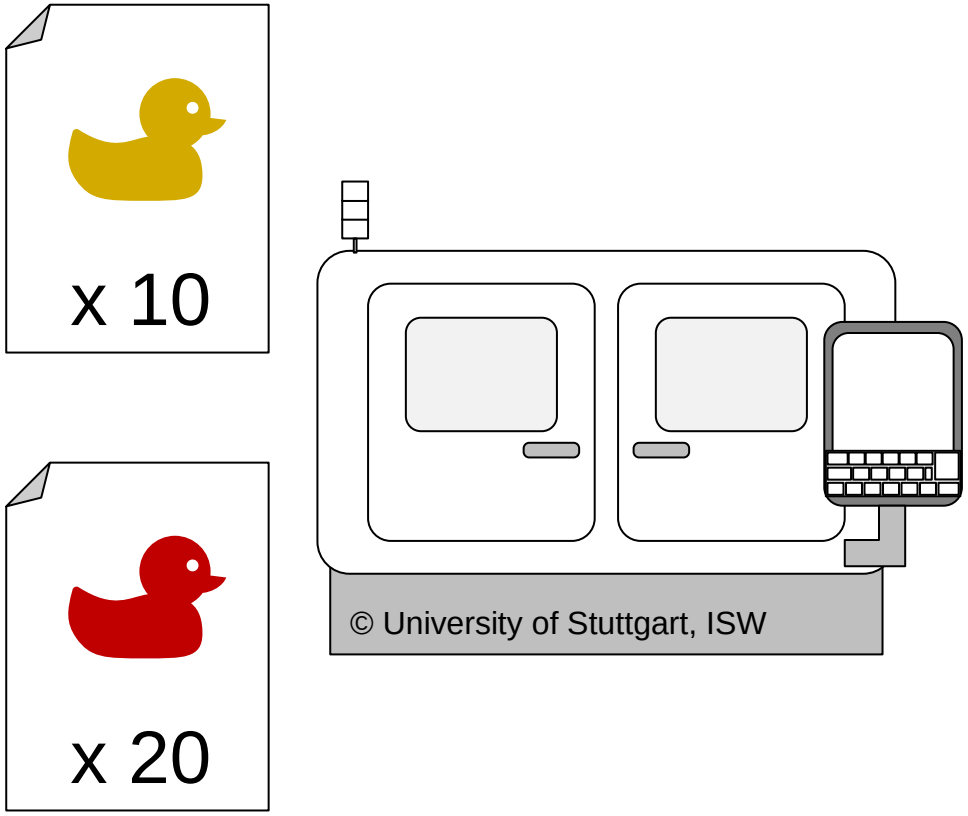
OPC UA 40501-1

Introduction OPC UA for Machine Tools Job Model

Tonja Heinemann, University of Stuttgart, ISW

Intention for the Machine Tools Job Information Model

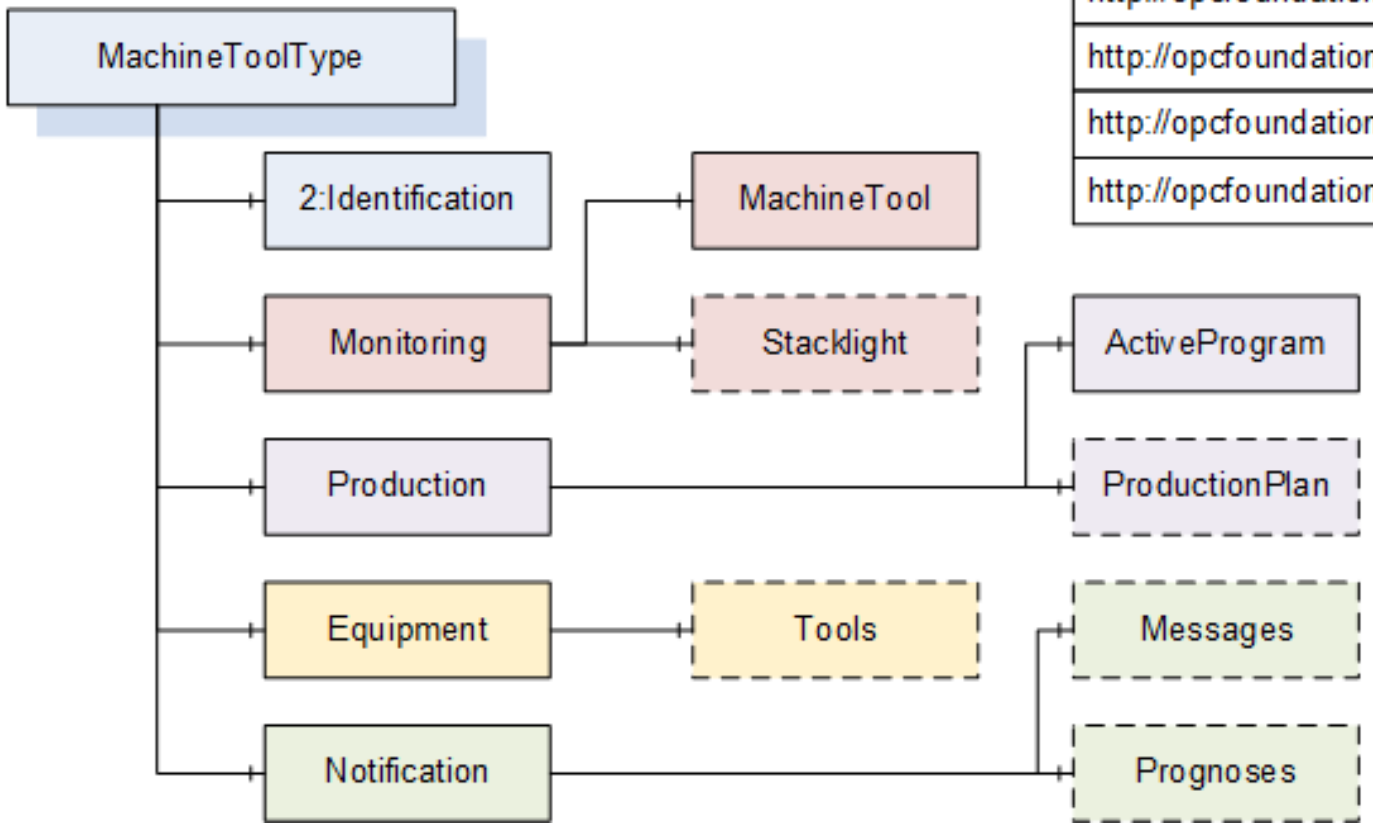


How productive is the machine?

How long does one job take?

What is the machine doing?

