

SAPfans.io · EgD-SAPF-001

The Additive Doctrine

The next generation of SAP fans, in the sovereign domain.

SAPfans in the sovereign domain — a practitioner charter for adding capability to the enterprise ledger without displacing it.

Published	sapfans.io
Held at	the SAPfans.io repository — clonable in full
Contract	EgD-BOOT-001
Independence	Not affiliated with, endorsed by or sponsored by SAP SE or any service provider

Summary

This paper states the position SAPfans.io is built on, so that a practitioner can read it once and know whether they belong here.

The enterprise ledger — SAP above all, but not SAP alone — is the accumulated record of how organisations actually run. It was built by engineers over five decades and it works. Nothing in this doctrine seeks to replace it, route around it, or embarrass it.

What this doctrine seeks is the right to add. To mirror what is needed, read-only, into custody the customer controls, and to build the new-dimensional services that the ledger was never designed to provide — without write-back, without displacement of the system of record, and without asking anyone to abandon an investment that is working.

That is the whole of it. Additive, not substitutive. Mirror, never cannibalise. In conflict with none.

Why now

Two things changed at once, and their combination is what makes this urgent rather than merely interesting.

The first is that artificial intelligence made the enterprise ledger newly valuable. Not the software — the record. Decades of posted documents, master data, master recipes, plant maintenance histories and supplier behaviour are exactly the substrate that makes a reasoning model useful rather than plausible. Whoever holds that record in a form a model can read holds the leverage.

The second is that the platforms noticed. The response across the industry has been to offer intelligence inside the platform boundary, on platform-operated infrastructure, at platform-set terms — and to describe the result as sovereign.

It is worth being precise about what that word is being asked to carry. At SAP Sapphire in Madrid on 21 May 2026, the sovereign options presented were SAP-operated and in-region: European infrastructure,

European jurisdiction, European models. That is a genuine and useful thing. It is data residency. It answers the question where do the bytes sit. It does not answer the question who can move them, and on whose terms — and that second question is architectural sovereignty. Conflating the two is the single most consequential imprecision in enterprise software today.

	Data residency	Architectural sovereignty
Question asked	Where do the bytes sit?	Who can move them, and on whose terms?
Satisfied by	In-region, vendor-operated infrastructure.	Customer-administered custody of a complete copy.
Survives vendor exit	No.	Yes.
Survives a terms change	No.	Yes.
Commonly called sovereign	Yes.	Yes.
Actually sovereign	No.	Yes.

The additive claim

The claim is narrow on purpose, because a narrow claim can be honoured by a vendor without loss, and a broad one cannot.

We ask for read access to the record, and we give up every right that would threaten the vendor's position. No write-back. No displacement of the system of record. No shadow ledger competing for authority. No claim that the platform is unnecessary — it plainly is necessary, and saying otherwise would be false as well as rude.

In exchange, the practitioner gains the right to mirror what is needed into a store the customer owns, and to build on top of it. A supplier-call agent that raises the yield of an SAP capital asset the customer already paid for. A twin that answers a question the standard reporting layer was never shaped to answer. A model that reasons over the organisation's own history rather than a generic corpus.

Every one of those makes the underlying platform more valuable, not less. A customer who can do more with their ledger renews it. That is why this is a non-disruption play and not a competitive one, and why it can be put to a vendor as an invitation rather than a demand.

Sovereignty, stated precisely

Four tests. A deployment is architecturally sovereign when all four pass, and each one fails independently.

Custody. The customer holds a complete, current copy of their own record, in a store they administer, in an open format, without asking permission to read it.

Portability. The copy is loadable somewhere else. Canonical column names, documented semantics, plain DDL. If it only reloads into the system it came from, it is a hostage with a backup schedule.

Exit. Leaving is priced and possible. The cost of departure is known in advance and does not rise with the length of the relationship.

Continuity. If the vendor changed its terms tomorrow — or was acquired, or withdrew from the market — the organisation would still be able to answer questions about its own history.

Note what is absent from those four. Nothing about jurisdiction, nothing about region, nothing about which flag flies over the datacentre. Those matter for other reasons. They are not this.

Test	Passes when
Custody	A complete, current copy of the record, in a store the customer administers, in an open format, readable without permission.
Portability	The copy loads somewhere else. Canonical names, documented semantics, plain DDL. Not a backup — a destination.
Exit	The cost of leaving is known in advance and does not rise with tenure.
Continuity	If the vendor's terms changed tomorrow, the organisation could still answer questions about its own history.

