



TRACK 5

MyTechnology



SOFTWARE LANGUAGE OVER THE LAST 50 YEARS: WHAT WILL BE NEXT?

Pieter Zuliani - Joachim Jansen
COO Pegus Digital - C++ Lead developer

Pieter.zuliani@pegus.digital - Joachim.jansen@pegus.digital

INNOVATION
DAY 18.10.2019

CONFERENCE BY MASTERS IN INNOVATION
ON NEW PRODUCTS AND BUSINESS INNOVATION

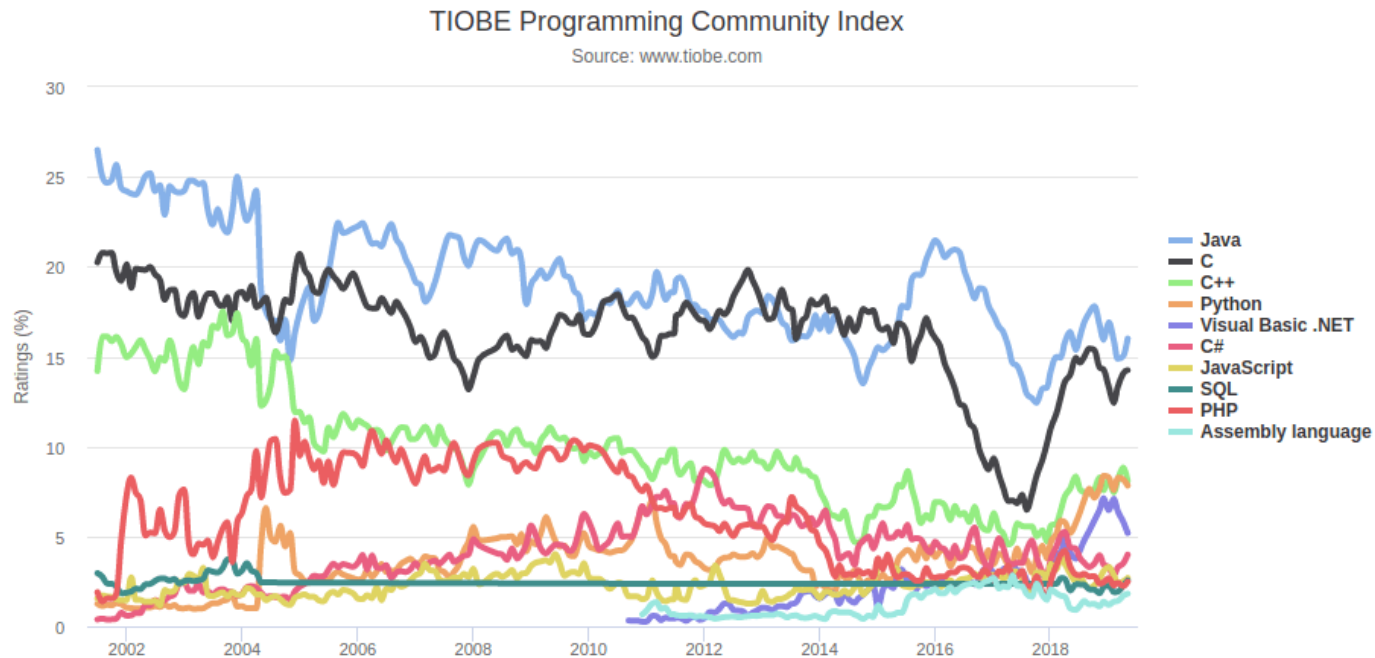


CONTENT

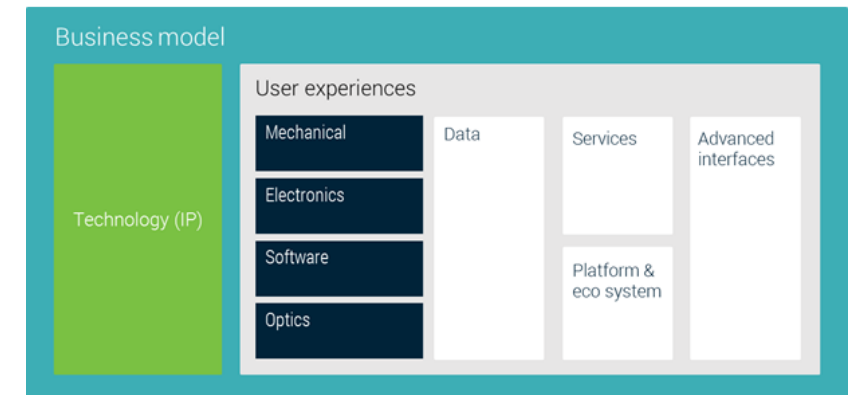
- 1 Intro – Programming language influence on integrated product development
- 2 What is a programming language
- 3 History of programming languages
- 4 Current usage statistics of programming languages
- 5 Pegus Digital Use Cases – Why did we choose a certain technology stack?
- 6 The future of programming languages
- 7 Conclusion

