THE COMPLETE FEATURE GUIDE · RING RS3000

Every New Feature, Explained

The in-depth companion to the pre-meeting brief — every new capability, what it does, and **why we built it.**

CONTENTS

🎓 **Prepping to present, or expecting to get grilled?** There is a companion **Feature Deep Dive (with Q&A)** that explains every feature from first principles — assuming zero background — and arms you with the likely meeting questions and answers, plus an honest "watch-out" on each: ring-rs3000.pages.dev/feature-prep (download PDF).

Introduction

This is the in-depth companion to the short pre-meeting brief. The brief gives you the headlines; this guide gives you everything underneath them. It walks through every new capability we built into Ring RS3000, one at a time, in plain language — what each feature does, why we built it, and how it improves on what the plant has run for the last twenty years. You don't need a technical background to follow it. Wherever something is genuinely clever under the hood, we've translated it into plain terms and pulled out the facts worth knowing.

Ring RS3000 is the new operator application for the starch-glue mixing plant — the screens our people use to run a shift, make a batch, and respond when something goes wrong. It replaces RS-360, the operator screen built back in 2002. Importantly, Ring does **not** replace the controller that physically runs the plant: that trusted Allen-Bradley controller stays exactly as it is, doing the same job it has always done. What changes is everything the operators see and touch on top of it — a modern, fast, far more capable experience layered onto the same proven hardware. Think of it as keeping the engine and replacing the entire dashboard, the gauges, and the logbook with something built for today.

We built Ring because the old screen, dependable as it was, had four gaps that quietly cost the plant every single day. **First, its memory is trapped and thin.** It kept basic batch records and could print batch reports, but locked that data in a dead, hard-to-open database — and it kept no trends, no record of how long each step took, and no equipment history. The live, in-the-moment picture scrolled away the instant it left the screen. **Second, it can't see cost.** RS-360 never computed or kept a single number about money: not what a batch cost, not where ingredients were being over-dosed, not what a scrapped batch or an hour of hold time was worth. Those questions simply went unanswered. **Third, its alarms are cryptic codes.** When something went wrong, operators could be left staring at a bare placeholder — in some cases a code like "A0004" with no description at all, or a label that read "Thermische veiligheid ??" (Dutch for "thermal safety ??") — and had to decode it themselves, mid-fault, under pressure. **Fourth, it's hard to run.** The old system spread the plant across roughly 97 separate screens, and operators had to already know the map — which screen fixed which problem, and how to get there by hand. A new hire faced a steep learning curve before becoming fluent. Ring was built to close all four gaps at once: it remembers, it costs, it explains, and it guides.

How this guide is organized

The new features fall naturally into five themes, and the chapters that follow are grouped the same way. Each chapter takes its features one at a time, and for every feature you'll find the same three things: **what it does** in plain terms, **why we built it**, and **how it beats the old system**.

- **Make or save money** — turning the plant's history into dollars: seeing what every batch and every week actually cost, spotting over-dosing and waste, comparing today's run to your best-ever "golden" batch, and finally answering "where is money leaking?"
- **Stay ahead of problems** — catching trouble before it becomes downtime: watching trends live, reviewing any tank's recent history, and getting an early warning when a pump or feed line starts drifting off its baseline.
- **Easier for operators** — running a whole shift from one screen instead of clicking across dozens, making a batch in a few clicks, and standing up a new operator (or a new site) far faster.
- **Smarter alarms** — faster, safer fault response: alarms that read exactly as the controller states them, severity and urgency at a glance, a one-tap jump to the screen that fixes the problem, and a built-in "first thing to check" card so even a green operator responds like a veteran.
- **Own your data** — keeping a true, automatic history in a modern, open database, and being able to hand over your own operational data — any of it — in a single click, never trapped inside an aging engine.

Read it cover to cover, or jump to the theme that matters most to you. Either way, by the end you'll understand not just *what* we built, but *why each piece earns its place* — and how far ahead of the 2002-era screen the plant is about to step.

1. Run a Whole Shift From One Screen

In the old world, an operator learned the state of the plant by hunting — clicking through one screen per area and assembling the picture in their head. Ring flips that on its head: it opens to a single command center that answers "is the plant healthy, and what's running?" before anyone touches a button. Just as important, it never lets the crew act on a number that has secretly stopped updating. The result is faster decisions, fewer false alarms, and a screen an operator can trust even after stepping away.

Walk up and know the whole plant in one glance

Ring opens directly to one home screen that already has the full picture assembled: the batch that's running, an overall plant-health verdict, every key number, the active alarms, and the tanks — all stacked on a single screen the operator scrolls through, never a maze of menus. Behind that one surface sit roughly fourteen distinct information panels (the running-recipe tracker, alarms, tank cards, recently finished batches, trends, spend, equipment health, stock forecast, and more), but the operator experiences it as one continuous view that reflows cleanly to any screen size and works in both light and dark display modes.

This matters because situational awareness is the first thing a shift needs and the hardest thing to rebuild from scratch. New hires get productive faster, and veterans run an entire shift from one surface instead of memorizing where each fact lives. The home view is also always live — it rebuilds itself fresh every time the operator navigates to it, so it can never show a stale leftover from the last visit.

Before, on the 2002-era system, assembling that same picture cost several screen changes: the old software opened to a splash screen and was organized strictly one screen per plant area, so the crew navigated to individual per-tank and per-mixer forms and pieced the plant's state together manually. There was no consolidated command center at all. After: one glance, everything present, always current.

Spot trouble from across the control room in one glance

At the top of the home screen sits a single large health verdict the crew can read from across the room. When nothing is wrong it stays calm and green and says "All systems normal." The instant a real alarm goes active, the whole panel turns red and reads "Attention required" — no scanning a list to find out something is wrong.

The clever part is that it stays honest. The panel reddens only when an alarm is genuinely active right now — a list full of old, already-resolved alarms will not falsely turn it red. When there are no active alarms it even shows a reassuring "No active alarms" message with a green check. That discipline is what keeps the red signal meaningful: because it never cries wolf, when it does go red the crew believes it and moves.

On the old system, an operator had to open a dedicated alarm screen just to learn whether anything was wrong — there was no at-a-glance health verdict anywhere. After: a single, room-readable green-or-red headline that lives on the home screen and only ever reacts to real, active trouble.

Catch a stalled step before the controller's own alarm does

Ring shows the running recipe as a live checklist: each step lists its operation, its target, and how long it's taken. Finished steps get a green check and their recorded time, the step running right now glows gold and ticks up a live minutes-and-seconds timer, and upcoming steps sit dimmed and waiting. If a step runs long, it turns amber as a warning. The card tucks itself away when nothing is running and automatically scrolls to keep the current step in view.

The real payoff is timing. A step is flagged as "running long" once it passes a sensible threshold — at least two minutes, or three times its expected mixing time, whichever is greater. That means a hung step becomes visible on the dashboard before the controller's own internal alarm even fires, so the crew can step in earlier and avoid the kind of silent stall that used to quietly cost an entire batch.

On the old system, a step could hang with zero visual feedback — it showed you which step was current but offered no per-step timers, no recorded durations, and no "this is taking too long" indication. The only signal was the controller's own alarm, after the fact. After: every step carries its own live clock and an early amber warning, so a stall is caught on the home screen first.

Never act on a frozen reading that looks live

This is one of Ring's most important safety features. If a communications hiccup with the controller stops a value from updating, the operator is not fooled into trusting a frozen number. Ring visibly dims the affected reading and stamps it with the time it was last known fresh — for example, "as of 14:32" — so a stuck value is never mistaken for a live one. Any tile that hasn't refreshed within about ten seconds gets this treatment automatically.

It goes a step further to be tamper-proof against the clock itself. Ring measures how old a reading is using a time reference that can't be tricked by daylight-saving changes or someone setting the system clock backward, and if the math ever produces a nonsensical "negative age," it treats the value as stale rather than fresh — a deliberate second line of defense. Communication drops are a known, observed reality at this plant, which is exactly why this guard exists.

On the old system, fill graphics and readouts simply froze while still looking perfectly live during a comms drop — no dimming, no last-fresh stamp, and no way for the operator to tell. (Even an early version of Ring had this same hazard before it was fixed.) After: every live number openly shows its provenance and dims within ten seconds of trouble, so the most dangerous failure mode there is — a stale reading that still looks current — simply can't catch the crew off guard.

One tap from the number to the screen that fixes it

The four numbers that most often decide an operator's next move sit permanently across the top of the screen, color-coded: open alarms, mix-tank temperature, how far along the current batch is, and which tank needs attention. Each one is a shortcut — tapping it jumps straight to the screen that handles that issue. Open alarms go red when there are any and green when there are none; the tank-attention tile goes amber when a tank is flagged and takes you directly to the right tank's screen.

The benefit is simply speed: the four action-driving numbers are always on glass, color-coded so a problem stands out, and one tap from the screen that resolves it. There's no hunting between menus to act on what you just noticed.

On the old system, these numbers existed only as raw readouts buried on individual area screens — there was no consolidated strip of key figures and certainly no way to click a number to act on it. After: a single color-coded row of the four numbers that matter, each one a direct shortcut to its fix.

No fake "100% done" for the last 20 minutes of a batch

Some recipes end with a few final steps that aren't measured by quantity. A naive progress bar would hit "100% done" the moment the measured portion finishes — even though twenty real minutes of work remain. Ring refuses to lie here: in that closing stretch it caps the headline at "99%" and clearly labels it "Finishing — Step 8 of 12," and it switches the time display from a fake countdown to an honest "time in this step" elapsed clock.

Why it matters: operators have to be able to trust the headline number, especially near the finish line where misplaced confidence causes mistakes. This isn't theoretical — on a real observed batch, the unguarded display read "100% / less than 1 minute" while the batch actually kept running from 15:08 all the way to 15:30, roughly twenty-one minutes of "done" that wasn't done.

Even an early, un-fixed version of Ring had this blind spot — on a real observed batch its display read 100% with under a minute left while roughly twenty-one more minutes of work remained. After: Ring holds the line at 99% and names the finishing phase outright, so the display stays truthful in exactly the moment the old one was most misleading.

Step away and still see what just finished

When a batch completes, its result doesn't vanish. Ring holds a plain-language summary on screen — for example, "Batch complete — Formula 1 to Storage Tank 2, 12 steps, 2h 04m, completed 15:30" — and keeps the finished recipe checklist visible with every step checked off. The summary stays until the next batch begins or the operator taps "Dismiss."

This solves a small but real frustration: an operator who looked away for a moment used to come back to a screen that had already erased a two-hour batch in a three-second snap, leaving nothing to confirm what just happened. Now the result waits for them.

An early, un-fixed Ring dashboard erased a finished two-hour batch in a three-second snap, and RS-360 — which had no consolidated home screen at all — likewise offered no held completion summary to fall back on. After: the just-finished batch stays on screen, in human terms, until it's intentionally cleared.

Catch a developing tank problem the controller can't even alarm on

The "tank attention" tile points the operator straight at the next tank that needs a look. Crucially, this includes problems the controller itself can't always raise an alarm for. When everything's fine it reads "All tanks OK" in green; when something's developing it turns amber, names the issue, and takes you to that tank with one tap.

The standout example is the temperature-control unit that conditions process water. Ring can flag a condition like "jacket water LOW" and route the operator there even in a situation where the controller's built-in alarm for that fault is unable to fire. That's a developing problem that would otherwise go completely unnoticed until it stopped the line — and Ring surfaces it before it becomes a stoppage.

On the old system, operators relied on a small red indicator on the main navigation for that unit; in fact, an early Ring build hid that unit behind a screen nobody opened while the dashboard cheerfully read "All systems normal" even though the unit was running low at the live plant. After: Ring synthesizes that low-water condition into the attention tile, routes you straight there, and recovers — then improves on — what the old indicator did, by adding one-tap click-through.

Make production decisions on real history, not invented numbers

The trends graph on the home screen plots genuine completed-batch data, grouped by shift or day, with real calendar dates and an honest "As of 14:32" timestamp so everyone knows how current it is. If there's nothing to show yet, it says so plainly rather than inventing a chart.

