Internet Printing Protocol: Model and Protocol Details

Scott Isaacson
Novell, Inc.

IPP Analysts Briefing
Boston, August 27, 1997

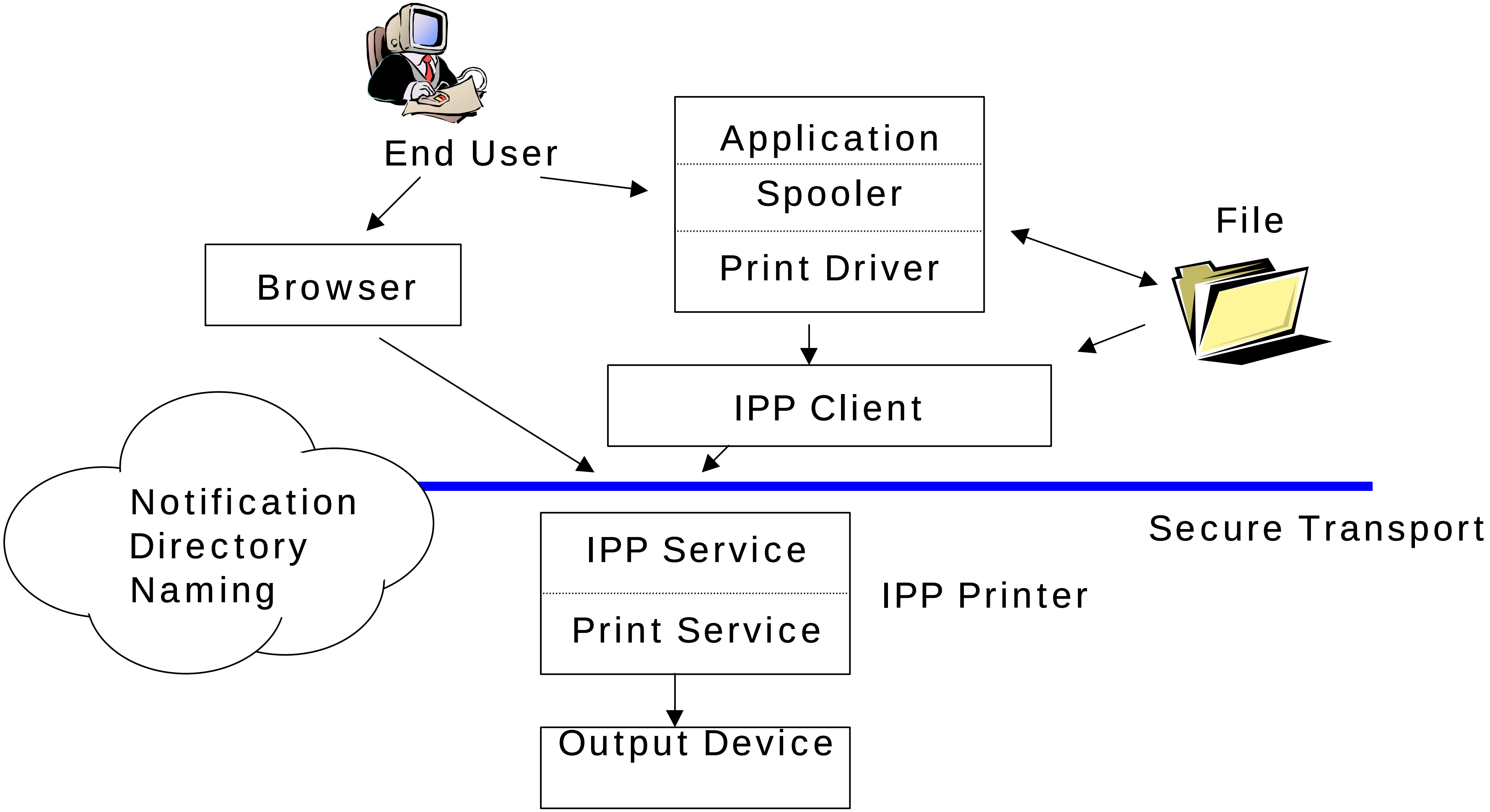
Model and Protocol Details

- Model and Semantics
  - Abstractions independent of encoding
  - Alignment with other Standards
    - Printer MIB, Job MIB, Host Resource MIB
- Protocol Specification
  - “on-the-wire” data representation
  - Transport specific mapping - HTTP/1.1

Architecture

- Distributed environment: Internet
- Client
  - Query the Printer
  - Submit Jobs
  - Query Jobs
  - Cancel Jobs
- Server (IPP Printer)
  - Implement and support the Protocol
  - Conform to the Model semantics
  - Enabling hooks for other services
    - Security, Naming, Directory, Notification

Layering


End User
Application
Spooler
Print Driver
File
Browser
IPP Client
Notification
Directory
Naming
Secure Transport
IPP Service
Print Service
IPP Printer
Output Device

Object Model

- Follow the industry lead
  - Objects with attributes
  - Operations to manipulate those objects
  - Operations to query object (status, attributes)
- Example
  - Print request creates a Job object
  - Client supplied attributes
  - Printer supplied attributes
    - state, submitter's authenticated identity, etc.

