

JDF - Technical Overview



Technical Overview of the Job Definition Format (JDF)

www.cip4.org

Dr. Markus Möller

mmoe@mro.man.de

MAN Roland Druckmaschinen AG

JDF Tech. Overview

Table of Contents 1



- Introduction
 - About JDF, Scope of JDF
 - CIP4 and JDF
 - structure of organization, members, working groups
- Goals of JDF
- Simple JDF Workflow
- JDF Properties
 - Relation to other standards

Table of Contents 2



- Architecture of JDF
 - Core: Nodes, Resources and Resource Links
 - Structure of a JDF-Node (High Level Elements)
 - Node Hierarchy, Tree-Structures of Nodes
 - JDF Execution Model
 - How these concepts play together.
 - Spawning and Merging of Nodes
 - Workflow Components
 - Structure of Resources, Resource Links, CustomerInfo, NodeInfo, Audits
 - Pipes
 - Job Messaging Format (JMF)
 - Dynamic Pipes

About JDF

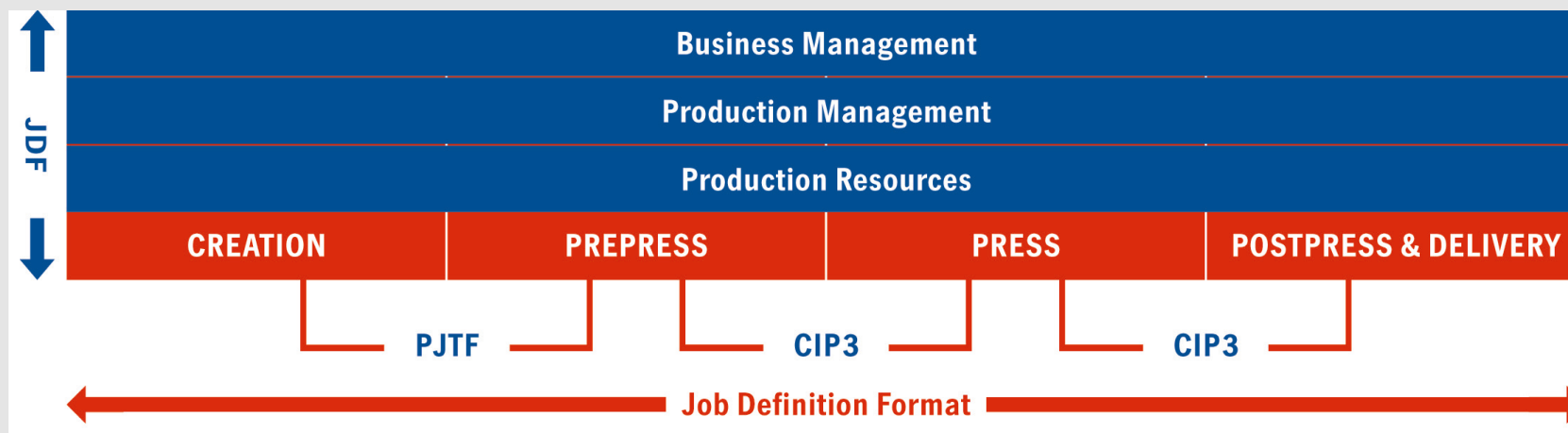


- A new, open standard for integration of all computer aided business and production processes around print media.
- Basis for future solutions.
- Common announcement was made on Seybold Conference 2000 by Adobe, Agfa, Heidelberg and MAN Roland.
- Release of Version 1.0 in April 2001
- Specification and further information available at:
<http://www.cip4.org>

Scope Of JDF



- Horizontal Job Description
 - Job Ticket
- Vertical Communication
 - Messaging
- Why JDF ?
 - Solutions for a Computer Integrated Manufacturing (CIM) in the Graphic Arts Industry need a comprehensive, vendor-independent standard.
 - Other formats do not cover all process steps.



CIP4 and JDF

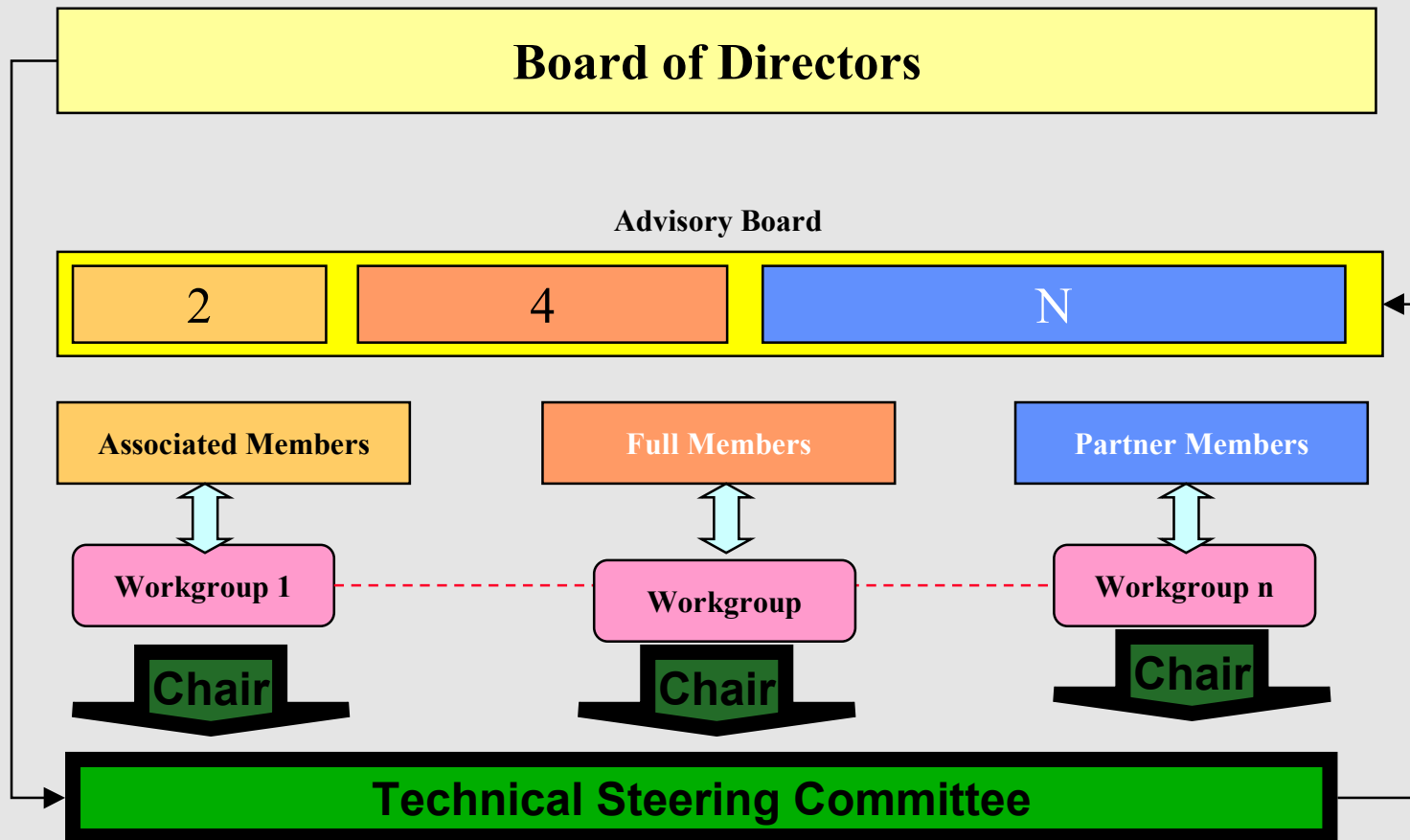


- CIP4 is the Organization that owns JDF
CIP4: **C**ooperation for the **I**ntegration of **P**rocesses in **P**repress, **P**ress and **P**ostpress, more: <http://www.cip4.org>
- Three classes of membership in CIP4
 - Partner
 - Full
 - Associate
- General Conditions
 - voting rights according to membership class
 - participation in working groups for all members
 - payment of class annual dues according to membership class
 - use of the CIP4 member logo
 - commercial use of open source for full + partner members

CIP4 and JDF



- CIP4: Structural Organization



CIP4 and JDF



- CIP4 currently consists of 78 members

- 13 Partner Members

- Adobe
- CreoScitex
- Heidelberger Druckmaschinen AG
- Koenig & Bauer AG
- Müller Martini
- Optimus
- Xerox
- Agfa
- EFI
- Hewlett-Packard
- MAN Roland
- NexPress Solutions LLC
- PrintCafé

- 42 Full Members

- 23 Associate Members

CIP4 and JDF



- CIP4 Technical Working Groups:
 - Working Group Agenda:
 - Develop addenda to the specification
 - Discuss and resolve technical issues
 - Working Group Structure
 - Chair + Alternate Chair
 - Member of the Technical Steering Committee
 - 1 Vote / Working Group
 - CIP4 members of all membership levels may actively participate in any technical working group

CIP4 and JDF

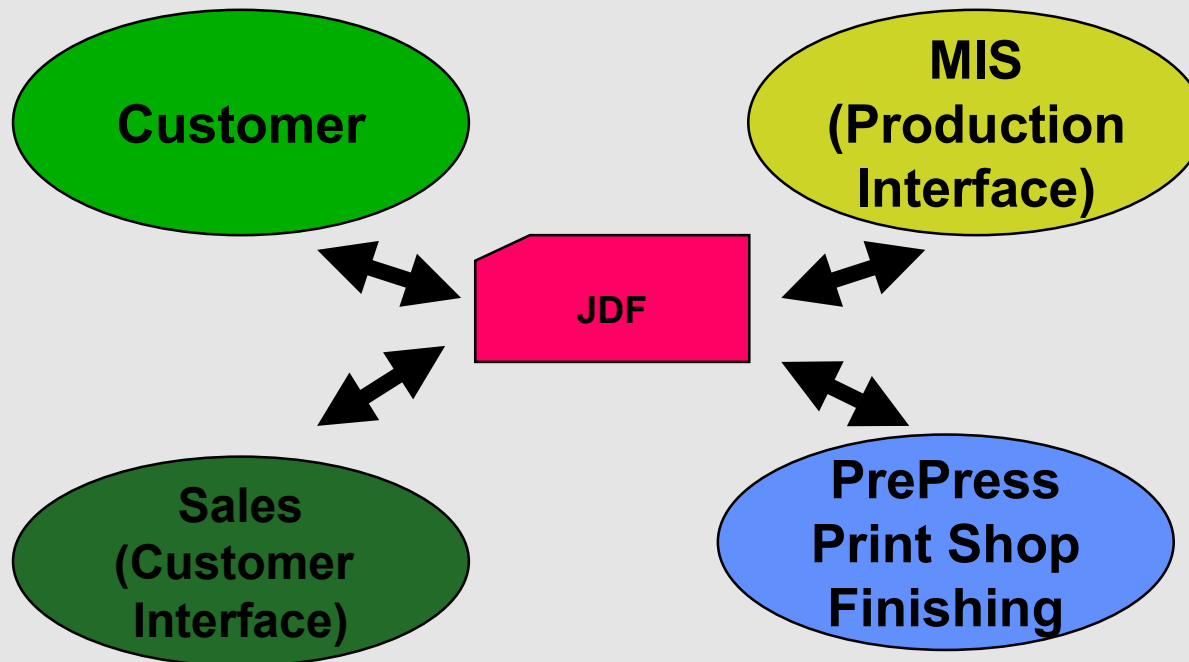


- CIP4 Technical Working Groups:
 - Advertising / Magazine Publishing (1)
 - Color Workflow (12)
 - Device Capability Description (22)
 - Device messaging / Job tracking (27)
 - eCommerce (25)
 - Finishing (9)
 - Gravure (4)
 - Newspaper (4)
 - Packaging & Label (4)
 - Process Ressources and Definitions (7)
 - Tools + Infrastructure (29)
 - Variable Data (13)
 - Use Cases / Compliance (23)
 - Web / Rotary Printing (4)

High Level Goal of JDF



- Represent and interchange Information about a Print Job between the workflow participants
- Create a digital “Job-Bag”
- Describe a Print Job in all Stages of its existence



