Paper 11684-2016

Multi-Tenancy in SAS® - Is It Worth the Fight?

Christopher Blake, RBS; Sergey Iglov, RBS; Dileep Pournami, RBS; Ekaitz Goienola, SAS UK Professional Services

ABSTRACT

At the Royal Bank of Scotland, one of our key organizational design principles is to "share everything we can share". In essence, this promotes the cross-departmental sharing of platform services. Historically, this was never enforced in our Business Intelligence platforms like SAS®, resulting in a diverse technology estate, which presents challenges to our platform team for maintaining software currency, software versions, and overall quality of service. Currently, we have SAS[®] 8.2 and SAS[®] 9.1.3 on the mainframe, SAS® 9.2, SAS® 9.3, and SAS® 9.4 across our Windows and Linux servers, and SAS® 9.1.3 and SAS® 9.4 on PC across the bank. One of the benefits to running a multi-tenant SAS environment is removing the need to procure, install, and configure a new environment when a new department wants to use SAS. However, the process of configuring a secure multi-tenant environment, using the default tools and procedures, can still be very labor intensive. This paper explains how we analyzed the benefits of creating a shared Enterprise Business Intelligence platform in SAS alongside the risks and organizational barriers to the approach. Several considerations are presented as well as some insight into how we managed to convince our key stakeholders with the approach. We also look at the "custom" processes and tools that RBS has implemented. Through this paper, we encourage other organizations to think about the various considerations we present to decide if sharing is right for their context to maximize the return on investment in SAS.

INTRODUCTION

SAS is RBS's strategic tool for Business Intelligence and Advanced Analytics. Over the years, there have been several SAS solutions installed in RBS, often without "sharing of platform" as a requirement. This has resulted in a diverse technology estate ranging from software versions more than 10 years old, to the modern SAS® Visual Analytics stack.

The platforms that we have are currently owned by various business teams and play a significant role in their day to do business operations. This ranges from ad hoc analysis, web based business intelligence reports to operational web applications which are used by RBS staff to serve customers in the branch as well as in call centers over the phone.

As part of the bank's technology rationalization agenda, there is a plan to simplify the SAS estate and consolidate into fewer platforms. Through this paper, we discuss the design considerations we went through into creating a single shared SAS platform to achieve this outcome. We will introduce the concept of 'multi-tenancy' on a shared platform, before we discuss how we gained stakeholder buy-in for our design and the enhancements we are making to the software ourselves, to support the multi-tenant design.

WHAT IS 'MULTI-TENANCY'

"Multitenancy is a reference to the mode of operation of software where multiple independent instances of one or multiple applications operate in a shared environment. The instances (tenants) are logically isolated, but physically integrated" - Gartner IT Glossary

In the context of a SAS environment, the principle means we will be using shared hardware as a starting point, to build several logical areas for our tenants. The tenants by and large, get the same software stack and a slice of the resources on the shared platform. The only exception to this is when we have a module that is needed only for a single tenant. E.g.: the fraud framework for the Fraud Analytics team

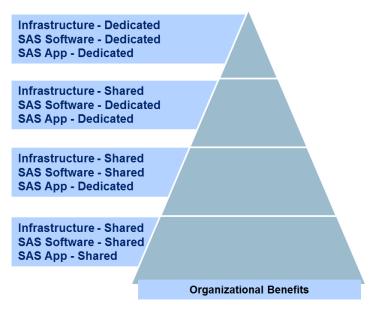


Diagram 1. Degrees of sharing available on a SAS platform

The diagram above describes the various degrees of sharing available on a SAS[®] platform. As you go from top to bottom, more and more of the resources on the platform are shared. So generally, this means increased utilization of resources. However, this also means that the individual application owners will need to give up some of their autonomy on the platform. This will have an impact on the way the platforms are used for day to day business operations. So it is important to strike the right balance between hardware utilization benefits and centralization of platform ownership.

CURRENT STATE

Currently in RBS, we have a diverse SAS estate, some of them are dedicated to business teams, but some environments do have some degrees of multi-tenancy or sharing incorporated. To give readers an idea of the diverse estate, here is a list of SAS solutions that the IT team is supporting at the moment:

- 1. SAS[®] Visual Analytics 7.1 a platform shared by 2 Customer Analytics departments.
- 2. SAS® 9.2 GRID a dedicated platform for Retail Risk Analytics.
- 3. SAS[®] 9.2 GRID mainly used by Wholesale Risk and some other teams. This platform is the closest we have to a multi-tenant design. Infrastructure and SAS software layers are shared
- 4. SAS[®] 9.3 GRID used by 3 teams for web reporting and text mining. The teams have their own SAS software instance, so in effect, only the infrastructure is shared.
- 5. SAS[®] 8.2 and 9.1.3 on the mainframe.
- 6. SAS® 9.1.3 and 9.4 PC solutions.

