



VERSION: 1.0

DATE: 2008-07-09

AUTHORS:

Michel Martin

SAP ARCHITECTURE BLUEPRINT

Object Layer Framework (from CRM Marketing)

PROJECT NAME/CPROJECT TITLE: **MARKETING OBJECT LAYER**

SPONSOR/PROJECT INITIATOR: **ERIK TIDEN**

PROGRAM/PROJECT LEAD: **OLIVER STANKE**

LEAD ARCHITECT: **RALPH KIEFER**

DEVELOPMENT: ☒ SAP AG and SAP Labs Canada

Document History		
Version	Date	Status (Comments)
1.0	2008-05-23	Initial Draft
1.0	2008-07-09	Final Version

I MARKET AND PRODUCT BACKGROUND OF PROJECT/PROGRAM

Planned release date:	With CRM 5.2 (approximately beginning of 2007)	
Underlying SAP NetWeaver release:	SAP Netweaver 04s (7.00)	
Used SAP NetWeaver stacks:	<input checked="" type="checkbox"/> ABAP	<input type="checkbox"/> J2EE/Java EE 5
Use cases targeted by the project/program:		
<ol style="list-style-type: none"> Having a framework which: <ul style="list-style-type: none"> Is easy to maintain, enhance and learn (e.g. new developers and IMS) Has better performance improvement possibilities (lazy load and multi load) Provides an integration of generic services But still uses the capabilities of the CGPL framework (but internally only!) Increased synergy and generic coding between both applications Marketing Planner and Account Planning which are built on the Generic Project Framework CGPL. Flexibility and extensibility of the CRM marketing business applications. Business logic decoupling between the different applications and objects. 		
Strategic goals SAP wants to achieve with the project/program:		
<ul style="list-style-type: none"> Lower the maintenance costs of the CRM marketing applications by: <ul style="list-style-type: none"> A lean IBOM implementation with generic accesses to the Object Layer Fully object oriented business model A framework that is easy to maintain, enhance and learn Reduce the TCO in the CRM Marketing area by using generic coding. 		
Mandatory software capabilities to address goals, use cases, and target market:		
<ul style="list-style-type: none"> Complete encapsulation of CGPL (a generic project framework previously known as Campaign Planning) objects to hide the CGPL complexity Collection / item pattern for standardized access Standardized assignment access Complete encapsulation of generic functionality Lazy load enablement for performance Generic IBOM classes to access Object Layer Generic super classes for the Object Layer development Pre-defined places for the business logic Pre-defined Object Layer interfaces (e.g. collection/item pattern; no CGPL parameters) 		

II. ARCHITECTURE

Prior to CRM 5.2, the marketing applications such as Marketing Planner and Account Planning were based on the CGPL framework. The CGPL framework was built in CRM 2.0 for Campaign Management and was later moved to the ABA. One of its well-known applications is the cProjects application.

Briefly, CGPL allows to quickly setup a project-driven application and it contains a lot of integrated services such as Hierarchy Handling, Status Management, Notes, Change Docs, Archiving, Authority Checks, and Integration to BI/ERP. However, CGPL brings an important list of disadvantages such as:

- CGPL is maintained in the ABA which has different release cycles;
- CGPL corrections are provided with delay;
- CGPL changes are almost impossible (i.e. single save is not supported);
- CGPL does not provide real and accessible business objects;
- CGPL has an inefficient enhancement concept, which is limited to the assignment concept;
- The business logic of these marketing objects is dispersed over too many places;
- The leading objects are implemented through attribute assignments;
- No decoupling of the business logic for the different business objects;
- No control before and during CGPL framework calls (high dependency).

This indeed brought important problems such as:

- Changes regarding object accesses and business logic have to be done on many places;
- Very error-prone due to code duplication ;
- Some processes within the CGPL framework cannot be influenced even if needed;
- Unstable workarounds needed;
- High effort for knowledge transfer for new marketing planner developers;
- Very High maintenance effort (during development and for IMS);
- Each new development makes the application more complex and harder to maintain.

Since a lot of new developments will happen in the marketing area throughout CRM 5.2, CRM 2007 and CRM 7.0, it is necessary to come up with the Object Layer framework (a.k.a. CRM Marketing Object Layer).

This Object Layer framework will serve as the new base of the CRM Marketing applications, will wrap all the CGPL complexity, and will ensure generic and unified access to the UI, publishing all the business objects through the IBOM (Internal Business Object Model).

The applications built on the Object Layer will be modeled with an Object Layer metadata, specifying the relationships, links, and features of every business object.

Main Architecture Concepts and Decisions (Runtime)

The Object Layer will serve as a framework business engine for CRM applications, such as in the Marketing area. The Object Layer will receive requests from the above layers via the IBOM (Internal Business Object Models).

