2030 VISION SERIES

INTEROPERABILITY IN MEDIA CREATION

Enabling flexibility and efficiency through interoperable and composable software-defined workflows

CONTENTS

EXECUTIVE SUMMARY

The MovieLabs "Evolution of Media Creation" white paper lays out a vision for media production workflows that can be more easily assembled, managed, changed, and secured than the current rigid and brittle workflows. The goal is to provide production teams both a greater *choice of tools and infrastructure and better capabilities for collaboration, turnovers, and iteration*. The former gives teams more creative options in how they work; the latter gives them the ability to spend more of their valuable time on creative tasks and less on mundane ones. Interoperability is the foundation for both.

> Without improved interoperability, each new workflow or vendor choice made by a production team will be unnecessarily costly in terms of time, risk, and engineering effort.

Let's break this down in more detail:

- Studios, production teams, and their vendors want to be able to use the tools and vendors best suited to a project. In a world where production work is heavily supported by software, this interoperability among software tools and SaaS offerings is essential to giving productions the ability to choose (and to try out) the tools and vendors that best meet their needs, with as little customization and integration as possible.
- They also want to take advantage of software to better support creative work through collaboration for quicker and better decision making and the automation of repetitive tasks for quicker turnarounds. This requires tools and SaaS offerings to provide the connection points for integration with other components and workflow automation systems.
- Last, but not least, infrastructure choices may be driven by the above preferences in toolchains, services, and teams or by other considerations of what best meets the needs of the production. Often, this will require some combination of on-premises, public cloud, private cloud, and SaaS offerings. The software used in workflows needs to interoperate seamlessly across all of these to render the infrastructure invisible to end users.

When developers of software tools and SaaS offerings provide this interoperability, the creative process and workflow becomes more efficient and productive and offers choice for users. This not only facilitates studio or vendor implementation of workflow solutions, but can also enable a new category of "production-platform-as-a-service" offerings that bundle available tools and services into complete, configurable, turnkey production workflows. This paper makes the case for developing the improved interoperability that will meet those needs and provide those choices.

The paper is aimed at Production Technology Strategists , Workflow Designers (studios and productions), Software Product Managers, Workflow and Solution Architects (studios, productions, integrators, and PaaS providers), and Developers of standards and open software.

First, we lay out the benefits of interoperability: providing creative teams more choices in tools, vendors, and infrastructure while also supporting the software automation that will increase efficiency and give them more time to focus on creative work rather than on mundane tasks.

The paper then provides a set of interoperability principles to guide the industry in implementing solutions and against which progress can be measured.

Through adoption of these interoperability principles, we can do the following:

- Enable dramatic improvements in workflow efficiency.
- Support the workflow flexibility needed for productions to embrace the dramatic changes new technologies are bringing to production.
- Measure progress toward those goals.

The interoperability principles are divided into two sections: general principles that apply regardless of the level of software support and then detailed ones that enable software support and automation.

GENERAL INTEROPERABILITY PRINCIPLES

1. **Asset Access** – Assets are securely and directly accessed by authorized participants where they are across different applications, XaaS[1] systems, and clouds.

2. **Common Data Models** – Applications use common data models to preserve and propagate data.

3. **Asset and Metadata Delivery** – The assets and metadata required as inputs to and available as outputs from workflows or workflow components use common formats and are fully described in delivery specifications.

    a. **Asset and Metadata Formats** – Assets and metadata exchanged between organizations or intended for long-term use are in publicly documented formats and schemas.

    b. **Asset References** – Assets are referenced by identifiers that are independent of location and are resolvable to asset location by receiving applications and systems across multiple clouds.

4. **Preferred Tools and Infrastructure** – Tasks in a workflow are easily changed to use different applications, XaaS systems, and clouds.

SOFTWARE-DEFINED WORKFLOW INTEROPERABILITY PRINCIPLES

5. **Data and Control Plane Functions** – Applications, services, and XaaS systems enable data exchange and control through common mechanisms, including across organizations and clouds.

    a. **Events and Notifications** – Applications, services, and XaaS systems enable the exchange of relevant notifications with third-party systems, including across organizations and clouds.

    b. **Security** – XaaS systems enable control of security, including authentication and authorization, by third-party systems.

6. **Reusable Integrations** – The integration of each component is reusable, e.g., by utilizing common APIs & platforms and minimizing custom point-to-point integrations.

7. **Automation of Repetitive Tasks** – Non-creative, repetitive tasks are automated.

    a. **Automated Session Provisioning** – Upon completion of a task, the next creative task(s) that use that work are fully and automatically provisioned so that work can be started, e.g., triggered by notifications of the completion.

    b. **Automated Work Orders** – Work is described in clear, unambiguous sets of instructions for automated execution.

8. **Tracking and Dashboards** – Work is described and tracked so that progress, resource utilization, and costs can be continuously determined.

---

[1] The term XaaS refers to anything-as-a-service offerings or a range of cloud-based service offerings that includes software as a service, platform as a service, integration platform as a service, etc., and, in our case, production platform as a service.

> **Reusable Integrations – The integration of each component is reusable, e.g., by utilizing common APIs & platforms and minimizing custom point-to-point integrations.**

We already have some of the tools to implement these interoperability principles. Standards and open-source efforts provide many interoperable file formats, metadata schemas, and APIs, but particularly in the areas of metadata and APIs, current efforts cover only a small fraction of what is needed. The MovieLabs Ontology for Media Creation[2] addresses some additional areas of data interoperability in workflows, enabling the connection of assets to script and production elements. But the availability of specifications is not enough; wider adoption is needed.

The paper concludes with a set of actions that all partners in the ecosystem can take toward implementing this future for production, including:

- Adopting common data models and encodings.
- Building and utilizing more open-source software and formats.
- Exposing more functionality in products for their integration into software-defined workflows.
- Developing new approaches to enable the reusability of workflow integrations.

2. https://www.movielabs.com/omc

SECTION 1
INTRODUCTION

We're working in challenging times. Creative teams are faced with historical levels of content production with more deliverables per title, along with increasingly complex workflows and rapidly changing technologies — and, as always, all under tight schedules and a scarcity of talent and resources. To help meet these challenges, we need to find ways to better support these teams in their creative workflows while also giving them the flexibility to adopt new technologies with less friction and risk. Succeeding at this will allow creatives to iterate more and better realize their vision within the same timeframe and with the same resources.

As software plays an increasing role in workflows, better interoperability of tool, service, and infrastructure offerings is essential to the future of production. To that end, this paper builds on the MovieLabs 2030 Vision white papers[3] by putting forward a set of interoperability principles to organize thinking and efforts around improving the interoperability of the tools, systems, and infrastructure that are assembled to create production workflows.

SCOPE

This paper provides a foundation for understanding and building the interoperability necessary to make media creation workflows more composable, manageable, and efficient, with a focus on the interoperability of software-based systems, tools, and infrastructure and all the assets and data that they exchange.

Some aspects of this interoperability occur in similar ways throughout workflows. Security is an example of this, with interoperability requirements for integration with authentication and authorization management systems present in virtually all workflows. Other examples of common patterns include things like reviews, approvals, and handoffs, which occur in most workflows.

[3] The Evolution of Media Creation (2019), The Evolution of Production Security (2020), and The Evolution of Production Workflows (2020).

But in addition, each principal task or workflow in a production has different needs, both internally and externally through its connections with other tasks and workflows, e.g., turning over specific types of assets and metadata. Improving the interoperability of these handoff points is a priority, and the specific needs of each must be understood and addressed. While this paper does in places provide examples of specific tasks where interoperability provides benefits, it does not attempt to enumerate the specific needs of each ever-evolving workflow. Instead, its goal is to provide a framework for the subsequent detailing of interoperability requirements, both for repeating patterns and for the specific needs of individual workflows.

INTENDED AUDIENCE

This paper is intended for the following sets of readers and uses:

- **Production Technology Strategists** – How should interoperability drive my technology roadmap?
- **Workflow Designers (studios and productions)** – What forms of interoperability do I need from my production technology providers?
- **Product Managers (tool, SaaS, infrastructure, and integration platform providers)** – What forms of interoperability do I design into my products and services?
- **Workflow and Solution Architects (studios, productions, integrators, and production-platform-as-a-service providers[4])** – What forms of interoperability do we need from our tool, service, and infrastructure providers? How do we design and specify our workflows to be more flexible?
- **Security Architects (tool, SaaS, infrastructure, and integration platform providers)** – What security features do I design into my products and services?
- **Standards Organizations and Open-Source Developers** – Where and how could a new or improved standard or open-source project address gaps?

This paper does not provide instructions on what to do or a detailed enumeration of gaps between where we are today and a future where these interoperable systems are commonplace. Instead, it provides a framework and set of principles on the forms and types of interoperability needed to realize the 2030 Vision so that subsequent work can detail the gaps and then address them.

4. We use production as a service to refer to SaaS offerings that integrate multiple tools and services into a turnkey media creation workflow.

SECTION 2

BENEFITS OF INTEROPERABILITY

We all know the benefits of standards, which are one of the main ways in which interoperability is achieved between tools and steps in workflows. Without standard image, video, sound, 3D, and other essence formats, the output of one tool would not be usable by another.

We all know the benefits of standards, which are one of the main ways in which interoperability is achieved between tools and steps in workflows. Without standard image, video, sound, 3D, and other essence formats, the output of one tool would not be usable by another.

This paper assumes we already have some level of interoperability there and looks toward what additional forms of interoperability are required to enable what we call software-defined workflows (SDWs), that is workflows that are supported by software that takes the friction out of media creation, enabling more productive collaboration, more time for creative iterations, and more rapid turnaround times.

At a fine-grain level, media creation workflows are supported by many software tools, databases, asset management systems, scheduling systems, etc., running on many different infrastructures, whether on-prem, public cloud, private cloud, or SaaS platforms, using many different types of assets. At a coarser-grain level, they consist of handoffs between departments and vendors. Interoperability can enable automation and recomposability at both these levels, enabling both choice of individual tools and components as well as choices of vendors and infrastructure.

Figure 1 below shows the coarse-grained data flows between workflows where improved interoperability is useful.

