IBM Cúram Social Program Management

Cúram Solution Architecture

Version 6.0.5

IBM Cúram Social Program Management

Cúram Solution Architecture

Version 6.0.5

Inhaltsverzeichnis

Abbildungsverzeichnis

v

Tabellen

Kapitel 1. Einführung

1.1 Einführung

This document provides an overview of the solution architecture of IBM® Cúram Social Program Management ™ from a business, development and deployment perspective. Cúram is a comprehensive commercial-off-the-shelf (COTS) solution for Social Enterprises. IBM Cúram Social Program Management™ also includes a environment for organizations wishing to develop their own Social Enterprise Management applications, or to augment those provided by IBM. An architectural description of IBM Cúram Social Program Management™ must thus describe the components of existing „out-of-the-box" business applications, and the technical architecture of the framework which underpins those applications and is reusable by developers building additional ones.

Kapitel 2. Business Architecture

2.1 Business Architecture

IBM Cúram Social Program Management™ is designed specifically for human services, health, labor, social security, and military and veterans' organizations - collectively referred to as social enterprises. The IBM Cúram Social Program Management™ solution is comprised of multiple layers, each designed to address specific and unique business and technical requirements of social enterprises.
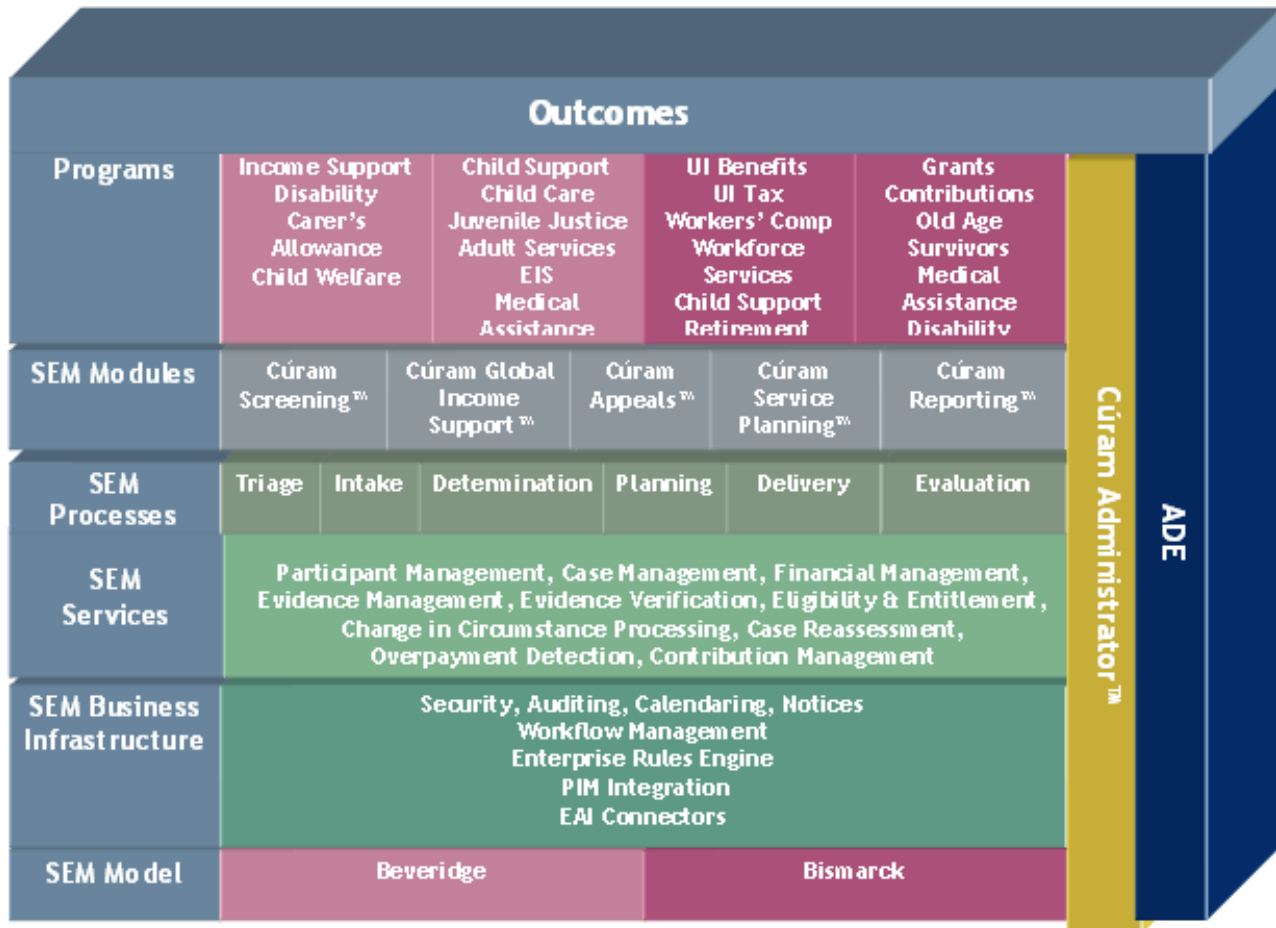
Abbildung 1. Cúram Business Model

By supporting both the Beveridge and Bismarck program delivery models, IBM Cúram Social Program Management™ provides the foundation required to administer benefits and services to support both needs-based and contribution-based programs. The Business Infrastructure is designed to provide support for many of the common processes found throughout the enterprise including security, workflow, and integration tools.

A Services layer includes the common services that support the service and benefit delivery processes for social enterprises. These services include industry-leading integrated case management and integrated eligibility and entitlement, as well as participant management, financial management, contribution management, and evidence management.

Common across social enterprises, the SEM processes define the mission-critical activities performed in the delivery of services and benefits. These common processes include: Triage, Intake, Determination, Planning, Delivery, and Evaluation. Each of the processes is supported by through supplied business processes based on global best practices.

A series of application modules are designed to support common business functionality and global rules and evidence required to address program delivery. For example, Cúram for Global Income Support™ supports the collection, management, application, and processing of business rules and evidence associated with income support programs world-wide.

