News

Revised reprint

ROHDE&SCHWARZ

Mastering convergent content monitoring in a diverse world of media and broadcast

Paper originally presented in invited talk at NAB Broadcast Engineering and Information Technology Conference (BEITC) 2017, Las Vegas

Mastering convergent content monitoring in a diverse world of media and broadcast

Abstract – In the past, broadcast networks were one-dimensional. Classical content monitoring devices had mostly static requirements, resulting in hardware-dominated designs. Today's broadcast networks are multi-dimensional, resulting in numerous new challenges for monitoring: diverse source signals in contribution including migration to IP, diverse end devices in distribution, diverse OTT protocols, collection and aggregation of information from numerous different protocol layers and technologies, and rising cloud architectures using asynchronous data transfer structures.

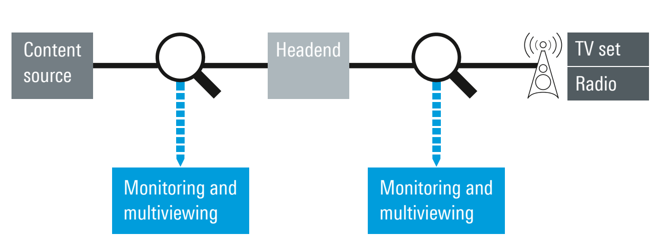
As a solution this work describes a new, purely software-based approach to monitoring: COTS IT server hardware is used to build a unified, convergent monitoring system for both contribution and distribution. The complete layer 2-5 transport protocol stack and media content can be monitored with just a single convergent device. Characteristic KPIs to be monitored at Internet layers 2-5 will be explained. The result is a single, consistent picture, describing the complete state of the respective broadcast and media service. New monitoring functions like Content Compare become possible only in purely software-based systems. The paper closes with an outlook on future monitoring trends and challenges like migration into the cloud and remote use via mobile smart devices.

Introduction

In the past, classical content monitoring for broadcast services was a more or less straightforward task as illustrated in Figure 1. The world was practically one-dimensional with uniform and specialized transport technologies and media formats in contribution and distribution environments, respectively. Also the type and number of end devices consuming broadcast services was dedicated and these devices (TV sets, radios) did not have any true intelligence or "connectivity" of their own.

Consequently, monitoring devices were subject to more or less static requirements. Their resulting technical architectures were mostly hardware-oriented (see [Ref. 01] for some examples). This approach served the industry very well as long as the broadcasting environment did not change too much. However, highly specialized and focused hardware-dominated solutions are known to quickly become inflexible and insufficient as soon as customer needs and application scenarios become highly dynamic and fast changing.

Figure 1 Classical content monitoring in a one-dimensional, uniform world of TV and radio broadcasting



The challenges of the present

Today, in contrast to the past, the number, type and variation of media services (streams) and the devices consuming them is literally exploding (see Figure 2 for illustration). Consequently, the industry is quickly moving towards Internet protocol (IP) techniques as the prime transport method as well as IT-based deployments.

Thus, the industry presently is facing the following challenges:

Asynchronous, packetized signal transport

The long-time well established practice of monitoring a point-to-point connection is being challenged by monitoring "the cloud", where synchronous audio/video signals are split into pieces, packetized, and spread over asynchronous data transfer structures.

Plethora of signal sources

In contribution/playout environments, the number and type of source signals to monitor is increasing strongly.

Legacy SDI and SDIoP mixture

Until the transition to IP-based transport is 100% complete, a mixture of legacy SDI content sources and a set of (sometimes competing) IP-based source signals like SMPTE 2022-6/7, AIMS and ASPEN is to be monitored – preferably in a single device.

Uncompressed/mezzanine format mixture

Besides uncompressed signals, also new mezzanine formats like J2K and TICO are to be handled by the monitoring system.

Plethora of (intelligent) end devices

In distribution environments, the number and type of end devices is literally exploding (classical TV sets and radios, web browsers with plug-ins, a plethora of OTT streaming clients and VoD players in set-top boxes, smart home appliances, smartTVs, and mobile intelligent devices like tablets and smartphones).

Plethora of (ever growing) media transport protocols

Simultaneously, the number of media transport protocols to be supported (MPEG-DASH, HLS, HDS, RTMP, MSS/HSS, ICECAST, HBBTV, etc.) is exploding.

