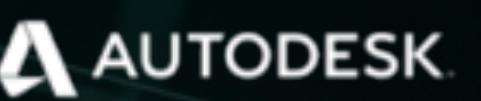


SHOTGUN ECOSYSTEM

# Advanced Shotgun Development



# Advanced Shotgun Development

---



Learn how to develop,  
package up, and distribute  
Toolkit to a remote user base.



## Manne Öhrström

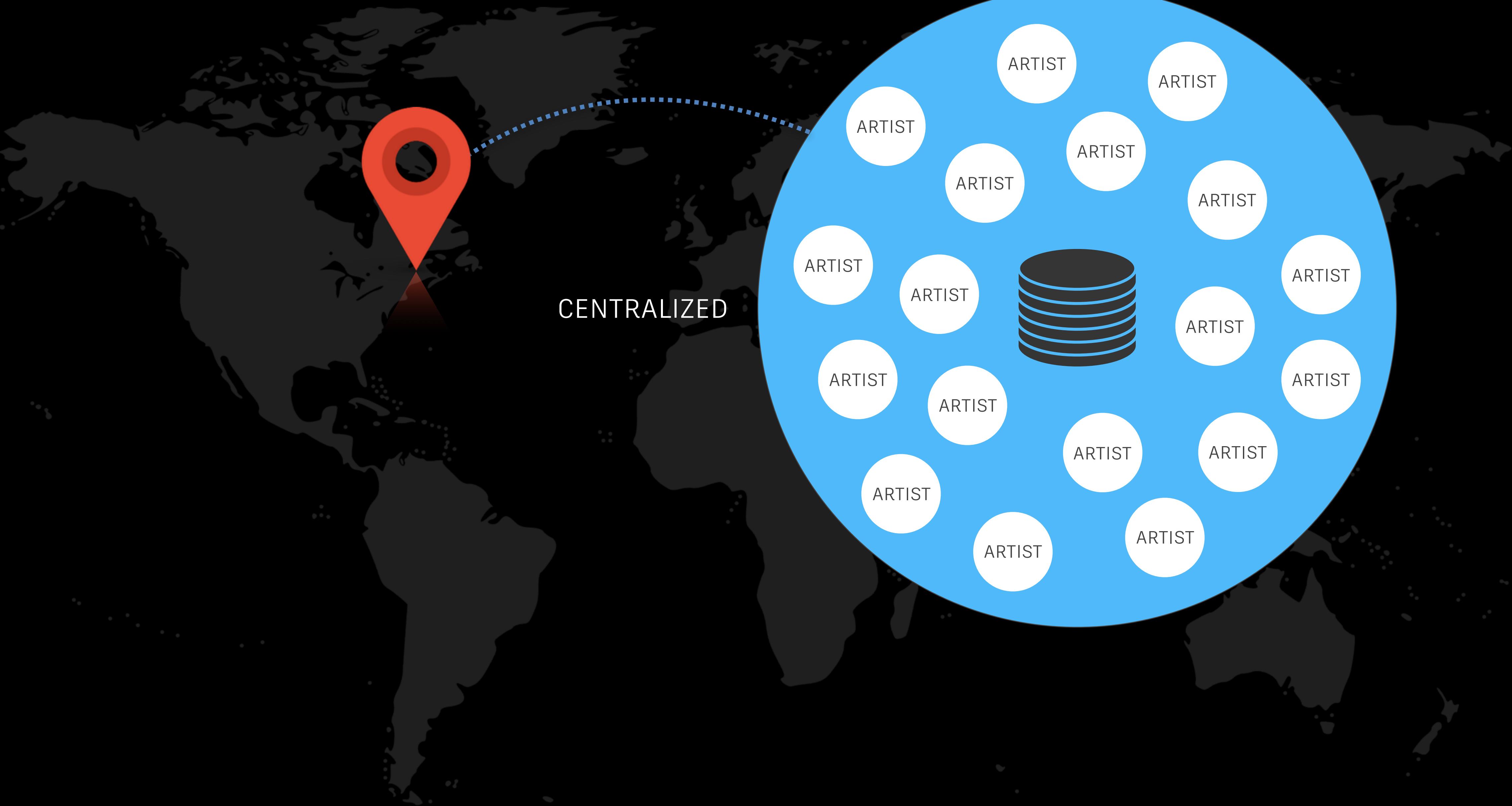
**Senior Principal Engineer, Autodesk**

*Manne has been working with pipeline design and workflow engineering in games and film production for the past 14 years at companies such as Electronic Arts, Sony Computer Entertainment, and Framestore Feature Animation. He joined Shotgun in 2008, Autodesk in 2014 and is the technical lead for Shotgun's integration platform.*