Kapitel 3. The Cúram Application Development Environment

3.1 Einführung

IBM Cúram Social Program Management™ provides an environment for producing Java® EE ™ -compliant applications for the Human Services and Social Security industry. This environment includes:

- development aids that make it easier to produce n-tier Java EE™ -compliant applications;
- a higher-level business infrastructure needed by most enterprise-class systems, and especially targeting the Human Services and Social Security industries. In particular, infrastructure is provided to allow the capture of eligibility and entitlement rules for Social Security „products" (or programs), and to execute these rules within Cúram applications;
- an application model (in UML, the Unified Modeling Language) for these industries;
- off-the-shelf software components based on the application model;
- a pre-built reference application constructed using these components.

3.2 Development Environment Objectives

Focus on the business problem:
> IBM Cúram Social Program Management™ minimizes the amount of non-business-specific functionality that needs to be developed. Because developers are freed up from some of the more tedious and error-prone aspects of client/server development, they can spend more time on activities directly concerned with the business solution.

Model-based development:
> The starting point for all IBM Cúram Social Program Management™ software development is a platform-independent application model, developed using the Unified Modeling Language (UML).

Code generation:
> Many parts of an application are formulaic in nature, and can be expressed concisely through stereotypes and patterns in the application design. By adhering to a model-driven approach, the tools provided with the environment maximize the amount of code that can be generated.

Avoidance of platform dependencies:
> The IBM Cúram Social Program Management™ architecture packages specific platform dependencies in generated code and infrastructure components, minimizing the effects of changing them at a later stage, and thus maximizing the architecture's portability.

Use of recognized architectural patterns:
> The IBM Cúram Social Program Management™ architecture makes extensive use of patterns, such as factory and proxy mechanisms, to enhance application maintainability, performance and portability.

Simplified user interface development:
> Considerable effort can be expended producing quality user interfaces. IBM Cúram Social Program Management™ generates user interfaces based on simple platform-independent definitions, resulting in a dramatic increase in developer productivity.

Industry-standard applications:
> IBM Cúram Social Program Management™ facilitates easier application development, producing robust client/server applications based on industry-standard technologies. Runtime performance, cross-platform deployment, and user interface elegance are key goals of IBM Cúram Social Program Management™ application development.

3.3 Using the Environment

IBM Cúram Social Program Management™ development generally utilizes the following broad approach to development, represented by the following linked cycles:

- Modeling
- Implementation
- Building
- Deployment

These cycles are described in more detail in the following sections.

3.4 Modeling with Cúram

The starting point for all development is the Platform-Independent Application Model. IBM Cúram Social Program Management™ applications follow a service-oriented architecture, and services to be provided by the application are defined as UML interfaces in the model. Lower-level services are also defined here, resulting in an application that uses a layered approach. All interfaces in the model are referred to as „business objects" . When we need to distinguish between services that are only consumed internally by the application and services which are visible to external applications and user interfaces, we refer to the latter as „facades" . These define the outside world's view of a IBM Cúram Social Program Management™ application. Internal services are provided by a combination of „business process objects" and „business entity objects" . Entity objects define the "things" modeled by the application. Entity objects support data access operations to persist and retrieve instances of entities.

It is important to remember that the application model is platform-independent. No particular middleware or component technology (such as EJB) is referenced in the model. The model simply defines service interfaces and which subset of those interfaces will be made externally visible. The IBM Cúram Social Program Management™ environment looks after middleware dependencies by automatically generating any required „plumbing" code. Developers generally do not need to be concerned with the intricacies of middleware interfaces.

3.5 Implementation

After modeling the required business objects, the developer provides an implementation of the interfaces that have been modeled. All of the source code interface definitions and some of the implementation is provided by the IBM Cúram Social Program Management™ environment. For instance, data access operations on entities are generated for free. The developer focuses on coding just business logic. Business object implementations are coded by developers as plain Java™ objects (POJOs). The use of code-generated business interfaces and abstract classes ensures that the developer must provide the necessary implementations with the correct model-enforced interfaces.

3.6 Building

The IBM Cúram Social Program Management™ environment provides facilities for building developed applications. Scripts are provided to invoke code generators which analyze the application model and generate the skeleton of the target application. This code generated output is compiled, along with handcrafted developer implementations of business objects. Any necessary middleware tools required to build the final application are also automatically invoked.

Developers do not have to be concerned about dependencies between various source code and compiled application artifacts, as this is looked after by the supplied scripts.

3.7 Deployment

IBM Cúram Social Program Management™ online applications are enterprise-scale client/server applications. Production deployment of such applications can be a complex task, involving many configuration options for deployment across multiple network nodes, database tuning, security set up, and so on. Developers generally require a much simpler deployment configuration in which business objects can be tested as part of an iterative development cycle.

Several deployment alternatives are provided for business objects under test. A single-node EJB deployment allows the application to be deployed in the organization's chosen application server environment. However, an even simpler online setup involves the use of RMI-IIOP middleware for IBM Cúram Social Program Management™ client/server communications. Since RMI is built into all Java Standard Edition™ implementations, no application server software is required in this configuration. This simplifies the deployment of applications for testing purposes, as well as reducing the need for application server licenses.

IBM Cúram Social Program Management™ business objects can also be invoked asynchronously through the use of messaging middleware. This style of invocation underpins the IBM Cúram Social Program Management™ workflow facility which allows „process definitions" to be composed of multiple steps including business object execution and interactions with online users.

