

## Whitepaper – edoras® suite

Die edoras® suite ist eine performante JEE Standard-Applikationsplattform für die Web- und Rich-Client-Softwareentwicklung. Die modulare Architektur stellt core components und composite applications zur Verfügung, mit denen die Entwicklungszeit verkürzt werden kann. Das Design einer Applikation basiert dabei auf einem übergreifenden Meta-Model.



mimacom ag  
Kirchbergstr. 107  
3400 Burgdorf  
Tel. +41 34 424 09 00  
Fax +41 34 424 09 01  
info@mimacom.ch  
www.mimacom.ch

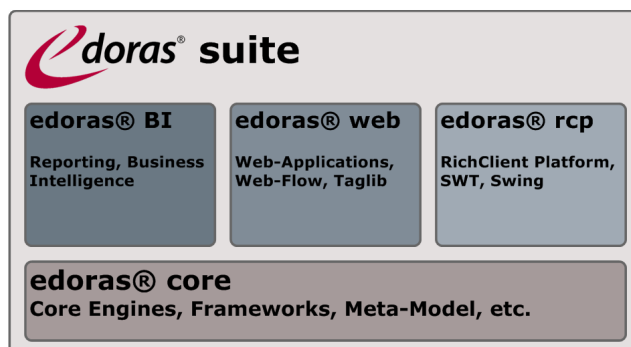
### Zusammenfassung

Die edoras® suite ist eine JEE Standardapplikationsplattform, mit der Applikationen mit aktuellen Technologien und methodischen Ansätzen (*Model Driven Development*) entwickelt werden können. Diese Plattform bietet die Möglichkeit, Rich-Client- und Webapplikationen auf demselben Fundament aufzubauen. Durch den Einsatz eines Meta-Models können fachliche Domänen (Entitäten und Attribute) spezifiziert, sowie vorgefertigte, erprobte und umfassende Basismechanismen eingebunden werden. Die edoras® core components und die edoras® composite applications bilden die Architektur der edoras® suite. Mit Hilfe dieser Komponenten wird eine Applikation aufgebaut. Dadurch wird die konzeptionelle Phase abgekürzt und der Entwicklungsaufwand reduziert sich.

### Motivation

Eine individuelle Software soll viele spezifische Kundenbedürfnisse abdecken und dennoch in einem bestimmten Kostenrahmen und mit den aktuellen technischen Java-Standards realisierbar sein. Bei einem vollständigen Neuaufbau einer solchen Software ist jedoch oftmals mit erhöhten Kosten und sich ständig wiederholenden Programmieraufgaben zu rechnen. Weiterhin bergen der Einsatz und die Evaluation neuer Technologien und unterschiedlicher Softwarekomponenten Risiken und verursachen zeitlichen Verlust. Die edoras® suite abstrahiert bewährte Technologien und bietet einheitliche APIs für die integrierten Technologien und Softwarekomponenten an. Sie wirkt erhöhten und nicht marktkonformen Entwicklungskosten entgegen und löst immer wiederkehrende technische Aufgabenstellungen. Die edoras® suite wurde unter Berücksichtigung der folgenden Aspekte entwickelt:

- JEE konforme und erprobte Applikationsplattform mit Fokus auf **Light-Weight, Performance** und **Qualität**
- Business Intelligence -, Rich-Client- und Webapplikations-Funktionalität stehen in einem Guss zur Verfügung
- Performanter Datenaustausch ist gewährleistet
- Wartungskosten bleiben auf Grund einheitlicher APIs und bekannter Entwicklungsbasis niedrig
- Integration in heterogene Umfelder (MS-Office, SAP, etc.) ist möglich
- Erprobte Basiskomponenten, -mechanismen und Standardapplikationen werden bereits zur Verfügung gestellt
- Einfaches *Customizing* gemäss kundenspezifischen Anforderungen durch *Model Driven Development* und graphische Designer



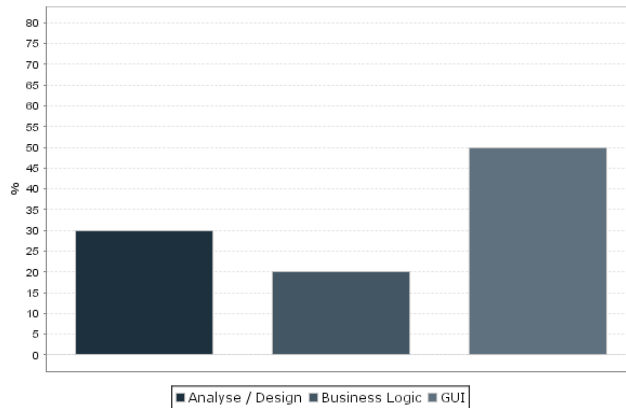
Bildlegende 1: Übersicht



- Standardfunktionalitäten stehen dem Benutzer wahlweise in einer Rich-Client-Plattform, in einer Webapplikation oder in einer kombinierten Variation (Hybrid-Applikation) zur Verfügung

Individuelle Kundenanforderungen werden bereits auf der edoras® suite erfasst und anschließend unter Verwendung der bestehenden Komponenten umgesetzt. Dadurch verkürzt sich die Realisierungszeit in den verschiedenen Disziplinen der Softwareentwicklung.

Die nachfolgende Grafik zeigt die durchschnittliche prozentuale Zeitersparnis in den Bereichen Analyse/Design, Erstellung der Business Logik und GUI.



Bildlegende 2: Zeitersparnis

### Highlights

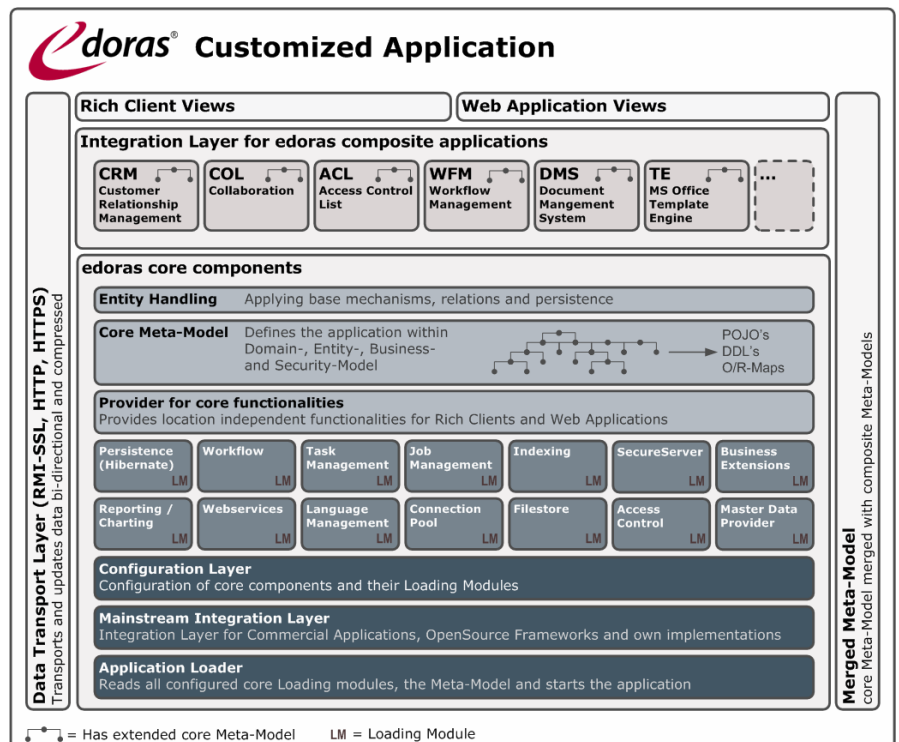
**Fokus:** JEE Applikationsplattform - Light-Weight, Performance und Qualität

**Zeitersparnis:** Analyse/Design 30%, Business Logic 20%, GUI 50%

Auf Grund der vielen bereits vorhandenen Basiskomponenten, kann die Realisierungszeit, je nach Vorhaben (z.B. ERP- oder CRM-Kundenapplikation), um ein Vielfaches verkürzt werden.

