



Open Services Gateway Initiative (OSGi)

Specification Overview

Version 1.0, January, 2000

Copyright © The Open Services Gateway Initiative (2000). All Rights Reserved.

Implementation of certain elements of the Open Services Gateway Initiative (OSGi) Specification may be subject to third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be a member of OSGi). OSGi is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

This document and the information contained herein are provided on an "AS IS" basis and OSGi DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL OSGi BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

All Company, brand and product names may be trademarks that are the sole property of their respective owners. All rights reserved.

Overview of the Open Services Gateway Initiative

The Open Services Gateway Initiative (OSGi) was founded in March 1999 with a mission is to create open specifications for the delivery of multiple services over wide-area networks to local networks and devices. Since then, the initial fifteen founding member companies have been joined by another twenty-two companies giving a total OSGi membership of over thirty companies from the software, hardware and provider businesses. OSGi is speeding the adoption of its technology and accelerating demand for products and services based on these specifications worldwide through the sponsorship of market and user education programs.

The three key aspects of the OSGi mission are multiple services, wide area networks, and local networks and devices. Unlike other initiatives, OSGi concentrates on the complete end-to-end solutions architecture from remote service provider to local devices. Because the OSGi specification focuses on providing an open application layer and gateway interface, it complements and enhances virtually all current local networking standards and initiatives. The OSGi specifications deliver an open, common architecture for service providers, system developers, software vendors, device vendors, and equipment manufacturers to easily develop, deploy and manage multiple services in a coordinated fashion.

