Master
ENGINEERING (MSE)

Hes·so VALAIS WALLIS

DeSEm
Laboratory - Development of the DeseNet Protocol Stack

Dominique Gabioud

Michael Clausen

Thomas Sterren

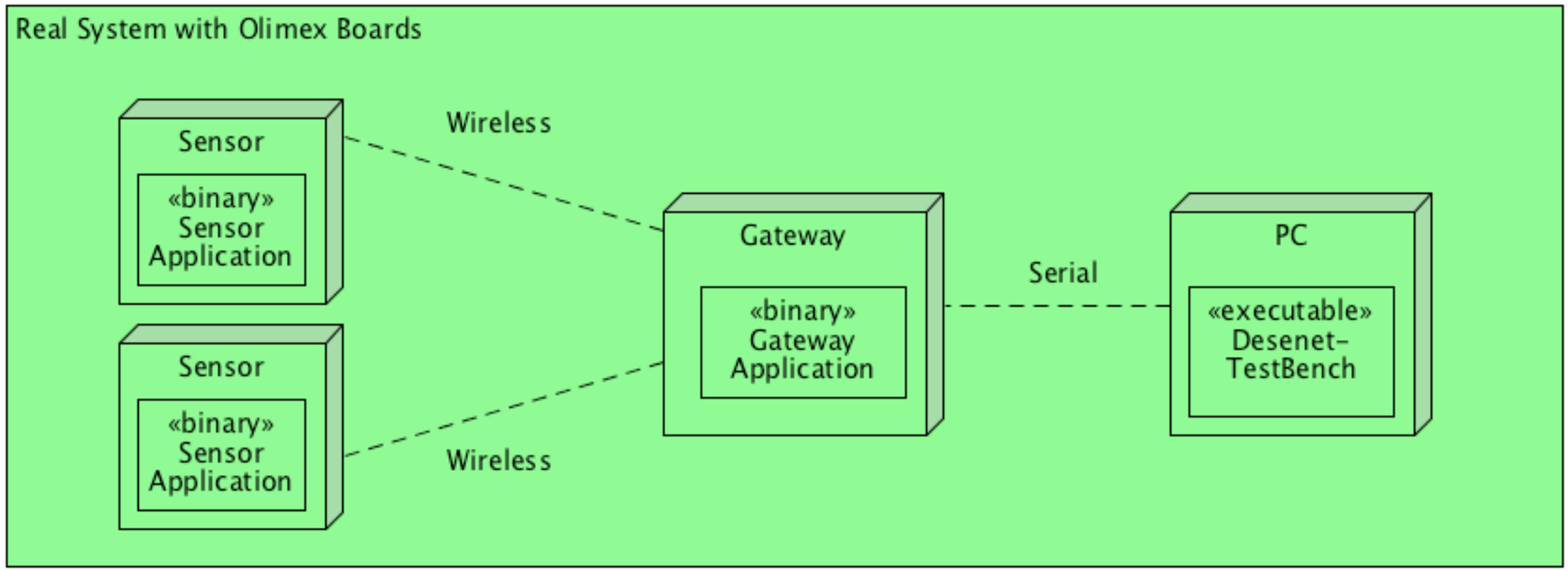
