



# Cross-system segregations of duties analysis in complex IT landscape



Pascal van Vugt MSc  
is a senior consultant at KPMG  
Advisory – GRC Technology.  
[vanvugt.pascal@kpmg.nl](mailto:vanvugt.pascal@kpmg.nl)



Hylke van Weperen MSc  
is a manager at KPMG Advisory –  
GRC Technology  
[vanweperen.hylke@kpmg.nl](mailto:vanweperen.hylke@kpmg.nl)

**This article explains the importance of access controls and segregation of duties in complex IT landscapes and elaborates on performing segregation of duties (SoD) analyses across multiple application systems. Practical tips for performing SoD analyses are outlined based on the lessons learned from a SoD project at a multinational financial services company. In this project, the Sofy Access Control platform solution was implemented to automate the SoD analysis and to overcome the challenges with SoD conflicts in an effective manner.**

# Ten lessons learned based on project at financial services company

## THE IMPORTANCE OF ACCESS CONTROLS AND SEGREGATION OF DUTIES

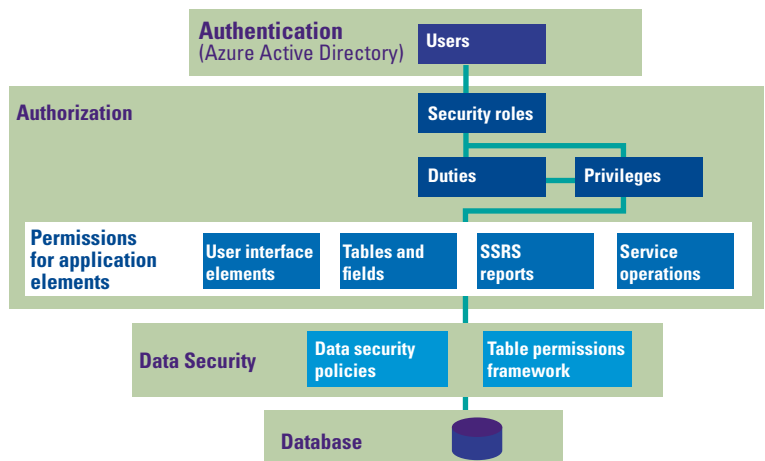
In a world where (digital) knowledge is power and the vast majority of all businesses work digitally to a large extent, security is an important element of the IT environment. Given that IT is more connected than ever, forming digital platforms, the need for a holistic security view over multiple platforms grows. Within security, the domain of access controls is tasked with the management of permissions, determining who can do what in an IT system. Setting the right level of permissions within a system is always a balancing act. If you set permissions too narrow, the system will become unworkable, but if you set permissions too broad, there will be an increased risk of security breaches. With employees switching functions, adding or dropping responsibilities and corresponding functions in applications, access management should be seen as an ongoing process.

Typically, the domain of access management consists of multiple safeguards or controls within the processes to ensure that the permissions handed out remain within the boundaries to keep it workable, which prevents security issues. The most common safeguards are the following:

1. User management procedures
2. Authorization (concept) reviews
3. Segregation of Duties (SoD) monitoring

Based on these areas, SoD monitoring is considered the most challenging for a number of reasons. Firstly, applications and their permission structures can be complex. Depending on the (type of) system, permissions can be either determined and granted in a structural way, for example through roles or profiles with (multiple layers of) underlying menus, functions, permissions or privileges, or in a less structural way by assigning individual

**Figure 1.** Example of layered access levels (based on Microsoft Dynamics).

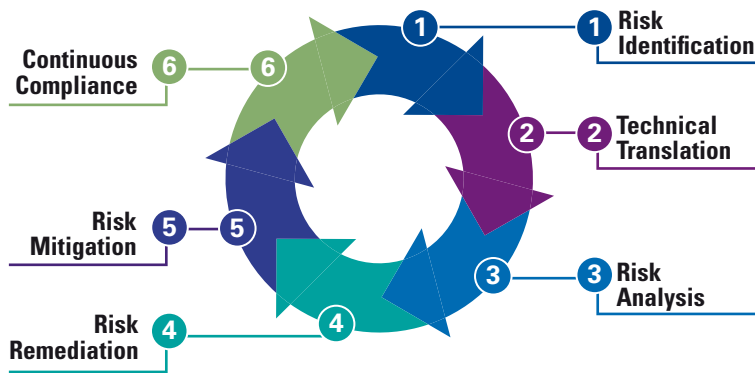


permissions to a user. An example of an application with multiple levels is displayed in Figure 1.

Secondly, combinations of assigned permissions or roles within the application need to be taken into account. It is insufficient to review the structure and the individual assignments to a user (e.g. a user has a role and can therefore execute a specific task in the application), as it will not detect the ability of a user to perform a task in the application by means of combination of permissions stemming from multiple roles.

Lastly, highly depending on the context, SoD conflicts might be inevitable. Whether it is an employee who needs a (temporary) backup colleague or a team that is simply too small to be split up for performing multiple tasks; sometimes it is just undesirable from a business efficiency perspective to enforce SoDs on a permission level in the application.

