



BMW Group Research and Technology / BMW AG /
TU Munich, November 2012

ENABLING RICH WEB APPLICATIONS FOR IN-VEHICLE INFOTAINMENT.

USING THE WEBINOS PLATFORM INSIDE THE AUTOMOTIVE DOMAIN.

**BMW
GROUP**



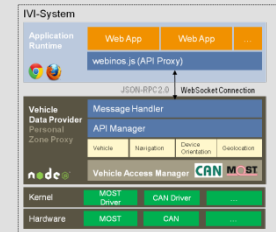
AGENDA.



Motivation
Why Web & Automotive?



Background
What Is webinos?



Our Approach
Vehicle Data for Web Apps



Live Demo
webinos Automotive Apps



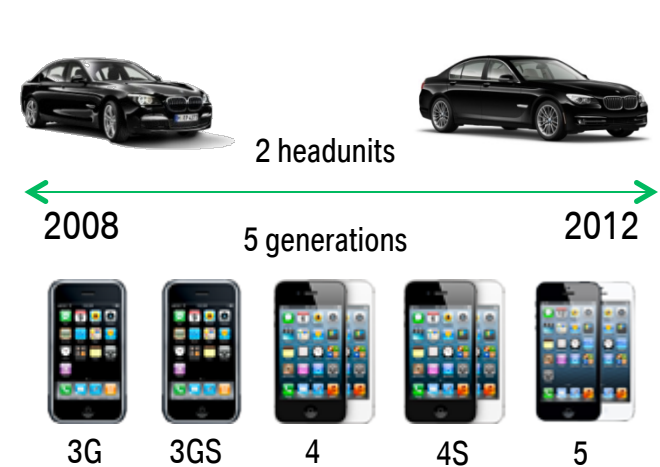
Open Questions
The Road Ahead of Us



Lessons learned
Take Home Message

MOTIVATION: CURRENT LANDSCAPE OF IVI SYSTEMS.

- Long **time-to-market** for in-car infotainment applications and services
- **Highly fragmented landscape** for In-Vehicle Infotainment (IVI) systems
- Customer **demand for more personalization** options on IVI systems (seamless use of services across different devices)
- Difficult to **attract third-party developers** for IVI systems



MOTIVATION: UBIQUITY OF THE WEB AND THE BROWSER.

- General **trend towards Web- and browser-based services and applications** on smartphones, tablets and desktops
- **Highly standardized** runtime environment for application and services
- **Large developer base**

Deduction:

The Browser is the preferred candidate for a runtime environment on IVI systems.