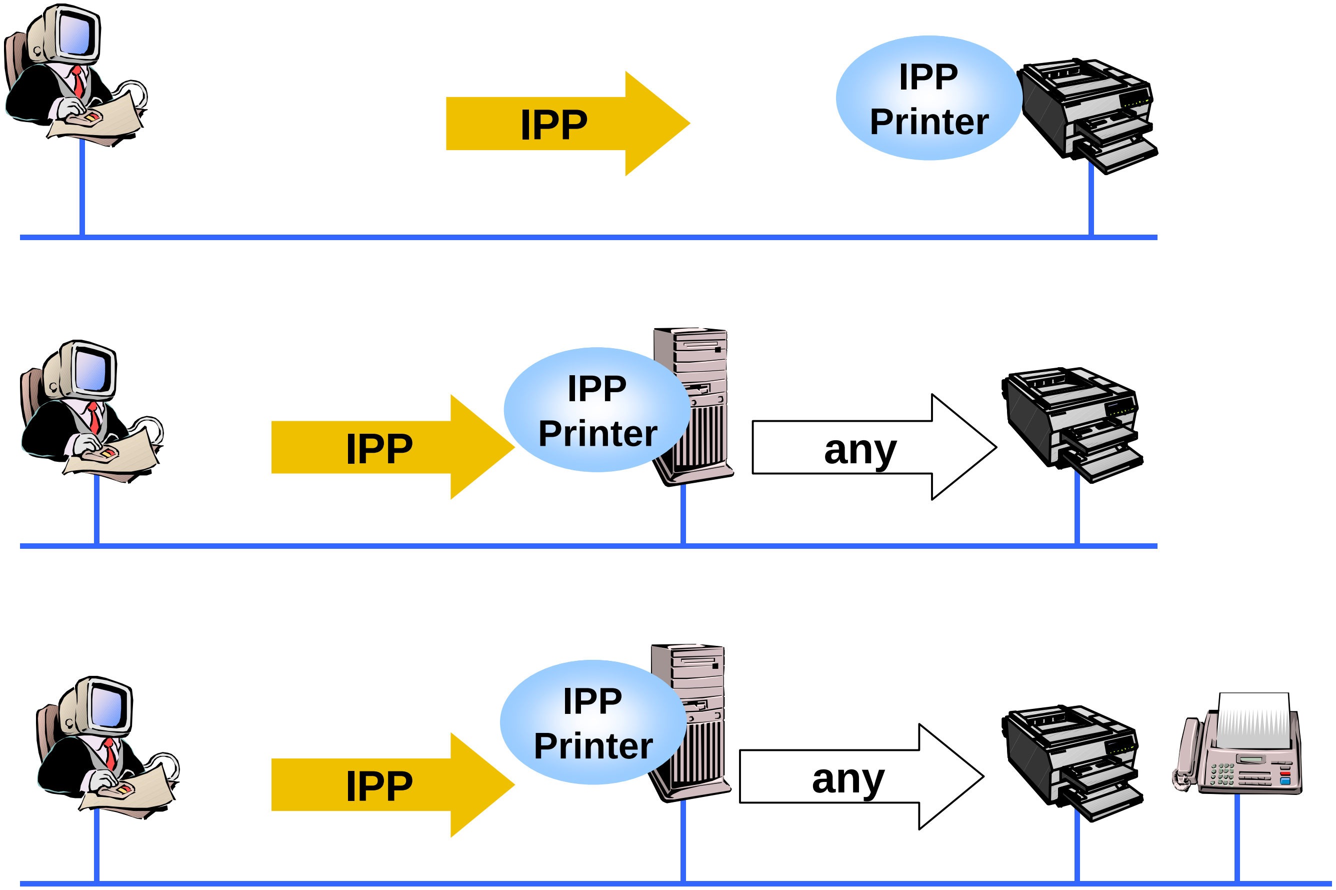
IPP Objects

- Printer
- Job
- Document

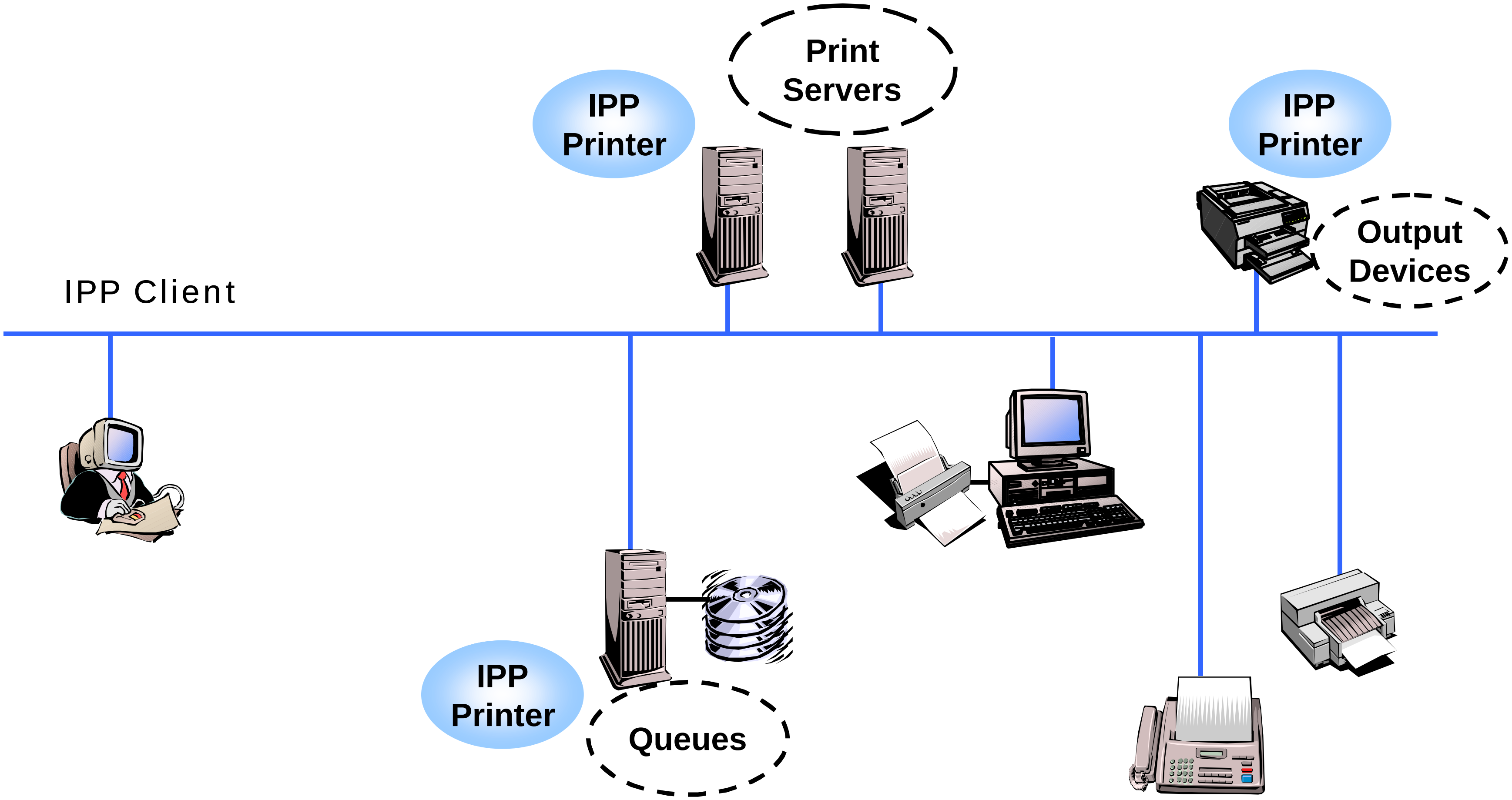
IPP Printer

- Implements IPP
- Logical or physical device
- Multiple configurations
  - server
  - embedded
- Job processing
  - Options for spooling and scheduling
  - Provide job status

IPP Printer Configurations



IPP Printer Implementations



Printer Attributes

- Uniform Resource Identifier (URI)
- Name
- State
- Accepting jobs
- Default languages
- Languages supported

Job Attributes

- Job Identifier
- Job Owner’s Identity
- Job State

Document Attributes

- Document Name
- Document Format (PDL)
  - PostScript
  - PJL
  - PCL
  - IPDS
  - EscapeP
  - Interpress
  - etc.

