

ISE

HW/SW Design

System Design (SD) aktiviteter

- **SD1.** **Generel system design**
- **SD2.** **HW arkitektur design**
- **SD3.** **SW arkitektur design**
- **SD4.** **Interface design**
 - **specifikation af komponenternes interface**



SD3. HW arkitektur design

- Denne aktivitet foretages parallelt med den tilsvarende software arkitektur aktivitet
- For nogle typer af produkter vil en given funktionalitet kunne implementeres såvel i SW som i HW
- HW arkitekturen kan dokumenteres vha. SysML diagrammer
- En vigtig aktivitet er definering af snitfladen mellem hardware og software
- (Se kompendium p. 154-174)

Design Specifikation

- "Requirements" Specifikation
 - Krav til systemet set med kundens øjne (Use Cases)
 - Set udefra
 - Hvad skal systemet gøre?
- Design Specifikation
 - Krav til system set med designerens øjne
 - Set indefra
 - Hvordan skal systemet udføre kundens krav?
 - Specificer systemets grænseflader
 - Henvender sig til konstruktøren/programmøren

Design Metrics

Embedded System Design

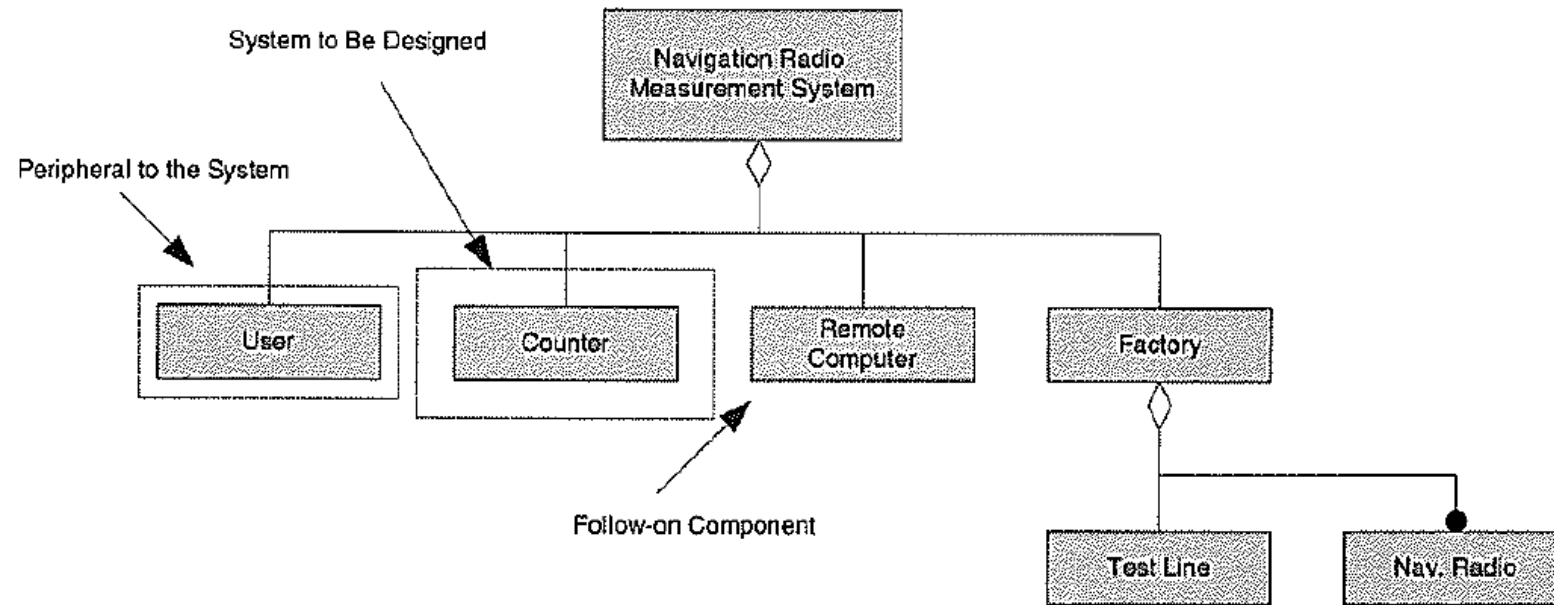
(Frank Vahid, kompendium s.202)

- Cost
 - NRE – engangs investering – udviklingstid og udstyr
 - Unit Cost – pris for produktion materialer og til
- Performance
 - Latency, response time and throughput
- Power
 - Hvad er kravet til effektforbruget og køling?
- Flexibility
 - Er det nemt at tilføje ny funktionalitet til systemet?
- Time-to-market
 - Tiden det tager at udvikle et nyt system før det kan sælges til kunderne
- Maintainability
 - Service venlighed – er det nemt og hurtigt at foretage reparation?
- Correctness
 - Opfører systemet sig som det skal i alle situationer?
- Safety
 - Kan system gøre skade på bruger eller andre?

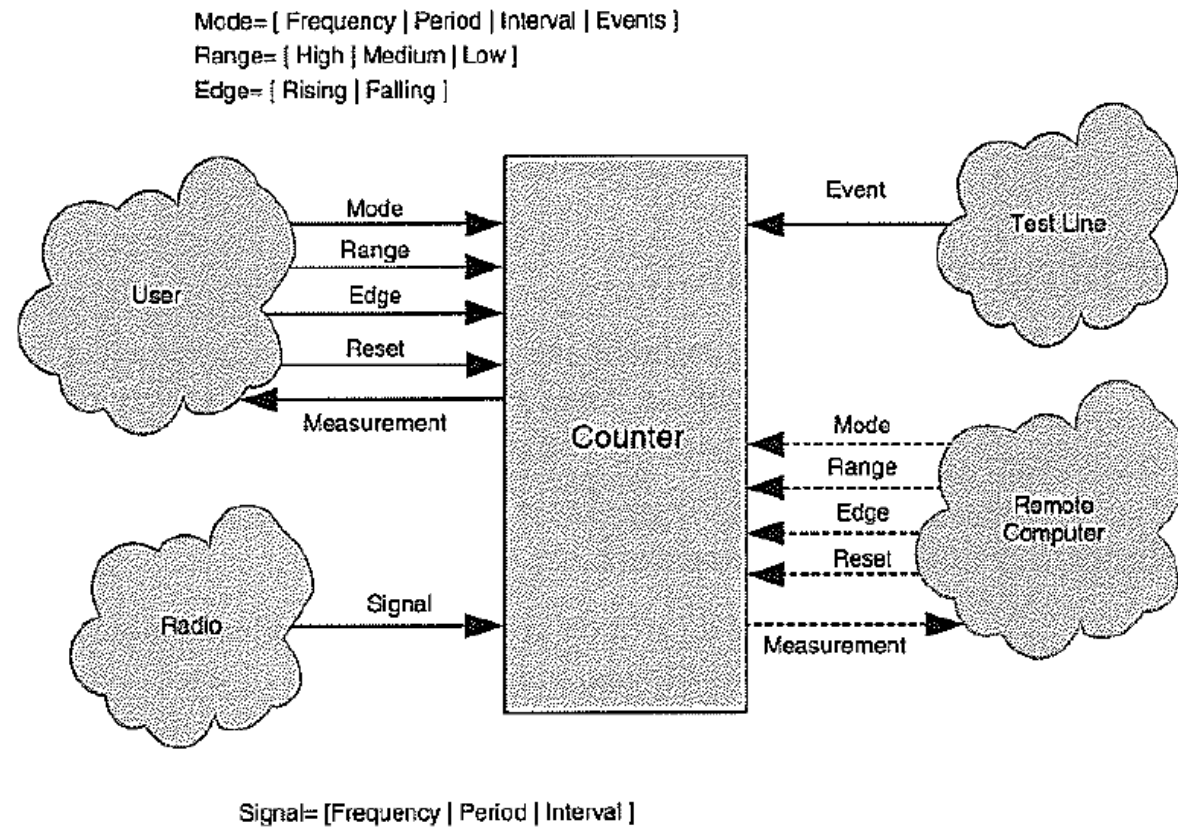
Kvalificering af systemet

- Inputs og Outputs
 - Navn på signaler
 - Retning af signaler
 - Naturen af signalet (Event, data, state, analogt, digital)
 - Målbare værdier (Values, timing, range, levels, tolerance)
- Funktion og Operation
 - Væsentlige funktioner
 - UML domain- og applikations-analyse
- Begrænsninger (Constraints)
 - Geografisk (Begrænsning af afstande for kommunikation)
 - Grænseflader og signaler (Elektrisk, Optisk, Netværk)
 - Brugergrænseflade (Læsbarhed, design af UI)
 - Svar tider (Hard/Soft real-time)
 - Strømforsyning (Tolerancer)
 - Sikkerhed (Safety) og pålidelighed (Reliability) – MTBF + MTTF

Model eksempel (Counter)



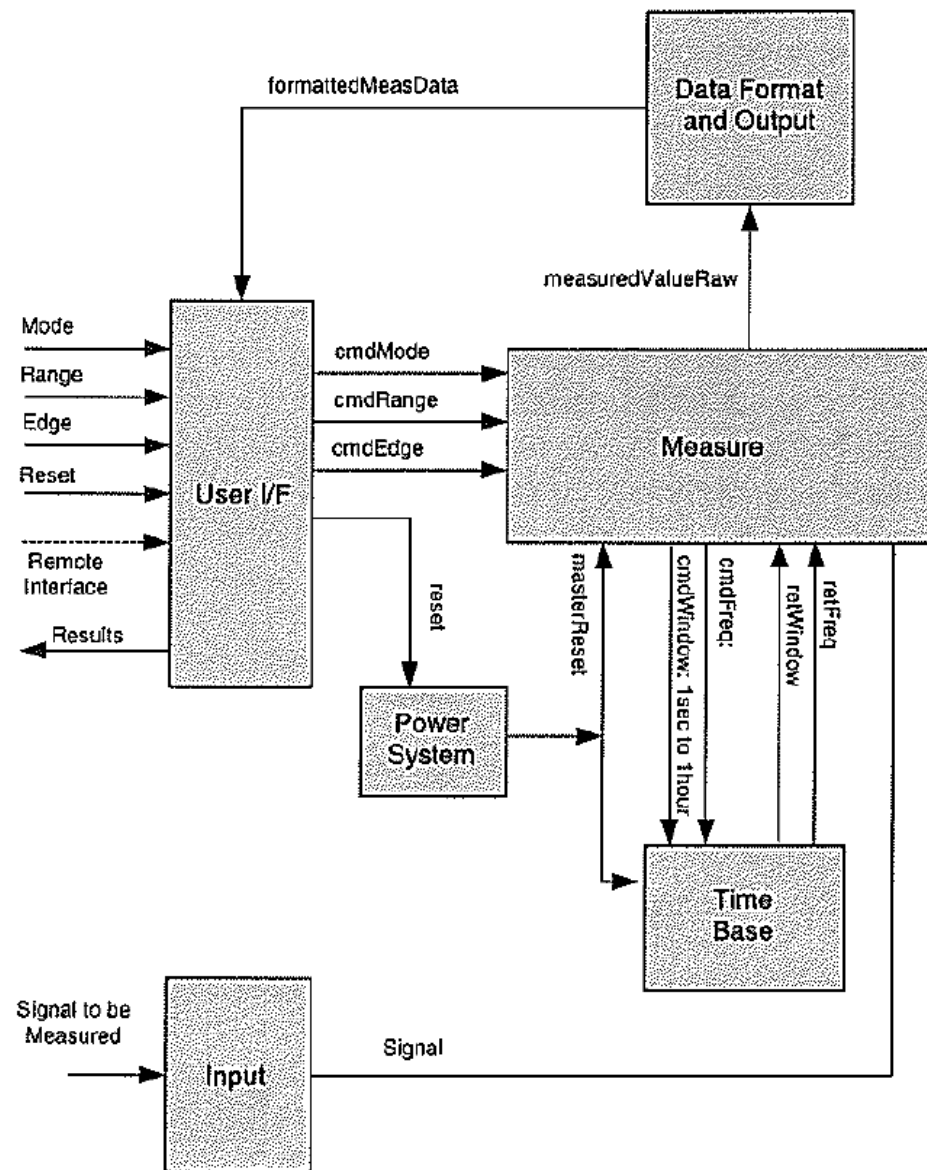
Counter-Environment Interface



Funktionel design

- Hvordan kravene kan blive realiseret af et system, der består af en **række funktionelle blokke**
- Dette er en **iterativ proces**, hvor systemet **dekomponeres** i funktionelle blokke, der hver har en **entydig funktion**, der skal realiseres
- Det funktionelle design beskriver den **interne struktur af adfærd** af det ønskede system
- Designet beskrives med **blokke og interfaces** (SysML diagrammer)
- Det **logiske view** er designet beskrevet uafhængigt for valg af hardware og software komponenter