Medard Rieder

HES-SO 2021/22

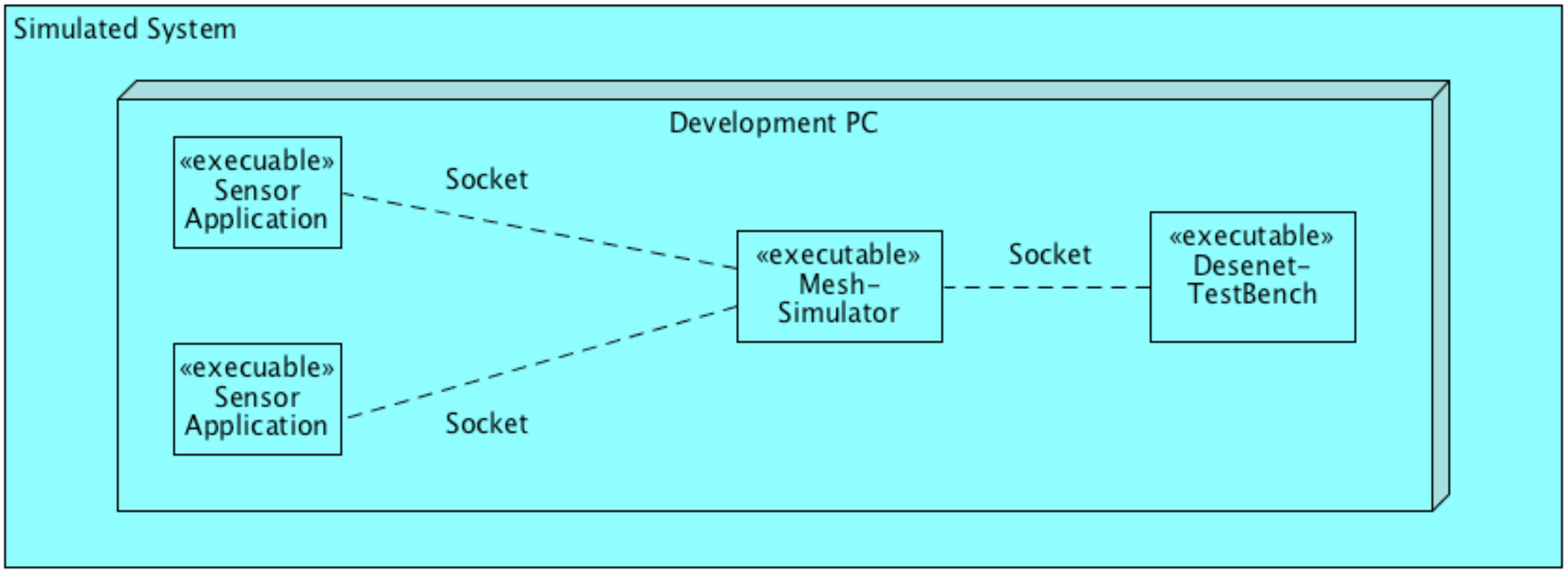
Table of Content

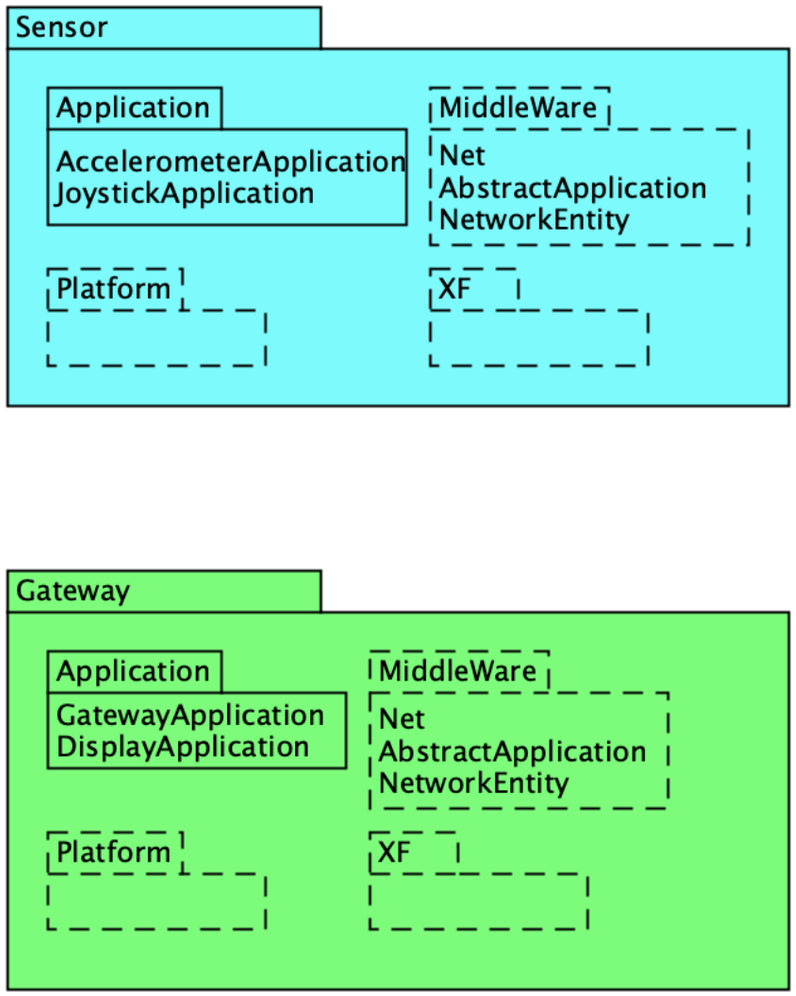
- Overview
- Architecture
- Protocol Design
- Task setting
- Grading

