Introduction to the *-ilities

Memi Lavi
www.memilavi.com

*-ilities = Quality Attributes

Technical capabilities that should be used in order to fulfill the non-functional requirements

Example

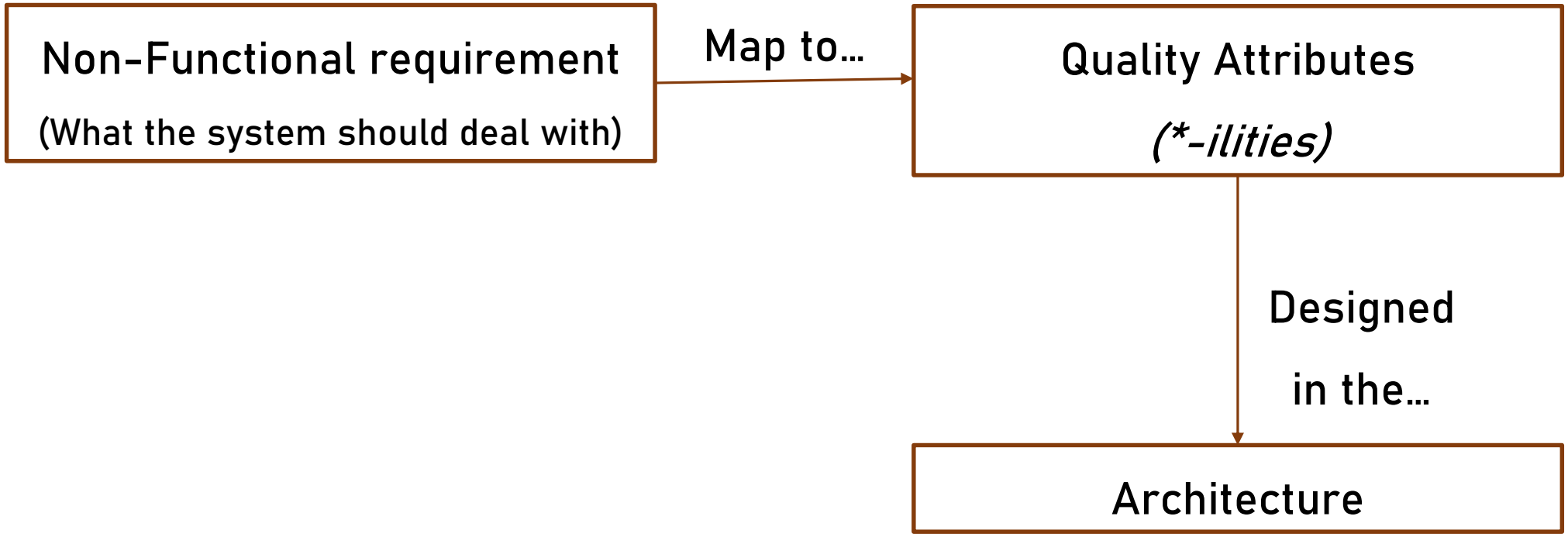
- Non-Functional requirement:

  "The system must work under heavy load, but should not waste money on unused resources"

- Required Quality Attribute:

  "Scalability"

Relationships



Quality attributes [edit]

Notable quality attributes include:

- accessibility
- accountability
- accuracy
- adaptability
- administrability
- affordability
- agility [Toll] (see Common Subsets below)
- auditability
- autonomy [Erl]
- availability
- compatibility
- composability [Erl]
- configurability
- correctness
- credibility
- customizability
- debugability
- degradability
- determinability
- demonstrability
- dependability
- deployability
- mobility
- modifiability
- modularity
- operability
- orthogonality
- portability
- precision
- predictability
- process capabilities
- producibility
- provability
- recoverability
- relevance
- reliability
- repeatability
- reproducibility
- resilience
- responsiveness
- reusability [Erl]
- robustness
- safety
- scalability

Source: https://en.wikipedia.org/wiki/List_of_system_quality_attributes

Our *-ilities

- Scalability
- Manageability
- Modularity
- Extensibility
- Testability

Scalability

“Adding computing resources without any interruption”

