Scalable Elastic System Architecture (SESA)

Dan Schatzberg, Boston University
Jonathan Appavoo, Boston University
Orran Krieger, VMware
Eric Van Hensbergen, IBM Research Austin

The goal

Perform more computation with fewer resources

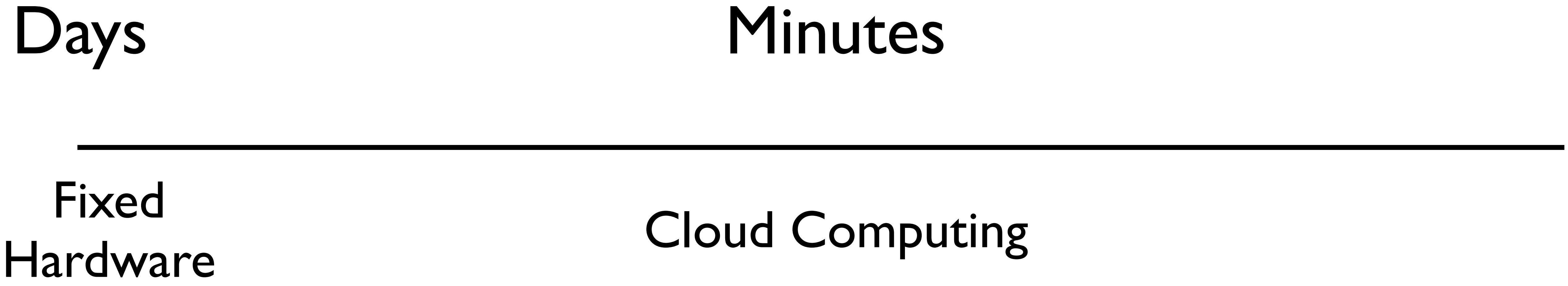
Fixed Resources

- Hardware as a fixed resource
- Focus on reducing computation's need for hardware resources
- Multiplex hardware resources for different computations

Elastic Resources

- Cloud Computing
  - Pay as you go hardware
- Focus on providing hardware to the computation that requires it

Time to scale hardware



Days

Minutes

Fixed Hardware

Cloud Computing

Time to scale hardware

Days

Minutes

Fixed Hardware

Cloud Computing

Elastic Applications

Time to scale hardware

Days

Minutes

Milliseconds

Fixed Hardware

Cloud Computing

?

Elastic Applications

Interactive HPC

- Medical imaging application
  - interactive
  - 1 megapixel image
  - quadratic memory consumption - ~14TB

Interactive HPC

- Fixed Hardware
  - Purchase a cluster

Interactive HPC

- Cloud Computing
  - Allocate a cluster
  - Maintain interactivity
  - 650+ EC2 instances - $8000 dollars / 8 hour day

Can we do better?

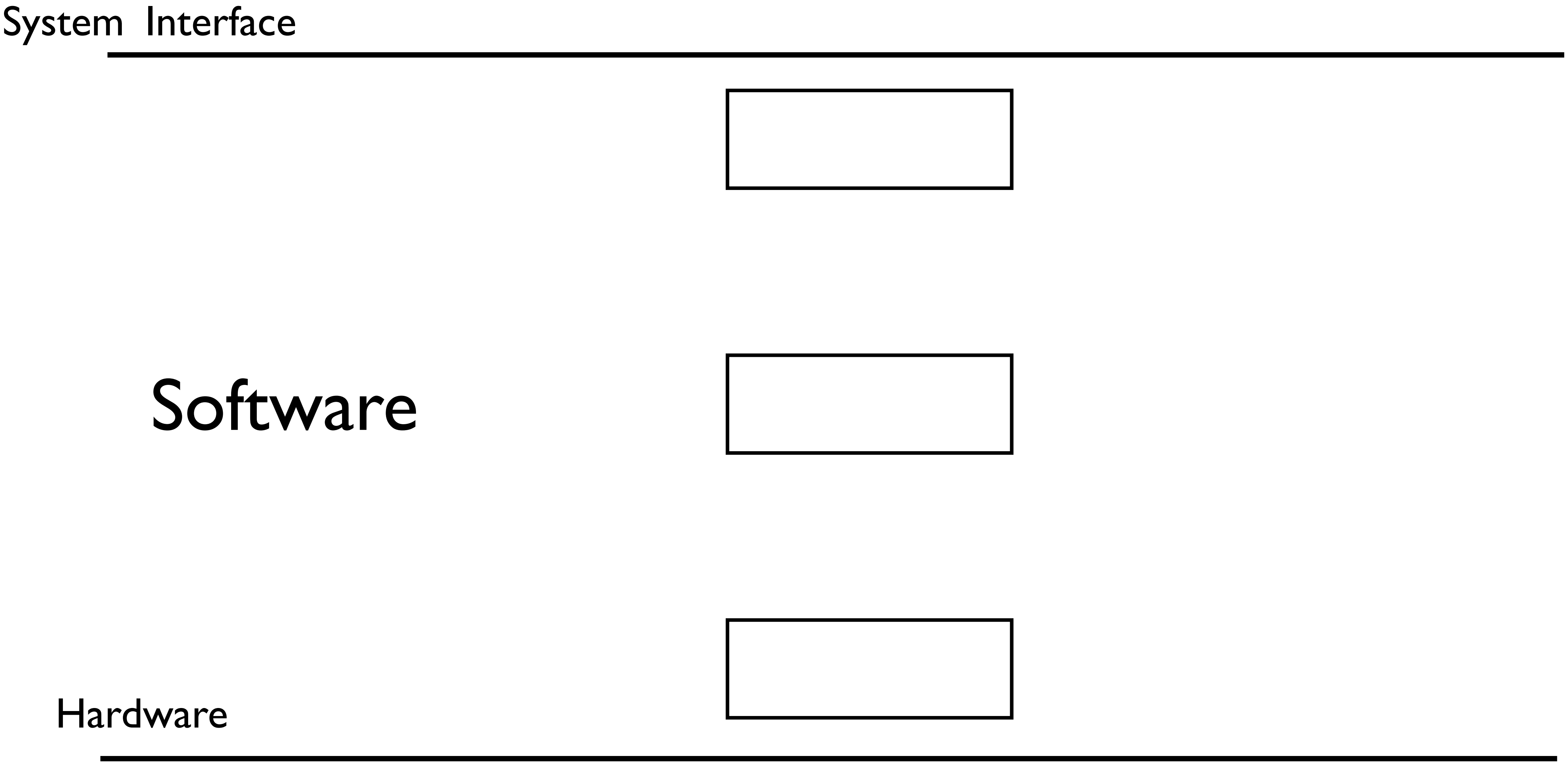
Where we're starting

Treat elasticity as a first-class system characteristic

OUTLINE

1. THE PROBLEM
2. OBSERVATIONS
   1. Top-Down Demand
   2. Bottom-Up Support
   3. Modularity
3. OUR TAKE ON A SOLUTION
4. PROTOTYPE & CHALLENGES

