

ANTON by openEXPERT

The ANTON family whitepapers · Autumn 2026 · Paper 9 of 11

From Once a Year to All the Time

Why checks, validations and monitoring can follow change instead of the calendar — and what that changes for people

> *Risk does not keep an annual clock. When a check costs hours instead of weeks, the question is no longer how often we can afford it, but what should trigger it.*

Version 1.0 · 10 December 2026

Created by Daniel Bardun · FutureChain AB

Open source ANTON is EUPL · github.com/altspace-hub/anton_desktop

See it live ai-portal.futurechain.eu (showcases with fictional data)

Before we start

Where we left off

Last week closed Act 2 of this series, the method. Over four papers we looked at how AI is made good and safe for regulated work: seven layers of knowledge around the model, a setup that matters more than the model's size, your knowledge kept on your side, and atoms — the small, owned, checkable pieces of learning each finished task can leave behind, gated until they have earned a place in the prompt.

This week Act 3 begins: **the new way of working.** The first two acts asked how to make AI help trustworthy. This one asks a different question. Once AI help *is* trustworthy, what should we do differently? In the language of the first paper, this is the time gap — and this act is where the series turns from how the work is made safe to where it is heading: compliance that follows change rather than the calendar, with people spending their time on judgement rather than on compiling.

The most common answer I hear is: *the same things, faster*. The same annual cycle, the same reports, the same calendar — just with less overtime in November. I understand that answer. I think it misses the most important opportunity in the whole subject.

A change of rhythm, not of headcount

In the foreword to our first whitepaper, *"The Starting Point"*, I wrote a paragraph that has stayed with me ever since, and it is the seed of this paper:

> **"When it can happen in a day, it can happen every quarter. Or every month."**

The full passage went like this: *"A gap analysis that used to take eight weeks and happened once a year can happen in a day or two. And when it can happen in a day, it can happen every quarter. Or every month. Suddenly you're not doing an annual check-up — you're running continuous diagnostics."*

Let me be precise about what that is. **It is my view, drawn from my own experience of how long these pieces of work take and why — not a measured result.** We have not run a controlled time study, and I will not quote a percentage. But I have spent over fourteen years in banking, financial crime prevention, regulatory work and tech consulting, and I have watched the same annual clock tick in every one of those settings. The argument does not depend on the exact numbers. It depends on one observation: **the clock was set by cost.** If the cost moves, the clock can move too.

In the first paper of this series I called this *a change of rhythm, not of headcount*. This paper is where that idea gets its full treatment: what continuous, AI-supported compliance looks like when it is built carefully, what we have built towards it, and — the part I care about most — what it changes for the people who do the work.

In this paper

- **Part 1, the problem:** the annual clock, three scenes, and why a finding found late is not a warning.
- **Part 2, our thesis:** from "how often can we afford it?" to "what should trigger it?" — and the role of people.
- **Part 3, our take:** seven principles, the triggers, and what we have built across the family.
- **Part 4, a worked example:** one threshold change, five reactions.
- **Part 5:** what this means for you on Monday, and next week.

Who this is for

Heads of compliance, MLROs, FCP leads, risk managers, model validators, internal auditors and outsourced compliance desks — anyone whose year is shaped by an annual plan. And the managers they report to, because the most important part of this paper is about how people's time is spent.

Part 1 · The problem

The annual clock

Most compliance and risk control runs on an annual clock. You know the rhythm without looking at the plan.

The BWRA (the business-wide risk assessment) once a year. The model validation every year or two. The policy review on its anniversary. Independent review on a three-year plan. The annual compliance plan — the *årshjul*, as we say in Sweden, the year-wheel of dated duties — turning steadily from January to December.

There is nothing foolish about this. **The clock was set by cost.** Each of those checks takes weeks of skilled people's time: gathering documents, reading them, testing, writing, reviewing, presenting. Skilled people are scarce and expensive. So you do each check as often as you can afford to — and for most of them, that means once a year.

