

Decentralized Stream Processing over Web-enabled devices

Masiar Babazadeh **Andrea Gallidabino** **Cesare Pautasso**

@masiarb

@agallidabino

@pautasso

name.surname@usi.ch

University of Lugano (USI)
Faculty of Informatics
Switzerland

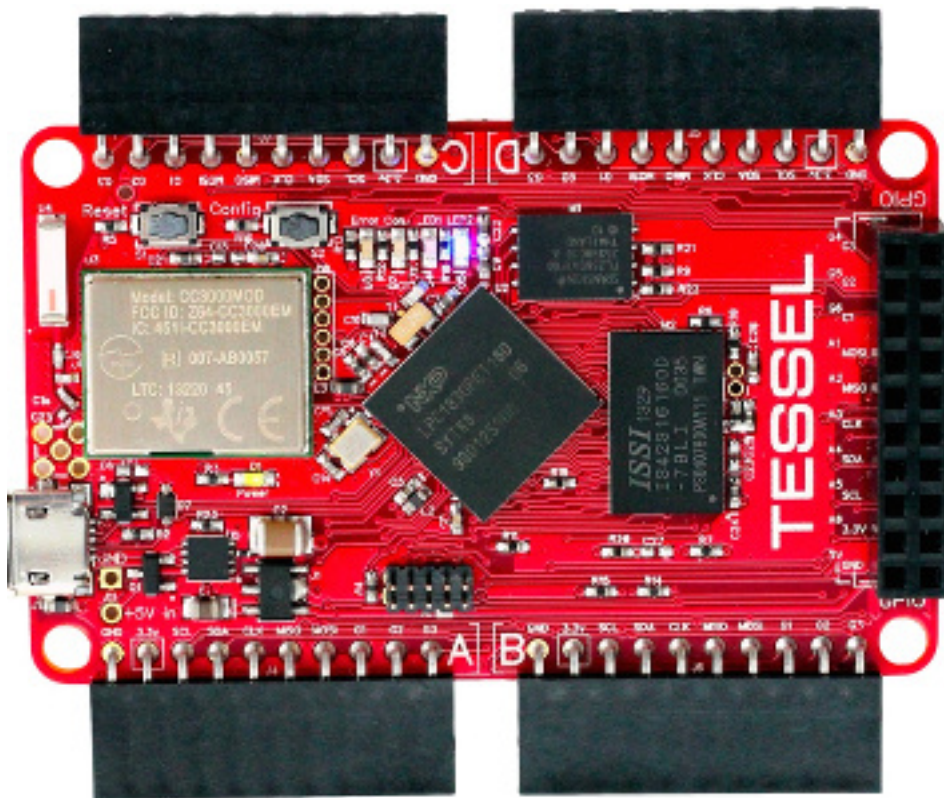




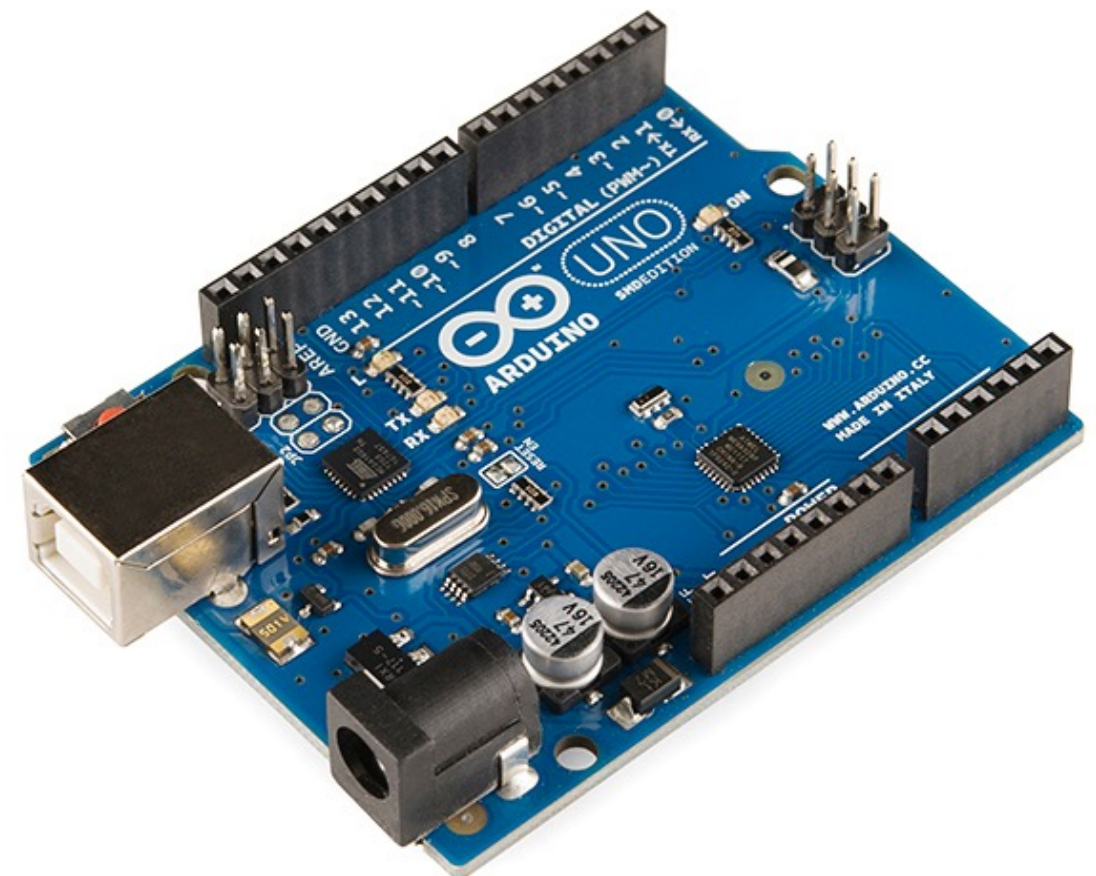
Smartphones



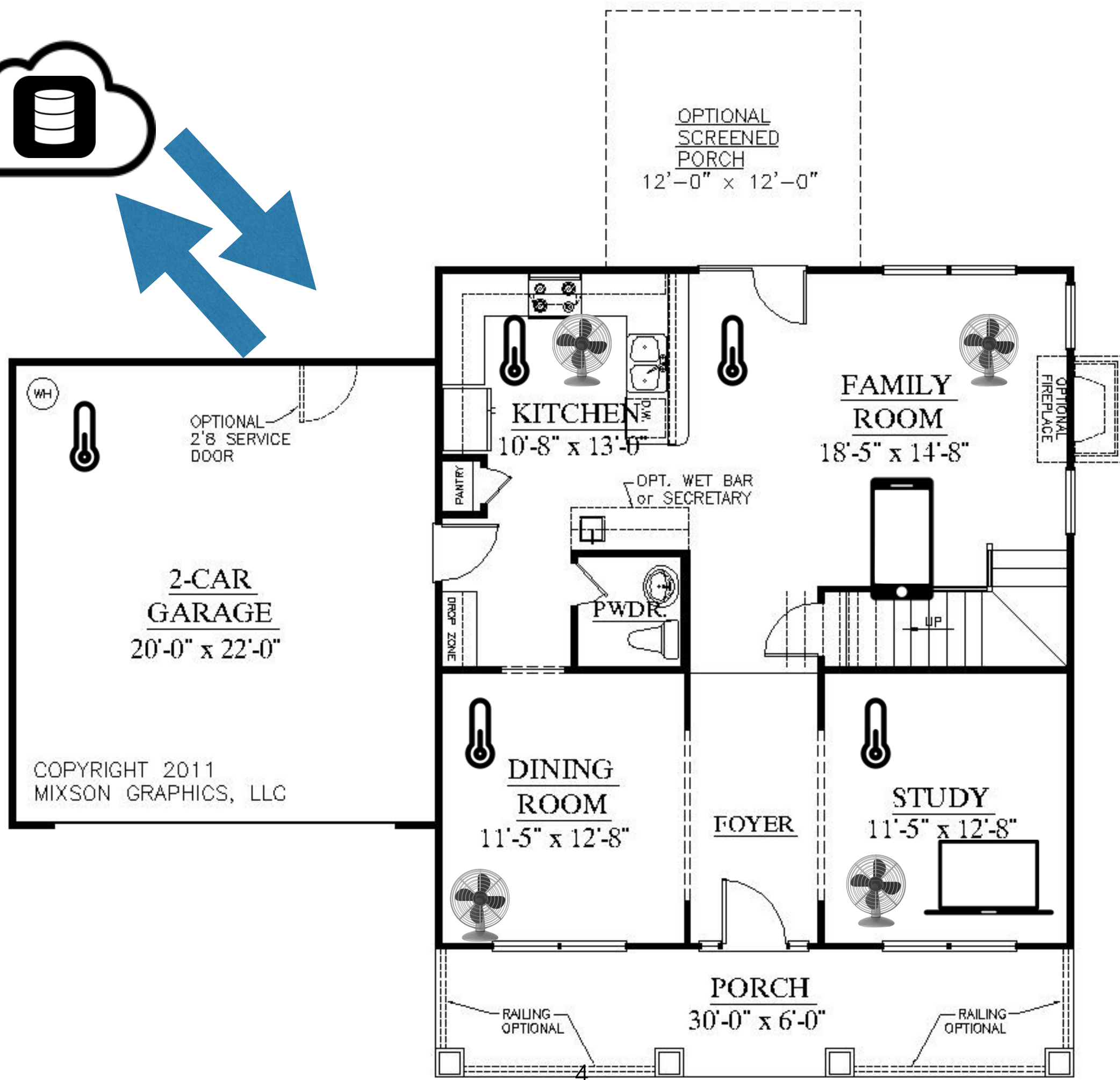
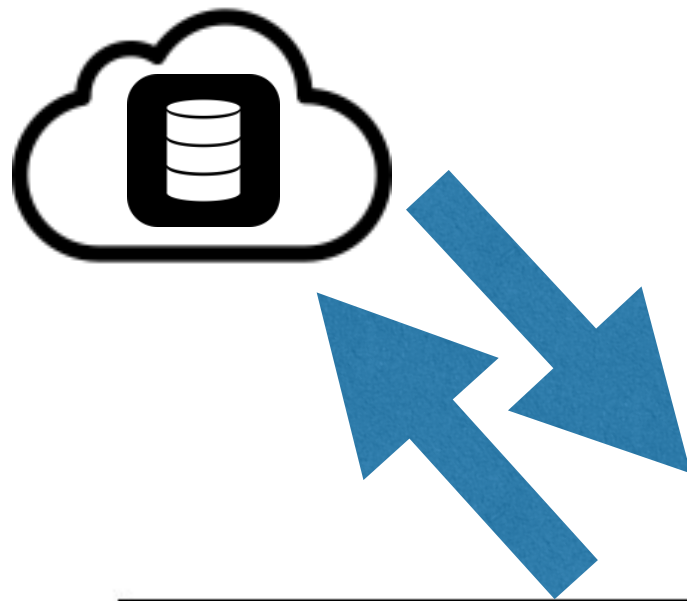
Raspberry Pi

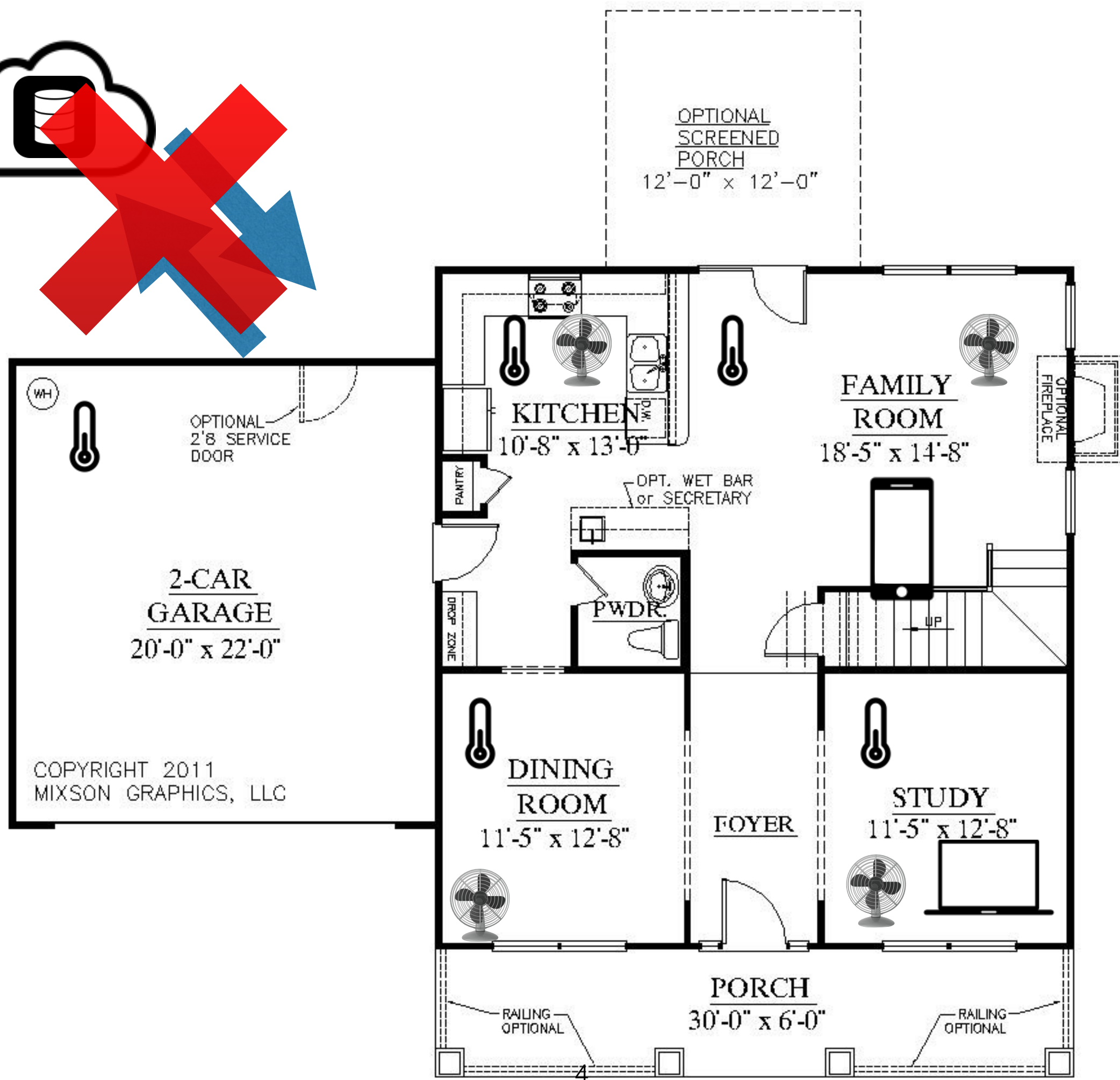
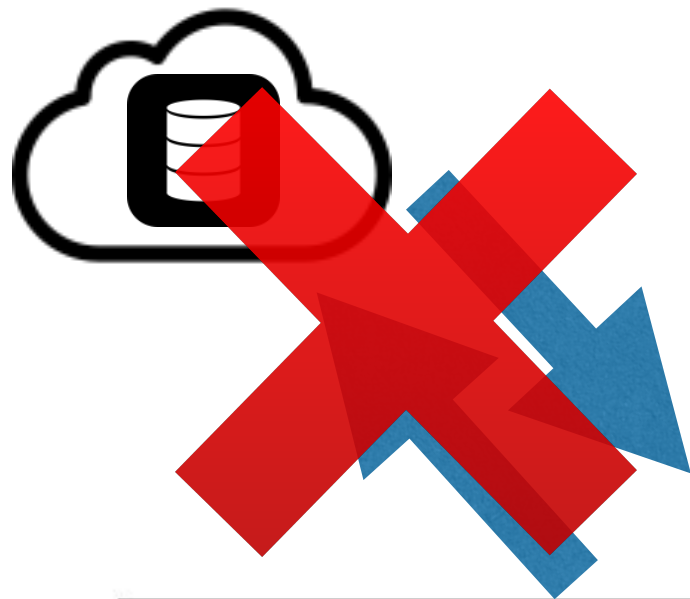


