MICRO SERVICES .NET

DVLPR
CNSLTNT
PRDCT MGR
DVLPR

VITALY BAUM

AGENDA

- leaving monolith
- example
  - dynamic proxy + communications
  - elastic search (elk)
  - service discovery
- drawbacks

WTF?!

What is it?

"Microservices" - yet another new term on the crowded streets of software architecture.

What is it?

Single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API

CAPTAIN OBVIOUS

LEAVING MONOLITH

PROCESS

namespaces
classes
functions

MICROSERVICES

process 1

...

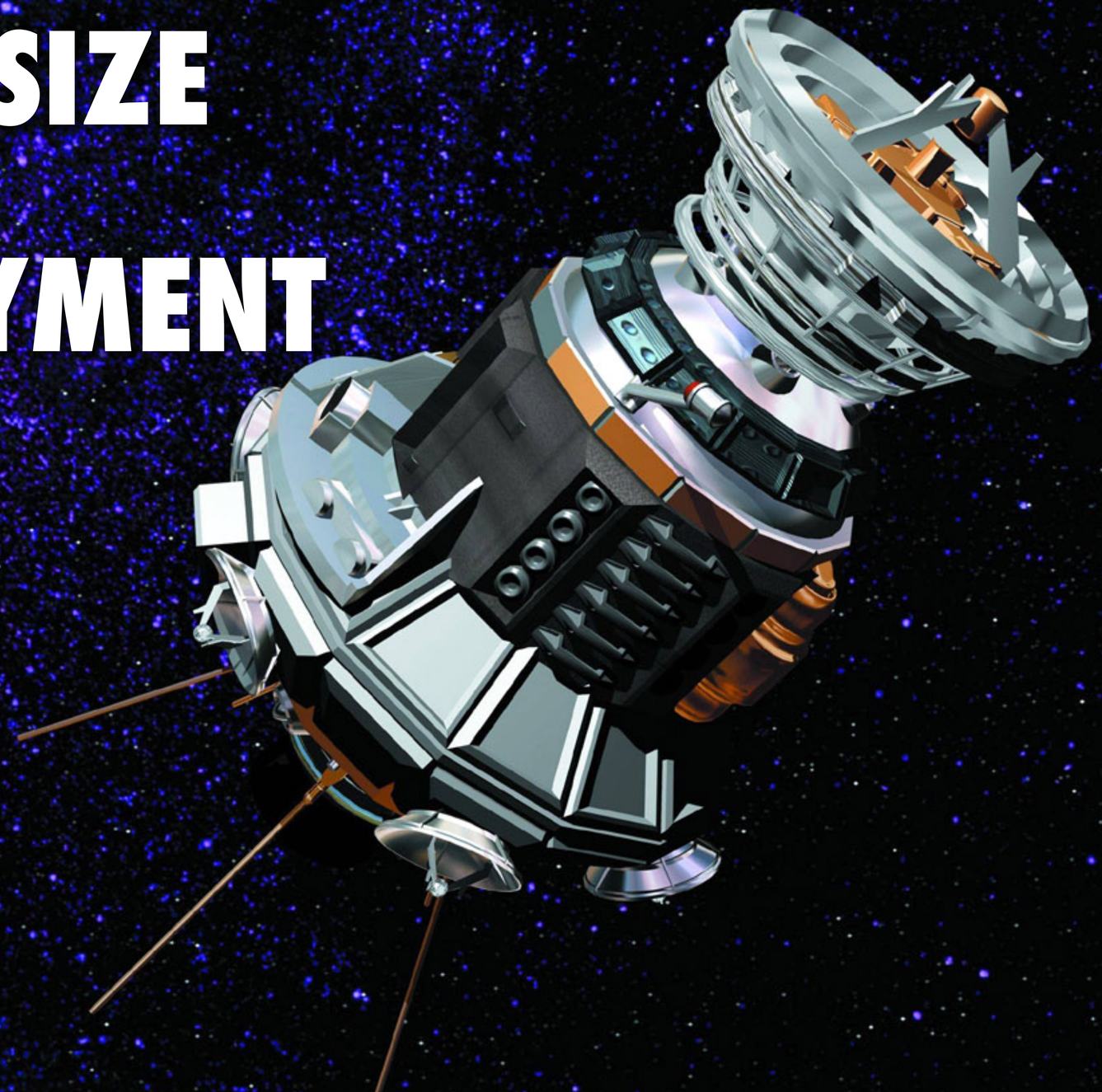
process N

SIZE

CHANGE IT.
OFTEN.
WITH CLOUDS

RIGHT SIZE
IS
SHORT-TERM
