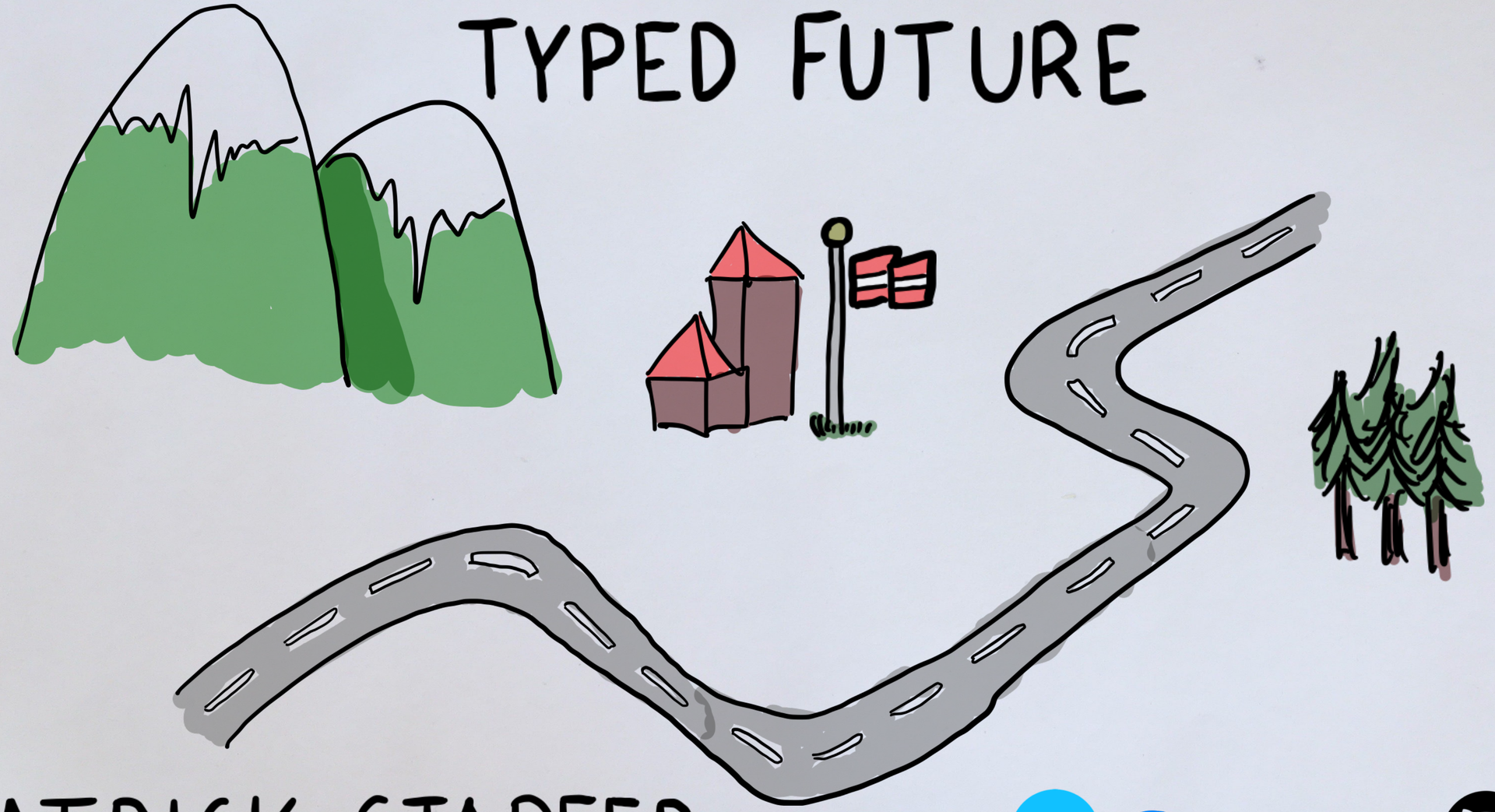
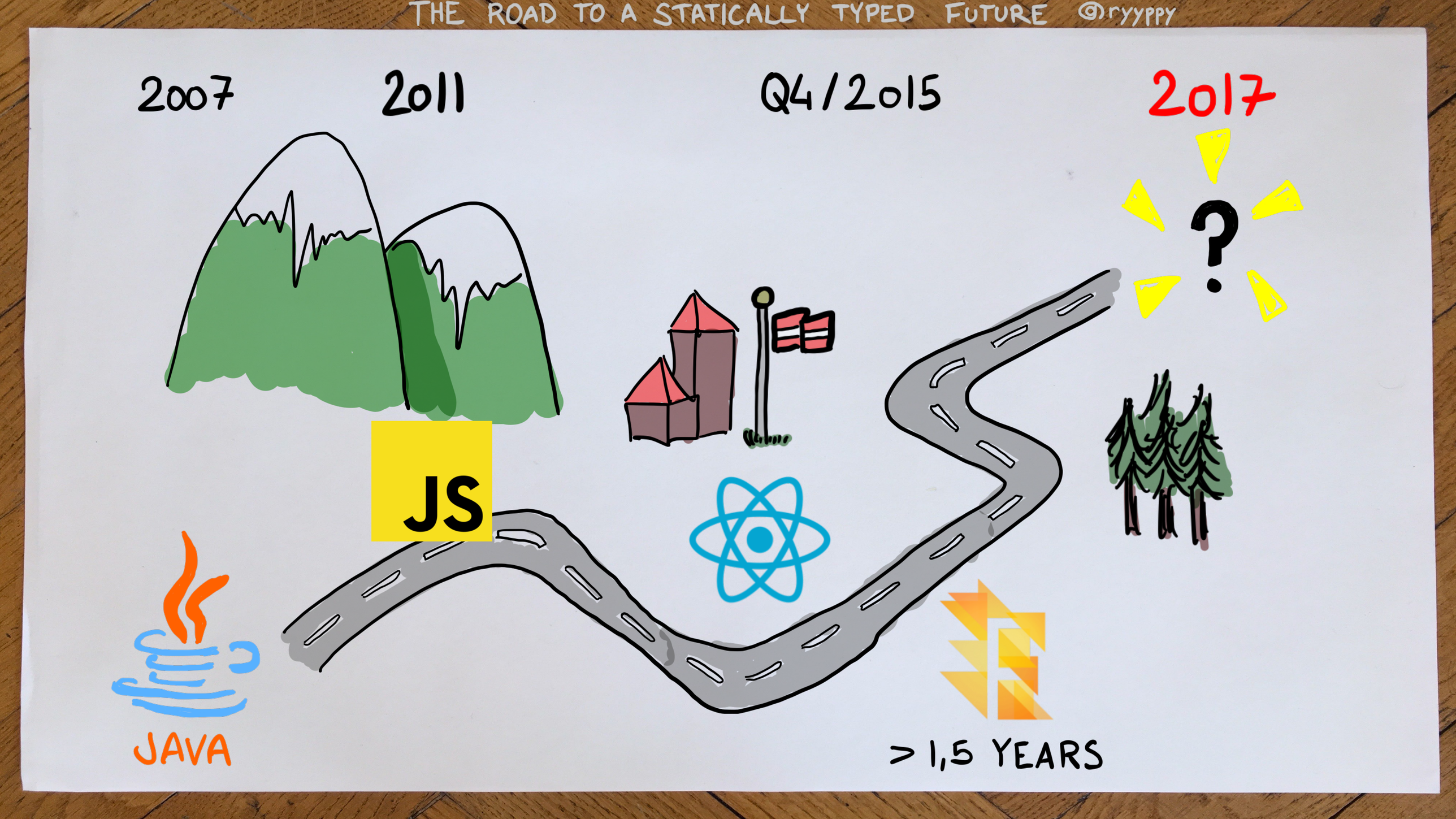
THE ROAD TO A STATICALLY TYPED FUTURE



