



UNIVERSITÉ DE NANTES



A Model Driven Approach for automated generation of Service-oriented Holonic Manufacturing Systems

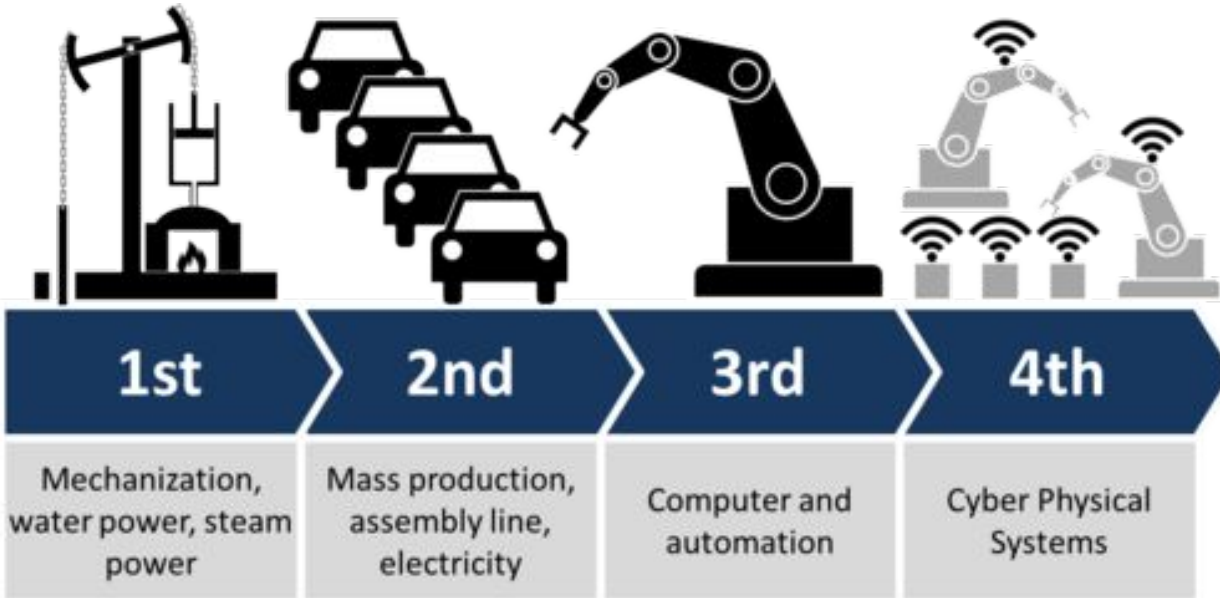
Mohammed El Amin TEBIB, Pascal André, Olivier Cardin
LS2N lab, University of Nantes, France

Outline

- **Context**
- **Problem Statement**
- **Proposed Approach**
- **Ongoing works**
- **Conclusion**

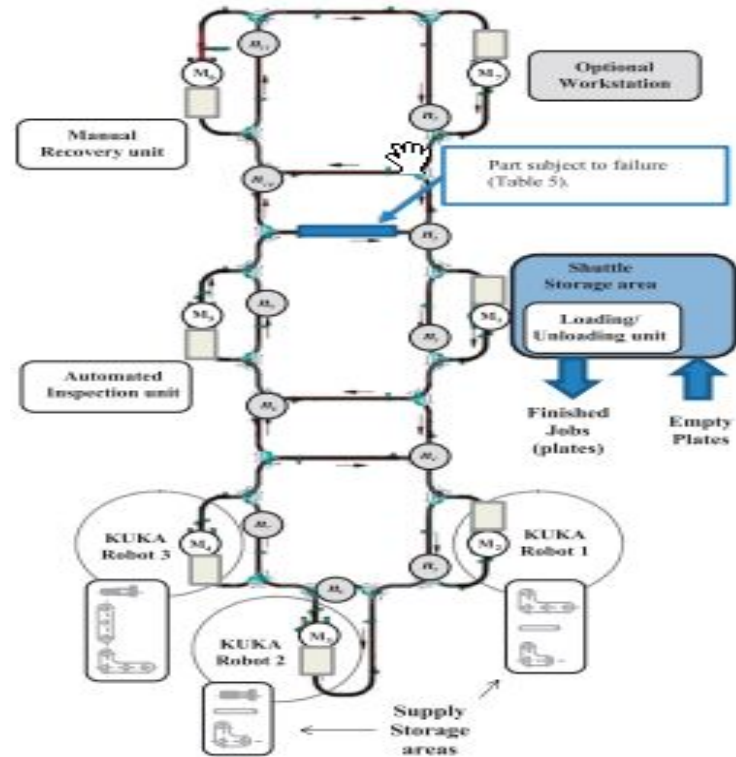
Context:

Manufacturing in industry 4.0



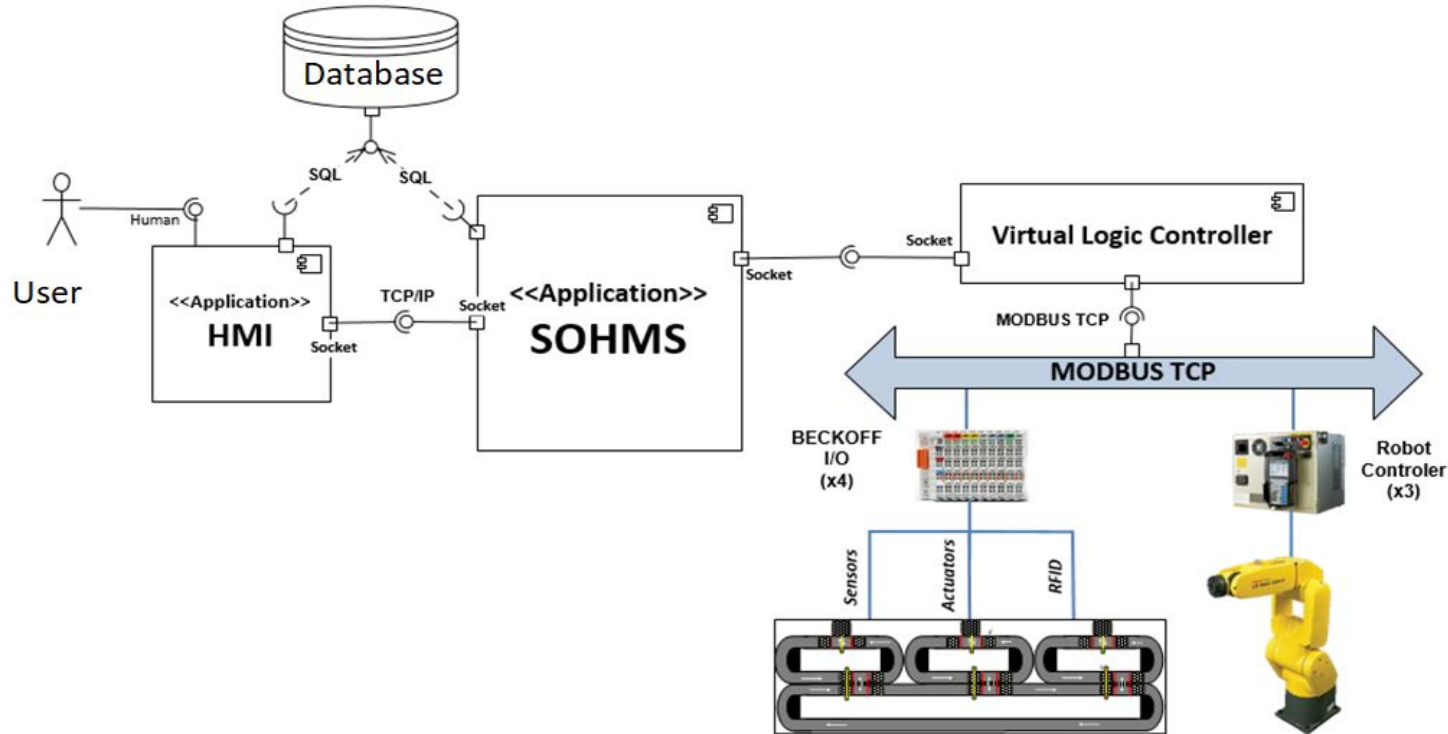
Context:

Manufacturing workshops



Context:

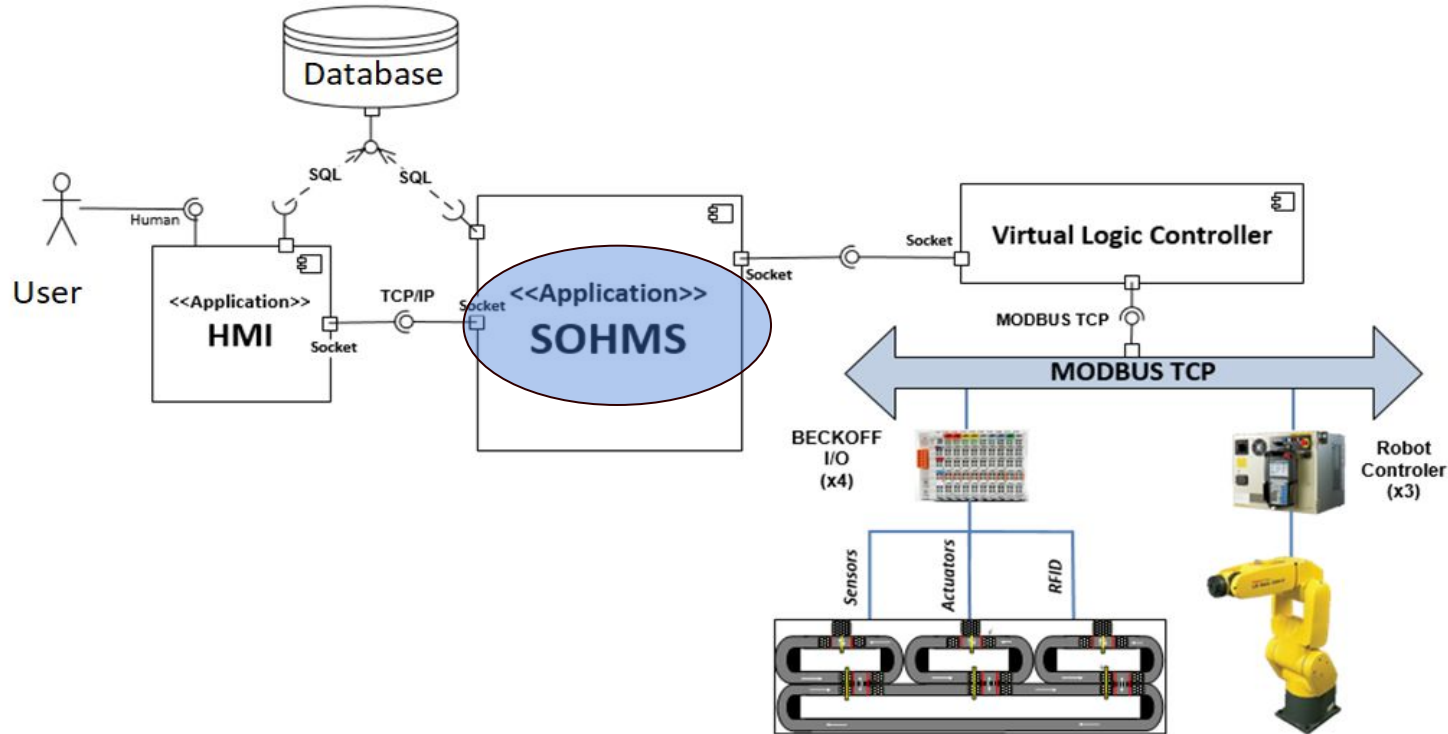
Manufacturing Systems Architectures



[Gamboa Quintanilla et al., 2016b]

Context:

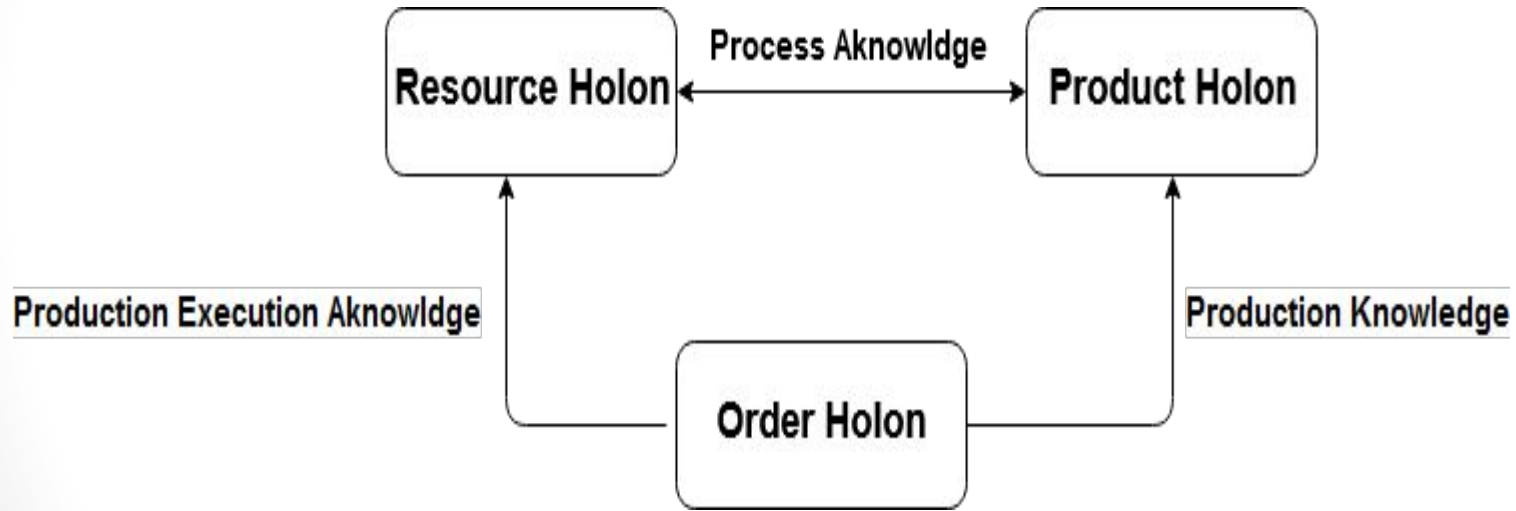
Manufacturing Systems Architectures



[Gamboa Quintanilla et al., 2016b]

Context:

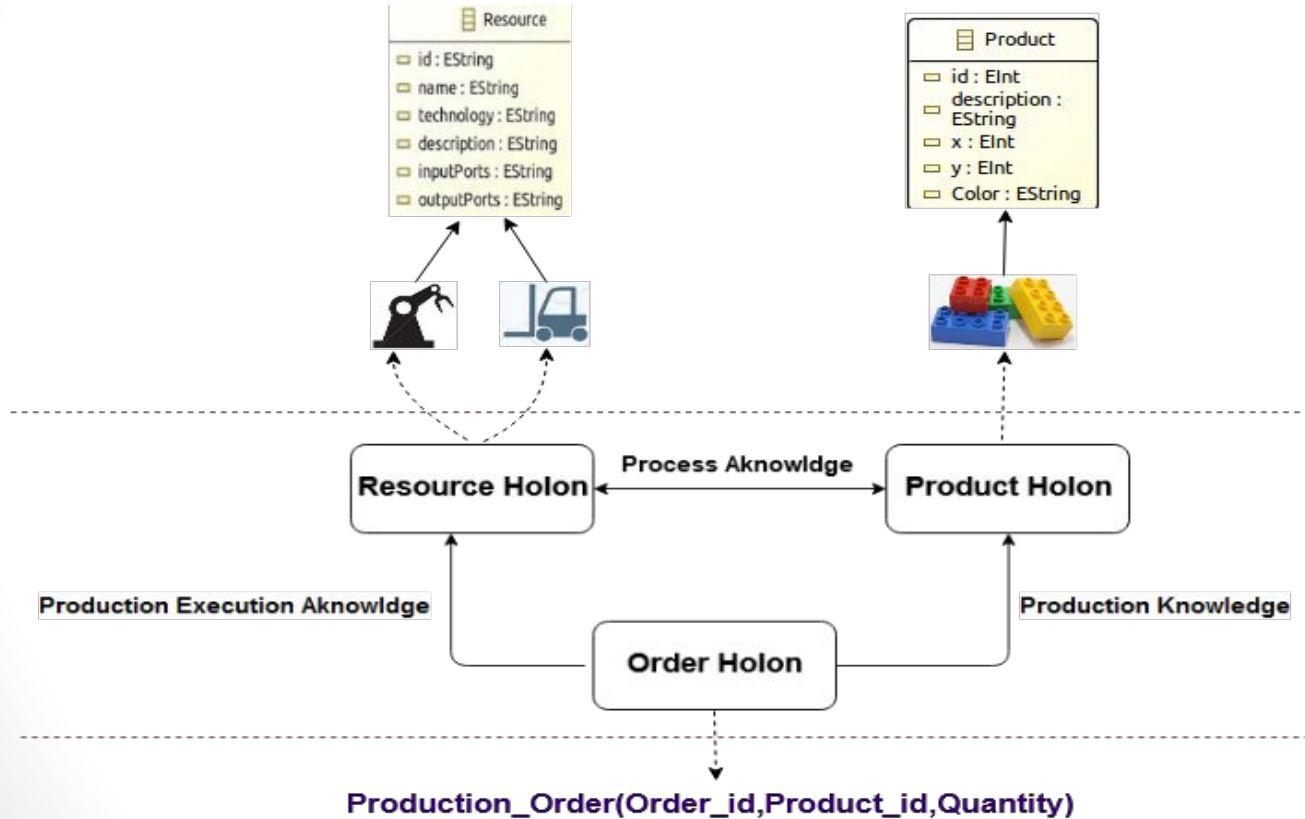
HMS Architecture



[PROSA Architecture]