### Lösungsansatz

Die edoras® suite ist vollständig mit der Programmiersprache Java entwickelt. Dabei wird der Fokus auf eine leichtgewichtige und performante Softwarearchitektur gelegt, mit der Kundenapplikationen modellgestützt entwickelt werden können.



Bildlegende 3: Softwarearchitektur

Die Softwarearchitektur von edoras® fügt sich aus den nachfolgenden, aufgabenbezogenen Elementen zusammen.

### edoras® core components

Basiskomponenten und -mechanismen, welche sich aus bestehenden, bewährten OpenSource- und eigens erstellten Komponenten zusammensetzen. edoras® abstrahiert diese Komponenten und vernetzt sie über eine einheitliche API. Einige eingesetzte OpenSource-Komponenten:

OpenSource Komponente	edoras® component
Hibernate	Persistence
Eclipse Enterprise Modelling Framework	Model Driven Designers
Code Generation Library (cglib)	Entity Handling
Apache Common Libraries (Log4j, etc.)	Logging
Eclipse BIRT, Apache POI	Reporting
Lucene	Indexing
Apache AXIS	Webservice

### edoras® composite applications

Anwendungs-Halbfabrikate, welche in einer Kundenapplikation mit Hilfe des Meta-Models eingebunden werden können. Für die endgültige Kundenapplikation wird ein Meta-Model erstellt, welches auf einem oder mehreren composite Meta-Models und dem core Meta-Model basiert.

### Integrationschicht für composite applications

Die Integrationschicht besteht aus folgenden Teilen:

- Dem übergreifenden Meta-Model, welches die zu erstellende Kundenapplikation beschreibt. Es ist eine Kombination aus den zusammengeführten Meta-Models der eingesetzten composite applications und core components.
- Erweiterungen basierend auf *Business-Extensions* (Siehe unten).
- View- und Docking-Framework für die Anbindung graphischer Komponenten.

### User Interface für Rich-Client- und Webapplikationen

Für Web- (HTML, JSP, AJAX) sowie Rich-Client-Applikationen (Swing) stehen gleichwertige graphische Komponenten zur Verfügung, welche für die Darstellung von Objektlisten und Objektdetails zuständig sind. Dabei stehen Standardmechanismen wie beispielsweise Sortierung und Filterung bereits zur Verfügung. Zusätzlich werden vorgefertigte und erweiterbare Ansichten für edoras® core components und composite applications angeboten. Diese Elemente beschreiben das View-Framework, welches im Docking-Framework so integriert ist, dass der Benutzer seine Arbeitsfläche selber gestalten, einstellen und abspeichern kann.

Zur automatischen Aktualisierung von Rich-Clients wird Java WebStart verwendet: Beim Starten der Applikation wird automatisch die neuste Applikationsversion von einem Webserver heruntergeladen, installiert und ausgeführt.

### Mainstream Integration Layer

Falls andere Komponenten als die vorhandenen core components verwendet werden sollen, so können hier auch Fremdkomponenten oder eigene Entwicklungen eingebunden werden. Zudem enthält diese Schicht Standardschnittstellen:

- Ausgabe von Textdateien: XML, CSV, etc.
- Ausgabe von MS Office Excel Formaten
- Schnittstelle zu MS Outlook (Synchronisation von Adress- und Kalendern)
- Schnittstelle zu TwixTel

### Highlights

**OpenSource:** Hibernate, Eclipse, cglib, Apache, Lucene

**UI Technologien:** Swing, HTML, JSP, AJAX

**Integration / Ausgabe:** MS Excel, MS Outlook, TwixTel, XML, CSV, etc.



### Configuration Layer

Die Konfigurationsschicht dient zur Konfiguration der edoras® core components, deren *Loading Modules* und Abhängigkeiten. Sie ist eine horizontale Komponente, welche es ermöglicht, Kernkomponenten weitgehend mit Hilfe der *Loading Modules* konfigurieren.

### Application Loader

Der *Application Loader* lädt sämtliche *Loading Modules* unter Berücksichtigung ihrer Abhängigkeiten untereinander. Diese *Loading Modules* initialisieren die Kernkomponenten. Ferner werden das Meta-Model und Ansichtskonfigurationen geladen, sowie die Präsentationsschicht (Rich-Client-/Web-GUI) zur Verfügung gestellt.

### Data Transport Layer

Der *Data Transport Layer* verbindet die *User Interfaces* (Presentation Layer) mit den core-Funktionalitäten (Business Layer / Middleware). Hierbei wird ein *Business-Delegate (BD) - Session-Facade (SF) - Pattern* benutzt, das anhand des *Business Delegate* spezifische Businesslogik und Standardfunktionalitäten ortstransparent zur Verfügung stellt. Der Layer für die Kommunikation zwischen BD und SF ist frei definierbar. Per Standard werden das Java-proprietäre RMI (Remote Method Invocation) für verteilte Applikationen, sowie eine lokale Kommunikation für Web-Applikationen verwendet. Es sind jedoch auch beliebige andere *Transport Layer* einsetzbar (EJB, SOAP, etc.). Weiterhin bietet der *Data Transport Layer* die Möglichkeit, spezifische Businesslogik im *BD* zur Verfügung zu stellen, ohne dass Schnittstellen angepasst werden müssen. Diese Funktionalitäten stehen ebenfalls ortstransparent zur Verfügung.

Auf Grund dieser Architektur können mit der edoras® suite unterschiedliche Applikationsarten erstellt werden:

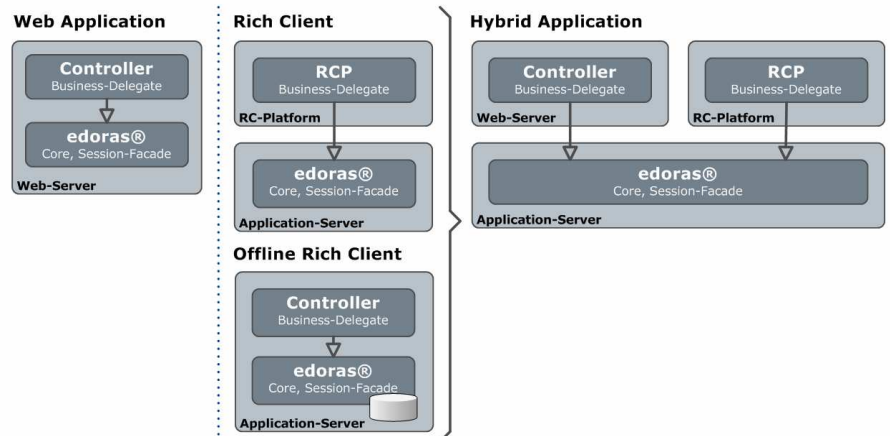
- Web-Applikation
- Rich-Client-Applikation
- Hybrid-Applikation (Kombinierte Rich-Client- und Web-Applikation)
- Offline Client

### Highlights

**Businesslogik:** Business-Delegate – Session-Facade – Pattern

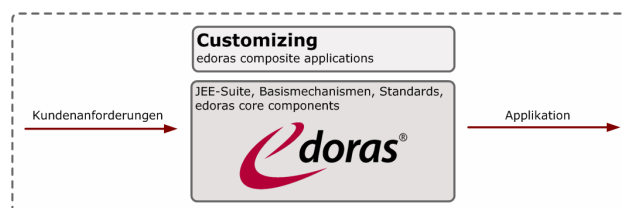
**Kommunikation:** RMI, EJB, SOAP, lokal und ortstransparent

**Standardkomponenten:** Einheitliches Vorgehen



Bildlegende 4: Applikationsarten

Mit Hilfe der edoras® suite und deren Standardkomponenten lässt sich auch ein einheitliches Vorgehen bei der Entwicklung von kundenspezifischen Applikationen ableiten. Dadurch können die Konzeptionsphase vereinfacht und Aufwandschätzungsprozesse abgekürzt werden.