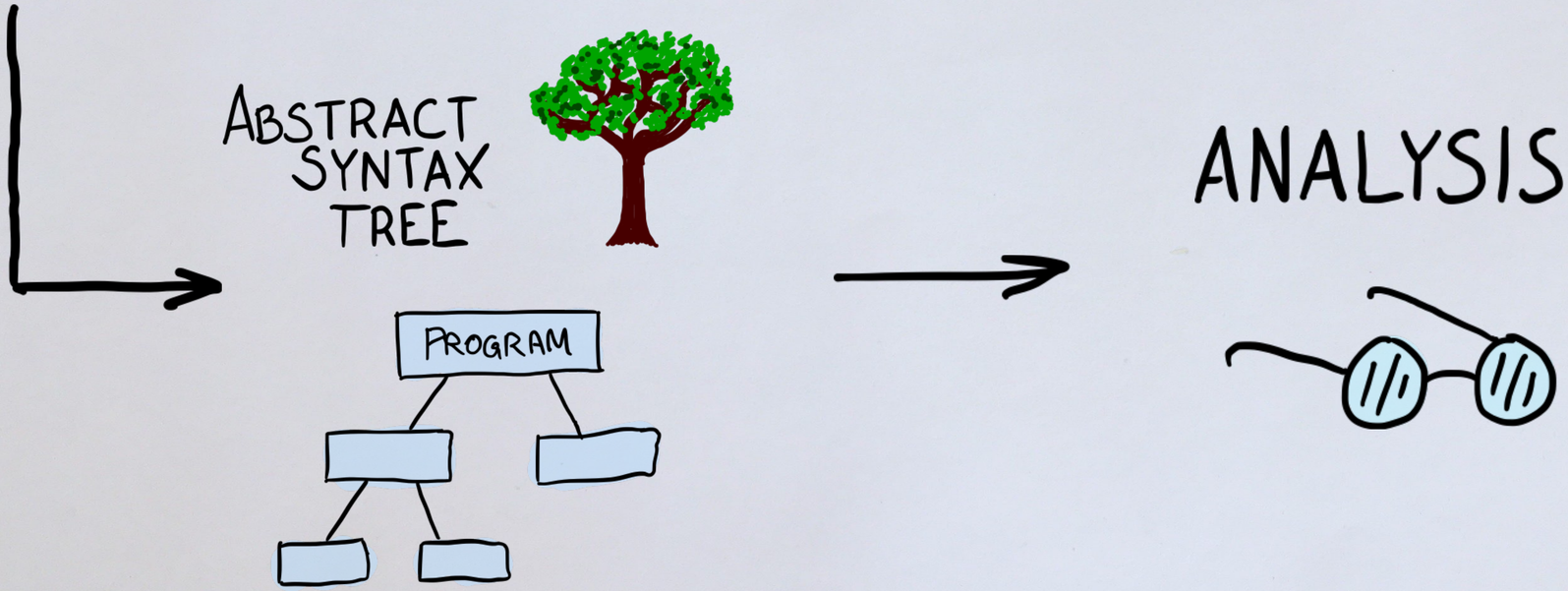
PATRICK STAPFER

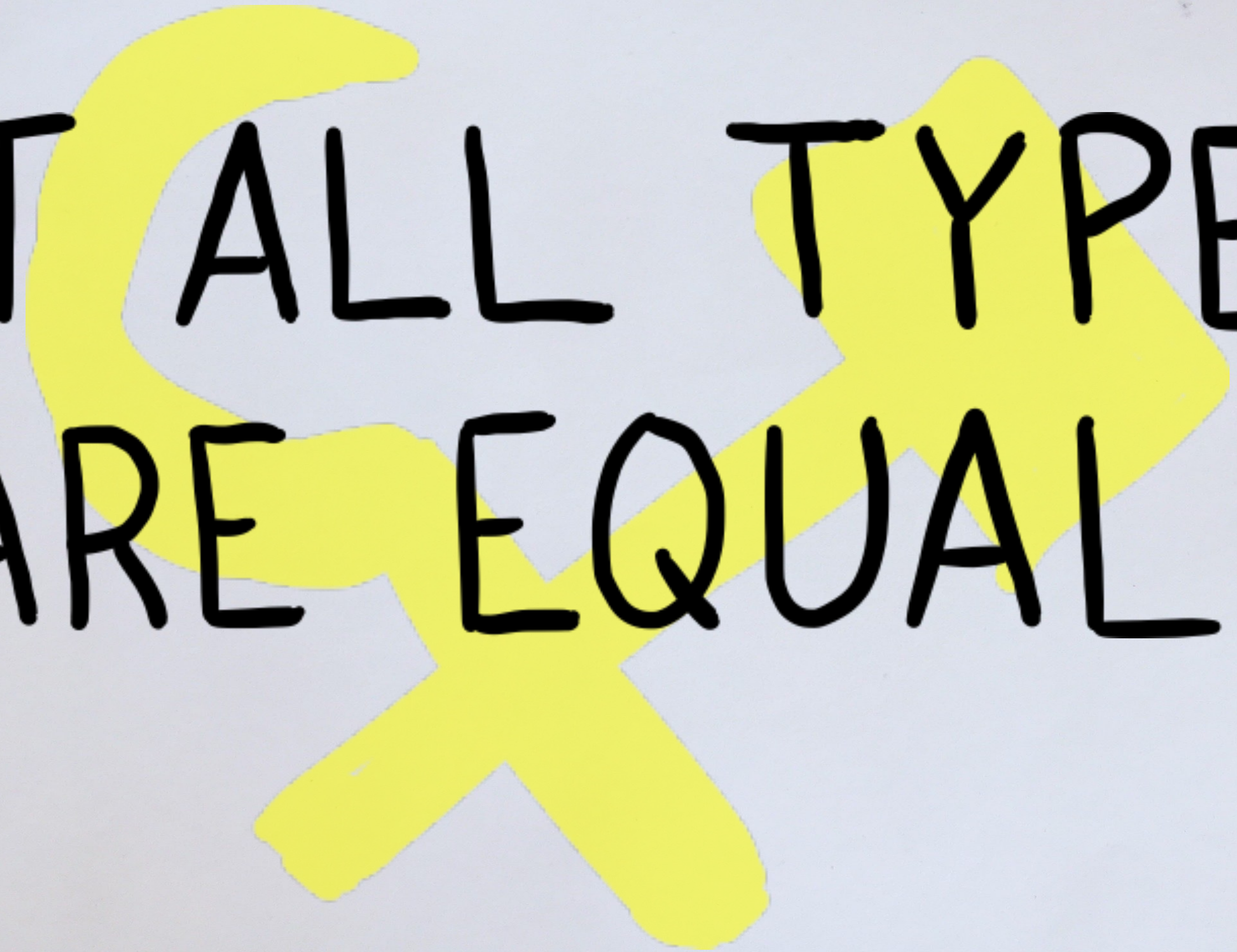
@ryyppy ryyppy


THE ROAD TO A STATICALLY TYPED FUTURE @ryyppy
2007
2011
Q4/2015
2017
?
JS
JAVA
> 1,5 YEARS

TYPE SYSTEM ?

```
function foo(a: number): string { .... }
```





NOT ALL TYPES ARE EQUAL

THE ROAD TO A STATICALLY TYPED FUTURE @ryyppy

TS

MAYBE TYPES	✓	?	✓
GENERICS	✓		✓
COVARIANCE	✓		✓
SUBTYPING	✓		✓
INFERENCE	✓	Quality	✓

RELIABLE TYPE INFERENCE
WILL INFLUENCE THE WAY