Design Goals of JDF



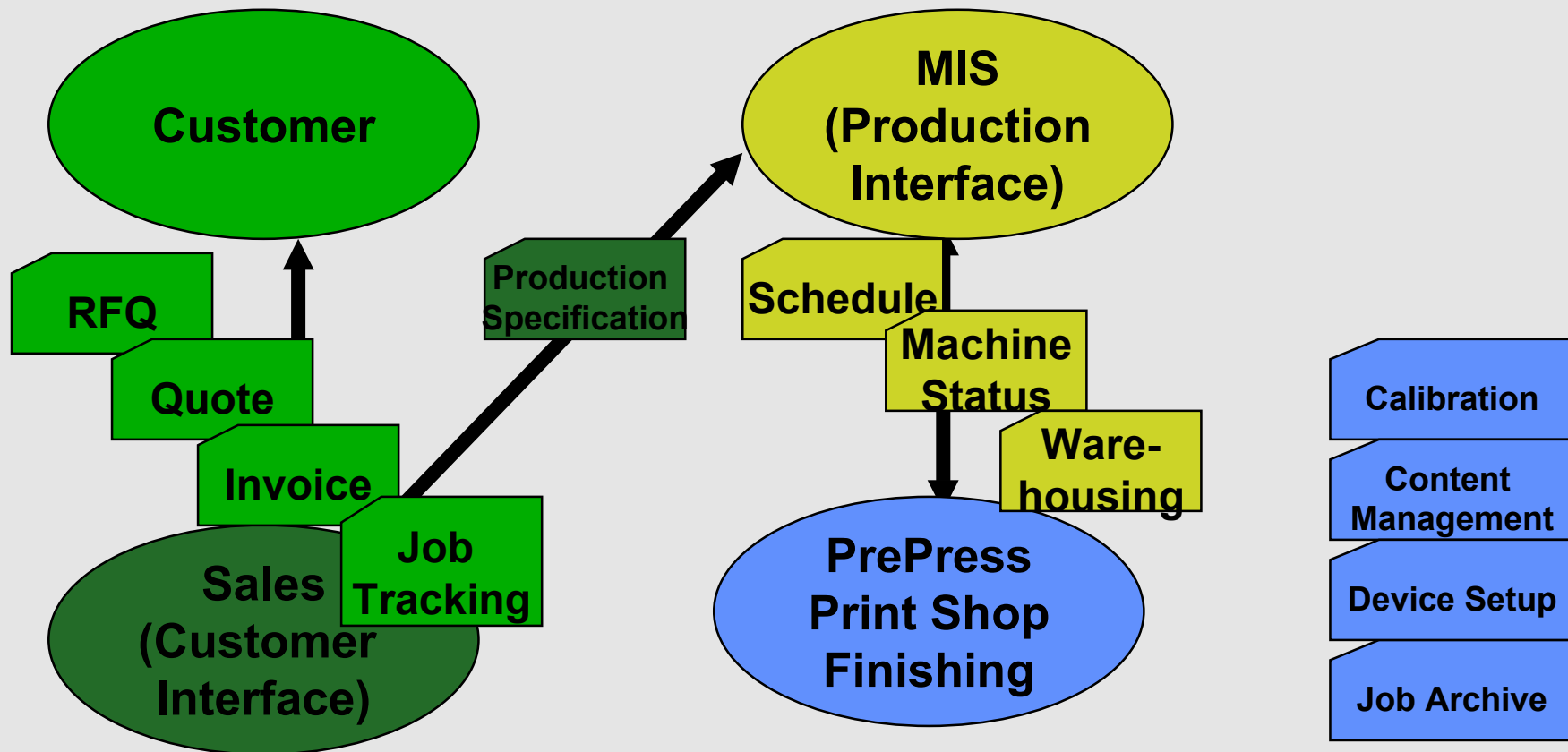
- Flexible mapping of myriads of existing workflows
- Allow both detailed production level representation and customer level representation of print jobs.
- Allow parallel work
- Use widely available technologies
- Leverage from existing standards, where possible
- Low redundancy of the specification
- Simple implementation (use of XML)
- Job Centric view of the world
- Object-Oriented design

Simple JDF Workflow



- Customer creates a JDF representation of the desired product and delivery.
- Customer and Print Sales negotiate contract; Print sales uses MIS to do this.
- Printer defines production workflow using MIS.
- Job gets executed. Quasi-Real-time information is transferred to MIS from the JDF enabled devices.
- Collected data is evaluated for calculation.
- Job is archived for potential rerun.

Simple JDF Workflow



JDF Properties



- **JDF is a Data Interchange Format Specification - not an Application or System.**
- Encoded in XML
- Extensible
- Based on Semantic structures defined by:
 - Adobe PJTF
 - CIP3 PPF
- Process Modeling with Product- and Process-Nodes and Resources.
- JDF Job Definition + JMF Messaging define the JDF Framework

Relations to other Standards



- Adobe PJTF: 1--1 mapping to JDF resources
- CIP3 PPF: simple mapping to JDF resources
- IFRATrack: JDF contains IFRATrack's functionalities
- PrintTalk: eCommerce (Request for Quote, Quote, Invoice, Change Order); wrapper ar. JDF

- JDF is independent of the Content Description Languages like PDF, PS, PPML (from PODi)...
- JDF contains Production Data

JDF Encoding

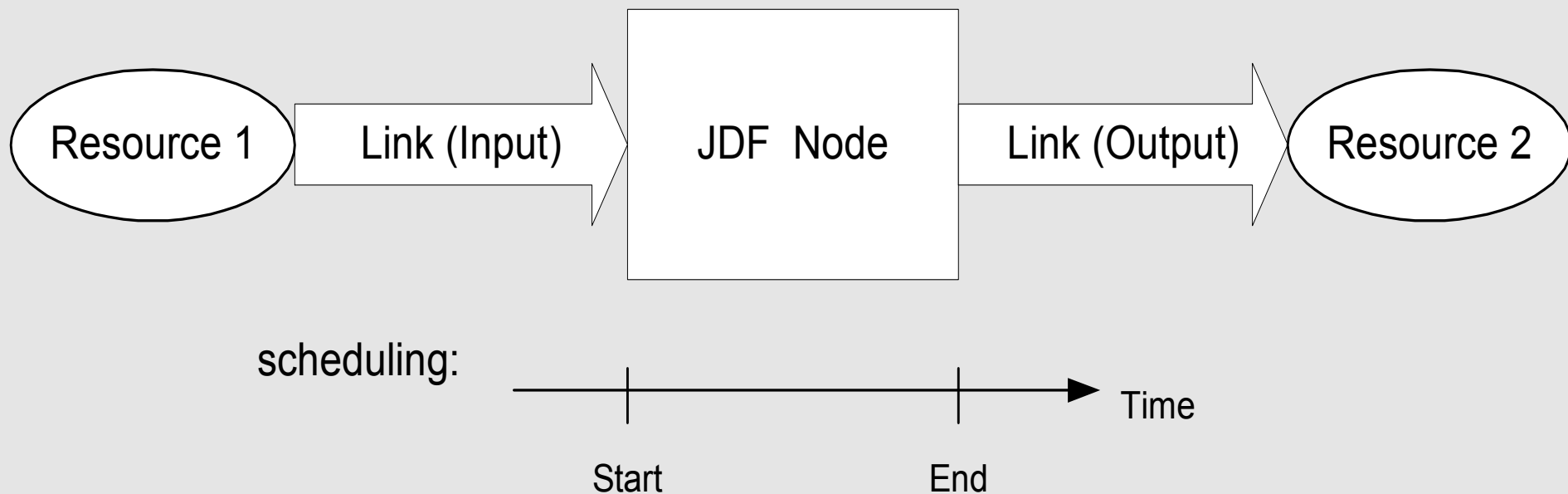


- XML Encoding
- External references via URI/URL
- Use of ID-IDREF pairs
- XML Schema for Data Type Definitions (Work in Progress)
- Extensibility using XML name spaces
- Optionally in a MIME/Multipart Wrapper
 - Allows single file with binary data
- Image Preview Data as Multiple PNG (Portable Network Graphics) Grayscale Separations

Core: Node and Resource Interaction



- JDF Node
 - Specifies a Product, Process or group of Processes
- Resources
 - represent Parameters or Physical Resources
- Resource Links
 - Bind a Resources to a Node: determine consumption, usage and production of resources



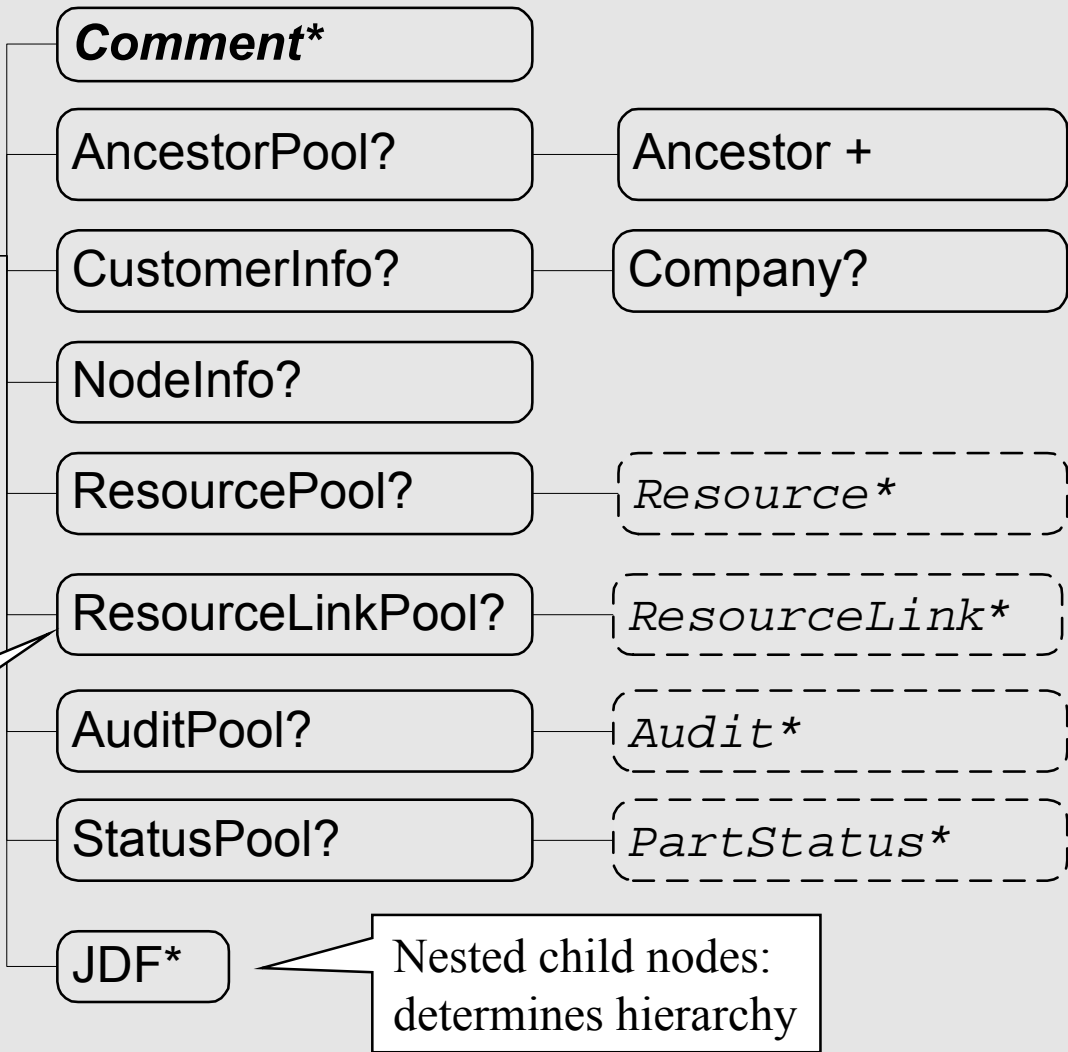
High Level Elements (1)



Type: determines node type
(Product, ProcessGroup,
Combined, any physical process)

- JDF**
- ID
 - Type
 - Status
 - Activation?
 - JobID?
 - JobPartID?
 - Types?
 - *Version?*
 - *CommentURL?*
 - *DescriptiveName?*

Content determines
resources used, consumed
or produced by this node.



JDF High Level Elements (2)



- JDF Node
 - Specifies a Product, Process or group of Processes
 - Modifies, consumes creates resources
 - May contain further nested JDF Nodes
- Resources represent:
 - Parameters or Logical Entities
 - Physical Entities: Quantity (Component), Handling Resource (ExposedMedia) , or Consumable
 - Implementation (Device or Employee)
 - Intent, used for Product Nodes
 - Selector and Placeholder, used for workflow definition

JDF High Level Elements (3)



- Resource Links
 - Bind Resources to a Node
 - determine the consumption, usage and production of resources
- StatusPool
 - Logs Status of Parts of Partitioned Resources

JDF High Level Elements (4)