And this doesn't include the various Development, Test and Pre-Production environments that come with them!

This presents several challenges to IT including:

- Maintaining software currency Upgrading or even applying small hotfixes can be significant project work due to the number of environments involved.
- · High license costs.
- Majority of the engineering effort is spent on resolving version specific problems.
- Support skills legacy SAS skills can be difficult to find in the market.

For most of the end-users, this means that:

- they do not have access to the latest functionality or version available in the market.
- getting access to SAS, involves setting up a project to commission a new environment. i.e. a long time.

As SAS is the primary tool analysts in RBS depend on to perform their data analytics work, this has a negative impact overall for the bank. As an organization that has grown through several mergers and acquisitions, this is not just an isolated case for SAS. But that just adds to the complexity of the situation.

The bank is now focusing on simplifying and modernizing its IT estate. In the Data & Analytics space, we are adding Hadoop as a data capability alongside our Teradata Enterprise Data Warehouse. High level plans were agreed to also modernize the SAS estate. An organizational design principle was also introduced to promote sharing of platforms across various teams.

TARGET STATE

Since the idea for the bank's technology rationalization program was to have fewer platforms, the SAS[®] project team started from the position that we will have an enterprise scale 'multi-tenant' platform for SAS.

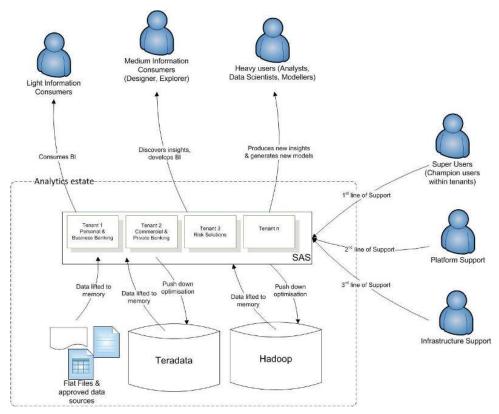


Diagram 2. High Level Solution Design

The diagram above shows how different departments will have logical areas on a shared platform. This platform will be the largest to-date in RBS; however, size is not the only thing that allows this platform to stand out from all others. A wide range of SAS components deployed and available for our service consumers helps us to satisfy all user groups, from web users (primarily report viewers/consumers) to ETL and model developers. The platform comprises of the following software stack aligned to support 3 high level capabilities as per below table.

Business Intelligence (BI)	Advanced Analytics	Data Preparation for Analytics
SAS [®] Visual Analytics	SAS [®] Enterprise Miner, SAS [®] Model Manager, SAS [®] Text Miner, SAS [®] Visual Statistics, SAS [®] Contextual Analysis	SAS [®] Data Integration Studio, SAS [®] Enterprise Guide

Table 1. High Level Capabilities

This is a major change for some of our business teams as SAS is currently used as a data warehouse as well as a data sourcing platform. As the high level design suggests, we will be using Teradata and Hadoop as our strategic data platforms. These 2 data platforms will have strong integration built into the SAS platform, which should enable us to avoid storing too much data in SAS. However, we do acknowledge in our design that Visual Analytics will maintain a hot cache of data in RAM and also a back up copy internally in Hadoop Distributed File System (HDFS) for resilience. Please note – the Hadoop shown in Diagram 2. High Level Solution Design is our enterprise BigData platform, not to be confused with the internal HDFS cluster in SAS® Visual Analytics.

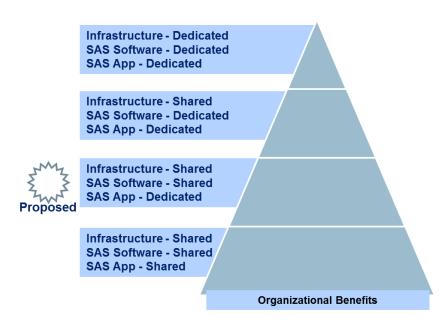


Diagram 3. Proposed multi-tenant approach

The diagram above shows our proposed design where Infrastructure and SAS software are shared, while tenants will have separate Application Server Contexts. This design is consistent with our Teradata and Hadoop platforms, which also follow a multi-tenant approach. Please refer to the section named 'Infrastructure view on multi-tenancy' for a more detailed understanding of our SAS implementation.

Even though this design makes sense from an architecture point of view, we quickly realised that we cannot take our stakeholder buy-in into this design for granted. The end-users and department sponsors currently enjoyed a lot of ownership and autonomy on the way some of the SAS platforms are maintained. They are landlords in effect, so why would they relinquish that control and move in as 'tenants'? We realised we had to win some opinions.

Ultimately, in a bank, 'cost' settles all disputes. So we looked for quantifiable benefits for multi-tenancy, and associated risks and mitigation. The actual cost benefit analysis we conducted is private to the bank; however, we are able to share some high level insight into the findings.