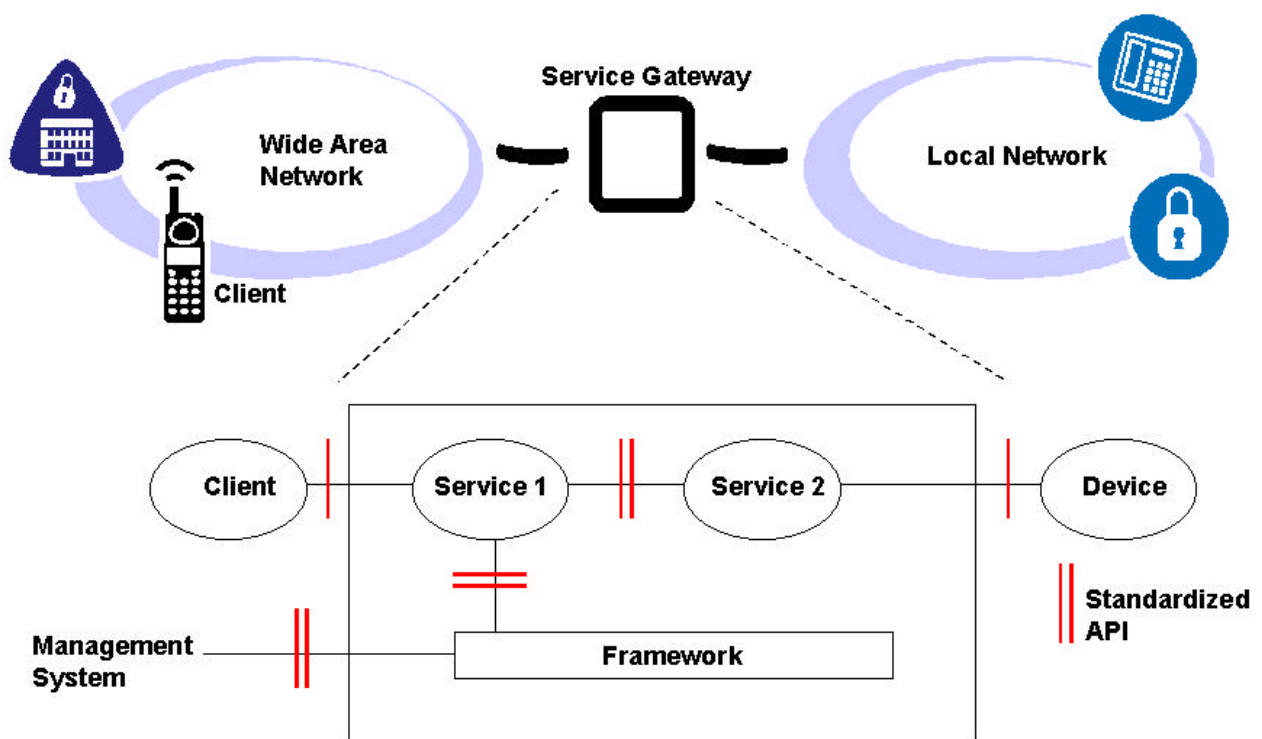


Figure 1. A Service Gateway

The central component of OSGi specification effort is the services gateway that functions as the platform for many communications based services. The services gateway can enable, consolidate, and manage voice, data, Internet, and multimedia communications to and from the home, office and other locations. The services gateway can also function as an application server for a range of high value services such as energy management and control, safety and security services, health care monitoring services, device control and maintenance, electronic commerce services and more. The gateway provides a focal point for service providers to deliver services to client devices on the local network(s). An implementation of the OSGi gateway may link client devices on the local network(s), such as energy meters, smart appliances or information appliances, to external service providers. An abstract view of the wide area network, local network, and services gateway is shown in Figure 1.

Though enabling the connectivity and management of entirely new categories of smart devices, the services gateway will likely be integrated in whole or part in existing product categories such as (digital and analog) set-top boxes, cable modems, routers, residential gateways, alarm systems, energy metering and management systems, consumer electronics, PCs and more. While Figure 1 show a single services gateway, the OSGi architecture supports multiple services gateways, multiple wide area network points of access, and multiple local networks working together to provide different services from different providers.

This overview concentrates on the technical aspects of the OSGi end to end system architecture and design of the services gateway APIs.

Key Benefits of OSGi

The founding member companies all recognized the need for open standards in the area of remotely deployable service platforms. Only by coming together, contributing ideas and defining specifications can we create standards that both vendors and service providers can accept. The following list is a high-level summary of the key technical benefits associated with the OSGi specifications.

Platform Independent

The OSGi APIs can be implemented on a wide range of hardware platforms and operating systems to tailor the services gateway to the unique needs of a particular market.

Application Independent

The OSGi specifications focus on defining common implementation APIs, making it suitable for services gateways for a variety of applications in different markets. While initially focused on the requirements of the residential gateway market, OSGi services gateways can be used in telematics applications, embedded in PDAs and cell phones, and supported on PCs and other consumer electronics devices.

Secure

The OSGi specifications incorporate various levels of system security features, ranging from digital signing of downloaded modules to fine-grained object access control.

Multiple Services

The OSGi specifications support hosting multiple services from different providers on a single services gateway platform. This flexibility allows a gateway operator to offer a wide array of services to their customers.

Multiple Local Network Technologies

A variety of wired and wireless, data and audio-video local transport standards are emerging. The OSGi specifications are purposefully designed to complement and enhance local network and device attachment technologies such as Bluetooth, the Home Audio and Video Interoperability (HAVi) standard, Home Phoneline Networking Alliance (HomePNA), HomeRF, IEEE-1394, LonWorks, powerline communication systems, Universal Serial Bus (USB), the Video Electronics Standards Association (VESA) home network, wireless systems, and other local network technologies.

Multiple Device Access Technologies

By concentrating on the service implementation aspects of the platform, OSGi supports a variety of device access technologies, such as Universal Plug and Play (UPnP) and Jini, as a part of the device access API. These device access technologies let users control devices from many different places and diverse client devices. The standards define the discovery process of devices on the network, how commands can be issued to other devices and how events from devices can be received. OSGi and device access technologies, such as UPnP and Jini, are very complimentary, solve very different problems and are a natural fit to cooperate.

Co-existence with other Standards

The OSGi specifications are designed to complement emerging standards in local device connectivity and in wide-area network access. OSGi is compatible with physical local transports such as Bluetooth, the Home Audio and Video Interoperability (HAVi) standard, Home Phoneline Networking Alliance (HomePNA), HomeRF, and Universal Serial Bus (USB). The device access architecture of OSGi is compatible with several different logical device access technologies, such as Jini and Universal Plug and Play (UPnP). OSGi service gateways can use narrowband Internet access approaches as well as the faster digital subscriber loop (DSL) and cable modem technologies. OSGi gateways may also be compatible with market-specific device and network management schemes, such as DOCSIS and SNMP.

Markets for Open Services Gateways

The OSGi architecture is designed based on the requirements from several new markets and businesses. The initial primary focus is on the residential gateway, though the services gateway concept has applications in telematics, commercial, enterprise and industrial settings. In the residential gateway market, OSGi has used several scenarios to guide the architecture and requirements development.