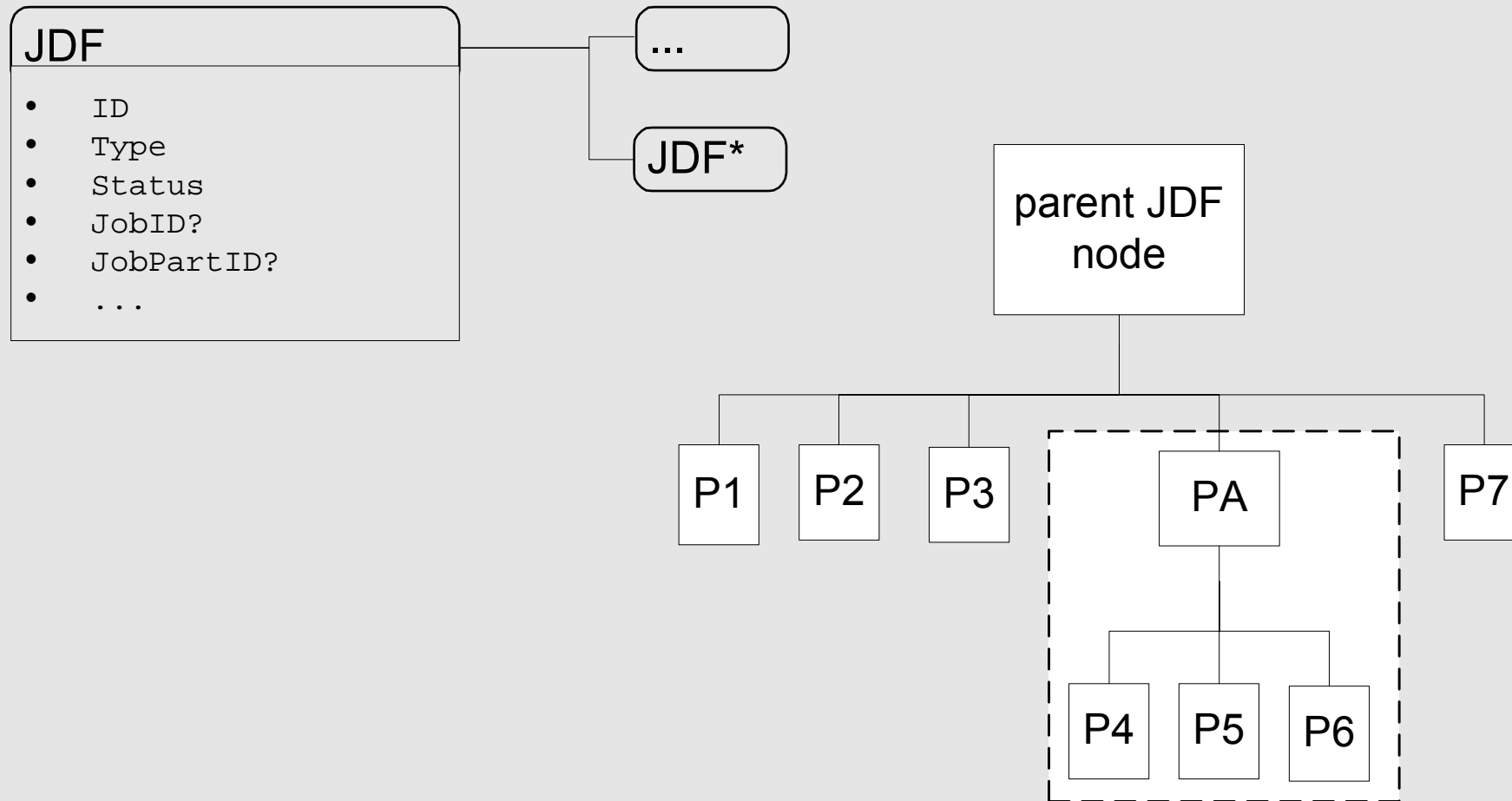
- NodeInfo
 - Contains scheduled, planned job properties
- AuditPool
 - Logbook: Logs actual events, job states, and post-facto job properties.
- CustomerInfo
 - Customer + Delivery Address
- Comment
 - Internationalization
 - Path / Box annotation

JDF Node -- Simple Example



```
<JDF ID="n20000824112251" Type="Product" JobID="some product ID"
  Status="Waiting" Version="0.9">
  <NodeInfo/>
  <CustomerInfo/>
  <ResourcePool>
    <SomeInputResource ID="Link0002" Class="Parameter" Status="Available"/>
    <Component ID="Link0003" Class="Quantity" Status="Unavailable"
      DescriptiveName="Some output resource"/>
  </ResourcePool>
  <ResourceLinkPool>
    <SomeInputResourceLink rRef="Link0002" Usage="Input"/>
    <ComponentLink rRef="Link0003" Usage="Output"/>
  </ResourceLinkPool>
  <AuditPool/>
</JDF>
```

Hierarchy of Nodes (1)



Hierarchy of Nodes (2)



- One JDF node type for Products and Processes
 - Allows Spawning and Merging of JDF for subcontracting, parallelizing
- Less Precise Product Intent at the Root
- Workflow Groups in between
- Detailed Processes in the Leaves
- Job + Job Part Identification

JDF Execution Model (1)



- Product Definition
 - No Process
 - Abstract
 - Segmentation by Product Components
- Serial Processing
- Parallel Processing
- Overlapping Processing
 - Pipes
- Iterative Processing
 - Informal Iterative Processing using Draft Resources
 - Formal Iterative Processing additionally using JMF Messages

JDF Execution Model (2)

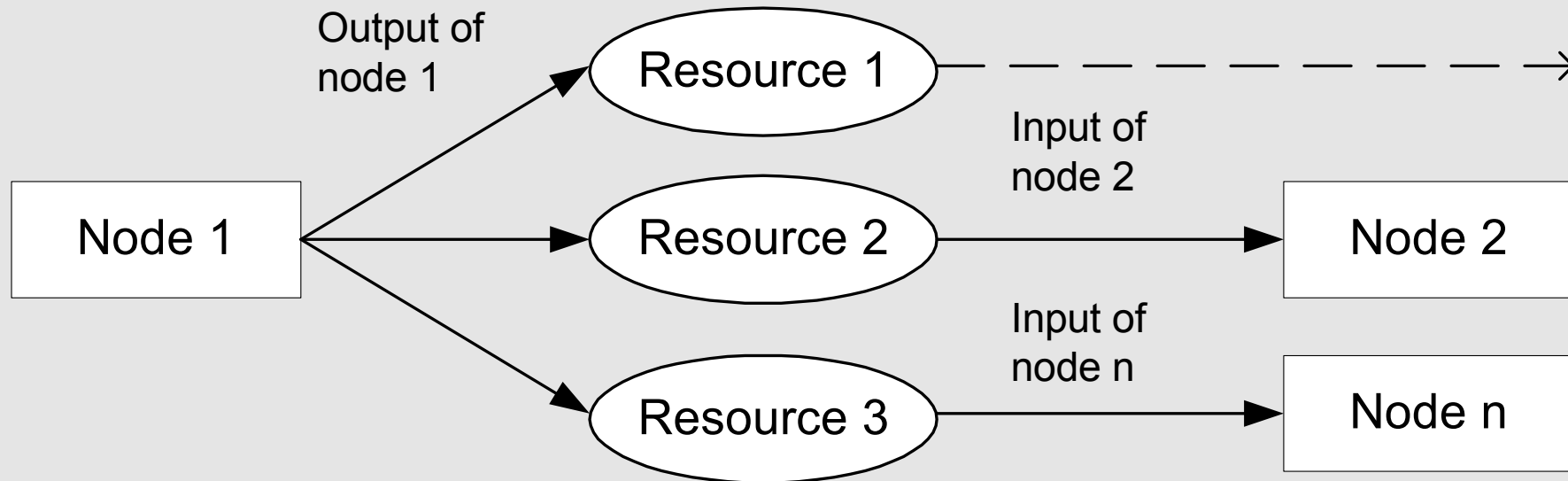
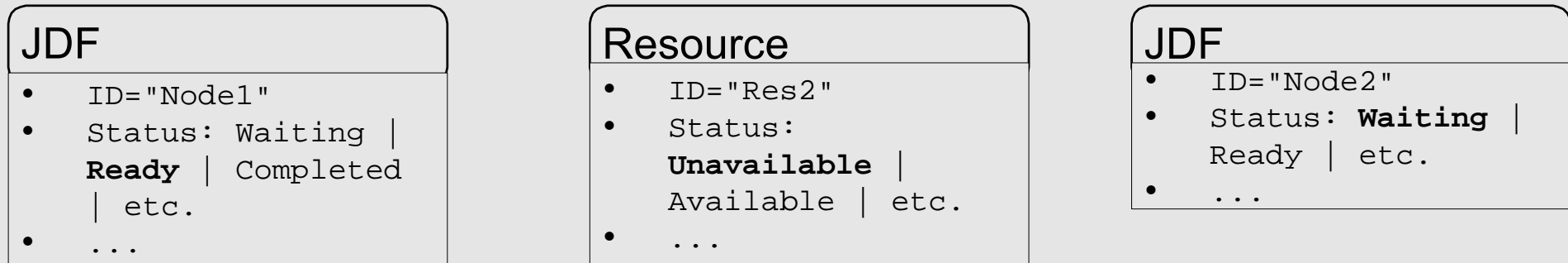


- A Node is executable when all required input resources are available.
- Define arbitrary workflow sequencing.
- Link one resource to multiple nodes.
- Node dependencies allow Process configuration.
 - A proof node can create an ApprovalSuccess Resource which is a required input resource for a printing node.

JDF Execution Model (3)



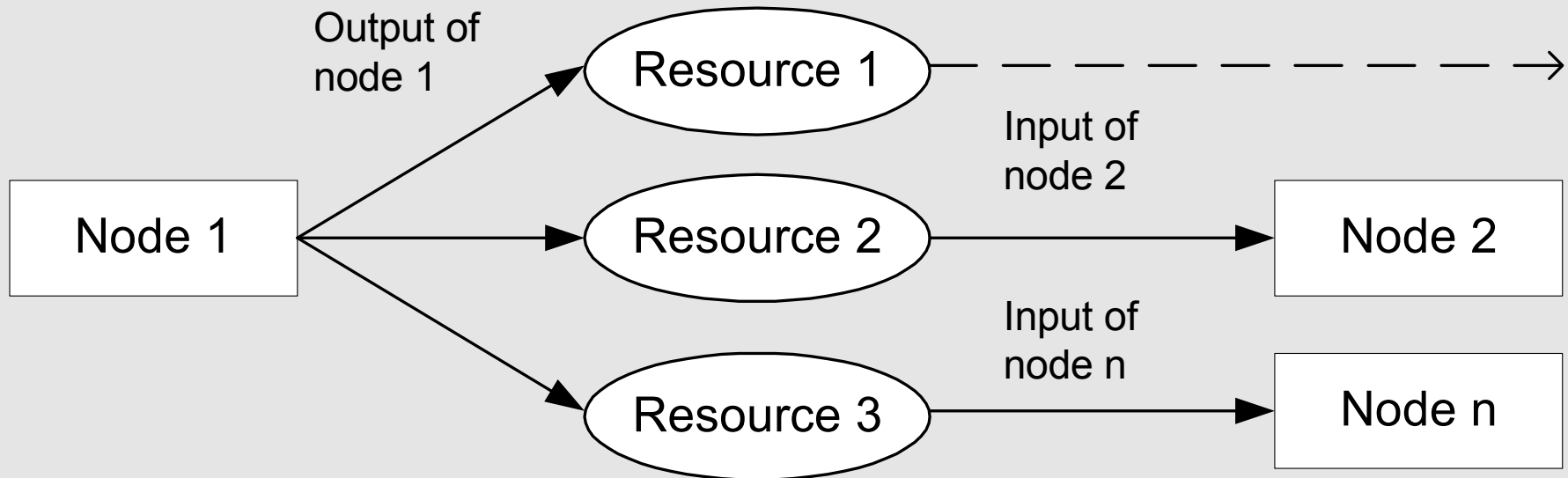
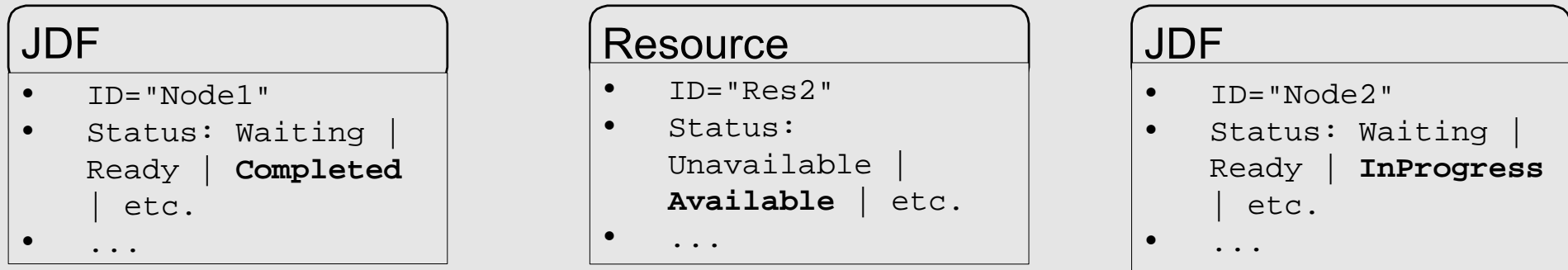
- Linking of nodes by resource links and resources.



