BROADCAST THE BRIDGE
Connecting IT to Broadcast

# Delivering High Availability Cloud

# Essential Guide

EG

ESSENTIAL GUIDES

# Introduction

By Tony Orme, Editor at The Broadcast Bridge

**It may seem strange, that after spending so much time delivering reliable broadcast infrastructures that we should even consider accepting failure as a principle. Traditionally, broadcasters have always assumed they can engineer-out failure and make broadcast workflows so reliable, that they were virtually bullet-proof. And this seemingly minor caveat, that is "virtually bullet-proof" lies at the heart of modern IT infrastructure design.**

No matter how hard we try, we will never make a broadcast workflow 100% reliable. It's fair to say that we will get very close, even to the point of eleven nines, but achieving 100% reliability is an impossibility. How many times has something gone wrong that "shouldn't have done", or a completely unexpected event has occurred without any warning whatsoever? Probably more times than we care to imagine. And it always seems to be the failures that we least expect have the highest impact on the transmission.

Attempting to engineer-out every fault or potential failure is almost a lesson in denial. With such complex systems, where we have so little control over so many components and sub processes, requires a new way of thinking which takes a more pragmatic and realistic view of failure. DevOps and modern IT working practices focus on the agile methodology which assumes there will be failure, but the good news is that we can measure it, and plan for it, especially when working in cloud environments.

Cloud service vendors provide reliability times where the uptime of a system can be calculated with very high accuracy. These uptime measurements provide a starting point for the broadcasters and their partners so that they can evaluate where the highest likelihood of failure is likely to occur. This further provides measures on resilience allowing broadcasters to continuously improve their designs and hence reliability.

Designing for resilience is key for any broadcaster looking to build infrastructures that are as reliable as they can possibly be. This does not only encapsulate the physical resource architecture, but also encourages the need to build highly effective monitoring systems from the outset. Monitoring, and the associated data gathering is critical for any broadcast IP facility as it allows broadcasters to detect patterns where failure is likely to occur, and then plan for it and provide the necessary remedy.

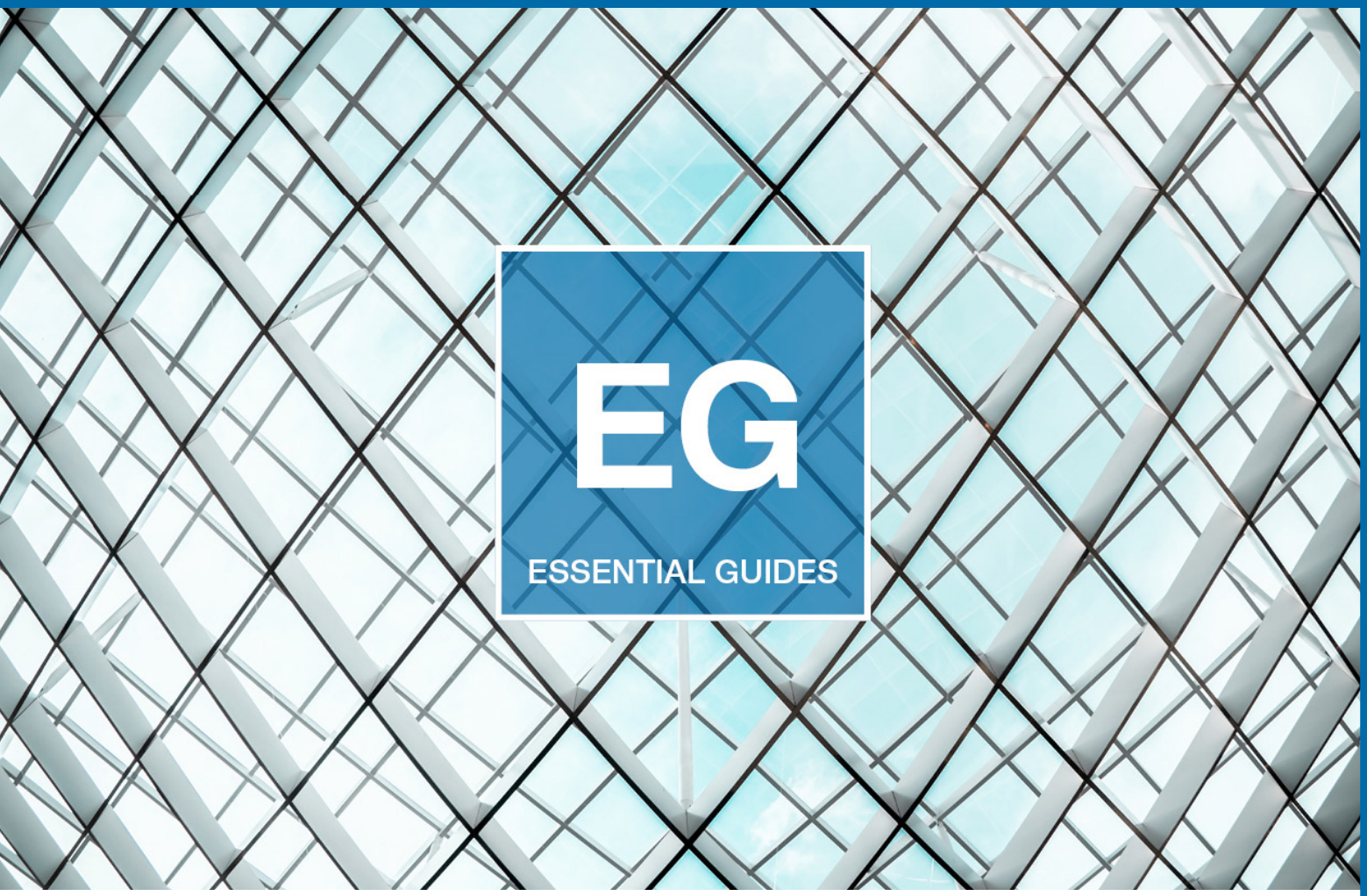
There is also the commercial element of dealing with failure to consider. For example, it may be possible to improve the reliability of a workflow by 0.003% using triple parallel redundancy, but this would be at a cost of $150K. Is this the right thing to do? It depends, because at this point, the decision falls under the domain of the CEO. As engineers we will present the data to the CEO, but as the CEO is heavily focused on risk-management, it is they who should make this decision.