Non-Scalable System

- Look for non-scalable code
- Rewrite non-scalable code
- Reinforce VM

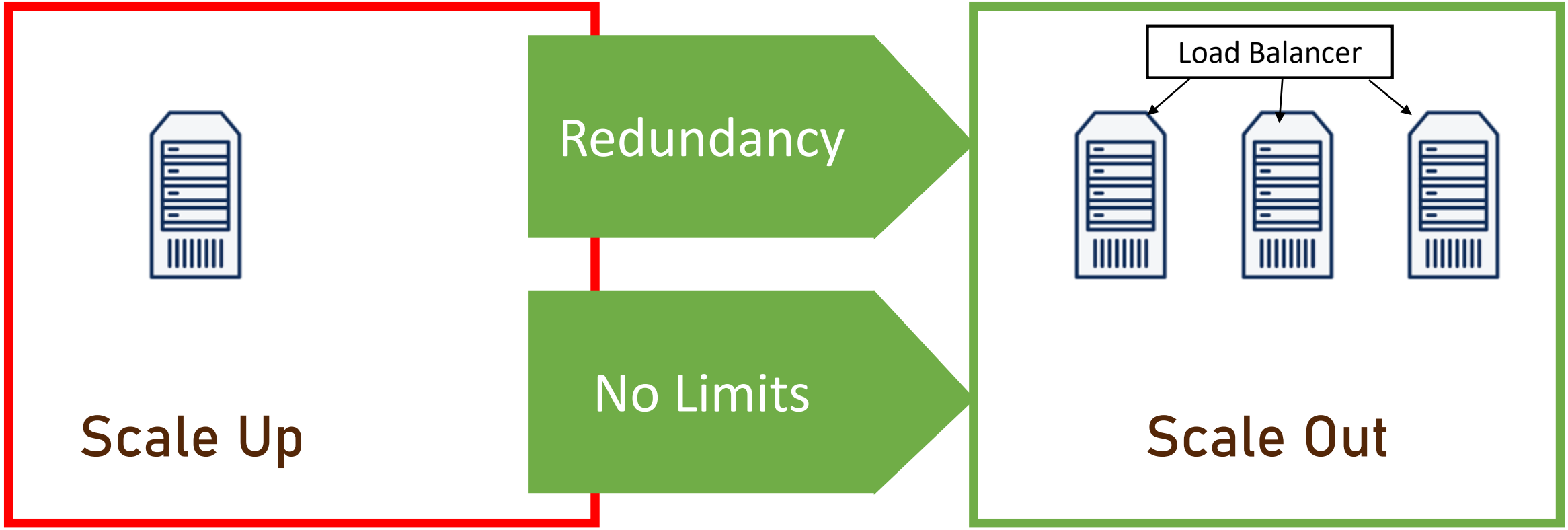
Long, Cumbersome

Scalable System

- Add VM
- Notify the Load Balancer

That's It!

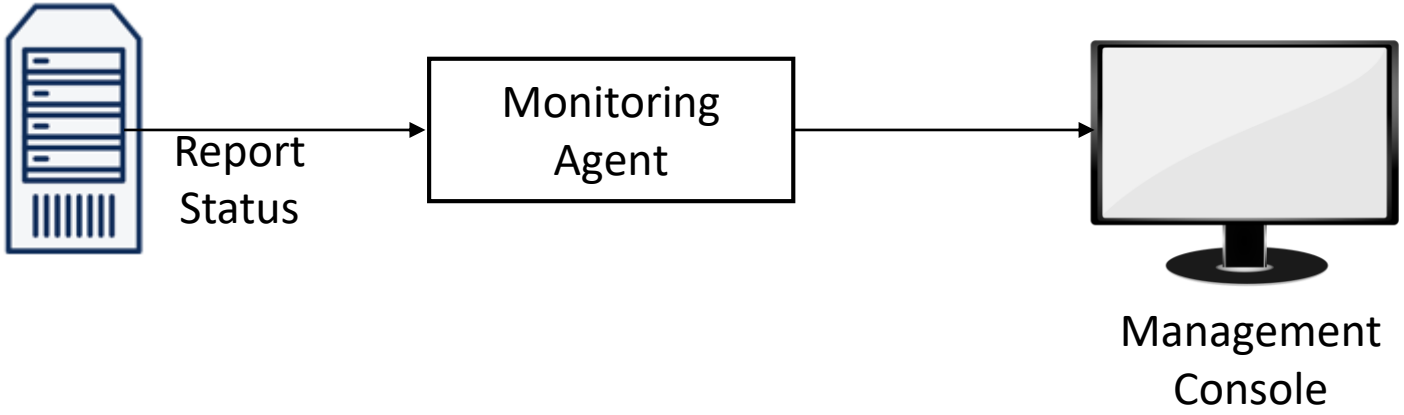
Scalability Types



Manageability

“Know what’s going on and take actions accordingly”

