VERSION: 1.1

DATE: 31.07.2007

AUTHORS:

Peter Conrad

Florian Weigmann

SAP ARCHITECTURE BLUEPRINT

SAP CRM 2007

PROJECT NAME/CPROJECT TITLE: **SAP CRM 2007**

SPONSOR/PROJECT INITIATOR: **BOB STUTZ**

PROGRAM/PROJECT LEAD: **HALA ZEINE**

LEAD ARCHITECT: **PETER CONRAD**

DEVELOPMENT: ☒ SAPLabs: WDF, Banaglore, Palo Alto, Montreal

☐ Partner/ISV ____________

Document History

Version	Date	Status (Comments)
0.9	2007-07-10	Draft
1.0	2007-07-16	Version for Blueprint Session
1.1	2007-07-31	SAP Internal Portal Version

I MARKET AND PRODUCT BACKGROUND OF PROJECT/PROGRAM

Planned release date:	End of 2007	
Underlying SAP NetWeaver release:	SAP Netweaver 04s (7.00) SAP Netweaver 7.10 (only limited usage of Data Orchestration Engine (DOE) in “van stock” scenario)	
Used SAP NetWeaver stacks:	☒ ABAP	☒ J2EE/Java EE 5
Use cases targeted by the project/program:		
• Successor release for all previous mySAP CRM releases • Make integration with SAP NetWeaver Portal optional • Support of Web Services/ Enterprise Services (Level 1) • Trade Promotion Management (TPM) • Claims and Funds • Market Development Funds		
Strategic goals SAP wants to achieve with the project/program:		
• Provide a successor release for SAP CRM 5.0 with full support of all functionality in the new CRM Web Client UI • Finalize UI unification of complete CRM solution to provide a comprehensive solution with a homogenous UI and reduce TCO drivers at customer side related to different UI technologies. • Address additional market demand in the TPM and Marketing area.		
Mandatory software capabilities to address goals, use cases, and target market:		
• Full support of unified CRM Web Client UI • Flexibility and Extensibility of the UI • Optional support of Enterprise Portal integration for business users. Exceptions: – Interaction Center – Mobile Solutions – CRM Web Channel (aka “e-Commerce”) • Restrict mandatory usage of J2EE stack as far as possible. • Support future waves of CRM on-demand so that core product can be leveraged as good as possible and the “on demand”-solution can configure as much as possible and reduce the on-demand specific code where possible.		

II. ARCHITECTURE

SAP CRM 2007as the next strategic CRM release, is supposed to be the successor release of all former mySAP CRM releases such as SAP CRM 4.0, 5.0 as well as 5.1 and 5.2. It embraces the components depicted in Figure 1. Thereby the Real Time Decision Engine is the acquired Ingeneo Server, and the SAP Business Communication Manager is the acquired Wicom Server. Both solutions address complementary market demand and were aquired to round off the CRM solution in these specific areas.

From a product architecture perspective SAP CRM 2007 is the final release to complete the original strategy, which is to provide the new, unified CRM Web Client UI over the period of three releases (5.1, 5.2 and 2007). It will contain all functionality and scenarios that were supported by former releases including all industry scenarios plus some new scenarios e.g in the marketing area (trade promotion management, etc.).

Inline with this strategy, SAPGui will only be supported for administrative tasks, like customizing in the IMG, and runtime administration of the ABAP server (e.g. Transport Management System). For the business users, SAPGui and CRM People Centric UI will not be supported anymore with SAP CRM 2007. Due to the conceptual discrepancy of the different UI technologies, an automatic migration of existing user interfaces can not be provided.

The CRM Web Client UI can be accessed using only a web browser as well as integrated into the SAP Enterprise Portal (as an option).

All business objects of CRM support the CRM Business Object Layer by implementing the respective Generic Interaction Layer Component Interfaces in order to allow unified access and thus are able to support the CRM Web Client UI as well as other consumers such as the CRM Web Service Tool.

Leveraging the CRM Business Object Layer including the models of the business objects, CRM offers a Web Service Creation Tool that allows customers to create basic web services (stateless create-, read-, update- and query services) on the fly without the need of touching any development tools.

