



# XJDF



**HEIDELBERG**

Dr. Rainer Prosi

# Where will JDF be 2016?



- **JDF 1.51**
  - Freeze development except for bug fixes
- **JDF 1.6**
  - No major changes
  - Business as usual
    - Add some attributes, Processes, ...
- **JDF 2.0 / XJDF**
  - Reevaluate existing XML tools and align JDF closer to mainstream XML developments
  - Simplify Interface
  - Additional Business as usual - see JDF 1.6 above



- XML was mainstream but tools were scarce
  - Simple DOM level 1
  - Schema was virtually non-existing (we had DTD)
  - SOAP was yet another fledgling technology
- The discussion was: should we encode in COS (PDF) or XML.
- JDF design focus was to create one **complete job ticket** that describes an **entire job** including every step in the workflow and gets passed from device to device.



# What is next?



Ein Teil von jener Kraft,  
Die stets das Böse will und stets das Gute schafft.

Was ist mit diesem Rätselwort gemeint?

Ich bin der Geist, der stets verneint!

Und das mit Recht; denn alles, was entsteht,  
Ist wert, dass es zugrunde geht;  
Drum besser wär's, dass nichts entstünde.

So ist denn alles, was ihr Sünde,  
Zerstörung, kurz das Böse nennt,  
Mein eigentliches Element.

© Goethe



- JDF is too complicated
- JDF is too academic
- JDF jobs are static – changes are difficult to manage
- JDF will never work out of the box
- It is much too expensive to implement JDF
- Only the big guys will ever implement JDF
- It takes years to understand JDF
- JDF is not implemented on a large scale



- JDF exposes too many internal implementation specific details
- Change Orders are difficult to implement
- Gang Jobs with multiple products are messy to describe
- What is the role of Intent and price negotiation in Production?
- Plug & Play is evasive.
  - Device Capabilities descriptions are very complex
- JDF has too many methods to specify similar semantics; e.g.
  - Imposition
  - Stripping
  - LayoutPreparation





# XJDF interface Design Goals



- Retain the gory details that we learned in 15+ years of JDF Development
- Simplify the Interface
  - Flatten the learning curve for implementers
  - Make using JDF simple for simple applications
  - Reduce the number of methods to express similar things
  - Enable Plug&Play integration
- Separate Interface from Implementation
- Goal: It's just **<XML/>!**



# XJDF Xpath Compatibility



- **No Inheritance**
  - Flat List Structures
- **Single XJDF Node**
- **References are limited ID-IDREF searches**
  - All referenced Elements **MUST** have an ID
  - No Name Mangling of Links
  - Reduce use of references to the absolutely necessary
- **refElements are inlined or replaced by IDRef**
  - Define exactly one option allowed per individual subelement in JDF, i.e. IDREF or element.
- **ICS format changes to 1 simple table:**  
| XPATH | Conformance | Description |





# Compatibility with modern XML tools

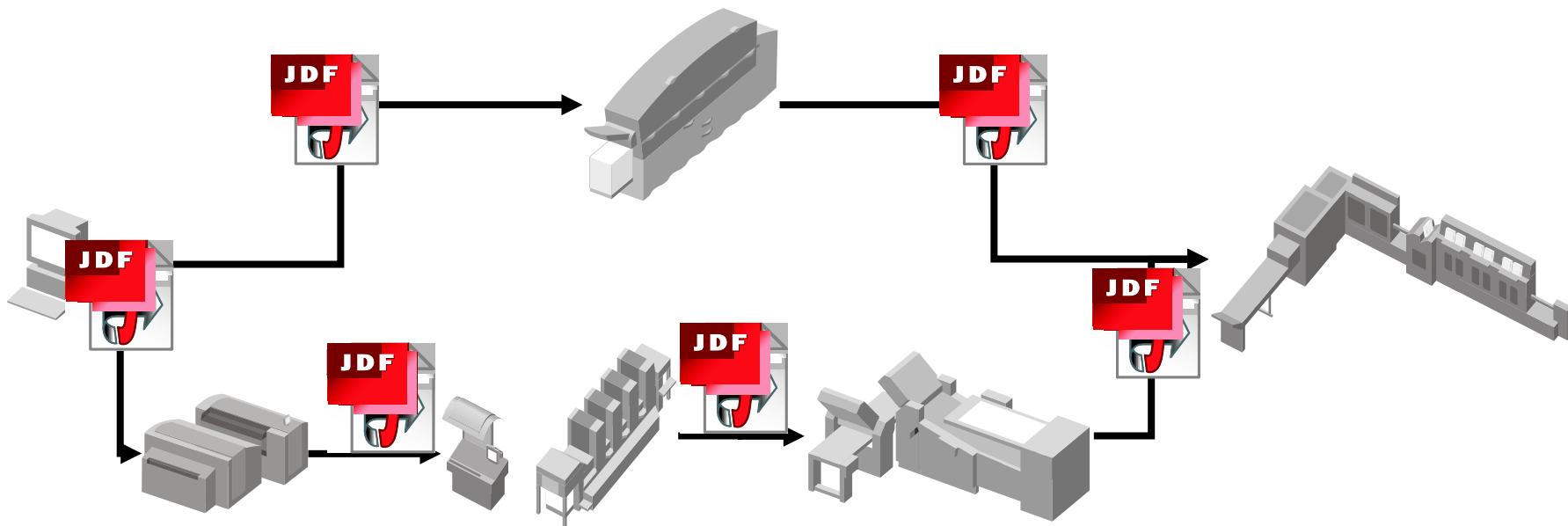


- **Make Compatible with**
  - **Standard class generators**
- **Allow simple Mapping to databases**
- **Define Simple XML Schema**
  - **Use XML Schema to describe device constraints**
- **Enable simple XSLTs for legacy format translations**
- **Enable XSLT for display, e.g. in Browsers**
- **Make standard XML knowledge more applicable to JDF**

