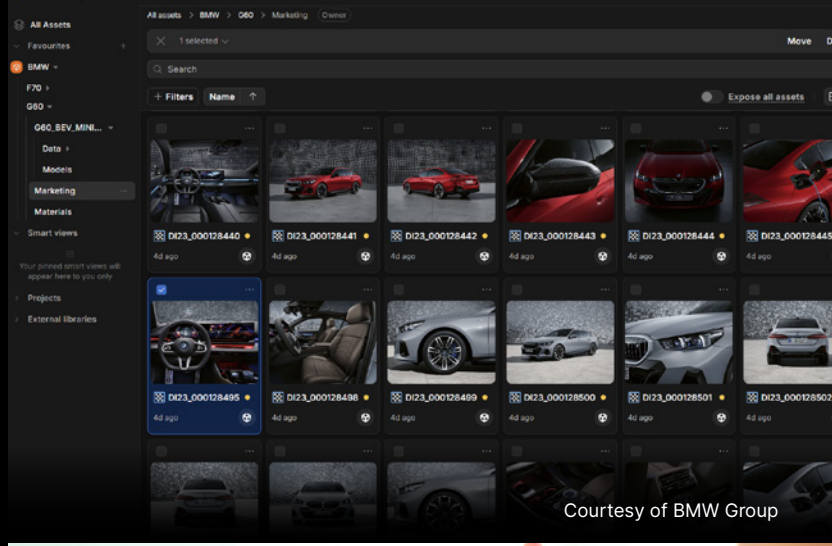
→ PLAYBOOK

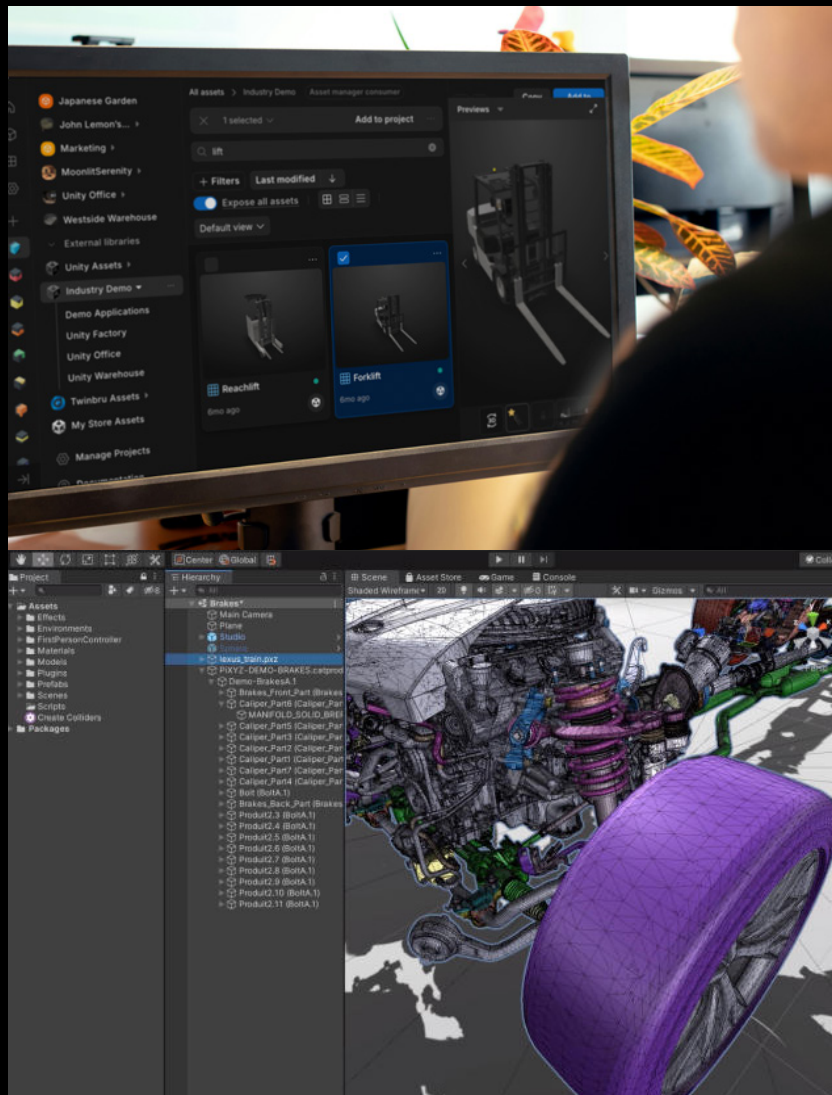
Establishing one source of truth for 3D asset data

How to connect your teams to the data they need: A playbook for engineering, training and operations teams seeking to unify 3D assets across the enterprise.

Unity®



Courtesy of BMW Group



Introduction

Most industrial teams have a wealth of 3D asset data, but it's stubbornly scattered across silos, owned by different groups and hard to reuse. That's a problem because, by definition, when your data lives in different places, people make decisions based on different truths. For example, a trainer might hand a new hire a VR simulation without knowing that engineering updated the line layout sometime earlier. The trainer has to either stop to rebuild with the latest data or continue with content they can't trust. Both cost time and credibility.

The solution is a single repository for all 3D assets that keeps hierarchy and metadata intact and grants everyone who needs it access to approved models only. When everyone pulls assets from the same library, you eliminate version drift and rework, while also gaining visibility into what changed and who changed it.

This playbook shows how to connect all your 3D data sources into a common library that keeps what's important at the time of import and prepares assets for easy access and use across roles. The goal isn't to deploy a new tool for its own sake but to reduce the time spent on nonvalue-added work and avoid unnecessary management complexity when applying access and compliance policies.

The disconnect: Scattered data and duplicate effort

Industrial data is everywhere but rarely connected. “A designer or architect might have access to CAD or BIM, but an engineer might not have the software, so teams end up re-creating models from scratch,” said Sarah Lash, senior vice president and general manager, industry, at Unity.

Silos often occur because of operational or technological gaps. For example, engineering might store CAD files in a product life cycle management (PLM) system, whereas other teams export copies to separate drives or apps to the point you end up with multiple versions of the same asset. “The most common issue is working from two different versions of the same model, which practically guarantees rework,” Lash said.

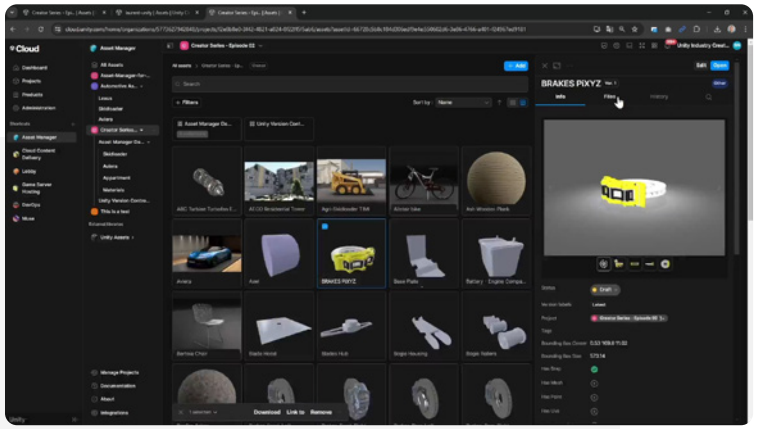
The use of multiple platforms, tools and formats, while often necessary, makes it difficult to track and share assets. This hampers productivity, with people spending hours searching for the right assets or even rebuilding ones that already exist because there’s no easy way to tell which one is the correct one.

As Henning Linn, senior director, industry customer success, at Unity, said: “If there’s a high risk of duplication, there’s often a need for rework, because assets aren’t aligned, and teams aren’t working from the same source file.”

The asset management complexity only increases as these mismatches surface downstream, such as during training and delivery, where they result in rework, missed deadlines and inconsistent experiences. But the cost isn’t just technical. It also undermines confidence between departments, as well as between brands and customers themselves. After all, trainers would rather avoid launching sessions using outdated or unverified models, even if it leads to extended and inconsistent onboarding for new hires. Other teams might end up building “shadow” libraries to stay productive, while IT scrambles to secure and govern an environment it can’t see. Without a common foundation, every new project has to start from scratch.

What to connect: Creating a centralized repository that preserves what matters

Your 3D asset library likely spans many tools and platforms, so the first step to leveraging the models you have is to import everything into a centralized repository like Unity Asset Manager. This way, you won't have to rebuild anything you already have. "We make sure we can import all the relevant data in a preferred and qualitative way, and there's practically no file format in the industry that we can't support," Linn said. "By converting the data into a harmonized format, enriching it as required and tracking every change, it becomes easier to manage those assets over their life cycles."



Before unifying your data, you need to understand what you're connecting and why. Most industrial 3D pipelines draw data from four main sources, each with different structures and priorities:

- Computer-aided design (CAD) models, typically stored in PLM systems, are the primary source of record for parts, assemblies and mechanical properties.
- Building information modeling/management (BIM) models include building and infrastructure data, are rich in spatial and compliance metadata, and are usually stored in BIM software or AEC (architecture, engineering and construction) repositories.
- Meshes from digital content creation tools include visualization assets used in areas such as marketing, training and user experience and are often optimized for visual fidelity rather than technical detail.
- Point clouds and scans, common in extended reality (XR) and virtual reality (VR) applications, include data captured from things such as laser scanning and photogrammetry.

Each source represents the same physical assets — a product, assembly line or entire facility — from a different perspective, such as function, space, appearance or, in the case of point clouds for XR, as-built physical reality.

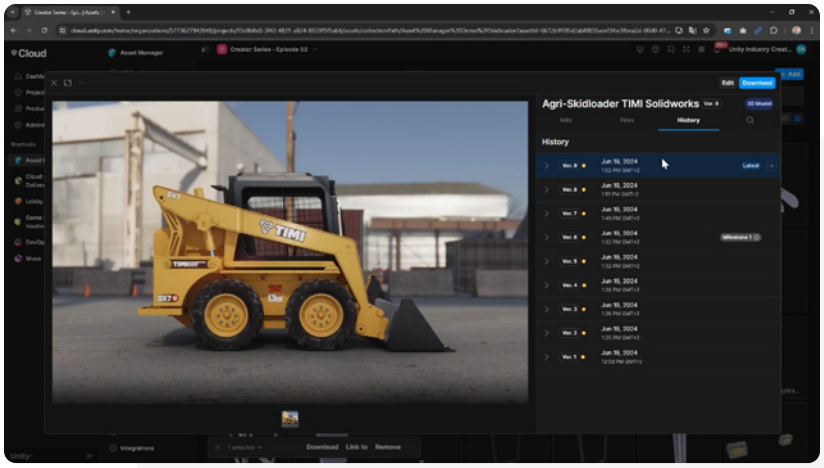
Start with the models your teams reuse most since you'll likely need to prioritize depending on the size of your asset library. A common requirement, no matter the source, is to preserve context during import, because losing vital metadata will only necessitate rework later. Before importing, identify the metadata fields you can't afford to lose. "Always keep identifiers and technical attributes, such as height and weight, and make sure to name and version everything so that assets stay traceable and usable," Linn recommended.

