

Asset Information Beyond the Application

Paper 2 of 4 — what the thing is, and what we know about it

A civil engineer recognizes a pump.

It is in the work area, correctly positioned, plainly relevant to what they are doing. They can see it is a pump. What they cannot see is which pump.

But what is it?

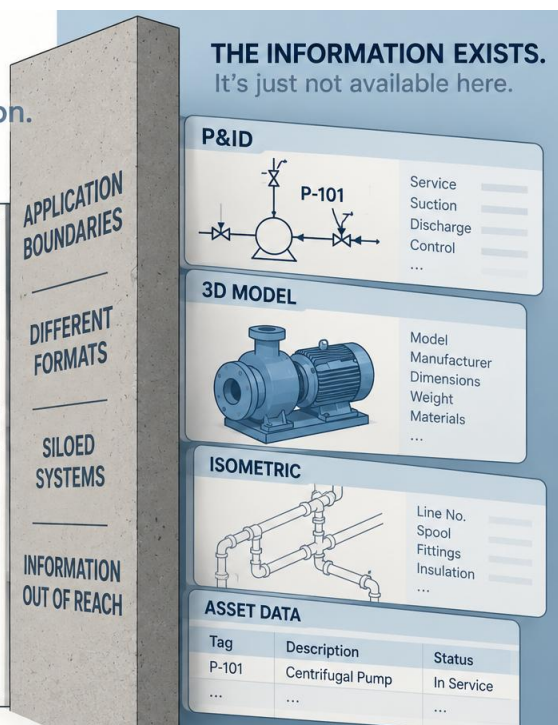
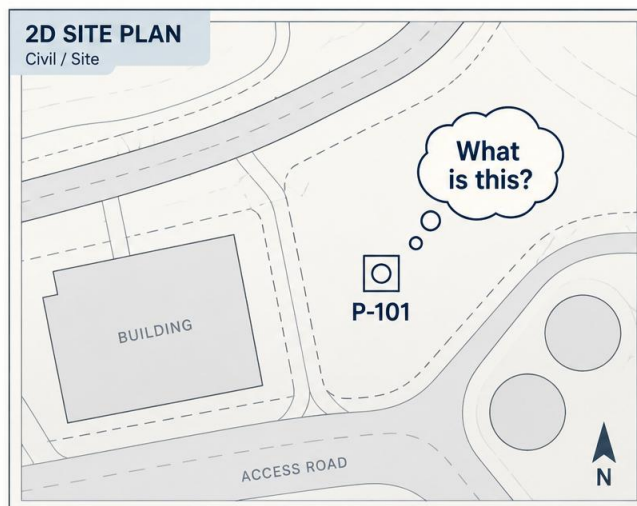
The foundation depends on the answer. A pump has a weight, a model, a mounting arrangement. Someone, somewhere, knows all of it. The engineer simply cannot reach it from the drawing where the question is being asked.

So they phone a friend. Or check a spreadsheet that may no longer be current. Or measure the geometry and take a guess.

The Problem

You can see the asset. But not the information.

The civil engineer works in a 2D plan with only a symbol and a tag. The information exists — it's just on the other side of the wall.



Disconnected. Incomplete. Hard to trust.

Different representations. Different information. No common understanding.

This

happens on almost every project, so routinely that most of us stopped noticing it was a problem.

The information exists. It just is not available where the work is happening. The Thing Is Not the Picture

A pump can appear as a symbol on one drawing, as solid geometry in a model, and as an installation detail on a third. None of those is the pump. Each is a picture of it, drawn for a particular purpose.

Three things are worth keeping separate:

The picture — how we need to see it for the job at hand.

The identity — what makes every picture refer to the same pump.

The information — what we know about it.

A thing that matters enough to be identified and described is an asset. A pump is an asset. So is a valve, a manhole cover, a security door, a lighting fixture. Geometry does not make something an asset. The information associated with it does.

Applications are excellent at pictures. They are good at information too, inside their own structures. What tends to go missing is the identity — the thread that lets one picture be recognized as the same thing another picture shows.

Lose the thread and the pictures survive while everything known about them stays behind.

