

When You Can't See All the Pieces, You Can't Put It All Together.

**How fragmented visibility is slowing network
deployments and what it takes to regain control**

Executive Summary

Network deployment projects are running at a scale and pace most execution models weren't designed to handle. Between multiple markets, tightening timelines, and funding structures that attach real consequences to delays, the margin for error in operational discipline has disappeared.

Three patterns are making programs harder to manage: scale that has outpaced antiquated management models, permitting that defines the pace of the entire schedule, and execution that breaks in the seams between workstreams.

The tools most teams rely on; multiple disconnected systems, manually assembled status reports, phase-based handoffs — were built for a different level of complexity. At the scale of today's programs, they don't provide useable information from the raw data nor the visibility required to make confident decisions.

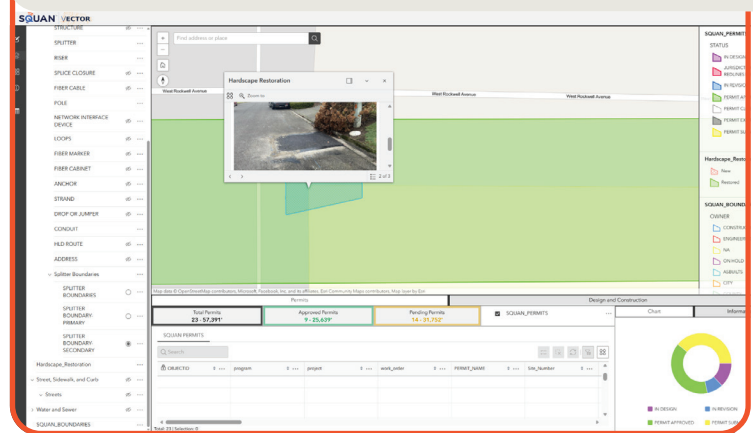
The Changing Shape of Accountability

There's a version of network deployment that lives in planning documents, funding applications, and milestone schedules. And then there's the version that plays out in the real world — across jurisdictions, crew schedules, permit queues, and field conditions that don't cooperate with static models that lack true visibility into the pipeline within one system for all of these functions.

Program leaders know the job is as much about delivering on commitments as it is deploying fiber. Those commitments are tied to capital release, compliance reporting, and the kind of credibility that takes years to build and moments to lose.

The audience for that performance has also expanded. Many programs now find themselves answering to lenders or grants tracking draw schedules, state broadband offices, boards and investment partners, and clients who need to see progress reflected accurately in reporting, not just a spreadsheet reviewed during the next status call.

That's why SQUAN created VECTOR, a GIS-based deployment management platform. It connects engineering, permitting, construction, and invoicing data in a single, live map, so teams can see what's happening, where it's happening, and what it means for the pipeline schedule, and budget. Built to deliver greater visibility, velocity, and execution oversight, VECTOR gives program leaders the operational clarity to manage constraints before they become delays.



Execution pressure, stakeholder scrutiny, and the gap between plan and field reality have always been part of the job. What's changed is the pace of today's programs, the funding structures that attach consequences to delays, and the speed at which small misalignments can compound.

The harder question is whether the systems built to manage those programs have kept up.

What's Getting in the Way

When experienced teams fall behind, it's not due to a lack of ambition or technical skill. It happens because the environment has gotten measurably more complex with stringent expectations, and the friction shows up in places that aren't always obvious until they've already cost something.

Scale Has Outpaced the Management Model

The programs running today look different from the ones that shaped most operators' current execution models. Multi-market deployments with consistent focus on speed to market, spanning different jurisdictions, regulatory environments, and local conditions are now the norm for anyone operating at meaningful scale. In a single-market build, one team can hold the full picture in their heads and fill gaps with informal coordination. This can quickly fracture when the same logic gets applied across five markets simultaneously.

As programs grow, the distance between what the data says and what's actually happening in the field tends to widen. More moving parts mean more handoffs. More handoffs mean more places where the picture degrades. And the larger the program, the longer it takes for issues to surface in a way that decision-makers can see and act on.

Permitting is the Critical Path Now More Than Ever

In some companies, permitting is simply treated as a necessary step in the process. That is falling short.

As fiber expansion has scaled across the country, permitting is at the front and center of all major projects. Municipal approvals vary widely between jurisdictions. Utility coordination introduces dependencies that sit entirely outside the operator's control. Environmental reviews add complexity that can't be engineered around or rushed through. Aerial permitting varies drastically in level of effort and timelines to create permits, track from submittal through make ready construction complete and a better visibility is needed to effectively manage this.

Individually, these variables are manageable. Across a large or multi-market deployment, they become tightly interwoven, meaning a permit that takes three weeks longer than expected affects construction sequencing, crew planning, and downstream activities/schedules.

