Ambientes Virtuais de Execução

Apresentação da plataforma .Net

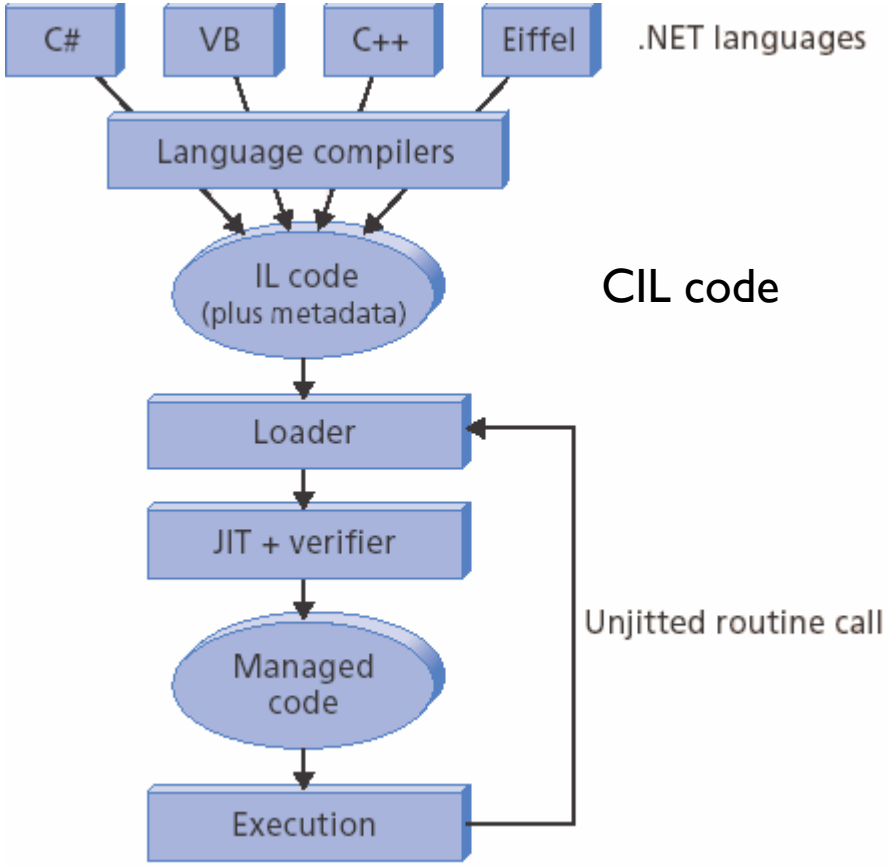
Plataforma .Net

- Consiste de:
  - Common Language Runtime (CLR)
    - Fornece uma camada de abstracção sobre o sistema operativo
  - Bibliotecas de classes base
  - Plataformas de desenvolvimento e tecnologia

.Net Framework Stack

Common Language Runtime (CLR)

- CLR é o ambiente de execução da plataforma .NET.
  - Todos os programas .NET executam sob a supervisão do CLR.
  - Garante propriedades na área de gestão de memória, segurança e manipulação de excepções.
- Todas as linguagens compatíveis compilam para CIL (Common Intermediate Language), uma linguagem intermédia.



Common Language Runtime (CLR)

- CLR implementa a Common Language Infrastructure(CLI)
  - A CLI é definida pelos standards ECMA-335 e ISO/IEC 23271:2003.
- A especificação da CLI descreve os seguintes aspectos:
  - O sistema de tipos - Common Type System (CTS)
  - Metadata
  - A linguagem comum de especificação - Common Language Specification (CLS)
  - O sistema de execução virtual – Virtual Execution System (VES)

Common Language Infrastructure(CLI)

- Common Type System (CTS)
  - O conjunto dos tipos de dados e operações que são partilhados por todas as linguagens de programação que podem ser executadas por uma implementação de CLI, tal como a CLR.
- Metadata
  - Informação sobre os tipos presentes na representação intermédia
- Common Language Specification (CLS)
  - Um conjunto de regras básicas que as linguagens compatíveis com a CLI devem respeitar para serem interoperáveis entre si. As regras CLS definem um subconjunto do Common Type System.
- Virtual Execution System (VES)
  - O VES carrega e executa programas CLI-compatíveis, utilizando a metadata para combinar separadamente as partes em tempo de execução.

