White Paper


# Velocity Framework

## Built for Business

### Overview

Spirent provides the leading Lab as a Service (LaaS) and Test as a Service (TaaS) solutions for network development and test teams. LaaS enables networking developers, testers, pre-sales, and post-sales support teams to control and consume networking lab resources in an on-demand basis. TaaS similarly enables on-demand continuous testing delivery. Both LaaS and TaaS integrate seamlessly into DevOps tool chains to optimize efficiency for networking equipment manufacturers, service providers and enterprises. Furthermore Spirent's Lab and Test as a Service Platforms are tightly integrated to maximize reuse, test coverage and user efficiency.

Both LaaS and TaaS orchestrate physical, virtual, and hybrid environments. They empower DevOps workflows with continuous testing across multiple testbeds. LaaS and TaaS are proven to reduce CapEx, increase quality, and yield time-to-market and efficiency gains. This is achieved by maximizing equipment utilization, accelerating test bed setup time, minimizing power consumption and providing actionable management information to improve decision making. Spirent's solution tightly integrates LaaS and TaaS to maximize reuse and test coverage and ensure a seamless user experience.

The LaaS and TaaS capabilities are delivered via Spirent's Velocity platform. Velocity itself is described in the Velocity Datasheet. This white paper focuses on the Velocity Framework that underlies the Platform. It also informs about incorporating the Velocity Framework in your DevOps toolchain.

This white paper details the Velocity Framework. It explains how your organization can develop on top of each component of the Framework in order for your organization to meet its time-to-market and quality objectives while cutting costs. The Velocity Framework turbo-charges your DevOps workflows with Continuous Testing across all your test beds.

## Velocity Framework

**Built for Business**

The Framework is designed and architected according to 4 Build It Smart (BIS) imperatives.
They are:

1. Built in **Simplicity**: Purpose-built and customizable LaaS/TaaS foundation services enable your developers to efficiently personalize for your organization's needs
2. Built in **Speed**: Dynamic, on-demand test environments can be rapidly and reliably delivered
3. Built in **Savings**: Leverage Spirent's over 100 person-years investment in customizable software services. You build only what's needed to create your own LaaS/TaaS experience
4. Built in **Scalability**: The Velocity platform scales up and down to satisfy your testing capacity demands

