인터넷 이용환경 개선을 위한 NPAPI 대체 기술 안내서

2015.07

미래창조과학부

KISA 한국인터넷진흥원

차례 CONTENTS

표 목 차

그 림 목 차

Ⅰ. 개요

1. 배경

Netscape Plug-in API(이하 NPAPI)는 브라우저 내에서 외부 프로그램을 사용할 수 있도록 연결해주는 플러그인 기술로, 1995년 넷스케이프 브라우저에서 어도비가 PDF 파일 형식을 브라우저에서 다운로드하고 전용 뷰어를 통해 랜더링 할 수 있도록 개발된 확장 기술이다.

초기 NPAPI는 다음과 같은 기능을 지원하기 위해 사용되었다.

- 하나 혹은 그 이상의 MIME 타입 등록.
- 브라우저 윈도우의 특정 부분에 그리기.
- 키보드 및 마우스 이벤트 처리.
- URL을 이용해 네트워크로부터 데이터 얻기.
- URL들에 데이터 보내기.
- 새로운 URL로 링크되는 하이퍼링크나 핫스팟 추가하기.
- HTML 페이지의 섹션들에 그리기
- 네이티브 코드로 JavaScript/DOM과 통신하기.

NPAPI 지원 기능 중 현재 가장 많이 활용하고 있는 기술은 ActiveX와 유사하게 웹 페이지 내에서 컴파일 된 네이티브 코드를 직접 호출하거나, 브라우저와 OS에 설치된 응용 어플리케이션이나 서비스에 메시지를 연동하고 브라우저에서 사용자 입력 상태를 모니터링 하기 위한 용도로 활용되고 있다. 이러한 플러그인의 궁극적인 목표는 1. 네이티브 성능, 2. 네이티브 방식의 호환성 지원, 3. 샌드박스 보안 모델의 회피, 4. 운영체제 서비스와 메시지 연동을 목적으로 했으나, 3번, 4번의 기술적 문제점으로 인해 브라우저 개발사의 외면을 받았다.

NPAPI는 기존의 C나 C++ 로 구현된 네이티브 애플리케이션을 별도의 소스 변환 없이 브라우저 내에서 실행할 수 있는 호환성을 가지고 있으나, 보안과 안정성 측면에서 문제(멜웨어, 스파이웨어, 애드웨어 설치)와 더불어 코드를 더 복잡하게 만드는 주범으로 인식되고 있으며, 각

운영체제에 종속적인 컴파일 환경 때문에 NPAPI를 배포하고자 하는 모든 운영체제와 프로세서에 맞게 개발해야 하는 문제점(파편화)을 내포하고 있었다.

1990 중반에 개발된 NPAPI의 아키텍처는 기술적으로 최신 브라우저의 속도를 느리게 하고, 일부 오작동으로 브라우저의 안전성을 위협하고 있으며, 개인정보 탈취와 같은 보안 사고까지 일으키는 주범이 되었다. 이 때문에 Chrome, Firefox, Opera는 NPAPI 지원을 중단을 예고하고 있으며, Chrome의 경우 2015년 9월(Ver. 45) 중으로 대부분의 NPAPI 적용 서비스를 중단할 예정이다.

2. 목적

본 안내서는 국내 웹 개발자 및 웹 사이트 운영자가 NPAPI와 같은 비표준 기술을 개선하고자 할 때 참고할 수 있도록 주요 사용 현황, 브라우저 개발사의 정책, 대체 기술, 기술 별 특징, 구현 방법, 적용 시 고려사항 등에 대한 내용을 제공한다.

3. 요약

본 안내서는 NPAPI 를 사용하고 있는 전자결제, 보안, 인증, 게임 런처, 멀티미디어, PC 제어, 파일 처리, 전자 문서에 대한 실태조사 부문과 운영체제 및 브라우저 정책, NPAPI 대체기술로 구성되어있다. NPAPI 실태조사 결과에 대해 각 분야에 대해 비표준 기술을 사용하지 않고 기술 호환성과 상호운용성을 확보하는 방법 및 기술을 제시하고 있다.

비표준 기술 대체 방안은 최신 웹 표준(HTML5) 기술을 이용하는 방법, 브라우저 개발사의 전환가이드 적용 방법, 운영체제 기반의 응용 프로그램을 제공하는 방법, 브라우저 개발사 권고 확장 기술을 적용 방법, 브라우저 별 White list 신청하는 방법, 크롬 앱스토어의 IE Tab 앱을 사용하는 방법, 타 브라우저 사용 권고 등을 대안으로 제시하고 있다.

4. 적용환경

본 안내서에서 제시하는 방법은 기존 NPAPI가 지원하는 환경인 윈도우와 맥 OS 환경에서 시험되었다. 웹 서버는 윈도우 IIS와 리눅스 OS 기반의 아파치 웹 서버를 기준으로 하였으며, 클라이언트는 윈도우 및 맥 OS에서 동작하는 크롬, 파이어폭스 등에서 시험되었다.

HTML5 표준과 자바스크립트를 이용해서 구현된 기능은 HTML5를 지원하는 최신 브라우저에서만 동작하므로, 웹 개발자는 HTML5 표준 웹 기술을 이용하여 대체기술을 구현, 적용 할 경우 최신 브라우저의 버전과 http://html5test.com, https://www.modern.ie/ko-kr과 같은 표준 규격 지원 확인 사이트를 통해 서비스를 제공 하고자 하는 브라우저의 HTML5 기능 지원 여부를 확인해야 한다. 참고로 2015년 6월 기준으로 PC용 크롬 43, 파이어폭스 38 이상 버전에서 HTML5 표준 규격을 대부분 지원하고 있다.

5. 활용대상

본 안내서는 NPAPI 개발자, 웹 개발자와 웹 사이트 운영자, 웹 솔루션 개발자를 대상으로 하였다. 본 안내서에서 개발자는 PM(Project Manager), Front-End, Back-End, Desktop Application 개발자를 모두 포함하는 개념으로 정의하였다.

6. 문서의 한계

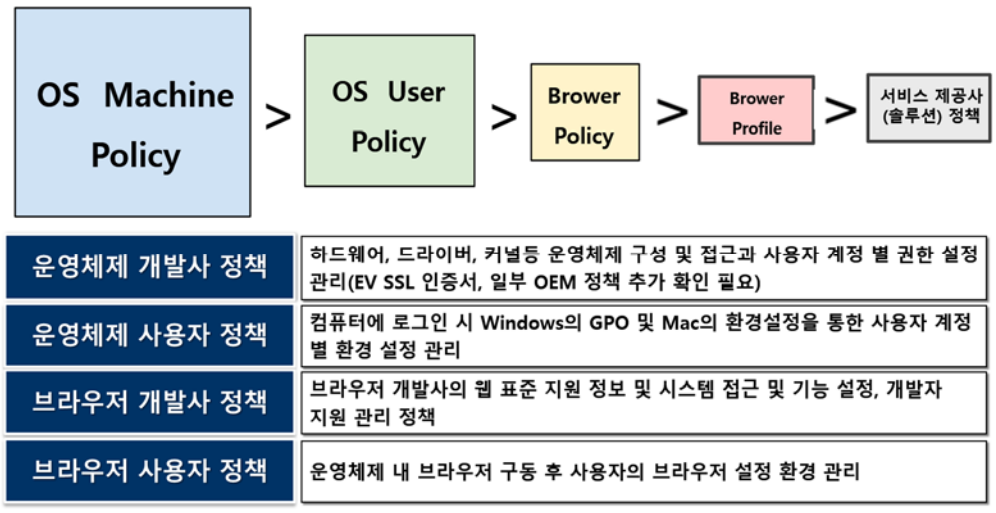
본 안내서에서 제시된 기술은 현장에서 사용되는 다양한 적용 사례 중 NPAPI에 대해 전환 가능한 일부 기술만을 제시하였다. 따라서 제시된 기술을 활용하여 상용 솔루션과 서비스를 구현 할 경우 법제도, 운영체제 및 브라우저 정책, 조직 내부의 서비스 정책, 대상 고객의 이용환경 등을 고려하여 수정, 보완해서 적용하여야 한다.

II. 멀티 운영체제 및 브라우저 정책

1. 멀티 운영체제 지원

운영체제 및 브라우저 개발사는 자사의 전략적 이해 관계에 따라 기술 지원 방식을 제공하고 있다. 운영체제와 브라우저가 결합된 방식으로 제공되고 있는 MS와 애플은 최근 자사 기술 정책과 로드맵에 따라 설치형 응용 프로그램 기술과 웹 기술을 상호 독립적으로 운영하는 정책을 유지하고 있다. 자사 운영체제 점유율이 낮거나 일부 플랫폼만 제공하고 있는 구글과 모질라 재단의 경우 높은 브라우저 점유율을 기반으로 윈도우와 맥 운영체제에 설치형 웹 애플리케이션을 구동하거나 브라우저 내에서 3D 게임과 멀티미디어 지원을 위한 다양한 확장 기술을 지원하고 있으며, 기존 윈도우와 맥 운영체제 설치 응용 프로그램과 브라우저 간 연동을 위한 브라우저 개발사의 확장 기술은 보안 취약점 및 이용자 서비스 선택권을 제약하는 한계로 인해 최근에는 브라우저 개발사의 정책에 따라 안전성이 보장된 웹 표준 기술로 개발하는 방향으로 전환하고 있다.

대부분의 웹 서비스(솔루션) 개발사는 운영체제 및 브라우저의 개발자, 사용자 적용 정책은 시스템 개발사 정책, 시스템 사용자 정책, 브라우저 사용자 정책을 인지하고, 가이드에 따라 개발 방법과 운영 방법을 관리해야 한다.



[그림 2-1] 운영체제 및 브라우저 정책 우선순위

Windows, Mac OS X, Linux등 운영체제에 적용할 수 있는 간단한 확장 기술은 현재까지 Custom URI Scheme(Protocol Handler 레지스터 등록)을 통한 브라우저와 운영체제 설치 응용 프로그램 간 파라미터 연동 방식이 대표적인 기술이다. (URI Scheme은 IETF RFC 4395 규격 참고) 최근엔 Custom URI Scheme을 통해 브라우저와 운영체제 설치 응용 프로그램 호출 후 직접 메시지를 교환하지 않고 HTTPS(SSL)을 통한 중계(서비스) 서버 연동을 통해 메시지를 암호화하고 인증서를 이용해서 메시지를 교환하는 방식을 사용함으로써 보안 취약점 문제를 일부 해결하고 있다.

<표 2-1> 운영체제 연동 기술 현황

지원 OS	브라우저와 응용 프로그램간 통신 방법	개발 지원 사이트
공통	IETF RFC 4395 표준 URI 정의	http://www.ietf.org/assignments/uri-schemes/uri-schemes.xml http://en.wikipedia.org/wiki/URI_scheme
Windows	Registering an Application to a URI Scheme	http://msdn.microsoft.com/library/aa767914(VS.85).aspx#prot_sec
Mac OSX	Launch Services Programming Guide	https://developer.apple.com/library/mac/documentation/Carbon/Conceptual/LaunchServicesConcepts/LSCIntro/LSCIntro.html#
Android	URL Scheme(Intent Scheme)	http://developer.android.com/training/basics/intents/filters.html
iOS	Custom URL Scheme 연동	https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Inter-AppCommunication/Inter-AppCommunication.html#//apple_ref/doc/uid/TP40007072-CH6-SW2

2. 브라우저 서비스 개발 정책

브라우저 개발사들은 자사 기술 전략에 따라 다양한 확장 기술을 제공하였으나 보안 취약점과 피싱과 같은 문제가 지속적으로 발생하여 최근에는 Sandbox 보안 모델을 준수하는 브라우저 별 확장 기술이나 HTML5 기반의 웹 표준 기술을 권고하고 있다.