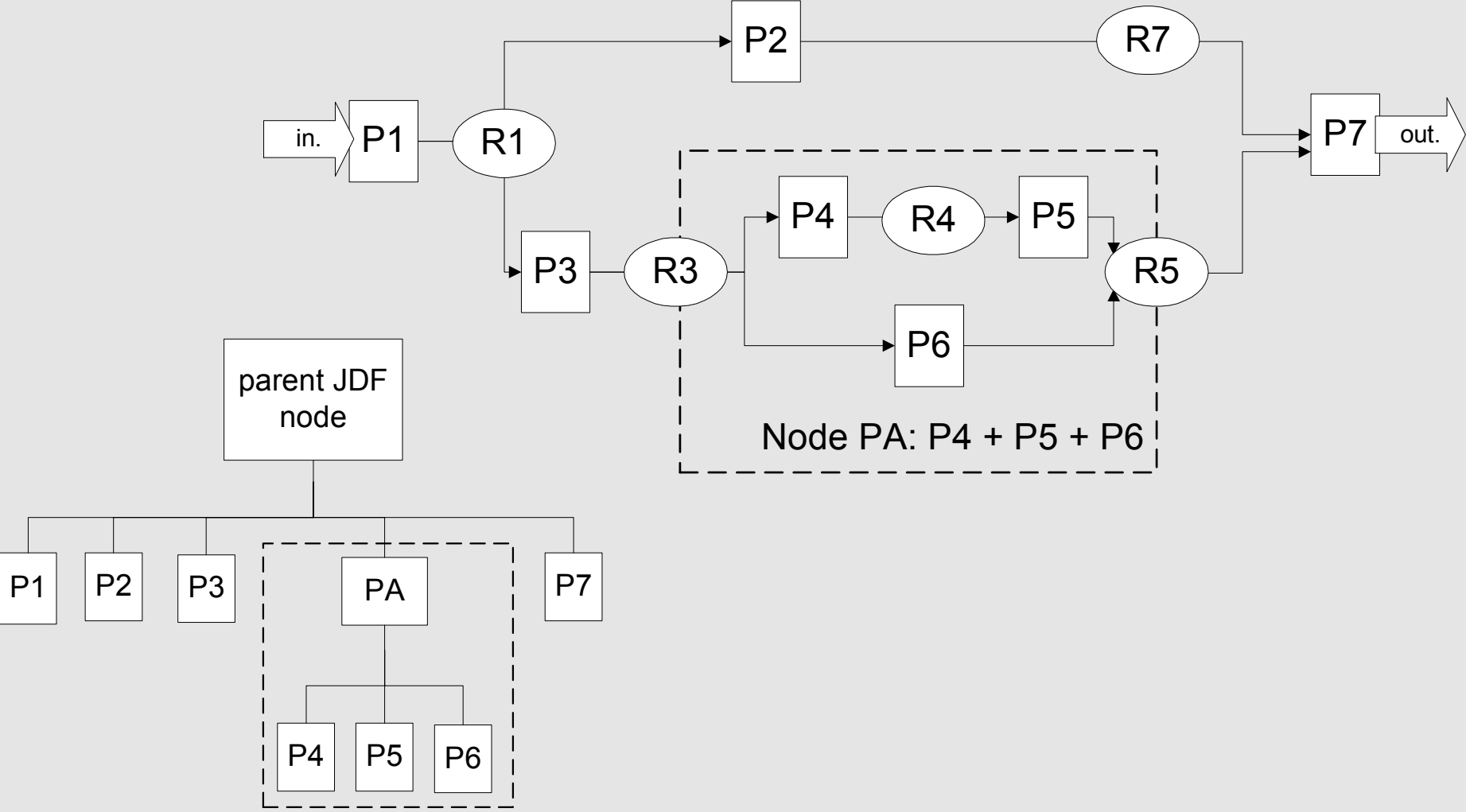
JDF Execution Model (4)



- Linking of nodes by resource links and resources.



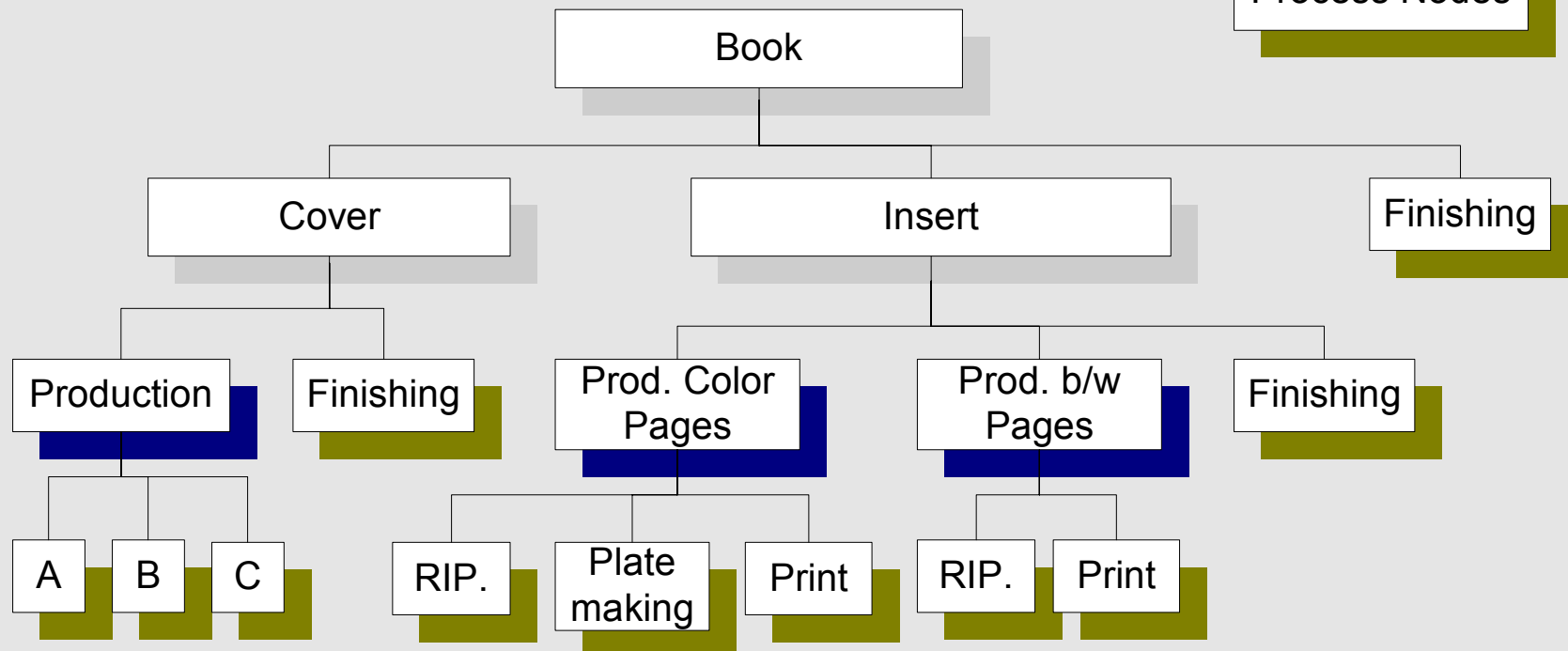
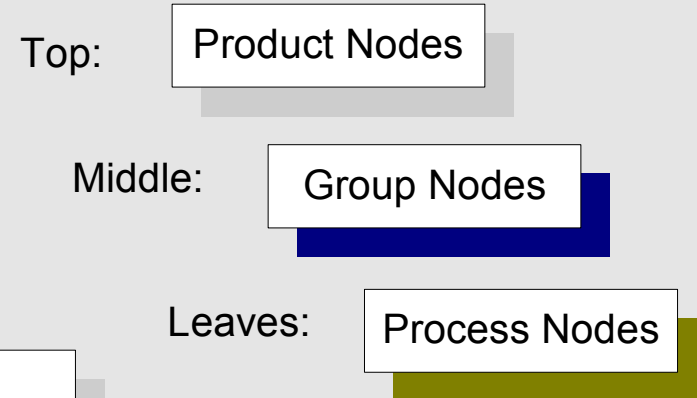
JDF Tree / Network Structure



JDF Job Example



Top Down Hierarchy of Nodes



JDF Tech. Overview

Combination of JDF Nodes



- Define a limited number of “atomic” Processes.
- Combine multiple predefined processes into one process, e.g.:
 - inline finishing = printing + folding + cutting;
 - in-RIP trapping = trapping + RIPping
- Two types of Combination Nodes
 - **Combined Node**: All internal interfaces are hidden
 - Smart multi-function device
 - **ProcessGroup**: Internal nodes are accessible
 - Workflow group in a department
 - Subcontract

Sub-Table of Contents



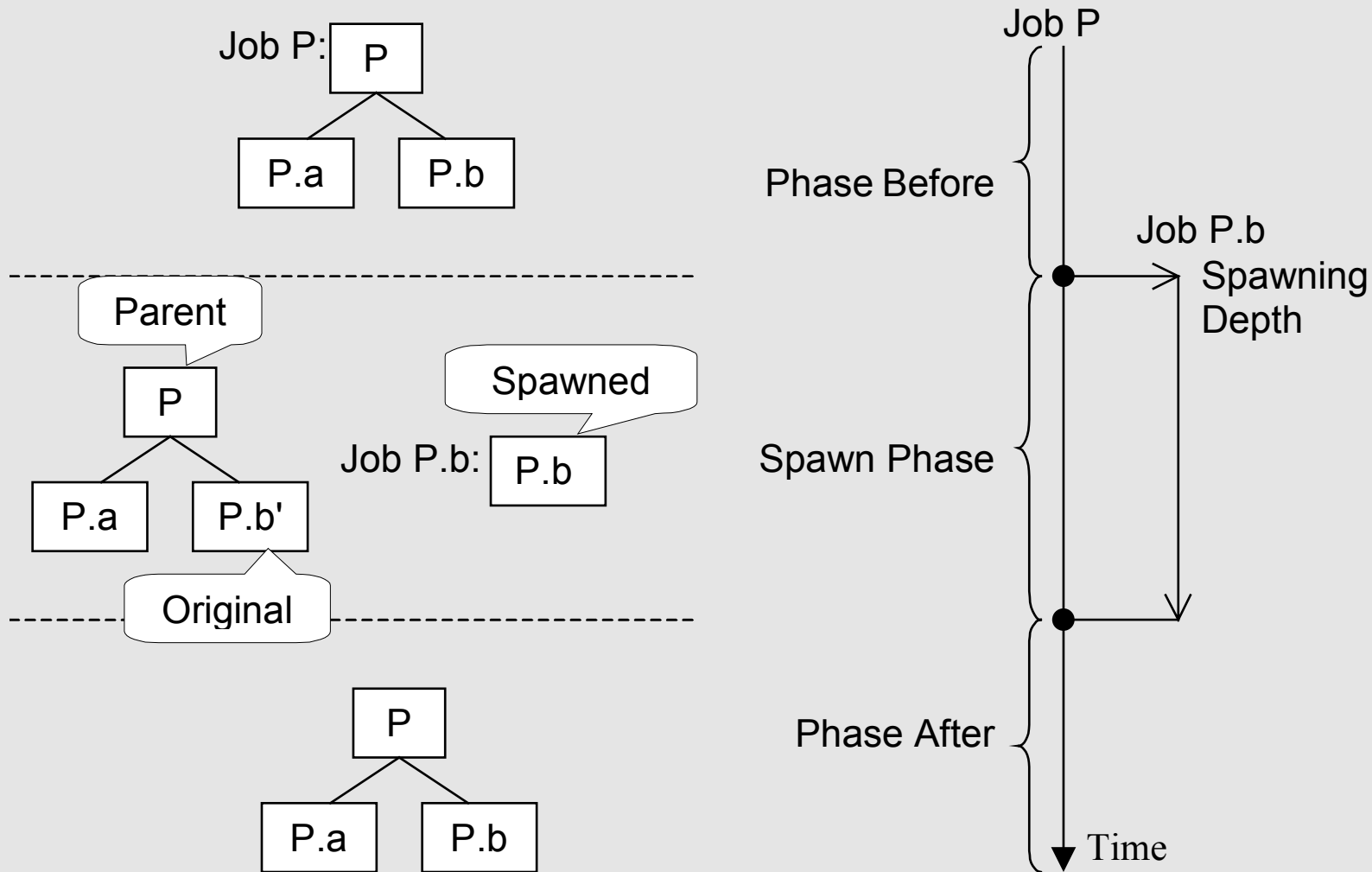
- JDF - Spawning and Merging
 - Basic Mechanism
 - Recursive Spawning & Merging
 - Merging of Independent Jobs

JDF - Spawning and Merging



- Spawn parts of the JDF Tree for independent Processing
- Merge back after Processing
 - Basic Mechanism
 - Recursive Spawning and Merging
 - Independent Spawning and Merging