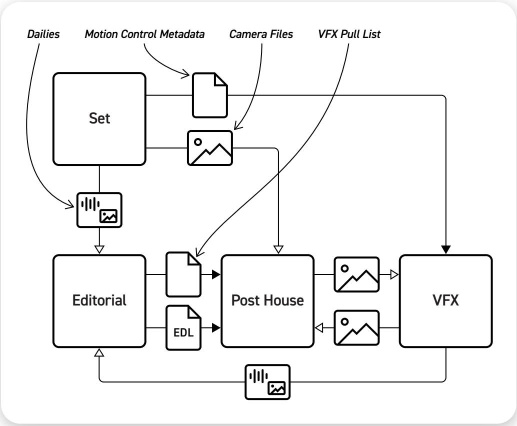

Figure 1: *Coarse-grained data flows between workflows*

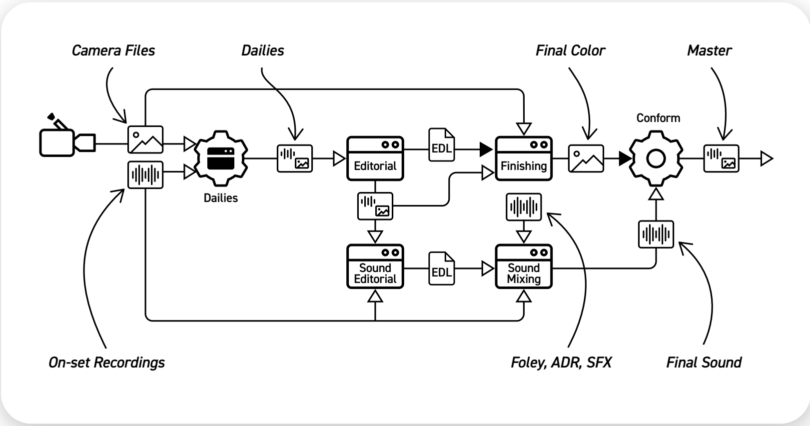

Figure 2: *Finer-grained data flow between tasks*

Figure 2 above shows the finer-grained data flows between tasks within workflows. An even finer example would show the data flows between individual tools and services.

The importance of interoperability at each of these levels depends on the priorities for automation and flexibility at each of those exchange points. The result of this interoperability is captured in the dynamically created workflows of 2030 Vision[5] Principle 9, which envisions building blocks that can easily be configured for and composed into different workflows.

IMPROVED WORKFLOWS

The most basic level of interoperability for SDWs is that which enables workflows and their individual components and processes to be connected and supported by software. It is this integration that realizes many of the benefits of interoperability, including improved collaboration, less time spent waiting for work orders to be completed, and more rapid turnarounds. Combining SDWs with interoperable data models enables even more powerful use cases.

[5.] The Evolution of Media Creation (2019) defines the ten principles of the 2030 Vision (https://movielabs.com/prodtech/ML_2030_Vision.pdf)

Below are some examples of the capabilities that we are seeking.

AUTOMATING HANDOFFS ACROSS DEPARTMENTS OR VENDORS

- Editorial approves a new sequence. The VFX pull is generated automatically, and the VFX houses working on that sequence are automatically notified and given access to the timeline, proxies, and plates.
- Editorial approves the removal of some shots from a sequence. The VFX houses working on elements for that sequence are given automated notification of the updated timeline, which propagates into their workflow management systems so that a VFX Supervisor can immediately see what tasks are no longer needed.

USING AUTOMATION TO EFFICIENTLY HANDLE CHANGES

- A proposed script change to a live action show would require a pickup shot requiring an additional shoot day. Automated tools generate a new script breakdown and provide immediate cost estimates.
- A change to a scene is approved that changes the production location of a scene and associated props, set dressing, etc. The information is updated in a common data platform accessible to authorized departments. Based on the scene, the multiple departments and vendors involved are notified of the changes; in parallel each uses its own tools to start updating its work plans.

USING COMMON DATA MODELS WITH RELATIONSHIPS IN ARCHIVE

- Multiple departments use a common data model to store the relationships of all on-set captured assets and metadata to scene, shot, and slate. Using this data, the archive can easily determine all of the on-set elements (e.g., OCFs, camera and lens metadata, witness cams, etc.) related to the final edit and apply the correct policies to their preservation.
- Using a common data model with those relationships, the archive can support queries that can deliver insights across assets and metadata to make the relevant material easily available for use and analysis around budget, future scheduling, and planning, and to incorporate footage into sequels, directors recuts.
- Production has begun on a sequel. The previous installment used a common data model to store the relationships of all pre-production and on-set elements (e.g., location scouting, previsualization, technical visualization, OCFs, camera and lens metadata, witness cams, etc.) to scene, shot, slate, and location. As the production plans for a shoot in the same location with the same actors, the on-set assets from the previous version are recalled from archive and used by various departments for planning or actual use in the new production, saving time and work.

MANAGING WORKFLOWS AND RESOURCES

- A character animator, working remotely, has been assigned work. A virtual workstation is automatically provisioned in a data center near their location with all the assets that they need.
- An animation studio has hundreds of artists on a production using virtual workstations. They want to make sure that workstations are suspended or shut down when not in use.
- An archive labels some assets as archival. Storage management systems automatically increase the level of storage robustness of those assets to match their classification.
- A VFX house needs to re-render a sequence. It needs to know the time and cost trade-offs for using burst rendering infrastructure versus an on-prem rendering farm.

SECURING WORKFLOWS

- A director wants to limit access to dailies until they have reviewed and approved them. Their clicking an approval button changes the access controls on the dailies so that other production members can see them (only) after approval.
- Editorial approves a change in a timeline that lengthens a VFX sequence being worked on by different VFX houses. Their clicking on an approval button changes the access controls on the EDLs and additional plates so that (only) those VFX houses are given access to the updated EDLs and new plates.

IMPLEMENTING FAILOVER OR CONTINUITY PLANS

- An outage at a cloud data center causes plates stored there to become unavailable. Because the finishing workflow had interoperable backup storage on other infrastructure, it was able to switch the automated VFX pull system to use alternate storage with minimal disruption to service.
- A failure at a SaaS provider for editing causes a service outage. Because the workflow automation was integrated with both the SaaS and remote workstation offerings, the production was able to quickly provision the same assets and session configuration to a remote workstation so work could continue.

But with only baseline interoperability, significant engineering efforts may be required to perform format and metadata translations and to build bespoke point-to-point software integrations between components or between workflows across departments and organizations.

COMPOSABLE WORKFLOWS

Once we have enough functionality exposed to create software-defined workflows (albeit with some difficulty due to diverse data models and APIs), the next level up entails the interoperability to more easily build or compose an SDW. This same interoperability also makes it easier to make changes to an existing workflow to accommodate the needs of a particular production or type of production, i.e., to recompose the components in an existing workflow. These are what we call composable and recomposable workflows.

The ideal result of full interoperability is the ability to easily replace any part of an SDW at any scale without any additional engineering. But full plug-and-play interoperability at every touch point across all of these hundreds or thousands of elements would not be feasible — and, given the value of experimentation and innovation, perhaps not desirable.

One needs to start where the greatest benefits can be realized. So, we focus here first on the ability to recompose workflows at a coarser granularity, at exchange points between departments, vendors, and major tasks, rather than starting with the internal workings of tasks and the complexities of making work in progress interoperable. This focus serves the important goal of enabling flexibility in choosing vendors and SaaS offerings while also addressing the important issue of handling workflows across multiple infrastructures.

ENABLING CHOICE

The benefit of composability is that it gives production teams choices and flexibility while minimizing the risk and cost in time and engineering resources. These include choices of preferred tools, of the best workflow for the production, and of vendors, even if different from the current ones.

Examples of these choices include the following.

CHOICE OF PREFERRED TOOLS

- An editor, sound mixer, or colorist has a tool which they want or need to use. The production workflow needs to accommodate that tool, even if the workflow and its automation was originally built using a different one.
- A production team wants to try an innovative new tool or service. The new tool needs to be integrated into the workflow for testing and evaluation by the team.

CHOICE OF WORKFLOW

- A production wants to use ACES for the color workflow. If the existing workflow used something else, it needs to be reconfigured to use ACES.
- A production wants to use LED walls for in-camera VFX (ICVFX) on some shots. If the existing workflow only supports traditional VFX in post, it needs to be reconfigured for ICVFX virtual production.

CHOICE OF VENDORS

- A production is running late and wants to add a new VFX house. The existing workflow supports automated VFX pulls and pushes with other vendors. The new vendor needs to be integrated into the automated system.

CHOICE OF PREFERRED INFRASTRUCTURE

- A production wants to use a SaaS tool that is only available from a particular infrastructure provider. If the workflow was built around a different infrastructure, it needs to be reconfigured to integrate with another infrastructure.
- A studio prefers a particular infrastructure. If the production team is accustomed to using different infrastructure, the studio wants to provide its preferred infrastructure with minimal differences visible to the production.
- A set of large assets are stored on one infrastructure, but the production wants to process them with services only available in a different infrastructure. The production wants to avoid maintaining two copies of the assets. The services need to be able to access the assets across infrastructures, even if a cached copy needs to be created for the duration of the processing due to performance reasons.

> Interoperability between the components in a workflow of their inputs, outputs, and control functions enables the above reconfigurations to be implemented while minimizing the amount of custom translation and re-integration work that needs to be done, reducing both cost and risk.

SECTION 3
INTEROPERABILITY

Interoperability can be broadly defined as *"the ability of a system to work with or use the parts or equipment of another system."* [6] In the context of media creation workflows, this is the ability of the tasks and processes to be more recomposable even when assembled into software-defined workflows. The parts can be regarded as the individual tools and services used in the creation of the work and the equipment as the larger combinations of these tools and services into workflows, which also involve processes. We want interoperability at both fine and coarse granularities. And one key aspect of the interoperability of content creation systems that process inputs into outputs is the ability of those tools, services, and workflows to handle the same inputs and outputs, including metadata.

FORMS OF INTEROPERABILITY

We can break this interoperability into two main areas:

Data Ins and Outs: The asset types, file formats, and metadata systems that the tools and services produce and consume, e.g., OpenEXR, OTIO, FCPXML, USD, DCDM, OMC, etc.[6] Interoperability here starts with what the available tools and services can produce and consume, and then moves to the data flow between tasks in an actual workflow. A tool may support many formats, but interoperability in a particular workflow may require following specific delivery specifications for assets and metadata. In the greater end-to-end workflow with many tasks, this also includes the capture, propagation, and preservation of metadata for downstream tasks. The clear specification of these data flows is useful independent of automation.

