Non-NPAPI VidyoWeb

Technical Design

Draft 1.0, May 2015

Introduction

VidyoWeb since its original launch was comprised of two components – HTML UI and a Web plugin which is based on NPAPI. While all of the Vidyo customers who use VidyoWeb use the Web plugin, many customers choose to deploy their own HTML interface on top of the VidyoWeb NPAPI plugin.

In April 2015, Google has deprecated the Chrome browser support for NPAPI plugins. Administrators are still capable of re-enabling NPAPI plugins in places where enterprise policy is used. Users are also capable of re-enabling NPAPI plugins by using //flags/#enable-npapi.

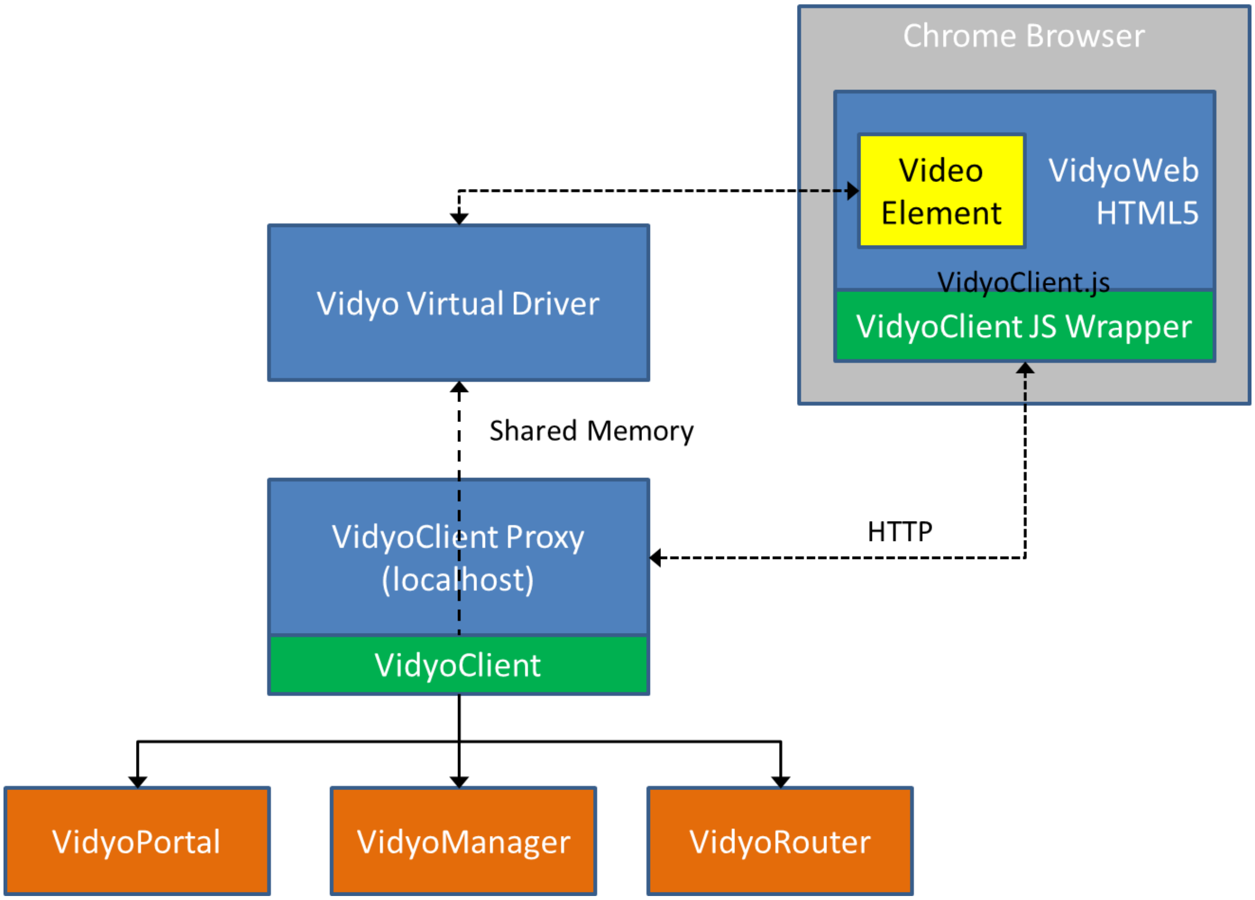
For users of VidyoWeb, this deprecation means that when they upgrade their Chrome browser to version 42, the VidyoWeb NPAPI will stop working and the browser will constantly prompt the user to download and install the plugin, even after this has already been done.