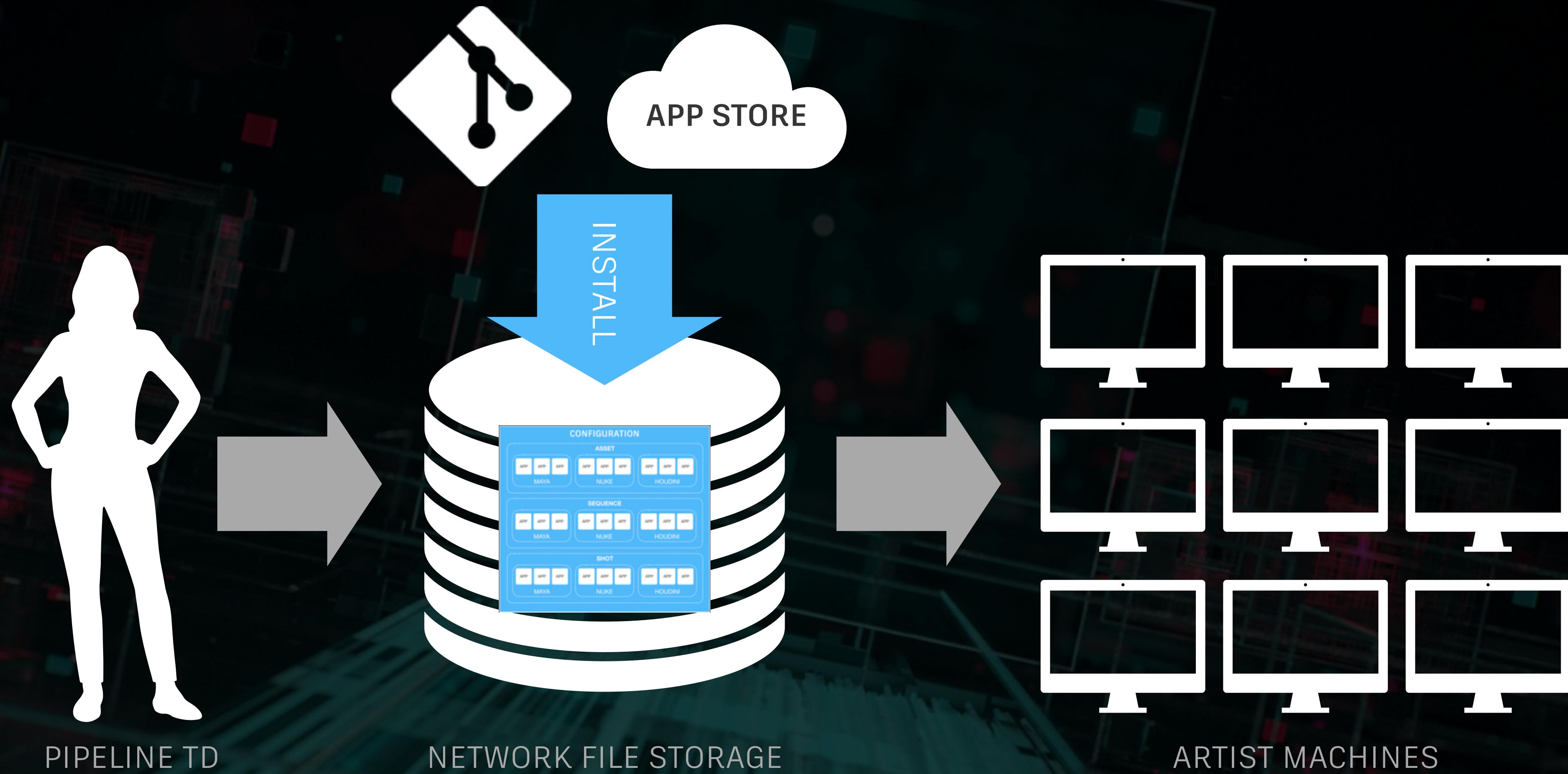


# Motivation

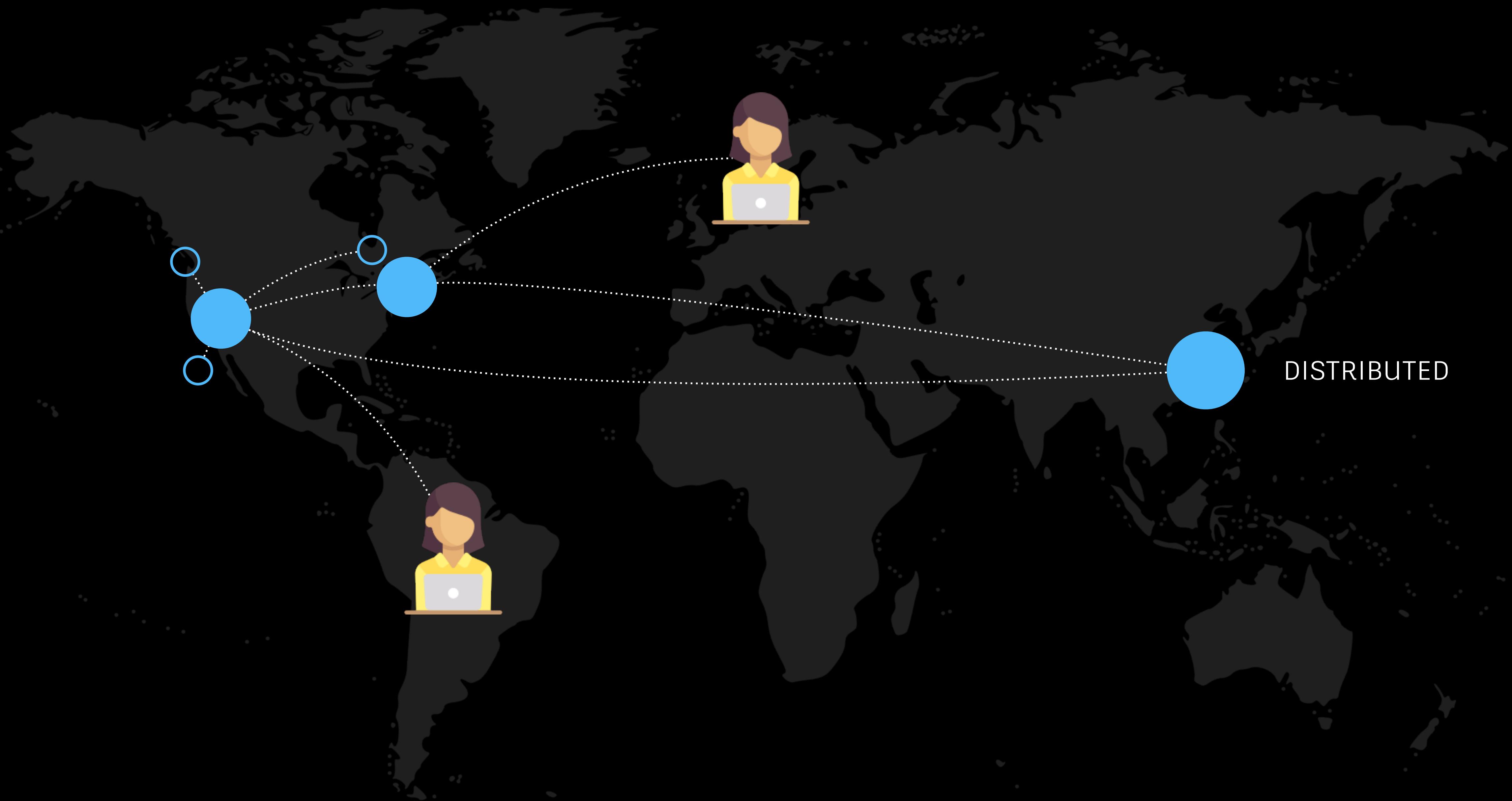
Why are distributed configs important?



