

Serving SAS®: A Visual Guide to SAS Servers

PA-01

Gregory S. Nelson

President and CEO

ThotWave Technologies, Cary, North Carolina

ABSTRACT	1
INTRODUCTION.....	2
FOUNDATIONS VERSUS METADATA MANAGED IMPLEMENTATIONS.....	2
OPERATING SYSTEM AND SECURITY IMPLICATIONS.....	3
SAMPLE ARCHITECTURES	4
SAS FOUNDATION	5
SAS METADATA MANAGED ARCHITECTURES.....	7
<i>SAS Enterprise Data Integration</i>	<i>8</i>
<i>SAS Enterprise Business Intelligence</i>	<i>11</i>
<i>SAS Grid Manager</i>	<i>16</i>
SUMMARY	21
REFERENCES AND RECOMMENDED READING	21
BIOGRAPHY	21
CONTACT INFORMATION:.....	22

Abstract

SAS® has been running on servers since the late 1960s. Despite the emergence of PCs and workstation-class machines, SAS still reigns supreme on the server. With the introduction of the SAS platform, the number and types of servers have grown exponentially.

As any good student of the DATA step will attest, knowing what SAS is doing is a critically important step in debugging and authoring efficient programs.

In this paper, we provide a visual tour of SAS. We cover what SAS is doing, how it works, which server is doing what, when the operating system plays a role, how security functions, and what happens to your data through this entire process.

Introduction

Beginning in the late 1960's "S.A.S." started out as the "Statistical Software System" – a project incubated at North Carolina State University to help agricultural research stations throughout the South East in the analysis of data. Fast forward another three decades and we find SAS (sans-acronym) and over 50,000 sites worldwide and matured well beyond its research roots. Today, SAS is used in across virtually every industry to solve enterprise class problems for companies large and small.

In the mid-pre-teens (2004), SAS 9 was introduced and SAS moved beyond its foundational roots to a metadata-managed, enterprise class, analytics software. Along with this maturation, SAS also introduce us to a number of "servers" that heretofore were unknown to the tens of thousands of SAS-enthusiasts.

As a practitioner specializing in the implementation of SAS solutions, and, in particular, helping people understand how SAS works, we spend a lot of time ensuring that SAS can be properly used, managed and tuned. As anyone close to me knows, it is hard to keep me away from a white board. I can't tell you how many times I've jumped up to the white board to explain how SAS works – that is, what happens when the user clicks something and then to explain what the server is doing in response and where data goes and how security and the operating system comes into play.

This paper is my attempt to give up on my dreams of becoming an artist and to finally create (and share) a set of visualizations that help both users and administrators understand how SAS works.

Foundations versus Metadata Managed Implementations

Over the past seven years, SAS has introduced us to some new terminology. It used to be that we had SAS modules such as Base, STAT, GRAPH and so on (of course, we are taking liberties with what SAS actually calls them – for example, SAS/STAT). While we still have those "modules", we now refer to the core of what SAS does with all of these products as "SAS Foundation". So as SAS (the company) has built on solutions such as SAS Enterprise Business Intelligence, at the core, SAS Foundation(al) products are still there doing the actual work. The biggest difference now is that SAS Foundation is being called by a SAS "Server" (such as the SAS Workspace Server or Stored Process Server), which in turn is being called by the Object Spawner, which was spawned in response to a request from any number of clients, which was authenticated by the operating system which was authorized by the metadata server (and that it is a simple scenario!) *Confused yet?*

Now you see why I wanted to create a library of visualization for how SAS works – so that us visual learners would have half a chance to understand what was really going on underneath the covers.

In a nutshell, SAS can be distinguished by whether or not it is a “foundation” install versus a metadata-managed implementation. The former being just a standalone SAS environment which many of us group up knowing. The latter, introduced with version 9, is what gives us the power of SAS for large, multi-user, enterprise class capabilities. The SAS 9 architecture relies on a new component, the SAS Metadata Server, to provide an information layer between the programs and the data they access. By providing a single point of access for this kind of information, SAS servers can be located just about anywhere (consolidated or distributed) and on any platform (operating system and hardware), to be accessed by SAS clients.

The SAS 9 platform can be configured in any number of ways, including a myriad of server topologies and client offerings. In these architectures, the SAS client tools take on the form of special purpose applications designed to satisfy the needs of various types of SAS users. Tools used to access information now include a substantially improved Enterprise Guide, the SAS Add-in for Microsoft Office, and the web-based clients Information Delivery Portal and Web Report Studio, as well as Base SAS components (which we now call SAS Foundation).

Management of SAS metadata is done through specialized clients such as the SAS Management Console. Management of data (and metadata) is facilitated through the use of tools such as Data Integration Studio, OLAP Cube Studio and Information Map Studio (on top of existing SAS products such as SAS/Access and Base SAS.)

So while there is no normal, there are some common implementations of SAS that are worth noting. In the section entitled “Sample Architectures” we will outline some of these in both words and pictures.