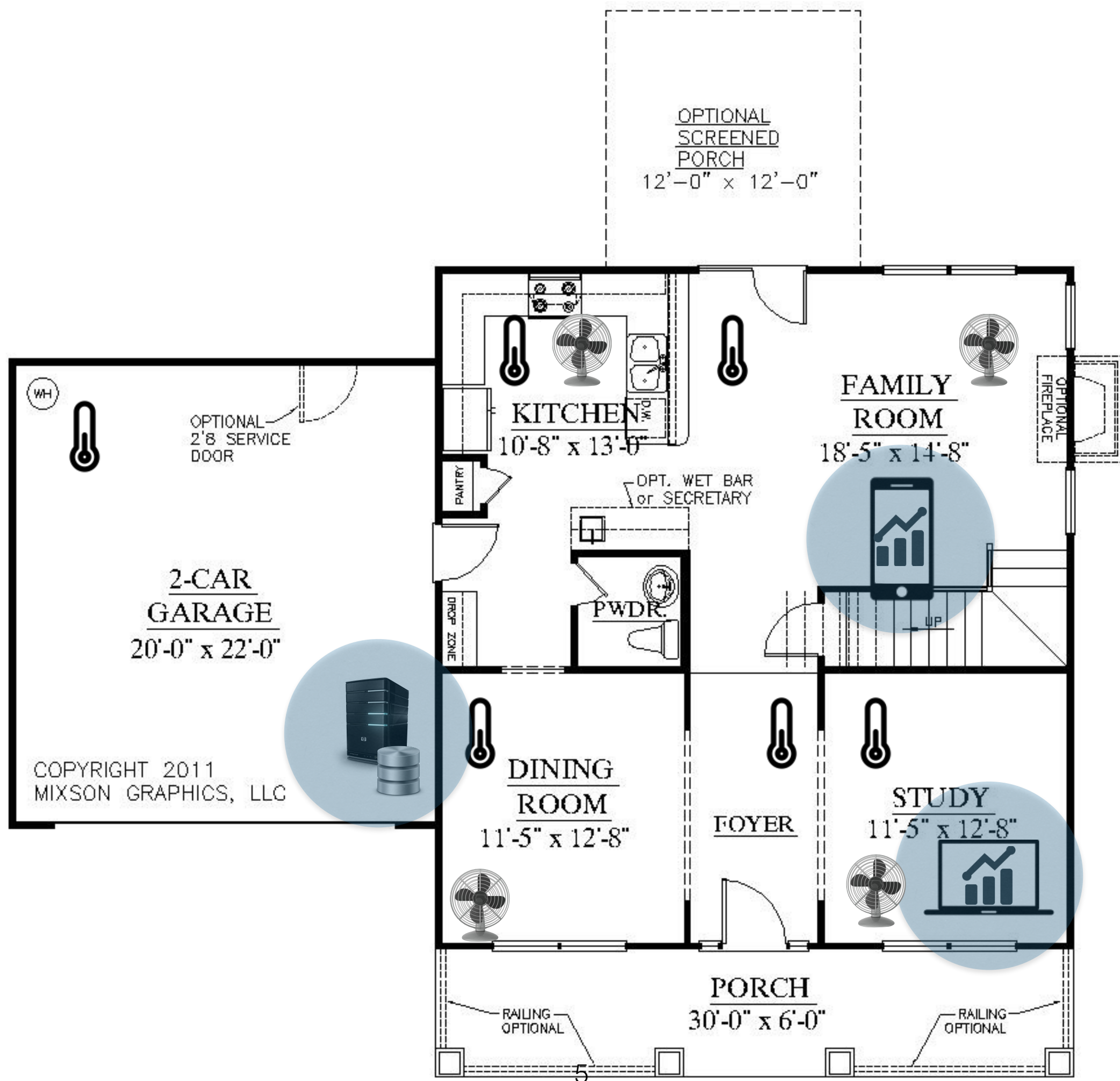
Tessel

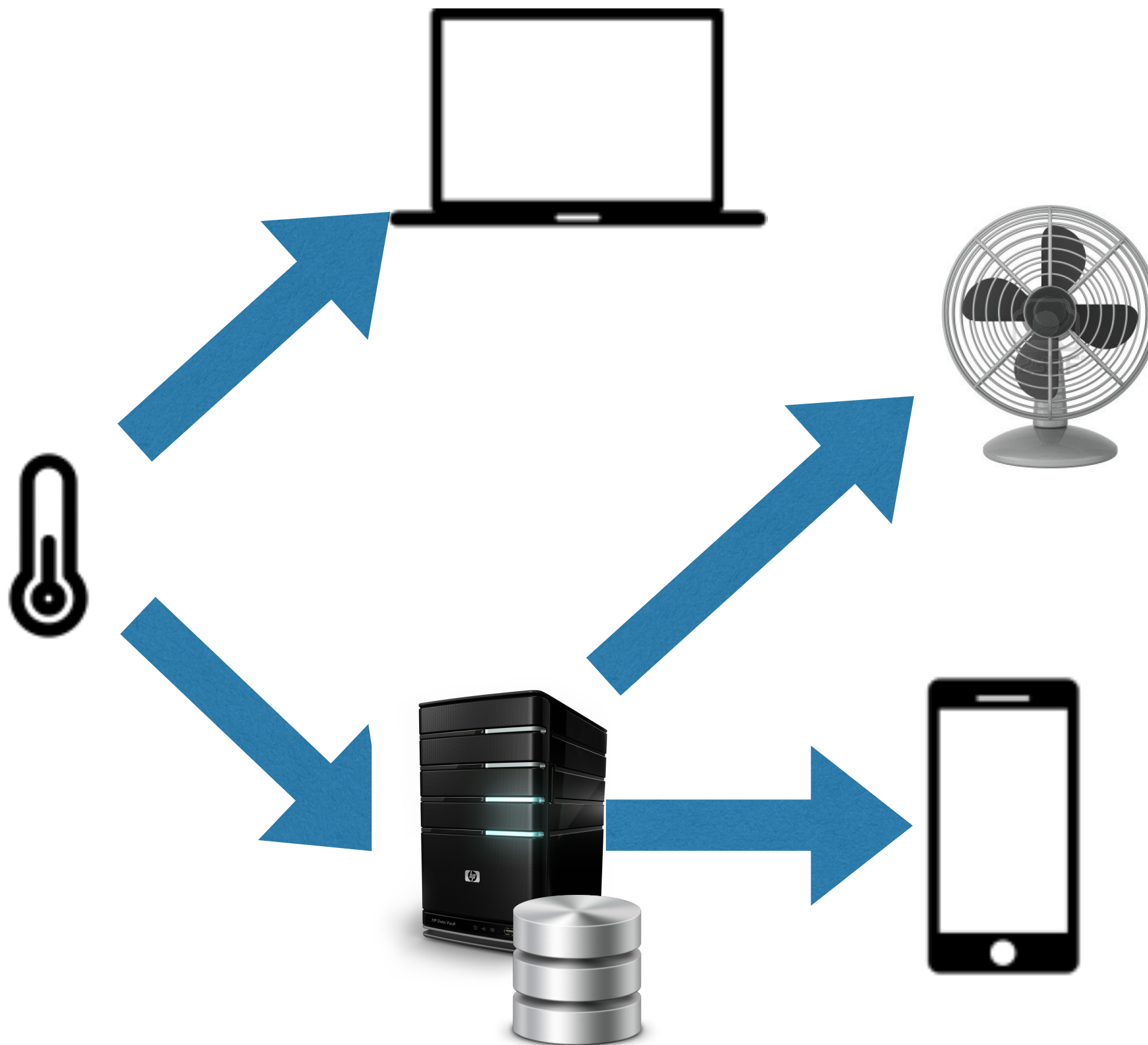


Arduino





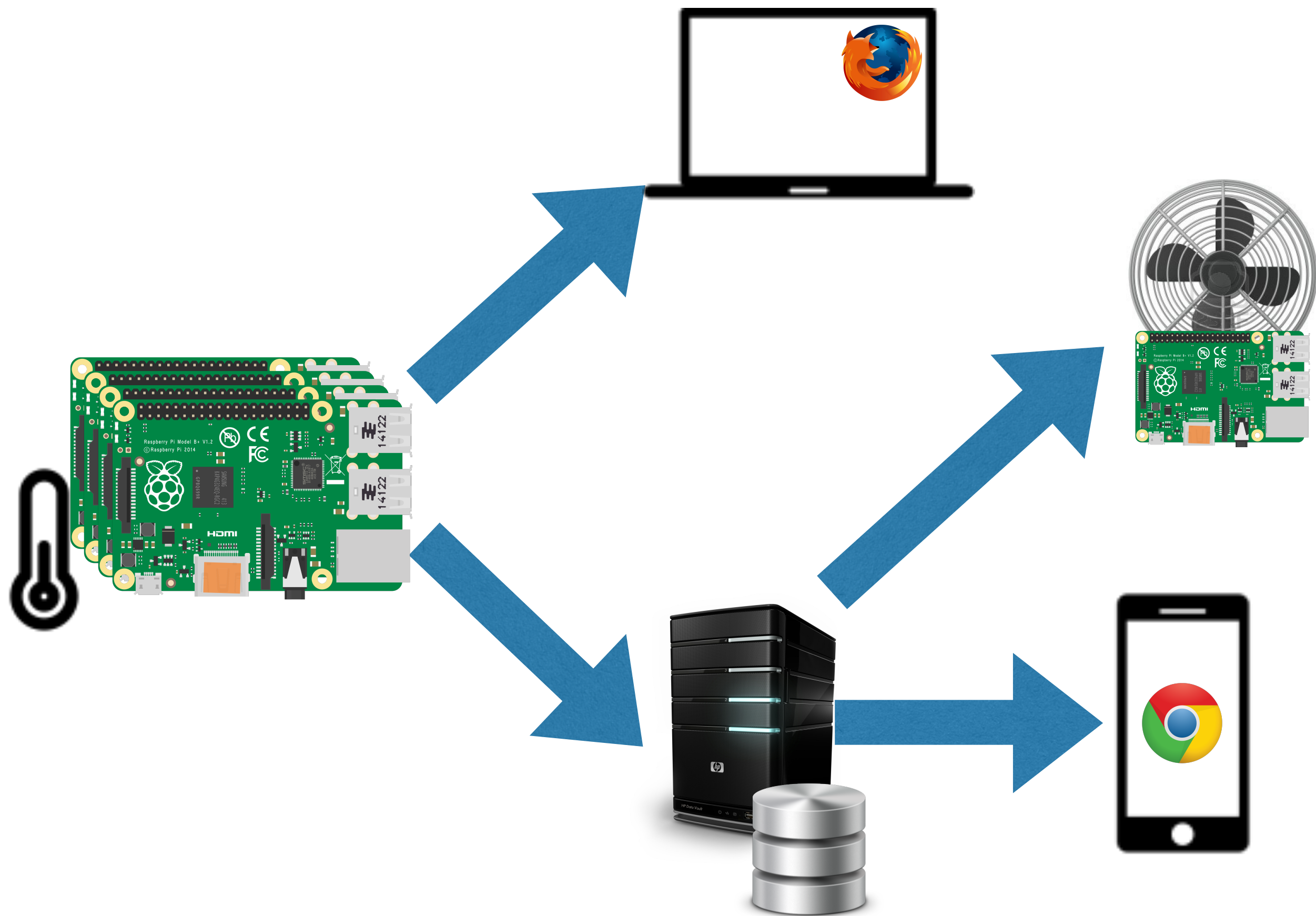




Stream Processing

Heterogeneous Hardware

Dynamic Environment



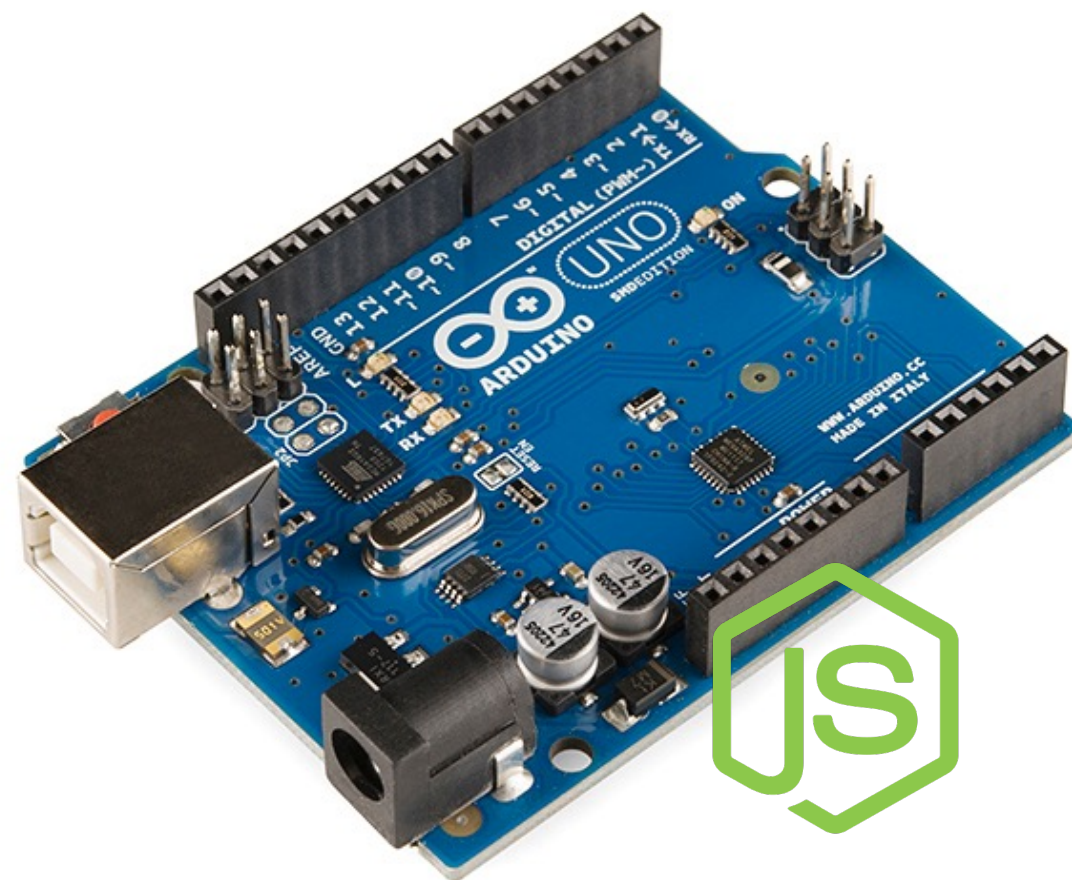
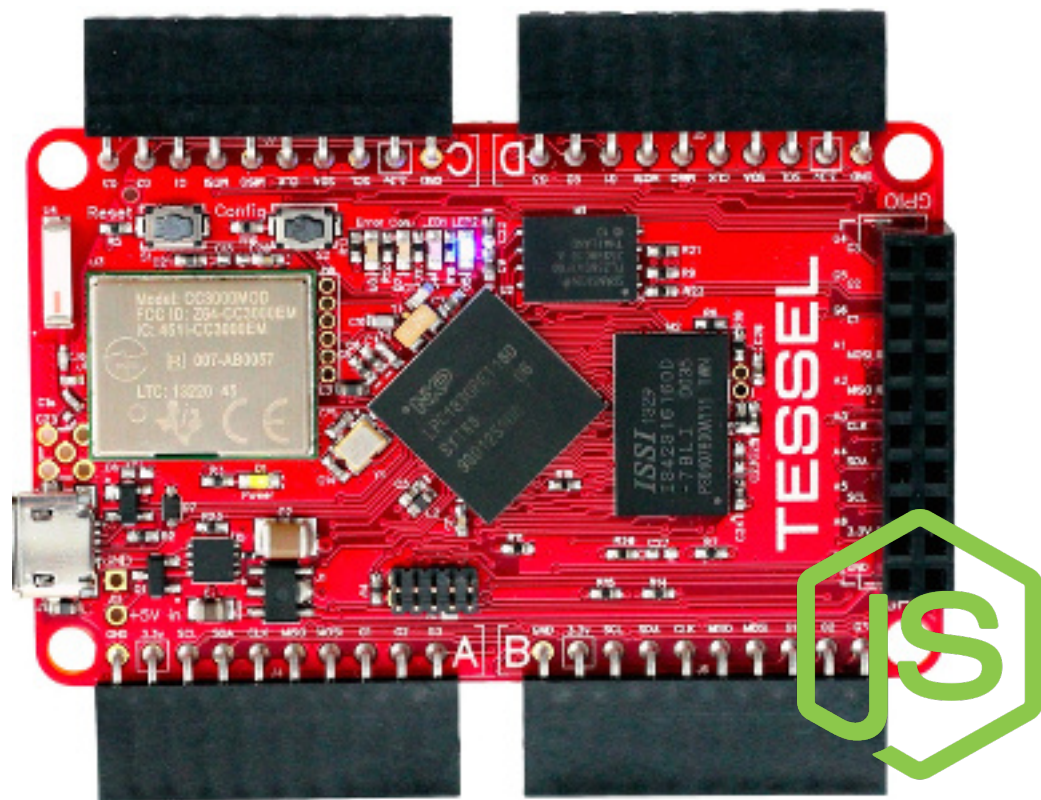
Web  RTC