1. User on-boarding time will reduce by 80%. Adding a new business user group onto our shared platform will be obviously quicker than creating and configuring a separate platform for them. With some of our automation work, we hope to bring this even lower.

- 2. Software upgrade time will reduce by 66%. A single platform to upgrade, as opposed to 6 or 7.
- 3. Software integration work will reduce by 66%. If we license and deploy a new software component, e.g. a SAS/ACCESS[®] Interface to Teradata or Hadoop, we will have just one SAS platform to integrate to.
- 4. Save on license costs! SAS comes with lower limits on some the configuration (e.g.: a minimum of 4 nodes in a VA environment with distributed LASR™). Thus having a multi-tenant design means a lot of rationalisation opportunities. Please do the math based on your organization's per core price.

We also had some decent soft benefits to add on:

- 1. A single SAS platform means it will be easier for Architecture and Data Governance teams to apply policy and visualise data lineage.
- 2. For the users, this also means that everyone gets any new functionality all at once.

But what is the catch? Yes, there is one. It is not easy. As part of the analysis, we looked at the risks of a multi-tenant approach:

- 1. High degree of orchestration required between departments for downtimes.
- 2. The security model needs to accommodate the entire user base.
- 3. Not all modules of the SAS software stack are multi-tenant aware (e.g.: theme files in VA).
- 4. Workload management and resiliency requirements increase.

Points 1 - 3 can be somewhat addressed by investing in 3 key areas of expertise:

- a) Service Management: rigorous planning required to orchestrate downtimes.
- b) Engineering: right skills required to certify software builds and upgrades into a multi-tenant platform. Engineering is also key in customising the software stack to make it fit in the multi-tenant context.
- c) Design: right skills required in understanding security model requirements from a wide range of teams, to work out a design that meets all their requirements.

However, workload management and resiliency were the focal points in our debate with the 'landlords'. Even though SAS® 9.4 provides several architectural improvements, we realised we needed to look a bit deeper, given our multi-tenant design.

BUILDING BLOCKS OF THE PLATFORM

Bankwide Genie, the business name for the SAS® 9.4 Shared GRID platform, is the first SAS service in the bank built with multi-tenancy as the basis for the platform design. It is also the largest and most complex in the organization to-date. It is a combination of the traditional SAS processing over SAS® Grid Manager alongside a cluster of servers hosting SAS® Visual Analytics and SAS® Visual Statistics. This combination allows our customers to leverage both the high performance analytics capabilities delivered by a grid-enabled configuration and in-memory analytics processing. It also allows IT, as service providers, to target the offering to the business users more carefully and ensure that we do not expose functionality of the platform that is not required and at the same time allow users access to the capabilities that will help them to get maximum out of the platform.

In this section, we will take a deep dive into 3 important aspects of our platform design:

- 1. Infrastructure Design that is built around the principles of multi-tenancy, incorporates measures to ensure resilience and efficient workload management.
- 2. Security Model that is generic enough for a shared platform, but at the same time, flexible to meet specific tenant requirements.
- 3. Custom Processes processes and applications we have built in-house to make the SAS software stack more multi-tenant aware.

INFRASTRUCTURE DESIGN

Diagram 4 below depicts the high level platform design. Its complexity is driven by the need to address non-functional requirements such as scalability, resiliency and availability, and workload management to support our multi-tenancy principle.

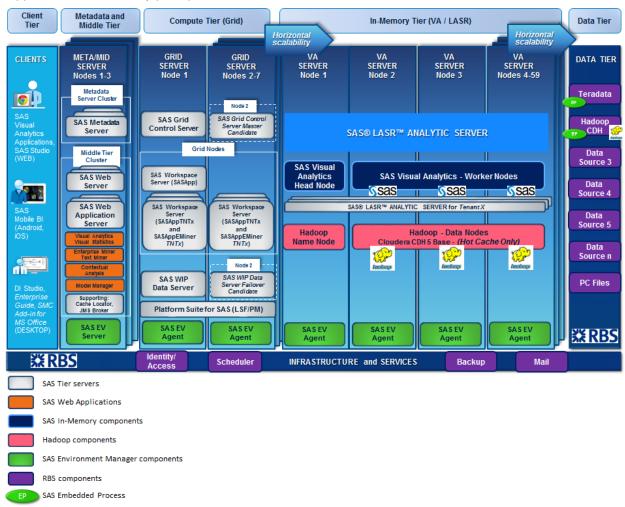


Diagram 4. Technical Architecture Overview

During the Technology architecture design phase we considered various approaches and made the decision to implement multi tenancy at the metadata level, using the security model as the basis for carving the platform into secure, customer orientated and managed areas. This approach allowed us to employ the power of SAS metadata controls while the underlying hardware is shared across all tenants and the way it is shared is absolutely transparent for our customers. This approach has its pros and cons that are described in Table 2 below.

