W3C Web and Automotive Workshop

Is it time to power Infotainment and Car Portal Applications with HTML5 and Device APIs?

Vodafone Group R&D
Rome, November 2012

Diana Cheng – R&D, Web Expert

diana.cheng@vodafone.com

daianacheng

W3C DAP and SysApps member

Markus Münkler – Head of R&D Automotive

markus.muenkler@vodafone.com

http://de.linkedin.com/pub/markus-muenkler/0/247/

Contents

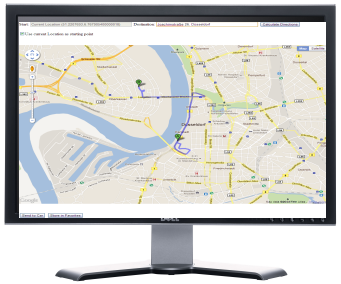
The Automotive Technology Ecosystem

Auto

- Fragmented
- Many proprietary / in-house made platforms and devices
- Others start to emerge using: Android, Windows, Blackberry Tablet OS, etc.

The Automotive End-to-End User Experience

Auto



Driving directions

Music Collection Sync

Walking directions

PC

IVI Systems

Mobile

The customer won't care about the platform. He'll just want it to work!

Driving directions

Tablets

The Web Technology Ecosystem

(*) HTML5

- Cross-platform
- Open technologies: HTML, JavaScript (incl. available device APIs) & CSS
- One skill set, one project
- Develop once, adapt(!), run on different devices
- Immediate push of updates and features to hosted apps

(!) But support, in particular in Mobile Browsers, is still limited

* Used as an umbrella term. Refers to a number of “new” web technologies such as HTML5, CSS3, browser- based (Device) APIs, etc.

Web – So what's new?

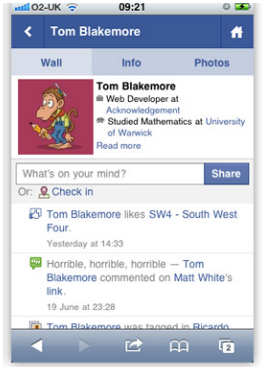
HTML5

- New web platforms are pushing the boundaries of web technologies (Firefox OS, Chrome OS, Tizen)
- They are specifiying new APIs
- The gap with native apps is closing

Are HTML5 and Browser-based APIs ready?

HTML5

- HTML5 is often believed to be „less powerful than Native“.
- But that highly depends on the nature of the application, e.g. Financial Times Mobile Web app, Facebook Mobile Web app



So what are the requirements for Automotive?

Do we really need all the power of Native?

What needs to improve in HTML5?

Automotive Requirements: 2 Main Fronts

Auto

In-vehicle Infotainment

- Session Handover
- Speech Recognition
- Text-to-Speech
- Navigation
- Installable Apps
- Offline Support

Mobile Car Portals

- Map Visualization
- Touch Interaction
- Real-time notifications to devices
- Camera Access

Deep Dive

HTML5

Web features that can enable these requirements

IVI Applications – Session Handover (1/2)

Auto

- Continue in-car session when leaving the vehicle:
  - continue listening to the same music/radio track
  - continue navigation with walking directions

IVI Applications – Session Handover (2/2)

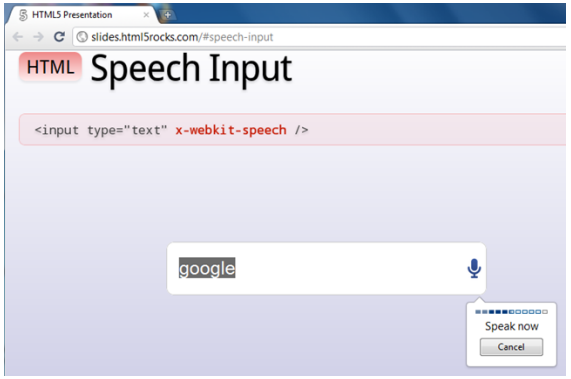
- Native Push Notifications:
  - Android C2DM/GCM, APNS for iOS, Blackberry Push, etc.
  - Have their restrictions: Message Quota, No adds allowed (iOS)

