

# Eclipse Scout

## Entwicklung von Geschäftsanwendungen

**Mit Eclipse Scout wird für die Entwicklung von Client/Serverapplikationen erheblich weniger Zeit benötigt. Scout-Applikationen sind zu 100% SOA-konform und in reinem Java geschrieben. Eclipse Scout hat sich auch in der Praxis bewährt: Scout Anwendungen sind seit Jahren weltweit bei tausenden von Anwendern im täglichen Einsatz.**



**Simple. Stable. Flexible.**

### **Einfach.**

Eclipse Scout ist für Java-Entwickler geschaffen. Nach nur drei Tagen Schulung können sie eigene Business-Applikationen erstellen. Für die Arbeit mit Scout SDK ist nur ein allgemeines Verständnis der zugrunde liegenden Eclipse/OSGi/J2EE-Technologien erforderlich.

### **Stabil.**

Scout-Applikationen laufen in allen gängigen Umgebungen stabil. Das Framework bietet ein breites Spektrum an sofort einsatzbereiten Funktionen an. Der Zugriff auf relationale Datenbanken und alternative Persistenzschichten ist konzeptionell sehr einfach gehalten. Auch die Bereitstellung und Nutzung von Webservices wird direkt durch das Framework unterstützt. Je nach Kundenanforderungen kann der Entwickler zwischen SWT oder Swing als UI-Technologie wählen. Scout unterstützt beide. In naher Zukunft wird neben SWT und Swing zusätzlich das Web als alternative UI Technologie unterstützt.

### **Flexibel.**

Scout-Applikationen können leicht über bereits vorhandenen Services oder existierende Kundensysteme angebunden werden. Das Framework basiert auf Standard-Eclipse Technologien wie OSGi-Services und Extension Points, die den Austausch oder die Erstellung eigener Komponenten ermöglichen. Um vollständige Flexibilität zu bieten, nutzt Scout nur POJOs (Plain Old Java Objects, ganz normale Java-Objekte) und arbeitet ohne Meta-Modell.

# Scout Runtime

## Services nutzen

**Eine dienstorientierte Architektur (SOA) ermöglicht eine Umgebung von lose gekoppelten Systemen. Eclipse Scout-Applikationen lassen sich gut in heterogenen Landschaften integrieren.**

Unterschiedliche Anforderungen und Freigabezyklen machen die Kommunikation zwischen verschiedenen Systemen fehleranfällig. Die Kommunikation über HTTP und SOAP entkoppelt solche Systeme. Ein Scout-Server kann in so einer Umgebung Webservices bereitstellen und nutzen. Darüber hinaus wird die Kommunikation zwischen Scout-Client und Scout-Server auf die gleiche Weise modelliert. Auf der Seitenleiste gibt es hierzu weitere Anmerkungen.

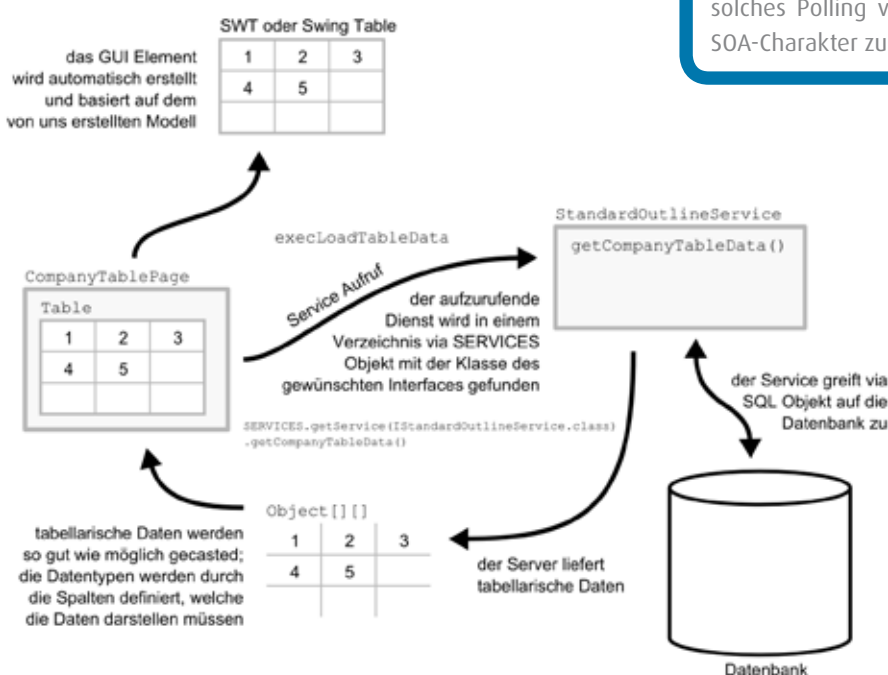
Jeder verfügbare Dienst wird durch einen Proxy auf dem Scout-Client repräsentiert und in einem lokalen Verzeichnis registriert, wie im Diagramm unten gezeigt. Der Benutzer findet den geeigneten Proxy durch Verwendung der Service-Schnittstellenklasse und verwendet die Methoden auf dem Proxy. Eclipse Scout serialisiert die Funktionsaufrufe

### Die Realität: Performance und Komplexität

Standardmässig erfolgen alle Interaktionen mit Datenquellen über Service-Aufrufe. Da SOAP-Services reichlich Bandbreite benötigen, verwendet Eclipse Scout standardmässig serialisierte und MIME-codierte Java-Objekte. Bei Bedarf können Entwickler von externen Clients einzelne Services auf reguläres SOAP-Format umschalten.

