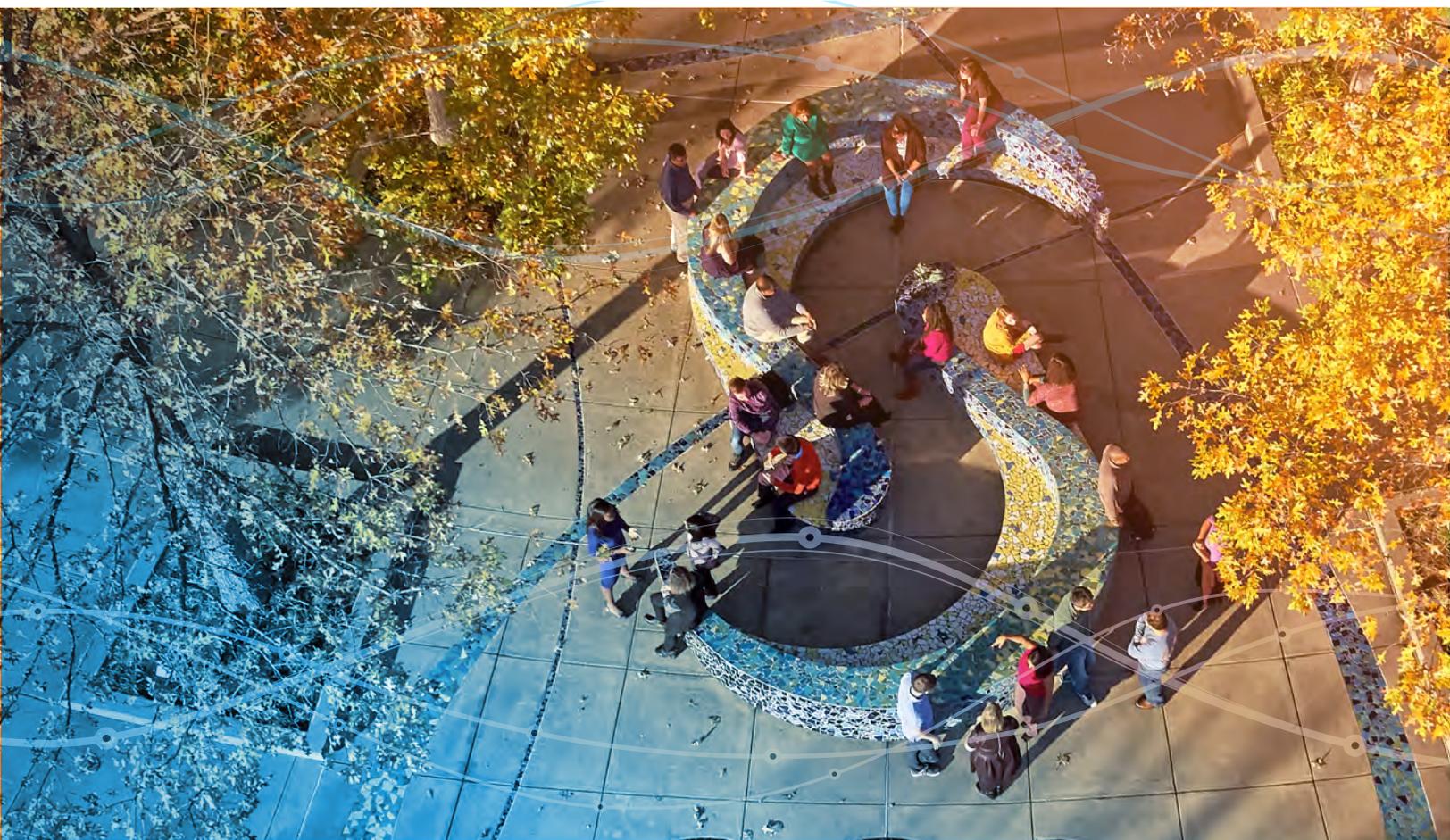


# The Quality Imperative: SAS Institute's Commitment to Quality

A corporate statement of SAS' commitment to product quality,  
service quality, and customer satisfaction



# SAS® Architecture: Quality by Design

## SAS® Viya®

SAS Viya was created to deliver an elastic and scalable cloud-ready analytics engine that embraces open analytics coding environments. SAS Viya provides a unifying environment for the entire analytics life cycle, with powerful analytic techniques that are accessible from a variety of interfaces, including programming, scripting, and visualization. These include:

- A Multicloud Architecture with no infrastructure lock-in. SAS Viya can scale to accommodate growing data volumes, more users, or more complex analytics. SAS Viya supports both public and private cloud deployments.
- Supporting a single, consistent platform for management of the entire analytics life cycle, which is open to both SAS and other programming languages such as Python, R, Java, and Lua calling into a single, underlying analytics code base.
- Providing access to analytic techniques (machine learning, descriptive statistics, forecasting methods, optimization algorithms, and so on) from a variety of interfaces – programming, scripting, and visualization.
- Automatically distributing data and analytical workloads across the cores of a single server or the nodes of a massive computing cluster, taking advantage of parallel processing regardless of data size.

