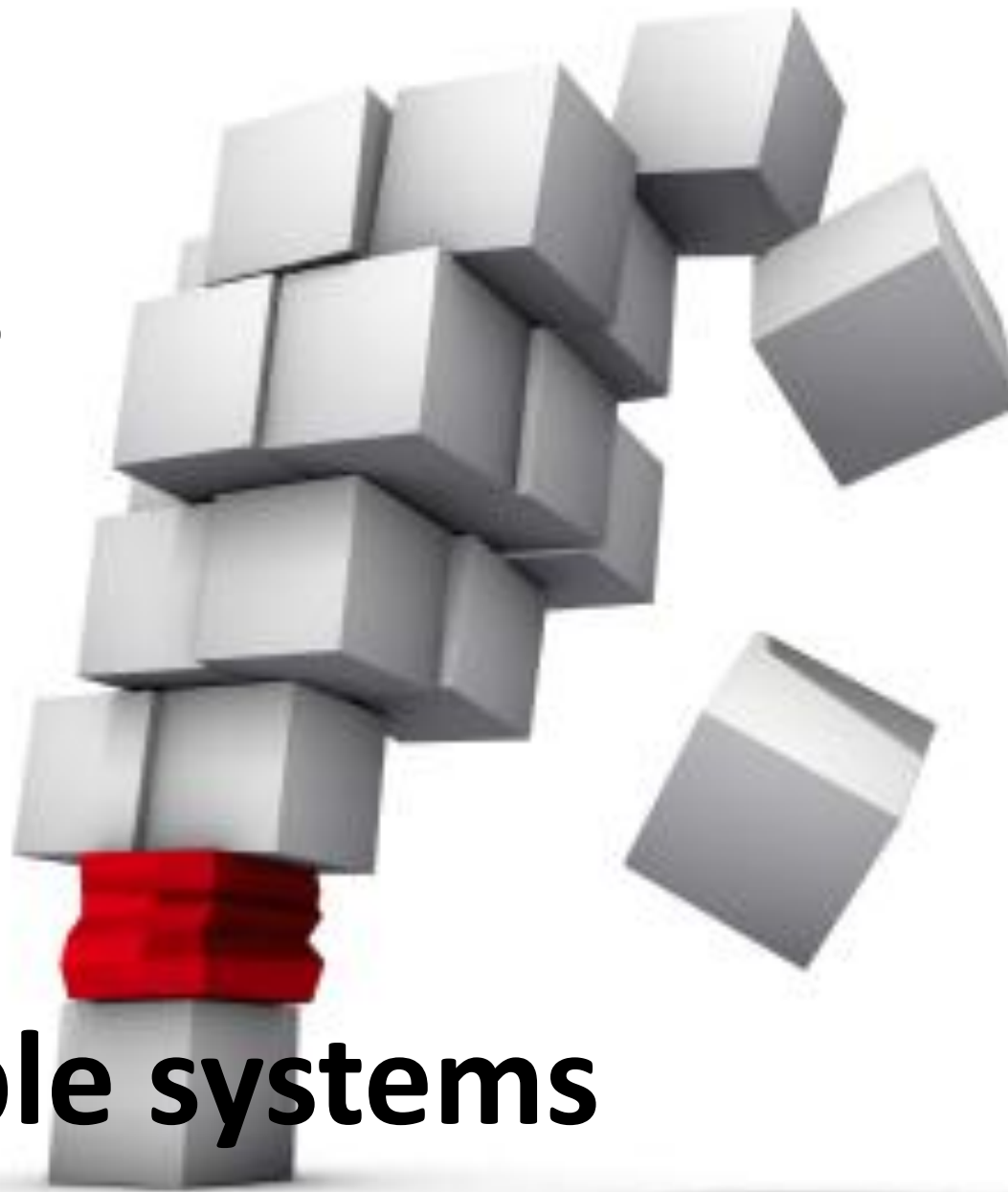
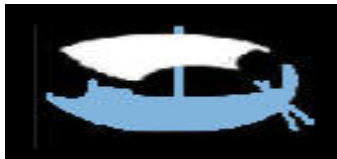
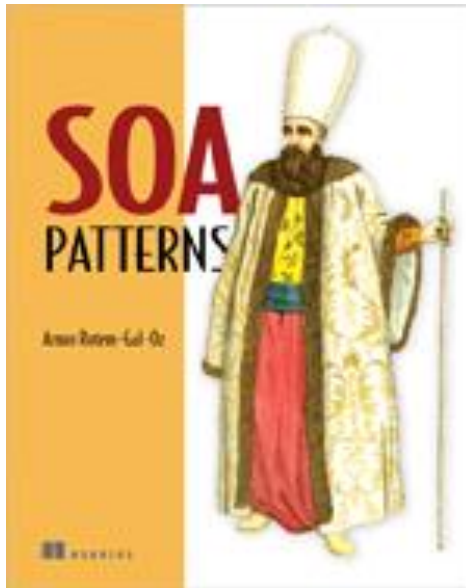


**Arnon
Rotem-Gal-Oz**



**Building reliable systems
from unreliable components**

Arnon Rotem-Gal-Oz



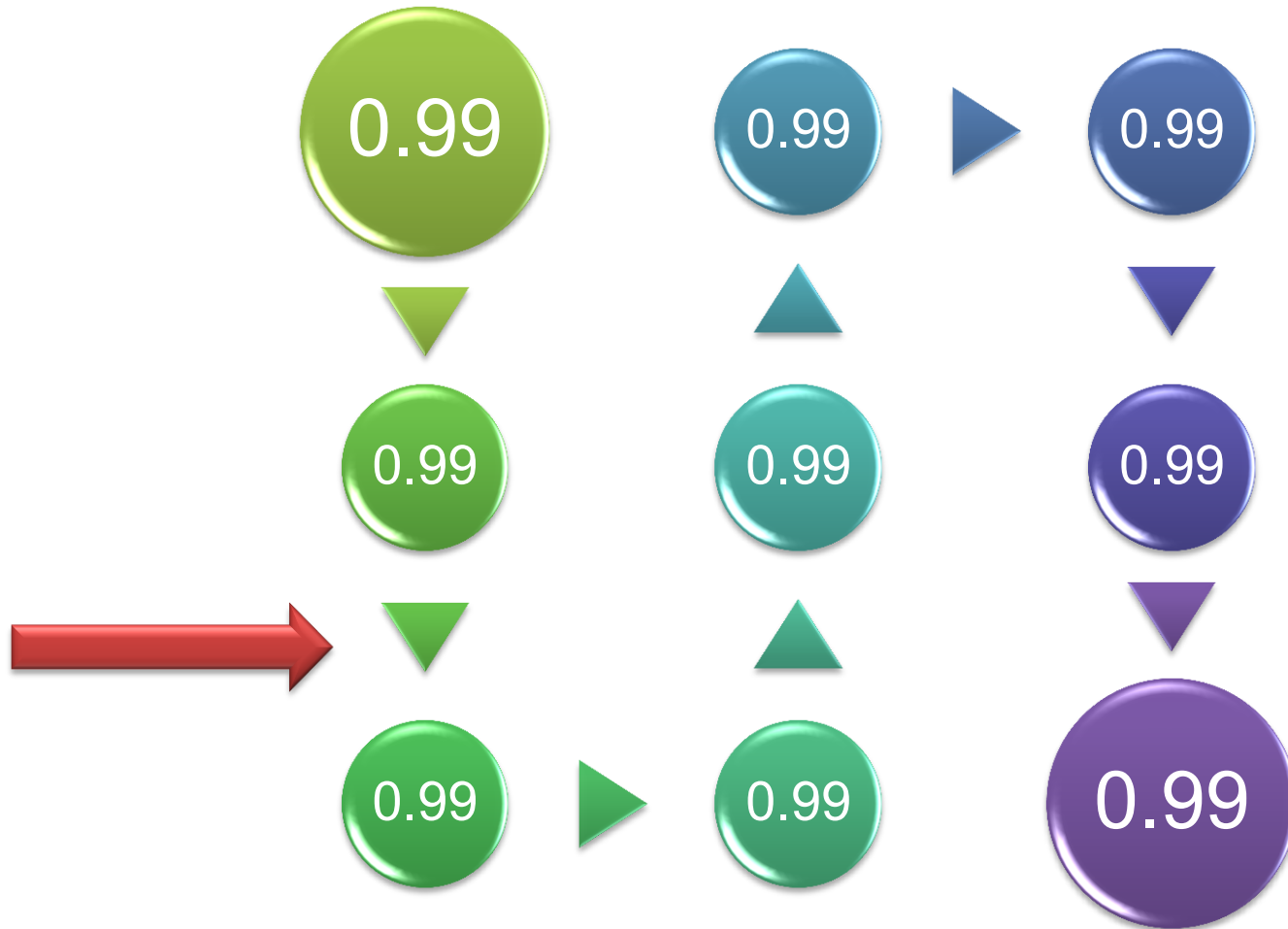


What's in a 9

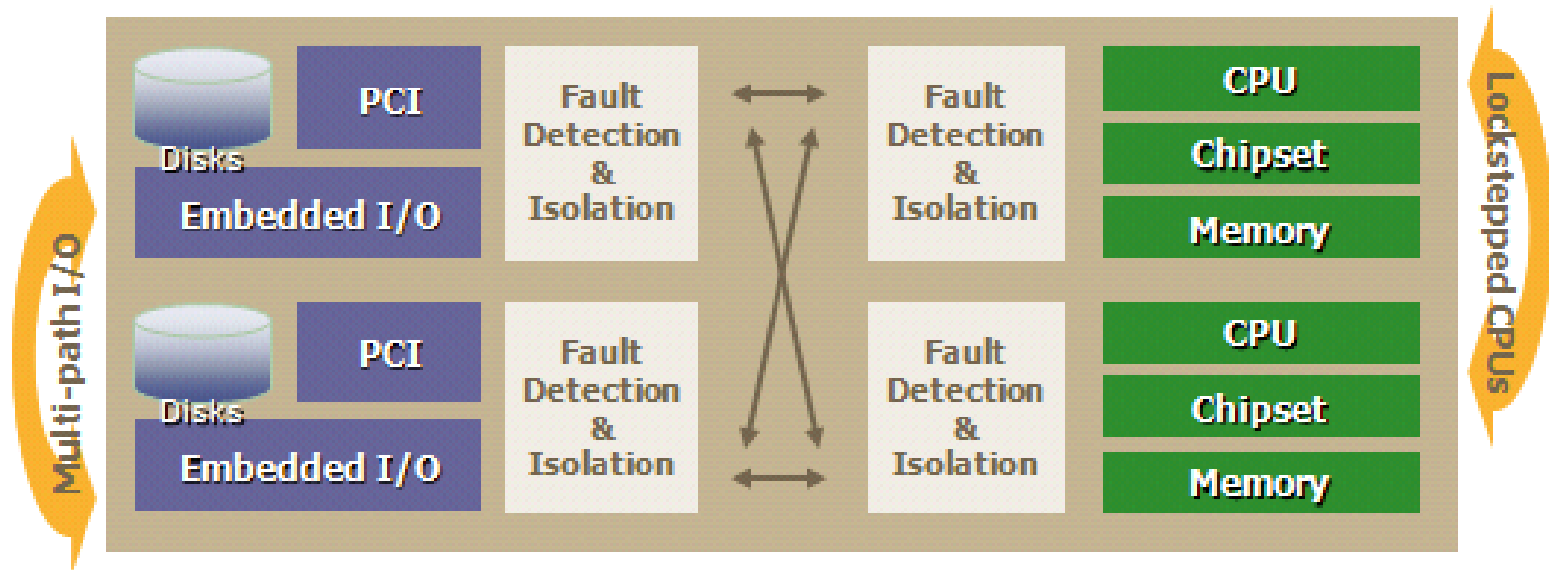


We have a nice little legacy
business component

And we move it to SOA



Failsafe hardware



Status Technologies
FT Server

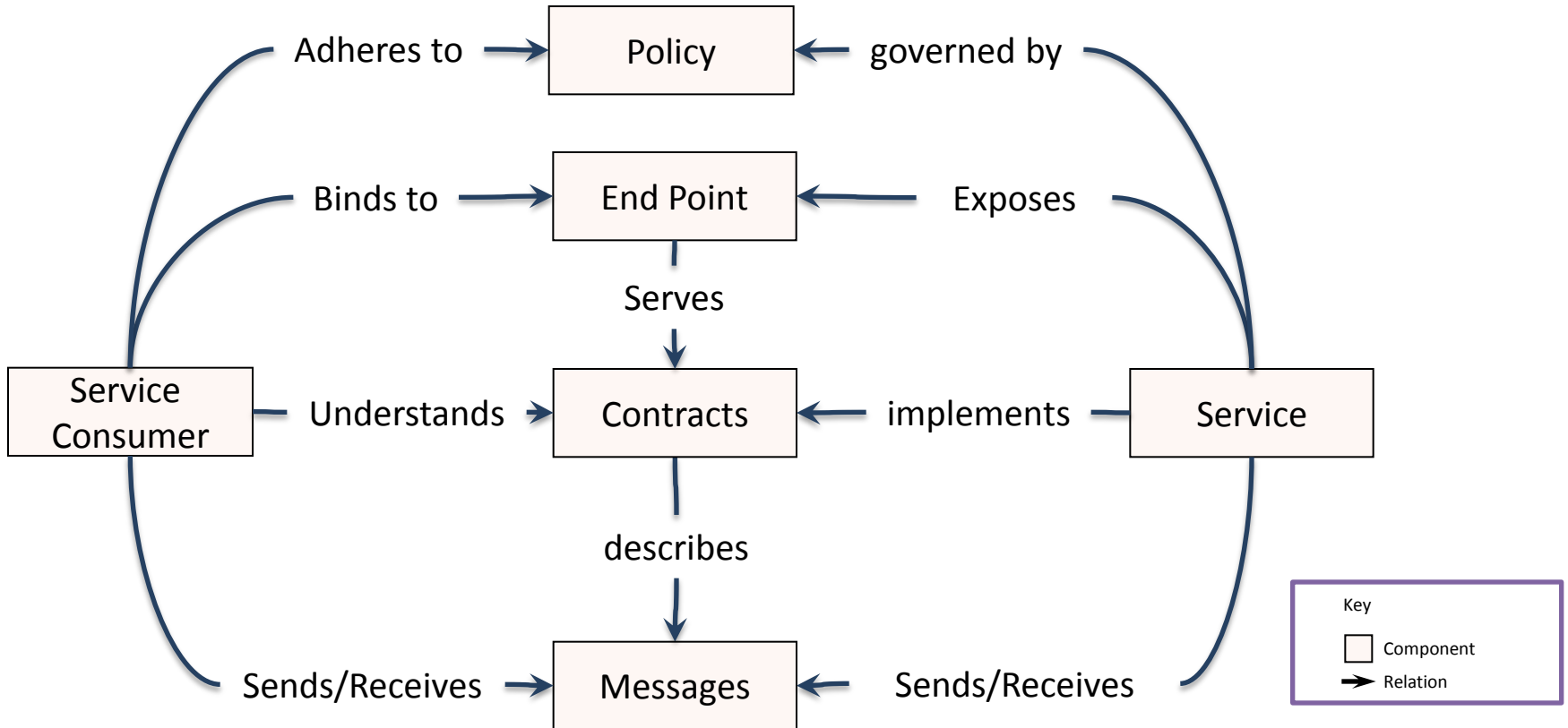
Or try to detect failure , handle it and minimize its effect on the business service