Manageable System

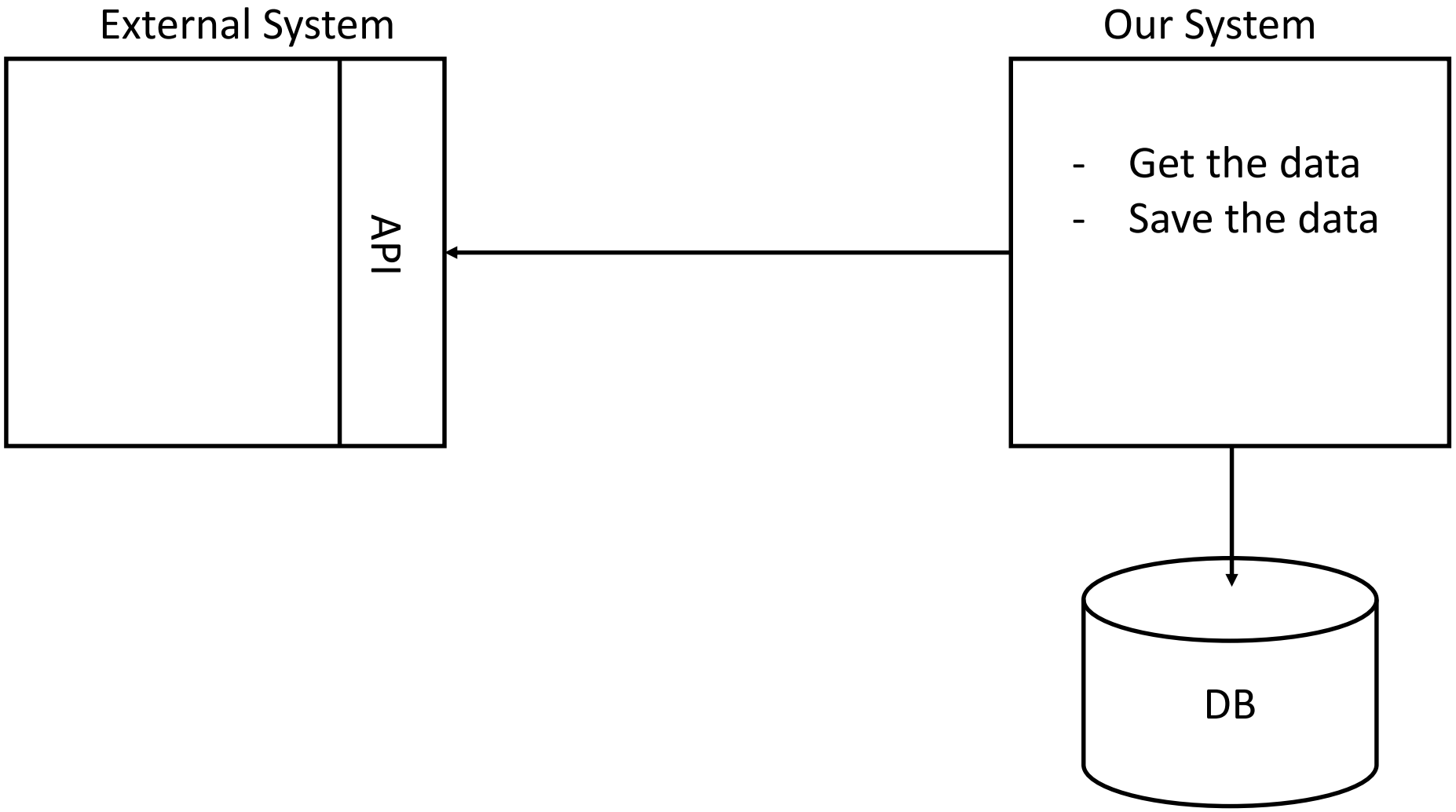


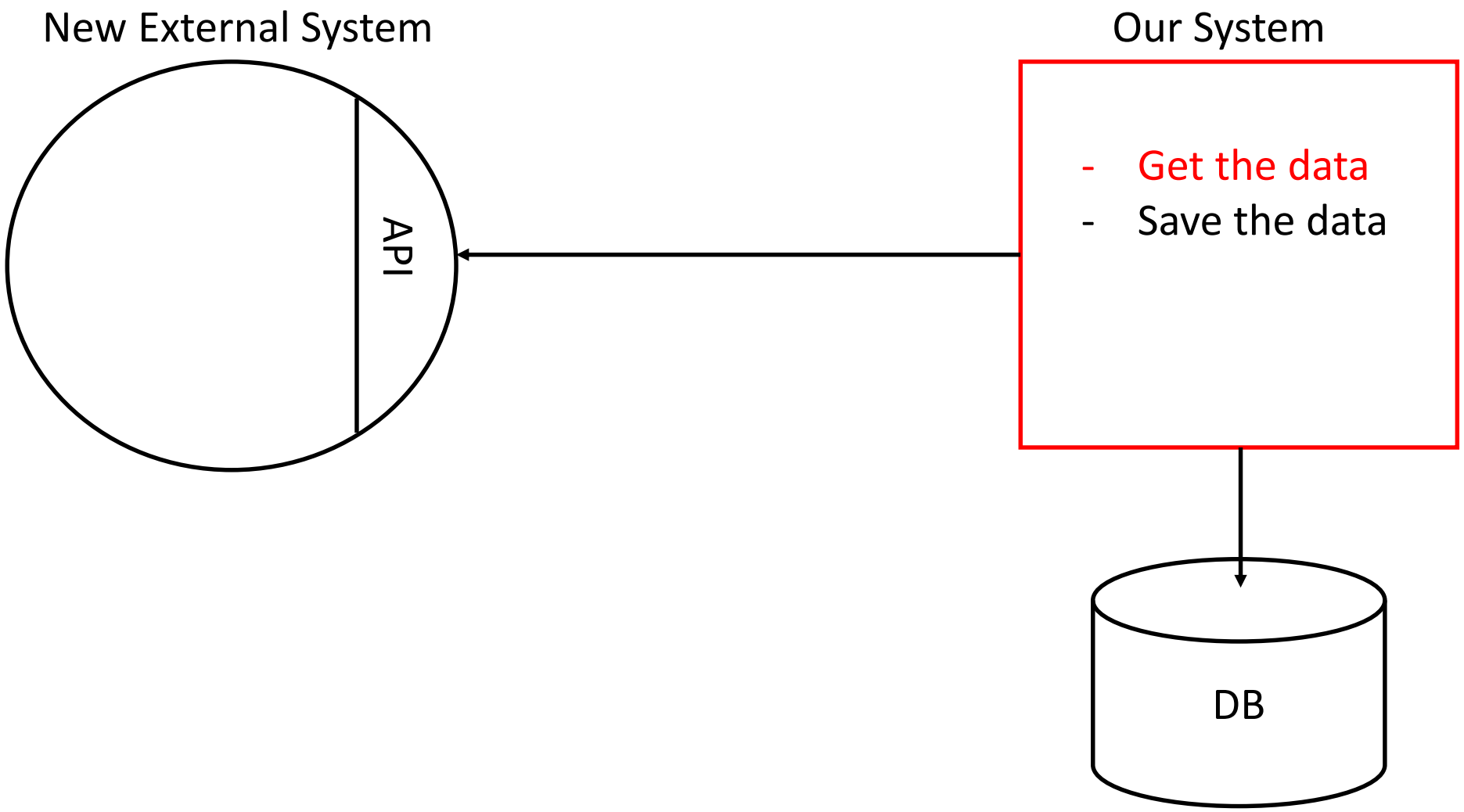
Is Your System Manageable?

