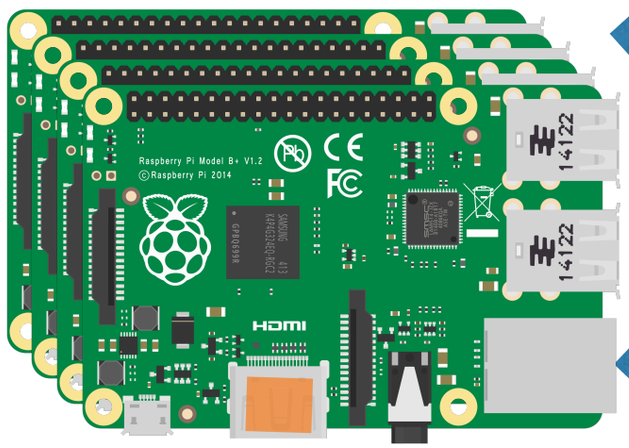
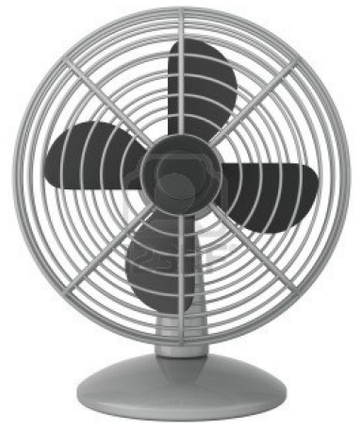
Liquid Stream Processing Across Web Browsers and Web Servers

Masiar Babazadeh
@masiarb
Andrea Gallidabino
@agallidabino
Cesare Pautasso
@pautasso
name.surname@usi.ch

University of Lugano
Switzerland

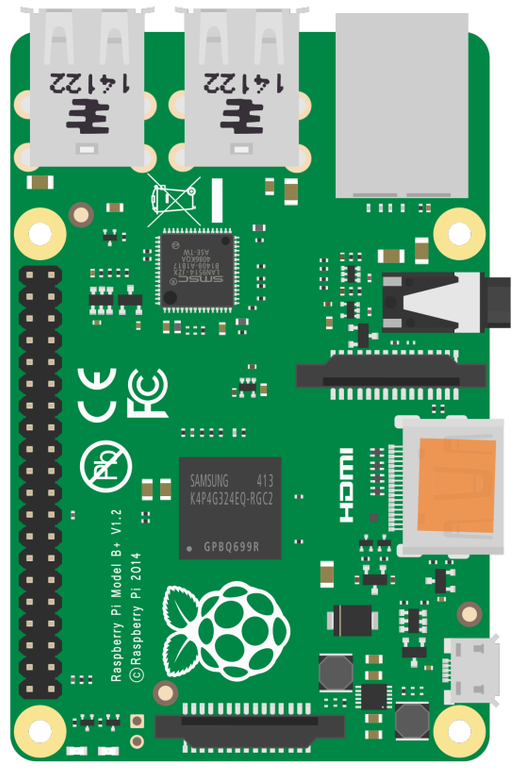




JS

Web Liquid Streams

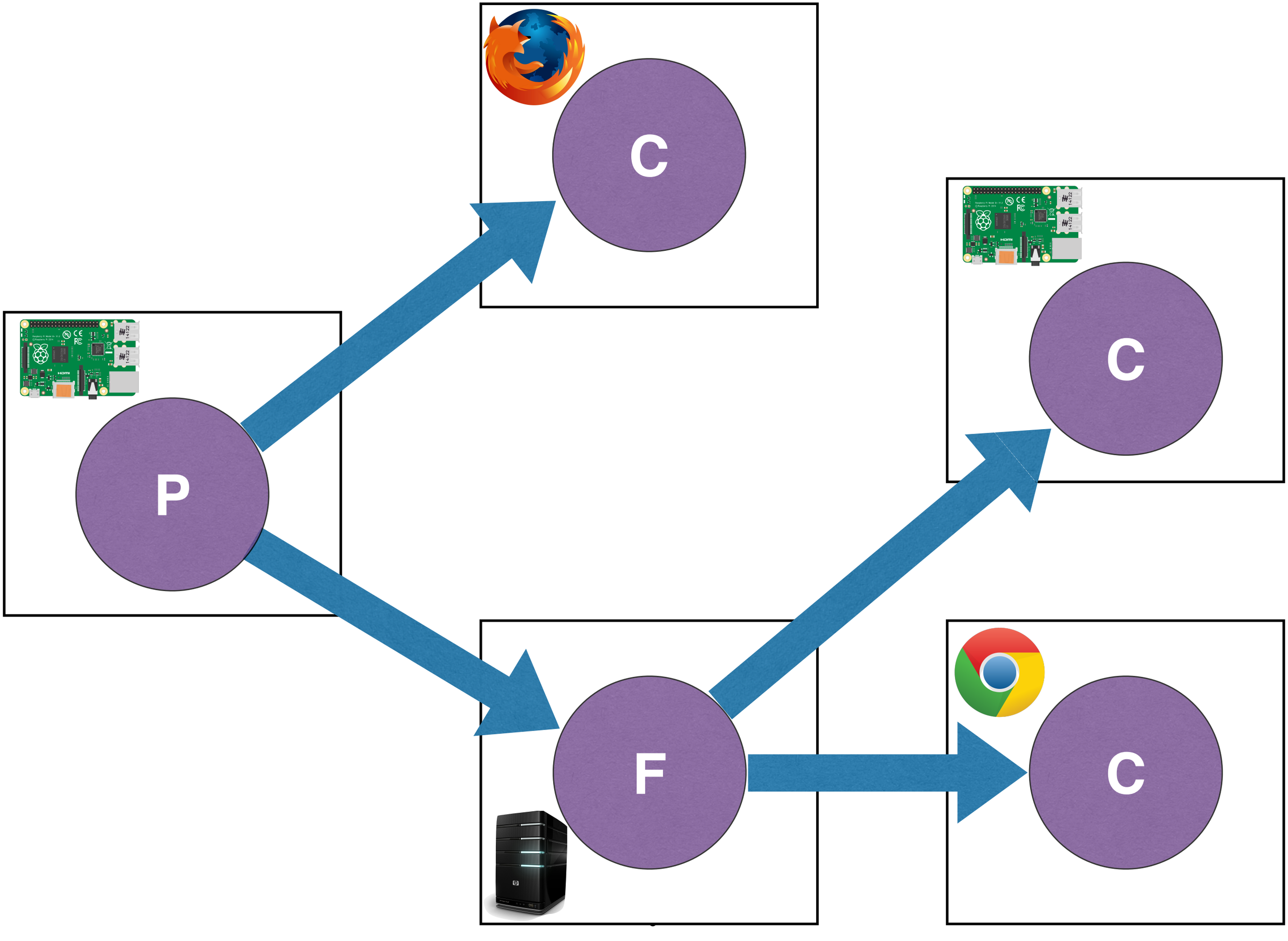

WebRTC

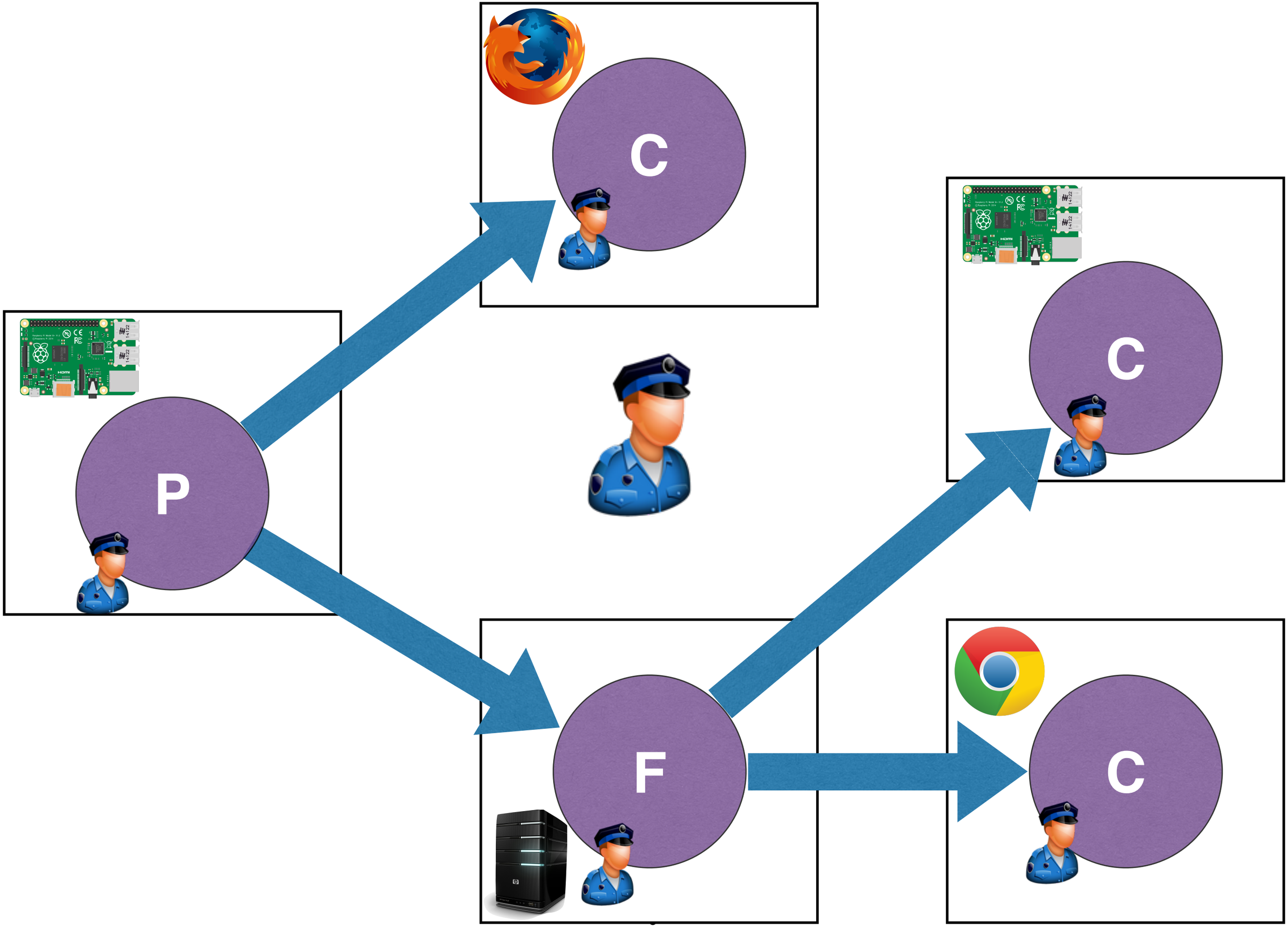

SAMSUNG 413
K4P4G324EQ-RGC2
GPBQ699R
Raspberry Pi Model B+ V1.2
©Raspberry Pi 2014
HDMI

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


JS

```
{
    topology : {
        id : “HAS_topology”,
        operators : [
            {
                id : “producer”,
                script : “pi_prod.js”,
                constraints : [“temperature”],
            },
            …
           ],
           bindings : [
              {
                 from : “producer”,
                 to : [“filter”, “web_consumer_1”]
                 type : “broadcast”
              },
              …
           ]
    }
}
```


C
P
C
F
C


C
P
C
F
C

			R_1	R_2	R_3
Peer	*Con.*	Hardware	x	x	x
		CPU		x	
	Variables	Plugged		x	
		Battery		x	
		Parallelism		x	
		Deployment Cost			
		Queue Size			x
		Execution Time			x
Connector	*Con.*	Network Typology			x
		Bandwidth (Upload)			x
		Bandwidth (Download)			x
	Var.	Message size (incoming)			x
		Message size (outgoing)			x
		Message rate (incoming)			x
		Message rate (outgoing)			x
		RTT			x
Topology	*Con.*	Hardware Dependencies	x	x	x
		Whitelist	x	x	x
		Blacklist	x	x	x
	Var.	Delay			
		Throughput			

Algorithm 1: Peer Ranking for Topology Initialization

```
Data: Known Peers, Topology
Result: Peers Ranking
P ← Known Peers ;
foreach Operator o in Topology do
    foreach Peer p in P do
        if !compatible(o, p) then
            P ← P \ p ;
        end
    end
end
if P = ∅ then
    return cannot deploy Topology
else
    foreach Peer p in P do
        Poll p for its current metrics M_i(p);
        Compute ranking function r(p) according to Eq. (1);
    end
    return Sorted list of available Peers P according to their rank r(p)
end
```

$$r(p) = \sum_{i=1}^{n} \alpha_i M_i(p)$$

Algorithm 1: Peer Ranking for Topology Initialization

Data: Known Peers, Topology
Result: Peers Ranking
$P \leftarrow$ Known Peers ;
foreach *Operator o in Topology* **do**
  foreach *Peer p in P* **do**
    if $!compatible(o, p)$ **then**
      $P \leftarrow P \setminus p$;
    end
  end
end
if $P = \emptyset$ **then**
  return cannot deploy Topology
else
  foreach *Peer p in P* **do**
    Poll p for its current metrics $M_i(p)$;
    Compute ranking function $r(p)$ according to Eq. (1);
  end
  return Sorted list of available Peers P according to their rank $r(p)$
end

Filter Peers

$$r(p) = \sum_{i=1}^{n} \alpha_i M_i(p)$$

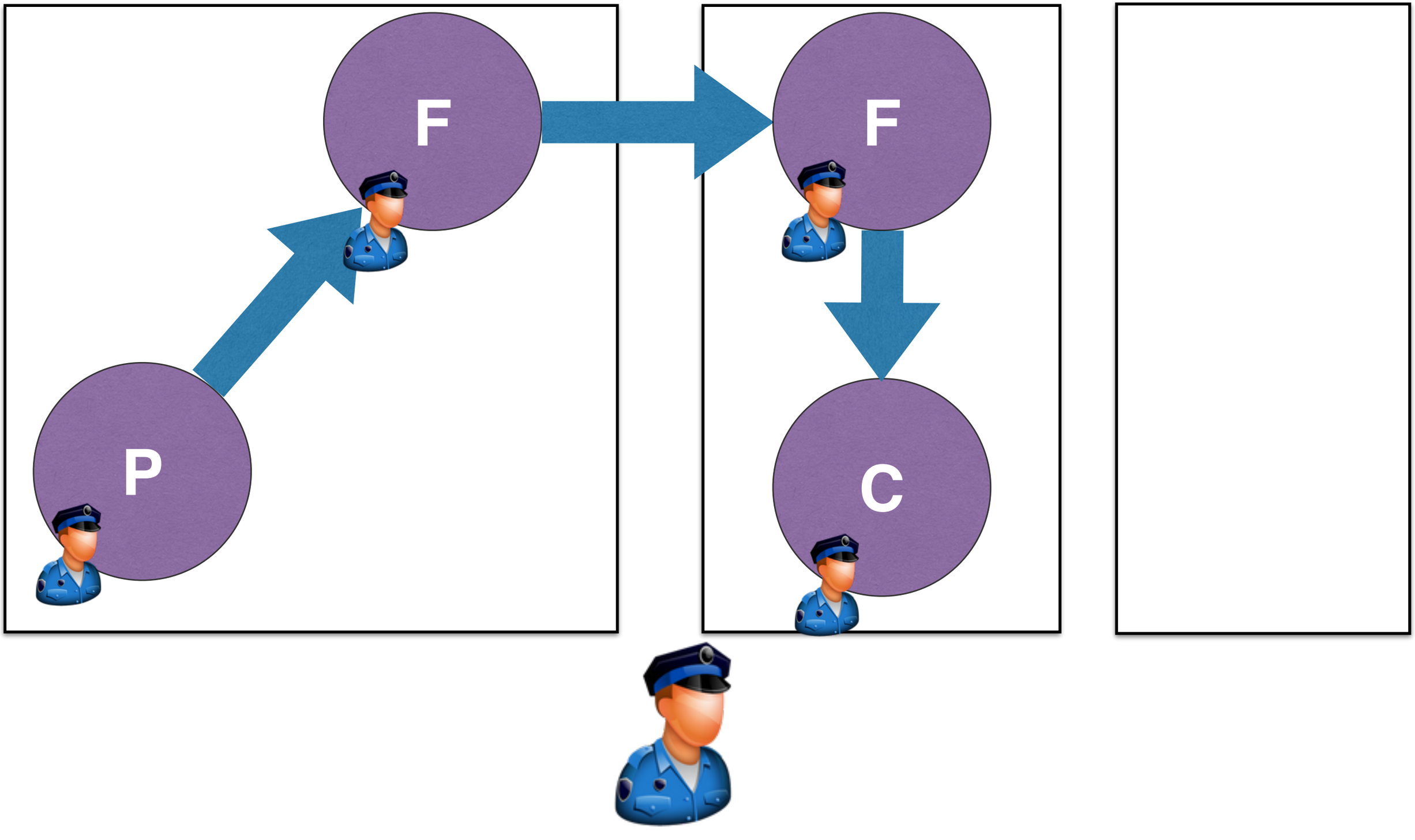
Algorithm 1: Peer Ranking for Topology Initialization

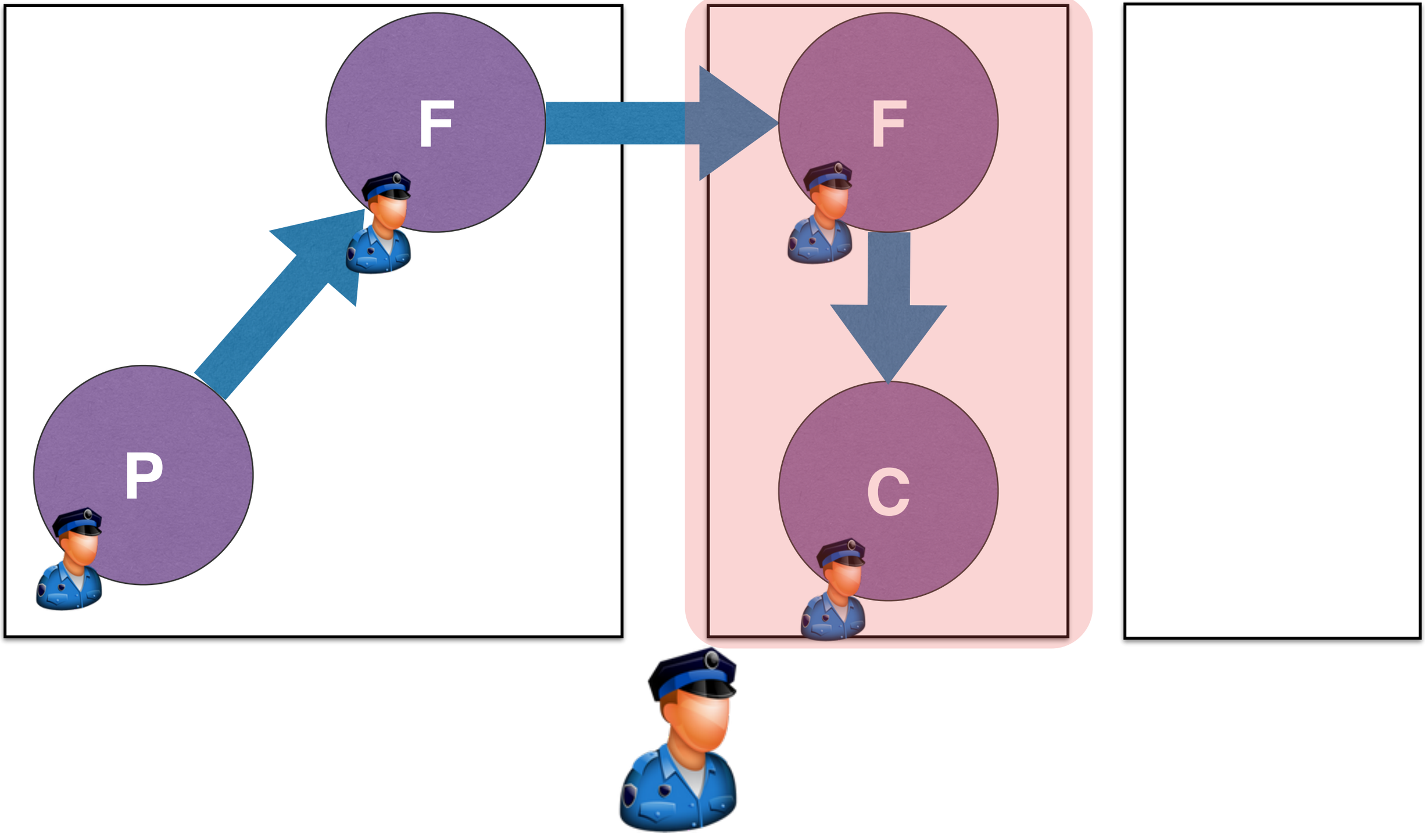
Data: Known Peers, Topology
Result: Peers Ranking
$P \leftarrow$ Known Peers ;
foreach *Operator o in Topology* **do**
  foreach *Peer p in P* **do**
    if $!compatible(o, p)$ **then**
      $P \leftarrow P \setminus p$;
    end
  end
end
if $P = \emptyset$ **then**
  return cannot deploy Topology
else
  foreach *Peer p in P* **do**
    Poll p for its current metrics $M_i(p)$;
    Compute ranking function $r(p)$ according to Eq. (1);
  end
  return Sorted list of available Peers P according to their rank $r(p)$
end

