

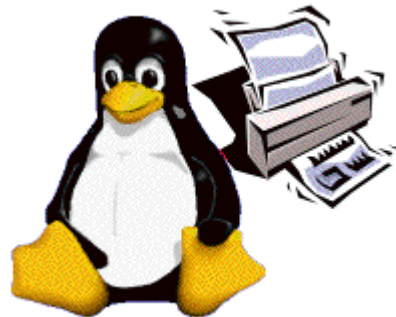


**Free
Standards
Group**

OpenPrinting



*By: Glen W. Petrie
Senior Software Architect
EPSON*





■ Introduction

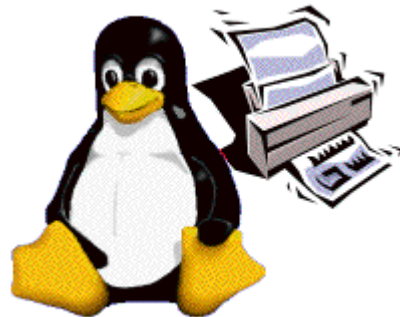
■ Working Group Reviews

- ✗ Architecture
- ✗ Job Ticket
- ✗ Application Interface
- ✗ Driver – Vector/Raster
- ✗ Status Monitoring
- ✗ Print Channel Monitor



**Free
Standards
Group**

Introduction





Organization



Free Standards Group



OpenPrinting

America/Europe

Japan

Steering Committee

Steering Committee

Architecture

Vector Driver

Job Ticket

Status Monitoring

Application Interface

Print Channel Monitor

Raster Driver





Objectives/Mission

“Standardizing on a Scalable Print Environment in Linux.”

Embedded/Mobile ↔ Desktop/Home ↔ Office ↔ Production

Mission Statement

The goal of the OpenPrinting is to

develop and promote a set of standards

that will address the needs of

desktop to enterprise-ready printing;

including

management, reliability, security, scalability, printer feature access and network accessibility.

<http://www.openprinting.org/>



Objectives

■ OpenPrinting : Input

- ✘ OpenPrinting accepts input from interested parties in IT industry, government, education and the open source community

■ OpenPrinting : Output

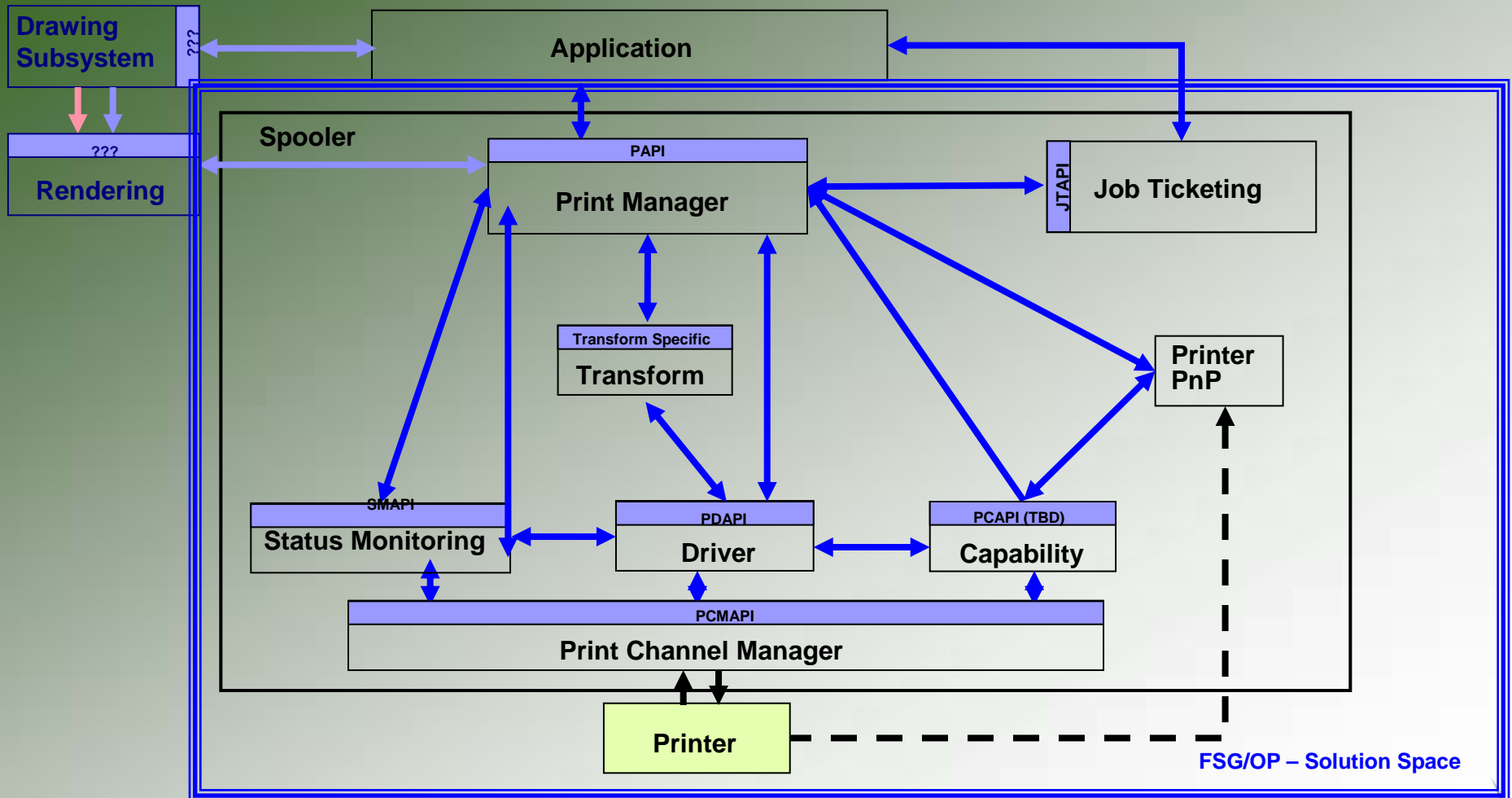
- ✘ Draft specification and implementation released together
- ✘ Public review of work
- ✘ Once implementation has been accepted by *both* upstream authors and at least two Linux distribution vendors, then it is “real”:

A published specification that points to a globally utilized, open source implementation



OpenPrinting Reference Model

Preliminary Reference Model





Working Groups and Objectives

- Steering Committee
 - ✘ Provide overall priorities and top-level coordination.
- Architecture
 - ✘ Develop a modern Print System for Linux.
- Job Ticket API (JTAPI)
 - ✘ To create/consume job tickets; edit job tickets; write/export job tickets.
- Application/Program Interface (PAPI)
 - ✘ Provide applications print services neutral interface.
- Printer Driver (Raster/Vector) (PDAPI)
 - ✘ Provide a neutral interface for printing to any printers
- Status Monitoring API (SMAPI)
 - ✘ Provide a neutral interface for acquiring static and real-time printer status
- Print Channel Monitor API (PCMAPI)
 - ✘ Provide a neutral interface for any output portal/destination.



- **Oct 25-26, 2001 – Print Summit Meeting**
 - ✗ Began discussion of OpenPrinting and needs.

- **April 2003 – FSG Meeting**

- **June 10, 2003 – FSG Portland Face-to-Face Meeting**
 - ✗ Defined Reference Model

- **March 2004 – FSG/Japan-US Team Face-to-Face Meeting**
 - ✗ Refined Reference Model

- **July 4-7, 2004 – Libre Conference – France**

- **Nov 21, 2004 – FSG San Antonio Face-to-Face Meeting**
 - ✗ Working Group Status and Interaction

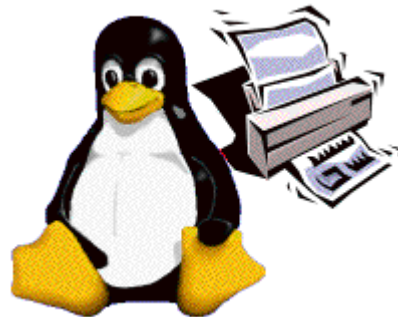
- **Weekly Phone Meetings – all working groups**

- **Email Communications – all working groups**



**Free
Standards
Group**

Steering Committee





Steering Committee & Objectives

■ Steering Committee

- ✘ Divided into two major teams
 - ◆ US/Europe
 - ◆ Japan

■ Objectives

- ✘ Provide oversight to OpenPrinting Activities
- ✘ Provide direction setting to OpenPrinting Activities
- ✘ Provide coordination between OpenPrinting Working Groups
- ✘ Set priorities for OpenPrinting Activities
- ✘ Release Specification/API from OpenPrinting Working Groups
- ✘ Report to FSG Board on OpenPrinting Activities