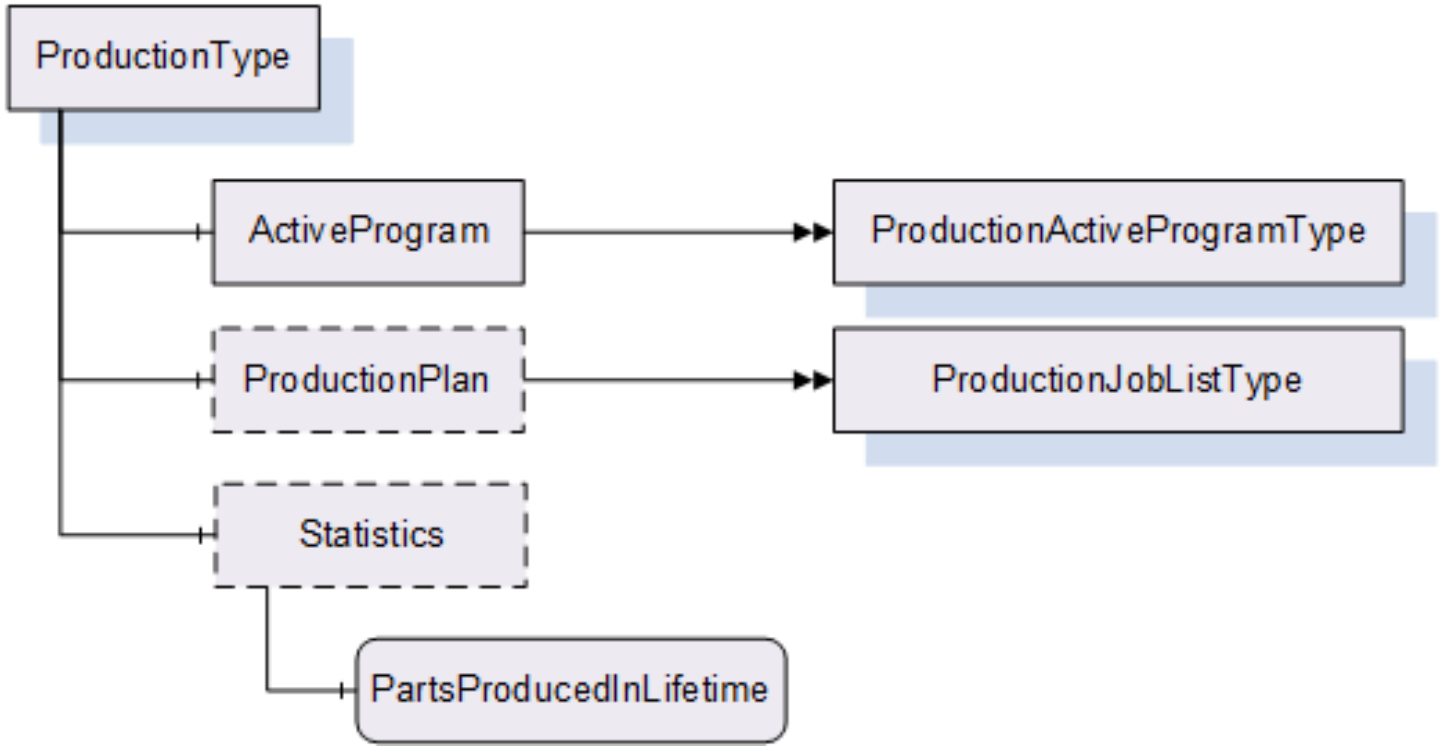
Machine Tools Information Model



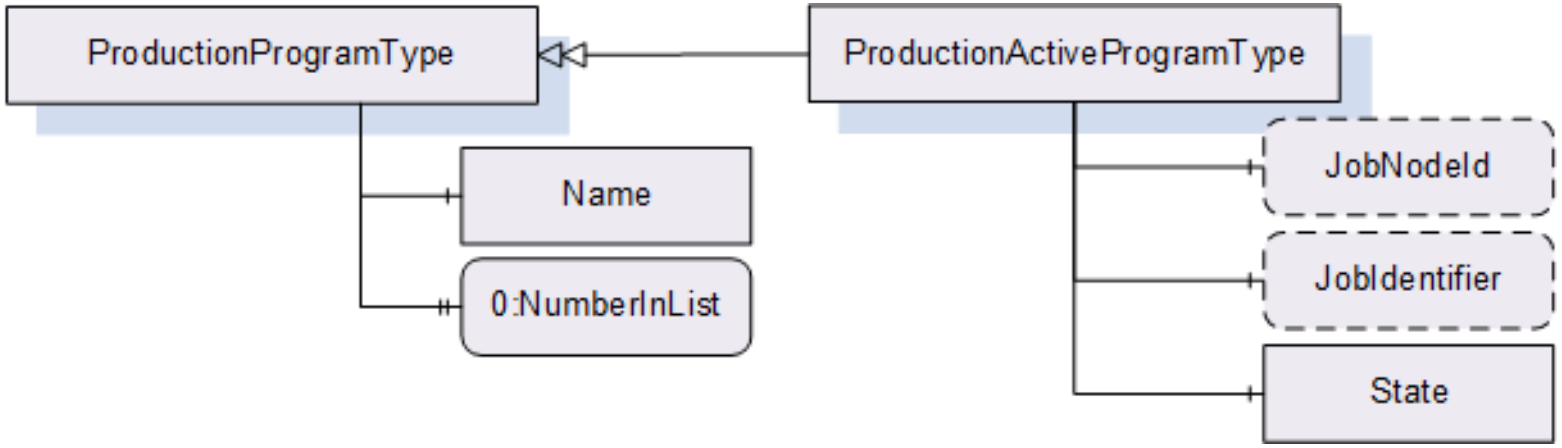
NamespaceURI	NamespaceIndex
http://opcfoundation.org/UA/	0
http://opcfoundation.org/UA/DI/	2
http://opcfoundation.org/UA/Machinery/	3
http://opcfoundation.org/UA/IA/	4

Specification online reference

Production

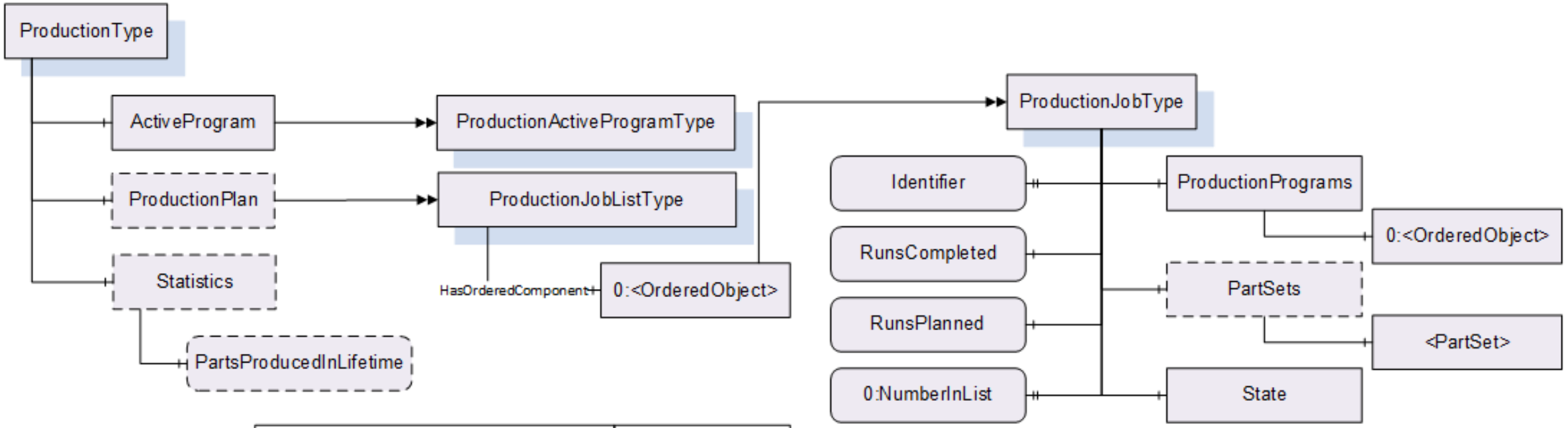

ProductionType
ActiveProgram
ProductionActiveProgramType
ProductionPlan
ProductionJobListType
Statistics
PartsProducedInLifetime

ProductionActiveProgramType



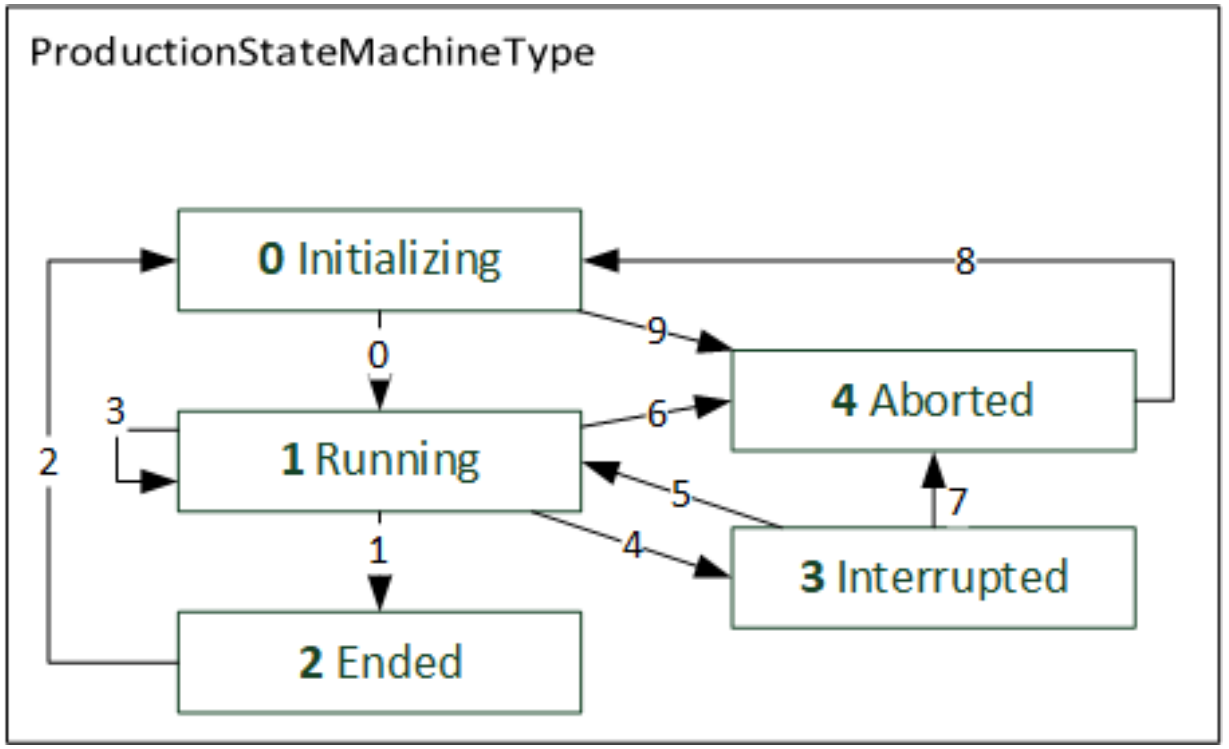
NamespaceURI	NamespaceIndex
http://opcfoundation.org/UA/	0
http://opcfoundation.org/UA/DI/	2
http://opcfoundation.org/UA/Machinery/	3
http://opcfoundation.org/UA/IA/	4

