



[WEB AND AUTOMOTIVE] Shift into High Gear on the Web

W3C WORKSHOP
14-15 NOVEMBER 2012, ROME, ITALY
HOSTED BY INTEL



Gadget Car

Marcin Hanlik, Shinjiro Urata
2012.11.14

Advanced Software Solutions for
Mobility • Internet Appliances • Services • Networks

Agenda

1. Trends and Scenarios

a. Connectivity

b. Remote access

2. HaaS – the Apps

a. nBox

b. REST API + JS Lib vs. JS APIs

c. Integration

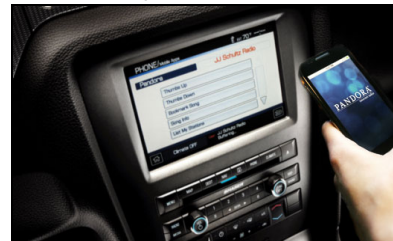
3. Demonstration

Trends of IVI platform evolution

- Install the application on your smartphone, IVI (In-Vehicle Infotainment) communicates with the app on your smartphone



Ford (Sync AppLink)



Toyota (Entune)



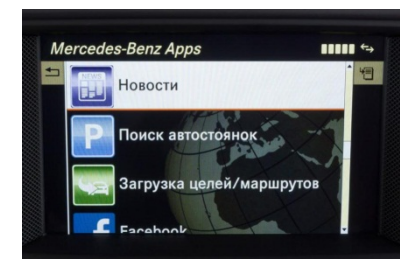
- Use Cell Phones / Smartphones, connected by Bluetooth



QNX Car



Daimler (COMAND Online)



The standards

- W3C
- CEA
- UPnP Forum
- DLNA

- Vehicle API as part of Device APIs
 - Mobile: OMTP BONDI, WAC, GSMA, Webinos
 - W3C: DAP, SysApps

 - Ca. 6 attempts to standardize Calendar API

Connectivity: UPnP

- **What is UPnP?**

- Protocol for connecting and coordinating mobile phones, AV equipment, PCs based on current internet standard technologies.
- Aims to eliminate complicated configuration and to realize plug and play usability.
- Available over networks such as Ethernet, wireless LAN, IEEE1394, etc.

Terminal Mode Application		
TmServerDevice:1 Device	VNC	DAP
TmClientDevice:1 Device		
<i>UPnP Device Architecture</i>		
TCP/UDP		
Internet Protocol		
Wifi, Bluetooth, ZigBee		