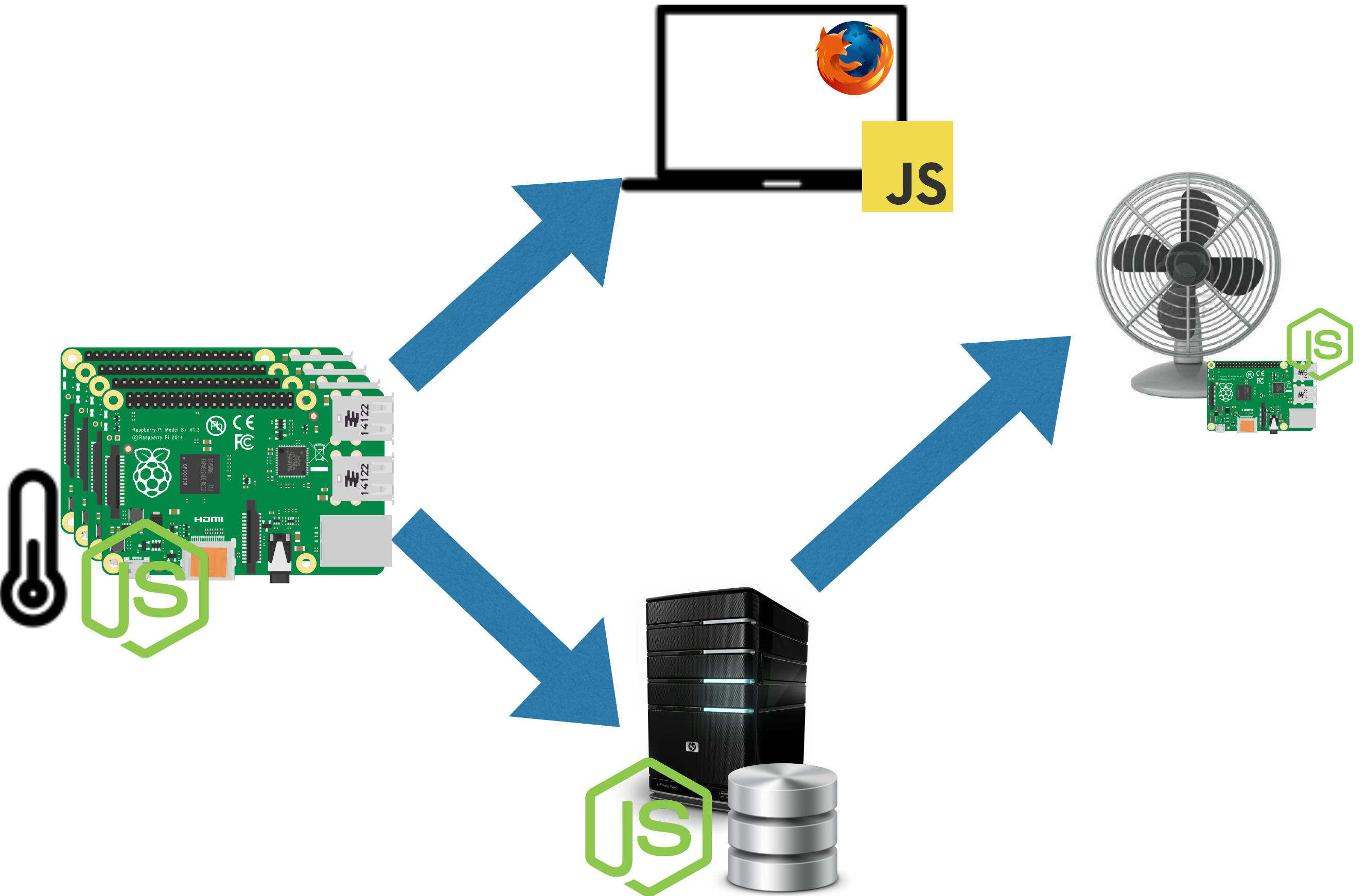
HTML

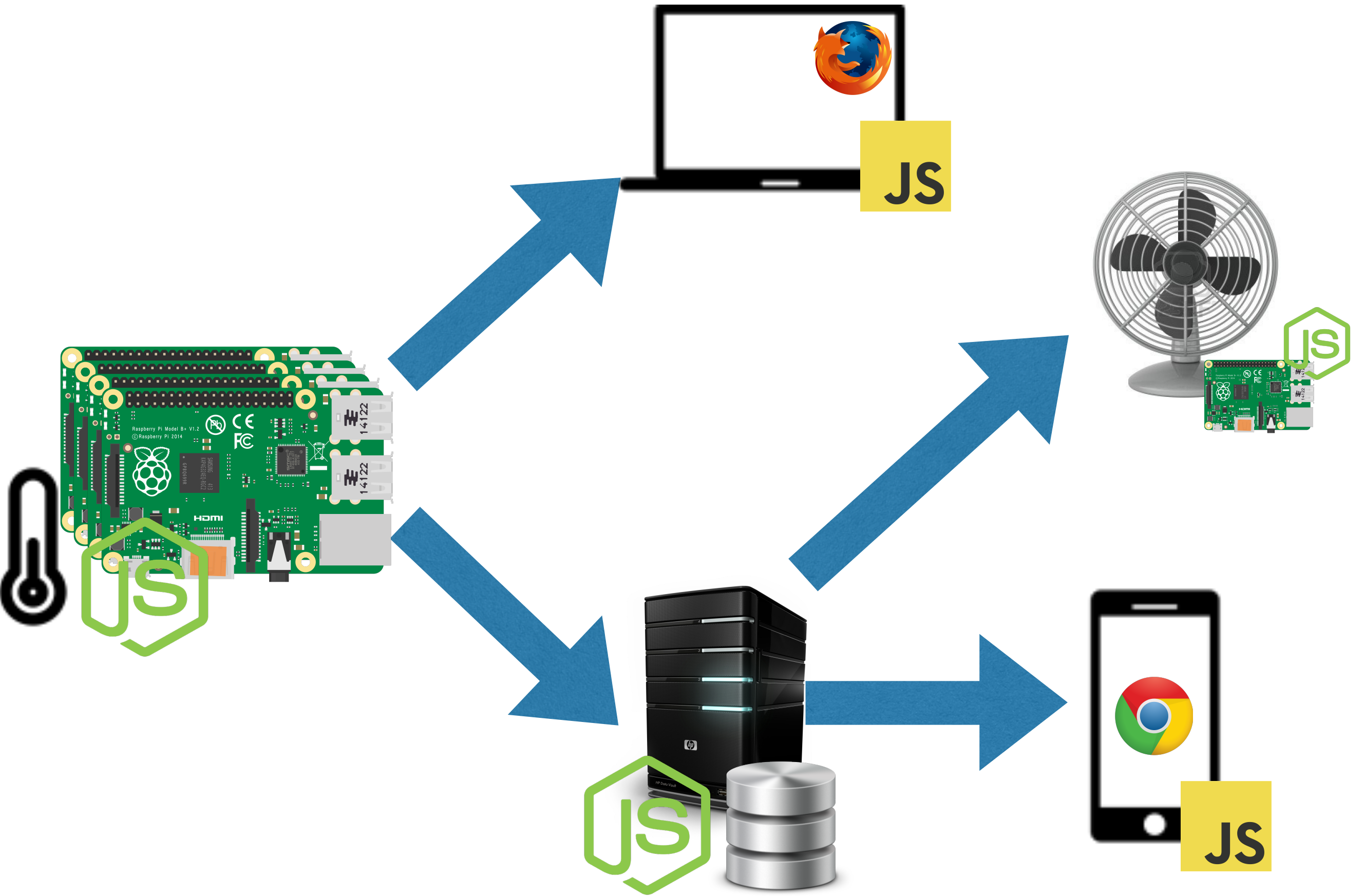


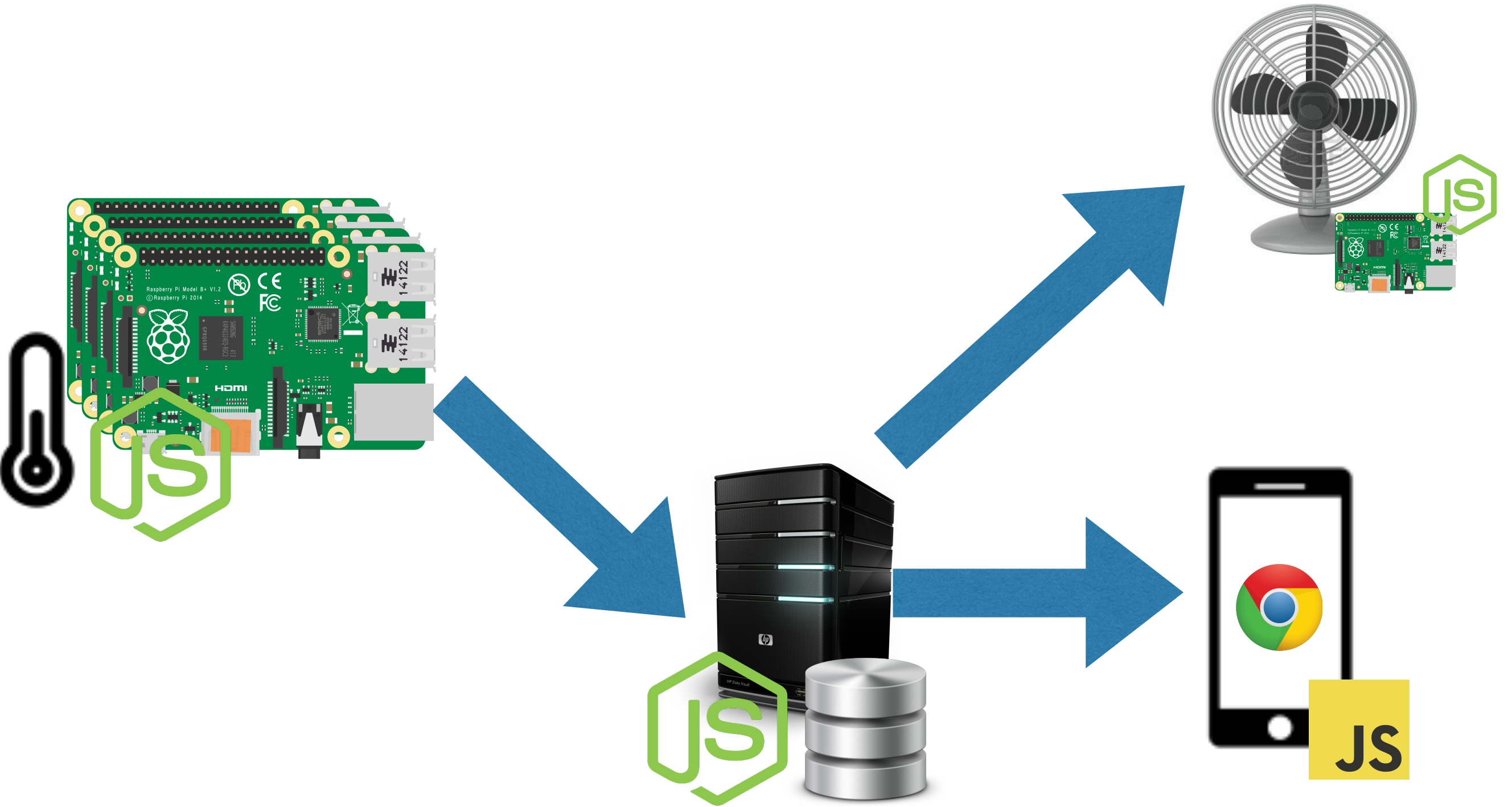
J S



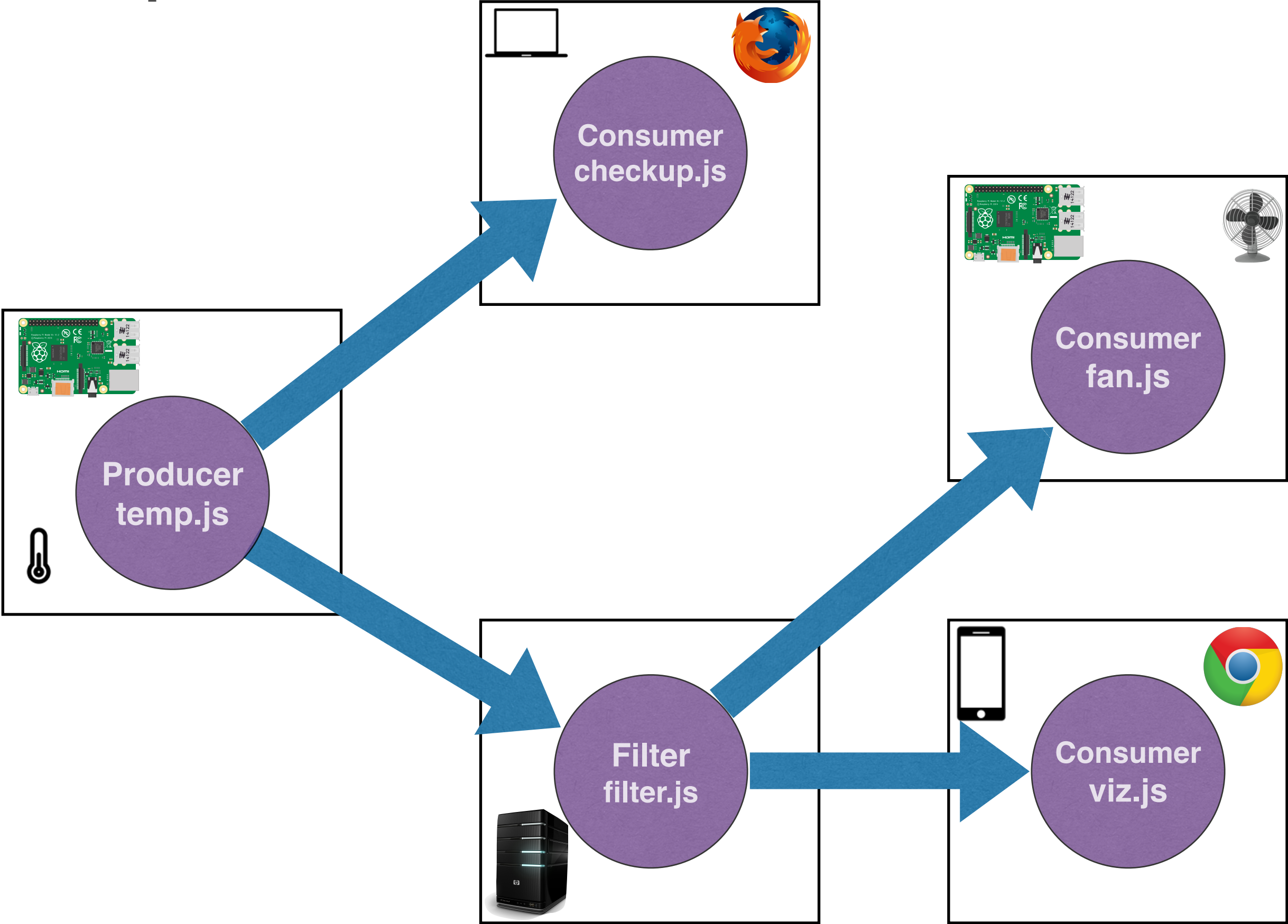




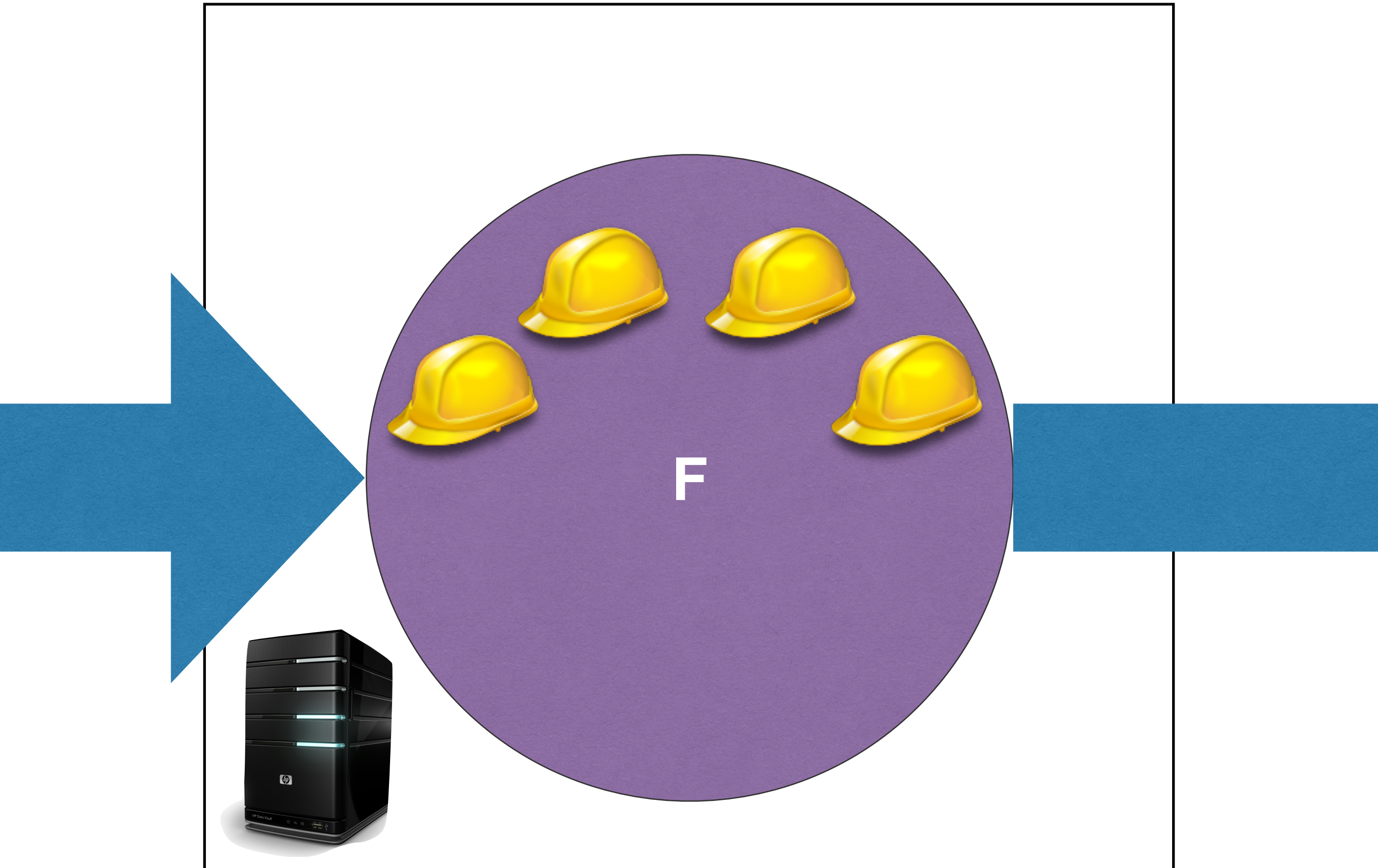




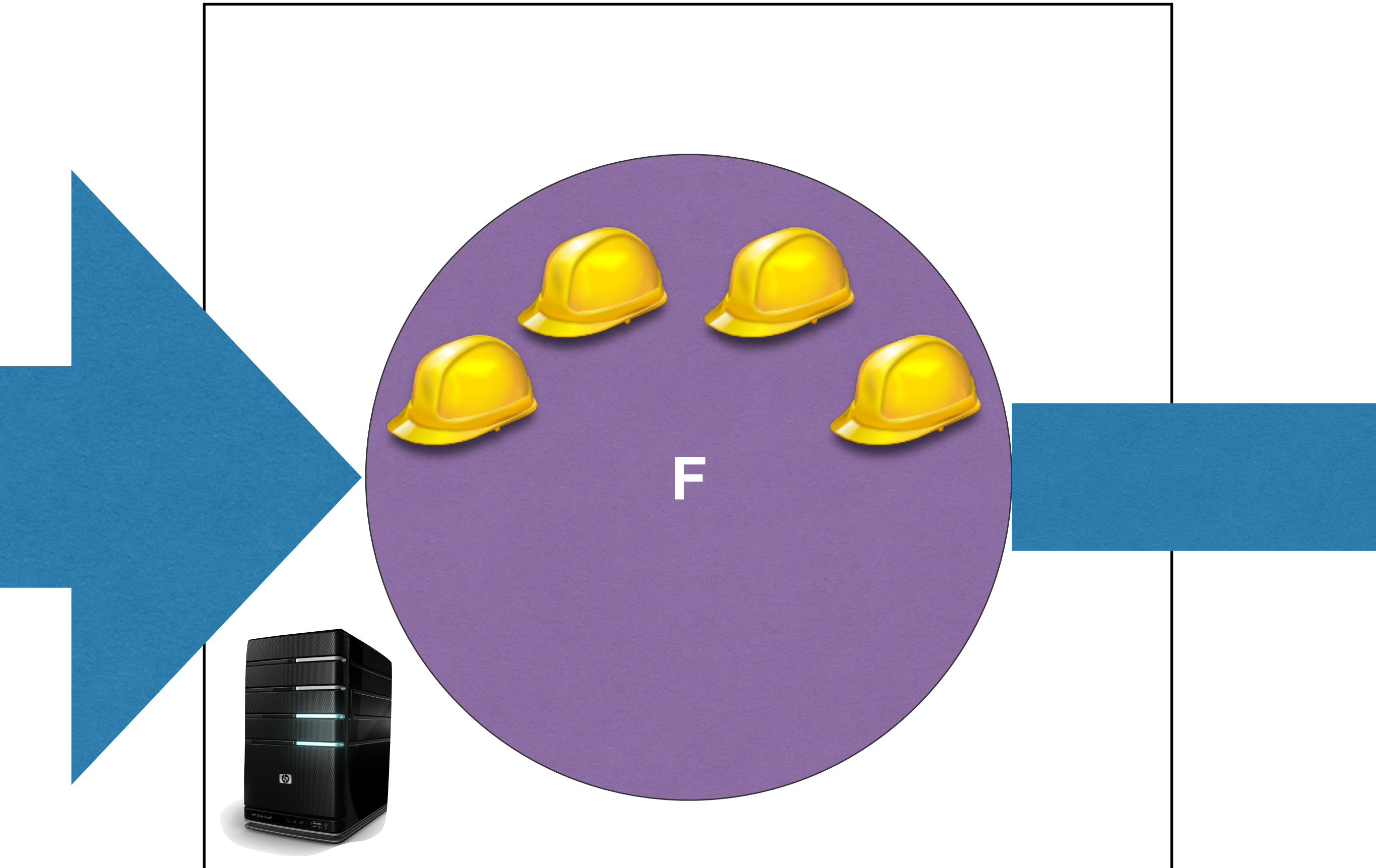
Web Liquid Streams



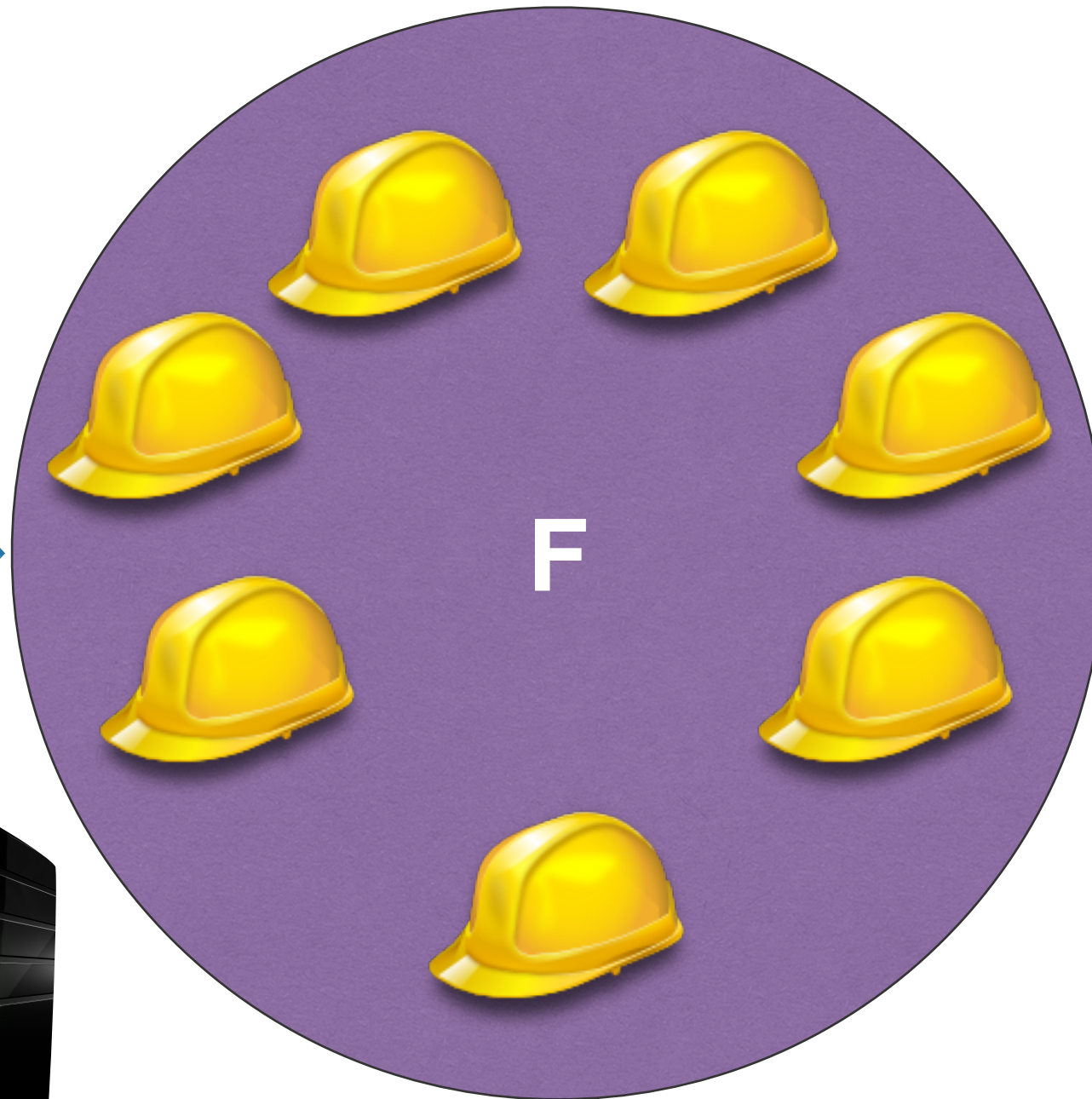
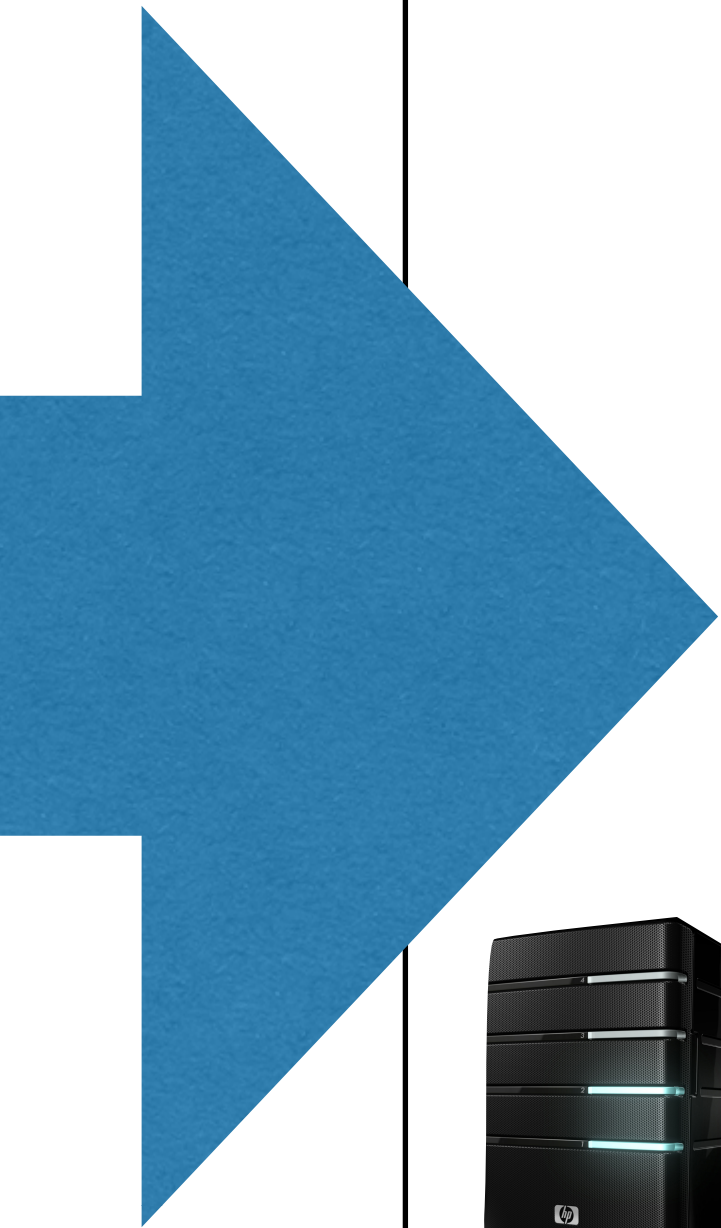
Operator Elasticity



