WSDL - XML format

Client

Server

class SmsMessageSenderProxy

class SmsMessageSender

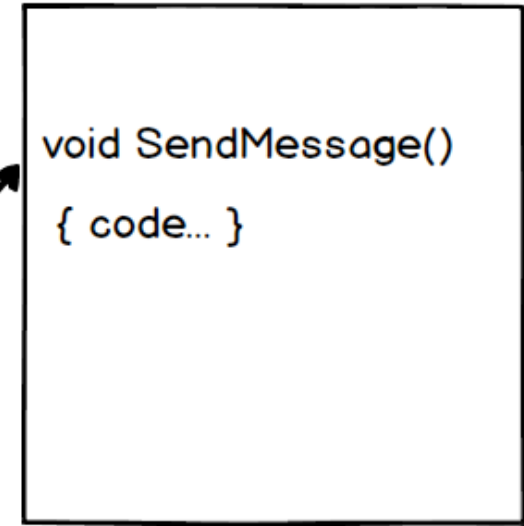void SendMessage()

{ code... }

{ code... }

void SendMessage()
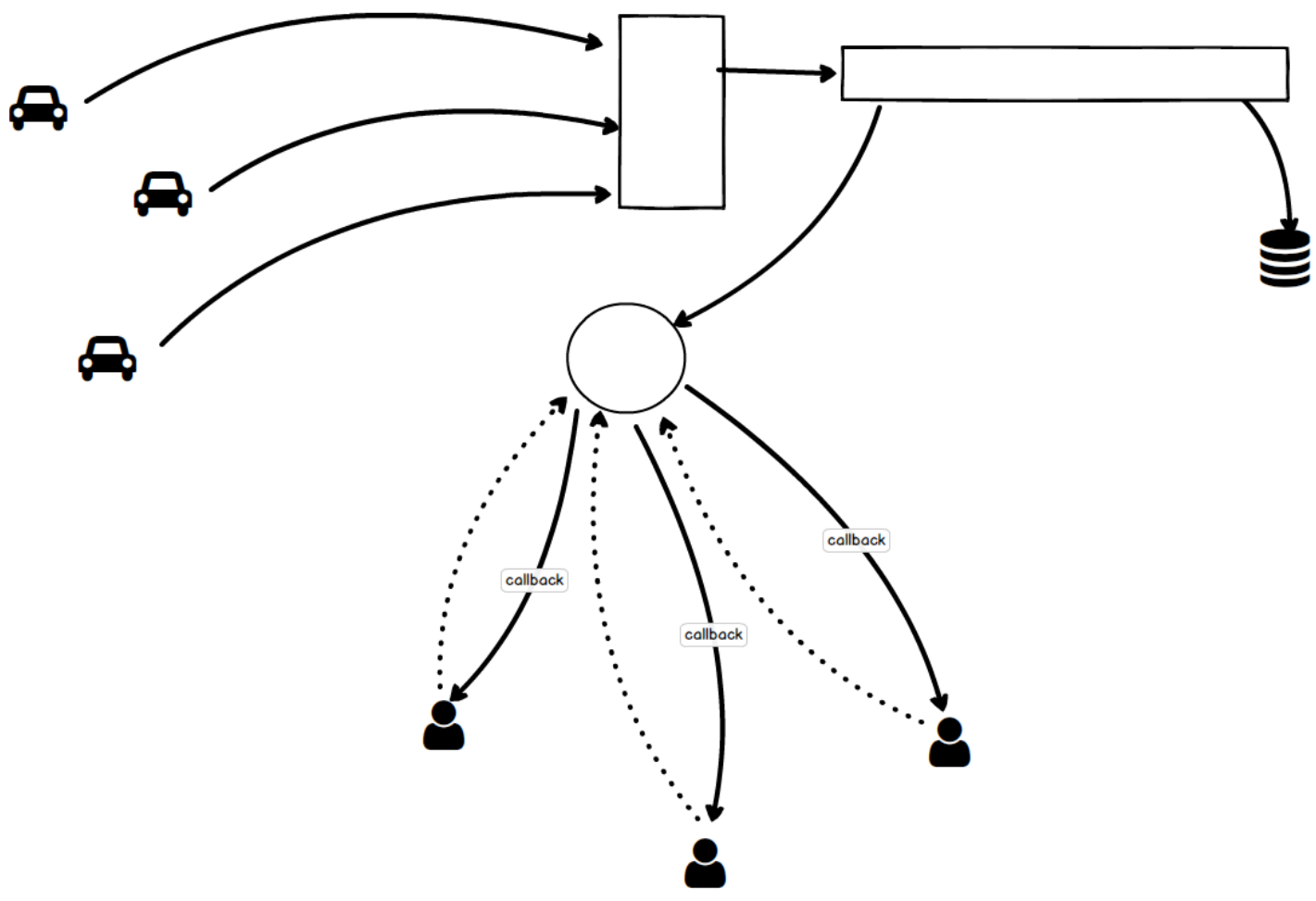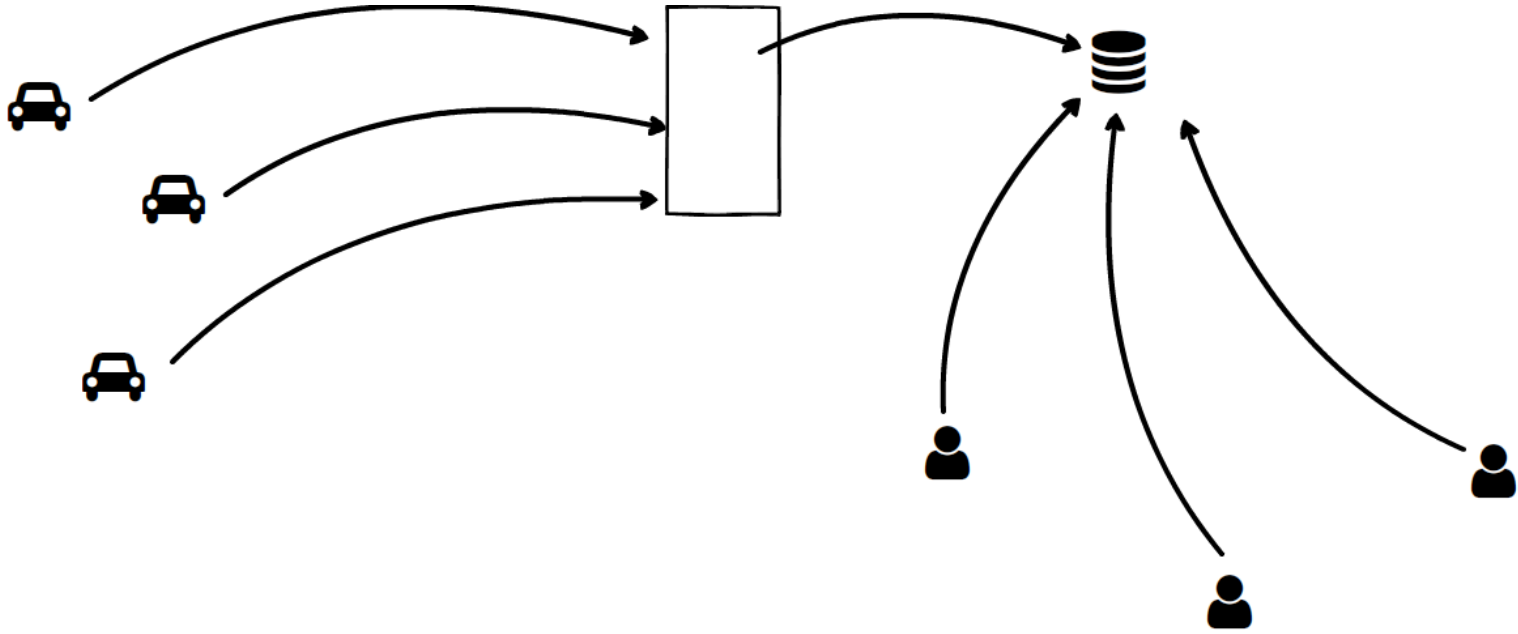
{ code... }
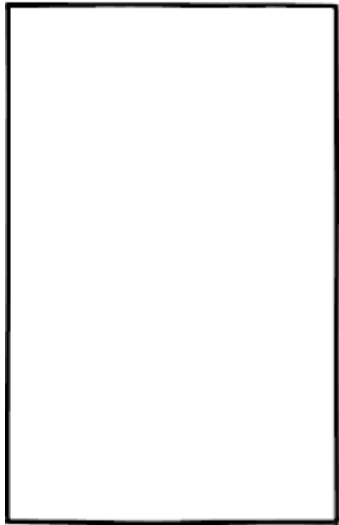
Client
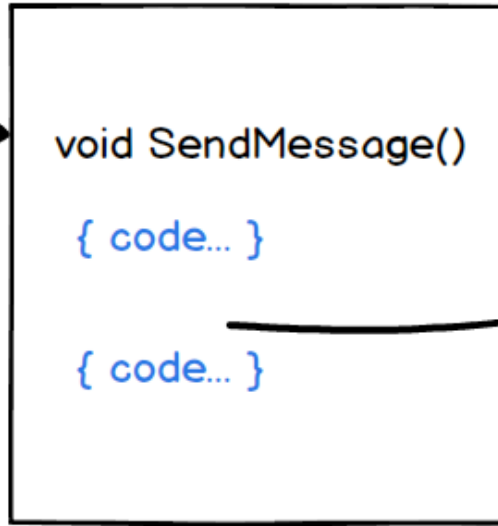
Proxy Class