Because the Object Layer will encapsulate and hide the CGPL complexity and limitations, the Object Layer will enable application developers to build easily extensible business applications.

The Object Layer will only load the required set of objects requested by its users, such as the IBOM, hence saving on memory consumption and performance.

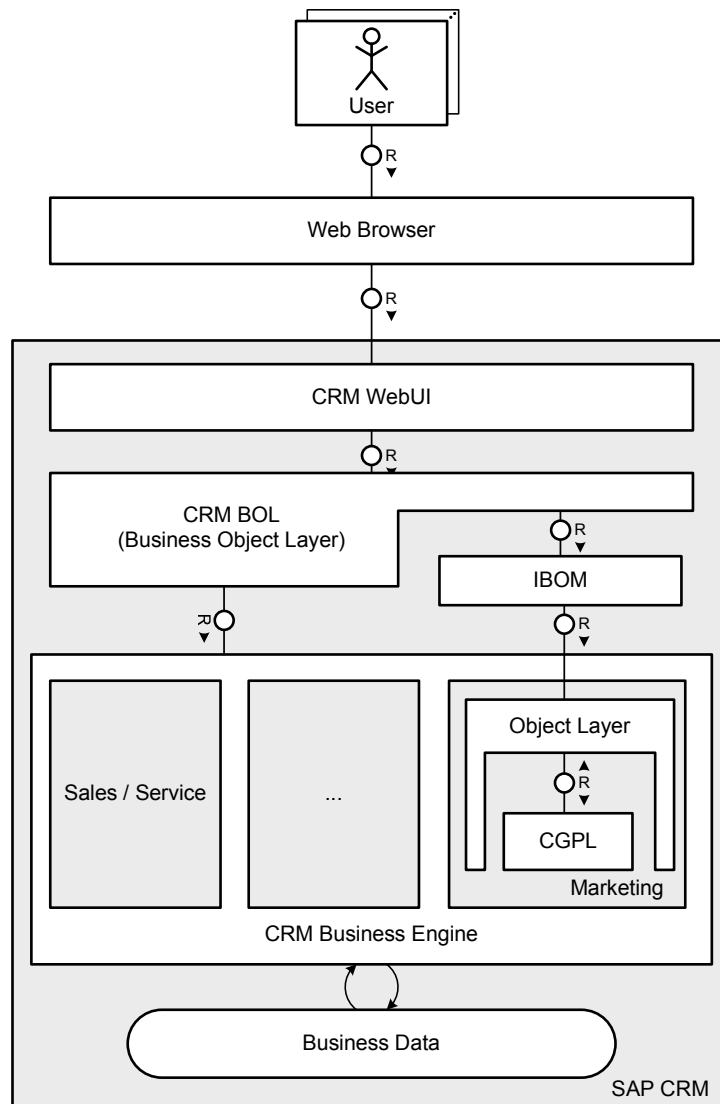


Figure 1: SAP CRM High Level Architecture with Marketing Object Layer

The Object Layer encapsulates two main parts:

1. The business applications implementing the Object Layer
2. The Object Layer core and its generic code

The links with CGPL will remain minimal, only when really required. A CGPL wrapper will serve as a proxy to access CGPL information. Several managers will serve unique purposes such as on authorizations, BAdIs, and events. A message handler will be used to keep messages to be shown to the users. Finally, a metadata provider will provide the metadata models to the Object Layers.