# Classic toolkit



# A distributed workforce



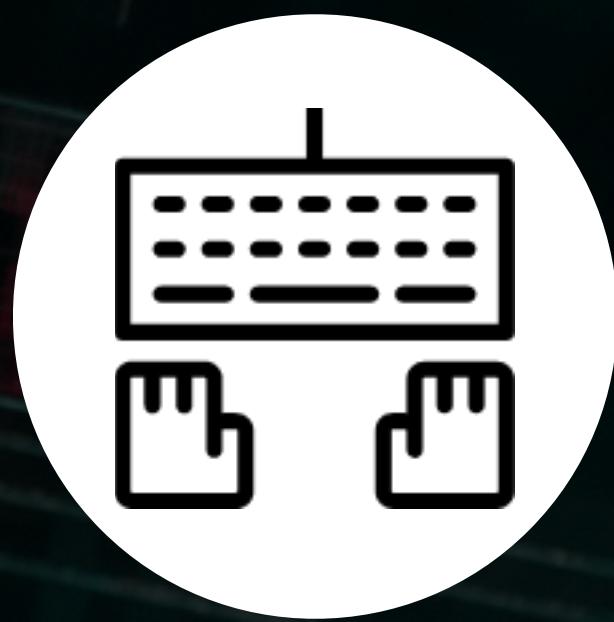
# What you will learn in this session

---

## DISTRIBUTED TOOLKIT CONFIGS



Setup



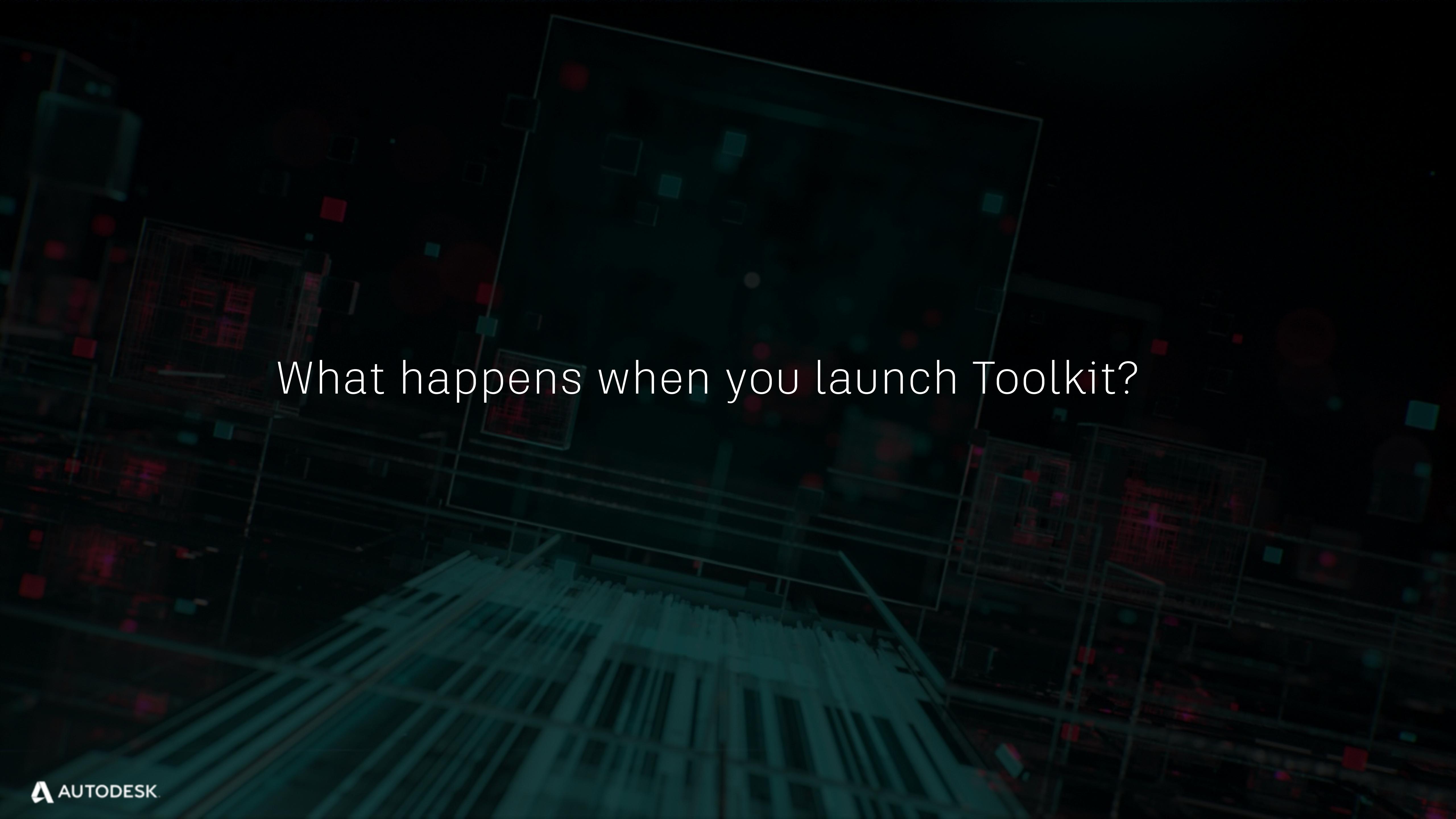
Update



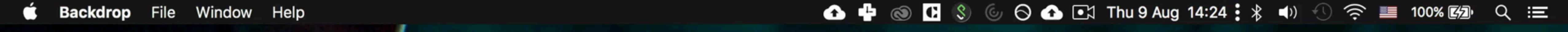
Source  
Control



Next  
Steps



What happens when you launch Toolkit?



TOOLKIT  
APP STORE

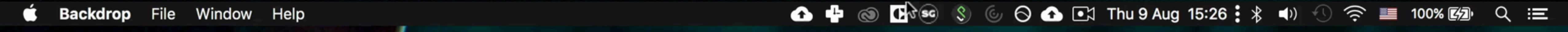






NEW  
FEATURES!

# Distributed Project Setup



# What just happened?





Shotgun

Secure | https://sg-devday-2018.shotgunstudio.com/page/project\_overview?project\_id=116

manne

SHOTGUN Inbox My Tasks Media Projects All Pages People Apps

Search Shotgun... + manne

HYPERSPACE MADNESS Overview Media Sequences Assets Shots Schedule Mocap Takes Versions PDFs Other Project Pages Project Actions

Hyperspace Madness

All Note Versions Publishes

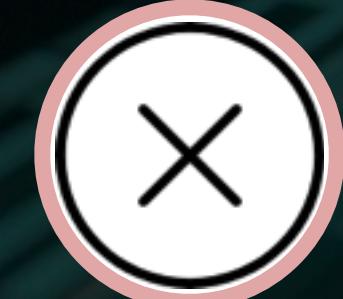
Send a note to everyone on this Project...

api\_101 1.0 changed the status of Version checkerboard.mov  
Status:

Media

## Summary

---

-  Push tools to remote artists
-  Easy to update - just restart desktop
-  Distributed, but can still use shared storages
-  Caches apps on local machines
-  Does not transfer your files



# Making Changes

Creating a developer sandbox

Checking for updates

Deploying your changes

Apps

< Hyperspace Madness ▾

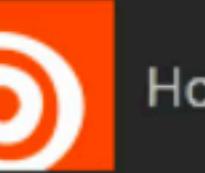
▼ RECENT

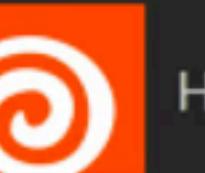
 Maya  
2017

▼ STUDIO

 Publish...

▼ CREATIVE TOOLS

 Flame  Houdini

 Houdini FX  Mari

 Maya  Nuke

 NukeAssist  NukeX

▼ EDITORIAL TOOLS

**SHOTGUN**

Pipeline Configurations : Hyper x

Secure | https://sg-devday-2018.shotgunstudio.com/page/2291

User Restrictions updated on Manne's Config ... Undo

SHOTGUN

Inbox My Tasks Media Project Pipeline Configurations

Search Shotgun...

HYPERSPACE MADNESS Overview Media Sequences Assets Shots Schedule Mocap Takes Versions PDFs Other Project Pages Project Actions

## Pipeline Configurations

+ Pipeline Configuration Sort Group Fields More

Config Name	User Restrictions	Plugin Ids	Uploaded Config	Descriptor
<a href="#">Manne's Config Sandbox</a>	<a href="#">Manne Öhrström</a>	basic.*	<a href="#">v1.1.11.zip</a>	sgtk:descriptor:dev?path=/Users/manne/Documents/dev/hsm_sandbox
<a href="#">Primary</a>		basic.*		

User Restrictions

Manne Öhrström

Descriptor

sgtk:descriptor:dev?path=/Users/manne/Documents/dev/hsm\_sandbox

1 - 2 of 2 Pipeline Configurations

25 per page

The screenshot shows a web browser window with the following details:

- Title Bar:** Descriptors — tk-core v0.18.15
- Address Bar:** Secure | https://developer.shotgunsoftware.com/tk-core/descriptor.html
- Toolbar:** Includes icons for star, refresh, search, and other browser functions.
- Page Content:**
  - Header:** SHOTGUN Pipeline Toolkit, tk-core
  - Search Bar:** Search docs
  - Sidebar (Left):** Overview, Initialization and startup, Core, Apps, Engines and Frameworks, Utilities, Descriptors (expanded), Descriptor types (selected), API reference, Authentication, Alphabetic Index.
  - Page Body (Main Content):**
    - ## Descriptor types
    - Several different descriptor types are supported by Toolkit:

      - An **app\_store** descriptor represents an item in the Toolkit App Store
      - A **shotgun** descriptor represents an item stored in Shotgun
      - A **git** descriptor represents a tag in a git repository
      - A **git\_branch** descriptor represents a commit in a git branch
      - A **path** descriptor represents a location on disk
      - A **dev** descriptor represents a developer sandbox
      - A **manual** descriptor gives raw access to the bundle caching structure
    - The descriptor API knows how to access and locally cache each of the types above. You can control the location where the API caches items and supply additional lookup locations if you want to pre-bake your own collection of caches. Descriptors that are downloaded (cached) to the local disk are called **downloadable** descriptors. The **app\_store**, **shotgun**, **git** and **git\_branch** descriptors are downloadable descriptors, while the **path**, **dev** and **manual** descriptors are accessed directly from the specified path.
    - ## The Shotgun App store
    - The Shotgun app store is used to release and distribute versions of Apps, Engines, Configs etc. that have been tested and approved by Shotgun. App store descriptors should include a name and version token and are on the following form:

Apps

< Hyperspace Madness ▾

Manne's Config Sandbox CONFIGURATION

▼ RECENT

Maya 2017

▼ STUDIO

Publish...

▼ CREATIVE TOOLS

Flame	Houdini
Houdini FX	Mari
Maya	Nuke
NukeAssist	NukeStudio

SHOTGUN

Apps

< Hyperspace Madness ▾

Manne's Config Sandbox CONFIGURATION

▼ RECENT

 Maya  
2017

▼ STUDIO

 Publish...