Some fields should always come through intact:

- Identifiers such as part numbers, version codes or unique IDs make it easier to trace assets back to their original sources for verifying their authenticity.
- Hierarchy and grouping metadata, rather than a merged mesh, show how parts fit together or how facilities are organized, allowing engineers to isolate and swap parts as needed.
- Material and unit information, such as density or tensile strength, and descriptive information, such as name or vendor, ensure assets maintain the correct appearance and behavior.

Preserving these details keeps the models useful across applications, whereas without them, they're little more than a picture for quick reference that's disconnected from its purpose.

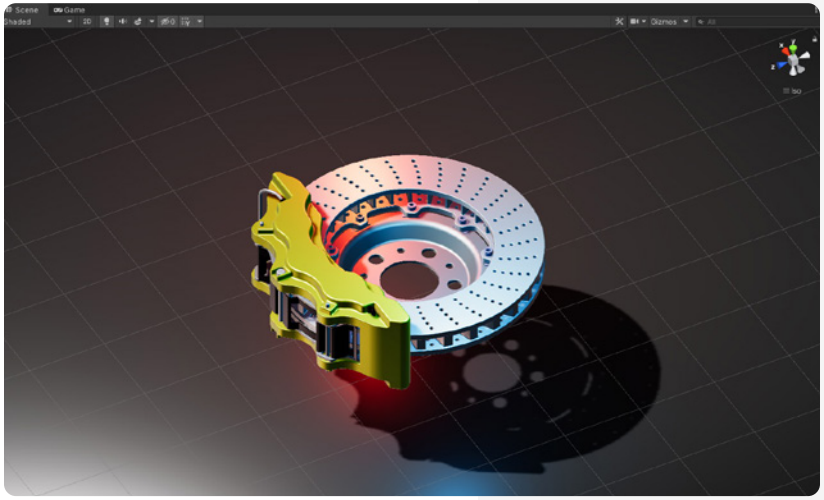
> "Always keep identifiers and technical attributes, such as height and weight, and make sure to name and version everything so that assets stay traceable and usable."
>
> Henning Linn
> Senior Director, Industry Customer Success, Unity

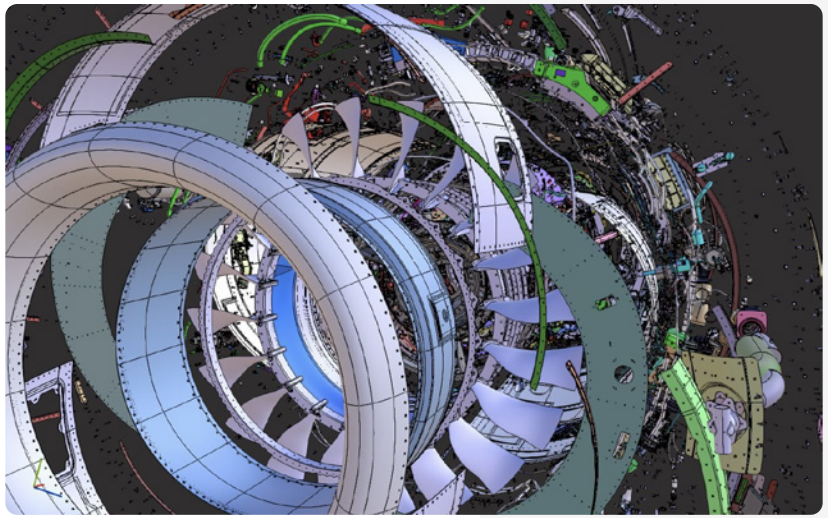


Preparing your assets for import

Assets must be properly prepared before and during import, and there are some common pitfalls to avoid to maintain the health of your repositories. That's especially true for large-scale manufacturing or construction, where 3D datasets can be enormously complex. For instance, a digital twin of an entire factory or a complete model of an automobile might consist of tens or even hundreds of thousands of parts. This can easily overwhelm even the most powerful software and hardware, so breaking everything down into smaller logical groups — such as microchips, connectors or mechanical components — helps keep imports manageable.

Software that prepares 3D data — such as Unity's Asset Transformer Toolkit — is designed to handle these challenges by supporting many widely used CAD and BIM formats, preserving structure and metadata and automatically simplifying and standardizing models during import as needed. For instance, a real-time use case such as an XR simulation for employee training doesn't need every bolt or rivet from the original CAD file. What matters in this case is that the level of realism supports the task. "Depending on the end point, polygon production can range from modest cuts to as much as 90%," Linn said.





The goal is to keep models as light as possible to optimize performance and usability:

- For training and simulation use cases, the goal is to achieve as much visual fidelity as possible at smooth frame rates. Aim for stable frame rates that match the headset's refresh rate, because anything lower can lead to user discomfort.

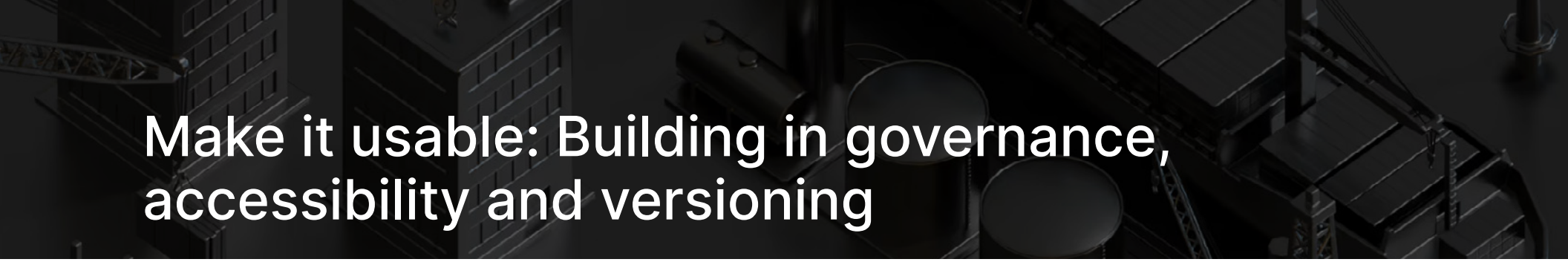
- 3D collaboration and design reviews typically require high functional detail, allowing engineers to inspect things such as fasteners and interfaces. To better manage the demand for high polygon counts, use levels of detail (LODs) for subassemblies that render only in close proximity.

- Human-machine interfaces such as embedded systems and industrial controls typically have limited graphics processing power, so it's best to target the lowest possible complexity, using prebaked lighting and simpler shaders where feasible.

- Customer experience applications can be difficult to optimize for because of the huge diversity in target devices. Aim for a compromise between visual fidelity and loading times, and validate on midtier mobile devices and popular web browsers.

As a rule of thumb, it's best to start with a simplified model and test it on the weakest device you plan to support. Then add detail only where performance allows to avoid having to fix a bloated model after it's already been deployed. For instance, you can generate LODs at import so every asset ships with scalable detail, allowing you to accommodate a wider range of devices and use cases later on as needed. There's no universal correct polygon count, however. What matters is the one that reliably hits your frame-rate and loading-time goals on the target devices. As Linn said, "The range can go from several million polygons to just a few 100,000 for the same asset. What's important is that you're still using the same source file, including all the metadata connected to it."

Courtesy of HERE HMI



Make it usable: Building in governance, accessibility and versioning

With your assets imported and right-sized, the next step is making them secure and discoverable for every role that needs them. The goal here is to curate a single library where your teams know where to look without being overburdened with assets that aren't relevant to them. You need a permission-based access model that doesn't slow people down and lets you maintain a complete audit trail to ensure that every update propagates across your projects.

Role-based access controls (RBAC)

Linn recommended keeping access permissions simple. "You'll have an administrator that oversees the import and creation of assets, and then you'll have contributor and reviewer roles, and that's normally enough." For example, have an admin to define structure and standards, manage users and retention, and approve or archive versions. Keep this user group as small as possible.

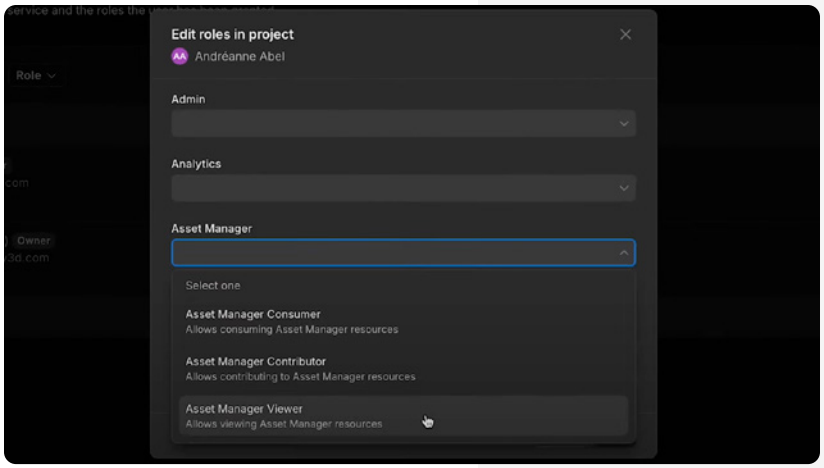
Account type	Level	User role	ID
User	Organization	Asset Manager Admin	c8b054a9-1940-4f05-9059-f3859f1704b3
User	Project	Asset Manager Viewer	4101ecc8-2223-4066-aa4b-6b6f4d0d01e6
User	Project	Asset Manager Consumer	e5411482-bf96-4166-a1e5-07726894b8b5
User	Project	Asset Manager Contributor	c7fcf669-df1b-4a47-a662-4c15e0a29425
Service account	Organization	Asset Manager Search	009c9403-7516-40cb-8bc1-7f708ec307b7
Service account	Organization	Asset Manager Admin	d0d7803a-b0a1-4a71-9350-0712c3da7e59
Service account	Project	Asset Manager Viewer	b5b10e2a-5838-4235-ac3a-9be9fcec6ead
Service account	Project	Asset Manager Consumer	bc1a173e-9f7a-4001-959e-ffc12ff98a7b
Service account	Project	Asset Manager Contributor	5e91cc47-a12f-45ab-bbc5-3dfcdac56b3c

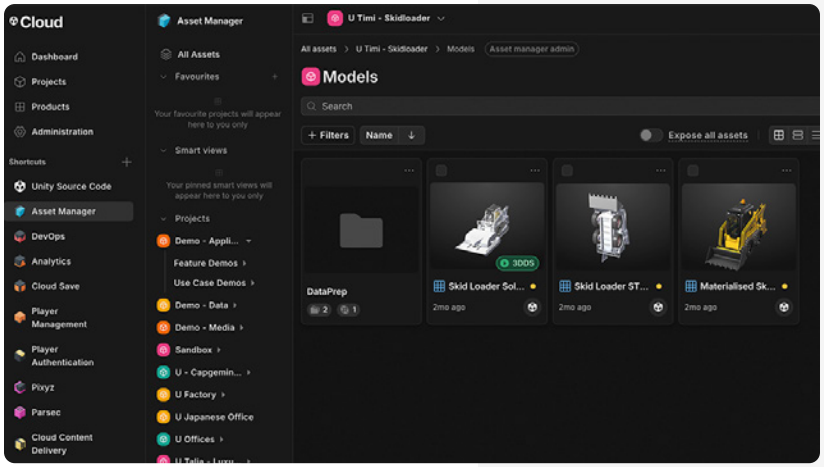
Next in the hierarchy, you might have designers, contributors and editors who can import new assets, edit metadata and publish updates, with admin approval if required. They'll be the ones maintaining the library day to day. Finally, you'll have your consumer roles: employees who can search for, preview and download approved assets but won't be able to modify or publish them.

