Internet Printing

Microsoft
Hewlett Packard

Babak Jahromi
(babakj@microsoft.com)

Lead Software Design Engineer
Printing Development Team

Start

Internet Printing

- Print to URL
- HTML queue status from URL
- Web point and print
- Install drivers from URL

Seamless integration with printing architecture

How Do We Get A Printer URL?

- Have “Internet Information Server” or “Peer Web Server” running
- Share the printer
- The URL will be: http://ServerName/PrinterShareName

Printer URL Example

In an Intranet, finance department will be: http://Finance/LJet4si

Printer URL Example For Internet

- Submit a color job to downtown Redmond branch of Kinko’s: http://Redmond42.kinko.com/EpsonCol
- Send a fax to Toshiba customer service: http://Fax.Support.Toshiba.com/FaxMe

Demo

- Browse a printer URL
- Remote printers folders URL
- Create a printer, then share it. It shows up in the HTML printer folder

Technology

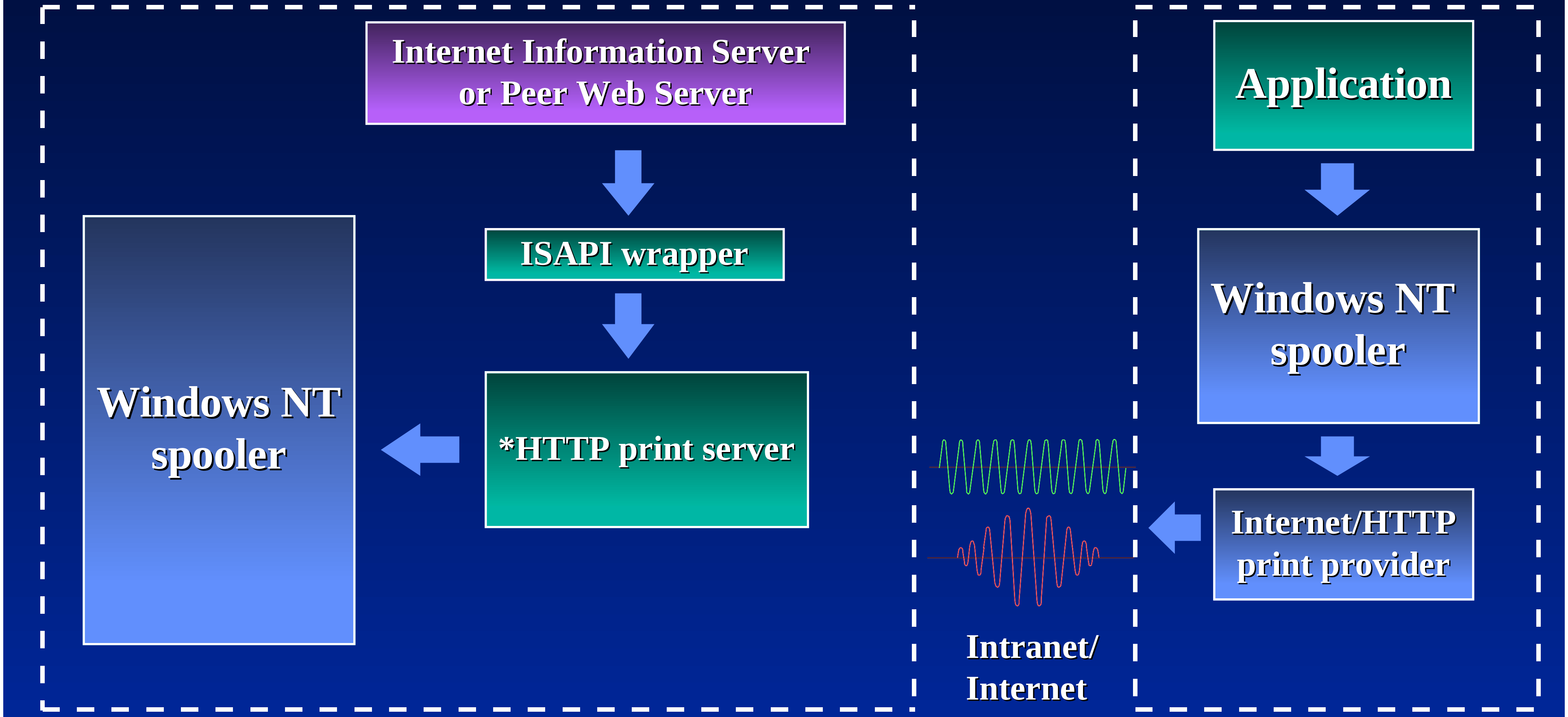
- HTTP/HTML-based
- Any browser on any client platform views the printer or job information
- Server components implemented as an Internet Information Server ISAPI dll

