

HTML5 standardization in auto

Andy Gryc, Automotive Product Marketing Manager



QNX SOFTWARE SYSTEMS

Three topics

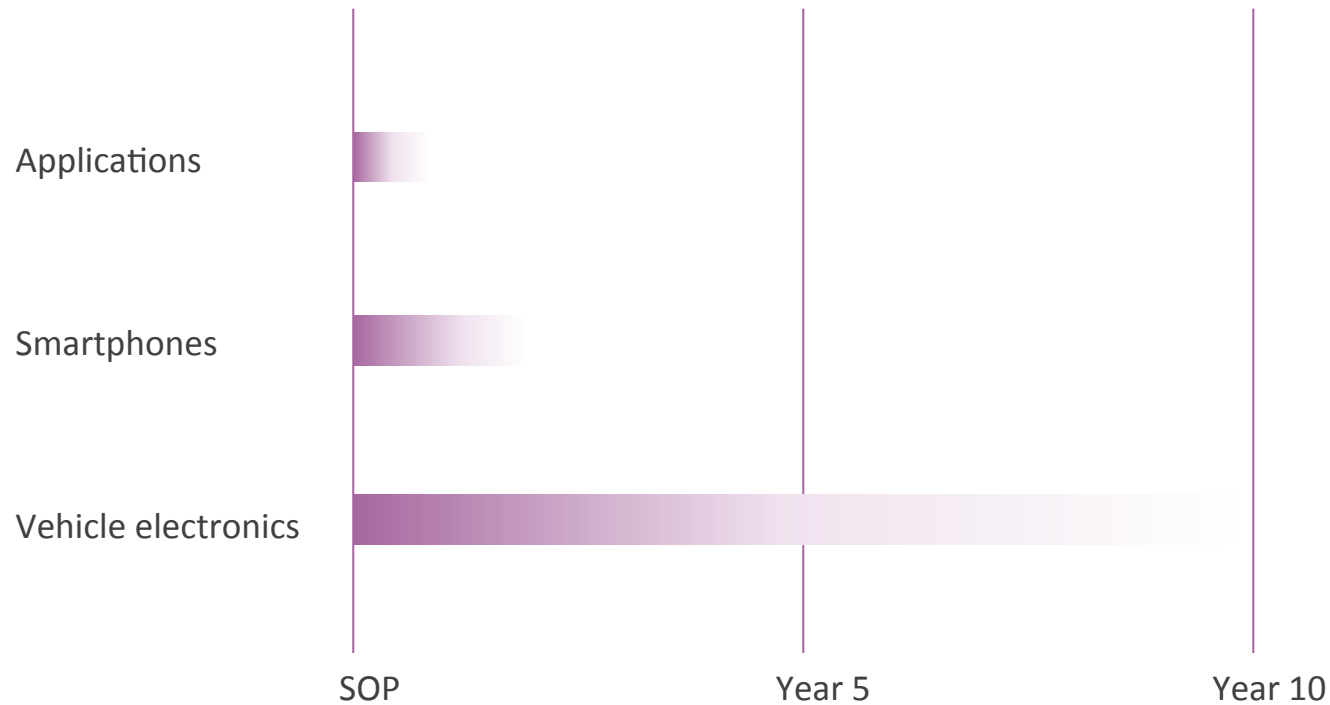
- Why HTML5 for the car?
- What can QNX contribute?
- Where do we think the W3C should focus?



Why?