YOU WRITE CODE

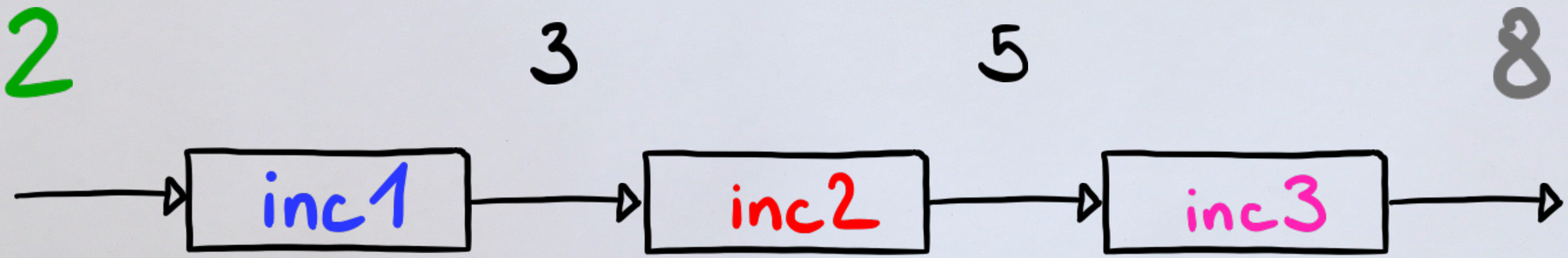

```
const inc = (n: number) => n + 1;
[1, 2, 3].map(inc);
```

```
const inc = (n) => n + 1;
[1, 2, 3].map(inc);
```

number | string

FUNCTIONAL PATTERNS

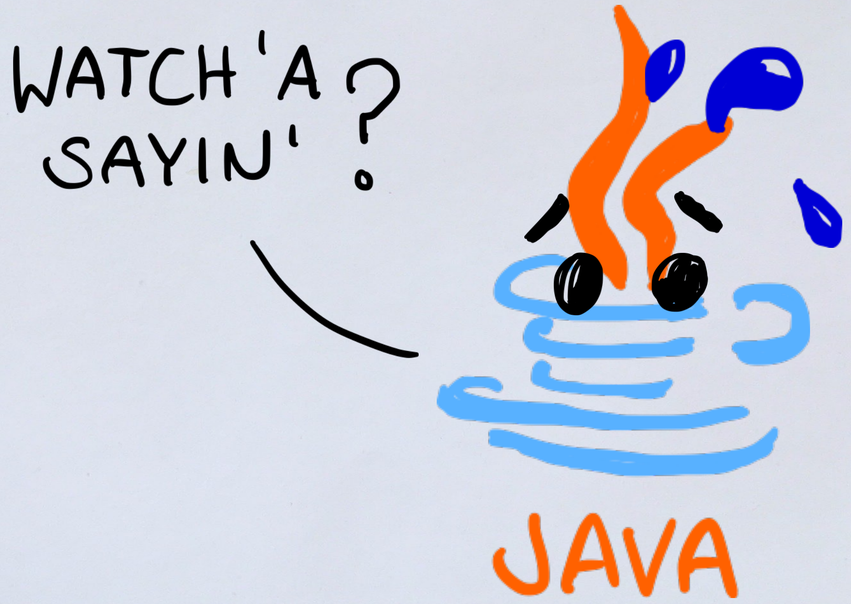
```
const inc6 = compose(inc1, inc2, inc3)

inc6(2)
```



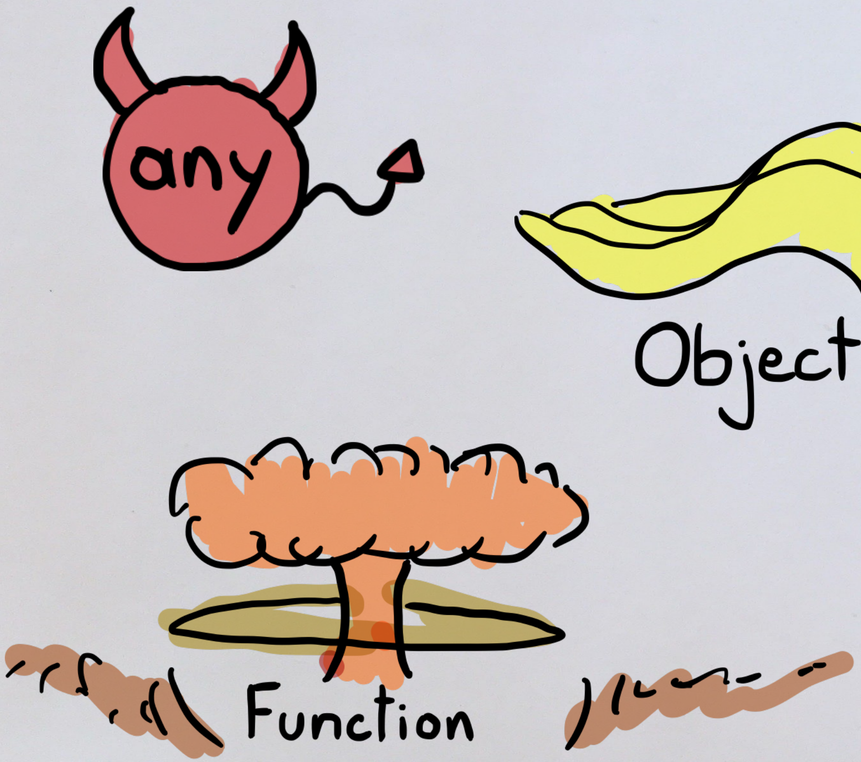
FUNCTIONAL PATTERNS

```
const Comp = (props: { user: string}) => {
    return (<div> {props.user} </div>);
};
```