REWRITING

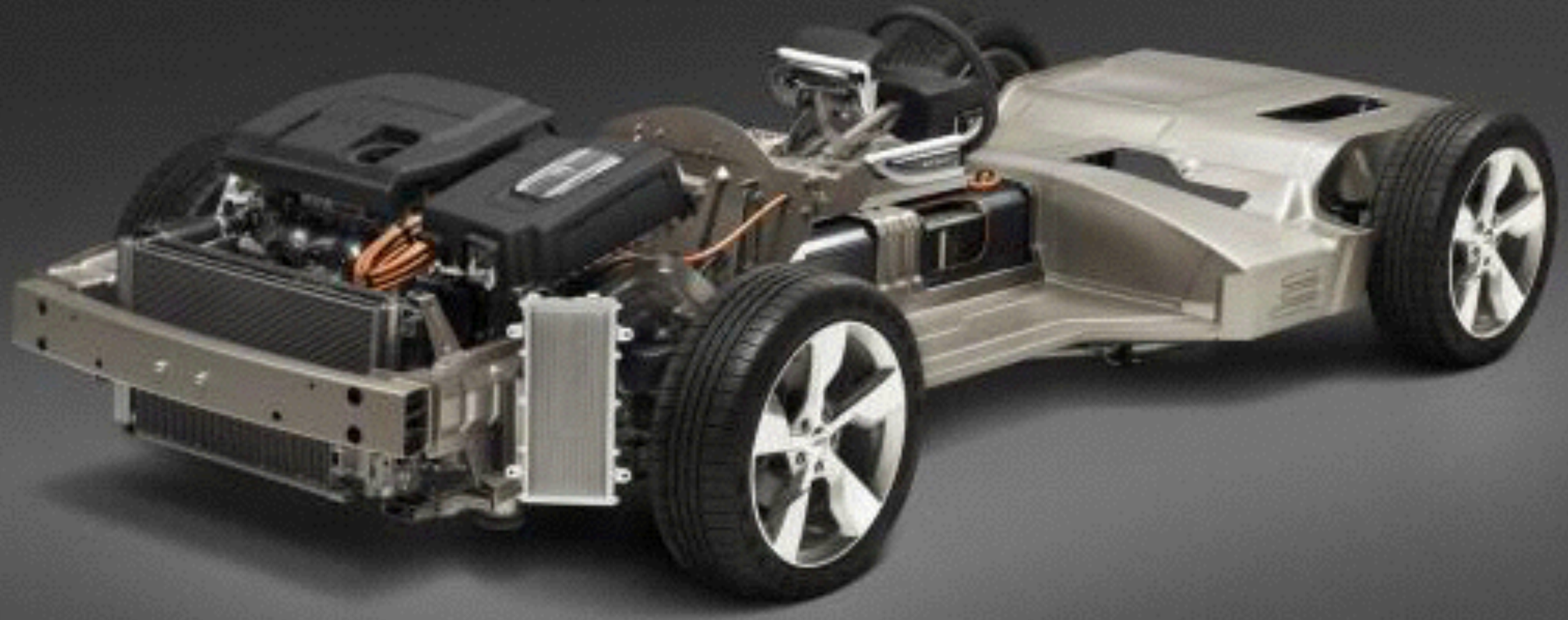
RIGHT SIZE IS DEPLOYMENT UNIT



FREEDOM OF DEPLOYMENT AND SCALE

DEPENDECIES

CHANGE SERVICES. NOT COMPONENTS

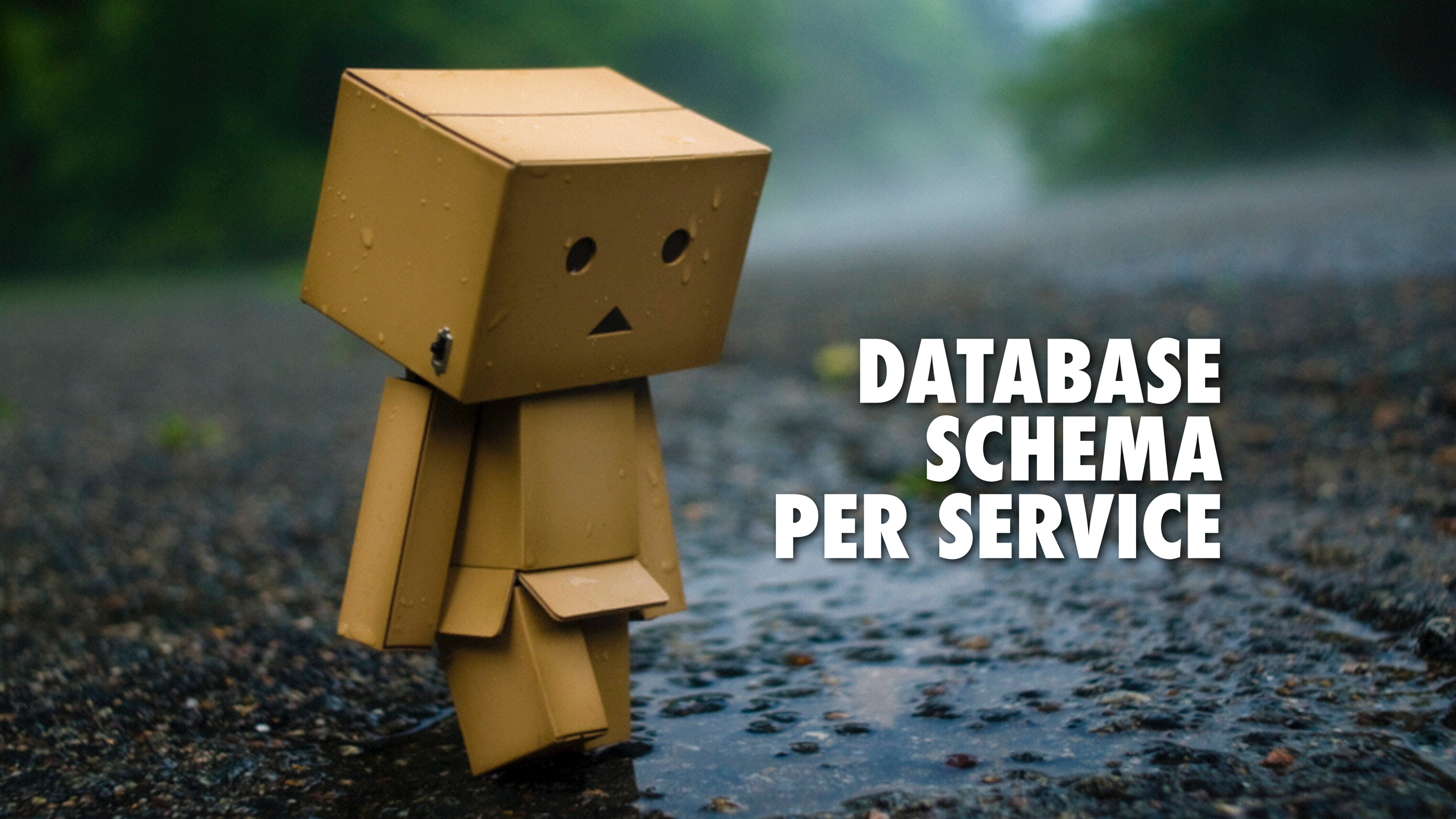


DO NOT
SHARE
LIBRARIES

BOUNDARIES.
MAKE IT RIGHT

DO SINGLE THING.
WELL
hdwallpapersdesktop.com

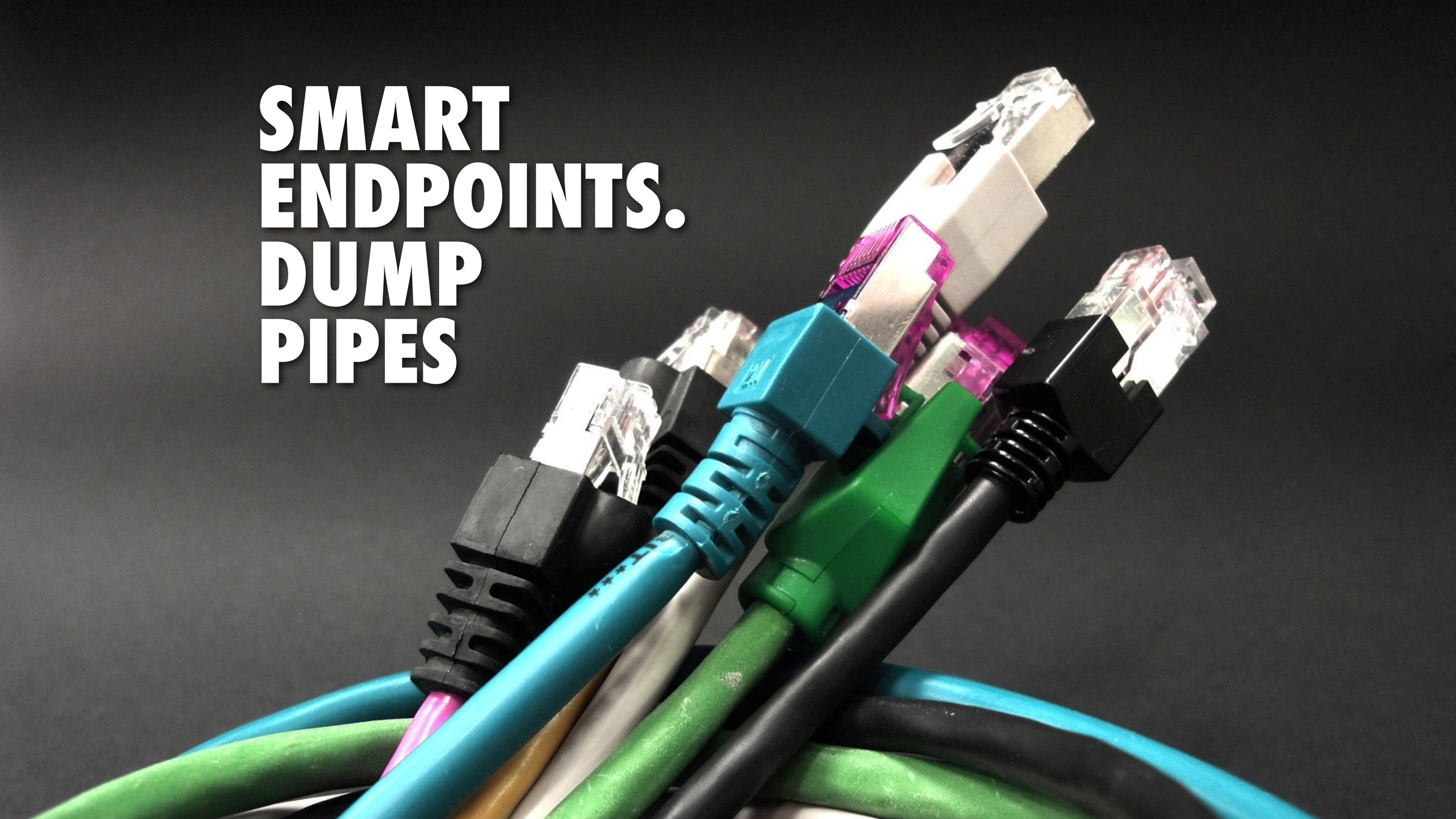
ISOLATION


DATABASE
SCHEMA
PER SERVICE

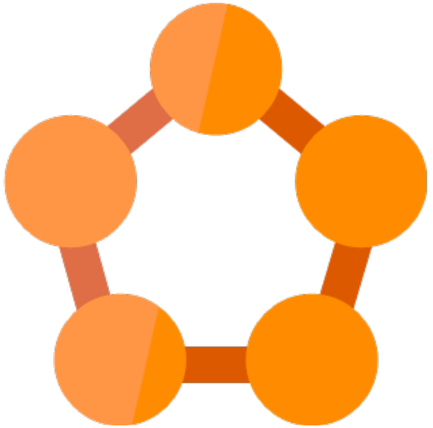
COMMUNICATIONS

DON’T
TRUST
NETWORK

SMART ENDPOINTS. DUMP PIPES



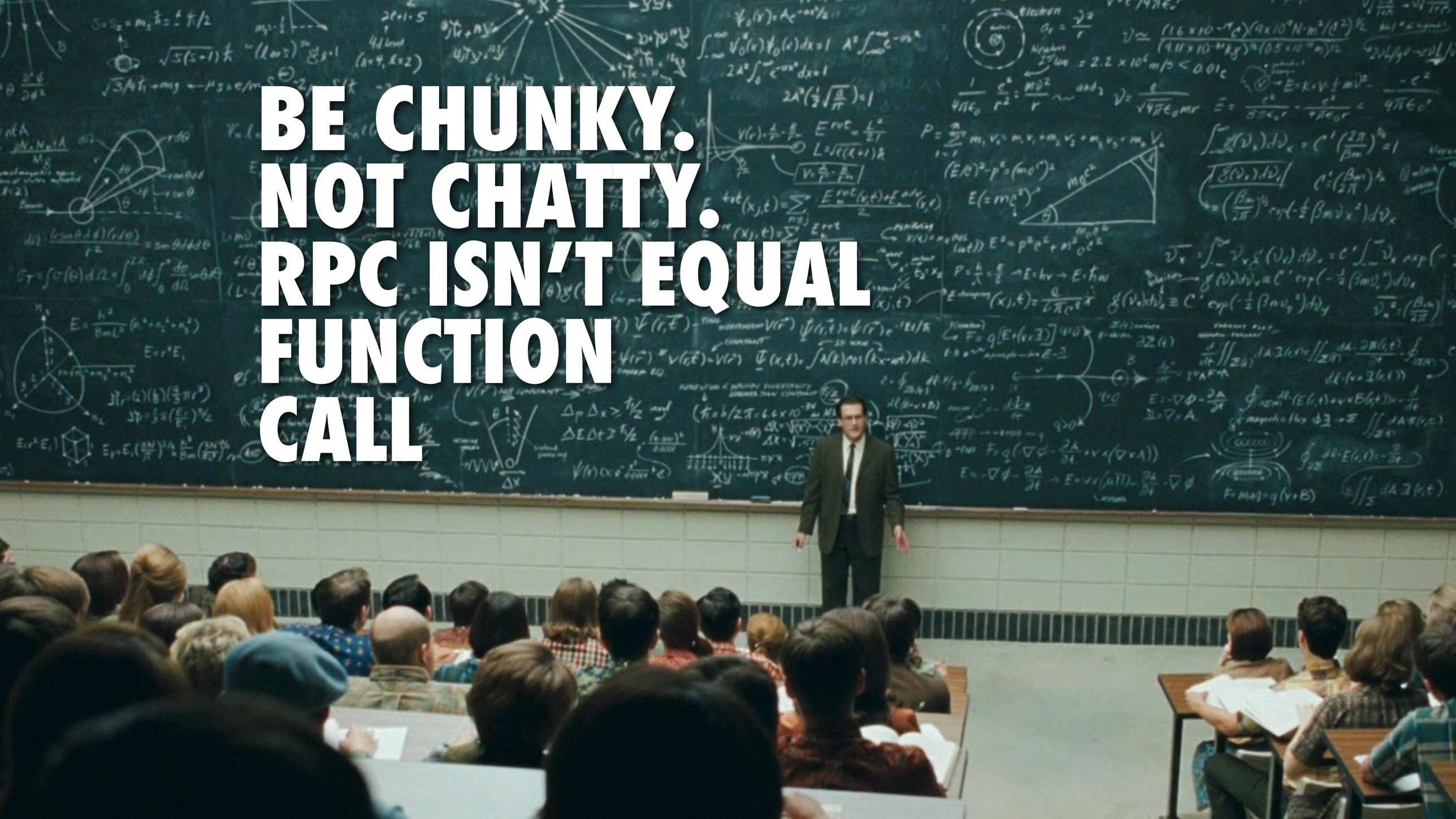
MICROSERVICES PLATFORMS



Azure
Service Fabric

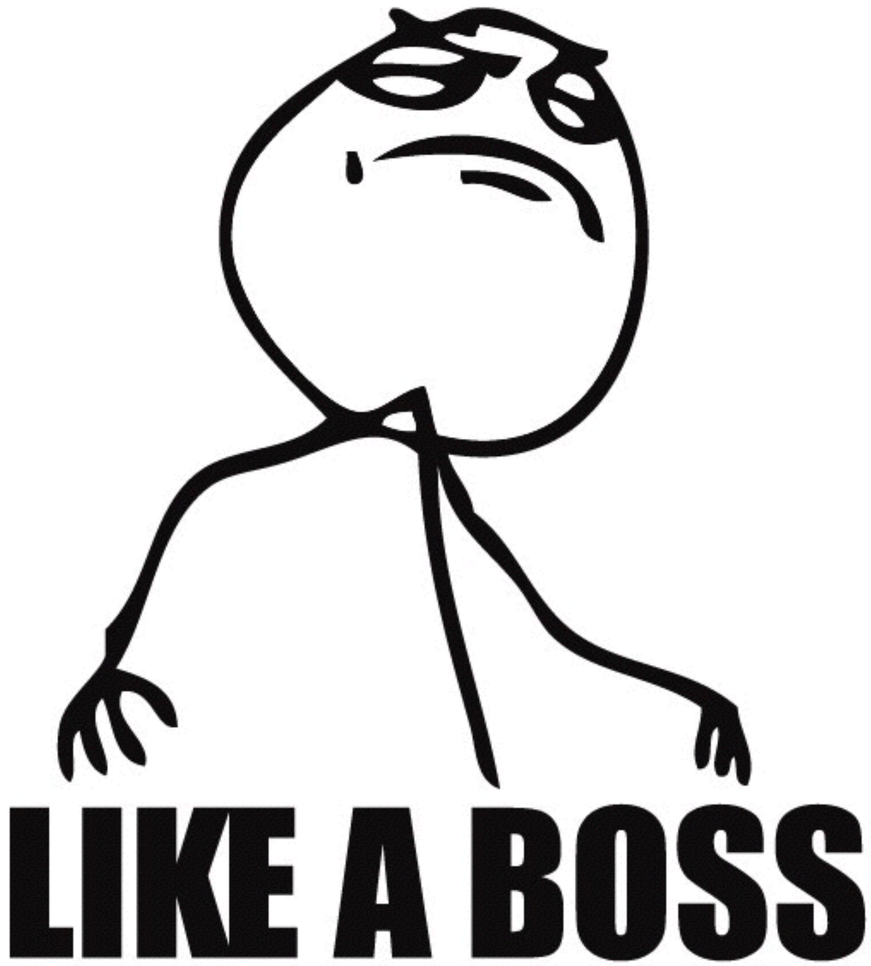