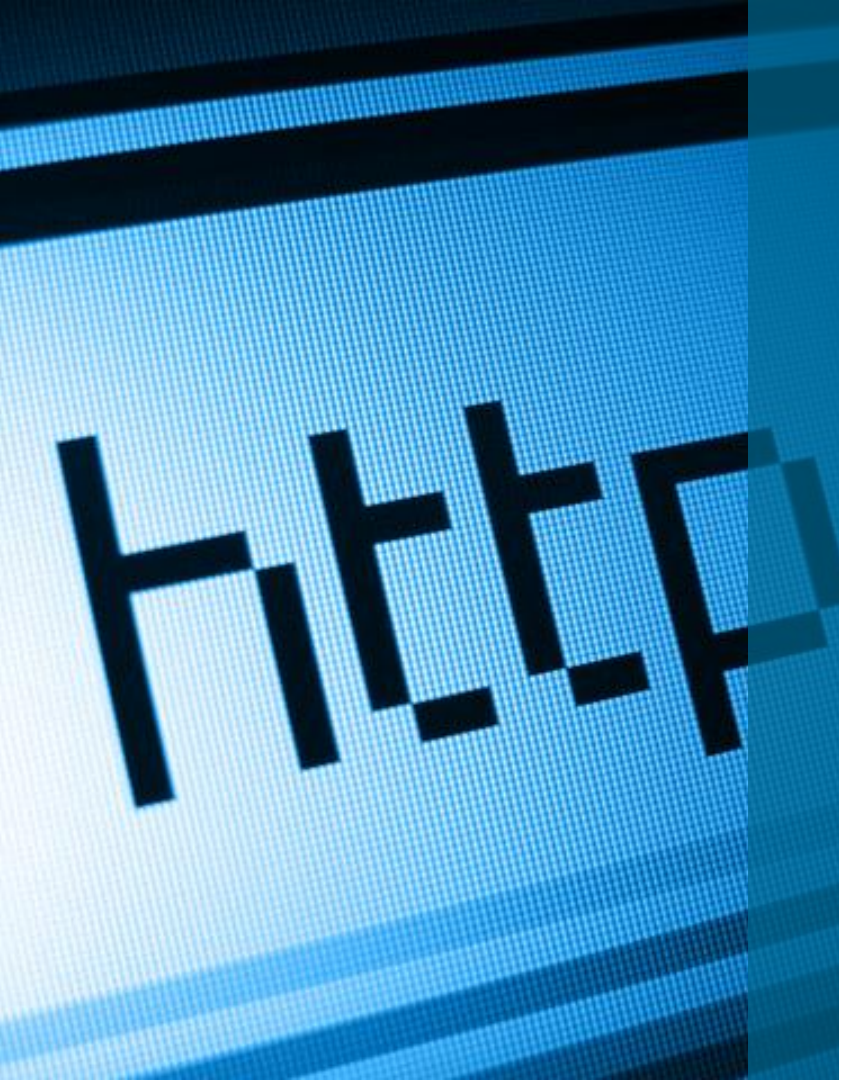


Consumer vs OEM lifecycle



Increasing consumer demand

- Car-device connectivity
 - Strong desire for ubiquitous access to online content and communities
- Up-to-date electronics
 - Consumer expectations of car systems influenced by increasingly better mobile devices
- Personalized experience
 - New era of 'self branding' apparent everywhere: ringtones, wallpaper, Facebook, etc

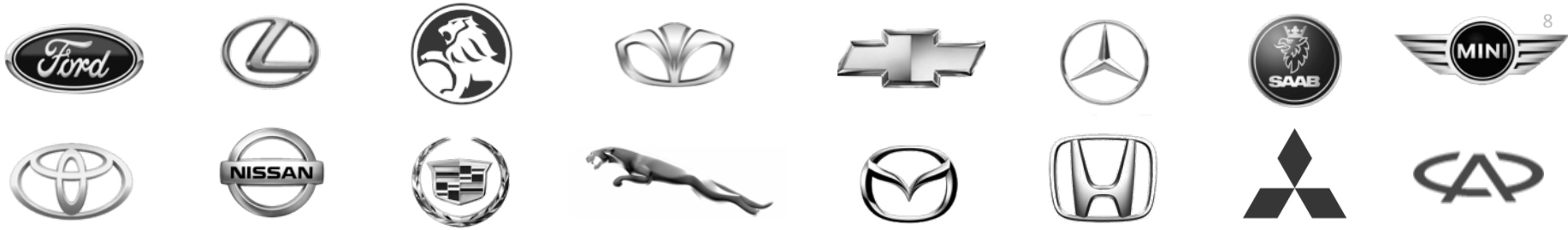


HTML5 is a natural choice

Build on something bigger than automotive

- **Ecosystem** – developers, tools, companies
- **Standards** – no vendor lock-in
- **Flexibility** – fluid deployment and architectures
- **Branding** –brand same app on different cars with CSS
- **Lifespan** – will be supported for long time
- **Time to market** – easy to use and leverage
- **Powerful** – rich application environment
- **Cross-platform** – deploy apps on phones and car

What?



