Moving Swiftly into Swift for mainframes...

think 2018

Frank van der Wal
Digital Transformation
Technical Lead IBM Z BeNeLux
thewalll@nl.ibm.com

Why do we need modern languages on IBM Z?

Why do we need modern languages on IBM Z?

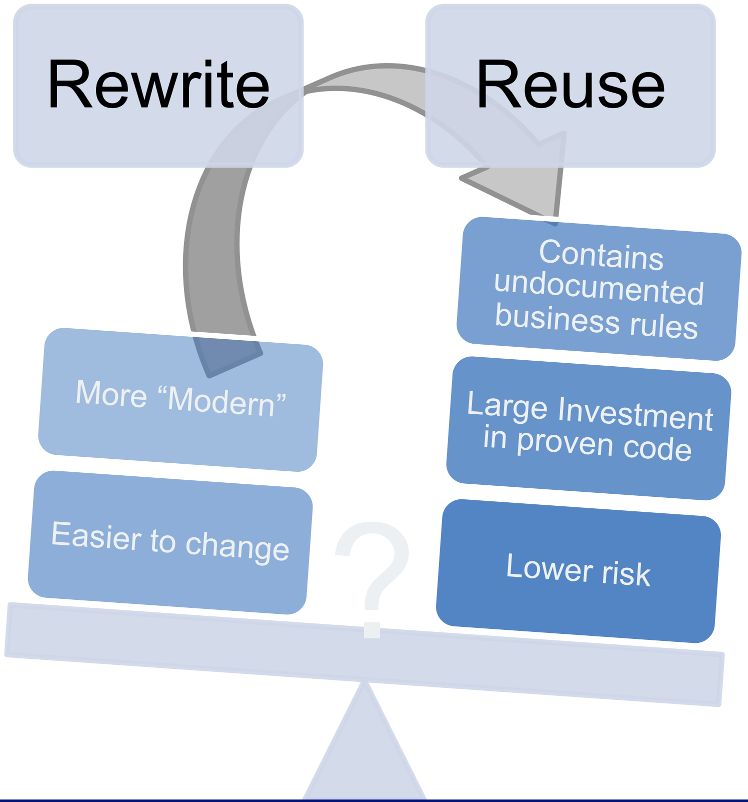
1 Skills: Millions of Available Developers

More than

14M

developers are using Java, JavaScript, and Swift, worldwide.

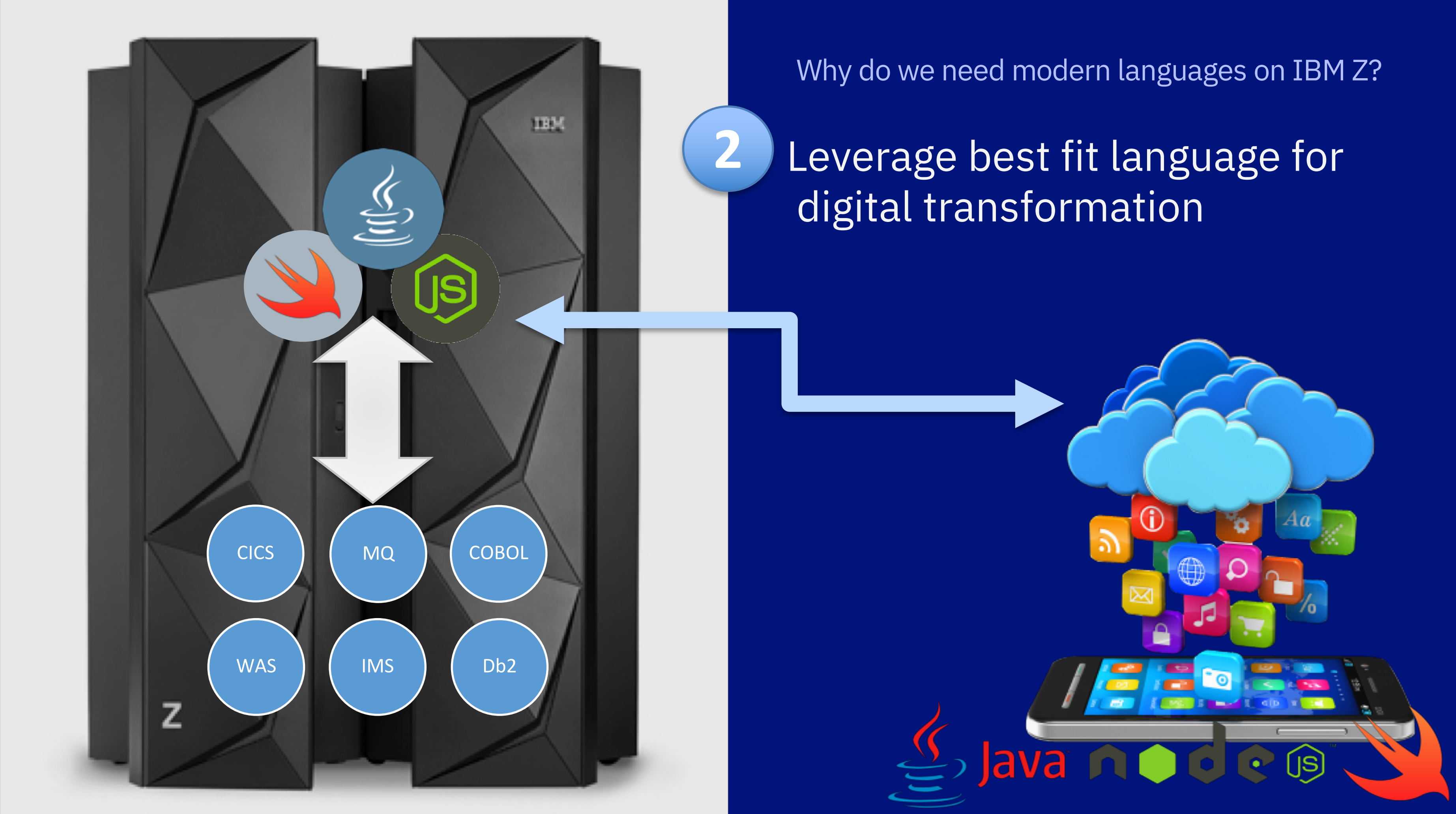
Remember: Your existing code is a valuable asset!