- Web Push: HTML5
  - Server-Sent Events: 1-way Push, Standard DOM events, no specific implementation needed
  - WebSockets: bi-directional channel, timeout when page is idle, requires active start of the application by the user
  - Limited mobile browser support, although libraries and polyfills can help

IVI Applications – Speech Recognition

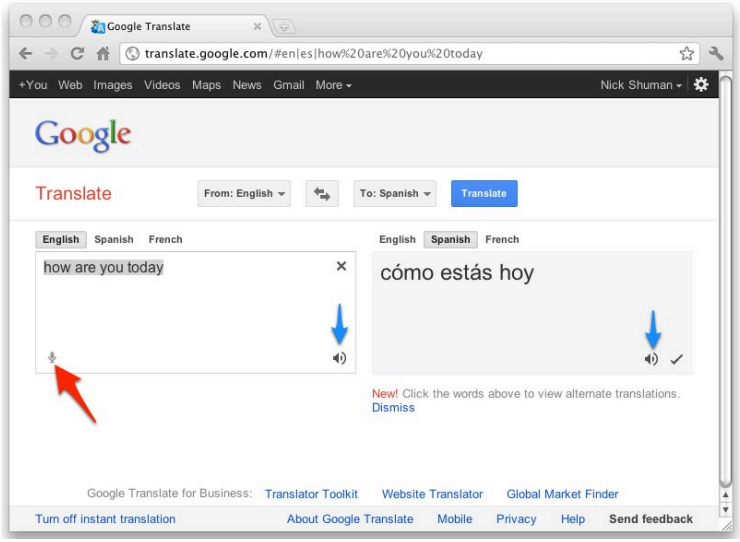
- Voice is the main interaction interface: provides minimal distraction
- Embedded automotive solutions offer „good“ capabilities
- HTML5 Speech Input: HTML5

  - Spec out of Speech XG: agnostic of underlying engine. New W3C WG might be chartered soon?
  - But hardly any implementation (WebKit) and dependent on backend services
  - Breaks the offline requirement! ☹ Cellular-based connectivity in the car can‘t be guaranteed.



IVI Applications – Text-to-Speech

- Reading of incoming SMS, incoming real-time updates, driving directions
- A few browser-based APIs:
  - e.g. Google Translate API
  - Dependent on a backend service too
  - A few proposals over time at the W3C but no spec under standardization



IVI Applications – Navigation (1/2)

- Sophisticated dedicated devices
- HTML5 covers many of the requirements: HTML5
  - GPS coords via the Geolocation API, widely implemented
  - Web Workers for route calculations, requires more mobile support (*)
  - Offline?

(*) http://caniuse.com/#feat=webworkers

IVI Applications – Navigation (2/2)

HTML5

- Offline access to previously loaded maps and other assets
  - AppCache: but „widely documented to be a bit rubbish“ (*)
  - WebStorage: Small quota and storage of **binary assets** is not easy, has performance issues due to its synchronous nature

- Other required storage APIs are still making slow progress:
  - In-browser Database: IndexedDB, but no mobile implementations
  - FileSystem & File Writer API: Disagreement on specs and hardly any implementations

(*) http://labs.ft.com/2012/08/basic-offline-html5-web-app/
http://www.alistapart.com/articles/application-cache-is-a-douchebag/

IVI Applications – Installable Applications

HTML5

- We have W3C Widgets of course:
  - But currently no mechanism to automatically update them
  - Lose this advantage in comparison to hosted apps
  - Not very popular: App packaging is fragmented. This needs to be addressed.