Functional Design of the Counter System (Logisk View)



Partitioning and decomposition of a system

- **Decomposition**

- Nedbrydning af systemet i komponenter
- Formålet er at indkapsle systemets funktionalitet i adskilte komponenter

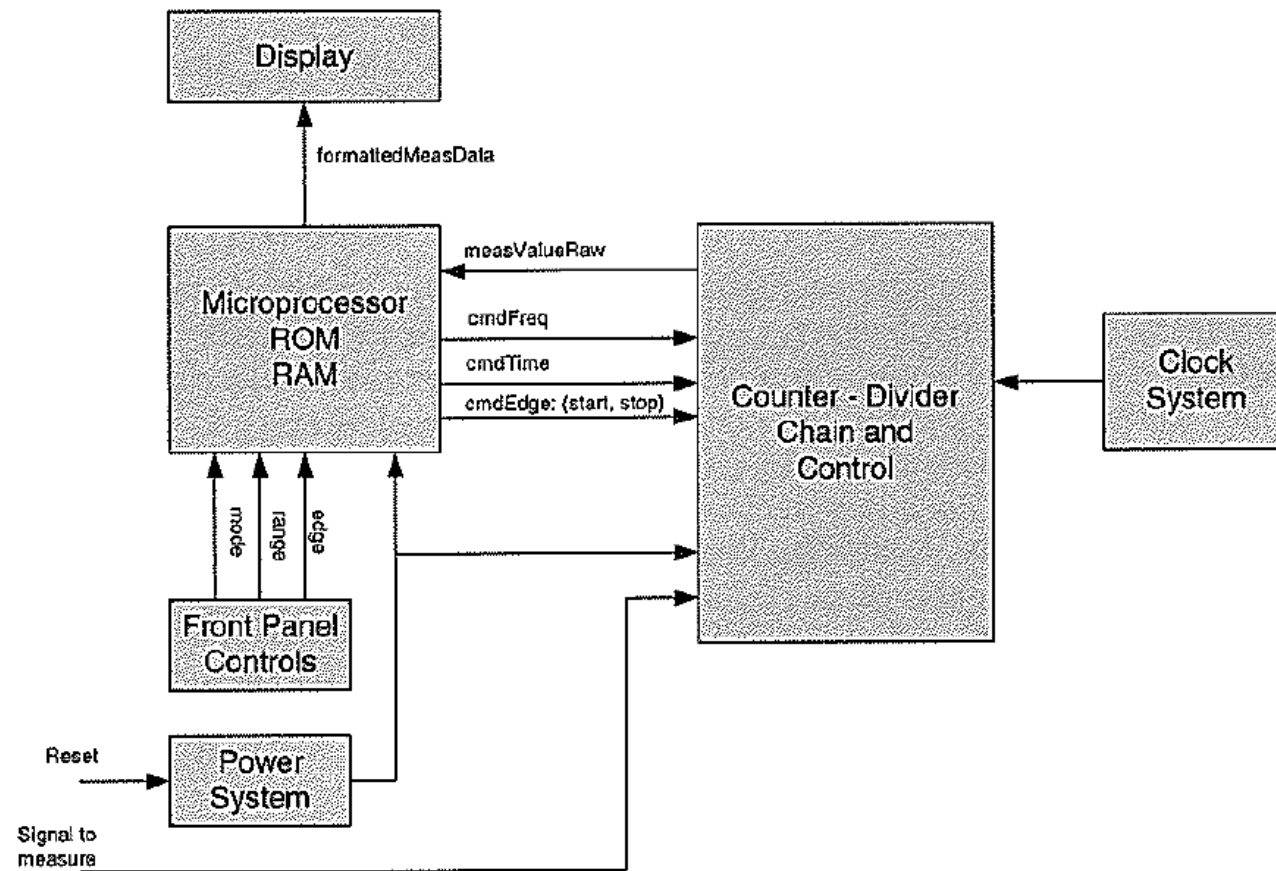
- **Partitioning**

- Disse komponenter fordeles i systemet og udføres af enten hardware eller software
- Nogle komponenter kan implementeres i både software eller hardware

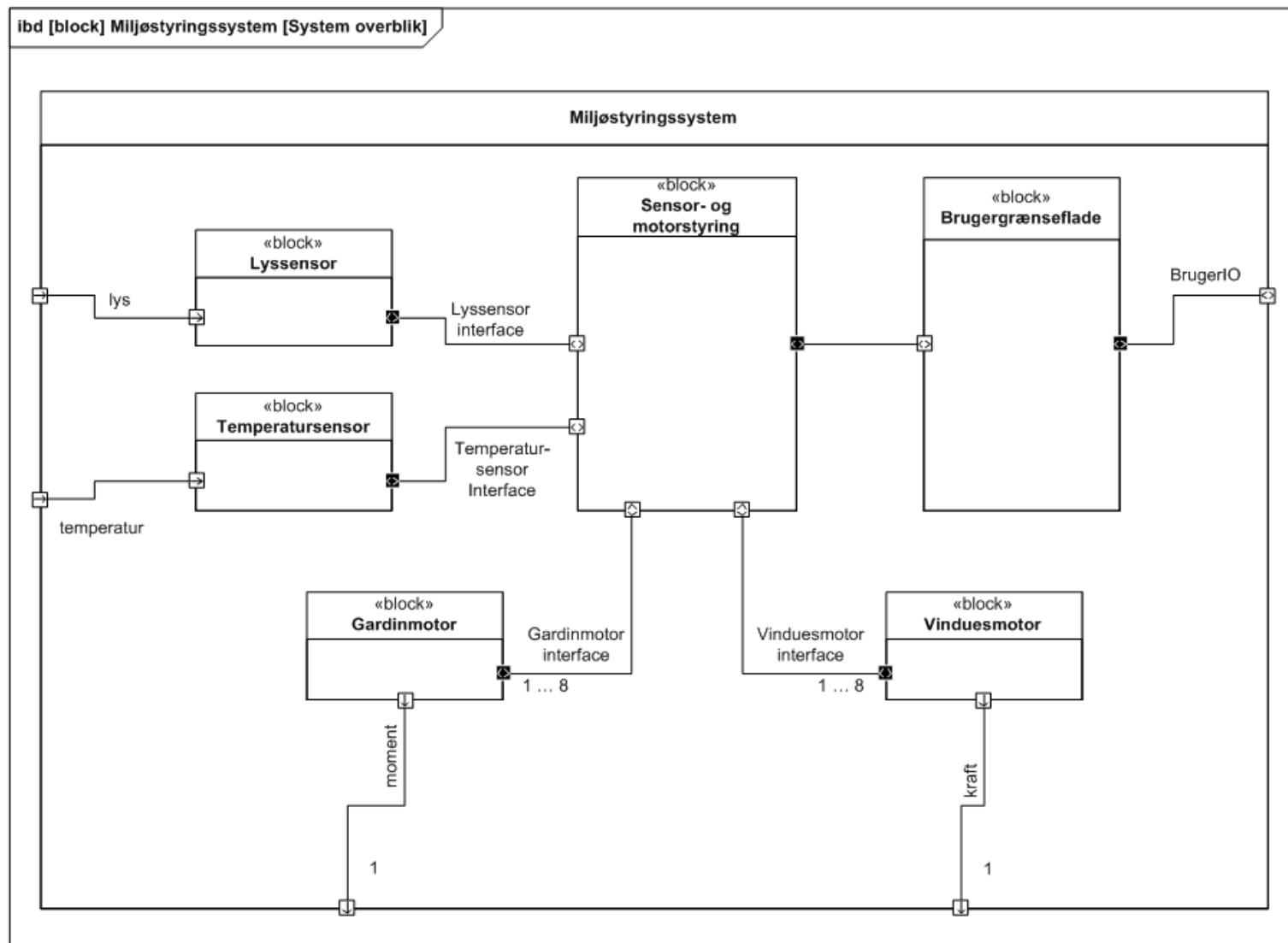
Architectural Design

- **Hardware og software komponenter vælges**
- **Mapping**
 - Mapning af funktioner til hardware
 - Funktioner mappes til den fysiske hardware block
- **Alloker logiske SW blokke til den fysisk HW platform med SysML allocate**
- Eksempel
 - Strømforsyning, display, porte mappes til hardware
 - Operativ System og driver mappes til software
 - Kontroller mappes til mikroprocesser

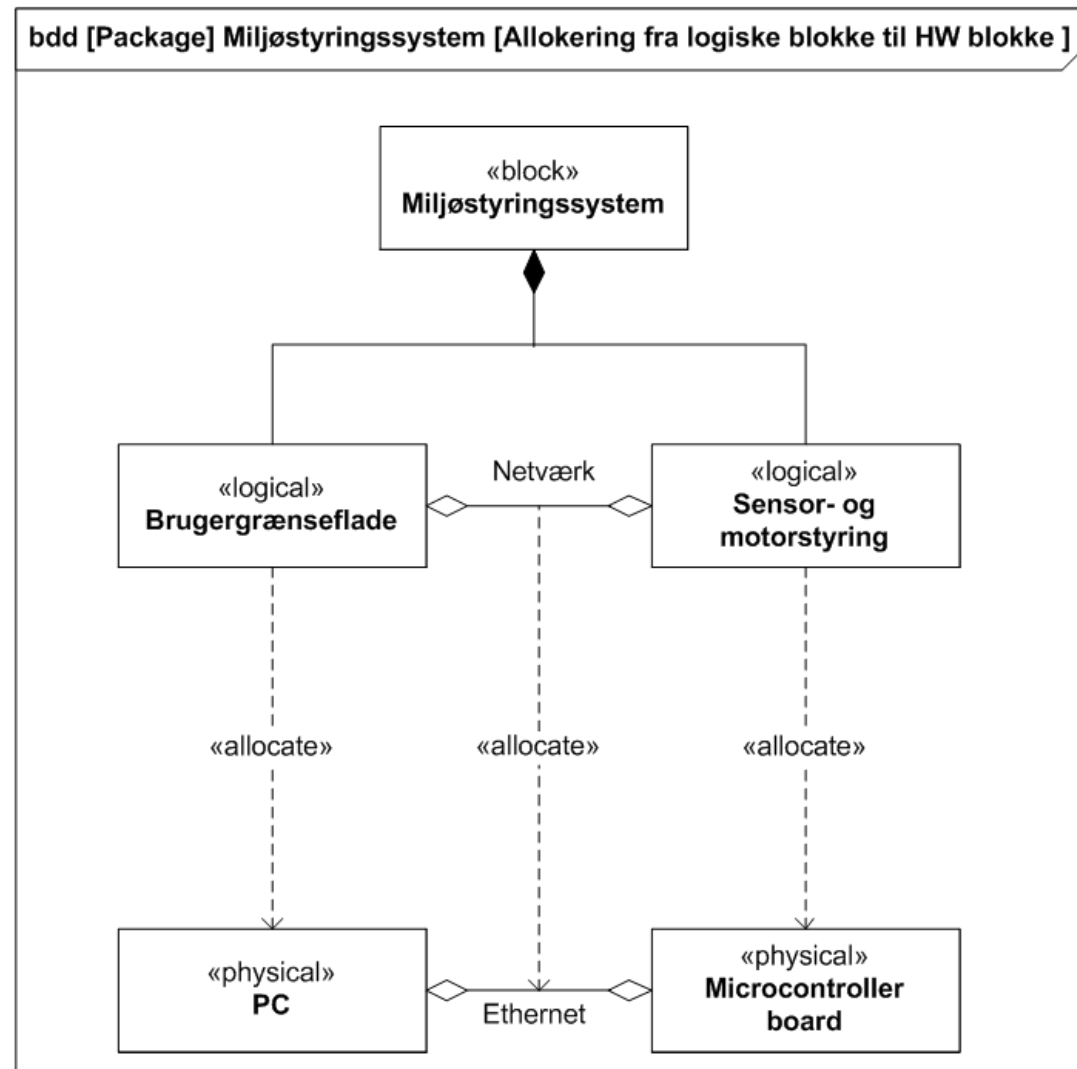
Arkitektur eksempel (Counter)