Bildlegende 5: Customizing

## Vorgehen

1. Bei der Konzeption einer Kundenapplikation wird zuerst festgelegt, mit Hilfe welcher core components und composite applications am meisten Funktionspunkte abgedeckt werden können.
2. Ist dies erfolgt, wird ein edoras®-Projekt in der eclipse IDE angelegt.
3. Die core components, respektive deren *Loading Modules*, werden im *Configuration Layer* konfiguriert, oder wenn nötig mit anderen oder eigenen Implementation mit Hilfe des *Mainstream Integration Layers* versehen.
4. Als nächstes werden die composite applications mit den dazugehörigen Meta-Models eingebunden. Alle core components und composite applications basieren auf relationalen Datenbankmodellen.
5. *Customizing* der Applikation. Dies ist in allen Architekturelementen möglich.

## Lösungsübersicht

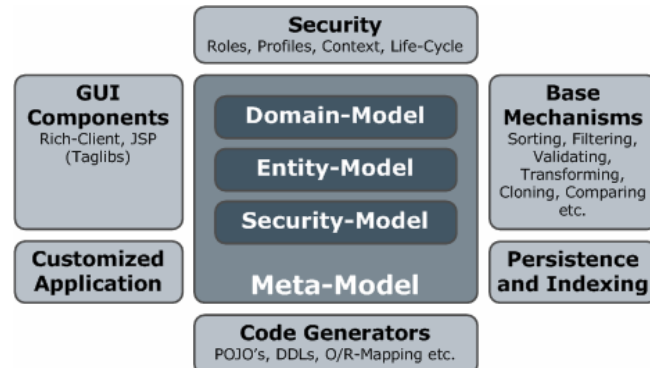
Das zentrale Architekturelement stellt das Meta-Model dar. Die zu erstellende Applikation wird in den folgenden Kernbereichen vollständig modelliert:

- **Domain Model:** Festlegung der Domänen (Domänen der eingebundenen composite applications sowie der spezifischen Domänen).
- **Entity Model:** Festlegung der Entitäten des *Domain Models*, inkl. deren Attribute
- **Security Model:** Abbildung von Rechten mit Hilfe von Rollen und Profilen auf Anwendungskontexte (Formular, Life-Cycle, etc.). Rechte können bis auf Attributebene genaustens festgelegt werden.

## Highlights

**Meta-Model:** Domains, Entities, Security, Anbindung GUI-Components, Basismechanismen

**Customizing:** Einheitliches Vorgehen



Bildlegende 6: Meta-Model

Neben den Kernbereichen werden weitere Bereiche modelliert:

- **Security:** Die im *Security Model* angegebenen Zugriffsrechte auf Attribute werden anhand von Security-Profilen und Benutzerrollen der edoras® composite ACL Benutzerverwaltung so zusammengefasst, dass diese in der Applikation für Benutzer und Gruppen verwaltet werden können.
- **GUI Components:** Beim Erstellen von eigenen Ansichten (Rich-Client- und Web-GUI), können die graphischen edoras®-Komponenten direkt mit den Entitäten des Meta-Models verknüpft werden. Somit stellen sich diese selbst richtig dar und die dazugehörigen Daten sind bereits bearbeitbar (*Create, Read, Update und Delete – CRUD*).
- **Base Mechanisms:** Basismechanismen können direkt auf den Entitäten definiert werden. Diese werden bei Ausführung einer CRUD-Funktion in der Applikation automatisch angewendet, gemäss den Meta-Model-Einstellungen. Die folgenden Basismechanismen können dabei insbesondere genutzt werden:



- ☞ Erstellen eines Zeitstempels auf einem neuen Objekt
- ☞ Erstellen des Benutzernamens auf einem neuen Objekt
- ☞ Führen des Zeitstempels auf einem aktualisierten Objekt
- ☞ Führen des Benutzernamens auf einem aktualisierten Objekt
- ☞ Versionierung eines Objektes (einfach, mono- und bi-temporal)
- ☞ Statusbehandlung eines Objektes und Anknüpfung an einen edoras® Workflow
- ☞ Schreibschutzstrategie eines Objektes (Optimistic / Pessimistic Locking)
- ☞ Relationen (One-To-One, One-To-Many, Many-To-Many), welche bis auf das Datenbankmodell angewendet werden
- ☞ Standardsortierreihenfolge, welche auf Tabellen standardmässig angewandt werden soll
- ☞ Angabe einer Basisentität für Vererbungen im Modell
- ☞ Angabe der Zugriffsrechte auf Attributsebene für das *Security Model*
- ☞ Einrichtung eines Mutationsjournals, welches auf Attributsebene angewendet werden kann
- ☞ Optional: Angabe einer Entitätsfassade, um die grundsätzliche Entitätsbehandlung zu beeinflussen oder abzuändern

Eine edoras®-Applikation injiziert die im Meta-Model definierten Basismechanismen einer Entität aspektorientiert mit Hilfe der *cglib* ([cglib.sourceforge.net](http://cglib.sourceforge.net)) zur Laufzeit in den Java-Bytecode. Dies hat zur Folge, dass sich der Entwickler weder um den Zeitpunkt des Aufrufes einer solchen Funktionalität, noch um die Anbindung und Codierung derselben kümmern muss.

## Highlights

**Basismechanismen:** Relationen, Mutationsjournal, Zugriffsrechte, Indexierung, Versionierung, etc.

**Persistence, Indexing:**  
Unterschiedliche Möglichkeiten → Datenbank, Indexierung, etc.

**Code Generation:** POJOs, O/R-Mappings, DDL's

- **Persistence, Indexing:** Unter Anwendung der im Meta-Model definierten Basismechanismen werden Objekte persistiert. Dabei stehen unterschiedliche Modelldefinitionen zur Verfügung:
  - ☞ Das Objekt kann persistent auf einem Datenträger abgelegt werden – im Normalfall ist dies eine relationale Datenbank.
  - ☞ Das Objekt wird transient im flüchtigen Rechnerpeicher belassen.
  - ☞ Verwendung von anderen Speicherverfahren mit Hilfe der Definition eines eigenen *Persisters*.
  - ☞ Nutzung weitergehender Speichermöglichkeiten, wie der Indizierung, zu der bereits eine eigene Kernkomponente existiert.
- **Code Generators:** Mit Hilfe der Code Generatoren werden folgende Erzeugnisse erstellt:
  - ☞ O / R – Mapping-Dateien für Hibernate.
  - ☞ *Plain Old Java Objects* – *POJO's*, welche nur Setter- und Getter-Methoden aufweisen.
  - ☞ Datenbank DDL-Scripts.
- **Customized Application:** Der spezifischen Kundenapplikation stehen das Meta-Model und die darin enthaltenen Metainformationen jederzeit zur Verfügung.

## edoras® core components

Die core components bilden das Fundament jeder edoras®-Applikation und werden grösstenteils auf Grund der definierten Basismechanismen des Meta-Models eingebunden. Diese Kernkomponenten können natürlich auch separat verwendet werden. Sie werden in einer core-Konfiguration mit Hilfe von *Loading Modules* konfiguriert, welche untereinander Abhängigkeiten aufweisen. Damit wird eine korrekte Reihenfolge beim Ladevorgang durch den *Application Loader* gewährleistet. Eine Kernkomponente kann mit Hilfe des edoras® *SingletonProviders* von jedem Punkt aus angefordert und verwendet werden.