Delivering high availability cloud is more than just an engineering methodology, it is a complete discipline that makes demands on the whole business, and this methodology demands continuous improvement and scrutiny. To effectively deliver the highest success possible, achieving high availability cloud becomes a way of life.

Tony Orme.

Tony Orme
Editor, The Broadcast Bridge

# Delivering High Availability Cloud




By Tony Orme, Editor at The Broadcast Bridge

**Broadcast television is a 24/7 mission critical operation and resilience has always been at the heart of every infrastructure design. However, as broadcasters continue to embrace IP and cloud production systems, we need to take a different look at how we assess risk.**

The standard workflow model is based around the main-backup philosophy. That is, for every live infrastructure, a live workflow is paralleled with a backup workflow that imitates its operation. At key points along the workflows, automated and manual intervention facilitates the signal re-routing should a problem occur, such as equipment failure or power loss.

With such infrastructures, it doesn't take too much digging to find holes in the system where single points of failure quickly manifest themselves.

Cloud and virtualized infrastructures are not magic, they still suffer equipment failure just like any other electronic and mechanical system. But the good news is that the ability to provide greater workflow resilience is much easier to achieve.

One of the key advantages for broadcasters migrating to IP and the cloud is that they can take advantage of the technology gains in unrelated industries. Finance and medical are two examples where broadcasters can

learn from building resilience into their infrastructures.

## Accepting Failure

After decades of building highly resilient systems it might seem strange that broadcasters should even consider accepting failure, but this is exactly how IT centric companies think when employing enterprise grade datacenters. Instead of attempting to avoid failure, which is invariably difficult if not impossible to achieve, they accept it and build cloud native services to respond to failure and return to a fully functioning state as quickly as possible.

Traditional broadcast workflows can be thought of as monolithic systems, each stage relies on the previous stage in the signal flow, and if any of those fails then the whole workflow fails. In effect, the workflow is only as good as the weakest link. Developing the main-backup workflows does alleviate this, but only in a limited fashion. With the traditional broadcast main-backup model we simply have one monolithic workflow backing up another.