Arkitektur for Miljøstyringssystemet (Logical view)



Deploymentmodel beskrevet med SysML og "allocate"



System Design (SD) aktiviteter

- **SD1.** **Generel system design**
- **SD2.** **HW arkitektur design**
- **SD3.** **SW arkitektur design**
- **SD4.** **Interface design**
 - **specifikation af komponenternes interface**



SD3. SW arkitektur design: metodeguidelines

For de Use Cases, der indgår i iterationen udføres aktiviteterne A, B og C parallelt

A. Bearbejdning af den logiske model fra analysen

A1. Opdel den logiske model i delsystemer

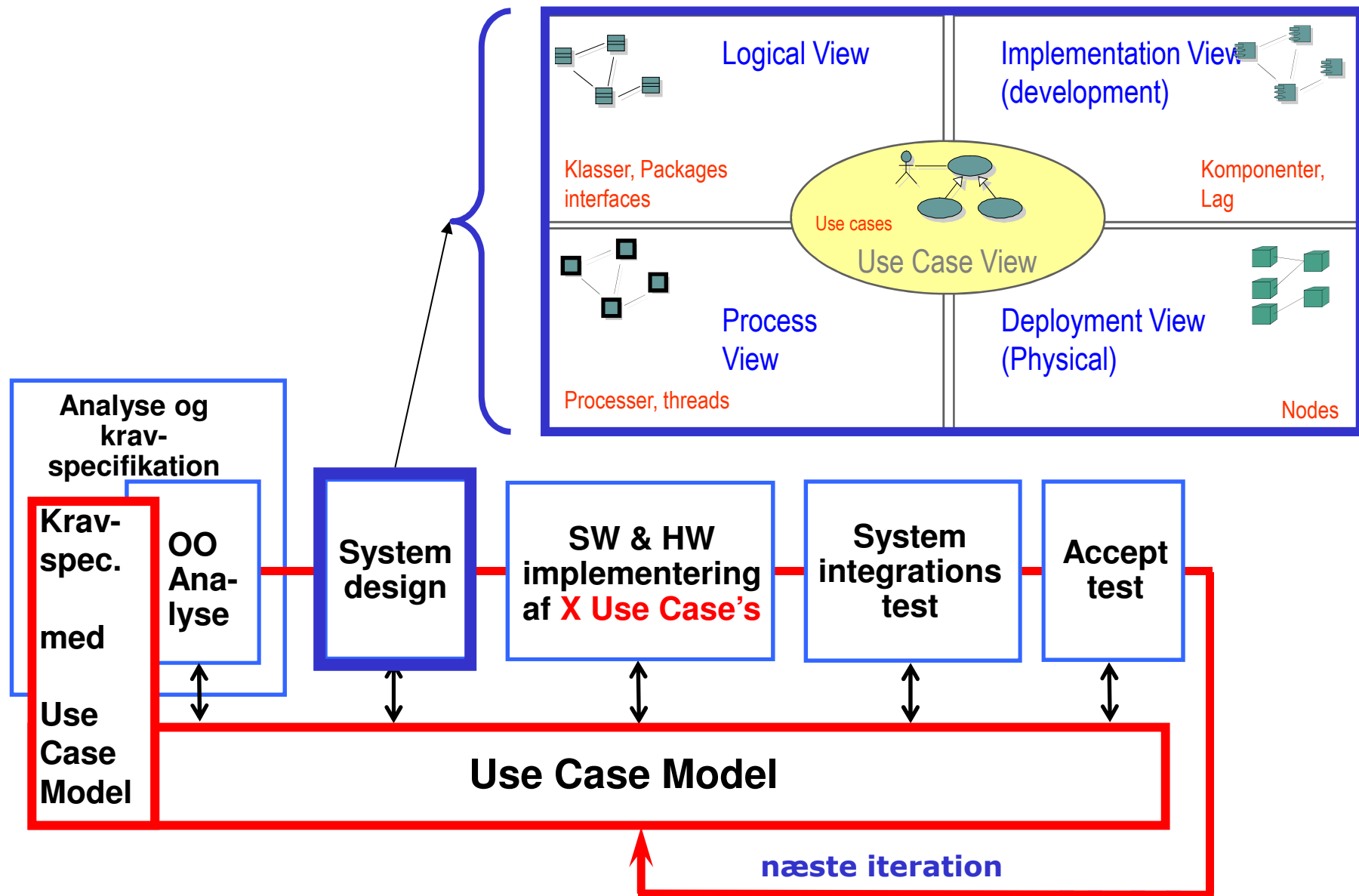
A2. Overvej/anvend Arkitektur Patterns (Lagdelt, Client/Server)

B. Udarbejd en deploymentmodel

- Definerer computere og øvrige HW enheder

C. Dokumenter og review Use Casen

Design Views med 4+1 View modellen



SD3.A1: Opdel den logiske model i delsystemer

- Fordele ved opdeling i delsystemer:
 - Abstraktion: overordnet systemforståelse
 - Dekomposition: parallelle udviklingsforløb
 - Genbrug: generelt anvendelige delsystemer
- Delsystemer indeholder klasser, relationer imellem klasser samt eventuelt andre delsystemer
- UMLs Pakker (Packages) anvendes til at dokumentere delsystemerne

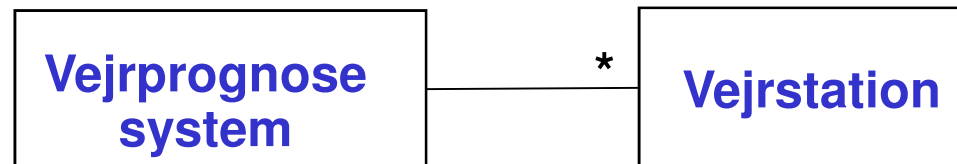
SD3.A1. Guidelines for opdeling i delsystemer

- Opdel den logiske model i delsystemer ud fra følgende prioriterede liste:
 - a) fysiske enheder (input fra deploymentmodel)
 - b) eksekveringsfaser
 - c) funktionalitet
- Gå derefter et niveau ned i detaljeringsgrad og gentag opdelingen for hvert delsystem
- Dokumenter opdelingen vha. pakker (UML Packages)
 - Elementer som udfører fælles eller relateret funktionalitet og som har en indbyrdes høj kobling grupperes i samme pakke
 - Der tilstræbes en relativ lav kobling imellem de forskellige pakker

Eksempel på opdeling i delsystemer

Opdel systemet i delsystemer ud fra følgende prioriterede liste

1) Fysiske eksekveringenheder :



2) Eksekveringsfaser:

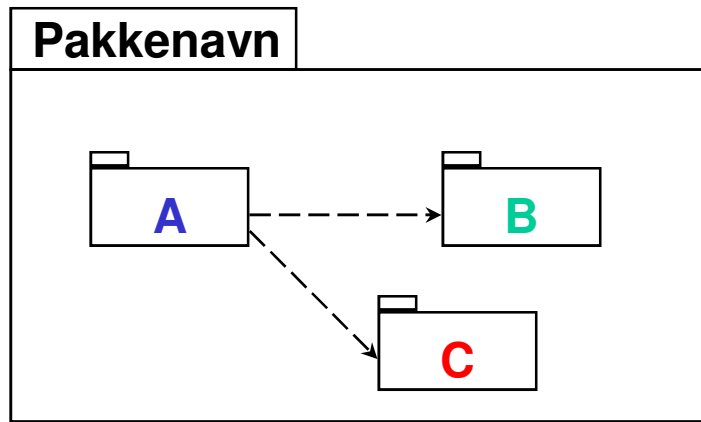
-
- A vertical arrow points from the "Vejrprognose system" box in the previous diagram down to a list of three execution phases. A large curly bracket on the left side of the list groups these phases together.
1. Indsamling af vejrdato
 2. Beregning af vejrpoggnose
 3. Præsentation af vejrpoggnose

3) Funktionalitet:

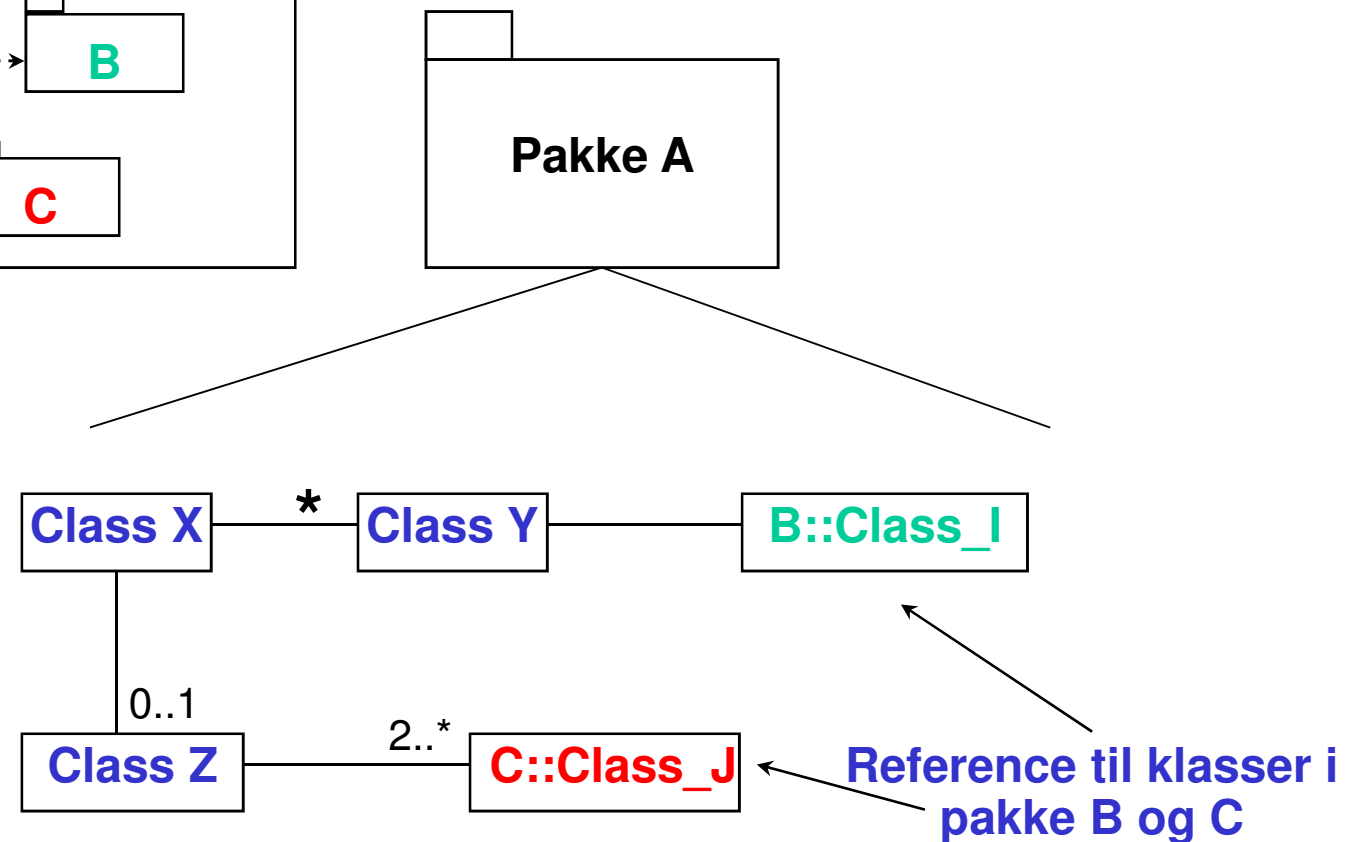
-
- A large curly bracket on the left side of the list groups these four functional items together.
- Vejr statistik funktion
 - Stormvarslings funktion
 - Dagsprognose
 - Langtidsprognose