But risk does not keep an annual clock. A new product launches in March. A threshold is changed in May. A regulator publishes a guideline in June. A key control owner leaves in August. And the annual check in November finds what went wrong in spring — by which time it is already a finding, not a warning.

> **A finding found late is not a warning. It is a record of how long nobody looked.**

Three things the calendar missed

These are composites, as always. No names, no clients.

Scene 1: the stale BWRA

A payment institution's business-wide risk assessment is approved by the board in January. It is a good assessment: thorough, well evidenced, signed.

In April, the institution launches a crypto on-ramp for merchants. The product team did the work properly — a new-product review, a risk sign-off. But the BWRA itself is not updated until the following January, because that is when the BWRA is updated.

For nine months, the board's picture of the firm's risk is wrong. Not catastrophically wrong — just quietly, structurally out of date. And here is the uncomfortable part: everyone knows it. The MLRO knows. The product owner knows. The board, if asked directly, would probably guess. Nobody is negligent. The calendar simply says *not yet*.

Scene 2: the threshold nobody re-checked

After a busy quarter, a transaction-monitoring threshold is lowered to bring alert volumes back under control. The change is reasonable, approved and documented.

The model validation was last year. The next one is next year. Nobody re-tests the change, because re-testing is the validator's job, and the validator's calendar is full until then. Fourteen months later, the validator finds it. The change may have been fine. Nobody knows, because for fourteen months nobody looked.

Scene 3: the guideline that sat in an inbox

A new EBA guideline is published. Someone in the regulatory change team spots it and forwards it. It lands in three inboxes. Each recipient assumes one of the others has it in hand.

The policy is updated at the next annual review, seven months later. Again, nobody did anything wrong in the narrow sense. The process said *annual review*, and the annual review happened.

What the scenes have in common

Look at the three scenes together and something stands out. In none of them was the change invisible. The product launch was approved. The threshold change was documented. The guideline was forwarded. **The event was seen. It just did not trigger anything.**

That is the real problem with the annual clock. Not that people do not notice change — they usually do — but that noticing a change has no consequence until the calendar comes round. The work is organised around dates, not around events. And so the organisation carries known, unexamined change for months at a time.

There is a second cost, which is quieter. When the annual check does come round, it is enormous. Everything that changed in the year has to be found, read and assessed at once. The team spends weeks compiling and checking, and very little time on what the results mean. The yearly scramble crowds out the judgement it was supposed to inform.

Why this matters now

The workload is about to peak. The AMLR applies from 10 July 2027, and AMLA, the new EU authority, is publishing standards and guidance as it goes. That is not one change; it is a stream of them, landing over months. A once-a-year cycle is a poor fit for a stream.

And the cost of a check is changing. With the right setup — the method in the module, the knowledge in layers, the numbers computed by rules, as earlier papers in this series described — an AI-supported first pass of a check can be prepared much faster than the same pass done entirely by hand. The direction is clear enough to change the question.

Part 2 · Our thesis

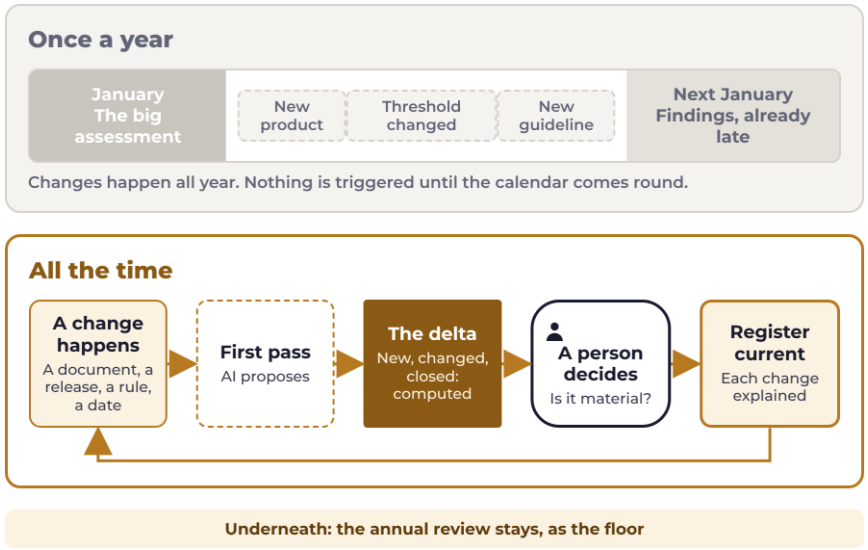
From one big check a year to many small, explained ones.