**Figure 2.** Example of layered access levels (based on Microsoft Dynamics).

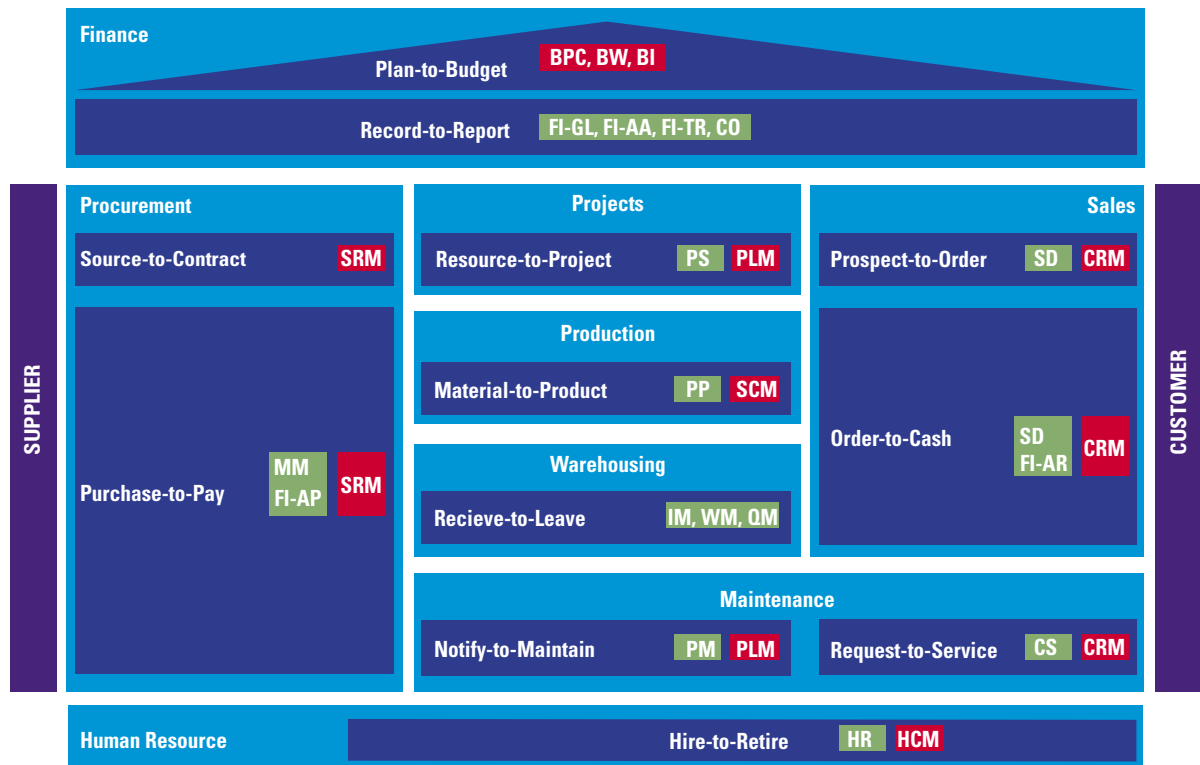


Businesses that strive to implement solid SoD monitoring and overcome the challenges, should take a structured approach. KPMG has developed a standardized method to put SoD monitoring and follow-up in place. This is a generic approach, not focused on specific technologies or (ERP) applications. The method consists of six steps that are required for control with respect to SoDs (see Figure 2):

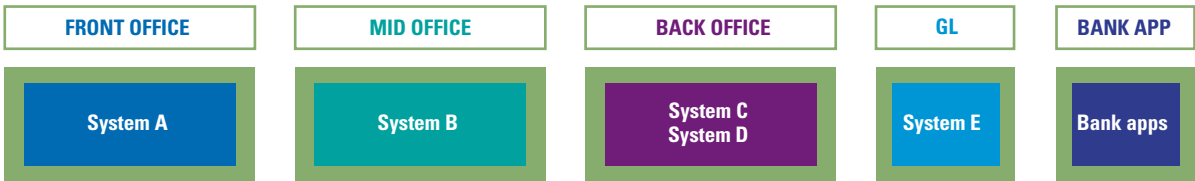
1. *Risk Identification.* Identify risks and the related SoD rules in business processes.
2. *Technical Translation.* Translate critical tasks into technical definitions of user permissions based on data extracted from the applicable application.
3. *Risk Analysis.* Use the data from your application(s) to analyze if users have possible combinations of critical tasks that are not allowed. These are called SoD conflicts.
4. *Risk Remediation.* Remediate the risks and fix the SoD conflicts by changing or revoking access rights from users.
5. *Risk Mitigation.* In case remediation is not possible or undesirable, mitigate the risks by implementing (automated) controls.
6. *Continuous Compliance.* Implement measures and tools to structurally monitor SoD conflicts, follow up on conflicts and demonstrate compliance to business owners, regulators and other stakeholders.