Operating System and Security Implications

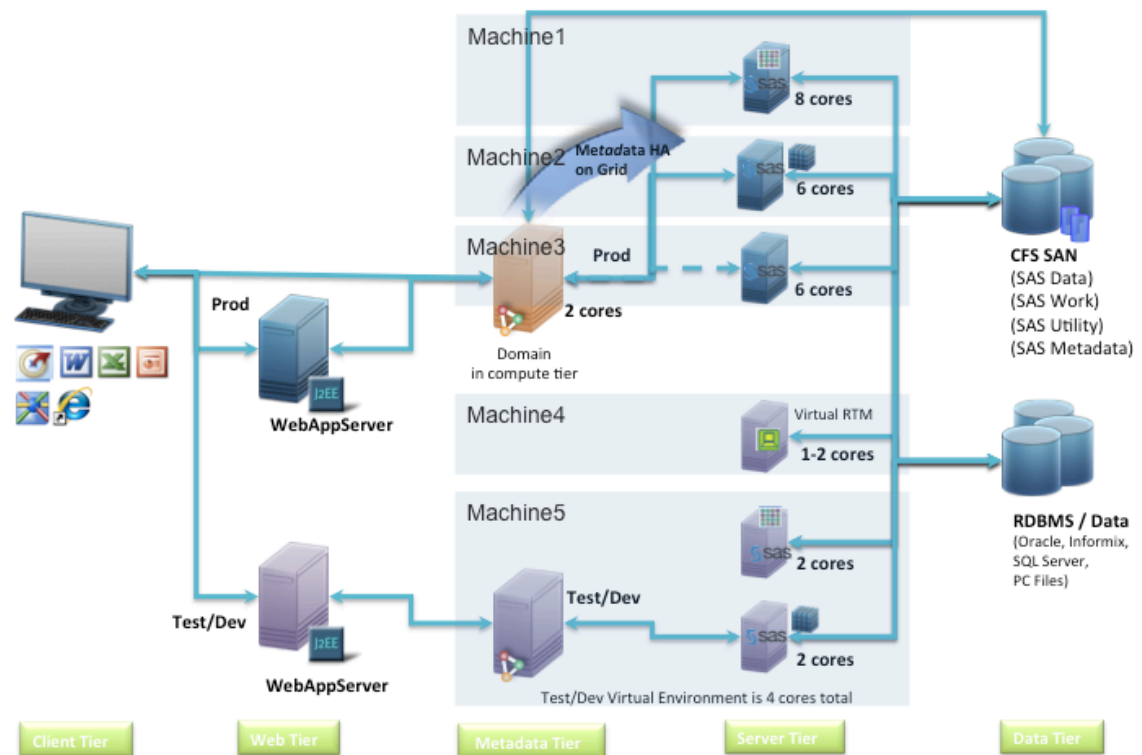
SAS currently runs on a variety of UNIX, Windows and z/OS environments (see [Supported Operating Environments for UNIX](#), [Supported Operating Environments for Windows](#) and [Supported Operating Environments for z/OS](#)). While we won't spend a lot of time discussing implications of the various operating systems, we will highlight those interactions where critical to understanding. The types of interactions that you might see include the role of the operating system in the following areas:

- Authentication provider
- File system
- Schedulers
- File versioning

For a more detailed discussion of how SAS interfaces with the operating environment, please refer to Nelson and Loether, 2007 or Nelson and Swirski, 1997.

Sample Architectures

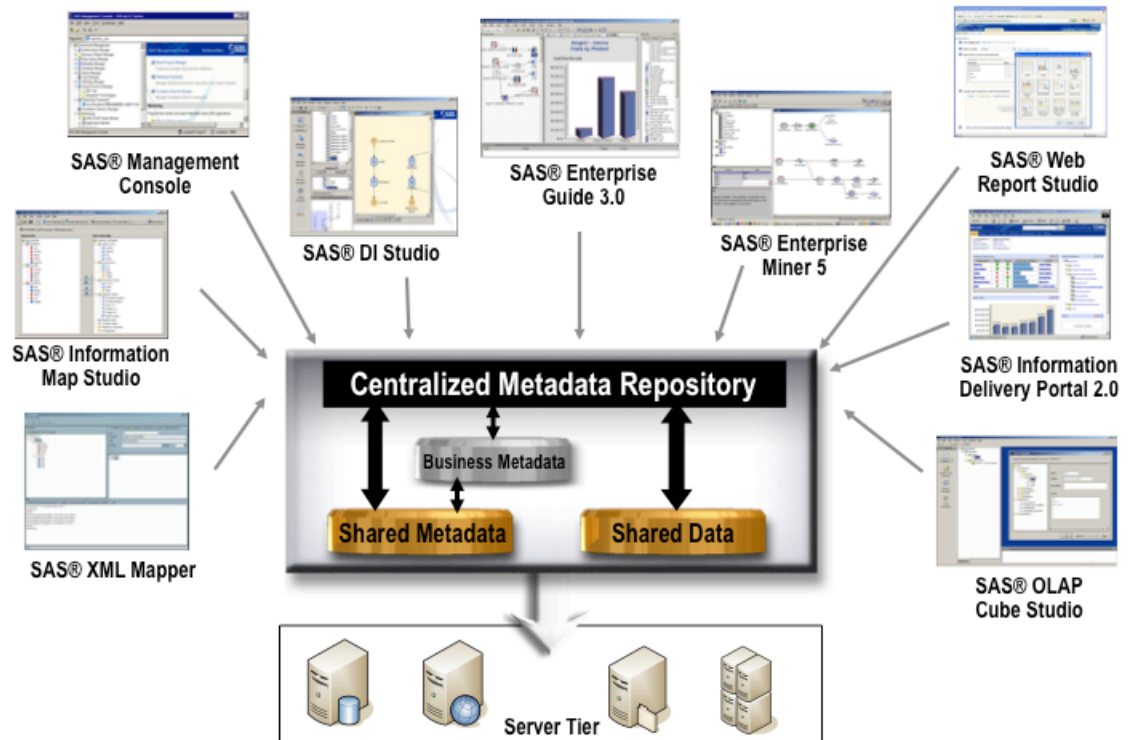
Each implementation of SAS 9 can be fundamentally the same and different depending on the profile of the business problem being solved. These profiles make use of the new services in SAS 9 in varying degrees. The figure below depicts one such topology for an enterprise configuration of SAS 9 using SAS Grid Manager.