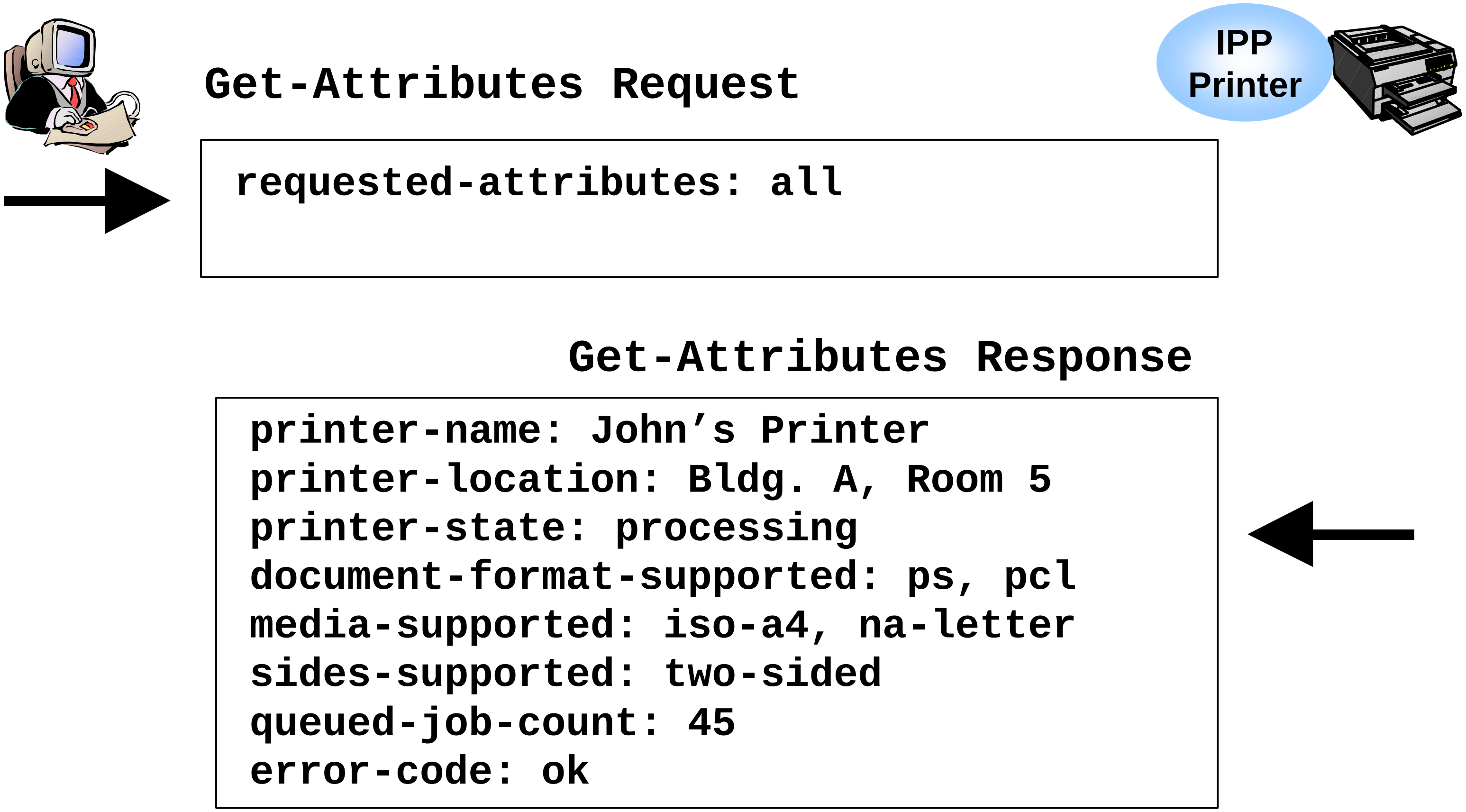
IPP Operations

- Request/Response
- Operation attributes
  - Print-Job: Job Template attributes
  - Cancel-Job: Message
  - Get-Attributes: Set of attribute names
- Status Codes
  - OK, Server-Error, Client-Error

Printer Operations

- Get-Operations
  - Responds with a list of supported operations
- Print-Job
  - Submits a Job, “pushes” job data to the Printer
- Validate-Job
  - Validates client supplied attributes (no job data)
- Get-Jobs
  - Lists jobs at the Printer
- Get-Attributes
  - Responds with supported attributes

Example: Get-Attributes



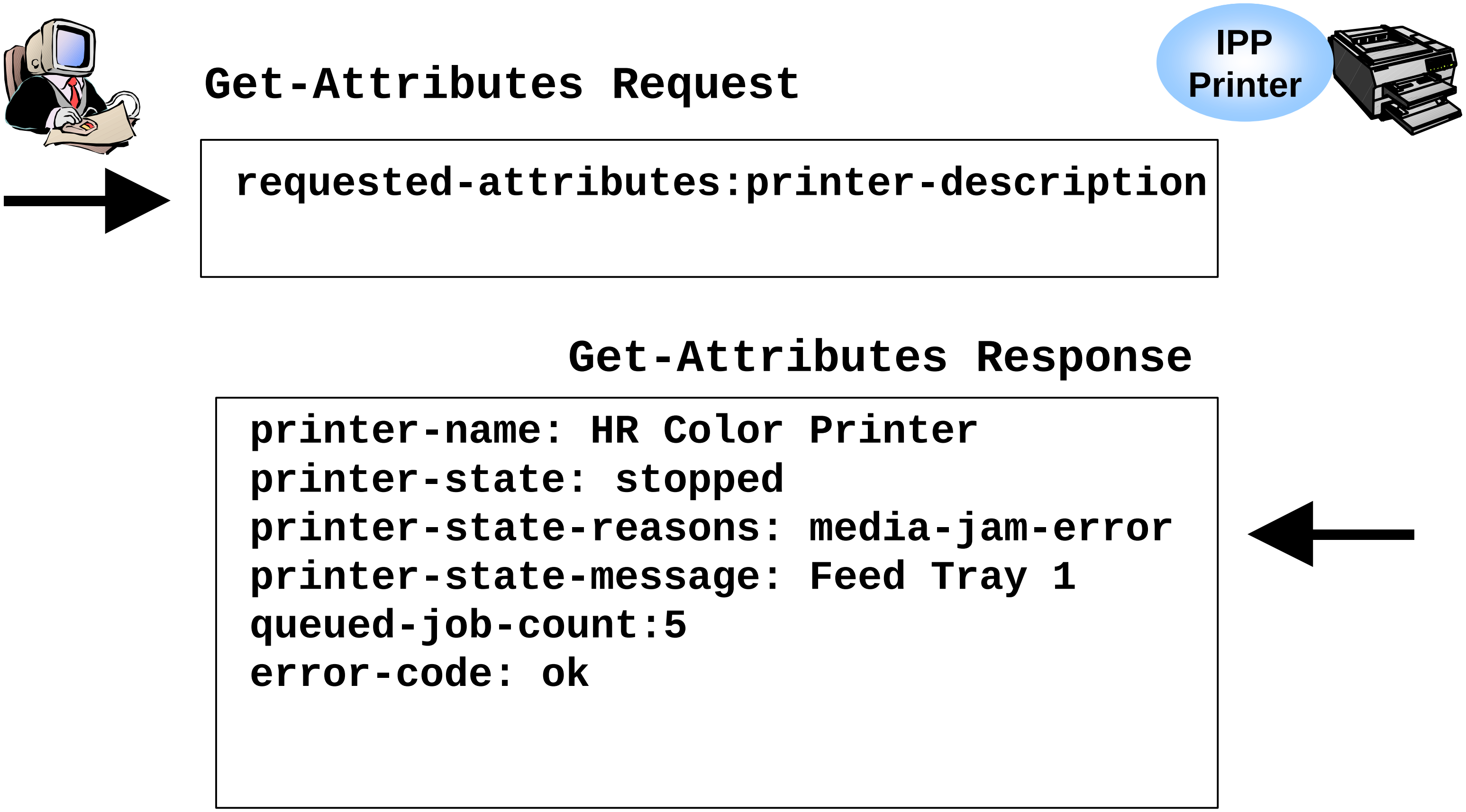
Get-Attributes Request

```
requested-attributes: all
```