The Residential Gateway

A Residential Gateway is the device that connects a home area network to the Internet. With the deployment of high-speed Internet connections and the push by service providers to offer integrated voice, data and video services over the same high-speed pipe to different nodes throughout the home, the residential gateway is expected to become a key integrated services enabler. Technology drivers of the emerging residential gateway market are:

- The rapid emergence of viable, standards-based home networking technology such as those based on HomePNA and HomeRF,
- New entertainment options including Digital TV and web based/interactive TV,
- Non-PC based Internet Appliances.

The number of different types of services, as outlined below, shows the possibility for multiple services gateways within a residence, each with potentially different means of wide area network access. Some of these gateways will rely on wide area network access from the gateway operator/carrier and others will use whatever Internet access the residence already has. Alternatively, service aggregator may bundle services from each category together creating different market spaces from those currently envisioned.

Communication Services

Communication services are likely to be among the first service segments to develop. The primary consumer applications are likely to be shared Internet access, both for PC and Internet appliances, the ability to network multiple devices in the home, and value added telephony services, such as voice over IP and multiple phone connections over one line. The key architecture requirements are simplicity, affordability and the ability to personalize.

The primary business opportunity will be to provide the full range of communication needs to the consumer. The service provider will be able to generate multiple revenue streams from each consumer, create economies of scale efficiencies across multiple parts of their value chain, and build greater consumer loyalty and entry barriers for competitors.

Energy Services

Driven by deregulation, utilities are looking for new ways to manage costs, create revenue and ensure customer loyalty. Obvious services are automatic meter reading and load management. Distribution companies have much to gain from these services when it

comes to acting as a clearinghouse for energy producers and in network optimization, load balancing, and outage detection. Service gateways combined with new local network technology opens up opportunities for energy management services that help consumers reduce their energy consumption and/or increase comfort.

Home Automation Services

While current home automation starter kits are relatively inexpensive, the complexity of the interface and the sophistication of the programming language have limited growth. The key drivers for this application segment will be ease of use and the penetration of network enabled devices. The primary business opportunity will be to bundle home automation services with other residential services. While the ability to automate the home is unlikely to be compelling to most consumers from a stand-alone perspective, it could be very interesting when combined with services that have a clear consumer utility.

Security Services

Security services are already a relatively mature market. However the shift towards bundled integrated systems and new entrants from the telecom and energy industries are changing the market. From a consumer perspective, security systems based on a services gateway would be a substitute for existing security systems and would provide a more flexible platform with new and more advanced services such as remote monitoring and control of the system. Remote access to the security system could be accomplished using a range of client devices, such as a standard Internet browser or a mobile phone. The primary business opportunity for service providers is to integrate security systems with current core services, and to offer a broader range of security services.

Remote Home Healthcare Services

The home care segment represents a large and relatively untapped market for service providers. The user benefits come from providing “peace of mind” to relatives of elderly and disabled people through security and monitoring services, and offering communication services such as user friendly home shopping and video telephony. The likely services for this segment are monitoring and sensing devices (such as, motion detectors), personal alarms, assistance tools (simplified Internet home shopping) and communication (for example, e-mail and video conferencing). Again, these services may be bundled with other services to serve vertical-style user markets from a set of horizontal services.

Architectural Overview

The OSGi technical specification creates an open standard for programmable user-premises equipment that includes: downloading of software, application lifecycle management, gateway security, attached device access, resource management, and functions necessary for remote administration of the gateway.

The major identified components of the complete end to end OSGi architecture model are:

- Services gateway
- Service Provider
- Service aggregator
- Gateway operator
- Wide area network and carrier/ISP
- Local devices and networks.

The relationships between these entities are shown diagrammatically in Figure 2.