## edoras® core Persistence

Die Persistierung der Objekte wird mit Hilfe des im Meta-Model definierten *Persisters* durchgeführt. Per Standard wird dies mit dem Open Source Framework Hibernate ([www.hibernate.org](http://www.hibernate.org)) vorgenommen. Falls im Meta-Model *Many-To-Many* – Relationen definiert wurden, werden die Relationsentitäten beim Speichern selbständig angelegt. Grundsätzlich wird ein zu persistierendes Objekt anhand eines Objektgraphen so inspiziert und traversiert, dass die relationalen Objekte dieses Objektes rekursiv und unter den im Meta-Modell konfigurierten Bedingungen (Constraints) ebenfalls richtig persistiert werden.

## edoras® core Workflow

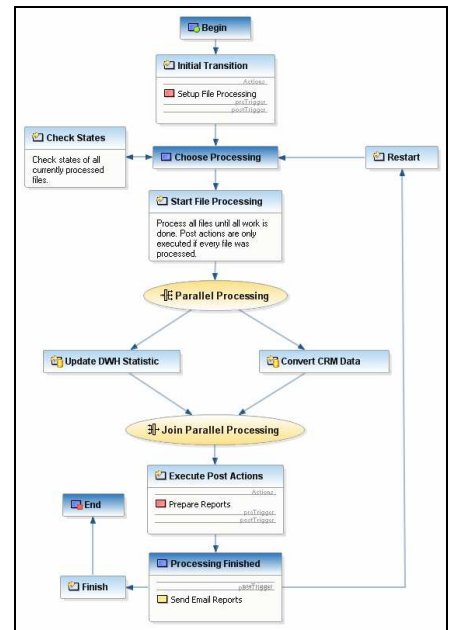
Der edoras® core Workflow bildet das Grundgerüst, um Workflows und GUI-Workflows (Wizards und Webflows) zu definieren. Dafür steht ein graphischer Workflow-Designer zur Verfügung, mit dem man klassische State-Event-Modelle erstellen kann, die von der core Workflow-Engine betrieben werden.

Auf dieser Komponente basiert das edoras® composite Workflow Management, welches die Pflege von spezifischen Workflows erlaubt.

Durch die Verwendung eines Workflows in der Applikation

- wird der Benutzer sinnvoll durch die Applikation geführt,
- ist die Nachvollziehbarkeit jederzeit gewährleistet.

Die im Workflow festgehaltenen Beschreibungen können in der Applikation direkt als Hilfetexte (Tool-Tips) verwendet werden.



Bildlegende 7: Workflow

## Highlights

**Workflow:** Graphischer Designer, serverseitige Workflows und GUI-Workflows

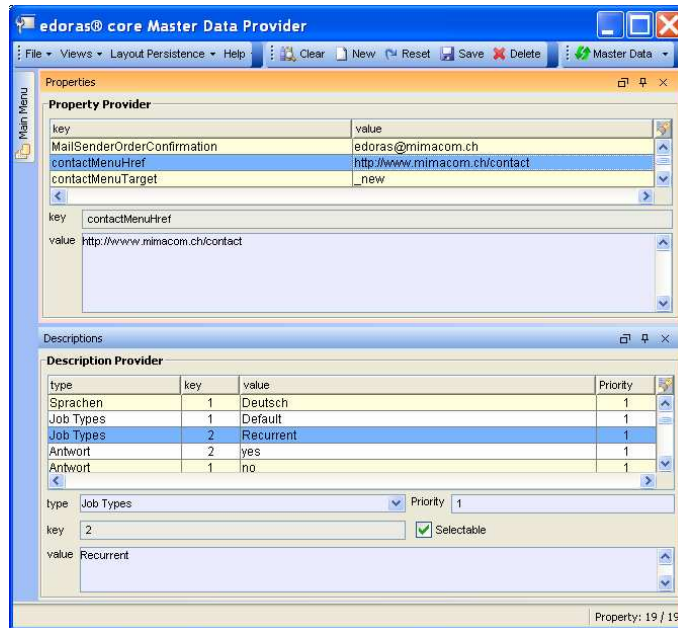
**Master Data:** Descriptions, Properties, etc.

**Persistence:** Abbildung von Relationen auf Meta-Model-Ebene möglich

## edoras® core Master Data Provider

Der Master Data Provider bietet spezifische Stammdaten einer edoras® Applikation an:

- *Descriptions:* Verwaltung von Werten und Codes. Codes werden von fachlichen und anderen Kernkomponenten genutzt. Codes werden vom Entwickler für Typen wie zum Beispiel Jobtypen, Geschlecht oder Anrede festgelegt und mit Beschreibungen hinterlegt. Beschreibungen werden mit Hilfe des edoras® core Language Manager in Fremdsprachen übersetzt. Diese Daten werden üblicherweise durch eine Auswahl-Combo-Box dargestellt.
- *Properties:* Einstellungen der Applikation. Beispiele für die Verwaltung von einfachen Programm-Parametern sind etwa Laufwerkspfade oder andere Konfigurationen. Diese Werte können durch einen Benutzer Administrator global und zur Laufzeit geändert werden.
- Es werden konkrete Stammdaten angeboten, beispielsweise eine Länderliste mit Ländercodes und Ortschaften mit PLZ. Weitere solche Providers können durch den Entwickler selbst erstellt und eingebunden werden.



Bildlegende 8: Master Data Provider

## Highlights

### Task Management:

Aufgabenmanagement für Gruppen und Benutzer

### Job Management:

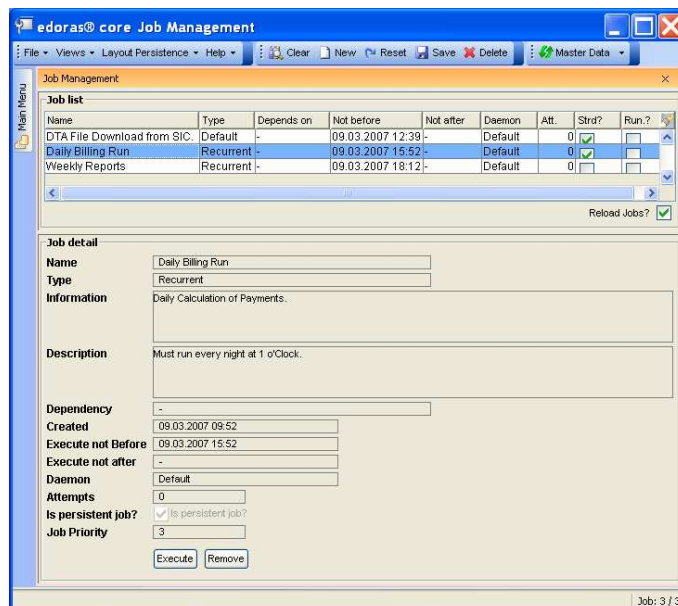
Asynchrone Jobs für Massenverarbeitungen, periodische Aufgaben, etc.

### edoras@ core Task Management

Das Task Management ist ein klassisches Aufgabenmanagement, mit dem Benutzern oder Gruppen einer edoras@-Applikation Aufgaben zugewiesen werden können. Die dabei hinterlegten Aktionen und Abfolgen werden im edoras@ core Workflow festgelegt.

### edoras@ core Job Management

Das Job Management bietet vergleichbare Funktionalitäten wie *UNIX Cron-jobs* an. Es können einmalige oder sich periodisch wiederholende Aufträge erstellt werden. Diese werden per Standard in der Datenbank gespeichert und zum angegebenen Zeitpunkt ausgeführt. Diese Jobs eignen sich beispielsweise für die Implementierung nächtlicher Massenverarbeitungen oder den Versand von abonnierten Reports. Es können Jobs definiert werden, welche nur im flüchtigen Speicher angelegt werden und die nach einem Neustart des Systems nicht mehr vorhanden sind.