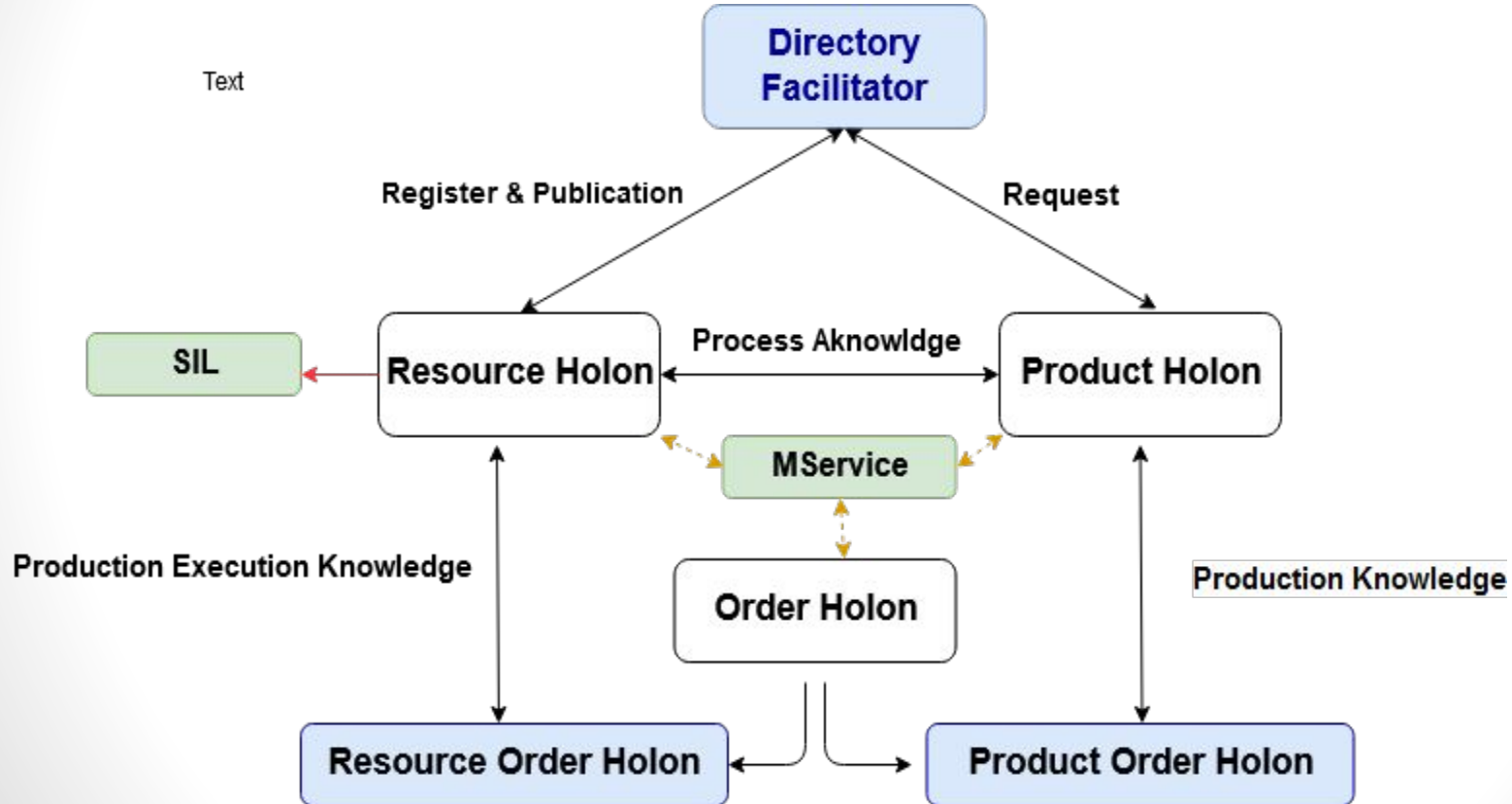
Context:

HMS Architecture



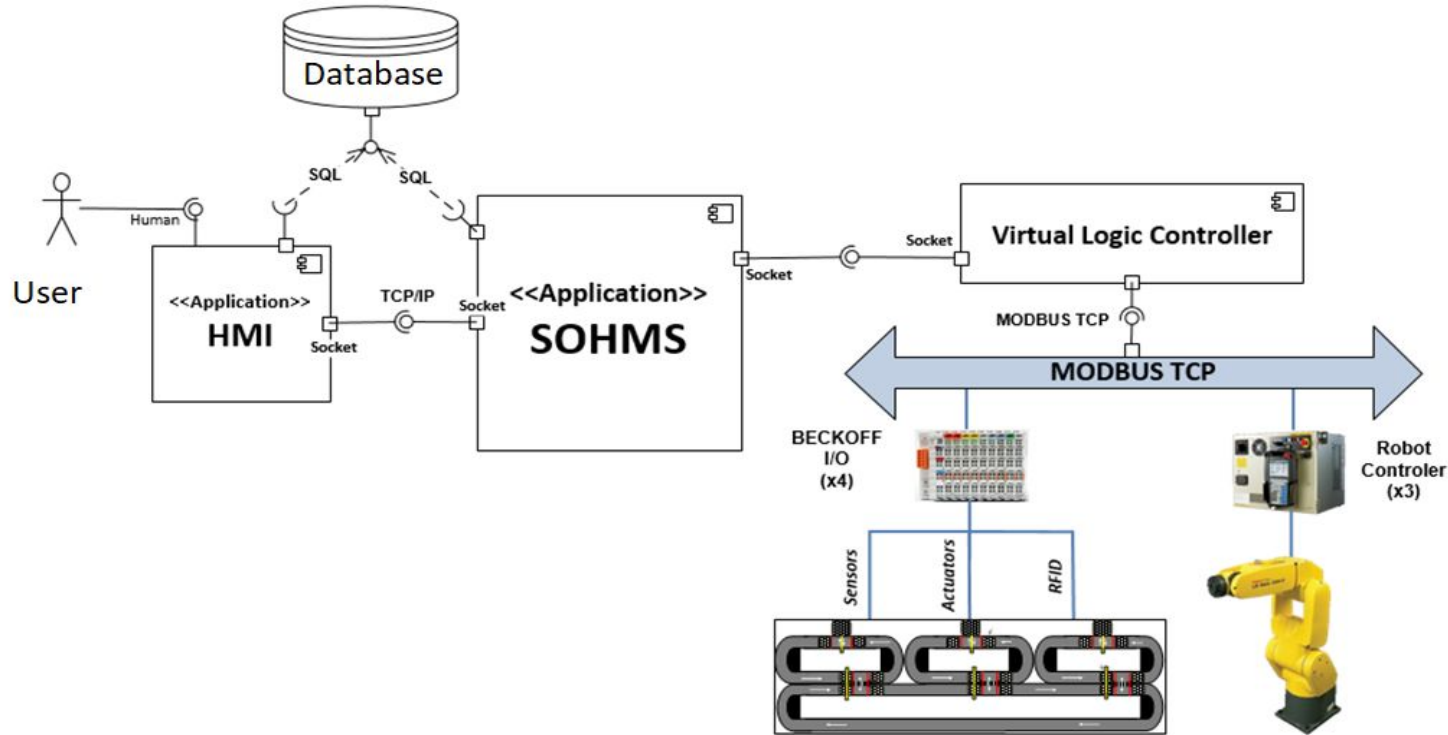
Context:

Service Oriented Holonic Manufacturing Systems



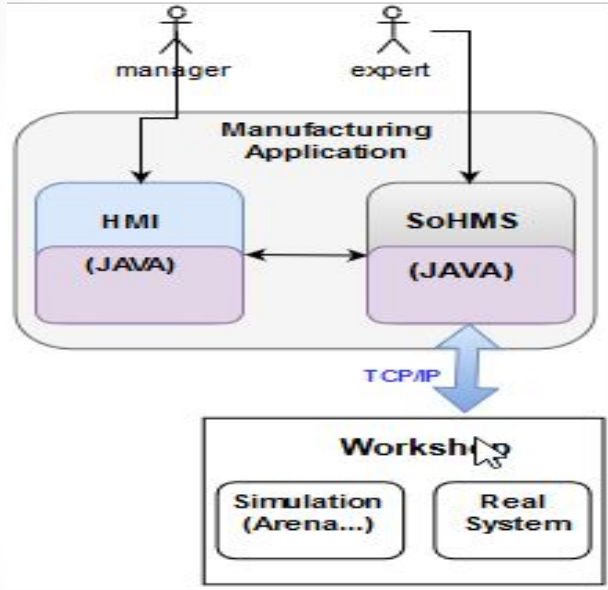
Context:

Manufacturing Systems Architectures

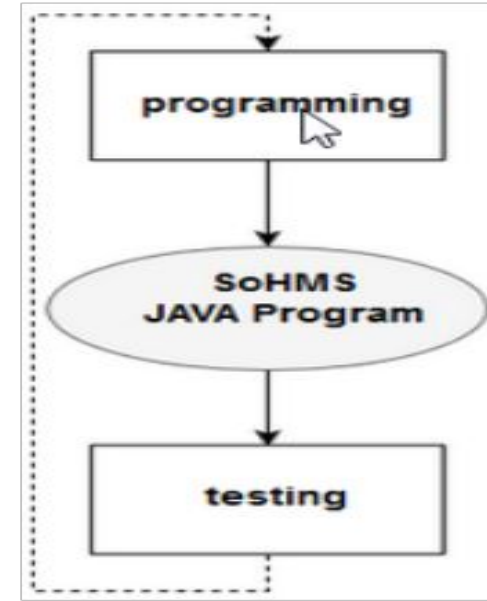


[Gamboa Quintanilla et al., 2016b]

Current system:

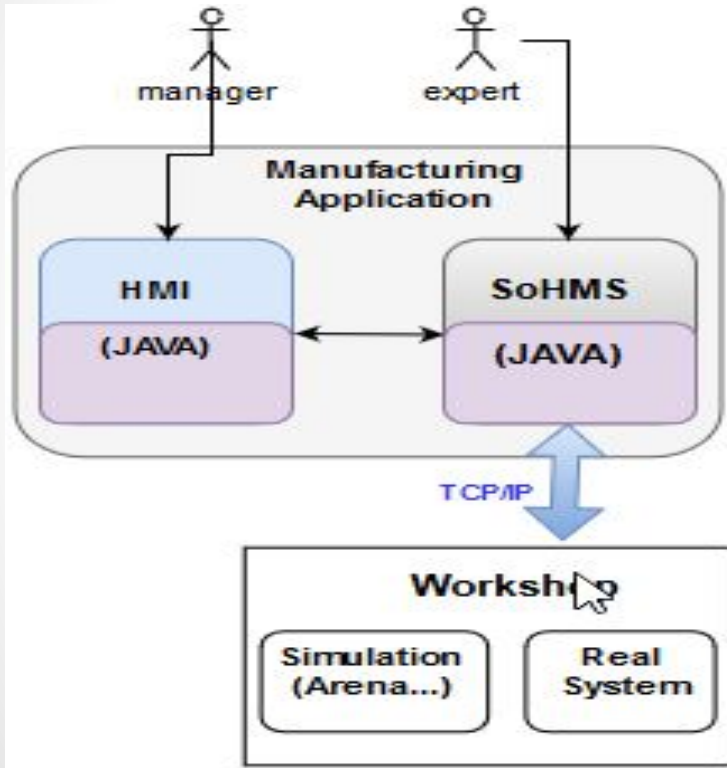


Application Architecture



Software Construction Process

Problem Statement: Application Architecture

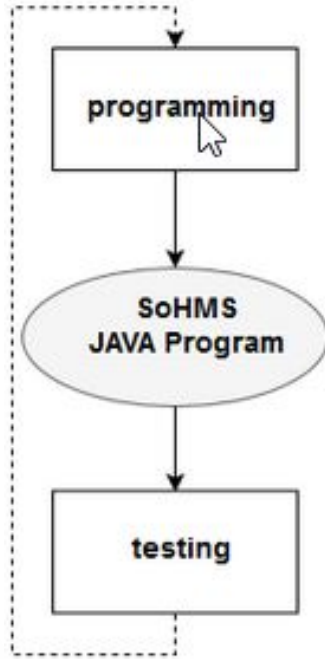


**160 classes,
1240 methods,
14802 lines of code**

- **No Modularity**
 - **Crosscutting concerns.**
 - **No variability**
 - **Specific Softwares**

Problem Statement (2):

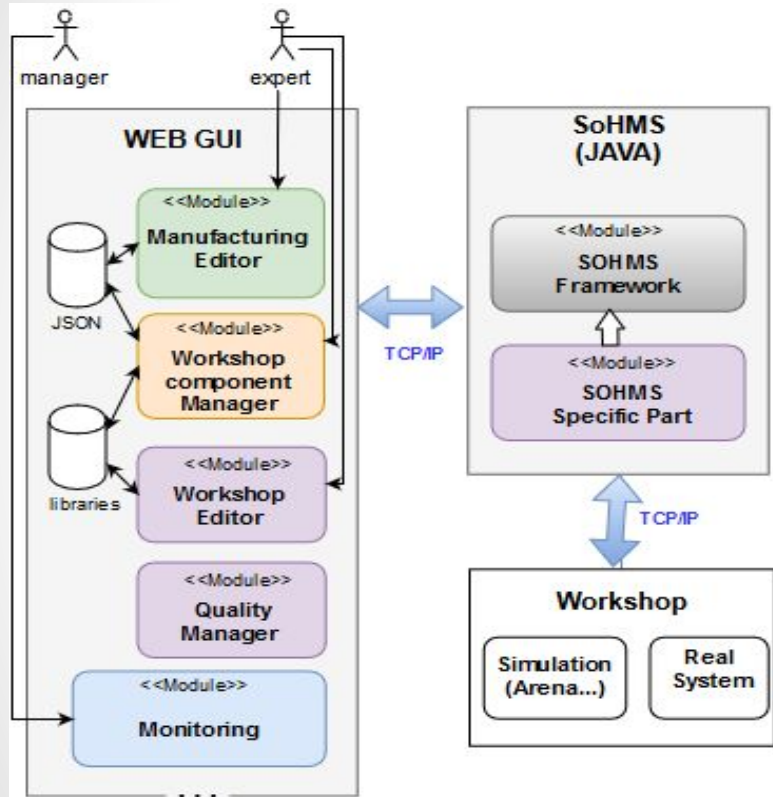
Software Construction Process



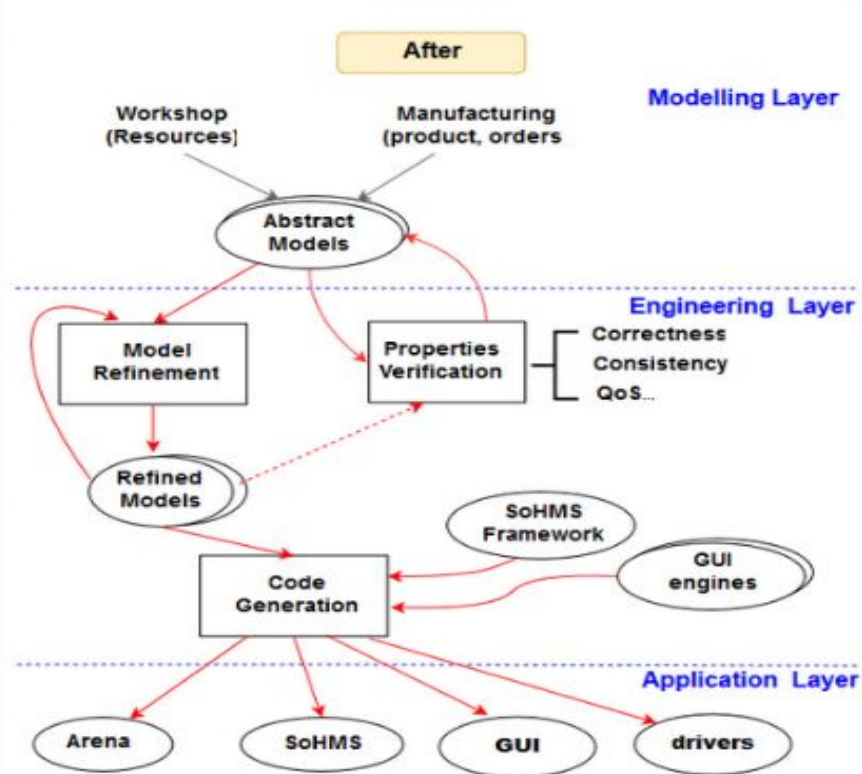
- **No Abstraction**
 - **Hard Reconfiguration**
 - **Hard Revolution**
 - **Hard Verification**
(Proving system properties at implementation level is too expensive)

Face these challenges ...

New Vision:

