



Orchestrating Production Processes with Cloud Native Pipelines

Anvita Jain

Senior Software Engineer, Netflix, USA
anvijain@gmail.com

Dan Golembeski

Product Manager, Netflix, USA
dgolembeski@netflix.com

Alex Schworer

Product Manager, Netflix, USA
aschworer@netflix.com

Gina Granger

Senior Software Engineer, Netflix, USA
gina.sc.granger@gmail.com

ABSTRACT

This paper gives the reader a behind the scenes look at how cloud native automation pipelines play a significant role in the production of Live Action, Visual Effects, and Animated content at Netflix Studios. Netflix's Studio Orchestrator is used to connect creators around the globe by efficiently sharing, tracking, and transforming production data on a scalable architecture that can be customized to meet the needs of production. We will uncover how our approach provides a way for geographically distributed creative teams to share and collaborate on assets as well as save our developer's time by leveraging shared infrastructure and shared components to build automation pipelines. We will use specific examples of automation pipelines that are in production today across Live Action, Visual Effects, and Animation to highlight the benefits of our approach to solving common industry workflows.

CCS CONCEPTS

• **Networks** → Cloud computing; • **Computing methodologies** → Computer graphics; Graphics systems and interfaces; • **Computer systems organization** → Pipeline computing.

KEYWORDS

Pipelines, Cloud Computing, Media Pipelines, Live Action, VFX, Animation

ACM Reference Format:

Anvita Jain, Alex Schworer, Dan Golembeski, and Gina Granger. 2022. Orchestrating Production Processes with Cloud Native Pipelines. In *The Digital Production Symposium (DigiPro '22)*, August 07, 2022, Vancouver, BC, Canada. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3543664.3543682>

1 INTRODUCTION

Netflix Studio operates at an unprecedented global scale which requires unique solutions to help our creative teams operate effectively and collaborate globally. Across the industry it is common for production teams to rely on manual or semi-automated processing pipelines as well as on premise infrastructure to manage and track critical production data, and there have been efforts at various

studios in VFX, Animation and Post Production to operate fully cloud-native workflows [Golembeski et al., 2017] [Vanns and Carey, 2016]. However, to achieve operational excellence across Netflix's large and diverse slate with teams distributed around the globe, we needed to develop tools that would help save our production teams time and reliably orchestrate the flow of data throughout the production process.

We developed Studio Orchestrator to solve this problem. Studio Orchestrator enables developers working in the Netflix ecosystem to rapidly build automation pipelines in service of production's needs by:

- Focusing developers on the business logic and not on the complexities of the infrastructure, systems and external services being used
- Sharing common processing tasks across multiple automation pipelines
- Configuring pipelines for reuse across multiple contexts (e.g., different productions, sequences, etc.)
- Providing scalable and secure-by-default authorization primitives for developers building workflows that work on sensitive production data

2 TECHNICAL DESIGN

Studio Orchestrator enables the execution and querying of media-related processes, supporting the creation of content destined for the Netflix service. Example processes include "allow a VFX vendor to submit media for review" and "allow a finishing vendor to deliver unfinished shots to a VFX vendor". There are dozens of similar workflows that occur on productions, and as creative teams push the envelope and tell new stories, new needs will continue to arise. Like conclusions made by other studios, building a single, static, monolithic system to serve these needs isn't feasible [Golembeski et al., 2017] [Kelkar, 2019]. Our approach leverages several existing infrastructure components within the Netflix ecosystem to perform the heavy lifting, like managing asset metadata and relationships, moving data around globally-distributed storage systems, and generating video proxy media. Figure 1 represents the overall placements of different systems that enable a successful pipeline.

Studio Orchestrator introduces the concept of a Pipeline to help developers build scalable, reusable production workflows. Figure 2 shows the three main components that make a Pipeline.

- **State Machine:** The state machine enables developers to describe the business process of a workflow, and enforces authorization rules. An example business process may model a



This work is licensed under a Creative Commons Attribution-NoDerivs International 4.0 License.

DigiPro '22, August 07, 2022, Vancouver, BC, Canada

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9418-5/22/08.