Top-Down Demand

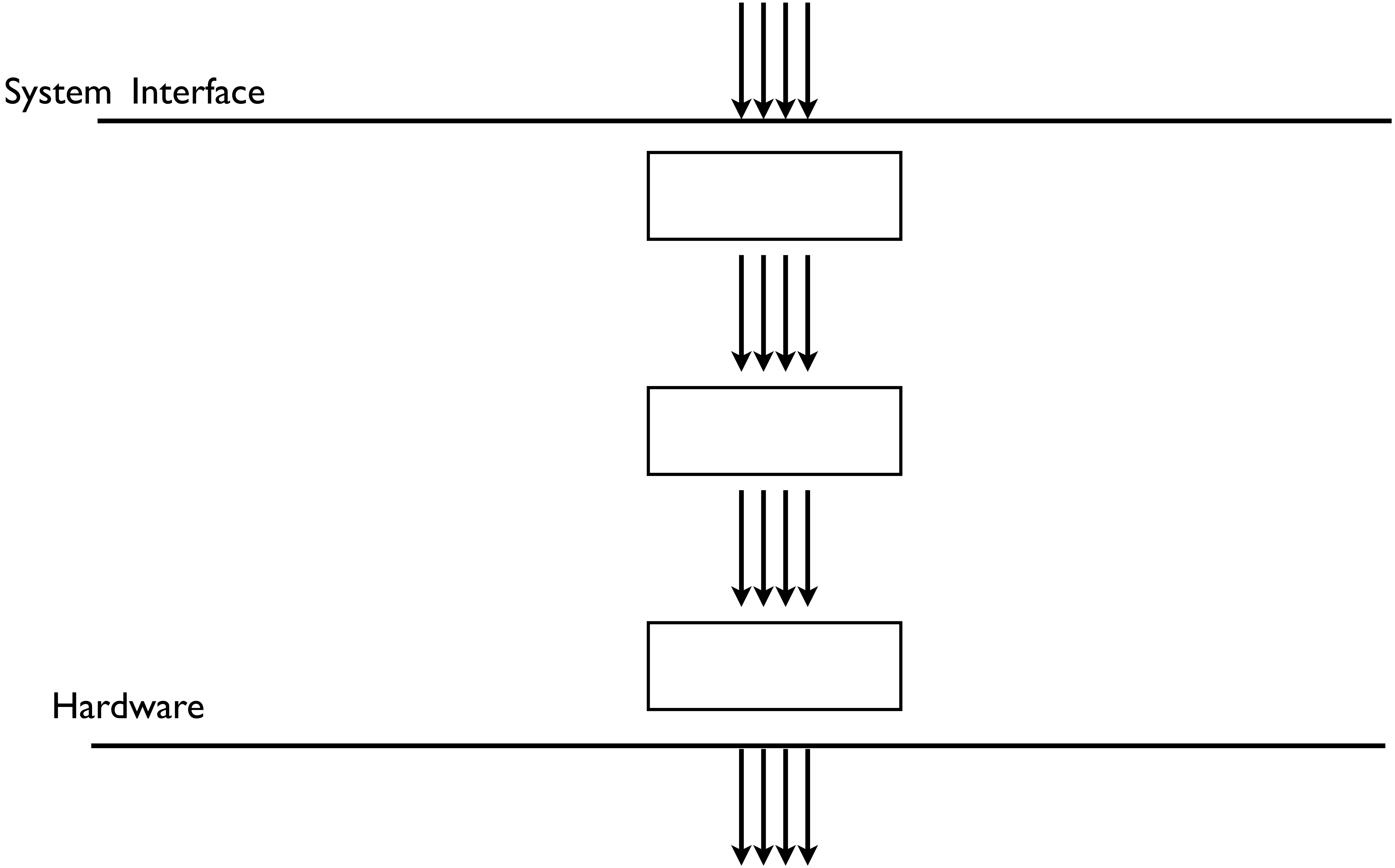


System Interface

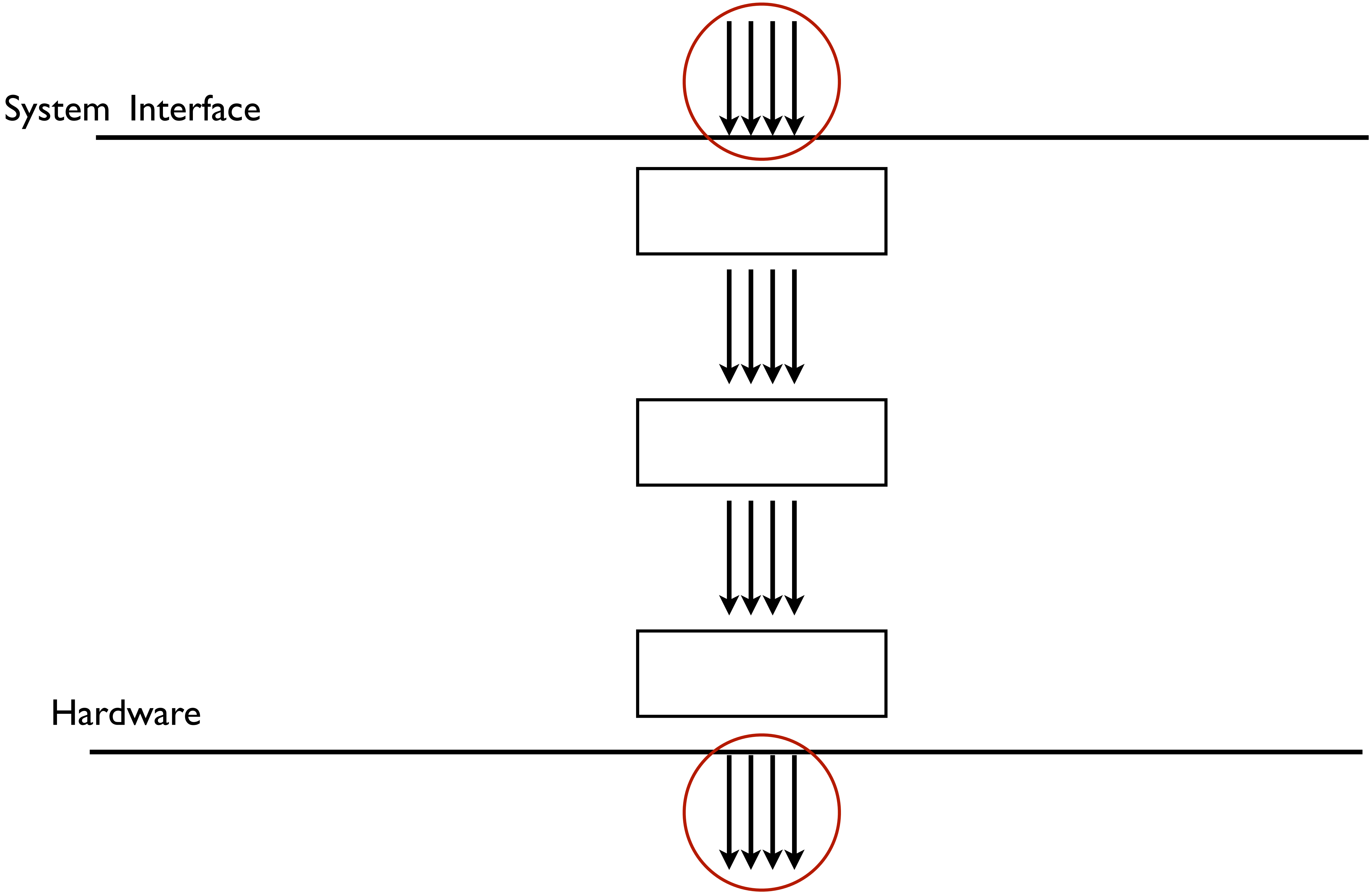
Software

Hardware

Top-Down Demand



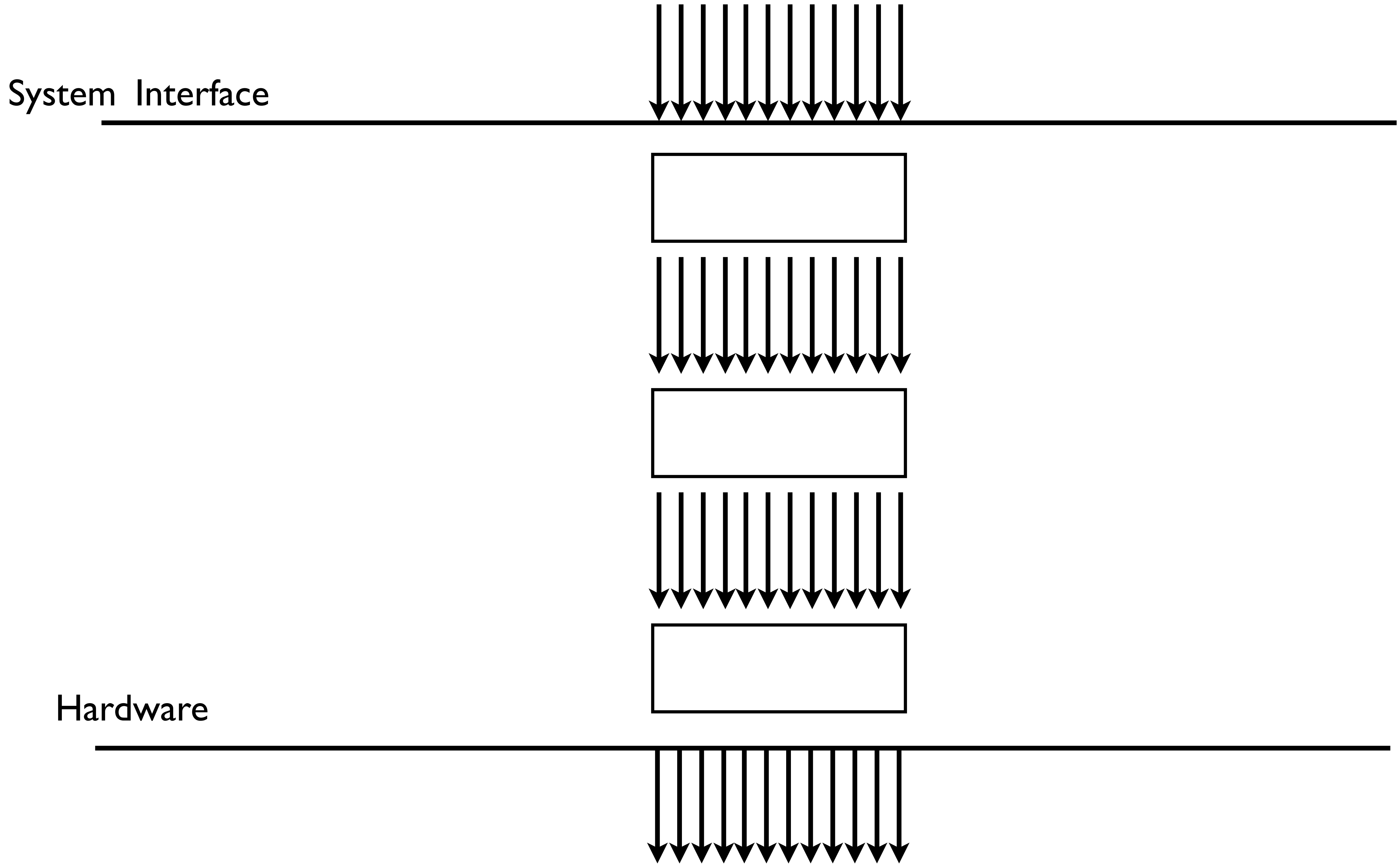
Top-Down Demand



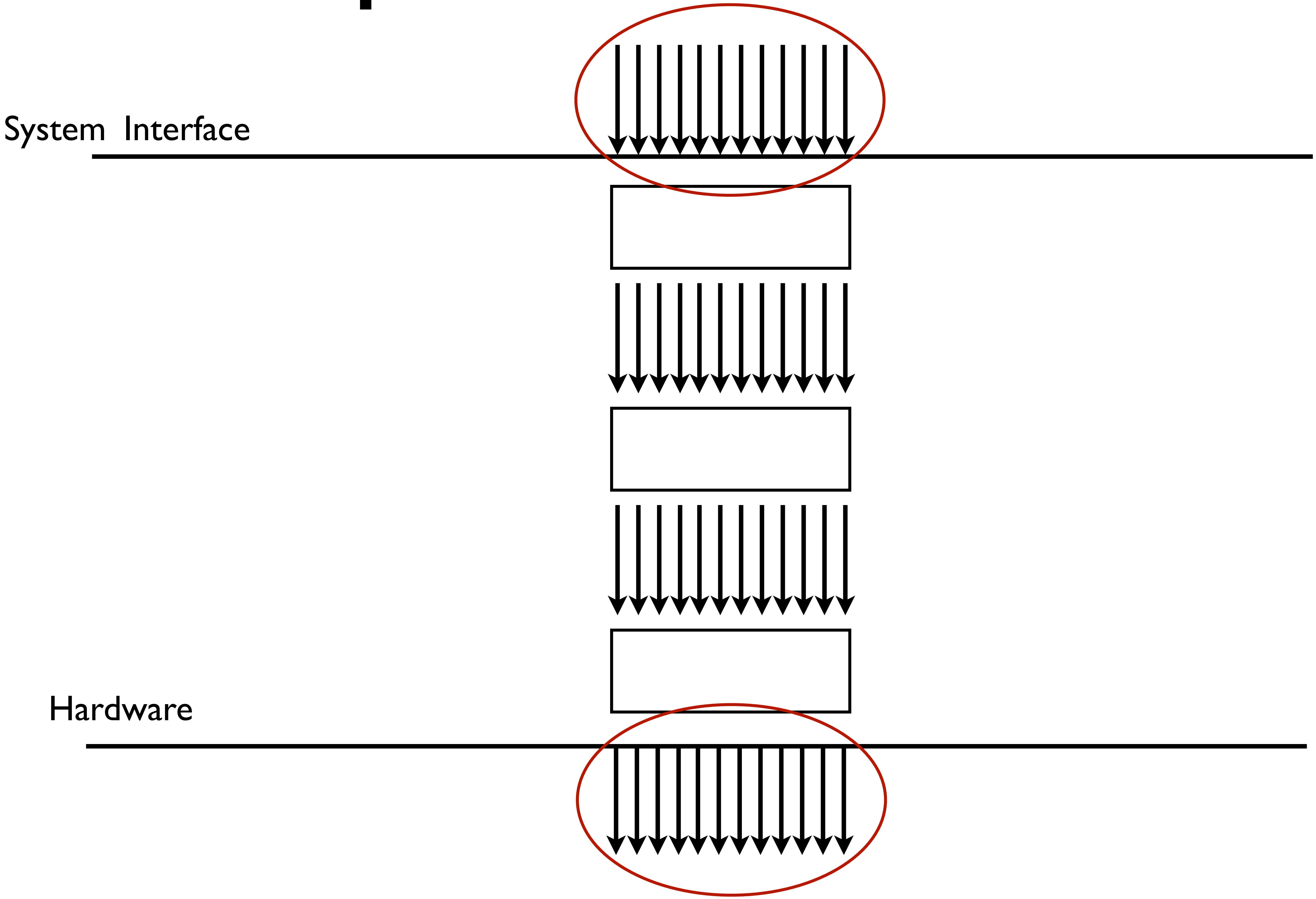
System Interface

Hardware

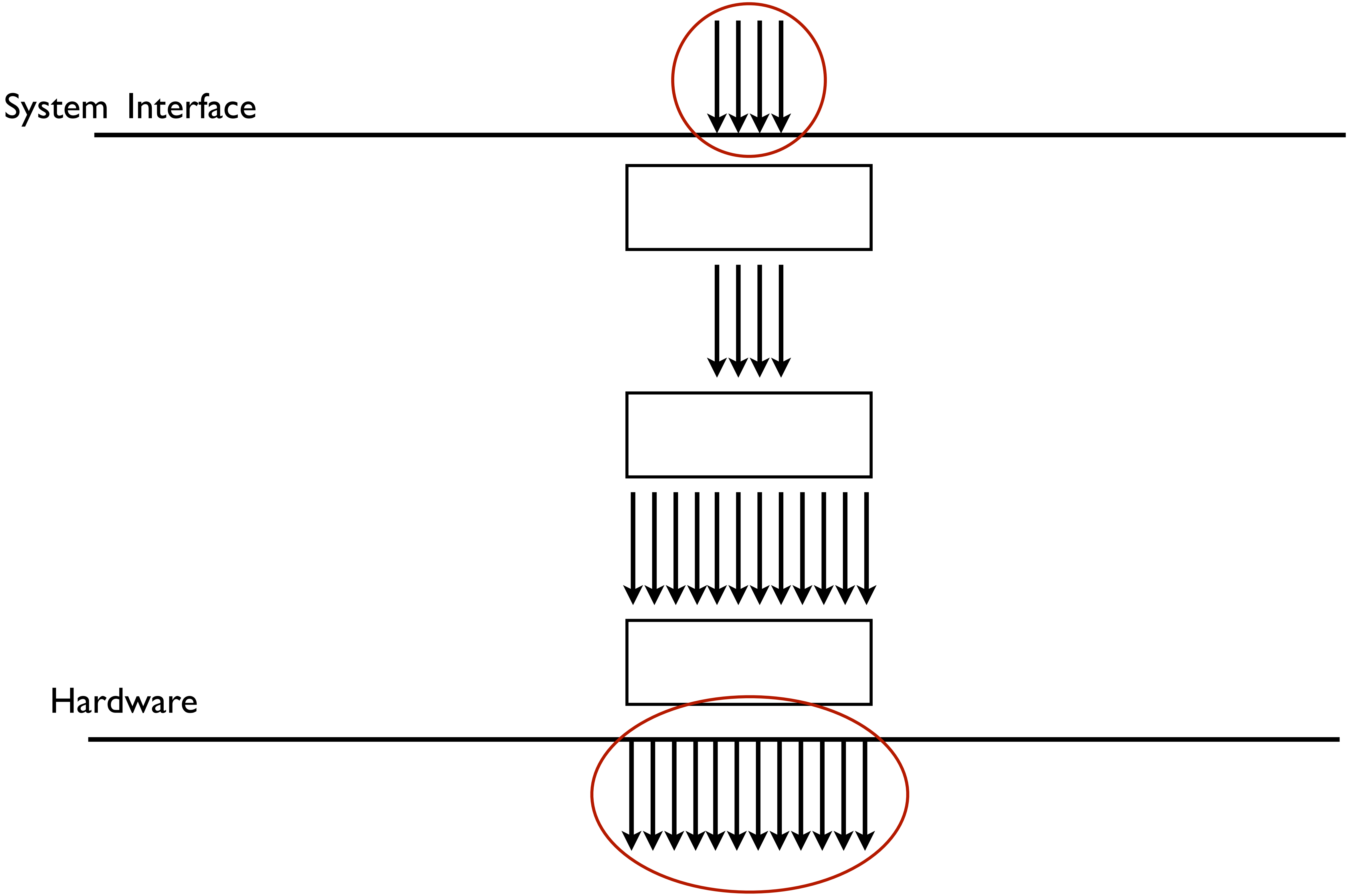
Top-Down Demand



Top-Down Demand



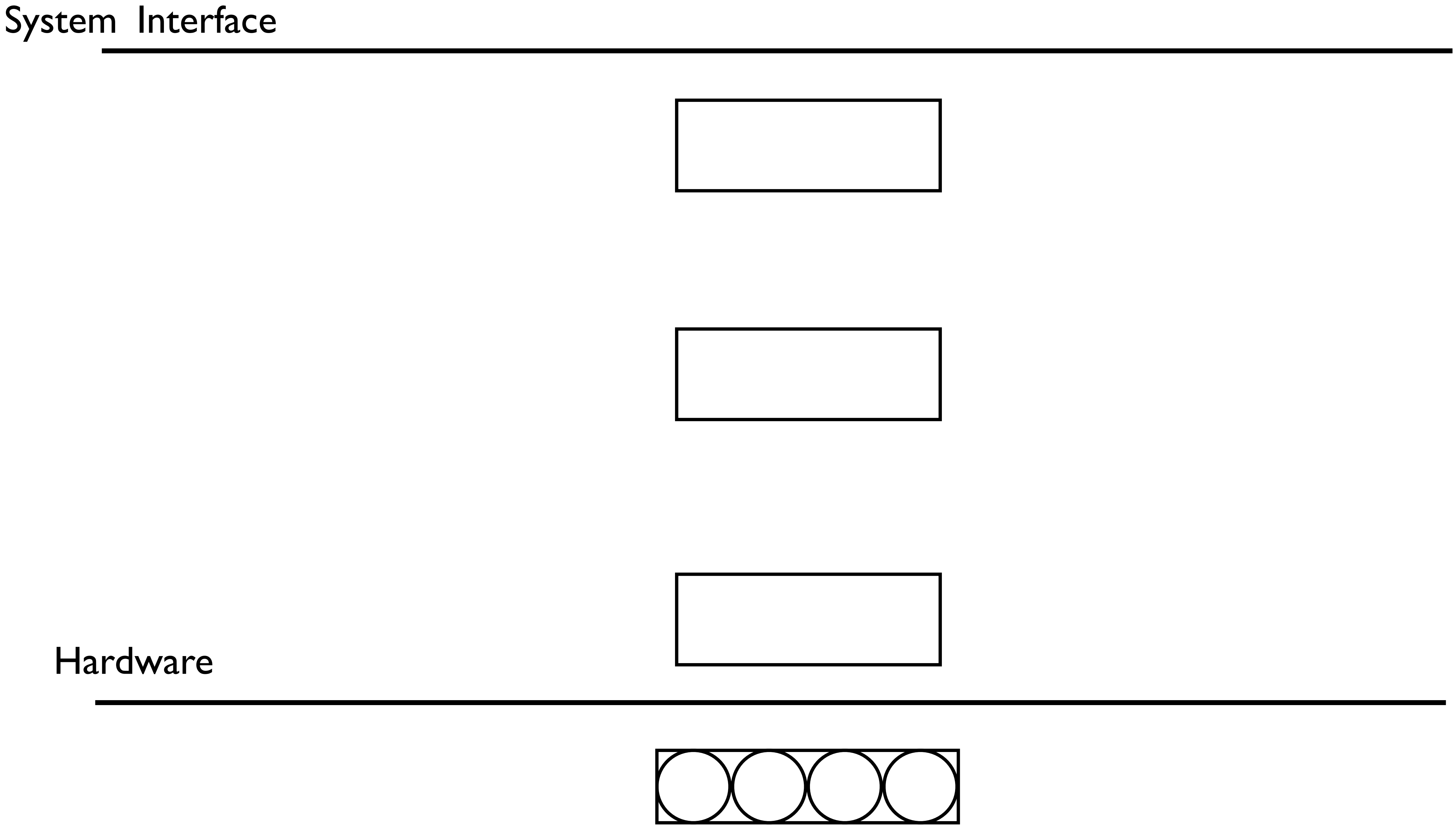
Top-Down Demand



Events as Load

- Treat a service request as an event that is dispatched to resources
- As events occur, load increases
- As events are handled, load decreases
- Each layer being event-driven forces demand to flow top-down