Pros	Cons
Better hardware utilization. We do not limit execution of the customer's queries to the certain servers within the cluster.	In case of hardware failure it will affect all tenants by either reducing cluster capacity or resulting in the complete outage.
Better scalability and resilience. We can extend the existing platform by adding more nodes to the cluster and after they are configured no extra tenant specific configuration is required.	Tenants are competing for system resources. It can cause situations where one tenant's activity is affecting others.
Easy to Administrate. We keep all servers in the cluster identical, which allows us to implement a release management framework and reduce the number of manual administrative tasks.	

Table 2. Pros and Cons of a Metadata Security-based Multi-Tenancy Approach

One of the selling points for our business users was the fact that this platform will provide connectivity to almost all data sources available in the bank. Previously, data was hosted on the various unrelated platforms that made it isolated and unobtainable to a certain extent. With our new design we promote a "data-less" concept whereby only a very small subset of data can be stored in the staging area on the platforms storage. Extensive connectivity capabilities coupled with no dependency on other SAS platforms for obtaining data attracted the attention of our customers and also helped us to make progress towards consolidating data sources in support of the bank's enterprise data warehouse strategy.

In addition to this, previous experience of using clustered file systems showed us that it can cause headaches to the platform administrators. To address that, we built Genie with a "shared nothing" concept in mind. Almost all file systems are locally attached and aren't shared across the cluster, with the exception of the IBM Platform LSF file system and a few others that so require as per their failover and resiliency needs. By using this configuration we reduce complexity of the platform design and also improve scalability characteristics of the platform making it easier to provision additional servers.

When a platform serves a single business area, the impact of service failure (i.e. unplanned downtime) is isolated. However, when the platform serves the entire bank and is cross-divisional by nature, its criticality profile is raised. To-date, isolated implementations have been used to deliver SAS platforms. With Genie, we made a conscious effort to consolidate and rationalize the existing SAS estate, with the intention to reduce the maintenance and associated labor cost.

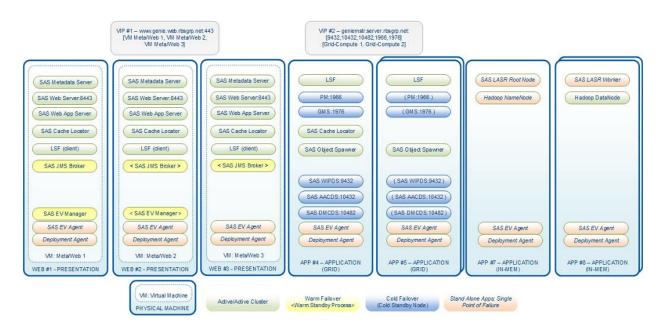


Diagram 5. Software Components Design

As shown on the Diagram 5 above, we made the decision to separate platform components to support scalability and improve platform administration processes. We have 3 virtual machines dedicated to the Metadata and Middle Tier clusters. This is a minimum requirement to ensure that we have quorum and benefit from the high-availability and resilience provided by the clustered Metadata Server. At the same time we "coupled" the Metadata Server cluster and Middle Tier one to ensure that we "sweat our assets". Historically at RBS we have had dedicated Metadata Servers and we have seen that these were significantly underutilized and were not cost effective.

The SAS Compute Tier consists of physical servers. Due to the system requirements we decided not to go for VMs for Compute Tier. Following a conventional configuration, we have a master node and a secondary node that is used to failover components like SAS® Grid Control Server and SAS® WIP Data Server. Enterprise Grid Orchestrator functionality is used to support failover for SAS Compute Tier components.

SAS[®] Visual Analytics capabilities are supported by the cluster of 50+ servers where SAS[®] LASR™ Analytic Server and HDFS platform components are deployed.

The Technical Architecture design that we chose enables horizontal scalability that we plan to use when the existing hardware reaches its capacity. The fact that we have the SAS Metadata, Compute, Middle Tier, and In-Memory components deployed across separate clusters helps us create tier-specific deployment plan files which reduce the overhead associated with adding new servers.

