Beyond Purdue: The Works, Narration

Narration V1.0 | 3 November 2026 | Petr Roupec, BM Global | Video V1.0, about 5 min 47 s, voice algieba

The video follows the order of the works: the laws first, then data in independent nodes, then names and aliases, then everything that results from them. New scenes: the laws, independent nodes and names. New narration: title, two copies, the data diode, the rules and the end. The other scenes keep their V0.2 narration.

#	Start	Scene	Narration
0	0:00	Title	Let me show you Beyond Purdue, step by step. First the laws, then the data, then the names, and then the shouts that move everything.
1	0:11	The laws of authority	It starts with the laws of authority. Purdue sorts devices by where they sit. We sort them by what they are allowed to do. The simpler and more predictable a level is, the more authority it holds. Protection and isolation come first, and a lock outranks every level. The operator decides. And the AI has a voice, but never a vote.
2	0:35	Data lives in independent nodes	The second idea is where the data lives. Not in one big database, but in many small, independent nodes. Each node has its own name, its own data and its own meaning. Only L zero and L one touch the process. They shout what they know, one way, into these nodes. Everything else in this video follows from that. And we build it from technology people already know.
3	1:01	Names and aliases	Naming and asset management is always the big issue. So every node has a name in the style of DNS, because people already know how to read it. The same node can have aliases: its KKS code, the vendor tag, the name on the HMI, the asset number. They all lead to the same node. And a name can be updated automatically, by a person, or by a shout.
4	1:25	The pattern: shout, never ask	The pattern itself is old and proven. It's publish and subscribe over multicast. Nobody opens a connection, nobody waits for a reply, and nobody sends an acknowledgement. Substations have run their GOOSE messages this way for years. And because nothing ever has to come back, a shout passes through a data diode unchanged.
5	1:48	1 A value changes, a shout goes out	So let's follow one value. A steam temperature changes. The adapter sends a single shout with the name, the value, the time, the quality, a sequence number and a signature. It crosses the diode onto the boiler group, and every node on that group hears it. But only the two copies of H P steam keep it.
6	2:15	2 Repeat instead of acknowledge	The sender never waits to hear back. Instead, it simply repeats itself: once straight away, and then again with longer and longer gaps. Here one of the repeats gets lost, and it doesn't matter, because the next one carries the same value. After that, a slow heartbeat tells everyone the node is still alive.
7	2:39	3 Two copies, no sync	Why two copies of everything? Because one copy is never enough. Every node lives in at least two copies, on different hardware. Both listen to the same group, so they hold the same data without ever syncing. When a host loses power, the second copy just carries on, and a new copy starts on another host by itself. Nobody had to touch it.
8	3:03	4 Ask by shout, answer by shout	Asking works the same way. The AI shouts a question to the group: which tags had events in the last twenty-four hours? Each node waits a small random moment before it answers, so the replies don't collide. And if one copy hears the other answer first, it simply stays quiet.
9	3:26	5 A new tag is born	New signals look after themselves too. When an edge adapter hears something nobody has named yet, it shouts: new tag. The registrar gives it a name and two addresses, signs the change, and shouts it to every host. Two hosts start the new node, and nobody had to configure a thing.
10	3:48	6 The data diode, the enabler	The data diode is the enabler. Light travels one way only, so data and commands can never mix. The levels hold because the diode holds. A passive diode has no power and nothing to configure, and it gets a name like any other asset. For a shout, it's just a longer piece of wire. Headquarters keeps its own copies and asks those, because no question can ever travel back.
11	4:13	7 A command is recorded	Commands follow one path only: the L two in lane, from the operator to the controller. But L three still has to know about every one of them. So at the same moment, the operator shouts through its own diode: I gave command oh four one seven. The controller doesn't repeat the command. Once it has acted, it only shouts: command oh four one seven executed. L three pairs the two shouts into one record. And if a command is given but never reported as executed, L three raises an alarm.
12	4:48	8 Released for work	This shout is about people. Maintenance is part of operation, and people work inside the plant. When a pump is isolated and locked for work, the controller shouts: one one, M B N, one two, A P, zero zero one, released for work. L three pairs it with the permit. A lock outranks every level, so nothing starts that pump while someone is inside it.
13	5:13	The rules	So those are the rules. The laws come first. Data lives in independent nodes, with names people can read. At least two copies, that never need to sync. Everything goes one way, through the diodes. Repeat instead of acknowledging. Commands keep their own lane, and L three records both the order and its execution. And protection stays hardwired.
14	5:38	End	Beyond Purdue. The laws first, then names and data, all as shouts.