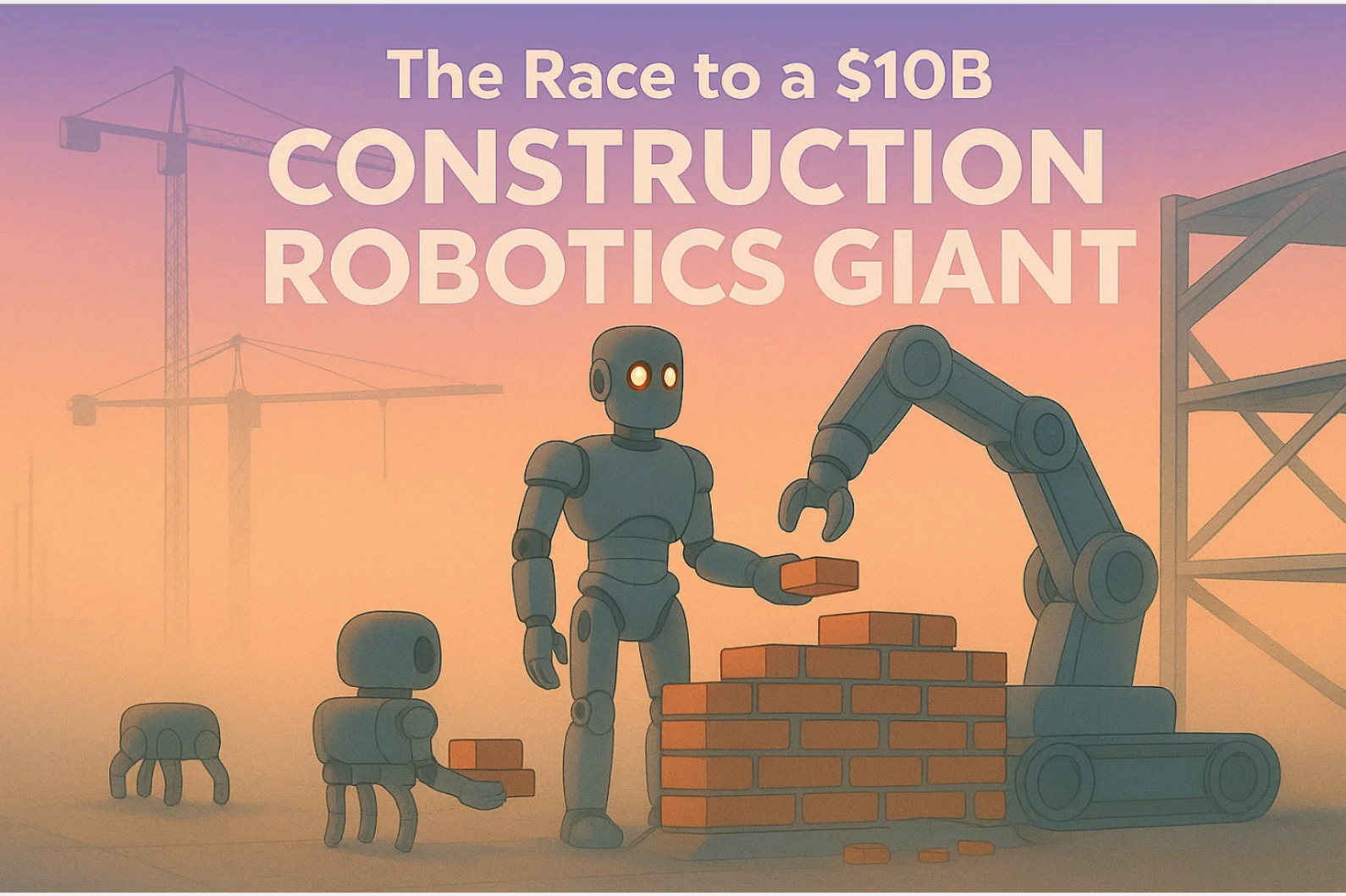
OUTCOMES vs RAILS

FEB 2026



Robots are finally good enough to work in real life construction environments. We think this will lead to trillions in value creation, as one of the world's largest industries is disrupted. But, what will the winning players have in common?

PT1 is a pre-seed and seed-stage venture capital firm investing in transformative technologies across Europe that make a tangible impact in real estate technologies, energy transition, infrastructure resilience and climate adaptation.

[PT1]

The race to a $10B construction robotics giant

Every week someone drops a fresh take on robotics: humanoids are the next big thing, LLMs will run factories, or everything will be automated "sooner than you think." And fair enough, the energy in the space is real, and some of the breakthroughs are genuinely exciting. Amid the noise, robotics is quietly making inroads into our favourite sector: construction. Not just because the tech is ready, but because the market needs it. The fundamentals are dire:

1. a dwindling workforce, with 41% of US construction workers retiring in 5 years;
2. rising construction costs, with prices up 40% from pre-Covid levels;
3. stagnating productivity, improving at a moribund 1% per year for the past two decades.

Robotics could be the solution to this perfect storm of issues.

But, it hasn't worked yet. The early generations of robots failed. They couldn't survive the rough, unstructured environment of real projects: they were too expensive, too fragile, or just unable to adapt to each site. This made adopting robots impossible. Compare this to manufacturing, where robots excelled.

After all these failures, it's fair to assume that maybe construction would never be robot suitable. The market consensus disagrees, and instead these failures galvanised a new generation of founders to make this work with better tech. Today's startups are leagues ahead, commercially and technologically. Commercially - they have leaner balance sheets, tighter spending, and a laser focus on getting shit done. Technologically - these robots are armed with decades of innovations in AI, compute, sensors, servos, and more. New machine learning models and better sensors allow robots to perceive and adapt to unstructured environments better than ever. Modern simulation tools and AI frameworks have also cut development time and costs. In short, the tech is catching up to the challenge. For the first time, we're seeing robots not just demo well, but actually deliver on site.

Investors have taken notice. After the dip in 2022–2023, capital roared back into robotics in 2024, with volumes nearing the 2021 ZIRP peak. While much of this goes to high-profile bets in autonomous vehicles, defence drones, and humanoids, construction robots are quietly becoming a breakout category. In Q3 2025 alone, 37% ($1.36B) of all ConTech investments in the first nine months of the year went into Robotics solutions (a 125% increase over total 2024 levels).