PROGRAMMING LANGUAGES - INFLUENCE ON INTEGRATED PRODUCT DEVELOPMENT

INFLUENCE ON INTEGRATED PRODUCT DEVELOPMENT

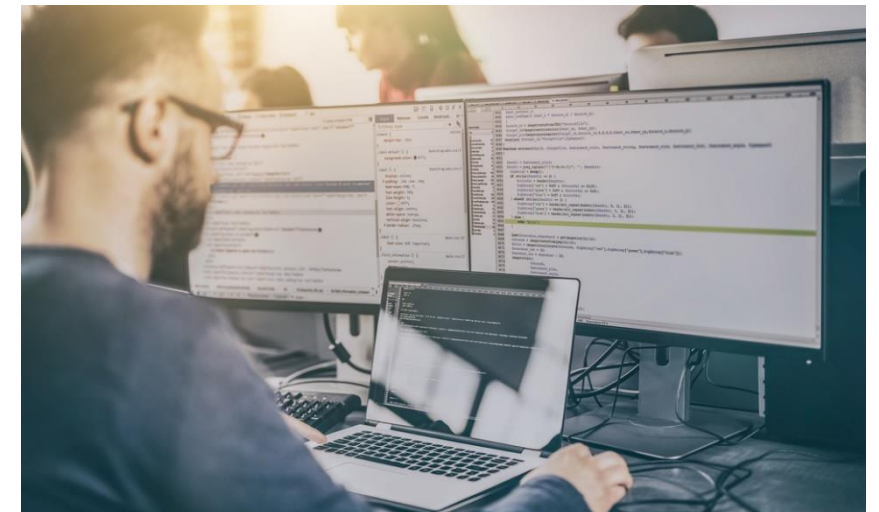
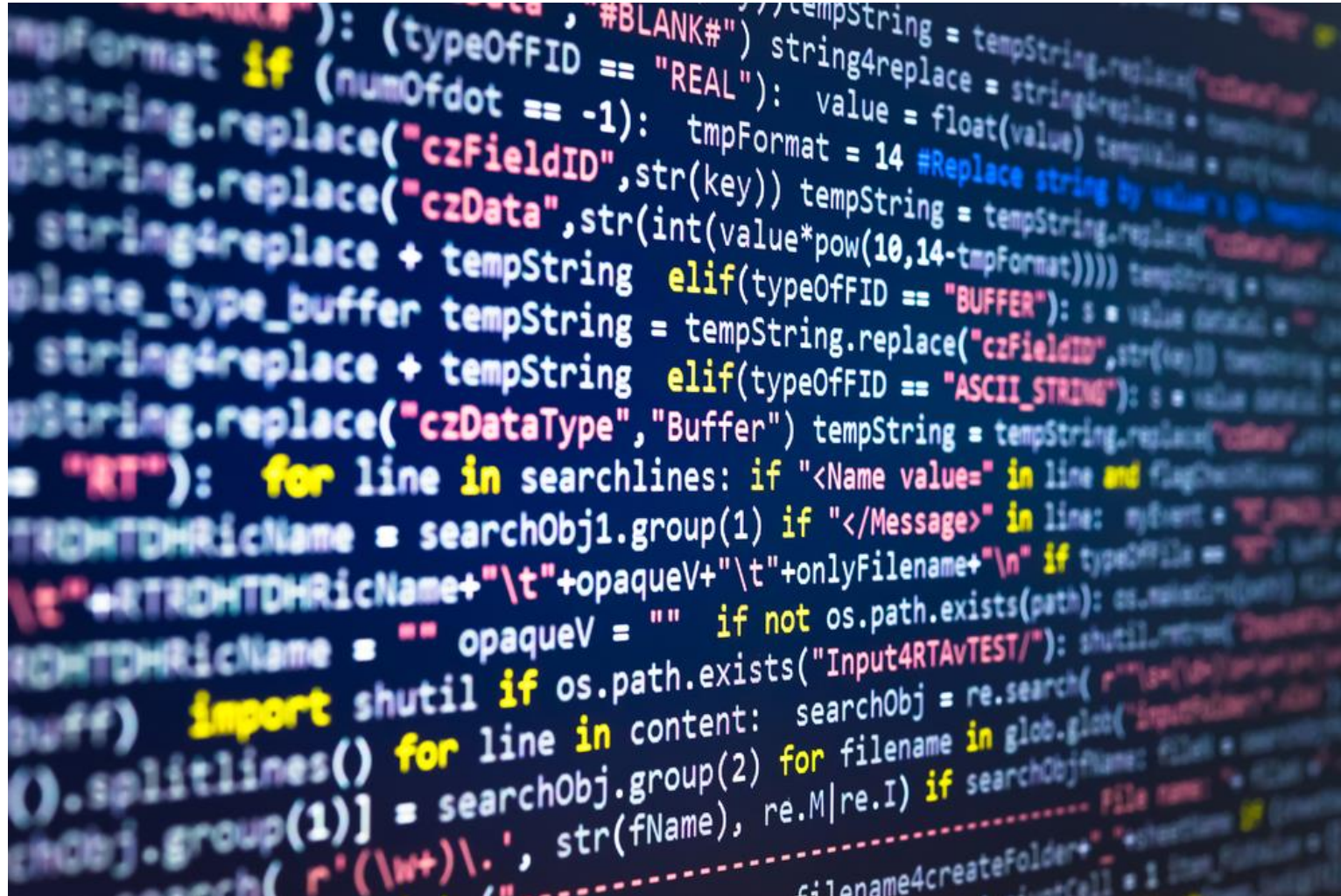