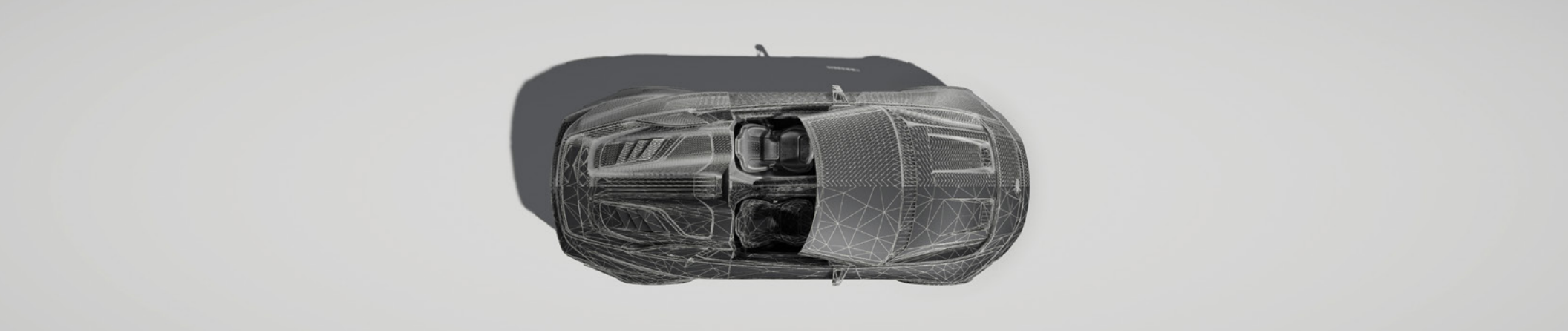
Role-based access controls are also critical for security. Many industrial 3D asset libraries contain highly sensitive information, some of which are subject to regulatory controls. For instance, export-controlled designs for government, aerospace or defense must conform with federal regulations that mandate a strict need-to-know posture, full auditability and deployment in tightly controlled environments. Linn recommended using a virtual private cloud deployment if you're operating in a highly regulated industry.

Regardless of your industry, there are some nonnegotiables:

- Always use RBAC with least-privilege defaults.
- Encrypt all data in transit and at rest in the platforms you use.
- Maintain comprehensive audit logs and approval statuses per version.
- Segment projects for sensitive work and, where needed, data residency options.







Versioning control and auditability

Governance isn't about just security and compliance but also standardizing asset management for ease of access and use. That said, most governance challenges stem from one of two extremes — everyone can change everything or no one can change anything. For instance, a trainer might overwrite a model the engineering team just approved. That's not necessarily due to carelessness, but more often a lack of clear versioning controls. "You'll often have multiple branching incarnations of the same asset, and it all needs to be managed cohesively and in a standardized manner," Linn said.

That's where versioning comes in. The goal is to ensure changes are clear, reversible and intentional. After all, any usable asset library is going to change regularly: A digital factory twin might pick up weekly layout tweaks, while a training program might be updated with new models ahead of a rollout of a next-gen XR headset.

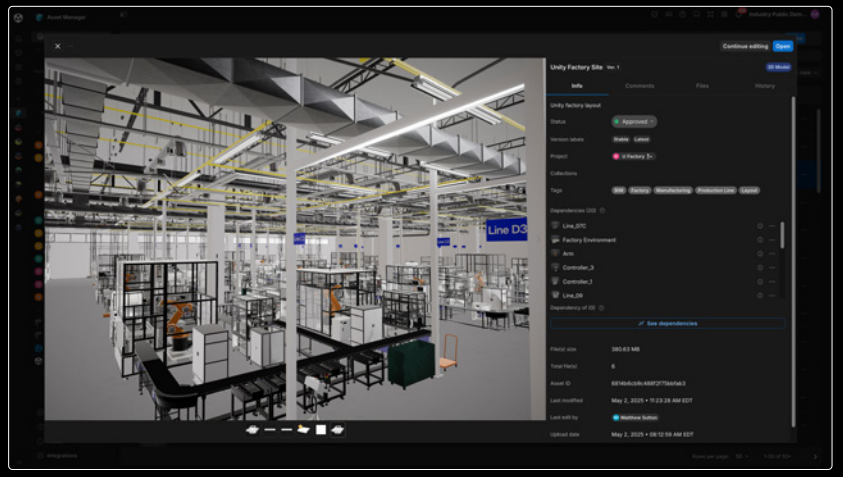
Treat your library as a single — albeit ever-changing — source of truth for real-time assets by applying a linear version history (v1.0, v1.1, etc.) and asset state (draft, in review, approved, retired, etc.).

In many industrial settings, engineering teams maintain source CAD files for official designs. However, visualization or training teams typically use real-time versions of those same models, optimized for their specific purposes. These assets should be linked rather than merged, otherwise it's impossible to tell which CAD version the visualization-ready model came from. For example, if engineering updates a drawing, a training model built months ago might still reflect the earlier version, and no one would know. That's a problem, because it breaks auditability and can lead to errors later on, such as training on the wrong configuration.

As for model variants, be sure to tag them according to their scope and purpose, instead of just making copies with new names. An automotive manufacturer might, for instance, tag them by geographical region, left- or right-hand drive, or form factor. That way, when a base part changes, you'll know exactly which variants need updating.

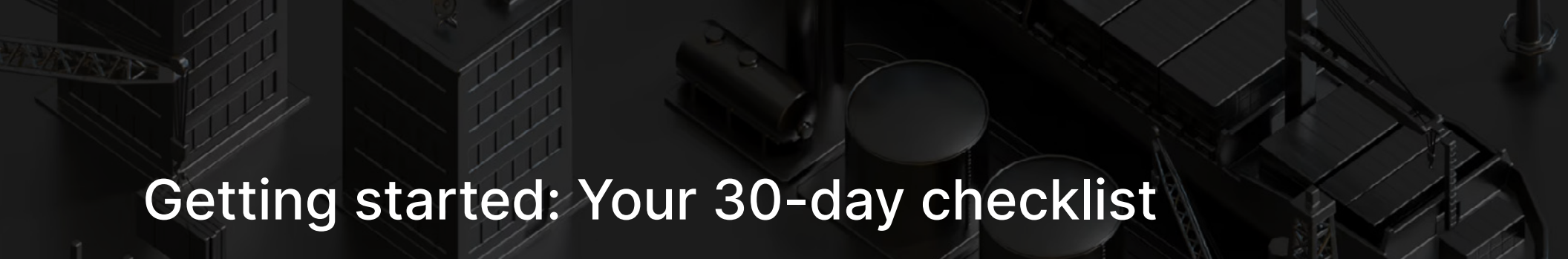
Similarly, if you have derivatives of certain base models, such as a simplified model used for training, label them as derivatives but maintain a reference to the original source and its CAD revision. This way, a trainer won't confuse a simplified mesh used for an XR/VR simulation with engineering's source of record.

Of course, such a granular level of versioning, while essential for usability, can be extremely difficult to achieve at scale, so automation is a necessity. Newer 3D asset managers typically include command line interfaces for performing bulk operations and event-based automation features for generating metadata, previews and tags for newly imported or updated assets.



Here's a quick checklist of operational changes you can make with the right asset manager:

- Adopt a three-role model, with admins, contributors and consumers. Add an approval role for projects that require additional sign-off.
- Map required metadata fields, such as asset identities, revision numbers, original sources, owners and approval states, and set approved-by-default views for most users.
- Link your CAD revisions to their real-time counterparts by labeling variants and derivatives clearly and automating updates where possible.



Getting started: Your 30-day checklist

Once your assets are imported, optimized and governed, the final step is putting your new system into daily use. Then you can quickly prove the value of a unified 3D asset library by creating the immersive experiences that drive real business results — whether it's for training, product development, customer experience or anything else.

Here's a quick recap of what you can do in just 30 days or less:

1. Inventory your data sources and their owners.
2. Select one or two representative models to pilot.
3. Determine which metadata fields to preserve.
4. Test the import to optimize your publishing cycle.
5. Set up your access controls and audit trails.

In our next playbook, we'll explore how to turn connected data into interactive 3D experiences — real-time experiences that enhance training programs, elevate product development, delight customers and more.

Are you ready to connect, create and deploy immersive experiences for your industry? Unity Industry provides a comprehensive suite of tools for training and guidance, 3D design collaboration, CX, XR, and HMI backed by three-year long-term support and a massive developer community.

Start your real-time 3D journey today!

