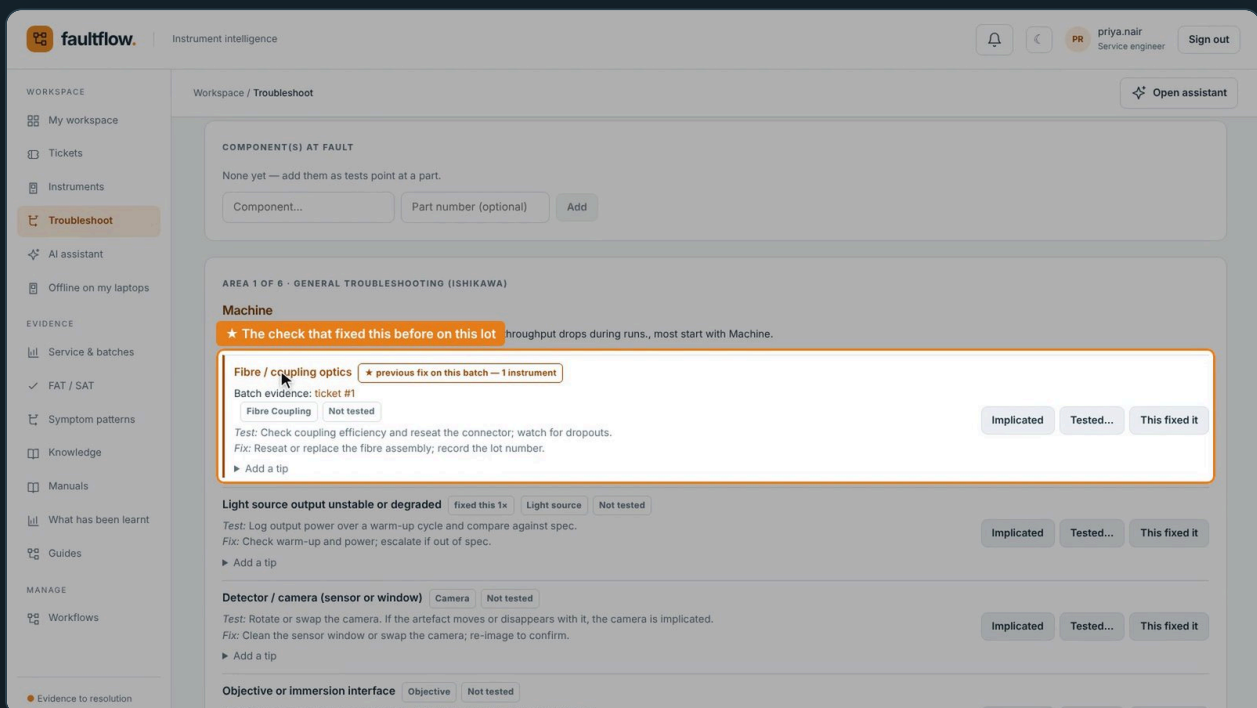


Find the signal. Fix the problem.

How a field service team turns one engineer's fix into the whole team's knowledge, online and offline.

PRODUCT GUIDE · OCTOBER 2026



The fix already exists. It just isn't where the engineer is.

When a complex instrument fails at a customer site, the answer is often already known. It sits in an old ticket, in a build record, or in one experienced engineer's head. So teams solve the same fault again and again, and every repeat visit costs time, travel and customer patience.

25%

of total service cost goes on failed visits at the median service organisation, and 44% for the weakest performers.

9–16 hrs

a week that field engineers at one large equipment maker reported spending looking for information.

37.5%

of Waters' 2025 revenue came from service. For instrument makers, service is a major part of the business.

Sources: Aquant 2026 Field Service Benchmark, February 2026 (161 service organisations, nearly 30 million service events). Earley Information Science, Applied Materials case study (survey of 600+ field engineers). Waters Corporation full-year 2025 results (service sales US\$1,188m of US\$3,165m).

What FaultFlow does

FaultFlow brings an instrument's build history, service tickets, manuals and troubleshooting evidence together, and puts what your team has already learned in front of the next engineer, on the web or on a laptop with no connection.



One fault, three people.

This guide follows the same story as the walkthrough film on the website. A customer's microscope keeps losing light through its fibre. Here is how FaultFlow carries one engineer's fix to the whole team.



Mia

Service manager · web app

Spots that every instrument built with one fibre coupling lot has an open ticket. Drafts a procedure with AI, checks it, and asks Sam to start with the coupling.

Film chapters from 0:19



Sam

Field service engineer · Windows app

Works offline at the customer site. Asks the laptop's own AI, tests the light source and rules it out, then finds the fault in the fibre coupling. Everything syncs when he reconnects.