<표 2-2> 브라우저 개발자 지원 정책 현황

브라우저	개발 지원 사이트
Microsoft Edge for Windows 10 Developer Guide	https://msdn.microsoft.com/en-us/library/dn997183(v=vs.85).aspx
Chrome Program Policies	https://developer.chrome.com/webstore/program_policies
Mozilla AMO Policies Mozilla Add-on SDK	https://developer.mozilla.org/en-US/Add-ons/AMO/Policy https://developer.mozilla.org/en-US/Add-ons/SDK
Safari for Developers	https://developer.apple.com/safari/

<표 2-3> 브라우저 별 확장 기능 지원 사이트

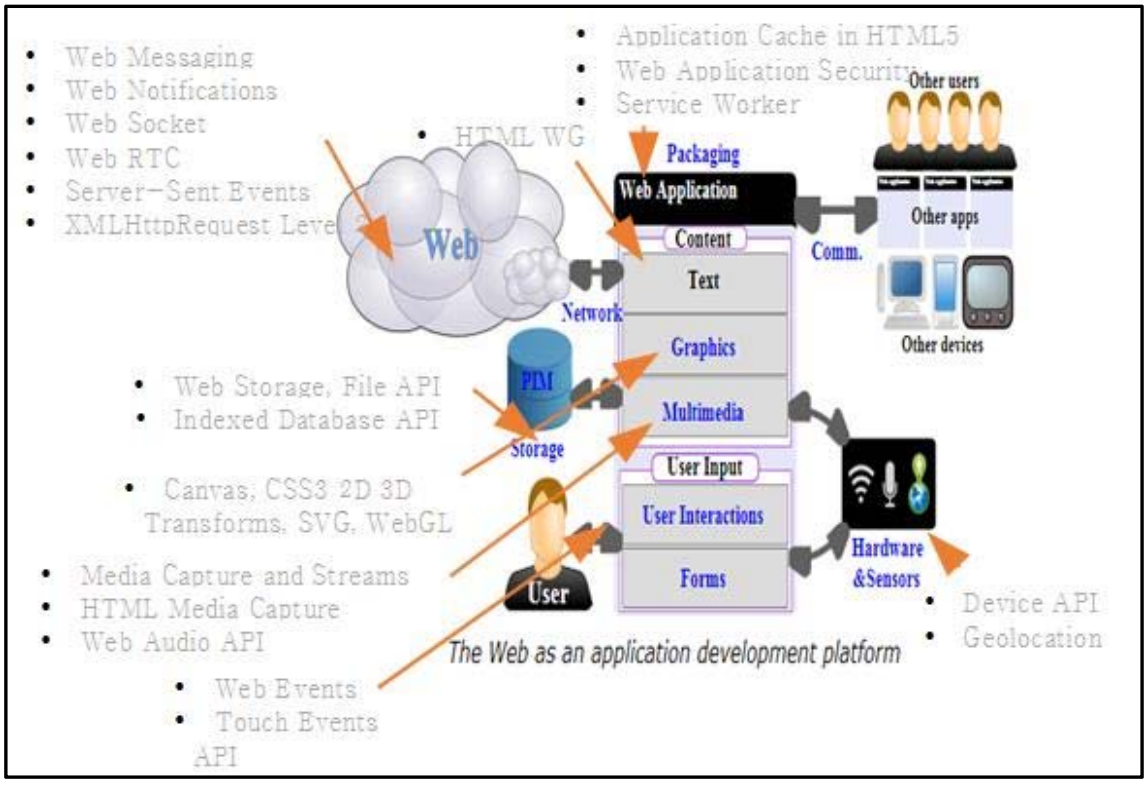
브라우저	확장 기능 개발 지원 사이트
Microsoft Internet Explorer Extension	https://msdn.microsoft.com/en-us/library/hh772404(v=vs.85).aspx
Chrome Extension	https://developer.chrome.com/extensions
Mozilla Add-on SDK	https://developer.mozilla.org/en-US/Add-ons/SDK
Safari Extension	https://developer.apple.com/library/safari/documentation/Tools/Conceptual/SafariExtensionGuide/Introduction/Introduction.html
Opera Extension	https://dev.opera.com/extensions/

3. 멀티브라우저 지원 방안

3.1 HTML5

HTML5 는 웹 서비스나 크롬 앱, WinJS 기반 웹 애플리케이션을 만들기 위한 HTML 표준 규격으로 2014.10 월 표준이 완료됐다.

HTML5 는 기존 비표준 ActiveX, NPAPI 플러그인 기술을 활용한 별도의 플러그인 프로그램(FlashPlayer, Java)의 설치 없이 브라우저만으로 멀티미디어, 게임, 오피스, 그래픽, 그리드, 챠트 등을 구현할 수 있는 기술로 아래 그림과 같이 마크업 이외에도 다양한 스크립트, API 기술 요소를 포함하고 있다.



[그림 2-2] 웹 애플리케이션 개발을 위한 연관 표준 기술

HTML5 표준만으로 웹 애플리케이션을 개발하는데 한계가 있으며 자바스크립트 기반의 API 나 전문적인 분야의 기능을 제공하는 라이브러리를 활용하여 개발해야 한다.

<표 2-4> HTML5 주요 기능

주요기능	설명	관련 W3C 표준명
웹 폼 (Web From)	이용자의 입력정보를 받기 위해 사용되는 입력형태에 대한 정의에 사용되는 마크업, 애트리뷰트와 이벤트	HTML5
Canvas	웹에서 즉시모드(immediate mode)로 2차원 그래픽을 그리기 위한 API와 <canvas>내 각종 객체를 회전, 변환하고 이미지 생성 등 각종 효과를 주는 기능에 대한 API	Canvas 2D API HTML Canvas 2D Context
SVG (Scalable Vector Graphic)	XML 기반의 2차원 벡터 그래픽을 표현하기 위한 언어	HTML5
Video/Audio	<video>는 비디오 또는 영화를 보여주기 위해 사용되는 미디어 element이며, <audio>는 사운드나 오디오 스트림을 표현하기 위한 미디어 element	HTML5
Geolocation	디바이스의 지리적 위치 정보를 제공하는 API 표준	Geolocation API
오프라인 웹 애플리케이션 (Offline Web Application)	인터넷 연결이 지원되지 않는 경우에도 웹 애플리케이션이 정상적으로 수행될 수 있도록 지원하는 기능으로 애플리케이션에 대한 캐시와 데이터에 대한 캐시로 구성	HTML5
로컬저장소 (Local Storage)	기존의 쿠키의 기능을 개선하기 위한 목적으로 개발된 기능으로 웹 클라이언트에서 키와 값이 쌍으로 구성된 데이터를 영구적으로 저장하는 기능	Web Storage Indexed Database API
Service Work	페이지나 사용자 인터랙션이 필요하지 않은 기능들을 위한 기회를 제공하고, 웹 페이지와는 별개로 자바스크립트 내에서 **register** 등록하고, 브라우저에 의해 백그라운드에서 실행되는 스크립트	Service Worker
File API	클라이언트에 있는 파일 선택 및 데이터 추출	File API
Web RTC	비디오채팅 기능과 P2P 데이터 공유 지원	Web RTC
웹 소켓 (Web Socket)	웹 애플리케이션이 서버 측의 프로세스와 직접적인 양방향 통신을 위한 API	Web Socket API
웹 워커 (Web Worker)	웹 애플리케이션을 위한 쓰레드(Thread) 기능에 대한 API	Web Workers
3D 그래픽 라이브러리 (WebGL)	WebGL은 웹 기반의 그래픽 라이브러리로 자바스크립트 프로그래밍 언어를 통해서 사용할 수 있으며 호환성이 있는 브라우저에서 인터랙티브 한 3D 그래픽을 사용할 수 있는 API	WebGL

※ HTML5 표준에 대한 상세 내용은 방송통신표준 "HTML5 웹 애플리케이션 개발 지침", TTA 웹 표준 참조

가. HTML5 웹 애플리케이션 API 구현 방법

HTML5 웹 애플리케이션은 개발자가 제공하는 다양한 구현 방법에 따라 웹 문서의 컨텍스트를 실행한다

- 스크립트 요소의 처리를 통해 구현한다.
- 인라인 'javascript: URL' 의 실행(예 : img 요소의 src 속성이나 CSS style 요소 블록의 @import 규칙)
- addEventListener()를 이용해서 DOM 에 등록, 명시적인 이벤트 핸들러 내용 속성을 사용하거나 IDL 속성 이벤트 핸들러를 사용하여 구현한다.
- 고유의 스크립트 기능을 가진 XBL, SVG 같은 기술로 구현한다.
- W3C Web Application WG 은 지속적으로 규격이 업데이트 되고 있으므로, 관련 표준 규격 확인 후 서비스에 구현해야 한다.

3.2 Custom URI Scheme

Custom URI Scheme 은 특정 응용 프로그램을 가리키는 고유의 지시어이다. 이 규격은 브라우저에서 다른 응용 애플리케이션을 호출 하거나 반대로 응용 프로그램에서 브라우저의 특정 페이지를 호출할 수 있다. 다만 URI Scheme 을 처리하는 응용 프로그램은 신뢰할 수 없는 메시지(악의적인 메시지)를 수신 할 수 있기 때문에, 대부분의 URI Scheme 방식으로 호출하는 응용 프로그램들은 이용자 계정 식별이 가능한 OS 기본 탑재 애플리케이션에서 주로 사용한다.

만약 웹 서비스 개발사가 Custom URI Scheme 를 통해 특정 응용 프로그램과 연동하도록 개발할 경우 보안 서버를 경유해서 메시지를 전달하는 방식과 같이 악의적인 데이터에 대응할 수 있는 방법을 고려하여 개발해야 한다.

가. Custom URI Scheme 등록

정의된 URI 체계를 처리 할 수 있는 응용 프로그램을 등록하려면 윈도우의 경우 HKEY_CLASSES_ROOT 에 대한 적절한 하위 키와 값과 함께, 새로운 키를 추가 해야 하며, 루트 키가 추가되는 URI 체계와 일치해야 한다. 아래는 HKEY_CLASSES_ROOT 에게 경고 키를 추가하는 예제이다.

```
HKEY_CLASSES_ROOT
    alert
        (Default) = "URL:Alert Protocol"
        URL Protocol = ""
        DefaultIcon
            (Default) = "alert.exe,1"
        shell
        open
        command
        (Default) = "C:₩Program Files₩Alert₩alert.exe" "%1"
```

[예시 : HKEY_CLASSES_ROOT에게 경고 키 추가]

나. Custom URI Scheme 등록

시스템 Shell 에 등록된 URI Scheme 은 아래와 같은 방식으로 호출되어 사용된다.

- 웹 페이지에서 하이퍼링크 클릭 시 URI Scheme 이 System 에 전달됨
- System 에 전달된 Custom URI Scheme 을 보고 실행 가능한 응용 프로그램이 있는지 확인
- 해당 URI Scheme 을 받을 수 있는 응용 프로그램이 있다면 실행시키고, URL 에
  포함된 값을 함께 전달한다.
- 응용 프로그램이 실행되면서 URL 에 포함된 값을 참고해서 사전에 정의된 특정 기능을 수행한다.
- 별도 호출 후 특정 시간까지 반응하지 않을 경우를 대비해 고객 통보나 설치 유도와 같은 사전 정의된 예외처리를 구현해야 한다.

```
<script>
var appstoreUrl = "http://itunes.apple.com/kr/app/id393499958?mt=8";


//url 은 "naversearchapp://search?qmenu=voicerecg&version=1"
function onClickApp(url) {
   var clickedAt = +new Date;
   naverAppCheckTimer = setTimeout(function() {
        if (+new Date - clickedAt < 2000){
            if (window.confirm("네이버앱 최신 버전이 설치되어 있지 않습니다.₩n설치페이지로 이동하시겠
습니까?"))
```

```
            {
            location.href = appstoreUrl;
          }
        }, 1500);
    }
}
location.href = url;
</script>
```

[예시 : Custom URI Scheme 예외처리]

다. Custom URI Scheme 활용 시 유의사항

정의된 URI Scheme 에 의한 특정 응용 프로그램 호출 및 메시지 전달 시 악의적인 추가 명령 줄 매개 변수를 전달하기 위해 추가 따옴표 또는 백 슬래시 문자를 사용할 수 있다. 이는 Protocol Handler 를 통해 메시지를 위변조하여 시스템에 악성코드나 바이러스 설치를 유도하는 방법으로 사용할 수 있어 외부 데이터에 의해 호출 메시지가 위변조 될 수 있는 상황에서는 가급적 URI Scheme 을 사용하지 않는 것이 바람직하다.

