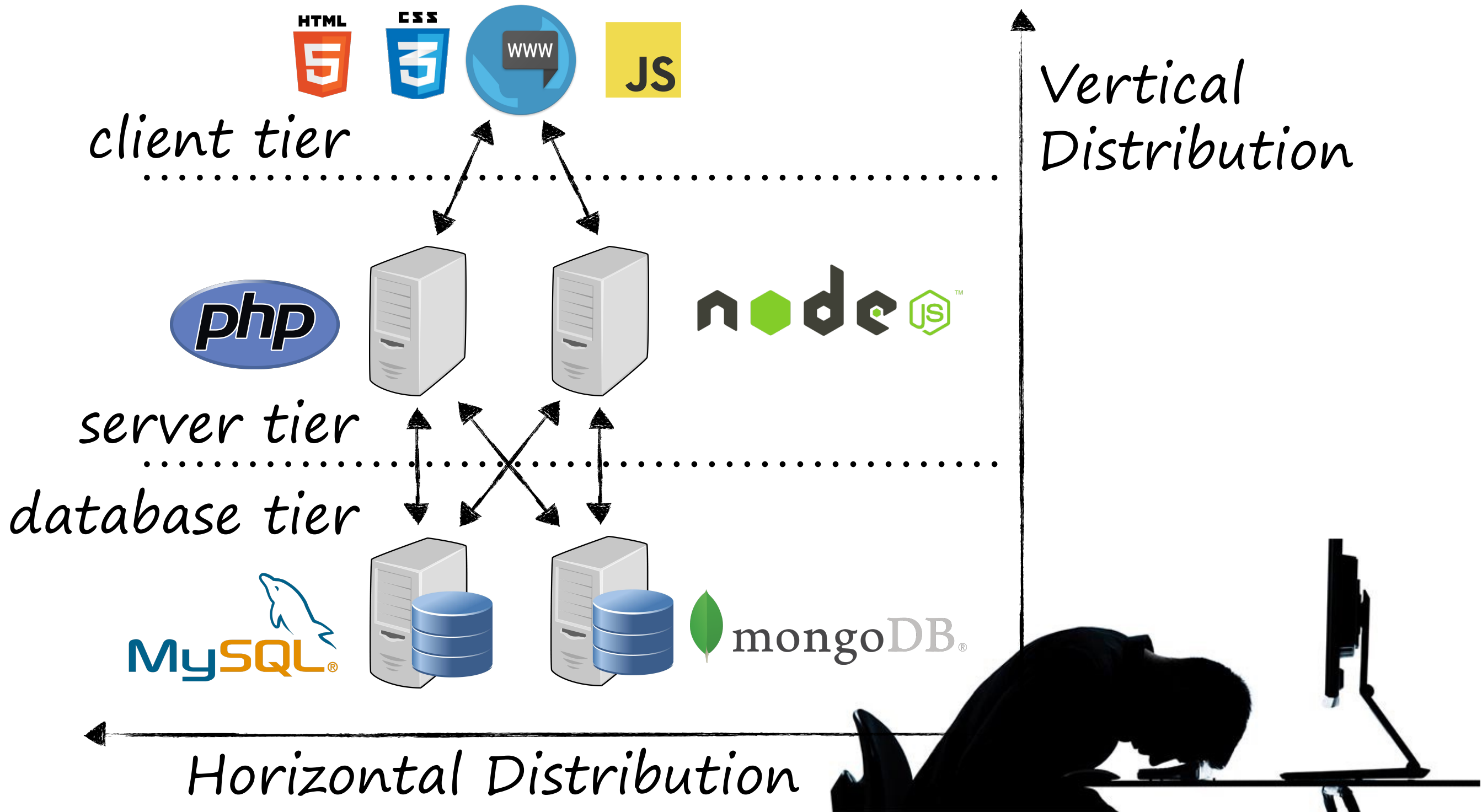


Tierless Debugging

Laurent Christophe
VUB SOFT 11-03-2017

Tier Programming



Tierless Programming



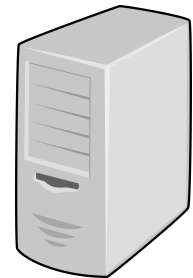
@tierA
@tierB
@tierC

```
for i in people.data.users:  
    response = client.api.statuses.user_timeline.get(screen_name=i.scre  
    print 'Got', len(response.data), 'tweets from', i.screen_name  
    if len(response.data) != 0:  
        ldate = response.data[0]['created_at']  
        ldate2 = datetime.strptime(ldate, '%a %b %d %H:%M:%S +0000 %Y'  
        today = datetime.now()  
        howLong = (today-ldate2).days  
        if howLong < daywindow:  
            print i.screen_name, 'has tweeted in the past', daywindow,  
            totaltweets += len(response.data)  
            for j in response.data:  
                if j.entities.urls:  
                    for k in j.entities.urls:  
                        newurl = k['expanded_url']  
                        urlset.add((newurl, j.user.screen_name))  
        else:  
            print i.screen_name, 'has not tweeted in the past', daywind
```

