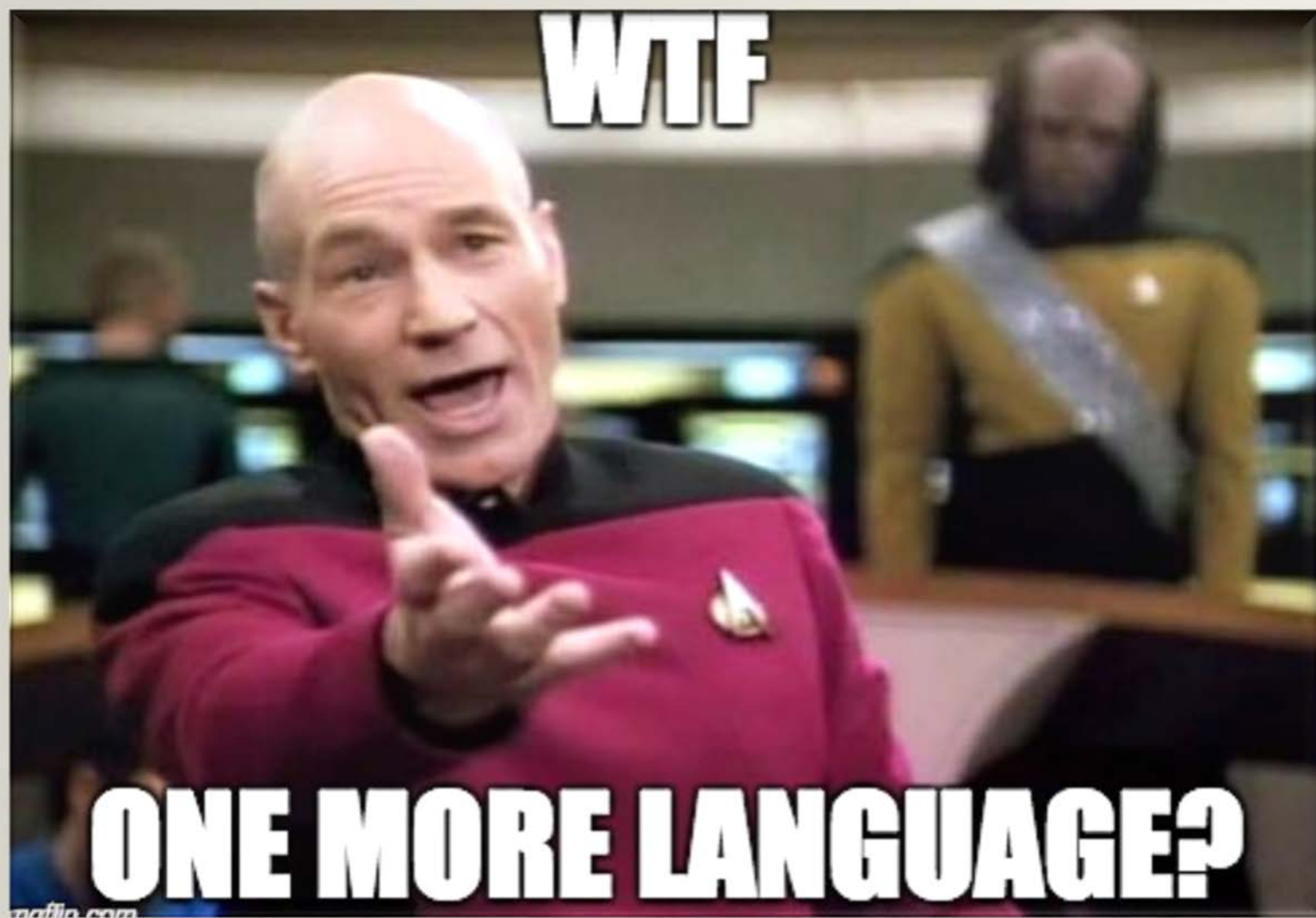


Typescript

is a language developed by

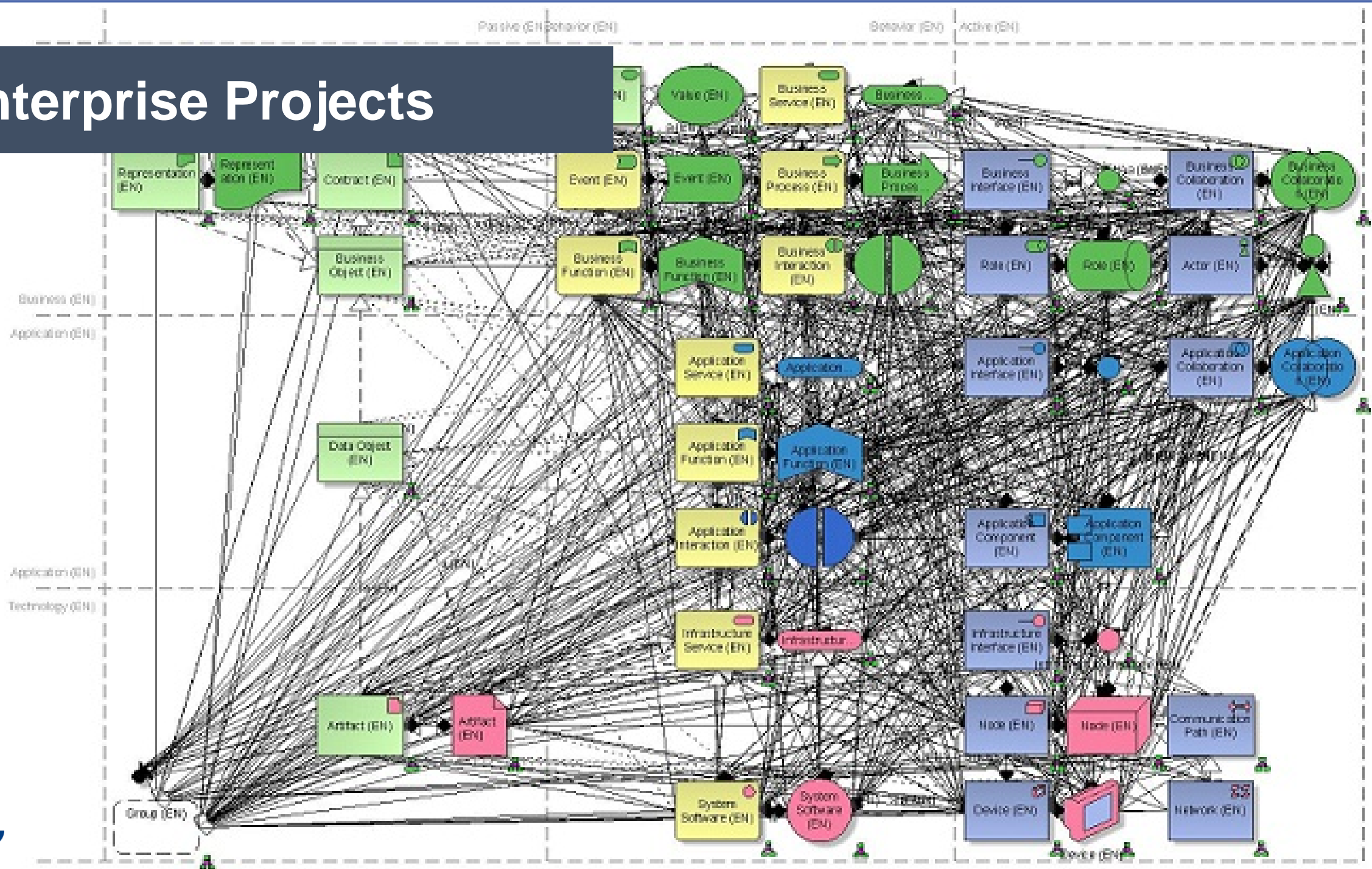




WTF

ONE MORE LANGUAGE?

Enterprise Projects





Atwood's Law

Any application that *can* be written in JavaScript, *will* eventually be written in JavaScript.

<http://www.codinghorror.com/blog/2007/07/the-principle-of-least-power.html>

JavaScript was originally developed for applications with a few hundred lines of code!

