

BUILDING A GLOBAL STREAMING PLATFORM

The Challenges of Worldwide Content Distribution in the Last Decade

Contents

Introduction	3
What should the goal be here?	4
Early Challenges	5
Secure Playback and Multilingual Content Management	5
Infrastructure and Content Delivery Network	6
Entitlement – The Affiliate Integration	7
App Development	9
Platform Operation	10
Multi-Tenant App Development – Rise of the SDK	11
Global Audience Launch	12
Rollout Strategy	12
New Devices	13
Disclaimer	13
Battery Usage	14
Customer Support	15
Multi-territory	16
Content Rating System	18
Subtitle Challenges	18
Single Manifest vs. Multi Manifest	18
Currency Support	19
Multilingual UI	19
Event Management	20
Providing the best video quality	22
The Last Mile Problem: Do we have enough bandwidth?	22
Video Quality and Adaptation Logic	23
Adaptive Streaming Mysteries	23
Measuring the Performance	25
Download vs. Temporary Download	26
Bandwidth Save	27
Content Catalog	27
The Results	29
About Blue Guava	30

Introduction

Media and data streaming has become one of the most high-performing and lucrative industries of these first two decades of the 21st century. But 10 years ago, only a very few people might have guessed the heights that would become possible with large-scale content distribution in entertainment.

At Blue Guava, we were a group of people who saw that streaming is the goose that would lay the golden eggs. A world-renowned, market-leading player in the entertainment industry also realized the potential. Through a fortunate series of events, Blue Guava was contracted to co-develop one of the world's first global content distribution platforms. This e-book is mainly about the development of that streaming platform, its global launch, and the challenges we faced along the way.

By imparting the knowledge and best practices concerning the creation and operation of content distribution platforms through our specific examples, we hope to educate the next generation of pioneers, who will take the world of streaming to the next level and beyond.

Of course, we cannot mention the client or the service by name due to contractual obligations. Still, we can discuss the process of working and cooperating on this content distribution platform and generally lay down a roadmap or guideline for tackling the many issues that arise during development and deployment alike.

Without further ado, let's immerse ourselves in the past and rewind the clock to more than a decade ago, when streaming services were still just ambitious concepts in the minds of a few visionary creators.

What should the goal be here?

Building a digital **content distribution platform** is founded on a single premise: to evolve your company into a modern organization. It is the affirmation of the strategy by which you retain the core business. Still, by taking the company and its services from 1.0 to 2.0, you will open new distribution channels that genuinely make your business future-proof.

In general terms, a content distribution platform that you can scale globally allows you to enter the market in an entirely new way, followed by incredible performance within that same market.

Even if you already have a content distribution platform, taking the necessary steps to achieve the next level and go global will deliver significant benefits. For instance, you can enable the advantages of the new or upgraded platform in your distribution chain, which will result in an increased footprint of the content library for your customers. Furthermore, your teams will be capable of providing multi-audio, multi-subtitle, and multilingual UI and support in general — all of which are becoming everyday expectations for today's consumers.

Early Challenges

As mentioned above, we will explain the challenges of building and — in the second part — deploying a worldwide content distribution platform through the examples of our own experiences. When we helped a global, market-leading client create its platform, we picked up on the nuanced skills required to understand and replicate the development of such platforms.

You can read about the essential features and areas considered must-haves for any modern content distribution platform in the subsections below. At the same time, you will also get a glimpse of the challenges that arise during implementation. Fortunately, as we specifically had to tackle these issues when building the client's platform, we also provide turnkey solutions to avoid getting stuck on them.

SECURE PLAYBACK AND MULTILINGUAL CONTENT MANAGEMENT

When we started working with the client, it was still a time when **Digital Rights Management** platforms did not exist on mobile devices. One of the earliest challenges was finding the best **video player** that could provide protected playback as well. We chose Microsoft PlayReady and were quite satisfied with what it could provide in terms of media file copy prevention, encryption, output prevention, and of course, DRM.

After the basics of security and content protection were in place, we looked at the road ahead. It is interesting to go down memory lane and remember these challenges, because none of the current high-tech solutions were available back then.

None of the adaptive streaming support methods of the present existed. So we used Microsoft Smooth Streaming and adapted it to meet our unique requirements.

There was still no multi-audio or multi-subtitle support available. Instead, we developed custom transcoders to provide this functionality.

Adaptive livestream support was still just a concept in the industry, and we had to wait for the first commercial product to hit the shelves. Then we could use it to integrate with the system.