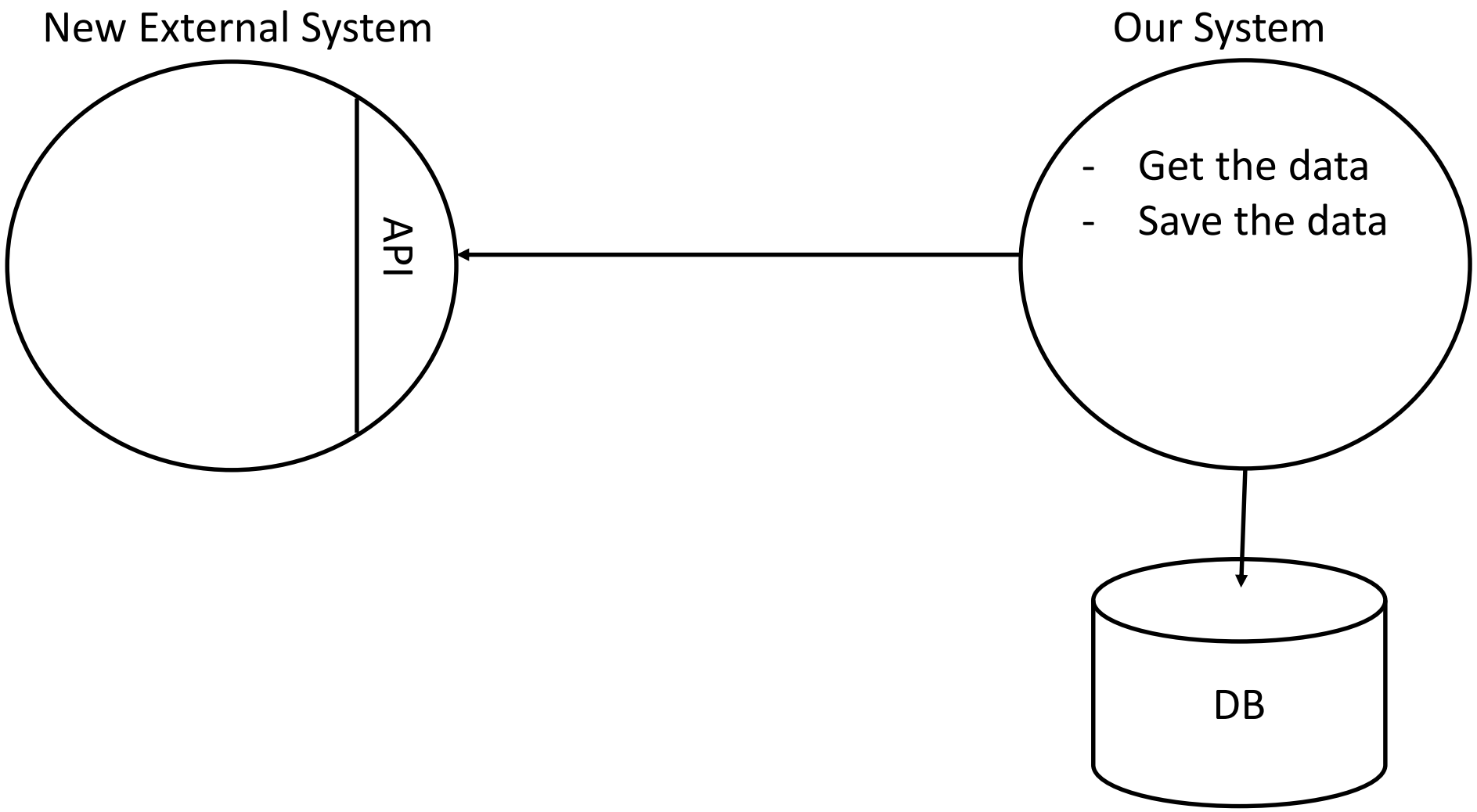
Who reports the problems?