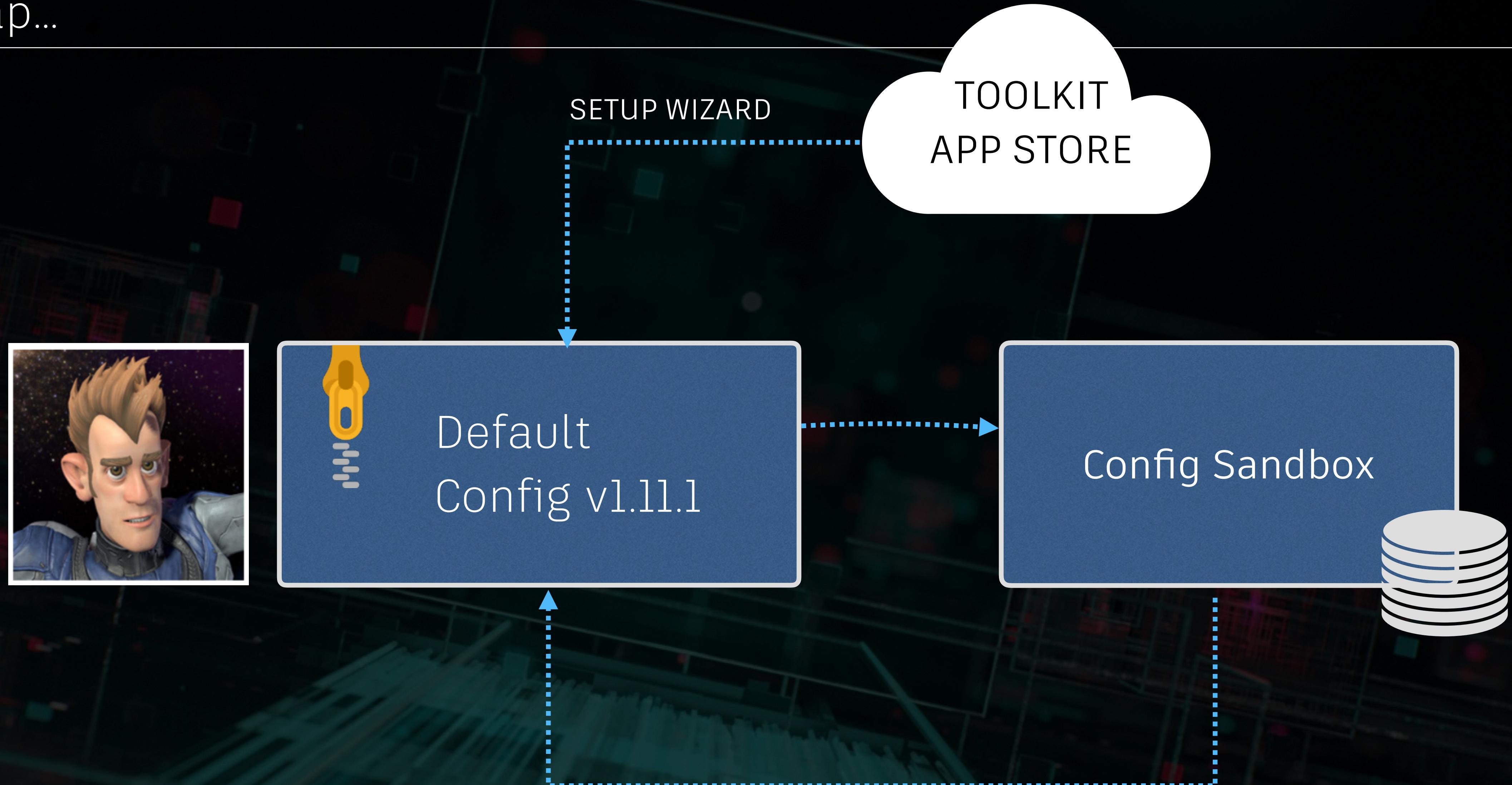
▼ CREATIVE TOOLS

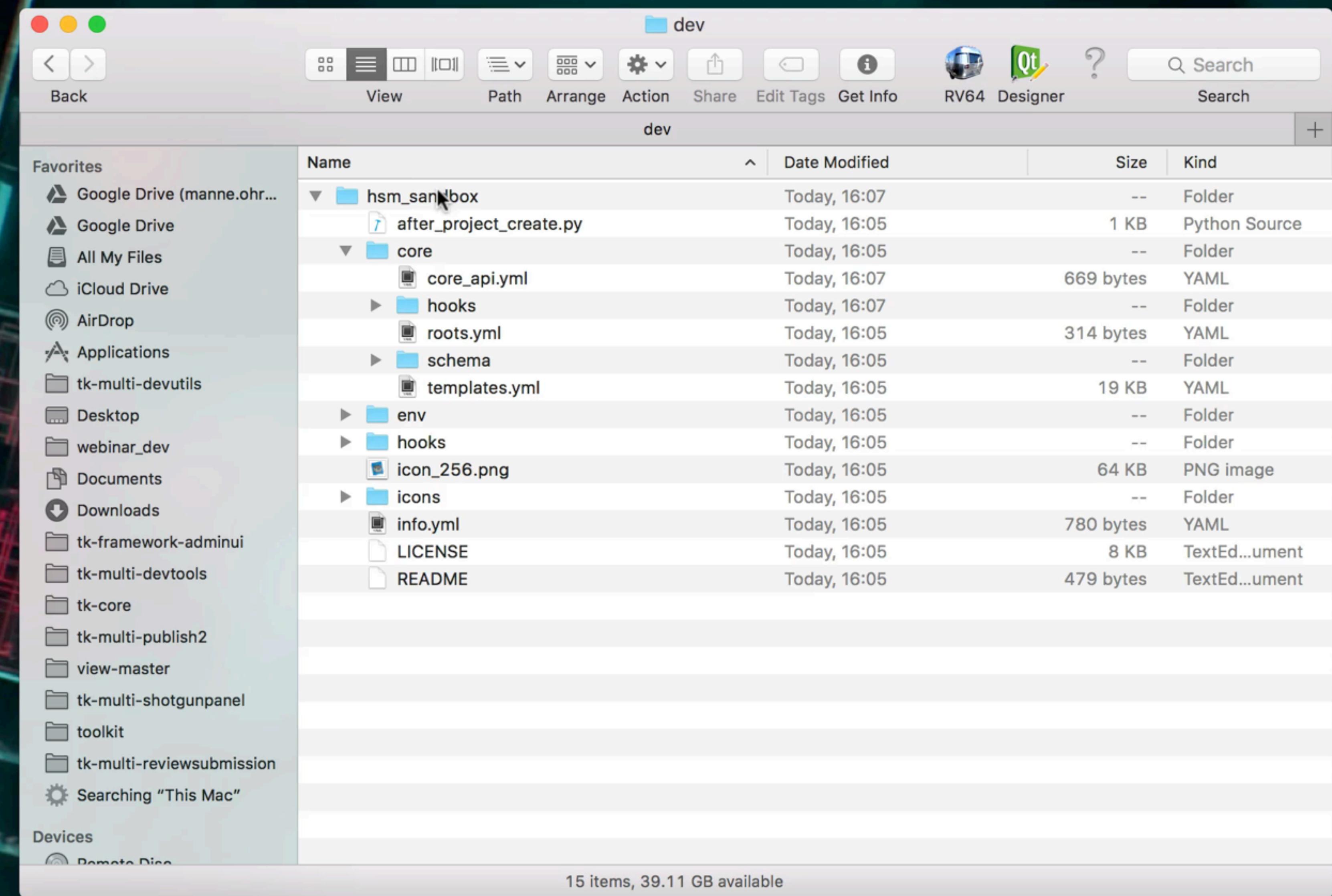
	Flame		Houdini
	Houdini FX		Mari
	Maya		Nuke
	NukeAssist		NukeStudio

SHOTGUN



## To Recap...







# A Version Controlled Pipeline

Learn how to set up a version controlled configuration for your studio.



Studio Config



v1.0.2



v1.1.0



Work  
Branch

Pull Request

v1.1.1



Sandbox

For SG-5387: Adds loader2 and breakdown to Hiero/Nuke Studio project environments. #34

**Merged** thebeeland merged 3 commits into `master` from `SG-5387_nuke_studio_clips` on May 19

Conversation 0 Commits 3 Checks 0 Files changed 4 +97 -7

thebeeland commented on May 15

This takes advantage of support for importing Hiero/NS clips from rendered images and video files.

thebeeland added some commits on May 15

- Adds loader2 and breakdown to Hiero/Nuke Studio project environments. b8d76a6
- Adds the `clip_import` actions to hiero/NS shotgunpanel configs. b9bbe1d
- Latest versions of loader2, shotgunpanel, and breakdown. 0f0ba7a

Revert

For SG-5387: Adds loader2 and breakdown to Hiero/Nuke Studio project environments. #34

GitHub, Inc. [US] | https://github.com/shotgunsoftware/tk-config-default2/pull/34

Search or jump to... Pull requests Issues Marketplace Explore Unwatch 38 Star 3 Fork 15

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

shotgunsoftware / tk-config-default2

manne

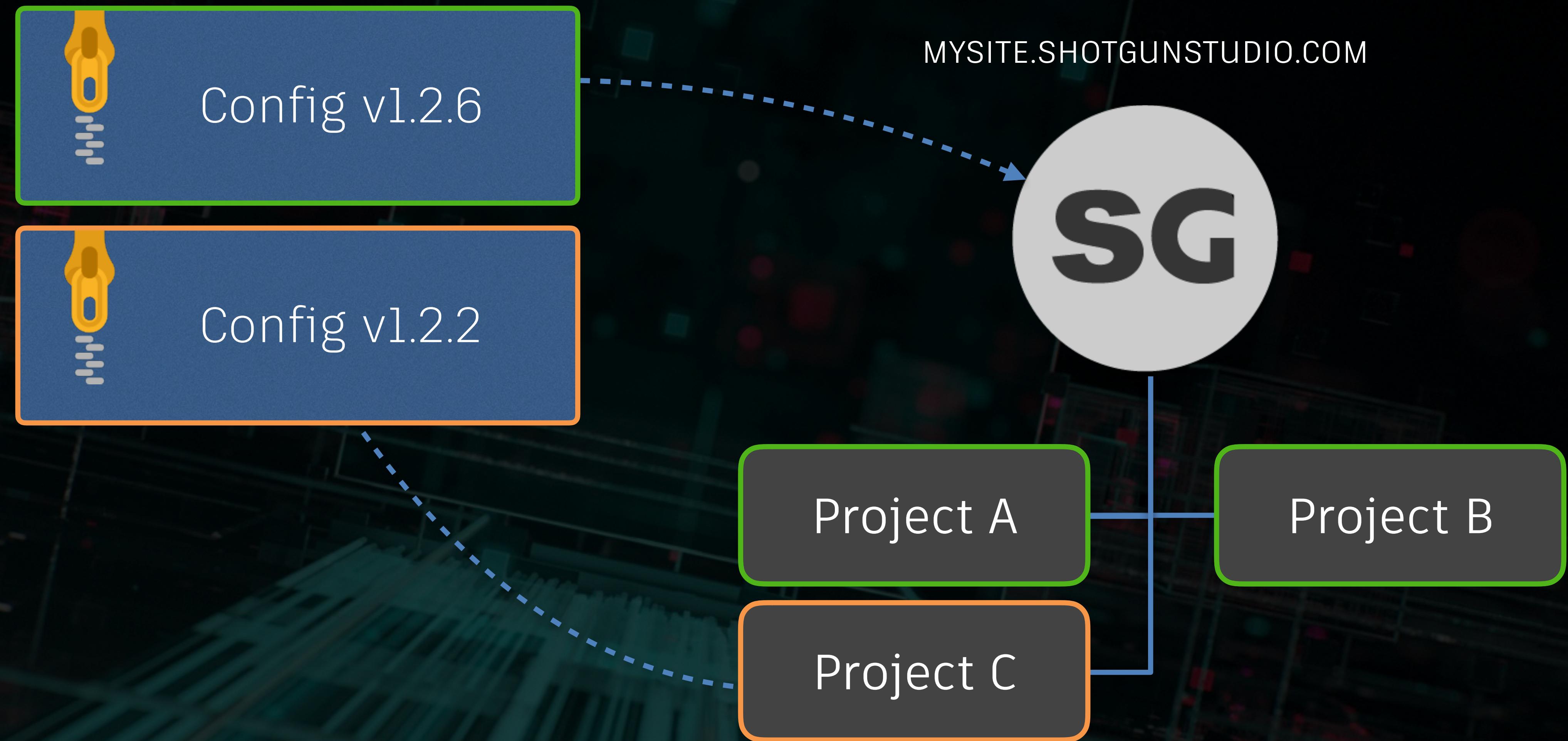
https://github.com/pulls



Beyond the basics

Learn how you can extend your basic setup.