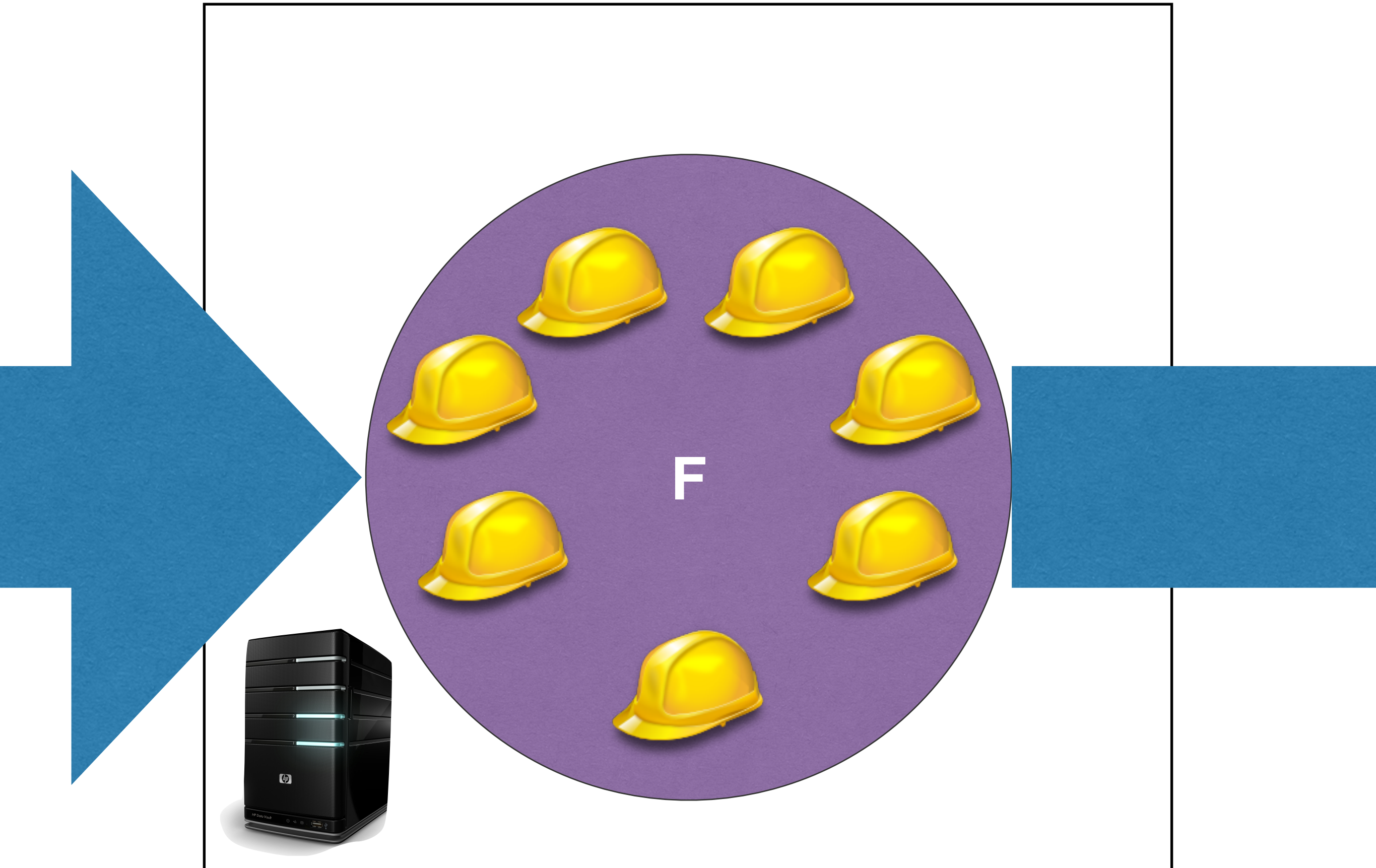
Operator Elasticity



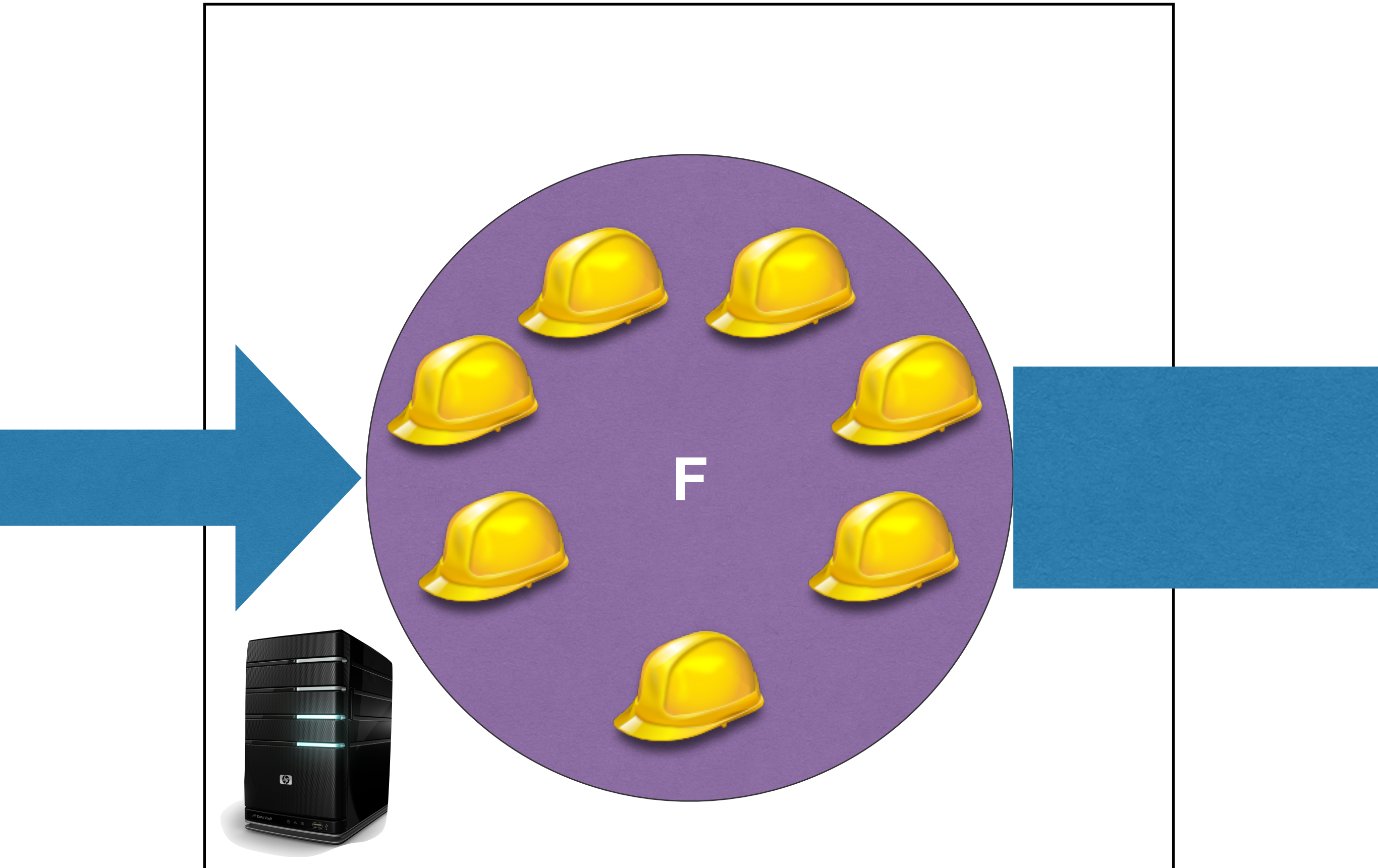
Operator Elasticity



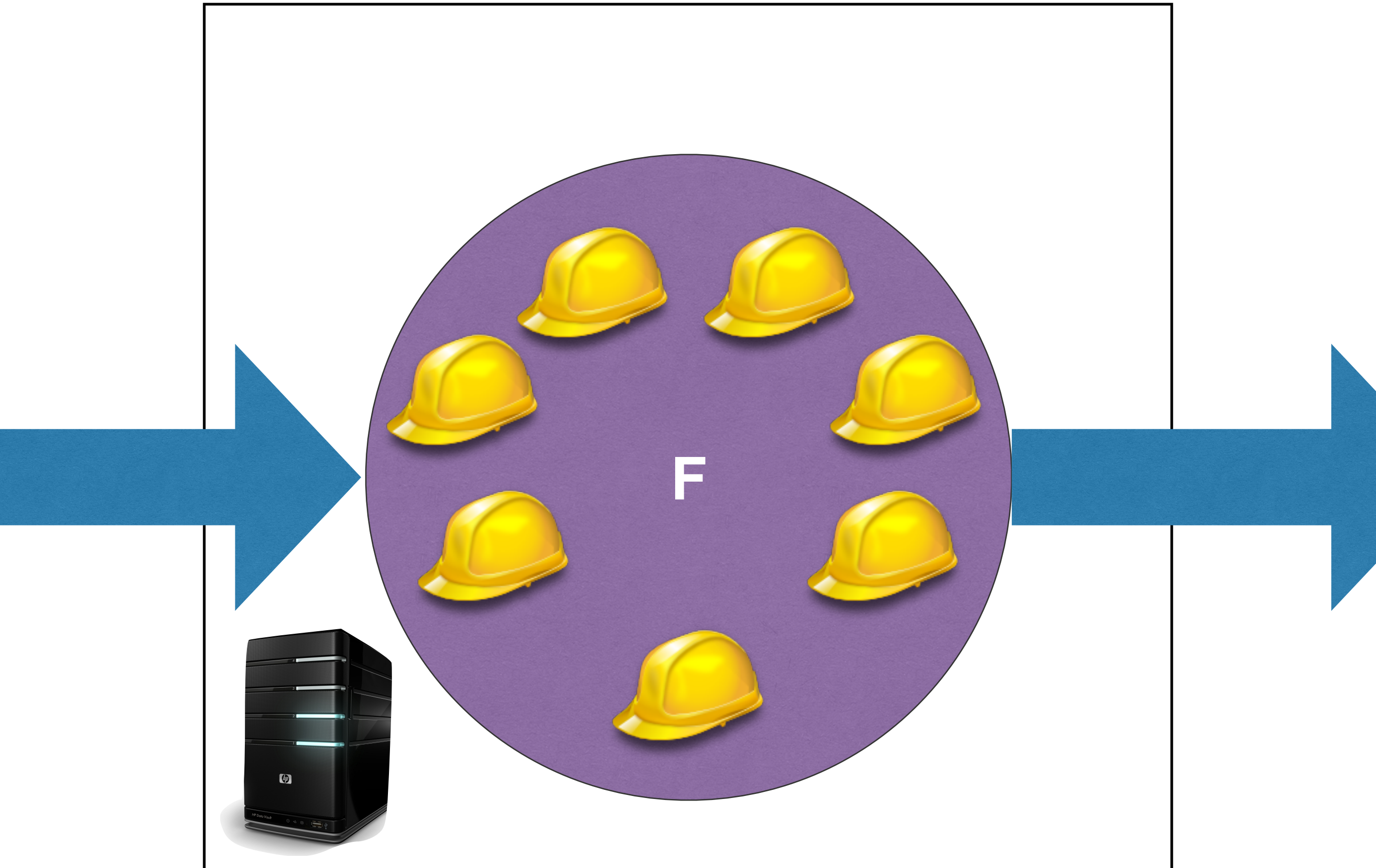
Operator Elasticity



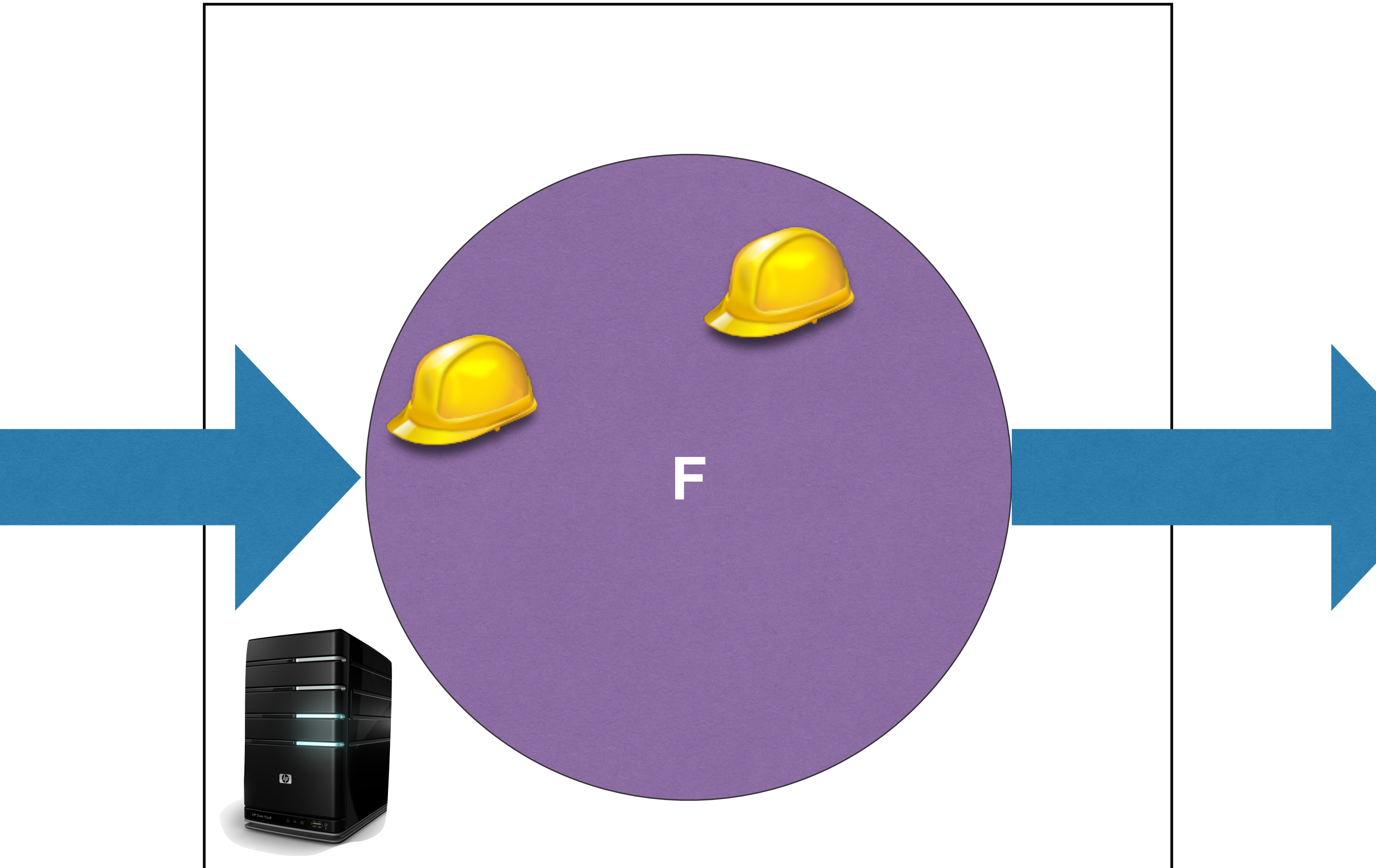
Operator Elasticity



Operator Elasticity



Operator Elasticity



```
var k = require( 'wls.js' );
```

```
var k = require( 'wls.js' );  
setInterval( function( ) {
```

```
}, 1000 );
```



```
var k = require('wls.js');  
setInterval(function(){  
    getTemperature(  
        function(temp, sensor_id){  
  
        }  
    );  
}, 1000);
```

```
var k = require('wls.js');
setInterval(function(){
    getTemperature(
        function(temp, sensor_id){
            k.send({
                "temperature" : temp,
                "id" : sensor_id,
                "ts" : new Date().getTime(),
            });
        }
    );
}, 1000);
```

[illegible]


```
var k = require('wls.js');

k.createOperator(function(msg) {
    var setpoint = k.db_get("setpoint", temp_data.id);
    var threshold = k.db_get("threshold",
temp_data.id);
    //[...] compute average

});
```

```
var k = require('wls.js');  
k.createOperator(function(msg) {  
    var setpoint = k.db_get("setpoint", temp_data.id);  
    var threshold = k.db_get("threshold",  
temp_data.id);  
    //[...] compute average  
    if(abs(average - setpoint) > threshold){  
        .  
        .  
        .  
    }  
    else [...]  
});
```

```
var k = require('wls.js');

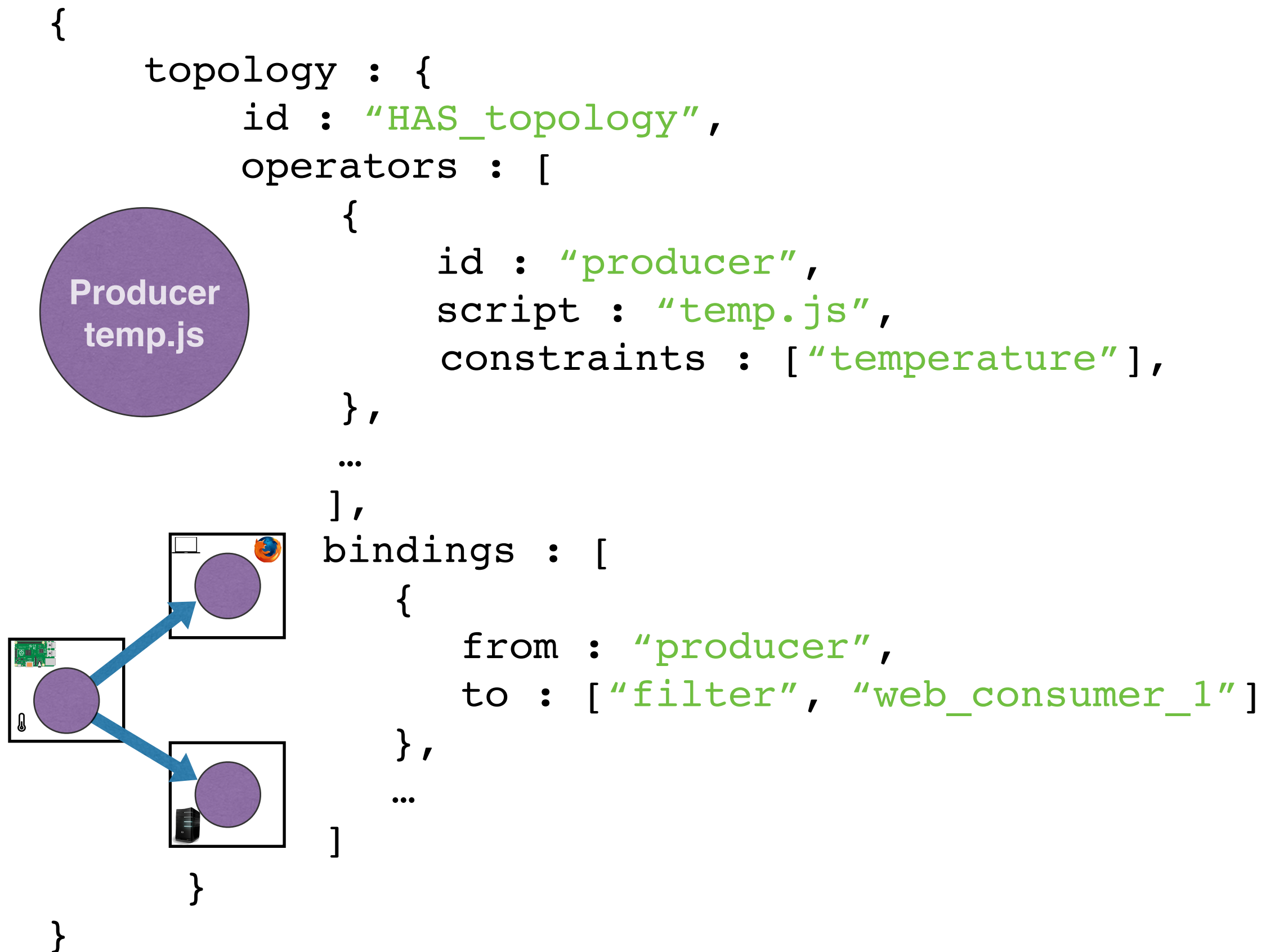
k.createOperator(function(msg) {
    var setpoint = k.db_get("setpoint", temp_data.id);
    var threshold = k.db_get("threshold",
temp_data.id);
    //[...] compute average
    if(abs(average - setpoint) > threshold){
        k.send({
            "start" : true,
            "room_id" : temp_data.id
        });
    }
    else [...]
});
```