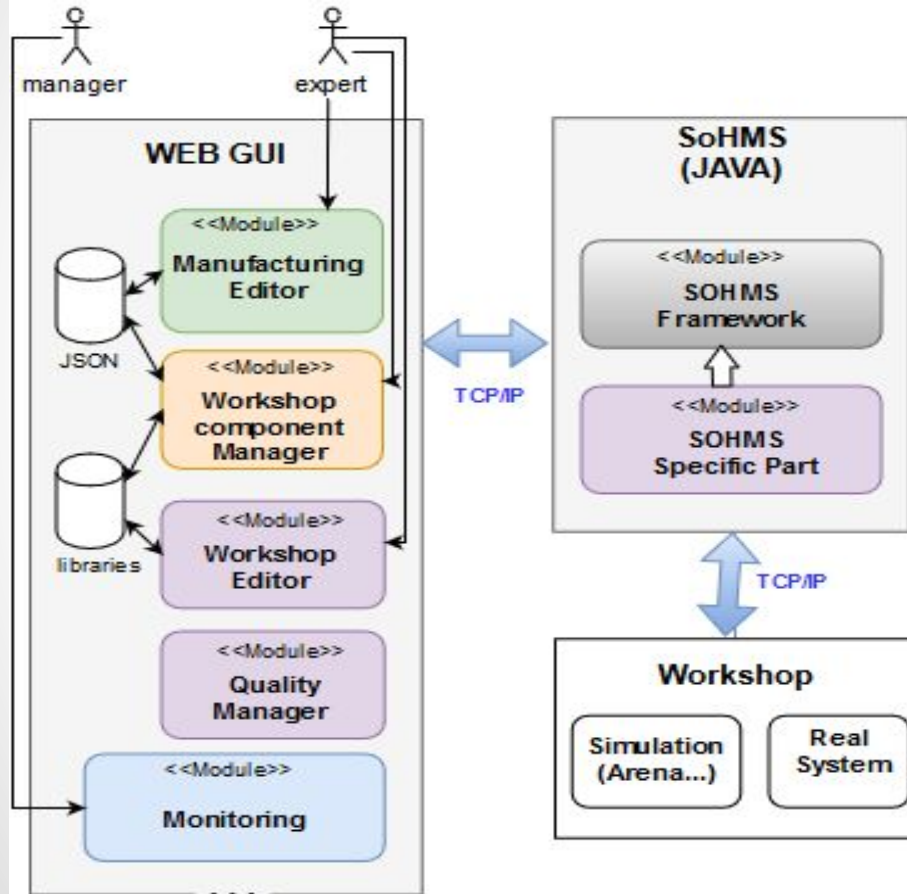


Application Architecture



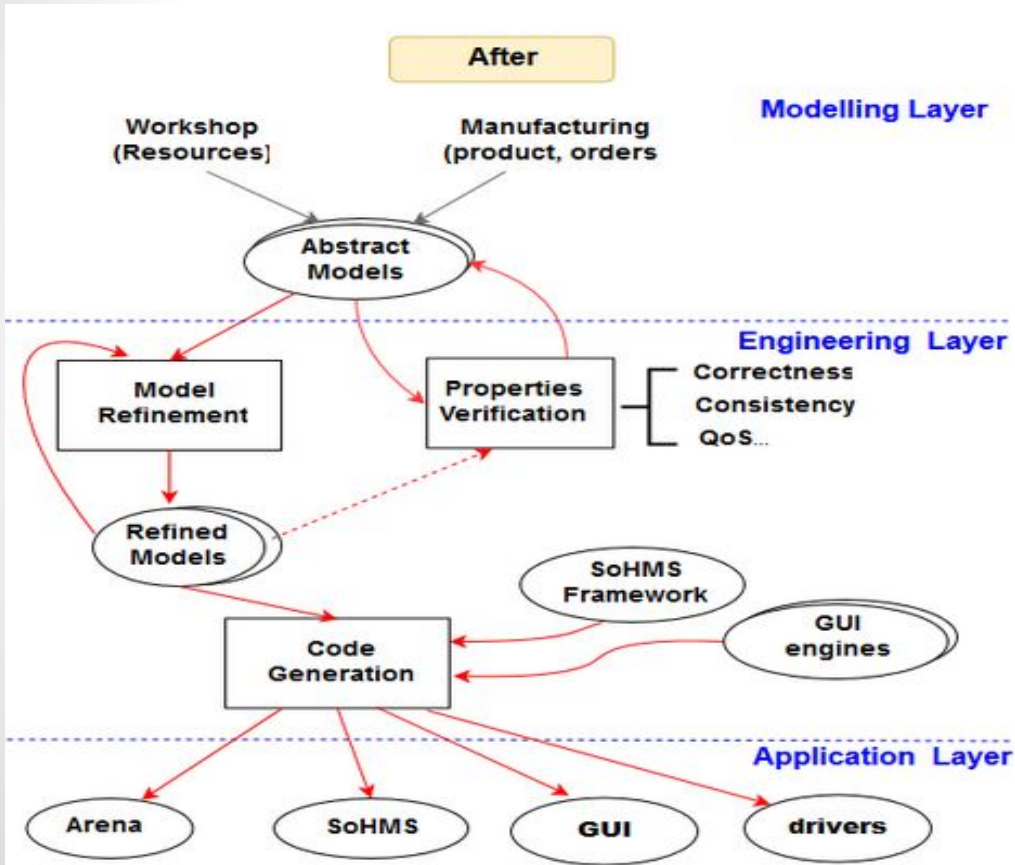
Software Construction Process

Proposed Approach: Application Architecture



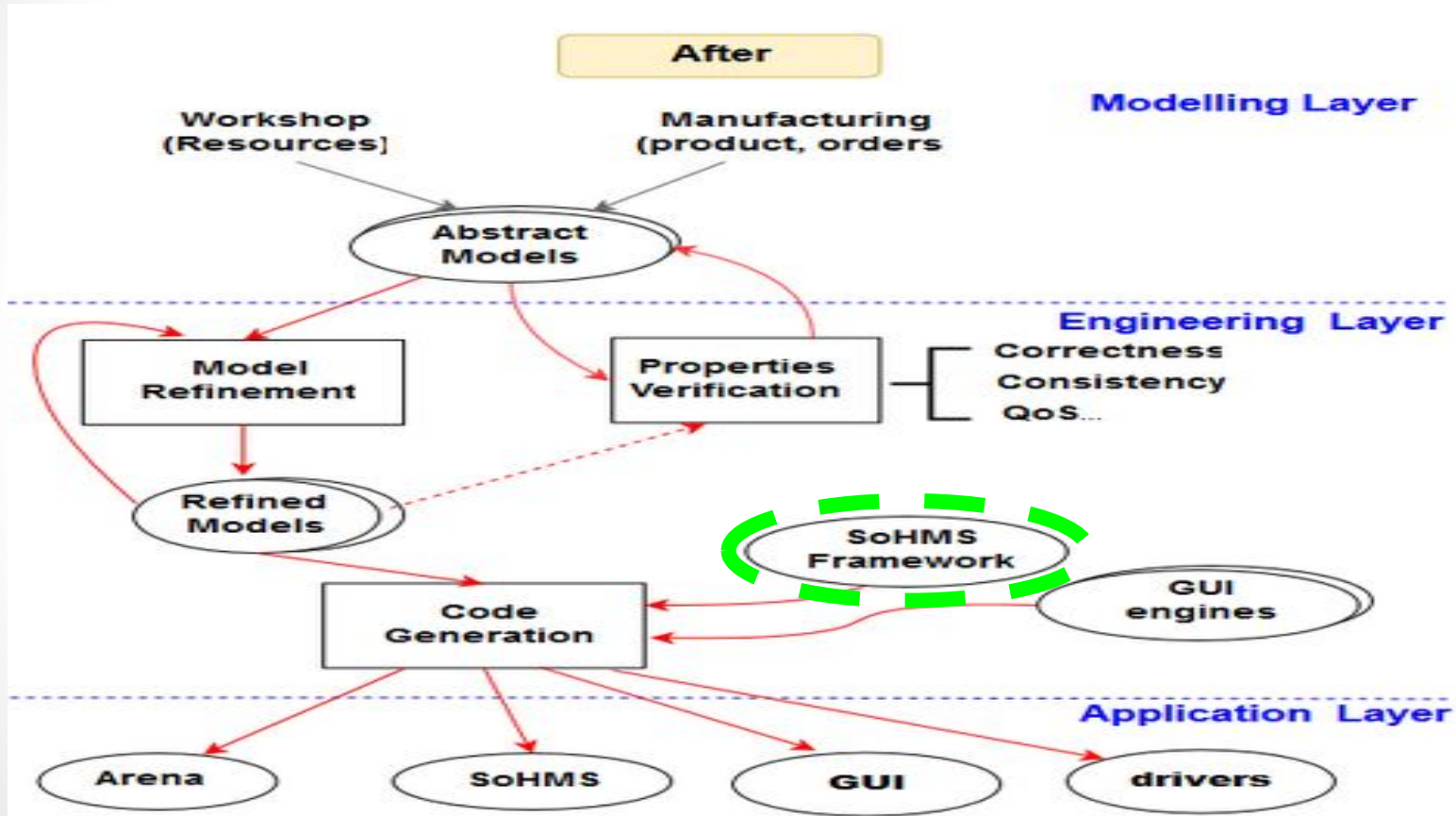
- **Architecture Modularity**
 - Three levels (Web, SoHMS, Workshop)
- **Extendibility & variability**
 - Separation of concerns (specific and generic parts)
 - Interoperability (Web technologies)

Proposed Approach: Software Construction Process



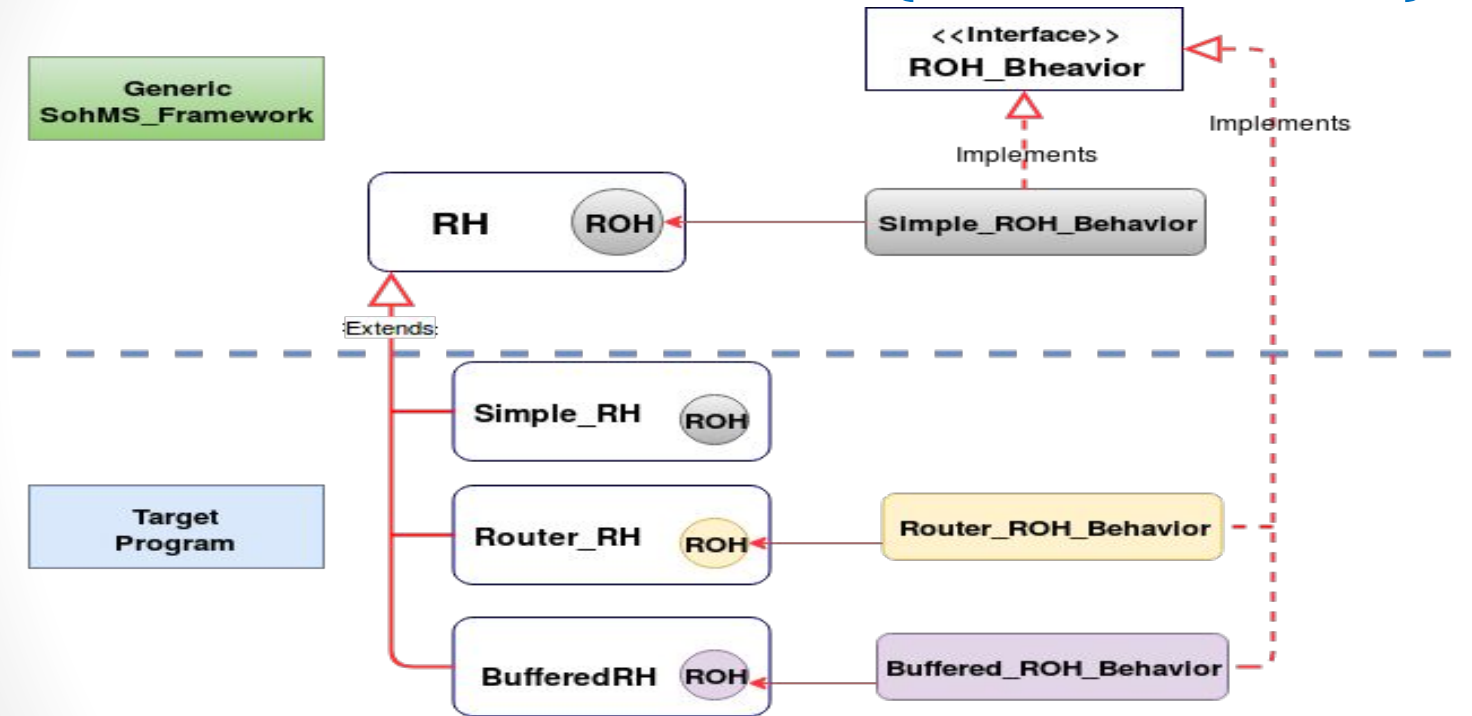
- **Modelling Layer**
 - Have an abstract models
 - Verification in First stages
 - Formal verification (High level of insurance)
- **Engineering Layer**
 - Automatic code generation for specific part
 - Generic Framework for the SoHMS Processing for Reuse.

