



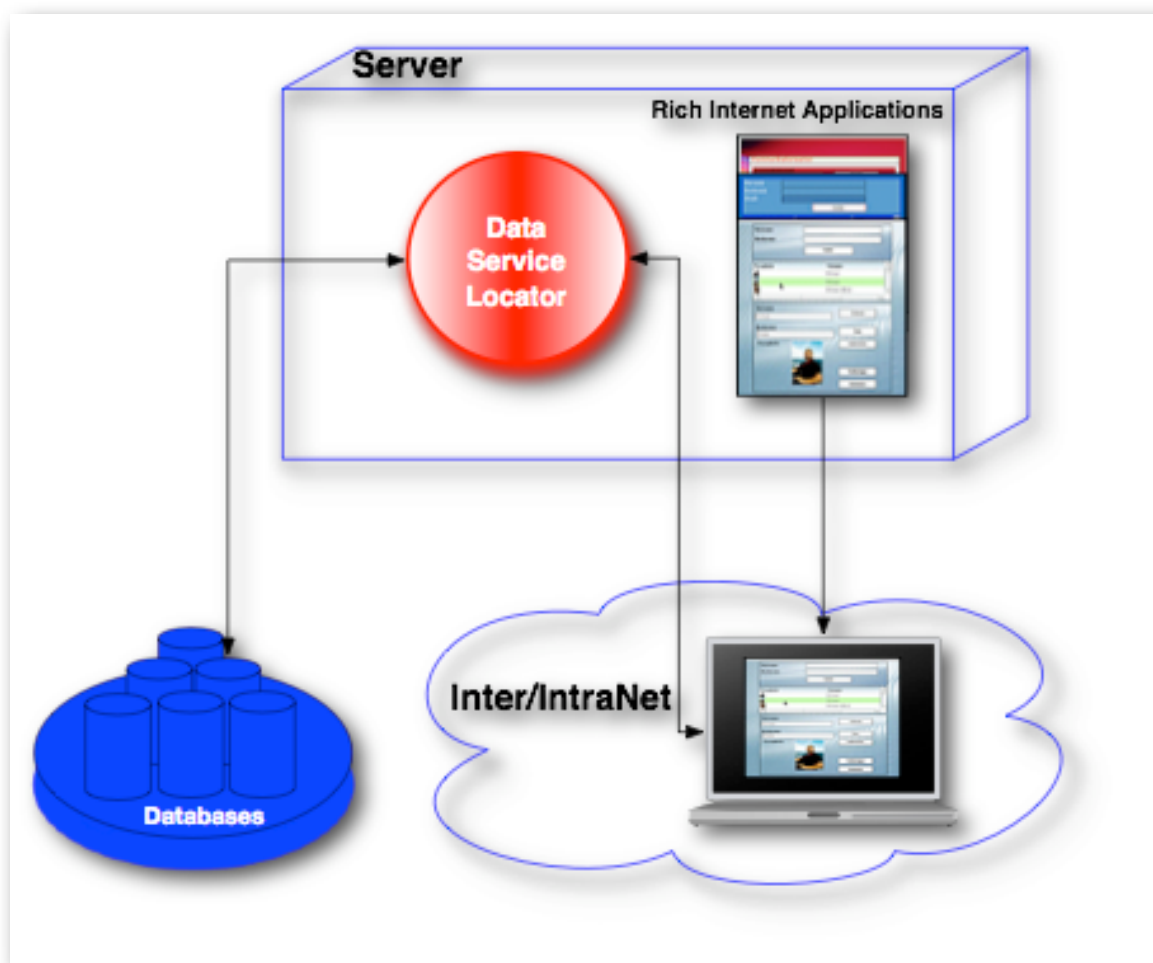
Contents

"Rapid RIA Development" Technology	2
Database Accessibility	3
Database Abstraction	4
Composing instead of Programming	5
Force4 Product Overview	5
Supported Technologies	5
Workflows	6
Modeler	11
Composer	12
The Server Side	13
Force4 Feature Summary	14
Force4 Advantages	15

"Rapid RIA Development" Technology

Force4 is a development system and framework for the "Rapid Development" of Rich Internet Applications for the connection of databases to different presentation technologies – but how does the technique function? This whitepaper describes the idea, the system requirements and the implementation of Force4 in detail.