What we do not do

This community keeps its critique aimed where it belongs, and the discipline is worth stating explicitly because it is easy to lose in a comment thread.

We do not attack engineers. The German engineering tradition that produced the enterprise ledger is one of the genuine achievements of industrial software, and the people who built it — and the architects, consultants and support staff who have kept it running for thirty years — are our colleagues, not our subject. Where there is criticism here, it is of executive lock-down provisions and licensing posture. Never of the craft.

We do not attack service providers. A practitioner arriving from a global integrator, a boutique, a customer's internal team, or their own single-person practice is the same practitioner. The pattern published here is deliberately provider-neutral: it requires no product, no partnership tier, and no permission from anyone's channel organisation. If your employer's methodology and this one disagree, follow your employer's and take what is useful.

We do not fork the community. SAP Community, the user groups, and the vendor's own forums do things this site will not attempt. Go there too.

We do not publish client material. Ever, in any form, however anonymised it feels.

The heritage we are claiming

The name is not decoration and it is not nostalgia. It is a claim about what this is for.

Before there was a vendor-run community, there were practitioners answering each other's questions on independent forums. SAPfans.com predated SAP's own developer network by several years — the SAP Developer Network was created in July 2003, and by then SAPfans had already been the place people went. A parallel site, sapfans.org, was set up in January 2000 specifically so consultants could reach

each other and find work. The technical polish was lower than the vendor's later platform. The independence was total.

And then a great deal of it was lost. Contemporary accounts of SAPfans.com note that its archive survived only in part, with material lost in server crashes because backups were not being kept. Years of practitioner knowledge — the accumulated debugging of a generation — evaporated because it lived on one machine that belonged to someone else.

That is the lesson this site is built on, and it is the same lesson at enterprise scale. A community that does not hold its own record does not have one. Everything published here is committed to a public repository, hash-stamped, and clonable in full by anyone who wants it. If this site disappears tomorrow, every practitioner who cloned it still has all of it. That is not a technical preference. It is the entire point.

What a practitioner does here

This is a working site, not a manifesto with a mailing list.

Take the pattern. The repository pattern, the twin structure, the provenance discipline and the handoff format are published in full and free to use. They were derived from an estate of 105 repositories and they encode mistakes already made, so that they do not have to be made again.

Point your own AI surfaces at the record. The largest avoidable cost in AI-assisted enterprise work is an agent that starts cold and searches the open web for something already committed to a repository it can read. The practitioner guide is a sequence for preventing that, and it works on any surface.

Build a twin. One real subject, one repository, provenance on every input, bulk indexed rather than committed. Start with something small that you own outright.

Extend it. This doctrine is expanded by practitioners, not by an editorial board. Corrections and additions arrive as pull requests and are merged under one rule: append, correct, supersede — never delete. The record is the asset. Erasure is the only unforgivable edit.

The invitation to the vendors

Stated plainly, because vendors read these things and a demand is easy to dismiss while an invitation is not.

We are not asking you to open your source, lower your prices, or concede a strategic position. We are asking for three specific relaxations, each of which costs you very little and returns a customer who stays.

One. Permit a customer to hold a complete read-only copy of their own operational record, in an open format, without a licence penalty for doing so.

Two. Document the semantics well enough that the copy is loadable elsewhere. Not helpfully — sufficiently.

Three. Price exit in advance, publicly, so that a customer's cost of leaving does not silently rise with their loyalty.

A customer who knows they can leave and stays anyway is a stronger position than a customer who cannot leave. The first is a preference; the second is only a lock, and locks are what customers eventually route around.

Closing

The enterprise ledger is a product of the commons, built by many hands over fifty years, and it is too important to be the private gate through which enterprise intelligence must pass. It is also too good to be discarded by people who have not understood what it does.

Both of those are true at once. Holding them together is the whole discipline of this site.

Additive, not substitutive. Mirror, never cannibalise. Custody with the customer. Critique aimed at provisions, never at practitioners. Expanded by whoever shows up with something true, regardless of the badge on their laptop.

In conflict with none.

This paper is free to read, quote, translate and cite with attribution to EVEglyphDesign and a link to sapfans.io. The repository pattern, templates and scripts it refers to are published for practitioners to use in their own work, for any client, at any employer, with or without attribution.

Held in the open at the SAPfans.io repository. If this document disappears, every practitioner who cloned it still holds the whole of it. That is the argument, applied to ourselves first.