Printing To URL

- Windows NT® 5.0 clients print to Windows NT 5.0 servers via an HTTP print provider supporting OpenPrinter(URL)
- Prints across firewalls to Internet

Architecture

Printing to a URL over HTTP



Internet Information Server or Peer Web Server

ISAPI wrapper

*HTTP print server

Windows NT spooler

Intranet/ Internet

Application

Windows NT spooler

Internet/HTTP print provider

Windows NT Print Server (any Windows NT Server or Windows NT Workstation machine)

Windows NT client

* Receives the RAW job via HTTP Post command

HTTP Print Provider

- Based on Win32 Internet APIs
- Win32 Internet APIs available for Win9x and WinNT

HTTP Print Provider->OpenPrinter()

```
// Get a valid session connection to the Internet
// g_szUserAgent is: ”Windows Internet Print Provider"
hSession = InternetOpen(g_szUserAgent,
                        PRE_CONFIG_INTERNET_ACCESS,
                        NULL,
                        0,
                        INTERNET_FLAG_DONT_CACHE);

InternetOpenUrl(hSession,
                "http://host/scripts/share/.printer",
                 NULL,
                 0,
                 INTERNET_FLAG_DONT_CACHE,
                 0);
```

HTTP Print Provider->StartDocPrinter()

```
hConnect = InternetConnect( hSession,
                            "kinko.com",
                            0,
                            szUserName,   // e.g. BabakJ”
                            NULL,
                            INTERNET_SERVICE_HTTP,
                            0, 0);

hReq = HttpOpenRequest( hConnect,
                        "POST",
                        "/scripts/ShareName/.printer?JobStart",
                        HTTP_VERSION,
                        NULL, NULL,
                        INTERNET_FLAG_DONT_CACHE,
                        0);
```

HTTP Print Provider->StartDocPrinter() continued…

```
// 37 happens to be this "Post" command's content length
// pidi is the DOC_INFO_2 sent to the server
InetHttpSendRequest( hReq, "Content-length: 37\n", (DWORD)-1,
                                                        pidi, cbSize);



// Get back the job id from the server. The server response is:
// Content-type: text/html\r\n\r\n<html><title>JobId=%d
// </title><body> </body>
InetInternetReadFile( hReq, (LPVOID)szResponse,
                                  sizeof(szResponse), &dwRead)
```

HTTP Print Provider->WritePrinter()