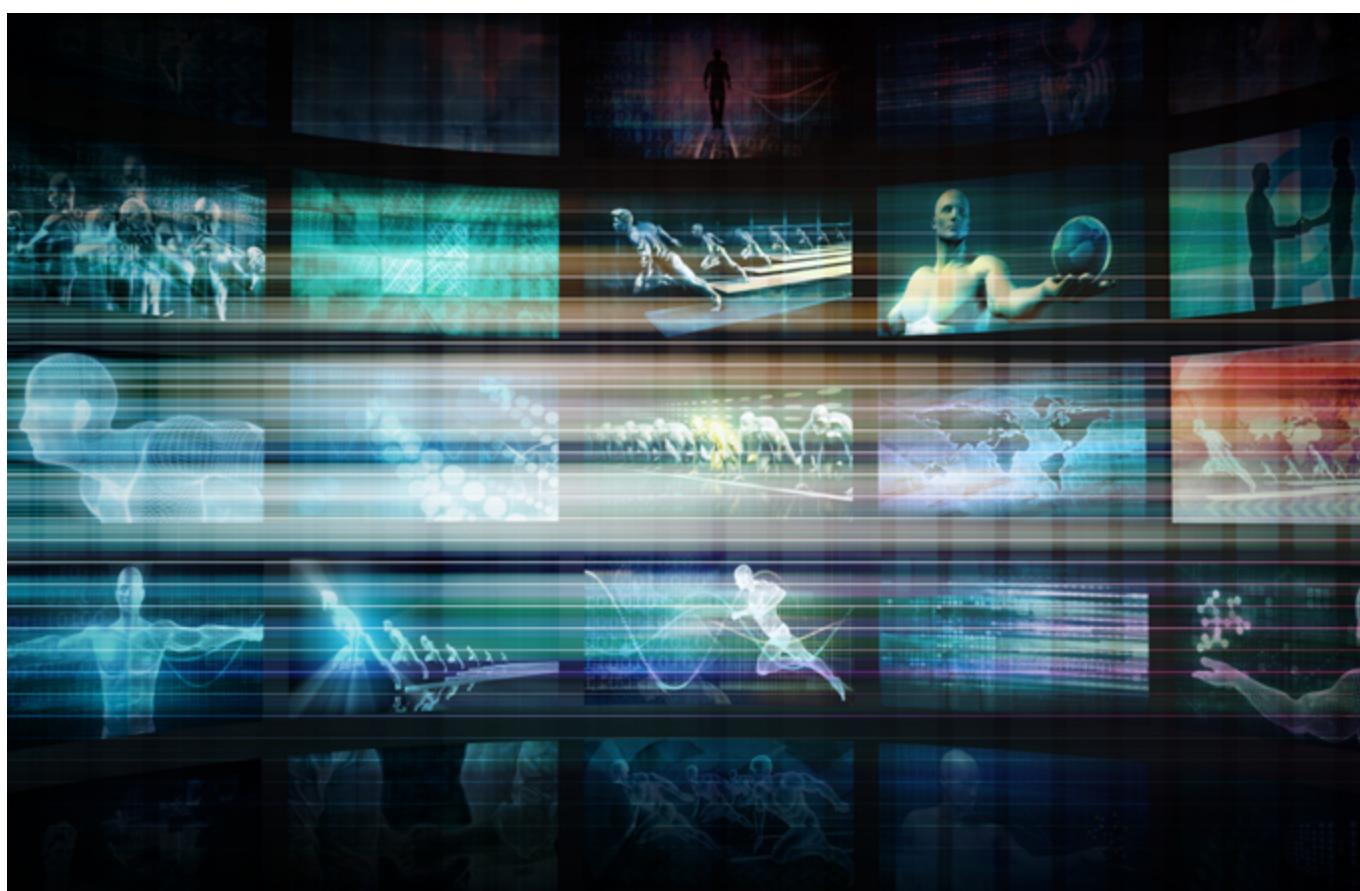
As you can see, it was still a time when most of the now widely available products needed to build a functioning and successful content distribution platform were either in development or still just dreams in the minds of their creators. Yet, we adopted tools that were already publicly available and adapted them to a specific project we were working on. You could also choose this route when developing your platform. Still, most of the time, you can get all these **essential content management features** (DRM, adaptive streaming support, multi-audio / multi-subtitle support, livestream support) either in a single package or by picking the best tools for a more customized approach.

INFRASTRUCTURE AND CONTENT DELIVERY NETWORK

We faced similar challenges when it came to the network infrastructure that we had with content management. Namely, much of the current, widespread, easy-to-use technology was not out there yet.

For instance, cloud providers were just established at the time, taking their first steps toward the heights they have reached today. However, the cloud's early versions were still not enough for the goals we were working on (nevertheless, it's important to note that since then, storing and distributing data and services in and through the cloud have not only become widely popular, but significantly more feasible as well).

With no robust cloud solutions at our disposal, we faced **CAPEX** vs. **OPEX** financial issues. Should we build a platform infrastructure that needs constant upgrading and maintenance with related expenditures or focus on a system that only requires major operational expenses? Or maybe a hybrid between the two?



In the end, however, we came up with a smart design that resulted in a cost-efficient yet streamlined and robust solution.