또한 URL 파라미터 위변조를 방지하기 위해 보안 서버를 통한 HTTPS(SSL) 통신을 고려해야 한다

3.3 브라우저 별 확장 기술

브라우저 개발사들은 자사 기술 전략에 따라 다양한 확장 기술을 제공하고 있으나 보안 취약점과 피싱과 같은 문제가 지속적으로 발생하여 최근에는 Sandbox 보안 모델을 준수하는 검증된 확장 기술이나 HTML5 웹 애플리케이션 기술을 권고하고 있다.

<표 2-5> 브라우저 별 확장 기술 요약

	플러그인 기반 기술	플러그인 기술	웹앱 기술
인터넷 익스플로러 (MS)	ActiveX (Edge Deprecated)	· Silverlight(Deprecated 예정) · Java 애플릿 플러그인 · Flash Player 시스템 플러그인 · Custom URI Scheme 연동 (Protocol Handler 레지스터 등록)	· HTML5 · WinJS · asm.js (WebAssembly)
크롬 (구글)	PPAPI(Bride) Native Client Portable Native Client NPAPI(45 Ver. Deprecated)	· Native Messaging · 커스텀 URI Scheme 연동 (Protocol Handler 레지스터등록) · Legacy Browser Support(옵션)	· HTML5 · Chrome App(IE Tab) · Chrome Web API · Chrome JavaScript API · asm.js (WebAssembly)
파이어폭스 (모질라)	NPAPI(Deprecated 예정) Native Client	· Java 애플릿 플러그인 · Flash Player 시스템 플러그인 · Custom URI Scheme 연동 (Protocol Handler 레지스터등록)	· HTML5 · Add-on SDK(Low Level APIs) · asm.js (WebAssembly)
사파리 (애플)	NPAPI(Deprecated 예정)	· Java 애플릿 플러그인 · Custom URI Scheme 연동 (Protocol Handler 레지스터 등록)	· HTML5 · asm.js (WebAssembly)
오페라	NPAPI	· Java 애플릿 플러그인 · Flash Player 시스템 플러그인 · Native Client	HTML5

브라우저 별로 아래와 같이 NPAPI 를 대체할 수 있는 플러그인 기술 및 확장 기술 사용한다.

- 구글 : 네이티브 클라이언트(NaCl, PNaCl), 설치형 웹앱, 네이티브 메시징 API 등의 사용을 권장하고 있다.
- 모질라 : WebGL, Web Sockets, Web RTC, asm.js 등을 활용 할 것을 권고하고 있다.
- 애플 : 기존 NPAPI 를 사용하지만 보안 충돌을 고려하여 사용을 최소화하고 HTML5 나 앱 방식으로 개발할 것을 권장하고 있다.
- 오페라 : 기존 NPAPI 를 사용하지만 보안 충돌을 고려하여 사용을 최소화하고 HTML5 나 PPAPI 로 개발할 것을 권장하고 있다.
- 마이크로소프트 : Edge 버전부터 HTML5 표준 사용을 권장하고 있다.

관련 대체 기술과 관련한 구체적인 내용은 아래 URL 을 통해 확인 가능하다.

- Browser Extension : http://en.wikipedia.org/wiki/Browser_extension
- Chrome Web API : https://developer.chrome.com/extensions/api_other
- NPAPI : http://en.wikipedia.org/wiki/NPAPI
- NaCl(Native Client) : https://developer.chrome.com/native-client
- WinJS : http://dev.windows.com/en-us/develop/winjs
- arm.js : http://asmjs.org

NPAPI 는 샌드박스를 거치지 않고 동작하여 보안 문제가 발생할 경우 접속 통제 및 사용 제한 없이 브라우저 강제 종료 및 사용자 시스템에 문제를 발생할 가능성이 있어 크롬, 파이어폭스, 오페라, 사파리에서 사용을 제한하거나 지원을 중단하고 있다.

이에 따라 각 브라우저 개발사 및 웹 서비스 개발사는 NPAPI 를 WebGL, Canvas, Web Socket, Web Worker, Video/Audio, Adaptive Streaming, Web RTC, WebVTT 와 같은 웹 표준 기술로 대체하고 있다.

Native Client (NaCl) **Portable Native Client(PNaCl)**	· 웹 페이지에서 운영체제 호환성 및 안전성을 유지하기 위해 브라우저에서 컴파일 된 Native 코드를 운영하는 오픈소스 기술 · NaCl과 PNaCl의 차이는 배포 가능한 채널과 Translator 필요 여부
PPAPI(Pepper Plugin API)	· NaCl에서 시스템 레벨의 기능을 지원할 수 있도록 하기 위한 API 기능 제공(Plugin registry 지원, 3D 랜더링을 위해 사용)
Native Messaging	· 브라우저 상에서 실행되는 웹과 Native 애플리케이션 사이에 메시지 전달 기능
Custom URI Scheme	· Custom URI Scheme은 특정 Native 애플리케이션이 제공하는 기능에 따라 브라우저에서 다른 Native 애플리케이션을 호출하거나 반대로 Native 애플리케이션에서 브라우저의 특정 페이지를 호출할 수 있는 기능
Legacy Browser Support	· 엔터프라이즈 내부의 Chrome 브라우저에서 ActiveX 등 특정 브라우저 기술이 필요한 웹 사이트에 액세스 하는 경우 자동으로 Chrome에서 해당 브라우저로 이동하여 열릴 수 있도록 정책을 설정하는 기능

[그림 2-3] 크롬 지원 NPAPI 대체 플러그인 기술

하지만 국내의 경우 NPAPI 를 공인인증서 관리를 위한 전자결제, 보안, 인증과 함께 파일처리, 멀티미디어, 전자문서, 게임 런처 등에 광범위하게 사용하고 있으며, 전자결제, 보안, 인증 등은 기존 기술과의 호환성, 법제도 및 인증 정책 등의 문제로 일부 기능에 대해 NPAPI 플러그인 대체 웹 표준 기술로 대체가 불가능하다.

Ⅲ. NPAPI 대체기술

1. NPAPI 대체기술 개요

브라우저 개발사들은 점진적으로 비표준 플러그인 기술(ActiveX, NPAPI)들을 아래와 같은 사유로 2 년전부터 폐지하거나, 대체 방안을 제시하고 있으며, 특히 구글 크롬 브라우저에서 지원되던 NPAPI 의 경우 45 버전부터 현재까지 유지하던 수동 설정 기능까지도 지원을 중단할 예정이다.

구글의 NPAPI 지원 중단 사유는 다음과 같다.

- 보안: 플러그인은 일반적으로 샌드박스를 거치지 않는 Low Level 언어(C, C++)로 웹 브라우저의 CPU 자원을 남용하는 원인으로 지목되어 왔다.
- 안정성: 플러그인 일반적으로 Low Level 코드로 시스템에 접근함으로써 웹 사이트 강제 종료 및 수동 종료를 유도하는 주된 원인 이였다.
- 성능: NPAPI 사용 웹 콘텐츠들은 그래픽 가속화등 구글의 최적화 기능을 사용할 수 없어 더 많은 전력을 소모하며, 전반적으로 브라우저의 성능을 저하시킨다.
- 호환성: NPAPI 는 전체적인 브라우저와 통합되지 않으며, 현재 급속도로 성장 중인 모바일 웹에서 지원하지 않는다.
- 사용량: 구글이 NPAPI 지원 중단을 시작하기 전부터 플러그인 사용량은 급속도로 감소하고 있으며, 지금 가장 자주 쓰이는 플러그인 조차 실행률은 1% 미만이다.
- 개발자 개발 부담: NPAPI 에 대한 지속적인 지원은 크롬 코드 베이스의 복작성을 증가시키고, 과도하게 개발 부담을 개발자에게 전가하고 있다.

가. 구글의 웹 사이트 및 개발자 지원 계획

구글은 이전에 발표한 "*NPAPI 중단 관련 개발자 안내서" 이외에 NPAPI 기능을 대체할 수 있는 최신 웹 기술에 대한 별도 계획은 없다. 특정 개발자 집단을 위한 개발 지원 계획이나 실행 프로그램을 제공하지도 않는다. 기본적으로 NPAPI 대체 기술과 관련해서는 웹 서비스 개발사가 자사 서비스 정책에 맞는 다른 기술을 적용하는 것을 추천하고 있다. 또한 스택오버플로워(stackoverflow) 같은 개발자 사이트 정보를 참고하는 것을 권하고 있다.

*https://www.chromium.org/developers/npapi-deprecation

나. 구글의 NPAPI 중단 관련 개발 안내서 요약

NPAPI 가 차단됨에 따라 기존에 각 운영체제 별 NPAPI 플러그인에 의존하던 기능을 실행할 수 있게 해주는 최신 기술에 관한 일부 개발자의 문의에 따라 일반적인 NPAPI 사용 사례와 표준 웹 기술 기반 대안 방안을 제공한다.

일반적으로 주요 웹 표준 기반 기술(HMTL. CSS, JS)은 대부분 샌드박스 내부에서 클라이언트 소프트웨어 개발에 적합하다. 만약 웹 샌드박스 외부의 기능에 대한 접근을 필요로 하는 경우 무수히 많은 *크롬 확장 기능과 **App API 가 제한적인 OS 기능에 대한 접근을 제공합니다.

2015 년 4 월부터 크롬은 NPAPI 지원을 기본설정으로 비활성화하고 있으며, 크롬 웹 스토어에서 NPAPI 플러그인을 필요로 하는 확장 프로그램을 삭제하고 있다. 사용자 및 기업들에게 임시로 NPAPI 를 사용할 수 있는 옵션으로 주소창에 chrome://flags/#enable-npapi 기입을 통해 NPAPI 플러그인을 사용

설정을 변경을 통해 사용할 수 있다. 그러나 2015 년 9 월 이후에는 강제 수단을 통해 NPAPI 지원 기능을 영원히 삭제할 예정이다.

* https://developer.chrome.com/extensions

** https://developer.chrome.com/extensions/api_index

다. 브라우저 개발사 추천 확장 기술

브라우저 별로 아래와 같이 NPAPI 를 대체할 수 있는 플러그인 기술 및 확장 기술 사용한다.

- 구글: 네이티브 클라이언트(NaCl, PNaCl), 설치형 웹앱, 네이티브 메시징 API 등의 사용을 권장하고 있다.
- 모질라: WebGL, Web Sockets, Web RTC, asm.js 등을 활용 할 것을 권고하고 있다.
- 애플: 기존 NPAPI 를 사용하지만 보안 충돌을 고려하여 사용을 최소화하고 HTML5 나 앱 방식으로 개발할 것을 권장하고 있다.
- 오페라: 기존 NPAPI 를 사용하지만 보안 충돌을 고려하여 사용을 최소화하고 HTML5 나 PPAPI 로 개발할 것을 권장하고 있다.
- 마이크로소프트: Edge 버전부터 HTML5 표준 사용을 권장하고 있다.(기존 IE 에서는 ActiveX 지원)

<표 3-1> NPAPI 대체 기술

기술 별 NPAP 적용 부분	세부기술	표준 대체 기술
비디오 및 오디오	기본 구성 요소	• HTML5 Media Elements : HTML5 표준은 <audio>, <video> 엘리먼트를 제공하며 대부분은 경우 <canvas>와 함께 사용 (예 : Video FX Chrome Experiment) • WebRTC : Peer간 실시간 통신을 위해 설계되었으며 실시간 미디어, 데이터의 라이브 스트리밍 시 사용 가능
	Adaptive Streaming (DASH	• Silverlight의 Smooth Streaming 혹은 QuickTime의 HTTP Live Streaming과 같이 모던 웹에서 각각의 사용자들에게 반응적으로 스트리밍 할 수 있는 기술 • HTML Media 엘리먼트의 Media Source Extension으로 구현 가능
	화상회의	• WebRTC 에서 제공되는 JavaScript API를 이용하여 구현 가능 (예 : Cube Slam Chrome Experiment)
	DRM (Digital Rights Management)	• EME(Encrypted Media Extensions) 표준으로 HTML5 video에 DRM 적용 가능하며, 공개 코덱인 WebM을 이용해서 비디오 구성 요소의 미디어 확장 기능을 사용할 수 있음
	폐쇄 자막	• WebVTT와 <video> 태그의 하위 엘리먼트인 <track> 구성요소를 이용하여 HTML 앱에서 시간 제한 텍스트 자막 기능을 추가할 수 있음
네이티브 어플리케이션과의 커뮤니케이션	-	• 크롬 앱 및 크롬 확장 Native Messaging API를 사용할 수 있음
게임 및 3D	-	• Native Client(NaCl)의 크로스 플랫폼 게임 개발 환경 사용 • WebGL 표준으로 3D 그래픽 가속 기능 사용
보안	-	• TLS 로 전환을 권장하며 추후에는 Web Crypto 표준 사용을 권장
하드웨어 접근	-	• 웹캠, 마이크 등과 같은 사용자 PC의 외부 장비에 대한 접근을 Media Capture 표준으로 구현 가능 • USB 하드웨어에 대한 액세스 및 블루투스 장치를 액세스하기 위한 응용 프로그램 API(chrome.usb, chrome.bluetooth) 사용
화면 캡쳐	-	• Desktop Capture 를 이용하여 전체 화면 캡쳐 가능 • Tabs API의 captureVisibleTabs를 이용하여 개별 탭 캡쳐 가능

2. NPAPI 국내 사용 현황 및 분류

국내 민간 주요 200 대 웹사이트 중 78 개 웹사이트에서 241 개의 NPAPI 를 사용하고 있으며, 주요 사용 기능은 결제, 보안, 인증, 게임실행, 멀티미디어, 파일 처리, PC 제어(바로가기 설치, 원격 제어), 전자문서(위변조 방지, 시점 확인, 출력)에 사용되고 있다. 분야별로는 포털이 NPAPI 를 가장 많이 사용하고 있으며, 결제, 보안, 인증 기능을 많이 사용하는 금융 분야가 두번째로 많이 사용하고 있다.

