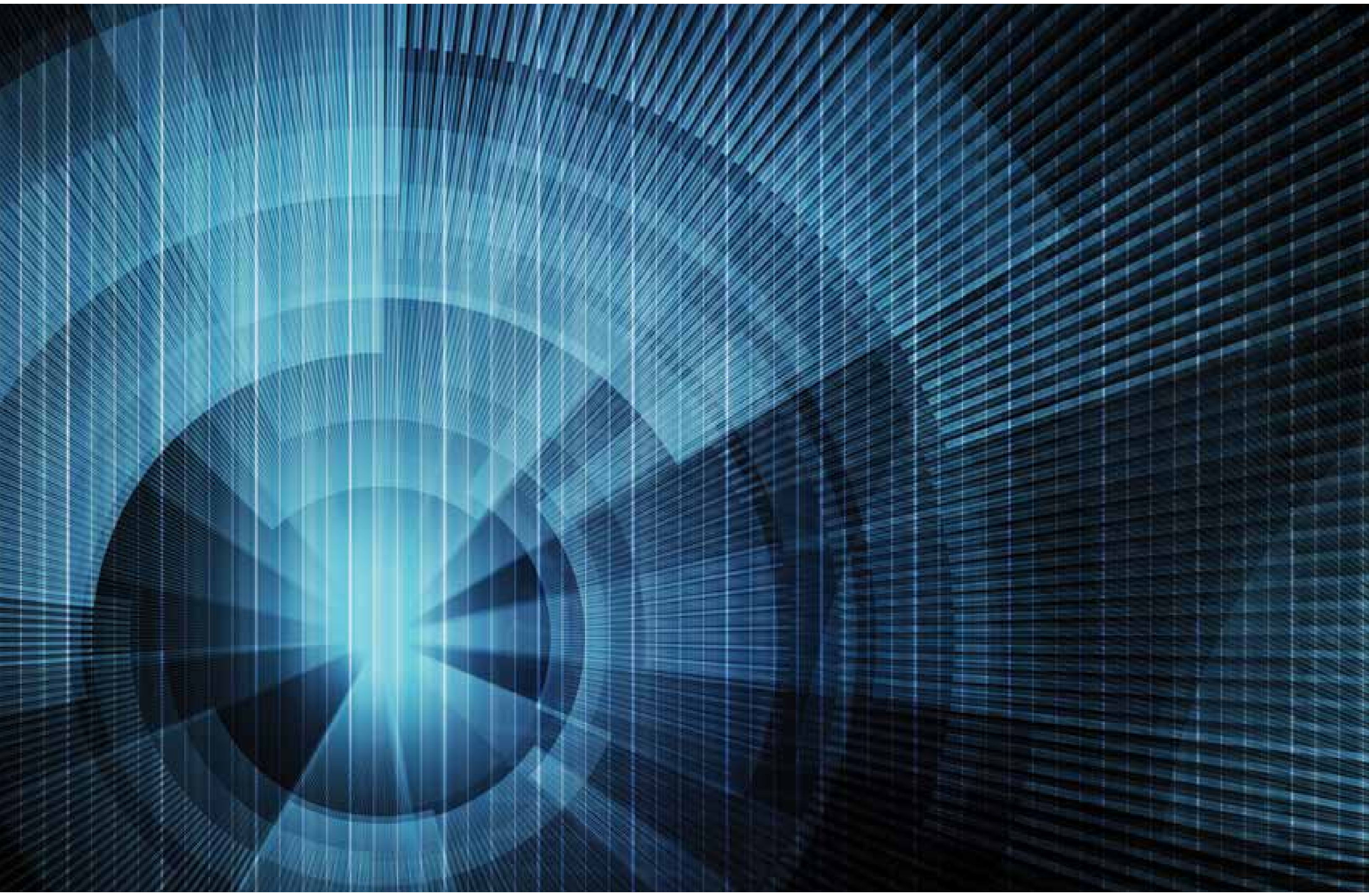
The Genesis of a Streaming Platform

How we created TNA, a high-performance, highly scalable live streaming architecture designed for our enterprise customers



TULIX SYSTEMS

The Genesis of a Streaming Platform

How we created TNA, a high-performance, highly scalable live streaming architecture designed for our enterprise customers

During our meticulous monitoring of the systems running TNA during these evaluation periods, we noticed the lowest CPU, memory, and drive subsystem utilization we had ever witnessed.

