

A3 Poster – Volledige Thesis (Fortran77 • Fortran95 • Ada95 • DLL-Architecturen)

A3 POSTER - OVERZICHT VOLLEDIGE THESIS

Fortran 77 • Fortran 95 • Ada 95 • Moderne DLL-Architecturen

1. Centrale Stelling

Fortran 77 is ongeschikt voor moderne software-architecturen.

Fortran 95 en Ada 95 vormen de minimale, toekomstvaste basis voor veilige, modulaire en interoperabele DLL-gebaseerde systemen.

2. Evolutie van de Talen

Fortran 77 → Fortran 95 → Ada 95

- **Fortran 77:** monolithisch, geen modules, geen C-ABI, geen dynamisch geheugen.
 - **Fortran 95:** modules, intent-specificatie, allocatable arrays, C-interoperabiliteit.
 - **Ada 95:** extreem sterk type-systeem, packages, generics, veilige pointer-semantiek, C-ABI.
-

3. Kernproblemen van Fortran 77

- Geen modules → fragiele afhankelijkheden.
 - COMMON blocks → onzichtbare koppelingen.
 - Impliciete typing → stille fouten.
 - Geen dynamische arrays → beperkte schaalbaarheid.
 - Geen C-ABI → ongeschikt voor DLL's.
 - Geen struct-mapping → geen moderne interoperabiliteit.
-

4. Waarom Fortran 95 de Minimale Moderne Standaard is

- **Modules** voor structuur en veiligheid.
 - **Intent(in/out)** voorkomt 80% van klassieke fouten.
 - **Allocatable arrays** voor dynamiek.
 - **bind(C)** voor stabiele DLL-exports.
 - **Geen runtime-overhead** t.o.v. F77.
-

5. Waarom Ada 95 Ideaal is voor Kleinschalige DLL-Modules

- **Sterkste type-systeem** van alle klassieke talen.
 - **Packages** voor modulaire architectuur.
 - **pragma Export (C)** voor stabiele symbol-namen.
 - **Veilig geheugenmodel** zonder pointer-fouten.
 - Perfect voor logica, validatie, state-machines, wrappers.
-

6. Rol van C-ABI in Moderne Architecturen

C-ABI vormt de **centrale verbindingslaag**:

COBOL → C → Fortran95

COBOL → C → Ada95

C → ML64

- Stabiele symbol-namen
 - Voorspelbare calling convention
 - Universele interoperabiliteit
-

7. ML64-Fastpaths

- Inzetbaar voor performance-kritieke kernels.
- Windows x64 ABI: RCX, RDX, R8, R9 + XMM0-3.
- Ideaal voor vector-loops, dot-products, bulk-berekeningen.
- Wordt aangeroepen vanuit Fortran 95 via bind(C).

8. Struct-Mapping (C ↔ Fortran95 ↔ Ada95)

C-struct → Fortran95 type, bind(C) → Ada95 record, Convention(C)

- 1-op-1 memory-layout.
 - Geen padding-problemen.
 - Volledig DLL-compatibel.
-

9. Vergelijkingstabel

Eigenschap	F77	F95	Ada95
Modulariteit	✗	✓	✓✓
Type-veiligheid	✗	✓	✓✓✓
C-interoperabiliteit	✗	✓	✓
Struct-mapping	✗	✓	✓✓
Pointer-veiligheid	✗	✗	✓✓✓
Numerieke performance	✓	✓	✓
Geschikt voor DLL's	✗	✓✓	✓✓

10. Aanbevolen Architectuur

Numerieke kern → Fortran 95

- vector-berekeningen
- matrix-operaties
- financiële en wetenschappelijke kernels

Logische kern → Ada 95

- validatie
- configuratie
- state-machines
- foutafhandeling

Performance-kern → ML64

- tight loops
- SIMD-operaties
- bulk-processing

Interfacelaag → C-ABI

- stabiele DLL-exports
 - universele interoperabiliteit
-

11. Conclusie van de Volledige Thesis

- **Fortran 77** voldoet niet aan moderne eisen.
- **Fortran 95** is de minimale moderne standaard voor numerieke interoperabiliteit.
- **Ada 95** is ideaal voor veilige, logische en structurele componenten.
- **C-ABI** is de verbindende laag voor alle talen.
- **ML64** levert maximale performance waar nodig.

Samen vormen deze talen een **krachtige, modulaire en toekomstvaste DLL-architectuur** die geschikt is voor zowel kleinschalige als middelgrote systemen.

Einde van de A3-Poster