Beim Laden und Speichern von Formularen sendet Eclipse Scout typisierte Datenobjekte hin und her. Wenn es sich jedoch um tabellarische Daten handelt, nimmt Eclipse Scout an, dass die gelieferten Daten praktisch immer durch Tabellen angezeigt werden können und verwendet ein Array von Objekt-Arrays, um dem Entwickler den Programmieraufwand für das Packen und Entpacken von tabellarischen Daten zu sparen.

Eine interaktive Applikation darf aus Zeitgründen nicht für jede Interaktion mit der Datenquelle eine neue HTTP-Anfrage stellen. Deswegen halten alle Clients auf dem Server eine dauerhafte HTTP-Sitzung offen und erneuern diese bei einer Zeitüberschreitung. Ein solches Polling verringert Verzögerungen, ohne den SOA-Charakter zu beeinträchtigen.



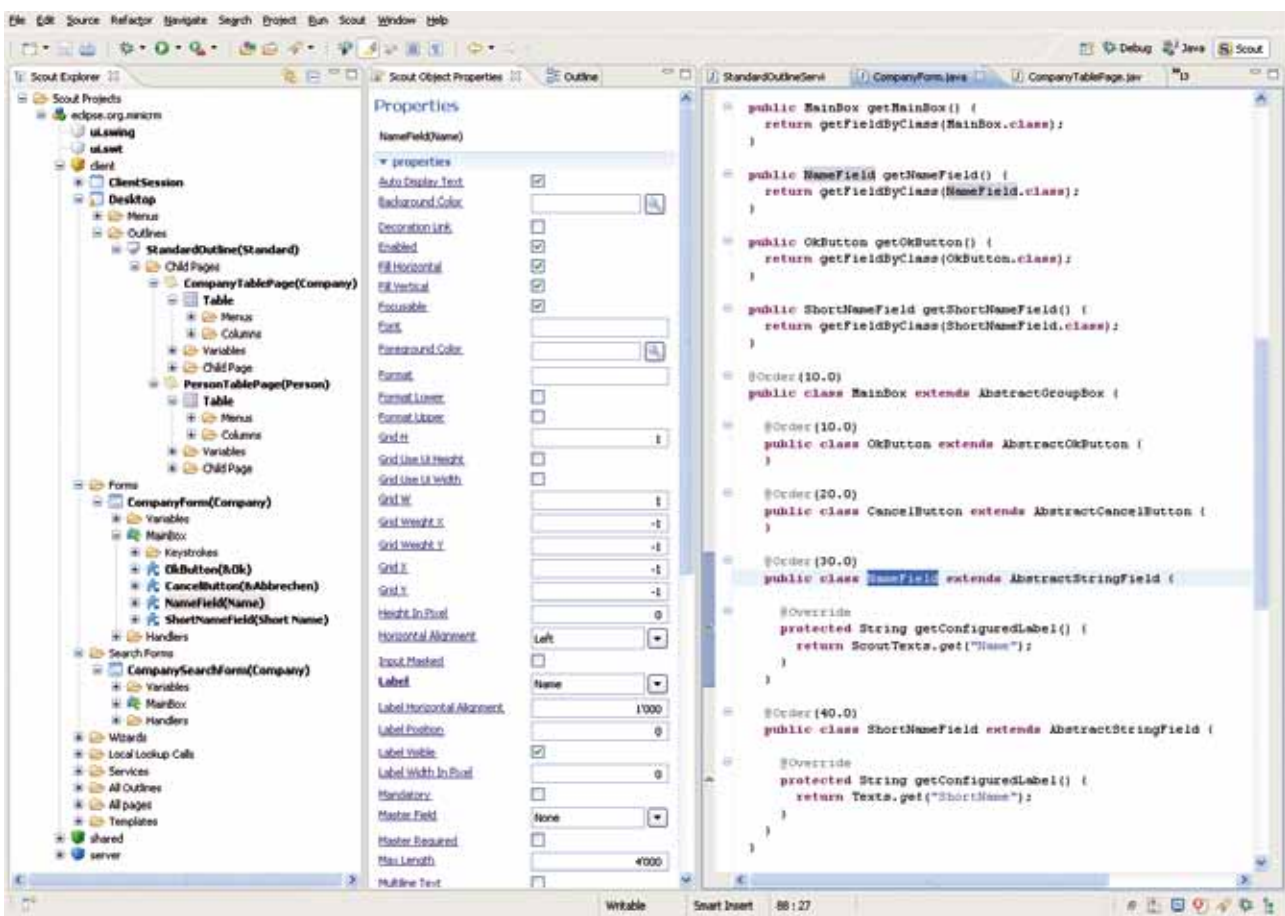
# Scout SDK

## Vorteile durch Eclipse

**Das Scout SDK beinhaltet als zentrales Element die Scout-Perspektive, welche den Entwickler beim Erstellen einer Applikation unterstützt. Das Anlegen und Anpassen von Formularen, Tabellarischen Übersichten, Services und anderen Applikationselementen wird durch Scout SDK massgeblich vereinfacht.**

Ein grosser Vorteil von Scout SDK besteht darin, dass keine Meta-Daten in externen Repositories erforderlich sind. Das Scout SDK analysiert alle Java-Quelldateien im Hintergrund und bewirkt so, dass der Quellcode und das Applikationsmodell immer synchron bleiben. Daher können Entwickler völlig frei zwischen der Scout- und Java-Perspektive hin- und herwechseln.

Eine Scout-Applikation wird in einem Satz von POJOs gespeichert. Beispielsweise ist eine Tabellenseite eine Klasse, deren Tabelle eine innere Klasse ist. Die Tabellenspalten sind ebenfalls innere Klassen und ihre Eigenschaften sind Methodenaufrufe. Genauso ist ein Formular eine Klasse, Felder sind innere Klassen und Feldeigenschaften sind als Methodenaufrufe realisiert. Diese Architektur wird von Scout SDK genutzt, wie unten im Scout Explorer oder in der Ansicht «Scout Object Properties» (Scout-Objekteigenschaften) gezeigt.



Dieser Screenshot des Scout SDK zeigt, wie ein Dialogformular erstellt wird. Die Spezifikationen des im Scout Explorer markierten NameField werden detailliert in der Ansicht «Scout Object Properties» aufgelistet und als Java-Code in der rechten Spalte angezeigt.