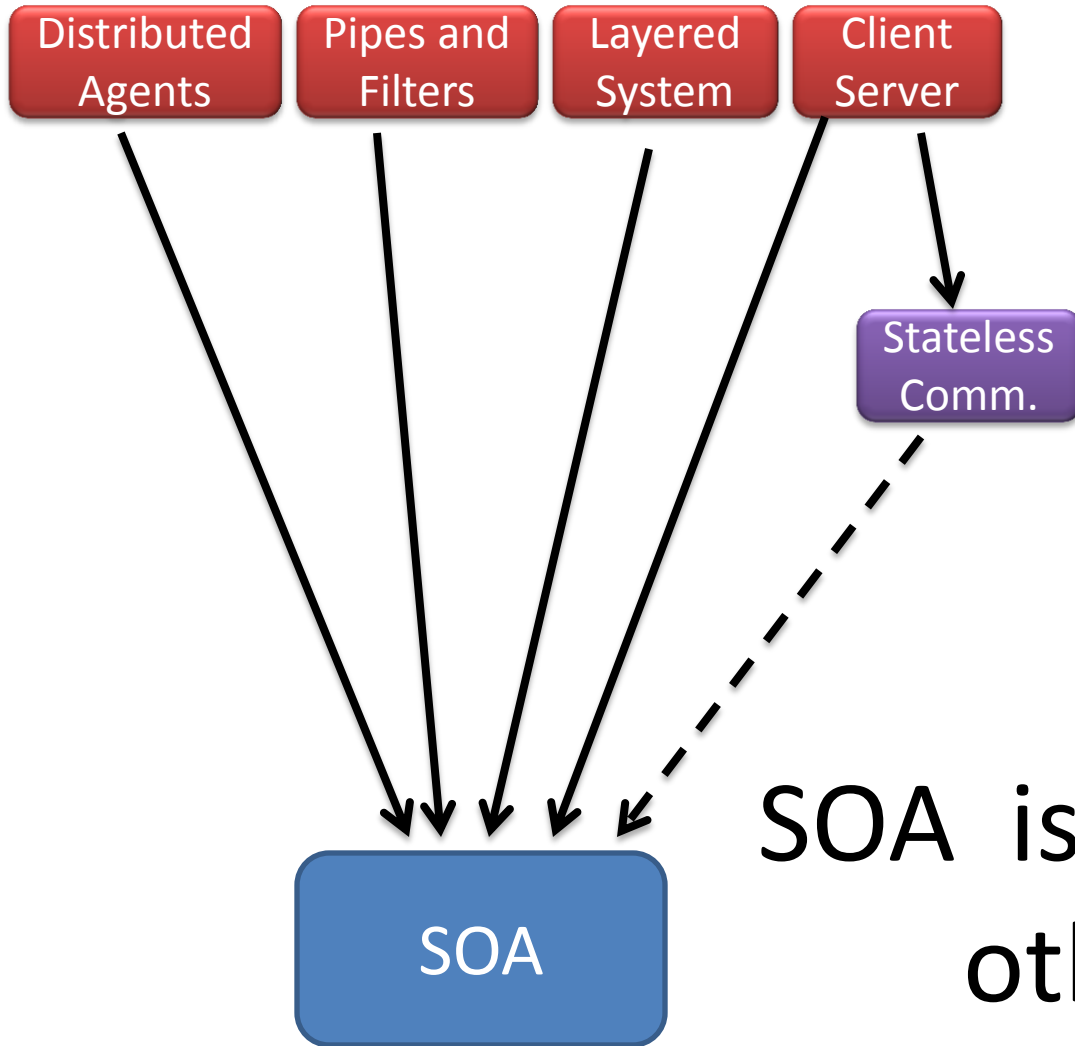




BEFORE WE BEGIN

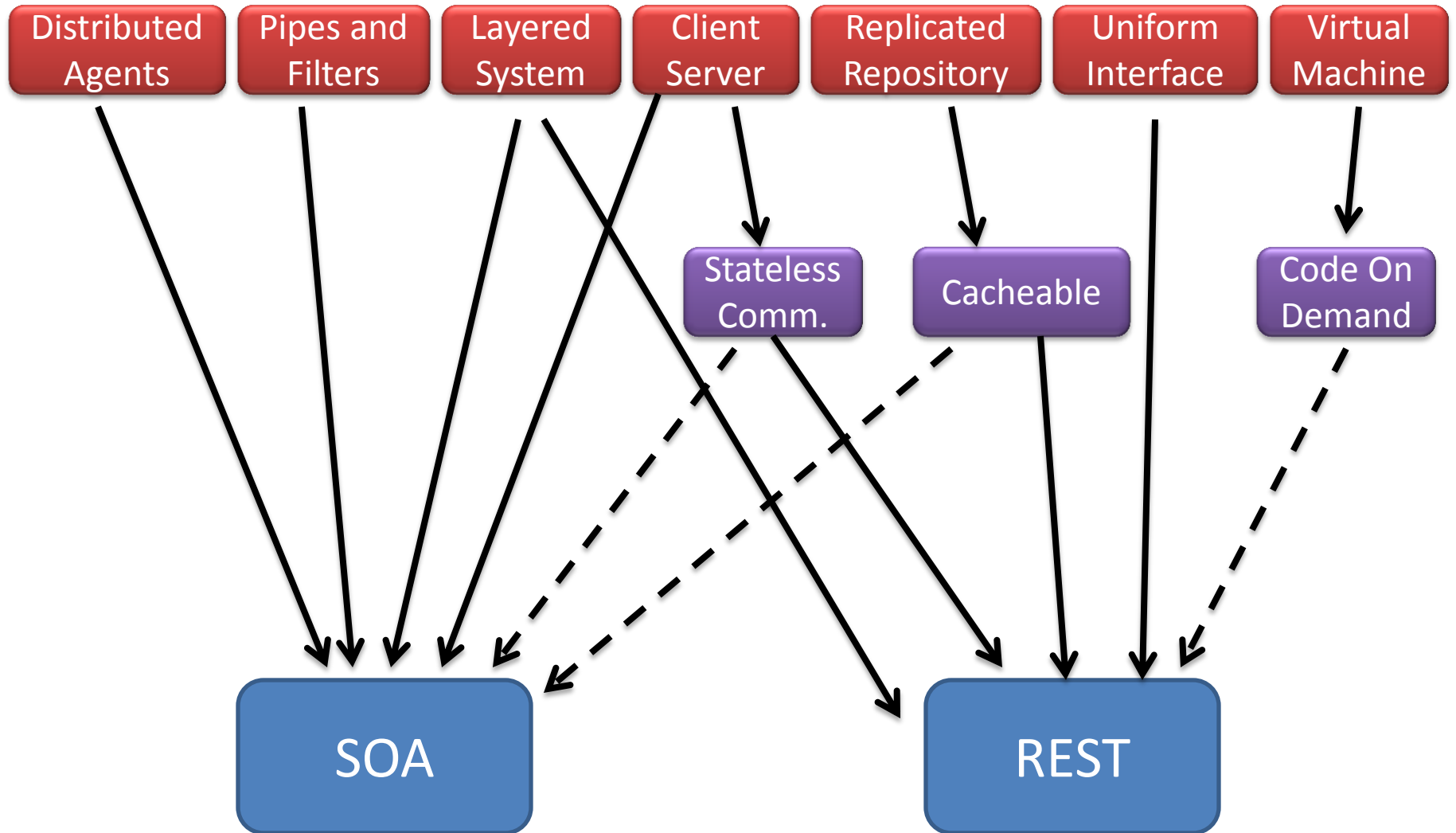
SOA





SOA is derived from other styles

SOA vs. REST





1.
Point



2.
Click



3.
Buy



4.
Enjoy!

System

Monitoring

Usage
Datmart

Billing

Mobile Integration

3G Video
Calls

MMS

Dedicated
Client

3rd parties

Applications

Acquisition

Interactions

branding

Ad Management

Targeted
Advertizing

Campaign
Mgmt.