UMLs notation for pakker

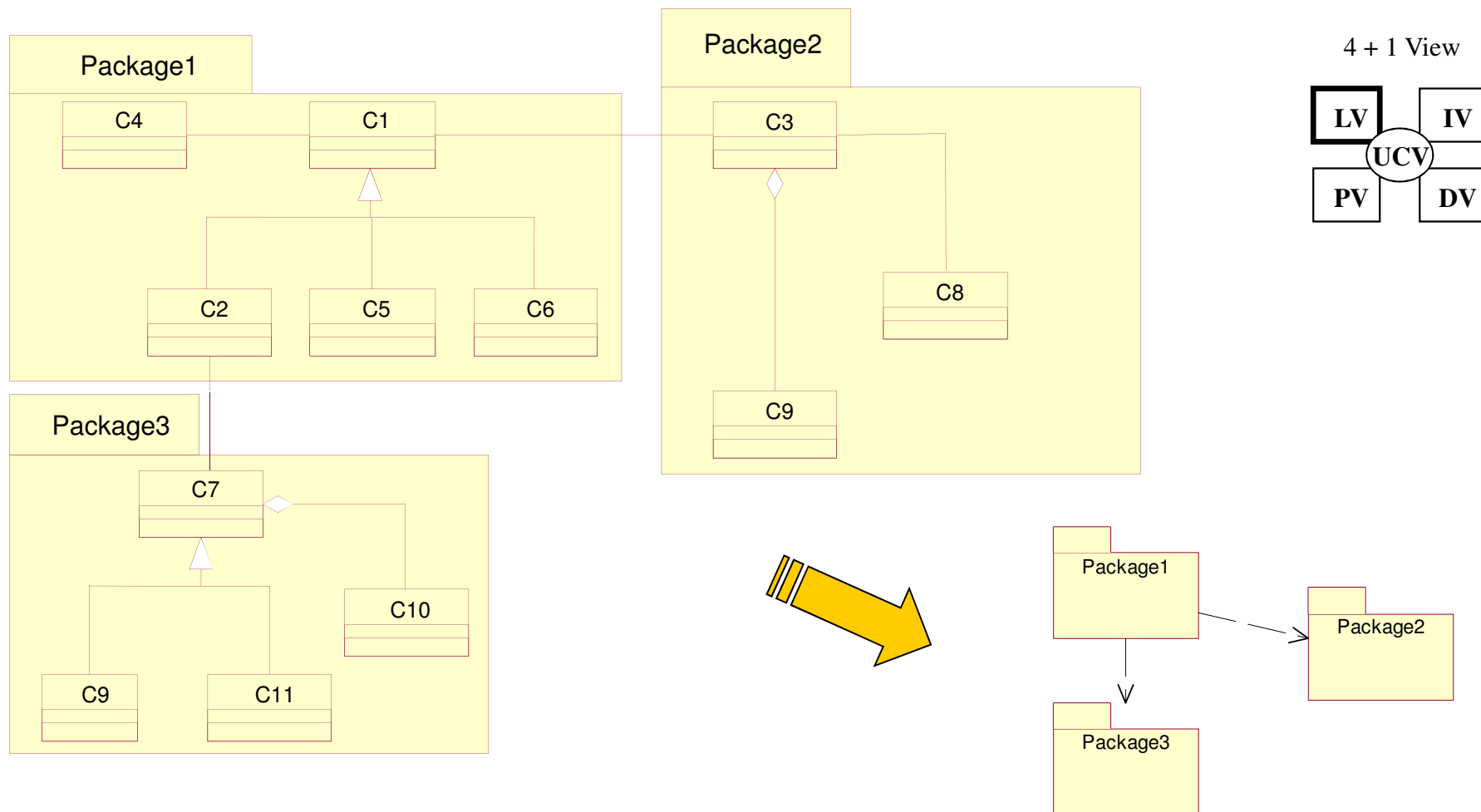
En pakke kan indeholde nye pakker



På det laveste niveau indeholder pakken et klasse diagram



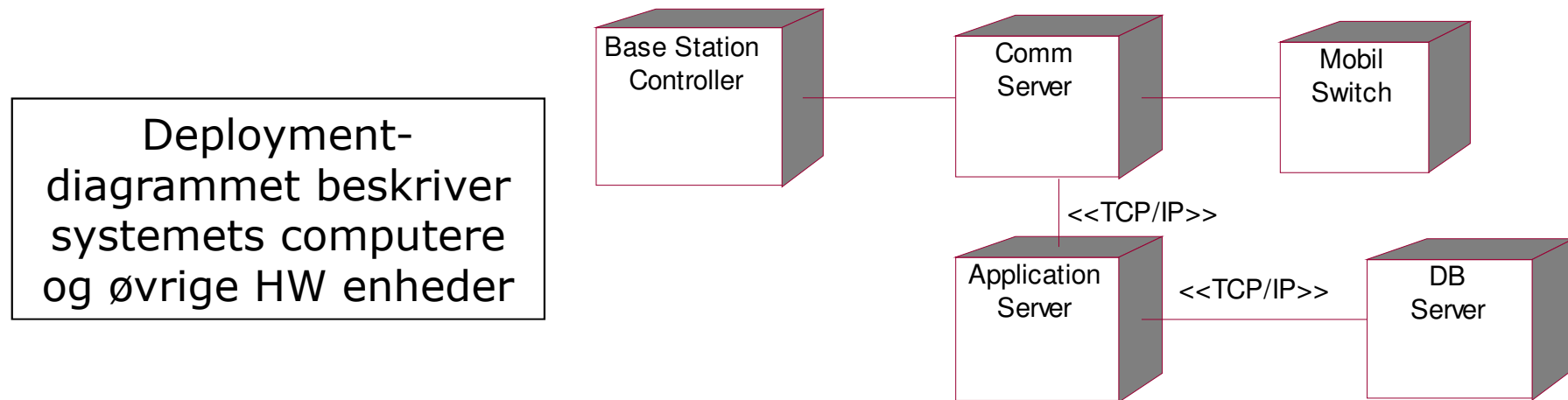
Opdeling af den logiske model i pakker



- Lav kobling imellem pakkerne
- Tætte relationer imellem klasser i en pakke (samhørighed)

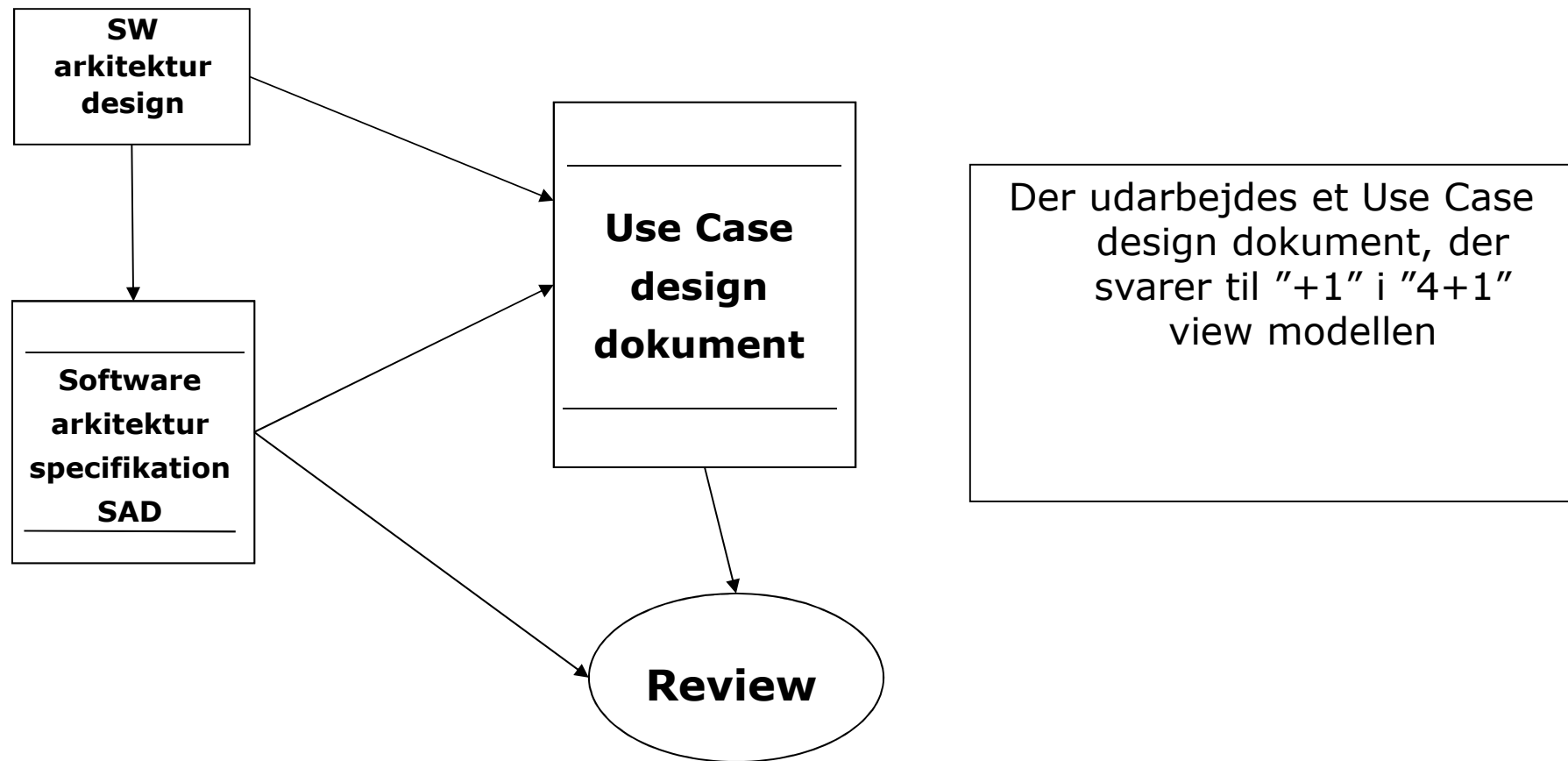
SD3.B: Udarbejd en deploymentmodel

Deploymentmodellen beskriver
deployment viewet i "4+1" modellen



- Den kan være fastlagt på forhånd
- Den kan være oplagt for det aktuelle system
- Der kan være forskellige HW konfigurationer
- Den kan udvikles i tæt samarbejde med HW arkitekterne

SD3.C: Dokumenter og review Use Casen

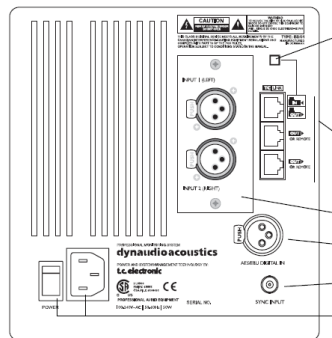


For hver Use Case, der indgår i den aktuelle iteration:
Dokumenteres Use Casens mapning over på elementerne i de fire andre views og dokumentet og SAD'en reviewes

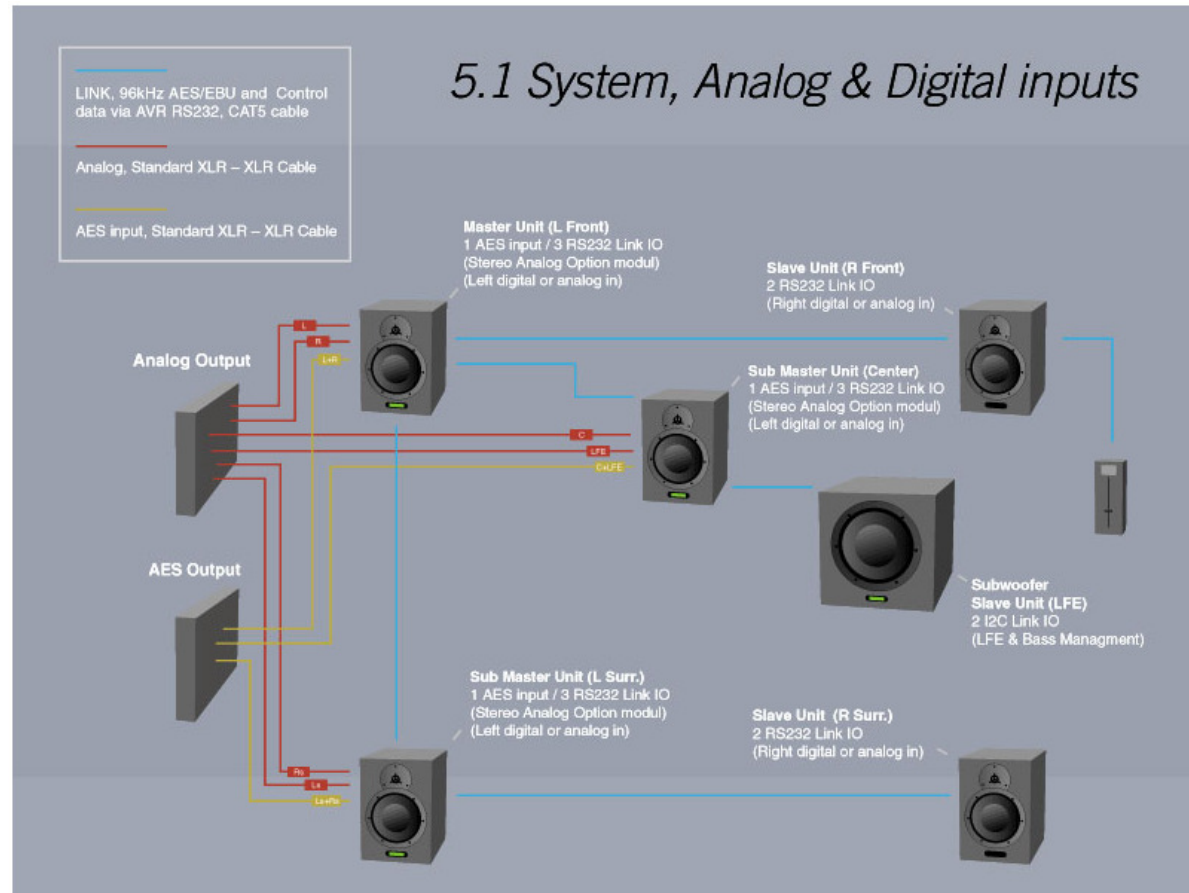
Dynaudio AIR Series – Studio monitors



Master Unit - Analog I/O



AirSoft – PC program to control setup



Analysis and Design Steps (Summary)

