Architecture Reconstruction & Documentation "APPLINKS"

Vikram Sri Nitesh Tantravahi
Drexel University, vt87@drexel.edu

Abstract *- Complexity is always a technology hurdle that arises from different streams of processes during any workflow operations in a software tool or a system processing various tasks or any camouflage of functionally operative elements. Scientific world is meant to work with application of technology but in today's world, everything works by "APP-link"ation using technology. Applinks is one such powerful technical and implementation software that enables deep-linking feature among various applications in a particular platform. The major interest behind the study of this technology is the cross-platform integration of this complicated and fragmented application development. The usual problems associated with the apps in various platforms are more logical and operation oriented than the GUI components or the visual aspects involved within the existing applications and the newly added features of the application. Another major and important aspect these days for successful software like Applinks is that it is at reach for any developer or a firm that is interested to use that particular tool for various purposes – Open Source. Applinks reduces such problems involving the integration components and thereby removes the complexity involved in the application and makes life simple.*

Index Terms – Application, cross-platform, deep linking, technology integration, mobile platform, open source, API reference, platform independent.

INTRODUCTION

Technology is an interesting genre of science that always acquires any new change in the environment. Application development is one such advancement in recent times that technology has absorbed, learnt, accepted and implemented in various forms that has changed the world. Every "*APP*" that is developed has functionalities and operative features as well as distinctive purposes. It might be a game, a business tool, a reading or learning application or some kind of an entertainment source for the user. Sometimes we might have to use multiple applications for a purpose, which is most common case for a user in today's *APP* world. Finding an application complementing existing applications is a very tedious task. To club or merge two applications for a single purpose or for a better usage, all the needed features cannot be hard coded and integrated into another application. There are various reasons for this because the internal functional integration that is termed as "deep linking" has a lot of standards that are to be taken into consideration during this process.

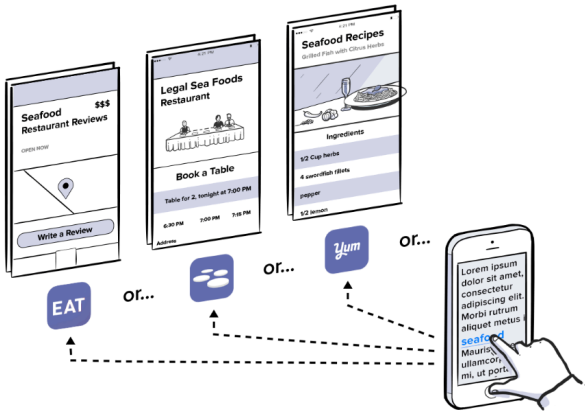


Fig 1: Deep Linking to various applications.

All this process mainly deals with the mobile development in various platforms like iOS, iPhone, iPad, Android, Windows Phone, Windows, Universal Windows, and Web Fallback etc.

ARCHITECTURE OVERVEIW

The best way to depict any architecture is through illustrations and pictorial representation because that is the best way of explanation for a better understanding. Conceptual elements and their relationships can be clearly defined in the architecture and mainly explains the high-level schematic of the architectural breakdown.
The architecture of any system can be usually divided into two major categories. They are:

- ✓ Functional Architecture
- ✓ Operational Architecture

Facebook mainly operates Applinks and its functionality and existing architecture are also defined and implemented by the Facebook developers. The Facebook docs contain a special category among their products called App Links. This Applinks functionality is as simple as its name says. This is a feature that provides links among various apps and it relates to other applications through the API that is integrated along with other application. This can clearly be observed in the figure below.

