

THE COMPLETE GUIDE FOR 2026

Segregation of Duties

A practitioner's guide for Internal Auditors, SAP Security, IT Basis teams, and Business Process Owners navigating the modern access risk landscape.

Filip Nowak | Partner @ GRC Advisory | smartgrc.eu

83%

of organizations face SoD violations in production systems

6×

more costly to remediate SoD issues post-audit vs. preventively

60%

of SAP roles contain at least one unresolved SoD conflict

2026

SAP GRC 2026 replaces GRC Access Control 12.0

Why Segregation of Duties has never mattered more	4
1. SoD Fundamentals	5
1.1 The Core Principle	5
1.2 Why SoD Programs Fail in Practice	7
1.3 The Business Cost of SoD Failures.....	13
2. Building and maintaining the SoD Matrix	14
2.1 What an SoD matrix actually is?.....	14
2.2 Defining business-level risks	14
2.3 Technical mapping: from risk to system objects.....	17
2.4 The Matrix must be a living document.....	20
3. SoD in SAP S/4HANA	21
3.1 Why the ECC SoD Matrix Cannot Be Copied to S/4HANA.....	21
3.2 The Technical Dimension: Fiori, OData, and the New Access Layers	22
3.3 The Business Dimension: New Processes, New Risks	23
3.4 S/4HANA SoD Matrix: Technical Reference	24
3.5 How to Approach the Matrix Migration in Practice	25
3.6 New Risk Areas Specific to S/4HANA Beyond Procurement.....	27
3.7 Validating the Updated Matrix.....	28
3.8 The 2026 Agenda: What makes S/4HANA SoD different right now	29
4. Conducting the SoD Audit	30
4.1 SoD Matrix Foundation: Starting from a Validated Baseline	30
4.2 Stage 2: Technical Mapping	31
4.3 Stage 3: Opening Balance Analysis	32
4.4 Stage 4: Remediation	33
4.5 Stage 5: Continuous Monitoring	34
5. Regulatory frameworks & SoD obligations.....	36
5.1 Sarbanes-Oxley (SOX): Section 404	36
5.2 GDPR - Access Control as a Data Protection Obligation.....	37
5.3 ISO/IEC 27001 - Access Control as a Core Control Domain.....	37
5.4 Aligning Across Frameworks	38
6. AI & Automation in SoD Monitoring.....	39
6.1 The State of AI in GRC today	39
6.2 AI-Assisted SoD matrix building	39
6.3 Usage analytics and active risk detection.....	40
6.4 AI-Assisted Access Provisioning with SoD Simulation.....	40
6.5 Firefighter Log Review and AI-Assisted Analysis.....	41
6.6 Risks and Limitations of AI in GRC.....	41
7. Cloud, Hybrid, and Non-SAP Landscapes	43
7.1 The Hybrid Reality.....	43

7.2 SAP Cloud Solutions: SuccessFactors, Ariba, Concur	43
7.3 Non-SAP Systems: Oracle, Workday, and Custom Applications	44
7.4 Cloud Access Models and Their SoD Implications	44
8.GRC Tooling Landscape 2026	45
8.1 SAP GRC 2026	45
8.2 SAP IAG (Identity Access Governance).....	47
8.3 smartGRC, Agility and cross-system coverage.....	48
8.4 One-Time Audit vs. Continuous GRC: The Decision Framework.....	49
9.SoD Maturity Model	51
9.1 The Five Levels of SoD Maturity.....	51
9.2 Common Traps at Each Level	52
9.3 Practical Roadmap to Move Up Levels	53
Quick-Reference: GRC Hacks Summary	54

Introduction

Why Segregation of Duties has never mattered more

From ERP basics to AI-driven continuous control. The definitive 2026 practitioner's reference guide.

Segregation of Duties, or SoD, is one of the oldest principles in internal control. Yet in 2026, it remains one of the most frequently cited audit findings across industries and geographies. The reason is not that organizations don't understand the concept. They do. The problem is execution, and execution has become significantly harder in recent years.

Take SAP S/4HANA as a concrete example. In the ECC era, managing SoD meant controlling transaction codes (T-codes) and authorization objects, a complex task, but a bounded one. S/4HANA with SAP Fiori changes the equation entirely. Fiori is not simply an interface update. It represents SAP's strategic, cutting-edge UX direction, setting the standard for modern enterprise software interaction and positioning SAP among the leaders in ERP user experience globally. Understanding Fiori is therefore not optional for SAP security practitioners and auditors. It is central to the job in 2026 and in the years ahead, if you want to manage user authorizations and SoD access risks in an efficient and reliable way.

What started as 25 apps in 2013 has grown to over 7,500 Fiori applications today, with more than 1,700 relevant to core functional areas like finance, procurement, and supply chain. Critically, the actual number of apps available in any given system depends on the Product Suite and Release Version. This means two organizations both running S/4HANA may be working with meaningfully different application landscapes. And this is not just a counting problem. Across versions, the same Fiori application can reference a different OData service, a different Semantic Object, or a different Action. The technical SoD access risk mapping that was accurate after last year's implementation may no longer reflect how the system actually behaves today.

The challenge goes beyond numbers, though. Fiori fundamentally changes how end-to-end processes are executed. A single app can now bundle what were previously separate T-code steps, for example creating a purchase order, approving it, and posting the goods receipt, into one streamlined user interface. For the business user this means efficiency and a better experience. For the SAP security team or auditor, it means that a role assignment which looks clean in the role catalog may quietly enable a significant SoD access risk in practice. The convenience Fiori was designed to deliver and the control SoD was designed to enforce are increasingly in tension, and that tension is not going away. The question of flexibility versus security will remain a defining challenge for SAP security practitioners in 2026 and well beyond.

This guide brings together practical knowledge from hundreds of SAP authorization reviews, GRC implementations, and external audits conducted across finance, manufacturing, and services sectors. It is written for SAP security professionals, internal auditors, GRC consultants, and business process owners who are often working on the same problem from different angles. Internal auditors need to assess and report on SoD access risks. SAP and IT teams design and maintain the authorization architecture. Business process owners bear ultimate accountability for what happens in their process area. This guide is built to be useful for all of them.

The guide covers the fundamentals of SoD, what it is, why it matters, and how it fails in practice. It then goes deep on SAP-specific topics including the critical differences between ECC and S/4HANA, the impact of Fiori and OData on risk definition, and the evolving GRC tooling landscape. It closes with the topics that will define the next three years: AI-assisted SoD monitoring, regulatory alignment across SOX, GDPR, and ISO 27001, and the unique challenges of cloud and hybrid system landscapes.

Throughout the guide you will find GRC Hacks, concise experience-backed recommendations that cut through theory and tell you what actually makes a difference in practice.

1

1. SoD Fundamentals

What it is, why it fails, and what it costs when it does

1.1 The Core Principle

Segregation of Duties is the practice of dividing critical business process steps between at least two different individuals so that no single person can both initiate and complete a transaction that could result in fraud or material error.

A good illustration is the purchasing process. The person who raises a purchase requisition should not be the same person who approves it, receives the goods, posts the vendor invoice, and executes the payment run. Each of those steps is meant to be a checkpoint. But there is another dimension here that goes beyond internal process design. Purchasing is one of the few business processes where an external party, the vendor, has a direct financial interest in how your company behaves internally. A vendor benefits when orders are placed unnecessarily, when quantities are inflated, or when prices are set above market rates. Your company pays for all of it using its own processes and its own money.

When a single employee controls the full purchasing cycle and maintains a close relationship with a vendor, the conditions for misconduct are in place. The vendor has every incentive to cultivate that relationship, offer personal benefits, and apply pressure to keep orders coming. The employee, executing transactions using company systems and company funds, becomes the point of failure. What looks like an access configuration issue in the SAP role concept is, in practice, a bribery risk with a motivated external actor on the other side of it.

The data from the ACFE Occupational Fraud 2024: A Report to the Nations, based on 1,921 real fraud cases across 138 countries, confirms just how exposed the purchasing function is. According to the report, corruption was by far the dominant fraud type in purchasing departments, appearing in 79% of cases, the highest rate of any department studied. The median loss per purchasing fraud case was USD 143,000. And when looking at the behavioral profile of fraudsters, the report found that 20% of all perpetrators maintained an unusually close association with a vendor or customer, with that figure rising to 30% specifically in corruption cases. In other words, the scenario described above is not a theoretical risk. It is the most common way purchasing fraud actually happens. This is why SoD in purchasing is not just about clean authorization design. It is about removing the opportunity before the incentive has a chance to act on it.

In top cutting edge ERP systems like SAP S/4 Hana, SoD is enforced through the authorization model and controlling which transactions, applications, and master data objects a user can access. A well-designed SoD framework translates business process risks into technical access restrictions, so that a conflicting combination of authorizations simply cannot coexist in a single authorization role and in particular in a user account.

The practical challenge is that organizations rarely start from scratch. Most SAP environments were configured years ago, roles were designed for functionality rather than control, and the access landscape has grown organically ever since. Role and authorization redesign projects lessons learnt experience tells a consistent story: on average, only 35 to 40 percent of the authorizations assigned to a user are actually used in day-to-day work. The rest sit idle, invisible in the role catalog, but very much active in the system. That unused access is not harmless. It represents a large pool of unnecessary permissions that quietly expand the SoD access risk surface without anyone noticing, and without any business justification to defend them. And it tends to get worse over time. When a new employee joins, the easiest path is to copy the role assignments from an existing user in a similar position. That copy carries not just the access the new user needs, but every redundant permission, every historical workaround, and every unresolved SoD access risk that accumulated in the original account over years. Repeat that across hundreds or thousands of onboarding cycles and the problem does not just persist, it compounds. What started as a manageable gap between assigned and used access gradually becomes a system where over-authorization is the norm and cleaning it up requires a project of its own.

The result is a system where theoretical SoD exists in policy documents, but real-world access tells a very different story. And this challenge is accelerating in 2026. Every Fiori app activation, every new release, and every organizational change adds another layer to an already complex picture. Fiori does not just expand the number of access points that need to be evaluated. It changes the nature of the question entirely. In the T-code world, you could ask "what can this user execute?" In the Fiori world, you need to ask "what end-to-end process steps can this user complete through a single app, and does that combination create an SoD access risk that would never have appeared in a traditional role review?" That is a fundamentally harder question to answer, and most organizations are not yet set up to ask it systematically. In 2026, getting that answer right has become one of the most pressing topics in SAP security and access governance. This guide is designed to help structure exactly that, providing a practical framework for practitioners who need to move beyond traditional role reviews and tackle SoD access risk in the way modern SAP landscapes actually demand.

1.2 Why SoD Programs Fail in Practice

The 9 most common SoD failure patterns in SAP environments

1. **Access shaped by urgency, not by design.** Users get 'enough access to work' under go-live pressure. Shortcuts become the unchallenged baseline for future system usage and users that requires access.
2. **Copy-paste or 'make like' onboarding.** New users are cloned from existing accounts, inheriting every SoD risk and over-authorization. The cycle repeats with every hire.
3. **The SoD risk framework nobody owns.** The matrix is copied from a template or a previous project and treated as good enough. Nobody built it from real business input, so nobody owns it afterwards. New transactions, Fiori apps, and custom developments go live without ever making it into the risk framework. Three years on, the analysis still runs but the rule set underneath it reflects a system that no longer exists.
4. **Mitigation overload.** Risks are mitigated instead of fixed. The compensating controls list grows until it is unmanageable and controls are signed off unexecuted.
5. **Audit-only reviews.** SoD reviews happen once a year, driven by external deadlines. Access granted on day one may not be reviewed for twelve months as since authorizations are working we should not challenge this.
6. **Fix it during the project or carry it forever.** Role design done properly is a one-time investment. Done poorly, it produces something that works on day one and becomes harder to touch with every month that passes. Once the business is running on a role concept, the cleanup window closes and the debt compounds quietly until nobody remembers what the original design was supposed to look like
7. **SoD as a documentary for auditors, not a design principle.** There is no owner, no review calendar, and no process between audit cycles. When auditors arrive the organization performs, data is pulled, evidence is packaged, and findings are responded to with urgency. When they leave, everything goes quiet.
8. **No SoD program at all.** Individual components exist but nothing connects them. The pieces are there but they just do not add up to a program.
9. **Excel instead of a dedicated GRC tool.** Analysis runs in spreadsheets outside the system landscape. Data is stale, errors are unavoidable, and real-time control is impossible.
10. **Nobody wants to be the person who says no.** Access gets approved because pushing back creates friction. Reviews get signed off without changes because challenging a colleague feels like an accusation. Known conflicts stay open because remediating them means a difficult conversation with someone more senior. The path of least resistance is always to approve, accept, and defer.

1. Access shaped by urgency, not by design. Under go-live pressure, the priority shifts from doing things right to getting things live. Users receive access based on what they need to unblock themselves on day one, not on what their job role actually requires. A finance clerk gets posting and approval rights because they need to test. A warehouse supervisor gets purchasing access because it was easier than waiting for IT. Nobody documents the deviation and nobody revisits it.

But there is a deeper problem underneath the urgency. Even when there is time and good intention, nobody can clearly answer what access a given role actually requires. HR job descriptions operate at a level of abstraction that does not translate into authorization objects and transaction codes. The supervisor knows what the team does but not which specific system actions they need to do it. Key users understand the process but are fully occupied with testing and training. External consultants know the system but not the business, and their time is expensive enough that detailed access scoping rarely makes it onto the project budget. So the question of what access is truly required goes unanswered, and the project fills that gap with approximations and best guess estimations. Someone who looks similar gets used as a reference. A standard role that covers most of the need gets assigned. A key user makes a judgment call under time pressure.

Within weeks these guesses become the new baseline. When a new person joins the same team, they are provisioned to match their predecessor, inheriting access that was never correct to begin with, just

never questioned either. The go-live pressure is temporary. The access profile it creates, built on assumptions nobody documented and decisions nobody remembers making, is not.

2. Copy-paste or make-like onboarding. When a new user joins, the path of least resistance is to find someone in the same team and copy their roles. It takes minutes, the new user is productive immediately, and the provisioning ticket is closed. What also gets copied is every SoD conflict, every excess authorization, and every role that was added as a temporary workaround some years ago.

The source account is rarely reviewed before it is cloned. But even if someone wanted to review it, they would quickly run into the same problem described in point 1. Nobody knows with confidence what that reference user should have access to in the first place. Their profile was shaped by a go-live approximation, then perhaps adjusted when something did not work, then copied to the next person who joined the team. That person may have had a slightly different role, so a few things were added. Then their account became the new reference. Each copy inherits not just the access but also the accumulated uncertainty behind it. Nobody along the way had a clear answer to the question of what was correct, so nobody could identify what was wrong.

Over time the problem compounds quietly. The access profile of the department drifts further from any intended design with every hire, but because each individual step looks reasonable, nobody raises an alarm. The new starter needs access today. Someone in the team has a similar job. Copy the roles, close the ticket, move on. No single decision looks irresponsible. The damage is not in any one moment but in the pattern that repeats, unquestioned, across years and across teams in large SAP landscapes.

By the time an audit or a security review surfaces the issue, tracing back what went wrong is nearly impossible. The original reference accounts may no longer exist. The people who made the provisioning decisions have moved on. The business cannot say what was intended and cannot explain what exists. The access landscape has become a layering of old decisions, workarounds, and guesses, written over each other until the original design is no longer legible underneath.

3. The SoD risk framework nobody owns. Most organizations treat the SoD risk matrix as a technical deliverable, something the security team or external consultants produce during the implementation project and hand over at go-live. In reality, building a meaningful SoD matrix is one of the most valuable conversations a business can have about its own processes, and most organizations never have it properly.