The reason for the limited backup capability comes from the need to provide application specific hardware in the workflow. For example, a standards converter will only ever convert between standards, and a transcoder can only transcode. It would be very difficult to convert a transcoder into a high-end standards converter, and vice versa. However, with software defined systems, such as those employed in the cloud, the hardware can adopt significantly more functions, and this in turn provides many more opportunities to include resilience.

## Single Points Of Failure

It's possible to argue that the traditional broadcast main-backup model does have resilience built in as any failure within the main signal flow is paralleled with the backup workflow. But how often is the backup workflow tested? In an industry where the saying "if it ain't broke, then don't fix it" prevails, then the regular testing of the backup is often feared to the point where it isn't checked as often as it should be.

The argument for the main-backup model is further strained when we look at where the equipment is located, how it is powered, and cooled. Although the broadcast workflow may be separated, its physical locality, power and cooling sources are generally not. Diverse power supplies with UPSs and generators are often employed, but to what extent? How many broadcast facilities have a completely diversified power supply system that guarantees power 24/7?

Disaster recovery models do exist where the entire infrastructure is mirrored in a location some miles away. But again, how many broadcasters can afford this level of resilience? And those that can are always looking to improve resilience and make cost savings.

Compare and contrast this to the mindset of the equivalent IT enterprise datacenter business using cloud and virtualized cloud services. Public cloud systems are built from the ground-up with resilience in mind. Datacenters are physically separated to provide A-B systems that are not only resilient, but have entirely separate power sources, air conditioning systems and security.

