware delivers a single consolidated report after the session concludes. Progressive delivery allows broadcasters to obtain timely insights from ongoing sessions while preserving the integrity and traceability of the overall measurement campaign.

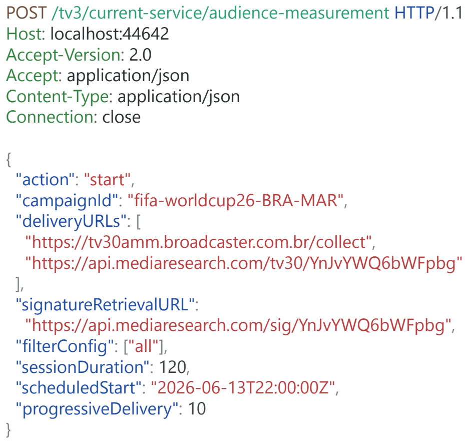
B. Procedural Initiation via TV 3.0 WebServices APIs

In the procedural model, audience measurement sessions are initiated through the TV 3.0 WebServices interface [5], which exposes platform functionalities as RESTful APIs using HTTP requests with JSON-encoded message bodies. This interaction model allows Broadcaster Applications to request audience measurement operations explicitly at runtime, rather than relying on preconfigured metadata. Listing 2 illustrates an example request issued by a Broadcaster Application acting as an associated local client.

As shown in Listing 2, session initiation is performed by sending an HTTP POST request to the standardized / tv3/current-service/audience-measurement endpoint. In the TV 3.0 WebServices design, the semantics of the requested operation are conveyed through a JSON object included in the request body. This object specifies the desired operation and the parameters required to execute it.

In the example, the JSON payload indicates a session creation request by setting the "action" field to "start". This action instructs AMM to instantiate a new audience measurement session associated with the currently selected DTV service and the activew viewer profile. The remaining elements of the JSON object provide the configuration data needed to define the session, using the same conceptual parameters available in the declarative BAM-based model, but supplied dynamically by the application at runtime.

The use of the current-service alias in the request path ensures that the operation is automatically scoped to the service currently tuned on the receiver, without requiring the application to resolve or transmit explicit service identifiers. This design simplifies application logic while preserving strict contextual binding within the middleware. The request is addressed to the local TV 3.0 WebServices HTTP server, which listens on a standardized port, allowing associated local clients to access the API directly without invoking the network discovery procedures defined for non-local clients.



```
POST /tv3/current-service/audience-measurement HTTP/1.1
Host: localhost:44642
Accept-Version: 2.0
Accept: application/json
Content-Type: application/json
Connection: close

{
  "action": "start",
  "campaignId": "fifa-worldcup26-BRA-MAR",
  "deliveryURLs": [
    "https://tv30amm.broadcaster.com.br/collect",
    "https://api.mediaresearch.com/tv30/YnJvYWQ6bWFpbg"
  ],
  "signatureRetrievalURL":
    "https://api.mediaresearch.com/sig/YnJvYWQ6bWFpbg",
  "filterConfig": ["all"],
  "sessionDuration": 120,
  "scheduledStart": "2026-06-13T22:00:00Z",
  "progressiveDelivery": 10
}
```

List. 2. Example HTTP POST request issued by a Broadcaster Application to initiate an audience measurement session via TV 3.0 WebServices.

Upon receiving the request, AMM evaluates whether the requested operation can be performed. This validation step includes checking the viewer's agreement for the relevant data processing purpose, verifying that the requesting application is authorized within the current service context, and ensuring that sufficient platform resources are available. If these conditions are satisfied, AMM creates a new audience measurement session and assign it a unique session identifier.

If the request is successfully processed, the API returns a JSON-encoded response confirming session creation. This response includes the session handle generated by the platform, which shall be referenced in all subsequent lifecycle control requests targeting the same session. In addition, the response conveys implementation-defined delivery parameters, such as the maximum number of report delivery attempts (maxDeliveryAttempts) and the interval between consecutive attempts after session conclusion (deliveryAttemptInterval), as well as the UTC time at which the start action took effect (time). The configured delivery endpoints (deliveryURLs) are also echoed in the response, enabling correlation between the request and subsequent report deliveries.