Reporting

Data mining
& Statistics

Reports

Resources

Interactions

Links

Reference
Data

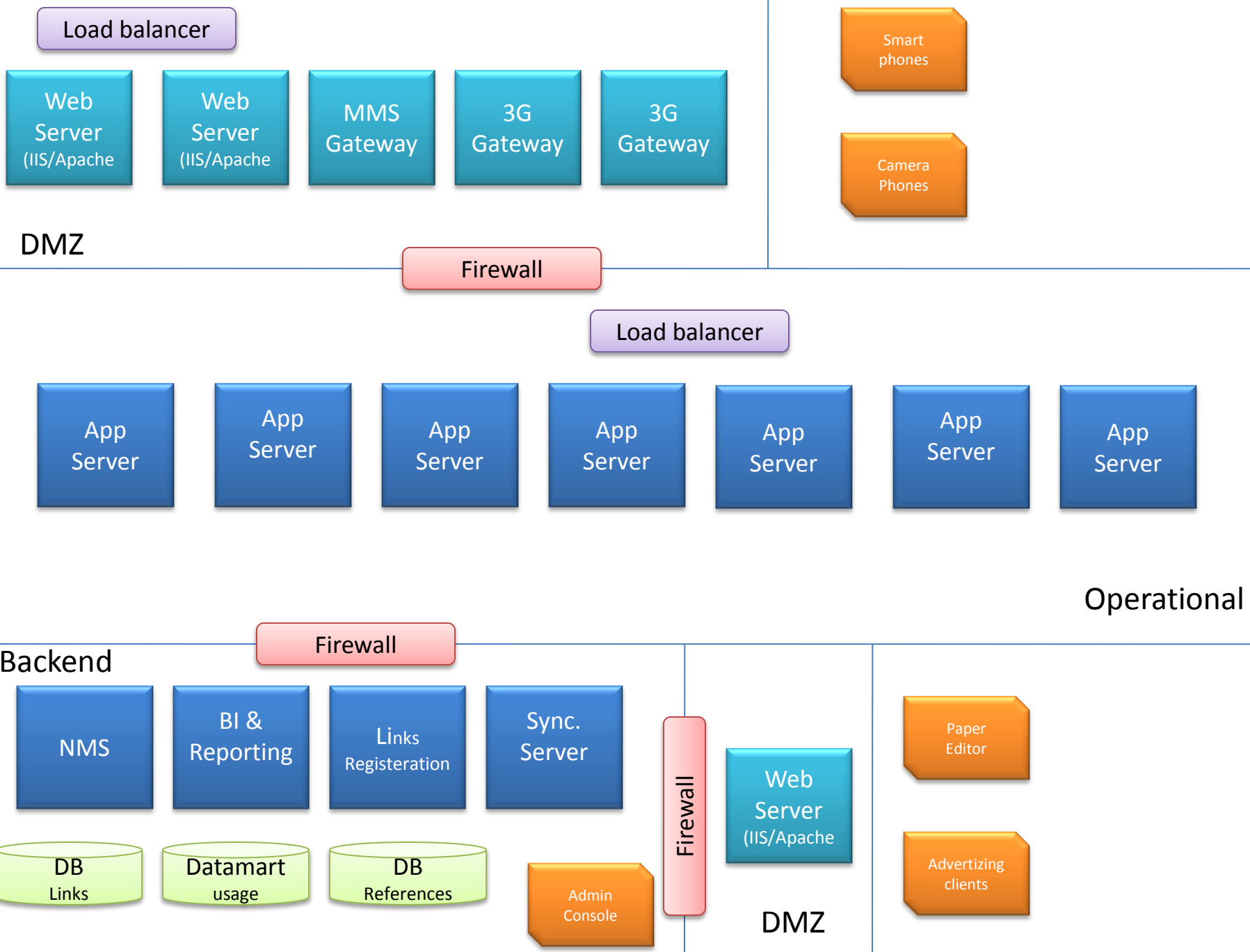
Link Management

Data
Interfaces

Web
Front-end

Publishing
tools
integration

Interaction
Designer

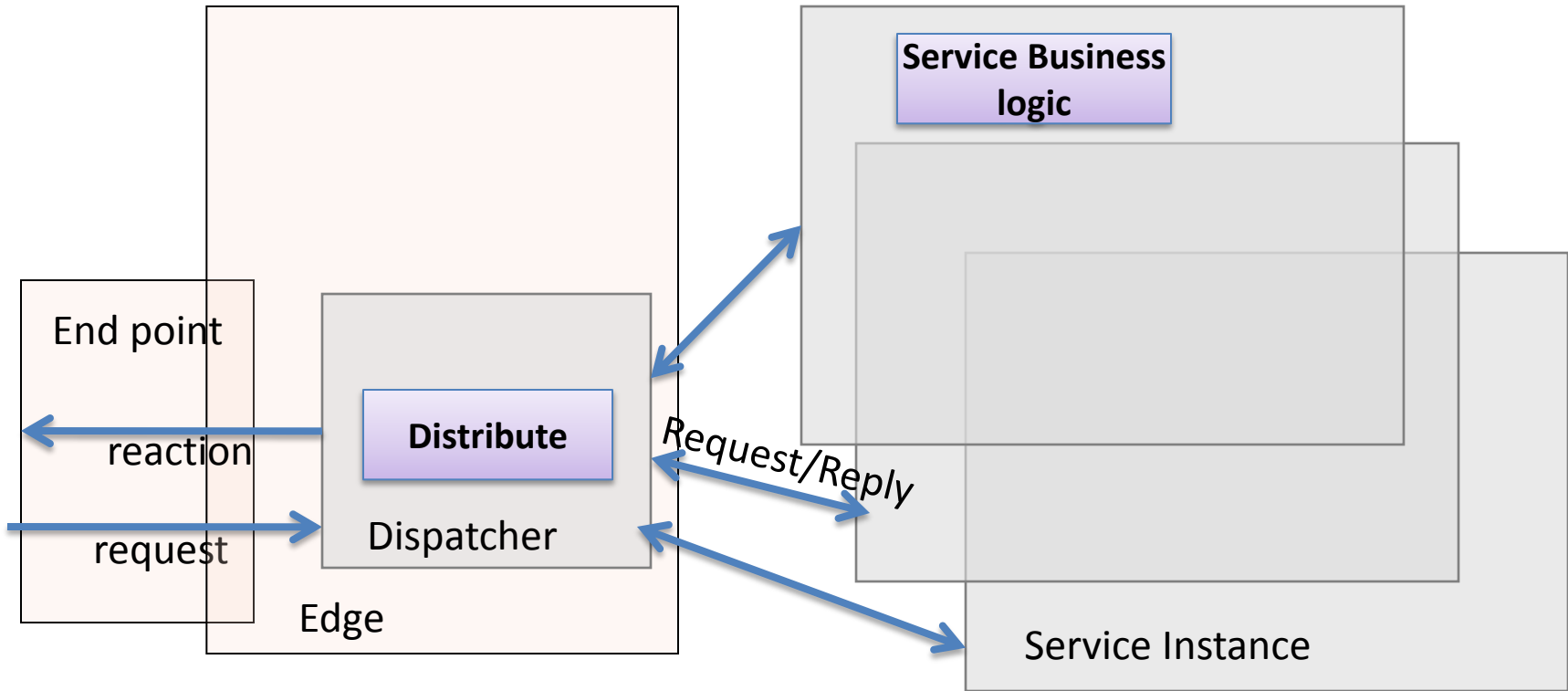


**CHALLENGE – SERVICE
AVAILABILITY**

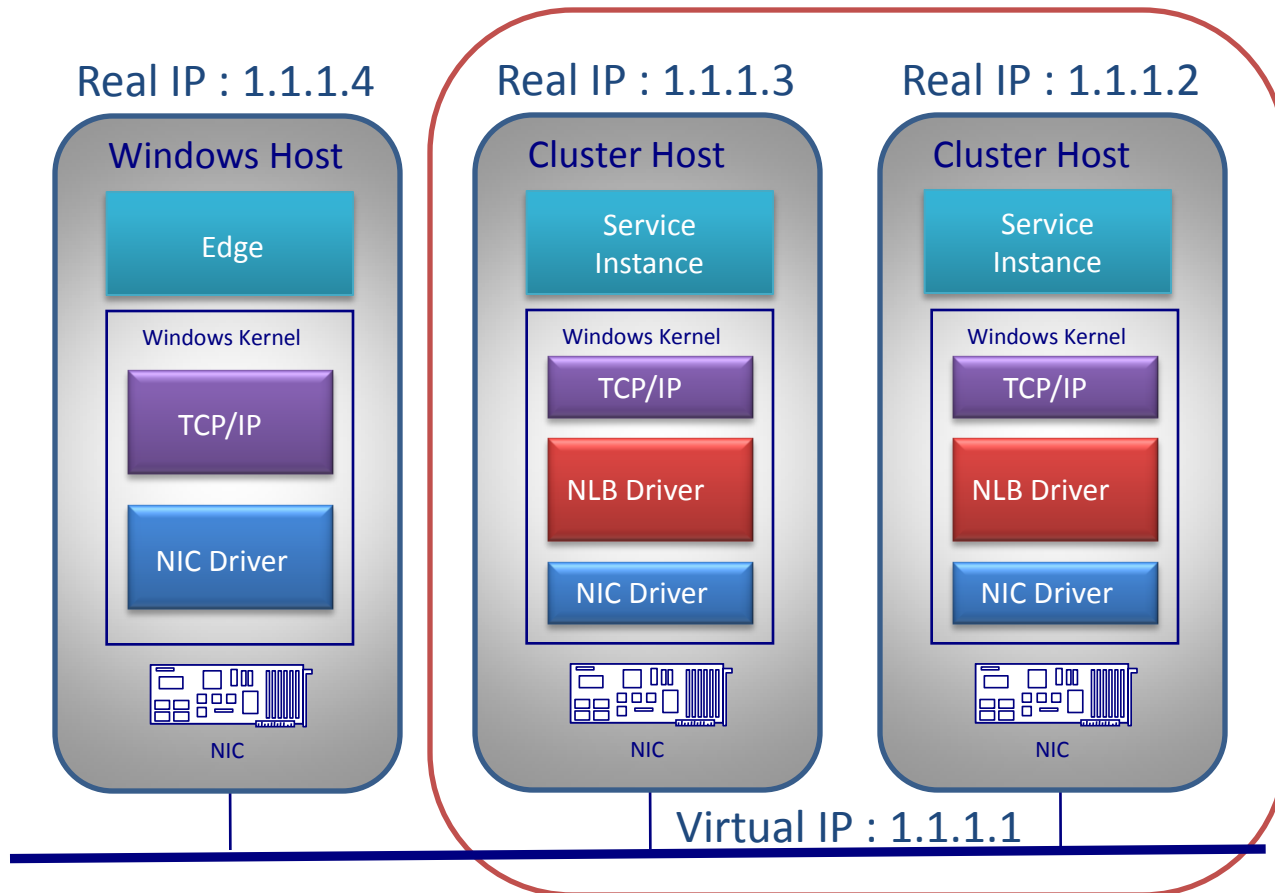
What's the effect of a failure - Server



Service Instance



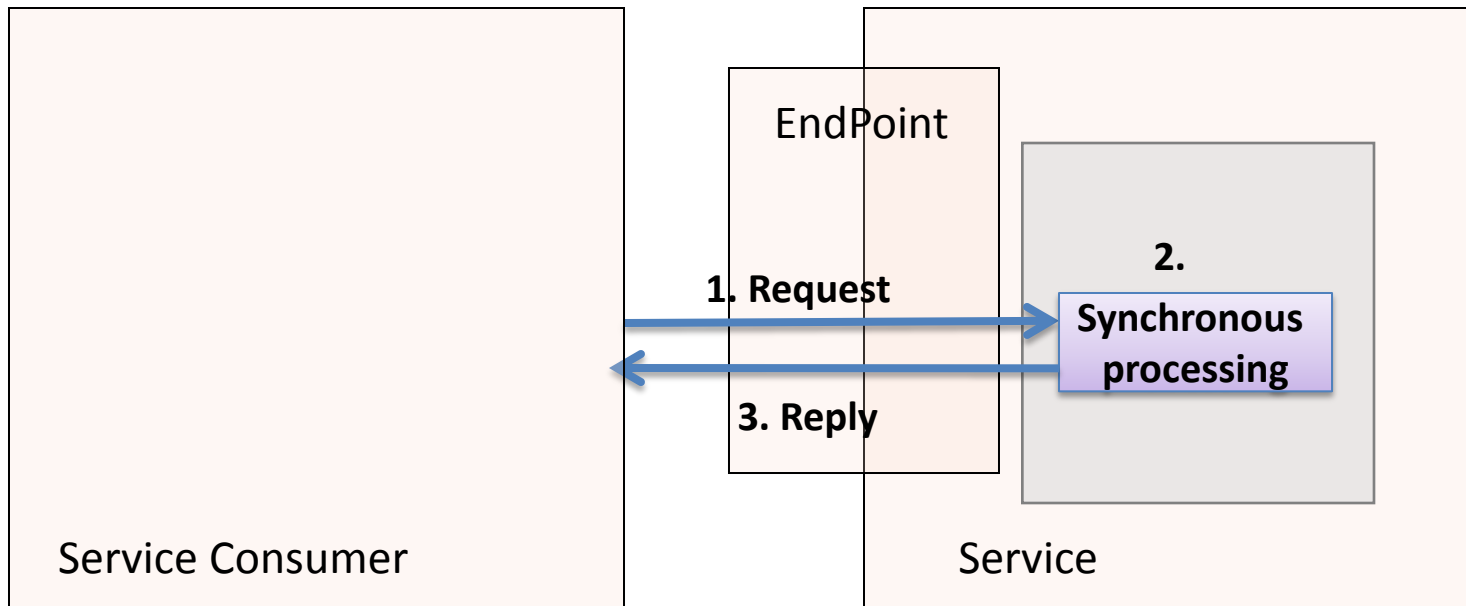
Service Instance with NLB



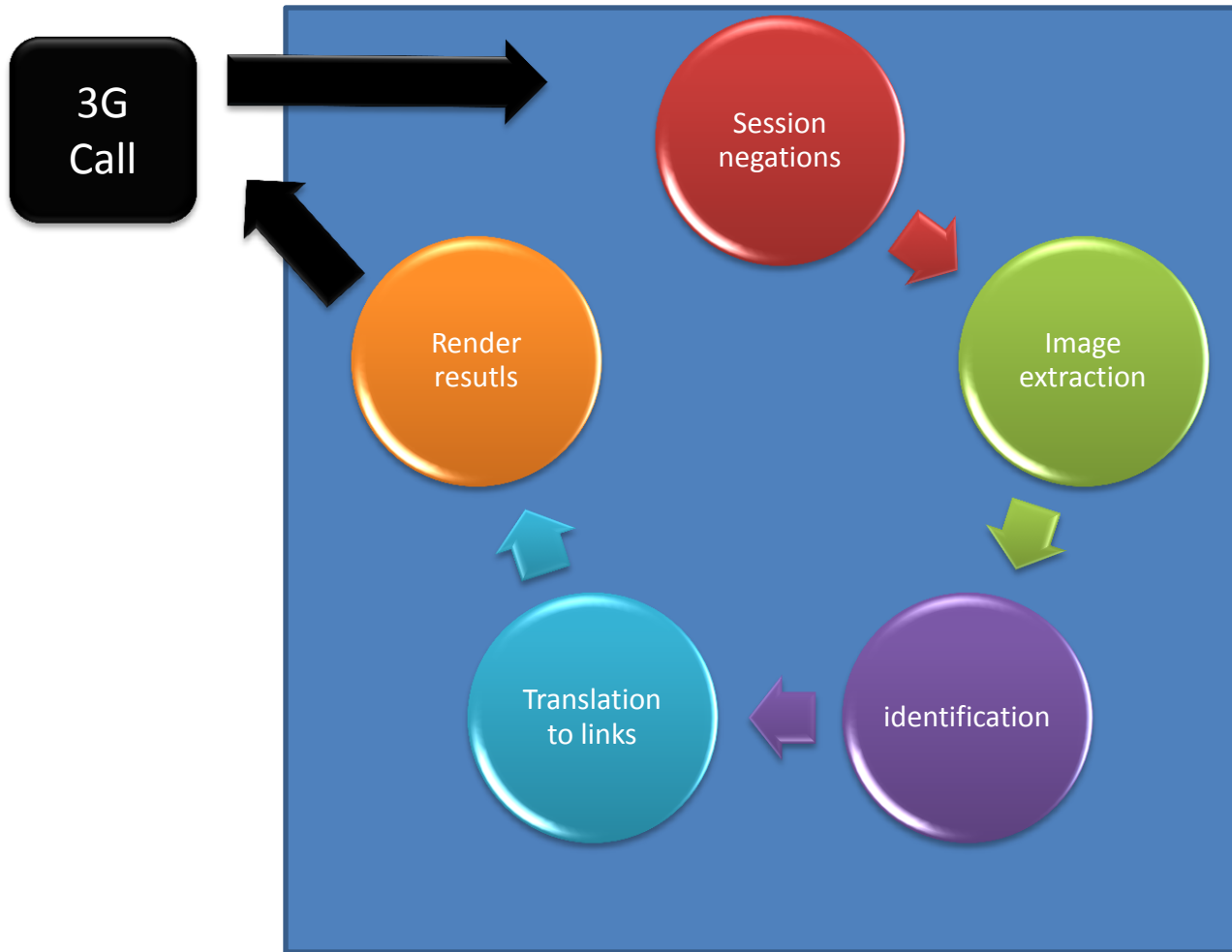


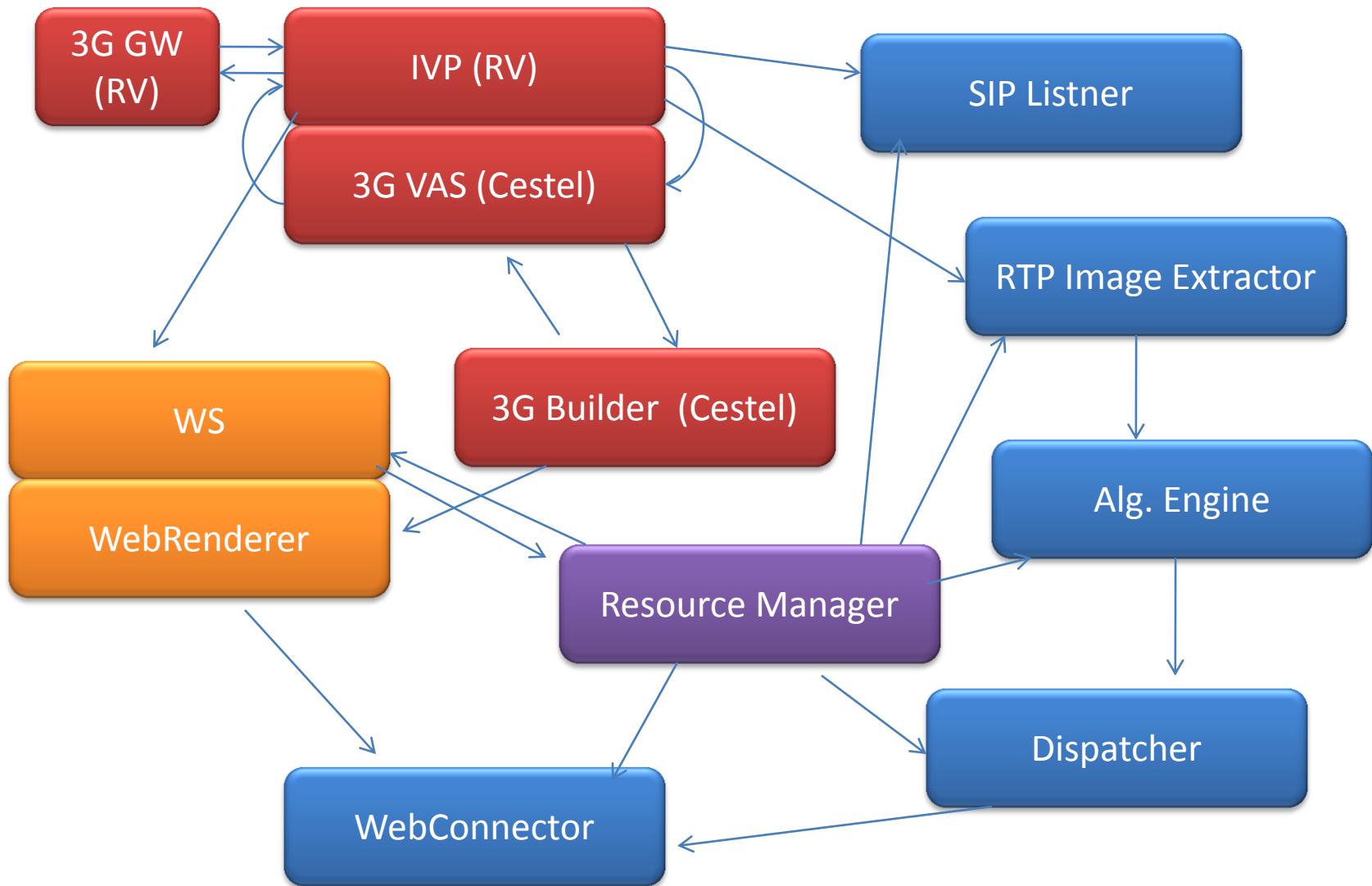
Virtual Endpoint

Request/Reply



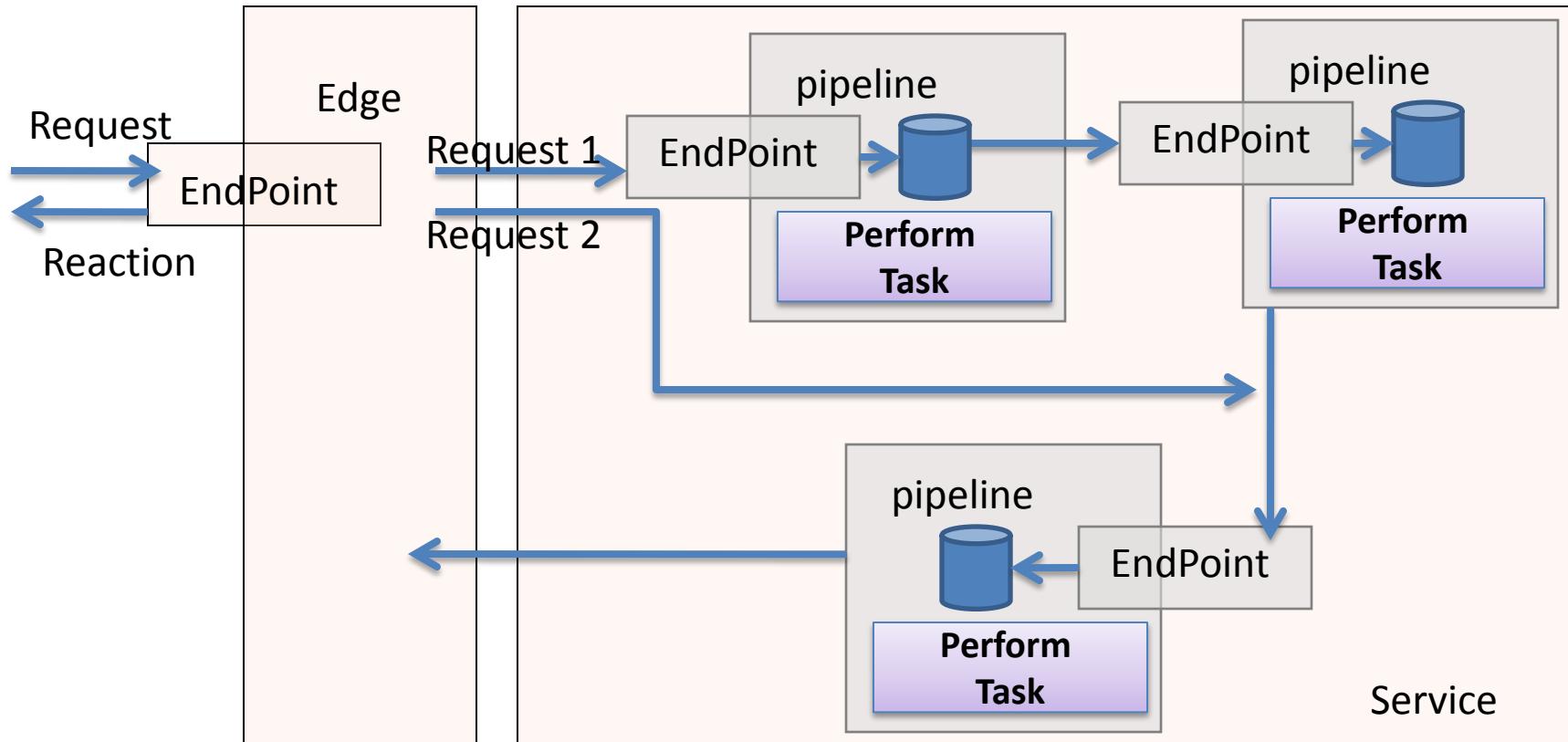
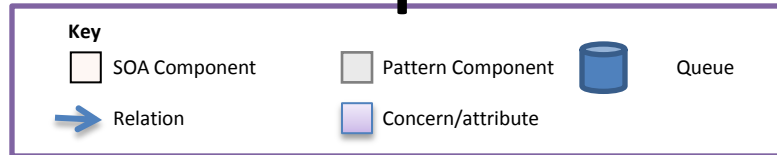
Things look Cool & Simple TM



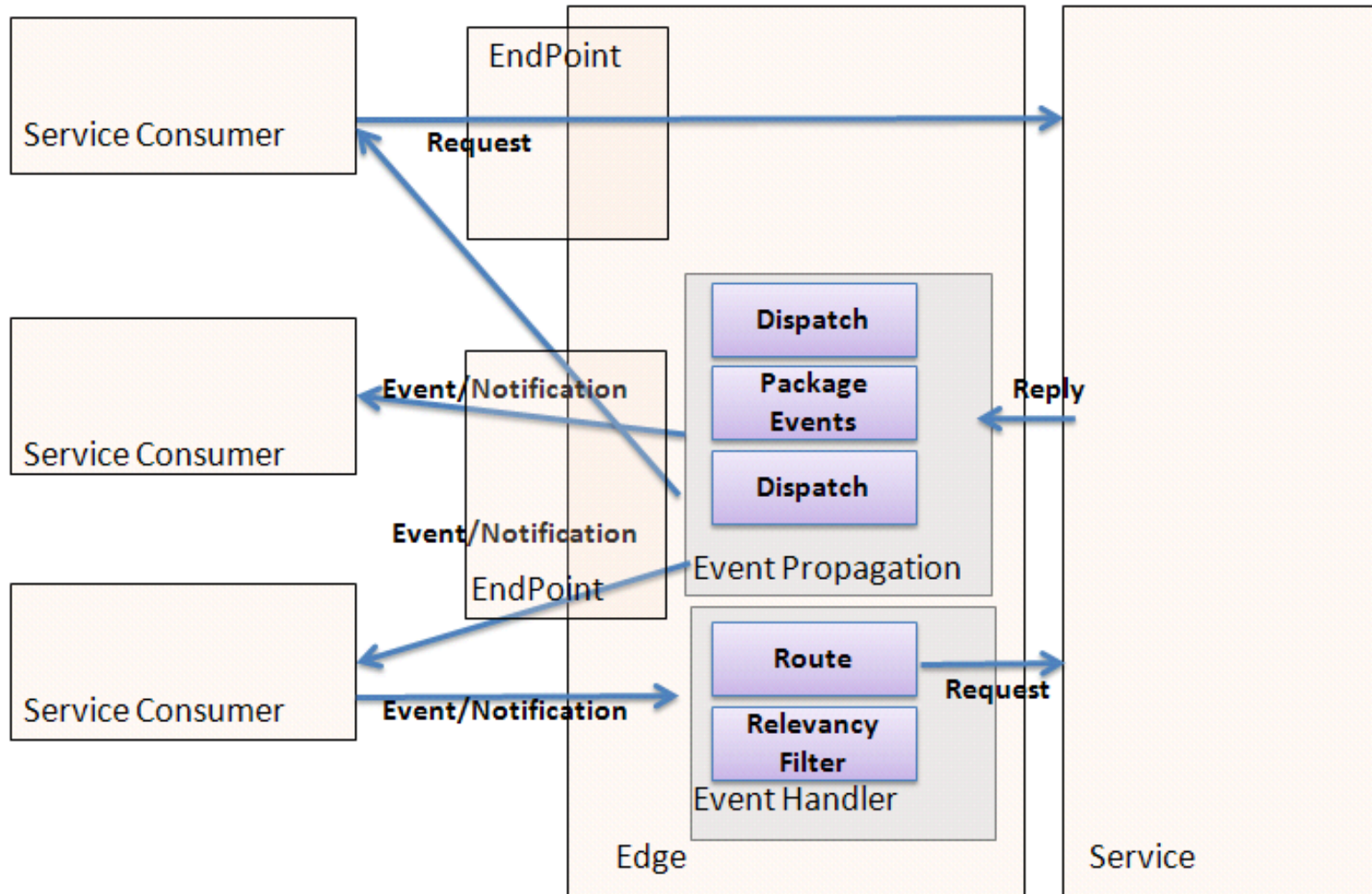


Turn out Complicated & Ugly™

Parallel Pipelines



Inversion of Communications



Consumer view

```
var sendMmsEvent = new SendMmsEvent()  
{  
    FromNumber = simpleMessageDetails.DialedNumber,  
    Subject = mmsContents.Subject,  
    ToNumber = simpleMessageDetails.Sender,  