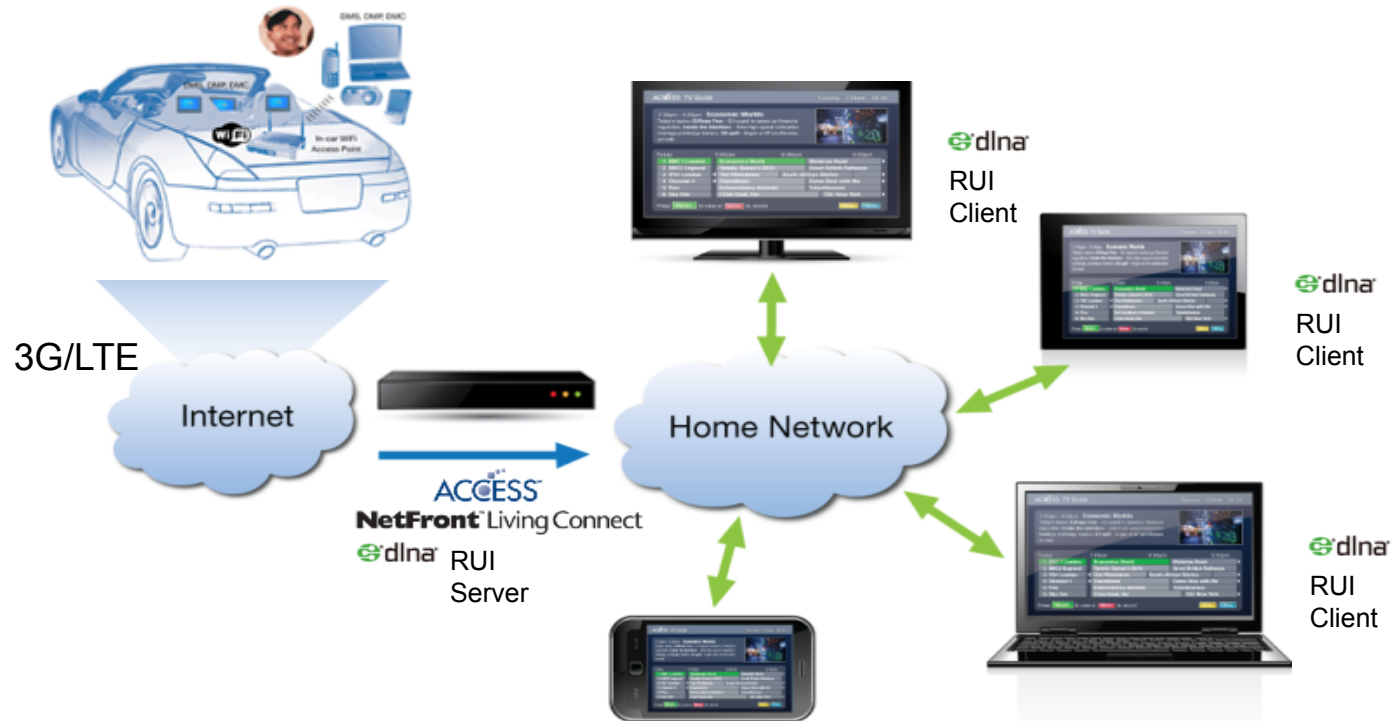

UPnP Library



Remote UI - Remote access

- **What is Remote UI ?**

- A new function to provide control UI from a Digital Media Server to client devices, defined in DLNA Guideline August 2009.
- Compliant to UPnP RUI Server/Client standard and UI content is based on CE-HTML.
- Possible to realize uniform usability by providing RUI on both Server and Client devices.