GET STARTED →

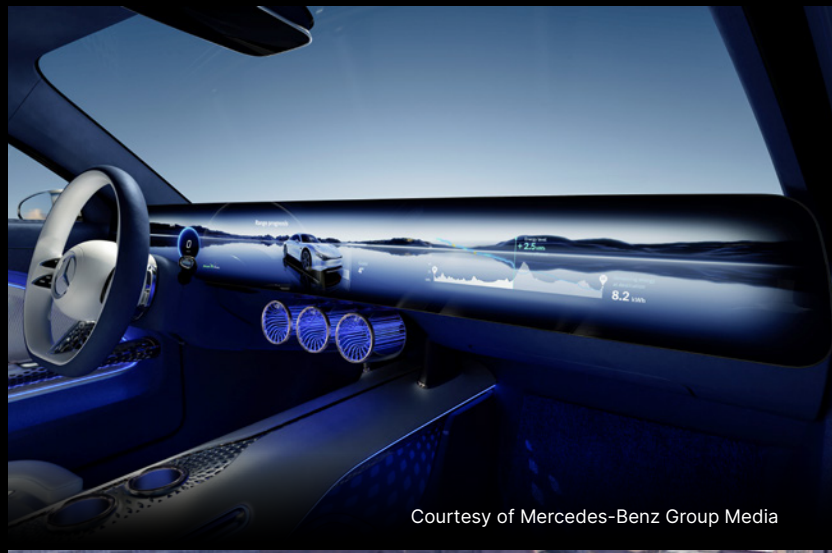
→ PLAYBOOK

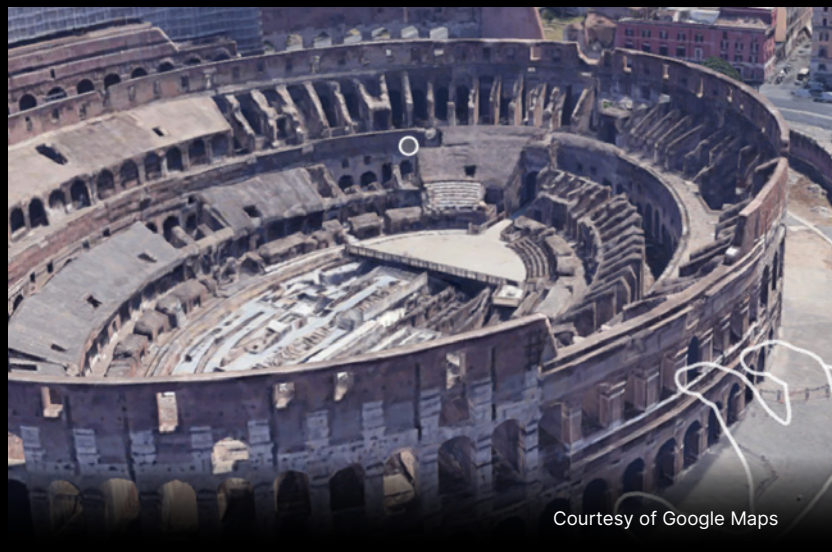
Turning connected data into interactive 3D experiences

How to create immersive experiences that drive business results: a practical guide for product, training and field service leaders looking to build interactive simulations from connected 3D data.

Unity®


Courtesy of Cincinnati Childrens Hospital


Courtesy of Mercedes-Benz Group Media


Courtesy of Google Maps

Introduction

A design review or training simulation rarely fails for lack of data. Most of the time, it fails because the people in the room aren't having the same experience of that data. Engineers bring section views and tolerance stacks, manufacturing shows a snapshot from the assembly line, while product teams show slides to align stakeholders.

There's nothing wrong with these presentations, but they exist in different frames of reference and draw data from different sources. They tell the part of the story that matters only to a specific user role. Your teams might be right in their own frame, but still, important questions linger because not everyone can be certain that the information being conveyed is correct, relevant and up to date.

Put those teams around the same 3D model, and the conversation changes. The assembly moves, tolerance levels breathe, service panels open and issues reveal themselves in seconds instead of days. That way, people can point, test and decide based on the same source of truth.

This playbook is about creating those kinds of experiences. In our previous playbook, we talked about laying the groundwork by building a governed, centralized 3D asset library that imports and preserves hierarchy and metadata. The next step is to turn that data into immersive, interactive experiences that re-create the products you're selling, the simulations you're running or the facilities you're monitoring.

Courtesy of ABB

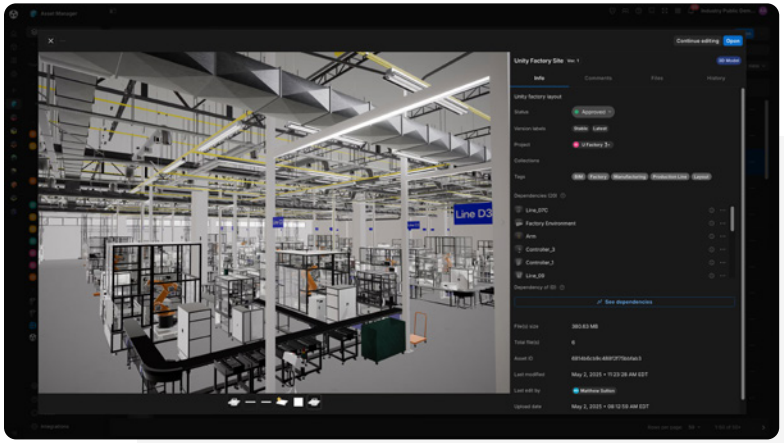
The bottleneck: Connected models, missing experiences

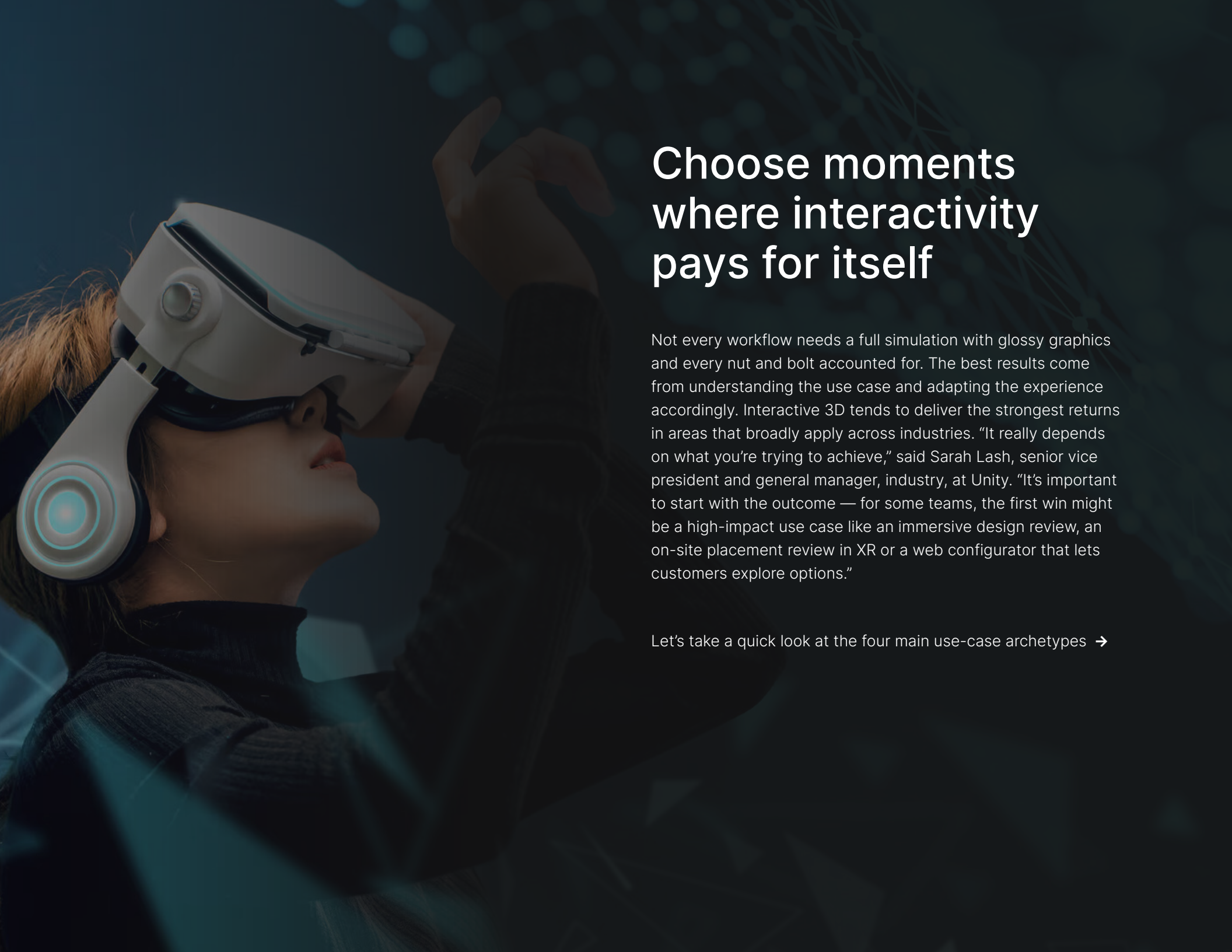
Raw 3D models, no matter how well they're organized, don't create value by themselves. Once you've built your asset library, the next step is to create experiences that give your end users what they need from those models.

Here are some common sources of friction at this stage:

- People have different expectations from the same model: Engineers expect functional fidelity, while trainers need clarity and realism to teach procedures, and customers want engaging visuals that are easy to understand.
- Static outputs don't convey moving problems: Screenshots and slide decks don't show things such as midmotion clashes or awkward tool paths, and sequence errors show up only when you step through the motions or run the task manually.
- All-or-nothing realism: If you're going for maximum detail on XR headsets, frame rates will tank, leading to uncomfortable experiences. Strip away too much detail, and you risk losing trust. Instead of debating aesthetics, focus on what must be accurate for the use case at hand.
- Inconsistent experiences across devices: What runs smoothly on a workstation might stutter on a headset or mobile device. Things such as excessively high-res textures, demanding-shaders or unbaked lighting aren't easy to fix just before a live demo or training simulation.

These issues all share a common theme — a lack of a shared, hands-on view. By treating the model as a starting point rather than a deliverable, you can focus on the desired outcome and design accordingly for that audience and device.





Choose moments where interactivity pays for itself

Not every workflow needs a full simulation with glossy graphics and every nut and bolt accounted for. The best results come from understanding the use case and adapting the experience accordingly. Interactive 3D tends to deliver the strongest returns in areas that broadly apply across industries. “It really depends on what you’re trying to achieve,” said Sarah Lash, senior vice president and general manager, industry, at Unity. “It’s important to start with the outcome — for some teams, the first win might be a high-impact use case like an immersive design review, an on-site placement review in XR or a web configurator that lets customers explore options.”

Let’s take a quick look at the four main use-case archetypes →

1. Training and simulation

An effective simulation can dramatically improve learning outcomes and safety, but they depend on immersion, accuracy and trust. A well-built training simulation exposes the correct sequences, the feel of consequence and the reasoning behind each step. For example, if you use a simulation to train a forklift or crane operator, load sway and momentum train judgment in a way that video never can. In maintenance tasks, realistic access paths and real-time feedback on errors help new technicians gain the confidence they need to work productively and, most important, safely.