<https://doi.org/10.1145/3543664.3543682>

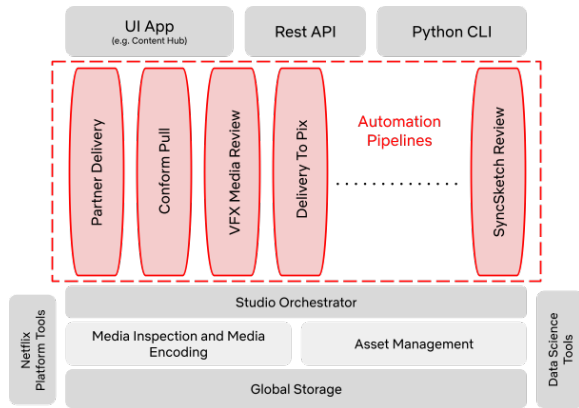


Figure 1: System Design

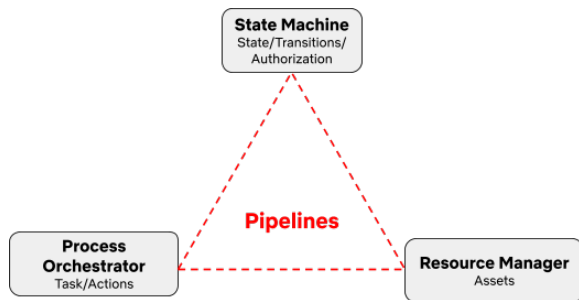


Figure 2: Pipeline Components

creative review/approval workflow with the creative leadership on a production, or describe the final frame QC process. The State Machine service also allows for fine grained authorization across different states of a pipeline, enabling or forbidding certain domains from accessing the data when the pipeline is at a certain state.

- **Resource Manager:** The resource manager provides a single interface to connect the pipeline to any data entities the pipeline operates on. Example supported entity relationships are to the Studio’s Asset Management service, to globally distributed file-systems (object storage & hierarchical), or to compute resources (render nodes and workstations). A Resource defines a state transition graph (to enable async/await logic based on a resource’s state, as described below), pre-/post-action data enrichments, and powerful query APIs.
- **Process Orchestrator:** The process orchestrator provides a powerful Async-Await event-based pattern to register a dependency on a Resource that is not yet available, automatically continuing the workflow once available. This eliminates costly polling by applications and enables cascading execution of sub-pipelines (assuming the Pipeline’s State Machine allows it) based on upstream Resource state changes.

In addition to the three core components defined above, we rely heavily on Conductor (<https://conductor.netflix.com>), an open-source workflow execution engine incubated at Netflix.

Reusability of pipelines and of individual tasks within workflows [Johnson et al., 2014] are a key design goal of Studio Orchestrator. We provide a Standard Task Library (called STLs) that has reusable connections to common infrastructure services, like the Asset Management Platform, various storage systems, encoding, rendering, and notification services. In addition to tasks for internal Netflix-operated services, there are common shared STLs for interacting with external media review platforms like ShotGrid, PIX, and SyncSketch.

To enable support and observability, Studio Orchestrator provides built-in alerts, monitoring, and analysis systems to track progress of pipelines, notify support teams, analyze pipeline usage (for example, to capture the pipeline success/failure rates, to inform future development).

3 ASSET ACCESS AND AUTHORIZATION FRAMEWORK

A production is a highly collaborative process, with hundreds of people working together to realize a single creative vision. Providing reliable access to the media that the teams need to do their work is a perennial problem.

A key design constraint we faced was in finding the right balance between the openness of access to assets and the operational complexity in managing permissions. To address this, Studio Orchestrator is designed to be Domain-aware. Domains are pre-defined groupings that roughly describe a phase or department within a production.

As shown in Figure 3, Production A includes a Domain for Editorial. If a user is granted access to the Editorial Domain, then they will be granted read/write access to the assets and files stored within, but not to any assets in the production that are in another domain (like Picture Finishing or VFX). To simplify onboarding, we also pre-define four levels of role-based access that a user may be granted in a Domain: Admin, Manager, Collaborator, or Uploader (write-only) as shown in Figure 4.