In our many years as a streaming service provider, Tulix has amassed a diverse array of customers with a similarly broad spectrum of requirements for their streaming solutions. We have grown from providing simple personal streaming accounts to working with large-scale subscription OTT/IPTV services that stream hundreds of channels to thousands of concurrent viewers on a daily basis. This expansion has necessitated building a scalable enterprise streaming infrastructure for our largest customers. Along the way, we have gained a lot of experience with the extensive variety of streaming software and hardware on the market, which has proved invaluable in creating custom configurations that are optimally tailored to the specific needs of our different customers.

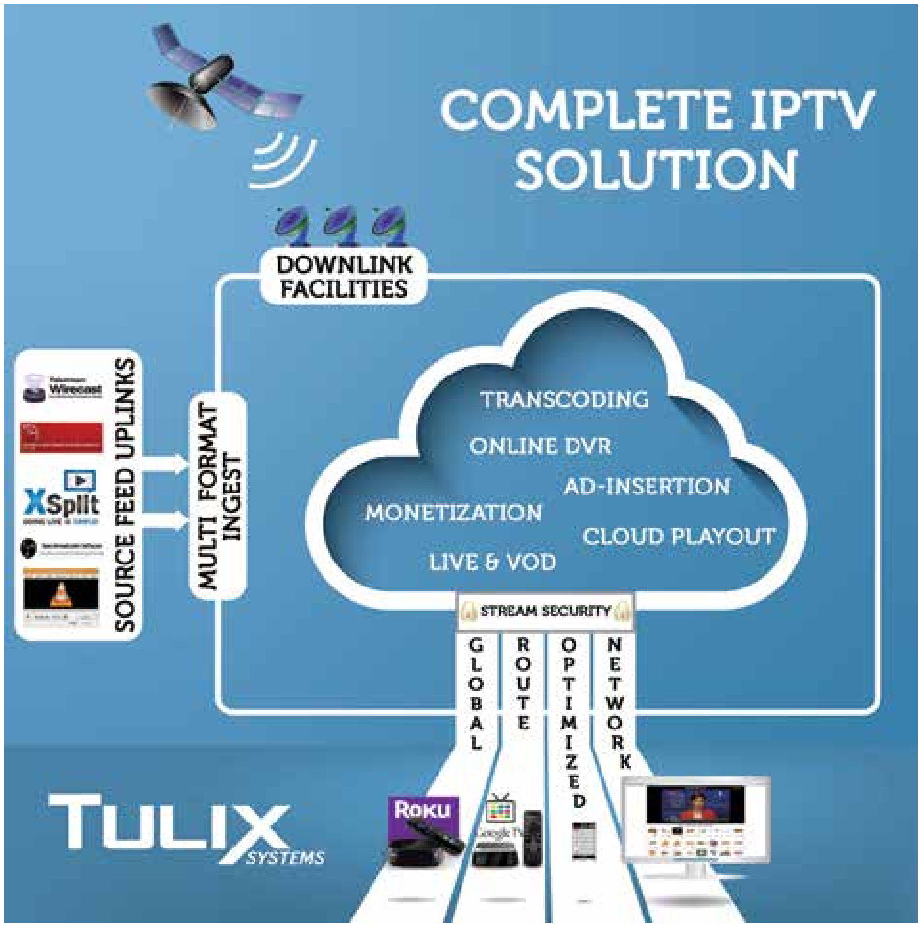
As a complete OTT video delivery provider, our biggest challenge was designing a new streaming architecture to provide the highest quality of service for our customers offering subscription- based live content. These customers have very high standards for stability, scalability, and redundancy, as they are held accountable by subscribers who expect around-the-clock uninterrupted access to the content they are paying for. As their CDN, we are responsible for ensuring that these standards are met and that their viewers receive a high-quality, reliable service. When abnormalities occur, whether widespread or confined to individual users, we will almost always hear about them, with the expectation that we will investigate their origin.