### LaaS

Fast–Instantiate environments in minutes

Easy–Model, automate and share with anyone, anywhere

Scalable–Enterprise design enables fast adoption and ROI

Open–DevOps capable framework augments existing systems

### TaaS

Intelligent–Environment aware test case management, execution and analysis

Easy–Publish, schedule and share tests with anyone, anywhere

Performance–System will intelligently deploy tests to minimize test times

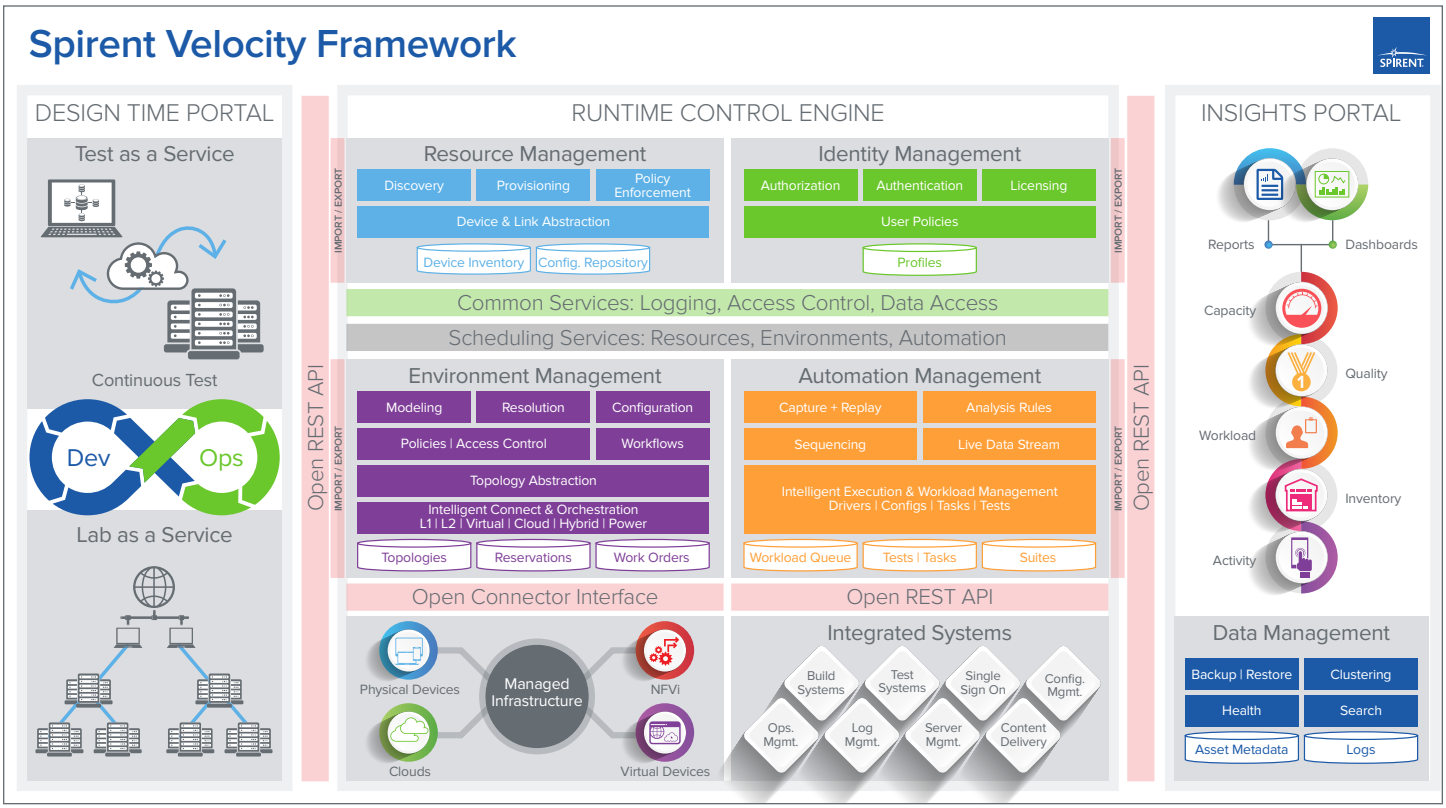
Go live–Streamline test execution from lab to production environments

## Framework Components Deliver Network Lab and Test as a Service

The framework has 2 portals for accessing, controlling and extracting actionable information. The portals are tightly integrated with the underlying Runtime Control Engine.

The key components of this framework are:

- ☑ **Design Time Portal:** provides the user interface for asset creation and consumption
- ☑ **Insights Portal:** Provides actionable information used by managers and/or engineers to perform their duties more effectively and efficiently
- ☑ **Runtime Control Engine:** provides the services needed to operate, manage and report on all information needed by the LaaS and TaaS platforms. The Runtime Control Engine includes services and APIs that deliver the LaaS and TaaS platform functionality.



# Velocity Framework

**Built for Business**

## Design Time Portal

The Design Time Portal is means with which users interact with the Velocity platform. Such interactions can be transacted via a web browser user interface or programmatically via REST API requests. The primary applications the portal supports are Lab as a Service and Test as a Service.

With Lab as a Service users can model, automate and deploy environments within their managed lab infrastructure. LaaS enables them to take advantage of this service to automatically provision the resources and test environments required by the tests being executed. Modeling is supported by inventory management and topology editors featured in the Velocity and iTest UIs. To satisfy this requirement, Velocity controls the managed infrastructure (including physical devices, switches and routers and/or hypervisors and orchestrators) to ensure the user's requirements for a test bed or topology are met. Once an environment is reserved or instantiated, Velocity deploys the specific configuration(s) needed to enable that environment's testing readiness.

With Test as a Service the Spirent platform enables users to create, automate and scale. That means automation (in the form of iTest test cases or Python and Bash scripts) can be created and consumed. Automation in this context means the ability to execute these scripts or test cases on demand or scheduled either on a one-time or recurring basis. Scale implies the user's ability to expand the amount or frequency of automation that the framework should manage.

## Insights Portal

The Insights Portal provides users the means to gain insights into all important aspects of the LaaS and TaaS solutions. This includes:

- ☑ Resource utilization
- ☑ Lab operational excellence
- ☑ Automation performance
- ☑ User performance
- ☑ Quality management

Dashboards are a primary mechanism for insights delivery. They are provided out of the box via open source Kibana which takes advantage of all the reporting and utilization data held in Elasticsearch. This includes historical data regarding every item of automation that has been executed including what it was, who initiated it, its parameters, requirements and environment used to execute it, and the final status and result of the execution. This allows users to gain insight into the quality of their automation such as how stable and effective it is. It also provides insight into activity levels and workload of lab resources including their managed infrastructure. Beyond dashboards, Velocity provides tailored reports that can be delivered to any specific audience including engineers, managers and executives.

