LIVING THE DIGITAL WORLD

INNOVAGENCY

EMSA – IMS

Technical Specification

Project	EMSA - IMS	Author	Innovagency
	EMSA	Date	16-07-2014
Version	1.0		

Changes

Date	Version	Description	Author
16-07-2014	1.0	Initial Version	Alexandre Marreiros Pedro Fonseca

Referenced Documentation

Description	Author

Table of Contents

1 Introduction

The aim of the IMS mobile application project is to deploy an operational mobile solution to EMSA users of integrated services. This document describes the technical design of the mobile applications and middleware.

2 Technical Overview

This section provides an overview of the mobile applications (WP1), middleware (WP2) components and these components integration with EMSA interfaces.

The mobile applications are developed for Android and iOS platforms. The Android implementation is optimized for large screen sizes. On iOS, the business logic is shared by the smartphone and tablet user interface implementation. To ensure that only authorized users are able to access EMSA resources, the mobile applications depend on the EMSA mobile access gateway to authenticate a user and obtain his role.

The mobile applications middleware orchestrates EMSA IMDatE and CSN services to provide a REST facade for the mobile applications. The middleware consumes the EMSA services through the EMSA Oracle Enterprise Service Bus (OSB). In addition to the IMDatE and CSN services, the middleware exposes the EMSA Electronic Nautical Chart service.

The cloud storage service component is still under assessment and is out of the scope of this document.

Error! Reference source not found. illustrates the global architecture of the solution.