This Is Not Really a CAD Problem

Modern projects are collections of specialized systems. Architecture, plant design, civil, structural, instrumentation, procurement, construction, operations. Each has good reasons for working the way it does.

The physical thing does not share those boundaries.

A security door does not become a different door because one person sees it in Revit and another sees the surrounding site in Civil 3D. A valve does not become a different valve because its representation was translated into another format.

But the information can become unreachable. So it gets entered again during procurement. Recreated for operations. Carried by hand from one discipline to another. Or quietly lost to the next person, who does not know where to look.

The fix does not have to be one database holding everything. It can be simpler: keep the identity of the thing, and make what is known about it reachable where the work is happening.

What AssetLink Is

AssetLink is a data service that works inside AutoCAD. It lets a thing in a drawing carry an identity, and lets information travel with that identity rather than with the graphics.

That is the whole of it. It does not draw, model, translate, or store the company's records. It answers one question, in the place the question gets asked: what is this, and what do we know about it?

What that makes possible today:

- A 69-file AVEVA set read completely — no seat, no vendor runtime, nothing installed from the original vendor.
- AVEVA engineering data readable by a Civil 3D, Plant 3D or plain AutoCAD user: select the object and Inspect shows what it is — the data itself, not converted shapes.
- Things found wherever they actually live in a drawing — placed directly, nested inside blocks, or several levels down in external references.
- It runs disconnected. The drawing carries what it needs, and no connection is required to work.

That last point matters more than it sounds. Identity has to be established where the work is done, by the person doing it, whether or not anything downstream ever asks for it.

Four capabilities, in plain terms:

Identity — the thing gets a name of its own, and keeps it.

Attribution — what is known about it travels with that name.

Presentation — the user sees what is useful to them, not a data dump.

Mobility — none of it depends on the drawing it started in.



A

thing may be buried three levels deep in external references. The user should not have to understand how the drawing was assembled to understand what they are looking at. That is the job.

Four capabilities: identity, attribution, presentation, mobility. Where Most Identity Comes From

Almost nothing gets its identity by hand.

Getting foreign content into AutoCAD is a separate job and a separate tool. Exchange interprets what another application authored — an AVEVA set, an AutoPLANT project, a CADWorx model, an IFC delivery — and reconstructs it as something a person can work with: AutoCAD solids, correctly positioned, at the right size.

That is the visible half. The other half is that the engineering data comes across too, and it has to live somewhere.

It lives in AssetLink. The geometry becomes AutoCAD entities. Everything that is not geometry — the properties, the classifications, the references back to where it came from — is held in the identity layer attached to it.

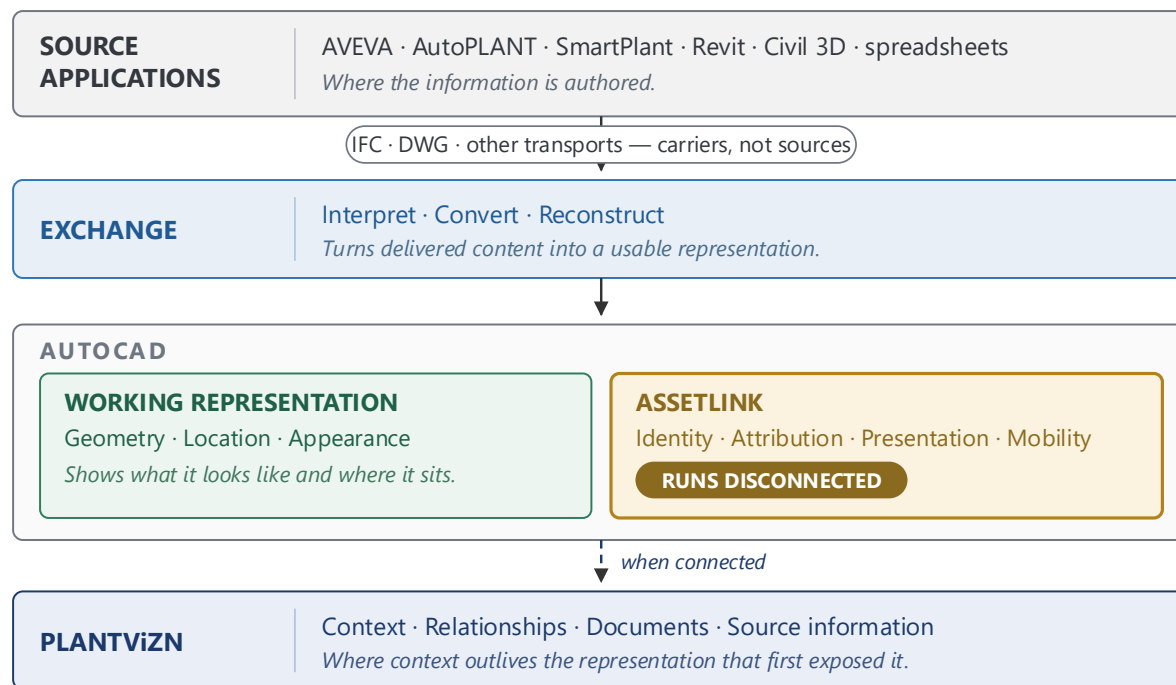
The separation between picture, identity and information is not an abstract distinction. It is the question that produced AssetLink: convert an AutoPLANT model into plain AutoCAD geometry and the custom objects that carried the data are gone. The shapes survive. Everything else needs somewhere to live.

