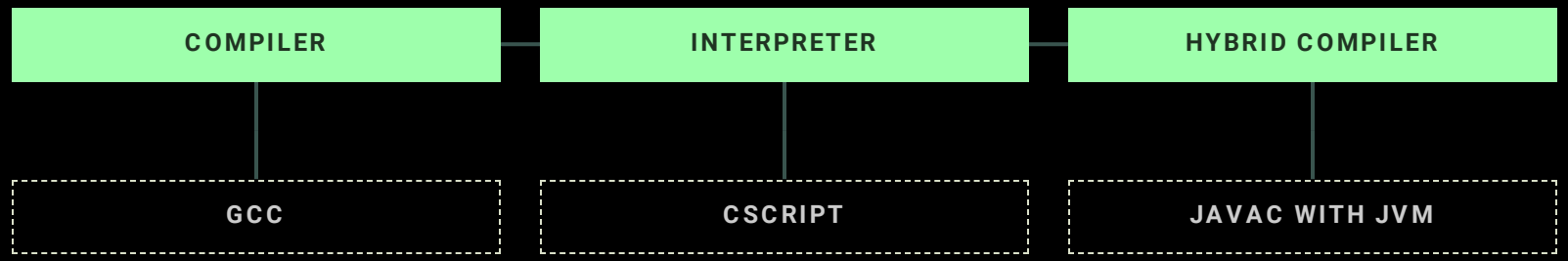
Compilers and Interpreters

Why, tho?

The type of language processor influences what tools we use.

It is a prerequisite to understand how to "reverse" the process aka **reverse engineering**

Types of Language Processors



Compiler

Hybrid compiler - at Compiletime

Hybrid - at Runtime

Interpreted - no compilation

SCRIPT

JSCRIPT

Interpreted - at Runtime

Tools of choice

MACHINE CODE

- loss of information
- decompilation is difficult
- disassembly is more accurate
- main tool: disassembler and decompiler side-by-side
  e.g.: IDA, Ghidra, Cutter

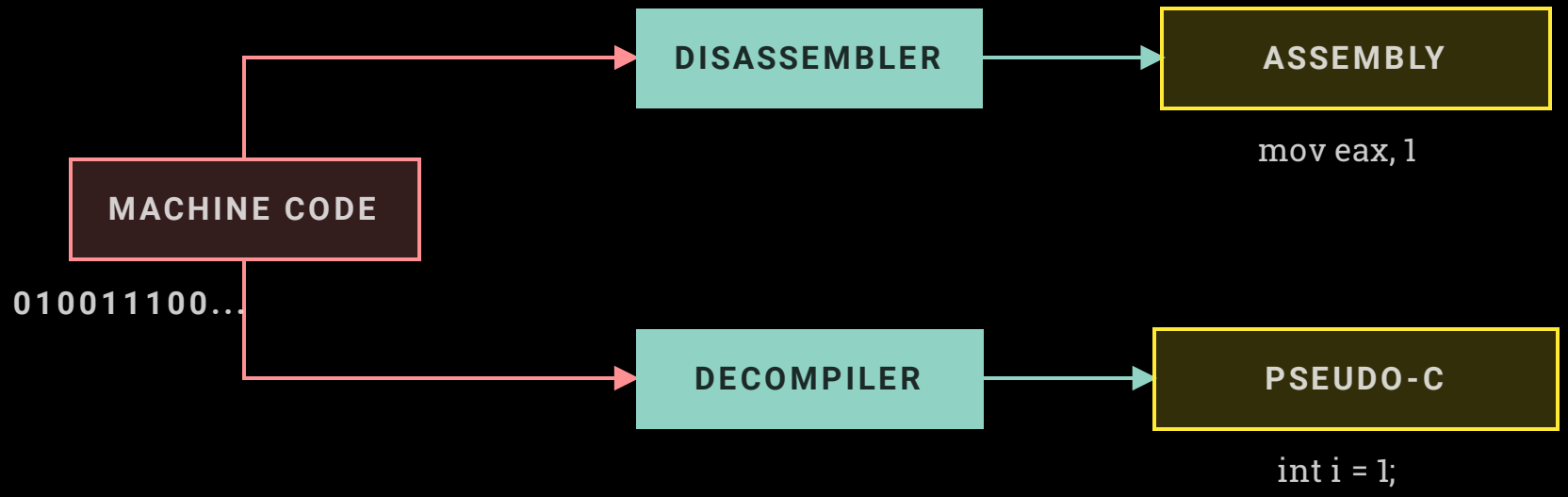
BYTECODE

- preserves lots of information
- decompilation usually works well
- disassembly only if neccessary
- main tool: decompiler
  e.g. DnSpy, Krakatau, uncompyle