## Runtime Control Engine

The Runtime Control Engine is the enabler for the Design Time and Insights Portal. It is a modular set of services that each customer can choose to use or not use based on their specific needs. For those that cannot afford the time and/or investment necessary to construct a solution themselves the engine seamlessly knits all the services into a functioning whole.

Of course, all the services can be accessed via a REST API which is consumable by Spirent and/or Customer supplied services or applications (e.g., Landslide or Jenkins). All access via this API is controlled by the API gateway, which is responsible for making a distributed set of services appear as one endpoint. These APIs are used by the Velocity web UI as well as iTest GUI. They are also the main mechanism through which Velocity can be integrated into a customer's automation framework.

The Runtime Control Engine is built out from four "pillars". They are:

- ☑ Identity Management
- ☑ Resource Management
- ☑ Environment Management
- ☑ Automation Management

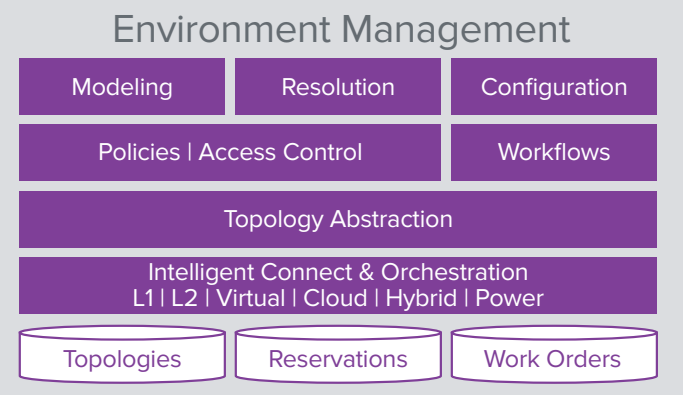
## Identity Management

Ensures only authorized users can access the system and ensures access policies are enforced. For example, every API request needs to be authorized. This is done via standards-based access tokens attached to each request. It is the Identity Management functionality that enables tokens to be obtained. Authentication is performed via a variety of credential management systems. This includes LDAP and Single Sign On solution, via oAuth or SAML. Once authenticated as a user, Velocity services validate whether the user is authorized and what configured policies apply to each. In addition, the set of resources the user is provided access to can be controlled using a combination of user groups (that the user belongs to) and resource groups which simplify controlling resources as sets. Licensing is then checked to ensure the access attempt has a license associated with it.

Every service in the framework has access to several common services. Logging allows the framework to collect and aggregate data about the health and performance of each service into a central location using our Kibana and Logstash stack. Access control allows each service to ensure that the user token being used is valid and the associated user has access to the requested API endpoint. Data access allows each service to access the data stores needed by that service in a unified way.

## Resource Management

This provides for tracking and controlling the resources inventory being modeled within Velocity. Discovery is performed using standards-based methods such as NMAP and SNMP. Provisioning and policy enforcement are involved in ensuring that the resource being worked with is setup correctly and that the user attempting to access it can do so under enforceable constraints. Device and link abstraction is a key part of our solution set. Velocity deals with resources via an abstraction layer known as "templates". Templates enable the user (typically an admin) to categorize resources as belonging to certain classes and establish rules and properties associated with all instances of that device class.




## Environment Management

This encompasses all the functionality required to instantiate an environment as well as to control and configure it. Modeling consists of all the functionality needed to support the Velocity graphical topology editor, including import and export of topologies in various standards-based formats, including TOSCA/YAML and TBML. Resolution is the act of mapping a desired topology containing a set of actual and abstract resources and links, specified via device and link constraints, to a set of concrete resources and the actions needed to setup the connectivity between them. Once the set of resources to be used is found and reserved, configuration can be applied to them in the form of images and configuration settings.

As with all services, the framework ensures that all configured user access control is checked and that user policies are followed when a user attempts to gain access to an environment managed by the framework. These policies may invoke various workflows, including work orders that involve physical cable management. Under control of the workflow system, emails can be sent to a configured set of users to perform required cabling tasks and react to when that task is complete to continue the automated setup of the environment.

