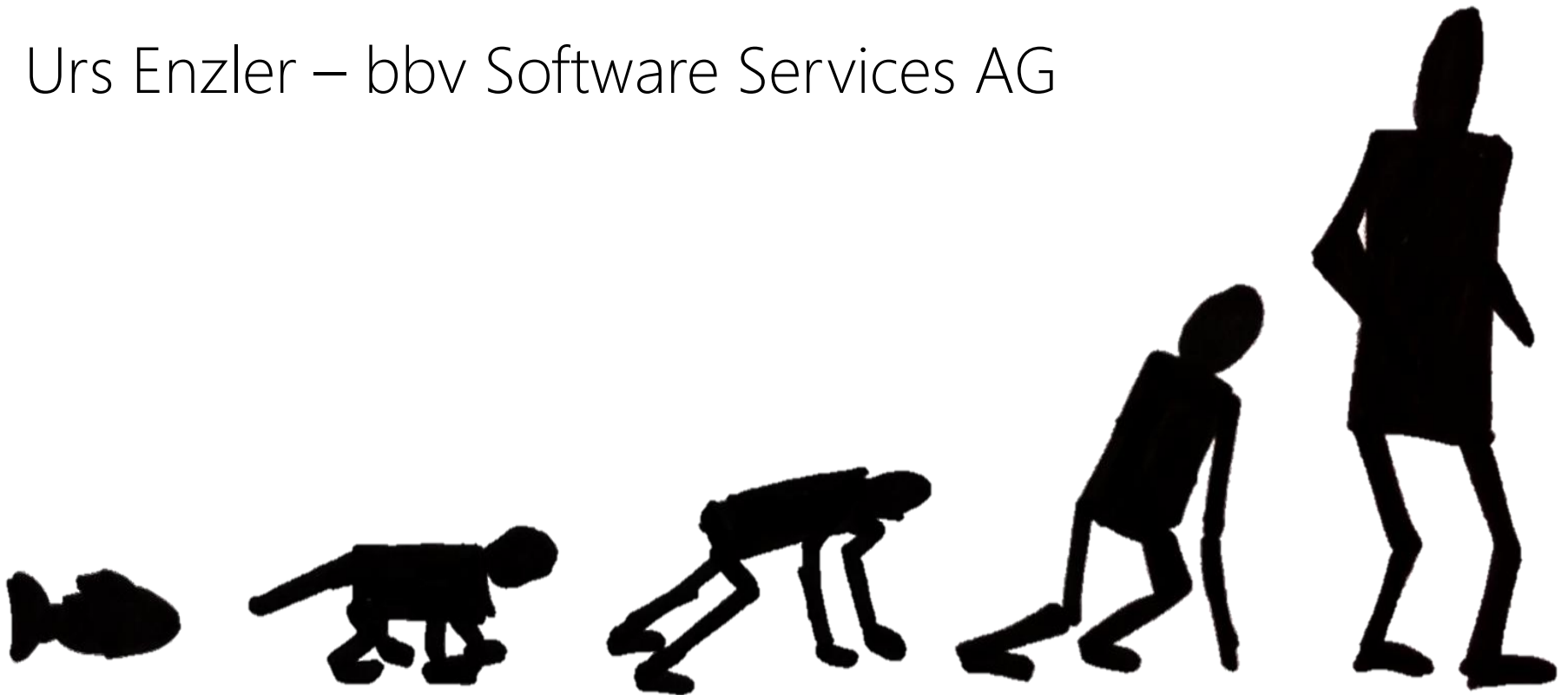


Agile Software Design

UrsENZler – bbv Software Services AG



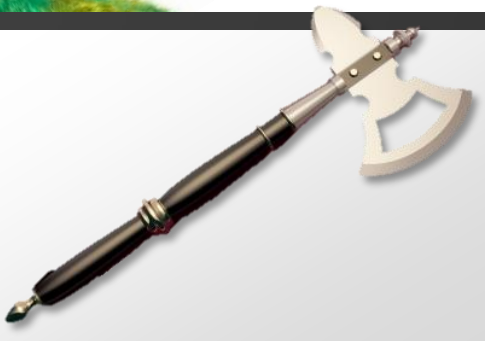




Welcome Change



Small Increments



Always Running

keep change local



Make change local



Keep change local

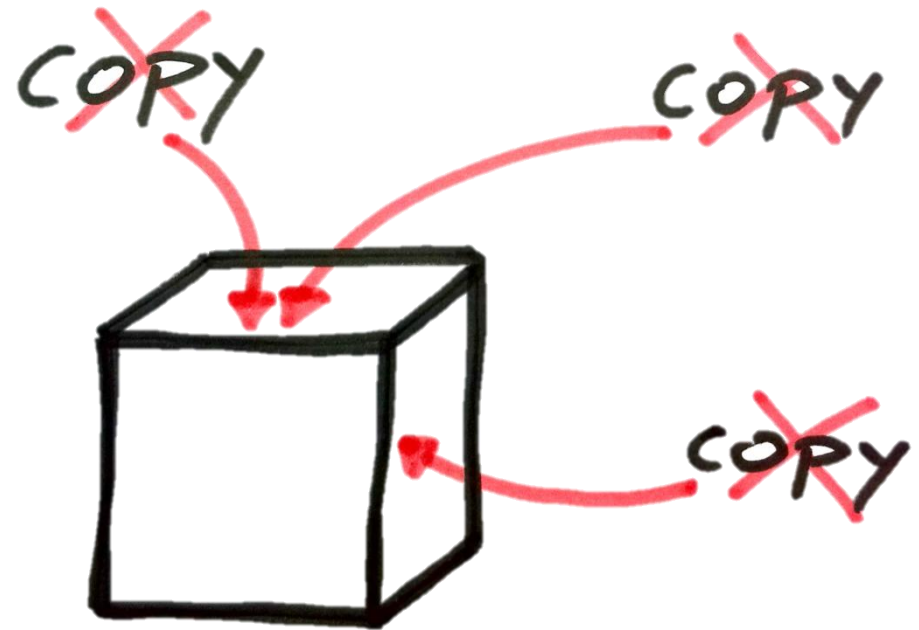


What you gain



Make Change Local

DRY – don't repeat yourself
change in a single place



```
public void DoMagic()  
{  
    this.dependency.DelegateMagic(  
        "magic",  
        42)  
  
    if (42 == i)  
    {  
        // ...  
    }  
}
```



```
if (instance.Attribute)
{
}
else
{
}
```

```
switch (instance.Attribute)
{
    case ...
}
```