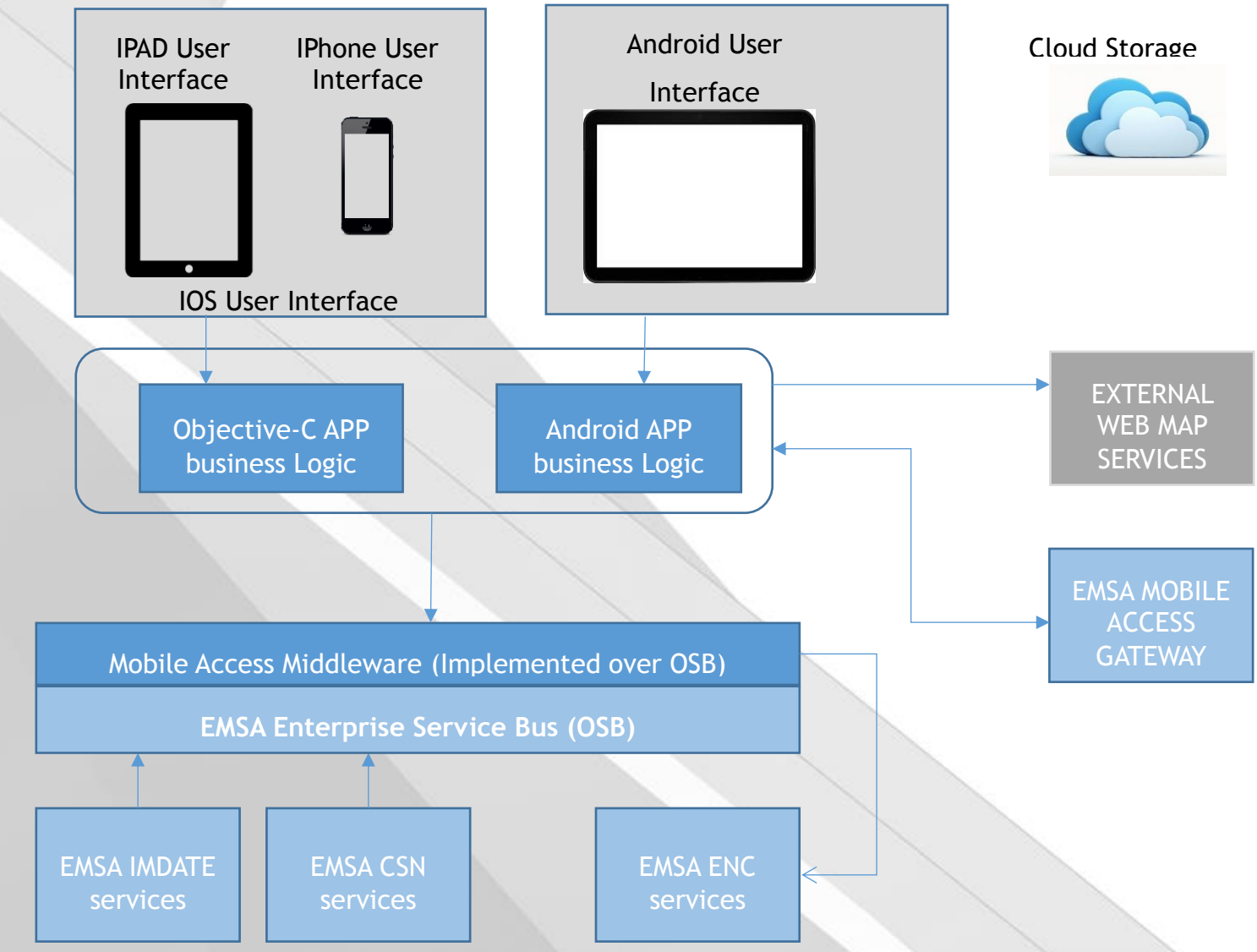


Figure 1 - High level view of the application architecture.

3 Mobile Applications

The mobile applications follow a Model-view-controller (MVC) architecture, centered on a view controller which orchestrates the map and related views. The map view controller depends on several loosely coupled modules that implement the application persistence and web service access layers.

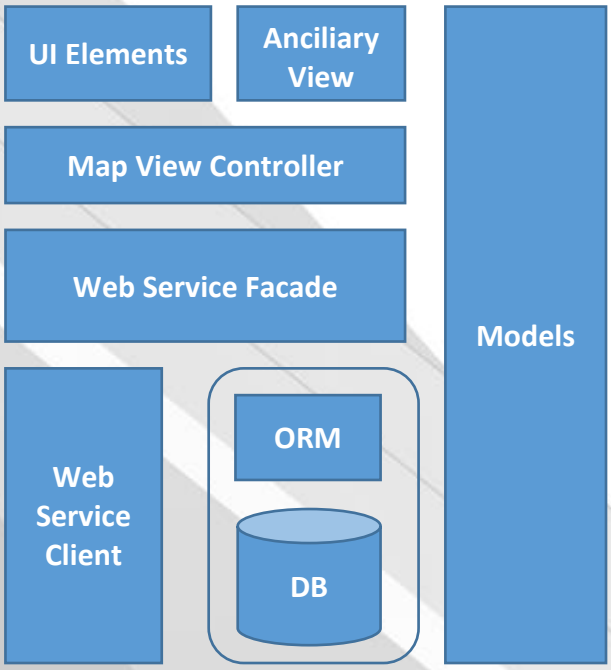


Figure 2 - Mobile application modules.

The following sections describe in detail the interaction between the application models.