## Consolidated Analytic Environment

SAS Viya can be accessed via modern visualization clients, REST APIs, and interfaces from other programming languages. The SAS Viya analytic procedures are consolidated in SAS Cloud Analytic Services (CAS server) with a single point of administration and management. All interfaces to SAS Viya access this layer for analytic processing so that no matter how users interact with SAS Viya, they receive consistent results.

## Cloud-Ready Technology Stack

SAS Viya is built on a cloud-ready technology stack. From the SAS Cloud Analytic Services that power SAS Viya analytics at the core to the microservices that supply the REST APIs and functional interfaces, SAS Viya is built to be cloud native. SAS Viya uses open-source technologies such as Java and Spring Boot to deliver a set of microservices. These microservices support common functionality, such as login and authorization, identity management, preferences, auditing, data management, data access, and more. SAS Viya uses the OAuth open standard for authorization, allowing SAS Viya to integrate with third-party clients and services. SAS Viya also provides public REST APIs and uses TLS to secure communications. SAS Viya can deploy on IaaS providers like AWS and Microsoft Azure, as well as on premises in private cloud, virtualized, and physical machine environments.

## SAS® Viya® 4

SAS Viya 4 is a continuation of SAS' journey to a fully cloud-native and optimized architecture. A system capable of managing the full analytics life cycle requires many different components.

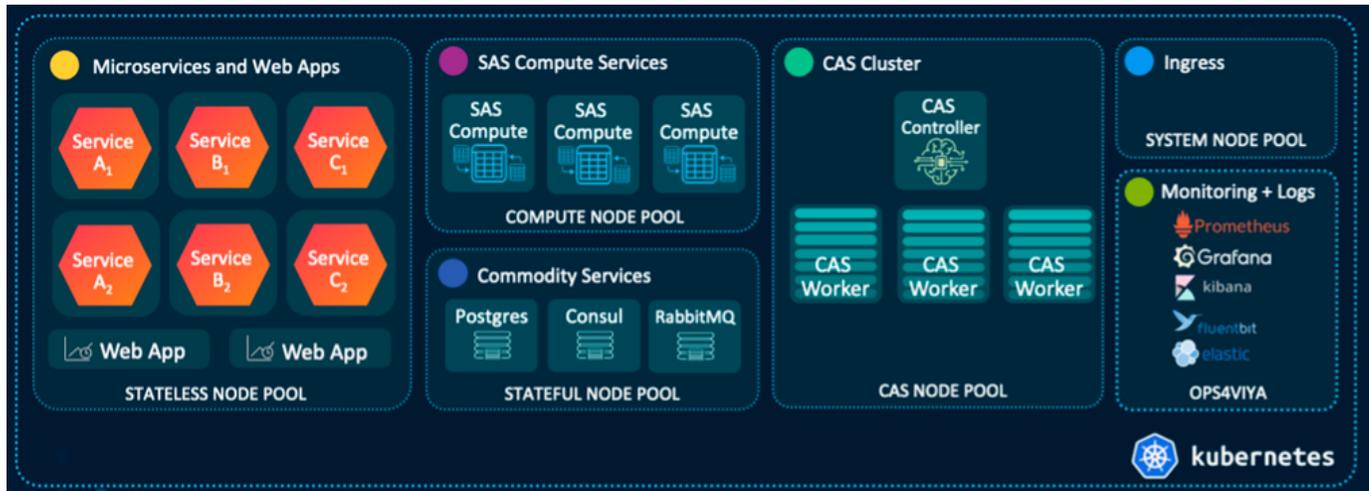


Figure 2: SAS Viya 4 Architecture.

### Cost Efficiency

Running enterprise software in a cost-efficient manner is not achieved through a single decision or goal. It requires a coordinated set of tasks that include limiting the baseline cost, allowing scale where necessary, and allowing an "off" switch when the software is not in use. The SAS Viya 4 architecture addresses each of these concerns:

- Many components have been moved from Java to a more memory-efficient platform, Golang.
- Components can be scaled individually, enabling customers to allocate cost where necessary to meet their business goals.
- SAS provides mechanisms within its deployment to turn off (and back on) as much of the system as practical when it is not being used.