Later (see the section “Ten lessons learned for cross-system SoD monitoring”), we will elaborate on this method with practical examples of how these steps were used in the SoD project at a global financial services company which is primarily focused on leasing products.

**Figure 3.** Overview ERP system: example a company using the modules within a single SAP ERP system.



**Figure 4.** Overview scattered landscape: example of a leasing company using separate systems for set of processes.



**Figure 5.** Example of a SoD policy configured in SOFY, including a set of SoD rules and related risks.

The screenshot shows the SOFY interface with a sidebar on the left containing navigation links like 'Home - All apps', 'Summary Dashboard', 'User Statistics', 'Maintain Ruleset', 'SoD conflicts', 'Critical Actions', 'Critical Action Definitions', 'Business Controls', 'User Cockpit', 'Role Cockpit', 'Access Manual', and 'Risk Simulation'. The main content area is titled 'SoD Conflict Rules' and contains a table with the following data:

Conflict ID ↑	Conflict Name	Risk Description	Priority	Critical Action 1	Critical Action 2	Action
CD01	Create Purchase Order AND Approve Purchase Request	To avoid unauthorized purchase orders.	High	CTA03	CTA09	Maintain Mitigating Control
CD02	Approve Purchase Order AND Approve Purchase Request	To avoid unauthorized purchase orders.	High	CTA04	CTA09	Maintain Mitigating Control
CD03	Create Purchase Order AND Setting Create Purchase Request	To avoid unauthorized purchase orders.	High	CTA03	CTA10	Maintain Mitigating Control
CD04	Approve Purchase Order AND Create Purchase Request	To avoid unauthorized purchase orders.	High	CTA04	CTA10	Maintain Mitigating Control
CD05	Create Purchase Order AND Approve Purchase Order	To avoid unauthorized purchase orders.	High	CTA03	CTA04	Maintain Mitigating Control
CD06	Prepare Accounts Payable (AP) Master data AND Process Incoming Cost Invoices	Risk of unauthorized / fraudulent payments if a user is able to maintain vendor/dealer/supplier data and is authorized to create/maintain an invoice.	High	CTA01	CTA17	Maintain Mitigating Control

## THE NEED FOR CROSS-SYSTEM SOD ANALYSIS

Stemming from a period in which organizations only had one main application system supporting all key processes, the typical SoD analysis is focused on one single application. This would then usually be focused on either the Enterprise Resource Planning-system (ERP) or the financial or General Ledger (GL) system. Recent trends, such as the movement of applications towards the cloud, digitization and platform thinking put an emphasis on (inter-) connectivity of applications within the IT environment.

As a result, more and more organizations are abandoning the idea of one single application in which all activities are performed, reintroducing point solutions.

Figure 3 shows a schematic image of business processes within one single ERP application in comparison to the scattered landscape of a company that uses multiple applications in Figure 4.

When an organization decides to spread its processes over multiple applications, access management will, as a consequence, have to be maintained for and synchronized across all applications. Correspondingly, the risk function will have to follow suit and make sure that the controls to prevent increasing security breaches or SoD conflicts are in place for multiple applications. As such, SoDs should be monitored across multiple applications.

A specific example which is considered crucial would be the access to the banking or payment environment of an organization. Often fed by either the GL or ERP system, payment orders are sent to the bank for further approval and execution. A good example of the criticality of cross-system SoD conflicts is the situation in which in the GL or ERP systems is defined which parties should be paid for what amount, and where the actual payment (or approval thereof) is performed within the banking application. Imagine a user being able to administer bank accounts in the GL or ERP system and having the ability to approve payments within the banking system, enabling enrichment. In conclusion, as organizations have multiple applications within their IT environment covering their main business processes, user and access management should also be performed with a holistic view, overseeing all relevant applications.