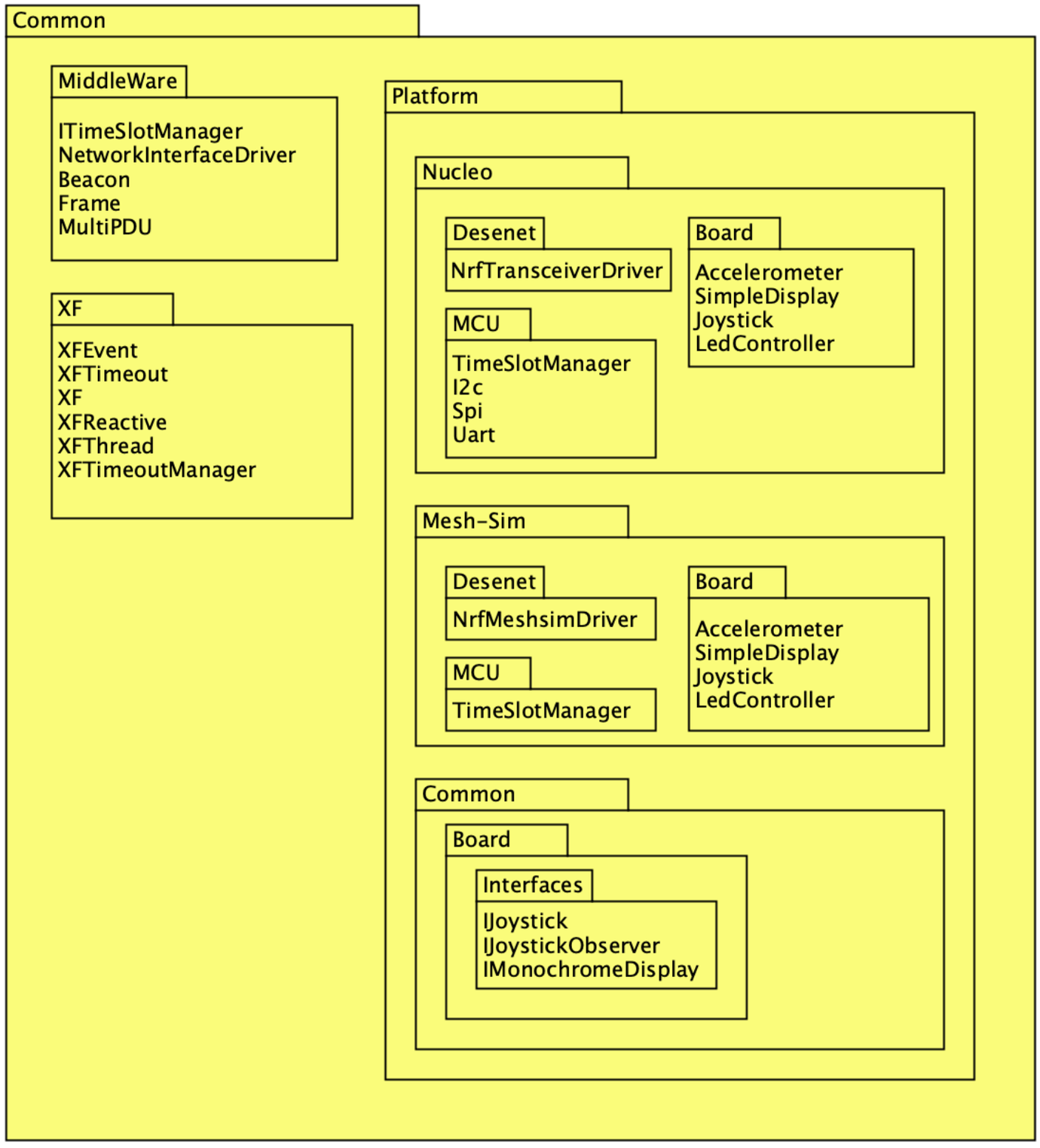
System with real Nodes



System with virtual Nodes

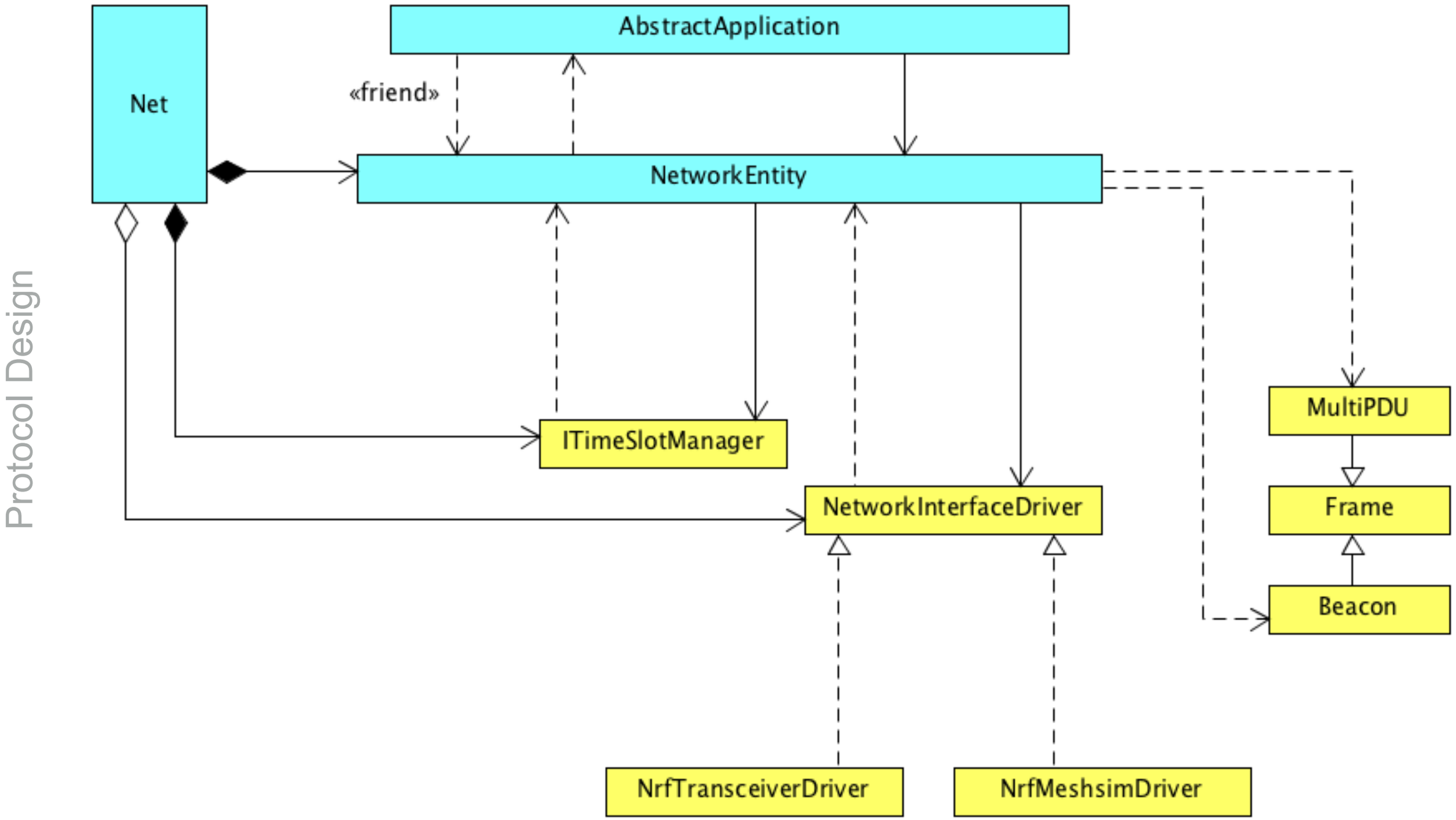