HTML

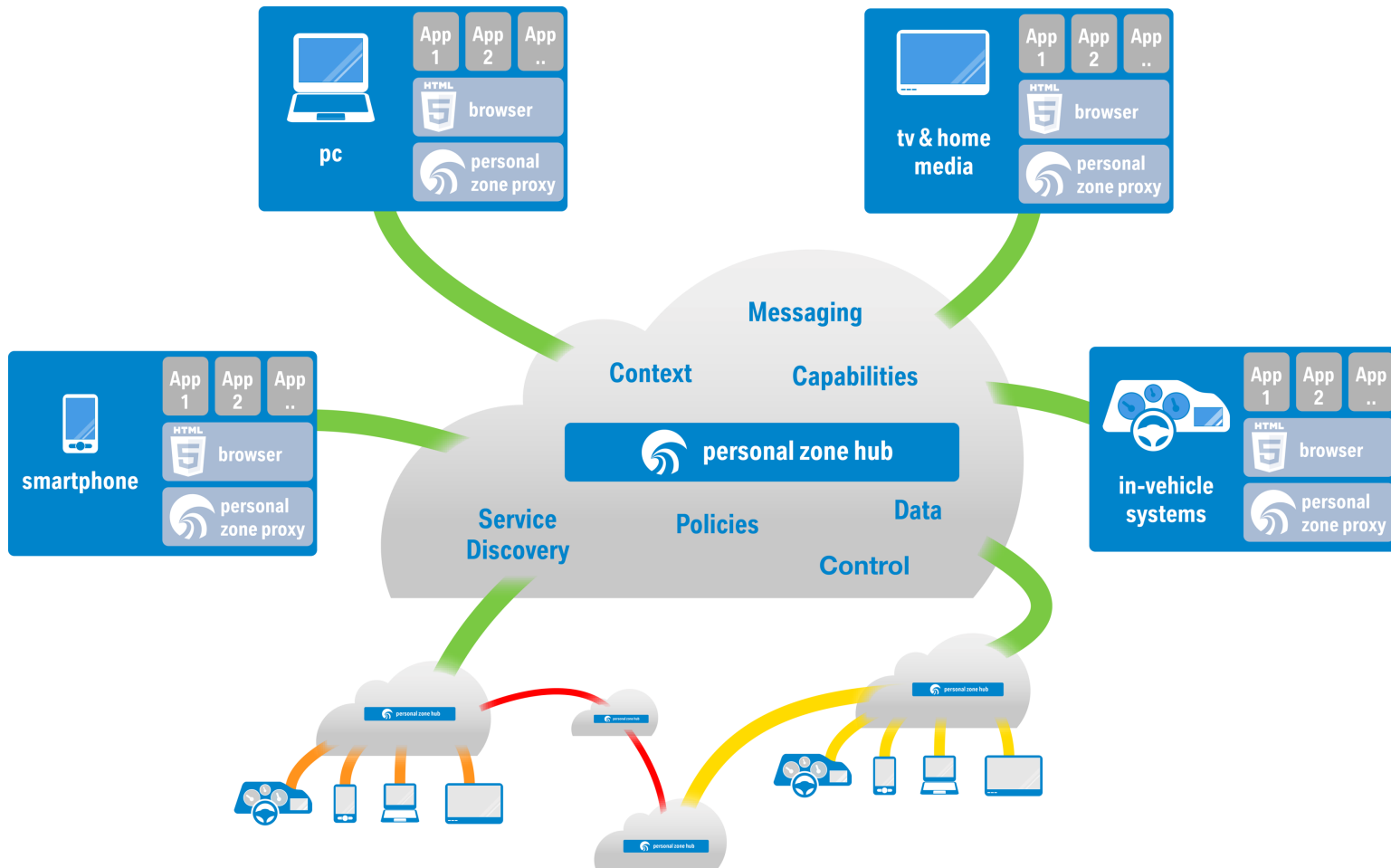


BACKGROUND: WHAT IS WEBINOS?