The matrix is not just a list of conflicting transaction codes. It is the answer to a set of business questions that someone needs to sit down and think through carefully. What would happen if the same person could create a vendor or update bank account details and process a payment? What is the real risk if a warehouse clerk can also post a goods receipt and release a purchase order? What does it actually mean, in business terms, if a single user controls the full procure-to-pay cycle without any checkpoint? These are not IT questions. They are process and control questions, and answering them requires business owners who understand what their people do and what could go wrong if the wrong person had too much access.

When that conversation happens well, it does something beyond producing a document. It changes how business owners think about access. They stop seeing system authorizations as a technical matter for IT to sort out and start understanding why copying access from one user to another is not a neutral act. They begin to see that more access is not more capability, it is more risk. The SoD risk matrix becomes a shared frame of reference for asking the right question every time access is requested: if this person can do this, what else does that make possible, and is that acceptable?

In practice that conversation rarely happens at the depth it should. What most projects do instead is take a baseline rule set from a previous implementation, a standard SAP template, or sometimes a rule set carried over from a completely different client, and treat it as good enough. The same copy-paste logic that drives user provisioning also drives the risk framework itself. The matrix arrives pre-populated, is reviewed briefly, and is signed off under time pressure. Nobody has genuinely asked whether the rules reflect the actual risk landscape of this business, this system configuration, and these processes. And because nobody built it from scratch with real business input, nobody feels responsible for it afterwards. It belongs to the project that created it, and that project is over.

That absence of ownership is what makes the maintenance problem inevitable. The system evolves continuously after go-live. New Fiori applications are deployed, Z-transactions are created for local requirements, OData services expose new functionality, and standard roles are modified to support process changes. For any of these changes to make it into the risk matrix, someone needs to notice the change, understand its implications, and take the initiative to update a document they do not formally own and were not involved in creating. That rarely happens. Three years after go-live, the matrix still references transaction codes that map to screens the business no longer uses, while the custom transactions that now carry the most significant risk have never been assessed at all. SoD analysis continues to run, reports are produced, and results are reviewed, but the rule set underneath it all reflects a system that no longer exists. The control looks active. It is measuring the wrong things, owned by nobody, and questioned by no one.

4. Mitigation overload. When an SoD conflict cannot be resolved quickly, because fixing it requires role redesign, process change, or a conversation nobody wants to have, the default response is to add a compensating control. The conflict is logged, a mitigation is documented, and the risk is classified as accepted. This is a legitimate approach for a small number of genuine edge cases. The problem arises when mitigation becomes the standard response to every conflict.

The compensating controls register grows to hundreds of entries. Each control requires a responsible owner and a periodic effectiveness review, but as the list grows the reviews become a checkbox exercise. Controls are signed off because they are due, not because they were executed. The underlying access risks remain in place indefinitely, covered by a layer of documentation that gives the appearance of control without delivering it.

5. Audit-only authorization reviews. The access review cycle is set by the external audit calendar, not by the risk profile of the system. Once a year, typically in the weeks before the financial audit, the team pulls a list of user roles, sends confirmation requests to managers, collects responses, and files the output. The review meets the audit requirement. What it does not do is catch access risks in the eleven months between cycles. A user who changes roles in February keeps their previous access until the following January review. A terminated contractor whose account was not disabled on time passes through undetected until auditors ask.

The audit process itself adds another layer of complexity. Auditors often arrive with their own SoD rule set, one that was not built for this business, this system configuration, or this version of SAP. The scan is executed under the pressure of an audit timeline, which leaves little room to question whether the rules being applied are the right ones. Results are escalated quickly, often reaching board or senior management level before anyone has had the chance to understand what they actually mean. Remediation pressure follows immediately. Teams are asked to fix problems before they have established whether those problems are real, or whether the findings reflect genuine risk in this system or are artefacts of a generic rule set applied without context.

And yet, for all its limitations, the audit does something important. It puts access and SoD risks on the agenda. It creates a moment where senior management pays attention, where the topic moves from an IT maintenance issue to a business control concern. Even when the findings are imprecise and the remediation is rushed, the audit brings a level of visibility and accountability that is often absent the rest of the year. The problem is not that audits happen. The problem is that for most organizations, the audit is the only thing that makes anything happen at all.

6. Fix it during the project or carry it forever. There is a fundamental difference between fixing a user's access profile and fixing the role concept the entire organization runs on. Getting individual access wrong is a problem. Getting the role design wrong is a structural problem that touches every user, every process, and every future change request simultaneously.

As described in points 1 and 2, the foundation is often already compromised before go-live. But even setting aside the quality of individual access decisions, the role architecture itself carries its own separate debt. Role design done properly during the project is a one-time investment. It requires time, business involvement, and the willingness to have difficult conversations about what each function genuinely needs. Done well, it produces a clean, maintainable architecture where SoD conflicts are designed out rather than managed around. Done poorly, it produces something that works on day one and becomes harder to touch with every month that passes.

Once the business is running on a role concept, every cleanup attempt creates immediate operational risk. Removing an authorization object breaks a transaction someone depends on. Splitting a role means re-testing user access across multiple departments. Changing a role assignment triggers a change request requiring sign-off from several stakeholders. The people who would need to approve and support the cleanup are the same people who are busy running the business on the system as it exists. The effort required grows with every month, and the appetite to take it on shrinks at the same rate. Most organizations never complete the cleanup. They carry the debt of a role design built for speed indefinitely, patching around it rather than fixing it, because the window to do it properly closed on the day the system went live.

7. SoD as a documentary for auditors, not a design principle. In organizations without a dedicated GRC function, SoD has no owner between audit cycles. There is no team responsible for maintaining the risk matrix, no calendar for access reviews, no process for evaluating new developments against existing controls, and no escalation path when conflicts are identified. The program exists on paper. In practice it runs only when someone external arrives to check on it.

When auditors come, the organization performs. Data is pulled, evidence is packaged, mitigations are documented, and findings are responded to with urgency. It looks like a functioning control environment because for those few weeks it briefly becomes one. When the auditors leave, the activity stops. The files are saved, the working group disbands, and the topic goes quiet until the next audit cycle begins.

The deeper problem is what this pattern does to how the organization understands its own risk. When SoD is driven by audit deadlines rather than business ownership, the questions being asked are always the auditor's questions, not the business's questions. The findings reflect what the auditor's rule set flagged, not necessarily what represents genuine risk in this system and this organization. Remediation is aimed at closing findings rather than fixing root causes. The real questions, why did this access exist, how did it get there, and what does it tell us about how we manage authorizations, rarely get asked in the middle of an audit response. The business learns to satisfy the audit rather than to control the risk, and over time those two things drift further and further apart.

There is also something lost in how SoD gets communicated. Audit findings travel up to senior management as compliance issues, presented in audit language, measured against audit criteria, and resolved through audit processes. The business conversation about what the risks actually mean, what could realistically go wrong, and what a genuinely controlled access environment would look like, never really happens. SoD becomes a periodic documentary about the organization's control environment, produced for an external audience, filed away, and forgotten until the next screening.

8. No SoD program at all. Most organizations have the ingredients. There is a risk matrix, even if outdated. There are mitigation records, even if nobody reviews them. There are access reviews, even if they happen once a year under audit pressure. There may even be a GRC tool with configured rules sitting in the landscape. From the outside, and sometimes from the inside, it looks like a program exists.

What is missing is the architecture that connects these pieces into something that actually functions. A defined risk appetite that tells the organization what level of access risk is acceptable and what is not. Clear ownership that makes specific people accountable for specific outcomes. A governance calendar that makes activity predictable rather than reactive. A feedback loop that catches when the program is drifting out of alignment with the system it is supposed to govern.

Without these, each component lives in its own silo. The risk matrix is not linked to the review cycle. The review cycle is not connected to the remediation process. The mitigation records are not monitored for effectiveness. Nobody is asking whether the program as a whole is working, because nobody owns the program as a whole. The organization has assembled the parts without ever building the thing. And because each individual part exists, there is rarely a moment of clear failure that forces the question. It just quietly does not work, and nobody is quite sure whose problem that is.

9. Excel instead of a dedicated GRC tool. SoD analysis managed in spreadsheets creates a structural gap between the appearance of control and the reality of it. Role exports are pulled manually from the system, pasted into a workbook, and compared against a risk rule set maintained in a separate tab. By the time the analysis is complete, the underlying access data has already changed. There is no audit trail of who ran the analysis, what version of the rules was applied, or what changed between one review and the next. Errors introduced during manual data handling are invisible. Simulating the impact of a proposed role change, a standard capability in any dedicated GRC platform, is simply not possible. When the person who built and maintains the spreadsheet leaves, the methodology leaves with them. The organization believes it is monitoring SoD because a review is produced on schedule. In practice it is producing a historical snapshot of access at a point in time, not a continuous or reliable control.

10. Nobody wants to be the person who says no. SoD programs fail for technical reasons, process reasons, and governance reasons. But underneath many of those failures is a simpler and more human problem. Taking access away from people is uncomfortable. Challenging a manager's provisioning request creates friction. Telling a colleague that the access they have relied on for three years needs to be removed feels like an accusation. And in organizations where the security or GRC team is small, outnumbered, and dependent on the goodwill of the business to get anything done, that friction has a cost that people learn to avoid. So access gets approved because the request came from a senior stakeholder and nobody wanted to push back. A review gets completed with everything marked as appropriate because the manager signing off did not want to create problems for their team. A known SoD conflict stays open on the mitigations register because remediating it would mean a difficult conversation with a process owner who has more organizational weight than the person raising the concern. The path of least resistance is always to say yes, to approve, to accept, to defer.

This dynamic is rarely visible in audit findings or risk registers. It does not show up as a control failure in any formal sense. But it shapes every decision in the access management process. The role that never gets cleaned up, the review that produces no changes year after year, the mitigation that gets renewed without being checked, these are not always failures of process or tooling. Sometimes they are the entirely rational behavior of people who have learned that challenging access creates enemies and approving it creates none.

Building a functioning SoD program means acknowledging this dynamic directly. It means giving the people responsible for access control the organizational backing to say no, the escalation paths to raise concerns without personal risk, and the senior sponsorship that makes access decisions a business matter rather than a personal one. Without that, even the best-designed program will be quietly softened into ineffectiveness by the entirely human instinct to keep the peace.

1.3 The Business Cost of SoD Failures

Regulators and auditors have long treated SoD violations as a significant control deficiency. Under Sarbanes-Oxley Section 404, unresolved SoD conflicts in financial systems can escalate to a material weakness if the control environment cannot compensate. Under GDPR, unauthorized access to personal data, which SoD failures frequently enable, carries fines of up to 4% of global annual turnover. Under ISO/IEC 27001, access control is a core Annex A control domain and SoD is an explicit element of the required access management process. SoD is addressed directly within this domain. The standard requires that conflicting duties and areas of responsibility be separated to reduce opportunities for unauthorized or unintentional modification or misuse of organizational assets. In practical terms this means that organizations seeking or maintaining ISO/IEC 27001 certification are expected to have identified where conflicts exist, designed controls to prevent or compensate for them, and be able to demonstrate that those controls are operating effectively.

The fraud data makes the stakes concrete. The ACFE estimates that organizations lose 5% of their annual revenue to occupational fraud, with an average loss per case of \$1.7 million. That figure is widely considered conservative, as it does not account for indirect costs such as reputational damage, lost productivity, and the long-term impact on business relationships. The most cited organizational weakness in fraud cases was lack of internal controls, identified in 32% of cases, while override of existing controls accounted for a further 19%. In other words, more than half of occupational frauds occur due to a lack of internal controls or an override of existing internal controls. SoD failures sit squarely within both categories. The median fraud scheme in the 2024 study took 12 months to uncover, with average losses running at \$9,900 per month. For organizations relying on annual manual access reviews, this means a conflict exploited the day after a review closes may go undetected for the full duration of the next cycle. The math is straightforward and uncomfortable.

Beyond the fraud risk, the operational cost of managing SoD poorly is significant in its own right. Organizations that rely on annual manual reviews typically spend three to six weeks per review cycle reconciling data, engaging process owners, and documenting findings. A recurring quarterly cycle supported by automated tooling reduces this to days. The investment in prevention is a fraction of the cost of a single fraud incident that exploited an access conflict nobody remediated, and a fraction of the regulatory exposure that accumulates in the meantime.

The pattern the ACFE data describes, controls absent or overridden, fraud running undetected for months, losses discovered only after the fact, is not an abstract risk. It is the predictable outcome of the ten failure patterns described in the previous section, playing out in organizations that had the components of a control program without ever building one that worked.

2

2. Building and maintaining the SoD Matrix

The foundation that every tool and audit depends on

2.1 What an SoD matrix actually is?

The SoD matrix is the structured document that defines which combinations of access are prohibited because together they create an unacceptable risk of fraud or error. It operates on two levels: the business level, where risks are defined in process terms ("a user who can configure or change existing purchasing approval workflows setup should not also be able to approve purchase orders"), and the technical level, where those risks are translated into specific system specific objects, examples in SAP S/4 Hana are: transaction codes, fiori applications, oData services, and authorization objects.

A simple example helps illustrate what the matrix actually captures. In the procure-to-pay process, the ability to create or modify a vendor bank account number in the system, recorded in SAP S/4HANA as a business partner, is one activity. The ability to execute a payment run is another. Each of those activities on its own is legitimate and necessary. Together, in the hands of a single person, they create a complete fraud pathway: invent a vendor, route a payment to it, and collect the money. The SoD matrix exists to say, formally and explicitly, that this combination is prohibited. Not discouraged. Not flagged for review. Prohibited.

The same logic applies across every major process area. In finance, the person who can post a journal entry should not also be able to reopen a previously closed accounting period. The combination creates a cut-off risk: costs that belong in January can be quietly moved back into December, distorting reported profit or creating a tax advantage that was never authorized. In inventory, the person who can post a goods receipt should not also be the person who can execute inventory count or adjustments to inventory count results. In HR and payroll, the person who can create or modify an employee master record (salary) should not also be able to run the payroll. In each case the individual activities are routine. The combination is the risk. Without a well-maintained SoD matrix, no GRC tool, no audit methodology, and no access review process can deliver reliable results. The matrix is the content layer. The tool is merely the engine that processes it.

GRC Hack #1

Don't design or build roles without an SoD matrix. Before you start designing authorizations, perform a business process risk analysis and use it to create your SoD matrix. Anyone doing it the other way around makes a conceptual error that will surface during the next audit. The SoD matrix is not a byproduct of role design. It is the prerequisite.

2.2 Defining business-level risks

At the business level, each SoD risk describes two or more business activities that, when performed by the same person, create an exposure. The activities should be described in plain language that business process owners can understand and validate, not in transaction code nomenclature that only IT specialists recognize.