Logic

```
POST XML (SOAP) over HTTP
```

<soapenv:Envelope
xmlns:soapenv="http://
schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:Divide>
      <!-Optional:->
      <tem:x>4</tem:x>
      <!-Optional:->
      <tem:y>0</tem:y>
    </tem:Divide>
  </soapenv:Body>
</soapenv:Envelope>

server

```
XML (SOAP) (over HTTP)
```

.NET Core / Web API 2 (.NET Framework)

```
POST api/customers HTTP/1.1
Host: localhost
Content-Type: application/json
{blank-line}
{
   "firstname": "Marcin", "lastname":
"Sulecki"
}
```

```
POST api/customers HTTP/1.1
Host: localhost
Content-Type: application/xml
{blank-line}
<xml>
   <firstname>Ashish</firstname>
   <lastname>Seghal</lastname>
</xml>
```

```
GET api/customers/10 HTTP/1.1
Host: localhost
Accept: application/json
```
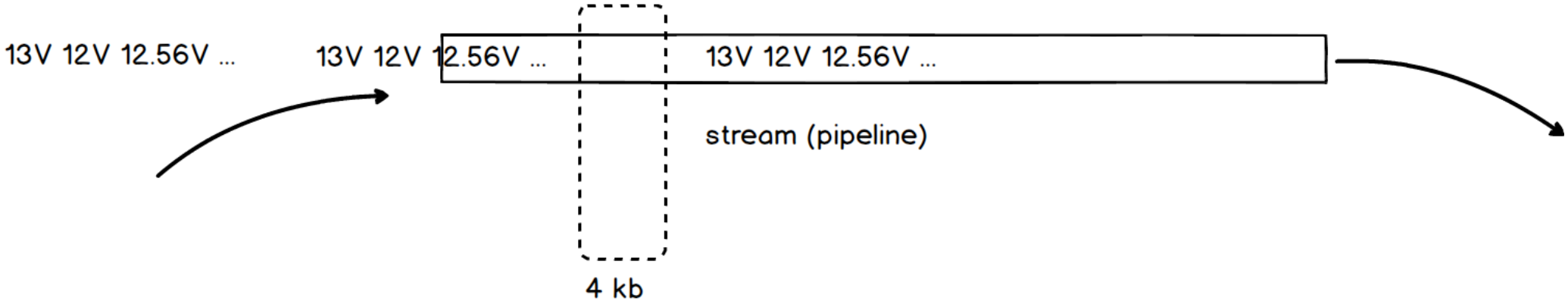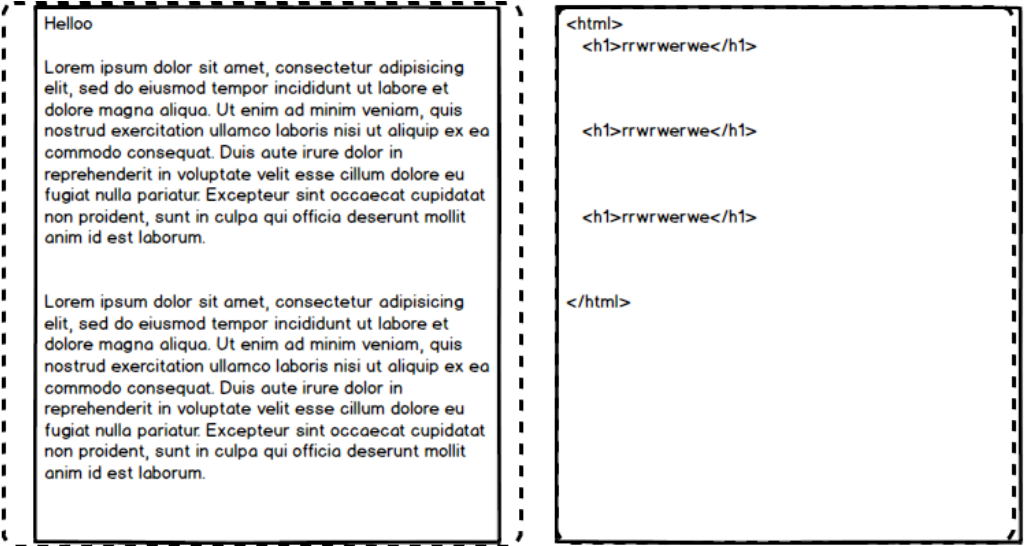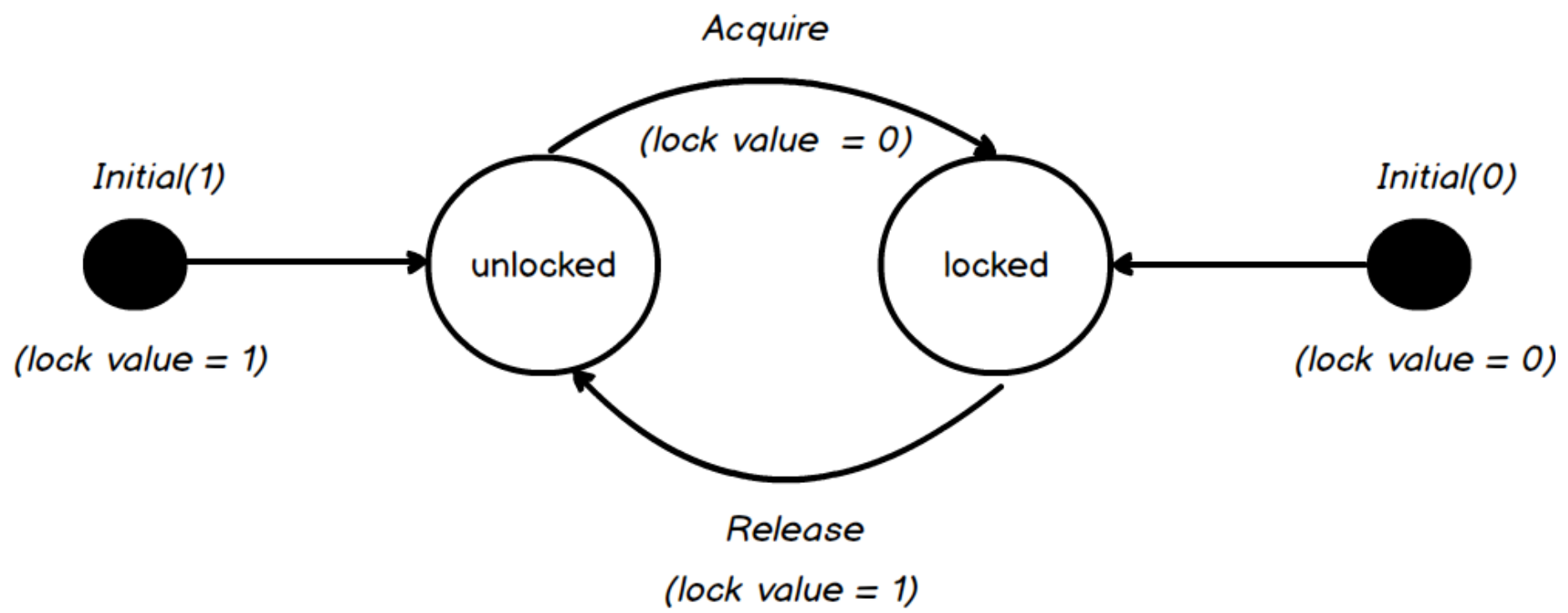