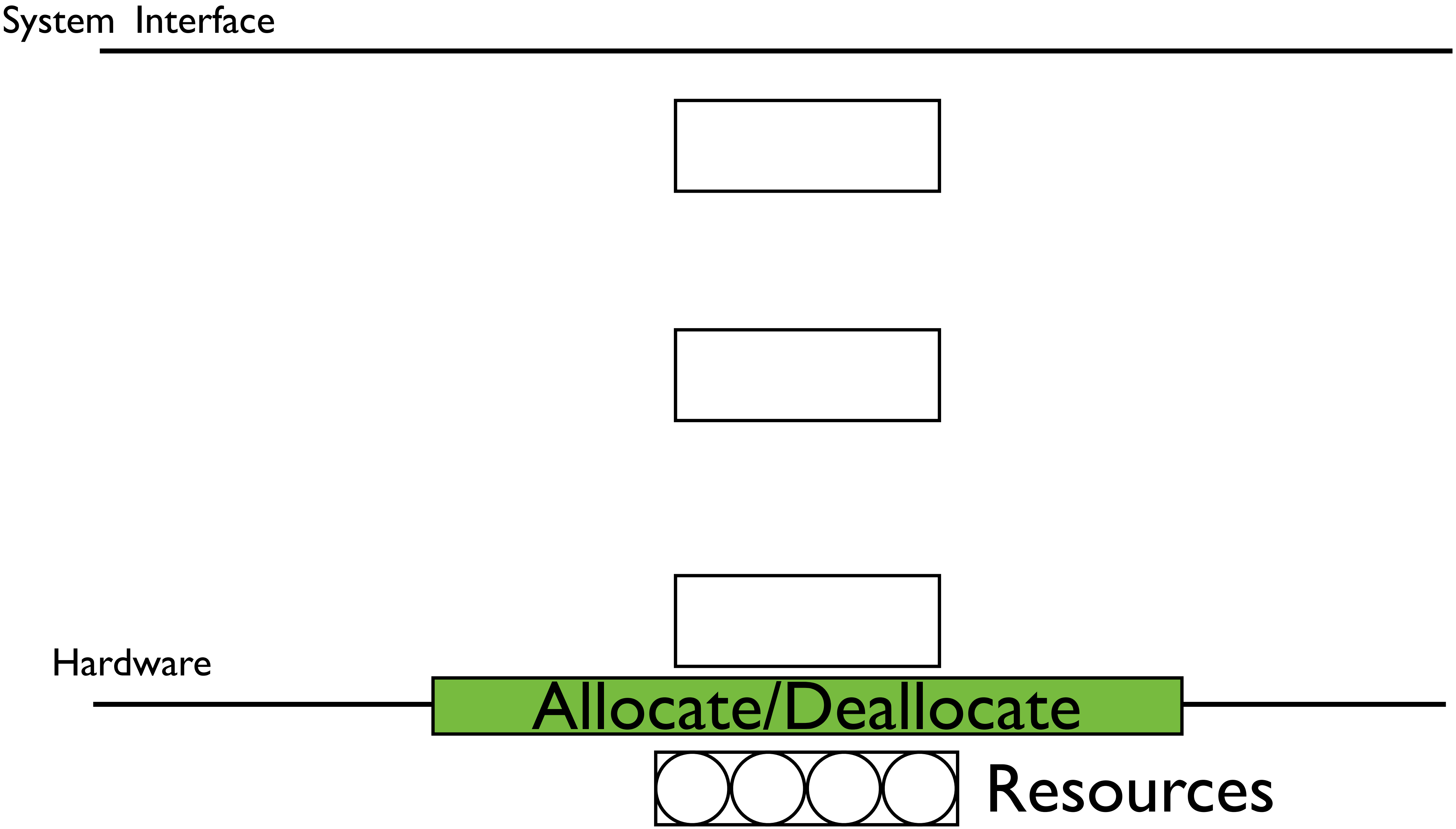
Bottom-Up Support



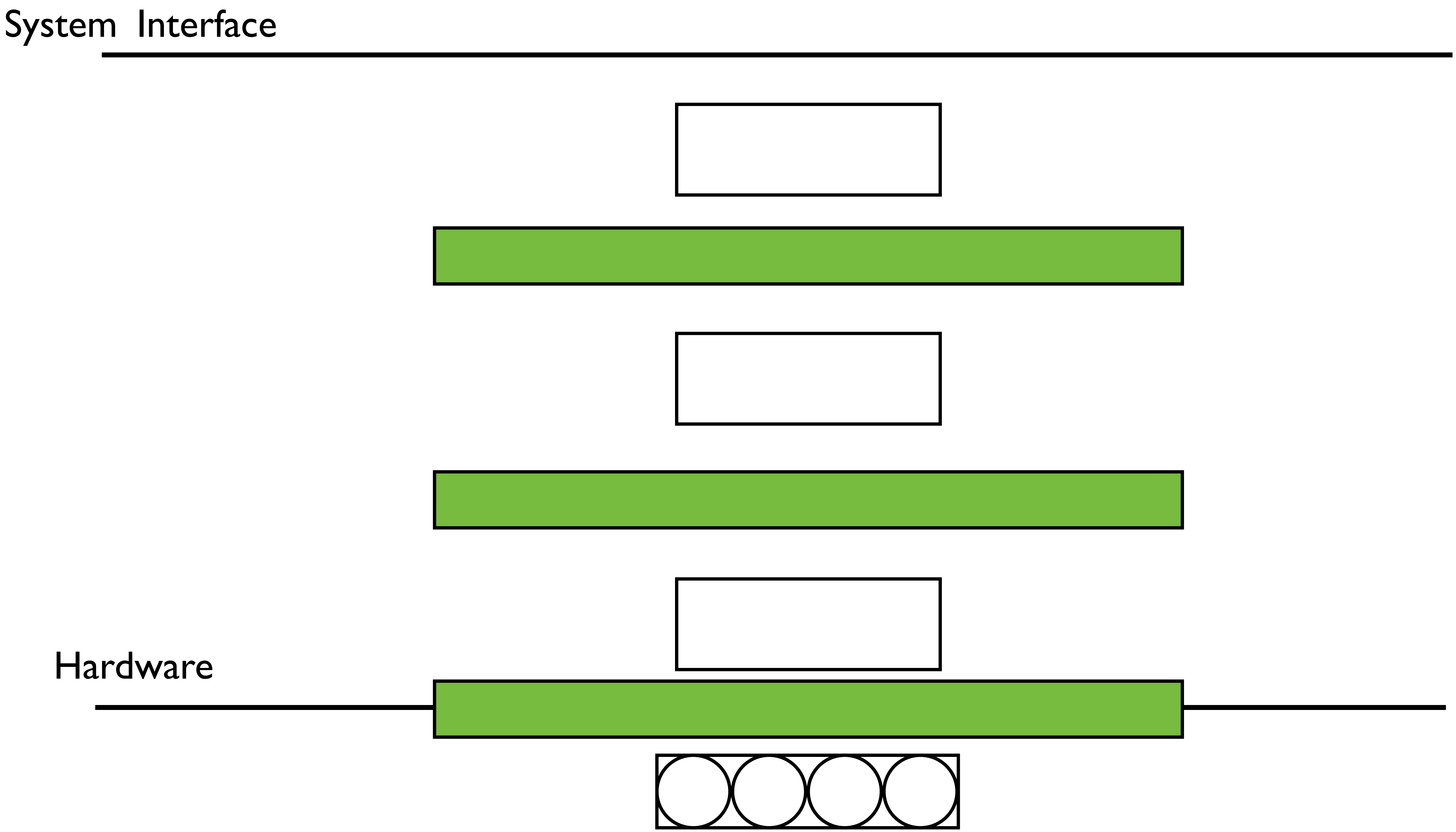
System Interface

Hardware

Bottom-Up Support



Bottom-Up Support



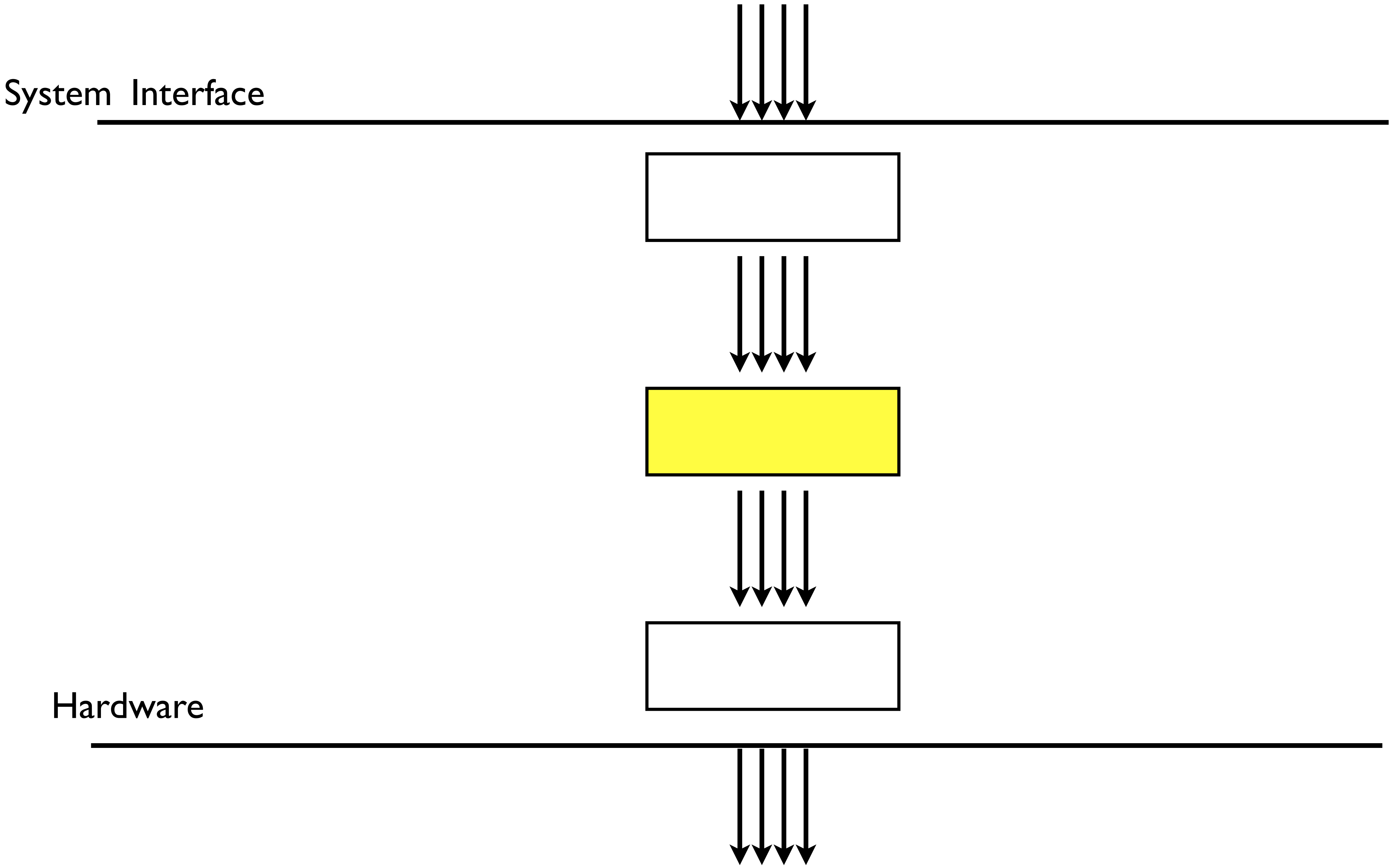
System Interface

Hardware

Elastic Interface

- Support elasticity by interfacing via allocation and deallocation of physical or logical resources
- Each layer is constructed by being explicit with respect to resource consumption
- Be explicit with respect to time to meet a request