Static
Tier
Splitter



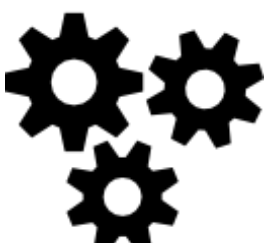
tierA



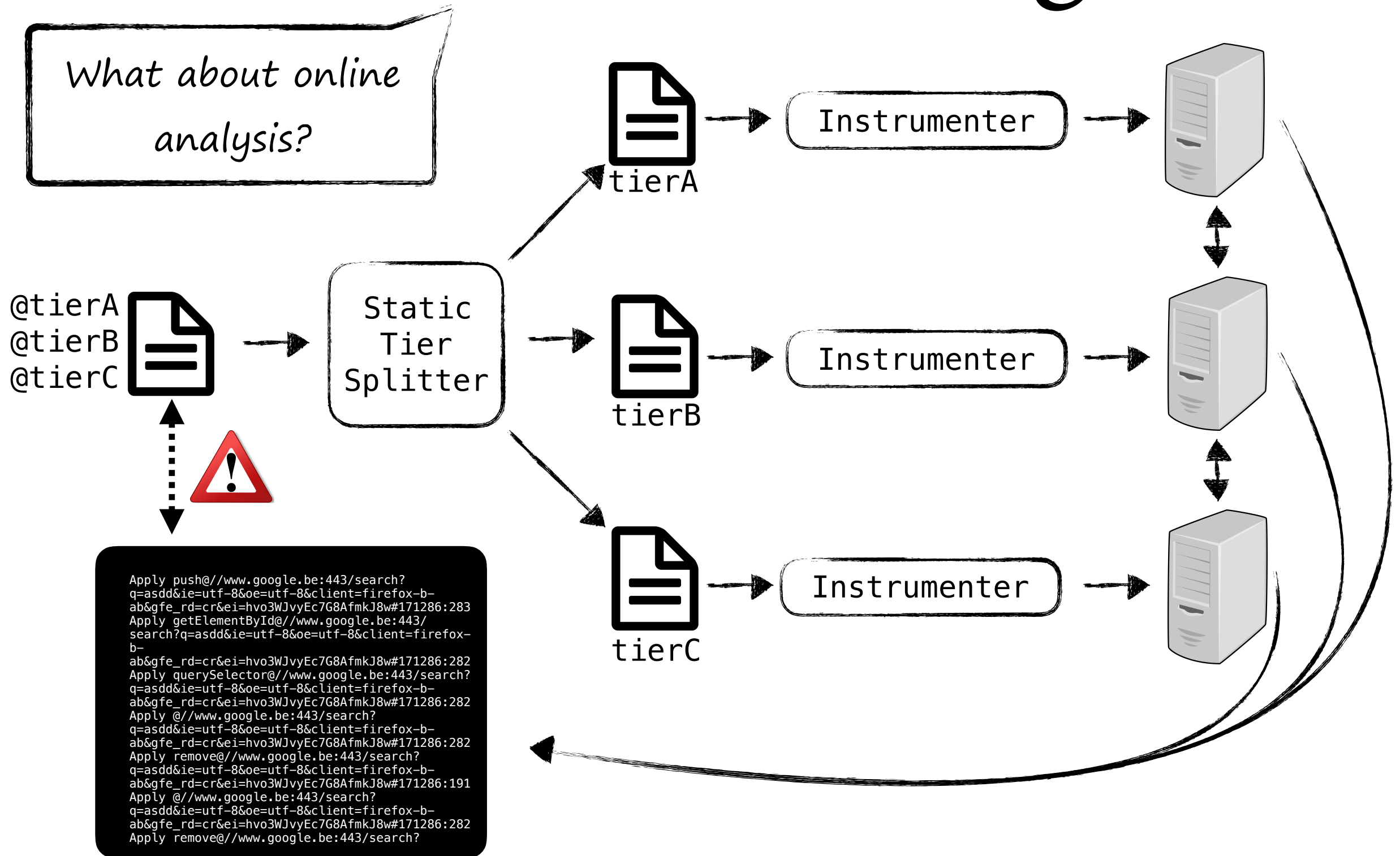
tierB



tierC



Tierless Tracing



Remote Analysis

Melf

fantasio.js

```
1 var Melf = require("melf/browser");
2 var melf = Melf({
3   channel: "melf-channel",
4   alias: "fantasio"
5 });
6 melf.sync.register()
7 melf.sync.register("go", function (origin, data) {
8   console.log(melf.sync.trigger(origin, "where", "key"));
9   console.log(melf.sync.trigger(origin, "where", "car"));
10  return origin+", I'm ready!";
11 });
```



```
fantasio, the key is right there!
fantasio, the car is right there!
I'm ready!
```

```
["spirou",0,"go","let's go!"]
[0,"fantasio, the key is right there!"]
[1,"fantasio, the car is right there!"]
```



spirou.js

```
1 var Melf = require("melf");
2 var melf = Melf({
3   boxdir: __dirname+"/boxdir",
4   alias: "spirou"
5 });
6 melf.sync.register("where", function (origin, data) {
7   return origin+", the "+data+" is right there!";
8 });
9 console.log(melf.sync.trigger("fantasio", "go", "let's go!"));
```

```
I'm ready!
```

```
["fantasio",0,"where","key"]
["fantasio",1,"where","car"]
[0,"spirou, I'm ready!"]
```

Khala

fantasio.js

```
1 var Melf = require("melf");
2 var Kalah = require("kalah");
3 var melf = Melf({
4   boxdir: __dirname+"/boxdir",
5   alias: "fantasio"
6 });
7 var kalah = Kalah(melf);
8 var marsupilami = {
