SpyLogix™ for IIS was designed is response to customers wanting continuous visibility and real-time-analysis for user tracking when using web applications. Business users responsible for web based online information systems want feedback to effectively measure and disseminate timely information about how web applications are performing in support of business process financial objectives. IT staff tasked with supporting application infrastructures need efficient ways to quickly troubleshoot and resolve problems or maintain proactive operational awareness across complex technologies involved with keeping business information safe.

SpyLogix for IIS is a specially designed HTTP module which natively interfaces with IIS to harvest and centralize events for real-time processing using SpyLogix Platform security middleware services.

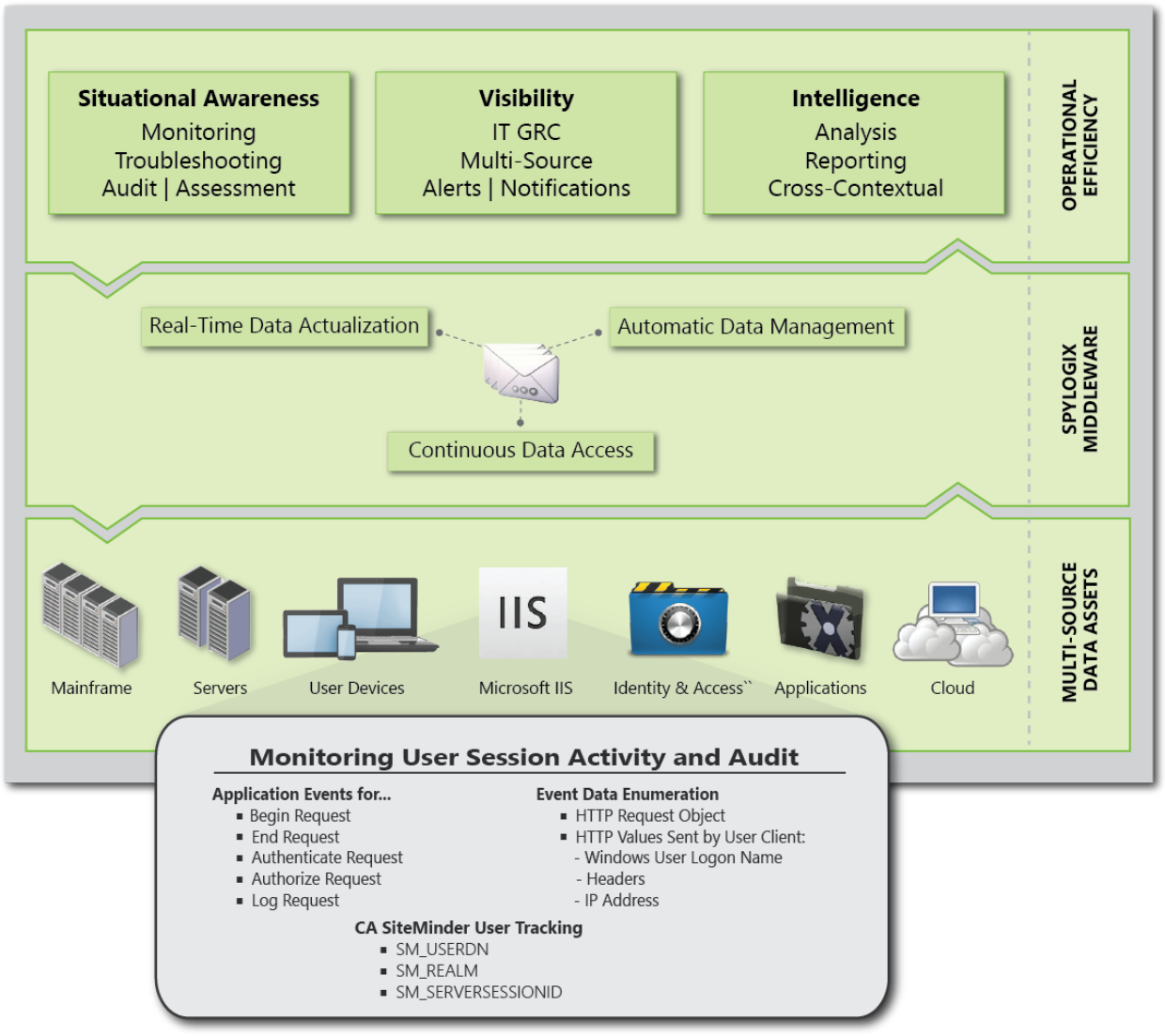

Figure 1 SpyLogix Enterprise

Who uses and benefits from access to new information?

