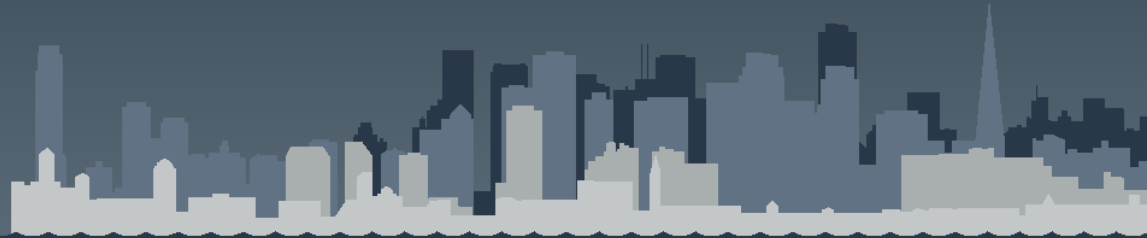


TIZEN™ DEVELOPER CONFERENCE MAY 7-9, 2012



**Helium Push Notification
Architecture
Jimmy Dee, Urban Airship**

Introduction

Urban Airship's Helium service is a mature, robust solution for push notifications.

Push notifications are critical to the business strategies of a broad array of companies.

Helium has a track record of easy implementation by many clients (ESPN, et al.).

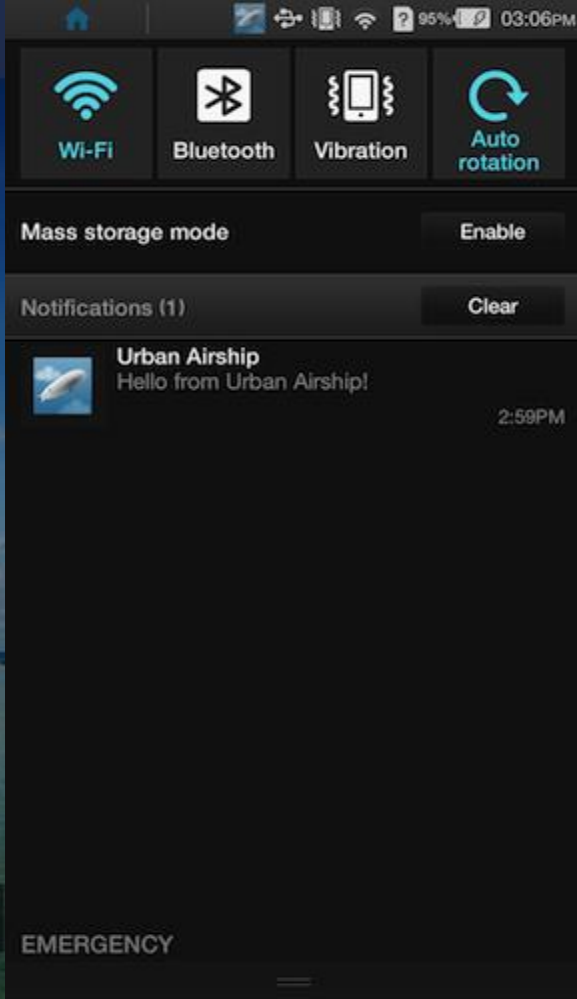
We've recently built a C++ Helium client library for Linux and are working on Tizen integration.

Helium Push Notifications

- Client devices are always connected to Urban Airship's Helium cloud service, allowing for prompt delivery of notifications, even when an app is not running.
- Separate, high-profile channel directly to a user.
- Notifications can be sent directly to a specific device, to all devices (broadcast) or a subset identified by a tag or other criteria (such as current location).
- Notification payloads can direct the user to a specific view in an app, so a breaking news push, e.g., can take a user directly to the related story when tapped.

Helium Push Notifications (cont'd.)

- Apps display badges, where available, to indicate the number of unread push notifications.
- Apps can brand their pushes with custom icons and sounds.
- Users can customize their pushes by opting in or out of each app's push notifications and also enabling/disabling sound or vibration per app or setting a quiet time per app.
- Publishers use a web application or (typically) a REST API.
- Publishers have access to reports on analytic data (statistics regarding number of pushes, opens, opt-ins, etc.).