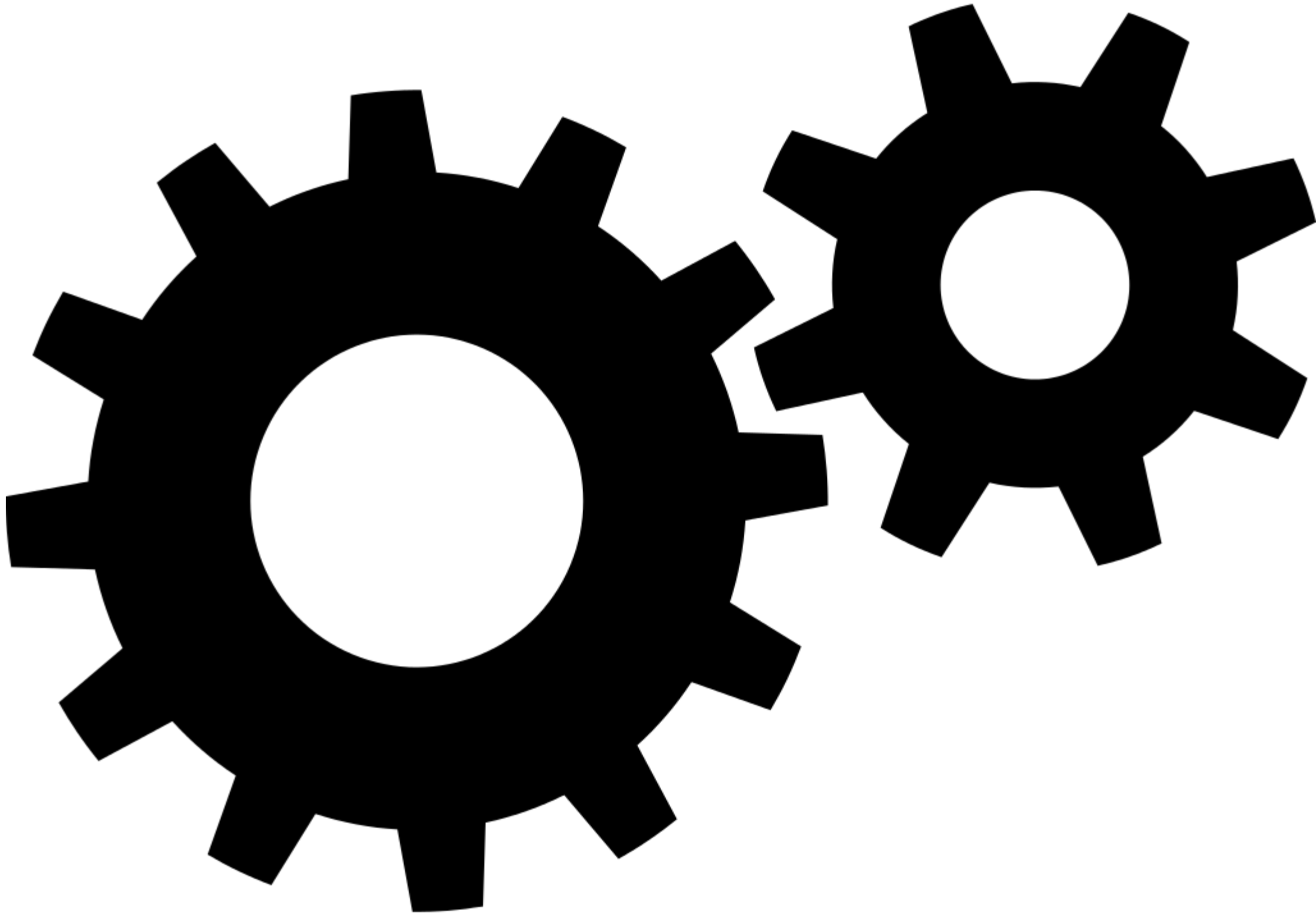
```
var k = require('wls.js');  
k.createOperator(function(msg) {  
  
    }  
});
```

```
var k = require( 'wls.js' );  
k.createOperator( function( msg ) {  
    if( this.id == msg.id ) {  
        if( msg.start ) {  
            startFan( );  
        }  
        else {  
            stopFan( );  
        }  
    }  
} ) ;
```