Which also makes Exchange AssetLink's largest consumer by a wide margin. For every asset somebody tags by hand, an import establishes thousands.

It also settles what running disconnected actually means. A converted drawing is not a picture of an AVEVA model with the data left behind on somebody's server. **The data is in the file.** Open it on a machine with no connection and none of the original software, and the answers are still there.

Information Beyond the Application

One thing. Many applications. Identity established where the work happens



to the Pump

Source applications, transports, and what arrives in AutoCAD carrying its data. The engineer still cannot finish the foundation from the drawing alone.

But one thing has changed. They no longer have to work out what the pump is by looking at it. They can ask, and the drawing can answer: the model, the weight, the vendor reference, whatever the class says matters for a pump.

From here on it has a name. P-101.

Different Systems, Different Words

Systems are not wrong when they disagree. They just speak their own language.

What it means	One system says	Another says	A spreadsheet says
Manufacturer	Manufacturer	Manufacturer	Mfr
Model	Model	Equipment Type	Model No.
Weight	Mass	Weight	Wt.
Part number	Part Number	Catalogue No.	Part #

The useful question is not which spelling wins. It is whether these mean the same thing. When they do, the relationship can be set up once rather than rebuilt as custom software every time two sources meet.

This matters because nobody can predict what will need describing next. A security device may need an IP address. A light fitting may need a colour temperature. A manhole cover needs something else entirely.

Owners do not think in application object types. They think about the things they have to look after. If it matters to the owner, it can be an asset.

What This Is Not

This does not require replacing the applications people already use.

It does not require every system to adopt the same internal structure.

It does not require a schema that anticipates every future class or source.

It does not require one product to become the owner of the truth.

And AssetLink is not a replacement for Revit, Plant 3D, Civil 3D, AVEVA, SmartPlant, or an owner's operational system. Those applications can stay good at what they do.

The Larger Idea

The industry has spent decades making models more intelligent. Engineering applications built sophisticated object models. Enterprise systems built powerful repositories. Cloud platforms made information reachable across organizations.

But the real world does not follow application boundaries.

A facility contains things. The same things appear in many systems. Systems change. New applications arrive, old ones are retired, and the work carries on with information that outlived both.

The thing remains.

Perhaps the goal is not to make one application or database the master of everything. Perhaps it is simpler:

Give the thing an identity.

Make what is known about it reachable where the work is happening.

Let the applications remain good at what they do.

Exchange brings the content across. AssetLink establishes what it is, and holds what is known about it. Knowing where something is was the first question. Knowing what it is, is this one.

But an answer is not the same as a reliable answer.

The engineer can now ask the drawing and get a reply. What they cannot tell is where each part of that reply came from, how old it is, or whether the things it was assembled from agree with one another.

You can get an answer. Should you believe it?

That is the subject of the next paper.