The number and volume of our customers streaming monetized content has increased substantially in recent times, and we have invested a lot of time and resources into building a streaming architecture that meets their needs. We have tested over a dozen different commercial and open-source streaming solutions, both by themselves and in various combinations, to build the best possible live-streaming architecture. Though many had excellent features and performed very well overall, we found that they suffered from stability issues when tasked with running the volume of channels and viewers that our larger customers required.

Tulix had to solve issues that looked (and sometimes were) unresolvable without looking beyond just a single commercial streaming server. After countless attempts at fine-tuning configurations that yielded only modest results, we realized that providing a stable and scalable streaming environment for our high-volume customers would require designing an entirely new architecture, one that was both flexible and extensible as well.

We also determined that it would be critical to be able to leverage state-of-the-art hardware in order to maximize resource efficiency within a globally distributed network architecture. The process of research and testing that followed took over a year and finally culminated in the creation of what we call TNA, a modular streaming platform that leverages the best components of different streaming engines to support hundreds of live channels and thousands of concurrent viewers on a single server.

The TNA streaming architecture is the product of a rigorous R&D methodology that involved active testing and evaluation with a few of the larger OTT providers who use our services. One of these provided paid subscriptions to over one hundred live channels via an OTT Android box, with source streams originating from India, Pakistan, Nepal, and Japan. The subscriber-base was similarly global, with viewers as far away as Australia from our origin servers in Atlanta. At the time, they were not our customer yet, but were looking to switch from their current CDN due to the problems they had been having, which included reports of lengthy loading times and frequent stuttering from customers all over the world.

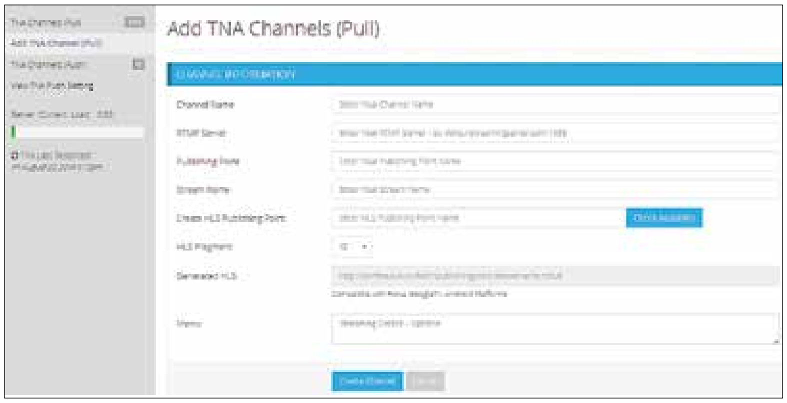


This customer promised us their business if we could prove to them that we were able to remedy their issues and provide them with a stable solution. We decided to use this opportunity as the first “field test” of TNA, and set up multiple channels on different platforms to test. Our customer had their representatives in different parts of the world watch these from their Android set-top-box and report all findings directly to us. The results wildly surpassed our expectations, with viewers reporting drastically lower loading times and an absence of any buffering issues from TNA. When testing from Atlanta to Australia, we received a report of average loading times below 2 seconds, five times faster than any individual server we had deployed until that point.

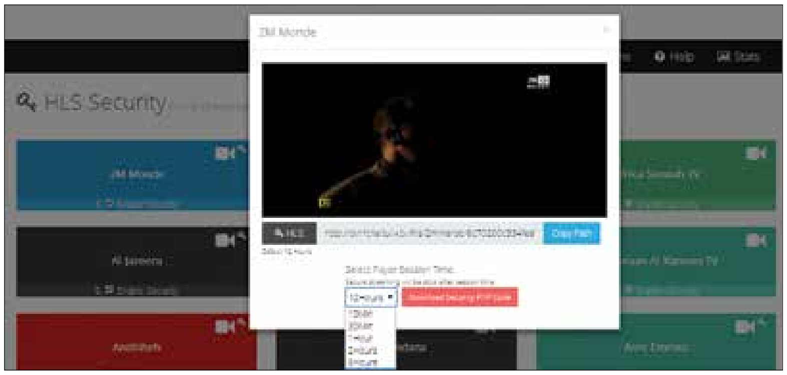
Following the success of this first deployment, we began inviting more of our customers to test TNA, initially migrating only a few channels to create a low-risk evaluation environment. We noticed that where other servers required restarts as often as weekly, TNA could run for weeks without interruption or reports of synching issues, buffering, or other problems. During our meticulous monitoring of the systems running

