

Cross-Language Architectuurdocument

Cross-Language Architectuurdocument

1. Overzicht

Dit document beschrijft de architectuur voor interoperabiliteit tussen Rocket COBOL 11.0, IFX Fortran, ML64-assembler en Intel C. De kern van de architectuur is een gedeelde C-ABI die door alle talen wordt ondersteund.

2. Hoog-niveau Architectuur

```
+-----+ +-----+
+-----+ | | ML64 DLL | +-----+ C-ABI | (sort8, etc.) |
+-----+ +-----+
+-----+ bind(C) | IFX Fortran +-----^ | (numeric code) | +-----+
```

3. ABI-Contract

3.1 Naamgeving

- Exportfuncties gebruiken unmangled C-namen.

3.2 Calling Convention (Microsoft x64)

- RCX = 1e argument
- RDX = 2e argument
- R8 = 3e argument
- R9 = 4e argument
- Stack is 16-byte aligned

4. Datacontract: PIC X(8) Tabel

4.1 COBOL-definitie

01 TABLE-SIZE PIC 9(9). 01 SORT-TABLE. 05 ELEM PIC X(8) OCCURS 1000 TIMES.

4.2 Geheugenlayout

```
+-----+-----+-----+ ... | elem1 | elem2 | elem3 | elem4 | +-----+-----+-----+-----+ 8
B 8 B 8 B 8 B
```

5. Taalinterfaces

5.1 ML64 (pseudo-assembler)

PUBLIC sort8

; RCX = pointer to bytes ; RDX = element count

sort8 PROC ; Implementatie ret sort8 ENDP

5.2 Intel C

extern "C" __declspec(dllimport) void sort8(char *arr, int n);

5.3 IFX Fortran

subroutine sort8(arr, n) bind(C, name="sort8") use iso_c_binding character(kind=c_char),
dimension(*) :: arr integer(c_int), value :: n end subroutine

5.4 Rocket COBOL 11.0

CALL "sort8" USING BY REFERENCE SORT-TABLE BY VALUE TABLE-SIZE.

6. Laag-indeling

[Application Layer]

- COBOL main programs
- C tools / test harnesses
- Fortran scientific drivers

[Interop Layer]

- C headers (prototypes)
- Fortran bind(C) interfaces

- COBOL CALL contracts

[Core Performance Layer]

- ML64 DLL (sort, crypto, bitops)
- Fortran DLL's met bind(C)

7. Versiebeheer & Deployment

- DLL's vormen het centrale versiepunt.
- Talen hoeven niet opnieuw gecompileerd te worden zolang de ABI gelijk blijft.

8. Best Practices

- Eén centrale C-header als bron voor alle talen.
- Geen taal-specifieke types in de ABI.
- Alleen C-achtige types: pointers, integers, vaste buffers.
- Unit-tests in C voor snelle feedback.

Einde van document.