The thesis

When a check costs hours instead of weeks, the right question changes. It is no longer *"how often can we afford it?"* It is *"what should trigger it?"*

Checks, revalidation and monitoring should run **when something changes** — a document, a release, a regulation, a threshold, a date — and on a sensible **minimum interval** underneath, so nothing goes unexamined for too long. And people should decide what the results mean.

That last clause is not a courtesy. It is the point. Continuous compliance done badly is a machine generating alerts that nobody has time to read. Continuous compliance done well is a stream of **small, explained changes**, each one prepared by the machine and decided by a person.



From one big check a year to many small, explained ones.

Why we believe it

1. The cost argument has moved. This is the foreword's argument, and it is the one everything else rests on. A check was annual because it took weeks. With the right setup around the model, the first pass of many checks takes a fraction of that time, so it can run more often. You do not need a percentage to see the principle. If something gets much cheaper, you can afford to do it at the moment it is useful, rather than at the moment it is scheduled.

2. Most change is visible as an event. A new version of a document. A new product. A changed parameter. A published regulation. An action that has passed its due date. A date on the calendar. Each of these is something a system can see — and when a system can see an event, it can start a check. The three scenes all had a visible event. What they lacked was a trigger.

3. "What changed since last time?" is a better question than "what is the state of everything?" Re-reading a whole report takes days. Reading what is new, what has changed and what has closed since the last cycle takes minutes. Continuous work is mostly about these *deltas* — the differences between one cycle and the next. That is what makes it bearable: you are not re-doing the annual assessment every month. You are reading its changes.

4. People move up the chain. When the machine prepares the first pass on each change, people spend their time where they are irreplaceable: on judgement. Is this change material? What do we do about it? Who decides? That is the work that was always crowded out by the yearly scramble — and it is the work the foreword's backlog was full of.

An analogy: the chart and the instruments

Think about how a ship is kept on course. Before a long voyage, the navigator plots the route on a paper chart: a careful piece of work, done once, checked and signed off. And every so often the ship comes into harbour for its inspection — hull, engines, safety equipment, a list of findings. Both are valuable, and nobody suggests abolishing either.

But no navigator relies on the chart alone once the ship is at sea. Wind and current push it off its line from the first hour, a little at a time, and the plotted course says nothing about that. So the ship has instruments: a compass, a position fix, a depth sounder. When the ship drifts, they show it — today, not at the next harbour. The harbour inspection is the **floor**: the thorough look that happens no matter what. The instruments are the **trigger**: the small, specific reading that changes when something actually changes.

Most compliance functions today have the chart and the harbour inspection, and no instruments. This paper is about adding the instruments — and, crucially, about making sure each reading is *explained*, so the navigator knows what it means and decides whether to change course. A bridge full of unexplained alarms is worse than a quiet one.

The role of people

This is the part of the paper I most wanted to write, because it is where the series' first promise — AI as a coworker, not a headcount cut — becomes concrete.

When checks follow change instead of the calendar, the shape of a compliance professional's week changes. Not the number of people. The shape of the work.

The mindset shift

- **From doing the annual check to deciding on a stream of small, explained changes.** The machine prepares; a person judges.
- **From one big report to a register that is always current,** with each change explained and dated.
- **From finding problems late to being warned early,** and having time to act before it becomes a finding.
- **From the yearly scramble to a steady rhythm,** with fewer surprises for the board.
- **What does not change:** a named person decides, four eyes for anything material, and AI never sets the number.

