DRAFT Report of the Patent Advisory Group on the Widgets Access Requests Policy Specification

Published on 26 October 2011

Table of contents

PAG Home
Prior Art
Call for prior art
Public mailing list
Group page
[Member only]

Executive Summary

In reponse to Apple's withdrawal of IP, the Widgets Access Request Policy (WARP) PAG concluded that the claims in US patent 7,743,336: *Widget security* (hereinafter also called 'the 336 patent') which could potentially be read to cover the WARP Specification are anticipated by substantial prior art. Consequently, the PAG does not believe there are valid 'Essential Claims' against the WARP Specification. Furthermore, the US Patent Application 20070101146: *Safe distribution and use of content* (hereinafter also called 'the 1146 Application') is seen as orthogonal to the Widget Access Request Policy Candidate Recommendation 20 April 2010 (hereinafter 'WARP Specification').

Consequently, the WARP PAG recommends that work on the Widget Access Request Policy should be continued without changes.

Introduction

The W3C Patent Policy sets a goal for Working Groups to produce specifications implementable under the W3C Royalty free terms. Patent Advisory Groups are W3C's response to a threat to this goal, and are created when such threats occur. They examine the situation, provide a platform for negotiations, and end with a recommendation to W3C and the Working Group. This report concludes the activities of the Widgets Updates PAG.

The W3C Web Applications (WebApps) Working Group is chartered to create specifications that enable improved client-side application development on the Web, including specifications both for application programming interfaces (APIs) for client-side development and for markup vocabularies for describing and controlling client-side application behavior. Apple Inc. participates in the Web Application Working Group. By participating in WebApps, Apple has given Royalty Free commitments according to the W3C Patent Policy on all Specifications

produced under the Webapps charter.

After having excluded Patents on 5 March 2009 on the Widgets Update Specification, the Widgets Update PAG concluded on 8 October 2009 that the Widgets Update Specification was covered by prior art. On 13 November 2009 Apple excluded US patent 7,743,336: *Widget security* and US Patent Application 20070101146: *Safe distribution and use of content* from the Work on Widget Access Request Policy.

Procedure

This section traces the necessary procedural steps followed by the patent exclusion and the creation of the Patent Advisory Group.

The WebApps Working Group is chartered until 30 June 2012. Part of its charter is to produce the Widget Access Request Policy Specification. The Widget Access Request Policy Specification had its First Public Working Draft published on 18 June 2009. This triggers a period of 150 days for exclusions, which ran until Friday, 15 November 2009. On Friday 13 November 2009 Apple Inc. excluded US Published Patent Application No. 11/432,295 and US Published Patent Application 11/409,276. On Saturday 14 November 2009, Apple Inc. issued additional information by informing W3C that said Applications had the Application Publication numbers US-20070101433 and US 20070101146 respectively.

The Web Application Working Group was correctly chartered. The First Public Working Draft of the Widget Access Request Policy Specification was correctly announced and triggered the above exclusion opportunity. Apple, as a Member of the WebApps Working Group, was entitled to an exclusion statement. Apple's exclusion statement as well as the complementary information was issued in time and is valid.

The Widgets Access Request Policy Patent Advisory Group was validly set up on 27 March 2009 according to the rules set forth in Section 7 of the Patent Policy and in conformance with the rules set forth in the Procedures for Launching and Operating a Patent Advisory Group. Apple did not attend any of the PAG's calls which is considered as unpleasant behavior.

In the meantime, the Application No. 11/432,295 turned into US Patent 7,743,336 ["Widget Security", the '336 patent] filed on May 10, 2006, and issued on June 22, 2010. The Patent combines several Applications and derives its priority from Application No. 60/730,956, filed October 27, 2005. Consequently, the WARP PAG shifted its review of claims to '336 patent,

On 8 July 2011 the PAG issued a call for prior art. Subsequently, the PAG received feedback on its archived public mailing list. The PAG created a page on prior art from the contributions received.

In conclusion: Apple made a valid exclusion and the PAG was created following the relevant rules. There were no issues with the procedure.