MyFutureProduct



WHAT IS A PROGRAMMING LANGUAGE

WHAT IS A PROGRAMMING LANGUAGE



WHAT IS A PROGRAMMING LANGUAGE

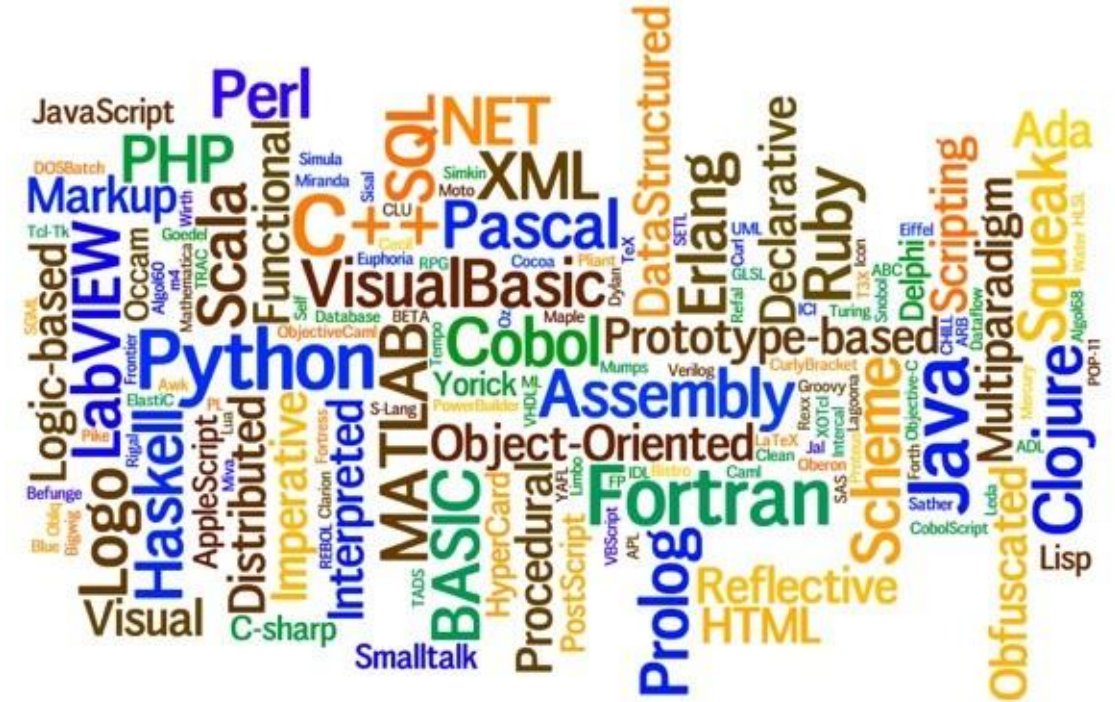
Programming Language

=

Tool to tell the computer what to do

WHAT IS A PROGRAMMING LANGUAGE

- Formal language
- Syntax (symbols) + Semantics (meaning)
- Set of instructions
- Various kinds of outputs
- Many different programming languages



WHAT IS A PROGRAMMING LANGUAGE

```
#include <iostream>

int main() {
    std::cout << "Hello World";
}
```