To optimize business process efficiency using online information systems, deliver information easily to users, and keep data safe from ever-evolving cyber threats, innovative solutions are being sought to efficiently monitor digital assets continuously, and then using this new data proactively so people can perform effectively their assigned roles in keeping business information safe. SpyLogix works for:

- Business stakeholders continuously managing business return of application investments
- IT staff tasked with assessing daily operational awareness of applications
- IT staff performing periodic troubleshooting of complex information technologies
- Administrative staff (e.g. HR, help desk) helping users with proper access to information
- Actively sharing SpyLogix managed data with existing IT processes or tools

Special Capability - User Tracking for CA SiteMinder-Secured Websites

Designed with CA SiteMinder-protected IIS web servers in mind, the following specific user tracking data is mapped into each SpyLogix Message: SM_USERDN, SM_REALM, and SM_SERVERSESSIONID. Now the richness of user tracking data from IIS and SiteMinder data become immediately linked for continuous, real-time processing by SpyLogix Platform servers.

SpyLogix Platform Services Help Use Data

SpyLogix Platform servers make available services for:

1. Communication of standardized SpyLogix Messages (using industry-standard broker/protocol)
2. Data Management to automatically parse, translate and store SpyLogix Message data
3. Data Actualization, which:
    a. Analyzes streaming SpyLogix Messages in real-time using saved policies
    b. Analyzes streaming or historical data via an interactive console
    c. Shares new information faster with people, processes and technologies

Summary

SpyLogix impacts (top line) revenue and reduces (bottom line) costs related to providing users with online access to business information safely by…

1. Reinforcing business and IT management visibility for operational awareness
2. Improving effectiveness of work quality/productivity for business and IT staffs,
3. Enhancing efficiency of existing information security (IS) processes (a.k.a. IT services),
4. Mitigating risk of data security breach thru real-time monitoring for security technologies, and
5. Enabling ongoing secure online information access supporting business objectives.

How does SpyLogix process data to use enterprise-wide?

SpyLogix Enterprise is a scalable system for collecting data from multiple sources, streaming it enterprise-wide and continuously processing it for business optimization. This outcome is accomplished by SpyLogix Enterprise using four major components:

1. SpyLogix Modules access data natively from any source technology, build standardized SpyLogix Messages, and then send them to one or more SpyLogix Platform servers for processing.
2. SpyLogix Platform servers:
    a. Communication Services efficiently consumes native data and/or SpyLogix Messages, and then delivers data as configured to one or more SpyLogix Platform servers.
    b. Data Management automatically parses data from SpyLogix Messages, optionally translates data to human-readable form, and then stores data.
    c. Data Actualization includes ActionLogix™ to analyzes by policy in real-time streaming SpyLogix Message data and enacts pre-configured actions, Interactive Console to manually view/analyze streaming and historical data, and Web Services to share data with IT services or tools.

Click here for more information on SpyLogix Enterprise.

