The Forward Deployed *Engineer*

The job is not writing software. The job is standing in the gap between software that works in a demo and a customer whose reality it was never built for, and **closing that gap before anyone loses faith.**

By **Paden Gayle** · Forward Deployed Engineer at Notion
Previously Google and Bloomberg

CONTENTS

01 What the role actually is 3

02 What the work looks like 4

03 How it differs from adjacent titles 5

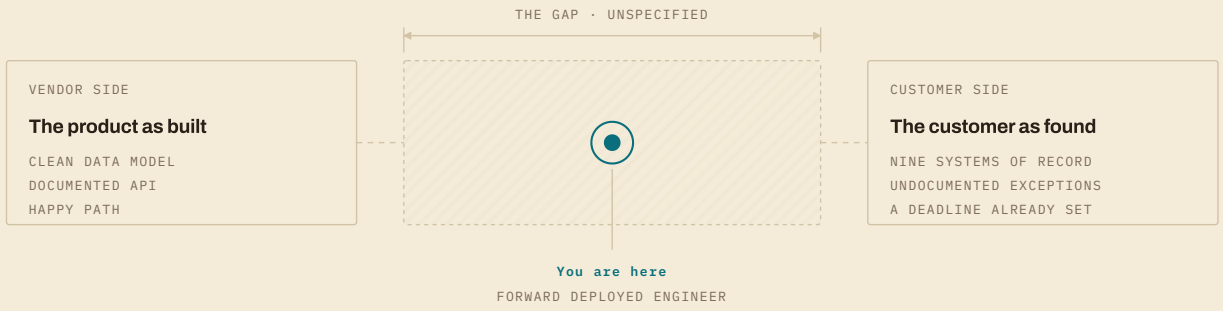
04 What the role pays 6

05 How to get into the role 7

06 Questions people actually ask 8

THE POSITION

Fig. 01: Position of the forward deployed engineer — Not to scale



The forward deployed engineer occupies the gap that specification could not close.

01 What the role actually is

A forward deployed engineer is a software engineer who works inside the customer's environment rather than behind the product, building the integrations, data models, and last-mile software that make a general-purpose platform do a specific organisation's job.

The title came out of Palantir, and the military borrowing is deliberate. A forward operating base is not the headquarters; it is the position closest to the actual terrain. An FDE works the same way. You are employed by the vendor, you carry the vendor's product, and you spend your time in the customer's systems, their constraints, and their politics.

What makes the role distinct is not the technology. It is **who owns the outcome.** A product engineer owns a feature and ships it to everyone. A forward deployed engineer owns whether one named customer succeeds, and has the latitude to write whatever code that requires: scripts, connectors, migrations, internal tools, occasionally an entire application the roadmap will never contain.

> **The demo works because the demo data was built to make it work. Everything after the signature is the job.**
>
> FIELD NOTE

The trade is real. You give up the clean abstractions and the long build cycles. You get proximity: you find out within days whether the thing you built survived contact with the people who have to use it. Very few engineering jobs offer that feedback loop.

That gap in the diagram is where the role lives. It is unspecified by definition. If it could have been specified, it would have been built into the product. The FDE is the organisation's answer to the part of the problem that cannot be generalised.

02 What the work looks like

Job descriptions describe the role in terms of technologies. That is the least useful way to understand it. Here is the shape of the work as it actually runs, roughly in the order it happens on a deployment.

01 **Find the real problem**

The stated requirement is rarely the operative one. Someone's quarterly number depends on this, and that person is usually not in the kickoff meeting. Finding them is the first technical act of the project.

02 **Map the systems as they are**

Not as the architecture diagram claims. There will be more systems of record than anyone admits, at least one critical spreadsheet, and an integration that only one person understands.

03 **Build the smallest thing that proves the path**

Something real in the customer's own data, fast enough that belief survives. This is where FDE work diverges hardest from product engineering: the first deliverable is evidence, not infrastructure.

04 **Harden what survives, delete what doesn't**

Most of what you build early is scaffolding, and knowing which parts to throw away is the skill. Everything that stays has to run without you.

05 **Hand it over and leave**

A deployment that depends on you is a deployment that failed. The exit is part of the design, and if you have done it properly the customer's own team can extend the work after you go.

> **Notice how little of that is about a language or a framework. What separates a good FDE from a fast one is judgment: which problem, which system, and when to stop building.**
>
> FIELD NOTE

03 How it differs from adjacent titles

Most confusion about the role comes from four neighbouring jobs that look similar on a résumé and are structurally different in practice. The distinguishing questions are simple: **do you write production code, do you own an outcome, and who pays for your time?**

TABLE 01 · THE FDE AGAINST ADJACENT ROLES

ROLE	WRITES PRODUCTION CODE	OWNS THE OUTCOME	PRIMARY LOYALTY	MEASURED BY
Forward deployed engineer	Yes, in the customer's environment	Yes, for named accounts	Vendor, deployed to customer	Whether the deployment works
Solutions engineer	Rarely, demos and prototypes	No, owns the sale	Vendor sales org	Pipeline and close rate
Solutions architect	No (designs, does not build)	Partially, owns the design	Vendor	Architecture accepted
Consultant	Sometimes, billable	Contractually, per statement of work	The firm, billed to client	Hours and scope delivered
Product engineer	Yes, in the product	Yes, for all users at once	The product	Feature shipped and adopted

The row that surprises people is the consultant. Both write code in a customer's environment, and the difference is what happens when the project is finished. A consultant's incentive is the next statement of work. An FDE's incentive is a customer who renews without needing anyone sent back. The product is supposed to absorb what you learned.