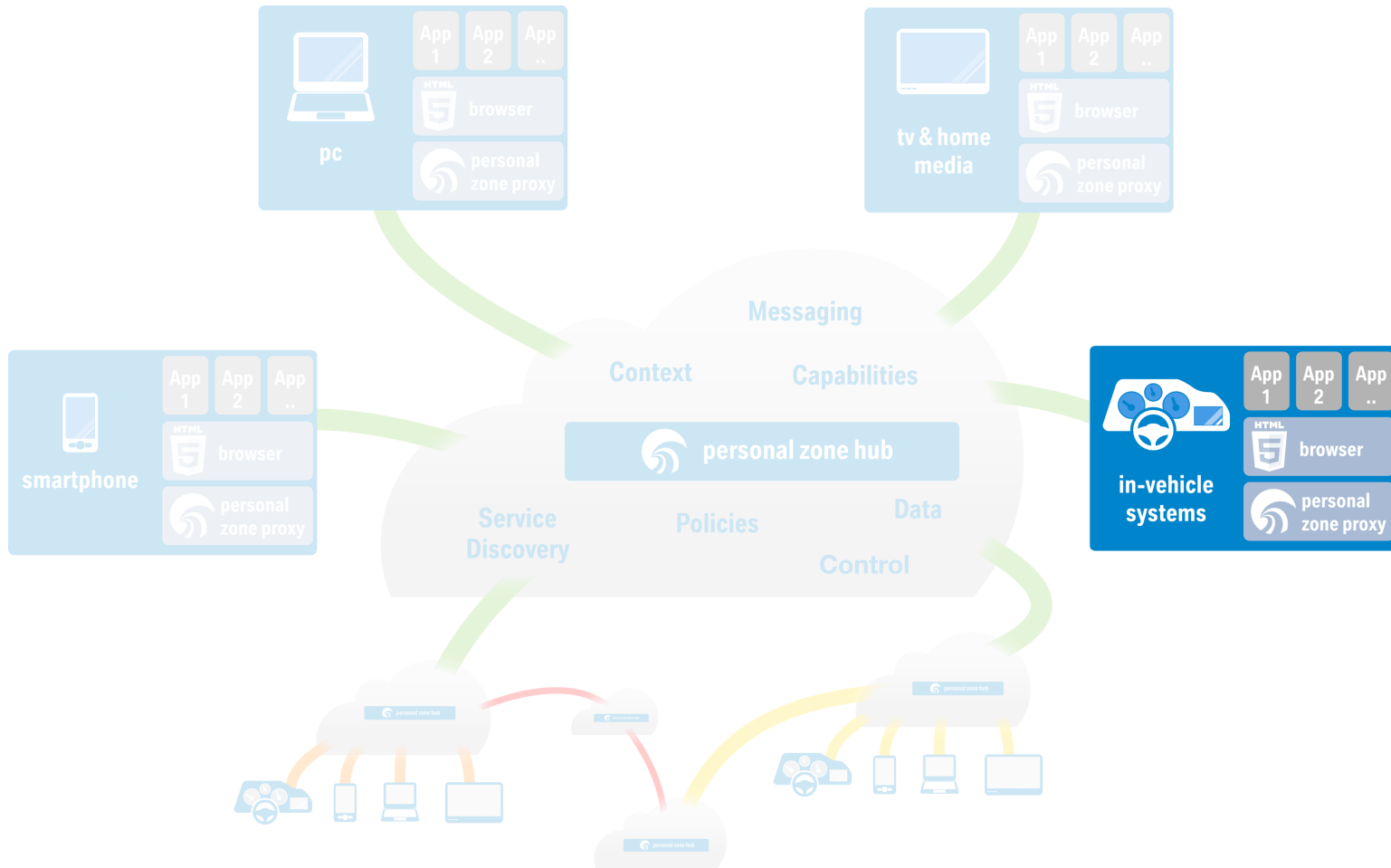


- An **open, cross-device** and **browser-based application platform**
- Research project funded by the European Commission
- 30 partners
 - Device manufactures
 - Automotive manufactures
 - Mobile network operators
 - Small and medium businesses
 - Research Institutes
 - Standardization bodies