Basic Spawning and Merging Mechanism



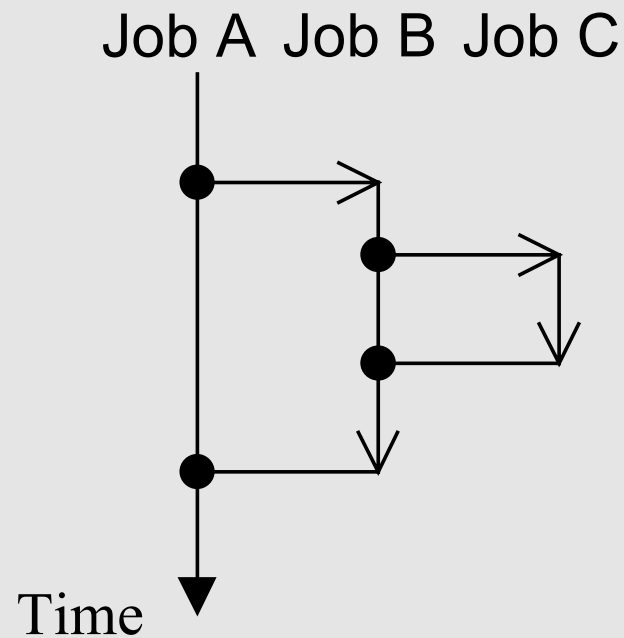
Recursive Spawning & Merging



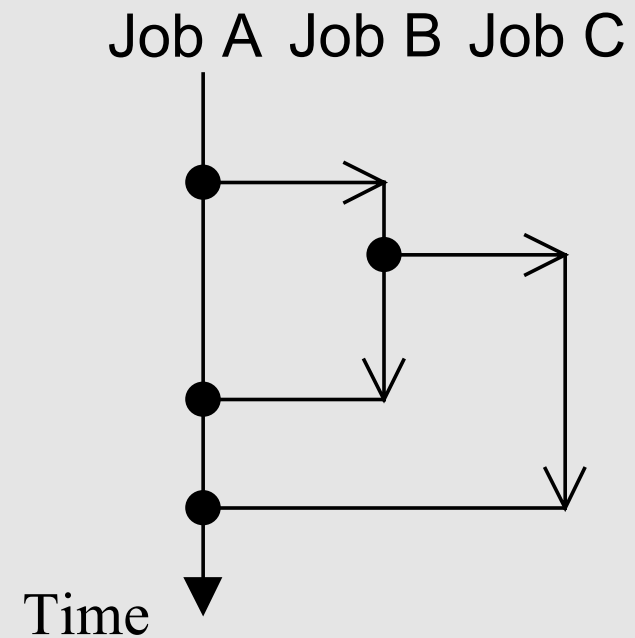
Spawning depth

Spawning

correctly nested



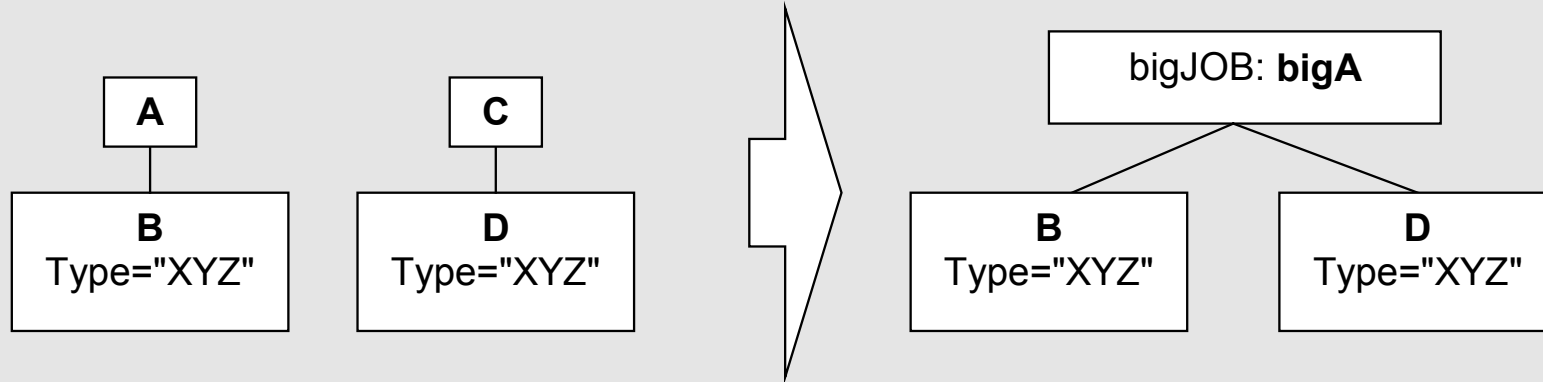
reversely nested



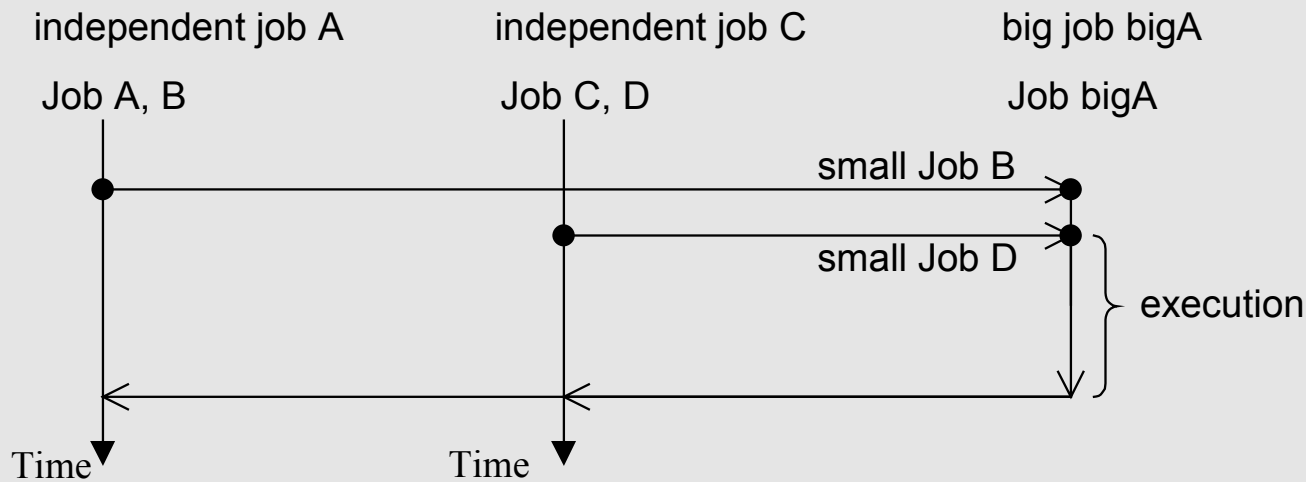
Merging of Independent Jobs



e.g. for Combined Processing in one Press Run



Submission Diagram



Sub-Table of Contents



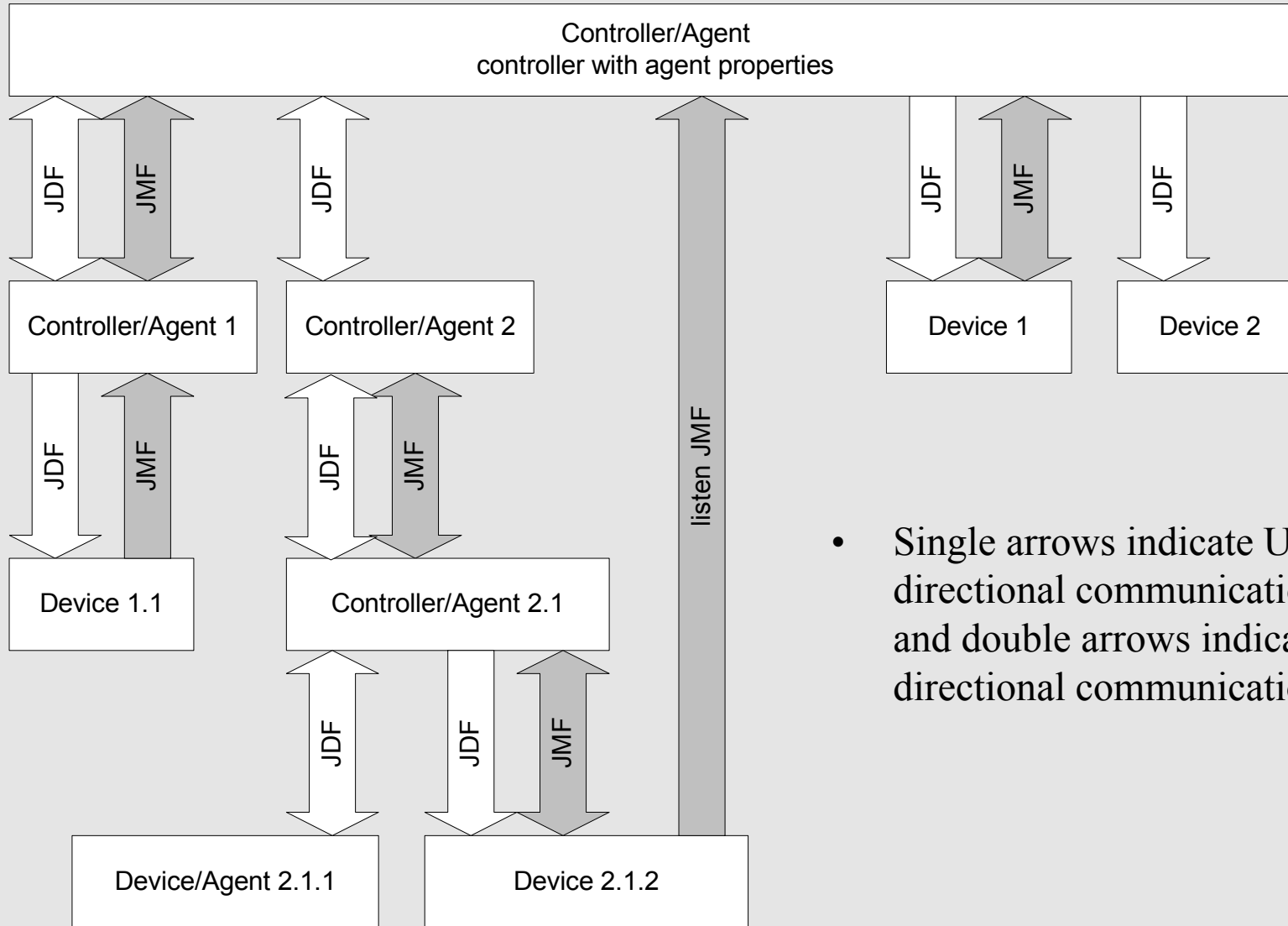
- Workflow Components
 - Some Definitions: Machines, Devices, Agents, and Controllers
 - Example of Component Interaction

Definitions of Workflow Components



- JDF does not dictate how a JDF/JMF system should be designed, built, or implemented.
- **Machine:** has no JDF interface.
- **Device:**
 - reads JDF.
 - executes JDF nodes.
 - may support JMF.
- **Agent:**
 - writes JDF.
 - creates jobs and JDF nodes, modifies JDFs, composes JDFs.
 - may support JMF.
- **Controller:**
 - routes JDFs to devices/agents.
 - initiates processes on at least one device.
 - Communicates via JDF file-exchange protocol, may support JMF.

System Interaction Example



- Single arrows indicate Uni-directional communication channels and double arrows indicate bi-directional communication.

Sub-Table of Contents



- Inside JDF
 - JDF Resources, Resource Classes, Partitioned Resources
 - ResourceLink, Pipes
 - CustomerInfo and NodeInfo
 - Audits

JDF Resources



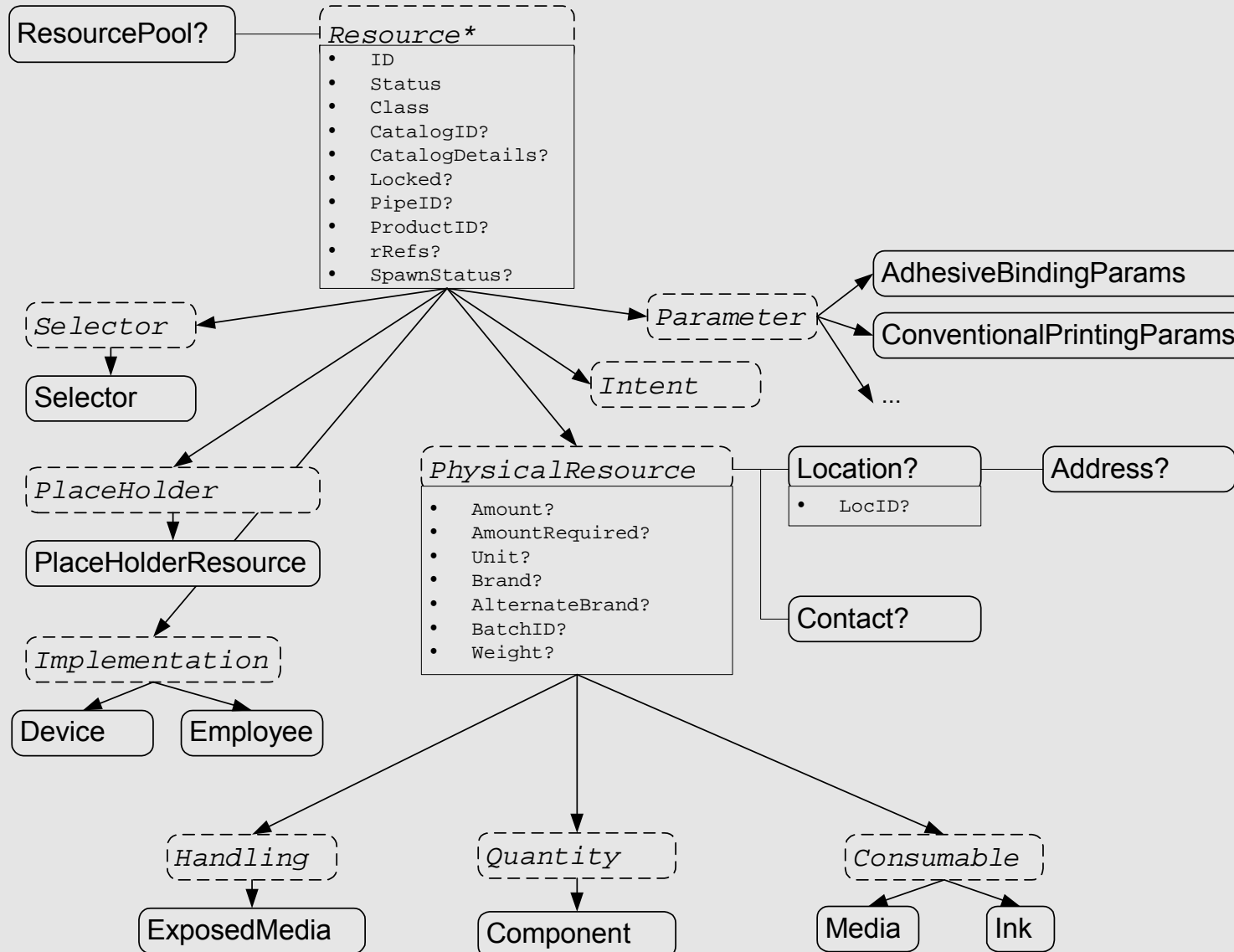
- Based of PJTF and CIP3 PPF
 - Sheet definition, Colors : PJTF
 - Press, Finishing: CIP3
 - Extensions where appropriate.
- Internal to JDF
 - Parameters
 - URL definitions
- Internal to JDF or External Links to well-defined Formats
 - Thumbnails,
 - ICC Profiles
 - Content Data