9   toys: [],
10  play: function () {
11    console.log("marsupilami is playing with: "+this.toys);
12  }
13 };
14 console.log(melf.sync.trigger("spirou", "marsupilami", kalah.export(marsupilami)));
15 melf.close();
```

```
marsupilami is playing with: car,plane
that was funky
```

```
["fantasio",0,"marsupilami",{"o":"fantasio/1"}]
[0,{"o":"fantasio/2"}]
[1,{"f":"fantasio/3"}]
[2,1]
[3,{"o":"fantasio/2"}]
[4,{"f":"fantasio/3"}]
[5,2]
[6,{"f":"fantasio/4"}]
[7,{"u":1}]
```

spirou.js

```
1 var Melf = require("melf");
2 var Kalah = require("kalah");
3 var melf = Melf({
4   boxdir: __dirname+"/boxdir",
5   alias: "spirou"
6 });
7 var kalah = Kalah(melf);
8 melf.sync.register("marsupilami", function (origin, data) {
9   var marsupilami = kalah.import(data);
10  marsupilami.toys.push("car");
11  marsupilami.toys.push("plane");
12  marsupilami.play();
13  return "that was funky!";
14 });
15 melf.close();
```

```
["spirou",0,"kalah-get",[1,"toys",{"o":"fantasio/1"}]]
["spirou",1,"kalah-get",[2,"push",{"o":"fantasio/2"}]]
["spirou",2,"kalah-apply",[3,{"o":"fantasio/2"},["car"]]]
["spirou",3,"kalah-get",[1,"toys",{"o":"fantasio/1"}]]
["spirou",4,"kalah-get",[2,"push",{"o":"fantasio/2"}]]
["spirou",5,"kalah-apply",[3,{"o":"fantasio/2"},["plane"]]]
["spirou",6,"kalah-get",[1,"play",{"o":"fantasio/1"}]]
["spirou",7,"kalah-apply",[4,{"o":"fantasio/1"},[]]]
[0,"that was funky!"]
```