We mapped out a lot of the startups in this space and were able to clearly categorise them into two buckets. On one side, there are huge rounds for the AI infrastructure plays: startups building the "foundational brains" for future robots. Skild AI raised $1.4B in a Series C round valuing the company north of $14B to scale its general-purpose "robot brain" software while Genesis AI pulled in $105 million to advance general-purpose autonomy. On the other hand, VCs are doubling down on vertical robotics startups, i.e. those honing in on specific jobs on site. Take Field AI, which recently raised $314M Series B to autonomously operate in highly dynamic environments and collect data such as site progress, safety conditions, and documentation. Heavy equipment automation led the category, representing over 50% of robotics investment with seven deals totaling $684M YTD. Most of these rounds are

markedly smaller than the infrastructure plays, but these companies are at the top of the value chain. The trillions of euros spent in construction annually could go straight to them.

Both of these camps provide something to their customers. The first provides the connective tissue that makes complex autonomous robotics fleets happen. Autonomy platforms, robot orchestration, remote monitoring: the kind of plumbing that lets ten different machines play nicely on one site. **We call this the rails**. The other camp uses the rails to deploy robots that get shit done. Laying bricks, plastering walls, tying rebar. The stuff that has to get done to make a building. **We call this the outcomes.**

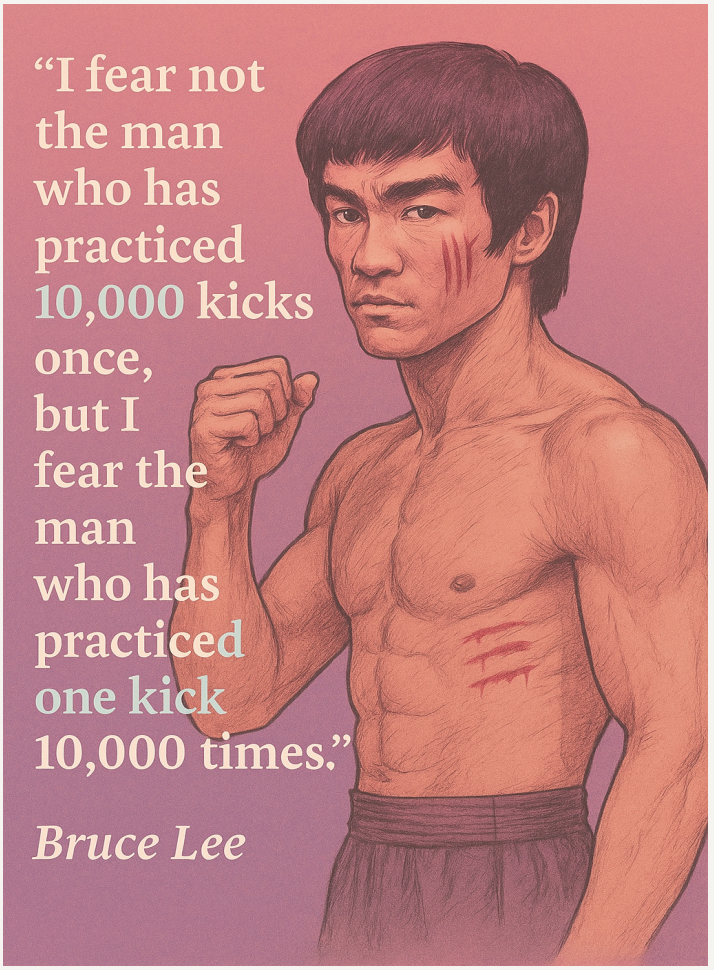
In time, we think everyone will converge into one of these two camps. That's exciting for us as an investor. But which camp will produce a durable and defensible decacorn that this space expects to see?

Let's talk about outcomes vs rails.

Build it and they will come

In venture it's easy to over-analyse and lose track of the actual problem. The problem here is simple. Bricklaying is a skilled profession, and we're running out of bricklayers. Demand for them is increasing, supply is falling, and it doesn't take a Nobel laureate economist to figure out what that will do to the cost of laying bricks.

The outcomes camp are directly fixing this. They make vertically focused robots that do one thing, and only one thing, very well. To paraphrase a legend, fear the robotics construction startup that has practised one thing 10,000 times.



Using our bricklaying example, we have Monumental who are building and deploying autonomous bricklaying robots designed for real-world construction sites. There are other startups doing other things too, like autonomous rebar tying (Toggle), or drywall finishing (Canvas).

Construction companies care about results, not tech for its own sake. If you're promising to *literally lay bricks* or *tie rebar* with a robot, and do it cheaper or faster than a crew, you have their attention. A vertical robot makes it easy for a customer to say "yes" because it's immediately clear what they're paying for (e.g.

walls are getting built for me) with a direct impact on the project budget and timeline. In an industry known for thin margins and chronic labour gaps, a well-executed single-task robot can be a no-brainer.

Victory through focus

Vertical robots win by fixing known and well-defined problems: the issues that construction firms already know they have. All those construction headwinds we spoke about at the start? Vertical robots are the solution.

This also makes them easy to evaluate. Compare them to a human crew, measure the improvements in time/cost/accuracy and you'll realise whether the robot earns its place on site. What we like about construction is how unapologetically commercial it can be. The developers just want to hit their IRR hurdle. Accordingly, time and budget is everything. A vertical robot that cuts labour hours, reduces rework, or improves safety can quickly check boxes for customers. Even modest productivity gains are compelling in an industry where profit margins often sit below 5%. It's important to talk about the GTM here. Sometimes, managing the robot yourself (selling an outcome) is easier than selling a robot which needs to be managed. There is also a structural procurement advantage here. Construction buyers are set up to purchase scoped work packages, not the technology that enables it. A lot of it comes down to insurance, bonding, and contractual risk: it's better to have it sit with someone who delivers the work. A construction buyer using a robot to do a task they previously outsourced suddenly takes on more risk. In addition to this, the outcome providers fit cleanly into existing procurement, compliance, and risk frameworks. This further accelerates adoption for outcome models, independent of technical readiness.

GOLDBECK'S EXPERTISE GROUNDS THE HYPE: ROBOTS SUCCEED ONLY IF THEY DELIVER IN PRACTICE

Goldbeck is one of Europe's leading general contractors, delivering hundreds of projects annually with multi-billion turnover and a workforce of ~13,000 across the continent.