Finally, business objects can be deployed „standalone" , which allows for easy batch mode operation or batched unit testing. In this configuration, the target program can simply be executed from the command line, or from a script.
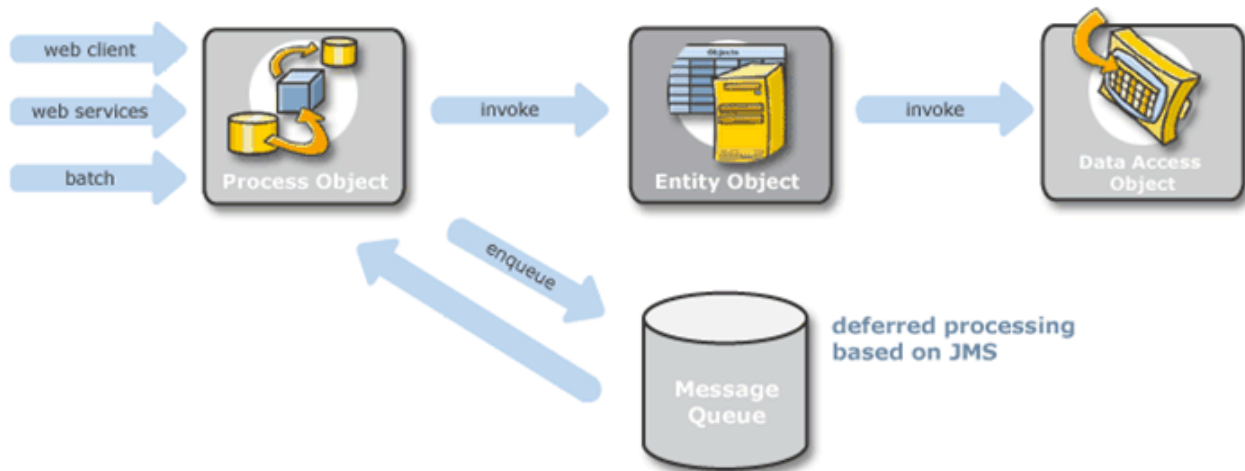

Abbildung 2. Invocation Options

Kapitel 4. Runtime Architecture

4.1 Einführung

At a logical design level, IBM Cúram Social Program Management™ is platform independent. The concrete realization of a IBM Cúram Social Program Management™ application must, of course, be deployed in some „real world" environment. As already mentioned, business objects can be deployed in a variety of ways. One of the more important (and complex) environments is the online client/server environment. Online applications follow the Java EE™ architecture. This is a modern n-tier architecture with separate *Presentation* , *Application* and *Persistence* tiers. This logical three-tier architecture has become the standard for developing client/server applications. The separation of presentation, application logic, and persistent storage allows the very different concerns of these tiers to be considered in relative isolation and promotes easier design. IBM Cúram Social Program Management™ simplifies this concept even further by hiding much of the complexity of n-tier application development. This chapter looks at what's „under the hood" in the runtime architecture of IBM Cúram Social Program Management™ online applications.

4.2 The Relationship to Java EE Architecture

Consider the Java Enterprise Edition™ (J2EE) architectural layers (see the Oracle J2EE 1.4 tutorial for more detail):

- Client-side presentation:
  - Browser (HTML, Java Applet™)
  - Desktop (Java™ application)
  - Other devices (J2EE™ client)
- Server-side presentation:
  - Web Server (JSP, Java Servlet™ , J2EE™)
- Server-side business logic:
  - EJB Container (EJBs, J2EE™)
- Enterprise Information System (EIS):
  - Various databases

The IBM Cúram Social Program Management™ client-side presentation tier consists of HTML user interfaces rendered by a standard browser program on the user's desktop. Only pure HTML user interfaces are directly supported using client generation tools. Other types of clients could also be developed using generated Server Access Beans to connect to the server.

At runtime, the HTML user interface is generated by a server-side presentation layer consisting of Java™ Server Pages. Browser clients communicate with this layer over http, typically encrypted using SSL for security reasons.

The server-side presentation layer communicates with the server-side business logic using the RMI-IIOP protocol. IBM Cúram Social Program Management™ typically presents business objects in the business logic tier as Session EJBs although, as mentioned previously, they can also be simple Java™ RMI objects for the simpler deployment option often used during application development. In any event, business objects are ultimately plain Java™ objects (POJOs) with the middleware plumbing filled in transparently during the application build.

The back end of the IBM Cúram Social Program Management™ architecture is a relational database as well as other enterprise and legacy applications. Again, the middleware „plumbing" required to communicate with the EIS is generated.

4.3 Summary of Java Technologies Used

EJB Enterprise Java™ Beans
> IBM Cúram Social Program Management™ uses Enterprise Java™ Beans for its server component model.

Java™ Servlets
> Java™ Servlets are used by the presentation tier.

JSP Java™ Server Pages
> Java™ Server Pages are used to generate the user interface.

JTA Java™ Transaction API
> Java™ Transaction API is used for starting and committing transactions.

JDBC Java™ Database Connectivity
> Java™ Database Connectivity is used for the middleware to communicate with the application database.

JMS Java™ Message Service
> Java™ Message Service is used for deferred processing and workflow within IBM Cúram Social Program Management™.

JNDI Java™ Naming and Directory Interface
> Java™ Naming and Directory Interface is used in IBM Cúram Social Program Management™ both for application initialization-time lookup of Data Sources and Queues as well as to locate Enterprise Java™ Beans from the Presentation Tier.

RMI-IIOP Remote Method Invocation
> Remote Method Invocation over IIOP is used as the communications protocol between the presentation and application tiers.

4.4 Java EE™ Design Patterns

Many of the Oracle Java EE™ Design Patterns are used by IBM Cúram Social Program Management™.

The Struts™ framework used in the IBM Cúram Social Program Management™ presentation tier utilizes the Front Controller pattern. Generated Server Access Beans in the presentation tier implement the Service Locator pattern to hide the complexities of locating server-side objects.

The IBM Cúram Social Program Management™ application server uses the Session Façade pattern to provide coarse-grained access to business logic. Value Objects are used to communicate between different business objects on the server, and between the server and the web tier. The Data Access Object pattern is used for all database accesses in the application server.

4.5 Presentation Tier Runtime Architecture

At runtime, the client user interface is generated by Java™ Server Pages (JSPs) compiled as servlets. These servlets run in a web container that sits between the browser client and the back-end IBM Cúram Social Program Management™ application server.

The Browser Client, which is a standard internet browser (e.g. Microsoft® Internet Explorer ™), displays the HTML pages to the user. Events generated by user interaction result in a HTTP request being sent to the web server. This is a standard web server such as IBM® HTTP Server ™ or Apache Web Server™ . Here, a JSP generated by the Cúram environment is converted into a Java™ Servlet which accepts the relevant parameters from the web server and calls the application server.