Bildlegende 9: Job Manager

Für das Job Management existiert eine Kontrollansicht, welche folgende Informationen zur Verfügung stellt:

- Name des Jobs
- Typ. Die Typen können vom Entwickler im edoras® core Master Data Provider festgelegt werden.
- Abhängigkeit von anderen Jobs
- Frühste Ausführung des Jobs
- Letzte Ausführung des Jobs
- Name des Jobs Daemons (Pool). Es können unterschiedliche Daemons konfiguriert werden.
- Anzahl misslungene Ausführungsversuche
- Information, ob es sich um einen persistenten oder transienten Job handelt
- Information, ob sich ein Job aktuell in Ausführung befindet

#### **edoras® core Access Control**

Diese Komponente stellt die grundlegenden Berechtigungsmechanismen für eine edoras®-Applikation zur Verfügung. Nach erfolgter Authentifizierung wird innerhalb der Applikation ein sogenanntes *User-Credential* generiert, welches folgende Informationen enthält:

- Mandantenschlüssel bei mandantenfähigen Applikationen
- Login-Name
- Sprache des Benutzers
- IP- und MAC-Adresse des Benutzers
- Funktionsrollen
- Applikationsrollen

Die core component Access Control steht auch als composite application zur Verfügung, die eine rollen- und sicherheitsprofilbasierte Verwaltung ermöglicht.

#### **edoras® core Indexing**

Die Indexierkomponente erlaubt es, die im Meta-Model für die Indexierung konfigurierten Objekte so zu indexieren, dass diese mit Hilfe von *Fuzzy*-Textpattern und/oder Metainformationen wieder aufgefunden werden können. Die Indexierung ist im Wesentlichen eine spezielle Ausprägung der Persistierung von Objekten. Eine wesentliche Anwendung dieser Kernkomponente ist das edoras® composite DMS (Document Management System), welches beliebige Dateien volltextindexiert ablegt. Indexierte Daten können wiederum mit Hilfe der Suchmaschine aufgefunden werden. edoras® core Indexing basiert per Standard auf der OpenSource Search-Engine Lucene ([lucene.apache.org](http://lucene.apache.org)).

#### **edoras® core Logging**

Für das programmtechnische Logging wird per Default Log4j ([logging.apache.org/log4j](http://logging.apache.org/log4j)) in der Logging-Komponente eingesetzt. Diese wurde so erweitert, dass weitere Log-Levels speziell für Softwareentwicklungstätigkeiten wie Debugging zur Verfügung stehen. Weiterhin können client-seitige Log-Informationen zum Server gesendet und ausgewertet werden. Das Logging-Verhalten ist zur Laufzeit veränderbar.

#### **edoras® core Filestore**

Der Filestore persistiert Dateien wahlweise in einer Datenbank oder im Dateisystem und ermöglicht die Versionierung von Dateien. Folgende Metadaten werden dabei insbesondere geführt:

- Speicher- und Modifizierungsdatum
- Benutzer, der die Datei gespeichert oder aktualisiert hat
- Version
- Format
- Dateiname und Grösse
- Kommentare
- Schlüsselwörter (Tags)
- Kategorie

### Highlights

**Access Control:** Rollenbasierte Benutzerverwaltung

**Indexing:** Volltextsuche in MS Office Dokumenten, Indexierung von Objekten

**Filestore:** Ablage von Dateien wahlweise in der Datenbank oder auf Dateisystem

edoras® core Filestore bildet zusammen mit der Indexing-Komponente das Herzstück des edoras® composite DMS.

## edoras® core SecureServer

Der SecureServer ist ein RMI-Server und stellt bei einem Rich-Client die Middleware dar. Er interagiert mit den angemeldeten Rich-Clients mit Hilfe des *Transport Layers* über eine sichere SSL-Verbindung und sorgt dafür, dass alle Clients dieselben Datenzustände aufweisen. Die Übertragung wird komprimiert, um möglichst wenig Datenverkehr zu verursachen. Das bereits erwähnte *Business-Delegate – Session-Facade – Pattern* wird bei der Verbindung zwischen Rich-Client und SecureServer – im Gegensatz zu einer Webapplikation – mit dem Java-proprietären Protokoll RMI (Remote Method Invocation) realisiert.

Vordefinierte und vom SecureServer zur Verfügung gestellte Datenmodelle werden beim Start eines Rich-Clients vorgeladen und in einem *SmartCache* abgelegt. Der edoras® SmartCache kann temporäre Objekte über eine bestimmte Zeitdauer verwalten. Dazu wird beim Registrieren eines Datenmodells eine relative Zeit (Timeout) angegeben. Es kann festgelegt werden, ob das Timeout beim Zugriff auf ein Objekt zurückgesetzt wird oder nicht. Nach Ablauf des Timeouts wird das Objekt aus dem Cache entfernt. Der SmartCache wird von allen Komponenten in edoras® verwendet.

Durch das Vorladen von Modellen und die Nutzung des SmartCaches wird die Performance von Rich-Clients in einer verteilten Applikation optimiert. Die vorgeladenen Daten werden durch den SecureServer nur noch bei Bedarf automatisch aktualisiert.

Der SecureServer kann auch in andere JEE-Container, wie beispielsweise *IBM WebSphere*, *JBoss* oder *BEA WebLogic* integriert und betrieben werden.

## edoras® core Business Extensions

Business Extensions sind applikationsspezifische, fachliche und serviceorientierte (Service Oriented Architecture – SOA) Erweiterungen, welche denselben Mechanismus wie das *Business-Delegate – Session-Facade – Pattern* nutzen. Diese Business Extensions können in der Applikation jederzeit aufgerufen werden, und stellen fachliche Funktionen ortstransparent zur Verfügung. Für die Nutzung einer Business Extension müssen keine speziellen *Remote-Interfaces* implementiert und *Client-Stubs* generiert werden. Die konfigurierte Funktionalität wird einem definierten Zugangskontext zur Verfügung gestellt.

## edoras® core Connection Pool

Über den Connection Pool lassen sich beliebig viele Datenbanken anbinden. Eine mandantenfähige Applikation kann mit Hilfe von verschiedenen Connection-Pools realisiert werden. Das Verhalten des Pools kann folgendermassen beeinflusst werden:

- Dynamische Skalierung des Pools
- Einstellen der Connection-Timeouts
- Überwachung der Datenbankverbindungen
- Auto-Collection von nicht mehr verwendeten, respektive von nicht geschlossenen Verbindungen (selbstheilend)

## edoras® core Language Management

Der in der edoras® suite vorhandene Sprachenmanager lässt eine Übersetzung der Applikation von 1-n Sprachen zu. Es wird eine Grundsprache definiert, in der die Applikation entwickelt wird. Danach kann die Applikation auf unterschiedliche Arten übersetzt werden:

Export einer MS-Excel-Datei, welche beispielsweise durch ein Übersetzungsbüro übersetzt wird. Danach kann diese Datei wieder in die Applikation importiert werden.

