Grass Valley
WE LOVE LIVE

WHITEPAPER

GV AMPP System Architecture

By Jochen Bergdolt and Ian Fletcher

Why We Built AMPP

We built AMPP (Agile Media Processing Platform) because we felt our customers would benefit from the advantages of software-as-a-service (SaaS). SaaS applications are written to run in a single controlled environment and therefore result in dramatically easier and faster development, debugging and support — leaving more time to deal with what matters the most: *Delivering innovative features for the media and entertainment landscape.*

AMPP is reinventing media production, allowing content creators to produce anywhere and distribute everywhere with globally dispersed teams, having total control over each step in the production.

With AMPP, our users get all the benefits of cloud computing by scaling to an almost infinite number of simultaneous tenants (customers of the AMPP SaaS applications) and to as many workloads as required without upfront buildout — all facilitated through a microservices architecture that provides the ability to spin up or tear down functionality in minutes.

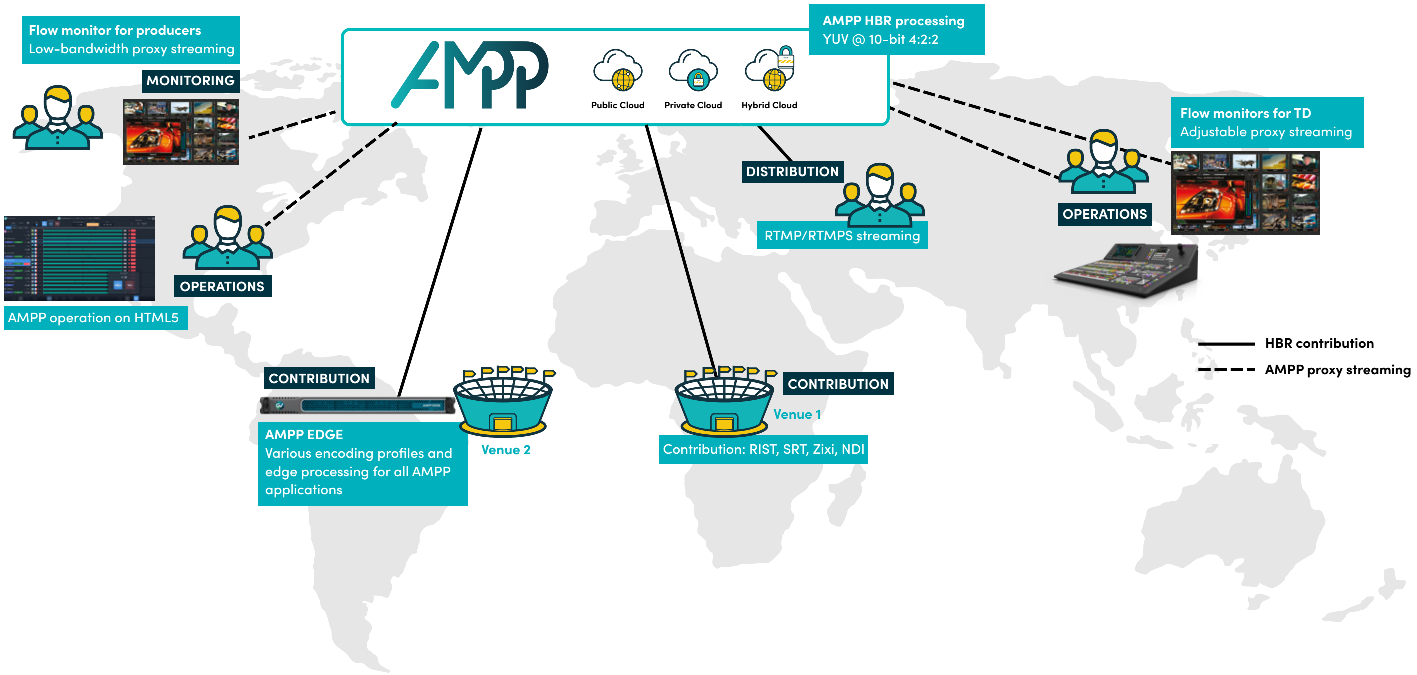


Figure 1: Built-in reliability, synchronization, low latency, security, firewall traversal and connection establishment.