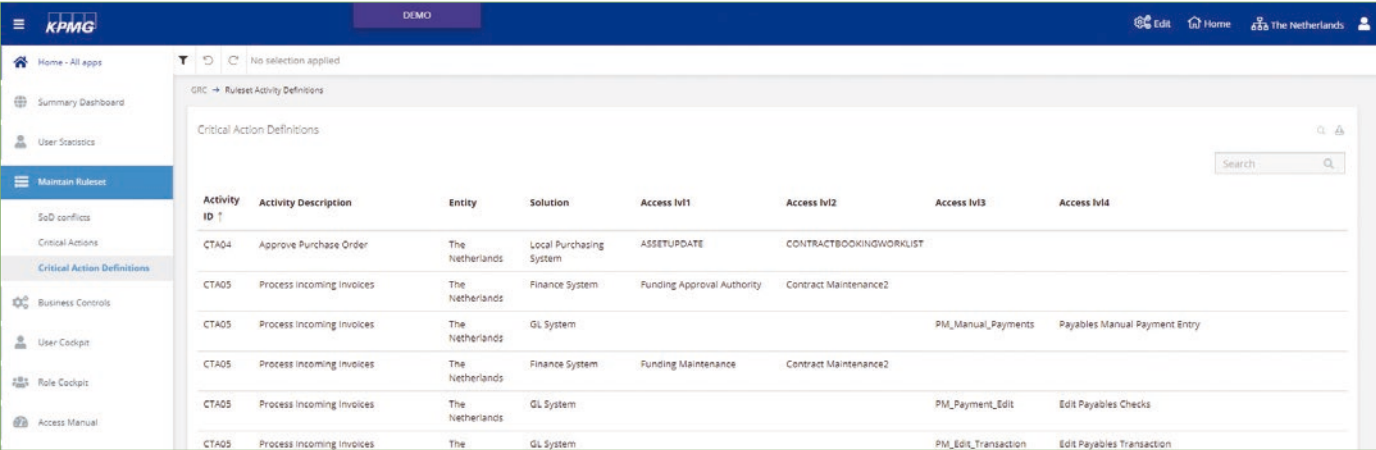
## TEN LESSONS LEARNED FOR CROSS-SYSTEM SOD MONITORING

### Introduction to the financial services company case

The financial services company (hereafter: the client) has identified challenges with respect to user access, user rights and Segregation of Duties (SoDs). Resolving the SoD issues has been proven complex due to several root causes such as unclear roles and responsibilities, knowledge gaps and limitations to application information.

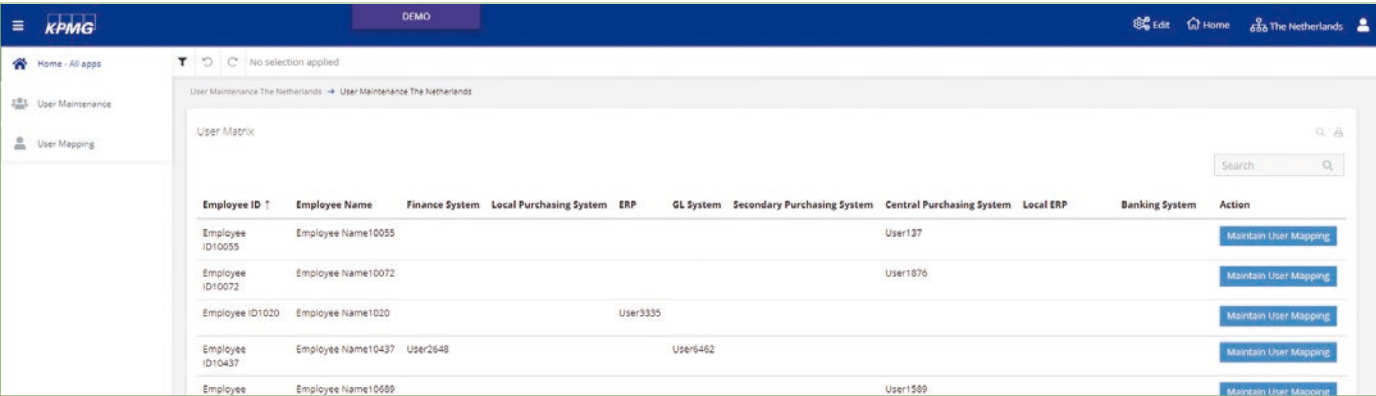


Figure 6. Critical activity definitions as defined in SOFY. The “access levels” refer to actual permissions required to perform the activity.



Activity ID ↑	Activity Description	Entity	Solution	Access Iv1	Access Iv2	Access Iv3	Access Iv4
CTA04	Approve Purchase Order	The Netherlands	Local Purchasing System	ASSETUPDATE	CONTRACTBOOKINGWORKLIST		
CTA05	Process Incoming Invoices	The Netherlands	Finance System	Funding Approval Authority	Contract Maintenance2		
CTA05	Process Incoming Invoices	The Netherlands	GL System			PM_Manual_Payments	Payables Manual Payment Entry
CTA05	Process Incoming Invoices	The Netherlands	Finance System	Funding Maintenance	Contract Maintenance2		
CTA05	Process Incoming Invoices	The Netherlands	GL System			PM_Payment_Edit	Edit Payables Checks
CTA05	Process Incoming Invoices	The Netherlands	GL System			PM_Edit_Transaction	Edit Payables Transaction

Figure 7. User maintenance function as implemented at the client, automatically mapping application user IDs to an employee.



Employee ID ↑	Employee Name	Finance System	Local Purchasing System	ERP	GL System	Secondary Purchasing System	Central Purchasing System	Local ERP	Banking System	Action
Employee ID10055	Employee Name10055						User137			Maintain User Mapping
Employee ID10072	Employee Name10072						User1876			Maintain User Mapping
Employee ID1020	Employee Name1020				User3335					Maintain User Mapping
Employee ID10437	Employee Name10437	User2648			User6462					Maintain User Mapping
Employee ID10689	Employee Name10689						User1589			Maintain User Mapping

A more centrally coordinated approach – coupled with local (business) responsibility and accountability – was required to successfully resolve these SoD issues and to design and implement a process to prevent similar issues going forward. The client embarked on a project – in corporation with KPMG – in which they analyzed and followed-up on possible cross-system SoD conflicts. The KPMG platform SOFY was implemented to support this project and is still being used as a tool to continuously demonstrate compliance. More than 40 applications used in more than 15 countries (local offices) were on-boarded on the platform by which the client was able to measure and analyze and mitigate cross-system SoD conflicts.