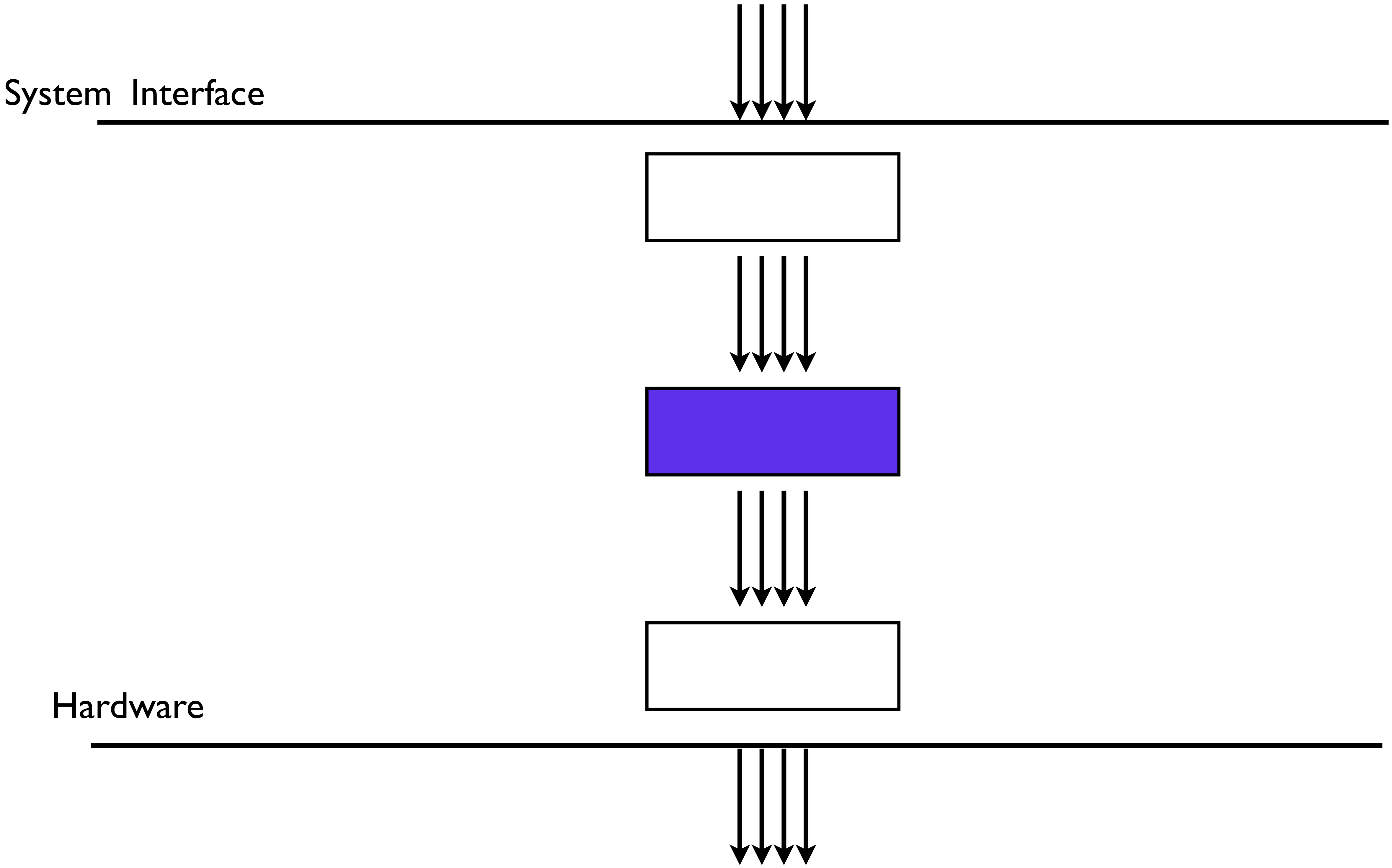
Modularity



System Interface

Hardware

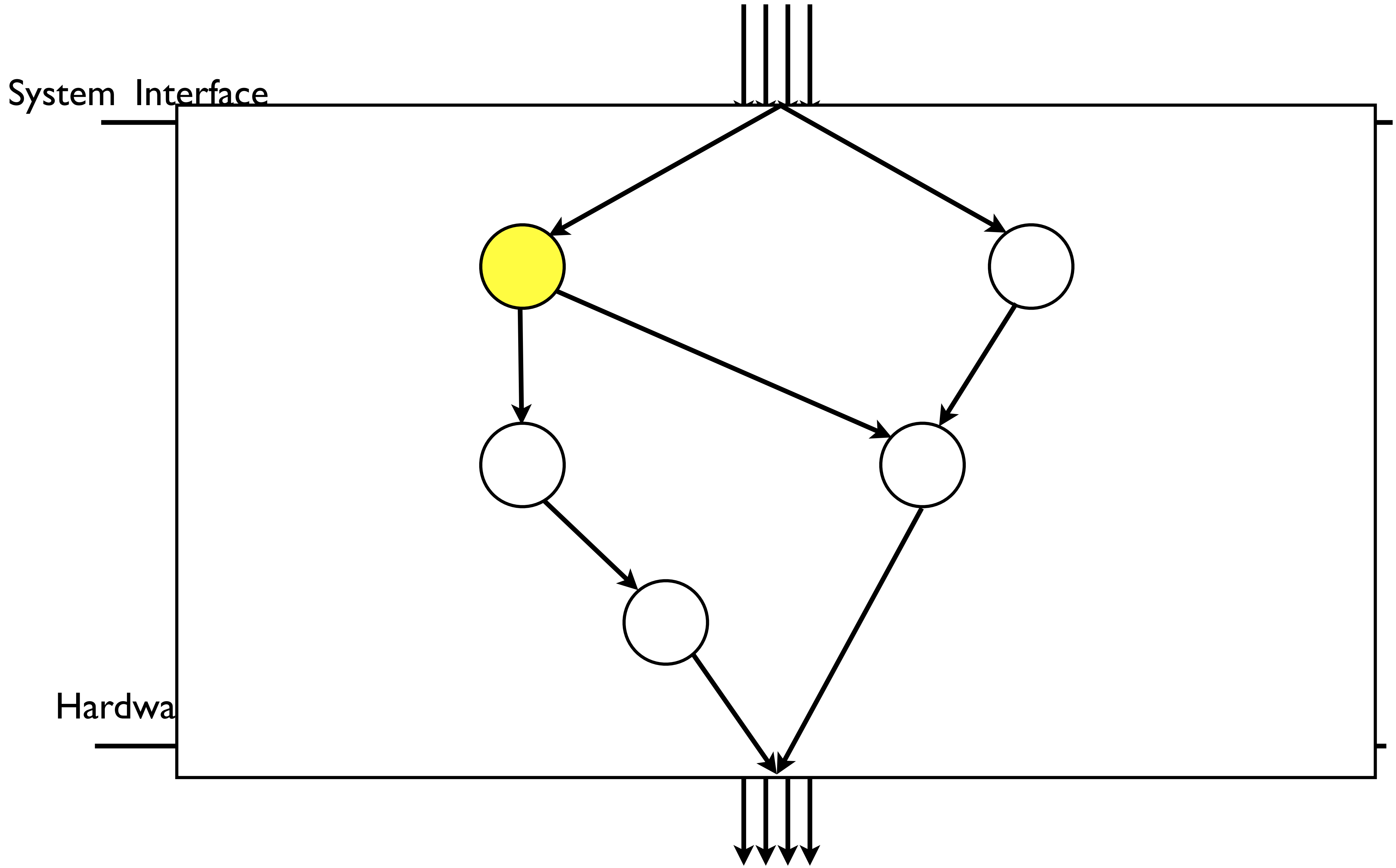
Modularity



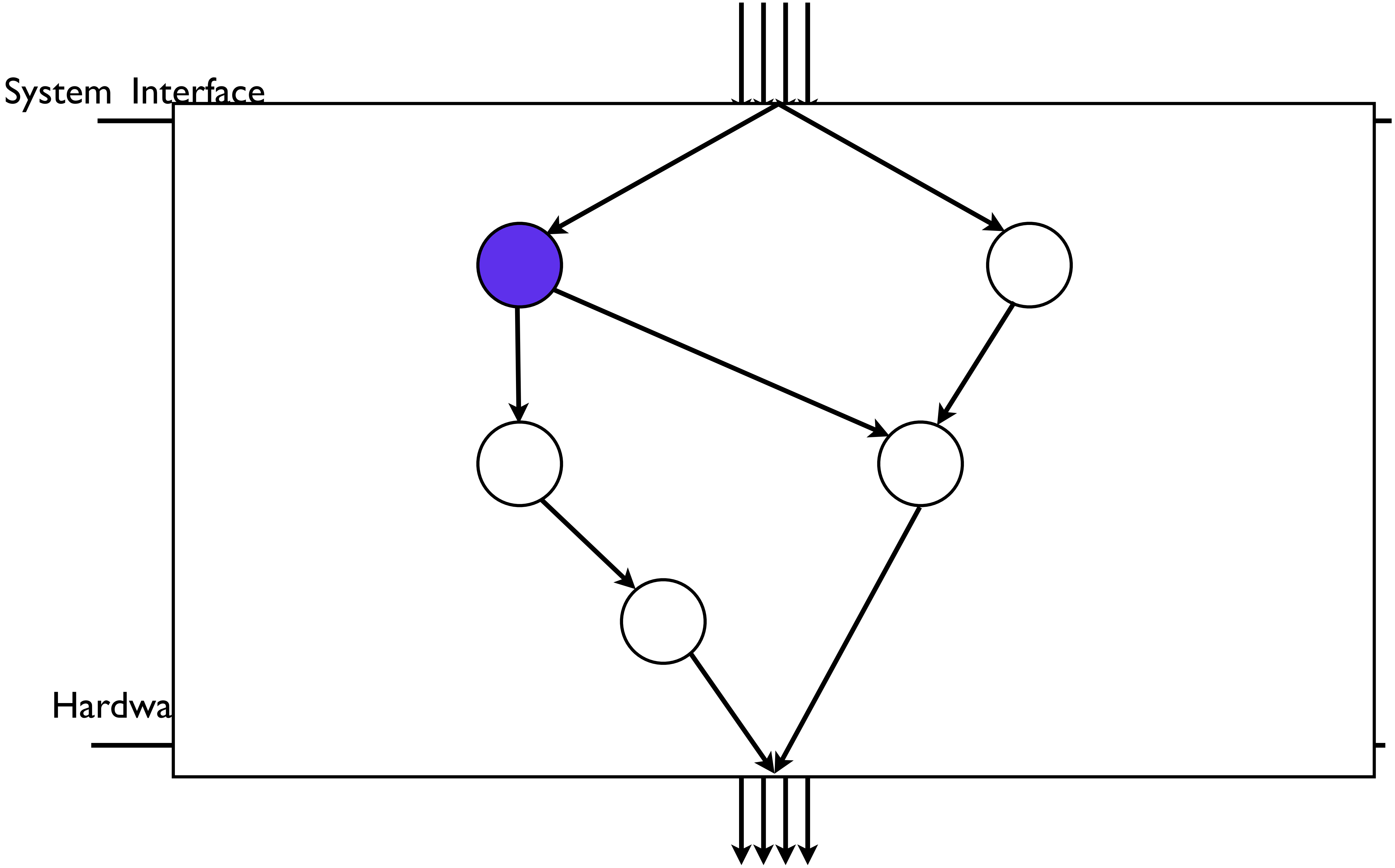
System Interface

Hardware

Modularity



Modularity



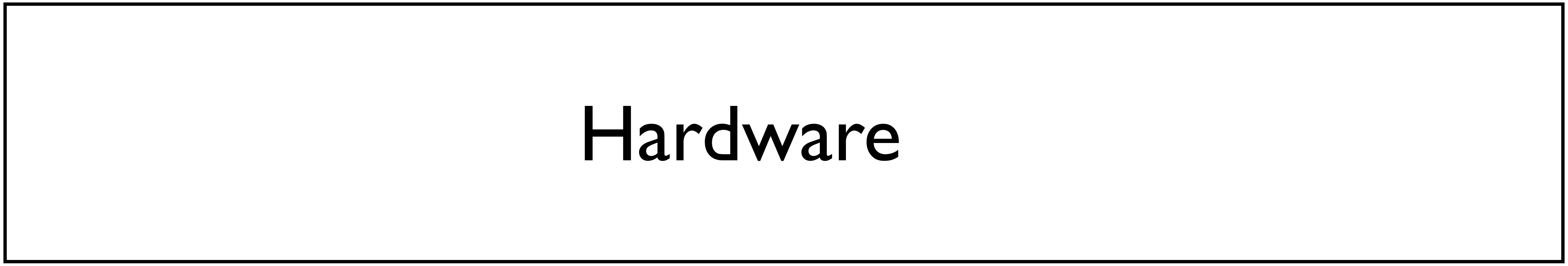
Object model

- Objects can take advantage
  - the semantics of their request patterns
  - the lifetime of an instance
  - the occupancy w.r.t memory, processing and communication
- We can optimize for elasticity by taking advantage of modularity in a system

OUTLINE

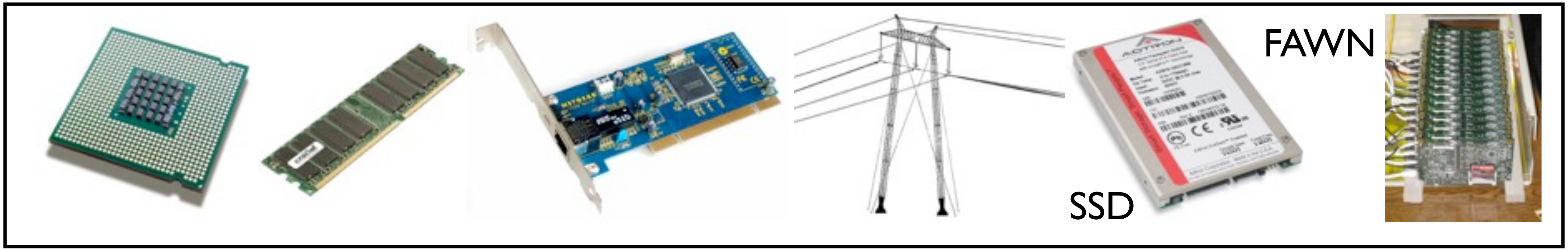
1. THE PROBLEM
2. OBSERVATIONS
3. OUR TAKE ON A SOLUTION : SESA
    1. EBB’s : Elastic Building Blocks
    2. SEE: Scalable Elastic Executive - A LibOS
    3. EPIC: Events as Interrupts
4. PROTOTYPE & CHALLENGES