Interfaces and Integration: The mechanisms and interfaces that tools and services use for integration into software-defined workflows, e.g., interfaces for managing assets in a SaaS system, for reading and writing cloud storage, or for managing security policies. These are essential for automation. As with the data flows, interoperability starts with the capabilities of the component tools and services, and then moves to the subset of those capabilities used in an actual integration as an SDW.

---

6. Merriam-Webster: https://www.merriam-webster.com/dictionary/interoperability

7. OpenEXR and OpenTimelineIO are ASWF projects. Final Cut Pro X is an XML format for timelines. USD is a Pixar open-source project for 3D scenes. DCDM is a SMPTE standard for digital cinema masters. OMC is the MovieLabs Ontology for Media Creation.

In both of these areas, the capabilities of the tools and services form the foundation of the interoperability that gives studios, production teams, and integrators flexibility in building actual workflows. For example, tool support of multiple file formats enables a studio or production choice in defining its own working, packaging, and delivery specifications. Or a SaaS offering supporting multiple authentication protocols enables integration with a studio or vendor's authentication services.

The first diagram below shows the ins and outs between two tasks. The next diagram shows the use of interfaces both for accessing assets and for security and workflow automation.

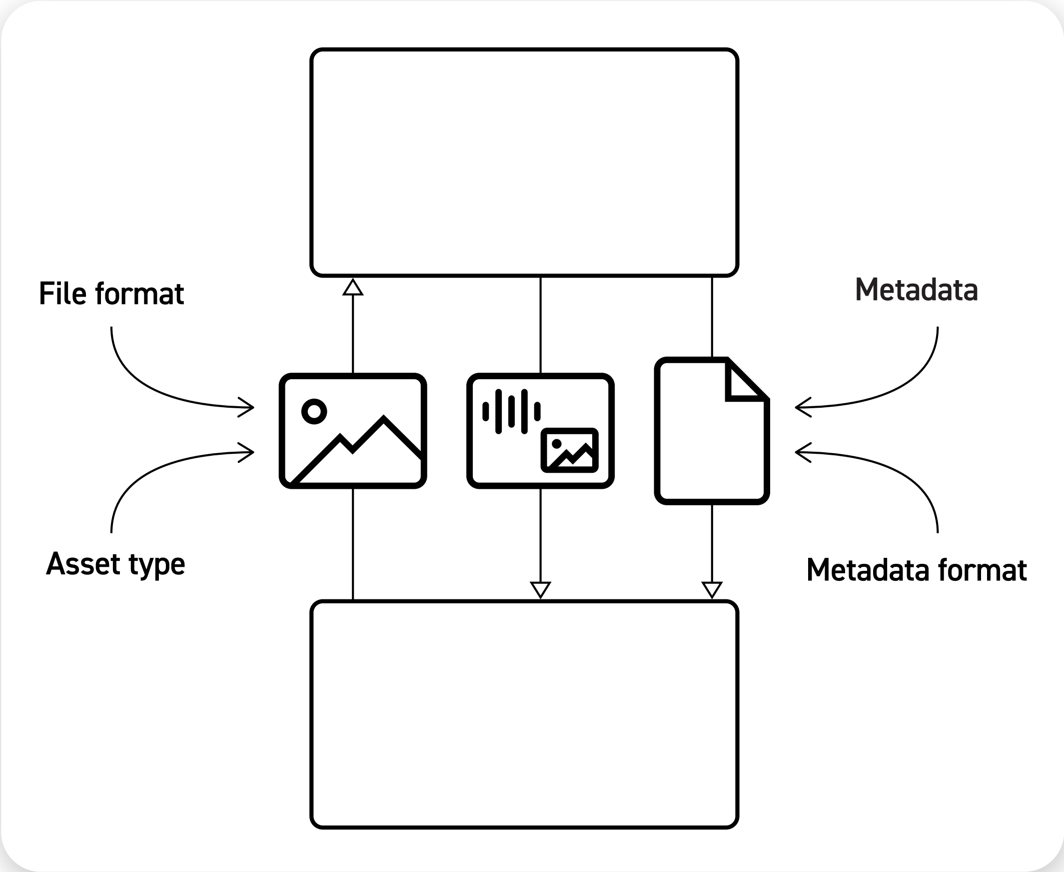

Figure 3: *Data ins and outs between two workflows or workflow components*

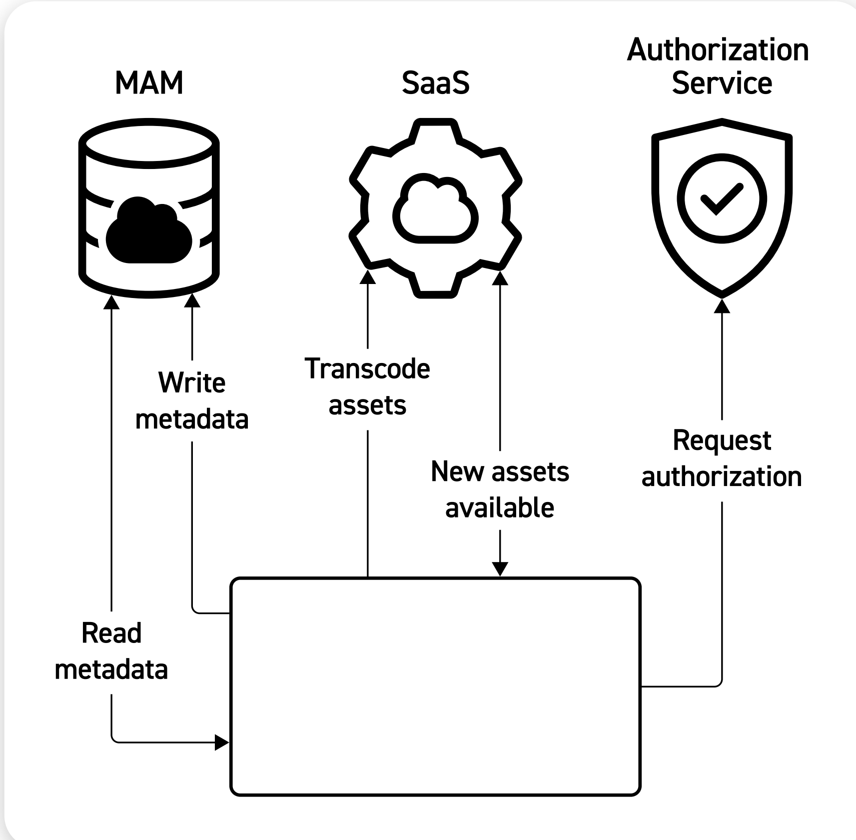

Figure 4: *Interfaces and integration points with a workflow component*

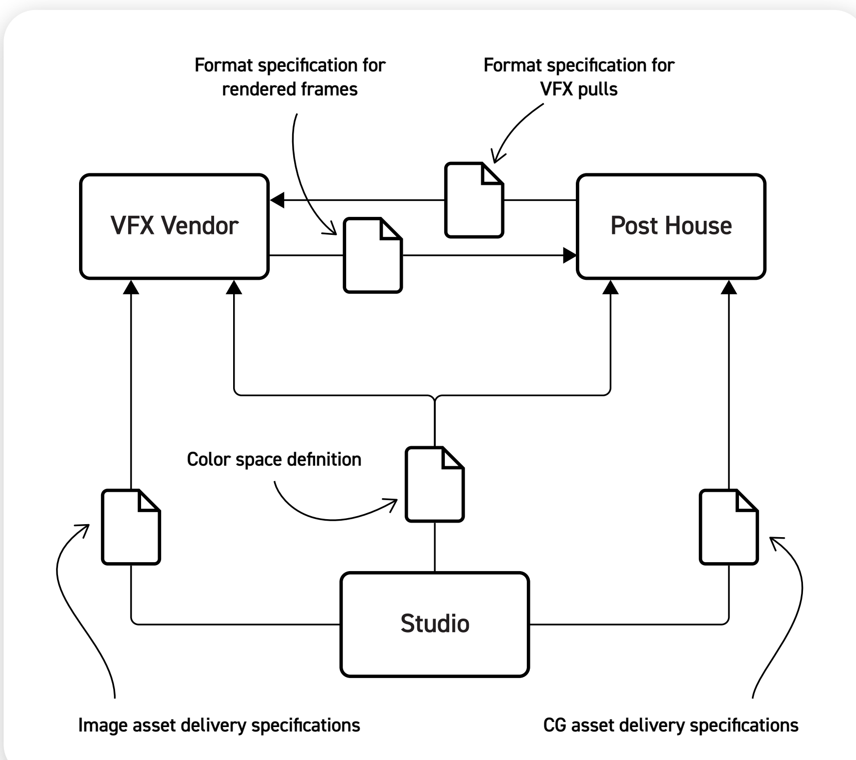

Figure 5: *Interfaces and integration points with a workflow component*

Our focus in this document[8] is on the software-based systems, tools, and infrastructure that are used in media creation. For these systems, the compatibility of the ins and outs is determined primarily by two things. First is the ability of the tools and systems to support the same or at least overlapping sets of input and output formats for assets and metadata. And second is common ways in which those formats are used, which is based on the compatibility of data models, workflow processes, and delivery/interchange specifications. As for mechanisms and interfaces, the table stakes are a set of these that is sufficient to enable tools and systems to be integrated into SDWs, even if not common or standardized. The ideal is for these integrations to be done using common mechanisms and interfaces so that the integrations are more reusable. Interfaces include things like APIs; whether to network services, databases, or other infrastructure; and other mechanisms, such as things like configuration files and exposing functionality indirectly through integration with platforms.

LEVELS OF INTEROPERABILITY

The baseline level for interoperability is translatability. Even if the asset formats are different, the output of one process can be transcoded or translated into a format the next process can understand. Of course, things can get lost in translation, so feature set compatibility is important, as is metadata preservation when assets are transcoded.

On the metadata front, common data models can enable better and less lossy translations, even when they are encoded in different formats or schemas, whether in files or in databases.

So, interoperability is a journey. It improves as one moves from lossy translations to lossless ones and from API shims (custom software connectors) to reusable integrations using common APIs. The goal of full interoperability is achieved when no translations or shims (custom software connectors) are required and integrations are reusable. This applies across the board from essence formats to metadata to messages and to APIs and control interfaces.

---

8. The interoperability of physical, electrical, and human interfaces, while also important and involving the compatibility of inputs, outputs, and interfaces, is not our main concern here.

As we move from lower to higher levels of interoperability, from the perspective of the workflow implementor, the building and recomposing of a software-defined workflow moves from something that:

1. requires the writing or rewriting of software to
2. low-code solutions that may involve modifying translation plug-ins or mapping tables and eventually to
3. no-code solutions where a simple drag and drop of the desired components builds and deploys a new workflow.