Film chapters from 4:37



Priya

Field service engineer · web app

A week later, gets a call about another instrument from the same lot. Logs it in her own words, and FaultFlow puts the check that fixed it before at the top.

Film chapters from 8:19

The loop FaultFlow closes

Every pass round the loop makes the next one faster.

1

A ticket arrives

From the service desk, or logged in FaultFlow

2

A pattern shows

Joined to build records across the fleet

3

Troubleshoot

Most likely fix first, online or offline

4

Record the fix

Tests and evidence, synced back

5

Publish a guide

Built from real investigations

6

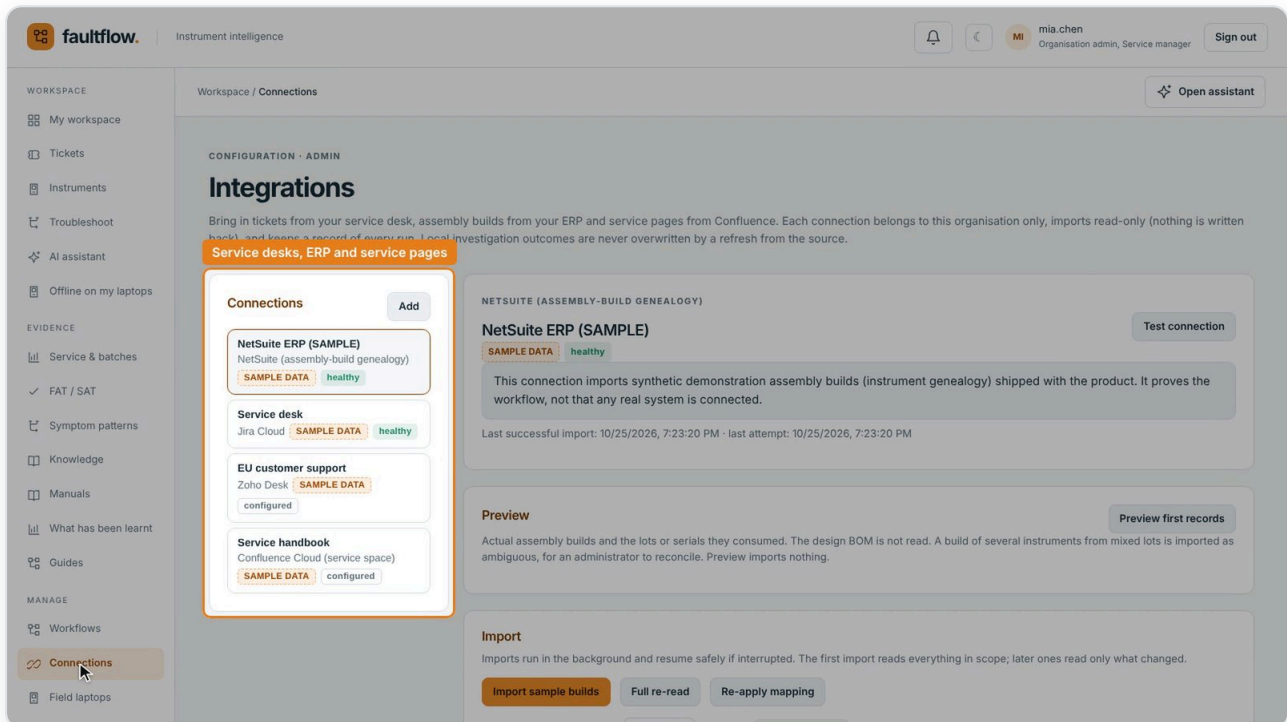
Next engineer

Gets the fix first, and adds to it

About the screens in this guide. They are real FaultFlow screens running on demonstration data: the TX-200 instrument family, its customers and engineers are all fictional.

Everything about the instrument, in one place.

FaultFlow connects to the systems you already run, and joins it all up around the physical instrument and its serial number.



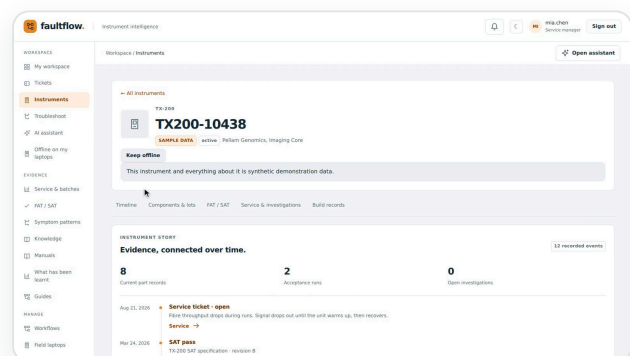
Connections: a NetSuite ERP, a Jira service desk, Zoho Desk and a Confluence service handbook.

