Related Working Groups

SysApps, NFC and their relevance to wallets and payment solutions

including

Trust & permissions handling

Dave Raggett <dsr@w3.org>

27 October 2014

Web Apps for Wallets & Payment Solutions

- Wallets and payment solutions could be implemented as web applications
  - Locally installed on a device
  - Remotely hosted in the cloud
  - Or a hybrid of the two approaches
    - Seems like the most flexible
- User registers wallet with browser
  - Browser knows how to launch local and remote wallets
  - Browser routes web payment request accordingly
- User registers payment solution with wallet
  - Wallet likewise supports local and remote solutions

Challenges

- Presenting illusion of single wallet across devices
  - Synchronization of installed payment solutions
  - Synchronization of data held by wallet
  - Secure storage and authentication
  - Revocation of device when marked as lost/stolen
- This is however up to the wallet implementors to implement & innovate as they see fit
  - Not covered by W3C standard, right?

System Applications WG

http://www.w3.org/2012/sysapps/

- Chartered in 2012 to work on system APIs for trusted applications
  - Packaged apps installed from app store
  - Hosted apps that are run from web server
    - But can be designed for offline use (e.g. with service worker)
- Work items split into two phases
- Phase 1 focus on execution & security model plus small number of APIs
  - In progress
- Phase 2 expands set of APIs
  - In discussion, but not yet officially started

SysApps Phase 1

- Execution & run-time model has resulted in following specs
  - App Manifest (JSON)
    - App metadata and permssions
    - transferred to WebApps WG in May 2013
  - App Lifecycle
    - Eventing model based upon Service Worker
  - App URI (Last Call WD)
    - Enables use of XHR for packaged apps
- Challenges for dealing with trust & permissions in an interoperable way

SysApps Phase 1

- Other phase 1 work items
  - Task scheduler
    - Basic support for launching app at given time
    - Could be used to implement phone alarm app
      - With addition of calendar and time zone logic
  - Contacts
    - For use in phone address book
  - Messaging
    - Send text and multimedia messages (SMS/MMS)
  - Telephony
    - For use in phone dialler app
  - TCP & UDP sockets
    - Enables apps to implement protocols on top of TCP, UDP and Multicast

SysApps Phase 2

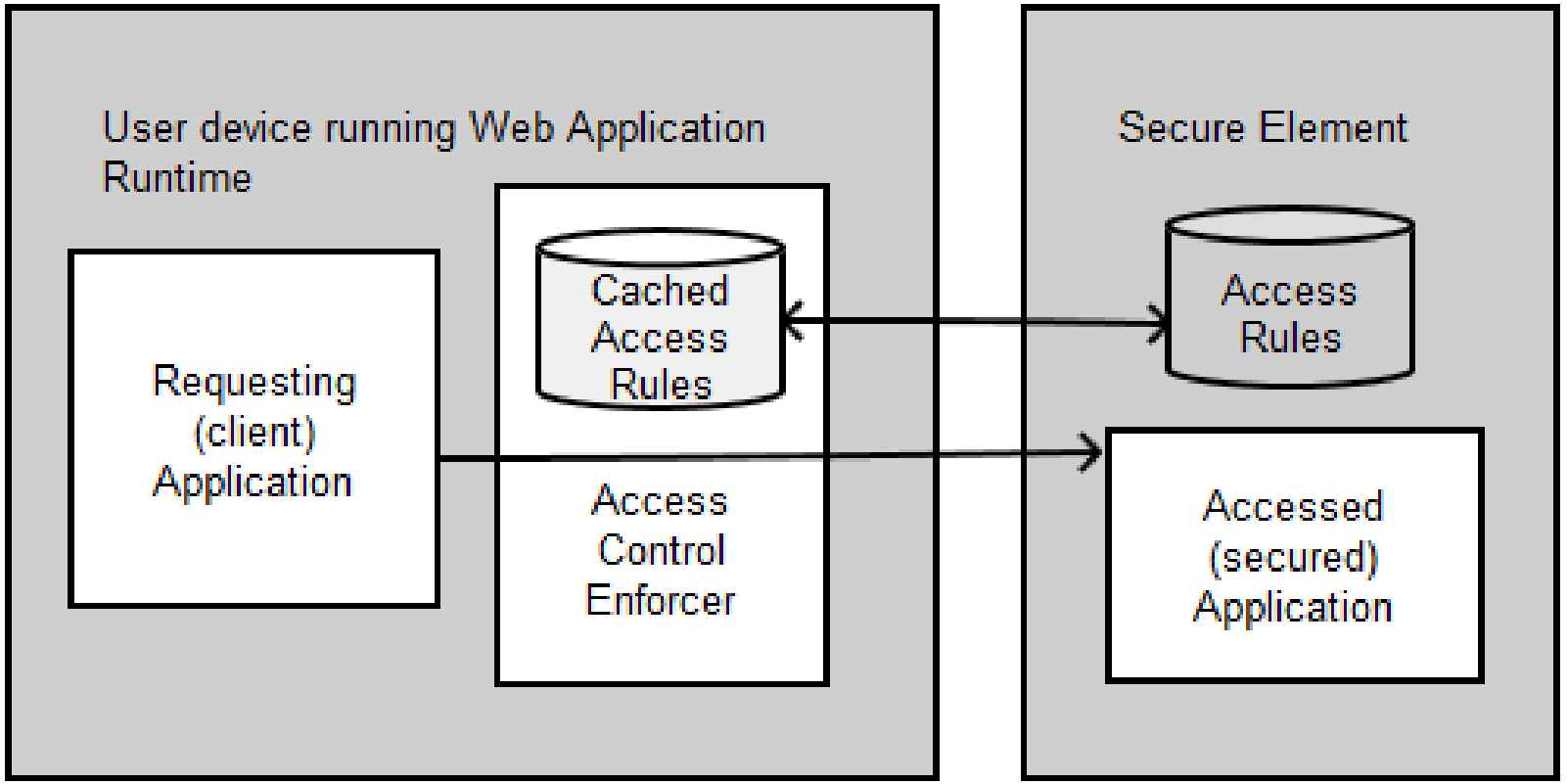
- Chartered work items planned for phase 2
  - Bluetooth API
  - Browser API
  - Calendar API
  - Device Capabilities API
  - Idle API
  - Media Storage API
  - Network Interface API
  - Secure Elements API
  - System Settings API