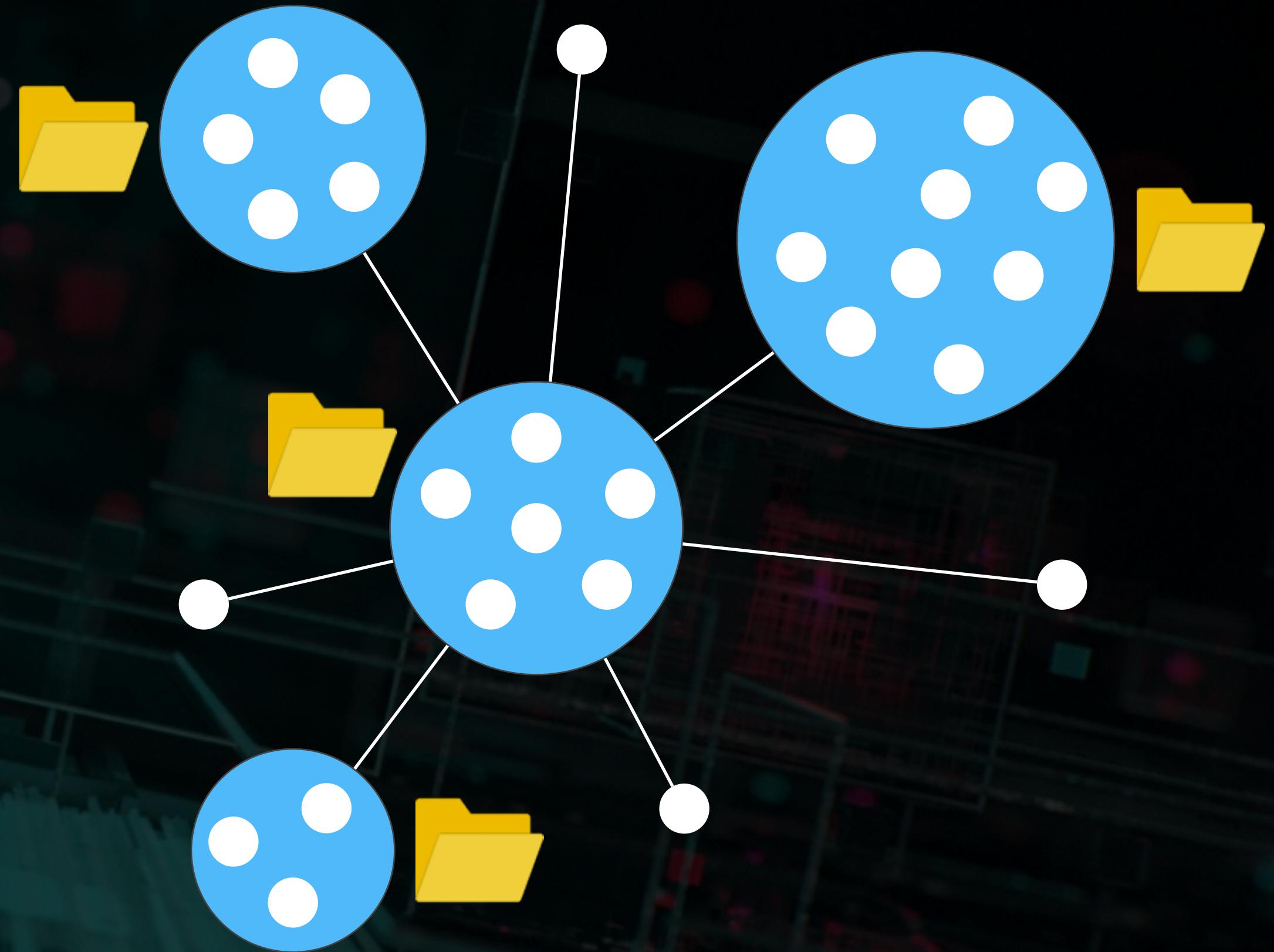
# One configuration for all projects



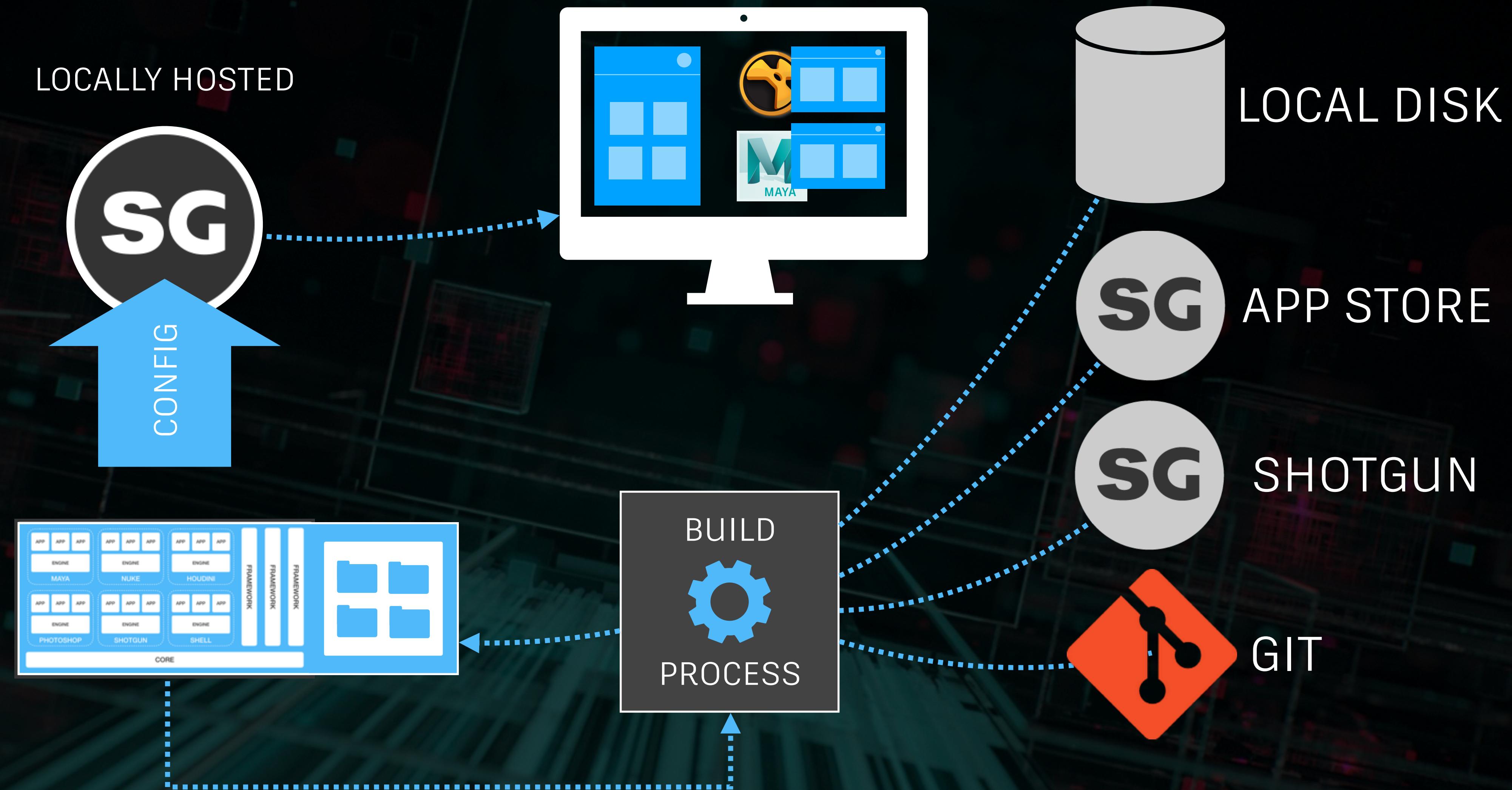
# Shared Bundle Caches



~/.shotgun  
~/Library/Caches/Shotgun  
%APPDATA%\Shotgun



# Self contained setups



# Further Reading

[developer.shotgunsoftware.com/tk-core](https://developer.shotgunsoftware.com/tk-core/initializing.html)

The screenshot shows a web browser window with the URL <https://developer.shotgunsoftware.com/tk-core/initializing.html>. The page title is "Initialization and startup — tk-core". The left sidebar contains navigation links for "Overview", "Initialization and startup" (which is expanded), "Core", and "Descriptors". The main content area is titled "Distributed configurations" and explains that these are defined as Pipeline Configurations inside ShotGun, managed by the Bootstrap API at startup. It describes a decentralized workflow where each user caches components locally. A diagram illustrates this: a central "Shotgun" node connects to four users (User A, User B, User C, User D). Each user has a local "Bundle Cache" containing "Project A Configuration" and "Project B Configuration". A note at the bottom states: "To make distributed configurations available to Shotgun, start ShotGun in a local file system."

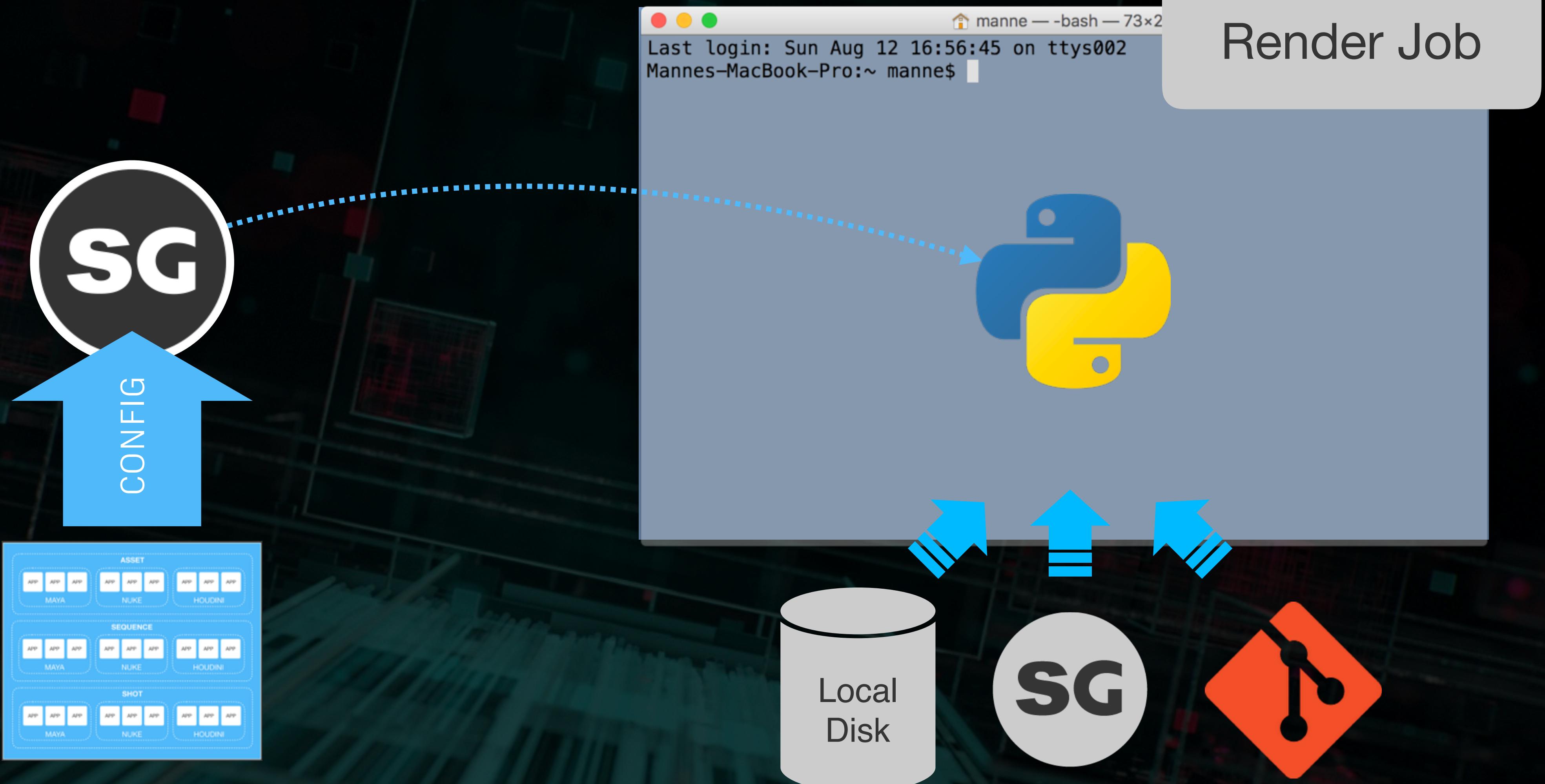
## Distributed configurations

Distributed configurations are defined as Pipeline Configurations inside ShotGun. At startup, these are detected by the Bootstrap API which will automatically manage your local configuration and resources on disk, ensuring that you have all the right apps, engines and other dependencies needed to run the configuration. This is a decentralized workflow where each user will automatically cache all necessary runtime components locally.