Steering Committee Information

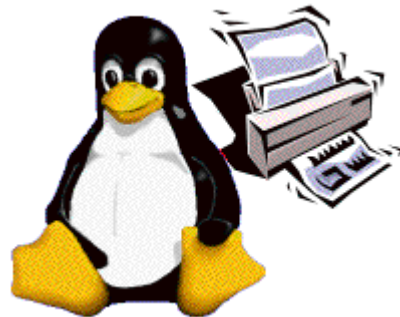
- Monthly FSG Steering Committee conference calls
 - ✘ First Monday of each month at 1:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Steering Committee mailing list:
 - ✘ <http://freestandards.org/mailman/listinfo/printing-sc>
- To post a message to FSG Steering Committee mailing list
 - ✘ printing-architecture@freestandards.org
- To view FSG Steering Committee mailing list archives
 - ✘ <http://freestandards.org/mailman/listinfo/printing-sc>
- To find FSG Architecture documents
 - ✘ ftp://ftp.pwg.org/pub/pwg/fsg/steering_committee
- Participants
 - Japan
 - ✘ Takaaki Higuchi (Sun)
 - ✘ Osamu Mihara (FUJI XEROX Printing Systems)
 - ✘ Keisho Shida (Canon)
 - ✘ Yasumasa Toratani (Canon)
 - ✘ Ide Kentaro (Epson)
 - US / Europe
 - ✘ Norm Jacobs (Sun)
 - ✘ Ira McDonald (High North Inc)
 - ✘ Glen Petrie (Epson)





**Free
Standards
Group**

Architecture





What is OpenPrinting Architecture ?

■ OpenPrinting Architecture is a:

- ✗ Reference Model of the overall solution
- ✗ High-Level Architecture abstracting interfaces between components
- ✗ Detailed Architecture defining interfaces between components

■ Architecture elements contain:

- ✗ Reference Model (RM)
 - ◆ Overall system diagram
 - ◆ Functional decomposition into and description of subsystems.
 - ◆ Identification of data and/or control and/or interface between subsystems and/or external systems.
- ✗ High-Level Architecture (HLA)
 - ◆ A glossary of terminology.
 - ◆ Analysis and documented Use-Cases and requirements.
 - ◆ Identification of applicable/recommended standards for subsystem interfaces.
 - ◆ Identification of recommended infrastructure for integrating products.
- ✗ Detailed Architecture (DA)
 - ◆ Specification of system structure (classes, packages, associations – using UML).
 - ◆ Specification of system behavior (activity and sequence diagrams – using UML).
 - ◆ Identification of the system process structure.
 - ◆ Identification of inter-process communication mechanisms.

■ An OpenPrinting Architecture:

- ✗ Guides development of abstract interfaces.





OpenPrinting Architecture Objectives

- Develop a modern Print System for Linux
- Document realistic Use-Models
- Extract and analyze requirements
- Identify applicable, existing, interface specifications for the requirements
- Apply to multiple print services or be print service neutral.
- Complete and document Reference Model, High-Level Architecture, and Detailed Architecture.





Use-Cases/Use-Models

Example Details Exceptions Diagram Requirements

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- 1. Mobile Printing**
 Mobile printing by reference with document data transformation.
- 2. PDA Printing**
 PDA printing directly with document content.
- 3. Desktop Personal (Consumer) Printing**
 Print to low end inkjet printer from an application.
- 4. Desktop Small-Office/Home-Office Printing**
 Print to mid-volume laser printer, office inkjet and impact printers.
- 5. Office Desktop Printing**
 Document from an application is printed on a printer shared by 3-5 workplaces.
- 6. Central (Print Room) Office Printing**
 Mid to large volume printing from document repository with data transformation and post finishing
- 7. Desktop File (Direct) Printing**
 Print to print device directly form the file-manager/shell-script without going through an application.
- 8. Pay-for-Print Printing**
 Print using a job ticket from a print driver through a print spooler to a mid-range color laser printer.
- 9. Production Printing**
 Large volume transactional printing to high-speed production printer.
- 10. Graphics Art Printing**
 Large volume journal printing to high-speed printing press.





Mobile Printing Use-Model

1.0 Mobile Printing

Mobile printing by reference with document data transformation.

Example Use Model:

Alice goes shopping for a new digital camera to the Fine Camera store downtown. She takes along a magazine review of the new Bright 3000. The sales clerk at the Fine Camera store tells Alice that she would probably prefer the Orion 777. Alice uses her mobile hand-held to browse the Web site of Bright and find the URL for the detailed specs of the Bright 3000. She uses the public access printer in the Fine Camera store to print the Bright 3000 specs. Sure enough, the Orion 777 is a better choice.

Details: (1)

Alice turns on her mobile hand-held and hits the 'I' (Internet) button. The mobile hand-held starts a Web browser application, which connects to Alice's wireless Internet service provider (ISP) over a 14.4Kbps cellular modem. Alice types the Bright URL (from the magazine review) into her Web browser and hits the Enter key. The Web browser connects to Bright's home page over the Internet (via the HTTP proxy in the cellular ISP's firewall). Alice searches for the Bright 3000 specs (available in HTML) and copies the URL into her GUI clipboard.

Alice hits the 'P' (Print) button. The mobile hand-held starts a print application, which discovers the store's public access printer over Bluetooth using FSG/OP PAPI (which does Bluetooth device discovery) and then forms an ad-hoc wireless Personal Area Network (PAN) with the Target Printer.

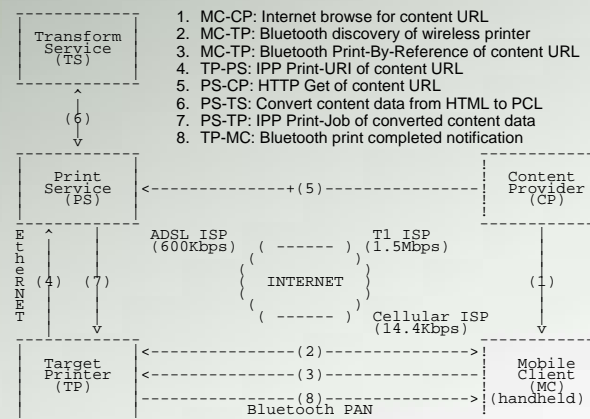


Requirements for this Use Model:

The FSP/OP Architecture MUST support:

- ◆ Administrative configuration of Print Services and their associations with Target Printers using FSG/OP PAPI;
- ◆ Administrative configuration of Target Printers and their associations with Print Services using FSG/OP PAPI;
- ◆ Dynamic discovery of network (IP, Bluetooth, IRDA, etc.) Print Services and Target Printers using FSG/OP PAPI;
- ◆ Multiple print protocol bindings of FSG/OP PAPI (for direct-connect and non-IP network printing);
- ◆ Print-by-reference operations (for example, printing from low-bandwidth mobile devices);
- ◆ Print-by-value operations (for example, printing of converted or local content);
- ◆ Content conversion by FSG/OP Transform Services (for example, reference printing);
- ◆ Job and Printer event notification (for example, job complete, printer intervention required, etc.).

Use Model Diagram:



What's next - schedule



❑ OpenPrinting Requirements Document

- ✗ Aug 2004 - Complete prioritization of identified sub-systems
- ✗ Sept 2004 - Complete Reference Model Alpha Release

**The Architecture has decided to move all discussion to email.
Activities to resume in December**

- ✗ 2005 - Integrate Use Models from FSG-Japan Use-Models
- ✗ 2005 - Complete Detailed Description for Use Models
- ✗ 2005 - Complete Requirements for Use Models
- ✗ 2005 - Complete Common Requirements & Reference Model 1.0

❑ High-Level Architecture/Reference Model

- ✗ 2005 - Completed document

Glen Petrie assigned to complete Reference Model.





Architecture Working Group Information

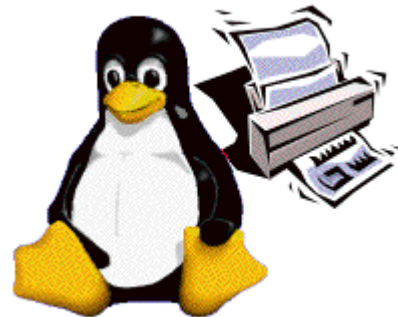
- Weekly FSG Architecture conference calls
 - ✗ Thursday at 3:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Architecture mailing list:
 - ✗ <http://freestandards.org/mailman/listinfo/printing-architecture>
- To post a message to FSG Architecture mailing list
 - ✗ printing-architecture@freestandards.org
- To view FSG Architecture mailing list archives
 - ✗ <http://freestandards.org/mailman/listinfo/printing-architecture>
- To find FSG Architecture documents
 - ✗ <ftp://ftp.pwg.org/pub/pwg/fsg/architecture/>
- Participants
 - ✗ Claudia Alimpich (IBM)
 - ✗ Jody Goldberg (Gnome)
 - ✗ Mark Hamzy (IBM)
 - ✗ Tom Hastings (Xerox)
 - ✗ Norm Jacobs (Sun)
 - ✗ Till Kamppeter (MandrakeSoft)
 - ✗ Ira McDonald (High North Inc)
 - ✗ Glen Petrie (Epson)
 - ✗ Pete Zannucci (IBM)





**Free
Standards
Group**

Job Ticket





JTAPI & Objectives

What is JTAPI ?