In these last two steps, AI offers significant promise for translating between data models without human semantic mappings and connecting components in a workflow via APIs without traditional code development. This can reduce the need for traditional forms of full data and API interoperability. While AI may dramatically reduce the effort required in integrating workflow components, those components will still need to expose the necessary functionality. At least for now, even AIs need integration points.

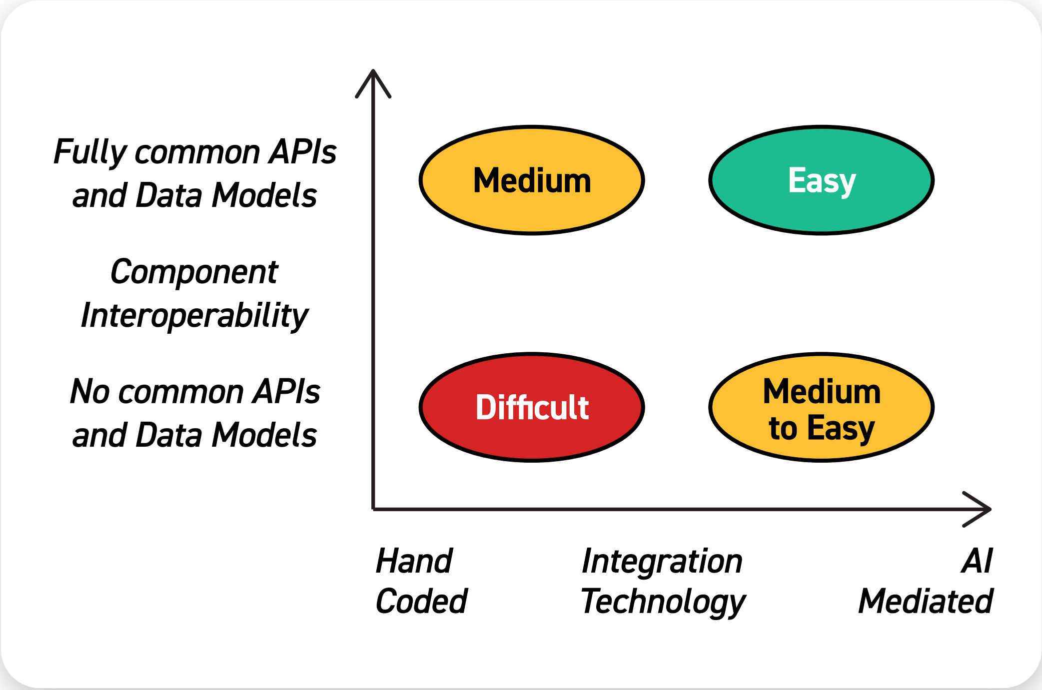

Figure 6: *Level of integration effort to achieve recomposable SDWs based on integration technology*

SECTION 4

DOMAINS OF SOFTWARE-DEFINED WORKFLOW INTEROPERABILITY

Given the large scope of interoperability in workflows, it is helpful to break the problem into parts. One useful distinction is that of the data plane versus the control plane.

DATA PLANE VS. CONTROL PLANE

By data plane functions, we mean the primary flow of essence and metadata, often in the form of files but also sometimes as data retrieved from a database using a particular schema. These include the ins and outs described above as well as the interfaces used to store and access data.

By control plane functions, we mean the exchange of the other information needed to coordinate workflow activities, like performing handoffs, provisioning infrastructure, orchestrating storage, or changing security access controls. Much of this is driven from events in the workflow, like an approval or a work assignment. These are the interfaces used to integrate and control components in SDWs.

Some operations may have aspects of both. For example, provisioning the storage for a creative session is a control plane function that uses the data plane.

DATA PLANE INTEROPERABILITY

Every major stage in a workflow is a producer or consumer of assets, and most usually both. The baseline requirement is that even if the "ins" of one task do not match the "outs" of the task(s) feeding it, they can still be translated. This applies to file formats for both assets and metadata, to media encodings, to data models, and to their encodings. In an ideal world, all would be fully compatible.

The second part of data plane interoperability concerns the mechanisms used to read and write the data, whether reading and writing data to files or to databases.

For the first aspect, that of the data itself, standard file formats are the cornerstone. While they exist in abundance, there are still gaps in the coverage of standard formats and in support for them by all of the relevant tools.

Editing tools traffic in proxies in standard video formats, such as ProRes, and for timelines, such as in file formats like EDLs or AAFs. VFX and animation use a plethora of formats for CG assets. ACES defines a workflow for managing color. The sets of these formats and the ways of using them in workflows is always evolving. In the long term, demand from users usually results in tools and translators supporting a "sufficient" set of formats. But as new technologies come into use, gaps do emerge that need to be brought to attention and addressed. We need to embrace this constant evolution.

> Instead of trafficking in file paths, we need to move to the universal use of identifiers with services that can resolve those identifiers into locations.

For the second aspect, that of reading and writing media data, file systems and the ability to read and write from them have historically been the primary mechanism for interoperable access. But in the cloud, many other forms of storage exist, and enabling interoperability across different types of storage in different infrastructures poses a significant challenge to interoperability. File paths are no longer a universal solution. To have workflows that span different types of storage, we need a way of referring to assets that is independent of the type and location of the storage for that asset.

For metadata exchange, databases with standard query languages, such as SQL and GraphQL, and the client SDKs that use them support low-level interoperability. But that is only a partial solution. The key is the higher-level compatibility of common data models and schemas, or at least ones with known translations. The first step is to ensure well-defined and compatible data models, so that mappings can be performed.

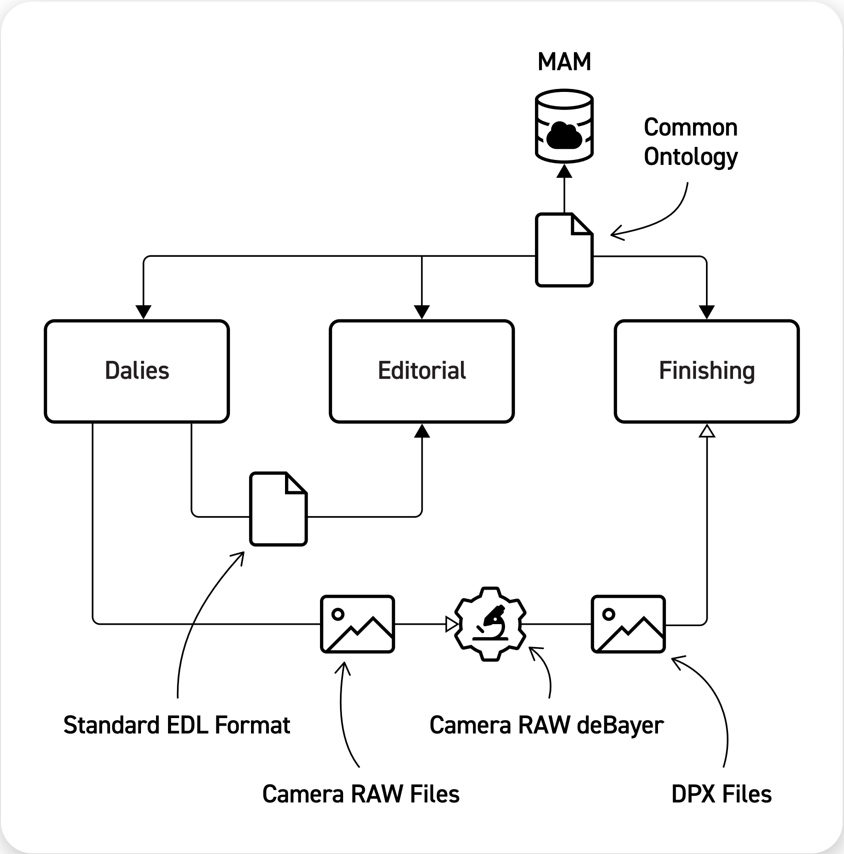

Figure 7: *Data flows of media assets, other assets, and database data*

CONTROL PLANE INTEROPERABILITY

The control plane is where automation and orchestration are controlled. Control plane functions may be invoked by a range of systems. At the simple end of the spectrum, clicking an approve button may trigger a standalone automation to "publish" dailies downstream. At the more complex end, workflow management systems may continuously track work assignments and their status, fully provisioning sessions for creative workers with all the correct assets and configurations for the productions workflow.

For SaaS applications, control plane functions are usually exposed through network APIs. In the case of workstation applications, they may also be exposed through plug-in APIs and configuration files.

> "The baseline for supporting software-defined workflows is that all the functionality required to automate the workflow is exposed for other systems to control.

The baseline for supporting software-defined workflows is that all the functionality required to automate the workflow is exposed for other systems to control. Some forms of functionality are common to many different workflows. For example, review, approve, and publish is a common pattern having common functionality associated with it, such as sending notifications and changing approval and completion status in asset and workflow management systems. And most systems that process media need a way to trigger the import and/or export of assets. But some functionality may be very specific to the workflow — e.g., for color grading, the ability to provision LUTs; or for editing, the ability to change handles.

In addition to exposing the necessary functionality, interoperable standards should be used wherever available to make it easier to swap out or recompose workflow components without reintegration work. In many cases, however, no standards exist for such APIs. In those cases, using a common data model across different workflows can at least partially mitigate the lack of fully standardized APIs, for example, in message and notification payloads.

EXAMPLES OF WORKFLOW CONTROL FUNCTIONS

- An editorial orchestration system populates bins for a SaaS editing session.
- An automated VFX pull system is notified of an approval and retrieves EDLs from a SaaS editing system.
- A color-grading orchestration system configures a grading session on a virtual workstation for an ACES workflow using a 100 nit SDR and a 1,000 nit HDR display.

Note that some of these control functions are initiated by the control plane but also use the data plane, e.g., for provisioning assets.

INTEGRATION AND WORKFLOW MANAGEMENT APPROACHES

When designing the software that integrates components into an SDW, many different approaches can be used. One important distinction is the difference between orchestration and choreography approaches to workflow automation. Orchestration tends to centralize control in a small number of workflow management systems that control the flow of assets and data across many components in a workflow, while choreography distributes the decisions and automation control to the handoff points between components, e.g., using code in DCC plug-ins. These approaches are not mutually exclusive, and most SDW implementations use a combination of both.