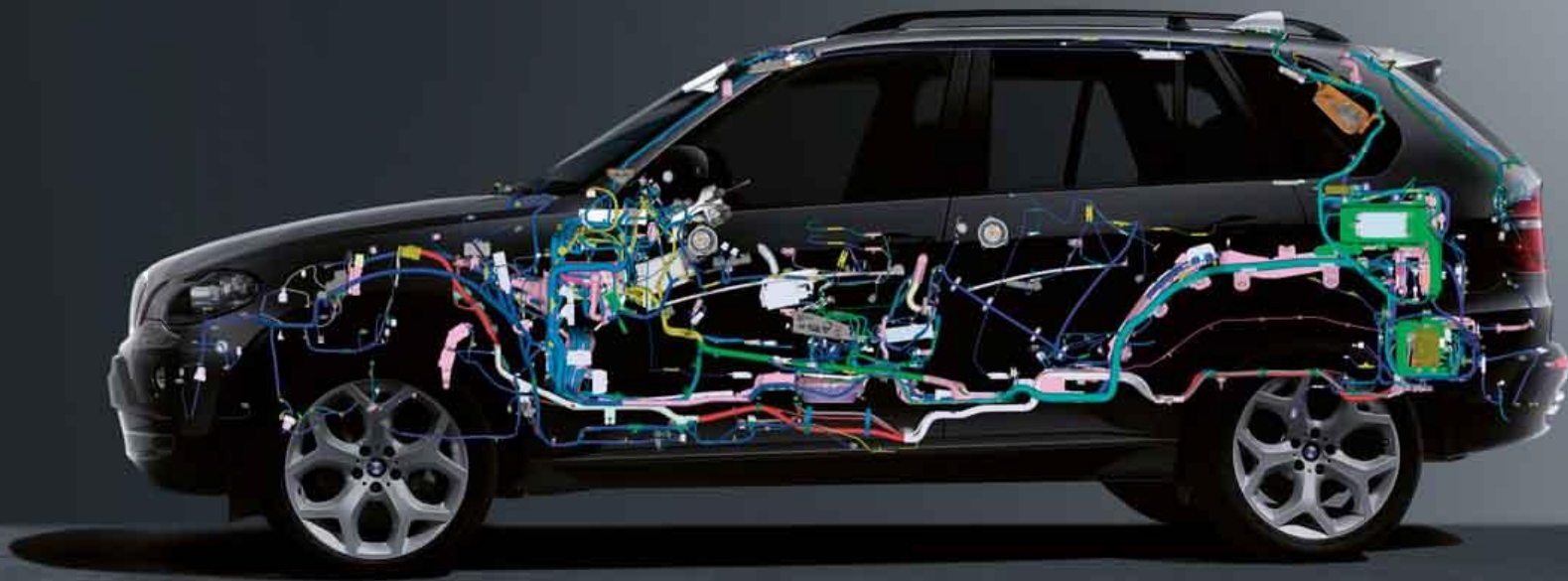
BACKGROUND: SAY HELLO TO WEBINOS!



BACKGROUND: SAY HELLO TO WEBINOS!



KEY QUESTION: HOW TO GET ACCESS TO VEHICLE DATA?



OUR APIS: DESIGN APPROACH.

- **Use data from other standardized APIs** instead of duplicating them
- **Asynchronous model** to retrieve dynamic vehicle data
- Use case driven grouping of data properties to **simplify access** for web apps
- **Minimize resource overhead** for providing vehicle data

Two 'automotive' specific APIs

- **Vehicle API** (for retrieving vehicle specific data)
- **Navigation API** (for interacting with navigation system)

VEHICLE API: GENERAL CONCEPT.

- Soley **read access** to vehicle data (at this time)
- Distinction between **static** (e.g. transmission type) and **dynamic** (e.g. gear) **vehicle data**
- Developer registers **callback handlers** for retrieving dynamic vehicle data
- Dynamic data can be **requested once** (`get`) or can be **monitored** (`addEventListener`)
- API is provided as part of `window.webinos.vehicle` object

VEHICLE API: WHICH DYNAMIC DATA IS PROVIDED?

Vehicle Data Objects																																			
Parking Sensor Data					Tripcomputer Data				Gear Data	Light Data	Engine Oil Data	Wiper Data	Window Data			Door Data		Tire Pressure			Climate Data			Seat Data											
Position [string]	outLeft [Short]	Left [Short]	Midleft [Short]	Midright [Short]	Right [Short]	OutterRight [Short]	averageSpeed [Short]	averageConsumption [Short]	Mileage [Short]	tripDistance [Short]	Range [Short]	Gear [Short]	lightId [string]	Active [boolean]	Level [string]	Position [string]	driver [Short]	behindDriver [Short]	passenger [Short]	behindPassenger [Short]	driver [Short]	behindDriver [Short]	passenger [Short]	behindPassenger [Short]	frontRight [Short]	frontLeft [Short]	rearRight [Short]	rearLeft [Short]	zone [String]	desiredTemperature [Short]	acStatus [boolean]	ventLevel [Short]	ventMode [boolean]	position [String]	settings [SeatSetting[]]

get

addEventListener

removeEventListener

```
vehicle.get('tripcomputer', tcHandler); //one-time request
vehicle.addEventListener('tripcomputer', tcHandler); //monitor tripcomputer
```

```
function tcHandler(data){
  console.log('Average consumption': + data.averageConsumption);
  console.log('Average speed': + data.averageSpeed);
  console.log('Average trip speed': + data.tripSpeed);
  // ...
}
```