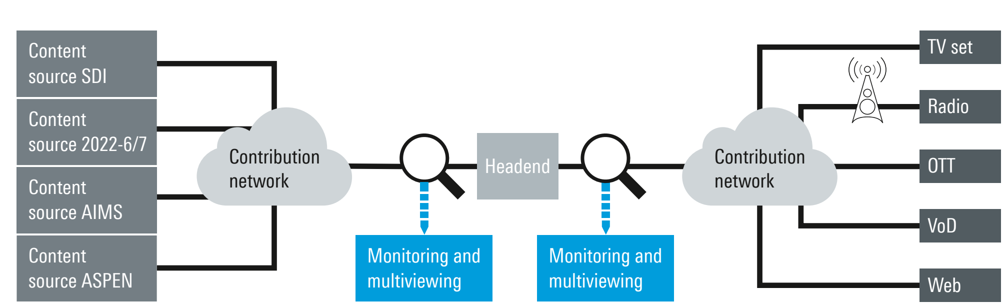
Complex layered protocol architecture

With a single broadcast and media service simultaneously involving numerous protocols from the physical up to the application layer (see Figure 3 for an example), complete and concise status information requires collection and aggregation of information from various different layers and technologies at the same time.

Given the aforementioned set of new challenges, the traditional approach of hardware-oriented monitoring devices outlined earlier can no longer satisfy the multiple and diverse requirements of today's broadcast and media service providers.

A new, more flexible solution to content monitoring in a multi-dimensional, diverse world of media and broadcasting is required.

Figure 2 Convergent content monitoring in a multi-dimensional, diverse world of media and broadcasting



Technical solution

To this end this section will describe design principles and architectural building blocks for a new, purely software-based approach to monitoring, using COTS IT server hardware to build a unified, convergent monitoring system for use both in playout/contribution and distribution environments.

These general design and architectural principles will be illustrated with two resulting key benefits which become possible only in future-proof, purely software-based systems:

- Multi-layer KPI monitoring, providing a single, consistent picture describing the complete state of the respective broadcast and media service
- New, advanced monitoring functions, e.g. Content Compare (CC), a new, advanced monitoring function for media content for automated convergent monitoring of simultaneous, heterogeneous media streams.

I. Design principles

A monitoring solution meeting today's and future requirements is expected to meet the following general design principles in order to provide the corresponding benefits:

Purely software-based with hardware-agnostic programming

This is an approach based solely on software ensures fast and flexible adaptation and extension to changing and newly arising requirements such as new formats for media encoding and transport. Hardware-agnostic programming assuming only mainstream, general purpose CPUs ensures a high level of portability for later deployment on various hardware and virtualized platforms.

Platform based on COTS IT hardware or OVF-compatible hypervisor

Leverage of COTS equipment provides low CAPEX and OPEX relative to proprietary hardware platforms. Compatibility to the Open Virtualization Format (OVF) [Ref. 01] for images of virtual machines ensures cloud-readiness of the software solution.

IP-based transport and signaling protocols wherever possible

Putting transport and signaling data as fast as possible and wherever possible onto IP allows for the implementation to use the well-established and rich framework of the TCP/IP protocol suite.

Interworking to legacy formats/signals through interface cards

The leverage/interworking with the installed base of legacy format/signals is ensured through corresponding interface cards as required. As the needs for legacy formats/signals these cards can be phased out and removed such not burdening the installation and leaving only a future-proof purely software-based solution.

Open and modular software architecture

Use of open protocols/formats and modular functional building blocks also ensures fast and flexible adaptation and extension to changing and newly arising requirements. Separation into modules also improves system robustness and isolates potential software faults, e.g. due to male formed input signals.

Figure 3 Internet 5-layer model for IP-based transport of signals in playout/contribution and distribution networks

Application layer	**SDI** (serial digital interface)	SMPTE 259M/292M/424M		
	HBRMT (high bit rate media transport)	SMPTE 2022-6	**DASH** (dynamic adaptive streaming over HTTP)	ISO/IEC 23009-1: 2014
	RTP (real-time transport protocol)	RFC 3550	**HTTP** (hypertext transport protocol)	RFC 2616
Transport layer	**UDP** (user datagram protocol)	RFC 768	**TCP** (transmission control protocol)	RFC 793
Network layer	**IPv4/v6** (internet protocol)	RFC 791/2460	**IPv4/v6** (internet protocol)	RFC 791/2460
Data link layer	**MAC** (media access control)	IEEE 802.3, IEEE 802.11, ...	**MAC** (media access control)	IEEE 802.3, IEEE 802.11, ...
Physical layer	**Physics and mechanics** (voltages, wiring, connectors...)		**Physics and mechanics** (voltages, wiring, connectors...)	