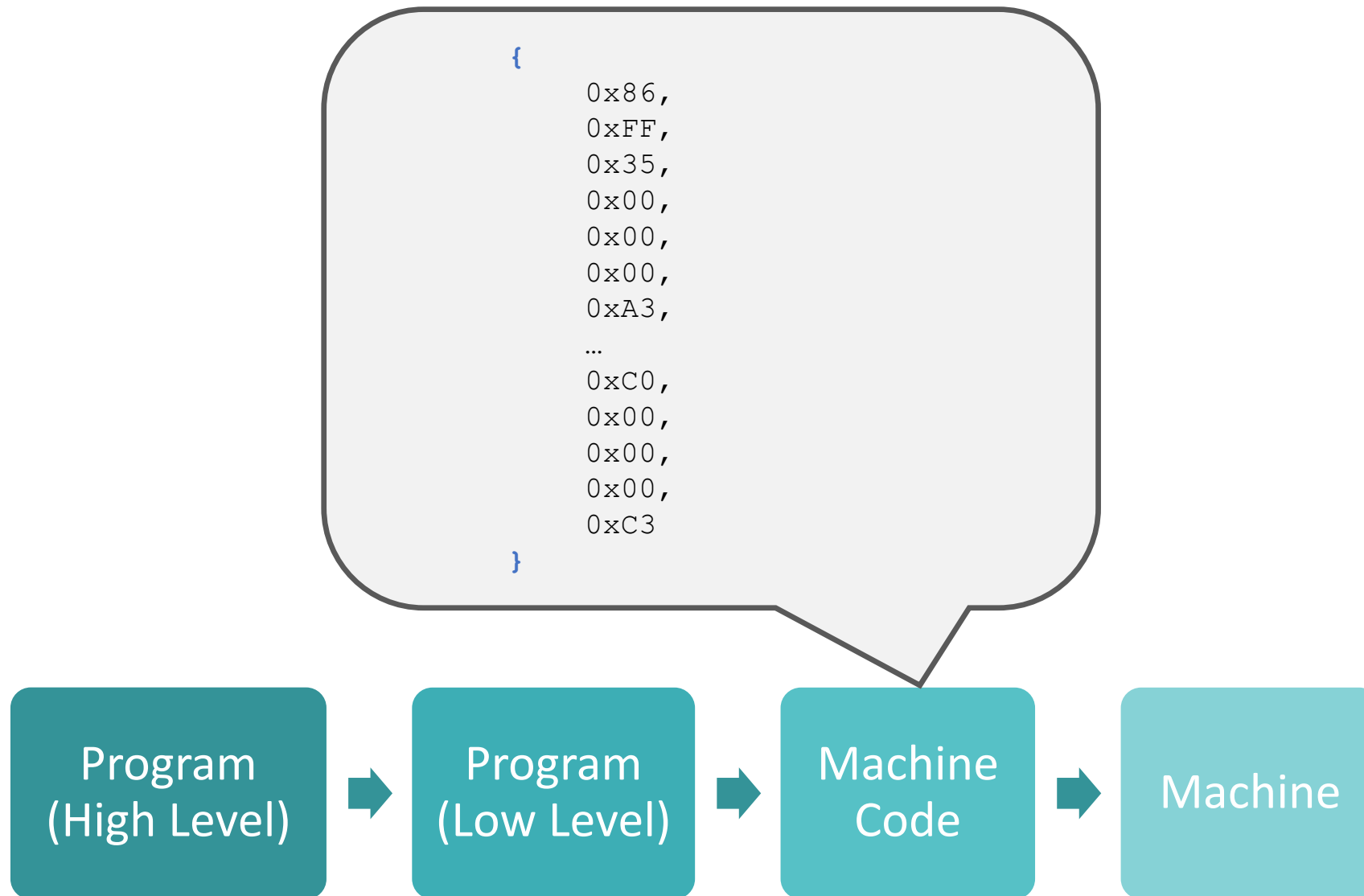
WHAT IS A PROGRAMMING LANGUAGE

```
section .text
    global _start:
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, string
    mov edx, length
    int 80h
    mov eax, 1
    mov ebx, 8
    int 80h

section .data
    string: db 'Hello World', 0Ah
    length: equ 13
```



WHAT IS A PROGRAMMING LANGUAGE



WHAT IS A PROGRAMMING LANGUAGE



HISTORY OF PROGRAMMING LANGUAGES

HISTORY OF PROGRAMMING LANGUAGES



HISTORY OF PROGRAMMING LANGUAGES

- First-generation languages
 - A.k.a. *machine code*
 - Binary codes of 0 and 1
 - not human-readable
 - No need for any translator or converter
 - Machine dependent

```
{  
    0x86,  
    0xFF,  
    0x35,  
    0x00,  
    0x00,  
    0x00,  
    0xA3,  
    ...  
    0xC0,  
    0x00,  
    0x00,  
    0x00,  
    0xC3  
}
```

HISTORY OF PROGRAMMING LANGUAGES

- Second-generation languages
 - A.k.a. *assembly languages*
 - Use mnemonics => more human-readable
 - “Low-level” programming language
 - **Assembler** converts into machine code

```
section .text
    global _start:
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, string
    mov edx, length
    int 80h
    mov eax, 1
    mov ebx, 8
    int 80h

section .data
    string: db 'Hello World', 0Ah
    length: equ 13
```


HISTORY OF PROGRAMMING LANGUAGES

- Third-generation languages
 - Functions, Types, Data structures, Objects, Libraries
 - Referred to as “*high-level*” languages
 - A **compiler** or **interpreter** translates to assembly language
 - Examples: FORTRAN, COBOL, PASCAL, C, C++, Java ...

```
#include <iostream>

int main() {
    std::cout <<
        "Hello World";
}
```

HISTORY OF PROGRAMMING LANGUAGES

- Fourth-generation languages
 - = Very high-level abstractions in a specific domain
 - Database mgmt.
 - Table-driven programming
 - Automatic Reporting
 - GUI Creation

```
Rectangle {  
    id: canvas  
    width: 250  
    height: 200  
    color: "blue"  
  
    Image {  
        id: logo  
        source: "pics/logo.png"  
        anchors.centerIn: parent  
        x: canvas.height / 5  
    }  
}
```

HISTORY OF PROGRAMMING LANGUAGES

Each next generation = more *abstractions*

Motivation: needed to construct larger, more complex software

Prerequisite: increased computing power



DEEP DIVE: EARLY HISTORY

```
#include <iostream>
```

```
int main() {  
    std::cout <<  
        "Hello World";  
}
```

```
section .text  
    global _start  
_start:  
    mov eax, 4  
    mov ebx, 1  
    mov ecx, string  
    mov edx, length  
    int 80h  
    mov eax, 1  
    mov ebx, 8  
    int 80h  
  
section .data  
    string: db 'Hello World', 0Ah  
    length: equ 13
```

```
{  
    0x86,  
    0xFF,  
    0x35,  
    0x00,  
    0x00,  
    0x00,  
    0xA3,  
    ...  
    0xC0,  
    0x00,  
    0x00,  
    0x00,  
    0xC3  
}
```