AMPP has been built from the ground up as a cloud-first microservices architecture that consists of a Grass Valley operated multi-tenant platform (provided as SaaS), and a private customer video processing environment running applications either in the cloud or on premise. This enables extremely flexible and hybrid workflows that have all the advantages of the cloud, but recognize that, for some use cases, processing video at the edge makes more economic sense.

To understand how this works in more detail we need to explore what some of these terms mean as they are often used incorrectly.

AMPP has been built from the ground up as a cloud-first microservices architecture

Cloud First

Cloud first refers to the way the software is designed to run so that it takes advantage of all of the native services available in public cloud platforms and consists of a set of microservices that are distributed across many physical computers. These architectures are normally defined as being high availability because the work is distributed across many microservices, any one of which can fail without impacting the overall performance of the system. As these types of architectures are expensive to run, they are normally defined to be multi-tenant so the costs can be aggregated across multiple customers. Multi-tenant systems lead to not just better value to an individual customer, but also better performance and reliability than you would get with a traditional "lift and shift" approach to the cloud, which often just installs traditional monolithic software, which often just runs on a dedicated virtual machine in the cloud.

Microservices

There is no single definition for microservices. A consensus view has evolved over time in the industry. Some of the defining characteristics that are frequently cited include:

- Services in a microservice architecture are often processes that communicate over a network to fulfill a goal using technology-agnostic protocols such as HTTP
- Services are organized around specific use cases, such as providing functions to store and retrieve some configuration data
- Services are small in size, messaging-enabled, bounded by contexts, autonomously developed, independently deployable, decentralized and built and released with automated processes. This makes it possible to upgrade and downgrade individual services with zero downtime and no impact on other parts of the system
- Services are normally stateless in that they don't hold any data in memory; this enables them to survive being restarted without loss of data

In an analogy that is common to the IT industry, the difference between traditional monolithic software and microservices is like the difference between cattle and pets. Traditional software is like a pet, you give it a name, keep careful watch over it, and are very upset when it is lost or malfunctioning. Microservices are more like livestock, the size of the herd is important and you don't get attached to any specific one and if a unit is lost, it is quickly replaced.

In order to manage a large number of microservices distributed over multiple computers in different data centers, you need to use a management layer that handles the lifecycle of stopping and starting all of the individual services and managing the resources they have available. Grass Valley uses Kubernetes, also known as K8s, to manage its platform. This is an open-source system for automating deployment, scaling and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the IT community.

- *A single AMPP Platform consists of approximately 100 individual microservices that are distributed across clusters of compute in three different availability zones. AMPP has multiple platforms distributed around the world so that customers access a platform that is closest to them, resulting in minimal delays.*

The following diagram illustrates how the application layer and data layer span across multiple availability zones:

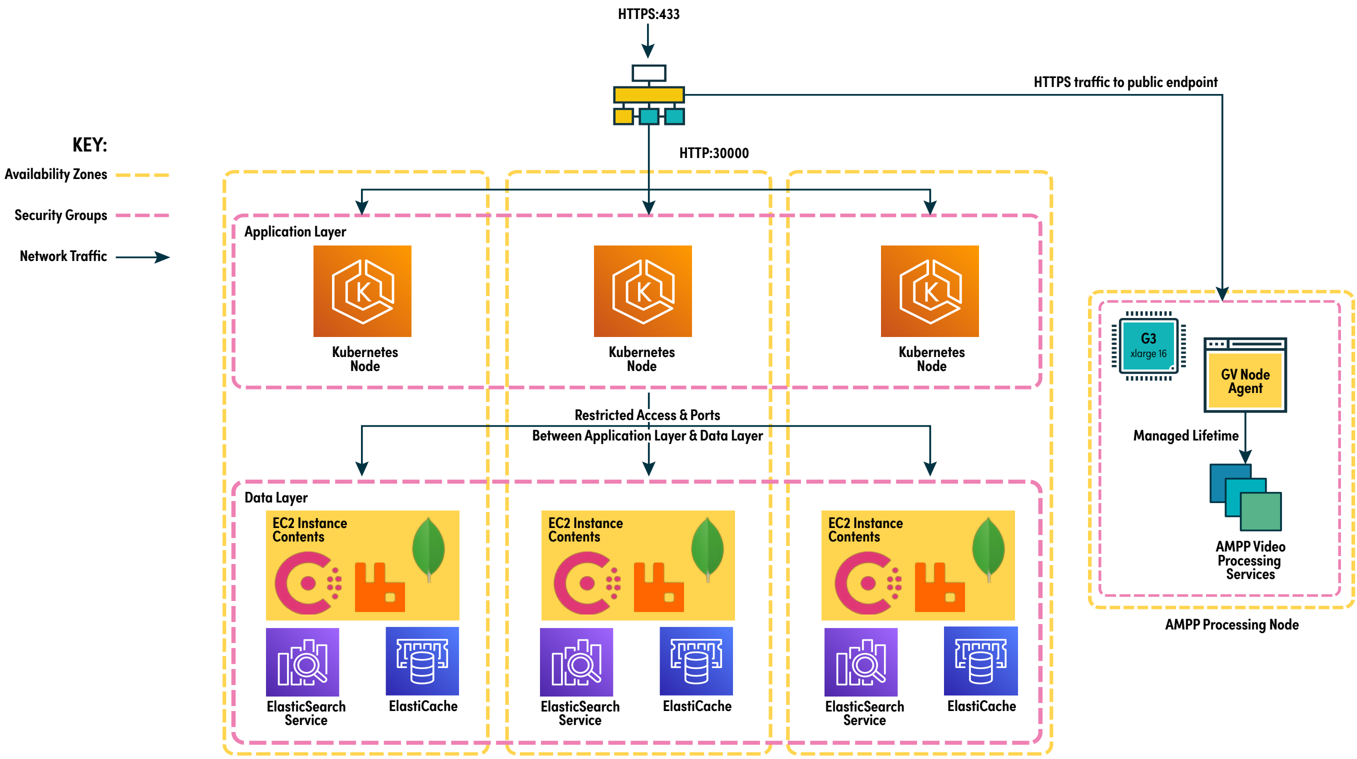


Figure 2: Single platform view – platform application and data layer spanning three availability zones.

The AMPP Platform

The platform consists of a front-end load balancer that sits behind a single secure HTTPS URL. This is referred to as the front door and is how all users and machines interact with the platform. As messages and requests arrive, they are routed to an available microservice that can process that request.

The platform is responsible for messaging and storing and retrieving data — no video processing is done in the platform and no individual applications for processing data are running. This means that functional data and content data are managed separately.

The platform is monitored 24x7 by the AMPP NOC, using a number of detailed dashboards that show the health and performance of the platform down to an individual service level.

The AMPP Applications Environment

This is where all the real-time video processing happens and is private to each individual customer account, either running on-premise or a public cloud hosted machine.

As a highly efficient architecture, many individual AMPP applications can be deployed on a single compute node. These consist of an application layer, which contains the business logic and exposes an HTML5 user interface, plus one or more video processing microservices.

These apps can be stopped and started individually as needed, but they all share access to a common set of shared video flows so that multiple apps can interact with the same frames of video very efficiently and without incurring any significant latency.

The video frames are stored in memory (Framecache) on the node and all processing is done on 10-bit/4:2:2 uncompressed frames, empowering high-quality UHD and HDR workflows.

You can run many instances of the same app, each with its own specific configurations, called workloads. Workloads can be managed from a central application — the AMPP Resource Manager. The advantage of this approach is that different productions can have their own workloads, which can be stopped and started as a block while preserving all of their individual show setups.

AMPP Control

One of the key design principles of AMPP has been to make it as open as possible — both for third-parties to control applications within AMPP, and for third-parties to expose external hardware and software in a consistent way so it can be controlled by AMPP.

Traditional protocols and mechanisms will not work in a cloud-scale secure architecture, so AMPP Control was developed using the standard web-based mechanisms of JSON messages of HTTPS REST (representational state transfer). We also wanted the SDK to be extensible and self-describing. Hence, all applications running on AMPP register a schema with the AMPP Platform, which describes the features and functions that can be controlled, and this schema acts as both live documentation and provides the data needed to build a user interface.

AMPP Control can be further used to assign functions to hardware control panels and soft panels, or it can be used to build macros of multiple instructions that can then be assigned to a single button.

The AMPP Control service also means third-parties do not have to know where a specific application is running in the customer account. They simply post the required payload to a single secure endpoint, specifying what the workload it is meant for, and the target application will act on the request wherever it is running in the world.

User Interfaces

All AMPP user interfaces are HTML5-based and therefore require no specific software to be installed on the client machine. This also means they work from any location, securely connecting through the cloud to the application being controlled. This makes AMPP very suitable for home working, especially when combined with our low-latency Flow Monitors, which allow high-quality video to be viewed on a browser with minimum latency.