Each risk should be assigned a severity level. A practical three-tier classification for segregation of duties works well in practice:

Severity	Definition	Example
High	Could enable direct financial fraud (\$\$\$), material financial loss, or manipulation of financial statements. Should be avoid in user access at all costs.	<p>A user who can create or modify vendor master data and also execute the payment run can invent a vendor, update an existing vendor's bank account to their own, and collect a payment before anyone notices the change. By the time the legitimate vendor raises a query, the money is gone.</p> <p>A user who can post journal entries and also reopen closed accounting periods can move costs between periods without leaving an obvious trail, understating expenses in one period to meet a profit target or shifting revenue to accelerate recognition. This is the kind of manipulation that leads to financial statement restatements and triggers regulatory investigations.</p>
Medium	Could enable unauthorized transactions, data manipulation, or process bypass with material business impact. Likely to result in auditors questioning financial statement balances, raising control deficiency findings, or requiring additional substantive testing. Requires immediate remediation or strong compensating controls that are actively monitored. Potential fine (legal entity) to be paid if mis-used.	<p>A user who can create purchase orders and also post goods receipts can complete the procurement cycle without any independent verification that goods were actually received. Over time this enables inflated inventory values, payments for goods never delivered, and vendor relationships that exist only on paper</p> <p>A user who can configure approval workflow thresholds and also approve transactions within those thresholds can quietly raise the limit above their own transaction values, effectively removing the approval requirement for their own activity without anyone formally authorizing the change</p> <p>A user who can adjust stock quantities and also post inventory write-offs can make small balance corrections that individually fall below materiality thresholds but cumulatively distort inventory values. Each adjustment looks routine. The pattern is only visible in aggregate</p>
Low	Increases the risk of error or unauthorized access but with limited direct financial exposure. Typically raises presentation and disclosure concerns rather than fraud risk. Findings at this level are unlikely to result in material misstatement but will appear in audit management letters and internal control reports	<p>A user who can display and export sensitive financial report or extract information outside the ERP system without leaving a meaningful audit trail. The immediate financial impact may be low but the data protection exposure.</p> <p>A user who can display vendor or customer pricing conditions. On its own this does not enable a transaction or create a direct financial loss. But access to pricing data creates an information advantage that can be used outside the system, in negotiations, in conversations with competitors, or in decisions that should be made without that visibility. It is the kind of access that looks harmless in a role review and only becomes a problem when someone asks why a particular person needed to see it in the first place.</p>

Three risk levels work better than four. Adding a 'Critical' category above 'High' feels intuitive at first, it seems logical to call out the most serious risks separately. In practice it creates more problems than it solves. The line between 'Critical' and 'High' becomes a constant source of debate, remediation conversations get stuck on classification rather than action, and the category tends to expand over time as stakeholders push conflicts up to Critical to signal urgency rather than to apply a meaningful distinction.

And then the reverse pressure starts. Business stakeholders begin pushing back on the 'Critical' category itself, arguing that the access cannot realistically be removed and therefore needs to be mitigated. So

'Critical' gets a compensating control pathway. Which immediately raises the question: if 'Critical' can now be mitigated, what does that make 'High'? Can 'High' be accepted? And if 'High' can be accepted, what exactly is 'Medium' for? The classification system that was supposed to simplify decisions ends up generating exactly the kind of circular conversation it was meant to prevent. In practice, four levels do not produce four clear outcomes. They produce four levels of negotiation. Three levels keep the conversation focused on what matters: what does this risk require us to do, and who is responsible for doing it.

'High' risks should not be mitigated, they should be removed. If a user has a High-level conflict, the answer is role redesign or access removal, not a compensating control and a signature. This is the category where direct financial fraud, material misstatement, or serious regulatory exposure is possible. There is no compensating control robust enough to make this access acceptable on an ongoing basis. The access should not exist.

'Medium' risks require an active, monitored compensating control. Acceptance without a control is not an option at this level. The control needs a named owner, a defined frequency, and evidence that it was actually executed. A medium risk with a compensating control that nobody checks is functionally the same as no control at all.

'Low' risks can be formally accepted with documented rationale, but that acceptance should be a conscious decision, not a default. Someone with appropriate authority should sign off on the acceptance, understand what they are accepting, and revisit it periodically rather than carrying it forward indefinitely.

Experience from SAP access projects shows a consistent pattern in how risks distribute across a well-built matrix: roughly 15-25% of conflicts fall into the 'High' category, around 45-55% into 'Medium', and 25-35% into 'Low'. If your matrix produces a significantly different distribution it is worth asking whether the risk definitions are calibrated correctly. A matrix where everything is 'High' has lost its ability to prioritize. A matrix where very little is 'High' has likely been softened to avoid difficult remediation conversations.

One note on system-level and superuser access which by nature are 'Critical' requires a separate category to cover basis administration, security administration, and roles that carry destructive or irreversible system capabilities, things like deleting financial documents, deleting or modifying audit logs, or removing transport controls. This is a legitimate and useful distinction, and it requires a separate governance workstream and 'Critical' level tag. That category is fundamentally different in nature. It is not about two people needing to share a process step. It is about certain system capabilities being so sensitive that they should barely exist in a production environment at all, let alone be assigned to a regular business user. Keep that conversation separate, govern it separately, and let the three-level matrix do its job on business process risks without the added complexity.

Getting the definitions right requires both sides of the business in the room. A risk defined only by the IT or security team will be challenged by process owners who do not recognize the business scenario being described. A risk defined without technical input will be imprecise at the mapping stage and produce unreliable results in the GRC tool. The best matrices are built in structured workshops where process owners define what the risk means in business terms and the technical team validates that the system objects reflect what is actually possible. That combination is what produces a matrix that both sides trust and are willing to act on.

**GRC
Hack #2**

Use three risk levels and agree what each one requires before the first risk is classified. **High** means direct financial fraud or material misstatement is possible and the access must be removed, not mitigated. **Medium** means material business impact is possible and an active compensating control is required. **Low** means limited financial exposure and formal acceptance with a named owner is sufficient. Experience from SAP access projects shows the distribution in a well-built matrix settles around 15-25% High, 45-55% Medium, and 25-35% Low.

2.3 Technical mapping: from risk to system objects

Once business risks are defined, each activity must be mapped to the technical objects that grant access to it in the system. This is where the business language of the SoD matrix gets translated into the precise technical conditions that a GRC tool can evaluate against actual user access.

In SAP ECC, that translation was relatively straightforward. Most access was controlled through transaction codes and authorization objects, and a risk rule typically specified which transaction codes, in combination with which authorization object values, constituted a conflicting activity. In SAP S/4HANA the picture is more complex. The same business activity may now be accessible through a classic GUI transaction, a Fiori application, an OData service called by a mobile or third-party interface, or a combination of all three. A technically complete mapping needs to cover all of these entry points, not just the ones that were relevant when the rule set was originally built.

A concrete example shows what complete technical mapping looks like in practice. Take a single business activity: posting a vendor document. In S/4HANA, this activity can be performed through multiple technical paths simultaneously. Through the classic GUI layer it is accessible via transaction codes for example -48, F-51, F-53, FB60, FB01, and several others. Through the Fiori layer, the same activity for F-48 is now covered by dedicated Fiori app "Post Supplier Down Payments" by a set of semantic object for example 'Supplier' and semantic action 'postDownPayment'. Beneath both of these sits the backend authorization object layer, where F_BKPF_BUK controls posting by company code, F_BKPF_KOA controls posting by account type, and F_BKPF_GSB controls posting by business area, each requiring activity value 01 to permit the action.

A risk rule that only checks the transaction code layer will miss every user who posts vendor documents exclusively through Fiori. A rule that checks fiori semantic objects and actions but does not validate the underlying authorization object values will produce false positives for users whose Fiori access is technically assigned but restricted at the object level in a way that prevents the action from being completed.

The current reality in most S/4HANA landscapes is that both worlds coexist. Most existing Fiori applications were built as front-end wrappers around classic backend functionality, which means they still trigger the same underlying authorization object checks as their GUI transaction equivalents. A user posting a vendor invoice through the Supplier-createIncomingInvoice Fiori app still needs F_BKPF_BUK, F_BKPF_KOA, and the relevant field values in the backend, just as they would if they were using FB60 directly. In that world, covering transaction codes and authorization objects gets you most of the way there, with Fiori app providing the additional Fiori access path coverage.

But this is changing. SAP is increasingly developing new S/4HANA applications natively for Fiori from the ground up, without a corresponding classic GUI transaction behind them. These applications do not map back to any T-code. They have their own authorization concept, their own OData services, and their own FAPP semantic object and action combinations that have no equivalent in the ECC or classic GUI

world. For these applications, a risk rule built entirely around transaction codes and traditional authorization objects will produce a false negative every time. The access exists, it is being used, and the SoD analysis will never find it because the technical objects it is checking were defined for a system architecture that this application was never part of.

Complete mapping today means covering all layers: transaction codes for classic access paths, service, semantic object and action combinations for Fiori, and the underlying authorization objects with their relevant field values. And it means building the discipline now to extend that mapping to native Fiori applications as they are deployed, before the gap between the system and the rule set becomes too wide to close.

This is particularly critical for organizations currently running SAP ECC who are planning or mid-way through a migration to S/4HANA. The SoD matrix built for ECC cannot simply be carried across. The technical objects that control access have changed significantly. Fiori applications introduce new access paths that have no equivalent in the ECC world. OData services controlled through S_SERVICE authorization objects expose business functions that were never accessible through the classic transaction layer. A matrix migration that maps old transaction codes to new ones without addressing these new access dimensions will have structural blind spots from day one of the new system going live.


Doing the technical mapping properly requires genuine SAP authorization expertise. It is not a task that can be completed by reading documentation alone. The people doing it need to understand how the authorization check sequence works in S/4HANA, how Fiori apps are linked to their underlying services and authorization objects, and how those relationships differ from the ECC model they are replacing. Several tools and approaches help with this work. System traces, particularly ST05 and STAUTHTRACE, allow the actual authorization checks triggered by a specific transaction or application to be captured and analyzed. The Fiori Apps Library provides a structured mapping of each application to its associated OData services, semantic objects, and actions. Authorization master data in the system itself can be interrogated to understand which objects are checked for a given activity and what field values are relevant.

Custom developments deserve particular attention at this stage and are consistently the area where the most significant gaps appear. Z-transactions, custom Fiori applications, and bespoke ABAP programs are often where shortcuts were taken during the original implementation. They may combine access to multiple sensitive functions in ways that standard SAP transactions would never allow, precisely because they were built to solve a specific business problem quickly without going through a formal authorization design review. Every custom object in the landscape needs to be assessed individually. There is no library, no SAP documentation, and no automated tool that will do this assessment for you. It requires someone who knows what the program does, understands the authorization implications, and has the patience to work through it systematically.

This is time-consuming work. In a large SAP landscape with years of accumulated custom development, a thorough technical mapping exercise can take weeks. The temptation is to scope it down, cover the standard transactions and leave the custom layer for later. That temptation should be resisted. The custom layer is precisely where the most organization-specific risks live, and leaving it out of the matrix means the GRC tool will never surface the conflicts that matter most to this particular business.

**GRC
Hack #3**

When migrating from ECC to S/4HANA, treat the SoD matrix as a migration workstream in its own right, not an afterthought. Every risk rule needs to be revalidated against the new technical objects, with Fiori and OData coverage built in from the start, not added later. Use system traces



to verify what the system actually checks rather than what the documentation says it should. And do not scope out the custom layer. Z-transactions and bespoke applications are where the shortcuts live and where the audit findings will come from if the mapping is not done properly.

2.4 The Matrix must be a living document

An SoD matrix built at go-live reflects the risk landscape of the system as it existed at that moment. It is a snapshot, not a control. The moment the system changes, and in any active SAP environment the system changes continuously, the matrix begins to drift away from reality. Without a defined process for keeping it current, that drift is invisible. The GRC tool keeps running, reports keep being produced, and nobody realizes that the rule set underneath it all stopped being accurate months or years ago.

The triggers for matrix updates are more frequent than most organizations expect. A new Fiori application goes live and introduces access to business activities that were not previously available through the user interface. A custom Z-transaction is built to support a local process requirement and grants combinations of access that standard transactions would never allow. An OData service is activated to support a mobile app or an integration and opens a new technical path to a sensitive function. A workflow is reconfigured, changing who can approve what and at which threshold. A new company code or plant is added, extending existing roles into a new organizational scope. Any of these changes can create a new SoD conflict or render an existing rule obsolete, and none of them automatically trigger a review of the matrix.

The practical consequence compounds over time. In the first year after go-live the gap between the matrix and the system is manageable. By year two it is noticeable. By year three, in a landscape that has been actively developed, the matrix may have significant blind spots covering some of the highest-risk access paths in the system. The GRC tool is still running. The reports still show a manageable number of conflicts. But the conflicts it is not finding are the ones that matter most, because they exist in the parts of the system that were built after the matrix was last reviewed.

Treating the matrix as a living document requires three things that most organizations do not have in place. First, a defined owner, a named individual or team with formal accountability for keeping the matrix current, not a project team that handed it over at go-live and moved on. Second, a change trigger process, a lightweight mechanism that flags relevant system changes, new transports, new roles, new applications, to the matrix owner for assessment before or shortly after they go live. Third, a periodic review cadence that goes beyond what the change trigger catches, a structured annual or biannual review that looks at the matrix as a whole, checks whether the business risk definitions still reflect how the processes actually work, and validates that the technical mappings still cover the right access paths.

None of this is complicated in principle. It is simply the kind of ongoing maintenance that gets deprioritized in every organization that treats the matrix as a project deliverable rather than a permanent operational asset. The cost of keeping it current is low and predictable. The cost of discovering, usually during an audit, that it has not been maintained for three years is neither.

GRC Hack #4

Build matrix review into your change management process, not your audit calendar. Every significant system change, new application, new role, new integration, should include a step that asks: does this change affect our SoD risk definitions or our technical mappings? That question takes minutes to answer for most changes and catches the gaps before they become findings. Waiting for an annual review means the matrix is always behind the system it is supposed to govern.

3

3. SoD in SAP S/4HANA

The new architecture, the new risks, and what ECC knowledge no longer covers

3.1 Why the ECC SoD Matrix Cannot Be Copied to S/4HANA

Migrating from SAP ECC to SAP S/4HANA is not a technical upgrade, it is a fundamental redesign of how users interact with the system and how business processes are executed. Organizations that attempt to copy their ECC SoD matrix directly into an S/4HANA environment will face two problems: their analysis will miss real risks that exist in the new environment, and it will generate false positives on risks that no longer materialize in the same way. Both undermine trust in the control framework.

The changes occur across two dimensions: the technical dimension, which affects how access is structured and controlled, and the business dimension, which affects which process steps are now available to which users and how approval logic works.

The **false positive** problem is more immediate and more visible. A false positive in SoD analysis means the GRC tool flags a conflict that looks real on paper but does not represent real access risk in the way the system actually works. In the context of an ECC-to-S/4HANA migration, false positives arise when the rule set still references transaction codes and authorization objects that existed in ECC but have been replaced, restructured, or made irrelevant by the new system architecture. A user may be flagged for a conflict involving a transaction they cannot actually execute, for example because the company has adopted a Fiori-first strategy and the classic GUI transaction is no longer accessible to business users even though it technically exists in the system. Another common source of false positives is an authorization object that no longer controls the activity it was mapped to in ECC, because the underlying process has been restructured in S/4HANA and the access check now happens through a different object entirely. The business challenges the finding, the security team cannot explain it clearly, and confidence in the entire analysis begins to erode.

A number of transaction codes that carried significant SoD weight in ECC fall into exactly this category in S/4HANA. The classic vendor and customer master data maintenance transactions example FK*, FD*, XK*, VD*, MK* and XD* have been replaced by the unified Business Partner model, where vendor data is now maintained through transaction BP and its associated Fiori applications, with access controlled through BP authorization objects rather than the vendor-specific objects that ECC SoD rules were built around. In inventory management, the classic goods movement transactions MB*, MB31, MIGO-related variants MB02, MB04, MBST, MBRL, and others have all been consolidated into the single transaction MIGO. An SoD rule that checks MB11 for goods issue posting or MB01 for goods receipt and treats them as distinct risk-carrying activities may not reflect how access is actually structured in S/4HANA, where MIGO controls all of these movements through a single transaction with movement type-level authorization differentiation.

In project systems, transactions CJ*, CJ03, CJ08, CJ13, CJ20, CN22, and CN23 have all been replaced by CJ20N. Rules that reference multiple project transaction codes as separate risk activities may be generating conflicts between transactions that are now all controlled through a single successor, producing findings that look like genuine SoD violations but reflect an access model that no longer exists.

In credit management, FD32 has been replaced by UKM_BP under the new SAP Credit Management component, with a fundamentally different authorization model. Rules built around FD32 will not fire

correctly against users whose credit management access is entirely within the new model, and rules that do fire may be checking authorization objects that are no longer relevant to how credit limits are actually maintained.