"Collaboration between humans and robots holds enormous potential for shaping our industry. But scaling depends not only on the technology itself, but also on construction site processes, digital readiness, and viable business models that deliver real economic value."

Maximiliane Straub
Teamlead Innovation @ Goldbeck
https://www.goldbeck.co.uk/en

By narrowing scope, vertical startups can go deep on engineering, building a robot that reliably performs one job, even in harsh, variable site conditions. This leads to tech that actually works day in, day out, which is essential in a sector where failed pilots mean blacklists and reputation is hard to recover.

And the stakes are not just reputational. In many trades, millimeter-level deviations can translate into thousands of dollars in rework, failed inspections, or cascading schedule delays, which means accuracy functions as liability control, not just performance. So if a vendor can confidently say, "we will hit ±X mm," they can sell not just "robot capability" but a compliant, insurable work product. Owning the full stack from perception to actuation allows outcome providers to guarantee tolerances that matter contractually, turning hardware precision into a competitive moat rather than a commodity feature.

This is precisely where earlier generations of construction robots failed: they could not deliver consistent, production-grade performance in the field. Purpose-built vertical systems close that gap by design, which is why they are the first to cross from pilots into real workflows.

That unforgiving reliability bar has a direct implication for how robotics can be commercialized in construction. Outcome-led deployments tolerate this technical immaturity far better than platform approaches. By selling completed work rather than robotic capability, providers can backstop automation with human labor when needed, then progressively automate more of the workflow as reliability improves. This allows companies to ship before autonomy is perfect, while still compounding data, operational experience, and customer trust. From the buyer's perspective, performance is stable from day one; from the vendor's perspective, automation becomes an internal margin expansion lever rather than a precondition for sale. Infrastructure providers do not have this luxury: they must demonstrate dependable autonomous capability upfront, because they cannot operationally absorb failures for the customer. This dynamic strongly favors outcome-as-a-service models in early market formation.

But is the slice too small?

Each vertical robot, by definition, addresses a single slice of the huge construction process. Even if that slice (say, drilling anchor holes in concrete ceilings) is valuable, the startup remains boxed into one area unless it expands its scope. There's limited opportunity for cross-trade leverage: a robot that only finishes drywall won't help with laying bricks. This means to grow beyond a certain point, the company either has to develop new robotic products for other tasks (essentially becoming a multi-vertical company, which is costly and complex) or find a way to broaden its use cases via software updates; accessories; or moves into adjacent spaces. Otherwise, it's a silo: you might dominate drywall or rebar tying, but that alone may not get you to a $10B valuation unless that niche itself is enormous worldwide. Many vertical robotics companies will eventually face the "what's next?" dilemma. Do they expand to adjacent tasks (risking dilution of focus) or partner with others to integrate into a larger system? The very focus that drives early success can later become a growth limiter.

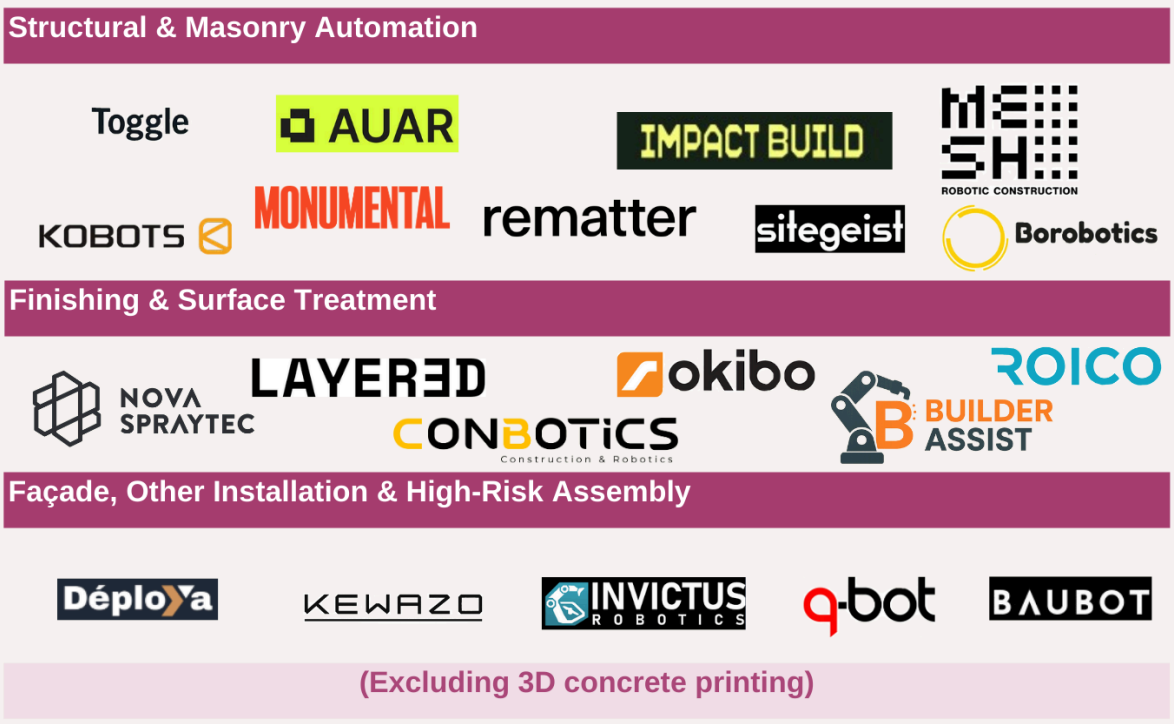
Delivering an outcome on a construction site often means you're delivering a service, not just a product. These robots don't (yet) run themselves, they need to be transported, set up, calibrated, and maintained, and often overseen by technicians (especially in early deployments). A vertical robotics startup can end up looking a bit like a specialty subcontractor or equipment rental company, providing a specific service on jobsites. This is operations-intensive and can squeeze margins. Every new project might require training crews, adapting to site-specific conditions, and handling on-call support if the robot hiccups. Scaling from 5 pilot projects to 500 active robots isn't like scaling SaaS, it's more like scaling a fleet and a field service organization simultaneously. Some high-profile construction robot

startups failed not due to weak tech, but because they couldn't scale the human logistics around the robot across hundreds of unpredictable sites. Supporting lots of variations in site layout, climate, materials, and workflows drives up costs. In short, no surprise, growing a vertical solution can be slower and more capital intensive than a typical software startup (Robots-as-a-Service pricing can help, but it's still a fundamentally boots-on-the-ground expansion).