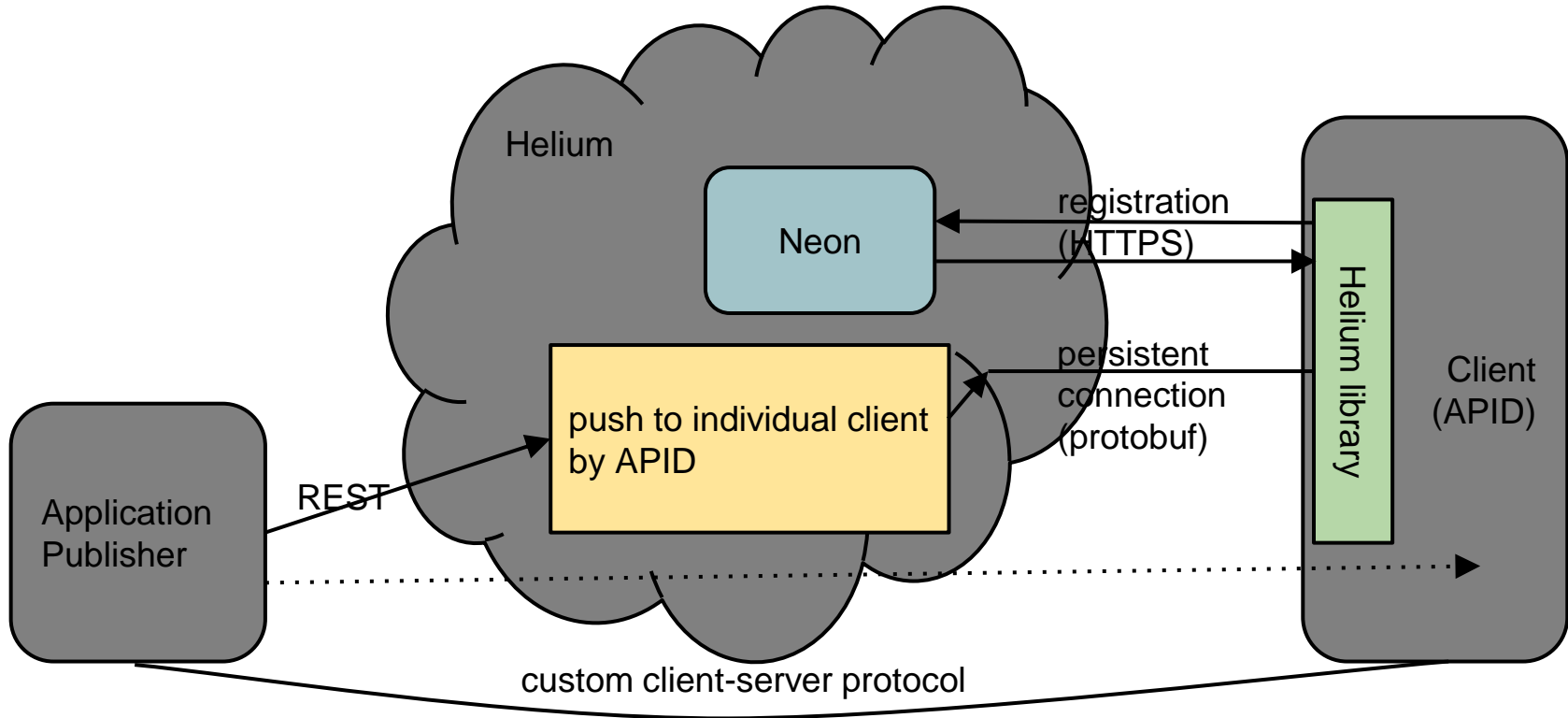


*

Helium Use Cases

- Dictionary.com sends a word of the day. This is a broadcast to all users who opt into push. It keeps their ad-based app high in users' minds and drives users to the app.
- The DNC uses push notifications to keep volunteers and interested voters informed of campaign developments and events.
- ESPN sends sports scores and breaking news during and after games.
- Tapulous reports friends' scores to draw players back to the game.

