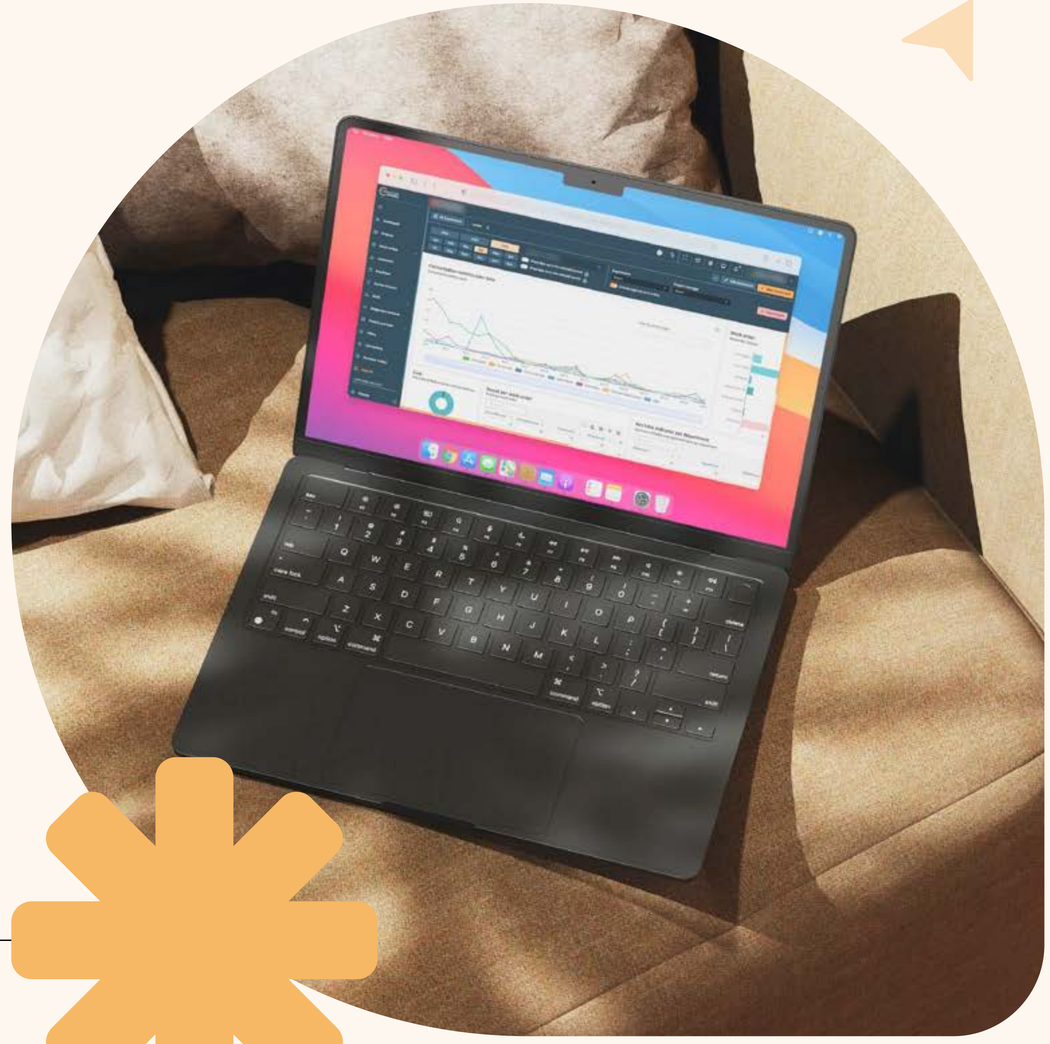
corzent x CONTRACTING WORKS

Smart Planning

Employee Suggestion Engine

Intelligent, context-aware scheduling
for field service operations

The Contracting Works Story



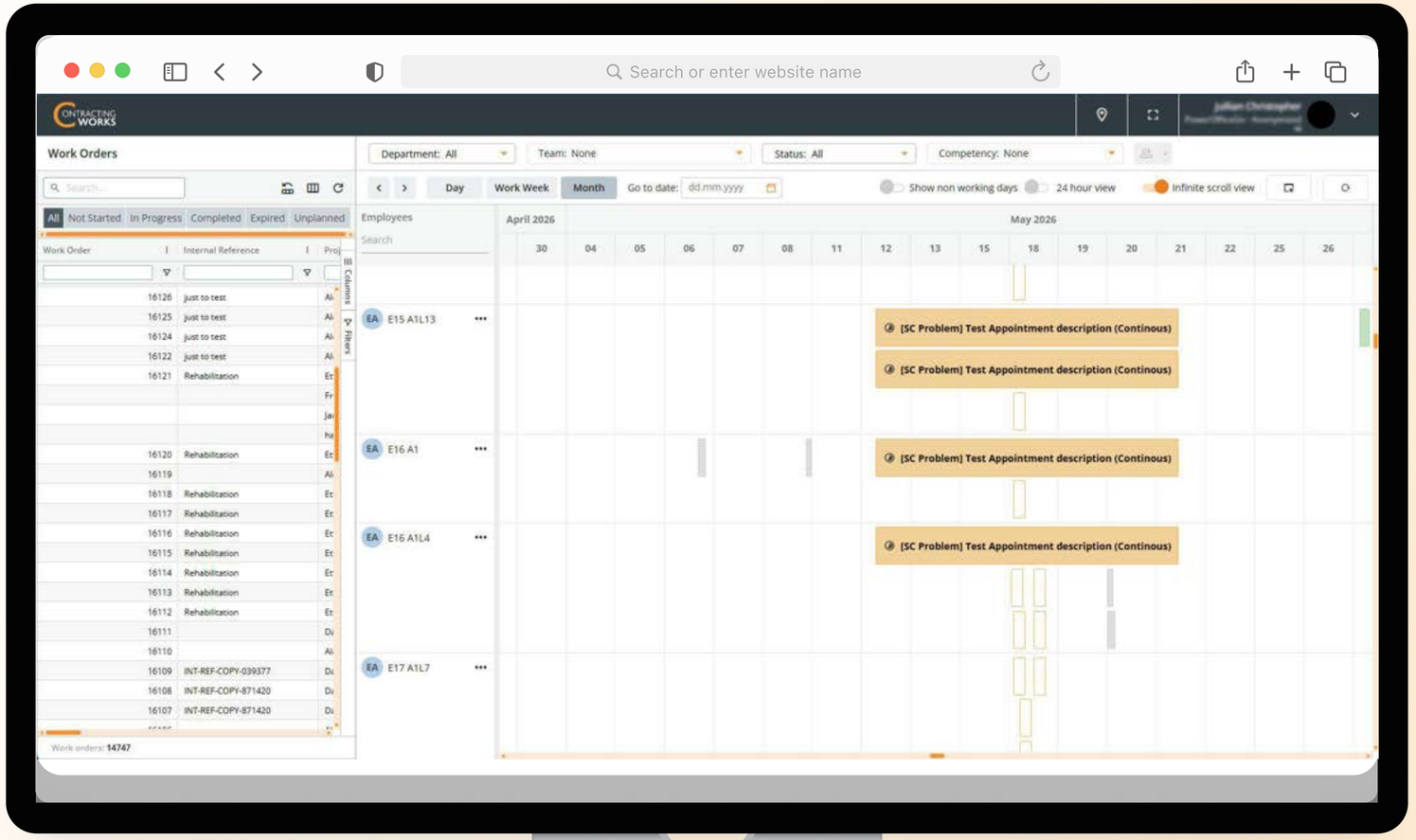
Feature Snapshot

At-a-glance overview for technical and non-technical stakeholders

Smart Planning automatically surfaces the right field employees with optimal time slots for a work order, based on live scheduling gaps, geo-proximity, work hours, and competency filters, all in a single click.

Module	Smart Planning - Employee Suggestions
Host Application	Contracting Works Planner App (calendar scheduling board)
Entry Point	Work Order → Appointments → Get employee suggestion → Smart Planning dialog
Trigger	User fills filters and clicks 'Apply Filters'
Backend Call	A single server-side call returns ranked employee suggestions
Output	Employee cards with ranked availability windows
Terminal Action	Click 'Create and Close' → Appointments created and visible in Planner App calendar

Platform context: Smart Planning is built as an intelligent enhancement to the existing Contracting Works Planner App, a full calendar scheduling board where planners manage all employee appointments across Day, Work Week, and Month views. The Planner App is the primary workspace planners already use daily; Smart Planning adds a suggestion layer accessible directly from a Work Order without leaving the app.