Courtesy of I-CAR

WHAT TO AIM FOR →

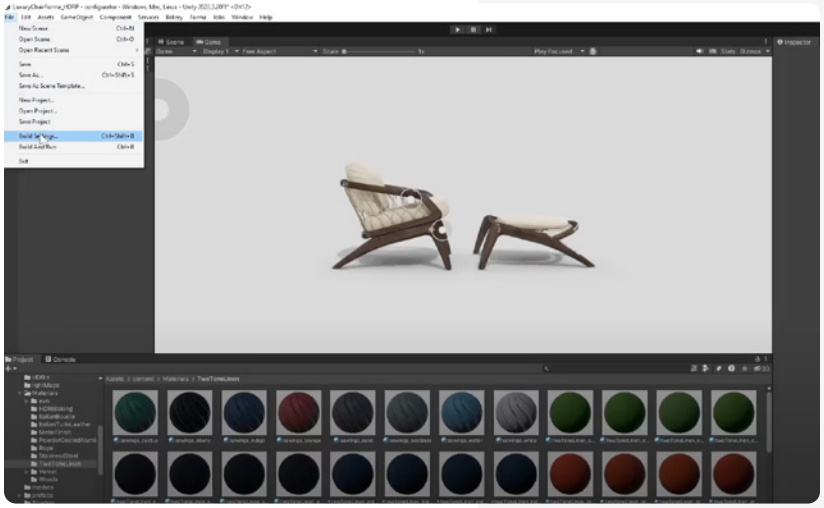
Believable and accurate behaviors relevant to the particular training scenario. Clear visual cues and stable frame rates on the target headset or tablet are much more important than aesthetics, although the simulation does still need to reflect the real-world environment or object.

Courtesy Specto Medical AG

2. 3D collaboration and design reviews

In design reviews, interactivity turns debate into meaningful demonstrations, or at least it should. This high-payoff use case allows engineers, designers and stakeholders to review designs in real-time 3D, allowing them to catch errors earlier and iterate faster. Reviewers should be able to switch between variants, move parts through their full motion to see if anything clashes and cut away layers to check clearances in context. Smarter decisions come a lot faster when everyone's looking at the same thing, at the same moment, from the same point of view. Visual polish is typically optional here, but realistic motion and accurate details are not.

Courtesy of Facebook Reality Labs



WHAT TO AIM FOR →

Accurate sizes and movements, such as those showing how far parts travel without colliding, are paramount. You might also include simple risk toggles, such as turning off guardrails and enabling self-service modes to reveal issues and worst-case scenarios.

Courtesy of Mercedes-Benz Group Media

3. Human-machine interface operations

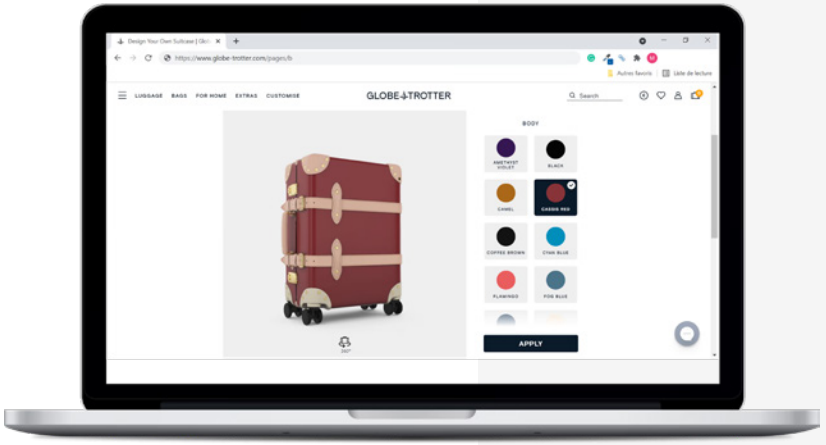
Whether it's an assembly line control screen or an in-vehicle display, clarity is most important in HMI applications. Operators need to know what's happening now and what they should do next, so lightweight 3D tends to win in these cases. Moreover, many embedded systems, which users interact with via an HMI, have limited graphics processing, so high visual fidelity isn't usually technically feasible anyway. Other factors, such as responsiveness and legibility, are nonnegotiable, because when situations change, operators may have only seconds to respond. Any lag or hard-to-read UI elements can result in errors, downtime or safety incidents.

WHAT TO AIM FOR →

Use lightweight 3D models that render instantly on the target hardware. Simplified meshes, prebaked lighting and high-contrast materials reduce performance demands and ensure the models remain readable in typical HMI use cases.

4. Customer experience and configurators

Customers spend more time and decide faster when they can interact with a product they're interested in virtually, such as spinning it around, zooming in, changing options or even dropping it into their own surroundings with augmented reality. Trust comes from having fidelity where it matters, so materials and lighting need to be realistic, and moving parts should behave as they do in real life. Nonetheless, while customers appreciate glossy visuals,they certainly don't want choppy frame rates or long loading times.



Courtesy of SmartPixels

WHAT TO AIM FOR →

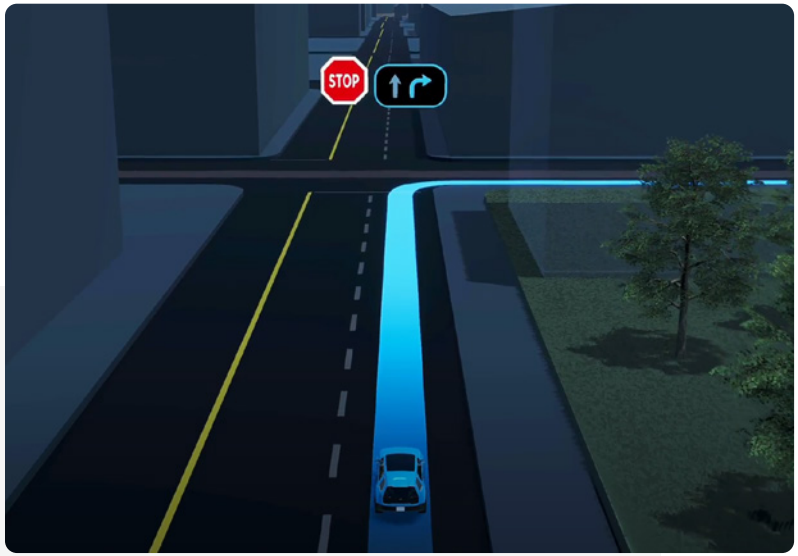
Prioritize what matters most to the customer experience, the so-called "hero" surfaces, named so because they're critical areas of detail — such as glass, metals, paint or fabrics. Fast-loading times on midtier devices are also important, so consider using multiple levels of detail.

Courtesy of Audi & Govar Studios

For best results, start with a single scenario within one of the above archetypes. Build out the use case most important for the target audience, and then you can expand using the same connected 3D models later on to serve other archetypes and use cases.

Design for truth, but build for performance

Confidence and trust come from two things: The scene should look right, and it should behave just as it would in the real world, albeit on the devices the end user actually has. To make your 3D experiences convincing and usable, it's best to take a layered approach: visuals, interaction and integration.



Courtesy of TomTom

1. VISUALS:
Tune for "minimum viable realism"

Finding the right level of visual detail for a given experience is one of the harder aspects. You want to give your end users the best experience possible, but you also need to accommodate the devices that will convey those experiences. On one hand, low-quality visuals can fail to carry enough meaning. On the other, poor frame rates result in experiences that feel unresponsive and less immersive.

Avoid overengineering visuals by making something more photorealistic than it needs to be. You don't want to spend weeks building out ultradetailed textures for an internal training experience designed to run on midrange VR headsets, for example. Trainees might not even notice the extra polish anyway, but they'll certainly notice performance lags. Overengineering also happens if teams import every nut, bolt and internal component from a CAD assembly into a real-time scene, even if those details aren't visible or even needed for the experience.

Underengineering visuals can be just as detrimental, however, especially when it comes to customer-facing experiences. For instance, a product demo that uses low-poly models and blurry textures isn't just going to look bad to customers — it can also hurt confidence and trust by representing the product in a poor light. Underengineering also includes missing details that affect understanding, such as a training simulation that omits an important warning label or safety guard the user needs to see. If end users don't trust the simulation or get confused when they view it, chances are it's too far detached from reality.

It's best to aim for minimum viable realism — in other words, the minimum level of visual fidelity you need to achieve the desired outcome. Decide what must remain exact to inform or influence decisions, whether it's materials, shapes, lighting cues or anything else. Start with a small scene communicating function and quality, and preserve geometry where it carries meaning. Then test the experience on your weakest target device, only adding detail when you have the extra frames to spare.

> "Not everything needs to be hyperrealistic. Photorealistic streaming gets very expensive, but if you reduce realism by just 20%, a lot of people aren't going to see the difference. Dial it back to what's 'enough,' and spend frames where it matters."
>
> Henning Linn
> Senior Director, Industry Customer Success, Unity

Courtesy of Travancore Analytics

2. INTERACTION:
Make experiences feel real

In most industry use cases, interaction is more important than visual realism. That means creating 3D experiences that obey real-world rules such as gravity, collisions and fluid dynamics. Of course, this also adds complexity and computation cost. Training outcomes, for example, depend on real-world behavior, but mimicking that behavior with prebaked presets alone is hard to fake. For instance, if you're training a new hire to operate a crane, your simulation must accurately convey things such as weight sway, friction and momentum.

In other cases, complex physics can be overkill. If, for example, you just want to simulate what happens when a door opens and closes in a product demo, a simple animation will suffice. It will look the same every time, but that's not an issue if you don't need variability. After all, a customer doesn't need to see what might happen if the hinge fails. An engineering team, on the other hand, probably does.

For best results, use physics selectively, applying them only where they add educational or experiential value. For everything else, where variability isn't required, use simple scripted animations, becausethey're far less computationally demanding.

For instance, a training simulation on device assembly might need to show screw-tightening with physics so that parts collide as they do in the real world, while noncritical movements, such as a panel sliding open or an indicator light turning on, can use a prescripted animation.

> "Let's say you have a demanding use case like a driving simulator. In this case, you might pair Unity's real-time visuals with a dedicated physics simulator. To keep frame rates high without sacrificing realism, you can use simple animations when determinism isn't needed."
>
> Henning Linn
> Senior Director, Industry Customer Success, Unity

3. INTEGRATION:
Keep your creations moving

A static model becomes a powerful 3D workspace once it's made interactive and connected to contextual data. The goal is to reflect reality in the moment that matters, such as approvals in design reviews, service paths to check for issues before release or specific operations during a training simulation. To give context and meaning to your otherwise static models, you can integrate data from other, non-3D sources.

In many industrial settings, one of the most impactful integrations is real-time IoT or sensor data that feeds directly into the 3D experience. For example, integrating IoT telemetry from an assembly line on a shop floor means simulations can convey current temperatures, speeds and pressures. In training and R&D applications, this real-world telemetry, gathered from connected sensors, boosts operational awareness and decision-making.