The Service Integration Workbench allows to create level 1 enterprise services from “legacy” interfaces in a semi-automated way by supporting GDT mapping, ESR export and code generation. CRM has invested into integrating the CRM Web Services into the Service Implementation Workbench.
In a further step (after CRM 2007 SP00) CRM will be able to deliver level 1 enterprise services based on the CRM Web Services built with the Service Implementation Workbench (SIW) via enterprise service bundles.

SAP CRM 2007 Overall System Architecture

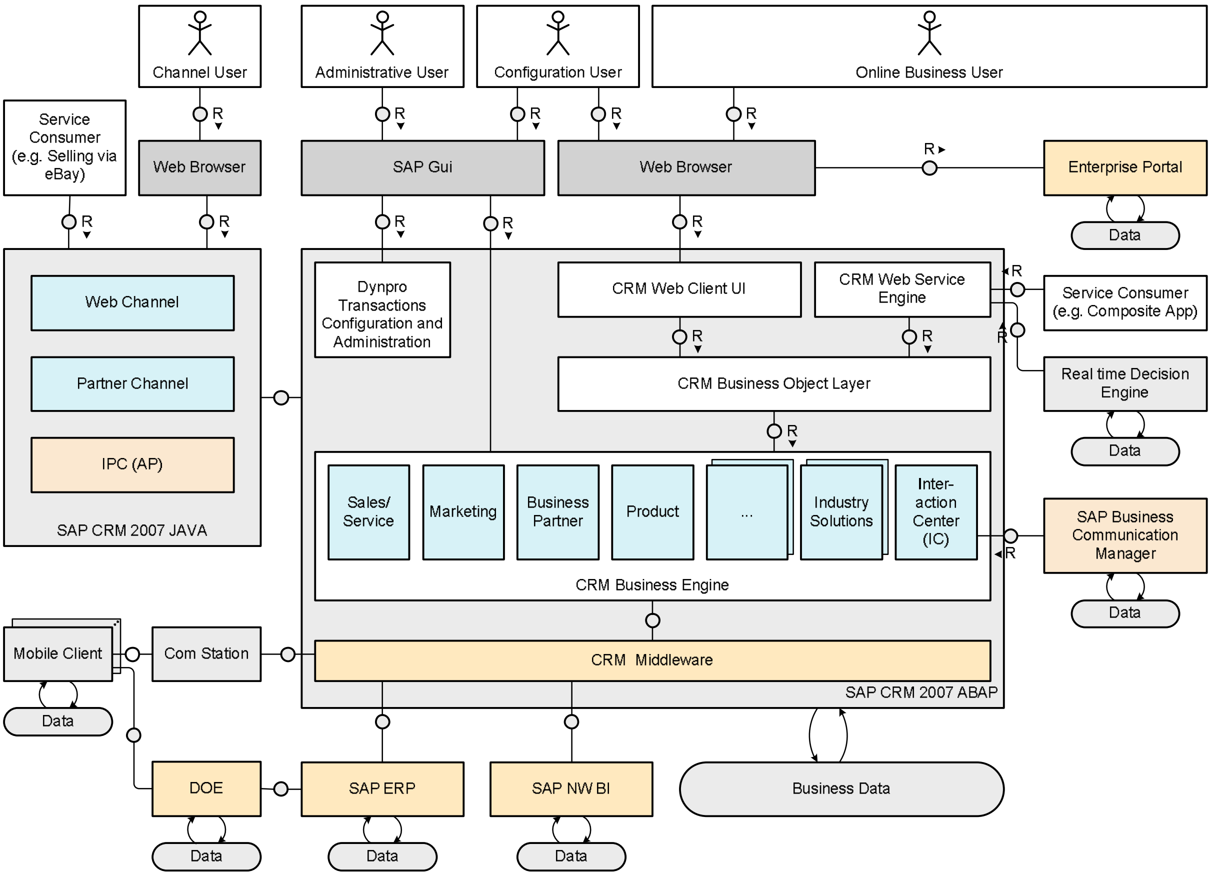

Figure 1 SAP CRM 2007 Overall System Architecture

Main Architecture Concepts and Decisions

1. CRM Web Client UI

The CRM Web Client UI is the only user interface for online business users. Only configuration (IMG) and administrative tasks are accessed via SAPGui transactions. The Web Client UI is used across different channels (business users, interaction center, partner channel management) as well as across different deployment channels (On demand and On premise).