The IBM Cúram Social Program Management™ application server uses generated Server Access Beans to call appropriate server business logic, in addition to providing data conversion logic. It passes back the

result and the servlet that handles this result generates the new HTML page and passes it back to the web browser via the web server.
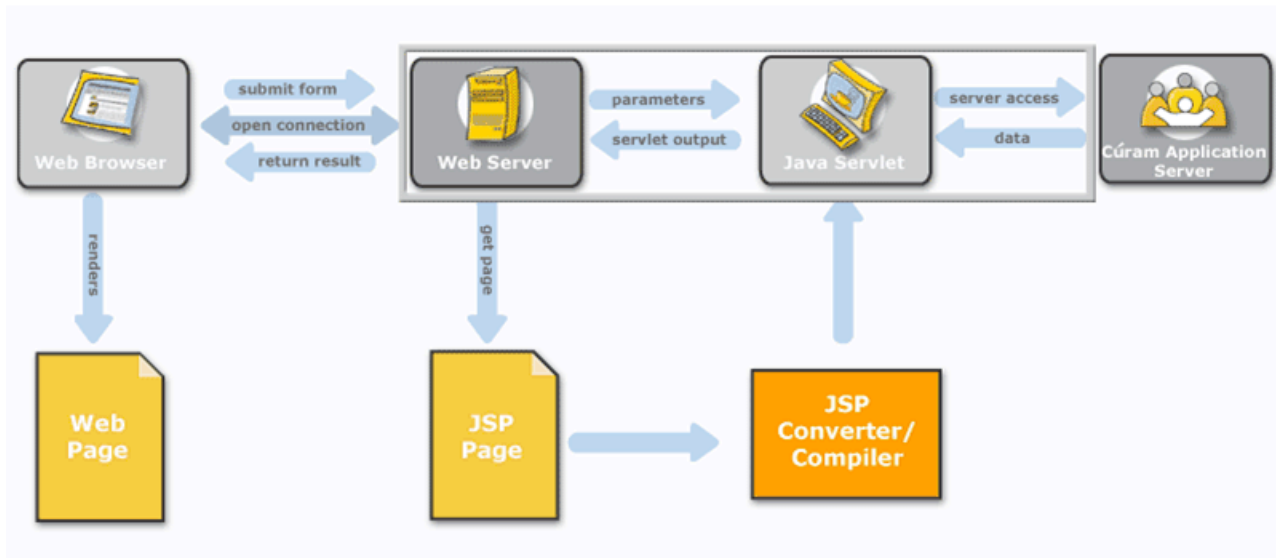
Abbildung 3. Client Runtime Architecture

The following more detailed view of the Web tier shows that client requests are intercepted by a controller servlet, which dispatches to the appropriate JSP servlet. Server Access Beans perform the job of converting data from HTML forms to an appropriate representation for transmission to the back-end application server. This stage of processing also includes data validations which have been previously defined in the application model. Results from the application server are formatted for display, and „list data" may optionally be sorted in a user-defined order.

Abbildung 4. Client Runtime Architecture (Level 2)

4.6 The Struts Framework
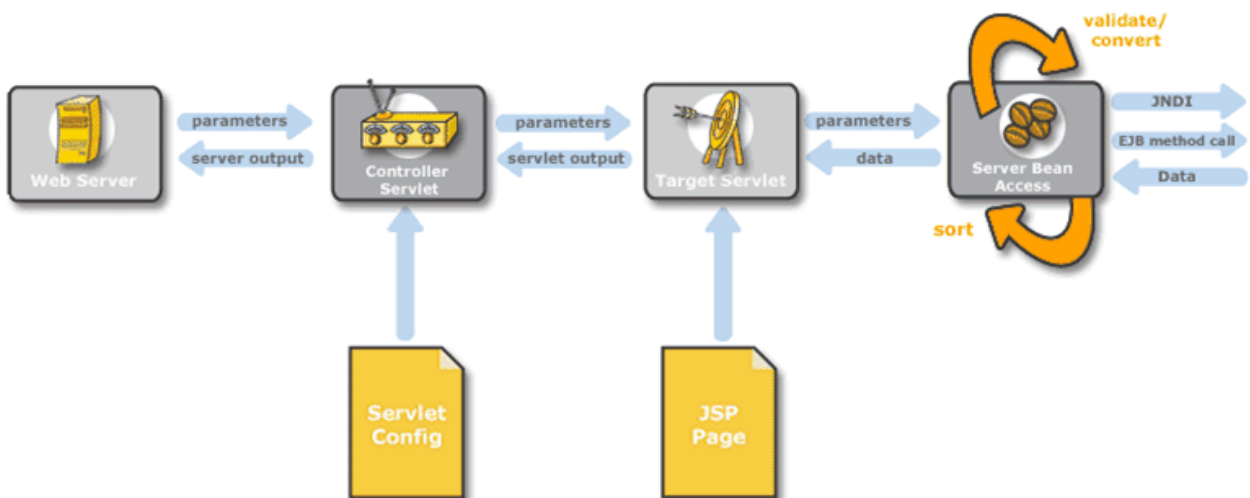
The Apache Struts Framework™ is an open source framework for building web applications based on standard technologies such as Java™ Servlets, Java™ Beans, ResourceBundles and XML. It encourages application architectures based on the Model 2 approach (a variant of the Model View Controller (MVC) design pattern). Struts™ provides its own Controller component and integrates with other technologies to provide the Model and the View. For the Model, Struts™ can interact with any standard data access technology, including Enterprise Java™ Beans, JDBC, and Object Relational Bridge. For the View, Struts™ work well with Java™ Server Pages, Velocity Templates, XSLT, and other presentation systems.

IBM Cúram Social Program Management™ uses Struts™, but generates most of the required components. Some custom JSP tags are provided with IBM Cúram Social Program Management™ and uses its own Server Access Beans, rather than Struts™ form beans.

Struts™ provides the front controller component and mandates the `struts-config` file format.