If session creation fails, the API reports the error using standardized TV 3.0 WebServices error codes. Of particular relevance are error code 102, indicating that the required viewer agreement for data processing was not granted, and error code 200, signaling that the platform is unable to start the session due to resource limitations or internal conflicts. These error conditions explicitly distinguish authorization-related failures from operational constraints and allow applications to adapt their behavior accordingly.

The same REST endpoint supports additional audience measurement operations beyond session creation. By submitting further POST requests with different values in the "action" field, Broadcaster Applications may request changes to the session's execution state, such as temporarily suspending data collection or concluding an active session. These lifecycle control operations, and their effects on session behavior, are described in detail in Section V.

C. Design Rationale for Dual Initiation Models

The coexistence of declarative and procedural session initiation models in TV 3.0 AoP reflects a deliberate architectural decision to decouple measurement control intent from measurement execution. By allowing sessions to be initiated either through static service metadata or through runtime API interactions, the platform accommodates heterogeneous broadcaster workflows while preserving a single, uniform execution model within the middleware.

From a systems perspective, this duality addresses two distinct classes of control requirements. Declarative initiation supports configuration-driven behavior, enabling audience measurement to be bound directly to service availability and platform state without requiring any application-level logic. In this model, broadcasters are exempt from developing dedicated Broadcaster Applications or modifying existing applications solely for the purpose of initiating audience measurement sessions. Measurement can be activated automatically by the platform as part of service activation, reducing implementation complexity and ensuring uniform behavior across receivers.

Procedural initiation, in turn, supports event-driven and adaptive behavior, allowing Broadcaster Applications to initiate measurement in response to runtime conditions such as content boundaries, live events, or editorial decisions. This approach provides broadcasters with fine-grained operational control when application logic is already present or required, while still delegating all data collection responsibilities to the middleware.

Importantly, both initiation paths converge to identical internal session representations managed by the Audience Measurement Manager. This convergence prevents divergence in lifecycle semantics, reporting structure, or governance policies, which are common pitfalls in application-centric measurement systems. Regardless of how a session is initiated, validation rules, contextual binding, state transitions, and delivery mechanisms are applied uniformly by the platform.

As a result, the TV 3.0 AoP architecture avoids coupling audience measurement capabilities to specific application architectures or development choices. Declarative and procedural initiation function as interchangeable entry points to a single, middleware-managed measurement process, reinforcing AMM's role as a flexible yet standardized platform service.

V. API CONTROL AND SESSION MANAGEMENT

Beyond session initiation, the Audience Measurement Manager exposes standardized APIs for controlling and observing the lifecycle of audience measurement sessions. These APIs are part of the RESTful TV 3.0 WebServices interface described in Section IV and use HTTP requests with JSON-encoded payloads. They allow Broadcaster Applications to manage session execution explicitly, while enforcement of context, authorization, and operational constraints remains centralized in the middleware.

Session creation using the start operation ("action":"start"), including the validation steps performed at creation time, is described in Section IV. Upon successful creation, the platform generates a unique session handle (handle) that identifies the audience measurement session throughout its lifecycle. This handle is required in all subsequent lifecycle control requests and status queries and serves as the sole reference to the active session.

Lifecycle control is performed by issuing additional POST requests to /tv3/current-service/audience-measurement endpoint, with the intended operation specified by the value of the action field in the JSON request body and the target session identified by the handle field. Supported operations include temporarily suspending data collection ("action":"pause"), resuming a paused session ("action":"resume"), irreversibly stopping data collection ("action":"halt"), and concluding a session in a controlled manner ("action":"finish"). These operations allow broadcasters to adapt measurement behavior dynamically during service execution without exposing data collection logic at the application level.

For each lifecycle control request, the platform returns a confirmation response containing the session handle and the UTC time (time) at which the requested state transition takes effect. If a control request references an unknown session handle, the API responds with error code 101, indicating that one or more arguments in the request body are invalid.