NPAPI 를 사용하는 사이트 대부분이 10 개 이하의 NPAPI 를 사용하고 있으며, 직접 NPAPI 를 개발하기보다는 웹 솔루션에 기능이 포함되어 사용하는 경우가 대부분이다.

<표 3-2> 국내 NPAPI 사용 현황

구분		민간 200대 사이트 NPAPI 사용 수	
기능별 NPAPI	결제(전자서명, 전자결제)	123	51.0%
	보안(백신, 방화벽, 로그인 보안, 키보드 보안)	56	23.2%
	게임(게임 실행)	26	12.8%
	인증(본인 확인, 시점 확인)	13	10.8%
	멀티미디어(그래픽편집, 동영상, 음원 재생)	9	3.7%
	파일 처리(대용량 다중 파일 처리, 고속 파일 다운, 업로드)	8	3.3%
	PC제어(시스템 정보 확인, 장치 연동 관리)	6	2.5%
	전자문서(문서 보호, 암호화, 출력)	-	
	기타(채팅 상담)	-	
합계		241(100%)	

출처 : 한국인터넷진흥원, NPAPI 사용현황 실태조사 결과

<표 3-3> 국내 NPAPI 사용 현황 및 분야

순위	구분	기능	NPAPI 명	업체명	중복수	비중
1		WP 업데이트	SandBox Plugin	think-bowl	41	17%
2	결제	전자결제	INISAFE CrossWeb NP Plugin	이니텍	30	12%
3		전자결제	INICIS INIPay Plugin	이니시스	16	6.6%
4	보안	플러그인 통합설치	Veraport Mozilla Plugin	베라포트	12	4.9%
5	결제	전자결제	KCP	한국사이버결제	12	4.9%
6		전자결제	LG Uplus XPay Plugin (npRuntime)	LG U+	11	4.5%
7	보안 관리	키보드보안	TouchEnKey for Multi-Browser	라온시큐어	10	4.1%
8		백신	AhnLab Online Security	안랩	8	3.3%
9		개인 방화벽	nProtect Netizen v5.5 Install.	잉카인터넷	8	3.3%
10	인증	인증서 관리, 암호화	SoftForum XecureWeb Control Plug-in	소프트포럼	7	2.9%
합계					155	64.3 %

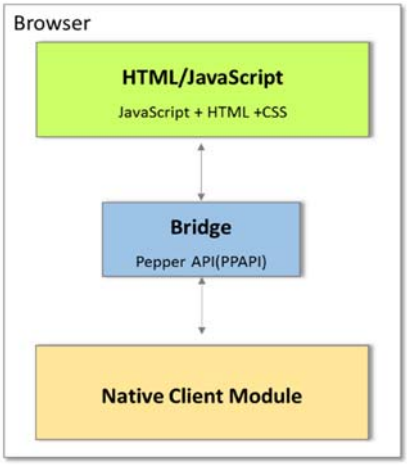
<표 3-4> 크롬 플러그인 실행 현황

	2013.09	2014.05	2014.10
Sliverlight	15%	13.3%	11%
Google Talk	8.7%	8.7%	7%
Java	8.9%	7.2%	3.7%
Facebook	6%	4.2%	3.0%
Unity	9.1%	3.1%	1.9%
Google Earth	9.1%	0.1%	0.1%

3. NPAPI 대체 기술

가. Native Client(PPAPI)

Native Client(이하 NaCl)는 웹 페이지에서 이용자가 기대하는 운영체제 호환성 및 안전성을 유지하기 위해 브라우저에서 컴파일된 네이티브 코드를 운영하는 오픈소스 기술이다. Native Client 를 이용하면 자바스크립트가 아니어도 웹 프로그래밍을 할 수 있기 때문에 개발자는 자신이 좋아하는 언어를 이용해 Windows, Mac, Linux 기반 Chrome 브라우저에서 C, C++로 Native Client 웹 애플리케이션을 개발할 수 있다.

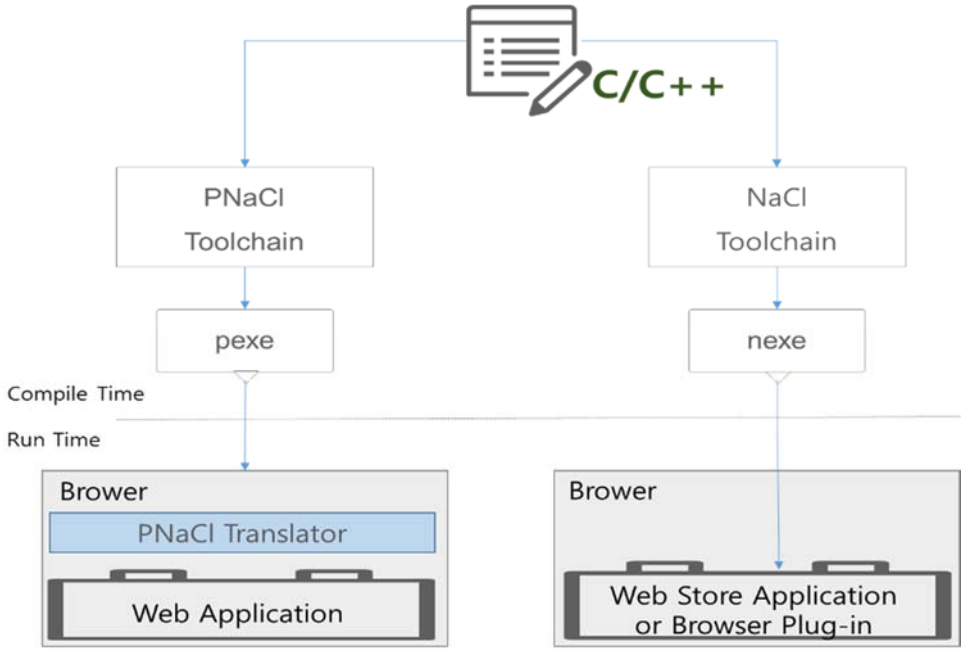


[그림 3-1] NaCl 동작 다이어그램

[그림 4-1]과 같이 브라우저의 자바스크립트 코드와 Native Client 모듈의 C 또는 C++ 코드 사이에 양방향 통신을 할 수 있다. 브라우저는 Pepper 플러그인 API(PPAPI)를 통해 Native Client 모듈에 메시지를 전송할 수 있고. Native Client 모듈은 자바스크립트의 메시지에 응답할 수도 있다. 반대로 Native Client 모듈이 자바스크립트에 먼저 메시지를 보낼 수도 있다. 이 경우 모든 통신은 비동기적으로 수행된다. 즉, 메시지를 전송해도 시스템은 응답을 기다리지 않는다. 이러한 동작은 클라이언트가 서버에 메시지를 게시하고 즉시 반환하는 클라이언트/서버 웹 통신과 유사하며, Native Client 메시징 시스템은 PPAPI 의 일부로서 작동한다.

그림과 같이 Sandbox 내 Native Client 모듈(.nexe 파일)은 <embed> 태그를 통해 웹페이지에 로드 되면 .html 페이지와 .nexe 파일이 웹 애플리케이션을 정의한다. Native Client 런타임 시스템은 시스템 리소스가 영향이 없도록 다음과 같이 안전하지 않은 활동은 차단한다.

- 기기 또는 파일 직접 조작(대신 웹 표준 기반 파일 시스템 API 가 제공됨)
- 운영체제에 직접 액세스, 보호가 된 메모리에 쓰기와 같은 코드의 목적을 숨기기 위해 자체 수정한 코드 이용



[그림 3-2] NaCl과 PNaCl 비교

Portable Native Client(이하 PNaCl)는 다른 브라우저와 호환성과 구글 웹앱 스토어에 게시하지 않고 운영체제에 배포할 수 있는 이식성 문제를 해결하기 위한 모듈로 PNaCl 은 NaCl 과 동일한 수준의 보안을 지원한다.

PNaCl 은 (IE, FireFox, Opera, Chrome)뿐만 아니라 Intel/AMD 의 X86 아키텍처 장비(PC 및 노트북)과 ARM 기반의 모바일, 태블릿 등 모든 곳에서 다 구동이 가능하다.

브라우저에서 실행되는 기존 웹앱을 NaCl(PNaCl)을 이용하여 확장하는 사례는 다음과 같다.

• 기존 컴포넌트 재사용: NaCl 은 네이티브 언어(현재 C 와 C++)를 지원하기 때문에 웹앱에서 기존의 소프트웨어 모듈을 다시 사용할 수 있다. 따라서 이미 정상적으로 작동하는 것이 입증된 코드를 다시 만들거나 디버깅하지 않아도 된다.

• 이전 데스크톱 애플리케이션: NaCl 을 이용하면 기존 데스크톱 애플리케이션을 웹앱으로 순조롭게 이전 작업을 진행할 수 있다. 애플리케이션의 연산 엔진에 대한 기존 코드를 NaCl 로 바로 이동하거나 다시 컴파일 할 수 있는데, 이때 이용자 인터페이스와 이벤트 처리 영역만 새 브라우저 플랫폼에 맞게 수정하기만 하면 된다. NaCl 를 이용하면 기존 기능을 브라우저에 직접 넣을 수 있고, 동시에 애플리케이션은 브라우저에서 효과적으로 처리하는 작업, 예를 들어, 이용자 인터페이스 처리, 이벤트 처리를 HTML5 표준 기반으로 최대한 활용할 수 있다.

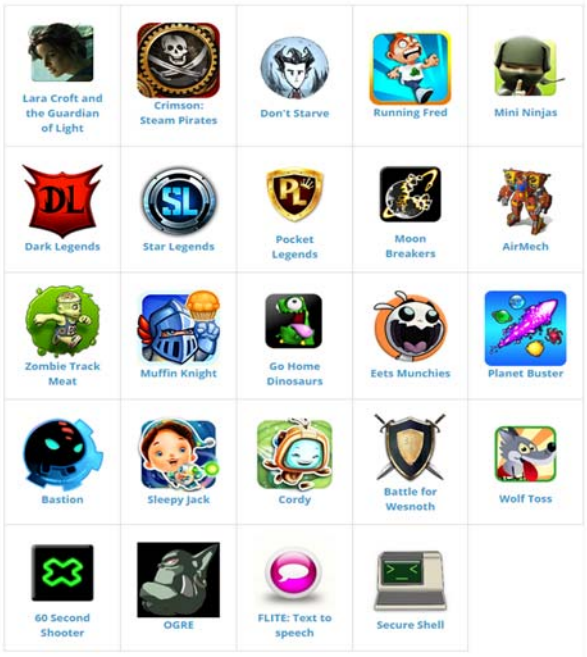
• 엔터프라이즈 애플리케이션(Native 연산 기능 활용): NaCl 은 대형 엔터프라이즈 앱에서 요구하는 숫자 연산을 처리한다. NaCl 은 이용자 데이터의 보호를 위해 브라우저 안에 바로 복잡한 암호화 알고리즘을 구축하여 암호화되지 않은 데이터는 네트워크를 통해 유출되지 않도록 차단한다.

