

# 1 Redegør for COM's arkitektur og terminologi samt forklar hvorledes man kan udvikle COM komponenter med brug af ATL.

- Forklar arkitekturen og terminologien
- Kommunikation på højt niveau
- C++ er inkapslet, men ikke binær
- Det er lidt ala. klippeboard funktionalitet, så forskellige programmer kunne snakke sammen.
- Binær indkapsling, så implementeringen er skjult, men kontrakter synlige i binær kode, så alle kan bruge dem
- Kan bruges på tværs af programmeringssprog, Tilpas simpelt til dem
- Brug virtuel metodetabel, som mapper via abstrakte kald
- C kan se det som et array a metodepointere
- Kan fysisk være .exe, men er normalt .dll
  - Har kun 4 simple metoder - class factory, registrer og fjern registrering, can-unload-now.
  - Returnere en pointer fra et interface til konkret implementering
  - Der kan være mange forskellige klasser, men det er kun interfacet man kan se
  - Man kan lave egne custom interface
  - Der findes mange, vi har kun set de 2 af dem. IUNKNOWN!!
    - \* Queue interface - kalder AddRef
    - \* AddRef - tæller en intern counter for, hvor mange der peger på dig.
    - \* Release - kaldes, når intern counter for referencer er 0.
    - \* Se 03b-Com-introduktion, slide 5 og 6
- ATL - værktøj til at lave bøvl
  - Der er en wizard (nogle få) - Templates og Libraries

## 1.1 ATL

- Active Template Library
- Meget ens kode, på nær nogle classes. Der bruges en wizard, hvor man sætter et par flueben
- *Apartment mode* - Simpel udgave, *Single* - kan integrere med Clipboard og andre programmer - skal være skrevet på samme mode.
- *Connections points* - C#'s events
- Tilføj funktioner via *Class View* og tilføj parameter og returtype
- Smart Pointers og så kalder "CCom pointer"
  - Det samme som en normal pointer, bare med flere funktioner
  - Pointer skal pege på interface
  - Husk at release
  - Kaldes *Auto pointer* i C++, så out-of-scope releaser
  - Import en .tlb-fil og få genereret .h-filen fra en producent