The Challenge

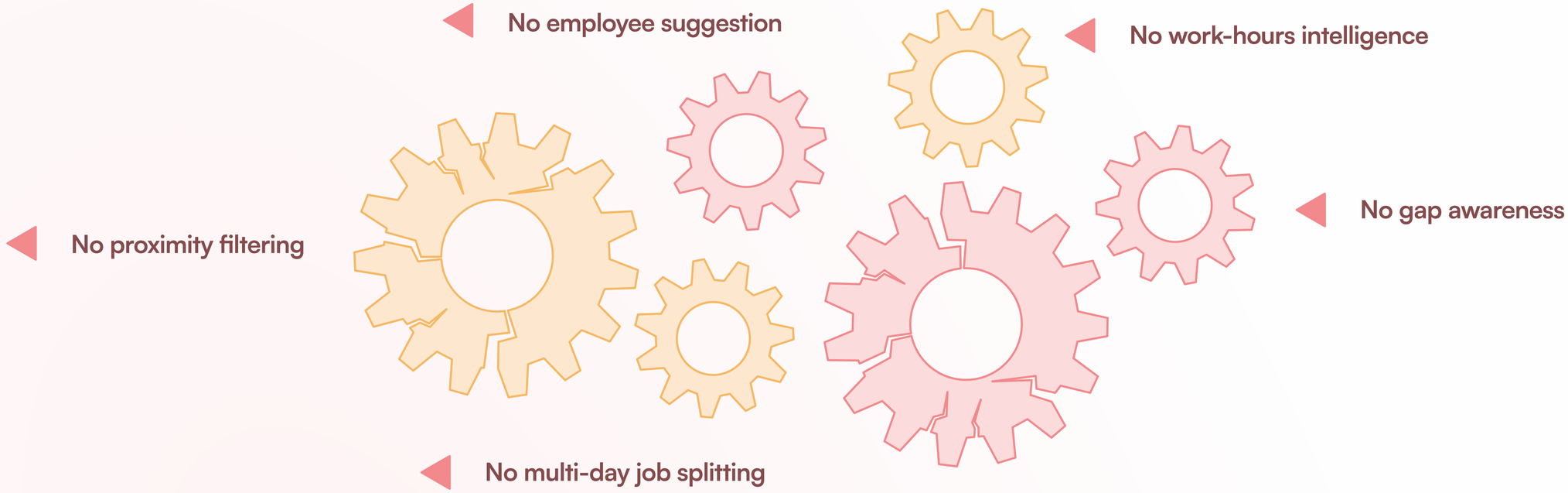
Why the existing Planner App wasn't enough?

CW already had a Planner App, a calendar-based scheduling board giving project managers a Day, Work Week, and Month view across all employees. The app shows each employee as a row with their appointments on a timeline, supports filtering by Department, Team, Status, and Competency, and allows appointment creation directly on the calendar by dragging or clicking a time slot.



What the Planner App Could Not Do

The app gave planners a visual schedule, but left the whole job of finding the right employee and the right slot on them.