Connections

- **Service desks:** tickets from Jira Cloud and Zoho Desk, with your own field mapping.
- **ERP:** assembly build records from NetSuite: which parts and lots went into each instrument.
- **Knowledge:** Confluence service pages, plus manuals you upload.
- **Directly:** engineers log a ticket in their own words, with files attached.

The instrument's whole story

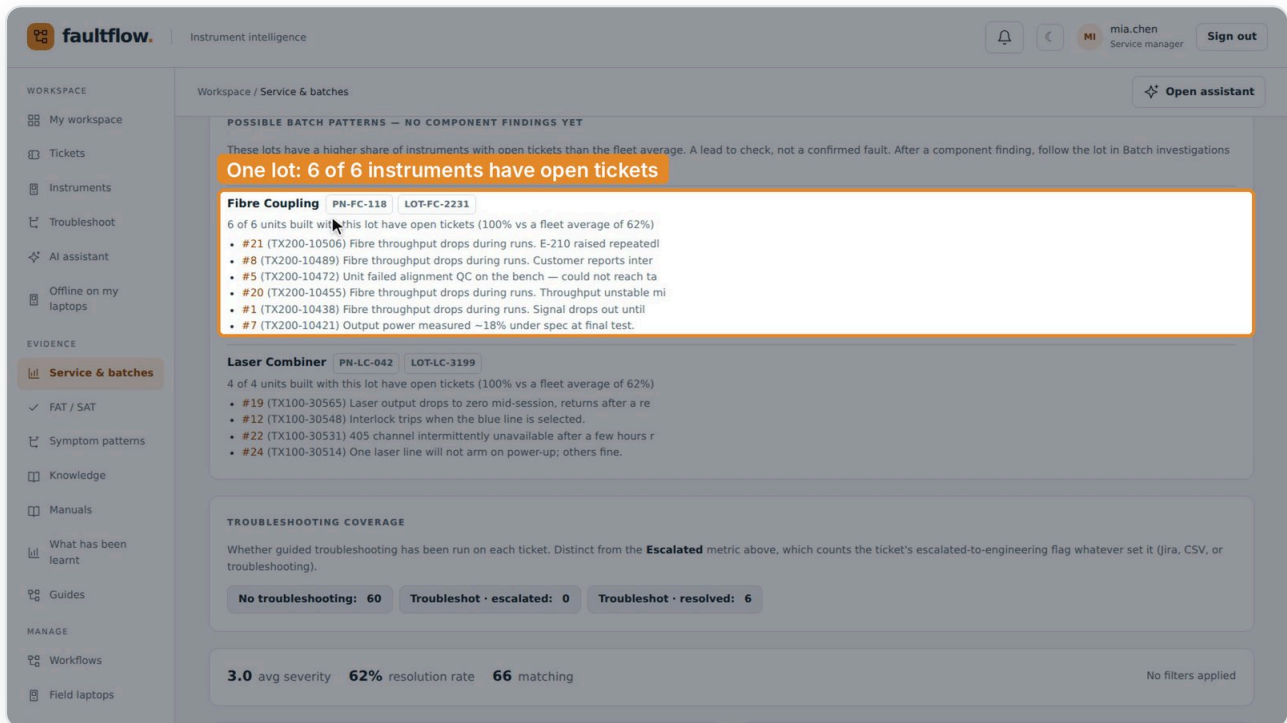
Each instrument has a timeline of builds, part installations, factory and site acceptance results, tickets and investigations, so a fault can be read against the parts fitted at the time.



An instrument's record: its story, components and lots, acceptance results and service history.

Patterns found in the data, not in someone's memory.

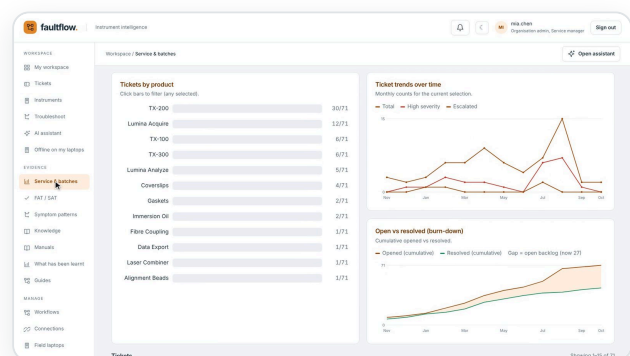
Because every ticket is joined to how its instrument was built, patterns across the fleet show up on their own.



Service & batches: every instrument built with fibre coupling lot 2231 has an open ticket, against 62% across the fleet.