“Netscape”, 1995

I JUST LET SOMEBODY ELSE



MAINTAIN MY JAVASCRIPT

TypeScript

A language for large scale
JavaScript development

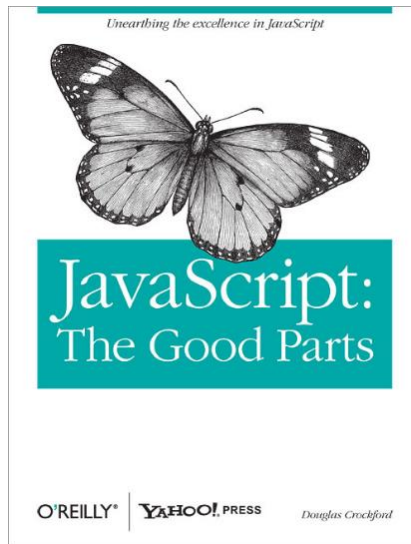
TypeScript

A typed superset of JavaScript that
compiles to plain JavaScript

Any browser
Any host
Any OS

Open Source

URL: <https://github.com/Microsoft/TypeScript>



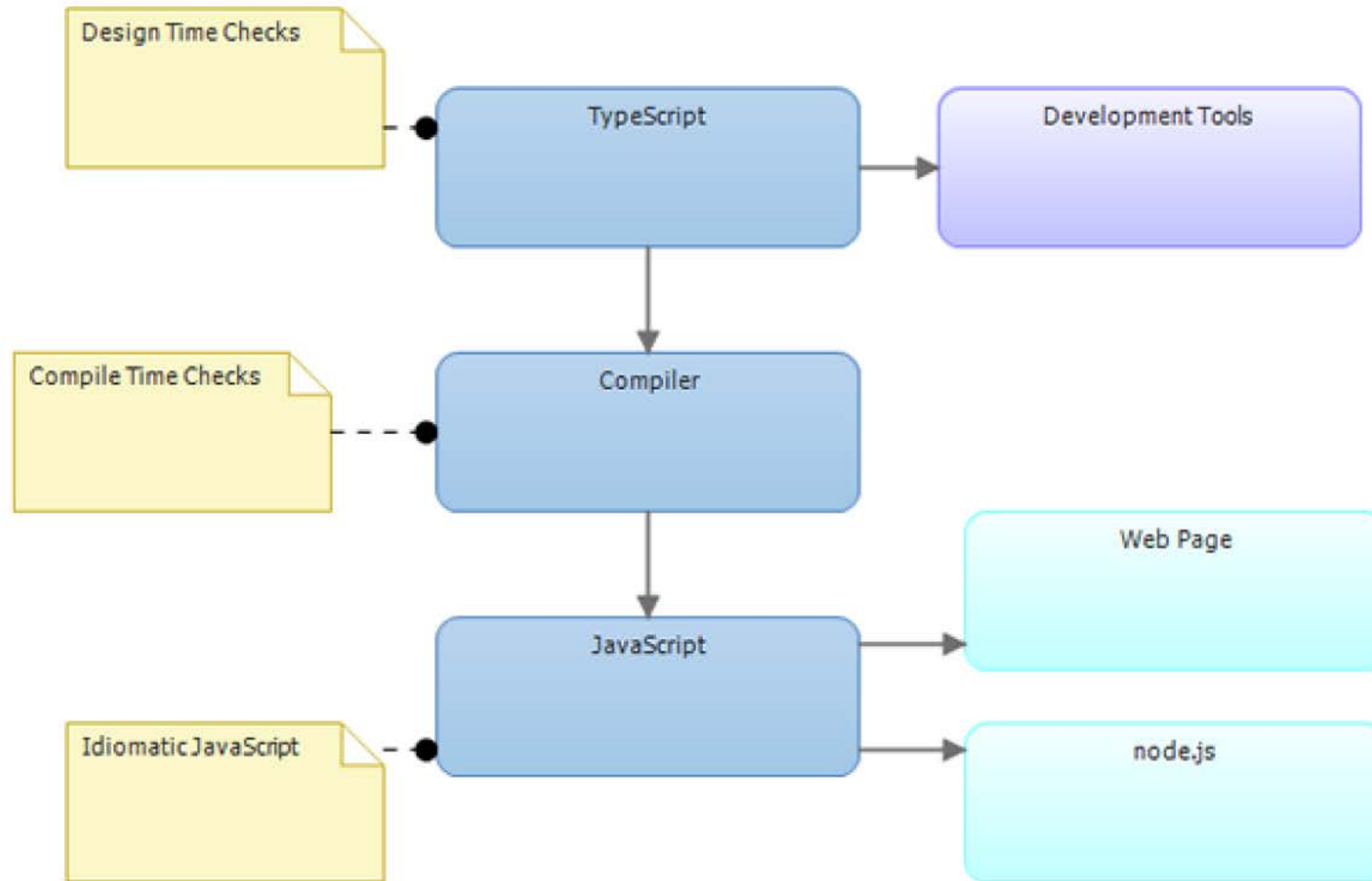
+



+



TypeScript life cycle



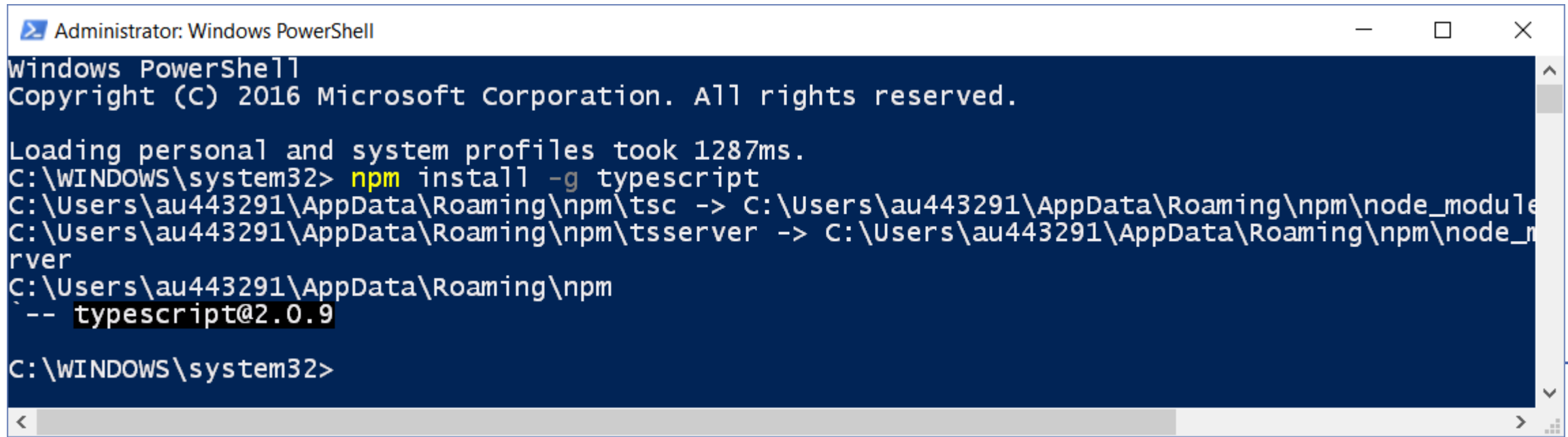
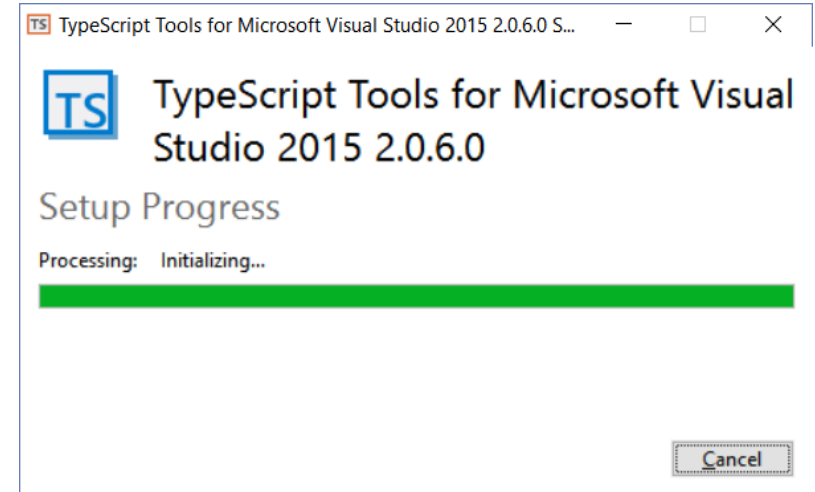
How to install?