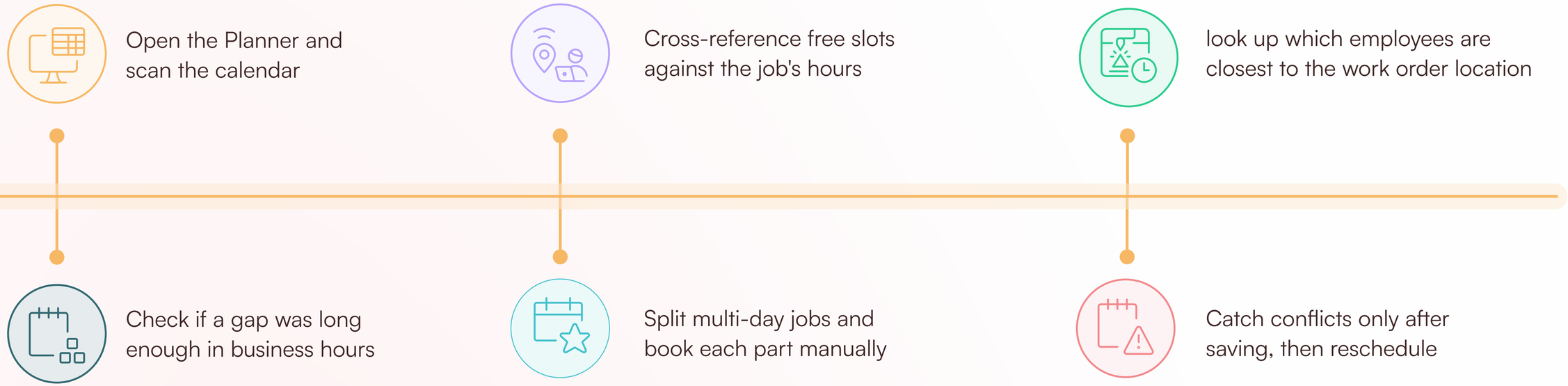


Why this mattered

With a growing workforce and increasing work order volume, the time cost of manually identifying the right employee, validating availability, checking proximity, and handling multi-day splits did not scale.

The 'Before Smart Planning' Workflow

Prior to Smart Planning, a planner scheduling a new work order would:



The Origin Story

How the Need was Identified?

01 The tool showed, but didn't guide

Planners used the calendar as a lookup, scrolling employee rows and judging slots themselves, with no guidance from the system.

02 Scheduling was inefficient

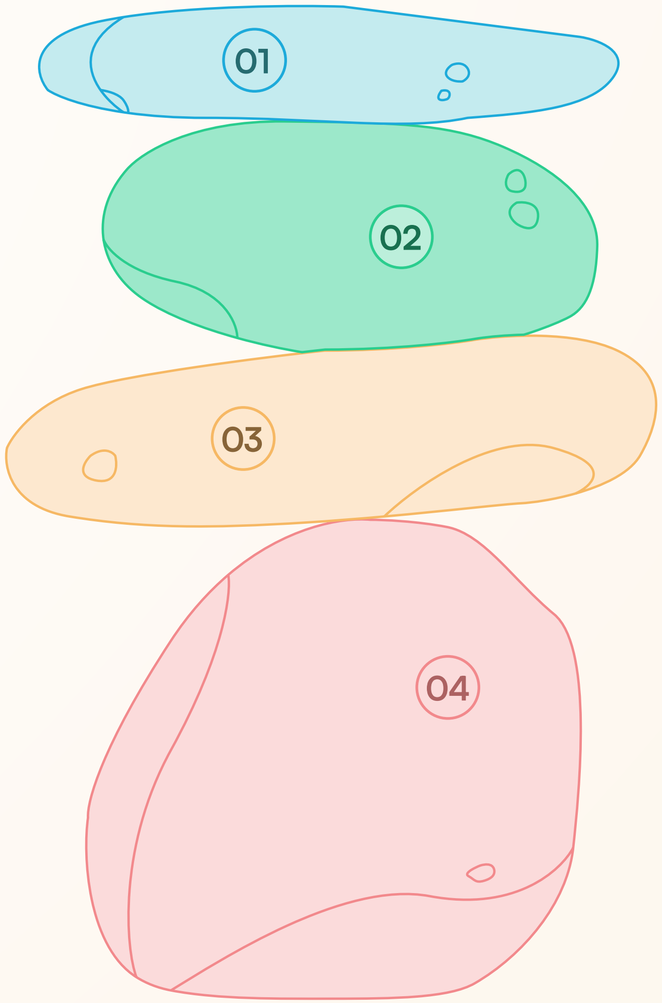
CW's operations team raised it as a top pain point: too much of the day went to deciding who works when, not to the work itself.

03 The core question had no automated answer

Who do I assign this 6-hour job to, starting today?' was solved by intuition, and that stopped scaling as volume grew.

04 The solution was already in reach

Every input needed, schedules, geo-coded addresses, working hours, teams, competencies, was already in the system, just never surfaced as suggestions.