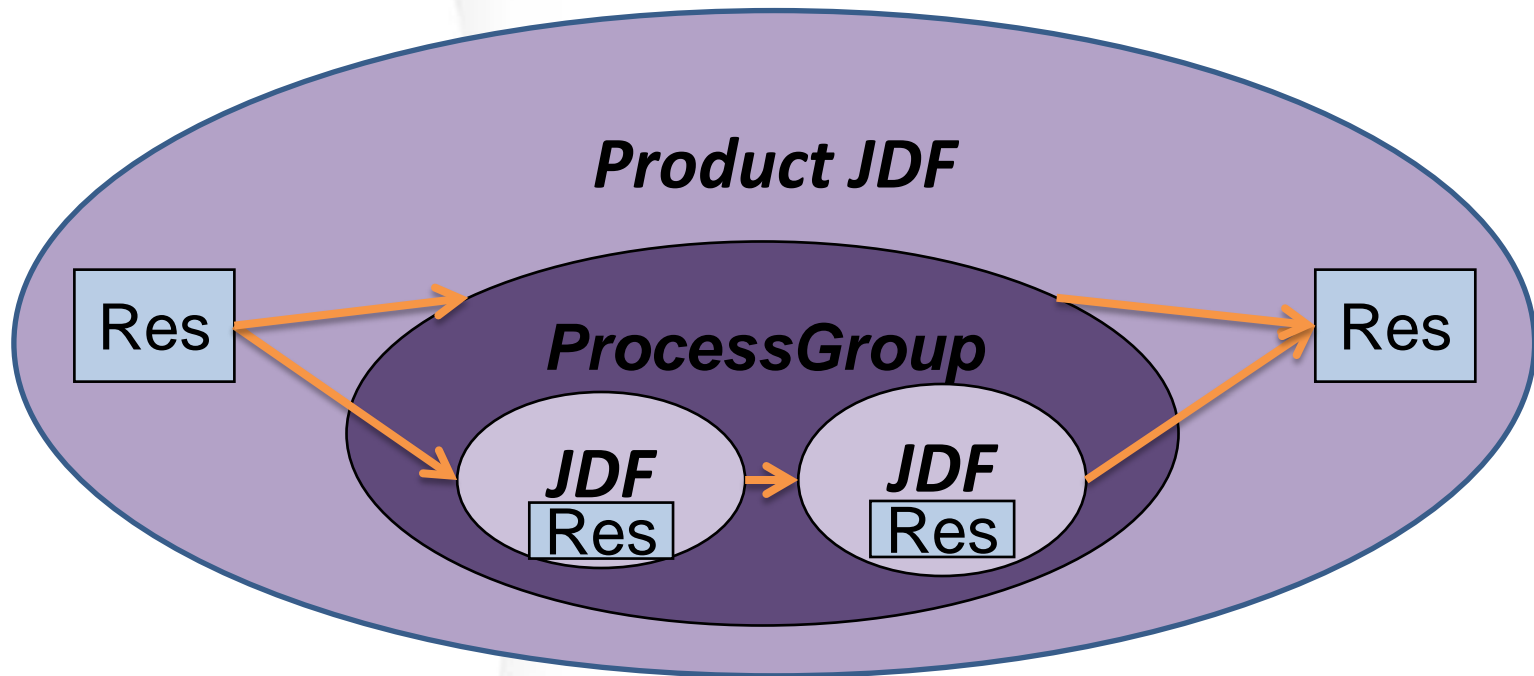
# JDF Process Description



- JDF describes the entire Production in one XML
  - Models Process interdependencies
  - All sub-processes are described in one single XML Job Ticket
  - May contain Manufacturing Instruction Details
  - Set of linked hierarchical JDF XML nodes
  - Devices are required to retain opaque data



