

OpenUSD Proposals Across Domains: Challenges and Methodology

Pixar, NVIDIA

/* ACADEMY SOFTWARE FOUNDATION

open
Source
days^{'26}

The Evolution of OpenUSD

2011-2015

Invention of USD



Pixar's Presto

Blue Umbrella POC

SIGGRAPH Debut

2016-2018

Apple Brings to World



Initial open-source release of OpenUSD

Apple releases USDZ for consumer/enterprise

2019-2021

NVIDIA Doubles-Down

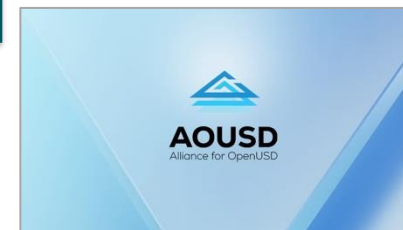


First collaboration between Adobe, Apple & NVIDIA (UsdPhysics for robotics)

Early exploration across non-M&E domains

2022-2024

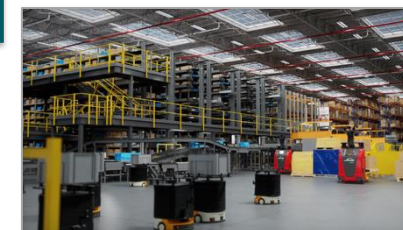
Cross-Industry Adoption



Alliance for OpenUSD (AOUSD) established

2025-Now

Setting Standards



USD Core Spec 1.0

- Blueprint for USD utilization
- Scales beyond a single implementation
- Cuts across technologies and industries

The Challenge

- As new domains join, they bring their own vocabulary, their own assumptions, and their own sense of what "the problem" is.
- Without a shared vocabulary, communities debate solutions to problems before reaching a shared understanding.
- Proposals stall and contributors disengage.

"How do I refer to this 3D object, so the tool knows exactly what I mean?"

Core Spec

Does identifier mean the path token or other metadata?

Physics

Is a rigid body a prim or an applied API schema?

Materials

Is 'roughness' same in PreviewSurface and OpenPBR?

Geometry

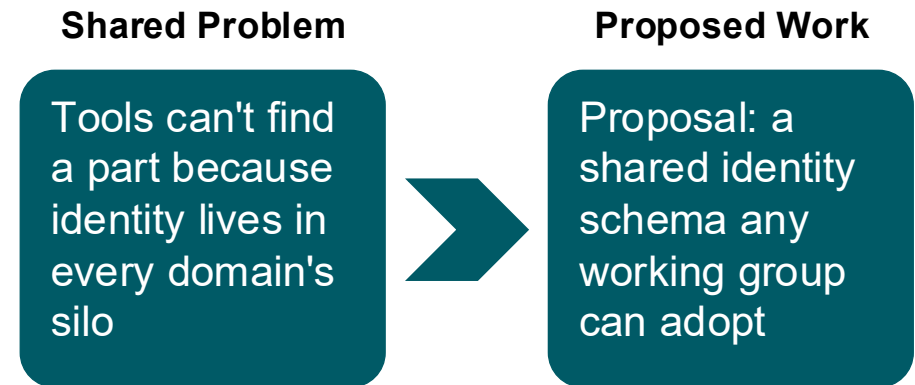
Is a Gaussian splat geometry, a volume, or a new domain?

Digital Twins

Where does my part number live so any tool can find it?

The Methodology

- Identify the root cause before jumping to a solution.
- The proposals process works when it starts with a communal problem statement.
- A user-domain, USD-agnostic, statement that anyone *from any domain* can read and recognize as their concern.



Problem Statement

1. **Untangle terminology:** The same concept often has different names across industries.
2. **Step back from USD:** If your problem statement requires knowing what USD is, it's not neutral enough yet.
3. **State the concern in shared terms:** Describe what goes wrong in the world without this, not what the solution should look like.

Presupposes Solution

We need USD prim name grammar to support asset part numbers.