Other valuable integrations include enterprise systems such as product life cycle management, ERP and maintenance databases. By connecting your 3D models to these data sources, you can consolidate knowledge and greatly enhance collaboration, reducing much of the back and forth involved in areas such as design approvals or service requests. For instance, imagine opening a model in your asset database and instantly pulling up its maintenance history and inventory status directly from your ERP system.

There are also more industry-specific integrations. For example, in oil and gas, construction, or smart city planning, you might integrate geographic information system and map data, allowing you to overlay your 3D models on real-world map coordinates or add BIM data into geospatial context to help stakeholders see projects in situ. Utilities and energy companies might overlay SCADA telemetry and weather forecasts onto a 3D model of a wind turbine or substation. Logistics and warehousing might integrate real-time traffic data and heat maps to reveal congestion and so forth.

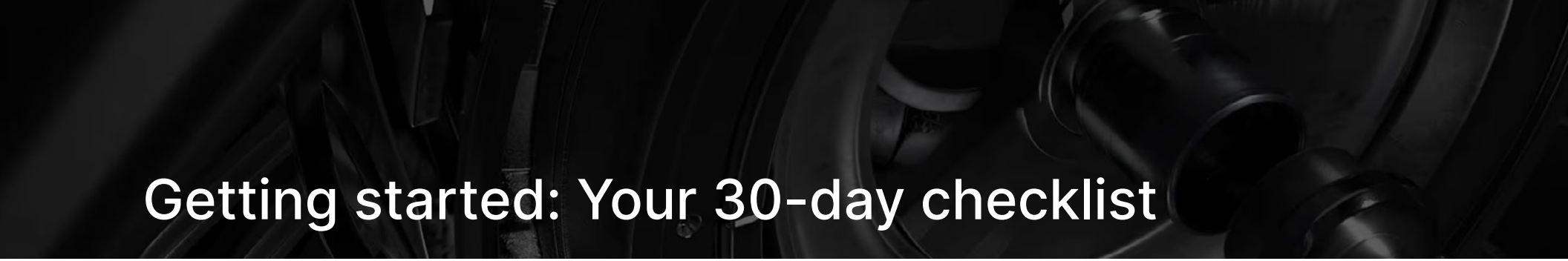
> "With Unity, you can easily integrate IoT data using industry-standard protocols like MQTT (Message Queuing Telemetry Transport) or OPC UA (Open Platform Communications Unified Architecture) into a simulation — whether it's a temperature overlay, live foot or vehicle traffic, or anything else."
>
> Henning Linn
> Senior Director, Industry Customer Success, Unity

4. DEMONSTRATE:
Reinforce stakeholder trust

No matter how well-crafted an experience, teams also need to measure outcomes that show proof of value. For example, you might measure training outcomes with KPIs such as time-to-competency or error reduction, collaboration by design review cycle time, and CX by conversion metrics. To capture these outcomes, teams might use simple usage logs, step completion or interaction frequency. For best results, pick one primary KPI per project and establish a baseline you want to target.

Teams also need confidence that their builds will stand the test of time, hence the value of long-term support commitments such as Unity's 3-year LTS. These provide stability, updates and expert guidance so enterprises don't feel as though they're experimenting in isolation. Confidence also comes from being able to tap into a community of thousands of other practitioners across industries for collective knowledge and open-source tools and assets.

> "Going forward, we're working on a no-code, web-based platform called Unity Studio that will allow nontechnical users to pull assets to create basic scene builds and simulations so that training managers, artists and plant managers can also create immersive applications without having to turn to developers."
>
> Sarah Lash
> Senior Vice President and General Manager, Industry, Unity



Getting started: Your 30-day checklist

Once your assets are imported, optimized and governed, the final step is putting your new system into daily use. Then you can quickly prove the value of a unified 3D asset library by creating the immersive experiences that drive real business results — whether it's for training, product development, customer experience or anything else.

Here's a quick recap of what you can do in just 30 days or less:

1. Inventory your data sources and their owners.
2. Select one or two representative models to pilot.
3. Determine which metadata fields to preserve.
4. Test the import to optimize your publishing cycle.
5. Set up your access controls and audit trails.

In our next playbook, we'll explore how to turn connected data into interactive 3D experiences — real-time experiences that enhance training programs, elevate product development, delight customers and more.

Are you ready to connect, create and deploy immersive experiences for your industry? Unity Industry provides a comprehensive suite of tools for training and guidance, 3D design collaboration, CX, XR, and HMI backed by three-year long-term support and a massive developer community.

Start your real-time 3D journey today!

GET STARTED →

→ PLAYBOOK

Deploying interactive 3D wherever your target audiences are

How to deliver your creations at scale: A playbook for device-aware deployment with the right fidelity, footprint and per-platform rollout strategy.

Unity®

Courtesy of Siemens

Courtesy of Esri


Courtesy of Random42

Introduction

No matter how polished or powerful a real-time 3D experience is, its usefulness depends on who can access it and where. Once you've connected and standardized your 3D asset library and optimized your models for their target use cases, the final step is to make those experiences available where they need to be — across desktop, mobile, web, XR headsets and HMI systems. "The most common platforms, like desktop and mobile, are naturally the easiest to deploy to," said Henning Linn, senior director, industry customer success, at Unity. "But XR and VR can be a bit tricky, because form factors aren't very heterogenous. HMI is even more complicated because every OEM does it differently."

Whether you're hosting an in-depth training walk-through using an XR headset, collaborating on design plans or delivering engaging customer experiences, each platform has its own strengths, weaknesses and use cases. The web is great for accessibility, XR or VR is the go-to choice for immersion, desktop and mobile are suited for everyday utility, and HMI delivers embedded intelligence. In a typical modern industrial setting, you'll probably use a combination of these platform types.

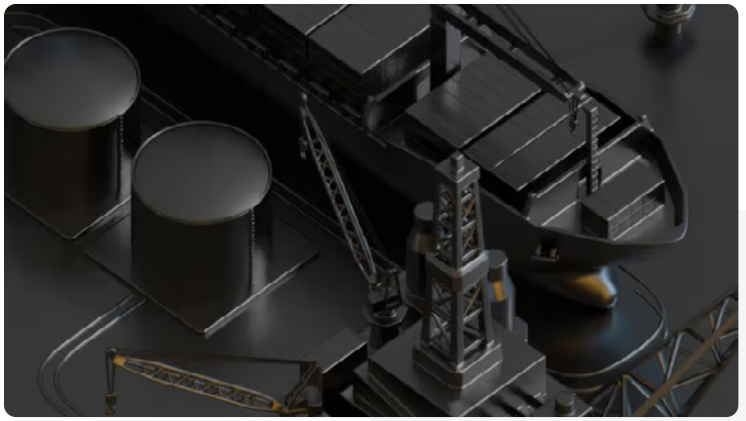
The goal now is to make those designs available to those who need them. But that's not as simple as it may sound because of the immense diversity of devices, each of which has its own capabilities and limits. That's especially true when it comes to customer experience (CX) use cases, where you need to accommodate the broadest range of platforms and devices. However, even internally, for use cases such as training, design and engineering, every team has different needs, expectations and devices.

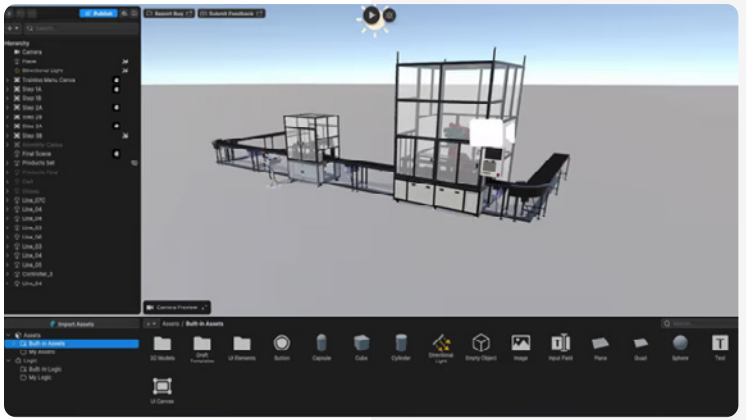
This playbook, the third and final in our series, assumes you have a centralized, connected 3D asset library and you've built and optimized the experiences for your target use cases. Here, we'll look at how to deliver those experiences to your stakeholders or customers no matter where they are or the devices they use. Deployment is all about removing that last mile of friction between great experiences and the people who'll benefit from them.

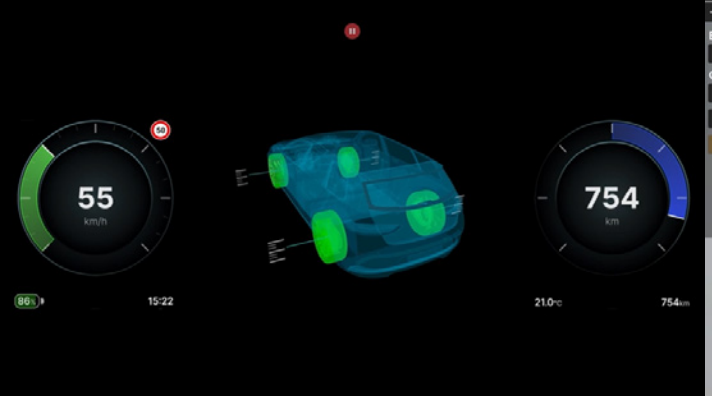
The last mile: Closing the gap between the lab and the field

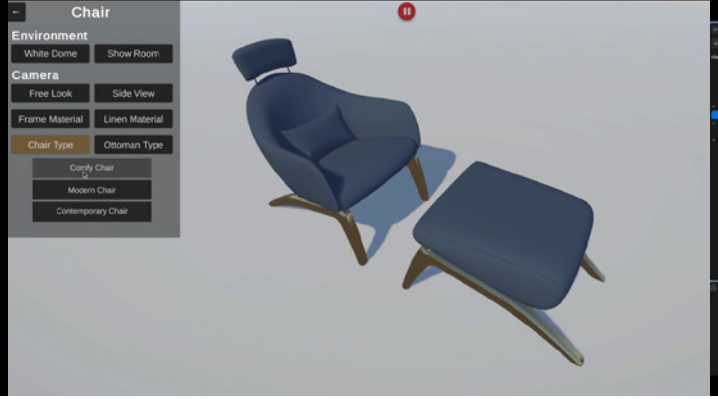
You've probably seen it before: An interactive 3D product demo looks great on a high-end workstation, but the moment it ends up on a midtier tablet on the factory floor, it falters. Frame rates plummet, downloads drag on, XR-tracking stutters under harsh lighting, and the pilot stalls before stakeholders ever feel the value. The overarching theme here isn't that the experiences are necessarily "wrong" — they're just not being deployed in the right way in the right places.