QNX in automotive

In 2011

- 60+% of infotainment systems shipped
- 9+ million world wide (5+ million in NA)
- 40+% of all cars sold in US





QNX CAR 2 application platform

BB10 software stack

- + HTML5 UI and frameworks
- + mobile connectivity
- + automotive specific technologies
- + automotive-hardening
- + driver-friendly UX
- + driver distraction sensitivity

QNX CAR 2 feature highlights



Navigation

- Elektrobit embedded
- Telenav hybrid
- TCS off-board

Infotainment + social media

- Media/AM/FM/HD
- Pandora, Stitcher
- Web browser, YouTube
- Facebook, Twitter

Automotive

- Personalization
- Climate control
- Virtual mechanic
- Audio control

Connectivity

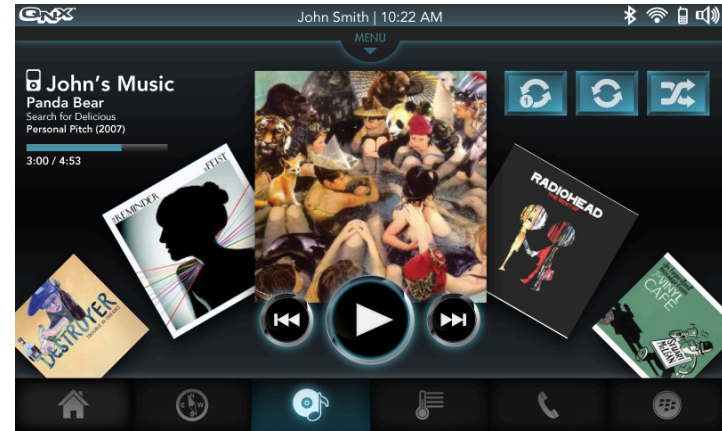
- Bluetooth + SMS
- MirrorLink + iPod Out
- Smartphone w/ HTML5
- DLNA