4.7 Business Logic Tier Runtime Architecture

The IBM Cúram Social Program Management™ application server can be logically split into three layers. A Remote Interface Layer handles the details of interacting with middleware. The Business Object Layer contains application business logic, implemented as plain Java™ classes. The Data Access Layer performs all interaction with the application database, or with other Enterprise Information Systems.
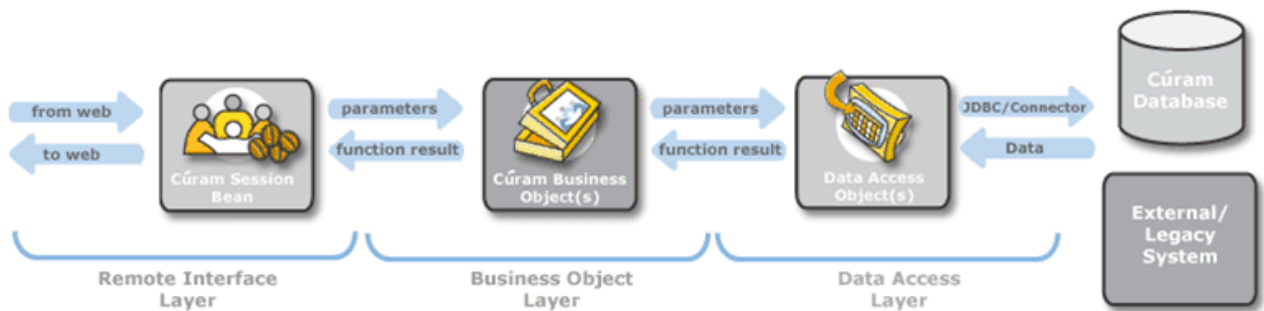

Abbildung 5. Server Runtime Architecture

4.8 Remote Interface Layer

The Remote Interface Layer controls various aspects of client/server interaction. It deals with middleware concerns that are outside the focus of the application's business logic and effectively shields the business logic code from the knowledge that it is participating in a client/server environment. The functions of the Remote Interface Layer are:

Business Object Life-cycle Management
Creating/locating appropriate Business Objects to service client requests;

Berechtigung
Checking that a client is entitled to execute a particular server function;

Transaction Control
Starting, committing, and rolling back atomic transactions;

Argument marshaling
Translating between different data formats involved in client/server transmission;

Last-chance exception handling
Handling exceptions that are not properly handled by application code, and logging them to a central location, as well as performing uniform reporting to client applications.

4.9 Business Object Layer

The Business Object Layer contains all of the application business logic including „Process" objects which implement the business-specific functionality and „Entity" objects which indirectly perform data accesses.

This is the only layer of the IBM Cúram Social Program Management™ architecture containing logic directly coded by application developers. Certain patterns are used by convention in „out-of-the-box" applications. The Business Object layer is further subdivided into Façade and Service Layers. The various types of objects in the following diagram are collectively known as „business objects" :

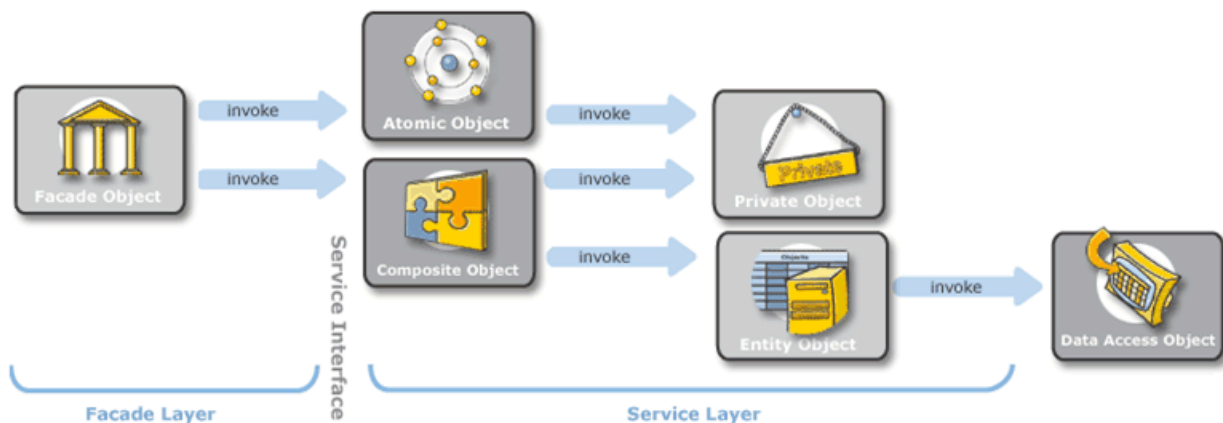The interaction between the Facade Layer and the Service Layer is shown in the following diagram

Abbildung 6. Business Object Layer

4.10 Facade Layer

A Facade layer is a higher-level business component that contains and centralizes complex interactions between lower-level business components:
- The Facade layer represents the public (remote) interface to the server;
- Facade objects may perform interface-specific processing;
- Facade objects sequence calls to a Service Layer
- Façade objects are „process" objects.

4.11 Service Layer

The service layer contains core IBM Cúram Social Program Management™ business processes. It uses the following process types within the Service Layer:
- „Atomic" processes are the lowest level units of processing that leave the server in a consistent state.
- „Composite" processes are convenience processes that sequence multiple atomic operations.

4.12 Factory Methods

IBM Cúram Social Program Management™ uses Factory Methods to provide an interface for creating business objects without specifying their concrete classes. All business objects (process and entity) are created by calling factory methods.
- Factories optionally support a very fine-grained trace facility using Java™ dynamic proxies.
- Factories are useful for debugging and understanding flow of control.

4.13 Connectors

IBM Cúram Social Program Management™ Connectors are a set of tools which facilitate Application Integration. They allow application developers to develop with objects representing data obtained from, or persisted on, legacy and other application platforms in exactly the same way that they use other IBM Cúram Social Program Management™ Business or Entity Objects.