Operation Modes

The CRM Web Client UI supports two distinct operation modes:

- Standalone Mode
This mode is used if the customer does not want to integrate the CRM system into SAP NetWeaver Portal. This mode is based on a CRM-specific UI context frame (“L-Shape”) that enables the CRM applications to run natively in the browser. The L-shape takes care of branding and navigation and provides access to general services such as online help, personalization and central search.
For the standalone mode the navigation model is defined via CRM Business Roles which are based on navigation profiles that can be shared across different roles. Each CRM business role maps to one PFCG role for the purpose of authorization management.

- Portal Integration Mode
In this mode the UI context frame runs without visible navigation bar and header area as these areas are provided by SAP NetWeaver Portal. CRM Business Roles can be uploaded

into the portal in order to enable customers to take these roles as starting point and enrich the portal roles with additional content. In this case each CRM application can be addressed separately from the portal.

Overview of Layers

From a high level perspective, the CRM Web Client UI works in a layered architecture where each layer is supposed to only interact with the underlying layer only (see Figure 2). Skipping layers is technically possible but shall be avoided wherever possible.

Via his browser the user interacts with the CRM system. The Web Client UI layer handles all requests from the browser, distributes the entered values to the respective views and models, calls the application event handlers and renders the HTML response for the browser.

Business data and functionality is accessed via the CRM Business Object Layer. This layer unifies the interface of the diverse business object APIs implemented in the Business Engine layer. It also buffers the business objects close to the UI for the fine granular access from the UI layer for performance optimization.

Business objects such as accounts, opportunities, marketing campaigns and service tickets including their related functionality are implemented in the Business Engine layer. The implementation of these objects is mostly based on implementation frameworks or architectures, such as the business transaction framework or the business partner API architecture. Implementations of this layer include also database access and updates which are usually encapsulated in separate data access functions or methods.