Production



NamespaceURI	NamespaceIndex
http://opcfoundation.org/UA/	0
http://opcfoundation.org/UA/DI/	2
http://opcfoundation.org/UA/Machinery/	3
http://opcfoundation.org/UA/IA/	4

ProductionStateMachineType



ProductionJobStateMachineType

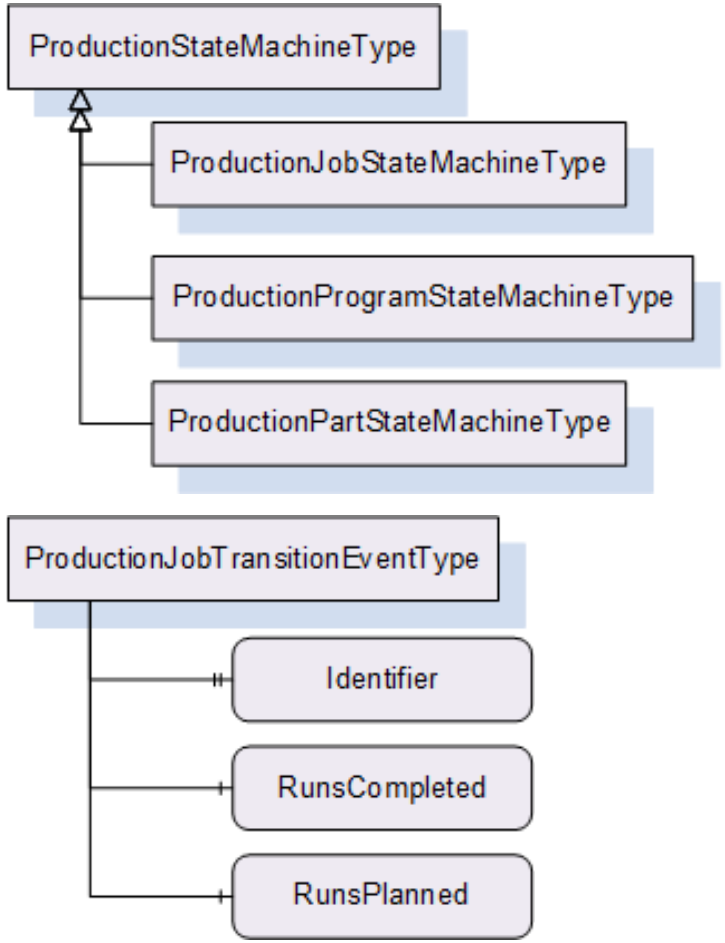
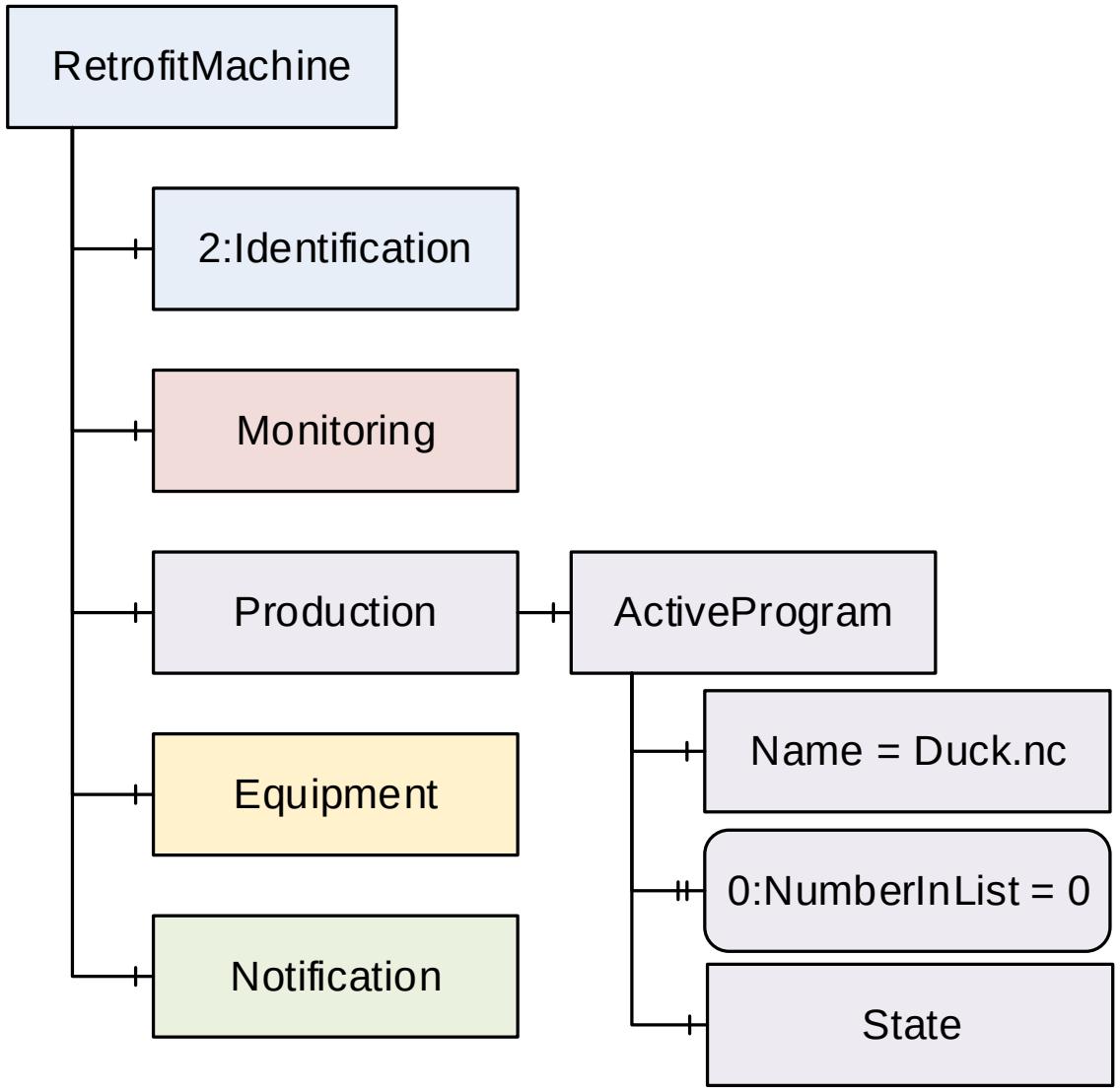