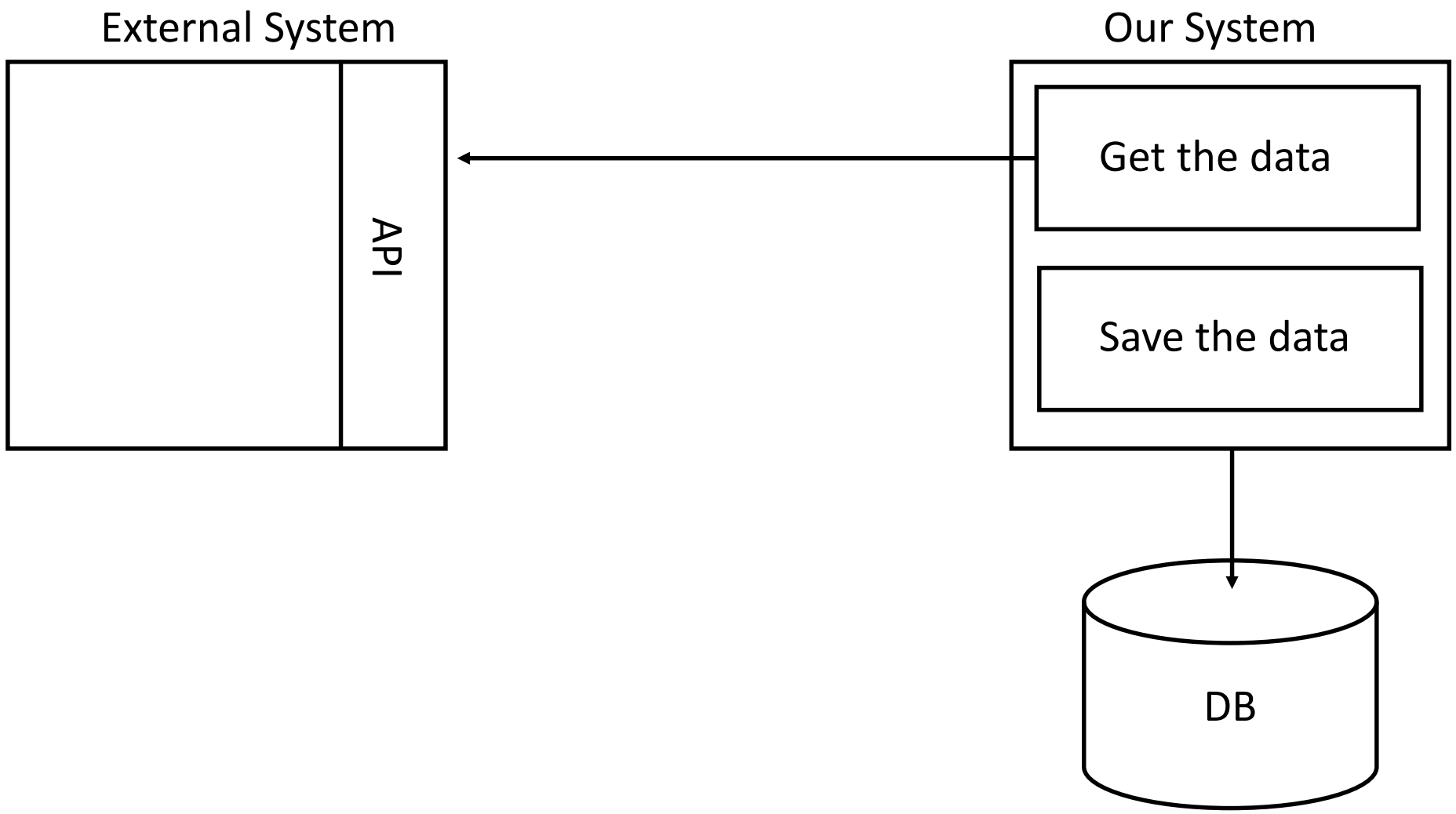
Modularity

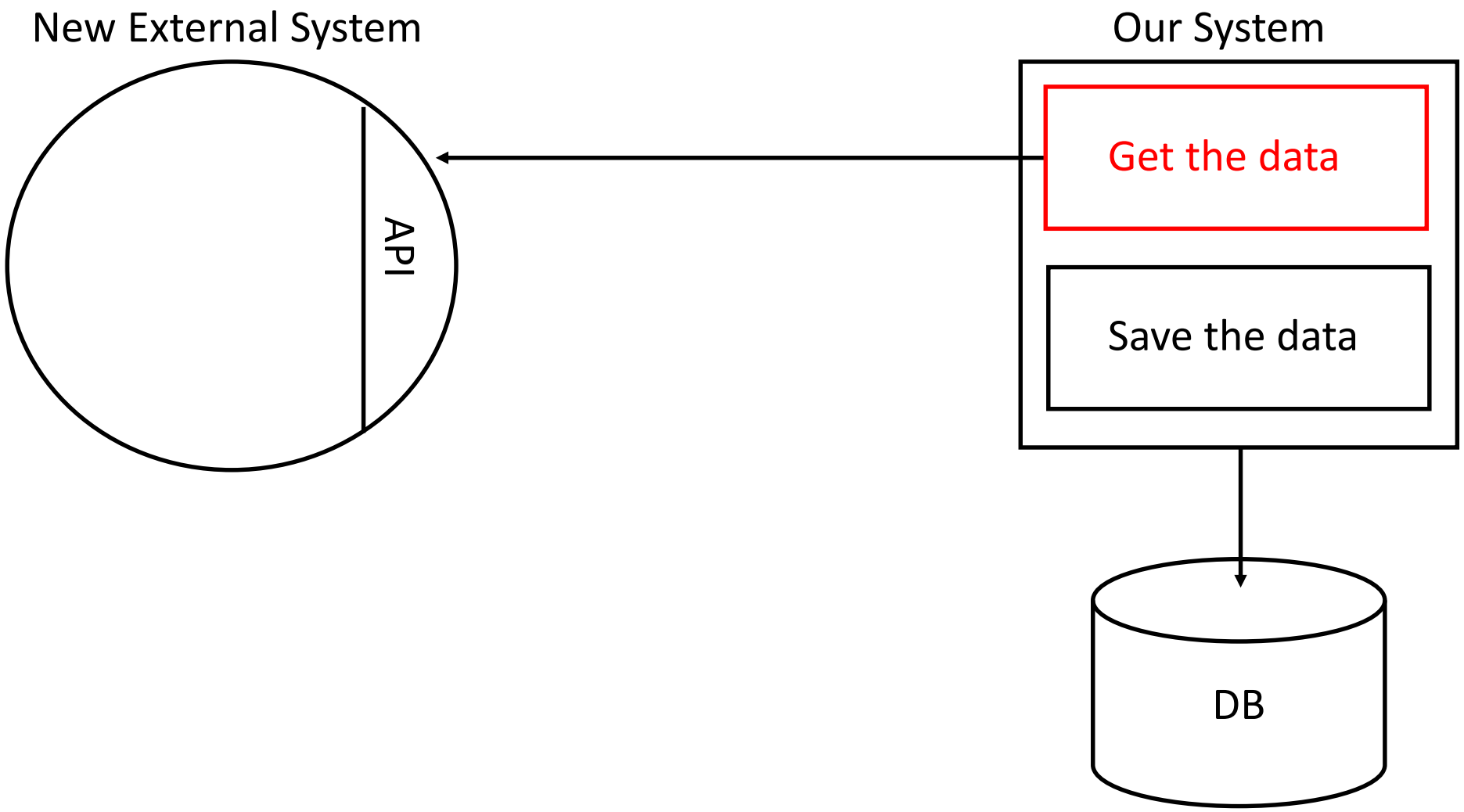
"A system that is built from building blocks, that can be changed or replaced without affecting the whole system."