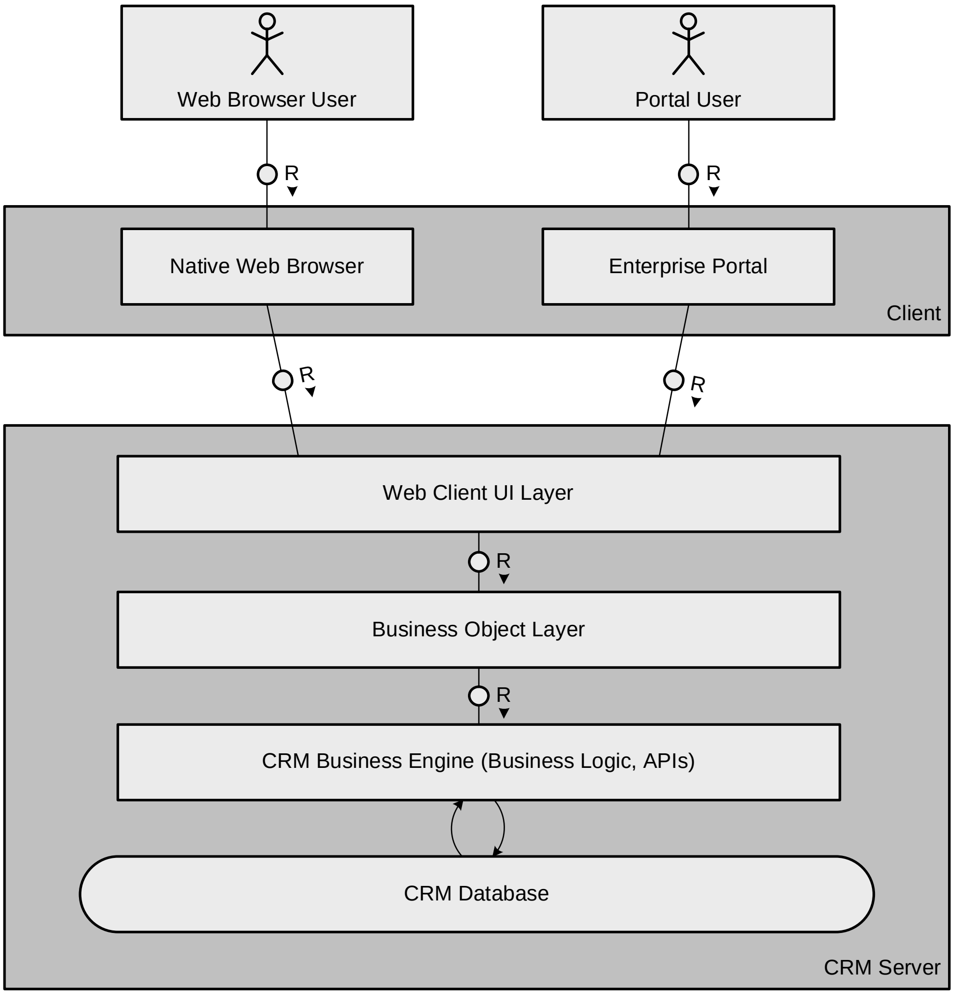

Figure 2 CRM Web Client UI – High-Level Architecture

Extensibility

In a CRM environment the flexibility and extensibility of the UI is the key non-functional requirement that needs to be fulfilled by the UI architecture. Specifically in the area of CRM, the market pressure is immense to allow these enhancements in a modification free way without restricting the flexibility for customers.

In SAP CRM 2007, the UI is highly configurable by the customer based on different determining dimensions without coding effort. Extension fields added via the Easy Enhancement Workbench are available in the business object layer automatically and can be directly used from the UI-configuration toolset to include them in the respective UI.

2. UI Framework

The CRM Web Client framework is based on the Model-View-Controller implementation of the Business Server Page Framework of SAP NetWeaver. On top it introduces basic Web Dynpro concepts such as components, windows, views - including contexts and view controllers - and custom controllers as well as the composition and navigation concepts. In contrast to Web Dynpro the CRM UI Web Client framework follows an “open” client strategy that allows applications to introduce their own application specific logic and UI elements on the client, such as applets, flex components, JavaScript or own html.

Request Handling

The Web Client UI layer handles all requests from the browser, distributes the entered values to the respective views and models, calls the application event handlers and renders the HTML response for the browser based on the page composed from multiple, different views. The UI layer also supports service requests (used for e.g. AJAX calls, data exchange with active components, MS Excel Export, etc.) bypassing the normal request cycle.

The request handling supports the custom use of AJAX with full controlled delta-screen updates as well as an automatic delta handling for the other cases to reduce the network load and optimize the client rendering time. The UI updates are processed flicker free, also in case delta handling is not used for a specific request. AJAX support can also be used to stabilize screen areas for active components so that the roundtrip of the browser part of the screen does not cause reloading the active component.

Rendering and UI controls

To ensure a common look and feel across all CRM applications and support the interaction paradigm defined by the CRM UI concept, the CRM Web Client UI layer offers an own BSP extension (THTMLB) with a set of appropriate UI controls to be used by the applications.

Visual appearance is controlled by a set of cascading style sheets that are shared between THTMLB and the UI context frame. A set of such style sheets form a skin. SAP delivers around 5 different skins, but customer can define their own skins according to their corporate identity and also restrict the usage to the ones that they want to offer their end users for personalization.

Skins are technically not identical to the portal themes. For the usage within a portal a skin mapping can be defined.

UI configuration

In addition the UI framework offers so called “configuration tags” for forms, tables and trees in a separate BSP extension (CHTMLB) that internally use the basic tags from the THTMLB library to compose a complete form, table or tree based on configuration files.