As we are generally talking about a wide area network (WAN) when implementing content distribution that extends over large geographic areas, it's best if the design's topology matches our needs and goals. We found both the L2 star topology and the L3 ring networks to serve this purpose well for a private WAN.

Without the cloud, we contacted the best and most cost-effective, neutral data centers and used them to store the distributable content. Of course, it then required precise planning on our part for developing deployment strategies and release schedules that could work with the data centers' infrastructure.

The **last mile issue** we faced was the lack of a content delivery network (CDN) in many regions where we wanted to distribute the content. Or, if there was a CDN, it was usually so far from the end user that the bandwidth's weakness has become intolerable.

What was the solution? We built a **custom CDN** together with the client. Not the cheapest method, but in the end, the ROI and the advantages in content distribution made it worthwhile. Of course, today, it's much easier to use available CDNs (Multi CDN, eCDN, or P2P CDN), as they have also become more open.

ENTITLEMENT – THE AFFILIATE INTEGRATION

Explaining **TV Everywhere** is easy – today. Even if someone has not heard the concept by name, when you tell them what it means, it will usually click in their mind in an instant as they think of their favorite streaming service providers and the shows they watch on their TV, desktop PC, laptop, tablet, and/or phone. Back in the day, when we were working on one such streaming service, it was more challenging to explain it to the affiliates of television providers who would benefit from the TVE solution.

In a nutshell, **TV Everywhere** (a.k.a. authenticated streaming or authenticated video-on-demand (SVOD)) is a **subscription-based business model** that provides access to streaming video content from a television channel. It requires users to authenticate themselves as current subscribers to the channel before accessing the content, usually done via an account registered to the pay TV provider. Customers then gain access to the broadcasters' regular channel content through internet-based services and mobile apps, live or on demand.



Again, TVE is a hugely popular business model across the world today. Still, as the idea only surfaced in 2009, with the first rollouts starting in 2010, you can imagine that there were concerns and confusion about the concept itself, especially among the TV providers' affiliates who sometimes didn't know what to make of it.

So, the first step was education. Along with the client, we had to present the business case and prove that this would be an incredibly profitable service. The process went reasonably well, even if it took some time. However, the technical challenges were yet to come.

As each affiliate had a **Customer Relationship Management** (CRM) system in place, each of them needed a unique integration with the client's global content distribution platform. It was challenging to manage, even within a single country, but imagine expanding it to more than 50 nations across three continents, with unique constellations of culture, language, and law. It looked like an impossible task at first.

Of course, we are always excited by the prospect of achieving the impossible, and this was no exception. The idea was to deliver an easily implementable and simplified solution that each affiliate could customize to their liking. As such, we created a universal sandbox environment where they could pick up the basics and work out their integration's specifics independently.

There's a saying that has become our creed here at Blue Guava: If you want to be a true pioneer in what you are doing, you cannot just copy proven, existing solutions!

This creed was always in the front of our minds when working on the client's platform. This was to be one of the first of its kind. We had to prove that we could achieve something that has not truly existed before.

This mindset was also carried over to how we approached the entirety of the development life cycle, where it was translated into optimizing the code to achieve high performance levels for the best experiences.

Of course, this approach has brought about its own set of problems. From device performance issues like with Android 2.1 or the first iPad memory limitation to multi-resolution design challenges, there seemed always to be another obstacle to be surmounted. Indeed, the ideal state of perfection is unattainable. Still, we made sure to get as close to it as physically possible by developing, iterating, and polishing the client's platform as much as we could before rollout.

There are some general ideas, tips, and tricks we want to share with you. These helped us overall throughout the development. We hope that they can also provide both guidance and assistance as you begin building — or upgrading — your global content distribution platform.

- The first crucial step is to **fully understand the product's core functionality**. This is the foundation — the very heart — of the platform. If you and your development team know it like the back of their hand, the development will stay on the correct course throughout its life cycle.
- In general, the addition of **streaming support to the device is essential**. So is the implementation of a multilingual UI that supports Latin, Cyrillic, and Chinese characters all at once. A good idea here is finding and working with a font to help in as many languages as possible.
- **Remote app management** will become one of your go-to tools during development. It is a precious resource that can help accelerate most processes. Also, be prepared for the insanity that is app certification. Validation can range from between 1–2 weeks to 2–3 months!

In the case of operational tasks, two primary considerations are vital to the overall project. One is to define the level of service quality that is to be provided. This sounds simple, but it is one of the most effective ways to ensure that operations are aligned with the right goals and methods to deliver the best performance.

The other consideration is the definition of work. In every operation, defining just precisely how such a content distribution platform works is always a crucial step. Which KPIs can help determine if it works or not? Furthermore, this isn't purely about the subsystems that run in the back, as it has more to do with how the end users experience the app — whether they can use it as intended or not. We usually look at the combined operation of several parameters and use that data to decide whether the app is functioning well or if there are issues — and which subsystems those issues affect.