Because vertical players concentrate on one problem, their defensibility often comes from *executional know-how* more than deep network effects. They accumulate specialised expertise and data by working on that task for years which is valuable, but it's not an impregnable moat. A determined competitor could target the same task with a different technical approach, or a large construction customer might even develop an in-house robotic solution once they're convinced it works. In other words, the moat is mostly in *doing it better* and having a head start. That lead can erode as technology improves or as bigger players wake up. Moreover, without broader platform lock-in, a vertical robot company must keep innovating and extending its lead to stay ahead.

In summary, the outcome-driven approach is customer-friendly and great for early traction, but it can hit a ceiling. It shines at proving value quickly in one area, yet it may struggle to either achieve broad platform scale or fend off fast-followers once the idea is validated. The best vertical-first startups will need a plan for what happens after they win their niche, whether that's expanding to adjacent tasks, franchising their model geographically, or evolving into a platform themselves.

To see how this plays out on the ground, we mapped a set of European robotics companies pushing task-focused automation beyond the lab and into real operating environments. This is not a comprehensive landscape, but a market signal: a snapshot of where customers are actively testing, adopting, and operationalizing robotics today.



OKIBO ARE USING CLEAR OUTCOMES POWERED BY PROPRIETARY RAILS TO OUTCOMPETE AND DELIVER BETTER AND MORE COST EFFECTIVE OUTCOMES OVER TIME

"Okibo is an outcomes company, but to deliver reliable finishing on real job sites we had to build proprietary 'rails' : perception, real-time planning, real-time collision detection, control, and materials-process tech - not available off the shelf. Owning our source code (no black boxes) gives us the freedom to solve what's needed without vendor limits or a glass ceiling. That investment lets us handle real-world variability today and improve cost and performance as we scale; as generic rails mature, we benefit while the hard problems we've solved remain key differentiators. These assets may be transferred directly to future solutions beyond our current mission."

Guy German, CEO

Okibo's autonomous finishing robot

Connections require complexity

So we've covered the first camp. Providing outcomes; actually building things; transforming materials into projects. The second camp is a lot more abstract. It's things a construction project manager would never think of, or care about. But it's the connective tissue that enables robots to work properly. We call this the rails of robotic construction. You can only deliver an outcome on these rails.

This categorisation isn't new to robotics. Let's draw parallels with the cloud. Websites like Spotify, Figma, and Amazon provide a service to their customers. But, they're only possible thanks to platforms like Cloudflare, Paypal, or the big cloud providers.

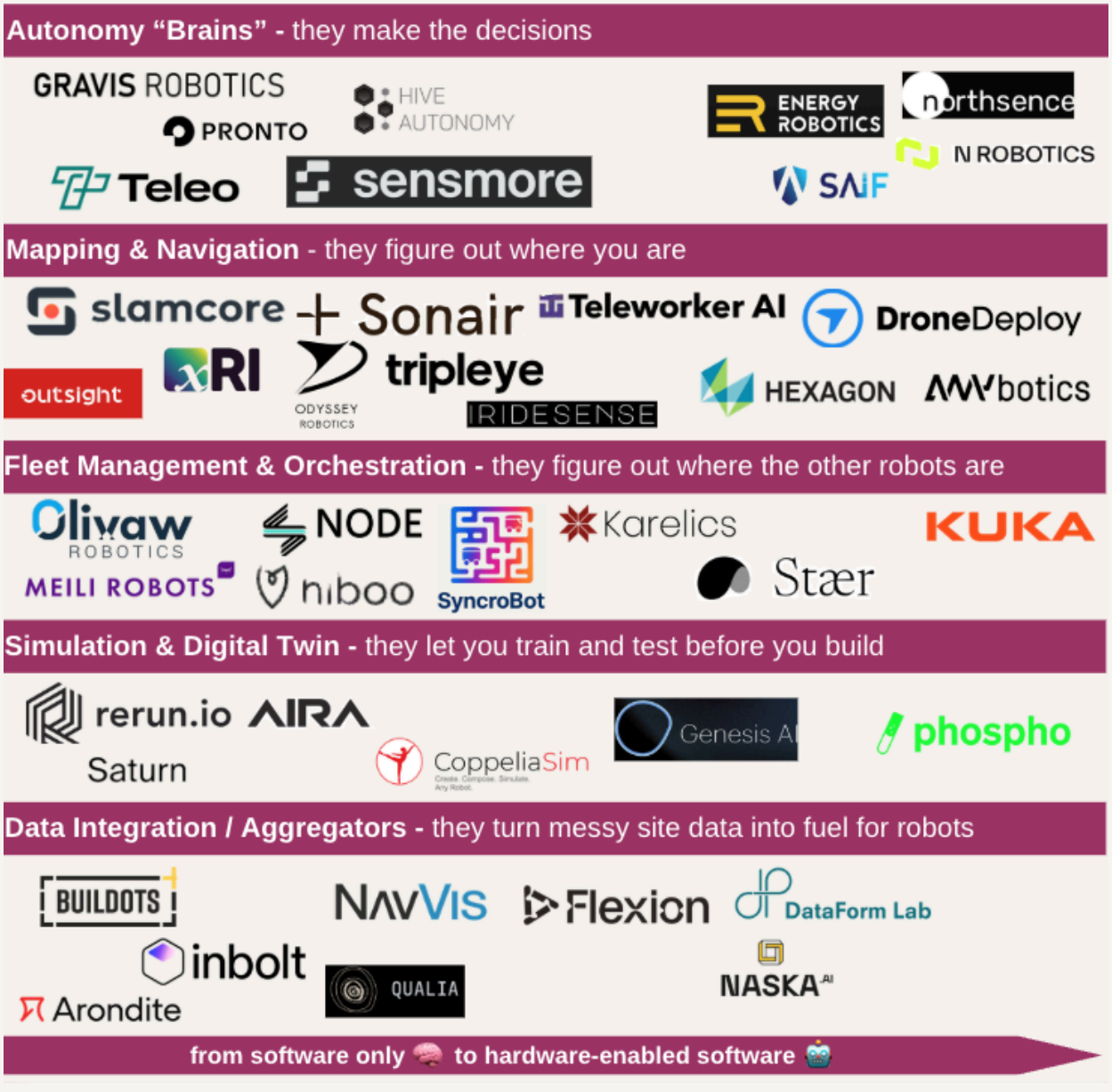
Back to the world of construction robotics. We've spoken about laying bricks. What's actually going on in a bricklaying robot? Well, you might have a robotic arm (probably sold by someone like ABB). You'd have a gripper, which could be proprietary. There'd be a decisioning and control stack too. It ought to connect to project plans (maybe through BIM) so it knows the dimensions of what to build. It should be able to see, so it knows where to pick bricks up and where to place them. This space gets very deep, very quickly. And as complexity scales (with autonomy, or fleet orchestration, or agentic responses); the stack gets deeper. We define the "rails providers" as those who produce the individual layers in this stack.