This matters because supervisors and evaluators base real production decisions on these numbers. An earlier Ring build had actually filled this flagship chart with random demo figures dressed up to look authoritative — exactly the kind of thing that erodes trust. That fabricated data was stripped out, and the chart now draws only from the real record of completed batches, with a visible timestamp proving its freshness.

On the old system there was little to no on-screen trend visualization of this kind at all. After: trustworthy, dated production history with clear provenance — and an honest empty state when there's genuinely nothing to plot.

Let the floor self-troubleshoot without specialist tools

Ring puts professional diagnostics right on the plant floor so maintenance can troubleshoot in place — no protocol analyzer, no engineering-workstation software required. It includes an on-screen numeric keypad sized for gloved hands, confirmations that don't freeze the rest of the screen, a built-in help overlay, a keyboard shortcut to jump straight to the running batch, a tool that can read any value coming from the controller, a view-only health panel showing the last successful contact and

recent connection changes, a live view of the raw communication traffic, and a one-tap button to bundle up diagnostics for support.

The benefit is self-sufficiency: the crew resolves communication questions on the floor, in the moment, instead of waiting on a specialist with special tools — and the gloved-hand input keeps them fast in a real touch-screen factory environment. Under the hood, Ring talks to the controller over a modern open industrial standard using plain named values, so there are no fragile background "bridge" programs that have to be kept alive. The team even measured how many simultaneous conversations the controller can handle and deliberately staggered Ring's start-up so it never floods the controller at launch.

On the old system, communications were logged to disk but there was no modern instrumentation — no way to inspect a live value, no connection-state view, no one-tap support bundle — and everything ran through bespoke background bridge programs that had to be babysat. After: direct, standards-based communication with first-class diagnostics built right into the operator screen.

End false "controller down" calls — know an outage from an idle controller

Ring can tell the difference between a controller that is genuinely down and one that is perfectly reachable but simply sitting idle with nothing to report. It does this with a single internal "heartbeat" classifier that sorts the connection into clear states — connecting, connected, going stale, starved, or disconnected.

The payoff is fewer wasted call-outs: the team stops chasing phantom "the controller is down!" escalations that turn out to be a healthy-but-quiet controller. And because that very same heartbeat signal also drives the on-screen dimming of stale readings, the operator's eyes and the maintenance team's alerts are always reading from one consistent source of truth — they never disagree about whether the plant is actually talking.

On the old system, communications were logged but there was no way to classify *why* data had stopped, so a reachable-but-idle controller and a true outage looked identical to the operator. After: a five-way connection classifier that feeds both the alerting and the dashboard's dimming from one signal, so a real outage and a quiet controller are never confused again.

2. Turn Plant Data Into Money

For more than twenty years, the old operator screen treated your plant's history like steam rising off a mixing tank: whatever the operator was watching simply evaporated the moment it scrolled off the display. No saved trends, no record of how long each step took, no equipment runtime, and — remarkably — not a single number anywhere about what any of it cost. Ring takes the opposite view: the plant's day-to-day data is an asset, not exhaust. It quietly remembers everything, then turns that memory into decisions and dollars — showing you what each batch and week actually cost, catching a pump that's starting to fail before it ruins a batch, comparing today's run to your best one ever, and letting you hand over your own data in a single click. Everything in this chapter runs off one quiet local database, with no added burden on the controller that actually runs the plant.

See Exactly What You Spend — And Catch the Money Leaking Out

Ring includes a true cost engine: for every batch and every week, it calculates what you actually spent — and just as importantly, where money is quietly leaking out through over-dosing ingredients, scrapped batches, and product held too long. You load in your roughly 1,400 batches of past history and your current ingredient prices one time, and the spend figures light up with real, actionable numbers. Because it remembers what each ingredient cost on the *date* a batch ran, the math stays honest even as prices change over time.

Why it matters: the owner can see plant spending and waste without driving in to the site. Once that history and today's prices are loaded, you effectively have about six months of detailed per-batch and per-week spending to comb through, and "giveaway" — the extra material you dosed beyond the recipe — becomes a real number you can chase down rather than a vague suspicion. The system is deliberately built to never *under*-state your spend: if an ingredient was used but has no price entered yet, Ring flags that batch as not-fully-priced and names the missing item, rather than quietly pretending that ingredient was free.

Before and after: the old screen left every cost question unanswered — it never calculated or kept any cost, giveaway, or waste figure at all, because no cost engine existed. With Ring, you see per-batch and per-week cost with price history preserved, watch over-dosing add up as real money, and print or email the whole thing as a clean cost report.

Make Faster, Better Calls With Decision Screens the Old System Never Had

Ring adds a whole set of brand-new insight screens that turn raw plant events into clear decisions. They include: alarm trends that show which alarms fire most often and how long they take to handle; a board showing which tanks are clean, dirty, or in use; a view that ties any alarm back to whichever batch was running when it fired; a planning board for laying out upcoming batches; a downtime and equipment-effectiveness screen; an overlay that compares any batch against your best "golden" run; and a prioritized alarm summary that helps silence nuisance noise.

Why it matters: these are eight dedicated screens that let you do things the old system simply couldn't support — kill nuisance alarms that distract operators, plan batches ahead of time, confirm a tank is actually clean before a product changeover, trace an alarm straight to the batch that triggered it, and spot a process drifting off course before it becomes a problem. Crucially, every one of these screens is view-only: they look at the data and help you decide, but they never reach in and change the controller.

Before and after: on the old screen, all of these decisions were guesswork — it offered one basic alarm-history lookup and some static print-outs, with no analytics, no equipment-effectiveness tracking, no clean-tank board, no batch planner, no best-batch overlay, and no way to shelve nuisance alarms. Ring delivers all eight of those decision tools as proper, purpose-built screens.

Watch Trends Live — and Review Any Tank's Last 30 Days

Ring lets you pick live values from the plant — a temperature, a level, a feed rate — and watch them draw themselves on a rolling chart, complete with markers showing when batches started and alarms fired, then export the whole thing to a spreadsheet file. No specialized engineering software is required to do any of it. Alongside that live view, a companion Tank History screen lets you look back at each tank's temperature and level over the past 30 days, so you can review what actually happened instead of wishing you'd been watching.

Why it matters: a plant expert can see live trends without ever opening the controller's engineering toolset, and maintenance can pull up any tank's recent temperature-and-level behavior right inside Ring. A question that used to be unanswerable — "was that tank running hot an hour ago?" — gets answered in seconds. The live chart keeps a rolling memory roughly eight hours deep and lets you zoom from a five-minute close-up out to that full window, hover for exact values, and save it all out.

Before and after: the old system left you blind to trends — it had no live moving-chart capability at all, only a static picture-builder for drawing fixed plant diagrams, and no per-tank history viewer whatsoever. Ring gives you genuine live trending with spreadsheet export and batch/alarm markers, plus a 30-day tank history the old system never offered.

Catch a Failing Pump Before It Scraps a Batch

Ring continuously watches how fast each ingredient is feeding into the mix and quietly compares it against what's normal for that recipe and that ingredient. When a pump or feed line starts drifting off its usual rate, Ring warns you in plain language — something like "starch feed is running noticeably slower than usual, check the pump" — so you can act before that drift turns into an alarm, a scrapped batch, or a line stoppage.

Why it matters: this is early warning instead of after-the-fact reporting. Ring learns each ingredient's normal feed rate from its recent batches, and to avoid crying wolf it only raises a flag when the drift is real and *sustained* — roughly a fifteen-percent change that holds across two batches in a row, not a one-off blip. The warning shows up on the main dashboard's equipment-health panel and goes out by email through the same notification system that drives Ring's other alerts, so it reaches the right person without any special setup.

Before and after: the old system gave no warning of any kind — it kept no equipment runtime or feed-rate data at all, so this kind of drift could never even be calculated, let alone flagged. Ring turns the data the plant generates anyway into genuine preventive maintenance.

Every Report You Rely On — From One Database, With Modern PDF

Ring keeps every printable report your operators depend on — what was batched, how much of each ingredient was used, batch and alarm history, recipes, inventory, and per-shift consumption — and brings them into the modern era with proper print preview and PDF output. Every one of them is now drawn from a single local database instead of an aging, effectively dead file system.

Why it matters: quality staff and supervisors get the exact report set they've always relied on, but now they can preview before printing, save as PDF, export to a spreadsheet, and filter by date or by tank. The batch-history-with-usage report can even be scheduled to run automatically and will cover your full imported history of roughly 1,400 past batches. A particularly nice touch: the per-shift consumption report correctly credits overnight usage back to the right shift rather than splitting it across a calendar midnight.

Before and after: the old system trapped your reports as fixed print-outs sitting on top of a database engine that's now obsolete, with several underlying record sets — inventory and shift data among them — effectively abandoned and nearly impossible to read. There was no PDF and no single place you could actually query. Ring delivers that whole report set from one modern database with preview, PDF, spreadsheet export, and filtering, and it revives the shift-consumption and inventory history the old system had left stranded.

Answer "Show Me Our Data" in One Click — Because You Own It

When an auditor, a manager, or a customer says "show me our data," Ring lets you answer in a single sitting: pick the records you want, choose which columns and an optional date range, and save your own data straight out to a standard spreadsheet or CSV file. Your operational data is never held hostage by an aging engine that might one day refuse to open.

Why it matters: your data stays portable and stays *yours*, available on day one without waiting on a specialist to extract it. Managers and auditors get a self-serve, point-and-click export, and the files it produces follow the universal spreadsheet standard so they open anywhere.

Before and after: the old system technically had an export feature, but the data underneath it was locked inside an aging database format that became harder and harder to get anything out of as the engine grew older. Ring exports any set of records to a clean, standards-compliant file from an open, modern store — so what's yours actually stays accessible.

Never Trust a Frozen Number — Live, Stale, or Off-Target at a Glance

Ring makes sure you always know whether a number on screen is real. Each reading carries a clear indicator of whether it's live or stale, temperatures and levels that drift off-target get flagged, and Ring shows how far a batch over- or under-delivered against its recipe. The goal is simple: nobody should ever act on a frozen number during a brief communications hiccup. When a reading's source goes quiet, its tile doesn't just sit there looking confident — it visibly dims down to about forty-five percent brightness and offers an "as of" timestamp so you can see exactly how old it is.

Why it matters: operators get plain-language link status, clear stale warnings, and instant over/under-delivery feedback, plus that automatic fade that makes a stale tile impossible to mistake for a live one. The dimming applies right where it counts — the main dashboard tiles, the status strip along the screen, and the window's own frame — so a stale reading announces itself everywhere at once.

Before and after: the old system could leave a dead reading looking perfectly confident — it just showed raw numbers with no sense of whether they were fresh, healthy, or on-target, so a communications blip would leave a stale-but-convincing value sitting on screen with no warning at all. Ring shows link state, health, and batch variance on every reading, and fades anything stale until its source comes back to life.

Keep a True History Automatically — With Zero Extra Load on the Controller

This is the quiet engine that makes the whole chapter possible. About once a minute, Ring records each tank's temperature, level, and current recipe, steadily building a genuine history you can later chart, compare, and cost. And it does this without adding any new burden to the controller that runs the plant — it simply reuses the readings Ring is already receiving, so the controller never sees a single extra request. This is the memory that powers live trending, best-batch comparison, and the cost engine.

Why it matters: quality and maintenance finally get a real historian — they can trend any tank's temperature or level over weeks, line a batch up against the golden profile, and catch drift before it ever becomes an alarm. Just as important, the history is *honest*: Ring keeps about thirty days of detail, automatically trims older data, skips empty tanks so there are no phantom blank entries, and — critically — refuses to record anything while the link to the plant is dead. That last rule means a frozen connection can never sneak in a row of fake, flat-line readings that look real later. Where there's a gap in the data, Ring shows the gap rather than inventing numbers to fill it.

