

# Firefox OS App Days

Web APIs

Author: José M. Cantera (@jmcantera)  
Last update: March 2013

# Open Web Platform Interfaces

## ■ Web APIs

- Expose native device functionality to the Web Platform
- Specified by Mozilla through an open process
- Developed by Mozilla and partners
- Pushed forward in W3C for standardization

# Web Activities and System Messages

# System Messages

- System Messages are global, asynchronous events delivered by the system to apps
  - Similar to DOM events but the system guarantees delivery even if the app is not currently running
  - Gecko and Gaia (System) will be in charge of starting the app if not running
  - Avoid background services which are usually overkilling
  
- System Messages are a key component of Firefox OS and a number of APIs depend on this mechanism
  - Alarm API (for scheduling tasks)
  - Web Activities API (for app interconnection and delegation)
  - Bluetooth
  - SMS and Web Telephony.
    - › Incoming call or SMS are notified through this mechanism

# System Messages (How they work)

- The receiving app must declare what system messages are of interest (manifest.webapp)

- Some messages could be subject to the permission granting on the associated API (alarms, telephony, ...)

- At application startup it must be set a message handler that will handle the concerned system messages

```
"messages": [  
  { "alarm": "/facebook/fb_sync.html" },  
  { "notification": "/dialer/index.html#keyboard-view" },  
  { "telephony-new-call": "/dialer/index.html#keyboard-view" },  
]
```

```
function newCall() {  
  openCallScreen();  
}  
navigator.mozSetMessageHandler('telephony-new-call', newCall);
```



# Web Activities

- Allow applications (clients) to request the execution of activities to be performed by other applications (providers)
- A mechanism that simplifies certain operations
  - Picking a Contact
  - Picking a photo from the gallery
  - Taking a picture from the camera
  - Dial a number
  - .....
- System App dialog allows to select between a list of potential app providers

# Web Activities (how they work)

- Provider declares what activities the app can offer to other apps
- Example: Gallery App manifest.webapp

```
"activities": {  
  "pick": {  
    "filters": {  
      "type": ["image/jpeg", "image/png"]  
    },  
    "disposition": "inline",  
    "returnValue": true,  
    "href": "/index.html#pick"  
  }  
}
```

- Client requests a service

```
var activity = new MozActivity('pick', 'image/*')  
activity.onsuccess = function (e) { e.target.result }  
activity.onerror = function() {e} { }
```

```
navigator.mozSetMessageHandler('activity', function(activity) {  
  activityName = activity.source.name;  
});
```

```
window.addEventListener('click',function() { activity.postMessage(selectedImage) });
```

# Web APIs



# Web APIs List

- Web Telephony (W3C Editor's Draft)
- Web SMS / MMS (W3C Editor's draft)
- Contacts (W3C draft)
- Settings
- Media Storage
  - › Getting access to media storage (sdcard)
- Camera Capture
- Mobile Network Connection, Wi-Fi Network, Web Bluetooth
- Power Management
- Alarms (W3C draft)
- Web FM Radio
- Battery, Vibration, Sensors (W3C CR)
- Screen Orientation (W3C WD)
- TCP Socket
- ... and more ...

# Web API – WebTelephony

- **Purpose** : Allows to initiate and control POTS phone calls
- **Status**: W3C Editor's Draft available
- **Spec**: <http://sysapps.github.com/sysapps/proposals/Telephony/Telephony.html>

## ■ Examples

```
var call = navigator.mozTelephony.dial('+34983367743'); // Launch a call
call.addEventListener('statechange', function(e) { // Listening
    if(e.target.state === 'alerting') window.console.log('Ringing the other party!');
    if(e.target.state === 'connected') { // Do something. The other party answered }
});
```

- **Next Steps** : W3C FPWD

# Web API – WebSMS / MMS

- **Purpose** : Allows to send and receive messages.
  - Includes database to keep folders in persistent storage
- **Status**: W3C Editor's Draft available.
- **Spec**: <http://sysapps.github.com/sysapps/proposals/Messaging/Messaging.html>

## ■ Examples

```
var req = navigator.mozSms.send('+34638883076','Hi, coffee?'); // Send  
req.onsuccess = function() { window.console.log('Sent!!') };
```

```
navigator.mozSms.addEventListener('received',receivedCB); // Listen to new
```

- **Next Steps** : Unify mozSMS, mozMMS → W3C FPWD

# Web API – Contacts

- **Purpose** : Create, read, update, delete and find contacts.

- **Status**: W3C Editor's Draft available

- **Spec**: <http://www.w3.org/TR/contacts-manager-api/>

- **Examples**

```
var options = {filterBy: ['tel'], filterOp: 'contains', filterValue: '983367743'};
var request = navigator.mozContacts.find(options);
request.onsuccess = function (e) {
    if (e.target.result.length == 0)
    else { contactName = e.target.result[0].name }
}
```

# Web API – Settings API

- **Purpose** : Read and modify setting properties from a vocabulary.
  - Locking avoids concurrency problems between apps

- **Status**: v1 Up and running

- **Spec**: [https://wiki.mozilla.org/WebAPI/SettingsAPI#Proposed\\_API](https://wiki.mozilla.org/WebAPI/SettingsAPI#Proposed_API)

- **Examples**

```
var request = navigator.mozSettings.getLock().get('language.current');
request.onsuccess = function(e) {
    selectedLocale = e.target.result['language.current'];
}
```

- **Next Steps** : Feedback from developers and W3C submission

# Web API – Device Media Storage

- **Purpose:** To retrieve and storage media files
- **Status:** v1 up and running.
- **Spec:** <https://wiki.mozilla.org/WebAPI/DeviceStorageAPI>

- **Examples:**

```
var store = navigator.getDeviceStorage('pictures');    // Picture store
store.addNamed(blob,'myPicture');                    // Add a new element
store.enumerate()                                     // Iterate over the store
```

- **Next Steps :** W3C Editor's Draft.

