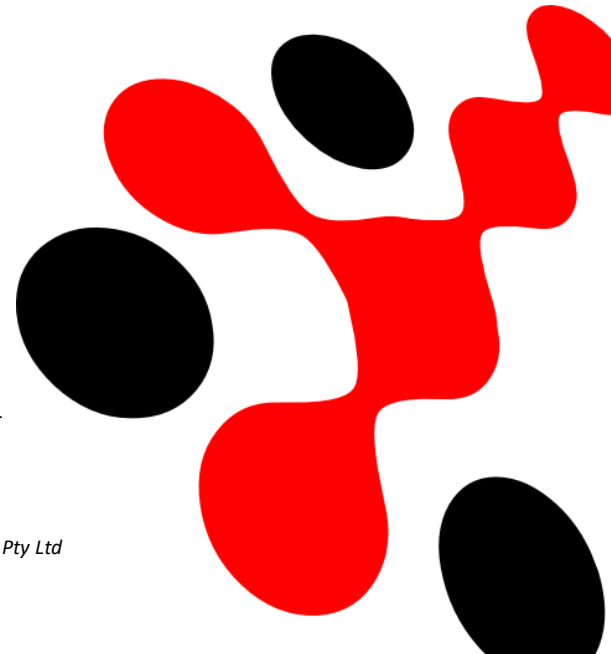
Weave Business Integration Framework

Architectural Overview

August 2020



Cohga Pty Ltd - Suite 405, 443 Little Collins Street, Melbourne Vic 3000 - P. 03 9036 3711

www.cohga.com - ABN 30 121 137 808

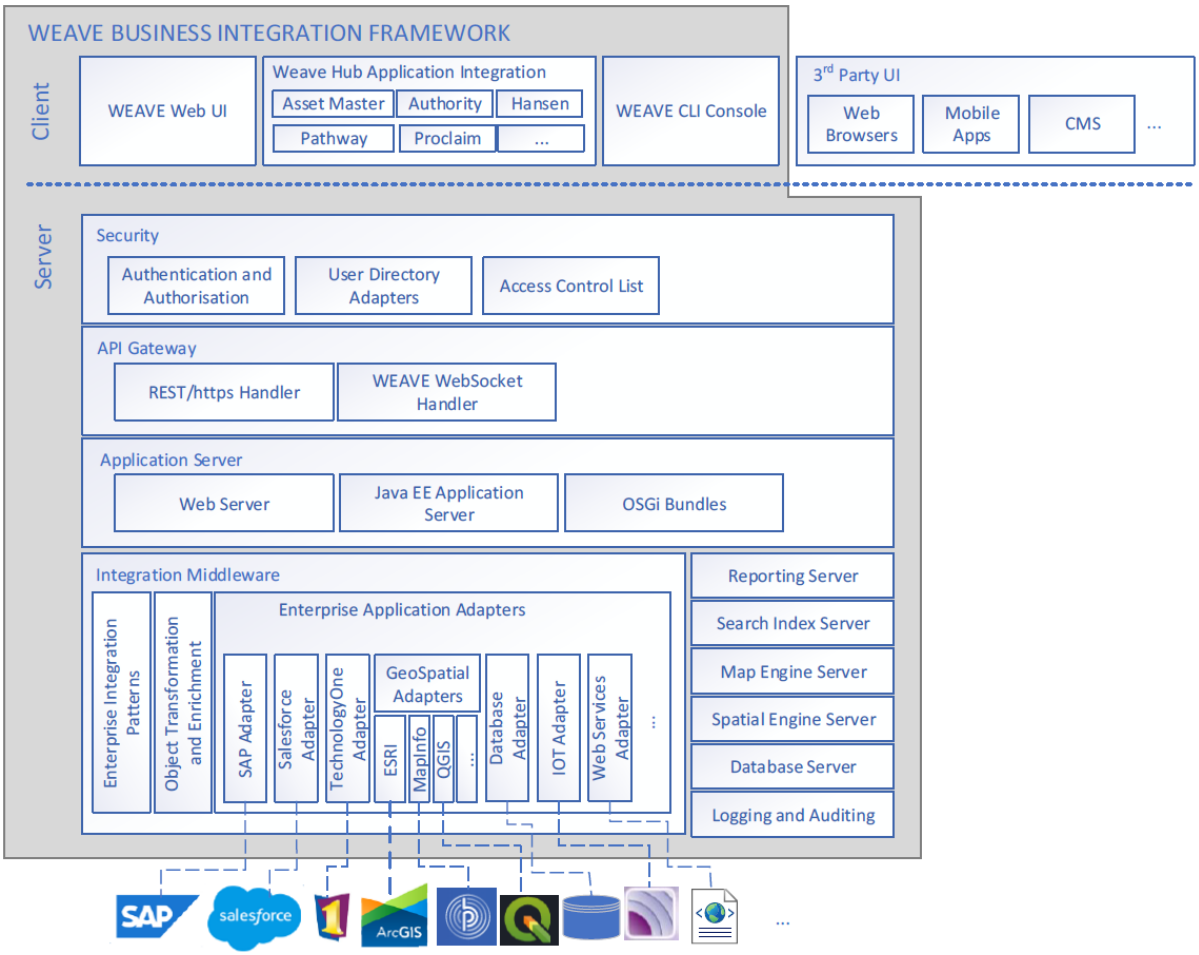
WEAVE BUSINESS INTEGRATION FRAMEWORK

Weave Business Integration Framework (WBIF) is a full-stack computer software with User Interface, Application Middleware, Integration Middleware and Database Server as its main Architecture blocks.