Conclusions

PAG Recommendations

Taking into account the wide variety of information made available to the PAG, the following recommendations are given:

1. The WebApps Working Group should continue to work on the Widget Access Request Policy Specification

Motivation

The '1146 Application

Given that the US Patent Application 20070101146: *Safe distribution and use of content* is not yet granted, it does not yet create absolute rights hindering the implementation of the Widget Access Request Policy Specification. But even if the WARP PAG makes the assumption that the US Patent Application 20070101146: *Safe distribution and use of content* would be granted as is, the WARP PAG believes that the technology described in the Application, if not covered by prior art, is completely orthogonal and does not apply to the Widget Access Request Policy Specification.

The '1146 Application claims do not read on the WARP Specification and here is why:

Claims of the '1146 Application and WARP Features

Claims of the '1146 Application	Evaluation against the WARP Specification
1. A method of distributing content, comprising: creating a cryptographic hash of at least a portion of content; creating a ticket file including the cryptographic hash; and distributing the ticket file to a user system.	• WARP does not make use of any cryptographic information. • WARP does not create a ticket file including the cryptographic hash. • WARP does not distribute a ticket file to a user system or anything to the system.
2. The method of claim 1, where creating a ticket file comprises: including in the ticket file information relating to downloading the content from a distribution site over the network; digitally signing the ticket file; receiving a request for the content from the user system; and sending the user system the ticket file separate from the content.	WARP does nothing relating to the above
3. The method of claim 1, further comprising: receiving notification from the user system that verification of the digital signature has failed; and initiating a security action in response to the notification.	WARP does not make use of digital signatures.
4. The method of claim 3, where initiating a security action comprises: initiating a revocation service.	WARP has no relation to initializing revocation services.
5. The method of claim 3, where initiating a security action	WARP does not rely in cryptographic methods to

comprises: preventing user systems from downloading the content.	prevent user systems from downloading content.
6. A method of downloading content from a network, comprising: sending a request for content to a content aggregator site; and receiving from the content aggregator site a digitally signed ticket file including a first cryptographic hash of at least a portion of content.	WARP does not download any content. WARP does not make use of any cryptographic hashes or any cryptography.
7. The method of claim 6, further comprising: downloading the content to a user system over a network, where the content is downloaded separately from the ticket file; and verifying the digital signature of the ticket file.	WARP has no relationship to digital signatures or ticket files.
8. The method of claim 7, where verifying the digital signature comprises: creating a second cryptographic hash of the downloaded content; comparing the second cryptographic hash with the first cryptographic hash; and if the hashes do not match, initiating a security action.	See above 7.
9. The method of claim 8, where initiating a security action comprises: deleting the content from the user system. WARP does not interact with the user's system.	WARP does not have the ability to demand that, or have any provisions that would require, content be deleted from the user's system.
10. The method of claim 8, where initiating a security action comprises: notifying the content aggregator site of a failed verification if the hashes do not match; and receiving a command for revoking the digital signature of the ticket file.	WARP has no means to contact other sites on failure.
11. The method of claim 8, where initiating a security action comprises: scanning the user system for instances of revoked content.	WARP most definitely does not have any provisions to scan the user's hard drive deleting content
12. The method of claim 1, further comprising: creating a security log on the user system that includes a history of content installation	WARP does not write anything to the user's hard drive or do any privacy violating things like keeping records of stuff on user's hard drives.
13. A computer-readable medium having stored thereon instructions which, when executed by a processor, causes the processor to perform the operations of: creating a cryptographic hash of at least a portion of content; creating a ticket file including the cryptographic hash; and distributing the ticket file to a user system.	WARP does not cryptographically do anything… specially nothing relating to content.
14. The computer-readable medium of claim 13, where creating a ticket file comprises: including in the ticket file information relating to downloading the content from a distribution site over the network; digitally signing the ticket file; receiving a request for the content from the user system; and sending the user system the ticket file separate from the content.	WARP does not cryptographically do anything… specially nothing relating to content.