HTML5 as a platform



HaaP – why?

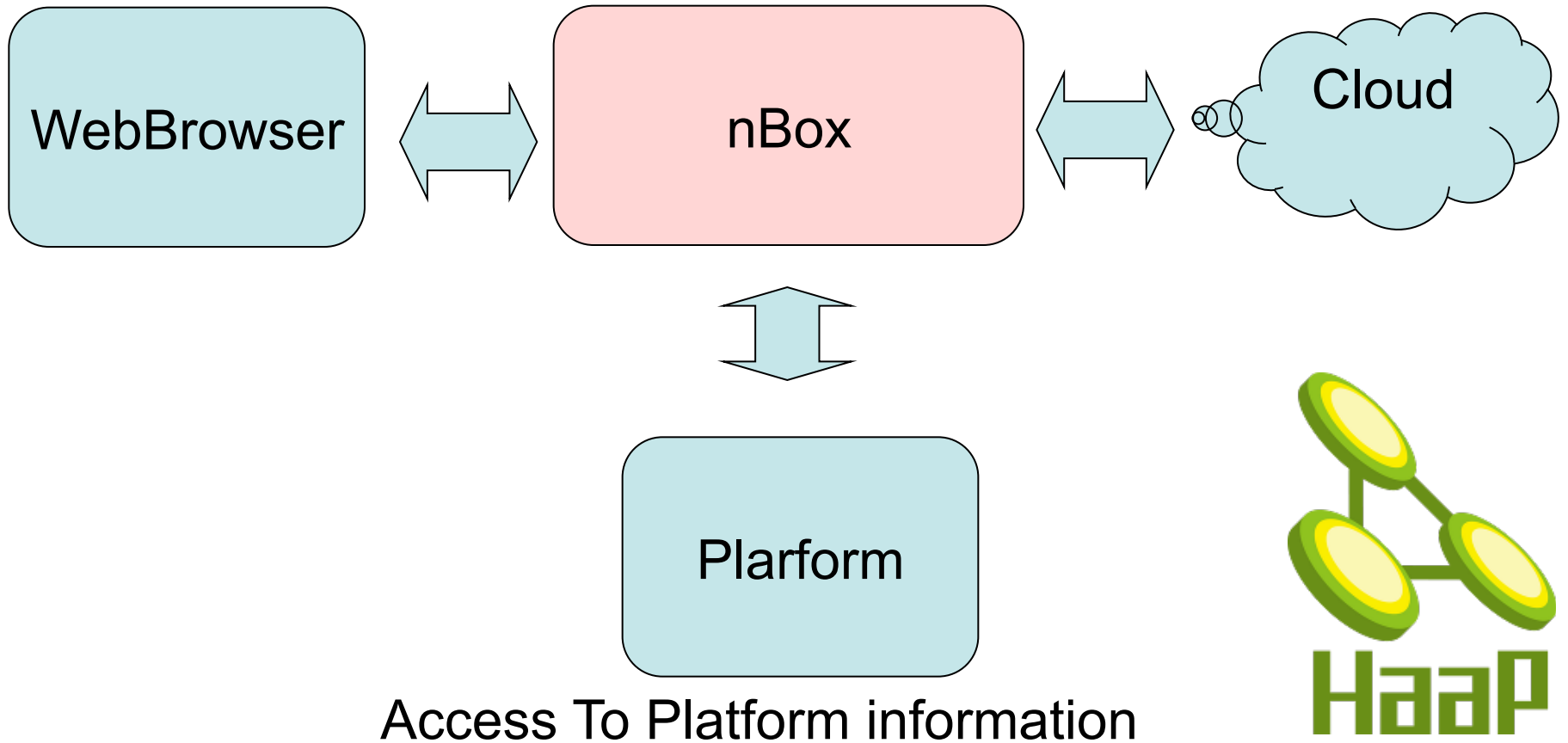
- Cross-browser web application hosting system
 - to create web applications by using HTML5, CSS and JavaScript
 - to share web applications between multiple devices
- Programmable embedded module
 - to access to platform information
 - to make use of cloud services