✘ JTAPI stands for:

- ◆ Job Ticket Application Programming Interface
- ◆ Pronounced “jay-tappy”, “Job Ticket API”, or “jay tee API”

✘ A job ticket contains:

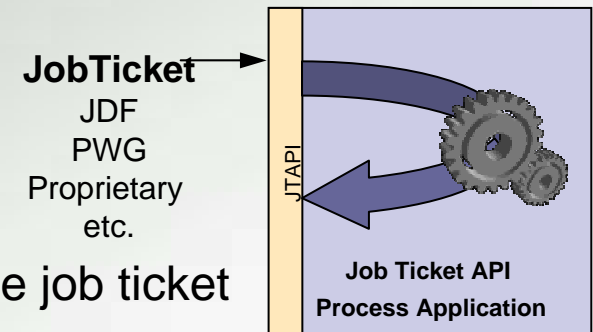
- ◆ **Instructions** describing how to process and/or print a job
- ◆ **Information** about the results of a job as it is processed and/or printed

✘ A JTAPI job ticket is:

- ◆ an electronic replacement of manual hard copy instructions and information

Objectives

- ✘ To create and consume job tickets
- ✘ To be job ticket syntax neutral
- ✘ To isolate the application from the content of the job ticket
- ✘ To be programming language neutral
- ✘ To import and export multiple job ticket formats





Existing Job Ticket Formats

Technical Review

■ CIP4 JDF (Job Definition Format) Job Ticket

- ✗ Defined by CIP4, a world wide standards body with over 150 members
- ✗ Is open, extensible, XML-based job ticket standard
- ✗ JDF Specification versions
 - 1.0 released April 2001
 - 1.1 released April 2002
 - 1.2 released May 2004
 - 1.3 to be released Mid 2005

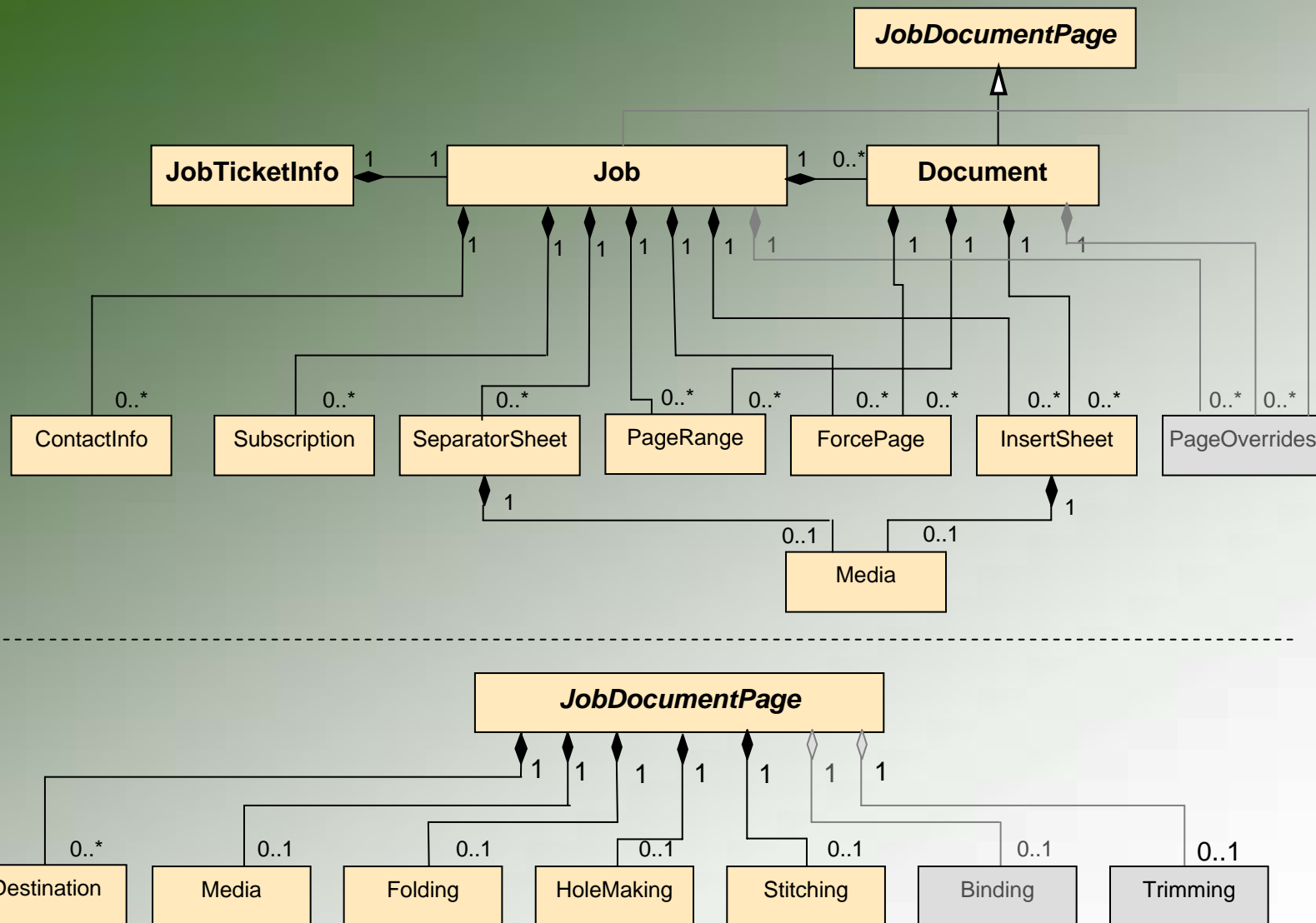
■ PWG Job Ticket

- ✗ Defined by PWG Semantic Model 1.0
- ✗ Is open, extensible, XML-based job ticket standard





JTAPI Object Model





JTAPI Version 1.0 Release

■ Completed C Header Files

- ✗ Each object in separate files
- ✗ Common extensible method for attributes
- ✗ Data/object model that is object oriented
- ✗ Defines objects that are familiar to the printing industry
 - ◆ Job, Document, Insert Sheet, Media, Stitching, Hole Making, etc.
- ✗ Defines relationship between objects
- ✗ Defines operations to be performed on objects
- ✗ Defines attributes of objects
- ✗ Defines well-known enumerated values of all attributes





JTAPI – Version 1.0 Release Details (1)

JobTicketInfo

- ✘ **fsgjtNewJobTicketInfo**
 - ◆ Creates a JobTicketInfo object. Used when a new job ticket is being created without a Job object and not from an existing job ticket.
- ✘ **fsgjtNewJobTicketInfoFromURI**
 - ◆ Creates a JobTicketInfo object from the job ticket file at the provided URI. Other JTAPI objects are also created based on the information in the job ticket file.
- ✘ **fsgjtNewJobTicketInfoFromBuffer**
 - ◆ Creates a JobTicketInfo object from the provided buffer that contains a job ticket. Other JTAPI objects are also created based on the information in the job ticket buffer.
- ✘ **fsgjtNewJobTicketInfoFromJob**
 - ◆ Creates a JobTicketInfo object using the provided Job object. Used when a new job ticket is being created. The Job object must be created first.
- ✘ **fsgjtWriteJobTicketToBuffer**
 - ◆ Writes a job ticket to the provided buffer.
- ✘ **fsgjtWriteJobTicketToURI**
 - ◆ Writes a job ticket at the provided URI.

Job

- ✘ **fsgjtNewJob**
 - ◆ Creates a Job without providing a Document.
- ✘ **fsgjtNewJobFromDocument**
 - ◆ Creates a Job containing the provided Document

Document

- ✘ **fsgjtNewDocument**
 - ◆ Creates a Document object.
- ✘ **fsgjtNewDocumentFromURI**
 - ◆ Creates a Document object using the provided URI that contains the document data.





JTAPI –Version 1.0 Release Details (2)

■ ContactInfo

✘ fsgjtNewContactInfo

- ◆ Creates a ContactInfo object having the specified name.

■ Subscription

✘ fsgjtNewSubscription

- ◆ Creates a Subscription object having the specified notification URI.

✘ fsgjtNewSubscriptionForEvent

- ◆ Creates a Subscription object having the provided notification URI and event.

■ SeparatorSheet

✘ fsgjtNewSeparatorSheet

- ◆ Creates a SeparatorSheet object to be placed at the provided location.

■ ForcePage

✘ fsgjtNewForcePage

- ◆ Creates a ForcePage object having the specified page and sheet side.