Instead of delivering an outcome, these "rails" startups provide the underlying components like autonomy "brains," navigation systems, fleet management clouds, simulation engines, or data integration layers that let robots function. The aim is to become the industry's backbone: a widely

adopted operating system or ecosystem that other robot makers (or end-users) plug into for essential capabilities (mapping, vision, coordination, safety, interoperability, etc). For example, startups like Pronto and Teleo provide autonomy retrofit kits for existing heavy equipment (turning standard dump trucks or bulldozers into self-driving robots), and platforms like Formant offer a cloud dashboard for managing and monitoring fleets of different robots on a jobsite. Rather than building a single robot for one job, these startups are selling the picks and shovels for the robotics gold rush: the enabling infrastructure that makes it easier for the whole industry to deploy automation.

So beyond a few well-known examples, what does the rails layer actually consist of?

We also mapped the infrastructure layer that underpins construction robotics. Some of these companies ship hardware, but primarily as a wedge into deeper software defensibility—the robot is the delivery vehicle, not the value capture. We grouped this layer into five functional stacks: Autonomy Brains, Mapping & Localization, Fleet Orchestration, Simulation & Digital Twin, and Data Integration. Not all of these players are construction-native, but their technology sits close to the problem space or is already being adopted in construction-adjacent deployments. Together, these layers form the connective tissue that turns isolated machines into scalable systems and makes on-site automation operationally viable.



One stack to rule them all

By creating shared tools and standards (for mapping, perception, locomotion, safety, data, etc.) an infrastructure provider can scale its technology across many different robots and workflows. This gives a compounding effect: each new robot type or application that adopts the platform adds to its ecosystem and validation. Over time, the platform can become the default choice for multiple use cases. Say your navigation software is used by a manufacturer of heavy machinery; a fleet manager operating robotic forklifts; and a drone integrator deploying autonomous surveyors. You're getting great data from each, and the insights and improvements will cross multiply. Instead of one product in one niche, you have technology that touches multiple niches. This can lead to a widening platform advantage. You get more data, you see more edgecases, you ship more. Eventually, your toolkit becomes the easiest to build on and the hardest to replace. Everyone plugs in because it's more efficient than building from scratch. A dominant construction robotics platform could similarly become the backbone that dozens of robotic solutions rely on, giving the platform owner a powerful position in the value chain. The prize is a much bigger TAM and the power to set standards. If you own the rails, you can collect tolls from all the trains running on them.

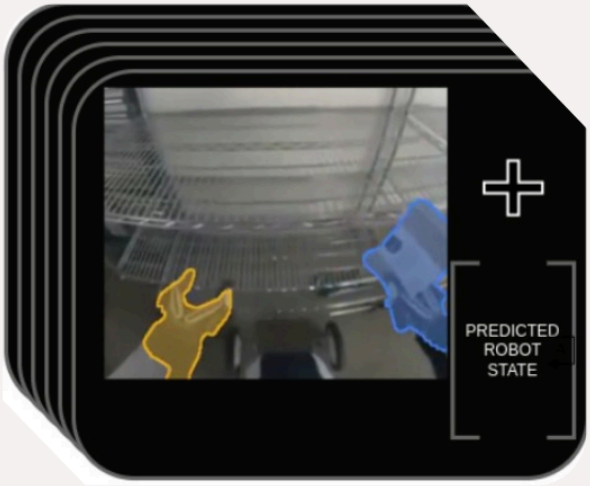
Each new client and therefore each robot connected generates useful data and feedback. If an autonomy platform ends up operating hundreds of machines across different sites, every hour of operation feeds more data into its AI models. This creates a virtuous cycle: more usage → more data → better algorithms → even more usage. Over time, an infrastructure company can accumulate a deep well of proprietary data and know-how that's hard for any one vertical-focused competitor to match. The platform's algorithms get smarter and more robust with scale, giving it a self-reinforcing edge. This is the kind of network effect we love to see: similar to how each new user of a software platform can make the whole product better for everyone, here each new robot or site makes the platform's AI "brain" wiser. An infrastructure play, if it achieves critical mass, can be highly defensible. Even if someone copies the concept, they won't have the years of real-world data, integration experience, and ecosystem partnerships that the leading platform has amassed.

Unlike a vertical startup that must build and ship physical robots for a specific task, many infrastructure-first startups position themselves as software or cloud service providers for the robotics industry. This allows them to scale faster and more efficiently by avoiding heavy manufacturing and focusing on high-margin software and data services. Instead of designing a unique machine for each task, these companies develop universal modules and APIs that retrofit onto existing fleets or support third-party robots. For example, Bedrock Robotics sell autonomy upgrade kits that turn off-the-shelf construction vehicles into robotic systems. These kits are often purchased not by contractors directly, but by equipment owners, rental operators, or OEMs aiming to expand functionality. Meanwhile, platforms like Formant offer a subscription-based control center built for robot manufacturers and integrators (not just the jobsite) providing a "single pane of glass" for managing multi-brand fleets.