nBox – the framework

Provide User Interface

Use Cloud Service



nBox goals

- *Cross-Browser*
 - *the web browser needs to support XHR or WebSocket*
 - *rendering differences between web browsers can be fixed by nBox which can generate web-browser-specific HTML documents*
- *Extensible without modifications to web browsers*
- *Extensible without modifications to native code*
 - *but limited by platform capability*
 - *it could be filled in gaps between platform capabilities by using cloud services...*

REST APIs vs. JavaScript APIs

1. Application management

1. [Install] *POST /apps/:appsID*
2. [uninstall] *DEL /apps/:appsID*
3. [get install apps] *GET /apps, GET /apps/:userID*

2. User management

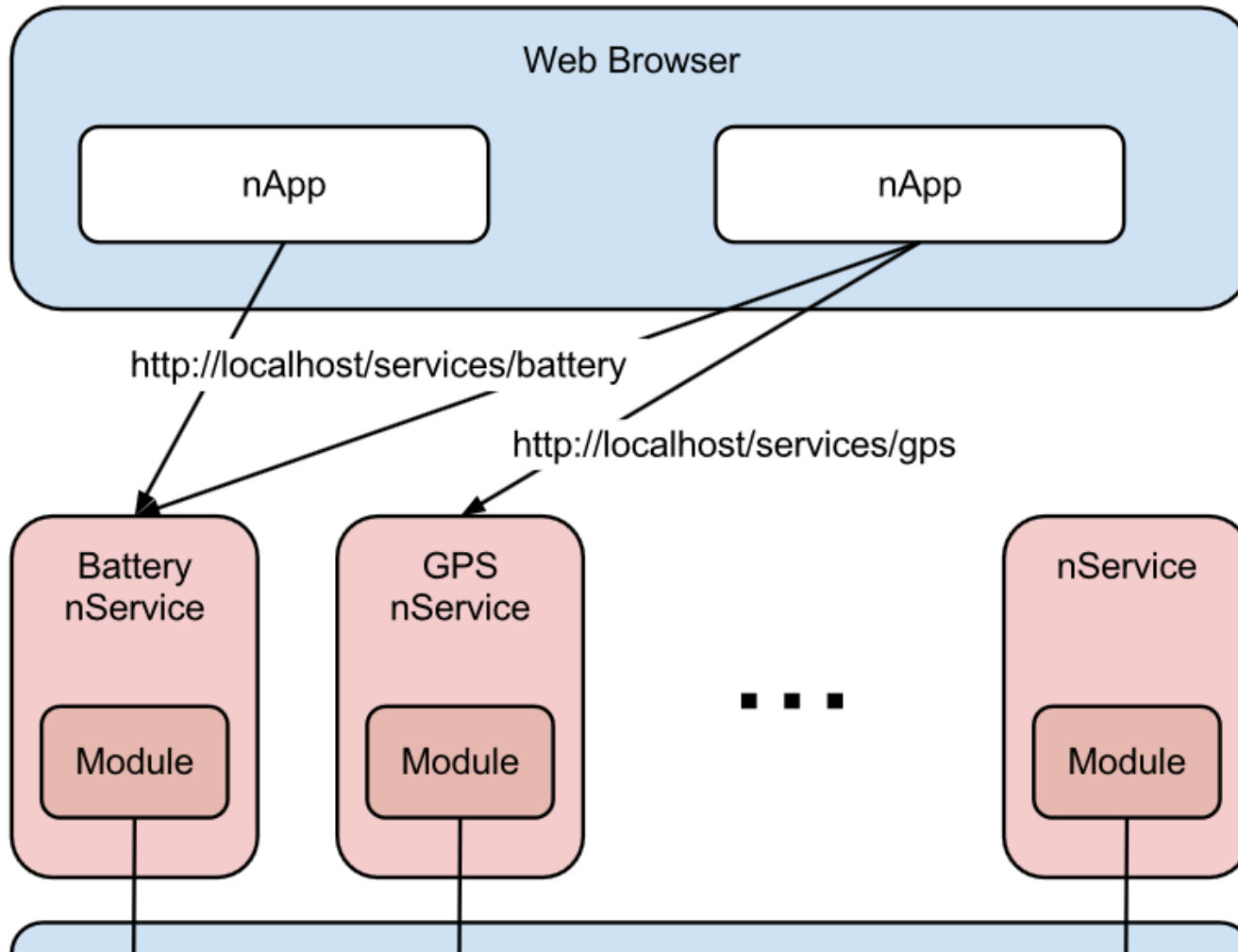
1. [create] *POST /users*
2. [login] *POST /users/:userID/login*
3. [logout] *POST /users/:userID/logout*