- Input a textual requirements specification (Eg. Use Case)
(Or expert knowledge of the domain and desired system)

Analysis - steps

- *Actor Interaction* with the system (Sequence diagram)
- *Domain Model* (Class Diagram)

Architecture design:

- *Subsystems* (Block Definition Diagram)

SW Design

- *Application Model* for subsystems (Class, sequence and states)
- *Logic Model* (Packages – SW subsystems)

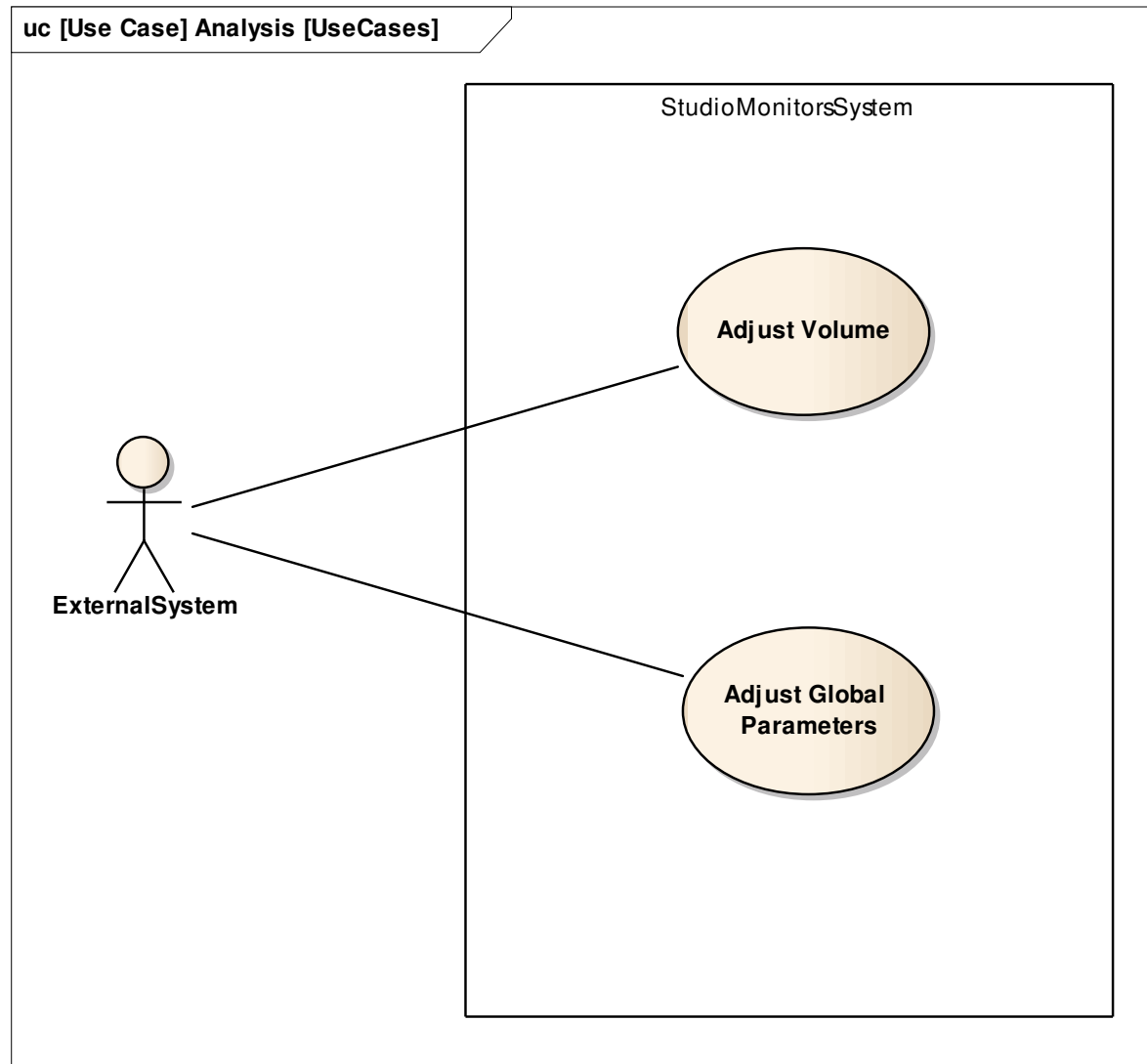
HW Design

- *Platform Model* for subsystems (Internal Block Diagram)
- *Deployment Model* (Allocation of logical model to platform)

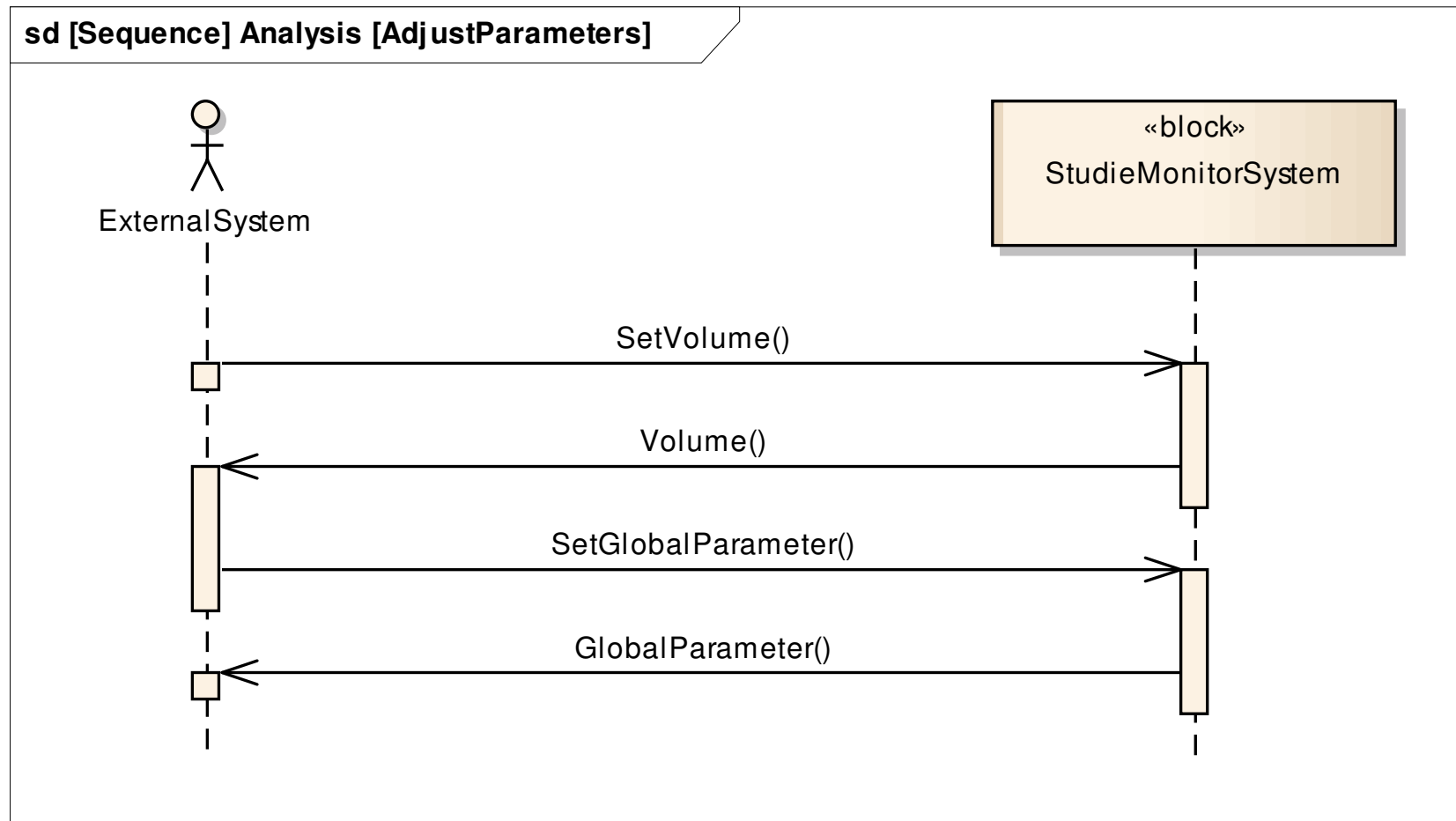
Interface Design

- Identification and specification between subsystems (HW+SW)