Key to all of this is topology abstraction wherein we can use constraints, using a rich constraint specification language to map desired resources to existing ones. During resolution, this abstraction is turned into a concrete manifestation of a topology which is then configured and if needed, instantiated via the framework's Intelligent Connect and Orchestration layer. This layer can use drivers to control layer 1 and layer 2 switching and routing infrastructure to setup the desired connectivity. It is able to control a number of different hypervisors and orchestrators, including OpenStack to instantiate virtual resources and their associated networking. It is powerful enough to manage hybrid topologies, setting up the connectivity between physical resources in a lab and virtual resources in a cloud. Drivers are powerful and flexible enough for the framework to be able to use them to control a variety of types of infrastructure devices including power management, and even perform health monitoring.

# Velocity Framework

**Built for Business**

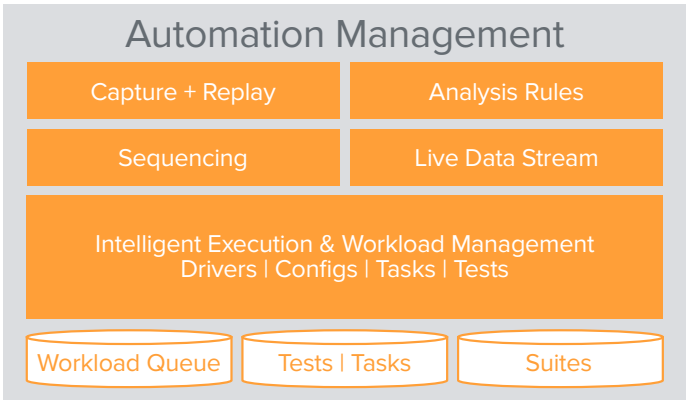
## Automation Management

Both LaaS and TaaS heavily rely on automation which is enabled by this portion of the framework. Automation Management enables several capabilities including 1. creation and execution of test assets, 2. driver development and execution, 3. setup and tear down of automation tasks.

To most rapidly create automation, users can use Capture and Replay to record actions and execute them on-demand. Capture supports a variety of devices and software classes, including SSH, Telnet, Web Applications, GUI and User Interface Test Software and Traffic Generators. Once captured, these steps are transformed into a reusable automation script in either iTest test case format or Python. Analysis rules can then be added allowing for advanced logic, such as branching and looping by taking advantage of iTest's patented response mapping technology. Existing automation assets are easily imported into the framework where they are available for sequencing. The sequencing capabilities of the framework are exposed through Runlists which are defined sequences of automation scripts that should be executed, with criteria defined for the environment needs to be provided, and what requirements the runtime environment should have to enable execution.

Intelligent execution and dispatch is responsible for queuing and sequencing automation execution. This ensures optimal utilization of lab resources. Intelligent dispatch finds the appropriate execution agent within the managed infrastructure to satisfy the requirements of the execution. For example, a specific OS or geographical location may be required.

As automation is being executed, the framework receives a stream of reporting events from the execution agent which is funneled into the Elasticsearch data store where it is available for the insights framework to use.




Part of the common services of the framework includes a scheduling service. This allows both topology and resource reservations to be scheduled for a later date as either a single occurrence or a recurrence. Likewise, automation execution can be scheduled as a single occurrence or a recurrence and may take advantage of a scheduled reservation of an environment that is needed for the execution.

All the data generated by the Runtime Control Framework, and made available to the Insights Framework, is stored in a set of data stores. These include Elasticsearch (where the reporting data is kept), as well as Postgres (where the resource inventory and resource utilization metrics are kept). All of that requires data management capabilities. This includes backup and restore, the ability to monitor health, and the ability to manage the scale of the data via clustering. It also encompasses the ability to search the data to enable insight into it.

The Velocity Framework is architected with individually selectable components for ultimate flexibility in order to fit in with your existing tooling. Please contact your Spirent Sales or Technical Professional to arrange an in-depth review of how the Velocity Framework can help you achieve your objectives.

spirent™
Promise. Assured.


**Contact Us**

For more information, call your Spirent sales representative or visit us on the web at www.spirent.com/ContactSpirent.

**www.spirent.com**





Americas 1-800-SPIRENT
+1-800-774-7368 | sales@spirent.com

US Government & Defense
info@spirentfederal.com | spirentfederal.com

Europe and the Middle East
+44 (0) 1293 767979 | emeainfo@spirent.com

Asia and the Pacific
+86-10-8518-2539 | salesasia@spirent.com



Rev C | 08/18