A user’s permissions are additive, so that they can be granted access to multiple domains within a production, with varying degrees of access (from write-only access to full permissions to read, write, delete and update data). We’ve found that this approach reduces the amount of manual curation of user groups by production teams and strikes the right balance for all but the most sensitive of productions.

Scalable permissions are baked into the design of Studio Orchestrator, and the State Machine described above relies on a user’s domain membership during pipeline execution to drive authorization (e.g., an Editor in the Production Editorial domain can Submit the Cut Review pipeline, but an Artist in the Production VFX could not).

4 PRODUCTION USE CASES ACROSS LIVE ACTION, VFX AND ANIMATION

In the last 3 years our team has developed a library of over 250 tasks that can be plugged into any pipeline. Over 20 different pipelines are used during pre-production, production, and post production phases of Netflix productions, and have been used to store and transform petabytes of data.

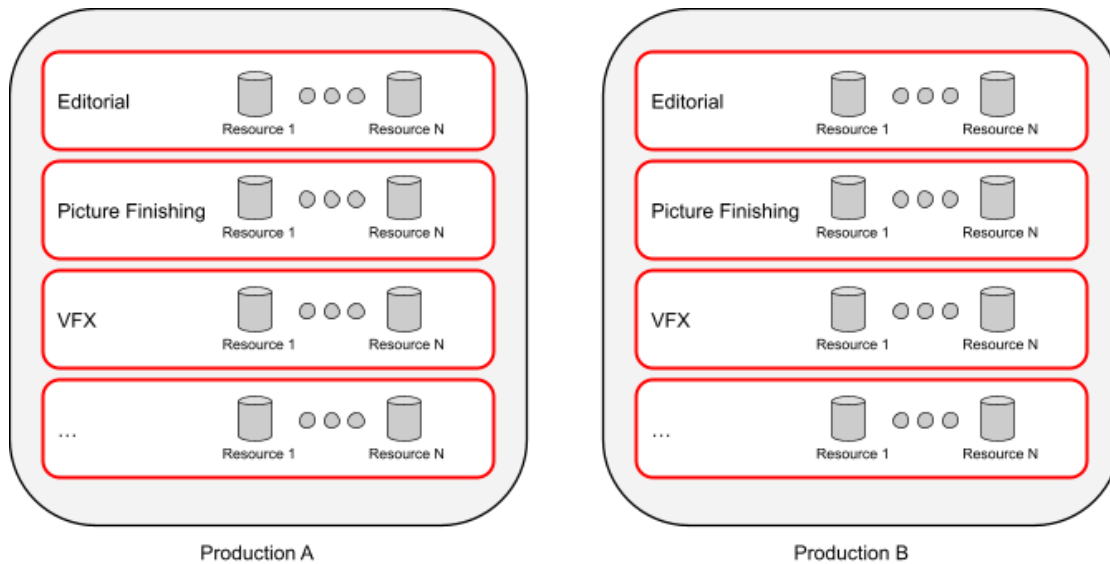


Figure 3: Pre-defined Domains within a production segment asset and file access

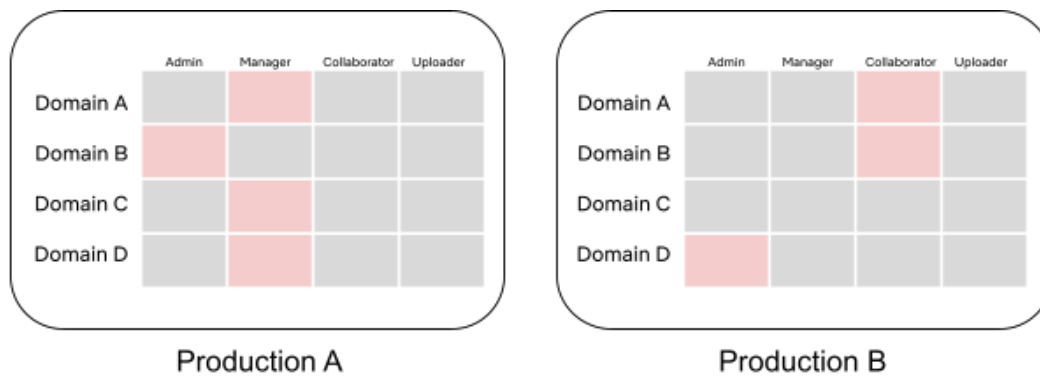


Figure 4: A single user can be granted access to multiple domains within individuals' productions, and access is additive

Studio Orchestrator sees more than 4,000 successful pipeline executions a month, which result in the creation of critical production entities such as playlists and proxy media for review.