Rewrite
Reuse
More “Modern”
Easier to change
Contains undocumented business rules
Large Investment in proven code
Lower risk
?

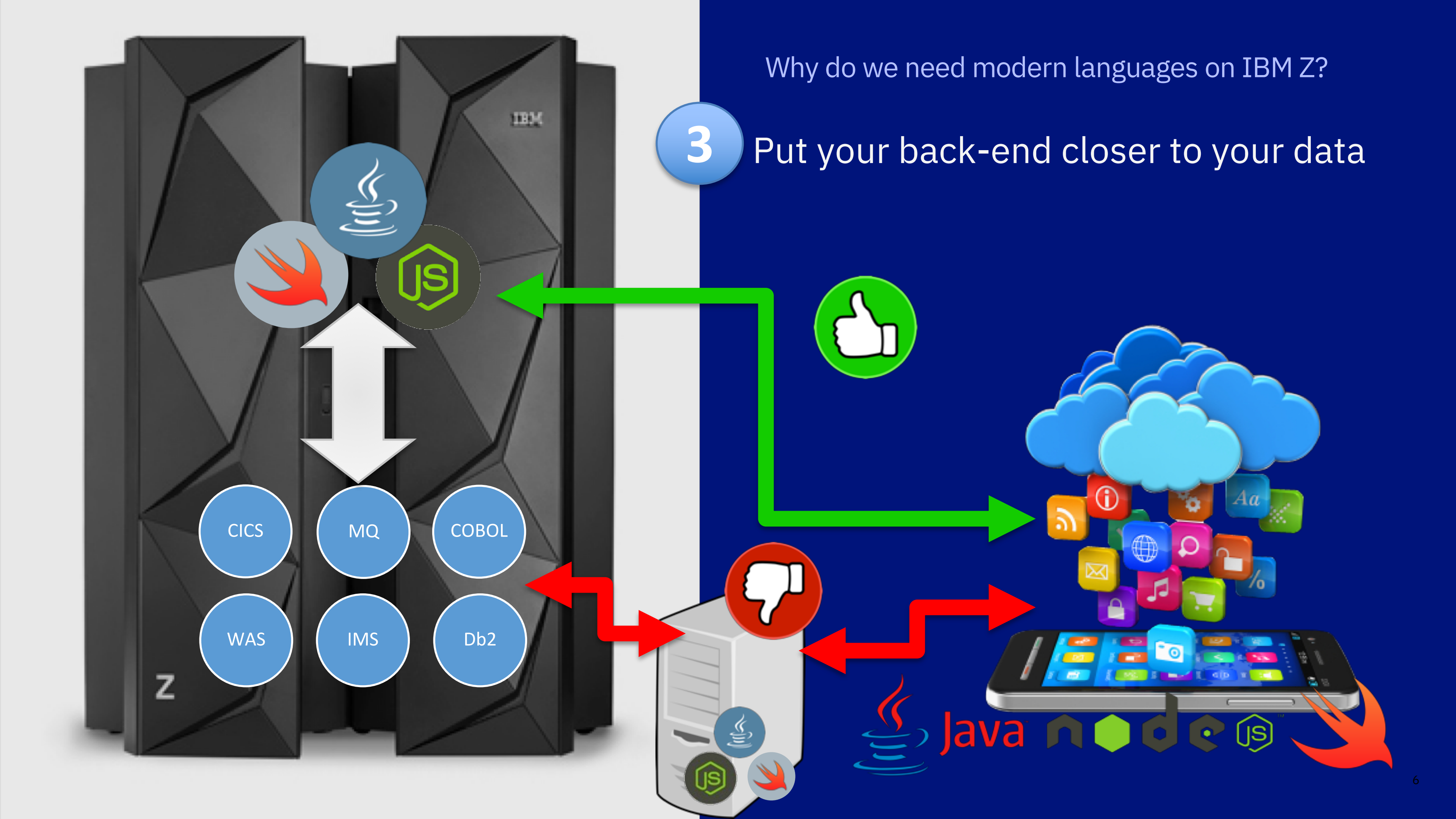
Get the best of the two strategies.

Why do we need modern languages on IBM Z?

2 Leverage best fit language for digital transformation



3 Put your back-end closer to your data



Swift Ecosystem

✓ Industry backing

✓ Developers acceptance

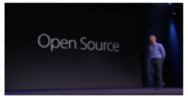
✓ Tools & packages

✓ Conferences & meetups

June 2014

WWDC 2014

Apple releases a new programming language called "Swift"



June 2015

WWDC 2015

Apple announces to open source Swift

"Swift is growing faster than anything else we track"

July 2015 RedMonk

Nov. 2015

Perfect.org

Dec. 2015

Apple releases Swift as **Open Source**.