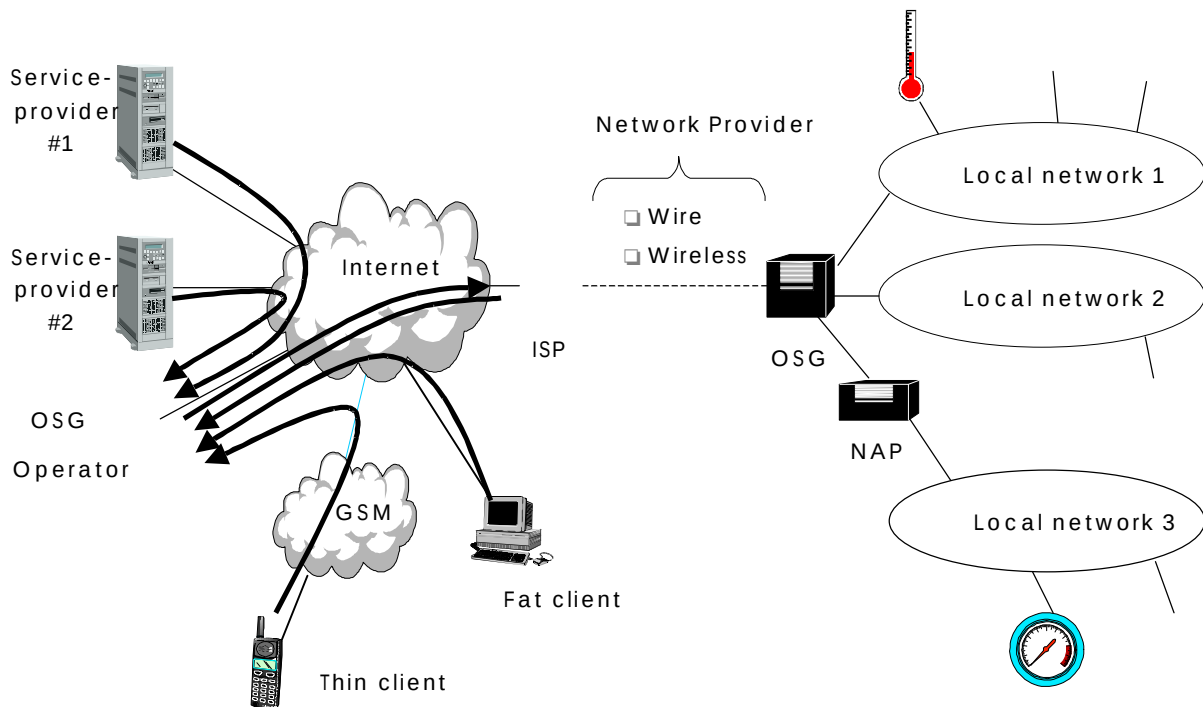


Figure 2. OSGi End to End Architecture

Services Gateway

The central component of the OSGi end to end architecture is the open services gateway (OSG). Technically, the services gateway is an embedded server that is attached to the wide area network to connect external service providers to internal clients and effectively separates the topology into the external network and the internal network. The services gateway is a secure, zero local administration device with connectivity to local devices. The OSGi specification includes APIs for service cradle-to-grave life cycle management, inter service dependency management, data management, device access and management, client access, resource management, and security. More details on the OSGi specifications are provided in a later section.

Service Provider

The service provider provides a service using the services gateway in conjunction with an optional remote server and possibly local client devices. The service on the gateway is a software application downloaded by the gateway operator into the gateway. Example services include:

- Maintenance service checking that the appliances in the house are in working order,
- Itemized invoicing according to electricity consumption,
- Physical security system monitoring sensors and activating alarms upon command from a remote server.

If a service provider is trusted by the gateway operator, the secure download facilities ensure that only trusted code is installed on the gateway.

Service Aggregator

A Service Aggregator is a special type of service provider who offers a set of services (for example, automatic reading for electricity, gas and water meter, alarms and energy management) bundled together. The service aggregator aggregates all of the services that are allowed to be loaded into the gateway and has the responsibility to ensure that bundled services are mutually compatible and do not conflict in use of devices or in resource requirements. Because of the responsibility for putting together services, the term service aggregator is often used interchangeably with gateway operator, though strictly speaking the roles are different.

Gateway Operator

The gateway operator manages and maintains the services gateway and its services. This management responsibility may overlap somewhat with that of the service aggregator and is the reason the terms are often used interchangeably. Some functions of the gateway operator are:

- Download, start, stop, update and remove a service,
- Control the resources of the gateway, check the operating status of the gateway, and manage service versions,
- Manage the gateway and monitor device status,
- Define and control access rights between the gateway and the service providers,
- Make the communications between the gateway and the service providers secure,
- Check the validity and the rights of any dependencies one service may have on other services,
- Control and manage the logical attachment of devices and local networks to the gateway, including authorizing the download of device and network drivers.

The gateway operator may also play the roles of gateway owner (e.g., when renting the gateway), gateway retailer for purchased gateways, gateway installer (providing professional installation), and the gateway hardware maintainer. The gateway operator

role can be combined with that of service provider, service aggregator or wide area network provider.

The gateway operator is trusted and secure download facilities ensure that only trusted code is installed on the gateway.