GV UI

In production control rooms, you often want to have multiple HTML apps active at the same time and operate in a more locked-down environment than a standard browser. For this use case, customers have the option of installing GV UI, a thick client that allows construction of custom layouts consisting of multiple HTML apps on the screen at the same time, and also allows fast navigation between different layouts. There is also a web-based version of GV UI available that does not require local installation.

Dashboards

One of the important trends we are seeing is the move to highly optimized operational interfaces focused on specific workflows or productions. By using AMPP dashboards, users can create custom displays that not only represent the production signal path, but also provide required controls to operate the various apps used in the production. By restricting the number of controls a user must deal with, you remove the chances for operational errors and make it easier for a single operator to monitor multiple simple use cases.

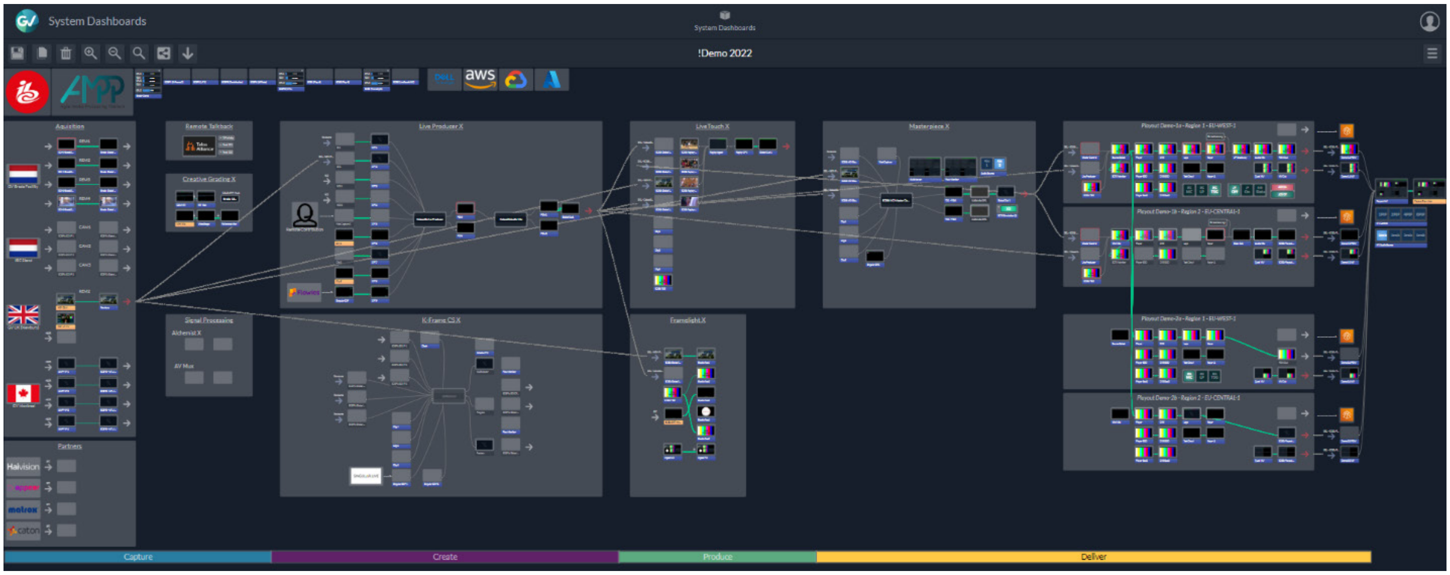


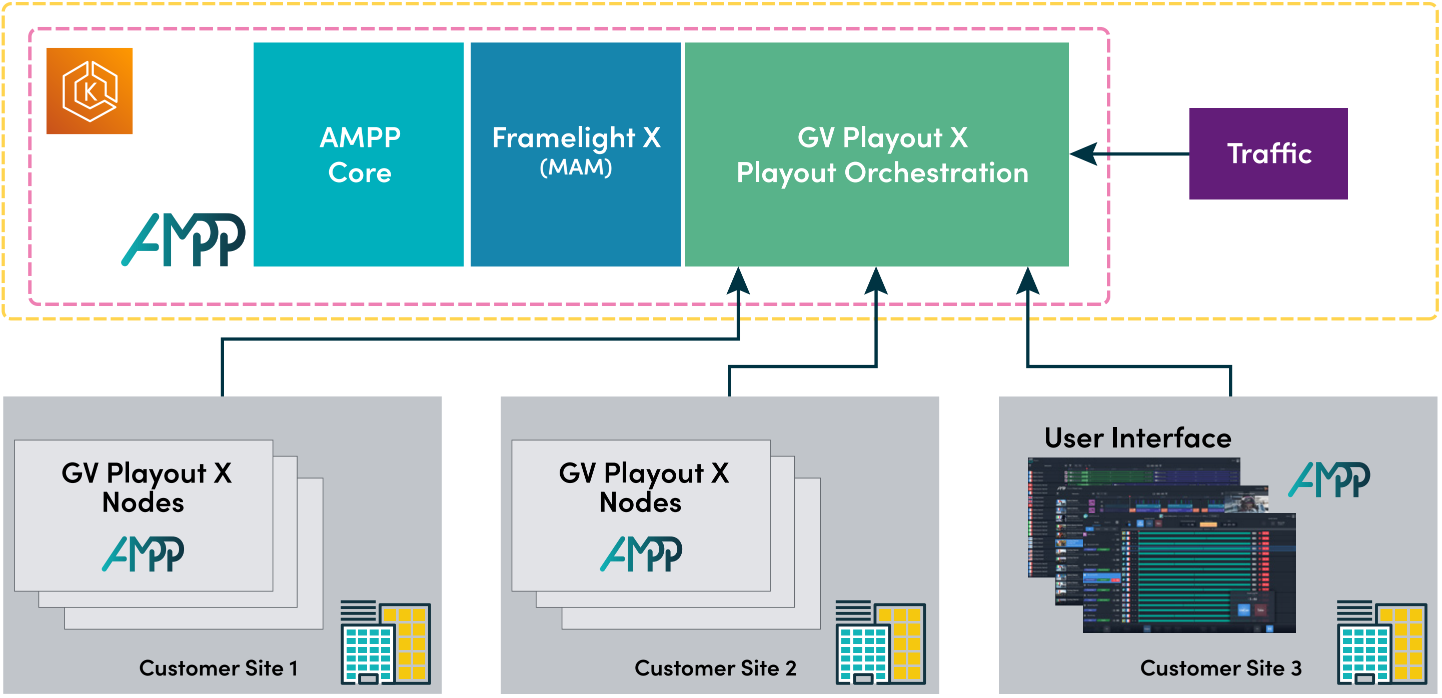
Figure 3: Example of an AMPP Dashboard.

GV AMPP Applications Architecture Examples by Use Case

The following diagrams illustrate concepts, entities and services used with regard to a specific application and/or a particular use case.