Program
(High Level)



Program
(Low Level)



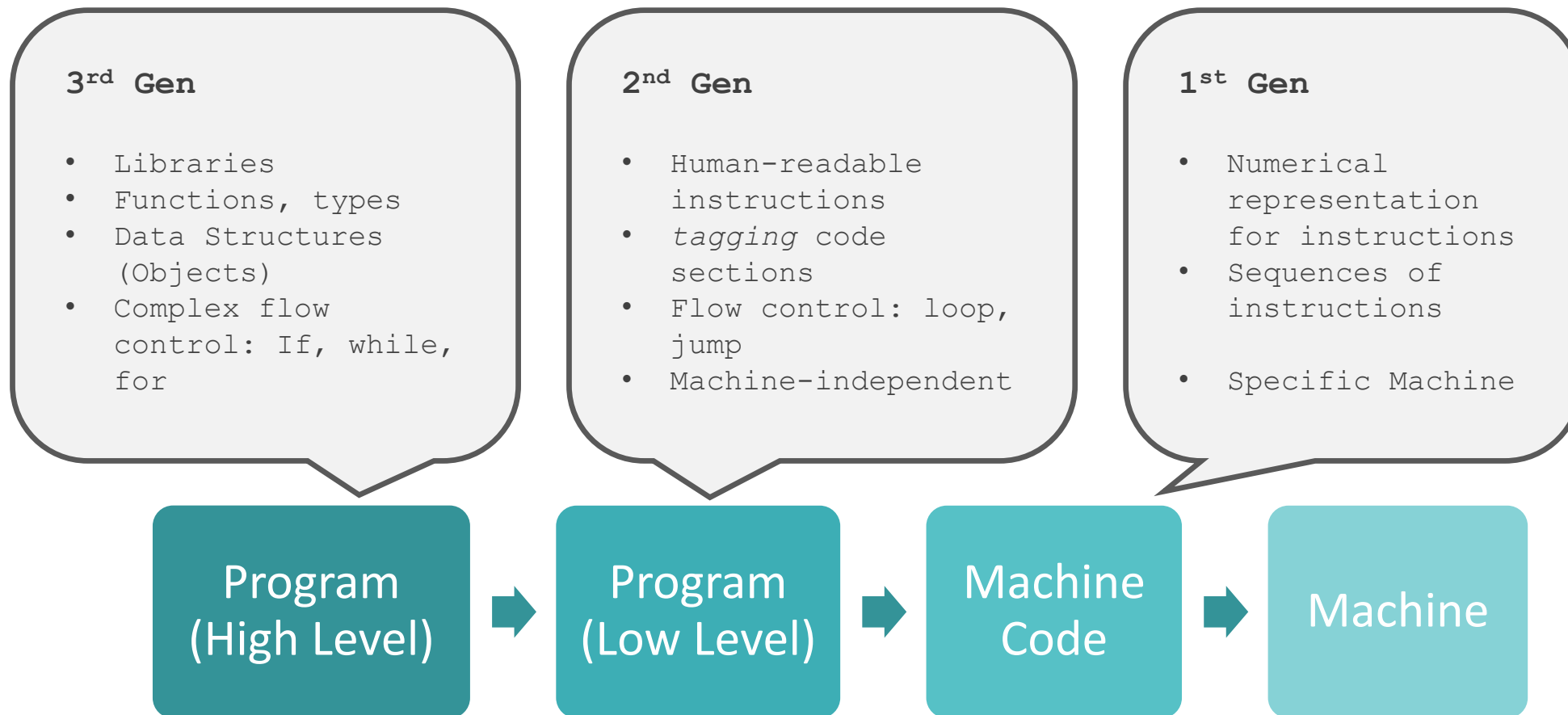
Machine
Code



Machine

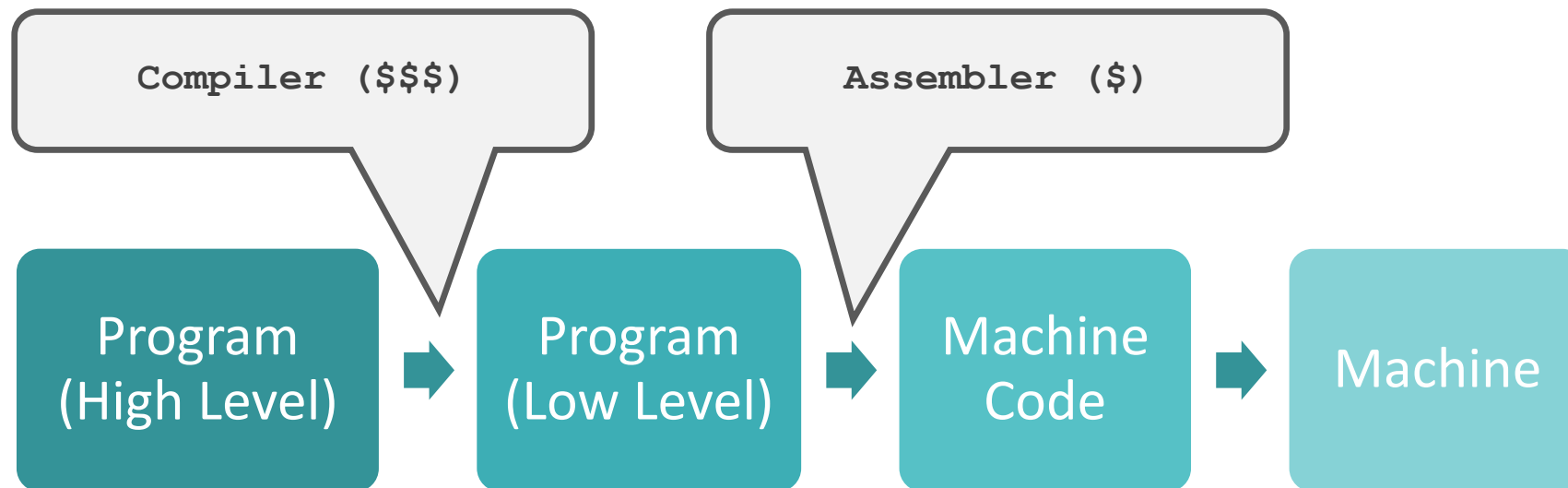


DEEP DIVE: EARLY HISTORY

