```
GET api/customers HTTP/1.1
Host: localhost
Accept: application/json
```
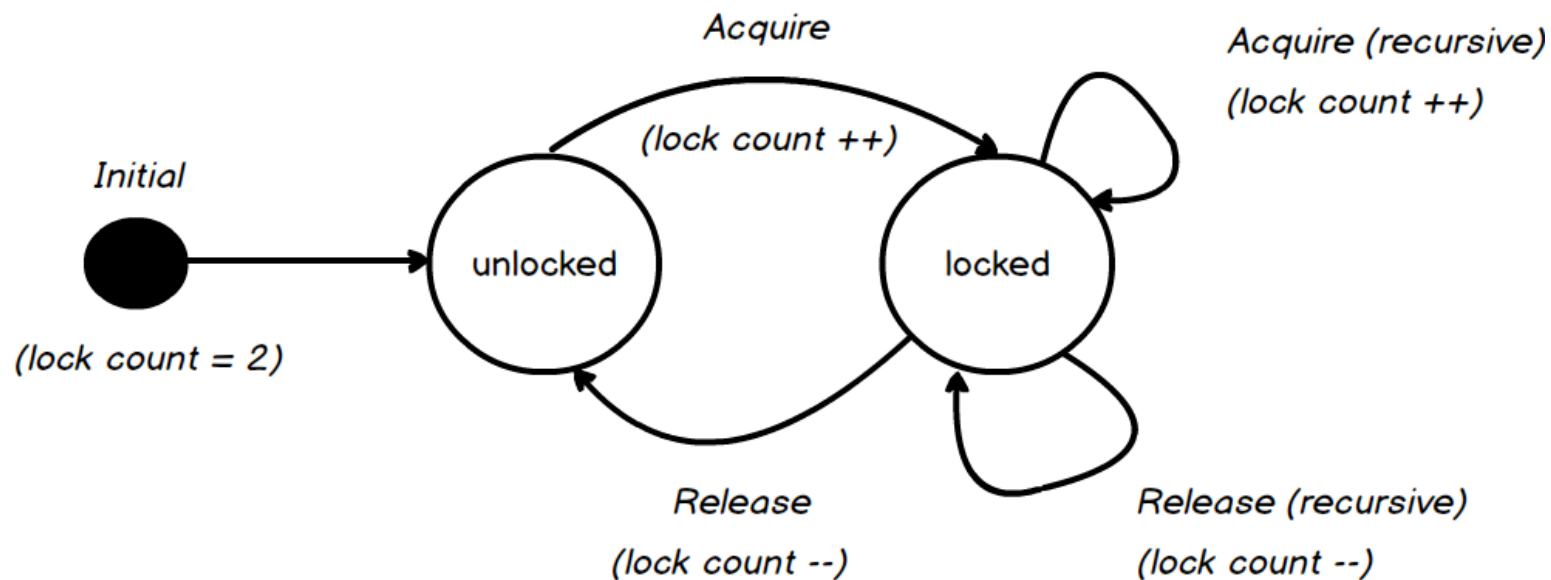
```
PUT api/customers/10 HTTP/1.1
Host: localhost
Content-Type: application/json
{
   "firstname": "Bartek", "lastname":
"Sulecki"
}
```

```
PATCH api/customers/10 HTTP/1.1
Host: localhost
Content-Type: application/json
{
   "firstname": "Bartek"
}
```

```
DELETE api/customers/10 HTTP/1.1
Host: localhost
```