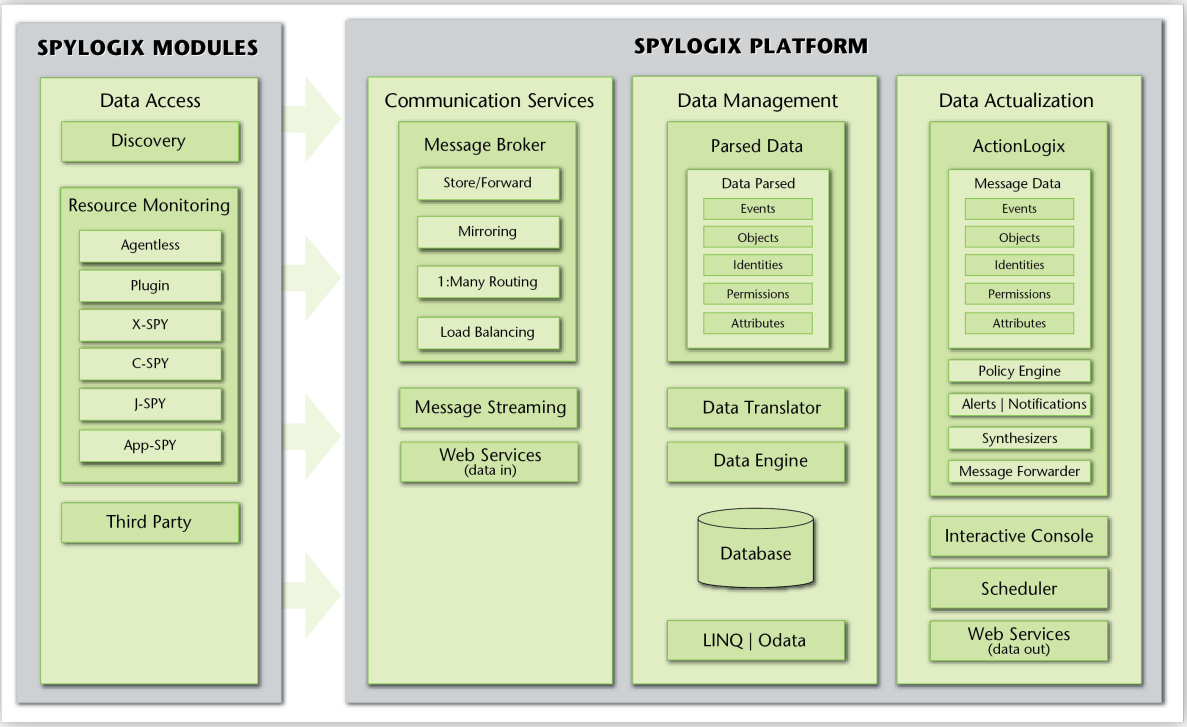

Figure 2 SpyLogix Enterprise Overview

What (technical details regarding) data are SpyLogix for IIS natively harvesting?

The SOA design of SpyLogix for IIS enables a scalable system for harvesting and leveraging streaming data from low to high "big data" rates. The shared-nothing software architecture is easy to deploy, with installation and configuration typically completed within an hour. SpyLogix may be deployed on premise, in a cloud or in various hybrid architectures to meet business needs.