Credit management access has been restructured under the new S/4HANA Credit Management component, where the classic FD32 transaction and its authorization model have been replaced by Business Partner credit segment maintenance with a different set of authorization objects and Fiori-based exposure monitoring apps. Invoice cancellation, previously handled through MR8M with its own authorization checks, follows a different process and authorization path in S/4HANA. An SoD rule built around the ECC versions of these transaction codes will fire against S/4HANA users who do not actually have the relevant access in the new model, producing findings that the business will immediately challenge and that will erode confidence in the entire analysis.

The false negative problem is less visible but more dangerous. A false negative means the GRC tool reports no conflict when a real access risk exists. The analysis looks clean, the results are accepted, and the access that should have been flagged continues to exist undetected. In an S/4HANA environment running an ECC-era rule set, false negatives arise because the rule set has no coverage of the new access paths that S/4HANA introduced. The tool is not failing. It is doing exactly what it was configured to do. The problem is that it was configured for a system that no longer exists.

S/4HANA has introduced Fiori applications that enable sensitive business activities in ways that have no meaningful equivalent in the ECC authorization model. Bank account management is a clear example. In ECC, house bank and bank account maintenance was handled through transaction FI12, a relatively restricted configuration activity with limited business user access. In S/4HANA, this has been expanded into a dedicated Fiori-based Bank Account Management component. The Manage Bank Accounts app, identified as F1366A in the Fiori Apps Library, allows treasury users to create, modify, and deactivate bank account master data directly through the Fiori interface. Access is controlled through the service FCLM_BAM_ACCOUNTWD_SRV, the UI5 component `fin.cash.brm.bankaccount.manage`, and the semantic object `BankAccount` with semantic action `manageMasterDataBank`. None of these technical objects exist in ECC. An SoD rule built around FI12 will never fire against a user whose bank account access is entirely Fiori-based, regardless of how sensitive that access actually is.

The same gap applies to procurement approval governance. The Manage Workflow for Purchase Requisitions app allows business users to define and modify procurement approval paths dynamically through the Fiori interface, a capability that in ECC was restricted to IMG configuration performed by administrators and was never a user-facing transactional function. And in financial accounting, the Fiori-based journal entry template management provides a more accessible and flexible interface for creating and managing reusable posting templates, with an authorization model that has changed in ways that ECC-era rules built around FBD1 and sample document transactions may not fully reflect.

In each case the risk exists in the new system in a form that an ECC rule set was never designed to detect. The tool reports clean. The risk is live. And nobody knows.

3.2 The Technical Dimension: Fiori, OData, and the New Access Layers

In SAP ECC, user access was primarily governed by transaction codes and authorization objects. If a user did not have the T-code, they could not perform the action. SoD analysis therefore focused on whether a user or role combined two conflicting T-codes with the necessary authorization objects.

In SAP S/4HANA, this logic still applies at the backend level, but a new frontend layer has been introduced. Users no longer enter transaction codes in SAP GUI. Instead, they work through the Fiori Launchpad, where they access applications assigned to their roles. Each application communicates with the backend via an OData service, which has its own authorization check through the S_SERVICE object. This means user access now depends on the interaction of four components:

- The Fiori application (identified by its application ID, e.g. F0842A for Manage Purchase Orders)
- The OData service called by that application (e.g. MM_PUR_PO_MAINT_V2_SRV)
- The Launchpad catalog and Space/Page assignment
- The backend authorization objects (which remain largely the same as in ECC)

Missing any one of these components causes an access error. From an SoD perspective, this means that access risks can now arise not only at the transaction level but also at the level of Fiori applications and OData services - and the SoD matrix must represent this.

GRC Hack #5

If you don't add Fiori applications to your SoD matrix, your analysis will be structurally incomplete. Reports will both miss real user access risks and generate false positives. Fiori apps fall into two categories: new-style transactional apps built on SAPUI5 (which introduce genuinely new risk surfaces) and classic GUI-wrapper apps (which launch traditional T-codes from the Fiori tile but still require classic authorization objects). Both must be in scope.

GRC Hack #6

OData services form a new access layer for business processes. Their authorization operates independently from classic transaction-level checks in the backend. A user may have the classic MIRO authorization for invoice posting but lack access to the Fiori app that uses MM_SUPPLIER_INVOICE_MANAGE - or vice versa. Both directions of this gap need to be visible in your SoD analysis.

3.3 The Business Dimension: New Processes, New Risks

S/4HANA introduces genuinely new business capabilities that create SoD risks that did not exist in ECC. The most significant examples involve the flexible approval workflow model for purchase requisitions, purchase orders, and supplier invoices. In ECC, approval control was handled by the static Release Strategy model, based on authorization objects (M_EINK_FRG) with fields for release group and release code. The entire process was static and fully embedded in the transactional system.

In S/4HANA, this model has been replaced by dynamic workflows driven by Fiori applications such as Manage Workflow for Purchase Requisitions and Manage Purchase Order Workflows. Users with access to these applications can define approval paths, set financial thresholds, modify the approval hierarchy, and change conditions that trigger or bypass the workflow. This is powerful business functionality - and it is a new source of critical access risk.

New SoD Risk Example - S/4HANA Only

A user has authorization to change Cost Center or WBS element master data (assigning cost ownership) AND the ability to approve a purchase requisition for that same object. The same person can give themselves control over a cost center or project, then approve related purchases - violating SoD, bypassing budget control, and creating the potential for fraud. This risk did not exist in ECC because approval was controlled by static authorization objects, not by configurable ownership assignment.

Other new risk dimensions in S/4HANA include configuration risks (users who can modify workflow parameters and approval thresholds), automation risks (background workflows that perform actions without human confirmation), and integration risks arising from API and OData-based cross-module linkages between FI, MM, CO, and SD.

GRC Hack #7

Add a dedicated activity to your SoD matrix: Manage Workflow Configuration. Users with this access can change the procurement approval design itself - not just execute transactions within it. This is a new type of risk that goes beyond the traditional create/approve/post framework. Monitor access to Fiori apps like Manage Workflows for Centrally Managed Purchase Requisitions and the related backend service SWF_FLEX_DEF_SRV.

GRC Hack #8

In S/4HANA, the highest risks are often where processes are configured, not where they are executed. A user with workflow management access who formally lacks posting rights can still change how documents are approved - which is arguably more dangerous than posting access with strong compensating controls.

3.4 S/4HANA SoD Matrix: Technical Reference

The table below illustrates how a properly updated S/4HANA SoD matrix entry looks in practice - combining the T-code, Fiori app, intent, OData service, and backend authorization objects for each business activity.

Business Activity	T-Code	Fiori App	OData Service	Auth. Objects
Create Purchase Order	ME21N	F0842A	MM_PUR_PO_MAINT_V2_SRV	M_BEST_EKG, M_BEST_BSA, S_SERVICE
Invoice Posting	MIRO	F0859	MM_SUPPLIER_INVOICE_MANAGE	F_BKPF_BUK, M_RECH_WRK, S_SERVICE
Post Journal Entry	FB50	F0718	FAC_FINANCIALS_POSTING_SRV	F_BKPF_BUK, F_BKPF_KOA, S_SERVICE
Manage Vendor Bank Data	FK02	F2708	BP_BP_MANAGE_SRV	F_KNA1_APP, S_SERVICE
Manage Purchase Workflow	-	App varies	SWF_FLEX_DEF_SRV	S_SERVICE, SWF_FLEX

Sales Order Management	VA01	F1873	SD_F1873_SO_WL_SRV	V_VBAK_AAT, V_VBAK_VKO, S_SERVICE
------------------------	------	-------	--------------------	---

3.5 How to Approach the Matrix Migration in Practice

Migrating the SoD matrix from ECC to S/4HANA is not a configuration task. It is a project, and it needs to be treated as one. Organizations that try to handle it as a side activity alongside the main migration, something to be done in the final weeks before go-live by whoever has spare capacity, will produce a matrix that is incomplete from day one and will spend the next two years explaining to auditors why their S/4HANA SoD analysis keeps producing results nobody trusts.

The migration should start at the same time as the technical project, not after it. The decisions made during the implementation, which Fiori apps are activated, which workflows are configured, which custom developments are built, directly determine what the updated matrix needs to cover. If the SoD team is not involved in those decisions as they are being made, they will spend months reverse-engineering a system that has already gone live and trying to map risks onto an access landscape that was never designed with them in mind.

A practical migration sequence. [The first step](#) is a gap analysis of the existing ECC matrix. Every risk rule needs to be assessed against three questions.

- Does this risk still exist in S/4HANA in the same form?
- Has the technical mapping changed, meaning are the same T-codes and authorization objects still the right ones to check, or has the activity moved to Fiori and OData?
- And are there new access paths in S/4HANA that enable this activity that the existing rule does not cover?

This assessment will typically find that a significant proportion of existing rules need to be updated rather than simply carried across, and that a number of new rules need to be created from scratch to cover S/4HANA-specific risks.

The [second step](#) is identifying the new risk areas that did not exist in ECC. As described in section 3.3, workflow configuration access is the most significant of these, but it is not the only one. The gap analysis needs to be supplemented by a structured review of the new S/4HANA functionality that has been activated in this specific implementation, covering which new Fiori applications are in scope, which OData services are exposed, and which business process capabilities are genuinely new rather than just technically repackaged versions of existing ECC functionality. This step is significantly easier and more accurate when it is done alongside the project team during the blueprinting and process design phase rather than after go-live. During blueprinting, the project team is actively documenting which processes are changing, which new Fiori applications are being activated, and which capabilities are being introduced for the first time. The SoD team sitting in those workshops can identify new risk surfaces in real time, as the process decisions are being made, and translate them directly into matrix requirements before the system is built. That is a fundamentally different and more efficient conversation than the one that happens after go-live, when the system is live, the project team has disbanded, and the SoD team is trying to reconstruct what was activated and why by reading transport logs, interviewing people who have moved on to other projects, and reverse-engineering a process landscape that was never

documented with access risk in mind. The information is the same in both cases. The effort required to collect it is an order of magnitude different.

The **third step** is the technical remapping. For each risk rule, the full set of technical objects needs to be validated against the live system rather than against documentation. This means using STAUTHTRACE and ST05 to capture actual authorization checks, cross-referencing against the Fiori Apps Library to identify FAPP semantic object and action combinations, and validating OData service assignments through the S_SERVICE authorization object. This work needs to be done by people who understand both the business process and the SAP authorization model. It cannot be delegated to a team that knows only one side.

The **fourth step** is building the custom and Z-object inventory. Every Z-transaction, custom Fiori application, and bespoke ABAP program in the landscape needs to be catalogued, assessed for what business activities it enables, and mapped to the appropriate risk rules. This is consistently the most time-consuming part of the migration and the part most likely to be scoped out under time pressure. It should not be. Custom objects are where the organization-specific risks live and where the gap between the ECC matrix and the S/4HANA reality will be largest.

The **fifth step** is validation in the GRC tool. Once the updated rules are configured, the analysis needs to be run against a representative set of user profiles and the results reviewed for plausibility. This is not a formal audit review. It is a technical quality check: are the rules firing when they should, are they silent when they should be, and do the results reflect the access landscape the team knows exists in the system? False positive rates significantly above what was seen in ECC are a signal that the technical mapping is too broad. Unexpectedly clean results in areas known to carry risk are a signal that coverage is incomplete.

The output of this process should be a documented matrix that clearly distinguishes between rules carried across from ECC, rules updated for S/4HANA, and rules created specifically for new S/4HANA functionality. That distinction matters for two reasons. It tells the governance team where the most uncertainty sits and therefore where the first round of post-go-live reviews should focus. And it gives auditors a clear answer to the question of how the organization approached the matrix migration, which in 2026 is a question they are increasingly likely to ask.

GRC Hack #9

When migrating from ECC to S/4HANA, treat the SoD matrix as a migration workstream in its own right, not an afterthought. Every risk rule needs to be revalidated against the new technical objects, with Fiori and OData coverage built in from the start, not added later. Use system traces to verify what the system actually checks rather than what the documentation says it should. And do not scope out the custom layer. Z-transactions and bespoke applications are where the shortcuts live and where the audit findings will come from if the mapping is not done properly.

3.6 New Risk Areas Specific to S/4HANA Beyond Procurement

The workflow configuration risk described in section 3.3 is the most widely discussed S/4HANA-specific SoD risk, but it is not the only one. Each major process area in S/4HANA has changed in ways that create new risk surfaces that did not exist in ECC, and organizations that limit their matrix update to the procurement area will have blind spots in finance, HR, and supply chain that are just as significant.

In finance, the introduction of the Universal Journal in S/4HANA consolidates what were previously separate ledgers in ECC into a single data structure. This changes the risk profile of journal entry access. In ECC, a user posting to FI and a user posting to CO were operating in different ledgers with different authorization checks. In S/4HANA, a single posting can affect both simultaneously through the Universal Journal. A user with access to post manual journal entries now has the potential to affect financial and management accounting in a single transaction in ways that would have required two separate access rights in ECC. The SoD rule that covered manual journal entry posting in ECC may not capture the full scope of what that access now enables.

In HR and payroll, the SoD risk landscape in S/4HANA on-premise has not changed as dramatically as in procurement or finance, but it contains gaps that traditional matrices consistently fail to cover. The classic SAP HCM payroll engine continues to run largely unchanged, and the core payroll execution transactions remain similar to ECC. The more significant gap is in how HR master data access is defined in SoD rules. The authorization model for HR master data in SAP HCM operates through infotype-level controls using P_PERNR and P_ORGIN authorization objects rather than through the transaction-based model that most SoD matrices are built around. A user with access to maintain payroll-relevant infotypes, IT0008 for basic pay, IT0009 for bank account details, IT0014 for recurring deductions, and IT0015 for additional payments, combined with access to release or post payroll results, represents a genuine and often undetected SoD conflict that a matrix built around PA30 and payroll posting transactions may not surface at the infotype level where the real risk sits. For organizations running a hybrid landscape with SuccessFactors Employee Central for core HR alongside S/4HANA for finance, the SoD challenge is compounded further because the two systems operate on entirely separate authorization models with no native cross-system SoD analysis, meaning that a conflict spanning employee data in SuccessFactors and payroll posting in S/4HANA will not be visible in either system's individual analysis.

In supply chain and inventory management, S/4HANA's embedded Extended Warehouse Management and the integration between MM and SD through the new inventory model introduce new cross-module access combinations that did not exist in ECC. A user with access to manage stock transfers and also post goods issues across the integrated warehouse and logistics layer can affect inventory values, revenue recognition, and cost of goods sold in ways that span multiple process areas. These cross-module combinations are difficult to capture in a matrix that was designed around single-module risk definitions and has not been updated to reflect the integrated process model that S/4HANA enables.

In master data governance, S/4HANA's Central Business Partner model replaces the separate vendor, customer, and contact person master data structures of ECC with a single unified object. This means that access to maintain business partner master data now covers a wider scope than vendor master maintenance did in ECC. A user with BP maintenance access in S/4HANA may be able to modify data that would have required separate authorizations in ECC, and the SoD rules that were built around FK01, FK02, XK01, and XK02 transaction codes may not fully reflect the consolidated risk that BP maintenance access now represents.

**GRC
Hack #10**

Do not limit your S/4HANA matrix update to the procurement area. Finance, HR, supply chain, and master data governance each have process and technical changes that create new SoD risk surfaces. Run a structured review of every major process area against the question: what can a user do in S/4HANA through this process that they could not do in ECC with the same access profile? The answers will identify the new rules your matrix needs that no ECC-era template will ever contain

3.7 Validating the Updated Matrix

An updated matrix that has not been validated is a hypothesis, not a control. Once the migration work described in section 3.5 is complete and the new rules are configured in the GRC tool, the next step is to test that the matrix actually works as intended before relying on it to govern access decisions and support audit evidence.