**Note**

To make distributed configurations available to Shotgun, start ShotGun in a local file system.

# Remote Compute / Bootstrap API



# Further Reading

DIGIPRO 2018:

## Cloud-based Pipeline Distribution for Effective and Secure Remote Workflows.

6 Distribution patterns.

**Cloud-based Pipeline Distribution for Effective and Secure Remote Workflows**

Manne Öhrström  
manne@shotgunsoftware.com  
Shotgun Software

Rudy Cortes  
rudy.cortes@pearlstudio.com  
Pearl Studio

Josh Tomlinson  
josh.tomlinson@shotgunsoftware.com  
Shotgun Software

Satish Goda  
satish.goda@pearlstudio.com  
Pearl Studio

**ABSTRACT**  
We present a secure, cloud based distribution system for just-in-time artist workflows built on the Shotgun Toolkit platform. We cover the original motivations behind this work, the challenges faced, and the lessons learned as the technology has come to unlock new patterns for managing where and how artists contribute on production. A case study of the technology and its use on production by Pearl Studio is included, showing how the company uses the system to meet their distributed organizational needs and why adoption has been beneficial for their technological and business goals. We show how the system began as a means of downloading and caching individual pipeline components via an app store, before organically evolving into a distribution mechanism for a studio's entire pipeline. We include real-world examples of these patterns that are in use by Toolkit clients and illustrate how this technology can be applied to cloud-based collaboration in a variety of ways.