■ InsertSheet

✘ fsgjtNewInsertSheet

- ◆ Creates a InsertSheet object.

■ Destination

✘ fsgjtNewDestination

- ◆ Creates a Destination object with a specific URI.





JTAPI – Version 1.0 Release Details (3)

Media

✘ fsgjtNewMedia

- Creates a Media object using the specified name.

Folding

✘ fsgjtNewFolding

- Creates a Folding object having the specified folding type.

HoleMaking

✘ fsgjtNewHoleMaking

- Creates a HoleMaking object having the specified hole count and reference edge.

Stitching

✘ fsgjtNewStitching

- Creates a Stitching object having the specified stitch type.

PageRange

✘ fsgjtNewRangRange

- Creates a PageRange object.





JTAPI – Version 1.0 Release Details (4)

Attribute

Generic support for all object/attributes

x fsgjtNewAttribute

- ◆ Creates a new Attribute object having the provided attribute name, value type, and value.

x fsgjtDestroyAttribute

- ◆ Free the memory used by the Attribute.

x fsgjtAddValue

- ◆ Add an additional value to this Attribute.

x fsgjtGetName

- ◆ Get the name of the Attribute.

x fsgjtGetNextValue

- ◆ Return the Attribute's next value.

x fsgjtGetNumValues

- ◆ Get the number of values that the Attribute contains.

x fsgjtGetValueType

- ◆ Returns the type of the Attribute's values.

x fsgjtReplaceValue

- ◆ Replaces the existing value(s) for this Attribute.

x fsgjtResetToFirstValue

- ◆ Reset the iterator to point to the first of the Attribute.





JTAPI – Version 1.0 Release Details (5)

❏ Miscellaneous / Helper

- ✘ **fsgjtDestory**
 - ◆ Free the memory used by the an object.
- ✘ **fsgjtGet**
 - ◆ Get the Attribute having the specific name.
- ✘ **fsgjtSet**
 - ◆ Set the Attribute having the specific name.
- ✘ **fsgjtSetIntegerAttribute**
 - ◆ Convenience function for setting an integer attribute.
- ✘ **fsgjtSetObjectAttribute**
 - ◆ Convenience function for setting an object attribute.
- ✘ **fsgjtSetObjectAttributeList**
 - ◆ Convenience function for setting an object list.
- ✘ **fsgjtSetStringAttribute**
 - ◆ Convenience function for setting a string attribute.
- ✘ **fsgjtSetStringAttributeList**
 - ◆ Convenience function for setting a string list.





JTAPI - Version 1.0 – Enumerations

Enumerations

- ✘ BindTypeEnum
- ✘ BooleanEnum*
- ✘ CollateEnum
- ✘ CompressionEnum
- ✘ ContactInfoRoleEnum
- ✘ FeedOrientationEnum
- ✘ FitPolicyEnum
- ✘ FoldTypeEnum
- ✘ HoldEnum
- ✘ ImageAlignmentXEnum
- ✘ ImageAlignmentYEnum
- ✘ InputTrayNameEnum*
- ✘ InsertSheetContentEnum
- ✘ JobTicketTypeVersionEnum*
- ✘ JogOffsetEnum
- ✘ LengthUnitEnum*
- ✘ MandatoryAttributesEnum
- ✘ MediaCoatingEnum
- ✘ MediaColorEnum
- ✘ MediaPrePrintedEnum
- ✘ MediaTypeEnum*
- ✘ OutputBinEnum
- ✘ PageDeliveryEnum
- ✘ PositionEnum
- ✘ PresentEnum
- ✘ PresentationDirectionEnum
- ✘ PrintContentOptimizeEnum
- ✘ PrintQualityEnum*
- ✘ ReferenceEdgeEnum
- ✘ RotationEnum*
- ✘ SeparatorSheetEnum
- ✘ SheetSideEnum
- ✘ SidesEnum*
- ✘ StitchingTypeEnum*
- ✘ SubscriptionEventEnum
- ✘ TrimmingTypeEnum
- ✘ ValueTypeEnum*





Accomplishments 2002

Introduction

- Feb 2002
 - ✗ Began job ticket discussions in FSG Open Print

- June 2002
 - ✗ Initial JTAPI proposal
 - ✗ Chartered FSG JT working group

- Nov 2002
 - ✗ Created IPP to JDF mapping table
 - ◆ In cooperation with PODi and CIP4 Digital Printing working groups

- Dec 2002
 - ✗ Prioritized features/functions of JTAPI
 - ✗ IBM ships first generation C JTAPI product
 - ◆ In a job submission GUI and printer control unit product
 - Based on initial JTAPI specification
 - Based on early version of JDF ICS Specification for Digital Printing





Accomplishments 2003

Introduction

- **May 2003**
 - ✗ Completed detailed JTAPI UML diagrams
 - ◆ 19 objects and 33 enumerations
 - ◆ 20+ draft versions

- **June 2003**
 - ✗ Defined subset of JTAPI 1.0 content for Alpha version

- **Nov 2003**
 - ✗ Completed version 0.82 JTAPI Specification
 - ✗ Started C header files
 - ◆ Initial contribution from IBM

- **Dec 2003**
 - ✗ IBM ships second generation C JTAPI product
 - ✗ IBM ships first generation Java JTAPI product



Accomplishments 2004



❑ JTAPI Software Development White Paper

- ✘ May 2004 - Draft white paper completed
- ✘ May 2004 - Completed and distributed to FSG/Japan

❑ JTAPI Version 1.0 Specification

- ✘ Oct 2004 - Updated JTAPI global UML diagrams
- ✘ Oct 2004 - Updated JTAPI Version 1.0 compliant C header files
- ✘ Oct 2004 - Updated Version 1.0 Specification
- ✘ Nov 2004 - Review Specification

❑ JTAPI Version C Header Files

- ✘ Nov 2004 - Review headers
- ✘ Dec 2004 - Finalize requirement / definition document

❑ JTAPI Version 1.0 Specification/Header Release

- ✘ Apr 2005 - Release Specification and headers
- ✘ Jun 2005 - Public Review of JTAPI 1.0



JTAPI Status July 2005



- JTAPI/1.0 Spec Completed
 - ✘ ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/JTAPI_Spec/fsg-openprinting-job-ticket-api-v0100-20050315.pdf

- JTAPI/1.0 C Headers Completed
 - ✘ ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/C_JTAPI_Code/CJTAPI_22Apr2005.zip

- JTAPI/1.0 UML Completed
 - ✘ ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/JTAPI_Diagrams/05Apr2005/*.png

- JTAPI Release Request Initiated
 - ✘ ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/Release_Request/JTAPI-Release-Request--V0102-2005-0422.pdf
 - ✘ Release Champion: Norm Jacobs (Sun)
 - ✘ Public Review open until 5 July 2005
 - ◆ No comments to date.

- WG Directory
 - ✘ <ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/>

- Slides – 2004 Face-to-Face Meeting
 - ✘ ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/Nov2004_San_Antonio_meeting/FSGOpenPrint_JTAPI_Nov2004_San_Antonio.pdf





JT Working Group Information

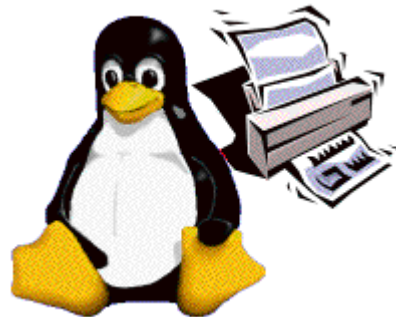
- Weekly FSG Job Ticket conference calls
 - ✗ Tuesdays at 3:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Job Ticket mailing list:
 - ✗ <http://freestandards.org/mailman/listinfo/printing-jobticket>
- To post a message to FSG Job Ticket mailing list
 - ✗ printing-jobticket@freestandards.org
- To view FSG Job Ticket mailing list archives
 - ✗ <http://freestandards.org/mailman/listinfo/printing-jobticket>
- To find FSG Job Ticket documents
 - ✗ <ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/>
- Participants
 - ✗ Claudia Alimpich (IBM) – chair
 - ✗ Jody Goldberg (Gnome)
 - ✗ Tom Hastings (Xerox)
 - ✗ Till Kampeter (Mandrakesoft)
 - ✗ Ira McDonald (High North)
 - ✗ Glen Petrie (Epson)





**Free
Standards
Group**

Application Interface





PAPI & Objectives

■ PAPI stands for:

- ✗ Printing Application Programming Interface
- ✗ Pronounced “pappy” or “P.A.P.I.”

■ PAPI contains:

- ✗ **Data Structures** for interacting with basic print service objects.
- ✗ **Functions** to perform various operations against a print service or its objects.

■ PAPI is an API for applications to use to perform print service interaction.