TULIX SYSTEMS

TNA lets users enable custom-length DVR archiving and create redundant stream paths from two or more sources.



TNA generates HLS from RTMP streams ingested from an encoder or other streaming servers.



Token authentication can be used to secure embedded streams with just a few clicks from the TNA control panel.

TNA during these evaluation periods, we noticed the lowest CPU, memory, and drive subsystem utilization we had ever witnessed. This efficiency had a very noticeable impact on the stability and quality of the channels being delivered from TNA, and we saw support requests regarding these streams drop precipitously relative to the channels originating from other platforms.

These beta installations were so successful that the customers running the trials soon requested a full migration of all channels to the new architecture. The system at that time still lacked a functional user interface and any useful statistical metrics beyond basic server traffic, so these requests were purely due to the very apparent superiority of TNA in terms of performance. Using TNA, we were finally able to use our 10Gbps servers to their full capacity, streaming hundreds of channels to thousands of concurrent viewers without the risk of encountering problems resulting from hardware overutilization.

The Evolution of TNA

As we migrated more customers to the new system, we began to receive requests for additional features. One of the first things we added was a graphical interface, so that our customers could create and manage their own streaming channels without having to involve our support staff. Thrilled with the great results we had managed to achieve with TNA in terms of performance, scalability, and stability, we began the process of enhancing its functionality to provide our customers with a greater level of control over the system.

Security

Many of our larger customers monetize and protect their content through controlled access, and requested stream security and the ability to easily implement geographic restrictions. To make this possible, we added two features: geoIP restrictions and token authentication. GeoIP blocking lets the platform admin configure access to specific channels by country (or locality, if desired) in just a few seconds. For example, if a TV broadcaster normally streams globally but only has the rights to stream a specific event in one country, they can restrict access to the stream to just that country for the desired period of time without having to fully black it out.

Token authentication can be used to secure access to a stream using a downloadable PHP script. Inserting this script into the code of a website or application validates the embedded stream. When a viewer visits an authorized website or application, a unique stream URL is generated for them, which cannot be shared, as it will only work on the device they visited on. It also expires after a customizable period of time, meaning the viewer will have to refresh the page or application to verify that they are an authorized user. This makes stream redistribution via unauthorized third parties impossible, ensuring that all traffic comes through affiliated websites and applications.

Big Data and Analysis

Accurate and detailed statistics are an essential component to a streaming platform, so we built an entirely new suite of analytics for the TNA platform, our most powerful statistics to-date. Users can see real-time and historical statistics for every channel, including unique views, a map of geographic distribution and connection speed by ISP, access by end-user device, and other metrics by customizable date ranges. TNA also tracks channel uptime, making it easy to pinpoint streams that are experiencing source stability issues.

Administration

Another functionality added was the ability to segment the TNA control panel between different sub-accounts, allowing customers with multiple source providers to give restricted access to each uplink location. This segmentation also means a server can be used as efficiently as possible by housing multiple moderate-use accounts.