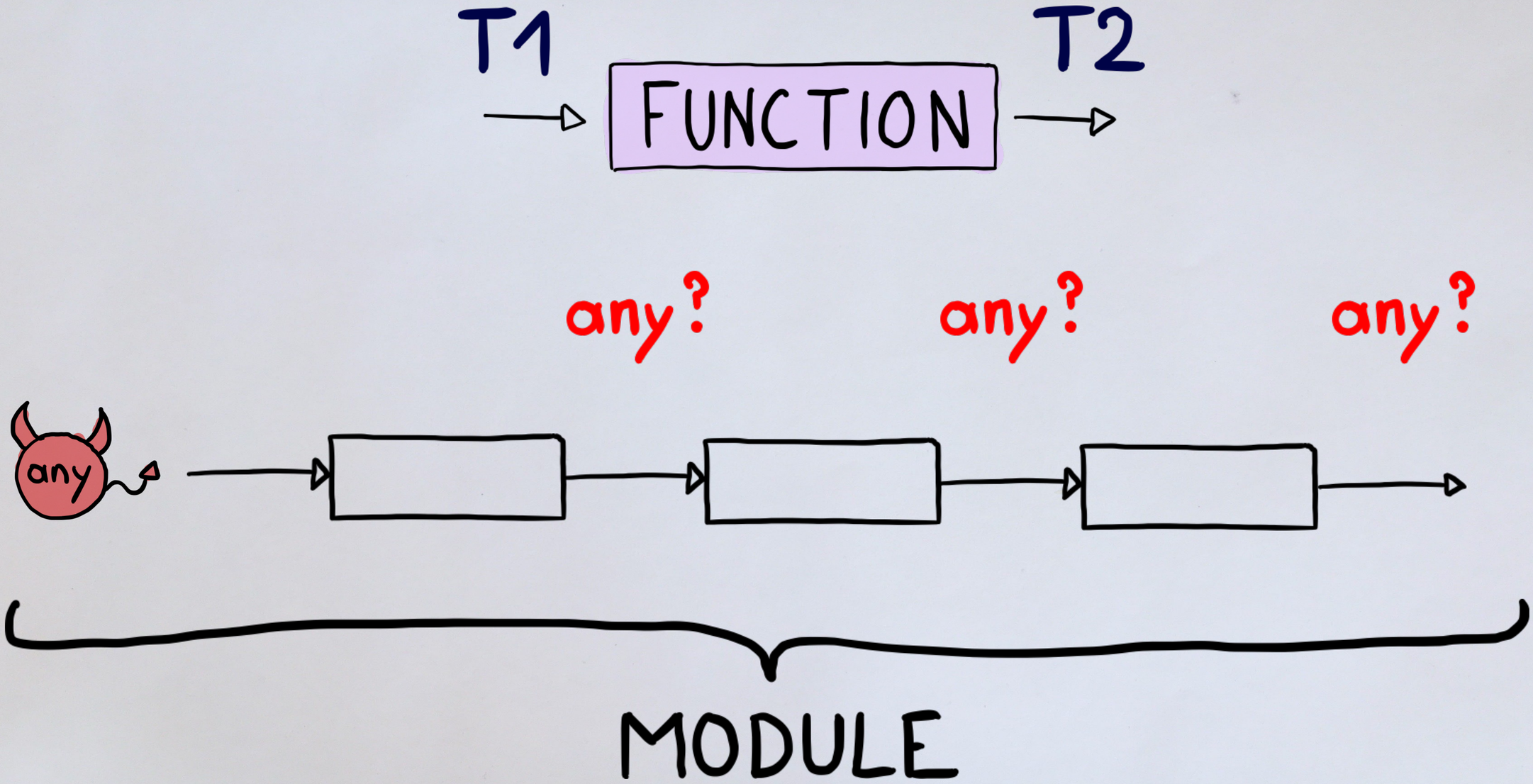
RELIABLE TYPE INFERENCE WILL MOTIVATE EXPERIMENTATION & CHANGES



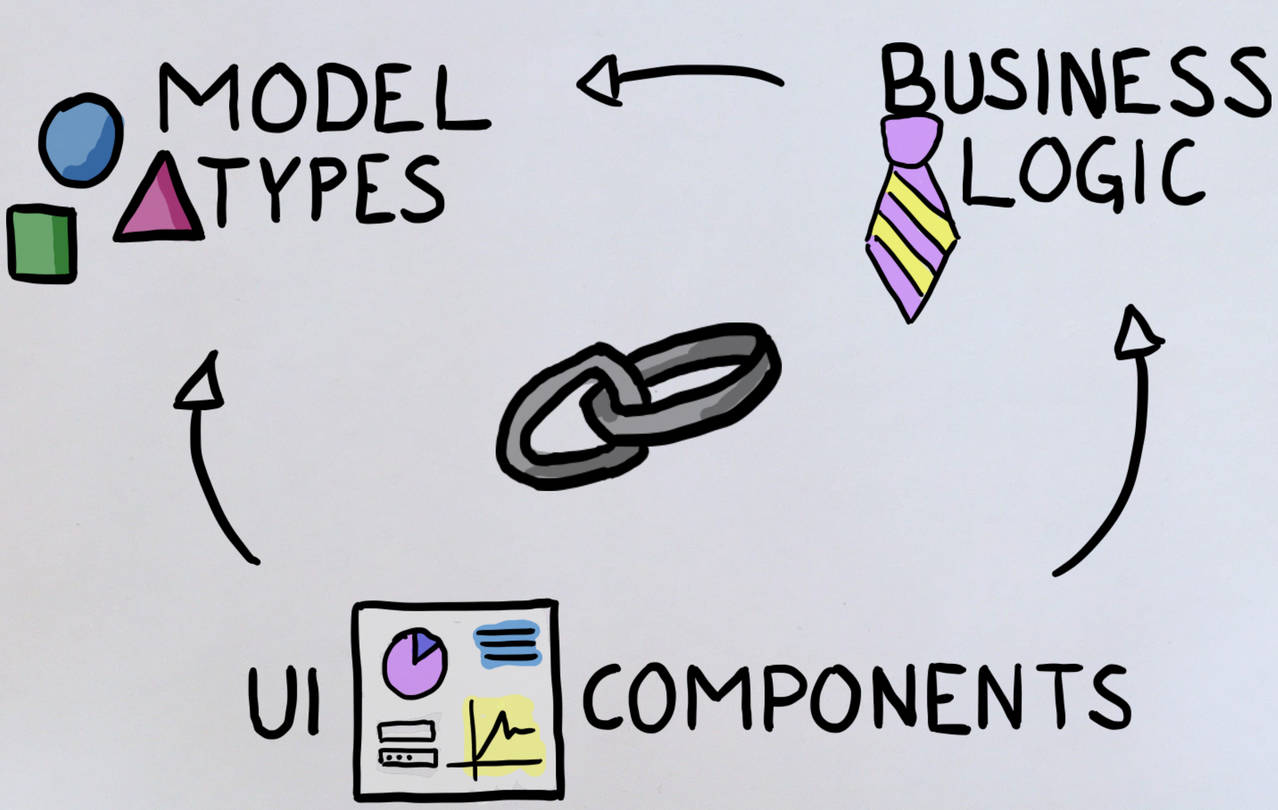
TRUST KILLERS

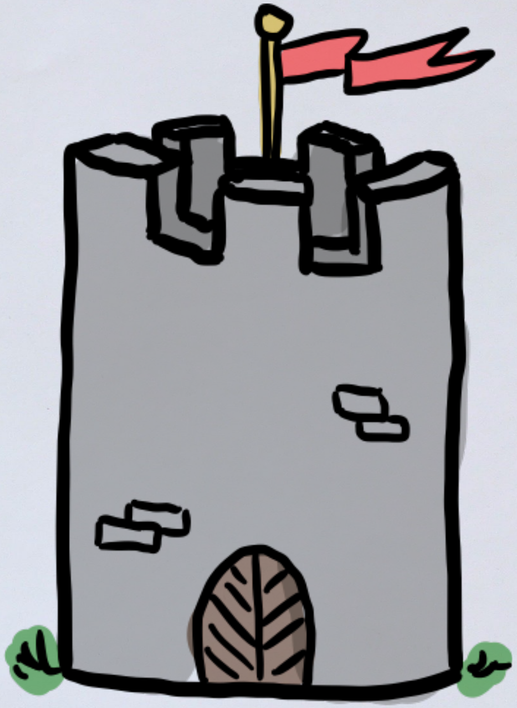



```
function foo(b: any) { ... }

const b: string = (someObj: any)
```