From a lead to a confirmed finding

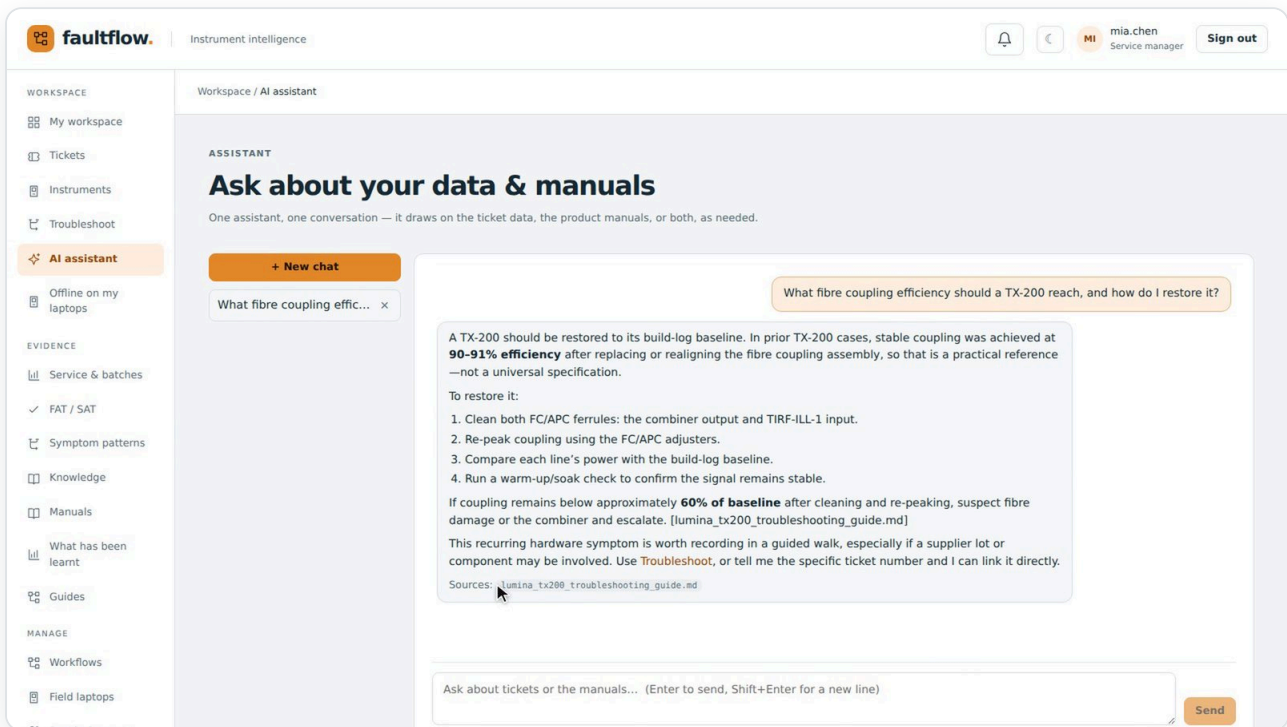
- **Possible batch patterns** flag lots with a higher share of open tickets than the fleet: a strong lead to test.
- **Batch findings** follow once an investigation confirms the faulty part. The other instruments built with that lot are flagged, with their open tickets.
- **Factory to site:** compare acceptance results, such as coupling efficiency, between factory and customer site.
- **Dashboard:** tickets by product, trends over time and open against resolved, all filterable.



Tickets by product and trends over time, filtered with a click.

AI that shows its sources. Procedures your experts own.

AI in FaultFlow helps people find and write things down. Your experts stay in charge of what the team actually does.



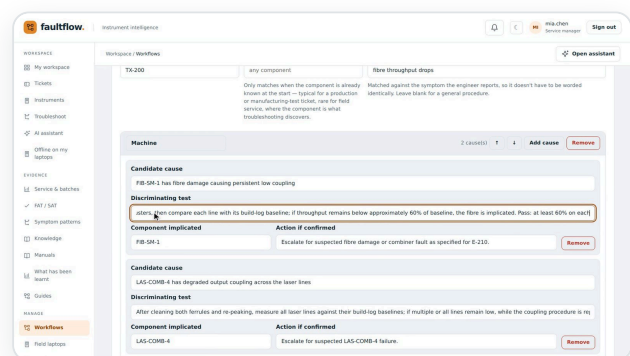
The assistant answers from the troubleshooting guide and past cases, makes clear which is which, and names its source.

The assistant

Ask about tickets, instruments or manuals in plain English, on its own page or alongside any screen. Every answer names its sources.

