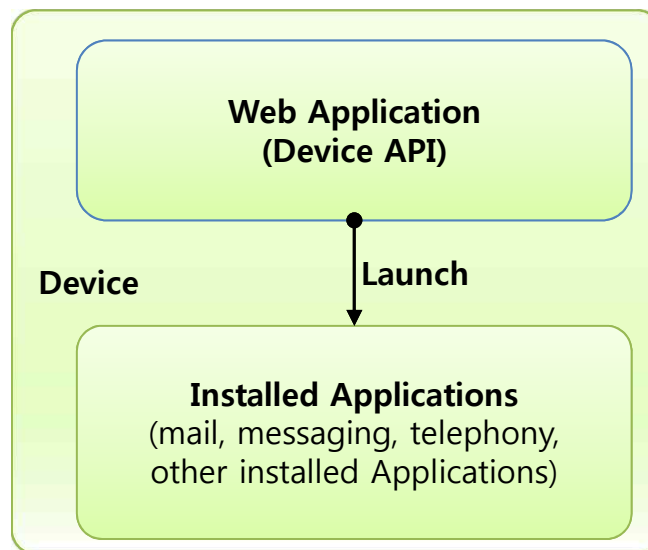


# W3C DAP Application Launcher (app-launcher) API

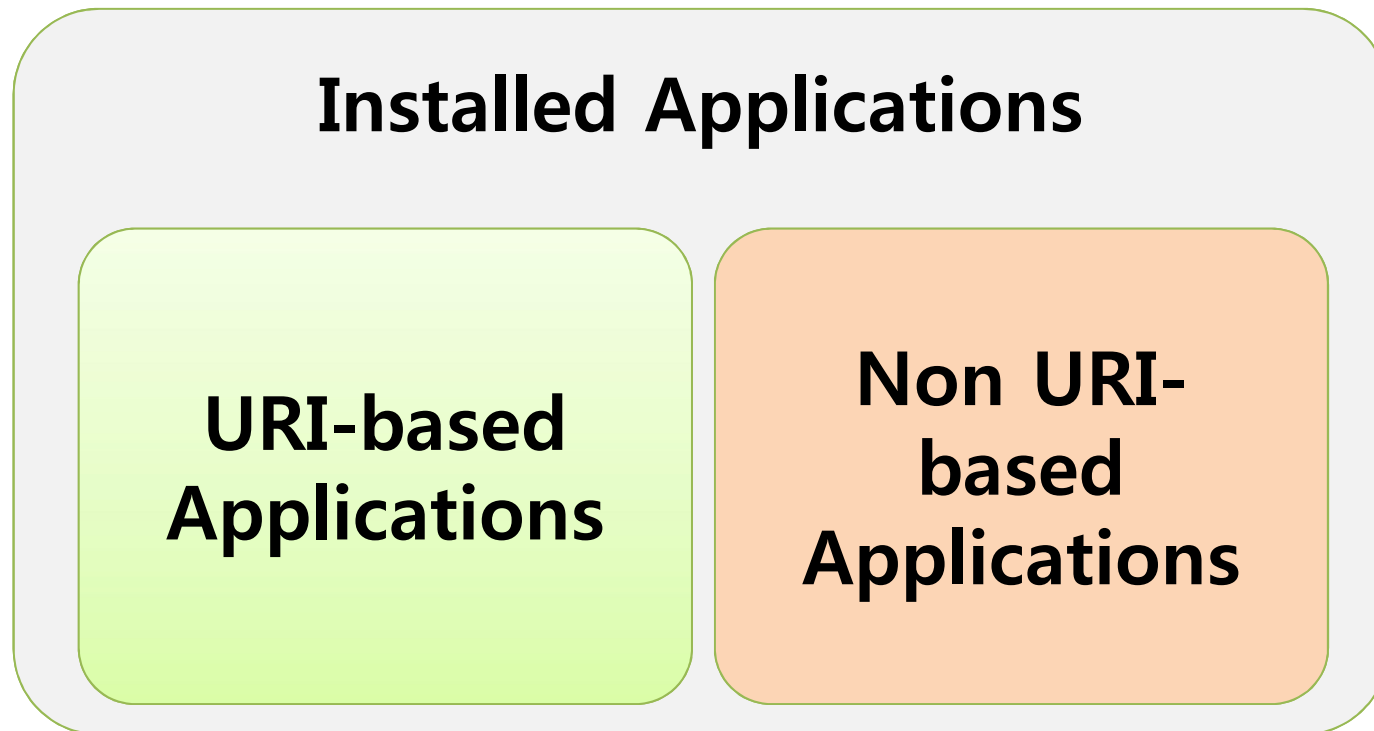
이강찬 (Kangchan Lee), 이원석 (Wonsuk Lee), ETRI  
Bryan Sullivan, AT&T

