
Running an experience centre securely

Why a museum, science centre or brand experience should be run like a well-governed IT estate, and how taleor does it: every device identified, every connection encrypted, every change attributed, every visitor anonymous.

White paper · October 2026 · Version 1.0

For directors, technical and IT heads, AV integrators, consultants and tender committees of museums, science centres, brand experience centres, heritage sites and attractions.

taleor

Contents

1	Executive summary	3
2	The problem: a venue is now an IT estate	3
3	What a venue actually needs	5
4	How taleor works	5
5	Secure by design	7
6	Set up by an assistant	11
7	Analytics: what a venue learns	12
8	Runs anywhere, works with what you have	13
9	Deployment models and how to evaluate	15
10	About taleor	17
	Sources and verification	17

1 Executive summary

An experience venue used to be a building with some screens in it. It is now a network: hundreds of computers, displays, projectors, loudspeakers, lights and sensors from a dozen manufacturers, each with its own firmware, an address and often a password, run by a small team that is not an IT team.

This paper argues that a venue should be run the way a well-governed IT estate is run: every device identified, every connection encrypted, every change attributed, every update verified, every visitor left anonymous. Section 3 sets out seven principles, sections 4 to 8 describe how taleor meets them, and section 9 closes with a checklist a tender committee can put to any vendor, including us.

taleor is a secure, agentic way to run an experience venue: its content, interactive applications, players and hardware, from one console. It rests on three pillars: it is secure by design; it is set up by an assistant that never applies a change without a named person's approval; and its analytics say "not measured" rather than zero. It runs as a web app, as a desktop app for Windows, macOS and Linux, and as a local server in your own building, including fully air-gapped. Its players run on Windows, macOS, Linux, Raspberry Pi and Android. It drives equipment of any brand over open standards.

2 The problem: a venue is now an IT estate

2.1 Hundreds of devices, dozens of brands, a small team

Count the computers on the floor of a contemporary museum: one behind most exhibits, plus networked displays, projectors, audio processors, lighting controllers, sensors, readers and buttons. In a venue of modest size the count reaches the low hundreds. The seed data used to develop taleor, modelled on a typical regional museum, holds 69 exhibits and 342 devices speaking 49 control protocols. That is an ordinary estate.

Each manufacturer brings its own protocol, its own management tool and its own idea of how updates work. The content changes every season. The team is usually two to five people whose jobs were not designed to include fleet management and patching.

2.2 Security exposure

Four weak points compound one another.

Default credentials. Professional AV ships with a management interface and a factory password that is rarely changed, because changing it on two hundred devices is a week of work that nobody owns. The Mirai botnet of 2016 recruited devices by trying a short list of factory passwords. Venue AV is not exempt.

Unpatched players. The computer behind an exhibit often runs the operating system it was commissioned with, because updating it means closing the exhibit. An unpatched machine on the same network as ticketing and building management is a path in. The frameworks against which public bodies in Germany, the Netherlands and Poland are measured (BSI IT-Grundschutz, the Baseline Informatiebeveiliging Overheid, and the obligations that flow from NIS2) treat timely patching and asset inventory as baseline controls.

Shared secrets. Many control systems authenticate a player with a key in a configuration file. Anyone who can read that file can impersonate the player.

Cloud dependence. A system that stops when the internet or the vendor's cloud drops has moved the single point of failure somewhere the venue cannot fix. For a public institution it is also a procurement question: who holds the data, under which law, and what happens when the contract ends.

2.3 Downtime, blind spots and lock-in

Downtime is the most visible failure and the least measured. A screen freezes at ten, a visitor mentions it at noon, the technician hears at two, and nothing recorded the four lost hours.

Most venues count visitors at the door and nowhere else. They cannot say which exhibit people stopped at, for how long, or whether a film was watched to the end. Where measurement exists it is often a separate system, too often one with cameras that identify people.

Each subsystem arrives with its own vendor and its own contract. When a vendor is acquired or drops a product, content built for a proprietary player cannot move, and hardware that only speaks to one controller becomes scrap. The question a committee should ask is not "does this vendor support our hardware?" but "if this vendor disappeared, what would we still own?"

3 What a venue actually needs

These problems have been solved before, in general-purpose IT. What has been missing is their application to the shape of an experience venue: many small computers, many brands, a small team, a public audience, and a building that must open at ten. We propose seven principles.