In addition to lifecycle control, AMM provides a dedicated endpoint for querying the status of an audience measurement session (GET /tv3/current-service/audience-measurement/{handle}). By addressing this endpoint with a valid session handle (handle), an application retrieves a structured snapshot of the session, including the campaign identifier (campaignId), the report identifier (reportId), delivery configuration parameters such as progressive delivery mode (progressiveDelivery), retry behavior (maxDeliveryAttempts, deliveryAttemptInterval), temporal attributes (startTime, finishTime), the current session state (state), and the configured report delivery endpoints (deliveryURLs).

If the provided session handle does not correspond to an existing or previously created audience measurement session, the status query operation returns error code 305, explicitly indicating that the requested session cannot be found. This clear distinction between invalid arguments in control requests and nonexistent sessions in status queries contributes to predictable error semantics across the API.

VI. SESSION LIFECYCLE AND FINITE STATE MACHINE

Audience measurement in TV 3.0 is governed by a formally defined session lifecycle modeled as a finite state machine, shown in Figure 2. This model provides a deterministic and auditable framework for managing audience measurement sessions from their creation through data collection, delivery, and termination. With explicit definitions of permissible states and transitions, predictable behavior of AMM is ensured across receivers as well as robust handling of runtime disruptions.

Each audience measurement session is bound to a unique collection context identified by the pair (service-globalServiceID, viewer-id). This context captures both the broadcast service that initiated the session and the active viewer profile that authorized data collection. Any modification to either element constitutes a disruption of the collection context and is handled explicitly by the state machine. This design guarantees that data collection remains strictly scoped to the intended service and viewer and prevents unintended continuity across context changes.

The session lifecycle begins in the initializing state, which is entered when a session is requested either declaratively by a BAM metadata or procedurally via the TV 3.0 WebServices API. In this state, AMM validates request parameters, verifies platform resource availability, and checks whether the required viewer agreement is in place. Depending on the request configuration, initialization may lead directly to active data collection, transition into a paused state when a future start time is specified, or terminate immediately due to an error condition.

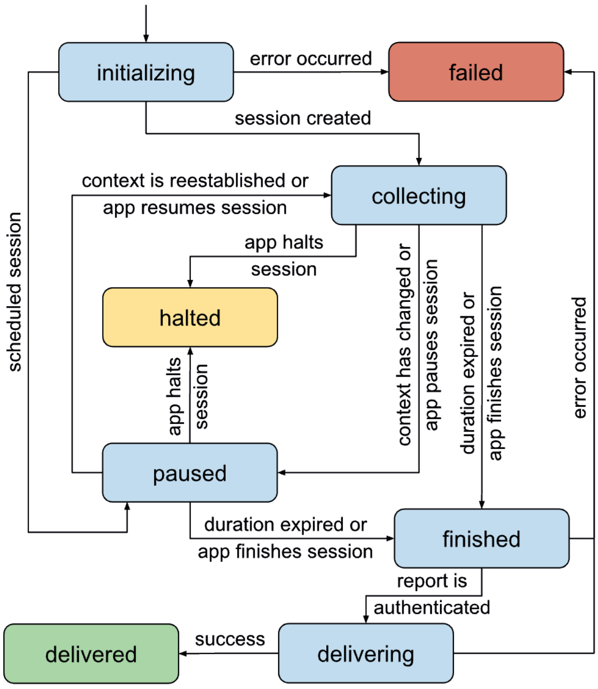


Fig. 2. Finite state machine of the TV 3.0 audience measurement session lifecycle. Adapted from ABNT NBR 25608 [5]

Once active, the session enters the collecting state, during which audience-related behavioral data is continuously gathered, temporarily stored, and protected within the receiver. Data collection proceeds transparently across application transitions as long as the collection context remains unchanged. Collection concludes either when the configured duration expires or when explicitly finished by the broadcaster, resulting in a transition to the finished state. Alternatively, any context disruption—such as service zapping or viewer profile changes—or an explicit pause request causes the session to move into the paused state.

The paused state represents a temporary suspension of data collection while preserving session continuity. From this state, collection resumes automatically when the original context is re-established, when a scheduled start time is reached, or when explicitly resumed by the broadcaster. The session may also be halted or finished directly from this state, allowing broadcasters to retain control over session termination regardless of runtime conditions.