Dozens of asset types are processed via Studio Orchestrator pipelines, including video, audio, still images, Original Camera Files (OCF), ACES image sequences, Alembic caches, Maya scenes, and VFX plates.

Below we outline four use cases in production today at Netflix Studio.

4.1 Partner Delivery Pipelines

4.1.1 Why. Figuring out a consistent and flexible way to share data between Netflix Animation Studio (NAS) and their Creative Partners is a challenge at scale. Typically, production teams spend hours engrossed in back-and-forth communication and manual file sharing workflows.

At Netflix Animation Studio, there was a need for a unified delivery system that would not only handle the transfer of files, but would also generate proper insights, create assets, track deliveries in ShotGrid, and send notifications during different states of the delivery to the appropriate recipients. We also wanted to automate downloads and uploads for both internal and external users to save them from waiting on manual workflows.

4.1.2 How. We built the Partner Delivery Pipeline to help streamline the process of incoming and outgoing deliveries. The pipelines are the source of truth of what has been sent and received, typically performing the following actions.

- Users can submit files either via a web UI called Content Hub or use a python-based command line tool to upload data into Content Hub Workspace Folder.
- Validate incoming files against a standardized delivery manifest to ensure all delivered files have the metadata required to be published to Netflix Asset Management.

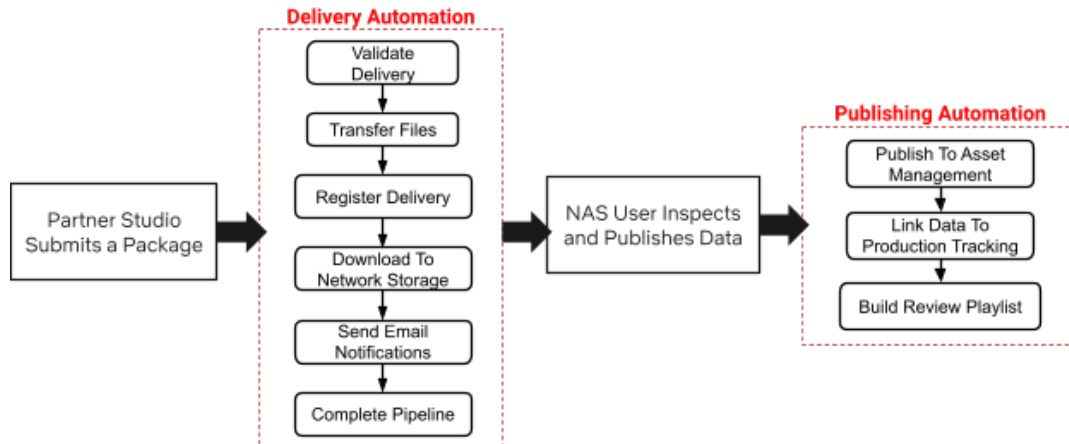


Figure 5: Incoming Partner Delivery Pipeline

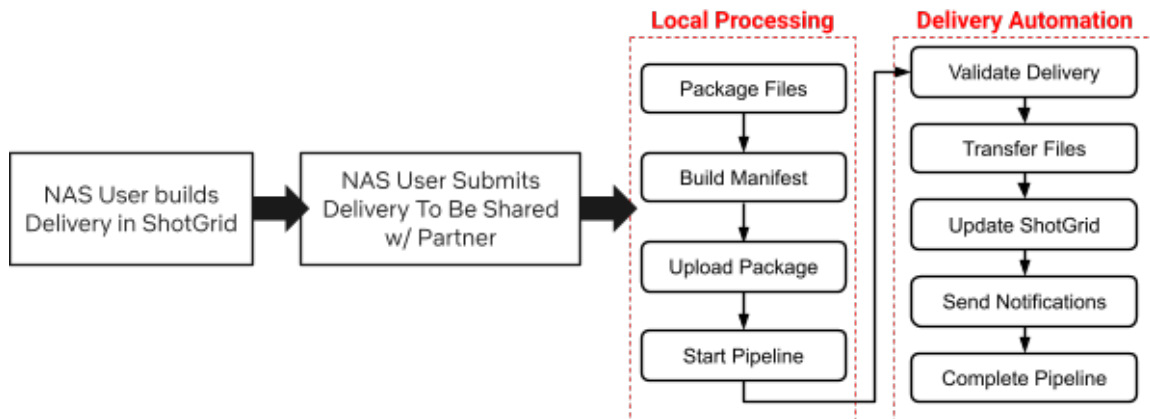


Figure 6: Outgoing Partner Delivery Pipeline

- Create assets per Netflix’s internally defined schemas in Netflix’s Asset Management System containing metadata as well as storage details.
- Connect with ShotGrid by automating the creation and updating of tracking entities within our various ShotGrid sites using STLs (reusable tasks).
- Transfer files to/from a large-scale distributed file system on the cloud.
- Send actionable email notifications that are dynamically generated by the pipeline based on the status of the delivery and the recipient of the notification. The content of the notification is carefully curated based on each user’s permissions on a production.

This pipeline can be kicked off from a web application (called Content Hub) or via a command-line tool called PyChapi, giving vendors the flexibility to choose their mode of submission when working with the animation studio. Figure 5 and Figure 6 outline the process flow described above.

4.1.3 Benefits.

- Reduced manual steps by automating manifest generation, uploads, and downloads.
- Enhanced tracking of production deliveries by integrating with ShotGrid.
- Improved notifications by dynamically generating actionable emails that direct users more quickly to the data.
- Standardized metadata ingest helps production teams spend less time updating information manually.
- By validating deliveries, the pipeline catches errors sooner in the process.

4.2 SyncSketch Review Pipelines

4.2.1 Why. In most studios, artists are expected to know how to navigate multiple third-party applications and are constantly jumping between them to gather all the context they need to work. We have a proprietary application designed around making artists more efficient. We built integration pipelines to ensure relevant information is transferred to the proprietary application so artists would not have to navigate away to third party applications such as SyncSketch for review notes.