Helium Architecture (unicast push use case)

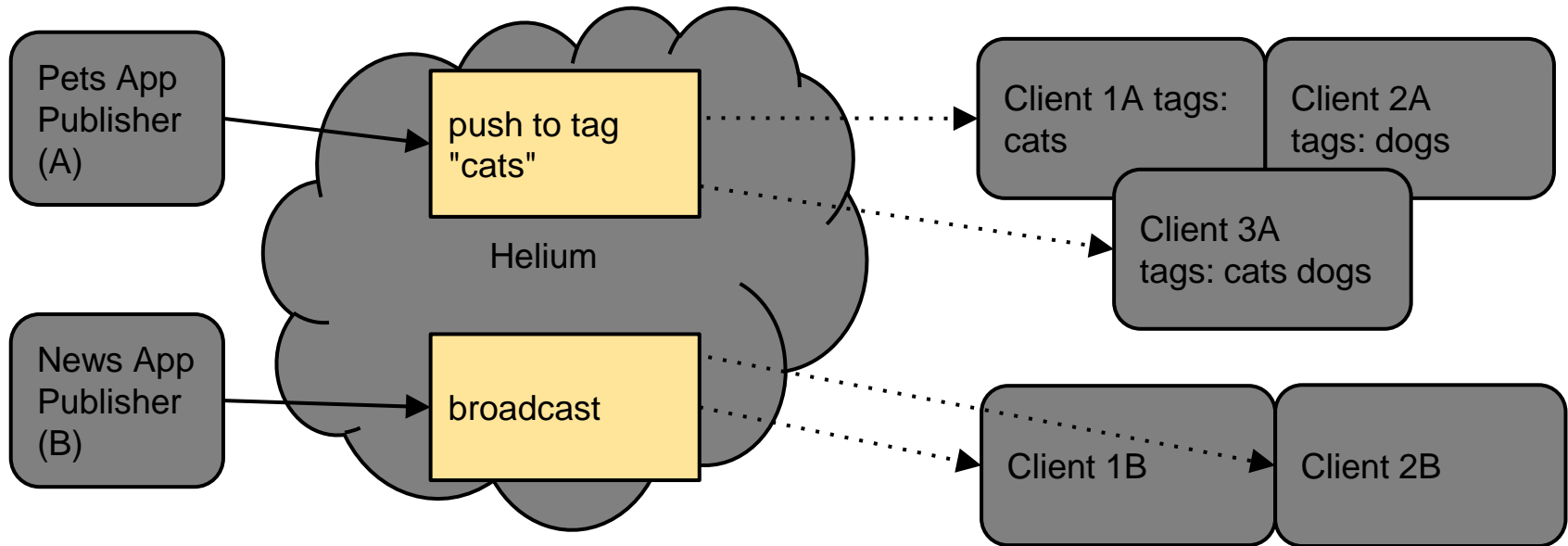


*

Helium Architecture

- Clients generate APIDS (UUIDS) locally and register them with Helium via the Neon registry service.
- Neon directs a client to one of several Helium servers, to which it establishes a persistent TCP connection.
- Publishers use a REST API to send pushes to individual applications by APID, tag, alias or broadcast.