Web Application

Address

Windows Application

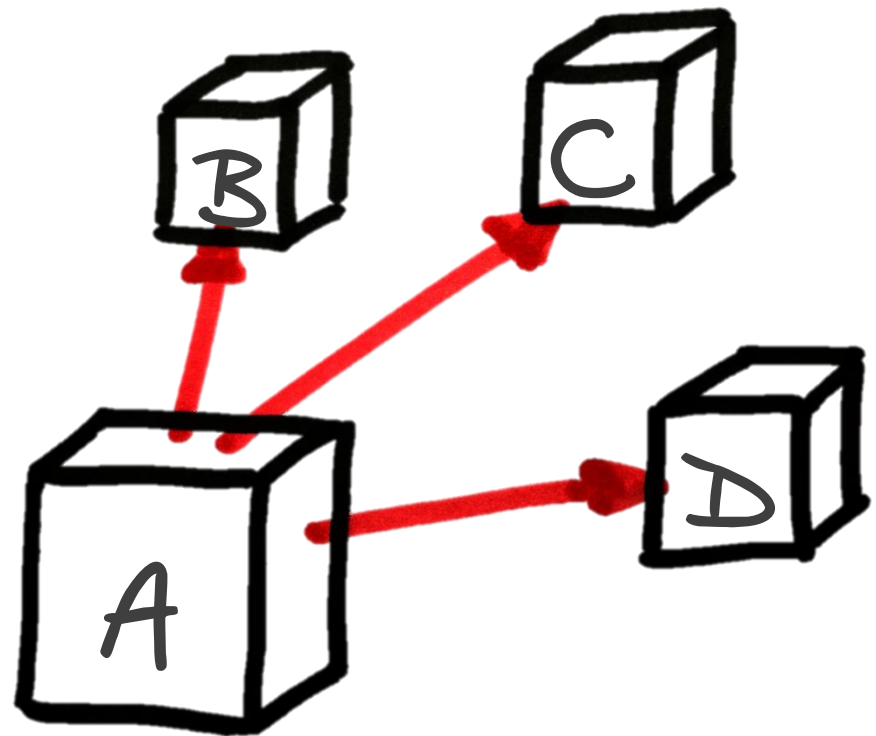
Address

Server

Address

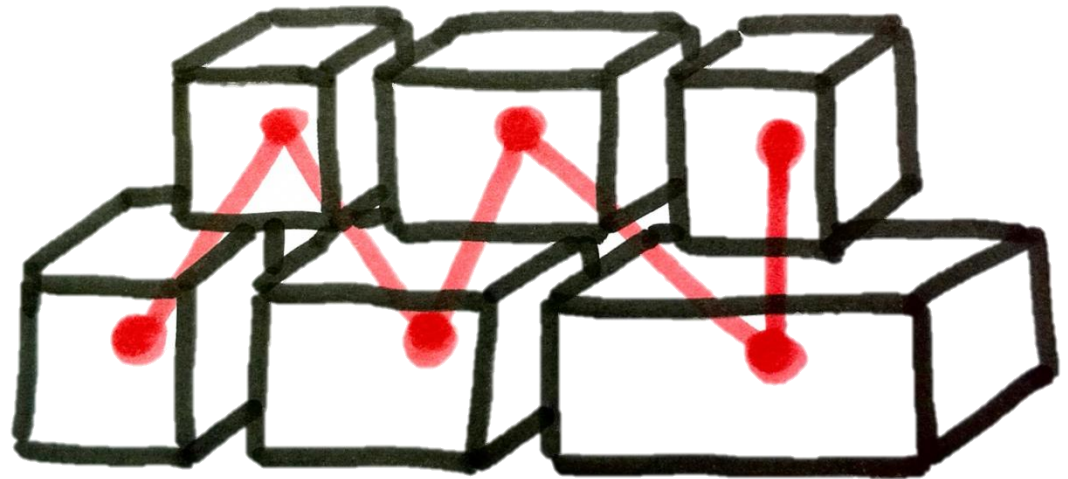