The UI configuration is usually done by the project implementation team or a customer power user in the on-premise case while in the on-demand environment this is done by the customer power user from within the running Web Client UI on the productive instance. The UI configuration allows defining the set of fields, their layout, their labels and some basic properties, such as "read-only" or "mandatory".

The UI configuration is interpreted at runtime from the configuration tags. Multiple configurations can be stored for the same view based on role, object type, sub object type and usage (for reused views) and independently for SAP delivery and the customer.

UI configuration does not cause any type of DDIC change or code modification to avoid recompilation and short dumps in the productive environment of on-demand. Thus, it also supports non-modification policies of customers.

3. Business Object Layer and Generic Interaction Layer

The Business Object Layer is a model based layer with the purpose to expose all CRM business objects to the UI layer as well as to other consumers (such as the CRM Web Service Tool) in a unified way with a unified interface. It internally consists of three layers:

- Business Object Layer
  A stateful layer which is accessed from the UI layer – the UI framework itself for data binding as well as from application code in the UI layer. It manages an in-memory cache based on entity instances in order to allow fine granular access on a single attribute level from the above layers without the need to send requests down to the business layer all the time.

- Generic Interaction Layer
  Whenever the Business Object Layer does not have enough information in its cache to response to a request from the above layers it requests the Generic Interaction Layer for delivering this information (see Figure 3). The Generic Interaction Layer manages the individual Interaction Layer Components and acts as a dispatcher of requests from above (Business object layer or other potential consumers). It operates stateless and allows the Business Object Layer to address several requests to different object instances at the same time. The Generic Interaction Layer looks up the model metadata to locate the component responsible to fulfill a specific request or a part of it and delegates the request (resp. the relevant part) to it. It aggregates the results of the multiple calls to components and delivers the result back to the requesting instance – usually the Business Object Layer.

- Generic Interaction Layer Components
  The Generic Interaction Layer Components are provided by the CRM application development; they implement the component interface to map the specific interfaces of their application APIs to a unified interface where the Generic Interaction Layer and Business Object Layer are built upon.