# Anwendungsfall BSI CRM

## Sich auf Produktionsqualität verlassen

**Wir glauben an Eclipse Scout, weil wir seine Vorzüge jeden Tag erleben.**

Wir bei BSI verlassen uns auf die Produktionsqualität des Scout-Frameworks. Zuerst erfassen wir die Anforderungen in Workshops zusammen mit unseren Kunden und dann erstellen wir mit Hilfe von Dummy-Services und Testdaten Prototypen, um die Ideen mit unseren Kunden zu validieren. Im Laufe des Projekts erarbeiten wir die Detailspezifikation und verfeinern die Software bis zur endgültigen Kundenapplikation.

Benutzer wählt eine Sicht

jede Sicht hat diverse Unterseiten

eine Seite mit Tabelle kann einen dazugehörigen Suchdialog haben

normalerweise wird die erste sichtbare Spalte als Namen des Knoten im Baum verwendet

jeder Knoten kann ein Kontextmenü haben

das Kontextmenü erstellt einen Dialog und übergibt einen Primärschlüssel

die Firmenseite holt tabellarische Daten vom Server und stellt sie dar

der Suchdialog hat diverse Felder, welche an den Dienst weiter gegeben werden; aus diesen Werten kann beispielsweise eine WHERE Klausel zusammengebaut werden

der Dialog verwendet den erhaltenen Primärschlüssel, um die nötigen Daten von einem Prozessdienst zu laden

Konkretes Beispiel: Das BSI CRM verwendet das Swing user interface und eine Oracle Datenbank. Die Dialoge bestehen nicht nur aus einfachen Textfeldern

**bsi**  
meet your clients

BSI CRM ist eine umfassende CRM-Suite für das Management von Kundenbeziehungen und wird hauptsächlich von Marketing- und Vertriebsabteilungen eingesetzt. Für die hohe Benutzerakzeptanz ist eine einfache und intuitive Benutzeroberfläche entscheidend.

### Drei Kernprodukte, über zwanzig Applikationen, mehr als zwanzigtausend Endbenutzer

BSIs Kernprodukte BSI CRM, BSI Contact, BSI CTMS und BSI Counter basieren auf dem Scout-Framework. Die Hauptvorteile von Scout liegen für uns bei der reduzierten Entwicklungszeit, der Nutzung von offener Software, der flexibler Integration in bestehende Softwareumgebungen sowie bei der schnellen Integration neuer Mitarbeiter in laufende Projekte.

**BSI** Business Systems Integration AG  
[www.bsiag.com/eclipse](http://www.bsiag.com/eclipse)  
[www.eclipse.org/scout](http://www.eclipse.org/scout)

**bsi**  
meet your clients