Helloo

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
<html>
  <h1>rrwrwerwe</h1>



  <h1>rrwrwerwe</h1>



  <h1>rrwrwerwe</h1>



</html>
```

13V 12V 12.56V ...

13V 12V 12.56V ...   12.56V ...   13V 12V 12.56V ...

stream (pipeline)

4 kb

Initial(1)

(lock value = 1)

Acquire

(lock value = 0)

unlocked

locked

Initial(0)

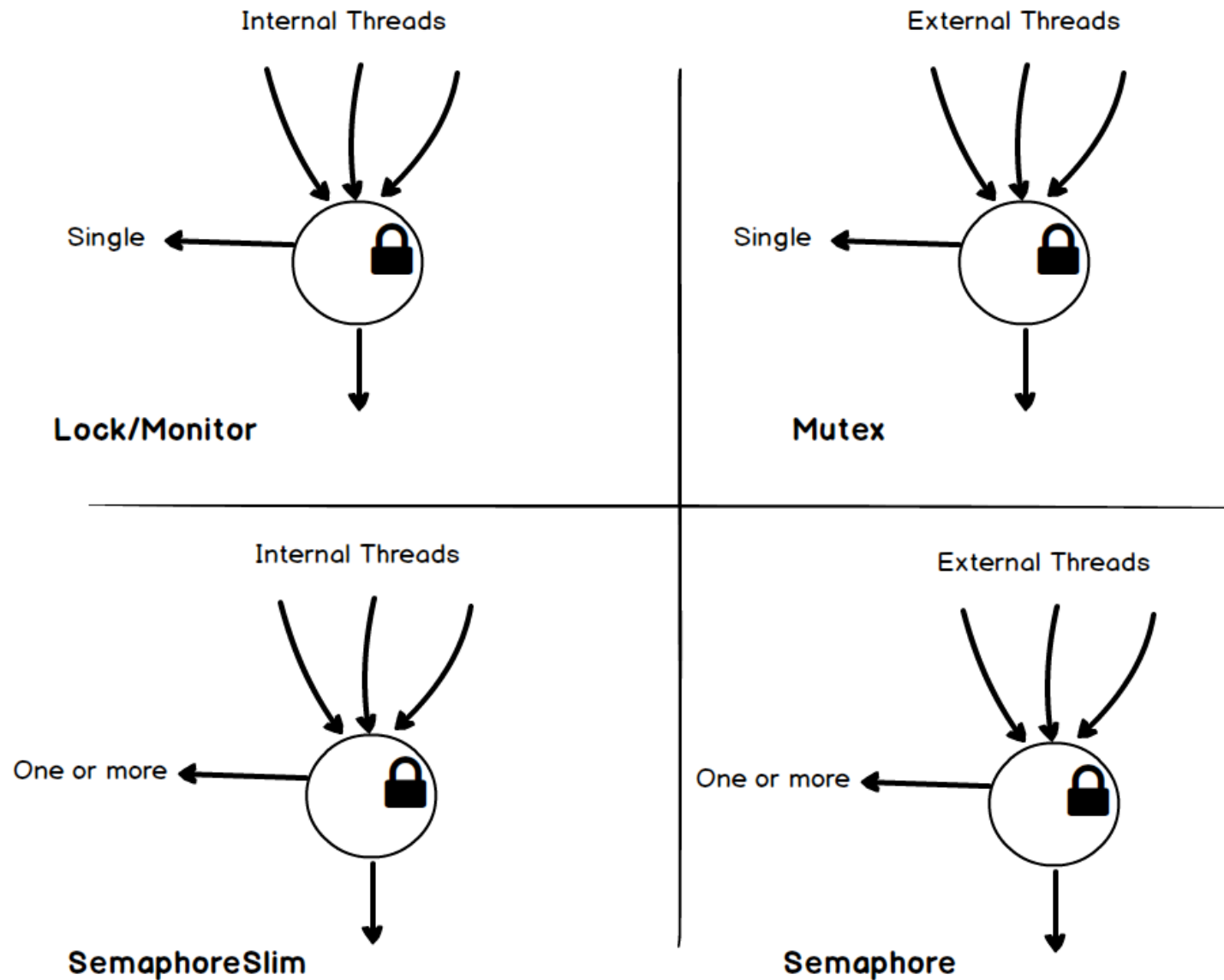(lock value = 0)

Release

(lock value = 1)

```
procedure V (S : Semaphore; I : Integer);
begin
  S := S + I (* operacja atomowa *)
end;

procedure P (S : Semaphore; I : Integer);
begin
  (* cała operacja jest atomowa *)
  repeat
    Wait();
  until S >= I;
  S := S - I;
end;
```

Monitor vs Mutex vs Semaphore vs SemaphoreSlim

Internal Threads

Single ←

Lock/Monitor

External Threads

Single ←

Mutex

Internal Threads

One or more ←

SemaphoreSlim

External Threads

One or more ←

Semaphore

thread1

thread2

Debit(100)

Credit(50)

balance = 1000

t1 — TryAdd() #1

t3 — TryAdd() #2 — TryRemove() — t2

Dictionary

time

t1 — Enqueue()

TryDeque() — t2

t3 — Enqueue()

Queue

Producers

Consumers

t1 — Add() →

t2 — Add() →

t3 — Add() →

Collection

Get() → t5

Get() → t4

Add()