1 Starting with a well-defined risk-based policy

As a starting point for the SoD analysis, it is important that the content of the SoD policy is carefully drafted. The client has developed a “Global Policy on Segregation of Duties” covering mandatory SoD principles. The SoD principles are applicable to the client’s core transactional systems covering the core lease initiation and contract management processes. These primarily relate

to front-office, back-office, general ledger and pay-out / e-banking system. An example of such a principle is: “The person that activates contracts cannot be involved in payment activities.”

As a minimum, the combinations of critical activities and application functionality that are not allowed to be combined should be described, avoiding related risks. (see Figure 5). Where discussions on authorizations regarding business and IT might become complex, confusing or overwhelming, a sharply defined policy helps to easily conclude whether these combinations are accepted or lead to a SoD conflict and if so, which risks should be mitigated.

2 Defining critical activities using raw data and low level of detail

In order to start the SoD analysis, the SoD policy has to be translated into local permissions for each application in scope for the analysis. It is important to have the correct starting point for this translation; an overview of the local permissions of the application. It is recommended to use raw data and the lowest level of detail at which

authorizations are configured, as this can be beneficial to prove completeness to other stakeholders (such as external auditors). At the client project raw data was used (e.g. unedited dumps without any filtering, restrictions or other logic) so that any filtering and logic is applied within the analysis. IT people, key users and external IT partners/vendors were involved to determine the relevant tables and to extract those tables.

The reason to define critical activities at the lowest level of detail at which they are configured within the application, is to make the analysis more robust (see Figure 6). If we take a sample application in which authorizations are handed out at screen/menu level as the lowest level of detail and in the future the menu needed to execute the task is removed from the role, the definition at role level will no longer be valid, whereas the definition at menu level (which has a derivation of the authorization structure from role to screen level) will be updated automatically.

### 3 Setting up the analysis through intensive collaboration with both business and IT

When using raw data extracts as described previously, there is a risk the key user will not recognize the technical names of the menus (or other levels of permissions) when making the translation from critical activities into technical names in the application. In order to prevent that from happening, it is recommended to organize sessions with both IT and the business (primarily key users). During these sessions in the SoD project at the client, the business provided (practical) input by showing how tasks are performed within the application, whereas IT assisted in making the transition to the underlying technical permissions and the extraction of that data. We have called this activity the “technical translation”, as shown

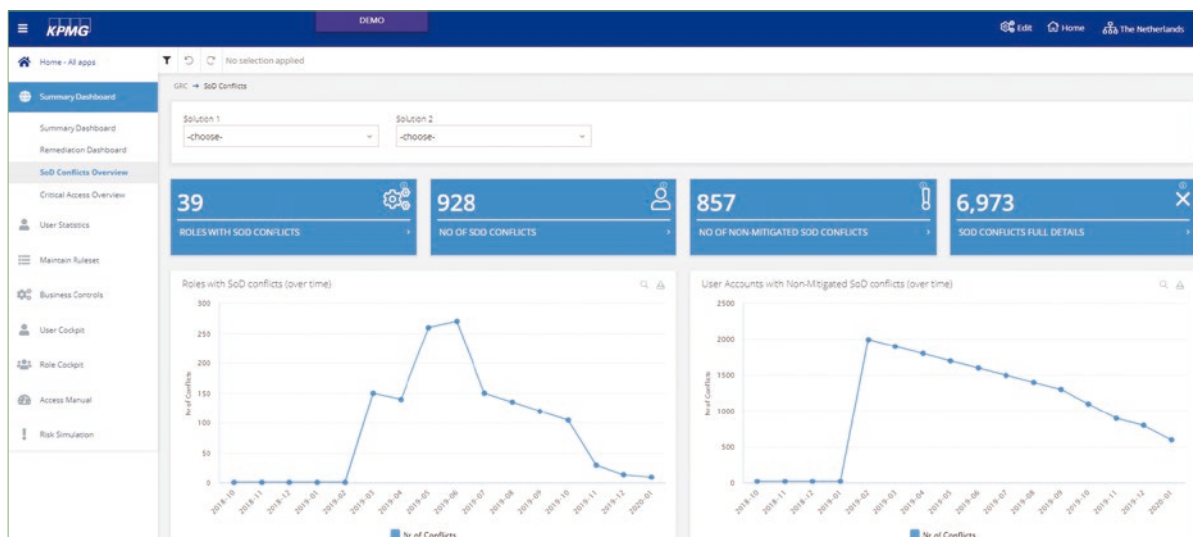
as step 2 of the “SoD Monitoring Model” discussed earlier (see Figure 2). Walkthroughs have proven to be effective and efficient ways of discovering all tasks that can be performed within an application. Alternatively, determining which users can execute which tasks is done by looking at historical transactional data (and depending on the information stored also which permissions they used).

