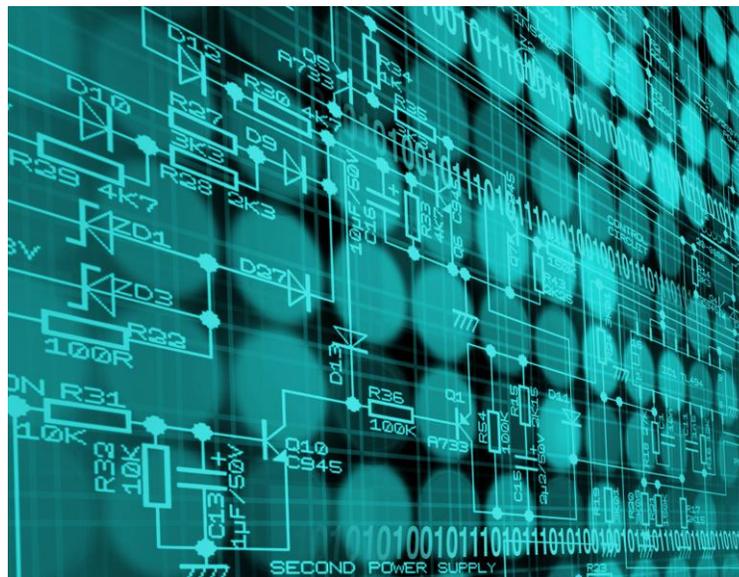


Decision Support: Building vs. Buying

Search and Discovery Application Assessment



Encompass White Paper
2014

www.perceptionsoftware.com

Contents

Introduction	3
Getting to the Decision Point: Buy or Build	3
“Confronted with the Problem”	3
Pros and Cons: Build vs Buy	4
The REAL cost of building a solution	6
Review of a COTS solution	7
Architecture	8
Platform Components	8
Security	8
Building a Knowledge Catalog	11
Building Encompass Applications	12
Summary	12

Introduction

The purpose of this paper is to explore the buy vs build process and how companies evaluate building their own applications vs. implementing commercially off-the-shelf products (COTS). In the following White Paper we will be examining a scenario around building your own search-based application vs buying a COTS solution. Taking this step is an important consideration for an organization, with the decision having a long-term impact on resources, time-to-market for new products and overall business efficiency. The following overview will provide some insight into the decision-making process and how you and your organization can arrive at the best decision.

Getting to the Decision Point: Build or Buy

You are the head of Business Applications within a large organization. Over the years, you've developed and deployed a number of business critical applications to cater to the needs of your business users. The fact is you and your teams are have done an adequate job of building what's needed but with this effort comes a tremendous strain on resources. The questions you are wrestling with are not only are we good enough at what we do but ***what's in the best interest of all the parties we serve?***

“Confronted with the Problem”

Your team has just completed an implementation of an application to address the needs of your product groups - users in the entire product lifecycle chain. This solution is one of many systems these product groups use. Everything went according to plan, the system is functioning as expected and users seem happy. Over time, you start getting complaints your users cannot find the information they need, when they need it. They have to hop between multiple systems trying to find the product data needed for their products. Getting the right information becomes a time-consuming chore with the design, product and supply chain teams spending more time searching for information than using it to design, build and support great products. You have a problem but not to worry, you've been here before. As in the past, you have a couple options to address this issue. One is to consolidate all the data into one system. While this sounds like an interesting option, the reality is the users need a unified view of the data and it is

housed in major enterprise systems across multiple domains.

Even a cursory glance at this option suggests that there is not enough time or money to move down this path.

The second option is to build or buy a new system that provides a consolidated view of all your product data. After further review and evaluation, you decide to implement a new solution both from a cost and technical perspective and start evaluating whether to build or to buy this solution.

Your evaluation process looks something like this. If you bookend the problem, the two key questions in the formulation of your offering would be:

What is the minimum specification we need to capture the largest possible set of use-cases that can be delivered in the shortest time possible?

What is in the best interest of all stakeholders over the long term where strategic capability, cost and competitive means are concerned?