Why We Decided to Build It

The intersection of three factors made this the right investment at the right time:

Existing platform foundation:

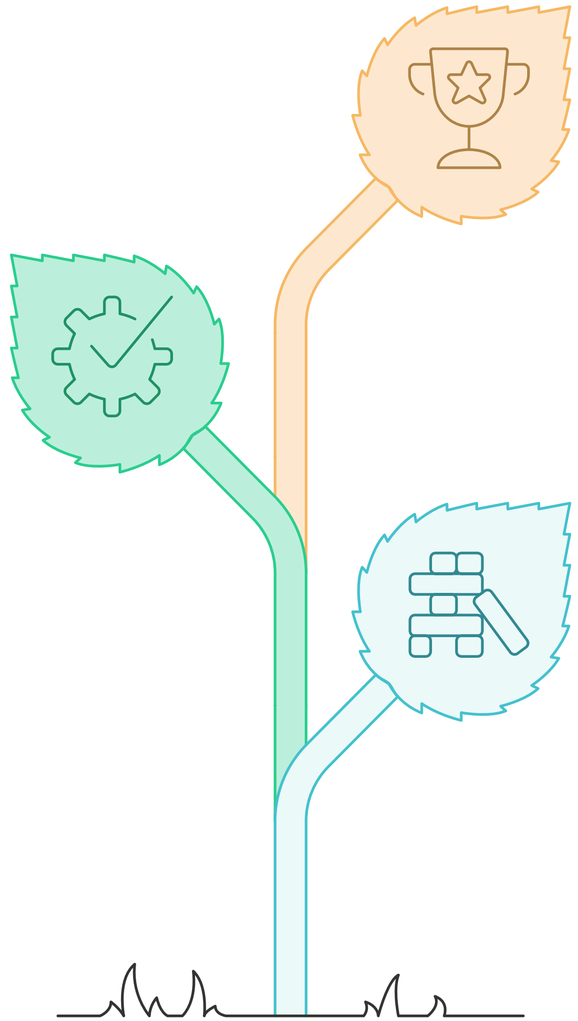
Smart Planning is an intelligent layer on top of the Planner App, the appointment model, employee data, and address geo-coding were already in place, so it needed no new data infrastructure.

High ROI within existing workflow:

Planners already lived in the Planner App. Launching suggestions as a dialog from the work order meant zero disruption, it plugged straight into their existing motion.

Competitive differentiation:

Intelligent, context-aware scheduling is a hallmark of mature FSM platforms. Building it natively cut CW's reliance on manual workarounds and moved the platform closer to market leaders.



The Solution

End-to-End Flow

Open the dialog

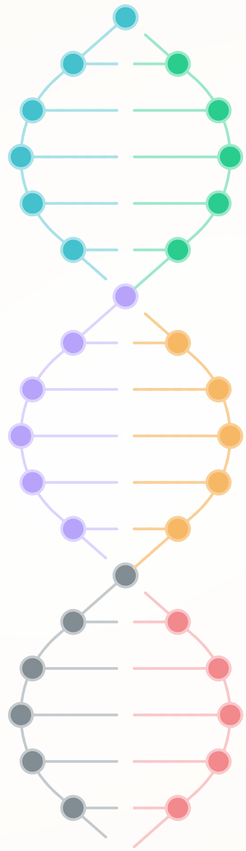
Launched from a work order; its address is pre-loaded as the origin for proximity

Engine executes

'Apply Filters' runs a single server-side call that returns ranked suggestions.

Select a slot

The chosen slot appears in a confirmation panel before any records are created.