Scalability
Leverage of Moore's Law [Ref. 02] in the form of ever-growing processing power in COTS general purpose CPUs through multi-thread/core programming techniques. With correspondingly designed software solutions, scalability is predominantly an issue of sufficient processing power and network connectivity as the growing number of services and delivery formats and platform types can easily be handled by adding additional CPUs or server capacity. With hardware-dominated monitoring solutions, a straightforward solution to scaling is almost impossible.

II. Architectural building blocks

Adhering to the aforementioned design principles, the proposed technical solution builds on state-of-the-art streaming technologies, new IP-based transport technologies replacing classical SDI along with new mezzanine compression technologies (contribution). In such a system architecture, legacy technologies can be included via special purpose interface cards to allow for monitoring of hybrid legacy signals and new IP-based transport formats.

Figure 4 illustrates the generic architecture of an extensible software framework and its key functional building blocks for convergent content monitoring. The software operates as a standard application in user land on a modern, general purpose operating system (e.g. Linux).

The operating system can reside either on a COTS hardware platform (industry server type) or on a suitably configured hypervisor compatible to OVF format for virtual machine images. The OVF format ensures fast, flexible and hassle free migration of the monitoring instance into different hypervisor instances when deployed in private our public cloud installations.

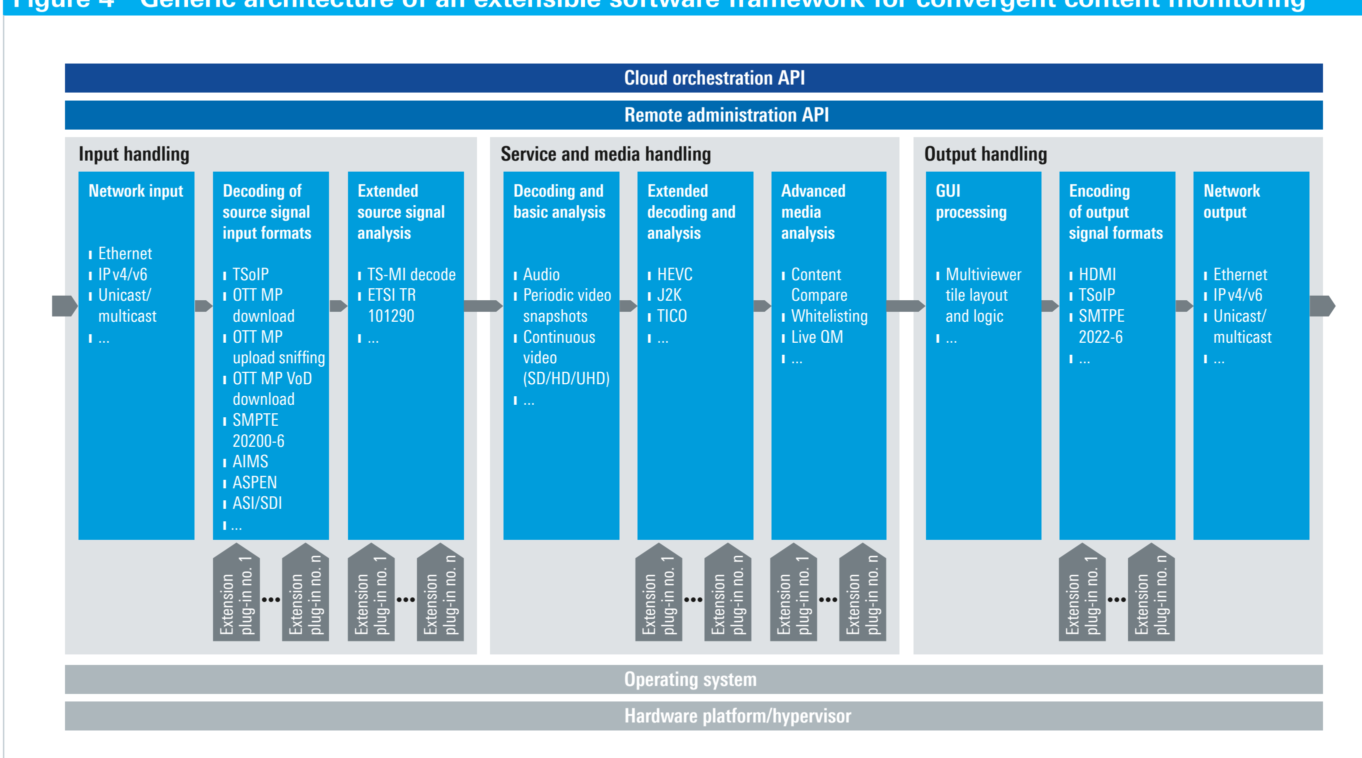
The extensible software framework is structured into the following key architectural building blocks:

Input handling
The "Network input" module takes care of the various physical media such as Ethernet and IP versions. It also handles reception of media transport streams either as unicast or multicast.

The module for "Decoding of source signal input formats" is key in our approach for convergent content monitoring. It unites the handling of the constantly increasing plethora of present (and future) formats for IP-based media transport in OTT/streaming and playout/contribution environments. It also contains the drivers for handling legacy input formats such as SDI and ASI. Future extensions to new source signal formats is ensured via plug-ins – an approach well established in IT to ensure a future-proof, purely software-based solution.

Optionally, input handling is completed by a third module for "Extended source signal analysis" which is also extensible through corresponding plug-ins, if required.

Figure 4 Generic architecture of an extensible software framework for convergent content monitoring



Service and media handling

This functional element starts off with a module for "Decoding and basic analysis" of the signal received from the function for input handling. Audio/video content is decoded and some basic analysis of the correctness and quality of the signal is performed.

To handle any advanced or new (mezzanine) encoding formats such as HEVC, J2K, TICO, etc., an optional module for "Extended decoding and analysis" exists. As the need by media and broadcast service providers for new decoders or more in-depth analysis arises, the function of this module is extensible by suitable software plug-ins. With the ever-growing power of CPUs in physical server platforms or in cloud deployments, there is no need for the costly deployment of special purpose, often proprietary hardware to handle advanced encoding formats anymore. The required functional extensions are all handled purely in software via plug-ins.

In the "Advanced media analysis" module the decoded content can be analyzed and monitored further, if required. Functions in this module include classical live quality monitoring or monitoring of freeze frames against a pre-stored set of whitelisted images. Another advanced monitoring task which becomes possible in a purely software-based solution is Content Compare and will be explained later in still greater detail for further illustration. Custom monitoring functions specific only to a single media and broadcast service provider can be implemented cost effective and flexibly as software plug-ins in this functional module. Again, such a high level of customer specific customization of monitoring functionality is almost impossible to achieve with traditional, hardware dominated monitoring solutions.

Output handling

The module for "GUI processing" handles tasks such as the administration of the layout and logic of the tiles for the multiviewer display. It also performs the output of any possible measurement parameter along with the monitored audio/video content in the respective tile.

The dynamically compiled image of the multiviewer screen in further processed in the "Encoding of output signal formats" module, e.g. for output on a physical video screen with HDMI or as several types of IP-based audio/video streams. Extension to other output types is supported via the provision of corresponding new software plug-ins.

If the target output format is an IP-based stream, the packetized output signal is finally send out by the "Network output" module via the selected IP protocol version, method and physical medium.

Remote administration

HTTP-based API for comprehensive remote control of monitoring instance and corresponding remote automation of monitoring tasks. Includes the sending of information on instance-related alerts and alarms via email to named recipients.

Cloud orchestration

Open API for inclusion of monitoring instance as an SaaS building block into a cloud-based, end-to-end workflow including other functional instances such as playout, en/trans/decoding, ad/logo insertion, DRM, etc.

III. Benefit 1: Multi-layer KPI monitoring

Figure 3 shows the Internet 5-layer model applied to IP-based transport of SDI signals in playout/contribution environments and to MPEG-DASH in OTT/streaming environments. The figure clearly illustrates that except for the top of the transport layer and of course the application layer itself, a great deal of technical overlap exists in IP-based scenarios for IP-based transport of SDI signals in playout/contribution environments and e.g. OTT video streaming in distribution environments. This great overlap can be leveraged in purely software-based architectures to provide cost effective, convergent solutions for the monitoring of all IP-based media transport formats in a single application.