Before and after: the old system forgot almost everything — it had a basic alarm lookup and some print-outs, but no continuous tank-by-tank history and no way to trend or analyze process values over time; some of its text logs were even erased before anyone read them. Ring captures a temperature, level, and recipe snapshot for every tank roughly once a minute into a fast, indexed modern store, keeps a rolling thirty days, handles gaps honestly, and does it all without a single extra request to the controller.

3. Running a Batch, Without the Guesswork

Every shift at the plant is a string of judgment calls: which recipe, how much, where it goes, and whether the last batch actually landed on target. On the old system, those calls were made against a wall of numeric codes, and the live, in-the-moment picture of a running batch scrolled away the instant it passed — its step timing and the day's trends did not survive a restart, even though basic batch records were saved and could be printed. This chapter walks through how Ring turns batch-making into a guided, point-and-click flow with a plain-language safety check, an honest live progress tracker, and a permanent, searchable record of everything the plant has ever produced.

Stop Losing Twenty Years of Plant Knowledge Every Time the Screen Restarts

This is the single sharpest reason to switch. On the old system, the live, in-the-moment picture scrolls away as it happens — a running batch's progress does not survive a restart, and there is no record of how long each step took or surviving trends. It did keep basic batch records and could print them, but not the richer, decision-driving history operators and owners actually need. Ring keeps all of it permanently. That durable record is the foundation, and on top of it sits a carefully vetted slate of upgrades chosen from thirty-one candidate ideas narrowed down to two dozen survivors, each scored on how much operators would love it, how well it fits the plant, and how much effort it takes to build.

The headline upgrade on that slate is a "batch day-planner" that tells the operator when the batch will next need their hands — for example, "next hold around 7:12, ready to move around 8:05" — so they can do their rounds instead of standing at the panel watching. It works purely by reading the live current step and comparing it against typical timings from past good runs, so it adds no extra load to the controller. Several related upgrades ride alongside it: a practice mode where operators rehearse on a replay of their own recent batches at ten times speed with zero plant risk, an automatic import of the plant's own twenty years of history so it appears on day one, and a "running long" warning that flags when a step is taking longer than usual — an early hint of a tiring pump or a clogging line.

These upgrade features are on the near-term roadmap — they are documented, designed, and committed as a plan, but not yet built. The crucial part, the permanent record they all depend on, is already live. Before: the old system showed you the live step and forgot every duration the instant it passed, so a question as basic as "when will this batch next need me?" was simply unanswerable. After: Ring keeps the step timing, the proven good-run profiles, and the needs-you-next timeline that make that question answerable at all — and finally surfaces the history the old system quietly recorded for twenty years but never let anyone actually see.

Confirm Exactly What the Plant Will Do — In One Plain Sentence — Before It Arms

Starting a batch in Ring is a guided flow: the operator picks a recipe from a list, types the target fill level, chooses the batch mode, and presses Start. Before anything reaches the plant, Ring shows one plain sentence describing exactly what is about to happen — for example, "Formula 1, Single Batch, No splitting, 625 gallons" — and asks the operator to confirm it. Only after that confirmation does the controller arm.

Under the hood, the order of operations is deliberately protective. Ring writes all the recipe settings first and arms the plant last, so a half-finished setup can never quietly start a batch. If the arming step fails, Ring automatically backs the configuration out rather than leaving the plant in a half-armed state. It also checks that the link to the controller is healthy before sending anything, and it gives the operator a clear "it worked" or "it failed" message instead of leaving them guessing whether the command landed. Even the typing is smart — entering a target like "1200" sends a single clean instruction rather than firing off four separate partial commands as the digits go in. Targets are validated to a sensible range so a fat-fingered number gets caught on the spot.

Before: on the old system, operators keyed recipes and levels directly against raw numeric machine codes — the ingredient list itself doubled as the code legend — with nothing more than a bare Yes/No box for confirmation, so a mistyped code could arm the plant unseen. After: the operator reads the action in plain English, confirms it, and trusts that the plant arms last and unwinds itself if anything goes wrong.

See Exactly Where the Batch Stands — Progress That Stays Honest Through a Network Hiccup

While a batch runs, Ring shows the recipe as a live checklist. Each ingredient appears with its target amount next to the amount actually delivered, marked Complete, In Progress, or Pending, alongside an overall percent-complete and a highlighted caption naming the current step that scrolls itself into view. At a glance, the operator knows where the batch is and whether each ingredient landed on target — no more inferring the situation from gauges.

The "stays honest" part is where the real engineering shows. Ring checks the batch's progress every couple of seconds in the background. If a reading momentarily fails to come through — a network blip — Ring holds the last known good picture on screen rather than flashing a false "nothing's happening." And before it declares a batch finished, it requires three confirmed idle readings in a row, roughly six seconds, so a single transient stutter can't blank the display mid-batch. There's also a fix for a specific old confusion: the final mixing and transfer steps of a recipe carry no target amount, which used to make a batch read "100% complete, under a minute left" for the entire twenty-minute-plus finishing stretch. Ring now explicitly labels that finishing phase, so the percentage stays truthful all the way to the end.

Before: the old system showed a per-step target-versus-actual table, but with no honest overall percentage, no "step 4 of 9," no time estimate — and it famously read "100% / under 1 minute" for a full twenty-one-minute finishing phase. After: Ring gives an amount-weighted percentage with a clear finishing-phase label, plus the hold-last-picture and confirmed-idle logic that keep the display trustworthy when the network stutters.

Keep a Searchable Record of Every Batch — Even One Cut Short by a Restart

Every batch Ring runs is recorded with its recipe, tank, start and end time, target versus actual volume, and status — Running, Complete, Cancelled, or Interrupted. Operators can browse the full list, filter by date or tank, search by batch number or recipe name, and print or save a clean, branded batch-history report. A global "go to batch" shortcut lets anyone jump straight to a specific batch from anywhere in the app.

The honesty of this record is the point. Ring writes a batch's starting details the moment it arms and finalizes the ingredient totals and actual volume in a single, all-or-nothing save when it completes, with guards so two events can't collide and corrupt the record. Crucially, if the screen crashes or reboots while a batch is running, that batch is closed out with a dedicated "Interrupted" status — because in-memory tracking simply can't survive a restart, and the truth is better than a phantom batch that appears to run forever.

Before: the old system did keep batch history and could print a report — over fourteen hundred batch records and nearly seventeen thousand step rows — but it offered no operator notes, no way to distinguish a crash from a clean finish, no full-text search, and the data sat locked inside an obsolete database format that needed a custom-built tool just to read. After: Ring lets you search every batch by text, trusts an honest "Interrupted" marker for reboots, and stores it all in an open, modern, easily queried database.

Edit a Recipe Once — and the Screen, the Plant, and the Audit Trail All Stay in Sync

Supervisors can edit any of the plant's six recipes step by step — operation, amount, mix time — with the ability to add, insert, delete, and reorder steps, all in a spreadsheet-like grid with built-in validation. A live readout shows the resulting batch weight and volume as edits are made, catching mistakes before a batch ever runs. A single "Apply" then does three things at once: saves the recipe to Ring's database, pushes the same recipe down to the matching slot in the controller, and snapshots the new version so you can see exactly what any recipe looked like at any point in the past.

The value here is that nothing can drift apart. On the old system, a recipe change synced across machines through a fragile multi-step handshake, and the records held only the current values — no version history at all. In fact, the only way anyone could even tell that operators had been editing recipes was to dig through nearly seventeen thousand old step records after the fact and reconstruct it. Ring's automatic versioning records every change with a date stamp, and it's deliberately fault-tolerant: if the version snapshot ever fails, it never blocks the actual save. The weight and volume preview mirrors the plant's own long-standing calculation method, so what the operator sees on screen matches what the plant will produce.

Before: editing a recipe meant raw numeric entry against an obsolete database, then a fragile multi-step handshake to sync the change across several machines, with no version history whatsoever. After: one "Apply" writes the database, syncs the controller, and files a dated version snapshot — with a clear message confirming each result.

Catch a Drifting Batch Early — Measured Against Your Own Proven Good Runs

Ring quietly builds a "golden" reference curve from past good batches of each recipe, then overlays the current live batch against it on screen — flagging when the temperature starts drifting away from the proven profile. While there isn't yet enough history to build a reliable reference, Ring is honest about it, showing a "still collecting data" state rather than a misleading comparison. The comparison kicks in once it has at least a few good batches to learn from, and it warns when the live batch wanders more than a few degrees off the reference.

What makes this remarkable is that it costs the controller nothing extra. Ring already reads each tank's temperature and level for the live displays; this feature simply samples that information it already has — about once a minute — and keeps a rolling thirty-day history that prunes itself automatically. From that history it builds a typical "good run" curve for each recipe and plots the live batch right on top of it, showing the current and worst deviation so far. (The comparison covers temperature, which is the meaningful signal this plant's controller actually exposes.)

Before: the old system had no historian and no golden-batch comparison at all — only end-of-shift production totals and bare batch headers, never a temperature trace over time, so there was simply no way to compare a live batch against past good ones. After: Ring keeps a self-maintaining thirty-day temperature history per tank, builds a reference curve from your own best runs, and flags drift live — brand-new intelligence that adds no extra burden to the plant.

Tie Tribal Knowledge to the Exact Batch — Ready for the Next Quality Investigation

Operators can attach a free-text note to any batch — things like "added extra borax, starch was clumping" or "stopped early, corrugator down" — written right from the batch report and saved with that batch permanently. It captures the "why" that used to live on a scrap of paper or only in someone's memory.

The payoff comes later, when something goes wrong. When a customer complains about a bad batch or quality drifts, that operator context is right there, tied to the exact batch it describes, searchable and printed on the report — instead of being lost the moment the shift ended. Ring threads the note through every place a batch shows up, so it appears consistently wherever you look the batch up. The note field was added cleanly so the records stay consistent and a blank note simply clears rather than cluttering the data.

Before: the old system captured no operator context at all — its batch records were a fixed set of columns (machine, recipe, tank, times, weights, status) with no room for a free-text reason, so the human knowledge behind a batch was simply lost. After: every batch carries its own operator note, searchable and printed on the report.

Re-Run the Usual Recipe in One Tap — By the Name Operators Actually Use

Ring is moving to show recipes and tanks by the names operators really call them — such as the plant's own custom recipe name rather than a generic "Formula 1" — and to put one-tap buttons on each tank's start screen for that tank's most-used recipes (drawn from the last thirty days, top three). The everyday case — running the same recipe on the same tank again — becomes a single tap, by a name the operator instantly recognizes. (This one is in progress.)

The familiar-names work is genuinely in flight. The machinery to relabel recipes by their real names is wired in, and custom names are saved and update live across the app. But an on-site review found a couple of screens still showing the literal "Formula N" label — specifically the storage-tank card and the mixer panel — and later work has been narrowing that gap. We present it honestly as parity in progress rather than fully done.

Before: the old system showed an operator-entered name on the tank card and mixer panel, resolved from the recipe number. After: Ring relabels the recipe pickers by familiar name and saves custom names persistently, with the remaining surfaces being brought into line, plus quick one-tap buttons for each tank's most common recipes.

Split a Batch to the Right Second Tank — With Overfill Protection That Follows It

Ring lets an operator split part of a batch into a second storage tank, chosen by name — "Split with Storage Tank 2," "...Tank 3," "...Tank 4" — and the confirmation dialog names the real target tank, not a position in a list. The destination is identified by its actual tank number, and the plant's overfill and high-level protection guards that very same tank. (This one is in progress, built and awaiting a final controls-engineer sign-off and a bench check before it runs live.)

This solves a real and subtle hazard. The plant's underlying logic routes the split using the absolute tank number, so Ring now maps the operator's choice straight to that real number. An earlier version of this feature wrote the position in the list instead of the true tank number, which meant the overfill protection could end up guarding the wrong tank — a serious issue that has been corrected in code. Because of how consequential that is, the fix is deliberately gated: it's built and ready but is presented as awaiting confirmation and a controlled bench test before split batching is used on the live plant.

Before: the old system technically supported splitting — the report could even print that a batch was split with a tank — but in six months and over fourteen hundred batches the plant never once used it, so this is added capability and safety rather than catching up to something lost. After: Ring maps the split to the real, absolute tank number with overfill protection that follows the operator's chosen tank, with the confirmation dialog naming exactly which tank will fill — built and gated, pending its final live sign-off.