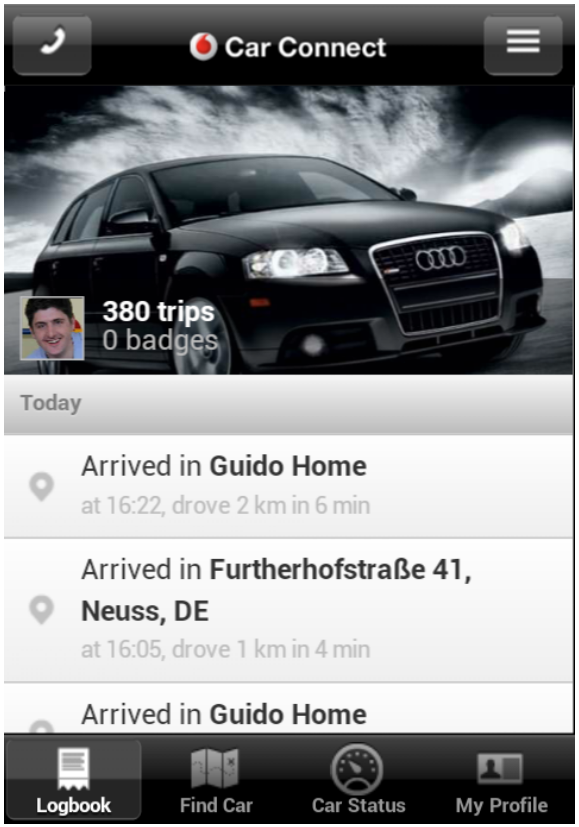
W3C

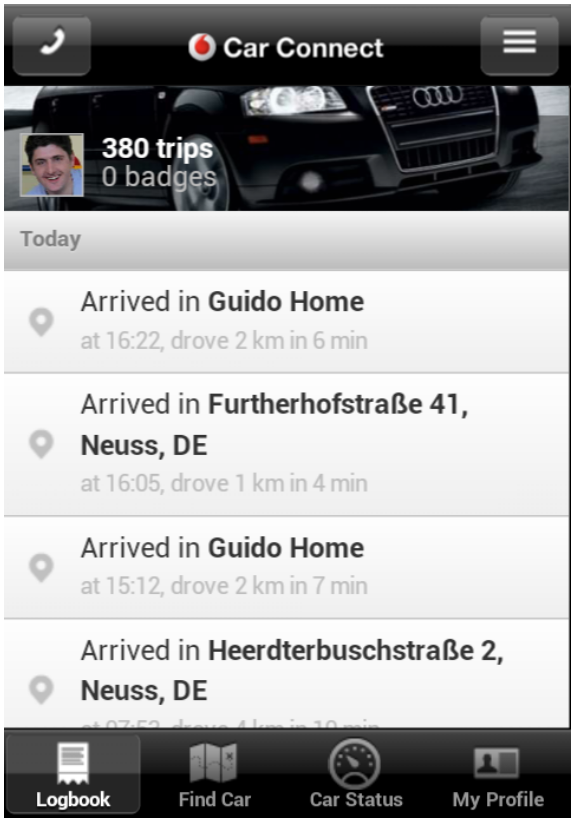
Widget Interface

W3C Candidate Recommendation

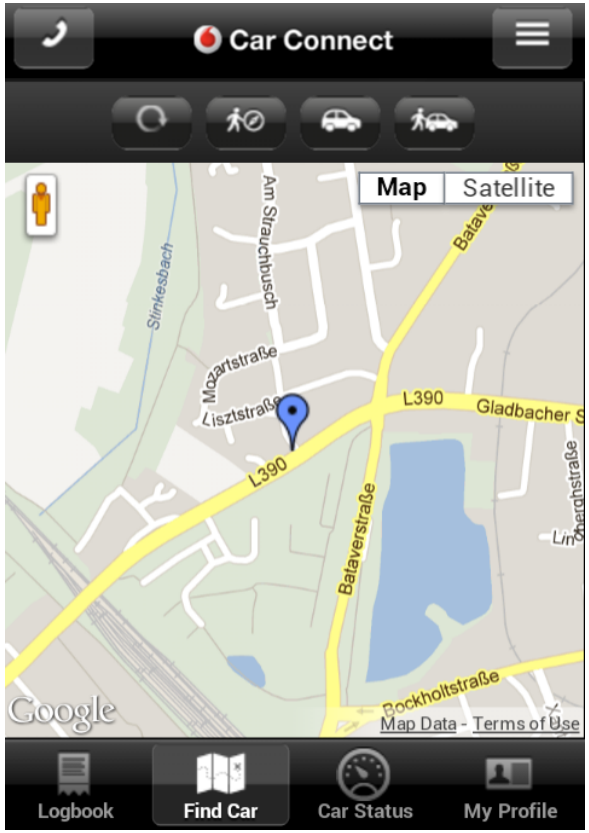
- PhoneGap as an alternative:
  - Native builds
  - Useful when Device APIs are missing in the targeted platform
  - Updated through the native app stores