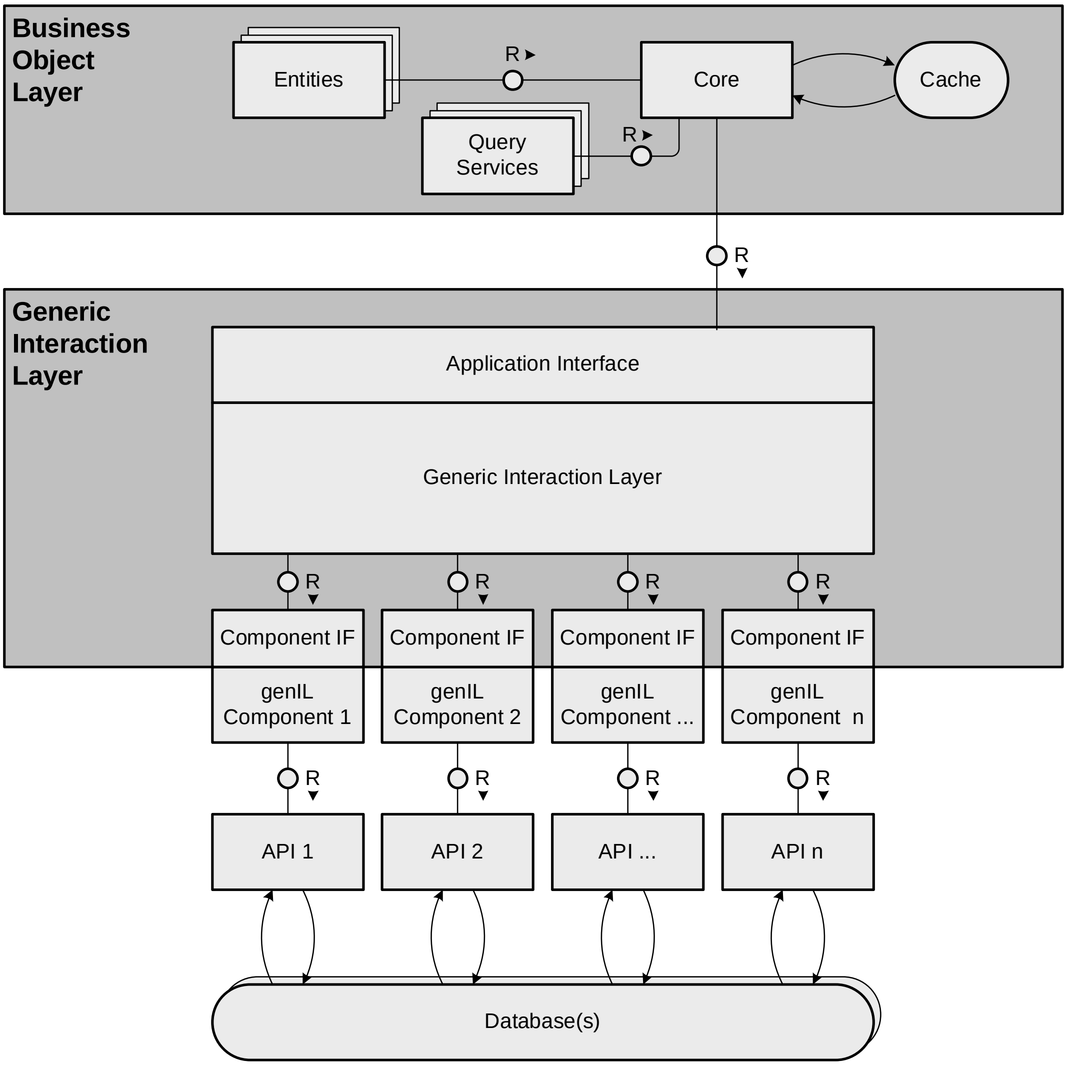

Figure 3 Business Object Layer and Generic Interaction Layer

4. The CRM Web Service Tool

The CRM Web Service Tool is a web tool that allows technical oriented users to create basic web services for CRM business objects without any additional coding effort. Web services created with the tool are stateless services and include the generic operation create, read, update and query. The services are created locally in the CRM ABAP server, are not aligned nor there is a data type mapping to global data types. Consequently, these services are also not available in the Enterprise Services Repository.

Nevertheless, the ability of the customer to quickly create tailored web services for specific needs without any coding effort - including customer enhancement fields - is seen as an big value-add for customers and partners.

The tool leverages the object and relationship information of the model of the Business Object Layer. It generates function modules including the interface and implementation and leverages the infrastructure of the SAP NetWeaver Application Server (ABAP) to wrap these into local web services.

5. CRM Enterprise Services

In future, CRM will leverage the Web Service Tool also to create enterprise services with significantly lower effort than implementing them manually in the classical way. The CRM Web Services are integrated into the Service Implementation Workbench (SIW+) so that Level 1 Enterprise Services can be created based on CRM Web Services. The SIW+ is a SAP tool which does not get shipped to customers but will we used by SAP development to create enterprise services (see Figure 4).

To create an enterprise service, the developer uses the SIW+ to import the respective interface definition of a CRM Web Service and to map the CRM local ABAP data types to ESR Core Data types (CDTs) and/ or Global Data Types (GDTs, see **Error! Reference source not found.**). This mapping effort is to be done manually but the SIW+ facilitates this by providing lists of appropriate GDTs. After the mapping is finished, the SIW+ can upload the service definition into ESR, trigger the generation of the proxy in the local CRM system and also generate the proxy implementation based on the CRM Web Service implementation. Prerequisite for creating a level 1 enterprise service still is an approved PIC0 service operation (Service operation, message type names and assigned business object).