In this diagram, typical SAS development relies on a client such as Enterprise Guide talking to a server-based instance of SAS. This model supports the concept of centralized server resources.

The SAS Institute has made a large investment in these new products and technologies, and these are the future of SAS for some time to come. It must be recognized that some of the new products, such as Information Map Studio, Web Report Studio, and the OLAP server, are aimed at solving general Business Intelligence problem areas. The figure below shows some of the clients and how they interface with SAS servers.

SAS 9 Clients

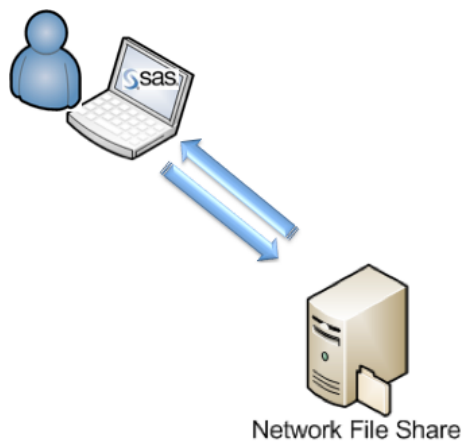


So let us turn our attention from the ethereal to the real – let’s visualize how these work. In the following pages, we will outline four basic scenarios and turn our words into pictures that demonstrate just how this all works.

SAS Foundation

While perhaps the easiest to understand, SAS Foundation-based architectures can indeed be enterprise class. In fact, for some companies that began using SAS 20-30 years ago, continue to use SAS today – and in some cases, use SAS Foundation on large mainframe, UNIX and Windows servers.

So let’s see our first example of a simple SAS Foundation architecture.



or

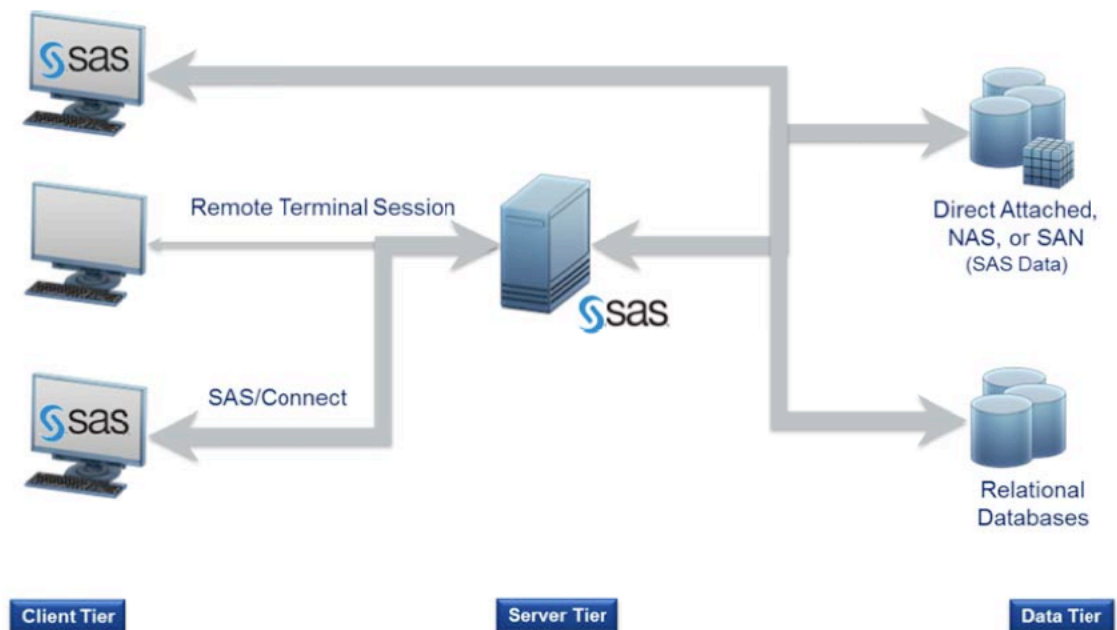


Figure 1. Traditional SAS Deployment Methods with Data Movement Across the Enterprise

<build with the simple case (top), then Remote Terminal (middle) and then SAS/Connect>

Here, we have a personal computer with SAS installed locally reading and writing files to disk (either locally or direct attached or a network attached file server such as SAN or NAS.) In fact, in the simplest form of a SAS Foundation install on a single computer. The user writes code, submits it, SAS processes the request, reads/ writes data and results are generated – all within the context of the computer where it originated. SAS can read and write to files on behalf of the users (as the user) so security is simple – if you can read or write to a file using the operating system, you can do so from SAS.

Now replace the laptop with a mainframe, UNIX server or Windows server – all are examples of SAS Foundation at work. Extending this notion, we can move the data to a file server or a database (Figure x), other data shared through SAS servers such as SPDS or SAS/Share (Figure x), or external data sources such as databases or non-SAS data formats (Figure x).

Modern approaches to implementing SAS Foundation include having users connect to servers through technologies such as Remote Desktop, Citrix XenApp/ XenDesktop or Microsoft AppV (Figure x). In these cases SAS may be running in a dedicated or virtualized environment, but don't be fooled – where SAS is installed on the same computer where the user interacts with it, then it is likely that you are dealing with a SAS Foundation architecture.

In the case of SAS Foundation, we did not require a metadata server, just access to the SAS executable. Some people might use SAS in batch (command line) or interactively (such as the SAS Display Manager System), but we are still submitting programs to SAS for execution.