Secure Elements API

- Intended to enable web apps to invoke code hosted by tamper resistant secure elements
  - Secure elements on UICC*, microSD, embedded
  - Via NFC to secure element on another device
    - Contact-less smart card, phone or other device
    - Firefox OS feature request 884594
- Draft specification submitted by Gemalto
  - http://opoto.github.io/secure-element/
- Use cases
  - Authentication, digital signature, payment, credential provisioning

* Universal Integrated Circuit Card – hosts SIM app on GSM networks

Secure Element API



Designed to layer on top of Global Platform API for APDU exchange

Bluetooth

- Innovations around Bluetooth Low Energy
  - Apple iBeacon, Paypal Beacon, ...
- W3C Bluetooth Community Group
  - http://www.w3.org/community/web-bluetooth/
  - Use cases
  - Draft API
- Using BLE to broadcast URIs to nearby phones
  - Google's Physical Web

NFC WG

http://www.w3.org/2012/nfc/

- Near field communications
  - Very short range for tap based interaction
  - NFC WG chartered in 2012
- NFC hardware is increasingly widely deployed
  - **Apple iOS**, no NFC API as yet
  - **Google Android**, see Chrome NFC API
  - **Microsoft Windows Phone**, see Proximity API
  - **Firefox OS**, see Web API
  - **Tizen**, see W3C NFC API
- W3C NFC API
  - http://www.w3.org/TR/2014/WD-nfc-20140114/

NFC API

- Reading and writing NDEF messages on NFC tags
- Sending and receiving NDEF messages with peers (e.g. smart phone or other device)
- Bluetooth and WiFi pairing (handover)
- Card emulation is **not** yet supported
  - But could be in future specification
- See Secure Element API for APDU access

NFC API

- Possible use cases
  - Tap to play e.g. a peer to peer game
  - Tap to share e.g. coupons, contacts
  - Tap to control another device via handover
  - Tap to connect via WiFi or Bluetooth handover
  - Tap to read NFC tag
  - Tap to write NFC tag

Example

```
var hello = new NDEFRecordText("hello world", "en-US", "UTF-8");

navigator.nfc.ontagfound = function(e) {
    window.console.log('NFC Tag found!');
    var tag = e.tag;
    tag.writeNDEF(new NDEFMessage([hello]));
}

navigator.nfc.startPoll().catch(
  function(e) {
    window.console.error(e);
  });
```

W3C NFC API based upon **Promise** design pattern, see this introduction

NFC – where next?

- NFC now increasingly common with support across all major phone platforms
- Opportunities for use in wallets and payment solutions
- How to move from proprietary APIs to open standards?
  - We can expect increasing pressure from developers for open standards

Trust & Permissions

- Apps need to be trusted before they can be given permission to use certain capabilities
  - Capabilities involving access to personal information
    - e.g. location, contacts, local files, camera, microphone
  - Capabilities that if misused could harm the user
    - e.g. payments, raw sockets
- Common approaches include
  - Ask for user consent when app is installed
    - e.g. Android
  - Ask for user consent when capability is used
    - e.g. iOS
  - Browser silently grants permission to platform apps