On-prem cloud and virtualized systems may well share the same shortcomings

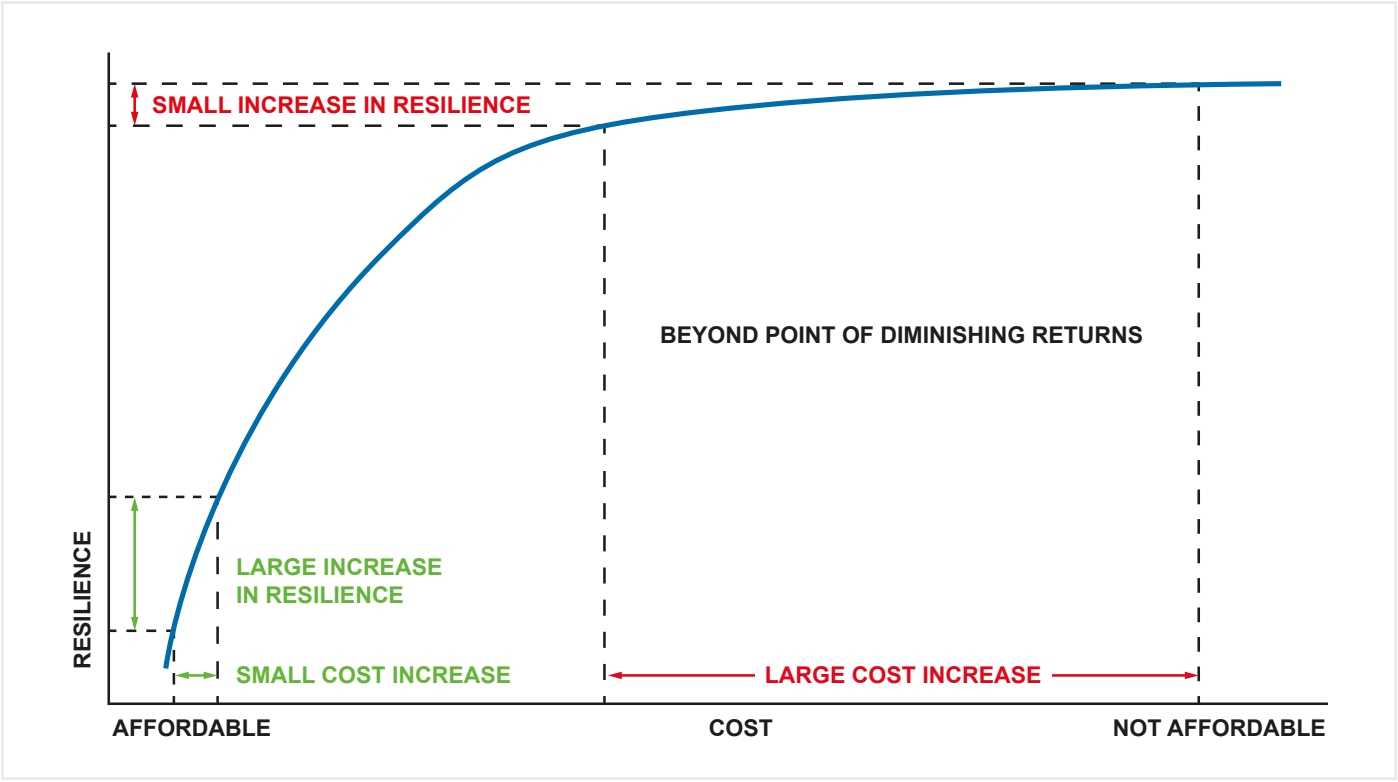



Figure 1 – Achieving resilience is far from linear. No matter how much money is spent on a workflow, the point of diminishing returns will be reached. Cloud and virtualized infrastructures help us quantify this so that the best balance between resilience and cost can be achieved.

of the broadcast main-backup model but they excel as the equipment they employ is much more versatile. Servers can take on a whole multitude of tasks from high-end standards converters to transcoders and proc-amps. The very nature of software applications running on servers makes them flexible, and completely assignable. Furthermore, cloud systems employing microservice architectures follow a similar core infrastructure. This allows them to be scaled and mirrored to other public cloud or off-prem private cloud systems thus creating very high levels of resilience.

Using cloud infrastructures encourages the broadcaster to not only adopt IT models but also adopt their mindset. That is, assume there will be failures and build a system that can resolve them quickly.

## Sources Of Interest

Working with cloud infrastructures with the IT mindset requires an understanding of where failure can occur. IP packets form the basis of an asynchronous system, and all the equipment processing, transferring, and storing the media data is also working in an asynchronous manner. SDI and AES infrastructures, by their very nature are synchronous point-to-point systems, consequently we think differently about them when compared to the IP systems.

Network latency is something broadcasters didn't really consider prior to adoption of IP but it is now a significant factor for cloud and IP infrastructures. Broadcasters need to know if they are suffering from latency issues and where are they occurring, especially for contribution services. It's all well and good switching to a back-up circuit, but what if the latency is a function of a long TCP flow that is hogging the network because a server has gone faulty and isn't obeying the rules regarding fair use policies?

There tend to be two sorts of faults, long-term and short lived, or transient. Transient faults are the most difficult to isolate and fix and they may only occur once a week with minimal impact on the transmission. Network logging is key to solving these types of transient faults so that forensic analysis can be carried out later. And long-term faults are generally easier to identify as they exist for longer periods of time and can be chased through the system. However, the asynchronous and software driven nature of IP means that the source of the problem may not be immediately obvious due to the interaction of software buffers and bursting data transmissions.

Orchestration services handle the creation of resource to meet peak demand, but the amount of resource they can create is limited by the availability of on-prem hardware, and costs for off-prem cloud systems. Again, monitoring is key to keeping these services running efficiently and reliably.

Well-designed cloud platforms can detect and mitigate many of the challenges raised, and more.

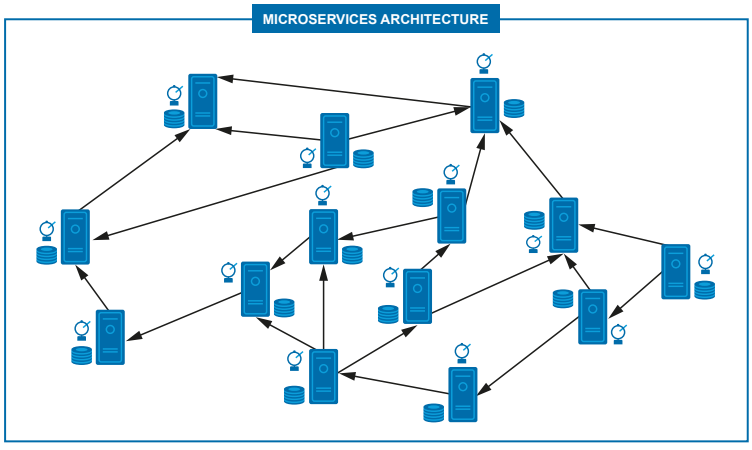



Figure 2 – Microservice architectures provide great scope for automated re-routing of workflows to significantly improve resilience.