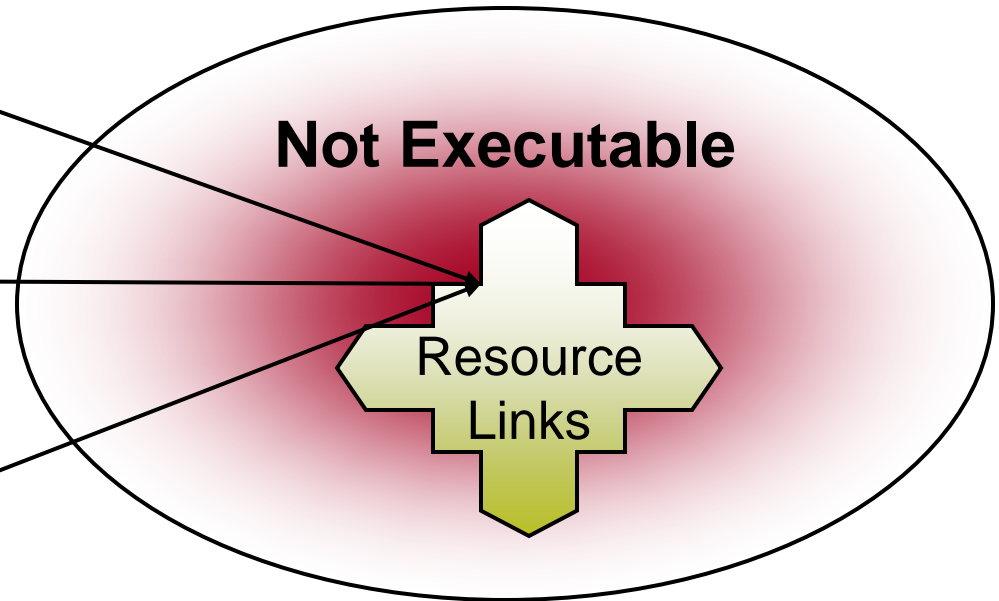
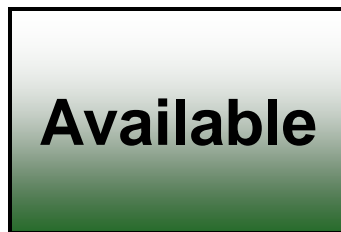
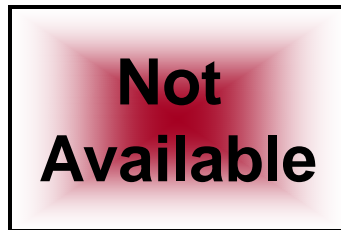
- Nested JDF Nodes
  - Resource Links define Resource Context



Resource

ResourceLink

JDF Node

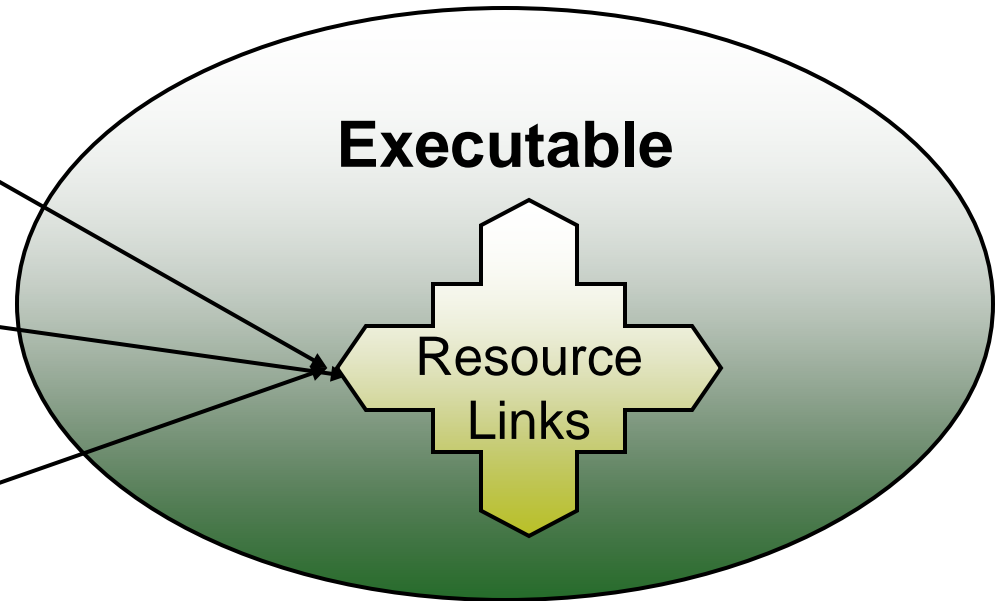
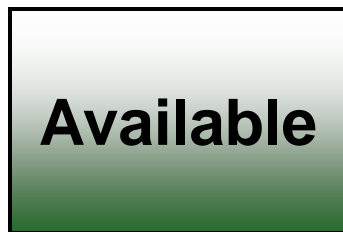
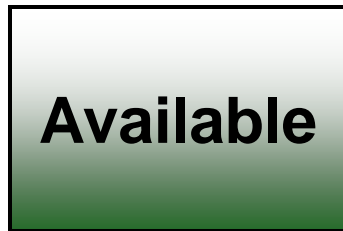


