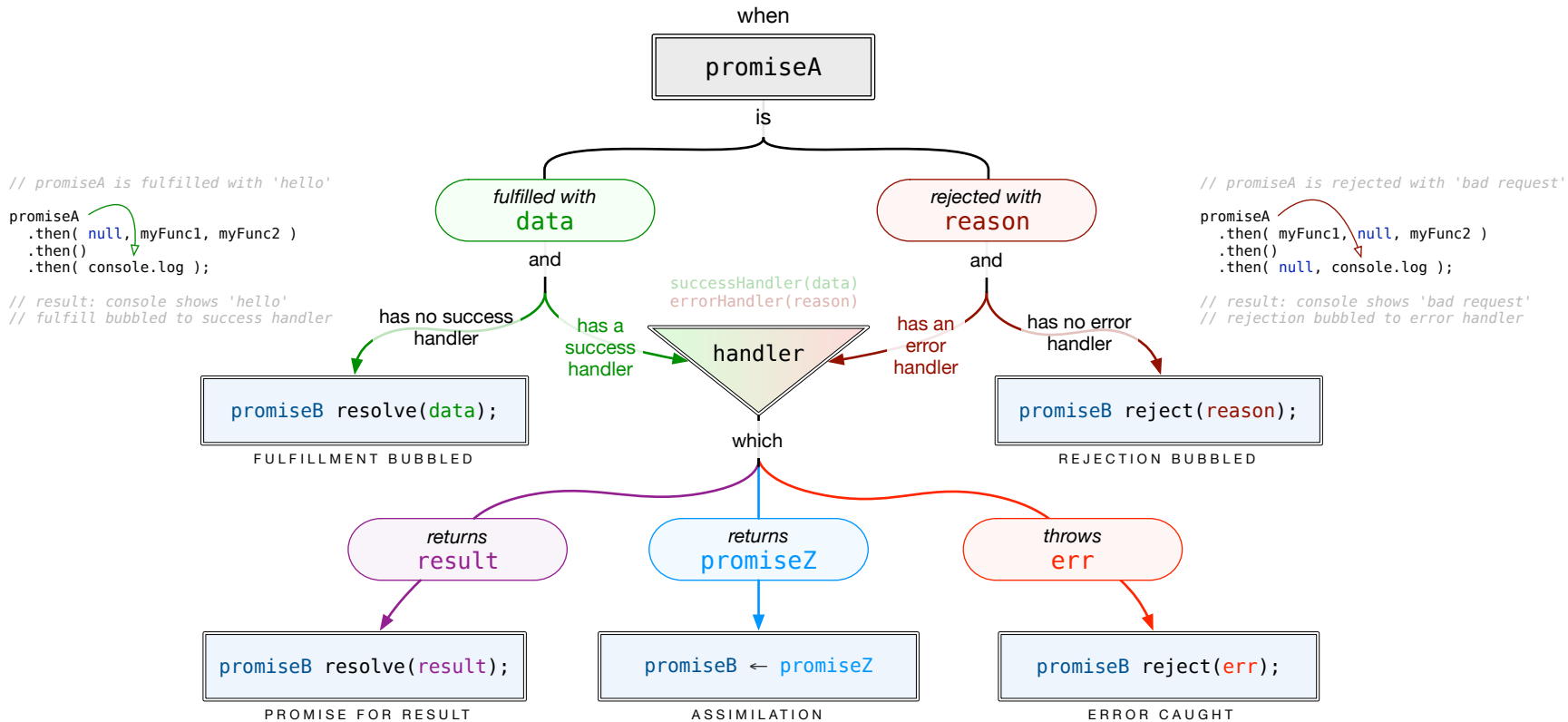


.then() always returns a new promise. What happens to that promise?

```
promiseB = promiseA.then( [successHandler], [errorHandler] );
```



// promiseForVal1 is fulfilled with 999

```
promiseForVal2 = promiseForVal1
  .then( function success (val1) {
    val2 = val1++;
    return val2;
  });
```

// result: promiseForVal2 is fulfilled with 1000.

// same idea, shown in a direct chain:

```
promiseForVal1
  .then( function success (val1) {
    // do some code to make val2
    return val2;
  })
  .then( function success (val2) {
    console.log( val2 );
  });
```

// output promise "becomes" returned promise

```
promiseForMessages = promiseForUser
  .then( function success (user) {
    // do some code to make a new promise
    return promiseForMessages;
  });
```

// same idea, shown in a direct chain:

```
promiseForUser
  .then( function success (user) {
    // do some code to make a new promise
    return promiseForMessages;
  })
  .then( function success (messages) {
    console.log( messages );
  });
```

// promiseForVal2 will be rejected with error '404':

```
promiseForVal2 = promiseForVal1
  .then( function success (val1) {
    // THROWN ERROR '404' trying to make val2
    return val2;
  });
```

// same idea, shown in a direct chain:

```
promiseForVal1
  .then( function success (val1) {
    // THROWN ERROR '404' trying to make val2
    return val2;
  })
  .then( null, function failed (err) {
    // handle the error as you prefer
    console.log('Oops!', err);
  });
```

// console shows 'Oops! 404'