## Designing For Resilience

It's difficult to quantify every element in a highly dynamic broadcast infrastructure employed in either on-prem or off-prem clouds and virtualization. However, due to the flexible nature of the infrastructure, high levels of resilience can be achieved by dispersing the workload throughout the facility.

It's worth remembering that IT professionals have been working with highly dynamic infrastructures that provide massive levels of reliability for many years. Although television is special due to the sampled nature of video and audio that creates synchronous data, as broadcasters move to IP, cloud and virtualized infrastructures, the underlying architecture employed is being used every day in equally challenging industries such as finance and medical.

Monitoring goes a long way in keeping cloud systems reliable as faults can be detected and rectified quickly, but many of the fixes are built into the architectures and software. For example, a "retry" strategy is often adopted when a microservice doesn't receive an acknowledge back from another microservice it is communicating with. The retry will resend the message several times in the hope that the message has been lost through a transient fault.

After a certain number of retries it's assumed that the fault is no longer transient and to stop the network being flooded with messages, the control software will instigate a circuit breaker. Just like an electrical circuit breaker, this method cuts off the server sending the messages into the network, thus stopping any potential network congestion. After a predefined length of time, the circuit breaker is automatically reset so the microservice can start sending messages again.

## Implementing Resilience

Although the traditional broadcast main-backup model may not be as resilient as broadcasters would like it to be, the cloud and virtualized equivalent can deliver much greater reliability and flexibility. The assignable nature of much of the hardware, such as the servers, means that we can identify and duplicate the infrastructure at many more points throughout the workflow.

A-B IT enterprise architectures have redundancy built into every part of the infrastructure. For example, servers have multiple disk arrays to form RAID storage that is fault tolerant, and dual power supplies are built into every device as a matter of course. Isolating functionality and distributing across multiple areas or sites is much easier as the hardware follows a design which is relatively easy to replicate.

Every element of the platform must be assessed for risk. Every, server, switch, router, and cable must be considered. If a cable fails then how is the operation affected and more importantly, what is the remedy? Although this sort of thinking isn't new to broadcast engineers, but the automated nature of finding a remedy is. Rather than thinking in terms of "if it ain't broke then don't fix it", now is the time to think "this will break, and this is my plan to resume operation in the shortest possible time".

The rip-and-replace model is at the core of any cloud or virtualized datacenter, and this has been boosted with microservice architectures. Rip-and-replace allows us to create services to scale infrastructures and delete them again when not needed. Although this doesn't go as deep as physical hardware, the concepts help understand how modern enterprise datacenters operate and how the people who design them think.

## Measuring Success

Understanding failure is important, but what is most important to the operation is success, otherwise known as uptime. The uptime, or availability is what the viewers are most interested in. They don't really care about how much effort has gone into designing resilience, but they are interested in watching their favorite programs undisturbed, especially for high-value live programs such as sport.

Measuring availability helps engineers quantify both the reliability and efficiency of their systems. Designing dual redundancy may increase availability by 0.5% and creating triple redundancy might increase availability by another 0.003%, but at an additional cost of $100K. Is the extra $100K a good investment? That's for the CEO to decide, but as engineers we can at least present accurate data to them so they can make an informed decision.

Mean Time Between Failure (MTBF) is a measure supplied by vendors to give an idea of how reliable their equipment is. And equally important is the Mean Time to Recovery (MTTR). These are combined to provide the availability, or uptime as follows:

$$Availability = \frac{MTBF}{MTBF + MTTR}$$

MTTR is the time taken to recover after a failure has happened and is dependent on the sort of problem that has occurred. Public cloud providers often cite availability with values such as 99.9999999%. As MTTR is inversely proportional to the availability, it can be seen that the shorter the MTTR, the higher the availability. This both works at a system and function level. But again, the weakest link prevails. So, if a datacenter only has an availability of 99.0% then no matter how good the architecture and application design, the best availability the system could ever achieve is 99.0%.