Single Responsibility

Give everything only one reason to change



Modules – make boundaries explicit

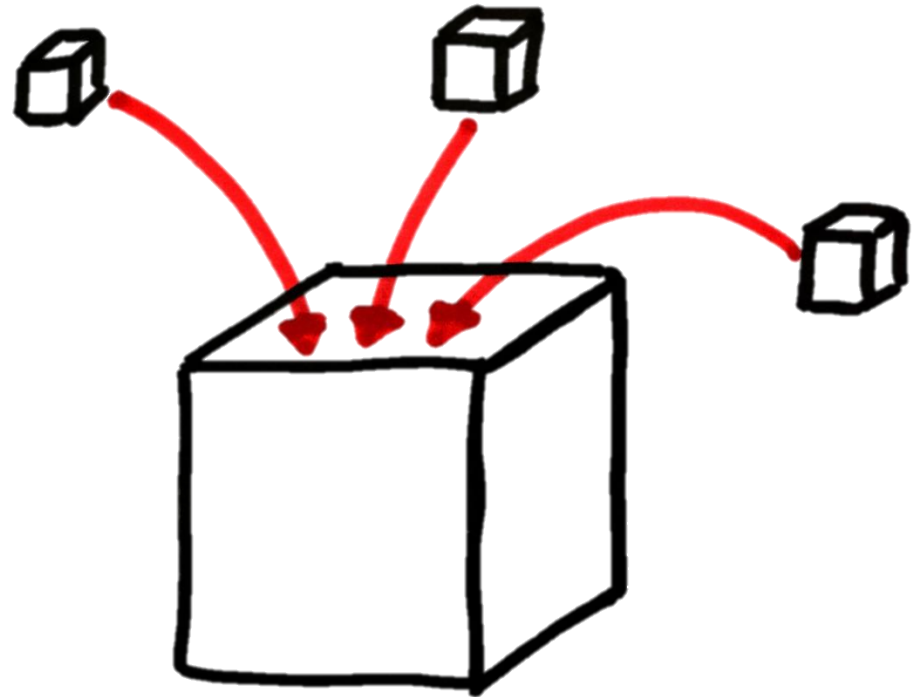
separate physically and temporally