Teams build schedules against expected approval windows, only to discover mid-program that those windows have shifted due to field conditions or jurisdictional capacity that weren't visible at the planning stage.

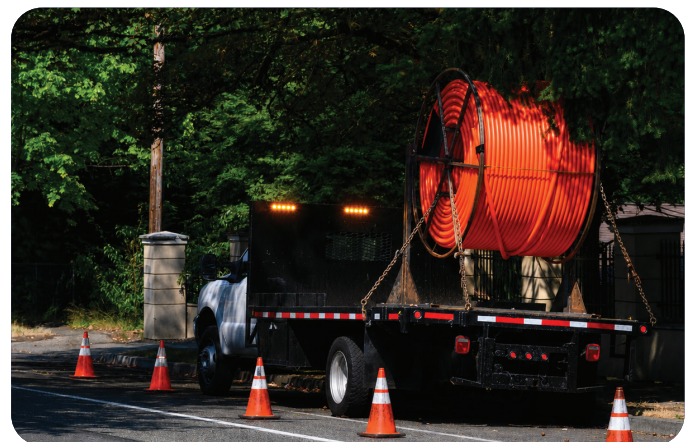
By the time that reality diverges from the model, crews are staged, capital is committed, and recovery options have narrowed.

Execution Breaks at the Seams

When programs run into trouble, it's not typically due to any single function. Engineering is doing its job. Permitting is pursuing approvals. Construction is ready to move. The problems show up in the space between functions.

For example, engineering may complete designs that meet technical requirements but haven't been pressure-tested against local permitting realities or construction sequencing. Permitting teams work from those designs, only to discover late-stage constraints that force a redesign. Construction mobilizes against an optimistic schedule, then slows when upstream issues appear in the field.

None of this looks like failure from inside a single workstream. But collectively, these gaps erode schedule confidence in ways that are hard to trace back to a single source. Add in the weight of fixed milestones, public funding obligations, and reporting requirements tied to capital release, and they lead to draw reviews, milestone check-ins, and tense stakeholder conversations.



How Teams Are Coping

The tools most deployment teams rely on weren't chosen carelessly. The problem is that the systems themselves were designed for a level of complexity most programs have already outgrown.

The Standard Toolkit

Most project management operations operate some version of the same setup: engineering is tracked in one system, permitting in another. Meanwhile, construction status lives in spreadsheets, customer portals, project management software, and GIS files that can't talk to each other in real time. Status gets assembled manually by someone whose job has become translating between systems rather than making decisions in them.

Build readiness lives in specific segments, tied to specific permits, field conditions, and crew availability. And the standard toolkit, however well-maintained, is almost always offering a summary of what already happened, not a view of what can happen next.

When Visibility Lags Behind Reality

Without a clear sense of what could happen with the build, teams mobilize based on the last report rather than current conditions. The downstream effects follow a recognizable pattern:

- ✗ Crews stall while upstream issues get sorted
- ✗ Schedules compress to recover time that's already gone
- ✗ Reporting becomes reactive, describing problems rather than managing them
- ✗ Stakeholder conversations shift from progress updates to explanations

None of this is catastrophic in any single instance. But at scale, misalignments don't stay small for long.

Phase-Based Execution as a Structural Limitation

There's a deeper issue underneath the tooling problem.

Network deployment is typically sequential: engineering finishes, permitting starts, construction follows. Accountability passes forward at each stage. It's a logical model and functions well enough when timelines are flexible and the phases stay relatively clean.

What it doesn't handle is the often-messy reality of modern deployment programs: the overlapping phases, constraints from one workstream influencing another, and supposedly "clean" handoffs that turn out to be anything but.

In a phase-based model, ownership gets murky at every handoff. So by the time a constraint is visible, the opportunity to re-sequence, re-prioritize, or re-route it has passed.