    ImageExtension = mmsContents.ImageExtension,  
    ImageAsByteArray = mmsContents.Image,  
    TextAsByteArray = mmsContents.Text  
};  
eventBroker.RaiseEvent(sendMmsEvent);
```



Service view

[ServiceContract]

```
public interface IPostOffice : ImContract, IHandleSendCoupon,  
IHandleSendSms, IHandleStatus, IHandleAdminStatus, IHandleWapLink,  
IHandleSendMms  
{  
  
}
```

[ServiceContract]

```
public interface IHandleSendMms  
{  
    [OperationContract]  
    int SendMms(SendMmsEvent eventOccured);  
}
```

[DataContract]

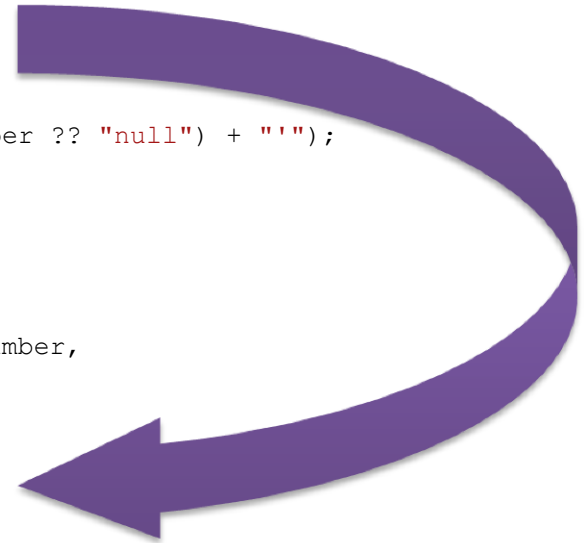
```
public class SendMmsEvent : ImEvent  
{  
    /// <summary>  
    /// end user's number. should be in international format: +[country-code]number. Example: +491737692260  
    /// </summary>  
    [DataMember]  
    public string ToNumber { get; set; }  
    /// <summary>  
    /// service's number, usually a short-code. Example: 84343  
    /// </summary>  
    [DataMember]  
    public string FromNumber { get; set; }  
    ///
```