• 멀티미디어 : 음성, 이미지 및 영상 처리 코덱을 브라우저에 추가할 수 있다.

• 3D 게임: NaCl 에서는 지연이 덜한 오디오 및 곧 출시되는 네트워킹 API 와 프로그래밍이 가능한 shader 가 있는 OpenGL ES 로의 저수준 액세스와 결합한 상태에서 기존의 멀티스레드/멀티코어 C/C++ 코드 베이스를 재사용하여 웹앱이 네이티브 속도에 가깝게 실행될 수 있다. 또한 물리 엔진 또는 고도의 웹 게임(RPG)을 구동하는 인공지능 모듈을 실행하는 데 적합하다.

• 가속이 필요한 애플리케이션: 웹 애플리케이션에서 중요한 부분의 가속화를 통해 최적화된 루틴을 제공할 수 있다.

• 그 밖에 Multimedia Editors, CAD modeling, Client-side data analytics, interactive simulations 등에서 활용할 수 있다.



[그림 3-3] NaCl을 이용해 개발한 애플리케이션들

브라우저에서 실행되는 기존 웹앱을 NaCl(PNaCl)은 다음과 같은 제약사항이 있다.

• 통합 IDE 를 통한 개발환경 없음(디버깅은 가능)

• 하드웨어 예외에 대한 지원 없음

• 프로세스 생성 및 하위 프로세스에 대한 지원 없음

• 고유 TCP/UDP 소켓에 대한 지원 없음(TCP 의 경우 웹소켓 및 UDT 의 경우 피어 연결 가능)

• 동시 발생 (차단) I/O 에 대한 지원 없음

• 사용 가능한 메모리로의 쿼리가 지원되지 않음

• 인라인 어셈블리가 NaCl 검사기와 호환할 수 있어야 함(SDK 의 ncval 도구(Tool)를 이용해서 확인할 수 있음)

• Pepper API 호출이 메인 스레드에서 나와야 함

웹 어플리케이션이 아래 사항에 해당된다면 NaCl 을 사용해야 한다.

• 어플리케이션이 인라인 어셈블리와 같은 아키텍쳐에 종속적인 기능을 요구한다. PNaCl 은 이식이 가능한 고성능 대체기능을 제공하려고 노력한다. 일례로 PNaCL 의 이식 가능한 SIMD 벡터가 있다.

• 어플리케이션이 동적 링킹을 사용한다. PNaCl 은 newlib C 스탠다드 라이브러리의 PNaCl 포트만 정적 링킹으로 지원한다. 동적 링킹과 glibc 는 아직 PNaCld 에서 지원되지 않는다. 미래 PNaCl 버전에서 동적 링킹을 지원하기 위해 노력 중이다.

• 함수 안의 함수 (nested function)나 계산된 goto 를 위해 label 의 주소를 이용하는 것 등 PNaCl 의 LLVM 툴체인에서 지원하지 않는 특정 GNU 익스텐션을 이용한다.

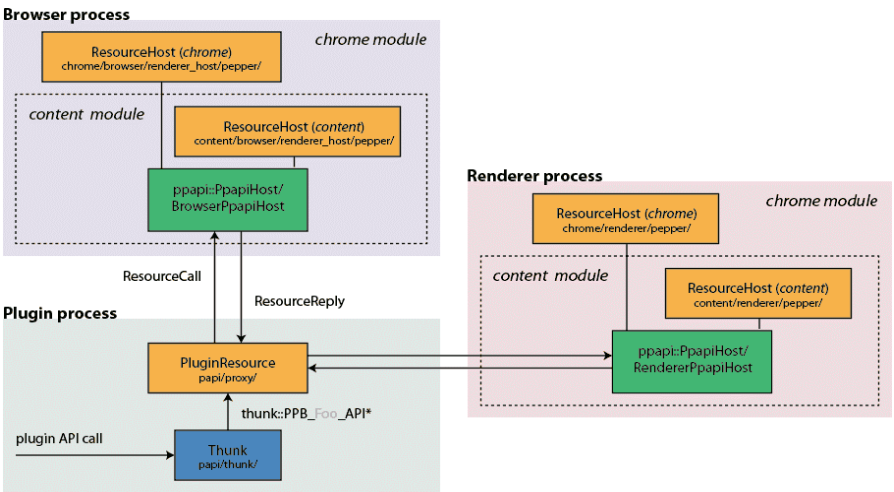
웹 어플리케이션이 아래 사항에 해당된다면 PNaCl 을 사용해야 한다.

• PNaCl 은 네이티브 클라이언트의 더 선호되는 툴체인이며 구글 웹앱 스토어를 통하지 않고 네이티브 클라이언트 모듈을 배포할 수 있는 유일한 방법이다. 만약 프로젝트가 아래의 "NaCl 을 사용해야 할 때"의 제약사항에 속하지 않는다면 PNaCl 을 사용해야 한다.

• 크롬은 브라우저 플러그인이나 어플리케이션의 설치를 요구하지 않으면서도 pexe 모듈의 해석과 웹 어플리케이션에서의 사용을 지원한다. 네이티브 클라이언트와 PNaCl 은 오픈소스 기술이며 우리는 기타 호스팅 플랫폼에도 이 기술들이 추가되기를 바란다.

• 만일 크롬 웹스토어를 통한 통제된 배포가 제품 계획에서 중요하다면 PNaCl 의 장점은 별로 중요하지 않다. 그렇다 하더라도 PNaCl 툴체인을 사용하고 어플리케이션을 크롬 웹스토어를 통해 배포하면 모든 지원하는 아키텍쳐에 맞게 어플리케이션을 명시적으로 컴파일 하지 않아도 되는 PNaCl 의 편리함을 누릴 수 있다.

NaCl 과 ActiveX 와 차이점은 1. 샌드박스(Sandbox) 보안 모델 준수 2. 바이너리 유효성 검증 3. 멀티 운영체제나 브라우저에 적용 가능한 범용성과 호환성이며, 이 중 샌드박스 보안 모델은 특정 메모리 영역에서만 리소스를 접근할 수 있도록 제한 함으로써 안전성을 높이는 방안으로 기술을 제공하고 있다.



[그림 3-4] PPAPI Proxy design

Pepper plug-in API(PPAPI)는 웹 브라우저 플러그인으로 오픈소스 기반 C/C++ API 이며, 호스팅 브라우저와 통신하기 위해 안전하고 휴대용 방법으로

시스템 레벨 기능에 액세스 할 수 있으며, C ++를 사용하는 경우 래퍼를 사용하는 것이 좋다.

이외 SDK, 개발방법 및 예제 소스, 프로젝트 리스트는 아래 사이트를 참조할 수 있다.

https://developer.chrome.com/native-client/quick-start

https://developer.chrome.com/native-client/devguide/tutorial/tutorial-part1

https://code.google.com/p/naclports/wiki/PortList

나. Native Messaging

Native Messaging은 크롬 브라우저에서 실행되는 웹과 Native 애플리케이션 사이에 메시지를 전달하여 통신할 수 있는 기능(API)을 제공한다.

Native Messaging으로 통신하기 위한 Native 애플리케이션은 Native 애플리케이션이 실행되는 OS에 Native Messaging Host로 등록되어 있어야 한다. Native Messaging으로 통신 시 크롬은 각 Native 애플리케이션을 분리된 프로세스로 시작하며 standard input/output (stdio) 스트림으로 메시지를 교환한다.

브라우저에서 실행되는 Native Messaging 를 활용하기 위해서는 아래와 같이 개발해야 한다.

• Native Messaging Host 등록

Native Messaging Host 등록은 통신에 필요한 설정으로 구성된 manifest 파일을 사용한다. 다음은 manifest 파일을 구성한 예이다.

```
{
  "name": "com.my_company.my_application",
  "description": "My Application",
  "path":
  "C:₩₩Program Files₩₩My Application₩₩chrome_native_messaging_host.exe",
  "type": "stdio",
  "allowed_origins": [
  "chrome-extension://knldjmfmopnpolahpmmgbagdohdnhkik/"
  ]
}
```

[예시 : manifest 파일 구성]

manifest 파일 구성 시 사용되는 필드 별 설명은 다음 표와 같다.

<표 3-5> manifest 파일 구성 필드

필드 이름	설명
Name	Native Messaging Host 이름.runtime.connectNative 과 runtime.sendNativeMessage API에서 전달되는 인자 중 Native Application을 지정할 때 사용한다.
description	Native Application에 대한 설명
path	Native Application의 binary 파일에 대한 경로 정보. 리눅스, OS X에서는 절대경로로만 지정해야 하며 Windows 경우에는 상대경로도 가능하다
type	통신 방법을 지정하는 필드. 현재 standard input/output 스트림으로만 통신이 가능하므로 이 필드의 값은 "stdio"만 사용할 수 있다.
allowed_origins	Native Application으로 접근할 수 있는 Chrome Extensions 목록을 정의한다.

구성된 manifest 파일은 Native 애플리케이션이 동작하고 있는 OS 에 따라 저장되는 위치가 틀리며 다음 표에서 확인할 수 있다.

<표 3-6> OS 별 manifest 파일 위치

OS	설명
Windows	manifest 파일이 위치한 경로를 레지스트리에 등록해야 한다. 등록할 레지스트리 키 값은 아래 둘 중에 하나가 되어야 한다. HKEY_LOCAL_MACHINE₩SOFTWARE₩Google₩Chrome₩NativeMessagingHosts₩com.my_company.my_application HKEY_CURRENT_USER₩SOFTWARE₩Google₩Chrome₩NativeMessagingHosts₩com.my_company.my_application.
OS X	아래의 경로에 manifest 파일이 위치해야 한다. /Library/Google/Chrome/NativeMessagingHosts/com.my_company.my_application.json 애플리케이션이 특정 이용자만 실행하도록 설치된 경우는 아래의 경로에 manifest 파일이 위치해야 한다. ~/Library/Application Support/Google/Chrome/NativeMessagingHosts/com.my_company.my_application.json
Linux	아래의 경로에 manifest 파일이 위치해야 한다. /etc/opt/chrome/native-messaging-hosts/com.my_company.my_application.json 애플리케이션이 특정 이용자만 실행하도록 설치된 경우는 아래의 경로에 manifest 파일이 위치해야 한다. ~/.config/google-chrome/NativeMessagingHosts/com.my_company.my_application.json

• Native 애플리케이션 연결 및 메시지 전송

runtime.connectNative 로 Native 애플리케이션으로 통신할 수 있다. 다음은 runtime. connectNative 의 사용 예이다.

```
var port = chrome.runtime.connectNative('com.my_company.my_application');
port.onMessage.addListener(function(msg) {
  console.log("Received" + msg);
});
port.onDisconnect.addListener(function() { console.log("Disconnected");
});
port.postMessage({ text: "Hello, my_application" });
```

[예시 : 연결 및 메시지 전송 예시]

위의 예와 같이 runtime.connectNative 로 연결하고 해당 객체(예에서는 port)로 계속 통신하는 경우 Native 애플리케이션을 실행된 상태로 유지하면서 통신하게 된다.

runtime.connectNative 를 사용하지 않고 runtime.sendNativeMessage 로 통신하는 경우 실행시마다 Native 애플리케이션을 다시 실행 시킨다

```
chrome.runtime.sendNativeMessage('com.my_company.my_application',
  { text: "Hello" },
  function(response) {
    console.log("Received " + response);
  });
```

[예시 : runtime.sendNativeMessage를 통한 메시지 전송 예시]

브라우저에서 실행되는 Native Messaging API 를 사용하는데 있어 다음과 같은 제약사항이 있다.

