

Trusted Internet Application Framework



A framework is a software environment that is defined to simplify application development and system management for a specialized application domain.

1. Introduction

Security ...

The Trusted Internet Application Framework from Ergon provides a secure environment for applications in the inter- and intranet. For all who write and deploy Web-based front-ends, this framework provides more security and flexibility than any other commercial solution on the market today.

It offers protection from Internet-based attacks as well as from Web server and application errors. With elaborate defenses built into the trusted operating system, the Trusted Internet Application Framework provides protection of a company's internal network, while securing the business transactions.

... and Flexibility

Trusted Internet Application Framework, the integrated session manager and the WebEval engine with its scripting language, allows the developer to build trusted Internet applications faster than anyone else. The framework has been proven within a growing number of business projects: *Direct Net*, the first Internet-banking solution in Switzerland of CREDIT SUISSE, *Paynet*, an Internet-Payment solution of Telekurs and many more.

2. Why you need security and flexibility in the Internet

The Internet offers incredible commercial opportunities but represents also a major security challenge. Any process that allows customers to connect to your internal systems also opens a window to sensitive, mission-critical data and applications.

Firewalls protect the internal network from attacks through known network protocols, but they do not offer any protection from attacks on the gateway platform being accessed from the Intranet. Bugs in the operating system, in the Web server software or in the applications that access your internal data expose your internal network to attacks.

The Trusted Internet Application Framework strengthens the gateway platform by running the Web server and the Web applications in distinct environments to minimize any damage caused by an application error. Its underlying trusted operating system enhancement from Argus surpasses B1 specifications and has been tested in the toughest defense systems and approved by ITSEC.

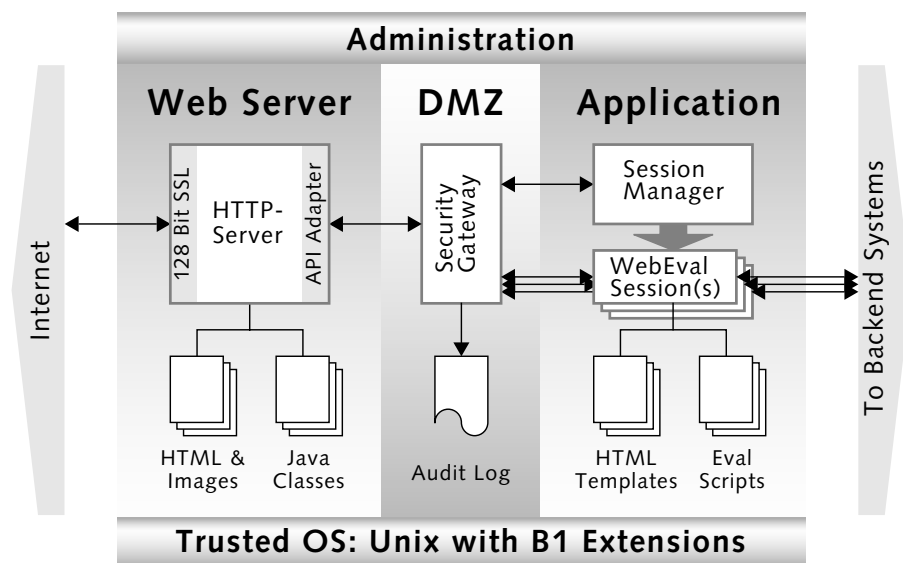
Due to its strengths, it has been chosen as the security solution by the first bank to operate on the Internet, CREDIT SUISSE.

On the other hand, cyber business needs cyber speed. Traditional languages like C, C++ or perl take too much time to develop an application. This framework provides you with the right tools to launch an application even before others start to think about it. WebEval with its fully integrated scripting language allows you to write and test your programs on the fly without recompiling.

Any backend system or database can be integrated with CORBA, DCE, SQL, or SNA LU2. You can even plug in your own adapters as loadable modules to WebEval and access them immediately from the scripting language.

3. Architecture

The trusted operating system replaces the super-user scheme and augments the Unix protection mechanism. The new security mechanisms allow the Web server, the WebEval sessions and its scripts to be run in separate environments.



The Trusted Internet Application Framework uses a Netscape Enterprise (US) or a Stronghold Web server, which communicates using the Secure Sockets Layer (SSL) security protocol. The server receives the customer request and tries to satisfy it by invoking a WebEval application script which delivers data in the proper format back to the browser. Due to security reasons this is not done with the Common Gateway Interface (CGI) like other products do, but with an API adapter which connects to the security gateway.

The security gateway is a small, trusted program which has been assigned the privileges needed to "cross the boundaries" between the Web server and the WebEval application servers. It runs in its own restricted, safe environment and approves the request before connecting to the current WebEval application session. It performs critical security functions before connecting to the WebEval session:

- Checks that the URL through which the script is addressed is "approved".
- Assigns privileges needed to connect to the WebEval application session in its own restricted environment.
- Passes the arguments of the request from the Web server to the WebEval session and the result of the request back to the Web server.