At lower layers, unified monitoring of the underlying IP transport network via characteristic KPIs (key performance indicators) such as effective throughput, delay, jitter, packet loss rate, re-routing frequency, etc. can be performed.

At higher signal-specific layers, KPIs such as allocated bandwidth per stream, bitrate, latency, jitter, packet loss, RTP status, source/destination checks, etc. specific for IP-based SDI signals (SMTE 2022-6) can be monitored. Simultaneously, the same application can monitor KPIs like manifest timeout/updates, fragment timeout, HTTP error

Figure 5 Sample screenshot of GUI output for Content Compare function.

codes, bit rate measurements – per service and per profile, buffer status checks (over/under flow), download times, etc. specific for OTT video streaming, too. At lower layers (e.g. the TCP layer) KPIs like packet retransmissions, packet loss, session reconnections, buffer measurements, etc. can be monitored, uniformly for all IP-based signal types.

The ability to retrieve, analyze, monitor and combine this information from different layers and signal types in a single application instance, provides the media and broadcast service provider with a concise and complete picture of the state of his various ongoing services and the underlying IP transport network through the complete set of protocol layers. The benefit comes from the convergence of such diverse information as the underlying IP transport network (bandwidth, delay, jitter, packet loss rate, etc.) or information on lost frames and encoding errors into a single, consistent picture. Full situational awareness can be achieved via a single convergent monitoring instance.

IV. Benefit 2: New, advanced monitoring functions

As already stated earlier, a prime example for new, advanced monitoring functions becoming possible with purely software-based solutions is called Content Compare.

Content Compare (CC) provides automated convergent monitoring of simultaneous, heterogeneous media streams carrying identical content.

A screen shot of the corresponding tile in a multiviewer GUI is shown in Figure 5.

In a heterogeneous broadcast environment where broadcasters deliver the same content, but in different resolutions across different platforms, combined with the complexity of combined Ethernet switching and IP routing, there is the potential of the wrong content being delivered to the wrong platform. In one potential critical scenario, adult-rated content may be erroneously encoded onto a children's TV channel. The ability to automatically monitor the content of an outgoing channel against a known good channel can prevent such type of scenarios. Direct analysis on a pixel by pixel level would prove too inaccurate and may cause many false alarms. So with CC, a more objective method of detecting the content is employed, where the automated system can decode and compare the images in the two content streams.

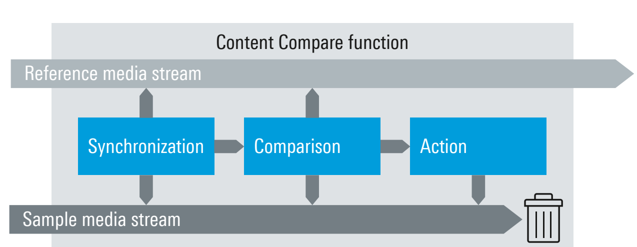
Figure 6 illustrates the principal steps involved in the CC function.

After an initial step of synchronizing the two media streams, the actual step of comparison of the sample relative to a reference streams uses criteria such as moving objects, scene cuts, average luminance levels, etc. to compare the images. Based on the result of the comparison, the function can trigger some pre-defined action, e.g. in the aforementioned example of a children's TV channel, discarding the erroneous sample stream.

In Figure 5, the video image shown in the multiviewer tile represents the reference media stream of Figure 6. In the upper left corner, the tile carries labels for two channels which are being monitored using the CC function relative to this reference stream. One channel is marked to have passed (green label), the second channel has failed (red label) the CC check.

While of course CC would be theoretically also possible e.g. in a FPGA-based implementation, such a hardware-dominated solution would proof to be technically and economically not feasible in practice given the fast evolving type and number of content formats and delivery platforms.

Figure 6 Principal steps of Content Compare function



Future trends and challenges

This section provides an outlook on future trends and challenges for the content monitoring of media and broadcast services. With its inherent capability to be adapted/extended (e.g. via the concept of plug-ins outlined in the description of the modular, functional architecture) a purely software-based monitoring system is able to address newly arising requirements such as:

I. Distributed monitoring

Today and more increasingly in the future, media and broadcast services will become very diverse and multi-dimensional (recall Figure 2 for illustration). This requires the system architecture of content monitoring systems to cope with requirements such as distributed (unmanned) collection of content information and the central processing/visualization of corresponding. In other words, data is to be collected from a set of distributed type of monitoring probes, but is to be processed in a single, central unit for network-wide situational awareness for broadcast and media service providers (see left half of Figure 7).