Set filters

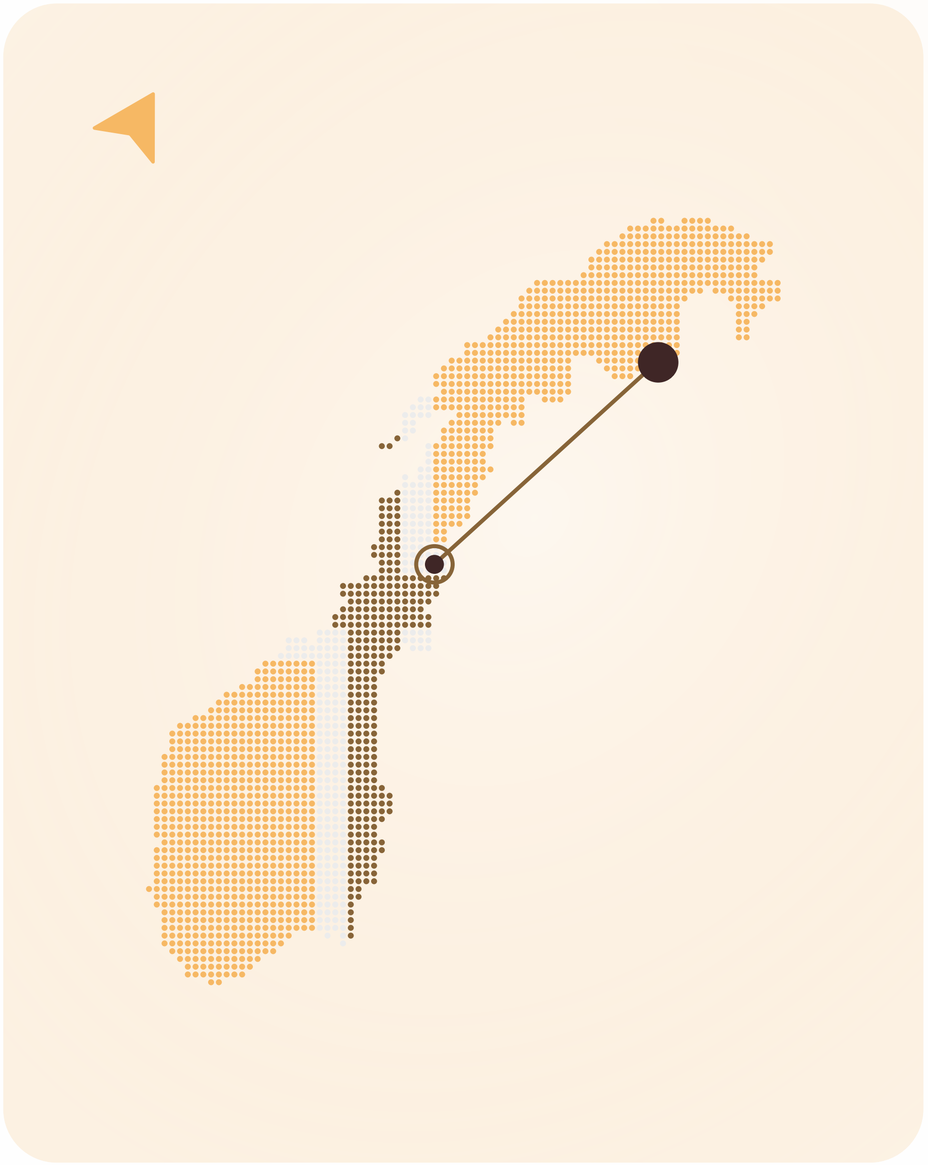
Start date, optional end date, search radius, and optional team, department, and competency filters. Custom hours can override the work order's.

Results returned

Results returned as ranked employee cards. Each card shows: distance (km), team membership tags, available dates, and number of appointments required.

Appointments created

Appointments created. Clicking 'Create and Close' creates the appointments in the system. Immediately visible in the Planner App calendar.



Geo-Distance Filtering

Geo-proximity is calculated natively within the system. Distance from the employee's effective address (previous appointment's site or home address, depending on date) to the work order's address is compared against the radius parameter, eliminating the need for an external mapping service during suggestion generation.

What Makes It Unique

Gap-first scheduling

The system prioritises fitting new jobs into existing schedule gaps before suggesting open-ended future slots, maximising employee utilisation rather than simply appending to the calendar tail.

Context-aware origin

For same-day suggestions, the engine uses the previous appointment's location as the starting point for distance calculations, more accurately reflecting true travel distance.

Work-hour inheritance hierarchy

Company → Team → Employee override, with sensible defaults, ensures every employee has a valid working window without manual per-employee configuration.

Business-hours-only scheduling

The system never suggests slots on weekends or outside configured working hours, even for multi-day jobs, removing an entire class of human error.

Deterministic day-of-week arithmetic

Uses date difference modulo logic rather than locale-dependent functions, ensuring consistent results across server regions and collations.