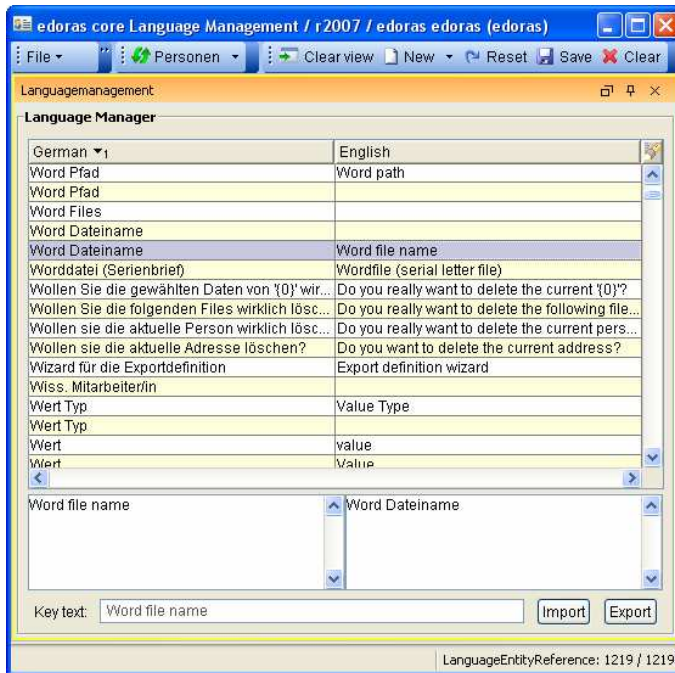
Übersetzung mit Hilfe der Masken des edoras® core Language Management. Eine Webapplikation kann kontextbasiert übersetzt werden. Das heisst, Texte werden direkt auf den verschiedenen Seiten der Applikation übersetzt.

## Highlights

**SecureServer:** Verschlüsselte und komprimierte Verbindung zu Clients, Middleware mit Integrationsfähigkeit in andere JEE-Container

**Business Extensions:** SOA mit ortstransparentem Zugriff auf Businesslogik

**Connection Pool:** Selbstheilender Connection Pool mit Anbindungsmöglichkeit an mehrere Datenbanken



Bildlegende 10: Language Manager

## Highlights

**Language Manager:** Übersetzung der Applikation in mehrere Sprachen

**Webservices:** Integration in SecureServer möglich

**Reporting / Charting:** Ausgabe von PDF, XLS, CSV, TXT, etc. Und Charts (Kuchen-, Linien-, Balken-Diagramme, etc.)

Der Sprachenmanager ist eine horizontale Komponente und übersetzt in einer edoras®-Applikation folgende Elemente per Standard: Namen für Labels, welche im Meta-Model definiert und für die Ansichten der Applikation verwendet werden

- Beschreibungen in Workflow-Modellen
- Fehlermeldungen
- Description-Werte des edoras® core Master Data Providers
- Menüeinträge
- Dialoge
- Mit edoras® core Reporting erstellte Reports
- edoras® composite applications

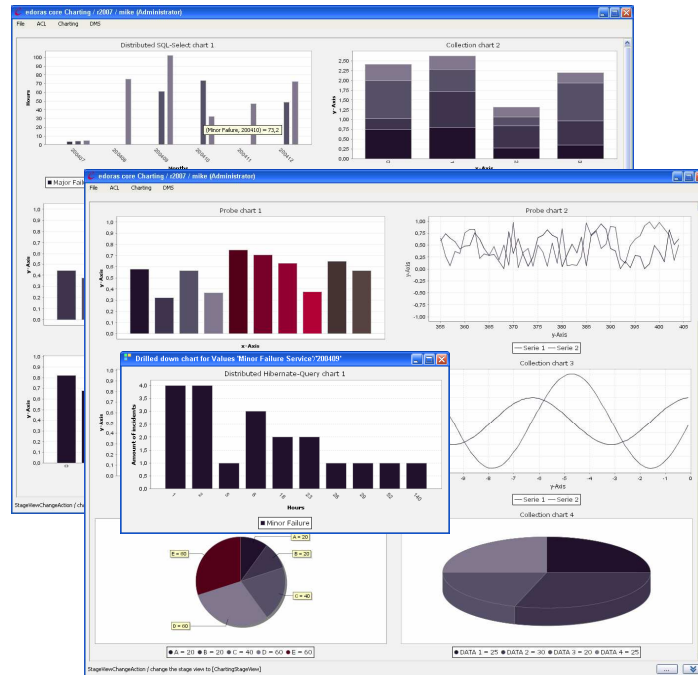
Die in der Anwendung zu übersetzenden Texte werden als Schlüssel für das betreffende Sprachelement hinterlegt. Denselben Schlüssel gibt es innerhalb der Anwendung nur einmal. Die Schlüssel der Sprachelemente sind per Standard die Initialtexte der Grundsprache. Die Texte der Grundsprache selbst können ebenfalls geändert werden, ohne dass der Schlüssel dabei verändert wird.

### edoras® core Webservices

edoras® erlaubt den Betrieb von Webservices im SecureServer. Dazu wird per Default Apache AXIS ([ws.apache.org/axis](http://ws.apache.org/axis)) eingesetzt. Alle AXIS-spezifischen Einstellungen werden über ein Loading Module vorgenommen. Danach können AXIS-Webservices einfach ausgeführt werden und es steht eine klassische, serviceorientierte Architektur zur Verfügung.

### edoras® core Reporting & Charting

Um Reportvorlagen und Reports erstellen zu können, wird in edoras® für PDF-Formate das *eclipse BIRT Reporting System* ([www.eclipse.org/birt](http://www.eclipse.org/birt)) und für MS-Excel-Formate *Apache POI* ([jakarta.apache.org/poi](http://jakarta.apache.org/poi)) per Default verwendet. Das Reporting-Modul stellt dabei die Reportvorlagen so bereit, dass die dynamischen Werte für den Endreport einfach ergänzt werden können.



Bildlegende 11: Charting

Um eine graphische Aufbereitung von Kennzahlen zu ermöglichen, kann in edoras® die *Charting Engine* eingesetzt werden. Diese basiert auf *JFreeChart* ([www.jfree.org/jfreechart](http://www.jfree.org/jfreechart)) und erlaubt die Anbindung unterschiedlicher Datenquellen:

- **Value-Objects:** Darstellung von Objekten, welche im Meta-Model definiert wurden und als Datensätze interpretiert werden können
- **SQL:** Darstellung von SQL-selektierten Daten
- **Sensoren (Monitoring):** Darstellung von Daten, welche während des Betriebes der Applikation von verschiedenen Programmteilen hinzugefügt werden
- **Collection:** Darstellung einer aufbereiteten Datenliste
- Die dargestellten Diagramme werden automatisch aktualisiert, beinhalten konfigurierbare Drill-Down-Funktionalität und können vergrößert, gedruckt sowie als Bilder abgespeichert werden.

### edoras® composite applications

Die auf Basis der edoras® core components gefertigten Halbfabrikate sind die edoras® composite applications, welche eine bestimmte Menge von fachlichen Funktionen bereitstellen. Einige davon können als alleinstehende Applikationen verwendet werden. Diese composite applications können beliebig kombiniert werden. Folgende edoras® composite applications stehen zur Verfügung:

- 🔗 edoras® composite CRM (Customer Relationship Management)
- 🔗 edoras® composite DMS (Document Management System)
- 🔗 edoras® composite ACL (Rollenbasierte Access Control List)
- 🔗 edoras® composite Collaboration
- 🔗 edoras® composite Workflow Management (Lifecycle Management und Geschäftskontrolle)
- 🔗 edoras® composite Template Engine (Verwaltung von Serienbriefen)

edoras® composite applications basieren auf einer modularen Architektur und können bei Bedarf auch mit Services (z.B. Webservices) kombiniert werden. Sie dienen zusammen mit den core components als elementare Bausteine für künftige composite applications und Kundenapplikationen. Eine einzelne Applikation fungiert so nicht mehr zwangsläufig als dominierende Plattform, sondern existiert als funktionaler Baustein neben anderen Softwarebausteinen.