We paid special attention to resilience and scalability to ensure that we protect the service as much as possible and at the same time maintain the flexibility required to address capacity related issues and provide scalability when and if needed. We have opted for the following configuration:

Metadata and Middle-Tier (co-located)

- A cluster of SAS Metadata Servers comprising of the minimum of 3 members
- A cluster of SAS Web Application Servers (3, front-ended by a cluster of web servers)
- A cluster of SAS Web Servers (3, front-ended by a hardware load balancer e.g. F5 BIG-IP)
- A resilient configuration for Supporting Services such as SAS JMS Broker (master/slave) and SAS Cache Locator (active/active)
- Standalone SAS Environment Manager (EV) and Deployment agents per host

Compute Tier (Grid)

- A cluster of compute servers leveraging SAS[®] Grid Manager and Grid-launched Workspace Servers
- An Object Spawner per host, of which one assumes the master role and communicates with the rest
- A resilient configuration for Supporting Services such as SAS Web Infrastructure Platform Data Server, SAS Advanced Analytics Common Data Server and SAS Decision Manager Common Data Server, all under the control of IBM Platform LSF through an Active/Passive configuration achieved by IBM Platform EGO
- Standalone SAS Environment Manager (EV) and Deployment agents per host

In-Memory Tier (VA/VS)

- A Single Point of Failure on the LASR™ Root Node and LASR Worker Nodes (driven by limitations on the MPI communication implementation)
- A Single Point of Failure on the Hadoop NameNode
- A cluster of Hadoop DataNodes (resilient with 3 factor block replication)
- Standalone SAS Environment Manager (EV) and Deployment agents per host

Despite all the effort we had to leave a few Single Points of Failure (SPoF) that we either deemed insignificant or we plan to remove later in the platform's lifecycle as part of the service improvement. In Table 3 below you'll find detailed information about the remaining SPoF and potential impact this can have.

Single Point of Failure	Impact
SAS [®] LASR™ Root Node	SAS [®] Visual Analytics and Visual Statistics will be unavailable until the Root Node is restored. No failover to Standby Node exists and thus a hardware failure will require for the entire node to be replaced and LASR Root Node software to be re-deployed
SAS [®] LASR™ Worker	SAS [®] Visual Analytics and Visual Statistics will be unavailable. The In- Memory cluster will require all data to be re-loaded from source or its co- located data provider. Service may be restored on a subset of Worker Nodes (removing the problematic node and restarting the service in the remainder)
Hadoop NameNode	The hot-cache provided by the co-located data provider will be unavailable. The application will fail to load data from HDFS to memory, and similarly, will fail to save memory tables in HDFS
SAS [®] EV Agent	The impact is minimal. The usage and health metrics of the SAS components monitored by the SAS® Environment Manager will not be collected in the server the agent runs on. No failover to Standby Node is required for the EV Agents as the scope is solely on the server it is running on
SAS [®] Deployment Agent	The impact is minimal. The deployment agent is used when capturing and recovering from backups as well as deploying additional members in a cluster of Metadata or Middle Tier servers. Neither of these tasks are required for the operation during business hours

Table 3. Single Points of Failure in the platform

SECURITY MODEL

Security of any system is of paramount importance, and more so in an environment where multiple cross-divisional stakeholders are cohabiting on the same infrastructure. The SAS Metadata Server provides a robust set of authorization controls via User/Group membership and Access Control Templates. These controls can be further enhanced by securing access to specific rows of data via Row-Level Security.

When designing the security layer for Genie, the starting point was to categorize users depending on their usage-profile. For example, distinct permissions and capabilities are needed by information consumers or data modelers. The following categorization was agreed upon:

- Infrastructure Administrators
- Platform (Application) Administrators
- Data Administrators
- Content Creators
- Modelers
- Viewers
- Reviewers

The goal was for the membership to these user groups to drive the metadata objects they should have access to, and to grant the users with the application capabilities they were going to want to use.

A further categorization exercise took place. We classified user groups as either Report Channels or Data Channels, depending on whether the content created by these had a requirement for data segregation or

not. Report Channels are those business functions that are happy to coexist with others in their parent business function (the overarching tenant). Data Channels on the other hand require that their data is fully segregated, though still benefit from accessing the shareable content produced by peers within same tenant/boundary.

Segregated Reports (Report Channels)

Diagram 6 shows a high level representation of the security model at the metadata layer.

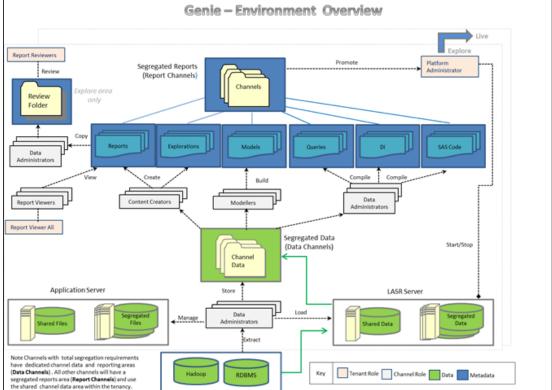


Diagram 6. Metadata Design for Genie

CUSTOM PROCESSES

On-boarding Automation

Quick time-to-market is one of the benefits typically associated with multi-tenancy. Removing the lead times involved in planning and procuring hardware can considerably reduce the time elapsed between a user or business area requesting access to a platform or service and the latter being available for use.