As a benchmark, availability is proportional to the number of individual processes in a system. If a single workflow is considered, then the total availability follows the following equation:

$$Availability_{Total} = (1 - \sum_{i=0}^{n} (1 - availability_i))$$

Where *i* is the number of processes in the workflow. For example, for three processes (where *i* = 2) with availability measures of 0.999, 0.998, and 0.997 respectively and connected in series, then the total availability will be:

$$Availability_{Total} = (1 - (1 - 0.999) + (1 - 0.998) + (1 - 0.997)) = 0.994$$

However, resilient systems that operate with parallel redundancy improve overall availability as the individual process failure rates, that is (1-availability), are multiplied as per the following equation:

$$Availability_{Total} = (1 - \prod_{i=0}^{n} (1 - availability_i))$$

Where *i* is the number of parallel (or resilient) processes in the workflow. For example, for two processes (where *i* = 1) with availability measures of 0.994, and 0.994 respectively and connected in parallel, then the total availability will be:

$$Availability_{Total} = (1 - (1 - 0.994) * (1 - 0.994)) = 0.999964$$

Therefore, just adding a single parallel workflow with adequate automated workflow re-routing will increase the availability from 0.994 to 0.999964. However, if another parallel workflow is added then the uptime will only increase from 0.999964 to 0.999999784. There has to be a point where the cost of increasing the amount of resource hits the point of diminishing returns.

## Conclusion

Cloud and virtualized computing are providing outstanding opportunities for broadcasters to not only improve the flexibility and scalability of their workflows, but to also increase their resilience. By looking at how other industries operate and understanding how they assess risk, broadcasters can both improve the resilience of their workflows and at the same time quantify the risk.

Building workflows that not only accept failure but also embrace it is key to designing and implementing reliability and uptime into broadcast workflows. The outdated thinking of "if it ain't broke then don't fix it" has been superseded by "this will break, and this is my plan to resume operation in the shortest possible time".

# The Sponsors Perspective

## Why Settle For A Or B?

By Ian Fletcher and Chris Merrill

AMPP provides many different configurations for high availability.

### Deconstructing The Monolith

Grass Valley's AMPP, the Agile Media Processing Platform, is built on a modern architecture that provides more options for high availability than a traditional media production environment.

Unlike a monolithic system which requires a duplicate copy of the system to achieve higher availability, AMPP gives users options that meet specific needs.

With AMPP we've deconstructed the monolithic model into separate components:

- Inputs
- Outputs
- Pixel processing
- Backend database
- Application logic
- User interface

Supported by

These component groups are loosely coupled. As a result, system owners can choose where to run them: on-prem, at the edge of the cloud, cloud-based, or any combination of the above.

## High Availability

What does a “loosely coupled” system mean? How does that affect high availability? Let’s break that down a little further.

### Backend Services

AMPP’s platform services can be run from an on-prem equipment room. A better model is to run them from the cloud where they run in multiple availability zones. For example, multiple data centers within AWS. AWS has a much better uptime than a single equipment room. Their data centers are staffed 24/7 by people whose only job is to keep those machines running.

The platform services are running many copies of the AMPP microservices. AMPP users can choose any released version of the software they prefer. Because a particular microservice operates in the same way, if one of the nodes in a cluster hiccups, another node takes over without any perceptible difference to the AMPP user. The Grass Valley DevOps team continually monitors those clusters to ensure that the latest security patches are installed, and the system is always running optimally.

### Pixel Processing

For additional high availability you can deploy your pixel processing for the same workflow in multiple places. For example, you could have two separate EC2s, or you could have one EC2 and one compute node on the ground. This allows multiple copies of an app to respond to the same instructions.

For many pixel processing applications, edge computing is the better model. Edge computing puts processing services at the edge of the network instead of in a datacenter. Proximity to end-users better achieves client objectives such as: dense geographical distribution and context-awareness, latency reduction and backbone bandwidth savings.

### Application Logic

AMPP has a common set of microservices. These unique microservices get used in different combinations. Some applications will use multiple microservices, some might only use one.

By using edge computing, the components of AMPP may be located according to production needs. For fastest response times, the app normally runs on the same edge node where the pixel processing is done.

### Inputs And Outputs

Inputs and outputs for AMPP can be hardware or software connections. They physically connect to a compute node on the edge and upload or stream to an IP address. Both hardware and software connections can be easily duplicated for redundancy.