Get-Attributes Response

```
printer-name: John’s Printer
printer-location: Bldg. A, Room 5
printer-state: processing
document-format-supported: ps, pcl
media-supported: iso-a4, na-letter
sides-supported: two-sided
queued-job-count: 45
error-code: ok
```

Example: Get-Attributes Printer Stopped



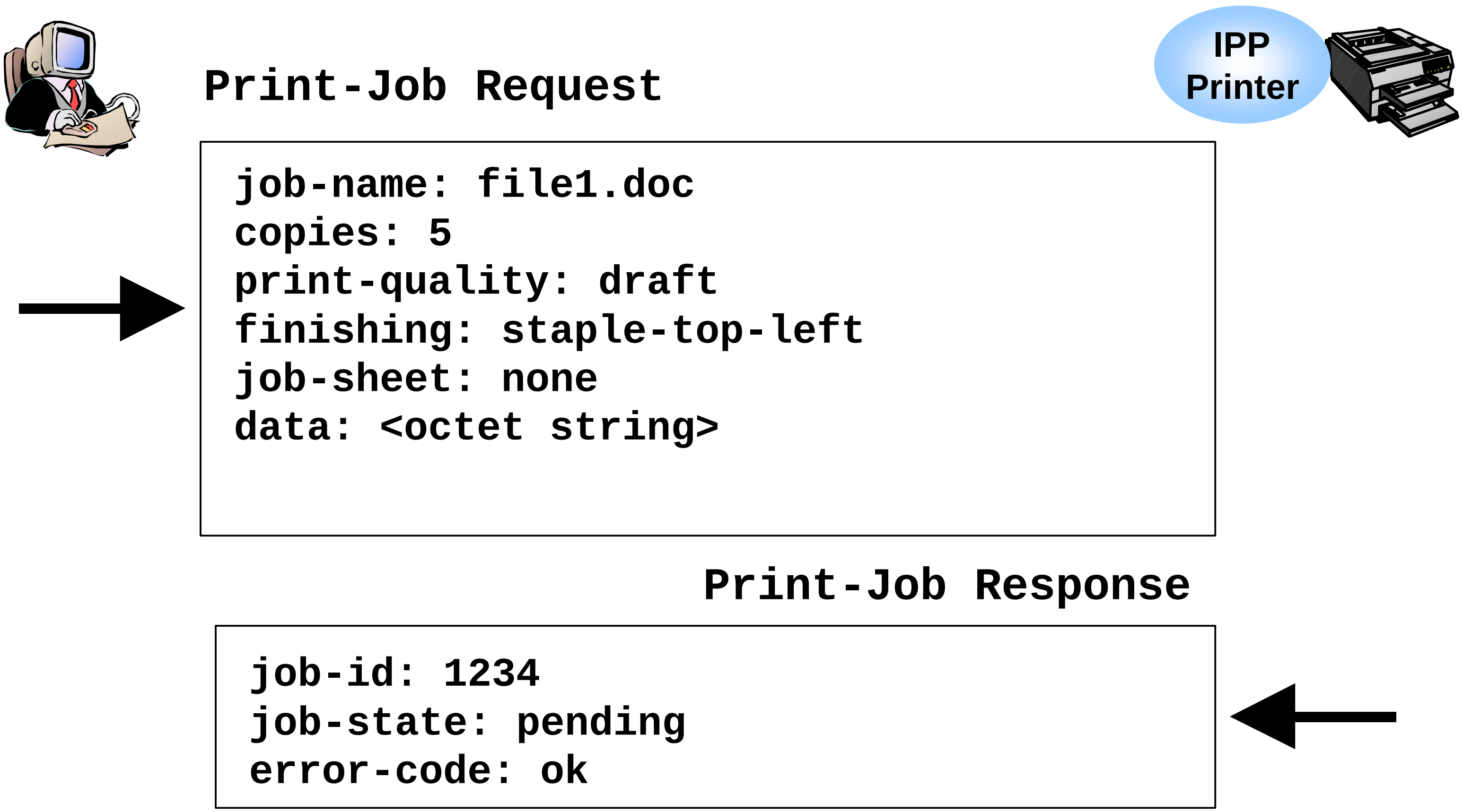
Get-Attributes Request

```
requested-attributes:printer-description
```