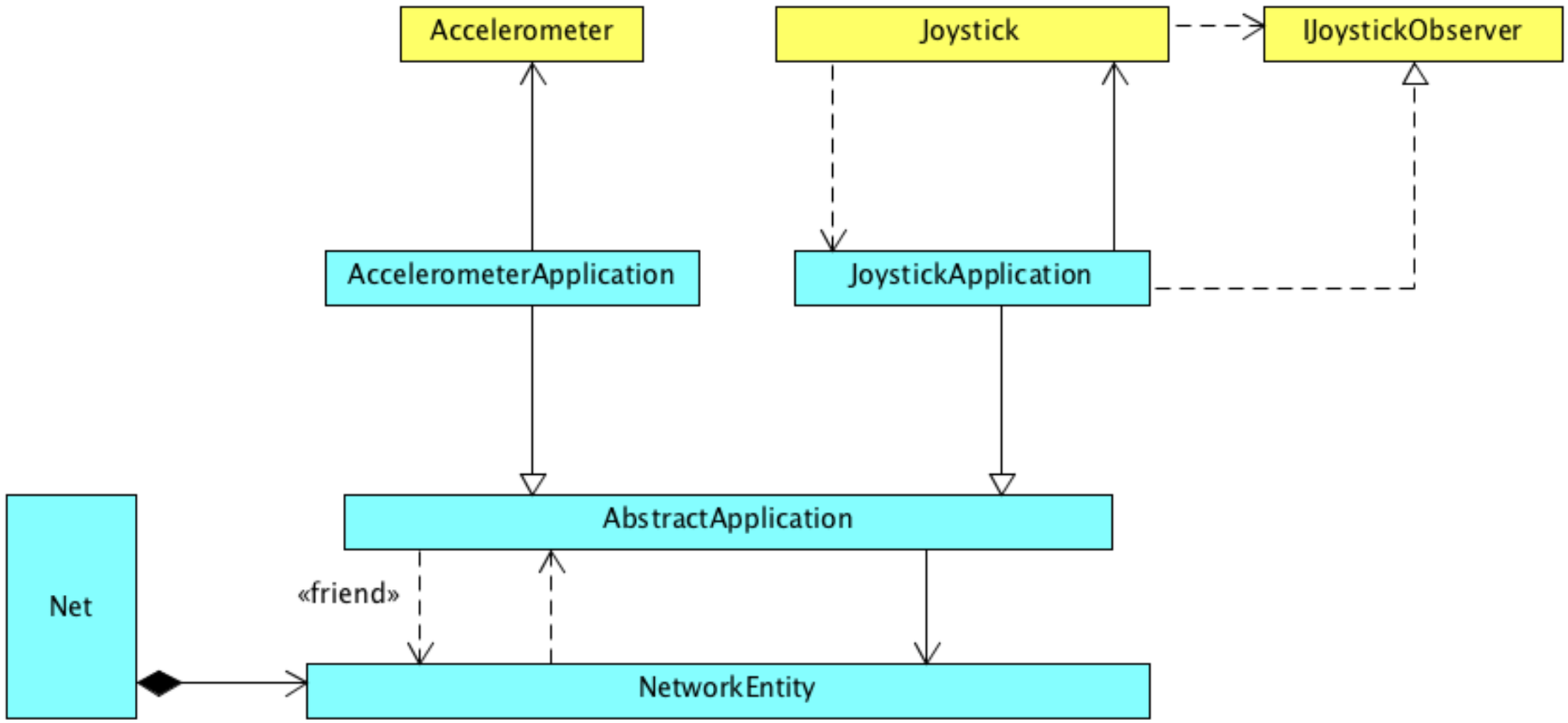


Architecture

Protocol Elements



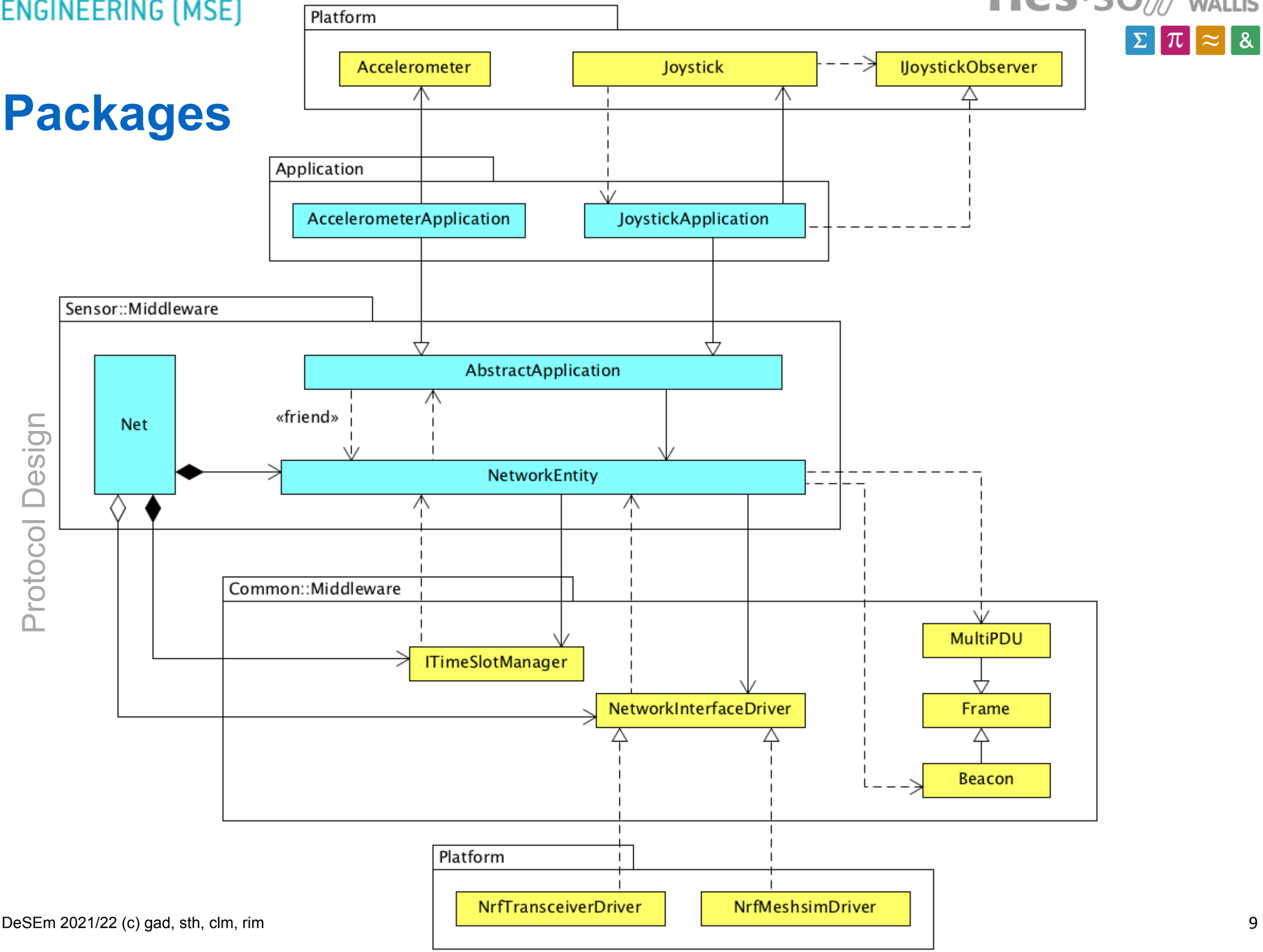
Application Elements



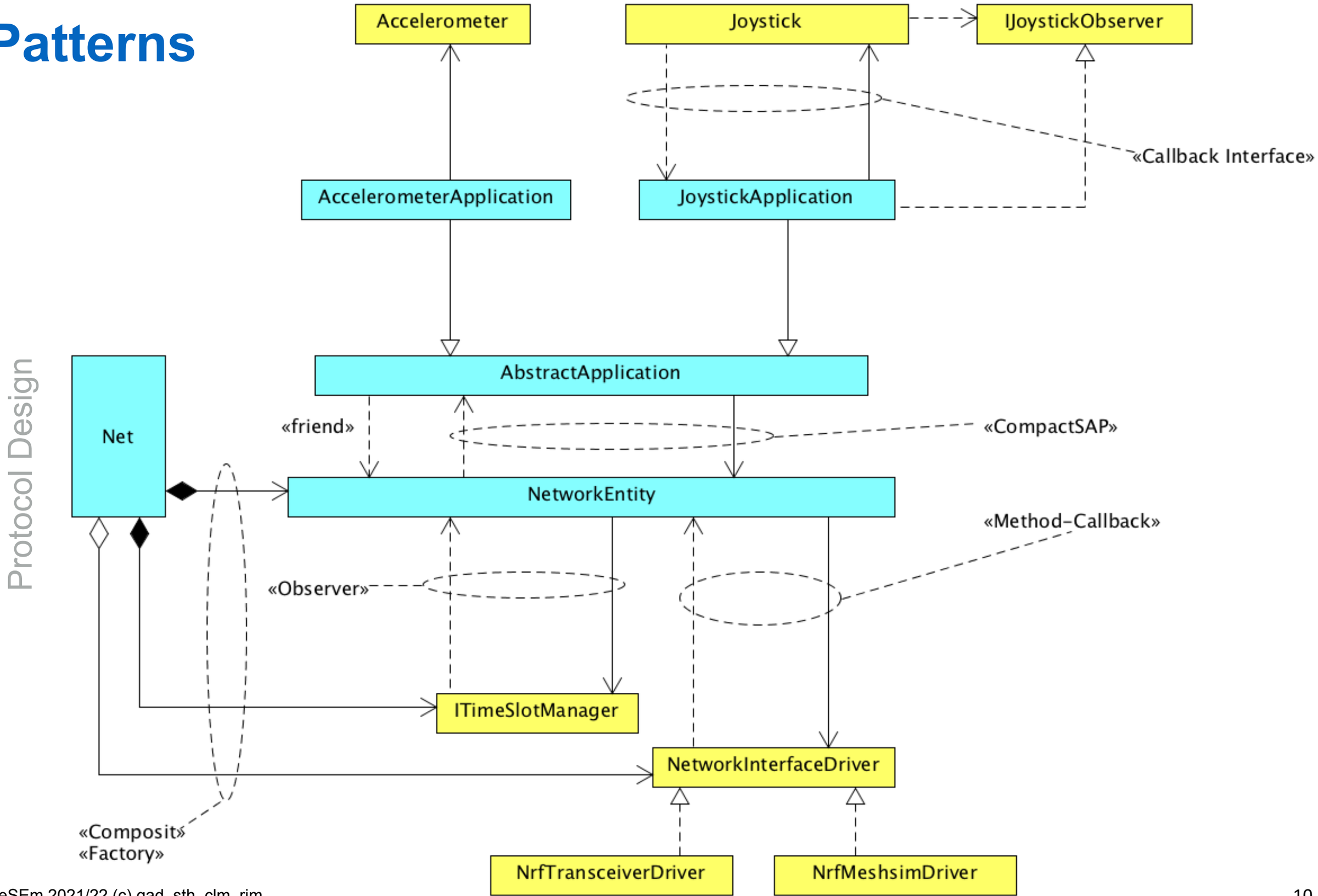
SAP

AbstractApplication
svSyncRequest():void # svPublishRequest(group:SvGroup):bool # evPublishRequest(id:EvId, evData:SharedByteBuffer):void # evSubscribeRequest(id:EvId):void - svSyncIndication(time:NetworkTime):void - svPublishIndication(group:SvGroup, svData:SharedByteBuffer):SharedByteBuffer::sizeType