Technology Stack

Component	Technology
Database Engine	Microsoft SQL Server
Core Algorithm	SQL CTEs + Recursive Queries
Geo-Computation	SQL Server Geography / STDistance
Time Zone Handling	Custom UTC Conversion Functions
API Layer	GraphQL
Frontend	Angular
Date Arithmetic	Deterministic Modulo Formula

Impact & Results

Measurable Improvements

Automated selection

From scanning all employee rows in the Planner App calendar to a ranked, filtered shortlist of eligible candidates with pre-computed slots

Reduced errors

Elimination of out-of-hours, weekend, and impossible gap placements, previously unvalidated when creating via the Planner App timeline

Automates multi-day splitting

Jobs exceeding one working day are automatically split into correct appointment chunks, previously created manually one by one

85%+ faster per work order

Reduced from ~15 minutes (manual search in Planner App) to under 2 minutes, estimated 85%+ reduction

Automated proximity

Eliminated manual map lookups, distance from work order to employee's effective location calculated automatically

Reduced cognitive load

Planners no longer need to hold employee schedules, addresses, and working hours in memory while using the calendar view

How It Made the Client's Life Easier



Everyone in a single view

11+ employees and their open slots, all at once.



Only nearby candidates

The radius filter drops anyone too far to be practical.



Cross-team visibility

Tags surface shared employees for matrix scheduling.



Confirm before committing

A preview panel shows what will be created first.



Flexible hours

Override estimated hours per job without altering master data.

Behind the scenes

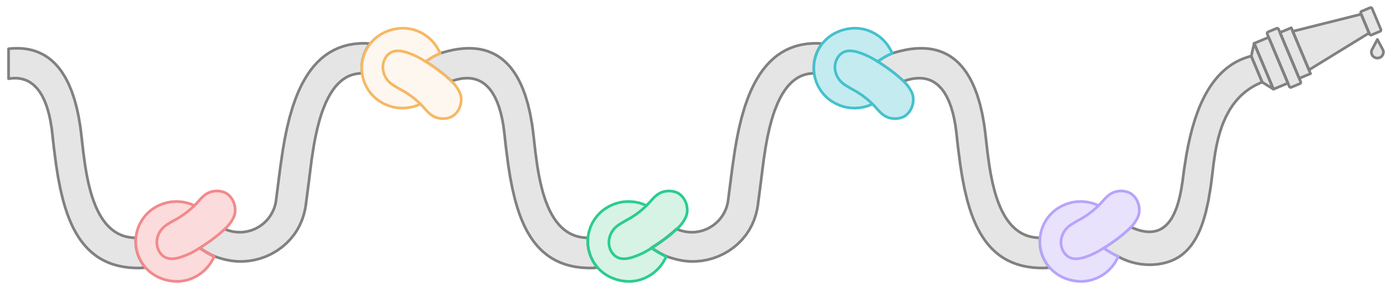
Development Challenges

Weekend & edge-day handling

The system had to treat weekends and week boundaries consistently, independent of regional date settings on the server.

Context-sensitive addresses

An employee's effective location shifts depending on whether a job falls on the same day as their previous appointment or a new one, the distance logic had to adapt to that.



Time-zone correctness

Field employees can work across time zones, so every time boundary had to stay accurate regardless of location or where the calculation ran.

Gap detection

Finding every usable free gap across a schedule, in order, while reliably stopping at the suggestion limit.

Multi-day hour allocation

Splitting a job's hours across days while respecting each day's working-hour ceiling took careful per-day arithmetic.

What's next

Planned Evolutions

Travel time buffer:

Factor drive time between appointments into gap calculations, not just availability.

Skills / Competency ranking:

Move from binary filtering to ranked matching, surfacing stronger fits higher up.

Real-time calendar sync:

Pull leave, sick days, and personal blocks from Outlook and Google Calendar as constraints.

Load balancing:

Weight suggestions toward employees with fewer scheduled hours to balance the workload.

Route optimisation:

For multiple work orders in a session, suggest sequences that minimise total travel.

Mobile planner access:

A mobile-friendly view for field supervisors who schedule on the go.

Corzent is a collaborative, people-first, digital product engineering company, empowering businesses of all sizes to build and scale. Established in 2021 and headquartered in Sri Lanka with a regional centre in UAE, Corzent leverages its expertise in UX/UI design, prototyping, SaaS & mobile development, to help clients seamlessly integrate their existing teams with our staff augmentation and dedicated development services.

For inquiries please reach hello@corzent.com