Table 43 - ProductionJobStateMachineType Transitions

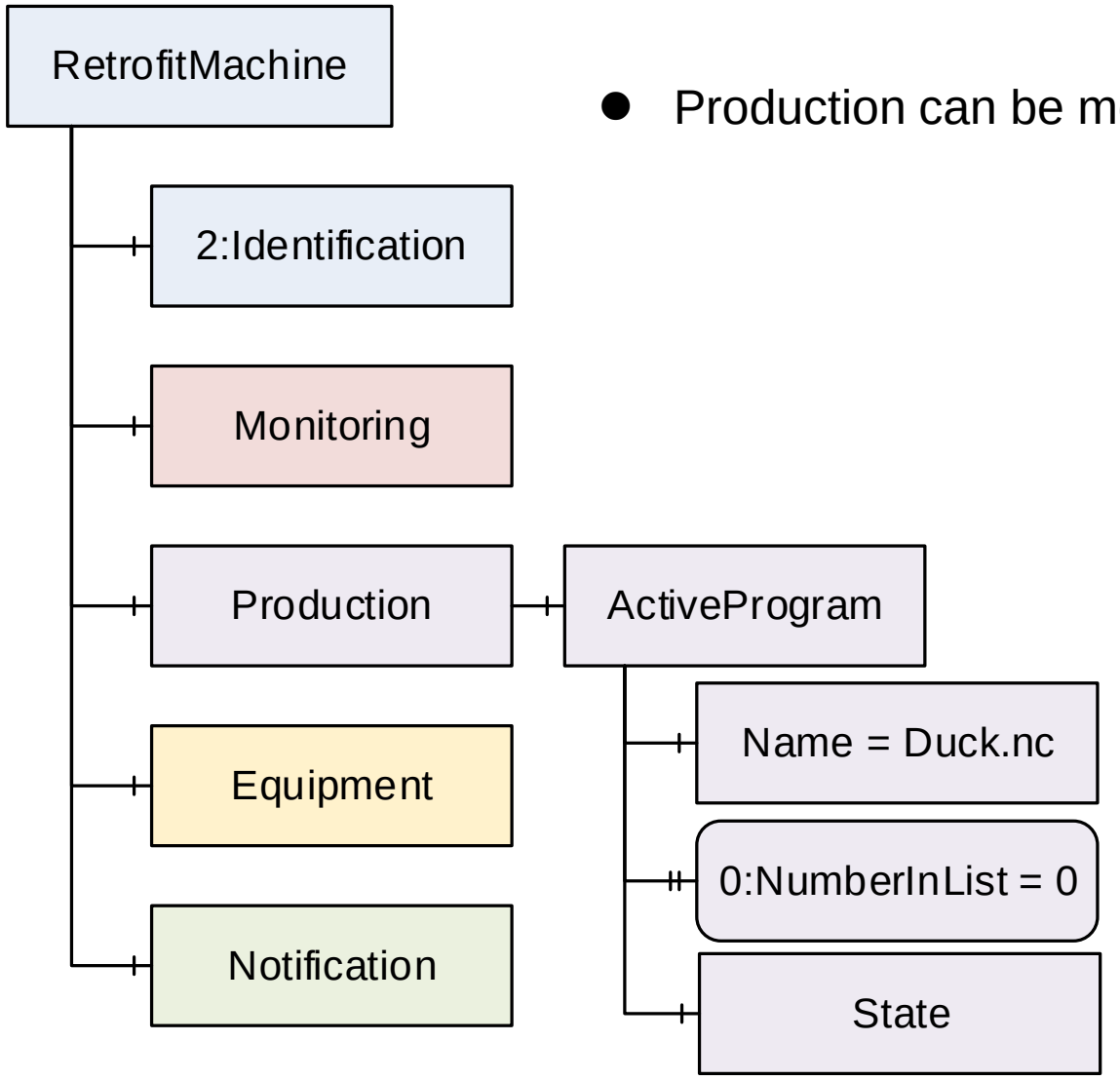
BrowseName	References	BrowseName	TypeDefinition
Transitions			
AbortedToInitializing	0:FromState	Aborted	StateType
	0:ToState	Initializing	InitialStateType
	0:HasEffect	ProductionJobTransitionEventType	*Event*
EndedToInitializing	0:FromState	Ended	StateType
	0:ToState	Initializing	InitialStateType
	0:HasEffect	ProductionJobTransitionEventType	*Event*
InitializingToAborted	0:FromState	Initializing	InitialStateType
	0:ToState	Aborted	StateType
	0:HasEffect	ProductionJobTransitionEventType	*Event*
InitializingToRunning	0:FromState	Initializing	InitialStateType
	0:ToState	Running	StateType
	0:HasEffect	ProductionJobTransitionEventType	*Event*

Machine Tool unaware of jobs

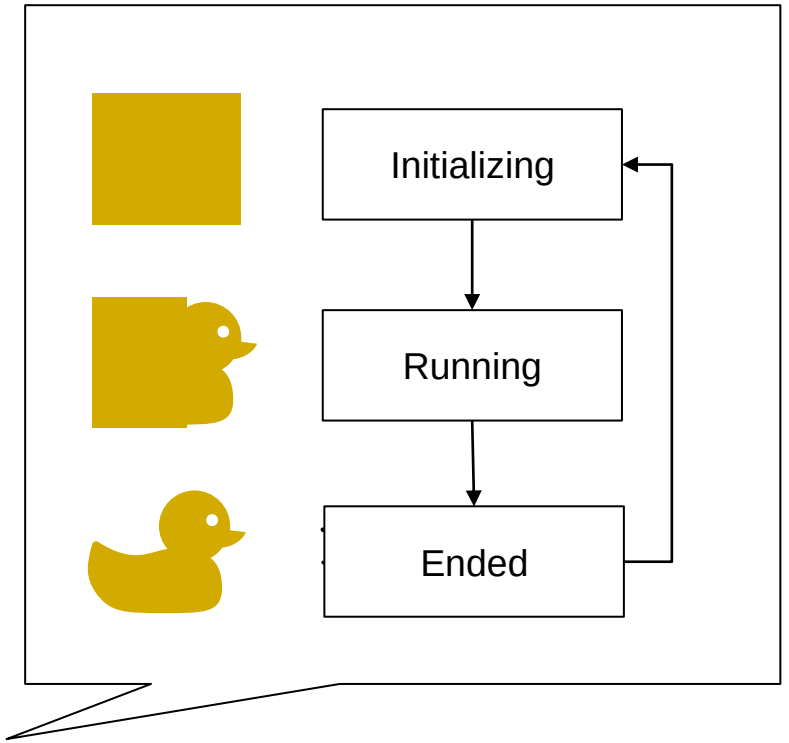


- Machine Tool only handles program files
- optional ProductionPlan node is not instantiated
- optional Components of ActiveProgram are not instantiated (*JobNodeId* and *JobIdentifier*)
- Programs can be identified by Name