External System
API
Our System
- Get the data
- Save the data
DB

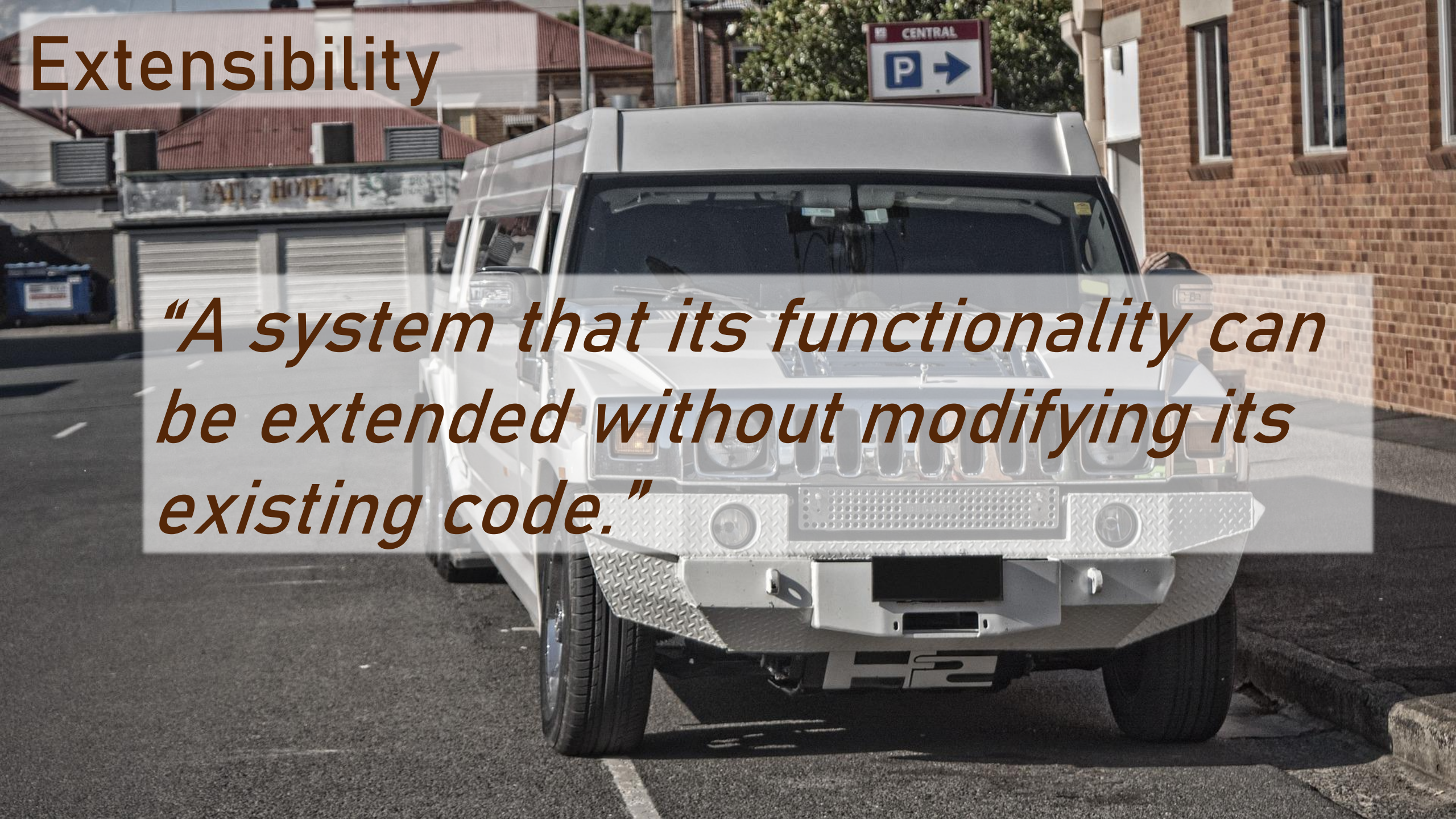

New External System
API
Our System
- Get the data
- Save the data
DB


New External System
API
Our System
- Get the data
- Save the data
DB


External System
API
Our System
Get the data
Save the data
DB


New External System
API
Our System
Get the data
Save the data
DB

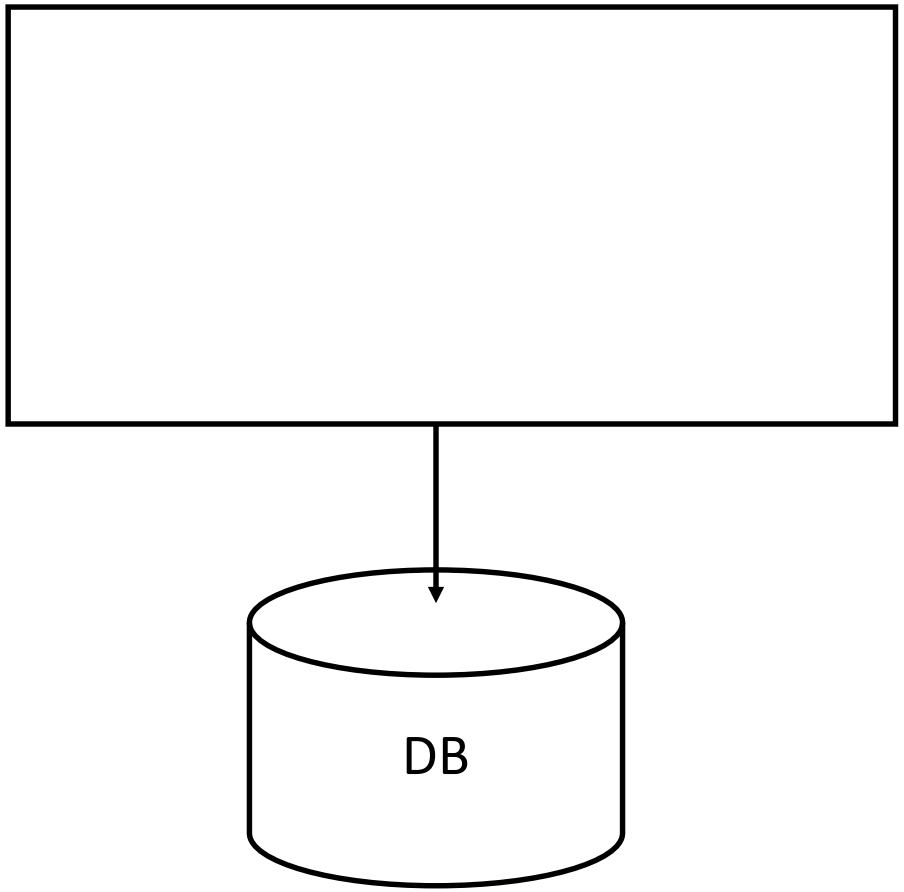