■ Objectives

- ✗ Provide applications a means of interacting with print services without being tied to a particular service or protocol
- ✗ Allow a rich, extensible set of information to flow between application and print service
- ✗ Support a rich enough set of operations to be useful to most applications with printing needs





PAPI Overview

- Provides abstraction of server, queues (printers and classes), and jobs.
- High-level API focuses on providing a common interface to multiple printing systems and protocols.
- Uses extensible attribute sets based on IPP for representing queue and job information.
 - ✘ Multi-valued and typed
- Currently supports
 - ✘ Server connection (and authentication were applicable)
 - ✘ Basic queue operations:
 - ◆ enumerate, query, enable, disable, modify, enumerate/purge jobs
 - ✘ Basic job operations:
 - ◆ submit, stream, query, cancel, hold, release, modify





PAPI Details

■ Data types introduced by the PAPI:

- ✗ **papi_status_t** status/error code enumeration
- ✗ **papi_service_t** server/service connection
- ✗ **papi_printer_t** printer object/handle/context
- ✗ **papi_job_t** job object/handle/context
- ✗ **papi_attribute_t** object attributes
- ✗ **papi_stream_t** stream object/handle/context
- ✗ **papi_job_ticket_t** job ticket
- ✗ **papi_filter_t** filter for limiting printer lists
- ✗ **papi_encrypt_t** get/set the encryption type

■ Attribute API :

- ✗ **papiAttributeListAdd***
 - ◆ Adds a single attribute (and value) to an attribute list
- ✗ **papiAttributeListDelete**
 - ◆ Delete an attribute from an attribute list.
- ✗ **papiAttributeListGet***
 - ◆ Retrieve the value associates with an attribute
- ✗ **papiAttributeListFind**
 - ◆ Search for an attribute in a list
- ✗ **papiAttributeListGetNext**
 - ◆ Walk through an attribute list
- ✗ **papiAttributeListFromString**
 - ◆ create/append an attribute list using a text-based format
- ✗ **PapiAttributeListToString**
 - ◆ Convert an attribute list to a text base string form
- ✗ **papiAttributeListFree**
 - ◆ Deallocate resources associated with an attribute list

■ Service API

- ✗ **papiServiceCreate**
 - ◆ create a new service object for submitting jobs, listing printers, etc
- ✗ **papiServiceDestroy**
 - ◆ destroys a service object and all associated resources
- ✗ **papiGet*() / papiSet*()**
 - ◆ Get or set the corresponding service object attributes

* = Supported object attributes UserName, Password, Encryption, AuthCB, AppData, AttributeList, StatusMessage

■ Miscellaneous API :

- ✗ **papiStatusString**
 - ◆ Return a textual representation of a papi_status_enumeration
- ✗ **papiLibrarySupportedCalls**
 - ◆ Enumerate all papiCalls that do not simply return PAPI_OPERATION_NOT_SUPPORTED
- ✗ **papiLibrarySupportedCall**
 - ◆ Determine if a specific function is supported by the implementation

* = Supported types of value, string, integer, boolean, range, resolution, datetime, collection, metadata





PAPI Details

Job API:

- ✘ **papiJobSubmit / papiJobSubmitByReference**
 - ◆ Submit a print job either immediately copying or postponing copy of the job data
- ✘ **papiJobValidate**
 - ◆ Verify that the document format and job attributes are valid for and supported by the designated print queue
- ✘ **papiJobStreamOpen / papiJobStreamWrite / papiJobStreamClose**
 - ◆ Open/write/close a stream for printing data on-the-fly
- ✘ **papiJobQuery**
 - ◆ Request job information from print service
- ✘ **papiJobModify**
 - ◆ Modifies the job attributes
- ✘ **papiJobCancel**
 - ◆ Cancels a print job
- ✘ **papiJobHold / papiJobRelease / papiJobRestart**
 - ◆ Hold, release, or restart a print job
 - ◆ Retrieve the corresponding attribute list from a job object for further lookup or enumeration using the Attribute interface
- ✘ **PapiJobPromote**
 - ◆ Promote a print job
- ✘ **papiJobGetAttributeList**
 - ◆ Retrieve the corresponding attribute list from a job object for further lookup or enumeration using the Attribute interface
- ✘ **papiJobGet***
 - ◆ Retrieve the corresponding attribute from a job object
- ✘ **papiJobListFree / papiJobFree**
 - ◆ Free a job object of list of job objects and their associated resources

Printer API :

- ✘ **papiPrintersList**
 - ◆ Retrieves a list of printer/class queues
- ✘ **papiPrinterQuery**
 - ◆ Queries for detailed information about a printer object
- ✘ **PapiPrinterAdd / papiPrinterRemove**
 - ◆ Creates/destroys printer object
- ✘ **papiPrinterModify**
 - ◆ Sets printer object attributes
- ✘ **papiPrinterPause / papiPrinterResume**
 - ◆ Stop and start job processing on a printer object
- ✘ **papiPrinterEnable / papiPrinterDisable**
 - ◆ Stop and start job queueing on a printer object
- ✘ **papiPrinterPurgeJobs**
 - ◆ Cancels all jobs on a printer
- ✘ **papiPrinterListJobs**
 - ◆ Lists jobs on a printer
- ✘ **papiPrinterGetAttributeList**
 - ◆ Retrieve an attribute list from printer object
- ✘ **papiPrinterListFree / papiPrinterFree**
 - ◆ Deallocate resources associated with a printer object or list of printer objects

* = Supported attributes are PrinterName, Id or JobTicket



PAPI Release

- Released PAPI based code on SourceForge
 - ✗ PAPI dynamic library
 - ✗ PAPI over CUPS
 - ✗ BSD/SYSV print commands implemented on top of the PAPI
 - ✗ Released PAPI over RFC-1179 support

- What's next
 - ✗ Expand functionality in PAPI
 - ◆ Administrative operations
 - ◆ Document object
 - ✗ Implement and Release PAPI over IPP support
 - ✗ Release IPP server implemented over PAPI
 - ✗ Integrate support for various Open Source projects:
 - ◆ GNOME, KDE, Open Office, Mozilla, Samba, etc.



PAPI Status July 2005

- PAPI/1.0 Spec Completed
 - ✘ <ftp://ftp.pwg.org/pub/pwg/fsg/jobticket/spool/papi-v1.0-2005-05-09.pdf>

- PAPI Release Request Initiated
 - ✘ <ftp://ftp.pwg.org/pub/pwg/fsg/spool/papi-v1.0-release-request-v1.01-2005-06-06.sxw>
 - ✘ Release Champion: Glen Petrie (Epson)
 - ✘ Public Review open until 15 July 2005
 - No comments to date.

- WG Directory
 - ✘ <ftp://ftp.pwg.org/pub/pwg/fsg/spool>

- Slides – 2004
 - ✘ ftp://ftp.pwg.org/pub/pwg/fsg/July2004_Libre-Conference/FSGOpenPrint_Libre_2004-0706.pdf





PAPI Working Group Information

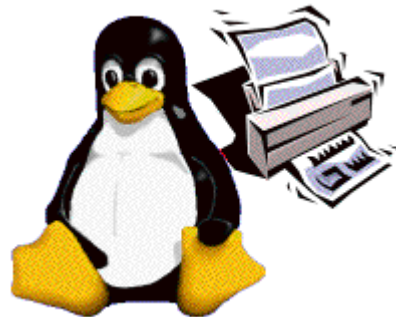
- To subscribe to FSG PAPI mailing list:
 - ✘ <http://freestandards.org/mailman/listinfo/printing-spool>
- To post a message to FSG PAPI mailing list
 - ✘ printing-spool@freestandards.org
- To view FSG PAPI mailing list archives
 - ✘ <http://freestandards.org/mailman/listinfo/printing-spool>
- To find FSG PAPI documents
 - ✘ <http://sourceforge.net/projects/openprinting>
- Participants
 - ✘ Norm Jacobs (Sun) – chair
 - ✘ Alan Hlava (IBM)
 - ✘ Mike Sweet (Easy Software)
 - ✘ Ira McDonald (High North Inc)
 - ✘ Glen Petrie (Epson)





**Free
Standards
Group**

Printer Driver





Printer Driver & Objectives

■ Printer Driver API is:

- ✗ A printer driver interface for requesting driver/printer information and accepting/printing print jobs.
 - ◆ Vector API's supporting PDL based printers
 - ◆ Raster API's supporting Raster based printers

■ Printer Driver API contains:

- ✗ Commands to query/set capabilities
- ✗ Commands to create and control print jobs
- ✗ Vector / Raster transfer commands

■ Objectives

- ✗ To be a common interface for printing to printers
- ✗ To isolate the application from the details of individual printers
- ✗ To isolate the application from the details of individual PDLs
- ✗ To have printer drivers support a set of common job properties
- ✗ Performance Optimization
 - ◆ Achieve full speed printing
 - ◆ Utilizes graphical acceleration feature supported by printer controllers





Printer Driver API

- Job Control
 - ✗ Open/Close driver
 - ✗ Set Job/Document/Page attributes

- Graphics State Operation
 - ✗ Set attributes for each graphics objects

- Drawing Operations
 - ✗ Path
 - ✗ Text
 - ✗ Bitmap Image
 - ✗ Scanline
 - ✗ Raster Image

- Stream Data (embedded PDL)





Printer Driver API – Details (1)

Printer Context

- ✗ **OpenPrinter()**
 - ◆ Create printer context
 - ◆ Register API entry pointers
 - ◆ Specify file descriptor for data stream
- ✗ **ClosePrinter()**
 - ◆ Closes printer context
 - ◆ Driver releases all resources

Job Control

- ✗ A print job consist of documents.
- ✗ A document consist of pages.
- ✗ **StartJob(), EndJob()**
- ✗ **StartDoc(), EndDoc()**
- ✗ **StartPage(), EndPage()**
- ✗ Job, doc and page attributes are specified by each **StartXxx()** function.