SECURITY INTEROPERABILITY

While security functions are technically part of the control plane of an SDW, their importance and pervasiveness places them in a category of their own. Security functions are necessary to enable the automated provisioning of user identities, tying services to authentication systems and authorization systems, provisioning of security policies, etc. As workflows move to using Zero Trust approaches that require authentication and authorization checks upon each access to assets and services, low-friction security interoperability is absolutely essential to managing integration efforts and costs.

MULTI-INFRASTRUCTURE

The 2030 Vision definition of cloud infrastructure includes not just public cloud, but also private cloud and on-prem systems that are used remotely or integrated with off-prem workflows. Consequently, the interoperability described in this document must function in a multi-infrastructure environment. On the data plane, applications need to be able to access assets and metadata anywhere they are, whether that means applications accessing them across infrastructures or applications being able to run on any infrastructure to be close to large assets. Similarly, control plane functions also need to be able to span infrastructures.

SECTION 5

FUNDAMENTALS INTEROPERABILITY PRINCIPLES

The first set of principles are fundamental in that they improve the interoperability of workflows, even ones that are not connected using software automation.

INTEROP PRINCIPLE 1: ASSET ACCESS

Assets are securely and directly accessed by authorized participants where they are across different applications, XaaS systems, and clouds.

The applications-to-assets model in 2030 Vision Principle 2 outlines the optimal case in which applications and services have the ability to access assets wherever they are without having to move or copy them.

First, this means applications and services may need to be able to directly access a range of cloud storage types, including across infrastructures, instead of assuming things are always on a filesystem.

Implicit in giving productions the ability to choose the components best suited to their needs is the choice to use a component that may only be available in an infrastructure different from the one where assets, metadata, or other components are located. So, this ability to access cloud storage needs to be able to span infrastructures, including on-prem, public cloud, and XaaS offerings.

Second, this has implications for storage systems and XaaS systems that internally store assets. To fulfill the apps-to-assets principle, they need to expose direct access to the assets they host, not just the ability to move assets in and out. And when the assets are stored outside of the XaaS system, those systems need to support direct access to assets stored on different infrastructures outside of their own systems.

When performance requires an application or service to run “close” to the assets, e.g., in the same infrastructure, one method to achieve this is for the application or service itself to be able to run in any infrastructure, e.g., because it is implemented and containerized in a way that it can be deployed in multiple infrastructures. In some cases, the integration of components and storage running in different infrastructures will pose trade-offs between performance and the creation of cached copies. Resource utilization may also impact the choice of when and what to copy. The key is that the decision to copy should not be driven by a lack of interoperability across infrastructures.

There are multiple approaches to facilitating this storage interoperability, including server-side abstractions, client-side abstractions, and direct integrations. The best approach will vary based on the use case, and some will likely require limited replication and caching.

Every asset access by or through a service must be checked for authorization. This security must also interoperate for all types of storage, whether accessed directly, proxied by a XaaS offering, or accessed internally by it. The use of infrastructures across organizations poses additional security interoperability challenges. Security is covered in more detail below in Interop Principle 6(b).

INTEROP PRINCIPLE 2: COMMON DATA MODELS

Applications use common data models to preserve and propagate data.

As discussed above, common data models are a foundation of data interoperability. While it would be great and highly interoperable if everyone and every tool used the same concept in the same way and with the same name, we at least need to have the same concepts if we want to be able to map them across systems. A common data model provides this conceptual unity, even when different encodings are used.

Many terms used in the creative process mean different things in different circumstances; "scene," for example, means somewhat different things during script breakdown and in filming and editing. Similarly, some concepts are referred to in multiple ways; "take," "revision," and "cut" all relate to what can loosely be thought of as versions of something. Ontologies formalize these concepts and the terms used for them, reducing human confusion and permitting machine-based interoperability.

Ontologies provide some common data (or metadata) elements for the objects they define, allowing applications that create or use those objects to communicate information that is important to managing the workflow and exchanging assets.

Finally, ontologies also provide a formally defined set of relationships between things, whether those things are concepts from a script or assets that are created as part of the production process. These relationships provide context for these objects, such as knowing that an asset is related to a particular scene or is an element of a completed CG model.

All of this is done within the scope of a particular domain, but, even then, an ontology cannot define everything in all cases. A good ontology constrains the scope to avoid being too complex. The concept of "connected ontologies" allows extensibility into areas where more detail is needed or where a more detailed data model is needed for areas that touch upon the domain of the ontology but have their own specific needs.

With all this in mind, the MovieLabs Ontology for Media creation (OMC) does this for the production process, starting with script breakdown and going through to post-production:

- Terms and concepts are formally defined.
- Data elements sufficient for handoff between parts of the production pipeline are defined for each concept.
- Relationships are defined, showing connections and dependencies across the elements of the production.
- Connected ontologies are created for areas where more granularity is needed, such as for recording camera metadata and managing the handoff of Audio and CG assets.

Because of the formal definition of the ontology, it can be used as a translation between different systems' data models, including other ontologies. It also can form the basis for the definition and adoption of common schemas, which facilitates the next level up in interoperability, especially when used throughout a workflow.

INTEROP PRINCIPLE 3: ASSET AND METADATA DELIVERY

The assets and metadata required as inputs to and available as outputs from workflows or workflow components use common formats and are fully described in delivery specifications.

Asset interoperability is rooted in the ability of multiple tools and services to generate and understand the same asset formats, which in turn depends on the use of well-defined common formats. In order to determine whether the outputs of one workflow or workflow component can be connected as the inputs of another, those inputs and outputs need to be fully described in delivery specifications. When specifications match, components or vendors can be easily switched while maintaining the continuity of the workflow. And when they do not match, they can be used to determine the feasibility of translation and transcoding.

INTEROP PRINCIPLE 3(a): ASSET AND METADATA FORMATS

Assets and metadata exchanged between organizations or intended for long-term use are in publicly documented formats and schemas.

The interoperability of assets is most important in exchanges across organizations or departments or when assets need to have a long shelf life.

There are many interoperability gaps today, and innovation continuously creates new ones. Proprietary formats or practices endure in some parts of workflows, e.g., for things like camera RAW files and character animation rigs, which have been tightly coupled to technological innovation of sensors and animation techniques, respectively. When such assets are exchanged between studios and/or among vendors, access to the appropriate proprietary tools is necessary for interoperability.

Often, trade-offs exist between interoperability and innovation. The use of extensible formats that can provide both baseline interoperability and room for innovation is one path by which new capabilities can be supported as extensions and perhaps eventually be incorporated into the format specification. Open source as an alternative to traditional standards can also help interoperability and innovation coexist.

Media Assets

Media or essence assets tend to be among the most standardized, including formats for things like frames or images (OpenEXR, DPX, J2K, …), video (ProRes, MP4, ...), audio (WAV, MP3, …), models (USD, FBX, …), etc.

Also, since most media formats support some form of user- or vendor-defined metadata, it is important when evaluating interoperability to consider not only whether the essence itself is standardized, but also whether any additional metadata carried in the file also follows a common practice, such as the MovieLabs Ontology for Media Creation (OMC).

Other Structured Assets

Other file-based assets include things like timelines, LUTs, metadata sidecar files, and various decision lists. These tend to be less standardized than the media formats themselves. And a documented format may often be used but with custom metadata. The same needs exist here as for metadata within essence files.

Today, most tools and devices support a range of standardized and proprietary formats for both media and structured metadata. Ideally, all of the key types of these assets would be covered by standardized or open-sourced formats, and tools would implement those formats following publicly documented practices. These formats and the implementations may in some cases be “lossy” compared to a tool’s internal format. The key is that all of the functionality that is needed downstream or for the long-term usage of the asset is captured in the open formats.

Databases and Other Data

Because of their many benefits, including enabling collaboration and workflow automation, databases are increasingly used for storing metadata that historically only existed in files.

Compared with metadata in files, which is often read and written by off-the-shelf tools, the interoperability of metadata in databases is often more challenging. The schemas are often custom designed to the particular needs of an application or workflow. For example, both editorial and VFX need to deal with the characters and props in each shot, but VFX likely needs to track many different variants for those characters and props. The workflow needs for improved metadata interoperability occur at handoff points like this one between departments or vendors. It is here that common data models for exchange, like the OMC, can help, especially for things like asset management and where metadata is imported and exported.

INTEROP PRINCIPLE 3(b): ASSET REFERENCES

Assets are referenced by identifiers that are independent of location and are resolvable to asset location by receiving applications and systems across multiple clouds.

Composable workflows need a way of referring to assets that is independent of an asset's location and of any particular MAM (in a world where assets are shared rather than copied, there will be many systems tracking the same asset). This allows for separation of metadata and assets and allows applications to be less aware of the mechanics of storage management and orchestration.

For this kind of interoperability, we need two concepts:

- Identifier: an unambiguous way of referring to an asset independent of its location. This is not a new idea — it is the basis of UUIDs, UPCs, ISBNs, EIDR IDs, and so on.
- Resolution System: a mechanism for turning an identifier into an actionable, usable location.

When an asset is created, it gets an identifier. This identifier is registered with the resolution system with a URL, which is used for accessing the asset. An identifier can have more than one such URL, e.g., for different formats, metadata versus asset essence, and so on. For full benefit, the same identifier should be useable and used throughout the workflow, and even across organizations. But it is also possible to have multiple identifiers referring to the same asset.

Most current systems conflate assets and their metadata, which can result in suboptimal solutions for managing both. For example, in production workflows, metadata is added to assets over time, especially information about how and where an asset is used and how it connects to other production elements. This is beyond the capabilities of many storage management systems, which are intended to manage files or file equivalents efficiently. Similarly, while shared databases are often a good place for dynamic metadata, they generally are not good at managing bulk storage for large assets.

Currently, this problem is partially addressed by specialized asset management systems (variously called MAMs, DAMs, and so on). These systems keep the asset and its metadata tightly coupled, trying to manage both efficiently. However, this presents some implementation issues both now and for the 2030 Vision:

- Traditional applications that use assets tend to want to have something like a file or a downloadable "object." The application will often refer to an asset with a file path, which is only a portable construct if there are detailed and enforced policies and practices for the production that work across all applications.
- If these applications use assets in asset management systems, they require a new integration with each asset manager.
- Some assets, such as reference material or library assets, might have different metadata based on a particular department or organization, such as films in which an asset was used in an archive or library, rights clearances in the legal department, and outbound licensing in the marketing department. These departments may have their own preferences for which MAM to use.
- Large assets are expensive to move and store, and a production can have very specific notions of where assets are best stored. Forcing asset storage to be done by the MAM results in data movement and potentially storing an asset in multiple MAMs. MAMs that operate with "bring your own storage" are more in line with the 2030 Vision.
- Applications and data structures that reference assets from more than one source become complex and fragile — if such can be done at all.