Jan. 2016 RedMonk

Feb 2016

IBM InterConnect 2016

IBM to bring Swift to the cloud to radically simplify end-to-end development of apps

Programming Language Rankings *Swift "meteoric rise" continues. "Swift moves up one spot from #18 to #17 Y/Y...and is within hailing distance of our Tier 1 languages".*

Mar. 2016

try! Swift Tokyo

KITURA

stackoverflow Developer Survey Results 2016

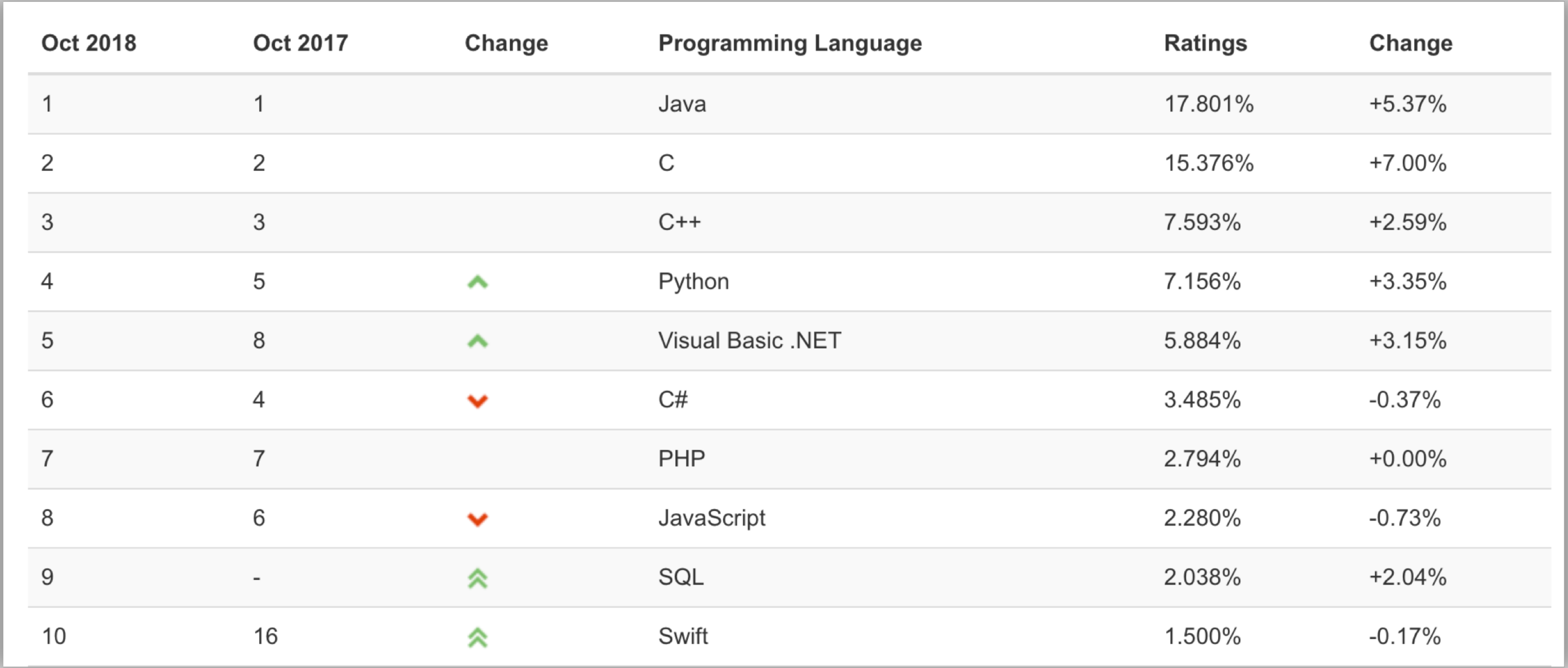
Swift #2 "***Most Loved***"

try! Swift NYC

VAPOR

Swift Summit

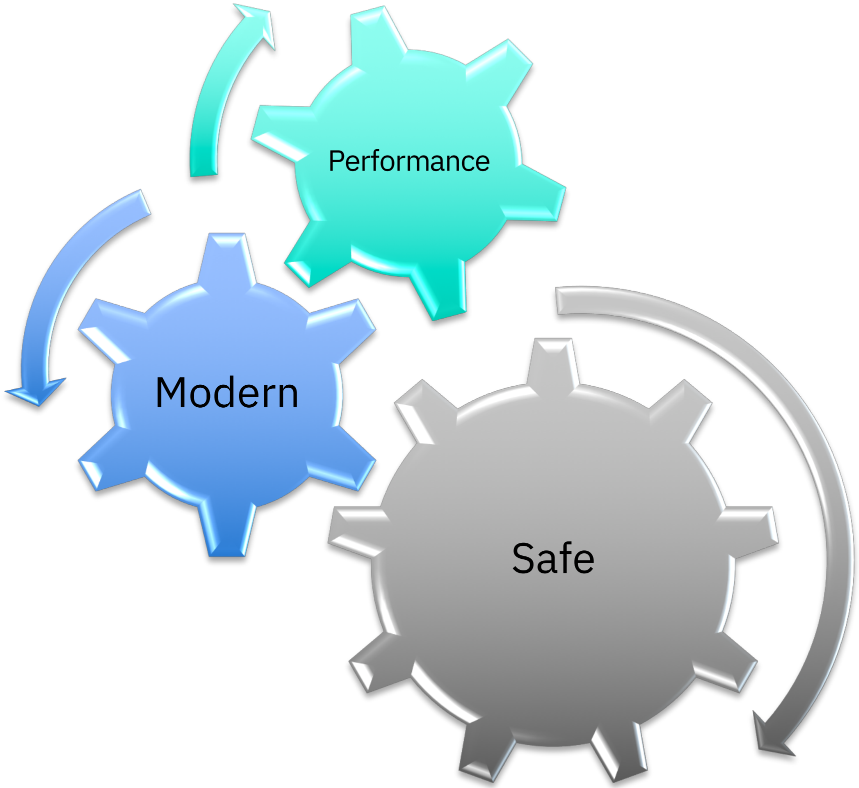
Swift is in the top 10 of most popular languages…