Car Portals

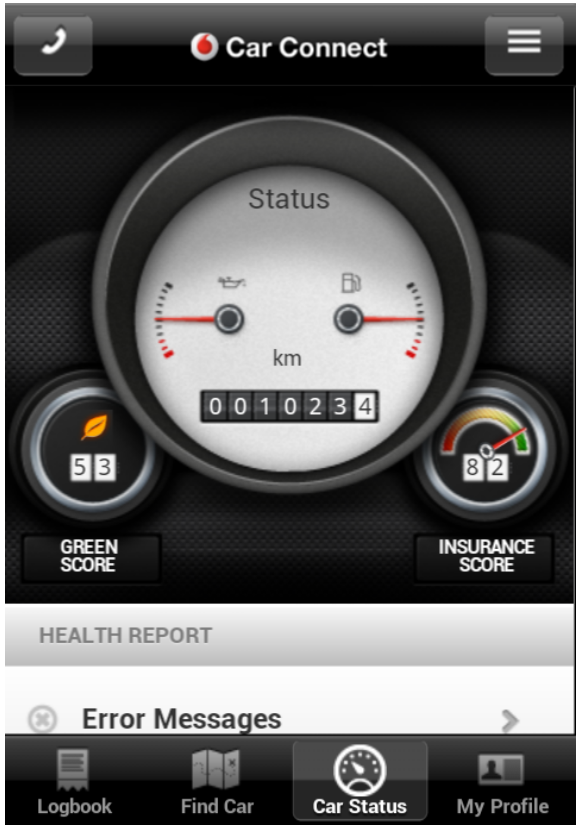

Car Connect
380 trips
0 badges
Today
Arrived in Guido Home
at 16:22, drove 2 km in 6 min
Arrived in Furtherhofstraße 41, Neuss, DE
at 16:05, drove 1 km in 4 min
Arrived in Guido Home
Logbook
Find Car
Car Status
My Profile

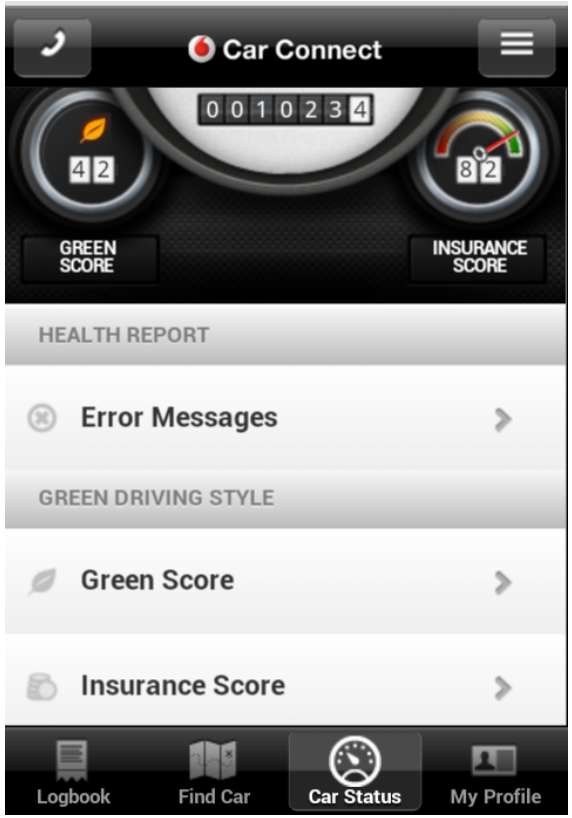

Car Connect
380 trips
0 badges
Today
Arrived in Guido Home
at 16:22, drove 2 km in 6 min
Arrived in Furtherhofstraße 41, Neuss, DE
at 16:05, drove 1 km in 4 min
Arrived in Guido Home
at 15:12, drove 2 km in 7 min
Arrived in Heerdterbuschstraße 2, Neuss, DE
Logbook
Find Car
Car Status
My Profile