Wide Area Network and Provider

The wide area network provides the necessary communications between the service gateway, the gateway operator, the service aggregator and the service provider. This communication service may be provided and managed by a carrier or, when the wide area network is the Internet, by an Internet Service Provider (ISP). The ISP can perform other functions and, in particular, may be the gateway operator. While combining ISP and gateway operator may be a popular in some market segments, such as the cable industry, it must be possible to have independence between the gateway operator and the ISP. For example, an energy utility should be able to run a services gateway without having to operate an ISP system and a residence may have several gateways, some of which are independent of the ISP. If one of the services provided by the gateway is Internet access the separation between ISP and gateway operator may be tricky in determining who is responsible for firewalls, access to files stored on home computer, routing, and other access issues.

Local Networks and Devices

The last major piece of the architecture is the local networks and devices attached to the services gateway. This attachment of devices makes the services gateway more than just another computer. Devices can be directly connected to the gateway using hardware links such as parallel or serial connections or indirectly connected by means of a local transport network such as BlueTooth, IEEE 1394, or wireless. The real value of the services gateway comes from the services it can provide in conjunction with remote service providers and local devices.

Services Gateway Overview

The major technical effort in the first release of the OSGi specification is on the APIs implemented on the gateway and available to service developers. Follow-on specifications will cover other parts of the end to end architecture. The first OSGi specifications will concentrate on Java APIs because Java is currently the best language for building **flexible** open services gateways. A Java-based service gateway is comprised of the following required components:

- Java Environment -- defines required packages and classes,
- Service framework -- defines API for creating and running services,
- Device Access manager -- defines APIs for accessing devices,
- Log Service -- defines a required service for logging information.

In addition to the required APIs present on all OSGi-complaint gateways, the following optional service is also being defined in the first specification release:

- HTTP Service -- defines a service API for an HTTP-based web server.

Follow-on API work will define additional optional services including:

- Client access service -- defines a service API for publishing information,
- Configuration data service -- defines a service for managing configuration information,
- Persistent data service -- defines a service API for persistent data.

The following sections provide additional technical details on each of the OSGi Java components.

Java Environment

To ensure a large target market for third-party service developers as well as a large selection of compatible services for gateway operators, OSGi specifies APIs standards for the gateway platform execution environment. Gateways must support these API standards in order to be compliant with the OSGi specifications. For a Java-based gateway, the APIs will be based on the Java 2 package and class definitions. The exact details on required classes and packages will be released along with the OSGi version 1 Java specification. The goal is to allow implementations of service gateways to be based on Personal Java (pJava) as well as on versions of Java 2 Micro Edition (J2ME) and other compatible Java runtime environments.

Service Framework

The service framework provides a context for developers to write service code to execute on the gateway. In this environment, services are swapped in and out, are dynamically updated, and must communicate in a structured and dependable way with other services. The framework provides a rich and structured development platform for component-based software architectures and takes advantage of the Java's ability to download code from the network. The primary goal of the service framework is to provide an environment that:

- Supports dynamic load or update of services without stopping the environment,
- Is usable in limited memory devices,
- Offers a concise and consistent component programming model for service developers,
- Manages dependencies between services,
- Is scalable.

To take best advantage of the framework, developers should design an application as a set of services, each implementing a segment of the overall functionality. These services and other extension services are then combined into a "bundle" and downloaded to the gateway device

A bundle is a Java archive (JAR) file that:

- Contains the resources implementing zero or more services. These resources may be class files for the Java programming language, as well as any other data (such as HTML help files, icons, and so on).

- States static dependencies on other resources, such as Java packages. If any dependencies are stated, the framework takes the appropriate actions to make the required resource available.
- Optionally contains classes that help the framework install, configure, activate and update a service.
- Declares which class should be used to start or stop a service.

By packing all these items together in a single JAR file, the framework can uniformly download and control a bundle. The JAR manifest file format is extended to contain the additional information about the bundle and the standard JAR signing technique is used to ensure the integrity and authenticity of downloaded bundles.

The framework maintains a set of mappings from services to their implementations and has a powerful query mechanism that enables an installed bundle to request and use any available service. The framework manages dependencies between services and bundles such that when a bundle is stopped, all the services that it registered will be unregistered automatically.

Device Access Manager

The device access manager provides on-demand, dynamically loaded services for devices and network hardware connected to the services gateway. These devices include items such as appliances, alarm systems with motion sensors, car navigation with a global positioning system, consumer electronics such as televisions, and so on. The device manager supports service providers who will want to access and communicate with devices and appliances without being burdened with low-level communication details.