Procedures drafted by AI

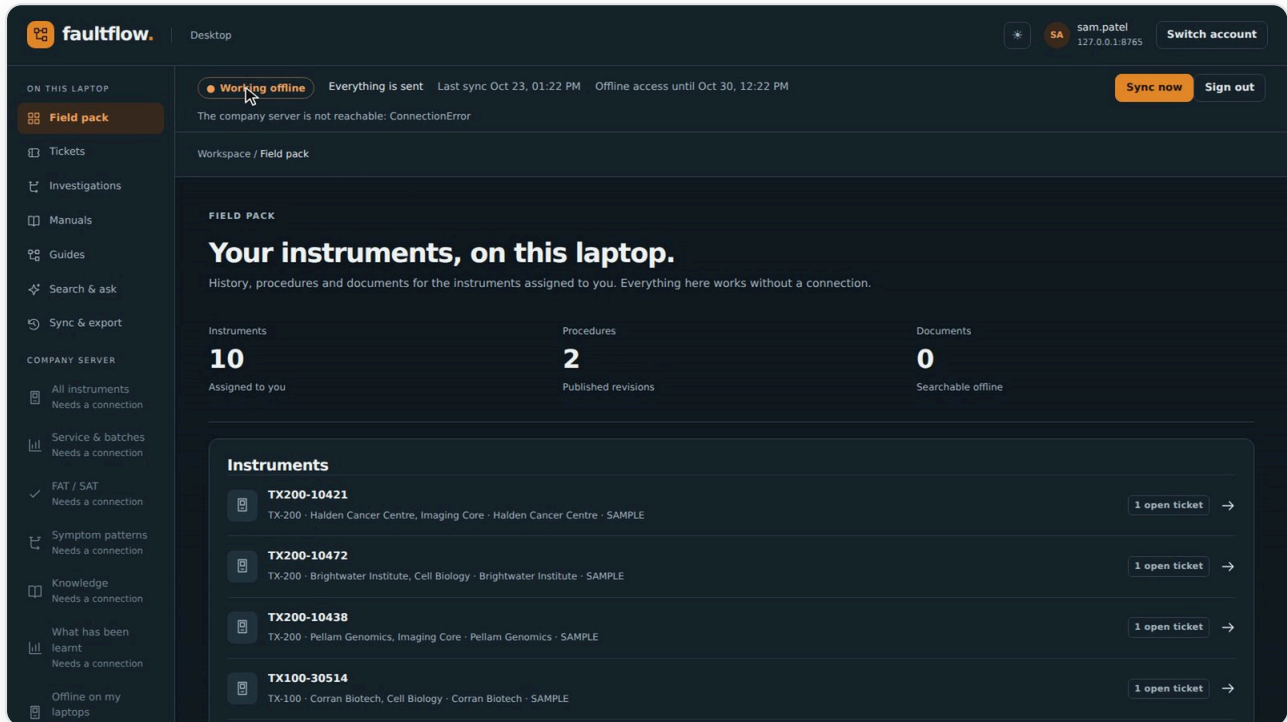
AI reads your manuals and drafts a troubleshooting procedure: the likely causes, each with a test to confirm or rule it out. Your expert edits every step, adds what the team has learned the hard way, and decides which tickets it applies to. Teams can also build procedures from scratch, as a fishbone or an ordered checklist.



An AI draft under review: each cause has a test, the part implicated and the action to take.

Keeps working where the network doesn't.

The FaultFlow Windows app downloads everything an engineer needs before the visit. When the connection drops, the work carries on.



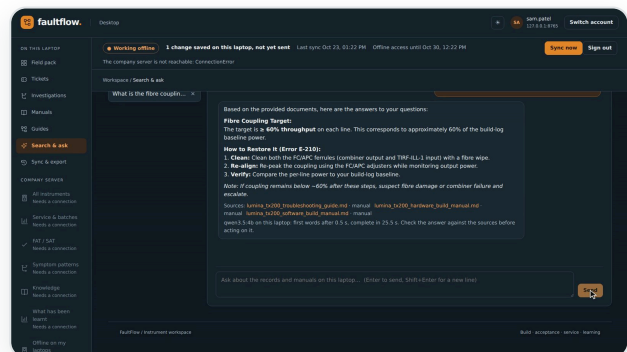
Working offline: the instruments Sam looks after, with their tickets, history, procedures and documents, held on his laptop.

Before the visit

- **Field packs:** managers assign instruments to engineers, and everything about them downloads.
- **Keep offline:** engineers add any ticket, manual, guide or procedure they want with them.
- **Questions travel too:** a colleague's question can be answered on the spot.