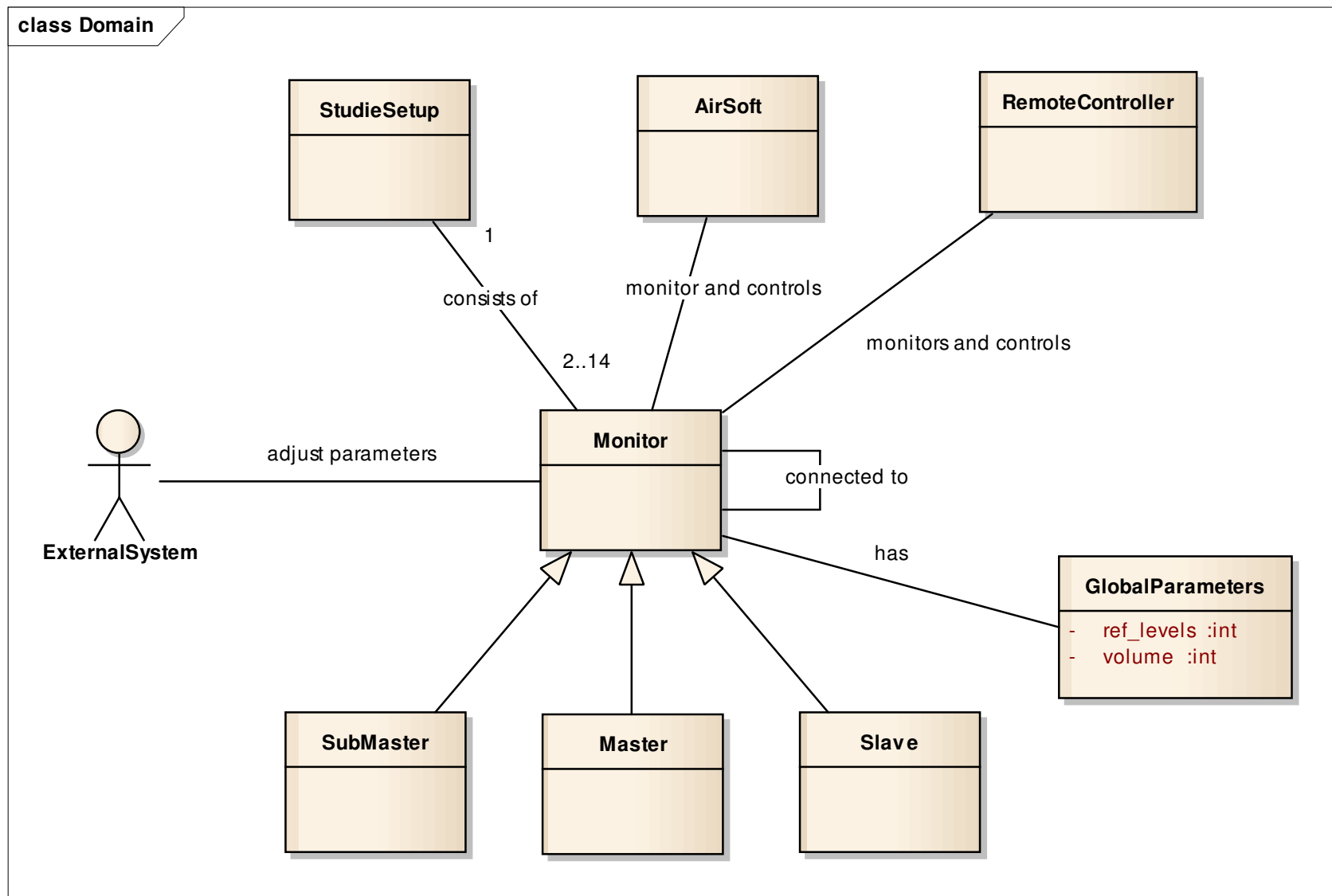
Analysis – Use Case



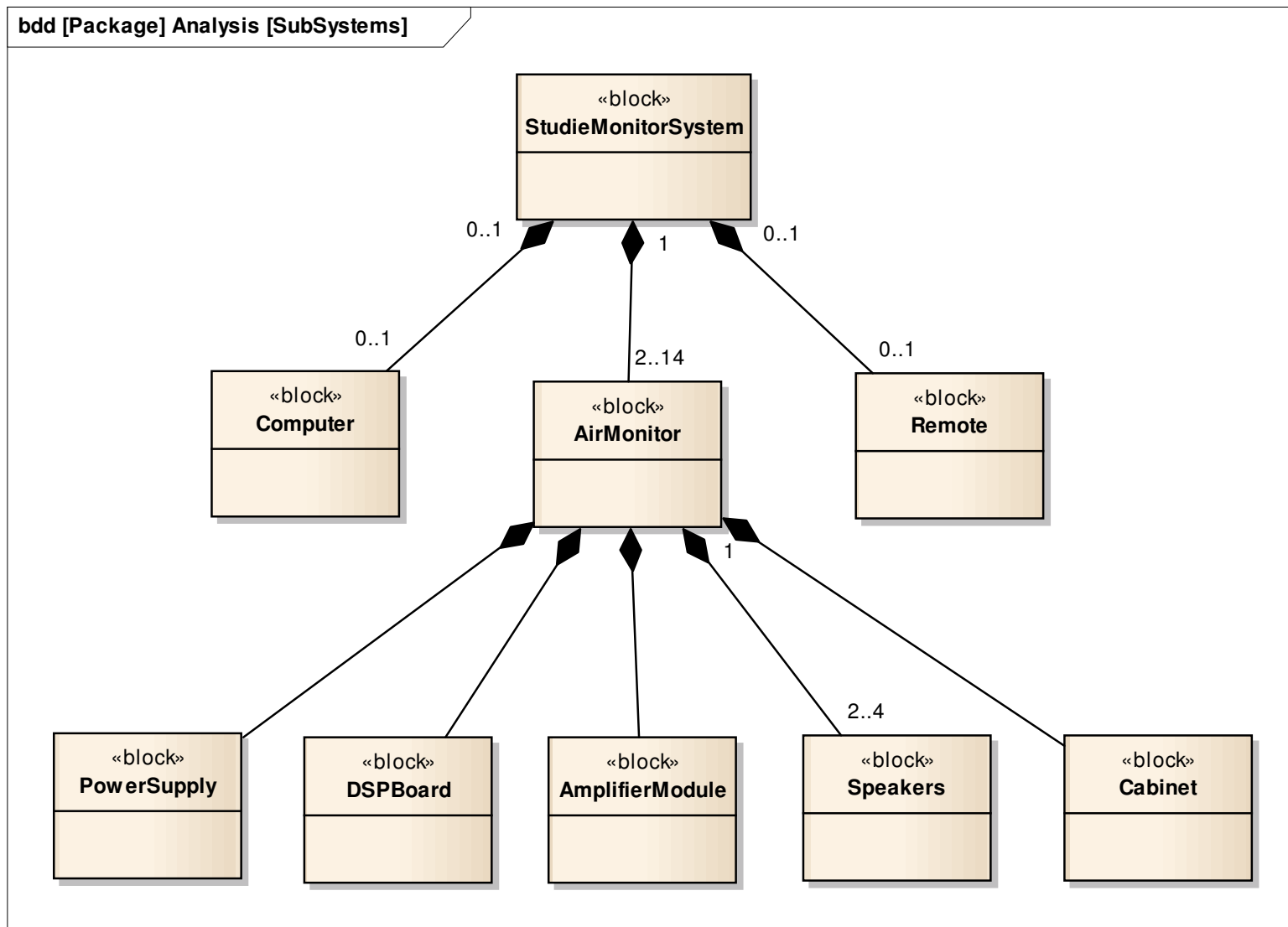
Analysis – Actor interaction



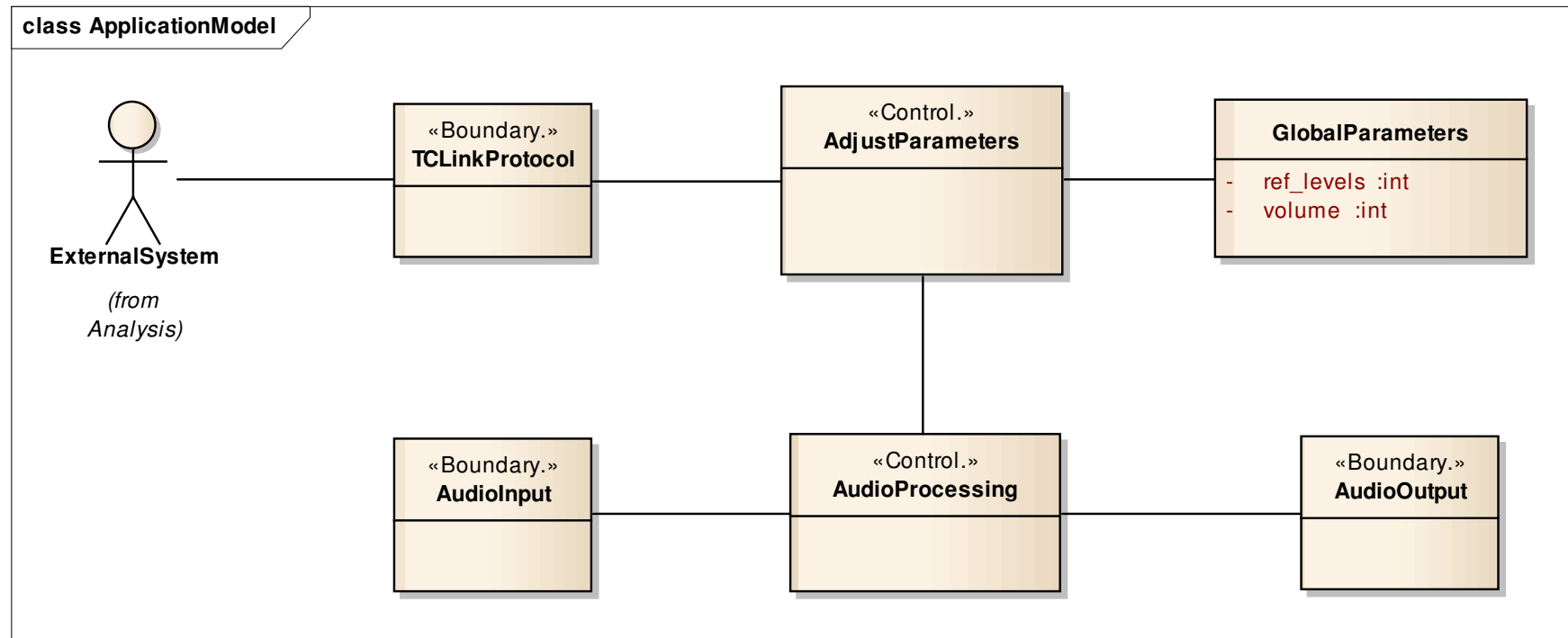
Analysis – Domain Model



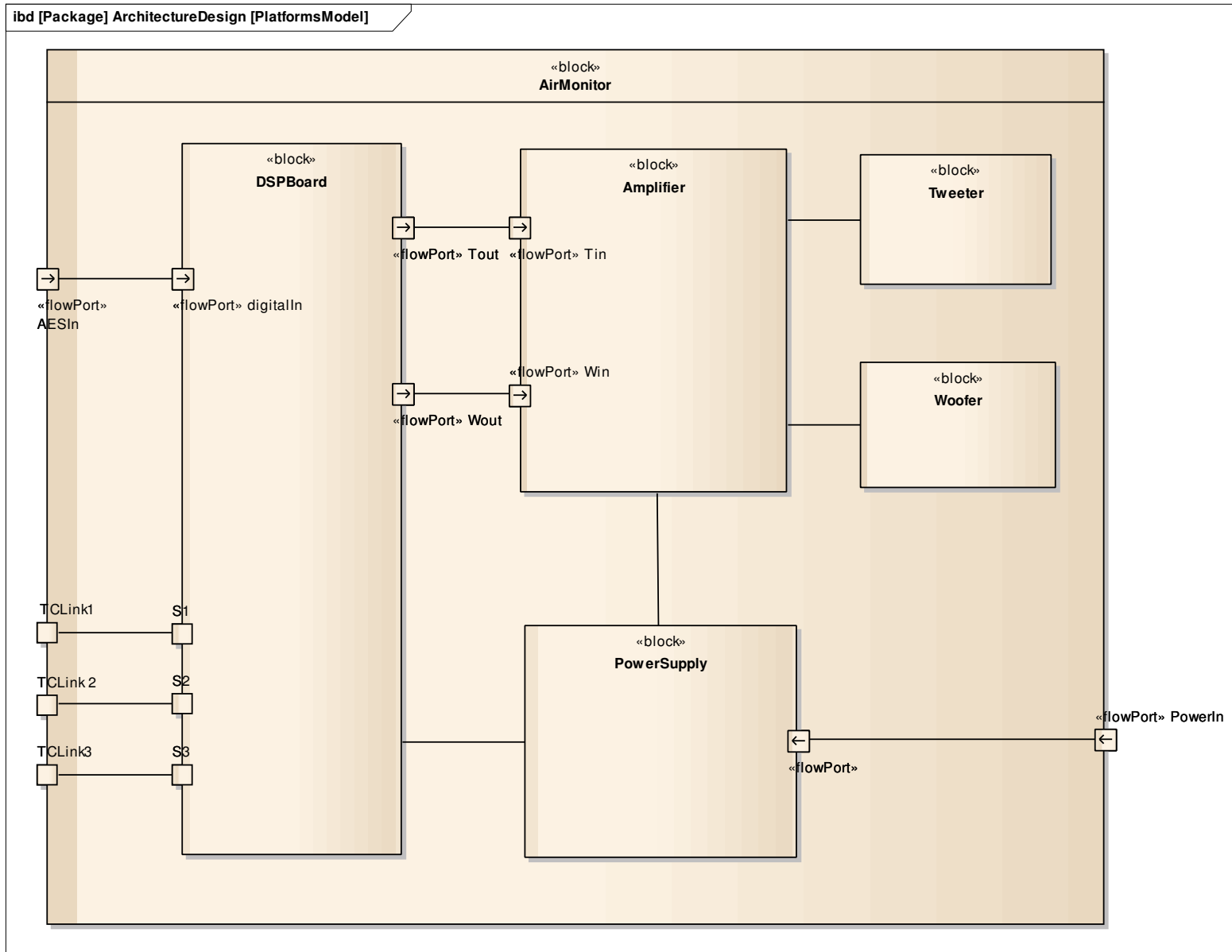
Architecture - Subsystems



SW Design - Application Model for AirMonitor (Subsystem)