15. The computer-readable medium of claim 13, further comprising: receiving notification from the user system that verification of the digital signature has failed; and initiating a security action in response to the notification.	WARP has nothing to do with digital signatures or events based on signatures failing of being verified.
16. The computer-readable medium of claim 15, where initiating a security action comprises: initiating a revocation service.	WARP has nothing to do with digital signatures or events based on signatures failing of being verified.
17. The computer-readable medium of claim 15, where initiating a security action comprises: preventing user systems from downloading the content.	WARP has nothing to do with digital signatures or events based on signatures failing of being verified. Hence this does not apply.
18. A computer-readable medium having stored thereon instructions which, when executed by a processor, causes the processor to perform the operations of: sending a request for content to a content aggregator site; and receiving from the content aggregator site a digitally signed ticket file including a first cryptographic hash of at least a portion of content.	WARP does not access content.
19. The computer-readable medium of claim 18, further comprising: downloading the content to a user system over a network, where the content is downloaded separately from the ticket file; and verifying the digital signature of the ticket file.	WARP does not access content. WARP does not download or deal with ticket files.
20. The computer-readable medium of claim 18, where verifying the digital signature comprises: creating a second cryptographic hash of the downloaded content; comparing the second cryptographic hash with the first cryptographic hash; and if the hashes do not match, initiating a security action.	WARP does not do any cryptographic checks.
21. The computer-readable medium of claim 20, where initiating a security action comprises: deleting the content from the user system.	WARP does not do any cryptographic checks. WARP does not do deletion of people's content.
22. The computer-readable medium of claim 20, where initiating a security action comprises: notifying the content aggregator site of a failed verification if the hashes do not match; and receiving a command for revoking the digital signature of the ticket file.	WARP does not do any cryptographic checks.
23. The computer-readable medium of claim 20, where initiating a security action comprises: scanning the user system for instances of revoked content.	WARP does not do any cryptographic checks. WARP does not do deletion of people's content.
24. The computer-readable medium of claim 18, further comprising:	WARP does not write log files about their

creating a security log on the user system that includes a history of content installation.	computer's content. WARP does not generate security logs.
25. A method of installing content, comprising: receiving content from a content distributor, the content including a first cryptographic hash of the content; creating a second cryptographic hash of the content; comparing the second cryptographic hash with the first cryptographic hash; and if the hashes do not match, initiating a security action.	WARP does not do this as it does not do any cryptographic checks.
26. The method of claim 25, further comprising: comparing the content to one or more lists of content to determine if the content can be safely installed.	WARP does not do this. it has nothing to do with content installation.

The '336 Patent

The PAG concentrated its main effort on the US Patent 7,743,336. It analyzed all input from various channels and listed them on the page for prior art. Additionally, the PAG evaluated the materials from the Patent application procedure[member only]. From this information, the PAG concluded that the claims in '336 patent which could potentially be read to cover the WARP Specification are anticipated by substantial prior art. Consequently, the PAG does not believe there are valid 'Essential Claims' against the WARP Specification.

The file wrapper reveals that a non-final rejection was first issued by the US PTO on December 5, 2008, to which Apple responded, and a notice of allowance was mailed on May 26, 2009.

Subsequently, on July 23, 2009, but before the patent actually issued, Apple submitted an 8-page Information Disclosure Statement listing 164 additional prior art references for the PTO to consider. On August 19, 2009, Apple disclosed 5 additional references; on September 14, 2009, 11 more; on September 22, 2009, 10 more; on October 16, 2009, 3 more; on October 20, 2009, 1 more; on October 23, 2009, 6 more.

On October 26, 2009, a second notice of allowance was mailed by the US PTO. Again before the patent issued, Apple submitted another Information Disclosure Statement on November 13, 2009, with 1 reference; on January 7, 2010, with 1 reference amending an earlier citation; and on February 4, 2010, with 3 references including a 50-page reference consisting of a VMware course handbook dated June 2005.