AI on the laptop

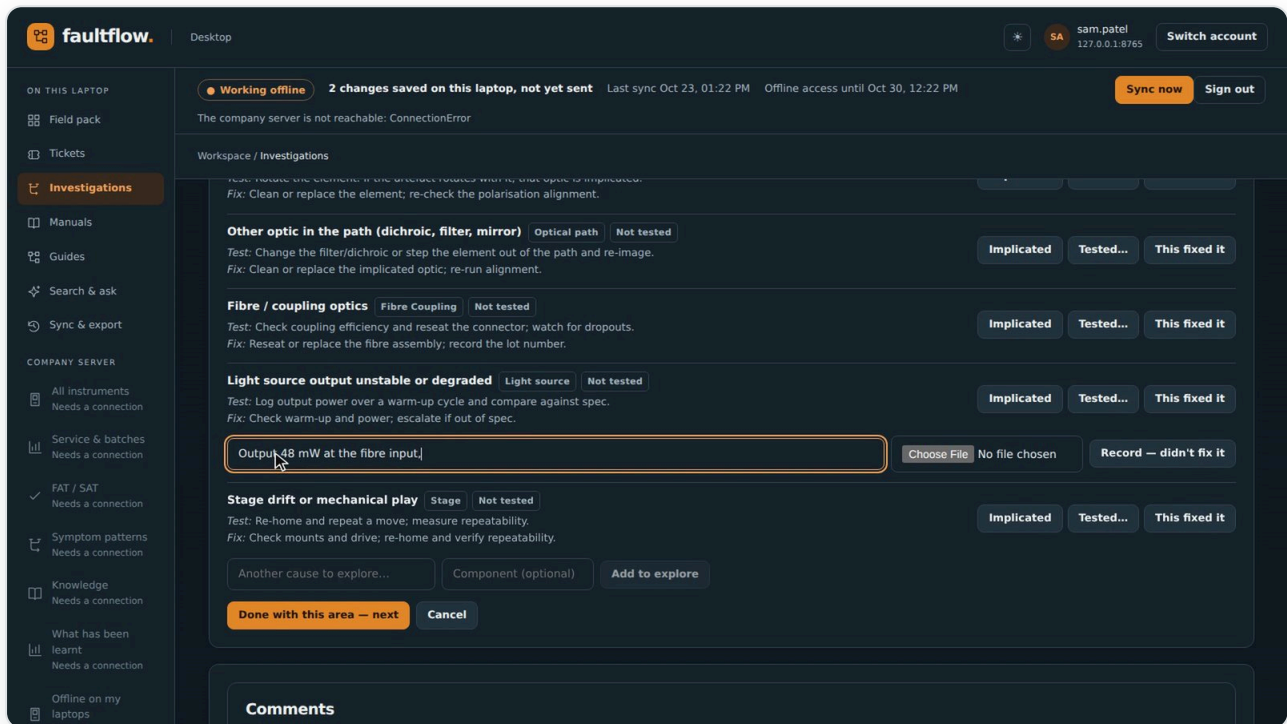
A local AI model runs entirely on the laptop, reads only the manuals stored there, and names its sources. Nothing leaves the device.



Search & ask offline: the answer, its sources, and how long the laptop took.

Troubleshooting that records the evidence.

Engineers work through the same fishbone diagram on the web and on the laptop, and every test they run becomes evidence for the next person.



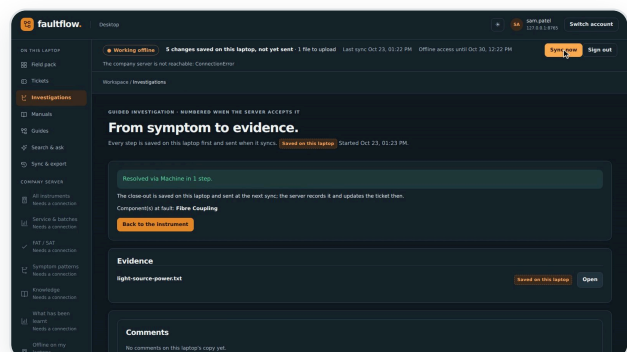
Sam tests the light source, records the reading and attaches the power measurement.

Every step is recorded

- Each cause has a test and an action. The engineer records what they saw and attaches evidence.
- **Didn't fix it** rules a cause out; **This fixed it** closes the investigation with the part at fault.
- A cause the procedure missed can be added on the spot.

Back in sync

Work is saved on the laptop first, then sent when it reconnects. The ticket, the answer to the question, the faulty part and the evidence all arrive at the office together. Anything that changed in the meantime is held for review rather than overwritten.



Resolved offline, saved on the laptop, ready to send.

FaultFlow counts what actually fixed each problem. From that evidence, a manager publishes a guide for the next engineer.