Extensibility



"A system that its functionality can be extended without modifying its existing code."

API

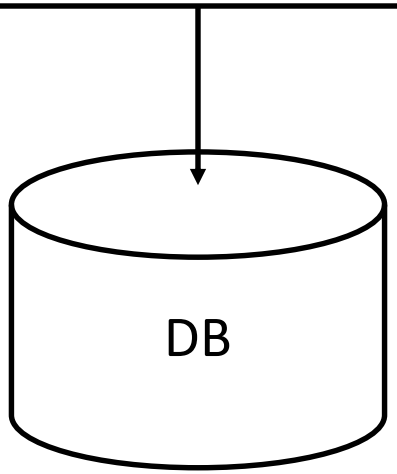
/api/query?format=[XML|JSON]



API

/api/query?format=[XML|JSON|CSV]

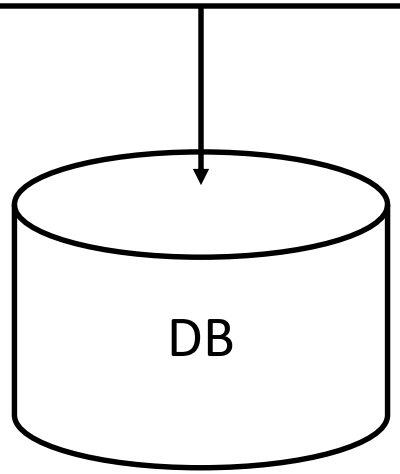
```
switch (format)  {
  case “xml”:
     return formatXml(data);
  case “json”:
     return formatJson(data);
}
```