The wrestling match now begins. Is there a way that we can solve a majority of the user's requirements quickly without being boxed in strategically over the long haul?

Pro and Cons: Build vs Buy

The build vs buy is a challenging decision for any organization. Let's look at both sides of the question to see what the potential advantages and disadvantages of both options might be.

Build – Pros

- Customization to meet the specific needs of your business
- Control over the end product
- Lower upfront cost
- Support team has better knowledge because it's built in-house

Build – Cons

- Lack of software development skills – it's not a core competency
- Upgrades are time-consuming and need to be done frequently
- Cost to design, develop, test and train is expensive and recurring
- No economies of scale

- The complexity of normalizing multiple data sources is a not a trivial process
- Point solutions are typically used for one project in one group for a short time (throw away code)
- Many of the capabilities being built are commodity items like admin panels and logging solutions when resources should be focused on the core application

Buy – Pros

- Software is the core competency of the vendor
- Real-world use – implemented by many companies
- Deployment is faster; saves time to get users on the app
- Economies of scale
- All the heavy lifting – design, development, testing – done by vendor
- Smaller amount of internal support required

Buy – Cons

- Upfront costs are usually higher
- Upgrades to software may not benefit your organization
- Some features and benefits may not be relevant to your needs

These are some of the top considerations to factor in to your decision-making process. The buy vs. build decision cannot be taken lightly and numerous factors must be analyzed to understand the true long-term impact. Even then, the changing dynamics of an organization often drive unexpected new requirements that then have to be added to the application.

If you're currently at the buy vs build juncture, the following questions may help guide you in determining the best answer for your organization:

1. What is the organizational problem we're trying to solve?
2. Who will be using the new application?
3. How many people will use it?
4. What is our timeframe to implementation?
5. What is our budget?
6. How much customization is required?
7. What ongoing support will be required?
8. Do we require control over the app?
9. How will software upgrades be managed?
10. Ultimately, what defines success?

There is no easy answer to the buy vs build decision and your company's different needs should be taken into consideration. By answering the questions above and others that more directly relate to your organization's requirements, you can certainly minimize the short and long-term risk of making the wrong call. Selecting carefully now will streamline your project, minimize errors and make your organization more efficient for years to come.

The REAL Cost of building a solution

When considering all the costs that go into building a search-based application, your organization should estimate not only the costs to get the application built but the lifetime costs of supporting, maintaining and upgrading the application as well.

The chart below is a quick example of how quickly costs can add up when considering all the factors that go into an in-house solution.

Building a Search Based Application (EST)		
Activity	Est. in Man Months	Cost at 200K per year
Requirements gathering, specifications building	2	
Development	24	
Product Testing QA	6	
Sub Total	32	\$ 531,200
<u>Year over Year Maintenance</u>		
Program Management	20% FTE	\$ 35,000.00
Product Support	15% FTE	\$ 15,000.00
Development (app/data)	2 times 15% FTE	\$ 45,000.00
Sub Total		\$ 95,000.00
Total Year One		\$ 626,200.00

Opportunity Cost to consider

- Best use of scarce resource
- Taking skilled, quality experts and making them be simple task project managers
- Effort spent developing/testing code instead of finding trends, patterns and problems specific to your business
- Losing costly investment in skilled subject matter experts

- Distraction -- When there is problem or burning new need in the software, it becomes a high priority
- Key decision makers with detailed knowledge of a Customer problem will be distracted to solve a software maintenance issue ... It's NOT the task of fixing...it's the unplanned interruption that can be disruptive.

Review of a COTS solution - Encompass

Encompass is an intelligent navigation solution that delivers product data unmatched in quality, immediacy and ease-of-use. It gives everyone across the information chain the ability to make smarter decisions, faster. It's an enterprise-grade platform for building information discovery and decision-support applications. Whether your users are engineers, product designers, operations, supply chain, executives, vendors or partners, Encompass provides purpose-built solutions to meet the needs of your business users.