New Vision:



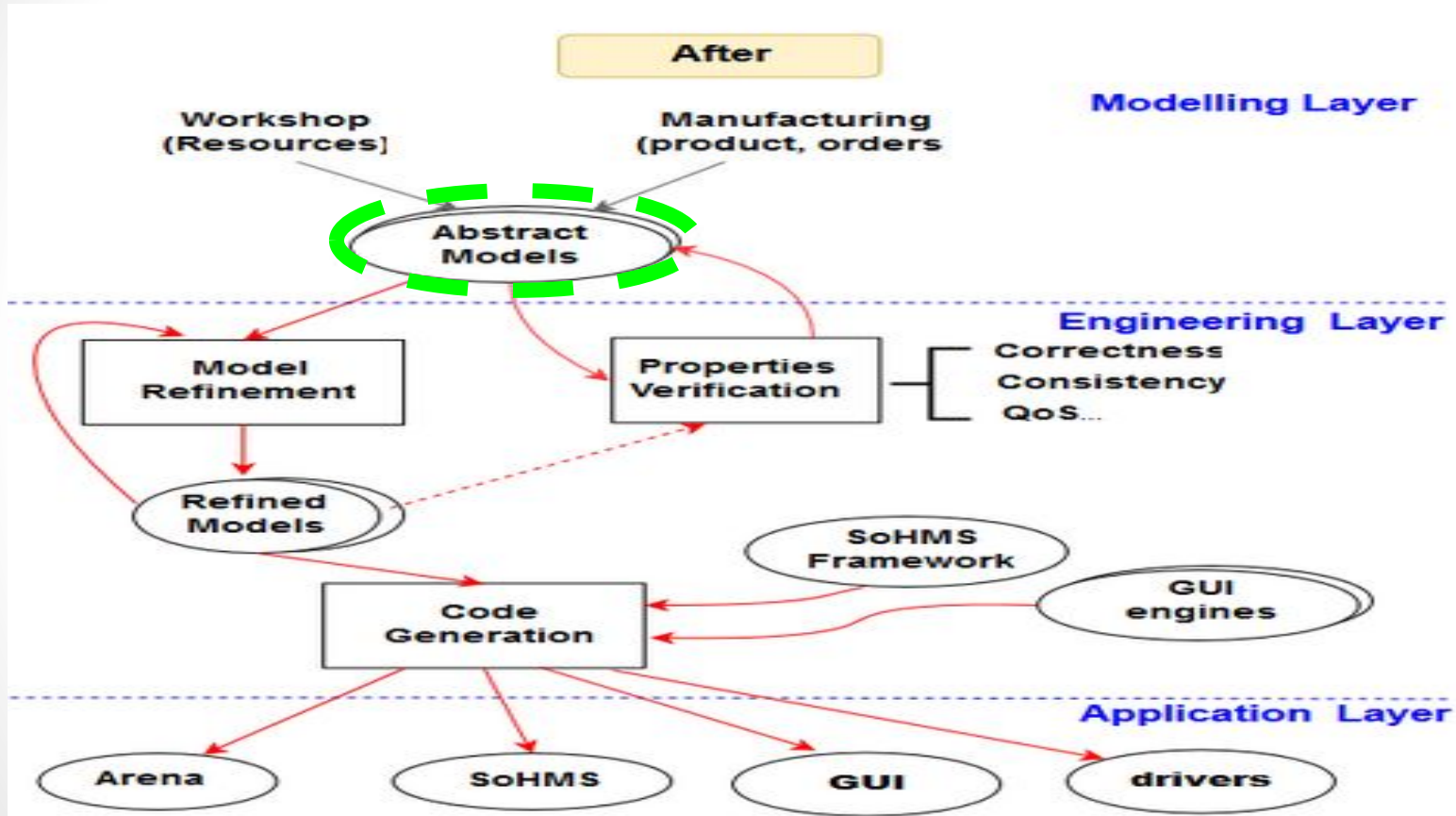
Ongoing Works:

Generic SoHMS Framework (More details ...)



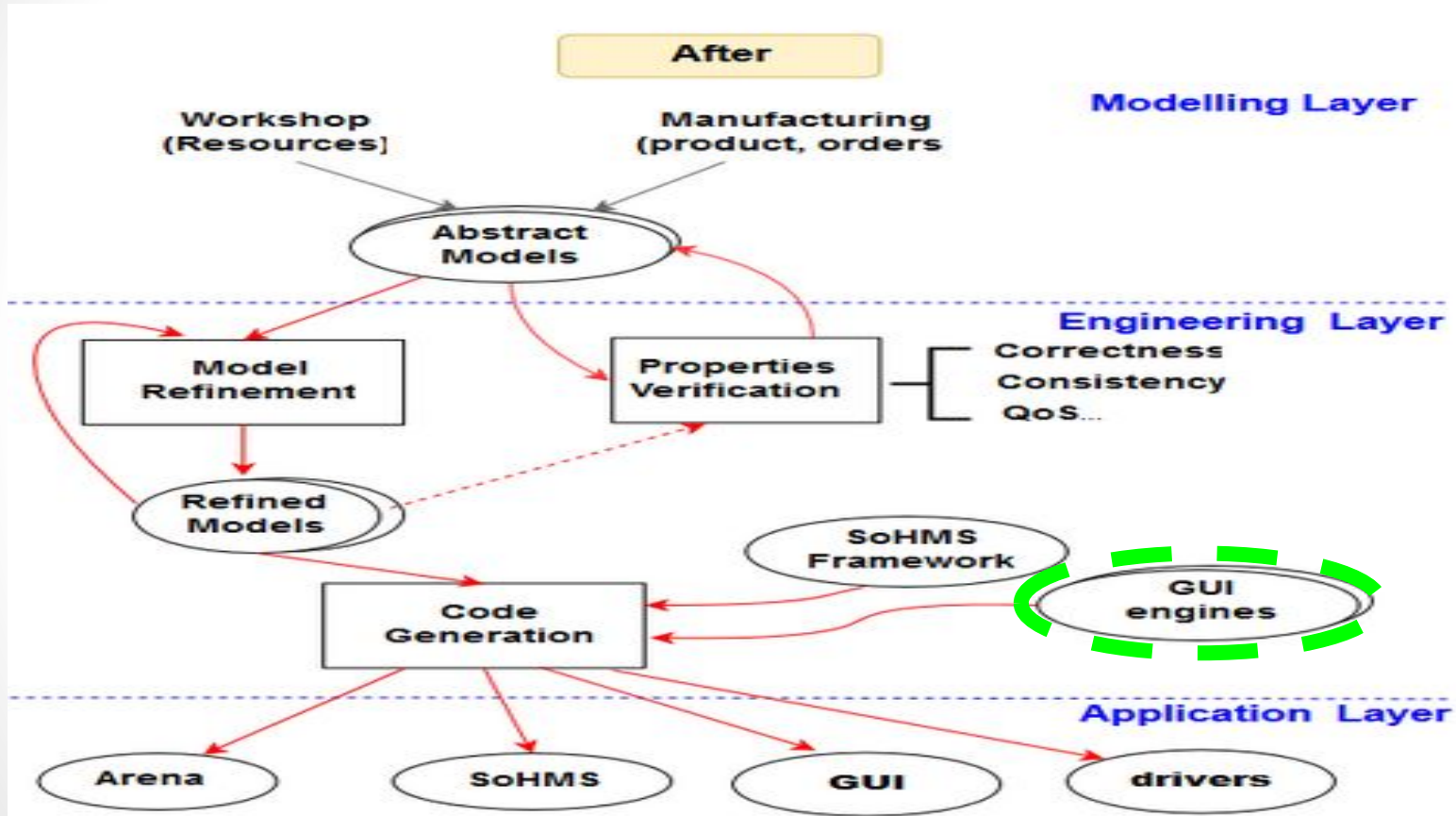
Example of generic use and implementation of resources

New Vision:



Abstract Models (More details ...)

New Vision:

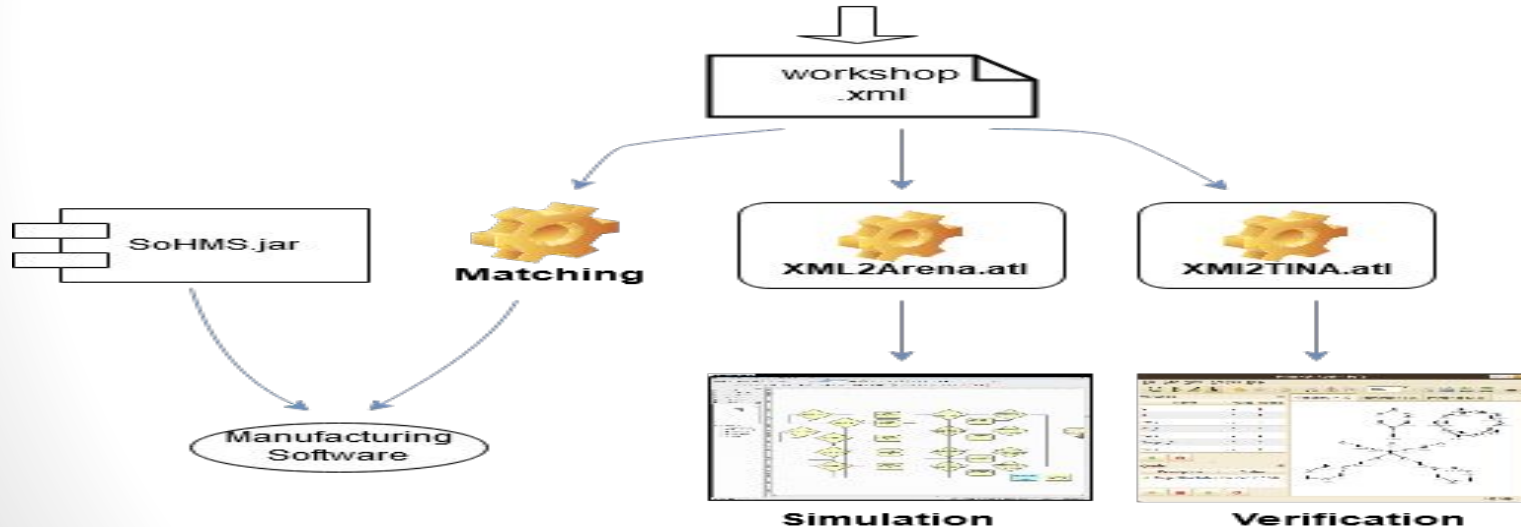


Ongoing works:

GUI engines & Code generation (More details ...)