Such a future requirement can readily be addressed by the IP-based, modular software architecture outlined in the previous chapter: The functional block for "Output handling" is extended by a corresponding module to send the monitoring data over the network to the central collection point where a matching plug-in in the "Input handling" block receives it and passes it on for further processing.

II. Secure remote monitoring

Building on the distributed functional work split described in the previous section, Figures 7 and 8 illustrate the solution to another increasingly pressing requirement: the secure provision of remote monitoring to service personnel which is on shift but may be offsite in a potentially unsecure environment. The challenges is to offer these users fully situational awareness without risking the integrity and security of the media and broadcast service providers' core network.

The architectural characteristics of the approach in Figure 7 are:

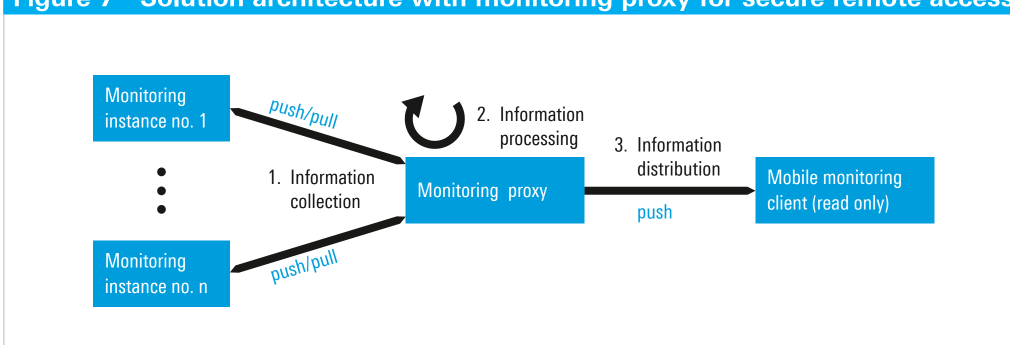
- "Push/pull architecture" between central proxy and a set of distributed monitoring servers.
- Event driven "push architecture" between central proxy and remote mobile clients.

In step 1 monitoring data is collected in push or pull mode from a set of distributed monitoring instances (probes/tap devices). Next, this information is processed centrally in a single monitoring instance in step 2. The central instance also acts as a proxy for the distribution of monitoring information to mobile monitoring clients in step 3. This information is distributed in the format of a "virtual multiviewer wall", leveraging standard OTT video streaming technologies. To ensure full data and network security, this distribution is performed from the proxy side in pure push mode (optionally event driven via alerts/alarms) and only to authenticated and authorized remote clients. Additionally, all clients have strict read-only access to monitoring information and cannot (re)configure any settings on the central monitoring proxy and on the distributed probes.

Figure 8 sketches a possible real-world deployment scenario of the schematic architecture outlined in Figure 7:

- Monitoring instances (probes/tap devices) located in several, distributed locations (e.g. MCR/NOCs) are connected via IP-links to the central processing instance and monitoring proxy. If required, these IP links can be secured e.g. via IPSec or equivalent means
- Mobile monitoring clients connect to the monitoring proxy either locally (e.g. via WLAN) or remotely using VPN secured connections. All clients have to provide authentication/authorization credentials before they are granted read-only access. Once clients are securely connected, the monitoring proxy pushes "virtual multiviewer wall" information to them
- Clients are mobile, hand-held devices like COTS tablets or smartphones operated by monitoring/service personnel which is either roaming around onsite or is offsite but needs full situational awareness of the media and broadcast service status

Figure 7 Solution architecture with monitoring proxy for secure remote access



The solution will allow for OPEX optimization as fewer manned monitoring positions are required and some of the on-shift monitoring personnel may even be offsite.

CAPEX savings can be achieved if parts of the outlined architecture are migrated into the cloud.

Needless to mention that implementation of the solution despite all its already built-in security measure still requires ongoing consulting and close supervision by experts on cybersecurity as new threads and attack vectors keep arising and constantly challenge any existing security concept.