4. Smarter Alarms, Faster Fixes

When something goes wrong on the plant floor, the clock starts ticking, and every wasted minute is product that isn't getting made. The old operator screen treated a fault as a line of text and left the rest to the operator: figure out what it means, find the right screen, remember what to check, and hope you picked the urgent one out of the list. Ring rebuilds that entire moment. Every alarm now reads exactly as the controller states it, sorts itself by urgency, links straight to the fix, comes with a built-in "first thing to check" card, and ties downtime back to the exact batch that was running. The result is fewer minutes lost to confusion and a new operator who can respond like a veteran.

Every Alarm Reads Exactly Like the Controller — No Guesswork Mid-Fault

When an alarm fires, the operator sees the same number and the same plain-English description the controller itself uses — phrases like "E-STOP has Been Pressed" or "Make Ready Tank Temperature Too High" — never a label someone invented or a bare code. Behind the scenes, Ring carries a faithful copy of the controller's alarm legend (76 active alarm numbers today), including which ones are critical and which are warnings, and it correctly expands the repeating blocks that cover each storage tank so every tank's alarms read correctly. It even mirrors the controller's own internal bookkeeping — the same alarm shown as active, silenced, or resolved — so what's on screen tracks the machine bit-for-bit.

Why it matters: in the middle of a fault is the worst possible time to be second-guessing what a message means. Because Ring's text matches both the controller and the screen operators already trust, they read the alarm and act on it instead of pausing to decode it.

Before and after: the old system left operators decoding cryptic placeholders. Its alarms were positional Dutch text in flat files with real gaps — two entries literally read "Thermische veiligheid ??" (thermal safety, unknown), two more were simply blank, and one alarm that still fires today had no entry at all, so it showed up as the meaningless code "A0004." Ring shows a real description for every covered alarm.

From "Alarm Fires" to "Fixing It" in a Single Tap

Each alarm row offers a one-tap link straight to the screen that addresses it — "Go to Storage Tank 2," "Go to Make Ready Tank," "Go to the temperature control unit" — so the operator lands in the right place without knowing the plant's screen map by heart. Ring works out the destination from the alarm itself: a storage-tank fault opens that exact tank, a make-ready scale, temperature, feed, or overload fault opens the make-ready screen, and so on. For genuinely plant-wide alarms like an emergency stop or a low controller battery — which don't belong to any single screen — Ring deliberately offers no link rather than send the operator somewhere unhelpful. The link is purely a navigation shortcut; it sends no commands to the controller.

Why it matters: a brand-new operator who doesn't yet know where everything lives still gets to the correct screen on the first tap, collapsing the time from fault to fix down to seconds.

Before and after: the old system made operators find the right screen themselves. Alarms were inert text, and you had to already know which area screen to open and navigate there by hand — knowledge that only came with experience. Ring puts that knowledge in the button.

Expert First-Response Guidance the Instant an Alarm Fires

Hover over a covered alarm and Ring pops up a plain-language troubleshooting card: what the alarm actually means, the very first thing to check, and who to call. About ten of the highest-priority alarms have a card today, and alarms without one yet simply point the operator to the manual. Crucially, supervisors can write and refine these cards themselves — and a supervisor's edited card always takes precedence over the shipped version — so the team's collective know-how gets captured on screen and keeps improving. The whole feature only watches and advises; it never touches the controller or the alarm itself.

Why it matters: on a covered alarm, a green operator can respond like someone with years on the floor the moment it fires, instead of flipping through a paper binder while the problem grows. And because supervisors can keep adding cards, the floor for the whole team keeps rising over time.

Before and after: the old system left operators on their own for most faults — at best a couple of hardcoded popups showed static "possible causes" text. There was no editable, plant-wide playbook of meaning, first check, and who to call. Ring turns binder lookups into instant on-screen coaching.

Tell a Live Critical from a Resolved Warning Across the Room — Color-Blind Safe

Every alarm row now shows two things at a glance: how serious it is (Critical, Warning, or Info) and where it is in its life (Active, Silenced, or Resolved). Importantly, the state is shown as a distinct shape, not just a color — a filled triangle for active, a half-circle for silenced, a check mark for resolved — so a color-blind operator reads the same answer just as fast. Resolved and silenced alarms fade back to about half brightness and drop to the bottom, while live faults rise to the top with the newest first. Ring also caches this information so it can refresh the whole list every two seconds without bogging down.

Why it matters: an operator can separate a live critical from a long-handled warning from across the room, and nobody mistakes a fault that's already been dealt with for one that still needs attention.

Before and after: the old system forced operators to read every line equally — a flat, single-color list with no severity tiers, no life-cycle states, and nothing for color-blind staff. (An early version of Ring even made the mistake of painting every row the same red.) Ring now shows urgency and status at a glance, in shape plus color, with handled alarms quietly dimmed.

Catch a Stalled Batch Before It Costs You — "On Hold" Shown Right on the Dashboard

If a running batch goes on hold — held for inspection, held too long, or running too slow — an amber "ON HOLD" chip appears right on the active-batch banner with the reason spelled out, and the time-remaining estimate changes to read " (paused)." Ring watches the live alarms for the family of hold conditions and, if more than one is active, surfaces the most serious. It's drawn entirely from information Ring is already collecting, so it adds no extra load on the controller.

Why it matters: the operator sees the batch is paused and exactly why without leaving the dashboard, and — just as important — the progress bar can no longer keep creeping forward and imply the batch is still moving when it has actually stalled. A stuck batch gets caught early.

Before and after: RS-360 had no on-screen chip tying a hold alarm to the running batch, so a stall did not stand out. (On one observed live batch, an early dashboard kept showing normal progress while the batch actually sat on hold for roughly twelve minutes.) Ring ties the hold straight to the running batch on the main screen.

Spot the Fault That's Been Sitting Too Long — Live Age and an "Oldest Active" Callout

Each alarm now shows how long it has been going — "4m," "1h 12m" — and the panel header calls out the single oldest unhandled alarm, the clearest possible signal of "is anyone actually dealing with this?" The age counts up live and is recalculated on the same two-second refresh the rest of the screen uses, with no extra machinery. If Ring genuinely doesn't know when an alarm started, it leaves the age blank rather than inventing one.

Why it matters: a supervisor immediately sees how long a fault has gone untouched and can pounce on the one that's been ignored. On that same observed live batch, the plant lost about twelve minutes to hold alarms that, under Ring, would have stood out instantly.

Before and after: the old system stamped a time on an alarm but had no live, ticking age and no "oldest active" callout, leaving supervisors to eyeball timestamps and guess how stale a fault was. Ring does the math and surfaces the answer.

Reserve Operator Attention for Real Criticals — Loud for E-Stop, Calm for Warnings

When a new alarm fires, Ring shows a popup styled to match its seriousness. Critical alarms blink for attention and put an Emergency Stop button one tap away; warnings appear calmly so they inform without crying wolf. If Ring ever encounters an unknown alarm, it deliberately treats it as critical — it would rather over-warn than under-warn. The popup only fires on a genuinely new alarm, not when the same alarm is merely acknowledged or silenced, so operators aren't pestered by repeats. There's also a full-window version that locks navigation behind a live alarm until the operator acknowledges it or explicitly chooses to keep monitoring — so a serious alarm can't be quietly clicked past.

Why it matters: by saving the loud, blinking, E-Stop-ready treatment for genuine criticals and keeping routine warnings calm, Ring stops "alarm fatigue" from training operators to tune everything out — which is exactly when the one alert that matters gets missed.

Before and after: the old system's popups were hardcoded per alarm with a fixed red look and an endless blink regardless of how serious the fault actually was — which taught operators to ignore them — and it had no screen-locking critical overlay at all. Ring tunes the alert to the real severity and makes the worst ones impossible to miss.

An Accidental Tap Can't Disrupt a Live Batch — Confirm Before Every Command

The familiar buttons that pause, restart, or reset a batch now each ask "are you sure?" before the command actually reaches the controller. The behavior operators know is otherwise unchanged — Resume and Reset still stay hidden until a batch has been put on hold, and the commands trigger the exact same controller actions as before — but a single confirmation step now stands in front of every command that changes the live process.

Why it matters: operators keep all their existing muscle memory, but a stray tap or a brushed elbow can no longer interrupt a batch in mid-run. It's a small guard that prevents an expensive mistake.

Before and after: the old system sent the command on the very first click, with nothing between the operator's finger and the live process. Ring keeps the identical control behavior and simply adds the "are you sure?" step.

Pin Downtime to the Exact Batch and Recipe That Was Running

Ring can now answer a question the plant could never answer before: when this alarm went off, what were we making? For any alarm, it matches the time it fired against every batch that was running at that exact moment — including long batches that started before the period you're looking at — and shows the batch and recipe alongside the fault. Because alarms aren't tied to a specific tank, Ring honestly shows all the batches that were running concurrently rather than guessing at one.

Why it matters: maintenance and supervisors can finally attribute downtime or a quality issue to the specific batch and formula that was in progress, instead of manually cross-reading two separate reports and hoping the timestamps line up. Root-cause and accountability conversations start from facts.

Before and after: the old system kept alarm history and batch history as two completely separate, unlinked logs — so "what was the plant making when this alarm fired?" was simply unanswerable. Ring joins them automatically.

Honest Status on Alarm Coverage — Today's Alarms Are Correct, the Rest Is a Tracked Task (in progress)

This one is a candid disclosure rather than a finished feature. Ring's alarm catalog faithfully reproduces the controller's legend for the alarms that fire today, and it already shows real text for alarms the old system left blank or coded. A careful comparison against the plant's full Dutch legend, however, found 94 additional alarm numbers Ring hasn't loaded yet (alongside 46 whose wording could be freshened and 30 already correct). Until those are added, importing the old legend quietly skips the un-loaded ones rather than bringing them in.

Why it matters: a buyer gets a precise, honest picture of where coverage stands — today's controller-decoded alarms are correct, and the remaining entries are a known, purely additive task on the near-term roadmap, not a hidden gap. The fix simply adds the missing rows and tells the importer to insert new entries instead of skipping them.

Before and after: the old legend was itself fragile, with the "??" placeholders, the blank entries, and the bare "A0004" code described earlier. Ring already improves on it for the alarms it covers; the remaining numbers are a documented expansion in flight.

On a Plant Box Left Running for Days, a Live Fault Never Silently Disappears

This is the quiet engineering that keeps the alarm screen trustworthy on a display that may run for days or weeks without a restart. Ring keeps its default date window rolling forward each day, so a screen left on since last week doesn't get stuck showing only that first day — and an active fault is always shown no matter what date range you're viewing, so a weekend-old alarm that's still active can never slip off the list. Just as important, when the screen refreshes every two seconds, Ring leaves the list untouched if nothing actually changed, so an operator reading a troubleshooting card, holding a sort, or scrolled down doesn't get yanked back to the top twice a minute. The live, ticking age is deliberately ignored when deciding whether anything changed, so the counter itself never triggers a disruptive refresh.

Why it matters: on a round-the-clock plant, live faults are never hidden by a stale date filter, and the operator never loses the alarm they were actively working just because the screen ticked.

Before and after: these are modern robustness behaviors built for an application designed to run continuously. The old single-shot, DOS-era display simply never faced — or solved — these long-running pitfalls.

5. On the Floor & Getting Started

Running a glue-mixing plant safely depends on a few unglamorous things going right every single hour: operators always knowing where they are and how to get back, never missing a lost connection or a live alarm, and handing off cleanly to the next crew. Standing up a brand-new station has historically been just as fragile - a full day of a specialist rebuilding a 2002-era developer machine by hand. This chapter covers the features that make the day-to-day calm and predictable, and that turn commissioning a new site from a project into a few guided minutes - while letting the plant keep its own equipment names, in its own language, from the very first start.