The PTO issued a final notice of allowance on February 25, 2010. The time that a patent examiner can allocate to reviewing documents disclosed in a patent application is very constrained. Some have reported that, without supervisor approval, a patent examiner is expected to dispose of a patent application in 14 hours. Apple's patent cites more material than W3C's own technical expert was able to review thoroughly in the first year of his PhD work, which was about "widgets". There is no possible way that the examiner can have reviewed each of these disclosed prior art references in the normal course of PTO business. We are faced with an issued patent with many more pages of disclosed references than pages of invention. Additionally, instead of helping this PAG in good faith and as encouraged by the Patent Policy FAQ Apple Inc. never sent a single message to the PAG thus undermining the intended social dialog between patent holder and Specification implementers.

We suggest, as explained further below, that the disclosure process engaged in by Apple may now actually work to its detriment. All of those references offer potential arguments for anticipation, thus invalidating the patent under 35 USC 102, or for obviousness under 35 USC

103, if Apple ever tries to assert it against implementations of the Widget Access Request Policy Specification.

Patent Claims:

Patent claim 1 in the '336 patent is:

- A widget security method, comprising:
- detecting a security event associated with a widget;
- generating data indicative of the security event;
- processing the data to determine a risk level associated with the detected security event,
- where the risk level is determined by an action selected from a group of actions consisting of
- reviewing information associated with the widget,
- comparing widget information with a user profile, and
- examining programming code associated with the widget;
- initiating a security action based on the risk level,
- where the detecting, generating, processing and initiating are performed by one or more processors of a hardware device.

The other independent claims of the '336 patent are claim 10, couched as a "computer readable media" containing a program that does what claim 1 states, and claim 19, couched as a "system" performing what is described in claim 1. These are three separate ways of describing essentially the same invention. Therefore, the PAG focused on claim 1.

What is a "Widget"

We cannot rely on the dictionary to explain what a "widget" is in the context of the '336 patent. The term originated in the early twentieth century to mean "a little device or mechanism, especially one whose name is unknown or forgotten" or "a hypothetical manufactured object, considered to represent the typical product of a manufacturer." [1] In common parlance used by members of the PAG, a "widget" resembles a "gizmo" or a "gadget" or a "thingy."

The term "widget" isn't much more specific in the software world. A "web widget", for example, is "a small application that can be installed and executed within a web page...." Wikipedia explains that other terms used to describe web widgets include "portlet," "gadget," "module," "snippet," and "flake."

The '336 patent specification itself defines widgets as follows:

"Generally, widgets are user interface elements that include information and one or more tools (e.g., applications) that let the user perform common tasks and provide fast access to information. Widgets can perform a variety of tasks, including without limitation, communicating with a remote server to provide information to the user (e.g., weather report), providing commonly needed functionality (e.g., a calculator), or acting as an information repository (e.g., a notebook).... Due in part to their simplicity, hundreds of widgets have been developed and publicly distributed."

The PAG experts assumed that a widget, in the context of the '336 patent, is a "software module" in the general sense. Some members of the PAG believe that such a broad patent claim would not be patentable under 35 USC 101 in light of Bilski and other recent court decisions.

Each of those "hundreds of widgets" is a candidate for prior art for the '336 patent.

Prior Art

The prior art disclosed by Apple in the '336 patent is generously broad. As described above, late in patent prosecution Apple cited early references to "Objects, Images and Applets"; "Three-Dimensional Widgets"; "Dashboard vs. Konfabulator"; " "Konfabulator & Widget Basics"; "Snippets Software"; "The OpenGL Graphics System" (a 334-page document!); "DesktopX" (72 pages!); "Portlet Communication"; and the "Microsoft Windows Graphical Environment User's Guide". The relevance of each of these documents to the '336 claim 1 quoted above is almost impossible to uncover.

Some of the prior art found by the PAG goes back as far as 1986. All prior art contributed to the PAG after the call for prior art was taken into account when creating the PAG's Recommendation. The PAG's focus was to find prior art that covers what the Widget Access Request Policy Specification does. Consequently this PAG report focuses on the fact that the Widget Access Request Policy Specification is covered by prior art rather than to see whether the technology excluded by Apple Inc. is valid IPR. Given the apparently broad meaning of "widget" in the patent, and the wealth of documentation about earlier portlets, gadgets, modules, snippets, desktops, konfabulators, and flakes, the PAG is convinced that the Widget Access Request Policy Specification is anticipated by much prior art.