### Highlights

**composite applications:** modulartige Halbfabrikate für Kundenapplikationen, fassen verschiedene Fachgebiete zusammen und basieren auf den edoras® core components: CRM, DMS, ACL, Collaboration, Workflow Management, etc.

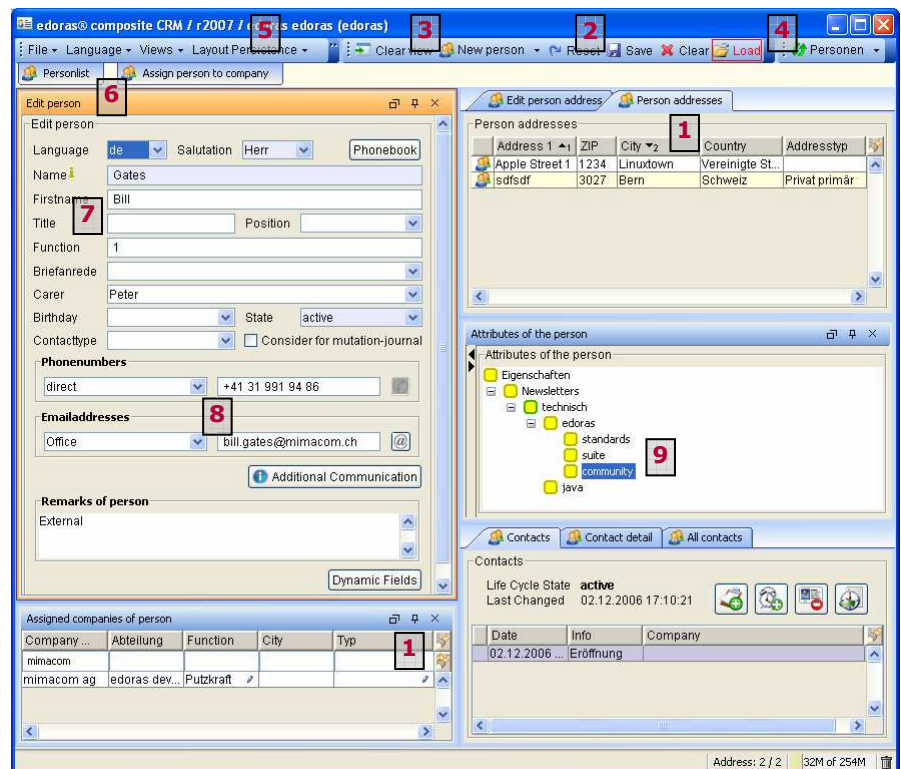
## edoras® View- und Docking-Framework

Die im View- und Docking-Framework integrierten Bedienelemente der core components und composite applications weisen eine Vielzahl von Standardfunktionen auf. Nachfolgend einige davon:

1. Mehrspaltige Sortierung und Filterung von einstufigen und hierarchischen Tabellen. Eingestellte Sortierungen und Filter können im Benutzerprofil abgespeichert werden.
2. Standardaktionen für das Speichern, Löschen und Zurücksetzen des aktuell bearbeiteten Datensatzes
3. Konfigurierbare und erweiterbare Aktionen für zu bearbeitende Datensätze
4. Schnelles Wechseln zwischen Ansichten
5. Konfigurierbare und rollenabhängige Menüs
6. Ansichten, Fast-Views (per Mauszeiger werden Ansichten expandiert) und Tabellenspalten können durch den Benutzer selbst angeordnet und ausgeblendet werden. Diese Einstellungen werden im Benutzerprofil abgespeichert.
7. Labels und Tooltips (Online-Hilfetexte) werden im Meta-Model definiert und beim Anzeigen gemäss der eingestellten Sprache übersetzt
8. Verknüpfungen mit der MS-Office-Welt beim Anwählen eines bestimmten MS-Office-Formats oder einer Email-Adresse
9. Unterschiedliche Datenansichten, wie beispielsweise Bäume, hierarchische und einstufige Tabellen, etc.

### Highlights

**View- und Docking-Framework:**  
Benutzerprofile, Mehrspaltige Sortierung und Filterung von Spalten, Standardaktionen für die Bearbeitung von Datensätzen, Schnelles Wechseln zwischen Ansichten, Verknüpfung mit der MS-Office-Welt, Konfigurierbares Look'n'Feel, Side-Menüs, etc.



Bildlegende 12: Standardbedienelemente

Weitere Features des View- und Docking-Frameworks:

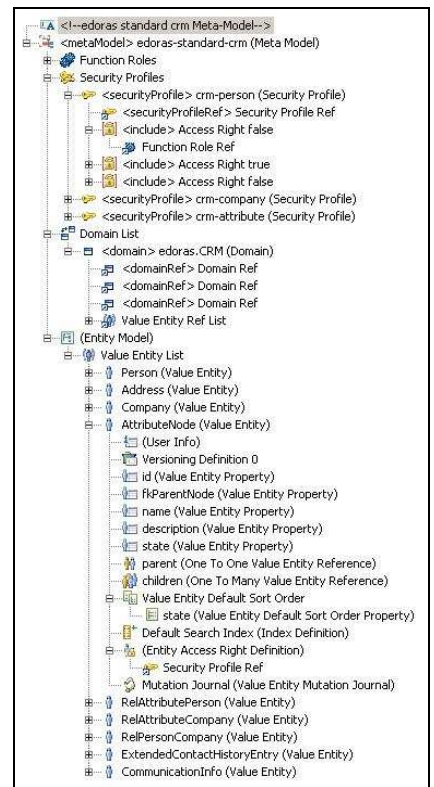
- Side-Menüs (ähnlich MS-Windows-Explorer)
- Navigation durch Datensätze
- Konfigurierbares Look and Feel (Windows XP, etc.)
- Sperrung von Ansichten, damit diese durch den Benutzer nicht geändert werden können
- etc.

## edoras® Model Driven Development

edoras® ermöglicht Model Driven Development und setzt dafür selbst entwickelte eclipse Plugins ein. Dabei wird ein Ansatz verfolgt, bei dem nicht nur Sourcecode generiert wird, sondern ein vom Entwickler erstelltes Modell die Applikation vollständig auf einer Meta-Ebene beschreibt. Das Modell wird zur Laufzeit interpretiert. Ein solcher Ansatz ist effizient und eliminiert das Risiko mit unbekanntem Code-Fragmenten konfrontiert zu werden. Das Modell kann in den flüchtigen Speicher geladen werden. Die nötigen Definitionen und die damit anzuwendenden Funktionalitäten sind in edoras® von jedem Punkt aus erreichbar.

## edoras® Meta-Model Designer

Wie bereits erwähnt, ist das Meta-Model das zentrale Element jeder edoras®-Applikation. Zum Erstellen dieses Meta-Models wird ein Designer verwendet, in dem die Entitäten, deren Attribute, die Basismechanismen, die Zugriffsrechte auf Attributsebene sowie die Referenzen auf Meta-Models verwendeter edoras® composite applications oder Drittapplikationen spezifiziert werden. Die edoras® composite applications beinhalten eigene Meta-Models, welche vom edoras® core Meta-Model vererbt werden. Für die Einbindung von composite applications in die eigene Applikation können vorhandene Meta-Models referenziert werden. Die darin enthaltenen Domänen werden als Referenz im eigenen Meta-Model eingebunden. Danach können die Elemente des eingebundenen Meta-Models verwendet werden (z.B. Vererbung einer Basisentität, Nutzung von Basisfunktionsrollen, etc.).



