GOSPLAN 2.0 · OPEN PLANNING

GOSPLAN 2.0 system architecture

How open planning is built: six layers from registries to publication, how the goal, execution and audit loops map onto them, which artefacts carry legal weight, and which parts already work.

Print version of a site section

Source: https://gosplan2.ru/en/architecture

Text updated: 2026-08-23 · PDF generated: 2026-09-05

What open planning is assembled from: six layers, three loops laid over them, and the rule that forbids the loops from merging. The three loops are a division of authority: who sets the goals, who executes, who verifies. Architecture answers a different question — what they are physically made of. Which stores, where the facts come from, what exactly gets signed, and what backs a conclusion. What follows is a bill of materials, not a list of technologies: here technology is always downstream of a requirement, and wherever paper and a telephone satisfy the requirement, nothing digital appears.

Six layers

Bottom-up. Each layer exists because one of the loops requires it, and none is there for later. Upper layers are useless without the lower ones: calculation without measurement yields plausible numbers, and publication without calculation yields a handsome shopfront.

L0Registries and reference data

The foundation of everything above it: stable identifiers for territories, plots, facilities, organisations and obligations. Without shared identity no two datasets can be matched, and the system becomes a heap of irreconcilable tables. This is the dullest layer and the most common cause of failure.

L1Measurement

The source of fact. Meters, network and fleet telemetry, satellite imagery, environmental sensors, event recorders. The governing requirement: whoever measures must not depend on whoever is measured — an indicator computed by the party being paid for it degrades within one contract cycle.

L2Event journal

A chronological record of what happened: which data arrived, which decision was taken, which version of the calculation was in force. It is amended only by appending — a correction is a new entry referring to the old one, never a rewritten line. This layer is what makes the question “what did we know at the time” answerable.

L3Calculation

Turning observations into indicators: aggregation, models, forecasts. The layer has one hard requirement — reproducibility: an outsider given the same inputs and the same version of the method must arrive at the same result. A calculation that cannot be repeated carries no legal weight.

L4Obligations and settlement

Contracts, quotas, limits and payments. This is where the execution loop lives: auctions, tenders, payment on delivered result. Automation is appropriate exactly as far as the result is measurable by L1 — otherwise a smart contract merely pays out faster on an unverified basis.

L5Publication

Everything the system emits: open data in machine-readable form, interfaces for the resident, the contractor and the auditor. What is published is the data and the method, not a report about the system — otherwise the conclusion cannot be checked from outside and layer L3 loses its point.

How the loops map onto the layers

Layers are the machinery; loops are authority. Their intersection is the architectural decision — and what matters is less which layers a loop uses than which ones it must never control.

Goal loop

Works from published data and from the registry of the objects it decides about. It has no access to the calculation layer or the obligation layer: a goal is stated in terms of the result, not in terms of who will deliver it and on what terms.

Execution loop

Enters obligations and settles them inside the frame the goal loop set. It reads indicators but neither computes them nor edits the method: being able to adjust the method is the same as being able to award yourself the result.

Audit loop

Owns measurement, the journal and the calculation method. It takes no part in obligations and states no goals. Its single function is to answer whether what was reported actually happened, and to what extent. The binding constraint of the whole architecture: the audit loop's infrastructure must not depend on the execution loop's. A shared server, a shared administrator or a shared support contractor merges the two loops technically even where paper keeps them apart — and the model collapses into self-reporting.

What carries legal weight

The system handles a great deal of data, but only a few artefacts create obligations. They are worth listing plainly: everything else is supporting material, and any dispute will turn on these five. Separating the signatures is not a formality. It is what turns “whoever executes does not

verify” from a declaration into a property of the system: breaking it requires forging another party's signature rather than reaching an understanding within one office.

The requirements that drive the technology choices

The order here is the reverse of the usual one. First state the property the system must have, then look for a way to provide it. A technology that satisfies none of these requirements does not enter the architecture, however current it may be.

Reproducibility

An outsider with the published data and the method version arrives at the same result. This is a requirement on L3 and L5 at once, and the reason methods are versioned at all.

Independent measurement

The party being measured does not control the measurer — not through the org chart and not through the maintenance contractor. Hence the requirement to duplicate critical indicators with sources of a different nature.

An immutable journal

The past is not rewritten. Hence the hash chain in L2 — not because blockchain, but because “what was known at the time” has to have an answer.

Degrading to manual

If any automated layer fails, the procedure must continue on paper under the same separation of authority. An architecture that halts governance when it breaks is unfit for infrastructure.

Cost of verification

Verifying a fact must cost markedly less than the fact itself. This requirement rules out national-scale end-to-end on-chain accounting: there, verification costs more than most of the operations verified.

Readiness, component by component

Mixing what works with what is merely designed is the costliest error in writing about digital governance. Readiness is therefore stated per component rather than as one verdict on the system.

In production use

- Machine-readable publication of procurement — the Open Contracting standard and the national systems built on it.

- Metering and network telemetry: layer L1 has existed and run in energy and transport for decades.
- Participatory budgeting as a goal-loop procedure — practised since 1989 and reproduced by hundreds of cities.

Assembled from existing parts

- A hash-chained journal at the scale of one territory: routine engineering; the hard part is the procedure, not the cryptography.
- Versioning calculation methods and publishing input snapshots — solved in engineering terms, adopted almost nowhere in organisational ones.
- Payment on verified result in a narrow subject with a natural unit — road works, waste collection, energy efficiency.

Unsolved

- National-scale end-to-end on-chain accounting: throughput and cost of verification are an open problem, not a matter of time.
- Goal-setting through a DAO: token-weighted voting reproduces a shareholders' meeting, not a civic loop.
- Independent measurement where the result has no natural unit — care, education, culture. Here the architecture is deliberately not applied. Nothing in the right-hand column is a precondition for anything in the left. The system is testable in parts, and each part is more honestly tested where a mistake does not cost too much.

What the architecture does not solve

Listed here rather than in a footnote: an architecture without stated limits reads as a promise, and a promise cannot be checked. It does not protect against capture of goal-setting. If the goal loop is formed undemocratically, flawless machinery will simply execute somebody else's will faster and more precisely. It does not widen where metrics apply. Where a result does not reduce to a unit of account, layers L3 and L4 are not deployed at all — a design decision, not an omission. It does not replace institutions. Signature, procedure and liability for false data live outside the system; the architecture makes a breach visible, not impossible. It does not scale by itself. Properties verified on one territory require separate verification at national scale — starting with the cost of verification.

What matters here

The architecture is deliberately modest: not one component is in it for the sake of being current. Layers appear where a loop needs a fact, a record, a calculation or a signature, and nowhere else. That makes the construction testable in parts — and lets any of its technological pieces be dropped without breaking the logic. Strip out the blockchain, the models and the telemetry, and a paper version remains with the same separated signatures: slower and dearer, but the same logic. Strip out the separation of loops, and what remains is automated self-reporting, which no technology can fix.