Think about what that means for an MLRO. Today, a large share of her year goes into *producing* the annual BWRA — collecting, reconciling, drafting, chasing. In a continuous model, the assessment is kept alive between cycles: each new product, each changed indicator, each new piece of guidance arrives as a small, prepared change, and her job is to decide what it means for the firm's risk. The annual BWRA does not disappear; it becomes the moment she signs off a picture she has been keeping current all year. **Less compiling. More deciding.**

Or think about a model validator. Today, the validation is an event: a big piece of work every year or two. In a continuous model, a change to the model — a re-tuned threshold, a new data source — starts a targeted revalidation from the last signed result. The validator spends her time on the question that matters: *is this change safe?* — not on reconstructing what the model looked like last year.

None of this happens by itself. It needs new skills: reading deltas well, judging materiality quickly, saying *"this one matters, these four do not"* with confidence and a reason. It needs managers to protect the time it frees, so that it goes to the backlog — the better control, the remediation, the conversation with the product team before the launch — rather than being quietly absorbed into more of the same. **A change of rhythm only pays if someone decides what the new rhythm is for.**

What we are not claiming

We are **not** claiming measured time savings. The foreword's "eight weeks to a day or two" is my experience and expectation, not a measurement, and we have not run a controlled time study.

We are **not** claiming that any organisation runs continuous compliance with the family today. The products described in Part 3 are on public showcases with fictional data, or in the products' main code.

We are **not** claiming the annual review goes away. It becomes the floor, not the whole house. Where a regulation sets a minimum interval, that interval still applies.

And we are **not** claiming that anything monitors itself. Nothing in the family acts on its own. Every check prepares something for a person; every action that leaves the system waits for one.

Part 3 · Our take: change as the trigger

Seven principles

1. **Trigger on change, with a floor.** A check runs when something relevant changes, and at least on a minimum interval — the regulatory one, where there is one.
2. **Deltas, not reruns.** Each cycle shows what is new, what has changed and what has closed since the last one.
3. **The last run is the baseline.** A re-assessment follows a frozen earlier one and never changes it; a revalidation starts from the last signed result.
4. **The client's word is not a level.** *"Implemented"*, from a client or a business unit, means *implementation claimed* until it has been tested.
5. **Drift is raised by the system and cleared by the system.** People acknowledge it; only the condition clearing closes it.
6. **AI proposes, rules compute, people decide.** As everywhere in the family.
7. **One board for all of it.** Planned: Pascal.

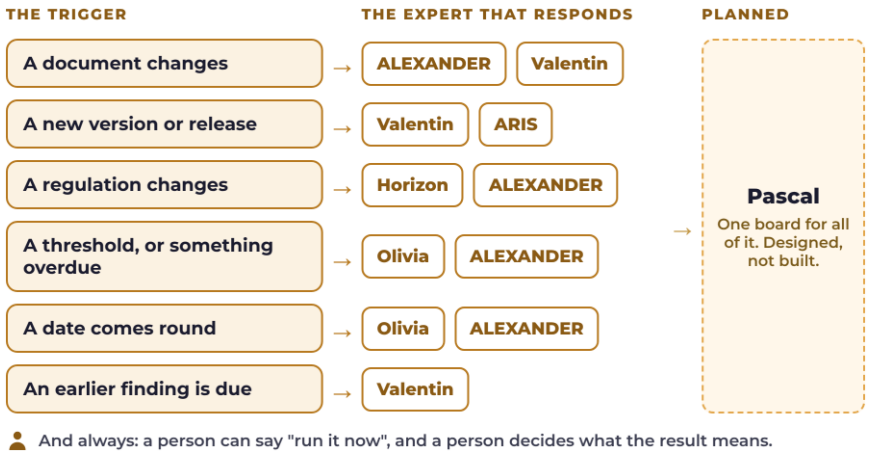
Two of these deserve a word.