Connectors also remove the necessity of writing „low level" code in order to access legacy systems. The mechanism of communication with the legacy platform is hidden from the developer.

Connectors are provided for:

- IBM® MQSeries® ™ ;
- Web services; and
- SQL.

Kapitel 5. Development Architecture

5.1 Einführung

The IBM Cúram Social Program Management™ Development Environment is composed of a Server Development Environment and a Client Development Environment.

The IBM Cúram Social Program Management™ development approach has the following key features:

- Metamodel-based development approach;
- Application model based on UML;
- Code generators that produce significant portions of client/server applications;
- Simplified coding of handcrafted business logic;
- Simplified development of user interfaces;
- High-level business infrastructure.

5.2 Server Development Environment

The Server Development Environment uses a model-based approach to development. An application model is defined in a business-centric and platform-independent manner using a UML modeling tool. The model is the key building block for the code generators as it defines all the required entity objects and process objects.

The generators will create the necessary classes and files for the application structure. This structure will also have all the Remote Interface Layer code, the Data Access Layer code and the Business Object Layer code as well as handcrafted code.

5.3 Application Model

The Application Model is the basis for all server development and consists of the following elements:

Domains:
    Application-specific datatypes;

Entities:
    The objects modeled and persistently stored by the application;

Processes:
    Related sets of activities to achieve some business goal;

Value objects:
    Passed as messages throughout the application.

5.4 Application ModelDomains

Application model domains are defined in terms of a „fundamental" datatype (string, integer etc.), or another domain. They have application-specific type names such as „SOCIAL_SECURITY_NUMBER" or „PAYMENT_AMOUNT" , and collectively form a tree called the „domain hierarchy".

Domains can have associated validations like UPPERCASE, range checks, code tables, pattern matches or custom validations. Attributes of entities and value objects are specified as domains.

The following illustrates the inheritance hierarchy of domain classes starting at a point called "root", with each lower-level, indented bullet inheriting from the preceding higher level:

- root
  - STRING
    - FIRSTNAME
    - SURNAME
    - ADDRESSLINE
  - INT_64
    - UNIQUE_ID
      - PERSON_ID
  - DATE
    - DATE_OF_BIRTH

5.5 Application ModelEntities

Entities have attributes which are defined as domain types, and can have primary and alternate keys, and foreign key relationships.

No implementation is required for „Create, Read, Update, Delete" (CRUD) style operations specified as stereotyped methods in UML. Other business methods can also be used whose signatures are defined in UML. Complex database operations can be specified in SQL.

The persistent store has pre and post access Exit Points.

5.6 Application ModelProcesses

Arbitrary business functions are represented within the IBM Cúram Social Program Management™ model as methods of process classes. Method arguments are modeled as Value Objects.

The model defines the interface, but not implementation, of process objects.

5.7 Server Code Generation

A platform-specific model is automatically generated from the platform-independent metamodel and then code is automatically generated for an EJB server application. The main generated artifacts of the server code generation process are shown below.
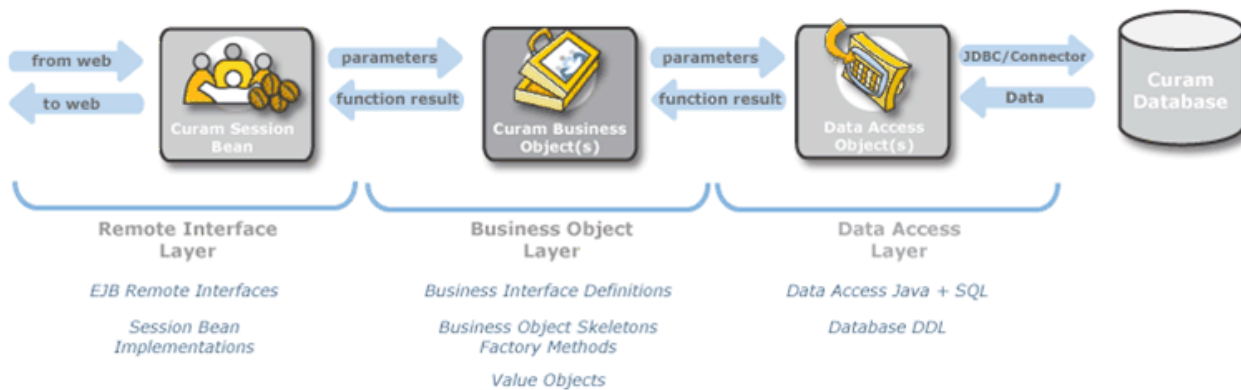

Abbildung 7. Server Code Generation

5.8 Client Development Environment

5.9 Übersicht

The IBM Cúram Social Program Management™ client consists of HTML Pages generated by JSPs which are in turn generated from XML screen definitions, with Style Sheets to provide formatting of screen pages. The XML screen definitions are presentation layer independent, and the IBM Cúram Social Program Management™ specific format is known as UIM (or User Interface Metadata).

Automatic data validation/conversion is based on application model definitions with support for custom widgets and JavaScript Exit points.

5.10 User Interface Development

A major goal of the IBM Cúram Social Program Management™ development environment is to simplify user interface creation. Client „pages" are associated with particular back-end server interfaces, and since metadata about these interfaces has already been captured in the application model this can be leveraged to provide much of the information required for user interface generation. Much of the remaining task for client developers is to list the fields that should appear on a given client page. Default „widget" types are provided for fields and controls on the screen, based on its knowledge of the datatypes associated with fields.

Fields follow a grid layout in „clusters" and „lists" specified in XML along with the overall „page hierarchy" . Widget types are determined automatically by connections.

An example of the User Interface Metadata (UIM) code for the „firstname" field is as follows:

```
<FIELD LABEL="Field.Label.FirstName">
<CONNECT>
<SOURCE NAME="Interface1" PROPERTY="firstForename"/>
</CONNECT>
a
</FIELD>
```

The LABEL attribute of the FIELD element describes the label text that will be associated with this field when a client page is displayed. The value „Field.Label.FirstName" is a reference to the actual label value in a separate property file. The use of strings externalized in property files allows for easy localization of client applications. The SOURCE element describes where this field's contents comes from when the page is displayed. The NAME attribute of the SOURCE element specifies a specific back-end interface defined in the application model. The PROPERTY attribute specifies a particular datum returned by the back-end interface.