On-boarding a brand new set of users, including their requirement for a new application server context, into an existing platform which is operating in production mode introduces complexities of its own. Consider the impact that planned service downtime will have on existing tenants. Now consider the impact of manually creating the corresponding ACT, user and group memberships in Metadata and accidentally affecting the authorization model of another tenant; after all, manually creating ACTs, users and groups at scale is a laborious, even tedious and often an error prone task.

This is why at RBS we have looked to streamline the process by introducing an automated framework for on-boarding of individual users, but most importantly completely new tenants which requires limited user interaction.

The SAS® bulk-load (%MDU) auto-call macros are bundled as part of the SAS® Enterprise Business Intelligence platform and are very useful way to synchronize an organization's Identity and Access Management provider, such as an LDAP Server or Microsoft Active Directory, with SAS' own Metadata Repository.

By leveraging Active Directory as the single point of user administration and leveraging the bulk-load utilities to warrant the Joiners, Movers, Leavers process is adhered to, we are able to on-board individual users within a matter of minutes.

Diagram 7 describes the end-to-end process that is currently being built to take care of our user and tenant on-boarding needs. The process will be front-ended by a PHP web application which will request the tenant to enter the inputs required to request access to the platform. A workflow will determine if sufficient capacity exists and whether the tenants business and non-functional requirements can be met by the application. Once confirmed, the on-boarding process will be triggered which will create the appropriate metadata structures (folders, ACTs, group memberships) and other necessarily configurations such as password-less SSH, as relevant.

Development is in full swing as at time of writing and we expect to go live with the on-boarding automation process in summer of 2016.

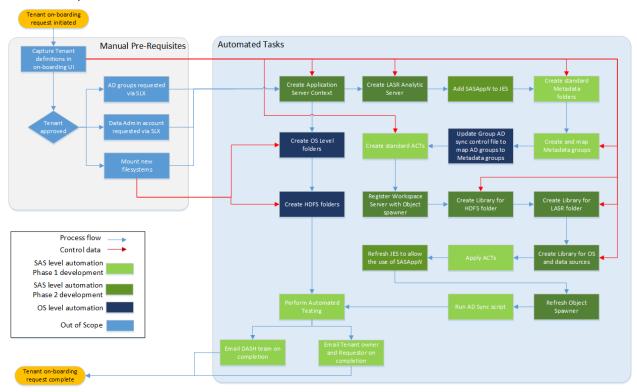


Diagram 7. End-to-End Tenant On-boarding Process

Platform Maintenance

Platform maintenance is vital to ensure that service that is delivered to our customers is stable, satisfies performance requirements and durable but at the same time flexible enough to allow for new components/capabilities to be implemented, updates and hotfixes applied as and when needed as well as to minimize downtime required to complete regular and ad hoc administrative tasks.

Sas.servers script improvement

While SAS® 9.4 delivers a lot of enhancements to the platform administrations tasks we still find that certain aspects need improvement and that is why we've developed a customized script that replaces the default sas.servers script. This script takes functionality available in the default script and extends it introducing the following additional features:

• When the status of SAS services is checked it not only queries the *pid* file but a "ps" command is executed to confirm that an active process with the corresponding PID exists.

- For certain SAS services extended checks are implemented. SAS Workspace Server and SAS
 Stored Process Servers are validated using the ValidateServer script to double check that not
 only the SAS service is up and running as an individual process on the OS level but that it is also
 responsive and can accept and process a request.
- SSH connectivity was built-in the script to allow platform administrators to execute the script once and
 run checks on all servers. Only SAS services deployed on the server will be checked, which makes
 the report generated by the script user-friendly and allows avoiding multiple failures due to certain
 components not being available on all servers across the environment.
- SAS Web Servers are also checked using wget. A special file is placed in the root folder that
 contains the word "Success"; the script then uses that file to validate that the SAS Web Server is
 responding.
- SAS Web Application Server and the web applications deployed are also checked. It is either done
 via wget and direct URL, like for the SAS WIP Clients Access application, or, for example, using a
 pre-defined STP that returns the word "Success".

The same way the default script has options to start, stop, restart and validate SAS services; the script that we've developed allows us to do the same but across all servers in the environment reducing E2E restart time by 60%. Also it helps us to reduce the overhead associated with manual checks that we conduct every morning across our Production platforms.

Command and control Portal

Together with the replacement for sas.servers script we are working on integration of SAS® Environment Manager with internal APEX based web application that is developed to produce MI information about:

- Number of users accessing the platform;
- 2. Duration of time each user spent connected;
- 3. Version of SAS software and hotfixes deployed on the platform;
- 4. System resource utilization;
- 5. Number of requests raised via the internal workload management tool and status for each request.