1. **Every device has an identity that cannot be copied.** A key generated on its own hardware that has never left it. Not a password. Not a token in a file.
2. **Every connection is encrypted and authenticated in both directions.** Anything less lets a stranger on the venue network stand up a convincing fake.
3. **Nothing runs on a player that was not verified first.** Drivers and updates are checked against a signature or a checksum on the player, not only at the source.
4. **Every change has a name and a time.** An audit trail that an administrator can shorten to a day, or edit, is not an audit trail.
5. **People get the permissions their job needs, and no more.** A facilities manager can restart a frozen screen without being able to delete it.
6. **Visitors stay anonymous by construction, not by policy.** A property of the database schema, and tested.
7. **The venue owns its system.** It can run everything in its own building, with no internet, and can move away without losing its content.

These are what a competent IT department would ask of any fleet of two hundred computers. The rest of this paper describes how taleor meets them.

4 How taleor works

4.1 The parts

The console. A web application where staff see every exhibit, player and piece of equipment, load content, set schedules and triggers, watch health and alerts, and read the analytics. It runs in any current browser, or as a desktop app for Windows, macOS and Linux.

The player agent. A small background service on the computer behind each exhibit. It holds a persistent, mutually authenticated connection to the server, plays what it is told, drives the equipment on that computer, reads its sensors, reports its health and keeps itself updated. It keeps a 4 GB local copy of its

content by default and keeps playing when the server is unreachable. Agents run on Windows, macOS, Linux, Raspberry Pi and Android. A screen without a computer pairs as a browser player with a six-digit code read off the glass.

Drivers. A signed package that tells taleor what a piece of equipment can do and how to speak to it. The console asks for a verb such as power, source or volume; the driver speaks the protocol.

The server. The database, the WebSocket hubs that carry live state, the certificate authority for each venue, media storage, the schedule engine, the alert monitor, the audit log and the analytics. It is the same software in our cloud or on a machine in your building, with or without internet.

4.2 One build, three ways to run it

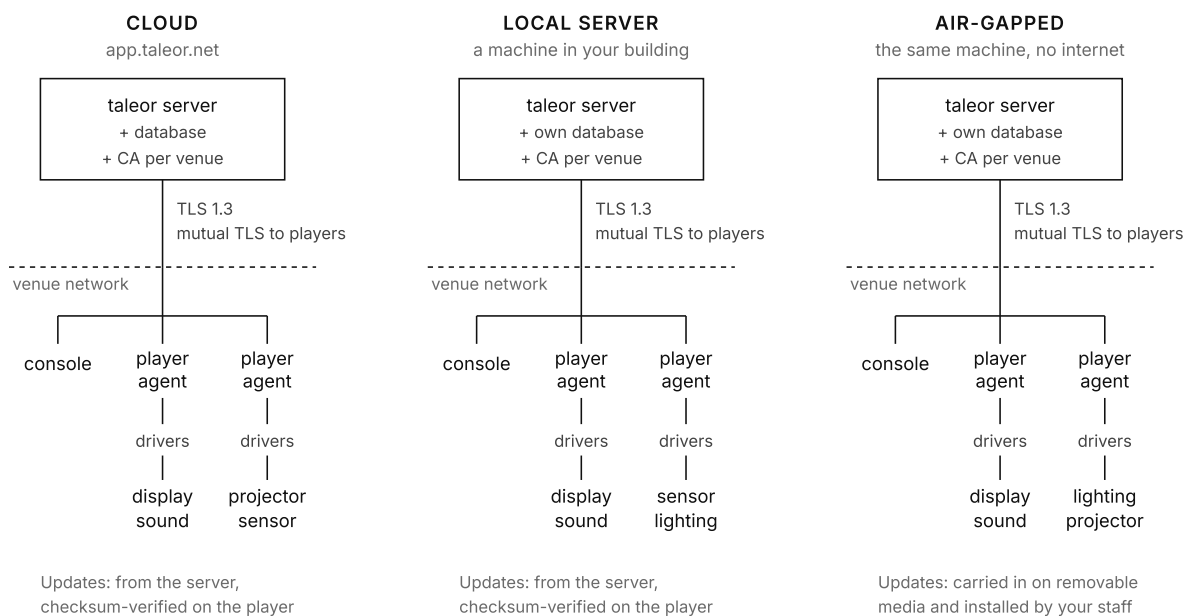


Figure 1. One build, three ways to run it. The same server, console and player agents in the cloud, on a local server in the building, or air-gapped with no internet.