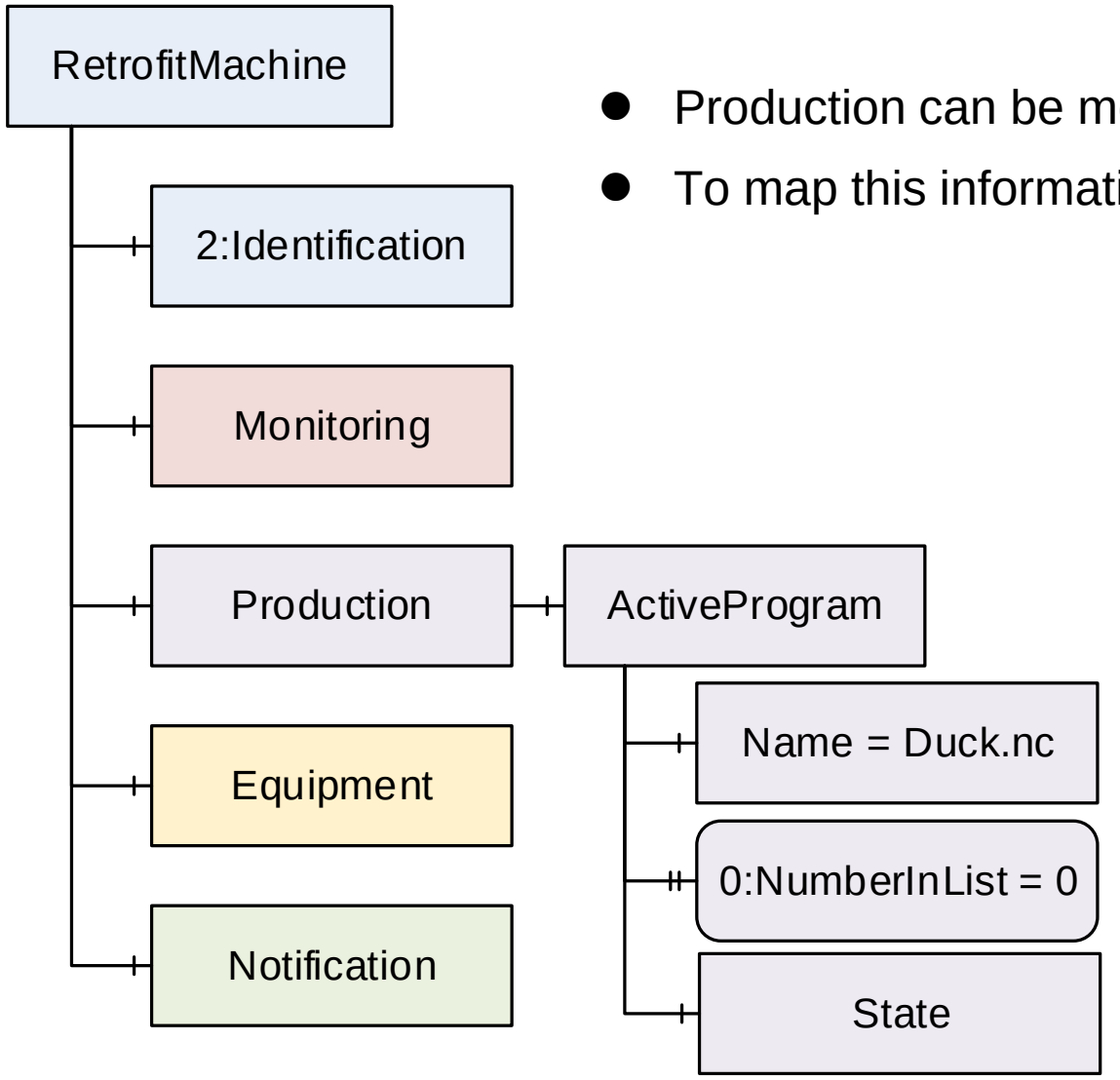
Machine Tool unaware of jobs



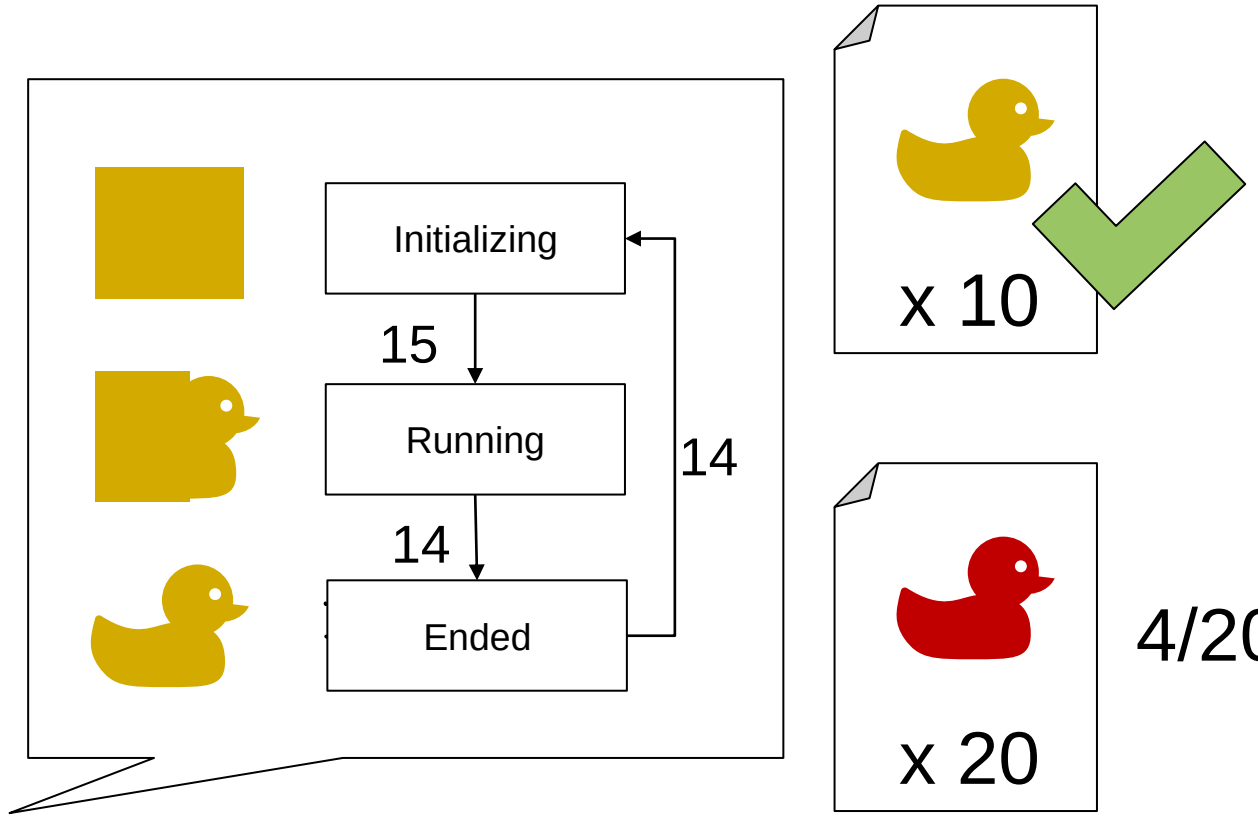
- Production can be monitored via the State of the ActiveProgram



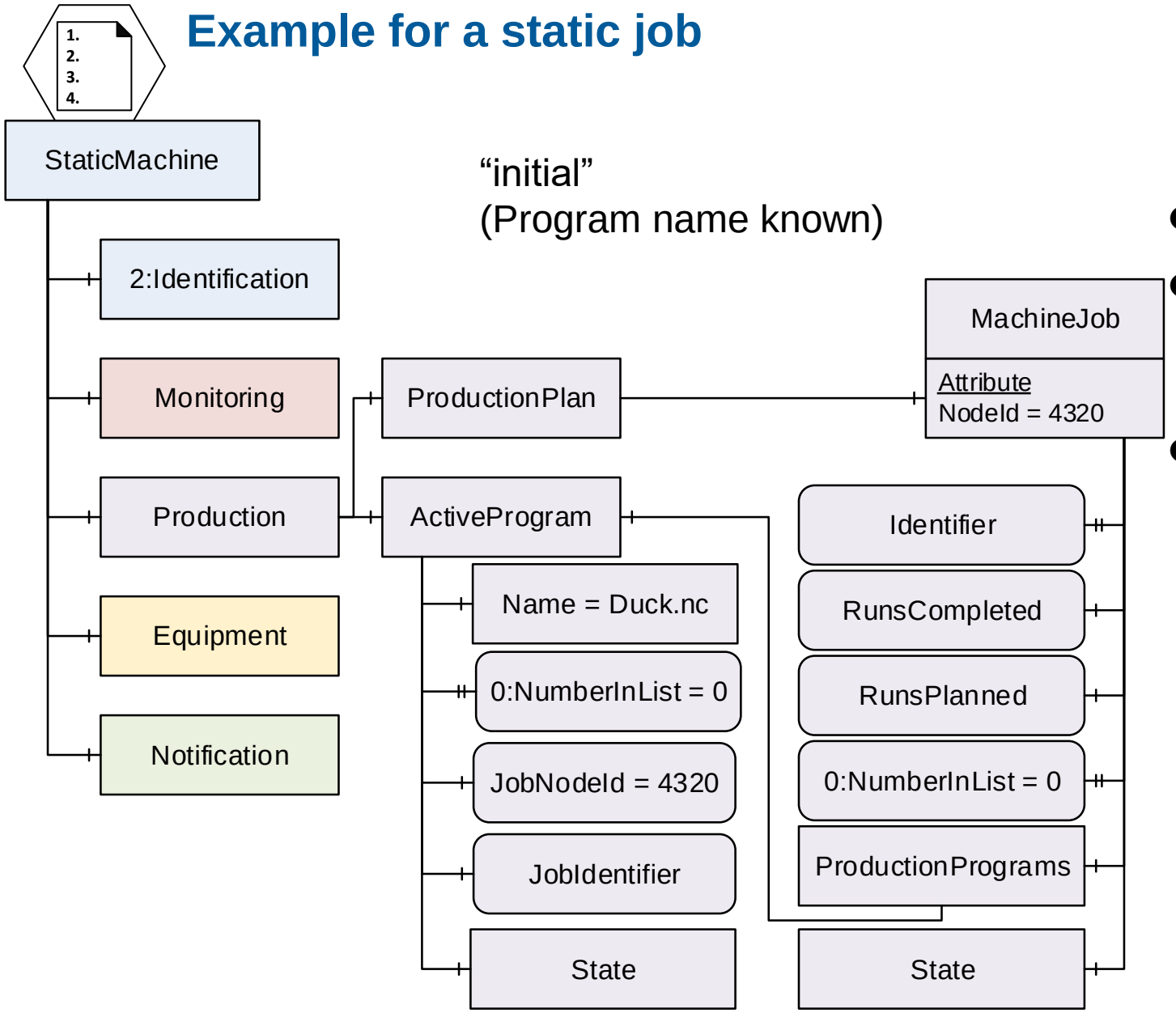
Machine Tool unaware of jobs



- Production can be monitored via the State of the ActiveProgram
- To map this information to the jobs, the state changes are monitored