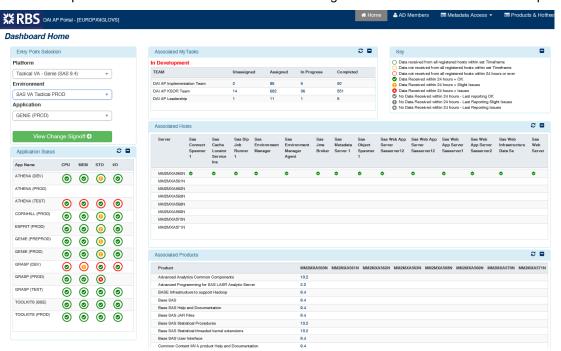


Diagram 8. Main Page of the Analytics Platform Portal

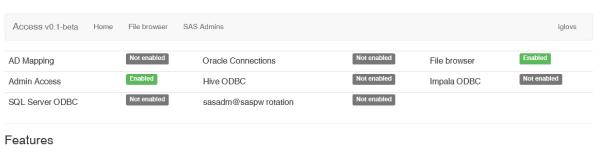
Privileged access management

Designing a platform as big and complex as this prompted us to reevaluate the way we manage membership of the SAS Administrators group. With a two-layer support model (i.e. application and infrastructure teams) and a security model design that allows tenants to be "landlords" to their respective boundaries, it became vital for us not to allow permanent elevated access that is associated with membership to the SAS Administrators group. At the same time there is no out-of-the-box functionality that allows platform administrators to control membership and move users in and out of that group without directly logging to the SAS Management Console and doing it manually.

According to the bank's security policy elevated privileged access cannot be granted permanently and has to be managed via a break glass process. We identified another constraint; the tools available in the bank do not integrate with SAS software and it is therefore impossible to control group membership using these.

These challenges triggered the development of an application called Access; a lightweight, PHP based web application that supports segregation of duties, allows platform administrators to request membership to the SAS Administrators group for a certain period of time and controls membership of SAS Administrators group by automatically removing accounts once the request period is over.

The Access application itself is secured; and thus, requests can only be submitted by authenticated users. Multiple additional checks are conducted during the request submission process before a request is sent for execution to prevent scenarios such as multiple overlapping requests, multiple requests by a single user or the use of an incorrect or past date-time value.



The following features are present in version 0.1-beta:

- View existing Active Directory to SAS Metadata Group Mappings
- Add new Active Directory to SAS Metadata Group Mappings
- View the Audit trail of who added which SAS Metadata Group to AD Group Mappings.
- Update the tnsnames.ora file for Oracle database connections
- Browse the UNIX file system
- View and tail files
- · Request to be a member of the SAS Administrator's group

Diagram 9. Access application Front Page

Diagram 9 shows the front page of the Access web application. As you can see not all features are enabled as yet, as we are still developing the utility, but Admin Access and File browser are enabled and available for use. The request handling process is fully automated and is completely hands-off.

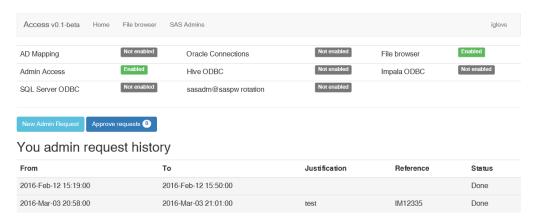


Diagram 10. Admin Access application Main Screen

While providing access to the SAS Administrators group covers the majority of the tasks that require elevated access, there are still a few that can't be completed without the use of the sasadm@saspw account password being exposed. The solution that we have in place at present rotates the password on a daily basis. Following discussions with the in-house engineering team we have recently decided to move away from the daily rotation and introduce a request based mechanism. This follows a similar pattern to the Admin Access tool described earlier, where the request for the sasadm@saspw password is managed via the Access web application. The development has completed and is currently on its test phase; once enabled an additional tab will appear in the main menu that will present users with the sasadm@saspw request main page. As this functionality is using the same mechanism as the SAS Administrators group membership request all interfaces have been developed identically to ensure a consistent user experience.

We faced a few challenges during the planning and development phases of the tool. One was the fact that the SAS Deployment Manager, which is used to update the password, has to be executed on all servers where the SAS Metadata Server is deployed. In the Genie platform, this means that SDM has to be executed 3 times and only if all executions are successful can the process move forward. In the event of failure of one or more executions the password has to be rolled back to the original state on the servers where it was successfully changed to ensure consistency across the SAS Metadata Server cluster. We solved this by developing the script so that it runs SDM on each server sequentially and proceeds to the next only if the previous execution was successful. This approach is a bit longer than running them in parallel but a safer and easier to manage option.

Another useful tool that was developed to improve platform maintenance is File browser. It provides readonly access to the OS level allowing the administrator to navigate the file and folder structures via browser window. This tool allows platform administrators to check configuration files, tail log files (with real time updates) and copy/paste information.