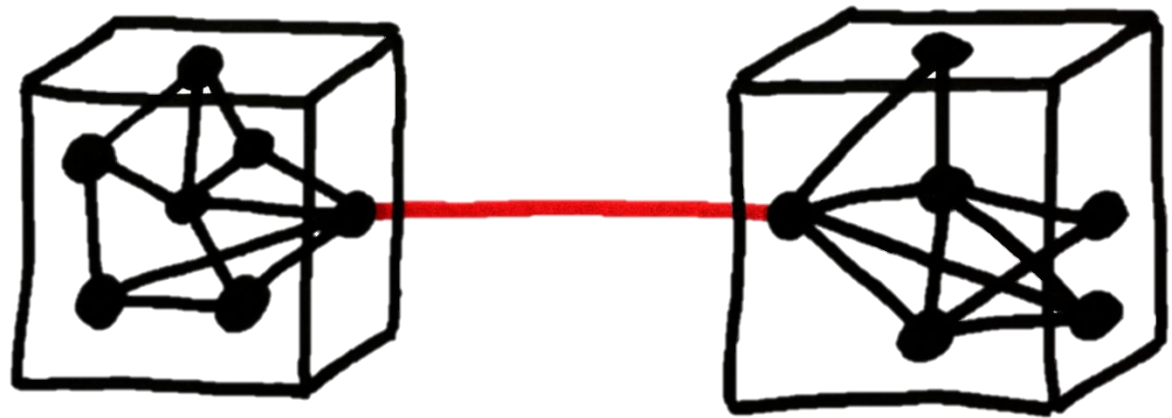
Inversion of control

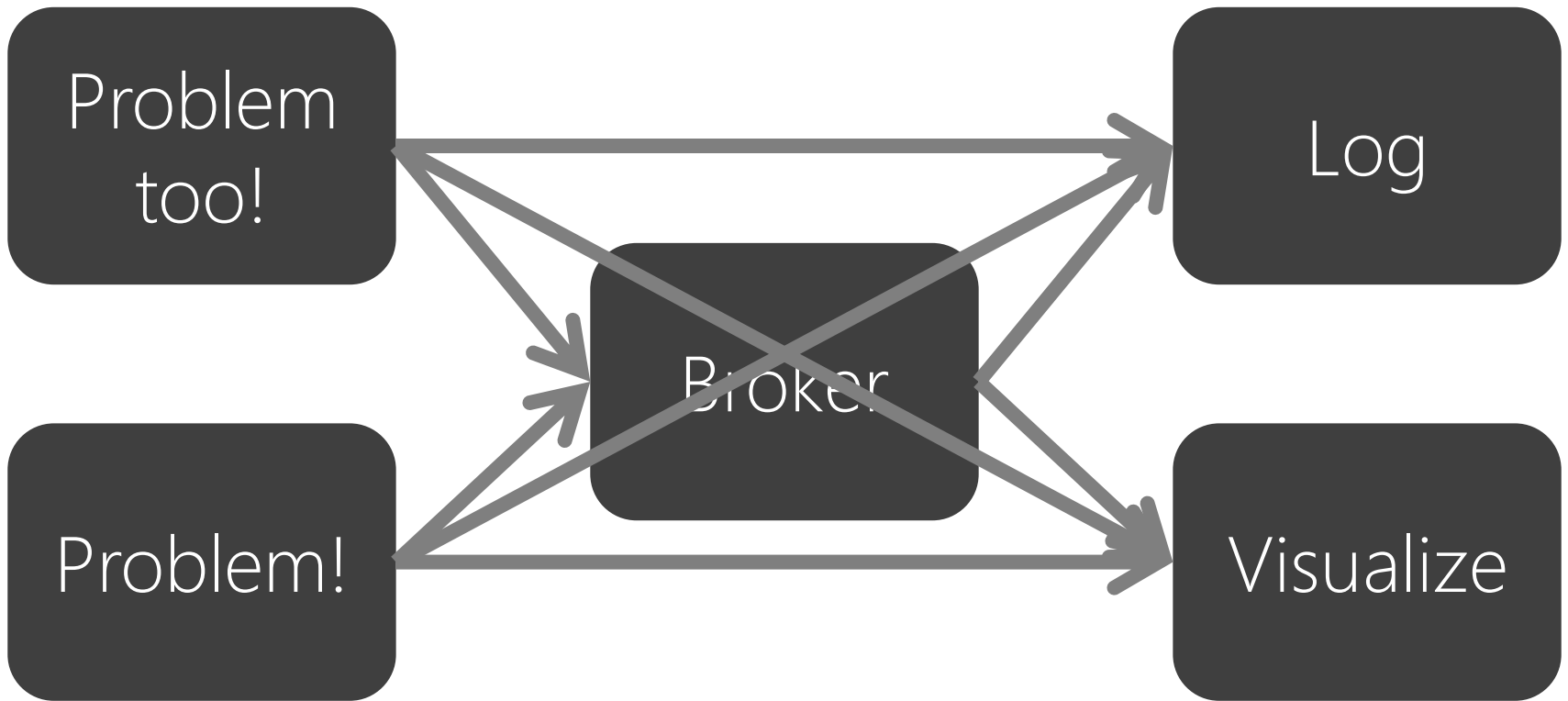
manage construction, lifetime, configuration