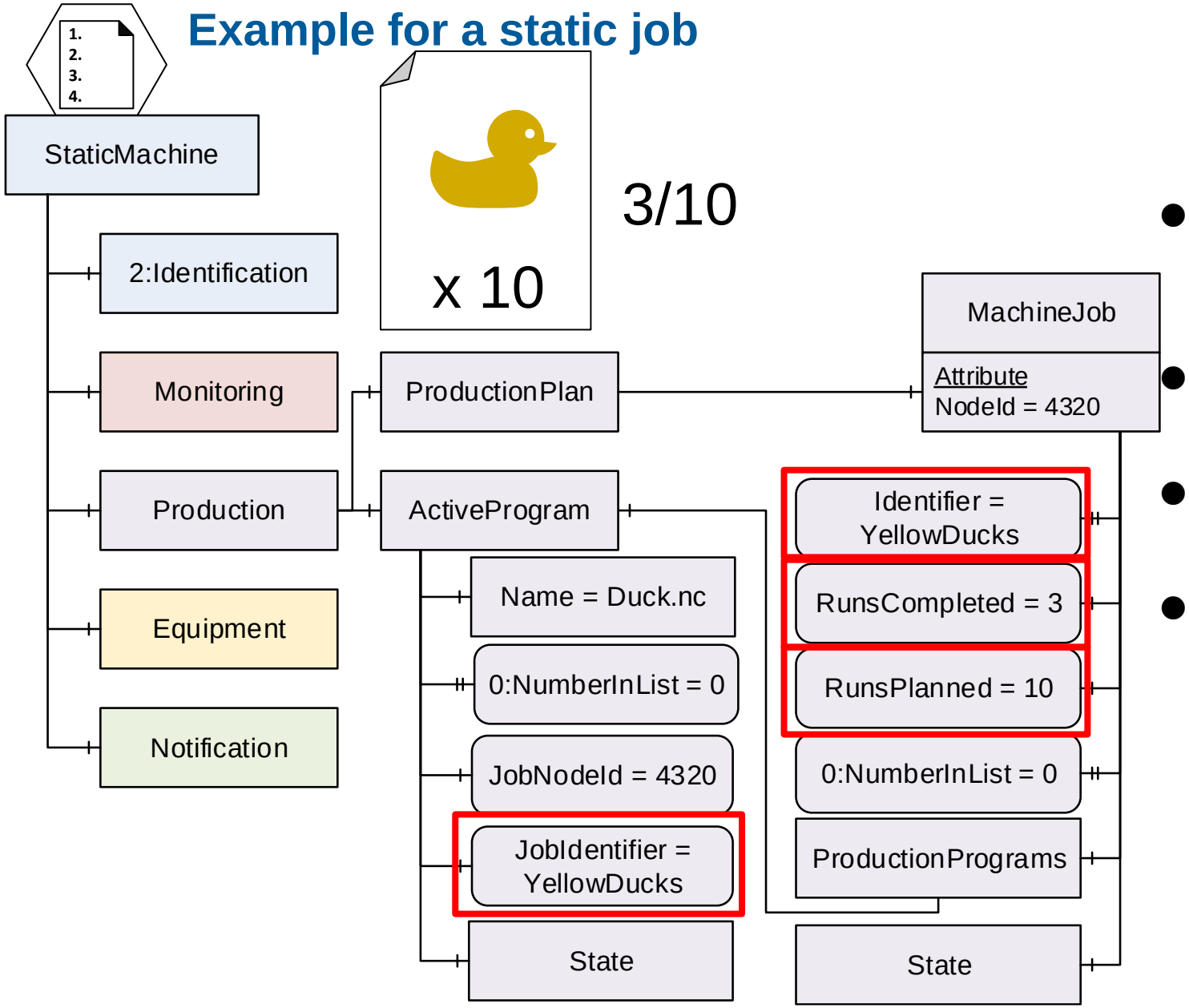


Example for a static job



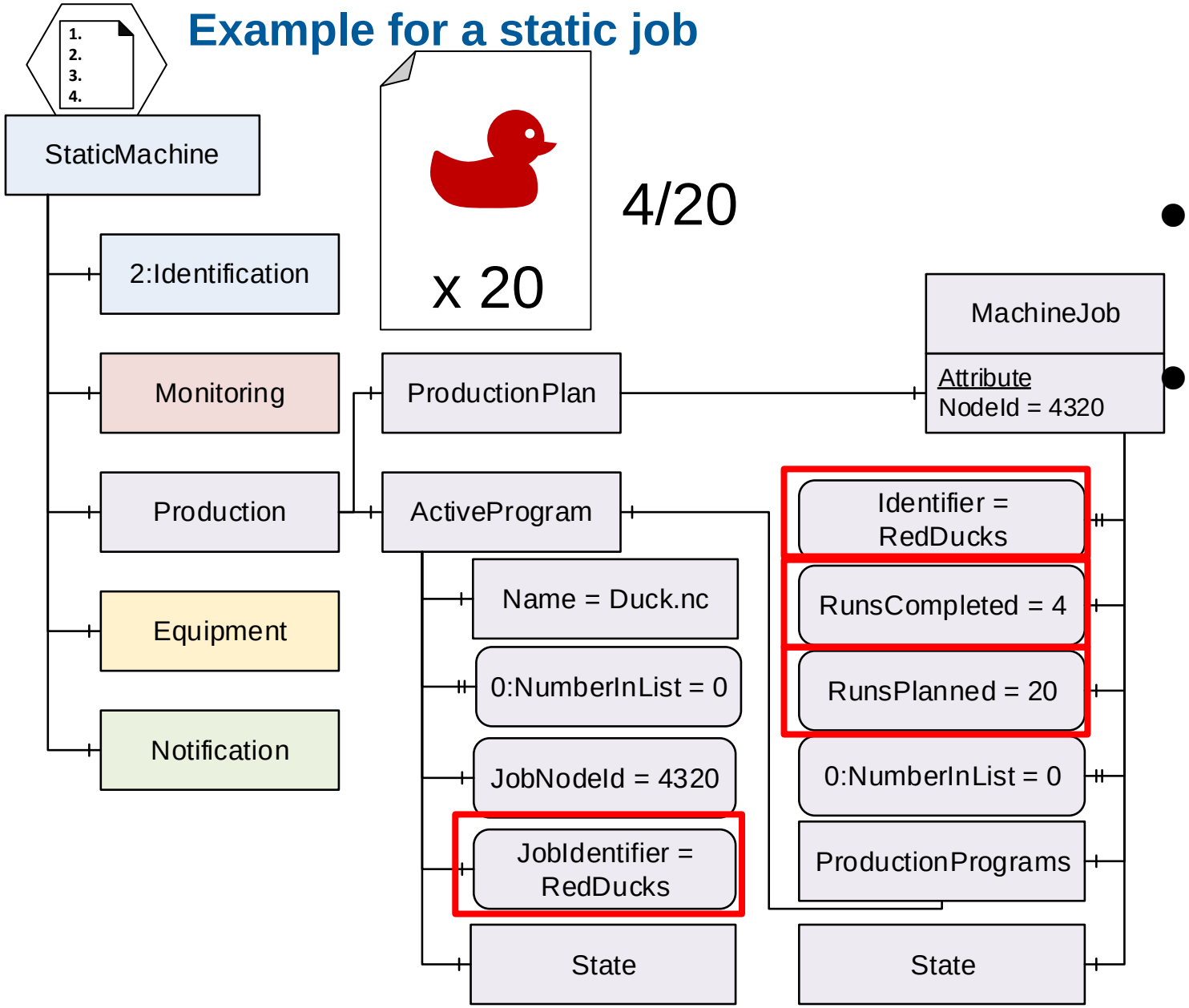
- ProductionPlan exists
- ProductionPlan contains fixed number of Jobs (Instances of ProductionJobType); example: 1
- ProductionPrograms contains fixed number of Programs (Instances of ProductionProgramType)
  - in the image: ActiveProgram is used
  - this is technically possible, probably impractical in most cases
  - Other Possibility: all values of ActiveProgram have the same content as one of the ProductionPrograms

