

Workflow Automation with CIP4 Technology

CIP4 Technologies I / Exchange JDF

Heiko Angermann

XJDF / Difference to JDF

- ▶ XJDF stands for Exchange Job Definition Format. The current release 2.1, was published 16 August 2020:
 - » [Specification](#)
 - » [XJDF Website](#)
- ▶ XJDF is based on XML language (~~e~~XML).
- ▶ XJDF is the younger sibling of JDF, and is easier to implement than JDF.
 - » It is a simplified version of JDF.
 - » It is a re-design of the older JDF.

XJDF / Difference to JDF

- ▶ XJDF workflow descriptions are not part of XJDF any more:
 - » JDF describes the production workflow.
 - » XJDF is a communication protocol between two applications.
No production context.
- ▶ XJDF is not compatible to JDF regarding the code:
 - » XJDF can be automatically converted to JDF and vice versa.
However, workflow logic will be ignored.

XJDF / Difference to JDF

- ▶ XJDF and JDF are used in the industry in parallel.
- ▶ XJDF and JDF are using mostly the same processes and resources.
- ▶ XJDF does not contain the entire workflow logic:
 - » After a print job has been produced, JDF contains the complete description of the production details (for the production processes that use JDF).
 - » XJDF is merely an interface protocol between a production controller and specific devices.
 - » XJDF is not necessarily stored
 - » Easier to implement, less costly – especially for the devices.
 - » Workflow logic is stored in private data bases of controller.

XJDF / Reduction of Complexity

- ▶ Reduction of Complexity, as only JDF is designed to be a complete electronic job-ticket. XJDF in contrast:
 - » Is designed to be a pure information-interchange interface typically between one controller and one device.
 - » This leads to faster, simpler, and more robust integration of devices and applications.

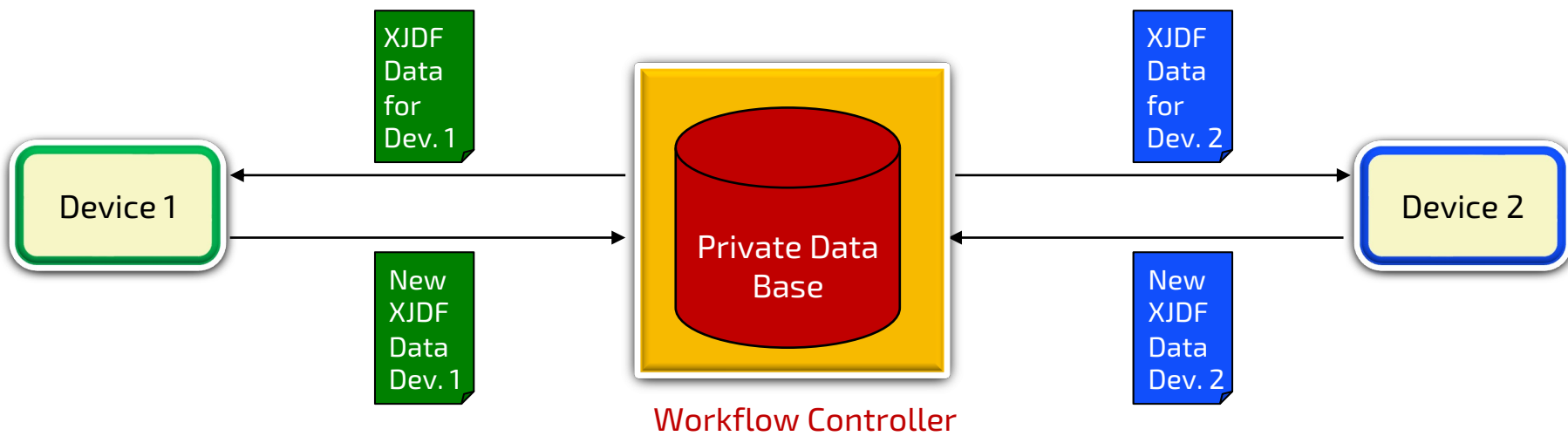
- ▶ Reduction of Complexity, as like JDF, XJDF is also XML encoded. However, instead of JDF nodes, XJDF nodes are now specified.
 - » JDF is not pure XML, but has additional properties such as inheritance

XJDF / Reduction of Complexity

- ▶ Reduction of Complexity, since the XJDF addresses a single device only, an XJDF node has no children:
 - » Therefore, the **no** nested node tree we saw in JDF exists.
 - » Each device receives exactly the information it needs and does not have to search for the appropriate information.
 - » Information reduced to details actually required.
- ▶ Reduction of complexity, as In JDF each print job yields to a separate JDF. With XJDF, different products can be written to a product list if they are all to be processed in the same way by a single XJDF compatible device.

XJDF / Reduction of Complexity

- ▶ Reduction of Complexity, summary:
 - » Devices need not to extract relevant data from the job ticket
 - » Data bases are handling data update.
 - » No need for extra mechanism (*spawn & merge*) as with JDF.
 - » One XJDF node only with XJDF, no node tree, no ResourceLinks.
 - » XJDF is a new version with the same idea.



| Criteria | JDF | XJDF |
|----------------------|------------------------------|----------------------------------|
| Nodes | Node Tree | Single Node |
| Resources | Located in a ResourcePool | Inside the node in a ResourceSet |
| ResourceLinks | Yes | No |
| Workflow Description | Yes | No |
| Node Hierarchy | Yes | No |
| XML | Extended XML | Standard XML |

XJDF / Reduction of Complexity

- ▶ Reduction of Complexity, ~~results~~ Benefits for the user:
 - » Devices are easier and faster to implement.
 - » In future: More XJDF devices on the market.
 - » Technology less expensive for users.
 - » XJDF more suitable for smaller companies.

Literature / More XJDF Articles

- ▶ [Specification](#)
- ▶ [XJDF Website](#)
- ▶ [Whitepaper XJDF](#)
- ▶ [XJDF book](#)