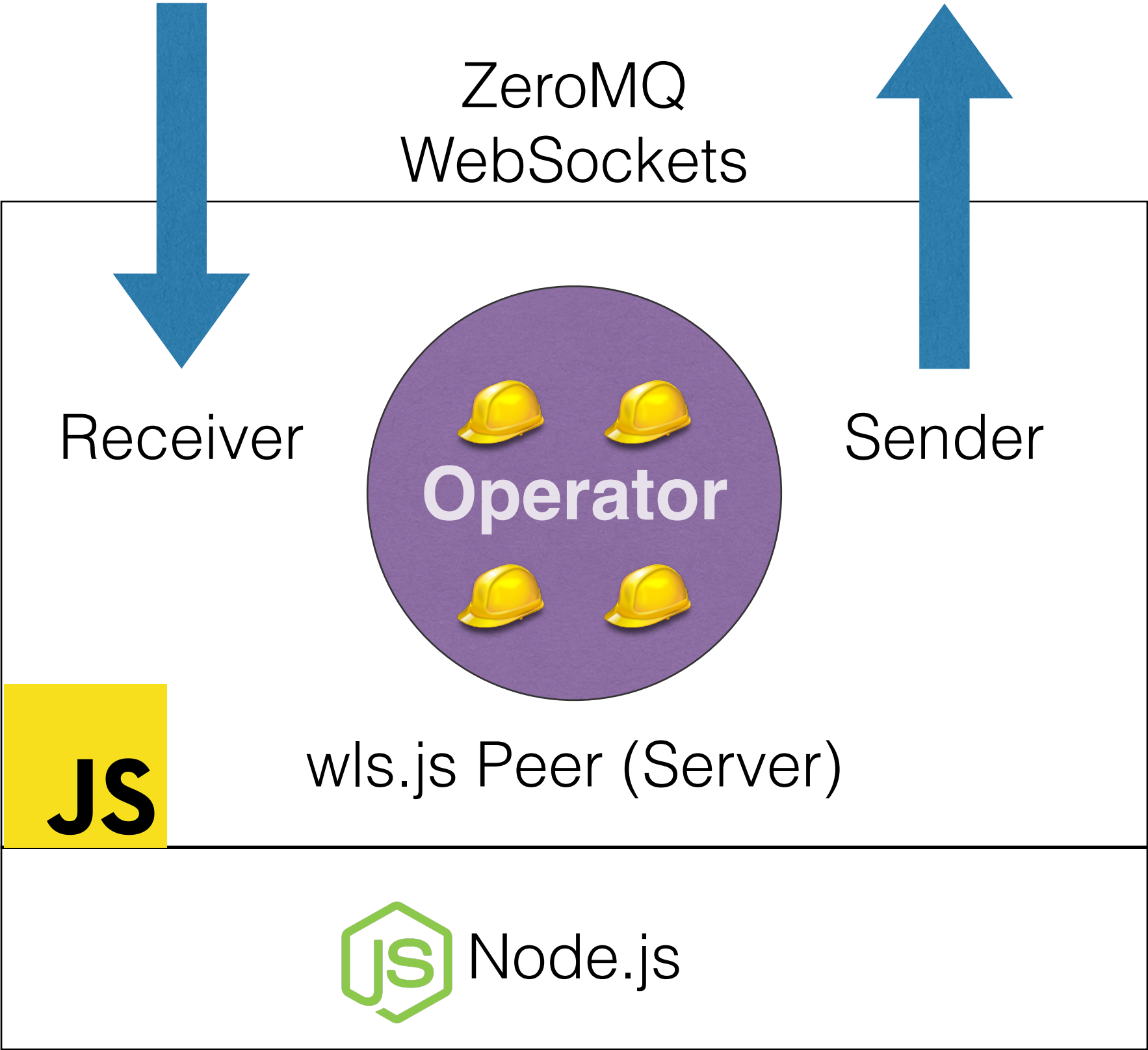
JSON Topology



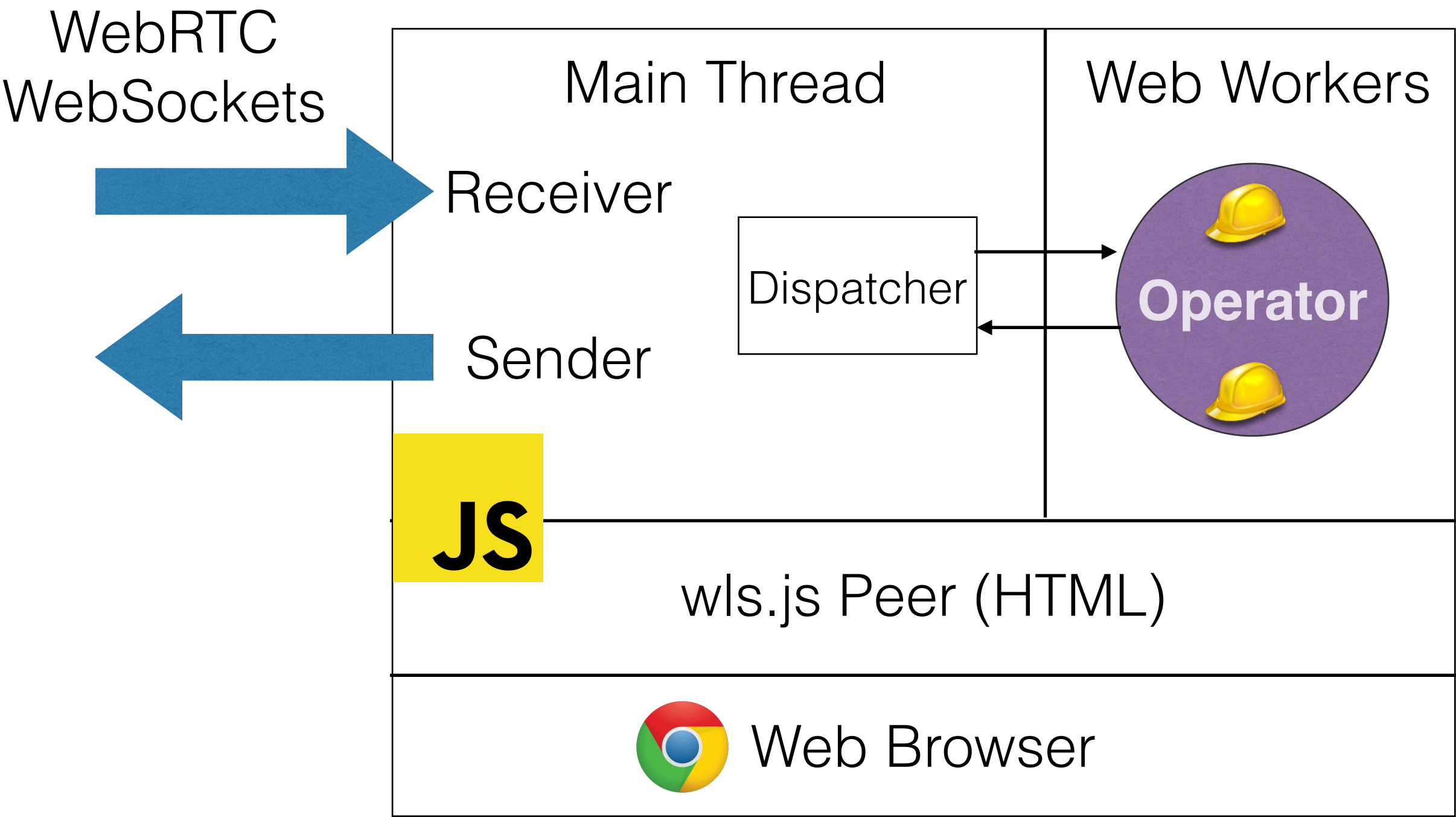
Web Liquid Streams Runtime



Web Server Runtime

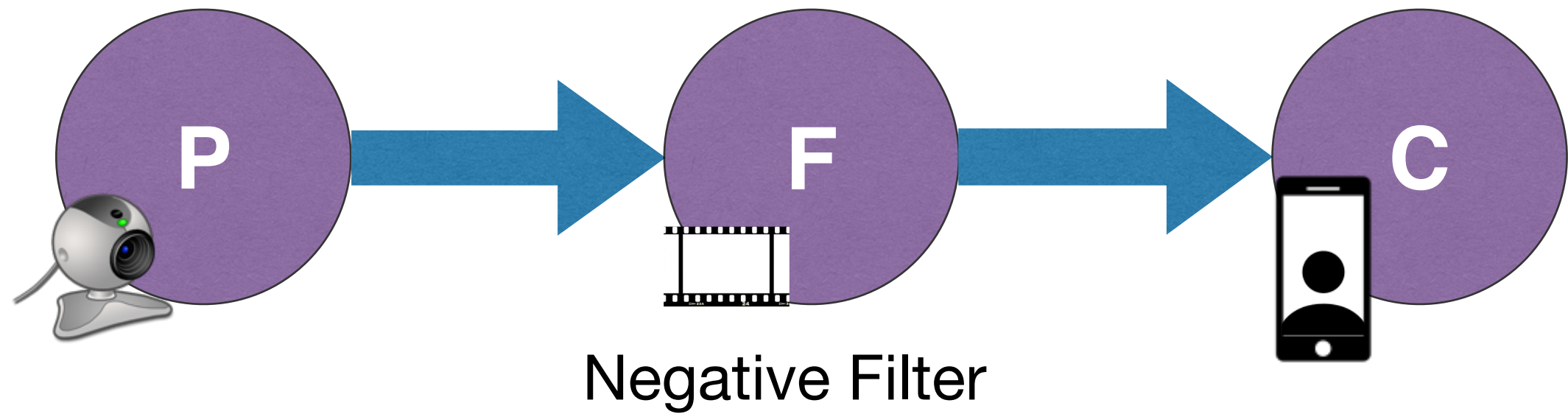


Web Browser Runtime

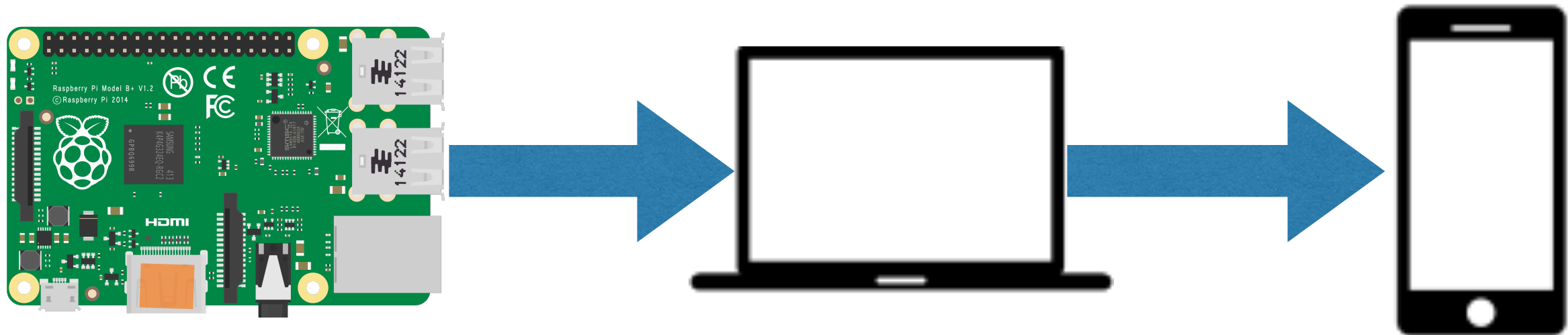




Variable Workload



Variable Workload

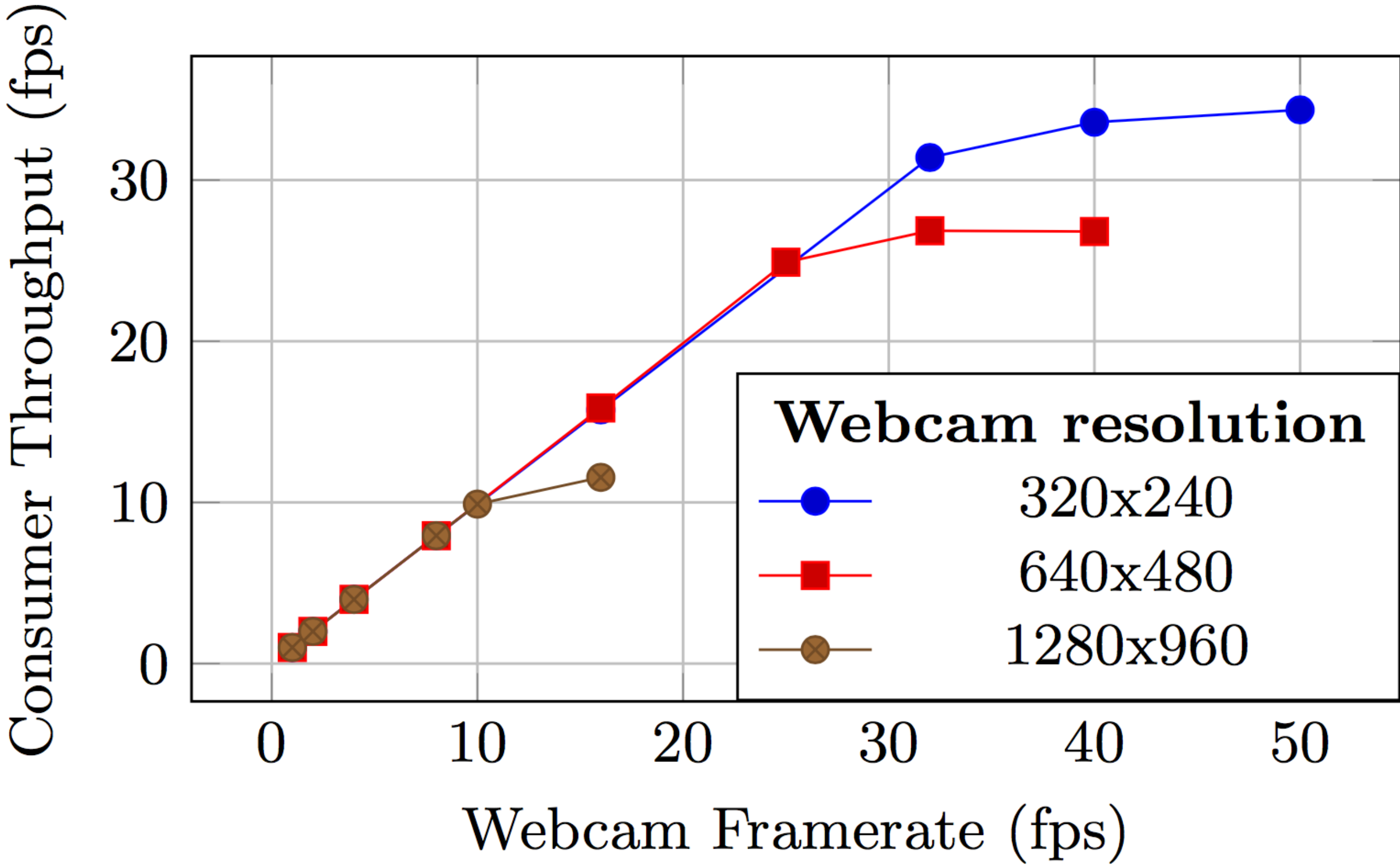


Raspberry Pi
1 core 700MHz
256MB RAM
NodeJS v.0.10.15

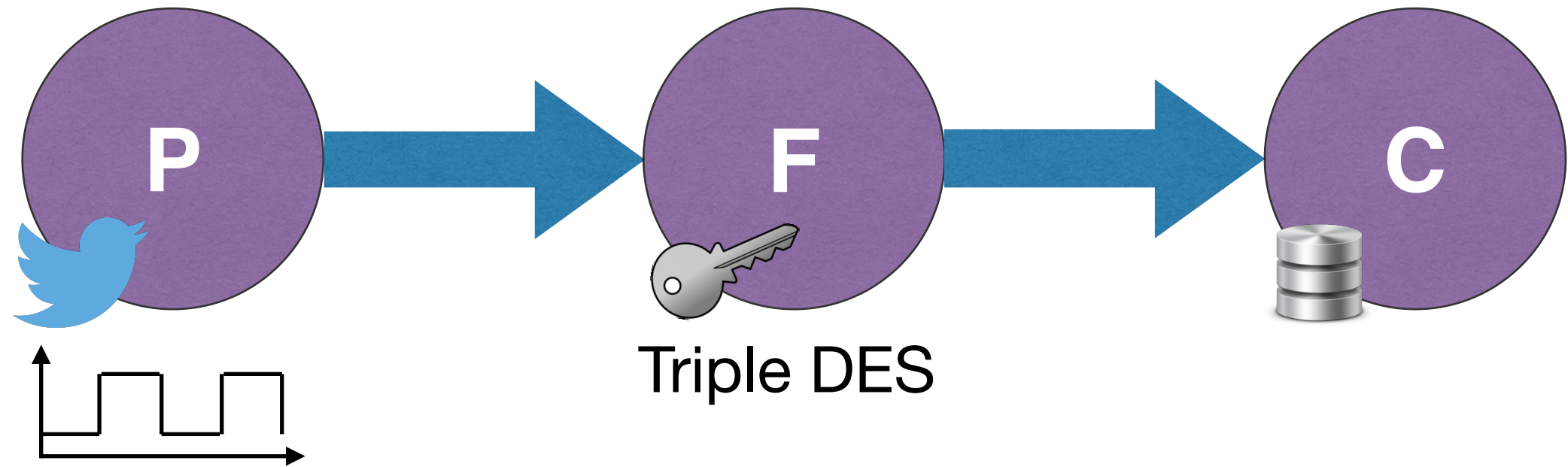
MacBook Pro (2012)
4 cores 2.3GHz
16GB RAM
Chrome v. 37

iPhone 5S
1 core 1.3GHz
1 GB RAM
Chrome v. 40

Variable Workload



Variable Workload