To offer this support, the features of the device access manager are:

- Network independence -- The device access manager does not require any specific network architecture. New network technologies will simply require new network device drivers.
- Device independence -- The device access manager does not require any specific devices. New devices will simply require new device drivers.
- Automatic discovery -- The device access manager, where possible, is able to detect the addition of hardware devices, and then locate and load supporting device bundles with a minimum of user interaction. Bundle locations are implementation specific and hidden by the device access manager.
- Legacy support -- The device access manager supports older network technologies that do not support automatic device discovery. In addition, it supports devices that do not contain OSGi specific functionality because they were designed prior to the OSGi specification release. The device access manager controls all user interactions need to gather information about legacy devices.

The device access manager uses the concepts of a network bundle and a device bundle. The network bundle is a special type of service framework bundle that contains the necessary protocol stacks, drivers, and other resources to communicate to devices attached to the network. For example, the Universal Serial Bus network bundle contains the code to manage USB data transfer, detect new devices, and handle other USB

functions. The device bundle is another special type of service framework bundle that contains the code necessary to communicate with a specific type of device. The device access manager links together device and network bundles so that a single device bundle implementation can work with several different network bundles; for example, only a single implementation of a sensor device bundle is needed regardless of the type of network the sensor is attached to.

Log Service

The log service, the only required OSGi service, defines a service API to allow Java-based OSGi services to write entries to a log and for entries to be read from the log. The facility of a log as a required part of the services gateway will aid developers and operators in problem determination and resolution. The log service has the following general capabilities and features:

- Records the current system time in the log entry,
- Records a severity level in the log entry,
- Records a text message in the log entry,
- Records an optional Java Throwable object in the log entry,
- Records the identity of the Java program running in the Framework that created the log entry,
- Listens to Framework events and create log entries representing these events,
- Reads past log entries via an enumeration,
- Notifies listeners of log entries as the entries are created,

Gateway implementers are free to construct the log service in a variety of ways depending on the capabilities of the gateway, including spooling to a local log file or transfer to a central management agent.

HTTPService

Many services will be based on web protocols so an API has been defined for an HTTP-based web server that runs as part of the services gateway device. The API provides facilities to configure the server, publish static content, and publish dynamic content generated from Java Servlets. The HTTP Service provides the following capabilities and features:

- Support for the Java Servlet 2.1 (or higher) API specification,
- A Java Servlet Engine to provides the runtime environment for Java Servlets
- Support for HTTP/1.0 or HTTP/1.1 protocol with the recommendation that implementations of the HTTP/1.0 protocol support the Keep-Alive feature of HTTP/1.1 as a performance enhancement.
- An API to allow services to dynamically register and unregister static content, Java Servlets and other resources into the URI namespace of the HTTP server.
- An API to allow services which have registered content to verify the identity of the requestor (HTTP authentication) and if the communication channel is secure (SSL encryption).

Registering resources allows HTML files, GIF files, class files, etc., to be made visible in the URI namespace of the services gateway by a requesting service.

Client Access Service

In many cases, the services running on the gateway will need to occasionally interact with end users. Some of this interaction can be managed from remote servers. In other cases, a gateway-resident mechanism is needed to allow end users to view information in the gateway, modify configuration information, receive notifications, and otherwise interact with services in a uniform way.

The client access service API provides a framework for managing user interactions that minimizes the work for the service developer and goes beyond the basic facilities provided by the HTTPService. The client access service provides automated support for URL name space management and supports data format negotiation to support access devices ranging from mobile phones to PC browsers.

Configuration Data Service

Many of the services running in a gateway will need additional configuration information to perform their tasks. This information can be as simple as a short list of authorized devices or as complex as user personalization information. The configuration data service provides a common service API for services to store and access this data and a means by which a remote administrator can change configuration information.

Persistent Data Service

Many of the services running on a gateway will need to store and retrieve information that persists beyond the life of the service and that can be shared with other services. Rather than having each service have code to efficiently read and write information to a file system and recover from errors, a common persistence data service is provided that can be used by other services to store and retrieve information. Using this API, a service can make any piece of information persistent, is able to search the persistent information store using a high level query language, is able to recover from failure after partial updates, and is able to synchronize the data with a server database. Implementations of the persistent data service may allow persistent information to be backed up on a server machine (perhaps one managed by the gateway operator) as an extra measure of reliability.