- There are two main ways to get the TypeScript tools:

- Via npm
- By installing TypeScript's Visual Studio plugins

- For NPM users:

```
npm install -g typescript
```

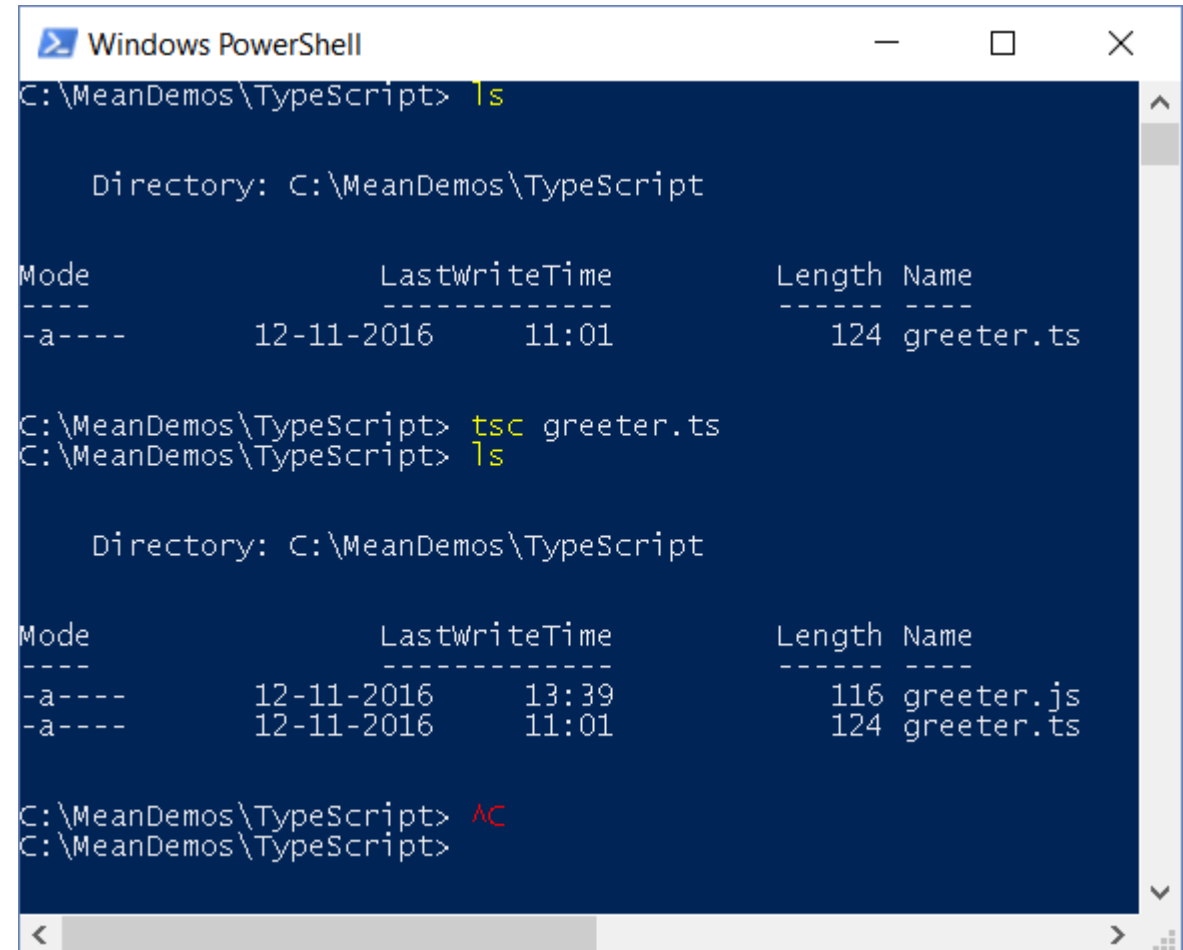


Compiling

- Run the TypeScript compiler:

```
tsc fileName.ts
```

- You may need to update the PATH environment variable manually



The screenshot shows a Windows PowerShell window with the following commands and output:

```
C:\MeanDemos\TypeScript> ls
```

Directory: C:\MeanDemos\TypeScript

Mode		LastWriteTime	Length	Name
-a----		12-11-2016 11:01	124	greeter.ts

```
C:\MeanDemos\TypeScript> tsc greeter.ts
C:\MeanDemos\TypeScript> ls
```

Directory: C:\MeanDemos\TypeScript

Mode		LastWriteTime	Length	Name
-a----		12-11-2016 13:39	116	greeter.js
-a----		12-11-2016 11:01	124	greeter.ts

```
C:\MeanDemos\TypeScript> AC
C:\MeanDemos\TypeScript>
```