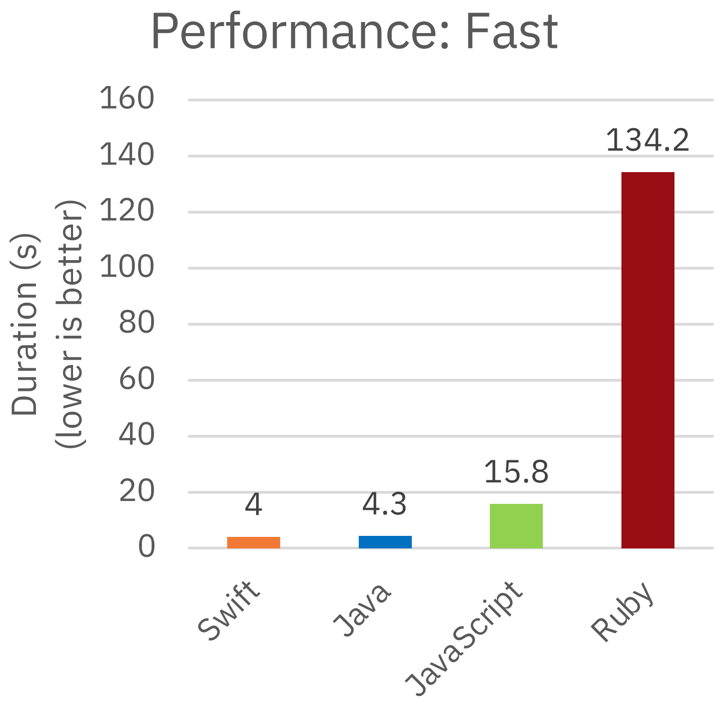
Oct 2018	Oct 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.801%	+5.37%
2	2		C	15.376%	+7.00%
3	3		C++	7.593%	+2.59%
4	5	▲	Python	7.156%	+3.35%
5	8	▲	Visual Basic .NET	5.884%	+3.15%
6	4	▼	C#	3.485%	-0.37%
7	7		PHP	2.794%	+0.00%
8	6	▼	JavaScript	2.280%	-0.73%
9	-	▲▲	SQL	2.038%	+2.04%
10	16	▲▲	Swift	1.500%	-0.17%

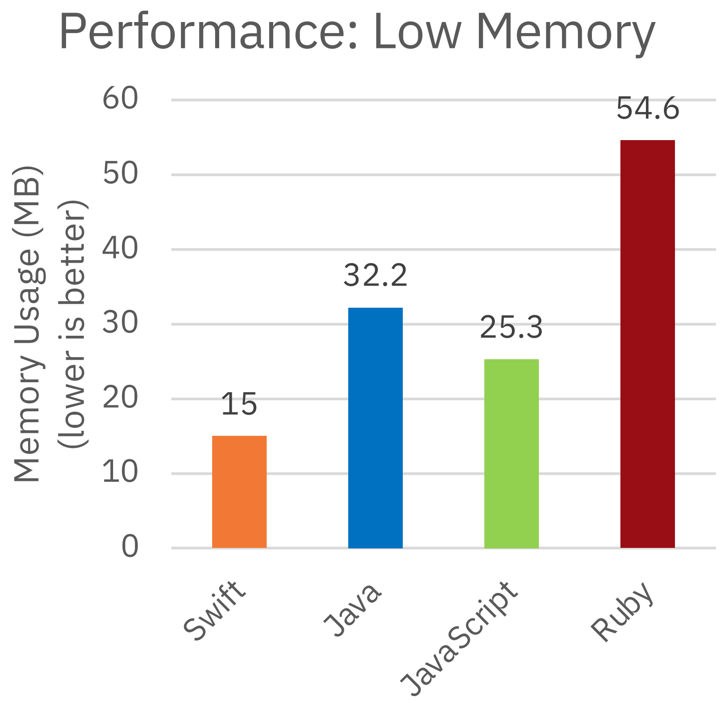
http://www.tiobe.com/tiobe-index/

Why Swift?



Why Swift? **Performance.** It's a compiled language after all





Source: benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm

Why Swift? **Performance**

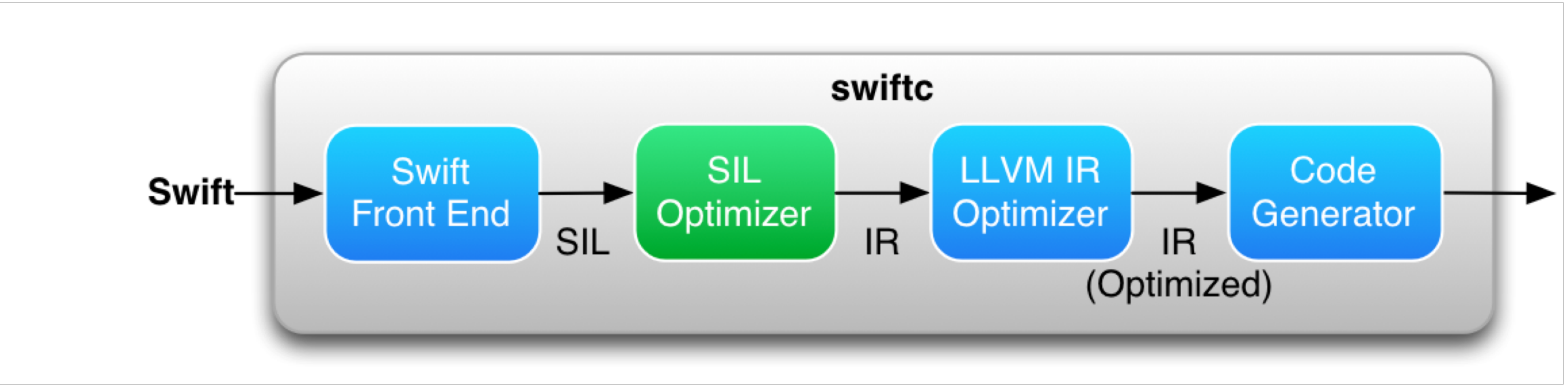
- Compiled to *native* code
  - Leverages LLVM Back-end and Optimizer
- Slightly better than Java in *speed/memory*
- Concurrency built-in
- Automatic Code Optimizer
  - -O (inlining, loop optimizations, etc)
  - -whole-module-optimization
- Can *directly* call C Libraries via C interface