Edge translates external structures to internal ones

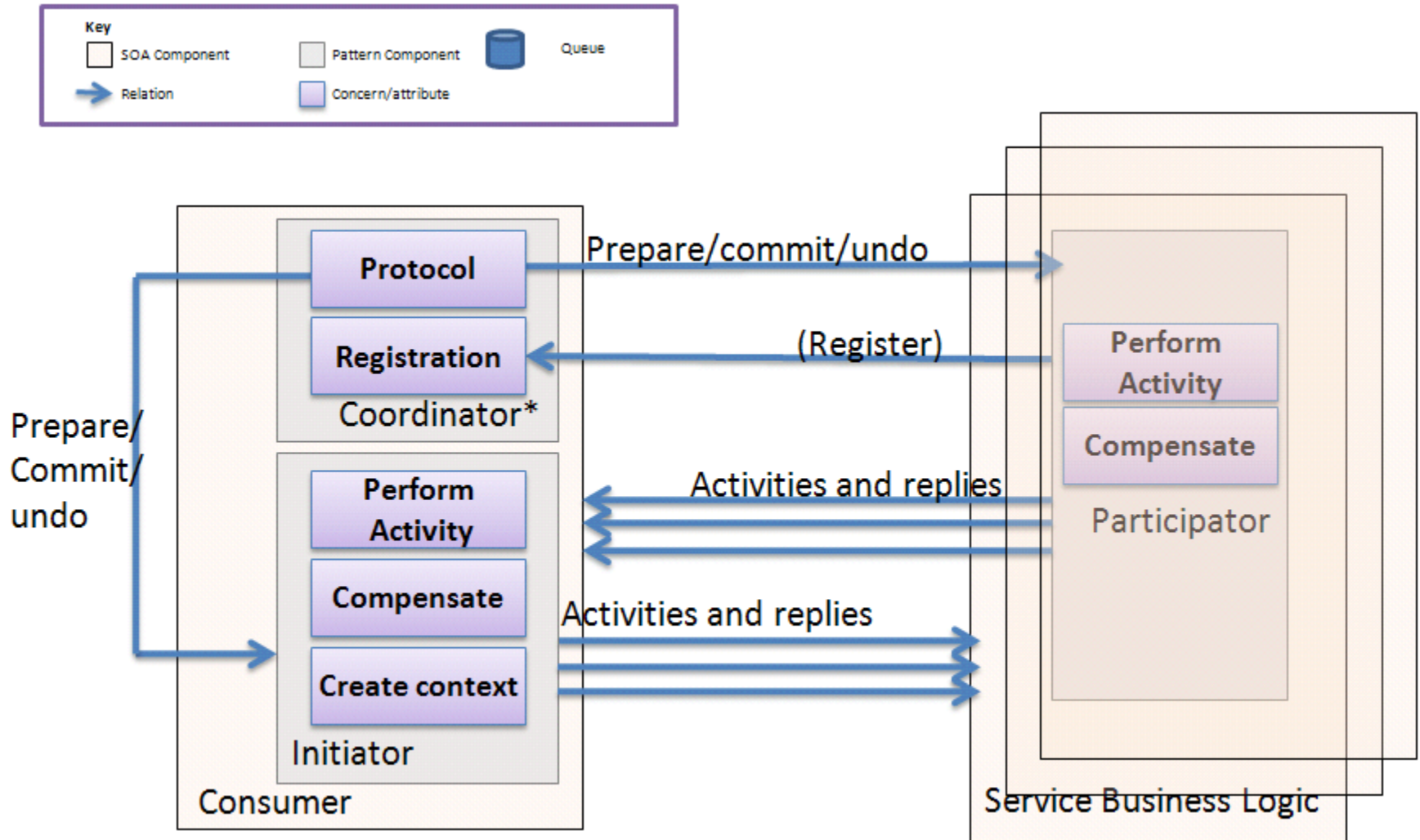
```
public int SendMms(SendMmsEvent eventOccured)
{
    var eventContext = eventOccured.ToString();
    if (log.IsDebugEnabled)
        log.Debug("inside 'SendMms', event context = [" + eventContext + "]);
    var fromNumber = eventOccured.FromNumber;
    var sender = mmsSenderFactory.Get(fromNumber);
    if (null == sender)
    {
        if (log.IsWarnEnabled)
            log.Warn("cannot get mms sender derived from '" + (fromNumber ?? "null") + "'");
        return 0;
    }

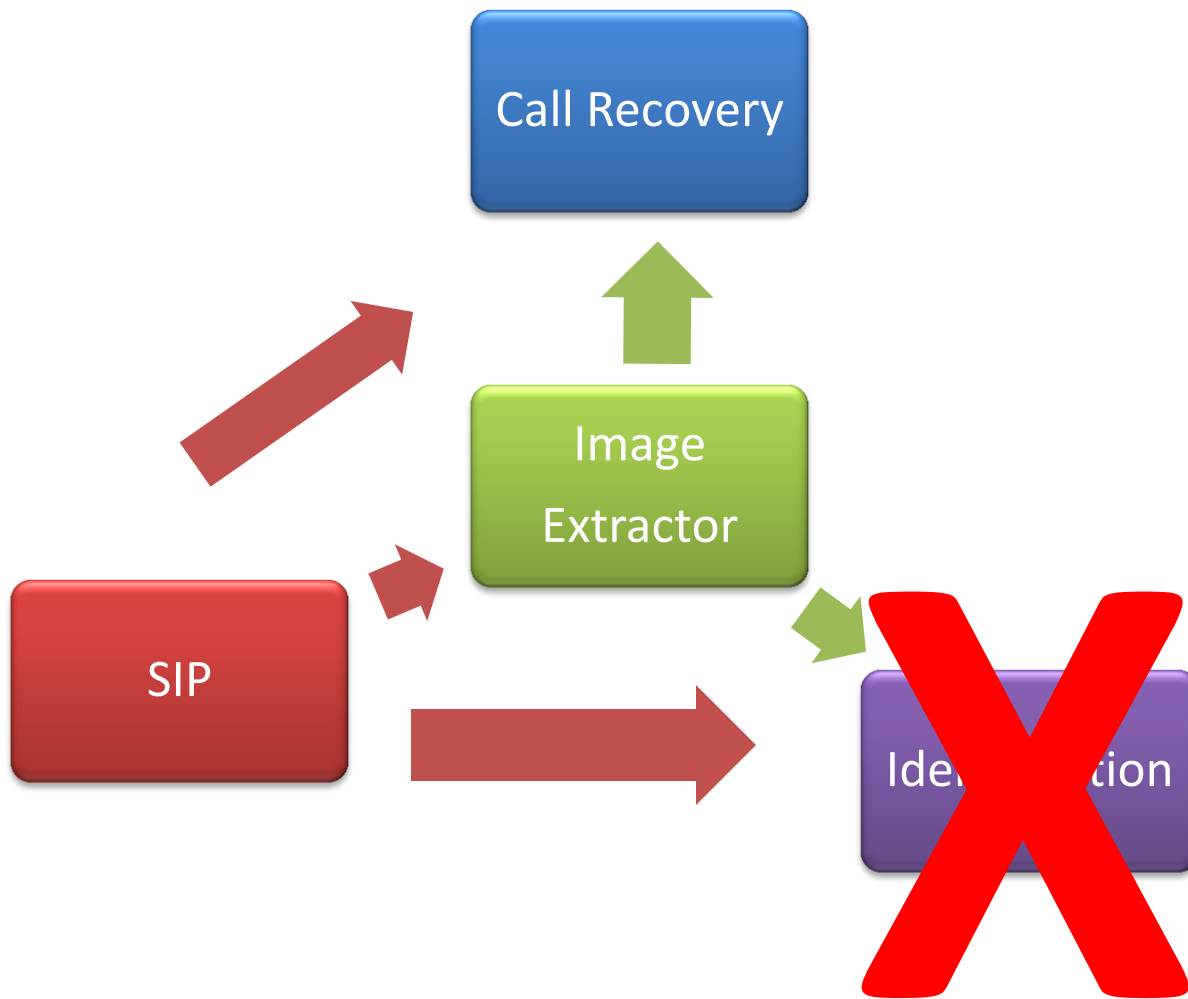
    IMmsSubmitResponse response;
    try
    {
        var mmsMessageDetails = new MmsMessageDetails(eventOccured.ToNumber,
            eventOccured.TextAsByteArray,
            eventOccured.ImageAsByteArray,
            eventOccured.ImageExtension,
            eventOccured.Subject);

        response = sender.Submit(mmsMessageDetails);
    }
    catch (Exception ex)
    {
        log.Error("cannot send mms message, context = [" + eventContext + "]", ex);
        return 0;
    }
    if (log.IsInfoEnabled)
    {
        var responseMessage = (null == response) ? "null" : response.ToString();
        log.Info("sent mms with event context = [" + eventContext + "], response = [" +
responseMessage + "]);
    }
}
```