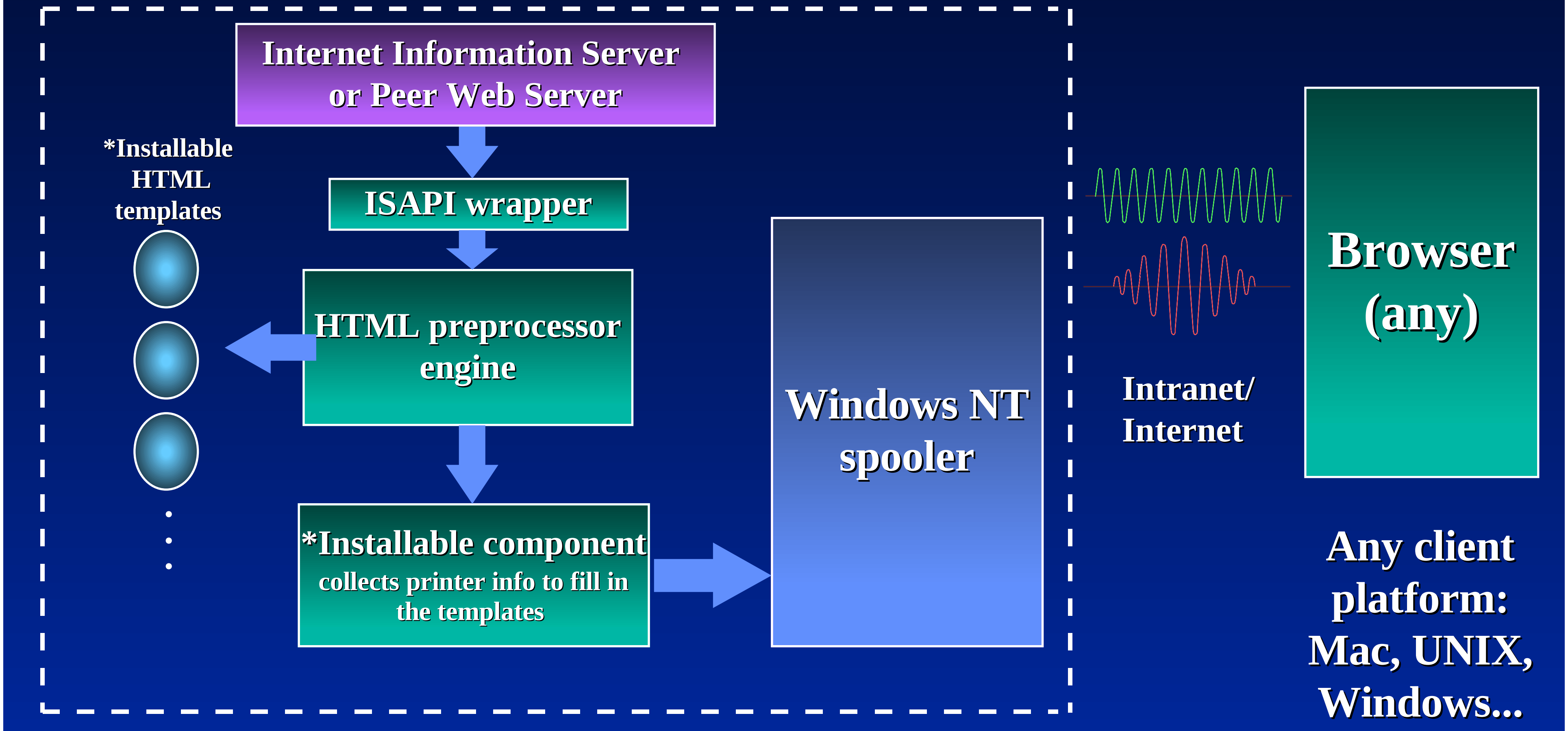
```
// The job ID received from the server happens to be 5
hReq = HttpOpenRequest( hConnect,
                        "POST",
                        "/script/ShareName/.printer?JobData&5",
                        HTTP_VERSION,
                        NULL, NULL,
                        INTERNET_FLAG_DONT_CACHE,
                        0);

// Send the raw job data
// 1045 happens to be this "Post" command's content length
InetHttpSendRequest(hReq, "Content-length: 1045\n",
                              (DWORD)-1, pBuf, 1045);
```

HTML Printer Status

- HTML view of printer or job information generated automatically by the server
- Customize HTML view by providing HTML templates
- Add-on components embedded in HTML tags providing vendor-specific views

Architecture

HTML view of printers



Any client platform: Mac, UNIX, Windows...

Windows NT Print Server (any Windows NT Server or Windows NT Workstation machine)

* Microsoft provides default implementation

Tags and Tag Processors

{=TagProcessorDllName.EntryPoint}

•

•TagProcessorDllName is the Tag Processor DLL name

•

•EntryPoint is exported from TagProcessorDllName.dll by name

•

•Example: {=WinPrint.PrinterName}

Tag Processor Entry Points

The custom tag processor should have an export per tag:

typedef BOOL WINAPI
(*PFNINETPRNCALLBACK)(PINETPRINTINFO pInetPrn);

pInetPrn contains fields like:

-
- EXTENSION_CONTROL_BLOCK *pECB;
- int iNumQueryArgs;
- BOOL fQueryArgIsNum[MAX_Q_ARG];
- DWORD QueryArgValue[MAX_Q_ARG];
-

Tag Processor Entry Points

```
LPSTR      pszFriendlyName;
LPSTR      pszShareName;
HANDLE     hPrinter;
DWORD      dwJobId;

// Writes binary data to the client
void (*fpHTMLWrite)(HANDLE hCallBack, void const *pv,
ULONG cb );

// Writes a string to the client using printf syntax.
void (*fpHTMLPrintf)(HANDLE hCallBack, const char *
pszFormat, ...);
```