That is also why the role sits awkwardly on org charts. Engineering thinks it is a customer function, sales thinks it is engineering, and the FDE reports the truth to whichever one will act on it.

04 What the role pays

Forward deployed engineering pays at or above equivalent product engineering levels at the same company, for a reason that is worth understanding: the role is difficult to hire for. It needs someone who can write real software *and* hold a room with an executive who is losing patience. That intersection is thin, and compensation follows scarcity.

$216K
MEDIAN POSTED SALARY, FDE ROLES

$180–250K
TYPICAL PUBLISHED RANGE

Rising
OPENINGS, AS AI VENDORS STAFF DEPLOYMENT

Those figures come from the roles on this site's own board: 109 of the current listings publish a US annual range, and the median midpoint of those is $216K. It is base range as advertised, not total compensation, so equity and bonus sit on top and packages vary widely by company stage and travel load. Only about four in ten employers publish a number at all, which biases the sample toward companies in states that require it.

The structural point matters more than any single number. Every AI company selling to enterprises has discovered the same thing: **the model is not the deployment.** Someone has to sit with the customer and make it real, and demand for people who can do that is growing considerably faster than the supply.

05 How to get into the role

There is no degree for this and no certification worth holding. What there is, is a set of things you can demonstrate, and most candidates fail on the second one, not the first.

01 **Be genuinely able to build**

The bar is real engineering: data modelling, APIs, auth, moving data between systems that were not designed to co-operate. You cannot talk your way through a deployment.

02 **Show you can be trusted in front of a customer**

This is the filter that eliminates most strong engineers. Can you deliver bad news early, say “I don’t know” without losing the room, and disagree with a VP while keeping the project alive?

03 **Have one deployment story you can tell in detail**

Something you shipped for a real user with real constraints, including what went wrong. Interviewers probe for the failure. A story with no failure in it reads as a story you were not actually responsible for.

04 **Aim at companies that need it structurally**

Enterprise platforms and AI infrastructure companies hire FDEs because they must. The role is often titled something else: deployment strategist, solutions engineer, technical account engineer. Read the responsibilities, not the title.

> **Every framework you learn this year gets deprecated. What survives is knowing which problem is worth solving, who in the room actually decides, and when to stop building.**
>
> FIELD NOTE

06 Questions people actually ask

Is a forward deployed engineer a real engineering role, or is it customer success with a better title?

It is a real engineering role. FDEs write and ship production code (integrations, data pipelines, custom applications) inside customer environments. The distinguishing test is simple: if the job does not involve writing software that runs after you leave, it is not forward deployed engineering regardless of the title on the offer letter.

What is the difference between a forward deployed engineer and a solutions engineer?

A solutions engineer supports the sale and is measured on pipeline; the work is largely demos and prototypes, and it usually ends at the signature. A forward deployed engineer starts at the signature and is measured on whether the deployment actually works in production. Solutions engineers rarely own outcomes after close. FDEs own almost nothing else.

Do you need to travel constantly to be a forward deployed engineer?

Less than the name suggests. “Forward deployed” describes organisational position rather than geography, and a large share of the work is now remote, in the customer’s systems and meetings rather than their building. Travel varies by company and by account; some roles are heavy, many are a few trips per quarter for kickoffs and critical moments.

What does a forward deployed engineer earn?

Compensation generally matches or exceeds equivalent product engineering levels at the same company, because the combination of real engineering ability and customer-facing judgment is scarce. Across the roles on this site’s board that publish a US range, the median advertised midpoint is about $216,000 base, with most ranges falling between $180,000 and $250,000. Equity and bonus sit on top of that, and packages vary substantially by company stage and travel load.

How do I become a forward deployed engineer with no FDE experience?

Nobody’s first FDE role is their second one. Everyone arrives from somewhere else, usually product engineering, consulting, or solutions work. What you need to demonstrate is genuine building ability plus evidence that you can be trusted in front of a customer. One deployment story you can tell in detail, including what went wrong, is worth more than any certification.

Which companies hire forward deployed engineers?

Palantir originated the title, and it is now standard at enterprise platform companies and across AI infrastructure: every vendor that has discovered the model is not the deployment. Search on responsibilities rather than titles: the same job appears as deployment strategist, technical account engineer, and solutions engineer depending on the company.

Is forward deployed engineering a good long-term career, or a stepping stone?

Both, legitimately. It is one of the fastest ways to learn how enterprises actually buy, adopt, and abandon software, which makes FDEs unusually well positioned to move into product leadership, founding roles, or field engineering leadership. It is also a career in itself, and senior FDEs who stay become the people their companies send when a deployment cannot be allowed to fail.

Where to go from here

I deploy Notion inside enterprises for a living. Before that, Google and Bloomberg. I am writing the O'Reilly book on this role because when I started doing the work there was nothing to read.

Most of what is written about forward deployed engineering is recruiting copy. The actual craft gets passed around verbally between people who have done it, and never written down: how you run discovery when the stakeholder is hostile, how you scope something you have never seen before, when to build the throwaway version. That is what this manual and the book are for.

THE BOOK

The Forward Deployed Engineer

The full treatment, forthcoming from O'Reilly Media.

THE JOB BOARD

Every live FDE role, updated daily

Gathered from company job boards, checked daily, free to browse. padengayle.com/fde-jobs

THE COMMUNITY

A small group, on purpose

Practitioners comparing notes on deployments that are live right now.

THE FORWARD DEPLOYED ENGINEER — A FIELD MANUAL
Written by Paden Gayle · Revision 2026.1
padengayle.com