Variable Workload

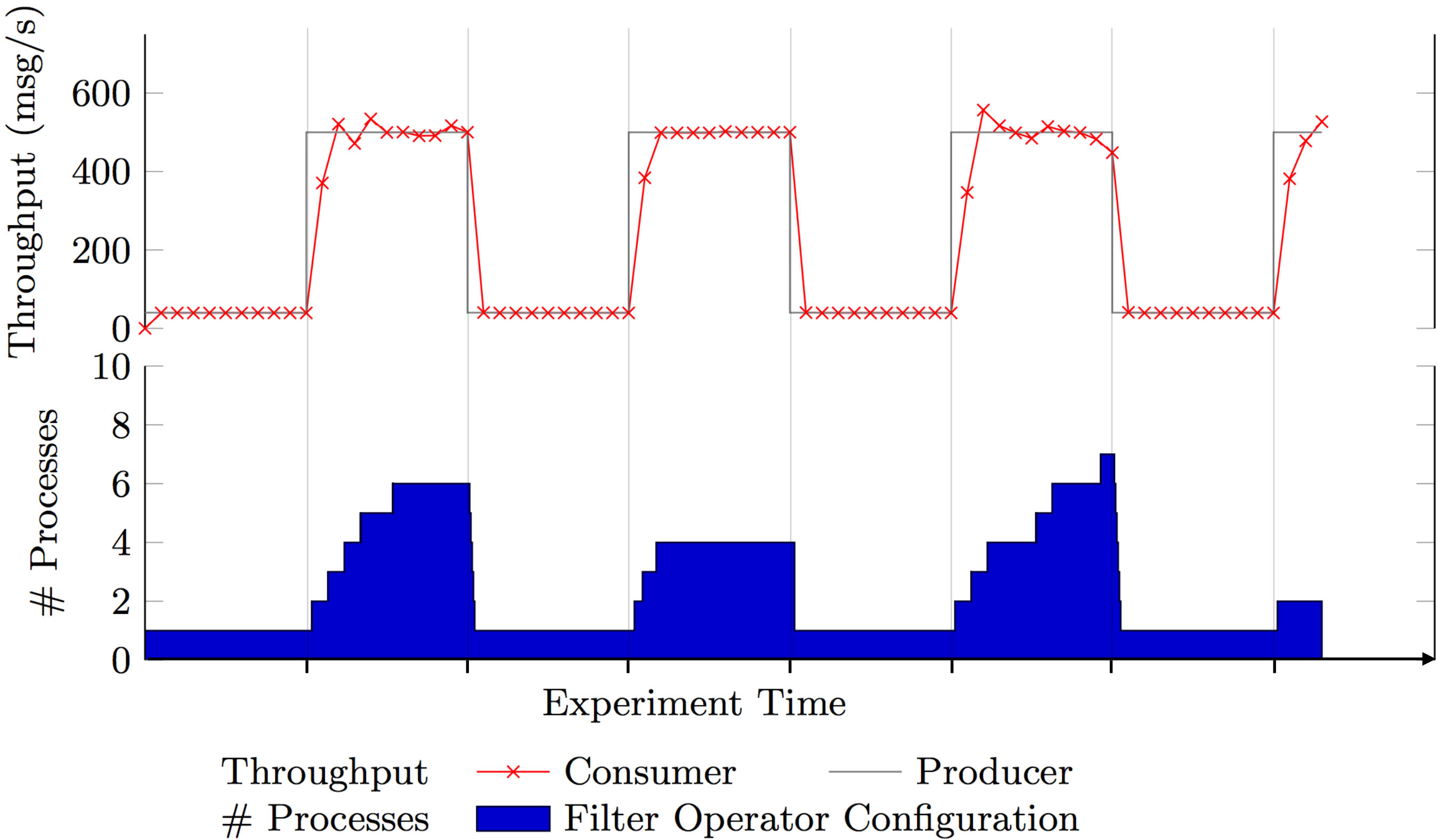


MacBook Pro (2012)
4 cores 2.3GHz
16GB RAM
NodeJS v. 0.10.15

Web Server (Ubuntu)
24 cores 2GHz
128GB RAM
NodeJS v. 0.10.15

Web Server (Ubuntu)
4 cores 3GHz
8GB RAM
NodeJS v. 0.10.15

Variable Workload



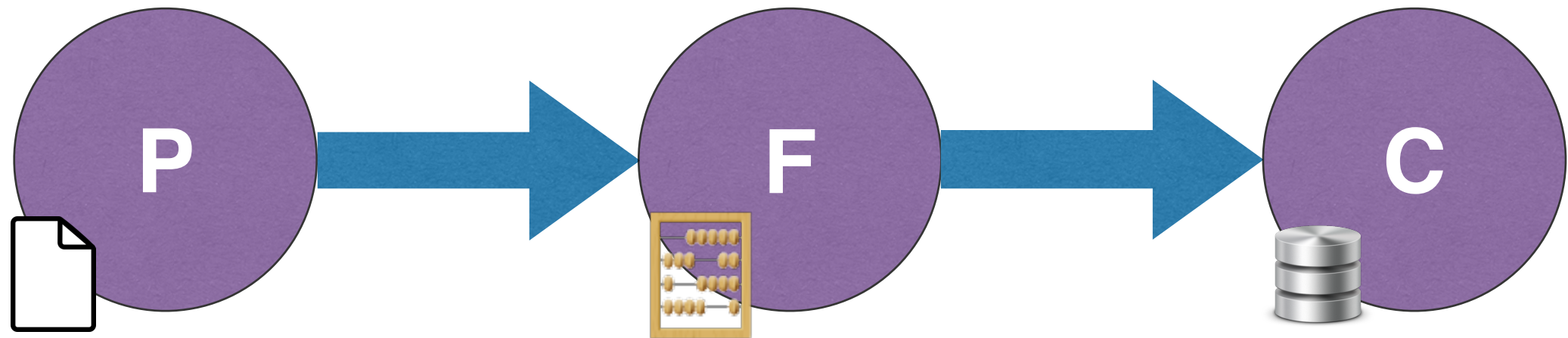
Centralised vs. P2P



vs.



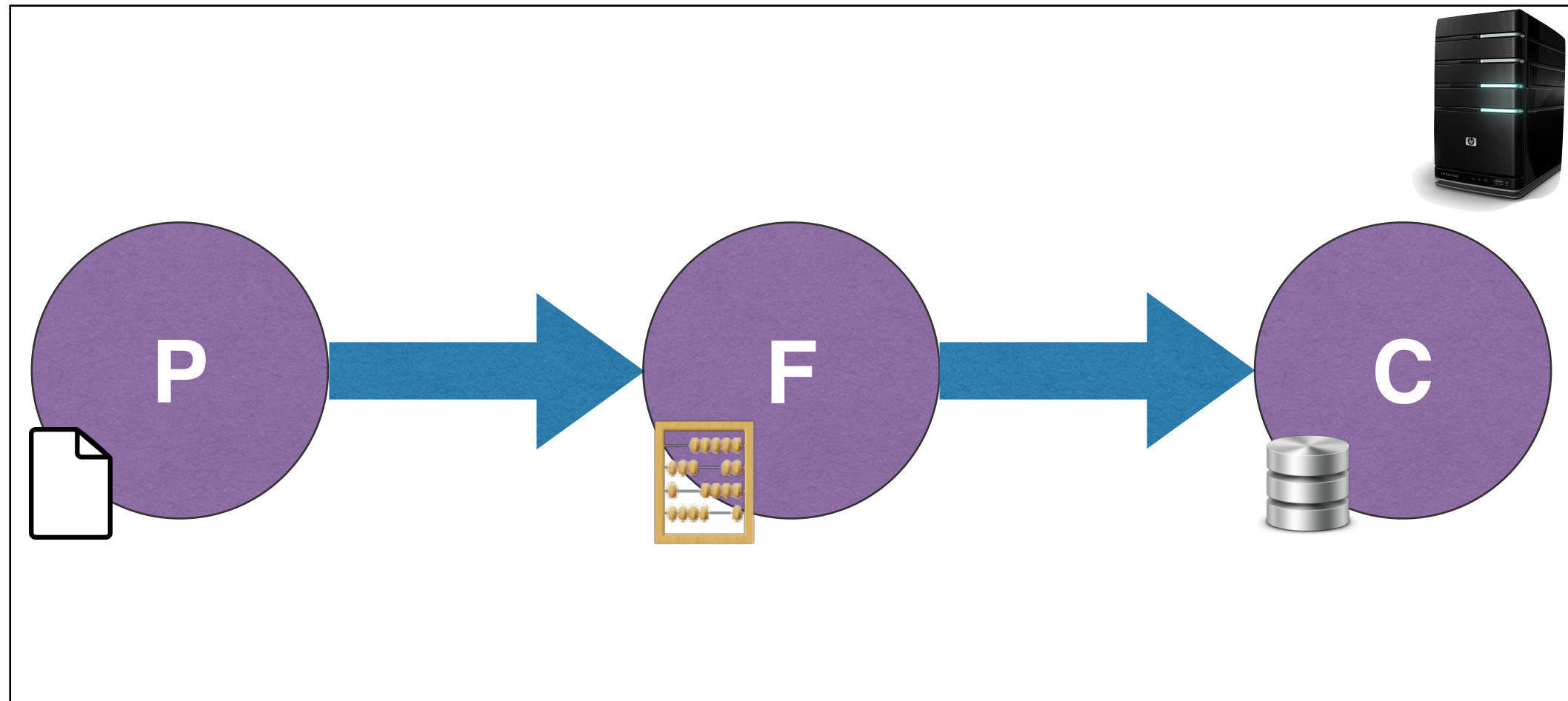
Centralised vs. P2P



Prime Numbers

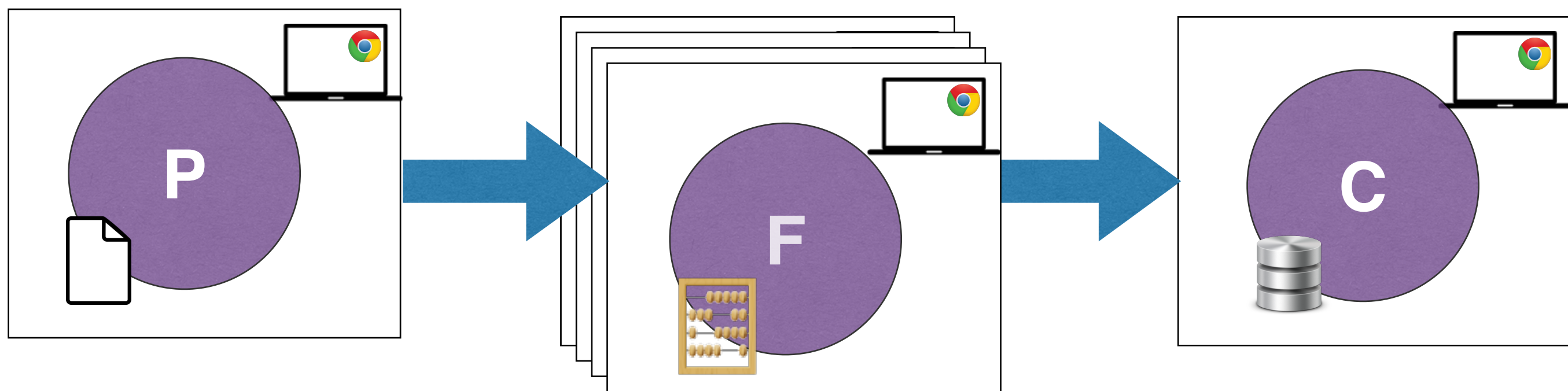
Centralised

Single WLS Peer



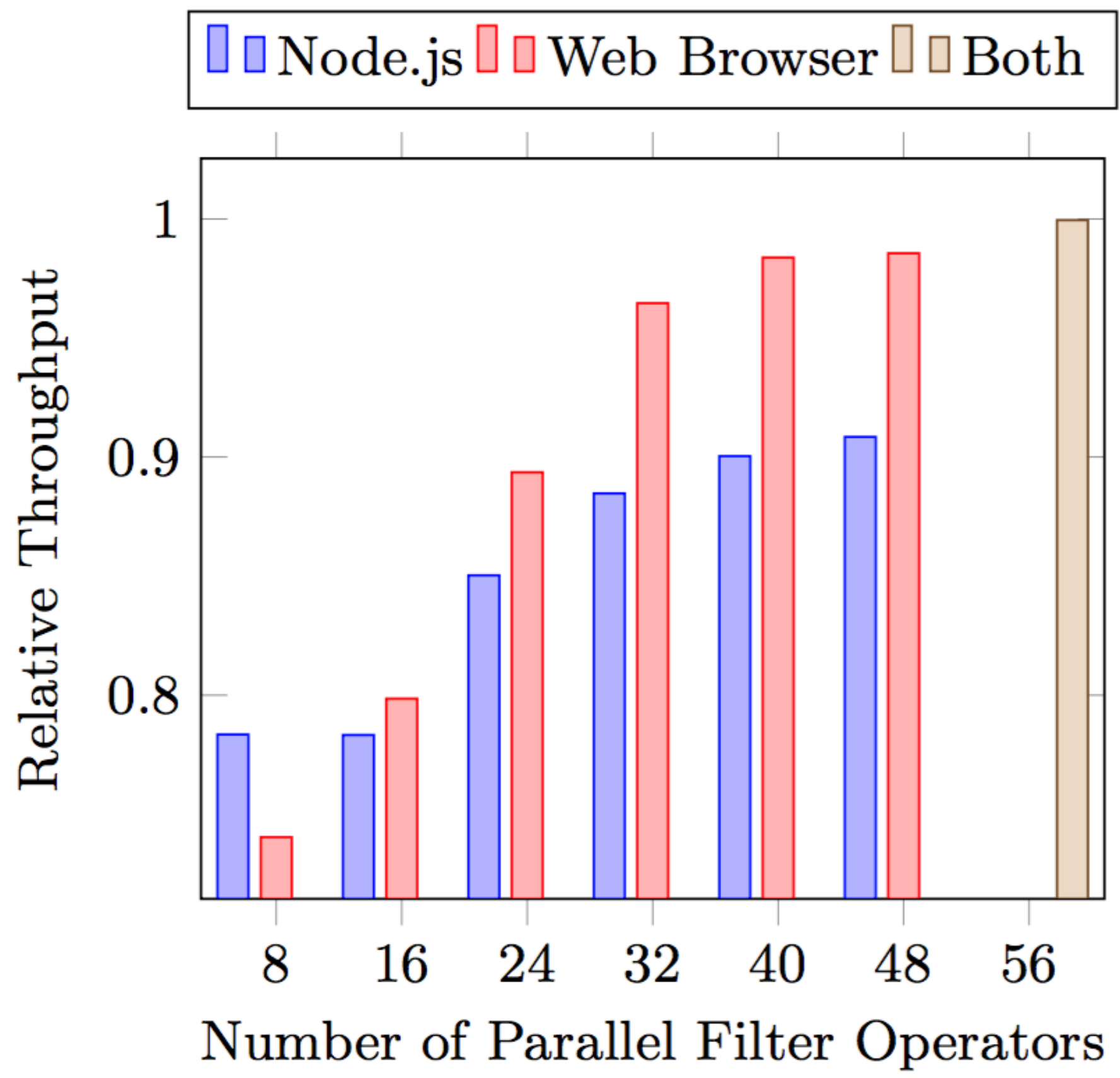
Web Server (Ubuntu)
24 cores 2GHz
128GB RAM
NodeJS v. 0.10.15