Invocation: swiftc –O main.swift –o a.out

Why Swift? **Performance**

Based on the well-established **LLVM framework**

- Actively developed compiler and toolchain, IBM/Google/Apple contributing to it
- Geared to work with LLDB
  - LLDB is a next generation, high-performance debugger
- Has a REPL (interactive compiler)



zArch Machine Code

Why Swift? **Safe**

- Type safe
  - Helps developers refactor, extend, iterate on solutions.
  - Error detection at compile time
- Automatic initialization
  - Variables are automatically initialized
  - Memory is automatically allocated and managed.

Why Swift? **Safe**

- **Type inference**
  - Catch potential run-time errors at compile-time

- **Memory management**
  - Uses ARC (automatic reference counting)
  - Pointers are allowed but ***discouraged***
  - *Optionals* are intended as a replacement

- Variables and constants always *initialized* and array bounds are always *checked*.

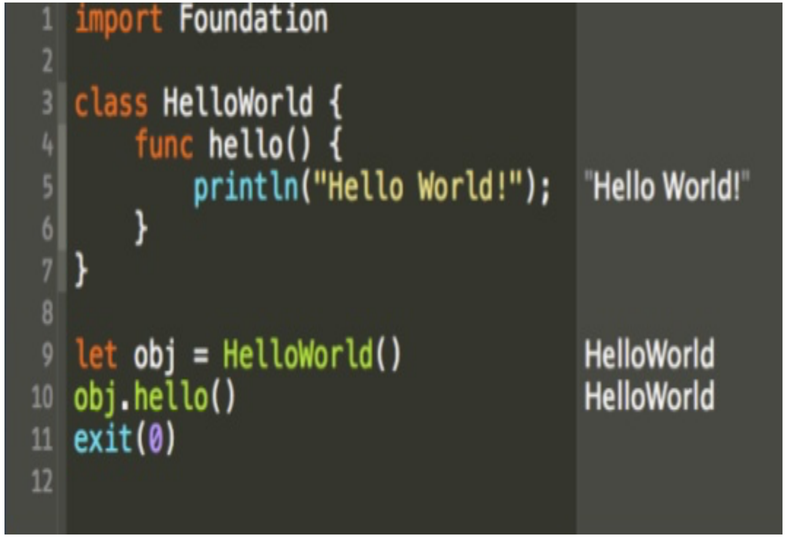
```
let pi = 3.14159
// constant pi is inferred to be of type Double

var msg = “Swift on z/OS”
// variable msg is inferred to be of type String
```

```
var name:String?
// name is of type Optional String
name = “z/OS”
name = nil
```

Why Swift? **Modern**

- Easy to learn
    - Concepts are similar in other popular languages (C, C++, Java)
- Concise and straightforward to read
    - Little verbiage
    - Similar syntax to Java and C++



```
import Foundation

class HelloWorld {
    func hello() {
        println("Hello World!");    "Hello World!"
    }
}

let obj = HelloWorld()              HelloWorld
obj.hello()                         HelloWorld
exit(0)

```

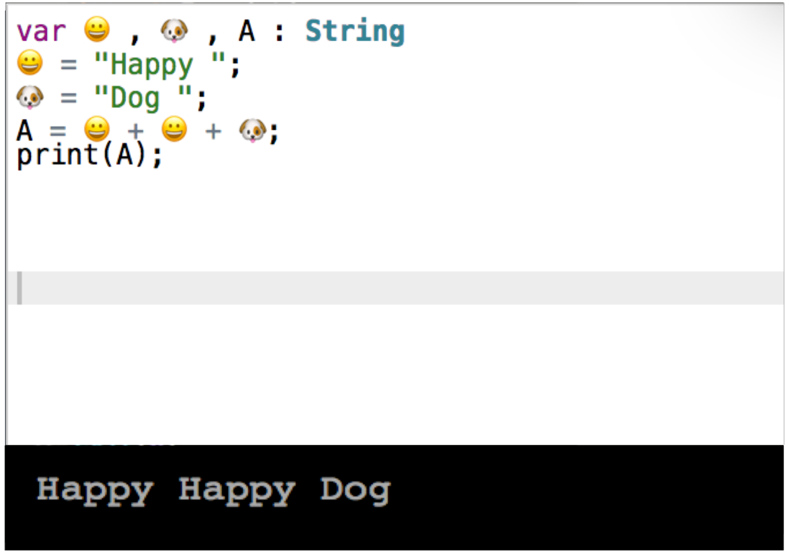
Why Swift? **Modern**

Language Concepts

Unicode language (variable names and values)

Classes

Initializers, Deinitializers, Inheritance, Methods, Parameters , Setters/Getters



```
var 😀 , 🐶 , A : String
😀 = "Happy ";
🐶 = "Dog ";
A = 😀 + 😀 + 🐶;
print(A);
```

```
Happy Happy Dog
```

```
class Example {
    var a = 0
    var b: String

    init(a: Int) {   // Constructor
        self.a = a
        b = "name"   // An error if a declared property isn't initialized
    }
}
```

