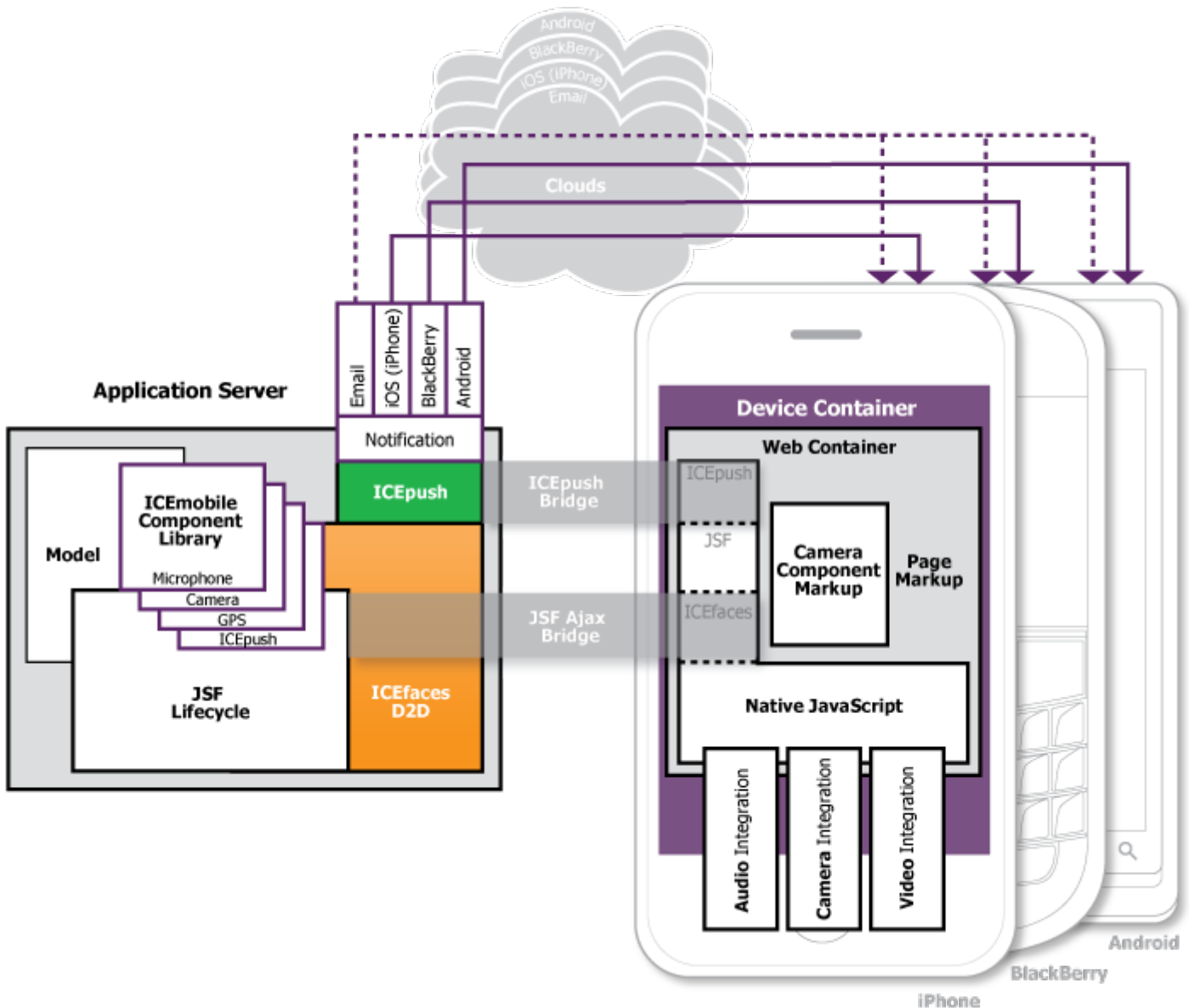


ICEmobile : ICEmobile JSF-ICEfaces Architecture

This page last changed on Aug 07, 2012 by [steve.maryka](#).

ICEmobile JSF/ICEfaces Integration Architecture

ICEmobile JSF/ICEfaces support is based on JSF 2 standard and has the [ICEfaces 3](#) framework at the core of its architecture, as illustrated in the diagram below.



Major architectural components of the JSF/ICEfaces integration include:

- **JSF 2:** The JSF 2 core framework provides the standards-based foundation.
- **ICEfaces 3:** The core ICEface 3 framework extends JSF 2 in key ways that improve developer efficiency, and extend the JSF feature set. In particular, [Direct-2-DOM Rendering](#), and [Automatic Ajax](#) simplify application development for both mobile and desktop application variants.
- **ICEpush:** [Ajax Push](#) capabilities in ICEfaces 3 are based on the notification mechanism provided by the [ICEpush](#) core. ICEpush has been augmented with cloud-based notifications, where priority notifications can be delivered by alternate cloud-based mechanisms when the normal ICEpush

connection to a client device is interrupted for any reason. Different cloud connectors are provided, including generic connectors like email, and device-specific connectors for each supported platform, including:

- [Apple Push Notification Service](#)
- [Android Cloud To Device Messaging](#)
- [Blackberry Push Service](#)
- **ICEmobile Component Suite:** A suite of JSF components that provides the UI controls needed to build mobile applications that deliver a native device user experience across a wide spectrum of mobile device types. The suite includes:
 - **Layout and Navigation Controls** for page layout, navigation, and theming
 - **Input and selection Controls** for handling user input
 - **Media Controls** for presenting audio, video and images
 - **Native Controls** for accessing device capabilities like camera, camcorder, and microphone
- **[ICEmobile Containers]:** Native device integration requires a client-side implementation that binds a JavaScript interface to the native platform support for a particular device capability. These integration points combine with a generic web container to provide a hybrid browser implementation with extended capabilities that can be accessed with native components from the mobile component suite. Device containers are installed on the device as native applications, but their implementation relies on a web application to present the user interface via standard HTML markup.