VEHICLE API: WHICH STATIC DATA IS PROVIDED?

- brand (string)
- model (string)
- year (string)
- fuel (enum)
- hybrid (enum)
- steeringwheel
- transmission



Brand= BMW
Model= ActiveHybrid 7
Year=2012
Fuel=Gas
Hybrid= mild
Steeringwheel= left
Transmission= automatic

NAVIGATION API: GENERAL CONCEPT.

- simple API to **interact with navigation software**
- API is provided as part of the `window.webinos.navigation` object
- **query for Point-of-Interests** within a specified area
(`findDestination`)
- Set the next destination of the navigation system (`requestGuidance`)
- Retrieve status of the navigation system (`getStatus`)

NAVIGATION API: CODE EXAMPLE.

Find a destination

```
var destinations =new Array();

webinos.navigation.findDestination("BMW
Welt", destinationCB, errorCB);

function destinationCB(pois){
  if(destinations.length > 0){
    navigateTo(destinations[0])
  }
  else{
    console.log("No POI found");
  }
}
```

Request guidance

```
function navigateTo(destination){
  webinos.navigation.requestGuidance(
    destination,false, navigationHandler);
}

//callback handler for guidance events
var navigationHandler = {
  onRequest: function(id, poi){
    console.log('Guidance set to' + poi.name);
  },
  onReach: function(id, poi){
    console.log(poi.name + ' reached. ');
  },
  onCancel: function(id, poi){
    console.log('Guidance to ' + poi.name + '
is
cancelled. ');
  }
}
```

NAVIGATION API: CODE EXAMPLE.

Find a destination

```
var destinations =new Array();

webinos.navigation.findDestination("BMW
Welt", destinationCB, errorCB);

function destinationCB(pois){
  if(destinations.length > 0){
    navigateTo(destinations[0])
  }
  else{
    console.log("No POI found");
  }
}
```

Request guidance

```
function navigateTo(destination){
  webinos.navigation.requestGuidance(
    destination,false, navigationHandler);
}

//callback handler for guidance events
var navigationHandler = {
  onRequest: function(id, poi){
    console.log('Guidance set to' + poi.name);
  },
  onReach: function(id, poi){
    console.log(poi.name + ' reached. ');
  },
  onCancel: function(id, poi){
    console.log('Guidance to ' + poi.name + '
is
cancelled. ');
  }
}
```

NAVIGATION API: CODE EXAMPLE.

Find a destination

```
var destinations =new Array();

webinos.navigation.findDestination("BMW
Welt", destinationCB, errorCB);

function destinationCB(pois){
  if(destinations.length > 0){
    navigateTo(destinations[0])
  }
  else{
    console.log("No POI found");
  }
}
```

Request guidance

```
function navigateTo(destination){
  webinos.navigation.requestGuidance(
    destination,false, navigationHandler);
}

//callback handler for guidance events
var navigationHandler = {
  onRequest: function(id, poi){
    console.log('Guidance set to' + poi.name);
  },
  onReach: function(id, poi){
    console.log(poi.name + ' reached. ');
  },
  onCancel: function(id, poi){
    console.log('Guidance to ' + poi.name + '
is
cancelled. ');
  }
}
```


NAVIGATION API: CODE EXAMPLE.