Trust & Permissions

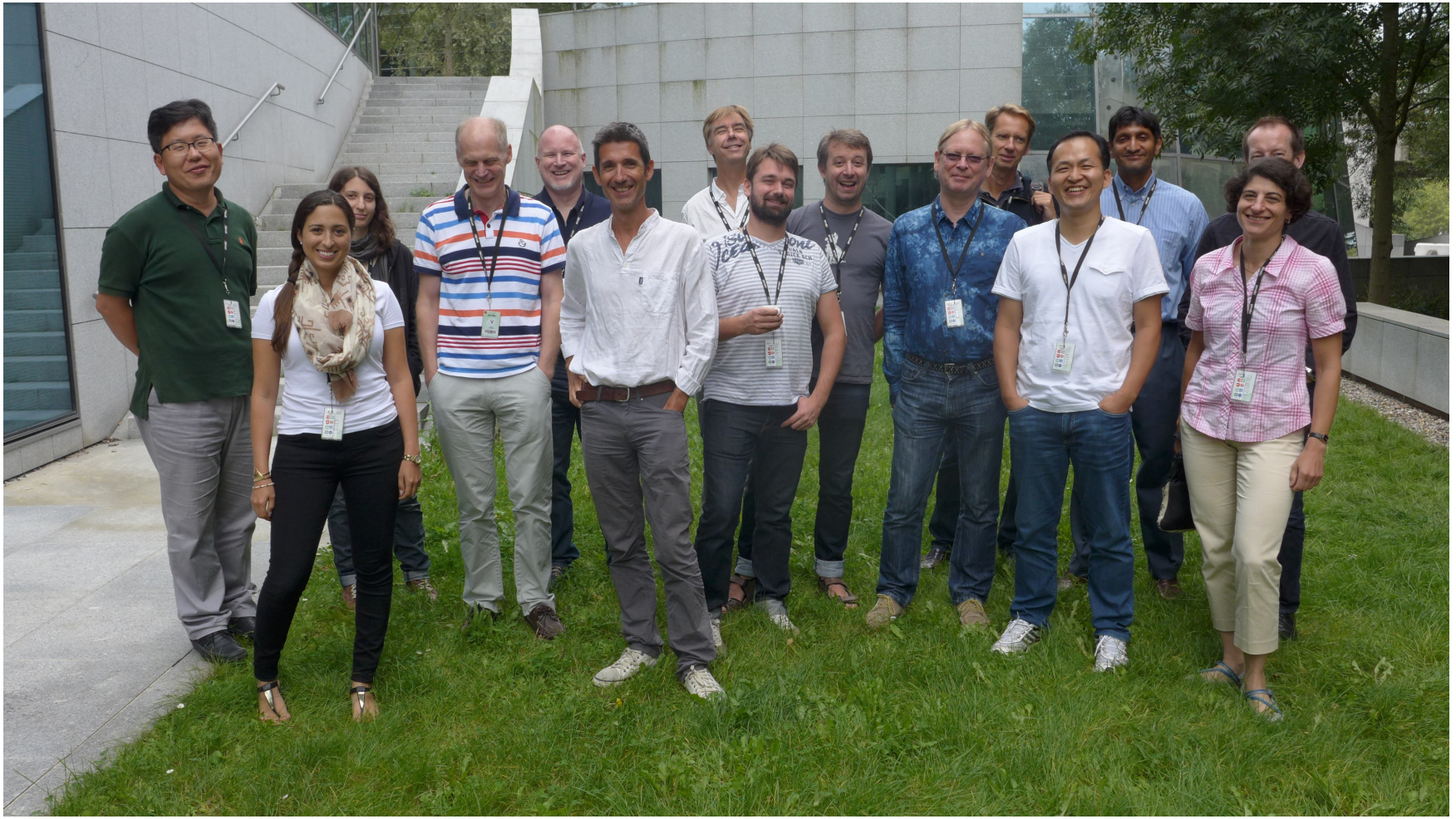
- Native platforms handle this in a proprietary way
  - Apple iOS
  - Google Android
  - Microsoft Windows Phone
- Hybrid platforms subject to host platform
  - Apache Cordova/PhoneGap
- The Open Web Platform (HTML5)
  - Geolocation, Full screen, WebRTC, ...
- Web OS platforms that extend the Open Web Platform in proprietary ways
  - Mozilla Firefox
  - Tizen
  - etc.

Trust & Permissions

Paris, 3-4 September 2014

http://www.w3.org/2014/07/permissions/

Meeting organized by SysApps WG and hosted by Gemalto



Paris meeting

- We shared experiences
  - Native platforms
  - Web platforms
  - Research studies
- And discussed ideas for extending the Open Web Platform
  - But not ruling out packaged apps

- Participants from
  - Apple
  - Ericsson
  - ETRI
  - Gemalto
  - GM
  - Google
  - Intel
  - Microsoft
  - Mozilla
  - Samsung
  - Sony
  - Qualcomm

Trust & Permissions

- Agreement that we need shared standards for the Open Web Platform
  - Building on precedents with existing APIs
- The ship has already sailed for packaged apps
  - Entrenched differences across vendor platforms
    - But opportunity for adopting best practices
- Innovation by browser vendors for detecting misbehaving apps
- Increasing role for endorsements by trusted 3rd parties as a way for users to delegate trust decisions

Next Steps

- General agreement In Paris on launching a W3C Trust and Permissions Community Group
- Focus on best practices and emerging techniques, e.g. trusted UI controls*
- You're welcome to come to the break out session on trust & permissions this Wednesday!

* User-Driven Access Control: Rethinking Permission Granting in Modern Operating Systems, Roesner et al. 2011

Questions?