Hardware is getting easier. It's no longer the black box it once was. Thanks to advances in off-the-shelf components, faster prototyping, and better documentation, building capable hardware has become more accessible, even for smaller teams. Purely having the best hardware isn't enough. That levels the playing field. The real differentiator now often lies in the software stack: who can orchestrate fleets, manage autonomy, integrate data streams, and offer real-time visibility across a site. By building the

connective tissue rather than the individual limbs, these startups avoid the costly logistics of hardware deployment and instead monetise through recurring software licenses, cloud services, and data analytics. For customers, this often means lower upfront cost and the ability to extend automation across their existing equipment. This gets us excited: scalable, sticky, and increasingly essential to the future of construction automation.

SATURN'S SYNTHETIC WORLD MODEL TEACHES ROBOTS HOW TO THINK FOR THEMSELVES



The Saturn API:
Spins multiple futures by simply prompting Saturn with video and extra contextual information.
https://www.asksaturn.ai

Saturn is a data-driven world model that learns and simulates real-world physics from visual and contextual input. Built on advances in control and video modeling, it predicts future scenes and robotic trajectories without manual asset design. Unlike traditional physics engines, Saturn captures complexity directly from data, enabling flexible, general simulations where behavior emerges from learned patterns rather than fixed rules.

Elisa Seghetti,
Co-founder @ Saturn

Even if you can build it… they still might not come

Platforms and rails seem great. Write some cool software, sell it to hundreds of different clients quickly, iterate, build a moat, exit. If only it were that easy.

By definition, platforms are broad. Too broad, in fact, to build and prove out quickly. Unlike the outcome providers that can craft a contained product (one robot that does one thing well), the rails providers must tackle a wide and complex problem from day one. This can lead to a long development cycle and the risk of being too generic early on. For a small startup, it's a heavy lift to build an "operating system" or universal toolkit with limited resources. Early versions might not perform any single task *better* than a dedicated solution, because they're busy accommodating many possibilities. There's also significant go-to-market friction: selling a platform rather than an outcome means convincing more stakeholders. It's not like approving a robot-enabled job subcontractor.

More importantly, much of the autonomy stack that infrastructure strategies rely on is still not production-grade for construction environments. Many teams are betting on end-to-end

perception-to-control systems, where a single model goes from camera input straight to robot actions. These systems can work in demos, but they remain hard to interpret, debug, and certify, especially on messy jobsites where layouts, lighting, and materials change daily. When something goes wrong, it is often unclear whether the failure came from perception, control, data drift, or rare edge cases, which makes troubleshooting slow and unpredictable. Construction workflows, however, cannot tolerate opaque failures or long downtime, and safety, insurance, and contractual liability all favor predictable, deterministic behavior. As a result, rails providers must clear a much higher reliability bar before customers can depend on them in production, while outcome-led solutions can keep task scope narrow, use deterministic fallbacks, and absorb remaining failures operationally. In practice, this shifts innovation risk away from customers in outcome models and onto the vendor, enabling faster real-world deployment.

So it's harder to build, and it's harder to sell. Also, you need the network effects to prove it works well, but to do that you need to sell.. Building a platform can become a classic catch-22. Resolving this catch-22 is possible, but you need to find visionary customers who believe in you and willing to wait for the payoff. A common solution here is identifying a beachhead use case: a specific application where you can show quick wins. For example, an autonomy platform like Sensmore might first target heavy machinery operations in mining (a controlled environment with clear metrics) to prove it can boost productivity, then expand to general construction sites. But until that happens, you might struggle with customer questions like "What am I really getting for this subscription fee?" or "Can you prove this will save me money *this quarter*?".

The nuance is that not all data compounds equally in construction. The most valuable data comes from failure. Outcome providers can capture this because they own the deployments and fix problems first-hand, while the infrastructure platforms often see only partial, filtered signals. This slows learning for rails and accelerates proprietary advantages for vertically deployed systems

Ultimately, platforms have more barriers between them and the main customer (the source of the funds). They'll get a trickle of the money which goes to the outcome providers. But, it'll be a trickle of a much bigger pie. That said, infrastructure strategies benefit from a powerful distribution lever that outcome providers often lack: OEM and rental channels. Construction equipment is already sold, financed, and serviced through global dealer networks, and autonomy or fleet software that integrates cleanly into those workflows can scale far faster than any startup deploying its own robots one project at a time. This is why many rails players target retrofit kits and OEM partnerships early. If these channels fully open, infrastructure layers could propagate across thousands of machines long before outcome providers can operationally scale equivalent fleets.

The flip side is that these same channels are politically and commercially complex. A broad platform can easily step on the toes of incumbents with proprietary systems, including major OEMs, construction software vendors like Autodesk or Procore, and other suppliers embedded in the stack. As a result, go-to-market becomes a negotiation across multiple entrenched stakeholders rather than a straightforward sale, slowing adoption and increasing execution risk.

Even if you win early partnerships, the long-term defensibility isn't guaranteed. As platforms mature, they'll end up converging. If dozens of customers demand similar data outputs, many infra players will converge around shared schemas and API formats. At that point, you're not a unique system: you're a

plug-and-play backend. In the worst case, your competitors just replicate your endpoints, and switching becomes a one-sprint job for a dev team. The very abstraction that makes infrastructure elegant can also make it easy to replace. Outcome-driven solutions, by contrast, integrate deeper into workflows and can become stickier. There is an important exception to the idea that infrastructure will always commoditize: construction is increasingly run through mandatory digital systems like BIM, digital twins, inspection logs, and regulatory reporting, which are required to plan work, pass inspections, and release payments. If robotic platforms are tightly integrated into workflows, then they become part of how projects are managed. In that case, switching costs rise sharply, because replacing the platform would mean reworking project controls and compliance processes, not just swapping autonomy software. Even if the robot intelligence itself is replaceable, the workflow integration is not, which favors infrastructure players that embed into required construction systems rather than only into robot control loops.

A further uncertainty for defensibility in this space is data ownership. In consumer AI, users generate training data passively - Tesla's fleet learning is the classic case, much like CAPTCHA once crowdsourced OCR training. Construction is different: contractors control site access and may resist sharing data, slowing the compounding flywheel. Unless incentives are engineered through pricing, warranties, or even regulation, the network effects that power most AI models may never fully materialise here.