Action Commands

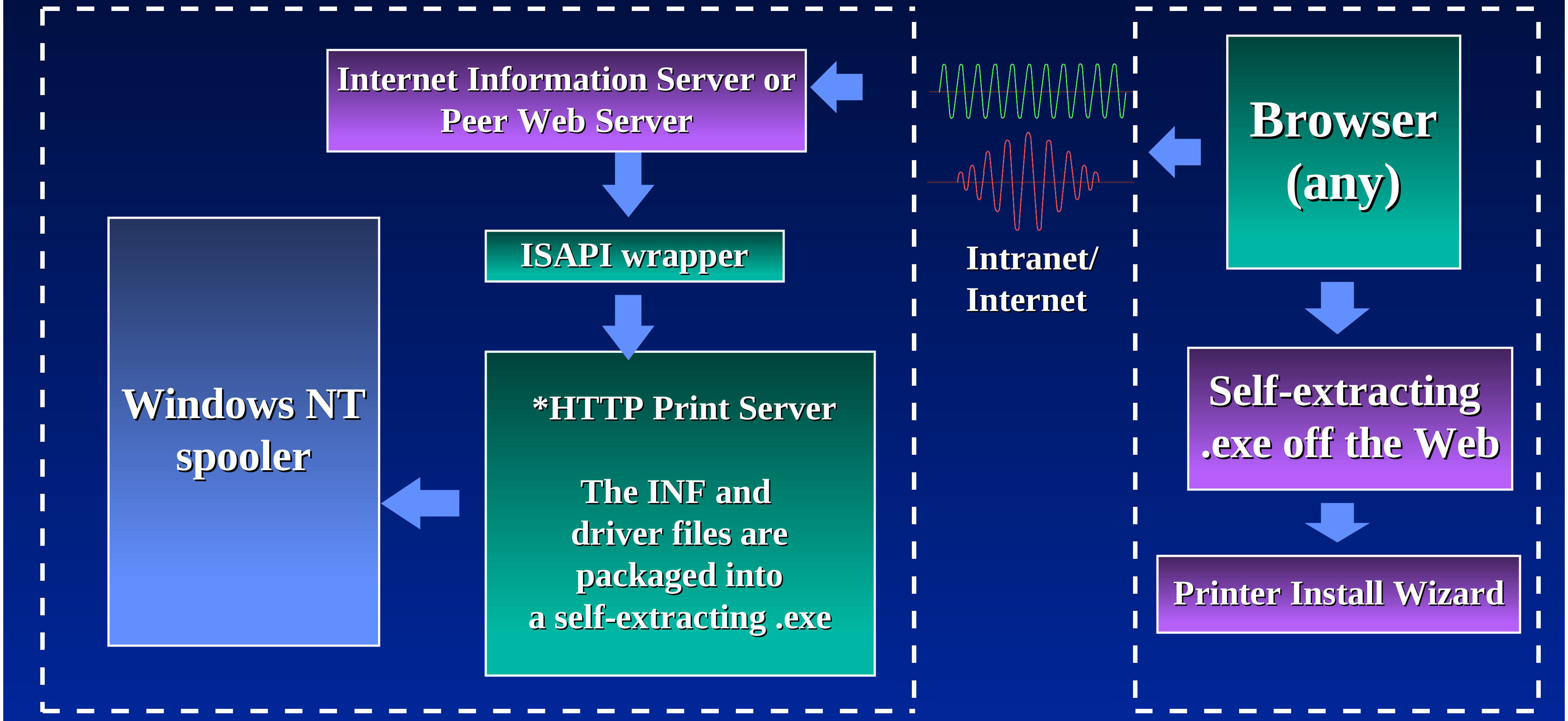
-
- PFNINETPRNCALLBACK TagProcCommand(PINETPRINTINFO pInetPrn);
-
-
- A URL with the format: {=WinPrint. PrinterURL}**?mytagproc.dll&arg1&arg2…** would direct the Windows NT template processor to call the action command inside the tag processor dll.
-
-

Web Point And Print

- Hot link in HTML view
- Drivers automatically downloaded off the print server

Architecture

Web point and print



Internet Information Server or Peer Web Server

ISAPI wrapper

*HTTP Print Server

The INF and driver files are packaged into a self-extracting .exe

Windows NT spooler

Intranet/ Internet

Browser (any)

Self-extracting .exe off the Web

Printer Install Wizard

Windows NT Print Server (any Windows NT Server or Windows NT Workstation machine)

Windows NT client

* Receives a browser command to create and send a self-extracting install executable

Drivers From The Web

- AddPrinter Wizard downloads INFs and drivers from vendors’ Web sites
- INF files can have vendor URLs

Questions?