• Native Messaging 을 사용하는 경우 크로스 사이트 스크립팅 공격에 노출될 수 있어, 크로스 사이트 스크립트 공격에 대하여 안전하도록 처리가 필요하다.

```
chrome.tabs.sendMessage(tab.id, {greeting: "hello"}, function(response) {
  // WARNING! Might be evaluating an evil script!
  var resp = eval("(" + response.farewell + ")");
});
```

```
chrome.tabs.sendMessage(tab.id, {greeting: "hello"}, function(response) {
// WARNING! Might be injecting a malicious script!
  document.getElementById("resp").innerHTML = response.farewell;
});
```

[예시 : 크로스 사이트 스크립팅 공격을 받을 수 있는 예]

```
chrome.tabs.sendMessage(tab.id, {greeting: "hello"}, function(response) {
   // WARNING! Might be evaluating an evil script!
   var resp = eval("(" + response.farewell + ")");
});
```

```
chrome.tabs.sendMessage(tab.id, {greeting: "hello"}, function(response) {
   // innerText does not let the attacker inject HTML elements.
   document.getElementById("resp").innerText = response.farewell;
});
```

[예시 : 크로스 사이트 스크립팅 공격에 안전하게 수정된 예]

이외 SDK, 개발방법 및 예제 소스, PPAPI 개발은 아래 사이트를 참조할 수 있다.

https://developer.chrome.com/extensions/nativeMessaging#native-messaging

https://developer.chrome.com/native-client/c-api

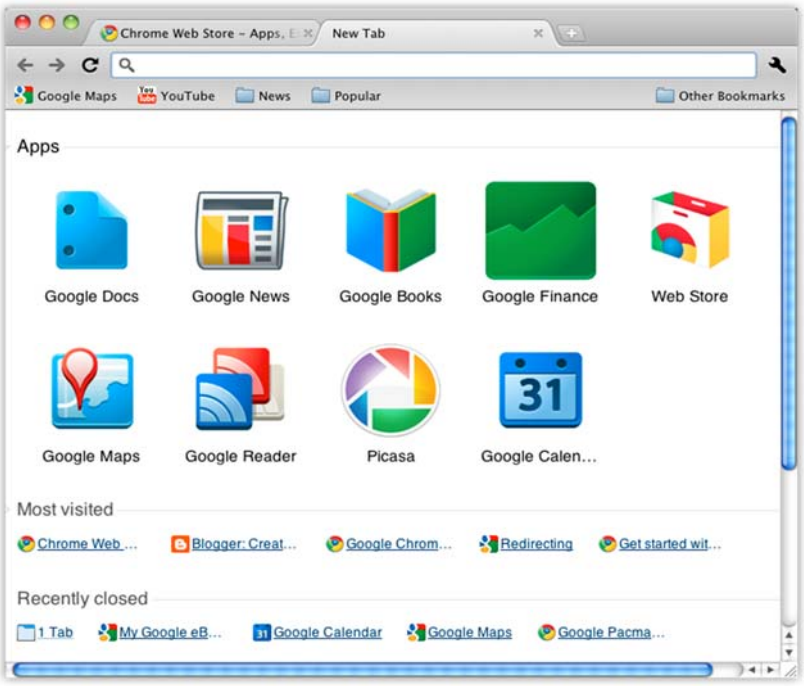
https://code.google.com/p/naclports/wiki/PortList

http://www.chromium.org/nativeclient/getting-started/getting-started-background-and-basics

다. Chrome App

Chrome App 은 Hosted Apps 과 Packaged Apps(installable Web Apps 이라고도 함)으로 구분하며. HTML5, 자바스크립트, CSS 와 같이 순수 웹 기술로 개발한다. Packaged Apps 은 Chrome Web Store 를 통해 패키지 방식으로 배포하는 웹 애플리케이션으로 서버 없이 단독으로 Native Application 과 같은 방식으로 구동하고 서비스를 제공할 수 있다.(Firefox OS, Tizen OS, LG WebOS 도 유사한 방식으로 웹앱을 배포 구동한다.)

이용자는 자신이 좋아하는 웹 애플리케이션을 설치, 배치하고 북마크와 같이 별도 서비스 주소를 입력하지 않고 쉽고 빠르게 서비스를 이용할 수 있는 장점을 가지고 있다. Chrome App 은 기존 웹 사이트에서 제공하거나 액세스 할 수 없는 기능들을 활용할 수 있다. 이를 통해 백그라운드 실행, 네트워크 및 하드웨어 장치 접근, 미디어 도구, 강력한 커뮤니케이션 프로그램을 개발할 수 있다.



[그림 3-5] 크롬 웹 스토어

크롬 브라우저에서 실행되는 Chrome App 특징은 다음과 같다.

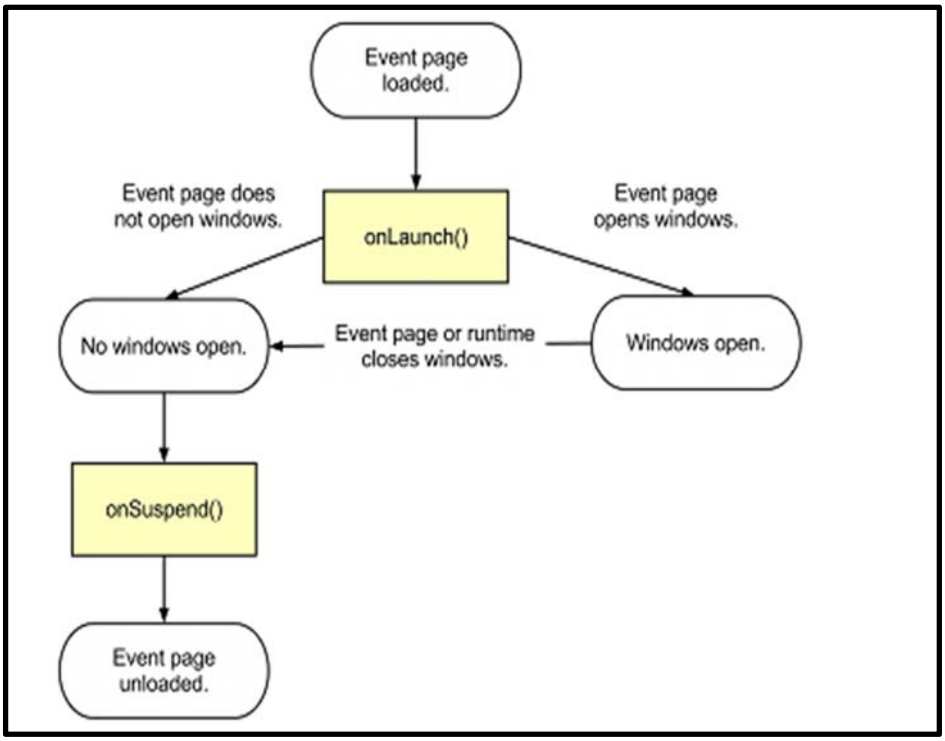
• Chrome 웹 애플리케이션 런타임 및 이벤트 페이지는 Chrome App 수명주기를 관리하며, 런타임은 언제든지 응용 프로그램을, 응용 프로그램 설치를 관리하는 이벤트 페이지를 제어 할 수 있다. 종료, 이벤트 페이지는 런타임에서 이용자나 서버의 이벤트를 수신하고 관리한다.

• Chrome App 형태는 HTML, CSS, 자바스크립트 파일이 .crx 파일로 묶음 처리된 크롬 확장 기능을 사용할 수 있는 웹 응용 프로그램으로 .crx 파일 내에서 적어도 하나의 HTML 파일이 포함되어 있어야 하며 아래와 같은 다양한 형태로 제공할 수 있다.

• Chrome App 을 통한 하드웨어 접근은 외부저장장치, 오디오 입력, 오디오 출력, 동영상 입력 등이 가능하며, 각각의 하드웨어 접근 시 Chrome 정책에 따라 이용자에게 기능 동작 여부에 대해 사전 승인 후 관련 기능을 제공할 수 있다.

• Chrome App 개발을 위해서는 아래와 같은 준비사항이 필요하다.

- 웹 애플리케이션 유사 여부 및 소유권 확인 및 Chrome Developer Dashboard 위치를 확인(일회서 개발자 등록 5$ 지불)
- manifest 파일 생성
- 아이콘 및 대표 image 선택
- 애플리케이션 동작 확인 및 검증
- 애플리케이션 zip 으로 패키징
- 애플리케이션 업로드
- 앱스토어에 양식을 작성하고 이미지를 업로드
- 미리보기 및 앱 목록 리뷰 관리(지속적인 업데이트 관리)



[그림 3-6] Chrome Apps lifecycle works

Chrome App 개발 시 Chrome 정책에 따라 허용된 앱 및 확장 프로그램 유형만 지원이 가능하며 가능한 앱 유형은 아래와 같다.

- 확장 프로그램(NaCl, Legacy Browser Support (LBS)등을 통해 개발된 프로그램)
- 테마
- 구글 앱스 스크립트
- 호스팅된 앱(크롬 웹앱 스토어를 통해 배포된 호스팅 앱, 링크 주소만 게시)
- 크롬 패키지 앱(크롬 웹앱 스토어를 통해 배포된 앱, 서비스 실행)
- Legacy Browser Support 를 이용한 IE 브라우저 지원 확장(IE Tab)
- In-app 페이먼트 및 One-time 페이먼트를 활용한 유료 앱

또한 크롬은 제조사, OS 시스템 정책, 이용자 정책을 우선 적용하며, 이용자 정책은 크롬 브라우저에 개인 계정으로 로그인 했을 때 프로필을 적용한다.

라. Chrome Web API

대다수 확장 기능의 경우 웹 페이지나 웹 애플리케이션에 크롬 자바스크립트 API 와 크롬 Web API 를 포함하여 개발할 수 있다. 특히 네이티브 코드(C, C++)를 추가하지 않고, 구글이 만든 추가 기능을 API 를 통해 쉽게 활용할 수 있다.

크롬 브라우저에서 확장 할 수 있는 API 샘플은 아래와 같다.

• 표준 자바스크립트 APIs: 일반적인 웹 응용 프로그램에서 사용할 수 있는 (DOM) API 를 동일한 코어 자바 스크립트와 문서 객체 모델

• XMLHttpRequest: 하나 이상의 서버로부터 웹 서비스 데이터를 요청하는 XMLHttpRequest 객체를 사용한다. 확장 기능을 호스팅하는 매니페스트 권한 필드 요청을 보낼 수도 있다.

• HTML5 and other emerging API: 구글 크롬은 새로운 API 와 함께 HTML5 기능을 지원한다. 다음은 사용할 수 있는 일부 API 는 다음과 같다.

- audio, video
- application cache
- canvas
- geolocation
- local storage
- notification
- web database 이외 최신 튜토리얼과 정보는 html5rocks.com 를 통해 확인할 수 있다.

• WebKit APIs: 구글 크롬은 웹킷 기반으로 구축되어 있기 때문에, 확장을 위해 웹킷 API 를 사용할 수 있다. 특히 CSS 확장 기능을 통해 이미지 필터, 애니메이션, 이미지 변형 등을 쉽게 제공할 수 있다.

• V8 APIs, such as JSON: JSON 은 V8 에 있기 때문에, JSON 기능을 사용할 수 있는 JSON 라이브러리를 포함 할 필요가 없다.

• APIs in bundled libraries: 브라우저 (예를 들어, jQuery)를 제공하지 않는 라이브러리를 사용하는 경우, 확장 기능과 라이브러리의 JavaScript 파일을 번들 할 수 있다. 번들 된 라이브러리들은 다른 웹 페이지와 마찬가지로 작동한다.

마. asm.js(WebAssembly)

asm.js 는 NativeClient 와 유사하게, C/C++ 로 개발된 소스를 웹에서 실행 가능하도록 단순히 자바스크립트로 변환해 주는 기술은 지속적으로 개발되어 왔다.

asm.js 코드는 (LLVM 기준) Emscripten 등의 소스 컴파일러에 의해 번역되고 수동 메모리 관리와 정적으로 입력 된 언어로 작성되는 자바스크립트의 엄격한 서브 세트로 구성되어있다.

최근보다 빠른 웹과 asm.js 의 부족한 기능을 해결하기 위해 모질라, 구글, MS, 애플이 새로운 바이너리 표준(low-level programming language)인 WebAssembly 를 개발하고 있다. 이러한 WebAssembly 는 웹 브라우저에 가상현실, 암호화 기술, 동영상, 3D 게임 등에 활용될 것으로 기대하고 있다.

asm.js 를 이용하여 웹 서비스를 확장 할 수 있는 방법은 아래와 같다.