Get-Attributes Response

```
printer-name: HR Color Printer
printer-state: stopped
printer-state-reasons: media-jam-error
printer-state-message: Feed Tray 1
queued-job-count:5
error-code: ok
```

Example: Print-Job



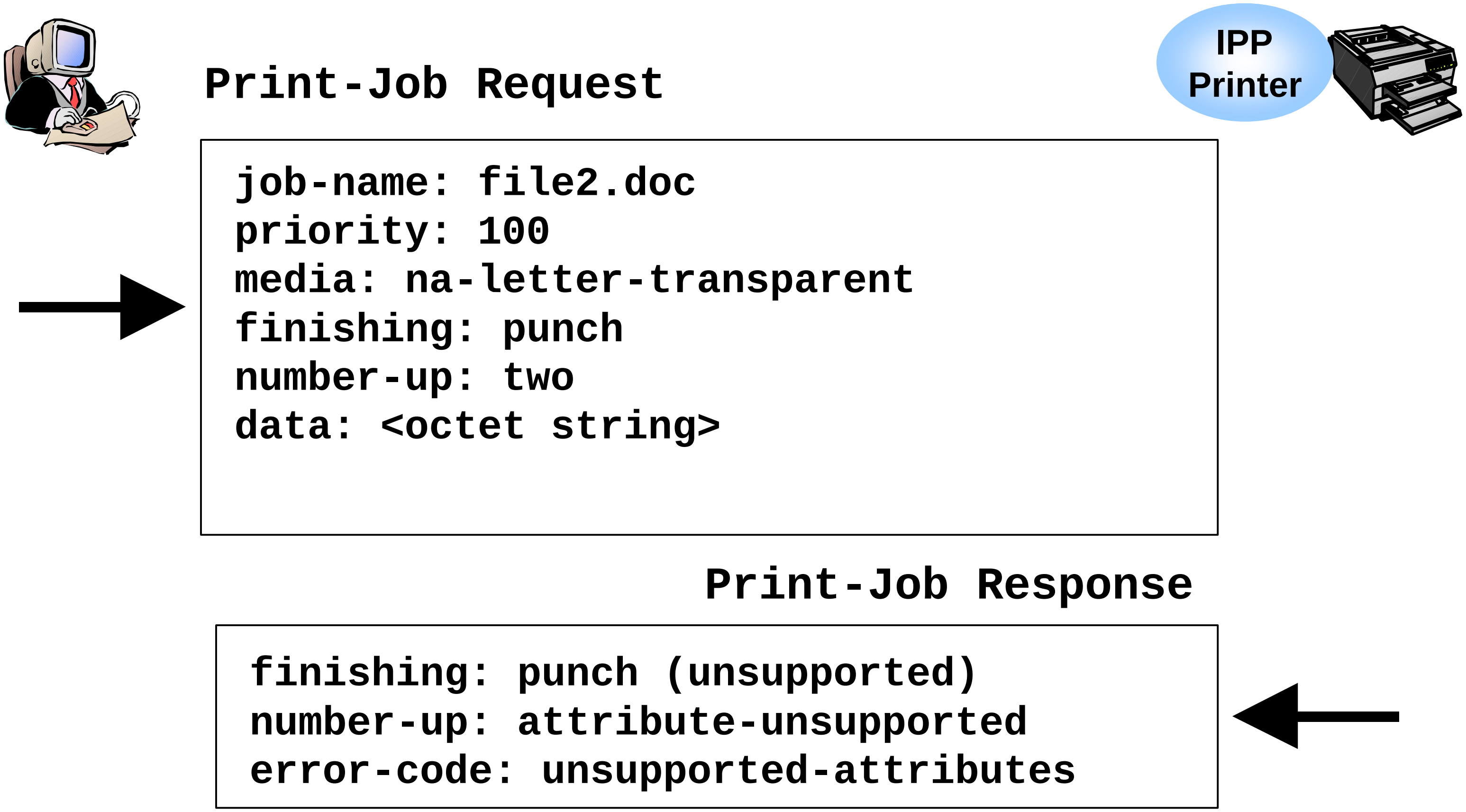
Print-Job Request

```
job-name: file1.doc
copies: 5
print-quality: draft
finishing: staple-top-left
job-sheet: none
data: <octet string>
```