In cloud mode the console and server live at app.taleor.net and the agents connect out to it; equipment on the venue network is driven by the agents inside that network. In local-server mode the same package installs on a Windows, macOS or Linux machine in the building, with its own private database and certificate authority, so console, players, media and audit log all stay inside. In air-gapped mode that server runs with no internet at all: runtime and database ship inside the package, nothing is downloaded at install, and updates are carried in on removable media by your own staff. One codebase, one console; a venue that starts in the cloud can move in-house.

5 Secure by design

5.1 Identity: a key that never leaves the machine

talor replaces the shared secret with a certificate. When a player is enrolled, it generates an asymmetric keypair on its own machine and sends only the public half, in a certificate signing request, to the venue's certificate authority, which signs and returns a certificate. The private key never travels. From then on the player proves its identity by making a signature, live, during every TLS handshake. There is nothing to copy.

One certificate authority per venue, opened from the console by an administrator. A compromise at one venue cannot mint an identity at another. The authority's private key is sealed at rest under a secret separate from the server's other secrets.

The request is not trusted. Only the public key is taken from the signing request; the subject, the validity period and every extension are set by the server, so a player cannot ask to be called by another player's name or to become a certificate authority. Player certificates are issued for client authentication only, so a stolen one cannot stand up a fake server.

There is no "issue certificate" button. A certificate whose private key the server once held would defeat the purpose.

5.2 Transport: encrypted and checked, in both directions

Every connection between a browser, a player and the console is carried over TLS 1.3 (1.2 is accepted for older clients). Player connections use mutual TLS: each side presents its certificate to the other.

A valid certificate is necessary but not sufficient. On every connection the server also checks, against the database rather than a cached list, that the certificate is not revoked or expired, that the device it names still exists, and that the device belongs to the venue the certificate was issued for.

Certificates last 398 days, the CA/Browser Forum ceiling for publicly trusted certificates, and renew in their final 30 days over the player's own mutual-TLS connection, so a fleet rotates its credentials without a site visit. Issuing a new certificate revokes the previous one.

Turning mutual TLS on is staged: a rotating mode accepts both a certificate and the older credential while the fleet moves over gallery by gallery, then a required mode once every endpoint is certified. The server refuses that switch while any

connecting device would be locked out, because the alternative is a dark museum. Setting the mode back to off rolls back at once.

5.3 Integrity: nothing runs that was not verified

Drivers are signed. A driver is a package with a manifest and a detached Ed25519 signature over the package digest. A production server refuses an unsigned package by default, because a driver is code that opens sockets and talks to hardware. Integrators and manufacturers can write and sign their own with the driver kit.

Player updates are checksum-verified. The server publishes a version with its SHA-256 checksum. The player verifies the checksum before anything else, extracts the update to a staging directory, swaps it in and restarts. A failed verification is a refused update.

Updates respect the building. An update installs only outside opening hours. A new version that fails its health check within ten minutes is rolled back and blocked from being retried. A venue may opt a player out; an opted-out or lagging player is marked "not current" in the console, so it is never silently forgotten.

5.4 Access and roles

taleor has five assignable roles.

Role	What it is for
Network admin	Everything, across every venue the organisation runs
Estate admin	Every venue-wide permission, across the venues in an estate
Venue admin	Every venue-wide permission, in one venue
Content manager	Content, reels, schedules and templates
Facilities manager	Operating equipment: restart, power, source, volume

The roles are deliberately coarse, because a permission model that nobody can explain is one that nobody applies. Every route checks both the role and the venue, so a person with rights in one venue sees nothing of another. Sign-in is per person, never shared; a new account must change its password on first use.

5.5 The audit trail

Every write lands in the audit log with the person, the action, the time and the network address it came from. This includes actions taken by the assistant, which are recorded as applied by the assistant on behalf of a named person. The log can be filtered in the console and exported as CSV with the same filters, so an auditor can take it away.