The halted state represents an explicit and irreversible interruption of data collection requested by the broadcaster. In this state, any data collected during the session is discarded and no delivery is performed. AMM retains only minimal metadata, such as the session identifier, final state, and context identifiers, for a limited period, supporting traceability without preserving audience data.

When a session reaches the finished state, data collection has concluded successfully. At this point, AMM performs data authentication procedures to ensure integrity and origin verification. If configured, this includes retrieving digital signatures from an external signature retrieval service and applying cryptographic mechanisms based on ECDSA [15] and SHA-256 [16]. Successful authentication leads to the delivering state, while repeated failures during authentication result in a transition to the failed state.

In the delivering state, AMM attempts to transmit the collected audience measurement report to the broadcaster-defined delivery endpoints. Delivery follows implementation-defined retry policies governing the number of attempts and the interval between them. Successful transmission results in the delivered state, whereas repeated delivery failures exhaust retry limits and transition the session to the failed state.

The delivered state marks the successful completion of the audience measurement procedure. At this point, all collected data is discarded from the receiver, and only minimal session metadata is retained temporarily. Similarly, the failed state represents abnormal termination due to errors during initialization, authentication, or delivery. In this case, any collected data is discarded, and only diagnostic metadata is preserved for a limited time.

VII. DATA COLLECTION AND REPORT STRUCTURE

Audience measurement in TV 3.0 is based on the systematic collection of behavioral, contextual, and system-level data generated during the execution of an audience measurement session. The specification defines a comprehensive set of data identifiers, organized into semantic categories that reflect the broadcast service context, receiver environment, platform operation, application behavior, and viewer interaction. Together, these categories enable a detailed and context-aware representation of how broadcast services and applications are consumed on the receiver.

Every session includes a core set of identifiers that are always reported and that uniquely characterize both the measurement context and the session itself. These identifiers include the broadcast service identifier (service-globalServiceID), the active viewer profile identifier (viewer-id), the session handle, report identifier, campaign identifier, temporal boundaries, current session state, and the delivery endpoints configured by the broadcaster.

In addition to these core identifiers, the specification defines optional categories that may be selectively included according to broadcaster configuration. These categories cover technical characteristics of service delivery, such as physical layer parameters and reception quality; receiver attributes, including manufacturer, model, operating system, and software versions; and platform status information, such as memory availability, connectivity state, audio configuration, and geographic information. This contextual data provides the necessary background for interpreting audience behavior and assessing service conditions.

A substantial portion of the data model is dedicated to capturing the runtime status of TV 3.0 AoP and media playback components. This includes player state transitions, currently presented media and content identifiers, screen position and layering, audio and video parameters, bitrate metrics, and events signaled through DASH MPD metadata. These signaling events encompass SCTE-35 markers, in-band events, and alternative content replacement notifications, enabling precise correlation between audience behavior and media-level dynamics.

The specification also includes detailed support for application-level telemetry. This encompasses the identification and state of available broadcaster applications, foreground application changes, and standardized application lifecycle events related to preparation, presentation, attribution, and selection. Multimodal interaction events, such as voice commands, gestures, facial expressions, and other recognition-based inputs, are explicitly represented, reflecting the interactive and multimodal nature of TV 3.0 services.

Viewer-related data completes the behavioral model by combining static profile attributes, such as language preferences, accessibility settings, and parental control configurations, with dynamic interaction events. These events include key presses from remote controls and other input devices, as well as multimodal inputs recognized during service usage. The model further allows broadcasters to extend viewer profiles with broadcaster-defined attributes, while preserving the hierarchical and standardized structure of the viewer profile.

All data collected during a session is delivered using a standardized JSON-based report format designed to support both end-of-session and progressive delivery. The report is structured into three main objects. The am-session-header object contains the essential session identifiers and metadata and is included in every delivery. The am-session-initstatus object provides a snapshot of the initial values of all selected data identifiers at session start and is included only in the first delivery. The am-session-event object represents the dynamic portion of the report as a chronological list of value changes, each described by a timestamp, category, identifier, and new value. This event-based structure enables efficient progressive delivery, allowing reports to be segmented and transmitted incrementally without losing contextual integrity.

