



IPP Everywhere / Cloud Printing New Operations for Mobile Printing

22 October 2010

IPP Everywhere Project

Rationale for New Operations for Mobile Printing



- IPP Everywhere and Cloud Printing models use Print Services accessed via TCP/IP connections
 - Firewalls and routers block incoming connections
 - Print Services must notify Printers of available new jobs
 - Email is slow and requires device configuration
 - Instant messaging is resource intensive and requires device configuration
- New operations should use outgoing connections from the Printer to the Print Service
 - Printer takes Client role to initiate connections and send operation requests to the Print Service (Server)
 - Print Service takes Server role (i.e., Printer) to accept incoming connections and receive operation requests

Proposed New Operations for Mobile Printing – 1 of 4



- Create-Printer-Registration
 - Create-Printer-Registration request (request-id, printer-uuid, [registration-lease-duration])
 - register Printer with Server (X.509 certificate in TLS layer)
 - Create-Printer-Registration response (request-id, status-code, [server-uuid], [registration-lease-duration], [fetch-interval])
 - respond w/ Registration lease duration and fetch interval (i.e., Server-directed polling)
- Create-Client-Registration
 - Create-Client-Registration request (request-id, client-uuid, [registration-lease-duration])
 - register Client with Server (X.509 certificate in TLS layer)
 - Create-Client-Registration response (request-id, status-code, [server-uuid], [registration-lease-duration])
 - respond w/ Registration lease duration

Proposed New Operations for Mobile Printing – 2 of 4



- Get-Available-Jobs
 - Get-Available-Jobs request (request-id, printer-uuid, get-timeout)
 - query for list of available Jobs from Server w/ timeout
 - Get-Available-Jobs response (request-id, status-code, *[job-uuid])
- Fetch-Jobs
 - Fetch-Jobs request (request-id, printer-uuid, fetch-timeout, *[job-uuid])
 - fetch any or specific available Jobs from Server w/ timeout
 - Fetch-Jobs response (request-id, status-code, *[Create-Job request or Print-URI request])
- Reply-To-Jobs
 - Reply-To-Jobs request (request-id, printer-uuid, 1*[Create-Job response or Print-URI response])
 - accept or reject Jobs after fetching from Server
 - Reply-To-Jobs response (request-id, status-code)

Proposed New Operations for Mobile Printing – 3 of 4



- Fetch-Documents
 - Fetch-Documents request (request-id, printer-uuid, fetch-timeout, [job-uuid])
 - fetch Documents for Job fetched from Server w/ timeout
 - Fetch-Documents response (request-id, status-code, *[Send-Document request])
- Reply-To-Documents
 - Reply-To-Documents request (request-id, printer-uuid, job-uuid, 1*[Send-Document response])
 - accept or reject Documents after fetching from Server
 - Reply-To-Documents response (request-id, status-code)

Proposed New Operations for Mobile Printing – 4 of 4



- Report-Job-Progress
 - Report-Job-Progress request (request-id, printer-uuid, job-uuid, job-state, job-state-reasons, job-impressions-completed)
 - report job progress (i.e., output impressions)
 - Report-Job-State response (request-id, status-code)
- Report-Job-State
 - Report-Job-State request (request-id, printer-uuid, job-uuid, job-state, [job-state-reasons])
 - report job state change (processing, completed, etc.)
 - Report-Job-State response (request-id, status-code)
- Report-Printer-State
 - Report-Printer request (request-id, printer-uuid, printer-state, [printer-state-reasons])
 - report printer state change (idle, processing, etc.)
 - Report-Printer response (request-id, status-code)

Next Steps for New Operations for Mobile Printing



- Review operations via IPP WG calls and email
 - Consensus – November 2010
- Define operations in full RFC 2911-style
 - First draft – November 2010
- Define operations in new IPP JPS3 spec
 - First draft – December 2010 / January 2011
- Open issues that should be addressed
 - Required mutual authentication of Printer and Print Service during connection setup
 - Federation of Print Services across Mobile/Cloud environments