In the 2030 Vision, applications communicate using a common system for linking assets with each other and their locations. Identifiers are essential to that linking, and identifiers are resolved to a location only when necessary. As an example:

- An image asset is created by an artist.
- As part of the approve-and-publish cycle, the asset is given an identifier and stored in the cloud. The resolution system is made aware of the location.
- Metadata for the asset (provenance, approval state, relationships to other production elements) is stored in an asset manager using the same identifier.
- If another application asks the asset manager for the asset itself, it uses the resolution mechanism to return the asset's location.
- When the asset's metadata is returned from the database, it includes the identifier so the requester can get the image if it needs to.

Many of these steps can be collapsed together, and asset managers can also act as resolution systems, but the underlying mechanism is the same.

Identifiers have the added benefit that they make managing relationships across assets easier. If an asset is referenced only by its location, when it moves, the relationships involving it have to be updated, whereas if the relationships use identifiers, they remain valid no matter where the asset is stored.

There are several considerations worth keeping in mind when transitioning to resolvable identifiers:

- There is a straightforward path from the use of asset locations to the use of identifiers; just use something based on the initial asset location as the identifier. Implicit metadata from the path can be extracted and stored with more interoperability in a database with a known schema, e.g., using the model provided by OMC.
- Versions are best managed as asset metadata or as new assets with connections to previous versions; the resolution system does not need to know about them, though an asset management system will.
- This mechanism is used inside and across organizations, so the resolution system and actual data access both need fine-grained access controls.

Interoperability of resolution systems is important, especially across organizations. This could be managed through common APIs presented across resolution systems and asset management systems or via sets of plug-ins. For some thoughts on this and on implementation paths, see the 2030 Vision blog at movielabs.com.

INTEROP PRINCIPLE 4: PREFERRED TOOLS AND INFRASTRUCTURE

Tasks in a workflow are easily changed to use different applications, XaaS systems, and clouds.

This principle embodies getting to a place where studios and production teams can readily choose to use the tools, workflows, and vendors that they believe are best suited to a particular production with minimal risk to schedules and budgets. It is one of the primary goals of interoperability and a realization of the dynamic creation of workflows from building blocks covered in 2030 Vision Principle 9.

Getting to this point depends on many of the other principles outlined here. On the data plane, it depends on the interoperability of the "ins and outs" using common file formats, data models, and data encodings as well as the mechanism for accessing those elements in storage systems and databases. On the control plane, it depends on the interoperability of the integrations of tools and services into software-defined workflows, which involves common workflow patterns and the messages and events that support them.

SECTION 6

SOFTWARE-DEFINED WORKFLOW INTEROPERABILITY PRINCIPLES

These principles build on the fundamental principles and add additional requirements and detail on what is needed to support their basic integration into SDWs and then to do so in an interoperable way that enables the recomposability of workflows and the reusability of integrations.

INTEROP PRINCIPLE 5: DATA AND CONTROL PLANE MECHANISMS

Applications, Services, and XaaS systems enable data exchange and control through well-specified common mechanisms, including across organizations and clouds.

Software-defined workflows envision workflows with seamless exchanges of assets and metadata and the automation of menial, non-creative work. This requires software that integrates each tool and service into the desired workflow. These integrations rely on mechanisms that can include:

- Configuration files.
- APIs exposed by tools and services for external systems to integrate with.
- APIs exposed by platforms and infrastructure for tools and services to integrate with.
- Plug-in APIs that allow customization of the way applications or services interact with infrastructure.
- Tools or other mechanisms exposed by platforms that may be invoked from scripts, plug-ins, or other tools.

The functionality that needs to be accessible through those mechanisms varies for each workflow and task. But there are some broad categories of the capabilities that application, services, and XaaS systems need to provide, including:

- Reading and writing file-based assets across the set of infrastructure and storage types required by the workflow. (This is the software instantiation of Interop Principle 1.)
- Reading and writing metadata across the set of infrastructure and data storage systems required by the workflow.
- Enabling third-party systems to drive the provisioning of sessions.
- Enabling third-party systems to set up and trigger services to process metadata and assets.
- Giving third-party systems access to changes in the state of work.

The mechanisms used for these integrations may vary based on the needs of the workflow and on implementors' engineering choices. For the resulting workflows to be flexible and recomposable, agreement on common mechanisms and patterns is essential. In our vision of interoperability, all of these capabilities are built on systems providing common, documented mechanisms (APIs, interfaces, work descriptions, etc.). The interfaces should be described in machine-readable forms, e.g., OpenAPI, to facilitate validation. And, of course, when data exchange is involved, they should be based on common data models.

INTEROP PRINCIPLE 5(a): EVENTS AND NOTIFICATIONS

Applications, Services, and XaaS systems enable the exchange of relevant notifications with third-party systems, including across organizations and clouds.

Workflows are inherently driven by events such as the completion of a task, replacement of an asset with a new version, approval or return of work, and changes in editorial timelines. Any system that is tracking work and related metadata must provide the ability for interested participants (whether human or machine) to be notified when something they care about changes.

Take, for example, a common pattern in production workflows, that of an approval leading to the propagation of the assets and metadata necessary for the next task. With the 2030 ideal that assets do not move unnecessarily and that propagation is a publish (rather than a copy) operation, folder watches are no longer an option. Instead, asset and work management systems can enable downstream systems to subscribe to publication events.

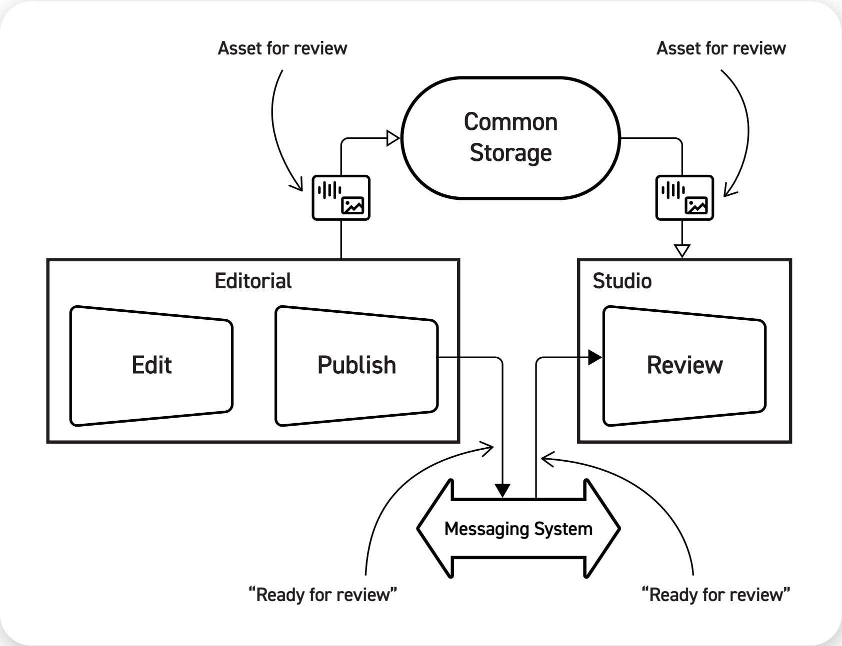

Figure 8: *Cross-organizational publish function using a messaging system for notifications*

Different data flow patterns could be used to generate the notification. The component making the state change could also send out the notification as in the diagram above. Alternately, if a workflow data platform maintains approval status information, that platform could provide notifications to interested subscribers when status changes on particular assets. For more complex downstream tasks with multiple dependencies, workflow-savvy services listening to those status changes could determine when all the prerequisites for a downstream task are ready and then prepare a delivery. Many data flow patterns are possible to get to that handoff point. Agreement on which to use in common situations would improve interoperability among those automation components. In the meantime, focusing efforts on the handoff points and specifically on the use of common data models at those interchanges is a good first step.

Reliability in the delivery of notifications must also be considered. Especially when notifications are being sent across organizations, the sender may not know and cannot rely on downstream system availability. For this reason, the reliable delivery of notifications is important. Many workflows also rely on the ordering of events for correct processing. In these cases, maintaining sequencing is important. Many event busses and messaging systems can provide this reliable and ordered delivery of events and messages when needed. This design consideration is particularly important for tools and platforms that intend to support a variety of use cases.

INTEROP PRINCIPLE 5(b): SECURITY

XaaS systems enable control of security, including authentication and authorization, by third-party systems.

Part of the control plane of an software-defined workflow relates to security. "The Evolution of Production Security"[9] discusses the need to move beyond perimeter security to Zero Trust architectures in which every service request and asset access is authenticated and checked for authorizations. The functionality needed to do this encompasses several different use cases, from basic authentication and authorization of API requests to initialization and provisioning of users and policies managed within SaaS offerings or network-based services.

Note that many software components, like libraries or standalone applications, have historically relied on external security, e.g., security provided by their operating environments. While this may remain true in some cases, in a world of cloud-based services, many of these components will serve as services or clients to services — for example, a library that originally only accessed files may now need to access cloud storage, or a tool that was invoked only locally may be packaged as a cloud service. These components need to support mechanisms for authentication and authorization.

Our recommended architecture for integrating this Zero Trust security into SDWs is documented in the MovieLabs Common Security Architecture for Production (CSAP).[10]

---

9. https://movielabs.com/prodtech/security/ML_Securing_the_Vision.pdf

10. https://mc.movielabs.com/docs/security/

And as with SDWs, interoperability here is not just about getting the functionality exposed; it is also about exposing it using standardized mechanisms wherever possible. The good news is that standardized protocols are much more mature for security, especially authentication, than for workflow automation. But ideally, many of the integration points, such as those documented in CSAP, would be standardized as well.

Any SaaS offering, web application, or other system providing network APIs needs to support authenticated access to those APIs. And most of these offerings will need to provide different authorizations and access to different users within the offering itself. This in turn raises issues of how users are provisioned and how authorization policies are controlled. Historically, many SaaS offerings relied primarily on GUIs for manually configuring users and security policies.

> A fully software-defined workflow requires that workflow provisioning and initialization of security, as well as authorization changes resulting from workflow execution, be supported by automation.