Validation covers three things, each catching different failure modes. Technical validation confirms that the rules fire correctly in the GRC tool when a user has the relevant access combinations. This means testing each new or updated rule against a set of user profiles that are known to carry the conflicting access, verifying that the tool surfaces the conflict, and checking that the technical object combinations in the rule match what STAUTHTRACE shows as the actual authorization checks being performed in the system. A rule that looks correct in the GRC configuration but does not fire against a user who demonstrably has the conflicting access has a mapping error somewhere that needs to be found and corrected before the analysis can be trusted.

Coverage validation checks that the matrix does not have significant blind spots. This is harder than technical validation because it requires knowing what you do not know. A practical approach is to take a sample of user profiles that process owners identify as carrying broad access, senior finance users, procurement managers, and system administrators, and review the full set of their access against the matrix manually. If the manual review surfaces combinations that look risky but the GRC tool did not flag, those combinations need to be traced back to either a missing rule or a mapping gap. Doing this exercise on a targeted sample of high-risk profiles is more efficient than reviewing the entire user population and more likely to surface real gaps than a purely automated check.

Business validation confirms that the risk definitions still reflect how the business actually operates rather than how it operated at go-live. Business processes change, organizational structures evolve, and the access combinations that represented real risks at the time the matrix was built may have shifted in significance as the business has grown and changed. A structured review with process owners, covering each major risk area and asking whether the scenario described in the risk definition still reflects a genuine exposure in this business today, is the step that keeps the matrix connected to operational reality rather than becoming a historical document that governs a business it no longer accurately describes.

3.8 The 2026 Agenda: What makes S/4HANA SoD different right now

The topic of SoD in S/4HANA has been discussed since the platform was introduced, but 2026 is the year it has moved from a future planning consideration to an immediate operational reality for a large number of organizations. Several converging factors make this the moment when getting S/4HANA SoD right has become genuinely urgent in a way it was not two or three years ago.

Migration volume is part of it. The wave of S/4HANA projects that accelerated after SAP extended its ECC maintenance deadline has put a significant number of organizations into their first or second full audit cycle on the new platform. Gaps in matrix coverage that were theoretical before go-live are now appearing in findings that are documented and on record.

SAP's own trajectory is the second factor. Native Fiori application development has accelerated with each release, and the proportion of business functionality accessible only through the Fiori layer continues to grow. Organizations that deferred Fiori and OData coverage because most users were still on classic transactions are finding that assumption harder to sustain. The system keeps moving; the rule sets are not keeping pace.

Auditor sophistication is the third factor. Teams that two years ago were still applying ECC-era procedures to S/4HANA environments are now asking specific questions about Fiori coverage, OData service authorization, and workflow configuration access. Organizations that cannot demonstrate that their SoD matrix has been updated to reflect the S/4HANA architecture are increasingly receiving findings that go beyond the traditional access conflict findings and question whether the control framework itself is fit for purpose on the new platform.

The fourth is the emergence of AI-assisted access analysis tools, both within the SAP ecosystem and from third-party GRC vendors, that are beginning to change what is possible in terms of continuous monitoring and access risk simulation. These tools create new opportunities for organizations that have a well-maintained and technically complete matrix to move toward real-time SoD governance. But they also widen the gap between organizations that have done the foundational work and those that have not. An AI-assisted tool running against an outdated ECC-era rule set will produce AI-assisted noise rather than AI-assisted insight. The quality of the output is bounded by the quality of the rule set underneath it, and in 2026 that rule set either reflects how S/4HANA actually works or it does not.

The organizations that are best positioned in 2026 are those that treated the matrix migration as a first-class workstream during their S/4HANA project, involved the right people at the right time, built Fiori and OData coverage from the start, and established the governance processes to keep the matrix current as the system continues to evolve. For everyone else, the work is not behind them. It is waiting for them, growing more complex with every release and every audit cycle that passes without it being addressed.

4

4. Conducting the SoD Audit

From scope to remediation - the five-stage roadmap

4.1 SoD Matrix Foundation: Starting from a Validated Baseline

Every authorization audit begins with a scoping decision: which business processes, which systems, and which user populations are in scope. For organizations with a primarily SAP landscape, the scope typically covers the core financial processes (AP, AR, GL, Treasury), procurement (MM/SRM), and HR (HCM or SuccessFactors). For organizations with hybrid or multi-system environments, the scope must explicitly address how non-SAP systems are included.

The SoD matrix must be validated with business process owners before any analysis begins. A matrix built by IT without business input will inevitably generate findings that process owners do not recognize as meaningful, leading to disputes and delays in remediation.

That said, starting from a blank page is rarely the right approach. Most major GRC vendors provide standard rule sets that serve as a well-established benchmark, and these have been refined through years of real-world use. The practical challenge is that many of these standard rule sets were created years ago and are not regularly updated and there is cautious optimism that the 2026 GRC matrix templates may finally see meaningful revision.

The most persistent quality issue we encounter is not the rules themselves, but the risk descriptions attached to them. Many standard descriptions are vague and lack concrete examples. A risk entry that simply flags "PO creation and GR posting" without explaining what that combination actually enables in the system and more importantly how it can be exploited, provides little foundation for a productive workshop. Participants cannot calibrate the severity of a conflict they cannot visualize.

Effective workshops require enriched risk descriptions that include specific fraud patterns and step-by-step transaction sequences (Fiori or t-code chains) illustrating how a wrongdoing would actually be executed. Without this level of detail, discussions tend to drift in one of two directions: risks are either underestimated because participants do not recognize the operational pathway to abuse, or overestimated because the theoretical worst case is assumed without considering real system constraints. Neither outcome serves the remediation effort.

For the workshops we lead, we invest significant preparation time in updating risk descriptions with concrete examples and transaction-level fraud scenarios before the first session. A facilitator who has run these workshops before brings an additional advantage: they know in advance where the ambiguity lies, where business owners typically push back, and where genuine complexity requires more time to ensure that the workshop produces decisions rather than open questions.

GRC Hack #11

*Start SoD matrix design with business processes and risk areas and not transaction codes or technical objects. Anchor every rule to a business consequence stakeholders can own, enrich standard vendor rule sets with concrete fraud patterns and execution sequences, and bring a facilitator who has run these workshops before. **The difference between a productive workshop and an inconclusive one is usually preparation, not participation.***

4.2 Stage 2: Technical Mapping

Once the SoD risk matrix is validated, each business activity must be mapped to its technical counterparts in the system. In S/4HANA, this means mapping to t-codes, Fiori app ids, OData services, and authorization objects. For non-SAP systems in scope, it means defining the technical access structure in a format the GRC tool can process. The quality of the technical mapping determines the quality of everything downstream. A conflict that is not technically mapped will not appear in any report, no matter how advanced the tooling.

Vendor templates save significant time and should always be the starting point a flawed template is still better than a blank page. That said, standard SAP rule sets require careful review before use. A known category of error is where a single T-code covers both sides of an SoD conflict for example, a transaction that handles both posting and clearing of open items. Any user with that access will always trigger a conflict, making the finding unconditional and effectively unactionable. These cases must be identified and corrected in the ruleset before analysis begins, otherwise they inflate findings and erode process owner confidence in the results.

The more significant challenge is a custom code. Standard vendor matrices do not cover custom T-codes or Fiori extensions, and this is precisely where real-world risk tends to concentrate because custom developments are rarely designed with SoD in mind from the start. Two distinct problems appear in practice. The more common is missing documentation: the security logic exists at the object level but nobody recorded what authorization objects the code actually checks. The more serious is that no security requirements were ever defined or implemented at all. A common example is a custom mass posting transaction built to serve one company code that carries no organizational-level restrictions meaning any user with access can execute it across the entire system landscape. Nobody requested security requirements, nobody designed them, and nobody implemented them.

For custom code coverage, a practical triage approach works well: list all custom T-codes and Fiori apps and make an initial classification between display/reporting transactions and those that change data. Data-changing transactions are the priority for SoD analysis. For those lacking authorization documentation, authorization traces should be executed to identify which objects the code actually checks. These traces must be run on a recent production system copy, typically the QA system and not in live production, where the trace activity itself can impair data integrity. Developers should be directly involved in this source code and code execution analysis: they understand the code intent, can identify where security was never considered, and are best placed to assess whether missing object-level controls represent a documentation gap or a genuine design flaw that requires remediation.

Looking in 2026 and ahead, the clean core strategy is steadily reducing the volume of custom code in new and upgraded S/4HANA implementations. The era of custom multi-options cockpit power t-codes is largely over and Fiori and standard extensibility frameworks design for better UX and user experience have replaced them. However, country-specific localizations and regulatory requirements will always demand some system adjustment. Poland's JPK and KSeF obligations are a current example: technically necessary, locally driven, and outside the scope of any global vendor template. These extensions will remain a permanent feature of any serious SoD coverage program.

**GRC
Hack #12**

Do not rely on the standard out-of-the-box vendor matrix alone. Vendor templates are the right starting point, but they require two layers of work before use: correcting known ruleset errors, such as T-codes that unconditionally cover both sides of a conflict and extending coverage to include custom developments. For custom code without authorization documentation, run traces on a recent production copy with developer involvement. Classify custom transactions by data impact first: those that change data take priority over display and reporting transactions.

4.3 Stage 3: Opening Balance Analysis

Once the technical mapping is complete, the opening balance analysis is the moment the system shows access risks analysis state. This the full picture of where SoD conflicts and sensitive accesses actually exist across the user population. The GRC system applies the validated rulebook to the current authorization state of every user in scope, and for most organizations, what comes back is sobering.

The scale of findings in a first-time analysis is rarely small. In a typical mid-to-large S/4HANA landscape, it is not uncommon to see SoD risks present in over 95% of active dialogue users. Conflict counts are rarely evenly distributed and a small number of users tend to accumulate a disproportionate share of findings. In representative engagements, the top five users by conflict count can account for a significant portion of all findings in the system, and the single highest-exposure user may carry conflicts numbering in the dozens of distinct risk combinations. This concentration pattern matters for remediation planning: it means that targeted action on a small population can deliver outsized risk reduction.

Reports should account for organizational levels a conflict within the same company code is significantly different from one that only arises across company codes. They should also segment findings by process area. In many landscapes, Finance (FI) risks dominate, often representing the majority of all SoD findings. Knowing this distribution before remediation planning begins allows effort to be directed where exposure is highest.

Role design problems must be surfaced at this stage, not discovered later. The opening balance analysis will typically reveal that conflicts are not primarily a user assignment problem as they are a role design problem. When a single role contains both sides of a conflict, every user assigned to it inherits the risk. Fixing the role eliminates the finding across the entire user population it serves; fixing individual users one by one does not. The opening balance report must therefore present role-level analysis alongside user-level findings, and the action plan must address both dimensions.

Visualization is critical here. A spreadsheet of thousands of rows does not communicate system state as it obscures it. The most effective opening balance presentations show where problems are concentrated: which process areas carry the most risk, which roles are the structural source of conflicts, which users sit at the highest exposure, and how the risk profile breaks down between theoretical access (entitlement exists but the user has never executed both sides) and active risk (both sides of the conflict have been recently used in the system). This distinction is not cosmetic as it determines remediation priority. A user with a critical conflict they have never exercised represents a different urgency than one who actively executes both conflicting actions on a regular basis.

**GRC
Hack #13**

The best opening balance reports are not static lists but interactive task lists. They allow process owners to take direct action: assigning findings to responsible parties, submitting access for removal, attaching evidence of mitigating controls, or escalating for management review. Static reports generate work; interactive reports generate

decisions. The opening balance workshop is not a readout - it is the moment the action plan is born.

4.4 Stage 4: Remediation

Remediation is the stage where concrete decisions are made about what to do with each finding. The security team or process owners determine which authorizations should be revoked, which activities should be reorganized across roles, and what changes need to be made to the underlying authorization model. From this analysis, a technical specification is created for system administrators.

Before any remediation action is taken, the opening balance analysis must be translated into clear structural conclusions. In most first-time audits, the findings tell a consistent story: SoD risks are structural, not incidental. They arise from how roles were built, not from isolated user assignment mistakes. Business roles are typically over-permissioned and inconsistent so frequently combining operational, control, and configuration activities in a single role in ways that were never intentional but accumulated over years of incremental change. In this context, modifying existing roles does not eliminate the source of risk; it only preserves the historical errors in a slightly different form. A role model that cannot maintain SoD integrity today will not scale as business processes evolve, new users are onboarded, or system functionality expands. These conclusions have a direct implication for remediation strategy: patching individual users is not the answer. The structural diagnosis points to a structural response and role redesign.

Role redesign principles

When the opening balance analysis reveals systemic role-level problems, the remediation roadmap should be anchored in the following principles: Build rather than patch. Modifying existing roles accumulates technical debt and preserves historical design flaws in a slightly different form. A new role model, designed area by area from clean foundations, is more sustainable than any number of incremental corrections to what already exists.

Embed SoD as a design input, not a control layer added afterward. Conflicts discovered during risk analysis are design failures. A role catalog built with segregation in mind from the start will generate far fewer findings than one where SoD was retrofitted after the fact.

Standardize naming, construction, and scope across the organization. Inconsistency in role design is one of the primary reasons SoD conflicts remain invisible until a structured audit forces them into view.

Scope every role to the minimum access the function actually requires. Broad roles built for operational convenience are the structural origin of most SoD conflicts. Finally, deploy new roles in parallel with the retirement of existing ones, process area by process area. Phased migration minimizes operational disruption and prevents the old model from generating new risk during the transition.

Remediation roadmap

In practice, role redesign at scale is not completed in a single cycle. A pilot-first approach reduces risk and builds organizational confidence. A representative pilot in a high-exposure process area for example Finance is a natural starting point given its typically disproportionate share of SoD findings and allows the new design principles to be validated before broader rollout.

A structured pilot follows a consistent sequence: define the user population in scope, identify pilot participants using usage data from the opening balance analysis, analyze transaction usage for those users, consult with business owners on the proposed role catalog, build and validate the new roles, conduct functional testing, complete UAT, and then deploy the new roles while retiring the previous ones. This sequence is not bureaucratic process so each step exists because skipping it produces a role model that looks clean in the GRC tool but fails in production. The full remediation roadmap then follows in waves, applying the same sequence process area by process area, until the legacy role model has been fully replaced.

A critical principle applies throughout: before revoking access from individual users, improve the roles and the underlying authorization model first. User-by-user remediation without role cleanup produces a temporary improvement that reverts as soon as the next user is provisioned with the same flawed roles. Individual access revocations are appropriate for acute cases identified during the opening balance - users with active high-severity conflicts that cannot wait for the role redesign cycle. They are not a substitute for structural remediation.

GRC Hack #14

Allocate dedicated time and resources for remediation before the audit begins - and be explicit that the remediation workload will exceed the analysis phase. The opening balance produces findings in days; resolving them structurally takes months. Organizations that do not plan for this discover it the hard way when findings age without action and the next audit finds the same problems. The remediation roadmap should be agreed before the opening balance is presented, so that findings land in a process that is ready to act on them.

4.5 Stage 5: Continuous Monitoring

An access audit that ends at remediation is a point-in-time improvement that will erode. The final and most important stage is establishing continuous monitoring and a recurring process of checking for new conflicts as access changes, user populations evolve, and system functionality expands.

GRC-class tools deliver their greatest ROI at this stage, enabling automated recurring analyses, scheduled risk reports, alert mechanisms for high-severity findings, and integration with access request workflows to prevent new conflicts from being introduced in the first place.

Preventive controls: Stopping conflicts before they occur

Continuous monitoring alone is not sufficient if the access provisioning process remains unrestricted. One of the most common sources of SoD conflict accumulation is copy-based provisioning - where a new user is set up by copying an existing user's role assignments without evaluating whether that combination is appropriate for the new person's actual responsibilities. The copied user may carry years of accumulated access, including conflicts that were never resolved. Without a preventive SoD check at the point of access request, those conflicts are replicated instantly and at scale.