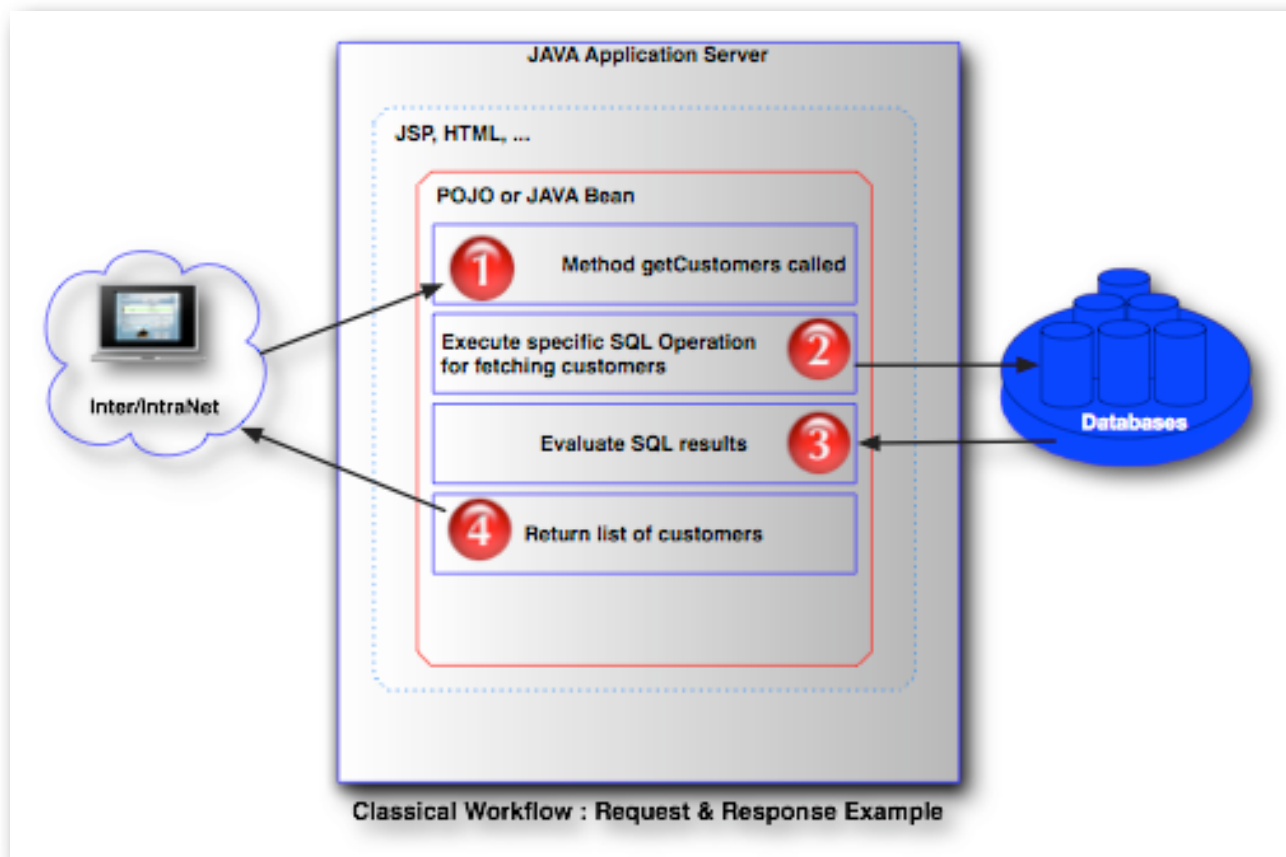
When developing Rich Internet Applications (RIA), and thus replacing Web sites by intelligent clients and offering functional and rich user interfaces, it soon becomes clear that an attractive user interface of the application is only one aspect of the RIA. Only good functionality will help the user to accept the application. This functionality is created by business logic and access to backend systems, in most cases a relational database (RDBMS).



The current state-of-the-art presentation technologies for RIAs are able to connect to backend systems at a basic level, however, a fully-fledged connection to a RDBMS still demands a lot of programming work. Force4 comes in at exactly this point! Force4 offers the developer a tool for both the connection of RIAs to database systems easily without any programming requirement and to provide the usage of their functional possibilities as complete rich internet applications. The idea behind Force4 was to develop a tool, which would be able to create web-client-based, multi-tier applications without any programming requirements for the developer. In addition, the system had to allow for the possibility to enrich automatically-created applications with additional business logic and design elements. Thus Force4 applications can be used for rapid prototyping as well as finished applications.

Database Accessibility

The classic way to use RDBMS back-end systems directly from a RIA, for example Macromedia Flex or Open Laszlo (both solutions imbedded in J2EE Server), is the usage of JAVA Plain Objects or JAVA Beans, which are programmed as Business Objects. These objects execute certain database operations and deliver the results, under certain circumstances enriched with business logic, back to the web client, for example to the Macromedia Flash Player.



This method of accessing data is derived from the current available application server solutions such as: JSP, ASP and .Net etc. Here the complete logic of a web application is executed on the server side and HTML is used as the presentation layer, which is, as we all know, very limited in its functionality. This division is in accordance with the N-Tier approach. Modern intelligent web clients (Macromedia Flash, Raining Data OMNIS) grant the dislocation of the complete business logic or just a part of it on to the client. Thereby allowing more programming facilities and the ability to run the whole application independently. The OMNIS Web Client also allows for parts of the application to be run on the server, without creating the common technology splicing (i.e. Flex MXML/ Action script and JAVA).

Naturally, a connection to a backend system will be required in order to be able to frequent the databases. Here the "proprietary" methods such as Macromedia's AMF or the OMNIS binary protocol and also Web services/SOAP are available. The proprietary methods are more commonly used because they have been created for optimal data transmission to particular clients and for enabling the uncomplicated implementation of persistence between server objects and their client representations. Web services, however, are the more universal instrument, but offering less performance and requiring more work in order to achieve the required persistence. The use of HTML, JSPs etc., serving as carriers of the RIAs for session control is not a topic of this paper, as Force4 concentrates on RIAs.

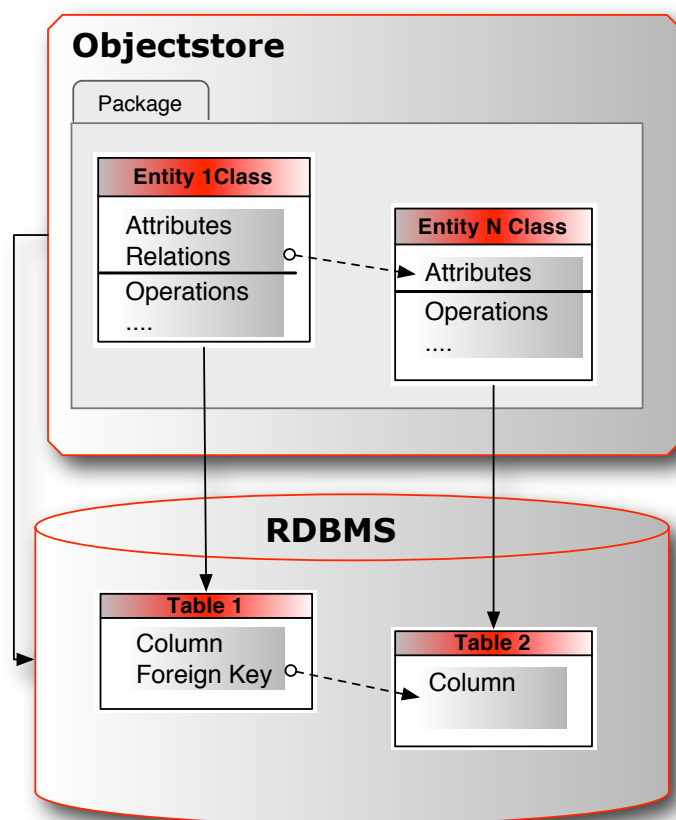
A requirement of Force4 was to be able to make a division between web-client-dependent communication methods and other Force4 components (persistent framework, presentation framework...). At the moment, Force4 (Version 2.0) contains a JAVA Bridge, which is served by the AMF