SAS Metadata Managed Architectures

The SAS 9 architecture is fundamentally different from any prior version of SAS. In the SAS 9 architecture, SAS relies on a new component, the Metadata Server, to provide an information layer between the programs and the data they access. Metadata, such as security permissions for SAS libraries and where the various SAS servers are running, are maintained in a common repository.

By providing a single point of access for this kind of information, SAS servers can be located practically anywhere (consolidated or distributed) and on any platform (operating system and hardware), to be accessed by SAS clients. The SAS client tools now take on the form of special purpose applications designed to satisfy the needs of various types of SAS users. The SAS 9 platform can be configured in any number of ways, including a myriad of server topologies and client offerings. For example, management of SAS metadata can be done through the SAS Management Console and Information Map Studio. Or management of data can be done through Data Integration Studio in addition to the existing SAS products, such as SAS/Access and Base SAS. Tools used to access information now include a substantially improved Enterprise Guide, the new add-in for Microsoft Office, and the web-based clients (Information Delivery Portal and Web Report Studio), as well as Base SAS components.



In the pages that follow, we will go through some common SAS 9 architectures and see how things work.

SAS ENTERPRISE DATA INTEGRATION

Remember before we talked about the Workspace Server – it is simply a server that takes requests and turns them into something that SAS Foundation can process. Now that we have learned about SAS Foundation, we'll use that knowledge to extend our architecture to include SAS technologies, which help us manage data warehouses – and in particular, the ETL process (extract-transform-load).

In the SAS Enterprise Data Integration implementation¹, we see some new components:

- Clients
 - SAS Data Integration Studio
 - SAS Management Console
 - Data Management Studio (Data Flux)
- Servers
 - SAS Data Integration Server
 - SAS Metadata Server
 - SAS Workspace Server

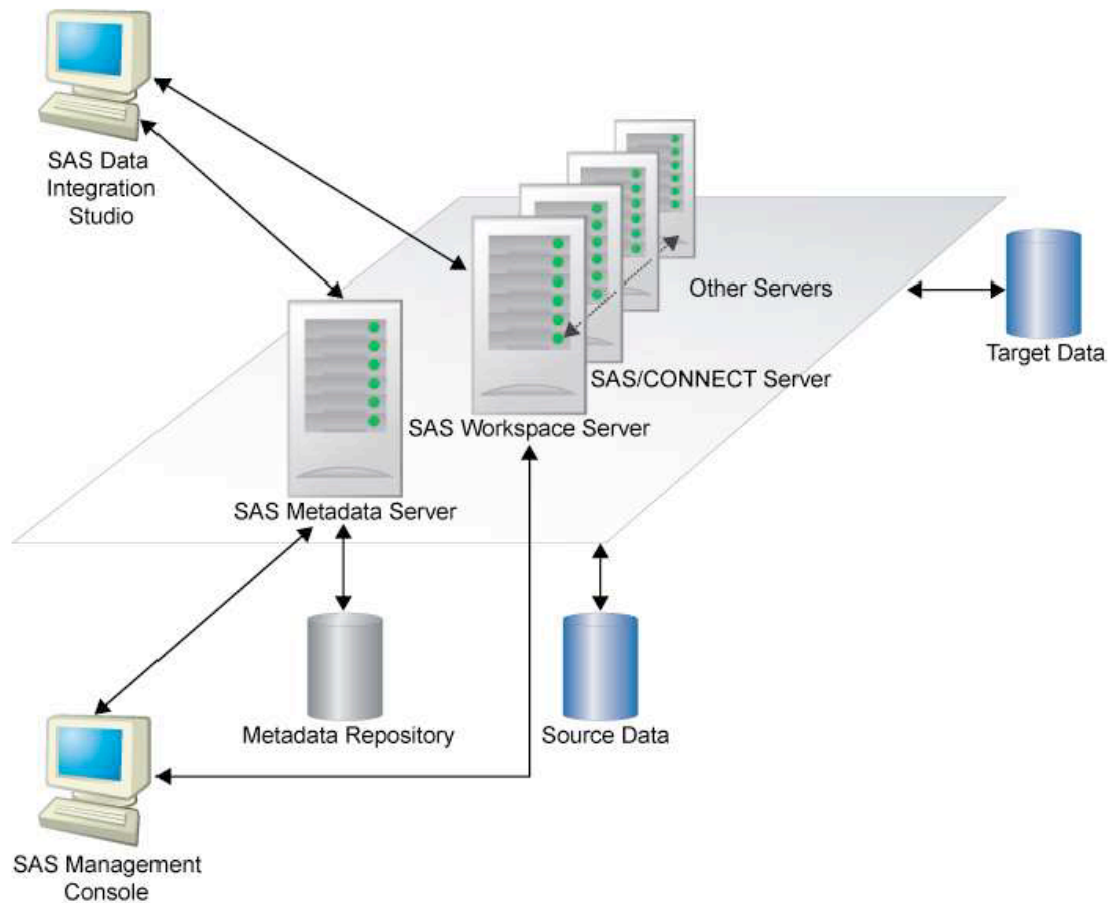
¹ Note: "typical implementation" doesn't mean that your installation will necessarily include all of these nor is this list comprehensive

- SAS Integration Technologies
- SAS/SHARE
- SAS/CONNECT
 - Platform Computing Suite for SAS (Platform Computing)
 - SAS Data Quality Server

Instead of talking through how this all works, let's show you! Here we will explore a simple use case where someone is using SAS Data Integration Studio for developing an ETL process.