Every access grant - whether for a new user, a role addition, or a temporary access extension - should trigger a preventive SoD simulation before approval. If the requested access would create a conflict, the request should be flagged, routed for review, or blocked depending on the severity of the risk. This single control, consistently applied, prevents the opening balance from deteriorating between audit cycles.

The AI Frontier in Access Management

The access request process has historically been one of the weakest points in the SoD control environment - not because organizations lack the right tools, but because users and managers do not have the context to request access correctly. A user joining a finance team knows their job title, not which roles in the system correspond to their actual responsibilities. The result is over-provisioning by default: when in doubt, request more access, and let the audit find the problems later.

This is the area where AI integration will deliver the most significant near-term impact on SoD management. Rather than requiring users to navigate role catalogs they do not understand, an AI-assisted access request process can propose the right role set based on a combination of inputs: the user's department, their position in the business process, their stated responsibilities, and reference users in comparable roles - while simultaneously evaluating the SoD conflict profile of the proposed assignment and adjusting recommendations to minimize risk exposure before the request is submitted.

This is not a theoretical capability. We are currently working on a first implementation for smartGRC - a GRC tool integrated with a company's internal AI engine - that supports users through the access management process in precisely this way. The AI operates on local infrastructure, keeping sensitive authorization and organizational data within the company's own environment, and guides the requestor toward a role set that is both operationally sufficient and SoD-aware from the moment of request. The goal is to shift the control left: rather than detecting conflicts after access has been granted and used, prevent them from being requested in the first place.

**GRC
Hack #15**

The greatest value from continuous monitoring comes from pairing it with preventive controls at the access request stage. Scheduled recurring analyses tell you what has gone wrong; preventive SoD checks stop it from happening. In 2026, the most forward-looking implementations are adding AI-assisted role recommendation to this layer - reducing over-provisioning at the source and making SoD awareness a built-in feature of how access is requested, not an afterthought discovered in the next audit.

5

5.Regulatory frameworks & SoD obligations

SOX, GDPR, ISO 27001 what each requires and how SoD addresses it

SoD is not only a best practice, for many organizations it is a direct regulatory obligation. SOX, GDPR, and ISO 27001 each create explicit or implied requirements for access control and segregation of duties. Understanding what each framework actually demands and how auditors and regulators test for it, is essential for designing a GRC program that satisfies compliance requirements rather than merely appearing to.

5.1 Sarbanes-Oxley (SOX): Section 404

For publicly listed companies and their subsidiaries, SOX Section 404 requires management to assess and report on the effectiveness of internal control over financial reporting (ICFR). Access controls and SoD specifically are a core component of that assessment. External auditors evaluate whether the control environment is sufficient to prevent or detect material misstatements.

What is often underappreciated is that SOX auditors treat fraud risk as a top priority in every financial statement engagement, not as an edge case but as a baseline assumption. SoD is one of the primary preventive controls standing between a company and a fraud-related misstatement. When SoD is absent or poorly designed, auditors' assessed fraud risk increases and that elevated assessment changes the entire audit approach.

The second and equally important dimension is IT General Controls. When auditors place reliance on a financial system so trusting that SAP produces complete and accurate data. That reliance depends on ITGC operating effectively. Access management and SoD are central ITGC components. If they are not functioning, auditors cannot conclude that the system's outputs are reliable. The consequence is a shift to extensive manual substantive procedures: larger sample sizes, transaction-level journal entry reviews, additional reconciliation testing, and document-by-document verification of what the system would otherwise be trusted to confirm. In practice, this means significantly more audit work to reach the same conclusion at considerable cost to the organization in both time and audit fees.

In SAP environments, the access combinations that attract the most auditor attention cluster around procurement and payment processing. The ACFR report and procurement cycle data provide auditors with both the analytical lens and the transactional evidence to identify where SoD failures have created fraud exposure. A user who can create a vendor, approve a purchase order, post a goods receipt, and release a payment represents a textbook material weakness and these combinations appear more often than most organizations realize until a structured SoD analysis forces them into view.

What SOX Auditors Look For

- A documented, risk-based SoD matrix covering all financially significant processes
- SoD conflicts identified, tracked, and either remediated or formally mitigated with evidence
- Periodic access reviews with documented decisions and follow-up

- Enhanced controls over sensitive functions: payment execution, journal entry posting, master data changes
- Preventive controls at the provisioning stage, not only detective controls after the fact

5.2 GDPR - Access Control as a Data Protection Obligation

The General Data Protection Regulation does not use the term "segregation of duties" explicitly, but Article 5(1)(f) establishes the principle of integrity and confidentiality. IT requires that personal data be processed in a manner that ensures appropriate security, including protection against unauthorized access, loss, and destruction. Article 32 operationalizes this by requiring the implementation of appropriate technical and organizational measures to ensure ongoing confidentiality, integrity, availability, and resilience of processing systems. SoD is one of the access control measures that can contribute to satisfying this obligation, particularly where the risk of unauthorized or unintentional modification of personal data by internal users is material.

The practical GDPR risk in SAP environments is frequently not a dramatic data theft and it is quiet, structural over-exposure of personal data that nobody designed and nobody noticed until an incident made it visible.

A representative scenario: during an S/4HANA conversion, roles are created quickly to keep the business running. Security is deprioritized with the intention of fixing it after go-live. After cutover, almost every employee can use F4 help to search Business Partner personal like data private addresses, maiden names, sensitive contact details only through standard Fiori apps and T-codes such as FBL1N. Not by design. Not by business requirement. Simply because roles were built in a hurry and never reviewed. The root cause is typically the same: no Authorization Groups assigned to Business Partner data, broad or missing values in B_BUPA_GRP, and roles that were never subjected to a post-go-live security review. The result is massive over-exposure of personal data across the entire user population. "We'll fix it after go-live" had become permanent until a GDPR incident report made it urgent.

The lesson is structural: roles created quickly are not GDPR-safe by default. Business Partner data in S/4HANA requires explicit authorization group configuration. B_BUPA_GRP must be restricted. And go-live timelines are not a valid reason to defer security design. In practice, what ships at go-live tends to stay. GDPR relevance means that SoD analysis must extend beyond financial transactions to include HR systems containing employee personal data (particularly SuccessFactors or SAP HCM), customer data in CRM and order management processes, and any system where an administrator could access, modify, or export personal data without review or audit trail.

GRC Hack #16

Include HR and customer data systems in your SoD scope not just financial modules. Under GDPR, unauthorized access to personal data due to SoD failures is a notifiable breach with potential fines of up to 4% of global annual turnover. For most organizations, the reputational consequence arrives before the financial penalty. And always review B_BUPA_GRP in your S/4HANA roles as it is one of the most commonly misconfigured authorization objects in converted landscapes.

5.3 ISO/IEC 27001 - Access Control as a Core Control Domain

SO/IEC 27001:2022 Annex A includes access control as a dedicated control domain. Control A.5.3 specifically requires segregation of duties: conflicting duties and areas of responsibility shall be segregated to reduce opportunities for unauthorized or unintentional modification or misuse of assets. This is one of the few controls in ISO 27001 where the mechanism like SoD is explicitly named rather than implied.

For organizations pursuing or maintaining ISO 27001 certification, SoD is not optional. The Statement of Applicability must address it, and certification auditors will test whether the control is implemented and operating effectively. Documenting SoD as applicable in the SoA without evidence of actual implementation, a defined matrix, periodic review results, and remediation tracking and will not satisfy a competent certification audit.

What "operating effectively" means in practice: auditors will expect to see a defined SoD policy, a risk-based matrix with documented scope, evidence that reviews have been conducted at defined intervals, records of findings and how they were resolved, and a process for handling exceptions. A GRC tool with documented SoD analysis, periodic review evidence, and remediation tracking provides exactly the audit trail that certification audits require. Organizations that manage SoD through spreadsheets and email chains typically struggle to demonstrate operating effectiveness, not because the control does not exist, but because the evidence does not hold up under scrutiny.

5.4 Aligning Across Frameworks

Many organizations are subject to multiple frameworks simultaneously. A publicly listed company operating in the EU with an ISO 27001-certified IT environment must satisfy SOX, GDPR, and ISO 27001 requirements - all of which carry an access control and SoD dimension. The good news is that these requirements are largely complementary: the same SoD matrix, the same access review process, and the same GRC tooling can satisfy all three if properly designed and documented.

The risk is treating each framework as a separate compliance workstream. Organizations that build a SOX SoD program, a separate GDPR access control review, and a separate ISO 27001 control set end up with three overlapping but inconsistent processes - each requiring its own evidence, its own reporting cycle, and its own remediation tracking. A unified GRC program, anchored in a single well-designed SoD matrix with appropriate scope, eliminates this duplication and produces a control environment that is more robust precisely because it is not fragmented.

Framework	Primary SoD Obligation	Key Evidence Required
SOX s.404	Prevent/detect material misstatements in ICFR	Documented SoD matrix, access review results, remediation log, management sign-off
GDPR Art. 32	Appropriate technical security for personal data	Access controls over personal data, breach notification readiness, DPA records
ISO 27001 A.5.3	Segregation of conflicting duties	Statement of Applicability, SoD policy, evidence of periodic reviews, non-conformity log
Local statutory audit	Varies by jurisdiction - typically ICFR-aligned	Auditor-ready reports from GRC system, access history, mitigation documentation

6

6.AI & Automation in SoD Monitoring

What is real, what is coming, and what to watch out for

6.1 The State of AI in GRC today

Artificial intelligence is becoming a meaningful part of the GRC tooling landscape. The most mature AI capabilities currently available in production SoD tools fall into three categories: AI-assisted SoD matrix building, usage-pattern analysis, and risk prioritization. More advanced capabilities including natural language access risk querying, role design recommendation, and automated conflict prediction are in active development and early deployment.

It is important to distinguish between genuine AI capability and marketing. Many tools describe rule-based automation as "AI-powered." The meaningful differentiator is whether the system can learn from data improving its risk assessments as it observes how access is actually used, flagging anomalies that the rule-based SoD matrix would not catch, and suggesting risk definitions or matrix updates based on system change patterns.

In 2026, the most promising AI applications in GRC are not replacing human judgment as they are reducing the cognitive load required to exercise it well. Role selection, conflict simulation, usage pattern analysis, and risk prioritization are all areas where AI can surface the right information at the right moment, while the human remains accountable for the decision.

6.2 AI-Assisted SoD matrix building

One of the most time-consuming parts of any SoD program is building and maintaining the matrix. AI-assisted tools can significantly accelerate this process by analyzing the SAP system landscape to suggest which transaction codes and Fiori apps are in use, recommending risk combinations based on standard process knowledge, and surfacing examples of how specific conflicts have materialized historically.

Tools like smartGRC incorporate an internal AI engine operating on local infrastructure to keep sensitive authorization and organizational data within the company's own environment to support risk definition workshops. The AI suggests risky activity combinations, provides examples of how risks can materialize, and maps those risks to the relevant SAP transactions or Fiori apps. This shortens workshop preparation time and improves the quality of the resulting matrix by giving participants concrete, system-grounded examples to react to rather than abstract risk descriptions to interpret.

GRC Hack #17

Use AI assistance in the matrix-building workshop, not as a replacement for it. AI-generated risk suggestions are a powerful starting point, but they require validation by people who understand the specific business context, organizational structure, and process ownership model. An AI-generated SoD matrix that no process owner has reviewed is not a valid control and the human who approves it remains fully accountable for its content.

6.3 Usage analytics and active risk detection

Static SoD analysis answers the question: does this user have access to a conflicting combination? Usage analytics answers a more operationally critical question: has this user actually exercised that conflict?

In practice, the split between theoretical and active conflicts in a typical opening balance is significant. A substantial portion of flagged conflicts represent access combinations that exist in the authorization profile but where one or both sides have never been used or have not been used within any operationally meaningful timeframe. Without usage data, all findings look equally urgent. With usage data, the risk register becomes actionable: remediation effort concentrates on users who are actively executing both sides of a conflict, while theoretical conflicts can be assessed separately, mitigated, or scheduled for role cleanup in the next provisioning cycle.

This distinction also matters for audit defense. Providing evidence that a flagged conflict has never been operationally exercised is a materially different conversation with an auditor than presenting an unfiltered list of thousands of access combinations. Usage data does not eliminate the finding the access combination still needs to be addressed but it contextualizes risk in a way that supports proportionate, defensible remediation decisions.

GRC Hack #18

Usage data is your most powerful tool for prioritizing remediation and defending audit findings. If your GRC tool does not show actual transaction and Fiori app usage data alongside the risk report, you are working without one of the most important dimensions of the analysis. Access entitlement and access exercise are different things and auditors increasingly understand this distinction.

6.4 AI-Assisted Access Provisioning with SoD Simulation

The most effective preventive control is to stop SoD conflicts from being created in the first place. Access request workflows that include a real-time SoD simulation running the GRC rulebook against the user's existing access profile plus the requested access before approval can prevent the introduction of new conflicts at the point of provisioning.

One of the most common sources of conflict accumulation is copy-based provisioning: a new user is set up by copying an existing user's role assignments without evaluating whether that combination is appropriate. The copied user may carry years of accumulated access, including unresolved conflicts. Without a preventive SoD check, those conflicts are replicated instantly across every user provisioned the same way. A single flawed reference user can silently propagate a structural risk across an entire department.

AI adds a further layer of capability at this stage and this is where the most significant near-term optimization potential lies. Rather than requiring users to navigate role catalogs they do not understand, an AI-assisted provisioning process can propose the right role set based on the user's department, their position in the business process, their stated responsibilities, and comparable reference users, while simultaneously evaluating the SoD conflict profile of the proposed assignment and adjusting recommendations to minimize risk exposure before the request is submitted.

In 2026, this capability exists in prototype form. smartGRC's integration with a local AI engine supports users through exactly this workflow: the AI proposes roles based on a description of the tasks the user will perform, surfaces the SoD implications of each proposed combination, and presents options that balance operational access with conflict minimization. The user reviews the proposal and makes the final decision accountability remains human. The AI's role is to ensure that decision is informed, not to make it. The shift is straightforward: stop conflicts from being requested in the first place rather than detecting them after access has been granted and used.

**GRC
Hack #19**

Never provision by copy without a preventive SoD simulation. A copied user profile is not a clean starting point it is an inheritance of every access decision, shortcut, and unresolved conflict that accumulated in the reference user's history. AI-assisted role recommendation combined with real-time SoD simulation at the provisioning stage is the most effective way to keep the opening balance from deteriorating between audit cycles.

6.5 Firefighter Log Review and AI-Assisted Analysis

Firefighter SAP's emergency access mechanism sits outside the SoD control framework by design. It exists precisely to bypass normal access restrictions when an urgent operational situation requires it. That is its legitimate purpose. The control is not the access restriction itself but the review of what was done during the session.

In practice, Firefighter log reviews are one of the most commonly neglected controls in SAP environments. Logs are generated, but systematic review rarely happens with the frequency or rigor the control requires. This is an area where AI can add meaningful value: automatically scanning session logs for critical actions for example mass postings, configuration changes, financial document reversals, master data modifications and surfacing the subset of sessions that warrant human review rather than requiring a reviewer to manually assess every log entry.

This is not an SoD topic, but it is a closely related privileged access control and organizations that invest in SoD monitoring without addressing Firefighter log review have left a significant gap in their access control environment.

6.6 Risks and Limitations of AI in GRC

The potential of AI in GRC is real, but so are the risks of misapplication. The most important principle is one that applies across every AI use case in this domain: the human remains accountable for every decision the AI supports.

An AI-generated risk description that no process owner has validated is not a control. An AI-proposed role set that a manager approves without review has not been reviewed it has been rubber-stamped. The risk of AI in GRC is not that it will make wrong suggestions; it is that its suggestions will be accepted without the critical engagement that gives the control its meaning. AI reduces the effort required to make a good decision. It does not make the decision good by itself.