Platform + framework

- Torch browser
- Composition manager
- HTML5 framework
- App store support

QNX CAR 2 Applications

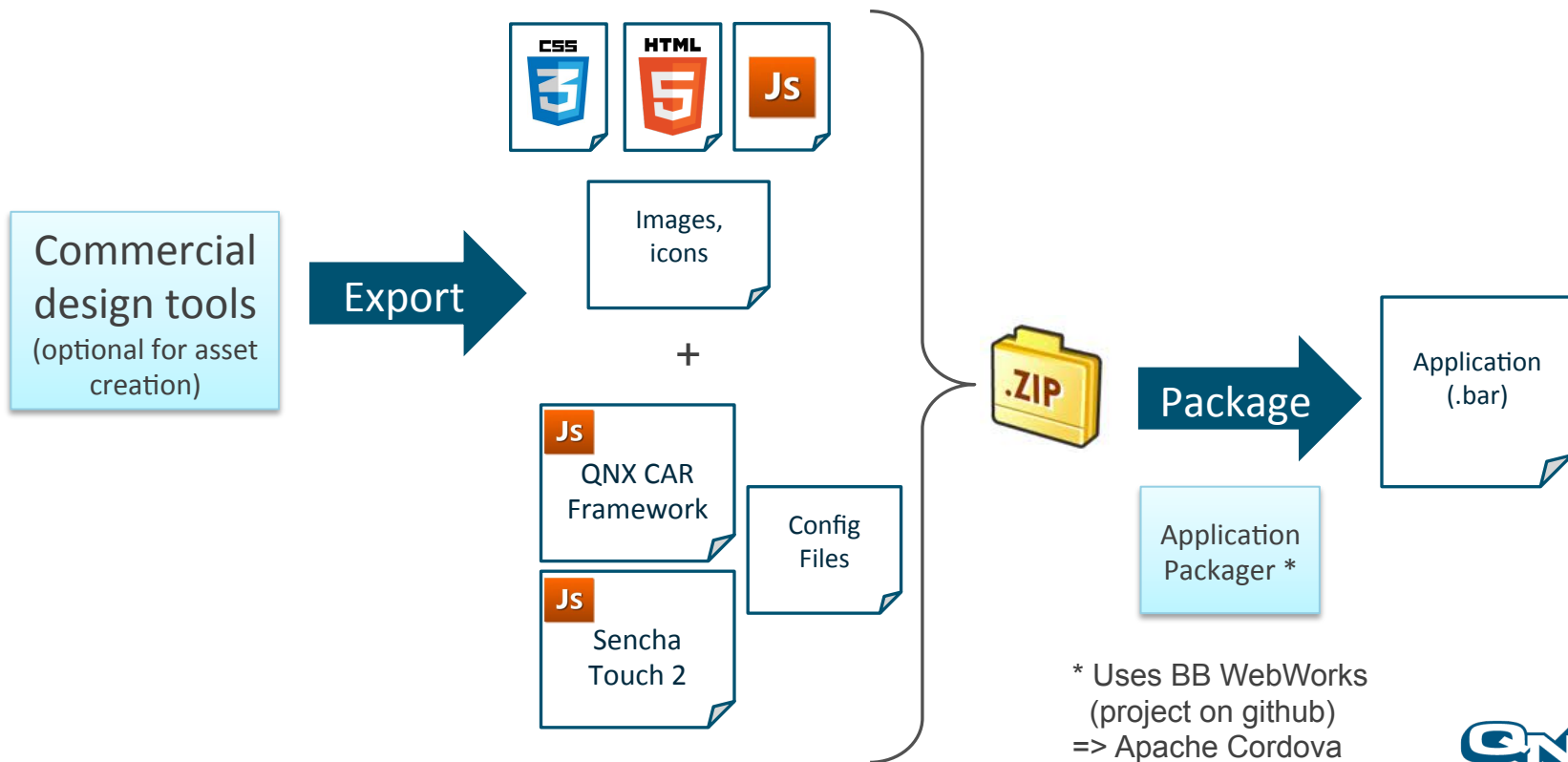
Sample HMI Designs



QNX CAR HTML5 framework

- Web technology framework for automotive
 - HTML5 environment w/ front-end + apps
 - allows entire user interface to be easily reskinned
 - launcher, controls, apps, behaviour, etc
 - Sencha Touch Mobile and JQuery for widgets
 - QNX-designed infotainment skin
- App store integration
 - designed for downloading app bundles