Akka

OTP


BE CHUNKY.
NOT CHATTY.
RPC ISN'T EQUAL
FUNCTION
CALL

EXAMPLE

FROM MONOLITH TO MICROSERVICE IN A MINUTES


LIKE A BOSS

CUSTOMER:
COUNT
MY MONEY

PROCESSING

REPORT

PROCESSING

EVENTS FLOW
REAL-TIME
CRITICAL

```
while (true)
  _valuesService
    .Receive(Magic.input())
```

WEB-BASED SERVER RENDRING

REPORT

```
_reportService.GetForLast5Min()

["123", "345", "678"]
```

```
IEnumerable<string> GetForLast5Min() {

  return _valuesService.Get(
    DateTime.Now.AddMinutes(-5);
  );
}
```

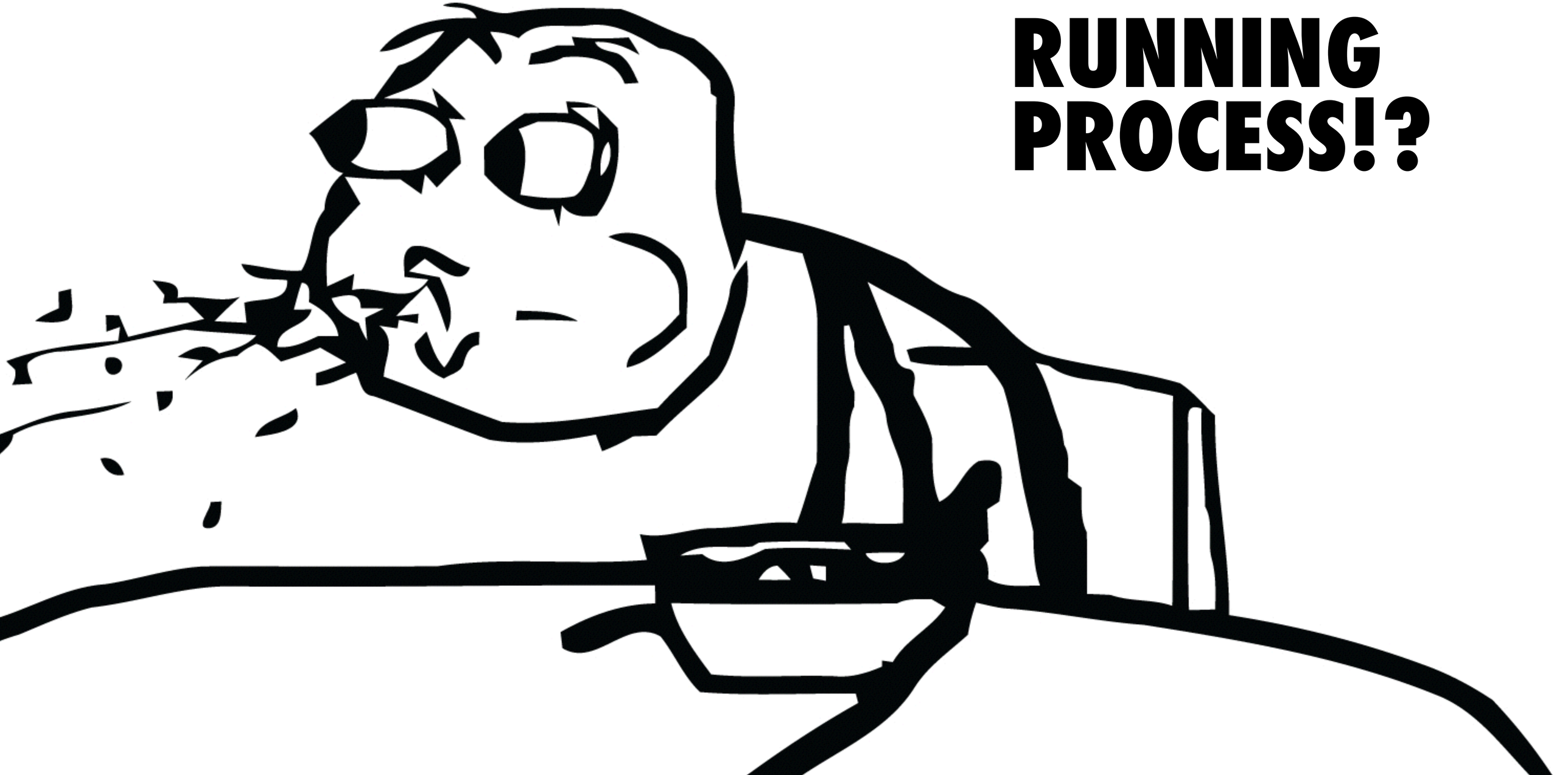
DEPENDENCY INJECTION

RUN FOR PROFIT!