# Web API - Resource Lock

- **Purpose:** Prevents resource status change. Ex. Prevent switching off the screen while watching a video
- **Status:** v1 up and running
- **Spec:** <https://wiki.mozilla.org/WebAPI/PowerManagementAPI>
- **Examples:**

```
var lock = navigator.requestWakeLock('screen');  
lock.unlock();
```
- **Next Steps :** TBD.



# Web API – Notifications API

- **Purpose:** Allows applications to publish notifications

- **Status:** v1 up and running based on old W3C draft

- **Issues:** Alignment with current W3C draft

- **Spec:** <http://www.w3.org/TR/notifications/>

- **Examples:**

```
var notif = navigator.mozNotification.createNotification ("Meeting about to  
begin", "Room 101", 'icon.png' );  
notif.onclick = function() { // Do something };  
notif.show();
```

- **Next Steps :** W3C Working Draft available.

# Web API – Alarms API

- **Purpose:** Allows applications to schedule tasks at a time. Based on System Messages.

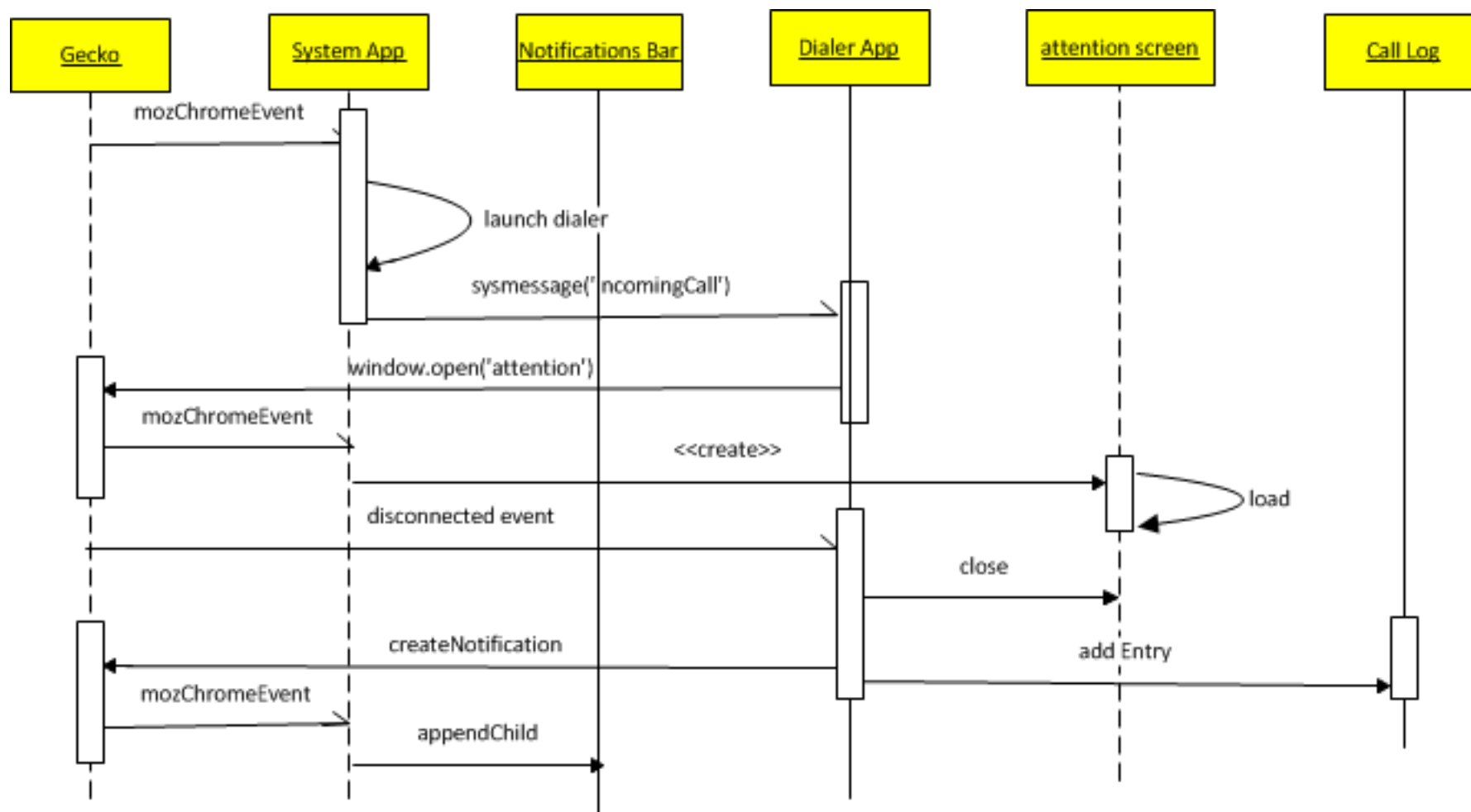
- **Status:** v1 up and running based on W3C draft

- **Spec:** <http://www.w3.org/TR/web-alarms/>

- **Examples:**

```
var req = navigator.mozAlarms.add(  
    new Date('March 21, 2013 20:00:00'), 'ignoreTimezone');  
req.onsuccess = function(e) {  
    window.console.log('Alarm Id:', req.result);  
}
```

# Web APIs in action



*Telefónica*

---