Why Swift? **Modern**

Language Concepts

Tuples

```
let http404Error = (404, “Not Found”)
```

Closure Expressions

```
reversedNames = names.sorted(by: { (s1: String, s2: String) -> Bool in
            return s1 > s2 })
```

Protocols (aka interfaces)

```
protocol SupportsToString { func toString() -> String }
```

Operator Overloading

```
static func +(left: Vector, right: Vector) -> Vector
{ return [left.x + right.x, left.y + right.y, left.z + right.z] }
```

Generics

```
func addTwoValues<T>(_ a: inout T, _ b: inout T) { return a+b }
```

Why Swift? **Modern**

JSON Codable serialization/ deserialization

Logging

Networking

File IO

Containers

Arrays, Sets, Dictionaries, etc

For..in loops

```
let numberOfLegs = ["spider": 8, "ant": 6, "cat": 4]

for (animalName, legCount) in numberOfLegs {
    print("\(animalName)s have \(legCount) legs")
}
```

IBM Toolkit for Swift – Linux on z Systems

- Core tools to develop in Swift:
  - Compiler
  - Swift Runtime
  - Libraries
  - Debugger (lldb)
  - Web framework (Kitura)
  - Package Manager

Community Edition
(free of charge)

Enterprise Edition
(License + S&S)

https://www.ibm.com/marketplace/swift-compiler

IBM Z | Swift@IBM

IBM Toolkit for Swift on z/OS

Community Edition

https://developer.ibm.com/mainframe/products/ibm-toolkit-swift-z-os/

Key features in Swift 4.1

- Swift compiler
- Standard Library
- Core Libraries
- Package Manager
- Sample Swift application based on Kitura

- Interoperability with C, PL/I, assembly, VSAM, and DB2.
- Free of charge

Sample Scenario:

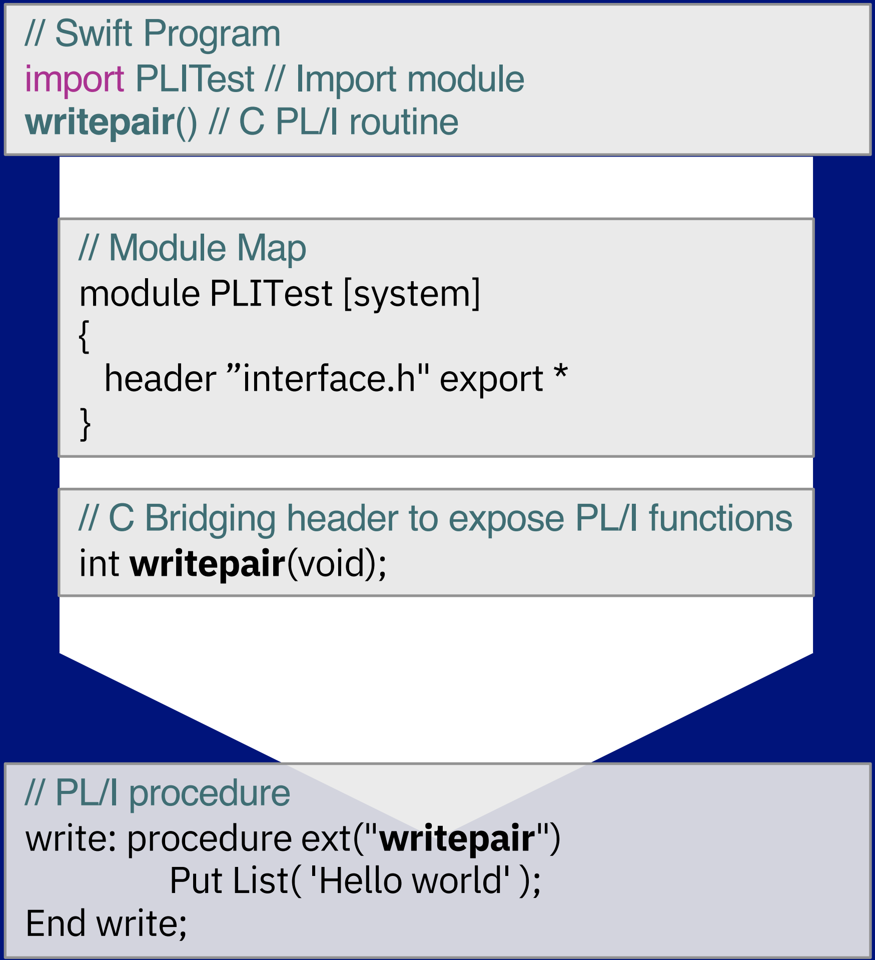
Call PL/I directly from Swift

Swift supports interlanguage calls to PL/I

Requirements:

1. PL/I procedures compiled as 64-bit (-qlp=64)
2. Swift Module Map to expose PL/I library
3. C bridging header to expose PL/I routines