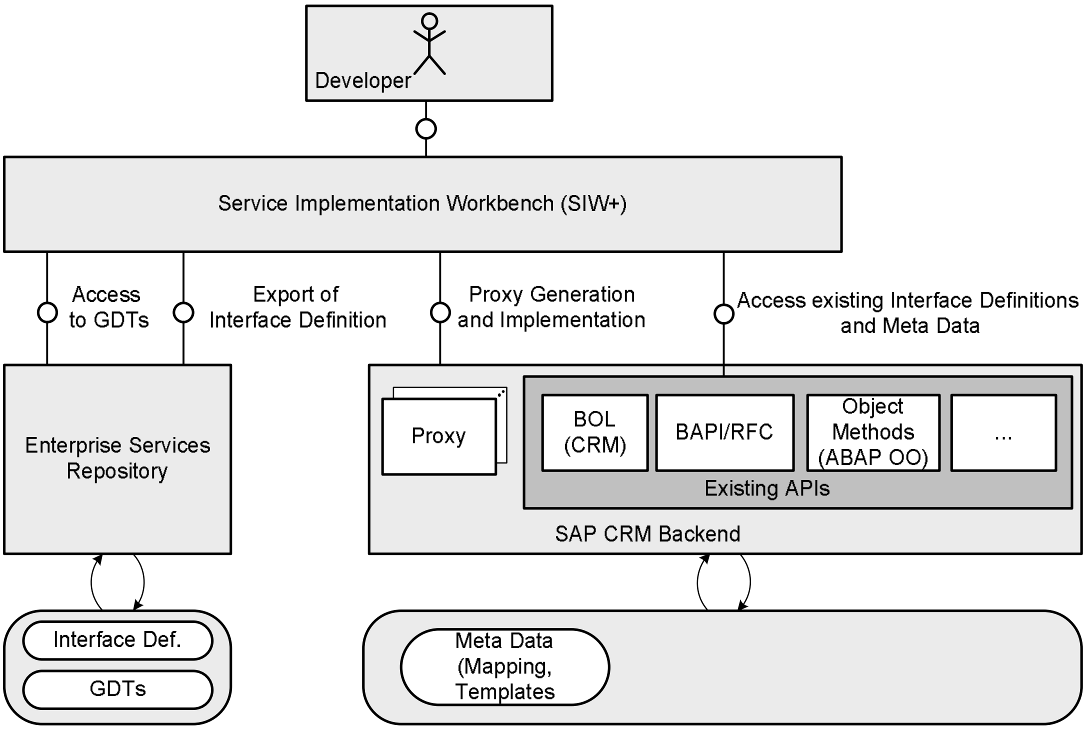
Developer
Service Implementation Workbench (SIW+)
Access to GDTs
Export of Interface Definition
Proxy Generation and Implementation
Access existing Interface Definitions and Meta Data
Enterprise Services Repository
Proxy
BOL (CRM)
BAPI/RFC
Object Methods (ABAP OO)
...
Existing APIs
SAP CRM Backend
Interface Def.
GDTs
Meta Data (Mapping, Templates

Figure 4 Implementation of an Enterprise Service with SIW+

Total Cost of Ownership

Even though the overall product architecture looks pretty complex (see Figure 1), CRM focuses on a lean setup for straight forward business users.

The following components are optional:

- SAP NetWeaver Portal
- SAP CRM 2007 Java
- SAP NetWeaver BI
- Mobile Clients incl. COM Station
- DOE (only used for Van Stock scenario)
- SAP ERP
- Realtime Decision Engine
- SAP Business Communication Manager

If the NetWeaver Data Orchestration Engine (DOE) would be available on NetWeaver 7.0 instead of only on 7.10, this would decrease the TCO for those customers using the “van stock” scenario.

Deployment

In general, the provided architecture supports on-premise as well as on-demand deployment options. CRM positions a “Hybrid model” in the market that allows an on-demand customer to migrate to an on-premise installation based on their evolving business environment.

SAP CRM uses the same UI technology and configuration toolset for both deployment options.

The lean setup allows SAP CRM 2007 to be used also in the on-demand environment with only the SAP CRM 2007 ABAP server. For on demand usage, SAP NetWeaver BI is optional and the SAP NetWeaver Portal is not supported.

Architecture Documentation

The ERD for SAP CRM 2007 can be accessed from the CRM program page in the SAP Corporate Portal https://portal.wdf.sap.corp/go/crmprogram