Query Device Capabilities & Information

- ✗ **QueryDeviceCapability()**
 - ◆ Query if the device can do number-up, duplex, etc.
 - ◆ Information such as media size, media source and etc. which are supported by the device can be retrieved.
- ✗ **QueryDeviceInfo()**
 - ◆ Query current settings of the device.

Graphics State Object Operations

- ✗ Graphics State is managed as GS object
 - ◆ Operation to GS – **InitGS, SaveGS, RestoreGS**
- ✗ Controls to each items in GS
 - ◆ CTM (Coordinate Translate Matrix)
 - ◆ Color Space
 - ◆ Raster Operation – ROP3
 - ◆ Fill Mode – even/odd or winding
 - ◆ Alpha Constant
 - ◆ Line Style – width, dash/solid, cap, join
 - ◆ Paint Mode – opaque or transparent
 - ◆ Stroke and fill color – brush control
 - ◆ Foreground and background color – solid brush

Path Operations

- ✗ A path is a virtual track object
 - ◆ Will be visible by stroke or fill operations
 - ◆ Will be used to define clip region
- ✗ Lines, rectangles, polygons, arc/pie and bezier are all treated as “path.”
- ✗ Operations:
 - ◆ **NewPath()** – Declare start of a path
 - ◆ **EndPath()** – Declare end of a path
 - ◆ **StrokePath(), FillPath(), StrokeFillPath()** – make visible path
 - ◆ **SetClipPath(), ResetClipPath()** – defines clip region by current path





Printer Driver API – Details (2)

✦ Text Operations

- ✗ Still under investigation...
- ✗ Current DrawBitmapText() will be removed.
- ✗ Text Operations will includes:
 - Define and Query font metrics
 - Device Font Utilization
 - Font Downloading

✦ Bitmap and Scanline Operations

- ✗ Bitmap is a bit oriented image data drawn in rectangle region
 - DrawImage()
 - StartDrawImage(), TransferDrawImage(), EndDrawImage()
- ✗ Scanline is a horizontal line defined by start and end point pairs.
 - Used to draw graphics rendered by renderer
 - StartScanLine(), ScanLine(), EndScanLine()

✦ Raster Image Operations

- ✗ StartRaster(), TransferRasterData(), EndRaster()
- ✗ Set to be extended by Raster Team

✦ Stream Data Operations

- ✗ StartStream(), TransferStreamData(), EndStream()



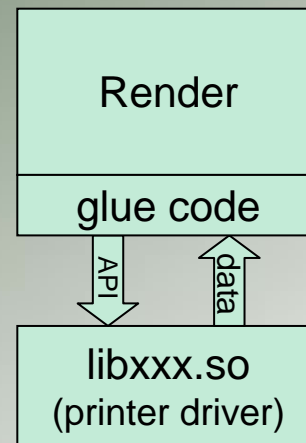


Linking

- Printer driver is provided as a dynamic library.
- Driver can be linked dynamically or via RPC.
 - ✗ avoids license problem

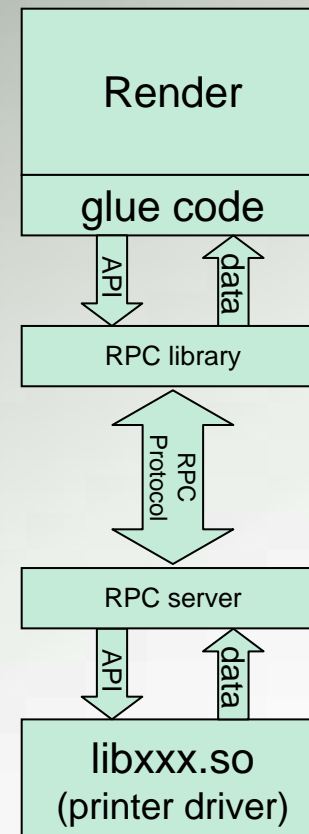
direct linking

R: GPL
D: GPL
or
R: MIT
D: Closed or LGPL



RPC linking

R: any
D: any





PDAPI Raster- Common Job Properties

Features

- ✘ Standardized name for common features
 - ✘ Standardized keys and the values
 - ✘ An extensible paradigm for non-standard features
 - ✘ Coherence across the FSG OpenPrinting model
-
- | | |
|---------------------------------|--------------------------|
| ✘ ColorInput | ✘ PrintQuality |
| ✘ ColorOutput | ✘ Resolution |
| ✘ Copies | ✘ Rotation |
| ✘ Margins | ✘ ScalingType |
| ✘ MediaBackCoating | ✘ ScalingPercentage |
| ✘ MediaColor | ✘ SheetCollate |
| ✘ MediaFrontCoating | ✘ Sides |
| ✘ MediaInputTrayName | ✘ StitchingPosition |
| ✘ MediaSizeName | ✘ StitchingReferenceEdge |
| ✘ MediaType | ✘ StitchingType |
| ✘ MediaUnprintableMargins | ✘ StitchingCount |
| ✘ NumberUp | ✘ StitchingAngle |
| ✘ NumberUpPresentationDirection | ✘ Trimming |
| ✘ OutputBinName | |





PD-Vector Working Group Information

- To subscribe to FSG Vector Printer Driver mailing list:
 - ✘ <http://freestandards.org/mailman/listinfo/printing-japan>

- To post a message to FSG Vector Printer Driver mailing list
 - ✘ printing-driver@freestandards.org

- To view FSG Vector Printer Driver mailing list archives
 - ✘ <http://freestandards.org/mailman/listinfo/printing-japan>

- To find FSG Vector Printer Driver documents
 - ✘ <ftp://ftp.pwg.org/pub/pwg/fsg/vector/>

- Participants
 - ✘ Osamu Mihara (Fuji Xerox Printing Systems Co., Ltd)





PD-Raster Working Group Information

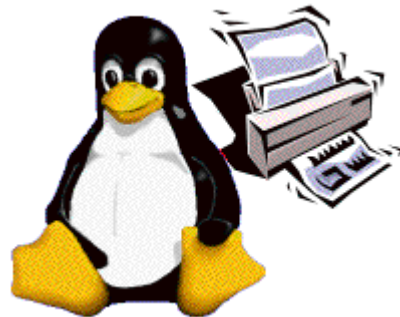
- Weekly FSG Printer Driver conference calls
 - ✗ Mondays at 2:00 PM US Eastern for 1-2 hours
- To subscribe to FSG Printer Driver mailing list:
 - ✗ <http://freestandards.org/mailman/listinfo/printing-driver>
- To post a message to FSG Printer Driver mailing list
 - ✗ printing-driver@freestandards.org
- To view FSG Printer Driver mailing list archives
 - ✗ <http://freestandards.org/mailman/listinfo/printing-driver>
- To find FSG Printer Driver documents
 - ✗ <ftp://ftp.pwg.org/pub/pwg/fsg/driver/>
- Participants
 - ✗ *Mark Hamzy (IBM)*
 - ✗ *Till Kamppeter (MandrakeSoft)*
 - ✗ *Glen Petrie (Epson)*





**Free
Standards
Group**

Status Monitoring





Status Monitoring & Objectives

■ Status Monitoring API is:

- ✗ A neutral interface for acquiring static and real-time printer status

■ Status Monitoring API contains:

- ✗ Commands to open/close a status monitoring channel.
- ✗ Commands to translate device-dependent status data to generic format data.
- ✗ Commands to handle events from a printer.