Two properties protect the log from the people with the most power over it. The audit class has an enforced retention floor of 180 days: a venue may keep its log longer, never shorter. And when a staff member exercises their right to erasure, the account becomes a tombstone: identity redacted, network addresses dropped, actions kept, so the log still says that someone, since erased, published a scene on the third. An administrator cannot erase their own account.

5.6 Visitor privacy, by construction

A venue's visitors are members of the public, many of them children. The right design is one in which the system is unable to record who they are.

The two tables that hold visitor behaviour, presence events and interaction events, carry a venue, a device, a zone, a state or event word, and a timestamp. They carry no identifier of any kind for the person: no network address, no face, no cookie, no session. This is a property of the schema, not a promise in a policy; a test asserts it against the live table definitions and fails the build if anyone adds such a column.

Behind that sits a record of processing that lives in code and is reviewed in the same change as the code that creates the data: purpose, lawful basis, degree of identification, recipients and retention period for each class. A nightly sweep deletes rows past their retention, in chunks so that it never stalls an open venue, and writes a row for every run, including failures. That record is the evidence an auditor asks for first.

Class of data	Default retention	Floor
Presence at an exhibit	400 days	1 day
Touches and choices	400 days	1 day
What played, and when	1825 days	365 days
Player logs	30 days	1 day
Who did what (audit)	730 days	180 days

The controls are built to the technical requirements of ISO/IEC 27701, the privacy extension to ISO/IEC 27001. We are careful with this claim: a product cannot be certified to 27701; an organisation can, after an audit. taleor implements the controls the standard makes technical and produces the evidence an auditor needs; the institution still has to appoint a privacy lead, carry out its impact assessment, hold its processor contracts and put notices in the galleries.

5.7 Emergency

One control places a hold on the whole venue, one floor, one zone or chosen players. The message takes the full screen, a notice runs along the bottom, and exhibit sound goes quiet. A player that reconnects during a broadcast still receives it. The hold can be started and ended from a staff handheld, and it is audited like any other action. We say "exhibit sound" deliberately: taleor silences what it controls, and it is not a public address system.

5.8 Isolation and air-gap

The local server package installs its own private PostgreSQL cluster, listening only on the loopback interface and on a non-standard port, so it never meets a database the machine already has. Nobody types the database password: a random one is generated at install and lives only in a configuration file readable by the system and its administrators. No container runtime is required. Air-gapped mode is the same package with no internet (section 4.2).

Equipment on the venue network is always driven from inside that network, by the player agents or the local server. The cloud console never needs a route into your building; the agents connect out.

5.9 Failure modes that fail safe

- A player that loses the server keeps playing its last good configuration from its local cache, through reboots, until the server returns.
- A player that drops off raises an alert naming it, where it is, what it should be showing and since when. The alert clears itself when the player returns; a repeated fault updates the existing alert.
- The server backs up its database on a schedule, with a rolling history and restore from any point. Anything deleted from the content library is recoverable for 30 days.
- The server refuses to lock a venue into mutual TLS while any device would be left outside, refuses a stale assistant plan and refuses an unsigned driver. Between guessing and stopping, taleor stops and says why.

5.10 How we check our own work

Before launch, taleor's console, player and website were put through an application security review: a written threat model, proof-of-concept exploits for each suspected weakness, dependency and static-analysis scanners, and a re-scan after fixes. Findings carry a severity, an owner and a verified fix; a gate is not passed while a blocking finding is open. Ask any vendor for the same.

6 Set up by an assistant

6.1 What it does

Setting up an interactive exhibit has always been specialist work that one person in the venue knows how to do. taleor includes an assistant, called Oli, in the console. You type what you want:

When someone comes within a metre of the lobby screen, switch to the welcome film, and back after thirty seconds of nobody.

The assistant reads the venue (exhibits, players, reels, declared inputs) and proposes a plan with the exact changes listed:

Reads lobby screen: presence sensor declared, 2 reels.
Proposes: on distance 100 cm or less, play "welcome film", return after 30 s.
Waiting for your review.

You read it. You press Apply, or Discard. When something is missing, the assistant says what: no reel called "welcome film", or no presence sensor on that player. It asks one question at a time when a request is ambiguous, and it never guesses hardware that is not there.