Manufacturing Modelling Editor

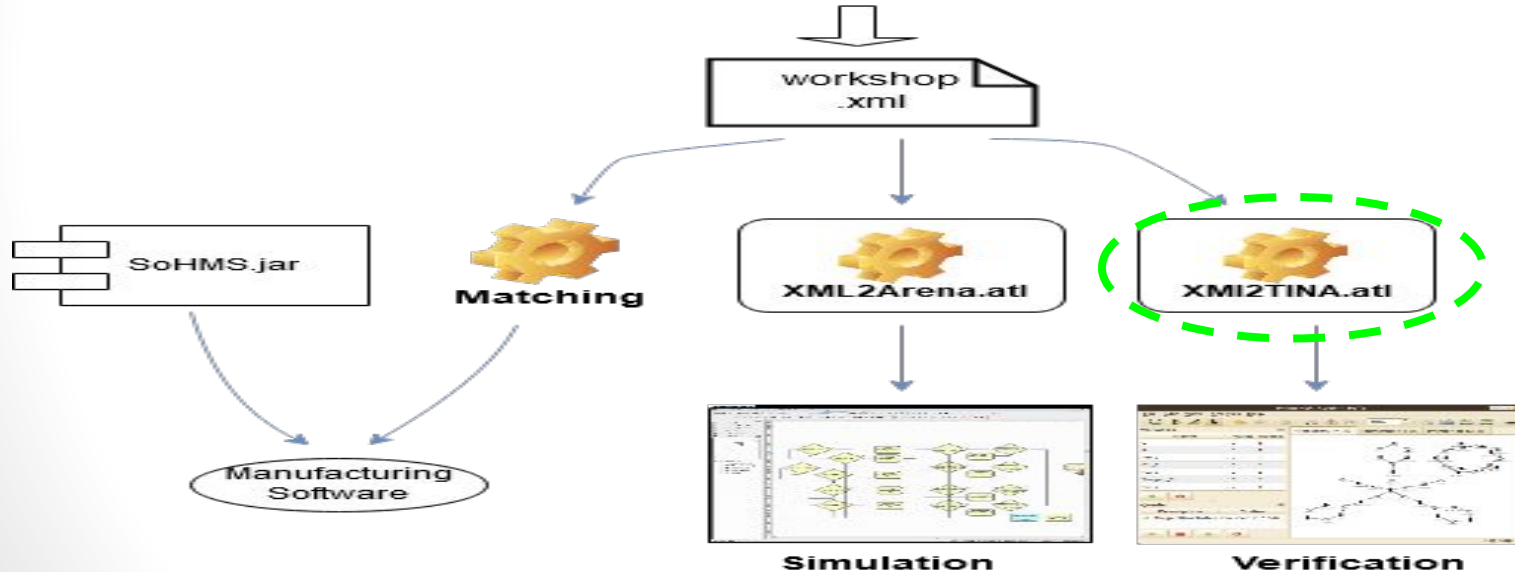


MME (Modelling Manufacturing Editor)

GUI engines & Code generation (More details ...)

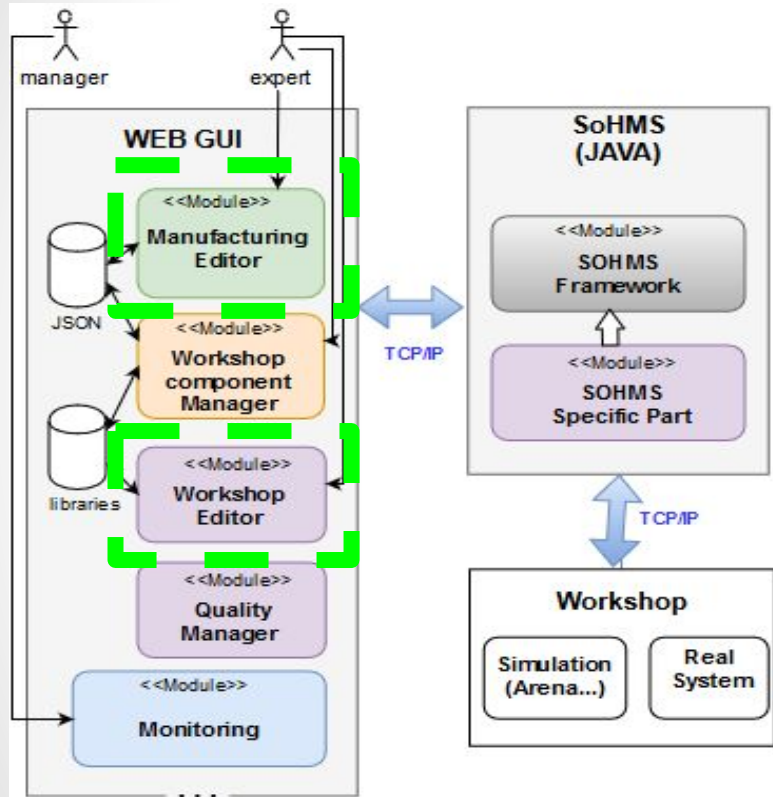


Manufacturing Modelling Editor

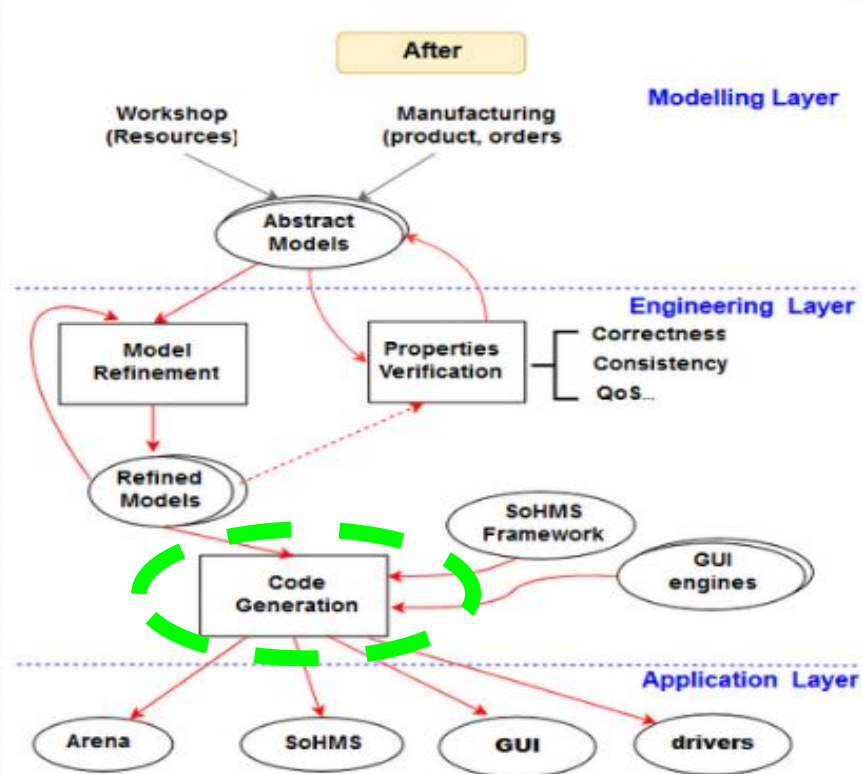


MME (Modelling Manufacturing Editor)

New Vision:



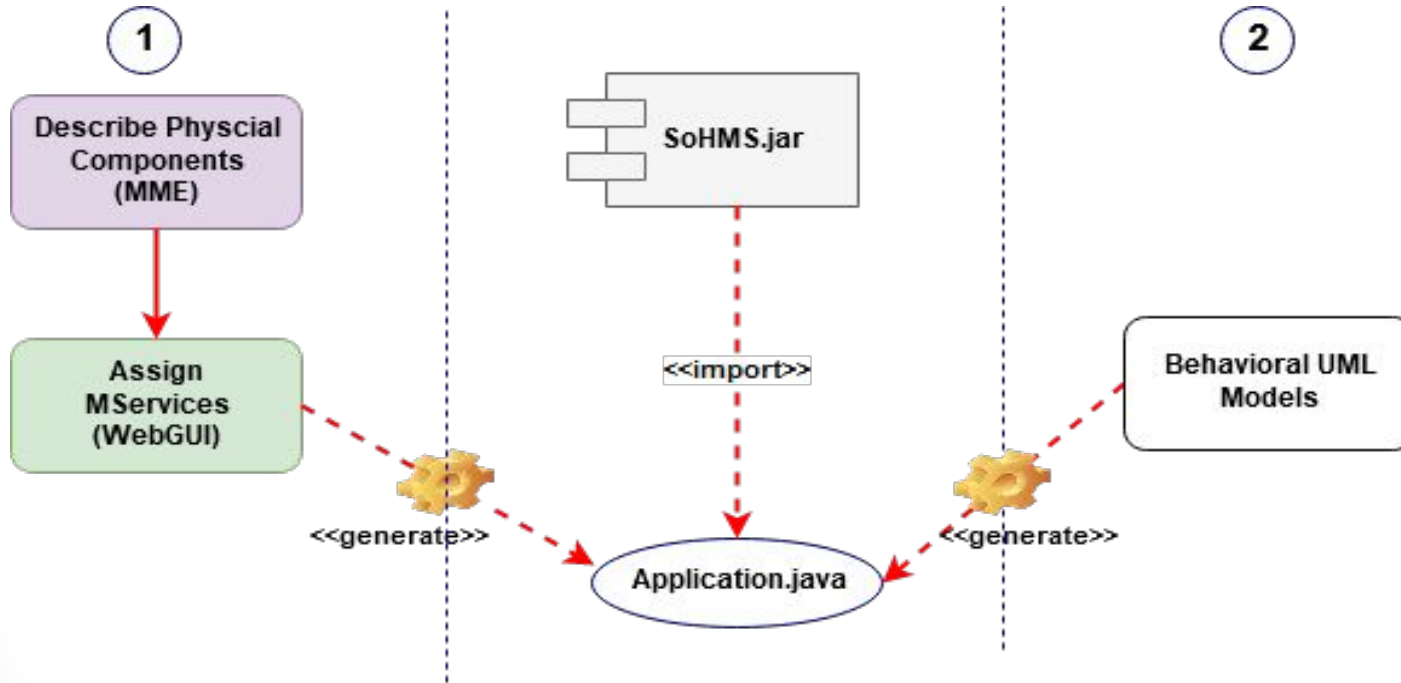
Application Architecture



Software Construction Process

Ongoing works:

Code generation (More details ...)



