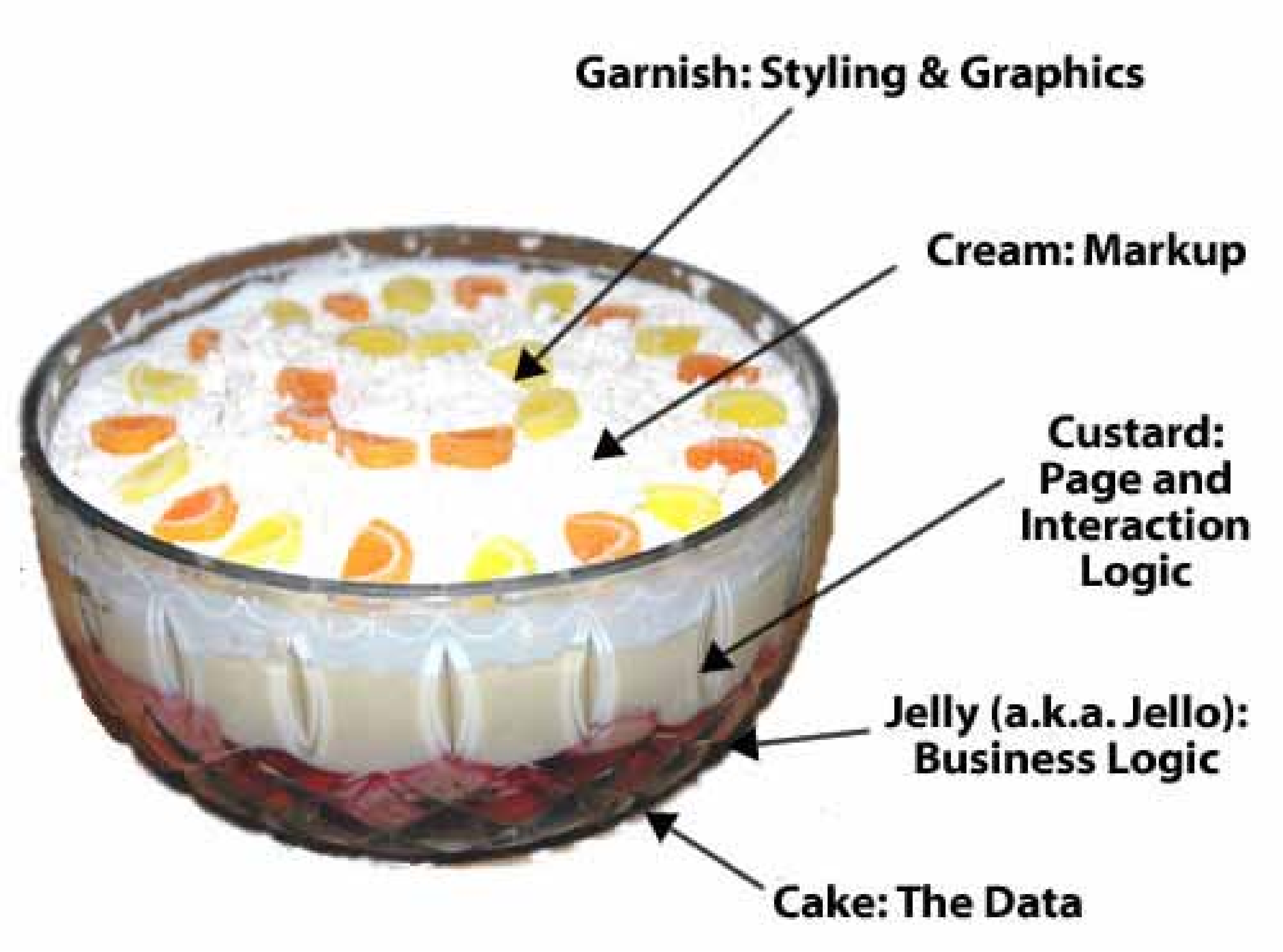
EDQ’s Delicious Architecture

EDQ’s Architecture
as an English Trifle

(based on a presentation by Cal Henderson)


Garnish: Styling & Graphics
Cream: Markup
Custard: Page and Interaction Logic
Jelly (a.k.a. Jello): Business Logic
Cake: The Data

Cal Henderson was the Lead Engineer in building Flickr and has a wonderful description of a layered architecture using a food analogy that is rather tasteful (couldn’t resist). Here is a brief description of each of the layers. What follows is a paraphrasing of his presentation.

“

Software can be divided into layers, each layer having a particular set of responsibilities. The higher a layer is, the closer it is to the user, with the topmost layer being the one responsible for interacting with the user. The lower a layer is, the closer it is to the underlying data and logic for manipulating that data.

Layers are “black boxes” in that their inner workings are hidden from other layers. A layer usually communicates only with the layers immediately above and below it, and only through the interface defined between layers. By “interface”, we mean the set of allowed communications between layers.

Garnish

Styling is the cosmetic stuff – it specifies things like fonts, graphics and the like. There is no layer above this one -- rather, it’s the user who exists above this layer. The layer below this is the Markup, which get a “look” applied to it by this layer.

Cream

The markup is the user interface which is displayed. In a web application, it’s the HTML, in a desktop application, it’s the GUI, and to applications accessing this application via the API, it’s the XML that is returned by the API calls.

The layer above this one, Styling, applies a “look” to this layer. The user’s commands are received by this layer and sent to the layer below -- Page and Interaction Logic. The layer below then provides the responses to those commands, which are then displayed in this layer.

Custard

The page and interaction logic determines how users interact with the business logic. It determines things like which data is displayed to the user or what steps the user must take to add, change or delete data from the system.

This layer receives user commands from the Markup layer above and sends back the appropriate responses. It may need to perform some kind of action that requires fetching data or manipulating it; it does this by accessing the layer below it: the Business Logic layer.

Jelly aka Jello

The business logic is the set of rules which define how data is accessed and how it may be manipulated. The only way the data should be accessed is via the business logic. Cal says that it “defines what is different and unique about the application”.

This layer receives user commands from the layer above it; it sends its responses to those commands to that layer as well. Whenever this layer needs to access data, it does so from the layer below it, the Data layer.

Sponge Cake

The data layer forms the underpinning of the application. As with trifle, this layer is big, solid and reliable. All layers above this are “transient and whimsical” (He will always have a home at Oracle.)

This layer responds to requests for data or to change data from the Business Logic layer above it. It sends either the requested data or error or confirmation messages to that layer in response to the requests.

And there you have it, EDQ’s wonderful architecture presented in a way that is, shall we say, sticky?

Title: EDQ’s Delicious Architecture
Author: William Indest
Wranglers: Brian Nault and Todd Lesser

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

EOF

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.