Kalah is Type Aware

```
2 var descriptor = {
3   configurable: "boolean",
4   enumerable: "boolean",
5   writable: "boolean",
6   any: "any",
7   get: "any",
8   set: "any"
9 };
10
11 exports.arguments = {};
12 exports.arguments.getPrototypeOf = ["target"];
13 exports.arguments.setPrototypeOf = ["target", ""];
14 exports.arguments.isExtensible = ["target"];
15 exports.arguments.preventExtensions = ["target"];
16 exports.arguments.getOwnPropertyDescriptor = ["target", ""];
17 exports.arguments.defineProperty = ["target", ""];
18 exports.arguments.has = ["target", "string"];
19 exports.arguments.get = ["target", "string", "any"];
20 exports.arguments.set = ["target", "string", "any", "any"];
21 exports.arguments.deleteProperty = ["target", "string"];
22 exports.arguments.ownKeys = ["target"];
23 exports.arguments.apply = ["target", "any", ["any"]];
24 exports.arguments.construct = ["target", ["any"]];
25
26 exports.result = {};
27 exports.result.getPrototypeOf = "any";
28 exports.result.isExtensible = "boolean";
29 exports.result.getOwnPropertyDescriptor = descriptor;
30 exports.result.has = "boolean";
31 exports.result.get = "any";
32 exports.result.deleteProperty = "boolean";
33 exports.result.ownKeys = ["string"];
34 exports.result.apply = "any";
35 exports.result.construct = "any";
```

alice.js

```
1 var o = {foo:123};
2 melf.sync.trigger("object", "bob", kalah.export(o));
```

```
["alice",0,"object", {"o":"alice/0"}]
["bob",0,"kalah-get", [{"o":"alice/0"}, {"o":"bob/0"}, {"o":"alice/0"}]
["alice", 1, "kalah-get", [{"o":"bob/0"}, "toString", {"o":"bob/0"}]]
[1, {"f":"bob/1"}]
["alice", 2, "kalah-apply", [{"f":"bob/1"}, {"o":"alice/1"}]]
[2, "foo"]
[0, 123]
[0, 123]
```

```
["alice",0,"object", {"o":"alice/0"}]
["bob",0,"kalah-get", [0, "foo", {"o":"alice/0"}]
[0, 123]
[0, 123]
```

```
1 melf.sync.register("object", function (origin, data) {
2   var o = kalah.import(data);
3   return o[{"toString": function () { return "foo" }}];
4 });
```

bob.js

Remote Aran

```
1 var Aran = require("aran");
2 var Melf = require("melf");
3 var Master = require("remote-aran/master");
4
5 var namespace = "_aran_";
6
7 var traps = {};
8 traps.array = function (arr) {
9   var a = [];
10  for (var i=0; i<arr.length; i++)
11    a[i] = arr[i];
12  return a;
13 };
14 traps.apply = function (fct, ths, args, idx) {
15   var line = aran.node(idx).loc.start.line
16   console.log(fct.name+"@"+line);
17   return fct.apply(ths, args);
18 };
19
20 var aran = Aran({
21   namespace: namespace,
22   traps: Object.keys(traps),
23   loc: true
24 });
25
26 var melf = Melf({
27   boxdir: __dirname+"/boxdir",
28   alias: "master"
29 });
30
31 Master(melf, aran, traps);
```

```
push@2
push@3
join@4
bound log@4
```