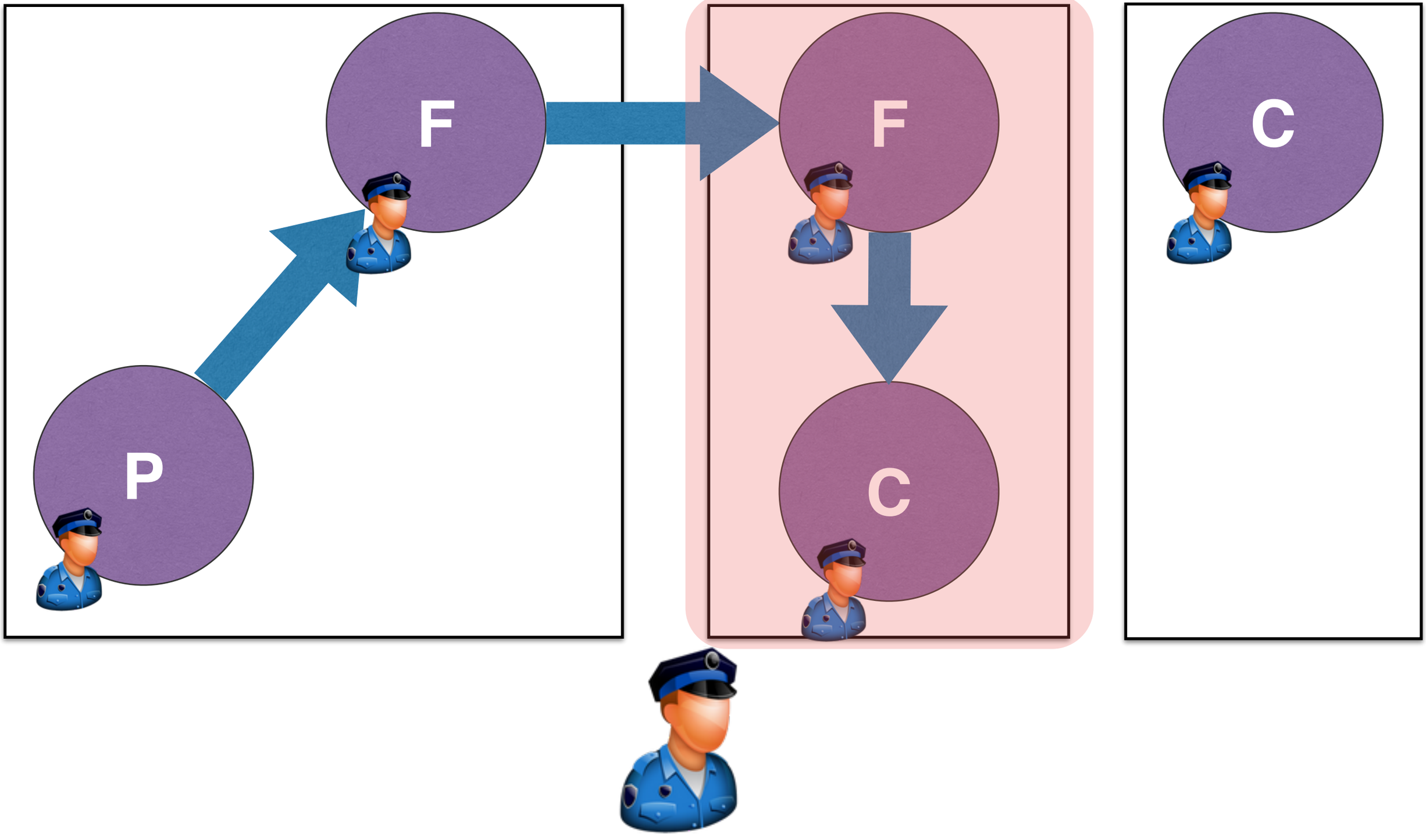
Filter Peers

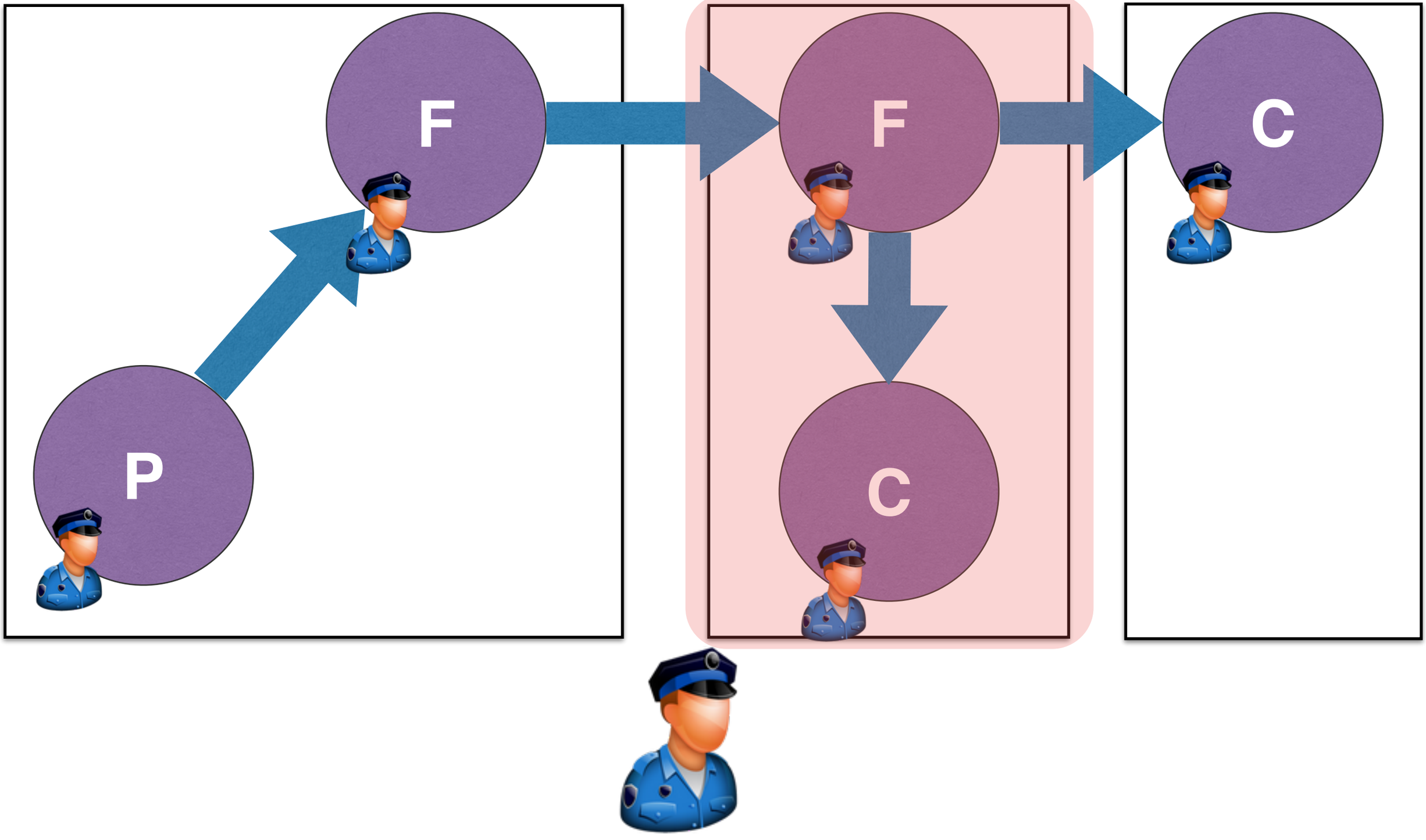
Compute the ranking

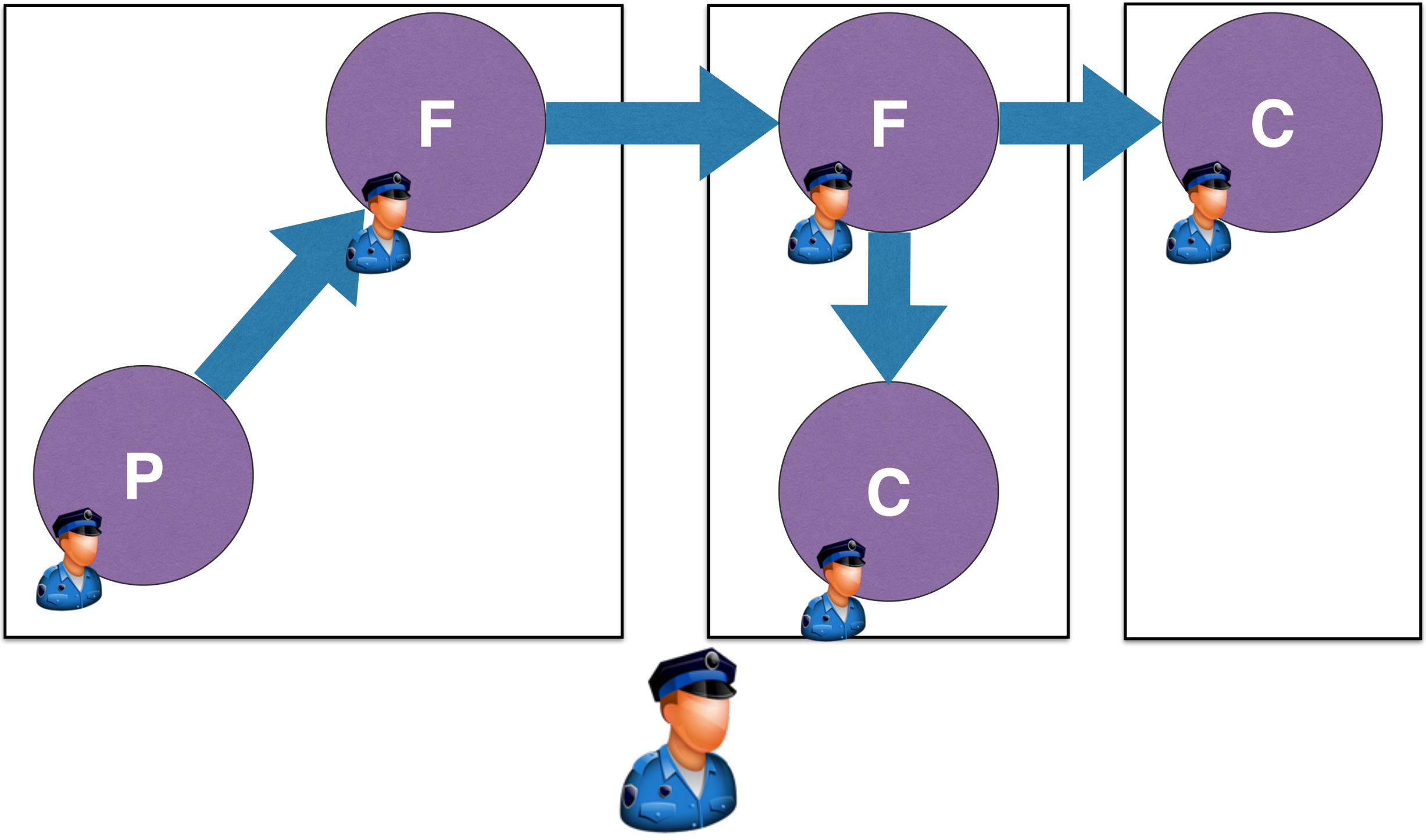
$$r(p) = \sum_{i=1}^{n} \alpha_i M_i(p)$$