### User Interface

Because AMPP’s user interface is HTML5, any device with an internet connection and authorized security credentials can reach the system regardless of where the processing and apps are located. Each operator’s production tasks are independent of where the sources and processing are located so multiple users can see and interact with the same application at the same time.

## Choosing The Best Options

AMPP can run as a standalone on-prem platform, but creating an isolated system generally negates the business objectives for adopting a new platform.

Because of AMPP’s flexible architecture, there are multiple ways to achieve different levels of high availability. Here are some things to consider when designing the right system for the application.

### Where Does Your Content Reside?

Having all your video flows already on-prem is an argument for running the system on-prem. Several customers integrate AMPP into their larger SMPTE ST 2110 production studio. Because all the camera and graphic sources are already 2110 it doesn’t make sense for them to go to the cloud and back. For these customers, the advantage of AMPP is the ability to dynamically provision workloads. The same space can be used for multiple smaller productions, a single large production or flipped from live production to master control for popup channels depending on the needs of the moment.

For other AMPP customers who need to regularly bring in sources from remote locations, the ability to send those sources to the cloud from the remote site and have them immediately available to the production team to begin working means they save both time and money by locating the rest of the production system in the cloud.

### Does Your Workflow Need To Stay Uncompressed?

AMPP supports many excellent compression formats which provide no noticeable difference in picture quality for most applications. Compression makes it easier and less expensive to transport signals over distributed networks. The production team can use these signals without conversion to a standard production format, thus saving time and avoiding multiple format conversions.

For applications where compressing a signal is not an acceptable option, AMPP uses uncompressed video formats - 10-bit YUV when exchanging flows between micro services on the same node. Because transporting uncompressed video signals to the cloud consumes large amounts of expensive bandwidth, it may be best in this case to run the system on-prem.

**Do You Need The Resultant Video Flows Back On-prem?**

Egress charges are one of the more expensive parts of cloud processing. If you are storing content or using a cloud CDN, it is generally most cost-effective to move the content to the cloud once and leave it there. For many AMPP customers, the ability to monitor live content, create localized versions, and play it out without bringing the content back to a terrestrial system is a highly efficient production model.

**How Often Do You Use The Workflow?**

As with most technology, the costs of cloud processing continue to decline. But if you are considering AMPP for workflows that operate continuously in an existing environment, operating on-prem may make the most commercial sense.

If you are building a 24/7 operation in a greenfield environment, a cloud-based operation may be more cost effective. You could buy hardware servers for less than a VPC, but then you need to provide power, cooling, and maintenance. You'll need to keep it updated with the latest patches and continually change out the hardware every few years. Once you add up the total cost of the operation, it may cost more to run your own datacenter.

For short duration events such as pop-up channels, or a championship tournament where there is burst of activity and then the system lies dormant, a cloud system can cost less than the purchase of hardware for that activity.

## Conclusion

It's time to rethink our approach to highly available media production systems. Running duplicate main and backup systems is more expensive and difficult to operate than today's solutions. AMPP's architecture, particularly when installed as a cloud-enabled edge network, can achieve better total uptime than local engine rooms while providing far more flexibility in your production workflows.

Ian Fletcher.

Chris Merrill.

### GV Hub Local Discovery

It's rare for major cloud providers to have an outage, but it is possible to have the occasional blip. AMPP Hub maintains high availability during these moments.

When running an edge network with AMPP Hub, all the on-prem computers are connected to AMPP through AMPP Hub. Under normal circumstances, AMPP Hub acts as a load balancer, managing the system traffic so that only the messages that need it go to the cloud. The rest of the traffic stays local. For example, a button press on a Switcher doesn't need to go to the cloud and back, it goes directly to the application running on the edge device.

If the connection to the Internet is briefly broken, AMPP Hub ensures all the local traffic continues without interrupting the operator. Some aspects of the system, such as adding a new workload or loading new clips to storage might be paused, but the basic production functions continue to keep the show on air until the connection is restored.

Supported by

BROADCAST
THE BRIDGE
Connecting IT to Broadcast

Themed Content Collection

EG
ESSENTIAL GUIDES

MEDIA

ARTICLES

For hundreds more high quality original articles and Essential Guides like this please visit:

thebroadcastbridge.com

03/2023

Supported by

Grass Valley
WE LOVE LIVE