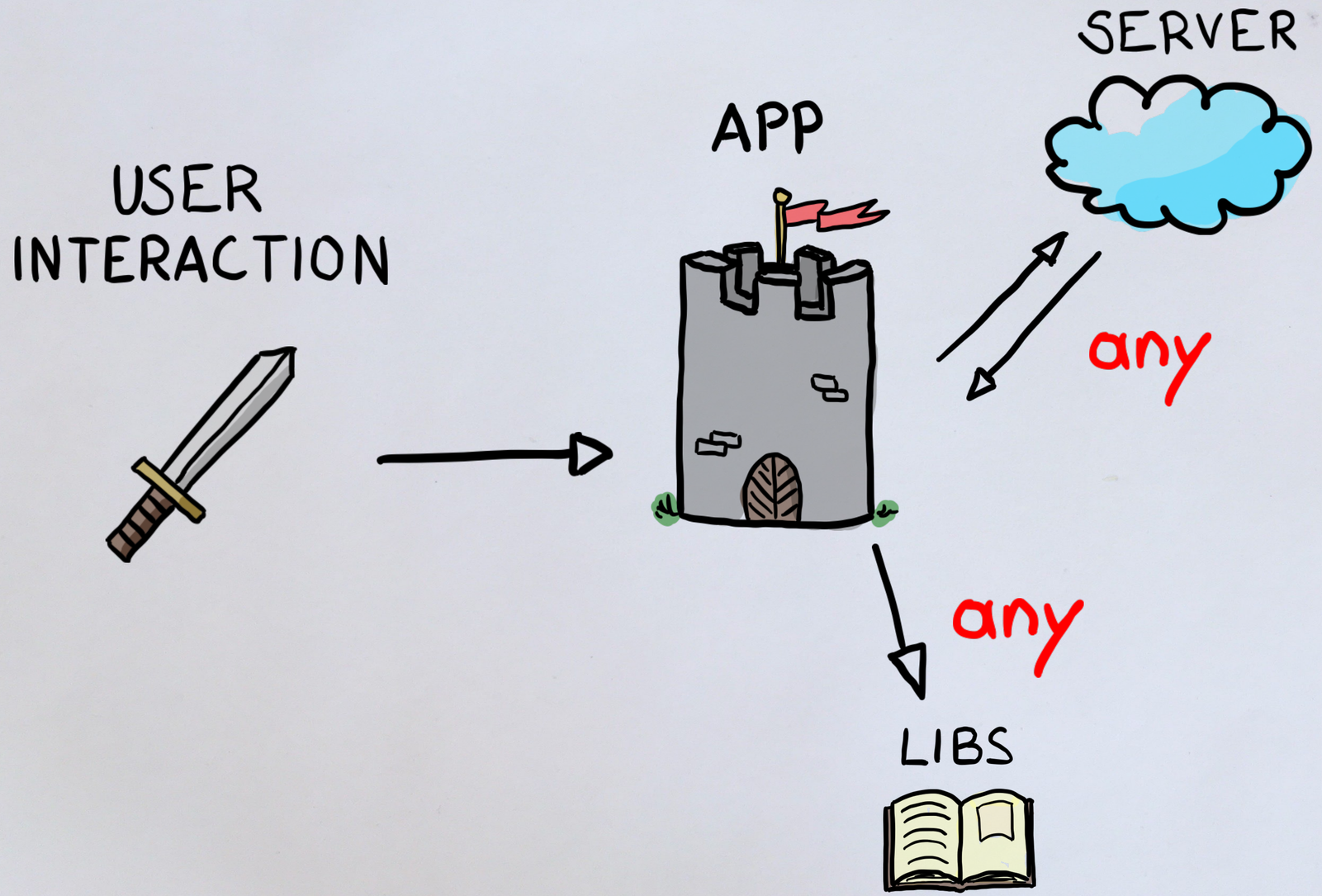

T1
FUNCTION
T2
any?
any?
any?
any
MODULE

APPLICATION



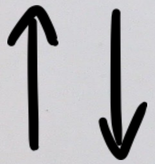


PARTLY TYPED ARCHITECTURE

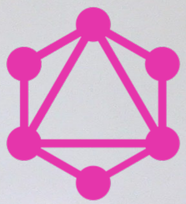


SERVER COMMUNICATION

REST



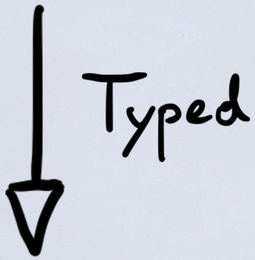
GRAPHQL

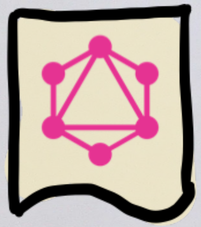


{JSON}

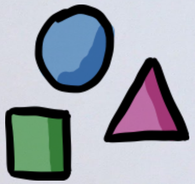
VALIDATOR

Typed

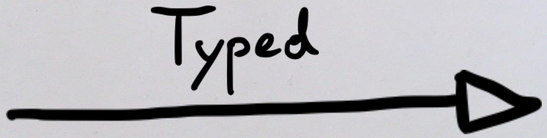


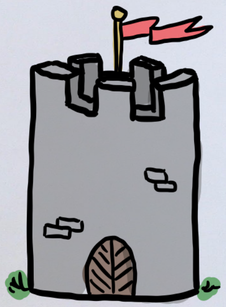


GEN



Typed





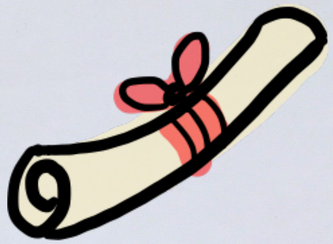
flow-runtime

andreypopp/validated

3RD PARTY LIBRARIES

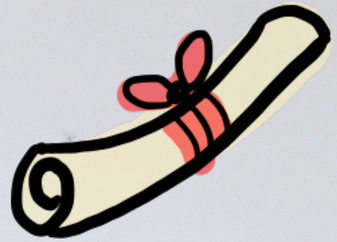
```
// flow-typed/npm/lodash_v4.x.x.js
declare module "lodash" {
    declare module.exports : {
        map : (...args : any) => any
    }
}
```