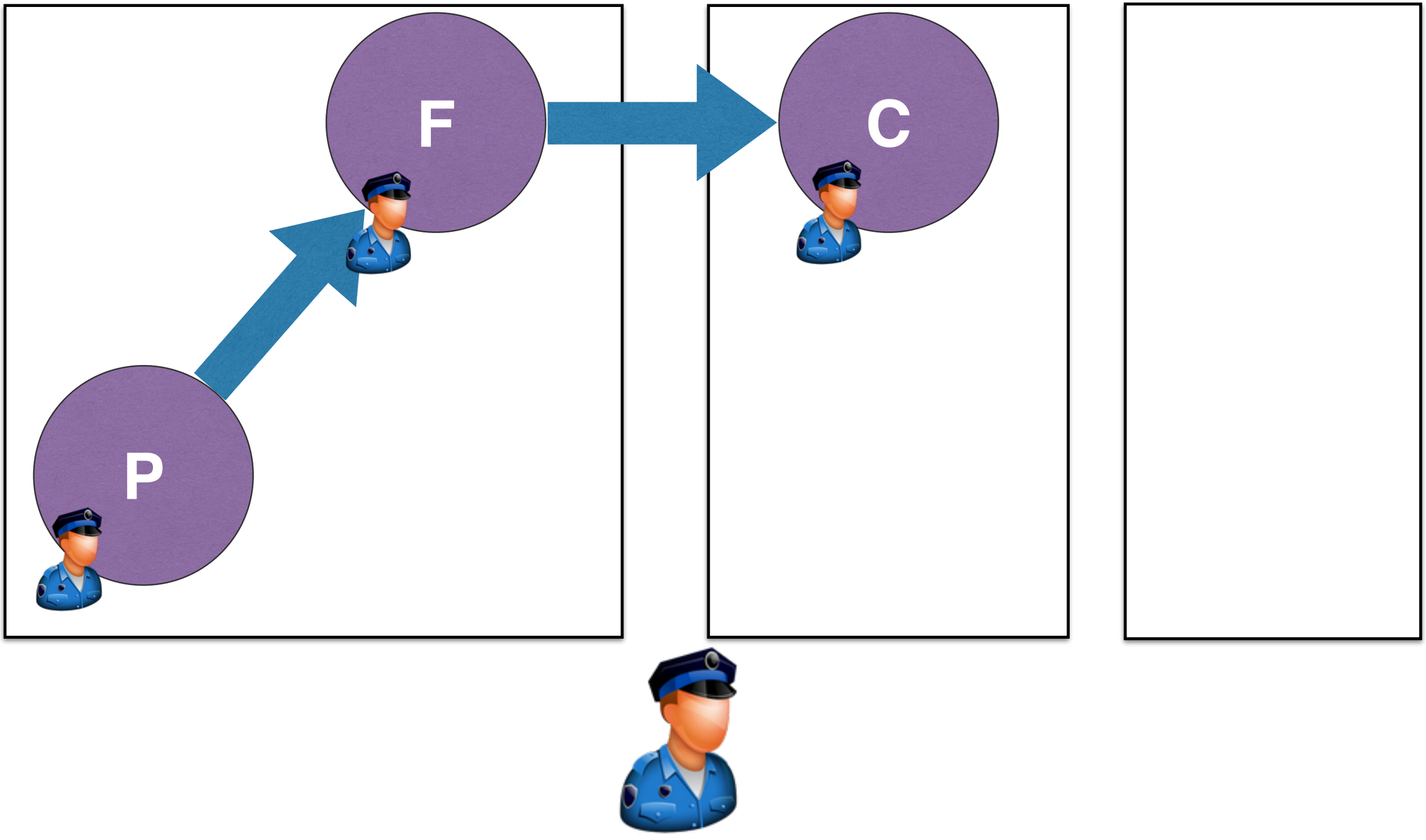

F
F
P
C

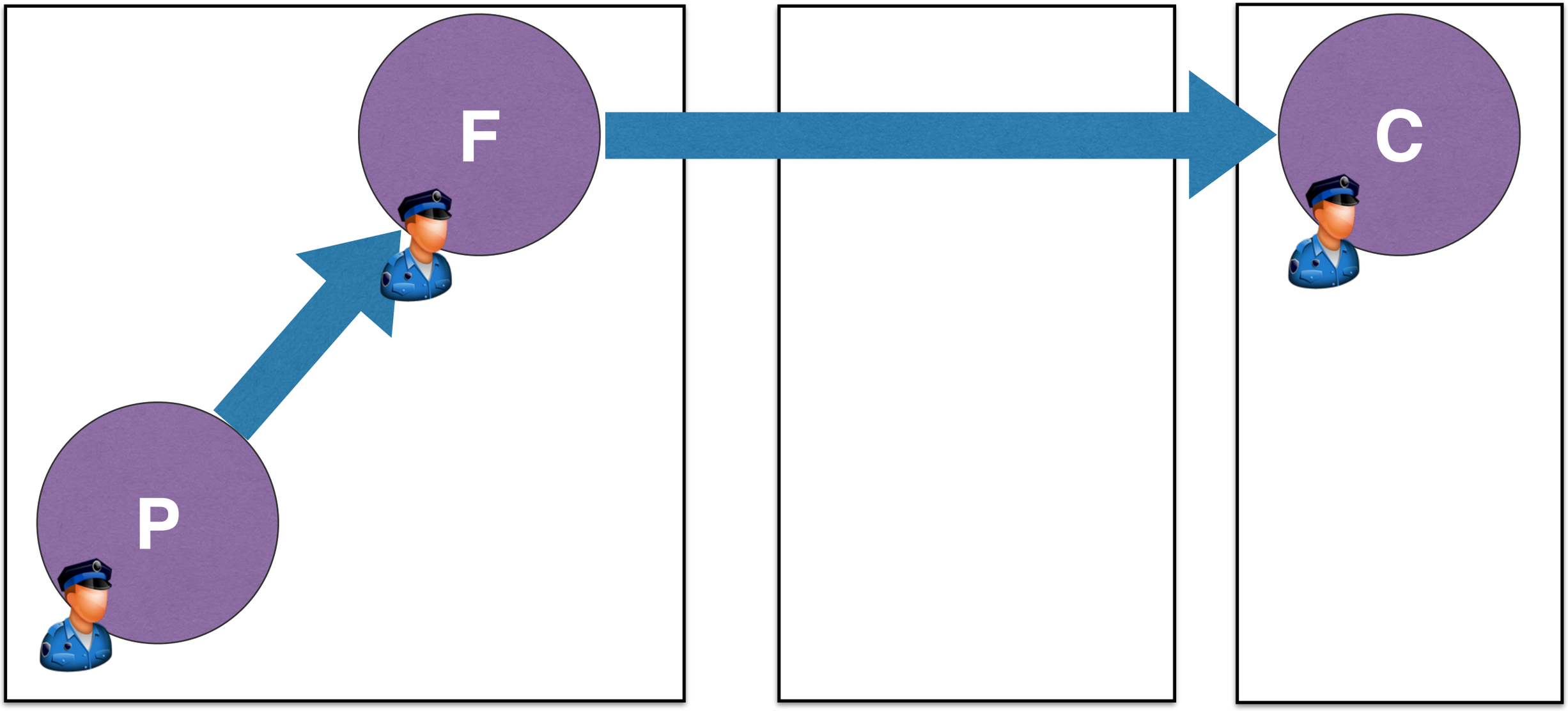

F
F
P
C

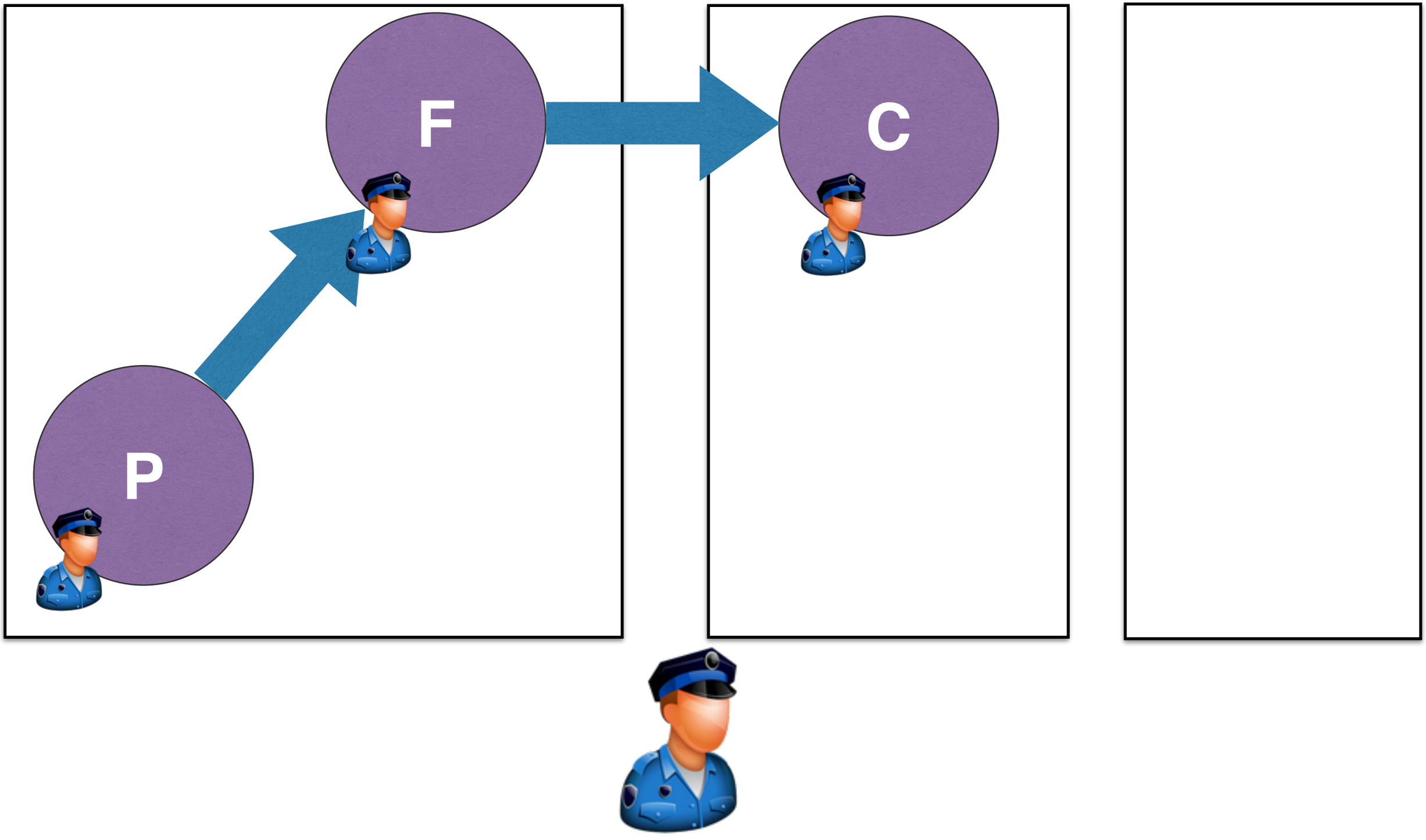

F
P
F
C
C

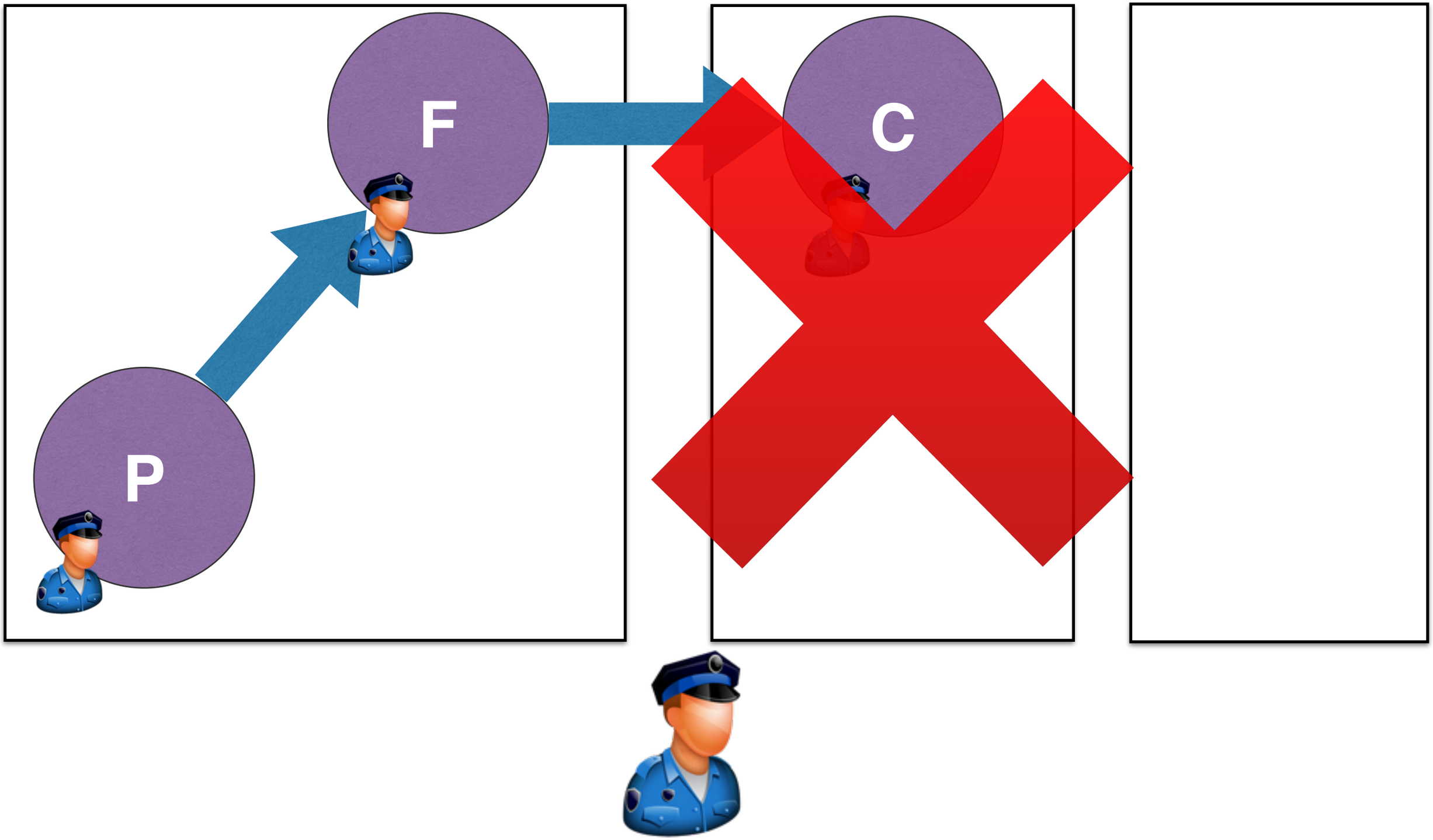

F
F
C
P
C

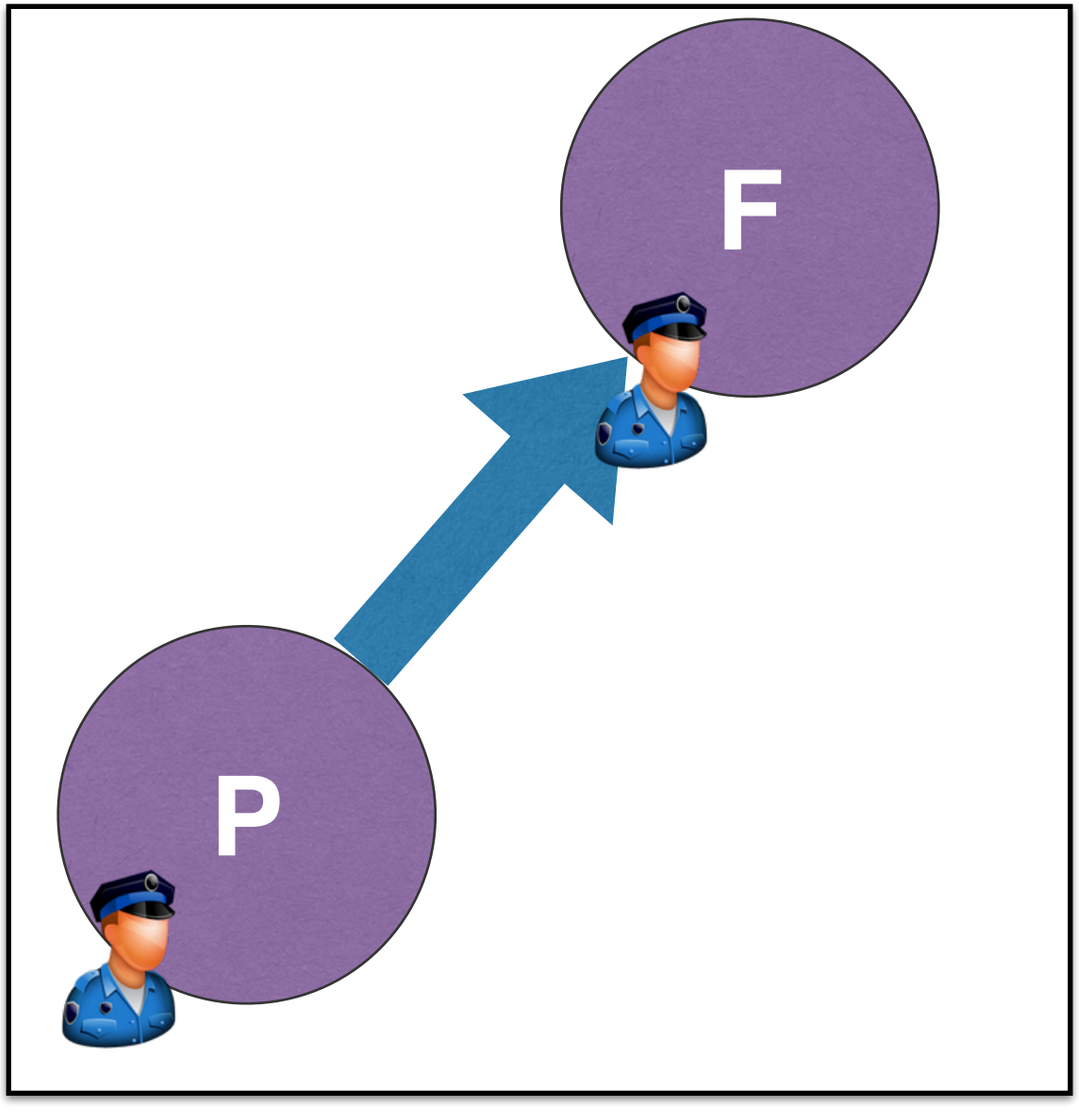

F
F
C
P
C

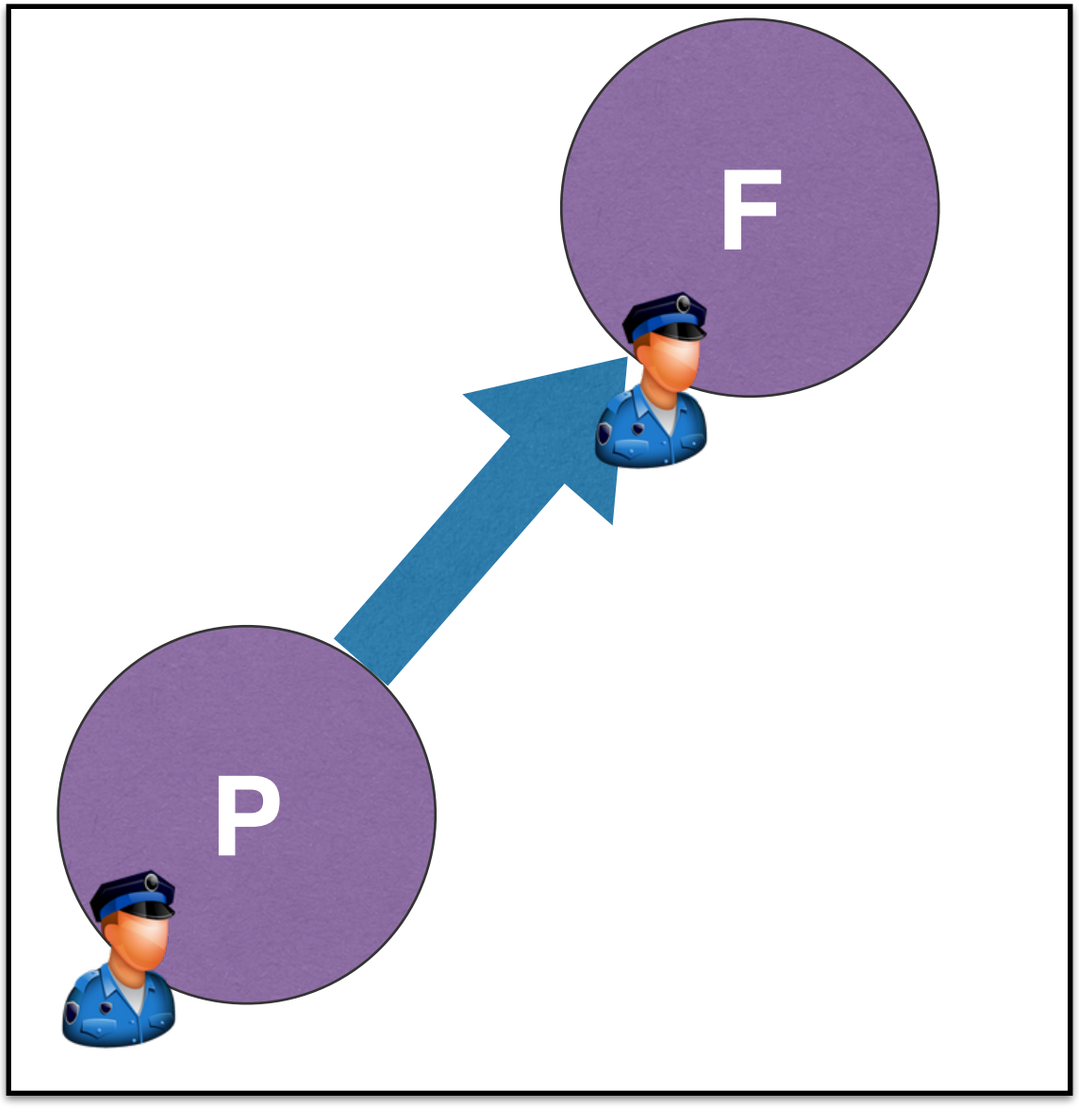

F
C
P

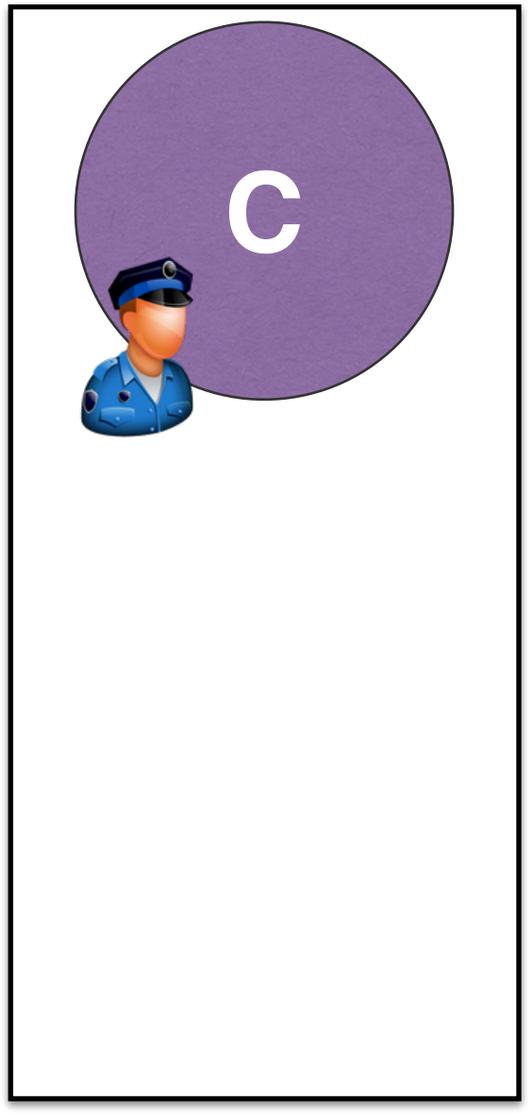

F
C
P

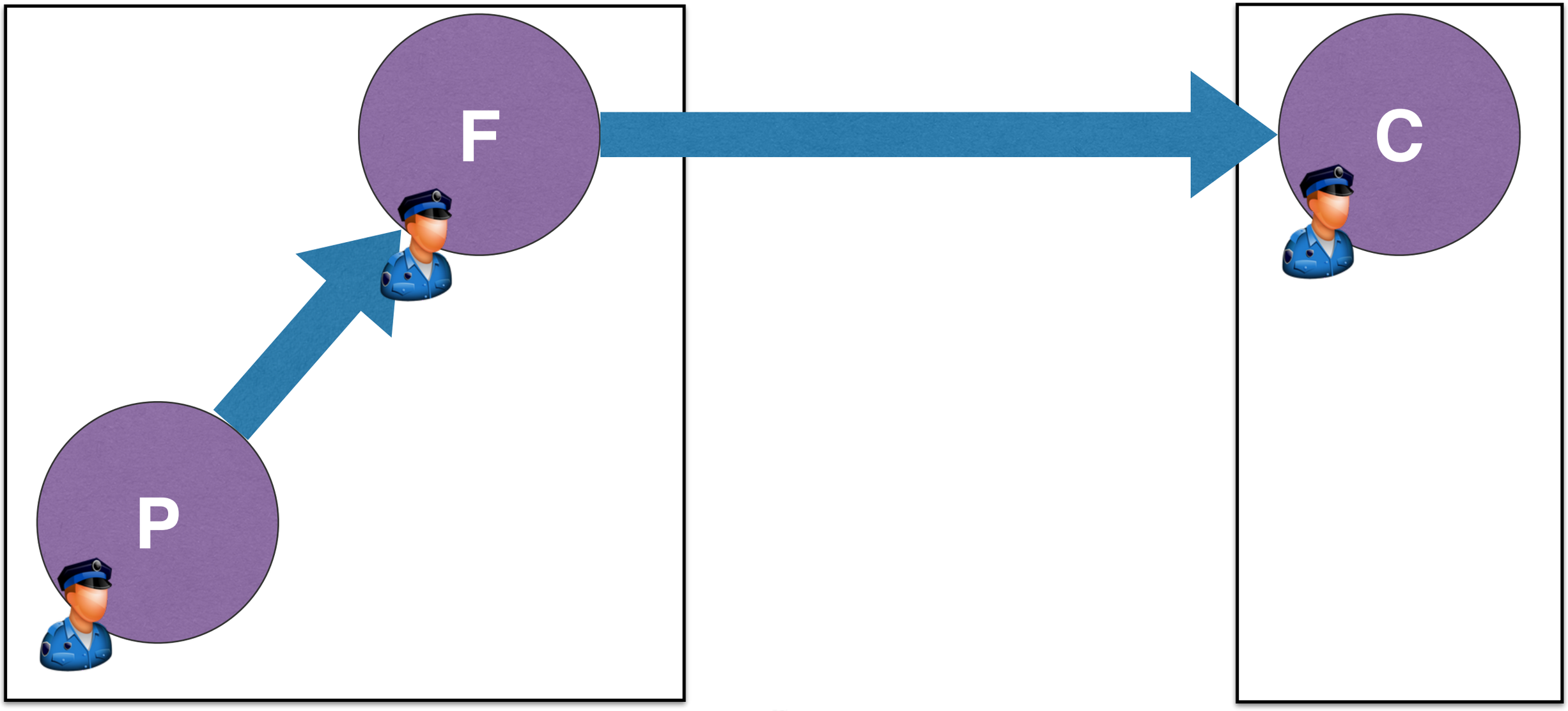

F
C
P


F
C
P

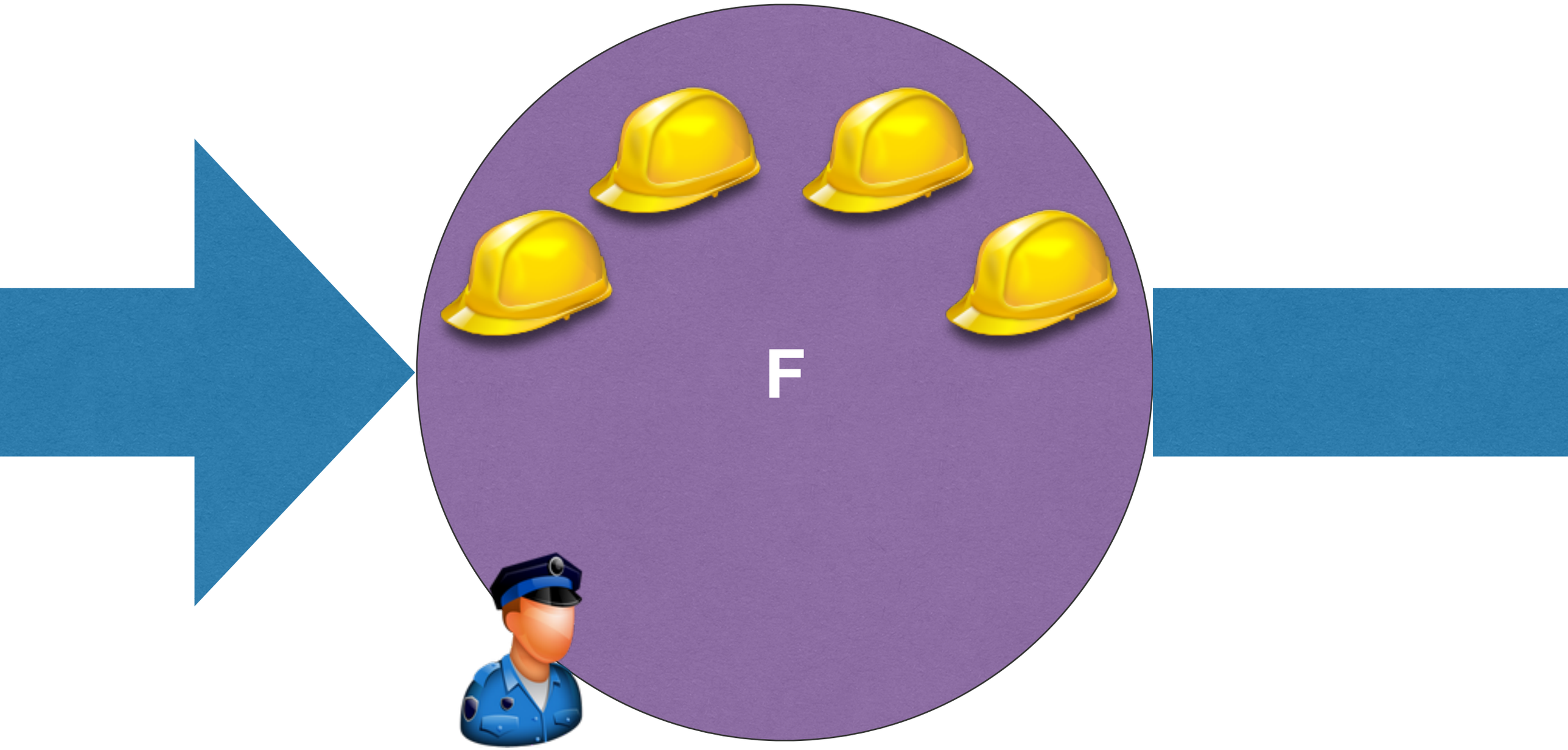

F
P


F
P

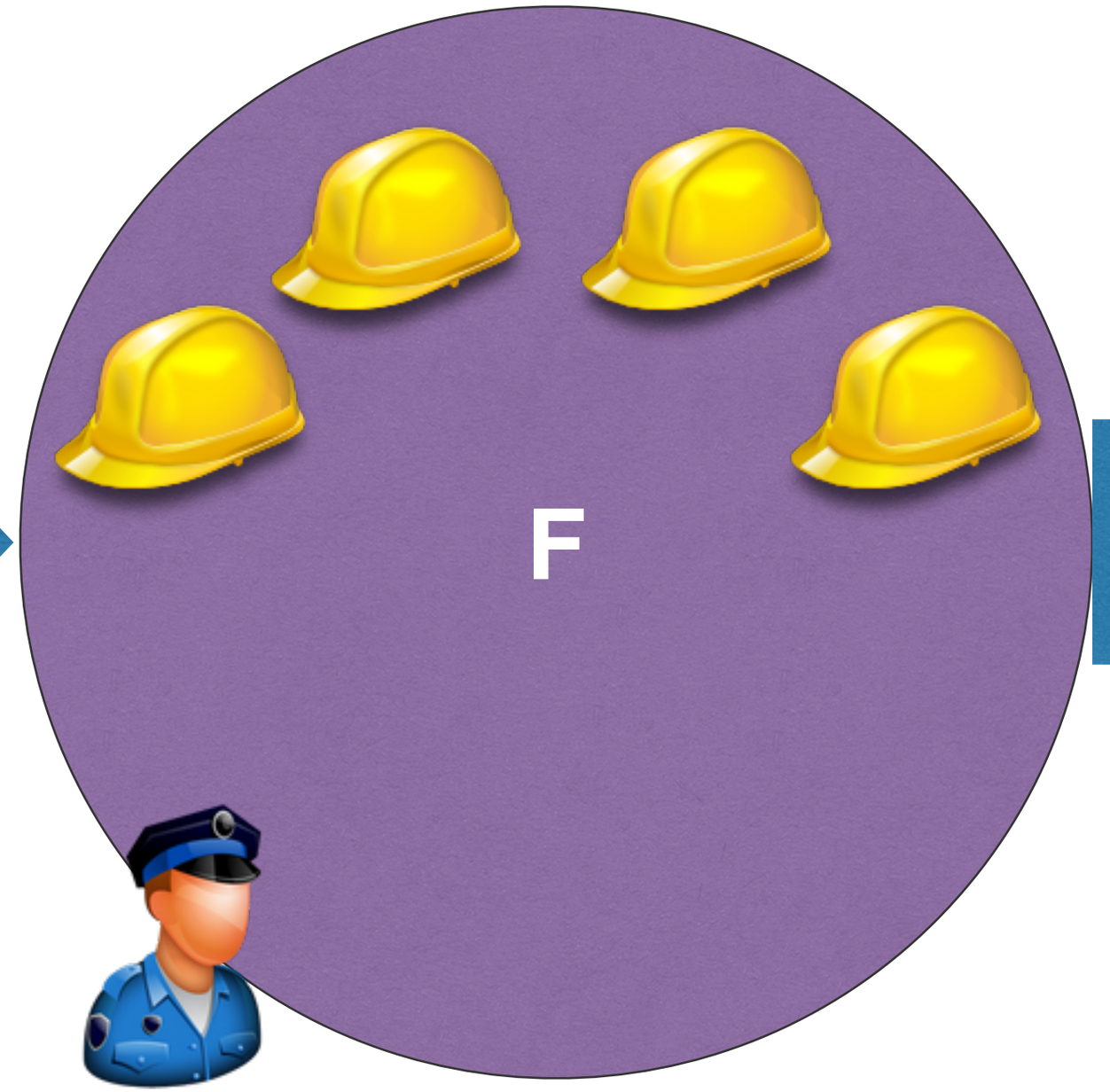