III. On-demand software licenses

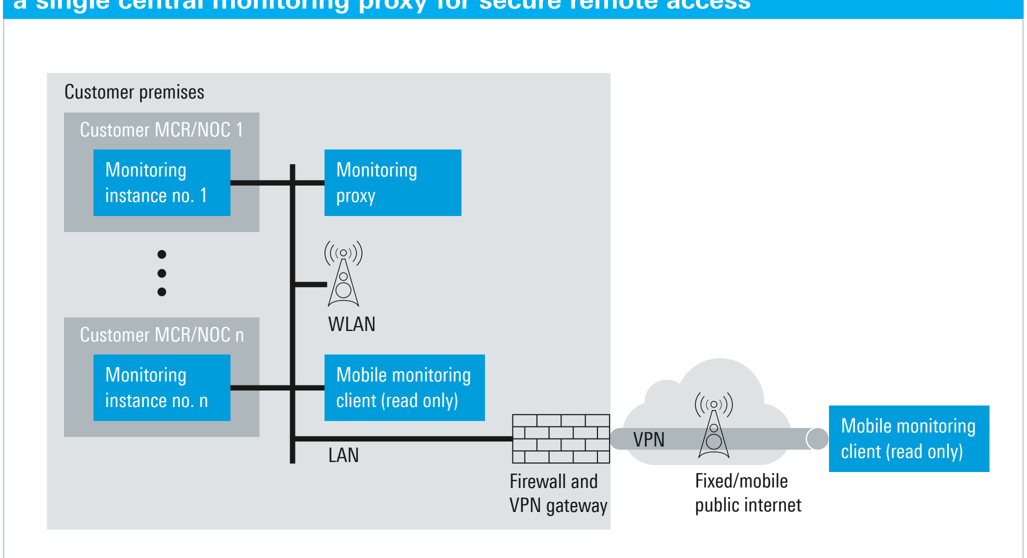
Software-dominated, distributed and network-based monitoring architectures also allow for further CAPEX optimization through on-demand software license allocation usually not available in traditional hardware-dominated solutions: a software key license server is used to centrally administer all licenses available to a pool of distributed monitoring instances which are connected over the IP transport network to the license server. If a particular monitoring instance requires a certain license key to locally enable a special monitoring capability in software only for a particular period of time, it dynamically requests the license key from the central license server. Based on administratively configurable policies, the license server grants the requested license key and internally marks it as in use. This dynamic, on-demand allocation and migration of software license keys minimizes the total number of software licenses a media and broadcast service provider needs to purchase to optimally equip his complete monitoring infrastructure. The net effect is a very cost effective consumption of potentially expensive software licenses as any idle software license may be re-used anywhere in the service provider's monitoring infrastructure.

The capability of on-demand shared software licenses is also an important prerequisite for the true cloud-deployment of a solution where monitoring instances can dynamically migrate through the cloud infrastructure all the time.

IV. Virtualization/migration into the cloud

The recommended software design principles outlined in Sect. I of the previous chapter ensure full virtualization of the monitoring solution. Thus, later easy migration of former hardware-dominated monitoring solutions into the cloud becomes feasible. Equipment vendors then can provide the same software and feature content not any longer only preinstalled on physical hardware but also as OVF image to their customers in the media and broadcast service provider community. Providers can then decide to operate their virtualized monitoring instances in their own private cloud infrastructure or in some rented public cloud (IaaS service model). In the extreme case, the monitoring service could even be implemented as SaaS for media and broadcast service providers by some public cloud operator.

Figure 8 Operational scenario with several distributed monitoring servers and a single central monitoring proxy for secure remote access



V. Orchestration in the cloud

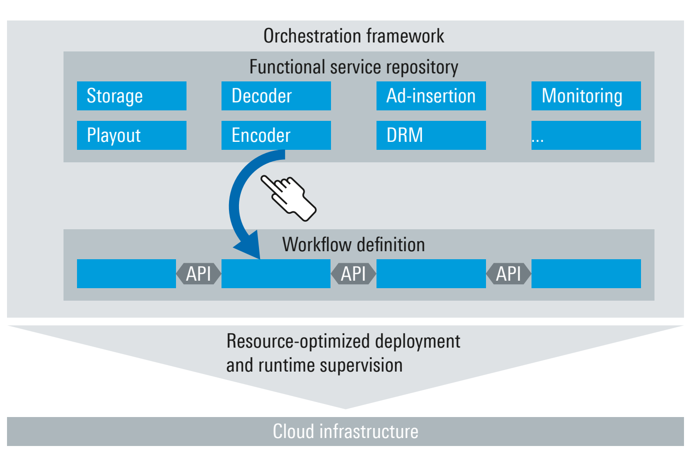
With key functional building blocks of the workflow for media and broadcast services (content source, encoder, ad/logo insertion, DRM, monitoring, etc.) being virtualized, formerly physical cable connections between the physical representations of these blocks do not exist any longer. Instead they are replaced by (networked) software APIs between the software instances now representing the functional building blocks.