Authentication Interoperability

Standardized protocols exist to support federated authentication. While most SaaS offerings support an internal or native authentication service, the use of studio or vendor identity systems is often a requirement in production workflows. Systems need to support authentication across infrastructures and organizations. Fortunately, standardized protocols, such as Open ID Connect, exist for integrating third-party identity systems with SaaS or web application offerings.

Regarding initialization, any SaaS offering with authorized users needs to provide an API mechanism for managing which users may be authenticated.

Authorization Interoperability

As with authentication, authorization needs to be interoperable. Systems need to support authorization checks across infrastructures and organizations.

Regarding initialization, any SaaS offering with authorized users needs to provide API mechanisms for managing the authorizations of those users, including any security user groups maintained in the offering. Most production workflows need granular access controls. And in many cases, these policies need to be dynamic, e.g., to support publish-in-place models or to support productions implementing workflow-driven, least-privilege approaches. SaaS offerings also need to provide mechanisms for this.

Baseline interoperability of these mechanisms requires a common understanding of how participants may be authorized based on their individual identity, their roles, or other associated attributes; and how the resources that they can access are also specified. Just as metadata exchange and workflow interoperability need a common data model, here we need a common security policy model so that components controlling security authorizations can at least use common concepts across different services and infrastructure. Consequently, as a foundational step, we need to develop a common way to think about and define the things upon which authorization policies are based.

For full interoperability, those mechanisms and APIs should use common machine-readable expressions of policies. The use of common policy description languages can improve the interoperability systems implementing authorization policies and those controlling them. Very powerful and general authorization policy languages exist, such as Open Policy Agent, but a constrained subset or simpler languages that can be mapped to multiple implementations may provide easier paths to interoperability.

Examples of Security Interoperability

- A SaaS offering uses OpenID Connect (OIDC) to support federated identity providers. A studio provisions the SaaS offering to use its OIDC-capable servers for authentication.
- A SaaS offering maintains a set of assets and provides APIs for authorizing which users have access to which assets.
- A SaaS offering uses SCIM (System for Cross-Domain Identity Management) to provision group membership from a customer's systems.
- A containerized microservice that does not implement security itself is embedded in a service mesh which uses mTLS to mutually authenticate with other services running within it.

INTEROP PRINCIPLE 6: REUSABLE INTEGRATIONS

The integration of each component is reusable, e.g., by utilizing common APIs & platforms and minimizing custom point-to-point integrations.

Reusable vs. One-Time Integrations

Ideally, our software-defined workflows would be fully recomposable and replacing one component with another providing the same workflow function would be completely plug and play. One way to realize this would be fully standardizing the data plane in terms of data flows and file formats and fully standardizing the control plane in terms of APIs. Unfortunately, as we have observed, very few standards exist for the control plane outside of security authentication and authorization systems and many control plane functions are specific to individual tasks.

Nevertheless, a number of approaches can improve reusability.

We have already discussed the importance of asset interoperability. The use of common industry formats that are either formal standards or defined in open-source projects enables the outputs generated by one component in the workflow to be used by the next component in the workflow. Examples include standards like IMF and formats defined by open-source projects, like OpenEXR, OpenTimelineIO, and USD. These make it possible to directly plug in another service or tool that uses them.

Reusability starts from a foundation of common data models (if not common encodings) and common patterns and data flows (if not common interfaces). For example, one could use a common ontology for the data model but not necessarily use standardized encodings of it. The common data model enables components to be reintegrated more easily, but the software connecting them may need to be ported to map to different encodings and interfaces.

The engineering effort for reintegration can be further decreased by using common encodings for metadata exchange. For example, messages sent to notify a vendor that work has been published could use common message payloads to encode the metadata about the publication. Such message payloads could easily be bridged to different messaging systems used by different vendors.

Full reusability becomes feasible when integrations are built using common APIs and common platforms. Reintegration can be as simple as pointing an implementation to the URL of a different service. These common APIs can be defined in standards, such as ones from SMPTE for media microservices, or open-source software projects, such as the ASWFs OpenCue.

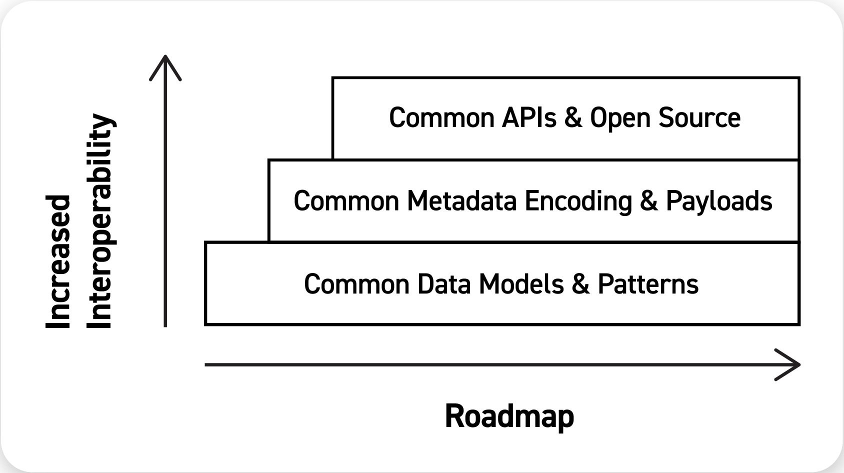

Figure 9: *Building interoperability*

Different architectural patterns can be chosen for the integration of components into SDWs. Workflow automation patterns range from centralized orchestration to decentralized choreography. Point-to-point integrations tend to be less reusable, especially when using component-specific mechanisms or patterns.

> **“Workflow automation patterns range from centralized orchestration to decentralized choreography.**

Platforms provide one alternative to point-to-point integrations. These include integration platforms as a service (iPaaS) that facilitate the integration of different services into event-driven workflows, as well as common data platforms that track metadata about work, assets, and their relationships and can provide notifications to drive automation. Consider a simple create-review-approve-publish workflow pattern. The SDW could be implemented with point-to-point communication between tasks as shown in Figure 10.

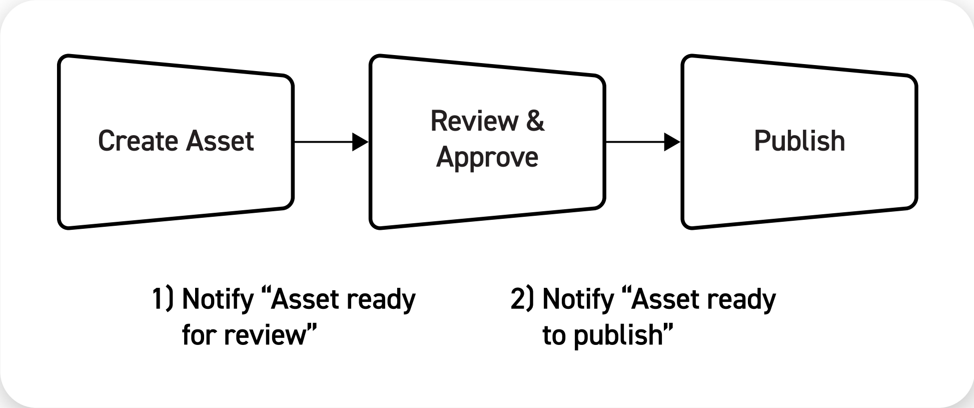

Figure 10: *Notification flow using point-to-point notifications*

Or the same workflow could be implemented using a data platform that tracks the status of assets, and then the services supporting the tasks receive notifications of changes to the status of particular assets.

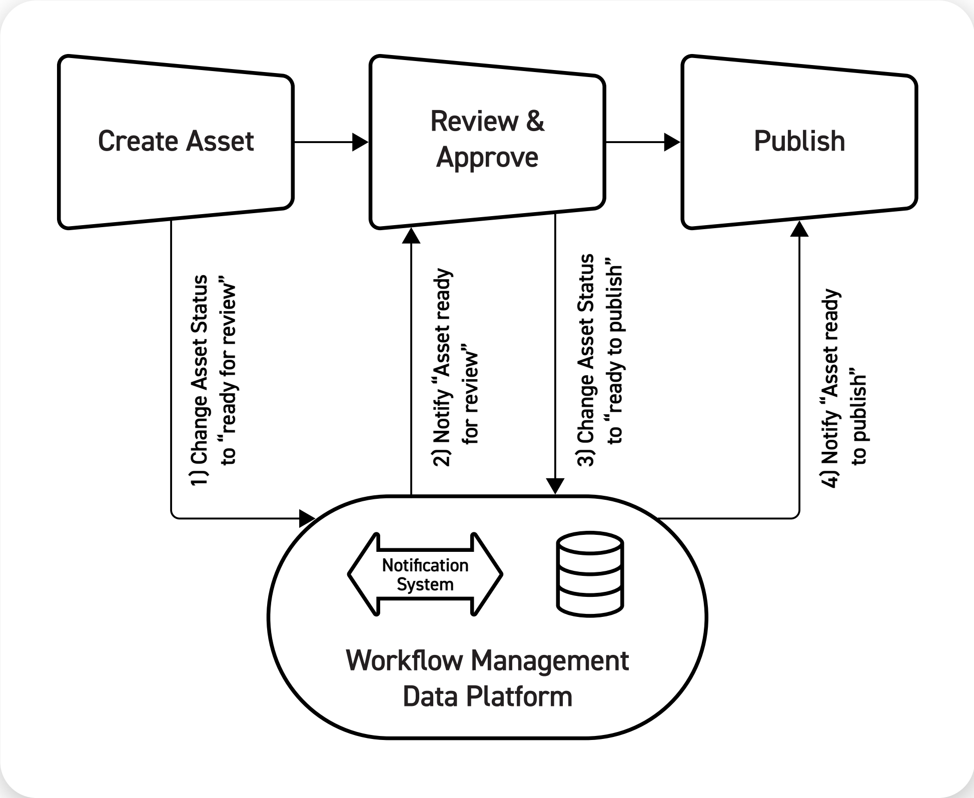

Figure 11: *Notification flows using a common workflow management data platform*

When tools and services communicate through a common platform that is based on a common data model, they can more readily be recomposed than if the integration relies on custom point-to-point connectors.

The use of common security models and protocols for authentication and authorization with the platform is also essential to reuse of the integrations.

“When tools and services communicate through a common platform that is based on a common data model, they can more readily be recomposed than if the integration relies on custom point-to-point connectors.

INTEROP PRINCIPLE 7: AUTOMATION OF REPETITIVE TASKS

Non-creative, repetitive tasks are automated.