dotnext — ~/Desktop/dotnext | bash — docker ◂ run.sh — 80×34

```
^^/D/dotnext >>> bash ./run.sh                    (*master+1) 13:18:08
Sending build context to Docker daemon 620.5 kB
Step 1 : FROM microsoft/aspnet:1.0.0-rc1-update1
 ---> 77ad3ef3ac04
Step 2 : RUN printf "deb http://ftp.us.debian.org/debian jessie main\n" >> /etc/
apt/sources.list
 ---> Using cache
 ---> 2ecb989efd41
Step 3 : RUN apt-get -qq update && apt-get install -qqy sqlite3 libsqlite3-dev &
& rm -rf /var/lib/apt/lists/*
 ---> Using cache
 ---> dfd53d39dbb0
Step 4 : COPY project.json /app/project.json
 ---> Using cache
 ---> 4cb3fb69de73
Step 5 : WORKDIR /app
 ---> Using cache
 ---> 58328e206ed2
Step 6 : RUN dnu restore
 ---> Using cache
```

CHANGE
REQUEST:
UPDATE ME
THE VIEW!


RUNNING
PROCESS!?

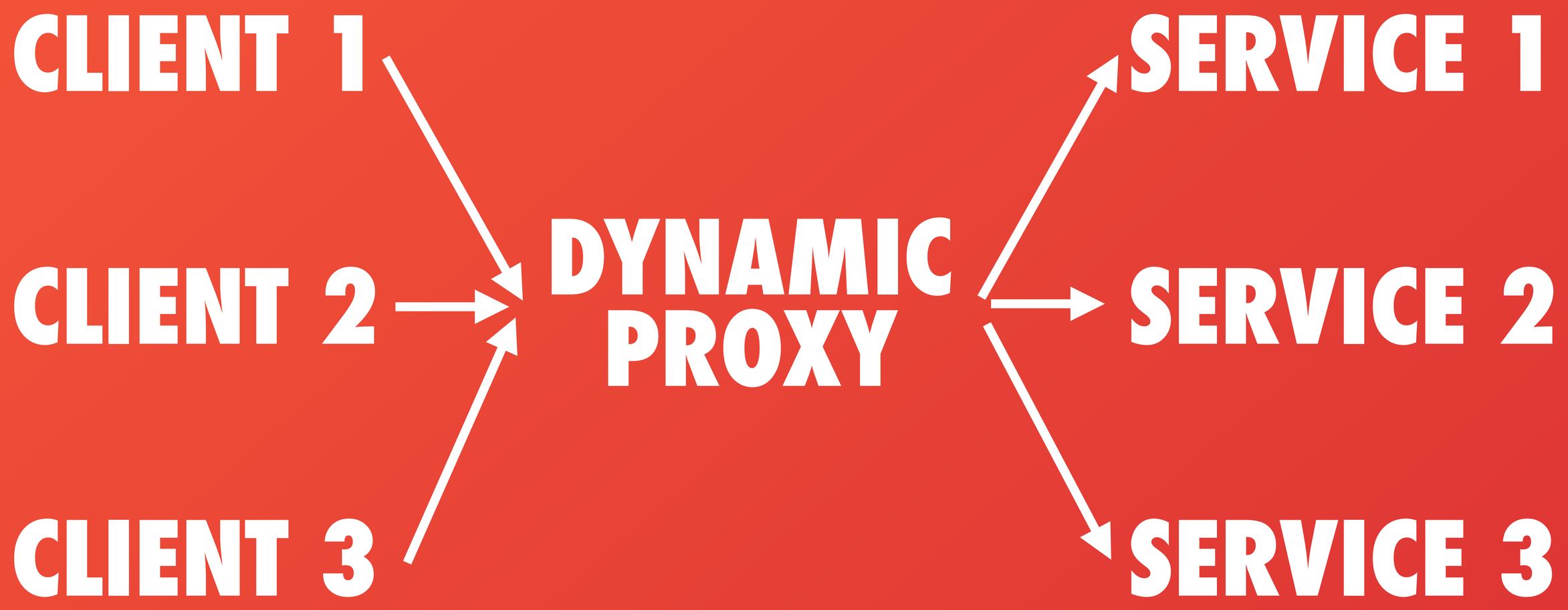
SOLUTION FOUND

```
IEnumerable<string> GetForLast5Min() {

  return _valuesService.Get(
    DateTime.Now.AddMinutes(-5);
  );
}
```

DEPENDENCY INJECTION

CLIENT 1
CLIENT 2
CLIENT 3
DI CONTAINER
SERVICE 1
SERVICE 2
SERVICE 3

```
IValueService _valuesService =
    DynamicProxy.Create<IValueService>();
```


