



Every number has a parent.

Provisor

The decision layer for order promising and stock policy — native to Salesforce, EU-hosted, worldwide.

Promises you can't keep, stock you can't move

An order date gets quoted from a single fake-precise number — or a spreadsheet, or a tool that won't say why. The outcome is the same: orders ship late, and capital sits in stock that never needed to be there.

The gap

Between a real order and a kept promise sits one decision — source, allocate and time supply against demand, under uncertainty. Most tools either automate it opaquely or leave it to a planner and a spreadsheet.

TODAY

“It’ll be ready the 14th.”

At risk — 120 u short on 14 Nov

One date. No range, no source, no reason.

One engine, two lenses

Provisor keeps one picture — supply versus demand over time — and reads it two ways. Same model, same solver, same data; a different question each time.

—

01

Promising

Turn a real order into a dated fulfilment plan and the supply orders it needs.

—

02

Policy

Set reorder point, order quantity, safety stock and default mode against a service-vs-cost trade-off.

—

03

The loop

Late or short promises flag a policy to revisit; policy reshapes the supply the next promise works against.

The same picture, read two ways

Promising

A realized order, near horizon → actions now: a dated plan + the orders to emit.

in: S0-10482 · 1 200 u · due 14 Nov
out: plan + supply orders

Policy

A forecast, long horizon → standing rules: reorder point, quantity, safety stock, mode.

in: demand forecast + lead times
out: reorder 880 · OQ 1 200 · SS 320

The planner's day starts here

Sales orders waiting to be promised (6)

Sort by Requested date ▼

Order	Customer	Item	Qty	Requested	Coverage check	
SO-10471	Arvo Dairy	SN-08 Seal kit	2,500	Fri 2 Oct	⚠ Short 860 by requested date	<button>Run promise</button>
SO-10482	Nordhaven Foods	VA-220 Valve assembly	1,200	Wed 14 Oct	⚠ Short 400 by requested date	<button>Run promise</button>
SO-10479	Teodora Bakeries	PM-14 Pump module	60	Fri 9 Oct	✅ Covered from stock	<button>Run promise</button>
SO-10477	Kessel Brewing	VA-220 Valve assembly	300	Tue 20 Oct	🕒 Needs incoming PO-7731	<button>Run promise</button>
SO-10474	Brightwell Labs	FL-30 Filter cartridge	4,000	Mon 26 Oct	✅ Covered from stock	<button>Run promise</button>
SO-10466	Marlow & Finch	PM-14 Pump module	240	Mon 2 Nov	🕒 Needs new supply order	<button>Run promise</button>

Coverage check is a quick look at stock and incoming only. Run a promise to get a full plan you can compare and commit.

Orders waiting to be promised, each with a one-glance coverage read — short, covered from stock, or waiting on an incoming order — before any plan is run.

Promising

A real order becomes a dated **fulfilment plan** — tranches of quantity, source, date and a **reliability tier** — plus the supply orders to emit. Every tranche shows where it comes from and how sure the date is. Nothing is fake-precise.

Reliability — never colour alone



Each tier is pattern + shape + word: ● **Confirmed**, ● **Scheduled**, ◌ **Estimated**.