Principle four is the one that surprises people, and I think it is one of the most important lines in this paper. In a continuous model, a great deal of information flows in from the business: *done*, *fixed*, *in place*. If every *"done"* closes a finding, the register fills with comfortable claims, and the continuous view becomes a continuous illusion. So *"reported done"* is its own state — between *in progress* and *remediated* — and it stays there until someone has tested it. That is not distrust of the business. It is the same discipline an auditor applies, made the default.

Principle five is its mirror image on the system's side. If a person can make a warning disappear by clicking *acknowledge*, then warnings get acknowledged rather than resolved. So acknowledging records that a person has seen it — and closes nothing. **The warning clears when the condition clears, and not before.**

The triggers

What should start a check? Here are the eight kinds of trigger the family recognises, with an example of each and the expert that responds.



Every change has a home — and, by design, every home will report to one board.

Trigger	Example	Where it is built
A document changes	A new version of the AML policy is uploaded	ALEXANDER starts a review cycle; in Valentin, a new version withdraws the sign-off
A new version or release	Thresholds are re-tuned; a new product launches	Valentin revalidates from the last run; ARIS runs a new-product approval
A regulation changes	EBA publishes a guideline; AMLA publishes a standard	Horizon's change register; ALEXANDER suggests impact for a person to confirm; ANTON's regulatory radar
A threshold or condition	A risk indicator crosses a limit; an action is overdue	ALEXANDER; Olivia's drift rules; ARIS indicators and events
A time interval	Quarterly, yearly, on a review date	ALEXANDER's schedule; Olivia's *årshjul*; ANTON's scheduled scans
An earlier finding	A follow-up is due	Valentin's follow-up rounds; ALEXANDER
A person	*"Run it now"*	All of them
A message from another expert	A hand-off file arrives	ALEXANDER; the family's hand-off files

Notice the fifth row. The calendar has not gone; it is one trigger among eight. **The annual review becomes the floor, not the whole house.**

What we have built

ALEXANDER: the continuous expert

ALEXANDER (live on its showcase) is built for a standing mandate — a client, a framework or a function that has to be kept under review over months and years, not assessed once.

Its work runs in a simple chain: an engagement, a trigger, a review cycle, a run, a register, a deliverable. Any of the triggers above can start a cycle — a schedule, a new document, a threshold, a regulation change, a message from another system, an earlier finding, or a person.

What makes it trustworthy is what each run is given, and what it records. Every run receives the mandate, the earlier findings, what has changed — and **why it is running now**. It records which instructions and sources it used. The register then follows each finding across cycles: *new*, *changed* or *closed*. Where the evidence does not allow a conclusion, ALEXANDER turns *"we could not tell"* into questions for the client, rather than a guess.

And it does not act on its own. Any action that would leave the system waits at a **checkpoint** for a person. ALEXANDER starts at its lowest level of independence — observe only — and only a person can raise it.

Picture: ALEXANDER's register on the showcase: findings marked new, changed or closed since the last cycle, with "why now" (fictional data)

Olivia: the outsourced team's continuous view

Olivia (in its main code; on a showcase with fictional clients) is built for outsourced and co-sourced compliance teams running many small clients at once. That is exactly where an annual clock hurts most: twelve clients, twelve calendars, and one desk trying to keep them all in view.

- **The annual cycle.** Baseline templates for different kinds of Swedish client turn into eighteen months of dated items, each with an owner, a reviewer and the evidence it needs.
- **Drift.** Every morning, a check runs across every client workspace against a set of drift rules — for example, proof that is overdue, or a duty "reported done" that nobody has checked. Each condition raises a row. The row clears only when the condition clears. People acknowledge; acknowledging closes nothing.
- **The inbox.** One list across all clients: what needs you, what has changed, what is late.
- **Re-assessment.** Yearly — or *triggered*, for chosen practices, with a reason recorded. A re-assessment always follows a frozen earlier one and never changes it. Facts that could be carried forward are *offered*, never recorded automatically. A delta view shows progress, and a what-if runs read-only and is labelled as a scenario.
- **Follow-up.** *"Reported done"* sits between *in progress* and *remediated*. Remediating and verifying keep four eyes.