CLIENT 1
CLIENT 2
CLIENT 3
DYNAMIC PROXY
SERVICE 1
SERVICE 2
SERVICE 3

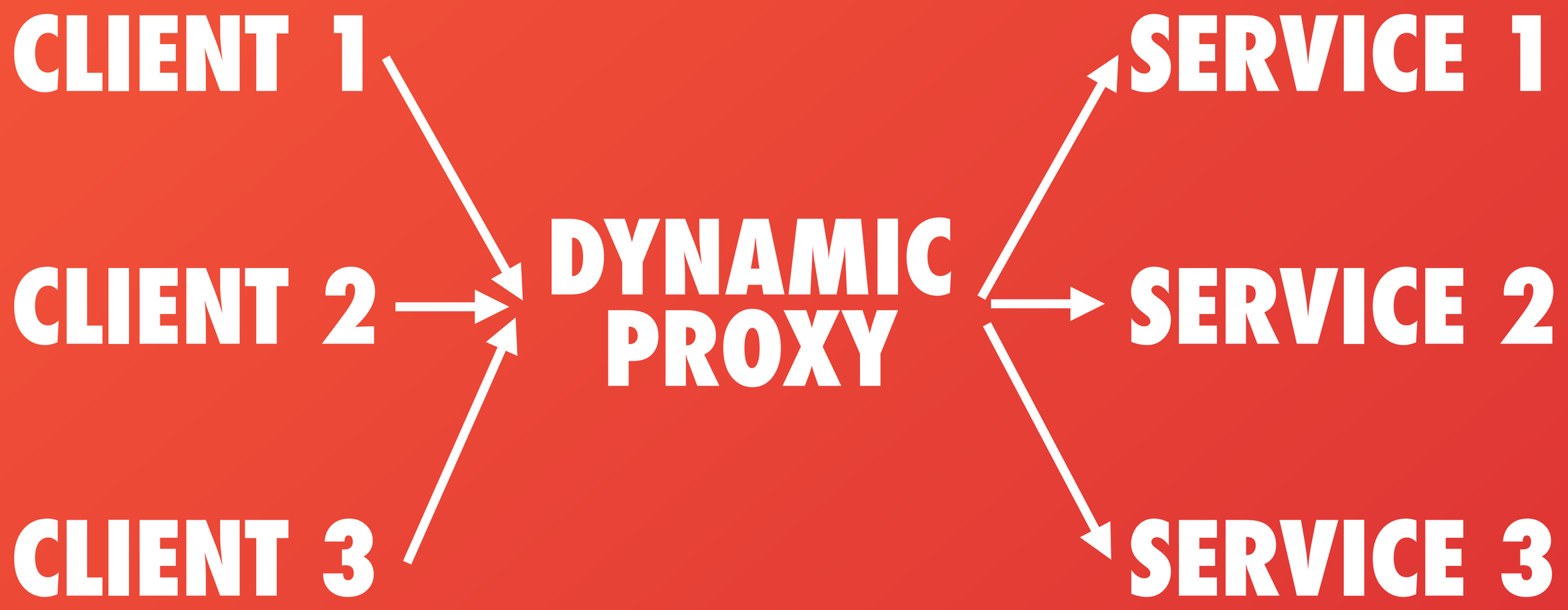
DYNAMIC PROXIES

http://www.castleproject.org/projects/dynamicproxy

https://github.com/mtamme/NProxy

https://github.com/ekonbenefits/impromptu-interface

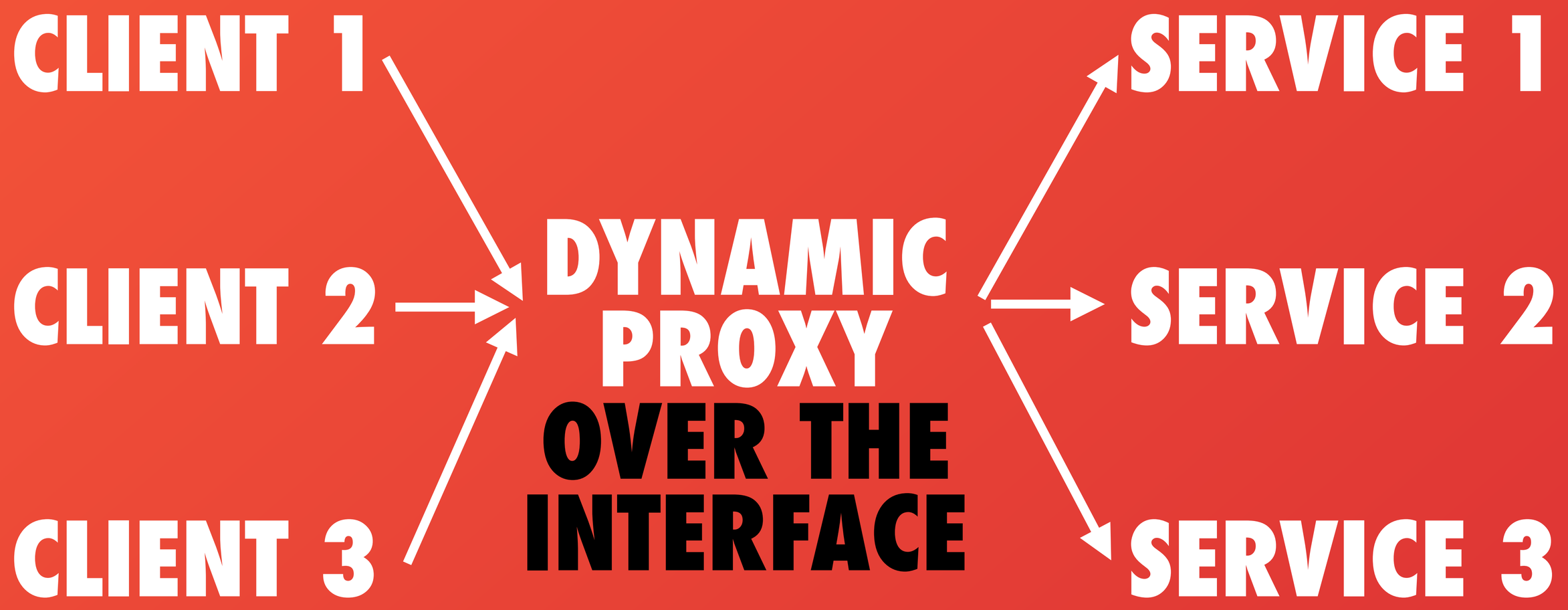
```
ClientInterceptor

  void Process(FunctionCall func) {
    func.Result = MakeHttpCall(func);
  }

GET myservice.local/values?since=5min
```


CLIENT 1
CLIENT 2
CLIENT 3
DYNAMIC PROXY
SERVICE 1
SERVICE 2
SERVICE 3

```
ServerInterceptor

  HttResponse Process(HttpRequest req) {
     return CallMyService(req);
  }

>> IValueService.Get(time);
```

BENEFITS?

```
//share of common interface on both
// sides (client and server) ==
// compiler-time checking

[HttpProperties (Port = 80, SSL = true)]
interface IValueService {

  IEnumerable<string> Get();
}
```


CLIENT 1
CLIENT 2
CLIENT 3
DYNAMIC PROXY
OVER THE INTERFACE
SERVICE 1
SERVICE 2
SERVICE 3

GRPC

```
message HelloRequest {
  string greeting = 1;
}

message HelloResponse {
  string reply = 1;
}

service HelloService {
  rpc SayHello(HelloRequest) returns
  (HelloResponse);
}
```

APACHE THRIFT

```
service Calculator extends
  shared.SharedService {

    i32 add(1:i32 num1, 2:i32 num2)
}
```

HOW TO TEST?

```
IEnumerable<string> GetForLast5Min() {

  return _valuesService.Get(
    DateTime.Now.AddMinutes(-5);
  );
}
```