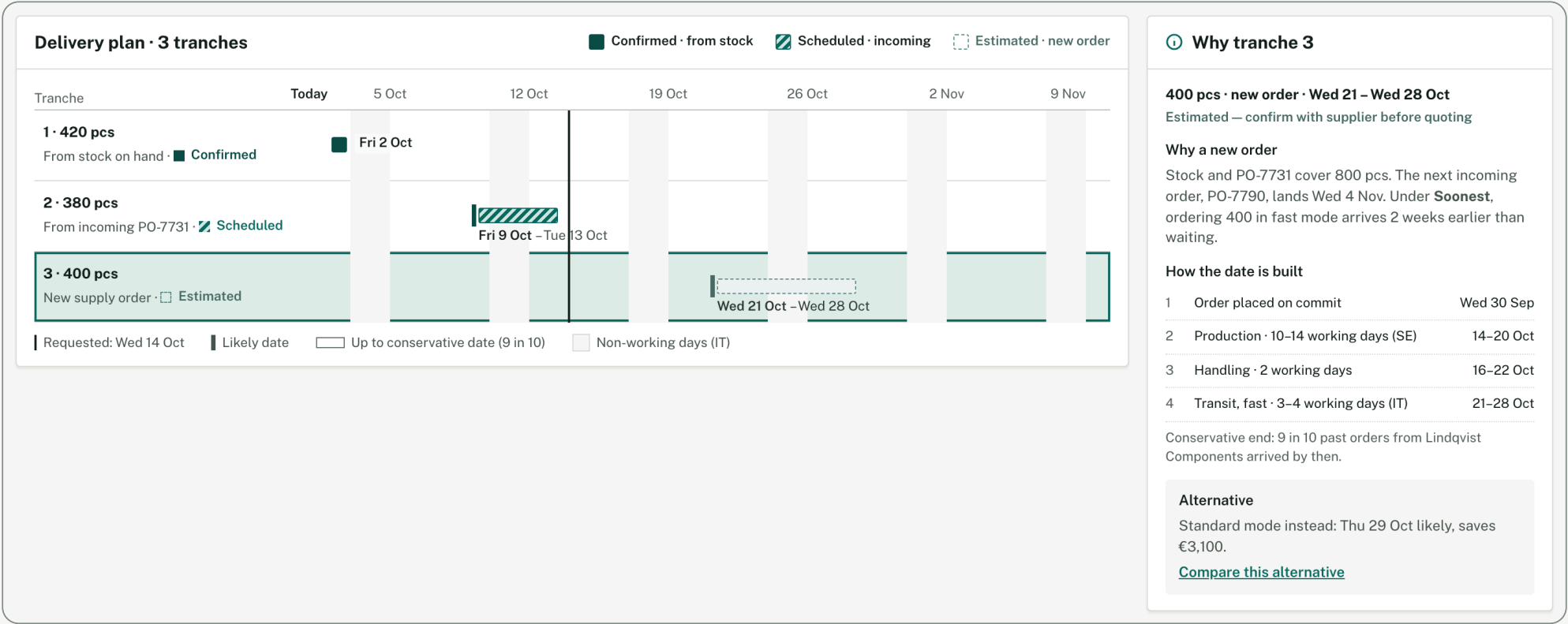
FULFILMENT PLAN — SO-10482 • 1 200 u

● Confirmed	420 u • on hand • 2 Oct
● Scheduled	380 u • PO-7731 • 9-13 Oct
◌ Estimated	400 u • new order • 21-28 Oct

emit: PO • 400 u • fast • ready 16-22 Oct

Promising · one order

A promise you can explain



Tranches by source, date and reliability tier; a timeline from likely to conservative; and a why panel that traces every date back to its parent.

Policy

The same engine, a forecast instead of one order: it recommends **reorder point**, **order quantity**, **safety stock** and a default mode, and shows the **service-vs-cost** trade-off behind the number — as a range, not a point.

Honest uncertainty

A likely value inside its range beats a single confident guess that turns out wrong.