SPYLOGIX FOR IIS HIGH-LEVEL ARCHITECTURE

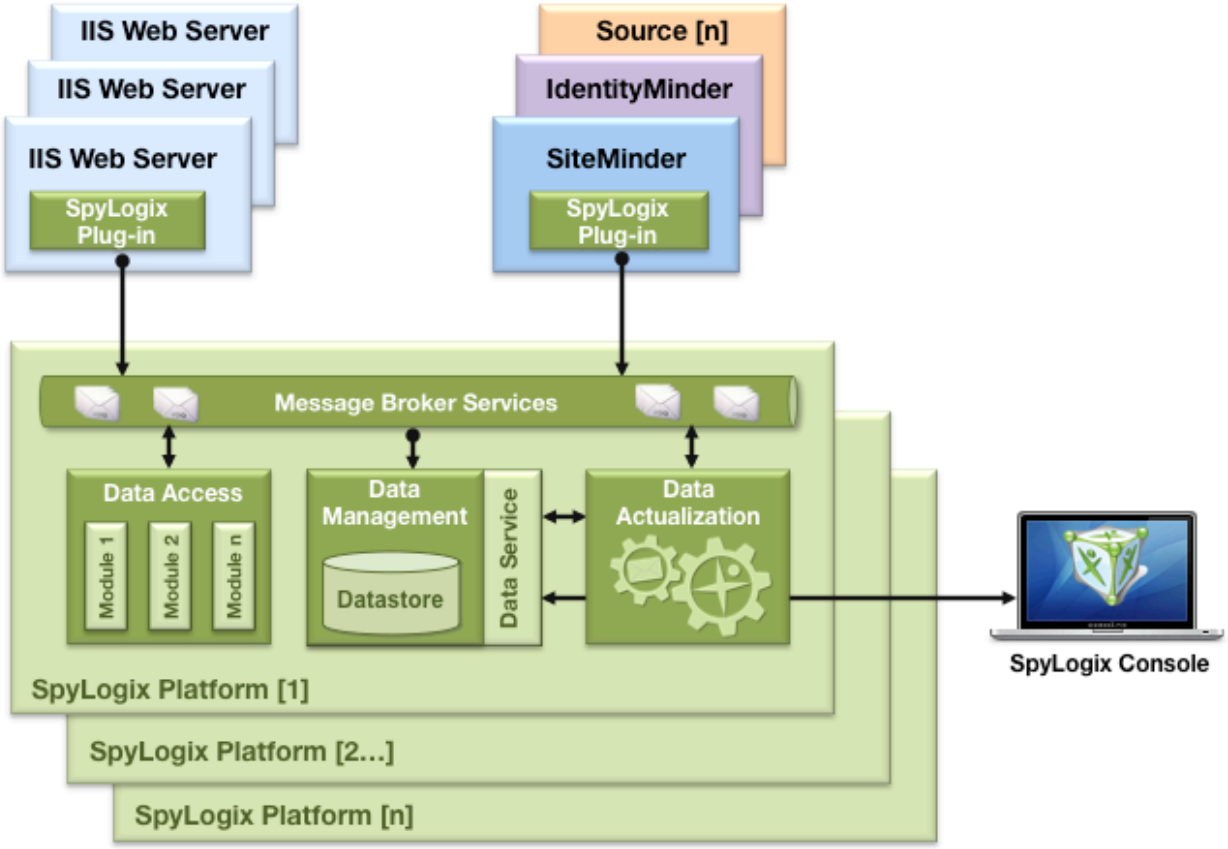

Figure 3 SpyLogix for IIS Deployment

IIS NATIVE DATA ACCESS

HttpApplication Class defines the methods, properties, and events that are common to all application objects in an ASP.NET application. This class is the base class for applications that are defined by the user. SpyLogix for IIS exposes HttpApplication Events as follows:

- BeginRequest Event occurs as the first event in the HTTP pipeline chain of execution when ASP.NET responds to a request. This event signals the creation of any given new request. This event is always raised and is always the first event to occur during the processing of a request.
- EndRequest Event occurs as the last event in the HTTP pipeline chain of execution when ASP.NET responds to a request.
- AuthenticateRequest Event occurs when a security module has established the identity of the user. This event signals that the configured authentication mechanism has authenticated the current request. Subscribing to the AuthenticateRequest event ensures that the request will be authenticated before processing the attached module or event handler.
- AuthorizeRequest Event Occurs when a security module has verified user authorization. This event signals that ASP.NET has authorized the current request. Subscribing to the AuthorizeRequest event ensures that the request will be authenticated and authorized before processing the attached module or event handler.
- LogRequest Event occurs just before ASP.NET performs any logging for the current request. This event is raised even if an error occurs.

Note that the BeginRequest and EndRequest events are only two of the events that occur during page processing. For more information about the events raised while processing a page, see Server Event Handling in ASP.NET Web Pages.

SpyLogix for IIS provides an event handler to for each of the aforementioned events and enumerates data from user tracking of access requests as follows:

1. HttpContext Class encapsulates all HTTP-specific information about an individual HTTP request. Properties implemented include:
    a. Request gets the HttpRequest object for the current HTTP request.
2. HttpRequest Class enables ASP.NET to read the HTTP values sent by a client during a web request. HttpRequest Class properties implemented include:
    a. Headers gets a collection of HTTP headers.
    b. LogonUserIdentity gets the WindowsIdentity type for the current user. Properties implemented include:
        i. Name gets the user's Windows logon name (overrides ClaimsIdentity.Name).
    c. Request.UserHostAddress gets the IP host address of the remote client.

SpyLogix for IIS module includes an event handler written as a private method. When the registered events are raised, ASP.NET calls the appropriate SpyLogix for IIS event handler method in the module, which builds a well-formed standardized SpyLogix Message and sends it to SpyLogix Platform using an industry-standard protocol.

Since the SpyLogix for IIS event handler would be a shared component between two or more applications on a Web server, this component is deployed to the server's computer-wide code cache, called the global assembly cache (GAC). Each computer on which the common language runtime is installed has a GAC. For more information, see the "Global Assembly Cache" topic in .NET Framework Help.