In the 2030 Vision, people are freed from working on mundane, repetitive tasks to provide more time and energy for the creative work that adds the most value to the final creation. The automation of these tasks not only frees people from repetitive work, it also enables them to complete tasks more quickly or to explore more options. For example, automation in turnarounds can reduce the time creatives spend waiting for results. In many cases, this enables the realization of the rapid real-time iteration anticipated in 2030 Vision Principle 10. Many of the technical building blocks that enable this are already captured in the previous principles, such as event-driven notifications that can trigger automation. The next two sub-principles capture two additional use cases that facilitate this automation.

“For example, automation in turnarounds can reduce the time creatives spend waiting for results.

INTEROP PRINCIPLE 7(a): AUTOMATED SESSION PROVISIONING

Upon completion of a task, the next creative tasks that use the results are fully and automatically provisioned so that work can be started, e.g., triggered by notifications of the completion.

This principle envisions workflows in which time spent waiting to start the next task is minimized through automation. For example, the automation of handoffs means that downstream tasks can be started immediately, potentially multiple in parallel.

The completion of one creative task usually involves a change in the status of assets, as assets have been created, approved, or annotated. A simple example are the tasks in a review, approve, and publish workflow. Upon completion of an editorial change, streaming assets are automatically generated for review and the reviewers notified. When the resulting work is approved, it can be published to the next step in the workflow.

Some session setups are more complex, requiring the organization of assets into complex bin or folder structures. In those cases, a deep understanding of the relationships and versioning of the assets is often needed. The use of a common data model that includes the interrelationship of those assets can enable automation to manage that asset organization.

INTEROP PRINCIPLE 7(b): AUTOMATED WORK ORDERS

Work is described in clear, unambiguous sets of instructions for automated execution.

To automate work, the nature of that work needs to be defined. In some cases, that definition may be hard coded. The approval of a particular output from a particular task could trigger a hard-coded set of operations, such as transcoding or the generation of proxies. But many tasks are more complex, involving various interrelated assets, e.g., provisioning a VFX artist's session, or tasks may occur across organizations, e.g., ingesting a complex set of assets.

When this work is described in machine-readable forms, the automation described in Interop Principle 8 can happen without someone with knowledge of the workflow having to get involved. Automated work orders also help with the tracking of work.

INTEROP PRINCIPLE 8: TRACKING AND DASHBOARDS

Work is described and tracked so that progress, resource utilization, and costs can be continuously determined.

The use of software to support and automate workflows can provide greater visibility into what is happening across different workflows.

The tracking of progress and remaining work enables quantifying risks to schedules and budgets. Various mechanisms can support this. The review and approval status of assets can be tracked in asset management systems. Adding in the relationships between those assets and contexts, such as scenes, shots, sequences, characters, props, etc., can then lead to a more comprehensive understanding.

When tasks and work assignments are tracked in workflow management systems, human resources can also be better understood.

And at the infrastructure level, both the historical and future resource utilization can be understood. Both require an understanding of how work, completed or expected, maps onto resources. As systems become smarter about automating the management of storage and computing resources across infrastructures, the impact that workflow choices have on resource utilization may not be easily understood by the people making those choices. Especially with SaaS systems or production-as-a-service platforms, providing visibility into those impacts is fundamental.

SECTION 7

CALL-TO-ACTION

The 2030 Vision of software-defined workflows seeks to support both creative work and enable automation of non-creative, routine and mundane tasks, while also enabling the seamless composition and recomposition of workflows that utilize preferred tools and infrastructure. This ability to reconfigure workflows is especially essential to supporting constant innovation as technology and workflows continue to evolve rapidly.

This interoperability does not come free. The work involved to develop and implement common data models, common encodings, and common interfaces for the many different tasks in media creation workflows is non-trivial. Developing portable applications and services almost always takes more effort than targeting a single platform. And using portable interfaces and formats sometimes involves making a trade-off between utilizing every feature of a specific tool, service, or infrastructure and being able to easily replace it with another.

Meeting the basic goal of enabling SDW integration will require an immense amount of work by all providers of tools, services, SaaS offerings, and infrastructure to expose functionality in interfaces and integrate with platforms and infrastructures. The ultimate goal of doing this in a flexible, recomposable, and non-brittle way is even larger in scope. The good news is that we don't need to wait until all opportunities for interoperable integrations are realized to see benefits. We will see improvements in workflow flexibility and productivity with each added locus of interoperability, and the benefits will only accelerate as more partners come in. So, where do we start?

IDENTIFYING THE PAIN POINTS

Media creation workflows include a large number of diverse tasks and tools. Our first step is to identify where the interoperability needs are the greatest. This has two dimensions. One is which workflows can most benefit from the increased use of software for control plane automation. We can then prioritize the tools and services in those workflows to provide mechanisms that support their integration into complete software-defined workflows. The other is where are recomposability and reduced switching costs needed the most, for example to enable choice or to support innovative workflows. We can then prioritize these specific tasks for improved interoperability on both the data and control planes.

Common ontologies, and their encodings, provide a foundation for data interoperability. Focusing on their use at interchange points in workflows enables a fast start to interoperability and that can then move upstream and downstream from those interchange points to support common patterns like review, approval, package preparation, publication, and session provisioning.

USING COMMON DATA MODELS

To achieve interoperability on the data plane, we need to start with data models. These underpin everything.

The first step is to ensure that data models and schemas are well-defined and documented, so that we can understand where incompatibilities exist and where mappings between them can be created. To avoid the friction and poor translations caused by incompatible models, the next step is for the industry to agree on common concepts and data models to describe, at least at a high level, the many relationships between tasks, assets, participants, and infrastructures. This includes tying them back to the context provided by script, capture on set, and all the additional work that follows in post.

To do this, we need to expand the coverage of broad data models, such as the MovieLabs Ontology for Media Creation,[11] and also connect them to the data models in commonly used specifications, standards, and open-source projects that can go deeper into very task-specific metadata and work in processes.

---

[11]. https://movielabs.com/omc

USING COMMON DATA ENCODINGS

Once data models are agreed upon, we need to move on to common encodings of them to fully realize data interoperability. These can then be used to standardize things such as payloads for messages and data platform APIs.

DEFINING COMMON PATTERNS AND DATA FLOWS

In conjunction with this, we need to get agreement on data flows, starting with the interchange points between major tasks and what control functionality is needed to support automation at those handoffs. Agreeing on the data flow of common patterns, such as review-approve-publish, can provide configurable building blocks that can be reused in many workflows. Even if the interfaces are not identical, common data flows make software shims and mappings possible.

And we know that, to achieve interoperability across infrastructures, we need to move away from relying on file paths and locations to refer to assets. All tools need to enable a new access pattern based on resolvable identifiers.

ENABLING REUSABILITY

Reusability is built on the commonality of many things: assets (both essence and metadata), workflow patterns, data models for workflow and asset management, encodings of those data models, data platforms to support them, messages, event payloads, etc. So, the actions above provide a foundation, but how do we put the pieces together?

The role of asynchronous communications systems is another area with significant promise. Workflows are event driven, and many data flow patterns need reliable, ordered, asynchronous delivery across organizations and infrastructures. One path to addressing this is agreement on common publication and subscription mechanisms that are secure and can be easily configured for use with different end points.

Open-source projects are another powerful approach that allows us to start with iterative explorations of solutions and then, when successful, provide all solution developers with key implementation components and iterate at a faster pace with many contributors building additional functionality.

Common platforms offer the promise of a one-time integration, at least for the aspects covered by the platform. We need to identify where this is most feasible, e.g., agreement on data platform patterns and interfaces to enable the development of data platforms that interoperate not only with tools and applications but with each other.

ENABLING INTEROPERABLE SECURITY

The support of federated identity systems is table stakes for a SaaS offering. The largest gaps in security interoperability for SaaS offerings are in the areas of automated provisioning of users, roles, and authorization policies. Especially for the latter to be interoperable, we need common understandings of the elements of a security policy and ways to make those policies more interoperable, even when using very different underlying infrastructures. A common data model or even a common language could be a solution to improving the interoperability of workflow management systems with disparate security systems.

Also, many software components were designed to rely on perimeter security. Everywhere a local service or resource is being replaced with a cloud one, we need to ensure authentication can be provided and access to resources controlled by authorization policies. We need to work to make this a priority for both commercial and open-source software.

WORKING TOGETHER

Ultimately, all of these elements serve the end goal of enabling people to build flexible production pipelines that can easily be instantiated in different configurations to support the needs of productions and their creative teams.

All participants in the ecosystem can benefit from improved interoperability; achieving it will require working together, iterating on designs and implementations to build the path forward. Everyone has a role to play in this effort, and the gains will also be shared with all boats rising in a more interoperable ecosystem..

At MovieLabs, we will be communicating more in our 2030 Vision blog and bringing forward promising approaches in our 2030 Showcase. Follow the progress at movielabs.com.

CONTRIBUTORS

This paper would not have been possible without the invaluable support and contributions from the following executives:

Shadi Almassizadeh, *Disney*

Bill Baggelaar, *Sony Pictures*

Richard Berger, *MovieLabs*

Annie Chang, *Universal*

Matt Daw, *MovieLabs*

Daniel De La Rosa, *Sony Pictures*

Eddie Drake, *Marvel Studios*

Raymond Drewry, *MovieLabs*

Anthony Guarino, *Paramount*

Jim Helman, *MovieLabs*

Renard Jenkins, *Warner Bros. Discovery*

Josh Limor, *Paramount*

Daniel Lucas, *MovieLabs*

Giridhar Manepalli, *MovieLabs*

Lauralea Otis, *Marvel Studios*

Horst Sarubin, *Universal*

Stephen Shapiro, *Sony Pictures*

Doug Sherman, *DreamWorks Animation*

Leon Silverman, *MovieLabs*

Dalia Spingarn, *Paramount*

Jeremy Stapleton, *Universal*

Spencer Stevens, *MovieLabs*

Daryll Strauss, *MovieLabs*

Mark Turner, *MovieLabs*

Matt Walters, *Disney*

Chris Vienneau, *MovieLabs*

Jamie Voris, *Disney*

MovieLabs is a jointly run industry lab that enables member studios to work together to understand new technologies and drive the right ones to adoption. We help set the bar for future technology and then define specifications, standards, and workflows that deliver the industry's vision. Our goal is always to empower storytellers with new technologies that help deliver the best of future media.

Email: info@movielabs.com

Twitter: @ MovieLabsNews

Website: movielabs.com