To continue support for Chrome users of VidyoWeb, Vidyo has decided to develop a new version of the VidyoWeb which will not be based on NPAPI. The selected alternative is composed of a number of components as depicted in the diagram in the architecture section of this document.

The purpose of this document is to describe the technical design of the non-NPAPI VidyoWeb, detail the solution's architecture and outline each of the components the solution is comprised of and explain their functionality.

Architecture

The various components which are part of the non-NPAPI VidyoWeb solution are depicted in the following diagram. Each of these components is described below.



Chrome browser

When opening a VidyoConference URL through a browser, the browser opens the VidyoWeb HTML interface. Once loaded, the HTML interface checks the browser type and version and loads the corresponding VidyoWeb.

When used on chrome browsers earlier than version 42, as well as on any other browser, the NPAPI VidyoWeb plugin plugin is used. When used on a Chrome browser which is version 42 or later, the non-NPAPI VidyoWeb plugin is used.

In both cases, the HTML interfaces starts by validating the availability of a corresponding VidyoWeb. For non-NPAPI VidyoWeb, the HTML validates the existence of the VidyoClient Proxy and the Vidyo Virtual Driver. If any of these does not exist, the non-NPAPI VidyoWeb installer (the installer bundles together the Vidyo Virtual Driver and the VidyoClient Proxy which also contains the VidyoClient library) is downloaded. Once the download is completed the user launches the installer and once the installation is done the VidyoClient Proxy will get launched.

VidyoWeb HTML and the Video element

The HTML user interface is the same regardless of the type of VidyoWeb plugin that is being used. Similarly, future features that are added to the HTML user interface will support both types of plugins. However, on the technical level, several differences exist:

- Sending commands and receiving responses
  With the NPAPI plugin, the HTML interface uses a JavaScript interface to send commands to the plugin and receive responses. With the non-NPAPI architecture, a similar JavaScript interface is being used but instead of communicating directly with the plugin, the commands are sent first as HTTP calls to the VidyoClient Proxy which transfers them to the underlying VidyoClient component.
- Rendering the video
  When using the NPAPI plugin, the video window is passed to VidyoClient which renders the video within the windows boundaries. With the non-NPAPI, the video is rendered inside a <video> HTML element. The video rendering is done by the Vidyo Virtual Driver component as explained within the section describing that component

VidyoClient JS Wrapper

The VidyoClient JS Wrapper is a thin abstraction layer which exposes a JavaScript interface to the HTML and internally converts received requests to HTTP requests which are sent to the VidyoClient Proxy.

Vidyo Virtual Driver

The Vidyo Virtual Driver is responsible for rendering the received video stream. A shared memory is established between the Vidyo Virtual Driver and the VidyoClient component which is packaged together with the VidyoClient Proxy. The frames that are received from VidyoRouter by the VidyoClient component are transferred to the Vidyo Virtual Driver through shared memory and are being rendered within the <video> element of the HTML user interface.

VidyoClient Proxy

The VidyoClient Proxy acts as a mediator between the HTML user interface and the VidyoClient component. It processes HTTP requests sent by the HTML user interface through the VidyoClient JS Wrapper and sends them to the VidyoClient component. Responses, received from the VidyoClient component, are being returned in the opposite direction.

VidyoClient

VidyoClient is a common component that is being used by any Vidyo endpoint and contains all the logic which is not platform specific. Among many other features, the VidyoClient component is responsible for the communication with all of the Vidyo backend components, including VidyoPortal, VidyoManager and VidyoRouter.

Installation

The installers of the non-NPAPI VidyoWeb are standard installers on both Windows and OSX which package all the relevant components. When the HTML identifies that VidyoWeb was not yet installed, it identifies the correct installer out of the four available installers (Windows non-NPNPI for Chrome on Windows, Windows NPAPI for IE and Firefox on Windows, OSX non-NPAPI for Chrome on OSX, OSX NPAPI for Safari and Firefox on OSX) and downloads it.

As part of the installation, the installer looks for earlier versions and uninstalls them if found.

Migration

For customers who use the VidyoWeb standard UI migrating from an NPAPI to non-NPAPI is transparent as VidyoWeb v1.3 or later contain both the NPAPI plugin, a non-NPAPI one and the HTML which uses them both as needed.

For customers who deploy their own HTMl UI, there are several changes required in order to add support for the non-NPAPI VidyoWeb.

These changes include:

- Identify which type of VidyoWeb should be used and loading (and downloading if it does not exist) the right VidyoWeb plugin
- When using a non-NPAPI VidyoWeb, communicate requests through the VidyoClient JS Wrapper
- When using a non-NPAPI VidyoWeb, use the <video> element for rendering the video