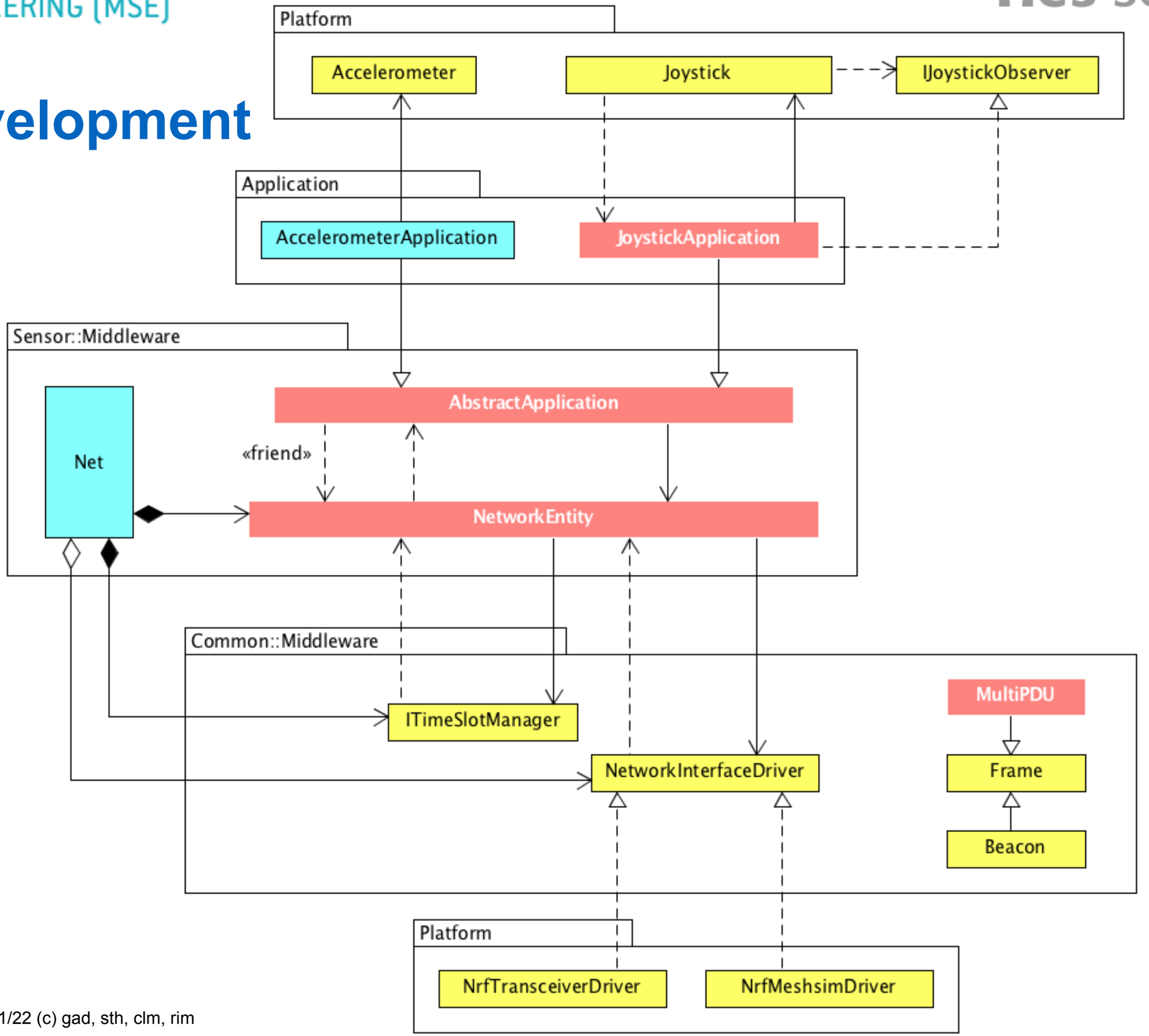
Packages


Platform
Accelerometer
Joystick
IJoystickObserver
Application
AccelerometerApplication
JoystickApplication
Sensor::Middleware
AbstractApplication
«friend»
Net
NetworkEntity
Common::Middleware
ITimeSlotManager
NetworkInterfaceDriver
MultiPDU
Frame
Beacon
Platform
NrfTransceiverDriver
NrfMeshsimDriver

Patterns


Protocol Design
Accelerometer
Joystick
IJoystickObserver
«Callback Interface»
AccelerometerApplication
JoystickApplication
AbstractApplication
Net
«friend»
«CompactSAP»
NetworkEntity
«Method-Callback»
«Observer»
ITimeSlotManager
NetworkInterfaceDriver
«Composit»
«Factory»
NrfTransceiverDriver
NrfMeshsimDriver

Development



Development Process

- Study the preset project and the structure of it
- Understand classes, their relations and patterns used (using these slides)
- Develop the simulated solution using the Mesh-Sim environment
  - Step 1: Receive beacons
  - Step 2: Implement notification of the applications on reception of the beacon
  - Step 3: Implement MultiPDU class
  - Step 4: Collect sampled values, put them into MultiPDU and send it a the right slot
  - Step 5: Implement the Joystick application
- Test the simulated solution using the Mesh-Sim environment
  - Define and describe test cases
  - Test and document each test case
  - Generate an error / a todo list

Development Process continued

- „Port“ your simulated solution to the Nucleo target
  - Step 1: Rebuild for Nucleo target
  - Step 2: Flash run and debug it
- Test your Nucleo solution against the Gateway that will be present in class room
  - Define and describe test cases