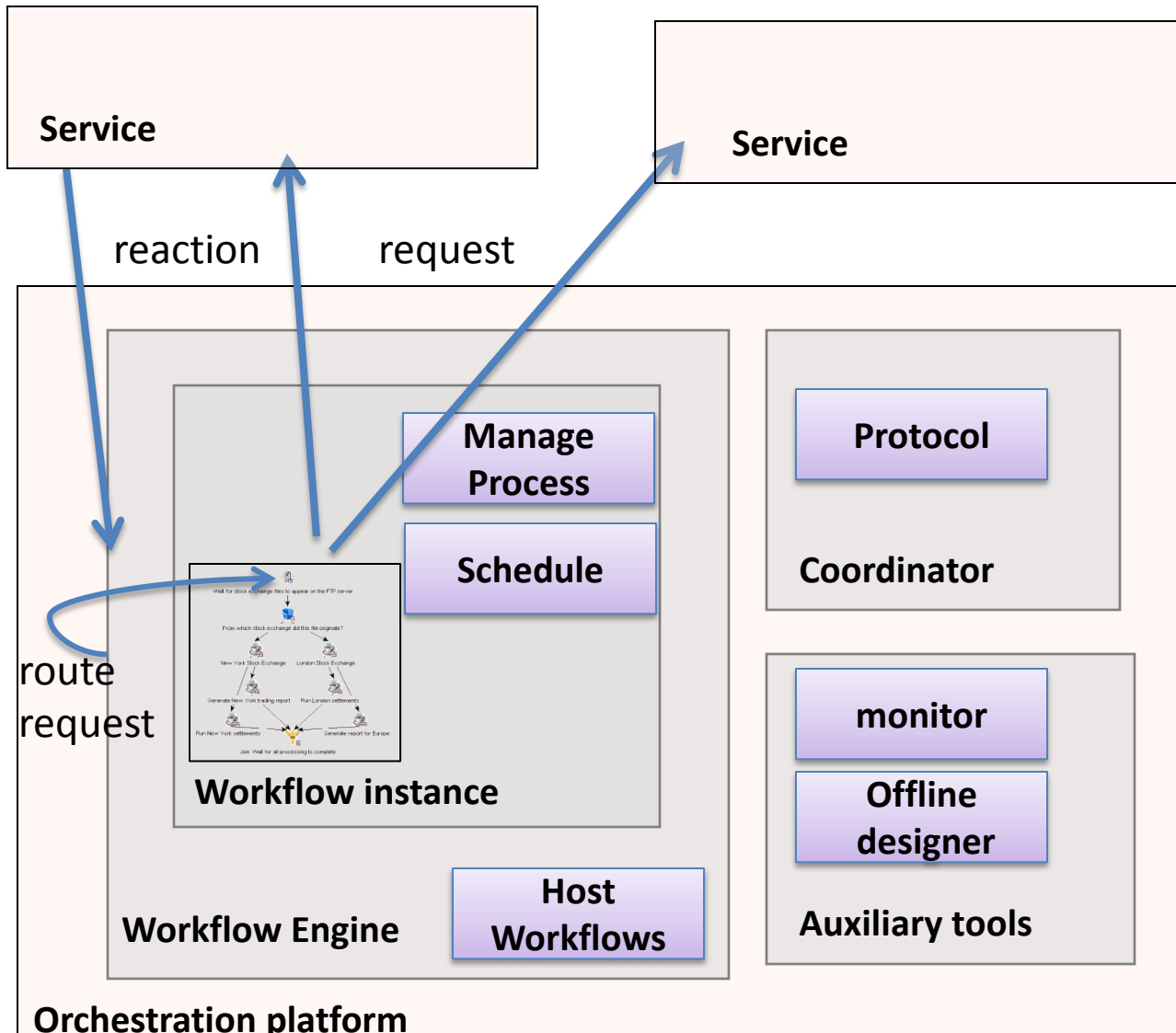


Sagas tie instances together for conversations





Alternative : Orchestration

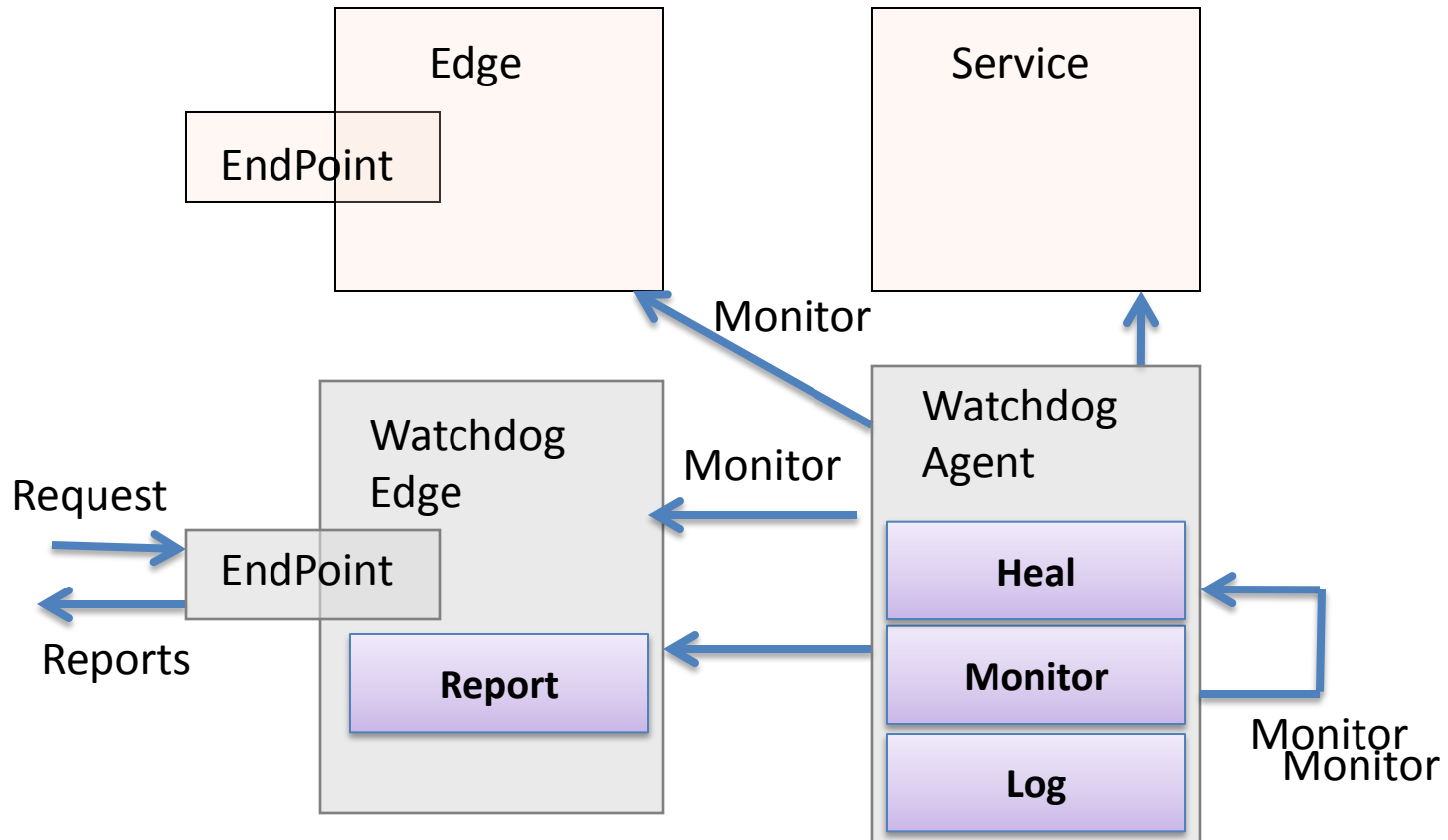


Be Wary of Nano-Services



CHALLENGE - MANAGEMENT

Blogjecting Watchdog



Blogjects concept is about collaborating objects



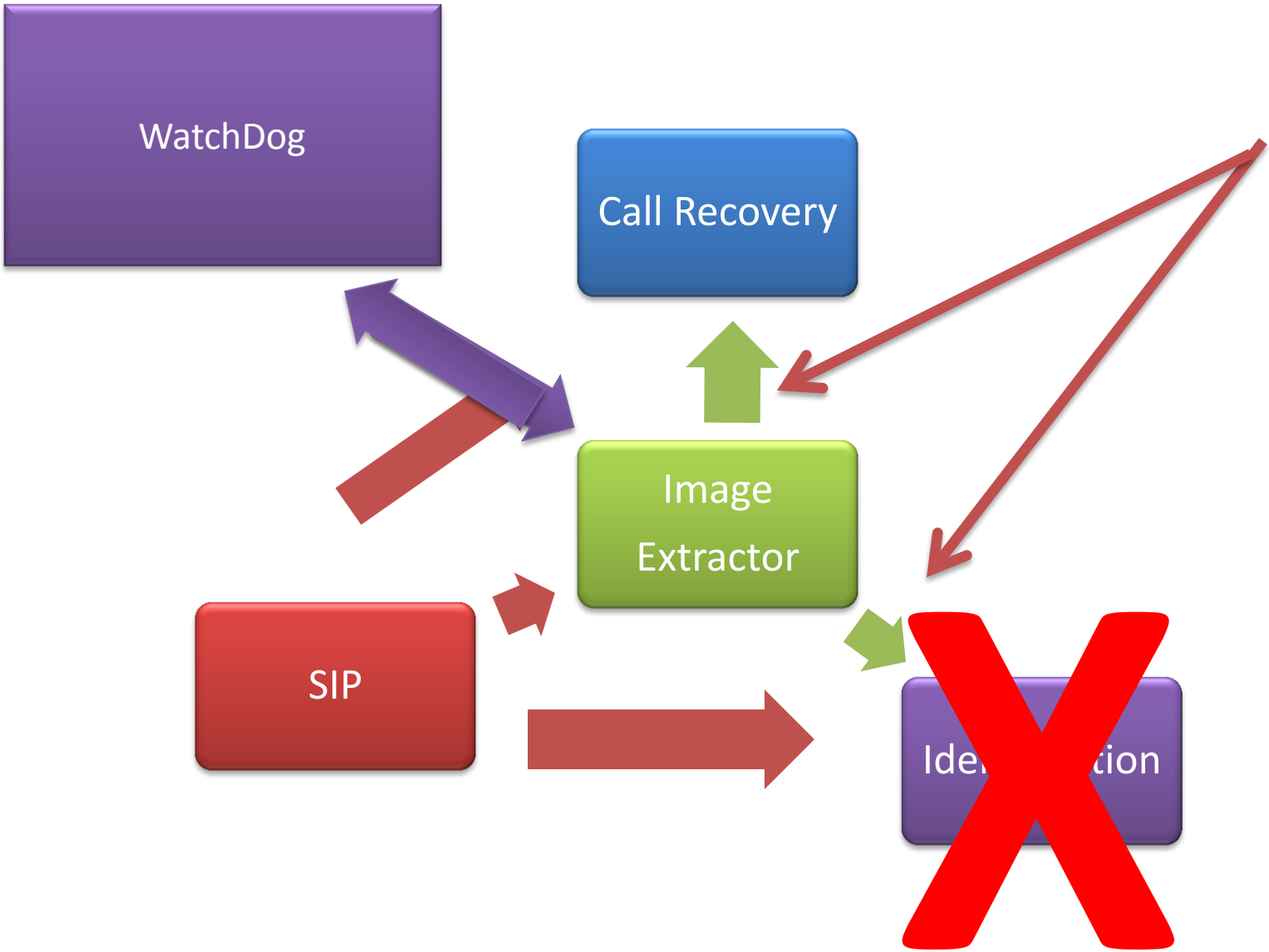
WatchDog

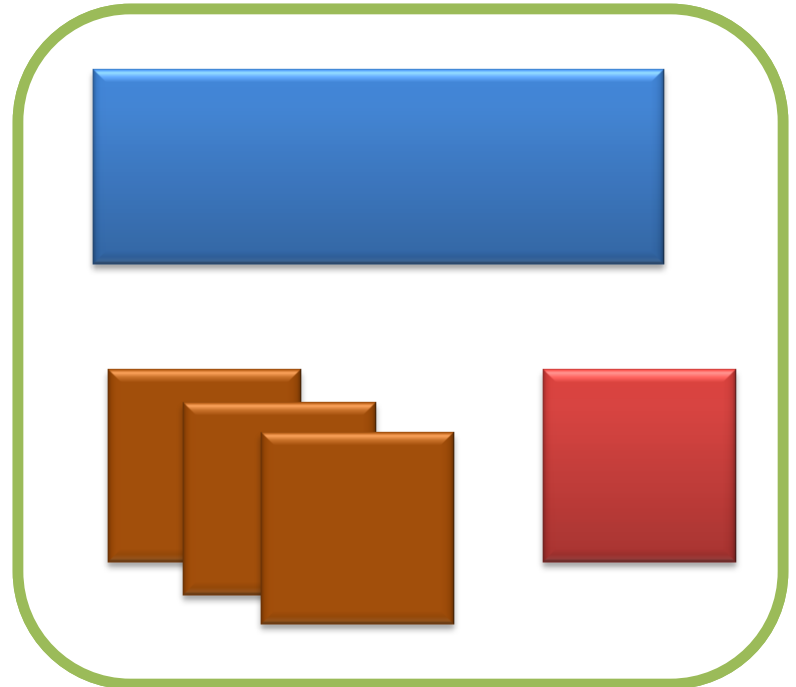
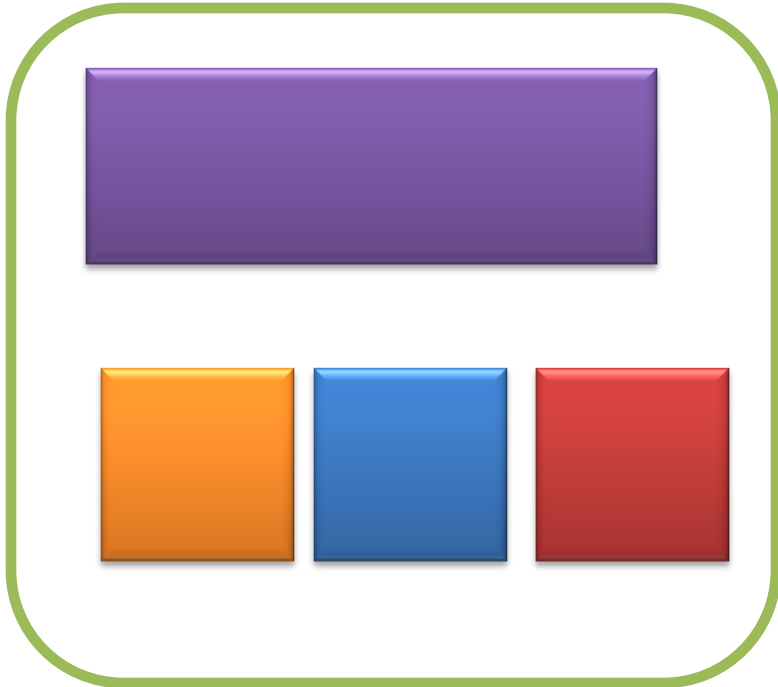
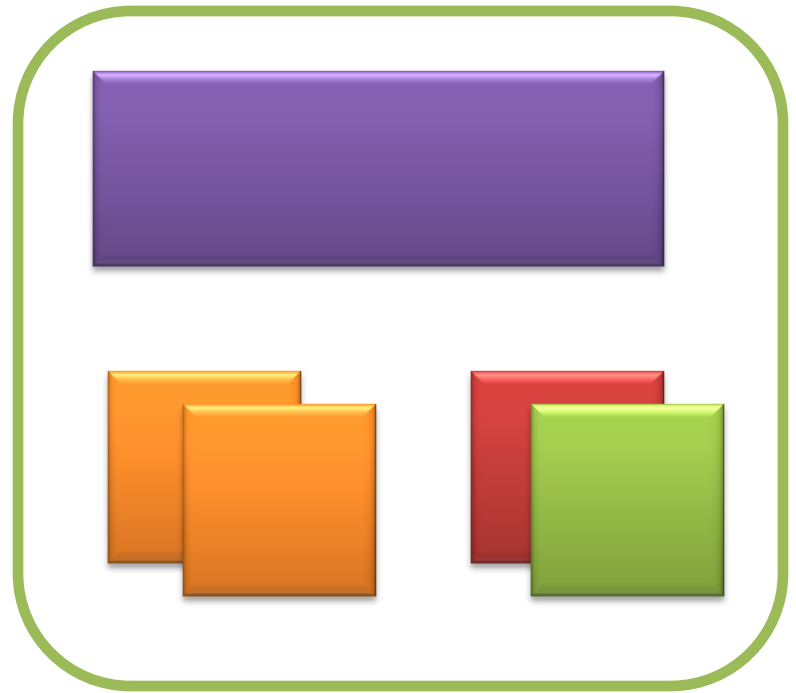
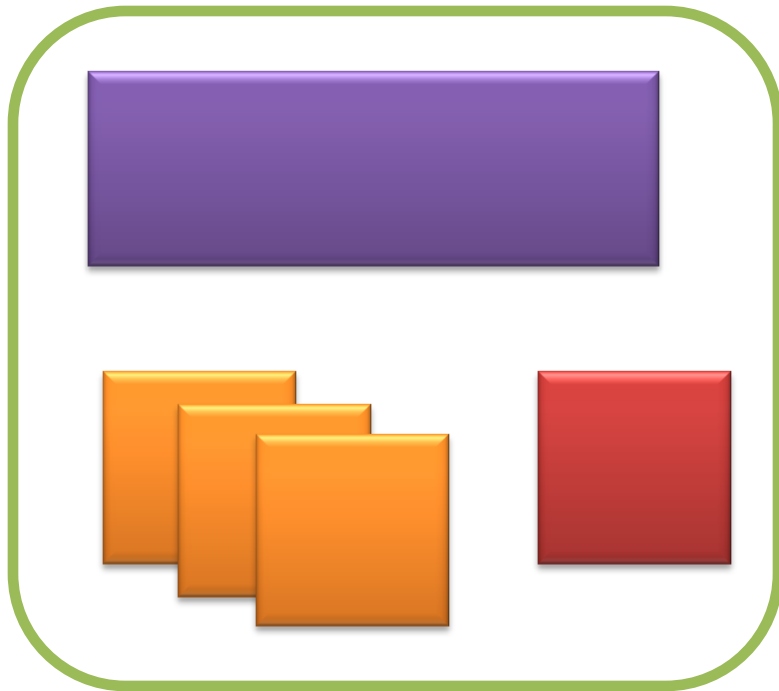
The diagram consists of a large purple rectangular box at the top labeled 'WatchDog'. Below it are three smaller rectangular boxes: an orange one on the left labeled 'Service A', a blue one in the middle labeled 'Service B', and a red one on the right labeled 'WDWatcher'. All boxes are contained within a light green rounded rectangular border.