based communication points (Flex AMF, OpenAMF, OpenLaszlo Java RPC). The Force4 Omnis version will contain an Omnis bridge.

Database Abstraction

In many web applications the developer implements the database operations directly in SQL which are then executed on the server . In JAVA, JDBC or JNDI are used as endpoints, under OMNIS, DAMs used as SQL endpoints. The SQL operations are built into the various business objects and the results are forwarded to the clients. This creates static and inflexible server objects, which have to be changed every time a change to the database structure takes place (Not to mention the switching to other RDBMS products).

In the last couple of years, Object/Relational Mapping (ORM) environments have been developed, which allow the abstraction of databases into models. At the same time, universal frameworks have been created which have been built on top of these models. Thus making it possible to run operations on RDBMS without explicit knowledge of the underlying RDBMS and programming of SQL operations. These models describe entities and relations and their attributes.



As a result of the foregoing, Force4 had to be able to offer the web clients a framework which offered universal access to one or more databases by using ORM technology. And Force4 had to be able to create automatically (entity, relations presenting) objects from the ORM. In the case of JAVA, the developer should be able to create JAVA classes without reaching out for a JAVA IDE. The manual creation of object mapping of a database is a complicated process. Moreover Force4 had to be able to do both, create models and reverse engineering of models from legacy databases. This is the case because very often RIAs have to replace existing HTML based applications. These databases and their structure are already strictly defined.

Composing instead of Programming

The existing RIA development systems (FlexBuilder, Eclipse Plugins for Flex and Open Laszlo, OMNIS Studio) allow for the implementation and attribution of visual containers, controls, etc.. To give 'life' to the application, more programming in the native language of the web client is required. In database-driven applications, this would mean that queries, result handling, filling of master-detail relations, transactions etc., have to be programmed manually. This demands that the developer must have the documentation of his database at his disposal, parallel to the knowledge of methods which map the database operations. He must also program methods himself in order to maintain results and events. In addition, the creation of ORM maps or business objects must be considered. Of course, all the GUI elements of the application must be created and linked to each other.

The prerequisite of Force4 was to automate these tasks as far as possible. The developer should be able to execute, with least effort, all occurring tasks with the help of state-of-the-art GUIs. Thus a composer must contain both visual components and functional components in order to reduce the creation of a query form down to just a few mouse clicks.

Force4 Product Overview

Force4 combines the topics, as mentioned above, in one product. It integrates the creation and reverse engineering of database models, the creation of the necessary objects and the access classes and composing of Rich Internet Applications.