Example for a static job



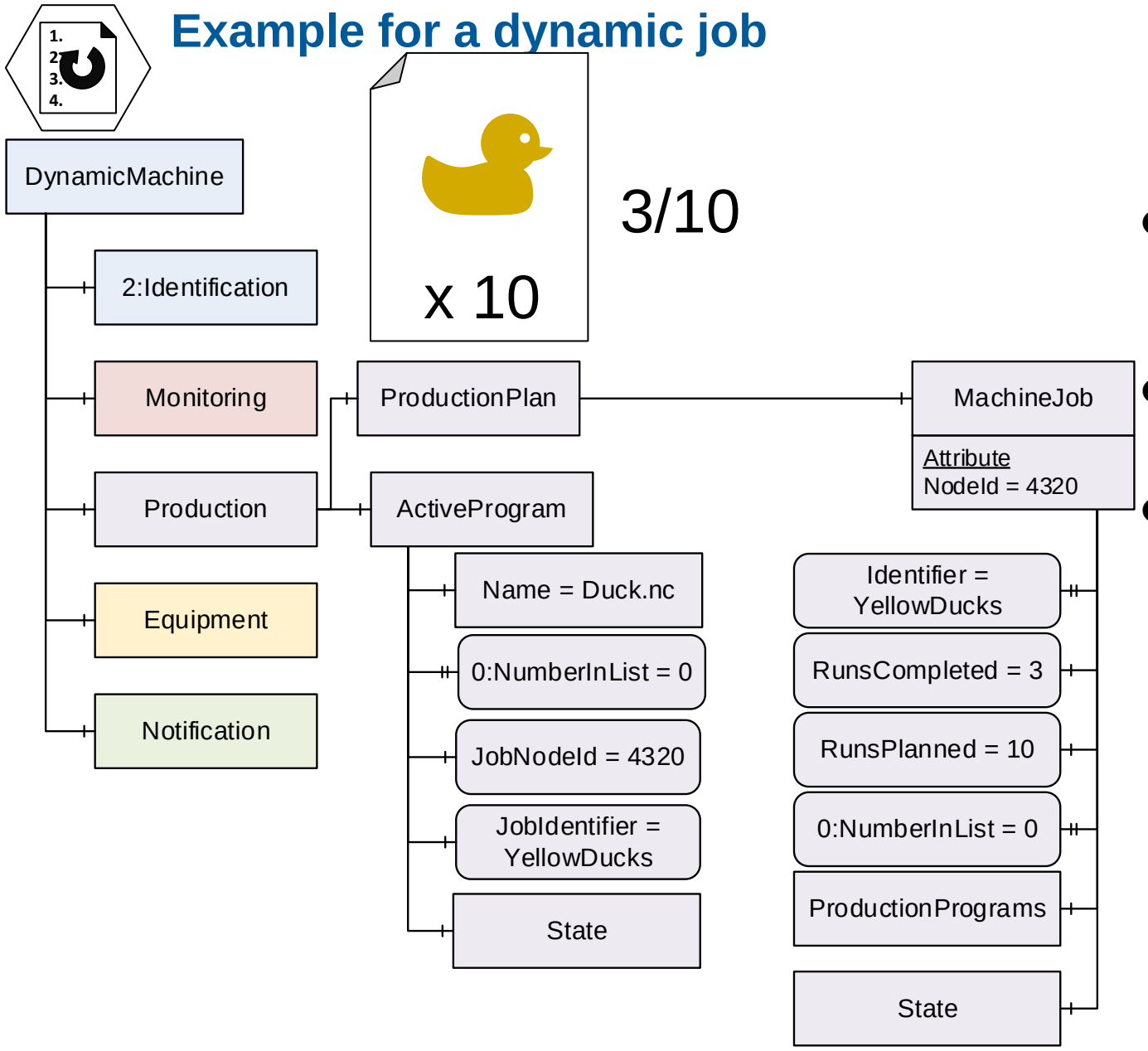
- If a job is represented, the Identifier, RunsCompleted, RunsPlanned and State show the appropriate information
- the JobIdentifier of the ActiveProgram is set
- only one job in Example - JobNodeId in ActiveProgram doesn't change
- RunsCompleted -> state changes don't have to be counted

Example for a static job



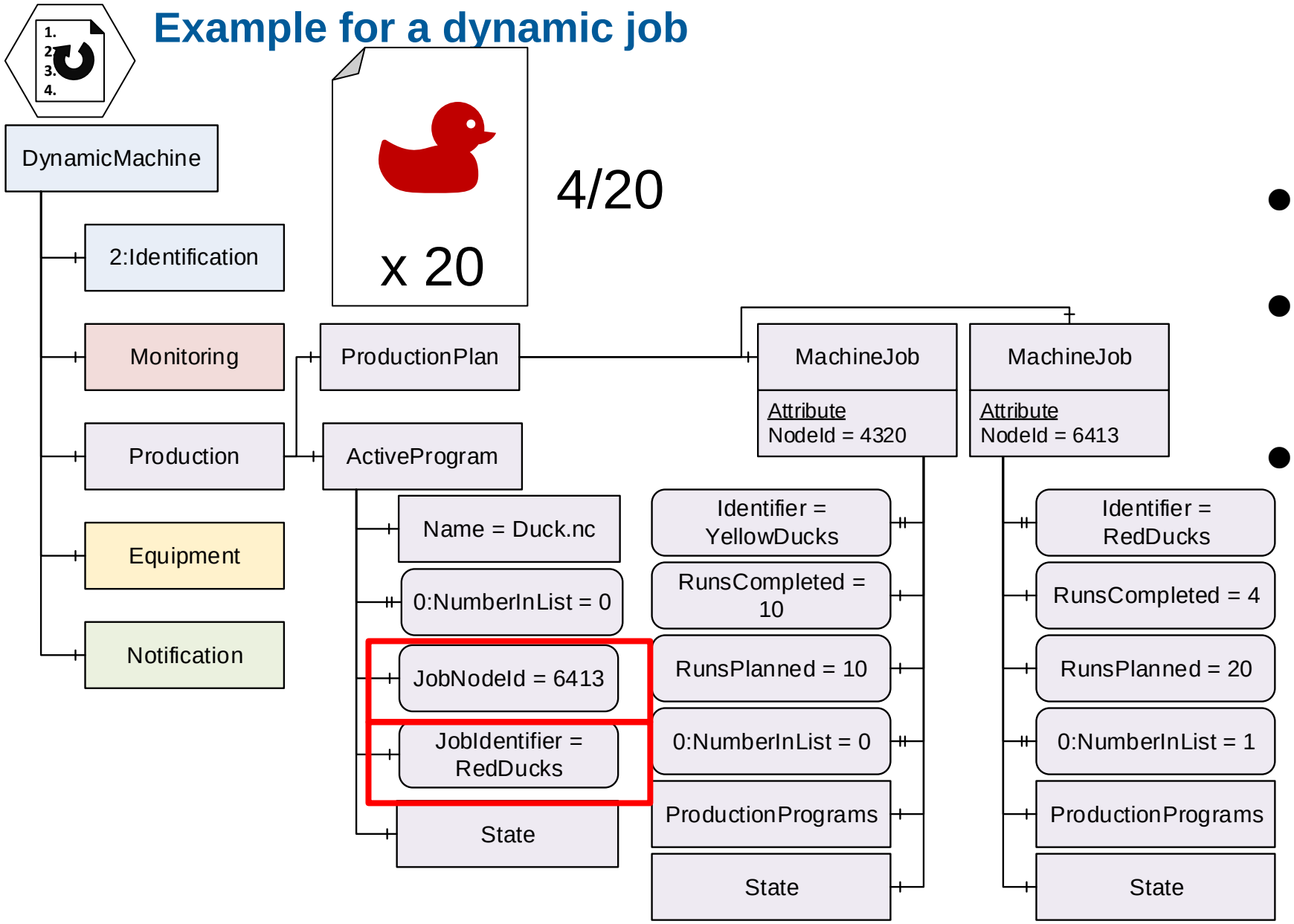
- Different job - Identifier, RunsCompleted, RunsPlanned and State show new information
- JobIdentifier in ActiveProgram shows new information