Loose Coupling – High Cohesion

manage needed knowledge

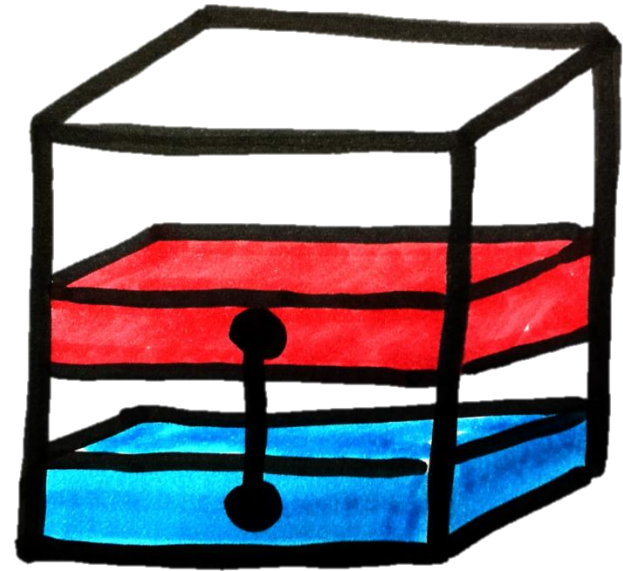




Publisher

Subscriber

Separate Cross-Cutting Concerns



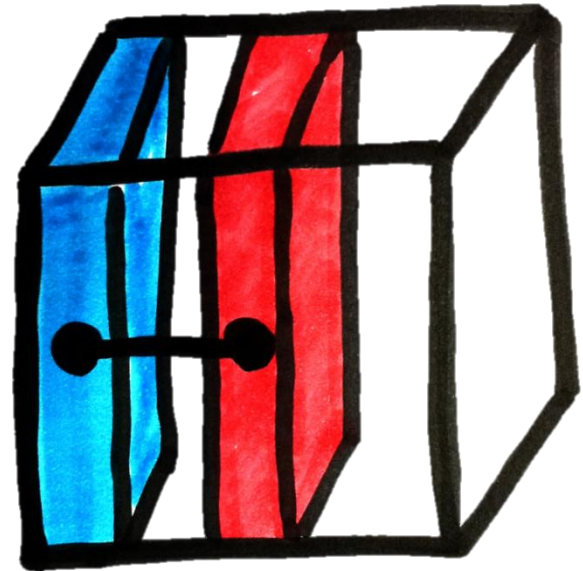
```
public void DoMagic()  
{  
    logger.Info("starting magic!");  
  
    // ...  
  
    logger.Trace("were're here.");  
  
    // ...  
  
    logger.Debug("magic achieved");  
}
```

```
public void DoMagic()  
{  
    logger.LogStartOfMagic();  
}
```

```
public static class LogExtensions  
{  
    public static void LogStartOfMagic(  
        this ILogger logger)  
    {  
        logger.Info("starting magic")  
    }  
}
```

Namespaces

Arrange Code per Feature



Feature

Client



Logic

Communication

Server



Façade

Business Logic

Data Access

Namespace per Feature

- MyCompany
 - MyProject
 - Feature1
 - Feature2
 - Grouping1
 - Grouping2
 - Feature3
 - Feature4
 - Grouping1
 - DetailStuff
 - Grouping2