DEEP DIVE: EARLY HISTORY

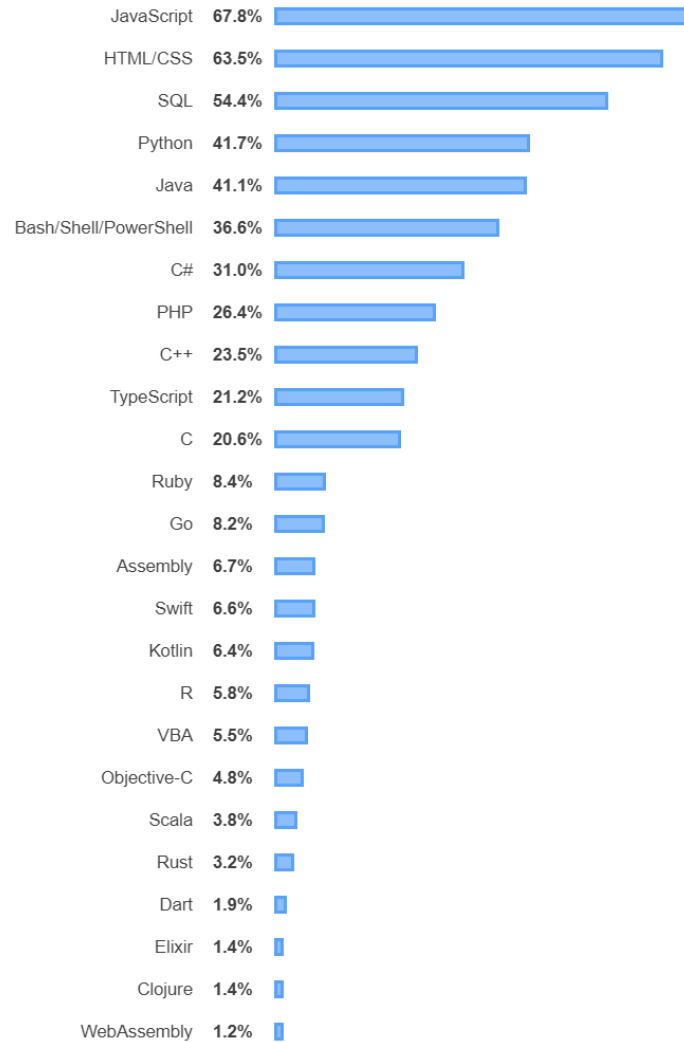


USAGE STATISTICS OF PROGRAMMING LANGUAGES

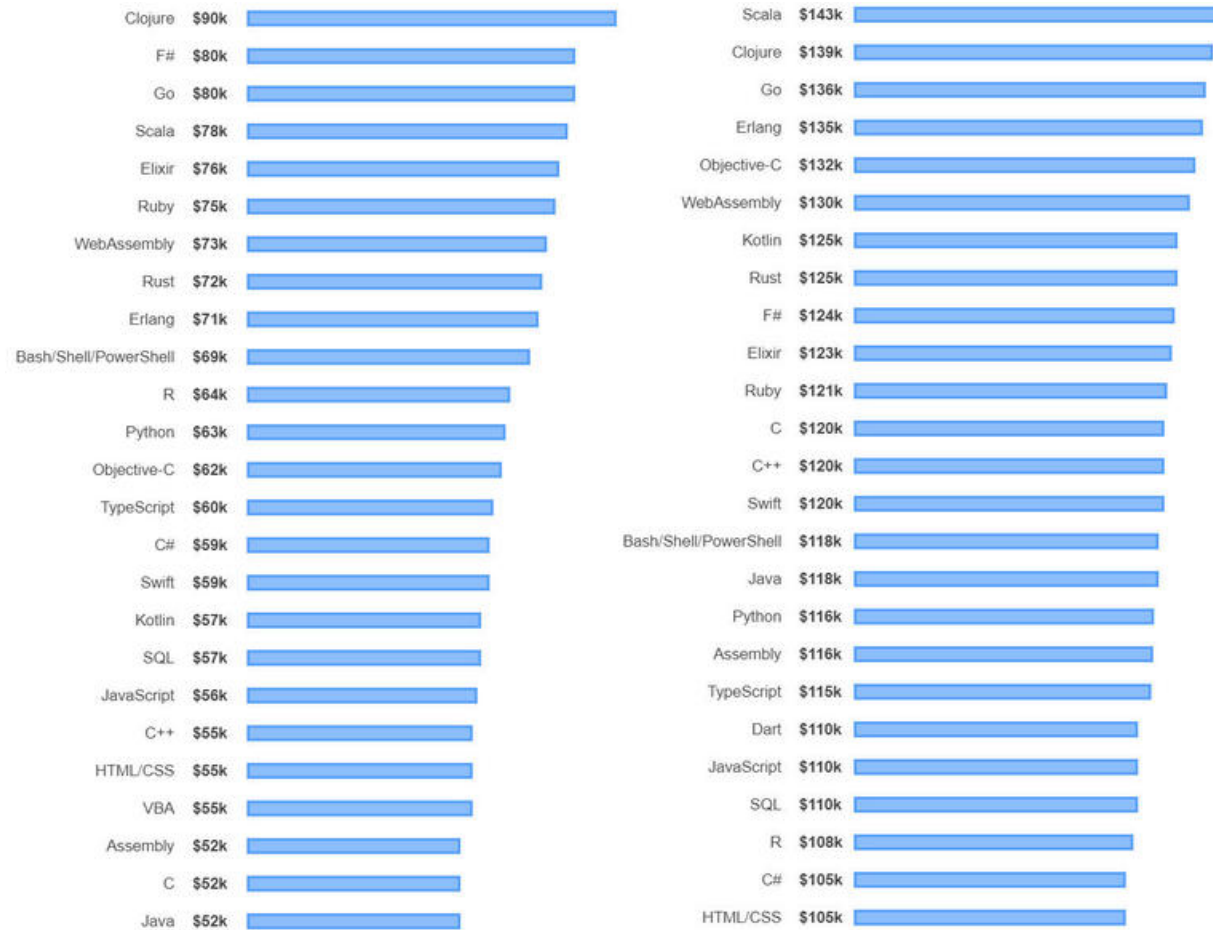
MOST USED

1	Java		11	MATLAB	
2	C		12	R	
3	Python		13	Perl	
4	C++		14	Assembly Language	
5	Visual Basic .NET		15	Swift	
6	Javascript		16	Go	
7	C#		17	Delphi/Object Pascal	
8	PHP		18	Ruby	
9	SQL		19	PL/SQL	
10	Objective-C		20	Visual Basic	