Note that there is no specification of exactly where on the screen the field should be displayed, nor is their any information about the field's datatype or the HTML control that will be used to display it. All of this information is filled in automatically at application build time. The simplicity of UIM pages makes them very easy to copy and paste from templates, making for a high level of developer productivity.
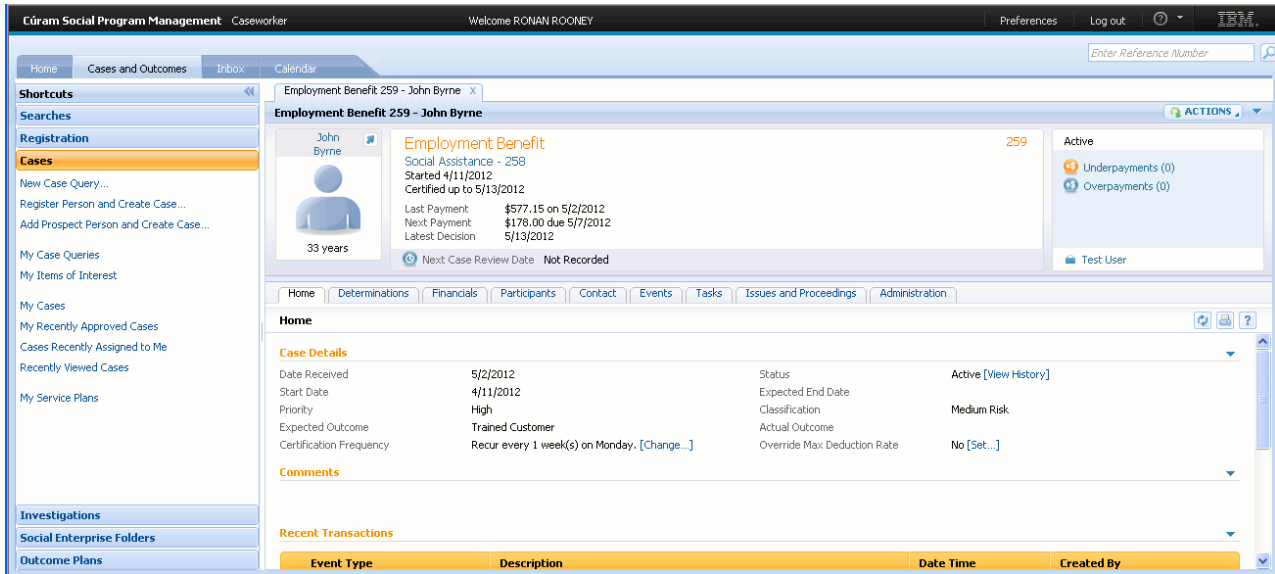
Abbildung 8. Sample Cúram User Interface

5.11 Client Code Generation

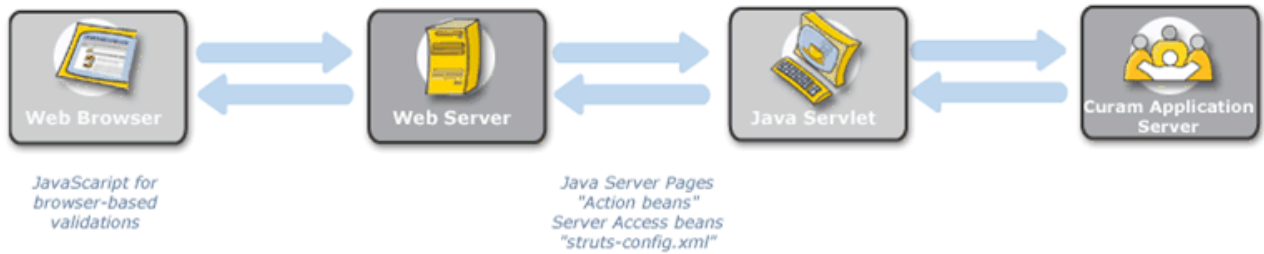The following diagram shows the main generated artifacts of the client code generation process.

Abbildung 9. Client Code Generation

Kapitel 6. Business Infrastructure

6.1 Einführung

IBM Cúram Social Program Management™ consists of a comprehensive business infrastructure functionality, which includes the following elements:

- The Rules Development Environment;
- XML & Printing; and
- Workflow.

In addition to these services, a complete façade-based authorization infrastructure is also provided.

6.2 Rules Development Environment

Cúram Express Rules, or CER for short, is a language for defining questions that can be asked, and the rules for determining the answers to those questions. Each question specifies:

- its name;
- the type of data which provides the answer to the question; and
- the rules for providing the answer (if the question is asked).

The answer to a question can be as simple as yes or no, e.g. the question "Is this person eligible to receive benefits?"; however, CER lets you define answer types to be as complex as you need, e.g. the question "Which groups of people in the household have an urgent need?" would be answered by providing a list of household groups, with each household group containing a list of people.

The environment for developing CER rules is the CER Rules Editor. This editor provides a user-friendly environment and interface for both technical and business users to create, edit and validate a rule set and its rule classes. For more information on the CER Rules Editor, please see the *CER Reference Manual* and the *Working With CER Guide*.

6.3 Eligibility and Entitlement Processing

The CER Rules Engine provides a mechanism for determining eligibility and entitlement on a case. The engine acts on the rules defined in the CER Rules Editor and the data, or evidence, captured on a case.

The flexibility of CER rules means they can used to define display rules, which is a way of conveying to a business user what has happened on the case. These rules can be as terse or as verbose as they need to be.

More on the CER Rules Engine, and the determination of eligibility and entitlement, can be found in the *Inside Cúram Eligibility* and *Entitlement Using Cúram Express Rules* guides.

6.4 XML and Printing

IBM Cúram Social Program Management™ print processing is based on merging data with document templates to produce the final document output. Document templates are developed in XSL (Extensible Stylesheet Language), and stored on the application database. Business processes that produce printed output produce data in the form of an XML document, and then send the data to a print server along with the id of the appropriate stylesheet used to format that data.