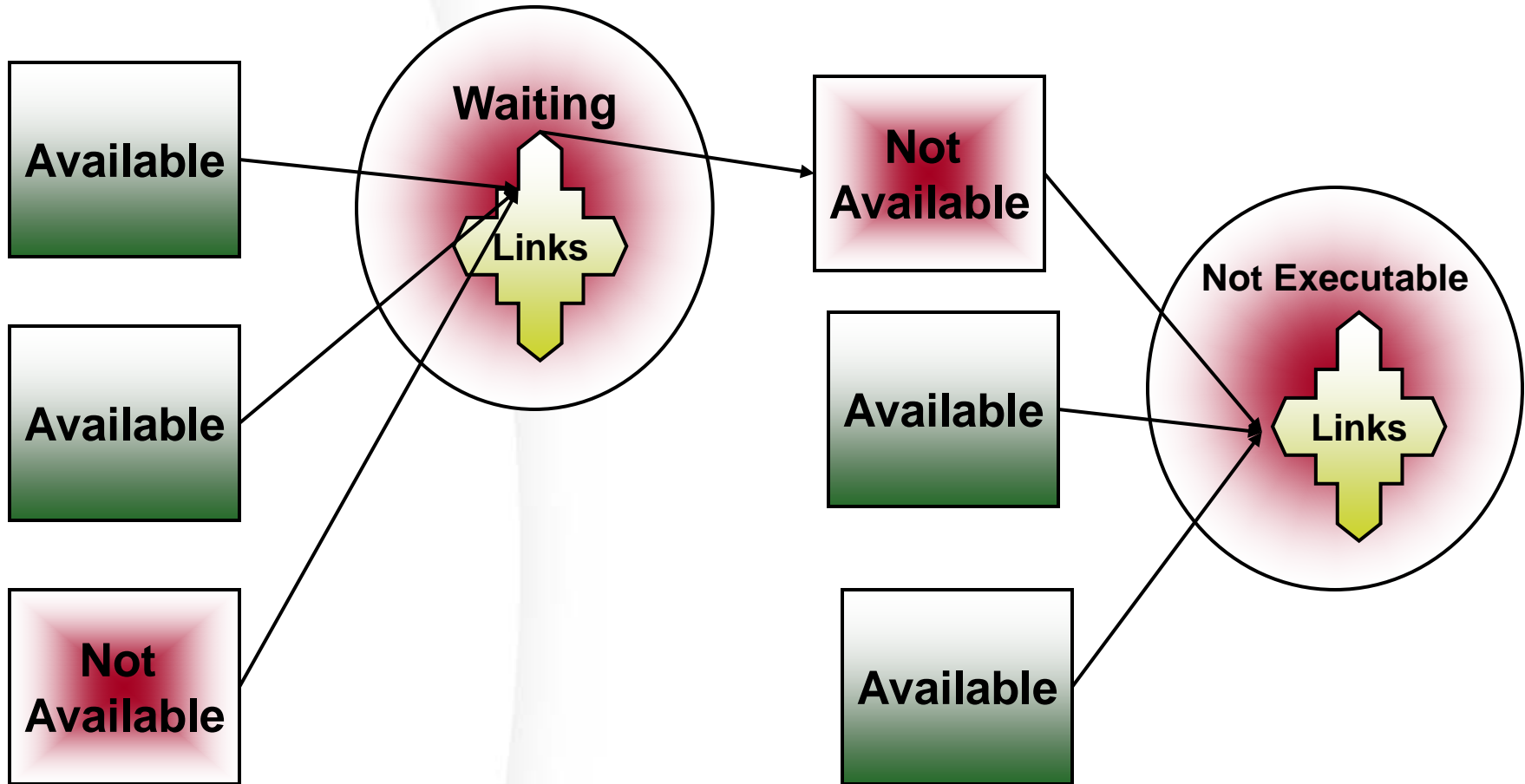
# Node Executability

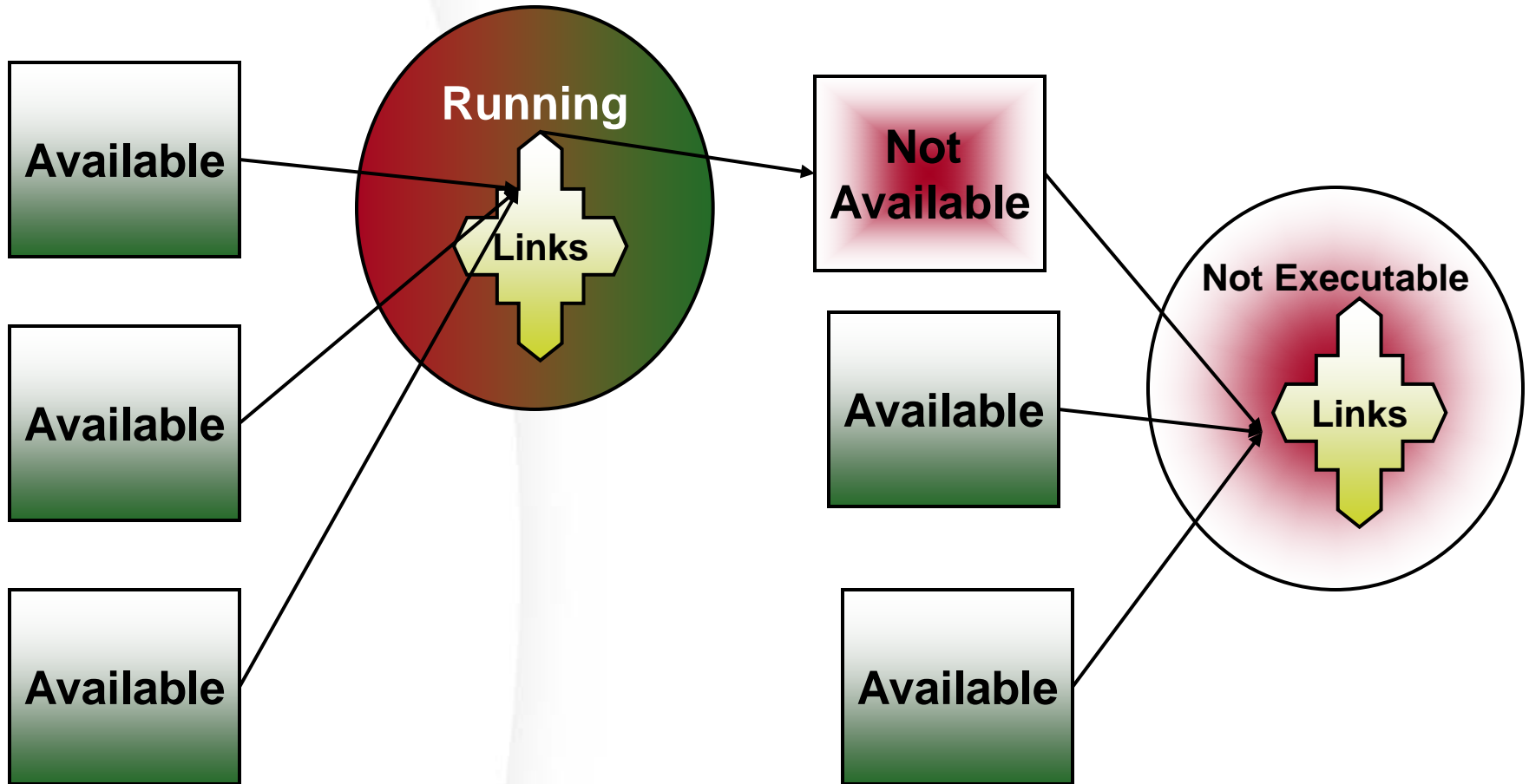
Resource

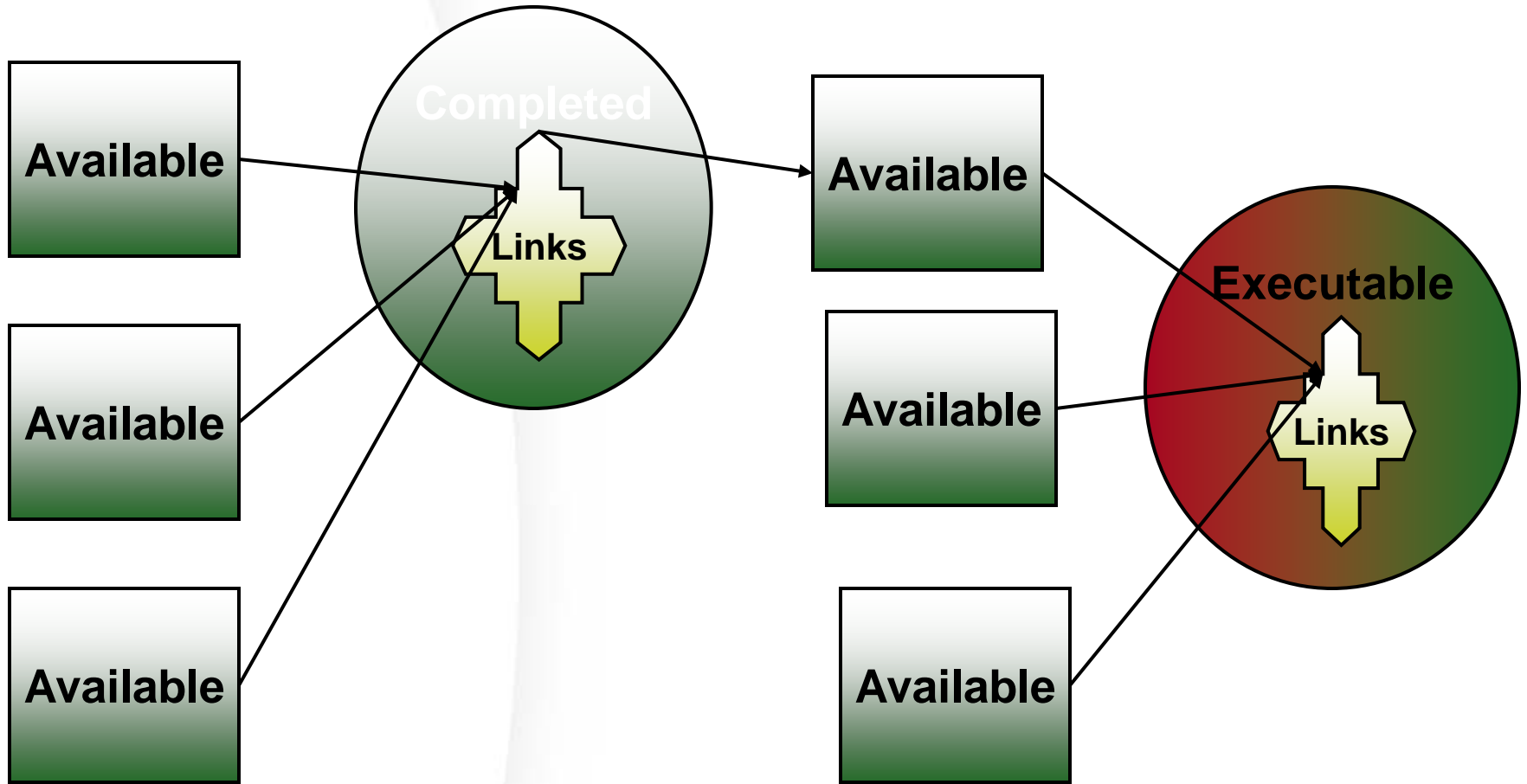
ResourceLink

JDF Node

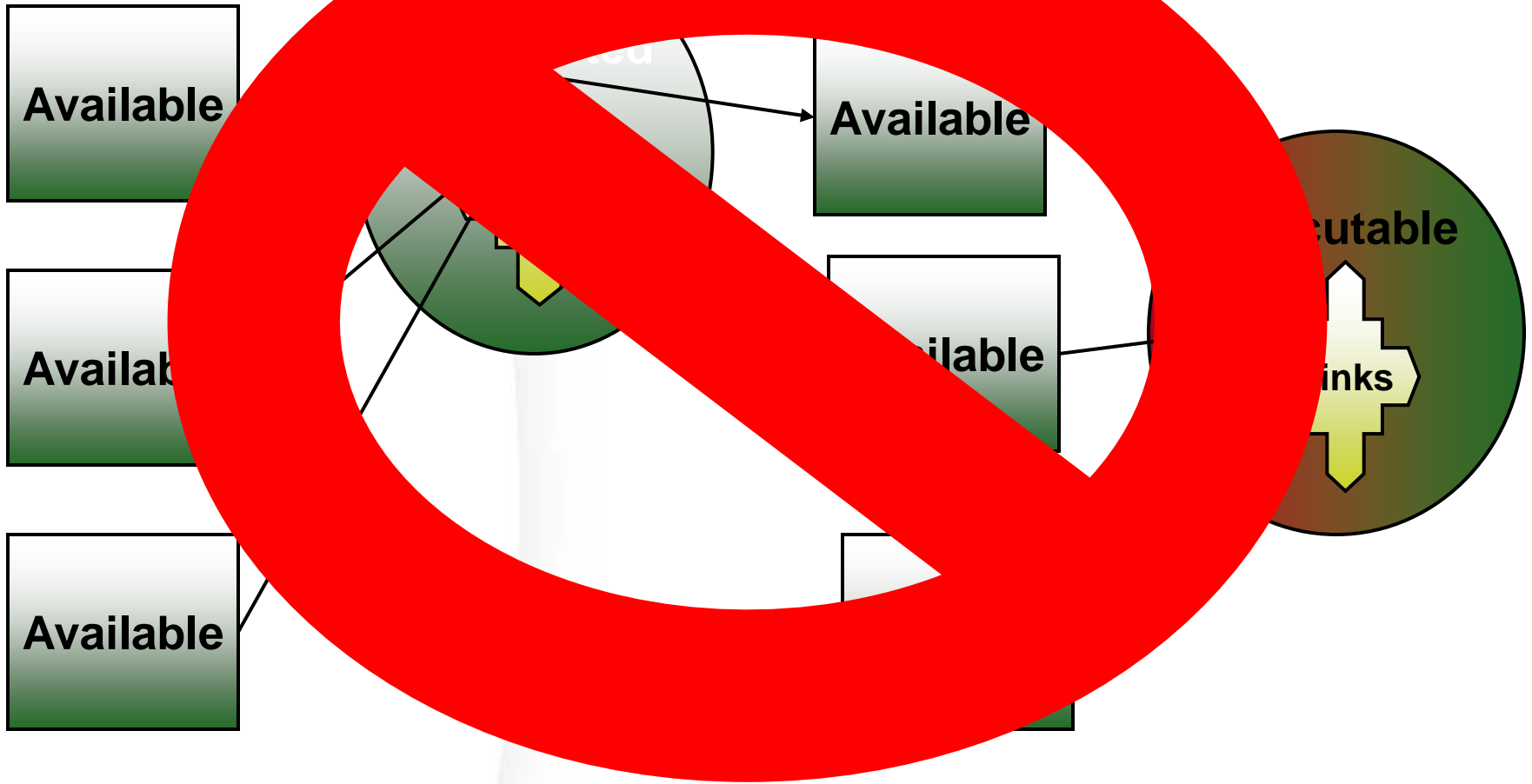










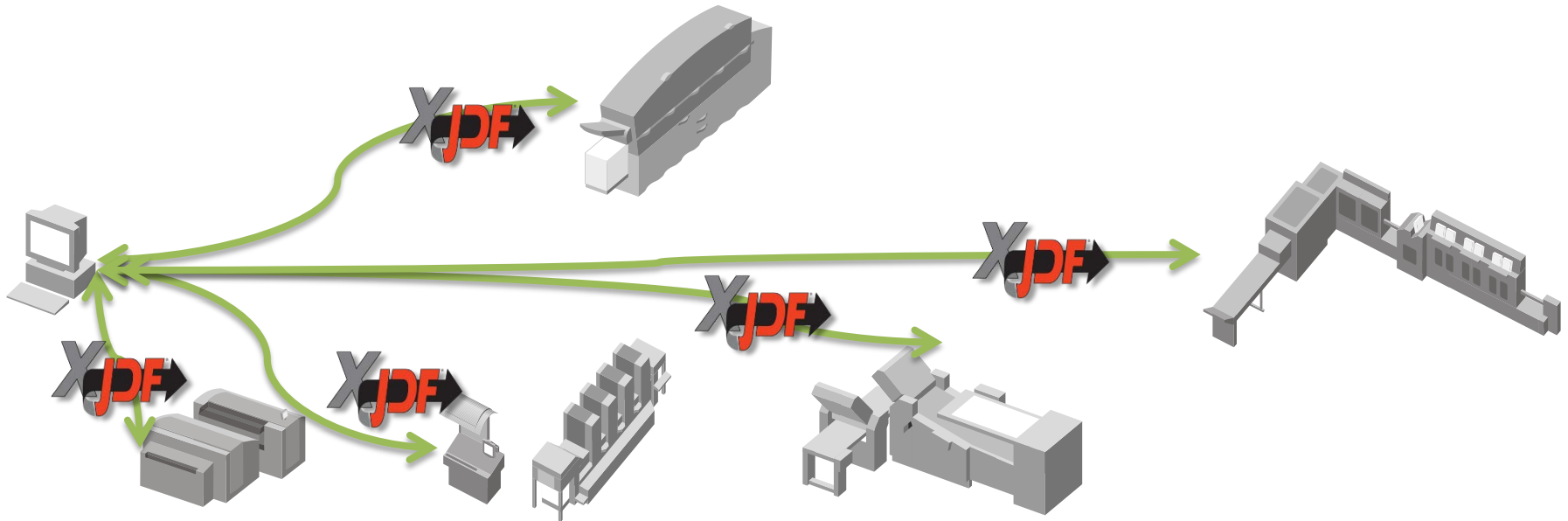




# XJDF Process Description



- **XJDF assumes that the process definition is proprietary**
  - Each transaction is dedicated to the recipient
  - May contain Manufacturing Instruction Details
  - Single XJDF XML node for each processor
  - No requirements to maintain opaque unknown data



# Job modification and Change Orders



- Anything but updating final Amounts or Scheduling is inherently difficult – regardless of the simplicity of the interface