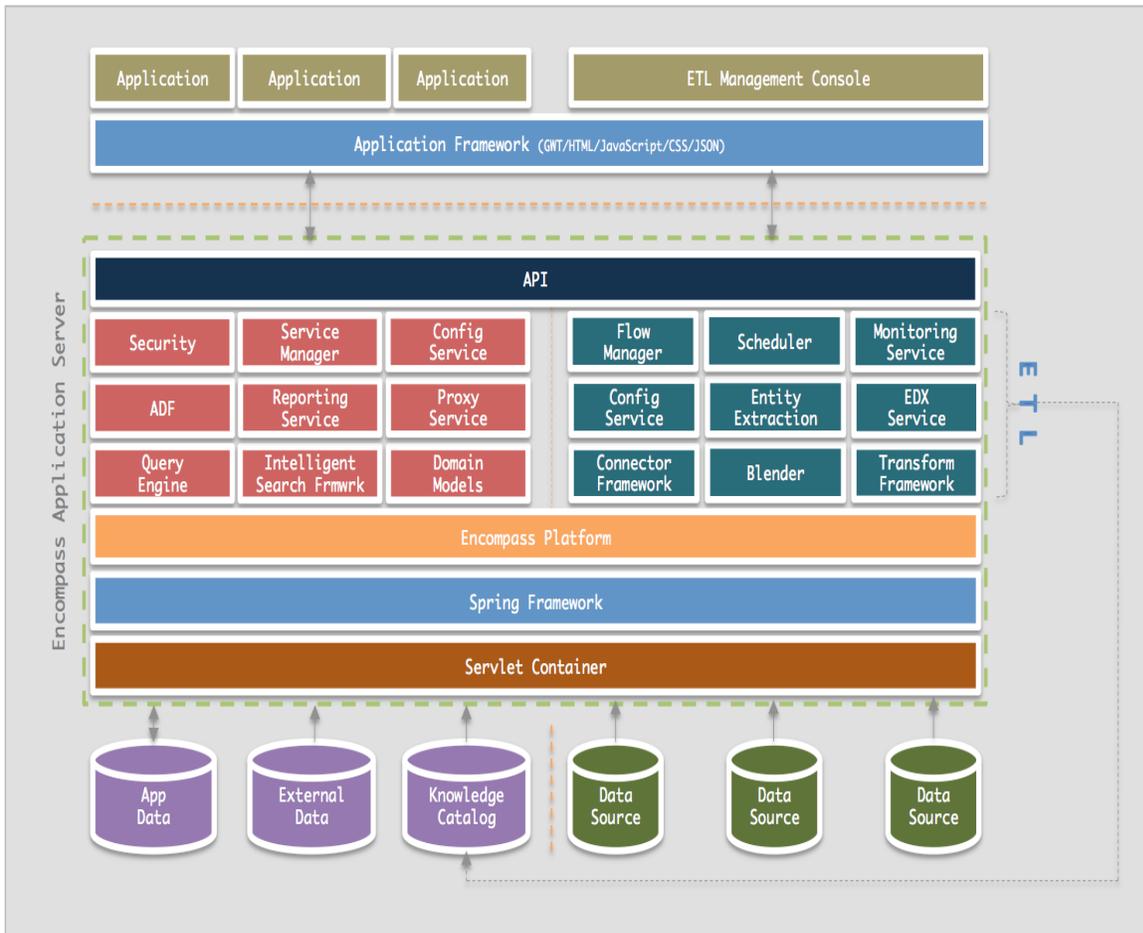
Encompass is currently in use in large enterprises just like yours. For years, thousands of users have been leveraging Encompass to navigate through the complex maze of their product data, both current and legacy, to make smarter decisions and be more productive. It provides all the advanced functionality you'd expect from a robust enterprise application - performance, security, scalability, reliability, reporting – without the complexity and challenges inherent in large scale enterprise apps. And it's a cloud-based app with mobile functionality.

The Encompass platform and applications are developed and maintained by an exceptionally talented and experienced engineering team. For those specialized times when one of the COTS solutions does not contain the functionality you are looking for, the platform provides all the tools and APIs to mold them exactly to your needs. Detailed documentation, tutorials and support to build both simple and advanced Encompass applications is available as well.