Centralised vs. _____

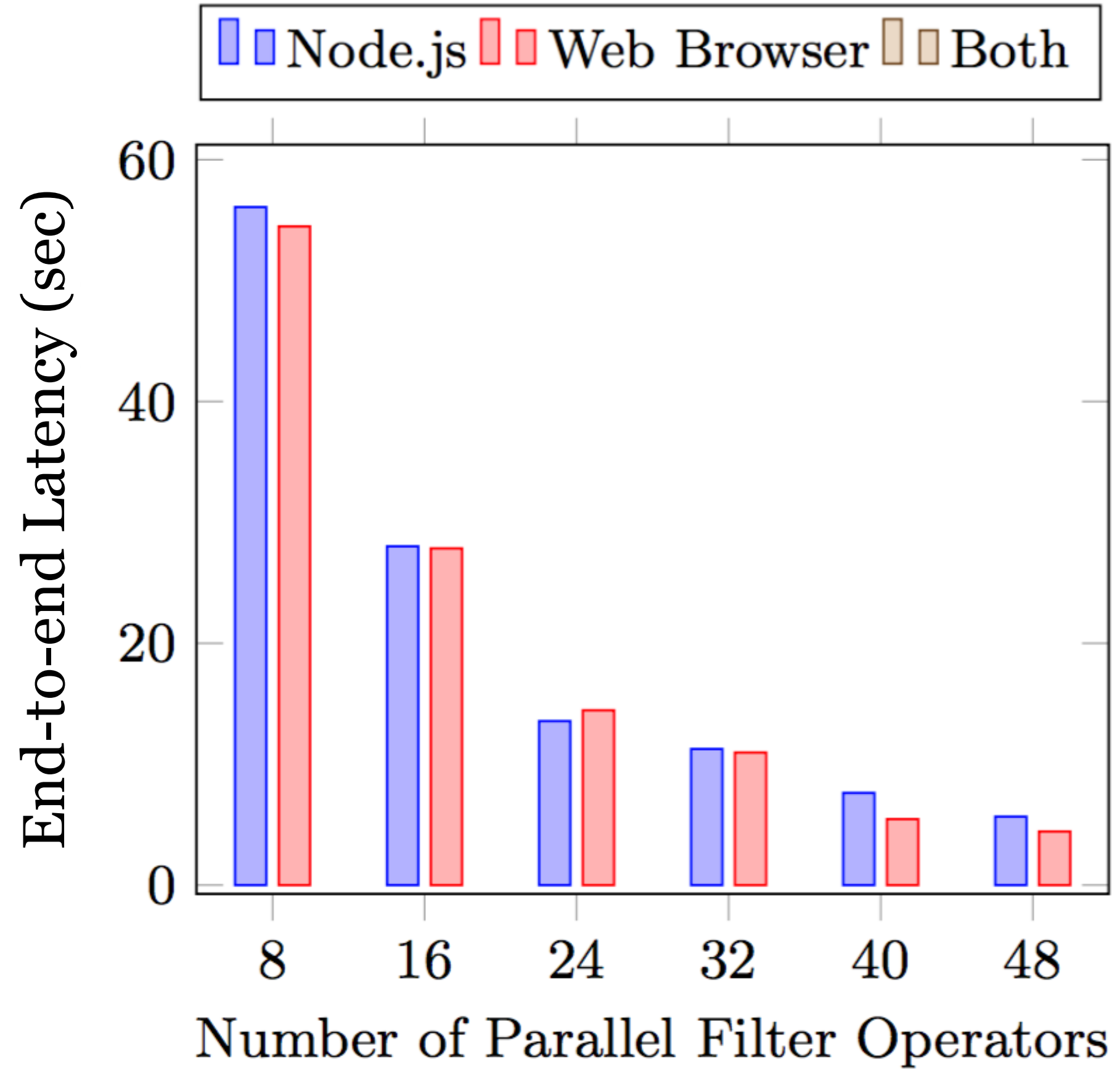


MacBook Pro (2012)
4 cores 2.3GHz
16GB RAM
Chrome v. 37

Centralised vs. P2P



Centralised vs. P2P

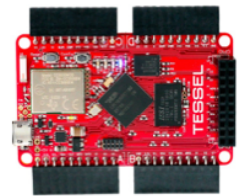


WEB

LIQUID

STREAMS

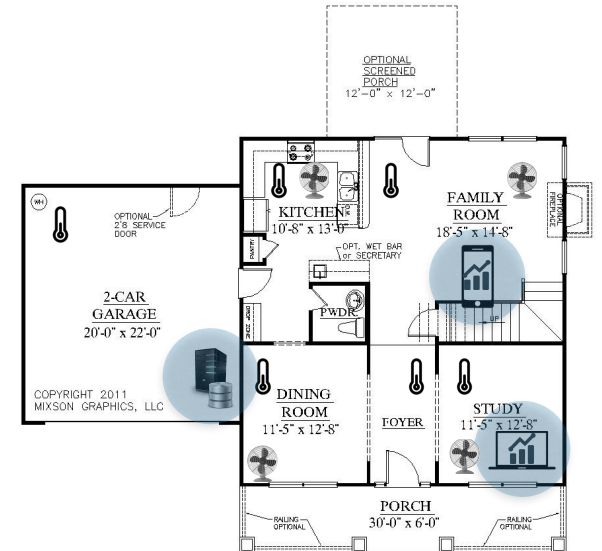
WEB



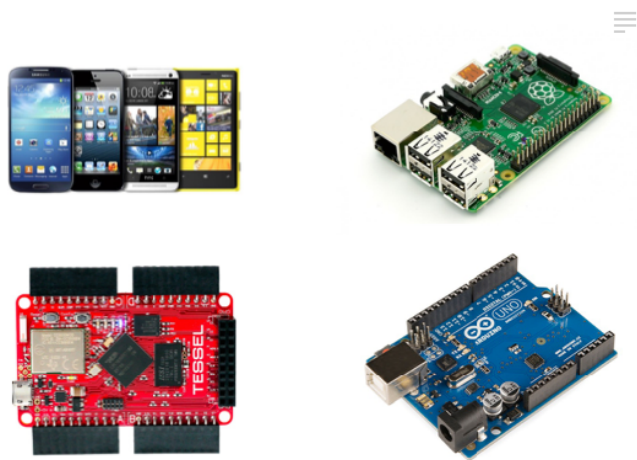
LIQUID



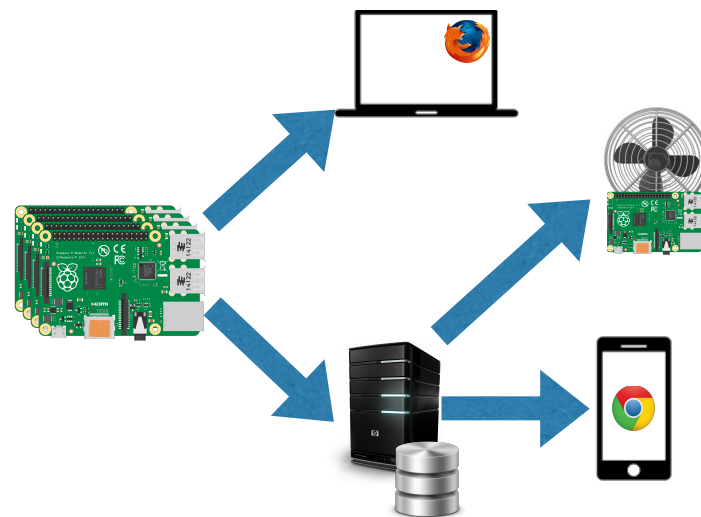
STREAMS



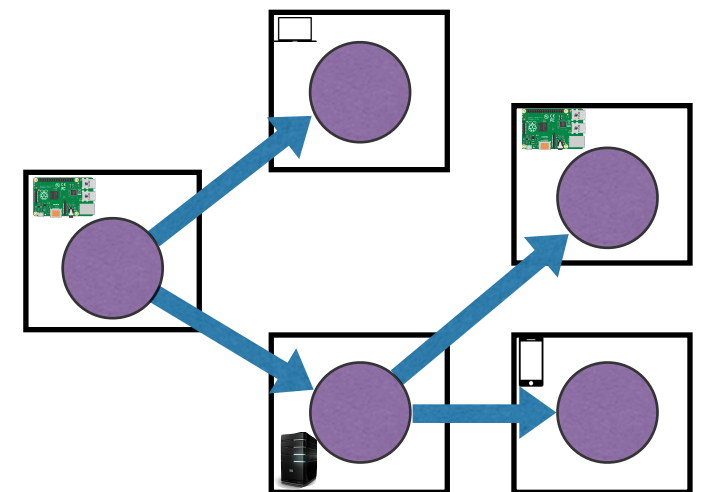
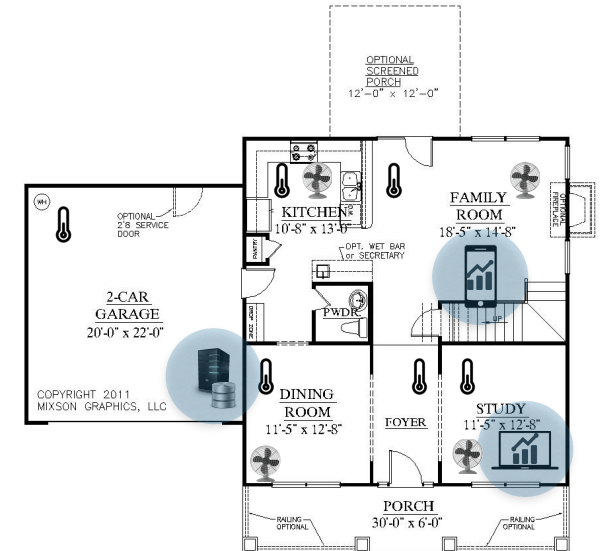
WEB



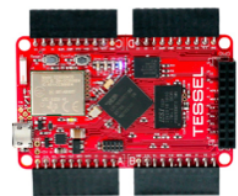
LIQUID



STREAMS



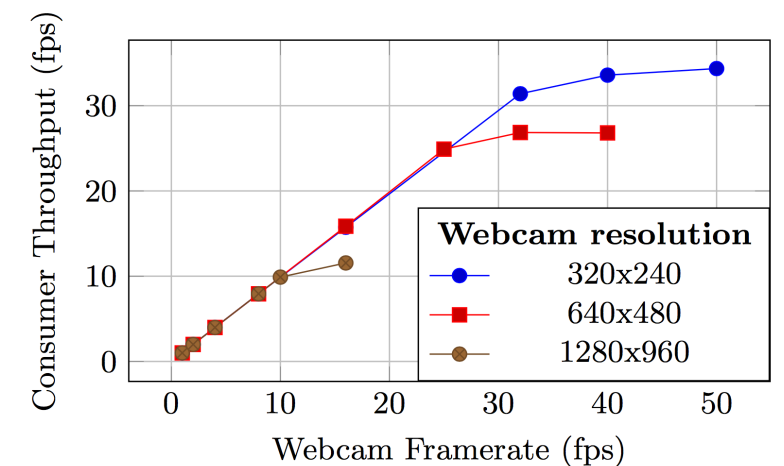
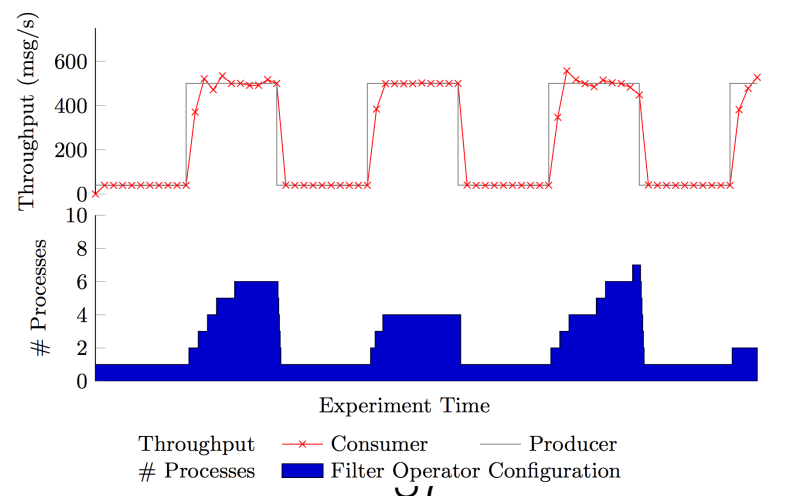
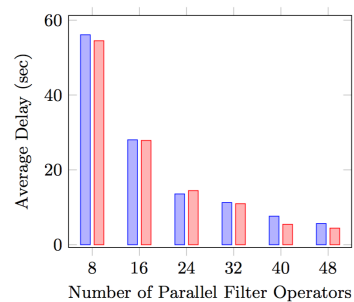
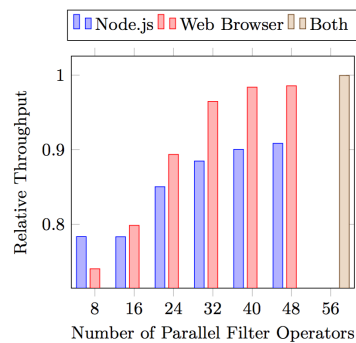
WEB



WebRTC



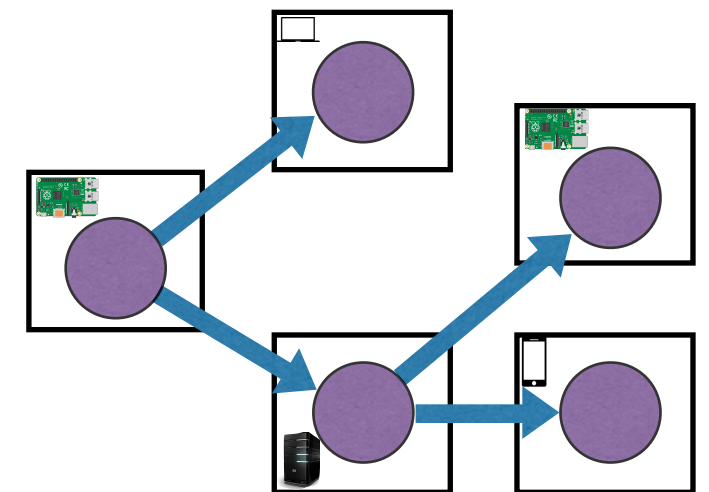
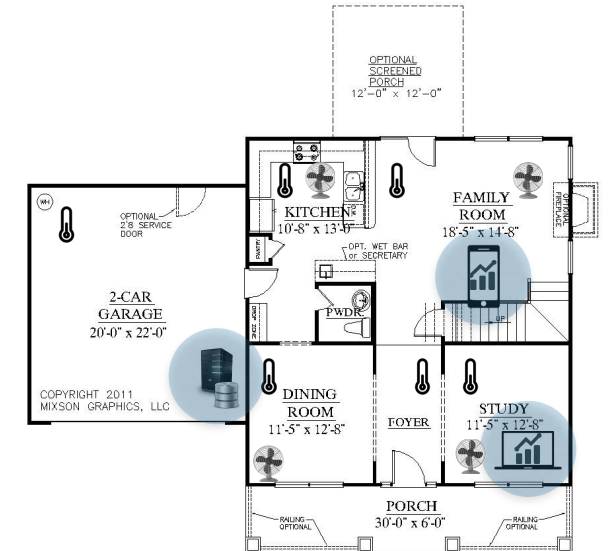
JS



LIQUID



STREAMS



Decentralized Stream Processing over Web-enabled devices

Masiar Babazadeh **Andrea Gallidabino** **Cesare Pautasso**

@masiarb

@agallidabino

@pautasso

name.surname@usi.ch

University of Lugano
Faculty of Informatics
Switzerland

Thank you.