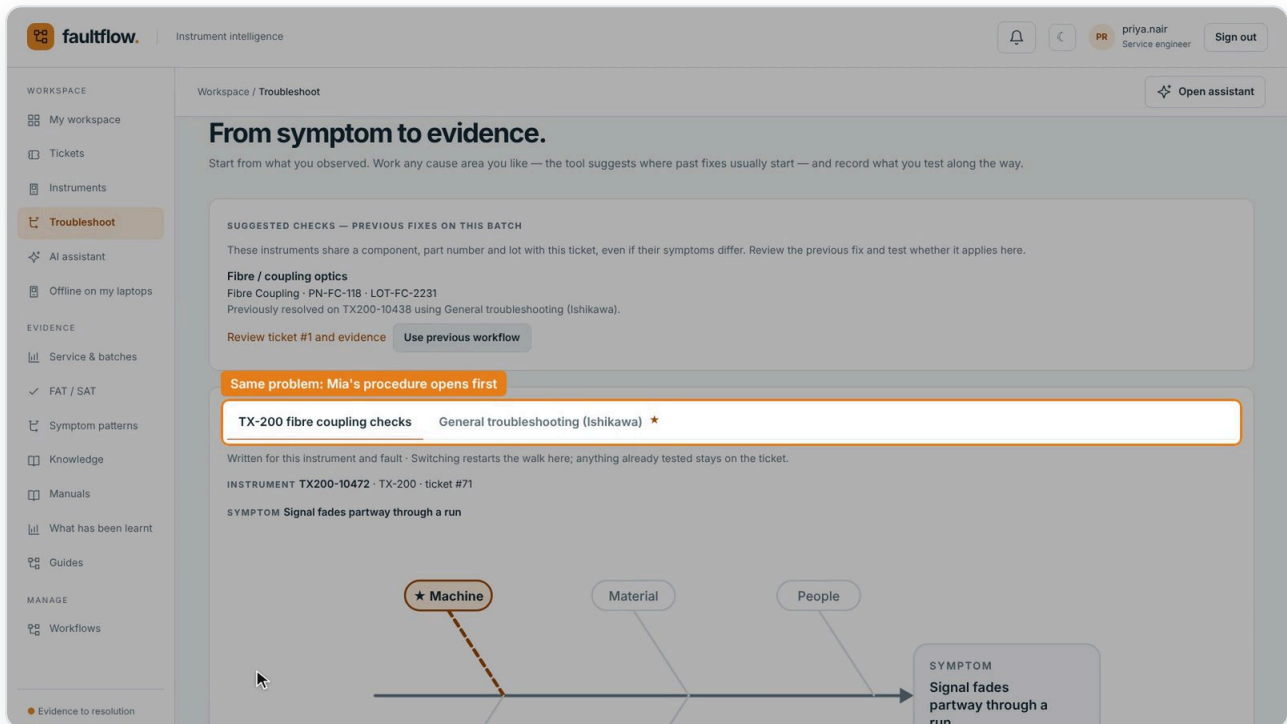
For each problem and product: the checks that fixed it, the checks that were ruled out, the usual order, and how long it typically took, each linked to the investigations behind it.

- **Draft a guide** lays out the checks in the order that works best, with the numbers behind each.
- AI can tidy the wording. The facts stay those of the investigations.
- Guides go through review before publishing, keep their version history, and flag when newer evidence arrives.
- Engineers can add attributed tips beside any check.



The next engineer gets the fix first.

A week later, Priya logs a new call in her own words. She doesn't have to match anyone else's wording: FaultFlow recognises the same problem.



Same problem: the procedure Mia wrote for it opens first, and her published guide is right there too.

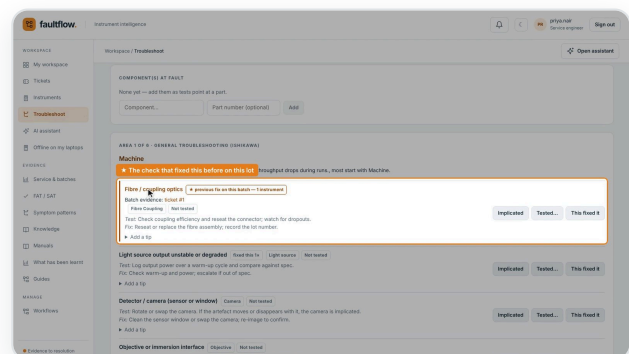
The most likely fix at the top

When an engineer starts troubleshooting, FaultFlow opens the procedure written for that problem, lists the published guide, and in the general fishbone puts a star on the check that fixed this before on this lot.

It's ranked by what actually fixed the problem, not by guesswork. The full diagram is always one click away, so the engineer stays in charge.

And it keeps getting better

Priya's fix is recorded like Sam's, so the next suggestion is stronger still.