Características da plataforma .Net

- Portabilidade
  - Representação intermédia
- Interoperabilidade entre linguagens
  - Sistema de tipos comum
- Serviços de execução
  - Gestão de memória
  - Segurança de tipos e controlo de acessos
- Funcionalidades
  - Biblioteca rica para desenvolvimento de diferentes tipos de componentes/aplicações

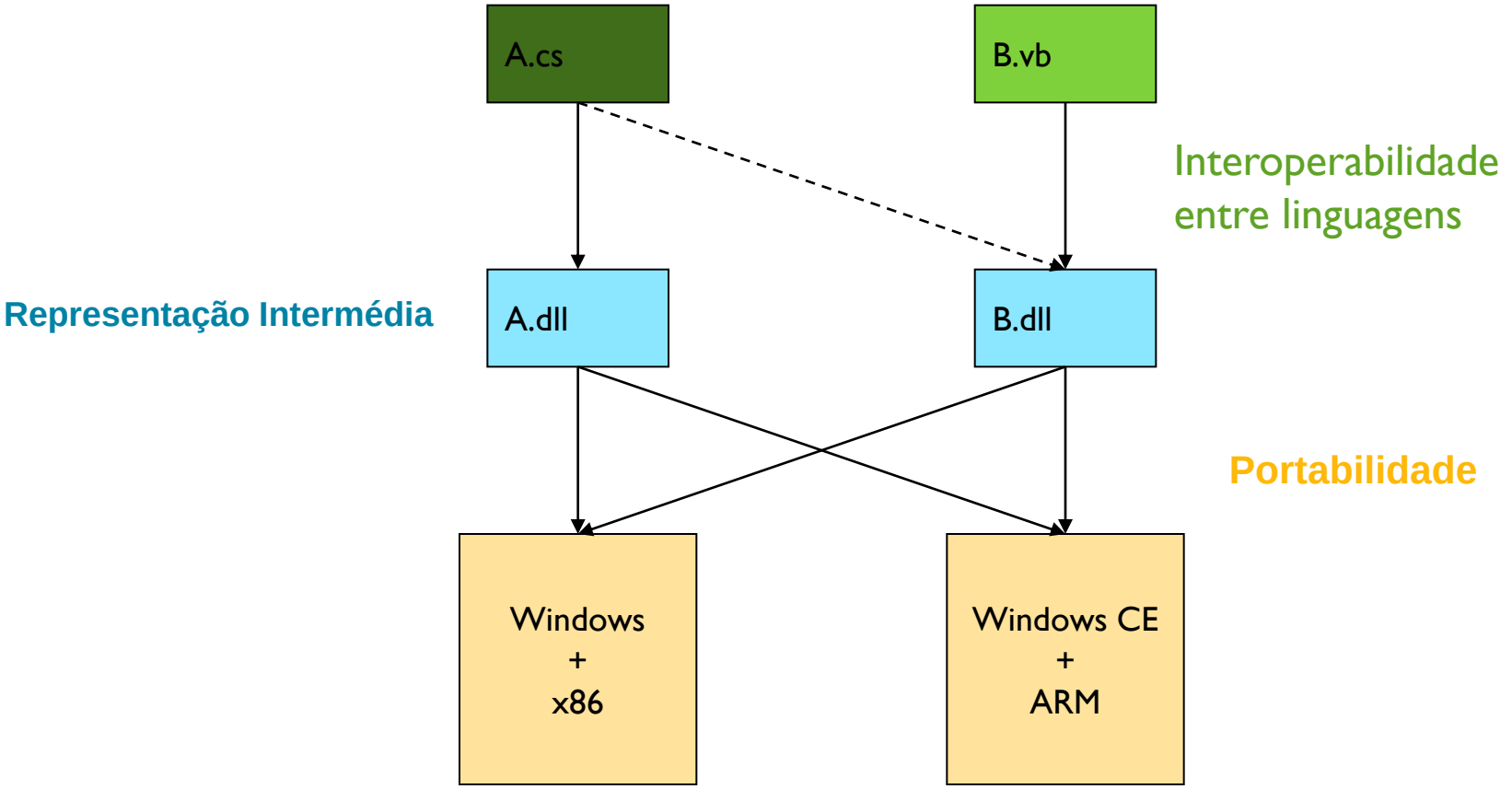
Portabilidade

- A capacidade de execução em diferentes plataformas (x86+Windows; x86+Linux; ARM+WindowsMobile;...) é conseguida através de:
  - Representação intermédia
    - Independente da arquitectura do processador
    - Mais compacta e fácil de analisar (automaticamente) que o código fonte
  - Ambiente de execução virtual
    - Tradução da representação intermédia para representação nativa
    - Abstracção do ambiente de execução fornecido pelo OS
  - Biblioteca de classes
    - Conjunto de funcionalidades fornecidas de forma independente do sistema operativo

Interoperabilidade

- Capacidade de utilização de componentes de *software* realizados em linguagens diferentes
  - Exemplo:
    - Realizar a classe X na linguagem A
    - Utilizar a classe X na linguagem B
    - Realizar, na linguagem C, a classe Y que deriva de X
- É conseguido através:
  - Linguagem intermédia com sistema de tipos independente da linguagem

Independência de linguagens



Serviços de *runtime*

- O ambiente virtual de execução fornece serviços de *runtime* às aplicações, por exemplo:
  - Gestão automática de memória ("garbage collection") – detecção e recolha de objectos não utilizados
  - Serialização ("serialization") - conversão de grafos de objectos em sequências de *bytes*
  - Segurança – "type safety" e controlo de acessos
    - Verificação de "type safety" em tempo de execução
    - Controle de acesso baseado na identidade do utilizador ou do código
- Estes serviços estão disponíveis porque existe informação de tipos (METADATA) em tempo de execução