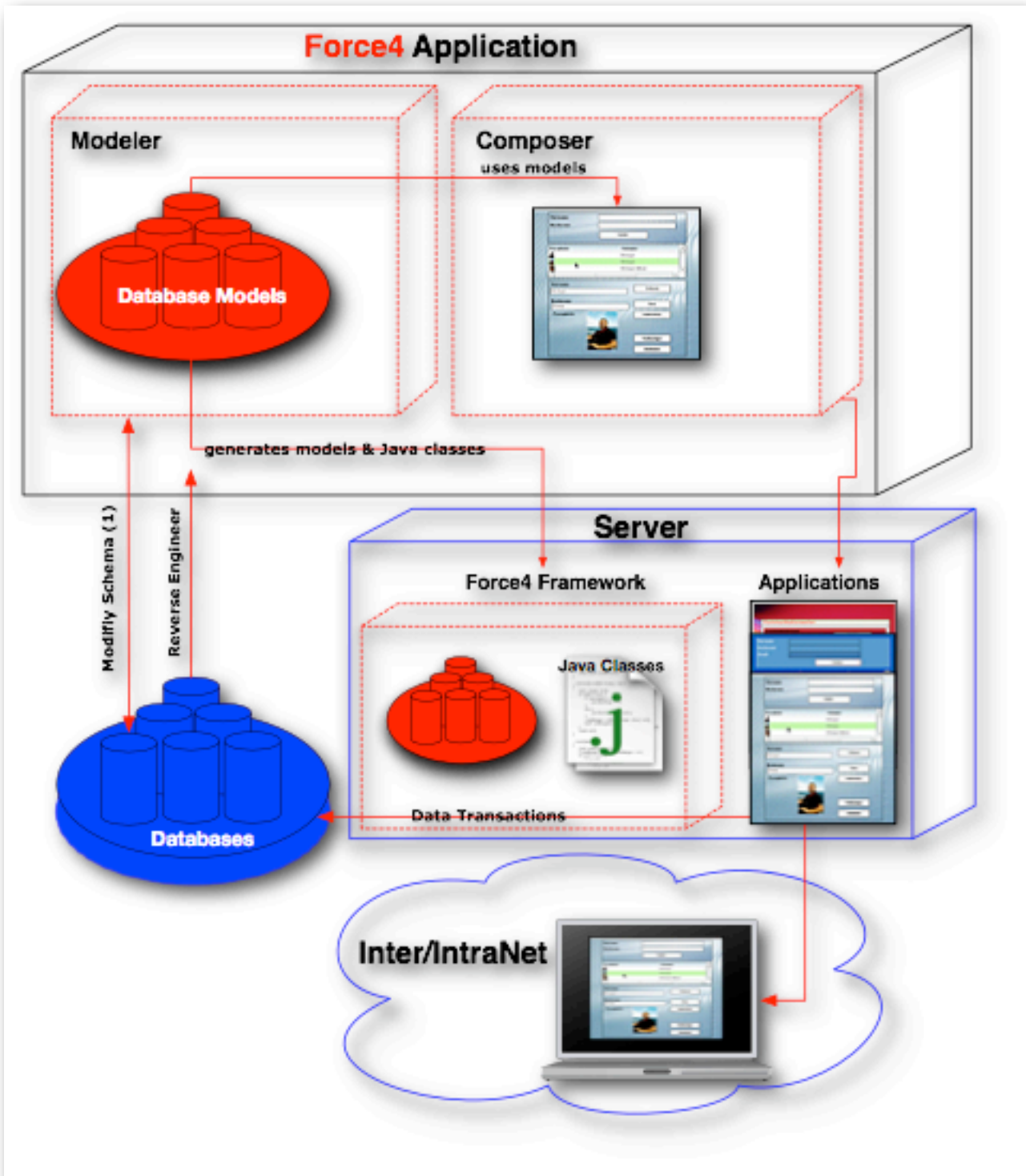
Supported Technologies

At the moment (July 2005), Force4 supports Macromedia Flex as presentation server and J2EE server as backend communication server. Soon Force4 will also support Open Laszlo as presentation server and RainingData OMNIS as presentation and backend client/server. With its backend frameworks, Force4 can communicate with all SQL-enabled database systems, if they contain a JDBC driver. The Force4 Composer/Modeler is available for Microsoft Windows 98, 2000, XP and Apple Mac OS X 10.3. The Force4 server components support Microsoft Windows based J2EE server at the present time.



Workflows

This illustration shows the Force4 workflow overview. There is a clear difference between the Force4 application and the Force4 server. The Force4 application is installed on the desktop computer and consists of the modeler and the composer. The Force4 server is the communications backend for the applications made by the Composer.



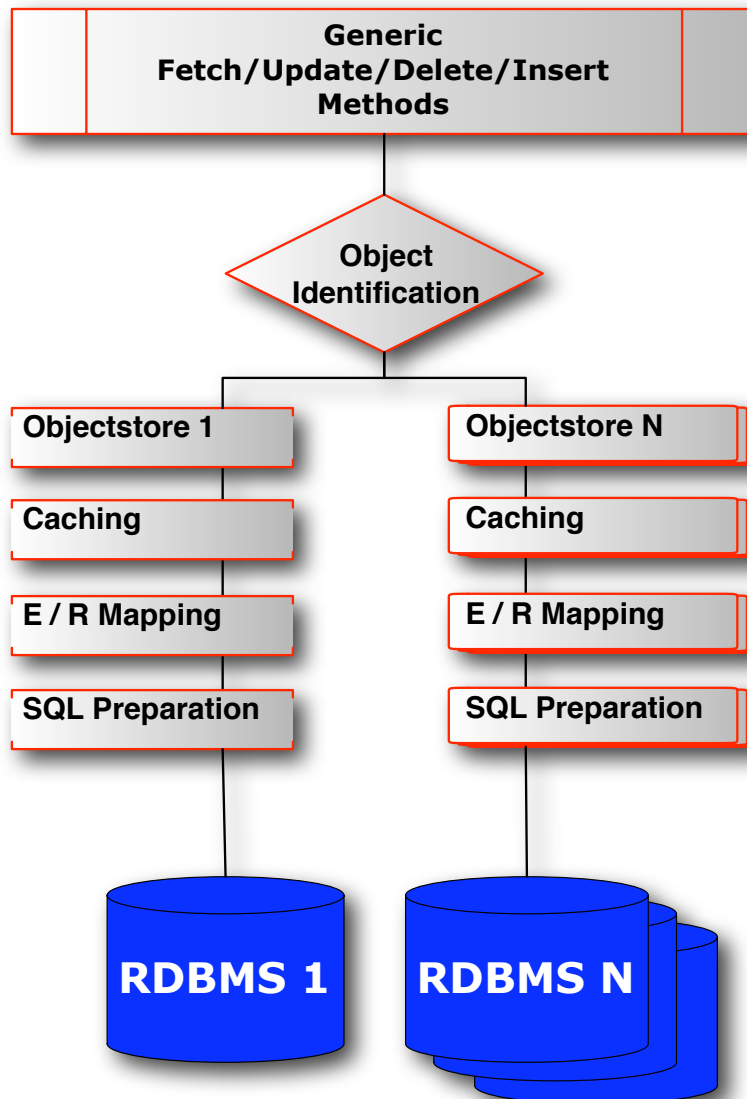
The creation of RIAs with Force4 is divided in the two parts; database modeling and application composing. The Modeler can do both, communicate with the database to read its structure (Reverse Engineering) and change its structure as well. The structures are saved in the models. These

abstract tables, views, columns etc. and their connections with each other into entities and relationships. In addition to that, the relevant properties of the database, such as available column types and JDBC parameters are saved. These models are both used by the composer to present the entities, attributes and relations and to connect them with visual and functional components.

If the Force4 Server is J2EE- based the models and the JAVA classes are saved on the server. They are used by the Force4 Server Framework to implement the object relational mapping. The modeler also creates both the classes and the access objects needed for the web client programming. These are referenced by the source code generated by the composer for the applications. These composer-generated applications are delivered from the presentation server to the web clients.