Architecture Overview



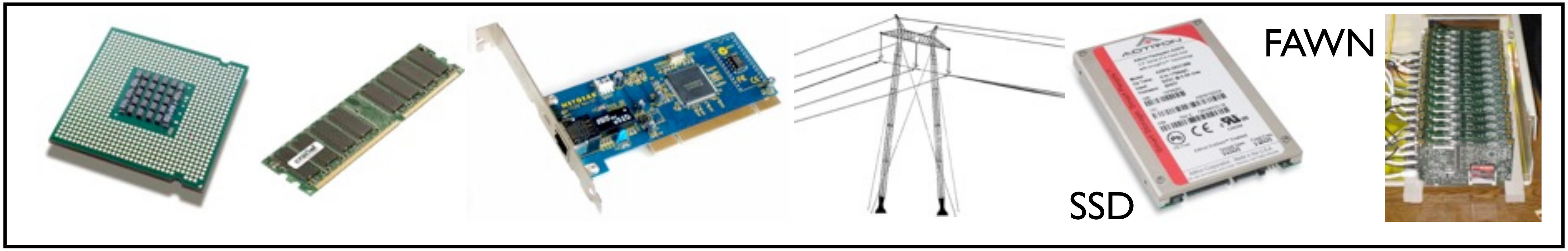
Hardware

Architecture Overview

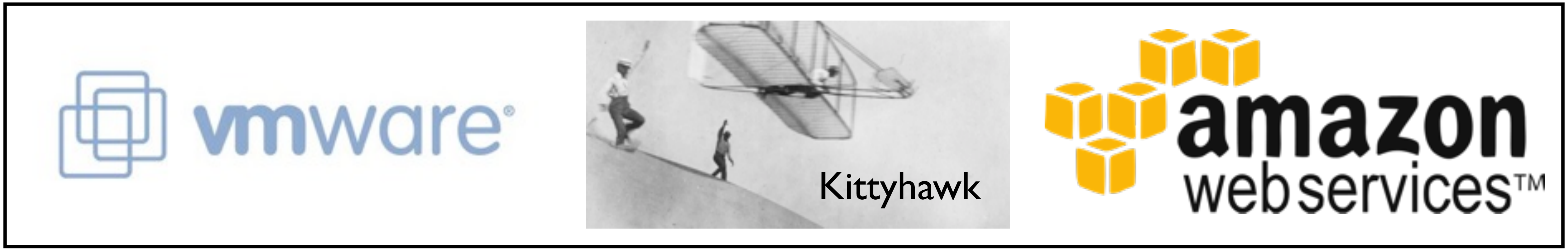


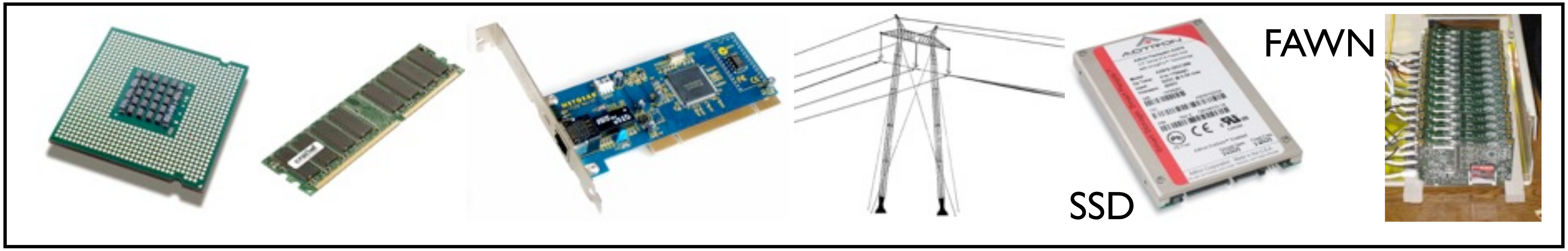
Architecture Overview

Partitioning

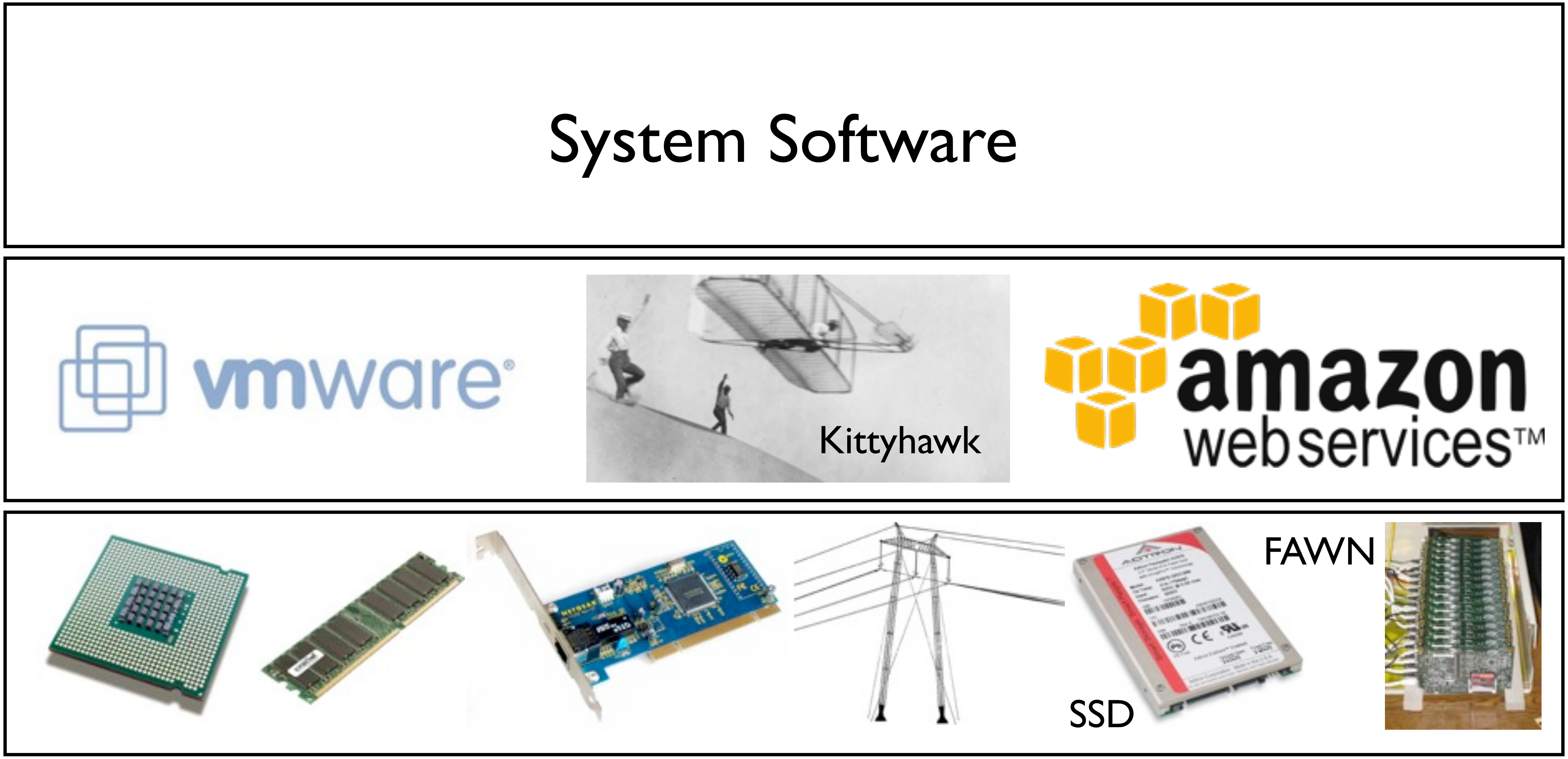


Architecture Overview





Architecture Overview



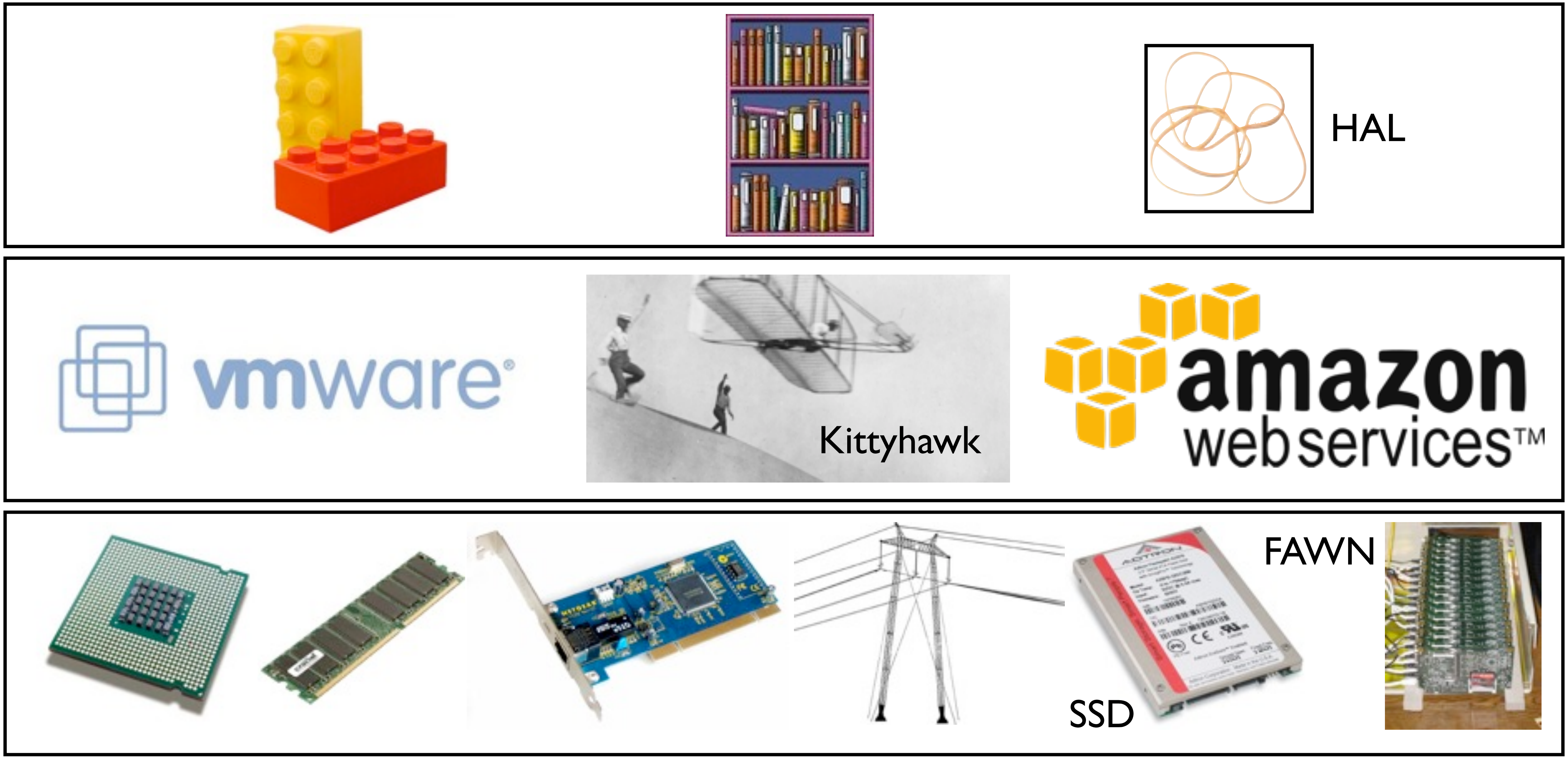
System Software

Kittyhawk

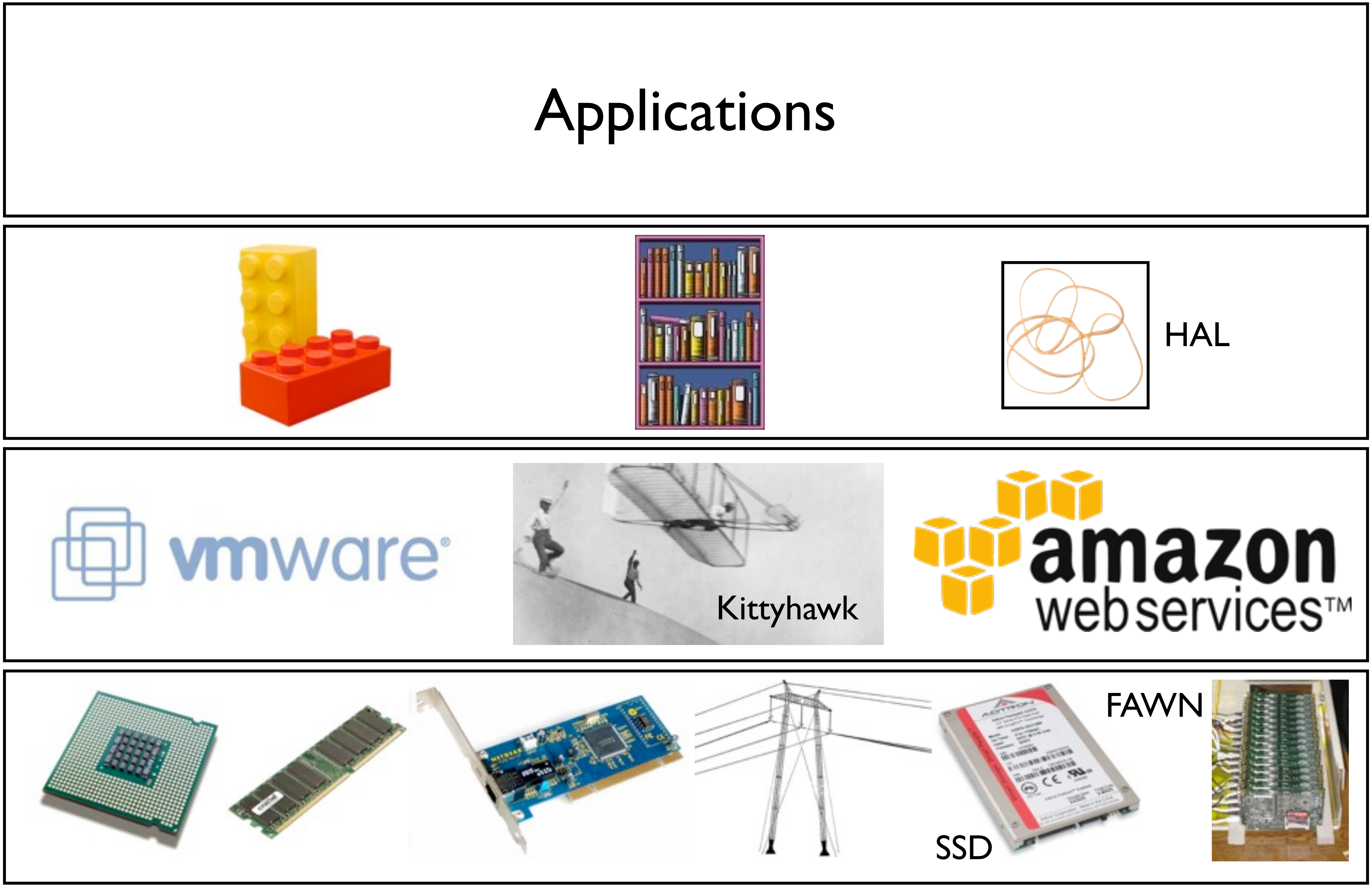
SSD

FAWN

Architecture Overview



Architecture Overview



Applications

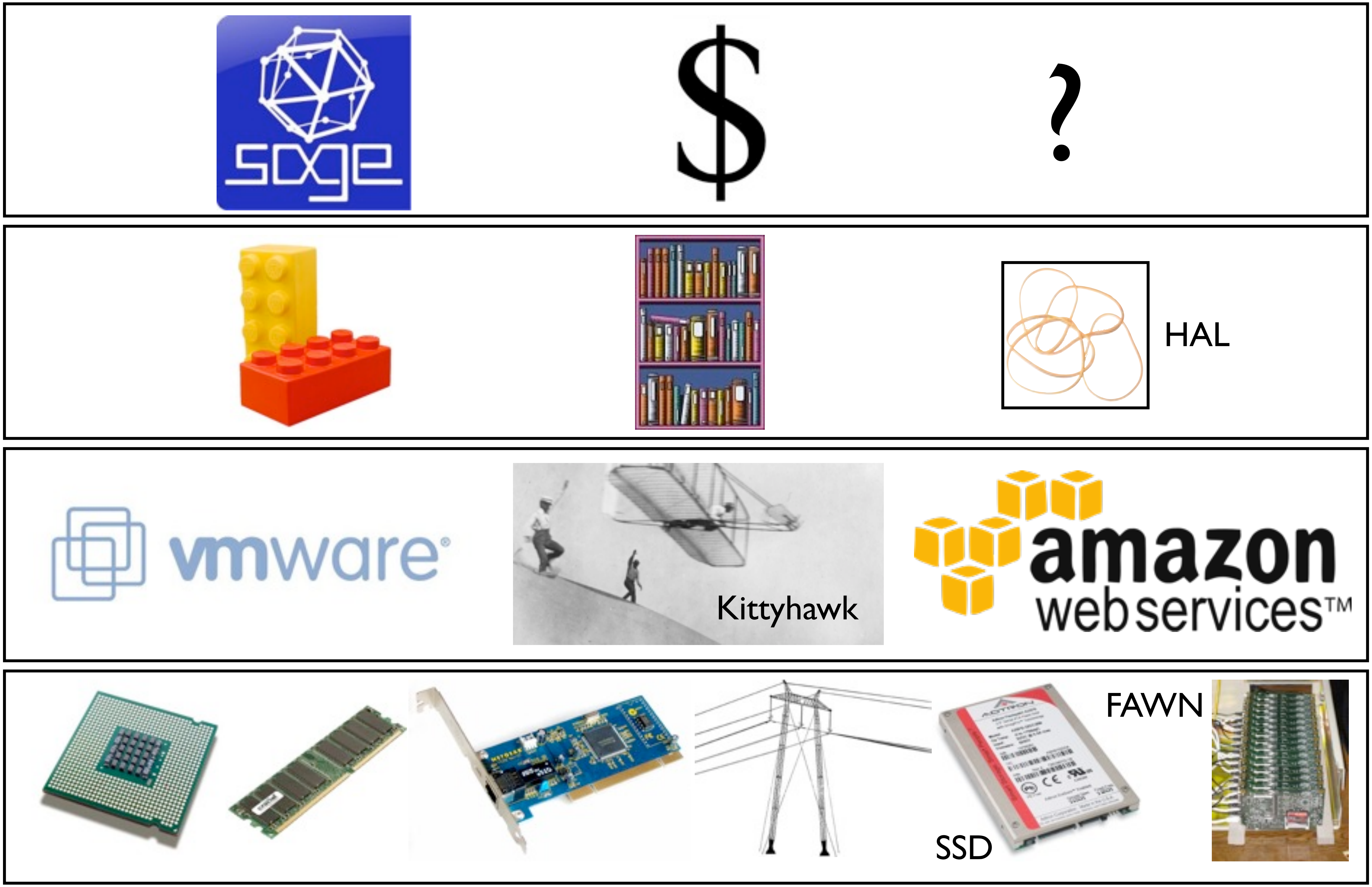
HAL

Kittyhawk

FAWN

SSD

Architecture Overview



$

?