  - Change number or size of pages - requires new impositioning
  - Modified finishing typically ripples back to prepress
- JDF:
  - Adding process steps to JDF requires insertion of a new exchange resource that is subsequently relinked.  
This is overkill for: “Please add varnish to this”.
- XJDF
  - All Attributes are optional in XJDF
  - Simply send a new XJDF with only modified values



# CIP4<sup>®</sup> Gang Jobs Organization

- JDF

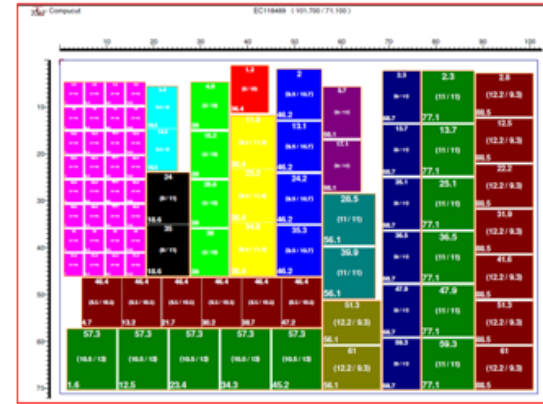
- The Product Hierarchy is inappropriate for gang jobs with multiple customers and/or final products

—A “Production Job” cannot be the child of multiple Customer Product elements.