C


F
P
C

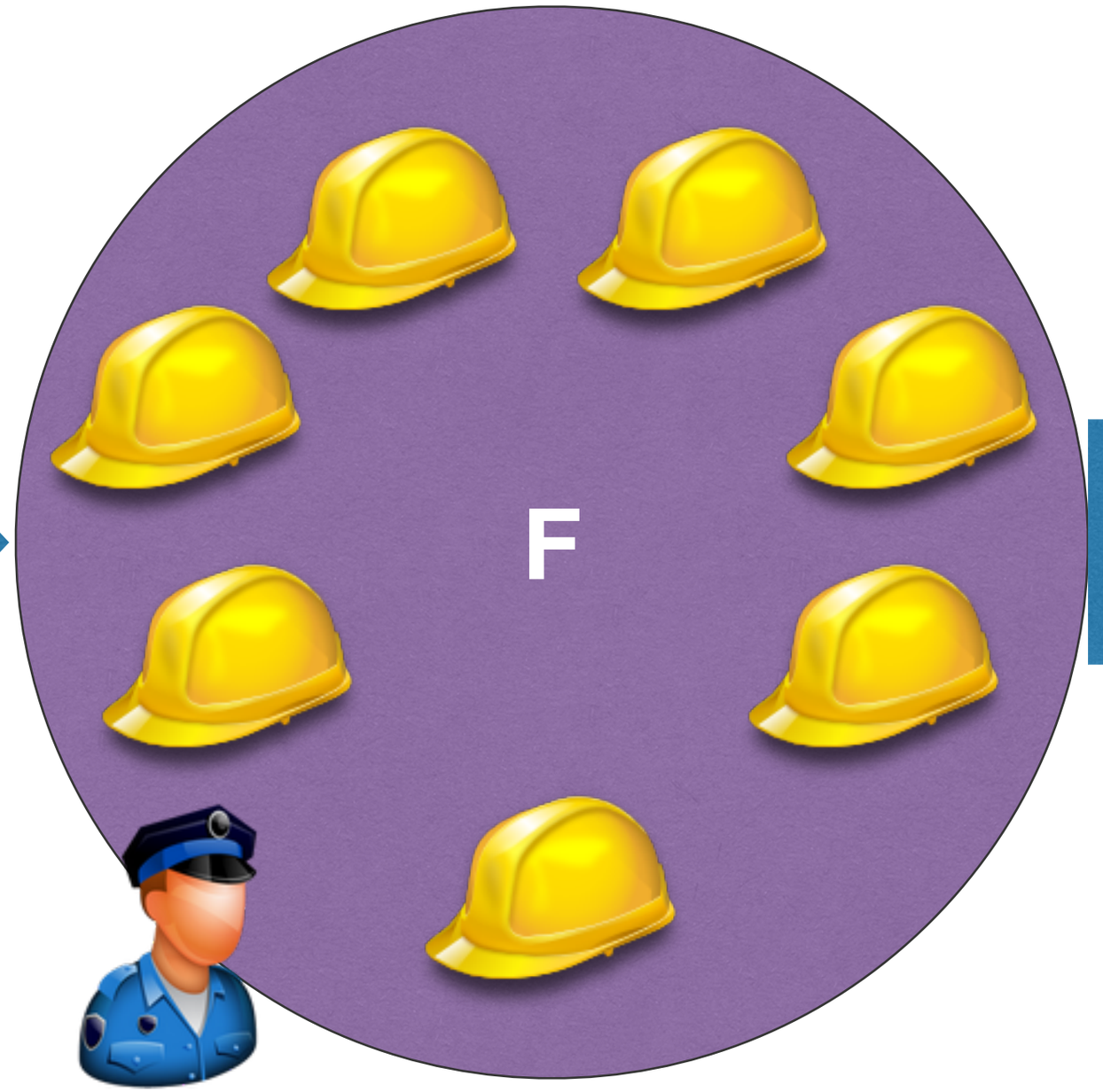
Operator Elasticity



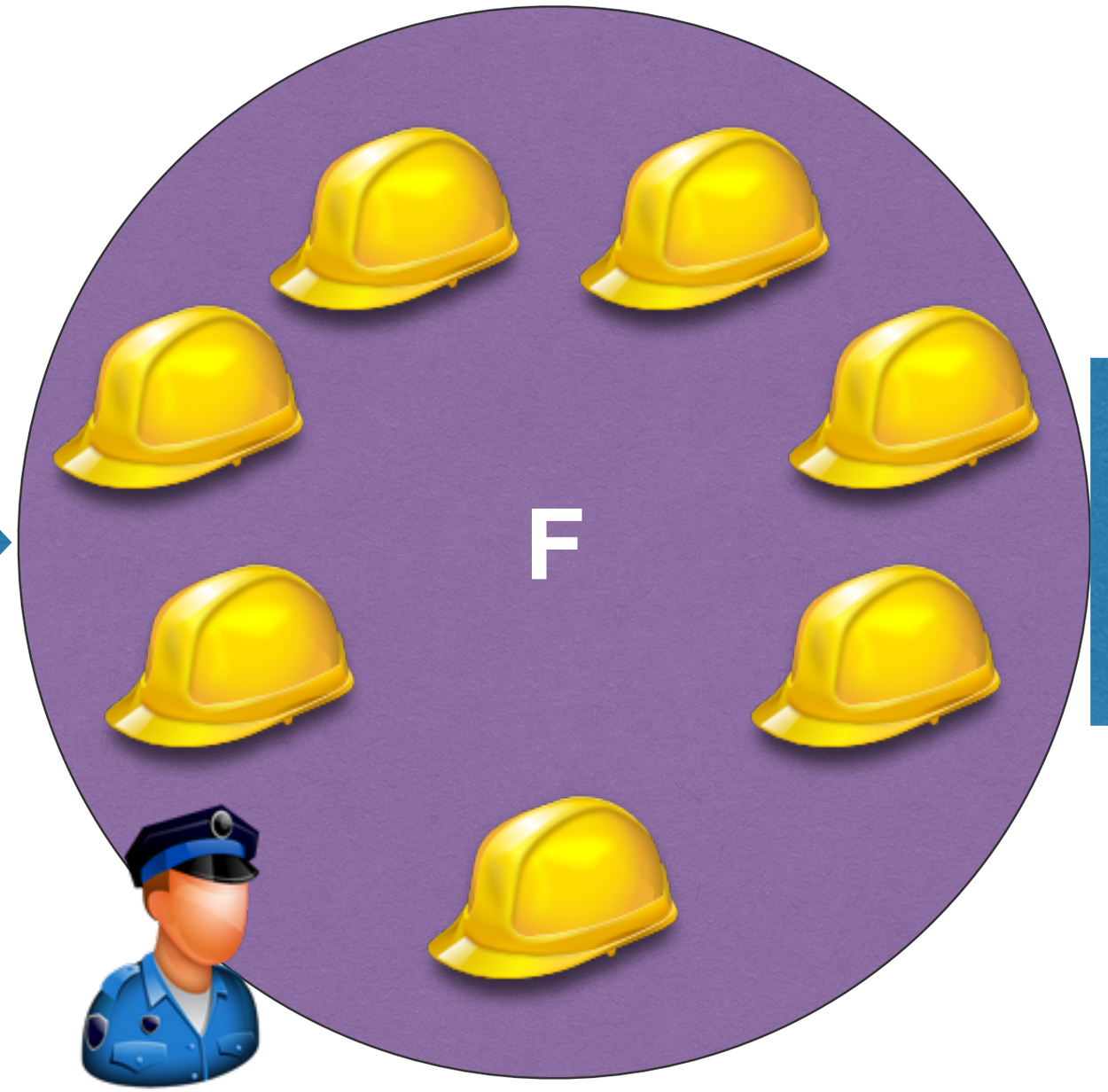
Operator Elasticity



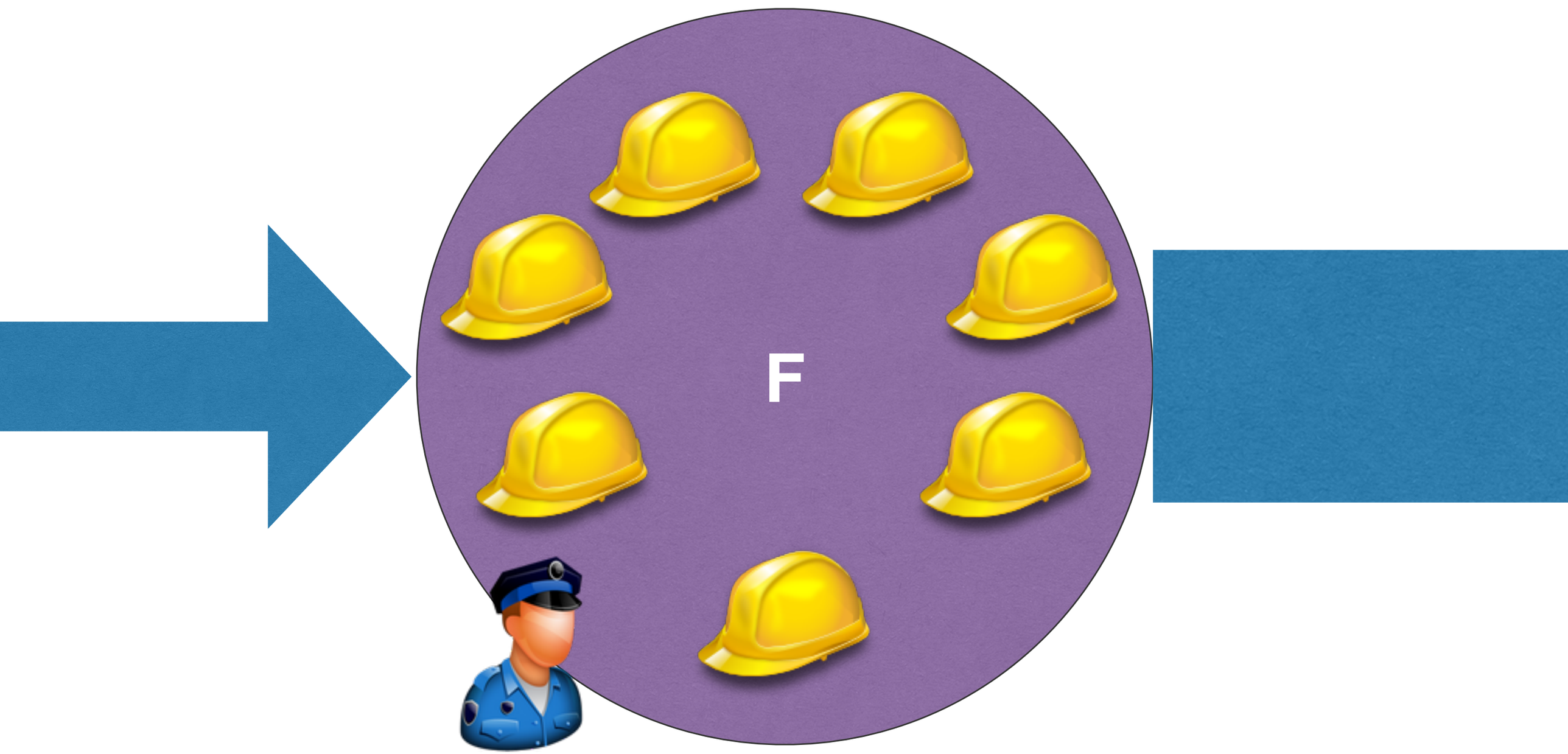
Operator Elasticity



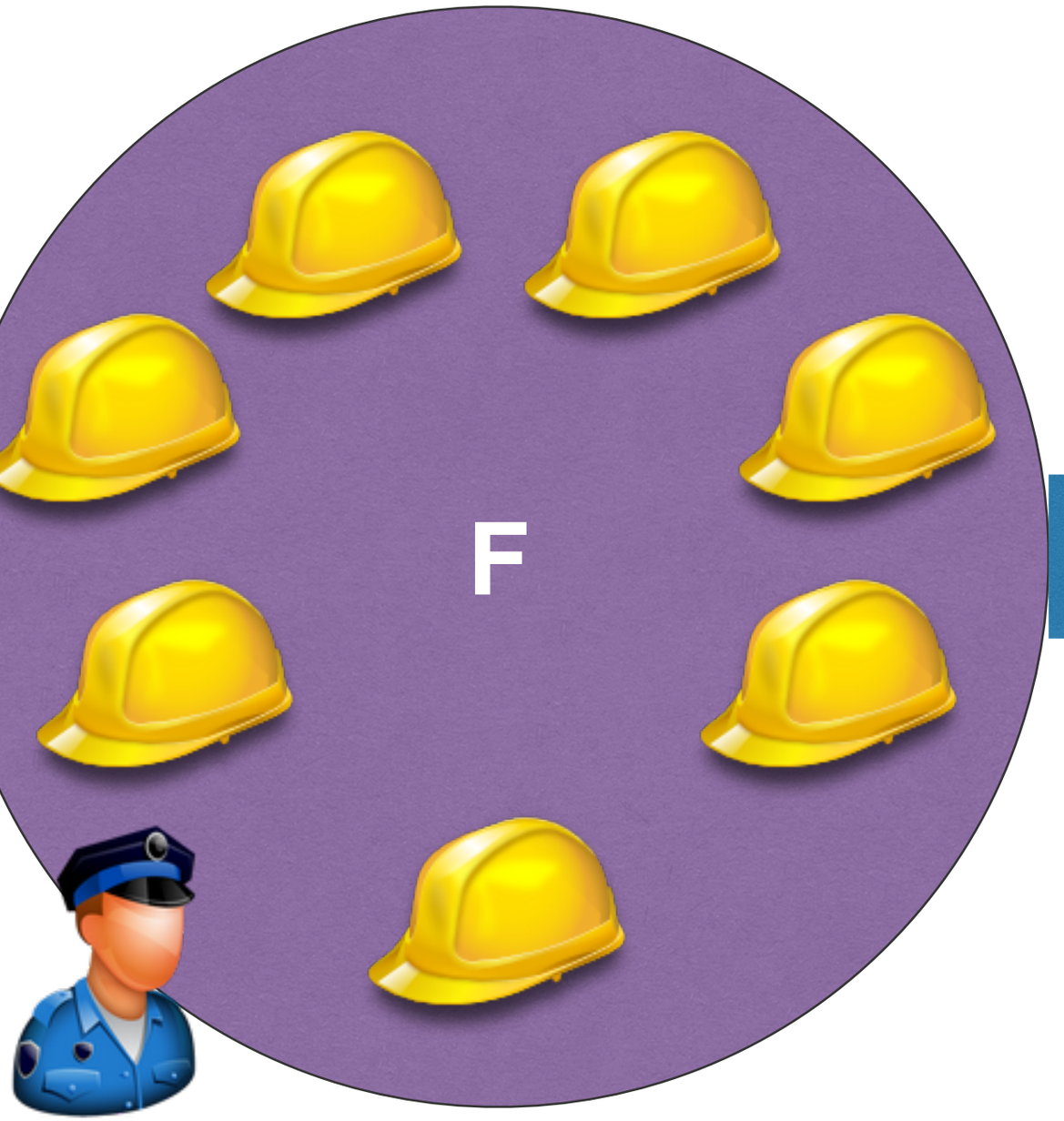
Operator Elasticity



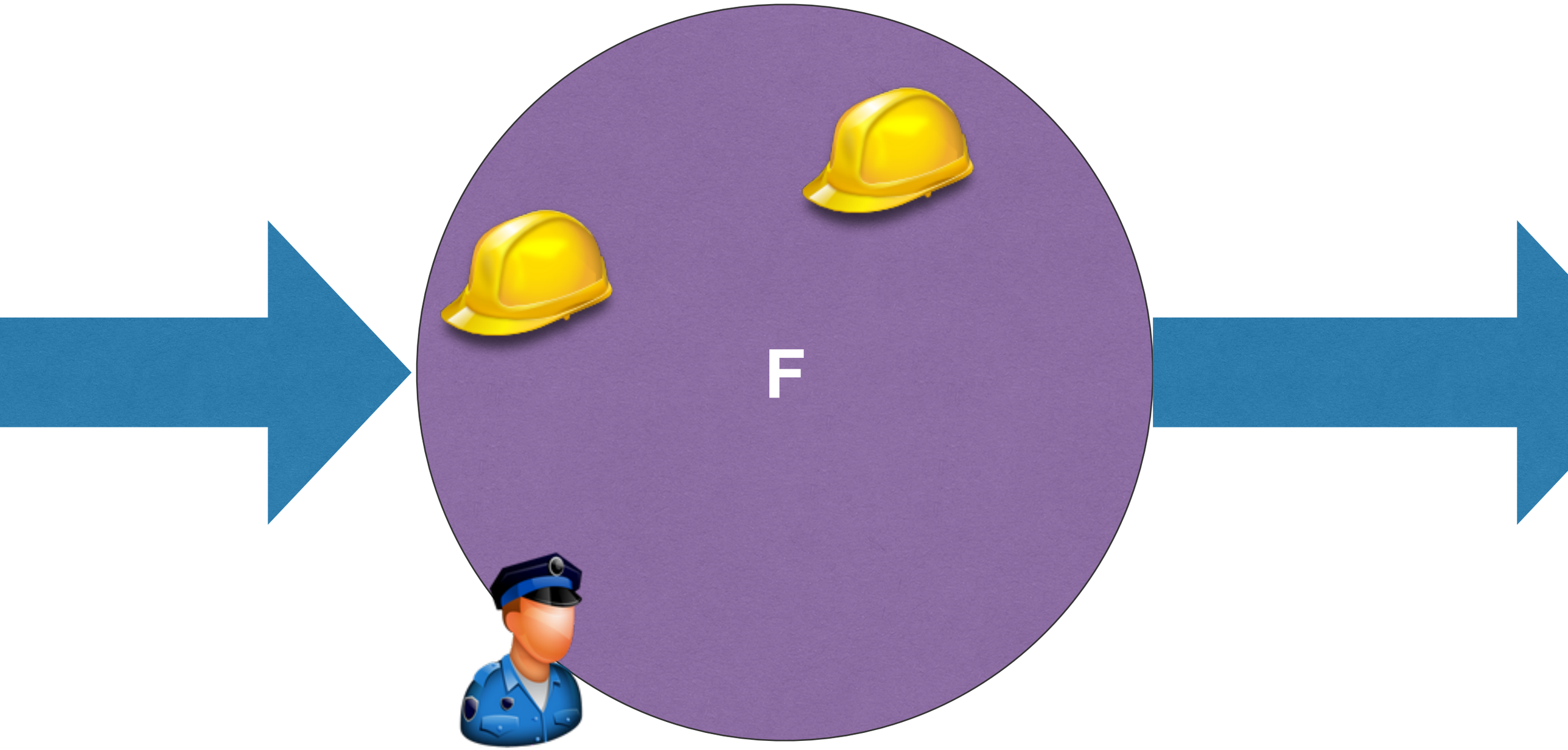
Operator Elasticity



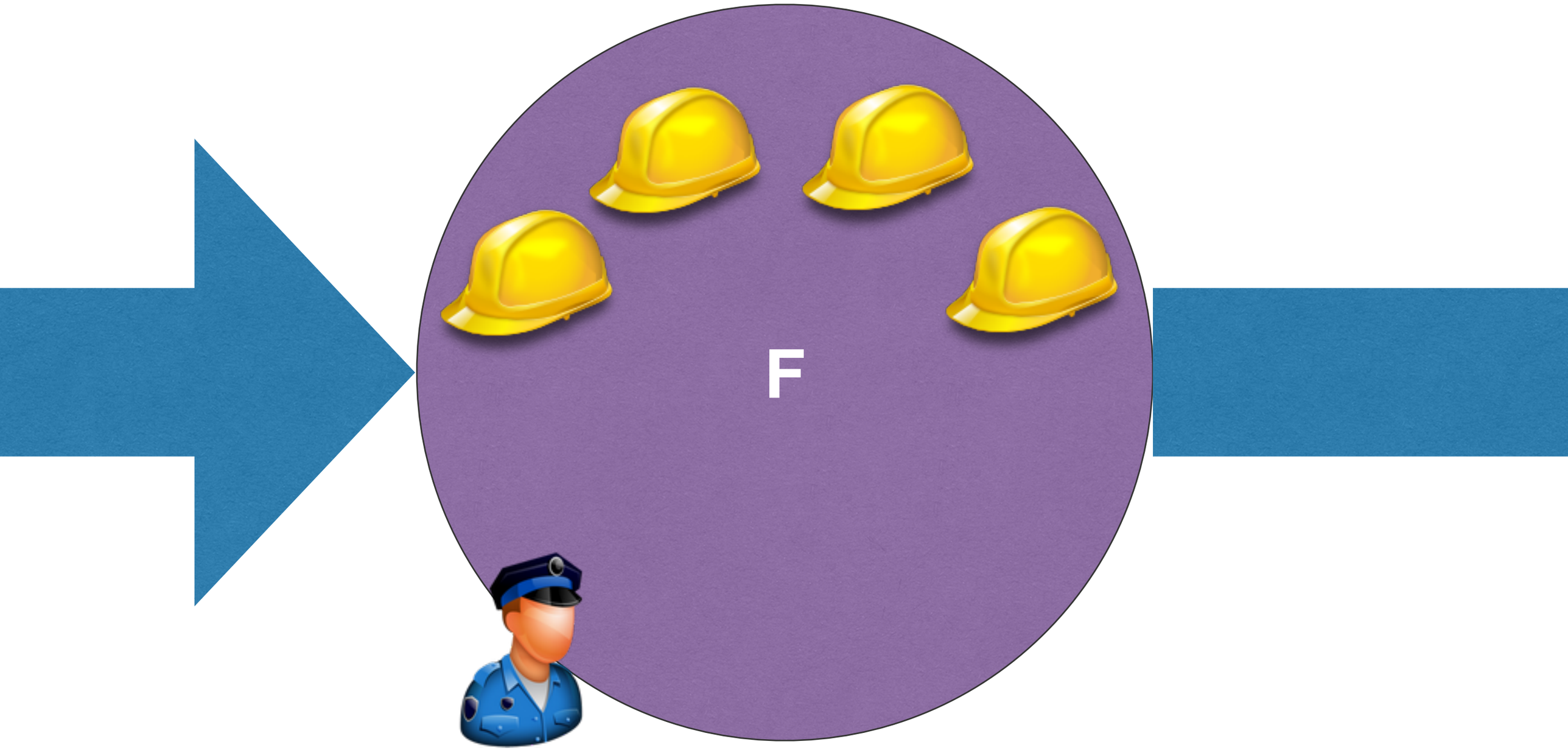
Operator Elasticity



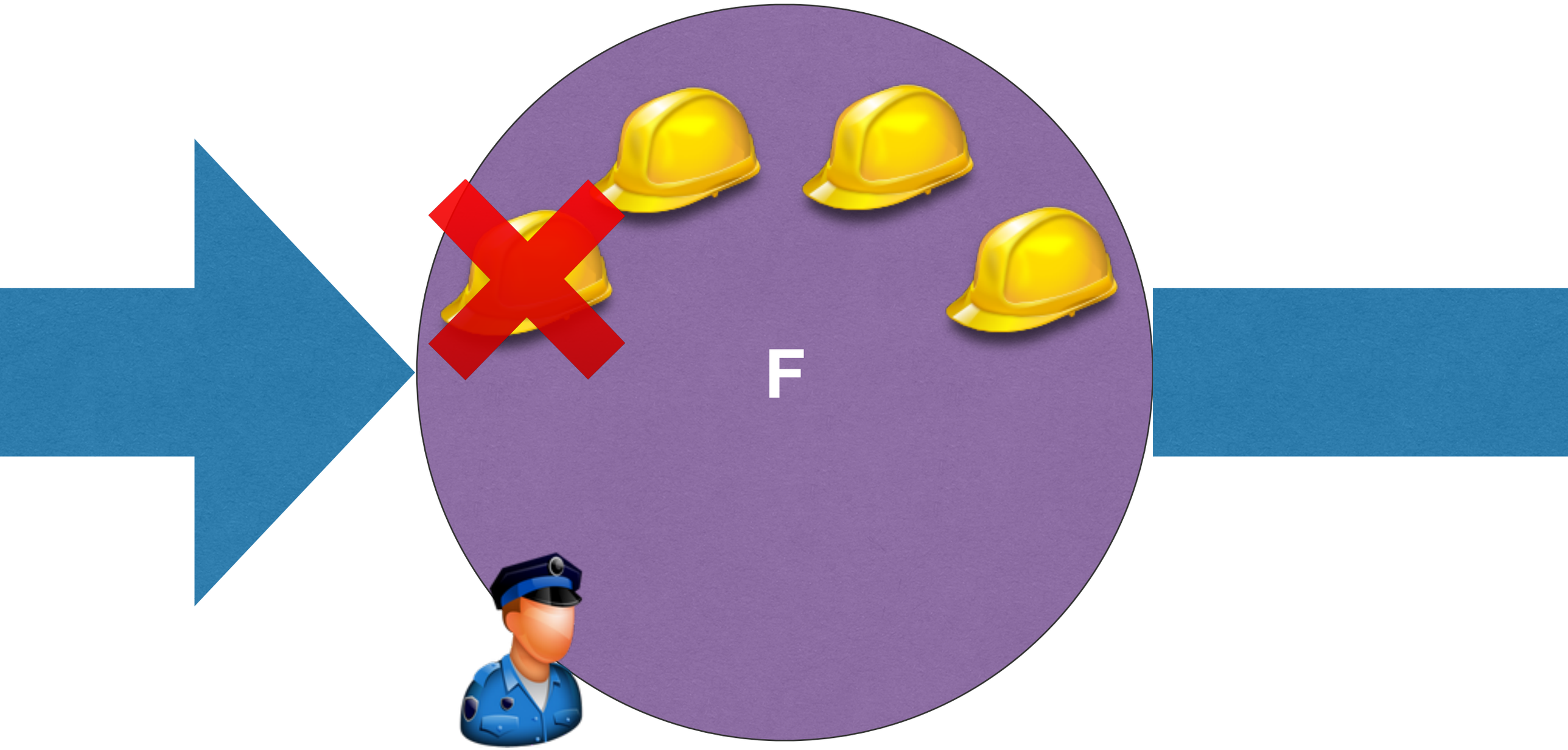
Operator Elasticity