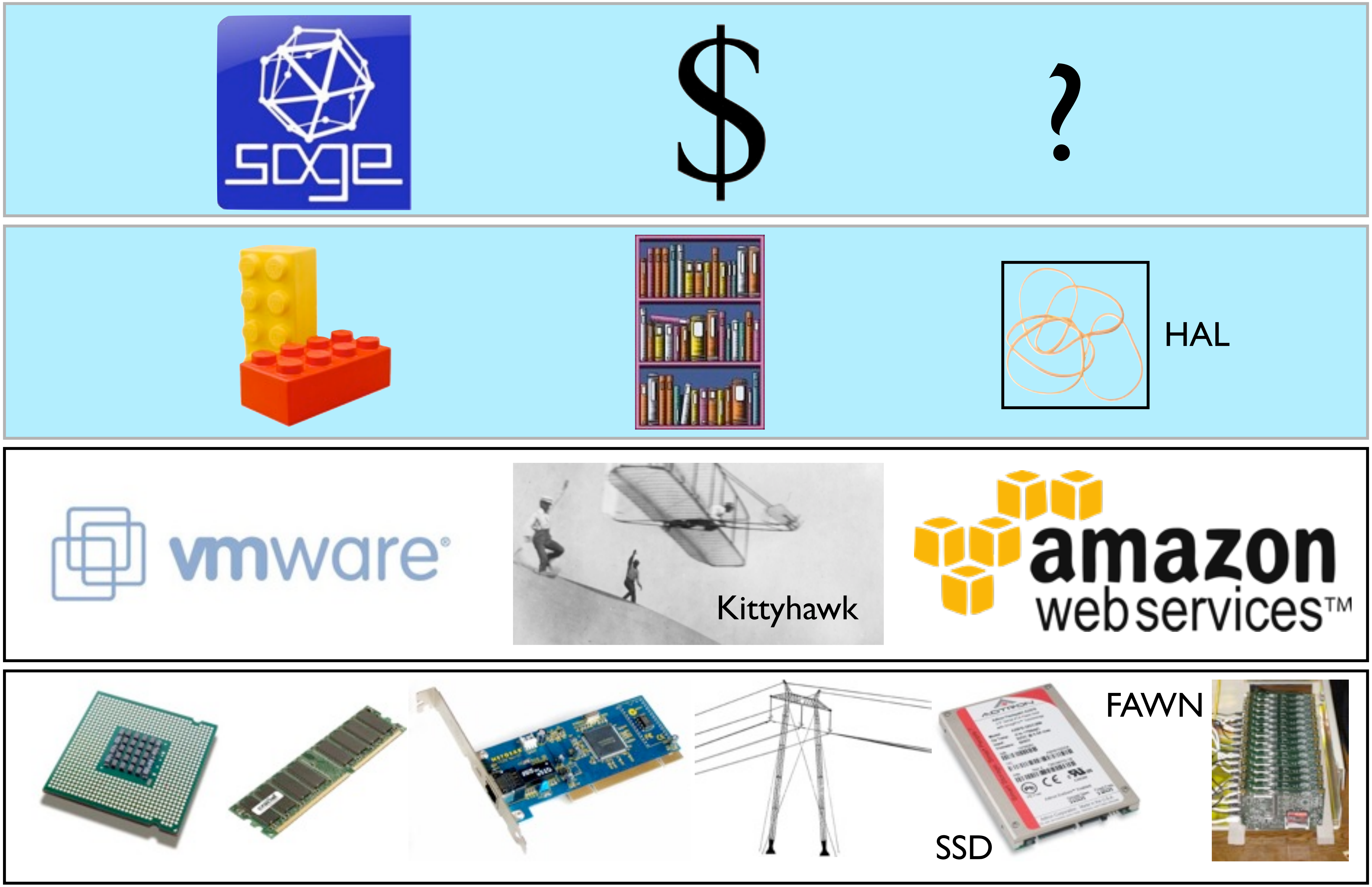
HAL

Kittyhawk

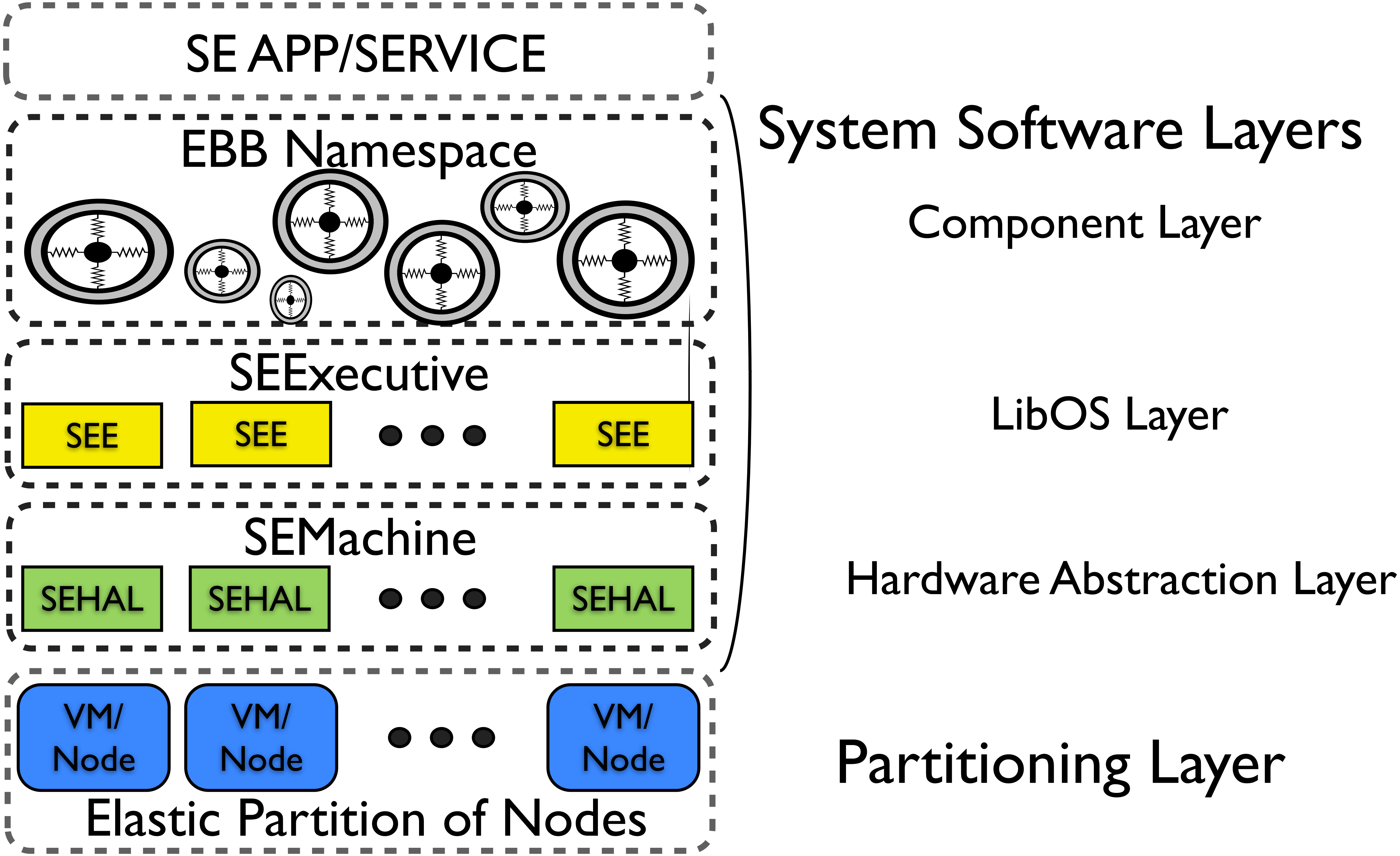
FAWN

SSD

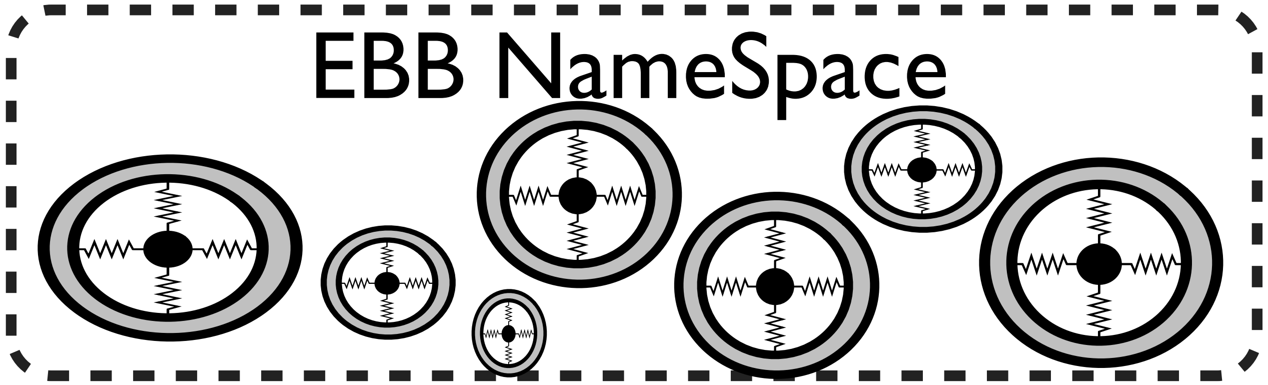
Architecture Overview



SESA


SE APP/SERVICE
EBB Namespace
SEExecutive
SEE
SEE
SEE
SEMachine
SEHAL
SEHAL
SEHAL
VM/ Node
VM/ Node
VM/ Node
Elastic Partition of Nodes
System Software Layers
Component Layer
LibOS Layer
Hardware Abstraction Layer
Partitioning Layer

EBB's

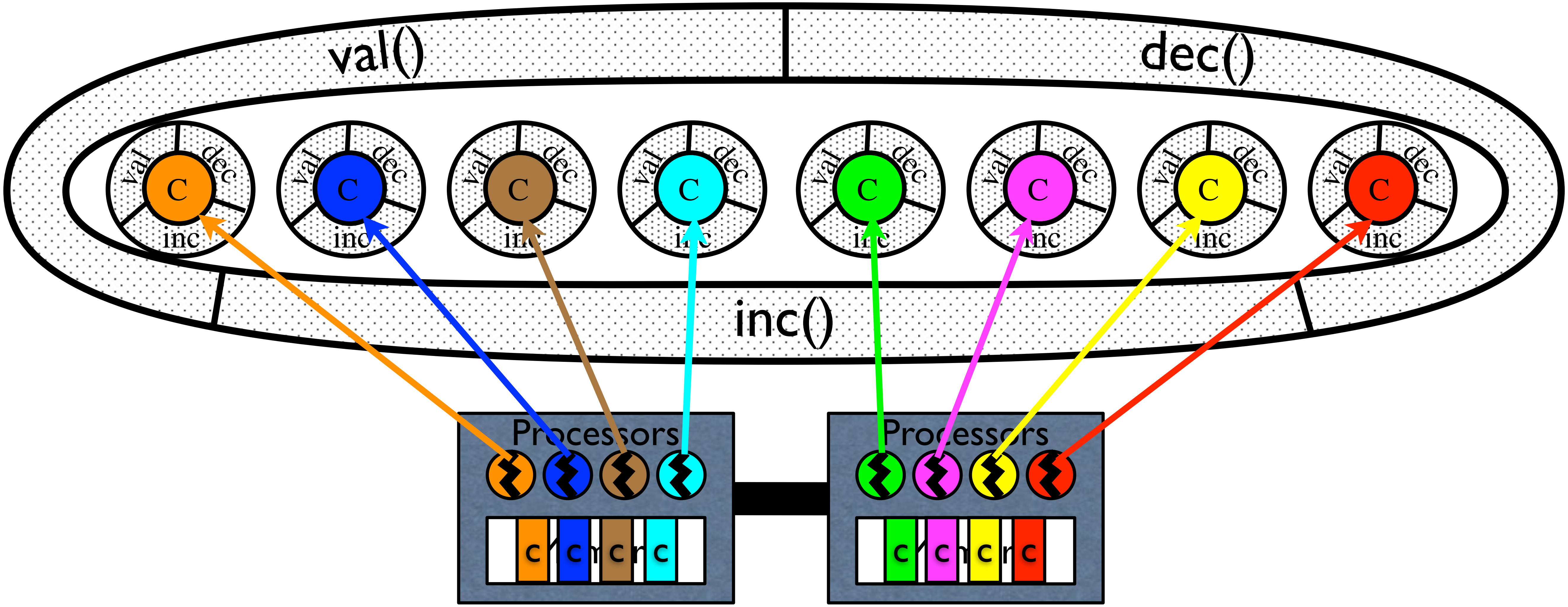


A new Component Model for expressing and encapsulating fine grain elasticity.

The Next Generation of Clustered Objects.

Clustered Objects (CO)

```
dref(ctr)->inc();
```

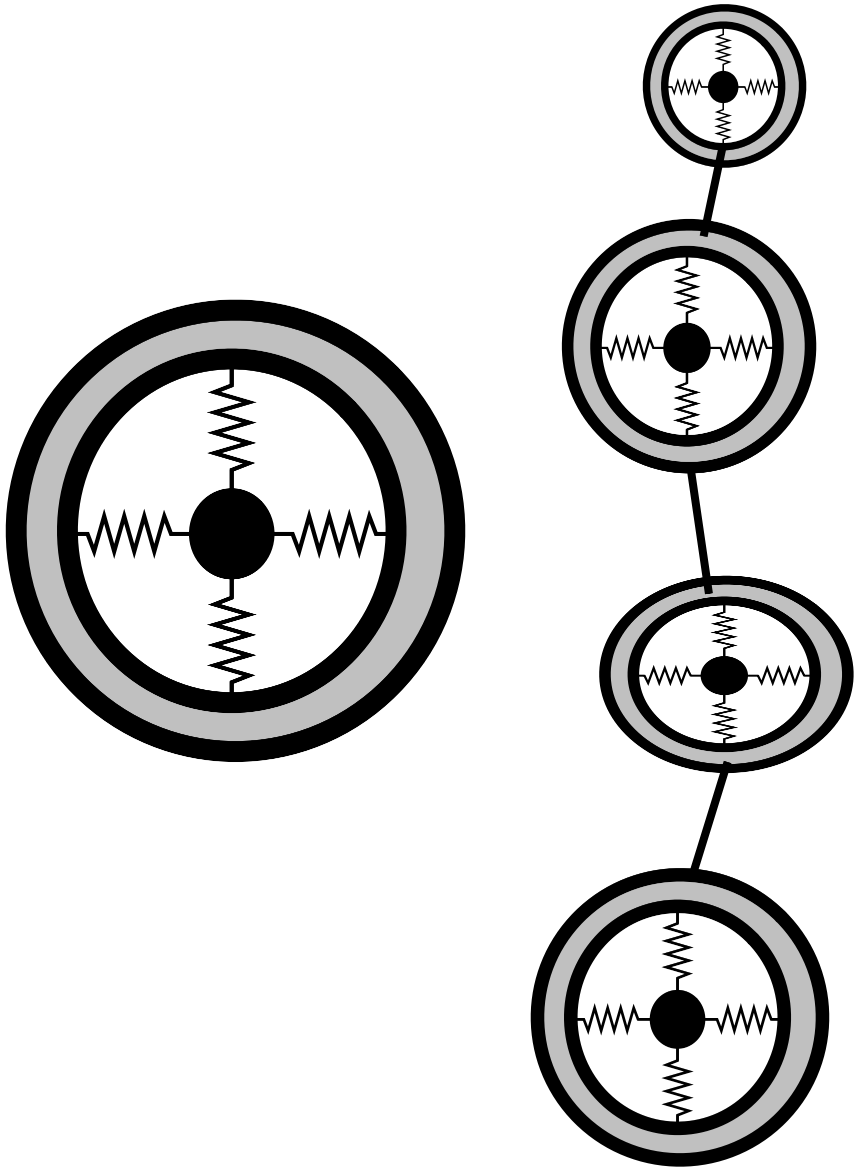


What did we learn?