<TODO visualizations>

Figure 1.1 SAS Data Integration Studio Environment



Step #	What the User Sees	What SAS Does
❶	User launches SAS DI Studio. DIS prompts for connection information (user, password, connection details for SAS metadata server)	<p>Metadata Server:</p> <ol style="list-style-type: none"> 1. Verifies user credentials via the authentication mechanism. Authentication mechanism² can include: <ol style="list-style-type: none"> a. Host authentication b. LDAP or Active Directory c. Integrated Windows Authentication (AWA) 2. Identifies the user in the SAS metadata repository 3. Resolves assignment of SAS metadata roles and access controls 4. Connection is established between client application and the metadata server
❷	Users creates a new process flow and adds which includes table metadata	Metadata Server receives request for table metadata, ensures that the user has the correct access permissions and delivers the metadata back to the client application.
❸	User submits a job for immediate execution ³	<ol style="list-style-type: none"> 1. The metadata server authorizes the user to create a new workspace server connection 2. The job is submitted to the SAS Application Server (and to any server that is specified in the metadata for a transformation within the job) 3. The object spawner creates a new workspace server session on behalf of the user 4. SAS Workspace Servers interact with SAS by creating a server process for each client connection. The workspace server process is owned by the client user who made the server request.

² Note: Web authentication is a fourth option for authentication, but since we are talking about DI Studio, it is not relevant here.

³ Note: There are a number of additional options for running a job not described in detail here. These including: deploy the job for scheduling, deploy the job as a SAS stored process or deploy a stored process as a Web service.

		5. Each workspace server process enables client programs to access SAS libraries, perform tasks by using the SAS language, and retrieve the results.
④	User closes/ quits DI Studio	1. The workspace session that was opened on behalf of the user is shut down when their clients have completed their work.

SAS ENTERPRISE BUSINESS INTELLIGENCE

SAS Enterprise Data Integration discussed above is most often used for what we call “beating data into submissions”. Once sufficiently beaten (prepared for use), we can now turn our attention to report development and consumption. Often, programmers will use interactive development tools such as Enterprise Guide to aid in the generation of results and then utilize SAS stored processes or web reports to allow others to consume the data in useful ways.

So let us now turn our attention to two of the most common SAS clients: SAS Enterprise Guide and Web Report Studio.

Interactive Programming (with Enterprise Guide)

SAS Enterprise Guide is a Windows application that allows either traditional programming or a point-and-click interface that guides users through the analytical process, making it easy to create reports, graphs and charts. SAS Enterprise Guide is very similar to SAS Data Integration Studio in that it is a client application that is authorized by the SAS Metadata Server and then creates a workspace session for the submission of SAS code.

Visually, this look like:

<insert figure – see slide 32>

Step #	What the User Sees	What SAS Does
❶	User launches SAS Enterprise Guide. EG prompts for connection information (user, password, connection details for SAS metadata server)	<p>Metadata Server:</p> <ol style="list-style-type: none"> 1. Verifies user credentials via the authentication mechanism. 2. Identifies the user in the SAS metadata repository 3. Resolves assignment of SAS metadata roles and access controls 4. Connection is established between client application and the metadata server
❷	Users expands the serve “SASApp” to view libraries and files	<ol style="list-style-type: none"> 1. The metadata server authorizes the user to create a new workspace server connection 2. The object spawner creates a new workspace server session on behalf of the user 3. SAS Workspace Servers interact with SAS by creating a server process for each client connection. The workspace server process is owned by the client user who made the server request
❸	User submits a program execution	<ol style="list-style-type: none"> 1. The workspace session that is assigned to the user session processes the program(s) and accesses SAS libraries, perform tasks by using the SAS language, and retrieve the results 2. The results are streamed back to the client application and the workspace server sits idle until the user makes another request or closes the application
❹	User closes/ quits SAS Enterprise Guide	<ol style="list-style-type: none"> 2. The workspace session that was opened on behalf of the user is shut down when their clients have completed their work

Report Viewing

Within the SAS Enterprise Business Intelligence solution, there are a number of SAS clients which make it easy to consume data via dashboards, OLAP cubes, or static and

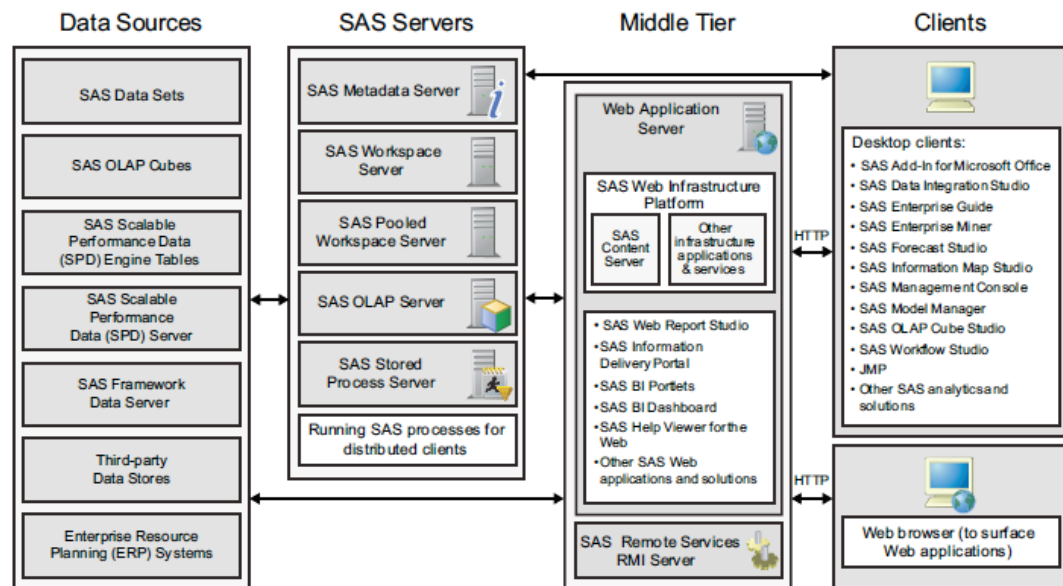
dynamic reports through a web interfaces, Microsoft Office applications or specialty applications. These clients allow end users to access data and SAS compute power through easy-to-use interfaces. One of the most popular web-based applications is SAS Web Report Studio.