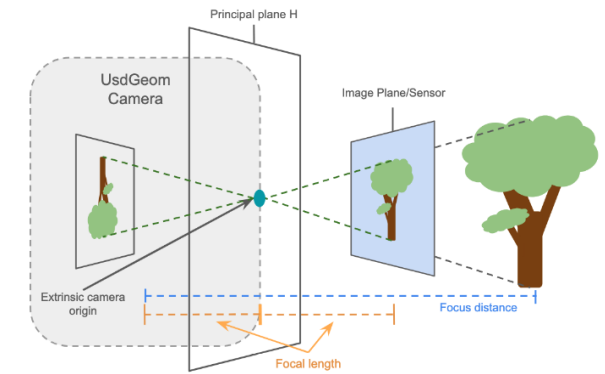
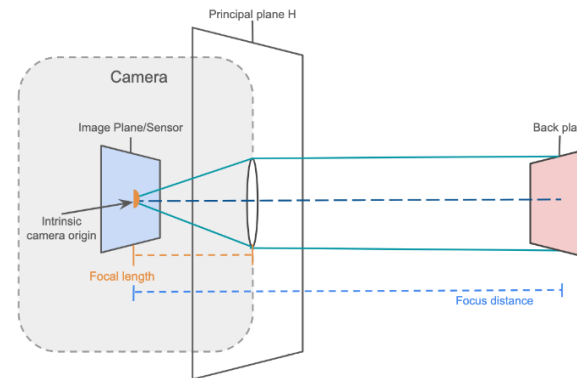
Rewrite

Neutral

Tools in different domains can't reliably refer to the same object.

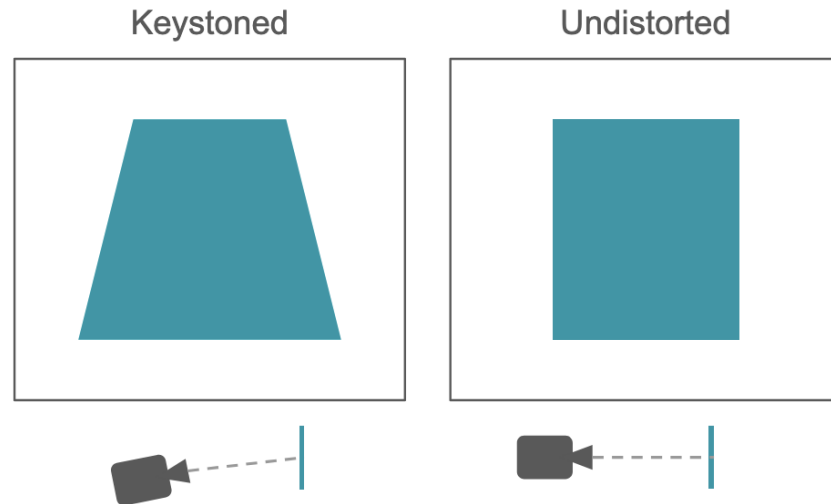
How to Read a Proposal

- Image Planes Proposal as Exemplar
 - Background
 - Interchange
 - Optical System
 - Proposal
 - Why Separate Schemas?
 - Properties
 - DCC Interchange
 - Engineering Work
 - Out of Scope
 - Alternate Implementations

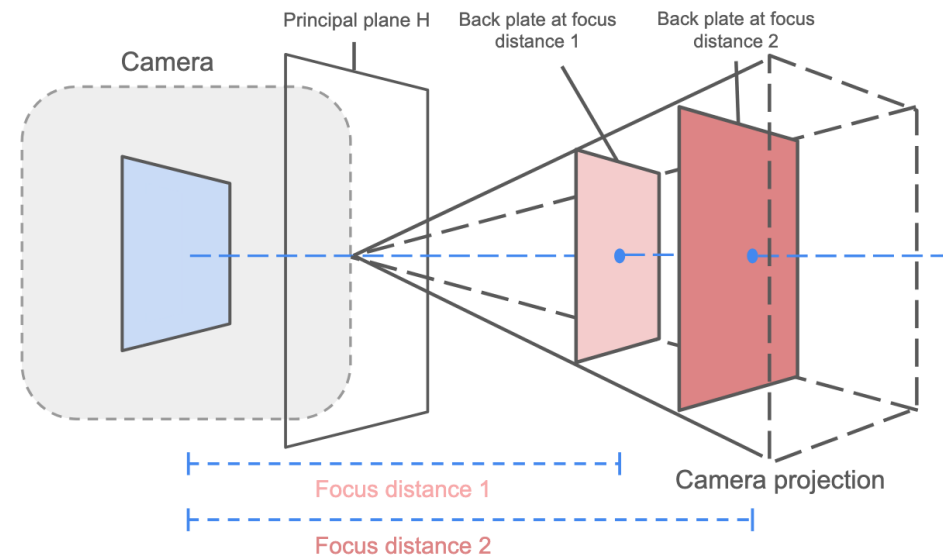


How to Write a Statement

- Image Planes Proposal as Exemplar



When the camera is not perpendicular to the subject, it can cause a perspective distortion where parallel lines appear to converge.