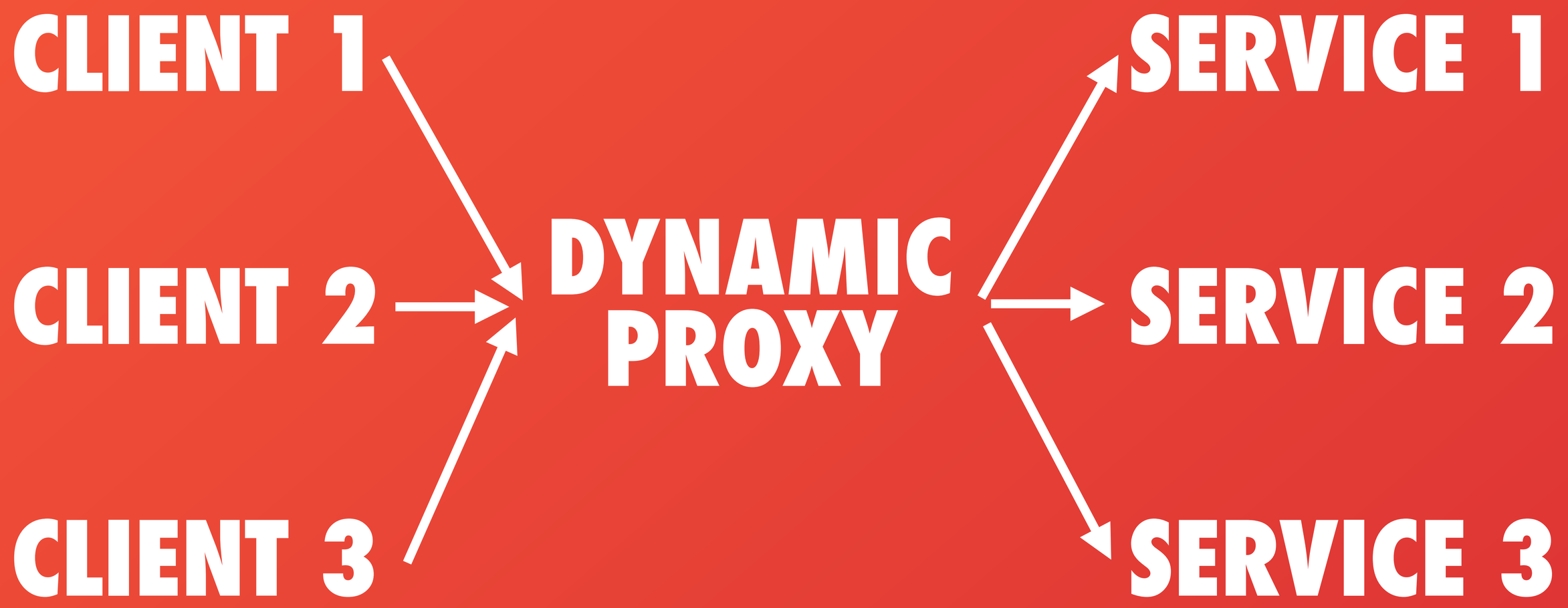
DEPENDENCY INJECTION

MANUAL TESTS

REST IS IN PLACE

0. Simple to use (HTTP) and read (JSON)
1. A lot of tools (wget, curl, postman)

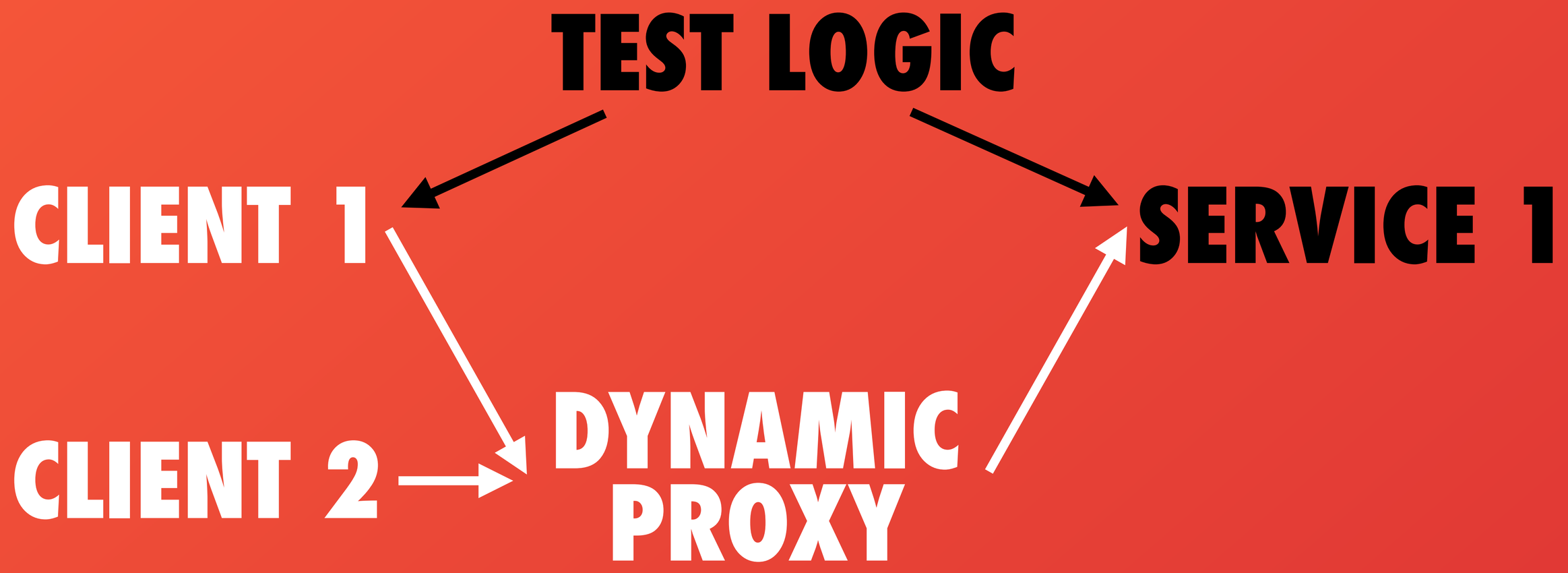
INTEGRATION TESTS


CLIENT 1
CLIENT 2
CLIENT 3
DYNAMIC PROXY
SERVICE 1
SERVICE 2
SERVICE 3

```
DynamicServiceClientInterceptor

  void Process(FunctionCall func) {
    func.Result = MakeHttpCall(func);
  }

GET myservice.local/values?since=5min
```

ACTUALLY FAKE SERVICE

CLIENT 1
CLIENT 2
CLIENT 3
DYNAMIC PROXY
SERVICE 1

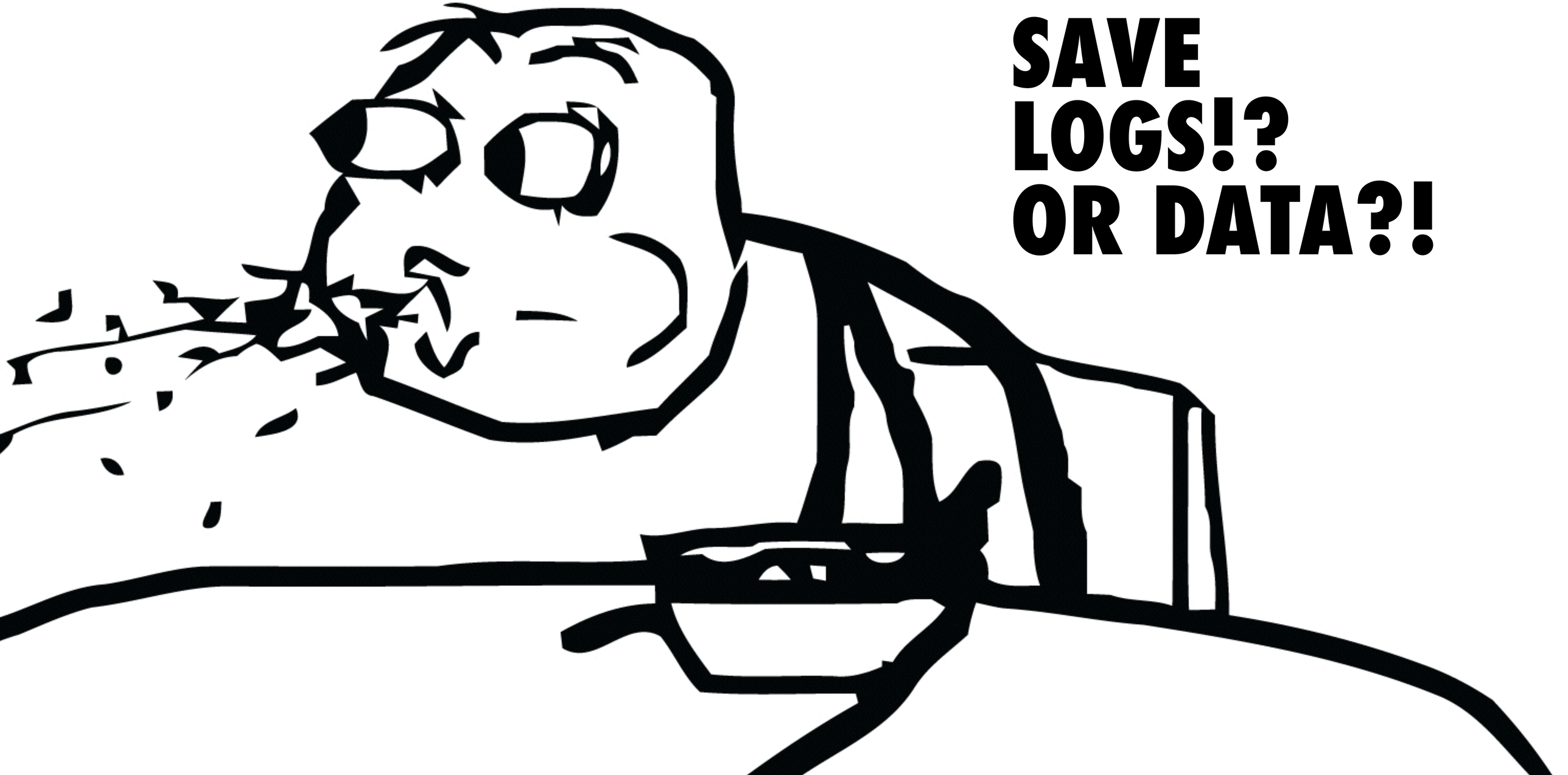

TEST LOGIC
CLIENT 1
SERVICE 1
CLIENT 2
DYNAMIC PROXY

```
DynamicServiceClientInterceptor

  void Process(FunctionCall func) {
    func.Result = _production
    ? MakeHttpCall(func)
    : MakeMockCall(func);
  }
```

LOCAL FAKE CALL

A QUESTION:

WHAT WAS THE DATA YESTERDAY?


SAVE
LOGS!?
OR DATA?!

WEB LOGS

kibana
elastic
YOUR APPS
LOGS

REST: HTTP + JSON

kibana

elastic

REQUESTS

RESPONSES

YOUR APPS

Logstash Search

Oct 10, 2014 17:18:05 to Oct 10, 2014 17:19:07

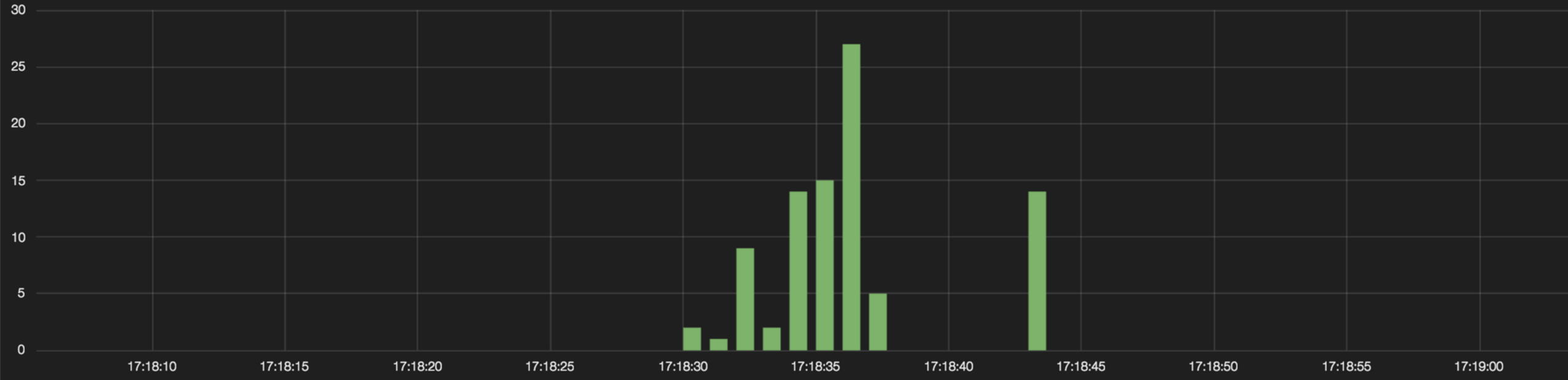
QUERY

*

FILTERING

EVENTS OVER TIME

View | Zoom Out | * (89) count per **1s** | (**89** hits)



ALL EVENTS

0 to **89** of 89 available for paging

Fields

All (24) / Current (15)

Type to filter...