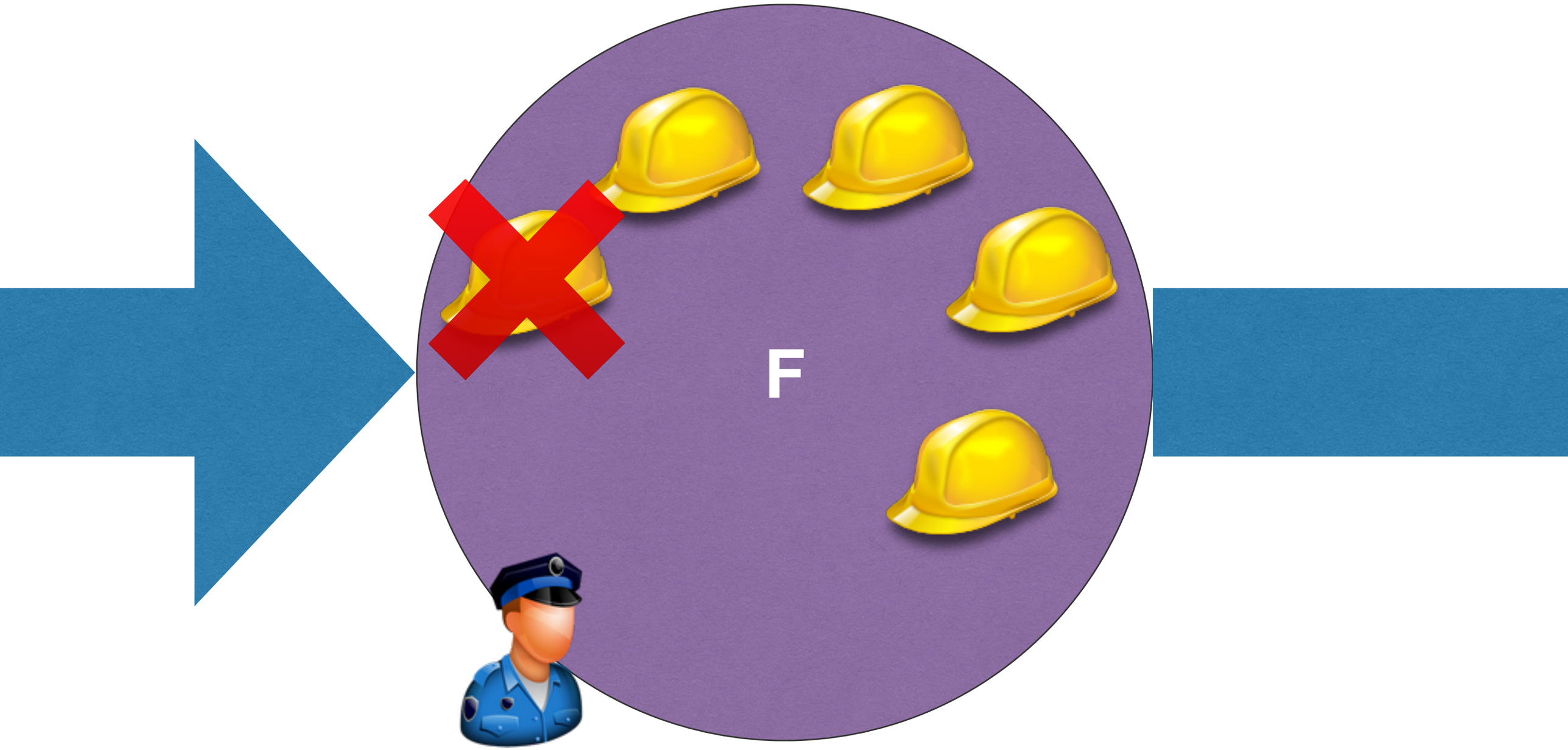
Fault Tolerance



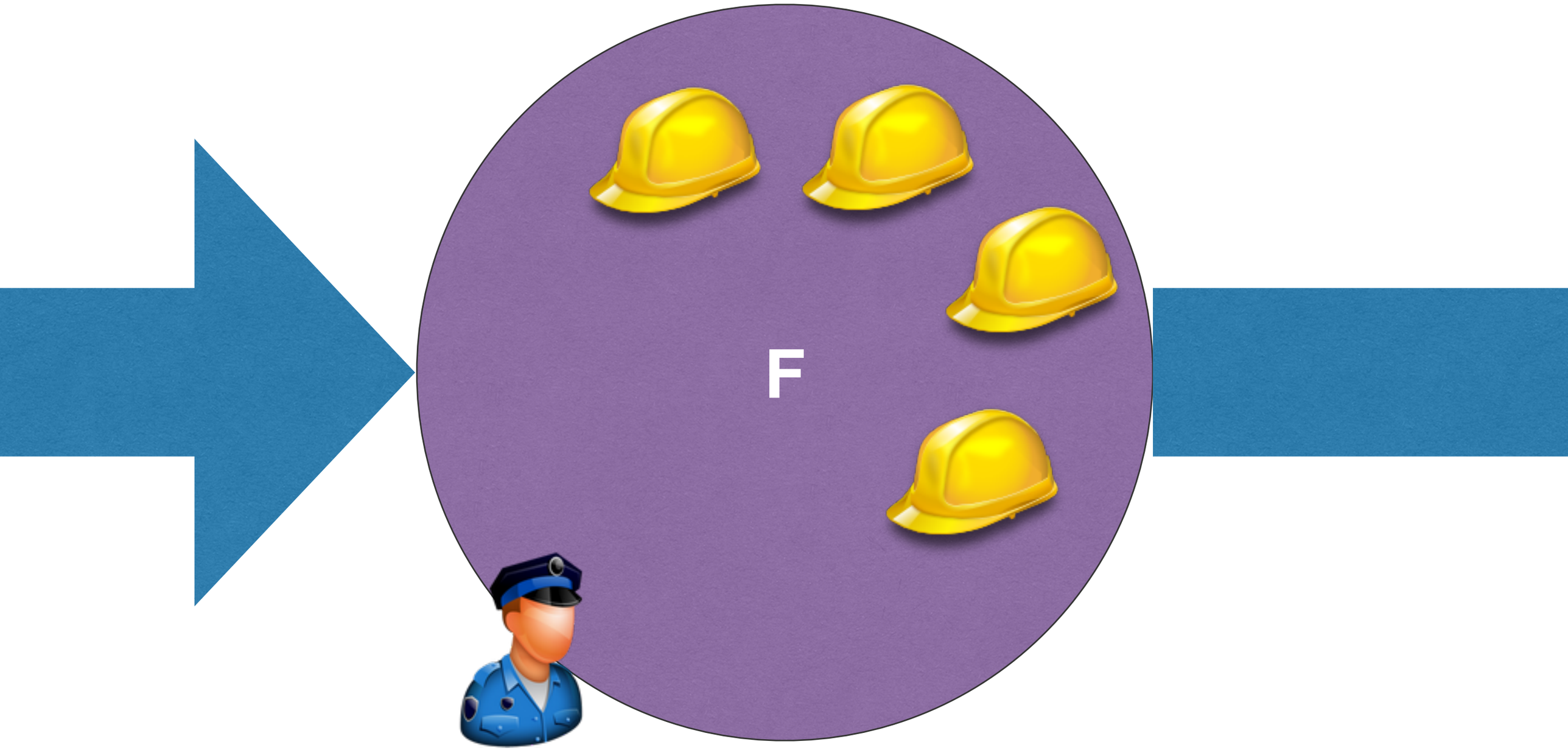
Fault Tolerance

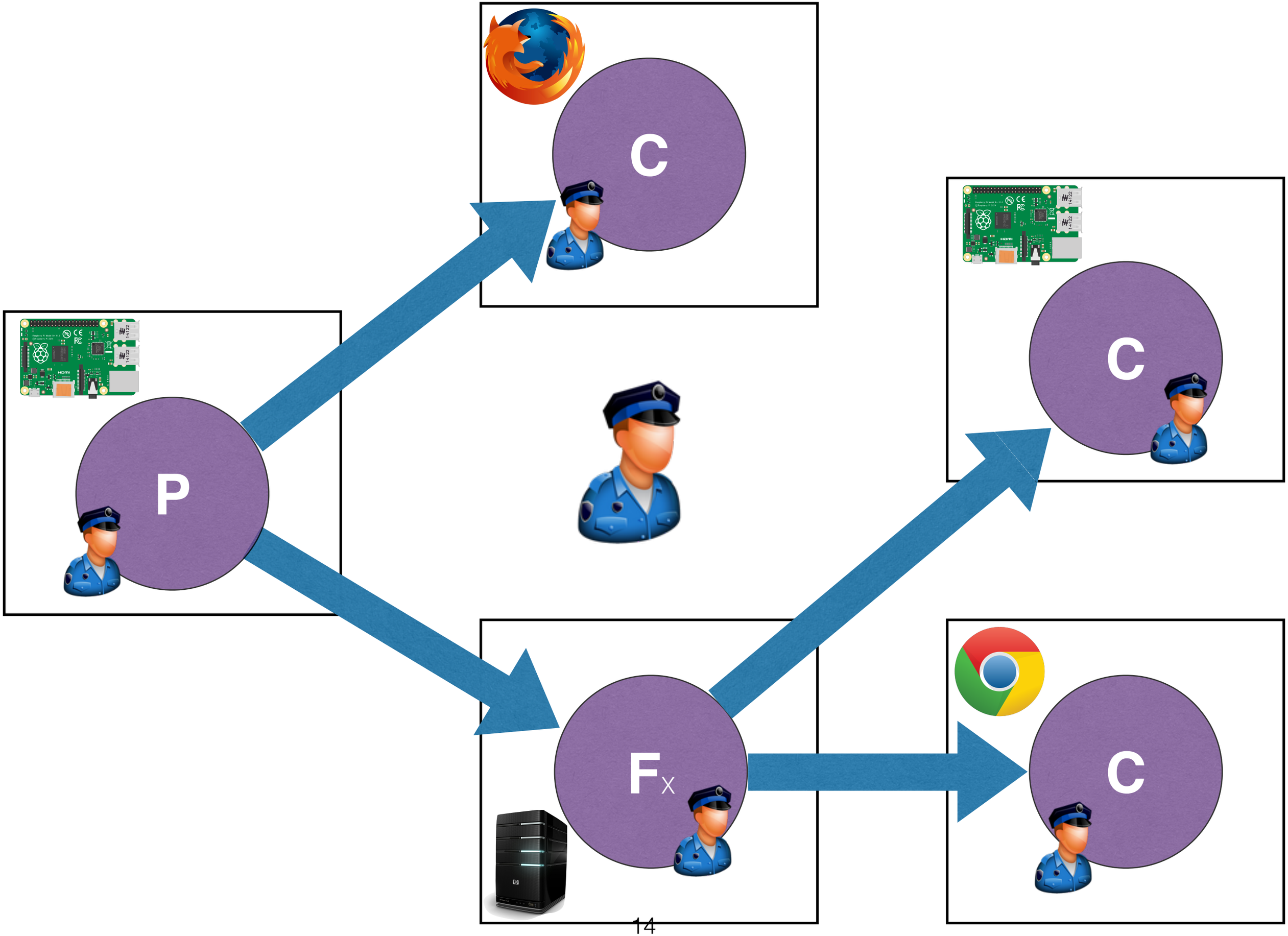


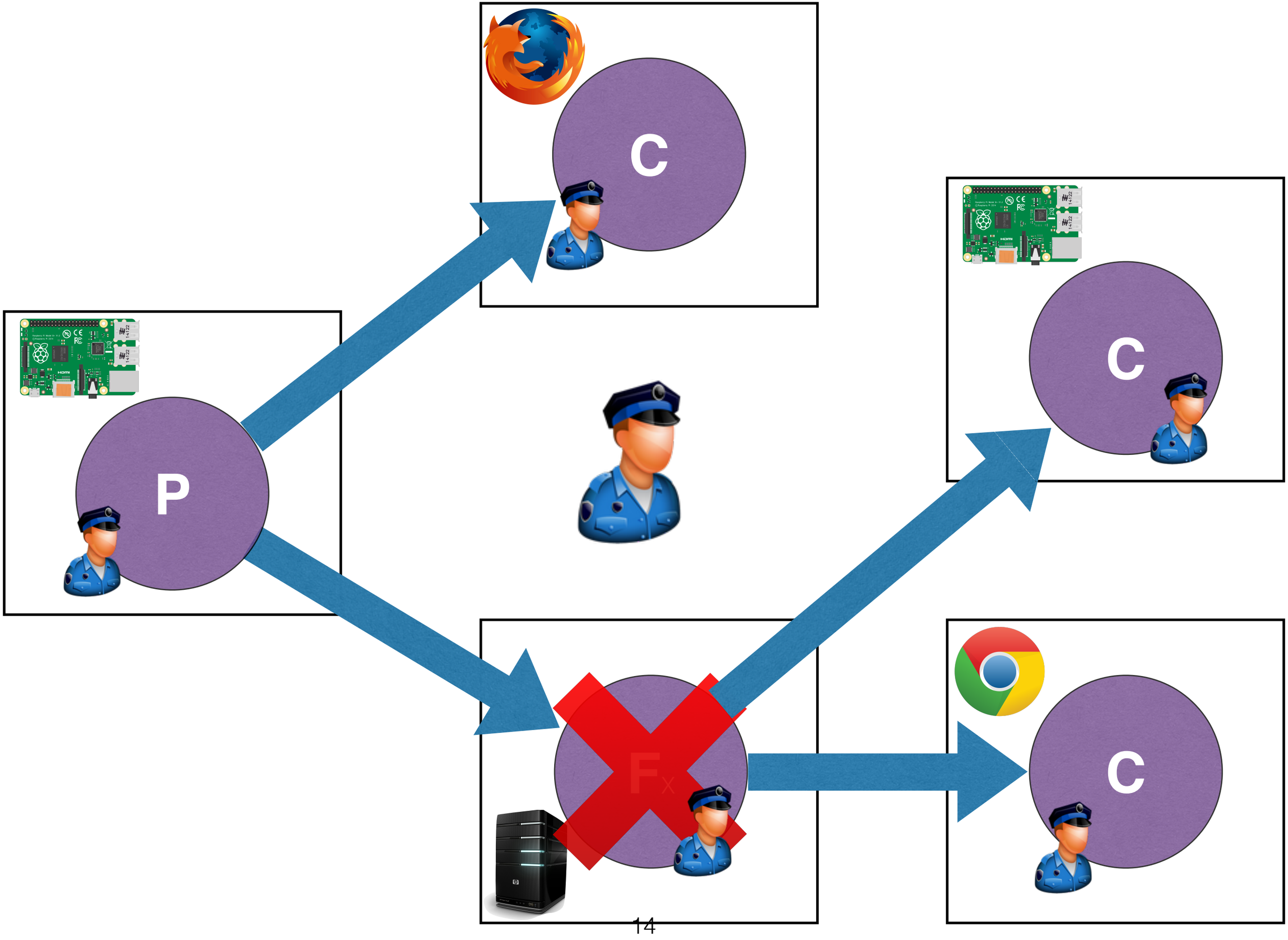
Fault Tolerance

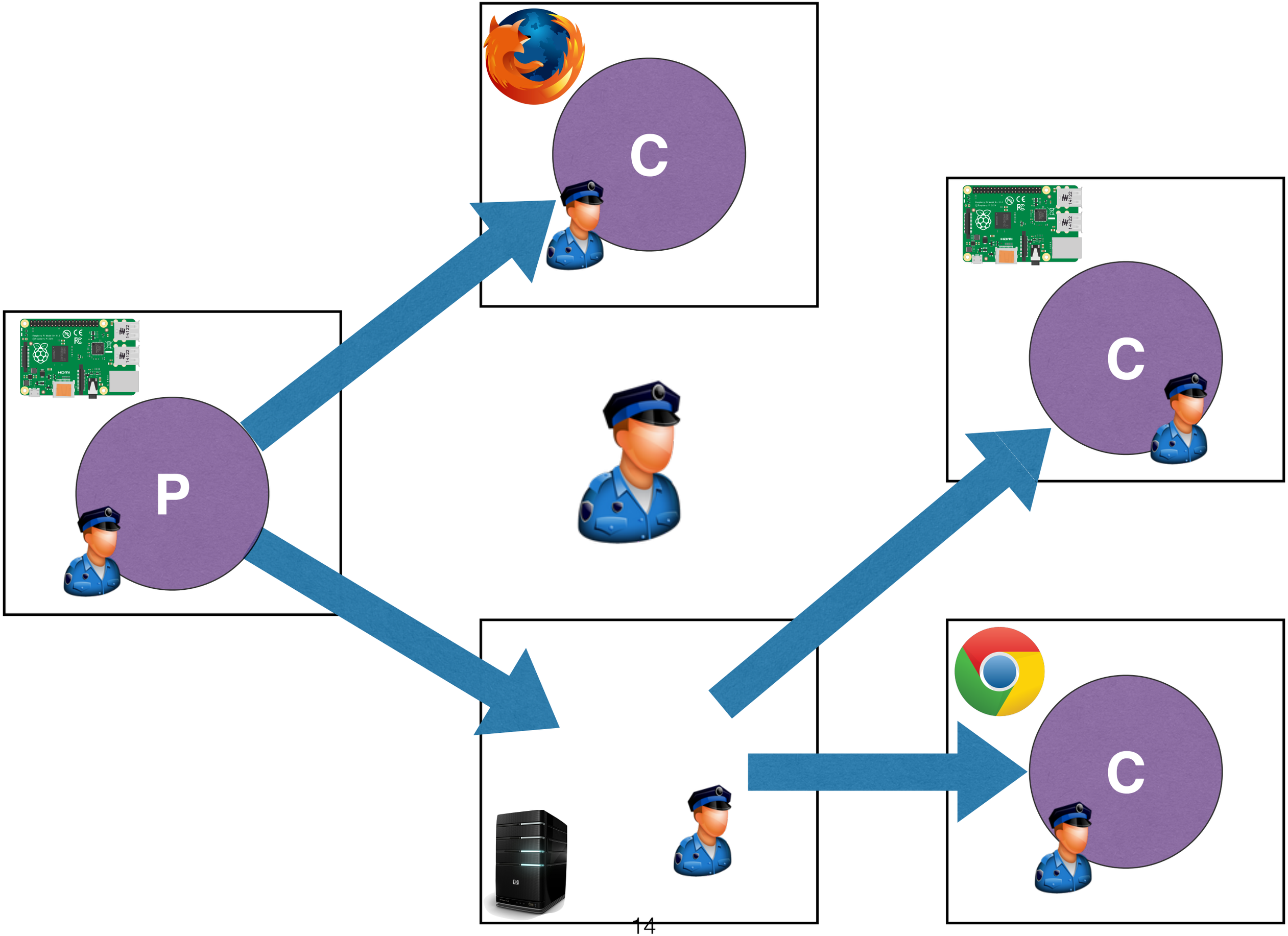


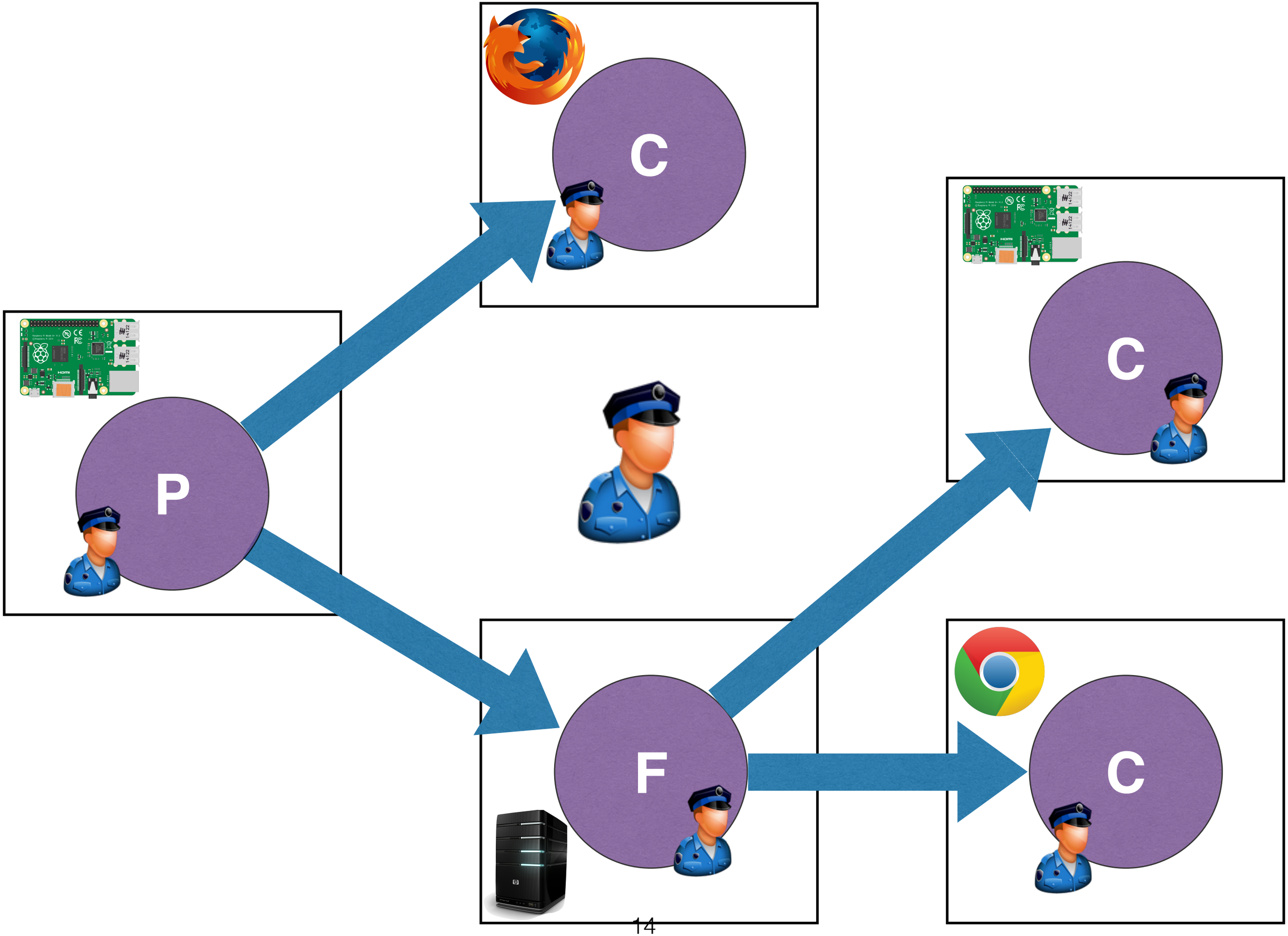
Fault Tolerance

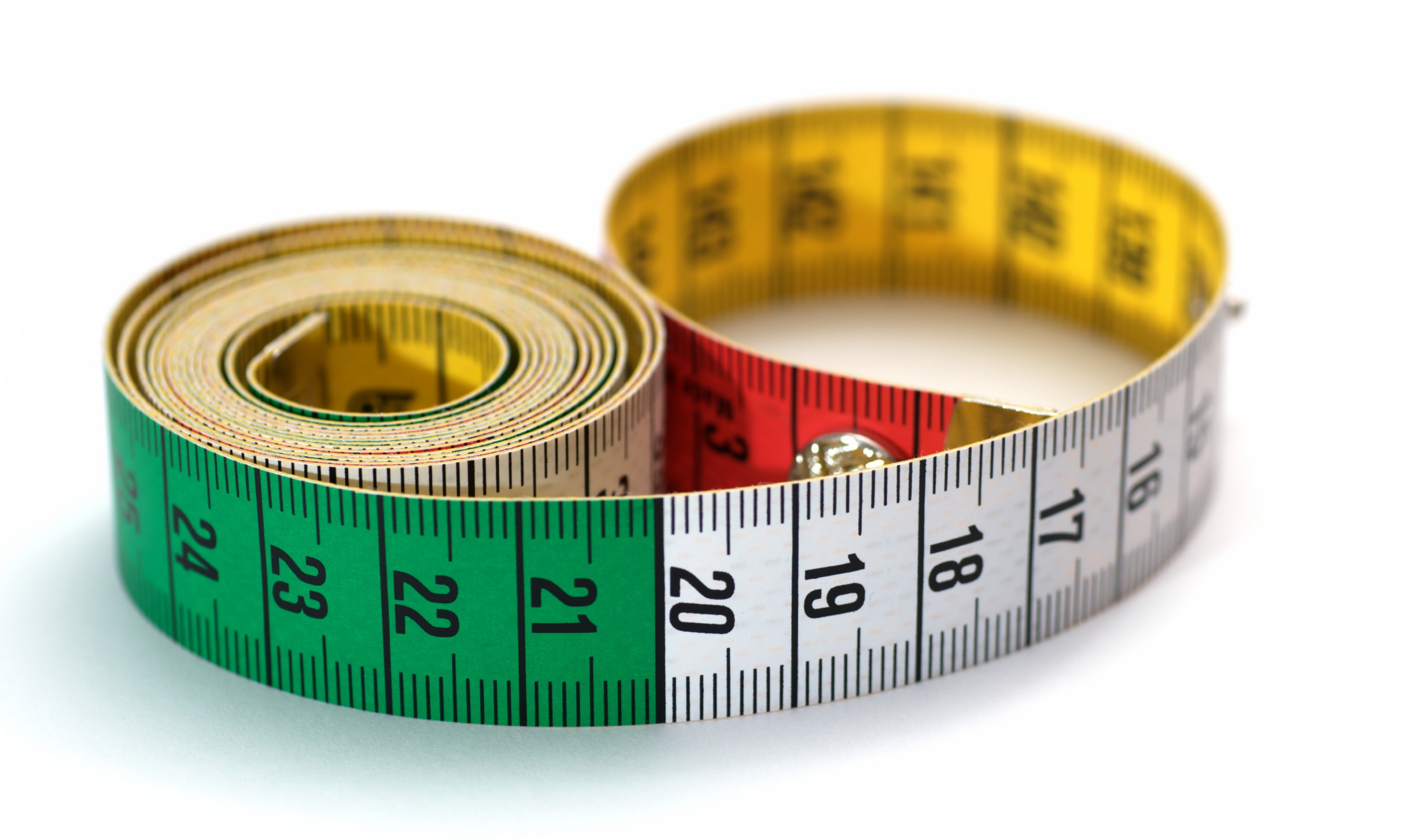


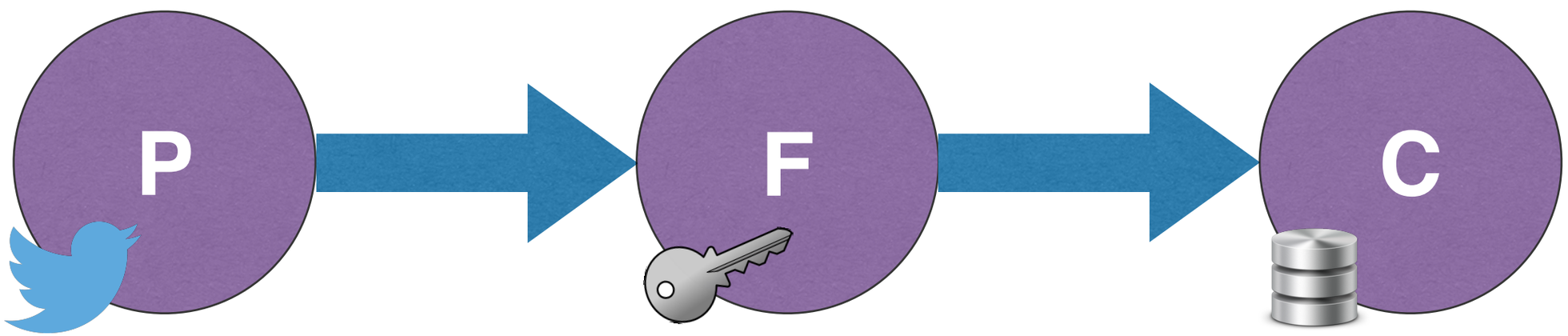

C
P
C
F_X
C

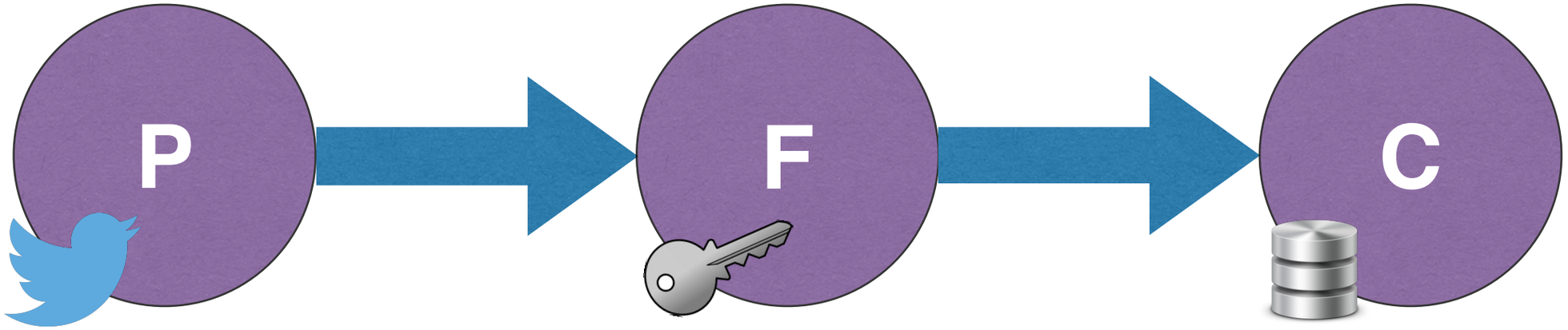

C
P
C
Fx
C

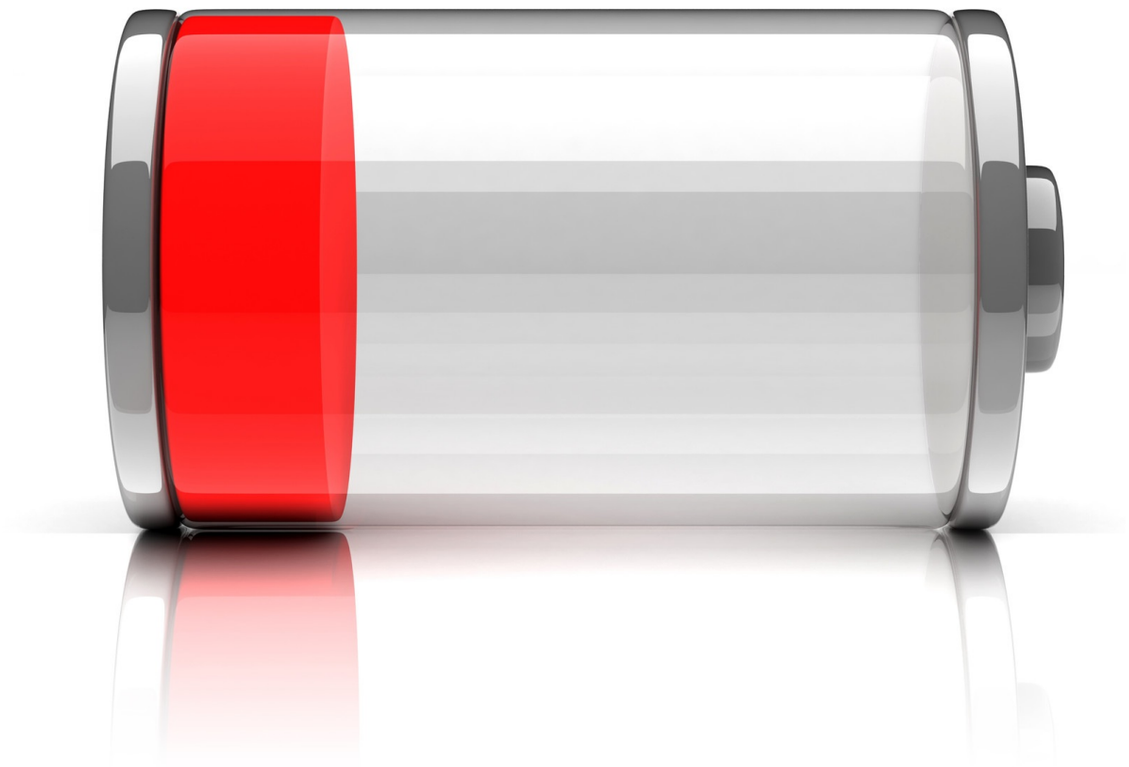