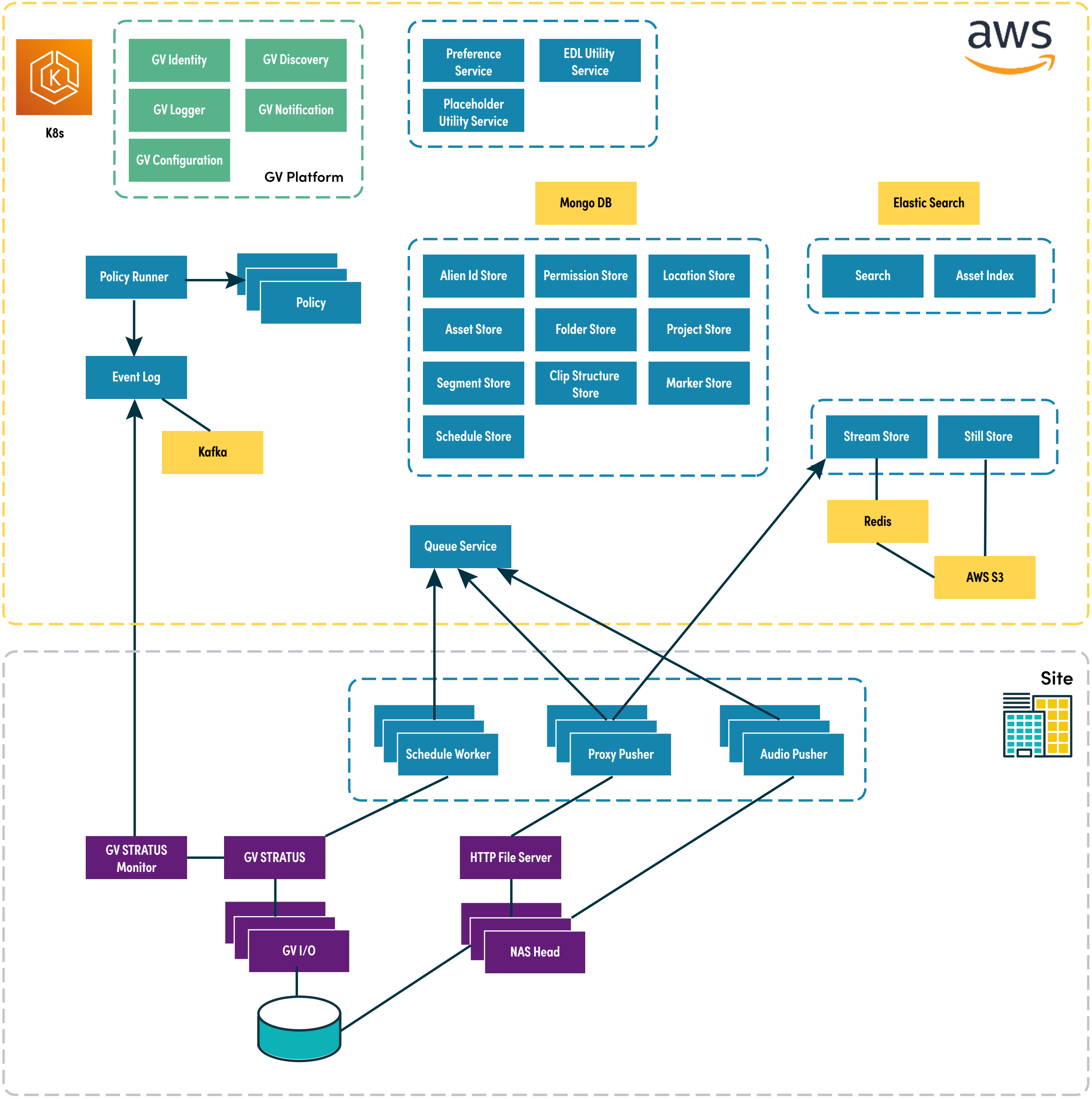
GV Playout X

In the diagram below, Playout Orchestration, running in a Kubernetes cluster, is the core component for GV Playout X, where all requests coming from services, workloads or user interfaces — or a customer's own services — are being handled. Playout Orchestration is multi-tenant, so customers will all use the same Playout Orchestration. However, each customer has an account, a tenancy, and all their data is restricted to that tenancy in order to achieve perfect tenant and data segregation.



Framelight X

In the following diagram illustrating a hybrid asset management solution, the AMPP-connected GV STRATUS is shown in the lower half, with the upper half illustrating the cloud services and entities engaged for Grass Valley's asset management solution, Framelight X. Several stores are employed, each representing a specific entity in the asset management world (Markers, Projects, Locations, etc.) to serve entity-related payloads in a highly efficient manner. The search subsystem is made up of Elastic Search, while other services like logging and authentication are co-used among the entire AMPP platform (as illustrated by the green boxes below).



The Onboarding Experience

Getting up and running with AMPP is a very fast process: we create one or multiple accounts in one or multiple regions of your choice and add an initial admin user and provide you the credentials. That admin user can then onboard additional users and give them specific security roles that control what they are allowed to do.

If you are hosting your own compute in the cloud, we will provide you with an Account Management Interface (AMI) to get you up and running with the first node. Once we have securely associated it with your account you will be able to start downloading apps and creating workloads. The whole process should normally take less than 20 minutes.

Please read more about SSO in our AMPP whitepaper "Single-Sign-On with AMPP."

We Care, You Run Your Business

At Grass Valley, we recognize that transitioning to new technology requires new knowledge and understanding of best practices. To learn more about using the AMPP platform for media production and distribution, see additional whitepapers from Grass Valley that include topics such as: Security, Latency, Reliability/Redundancy & Availability and Single Sign On (SSO).

This product may be protected by one or more patents. For further information, please visit: **www.grassvalley.com/patents** WP-PUB-3-1039A-EN

www.grassvalley.com Join the Conversation at GrassValleyLive on Facebook, Twitter, YouTube and Grass Valley on LinkedIn