Databases

The connection from the web client to the database is done by the Force4 Server. The web client never communicates directly with the database, rather only through a unified interface. This interface is on the one part based upon the force4-generated models and on the other part on strictly defined functions. Please also refer to the topics: "Applications" and "Data Flow"..



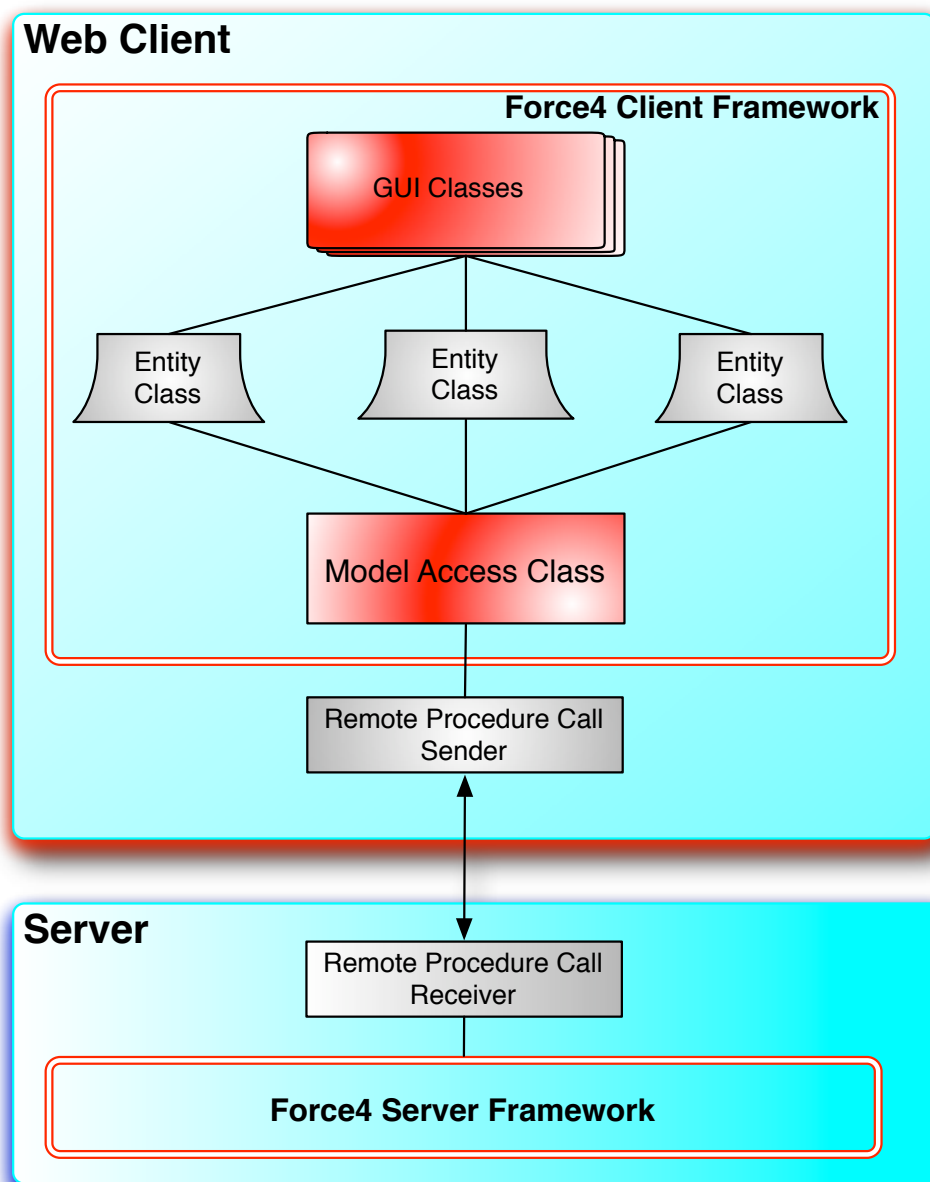
In the case of J2EE-based Force4 Server, RDBMS are generally addressed by JDBC. From the feature set of the specific JDBC driver results both the possibility of Reverse-Engineering and DDL creation since Force4 accesses the database directly through the java.sql package. These steps are exe-

cuted in the Force4 modeler. Runtime database accesses (e.g. selects, updates,...) are executed by the ORM Layer in the Force4 server component. The web clients call Java methods (in Flex over Remote Objects, in OpenLaszlo over JavaRPC), which obtain the data over the ORM Layer or send them to the database. The ORM Layer uses the Force4 Modeler created model to transform the abstract entities and relations used in the web client applications into database tables, columns and the required SQL dialect. The creation of primary keys needed for the insertion of new data is also executed. This can be configured in the modeler.

Primary keys are the foundation of object persistence between server and client. Although the client may have more than one instance of a database object available, on the server it is always the same one managed by the ORM Layer. Changes etc. to the server side objects are therefore directly available to all clients. The Force4 server ensures the integrity of the objects, even if several clients manipulate the same object. The ORM Layer takes care of both database integrity and also object caching. Thereby guaranteeing optimal performance and memory usage.

The above-mentioned results in the advantage that web applications need not know the real database construct (tables, data types, primary and foreign keys...) but only the abstract view. Thus it is possible to offer a unified framework for the applications. This solution allows database structures to be changed (to a certain extent) without having to change applications. Moreover the target database can be exchanged for another RDBMS System, as the Force4 Modeler can convert models between databases.