The large amount of prior art is linked from the prior art page and from the '336 Patent itself. The report picks two to exemplify that the WARP Specification is coverd by prior art:

1. In March 1996 Netscape Navigator 2.0 was released. It included the "same-origin" policy for network access by Javascript applications. Netscape also supported the ability to bypass the same-origin policy using a Javascript API: netscape.security.PrivilegeManager.enablePrivilege("UniversalBrowserRead").

2. Macromedia Exact Domain Security: The Macromedia Flash Player uses exact domains when determining security requirements. A widget can only access content on the network from one domain. If a widget attempts to access content from two different domains, the controlling software will present a security dialog box requesting the user's permission. To access content across multiple domains, a widget must make use of a "policy file" which relaxes the security policy associated with the widget. An online document describes the extensive security changes that were added to Flash Player 7.0 in December 2003. In effect, this software detects a security event (widget attempting to access content in multiple internet domains); generates data that informs the control software of that event; the control software then processes the data to confirm the invalid request; and it initiates a security action (presents a security dialog box).

3. Dashboard and Konfabulator: Apple disclosed Dashboard and Konfabulator in its Information Disclosures, although it did not explain the relevance of those disclosures to the patent examiner. [2] According to Macworld Magazine, Dashboard is an Apple software product that was demonstrated by Steve Jobs on June 28, 2004, more than one year prior to U.S. Provisional Patent Application No. 60/730,956, filed October 27, 2005. [8] That demonstration showed widgets accessing the web, which means that widgets in 2004 were doing essentially what the W3C Widgets Access Request Policy ["WARP"] Candidate Recommendation describes today. Dashboard was a part of Apple's operating system, Mac OS X Tiger, for which third parties had written Konfabulator, "a JavaScript runtime engine for Mac OS X that lets you run files called Widgets." A Blogpost on June 29, 2004, described an argument between Apple and the authors of Konfabulator disputing which of them actually created these features first.

Disclaimer

Although portions of this PAG analysis were drafted by attorneys following review of the facts, none of the authors is your attorney. No part of this report is intended as legal advice either to W3C or to its members. It is intended merely as a summary of what the PAG has learned to date. Rely on this report entirely at your own risk. However, nothing should prevent even an attorney from expressing his or her personal opinions, and so this analysis includes the personal opinions of the authors.

With respect to W3C, the publication of the WARP Candidate Recommendation would not, by itself, be patent infringement of any patents owned by Apple or anyone else. Implementers and distributors of software products, though, are encouraged to read the analysis below, consult with their own attorneys, and form their own conclusions.

THESE RECOMMENDATIONS OF THE PATENT ADVISORY GROUP ARE NOT LEGAL ADVICE. NEITHER W3C NOR ANY OF THE PARTICIPANTS OF THE WIDGET UPDATES PATENT ADVISORY GROUP OR THEIR RESPECTIVE EMPLOYERS TAKES ANY RESPONSIBILITY FOR THE ACCURACY, LEGAL CORRECTNESS OR OTHER FITNESS FOR ANY PURPOSE OF THE INFORMATION PROVIDED IN THIS REPORT. ESPECIALLY, NEITHER W3C NOR ANY OF THE PARTICIPANTS OF THE WIDGET UPDATES PATENT ADVISORY GROUP OR ANY OF THEIR RESPECTIVE EMPLOYERS MAKE ANY REPRESENTATION THAT FOLLOWING THE RECOMMENDATIONS HERE WILL AVOID AN INFRINGEMENT OF THE US PATENT NR. 7,743,336 OR ANY OTHER PATENT MENTIONED IN THE REPORT OR THE PAGE ON PRIOR ART.

[1] Encarta® World English Dictionary[North American Edition] © & (P) 2009 Microsoft Corporation. All rights reserved.

[2] E.g., Guber, John et al., "Dashboard vs. Konfabulator", Jun. 2004, 9 pages.

Put in place by Rigo Wenning, WARP PAG Chair, adapting creative input by Marcos Caceres and Larry Rosen, last updated $Id: pagreport.html,v 1.3 2011/11/01 22:44:28 rigo Exp $