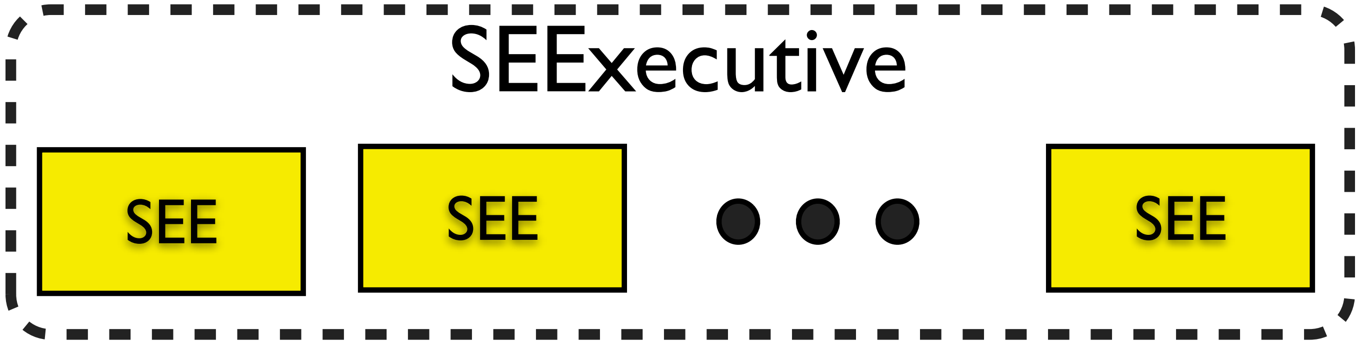
- Event-driven architecture for lazy and dynamic instantiation of resources
- Mechanism to create scalable software

Elastic Building Blocks

- Programming Model for Elastic and Scalable Components
- Span multiple nodes
- Built in On Demand nature -- encapsulation of policies for both allocation and **deallocation** of resources



SEE

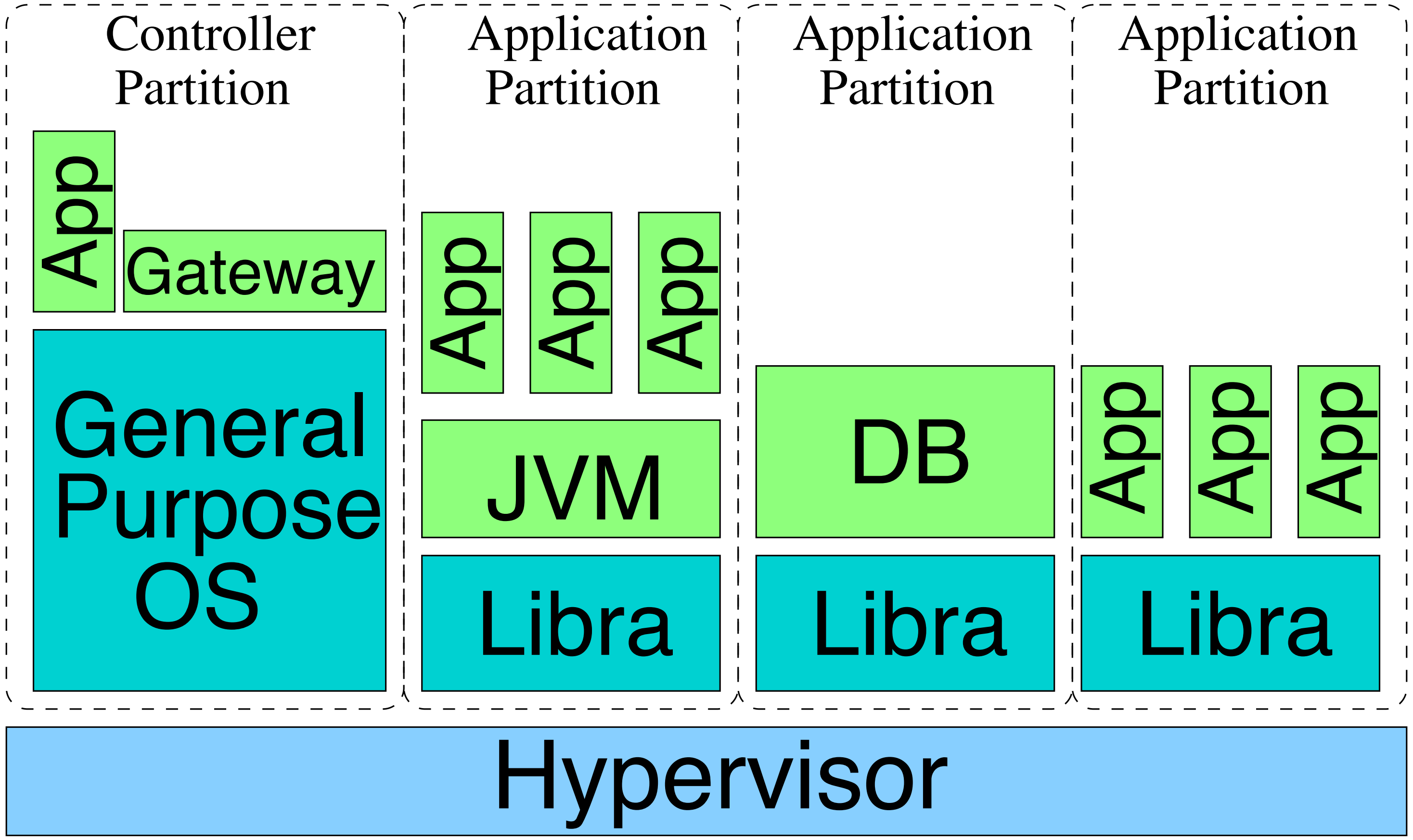


A Distributed Library OS Model designed to enable Elastic Software within the context of legacy environments.