Content drift is the second problem. AI-assisted matrix building produces outputs that need ongoing ownership. A partially AI-generated matrix that nobody has formally validated will not hold up under audit scrutiny, and it will not be maintained as the system landscape changes.

The third concern is the local versus cloud architecture question. For GRC applications, where the data being processed includes user authorization profiles, role assignments, and organizational structures,

the decision to run AI on local infrastructure versus a cloud-based engine carries data governance implications that organizations need to address explicitly before deployment.

**GRC
Hack #20**

Treat AI output in GRC the same way you treat any other control input, it requires human review, formal sign-off, and an identified owner. AI accelerates the work; it does not substitute for the accountability that makes the control valid

7

7. Cloud, Hybrid, and Non-SAP Landscapes

SoD beyond the SAP boundary

7.1 The Hybrid Reality

The vast majority of organizations running SAP do not have a purely SAP landscape. They have SAP as the core ERP, surrounded by cloud-native HR systems (SuccessFactors, Workday), procurement platforms (Ariba, Coupa), CRM systems (Salesforce), and any number of industry-specific or custom applications. This hybrid reality creates an SoD challenge that traditional SAP-focused GRC approaches are structurally ill-equipped to address.

The fundamental problem is that SoD risks do not respect system boundaries. A user who can create a vendor in SAP and approve a payment in SAP has an SoD conflict entirely within one system. But a user who can create a vendor in a third-party SRM platform and approve the resulting payment in SAP has an equally real conflict that spans two systems and that conflict will be invisible to an SoD analysis that only covers one of them.

Two cross-system patterns come up consistently in practice. An HR administrator who can create or modify employee master data in SuccessFactors and also process payroll in SAP S/4HANA holds a conflict that neither system alone reveals. A user who can manage vendor records or approve purchase requisitions in Ariba and execute or release payments in SAP S/4HANA has a procurement conflict that is just as material as the classic MM/AP SoD risk — and invisible to any SAP-only analysis. These are not edge cases. In organizations running both platforms, these cross-system combinations are common, and they are among the least monitored access risks in the landscape precisely because the tooling has not kept pace with the architecture.

GRC Hack #21

Map your cross-system SoD risks explicitly. Draw the process flows across systems, identify where critical process steps (create, change, approve, execute, reconcile) occur in different systems, and ensure that your GRC analysis can correlate access across those system boundaries. Cross-system conflicts are among the most commonly exploited and least commonly detected access risks in hybrid landscapes.

7.2 SAP Cloud Solutions: SuccessFactors, Ariba, Concur

SAP's own cloud portfolio introduces specific SoD considerations. SuccessFactors HR processes include sensitive access to compensation data, organizational management, and position management. Areas where SoD between HR administrators, payroll processors, and managers is directly relevant to fraud prevention and data integrity. Ariba procurement processes involve vendor management, purchase order approval, and invoice processing are mirroring the SoD risk landscape of classic SAP MM/AP but in a cloud-native environment with a different technical access model.

SAP IAG (Identity Access Governance) is positioned as the governance platform for this landscape, designed to provide centralized SoD analysis across SAP S/4HANA, SuccessFactors, Ariba, and Concur. In principle, it is the natural choice for organizations with a predominantly SAP cloud or hybrid landscape. In practice, the current state of IAG is more limited than its positioning suggests. IAG covers only a small

portion of the full GRC access control capability that mature organizations require it addresses cloud identity and basic access governance, but falls significantly short of the analytical depth, remediation workflow support, and cross-system risk management that a comprehensive SoD program demands. Organizations evaluating IAG should assess it against their actual requirements rather than its roadmap, and should not assume that SAP's native cloud governance tooling is equivalent in capability to a full GRC platform implementation.

7.3 Non-SAP Systems: Oracle, Workday, and Custom Applications

For organizations with Oracle ERP, Workday, or significant custom application landscapes, the challenge is that most GRC tools are SAP-centric. The SoD analysis capabilities that work perfectly for SAP authorization objects and Fiori apps may have limited or no coverage for Oracle's responsibility model or Workday's domain security. The options are to use a GRC platform with native multi-system support, to integrate non-SAP systems via a standardized data export format, or to run separate analyses per system and aggregate findings manually, the last option being the least robust and the most likely to produce an incomplete picture.

Tools like smartGRC address this through a universal XML-based technical definition format that allows any system's access structure to be modeled and analyzed, enabling a truly cross-system SoD view regardless of the underlying ERP platform. For mid-sized organizations running mixed landscapes, this makes cross-system SoD analysis viable without requiring a full SAP GRC implementation.

GRC Hack #22

Do not accept a GRC solution that cannot cover your full application landscape. If 30% of your financially significant processes run outside SAP, an SoD analysis that only covers SAP is missing 30% of your risk - and providing false assurance that the remaining 70% is sufficient. Cross-system integration is not a nice-to-have feature; it is a requirement for a meaningful control.

7.4 Cloud Access Models and Their SoD Implications

Cloud-native applications typically use role-based access control (RBAC) models that are simpler than SAP's authorization object hierarchy, but that simplicity can mask significant SoD risk. In many cloud HR or finance platforms, a role named "Finance Manager" may combine create, approve, and reconcile capabilities that should be segregated. The label sounds appropriate; the actual access combination does not.

SoD analysis in cloud environments therefore requires decomposing role definitions to the level of individual capabilities or permissions and applying the same conflict logic that governs SAP analysis. This is more work, but it is the only way to achieve a consistent and comparable SoD posture across the landscape. The alternative is accepting the vendor's role design as inherently safe because it ships as a standard is not a control position that will withstand audit scrutiny. Vendor-defined roles reflect operational convenience, not segregation of duties.

8

8.GRC Tooling Landscape 2026

Choosing the right tool for your organization

8.1 SAP GRC 2026

SAP GRC 2026 represents a significant architectural shift rather than an incremental update to Access Control 12.0. The platform consolidates all GRC modules into a single unified product running on HANA: Access Control, Process Control, Risk Management, Assurance and Compliance, and UI Masking and Logging. This unified platform is available for both on-premises and private cloud deployments, with the stated goal of future-proofing GRC processes and enabling AI-driven use cases on a modern technical foundation. The early adopter release is planned for March 2026, followed by general availability in early Q3 2026, with no new SKUs required and no contract conversions needed for existing GRC v12 S4 customers. Customers running GRC v12 on non-HANA databases will need to complete a database conversion to HANA as part of their upgrade path.

The platform organizes GRC capability across five pillars: Enterprise Risk Management, Business Controls and Compliance, Identity and Access Governance, Security and Data Protection, and Reporting and Monitoring. SAP frames the 2026 release around three strategic themes: optimized user experience, extended integrations, and automation with AI. Each product line is expected to deliver innovations across all three dimensions.

From an Access Control perspective, the 2026 roadmap includes updated Fiori apps for access request and emergency access management workflows, an enhanced analytics dashboard covering risk violations by process and priority, and an extended ruleset maintenance capability that allows impact simulation before ruleset changes are applied. This last point is a practical improvement for organizations that currently manage complex custom matrix extensions without a safe way to test changes in advance. The Advanced Analytics and Reporting capability surfaces as Analytical List Page (ALP) and Overview Page (OVP) views in SAP Fiori, bringing KPIs, charts, and drilldowns together in a single interactive interface.

Business Role Management receives a mass reconciliation capability, allowing business roles to be assigned to multiple users in a single operation with filtering, streamlined selection, and real-time feedback on progress. Integration is extended to SAP Cloud Identity Services and Microsoft Entra ID, addressing the identity source fragmentation that has been a consistent operational challenge in hybrid landscapes. Cloud Identity Services is introduced as an additional identity data source, with Entra serving as a user data source specifically for hybrid deployments. The integration with SAP Cloud Identity Access Governance extends coverage to additional cloud applications beyond the current scope.

Emergency Access Management receives two notable additions. The EAM optimization for HANA DB introduces mass assignment and role reconciliation capabilities. Separately, a HANA Plugin update in ABAP enables password-less logon for FireFighter sessions and supports technical user provisioning to HANA DB, eliminating a dependency that has historically created maintenance overhead. The Read Access Log integration with EAM enables auditors to retrieve logs showing who accessed regulated business or personal information, client financials, employee records, and other confidential data, during FireFighter sessions, directly supporting audit and compliance reporting requirements.

Access Control Integration with SAP Task Center unifies access request approvals into a single inbox, allowing users to act across applications without navigating multiple approval interfaces. Risk Assessment for SuccessFactors is extended to include target populations in rulesets, improving Segregation of Duties accuracy and consistency across cloud and hybrid environments.

New Features

- **Access Approver Fiori App Enhancement:** The Access Approver Fiori app streamlines reviewing and approving access requests within SAP environments. Updated UX improvements reduce approval cycle time and align the workflow with the broader Fiori design standard.
- **Task Center Integration for SAP GRC Access Control:** Integration with SAP Task Center consolidates access control approval tasks into a single inbox alongside other workflow items. Users no longer need to navigate into the GRC application separately to process access requests.
- **Analytics Access for Reporting:** Direct analytics access within SAP GRC Access Control for SAP HANA 1.0 enables reporting on compliance status and risk exposure without requiring separate BI tooling. Analytical List Page and Overview Page views bring KPIs and drilldowns into the standard Fiori interface.
- **Business Role Assignment Reconciliation:** The Fiori-based reconciliation capability allows business roles to be assigned to multiple users in a single operation, with filtering, selection, and real-time progress feedback. This addresses one of the more time-consuming manual steps in large-scale provisioning cycles.

Enhanced Features

- **Risk Assessment for SuccessFactors:** Extended to include target populations in rulesets, improving SoD accuracy across HR processes. Organizations running SuccessFactors alongside S/4HANA can now apply consistent risk definitions across both platforms rather than managing separate rule sets.
- **Extended Rulesets:** The extended ruleset capability provides additional flexibility for modelling complex SoD scenarios that standard rule definitions do not cover. This is relevant for organizations with non-standard process flows or regulatory requirements that require custom conflict logic.
- **Firefighter Log Review Automation:** Automated scanning of Firefighter session logs surfaces the subset of sessions that warrant human review — mass postings, configuration changes, financial document reversals, master data modifications — rather than requiring manual review of every log entry. This directly addresses the gap described in section 6.5.
- **Cloud Identity Integration:** SAP Cloud Identity Services is introduced as an additional identity data source within SAP GRC AC for SAP HANA 1.0, with Microsoft Entra ID serving as a user data source for hybrid deployments. This reduces the identity fragmentation that has historically required manual reconciliation across on-premises and cloud identity providers.
- **Passwordless Logon and HANA Provisioning Enhancement:** The HANA Plugin update in ABAP enables password-less logon for Firefighter sessions and supports technical user provisioning to HANA DB, eliminating the credential management overhead that has been a recurring maintenance burden in EAM-heavy environments.

The AI capabilities in the 2026 roadmap are notable, though important context applies. Two features are currently scoped for Private Cloud Edition only: AI-assisted User Access Review that proposes recommended actions to simplify reviewer workload, and Augmented Access Management via Joule conversational AI for access request processes. The Joule integration specifically enables access request creation through natural language, with auto-fill of roles, auto-generated justifications, and the ability to model requests using reference users and User IDs. Organizations running standard on-premises GRC deployments should verify which AI features apply to their edition before including these

capabilities in their planning assumptions. SAP has consistently used Private Cloud Edition as the proving ground before broader rollout. The timeline for that wider availability is not defined in the current roadmap.

On Process Control, the 2026 release introduces Joule-assisted creation of data sources and business rules for automated monitoring. Rule definition for continuous control monitoring has traditionally required specialist technical knowledge. Joule-assisted creation removes that barrier. The Regulatory Insights integration on BTP adds automated delta analysis when regulatory requirements change, which addresses one of the more time-consuming aspects of keeping control frameworks aligned with evolving compliance obligations.

One practical note on the 2026 transition applies regardless of module. SAP positions the migration path as automated for existing on-premises and private cloud customers, but the risk repository requires independent validation. A platform migration that carries over unreviewed legacy content, including incomplete risk descriptions, uncorrected standard matrix errors, and coverage gaps for current S/4HANA functionality, will not deliver the improvement in risk coverage the platform is capable of providing. The technical migration and the content review are separate workstreams and both need to be planned for.

GRC Hack #23

Before committing to the GRC 2026 migration timeline, validate which AI and automation features are included in your deployment edition. Several of the most compelling capabilities in the 2026 roadmap are currently scoped for Private Cloud Edition only. Plan the content review of your risk repository as a parallel workstream to the technical migration. The platform upgrade does not resolve content quality issues by default

8.2 SAP IAG (Identity Access Governance)

SAP IAG is the cloud-native evolution of GRC Access Control, designed for hybrid and cloud-first landscapes. It delivers SoD analysis, access request workflows, and role recommendations across SAP S/4HANA, SuccessFactors, Ariba, and Concur. For organizations migrating to the cloud or running a hybrid SAP landscape, IAG provides a unified access governance layer with real-time analysis and browser-based workflow handling. In S/4HANA migration projects, IAG is increasingly positioned as the central platform for access risk management where cloud deployment is preferred.

IAG is built on SAP Business Technology Platform (BTP) and organized around five core services, each of which can be deployed independently or in combination depending on organizational requirements.

- **Access Analysis:** Provides real-time SoD and critical access risk evaluation using pre-delivered and customizable rulesets. Risk findings cover SoD conflicts, critical access combinations, and critical permission assignments across connected systems. Results include user-level and role-level analysis, with mitigation control assignment capabilities for risks that cannot be immediately remediated.
- **Access Request:** Delivers self-service request workflows for users to request access to on-premise and cloud applications, with SoD simulation run at request submission so approvers receive risk context alongside the request. Workflow configuration supports multi-stage approval chains, role-owner stages, and line-item auto-approval rules. HR trigger integration with SuccessFactors Employee Central allows access requests to be initiated automatically from joiner, mover, and leaver events.
- **Role Design:** Supports creation and lifecycle management of business roles across connected systems. Roles are defined at the business level and mapped to technical authorizations in the underlying applications. Machine learning-based role mining functionality supports a bottom-up

approach to role design, analyzing existing access patterns to suggest optimized role definitions. Business roles serve as the primary unit of access assignment, grouping technical authorizations by job function or organizational position.

- **Access Certification:** Manages periodic access review campaigns across both cloud and on-premise connected systems. Campaign administrators define scope, duration, and workflow templates; reviewers receive structured review tasks showing each user's access alongside SoD risk counts and usage data to support informed decisions. Completed certification records provide the documented evidence trail that SOX, ISO 27001, and statutory audit requirements expect.
- **Privileged Access Management (PAM):** Replaces the on-premise Firefighter mechanism for cloud and hybrid environments. Users submit time-limited requests for elevated access; approved sessions are monitored and logged. Automated log review surfaces non-compliant activities for human review rather than requiring manual assessment of every session record. PAM covers both cloud-native applications and on-premise S/4HANA systems connected via the SAP Cloud Connector.

For organizations running a hybrid landscape -on-premise S/4HANA alongside cloud applications — the IAG Bridge scenario connects IAG with SAP GRC Access Control, allowing GRC to govern on-premise access while IAG handles cloud-side governance. This avoids the need to migrate the full on-premise GRC configuration to the cloud while extending governance coverage to cloud applications. The practical implication is that organizations can adopt IAG incrementally, starting with cloud coverage and extending to on-premise over time, rather than treating it as a full replacement for an existing GRC implementation.

Organizations evaluating IAG should assess it against their specific requirements and current functional scope rather than its long-term roadmap. A detailed practical assessment of IAG's current capabilities relative to full GRC platform requirements is covered in Chapter 7.