Applications

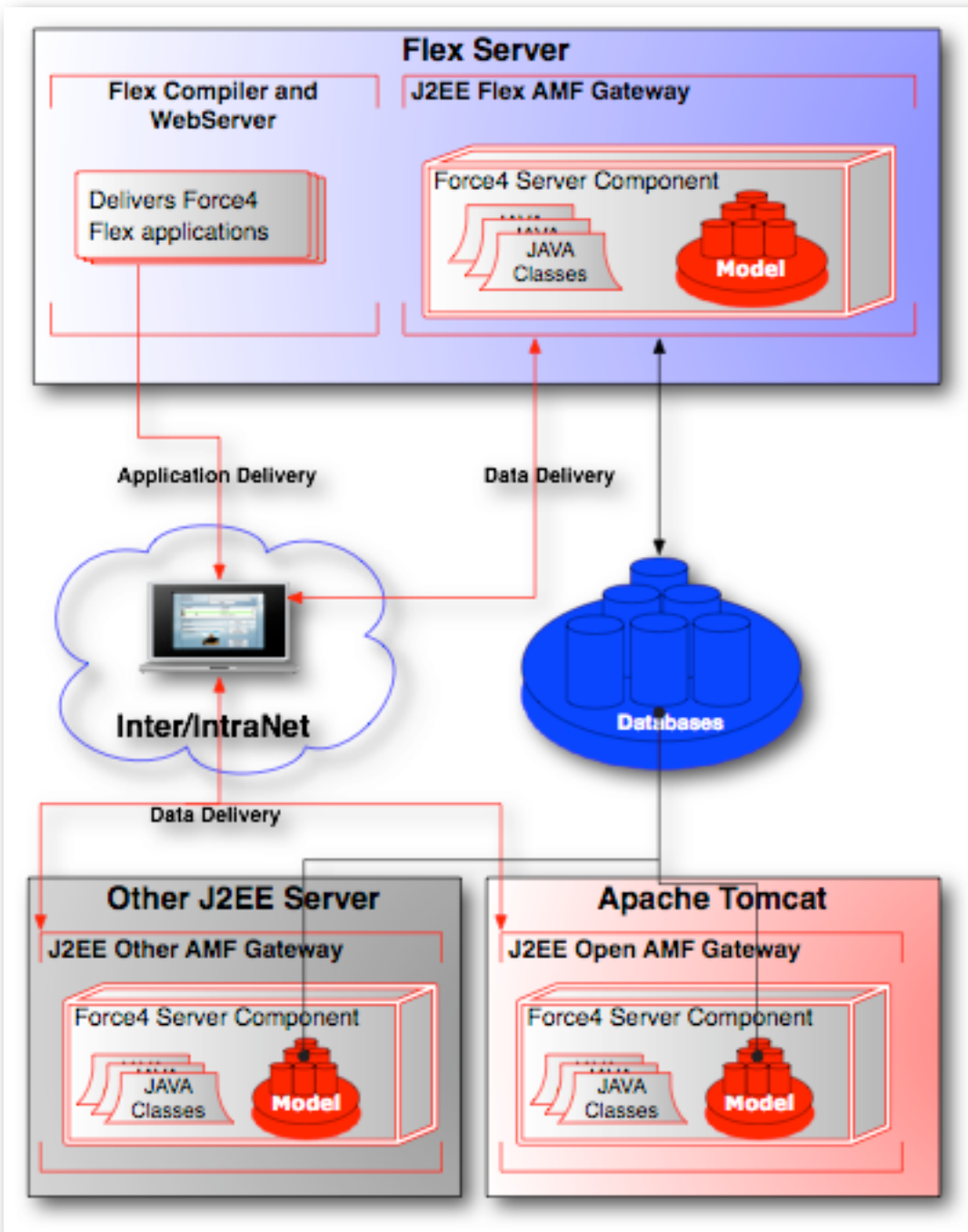


The Force4 composer-applications use the Force4 client framework and the object and access classes created by the Modeler to communicate with the database. The client framework offers basic DB access methods and an event management, which use the functional Force4 client components (search, list,...). The client framework is available in source code and is described in the developer documentation, so it can also be used in applications not created by the Force4 Composer.

Force4 RIAs consist of two kinds of components. First of course, the standard ones provided by the language of the presentation server. In order to introduce automatic functionality, so-called functional components are used. These components are derived from certain standard components. Their functionality includes search, lists and detail views. A search is specified by input fields such as: texts, date and popup lists. Subcomponents from the Force4 Client framework are used for this. The search action initiates a request through the Force4 server to the database. The results of these requests are listed and presented in the list component. Sole entries from the list can be shown in a detail component and also be edited and deleted therein. The functional components communicate with each other through an event mechanism, which relays changes to

components to other related components. This mechanism allows not for only simple search-list-detail RIAs to be created, but also complex master-detail dependencies to be visualised.