Print-Job Response

```
job-id: 1234
job-state: pending
error-code: ok
```

Example: Print-Job (error)



Print-Job Request

```
job-name: file2.doc
priority: 100
media: na-letter-transparent
finishing: punch
number-up: two
data: <octet string>
```

Print-Job Response

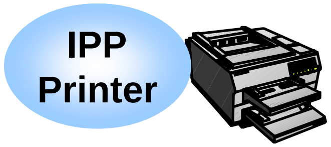
```
finishing: punch (unsupported)
number-up: attribute-unsupported
error-code: unsupported-attributes
```

Job Operations

- Cancel-Job
  - End User initiated abort
- Get-Attributes
  - Responds with current attribute values

Example: Cancel-Job

Cancel-Job Request



```
job-id: 1234
message: “Please discard any printed
          pages.”
```

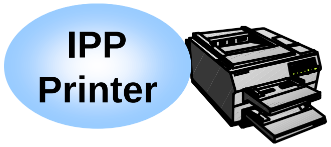
Cancel-Job Response

```
error-code: ok
```

Multiple Document Jobs

- Create-Job
  - Creates a Job object
  - Validates attributes
- Followed by Send-Document
  - Last Document Flag
- Optional support for multiple documents
- Existing practice

Example: Create-Job



Create-Job Request

→ `<attributes only, no data>`

Create-Job Response

```
job-id: 5678
error-code: ok
```
←

Send-Document Request

→ `last-document: false; document data; response: ok`

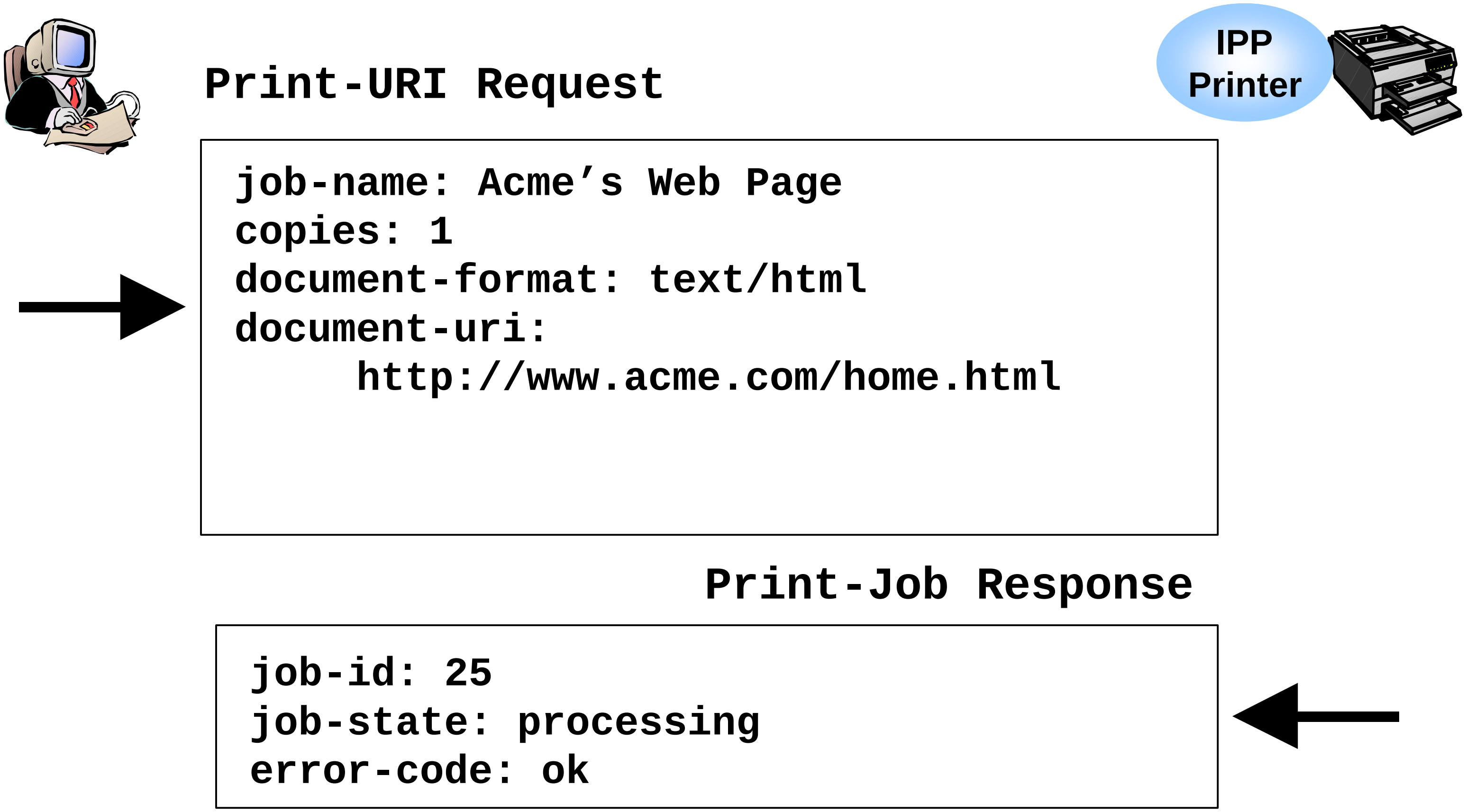
● ● ●

→ `<last-document: true; document data; response: ok`

Print by Reference

- Print-URI operation
  - Client supplies attributes
  - Client supplies URI reference to data
  - Printer retrieves document data
- Optional
- Multiple document
  - “Send-URI”