- The hierarchical structure of JDF requires complete rewriting to add a small part from an additional customer product

- XJDF

- ProductList and Product are in a separate element
  - Modifications have no effect on the overall XJDF structure



# CIP4<sup>®</sup> Product Intent



- JDF
  - JDF allows for ranges in intent descriptions to support negotiation. This adds complexity in production where ranges are no longer required
- XJDF
  - XJDF assumes that all negotiation takes place in a dynamic , e.g. web to print environment
  - All intent values are simple attributes
- PrintTalk and XJDF
  - PrintTalk 2.0 simply contains XJDF elements instead of JDF



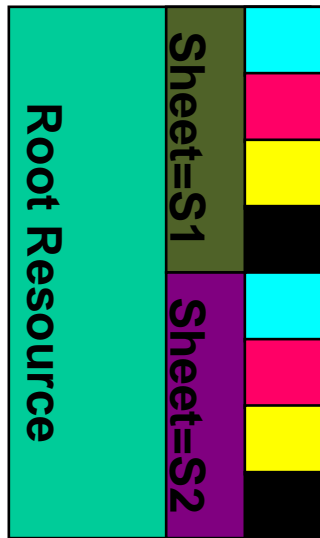
Integration through Cooperation

- How to describe the restrictions of a real device?
- JDF
  - The flexibility of JDF and many of the non-standard XML constructs make XML schema difficult to implement
- XJDF
  - Referenced XML Schema (it's just <XML/>...)
    - ✗ Slightly reduced functionality
      - No constraints descriptions
    - ✓ Standard XML Technology
    - ✓ Subset of the XJDF Schema



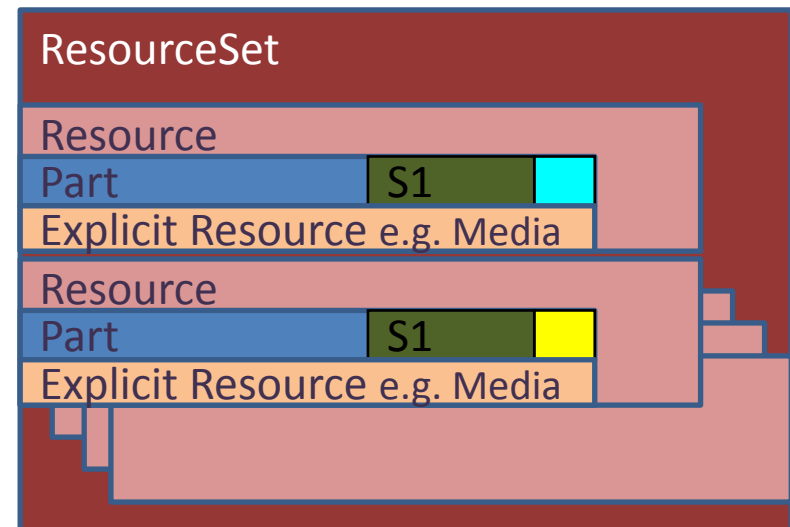
## JDF

- ✗ Hierarchical tree
- ✗ Element Inheritance
- ✗ Abstract Types



## XJDF

- ✓ Linear list
- ✗ Redundancy
- ✓ Explicit Elements





# Reduce redundant methods



- Remove all deprecated items
- Review and remove historical synonyms
- Retain all necessary details
- Keep translation XJDF  $\leftrightarrow$  JDF simple