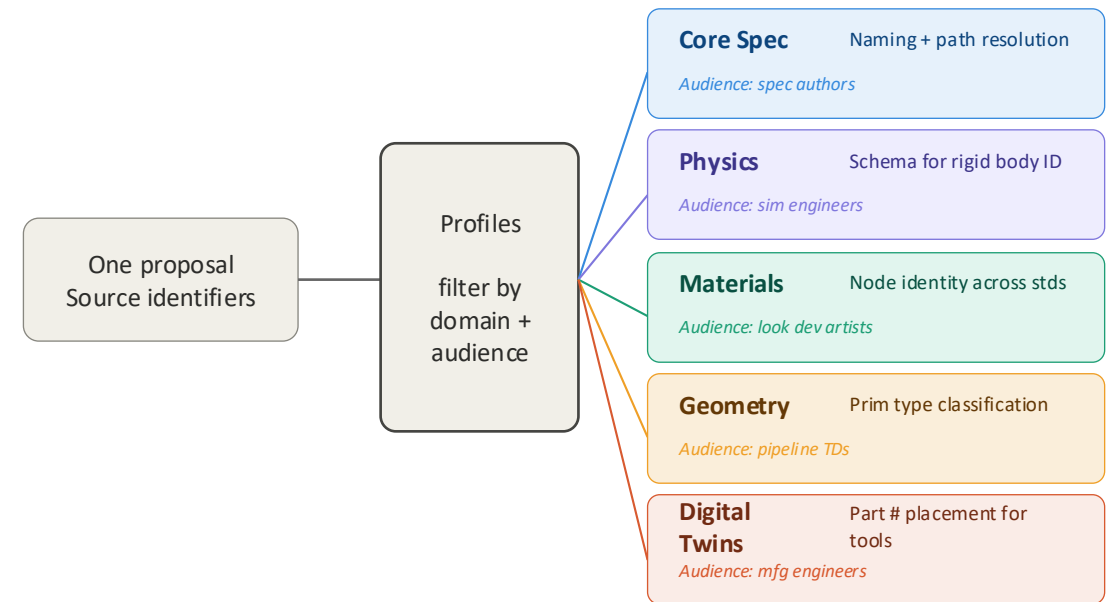


How to Write a Statement

- Image Planes Proposal – Separation of Concerns
- Out of Scope
 - Lens Distortion
 - Projection Painting

Profiles: What lanes are relevant to you?

- Not every proposal is relevant to every lane.
- Profiles bucket content capabilities and discussions by lanes such as domain, industry, and application.
- Profiles organize lanes— with core spec as the domain-agnostic lane.
- Proposals may span multiple lanes.



Same proposal. Each community reads only what's relevant to them.

Cross-Cutting Concerns

Source Identifiers

Correct: How does any tool reliably refer to the same object across domains?

Incorrect: We need USD prim name grammar to support asset part numbers.

Authorship

Correct: Who created this, who changed it, and does that travel with the asset?

Incorrect: We need USD to track provenance across layer stacks.

IP Protection

Correct: How do rights and restrictions move with an asset across tool boundaries?

Incorrect: We need USD to support access controls, and fall back to a useful placeholder

Metadata

Common Fields

Domain-Driven Examples

Driven by: M&E
NOT M&E-ONLY

LOD

M&E drove it — the schema reserves lanes for other domains

```
class "LODRootAPI" (inherits = APISchemaBase) {
  int lod:default:index = 0  # fallback if no
                              # heuristic understood
  rel lod:heuristics
}
class "LODHeuristic" {
  uniform token lod:domain
  # imaging | physics | audio
}
```

imaging = M&E's lane, filled now. **physics / audio** = lanes reserved for other domains to fill in **later iterations** — no rewrite.