Monitoring the execution of operational tasks and their results becomes an essential part of managing operations. Choosing an app to carry out the monitoring depends on what you want to track and analyze. Today, apps like Adobe, RayGun, Conviva, Grafana, and Prometheus are some of the best go-to tools on the market for this purpose.

Grafana is a multi-platform, open-source analytics and interactive visualization web app that helps users understand complex data via data metrics, while Prometheus is an open-source event monitoring and alerting tool that stores data locally, generates alerts, and establishes rules over data — quite handy when you need to monitor multiple microservices at once.

For us, managing operations within the **CDN** of a global content distribution came with some unforeseen, but educational challenges.

One of these challenges is a pretty common one that, unfortunately, is inevitable. **Cold start** problems cannot be avoided, but can be mitigated. After booting up any web service system, things have to warm up to reach operational capacity. You can do that by incrementally increasing the server's load, populating the cache, or requesting new resources. In case it is crucial for the system to work according to its peak capacity for a more extended period, **pre-warming** the system is a must. Otherwise, the cold start will kick in when the system is being strained to its limits.

Other, less well-known CDN-related issues that you should look out for are secret routes and redirections. In the case of secret routes, as CDNs have their own sets of weird rules, we have encountered some cases where a CDN's strange routing led to the downgrading of playback quality. As for the other issue, while most CDNs operate based on DNS, some subpar models out there can only route requests through **HTTP redirections**, which are **unsupported** by video players. It was quite a challenge to find workaround solutions to these issues and build them into the platform.

MULTI-TENANT APP DEVELOPMENT – RISE OF THE SDK

A brief, but important, area that must be discussed is using a **software development kit** (SDK).

In and of itself, an SDK can be a great boon when getting started on any development project, as it has all the necessary development tools in a ready-to-go shape within a single installable package. However, as they are specifically related to a hardware platform and operating system combination, it becomes increasingly challenging to design and test the app on all the devices you want to release.

This is because an SDK is more or less designed to develop sample apps; if you want to build the real application that your customers will use, it's not going to be enough. They are challenging to integrate with most platforms, which leads to additional complications that can simply be avoided by choosing custom development methodologies.

The real benefit to this approach is the faster **integration time**.

The developer team can bring the existing functionality away faster, like building everything from the ground up. Save time and reuse what's already working is the key message!



Global Audience Launch

ROLLOUT STRATEGY

As with all things in business, it is paramount to have a comprehensive, actionable strategy in place for the time you begin the deployment of your content distribution platform. Even if the work that went into developing it resulted in a spectacular product with great functionalities and amazing UX, almost just as much depends on how the rollout is implemented.

Product launches are essential to the product's success, but are also one of the business investments that carries the highest risk. Generally, if you can home in on certain key areas, such as the formulation of a proper launch plan, establishing a cross-functional launch team, and real-time tracking progress (measured against financial or schedule goals), you have already increased your chances for a successful rollout. Leveraging past successes (e.g., tried and proven processes) and filling in experience gaps with consultants outside your company will also go a long way toward ensuring a great launch.

These best practices are also applicable to the deployment of a streaming service. However, the challenges there are unique. As such, here we have compiled a checklist for those areas, steps, and considerations that every rollout strategy should contain.

- **Infrastructure Deployment**
- **Content Library Setup** — with emphasis on Territorial Content Rights, Audio and Subtitle Rules, and Live Feed Setup.
- **CDN Deployment** (if needed) — be prepared for the fact that switching CDNs or applying a new CDN is always a significant challenge.
- **Application Localization**
- **Affiliate Integration**
- **Application Certification** for the new market. You may have to calculate a longer certification time. We had this issue in the cases of Samsung and Roku.
- **Analytics System Configuration**
- **SKU Configuration** for in-app processes.
- **Time Zone Issues** (UTC vs. Local) — if the app covers multiple time zones, decide which should be the reference time.
- **General Availability** (GA) Launch — specifically focusing on questions of when and how to monitor activities.

We have had to deal with all these processes as we approached the launch of the client's platform. While some of these were incorporated early into our strategy, we could have had a much more relaxed time during rollout had we considered all of them in advance.



NEW DEVICES

One unfortunate aspect will always be a constant in any global launch: developers, operators, and providers should expect and calculate with suboptimal technical support for streaming from device producers — before, during, and after deployment.

In the table below, we have collected the major device families we tested our partner's content distribution platform on and the lack of support or other limitations that we encountered during development within each device group.

DISCLAIMER

This was the case when we were working on the platform 10 years ago. We are not saying that these issues remain — on the contrary, some of them have been remedied. However, as these were the device-related challenges of the time, we wanted to use them as examples. Today, developers would encounter a whole new set of technical limitations and devices when building a global content distribution platform.