Despite these challenges, the allure of the infrastructure play is that if it *does* take off, it can become the de facto ecosystem for an entire industry. Windows became the dominant platform for PC software, Android became the standard OS for mobile devices, and more relevantly, there are efforts like NVIDIA's COSMOS and others to become the go-to platform for robotic AI. In construction, a successful platform could mean that *every* robot on site (from drones to bulldozers) runs on your operating system or cloud. That prize is enormous, but so is the execution challenge to get there. And if it doesn't get there quickly enough, markets may punish the vision. A few infra players have raised megarounds at sky-high valuations (sometimes exceeding $1B or even $2B) despite limited revenue. That bet only pays off if usage compounds fast and the moat holds. If not, they risk painful down rounds or getting leapfrogged by more nimble peers. Hype around 'robot operating systems' is real but reality is never far away.

What are we missing?

At first glance outcomes and rails, or horizontal and vertical, seems like a real dichotomy. But we don't think their success is mutually exclusive. There'll be successful companies in both camps; just like Cloudflare and Spotify can both exist together. The two camps need each other to survive. That said, there are a few nuances and things to be aware of:

1. Could the camps merge?
2. How universal can construction get?
3. Is there room for a surprise disruptor?

From laying bricks to running sites

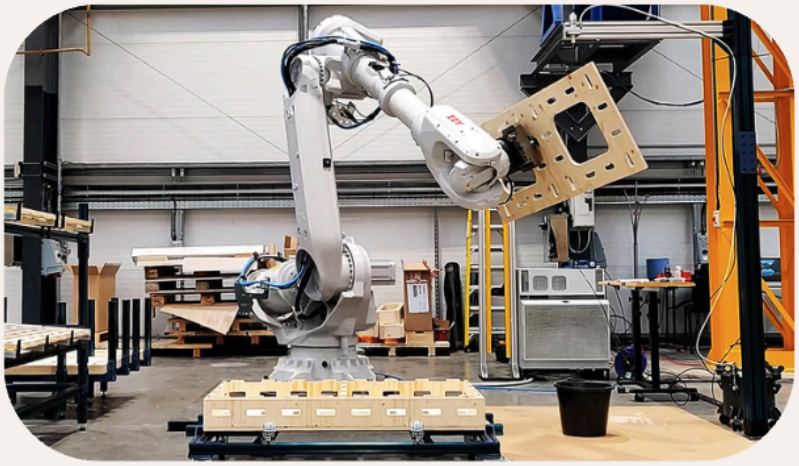
Once you conquer your niche, it's tempting to jump. We expect a few startups to take the plunge from vertical to horizontal, or vice versa.

A vertical-focused construction robotics startup might win its initial market by mastering one high-value task, and then attempt to expand horizontally, turning that solution into the foundation for a broader platform.

Contractors trust what works. Once you've nailed one job, they're more likely to let you handle the next. What starts as "a robot that finishes drywall" can evolve into "the finishing automation layer", a system that coordinates everything from plastering to sanding, and potentially even painting. Outcomes first, platform second. This builds credibility, captures workflow data, and starts to look like a moat.

Crucially, this convergence is not symmetric. Outcome-led companies already control deployments, workflow integration, exception handling, and customer operations, which gives them continuous access to the real-world edge cases and recovery patterns needed to internalize autonomy, orchestration, and data layers over time. Infrastructure-first companies, by contrast, depend on partners to generate the very production environments required to harden their platforms, slowing iteration and weakening learning loops. In other words, it is structurally easier for outcomes to grow into platforms than for platforms to grow into outcome businesses. We've seen this evolution with Toggle: they started with rebar-tying and prefabrication robots generating multi-million-dollar project revenues, then moved up the stack into automation and QA layers across rebar and prefab manufacturing.

AUAR STARTS WITH TIMBER BUT AIMS TO AUTOMATE ENTIRE BUILDING STRUCTURES



AUAR Micro-Factory Manufacturing

"Our approach moves from bits to atoms: using software and AI to drive hardware. Instead of automating a single wall element, we focus on automating all structural timber components of an entire building, for any typology and location where timber is used. The key is software that translates any design into local data (regulations, costs, materials) and generates manufacturing-ready robotic code. The breakthrough is robots that work autonomously and adapt to local conditions in real time, supported by a business model aligned with industry practice."

Mollie Claypool, Co-founder @ AUAR

Of course, the leap from vertical hero to platform leader is anything but easy. It takes more than great tech: it takes cross-functional trust, serious capital, and the ability to generalise a purpose-built tool into something flexible enough to orchestrate others. Most startups won't make that jump. But for the ones that do, that first outcome becomes the wedge: a beachhead that opens the door to recurring revenue, tighter integration, and platform-level defensibility.

Even in these cases, the "ex-vertical providers" can't get deep enough to dislodge the entrenched rails, in particular those at the bottom of the stack. Maybe they could orchestrate a few robots, or plug into the project management software. But replacing the hardware and software that lets robots navigate in a room with no GPS? No chance.

On the other hand, the rails providers usually start by providing a smaller piece of the big vision. A sliver of the shared autonomy stack, part of the robot orchestration layer, or just an interface to track every machine on site. But this is harder to sell to construction. That's why so many infrastructure players lead with a hero use case. You can't just talk about managing every robot on site; you need to demonstrate that you can manage one, really well. Often, that means embedding your system into a specific machine (say, turning a standard excavator into an autonomous one) or working with an early adopter to solve a tangible pain point end to end. The idea: make your platform real by making it useful.

Sometimes, that means building your own "hero" robot or teaming up with a partner to do it. Maybe you've developed a next-gen navigation stack: great, now show it working on a forklift or crane. Maybe you've built a vision system that can help robots on chaotic sites: prove it by applying it to layout or excavation. The goal isn't to become a vertical company, it's to seed the ecosystem with a killer app.

It's the same strategy tech giants used in the early days - Microsoft built Office to drive Windows, Apple built iMovie to sell Macs. In robotics, we're seeing the same playbook. Platform startups might build early applications not because they *want* to own every task, but because they need to show what's possible. Once the use case lands, it attracts others. That's how ecosystems start: not with abstraction, but with something that works.

They still have the same issue as their outcome providing friends though: the stack is too deep for pure convergence. Some players may get large enough to own their stack, like Google owning and running on their own servers. But most software unicorns (and decacorns) don't want to leave AWS. It's the same principle.