### Containers and Kubernetes

SAS delivers its Viya 4 software via container images. Container images are a deployment currency of the cloud due to their isolation properties. Using containers enables us to ensure that required libraries are installed correctly and that compute-related resources are utilized and shared appropriately. Containers delivered by SAS use the Open Container Initiative (OCI) specification and are compliant with it.

Managing many different containers can be a challenge without a framework and tooling in place designed for it. Kubernetes is a standard answer provided for today's cloud. By deploying in Kubernetes, SAS takes advantage of its many features to deliver a solid administrative experience such as automatic handling of restart, managed updates without disruption, and the scheduling of diverse workloads on appropriate hardware.

## Continuous delivery

Another characteristic of cloud-native software is increased frequency of updates. Features and fixes are made available to consumers when they are ready rather than waiting for a coordinated roll-out at some pre-defined interval. To achieve this, SAS has incorporated common continuous delivery practices into its software factory.

As code is being proposed for integration to the mainline branch, it goes through an initial set of validations that include peer review, unit tests, basic integration tests, and linting. Once merged, code goes through a set of increasingly narrow gates dealing with complex integration scenarios, performance validation, and more in-depth functional tests. Only after passing all of these gates does it make it to a point of being customer visible.

Frequent releases do not completely remove the need for cross-team collaboration. In these instances, SAS utilizes feature flags to avoid enablement of code prior to the coordinated release.

## SAS® 9.4

At SAS, quality by design is evidenced in several areas such as these:

- The use of intelligent components (intelligent clients, intelligent storage, intelligent servers)
- SAS software's MultiVendor Architecture
- Shared sublibraries and code reuse
- The use of maximum numerical precision

Third-party components are often integrated into SAS offerings, making intelligent architecture a necessity. Our intelligent architecture is used for SAS Foundation, the SAS Intelligence Platform, and SAS solutions.

## SAS® Foundation

SAS Foundation is based on MultiVendor Architecture, which facilitates developing, managing, and maintaining the source code of the system and enabling quality to be built in at the structural level. Because the SAS Foundation development process is built around SAS MultiVendor Architecture, the amount of code that is rewritten for each operating environment on which our software runs is minimized. Thus, the chance for errors decreases because about 85% of the code is reused on all operating environments that are supported by SAS software.

The fundamental goal of SAS MultiVendor Architecture is to provide the highest degree of portability across a broad range of operating environments while exploiting the advantages of each.

## SAS® Intelligence Platform

SAS 9.4 uses an n-tier architecture that enables distributed functionality across computer resources so that each type of work is performed by the resources that are most appropriate to the job. For a large company, the tiers can be installed across multiple machines with different operating systems. For prototyping, demonstrations, or small enterprises, all of the tiers can be installed on a single machine. The architecture consists of the following four tiers: Data Sources, SAS Servers, Middle Tier, and Clients.

SAS 9.4 middle-tier components include a service-oriented architecture that is built around its Web Infrastructure Platform. Java J2SE and J2EE technologies, which are portable and reusable, are used for desktop client and web application components of SAS®9.

The adoption of J2SE and J2EE technologies enables SAS to use the development, testing, and customer acceptance baselines of the broad Java vendor and customer community. These technologies take advantage of Java's portability to operating-system and application-server deployment environments. This further enables customers to use the IT standards, acceptance testing, support staffing, and operating practices that are established within their organizations for this class of application.

For more information, see "Architecture of the SAS Intelligence Platform," in SAS 9.4 Intelligence Platform: Overview, Second Edition (<https://go.documentation.sas.com/doc/en/bicdc/9.4/biov/titlepage.htm>).

## SAS® Solutions

SAS solutions provide industry-specific functionality in these key focus areas: Analytics Platform, Artificial Intelligence and Machine Learning, Customer Intelligence, Data Management, Fraud and Security Intelligence, Risk, and Retail. Most solutions extend SAS architecture by using a component based on Java for the business logic and surface the functionality through web-based thin-client and rich-client presentation layers. Using SAS architecture gives SAS solutions several advantages:

- Solutions can scale from one machine to multimachine implementations to meet the performance needs of the customer.
- The common metadata repository enables common data sharing and management across systems.
- Most important, the SAS architecture enables the solutions to draw upon the analytical power of SAS to differentiate SAS solutions from those of competitors.

### Release Information

The version of this paper is January 2022.

Unless otherwise indicated, this document relates only to SAS 9.4, SAS Viya, and the products that are available with SAS 9.4 and SAS Viya. It also relates to services from the date of this paper forward. Quality processes are continually evolving. Therefore, SAS reserves the right to modify the processes described in this document at any time. If you are using SAS 9.4 and SAS Viya and have questions about processes in those releases, send email to [qualitypaper@sas.com](mailto:qualitypaper@sas.com).

Learn more about SAS Solutions at [sas.com](https://sas.com).