HW Design – Platform Model for AirMonitor (Subsystem)



Eksempel

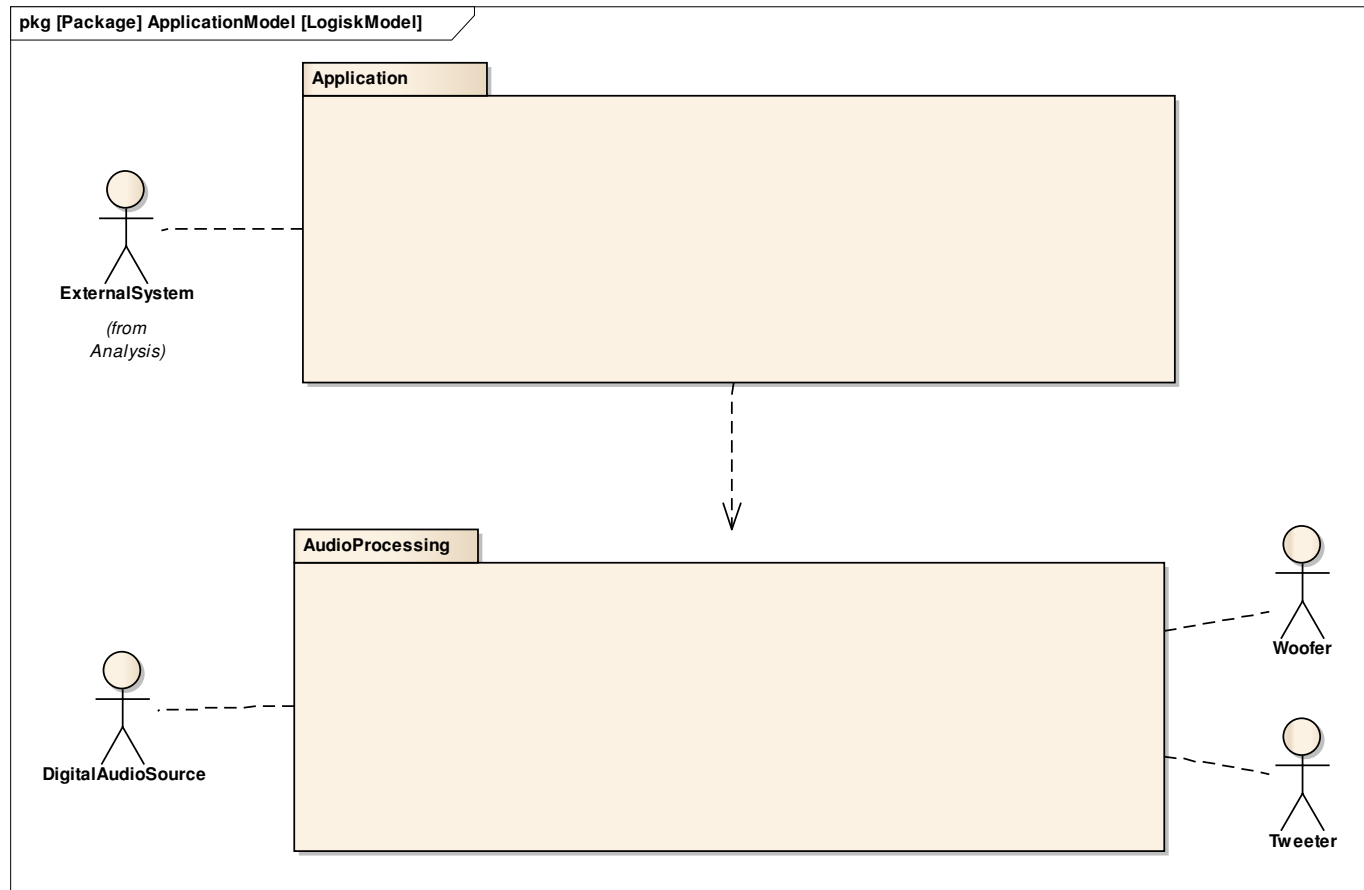
<http://staff.iha.dk/kbe/>

- Med udgangspunkt i ovenstående slides og nedenstående link ønskes skitseret et arkitektur design forslag til en AIR 6 højttaler.
<http://www.dynaudioacoustics.com/air-world.asp>

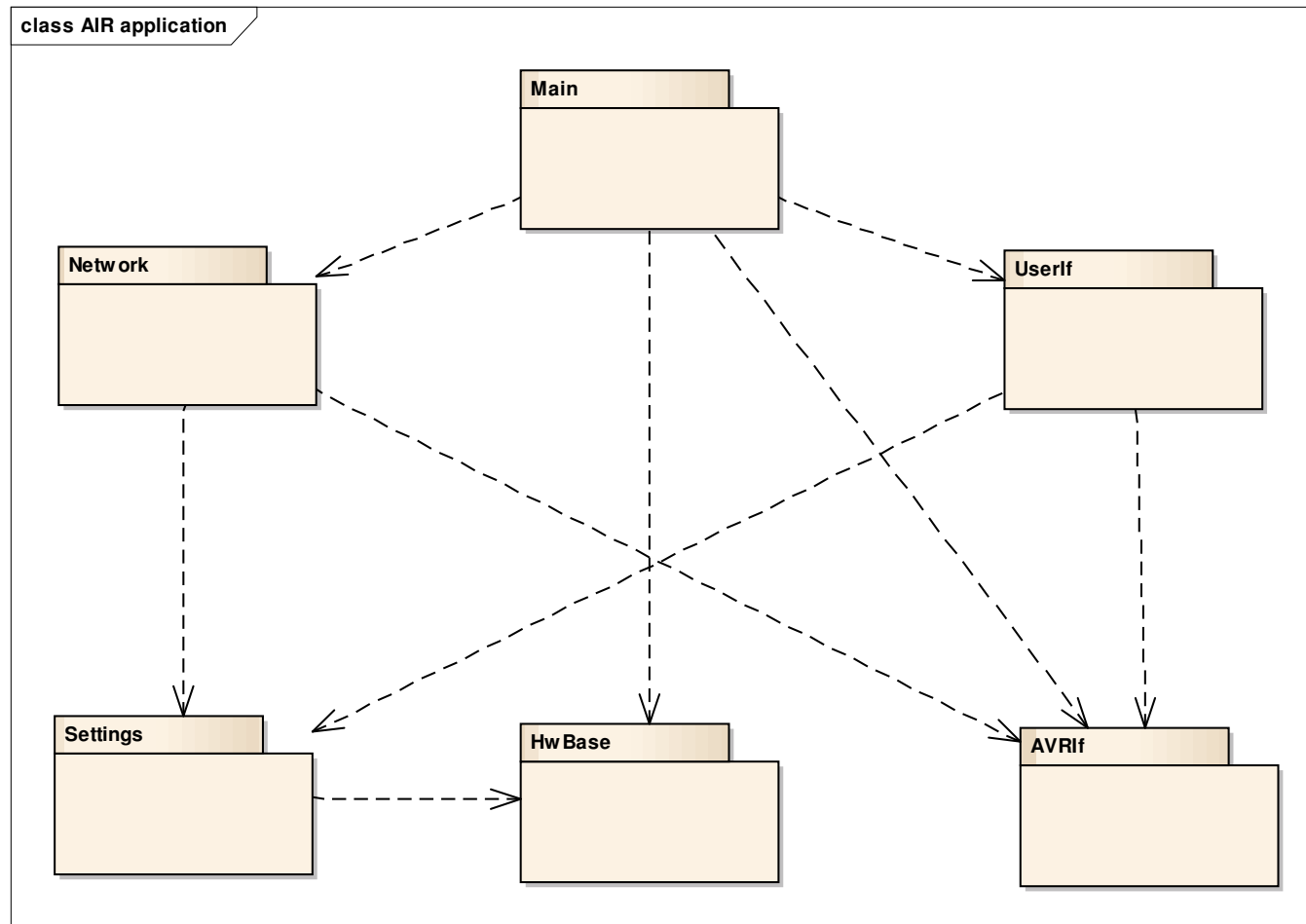
Brug SysML/UML til at beskrive arkitekturen:

1. Udarbejd en **logisk model** for en AirMonitor, med signalbehandling og kontrol software. Brug den todelte arkitekturmodel med brug af packages.
2. Udarbejd en **platforms model** for delsystemet DSPBoard incl. signalprocessor (DSP) og AVR (Display + TC Link).
3. Udarbejd en **deployment model** for AirMonitor højttaleren? (Brug SysML struktur diagrammer og "allocate" som for Miljøstyringssystemet, control klasser allocate til DSP og AVR)

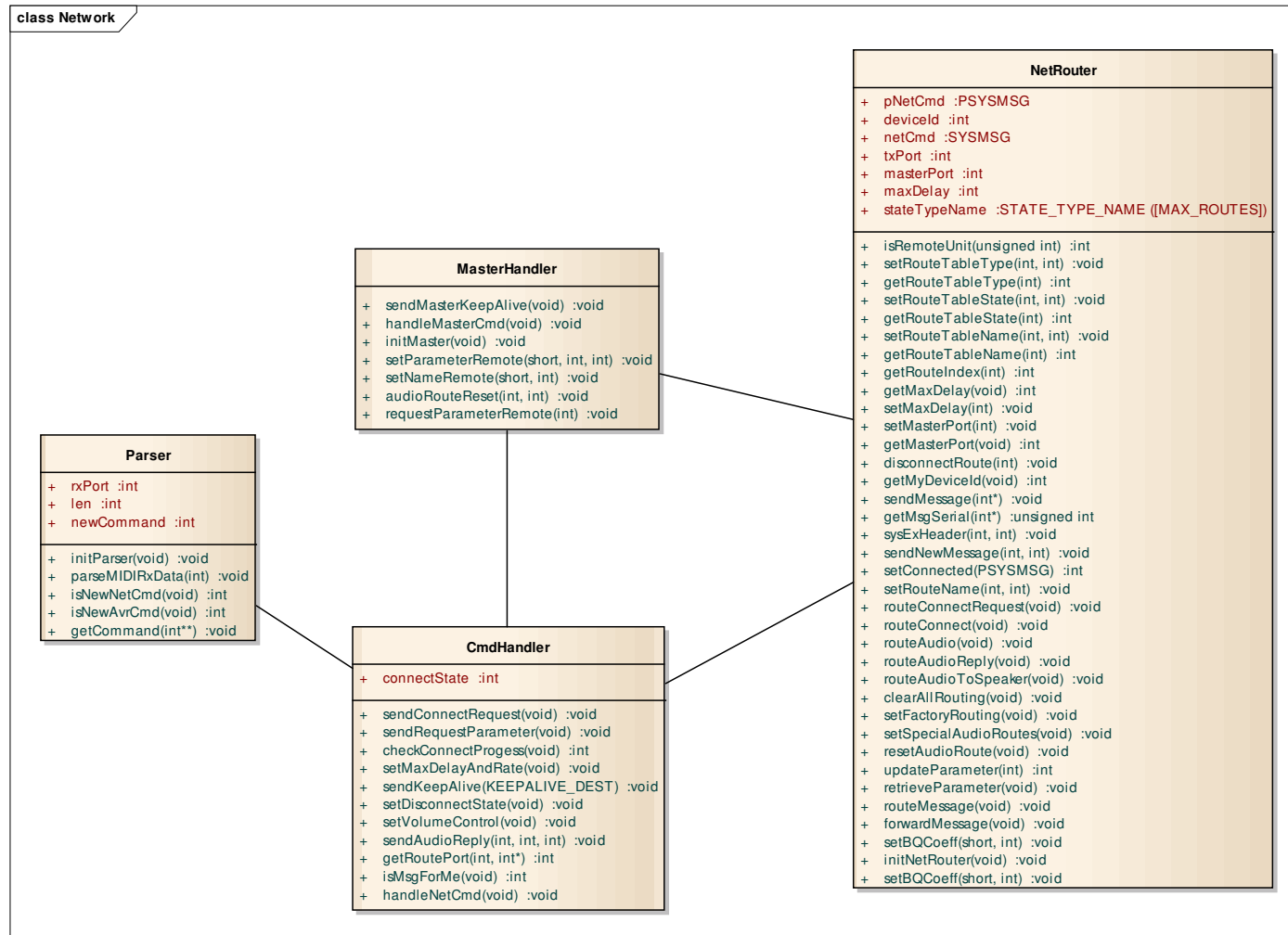
1 Logisk Model med todelt arkitekturmodel