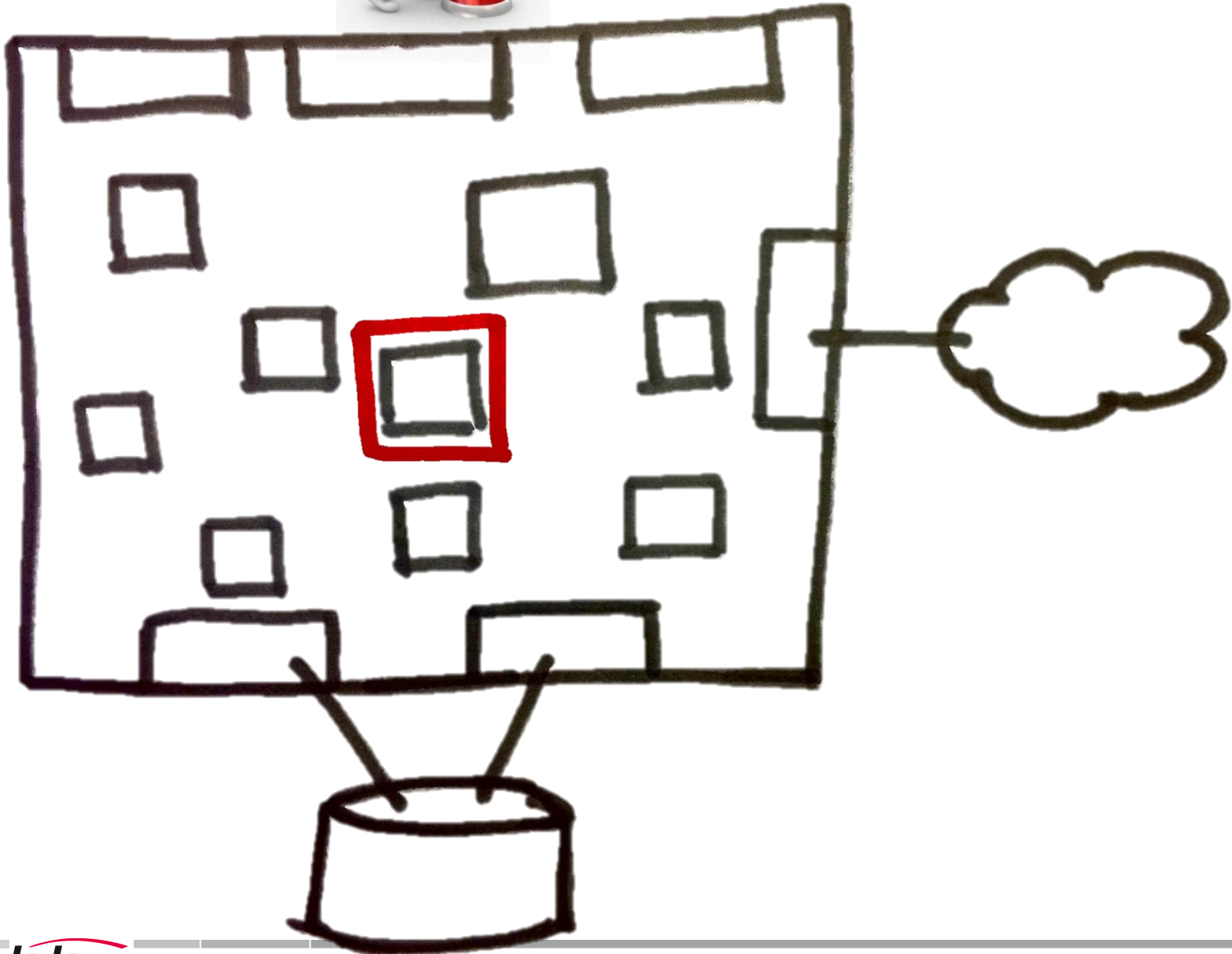
Keep Change Local



Unit Tests

class boundary

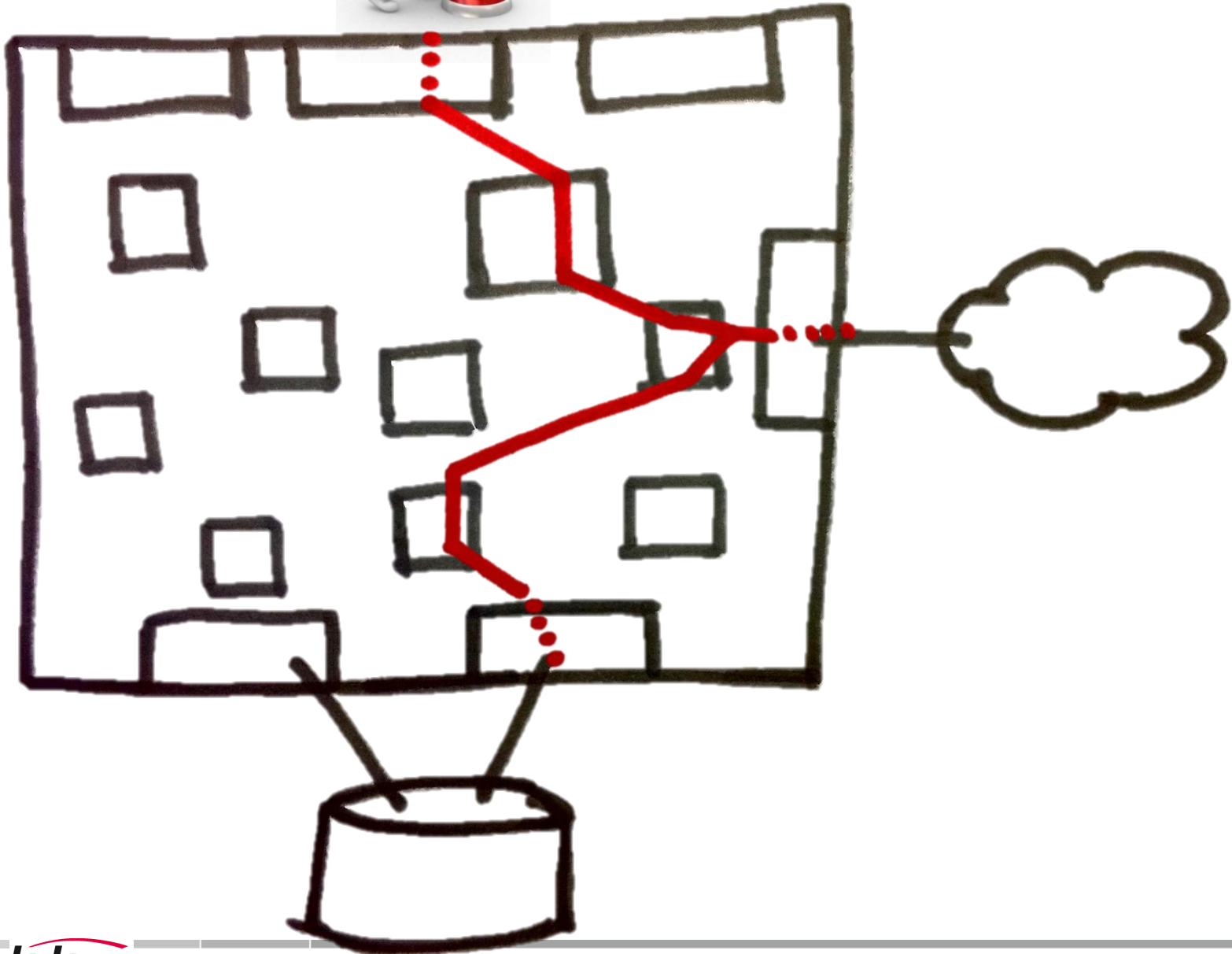




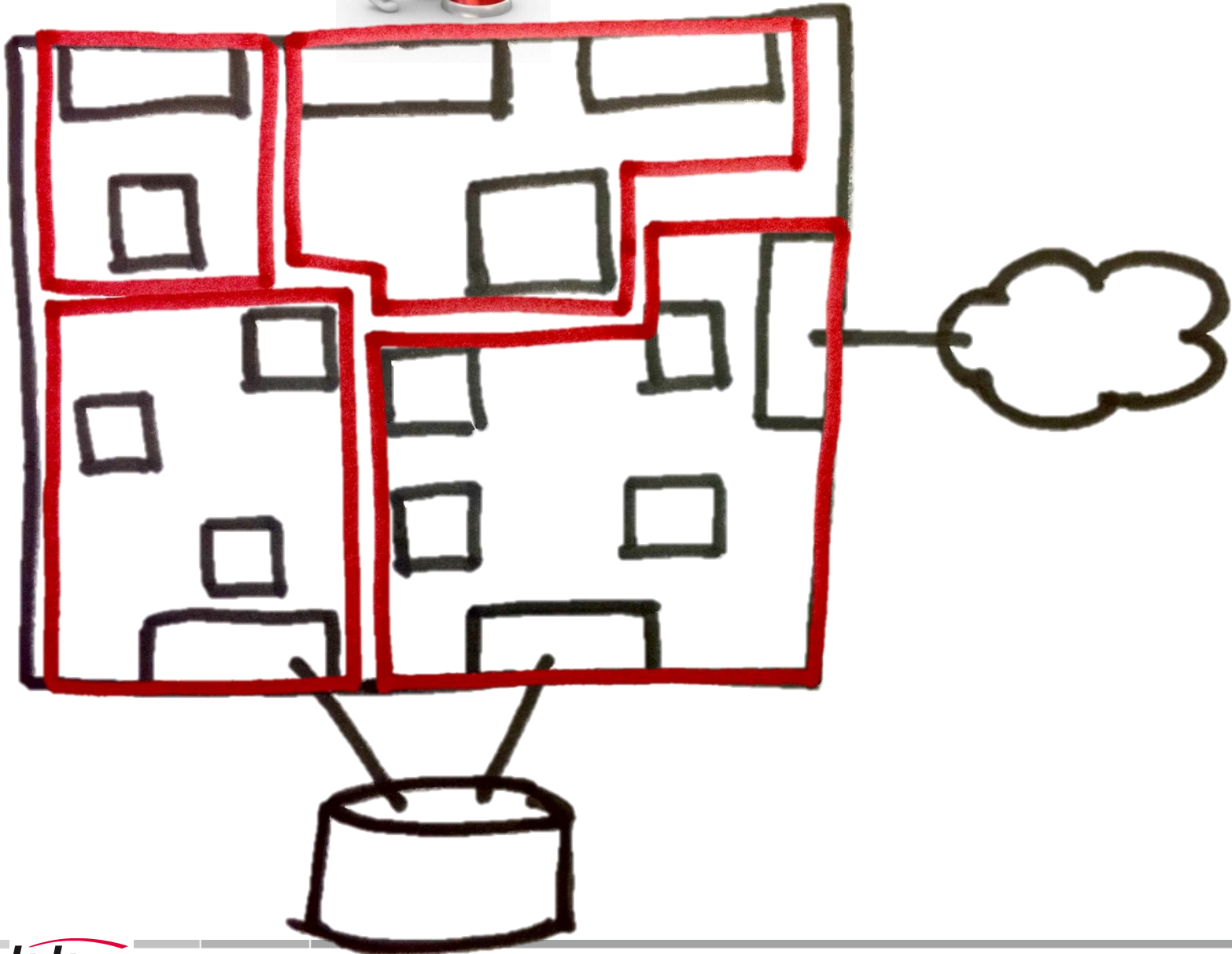
Acceptance Tests