C
P
C
C

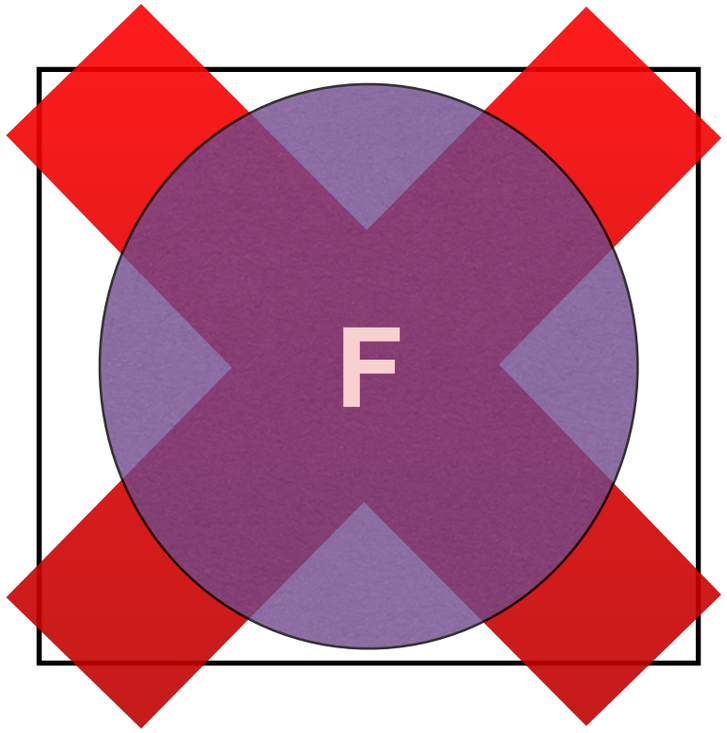

C
P
F
C


24
23
22
21
20
19
18
17
16


P
F
C


P
F
C

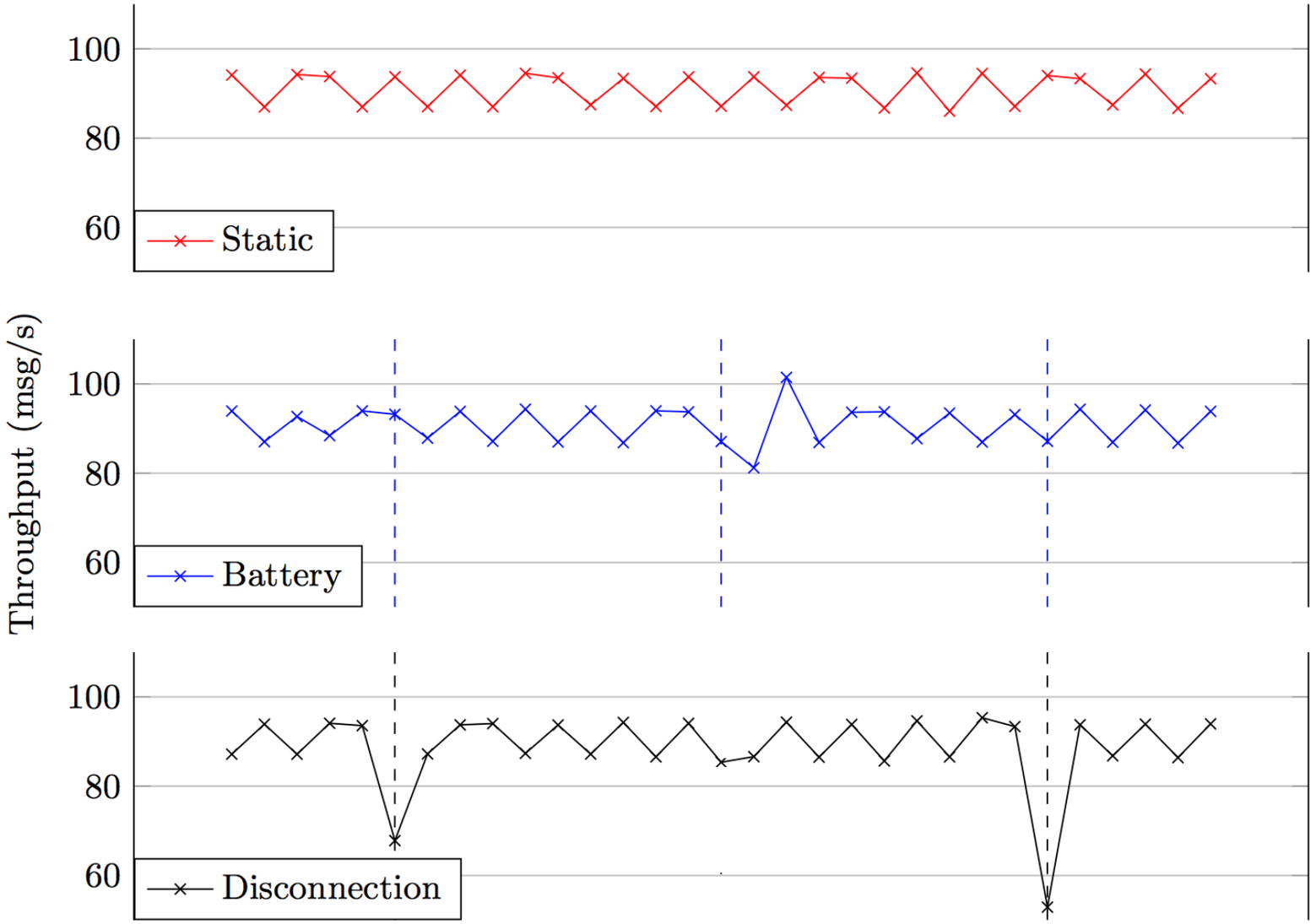



F

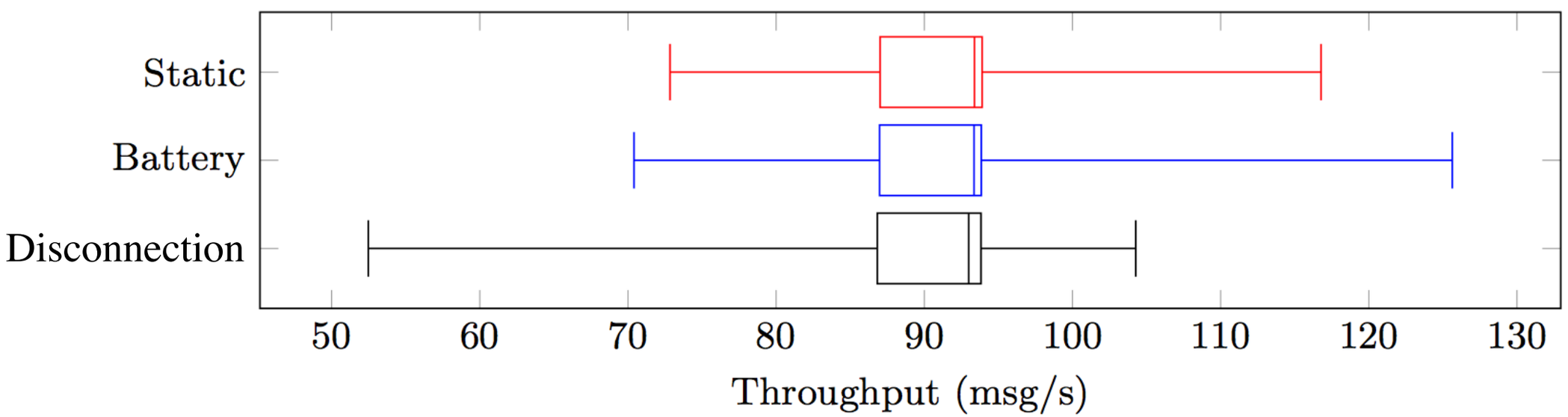
1

Every 5k msgs

Every 10k msgs

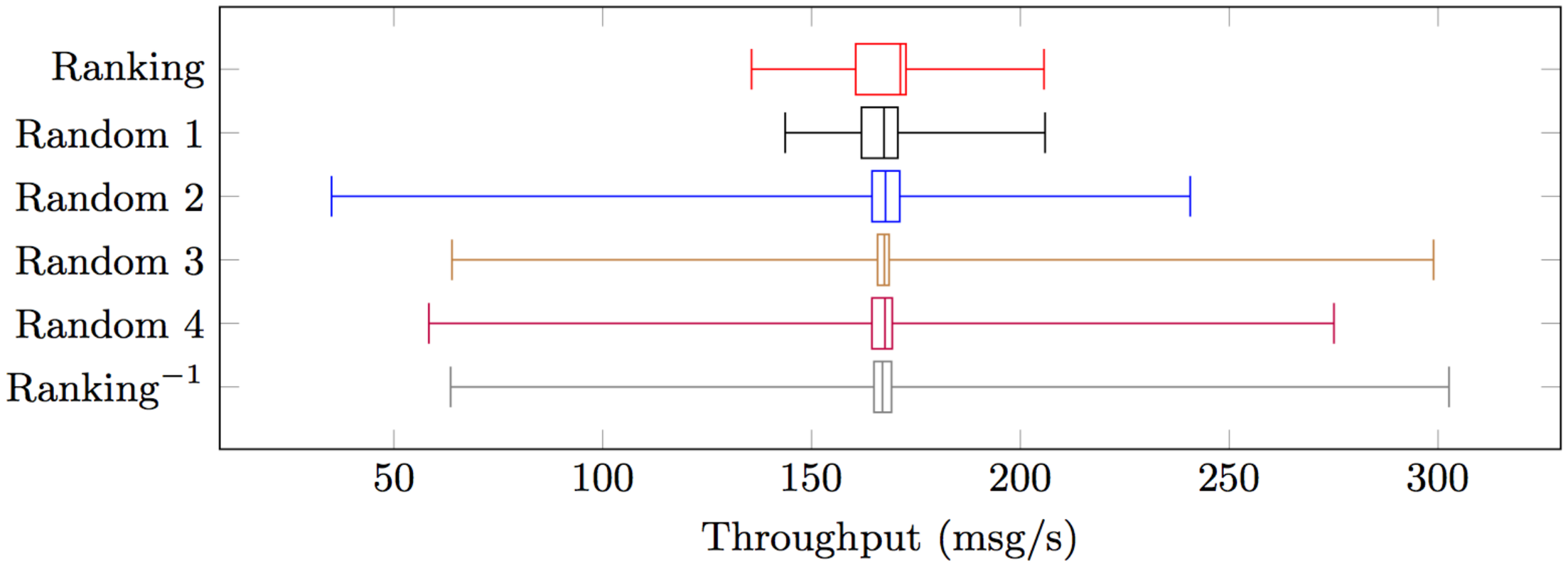


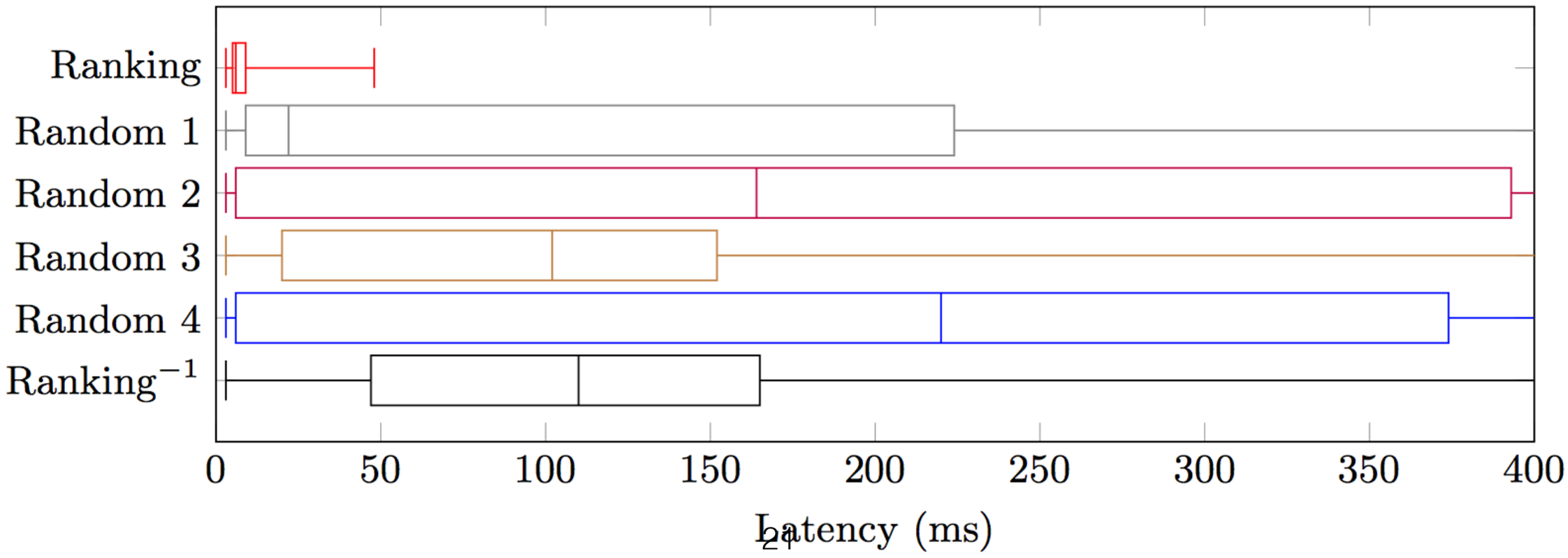
Experiment Time

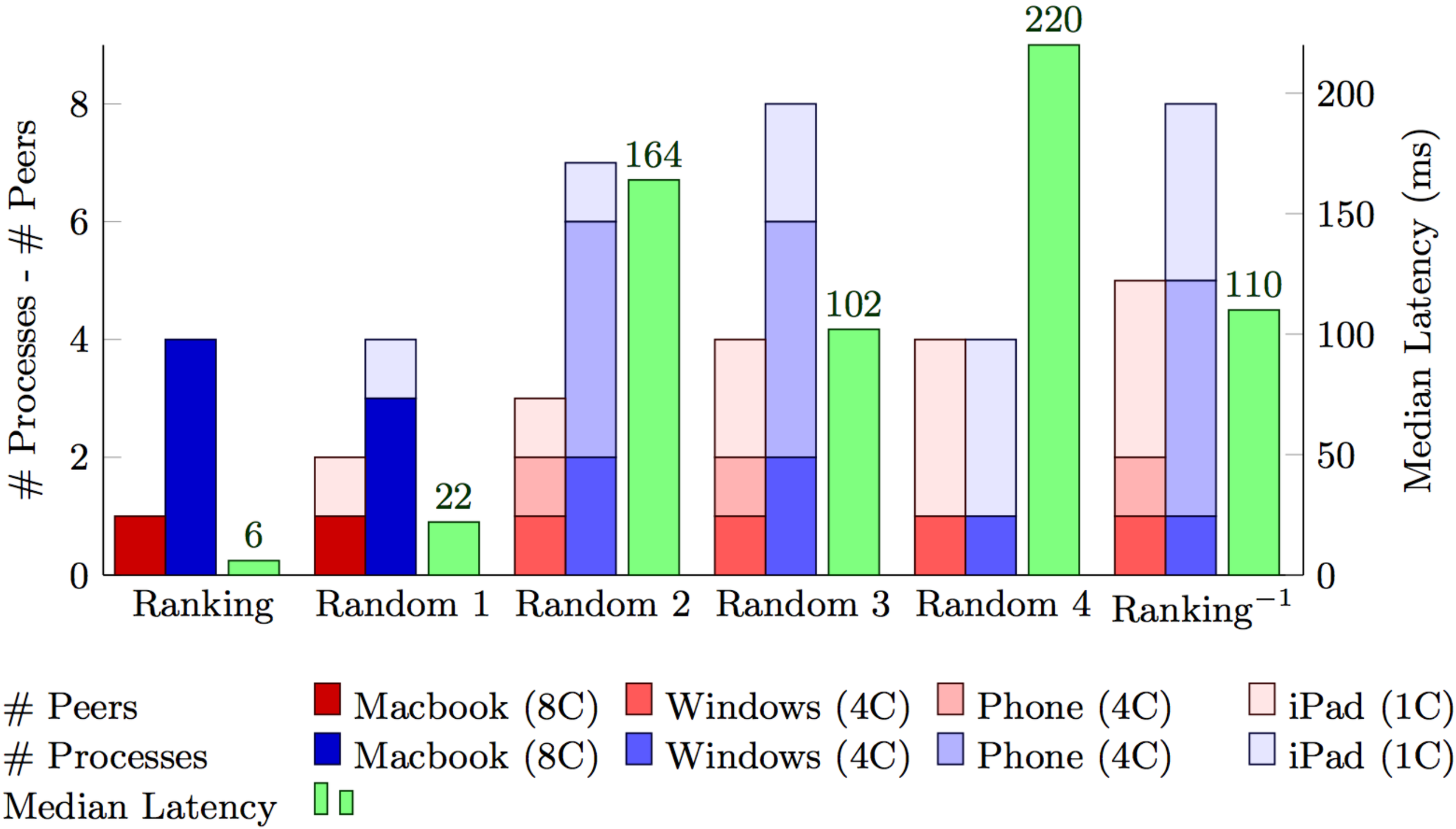


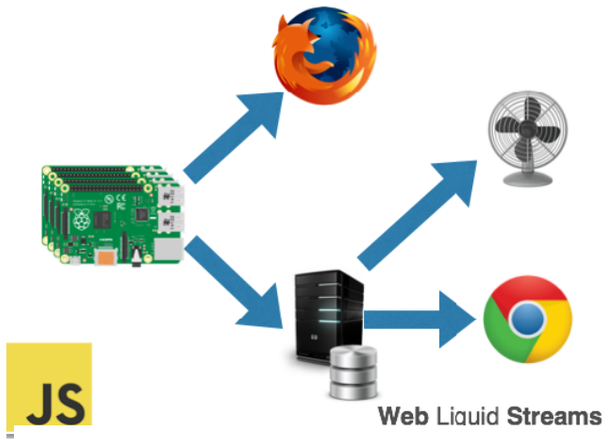
2

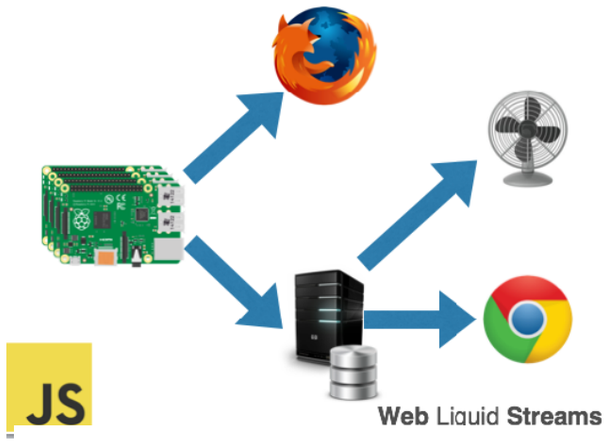
2


Ranking
Random 1
Random 2