**CCS CONCEPTS**  
• Information systems → Multimedia content creation;

**KEYWORDS**  
cloud, distributed, workflow, platform, pipeline, production pipeline, Shotgun, Shotgun Toolkit, Toolkit, Pearl Studio

**ACM Reference Format:**  
Manne Öhrström, Josh Tomlinson, Rudy Cortes, and Satish Goda. 2018. Cloud-based Pipeline Distribution for Effective and Secure Remote Workflows. In *DigiPro '18: The Digital Production Symposium*, August 11, 2018, Vancouver, BC, Canada. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3233085.3233096>

**1 INTRODUCTION**  
In today's global marketplace, production studios are more distributed than ever. Sister studios working in different timezones, artists working from home, and vendors in facilities many time zones removed from production management are not uncommon. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior special permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
DigiPro '18, August 11, 2018, Vancouver, BC, Canada  
© 2018 Association for Computing Machinery.  
ACM ISBN 978-1-4503-5895-8/18/08...\$15.00  
<https://doi.org/10.1145/3233085.3233096>

**2 MOTIVATION**  
As production studios trend toward more distributed production models, the traditional model of a single physical site with a large, centralized storage can present challenges for studios looking to evolve their pipeline. Such pipelines often exist as webs of complex code dependencies that can incur large-scale refactors or rewrites in order to function in a world where productions require more and more distributed resources, including data, compute, and workforce. Furthermore, due to increased security and workflow complexity, there is a growing need to ensure that the right people are given access to the particular tools and services they require.

In an ideal world, artists throughput, ease of collaboration, and the ability to effectively manage production data would remain at least constant, if not improve, as a studio evolves a pipeline towards a decentralized setup. To that end, it is important that studios consider ways to maintain familiarity of tools and processes across physical boundaries while building, rebuilding, or extending their pipeline into new locations and for new people. For newer studios, or facilities that have not yet invested heavily in pipeline development, access to an out-of-the-box workflow solution and a secure platform that supports a distributed infrastructure is highly desirable.

**3 BACKGROUND**

**3.1 Shotgun**

Shotgun is a subscription based cloud service for tracking production data. Subscribers have access to a dedicated Shotgun website that includes tools for project scheduling, tracking work to be done, and reviewing and sharing media, to name a few. Shotgun itself can be customized to track any additional custom data that a studio may require.

In addition to the web application, subscribers have access to a number of other technologies provided by Shotgun. These include Python and REST APIs for querying and updating the data being

DigiPro '18, August 11, 2018, Vancouver, BC, Canada

tracked in Shotgun, and the Toolkit platform which provides out-of-the-box integrations with many off-the-shelf digital content creation packages (DCCs).

Manne Öhrström, Josh Tomlinson, Rudy Cortes, and Satish Goda

3.2 Toolkit Platform

The Shotgun Toolkit platform (Toolkit) has offered a means to create and orchestrate highly modular pipeline building blocks for many years [Tomlinson et al. 2017]. Toolkit consists of configurable workflow Apps which are cross compatible with multiple DCCs. Toolkit Engines implement an adapter layer for each supported DCC, creating a communication bridge between the software and the Apps. The Apps are functional, typically user-facing components of a pipeline that are decoupled from the data that flows through them. Apps are highly reusable across various production departments, production types, and even industries [Tasker 2018]. Toolkit configurations (configs) allow studios to organize these components in different ways to build custom workflows used by their artists, and effectively codify a pipeline or a workflow. Configs can be orchestrated to describe site-wide, per-project, or even per-user pipelines.



Figure 1: Structure of a Toolkit Configuration

Toolkit configurations contain settings and customizations for all apps, effectively defining a series of pipeline workflows. The config also holds pointers to each app and engine it requires in the form of a **descriptor**. The descriptor defines which version and from where the code payload of the app or engine should be pulled during the config installation process. Toolkit supports several descriptor types including:

- **app\_store** - an item in the Toolkit App Store
- **shotgun** - an item uploaded to a Shotgun site
- **dev** - a local developer sandbox
- **git** - a tag in a git repository
- **git\_branch** - a commit in a git branch
- **path** - a location on disk

### 4 IMPLEMENTATION

Until recently, using Toolkit implied using the single location, single storage studio model. Under that model, Toolkit configs, and the components they reference, were downloaded and cached on a shared file system during the initial set up phase. This step-by-step installation process required a Pipeline TD or admin to oversee.

<sup>1</sup><http://gt-wcm.com/>

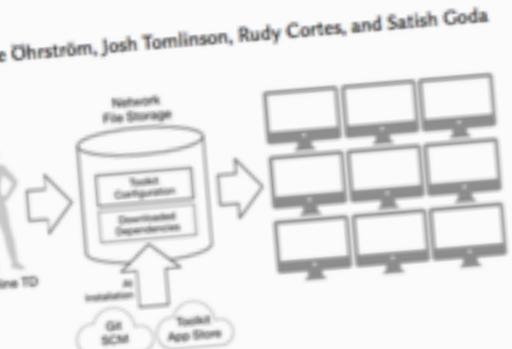


Figure 2: Toolkit config and components accessible by artists using a shared storage

Starting in 2016, based on observations from existing clients, industry trends, and the desire to provide an out-of-the-box experience for new clients, the Toolkit team began extending the platform to include on-demand pipeline distribution and installation. To make this possible, Shotgun was modified to allow studios to upload and store a complete Toolkit configuration and associate it with a project. Existing application integrations and APIs were then updated to determine the appropriate configuration at session startup based on the artist's working production context. These changes introduced the ability for a required config to be downloaded directly from Shotgun and for all of the workflow components it required to be subsequently downloaded and cached from their respective, potentially remote, locations. The end result is a just-in-time workflow installation process and dependency-management solution that requires no Pipeline TD or admin to oversee.

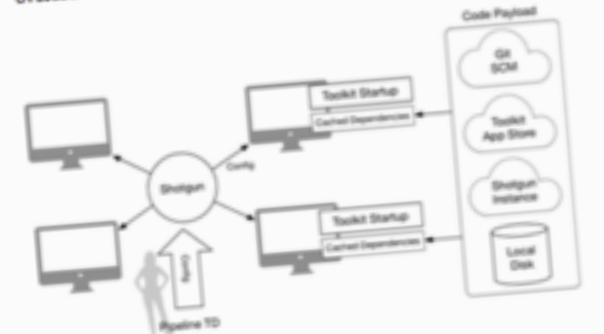


Figure 3: Shotgun serves complete workflows to artists as needed

These changes allow artists, working from anywhere, to launch their content creation applications with integrated, context-aware workflows backed by Toolkit and without the need for a pipeline installation maintained at a centralized shared storage.

This new workflow distribution process inherits the secure nature of all data stored in Shotgun. In order to access configs or apps uploaded to a Shotgun site, you need to be authenticated with that



Come talk to us!



Questions?