Digging in Dubai ain't drilling in Detroit

Construction has huge regional variations. Building archetypes, construction methods, the ratio of labour cost to project costs, the local climate, and even what materials are being used. There's no universal playbook. What works in one region or segment might fail in another.

European and East Asian companies, often operating in labour-constrained, highly regulated environments, tend to experiment earlier with multi-robot systems and may welcome platforms that tie everything together. Meanwhile, a family-run contractor in Texas or Poland may just want a robot that digs faster and nothing else.

Size also matters. A tier-one contractor with an innovation budget may co-develop a platform. A subcontractor with a six-month backlog wants a plug-and-play robot that delivers ROI next week. So vertical or horizontal isn't an ideological choice, it's contextual. The smartest startups geo-target and segment their go-to-market accordingly.

Where else can innovation come from?

We've been arguing that the confluence of material price inflation, dwindling labour supply, and a chronic productivity stagnation drives the demand for robots in construction. But, there's another approach too. MMC, or modern methods of construction, is a broader field that has robotics as a subset, but also includes things like modular or pre-fab construction. This could offer another approach, industrialising construction entirely and stripping away the complexity that forms the business case for most robot solutions. The best MMC startups we've seen have robotics at the core, but it'd be naive to discount this sector entirely.

There's also room for surprise innovation, displacing the incumbents. For example, ABB are a robotics giant. Their robotic arm is used worldwide in manufacturing, industry, and even in construction. Last year they sold over $2b of robots so they're clearly doing well. But we recently came across a startup that was building 3D printable robot arms that could compete with ABB in some contexts. We don't see them competing with ABB in high precision use cases, or customers who can afford it (ie automotive manufacturing); but they'll appeal to the innovators and the disruptors.

Beyond technology and business models, perhaps the biggest predictor of success is ecosystem design. Startups rarely succeed as lone disruptors; they must orchestrate OEMs, contractors, regulators, and integrators to make adoption real. Dedicated experimentation zones, like Japan's Tokku Zones, or hubs deliberately structured to connect the value chain, could prove decisive in moving the industry from promising pilots to scaled deployment.

But who'll win the race?

On a valuation basis, the infrastructure players are way in the lead. A few have already surpassed unicorn valuations. But, these are paper valuations, as a result of monster seed rounds. They're not tied to revenues or intrinsic company value. When it comes to creating real value in construction, we think the outcome providers will win the first mile. They can deploy fast, prove ROI, and deliver real outcomes in the field. They break open the market. But their scope is narrow, bounded by task, trade, and geography.

We don't think they're capital constrained either. Asset backed financing and securitisation are powerful levers for hardware intensive companies to scale without dilution. But, they'll be confined by their own P&L. We're not convinced that they can command the same revenue multiples as a SaaS. Geography is the next challenge. The market is huge, but fragmented. In theory it's big enough for these players to get billion euro revenues, but they'd either need to monopolise a major economy or break into several. Without consolidation (geographical or sectoral), we're not expecting decacorns here.

On the other hand, the rails providers build for scale. They connect machines, unify data, and grow smarter with every deployment. They don't compete with verticals, they enable them. And that choreography is where the compounding starts.

That's why we think the rails will win the race. But, it's a high variance space. Not all rails will lead to empires. Many will commoditise: useful, acquired, forgotten. Standardised APIs. Swappable middleware. Twilio, not Android. But one or two platforms will go the distance. Here is where we'll see the construction robotics decacorns. But it's a complex journey, against much bigger players. Nvidia will try here. Everyone will. But this isn't about large language models. It's about the real world complexities of working on site.

The infrastructure we back won't look like Nvidia. It'll look like Caterpillar meets GitHub: domain-deep, software-native, and impossible to rip out. It won't be the first to a unicorn, as a few are already there. But, it's the only way to build long term durable value.

PT1 is a pre-seed and seed-stage venture capital firm investing in transformative technologies across Europe that make a tangible impact in real estate technologies, energy transition, infrastructure resilience and climate adaptation.

📠 anais@pt1.vc
🌐 https://pt1.vc
🪶 Anaïs Cimbidhi, Burhan Pisavadi

References

- AUAR. (2025). Automated urban assembly robots for housing construction.
- BuiltWorlds. (2023, September 28). Four failures stunting adoption of construction robotics.
- ConstructConnect. (2022, May 10). A looming construction labor shortage: 41% of workers expected to retire by 2031.
- Davila Delgado, J. M., et al. (2022). Robotics in construction: A critical review of reinforcement and imitation learning paradigms. Advanced Engineering Informatics, 54, 101787.
- Engineering News-Record. (2025). Historical construction cost indices.
- Eurostat. (2025). Construction producer price and construction cost indices overview.
- F-Prime Capital. (2024, January). From surge to sobriety: The state of robotics investment in 2024.
- KFund. (2024, March 12). Where AI meets steel: Robots that learn, think, and work.
- Liu, Y., et al. (2024). Robotics in the construction sector: Trends, advances, and challenges. Journal of Intelligent & Robotic Systems, 110, Article 72.
- Lyu, B., Yin, Q., Tommelein, I. D., Liu, H., Ranka, K., & Shi, J. (2025). A granular framework for construction material price forecasting: Econometric and machine-learning approaches. arXiv.
- McKinsey Global Institute. (2017). Reinventing construction: A route to higher productivity.
- McKinsey & Company. (2024, August 9). Delivering on construction productivity is no longer optional.
- Moshkinitehrani, H., et al. (2025). Streamlining design-to-robotic-production (BIM-to-Bot) for panel framing. Advanced Engineering Informatics, 65, 103393.
- NVIDIA. (2025). NVIDIA Isaac robotics platform.
- Sifted. (2023). Construction robots: Here come the builder bots.
- U.S. Bureau of Labor Statistics. (2025). Construction labor productivity.
- Waqar, A., et al. (2024). Mitigating robotics implementation barriers: A sustainable construction management model. Environmental Challenges, 16, 100989.
- Yue, H., et al. (2025). Interactions between robotics and BIM in construction: A four-layer framework. Computers in Industry, 169, 104299.

Acknowledgements

Thank you to Bram Vanderborght, Jorge Campo, Jonathan Willet, Leonie Krone, Maximiliane Straub, Mollie Claypool, Elisa Seghetti, and Thomas Vandenbergh for their contribution or feedback on our thoughts!