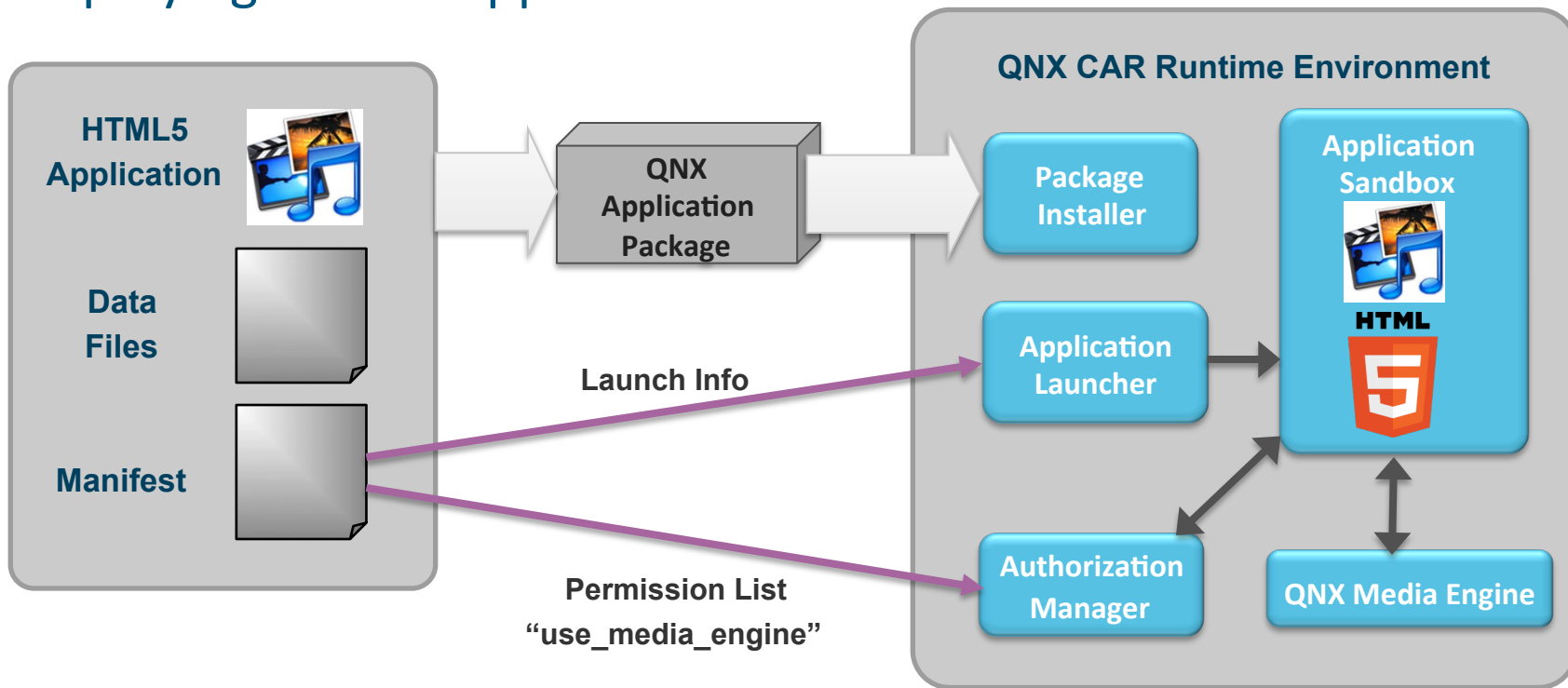
HTML5 app packaging



* Uses BB WebWorks
(project on github)
=> Apache Cordova


HTML5 Application Environment

Deploying HTML5 applications



Devices +

QNX Car

Orientation 

Platforms +

Platform QNX Car WebWorks

Version 1.0.0

Change Platform

Information +

Platform: QNX Car WebWorks
Device: QNX Car
OS: QNX Car QNX Car
Manufacturer: Research In Motion
Screen: 800x395
Density: 169.55 PPI
Browser(s): Webkit
User Agent: Mozilla/5.0 (QNX Car; U; QNX Car OS 1.0.0; en-US) AppleWebKit/534.8+ (KHTML, like Gecko) Version/0.0.1 Safari/534.8+.



Ripple for QNX CAR


```

mytest.c | in stub.h | 10 resmgr
tattr.context_free = dispatch_block_free;
tattr.block_func = dispatch_block;
tattr.unblock_func = dispatch_unblock;
tattr.handler_func = dispatch_handler;

/* Initialize the message dispatcher */
Dispatch = dispatch_create (&tattr, lo_water = 2;
->Dispatch == NULL) hi_water = 8;
tattr.increment = 1;
tattr.maximum = 50;

if (error)
mgr->ThreadPool = thread_pool_create (&tattr, 0x00, sizeof (thread_pool_t);
error = errno;
goto error;
context_alloc = dispatch_context_alloc;
context_free = dispatch_context_free;
block_func = dispatch_block;
unblock_func = dispatch_unblock;
handler_func = dispatch_handler;
/* Initialize all the resource manager stuff */
iofunc_func_init (_RESMGR_CONNECT_NFUNCS, &
IOFuncs.close_ocb = handle_close;
IOFuncs.devctl = handle_devctl;
IOFuncs.unblock = handle_unblock;

lo_water = 2;
hi_water = 8;
increment = 1;
maximum = 50;

ThreadPool = thread_pool_create (&tattr,
->ThreadPool == NULL)

error = errno;
error;

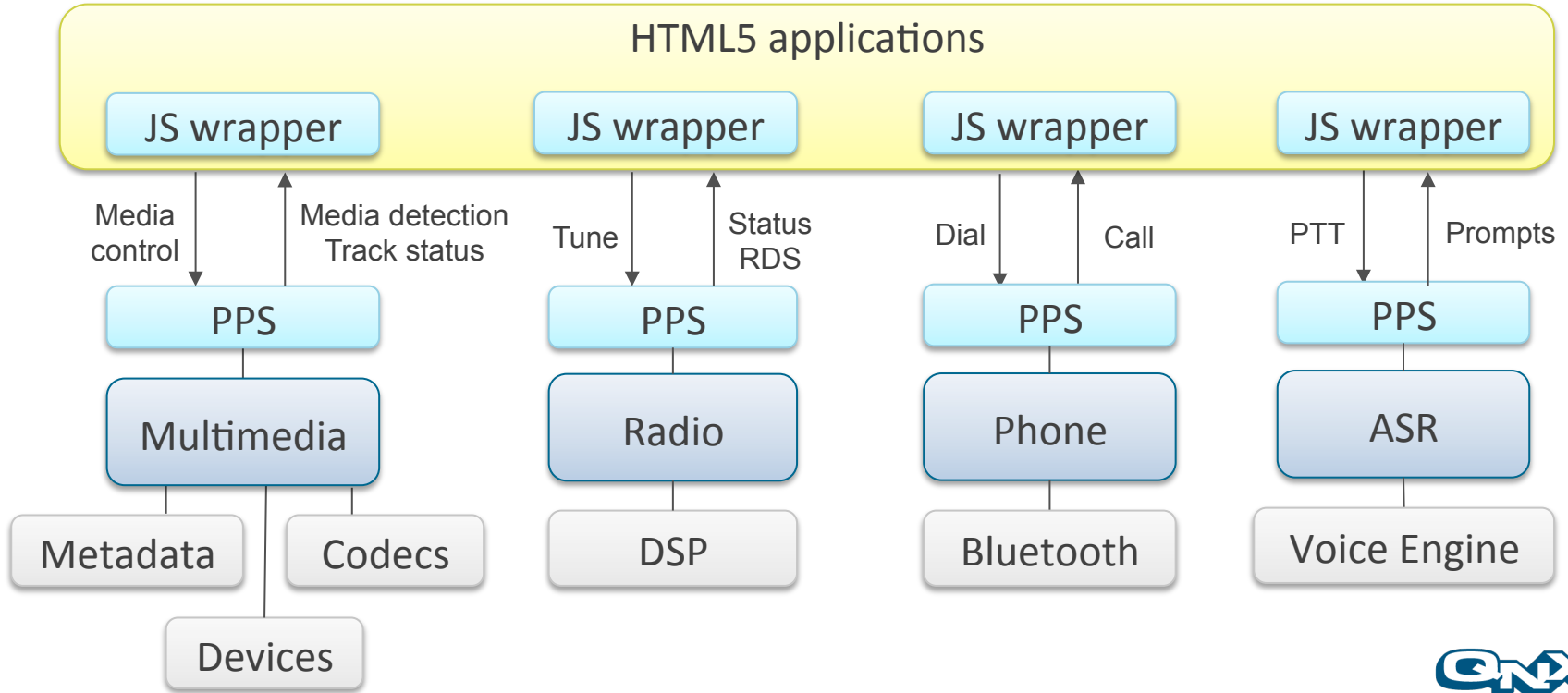
/* Initialize all the resource manager stuff */
iofunc_func_init (_RESMGR_CONNECT_NFUNCS, &
IOFuncs.close_ocb = handle_close;
IOFuncs.devctl = handle_devctl;
IOFuncs.unblock = handle_unblock;