■ Objectives

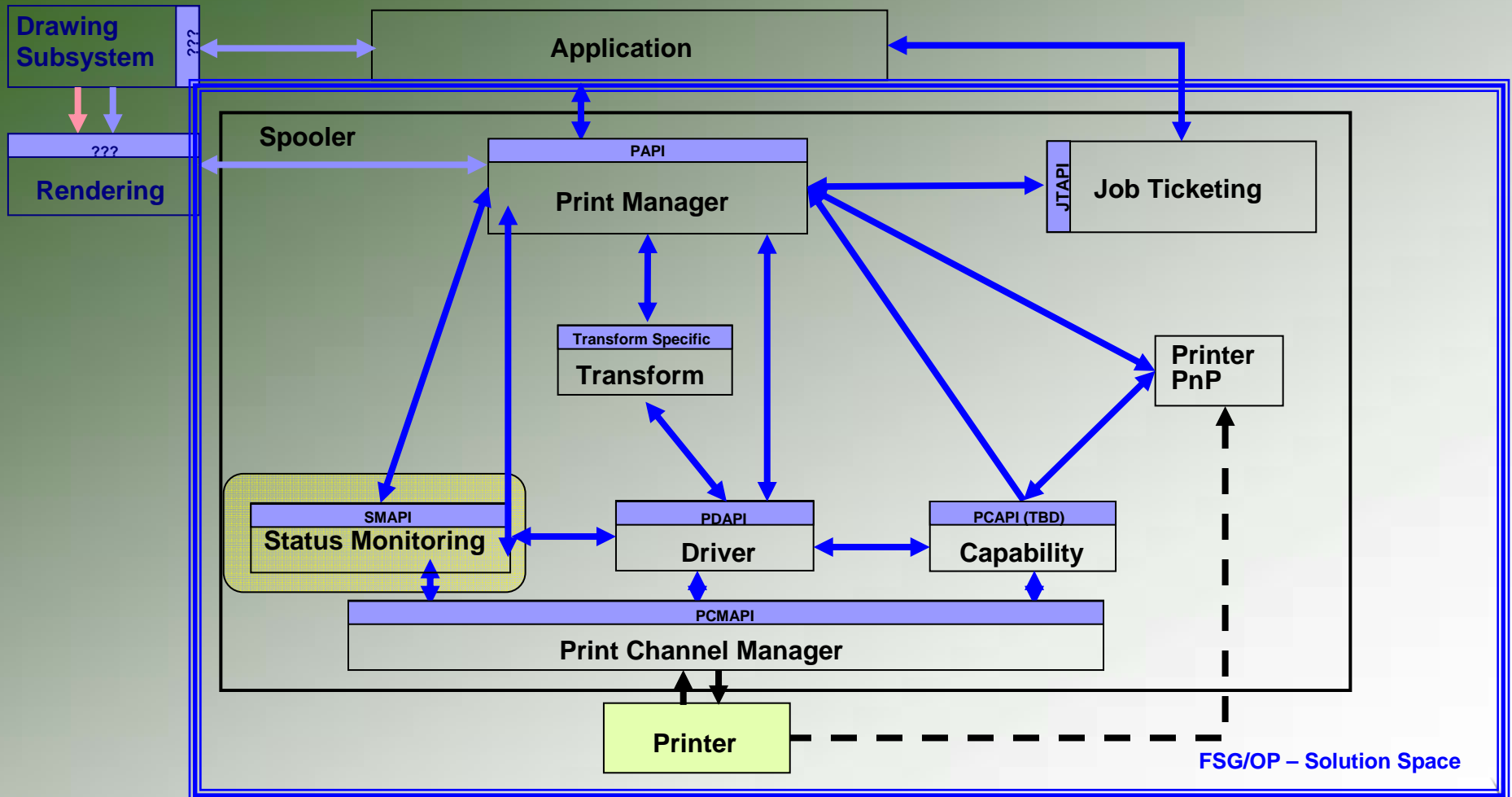
- ✗ To isolate software elements from differences due to device-dependent printer status format





OpenPrinting Reference Model

Preliminary Reference Model





Status Monitoring API

■ Channel Control

✘ New

- ◆ Create an object, and open a status monitoring channel via PCM.

✘ Destroy

- ◆ Close the status monitoring channel, and destroy the object.

■ Obtaining Status

✘ ReadStatus

- ◆ Read a printer status data via PCM, translate it to a generic format data, and store it into the buffer given by the caller.

■ Event Handling

✘ TBD





Status Monitoring API – Details (1)

■ Detail of API

✘ TBD

- We've decided that the bi-directional printer communication interface should be divided into two groups, PCM for wrapping low level I/O dependencies, and SM for wrapping status data format dependencies.
- Status Monitoring API depends on the PCM API specification.
- We will re-start our discussion after the first draft of PCM API finished and authorized in OpenPrinting WG.





- Discussion about SMAPI
 - 2003/9 First pre-draft of the SMAPI specifications.
 - 2004/3 Open to working group.
 - 2004/4 Divide bi-di functions into two groups, PCMAPI and SMAPI.

- Discussion about SMAPI
 - ✗ Printing-Japan ML
 - ✗ 2003 - 2004 Face to Face meeting – Monthly
 - ✗ 2005 Focused on the PCM activity.

- Milestone
 - ✗ 2005/2Q First draft and review

- To-Do
 - ✗ Define APIs in detail
 - ✗ Consider the event handling





SM Working Group Information

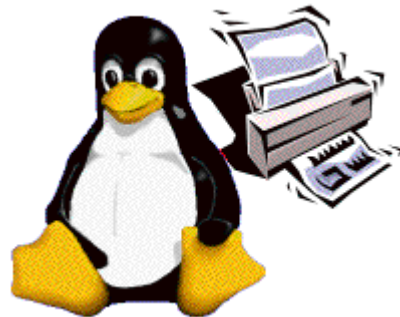
- To subscribe to FSG SM mailing list:
 - ✘ <http://freestandards.org/mailman/listinfo/printing-japan>
- To post a message to FSG SM mailing list
 - ✘ printing-japan@freestandards.org
- To view FSG SM mailing list archives
 - ✘ <http://freestandards.org/mailman/listinfo/printing-japan>
- To find FSG SM documents
 - ✘ To be determined
- Participants
 - ✘ Toratani Yasumasa Canon Inc.
 - ✘ Osamu Mihara FUJI XEROX Printing Systems Co. Ltd.
 - ✘ Ide Kentaro SEIKO EPSON CORPORATION
 - ✘ Nomura Kazuo EPSON SOFTWARE DEVELOPMENT LABORATORY, INC.
 - ✘ Kanjo Hidenori BBR INC.
 - ✘ Yoshida Mikio BBR INC.
 - ✘ Shinpei Kitayama EPSON KOWA CORPORATION
 - ✘ Yamagishi Toshihiro Turbolinux, Inc.
 - ✘ Hisao Nakamura E&D
 - ✘ Koji Otani AXE





Free
Standards
Group

Print Channel Manager





Print Channel Manager & Objectives

■ Print Channel Manager API is:

- ✗ A portal independent I/O interface for sending/receiving data and control from printer include MFP

■ Print Channel Manager API contains:

- ✗

■ Objectives

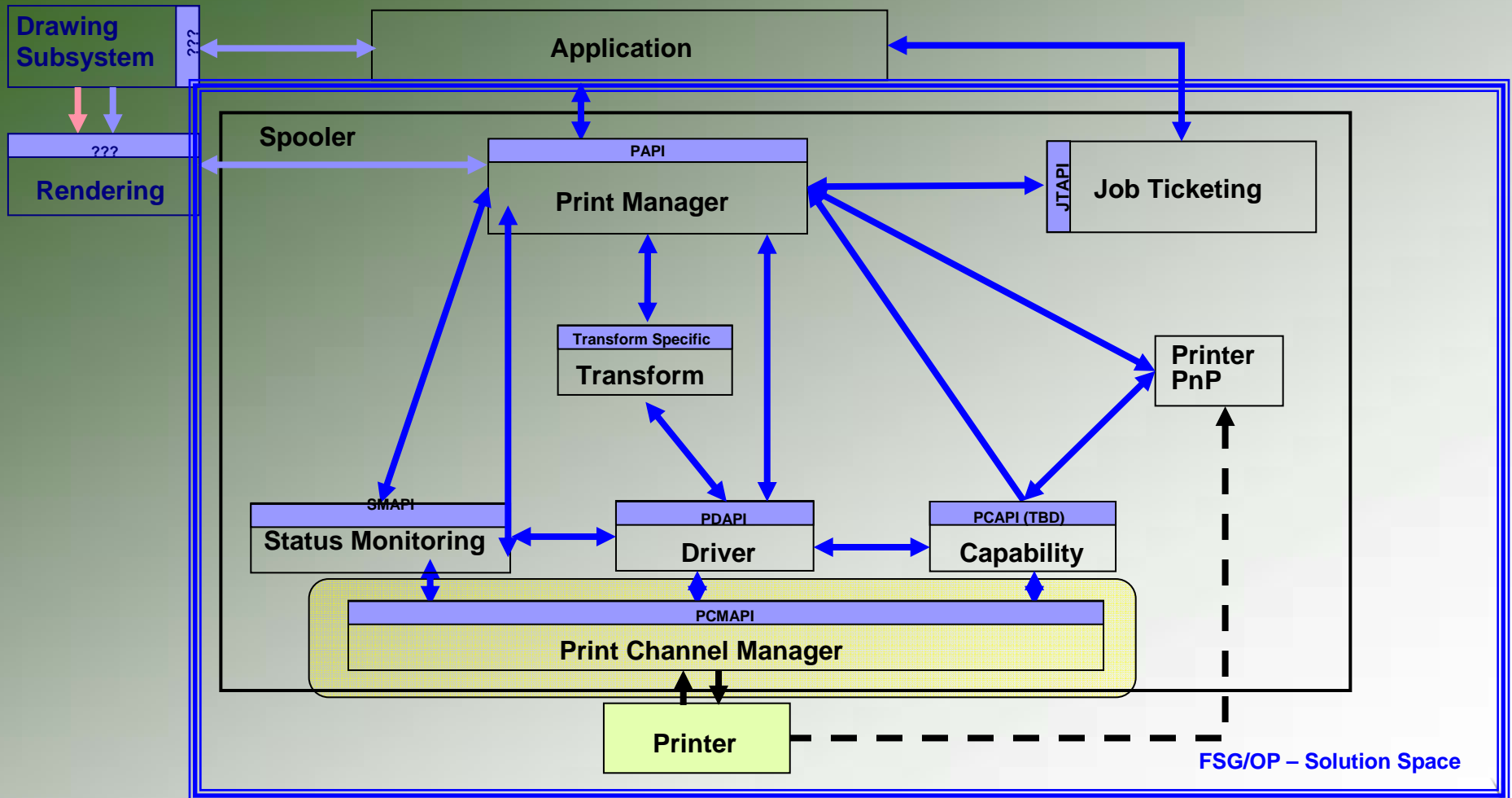
- ✗ To isolate software elements from portal physical differences
- ✗ To isolate software elements from portal protocol differences