Resource Classes (I)



- Base class resource is abstract
- Parameter, like:
 - Process parameters
 - Content file reference
- Physical Resources (abstract):
 - Consumable, like paper, ink, raw plate
 - Handling, (reusable) like ExposedMedia (developed film)
 - Quantity, like printed or processed material
- Implementation , like:
 - Devices (limited to string + family in v1.0)
 - Operators: id, shift, role
- Intent, describes the content of product nodes
- Selector
- Placeholder

Resource Classes (II)



Partitioned Resources (I)



- Inheritance of common Data
 - Overwrite defaults
- Partition keys are predefined, like:
 - SheetName
 - Separation
 - Side
 - TileID
 - etc.
- Access individual parts of a large resource
 - Only the yellow plate of the front surface of sheet 17
- Mechanisms for Parallel Processing of Partitioned Resources

Partitioned Resources (II)



Resource PartIDKeys="KeyA KeyB" DescriptivName="Root part"

Resource KeyA="keyA-id1" DescriptivName="Part 1"

Resource KeyB="keyB-identifier1" DescriptivName="Leaf 1.1"

Resource KeyB="keyB-identifier2" DescriptivName="Leaf 1.2"

Resource KeyA="keyA-id2" DescriptivName="Part 1"

Resource KeyB="keyB-identifier1" DescriptivName="Leaf 2.1"

Resource KeyB="keyB-identifier2" DescriptivName="Leaf 2.2"

Partitioned Resource (III) Examp.



```
<ExposedMedia Class="Handling" Brand="Gooley" ID="L1" Status="Available"  
  PartIDKeys="SheetName Side Separation" Amount="2">
```

```
<Media MediaType="Plate" Dimension="500 600"/>
```

```
<ExposedMedia SheetName="S1">
```

```
<ExposedMedia Side="Front">
```

```
<ExposedMedia Separation="Cyan" ProductID="S1FC" />
```

```
<ExposedMedia Separation="Black" ProductID="S1FK" Status="Unavailable" />
```

```
</ExposedMedia>
```

```
<ExposedMedia Side="Back">
```

```
<ExposedMedia Separation="Cyan" ProductID="S1BC" />
```

```
<ExposedMedia Separation="Black" ProductID="S1BK" />
```

```
</ExposedMedia>
```

```
</ExposedMedia>
```

```
<ExposedMedia SheetName="S2" Side="Front">
```

```
<ExposedMedia Separation="Cyan" ProductID="S2FC" />
```

```
<ExposedMedia Separation="Black" ProductID="S2FK" />
```

```
</ExposedMedia>
```

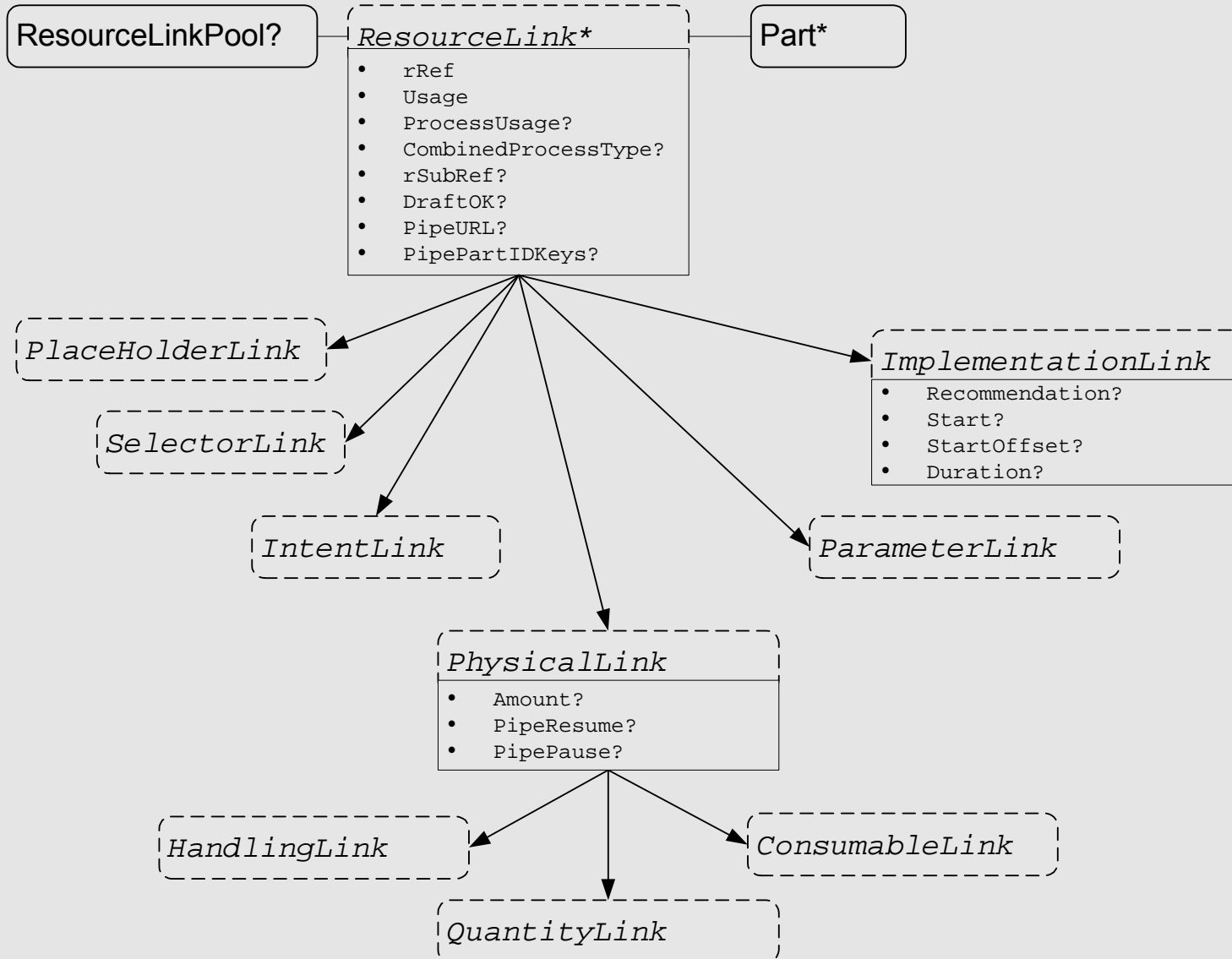
```
</ExposedMedia>
```

ResourceLink (I)



- Bind a Resource to a Node
 - A Resource is NOT bound to the JDF that contains it unless it is linked by a ResourceLink.
- Define Resource Usage (Input or Output)
- May Link to a Subset / Part of a Resource
- May Contain Pipe control meta-data.
- Live in ResourceLinkPool of a JDF node.
- Name is derived from the linked resource
- Allow reuse of Resources by multiple processes
 - One resource may be linked by multiple ResourceLinks

ResourceLink (II)



ResourceLink (III) Examples



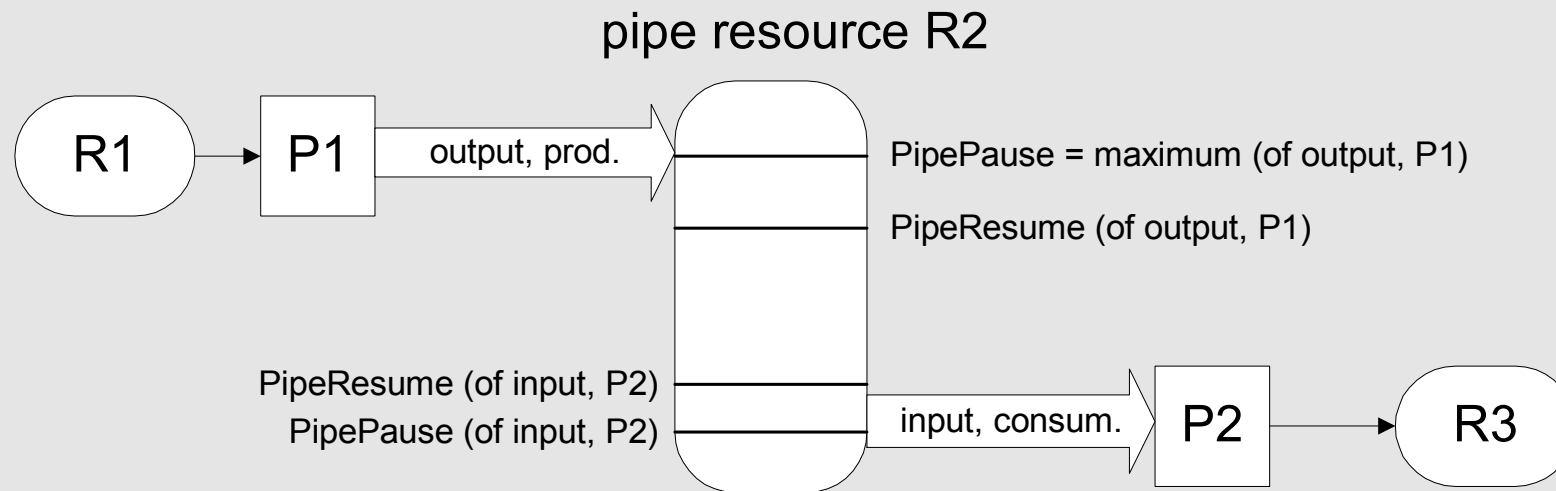
- Simple
 - **<ScanParameterLink rRef="ScanLinkID" Usage="Input"/>**
- Quantity
 - **<MediaLink rRef="MediaLinkID" Amount="6642" Usage="Input"/>**
- Part (Links to previous example)
 - **<ExposedMediaLink rRef="L1" Usage="Input">**
<Part SheetName="S2" Side="Front" Separation="Cyan"/>
</ ExposedMediaLink >

Pipes (Transient Resources)



- Overlapping Processing
 - Print 10 pallets and start folding when one is ready;
- Undefined Amounts
 - Request new plates in long press runs
- Data streams
- Buffer Handling
- Synchronization Messages

Pipe Resource Linking

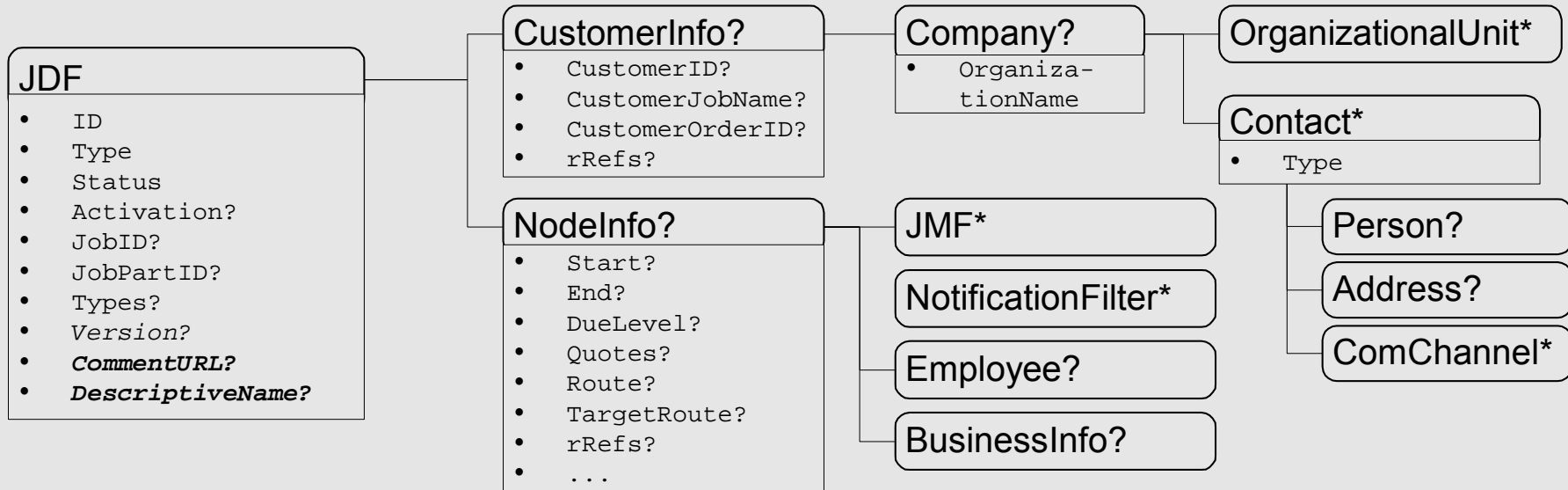


Customer / Node Information



- CustomerInfo
 - Map Subcontracting via Localized Customer Information in any JDF Node
 - Customer ID
 - Addresses (Delivery, Accounting, ...)
- NodeInfo
 - Scheduling
 - Deadlines
 - Processing Time Estimation