Car Portals

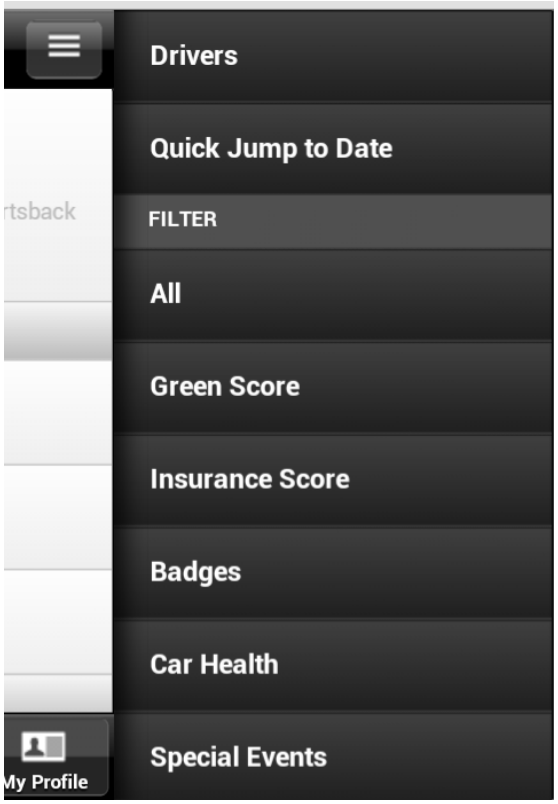

Car Connect
Map
Satellite
Am Strauchbusch
Stinkesbach
Mozartstraße
Lisztstraße
L390
Gladbacher S
Bataverstraße
Bockholtstraße
Google
Map Data - Terms of Use
Logbook
Find Car
Car Status
My Profile

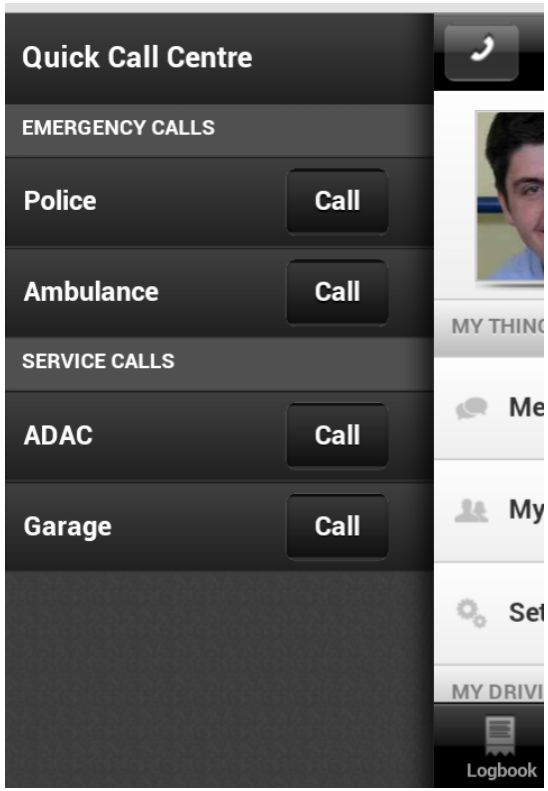
Car Portals


Car Connect
Status
km
0 0 1 0 2 3 4
5 3
8 2
GREEN SCORE
INSURANCE SCORE
HEALTH REPORT
Error Messages
Logbook
Find Car
Car Status
My Profile


Car Connect
0 0 1 0 2 3 4
4 2
8 2
GREEN SCORE
INSURANCE SCORE
HEALTH REPORT
Error Messages
GREEN DRIVING STYLE
Green Score
Insurance Score
Logbook
Find Car
Car Status
My Profile

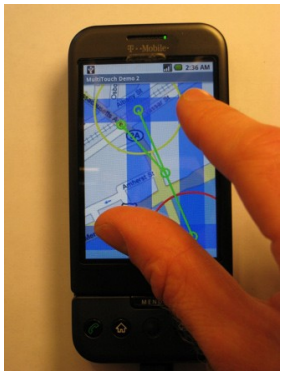
Car Portals


Drivers
Quick Jump to Date
FILTER
All
Green Score
Insurance Score
Badges
Car Health
Special Events
My Profile