Find a destination

```
var destinations =new Array();

webinos.navigation.findDestination("BMW
Welt", destinationCB, errorCB);

function destinationCB(pois){
  if(destinations.length > 0){
    navigateTo(destinations[0])
  }
  else{
    console.log("No POI found");
  }
}
```

Request guidance

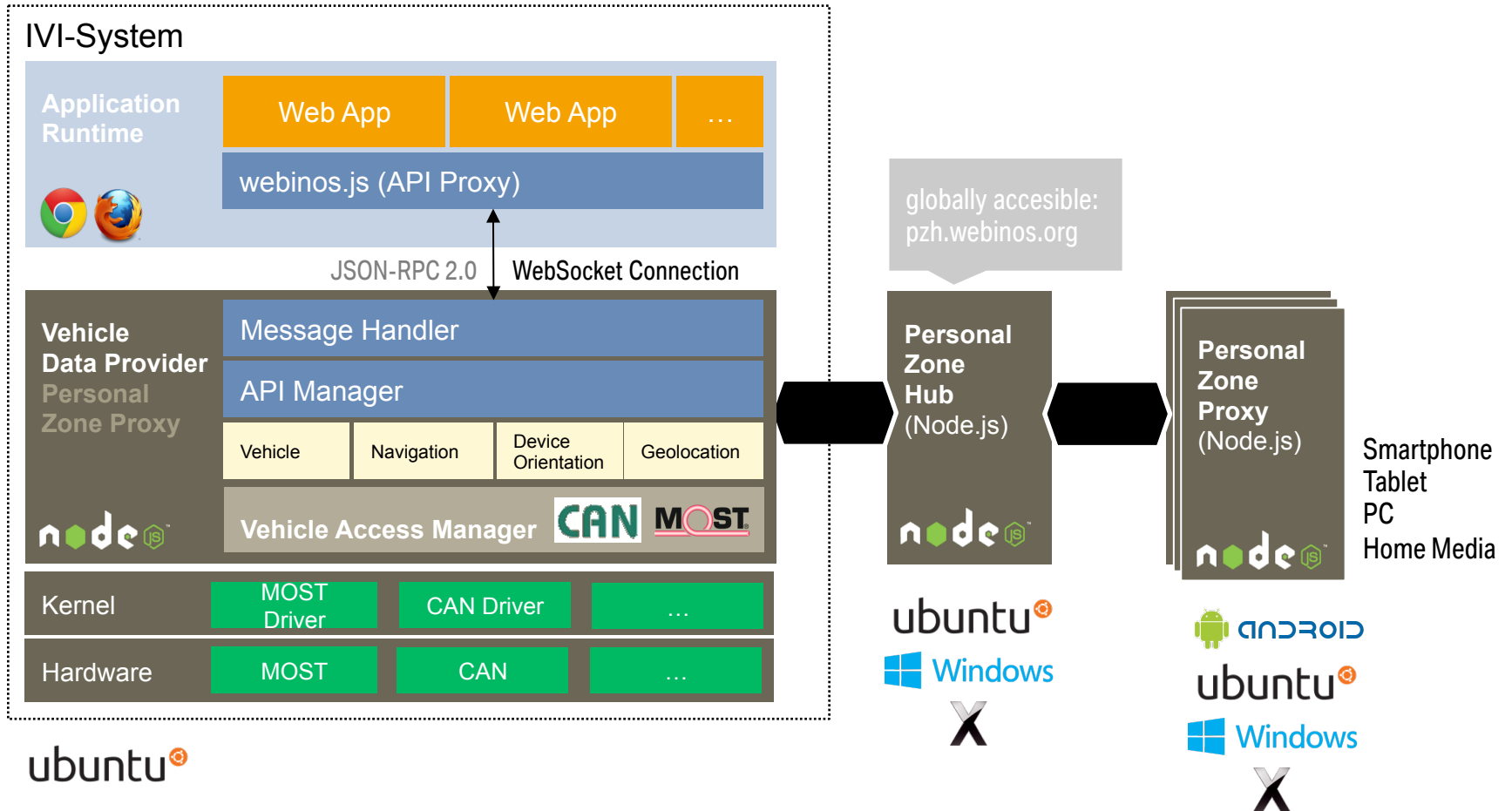
```
function navigateTo(destination){
  webinos.navigation.requestGuidance(
    destination,false, navigationHandler);
}

//callback handler for guidance events
var navigationHandler = {
  onRequest: function(id, poi){
    console.log('Guidance set to' + poi.name);
  },
  onReach: function(id, poi){
    console.log(poi.name + ' reached. ');
  },
  onCancel: function(id, poi){
    console.log('Guidance to ' + poi.name + '
is
cancelled. ');
  }
}
```