Device	Limitations
Smart TV / Gaming Consoles	<ul style="list-style-type: none"> • Supported only the proper order of tracks in the Smooth Streaming manifest • No support for Live Smooth Streaming • No support for Dolby Surround content on many models • Higher crash ratio on some new models • Apps have to adjust for new OS or devices • Different chipset and device firmware/region
TV extenders	<ul style="list-style-type: none"> • No support for Live Smooth Streaming • Random issues • Silent firmware upgrades which cause production problems
Mobile Devices	<ul style="list-style-type: none"> • Weird ABR implementation on the player side • No support of multilanguage audio • No support for two different audio codecs in one manifest • Do not support redirect type of CDN
Set-top boxes	<ul style="list-style-type: none"> • Memory leak issues • Limited size response from the server • Poorly implemented network stack • No support for HTTPS streaming

BATTERY USAGE

On top of it all, there was the matter of battery settings. Some countries and phone producers operate with built-in processes that optimize battery consumption. They might kick in when they determine the battery is low, and if you are watching a movie when this happens, the video performance could be severely downgraded, ruining the experience. This is the case with many laptops programmed to optimize battery usage when there's a video playback in the HD. In this case, the player cannot scale up to the highest quality, because that will consume more energy from the battery. The result? A lower playback than what is expected.

On the other hand, watching a video in broadcast quality can take its toll on the battery charge. As such, the challenge was to find the balance between the two and ensure good performance and quality without draining the battery. The result should be a satisfied consumer whose phone doesn't shut down and who can watch the movie from beginning to end in a presentable resolution.

When providing impeccable customer support during and after the launch of a global content distribution platform, one of the first and most important steps is to discover a way to isolate and categorize issues. After all, depending on the size of the customer base, hundreds, maybe thousands of support tickets may come in on a single day.

We found that a straightforward solution — and possibly many others will agree with this — was to group and filter issues based on country, platform, content, and the CDN used for distribution.

It's also necessary to have a **clear vision of what constitutes an issue** or error. Therefore, we recommend building one knowledge base for all known issues and a separate one for instances that are not issues but were reported as such. This way, you can see the ratio of problems to nonissues and see how big each reported issue is.

Another good piece of advice we have for those who deliver a client's content distribution platform: understand that their VP and any other C-level officers are all users first, stakeholders second. If they know the services and platform behavior and then use them accordingly, they will provide significant business insights and “function” as early testers in need of customer support. As we learned from lessons of our own making, they should also always be informed of an upcoming marketing event!

We have also collected several general tips and tricks (also gained from lessons learned) that are going to be helpful at all times when providing customer support:



- If an issue appears in the top country, it does not necessarily mean that it is present only there.
- After deployment, make sure to check logs and performance metrics periodically.
- Operation teams must inform about the marketing events.
- Understand the difference between unexpected behavior vs. a real bug.
- Be prepared that an issue could stay dormant during deployment and emerge later!
- You are not the only one who's doing a deployment at the same time! The infrastructure, network, or partner affiliates also make changes without any notice.
- Remain wary of app level caches during and after deployment as well.
- Focus on managing the correct error messages.



MULTI-TERRITORY

The multi-territory aspect to a global launch depends on four considerations: language, culture (or law), content release strategy, and time zone.

The last one is easy to define, but sometimes easy to forget — and it is crucial to map the content releases to the **correct time zones!**

The planning and implementation of content release strategies are related to the unique challenges we encounter within each country and the constellation of their linguistic, cultural, and legal concerns.

For instance, approaching content distribution in Latin America or Asia does not mean that you have to stream in additional countries now. Culture, language, and law all come into play.

The Brazilian LGPD is different from the European GDPR, just as their content rating systems also differ. In Latin American countries, subtitle support is mandatory, and subtitles must be colored.

Some Asian countries censor certain content, and they also have different rating systems in place. Even in Europe, with a unified GDPR and content rating system, there is a content library for each country because of its other subtitles.

Once we dealt with these obstacles and harmonized each content library with local laws and language, we could finally work on the actual content release strategy. As we have deployed this platform in three localized versions in Asia, Europe, and Latin America, we have identified the key elements that were crucial to our success when releasing in each of those regions.

	ASIA	EUROPE	LATIN AMERICA
CDN	Single	Multi	Multi
CDN type	Push	Push/Pull	Push/Pull
Manifest	Per territory	Per territory	Single
Rating	Censorship MPAA	Age-based	MPAA Brazil with advisory
Localization	Original audio track with subtitle	Original audio track with subtitle Dubbing Voice-over in a few countries	Original audio track with subtitles Dubbing Colored subtitle

Content rating systems vary from country to country. When compiling a content library and designing how that content will be distributed across multiple countries, checking in with rating agencies in those countries is a step that simply cannot be skipped.

The fact that each country works with different rating standards makes organizing content even more complicated. For instance, the American Motion Picture Association of America's film rating system is different from the European standardized rating, while the Brazilian rating system differs vastly from both.