The assistant proposes triggers, reel assignments and the enabling of a trigger, and applies them once you approve. Every control remains available by hand.

6.2 The permission model

The assistant is not a privileged user. It has no permissions of its own.

- **Reading.** Its tools for looking at the venue are read-only and scoped to the venue the person is working in.
- **Proposing.** Its plan tools only record a proposed step. Nothing executes. This is tested.

- **Applying.** On Apply, the stored steps run through the same service functions, validation and permission checks as if the person had made each change by hand, so a person who may not change a thing cannot have the assistant change it for them. The audit log records the result as applied by the assistant for that named person.
- **Staleness.** A plan that has gone stale, because a reel was deleted between proposal and approval, is refused rather than guessed at.
- **Containment.** The assistant is off and hidden when not configured. It is metered, with a hard monthly cap for the whole server. Errors from the underlying service are never shown verbatim.

6.3 Why review before apply

A system that changes what plays on two hundred screens must be able to answer "who changed this, and did a person approve it?" Put that question to any product that advertises automated setup.

7 Analytics: what a venue learns

7.1 Built from what the players actually report

taleor's analytics, called Insights in the console, are built from what the players and sensors report, not from a separate counting system: the sensor that starts a film is the sensor that counts who stopped and for how long. Every figure is scoped to the whole venue, a floor, a zone or a player, over today, this week, this month or this year, with the window before it for comparison.

7.2 What you see

Footfall and dwell. Visitors per hour, day or month, from sensors and touch. Average dwell for engaged visits. The busiest zones and hours, as a zone-by-hour heatmap.

What worked best. Exhibits ranked by visits and dwell, so a quiet corner shows up before a curator walks past it. Visitors and dwell belong to the exhibit; uptime and plays to each player.

Proof of play. Every item, every play, played through or not, with a CSV export that honours the same scope and window. Kept for five years by default, so it outlives a licence term and a sponsorship agreement.

Player health and uptime. Uptime per player over the window, from health snapshots. No snapshots in a window reads as unknown, not as healthy.

7.3 Honest numbers

A figure that cannot be measured says "not measured", never zero. A zone with no sensor does not report zero visitors; it reports that visitors were not measured there.

7.4 Privacy by construction, again

The events behind every figure are those of section 5.6, with no identifier of the visitor: no cameras, no face detection, no device tracking, no cookies.

8 Runs anywhere, works with what you have

8.1 Platforms

Component	Runs on
Console (web app)	Any current browser, at app.taleor.net or on your local server
Console (desktop app)	Windows, macOS, Linux
Local server, including air-gapped	Windows, macOS, Linux
Player agent	Windows, macOS, Linux, Raspberry Pi, Android
Browser player	Any current browser; pairs with a six-digit code

The desktop app opens the console in its own window, can play the exhibit on the same computer, and keeps the last good configuration through a reboot.

8.2 Hardware, over open standards

taleor drives equipment of any brand over the standards it already speaks.

Family	Examples	Over
Displays	Any screen connected to a computer running the taleor player, or showing the browser player	taleor player · browser player · HDMI
Projectors	Any projector that speaks PJLink; manufacturer-specific serial protocols	PJLink · ESC/VP21 · RS-232 · TCP
Sensors	Presence, distance (lidar, ultrasonic), air button, push button, RFID; sensor controllers behind any player	Serial · USB · TCP
Controllers	OSC in from a lighting desk or show controller, OSC out from a trigger, USB keys, pedals and serial boards, a handheld	OSC · USB · serial
Sound	Networked audio processors and loudspeakers, from a headset at one exhibit to a hall	REST · TCP
Lighting	Architectural and exhibit lighting	DALI

Standards, in full: PJLink · RS-232 · TCP · DALI · MQTT · OSC · REST · WebSocket · serial · USB.

Drivers are available for these families, and we test them with your equipment before you open; we say so in writing rather than claim a list of logos. If a device is not covered, the driver kit lets an integrator write one: a manifest, an adapter, a signature. Manufacturers can bring us a device and we will build its integration.

8.3 What this means for lock-in

The control layer speaks open standards, so the hardware is not tied to taleor. The server runs on your own machine with its own database, so the data is not tied to our cloud. Content is ordinary media files, and configuration exports as plain data, so a venue that leaves takes its content and its history with it. Media can also stay in the venue's own Google Drive: taleor reads only the files picked, and screens keep a copy for offline playback. We would rather be kept because the product is good than because leaving is impossible.