- @timestamp
- @version
- _id

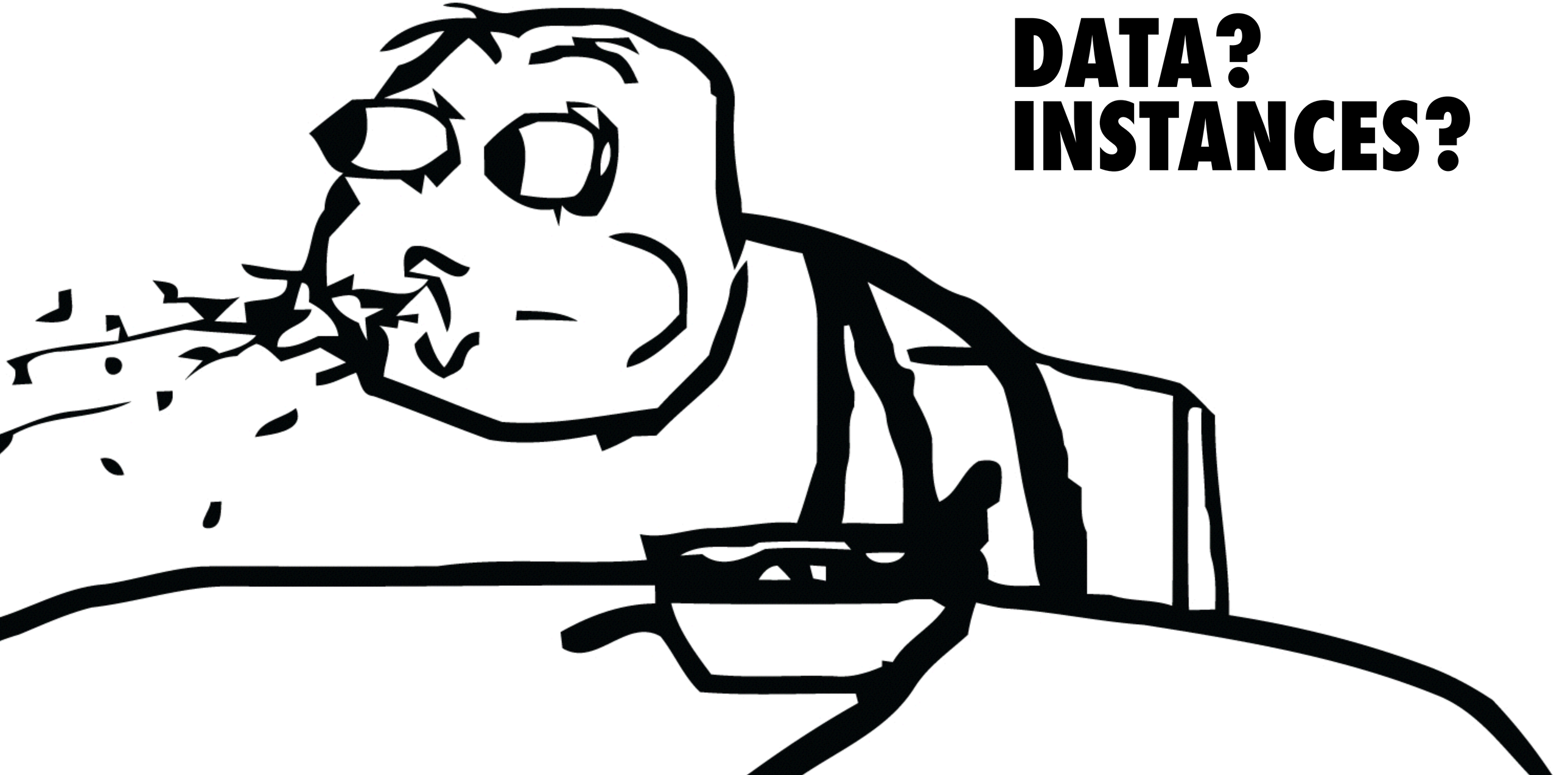
_source (select columns from the list to the left)

{"message":"2014-10-10 17:18:42.512 INFO 11808 [taskExecutor-1] --- o.s.b.c.l.support.SimpleJobLauncher : Job: [FlowJob: [name=flatfileJob]] completed with the following parameters: [{pathToFile=classpath:partner-import.csv}] and t
[COMPLETED]","@version":"1","@timestamp":"...

{"message":"2014-10-10 17:18:42.510 INFO 11808 [taskExecutor-1] --- d.c.batch.listener.ProtocolListener : \n++ \nProtocol for flatfileJob \n Started : Fri Oct 10 1
Finished : Fri Oct 10 17:18:42 CEST 2014\n Exit-...

{"message":"2014-10-10 17:18:42.498 INFO 11808 [taskExecutor-1] --- d.c.batch.item.LogItemProcessor : test","@version":"1","@timestamp":"2014-10-10T15:18:43.141Z","host":"denschu","path":"/tmp/server.log","logtime":"2014-10-1
17:18:42.498","loglevel":"INFO","pid":"11808","threadname":"[ta...

WE
NEED
YOU
SERVE MORE
DATA


DATA?
INSTANCES?

```
ClientInterceptor

  void Process(FunctionCall func) {
    func.Result = MakeHttpCall(func);
  }

GET myservice.local/values?since=5min
```

HARDCODED

HIGH AVAILABILITY

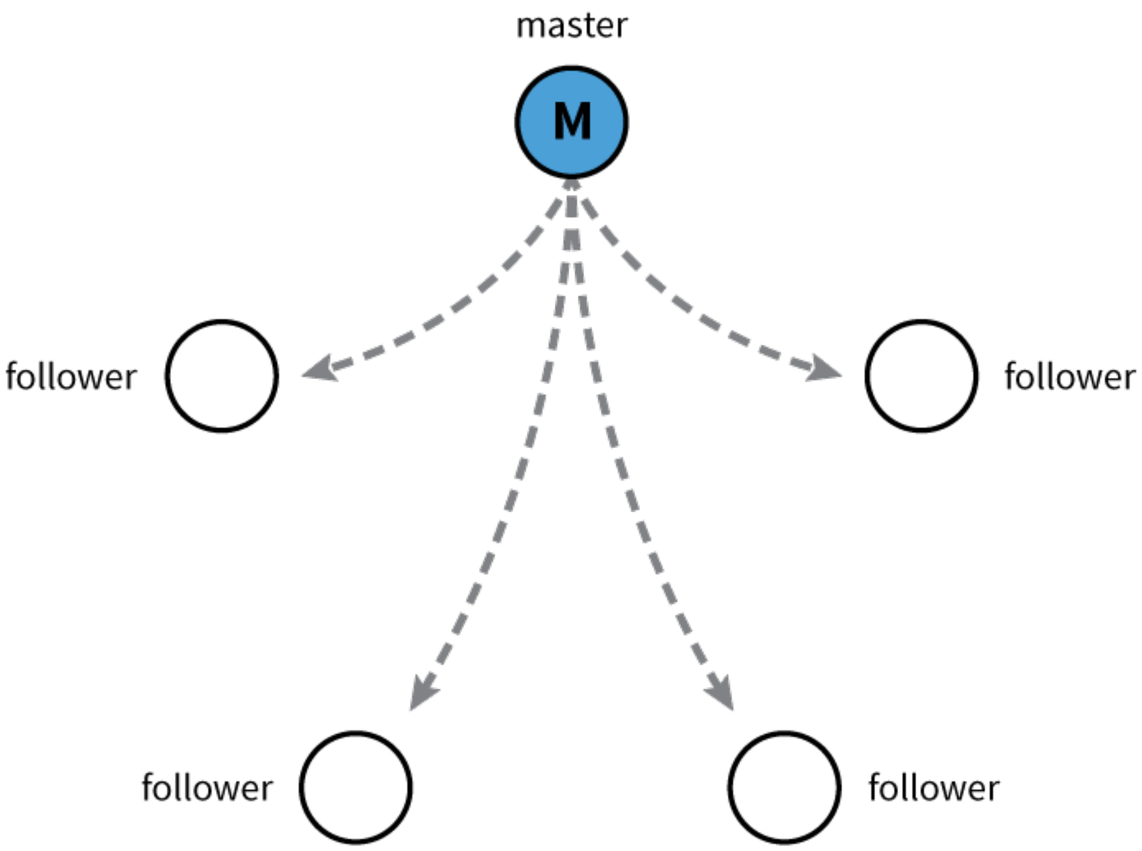
SERVICE DISCOVERY

DISTRIBUTED KEY VALUE STORAGE

WHY ETCD?