Service A

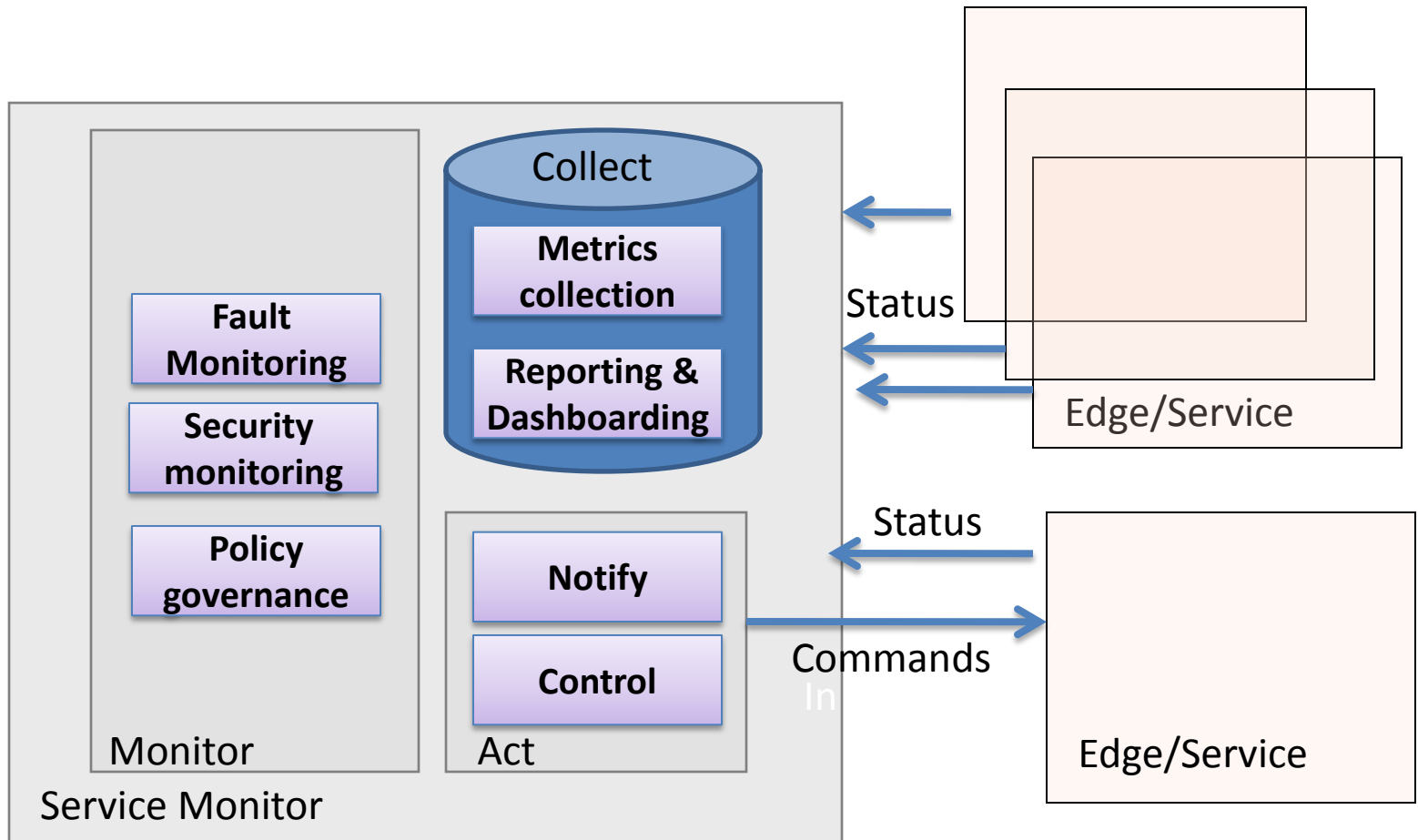
Service B

WDWatcher

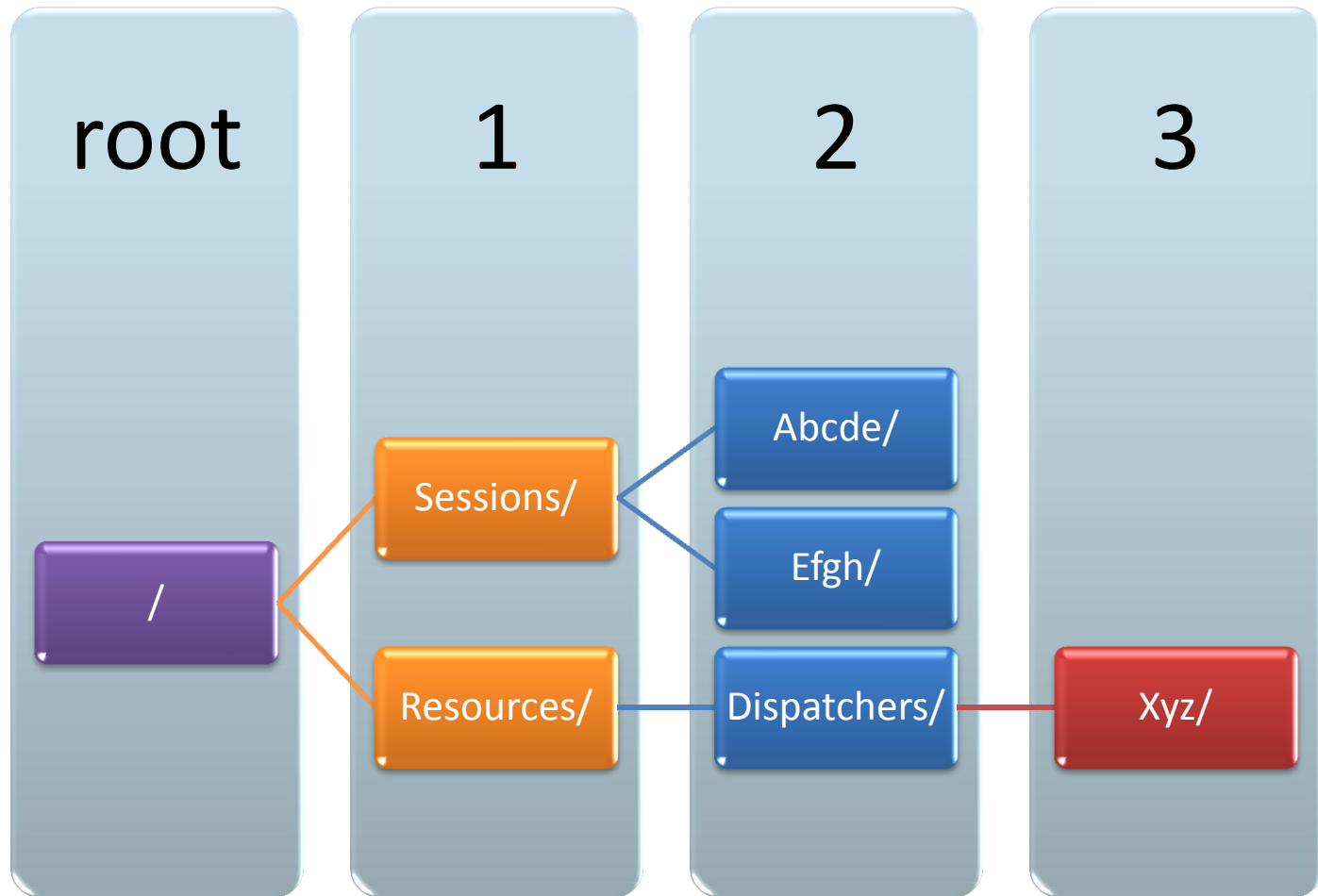




Service Monitor



RESTful resource management



<http://devrig:52141/RM/Sessions/abc/>

- ATOMPUB

- Session details

- URI (ID)

- State (start/end/status etc.)

- Resources

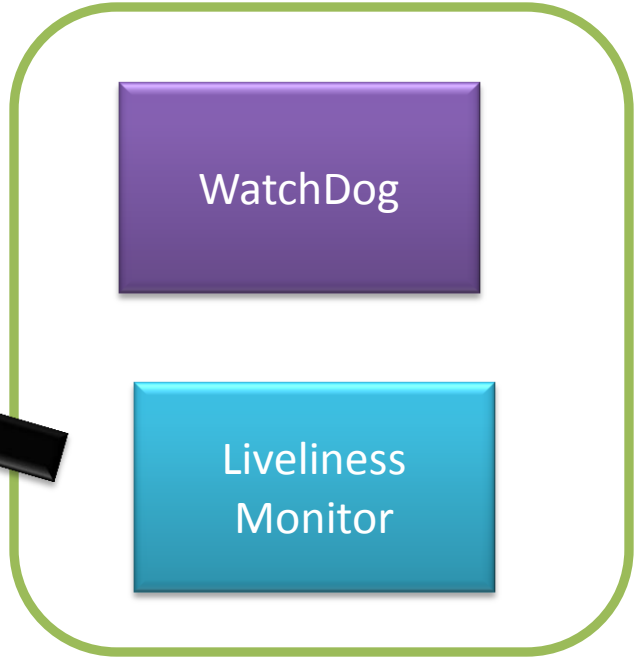
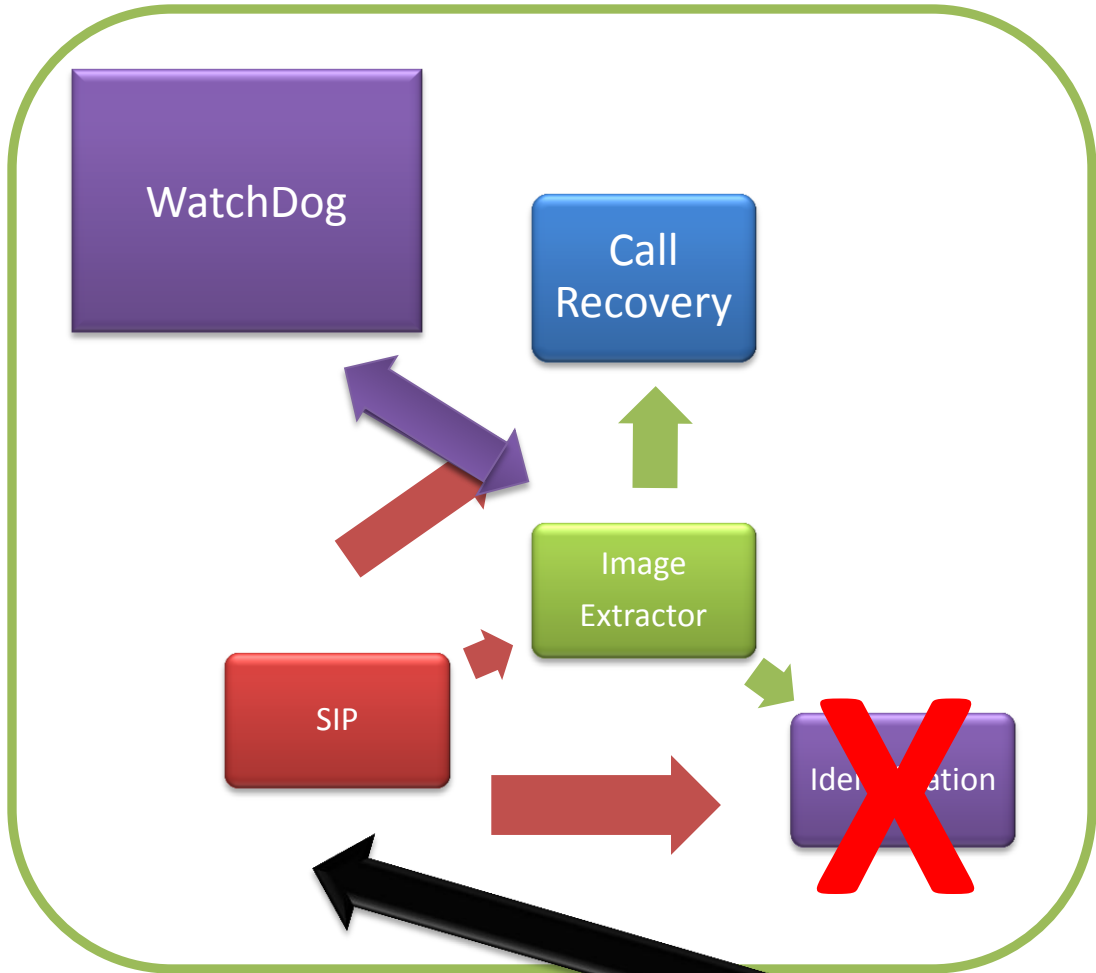
- Knows status

