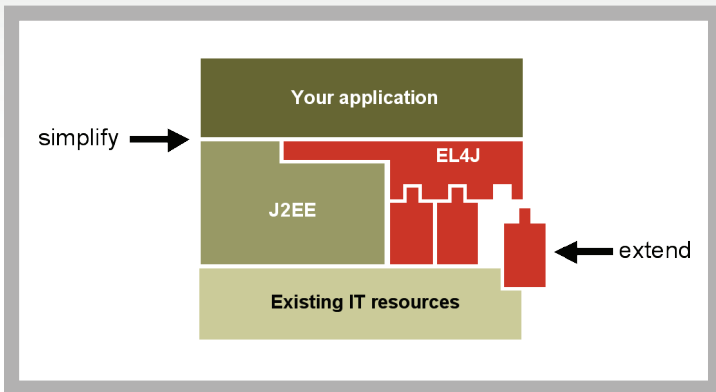


What is EL4J?

EL4J is an open source framework that simplifies and speeds up the development of J2EE applications. At its core, **EL4J uses the popular Spring Framework**. It adds other leading open frameworks and extends them with samples and improvements. J2EE is a very powerful development platform, with an enormous collection of tools and frameworks. Unfortunately, the price to pay for this vast set of facilities is complexity. **EL4J makes it easy to use best-of-breed J2EE technologies**.



Basic idea: simplify and extend the J2EE.

What are the benefits of EL4J

Developing J2EE applications with EL4J has the following benefits (compared to development with pure J2EE or Spring):

- Developers are instantly productive due to pre-existing technology choices, a model architecture, and **application templates**. Design debate is reduced to the essential.
- Go with the **winning technologies**: the included frameworks are de-facto standards. Profit from the **managed evolution** of EL4J.
- The **lightweight approach** let's you get to the point of your problem immediately. You work directly with **POJOs** (Plain Old Java Objects), leading to fast development round-trips, simplified tests, and minimal technology lock-in.
- Effectively **separate business concerns** – developed with standard Java technology – from technical concerns – solved orthogonally via interceptors and other AOP mechanisms.

We originally created EL4J for ourselves, to support internal and external projects. Within ELCA, **it has rapidly become the standard approach for J2EE enterprise applications**. EL4J has been used in more than 15 projects.

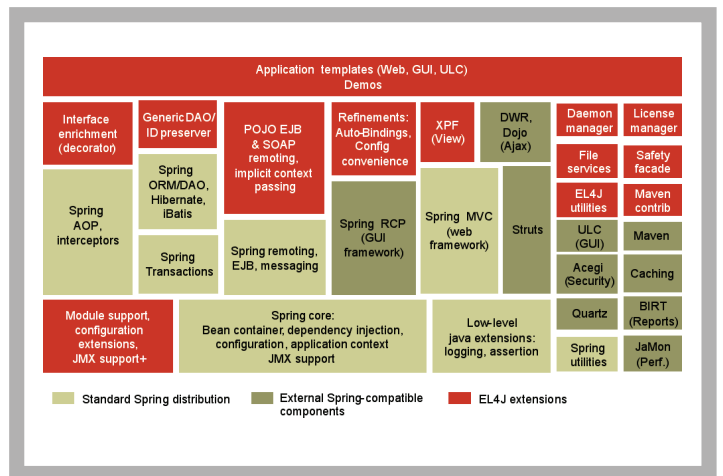
What are the parts of EL4J?

The following standard components are part of EL4J:

- Spring Java Framework 2.0
- Hibernate or iBatis
- Maven 2.0, JUnit, JWebUnit, Commons logging, and log4j

- The Spring Security (Acegi) Framework
- Spring Rich Client Platform (Spring RCP)

In the following we describe the features of EL4J with all included components.



The components that make up EL4J.

What are the features of EL4J?

Transparent remoting: All client-server interactions between logical tiers are modeled as method invocations on normal POJOs. POJOs can be either co-located with the client, or deployed remotely in the EJB or Web server or in a standalone application, without changing the code.

Advantages: Simpler development, more deployment flexibility, and simple interoperability with SOAP, CORBA or .NET.

Extensibility and adaptability: For today's fast pace of change, Spring incorporates several extension and adaptation mechanisms. EL4J complements them where appropriate:

- POJOs can be added, changed, and removed conveniently via **configuration**, by simply adapting a running default configuration. A service is automatically enabled when you add its jar file to the class path; there is no further need for configuration.