The outcome of this analysis should result in a conclusion at the same level for each application (e.g. user X is able to execute task 1). Even though the underlying permission structure can be different for each application, it is important to conclude at the same level as input for the SoD analysis itself. Therein, the combination of critical tasks on a user level will be calculated and reported as a “hit” when these tasks are present as conflicting tasks within the SoD policy.

### 4 Linking users within an application to an employee ID

In order to be able to analyze permissions of the same employee over multiple applications, it needs to be identified which user accounts within the applications belong to the same employee. Within the client’s company, some employees were linked to different usernames over multiple applications due to application-specific constraints or naming conventions. As such, it is recommended to determine a unique employee ID (e.g. personnel number) and link each of the application users to that employee ID. Linking these accounts is preferably processed automatically in order to prevent manual, repetitious and error-prone efforts (see Figure 7). Prerequisites for automation would be a logical naming convention and sufficient user details (e.g. e-mail address or personnel number) stored within the application to create the links.

**Figure 8.** SoD Conflicts Overview dashboard; including timelines and functionalities to filter and “dive” into the results.



It is recommended to periodically review the list of users for which automated linking to the employee ID could not be performed, as it is a prerequisite for a cross-system SoD analysis. The SOFY solution provides functionality to maintain the user mappings.

## 5 Using tooling to validate the analysis in quick iterations

When discussing application permissions with key users and IT to define the critical tasks (as described in step 4), it can become quite abstract. In order to make the effects of the agreed upon definitions tangible, we recommend working with tooling. At the client, the SOFY platform was used to demonstrate the effects of including or omitting single permissions from the definition for an application by simulating the results (when applying the current definition). SOFY Access Control is a tailored tool with dashboards, KPIs and functions to analyze and dive into SoD conflicts and underlying user permissions (see Figure 8).

In the client's SOFY dashboards of the leasing company, it was chosen to focus on user and role level analysis. This means that the results include numbers and details of the users and roles with SoD conflicts, so root causes of conflicts could be analyzed on either one of these levels. This set-up of the analysis in the tool facilitates completeness in the analysis and follow-up; if no users and no user roles within applications contain SoD conflicts (including cross-system), there will no longer be any SoD risks.

Once the SoD analysis has been set up initially, and raw data extracts are used as described, the effort needed to

automate the extraction process and conduct a cross-system SoD analysis is usually limited. As authorizations change over time, due to employees joining, leaving or moving within the organization, controls related to authorizations and SoDs are typically executed periodically. In doing so, this will contribute to (the demonstration of) more control on the access management domain.

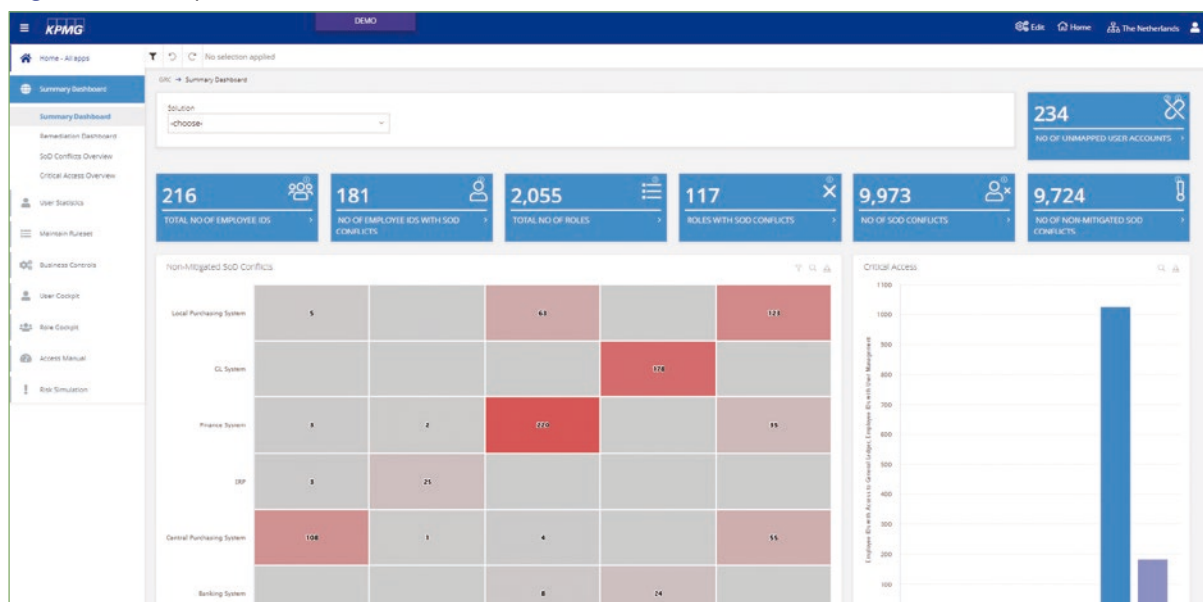
## 6 Following up on results in a phased manner