Valentin: follow-up and revalidation

Valentin (live on its showcase) is our expert for independent review and model validation, and two of its features are continuous by nature.

Follow-up rounds. After an audit or a validation, the client's updates are recorded as statements — *implementation claimed* — never as closures. Closing a finding needs a re-test: re-performance, inspection or a new sample, reviewed by someone who did not perform it. Where a risk is accepted instead of fixed, the acceptance carries a role, a reason and an expiry date, so it comes back for a decision rather than lasting for ever.

Revalidation from the last run. A new validation does not start from a blank page. The last signed validation is imported, checked against its fingerprint, and its models and findings are matched to the new period. A lead or reviewer who did not do the import confirms the mapping. The validator starts from what was established last time, and spends her effort on what has changed.

And, as paper two described, **a new version of a document withdraws the sign-off automatically**. In a continuous world, that small rule matters a great deal: a signature cannot quietly outlive the text it was given for.

Horizon: change, fed in

Horizon (live on its showcase) is where regulatory change enters the family. It watches official sources — among them the EU Official Journal, AMLA, the EBA, Finansinspektionen, the FCA, OFSI, FinCEN and OFAC — and records new developments in the law's own words. It tracks deadlines and mandates, with estimates clearly marked as estimates. Each entry in its change register is approved with four eyes. Weekly, quarterly and board packs are frozen once issued. And its public change items can be exported as a file other experts can take in.

Scene 3's guideline, in other words, does not arrive as an email to three people. It arrives as a register entry with an owner and a date, and as a file the next expert can act on.

ARIS: the BWRA, kept alive

ARIS (live on its showcase), our expert for banks' business-wide risk assessment, answers Scene 1 directly. A new product goes through an approval pipeline with a small BWRA of its own and a

what-if before launch. Between annual cycles, issues, risk indicators, events and control tests keep the assessment alive. And residual risk moves only when a person changes a score or a rating — the machine can show that something has changed, but it cannot move the number.

ANTON: a living register and checks after each answer

ANTON brings the same idea to its generalist work (in ANTON's main code):

- **The Risk Atlas** is a living risk register for any business. Fixed integrity rules surface live findings on its dashboard — for example, a high residual risk with no appetite statement, a control marked "strong" with no evidence behind it, or a path outside appetite with no action or target date. Review cycles and escalation triggers are part of its design.
- **The regulatory radar** scans sources on a schedule an administrator sets, and can be switched off.
- **A set of compliance checks runs after every answer**, so problems are flagged as the work is done rather than at the end.

Pascal: one board for all of it — designed, not built

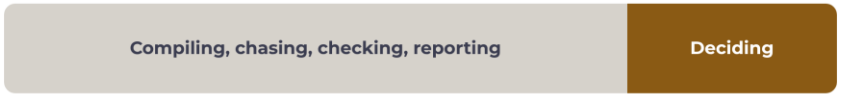
Designed, not built: Pascal

Pascal is designed to be the board where all of this comes together, for people and AI experts alike:

- **Recurring work** — the *årshjul* — as recurring cards.
- **Work items pushed in** from Horizon, Olivia, Valentin, Gabriel and ANTON, with status sent back.
- **Who did each task:** a person, an AI expert, or both — and how long it took against the estimate, so estimates get better over time.
- **People's personal ratings stay private;** figures are shown only for groups large enough to protect individuals. Consultation with employees and a data protection impact assessment are part of its delivery plan.

The design exists. **Pascal is not built yet.**

Before: the annual clock



Much of the time goes into producing the check.

After: change as the trigger



Same people. A different shape of week.

Not fewer people. Less compiling, more deciding — an illustration of the shift, not a measurement.