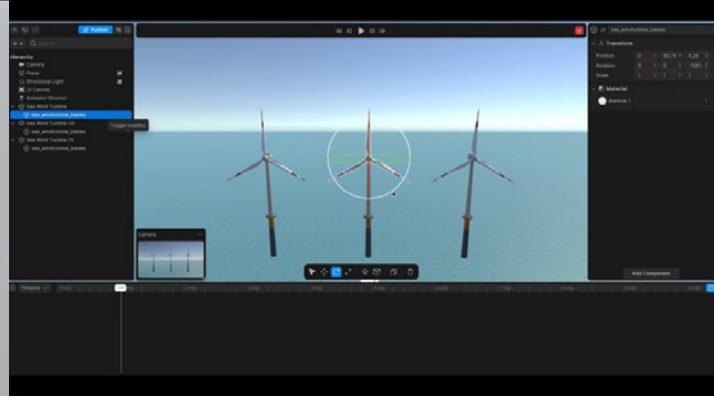
"For many organizations, the issue is a limited pool of developer resources or access to funding, so teams often get stuck at the proof-of-concept stage," said Sarah Lash, senior vice president and general manager, industry, at Unity. "If that happens, the ability to iterate or make changes is limited, which is why we're working on our Unity Studio solution that allows nontechnical teams to whiteboard in a Unity application and start that design and iteration process themselves."











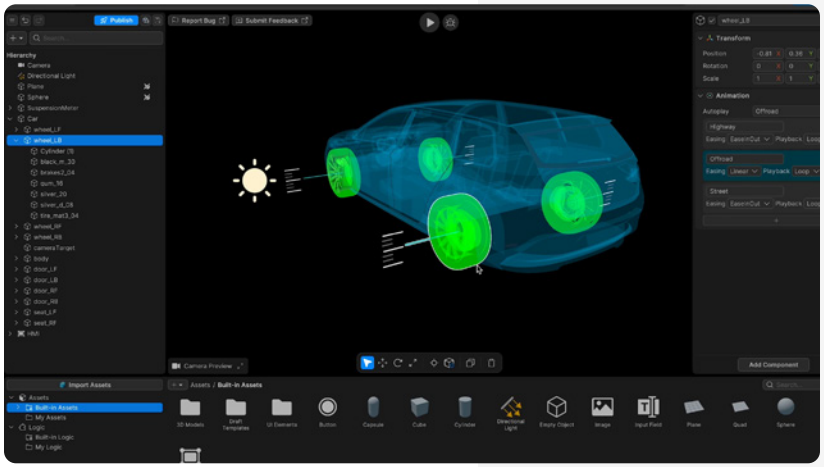
From a technical perspective, there are several common reasons for deployments to fail:

- Performance mismatches happen when an experience isn't optimized for the target hardware. It's normal to use high-end workstations to build and prototype experiences, but you also need to adjust for weaker platforms, such as mobile, web, smartphone or XR.
- Device compatibility and variation result in experiences that might work well on one device but not on another. Experiences can even differ greatly between systems of the same type, such as different web browsers, smartphones or graphics processing units (GPUs).
- Varying real-world conditions can also get in the way of great experiences. For example, an XR application might work in a typical lab environment, but the moment it ends up on a brightly lit factory floor, motion tracking could become unstable if not suitably optimized beforehand.
- Scalability problems, especially in the case of larger projects, tend to manifest because of a lack of robust multiplatform or multiuser architecture. For instance, a content delivery network (CDN) might not be ready to handle multiple simultaneous downloads of a large model.
- Finally, there's the human element, where a lack of sufficient training or support for end users can lead to low adoption rates. Even something seemingly minor can result in users seeking undesirable workarounds or, in the case of customers, turning to competitors instead.

Of course, some devices are more challenging to accommodate than others, but again, it depends on the use case and your desired outcome. Web browsers, despite being universal, are one of the tougher platforms to accommodate because of limited memory and a lack of a dedicated GPU in many cases. XR headsets (AR or VR) present challenges in both performance and comfort. Embedded HMI systems, such as those found in vehicles or on assembly lines, tend to run on highly customized hardware, which is rarely best-in-class when it comes to graphics processing.

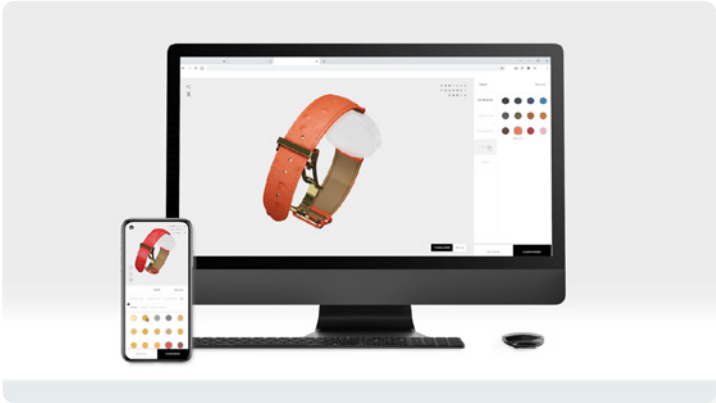
The list goes on, but there is a foolproof way to test deployment readiness and optimize as necessary. By testing on a range of target devices in the appropriate real-world conditions, as early and as often as possible, you can iron out any issues before it goes live. “The earlier you have a chance to test things, the easier the final production deployment will be,” Linn said. “Test on real end points earlier so deployment ‘just works’ when it matters. For example, start with complex device types like embedded HMI or XR to better understand the hardware patterns and requirements and iterate from there.”

> “The earlier you have a chance to test things, the easier the final production deployment will be. Test on real end points earlier so deployment ‘just works’ when it matters.”
>
> Henning Linn
> Senior Director, Industry Customer Success, Unity



Optimize for performance across devices and end users

Deployment isn't about rebuilding your asset library or redesigning your creations. It's about choosing and accommodating your target devices and optimizing the tool chains that connect experiences to the desired outcomes. A single authoritative 3D model can — and should — work well on any platform, provided you plan for multiple levels of detail (LODs), informed lighting choices, and adjustable physics settings tied to device types and their capabilities. As Linn said: "It really depends on what you want to achieve, but you can come to some good results with certain compromises, and that applies to HMI and XR experiences, too. It's important to know all your end points up front and ramp up certain fidelity levels for higher-end end points."



Courtesy of SmartPixels

Courtesy of Prespective

1. WEB:
Universal low-friction access

Nothing beats the web for reach and accessibility. Most web browsers are largely cross-compatible in that they all support HTML5, CSS and JavaScript. At the same time, even modern browsers share much the same limits, especially with regards to download sizes, memory footprints and startup times. However, rendering 3D graphics without plug-ins — which is necessary for supporting the vast majority of users — also requires adequate hardware. For example, while integrated GPUs can handle lighter scenes, higher payloads and shader complexity will bog down most midtier laptops or mobile devices.

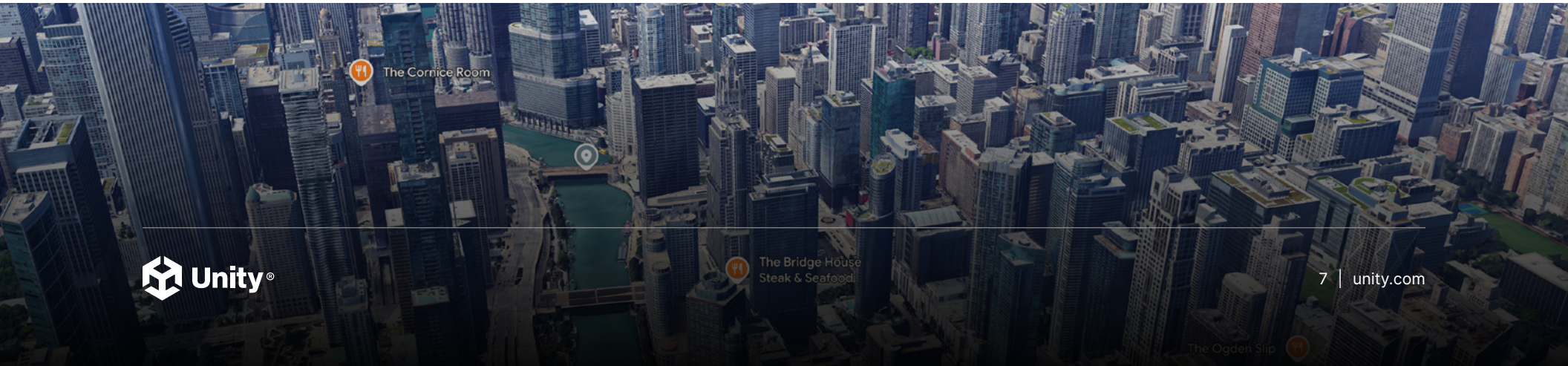
Treat initial load times as prime real estate, especially in the case of customer-facing experiences. If an experience doesn't load up in seconds on a typical broadband connection, people aren't likely to stick around. For more complex assets, it's best to stream multiple LODs on demand, rather than trying to load everything at once. Also, keep an eye on device variability.

Older laptops with integrated graphics, for instance, will expose edge cases where the experience stutters or lags. To avoid these pitfalls, test your experiences on multiple midtier devices, set fallbacks for unsupported features and aim for at least 30 frames per second (FPS) for casual viewing.

WHAT TO AIM FOR →

Aggressive control over texture size with multiple LODs, removed or simplified meshes for invisible internals, and streamed content loading that prioritizes "hero" surfaces (critical areas of detail — such as glass, metals, paint or fabrics) and interactions.

Courtesy of Google Maps



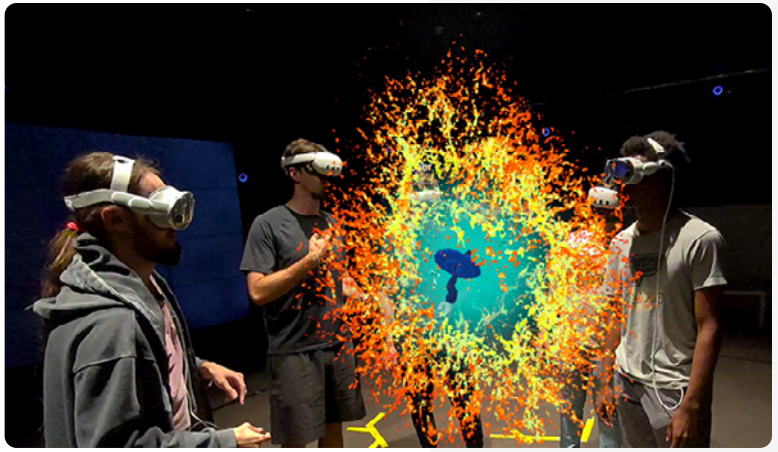
2. XR:
Immersion that demands stability

Extended reality experiences, such as virtual reality (VR) and augmented reality (AR), are unbeatable in cases where spatial understanding is important, such as hands-on training and complex walk-throughs. On the other hand, they're also the least forgiving of performance lags. Most stand-alone headsets use mobile-grade chipsets, albeit relatively high-end ones, but they require high stable frame rates to avoid user discomfort and motion sickness. As a general rule, the experience should render at the device's native refresh rate, which, in the case of VR headsets, tends to be 90 or 120Hz, equivalent to the same in FPS. Consider 72 FPS the bare minimum for comfortable VR headset experiences.