Through this combination of selective data inclusion, event-driven reporting, and flexible delivery strategies, the TV 3.0 AMM provides a scalable and interoperable mechanism for capturing rich viewer engagement data while preserving contextual correctness and traceability.

VIII. CONCLUSIONS AND FUTURE WORK

This paper presented the specification of the Audience Measurement Manager (AMM) as standardized within the TV 3.0 Application-Oriented Platform, a result from Academia R&D effort. By elevating audience measurement to a middleware-managed capability, AMM introduces a consistent, auditable, and context-aware model for collecting audience data in next-generation digital broadcast systems. The specification formalizes session initiation, lifecycle management, data collection, and structured reporting, ensuring that audience telemetry remains strictly bound to service and viewer contexts while supporting both declarative and API-driven control models.

Compared to application-centric tracking approaches and existing service usage reporting standards, AMM provides a more robust architectural foundation for audience measurement. Its middleware-enforced session management model ensures continuity across application transitions, deterministic behavior under context changes, and a clear separation of responsibilities between broadcasters, platform implementers, and external data consumers. The standardized JSON report schema enables interoperability and progressive delivery while preserving traceability and auditability of collected data.

Although the present work focuses on the architectural and operational aspects of AMM, several important research and development directions emerge from this specification. One natural extension is the development of analytical frameworks and processing pipelines capable of transforming raw AMM reports into actionable knowledge. This includes scalable data processing architectures, statistical aggregation techniques, and machine learning models tailored to session-based, event-driven audience telemetry.

Another relevant direction concerns the definition of operational guidelines for secure report authentication and delivery, addressing practical aspects of cryptographic key management, signature validation workflows, and interoperability between broadcaster-side and receiver-side trust infrastructures. While the specification defines the necessary mechanisms, further work is needed to establish best practices for deployment, governance, and cross-organizational verification at scale.

The richness of AMM reports also motivates the exploration of data visualization and exploratory analysis tools designed specifically for session-based audience measurement. Such tools may support temporal analysis, cross-layer correlation between media events and viewer interactions, and progressive insight generation, enabling broadcasters and analysts to better interpret complex engagement patterns.

From an implementation perspective, future work includes the design of conformance testing and validation tools for verifying AMM implementations in receivers and the correct behavior of report delivery endpoints. Automated test suites and reference validators would contribute to interoperability, compliance assessment, and certification processes within the TV 3.0 ecosystem.

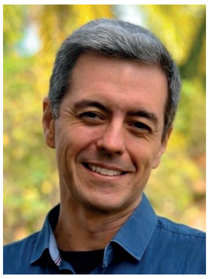
The standardized nature of AMM data further enables advanced Broadcaster Applications that consume audience measurement knowledge derived from analytical and artificial intelligence techniques. These applications may support adaptive content strategies, personalized service features, or operational optimization, while preserving the separation between raw data collection and higher-level inference.

Finally, an important avenue for future research lies in the integration of AMM-derived telemetry with traditional audience measurement datasets maintained by media research institutes. Combining panel-based estimations with platform-level session data may enable comparative studies, hybrid measurement models, and improved statistical inference, bridging established industry practices with the capabilities of next-generation broadcast platforms.

Together, these directions highlight AMM not only as a deployable standard for audience measurement in TV 3.0, but also as a foundation for a broader research and innovation agenda spanning analytics, security, visualization, interoperability, and cross-industry data integration.