Example: Print-URI



Print-URI Request

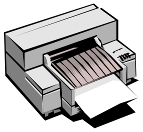
```
job-name: Acme’s Web Page
copies: 1
document-format: text/html
document-uri:
     http://www.acme.com/home.html
```

Print-Job Response

```
job-id: 25
job-state: processing
error-code: ok
```

Job Template Attributes

- Printer supports some values
  - Can be queried
  - Printer has a default value
- Client requests a certain value
- Example: finishings
  - Printer supports: staple, cover, bind
  - Printer default: staple
  - Client requests: cover, bind



Printer Supports: a, b, d

Printer Default: b

Client Requests: a

Job Template (cont.)

- Cover and separator sheets
- Events
- Priority
- Hold
- Media
- Number UP

- Sides
- Copies
- Resolution
- Quality
- Document Format (PDLs)
- Compression

Fidelity Printing

- Semantics
  - “I expect the job exactly as specified - don’t print it if you can’t do it”
  - “Just print the job the best you can - ignore or make substitutions as needed”
- Client supplied “fidelity” attribute

IPP Override of PDL

- Client supplies Job Template attributes
  - Affect rendering, production, and finishing
- Conflicting instructions
  - Embedded PDL instructions
- Example:
  - Request “a4” media for a job that is “letter”
- Printer’s “PDL Override” attribute
  - Values: attempted, not-attempted

Optional Printer Attributes

- Location
- More Info
  - HTML page
  - Site contact info
- Description
- Make and Model
- Driver Installer
- Color Printing Supported
- PDL Override
- State Reasons
- Job Count
- Privacy Supported
- Security Supported

Optional Job Attributes

- More Info
  - HTML page
- Language
- State reasons
- Output device assigned
- Size
  - Octets, impressions
- Time submitted
- Time since
  - Pending
  - Processing
  - Completed
- Number of intervening jobs

Extensibility

- Defined mechanisms
  - Requires an update the specification
  - Requires approval of the PWG
  - Requires IANA registered
  - Allows for site-by-site additions
- Well defined space for
  - Private
  - Experimental

Security

- Does not reinvent the wheel
- Look to HTTP
  - basic, digest, and beyond
- Look to Transport
  - TLS
  - IPSec

IPP Protocol Specification

- Rationale: Encoding
  - Simple, regular, binary
  - Embedded solutions
  - “application/ipp”
- Rationale: HTTP/1.1
  - Ubiquitous (HTTP/1.0 still possible although not optimal)
  - Leverage features (URI naming, chuncking, etc.)
  - Printing has already embraced HTTP servers (administration)
  - Focus on simplicity (complexity relates to proxy servers)
  - Security

Transport Mapping

- Use HTTP/1.1
- Each IPP operation is an HTTP POST
  - “Request URI” is the IPP object
  - “Content Type” is “application/ipp”
  - Authentication/authorization checks
- Leverage HTTP headers:
  - Cache and proxy control
  - Language negotiation

Encoding Diagram: Operations

```
POST http://www.pwg.org/printer1 HTTP/1.1
Content-Type: application/ipp
```

VERSION	2 bytes - required	
OPERATION / STATUS CODE	2 bytes - required	
ATTRIBUTE TAG	1 byte	0 or more
ATTRIBUTE SEQUENCE	n bytes	
DATA TAG	1 byte - required	
DATA (PDL)	m bytes	

Encoding Diagram: Attributes

VALUE TAG	1 byte
NAME LENGTH (value is u)	2 bytes
NAME	u bytes
VALUE LENGTH (value is v)	1 byte
VALUE	v bytes

Value Encoding

Syntax of Attribute Value	Encoding
text	an octet string where each character is a member of the UCS-2 coded character set and is encoded using UTF-8
name	same as text
language	same as text (syntax specified by RFC 1766)
keyword	same as text (allowed text values are defined in the IPP model document)
uri	same as text
uriScheme	same as text
boolean	one binary octet (0x00 is 'false’, and 0x01 is 'true’)

Value Encoding (cont.)

Syntax of Attribute Value	Encoding
integer	a SIGNED-INTEGER
enum	same as integer (allowed integer values are defined in the IPP model document)
dateTime	eleven octets whose contents are defined by "DateAndTime" in RFC 1903
resolution	nine octets consisting of 2 SIGNED-INTEGERs followed by a SIGNED-BYTE (the values are the same as those specified in the Printer MIB)
1setOf X	encoding according to the rules for an attribute with more than more value
rangeOf X	same 1setOf X where the number of values is 2