If there are content aggregators involved, you should also have to conform to their rating systems.

We had to face scores of difficulties when it came down to a specific country or region's preferences, language, or even currency.

The first challenge is possibly the easiest to imagine: just consider the problematic process of synchronizing video FPS (frames per second) with the subtitle's FPS. Because yes, most of our cultural hardships during any launch stem from making sure that the subtitles are in place and according to the country's standards and expectations (e.g., mandatory support of colored subtitles in Latin America). On the other hand, all content doesn't need to use the same FPS.

Usually, there is an app manifest language list on your app in any typical case: the list of languages for which your app has declared support. However, in the case of global content distribution platforms where you have several highly localized manifests and, preferably, a global one, you need to think of a traveling user who flies across countries, entering and leaving service zones that show them these localized versions. The option to pick manifest preferences thus becomes another feature to implement as you tackle this problem.

Local currency support boils down to two questions: where to place the thousands separator, and are decimals separated by a dot or a comma? For example, in Costa Rica, they write "1.000,00," while in the U.S., it's "1,000.00." Dot for the thousands, comma for the decimals vs. comma for the thousands and dot for the decimals. Similarly, the matter of presenting prices can be just as frustrating. Believe it or not, having the app display \$10 can be confusing. Writing that could mean either Mexican pesos or U.S. dollars. You have to add the official currency abbreviation to the price: \$10 MXN and \$10 USD.

Finally, as developers, how can you validate the Chinese localization when you don't understand the language and have no Chinese-speaking testers or translators at hand? We usually solve this issue by playing around with the menus' positions, checking which English menu is aligned with Chinese. Thanks to the fact that you can search for content in two languages at once, we usually found what we were looking for in the end.



During launch or any other major event on the platform (e.g., the premiere of a show), there are some vital, mandatory steps that you should take to be prepared for the onslaught that is the waves of users crashing down on your platform. We have faced such events numerous times with the client's streaming platform — once at the start of the final season of one of the most-watched TV shows of all time.

First and foremost, for any successful event management, **load tests** are a must — the more before the event, the merrier. This will require several specified components, including an end-to-end (E2E) method for simulating user behavior.

DDOS protection is another compulsory element, but what is just as important (but often overlooked) is the application's behavior when put under a lot of stress. Ensure that you check how it responds to massive traffic, because if it's not handled well, the wrong responses can lead to a series of frustrating problems.

As for the content, the system, and user behavior in general, we have collected our best practices in the table below:

CONTENT

Fastlane the content — put it to the primary position and remove any unimportant item. Release it a bit earlier to ensure that it is all good (the right content items are available, nothing restricted was released, everything behaves as expected, etc.)

PLATFORM

Prewarm the system to avoid cold start issues.
Turn off all unimportant functionality at the top level.
Apply higher cache capacity during the event.

USER

User behavior could be different from country to country.
A little research can always help, but the idea is not to generalize and apply the same patterns to the audiences of other countries.
Reduce the user validation steps with a smart cache solution.

If you do these steps right, be ready; you will still run into issues as the event unfolds, but that is perfectly normal.



Consider this: when the launch of that premiere we had been working on started, our cloud service provider contacted us to ask if everything was all right. They described these events not as singular spikes in traffic, but as walls, so massive was the surge of users and their queries.

Despite these walls crashing down on us and the distribution platform, we persevered. But it could only happen because we followed a strict set of rules that gave us the edge to deal with this incredible traffic level:

- 1 **System monitoring** from as many different aspects as possible.
- 2 **Resource dedication** on all platform providers, including CDN, Global Firewall, IaaS providers, DRM provider, Analytics, and user-based third-party systems.
- 3 **A proper CDN** — whether CDN Load Balancing, Multi-CDN, or P2P CDN — is vital if you want to weather these events.
- 4 **Prewarming the CDN** — again, to avoid cold starts.

One more consideration to keep in mind is that no CDN will behave the same way in each country, so expect mild to extreme discrepancies in how the CDN communicates with end users and vice versa. Also, some countries will surprise you with the amount of traffic they can generate on a whim. In a way, they are like sleeping dragons that wake up once they get a sense of an ongoing feast of new content being streamed.

The demand for content has never been greater than it is today, and every research, survey, and analysis predicts that it will increase at an incredible rate. And with the demand for content surging, so will traffic increase across networks.

As such, operators need to leverage their network investments by gaining more control over the flow of content across their network infrastructure. However, that raises more questions than answers: how to meet high expectations for quality of experience? How to sustain capacity during high traffic and peak hours? What will happen to network costs?

Enhanced CDN caching strategy could provide an excellent solution for such problems, ensuring reduced costs of content explosion, the evolution of existing services, and the development of new services to generate revenue.

In this regard, we can talk about two interesting use cases for caching. The first is about the volume of video delivered over the last mile to customers. The solution is improved quality of service, monetization of the delivered video content, and cost control.