Web Report Studio is different from SAS Enterprise Guide and SAS Data Integration Studio in that there is a middle-tier between the clients and the servers. This middle tier sites within a web application server that handles the requests for the user – think of the middle tier as a middleman that handles the communications between the client and the server. In the former case, SAS Enterprise Guide and Data Integration Studio were able to interact directly with the SAS workspace server once created.

So let's take a look:

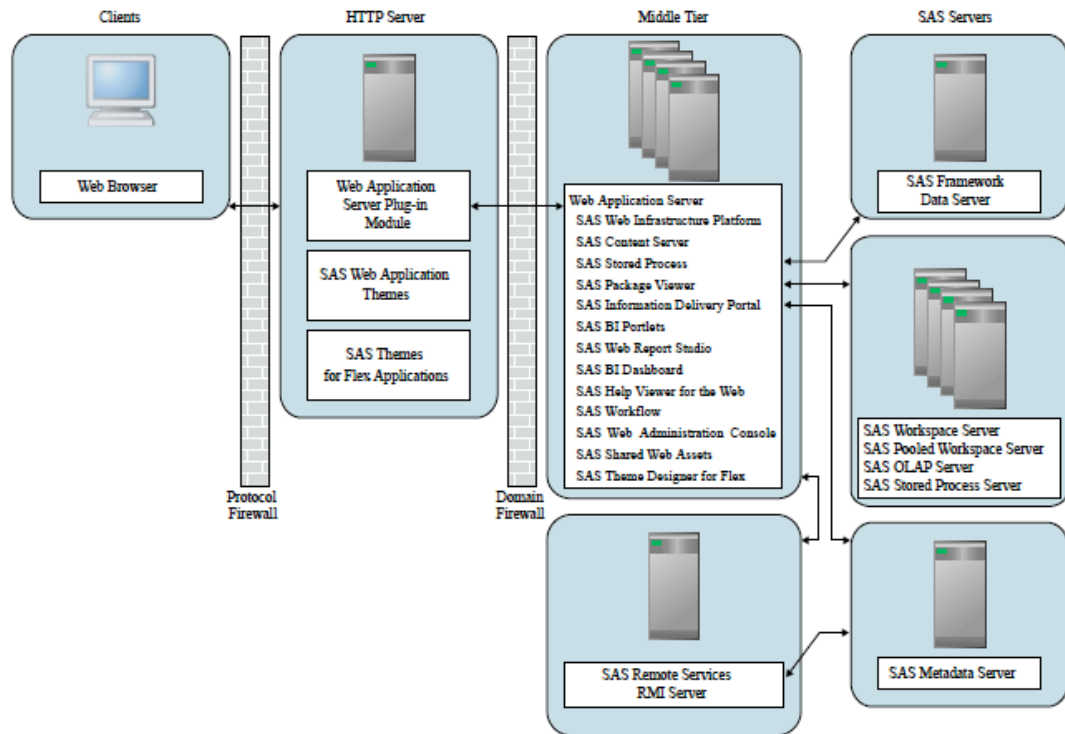
<insert figure>

Figure 2.1 Architecture of the SAS Intelligence Platform



OR

Figure 2.3 Scenario 3: Clustered Web Application Servers and a Demilitarized Zone



Step #	What the User Sees	What SAS Does
1	<p>User launches SAS Web Report Studio (page request).</p> <p>WRS prompts the user for the username and password.</p>	<ol style="list-style-type: none"> 1. A SAS component within the Web application server (the Logon Manager application) prompts the user for credentials. 2. The Web container's Java Authentication and Authorization Service (JAAS) login module coordinates the verification of the credentials 3. Verifies user credentials via authentication provider. 4. JAAS then passes the authenticated ID to the SAS trusted login module 5. A trusted connection is established to the SAS Metadata Server (SAS Authentication Service → SAS Remote Services → SAS Metadata Server) 6. SAS Metadata Server identifies the user in the SAS metadata repository 7. Resolves assignment of SAS metadata roles and access controls 8. Connection is established between middle tier application and the metadata server
2	Users requests a web report	<ol style="list-style-type: none"> 1. The request goes from the browser back to SAS Web Report Studio on the middle tier. SAS Web Report Studio must obtain both the structure for the report (how it will look) and the data. <ol style="list-style-type: none"> a. The structure for the report is stored in the SAS Content Server. SAS Web Report Studio sends a query to the SAS Metadata Server asking where the SAS Content Server resides and queries the SAS Metadata Server to make sure that your SAS ID is authorized to view this report and use the data sources that are associated with it.