Random 3
Random 4
Ranking^{-1}
50
100
150
200
250
300
Throughput (msg/s)

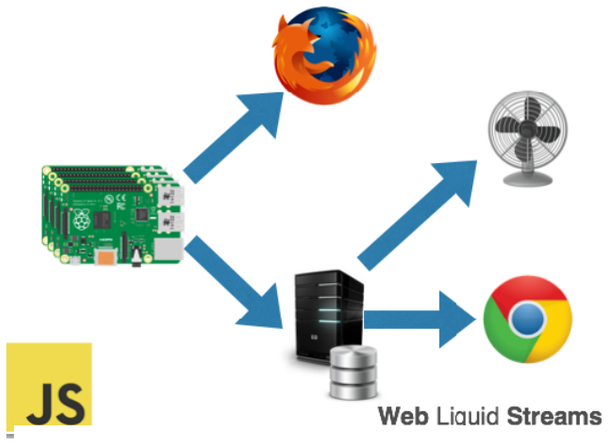

Ranking
Random 1
Random 2
Random 3
Random 4
Ranking^{-1}
0
50
100
150
200
250
300
350
400
Latency (ms)


220
164
110
102
22
6
Processes - # Peers
Median Latency (ms)
0
2
4
6
8
0
50
100
150
200
Ranking
Random 1
Random 2
Random 3
Random 4
Ranking^{-1}
Peers
Processes
Median Latency
Macbook (8C)
Macbook (8C)
Windows (4C)
Windows (4C)
Phone (4C)
Phone (4C)
iPad (1C)
iPad (1C)