• asm.js 의 경우는 현재 대부분의 최신 브라우저에서 지원되고 있기 때문에 Emscripten 이나 Mandreel 등의 컴파일러를 이용하여 기존의 C/C++ 로

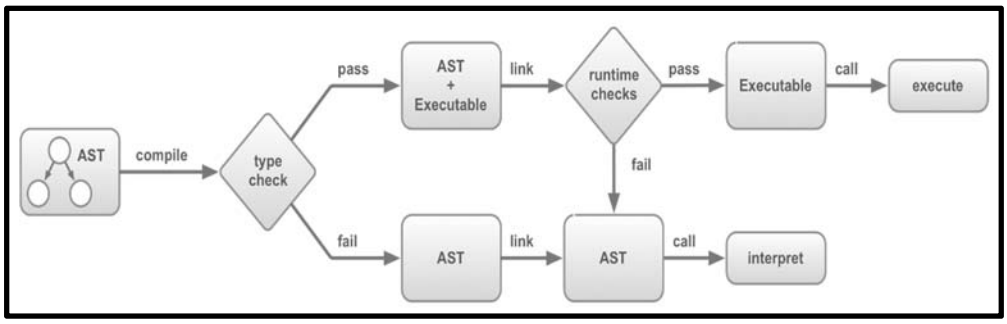
개발된 코드를 자바스크립트로 변환하고, asm.js 를 아래와 같이 선언하여 실행한다.

- asm.js 소스 코드는 아래와 같으며, unreal engine 에서 활용되어 초당 40 프레임의 퍼포먼스를 보여 주었다. 최근의 브라우저에서는 60 프레임까지 성능이 향상되는 것으로 나타나 상용화하기에 충분히 가능한 성능을 보여주고 있다.

```
function DiagModule(stdlib, foreign, heap) {
  "use asm";
  var sqrt = stdlib.Math.sqrt;
  function square(x) {
    x = +x;
    return +(x*x);
  }
  function diag(x, y) {
    x = +x;
    y = +y;
    return +sqrt(square(x) + square(y));
  }
  return { diag: diag };
}
```

[예시 : asm.js 소스 코드 예]

만약 기존의 게임이 C/C++ 이 아니라 Java 등의 다른 언어로 개발되어 있다면 XMLVM 등의 툴을 이용하여 자바 코드를 C++ 로 변환한 이후에 다시 Emscripten 등의 컴파일러를 활용하여 자바스크립트로 변환하면 된다.



[그림 3-7] asm.js 처리 플로우

asm.js 는 자바스크립트 이지만 일반적으로 사용하는 자바스크립트와 아래와 같은 차이가 있다.

- asm.js 는 숫자타입만 다루고 다른 타입은 다룰 수 없다.
- 모든 외부 데이터는 유일한 힙 객체에 typed array 힙 객체에 모두 저장된다.
- 변수에 접근하거나 설정할 때는 특정타입으로 일관성 있게 강제한다
- 컴파일이 필요한 asm.js 에는 "use asm" 디렉티브를 사용해서 명시적으로 컴파일 대상을 지정한다.
- "use asm" 디렉티브를 만나서 asm.js 코드를 처리할 때 엄격한 유효성검사를 통과하지 못하면 이는 무시한다.(오류는 아니고 콘솔에 경고를 출력한다.)

asm.js 및 WebAssembly 를 활용하기 위한 관련 정보는 아래의 링크를 참조하면 된다.

- asm.js 관련 정보 : http://asmjs.org
- Low Level JavaScript : http://mbebenita.github.io/LLJS/
- XMLVM 관련 정보 : http://xmlvm.org/overview
- WebAssembly 정보 : https://www.w3.org/community/webassembly/

바. Chrome for Work(인트라넷)

Chrome for Work 는 기관/단체의 요구사항을 충족시키는 100 개 이상의 정책을 제공한다. 직원들이 받는 애플리케이션 및 확장 프로그램을 설정하고 다양한 플러그인을 관리할 수 있을 뿐 아니라, 맞춤 및 선별된 앱을 갖춘 사설 웹 스토어를 배포하고 이전 앱의 호환성을 관리하는 등의 작업을 수행할 수 있다. 또한 자사 보안 정책에 맞춰 최신 보안 수정사항을 자동 업데이트 할 수 있으며, 제어 범위를 확대하려면 수동 업데이트를 선택할 수 있다.

이를 통해 최신 웹 기술이 적용된 브라우저를 쓰면서도 회사 정책 및 환경에 맞는 맞춤 기능을 제공할 수 있다.

Chrome for Work : http://www.google.com/intl/ko/chrome/business/browser/

사. Legacy Browser Support(기존 브라우저 지원)

Legacy Browser Support(이하 LBS)는 Chrome for Work 를 이용하는 조직에서 Chrome 브라우저를 활용하려고 하지만 사용자가 Internet Explorer 가 필요한 기존 웹사이트와 앱에 계속 액세스해야 하는 경우 이 기능을 사용하여 브라우저 간에 쉽게 전환할 수 있다. Chrome 기존 브라우저 지원 확장 프로그램을 사용하면 사용자가 Chrome 과 다른 브라우저 간에 자동으로 전환할 수 있다. 사용자가 ActiveX 가 필요한 사이트 등 기존 브라우저로 열어야 하는 링크를 클릭하면 URL 이 자동으로 Chrome 에서 기존 브라우저로 이동하여 열린다. 또한 네이티브 호스트 부가기능 실행파일을 설치하면 LBS 에서 사용자가 Chrome 에서 기존 소프트웨어에 의존하지 않는 링크를 클릭하면 IE 에서 링크가 열린다. 이 네이티브 호스트 부가기능을 설치하면 Chrome 과 IE 는 액세스하는 앱이나 웹사이트에 따라 최적의 브라우저로 전환한다. LBS 는 Internet Explorer 6, 7, 8, 9, 10, 11 과 호환되며, LBS 가 작동하려면 Windows 의 향상된 보호 모드를 사용 중지해야 한다.

LBS 소개: https://support.google.com/chrome/a/answer/3019558?hl=ko

4. 주요 기능 별 대체 기술 적용 방안

구분	세부 구분	웹 표준	브라우저 확장기술	특이사항
인증	공인인증서 관리	부분 지원	NaCl, Native Messaging, EXE 실행파일	
	전자서명(부인방지)	지원		입력창 보안
보안	데이터 암호화(선로)	지원		SSL
	개인방화벽	부분 지원		
	키보드 보안	미 지원	EXE(URI Scheme), PIPE. NaCl	
	로그인 보안	지원		가상키보드, 입력
	백신	부분 지원		
전자결제	결제창	지원		
	전자결제	부분 지원	NaCl, Native Messaging	
	간편결제	지원		
그래픽 및 챠트	그래픽	지원		
	챠트	지원		
	그리드	지원		서버 연동
	레포팅 툴	지원		서버 연동
멀티미디어	동영상 재생	지원		
	음악 재생	지원		
	동영상 DRM	지원		EME
파일관리	다중파일 업로드	지원		
	다중파일 다운로드	지원		

구분	세부 구분	웹 표준	브라우저 확장기술	특이사항
전자문서	전자문서 뷰어/편집	지원		
	문서보안(문서암호화)	부분 지원	NaCl(PPAPI), Native Messaging	서버 연동 방식
	위변조방지	부분 지원	NaCl(PPAPI), Native Messaging	서버 연동 방식
	워터마킹/캡쳐방지	부분 지원	NaCl(PPAPI), Native Messaging	서버 연동 방식
	문서출력	부분 지원		클라우드 프린터
게임 런처	게임 런처(구동)	부분 지원	PNaCl, PPAPI, asm.js	3D 연산 및 가속화
시스템접근제어	장치제어	부분 지원	NaCl, Native Messaging, EXE 실행파일	
	버전관리	부분 지원	NaCl, Native Messaging, EXE 실행파일	
	시스템 정보확인	지원		Device.js, CSS 미디어 쿼리
기타	화상, 채팅 상담	지원		Web RTC, electron

5. NPAPI 전환 적용 사례

가. Adobe PPAPI

어도비의 FlashPlayer는 ver 14부터 PPAPI에 대한 콘텐츠 디버거를 지원 했으며, ver 16 부터 윈도우와 맥 운영체제에서 공식적으로 다운로드를 지원하고 있다.

기존 Flash Player는 Flash runtimes과 Adobe Flash Player browser plug-in으로 구성되어 있으며 AIR는 Flash runtimes core위에 개발자가 독립적으로 콘텐츠를 배포 실행할 수 있도록 지원하는 런타임으로 구성되어 있다.

FlashPlayer는 다음과 같은 기능들로 구성되어 있으며, 공식적으로 PPAPI 적용 사항은 아래와 같다.

- Animation
- Vector-based graphics
- Audio (including MP3) (PPAPI)
- Multicast Video(PPAPI)
- Hardware video decoding(PPAPI)
- Microphone and webcam access
- Low-level bitmap manipulation(PPAPI)
- Binary-based sockets
- Strongly typed, class-based programming language
- Hardware-accelerated 2D and 3D content(PPAPI)

현재 기능 중 PPAPI 를 사용하지 않는 기능들은 브라우저가 지원하는 HTML5, WebApplication API, SVG, CSS, Web Animation 을 통해 구현할 수 있다.

IV. 부록

1. 용어집

자바스크립트 (JavaScript)	미국의 넷스케이프 커뮤니케이션즈 사(Netscape Communications)가 개발한 스크립트 언어. 1996년 2월에 발매한 월드 와이드 웹(WWW) 브라우저인 넷스케이프 내비게이터(Netscape Navigator) 2.0에 실장하였다. 브라우저에서 실행하는 스크립트 언어를 기술한다. 언어 규격은 자바의 부분 집합(subset)으로 되어 있다. 하이퍼텍스트 생성 언어(HTML) 문서를 작성하는 수준의 이용자가 사용하는 것을 주안점으로 하여 자바의 언어 규격으로부터 변수의 형(정수형이나 문자열형 등)을 생략하거나 새로운 클래스 정의를 할 수 없도록 하였다. 스크립트는 HTML 문서 속에 직접 기술하며, 'script'라는 꼬리표를 사용한다. 자바스크립트와 같은 기능을 갖는 것으로서 MS에서는 'Virtual Basic Scripting Edition'을 개발하였다.
액티브X (ActiveX)	윈도 이용자들이 인터넷을 편리하고 쉽게 이용하도록 MS에서 개발한 것으로, 기존의 응용 프로그램으로 작성된 문서 등을 웹과 연결시켜 그대로 사용할 수 있게 하는 기술. 인터넷 익스플로러를 위해 고안되었으며, 실생활 페이지에 접속하면 자동으로 내려받기 되어 설치된다. 선 마이크로시스템즈사의 자바(Java) 기술에 대항하는 기술이다.
플래시 (Flash)	어도비사에서 개발되었으며 액션 스크립트와 모션 그래픽을 이용하여 PC용, 웹용 모듈을 작성할 수 있는 도구이다.
이용자 에이전트 (User-agent)	HTTP 요청 시 HTTP 헤더에 넣어 전송하는 문자열로 브라우저에서는 브라우저 이름, 버전정보 등이 포함된다. 각 브라우저 , 검색로봇 마다 다른 문자열이 포함된다.
하이퍼텍스트 전송 규약(HTTP)	인터넷의 월드 와이드 웹(WWW) 서버와 WWW 브라우저가 파일 등의 정보를 송수신하는 데 사용되는 클라이언트/서버 규약. WWW 브라우저의 화면상에서 URL(uniform resource locator)를 지정하는 데 사용된다. 예를 들면 'http://www.snu. ac.kr/index.html'과 같이 'http://'로 시작되는 URL을 지정하면, 여기에 있는 데이터를 하이퍼텍스트 전송 규약(HTTP)을 사용하여 서버에서 브라우저로 전송한다.
XMLHttpRequest	브라우저에서 XML을 요청할 수 있도록 하는 내장객체로 자바스크립트로 제어가 가능하여 동적인 웹페이지를 만드는데 사용된다.
암호수출제한정책(Encryption Export Controls)	미국 정부에서 전통적으로 강력한 암호에 대해서 강력한 통제를 지속해왔다. 1996년부터 비군사적인 항목에 대해서 조금씩 완화되기 시작하여 여러 차례의 관련 규제조항의 수정을 거쳐서 적대국가가 아닌 나라에 대해서는 2001년까지 대부분 완화 되었다.
중간자공격(Man in the Middle Attack, MITM)	네트워크 통신을 중간에서 조작하여 도청하거나 변경하는 공격