Stand Up a New Station in Minutes, with No Specialist On-Site

When you bring a new station online, Ring greets the technician with a guided setup that feels like configuring a new smartphone. A ten-step wizard walks them through choosing a language, connecting to and live-testing the plant controller,

confirming the safe operating mode, setting up email and who gets alerts, loading sensible starting data, adding the plant's branding, and importing the plant's real equipment names - then it's finished. Every step is built so a bad entry can never trap the technician: if anything goes wrong inside a step, it simply moves on rather than blocking the whole setup.

This matters because it replaces what used to be a day of expert labor with a few minutes of guided clicks that a commissioning technician - not a software developer - can complete. The setup also remembers it's been done, so it won't nag on every restart, and it can be re-opened later from the configuration menu to fix any setting, without anyone ever touching hidden text files.

Before, standing up an old RS-360 station meant painstakingly reconstructing a 2002-era developer machine, with all the real configuration buried in cryptic text files, handled by sprawling code (one module alone ran to roughly seven and a half thousand lines) - plus a separate helper program. After: a single guided, fail-safe wizard takes a station from blank to live in minutes, and stays available whenever a setting needs a touch-up.

See Your Own Equipment Names on Day One, Imported Straight From the Old System

On the very first day, the plant sees its own familiar names - the actual Dutch tank, ingredient, and recipe names it has used for years - instead of generic placeholders like "Storage Tank 4." Ring reads the existing names, custom recipe names, and even alarm wording directly out of the old RS-360 system and loads them in. A preview screen shows a supervisor exactly what will change before anything is saved, so a person signs off and nobody has to retype dozens of fields.

The clever part is that this works in any language without anyone telling Ring what the words mean. It matches equipment by its position number rather than by reading the labels - so the tank in slot one is always the mixer and ingredient slot four is always the same starch, whether the old system called it "Natief Zetmeel" or "Domestic Starch." It reads the old files in their original character set and stores everything in a universal modern format that supports a wide range of languages and scripts. Each item in the preview is tagged as new, changed, or unchanged with a confidence level, and routine factory-default recipe names are skipped so they don't clutter the review.

The old way to move off RS-360 was to re-key everything by hand, because its data was locked inside an aging storage format and text files with no real way to export it. The new way keeps your own data: a purpose-built, preview-then-confirm importer pulls exactly that information across and writes it in so the screens refresh immediately - so the new system never feels like foreign software dropped on the floor.

Never Miss a Lost Connection or a Live Alarm, Because Safety Is Visible on Every Screen

A problem can never quietly hide. Every screen is wrapped in a frame of status indicators: a banner that shows whether the plant controller is connected, with a one-click "retry now" if it isn't; a deliberately loud banner whenever the system is in its safe operating mode; a full-screen alarm overlay that locks the menu until an alarm is dealt with; small confirmation pop-ups that acknowledge an action without interrupting work; and a bottom strip that always shows the clock and connection state.

The most quietly valuable piece is a smarter connection indicator with five distinct states. Instead of a crude "connected or not," it can tell "truly offline" apart from "reachable but momentarily idle" - which dramatically cuts false "the controller is down" calls that send people running for nothing. The alarm overlay sits above everything else on screen and locks navigation so a live alarm physically cannot be clicked past, while the action confirmations stay clear of the menu so they never accidentally swallow a click.

On the old RS-360 system a dropped connection could go unnoticed until something actually went wrong - it had an alarm form and a communications log, but no always-on connection banner, no safe-mode banner, no full-screen alarm that locks

navigation until it's handled, and no way to tell offline from idle. With Ring, the link state, the safety mode, and any live alarm stay visible on every screen at all times, and operators learn to trust the indicator because it doesn't cry wolf.

Hand Off Cleanly Between Crews Instead of Relying on Memory and Scribbled Paper

Shift changes are where context quietly gets lost. Ring gives crews a clean, written handoff: an operator can formally start and end a shift, see the current shift status, leave a written end-of-shift note for the next crew, and open a one-page summary of what actually happened during a shift - the batches that ran, the alarms that fired, and anything still running. The incoming crew inherits exactly the context the outgoing crew had.

The payoff is fewer "nobody told me" errors at the most error-prone moment of the day. The handoff notes build up as a durable, time-stamped trail rather than disappearing with a paper note, and the shift summary is read straight from Ring's own records, so it's always accurate.

The old RS-360 system handed off verbally or on paper; it had a basic shift start/stop control but no written handoff trail and no digital shift summary to hand over. Tellingly, an early version of Ring's own shift screen had been little more than a visual mock-up with no real saving. Now the shift control is a fully working, record-backed screen, and the written handoff trail and the "what happened this shift" summary are entirely new capabilities with no equivalent in the old system.

See Every Tank Group Live, Under the Plant's Own Names

Operators watch every part of the plant live, on dedicated screens labeled the way the plant labels them: the mixer (the make-ready tank), the storage tanks as a group, the use/doser tanks as a group, the agitator overview, the viscometer readings, and a detailed view of any individual tank. The screens check the controller about once a second, so the numbers are genuinely live, and the card headings show the plant's real names rather than generic ones.

This means the quality-critical thickness reading and any single feed tank are one click away rather than buried inside a group view. The viscometer screen in particular gives operators a dedicated place to log a manual reading, check it against the expected range, and see recent history - important because this viscometer reading is a bond-quality-critical measure for corrugating glue. Drill-downs into individual feed tanks let an operator focus on one glue line in detail; the first of these is fully live, with the remaining two built and waiting only on the plant team to confirm their names.

The old RS-360 system scattered all of this across many separate forms, and an earlier review had flagged the use-tank group, the viscometer screen, and the individual-tank drill-down as missing or only half-built. In Ring they are all fully built out and named the plant's way, so the whole plant reads as one coherent, familiar system.

Rename, Reconfigure, and Back Up Yourself, with No Developer Callout

Behind a supervisor or admin password, Ring opens a configuration area with more than two dozen screens that let the plant manage itself: rename tanks and groups, edit recipes, set shift names and times, manage inventory, set the controller's clock, back up the database, and much more. A handful of buttons are deliberately marked "coming soon" for advanced re-commissioning tools, but the core is all live today. Crucially, changes are saved permanently and survive a restart - so when a supervisor renames a tank, that name sticks.

The value here is independence: the plant's own supervisors handle everyday configuration without paying for a developer callout, and the password gate cleanly separates routine supervisor tasks from more powerful admin actions like a full factory reset.

On the old RS-360 system, configuration lived in editable text files, and even some early Ring screens had once been session-only - a tank rename, for instance, would vanish on the next restart. That gap is closed: edits now persist properly, so renames and configuration survive every restart.

Fix Faster and Stay Informed Off-Site, with Alarm Emails That Include the Fix Steps

Ring turns the plant into something you can keep an eye on from anywhere. A maintenance tech can get an alarm - along with the recommended first steps to fix it - on their phone before they even drive in. The owner can get the Monday-morning report by email without coming to the plant. Ring can email any report on a daily, weekly, or monthly schedule to whoever you choose, send a batch certificate when a batch finishes, and even post alerts into team-chat tools like Microsoft Teams or Slack. It's all set up in the wizard and a configuration screen, with a live "send a test" button to prove it works.

Several thoughtful touches make this genuinely usable rather than annoying. Email sending happens in the background so it never freezes the screen, and it automatically retries up to three times if delivery hiccups, logging every message. Alarm emails include the recipe and step that were active when the alarm fired, plus the first things to check - and built-in storm control means that if alarms flood in, recipients get a single digest instead of a hundred separate emails. The scheduler is careful never to double-send a report and can even catch up on a recent missed send.

The contrast is stark: the old RS-360 system left you completely blind off-site, with no email or notification capability at all - alarms and reports were screen- and print-only. Ring adds a full notification platform with reliable queued email, scheduled report attachments, batch certificates, and team-chat alerts - all genuinely new value with no equivalent in the old system.

Turn the Operator Station Into an Owner's Management Tool

Beyond running the plant, Ring gives the owner real management tooling the old system never had. There are screens for ingredient costs and currency, so you can see what a batch actually costs; a guide of what to do when each alarm fires, which a supervisor can edit; calibration and maintenance reminders shown on a color-coded due/overdue board; and a way to export the whole plant configuration as a portable profile to carry to a second station.

The benefit is that the station becomes a decision-making tool, not just a control panel. Cost visibility informs pricing and waste decisions, the editable alarm guide turns hard-won troubleshooting knowledge into something every operator can follow, the maintenance board makes sure calibrations don't slip, and the portable profile makes standing up a second site dramatically faster.

The old RS-360 system gave owners no management layer whatsoever - no cost module, no alarm guidance, no maintenance board, and no configuration export. Ring delivers each of these as a built, saved screen, transforming the operator station into a genuine management tool.

Operators Read the Screen in Their Own Language and Script (in progress)

Ring runs in thirteen languages - English, Dutch, German, French, Spanish, Italian, Portuguese, Turkish, Polish, Korean, Japanese, and both written forms of Chinese - and switches on the fly without restarting. English is the default, and the Dutch wording comes straight from the live plant's own old-system dictionary, so operators see the exact equipment terms they already know rather than a generic translation.

A nicely engineered detail: English is always loaded underneath as a safety net, so if any phrase in another language is ever missing, the screen falls back to English rather than showing a blank. Switching language repaints the screen instantly and remembers the choice for next time. Importantly, the language choice only affects what operators read - it deliberately never

changes how the system handles numbers, so recipe math and controller readings are completely unaffected by the display language. An installer can even flip languages live in front of a senior operator as an instant trust-builder.

The old RS-360 system did ship many language dictionaries and had a language menu, but changing language required a reload to take effect and offered no fallback for missing words. Ring switches instantly and safely. This one is honestly noted as in progress: the core spine - the wizard, home screen, menu, common buttons, and core equipment terms - is fully translated, while full coverage of every deep screen body is still being completed.

End the "Where Am I and How Do I Get Back" Confusion of a Deep Menu Tree

Ring replaces a sprawling menu structure with a single, always-visible menu bar down the left edge. Click a section like Reports or Setup and a small menu flies out beside it; the button for the screen you're currently on stays highlighted, and a Back button retraces your steps. Operators always know where they are and how to return.

This removes the disorientation and lost time of digging through layers of menus and stacked windows. The "you are here" highlight and the Back button are small touches that make a huge difference to how confident an operator feels, and the fly-out menus keep the main bar short and uncluttered.

Getting around the old RS-360 system meant navigating roughly ninety-seven separate forms through a 2002-era menu bar with hand-built per-tank entries and a clutter of stacked child windows, with no sense of where you currently were. Ring replaces all of that with one consistent rail, an active-screen highlight, and a reliable Back button.

Root-Cause Faster and Protect Your Data, with a Live Value Inspector and One-Button Backup

This is the toolbox for engineers and maintenance. Ring lets them self-serve troubleshooting right on the floor: read any live value from the controller without opening specialized engineering software, capture a one-shot system snapshot to send to support, watch a live feed of controller messages to see why communication hiccuped, set the controller's clock, run one-button database backup and restore, and - for admins only - perform a factory reset.

The standout tools are the live value inspector - type in the name of a controller value and instantly read it, no engineering software required - and the one-shot diagnostic snapshot that bundles up the system's state to screenshot and send to support. Together they cut the time and cost of figuring out what went wrong. The safe backup and restore, plus the password-guarded factory reset, mean the plant's data is protected and recoverable.

The old RS-360 system had a communications log and a clock-setting screen, but no live value inspector, no one-shot diagnostic snapshot, no built-in database backup or restore, and no factory reset. Ring keeps the familiar communications log and adds all of those, so the plant can root-cause faster and keep its data safe.

Get Hand-Logged Readings Off Paper and Into Reports, with Your Own Field Labels

The readings operators record by hand - shift readings, manual measurements - stop being trapped on paper and become reportable data. Ring provides five customizable fill-in forms, each able to hold up to eleven fields, and a supervisor can rename every field label to match the plant's own terminology so the data lines up with how the plant already talks. The captured readings then flow into Ring's reports.

The payoff is simple but significant: the plant keeps its existing hand-logging habits, but the information becomes searchable, reportable, and durable instead of sitting in a binder. The supervisor-controlled labels mean operators see exactly the prompts they expect.