Once the analysis is done, it is time to start working with the results. The first step should always be to validate the outcome of the analysis. If the analysis is not correct, despite the efforts of key users and IT, it should be corrected. Any follow-up that should normally be performed can then be skipped, making it the most efficient option. However, when the analysis is validated and correct, follow-up should be done in a phased manner. We recommend starting the clean-up of the “low-hanging” targets to get familiar with the way of thinking and gain some momentum within your organization as a result of the improvement experienced. The following categories are considered “easy” categories:

1. Inactive users having SoD conflicts
2. Employees having multiple user accounts
3. Super users having (all) SoD conflicts
4. Roles with inherent SoD conflicts

These follow-up activities have taken place in a structured manner in the client's company, based on the details provided by the analysis and automated results in the SOFY application. For each of the abovementioned follow-up categories, clear dashboards, KPIs and overviews were created.

**Figure 9.** Summary Dashboard of SOFY Access Control.



## 7 Assigning responsibilities at a local, proper level

In order to address SoDs and resolve possible SoD conflicts, it is critical to have good operational governance. The client started with a project in which addressing the urgency and local responsibility for each of the company locations involved were important goals. The right tone at the top and proper post-project day-to-day governance made sure the organization kept paying attention to SoDs. To stay in control, the responsibilities of the three lines of defense were described as follows:

- 1st line (business and IT): Review and resolve SoD conflicts, functional sign-off translation roles/permissions into critical tasks (yearly process), advise within key IT projects: Risk authorization work stream, etc.;
- 2nd line: Maintain the SoD Policy, authorize possible exceptions, validate business controls (e.g. mitigating controls) of locations/countries, etc.;
- 3rd line: Conduct periodic reviews on implementation and effectiveness of SoD controls, perform reviews on mitigating controls, etc.

Like already addressed in lesson 7, the use of tooling is recommended to make this governance structure and processes feasible. Especially with the tasks outlined above, those involved should have access to proper tooling. For instance, the second line at the client has access to a KPI summary dashboard for monthly monitoring (targets) and managing the results (see Figure 9).

## 8 Only accept mitigation after exploring remediation options

Once responsibilities are assigned and the organization operates according to the designated Lines of Defense (see previous lesson), there are multiple ways to respond to SoD conflicts. Either remediation of the conflict by resolving the root cause or mitigation by minimizing or removing the resulting risk of the SoD conflict. Before handing out targets on reducing the number of conflicts, it is beneficial to reflect on the desired solution to the SoD conflicts first. Whereas mitigation of SoD conflicts is most often quickly arranged by implementing an additional check or another control (e.g. periodically take a sample for a number of transactions and check their validity), this might not be the preferred approach for an overall SoD solution. Especially when performing cross-system SoD analysis, there might be several different routes to a SoD conflict (when a task can be executed in multiple applications, this results in multiple routes for a SoD conflict), which in turn might require its own specific additional control.

When remediating, the solution is often slightly more difficult. Either adjusting the roles within the application, removing permissions from users (which can be accompanied with challenging discussions on why they

do need the access) or adding a control within the application, all require more effort than coming up with an additional control. However, these solutions do tend to provide a more permanent resolution to the SoD conflict, saving the periodical effort of having to perform the additional control. The SOFY platform was leveraged in the project to bring down the effort needed for remediation as well, for example by simulating the effects of removing permissions from roles.

## 9 Utilizing the available data for additional insights

Gathering authorization data of multiple applications in a single location, in combination with increased awareness and momentum on access management, can result in non-SoD related improvements. For example: the same data needed to analyze the permissions of a user or a role within an application can also be used to perform the regular user/role reviews as part of the access management controls. Even better, instead of reviewing all individual permissions contained in a role or assigned to a user, reviewing the critical activity level can be performed as well, making the review more efficient.

Secondly, due to the links between an application user and an employee, it is very easy to detect if there are any employees having multiple user accounts for one single application. This also helps to clean up the system as part of user management activities. Lastly, when combined with sources such as the Active Directory, interesting new insights can be generated. Employees that have left the organization (and are marked as such in the Active Directory) and are linked to active users in applications, can be easily listed. These exception reports are supportive with keeping an application authorizations clean and in reducing SoD conflicts.