Petri net generator (Example...)

```
query XML2PN = MM!Layout->allInstances()->  
  asSequence()->first().generatePN().  
  writeTo('/XML2TINAPN/ws.net');
```

```
helper context MM!Layout def: generatePN() : String =  
  let ports : Sequence(MM!Ports) = MM!Ports.allInstances() in  
  'net generatedPetriNetFromManufacturingEditor\n'+  
  self.nodes->iterate(station; places : String = "" | places +  
  station.generatePlace()+'\n')+  
  ports->iterate(port; chain: String = " | chain +  
  port.generateTransition()+'\n');
```

.....

Generated Petri Net (Example...)

net generatedPetriNetFromManufacturingEditor

pl loop2

pl loop1

pl loop3

pl loop4

pl agv1(1)

pl agv2(1)

pl Post1

pl poste2

tr t1 poste2*1 -> loop2*1

tr t2 loop2*1 -> agv1*1

tr t3 loop1*1 -> poste2*1

tr t4 agv2*1 -> loop1*1

... ..

Generated Petri Net (Example...) (2)

76 C:\Users\tebmed\Desktop\sohms-dev\Transformation\XI

File Edit View Tools Help

```
net generatedPetriNetFromManufacturingEditor
pl loop2

pl loop1

pl loop3

pl loop4

pl agv1(1)

pl agv2(1)

pl Post1

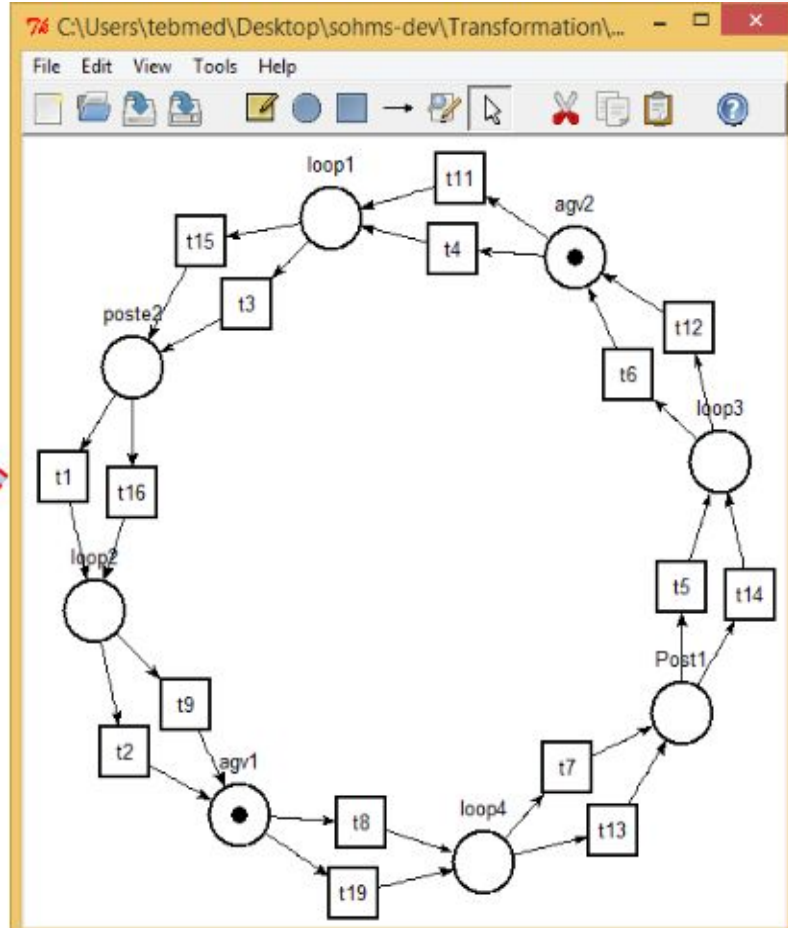

pl poste2

tr t1 poste2*1 -> loop2*1
tr t2 loop2*1 -> agv1*1
tr t3 loop1*1 -> poste2*1
tr t4 agv2*1 -> loop1*1
tr t5 Post1*1 -> loop3*1
tr t6 loop3*1 -> agv2*1
tr t7 loop4*1 -> Post1*1
tr t8 agv1*1 -> loop4*1
tr t9 loop2*1 -> agv1*1
tr t10 agv1*1 -> loop4*1
tr t11 agv2*1 -> loop1*1
tr t12 loop3*1 -> agv2*1
tr t13 loop4*1 -> Post1*1
```

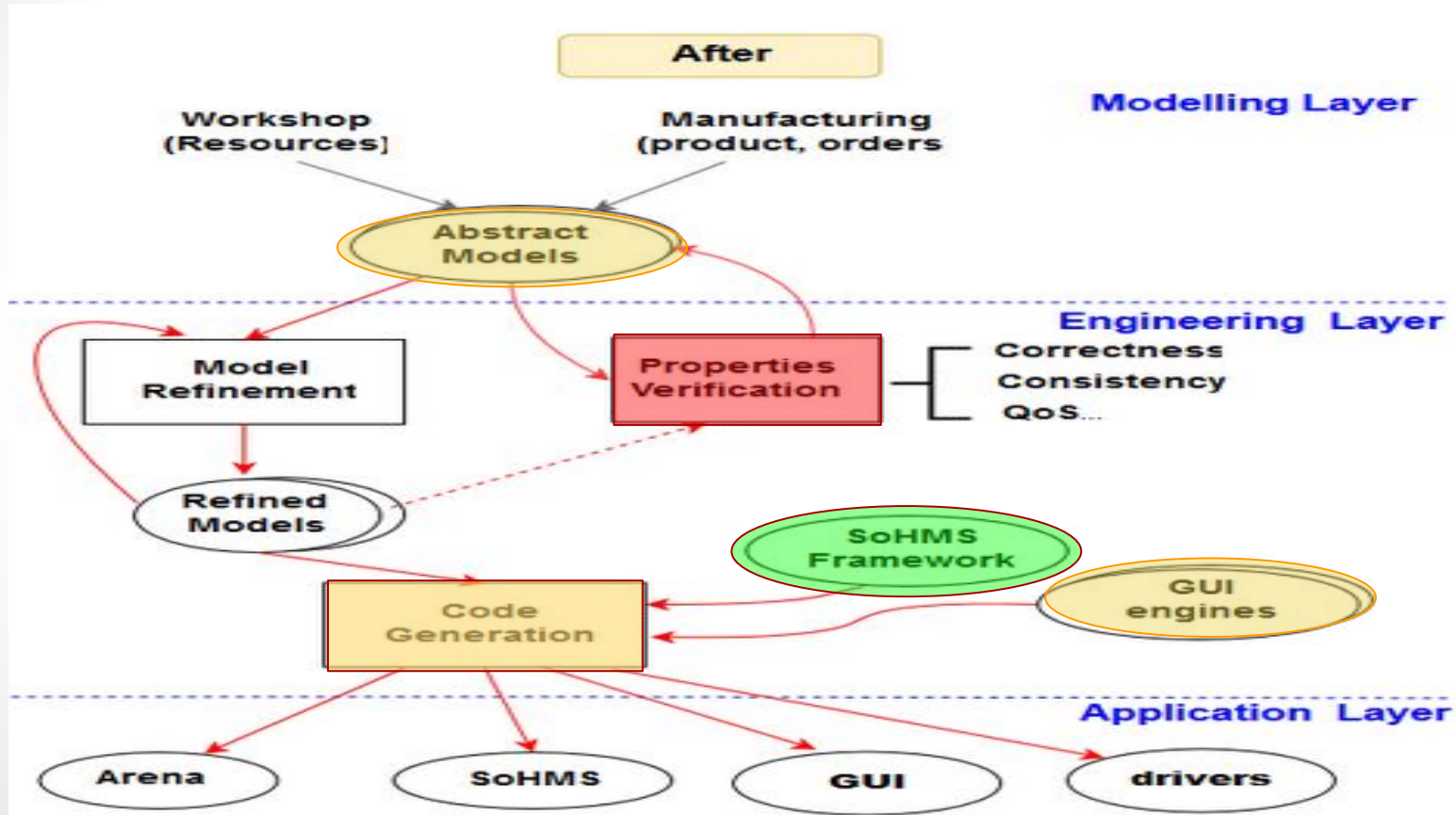
76 nd warning

syntax check successfull

OK



Conclusion



Thank you for your attention

Any Questions ?!