8.3 smartGRC, Agility and cross-system coverage

smartGRC was designed to address the need for simpler, faster, and more flexible access risk management. It provides a unified platform for SoD matrix maintenance, conflict analysis, periodic access reviews, and integration with business processes. Unlike classical GRC tools, smartGRC can operate as a complement to SAP GRC/IAG or as a standalone audit platform, making it a practical option for mid-sized organizations and multi-system environments.

The tool supports direct extension of the SoD matrix with custom Fiori apps, OData services, and non-SAP systems, extracting authorization data and storing it in a universal XML format using a Composite and Atom structure. This enables modeling of any authorization concept and allows the system to automatically validate imports, flagging issues such as missing Atoms in Composite roles or Atoms assigned to nonexistent users. It integrates with a local AI engine for AI-assisted matrix building and risk definition, keeping sensitive authorization and organizational data within the organization's own infrastructure. The SoD matrix covers the current S/4HANA release, supports import and export in CSV and XML for ITSM and JIRA integration, and includes active reports with trend analysis over time. First results are typically achievable within one month of deployment.

The periodic access review capability, delivered through the smartReview module, addresses one of the most operationally painful aspects of compliance programs. In organizations managing thousands of users across multiple SAP systems, traditional review processes built on Excel files and email chains are slow, error-prone, and difficult to defend under audit scrutiny. smartReview replaces that with a structured

workflow covering scope definition, verifier assignment, decision-making with usage history and bulk operations, and implementation monitoring with a real-time status dashboard. Organizations using the module have reported review cycle times reduced by up to 75% compared to manual processes, with the added benefit that decisions are tracked through to actual system implementation rather than stopping at a recorded finding.

The reporting layer distinguishes between theoretical risk, where an access combination exists in a user's authorization profile, and active risk, where both sides of a conflict have actually been exercised within a defined period. Acting on it changes remediation: effort concentrates on users who are actively executing conflicting combinations, while theoretical conflicts can be assessed separately, mitigated, or scheduled for role cleanup. It also changes the conversation in external audits. An unfiltered list of thousands of access combinations and a prioritized register of active risks that have actually been executed are not the same thing, and auditors know the difference.

**GRC
Hack #24**

The difference between a useful GRC report and an overwhelming one is whether it tells you what to do next. Active risk reporting with usage data, interactive task assignment, and implementation tracking turns an access review from a compliance exercise into an operational control. A static list of findings with no action layer attached is not a control, it is a document.

8.4 One-Time Audit vs. Continuous GRC: The Decision Framework

Organizations frequently face a decision between commissioning a one-time external SoD audit and implementing a GRC system for continuous control. Both have a legitimate place, but they serve fundamentally different purposes, and confusing them leads to either underinvestment in continuous control or overinvestment in repeated point-in-time work.

When a one-time audit makes sense

A one-time audit is the right tool for an initial baseline assessment, establishing the current state of access risk before any remediation program begins, preparing for an upcoming external audit, or scoping a larger authorization project. It is fast to initiate, requires no tooling investment, and delivers actionable findings within weeks.

In practice, a well-executed one-time audit follows a straightforward process: authorization data is extracted from SAP using a lightweight reader tool that requires only table read access and makes no changes to the system, loaded into an analysis environment, evaluated against a validated SoD matrix, and presented as both an executive summary and detailed technical findings. The executive summary identifies areas requiring immediate action and sets priorities for both short-term corrections and longer-term role model improvements. The detailed output provides role- and user-level findings with sufficient technical specificity to act on directly. The entire process from data extraction to results typically takes two to three weeks.

A one-time audit is well suited to organizations with urgent information needs, budget constraints, limited internal infrastructure, or those who need a diagnosis before committing to a full GRC implementation.

Where one-time audits fall short

The fundamental limitation of a one-time audit is that it captures a moment in time. In a dynamic SAP environment, where users are added, roles change, organizational structures evolve, and new

functionality is deployed, the findings from an audit conducted six months ago may no longer reflect reality. Access risk is not static, and a snapshot cannot monitor a moving landscape.

The deeper structural problem is the remediation paradox: the purpose of an audit is to identify problems so they can be fixed. But if there is no continuous monitoring in place after remediation, the only way to verify that fixes held, and that new conflicts have not been introduced, is to commission another audit. This creates a cycle where audits are used not just for initial assessment but as a substitute for ongoing control. That cycle is both operationally fragile and increasingly expensive.

In practice, when an organization reaches the point of commissioning three to four one-time audits per year, the cumulative cost of those engagements typically exceeds the annual investment in a full continuous GRC platform. At that frequency, the economics clearly favor a permanent solution, and more importantly, the organization is still operating without the preventive controls, automated monitoring, and audit trail that a GRC system provides between engagements.

The decision threshold

A one-time audit is the right starting point. It is not the right long-term answer for any organization with dynamic access changes, regulatory obligations, or a remediation program that needs to be verified over time. The transition point from periodic auditing to continuous GRC is reached when the cost of repeated audits approaches the cost of a permanent solution, when regulatory requirements demand ongoing evidence rather than point-in-time snapshots, or when the organization has completed an initial remediation cycle and needs to protect that investment from eroding.

<h3>One-Time Audit</h3> <p>Best for:</p> <ul style="list-style-type: none"> Pre-audit snapshots and remediation planning Organizations with infrequent access changes First-time SoD assessments to baseline risk Budget-constrained environments 	<h3>Continuous GRC System</h3> <p>Best for:</p> <ul style="list-style-type: none"> SOX/regulated environments requiring ongoing evidence Dynamic SAP environments with frequent access changes Organizations scaling SoD across non-SAP systems Access provisioning with preventive SoD simulation
---	--

GRC Hack #25 A one-time audit is a snapshot. In dynamic SAP environments, access risks change constantly - new users, new roles, system changes, and organizational restructuring mean that findings from six months ago may no longer reflect reality. Without continuous oversight, repeated audits become inevitable and cumulative audit costs often exceed the investment in a GRC system within two years.

9

9. SoD Maturity Model

Where you are and where you need to be

9.1 The Five Levels of SoD Maturity

Most organizations exist somewhere on a spectrum from reactive and ad-hoc access management to proactive and automated continuous control. The following maturity model is a practical tool for benchmarking current state and planning the next stage of improvement.

Level	Description	Typical Characteristics
1 - Ad Hoc	No formal SoD program	No SoD matrix; access given based on requests; SoD only discussed when audit issues arise
2 - Reactive	Basic matrix exists; reviews are audit-driven	ECC-era SoD matrix; annual access review; manual Excel analysis; findings not tracked
3 - Defined	Structured process with tooling	Current SoD matrix covering key processes; GRC tool in use; periodic reviews; remediation tracked
4 - Managed	Data-driven, continuous monitoring	Usage data in reports; quarterly reviews; preventive SoD check in provisioning; non-SAP systems included
5 - Optimized	AI-assisted, fully integrated	Continuous real-time monitoring; AI-assisted matrix updates; cross-system SoD; integrated with ITSM/JIRA

Most organizations undergoing their first systematic SoD program are at Level 2 or transitioning from Level 2 to Level 3. The most common jump in a single program cycle is from Level 2 to Level 3 - establishing a current SoD matrix, deploying GRC tooling, and implementing a structured periodic review process. The jump to Level 4 typically requires integration of usage data and extension to non-SAP systems. Level 5 is an emerging state driven by the AI and automation capabilities described in Chapter 6.

9.2 Common Traps at Each Level

Understanding where you are is only useful if you also understand what keeps organizations stuck there. Each maturity level has a characteristic trap that prevents forward movement, and most of them have nothing to do with technology.

At Level 1, the trap is invisibility. There is no SoD program because no one has formally owned the problem. Access management is treated as an IT function, risk ownership sits nowhere in particular, and SoD only surfaces as a topic when an auditor raises it. The way out is not a tool, it is assigning accountability. Someone needs to own the SoD matrix and the access review process before any technology investment makes sense.

At Level 2, the trap is the annual cycle. The organization has a matrix and conducts reviews, but both are driven entirely by the audit calendar. The matrix was built years ago and has not been updated since the ECC implementation. Reviews produce findings that are documented and then largely forgotten until the next audit. The access landscape deteriorates steadily between cycles because there is no mechanism to detect new conflicts as they are created. Organizations at Level 2 often believe they are more mature than they are because the audit has not yet produced a material finding, but the absence of a finding is not evidence of a clean landscape, it is evidence of an incomplete analysis.

At Level 3, the trap is tooling without process ownership. The GRC system is running, the matrix has been updated, and periodic reviews are happening, but the business does not engage meaningfully with the process. Review decisions are made by IT teams who do not understand the business context, or by business managers who approve everything without genuine assessment because the volume is too high and the interface too complex. The tool is producing findings; the findings are not producing change. The way forward requires structural improvement to the review process itself: usage data to focus attention on active risks, interactive workflows that make decisions actionable, and genuine business owner engagement.

At Level 4, the trap is coverage complacency. The core SAP landscape is well-controlled, usage data is being used effectively, and preventive SoD checks are in place at provisioning. But hybrid systems, SuccessFactors, Ariba, Concur, custom applications, sit outside the SoD framework entirely. Cross-system conflicts that span both SAP and cloud platforms are not being detected. The organization has a mature control environment for 70% of its financially significant processes and no visibility into the remaining 30%. Auditors running hybrid-aware procedures will find it.

At Level 5, the trap is automation without accountability. AI-assisted matrix updates, automated risk prioritization, and conversational access request workflows all reduce the cognitive load required to manage access risk, but they also create the conditions for rubber-stamp approvals and content drift. If no one is reviewing AI-generated suggestions with genuine critical engagement, the control loses its meaning regardless of how sophisticated the technology behind it is. The principle established in Chapter 6 applies here with particular force: AI accelerates the work; it does not substitute for the accountability that makes the control valid.

9.3 Practical Roadmap to Move Up Levels

Moving from one maturity level to the next is not primarily a technology decision. The tooling is the enabling layer. The real work is process, ownership, and content.

The move from Level 1 to Level 2 requires three things: an owner, a matrix, and a review cycle. The owner does not need to be a dedicated GRC resource, in smaller organizations it is typically the head of internal controls or the CFO's office. The matrix does not need to be perfect, a validated vendor baseline covering the core financial processes is sufficient to start. The review cycle does not need to be quarterly, an annual review with genuine business owner participation is worth more than no review at all. The goal at this stage is to make the problem visible and owned.

The move from Level 2 to Level 3 is where most structured SoD programs begin in practice. It requires a current SoD matrix that reflects the actual system landscape, including S/4HANA Fiori apps, current custom developments, and any recent process changes, a GRC tool that makes analysis and review workflows manageable at scale, and a remediation tracking process that follows findings through to actual system implementation rather than stopping at a documented recommendation. This is also the stage where the role model improvement described in Chapter 4 becomes the priority: structural remediation of the role design produces sustainable results; user-by-user access revocation without fixing the underlying roles does not.

The move from Level 3 to Level 4 has three requirements, and most Level 3 organizations are short on at least two. Usage data needs to be integrated into risk reporting so active conflicts can be separated from theoretical ones. SoD analysis needs to extend to non-SAP systems so the cross-system patterns described in Chapter 7 are actually in scope. And preventive SoD checking at the access request stage needs to be operational — stopping conflicts from being created is worth more than any volume of detection after the fact.

The move from Level 4 to Level 5 is less a single transition and more a continuous improvement trajectory. The AI capabilities described in Chapter 6, AI-assisted matrix building, role recommendation at provisioning, conversational access request interfaces, are real and increasingly available, but they require a stable Level 4 foundation to deliver value. Organizations that attempt to deploy AI-assisted GRC capabilities on top of an immature process and a poorly maintained matrix will find that the AI accelerates the production of low-quality outputs rather than high-quality ones. Get the foundation right first.

GRC Hack #26

Before deciding what technology to invest in next, assess which maturity level you are actually at, not which level you aspire to be at. Most organizations overestimate their maturity by one level because they conflate having a tool with having a functioning process. A GRC system that is running but not producing decisions that are implemented in the system is a Level 2 organization with Level 3 tooling. The maturity is in the process and the outcomes, not in the software license. Progress from where you actually are, not from where you think you are, and remember that moving from Level 2 to Level 3 consistently delivers more value than moving from Level 4 to Level 5 theoretically.

Closing Thought**Your SoD program is your strongest internal control asset**

-
if it is designed well, maintained actively, and built on a foundation that reflects how the system actually works.

In S/4HANA, in cloud landscapes, and in an era of AI-assisted monitoring, the organizations that manage access risk most effectively are those that treat their SoD matrix not as a compliance document but as a living control mechanism - one that evolves with the business, the system, and the threat landscape. Every access review we have run across SAP ECC, S/4HANA, and hybrid landscapes has found the same pattern: the organizations with the cleanest access risk profile are not the ones with the most sophisticated tooling. They are the ones where someone senior owns the matrix, the process owners take the review seriously, and findings get implemented in the system rather than documented and filed. The tool matters. The process behind it matters more. S/4HANA, cloud integration, and AI-assisted analysis have raised the technical complexity of SoD significantly. The underlying failure mode has not changed. Access risk accumulates when no one is paying attention, and it compounds faster than most organizations expect.

Quick-Reference: GRC Hacks Summary

All 26 GRC Hacks from this guide at a glance:

#	GRC Hack
1	Never design roles without an SoD matrix. The matrix is the prerequisite, not a byproduct.
2	Build the matrix in cross-functional workshops. Business and IT input are both non-negotiable.
3	Use tools that support systems beyond SAP. Cross-system SoD risks are real and underdetected.
4	Treat the SoD matrix as a living document aligned to your change management cycle.
5	Add Fiori applications to your SoD matrix. New-style and classic apps must both be covered.
6	Include OData services - they form an independent access layer that T-code analysis misses.
7	Add Manage Workflow Configuration as a distinct SoD activity in S/4HANA.
8	The highest S/4HANA risks are where processes are configured, not just where they are executed.
9	Start audits with business processes, not T-code lists.
10	Do not limit your S/4HANA matrix update to procurement. Finance, HR, supply chain and master data governance each carry new risk surfaces.
11	Show actual usage data alongside risk findings to prioritize remediation intelligently.
12	Use interactive, actionable reports - not static lists - to drive remediation decisions.
13	Allocate dedicated time and resources for remediation before the analysis begins.
14	Monitor continuously. Quarterly beats annual every time.

15	Include HR and customer data systems for GDPR compliance - not just financial modules.
16	Use AI as a matrix-building accelerator, not a replacement for human validation.
17	Usage analytics is your most powerful tool for risk prioritization and audit defense.
18	Map cross-system SoD risks explicitly. Process flows do not stop at system boundaries.
19	Demand full landscape coverage from your GRC solution. Partial coverage provides false assurance.
20	Migrate your risk repository before upgrading to SAP GRC 2026.
21	A one-time audit is a snapshot. Continuous control is the destination.
22	Never provision access by copying a user. Every copy inherits unresolved conflicts. Use preventive SoD simulation at the provisioning stage.
23	Validate AI output in GRC the same way you validate any control input. AI accelerates the work; it does not replace human accountability.
24	Before committing to GRC 2026 migration, verify which AI features are available in your deployment edition. Plan content review as a separate workstream.
25	Active reports with usage data and implementation tracking turn access reviews into operational controls, not compliance exercises.
26	Assess your actual maturity level before investing in the next technology layer. Progress from where you are, not from where you aspire to be.

About the Author

Filip Nowak is a Partner at GRC Advisory, specializing in SAP authorization and access risk management. He has led SoD programs and GRC implementations for organizations across manufacturing, financial services, and retail, covering both SAP ECC and S/4HANA landscapes.

Resources

smartgrc.eu

SAP Fiori Apps Reference Library:

fioriappslibrary.hana.ondemand.com

SAP GRC 2026 / SAP IAG documentation:

help.sap.com

ACFE Occupational Fraud 2024 (*<https://www.acfe.com/rtn>): A Report to the Nations