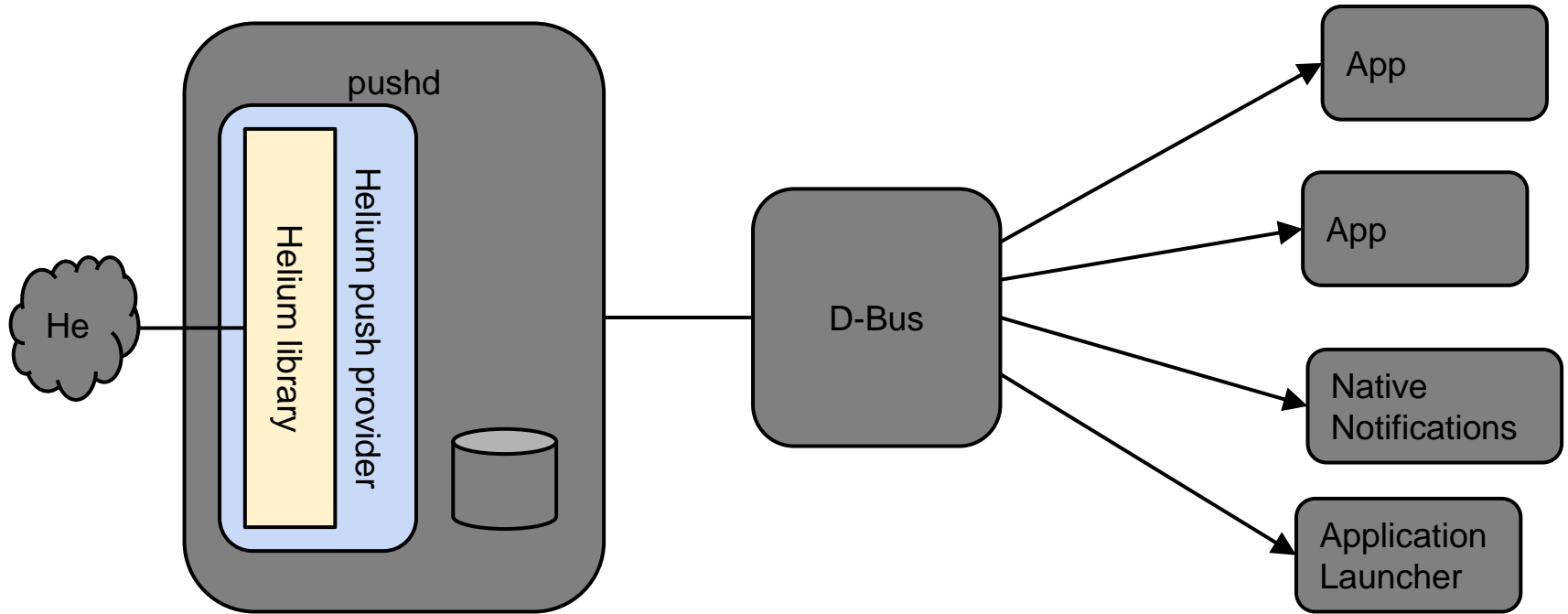
Helium Architecture with Tags and Broadcast