The second use case suggests adding CDN services and expanding the portfolio with products in areas adjacent to services they already render on their network.

However, what all of this boils down to is that ISPs are still very much in charge of the last mile problem itself. As such, the options are either using their services or seeking a custom CDN solution.



The other consideration in this area is concerning seamless video playback.

Every service provider tried to reach the highest quality playback on every device. Then the question began revolving around how sound could become the adaptation logic for multilevel streaming.

Many players on the market have multiple strategies to achieve that and keep it during the various conditions.

Some people might say that it isn't a question in the current Wi-Fi/5G world where everyone has a reliable internet connection at home. Yes, that is true, primarily across Europe, but less so in other parts of the world where users use mobile internet to access these services. The higher variant of mobile services requires a more sophisticated setting. Also, it has to understand the high-resolution Android devices equipped with CPU or GPU units of lower performance in video processing.

It is also crucial to keep the seamless playback more essential. Instead of jumping to the highest quality and facing slow, erratic playback of the content because of buffering every 10 seconds, opt for scaling down to a lower rate, making the customer experience smoother.

As the title of this section suggests, one of the most problematic questions is how well the content is delivered in the last mile. Will your end users have enough bandwidth to enjoy the delivered content?

This is where a technique called adaptive streaming or adaptive bitrate streaming can save the day. It is a method for streaming multimedia over computer networks. Instead of using RTP or RTSP protocols for video or audio streaming, it is almost exclusively HTTP-based – which means it is capable of working efficiently over large HTTP networks like the internet.

Its ingenuity lies in the ability to detect the user's bandwidth and CPU capacity in real time and adjust the quality of the media stream accordingly by encoding a single source media at multiple bit rates. During streaming, the available resources ebb and flow, and the player client can switch among the different encodings to match those resources. This is a must-have solution when you want to provide your customers with benefits such as fast start time, a small amount of buffering, and an overall excellent user experience on both high-end and low-end connections.

When providing support for the client in adaptive streaming techniques, we didn't face many issues on the content level. Adding the necessary settings to make adaptive streaming adjustable is one thing. However, concerning the player side of things, it takes a bit of time to figure out all the nuances.

In the end, we needed a player that could measure CPU usage, frame drops, available screen size, current screen size, and player visibility for optimal performance. This design decision's driving principle was a realistic scenario: imagine you only want to listen to the content and the app itself is not visible. Why keep streaming HD quality video when you're not even watching? We wanted a smart player that monitors the metrics above and adjusts video quality accordingly.

The Silverlight player could provide all these functionalities and even yield a highly efficient adaptation logic.



In the mobile device and Smart TV capability, mostly the bandwidth-based adaptation provides a way to play with adaptive streaming adjustments. As for the HTML5 player, which is also bandwidth-based, it scales to the very top and stays there without budging.

05

MEASURING THE PERFORMANCE

With any such operation, it gathers data about performance and analyzes it, establishing which areas perform well and which require improvements. You can develop your solution for this purpose, but some great service providers have already established themselves as market leaders in streaming analytics.

We also searched for a provider whose platform could be integrated with the client's content distribution platform. As we looked at the options available, one such provider emerged. This was Conviva, the Google Analytics of the streaming world.

As Conviva entered the picture with the largest streaming video sensor network, our task of providing a comprehensive analytics layer for the content distribution platform became easier. Their integrated streaming intelligence platform worked like a charm and we could monitor performance and analyze user, bandwidth, and traffic data simultaneously in real time. With their AI-driven solutions, it became possible to have real-time, cross-screen insights about virtually anything that occurred during any initiated streaming. We could also see how the quality of experience affected content consumption. All this gave our clients a competitive advantage that allowed them to optimize every operation based on data, which in turn led to better content selection, reduced customer churn, increased viewer engagement, and an ever-growing ROI for the client — and that is just the tip of the iceberg in guaranteed benefits.

All in all, even if you opt for a different analytics solution to measure performance, the most important takeaway here is to understand your users and their behaviors — and that requires a functioning analytics system to gather that Big Data. Taking it a step further, if your design is based on the users' data and how they consume content, the better the results will be across all facets of the content distribution platform.



DOWNLOAD VS. TEMPORARY DOWNLOAD

Providing the ability to consume content offline to your customers is a common expectation. For example, they may be traveling to a remote location where they might not have enough bandwidth or the necessary amount of mobile data to stream broadcast-quality content. They download what they want to watch back home in the desired resolution and take that content along for the trip.

However, downloading an entire movie, especially in broadcast quality or something close to it, takes a lot of time and will consume a lot of storage space on the computer. As such, you have to be flexible in the options you give to your customers when they are not streaming and are going for offline playback. The idea is to provide the same user experience that they would typically have when streaming.

