APIs for Trusted Web Applications

Traditional Web Apps

- Run in Web Browsers
- Browser takes care to protect users from potentially malicious websites
- APIs designed to minimize privacy risks and to protect users from security risks
  - Browser offers picker for files for upload
    - Don't allow website free access to file system!
  - Likewise for access to contacts and device capabilities
- Device API's WG aims to standardise APIs for the browser security context

System Applications

- Contact Picker allows user to pick just what information to share with website
- But what if we wanted to use the HTML5 platform to create an Address Book?
  - Would need a much richer API that can create, delete, update, and list contacts without nagging the user
  - System applications require a much higher level of trust!
  - Needed to create a level playing field with native apps

Trusted Web App Platforms

- New breed of platforms
  - Webinos, Tizen, ChromeOS, FirefoxOS, PhoneGap
  - HTML + JavaScript in place of native app languages
  - Making it much easy to target multiple devices
  - New markets for developers of HTML-based Apps
  - Tapping the huge pool of Web developers!
- Locally installed vs Hosted Apps
  - Locally installed HTML, CSS, JS, images etc.
  - Signed Apps that are hosted in the Cloud
  - Not such a big difference as you might think …
    - Managed apps and local cache

A Vision of Things to Come

- Increasingly common for each of us to possess multiple networked devices
  - Phones, tablets, notebooks, TV's, and soon cars*
    - And don't forget the Internet of Things ...
  - How will we as users manage all of these?
- We need
  - a seamless means for individuals to manage their applications and services across all of the devices they possess
  - to enable a market for a new breed of applications and services spanning multiple devices, and multiple users
- The solution – Personal Zones

* W3C Web & Automotive workshop – 14-15 November 2012, Rome

Personal Zones

- For end-users a simple means to manage all of your devices, applications and services
  - Implies the need for synchronization and strong device security
- For developers, an abstraction layer spanning devices and exposing a consistent set of APIs across them
  - Including support for applications spanning devices and users
- A market of applications and services
  - Applications that run on web run-time with HTML-based UI
  - Services that run on servers embedded in devices or in the Cloud
    - Can be thought of as Personal Web Agents acting on your behalf
    - Implemented in JavaScript, e.g. as node.js modules
    - Common set of APIs with web run-time
- Research focus of webinos project

W3C SysApps WG Charter

- Spin off from Device APIs working group
- Chartered Sep 2012 through Oct 2014
  - Chaired by Adam Barth, Google and Wonsuk Lee, Samsung
- Initial focus on execution and security models
- Getting our feet wet (phase 1)
  - Alarm, contacts, messaging, telephony, raw sockets
- Applying what we've learned (phase 2)
  - Bluetooth, browser, calendar, device caps, idle, media storage, network interface, secure elements, system settings
- Great opportunity for formal contributions and for sharing use cases, insights and implementation experience

W3C NFC WG Charter

- Originally part of SysApps, but split off on W3C Member feedback
- Chartered Oct 2012 through Nov 2014
  - Chaired by Jacques Bourhis, Intel
- Scope
  - NDEF read, write operations
  - LLCP message channel
  - NFC Card emulation including APDU support
- Another opportunity for sharing experience

Where are we now?

- Sign up period as people join SysApps & NFC
  - http://www.w3.org/2012/sysapps/
  - http://www.w3.org/2012/nfc/
- Initial contributions of draft specifications
  - As well as use cases and requirements
- Volunteers for editors of each specification
- Establishing ways of working
  - Github for specifications
  - W3C wiki
  - Email, teleconferences and face to face meetings

Security Model?

- Some suggestions from John Lyle, Oxford University
  - Install before execute
  - Signed manifest
  - Restrictions on external sources and use of secure connections
  - Least privilege access control like native app platforms
  - Interoperable access control through policies or capability tokens
  - Hooks for monitoring system application activity
  - Restrictions on inline JavaScript and dangerous functions such as eval.
  - Isolation expectations, e.g. sandboxed storage & inter app communication
  - Application lifecycle – update, uninstall, revocation of privileges
  - Application provenance, e.g. app store certification
  - User interface guarantees, e.g. to limit clickjacking
  - Fingerprinting, linkability and privacy considerations

Execution model?

- Foreground/background
- Suspend/resume
- Adaptation to changes in the context of use
- Notifications
- Preference settings
- Apps vs Services
- Safety & Legal Liability – situational awareness and mitigation of driver distraction (for cars)

Discussion Topics?