The same scenario applies for C, C++, and PL/I



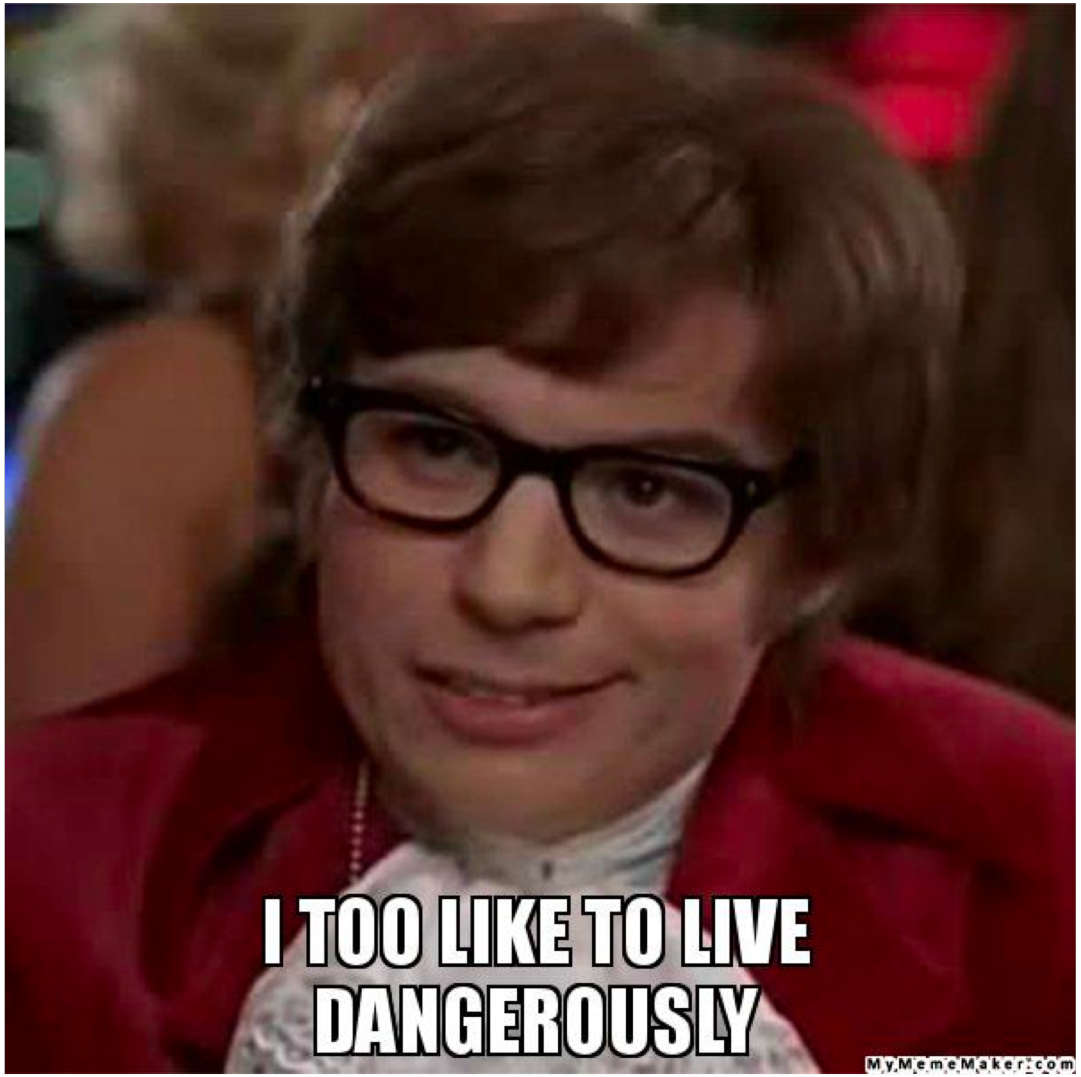
```
// Swift Program
import PLITest // Import module
writepair() // C PL/I routine
```

```
// Module Map
module PLITest [system]
{
  header ”interface.h" export *
}
```

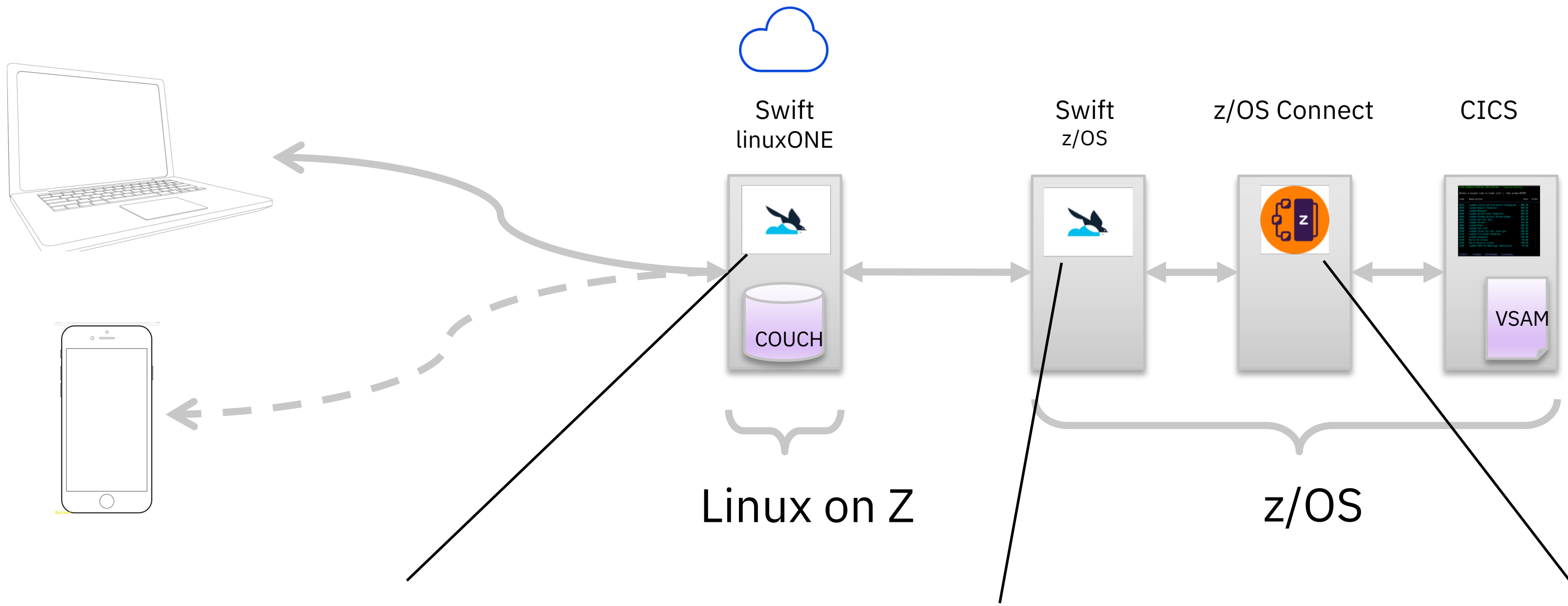
```
// C Bridging header to expose PL/I functions
int writepair(void);
```

```
// PL/I procedure
write: procedure ext("writepair")
            Put List( 'Hello world' );
End write;
```