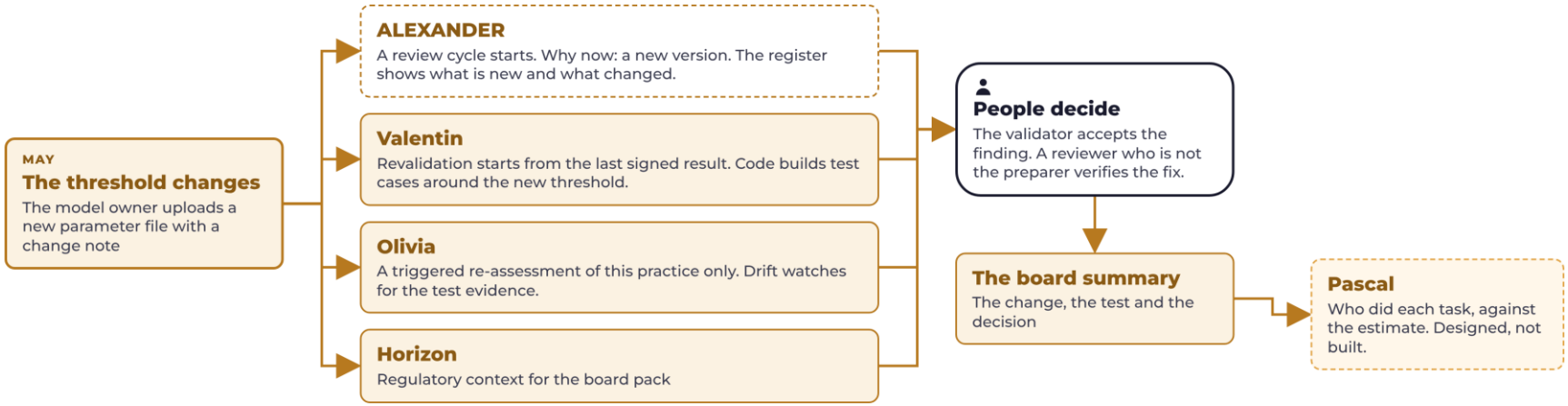
Status at a glance

- **Live** (public showcases, fictional data): ALEXANDER, Valentin, Horizon, ARIS.
- **In main code** (merged, not checked as deployed): Olivia's drift, re-assessment and follow-up; ANTON's Risk Atlas rules, regulatory radar and post-answer checks.
- **Designed or planned:** Pascal — designed, not built.

Part 4 · A worked example

One change, five reactions

Exempelbolaget AB is fictional, and so are its outsourced compliance desk and its validator. In May, the firm's transaction-monitoring threshold changes. Under the annual clock, that change would wait for next year's validation — Scene 2. Here is how the family is **designed** to respond when the change itself is the trigger. Each step is marked with who acts, and its status today.



The change was caught in May, not next year. People spent their time deciding, not compiling.

Step	What happens	Who acts	Status
1. The change is recorded	The firm's model owner uploads the new parameter file, version 3, with a short change note.	A person	—
2. ALEXANDER sees a new document	On the standing mandate, a review cycle starts. *Why now:* a new version of the parameter file. The register shows one new finding (the change was not tested below the line) and one changed finding.	ALEXANDER proposes; code records	Live (showcase)
3. Valentin scopes a revalidation	The validator imports last year's signed validation as the baseline, checked by its fingerprint. The threshold test is flagged for re-performance; code generates test cases just below, at and just above the new threshold for the firm to run.	Code; the validator	Live (showcase)
4. Olivia opens a triggered re-assessment	The desk re-assesses the transaction-monitoring practice only, following the frozen yearly assessment. If the firm's test evidence does not arrive by its date, drift raises *proof overdue*.	The desk; code (drift)	Merged
5. Horizon adds context	In this example, Horizon's register shows an open EBA consultation touching monitoring thresholds. The team notes it for the board pack.	Horizon; a person	Live (showcase)

Step	What happens	Who acts	Status
6. People decide	The validator accepts the finding. Once the re-test passes, the desk's reviewer — not the preparer — verifies the remediation. The board summary shows the change, the test and the decision.	People (four eyes)	Live / merged
7. On one board	Pascal shows the five tasks, who did each — a person, an AI expert or both — and how long each took against the estimate.	Pascal	Designed, not built

What changed, and who did what

Go back to Scene 2. There, the same threshold change waited fourteen months for someone to look at it. Here, it was looked at in May — not because anyone remembered, but because the change itself started the work.