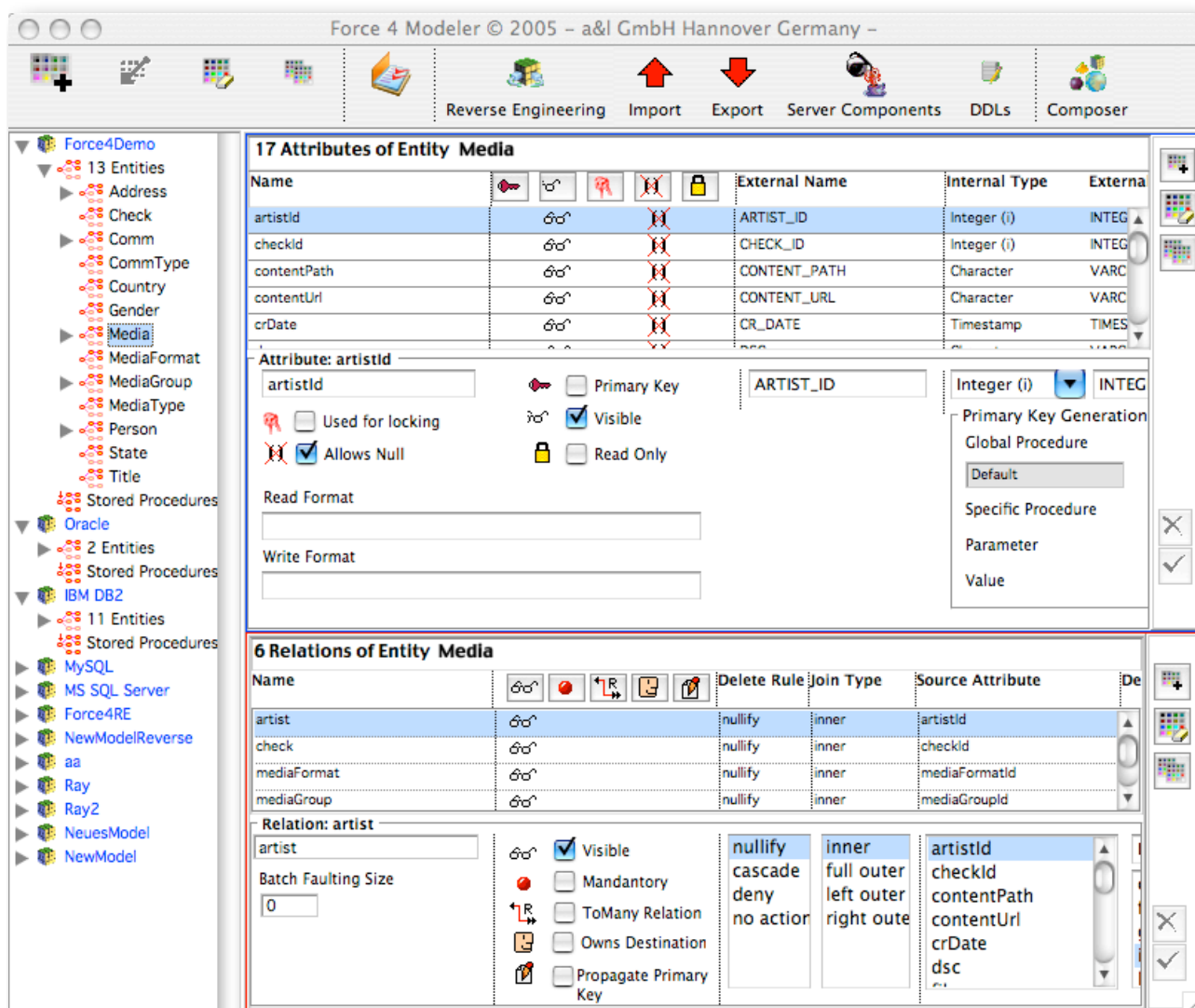
Data Flow



As already mentioned above, the data flow shows the division between the Force4 composer and the Force4 server component. The Force4 applications constructed by the Composer or handmade are delivered by a presentation server (Flex, Open Laszlo). Under certain circumstances the server will deliver parts of applications, libraries, pictures etc. afterwards. The client application uses, in case of Flex, so-called Remote Objects to communicate via an AMF gateway with the force4 server component. This means that all Force4 database operations are managed via this connection. The advantage of this concept is the possible division of the load between presentation server and Force4 server. Because the Force4 server component for the J2EE server is not tied to a Macromedia Flex it is possible to implement a standard Apache TOMCAT with OpenAMF as gateway.

Modeler

The Force4 Modeler is an E/R-modeling-tool to easily deposit data structures in a repository and to keep these up-to-date quickly and easily during the modeling phase of a process. Furthermore, the Force4-Modeler works as a development connection between the database and JAVA, Macromedia Flex, OMNIS, with their specific terminologies.



The model in Force4-Modeler is first an abstract "construct", consisting of entities, attributes and relations, in order to build and present data structures resulting, for example, from a business-process. The construction can be done manually in the Modeler. Reading the structures from existing databases through a direct connection using JDBC, a so-called reverse engineering is also possible.

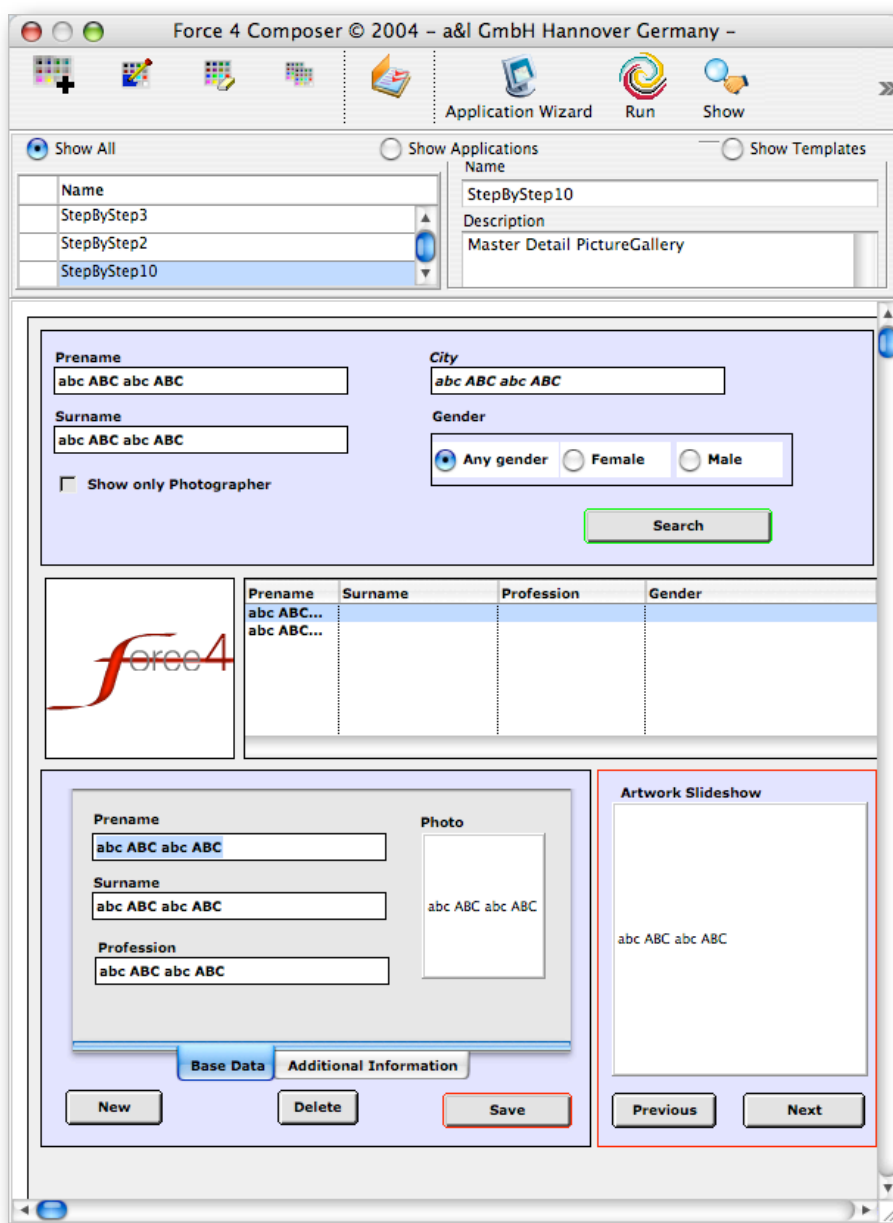
After a model has been created based on functional demands, it is also possible to create a database as well as direct connections to JAVA, Macromedia Flex and OMNIS.