3.1 Cache Design

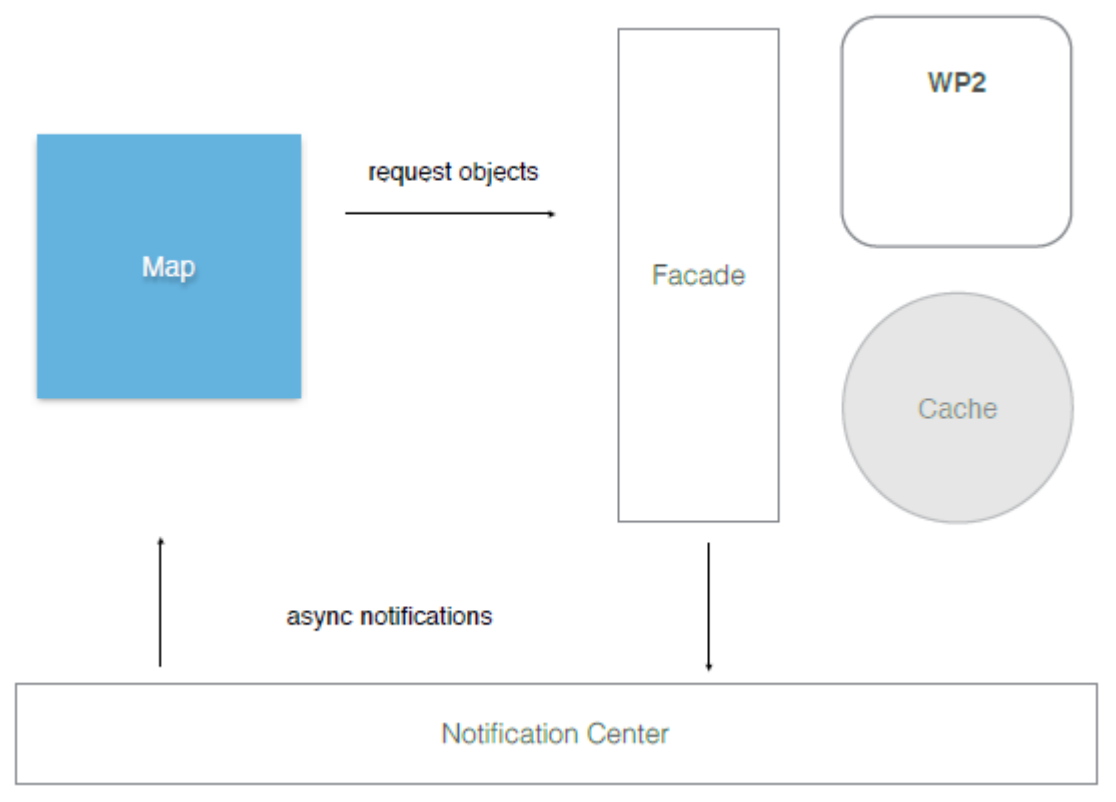
The application implements a cache for feature objects, like positions reports, oil spills, incident reports and a cache for map tiles. The feature cache is backed by a SQLite database exposed by an ORM abstraction layer that maps entities to database tables. To avoid costly geospatial queries, the map is split in a fixed size grid, where each cell is identified by a grid index or a “geohash”. Feature objects with a geographical reference are indexed by the respective map cell.

3.2 Maps Management

The map view, the main view of the application, displays several layers which are defined by the user or the user role. This view is managed by the map view controller, which responds to events initiated by the user on the map, such as a pan or a pinch gesture. When the region of the map

changes, the map view controller requests new tiles for the map and notifies the web service facade that it needs content for the defined layers. The map requests are fulfilled asynchronously by the web service facade or by the map tile client.

The web service facade manages the web service clients which consume the mobile access middleware and the feature cache. All requests are satisfied from the cache.



3.2.1 Custom map tiles

The map view display native maps (maps that are provided by a system framework) and custom maps, like the maps provided by EMSA ENC service or maps provided by WMS enabled servers. Map tilling, that is, splitting the map into square tiles with the same size, is handled by the map view.

The custom maps are implemented as providers that are instantiated by a factory in behalf of the map view controller. The tile provider handles requests for map tiles, converts the tile index to a projection that is supported by the web map service (e.g. EPSG:4326) and fetches tile from the map service.