Redundancy

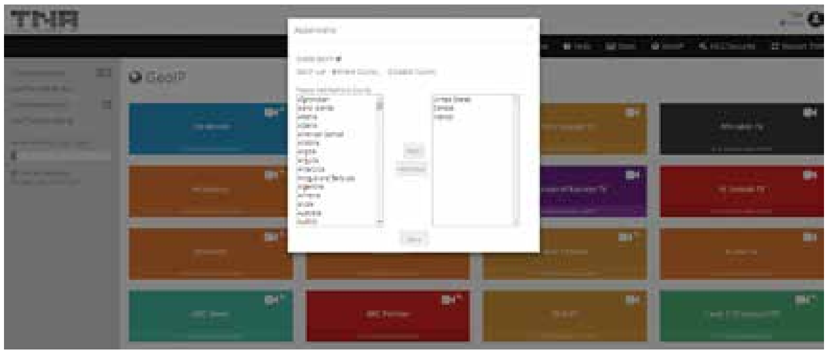
For those who need stream redundancy, TNA has a feature that creates a redundant streaming path from two separate but identical streams. So a content provider could use two separate servers to deliver the same stream. In the event that the primary stream fails, the system will automatically default to the back-up without the need to manually replace links on all websites and applications. The back-up channel does not need to come from TNA, and can originate from any third-party server that generates HLS video.

DVR Archiving

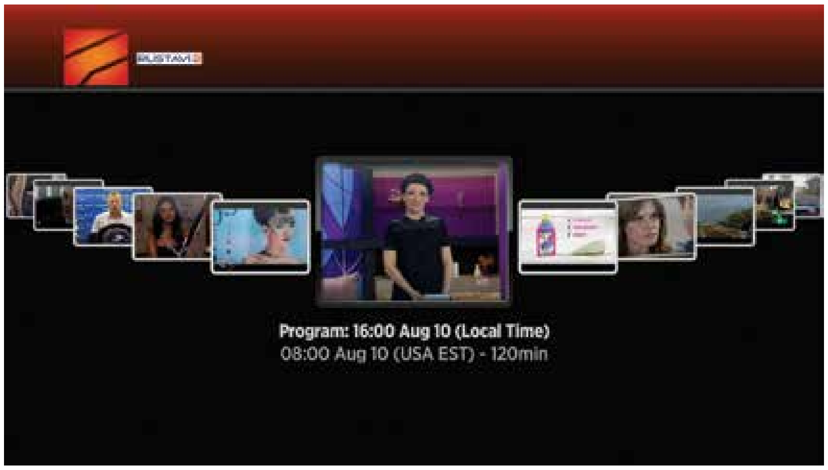
The most recently added feature is a DVR system that archives live feeds streamed through TNA for a customizable period of time. TNA will record a stream for the specified length in one hour chunks, discarding older files beyond the cut-off date and automatically replacing them with newer ones. The system automatically generates an embed code that can be used to make the playlist of recorded videos available on any website. It also provides an XML file that can be used to publish the archived videos on iOS, Android, Google TV, and Roku applications.

Built for the Future

Due to its modular nature, TNA is a highly versatile platform that can be customized using just about any open-source or commercial streaming software. For example, it has optional modules for the Wowza Streaming Engine™, Adobe Media Server™, live transcoding, and a number of open-source streaming and multimedia solutions. Thanks to its rapid loading time, it can also be used



Individual channels can be blocked by geographic location in an instant.


Program: 16:00 Aug 10 (Local Time)
08:00 Aug 10 (USA EST) - 120min

TNA's DVR archiving feature automatically generates an XML file that makes recorded videos available across web embeds and applications.

to stream globally from a single server or as a remote installation for different POPs. It can be used by content delivery networks or within closed networks operated by telecommunication companies, and MSO's.

Future plans for the continued evolution of TNA include the addition of crucial modules like ad-insertion, which will allow users to create rules for HLS ad-insertion using a web interface and target specific audiences based on the habits and profiles of viewers. Though we have long had a complete VOD system independent from TNA, we are adding VOD functionality and a cloud-playout system that will generate a linear stream from video assets directly uploaded to TNA.

Just about any component of the TNA architecture can be replaced, upgraded, and enhanced to meet the specific requirements of a project. For example, the system can easily be used for MPEG-DASH streaming with only minor modifications. With a highly intuitive user-interface, it can be used by anyone from first-time streamers to seasoned streaming veterans.

Started as an attempt to improve the quality of service for our customers, TNA is proving to be one of the fastest and most reliable, feature-rich streaming platforms on the market today. If necessity is the mother of invention, the future looks bright for TNA.