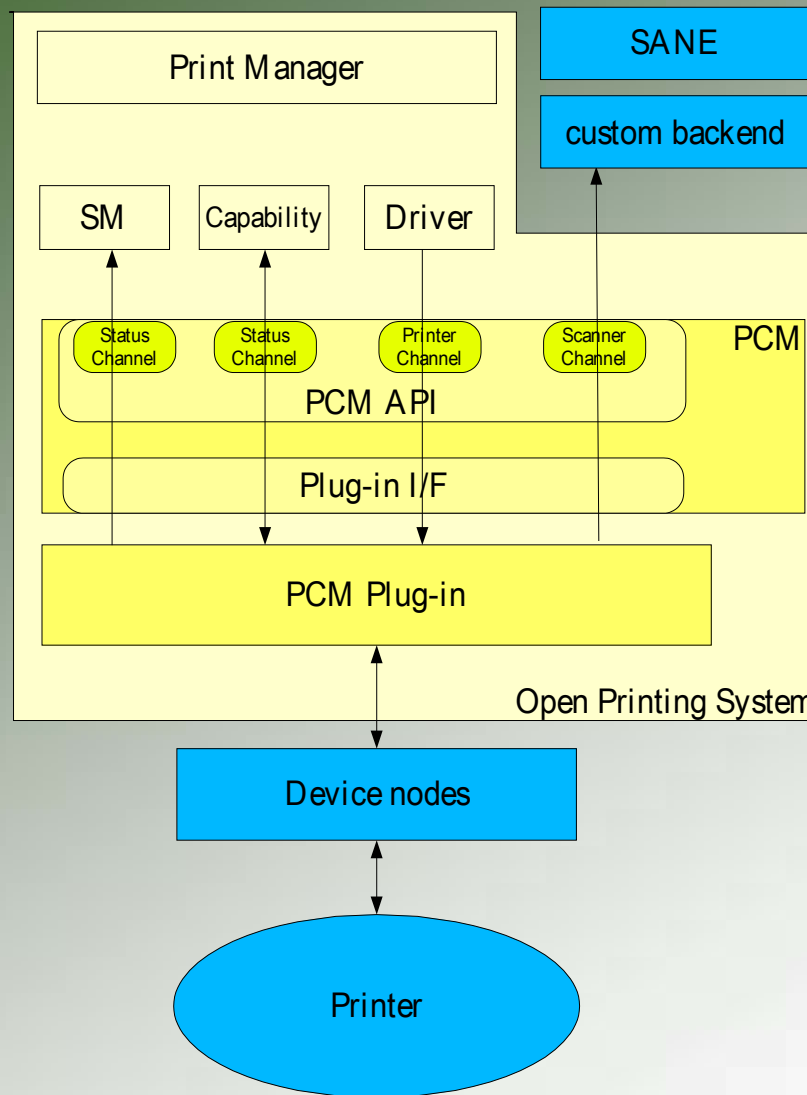
OpenPrinting Reference Model

Preliminary Reference Model





PCM outline





Print Channel Manager API

■ Channel

✘ Open/Close

- ◆ Open or close data channel to device
- ◆ Upper layer like a SM can open channel independently

✘ Read/Write

- ◆ Read and write data between upper layer and device
- ◆ Each channel have read or write function

✘ Control

- ◆ Any other data stream or control command send by control





Print Channel Manager API – Details (1)

■ Open/Close

- ✘ `int fsgpcmOpen (const char *pURI, FSGPCMctx* hFSGPCMctx);`
 - `pURI`: URI for specify device, Plug-in or device node, etc...
 - `hFSGPCMctx`: Handle to channel

- ✘ `int fsgpcmClose (FSGPCMctx hFSGPCMctx);`
 - `hFSGPCMctx`: Handle to channel





Print Channel Manager API – Details (2)

Read/Write

- ✘ `int fsgpcmRead (FSGPCMctx hFSGPCMctx, void *pBuf, size_t nBufSize, size_t nCompleteSize);`
 - `hFSGPCMctx:` Handle to channel
 - `pBuf:` pointer to data read buffer
 - `nBufSize:` Buffer size for reading
 - `nCompleteSize:` Size of read data

- ✘ `int fsgpcmWrite (FSGPCMctx hFSGPCMctx, void *pBuf, size_t nBufSize, size_t nCompleteSize);`
 - `hFSGPCMctx:` Handle to channel
 - `pBuf:` pointer to data read buffer
 - `nBufSize:` Buffer size for writing
 - `nCompleteSize:` Size of wrote data



Print Channel Manager API – Details (3)

Control

- ✧ FSGPCMResult fsgpcmControl (FSGPCMctx hFSGPCMctx, int iFSGPCMCommand, void *FSGPCMArg, void *FSGPCMResult);
 - ◆ hFSGPCMctx: Handle to channel
 - ◆ iFSGPCMCommand: Command ID
 - ◆ FSGPCMArg: Pointer to control description to device
 - ◆ FSGPCMResult: Pointer to return data from device



- Draft – working now!
 - 2004/5/31 Pre-draft of the PCM specifications.
(written only the main point.)
 - 6/ F Open to working group.
 - 6/ F Start discussion and writing the draft.
 - 2005/6/E Open public draft in Japanese
 - 7/F Open public draft in English
- Discussion about PCM
 - ✗ Printing-Japan ML
 - ✗ Face to Face meeting – Monthly
- Milestone
 - ✗ 2004/8/27 Create a model/architecture
 - ✗ 2005/1/30 Release 1.0 specification
 - ◆ Change timeline according to draft's schedule and change use case
- To-Do
 - ✗ Consider the other existing Printing Systems so PCM could be easily adopted
 - ✗ Consider the future extension
 - ◆ Support new I/F or new device like MFP
 - Today's public draft make consideration MFP





PCM Working Group Information

- To subscribe to FSG PCM mailing list:
 - ✗ <http://freestandards.org/mailman/listinfo/printing-japan>
- To post a message to FSG PCM mailing list
 - ✗ printing-japan@freestandards.org
- To view FSG PCM mailing list archives
 - ✗ <http://freestandards.org/mailman/listinfo/printing-japan>
- To find FSG PCM documents
 - ✗ To be determined
 - ◆ Available at <http://www.openprinting.org/moin.cgi/OpenPrinting/PrintChannelManager> soon!
- Participants
 - ✗ Kentaro Ide (Seiko Epson Corp)
 - ✗ Shinpei Kitayama (Epson Kowa Corp.)
 - ✗ Fumio Nagasaka (Seiko Epson Corp.)
 - ✗ Kazuo Nomura (Seiko Epson Corp.)
 - ✗ Koji Otani (AXE Inc)
 - ✗ Osamu Mihara (FUJI XEROX Printing Systems Co. Ltd.)
 - ✗ Yasumasa Toratani (Canon inc,)





Thanks
from the
FSG / Open Printing
Members