Following sections discuss the architecture and its components in detail that will talk directly to your developers, both application developers as well as data developers.

Architecture

Encompass Platform architecture is an extensible, [plugin-based](#), [micro-services](#) architecture. [Spring Framework](#), which is a proven technology for building enterprise Java applications, is the foundation for this platform. Following picture provides a very high level overview of this architecture.

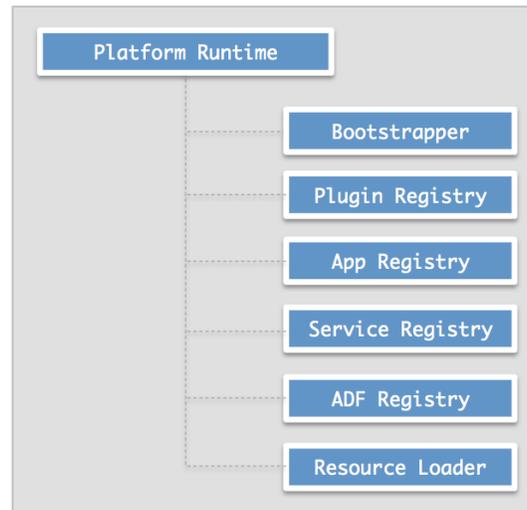


Platform Components

The following section offers some of the high level components of Encompass Platform.

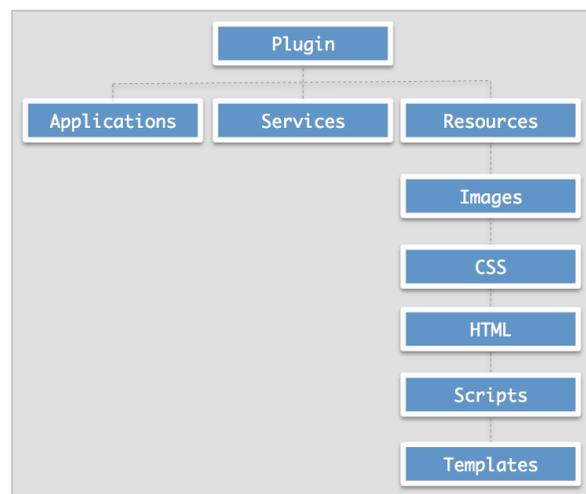
Platform Runtime

Encompass Platform Runtime is a core component that manages all the high level components of the platform. It contains different sub-components: Bootstrapper, Plugin Registry, App Registry, Services Registry, Servlet Registry, ADF Registry and Resource Loader.



Plugin

An Encompass Plugin is a deployable unit that contains related components like Applications, Services and Resources. The Plugin Registry loads all plugins up.



Application

Application is the user interface purpose built for specific users of a particular domain. Applications use different *services* to interact with the persistence layer. Applications can talk to each other, as well. Applications require different

resources, which are bundled along with your plug-in. The App Registry manages all applications.

Service

A service is the server side component that contains the business logic and has access to the data. Services can talk to each other for cross-functional features. The Service Registry loads all the services.

Service Manager

The Service Manager is the most critical component of Encompass Platform. It is the entry point for all applications making calls to the server. The applications communicate with the services thru request/response protocol. All the requests go thru service manager and it delegates them to the Service Registry to process the requested service.

Application Data Framework (ADF)

ADF provides a common persistence mechanism for all your application data. If your application needs to store and manage data, you can build a model, persist it using ADF and use its services to interact with that data. ADF contains data like user profiles, session history, logging information, etc. ADF Registry provides access to all the ADF Repositories.

Knowledge Catalog

A Knowledge Catalog is the enhanced aggregated data layer for Encompass. It is built for fast and intelligent search. Some of this intelligence is in the data transformations we do on the source data.

Intelligent Search Framework

This framework provides smarter search capability for your services. Critical components of this framework are Inference Engine, Adaptive Search, Discovery Engine, Search Analytics and Query Translator. Domain Models, Knowledge Catalogs and App Data are the inputs for this framework.