The old RS-360 system had a hand-entry form too, but on paper-and-screen only - and an early Ring review found that this entire family of screens simply didn't exist yet in the new system. It has since been fully built: all five forms, the label-renaming setup, and the report, with the supervisor in control of every field name.

Set Agitators to Cycle Themselves Instead of Babysitting Paddles by Hand (in progress)

Each storage tank's agitator (its mixing paddle) gets its own control screen where an operator can turn it on or off, or put it in automatic mode with on/off timer presets so it cycles by itself. The timer settings only become active in automatic mode, so it's hard to set up by accident.

The benefit is that operators stop manually managing paddles tank by tank; once set, an agitator can run its own on/off rhythm without further attention.

The old RS-360 system offered this through a single per-tank form, and an earlier Ring review had found the new agitator screens were just placeholders. They're now full, working control screens for each tank. This one is honestly in progress on one point: live-plant observation suggests the controller itself may drive the agitator as a built-in step of the batch sequence, so the decision about which system holds the final say is being deliberately held until that's confirmed at the plant - the screens are ready either way.

Keep Operators Productive Through the Switch, with the Manual a Click Away and an On-Screen Cheat-Sheet

To help operators learn the new system without losing speed, Ring keeps the familiar things close. The original product manual stays one click away from the Help menu, a Contact Us screen gives a clear path to support, and pressing the help key brings up an on-screen cheat-sheet of the application's keyboard shortcuts.

The value is a gentler transition: operators aren't cut off from the reference material they know, they have an obvious way to reach support, and the shortcut overlay helps them pick up the new keystrokes quickly so they get fast on the new system sooner.

The old RS-360 system had a help entry for the manual, but no in-app keyboard-shortcut help and no contact screen. Ring keeps the manual handy and adds both the shortcut cheat-sheet and the support screen, so onboarding stays smooth.

A Home Screen Shaped by What Operators Actually Glance At Every Hour (in progress)

Ring's home screen is built around what operators really look at, rather than guesswork. During an on-site visit, a real operator stood in front of both the old and new screens side by side and pointed out the six things the old main screen shows at a glance - the current recipe, the make-ready tank weight, the borax and caustic weights, ingredient-addition progress, mix-time progress, and the mix-time-remaining countdown. The Ring home screen started out showing only two of those six; a screen-only update brought it to all six, with no extra load on the controller. The proposals were scored by people standing in for an operator, a plant manager, and maintenance, and the operator's wish-list items scored top marks.

The benefit is a home screen with genuine at-a-glance density - big numbers readable from across the room, a timeline of what needs attention next - tuned to what operators truly use hour to hour rather than what designers guessed they'd want.

The old RS-360 system does keep a status bar and a live recipe-step table on every screen, but that table forgets each step's duration the instant it scrolls away. Ring delivers the same floor-shaped awareness and can keep the step durations the old

system throws away. This one is honestly noted as in progress: a couple of structural refinements - an always-on run/hold/alarm strip outside the dashboard and a louder "data is going stale" signal when communication drops - are still being finished.

6. Your Data — and a Platform Built to Last

Every plant runs on its own history: which batches were made, when alarms fired, how much starch and caustic and borax were consumed. The trouble with RS-360 is that all of that lives trapped inside a 2002-era engine that no vendor maintains anymore, in a format almost nothing can open. This chapter is about taking that data back and putting it on a foundation built to last for decades. The theme is ownership and trust: you own every record outright, in a form you can copy and read with free tools; a power blip never costs you the batch you just finished; an interrupted upgrade never leaves your records broken; and the whole system is built so it can be maintained, improved, and re-skinned for years to come without being chained to any single vendor or developer.

Hand an auditor every record on day one, with no vendor in the room

All of Ring's data lives in a single, self-contained file: every batch, every alarm, your inventory counts, recipes, shift definitions, costs, and maintenance records. Under the hood that file is organized into roughly forty neatly structured tables, each one with its own built-in rules that reject bad data before it can ever be saved. Crucially, it is an open, well-understood format that dozens of free, off-the-shelf tools can read. There is no proprietary engine standing between you and your own records, and no vendor you have to call to get at them.

Why it matters: when a supervisor, an auditor, or a plant manager says "show me our data," the answer is to hand over one file. No special license, no support ticket, no waiting. And because the data is structured with strict internal safeguards, a software hiccup or a communications glitch cannot quietly shred a record the way the old system's fragile tables could.

The contrast with RS-360 is stark. The legacy system scatters the same information across many separate tables, each made up of a cluster of cryptic companion files, all reachable only through an abandoned 32-bit database engine that has no maintained driver anymore. The format is so dead that simply extracting the live plant's own data required our engineers to write a reader from scratch, by hand, just to crack the files open. Ring replaces all of that with one portable file that any free tool can read today and will still read in fifteen years.

Self-service backups, a verified restore, and Excel exports on demand

Ring lets you protect your own plant history without anyone's help. You can back up the entire database while the application is still running and the plant is still producing, no need to shut anything down. Before you ever rely on a backup, Ring can verify it is sound, checking the file's internal integrity and confirming the essential tables are all present. And you can export any table, batch history, ingredient usage, alarm history, shift records, straight to a spreadsheet with a single click, with a simple picker to choose the table, columns, and date range you want.

Why it matters: weekly self-service backups become routine instead of a project. A verified restore is your safety net, you know a backup is trustworthy before you ever need it, not after. And the inevitable "can you send me that in Excel?" request from a manager or auditor is answered in seconds rather than days. The restore process is careful, too: before swapping in a backup it quietly takes its own safety copy first, so the operation can never leave you worse off, and your hand-named exports are never swept away by automatic cleanup.

Against RS-360, the gap is simply night and day. The legacy system has no built-in backup or restore at all, its reporting relies on an abandoned print component, and the underlying data is locked in a dead engine that is awkward even to read. Ring

keeps the familiar one-click report-and-export workflow operators already know, but layers it over an open, portable store backed by a verified restore the old system never had.

Keep the batch you just finished through a power blip, and never freeze under load

Two everyday hazards quietly threaten a plant database: a sudden loss of power, and many screens hammering the same data at once. Ring is engineered against both. Every time the application touches its database, it commits each change all the way down to physical disk before moving on, so the batch you just completed is genuinely safe the instant it is saved, even if the power cuts out a second later. And the database is configured so that many parts of the app, dashboard, batch screen, trending, alarms, the historian, can all read and write simultaneously without ever throwing a "screen is locked" error.

Why it matters: nobody on shift waits on a frozen display while another part of the app is busy, and a power blip in the middle of production does not cost you the record of what you just made. These protections are applied uniformly, automatically, on every single connection to the database, so there are no gaps where durability or responsiveness silently lapses.

By comparison, the legacy engine offers no such guarantees, no durability on commit, no documented way to handle concurrent access, and a well-known tendency toward corruption. Recovering a corrupted table in the middle of a night-shift line stop is a genuine hazard with RS-360. Ring is built so that simply does not happen.

Upgrade without a database specialist, and never fear an interrupted update

When you install a new version of Ring, the database upgrades itself, automatically, in the right order, with no specialist and no hand-run scripts. It keeps an internal version number, looks at what is already in place, and applies only the new changes that are missing, one step at a time. The chain of upgrades to date spans eighteen versioned steps, adding things like operator identity, the cost module, alarm-email rules, report subscriptions, equipment runtime tracking, the maintenance log, and batch notes, each one stacking cleanly on the last.

The truly important part is what happens if something goes wrong mid-upgrade. Each individual step is wrapped so that its changes and its version stamp succeed together or not at all. If power is cut in the middle of an update, that step rolls back cleanly or safely re-runs on the next start, rather than leaving you with a half-finished, half-broken database. That closes a subtle gap where an interrupted upgrade could otherwise have left the records inconsistent.

RS-360 has no concept of this at all. Its table structure is frozen in the 2002 build, and any change means hand-editing tables in a dead toolchain with no safety net whatsoever. Ring's database versions itself, applies only what is new, and protects every step, so upgrades are a genuine drop-in and your plant history survives both power events and version changes intact.

Inventory totals you can trust, even after a controller reset

This feature fixes a subtle but serious counting problem. The controller signals each new inventory event with a counter that, like a car odometer rolling over, eventually wraps back around to a low number, and a controller power-cycle or reflash restarts that counter from the bottom too. The original design risked mistaking a brand-new, genuine inventory record for an old one it had already seen, quietly discarding it and permanently under-counting how much material the plant actually used. Ring solves this by also looking at the controller's own timestamp on each event, which lets it tell a true repeat (same event, correctly ignored) apart from a counter that has simply reset (a different moment in time, correctly recorded).

Why it matters: your usage figures for starch, caustic, and borax stay accurate across controller resets and signal glitches, with no silent drift that would understate real consumption. That accuracy is the foundation under any cost, yield, or efficiency number you later draw from the system, so it has to be right.

The legacy system has no comparable safeguard, its inventory correctness simply leaned on the old engine and 2002-era logic with no recovery tooling at all. The fix was also delivered carefully: the upgrade only triggers when the old vulnerable design is detected, preserves every existing row, and is wrapped so that even a crash mid-change can never leave the plant without its inventory table.

Your tank, ingredient, and recipe names, and your language, stay put forever

When you rename a tank, an ingredient, or a recipe to match how your plant actually talks, Ring stores that name in the database itself, not in a fragile loose configuration file. The same goes for the language you choose. Because these live in the central database, they survive reinstalls and rebuilds, and they update live across every open screen the moment you change them, instead of getting lost or going stale on one display.

Why it matters: the plant's own vocabulary, and its choice of language, become permanent and consistent everywhere, with no risk of a routine software rebuild quietly resetting them. Ring deliberately keeps the language and first-run setup choices in the database precisely because a software rebuild wipes loose local settings, so storing them centrally is what makes them stick. Every one of these reads is also designed to fail gracefully, returning a sensible default rather than ever crashing the application at startup.

RS-360 scatters this same information across a tangle of flat text files and legacy tables, fixes the language per-installation with no way to refresh it live, and devotes thousands of lines of code just to shuffling configuration around. Ring keeps your names and language in one durable place, refreshes them instantly across screens, and never lets a missing setting take the app down.

A look you can read from across the room, re-skinned as a setting, not a rewrite

Ring's entire visual style, its colors, spacing, buttons, and status indicators, is defined once in a single place, using meaningful names like "danger," "success," and "surface" rather than hard-coded color values scattered across dozens of screens. Calm, muted tints carry routine status; saturated, attention-grabbing color is deliberately reserved for live danger. The result is a coherent control-room look an operator can read accurately from fifteen feet away.

Why it matters beyond aesthetics: because the look is defined in one central place with meaningful names, the whole operator interface could be re-skinned, for example into a dark control-room theme for night shifts, by swapping that one set of definitions rather than rebuilding every screen by hand. The visual system was explicitly designed with that future flexibility in mind. It also resolves a well-known display annoyance where button text was being clipped, so labels read cleanly throughout.

RS-360, by contrast, is a 2002-era interface of fixed-pixel forms and hand-built per-tank menus, each screen styled on its own with no underlying design system at all. Ring gives you a single source of truth for the look that drives every screen consistently and keeps a future re-theme as a configuration change instead of a project.

Your fixes and features arrive sooner, with the damage contained

Ring is assembled from many small, swappable building blocks rather than one giant, tangled program. Each capability, an email rule, an alarm channel, a particular data table, is a self-contained component that an engineer can repair or replace without disturbing the rest. The application wires nearly forty of these data components and a long list of services together through a single, well-organized assembly point, with each piece's lifetime and role deliberately chosen and documented.

Why it matters to the owner: this is what makes enhancements arrive sooner and with far less risk. An engineer can change one service, say, how alarm emails are sent, without reaching into the screens or the rest of the system, so a fix in one place is very unlikely to break something unrelated elsewhere. Lower risk and faster turnaround on improvements is a direct, ongoing payoff for the plant.