Example for a dynamic job



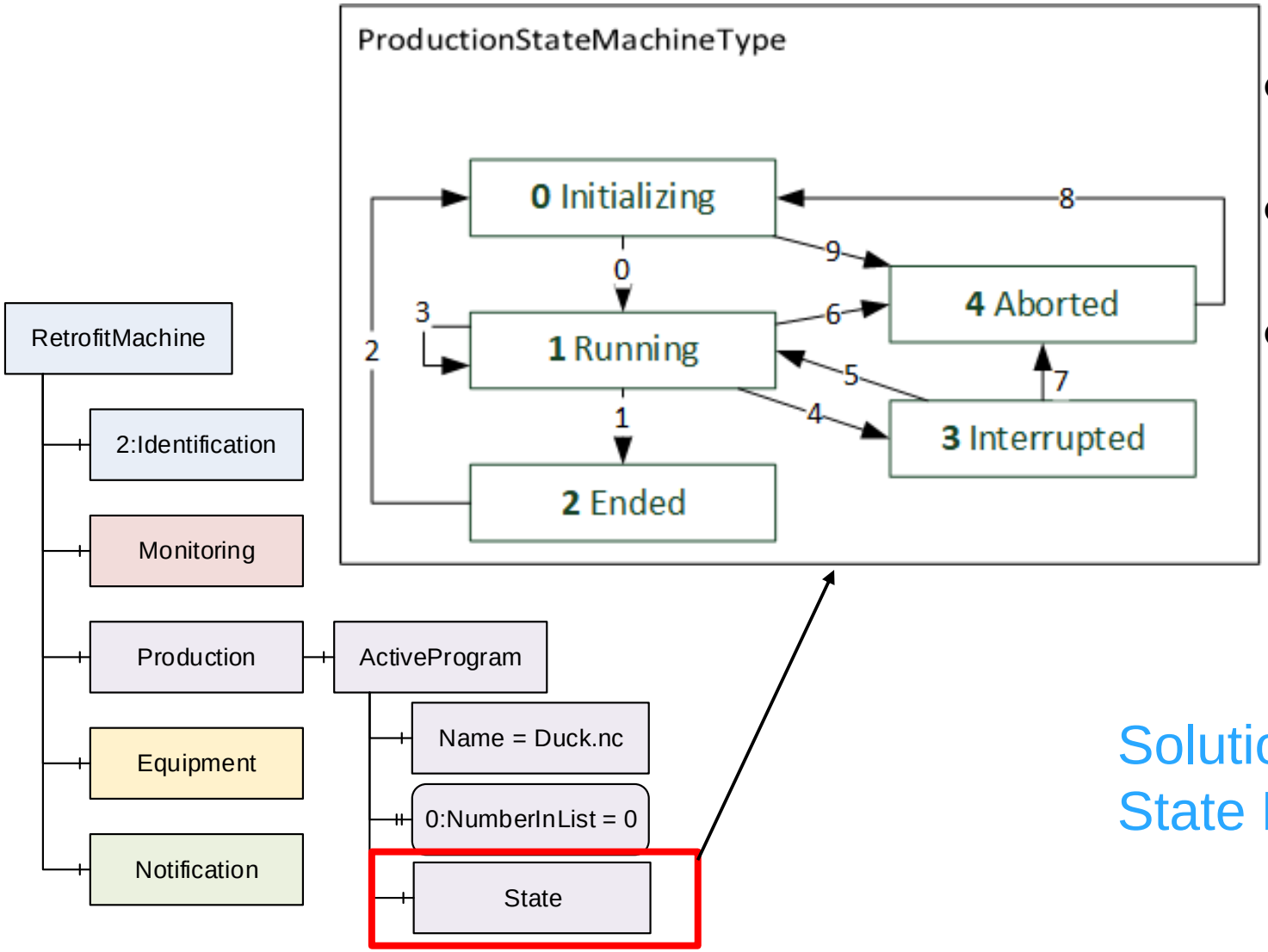
- If a job is represented, the Identifier, RunsCompleted, RunsPlanned and State show the appropriate information
- the JobIdentifier of the ActiveProgram is set
- RunsCompleted -> state changes don't have to be counted

Example for a dynamic job



- New Job is created as new Node
- JobIdentifier in ActiveProgram shows new information
- JobNodeId in ActiveProgram shows updated information

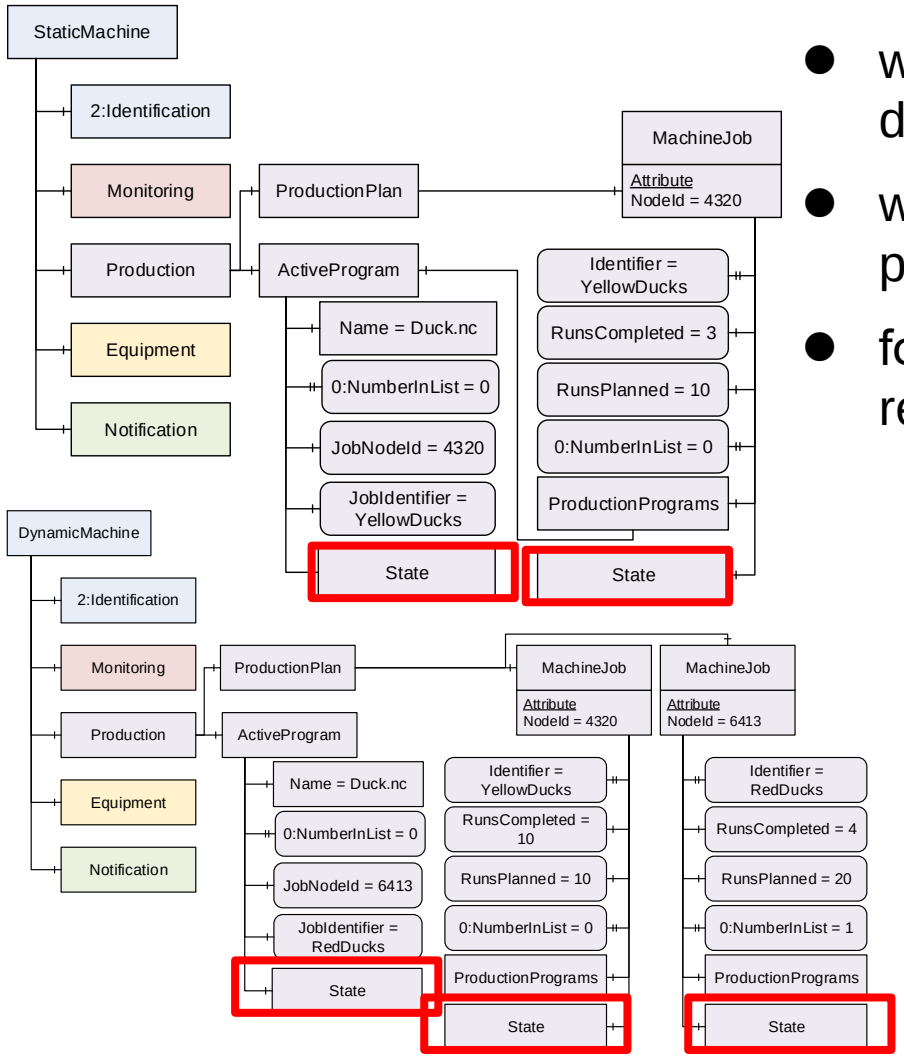
Usage with Data Access



- Every State node instantiates ProductionStateMachineType
- Read/Subscription as simple method to access the state information
- shortcoming: fast-paced jobs

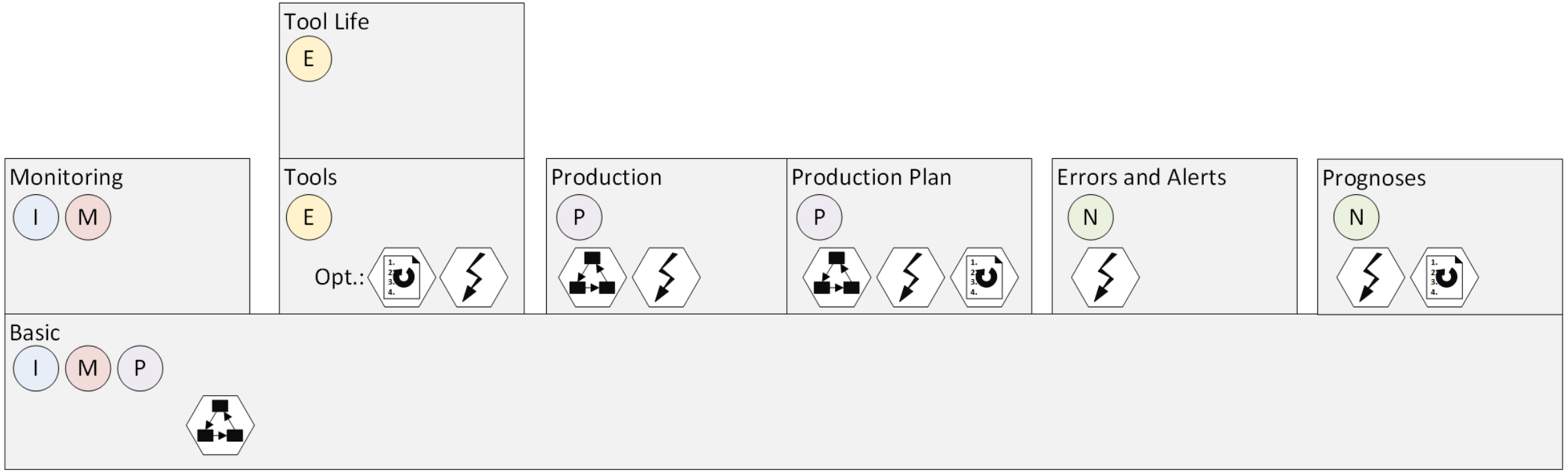
Solution: Events along with the State Machine Transitions

Using Events



- works in all three presented configurations (Program only, static jobs, dynamic jobs)
- with SourceTimestamp of every event, the client can reconstruct fast-paced jobs
- for calculations like KPI, it doesn't matter when the server sends/client receives the data as long as all data is available in the end

Facets



- I Identification
- M Monitoring
- P Production
- N Notification
- E Equipment

State Machine

OPC UA Events

Dynamic Lists

Thank you
for your attention!

For further information please contact: Goetz Goerisch g.goerisch@vdw.de