REFERENCES

[1] Webster, J., Phalen, P., & Lichty, L. (2013). Ratings Analysis: Audience Measurement and Analytics (4th ed.). Routledge.
[2] Eastman, S. T. & Ferguson, D. A. (2012). Media Programming: Strategies and Practices (9th ed.). Cengage Learning.
[3] Anselmi, G. et al (2024). Watching TV with the Second-Party: A First Look at Automatic Content Recognition Tracking in Smart TVs. In Proceedings of the 2024 ACM on Internet Measurement Conference (IMC '24). Association for Computing Machinery, USA
[4] Varmarken, J., Le, H., Shuba, A., Markopoulou, A. and Shafiq, Z., 2020. The TV is smart and full of trackers: Measuring smart TV advertising and tracking. Proceedings on Privacy Enhancing Technologies.
[5] Associação Brasileira de Normas Técnicas. (2025). ABNT NBR 25608: TV 3.0 – Application Coding [Standard]. ABNT, Brazil.
[6] Brasil, 2018. Law No. 13,709 of August 14, 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Compiled version. Official Gazette of the Union.
[7] Fórum SBTVD. (n.d.). TV 3.0 Project. Retrieved September 21, 2025, from https://forumsbtvd.org.br/tv3_0/
[8] Moreno, M. F. et al (2023). R&D Progress on TV 3.0 Application Coding Layer. SET International Journal of Broadcast Engineering (IJBE), 9(1).
[9] Napoli, P. M. (2011). Audience Evolution: New Technologies and the Transformation of Media Audiences. Columbia University Press, UK.
[10] Webster, J. G. (2014). The Marketplace of Attention: How Audiences Take Shape in a Digital Age. MIT Press, USA.
[11] Wang, S.Q, Gao, L., Chetty, M., Feamster, N. (2025). Understanding User Privacy Concerns of Shared Smart TVs. Proc. ACM Hum.-Comput. Interact. Association for Computing Machinery, USA
[12] European Telecommunications Standards Institute. (2023). Hybrid Broadcast Broadband TV (ETSI TS 102 796 V1.7.1). ETSI.
[13] Tagliaro, C., Hahn, F., Sepe, R., Aceti, A. and Lindorfer, M. (2023). I still know what you watched last Sunday: Privacy of the HbbTV protocol in the European smart TV landscape. In 30th Annual Network and Distributed System Security, NDSS 2023.
[14] Advanced Television Systems Committee. (2024). ATSC A/333:2024-04 Service Usage Reporting [Standard]. Washington, DC.
[15] ANSI. (2020). ANSI X9.142-2020: Public Key Cryptography for the Financial Services Industry – Elliptic Curve Digital Signature Algorithm (ECDSA). American National Standards Institute.
[16] National Institute of Standards and Technology (NIST). (2015). FIPS PUB 180-4: Secure Hash Standard (SHS). U.S. Department of Commerce.
[17] Moreno, M. F., Barrére, E., Saade, D. C. M. (2025). TV 3.0 Privacy Management: Signalling, Enforcement and Rights Control. SET International Journal of Broadcast Engineering (IJBE), 11(1).
[18] Associação Brasileira de Normas Técnicas. (2025). ABNT NBR 25602: TV 3.0 – Transport Layer [Standard]. ABNT, Brazil.



Marcelo F. Moreno is an Associate Professor at the Federal University of Juiz de Fora (UFJF), Brazil. He holds a Ph.D. in Computer Science from PUC-Rio and works in the areas of multimedia systems and computer networks. From 2022 to 2023, he was a Visiting Professor at the International Audio Laboratories Erlangen (FAU/Fraunhofer IIS), Germany. He co-edited the ITU-T Recommendation H.761 (“NCL and Ginga-NCL”) and has contributed to several international standards, coordinating ITU-T working groups for over a decade. Since 2015, he has led the Application Coding Working Group of the Brazilian Digital TV System Forum (SBTVD) and serves as editor of the ABNT standards for TV 2.5 and TV 3.0. His research focuses on application-oriented broadcasting, second-screen integration, audience measurement, and privacy-aware media platforms for next-generation digital television. He has published widely in the field and is a member of the IEEE and the Brazilian Computer Society (SBC).

Eduardo Barrére has been a full professor in the Department of Computer Science at the Federal University of Juiz de Fora (UFJF) since 2009. He holds a master's degree (1997) in Computer Science form the Federal University of São Carlos (UFSCar) and a doctorate (2008) in Systems and Computing Engineering from the Federal University of Rio de Janeiro (UFRJ). He is the coordinator of the Computer Application and Innovation Laboratory (LApIC) at UFJF. Develops research in the areas of computer networks, Digital TV and multimedia.

Received in 2025-07-24 | Approved in 2025-08-04