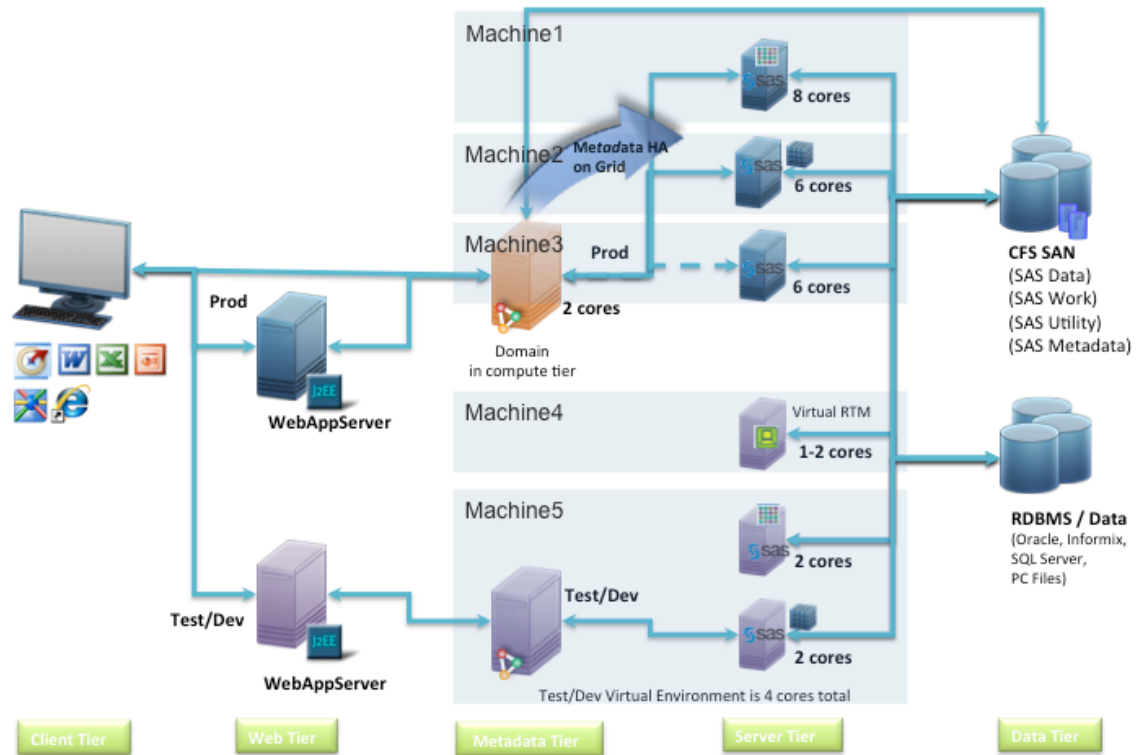
		<ol style="list-style-type: none"> b. WRS sends a to the SAS Content Server to retrieve the stored XML that describes the structure of the report. c. A request is sent to the SAS Metadata Server for information about where the data resides, the information map that is required to decode the query, and whether you have the authority to read that data. <ol style="list-style-type: none"> 2. Once SAS Web Report Studio receives the required information, it asks the SAS object spawner to give it access to a SAS Workspace Server or SAS Stored Process Server. SAS Web Report Studio sends the query to the SAS session that it obtained. 3. The SAS server then uses a configured data source engine, such as SAS/ACCESS, to contact the database, send the query, and receive the results. 4. The ResultSet is rephrased into the business language that is used by the information map and is sent back to SAS Web Report Studio, which sends the report to your browser.
③	User closes/ quits Web Report Studio	<ol style="list-style-type: none"> 1. The workspace session that was opened on behalf of the user is shut down when their clients have completed their work.

SAS GRID MANAGER

In the examples above, you may have noticed that each time a client connects to the servers, it eventually creates a connection to a workspace server. Once established, that workspace server essentially does all of the work for the client until the client closes their session and the workspace server is closed.

An alternative to that approach is to use SAS Grid Manager to manage the workload and determine where the best place to run the job. The benefits of this approach are numerous and include the ability to prioritize jobs based on the type of job or user (group) submitting the job and the workload can be balanced across servers based on actual load on the servers.

Here is a diagram of how this might be architected.



Let's go through an example of how SAS Grid might be used in both an interactive programming environment as well as via command line (for those coming from a UNIX environment).

Enterprise Guide and Grid

As we saw above, Enterprise Guide can be used to access the full power of SAS running on a server. In the case of a grid architecture, replace a single server with any number of servers that can carry out the work.

<insert figure –see slide 33>

Step #	What the User Sees	What SAS Does
❶	User launches SAS Enterprise Guide. EG prompts for connection information (user, password, connection details for SAS metadata server)	<p>Metadata Server:</p> <ol style="list-style-type: none"> 1. Verifies user credentials via the authentication mechanism. 2. Identifies the user in the SAS metadata repository 3. Resolves assignment of SAS metadata roles and access controls 4. Connection is established between client application and the metadata server
❷	Users expands the serve “SASApp” to view libraries and files	<ol style="list-style-type: none"> 1. The metadata server authorizes the user to create a new workspace server connection. If the workspace servers are load balanced, then the metadata server either uses a “cost” algorithm to determine where to create the workspace session or the grid algorithm. The grid algorithm uses Platform’s LSF (Load Sharing Facility) to determine which server is the best candidate for the new session 2. The object spawner creates a new workspace server session on behalf of the user 3. For the duration of the Enterprise Guide Session, the workspace server interact with client and manages the requests for the user
❸	User submits a program execution	<ol style="list-style-type: none"> 1. LSF (aka the Grid Manager) assigns the job a unique job ID & sends to appropriate queue 2. The job is evaluated and scheduled based on job attributes 3. Unlike what we saw earlier, the workspace session that is assigned to the user session doesn’t actually process the program(s) but creates a connection to a SAS session and accesses SAS libraries, perform tasks by using the SAS language, and retrieve the results – in this case the following occurs:

		<ol style="list-style-type: none"> a. A host selected for job to run b. A grid session is created on the host c. The job is dispatched to host(s) and executed d. When complete, the job is assigned a STATUS code e. The results are returned to the workspace session (where EG is connected to) f. Optionally, the grid session closes or remains open depending on how the environment is configured <ol style="list-style-type: none"> 4. The results are streamed back to the client application and the workspace server sits idle until the user makes another request or closes the application
4	User closes/ quits SAS Enterprise Guide	<ol style="list-style-type: none"> 1. The workspace session that was opened on behalf of the user is shut down when their clients have completed their work. (and optionally, the grid session closes)

Batch Execution of Grid Jobs

An alternative to using an interactive application like SAS Enterprise guide might be to use the SAS Grid Manager Client Submission Utility – or sasgsub.