Quick Call Centre
EMERGENCY CALLS
Police
Call
Ambulance
Call
SERVICE CALLS
ADAC
Call
Garage
Call
Logbook

Car Portals – Map Visualization

- Very rich Web UIs are possible
- Leverage of existing Web-based Map APIs: Google Maps, Nokia Maps, OpenLayers, etc.
- Touch is a requirement for interacting with Maps on mobile:
  - Several „modern“ mobile browser such as Android 2.3 offer poor implementation of touch events.
  - No double tap and pinch-to-zoom on maps feels awkward
  - Scrolling issue: ***overflow: auto*** problem (implementation issue)



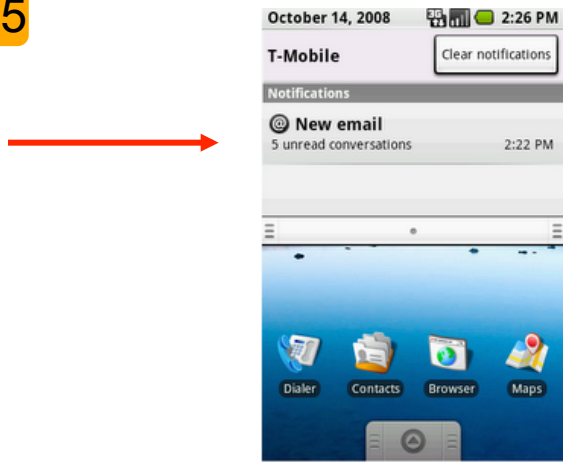
Car Portals – Real-time Notifications

- Status bar, native-like notifications outside of the browser context are needed:

  - Capture the user‘s attention: e.g. emergency alerts, etc.

  - Only available in Native stacks currently ☹

  - Web Notifications: spec to alert outside the „web page“; work in progress. Only implementation is Chrome Desktop HTML5



- Vibration API:

  - To couple with Real-time notifications

  - Worked on at W3C DAP, first implementation for WebKit

Key Take-aways (1/3)

- Web technologies *can* become a good enabler for end customers (IVI) and enterprises in the Automotive context.
- With its advantages in the development process, we can be fast to market and reachable by a wide range of customers due to its interoperability.
- Improving the aforementioned items would make the Open Web Platform a real contender for Automotive Systems.

Key Take-aways (2/3)

So….

Key Take-aways (3/3)

HTML5

What is *still* needed from the Web Platform:

- An agreement of which spec to use for storing large amounts of/binary data in web applications (IndexedDB? WebSQL on mobile? FileSystem API?)
- To finalize and support Web Notifications
- Broader support for Web Sockets and Server-Sent events in mobile browsers
- A mechanism for automatic update of Widgets so they can benefit from the immediate deployment. Or better:
    - Convergence of implemented Web App packaging approaches?
- Standardization of a spec for in-browser text-to-speech
- Support for advanced touch interactions
- Support for offline speech recognition in the browser (not just provided by backend services)

Questions?

Backup Slides

Problems with AppCache

- If you list one 100 files in your manifest, it will download all 100 of those files as quickly as it can – slowing down the user's interactions with the app –app will seem less responsive whilst the browser downloads all that.
- If you change any one of those resources, even just a one line change in one CSS file the browser will then re-download every single resource in the manifest. It can't do a single file update. It can only replace the entire contents in the cache.
- If any of the files fails to download for any reason it will get rid of all the files it's successfully downloaded and revert to the previous version it had in cache.

Problems with localStorage

- localStorage is synchronous in nature; when it loads it can block the main document from rendering
- localStorage does file I/O, i.e. it writes to hard drive, which can take long
- On a developer machine these issues can be hard to detect; different story for a mobile user
- In order to appear snappy, browsers load all data into memory on the first request – which could mean a lot of memory use if lots of tabs do it
- localStorage is persistent. If you don’t use a service or never visit a web site again, the data is still there