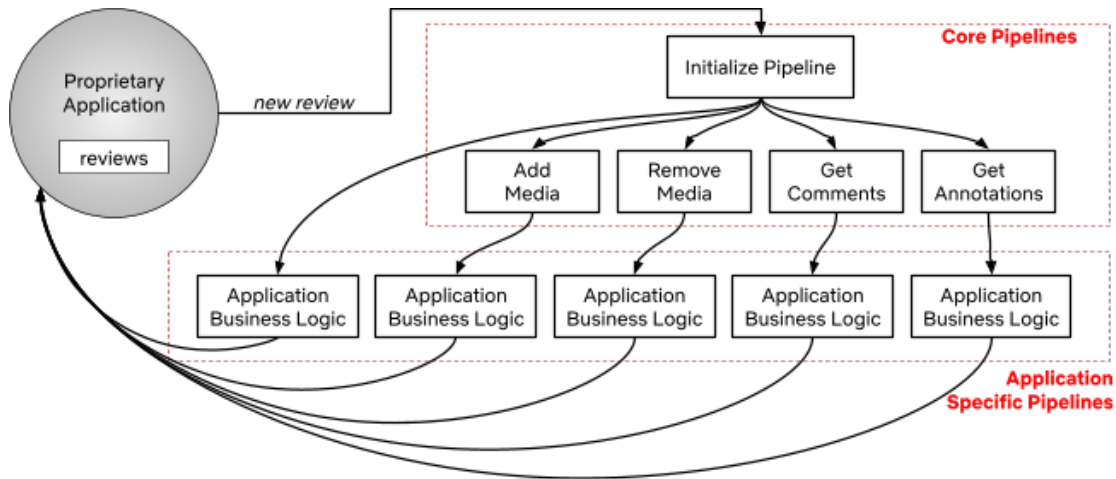


Figure 7: SyncSketch Review Pipeline

4.2.2 How. We built a SyncSketch Review pipeline to be our single source of truth for reviews. When an artist creates a review within our application, we automatically kick off a SyncSketch Review pipeline. Our application holds onto the pipeline id, and from that, can get all the relevant information about the review and its media. Each piece of media that is added to a review has a proxy automatically sent to the corresponding SyncSketch review, and the media details are stored as a top-level resource on the pipeline. Once the review is complete in SyncSketch, we have a pipeline action to fetch data such as comments and annotations from all the media within a review. This is triggered via our application by the production coordinator once the notes have been cleaned up within the SyncSketch review. This action has authorization rules, only permitting certain users to perform the action. The comments and annotations are stored as child resources to the corresponding media in the pipeline. The parent/child relationship of resources makes it easy to dive into the details of a given piece of media.

The pipeline is composed of multiple sub-pipelines, each that have a specific purpose. For example, the top-level pipeline is triggered once a review is created; the add media pipeline is triggered when an artist adds media to a review; the remove media pipeline is triggered when an artist removes media from a review; and the get comments and annotations pipeline is triggered when the review is complete, and the notes are ready to be brought back to our application. The core SyncSketch Review pipeline functionality is to create a review, manage media in SyncSketch, and retrieve comments and annotations. On top of this, we have added the ability to allow different applications to hook in their business logic after each key action. Below is the diagram of how the pipeline is laid out. The proprietary application only ever talks to the top-level pipeline. Depending on context (e.g., add/remove media), the top pipeline will initiate the corresponding sub pipelines. Figure 7 shows the overall architecture of this pipeline.

4.2.3 Benefits.

- **Reusability:** The SyncSketch Review pipeline is generic enough to be reused for any application wishing to leverage