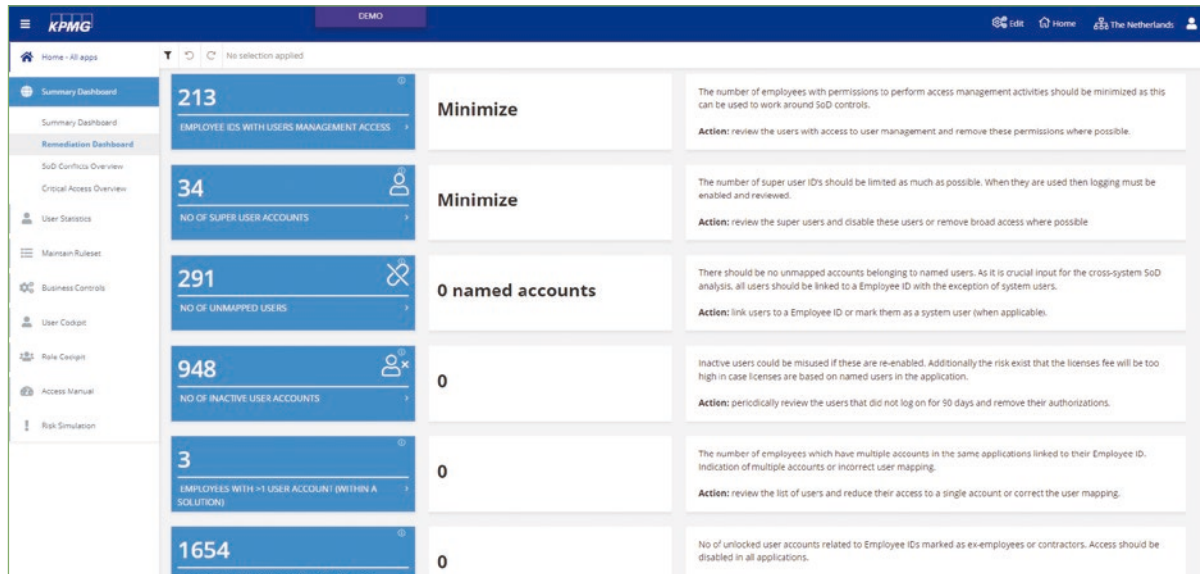
These examples were implemented as KPIs in a remediation dashboard in the SOFY tool of a financial services company. Figure 10 shows an overview of what such a dashboard would look like. The left-hand blue-colored column highlights the KPI actuals, whereas the middle column indicates the target value, and the right-hand column defines the KPI and advises on the required remediation action.

## 10 Aiming for a sustainable solution

Finally, when the analysis has been set up at the right level of detail, when it has been successfully automated and follow-up has been operationalized, one final aspect has to be taken into consideration. The subject of analysis (e.g. the IT environment analyzed for SoD conflicts) is subject to constant change. New functionality might be added, entire application systems might be phased out or introduced; this should all be reflected within the SoD analysis.



**Figure 10.** Remediation Dashboard of SOFY Access Control.



We recognize three different mechanisms to keep the reality reflected within the cross-system analysis:

1. The ongoing process of change management applicable to IT environments. Typically, changes are generated by well-structured processes or projects (depending on the size of the change) and should be evaluated on SoD relevance. If relevant, these changes should be communicated from the change process to be included in the analysis.
2. The second mechanism functions as a backup for the first and consists of a periodical review of all definitions applied in the SoD analysis. By periodically (for example yearly) distributing the current definitions which need to be conformed per application, it is ensured that any changes missed by applying the first mechanism are identified and that the definitions stay up to date.
3. As definitions are not the only critical input to a successful SoD analysis, other elements such as the linkages between application users and employees needs to be maintained as well. When a new employee joins the organization and obtains a new user account,

it needs to be (automatically) linked to the correct person. To include these prerequisites in the SoD KPIs reported, it is thereby enforced that these critical inputs are maintained during the entire year.

## CONCLUSION

In case an organization with a complex IT environment encounters challenges relating to access rights and SoDs, it is advised to use a platform to support the analysis. Moreover, the right platform will provide an organization with the tools for maintenance of the policies and (technical) rules which the SoD analyses are based on. The analysis should reveal conflicts on SoD and critical access and includes information on the related users, applications, roles and access rights. This information can be used to either remediate or mitigate conflicts. To monitor SoDs in a structured manner, it is key to automate the analyses and follow up on time. With the described 10 lessons learned, some practical tips are provided for a head start in their approach on SoDs and effectively demonstrate compliance.

## Reference

[Guntig] Günthardt, D., D. Hallemeesch & S. van der Giesen (2019). The lessons learned from did-do analytics on SAP. *Compact* 2019/1. Retrieved from: <https://www.compact.nl/articles/the-lessons-learned-from-did-do-analytics-on-sap/>

## About the authors

**Pascal van Vugt** is a senior consultant at KPMG Advisory – GRC Technology. Pascal has been working at KPMG since May 2016 and has performed multiple advisory engagements on IT Governance, Risk or Compliance (GRC) related topics. In these projects, he is specialized in digitizing internal controls by implementing GRC tooling.

**Hylke van Weperen** is a manager at KPMG Advisory – GRC Technology. He has been with KPMG since April 2014 and focuses on the intersection between business and IT. His areas of expertise are GRC tooling, data analysis, and access management. Recent projects were focused on authorization implementations (SAP) and authorization analysis (30+ applications).

We would like to thank Dennis Hallemeesch and Nick Jenniskens for their contribution to this article.