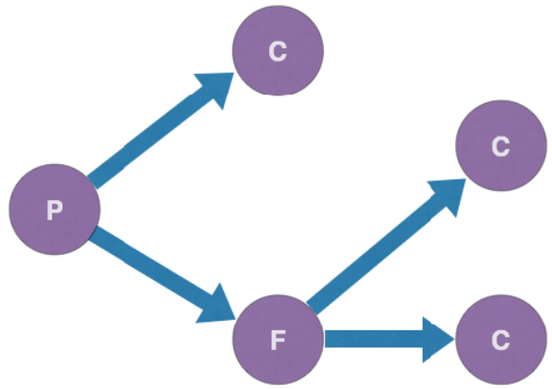

JS
Web Liquid Streams

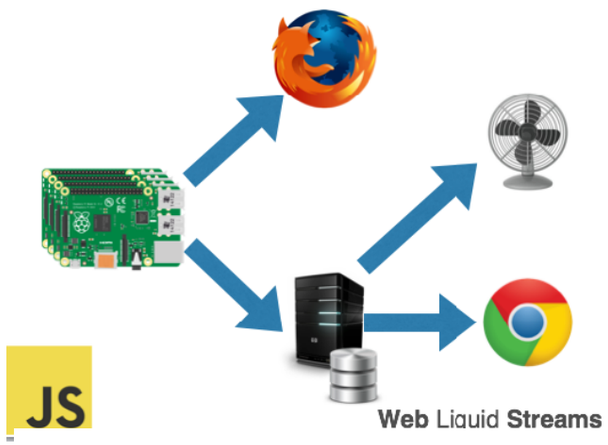

JS
Web Liquid Streams


WebRTC
JS

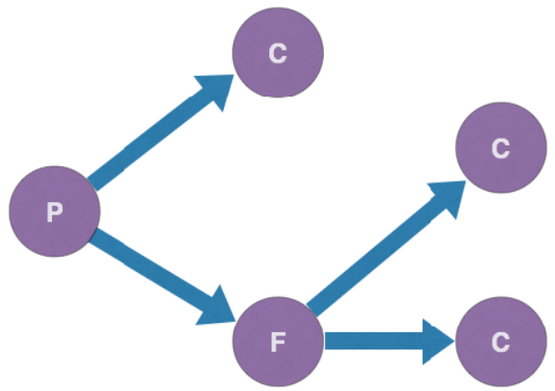

JS
Web Liquid Streams

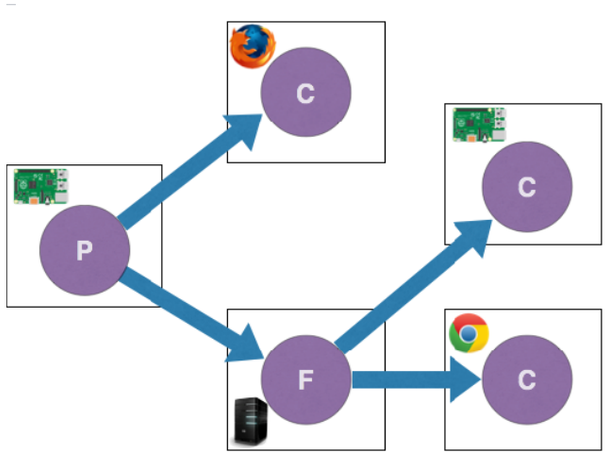

WebRTC

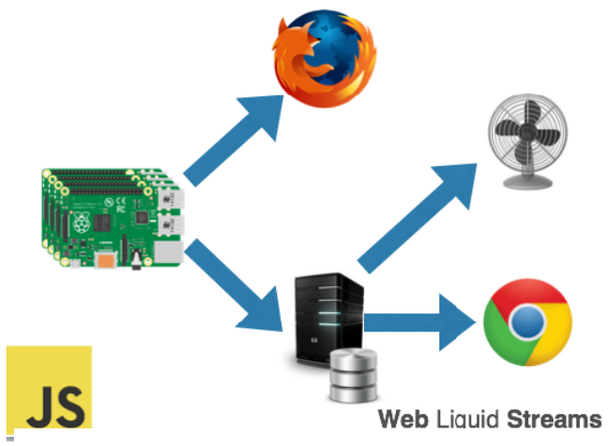

C
C
P
F
C

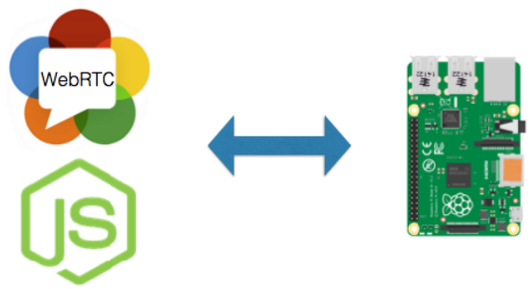

JS
Web Liquid Streams

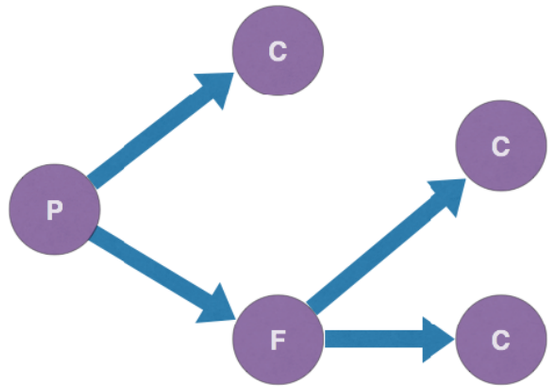

WebRTC

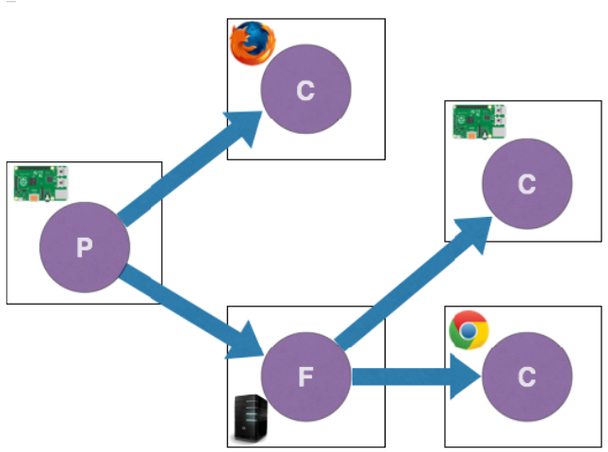

C
C
P
F
C


C
C
P
F
C

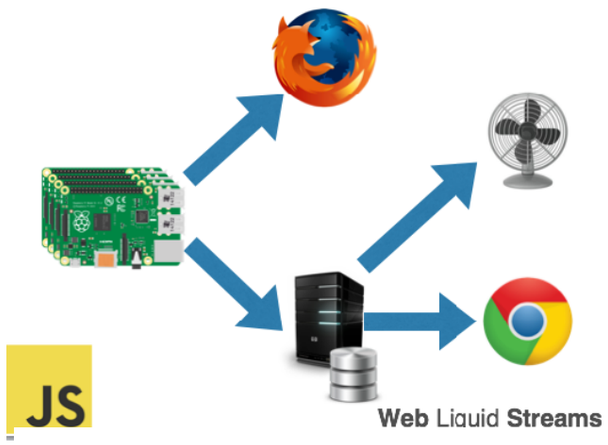


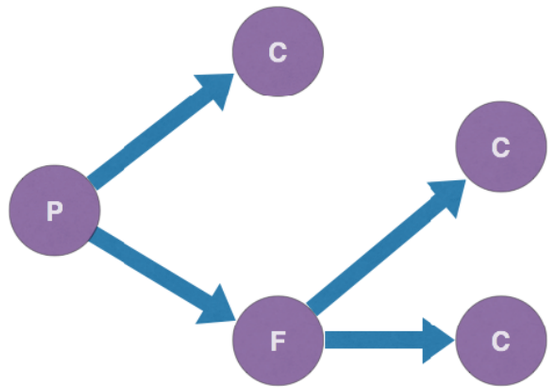






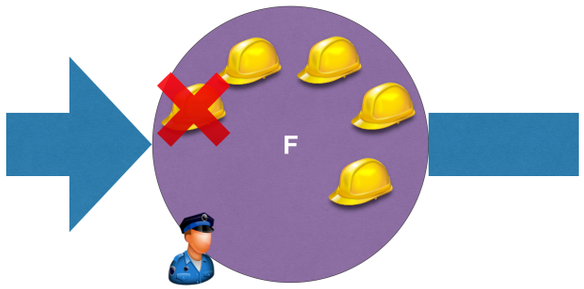
			R_1	R_2	R_3
Peer	*Con.*	Hardware	x	x	x
		CPU		x	
	Variables	Energy		x	
		Battery		x	
		Parallelism		x	
		Deployment Cost			
		Queue Size			x
		Execution Time			x
Connector	*Con.*	Network Typology			x
		Bandwidth (Upload)			x
		Bandwidth (Download)			x
	Var.	Message size (incoming)			x
		Message size (outgoing)			x
		Message rate (incoming)			x
		Message rate (outgoing)			x
		RTT			x
Topology	*Con.*	Hardware Dependencies	x	x	x
		Whitelist	x	x	x
		Blacklist	x	x	x
	Var.	Delay			
		Throughput			

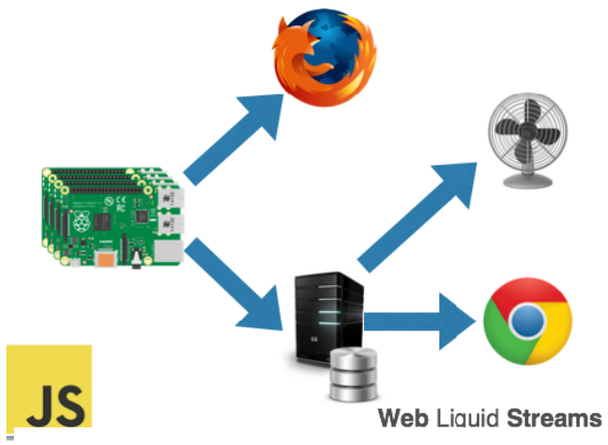


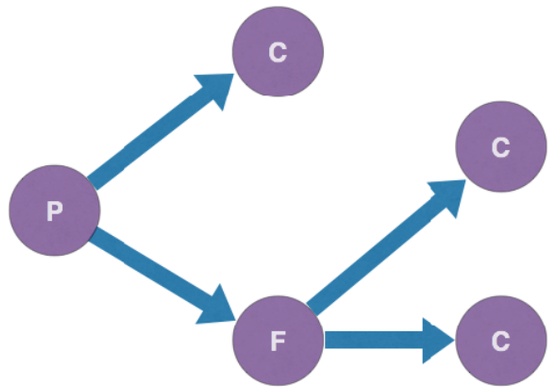


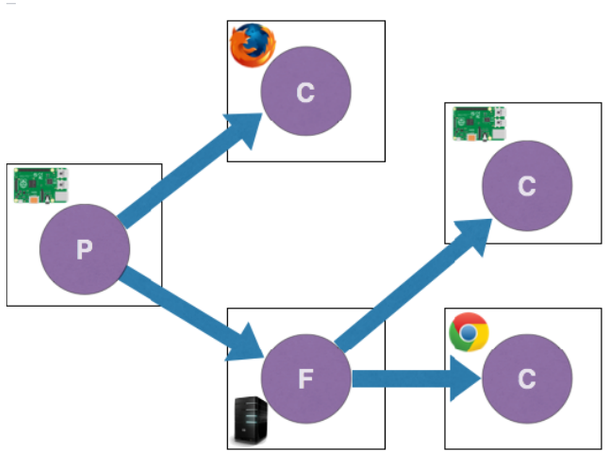
			R_1	R_2	R_3
Peer	*Con.*	Hardware	x	x	x
	Variables	CPU		x	
		Energy		x	
		Battery		x	
		Parallelism		x	
		Deployment Cost			
		Queue Size			x
		Execution Time			x
Connector	*Con.*	Network Typology			x
		Bandwidth (Upload)			x
		Bandwidth (Download)			x
	Var.	Message size (incoming)			x
		Message size (outgoing)			x
		Message rate (incoming)			x
		Message rate (outgoing)			x
		RTT			x
Topology	*Con.*	Hardware Dependencies	x	x	x
		Whitelist	x	x	x
		Blacklist	x	x	x
	Var.	Delay			
		Throughput			

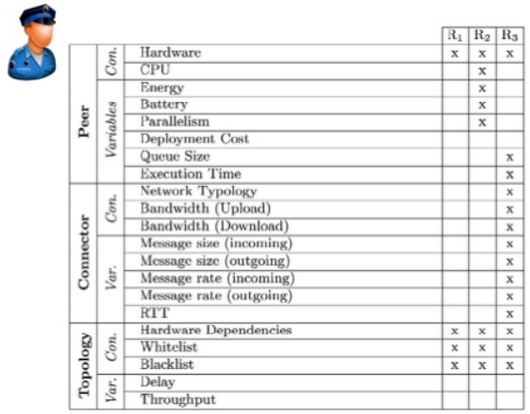
Fault Tolerance





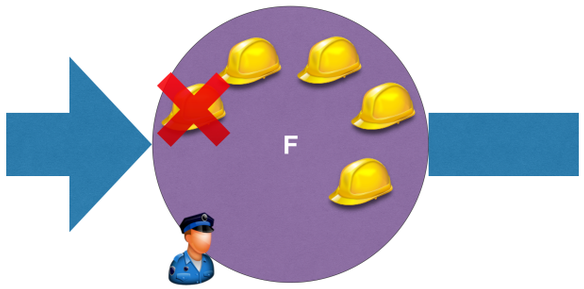


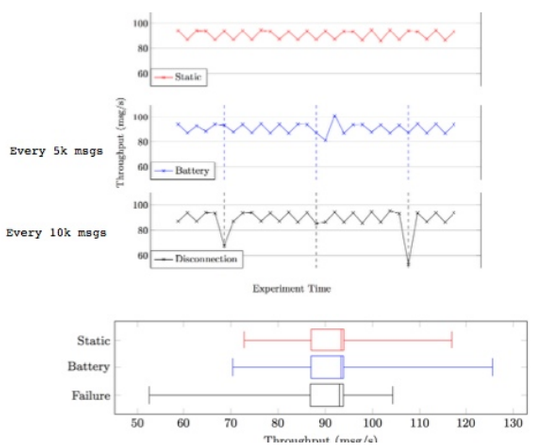


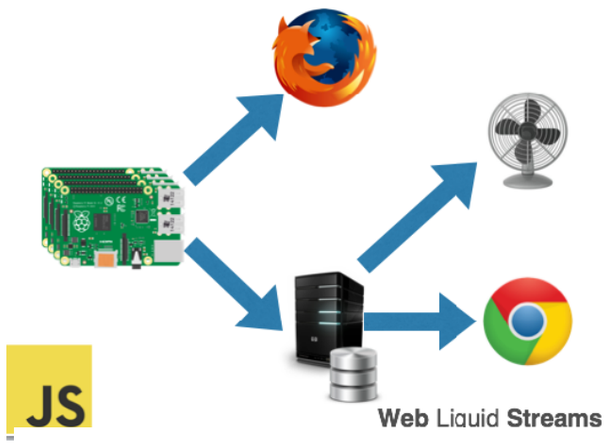


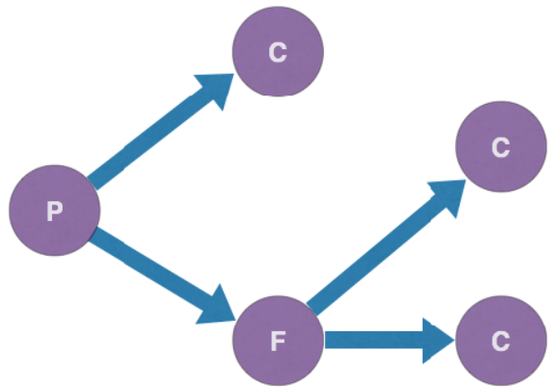
			R_1	R_2	R_3
Peer	*Con.*	Hardware	x	x	x
	Variables	CPU		x	
		Energy		x	
		Battery		x	
		Parallelism		x	
		Deployment Cost			
		Queue Size			x
		Execution Time			x
Connector	*Con.*	Network Typology			x
		Bandwidth (Upload)			x
		Bandwidth (Download)			x
	Var.	Message size (incoming)			x
		Message size (outgoing)			x
		Message rate (incoming)			x
		Message rate (outgoing)			x
		RTT			x
Topology	*Con.*	Hardware Dependencies	x	x	x
		Whitelist	x	x	x
		Blacklist	x	x	x
	Var.	Delay			
		Throughput			

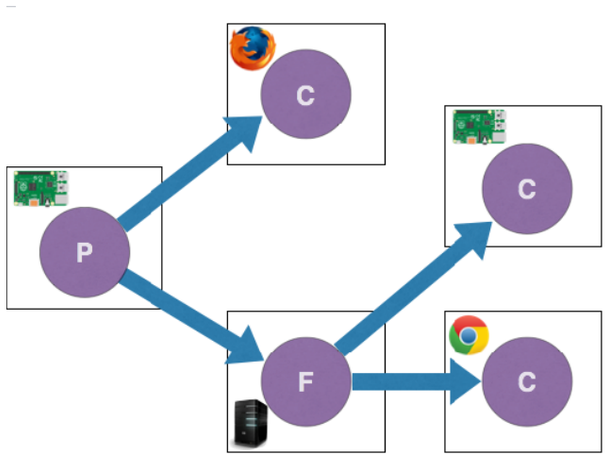
Fault Tolerance





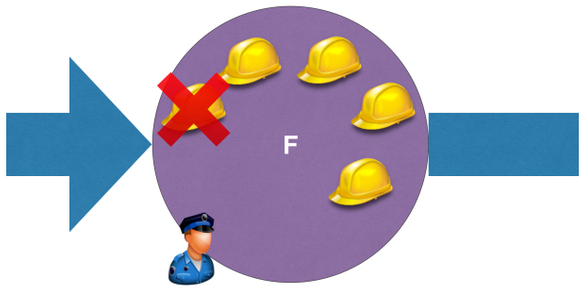


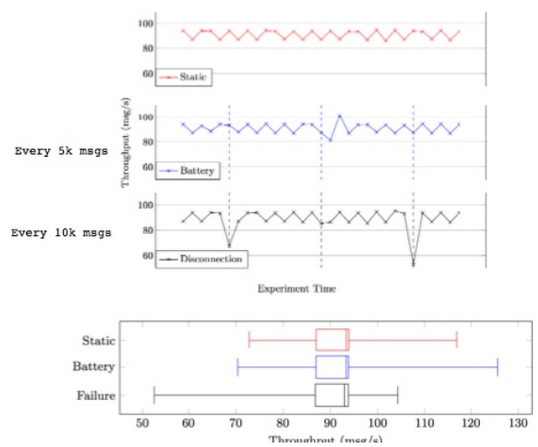


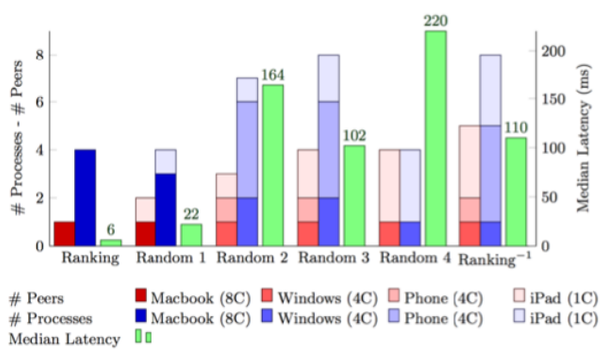


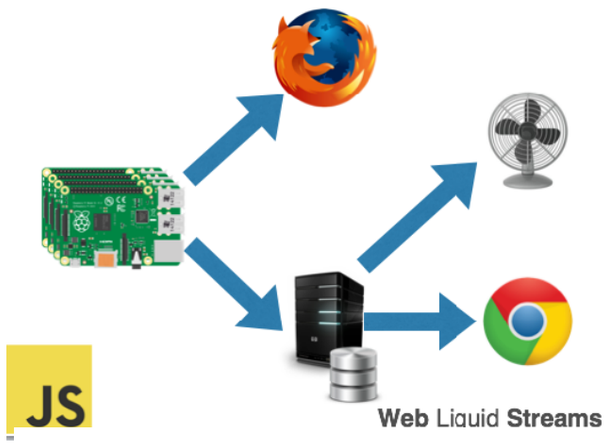
			R_1	R_2	R_3
Peer	*Con.*	Hardware	x	x	x
	Variables	CPU		x	
		Energy		x	
		Battery		x	
		Parallelism		x	
		Deployment Cost			
		Queue Size			x
		Execution Time			x
Connector	*Con.*	Network Typology			x
		Bandwidth (Upload)			x
		Bandwidth (Download)			x
	Var.	Message size (incoming)			x
		Message size (outgoing)			x
		Message rate (incoming)			x
		Message rate (outgoing)			x
		RTT			x
Topology	*Con.*	Hardware Dependencies	x	x	x
		Whitelist	x	x	x
		Blacklist	x	x	x
	Var.	Delay			
		Throughput			

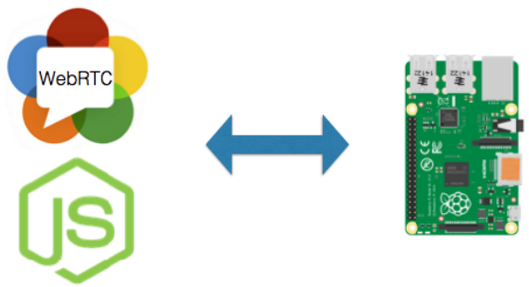
Fault Tolerance

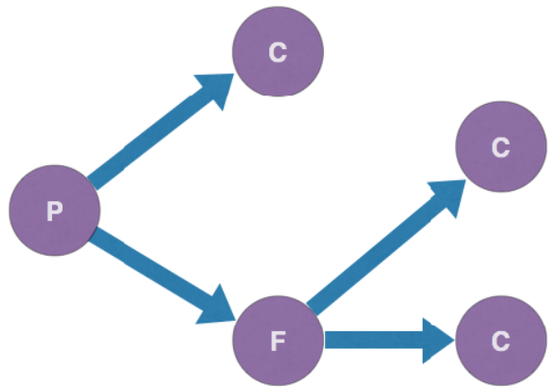


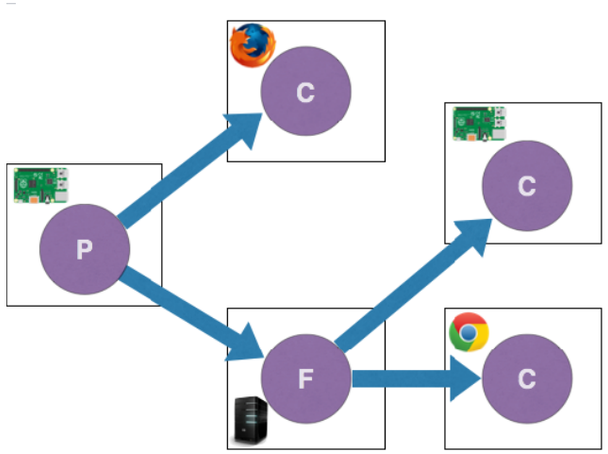


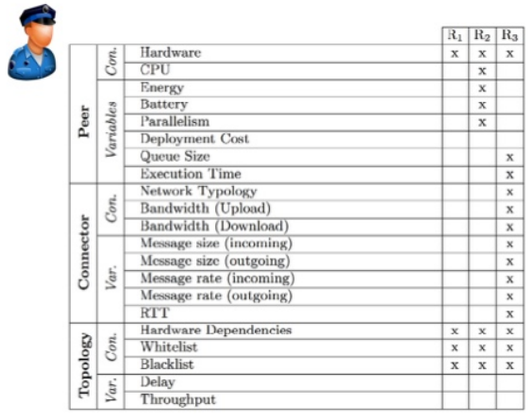






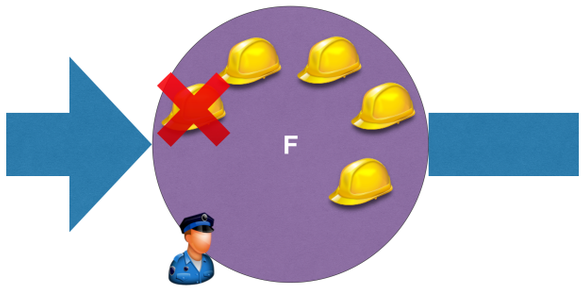


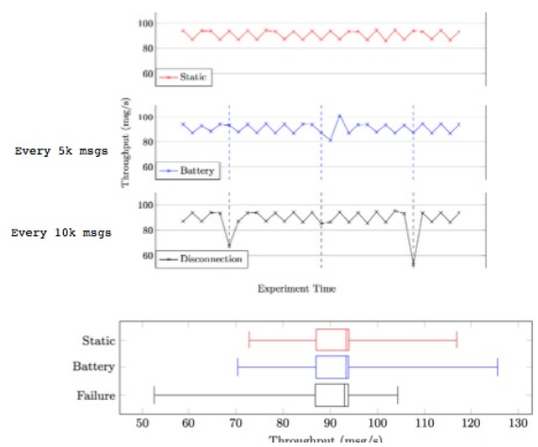


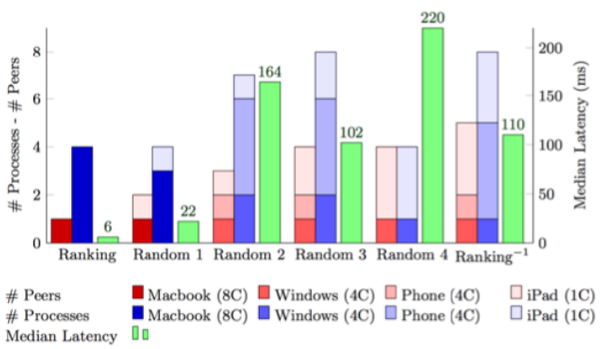


			R_1	R_2	R_3
Peer	*Con.*	Hardware	x	x	x
	Variables	CPU		x	
		Energy		x	
		Battery		x	
		Parallelism		x	
		Deployment Cost			
		Queue Size			x
		Execution Time			x
Connector	*Con.*	Network Typology			x
		Bandwidth (Upload)			x
		Bandwidth (Download)			x
	Var.	Message size (incoming)			x
		Message size (outgoing)			x
		Message rate (incoming)			x
		Message rate (outgoing)			x
		RTT			x
Topology	*Con.*	Hardware Dependencies	x	x	x
		Whitelist	x	x	x
		Blacklist	x	x	x
	Var.	Delay			
		Throughput			

Fault Tolerance








WebR
DEMO

Liquid Stream Processing Across Web Browsers and Web Servers

Masiar Babazadeh
@masiarb
Andrea Gallidabino
@agallidabino
Cesare Pautasso
@pautasso
name.surname@usi.ch

Università della Svizzera italiana

Faculty of Informatics

Thank you.