Investment over 13 years has yielded well-developed Application Integration capabilities featuring a collection of Enterprise Application Adapter assets to render automated and extensible cross IT systems connectivity. It is also equipped with supporting Enterprise IT functions like Reporting, Search, Logging and Auditing in conjunction with Mapping and Spatial functions to meet specific customers' requirements in selected domains.

ARCHITECTURE OVERVIEW

The following diagram depicts the Architecture Overview of Weave Business Integration Framework (WBIF).



As depicted in the picture WBIF is a Full stack Java EE application built on OSGi Framework, whose front-end and back-end components are highly extensible to meet the need of specific functions required by customers from various Industries and Services.

Out-of-the-box WBIF provides rich Enterprise Integration capabilities. The traditional implementation model is for Cohga consultants to conduct Requirements discovery and gathering, which includes identification of source systems to be integrated using WBIF. Based on the specific data sources, the Enterprise Application Adapters are either configured (for those existing adapters) or custom adapters are built for those data sources without existing adapters.

The full stack can be classified on a high-level Architecture perspective as Client Architecture and Server Architecture.

Client Architecture – consists of various technology serving different functions to end-users. A client component commonly used both for administration and daily operations is a web-browser based client. It supports most major modern web browsers like Internet Explorer, Google Chrome and Mozilla Firefox. There is also a client Application Integration Module (client-side integration) used to integrate with applications within Asset Management, Property Management, Document Management and Geospatial applications. The last component is the Command Line Interface console which is an alternative front-end for administration functions. The Client Architecture is extensible for integration with 3rd party clients using REST APIs exposed by the API Gateway component of the Server.

Server Architecture – built on the Java EE specification and the OSGi framework, providing extensive functionality across application and data integration based on Apache Camel. The Server Architecture is traditional enterprise grade layered architecture which includes Security, API Gateway, Application Server, Integration Middleware, Database Server, plus function specific servers like Reporting, Search, Map Engine, Spatial Engine and Logging-and-Auditing.

DESCRIPTION OF COMPONENTS

The following table provides a description of the Client and Server components of WBIF.

No	Component Name	Description	Technology
Client Architecture			
1.	WEAVE Web UI	Common User Interface of WBIF. It provides both administration and operational functions controlled by Application Security components (see below). The user interface framework is based on Sencha Ext JS which specialises in visualisation of Data Intensive application using HTML5.	• Major Web Browsers (Internet Explorer, Google Chrome and Mozilla Firefox). • Sencha Ext JS.
2.	Weave Hub Application Integration	Add-in components installed on the client machine to provide user-interface integration capability with selected enterprise application clients running on the client machine. Weave Hub Application Integration is the base framework, which is required as the main controller for application specific handlers. Some sample of application specific plug-ins are listed on the diagram, they are not the full list.	• Custom Build Plug-in framework based on Microsoft .Net Framework.
3.	WEAVE CLI Console	Command-line user-interface for Administration functions. This is an extension of OSGi Framework CLI.	• OSGi Framework
4.	3rd Party UI (not part of WBIF)	These components are not part of WBIF. They are included for completeness purpose.	• REST API

		WBIF provides Integration capabilities to custom 3rd party User Interfaces (e.g. Other websites, Mobile applications, Content Management Systems, etc.) by providing REST APIs.	
Server Architecture			
5.	Security	Security Layer is responsible for enforcing user authentication and command invocation authorisation of WBIF.	(see below)
5.1.	Authentication and Authorisation	Authentication and Authorisation Framework provide the basic capability of these security features. It is based on the Open Source Spring Security. It is configurable and flexible security framework that can be integrated to various user directory software.	• Spring Security
5.2.	User Directory Adapter	The Authentication and Authorisation Framework uses User Directory to authenticate user credentials and checking user (group) permissions to invoke protected resources. These User Directory can be one of Microsoft Active Directory, LDAP Compatible Directory or Database. The User Directory Adapter is configured to integrate to an enterprise User Directory	• Spring Security
5.3.	Access Control List	Access Control List (ACL) defines the permission of user-groups to access protected objects. The user-group is based on the User Directory discussed in the previous component.	• Spring Security
6.	API Gateway	Component managing interfaces provided by WBIF Server Components. There are 2 major types of Interface exposed on the gateway as discussed below.	(see below)
6.1.	REST/https Handler	Managing the requests and responses of REST interfaces exposed over https protocol.	• Apache Camel • Swagger (OpenAPI)
6.2.	WEAVE WebSocket Handler	The WebSocket protocol enables interaction between WEAVE Web UI and the WBIF web server with lower overhead than half-duplex alternatives such as HTTP polling, facilitating real-time data transfer from and to the server. This is used as control and communication mechanism to the Server regarding the local interaction between WEAVE Web UI and Weave Hub Application Integration plug-ins.	• Eclipse Jetty (default) - Option includes: Apache Tomcat and Glassfish.