- **Interceptors** can complement the behavior of existing services. There are predefined interceptors (e.g. for authorization, tracing or statistics collection), and you can add your own. Interceptors can even be added at run-time, allowing you to intervene in a live application, e.g. for performance measurements.
- **Optional context information** (for example the security principal, or the transactional context) can be **passed implicitly with remote method invocations**, keeping such data out of your business interfaces. This allows also sharing a same application instance between multiple organizations, by passing the principal of the concerned organization with each request.

These features keep the core of EL4J simple, while providing far-going extension potential and giving more power to the developer. **Advantages:** Protection against change of external standards and implementation, as well as improved system agility.

Modular build system: Through the abstractions of Maven2, the EL4J code is **split into modules** that can be combined flexibly; each distribution includes only the essential modules. **Dependencies between modules are transitive:** This means you import automatically all the modules of your parent modules. Settings for your environment (database, application server, etc.) are abstracted. We provide seamless integration with Eclipse and JUnit.

Advantages: Modularized and leaner applications, better support for multiple projects with differing needs and lifecycles.

Security service: For a more flexible and unified model than JAAS, EL4J integrates the **Acegi security framework**. Its security model works for the web, simple domain objects, and multi-tier applications that span multiple processes.

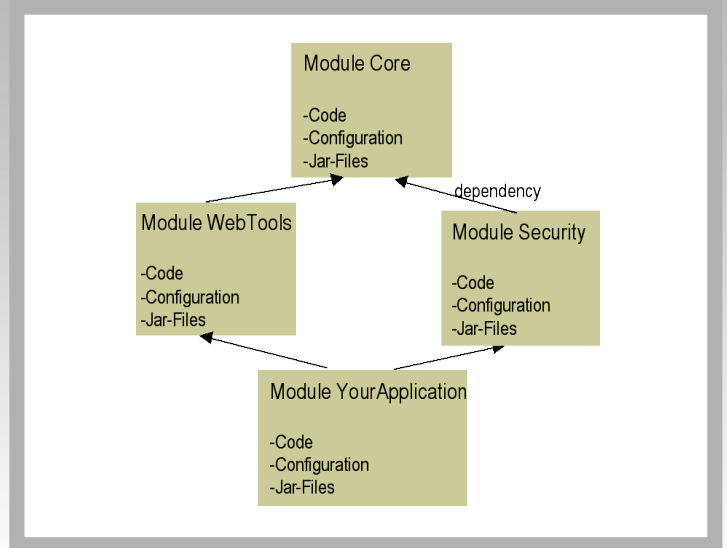
Advantages: A more flexible and unified security model. Shared security infrastructure reduces operational costs and minimizes the risk of inconsistencies.

Web applications: EL4J recommends the proven **Spring MVC** framework. It includes an end-to-end template application that helps getting started quickly. **Struts** is an alternative framework when the context demands it.

Advantages: Speed up web developments, promotes use of current best practices.

GUI applications: For quick development of professional Swing applications and to adhere to the best practices, EL4J uses the **Spring RCP** framework. EL4J extends and packages the Spring RCP and provides a template application. Your GUI only needs to declare where its canvas is different from a default GUI canvas. Then you add your custom dialogs and views, mostly by extending existing components.

Advantages: Get up to speed quickly with robust and good looking Swing applications, profit from current best practices.



Module with dependencies. For configuration convenience, adding a module dependency automatically sets its features up.

Daemon Manager: The daemon manager **executes and supervises multiple long-running daemons** (supervised Java threads). The JVM with the daemon manager is typically launched as an operating system service and can be managed from remote.

Advantages: Implement very reliable daemons with Java and administrate them from remote.

Other features: Other components complement EL4J with reporting, support for publishing the current configuration or other settings in JMX, persistent task scheduling, file operations, a license manager, caching, support for generic DAOs, or with XML merging.

Advantages: Ready to use, well-integrated components for various domains cut the time you need to get started.

More information

http://www.elca.ch/Solutions/Technology_Frameworks/EL4J/EL4J.php

Most of our extensions to Spring have been published in the open source under <http://EL4J.sourceforge.net>. Distinctive EL4J features are under <http://el4j.sourceforge.net/docs/pdf/UniqueEL4JFeatures.pdf>

Professional support in English, German and French is available.

With EL4NET and IIOP.NET ELCA provides similar projects.

IT-Solutions by ELCA

We make it work.

ELCA
Steinstrasse 21
8003 Zürich
Schweiz
Phone +41 44 / 456 32 11
Fax +41 44 / 456 32 00
info@elca.ch
www.elca.ch