CustomerInfo and NodeInfo

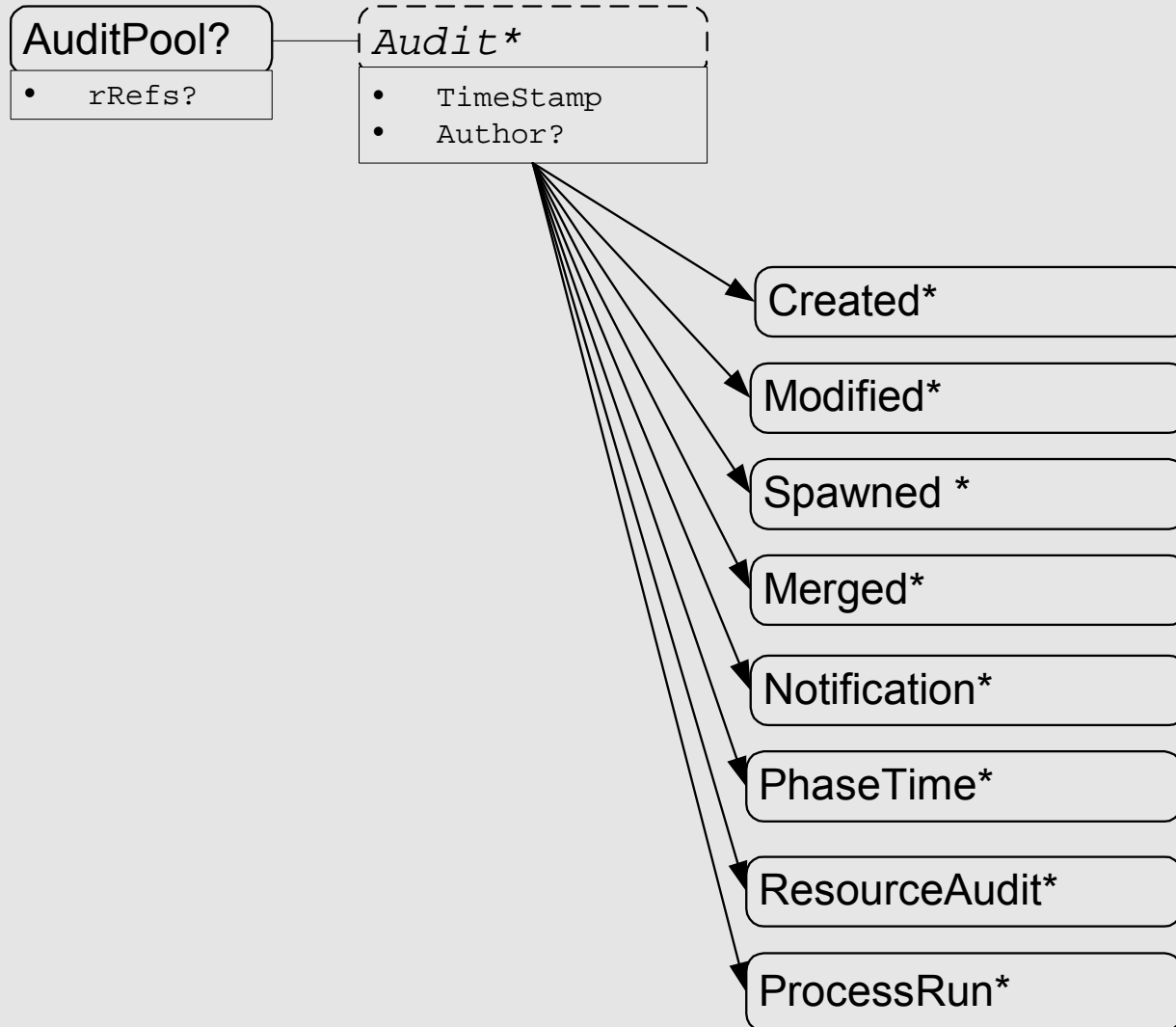


Audit Objects (1)

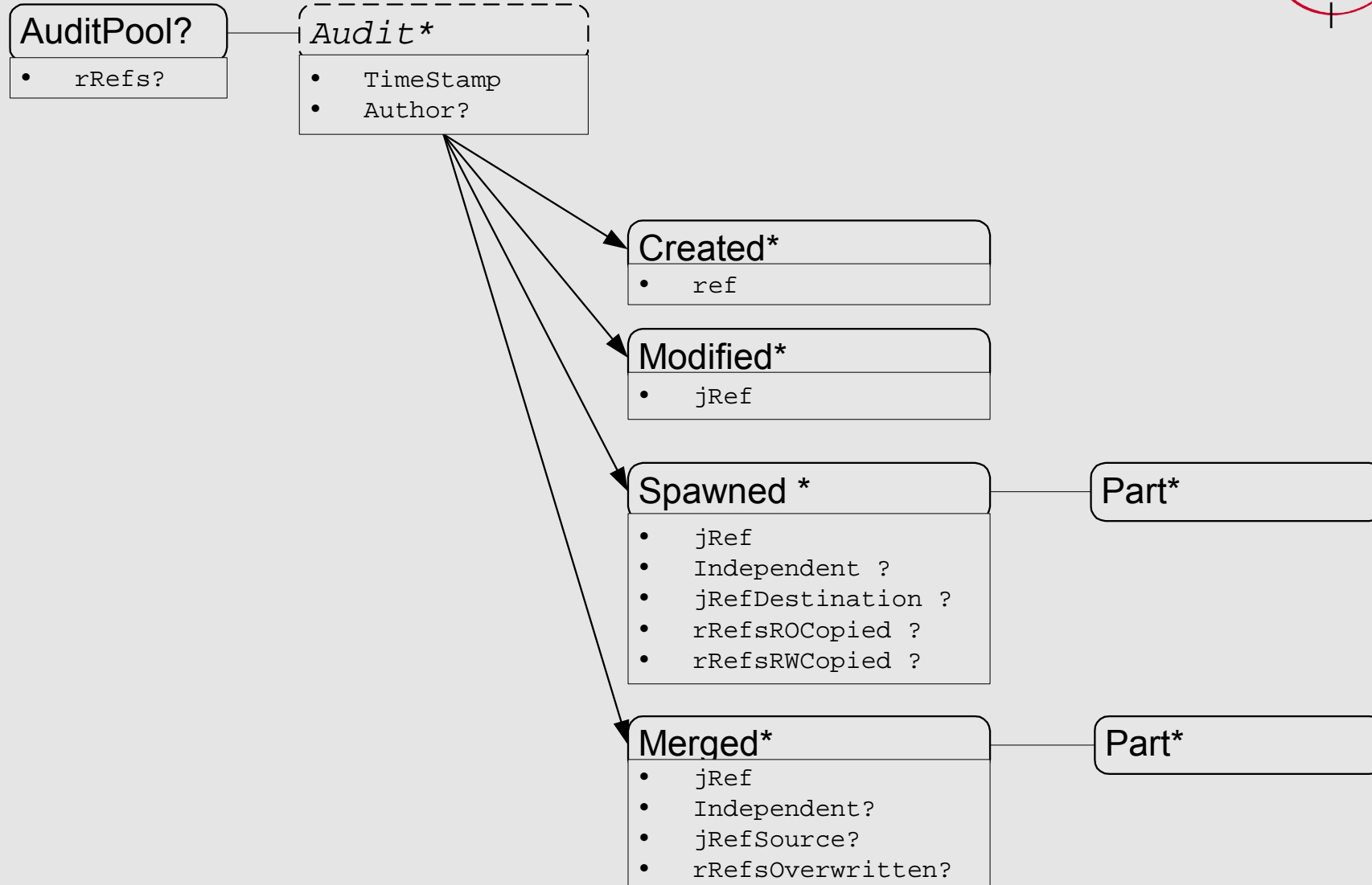


- Logging of Job Execution
 - Start time
 - End time
 - Phases
- Logging of Late Changes
 - Resources (used 85g Paper instead of 80g)
 - Consumables
- Status Summary
- Event Log

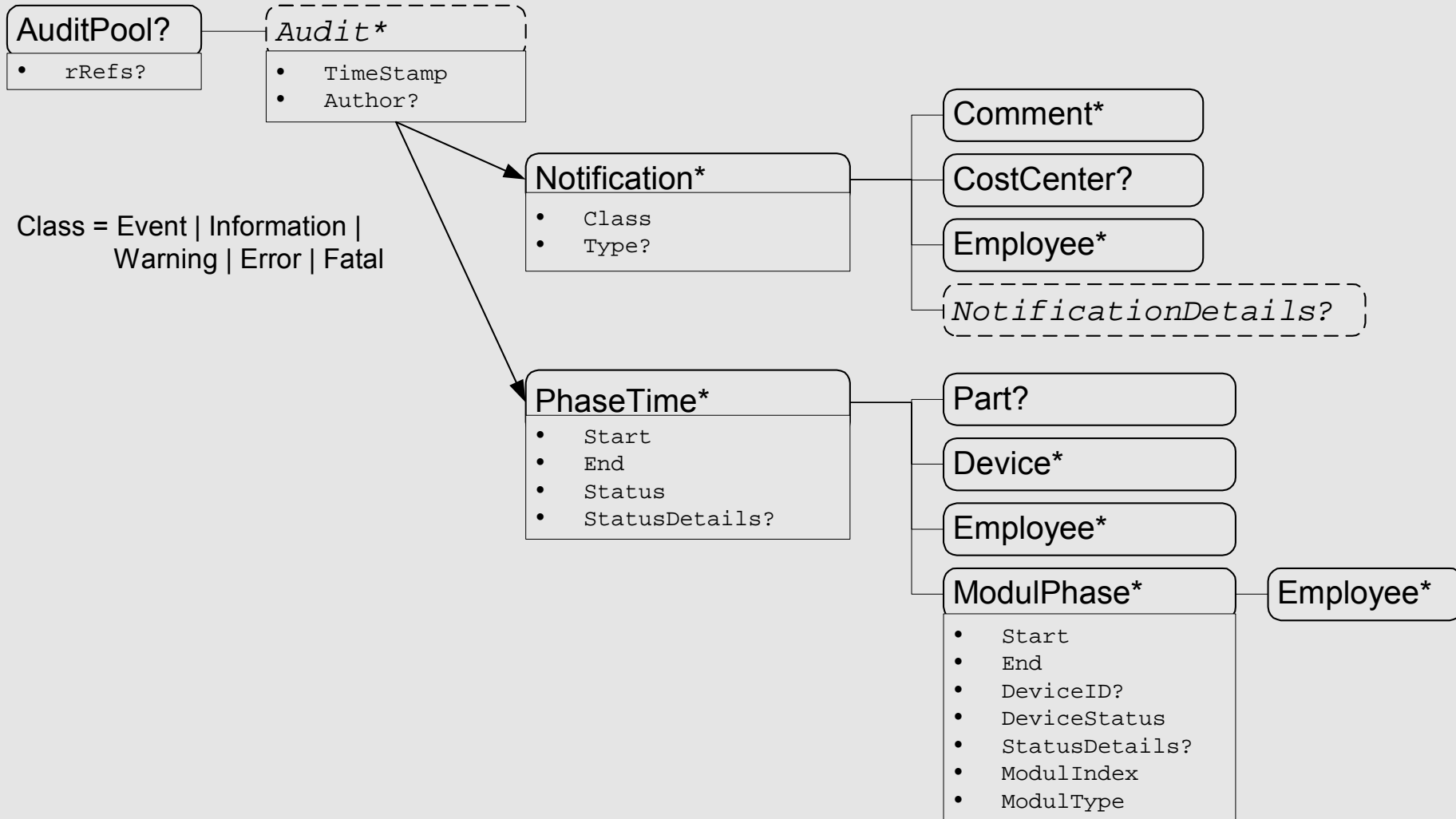
Audit Objects (2)