Demo…



High level architecture

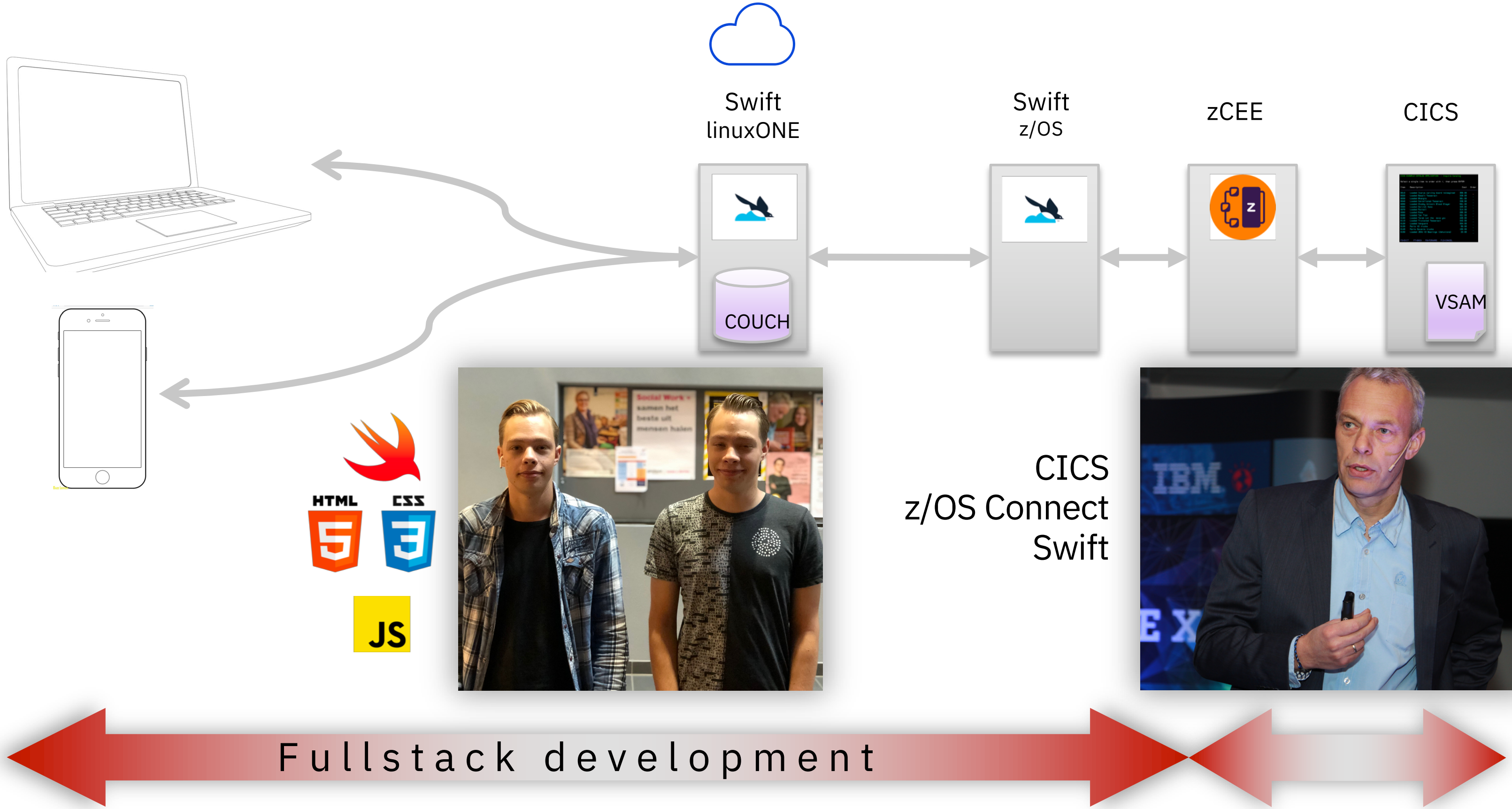


Kitura to route incoming requests
Receives response and …
…fetches pictures from COUCH DB
Render HTML using Kitura Stencil

Kitura to route incoming requests
Update database using existing PL/1*

Created REST API's into an existing CICS trx

Advantages of having a full stack development team



CICS
z/OS Connect
Swift

You can impact the future

- We are looking for **innovators and early adopters**
    - Validate user scenarios and get early access to the latest drivers.
    - If interested, please contact:
      shereen@ca.ibm.com
      thewall@nl.ibm.com

Resources

- Swift
  - IBM Marketplace (Swift on Linux on z): https://www.ibm.com/us-en/marketplace/swift-compiler
  - Swift @ IBM: https://developer.ibm.com/swift/
  - Extending Swift Value(s) to the Server (Free e-book): https://www-01.ibm.com/marketing/iwm/dre/signup?source=mrs-form-10468&S_PKG=ov55459
  - Free online course about server-side Swift: http://blog.udacity.com/2017/06/server-side-swift-with-ibm.html