feature boundary





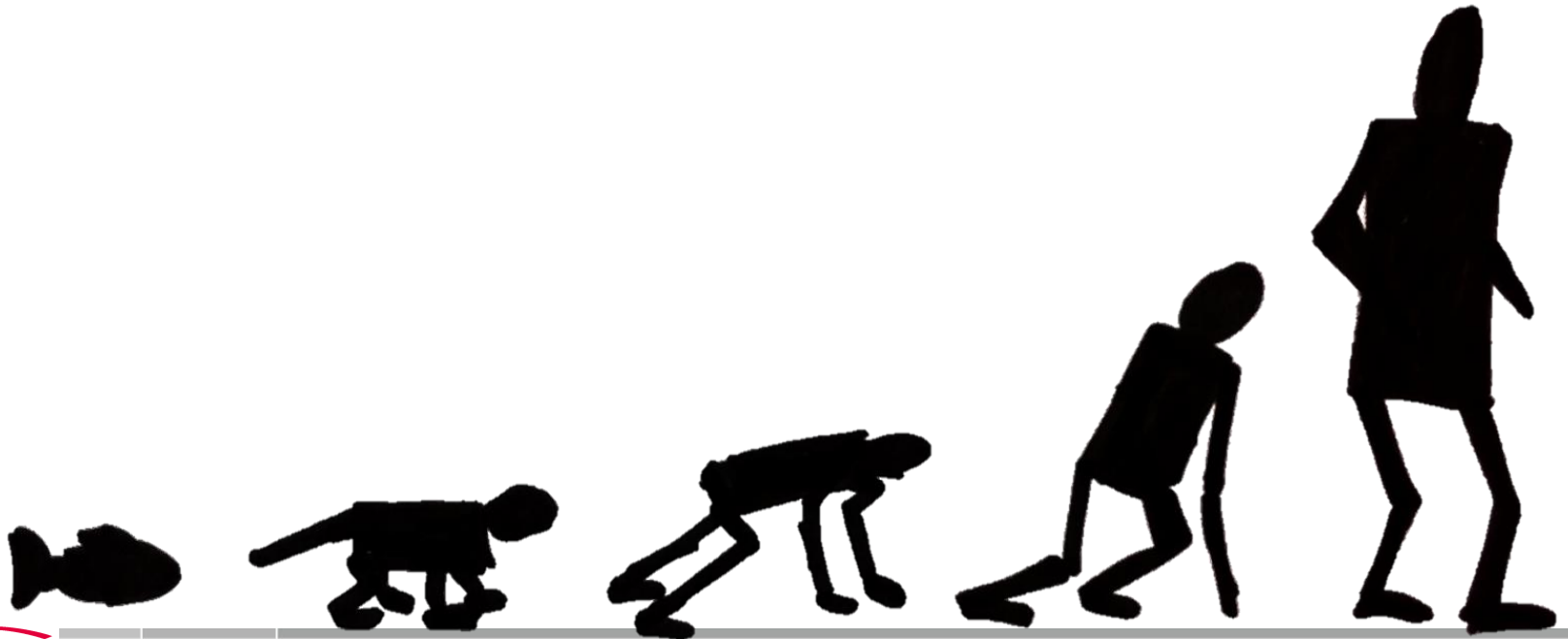
big system = \sum *small system*





What You Gain

evolving architecture



always validated by
working code



ask you this:



how easily can it
be removed ?



Urs Enzler
urs.enzler@bbv.ch

OSS lead:
blog:

Appccelerate
www.planetgeek.ch
www.bbv.ch/blog

twitter:

@ursenzler

user group: www.dotnet-zentral.ch