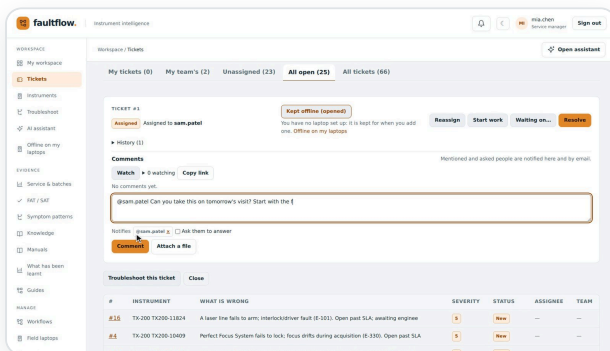


The star: the check that fixed this before on this lot.

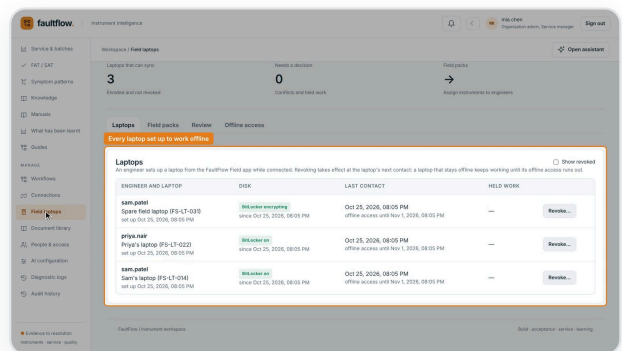
What each person gets.

Around the troubleshooting loop is what a team needs every day. People see and change only what their roles allow.

WHO	WHAT THEY USE
Field engineers	A home page with their next steps · tickets assigned to them and their team · troubleshooting with the most likely fix first · the assistant · manuals and guides · the Windows app with offline work and local AI
Service managers	Assigning tickets to engineers and teams · comments, questions and email notifications · the service dashboard and batch patterns · procedures and guides · field packs for each engineer
Engineering and quality	Instrument history, parts and lots · factory and site acceptance results compared · confirmed component findings · what has been learnt for each problem
Administrators	People, teams and roles · connections to your systems · document library · AI settings · email · field laptops and offline access · audit history of every change



Ask a colleague on the ticket: the question stays open until they answer, and they're emailed too.



Every laptop set up to work offline, with its disk encryption and offline access.

Built around your organisation.

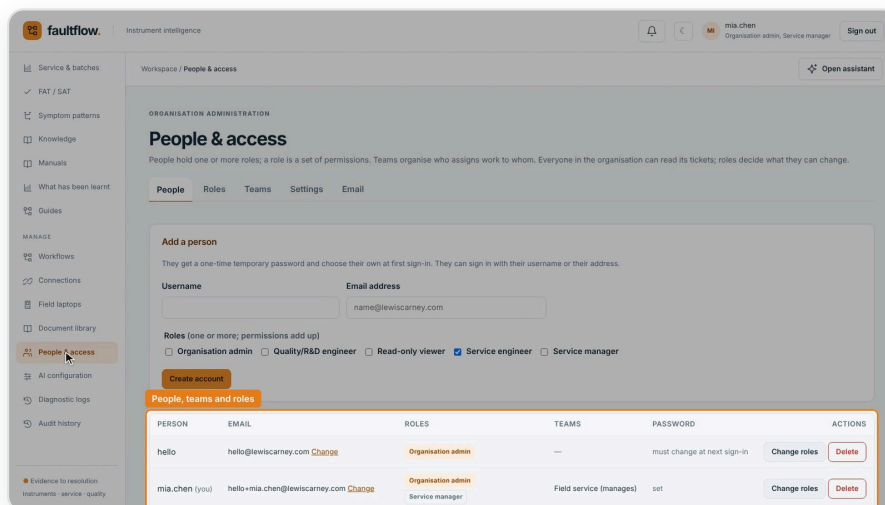
FaultFlow fits the systems and working practices you already have, rather than asking you to replace them.

Your data stays yours

- Each organisation has its own data, people and roles.
- It uses your own approved AI provider, so service data goes only where you have agreed.
- It can run on your own infrastructure.
- Field laptops report their disk encryption, have time-limited offline access, and can be revoked.
- Every change is recorded in an audit history.

Fits what you run today

- Service desks: Jira Cloud and Zoho Desk.
- ERP build records: NetSuite.
- Knowledge: Confluence and your own manuals.
- Other systems can be connected as part of setting up.



People & access: roles and teams decide what each person sees and changes.

How we'd start

I'm working with a small number of instrument companies first, on site and alongside your team.

One instrument family
and the systems its service data
lives in today

One recurring problem
that your team already
recognises

Agreed measures
so we both know what better
looks like

If you lead service, engineering or operations for complex instruments, let's talk.

lewiscarney.com/contact.html