```

QNX CAR and native access

- Bindings to native
 - general PPS, app launcher, authorization manager
 - graphics & composition manager (window management of other processes, etc)
 - SQL database
- Interact + control any app from HTML5

JavaScript access to platform services



Example JavaScript components

- Framework classes
 - application
 - event
 - keyboard
 - io
 - message
 - notification
 - settings
 - system
 - theme
 - users
- Access classes
 - audio mixer
 - audio player
 - bluetooth
 - hvac
 - media library
 - navigation
 - voice
 - phone
 - radio
 - sensors

Example audioplayer methods

- `setTrackSession (config, index);`
- `play ();`
- `playAt (index);`
- `pause ();`
- `stop ();`
- `next ();`
- `prev ();`
- `seek (position);`
- `setShuffle (mode);`
- `setRepeat (mode);`
- `isStopped ();`

Where?

Needed areas of W3C focus

1. Application packaging

- Ability to move app development across platforms
- App developers have larger addressable market
- OEMs require less customization to adopt specific apps
- Cordova?

Needed areas of W3C focus

2. Native access APIs

- Ability to consistently access underlying functionality
- Consistency for app developers and OEMs
- OEMs will still need ability to customize what's available where

Needed areas of W3C focus

3. Mobile + car integration

- Ability to bring mobile hosted apps into auto experience
- MirrorLink?

Needed areas of W3C focus

4. Distraction prevention and OEM skinning

- Consistent style sheet tags and mechanisms
- OEM-approved templates for common use cases
- With templates, frees app makers from deep understanding of auto concerns
- Gives OEMs ability to differentiate and create a consistent brand look and feel

Needed areas of W3C focus

5. App development guidelines

- “How to” for mobile developers entering the automotive space

© 2012 QNX Software Systems Limited. QNX, NEUTRINO, MOMENTICS, AVIAGE, PHOTON, PHOTON MICROGUI are trademarks of QNX Software Systems Limited, which are registered trademarks and/or used in certain jurisdictions. All other trademarks belong to their respective owners. The information herein is for informational purposes only and represents the current view of QNX Software Systems Limited (QSS) as of the date of this presentation. Because QSS must respond to changing market conditions, it should not be interpreted to be a commitment on the part of QSS, and QSS cannot guarantee the accuracy of any information provided after the date of this presentation. QSS MAKES NO WARRANTIES, REPRESENTATIONS OR CONDITIONS EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Andy Gryc
agryc@qnx.com