3. Sample store

1. [install] *GET /store/:pkgID*

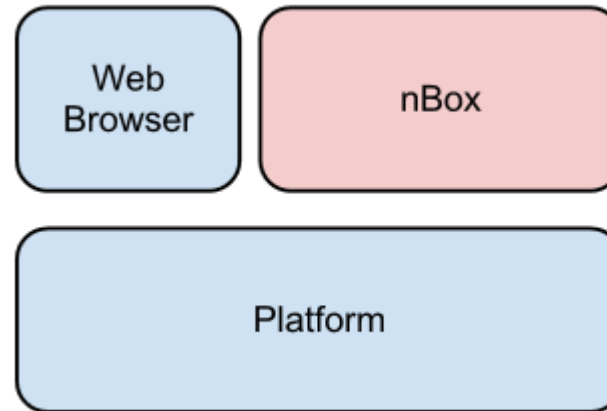


Applications and Services

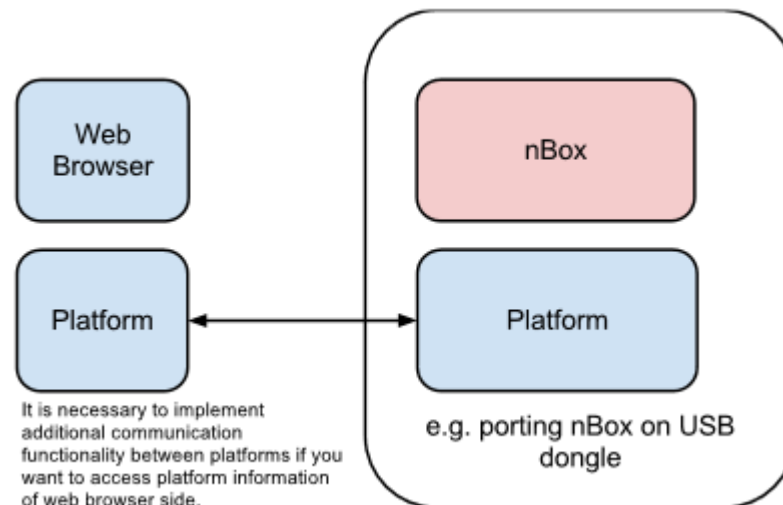


Integrations models

- **Embedded**

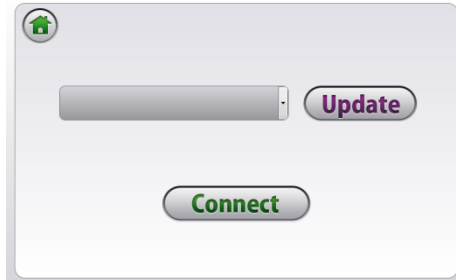


- **Separated**

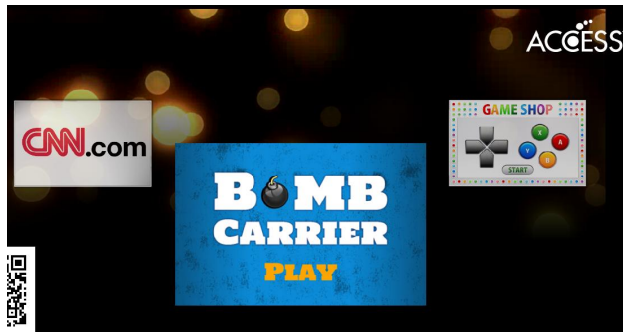


Demonstration

1. Connect IVI and Smartphone



2. Operate IVI menu from smartphone



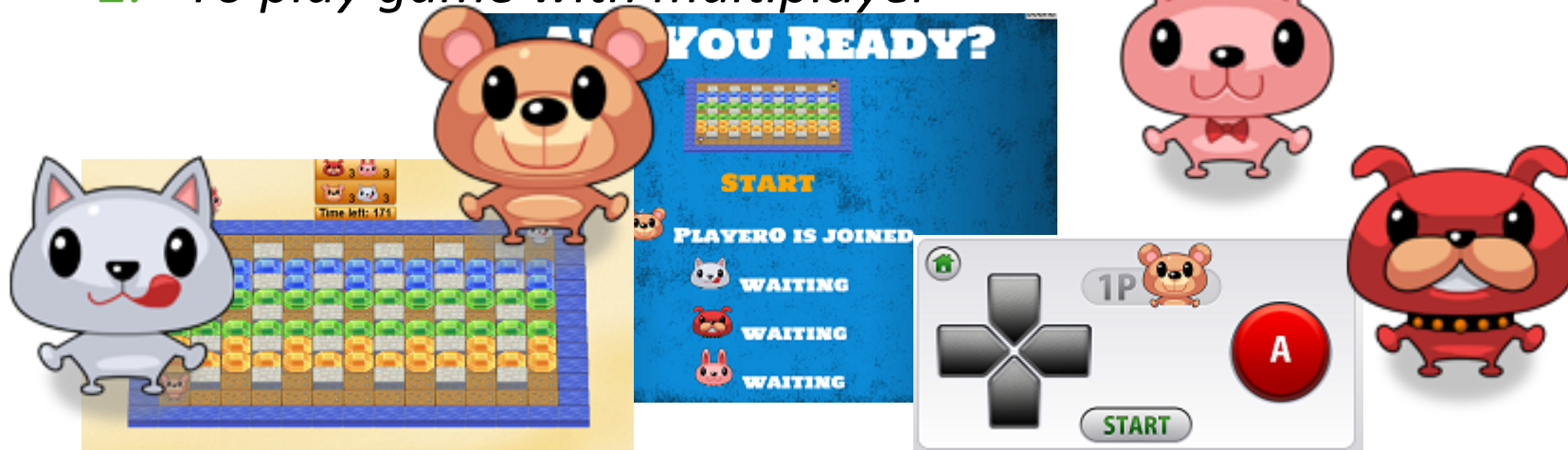
3. Install an application

Demonstration

1. To play HTML5 games



2. To play game with multiplayer



Thank you!



HTML5 for every device.