In this case, it's usually best if both regular and progressive types of download are available for the user. With the latter, they can begin playback of the content before the download is complete. The difference between that and streaming is that the media file is received and stored on the end user's device (usually in a temporary directory). This solution is preferable for many reasons, but it is also acceptable for most studios that lease their content to streaming services, as they would never allow a direct, permanent download of their property.

Even then, a **temporary download** is usually only available if the user accepts a **rigorous set of rules and restrictions** that come with it. For instance, most often, there is a cap on the number of devices that a single account can function on and on the number of downloads allowed this way. Such a rule system could be that a licensed film may only be temporarily downloaded and viewable on a maximum of two approved devices at any one time and that no more than 10 feature-length films in the aggregate across all content may be temporarily downloaded to a subscriber account at any one time.

Furthermore, additional restrictions are rental periods for the licensed content and expiration after the first playback (e.g., 48 hours after the content has been temporarily downloaded). All in all, be sure to count on making the necessary coding and preparations for these restrictions, as these are considered standard procedures among studios, distributors, and affiliates alike.

BANDWIDTH SAVE

It's also worth mentioning that end users also want bandwidth or data saver as a feature, similar to what is available in Spotify. Specifically, it comes with the option to pick the streaming quality that best suits your current bandwidth or data capacities (opting for mobile data when Wi-Fi isn't available). The ability to save the streamed content in the cache to stream later will use less data.

Bandwidth saver also helps avoid going over your monthly data limit by automatically halting streaming after four hours of inactivity.

Also understand that studios allow only limited options, as they put severe restrictions on several areas that matter the most to customers, such as limited download attempts, quality, devices, and accessibility after the first playback attempt.

CONTENT CATALOG

To create a structured and vast content library composed of thousands of titles, the main challenge is to present all that content at the right time to the right customers, each of whom have unique preferences and interests.

As the main challenge here is devising and implementing the content structure that best suits the audience, we faced a dilemma early on: whether we should use the same structure everywhere, but with a different design, or have other structures but with the same design.

In this case, the solution always boils down to a single attribute: whether it is **data-driven** or not. Using data as a basis of both structure and design when building your content library will solve many issues. It is more intuitive to develop, but it is intuitive to use and enjoy when browsing for the desired content.

Apropos of browsing, why let your customers worry about finding what to stream next? Instead, let an intelligent **recommendation engine** do the hard work for them. Indeed, the recommendation is also a must-have feature for any content distribution platform. It's both a sign of customer-centric design and proof that you have a knack for developing and delivering personalized experiences.

A great recommendation engine has to be data-driven as well to display the right content for everyone. Take a look at Netflix's example; according to its data, “more than 80% of Netflix shows that customers watched in the last two years have been as a direct result of Netflix's recommendation engine.”

Netflix's engine recommends titles that seem to have nothing in common with what you watched. But its algorithm goes beyond genre, identifying characters, plot, actors, and/or directors that are similar or related to what you just watched.

In summary, be data-driven and customer-centric in your content catalog and recommendation engine solutions and focus on personalization! The results will be fantastic — trust us, we have witnessed them.



THE RESULTS

What did all our hard work amount to? Take a look at the table below, and you will see how much you can achieve if you make the right decisions during the development and rollout of your content streaming platform.

AVAILABILITY	50+ countries across three continents
NO. OF PRODUCT BRANDS	2
LANGUAGE SUPPORT	17 (including Latin, Cyrillic, and Chinese alphabets)
VIDEO QUALITY SUPPORT	SD/HD/4K UHD + HDCP support
AFFILIATE INTEGRATION	150+
IN-APP INTEGRATION	iTunes, Google Play, Roku Pay, and Samsung Checkout
AUTHENTICATION TYPE	SOAP, REST, OAuth, and SAML support
PLATFORM SUPPORT	15 devices
HOURS OF CONTENT	20,000+
ONLINE / LIVE / OFFLINE PLAYBACK SUPPORT	Enabled
TVE / OTT / DIRECT TO CUSTOMER	Enabled

What we achieved with this streaming service has been one of the most excellent rides of our lifetimes. The solutions outlined in the previous sections enabled us to pull off the support for one of the biggest TV premieres in 2019 and implement all significant adaptive streaming and CDN support capabilities.

About Blue Guava

Our goal is to become the best long-term partner that any of our clients could wish for. With more than 10 years of state-of-the-art software development, streaming, and testing solutions, we have helped market-leader partners increase their revenue and the efficiency of their IT operations while cutting costs and time. Simultaneously, the software products we developed for them streamlined and optimized the streaming experience for millions of their customers across more than 50 countries on three continents.

At Blue Guava, we believe in exceptional customer service. Our passion is to provide our clients with nothing but the highest quality services that are guaranteed to meet their needs and help them in their quest to produce excellent software solutions.

Our content delivery, content management software solutions, and quality assurance services will help you maximize customer engagement, ultimately empowering your business's customer adoption and retention capabilities.

[Contact Us](#)