- Writes every request with its arguments to an audit log.

The WebEval application session executes the corresponding script called by the request and interacts with other servers on the internal network to retrieve information.

The security concept of Trusted Internet Application Framework relies upon the following features provided by the *Argus Security Extensions* to Solaris:

- Partitioning the system by using *MAC Security Labels (SL)*
- Disabling the UNIX super-user (root), splitting the "root right" into multiple *privileges* (which can be assigned & used separately)
- Having separate administrative *roles* (e.g. ISSO, SA, SO) for administering the system
- Secure boot and trusted recovery using a file integrity database

In addition to the isolation provided by the partitioning, the session manager assigns different user IDs to the separate WebEval application session processes (one per active session) to further isolate them (and their files produced) from each other.

An intruder breaking into the Web server or the network interface from the Internet cannot get to the application, to the internal network or other machines inside. Since there is no root account to break into, a penetration into the Web server does not compromise the security of the machine on which it is running.

Even if an intruder could subvert the Web server, he cannot plant an unauthorized CGI program for the Web server to execute. If he tries tricking the Web server into running a program, the program has no access to any privileges and cannot access the application, its files, databases or the inside network.

On the other hand, if the intruder could somehow trick one of the WebEval application session into starting a program, that program has no privileges and can neither access other application sessions, nor their data on the Internet.

4. Session management

The Session Manager allows you to maintain persistent Internet sessions for conducting real-time transactions over the Web. Persistent sessions are necessary for extending client/server systems to the Web and for conducting electronic transactions that are safe and reliable.

Due to the fact that HTTP is a stateless protocol, we lack a universal way of uniquely identifying a client and its application context. In normal environments, developers must solve this programmatically with session IDs and cookies, but in the Trusted Internet Application Framework this functionality is managed automatically by the Session Manager.

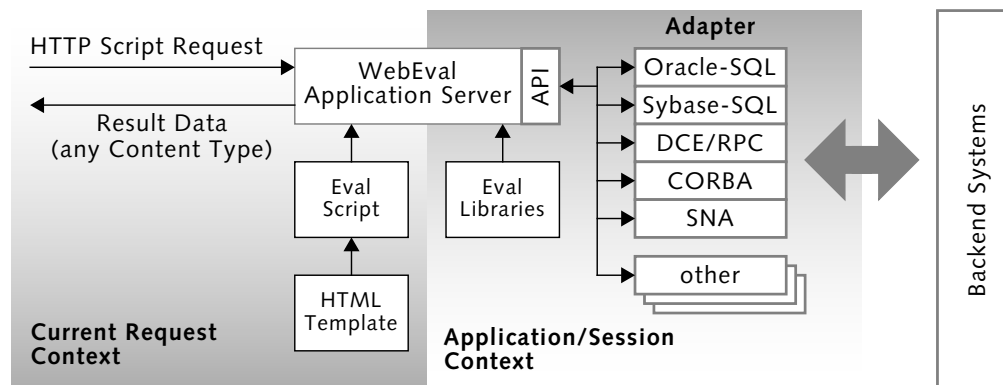
For each session the Session Manager starts an instance of the WebEval application server and assigns a unique session ID. Among other information, this session ID is stored in a session cookie, so the Security Gateway is able to identify the client and connect to the right application process. To avoid hacker attacks with cookie spoofing, the information in the cookie is encrypted in a safe way.

5. WebEval Application Server

This is where we start to differentiate ourselves. In a standard environment you have two basic choices for developing an application: you can either use CGI-Scripts or you can implement your application in C or C++.

The advantage of CGI-Scripts is a very short development cycle (write test, write ...), but for large systems, the overhead of spawning new application processes for each request becomes too high.

Writing the application with C or C++ gives you flexibility but makes the development slow and might create an inherently unstable system. Locating an error with a debugger is nearly impossible and is like searching for a needle in the hay stack.



The answer to this dilemma is quite simple: An extensible persistent Script-Interpreter for each application session, where the interpreter runs in a separate persistent process instance and the Session Manager provides the basic glue, resulting in: Short development cycle and fast execution.

Eval is an interpreted language optimized for developing Web-based applications. The language is intended to be easy to use, efficient, elegant and easy to extend. It combines some of the best features of C, Java, and Perl, so people familiar with one of those languages should have no difficulty using it. Expression syntax corresponds quite closely to the C expression syntax.

Unlike most other utilities, Eval does not arbitrarily limit the size of your data. If you've got the memory, Eval can read in your whole database as a single string. Recursion is of unlimited depth. And the associative arrays grow as necessary to prevent degraded performance. You don't have to take care of allocating and freeing memory, the built-in garbage collector will do the job for you.

Eval Libraries allow you to share common code and with the Eval-API you can write your own functions in C or C++, load them at runtime into your WebEval Application Server and access them directly from your Eval-Scripts.

The WebEval Application Server ships with a set of adapters which allow you to access several backend systems: Oracle and Sybase Databases, DCE/RPC, SNA and CORBA.