SyncSketch as a review system. Due to the nature of integrated hooks, each application has the flexibility to build its business logic on top of it. This means shows spend time building their business logic, rather than worrying about rebuilding the core functionality.

- **Context At Single Location:** An artist can go to a single application and find all the review notes for the asset being worked on. Because the proprietary application knows what reviews an artist has taken part in and the studio orchestrator has authorization rules around who can access what data, we are able to find the associated review pipelines and group and display notes for all reviews related to the asset. This provides the artist more context in a single location.
- **Visibility:** Because each pipeline has its own state machine, we can surface progress to the user in our application. And because all the pipeline information persists, we can gather metrics for all SyncSketch Review pipelines, allowing us to better predict to the user when data will land and allowing developers to find bottlenecks and optimize the pipeline workflows.

4.3 VFX Media Review Pipelines

4.3.1 Why. Our shows need to deliver VFX assets regularly to Netflix for review. This is not unique to Netflix; all shows must come up with a way to accomplish this. As Netflix grew its production slate, our studio operations teams were challenged to keep up with the operational complexity of managing review submissions from VFX studios. Additionally, each VFX production team had slightly varying workflows for reviewing material, including a wide range of applications such as Aspera, Google Drive, PIX, ShotGrid, Excel, and Google spreadsheets. Netflix needed a way to streamline the experience for both internal and external users by automating VFX media ingestion, encoding and publishing.

4.3.2 How. The VFX Media Review Pipeline allows vendors to submit VFX shots and related assets to Netflix Studio for review via the Content Hub web application, thereby enabling a centralized

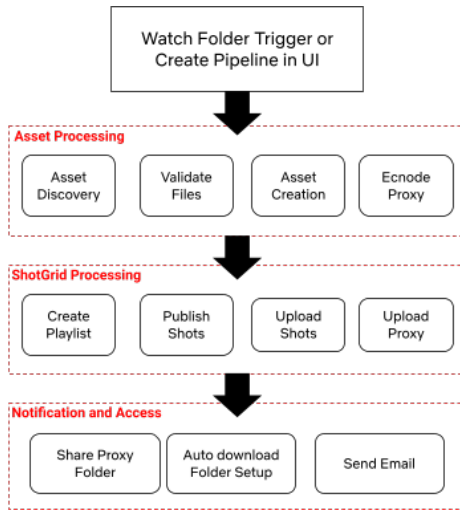


Figure 8: VFX Media Review Pipeline

review process. Production teams can submit up to 300 shots in a single submission that can include potential final visual effects, key work-in-progress versions, and asset references. Below are the steps involved in the pipeline. VFX Media Review Pipeline uses concepts of subpipelines to perform all the processes listed above. Figure 8 illustrates how different subpipelines make up the entire pipeline.