This in turn asks for a cloud-capable framework for software-based functional instances with defined APIs between them (see Figure 9 for illustration). APIs shall preferably be open and follow a common industry standard to allow customers to pick the best-in-breed software instance from different vendors at each position in the workflow.

The net effect of such a virtualized framework and set of software functions with defined APIs will be a high level of orchestration: the definition, configuration, operation and supervision of complete workflows for media and broadcast services via modern, GUI-based drag-and-drop methods. Besides faster "time-to-service", a high level of orchestration promises noticeable savings in CAPEX and OPEX.

It is worth mentioning that the framework also can take care of the resource optimized distribution of virtual software instances across the cloud-infrastructure (e.g. putting encoder instances on the most powerful hypervisors) and of intelligent runtime supervision of instances (e.g. automatic restart or migration of an instance in case of a software or hypervisor failure).

Figure 9 Cloud-capable orchestration framework for software-based functional instances with defined APIs



Conclusion

Today, media and broadcast networks are multi-dimensional with respect to the plethora of content sources, media formats and end devices to be supported. This results in numerous new challenges for content monitoring.

This work describes a purely software-based approach to monitoring in order to overcome these challenges: following the outline of key design principles, a generic architecture of an extensible software framework for convergent content monitoring is proposed. Key functional building blocks and modules of the architecture are identified and their respective tasks explained. Plug-ins are an approach well established in IT to ensure future-proof, purely software-based solutions. The same concept is suggested here to extend functional modules as new requirements for monitoring keep constantly arising.

As a prime example for new, advanced monitoring functions becoming technically and economically feasible with purely software based solutions, the so-called function Content Compare is explained.

In closing, the paper addresses some forthcoming trends and challenges in content monitoring. Notably, secure remote monitoring via "virtual multiviewer walls" on mobile smart devices and orchestration of workflows for media and broadcast services in the cloud are elaborated. Key elements of corresponding deployment architectures and frameworks are proposed and positive impacts on CAPEX and OPEX are identified.

The author likes to thank his colleague Nick Ward for valuable input and discussions.

Dr. Markus Lautenbacher

REFERENCES

[Ref. 01] Fischer, W., "Digital Video and Audio Broadcasting Technology", Third Edition, Springer, ISBN 978-3-642-11611-7, 2010.

[Ref. 02] Distributed Management Task Force (DMTF) Inc., Open Virtualization Format Specification V1.0.0 (preliminary standard), 2009. http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

[Ref. 03] Moore, G., E., "Cramming more components onto integrated circuits", Electronics, Vol 38, No 8, April 19, 1965.

Dr. Markus Lautenbacher is product manager within the Broadcast and Media Division of Rohde & Schwarz in Munich, Germany. He is responsible for R&S®PRISMON, a software-based solution of monitoring of audio/video content in broadcast and streaming media services.

He joined Rohde & Schwarz in 2008. Before his current position he worked at Rohde & Schwarz as a senior system architect and product manager for IP-based mission critical voice communication systems.

From 1996 to 2007 he worked as senior system architect for mobile data networks at Siemens Mobile Communication Networks and later at Nokia Siemens Networks. He has been an avid ARPANET/Internet user since the 1980's. He holds a doctorate in Theoretical Physics from the Technical University of Munich, Germany.

Regional contact

- Europe, Africa, Middle East | +49 89 4129 12345
  customersupport@rohde-schwarz.com
- North America | 1 888 TEST RSA (1 888 837 87 72)
  customer.support@rsa.rohde-schwarz.com
- Latin America | +1 410 910 79 88
  customersupport.la@rohde-schwarz.com
- Asia Pacific | +65 65 13 04 88
  customersupport.asia@rohde-schwarz.com
- China | +86 800 810 82 28 | +86 400 650 58 96
  customersupport.china@rohde-schwarz.com

www.rohde-schwarz.com

PD 5215.3462.92 | Version 01.00 | July 2017 (ch)
Mastering convergent content monitoring
Data without tolerance limits is not binding | Subject to change

5215346292

5215.3462.92 01.00 PDP 1 en