ORIGINAL SOURCE

- main tool: text editor or IDE
  e.g. Notepad++, Visual Studio Code

Analysis of compiled code



Tools of choice

MACHINE CODE

- loss of information
- decompilation is difficult
- disassembly is more accurate
- main tool: disassembler and decompiler side-by-side
  e.g.: IDA, Ghidra, Cutter

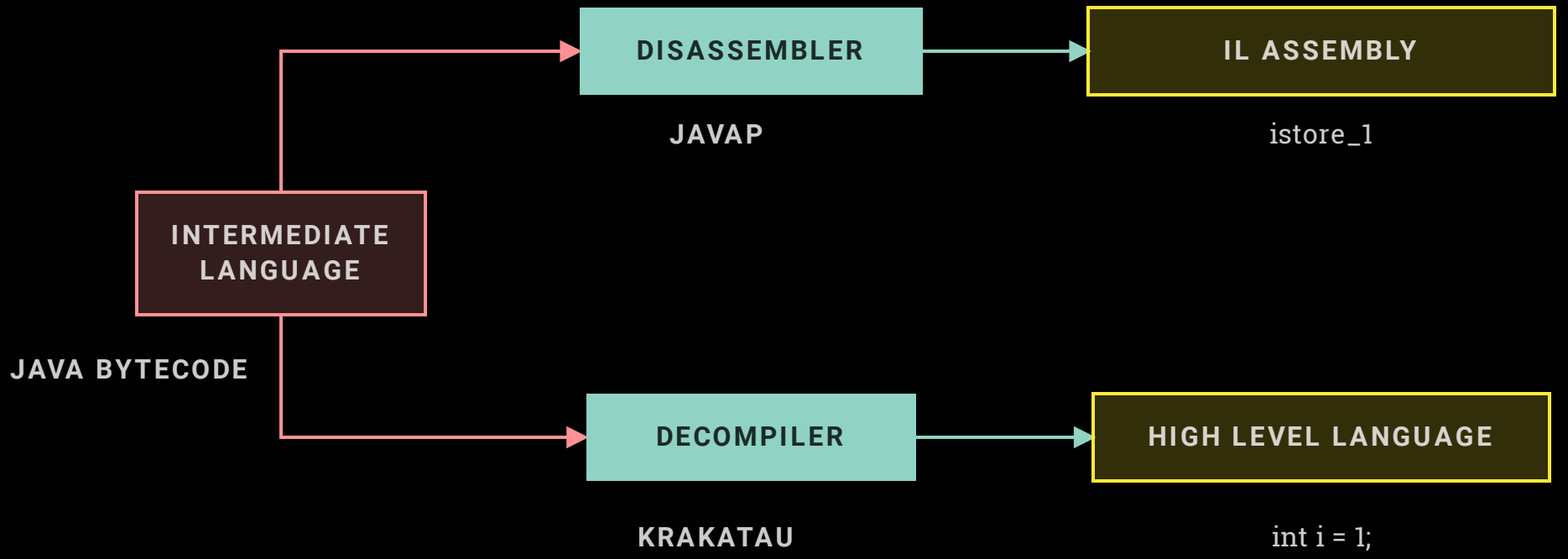
BYTECODE

- preserves lots of information
- decompilation usually works well
- disassembly only if neccessary
- main tool: decompiler
  e.g. DnSpy, Krakatau, uncompyle

ORIGINAL SOURCE

- main tool: text editor or IDE
  e.g. Notepad++, Visual Studio Code

Analysis of bytecode



Tools of choice

MACHINE CODE

- loss of information
- decompilation is difficult
- disassembly is more accurate
- main tool: disassembler and decompiler side-by-side
  e.g.: IDA, Ghidra, Cutter

BYTECODE

- preserves lots of information
- decompilation usually works well
- disassembly only if neccessary
- main tool: decompiler
  e.g. DnSpy, Krakatau, uncompyle

ORIGINAL SOURCE

- main tool: text editor or IDE
  e.g. Notepad++, Visual Studio Code

How do I know what I have there?

Triage will tell you

Typical Misconceptions / Myths

~~"LANGUAGES ARE COMPILED OR INTERPRETED~~"

Compiled/interpreted/hybrid are not characteristics of a programming language but of the language implementation.

Typical Misconceptions / Myths

~~"LANGUAGES ARE COMPILED OR INTERPRETED~~"

Anyone can write a compiler or an interpreter for any language

E.g., there is a machine code compiler for Java: gcj

As malware analyst you will also see non-standard language implementations

Typical Misconceptions / Myths

~~"PYTHON IS INTERPRETED"~~

The most common implementation of Python is CPython. The source is compiled to Python bytecode --> .pyc

Most of the time you will deal with the CPython bytecode when analysing Python based malware.