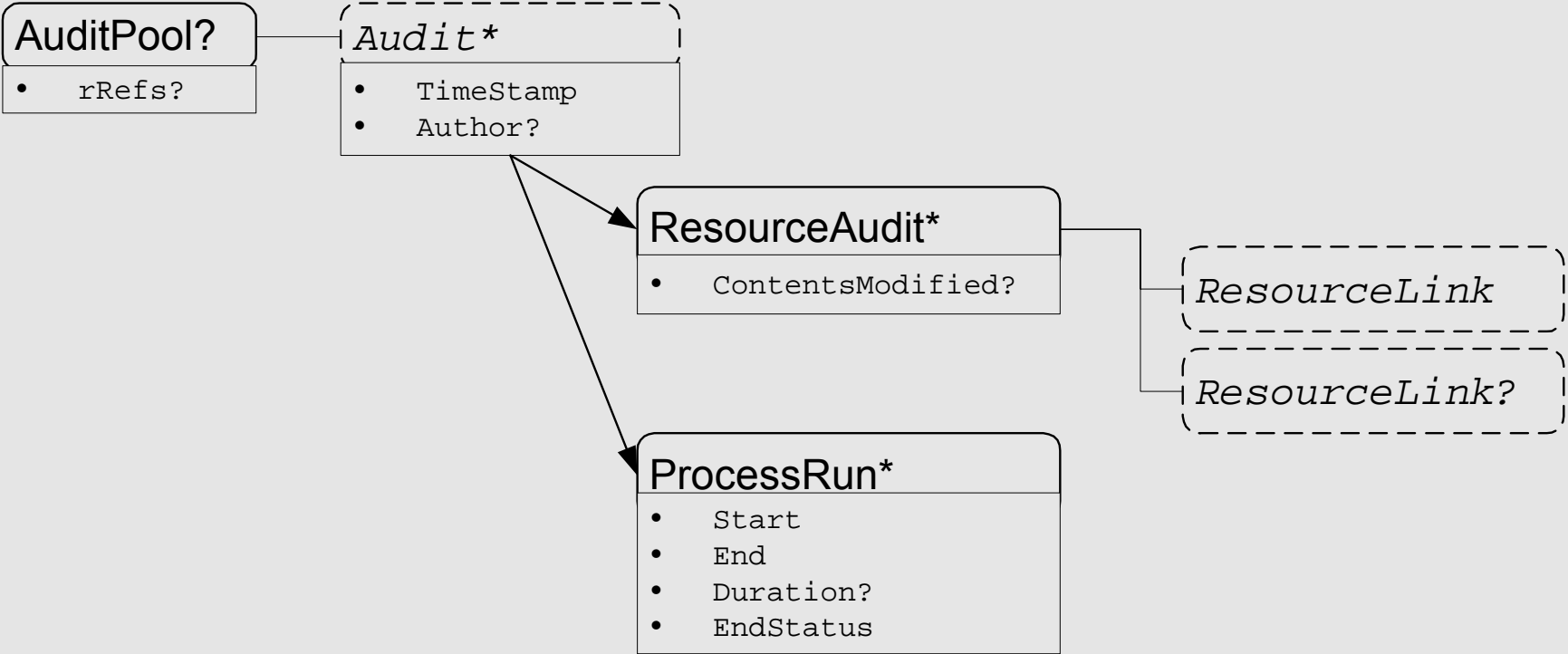
Audit Objects (3)



Audit Objects (4)



Audit Objects (5)



AuditPool -- Simple Example



```
<JDF ID="n0123" Type="Product" JobID="some product ID"  
  Status="Completed" Version="0.9">
```

...

```
<AuditPool>
```

```
  <Created TimeStamp="2001-05-18T12:15+01:00" ref="n0123"/>
```

```
  <ProcessRun Start=".." End="..." TimeStamp="..." EndStatus="Aborted"/>
```

```
  <Modified .../>
```

```
  <ProcessRun Start=".." End="..." TimeStamp="..." EndStatus="Completed"/>
```

```
</AuditPool>
```

```
</JDF>
```

Sub-Table of Contents



- JMF - JDF Messaging Format
 - Properties
 - Types I, II + III

JMF Messaging Properties

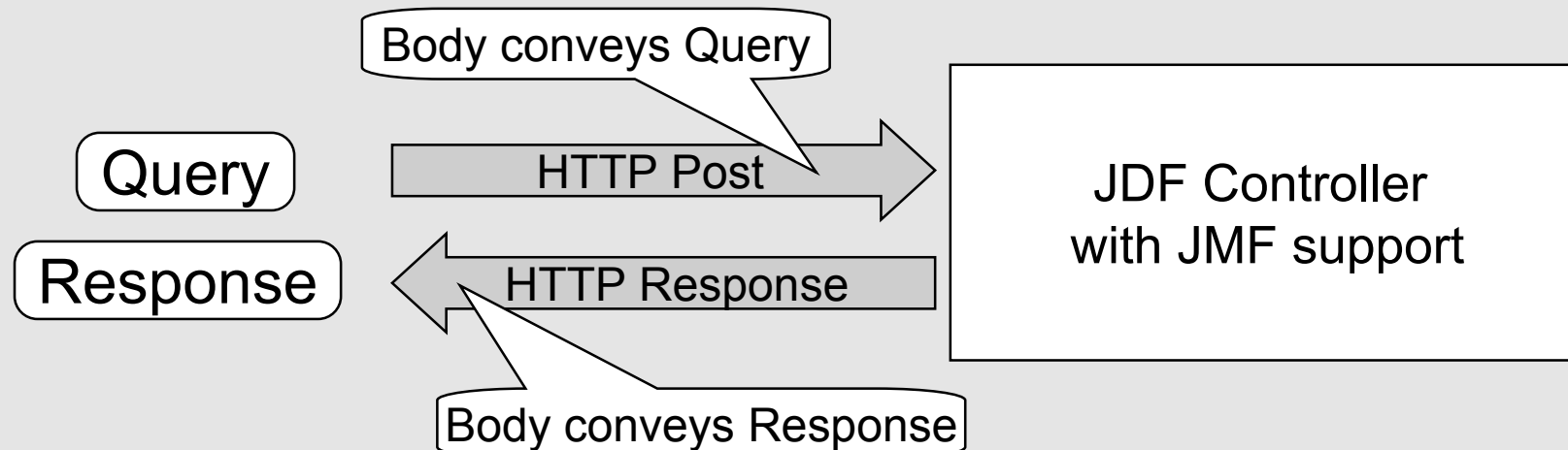


- Dynamic Process Interaction
- Five Message Families
 - **Query**: Request for information
 - **Command**: Request for a state change
 - **Response**: Immediate Answer to Query or Command
 - **Acknowledge**: Delayed Answer to Query or Command
 - **Signal**: Unidirectional Post
- Message Protocol using HTTP
 - Firewalls are not an obstacle
 - Easy to use and implement

Message Communication (I)



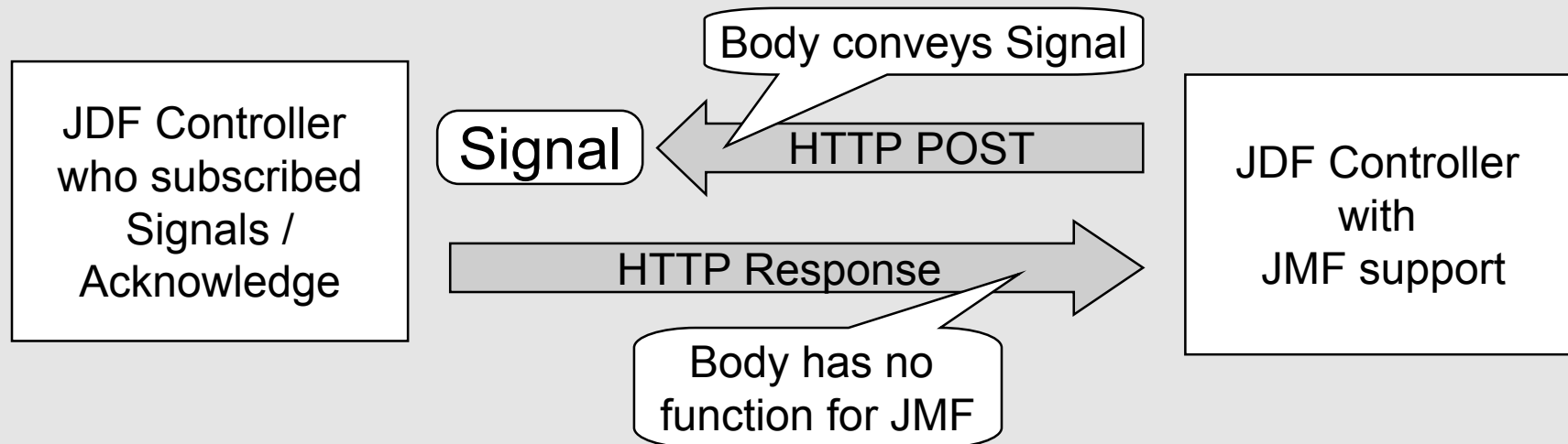
- Bi-directional
 - Query ==> Response
 - Command ==> Response
 - HTTP Protocol:
 - send: HTTP POST, Body contains the JMF::Query or JMF::Command
 - response: HTTP Response, Body contains the JMF::Response



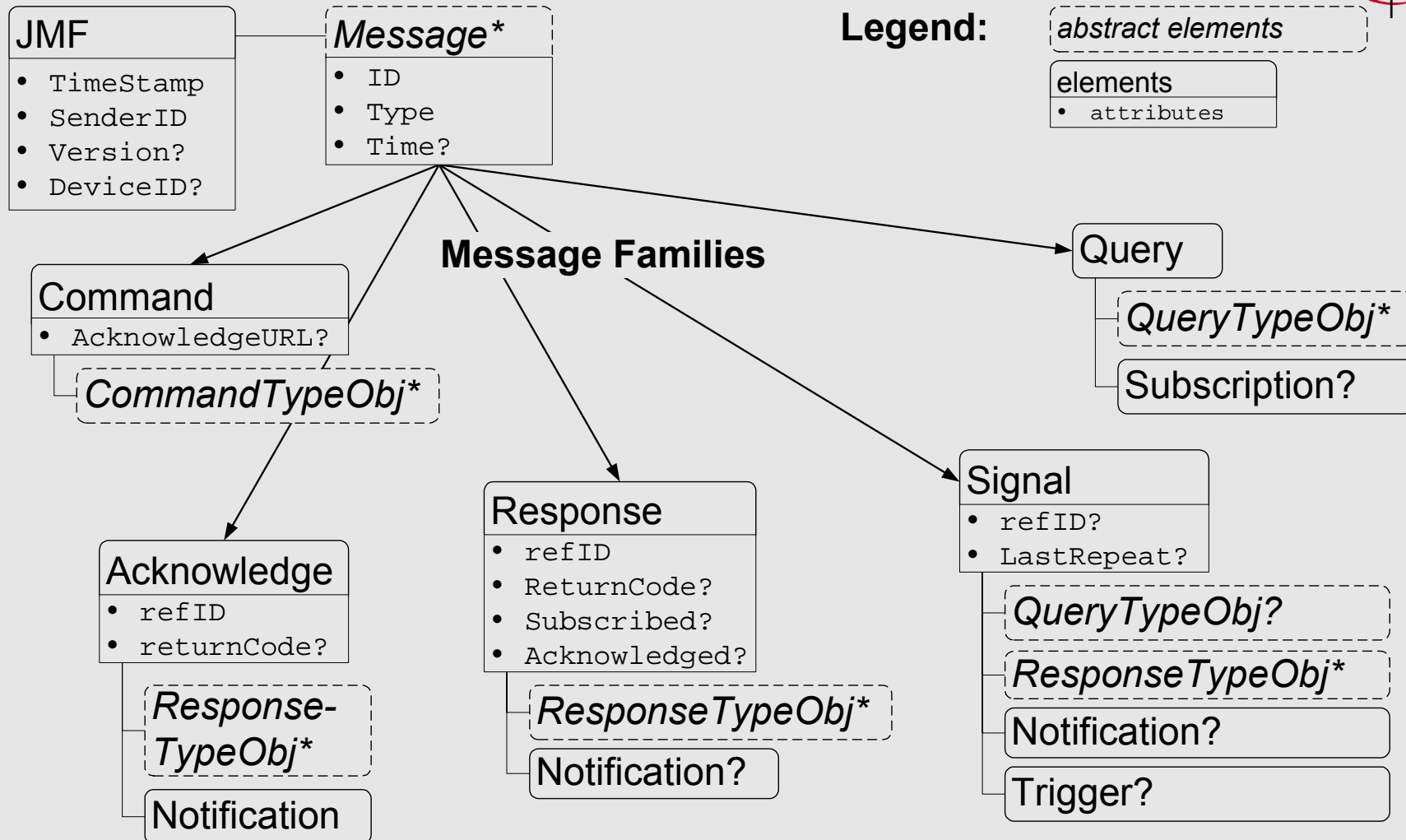
Message Communication (II)



- Uni-directional
 - Signal
 - Acknowledge
 - HTTP Protocol:
 - send: HTTP POST, Body contains the JMF::Signal or JMF::Acknowledge
 - response: HTTP Response, body stays empty.



JMF Message Structure



JMF Message Types (I)



- Initialization
 - Registration
 - Publish JDF Capabilities
- Device / Job Status / Progress Information
 - Consumable Level
 - Progress / Status
 - Settings
 - Currently executing jobs
 - Job Tracking

JMF Message Types (II)



- Queue Handling
 - Set Priority
 - Reorder / Group jobs
 - Hold / restart queued jobs (NOT running jobs)
 - Abort running Job
- Job Submission
 - Submission via HTTP
 - File Based JDF submission
 - hot folder
 - URL

JMF Message Types (III)



- Pipe Interactions
 - models the phone call from operator to operator
 - Start Production
 - Stop Production
 - High Water / Low Water marks
 - Individual Resource Requests
 - Resource Changes for iterative processing

Dynamic Pipe Linking (via JMF)