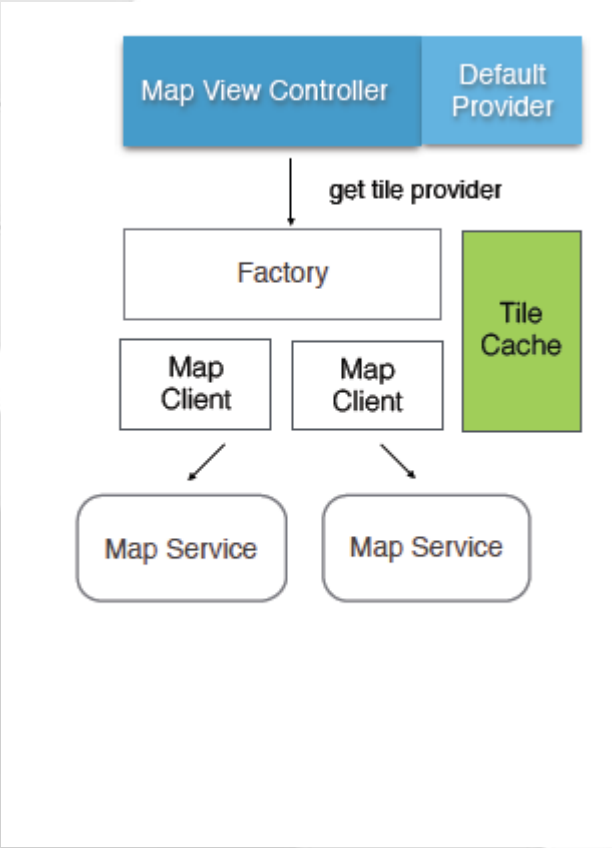


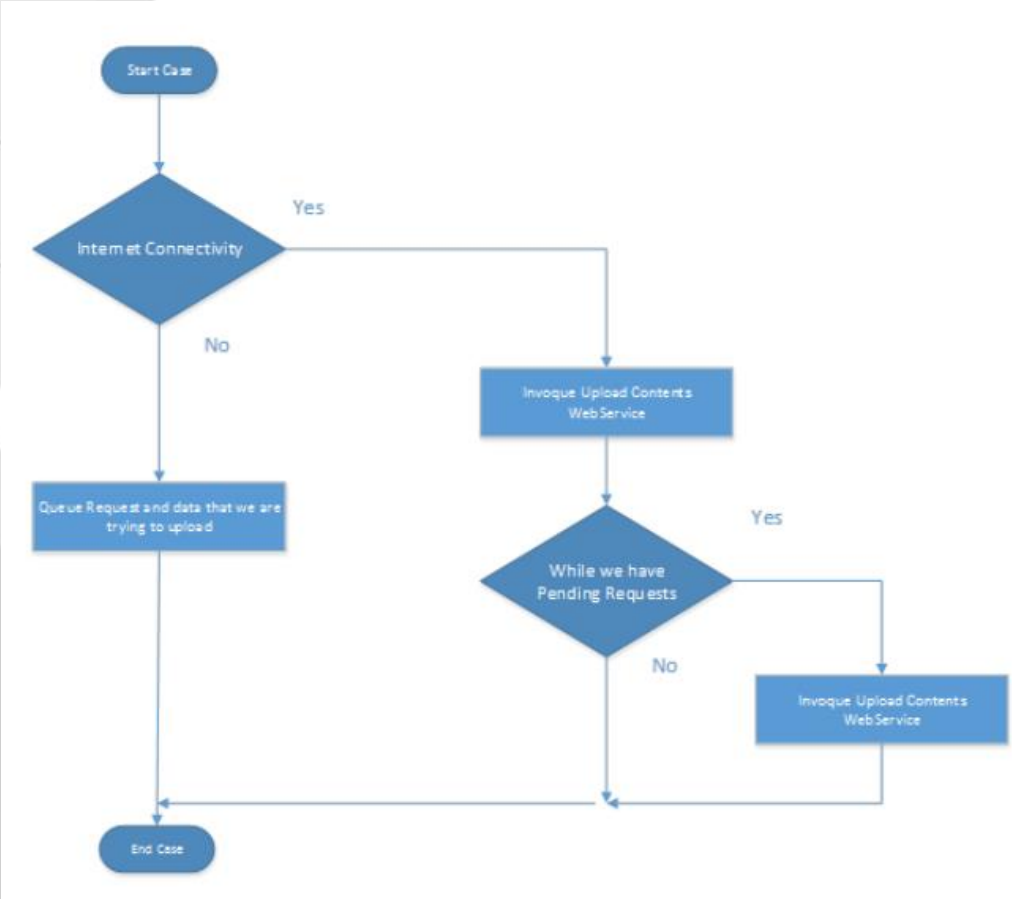
Figure 3 - Map providers.

.

4 Offline Usage And Information

If the mobile app tries to upload content when the device is offline, the request is serialized and stored in a persistent manner. When the application detects connectivity, the request is submitted again on a background thread.

The map tile cache supports the display of custom maps when the device is offline.



5 Mobile Access Middleware

The mobile access middleware provides a REST interface for the mobile apps. The middleware connects to EMSA Oracle Enterprise Service Bus (OSB) to consume EMSA IMDatE and CSN services and implements the business logic that exposes these services to the mobile applications. The middleware also acts as a gateway for services that are not plugged in the OSB.