9 Deployment models and how to evaluate

9.1 Three models

Cloud. The fastest start: open app.taleor.net, create the venue, install the agent on each exhibit computer. Agents connect out; nothing connects in. For venues without server infrastructure, and for integrators who manage several venues.

Local server. The server package on a machine in the building, with or without internet. For institutions with their own IT, and for any venue whose procurement rules require data on premises.

Air-gapped. The local server with no internet at all; updates carried in by staff. For sites with strict network policies or no reliable connectivity, and for anyone who has decided that their galleries should not depend on a connection they do not control.

All three run the same software and the same console, and the choice can change later.

9.2 A checklist for tender committees

Put these questions to any vendor. Our answers, with the supporting section:

Question	taleor	Where
How does each player prove its identity? Is a shared key or password involved?	Per-player certificate; key generated on the machine, never copied; one authority per venue	5.1
Is the player connection encrypted and authenticated both ways? Can certificates rotate without a site visit?	TLS 1.3; mutual TLS; 398-day certificates renew over the player's own connection	5.2
Are drivers signed? Are updates verified before they install?	Ed25519-signed drivers, unsigned refused; SHA-256 checksum verified on the player	5.3
Can a bad update take an exhibit down?	Installs outside opening hours; rolled back and blocked on a failed health check	5.3
Can a technician restart a screen without being able to delete it?	Five roles; yes	5.4
Does every action carry a person, time and address? Can an administrator shorten or erase the log?	Yes; CSV export; 180-day retention floor; erasure leaves a tombstone, not a gap	5.5

Question	taleor	Where
What does the system record about visitors, and for how long?	Venue, device, zone, state, time; no identifier, enforced by a schema test; retention per class, deletion evidenced	5.6
Is there an emergency hold, and what does it reach?	Venue, floor, zone or chosen players; full screen; exhibit sound quiet	5.7
Can the whole system run on premises, without internet?	Yes; local server and air-gapped	5.8, 9.1
What happens to the exhibits when the server is unavailable?	Players keep playing from local cache, through reboots	5.9
Has the product had a security review, and can we see the findings?	Yes; threat model, proofs of concept, scanners, verified fixes	5.10
If setup is automated, does a person approve each exact change, in their name?	Yes; propose, review, apply; audited as applied for the named person	6.2
Which platforms and hardware standards?	Players on Windows, macOS, Linux, Raspberry Pi, Android and any browser; PJLink, RS-232, TCP, DALI, MQTT, OSC, REST, WebSocket, serial, USB	8
What do we own if we leave?	Content as ordinary files, configuration as plain data, the database on your machine	8.3

9.3 What to ask for in a pilot

A paper is not a product. Ask for a pilot of a few exhibits that exercises the controls: enrol a player with mutual TLS and inspect its certificate; try a second machine with a copy of its configuration and confirm that it is refused; have a facilities manager restart a screen, then try to delete it; have the assistant propose a trigger, discard it, then approve one and read the audit log; pull a network cable and watch the player keep playing; export the audit log and proof of play as CSV; ask for the record of the last retention sweep. Each takes minutes and tells a committee more than any feature list.

10 About taleor

taleor is a secure, agentic way to run an experience venue: its content, interactive applications, players and hardware, from one console, in the cloud, on a local server or fully air-gapped. It serves museums, science centres, brand experience centres, heritage sites and attractions, and the people who run and buy for them.

taleor launches in November 2026. Write to hello@taleor.net to arrange a pilot or to request the data sheet for tenders in English, German, Dutch or Polish.

Sources and verification

Every technical claim in this paper rests on a named source in taleor's code, design documents or security review, recorded in a claim register that lists each claim, the section it appears in and the file it is drawn from. Write to hello@taleor.net for the register.

Public references used in section 2, none with figures: the Mirai botnet of 2016 (widely documented, including by the US Cybersecurity and Infrastructure Security Agency); BSI IT-Grundschutz (Germany); the Baseline Informatiebeveiliging Overheid (Netherlands); Directive (EU) 2022/2555, known as NIS2; and the CA/Browser Forum baseline requirements for the 398-day certificate lifetime.