SAS Grid Manager Client Utility is a command-line utility that enables users to submit SAS programs to a grid for processing. This utility allows a grid client to submit SAS programs to a grid without having SAS installed on the machine performing the submission. It also enables jobs to be processed on the grid without requiring that the client remain active.

<insert figure to similar to above except gsub client rather than EG>

Practically, the difference between using a client such as Enterprise Guide and sasgsub are visual in nature – that is, the user submits jobs via a command line rather than a graphical user interface. This is particularly useful if you want to submit and forget, check on the status of a long-running job or even stop a job that has been submitted to the grid.

Step #	What the User Sees	What SAS Does
❶	<p>User submits a command to tell sasgub to run a program.</p> <p>The user must provide details including the name of the SAS program, connection information to the SAS Metadata Server (server, port, username/password, etc.)</p>	<ol style="list-style-type: none"> 1. Verifies user credentials via the authentication mechanism. 2. Identifies the user in the SAS metadata repository 3. Resolves assignment of SAS metadata roles and access controls 4. The metadata server authorizes the user to utilize the application server context 5. LSF (aka the Grid Manager) assigns the job a unique job ID & sends to appropriate queue 6. The job is evaluated and scheduled based on job attributes 7. A host selected for job to run 8. A grid session is created on the host 9. The job is dispatched to host(s) and executed 10. When complete, the job is assigned a STATUS code 11. The results are returned to the GRIDWORK directory 12. The grid session closes
❷	User executes a “getresults” command via sasgsub	<ol style="list-style-type: none"> 1. Steps 1-4 above are executed 2. LSF retrieves the location of the job in GRIDWORK 3. The results are moved from the GRIDWORK directory to the requested location
❸	View job status	<ol style="list-style-type: none"> 1. Steps 1-4 above are executed 2. LSF retrieves the status of the job from LSF 3. The status is returned to the user’s console
❹	Stop/ terminate a job	<ol style="list-style-type: none"> 1. Steps 1-4 above are executed 2. LSF retrieves status of the job from LSF, if in process, LSF issues a termination command to the

		SAS grid session
		3. SAS terminates the running job and communicates back to LSF

Summary

SAS is a feature rich application. As architectures have evolved, a corresponding increase in complexity has also occurred. Knowing where things are happening is the first step in really understanding the modern, metadata-managed SAS environment. We have described some of the more common implementations of SAS and hopefully this has given you a better understanding of SAS and how it works. As this paper was intended to be a “visual” guide to SAS servers, we encourage you to view the visual treatment of this content at www.ThotWave.com.

References and Recommended Reading

Fenton, Ladd, and Spakes, “A Practical Approach to Re-Architecting a SAS® Deployment”, Invited paper presented at the SAS Global Forum, 2011.

Nelson, G. and Loether, J. 2007 “Architecting SAS in a Modern World: Best Practices for Design, Configuration and Management of SAS® 9”. Invited paper presented at the SAS Global Forum. Orlando, Fl. April, 2007.

Nelson and Swirski. “Choosing the Right Platform for SAS® Client/ Server Applications: An evaluation of factors in selecting NT and UNIX” ” Invited paper presented at the SAS Users Group International. Nashville, TN. March, 1998.

Schneider, Bennett, and Robison, “Understanding the Anatomy of a SAS® Deployment: What's in My Server Soup?” Invited paper presented at the SAS Global Forum, 2011.

Talley and Kalich “Troubleshooting Tips and Techniques for a SAS 9.2 Business Intelligence Environment”, Invited paper presented at the SAS Global Forum, 2011.

Biography

Greg Nelson, President and CEO of ThotWave Technologies, LLC.

Greg is a certified practitioner with over two decades of broad Business Intelligence and Analytics experience. This has been gained across several life sciences and global organizations as well as government and academic settings. He has extensive software development life cycle experience and knowledge of informatics and regulatory requirements and has been responsible for the delivery of numerous projects in private and commercial

environments. Greg's passion begins and ends with helping organizations create *thinking data*® – data which is more predictive, more accessible, more useable and more coherent.

His current area of interest is helping companies take advantage of the shifting world of convergence around data and systems and how modernization and interoperability will change the way that we discover new relationships, manage change and use data and analytics to improve organizational outcomes.

Mr. Nelson has published and presented over a 150 professional papers in the United States and Europe. Mr. Nelson holds a B.A. in Psychology and PhD level work in Social Psychology and Quantitative Methods and certifications in project management, Six Sigma, balanced scorecard and healthcare IT.

Greg can be reached at greg@thotwave.com or www.linkedin.com/in/thotwave

About ThotWave

ThotWave Technologies, LLC is a Cary, NC-based consultancy and a market leader in real-time decision support, specializing in regulated industries, such as life sciences, energy and financial services. ThotWave works at the juncture of business and technology to help companies improve their operational and strategic performance and recognizes the difference between simply accessing data and making data work for business. Through products, partnerships and services, ThotWave enables businesses to leverage data for faster, more intelligent decision making.

Contact information:

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

Greg Nelson greg@thotwave.com

ThotWave Technologies, LLC

Chapel Hill, NC 27517 (800) 584 2819

<http://www.thotwave.com>

thinking data® is registered trademark of ThotWave Technologies, LLC.

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

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.