RS-360 took the opposite approach: its screens were wired straight to raw database columns, and roughly twenty-one interdependent sub-programs shared memory with one another, so a change almost anywhere could ripple unpredictably almost everywhere. Ring's clean separation contains that blast radius, which is precisely what keeps long-term maintenance affordable and safe.

A screen that never freezes, and data that holds up to a security review

Two engineering disciplines underpin this feature. First, all communication with the controller happens on background workers, never on the screen's own thread. The app continuously gathers fresh readings in the background and the display simply shows the latest snapshot every half-second to two seconds, so the operator's screen stays smooth and responsive no matter how busy the controller is. If the controller or a network switch reboots, a background check automatically reconnects within about half a minute, with nobody needing to click "retry." Second, every single database query uses safe, parameterized commands, meaning data values are always kept strictly separate from instructions, so malformed or even malicious input cannot corrupt or expose your records.

Why it matters: you get a display that stays responsive under any load, self-recovers after a network or controller hiccup, and data integrity that stands up to a formal security review. A security pass over the system found the data layer clean across all of its nearly four hundred and eighty individual query points, with effectively no unsafe shortcuts anywhere.

RS-360 wired its screens straight to the database over a patchwork of bespoke communication links and shared memory, with no documented discipline around responsiveness or query safety, on top of an engine with no concurrency guarantees. Ring replaces all of that with a documented, modern model that keeps the screen alive and the data sound.

Faster, cheaper support, and months of unattended round-the-clock running

Ring is built to run for months on end, untouched, and to make the rare crash easy and cheap to diagnose. If the application ever does fail, it captures a small forensic report to disk first, and it does so from three different kinds of failure, so an engineer gets real evidence about what went wrong instead of a shrug. The app's own activity log is written by a dedicated background process, so even during a storm of controller communication problems the operator's screen never slows down waiting on the disk. And if someone deletes the log folder, the app simply rebuilds it and carries on.

Why it matters: faster, cheaper support when something rare goes wrong, because the diagnostic evidence is already sitting there waiting; a screen that stays responsive even through a prolonged communications outage; and the confidence to leave the system running unattended for long stretches without the logging quietly dying or the app slowly bogging down. There is even a built-in watchman that notices if the screen itself ever hangs. One real startup bug, where a brief dialog could silently kill the whole application, was found and fixed during this resilience work.

RS-360 offers a basic communications log and disk logging but nothing like this, no crash-report pipeline, no self-healing logger, no hang detector. Ring's instrumentation is exactly what turns "the system mysteriously stopped overnight" into a quick, evidence-backed answer.

No second copy fighting over the controller, and clear guidance if one is stuck

A subtle but dangerous mistake in any plant system is accidentally running two copies of the application at once, both trying to drive the same controller and write the same database. Ring prevents this entirely. Double-click the icon when Ring is already running, and instead of opening a second copy, it simply brings the running one to the front. This safety check runs first, before the app even touches the database or the controller, so a duplicate can never sneak in and cause a conflict.

Why it matters: there is never a confusing "I clicked it and nothing happened" moment, and there is never a split-brain situation where two copies corrupt data or double-drive the equipment. And if the running copy happens to be frozen, Ring

does not just sit there silently, it pops up a clear message naming the exact stuck program to close, so the operator knows precisely what to do rather than being left guessing.

RS-360 has no equivalent protection documented at all. This kind of guard, including its clear distinction between "already running and healthy" and "running but wedged," is part of the modern reliability layer that the legacy system simply lacks.

Any contractor can build it, and every release is checksum-verifiable

You are never locked to one developer's personal machine. Ring builds entirely with standard, freely available Microsoft tools, and the screen layouts are stored as plain, readable text files that an engineer can open, compare, and review like any other code. The exact versions of every outside component it depends on are pinned and recorded, so a build is repeatable. A packaging script produces a finished release together with a unique fingerprint, so you can confirm you are installing exactly the intended, untampered version. Broader automated build-and-test infrastructure is on the near-term roadmap (in progress), but the build is already fully repeatable on any standard developer machine today.

Why it matters: any competent contractor can pick up, review, build, and ship the software, which protects you from being held hostage by a single developer, and the verifiable fingerprint on each release reduces the chance of ever installing the wrong or a tampered build. That is genuine long-term insurance for the plant's most important software.

The contrast with RS-360 is almost hard to believe: the legacy project has no build instructions, no documentation, and hardcodes one long-departed developer's personal folder paths, and its screen layouts are stored as nearly a hundred unreadable binary files that cannot be compared or reviewed. Rebuilding it at all would mean painstakingly reconstructing a 2002-era machine from scratch. Ring replaces that dead end with standard tooling, readable layouts, pinned versions, and a checksummed release, with even richer automation still to come.

7. Why Replace It Now

The current operator system still runs the line every day — and that is precisely the argument for moving now. The old system works only because it has never had to change, and every month it sits on a 2002-era foundation that nobody can buy, patch, or hire for, the cost of the eventual forced switch climbs. The theme of this chapter is simple: replace a single hidden point of failure calmly, on your own schedule, while the plant is healthy — and turn an open-ended, growing liability into a known, bounded one.

Retire Your Single Biggest Hidden Risk Before It Forces the Decision For You

What it does: Ring lifts the entire operator experience off a foundation that has no support path and rebuilds it on mainstream, modern software. The old screens were built with a development toolset from 2002 that is no longer sold, no longer updated, and no longer something you can hire for. There is no one to call when a future Windows update breaks it, and no second person who can step in if the original developer retires. Ring replaces that with the same kind of software that runs ordinary business applications everywhere today.

Why it matters: This is not a rescue mission for something that is broken — it is insurance you buy while the house is not on fire. Because the old system keeps running the line during the transition, you move deliberately, one screen at a time, with no crisis pressure. The payoff is a plant that stays supportable for the next twenty years instead of sitting one retirement or one corrupted file away from a hard stop with no vendor to call.

Old vs. new: Today you are one event away from being stuck — an unsupported toolset, a fragile data store, a build nobody can reproduce, and a shrinking pool of people who even know the old technology. After the move, you are on widely-

supported software with a clean, modern internal design, changed over gradually and reversibly so the shift is controlled rather than forced.

Own Your Recipes, Batch History, and Alarm Log — Instead of Holding Them Hostage in a Dead System

What it does: Your recipes, your batch records, your alarm history, and your inventory currently live inside a 2002-era data engine that no modern tool can open, that has no repair utility, and that can quietly corrupt with no way to recover. Ring moves all of that into a modern, self-contained database you can read, back up, and export yourself, with built-in integrity checking and a one-click export to an ordinary spreadsheet file — including a picker to choose exactly which tables and columns you want.

Why it matters: The stakes here are concrete. The plant's real data is substantial — over 1,400 batch records, nearly 17,000 individual recipe-step entries, almost 3,000 alarm occurrences, and hundreds of shift records. On the old engine, a single corruption event on the night shift could halt production with no recovery path and no vendor to call. The old data was so locked away that the team had to build a brand-new reader from scratch just to get it out, because no maintained tool to open those files exists anymore. On Ring, the same data is portable, inspectable, and protected — so if anyone ever asks "show me our data," the answer is available immediately.

Old vs. new: Before, your operating history was trapped in a fragile, dead format that was awkward even to extract. After, it lives in a modern, portable database with integrity checks and automatic backups, plus one-click export — so the plant genuinely owns and can read its own records, permanently.

Make Changes Any Modern Contractor Can Do — Not a 2002 Build Wired to One Aging PC

What it does: Right now, the ability to rebuild the old system is tied to one specific developer's machine from 2002. The project files literally hard-code one person's folder paths from that PC, there is no written instruction anywhere for how to rebuild it, and the screens themselves are stored in a sealed format that can't be meaningfully reviewed or compared. Ring is built with standard, current tools from plain, readable files, so any competent contractor in the modern software world can pick it up.

Why it matters: This removes the single most dangerous form of dependency — the kind that walks out the door when one person retires. The scale of the old tangle is striking: the legacy system is twenty-one interlocking sub-programs talking to each other through shared memory, with hundreds of underlying files and two single files running past 5,000 and 7,000 lines each. Most of its screens — 88 of 97 — are stored in a sealed form that can't be properly inspected or compared, meaning even a careful expert can't safely review a change before it ships. On Ring, every screen is a readable file, backed by written, repeatable build and release procedures, so changes ship with confidence instead of guesswork.

Old vs. new: Before, making any change realistically meant reconstructing a specific 2002-vintage developer setup on one machine, then editing screens you can't even review. After, a standard contractor opens readable files in current tools, makes the change, reviews it normally, and ships it — slow, expensive, fragile work becomes routine.

Buy Bounded, Shrinking Risk — and Stop Paying for Unbounded, Growing Risk

What it does: This is really the business case itself, framed honestly. Instead of overselling productivity miracles, the case rests on a straightforward asymmetry: the risk you carry on the old system is open-ended and growing, while the risk of moving to Ring is known, scoped, and shrinking by the day.

Why it matters: On the legacy side, the downsides have no ceiling and no vendor behind them — a data corruption with no recovery tool, a Windows update that breaks an unpatchable old system, or the retirement of the one person who understands the build. On the Ring side, the remaining work is fully catalogued with a plan, including a single clearly-scoped outside line item — restoring the control logic that drives the machinery, quoted precisely per site and roughly in the range of a modest engineering engagement rather than a blank check. The honest framing is the selling point: the value is real risk avoidance and supportability today, with the bigger productivity gains arriving in later releases — and no invented payback number or uptime figure that would erode your trust the moment it was questioned.

Old vs. new: On the old platform, the risk effectively is the whole argument — it can't even produce a list of its own problems, and emergency recovery has nobody to call. On Ring, every remaining item comes with a plan and, where outside work is needed, a known number instead of a surprise.

Verify Every Claim Against the Real System Before You Commit

What it does: This is the promise that you don't have to take the pitch on faith. The full positioning guide ties every claim back to the actual software, names the genuine remaining gaps plainly, and explicitly forbids inventing flattering numbers.

Why it matters: For a buyer who wants to kick the tires, this is reassurance that what you see is what you get. The guide openly separates what has already shipped from what is still on the roadmap, and it states the three honest gaps up front: a per-person named sign-in history is not in place yet, the polished modern visual refresh is slated for the next release, and the control logic that drives the machinery still needs its own scoped switch-over window. It also bans the usual sales inflation — no fabricated uptime, payback period, or customer count — and it states plainly that the system has already been proven running against a real controller on a test bench. Whatever an operator is shown in a demonstration is something that is actually built and safe, because the demonstration runs with all commands to the machinery deliberately held back, and the old system stays installed as a fallback throughout.

Old vs. new: On the legacy side there is simply nothing to verify — no written documentation, no build notes, nothing that explains itself. On Ring, you get a grounded, cross-referenced guide with an explicit list of what's done and what's pending, so a careful buyer can confirm the story against the real thing rather than trusting a brochure.

8. A Reliable, Honest Connection

Everything in a control room rests on one quiet assumption: that the numbers on the screen are real and current. The old operator screen could not honor that assumption. A green light only told you the wire was plugged in, not whether the plant was actually running, and a frozen screen could keep showing confident, dead numbers for minutes. This chapter is about a connection you can trust. Ring knows the difference between a controller that is unplugged, frozen, or genuinely running; it refuses to show stale data as if it were live; it heals itself after a power blip; and it rides out a network hiccup without flooding the plant with errors. The result is fewer false alarms, fewer wasted maintenance calls, and operators who can act on honest information.

Honest plant status — Ring tells unplugged, frozen, and running apart

Ring continuously watches a tiny counter inside the controller that ticks upward only while the controller is actually executing its program. From that one steadily advancing number, Ring can distinguish three very different situations that a simple "connected or disconnected" light cannot tell apart: the controller is unplugged and not replying at all; it is reachable and answering, but its counter is stuck (meaning it is alive on the network but its control program has stalled); or it is genuinely, healthily running. Ring states which of these is true in plain language. It checks roughly every second and a half, and if the

counter sits unchanged for a few seconds it flags the link as going stale; if it stays frozen past about twelve seconds, Ring calls out that the controller is reachable but wedged.