Producing XML data in IBM Cúram Social Program Management™ processes is easy as any „value object"
can be converted into XML data. Since all communication between business processes and data access
objects is in the form of value objects, effectively any data can be produced in XML form as easily as accessing the application database.

The IBM Cúram Social Program Management™ XML server can produce PDF™ , RTF, HTML or text from
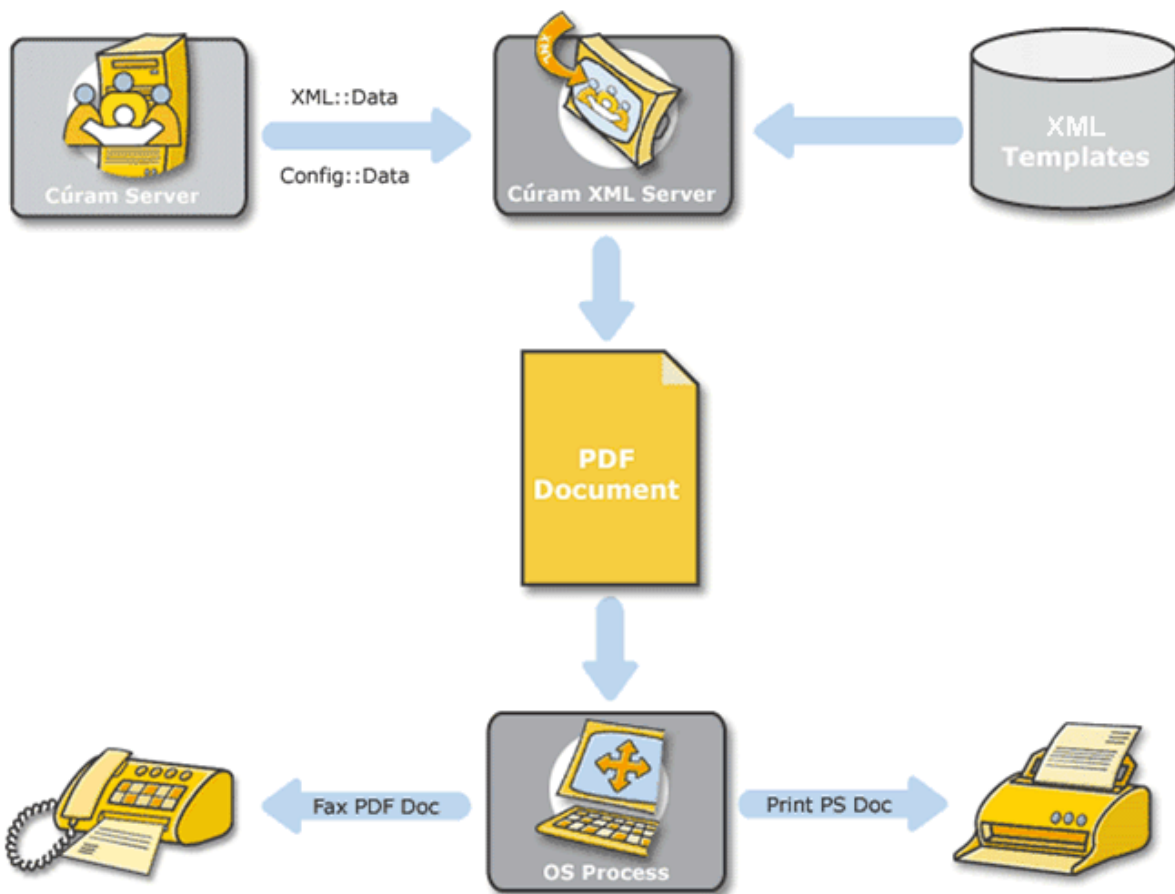the XML/XSL definitions.

Abbildung 10. XML and PDF Printing

6.5 Workflow

The IBM Cúram Social Program Management™ workflow management system, which is based on the
Workflow Management Coalition's standards, allows organizations to break down business processes into
their constituent activities and to then build flexible relationships between them. Procedural automation
of a business process is done by managing the sequence of work activities and the allocation of appropriate human and/or system resources associated with the various activity steps.

The IBM Cúram Social Program Management™ Workflow Management System comprises an interactive
Process Definition Tool used to define the workflow activities and transitions, and a Workflow Engine
which manages the workflow in the production environment. It also includes an administration component for maintaining instances of workflow process definitions.

This approach ensures that organizations can implement and manage their processes in a flexible manner and maximizes their ability to respond to change.

Workflow activities may involve an automated step (invoking a Cúram Business Process), or a manual step (creating a task for a user to perform manually).

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM-Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden. Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing

IBM Europe, Middle East & Africa

Tour Descartes 2, avenue Gambetta

92066 Paris La Defense

France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden.

Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen. Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht. Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Bereitstellung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen.

IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können u. U. von den hier genannten Preisen abweichen.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellensprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Musterprogramme werden "WIE BESEHEN", ohne Gewährleistung jeglicher Art bereitgestellt. IBM übernimmt keine Haftung für Schäden, die durch Ihre Verwendung der Musterprogramme entstehen.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

Informationen zu Programmierschnittstellen

In der vorliegenden Veröffentlichung werden vorgesehene Programmierschnittstellen dokumentiert, mit deren Hilfe Kunden Programme für den Zugriff auf IBM Cúram Social Program Management-Services schreiben können.

Marken

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/us/en/copytrade.shtml.

Apache ist eine eingetragene Marke der Apache Software Foundation.

Microsoft and Internet Explorer are trademarks of Microsoft Corporation in the United States, other countries, or both.

Oracle, Java and all Java-based trademarks and logos are registered trademarks of Oracle and/or its affiliates.

Andere Namen können Marken der jeweiligen Rechtsinhaber sein. Weitere Firmen-, Produkt- und Servicenamen können Marken oder Servicemarken anderer Unternehmen sein.

IBM ®

Gedruckt in Deutschland