Data Connectors

Data connectors are the workhorse of pulling in all the data from a variety of sources and adding intelligence thru transformations before going into a knowledge catalog. It has different sub-components like the connector framework (for extracting data), a transformation framework (from transforming data), a flow manager that pieces together different logical steps, a monitoring service to check on a process, a scheduler to schedule your data connector jobs, etc.

Application Framework

This component provides all the APIs and protocols for defining and building your applications. Applications in Encompass can be built using GWT or plain JavaScript.

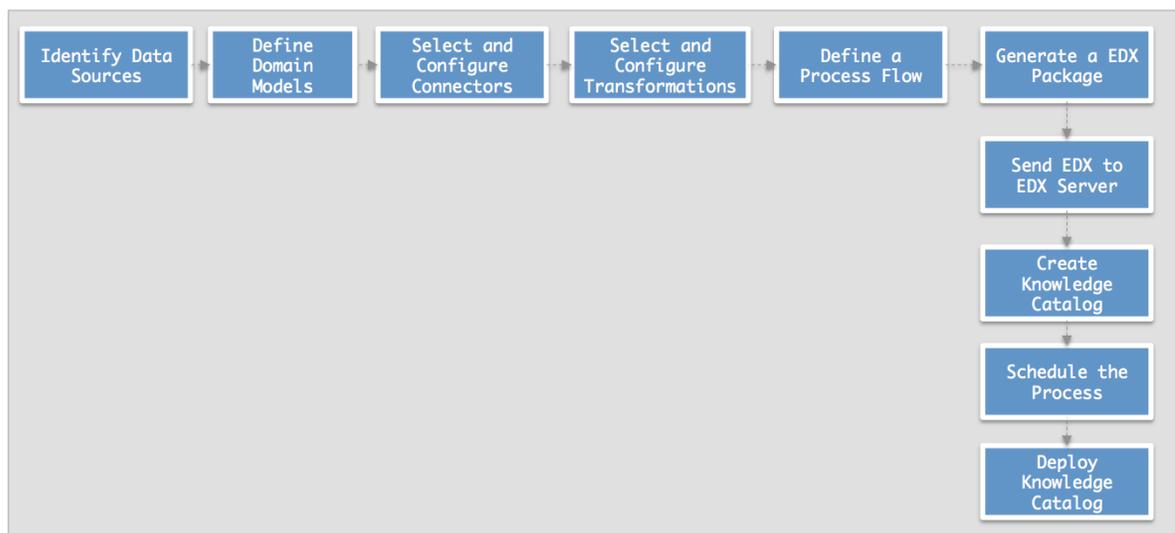
System Security

Encompass provides enterprise-grade security in a layered approach. Encompass supports standard enterprise authentication schemes used in most SSO (Single Sign On) environments. Encompass will work with Oracle OAM, CA-Site-minder, MS SSO and SAML based infrastructures. Encompass can also be configured to work with standard LDAP servers.

Encompass provides the ability to restrict SSO authenticated users with application level security. This will deny access to the entire application and may be synchronized with externally managed roles and groups. Encompass allows complete items to be removed from view of the users. Encompass allows for security policy to be defined based on data in the encompass knowledge catalog, and this policy associated to a group of users. Encompass can be configured to restrict access to data based on the source. For example, an external user may not see any data from the ERP system.

Building a Knowledge Catalog

The following picture illustrates a typical flow for creating a knowledge catalog.



Building Encompass Applications

A typical workflow for building Encompass Applications (assuming your data is defined and extracted) may look like this:



More detailed documentation and an example with a tutorial is available for a hands-on walk through.

Summary

Build vs Buy is a complicated decision process that requires a lot of analysis. But often, under pressure to deliver solutions quickly, companies rush to build what they think is a simple solution that in the end turns out to be far more complex and costly than they ever imagined. Encompass provides the most cost effective solutions for navigating your many data sources and delivering detailed, accurate results in seconds.