LIBDEF



flow-typed

create



publish

publish

npm

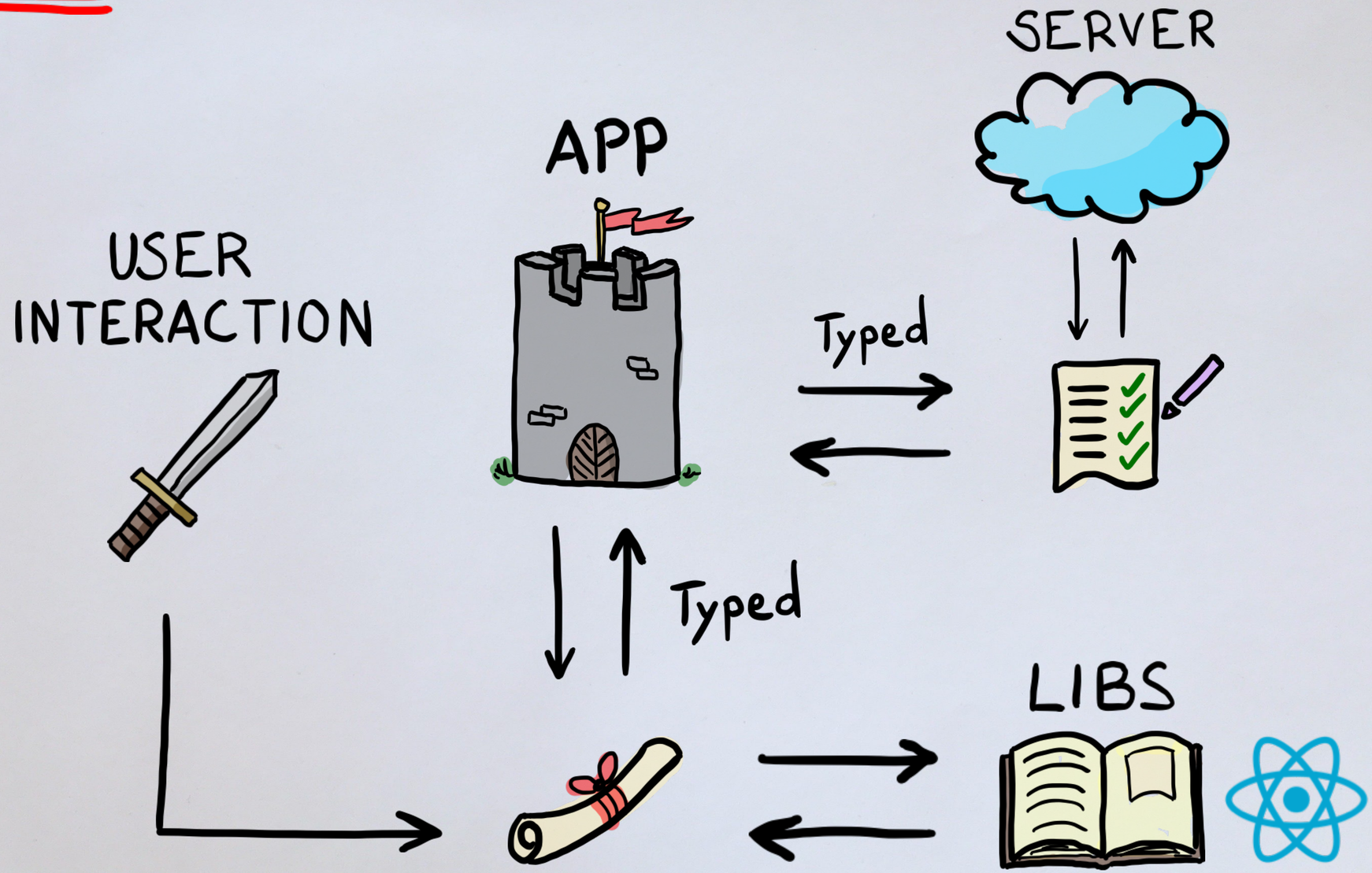
flow-typed

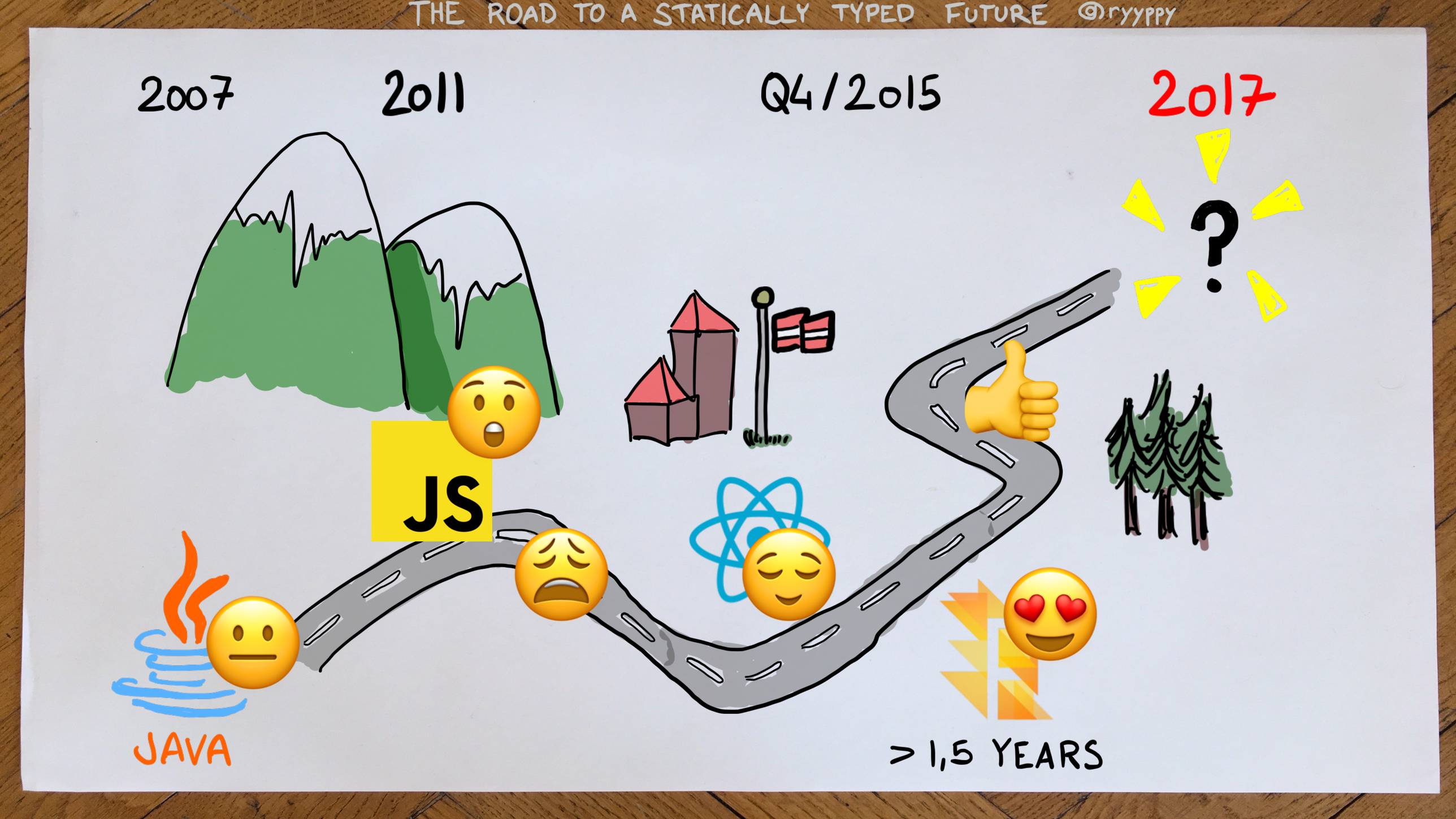
$ npm install

$ flow-typed install

PROJECT

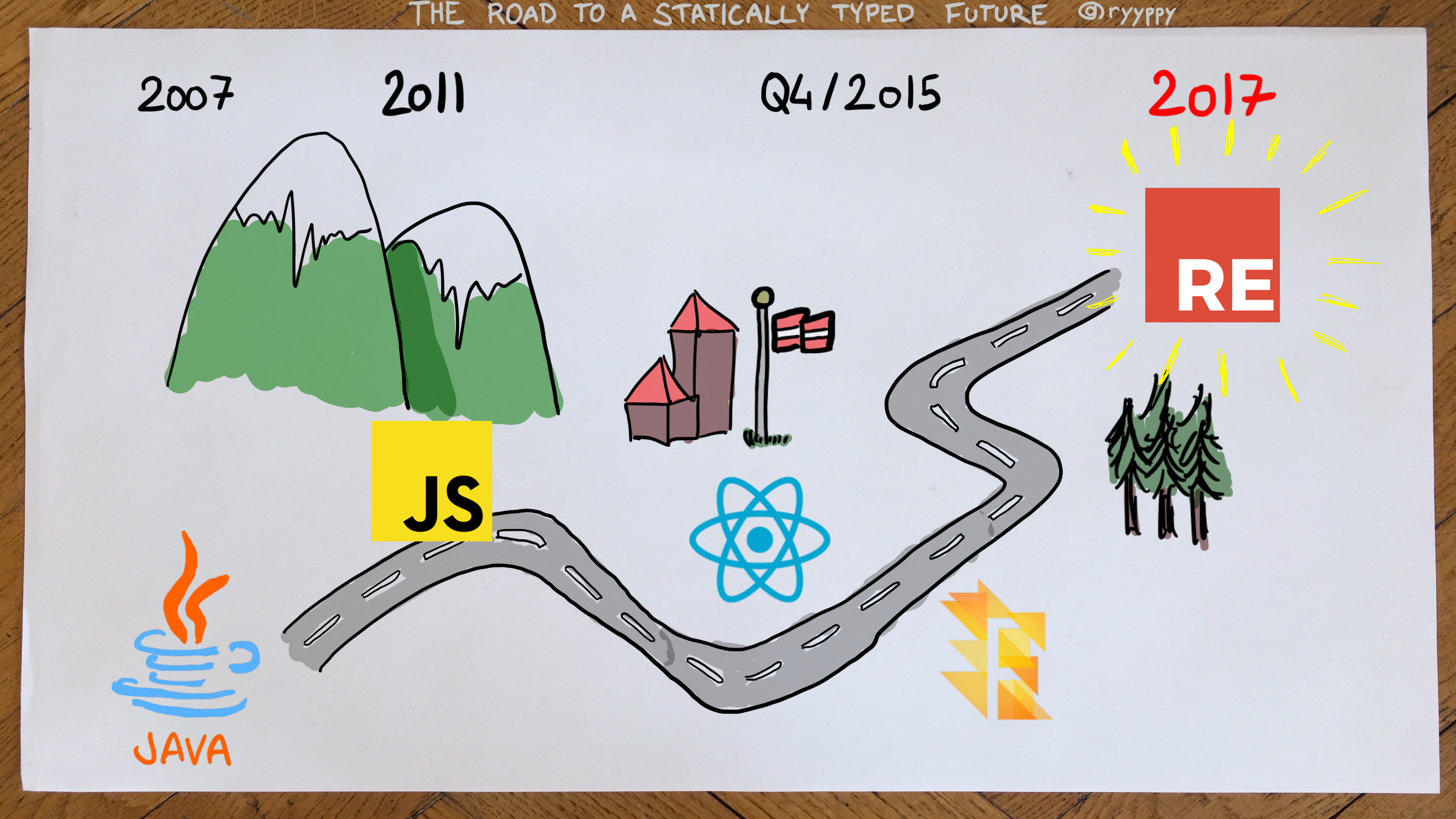
FULLY TYPED ARCHITECTURE



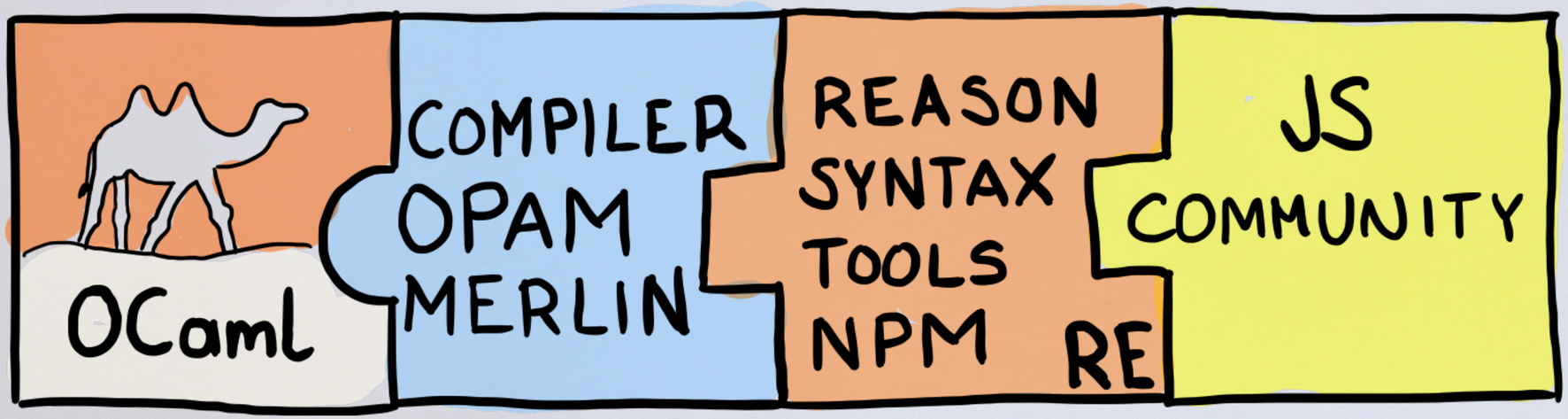

THE ROAD TO A STATICALLY TYPED FUTURE @ryyppy
2007
2011
Q4/2015
2017
?
JS
JAVA
> 1,5 YEARS