# Application Launcher API

- Provides interface for accessing to the applications



# The Scope of Applications



# URI-based Applications

- An installed native application can be launched by URI
- Scheme of the standard URI Generic Syntax [RFC3986]
  - http [[RFC2616](#)], https [[RFC2818](#)]
  - tel [[RFC3966](#)]
    - e.g. tel://+82-42-860-6659
  - sms [[RFC5724](#)]
    - e.g. sms://+821020735612?body=hello
  - mailto [[RFC-duerst-mailto-bis-10](#)]
    - e.g. mailto:chan@w3.org
  - File [[RFC1738](#)]
    - e.g. [file:///location/files.xxx](#)

# Other URI Scheme

- Uniform Resource Identifier (URI) Schemes
  - <http://www.iana.org/assignments/uri-schemes.html>
  - [RFC4395] defines an IANA-maintained registry of URI Schemes.
- What is other URI Scheme for Application Launcher API?
  - geo      Geographic Locations      [RFC5870]
  - rtsp     real time streaming protocol      [RFC2326]
  - sip      session initiation protocol      [RFC3261]
  - tv        TV Broadcasts                      [RFC2838]
  - ...

# Non URI-based Applications

- Camera
  - HTML input element can be used to launch the camera directly
    - `<input type="file" accept="image/*;source=camera">`  
<http://lists.w3.org/Archives/Public/public-device-apis/2010Jun/0106.html>
- Any other Applications?
  - Calendar, (background) Voice Recording, Calculator, Game, Weather, and etc ???
  - non-PC device such as EPG Application on TV?

# Open Issues

- Is it need to identify the minimum supported protocols and their native applications?
  - Browser, Telephony, SMS, Mail, Files
- How can set the default application for the URI Scheme? (E.g. 2 or more browser installed)
  - Is it need to support API to change the default application?
- How to execute Non URI-based Applications with standard approach?

# Use Cases

- Use Case 1:
  - Launching the device browser
- Use Case 2:
  - Launching applications by URI (as handler), identity on the device, or “well-known name”
- Use Case 3:
  - Passing parameters to the launched application



# Use Case 1: Launching the device browser

- A widget needs to launch the device browser in order to provide a “full browser experience” for a website.
- Without the ability to explicitly launch the browser (as a “well-known” application or through a URI scheme e.g. `http://` or `https://`), all the widget can do is to load the website content directly, if it can access it.
- This may prevent the intended user experience, e.g. the launching of a browser session with a content storefront link.

# Launching applications by URI (as handler), identity on the device, or “well-known name”

- The basic ability to invoke the registered handler for a standardized URI scheme is the most interoperable way to launch applications.
- But because not all applications are associated with URI schemes, e.g. a media player, text editor, camera, etc, there is the need to launch applications more specifically. Using a “well-known” name for typical types of applications (e.g. browser, camera, text editor) is also an interoperable way to invoke them, since it does not depend upon the local representation of a file or identity of an application.
- Finally, the ability to invoke applications by identity on the device (whatever that is represented as in the device, e.g. a filename) enables apps to invoke arbitrary applications, given that the app is aware of the host device type and the identity of an application that it needs to invoke.

# Use Case 3: Passing parameters to the launched application

- Being able to invoke an application, e.g. a text editor, and indicate the file to be edited will improve the user experience by simplifying the process of integrating with external applications, and enable some use cases that would not otherwise be supportable (e.g. if the file was in a temporary space where the user could not browse to). An arbitrary number of text strings should be supported as parameters to be sent to the application as launched, using whatever convention is supported by the device.

# Requirements

- It SHOULD be possible to listing an installed applications in device.
- It SHOULD be possible to launch an application based on URI scheme.
- It SHOULD be possible to pass arguments to a launched application.
- ...

# Security Consideration

- URI → exposed to public
  - The URI may be used to place malicious or annoying calls.