API

/api/query?format=[XML|JSON|CSV]

```
switch (format)  {
  case “xml”:
     return formatXml(data);
  case “json”:
     return formatJson(data);
  case “csv”:
     return formatCsv(data);
}
```

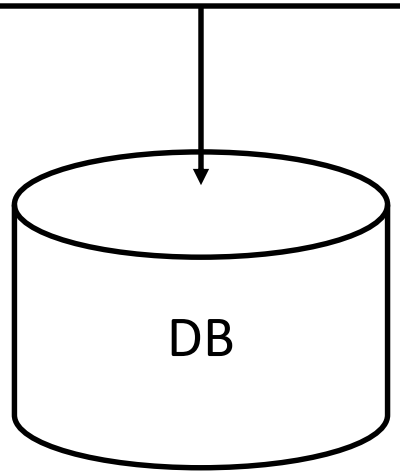


API

/api/query?format=[XML|JSON|CSV]

```
String formatQuery(string format,
                   string data)  {
    IFormatter formatter =
              GetFormatter(format);

    return formatter.Format(data);
}
```



Testability

"How easy it to test the application?"

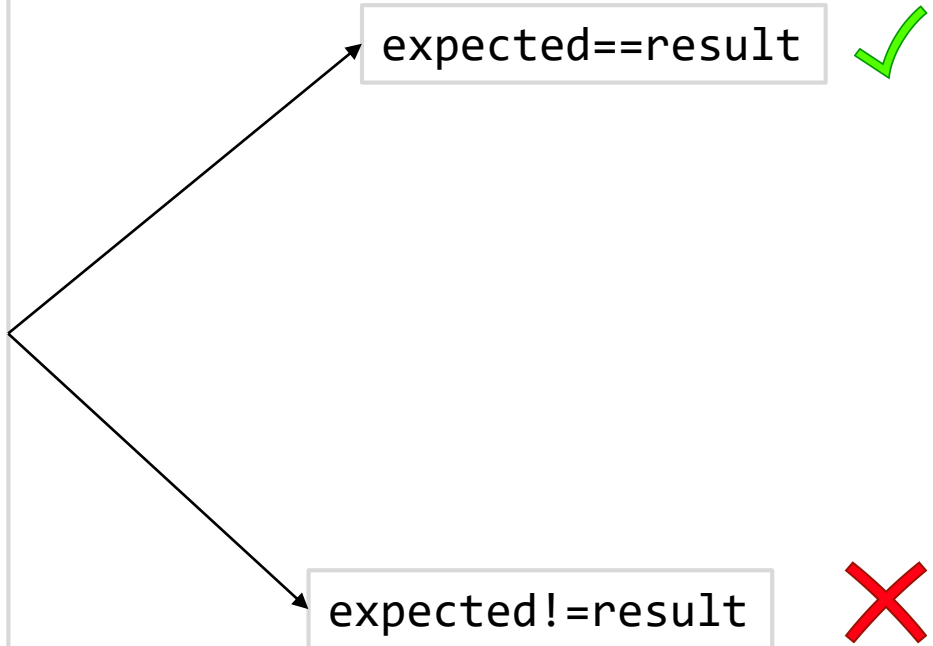
Types of Testing

- Manual
- Unit Testing
- Integration Testing

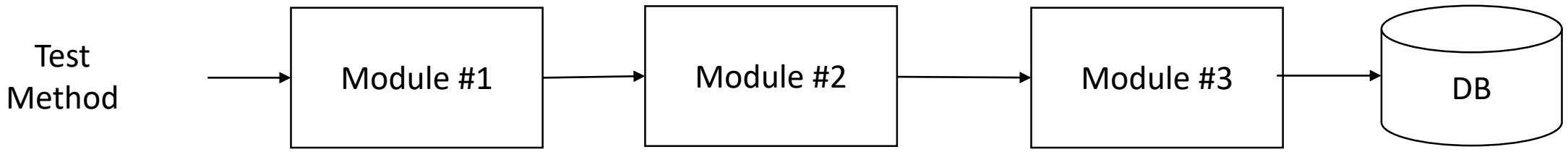
Unit Testing

```
int Add(int x, int y)  {
    return x+y;
}
```

```
[TestMethod]
public void Add_Positives() {
    // arrange
    int num1 = 9;
    int num2 = 5;
    int expected = 14;
    var calc = new Calculator();
    // act
    var result=calc.Add(num1,num2);
    // assert
    Assert.AreEqual(expected, result);
}
```



Integration Testing



Testability

- Easy to test using:
  - Unit Testing
  - Integration Testing

Testability's Characteristics

- Independent modules and methods
- Single responsibility

Single Responsibility

```
int Add(int x, int y)  {
    if (x>=0 && y>=0)  {
        return x+y;
    }
}
```

Single Responsibility

```
int Add(int x, int y)  {
   if (CheckForPositive(x,y))  {
      return x+y;
   }
}
```

*-ilities - Summary

- Non-Functional requirements -> *-ilities
- Represent technical capabilities
- Most common:
  - Scalability
  - Manageability
  - Modularity
  - Extensibility
  - Testability

https://en.wikipedia.org/wiki/List_of_system_quality_attributes