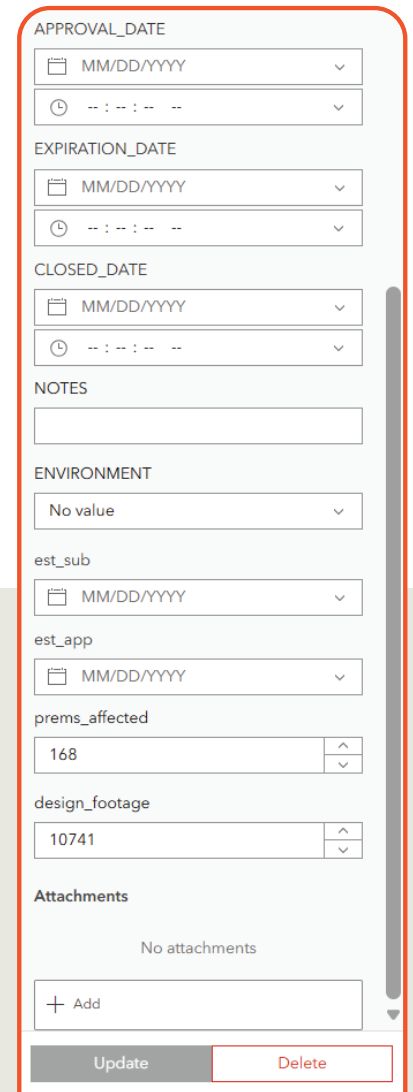
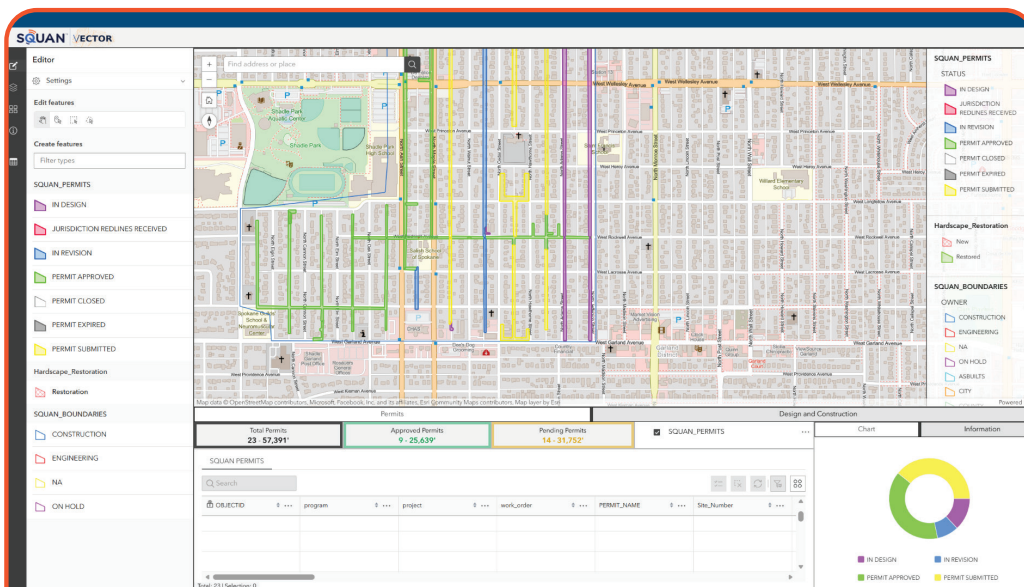
What Controlled Execution Looks Like

Programs that maintain schedule confidence under real deployment pressure share a common characteristic. They treat engineering, permitting, construction, and closeout as parts of a single delivery system rather than a sequence of handoffs. Ownership extends beyond scope boundaries. Constraints get surfaced when they can still be managed, not after they've become a reporting problem.

In practice, this looks like:

- ✓ Construction and permitting work from the same view, so permits can be acted on as soon as they're ready instead of waiting for a phase handoff
- ✓ Reporting reflects real field conditions, making it useful for both decision-making and stakeholder management
- ✓ Permitting informing sequencing in real time, so designs are prioritized against what can actually be approved and built
- ✓ Long poles in the tent identified early enough that they can be mitigated rather than stall

This model hinges on real-time geographic visibility. With SQUAN's VECTOR, teams can make that model operational.



How VECTOR Makes It Operational

VECTOR is a GIS-based deployment management platform that puts engineering, permitting, and construction data on a single, live map

By showing status geographically, VECTOR changes how fast and how confidently teams can act on it. Reconciling 25 permits across a multi-jurisdiction program now just takes a glance at a map, not hours of cross-referencing.

The platform is organized around three capabilities that correspond directly to where deployment programs most commonly lose control.

Total Visibility

Most programs aren't hurting for data. But rarely does all that data feed into a unified view of the build rather than a summary of each workstream's status.

VECTOR replaces that fragmented picture with a single, live map of the program:

- ✓ Every permit is visible on the map, colored by status, clickable for detail, and updated in real time as it moves through the coordination process
- ✓ Engineering, permitting, and construction data is connected in one view, replacing the jumble of disconnected files, separate permit portals, and individual Excel trackers
- ✓ Permit documents, as-built records, and relevant GIS data are accessible from the same interface
- ✓ Photo documentation is geo-tagged to specific locations tied to the relevant structure, so closeout packages come together without hunting through thousands of unlabeled photos

This way, determining what can move means looking at the map, not assembling a report.