- **Trigger the Pipeline:** There are two ways to kick off a pipeline, manually via the Content Hub UI, or by dragging-and-dropping material onto a pre-configured watch folder in the Content Hub Workspaces application.
- **Asset Discovery:** Extracts metadata from file-and-folder structure and creates asset relationships in the Asset Management service.
- **Validation:** Ensures all media is in the correct spec (bitrate, resolution, color space), and all required metadata is provided.
- **Encoding:** Generates reviewable proxy media for multiple environments.
- **ShotGrid Processing:** Creates a playlist with relevant details, submission notes, and vendor information.
- **ShotGrid Processing:** Creates shot and version entities in ShotGrid as needed, populated with metadata from the Asset Management service.
- **ShotGrid Processing:** Uploads generated proxy media to ShotGrid.
- **Notifications:** Sends an email notification to the users on a production, based on their Domain membership. The email includes details on how to view proxy media and contains ShotGrid links for review.
- **Auto-Download:** Sends auto-download notifications to configured client applications for localization review media to nearby storage systems or local storage.

4.3.3 Benefits. As of May 2022, we have received more than 270,000 VFX Shot versions across hundreds of productions utilizing

the VFX Media Review integration between Content Hub (UI App) and ShotGrid by leveraging its toolkit [Tomlinson et al., 2017]. As a result, we’ve seen the following benefits of using this pipeline at Netflix.

- Centralize encoding for proxy generation, so that VFX studios don’t need to generate proxy media themselves.
- Flag metadata and file spec issues early in the process, so that vendors can find and fix issues early.
- Remove silos by storing all media in central Asset Management service.
- Fast onboarding for new vendors by leveraging watch-folders for pipeline integration.

4.4 Delivery to PIX Pipelines

4.4.1 Why. PIX is a media review system commonly used for media review (especially Dailies) during Live Action production. It is common for production teams to manually upload media to PIX during production, and enter metadata into PIX for teams to review, to help drive creative conversations and enable media to be easily found within PIX’s asset management library.

4.4.2 How. Using a pipeline, production teams can now automatically fetch footage, formulate metadata and publish these metadata and assets to PIX. Once the pipeline has been initialized, STL tasks are used to query for the PIX project details, authenticate, and deliver the data to the corresponding PIX project.

Next, a new folder for the day’s footage is created in PIX, and this folder is attached as a resource to the pipeline for reference. The pipeline then fetches the pre-computed metadata and storage locations for all the files that are part of the delivery. These files are then uploaded to PIX along with the relevant metadata for each file. In return, PIX generate URLs to access the files. Using Studio Orchestrator, we attach the URL information to the pipeline so it can be queried later, displayed within in-house apps such as Content Hub, and used to send emails containing the links to the end user depending on their domain membership on the project. Figure 9 illustrates the complete pipeline process.

4.4.3 Benefits. Once all the files are delivered to PIX with metadata, users can access the PIX folder directly by using the link in the email they received or by viewing the pipeline on apps such as Content Hub. Users can perform grouping/searching/sorting in PIX based on the uploaded metadata. This allows creative teams to have more meaningful conversations based on media, get to assets quickly, and be confident that all the media is reliable and pre-validated. With this pipeline, Netflix can automatically and consistently push metadata for any show produced anywhere in the world for easy review in PIX.

5 CONCLUSION

Studio Orchestrator has helped our productions and Netflix Studio operations teams spend less time on media logistics, and more time on creative work. Developers building workflows have access to reusable components and powerful abstractions over the business process, resources, and execution of a workflow. Because production data is sensitive, and hundreds of people work on productions, the

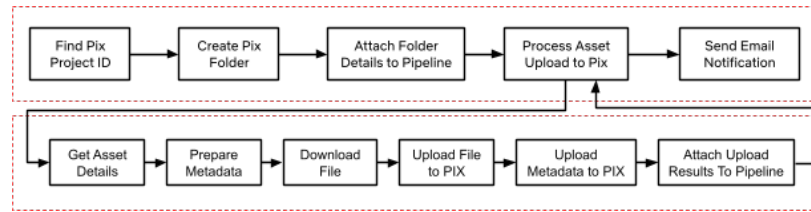


Figure 9: Pix Metadata Pipeline

Domains model provides a good balance between strict access and increasing operational overhead.