Image Planes & Back Plates

M&E-shaped, not M&E-only — and landing in iterations

```
class "BackPlateAPI" (apiSchemaType = "multipleApply") {
  asset image
  asset alpha:image
  asset depth:image
  token plateVisibility = "solo"
  # all | solo | mute
}
```

Thin-lens optics, on-set match-move = **M&E fingerprint**. The same card also feeds **AV / synthetic-data** capture.

✓ **BackPlateAPI** — landed in dev

... **ImagePlaneAPI** — in review

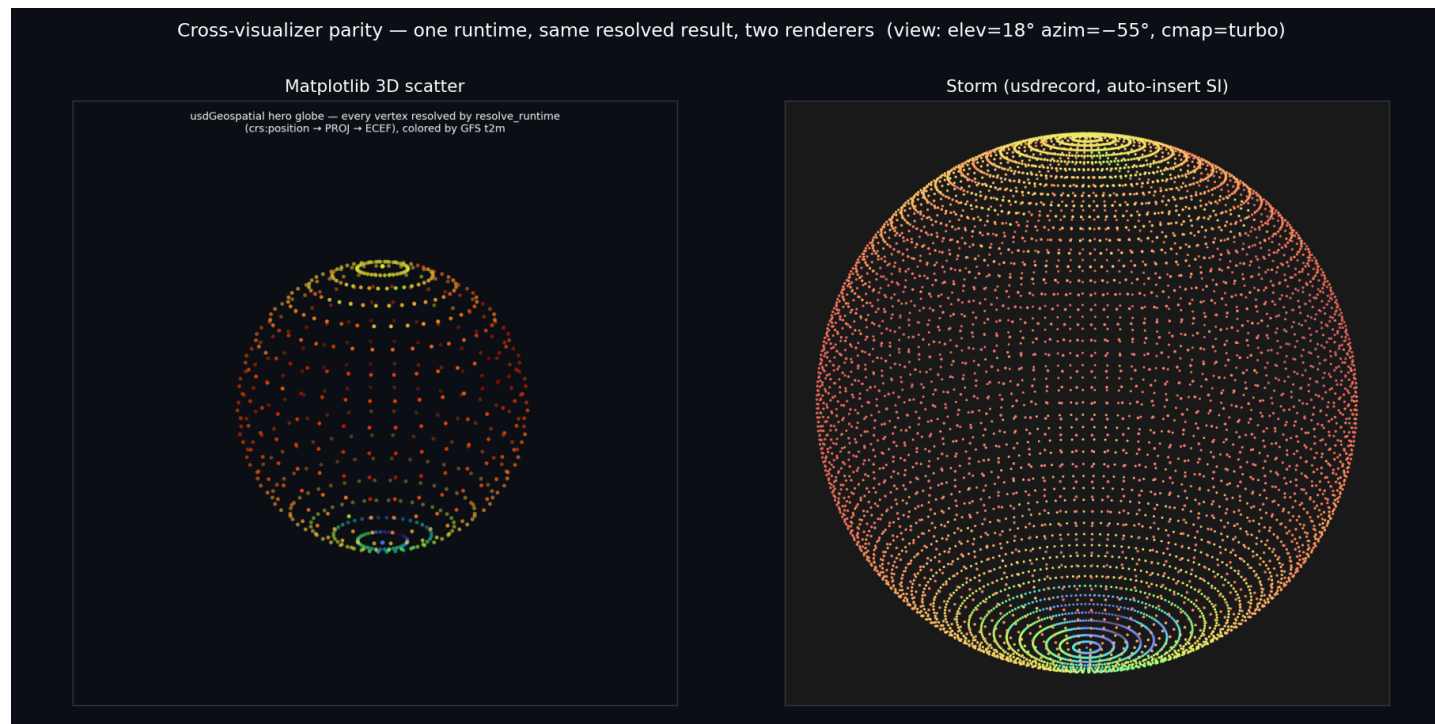
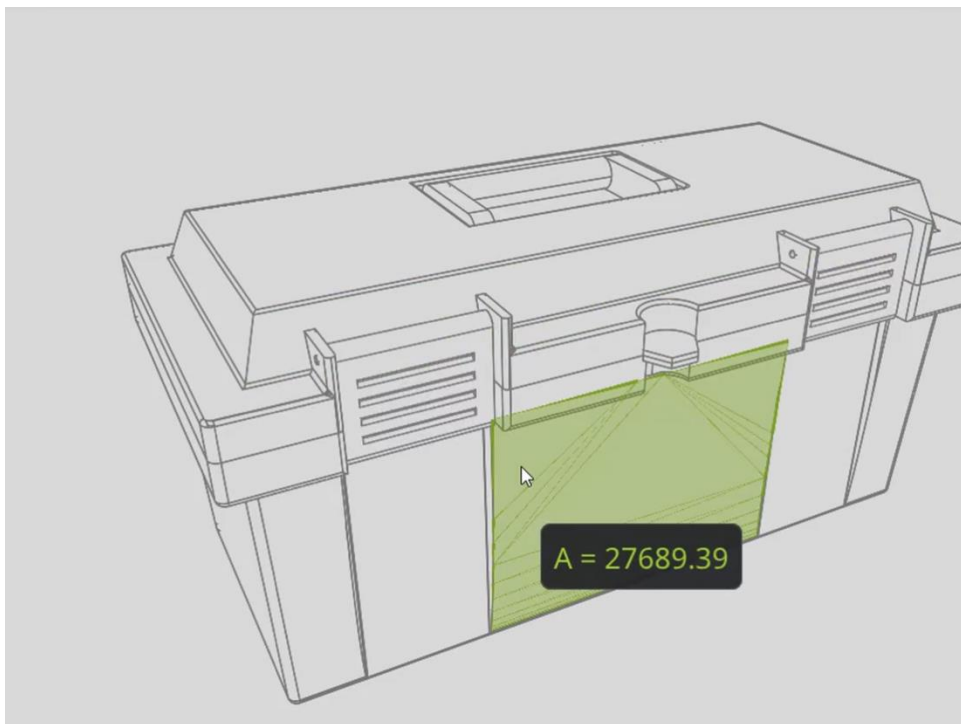
Proposals arrive **domain-driven**. It's fine for the most-interested domain to drive and scope **for now**; others engage and refine in **later iterations** of whatever lands in the meantime — the method, not a compromise.