Why this matters: the most dangerous failure in a plant is not an obvious outage — it is a controller that has quietly stopped running its program while still answering the network, so the screen keeps showing happy, live-looking numbers that are actually frozen. Ring catches exactly that case by name. Just as important, all of its timing is measured against an internal stopwatch that ignores the wall clock, so someone manually setting the controller's clock — or an automatic time correction nudging it in the middle of a batch — can never accidentally hide a real loss of communication. The practical payoff is fewer false "the system is down" calls and faster, more accurate fault isolation.

The contrast with the old screen is stark. With the old system, a frozen or paused controller that still answered the network could leave the screen looking perfectly normal — it had no concept of these in-between states, so operators could trust numbers that were already dead. Ring names the reachable-but-not-running condition explicitly, exposing a whole class of failure the old system could mask.

A command never lands in a frozen controller

When live control is enabled, Ring sends a command to the controller only at the exact moment it has just confirmed the controller is genuinely running. If that internal heartbeat so much as hesitates, commands wait rather than firing into a possibly-frozen controller. At the same time, reading from the controller is allowed to continue through a brief hiccup, so a momentary blip never makes the whole screen flicker to "disconnected."

This is a deliberate, carefully drawn line. Ring uses two different standards: a forgiving one for reading, so a few-second stumble does not freeze every display on the screen, and a strict one for sending commands, which fire only when the controller is provably advancing its program. Every operator command — starting a batch, selecting a recipe, changing a target value, running the mixer, silencing an alarm — is therefore delivered only to a controller that is demonstrably alive. Nothing gets lost or misapplied in a wedged or in-between controller, and the screen stays responsive through minor jitter.

The old screen issued its commands with no such precondition. It could send an instruction to a controller that was not actually running its program, with no check tying the command to a verified, advancing controller. Ring turns "only command a controller that is genuinely alive" into a guaranteed rule — and keeps reading resilient at the same time.

Operators never decide on a frozen number that looks live

Ring will not paint an old reading as if it were current. If the controller stops answering, or only partially answers, the affected numbers are flagged out-of-date instead of being silently re-stamped as "now." Every reading Ring captures carries the exact time it was taken, and Ring checks that age before trusting it.

The clever part is what happens during a partial outage. If Ring's normal bulk read of the controller's data fails, it falls back to reading a smaller core set of values one at a time. If even those all fail, Ring deliberately does not update its stored values and does not refresh the timestamp — it would rather show nothing than re-stamp minute-old data as fresh. And when only some values come back, Ring knows which ones are genuinely current and which are not, so a reading that was not actually refreshed still correctly shows as stale even though the read attempt itself just happened. During a blip, a live value on screen simply fades to a dash instead of showing a frozen-but-green number.

The payoff is trust. Tank levels, weights, temperatures, and batch-step status are always either current or visibly flagged as out-of-date — eliminating the single worst trap, where convincing-looking numbers are actually minutes old. A typical legacy screen just caches the last reading and keeps displaying it, with no notion of whether any individual field is still fresh; a communications outage can leave it showing believable but dead values that an operator acts on. Ring's refusal to re-stamp old data is an explicit defense against exactly that.

No more morning restarts — it reconnects on its own

If Ring starts up before the controller is ready — common after a power blip or first thing in the morning — or if the network drops mid-shift, Ring does not give up. It keeps quietly re-checking every thirty seconds and reconnects on its own the instant the controller comes back, with no operator clicks required. There is also a "Retry now" button right on the status banner if someone wants to force it immediately.

This is genuine set-and-forget reliability. Ring's startup check for the controller runs quietly in the background, so it never freezes the screen while it is waiting, and if the controller is not there yet, Ring schedules a patient background retry that reruns the exact same connection routine until it succeeds, then quietly stops. It is careful not to trip over itself — a slow check never overlaps with the next attempt — and a known crash that came from an abandoned connection attempt has been closed off.

The before-and-after is simple. The old approach was: try once at startup, and if communication was not there, give up — leaving an operator to step in to get the screen back. Ring heals itself every thirty seconds with zero clicks, plus that one-button manual retry on the banner.

A frozen screen cannot lie to your operators (in progress)

This feature closes one of the worst failure modes any operator screen can have: the display locks up but keeps showing old data as if it were live. A background safety watch inside Ring constantly checks that the display is still responsive — it sends the screen a quiet "still there?" nudge every five seconds and notes whether the screen actually answers. If the screen ever stops answering for a full minute, Ring knows it has frozen, and it writes a detailed record of the event for later review.

Why this matters: a frozen screen still looks alive, which means an operator can stare at stale pixels for minutes without realizing nothing is updating. Ring turns that silent, invisible failure into a detected, recorded, diagnosable event — giving the plant a clear trail to understand what happened and a path to automatic recovery. Detection is already in place and working; the automatic restart that would relaunch the screen after a freeze is built and ready, deliberately kept switched off until it has been confirmed on the actual plant-floor computer, so a freeze for now produces a recorded, recoverable event rather than any risk of an unintended shutdown. (In progress: detection shipped, automatic recovery pending final on-site confirmation.)

The old screen had no awareness of its own responsiveness at all — a hung old screen was discovered only when a person happened to notice that nothing on it was changing. Ring builds in an internal safety watch that turns a silent failure into one that announces and records itself.

See exactly what flows to the controller — and shed the 2002 driver

Ring talks to the controller using a modern, open, well-supported communications method over ordinary Ethernet networking — not a proprietary, vendor-specific driver baked into the old screen. On top of that openness, Ring gives you total visibility: every single read and write to and from the controller can be watched live on screen and saved to a daily log file, so operators and engineers can see exactly what is flowing in both directions.

The live message view keeps the most recent several hundred exchanges in memory at all times, and can additionally write a day-by-day record to disk for later review. Those disk writes are handled quietly in the background so they never slow down or interfere with talking to the controller, and old logs are automatically cleared as the days roll over. One nice touch: when Ring is in a mode where it is deliberately not sending commands, the monitor still records what it would have sent — so you can see Ring's intended actions even when it is holding back. This is a faithful modern remake of the message monitor operators knew from the old system, but built on an open foundation.

The legacy contrast is significant. The old screen's link to the controller is a bespoke, decades-old proprietary communications module welded into the screen, still carrying compatibility patches left over from an even older generation of plant equipment

— a 2002-era dependency that has to be kept alive indefinitely just to keep the screen talking. Ring connects directly to the controller's data over an open, industry-standard method, with a live feed of every read and write, a saved daily record, and an inspector for maintenance — and no fragile, screen-baked proprietary driver to maintain.

No error storms, no network overload — it rides out an outage and snaps back

When the controller stops answering, Ring does not hammer the network or flood the logs with errors. It automatically slows down how often it checks in, then speeds right back up the instant the controller responds again. At startup it also deliberately spaces out its connections so it does not overwhelm a modestly-sized controller all at once, and it runs cleanly with no controller attached at all — handy for office demonstrations or operator training.

Under failure, Ring's various background readers ease off after a couple of failed attempts — stretching their check-in intervals out to several seconds — and instantly return to their fast, live rhythm on the first successful response. At startup, the heavier per-tank data requests are fanned out a fraction of a second apart rather than all firing at once, which avoids briefly overwhelming the limited number of conversations a controller can hold at the same moment. Ring is also careful never to start a new check while a previous one is still finishing, and it cleanly handles being shut down in the middle of a request, so reconnecting never leaves loose ends.

The payoff is stable, quiet operation: no error storms during an outage, no network overload on a modest controller, an instant snap-back to live updates on recovery, and the freedom to run the screen with no controller present for demos or training. The old monolithic system had no documented way to slow down during an outage, stagger its startup, or run without a controller — so an outage could turn into noise and instability. Ring centralizes all of this start-stop and pacing logic in one place, something the old screen simply did not provide.

9. Where the Work Stands

We believe the most trustworthy way to talk about a new system is to name both things plainly: what is finished and working today, and what is still being completed. This chapter does exactly that — first the capabilities that are already built and in use, then an honest, plain-language list of the pieces still being finished. Nothing buried, nothing oversold.

New on-floor diagnostics and self-service data backup the old system lacked

What it is. Ring puts a small toolkit of self-service tools directly into the operator application. With one tap, a floor supervisor can capture a full system snapshot to send to support. A live value reader lets them look up the current state of any single point in the plant — the position of a valve, the reading from a tank — just by typing in its name. A live message feed shows the back-and-forth between Ring and the plant equipment as it happens, so a communication hiccup is visible the moment it occurs. There is also a tool to set the equipment's clock, a one-button backup and restore for the system's records, and an administrator-only "factory reset."

Why it matters. In the old world, this kind of deep troubleshooting meant calling in an outside engineer with specialized equipment-programming software and waiting. That turned routine questions into multi-hour, billable callouts. Ring moves most of that work onto the floor, where a trained supervisor can answer the question in minutes — and the one-button backup means the plant's history is protected on a schedule instead of left to chance.

Before and after. The old operator screen offered a basic message log and a clock-setting form, but nothing to read a live value, capture a support snapshot, or back up its own records — and the records themselves were stored in a fragile, decades-old format that could fail without warning, with no built-in backup of its own. Ring turns floor troubleshooting from an outside-engineer job into a built-in, in-application one, and adds proper, supervisor-protected backup and restore on top.

A 13-language interface operators can switch on the fly

What it is. Ring ships with a 13-language interface — English, Dutch, German, French, Spanish, Italian, Portuguese, Turkish, Polish, Korean, Japanese, and both written forms of Chinese — with English as the default. The Dutch was seeded directly from the plant's own existing word list. Operators can change the language live, right on the screen, with no restart and no reinstall. Today the translation covers the most-used parts of the application: the first-time setup guide, the menus, and the high-traffic operator screens.

Why it matters. A worker reading the screen in their own language makes fewer mistakes and works with more confidence. Just as important, the language switch was built so it can never touch the way the system handles numbers — so changing the display language carries no risk to the batch math or to how Ring talks to the plant equipment. If a word in a chosen language has not been translated yet, the screen quietly falls back to English rather than going blank.

Before and after. The old system let you pick one language when it was first installed, and then it was frozen there forever — no switching, no adding to it later. Ring matches that multilingual reach with a modern engine that switches live and can be extended one screen at a time as the remaining engineering screens are translated.

What's still being finished

We would rather you hear these from us than discover them later. None of the items below blocks daily operation; each is a known, scoped piece of finishing work:

- **Full screen-by-screen translation.** The language engine is live and the main operator screens are translated, but the deeper engineering screens — the mixer, the storage tanks, the dosing tanks, and the recipe editor — are still in English while their text is keyed in. This is a deliberate choice: we would rather fully translate a screen than leave it half-done.
- **Per-person sign-in history.** Ring records what was done, and it carries forward the supervisor-and-operator roles from the old system. What it does not yet do is tie each action to a specific named person, because individual sign-ins are not built yet. The underlying record-keeping is already in place; switching it on is a defined piece of work we would scope with you based on how much named-person tracking your industry expects.
- **In-application user and password management.** Creating users and changing passwords is marked "coming soon" inside the app today; for now it is configured during setup rather than self-served on the floor.
- **A few analytics and detail screens being wired in.** A small number of capabilities are fully built but not yet placed on the operator's screen — the live, moving equipment diagram on the mixer detail view is one example. These are finishing-touch items: built, tested, and waiting to be connected.
- **Automatic batch-completion email.** Ring can email a per-batch yield summary to an owner or quality lead; today it is switched on in the system's settings during setup rather than with an in-app checkbox. The convenience toggle is a planned follow-up.
- **A short scoping conversation with your team.** A few facts are best settled up front rather than on installation day — confirming the full set of tanks at your site and importing their real names, and how much named-person tracking your industry expects, which decides how much of the per-person sign-in work applies. We flag these as questions for your team, not assumptions on our part.

Bottom line: 87 new capabilities, all on the same trusted controller — each one built to close a real gap in the 2002-era system or to extend what it already did. This guide is the full detail behind the short pre-meeting brief.