While we've been able to realize many benefits from deploying pipelines on Netflix productions, we've also learned along the way and have identified areas to further improve the Studio Orchestrator platform. Below are some of our key learnings.

- Sending enriched and actionable notifications to users keeps them informed throughout the various stages of processing.
- Providing users multiple ways to trigger automated pipelines to share data securely leaves little room for user error.
- A framework can be created to quickly build workflow automations for productions, helping save production teams time spent on manual logistics work.
- We leveraged open-source technologies like Conductor to power workflow execution. Conductor allows us to reuse standard tasks across other teams at Netflix and provides a good debug tool.
- But only powering workflow execution is not enough, having abstractions for Resources, State Machines, and Process Orchestration helps developers be more productive when building production workflows.

Some areas of improvements are:

- We would like to introduce a validation framework that production developers can use to customize validation rules per production. This will detect errors sooner and empower partners to resolve issues ahead of delivering their data to Netflix.
- We would like to give users better real time visibility into the progress of long running, expensive tasks in a pipeline and do not need to wait for notifications to know that the system is working normally.
- Production developers working with Studio Orchestrator have limited tools for authoring workflows, and documentation and onboarding for new developers is limited.

ACKNOWLEDGMENTS

Many teams within Netflix have contributed to the work presented in this paper, and there are too many contributors to list by name. This work would not have been possible without the partnership of engineering, product, design, and Studio operations teams within Netflix Studio, and the authors are grateful for the contributions of their colleagues in this effort.

The work described here grew out of Netflix Studio's initial archival and delivery engineering efforts that began in 2015 and is detailed in the Production Media Management:

Transforming Media Workflows by leveraging the Cloud blog post (<https://netflixtechblog.com/production-media-management-transforming-media-workflows-by-leveraging-the-cloud-1174699e4a08>) on the Netflix Tech Blog. Three backbones of Studio orchestrator are video encoding platform (<https://netflixtechblog.com/tagged/video-encoding>), Asset management platform (<https://netflixtechblog.medium.com/elasticsearch-indexing-strategy-in-asset-management-platform-amp-99332231e541>) and conductor (<https://conductor.netflix.com>). The linked blog posts give deeper insights into how their platforms were built.

REFERENCES

- Dan Golembeski, Ray Forziati, Ben George, and Doug Sherman. 2017. PipelineX: a feature animation pipeline on microservices. In Proceedings of the ACM SIGGRAPH Digital Production Symposium (DigiPro '17). Association for Computing Machinery, New York, NY, USA, Article 1, 1–4. <https://doi.org/10.1145/3105692.3105702>
- Jim Vanns and Aaron Carey. 2016. A fully cloud-based global visual effects studio. In ACM SIGGRAPH 2016 Talks (SIGGRAPH '16). Association for Computing Machinery, New York, NY, USA, Article 72, 1–2. <https://doi.org/10.1145/2897839.2927432>
- Chris Johnson, Josef Tobiska, Josh Tomlinson, Nico Van den Bosch, and Wil Whaley. 2014. A framework for global visual effects production pipelines. In ACM SIGGRAPH 2014 Talks (SIGGRAPH '14). Association for Computing Machinery, New York, NY, USA, Article 57, 1. <https://doi.org/10.1145/2614106.2614159>
- Josh Tomlinson, Manne Öhrström, and Don Parker. 2017. The shotgun pipeline toolkit: productizing and democratizing production pipelines. In Proceedings of the ACM SIGGRAPH Digital Production Symposium (DigiPro '17). Association for Computing Machinery, New York, NY, USA, Article 3, 1–7. <https://doi.org/10.1145/3105692.3105699>
- Natasha Kelkar. 2019. Building modern VFX infrastructure. In ACM SIGGRAPH 2019 Talks (SIGGRAPH '19). Association for Computing Machinery, New York, NY, USA, Article 49, 1–2. <https://doi.org/10.1145/3306307.3328194>