Domain-Driven Examples

Driven by:
Industrial
NOT INDUSTRIAL-ONLY



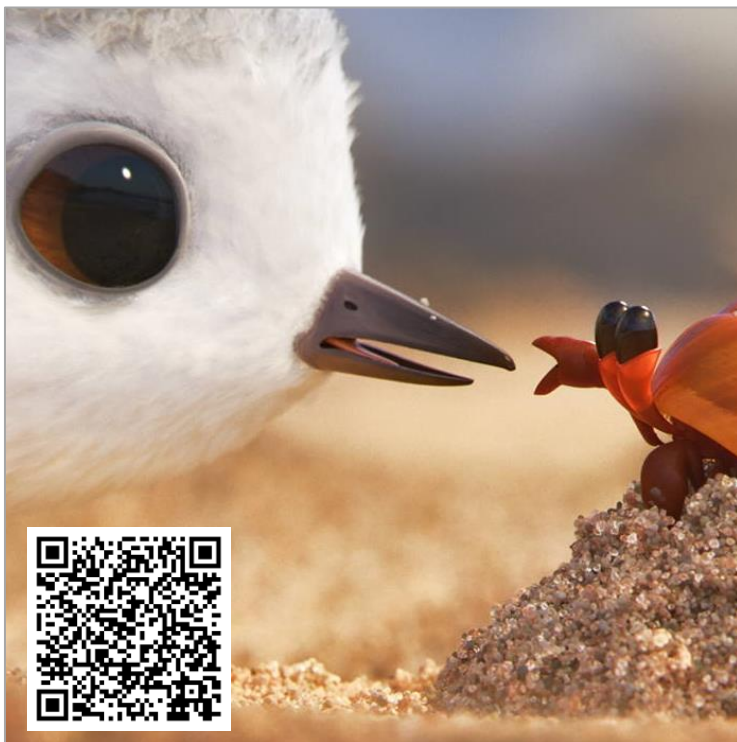
- BREPs
- Geospatial





Next Steps

Next Steps



Submit and Review Proposal PRs



Test Proposal Implementations

/* ACADEMY SOFTWARE FOUNDATION

open
Source
days^{'26}