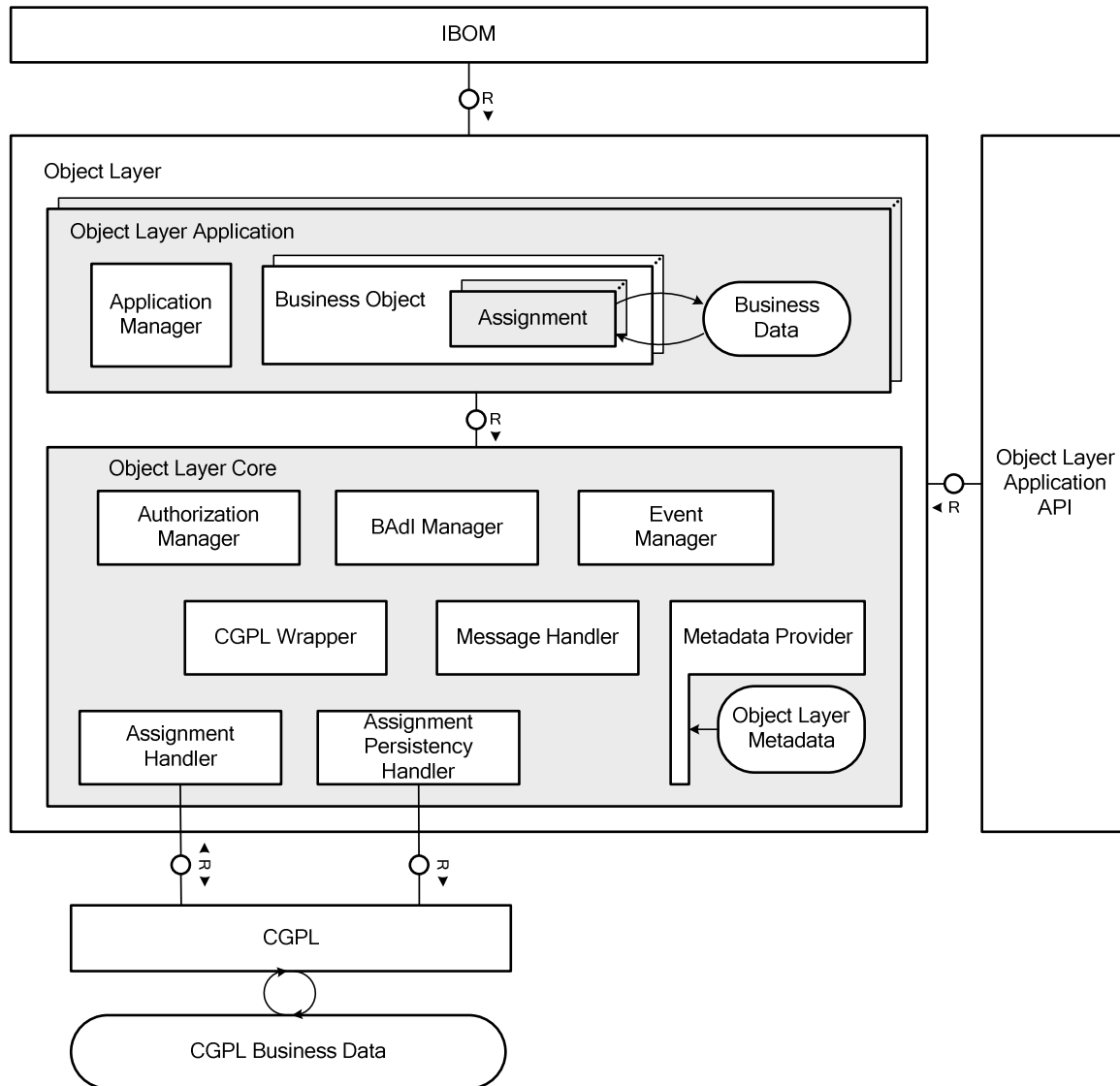


Figure 2: Marketing Object Layer High Level Architecture

Main Architecture Concepts and Decisions (Design time)

The metadata models will have to be specified and entered using the customizing transaction CRM_MKTGS_OL_MODEL by every business application developer reusing the Object Layer framework.

Such metadata will contain the following structure as shown in Figure 3. Additionally, implementing class names, cardinalities, relationships, check mode (e.g. check upon change, always check, check initially and upon change), and options on SAP basic features (e.g. archiving, change document) will be provided.

Having all these specifications on the application level, the business object level, and the assignment level, will lower the possibilities of bug, decrease code duplication, and minimize the risk breaking the model through the code, as it is specified in a meta-model, fully dynamically interpreted at runtime.