Bildlegende 13: Meta-Model

## Highlights

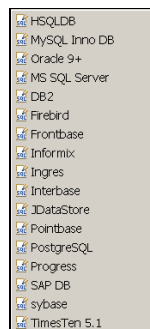
**Model Driven Development, mit eclipse Plugins:**  
Meta-Model Designer,  
Workflow Designer,  
Visual Designer

**Meta-Model Designer:**  
Spezifikation der Attribute,  
Entitäten, Basismechanismen, etc. /  
Systemdokumentation

**Unterstützte Datenbanken:**  
Oracle, MySQL, MS SQL Server,  
DB2, Sybase, SAP, etc.

Der Meta-Model Designer validiert das Modell bereits bei der Erstellung, so dass eine technisch fehlerhafte Spezifikation während der Designphase entdeckt werden kann. Das Meta-Model ist nach Erstellung integraler Bestandteil der Systemdokumentation.

Die gewünschten Basismechanismen werden mit Hilfe eines Kontextmenüs ausgewählt und gemäss den Anforderungen spezifiziert. Nach Erstellung des Meta-Models werden die Java-Objekte, die Datenbank-Mapping-Dateien und die Datenbankskripte für die Tabellen erstellt. Die Ausgabe der Datenbankskripte ist in unterschiedlichen Datenbank-Dialekten möglich.



Bildlegende 14: Mögliche DDL's



Bildlegende 15: Auswahl der Basismechanismen im Meta-Model Designer

## edoras® Workflow Designer

Mit dem edoras® Workflow Designer werden die fachlichen Wertschöpfungsketten einer Applikation modelliert. Dabei stehen folgende Workflow-Elemente zur Verfügung:

- **States:** Verschiedene Zustände wie Start-, Endstatus und Normaler Status.
- **Transiente und Persistente Transitions:** Bei entsprechender Auswahl wird der Übergang im Betrieb auf der Datenbank persistiert, um die Nachvollziehbarkeit zu gewährleisten. Jede Transition enthält eine Action, welche die eigentliche Business-Logik ausführt.
- **Verzweigungen zu Subflows:** Ein Workflow kann in einen anderen verzweigen.
- **Parallelisierung (Fork / Join):** Abläufe können untereinander synchronisiert werden.
- **Sensoren:** z.B. um Prozessdurchlaufzeiten messen zu können.
- **Groups:** Zusammenfassen von Elementen für dokumentarische Zwecke.
- **Triggers:** Elemente welche vor oder nach einem Status und Transition-Element asynchron oder synchron ausgeführt werden.
- **Action-Enabler:** Eine Aktion, welche in einer Transition ausgeführt wird und vor der Ausführung durch einen Benutzer mit Informationen versehen werden muss.
- **Connections:** Wegbeschreibende Verbindungen zwischen States und Transitions.
- Weitere Elemente für Workflows in Webapplikationen und für Wizards.

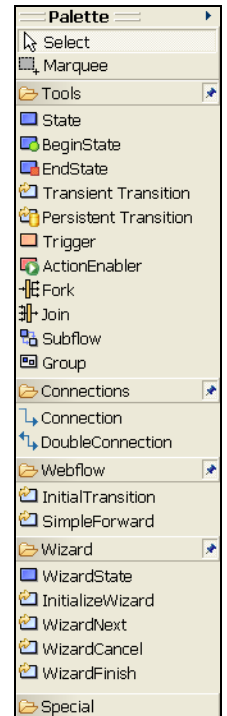
Das erstellte Modell liegt in XML-Format vor. Es wird nach dem Start der Applikation zur Verfügung gestellt und von der edoras® Workflow Engine betrieben.

## edoras® Visual Designer

Um die Daten der im Meta-Model definierten Entitäten mit einem Rich-Client-GUI verbinden zu können, steht im Visual-Editor der eclipse IDE eine edoras®-GUI-Palette zur Verfügung. Diese Palette enthält Grafikkomponenten, welche von den Swing-Elementen abgeleitet sind. Diese werden mit den im Meta-Model definierten Attributen einer Entität verknüpft und können so die Applikationsdaten darstellen. Ferner beinhalten sie folgende erweiterte Funktionalitäten:

- Die Bezeichnungselemente von Datenfeldern (JLabels, etc.) werden vom edoras® core Language Management je nach konfigurierter Sprache direkt übersetzt.
- Drag and Drop ist vorimplementiert; Der Entwickler implementiert nur noch fachliche Inhalte.
- Tool-Tips werden vom Meta-Model gelesen und im GUI auf dem entsprechenden Element angezeigt.

Auswahllisten können mit Angabe eines Typs einfach mit Daten aus dem edoras® core Master Data Provider versehen werden. In Tabellen angezeigte Objekte können über die Kopfzeile in jeder Spalte gefiltert und sortiert werden.



Bildlegende 16:  
WF-Palette

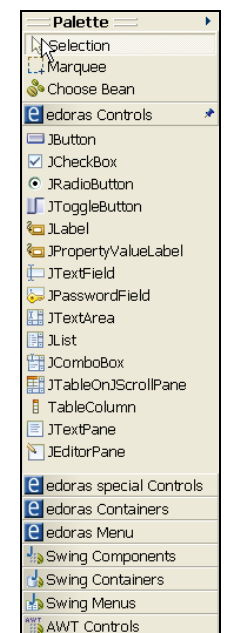
## Highlights

### Workflow Designer:

Abbildung von Wertschöpfungsketten / States, Transitions, Fork, Join, Subflows, Sensoren, etc.

### Visual Designer:

Erstellung von Rich-Client-Masken, Anknüpfen des Meta-Models für Grafikkomponenten, Drag and Drop, Tool-Tips



Bildlegende 17:  
GUI-Palette

## Vorteile

Durch den Einsatz der edoras® suite kann die Konzeptphase bei der Realisierung von Individualsoftware auf Grund der standardisierten Komponenten, dem einheitlichen Vorgehen und der Verwendung eines Meta-Models verkürzt werden.

Risiken werden minimiert, da bewährte edoras® core components und composite applications zur Verfügung stehen.

Folgende Merkmale sind für eine langfristige und erfolgreiche Strategie im Umfeld der Individualsoftwareentwicklung von Vorteil:

- ↪ 20-50% Kostenreduktion (Entwicklungszeit).
- ↪ Führende Entwicklungsmethodik (*Model Driven Development*), Technologie und Tools.
- ↪ Schnelle Entwicklung von Webapplikationen und Rich-Clients.
- ↪ Schulung / College.
- ↪ Einheitlicher Support / Maintenance / Release Management.
- ↪ Model Driven Development mit grafischen Designern basierend auf der eclipse IDE.
- ↪ edoras® beinhaltet mit den core components *Webservices* und *Business Extensions* eine serviceorientierte Architektur – *Service Oriented Architecture (SOA)*.
- ↪ Offene Plattform, keine Limiten für Systeme oder Entwickler.

edoras® wird stetig weiterentwickelt und veränderten Marktsituationen angepasst

edoras® ist eine in der Schweiz konzipierte und realisierte Open Source Applikationsplattform.

## edoras®

### JEE from tip to tail.

edoras® ist ein eingetragenes Warenzeichen der mimacom ag.

## Hardwareanforderungen

Minimaler flüchtiger Speicher: 512 MByte / Minimaler Festplattenspeicher: 100 MByte / Prozessoren: 1 Prozessor à ~ 1.0 GHz - 1.6 GHz

## Betriebssysteme

Server: Linux Redhat Enterprise Server 3 (oder höher) oder ähnliches Linux, MS Windows Server 2000-2003 / Client: MS Windows 2000, XP, Vista

## Systemsoftware

edoras® Rich Client: JRE 1.4.2 oder höher, Java WebStart-enabled, Apache Tomcat 4.1.27 oder höher / Datenbank: Oracle 9i+, MySQL und weitere

## Entwicklungswerkzeuge

Java, Eclipse IDE, edoras® Workflow Designer, edoras® Meta-Model Designer, edoras® Visual Designer, ANT Build Tools, Maven