7.	Application Server	The Application Server is the central middleware component managing client/server interactions as well as cross server modules realising application functionalities.	(see below)
7.1.	Web Server	Web Server is responsible to handle the http request received from clients and http response returned from the server. It manages the traffic by implementing standard http protocols.	• Eclipse Jetty (default) - Option includes: Apache Tomcat and Glassfish.
7.2.	Java EE Application Server	The Java EE Application Server is responsible to manage the components within Java EE platforms that are the foundation Services to provide functional and non-functional Services of WBIF. Only Java Servlet specifications are used on WBIF technology stack. ”Java EE” is the new name of what previously known as “J2EE” (and some of their other variants, e.g. J2SE, J2ME, etc.)	• Eclipse Jetty (default) - Option includes: Apache Tomcat and Glassfish.
7.3.	OSGi Bundles	OSGi provides an extensible and modular framework for Java EE application development. The major components responsible for extensibility are components called “Bundles”, they are modular user-defined custom components build to deliver specific functions. WBIF uses these Bundles to deliver its modular functions.	• OSGi
8.	Integration Middleware	As an Integration Framework, the Integration Middleware plays a crucial role in the overall WBIF stack. It provides capability to process incoming data to WBIF as well as distributing data to external systems using well-defined interfaces. This component is based on Apache Camel implemented as extensible components to cater for various integration sources.	(See below)
8.1.	Enterprise Integration Patterns	“Enterprise Integration Patterns (EIP)” is a collection of well-defined integration patterns that are foundation to IT Systems Integration as defined in the book of the same title. WBIF builds its integration capabilities based on these patterns, e.g. Message Router, Message Translator, Request Reply, CorrelationID, etc. Apache Camel is the middleware providing	• Apache Camel

		foundation capabilities to realise these patterns.	
8.2.	Object Transformation and Enrichment	Although this arguably is overall part of EIP capabilities, it needs a special mention as it is a key foundation capability of WBIF in processing (transforming and enriching) Business Objects it receives from and distributes to its many integrated source systems.	• Apache Camel
8.3.	Enterprise Application Adapters	Enterprise Application Adapters are custom adapters built to handle Business Objects and functional processes specific to an Enterprise Application. WBIF has extensive application adapters that were built over time, some of them are depicted on the diagram (not a comprehensive list). Among them are application of the types: Enterprise Resource Planning (SAP), Customer Relationship Management (Salesforce), Asset Management (TechnologyOne), Document Management (Trim), GeoSpatial and Mapping (ArcGIS, MapInfo), as well as technology adapters like Database Adapters, IoT Adapters as well as Web Services Adapter.	• Custom build based on Adapter Framework using Camel components
9.	Reporting Server	Reporting capability is a key function of WBIF. It provides a flexible and customisable reporting capability based on different business objects managed within WBIF. As WBIF can handle many different Business Object types (as the benefit of flexible metadata-based Data Architecture- see below), the reports produced are driven from these object types. WBIF is compatible with different Reporting systems, e.g. BIRT, MS SQL Server Reporting and SAP Business Objects.	• BIRT Reporting • MS SQL Server Reporting • SAP Business Objects.
10.	Search Index Server	WBIF offers extensive 'google-like' search capability across both its native Business Objects as well as ingested Business Objects from external sources. The search capability is based on Apache Lucene Search Index Server.	• Apache Lucene
11.	Map Engine Server	Map Engine Server is responsible to process mapping data including generation of User Interface containing map images. WBIF is compatible with many industry	• WMS • ArcIMS • ArgGIS • Weave Map Engine

		standard map engines, e.g. WMS, ArcIMS, ArcGIS, etc.	
12.	Spatial Engine Server	A Spatial Engine provides access to a service that can perform spatial operations, such as determining what entities fall within a polygon, but it also provides a spatial perspective to the entities that that the system will use. For example, providing details about the coordinate reference system that the entity is stored in. This will generally be something like ArcSDE, Oracle Spatial, etc. At least one will be defined.	• ArcSDE • Oracle Spatial • SQLServer Spatial • PostGIS
13.	Database Server	Database Server is the data store for persisted WBIF data. Various database servers are supported by WBIF, i.e. PostgreSQL, MySQL, Oracle, Microsoft SQL Server, IBM DB2.	• PostgreSQL, • MySQL, • Oracle - Microsoft SQL Server • IBM DB2.
14.	Logging and Auditing	Transactions across WBIF can be logged into system log files. These files are useful for Business and Technical auditing, as well as for application troubleshooting. The level of logging is configurable, depends on requirements and running environments. These capabilities are built upon Log4J framework.	• Log4J