HTML5 (HyperText Markup Language 5)	HTML의 차기 주요 제안 버전으로 W3C에서 표준을 제정하고 있는 핵심 마크업 언어. 2007년 초기 리뷰 기반으로 시작하여 2014년 4 분기 표준 권고안(Recommendation)을 목표로 하고 있으며, 다양한 서비스를 지원하기 위한 규격들이 추가되고 있다.
웹 애플리케이션	자바스크립트(JavaScript), HTML, CSS등 Web 기반 언어로 개발된 응용 프로그램에서 클라이언트로 브라우저나 웹 런타임을 사용하는 모든 응용 프로그램이다.
자바스크립트 API	자바스크립트와 같은 인터페이스 정의 언어(IDL)를 사용하여 정의된 웹 애플리케이션을 위한 프로그램 인터페이스. 일반적으로 디바이스 기능(device capabilities)에 대한 접근을 위해 웹 애플리케이션을 실행하기 위한 수단으로써 제공된다.
CSS(Cascading Style Sheets)	웹 문서의 전반적인 스타일을 미리 저장해 두는 기술로 월드 와이드 웹 컨소시엄(W3C)에서 표준화한 하이퍼텍스트 생성 언어(HTML)용 스타일 시트. HTML을 이용해서 웹 페이지를 개발할 경우 전반적인 틀에서 세세한 글꼴을 일일이 지정해 주어야 하지만, 웹 페이지의 스타일을 미리 저장해 두면 웹 페이지의 한 가지 요소만 변경해도 관련되는 전체 페이지의 내용이 한꺼번에 변경되므로, 문서 전체의 일관성을 유지할 수 있고 작업 시간도 단축된다. 하나의 스타일을 정의하면 여러 개의 문서에서 사용할 수 있으며 수정이 쉽다
DOM(Document Object Model)	문서 형식 선언(DTD : Document Type Declaration) 또는 DOCTYPE이란 어떤 SGML이나 XML 기반 문서 내에 그 문서가 특정 문서 형식 정의(Document Type Definition)를 따르는 것을 지정하는 것이다. 본래 DTD에 기반한 SGML 도구를 이용해 문서 해석 가능성과 유효성을 검사하기 위한 목적으로 문서 내에 삽입된다.
SEED	1999년 한국정보보호진흥원(KISA)에서 개발한 대칭키 블록 암호 알고리즘. 128bit로 표준화 되었고 256bit까지 있다.
Site Key	이용자가 이미 등록해 놓은 정보를 통하여 웹에서 이용자 사이에 상호 인증하는 방식. 피싱 방지 목적으로 사용 됨
CAPTCHA	자동화된 입력인지 사람이 입력하는 지를 식별하기 위한 방법. 임의의 숫자나 문자를 컴퓨터가 알아보기 어렵게 하여 표시한 후 이에 대한 값을 입력하도록 요청하는 방식. 때로는 사람도 알아보기 어렵다.

2. 약어집

AA	Amount Authentication
API	Application Program Interface
ARS	Advanced Record System
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CORS	Cross-Origin Resource Sharing
CMP	Certificate Management Protocol
CMS	Content Management System
CSR	Certificate Signing Request
DDoS	Distributed Denial of Service attack
DTD	Document Type Definition
ECMA	European Computer Manufacturers Association
FDS	Fraud Detection System
EME	Encrypted Media Extensions
HTTP	Hyper Text Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
IDL	Interface Definition Language
HSM	Hardware security module
ISP	Internet Secure Payment
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MPEG	Moving Picture Experts Group
NPAPI	Netscape Plugin Application Programming Interface
OTP	OneTime Password
OS	Operating System
OWASP	The Open Web Application Security Project

PCIDSS	Payment Card Industry Data Security Standard
PKI	Public Key Infrastructure
RDP	Remote Desktop Protocol
RFB	Remote Frame Buffer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTP	Trusted Third Party
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
SEED	128-bit Symmetric Block Cipher
SSL	Secure Socket Layer
VNC	Virtual Network Computing
VPN	Virtual Private Network
W3C	World Wide Web Consortium
XBL	XML Binding Language
XHTML	eXtensible HyperText Markup Language

3. 참고자료 및 인용

[웹 기술]

1. 웹 표준 진단도구

   http://www.koreahtml5.kr

2. W3C HTML5 표준

   W3C, HTML5, http://www.w3.org/TR/html5/, 2014

3. w3school.com

   http://www.w3schools.com/

4. New Tricks in XMLHttpRequest2(Eric Bidelman)

   http://www.html5rocks.com/en/tutorials/file/xhr2/?redirect_from_locale=ko

5. Internet Explorer 개발자 센터

   http://msdn.microsoft.com/ko-kr/library/ie/hh180177(v=vs.85).aspx

6. 브라우저 HTML5 지원 확인

   http://caniuse.com/

7. 네이버 개발자 블로그 - 브라우저는 어떻게 동작하는가

   http://helloworld.naver.com/helloworld/59361

8. Rendering: repaint, reflow/relayout, restyle

   http://www.phpied.com/rendering-repaint-reflowrelayout-restyle/

9. How (not) to trigger a layout in Webkit

   http://gent.ilcore.com/2011/03/how-not-to-trigger-layout-in-webkit.html

10. 프론트엔드 개발자를 위한 렌더링 성능 인자 이해하기

    http://cwdoh.com/workshop/2014/06/14/understanding-rendering-performance-matters-in-chrome/

11. W3C CSS3 MediaQuery

    http://www.w3.org/TR/css3-mediaqueries/

12. W3C Drag&Drop

W3C, HTML5, Drag and Drop http://www.w3.org/TR/html5/editing.html#dnd, 2014

13. W3C FileAPI

W3C, File API, http://www.w3.org/TR/FileAPI/. 2013

14. W3C FileWriter

W3C, File API: Writer, http://www.w3.org/TR/file-writer-api/, 2014

15. W3C IndexedDB

W3C, Indexed Database API, http://www.w3.org/TR/IndexedDB/, 2013

16. W3C WebMessage

W3C, HTML5 Web Messaging, http://www.w3.org/TR/webmessaging/, 2012

17. W3C Web RTC

W3C, WebRTC 1.0: Real-time Communication Between Browsers, http://www.w3.org/TR/webrtc/, 2013

18. W3C WebStorage

W3C, Web Storage, http://www.w3.org/TR/webstorage/, 2013

19. W3C XHR

W3C, XMLHttpRequest Level 1, http://www.w3.org/TR/XMLHttpRequest/, 2014

20. W3C XHR2

W3C, XMLHttpRequest Level 2, http://www.w3.org/TR/XMLHttpRequest2/, 2014

21. Bootstrap

http://getbootstrap.com/

22. jsPDF / pdf.js

https://github.com/MrRio/jsPDF

https://mozillalabs.com/en-US/pdfjs

23. MSXML 사용 방법

http://blogs.msdn.com/b/xmlteam/archive/2006/10/23/using-the-right-version-of-msxml-in-internet-explorer.aspx

24. XML 파서 사용 방법(libxml2)
http://xmlsoft.org/

25. HTML5 Video Player 비교
http://praegnanz.de/html5video

26. Top 10 Best HTML5 Audio Players
http://www.scratchinginfo.net/top-10-best-html5-audio-players

27. HTML5rocks- 크롬의 렌더링 가속: 레이어 모델
http://www.html5rocks.com/ko/tutorials/speed/layers/

28. Google검색- GPU Accelarated Compositing in Chrome
http://www.chromium.org/developers/design-documents/gpu-accelerated-compositing-in-chrome

29. HTML5rocks- Scrolling Performance
http://www.html5rocks.com/en/tutorials/speed/scrolling/?redirect_from_locale=ko

30. HTML5rocks- Jank Busting for Better Rendering Performance
http://www.html5rocks.com/en/tutorials/speed/rendering/?redirect_from_locale=ko

31. HTML5rocks- CSS 페인트 타임과 페이지 렌더 가중치
http://www.html5rocks.com/ko/tutorials/speed/css-paint-times/

32. HTML5rocks- 불필요한 페인팅 회피하기
http://www.html5rocks.com/ko/tutorials/speed/unnecessary-paints/

33. HTML5rocks- 불필요한 페인팅 회피하기: Animated GIF버전
http://www.html5rocks.com/ko/tutorials/speed/animated-gifs/

34. HTML5rocks- 고성능 애니메이션
http://www.html5rocks.com/ko/tutorials/speed/high-performance-animations/

35. Getting Started with WebRTC

http://eee.html5rocks.com/ko/tutorials/webrtc/basics/

36. Drag&Drop

W3C, HTML5, Drag and Drop http://www.w3.org/TR/html5/editing.html#dnd, 2014

[보안]

37. TLS Cipher Suite Registry

https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4

38. OWASP

https://www.owasp.org/index.php/Main_Page

39. 보안서버보급지원 정보 사이트

http://opa.or.kr/Business-secureserver_support.do

40. SSL 테스팅 툴

https://www.ssllabs.com/ssltest/

41. CryptoJS

Google, JavaScript implementations of standard and secure cryptographic algorithms, https://code.google.com/p/crypto-js/, CryptoJS 3.1

42. CSP

W3C, Content Security Policy 1.0, http://www.w3.org/TR/CSP/, 2012

43. CSP2

W3C, Content Security Policy Level 2, http://www.w3.org/TR/CSP2/, 2014

44. 브라우저 Sandbox 보안 모델

W3C, HTML5 Sandboxing, http://www.w3.org/TR/html5/browsers.html# sandboxing, 2014

45. Keygen

W3C, HTML5 keygen element, http://www.w3.org/TR/html5/forms.html#the- keygen-

element, 2014

46. MSRCRYPTO

Microsoft Research, MSR JavaScript Cryptography Library, v1.2, http://research.microsoft.com/en-us/downloads/29f9385d-da4c-479a-b2ea-2a7bb335d727/, 2014

47. SCJL

Stanford University, Stanford Javascript Crypto Library, http://crypto.stanford.edu/sjcl/, 2009

48. PKCS1

RSA Laboratories PKCS#1, RSA Cryptography Standard v2.1, 2001

49. PKCS5

RSA Laboratories PKCS#5, Pasword-Based Cryptography Standard v2.0, 1999

50. PKCS8

RSA Laboratories PKCS#8, Private-Key Information Syntax Standard, 1993

51. PKCS10

RSA Laboratories PKCS#10, Certification Request Syntax Specification, 2000

52. PKCS11

RSA Laboratories PKCS#1, Cryptographic Token Interface Standard v2.1, 2001

53. PKCS12

RSA Laboratories PKCS#12, Personal Information Exchange Syntax Standard v1.0, 1999

54. RFC2511

IETF, RFC 2511, Internet X.509 Certificate Request Message Format, March 1999

55. RFC4210

IETF, Internet X.509 Public Key Infrastructure Certificate Management Protocol

(CMP), 2005

56. RFC6454

IETF, The Web Origin Concept, 2011

57. RFC6712

IETF, Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP), 2012

[인증/결제]

58. MPay

https://mpay.lgcns.com:8443/web/getNoticeView.dev

59. 페이팔 개발자 센터

https://developer.paypal.com/

60. e-ID

BSI TR-03112 Das eCard-API-Framework, https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03112/index_htm.html, 2014

4. 집필진

안내서 목차	집필진
Ⅰ. 개요	한국인터넷진흥원 인터넷환경개선팀 엠트케어 박종일 대표
Ⅱ. 멀티 운영체제 및 브라우저 정책	엠트리케어 신현진 이사
Ⅲ. NPAPI 대체기술	엠트리케어 박종일 대표 페이게이트 이동산 부사장 플라이하이 김기영 대표 동국대 이창환 교수
Ⅳ 부록	엠트리케어 박종일 대표

※ 본 안내서 관련 문의처 : 한국인터넷진흥원 인터넷환경개선팀 김성훈 선임연구원(02-405-6637, shkim@kisa.or.kr)

인터넷 이용환경 개선을 위한
NPAPI 대체 기술 안내서

인 쇄 | 2015년 07월

발 행 | 2015년 07월

발 행 인 | 백기승

발 행 처 | 한국인터넷진흥원(KISA, Korea Internet&Security Agency)서울시 송파구 중대로 135
(가락동) IT벤처타워 Tel: (02) 405-6637

인 쇄 처 | (주)엠트리케어 TEL:070-4756-8722

<비매품>