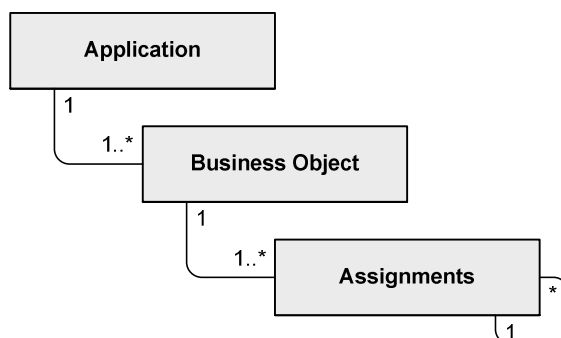
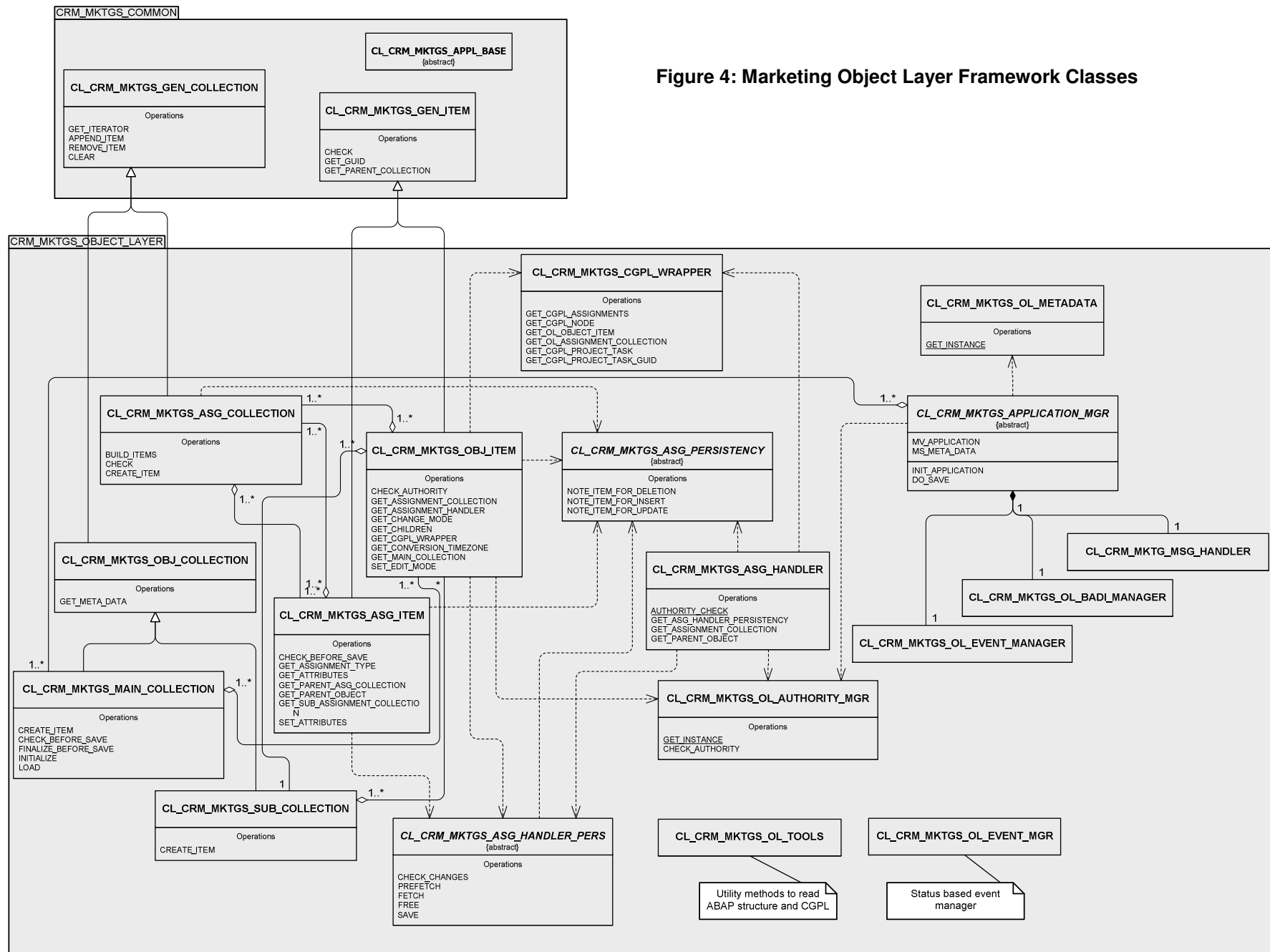


Figure 3: Marketing Object Layer High Level Structure

In Figure 4, main relationships and dependencies amongst the Object Layer framework classes are shown. This depicts the interactions between the collections and items, and their generic handlers.

All these generic classes will be reused in every business application implementing the Object Layer framework. This ensures a stable code base for every application, as only a small subset of developers and architects will perform future changes in these classes. This substantially helps to reduce the code duplication across similar and non-similar applications, and consequently, increases developers efficiency and effectiveness to build brand new applications on the top of the Object Layer.

There are currently three business applications implementing reusing the Object Layer and one of them is a brand new application as of CRM 2007. This last application has been developed quickly, with fewer efforts than usual.



Total Cost of Ownership

The Object Layer framework reduces the TCO of CRM Marketing applications, thanks to its improved flexibility, easier maintenance, faster knowledge transfers, its fully object oriented approach and its overall simplicity.

Deployment

The Object Layer has been shipped with CRM 5.2 and CRM 2007, and will be shipped with CRM 7.0 and future releases. Consequently, every customer using one of the marketing applications based on the Object Layer will run the Object Layer framework.

Maintenance

The maintenance of the Object Layer applications will be taken over by IMS upon the proper handovers. The maintenance of the Object Layer framework will remain under the responsibility of CRM Marketing architects.

Architecture Documentation

Additional information regarding the Object Layer can be found on the CRM Marketing Wiki pages:

<https://bis.wdf.sap.corp/twiki/bin/view/Sapinternal/CRMMarketingMKTPLObjectLayer> and
<https://bis.wdf.sap.corp/twiki/bin/view/Sapinternal/CRMMarketingMKTGSOBJECTLayer>.

Moreover, UML diagrams of the Object Layer are available on
<\\caymq001\CRM\CampaignManagement\Architecture\UML>.

Additional Important Aspects

Not relevant.