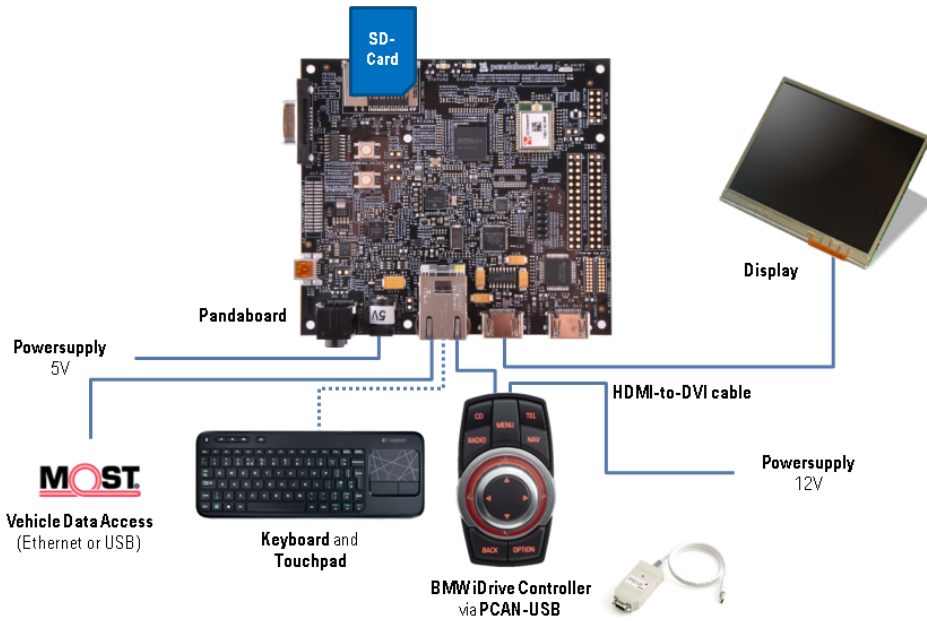
HOW TO INTEGRATE WEBINOS INTO THE VEHICLE.



WEBINOS CORE CONCEPT: SEPARATING APPLICATION RUNTIME FROM DATA ACCESS.



WEBINOS VEHICLE EVALUATION PLATFORM.



Basic infrastructure



Demonstrator box

WEBINOS VEHICLE DEMONSTRATION PLATFORM.



LIVE DEMO.



Browser-based trip computer for In-car head units

- Pure web technology (HTML, CSS, JavaScript, Canvas)
= HTML5 app
- webinos API for accessing vehicle data

Seamless trip planing on desktop, smartphone, and IVI-System

- Create travel itineraries
- Manage points of interest
- Data is automatically synchronized between the devices within a personal zone.

THE ROAD AHEAD OF US.



THE ROAD AHEAD OF US.

- How are we going to **control the access** to the vehicle bus?
- How to **enable write access** in conjunction with access control?
- **Adapting Web applications** to be safely used inside the vehicle
 - Handling different input controls
 - Adjusting graphical user interface
- Can we agree on a **common interface for vehicle data**?

TAKE HOME MESSAGE.

- **Proof-of-concept** for exposing vehicle data to Web applications
- **Read only** for vehicle data
- **Security** and **Safety** need to be solved properly
- **UI Constraints** need to be addressed