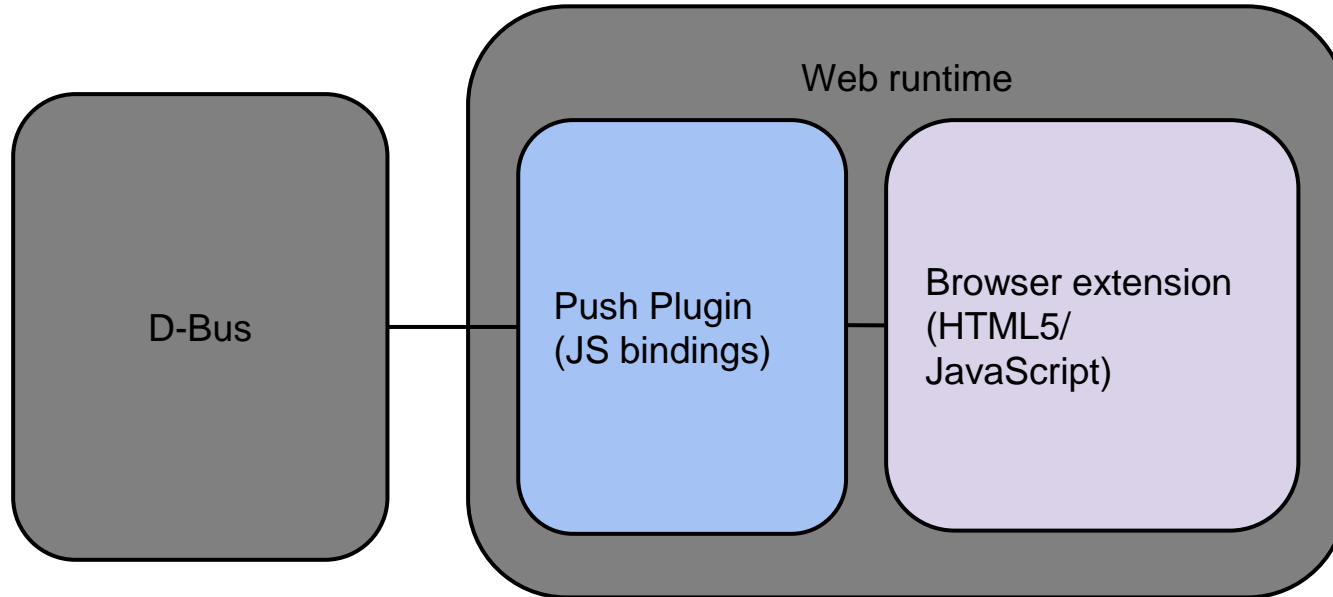
Linux Helium client architecture

- Open push architecture using generic, open-source components (appropriate for an eventual Tizen deployment)
- Improves on library-based offering.
- Pushd centralizes communication on a device or desktop.
- System service provides a persistent client connection if the app is not running.
- Web runtime plugin provides JavaScript bindings for easy development of browser extensions (HTML5/JS).

Helium client architecture on Linux



Push web client architecture on Linux



jQuery.push plugin

- Dramatically simplifies writing HTML5 push applications.
- Provides customization flags as custom HTML5 attributes.
- Provides full JavaScript flexibility for extension and customization.
- Integrates cleanly into jQuery, providing a standard, open-source JavaScript push API from Urban Airship.

jQuery.push plugin (registration)

- Clients provide Helium credentials in HTML meta tags
- Registration is asynchronous and automatic

```
<meta name="push-provider-name" content="com.urbanairship.push.Helium"/>
<meta name="push-extra"
      content="com.urbanairship.push.applicationKey:3s1DjNfsS529dbMSgr7EhA"/>
<meta name="push-extra"
      content="com.urbanairship.push.applicationSecret:Ve2CUR1ISwKN56myyZ57IA"/>
<link rel="stylesheet" href="jquery.push.css"/>
<script src="jquery-1.7.1.min.js"></script>
<script src="jquery.push.js"></script>
```

jQuery.push plugin (displaying notifications)

- Clients identify text elements to be populated using a pre-defined HTML class.
- Display may be customized using custom HTML5 attributes.

```
<span class="push-message"/>  
<span class="push-message" data-push-style="timestamp"/>  
<span class="push-message" data-push-style="timestamp highlight"/>
```

Last Message:

Hello from Urban Airship! 🙌

Last Message with Timestamp:

[10:33:28] Hello from Urban Airship! 🙌

Last Message with Timestamp and Highlight:

[10:33:28] Hello from Urban Airship! 🙌

jQuery.push plugin (accessing push data)

- Individual push-related fields may be used in an application by specifying appropriate HTML classes.

```
<span class="push-id"/>  
<span class="push-message-timestamp"/>  
<span class="push-message" data-push-style="timestamp highlight"/>
```


jQuery.push plugin (JS data access)

- Access data in JavaScript.
- Use custom events for notification.

```
; (function ($) {  
    $('body').live('pushregcomplete', function (e, status, pushId) {  
        if (status) {  
            /* Log push ID to the JS console (or whatever else you need to do with it)  
             * once it's been registered. */  
            console.log('APID is ' + pushId);  
        }  
    });  
})(jQuery);
```

TIZEN™ DEVELOPER CONFERENCE MAY 7-9, 2012

tizen.org

TIZEN™ DEVELOPER
CONFERENCE
MAY 7-9, 2012