HOW TO IMPROVE ?

- COMPILE TO JS
- MORE FUNCTIONAL
- TYPESAFETINESS
- STRONG TOOLING
- FAMILIAR SYNTAX
- EASY INTEROP

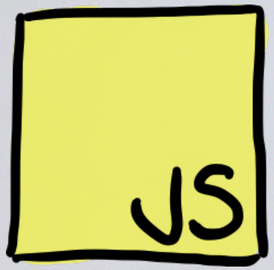

THE ROAD TO A STATICALLY TYPED FUTURE @ryyppy
2007
2011
Q4/2015
2017
JS
RE
JAVA

REASONML


OCaml
COMPILER
OPAM
MERLIN
REASON
SYNTAX
TOOLS
NPM
RE
JS
COMMUNITY

BUCKLESCRIPT


```
$ npm install -g bs-platform
```


JS

TOOLING


BABEL


ESLint


flow-typed


Lo


Ramda: Practical Functional Javascript


IMMUTABLE

TOOLING

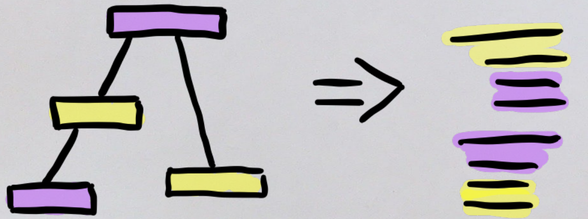
RE

~~SAFE~~

~~ESLint~~

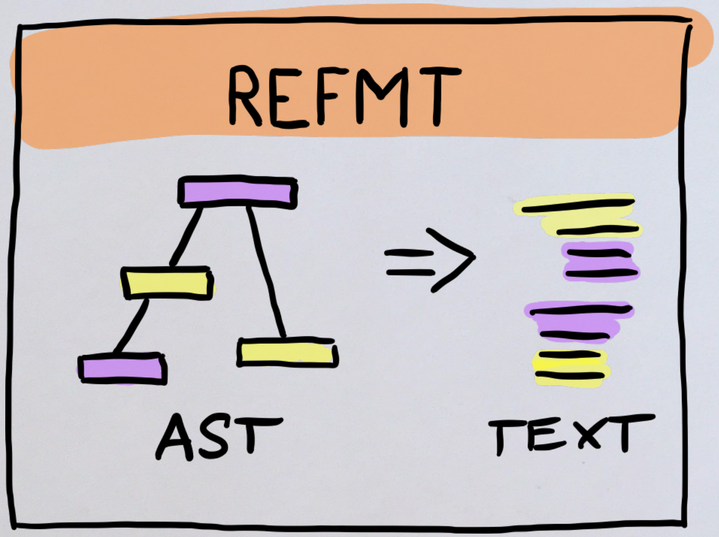
~~flow-typed~~

REFMT

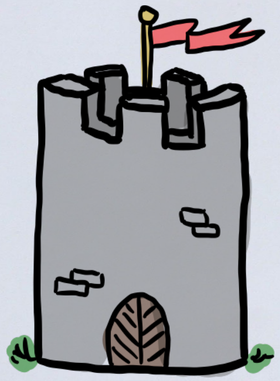


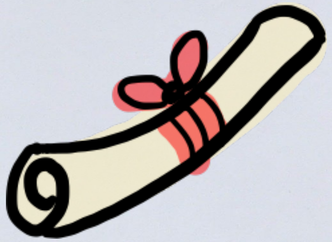
~~IMMUTABLE~~

TOOLING



RECAP





THANKS!

QUESTIONS?

PATRICK STAPFER

@ryyppy

ryyppy