  - Test and document each test case
  - Generate an error / a todo list
- Documentation
  - During all the development, create UML diagrams as possible or necessary. Use class and sequence diagrams as well as state charts.
  - Comment well your code !!!
- Delivery
  - Eclipse project without compiled code. UML diagrams in PDF format. Pack everything into a ZIP archive with the name „FirstnameLastname.zip“

How we will grade you

- This is how we will generate marks:
- No delivery at all : 1.0
- DeseNET protocol not working: 2.5
- DeseNET protocol and joystick application working on simulator (4.0)
- DeseNET protocol and joystick application tested on simulator and tests documented 5.0
- DeseNET protocol and joystick application working on target: 5.5
- DeseNET protocol and joystick application tested on target and tests documented 6.0
- No code documentation (-1.0)
- Bad or insufficient code documentation (-0.5)
- No model documentation (-1.0)
- Bad or insufficient model documentation (-0.5)
- No test documentation (-1.0)
- Bad or insufficient test documentation (-0.5)
- No pattern usage (-0.5)
- Copy from other : For involved persons maximum mark is 3.0

Plan Your Time well

9	22.11.21	Desem protocol entity design	Desem protocol design	Desem protocol design
10	29.11.21	Desem protocol implementation & test		
11	06.12.21			
12	13.12.21			
13	20.12.21			
14	10.01.22	Reserve week (Desem protocol implementation & test)		
15	17.01.22	Prepare Exams		
16	24.01.22	Exam		

Communication Theory | Communication Lab / Exercise

Project

Software Engineering Theory | Software Engineering Lab / Exercise