Type System

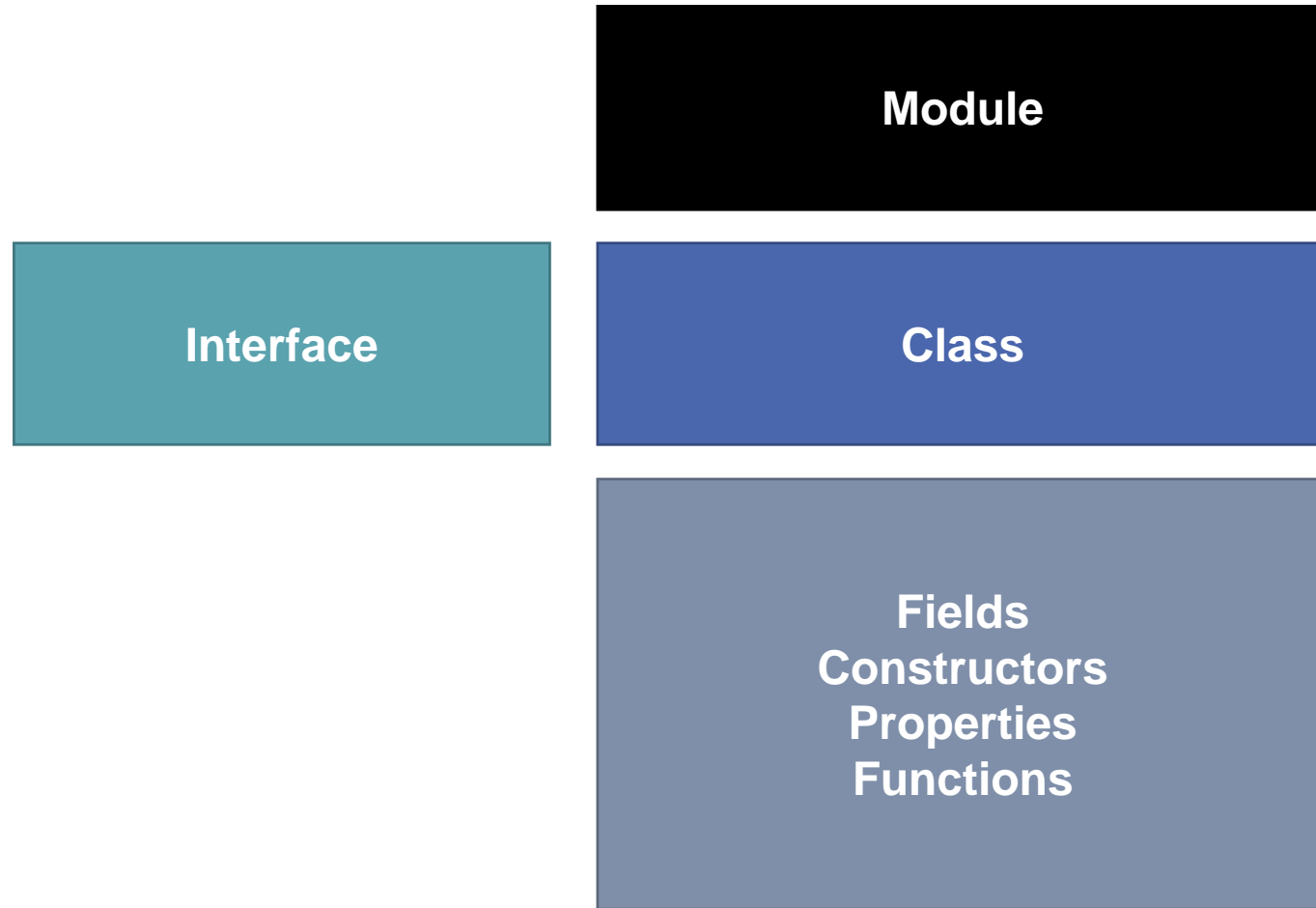
An accurate static representation of JavaScript's dynamic run-time type system

- Structural typing and type inference
 - In practice very few type annotations are necessary
- Generics
 - Increases accuracy and expressiveness of type system
- Works with existing JavaScript libraries
 - Declaration files can be written and maintained separately
- Types enable tooling
 - Provide verification and assistance, but not hard guarantees