AR requirements vary greatly depending on the target device. AR inherits much of the same comfort demands as VR, so it's important to aim for stable frame rates of at least 72 FPS. On phones or tablets, AR incorporates camera feeds and simultaneous localization and mapping (SLAM) to the render loop, which, on modern devices, means aiming for 60 FPS, with a strict minimum of 30 FPS. Outdoor lighting and thermal throttling can still be an issue, however, so plan for adaptiveness with dynamic resolution and readily accessible feature toggles.

WHAT TO AIM FOR →

Use prebaked lighting in most cases, multiple LODs and simplified shaders, especially if you need to accommodate low-end XR headsets or mobile AR experiences.



Courtesy of CollabXR

3. DESKTOP:
High performance for demanding experiences

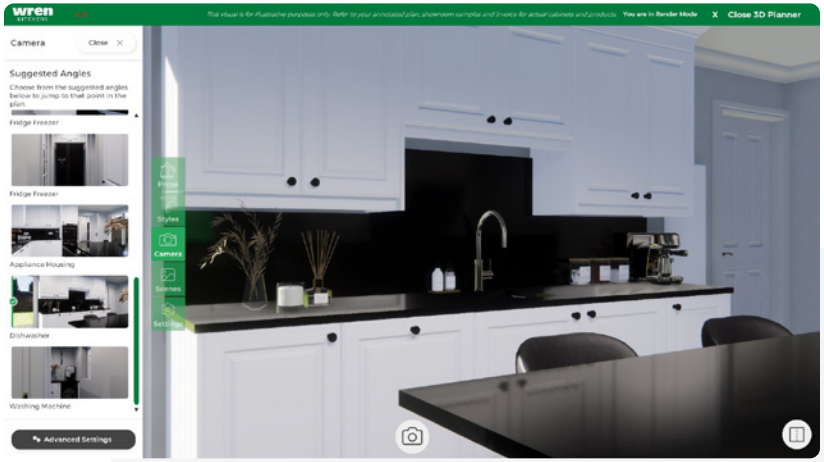
Desktops are ideal for detailed analysis or multimodal reviews, but they also serve a wide range of scenarios that don't necessarily need a flagship GPU. For example, day-to-day design approvals, quality checks, safety walk-throughs or manufacturing planning benefit most from responsive interaction and clear inspection tooling than from maximum fidelity, so target devices will often only be equipped with midrange, discrete GPUs or newer integrated graphics for lightweight viewing. In these cases, prioritize deterministic review tools, such as sectioning (cutting planes to reveal internals), measuring (snap-accurate dimensions) and exploding (staged assembly views with part callouts).

For heavier internal workloads, such as rendering massive assemblies and complex simulations, dedicated workstation-grade GPUs or remote GPU resources are the norm. However, not every end user needs such a costly device. High-end local hardware remains unrivaled for teams that author highly complex content, but for consumers, streamed sessions from a GPU server are far more cost-effective.

This also avoids the "but it runs on my RTX" surprises and reduces support load across mixed fleets of devices. However, network bandwidth can become a bottleneck, so it's important to incorporate local caching so that reviews don't stall if the network can't keep up.

WHAT TO AIM FOR →

Default to the highest stable tier, ensuring consistent frame rates on the target devices. Use deterministic review tools and on-demand streaming, and prioritize input responsiveness.



Courtesy of Wren Kitchens

Courtesy of Icon Group

4. EMBEDDED HMI:
Reliability and instant clarity

HMI brings 3D into the product itself. Think in-vehicle displays, industrial control panels and medical devices, where uptime and predictability are far more important than photorealism. In these use cases, operators often have to make time-critical decisions, while most embedded targets run on constrained hardware with long service life cycles. Even where premium hardware such as dedicated GPUs is available, it's rarely refreshed on consumer timelines; embedded devices on assembly lines and vehicles need to stay stable for years, so software has to do the job of managing tight hardware budgets.

Common use cases include setup, guided maintenance and training on assembly lines or visualizations of arm kinematics, reach envelopes and no-go zones in robotics systems. For automotive use cases, interactive 3D might be used in advanced driver-assistance systems. In these cases, legibility and motion smoothness are nonnegotiable. This is also true of many medical and lab devices, where the bar is also heightened by regulatory and safety demands.

WHAT TO AIM FOR →

Simplified geometry and prebaked lighting significantly reduce performance demands and loading times, while long-term support from vendors ensures lasting support for embedded devices.

Operate and scale your deployments for lasting success

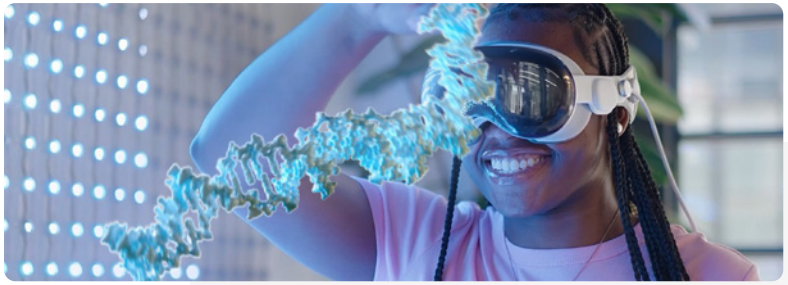
Teams often stumble by designing for the highest-end device, then accommodating lower-end devices by copying and pasting UI elements across interfaces and front-loading performance-hungry assets. However, with the right end-to-end platform, it becomes much easier to avoid these pitfalls. For example, using automatic scaling and streaming heavier assets on demand, experiences can degrade gracefully rather than failing abruptly. This ensures business continuity by providing some predictability for refreshes.

Unity's profiling tools are built to specifically accommodate these challenges: "We monitor things like asset-loading times, startup times and so forth," Linn said. "These are especially important in HMI deployments, where you need quality real-time feedback on how the model behaves on the actual device."

Initial deployment is just the beginning, especially given the dynamic nature of projects such as real-time digital twinning of production lines or shop floors. Operating successfully at scale requires a predictable way to deliver content without version drift, telemetry to see what's working (and what isn't), and governance that lets you move quickly — without breaking things.

"As for long-term rollout strategies, I think it's much the same as with any software, where you scale in waves and target the end points and users who can give you the most valuable insights on usage to identify any bugs or missing features early in the process, before you roll out to the next user or device group," Linn said. "Eventually, you'll reach full general access, by which time you'll have minimized the bugs and maximized the quality of the user experience."

That means paying close attention to update cycles, device telemetry, governance and long-term support. "Track things like session length and adoption rates, and seek direct end-user feedback where possible on things like the quality of the experience," Linn recommended.



Courtesy of Random42

Content delivery and rollout

Phased rollouts tend to work best, especially when you're targeting hundreds or even thousands of end users. With large-scale projects, such as customer-facing experiences or those spanning multiple factories and facilities, start with a single site or target user group, monitor for issues, and scale from there. This mitigates the effects of unforeseen problems and makes support much more manageable. You might also use remote configuration to ship the application or experience to everyone, but only turn specific features on for selected locations or user groups. Plan the network side, too, by precaching larger models using a suitable CDN, and schedule any major updates or downloads off-peak to prevent the first day becoming a bottleneck.

Device data that matters

You won't be able to tell if an experience is delivering the outcomes you're hoping for if you're not collecting the right telemetry data. To answer the critical questions of who's using it or whether it's working as intended, you'll need to track adoption and engagement metrics. For example, tracking session lengths and how often people come back gives you a broad idea of the project's success. However, deeper metrics, such as those spanning quality and performance, are important for continuous improvement and integration. For specific workflows, such as training simulations, define simple workflow steps to show the percentage of users starting a training step compared with how many finish it. Finally, break things down by device type to see if any group needs extra attention.

Courtesy of Vodafone Deutschland

Courtesy of Visionaries 777

Governance without complexity

Governance involves setting clear rules for who manages updates and how rollouts are organized to mitigate disruption and risk. Ideally, each release should have one named approver and a backup, with all decisions being logged in a shared place. You'll also need checks before anything goes live, but try not to overcomplicate things. For example, performance checks on target devices and sign-off from product and operations teams is usually enough, with the exception of use cases involving regulated or sensitive data, which will also require compliance and security reviews. During the release window, including the release of any updates, make sure to have a support owner on point with clear escalation paths so issues can be resolved quickly.

Long-term and community support

Hardware, drivers and operations never sit still, so always plan for long-term stability and adapt your deployment strategy to accommodate new platforms and updates as and when they're released. Working with specialized industry vendors that offer long-term support commitments also ensures you can continue supporting older devices, which is especially important in the case of high-end industrial systems equipped with embedded HMI that can't easily be upgraded. Many vendors offer open-source tools and assets as well, in which case you can treat the surrounding community as part of your support system by reusing proven assets, tracking release and update notes, and maintaining an internal knowledge base of best-in-class drivers, tools and device and use case-specific guidance.

Getting started: Your step-by-step deployment plan

Deploying interactive 3D experiences where your stakeholders and customers are allows you to deliver another dimension of information on any device. Whether you're hosting a cutting-edge training simulation, collaborating on design across desktop and mobile, or building customer experiences for web browsers, choosing the right platform ensures you get optimized results for any target.

Here's a quick recap of what it takes to get there:

1. Map stakeholders to devices and user contexts.
2. Define performance benchmarks per platform.
3. Pilot deployment across a limited number of platforms to surface issues early.
4. Set up content delivery and telemetry.
5. Establish governance roles for updates and support.

Are you ready to connect, create and deploy immersive experiences for your industry? Unity Industry provides a comprehensive suite of tools for training and guidance, 3D design collaboration, CX, XR, and HMI backed by three-year long-term support and a massive developer community.

GET STARTED →

About Unity

Unity [NYSE: U] offers a suite of tools to develop, deploy, and grow games and interactive experiences across all major platforms from mobile, PC, and console, to extended reality. For more information, visit unity.com

LEARN MORE →