Software Currency

One of the key platform maintenance tasks that hasn't yet been implemented fully across our existing SAS estate is the deployment of hotfixes and maintenance releases. With a wide range of software components deployed across the estate, it is vital to ensure that we have a robust process that allows us to deploy hotfixes with minimal downtime. At the moment we have developed a tool that allows us to get up-to-date information about newly released hotfixes and maintenance releases and automatically raise JIRA tickets for each of these. The script is scheduled to run once a month and reads data available from the SAS9x_HFADD_data.xml file and cross-references it with a listing of already installed hot-fixes using information from the DeploymentRegistry.txt file.

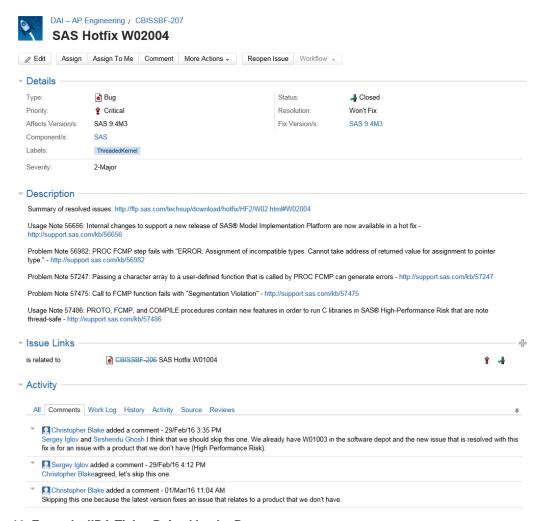


Diagram 11. Example JIRA Ticket Raised by the Process

This process is continuously being improved and the following functionality is lined up for future releases:

- Deployment of approved hotfixes will be performed using a script that will run once a week to coincide with the weekly application restart on Saturday evenings;
- SDM will be executed in quiet mode to install hotfixes across all servers in the environment;
- 3. Pre and post-installation steps will be scripted to ensure the hotfix deployment process is 100% hands off.

CONCLUSION

When we embarked on the journey of designing a bank-wide analytics platform, we were forced to think deeper around opportunities to save cost and increase efficiencies. Thus, a multi-tenant design, in our context, was a necessity, and not a luxury. We had to meet a variety of stakeholder requirements, within a constrained budget. We appreciate that other organizations might not have the same context, and that is why we are not prescriptive in our recommendations.

Through this paper, we have presented several considerations – technical, financial and operational that play a key part into selecting a design approach. Once we agreed our design approach, we had a clear vision of what we wanted to achieve. The next challenge was to create a software design that supports the vision. We are using the power of the SAS Metadata Server for our security design and general tenant segregation. We have explored various configuration options at the infrastructure level and we have also created our own in-built solutions to support our design. The role of our engineering, design

and service management teams in supporting this approach is also explored along with some insight into how we convinced our stakeholder community about the benefits of our design.

At the end of our current design phase, we are still acknowledging residual risks to our design approach. Some core components, such as the LASR™ Analytic Server, continue to be single points of failure, limiting configurations that support a high available deployment. Similarly, software that defaults to a single Application Server Context (e.g. SASApp), challenges the security model required to support a multi-tenancy configuration.

This paper discusses a proposed multi-tenant architecture for the BI and analyst community at RBS and the processes that surround its implementation. It is not a trivial challenge and whether it is worth fighting for or not, will depend on the organizational context. Our aim is to encourage other organizations to explore the concept, using our experience and findings as a guidance. We would also like SAS Research & Development to work towards more out-of-the-box features to support multi-tenancy and remove single points of failures in the technical architecture.

ACKNOWLEDGMENTS

The authors would like to thank Andrew Williams, Sirshendu Ghosh, Greg Cox, Jonathan Lloyd, Paul Johnson and the wider Data & Analytics Infrastructure team who contributed to the design and implementation of the above mentioned platform and processes.

RECOMMENDED READING

- SAS® 9.4 Intelligence Platform: System Administration Guide, Fourth Edition
- SAS® 9.4 Intelligence Platform: Security Administration Guide, Second Edition
- SAS® Environment Manager 2.5 Administration: User's Guide

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Christopher Blake 2 ½ Devonshire Sq. London, EC2M 4BA United Kingdom christopher.blake@rbs.com

Sergey Iglov 2 ½ Devonshire Sq. London, EC2M 4BA United Kingdom sergey.iglov@rbs.co.uk

Dileep Pournami
1st Floor Business House E
Gogarburn
Edinburgh, EH12 1HQ
United Kingdom
dileep.pournami@rbs.co.uk

Ekaitz Goienola SAS UK&I Professional Services Wittington House, Henley Road Marlow, Buckinghamshire, SL7 2EB United Kingdom ekaitz.goienola@sas.com SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.