Decision Velocity

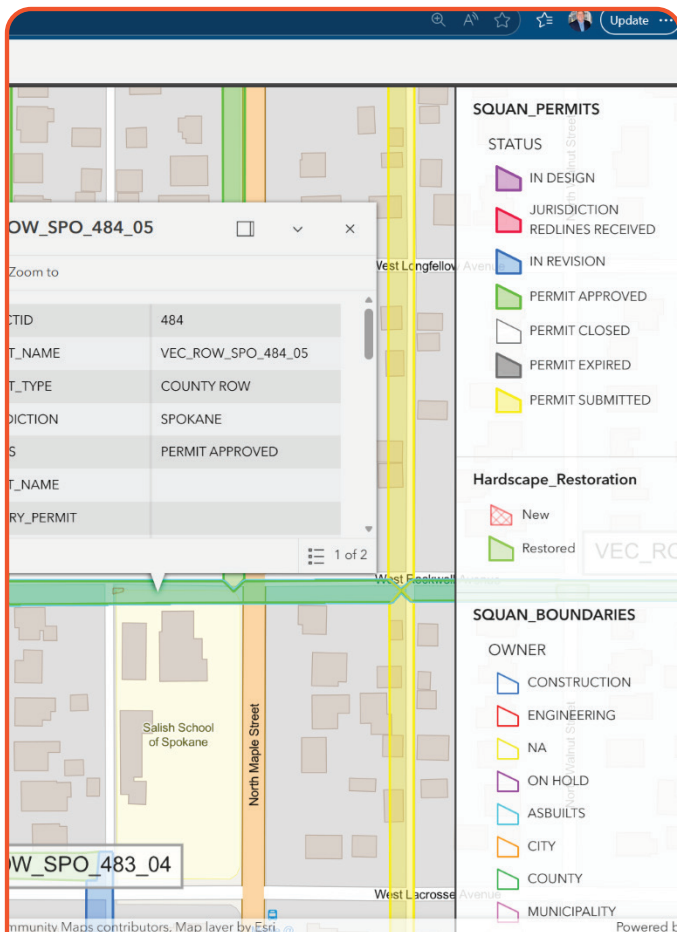
Visibility matters. So does the speed at which teams can act on it.

VECTOR answers this by reducing the time it takes program leaders to see what's happening in the field and respond accordingly:

- ✓ Drill from the program overview level down to projects, work orders, and individual permits, showing exactly where the critical path is under pressure and where resources should focus
- ✓ Custom dashboards surface crucial program metrics, including footage delivered, passings completed, permits pending, and ready-to-build construction
- ✓ Employees update status in real time from the field application, so the map reflects current conditions rather than last week's data
- ✓ Stakeholder reporting draws from the same live system, so it reflects current conditions rather than what was going on at the time of the last data pull

When every segment is trackable in real time, decisions don't need to wait for the next status call or the next report cycle.





Execution Control

The third capability is where accountability moves from principle to practice.

Most deployment tracking systems require progress to be measured against fixed segments in approximate terms. But field crews don't build in pre-defined chunks, creating a gap between what actually happened and what gets recorded as progress.

VECTOR closes that gap:



Span-level tracking is captured in real time by field crews, not approximated after the fact



Custom statuses, dashboards, and data attributes match how the program actually operates



A single source of truth supports both internal decision-making and external reporting, so the data used to manage the program is the same data used to report on it

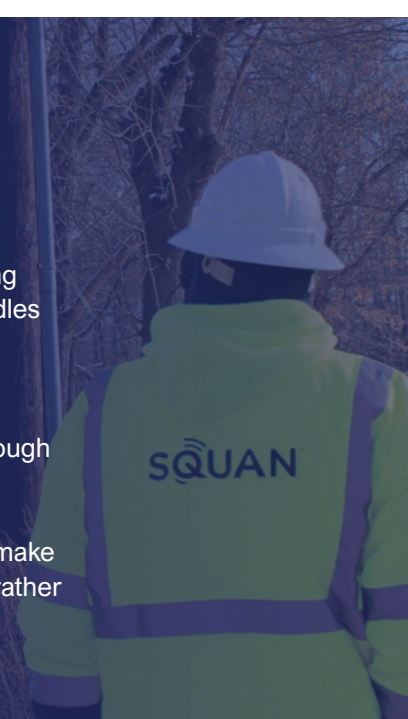
When the system accurately reflects what's truly happening in the field, reporting becomes something program leaders use to make decisions and build stakeholder trust.

The Proof is in the Programs

VECTOR runs SQUAN's programs, from single-market fiber builds to multi-state programs tracking thousands of simultaneous permits. That operational grounding is visible in how the platform handles the specific friction points of actual deployment programs, not hypothetical ones.

The difference between programs that stay on track and those that don't usually comes down to whether the constraints that will inevitably surface did so early enough to be managed, or late enough to become the story.

VECTOR doesn't eliminate the constraints impacting every deployment program. What it does is make them visible geographically, in real time, and early enough that the response can be sequencing rather than recovery.





SQUAN[™]
Engineered Infrastructure Realized.