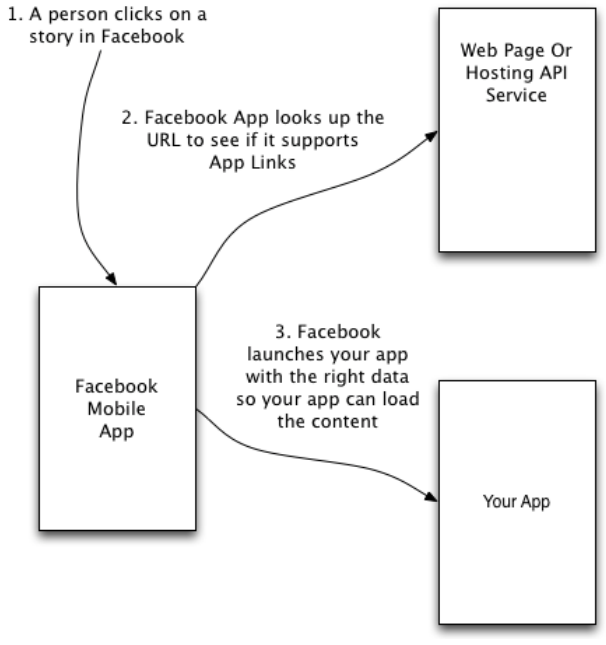


Fig 2: App Links as a Facebook Product

For a better compatibility with the operation and functionality of the architecture we can consider by taking an example, which is implemented in iOS platform.

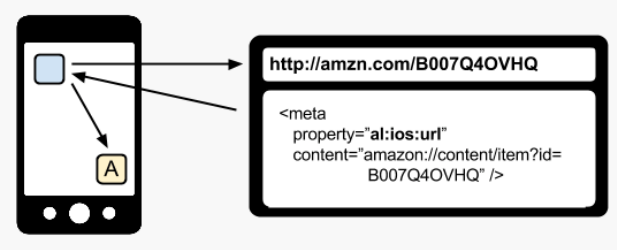


Fig 3: Amazon link as an example.

So when we have a link for an amazon product in a book or a reader, and when we click on that particular link it usually opens in the web browser in that particular platform. So if this is integrated using Applinks by adding some lines of code in the app by the developers, it opens up the existing amazon app in the phone and sign-in and makes it simpler for the user.

In terms of code, in iOS

UIApplication.SharedApplication.OpenURL("http://amzn.com/B007Q4OVHQ");
is the normal representation but this is changed to the following code for the Applink integration.

