# Security for access to device APIs

Stewart Brodie

ANT Galio Browser Software Team Leader

ANT Software Ltd.

# *WAFERs: Overview*

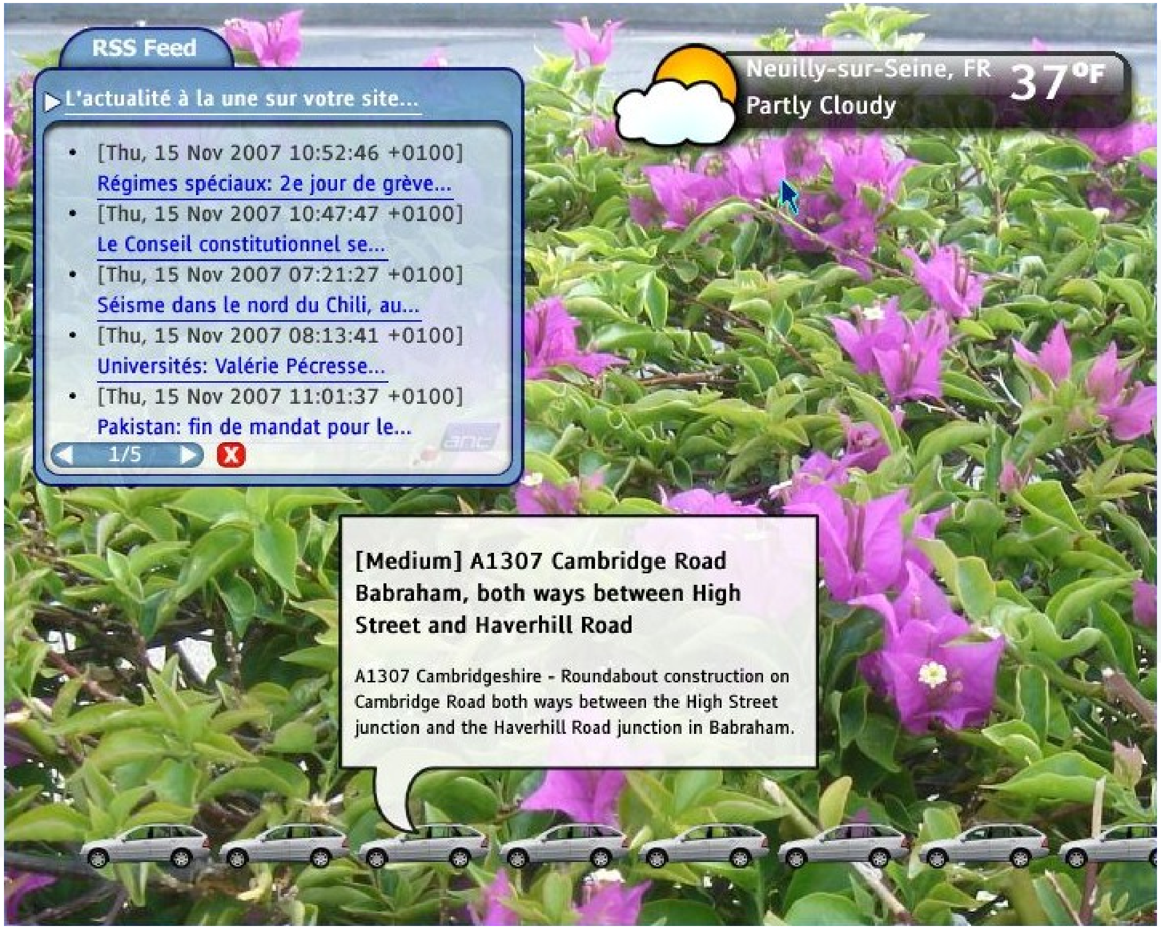
- *An application model for HTML + JavaScript content*
  - Requires no changes to an existing HTML document
  - Only difference is how they are launched

- *Supports multiple simultaneous applications*
  - Foreground and background applications
  - Independent browsing contexts

- *Main features:*
  - Support for visible applications (UI applications)
  - Support for invisible applications (services)
  - Applications can overlap on screen (and do by default)
  - Enables consistent event delivery across multiple apps
  - Applications are notified when system state changes
  - *Privileged access to extended APIs*

- *Does not cover application signalling*

# WAFERs in action



▶ L'actualité à la une sur votre site...

- [Thu, 15 Nov 2007 10:52:46 +0100] Régimes spéciaux: 2e jour de grève...
- [Thu, 15 Nov 2007 10:47:47 +0100] Le Conseil constitutionnel se...
- [Thu, 15 Nov 2007 07:21:27 +0100] Séisme dans le nord du Chili, au...
- [Thu, 15 Nov 2007 08:13:41 +0100] Universités: Valérie Pécresse...
- [Thu, 15 Nov 2007 11:01:37 +0100] Pakistan: fin de mandat pour le...

Neuilly-sur-Seine, FR 37°F
Partly Cloudy

[Medium] A1307 Cambridge Road Babraham, both ways between High Street and Haverhill Road

A1307 Cambridgeshire - Roundabout construction on Cambridge Road both ways between the High Street junction and the Haverhill Road junction in Babraham.

# *Protecting privileged APIs*

- *Privileged browsing contexts have additional properties and fewer restrictions:*
  - e.g. XMLHttpRequest same-origin checks are bypassed
  - Windows may be resized without regard to the minimum dimensions
  - Access to a set of API objects (one per-context, like the Navigator, Screen objects)

- *Built-in C code can add to the set of API objects, knowing that:*
  - only privileged browsing contexts can access these properties
  - this provides a level of security to separate applications & untrusted content
  - there is no need to perform any security checks when methods are invoked

- *Simple ...*
  - Easy to audit the permissions
  - Easy to enforce the permissions
  - No impact on performance
  - OK when the service operator's system is closed
  - ... *too* simplistic when applications are sourced from different providers

# *Drawbacks of current approach*

- *All-or-nothing approach is inflexible*
  - Hard to grant restricted set of permissions to an unprivileged application
  - Hard to grant restricted set of permissions to a privileged application, too!
  - One rogue application can hijack the system
- *Privileged applications can break the security model deliberately ...*
  - e.g. Careful applications can store closures in the global objects of unprivileged contexts
- *... but really should not.*
  - Careless applications can store the API objects, granting full access to those APIs!
- *Need a way to grant permissions in a controlled way to unprivileged applications*

# *Key requirements for API security*

- *Definition of permissions*
  - Must be easy to write, easy to audit, easy to verify
  - Build on MHP/OCAP?
  - Tamper-protection – digital signatures (and who needs to sign and how much will it cost?)
- *Define the scope for a set of permissions*
  - a browsing context?
- *Checking permissions*
  - Must be fast to evaluate - no expensive computation on each method invocation
- *Define mechanism for handling security violations*
  - Raise a DOM security exception?
  - Terminate the application?
  - Typically, prompting the user is not an *option*!

# Security for access to device APIs

Stewart Brodie
ANT Galio Browser Software Team Leader
ANT Software Ltd.