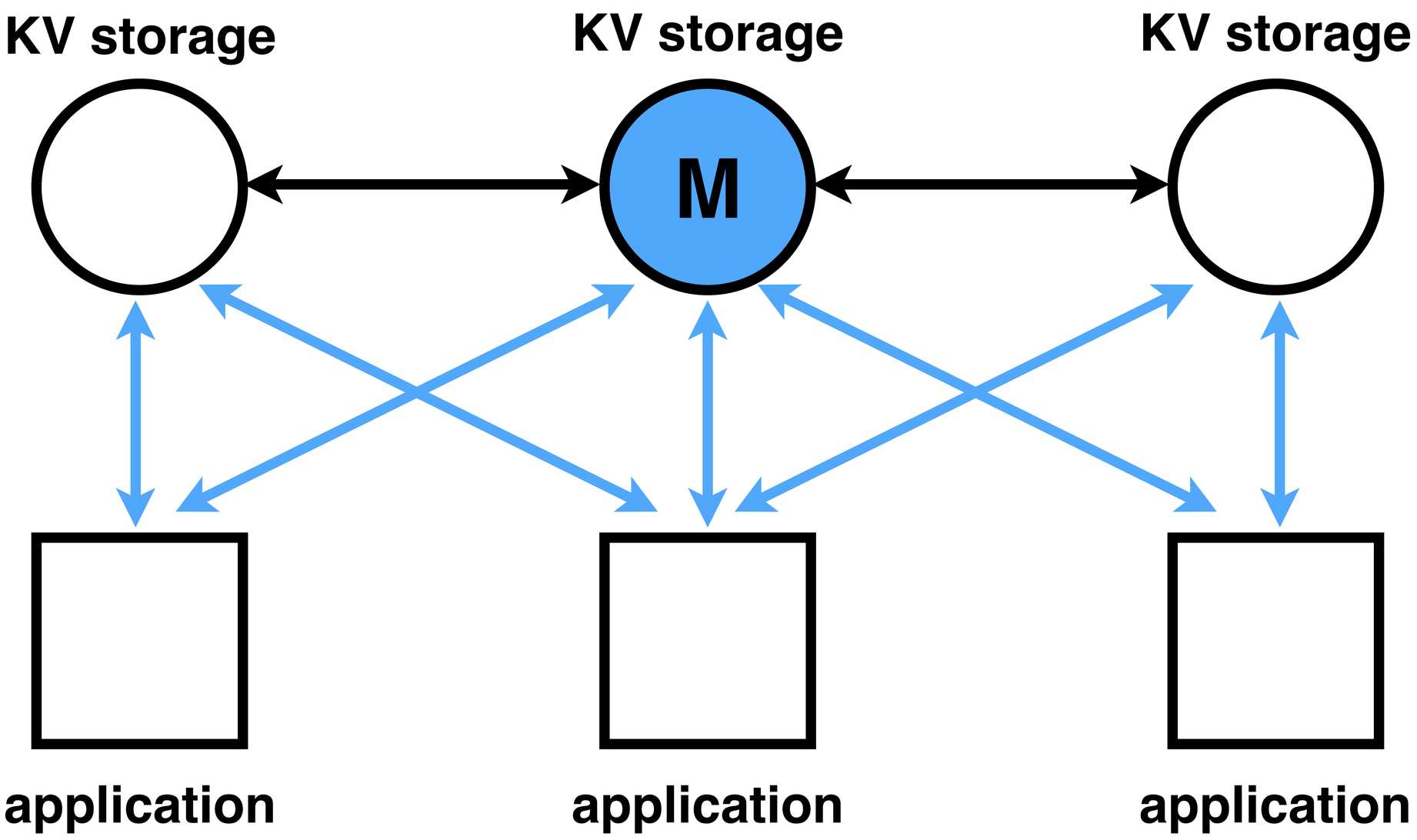
0. Simple to use (REST)
1. Key-Value storage
2. Like a directory tree
3. Sync data to all nodes
4. High available out-of-the-box

etcd



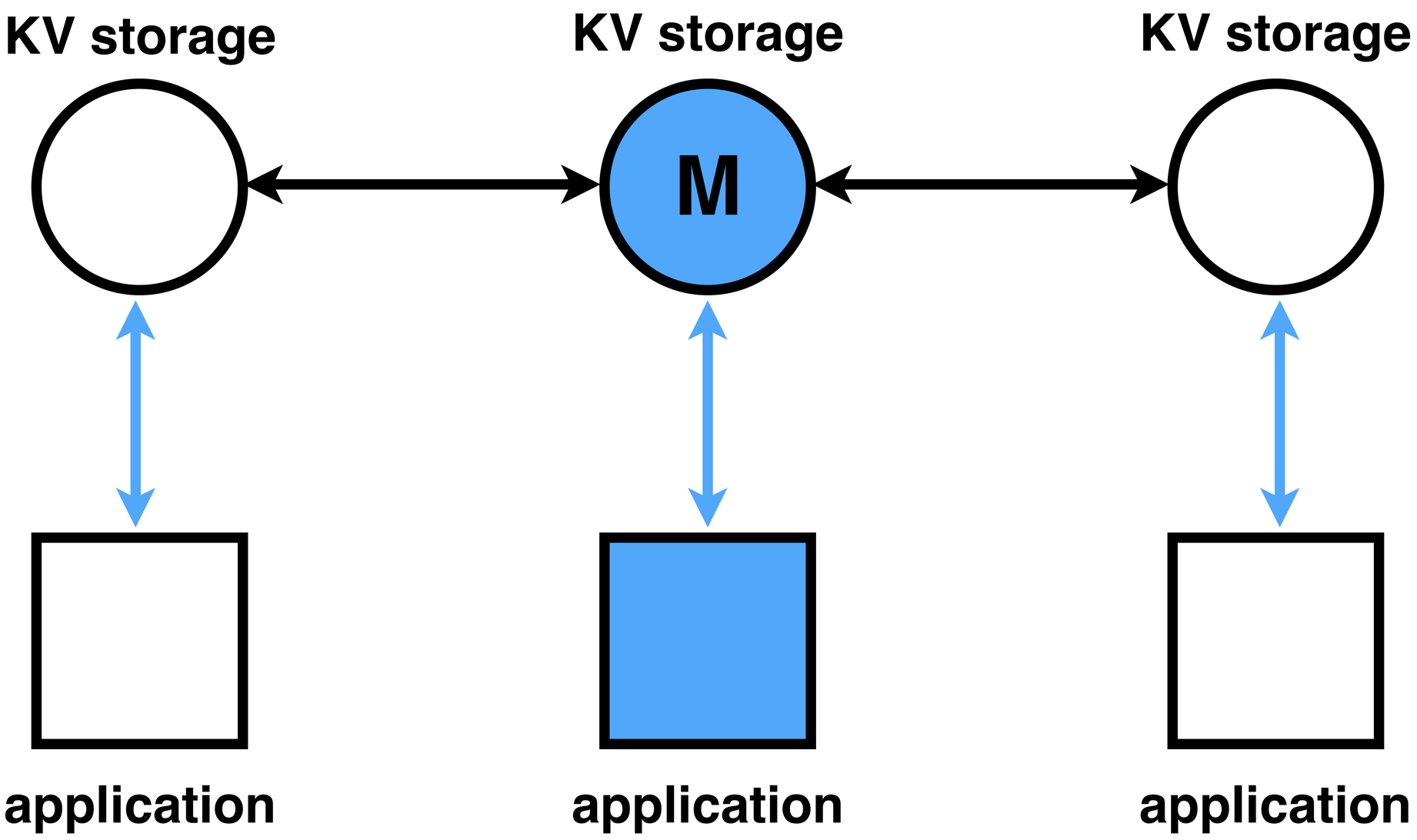
SERVICE REGISTRY

```
string ResolveUrl(string serviceName) {

  var url = _readUrlFor(serviceName);
  return url;
}

GET http://myetcd/v2/keys/valueService
```


KV storage
KV storage
KV storage
M
application
application
application

LONG POOLING AND CACHE LOCAL

```
// wait for update

GET http://myetcd/v2/keys/valueService?
wait=true
```


KV storage
KV storage
KV storage
M
application
application
application

MICROPHONE

https://github.com/rogeralsing/Microphone

Consul, ETCD
Nancy, Web Api

```
class Program
{
    static void Main(string[] args)
    {
        Cluster.Bootstrap(new WebApiProvider(),
new EtcdProvider(), "values", "v1");
        Console.ReadLine();
    }
}

public class ValuesController : ApiController
{
    public string Get()
    {
        return "WebApi Service";
    }
}
```

CLIENT-SIDE SERVICE DISCOVERY

```
// automatically load balanced over service
instances
var instance = await
Cluster.FindServiceInstanceAsync("orders");

// use Rest# or similar to call into the remote
service
MakeSomeCall(“/api/orders",
 instance.ServiceAddress, instance.ServicePort);
```

SERVER-SIDE SERVICE DISCOVERY

with Load Balancer:

- HAProxy
- NGINX

IMPROVEMENTS?

BE
REACTIVE

DO NOT CALL. SEND MESSAGES

kafka

RabbitMQ™

SQS

Kinesis

Queues

Service Bus

DRAWBACKS

DO NOT KEEP
IT MICRO.
IT IS NOT ABOUT
LINES OF CODE

COMPLEXITY
OVERALL
AS A DISTRIBUTED SYSTEM

DON’T
TRUST
NETWORK

POLLY

```
Policy
.Handle<DivideByZeroException>()
.CircuitBreaker(2, TimeSpan.FromMinutes(1))
```

CHANGES
ROLLOUT
IS COMPLEX

TeamCity

Octopus Deploy

MONITOR
EVERYTHING

elastic

InfluxDB

kibana

Grafana

DISTRIBUTED CONSISTENCY

TAKEAWAYS

- Use it if
  - Continuos Delivery
  - Fast Changes
  - Distributed Team

- Be ready for
  - Complexity

THANKS!

Q&A

VITALY BAUM

@butaji
vitaly.baum@gmail.com