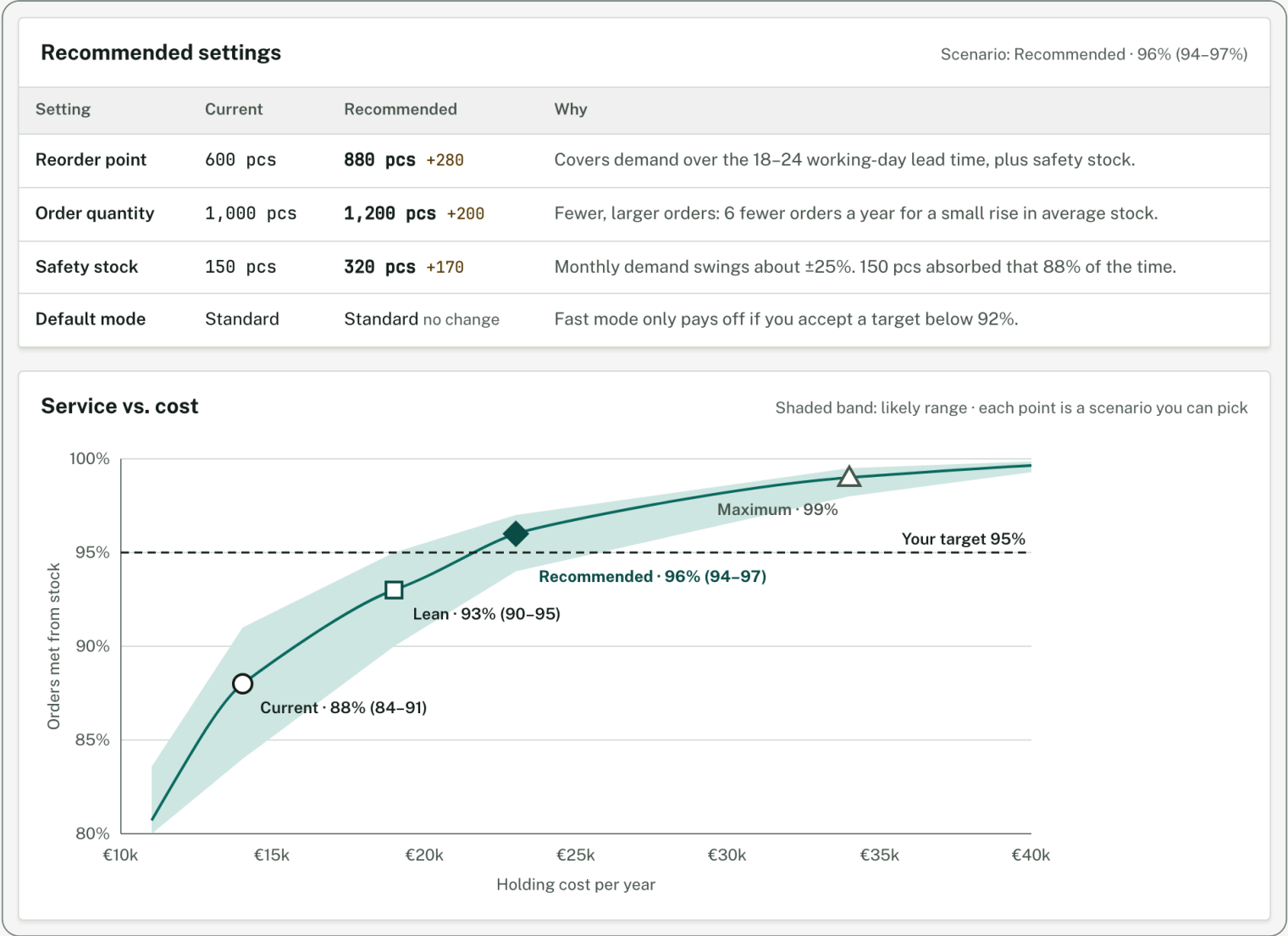
STOCK POLICY — item VA-220

	now	recommended
reorder point	600	880
order qty	1 000	1 200
safety stock	150	320

service vs holding cost



Service versus cost, made visible



Recommended reorder point, order quantity and safety stock — each with its reason — on a service-vs-cost curve where every point is a scenario you can pick.

Decision support, not a black box

Human in the loop

Provisor proposes; a person commits. Preview → refine → commit. Nothing is real until accepted.

Comparison

Objectives and scenarios side by side, on one axis.

Legibility over magic

Every number traces to a source and a reason. No unexplained scores.

One surface, two lenses

The promising-policy loop is visible and actionable.

Honest uncertainty

Ranges and reliability tiers, never fake precision.

Every number has a parent

On hand 420

P0-7731 380

New order 400

— **S0-10482 · 1 200 u**

Compare

Objectives, side by side

Objective	A · baseline Soonest	B Fewest new orders	C Lowest cost
Delivery timeline 30 Sep – 19 Nov · line = requested			
Fully delivered likely – conservative	Wed 21 Oct – Wed 28 Oct ✓ Best	Fri 6 Nov – Wed 11 Nov +16 days vs A	Thu 12 Nov – Thu 19 Nov +22 days vs A
Late pieces	400 · 7–14 days late ✓ Best	400 · 23–28 days late	400 · 29–36 days late
New supply orders	1 · 400 pcs, fast mode	0 ✓ Best uses incoming PO-7790	1 · 400 pcs, economy mode
Fulfilment cost	€9,800	€4,700 –€5,100 vs A	€4,100 ✓ Best –€5,700 vs A
Reliability	1 estimated tranche · to confirm	All confirmed or scheduled ✓ Best	1 estimated tranche · to confirm
Effect on other orders	None	Takes 400 of PO-7790: SO-10466 moves 3 days later	None
Deliveries	3	3	2 ✓ Best stock + PO-7731 ship together
On time by 14 Oct	800 of 1,200 pcs	800 of 1,200 pcs	800 of 1,200 pcs
	Open Accept A...	Open Accept B...	Open Accept C...

The same order under three objectives on one axis — dates, late pieces, cost and reliability, with every knock-on effect spelled out — so the trade-off is a choice, not a guess.

Onboard any org — by configuration

A declarative mapping reads each org's own objects and terms into Provisor's neutral model. No bespoke code per customer; a new company is a new **spec**, not a new build.

Posture

Multi-tenant and worldwide, with **EU data residency** — the engine runs in Frankfurt, each tenant isolated.

Declarative spec per tenant

No code change to onboard

Multi-tenant · row-level isolation

EU data residency · Frankfurt

Where it runs

Provisor is **native to Salesforce** today — a planner’s cockpit, with an Agentforce entry point — and a standalone web app later. The engine runs in the EU.

Residency

Hosted in the EU (Frankfurt); no data leaves the region.

Salesforce-native cockpit

Agentforce entry point

Standalone web app — later

Hosted in the EU · Frankfurt

Every number has a parent.