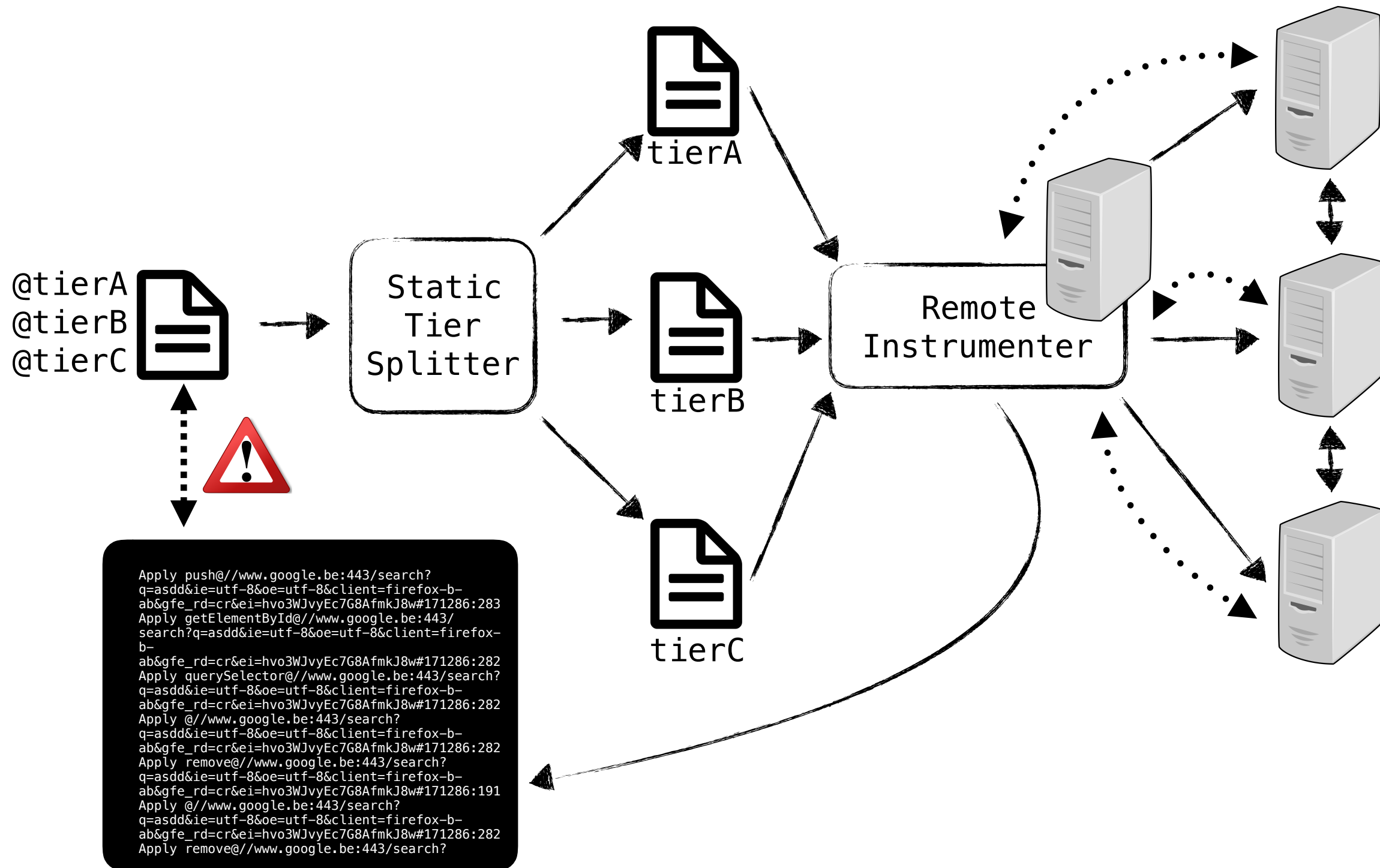
```
1 var marsupilami = {toys:[]};
2 marsupilami.toys.push("car");
3 marsupilami.toys.push("plan");
4 console.log("marsupilami is playing with "+marsupilami.toys.join(", "));
```

```
marsupilami is playing with car, plan
```

```
[0,"_aran__.__global__... "]
["master",0,"kalah-get",[1,"length",{o:"spirou/1"}]]
[1,{o:"master/1"}]
[2,{f:"master/2"}]
[3,1]
[4,{f:"master/2"}]
[5,2]
[6,{f:"master/3"}]
[7,"car, plan"]
["master",1,"kalah-get",[2,"name",{f:"spirou/2"}]]
["master",2,"kalah-get",[2,"apply",{f:"spirou/2"}]]
["master",3,"kalah-apply",[4,{f:"spirou/2"},[...]]]
[9,1]
[10,"marsupilami is playing with car, plan"]
[8,{"u":1}]
```

```
["spirou",0,"aran-instrument",["\nvar marsupilami = {toys:[]}; ...", null]]
["spirou",1,"aran-array",[{o:"spirou/1"},4]]
[0,0]
["spirou",2,"kalah-get",[1,"push",{o:"master/1"}]]
["spirou",3,"aran-apply",[{f:"master/2"},{o:"master/1"},["car"],6]]
["spirou",4,"kalah-get",[1,"push",{o:"master/1"}]]
["spirou",5,"aran-apply",[{f:"master/2"},{o:"master/1"},["plan"],11]]
["spirou",6,"kalah-get",[1,"join",{o:"master/1"}]]
["spirou",7,"aran-apply",[{f:"master/3"},{o:"master/1"},["", "],20]]
["spirou",8,"aran-apply",[{f:"spirou/2"},{o:"spirou/3"},["..."],16]]
[1,"bound log"]
[2,{f:"spirou/4"}]
["spirou",9,"kalah-get",[4,"length",{o:"master/4"}]]
["spirou",10,"kalah-get",[4,"0",{o:"master/4"}]]
[3,{"u":1}]
```

Tierless Online Analysis



Alternatives:

- 1) Full CPS both tierless and analysis code
- 2) Fiber npmjs.com/package/fibers