Classes, Interfaces, Modules

- Scalable application structuring
 - Classes, interfaces, and modules enable clear contracts in code
- Aligned with emerging standards
 - Class and lambda syntax aligns with ECMAScript 6
- Supports popular module systems
 - CommonJS and AMD modules in any ECMAScript 3 environment

Code Hierarchy



Type annotations

TypeScript:

```
function greeter(person: string) {  
    return "Hello, " + person;  
}  
  
var user = "John Doe";  
  
console.log(greeter(user));
```

JavaScript:

```
function greeter(person) {  
    return "Hello, " + person;  
}  
  
var user = "John Doe";  
console.log(greeter(user));
```

Classes & Interfaces

TypeScript:

```
class Student {
    fullName: string;
    constructor(public firstName, public middleInitial,
public lastName) {
        this.fullName = firstName + " " + middleInitial
+ " " + lastName;
    }
}

interface Person {
    firstName: string;
    lastName: string;
}

function greeter(person : Person) {
    return "Hello, " + person.firstName + " " +
person.lastName;
}

var user = new Student("John", "M.", "Doe");
console.log(greeter(user));
```

JavaScript:

```
var Student = (function () {
    function Student(firstName, middleInitial, lastName) {
        this.firstName = firstName;
        this.middleInitial = middleInitial;
        this.lastName = lastName;
        this.fullName = firstName + " " + middleInitial + " " +
lastName;
    }
    return Student;
})();

function greeter(person) {
    return "Hello, " + person.firstName + " " +
person.lastName;
}

var user = new Student("John", "M.", "Doe");
console.log(greeter(user));
```

Decorators

- A Decorator is a special kind of declaration that can be attached to a class declaration, method, accessor, property, or parameter
- Decorators use the form `@expression`, where expression must evaluate to a function that will be called at runtime with information about the decorated declaration

Decorator example

TypeScript:

```
function ClassDecoratorParams(param: string) {
    return function(
        target: Function // The class the decorator is
        declared on
    ) {
        console.log("ClassDecoratorParams(" +
            param + ") called on: ", target);
    }
}

@ClassDecoratorParams("a")
@ClassDecoratorParams("b")
class ClassDecoratorParamsExample {
}
```

JavaScript:

```
var __decorate = (this && this.__decorate) || function (decorators, target, key, desc) {
    var c = arguments.length, r = c < 3 ? target : desc === null ? target : desc, d;
    if (typeof Reflect === "object" && typeof Reflect.decorate === "function") r = Reflect.decorate(decorators, target, key, desc);
    else for (var i = decorators.length - 1; i >= 0; i--) if (d = decorators[i]) r = (c > 1 ? __decorate([d], target, key, desc) : d) ;
    return c > 3 && r && Object.defineProperty(target, key, r), r;
};

function ClassDecoratorParams(param) {
    return function (target // The class the decorator is declared on
    ) {
        console.log("ClassDecoratorParams(" +
            param + ") called on: ", target);
    };
}

var ClassDecoratorParamsExample = (function () {
    function ClassDecoratorParamsExample() {
    }
    ClassDecoratorParamsExample = __decorate([
        ClassDecoratorParams("a"),
        ClassDecoratorParams("b")
    ], ClassDecoratorParamsExample);
    return ClassDecoratorParamsExample;
})();
```

Tools

- Typewriter
 - **Nuget package** to VS that generates TypeScript files from c# code files using TypeScript Templates
 - This allows you to create fully typed TypeScript representations of server side API that automatically updates when you make changes to your c# code
 - <https://visualstudiogallery.msdn.microsoft.com/e1d68248-f30e-4a5d-bf18-31399a0bcfa6>
- DefinitelyTyped
 - The repository for high quality TypeScript type definitions
 - <http://definitelytyped.org/>

References & Links

- Anders Hejlsberg om TypeScript 2
<https://channel9.msdn.com/Blogs/Seth-Juarez/Anders-Hejlsberg-on-TypeScript-2>
<https://www.version2.dk/artikel/anders-hejlsberg-typescript-compiler-giver-javascript-udviklere-mere-intelligente>
- TypeScript tutorial
<http://www.typescriptlang.org/docs/tutorial.html>