Entities and attributes are abstract counterparts to tables and table columns in a database. In addition to the properties which are needed to define a database table, further properties of entities and attributes can be set to create compatibility to JAVA- and Macromedia Flex.

Relations in Force4-Modeler are connections between entities. The construction of these relations define the entities to form a relational structure. Relations on the database are represented by Foreign Key Constraints, conditions on database-level, which show the connection of the tables as long as the database supports these constraints.

Composer

The Force4 composer is the tool to create fast and easy Rich Internet applications from existing E/R Models, which have been made with the help of the Force4 modeler. Here WYSIWYG GUI compo-



nents and functions are put together using Drag & Drop. Applications can be created directly from templates, which have also been created by the composer. The individual parts of the application can be attributed in style, size, etc. Additionally, these so-called containers, may be reused in other applications. The composer saves the application in a local database and is independent of the created source text. These applications are generated automatically and can be previewed in real-time, if connected to a presentation server.

Offline mode is available if the presentation server cannot be used directly. The composer is able to create RIAs without any further programming effort. The application's source text may be edited by the developer at a later date. This may be necessary in order to integrate specific business logic

in the RIA. The Force4 Composer offers the basic tools for such integration. Thanks to the integrative character of Force4, Force4 is an ideal tool for rapid prototyping of database-supported rich Internet applications.

The Server Side Flex and OpenLaszlo

Force4 creates applications for Macromedia Flex and Open Laszlo RIA as source text, which are compiled by the presentation server and delivered to the client. The Force4 framework is used for this. The interface for RemoteObjects/JavaRPCs is defined in the framework. The framework is installed on the Flex/Open Laszlo server, at the same place where the composer saves its generated source text. This location will be used while compiling the Flash applications later on.

The framework is a part of the Force4 Server component, but requires no Force4 license and therefore it can be used easily on the presentation server which itself is not responsible for the Force4 database communication. The communication parts of the Force4 component can therefore be installed on all the common J2EE compatible servers. The current version expects a JRE 1.4.2 or compatible JAVA environment. A JDK is not necessary. Furthermore, the Force4 Server contains all the necessary packages to be able to run with standard JDBC (java.sql) in JRE.

Force4 does not use JDBC/JNDI name-binding and therefore it is totally independent from the J2EE server configured JAVA data sources. The necessary JDBC drivers needed for the database communication are also included in Force4. Extra drivers can be integrated as well.

Force4 Feature Summary

Modeler

- Able to access all SQL92-type databases
- e.g. IBM - DB2, ORACLE, FRONTBASE, MySQL, Microsoft SQL Server ...
- Integrated Database Modelling (Entity / Relationship)
- Reverse Engineering. Generates database models from existing databases
- Generates all required Access and ORM Objects (Object/Relational Mapping)
- Support for OpenAMF

Composer

- Composing of Rich Internet Applications using Drag & Drop
- Supplies functional components for all database transactions.
- Supports canvas-in-canvas and container-in-container techniques
- Composition of templates for ease of reproduction
- Ability to reuse all components and applications
- Support for relevant intelligent WebClient systems e.g. Flash and OMNIS

Force4 Advantages

Technology

- Connection to all (even existing) databases "on the fly".
- The use of database models rather than table-based SQL simplifies working processes.
- Web applications need not know the database rather only the abstracted view.
- Workloads can be divided between database and interface design.
- Real "frameworks" reduce maintenance and simplify training of new co-designers.
- Reuse of components and applications in all projects.
- ORM Layer ensures database integrity, Object Caching, optimum performance and memory usage.
- Allows changes to database structure (within the limits of SQL92) without having to change the application.
- Force4 models allow the target database to be exchanged for an alternative RDBMS system.
- Force4 can be used on various intelligent WebClient systems (Flash, Omnis, OpenLaszlo...). This widens the usability of applications.

Economy and Market

- Better 2nd generation internet applications offer user benefits and acceptance.
- Up to 90% savings in development times of intelligent web applications.
- Lower personnel and training costs as the software is less training intensive and special database knowledge is not necessary.
- Fast application prototyping ensures market advantages.
- Shortened "time-to-market" times.
- Support for various intelligent WebClient systems widens the market for applications.
- Force4 applications and models work with all SQL92 databases without readjustments