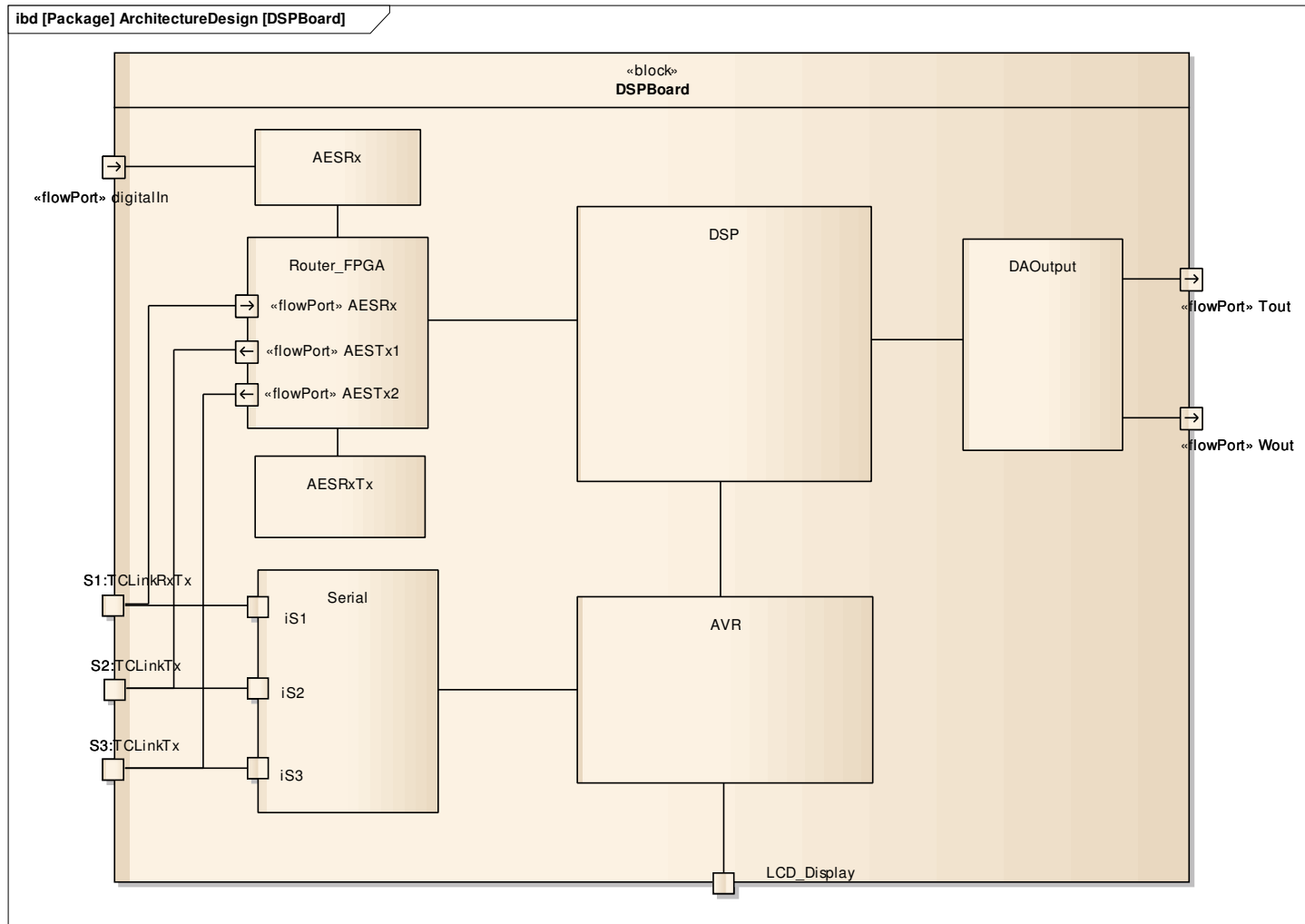
1 Indhold af logisk model – Application



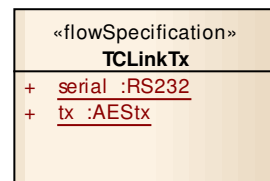
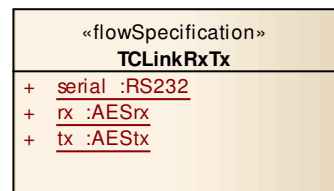
1 Indhold af application model - Network



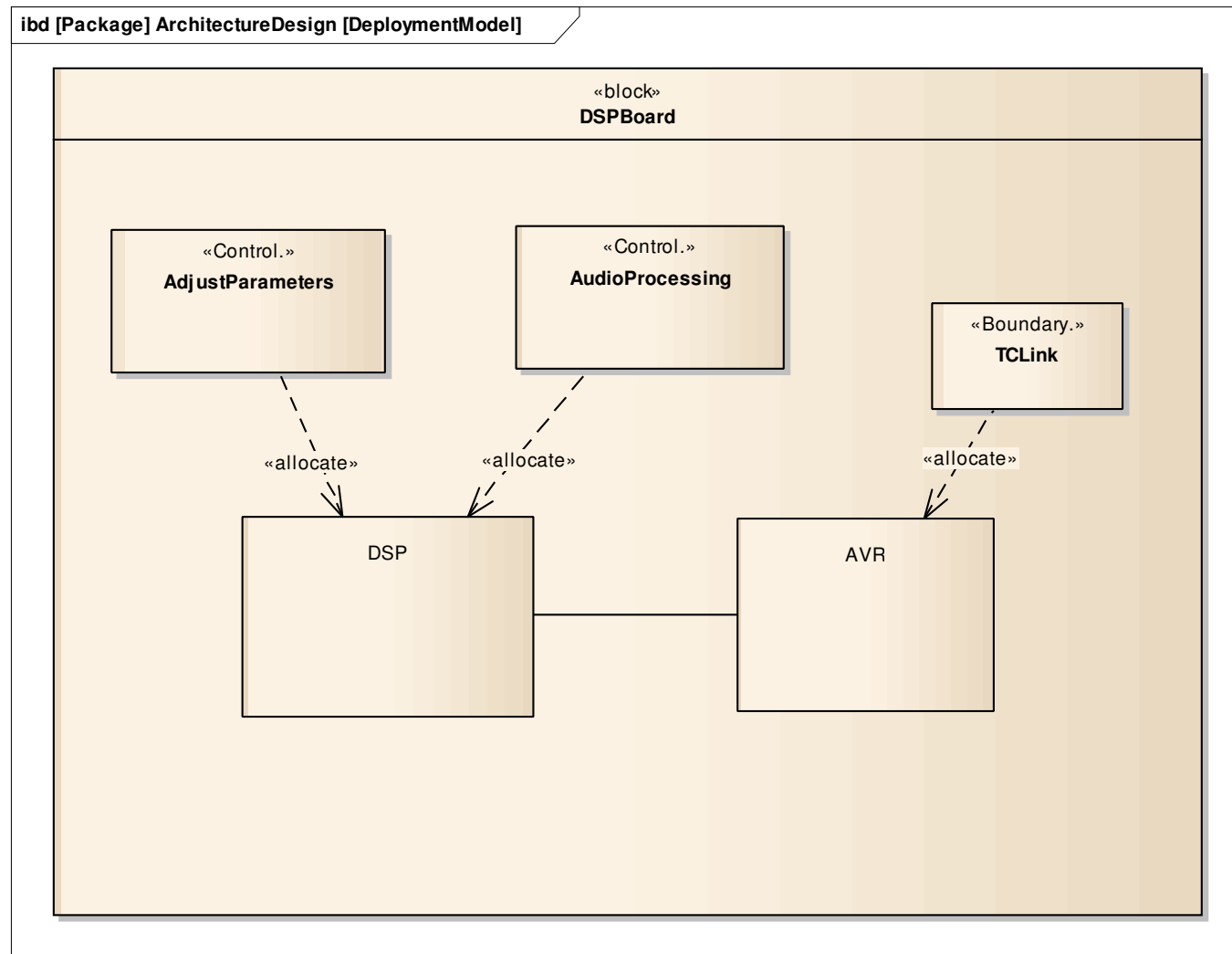
2 DSP Board – Platform Model



ibd [Package] ArchitectureDesign [DSPBoard]

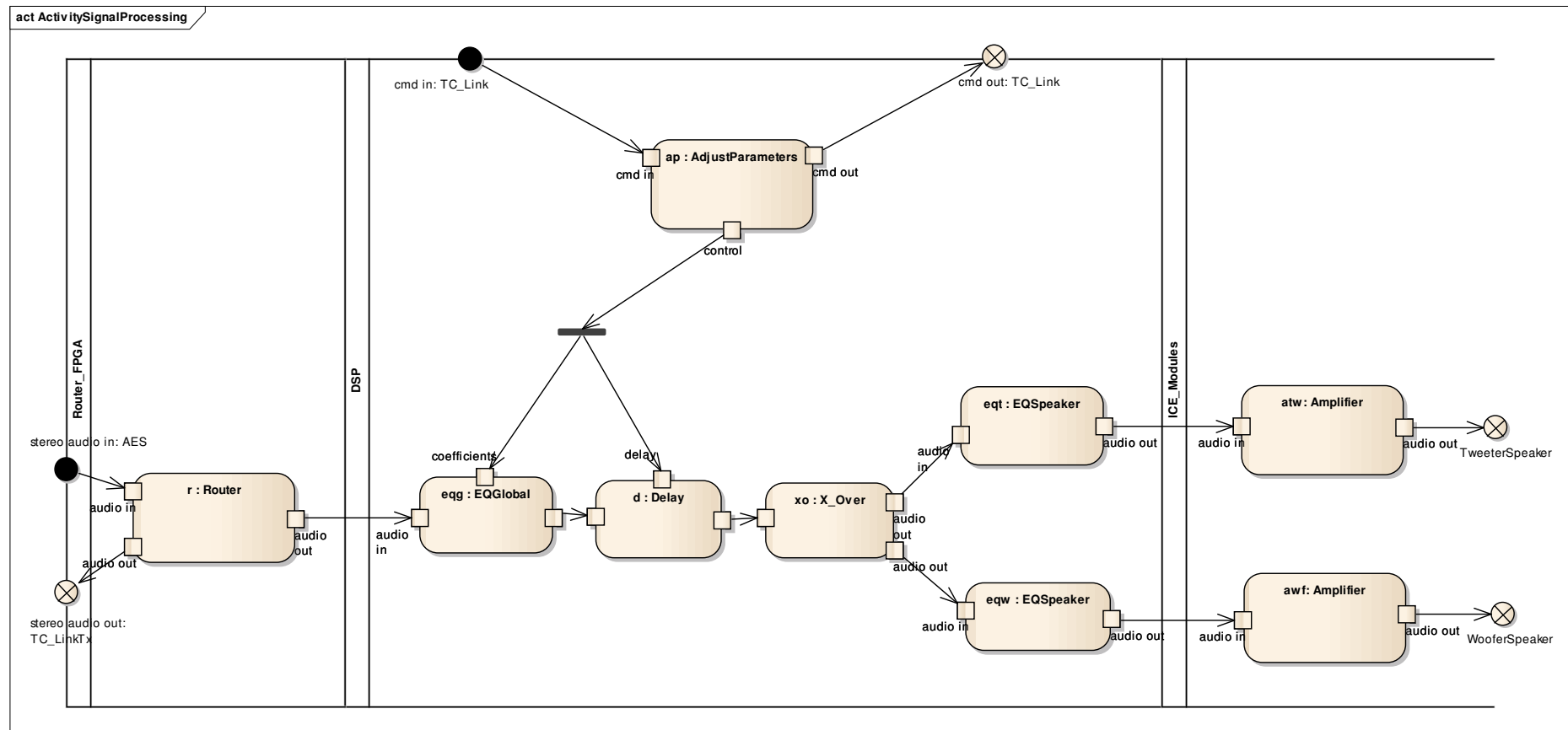


3 Deployment Model



SysML - Activity diagrams

Flow and Control, Partitioning



Opsummering

- Du har nu lært om delaktiviteterne i systemdesign
- Specielt har du lært teknikker og retningslinier for SW og HW arkitekturdesign, samt principperne for opdeling af systemet i delsystemer og komponenter