- URI for the Resource representation on the RM

- URI for the Resource itself

Keep the BIT

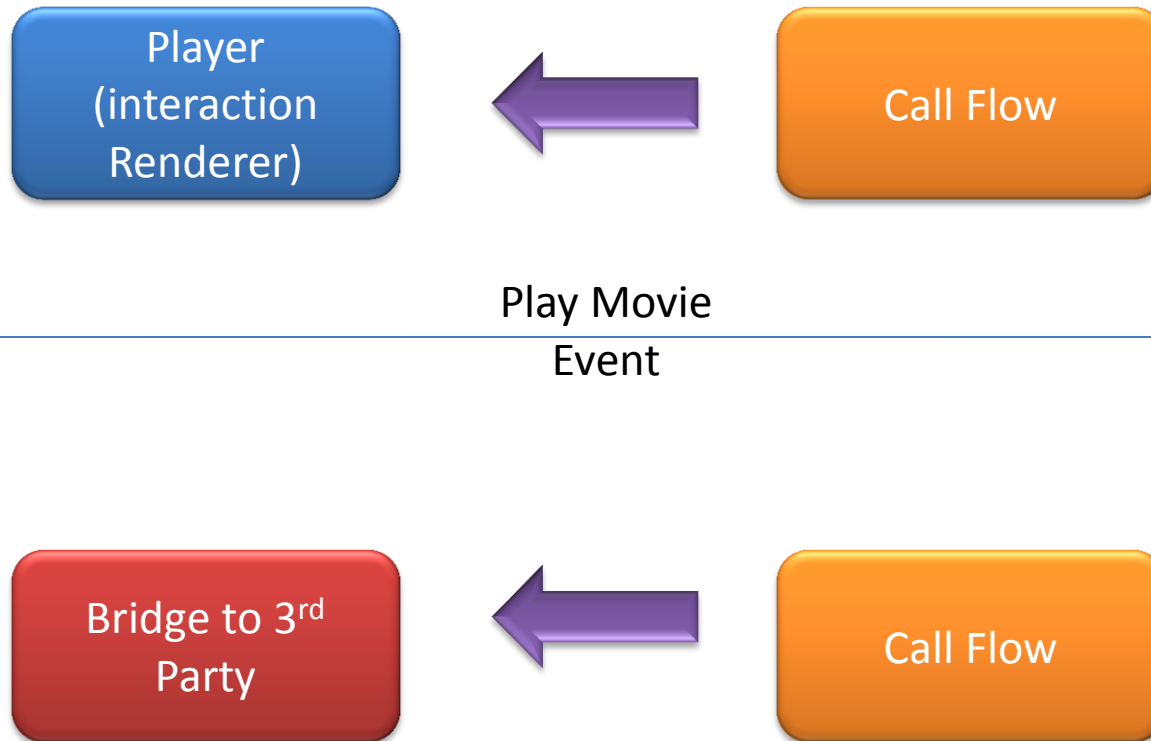




3G Call

**CHALLENGE – MULTI-TENNANCY
(LIES, I TELL YOU, ALL LIES)**

Same event different subscribers



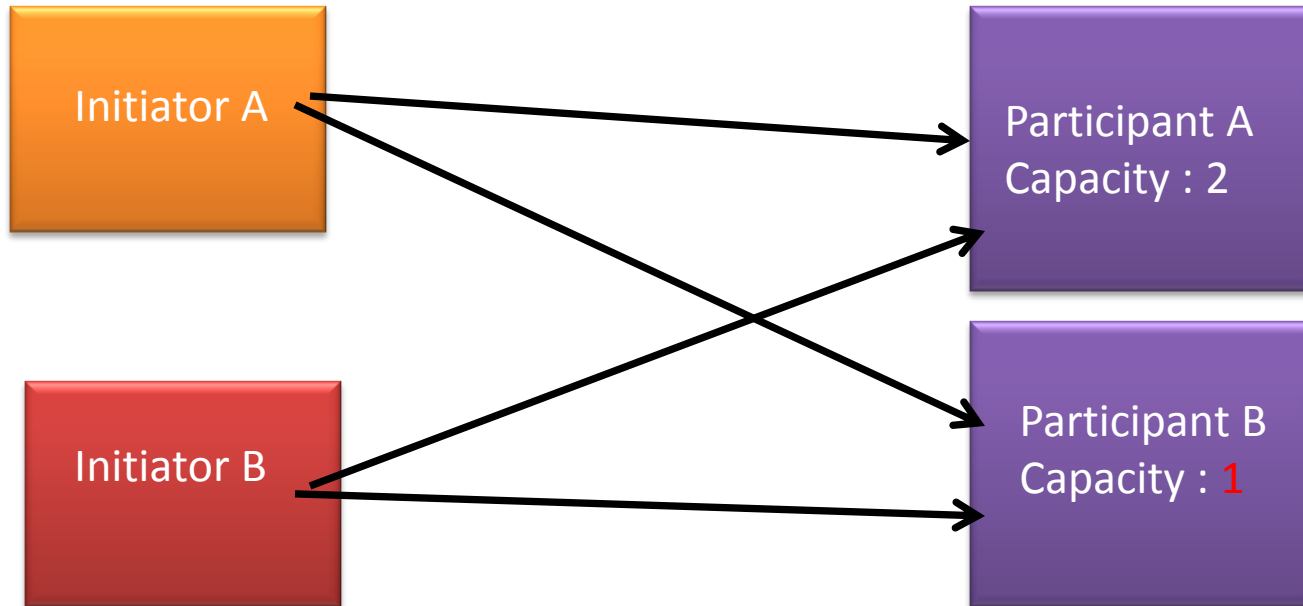
Routing

```
[ServiceContract]
[Participate("3G")]
public interface ImPlayer : ImContract, IHandleCallStarted, IHandle
IHandlePlayMovie, IHandleCallAborted
{
}
```

```
[ServiceContract]
[Participate("3GPartner")]
public interface ImXsightsGateWay : ImContract,
    IHandleCallAborted,
    IHandlePlayMovie,
    IHandleReadyForSearch,
    IHandleSearchStarted,
    IHandleJoinThirdParty

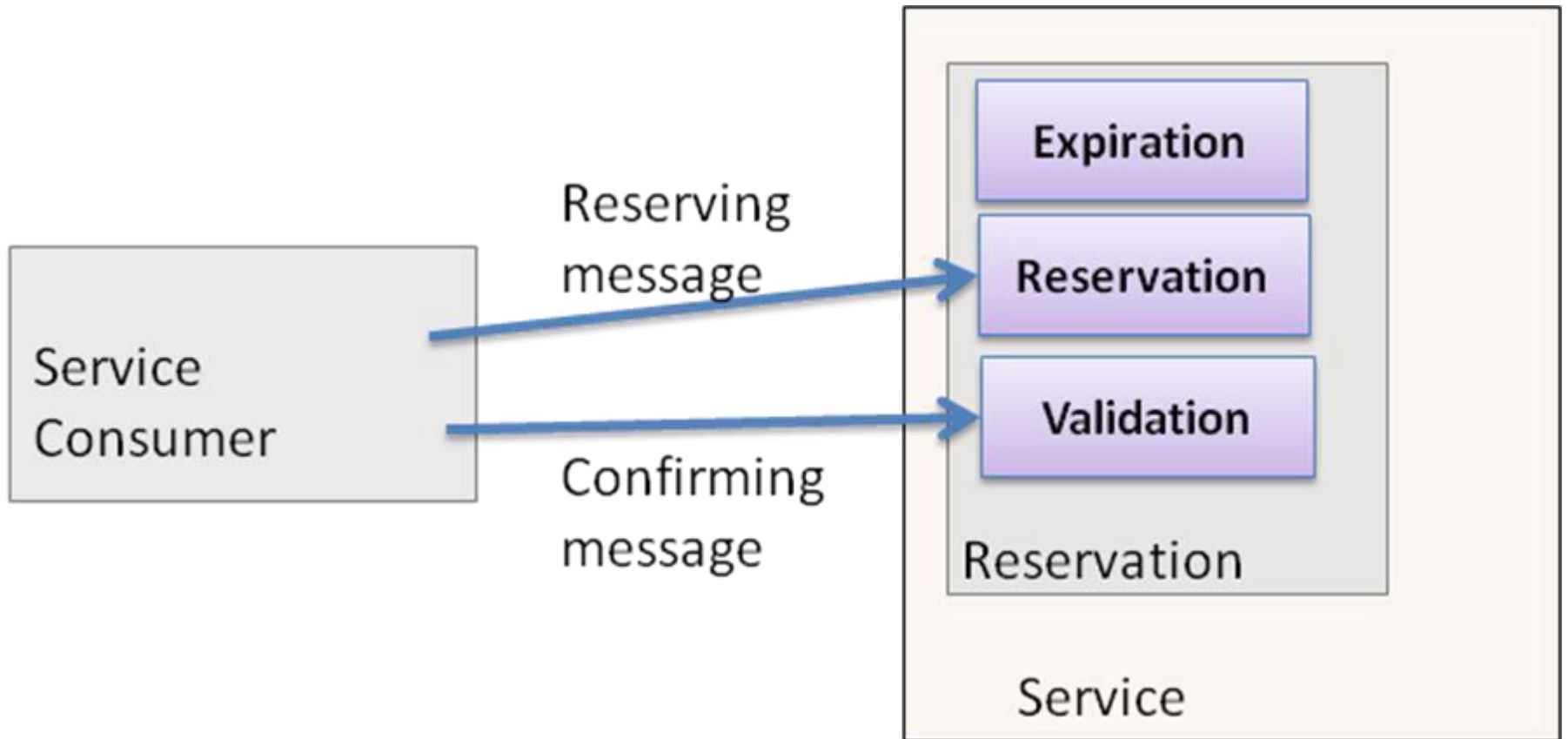
{
}
```

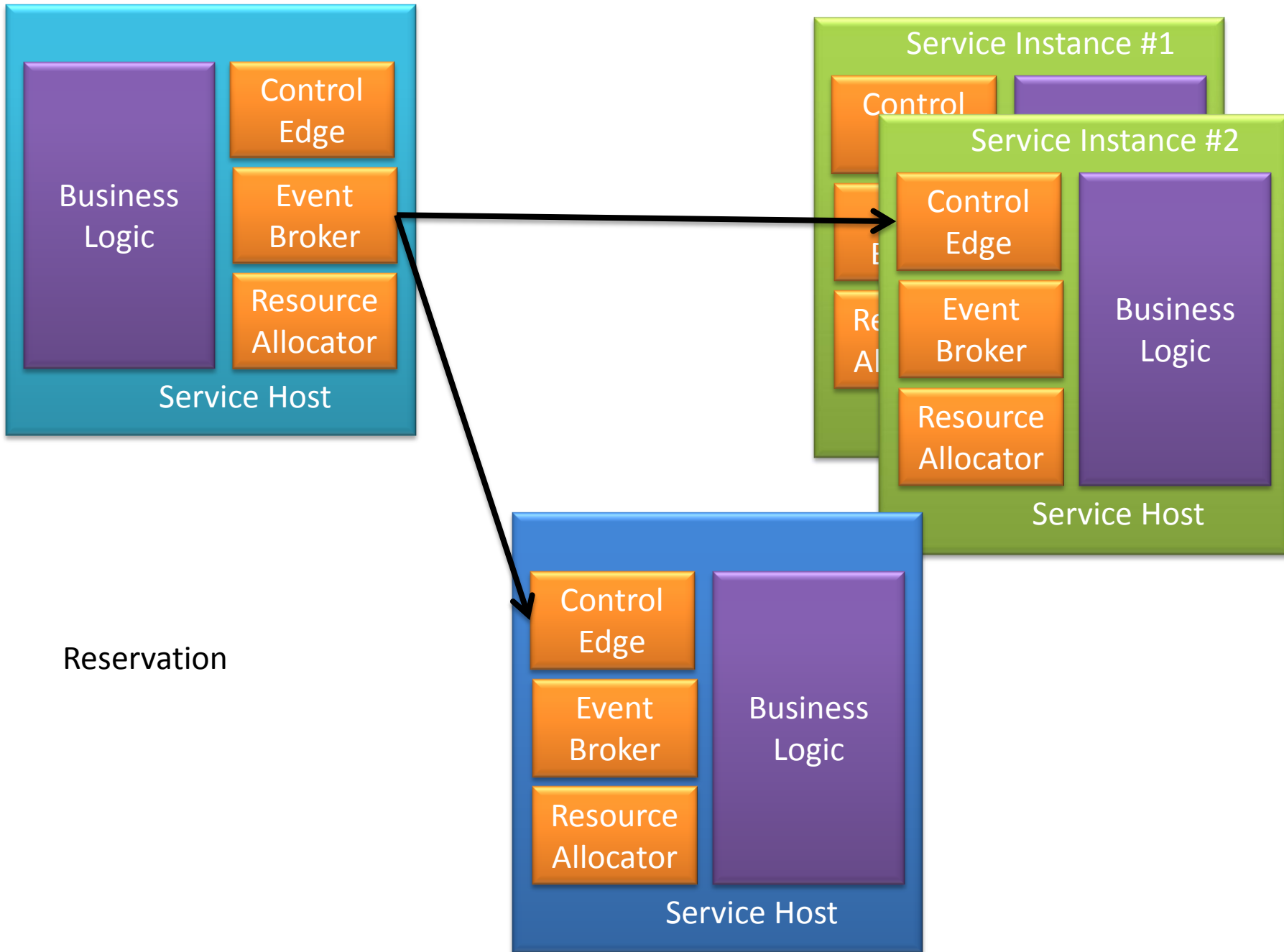
Raise a saga initiating event



Now what ?!

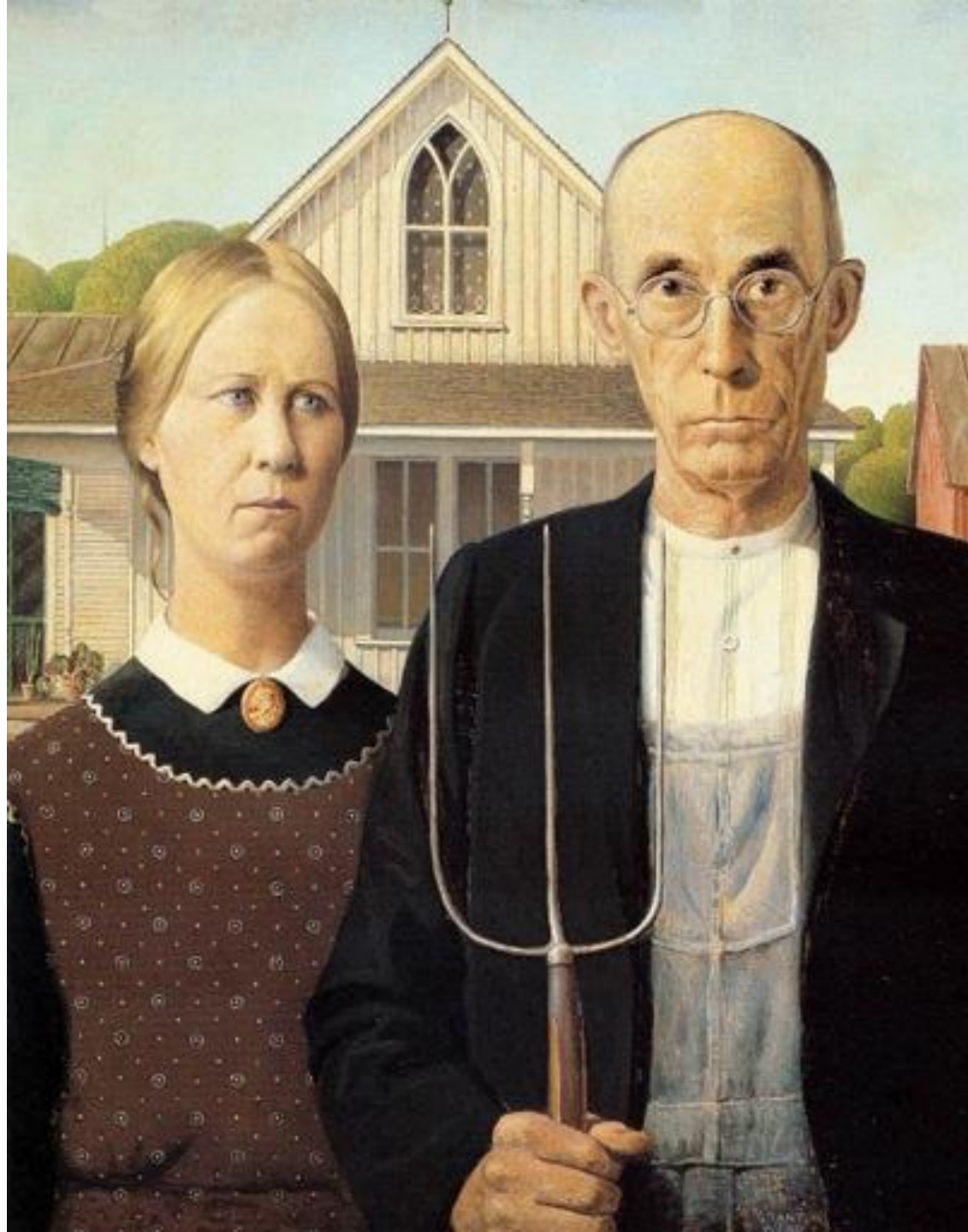
Reservation Pattern





Reservation

Good old
2PC to the
rescue



Takeaways



- Things break
- Decouple
- Fail Fast
- Monitor & Detect
- Compensate
- Throw state unto others

Arnon Rotem-Gal-Oz

[@arnonrgo](#)

<http://arnon.me>

<http://www.cloudvalue.com>