Notice how little of that work was compiling. Nobody spent a week reconstructing last year's validation; it was imported and matched. Nobody re-read the whole framework; ALEXANDER's register showed one new finding and one changed one. Nobody chased the test evidence by memory; drift would raise it on the day it became late, and keep raising it until it arrived. **The machine did the preparing. People did the deciding** — the validator on whether the finding stands, the reviewer on whether the fix is real, the board on what it means.

Notice, too, what did *not* happen. The firm's *"we have tested it"* did not close anything; it was a claim until the re-test passed and a second person verified it. ALEXANDER did not send anything to anyone on its own; it prepared, and waited at its checkpoint. No AI set a severity or a rating. And the annual validation did not disappear — it is still the floor. It will simply be shorter next year, because one of its biggest questions was answered in May.

And notice the last row. It is marked *designed, not built*, and I left it in because it shows where this goes: one board where a manager can see, honestly, which work was done by people, which by AI experts and which by both, and whether the estimates were right. That is the beginning of an honest answer to the question every manager asks about AI — *what did it actually change?* — and it is a question I would rather answer with a board than with a slide.

Part 5 · From here

What this means for you on Monday

You can start changing the rhythm without any new software. Five things to do this week:

1. **List your annual checks** — and next to each, the events that should really trigger it: a new product, a changed parameter, a new regulation, a departed control owner.
2. **Pick one and make it event-driven for a quarter.** Run a short first pass whenever its trigger happens, and keep the annual check as the floor.
3. **Report deltas.** Change one monthly or quarterly report to *"what changed since last time, and why"*. Watch how much shorter — and more useful — it becomes.
4. **Separate "claimed" from "verified".** Do not count a client's or a business unit's *"done"* until someone has tested it. Give "reported done" its own column.
5. **Talk about roles.** Ask your team which part of their week is compiling and which is deciding. Then agree, explicitly, what the freed time should go to.

Next week: the .anton format

If expertise lives in modules, skills, knowledge packs and rules, how do you share it, check it and keep it? Next week we look at one openly documented format for packaging expertise: what is inside an *.anton* file, how it is checked before anything is installed, and the shared universe of expertise it could grow into — with what is built and what is planned clearly marked.

A closing thought

For most of my career, the annual plan has been the backbone of a compliance function. It is a good backbone. It makes sure nothing is forgotten for ever. But it was always a compromise with cost — the best we could do when every look at the risk took weeks of skilled time.

That compromise is loosening. Not because AI is clever enough to watch everything on its own — it is not, and in our family it is never allowed to — but because a well set-up AI can prepare the first pass of a check quickly enough that we can afford to look when something changes, rather than when the calendar says so.

What we do with that is a choice. We can run the same annual cycle with less overtime. Or we can change the rhythm: let change be the trigger, read the deltas, keep the annual review as the floor — and give people back the time to do what the annual scramble always crowded out. **I think the second choice is the real promise of AI in compliance. Not fewer people. Earlier warnings, better decisions, and time to act on them.**

— Daniel Bardun

See it, try it, talk to us

- **See it:** the ANTON family's public showcases, with fictional data, at ai-portal.futurechain.eu.
- **Try it:** ANTON is open source — github.com/altspace-hub/anton_desktop.
- **Read the series:** all papers in the ANTON family whitepaper series, and the earlier whitepapers, are linked from the portal. [Daniel: confirm where the PDFs will live.]
- **Talk to us:** Daniel Bardun, on LinkedIn — questions, disagreements and "this is where AI stopped working for us" are all welcome.

Important notes

Product facts were checked on 11 October 2026; statuses are summarised in "Status at a glance". The public showcases use fictional or public data only, and no client uses them. ANTON is open source (EUPL); the other family products are not. The products support your work and help you evidence it; they do not make anyone compliant, and nothing in this paper is legal advice.