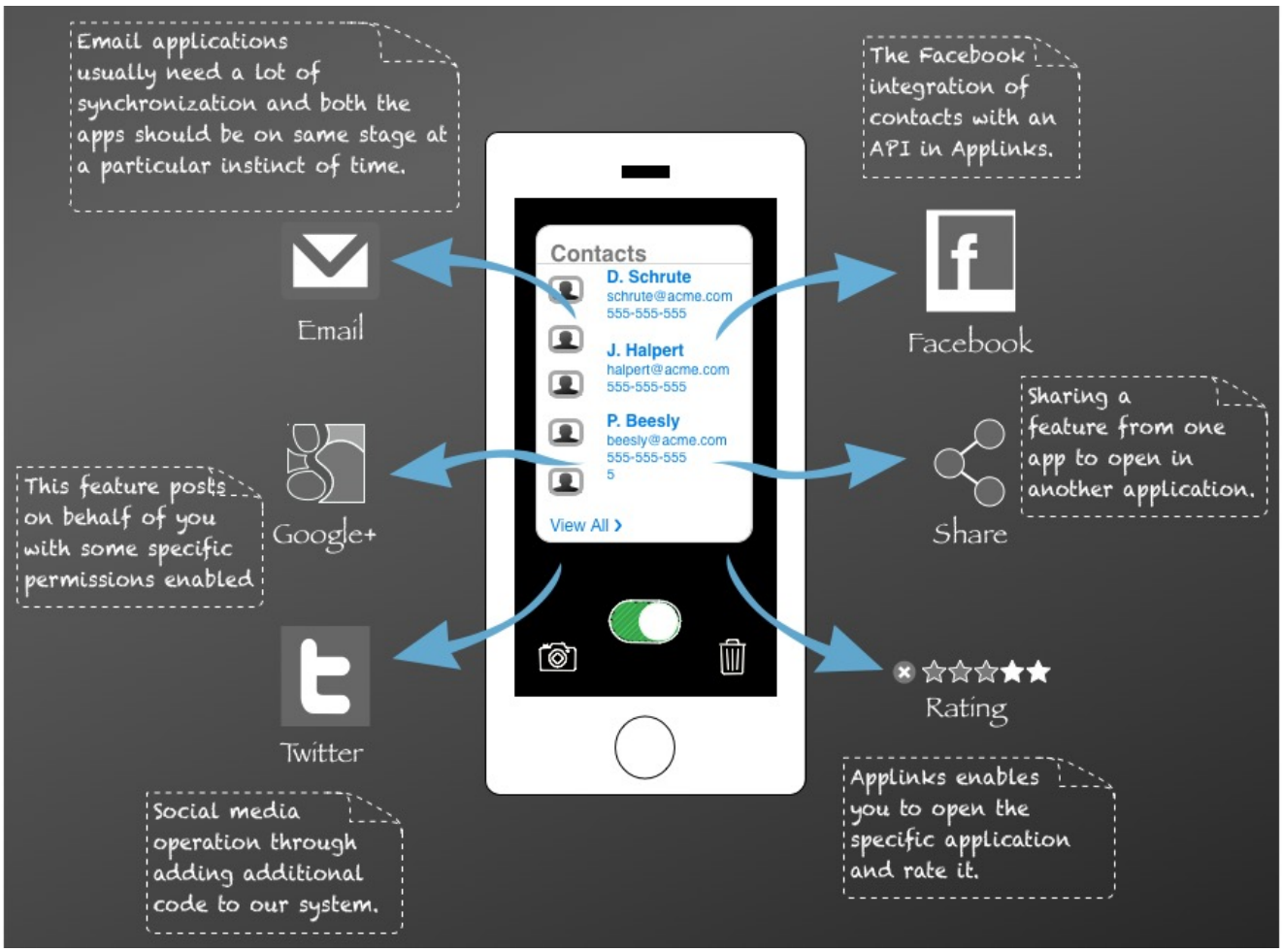
var result = await Rivets.AppLinks.Navigator.Navigate("http://amzn.com/B007Q4OVHQ");

So the app finds for this metadata and redirects to the desired location when required.

The same functionalities and application integrations are also performed in other platforms but in different forms and logics applied accordingly. Applinks is very feasible and flexible in terms of the usage and application. This is the major advantage of this tool and its growing usage compared to other platforms and other systems in this competitive world.

FUNCTIONAL ARCHITECTURE

The functional architecture of Applinks is described in the figure below.



As described in the figure, the architecture of APPLINKS enables the features similar to this application. So considering the example above, the contacts that are to be shared among the various platforms and the features to be used in other applications are mainly based on the linking of the functionalities that exists among the two linkable applications.

OPERATIONAL ARCHITECTURE

This is the same example that we have discussed above.

Bolts and Rivets are the applications as explained in the figure that enable the features and functionalities. C# implementation is same as the above but it has its own uses and advantages for people implementing or working on that platform.

CONCLUSION

From all the inferences and references and available documentation, implementations and available sources that are related to APPLINKS, we can say that this is an extremely powerful tool and this has a lot of documentation which is and always will act as a reference architecture to many applications and also to the technology world that exists today. This will be a major contribution in application development for mobile platform and the future is going to be more exciting because of such wonderful architectures.

REFERENCES

[1] *http://applinks.org - The official implementation website of APPLINKS.*

[2] https://developers.facebook.com/docs/applinks - Facebook Developers Product Documentation.

[3] http://mashable.com/2014/04/30/facebook-app-links/

[4] https://developers.urx.com - URX which is the similar integration of Applinks.

[5] http://redth.codes/what-are-app-links/ - Most famous documentation of APPLINKs in social forum.

[6] https://github.com/BoltsFramework/Bolts-iOS

[7] https://github.com/xamarin/Rivets