MOST LOVED

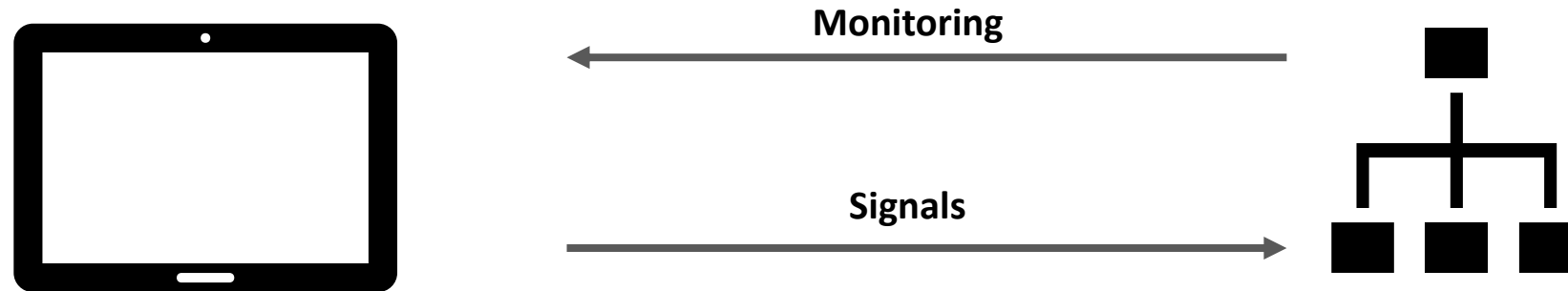


MOST PAID



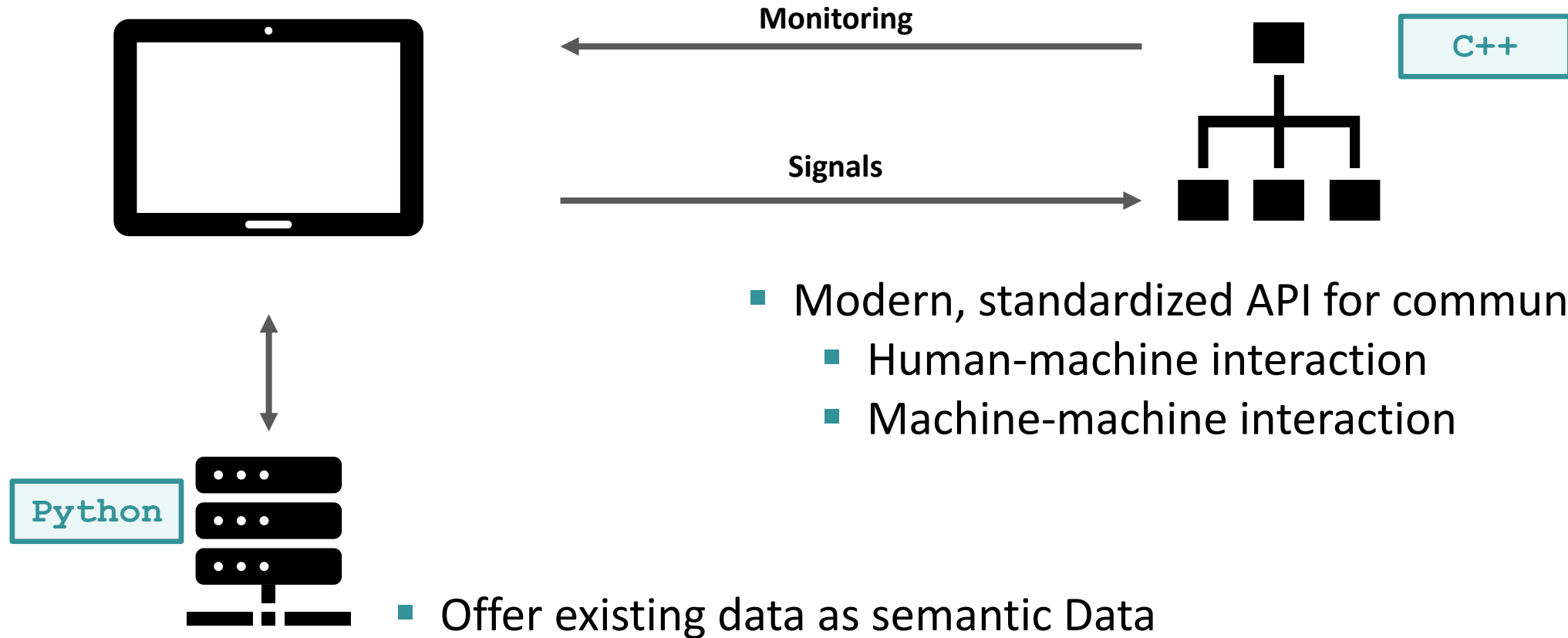
PEGUS DIGITAL USE CASES – WHY DID WE CHOOSE A CERTAIN TECHNOLOGY STACK?

USE CASES



- Offer existing data as Semantic Data
- Modern, standardized API for communication
 - Human-machine interaction
 - Machine-machine interaction

USE CASES



THE FUTURE OF PROGRAMMING LANGUAGES

THE FUTURE OF PROGRAMMING LANGUAGES

Fifth-Generation: “ultimate” level of abstraction

- Describe problem, not *algorithm* for solving the problem
- Abstracts away method needed for solving
- Examples: Genetic Programming, Machine Learning (e.g. Neural networks), Constraint-based Programming

Demo 1: Genetic Programming

<https://keiwan.itch.io/evolution>

https://rednuht.org/genetic_walkers/

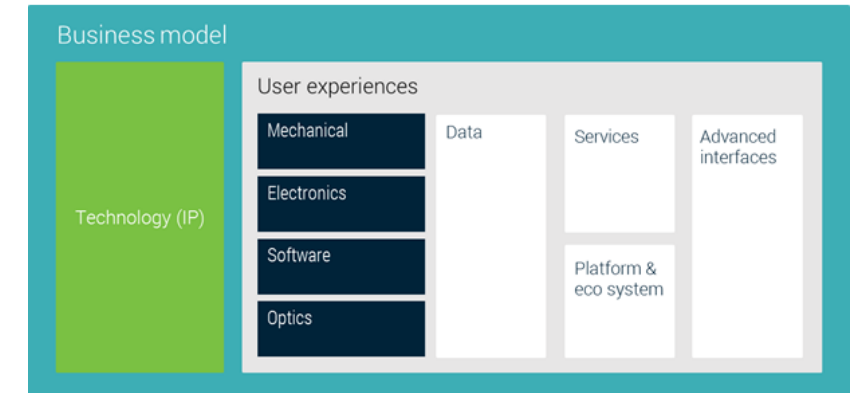
Demo 2: Constraint-based Programming (Logic-based)

<https://verne.cs.kuleuven.be/idp/server.html>

CONCLUSION

CONCLUSION

- IoT/Industry 4.0 -> More usage of low-level languages
- AI → Increase of Python, R, Lisp, Prolog
- Many languages to choose from
- Choice depends on:
 - Target hardware / system
 - Existing libraries
 - Community support
 - Expertise of your team / hiring market



One group, five brands

Our services are marketed through 5 brands each addressing specific missions in product development.

MASTERS IN INNOVATION®

INTEGRATED PRODUCT DEVELOPMENT



ON-SITE PRODUCT DEVELOPMENT



DIGITAL PRODUCT DEVELOPMENT



OPTICAL PRODUCT DEVELOPMENT