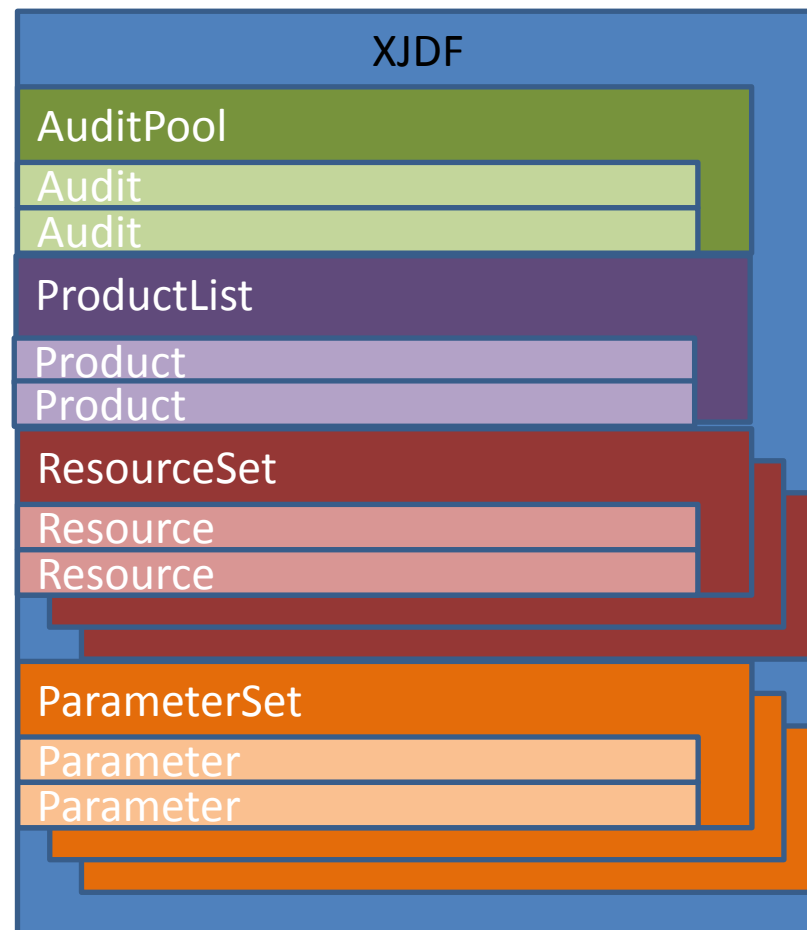
HEIDELBERG



Integration through Cooperation



- Audits are the same as in JDF
- Products are no longer JDF Nodes but are defined a a list of <Product>
- ResourceLink and Resource Elements have been merged into one ResourceSet or ParameterSet element



- The XJDF interface is simpler and therefore:
  - More robust
  - Projects can in be implemented more cost efficiently
    - Less Flexibility translates to less misunderstanding.
  - XJDF can be implemented by non-experts (e.g. in a home grown system)
- XJDF Projects can grow on an organic growth path
  - Amount/ Time/ Device
  - ...
  - Complete automation

- XJDF is **NOT** a JDF Dialekt.
  - Combatibility by format conversion
- XJDF must be implemented
  - Short-term Pain for long-term Gain
  - Long-term maintenance of 2 interfaces
  - XJDF can be implemented by non-experts (e.g. in a home grown system)



# XJDF project status



- Very active work group in CIP4
  - Weekly telephone conferences
  - Face to face meetings every 2 months
  - <https://confluence.cip4.org/display/PUB/XJDF>
- JDF 1.5 and XJDF co-exist in a common master specification
- JDF Open source Converter available (Java)
  - JDF → XJDF
  - XJDF → JDF
- <https://confluence.cip4.org/display/PUB/JDFEditor>

- **XJDF defines a simplified JDF structure that:**
  - Retains much of JDF 1.x semantics
  - Aligns with modern XML Tools
  - Is useful as an interface
  - Is less useful as an internal application representation
  - Enables quick and simple integration in printing
- **XJDF and JDF will be maintained in parallel**
  - Common master specification with conditional output
- **66% of the work is on chapter 1-4 of the spec**
  - Of the 66% work, at least 50% are deletions!



# XJDF Prototypes



- Heidelberg Prinect: Prototypes working for 4 years
  - Focus: WebToPrint; Print subcontracting, CAD Connectivity
- Partners:
  - Vendors, that have also implemented JDF
  - Web To Print vendors with no JDF experience
  - Printer software (home grown)



**XJDF is work in progress and still subject to major discussions and modifications.**

**For more details, see:**

**<https://confluence.cip4.org/display/PUB/XJDF>**

**You can also follow the discussions by registering as a CIP4 visitor (or even better – CIP4 Member) and then accessing:**

**<https://confluence.cip4.org/display/xjdf/XJDF+WG>**