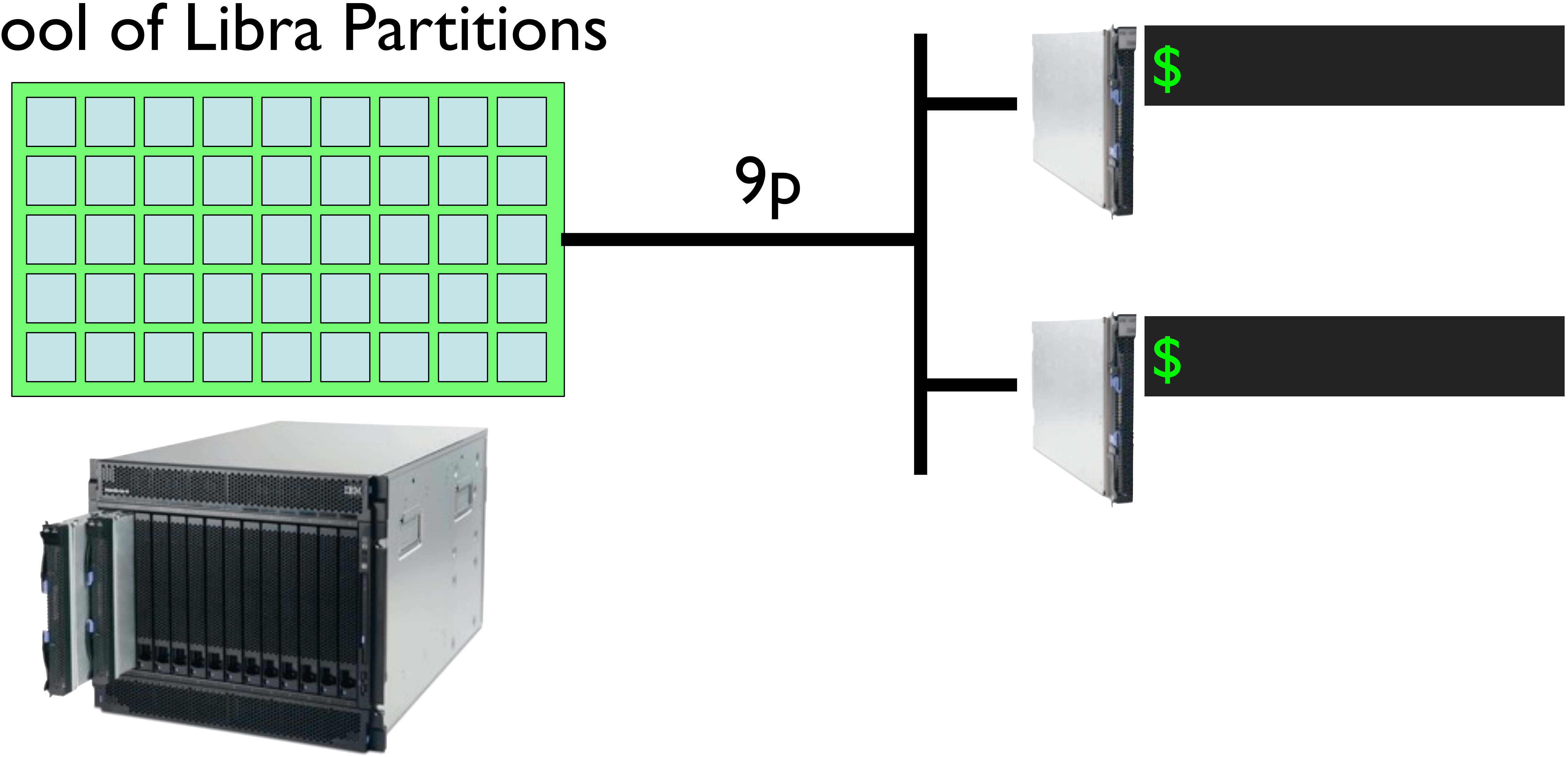
Next Generation of Libra

Libra

Architecture



Libra



X86 Linux Front Ends

Pool of Libra Partitions

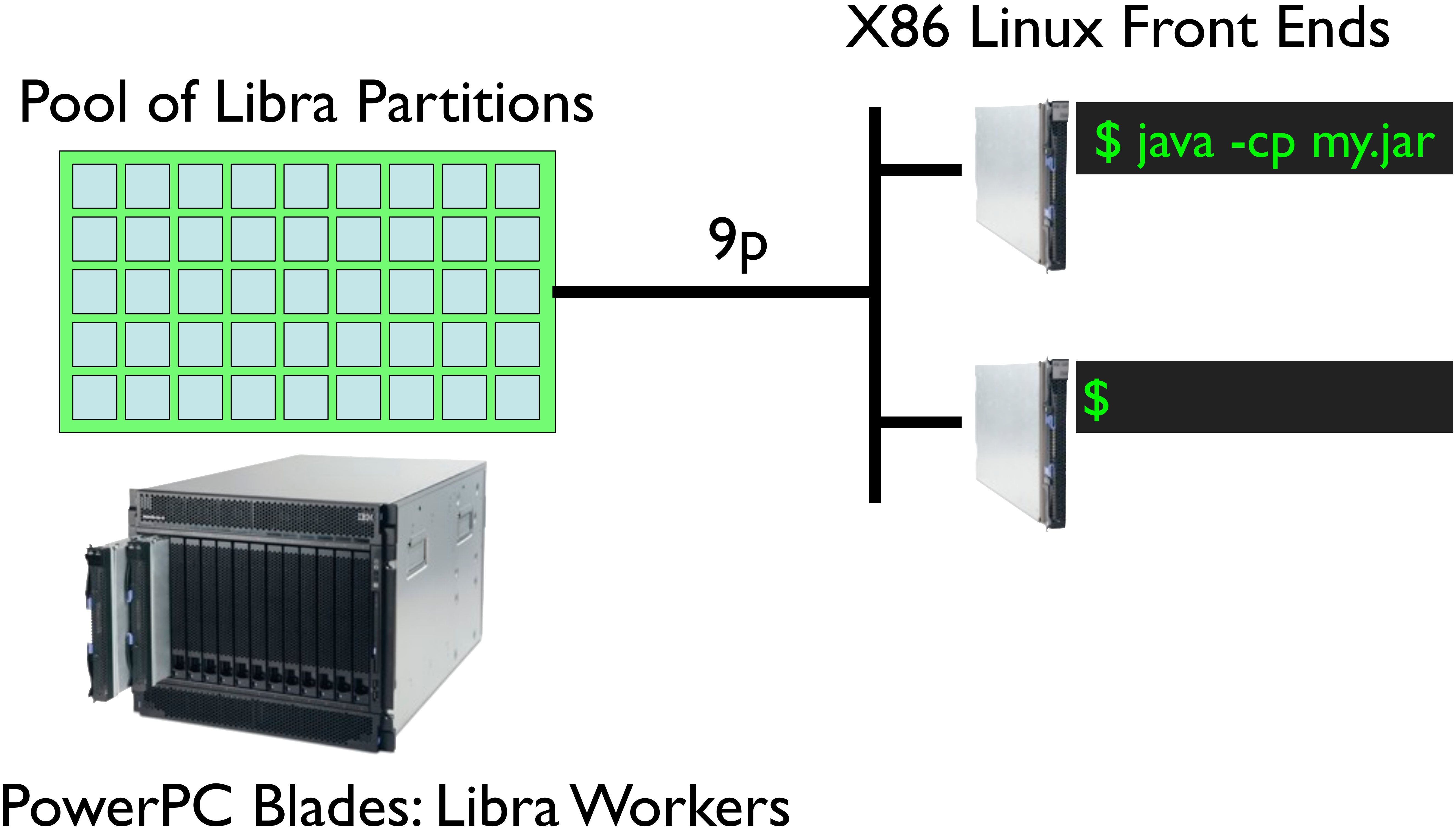
9p

```
$
```

```
$
```

PowerPC Blades: Libra Workers

Libra



X86 Linux Front Ends

Pool of Libra Partitions

9p

```
$ java -cp my.jar
```

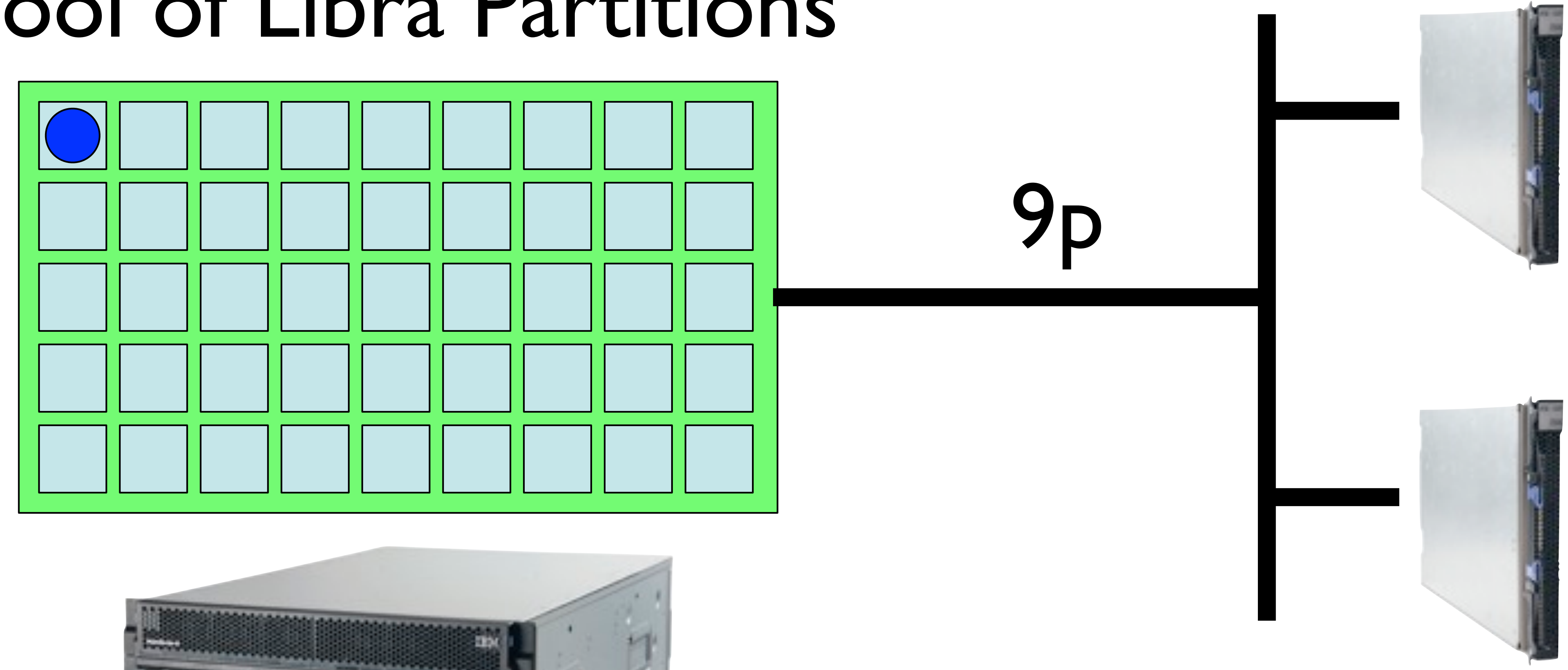
```
$
```

PowerPC Blades: Libra Workers

Libra

X86 Linux Front Ends

Pool of Libra Partitions



```
$ java -cp my.jar
```

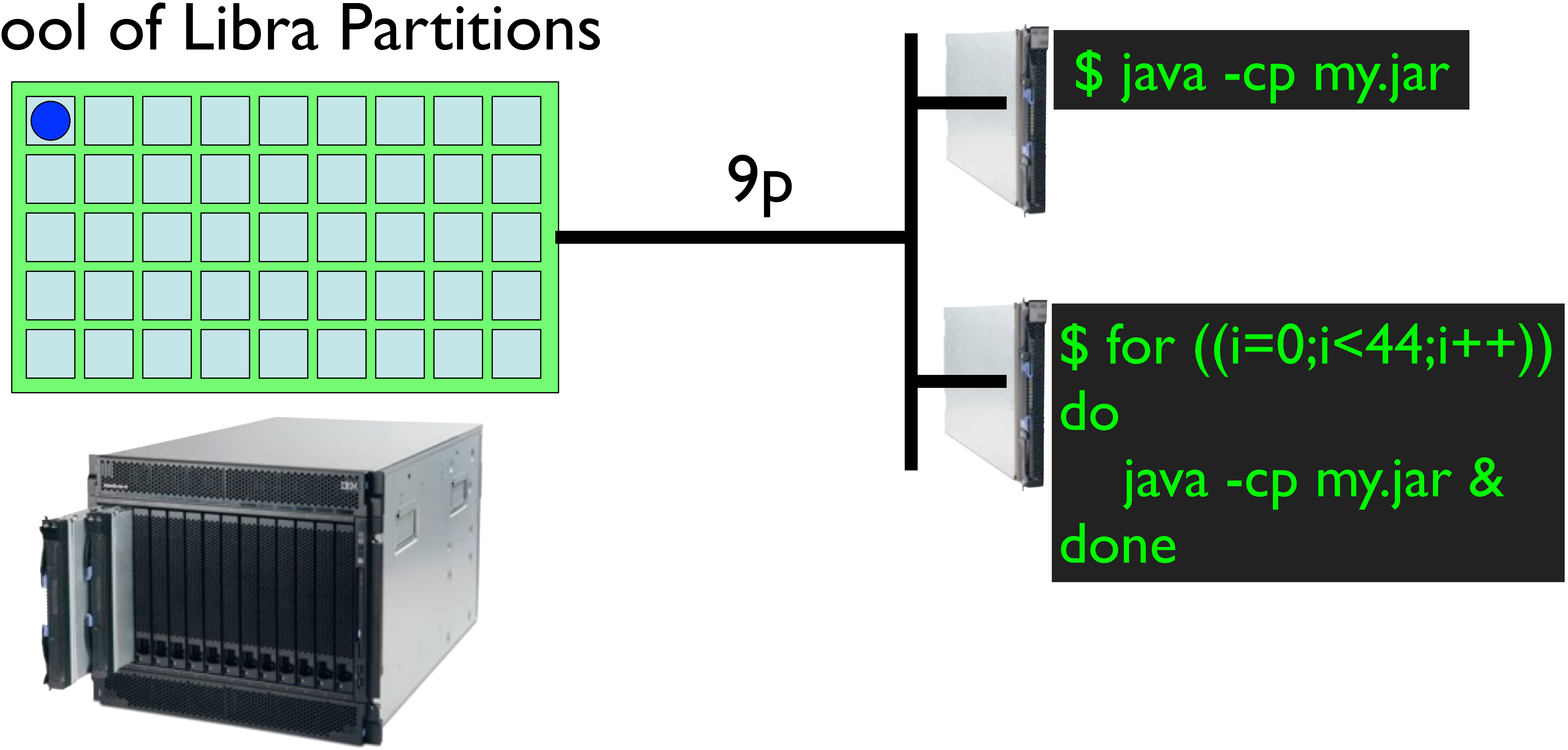
```
$
```

PowerPC Blades: Libra Workers

Libra

X86 Linux Front Ends

Pool of Libra Partitions



9p

```
$ java -cp my.jar
```

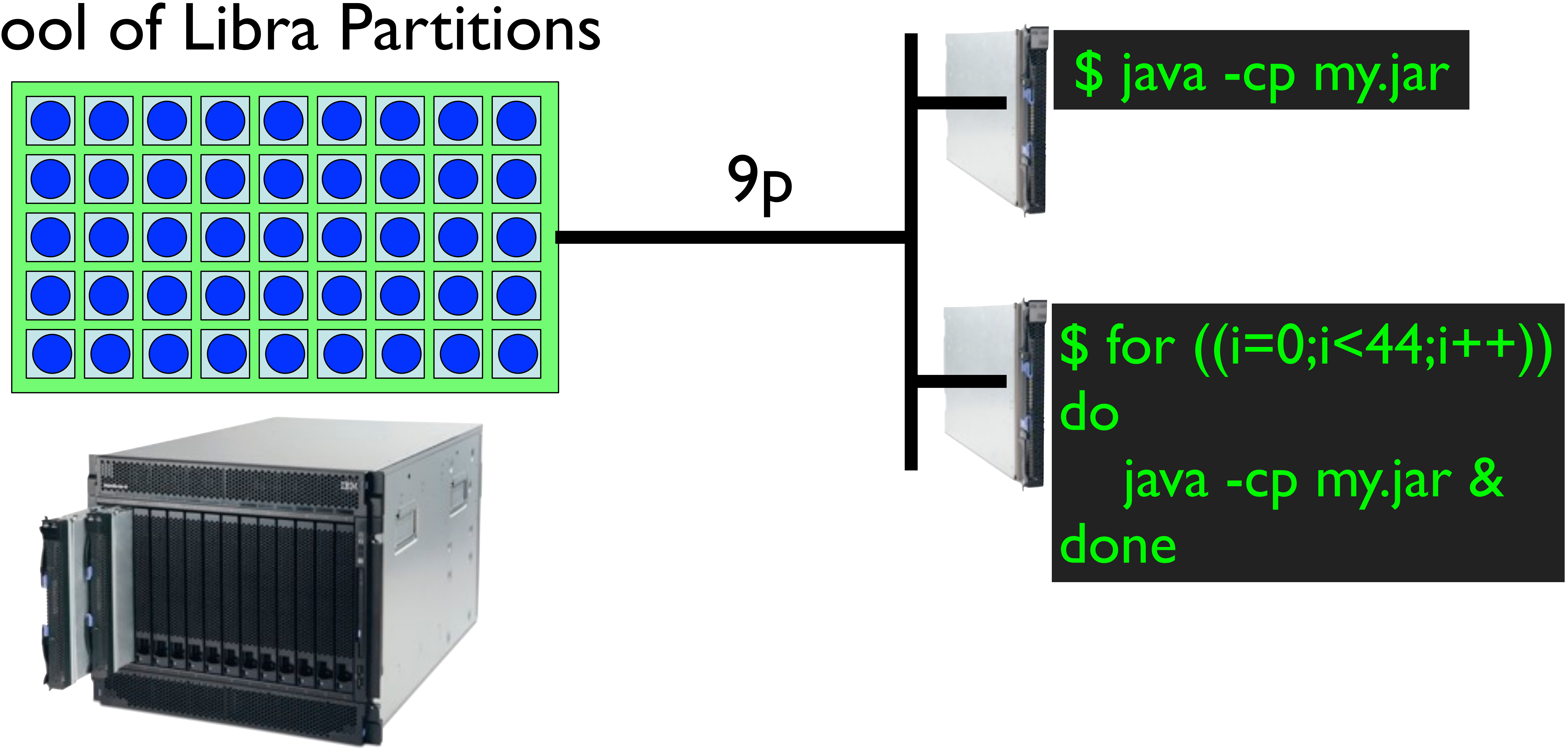
```
$ for ((i=0;i<44;i++))
do
    java -cp my.jar &
done
```

PowerPC Blades: Libra Workers

Libra

X86 Linux Front Ends

Pool of Libra Partitions



```
$ java -cp my.jar
```

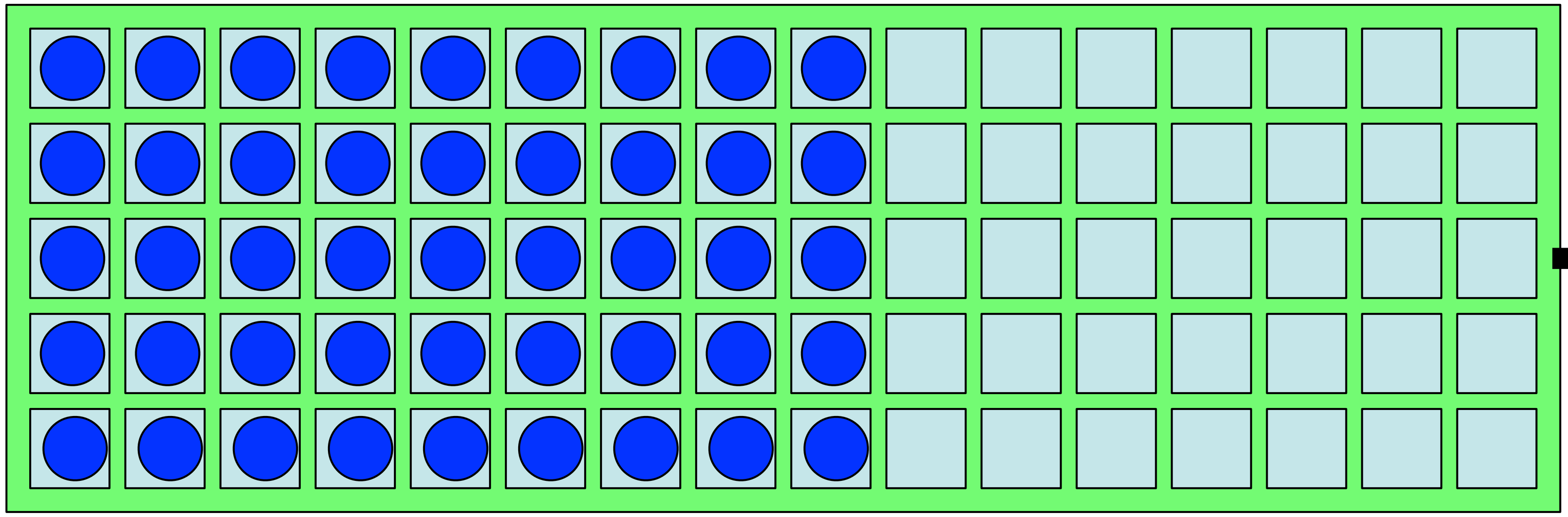
```
$ for ((i=0;i<44;i++))
do
    java -cp my.jar &
done
```

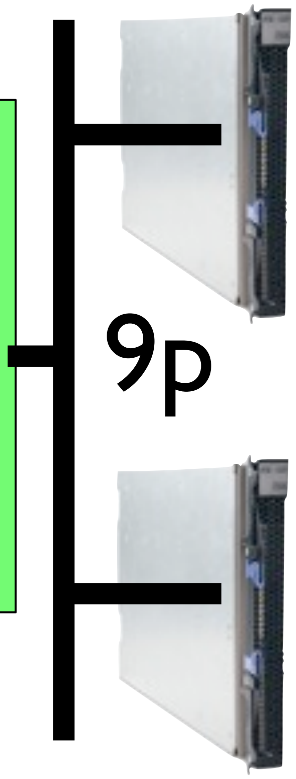
PowerPC Blades: Libra Workers

Libra

X86 Linux Front Ends

Pool of Libra Partitions





```
$ java -cp my.jar
```

```
$ for ((i=0;i<44;i++))
do
    java -cp my.jar &
done
```

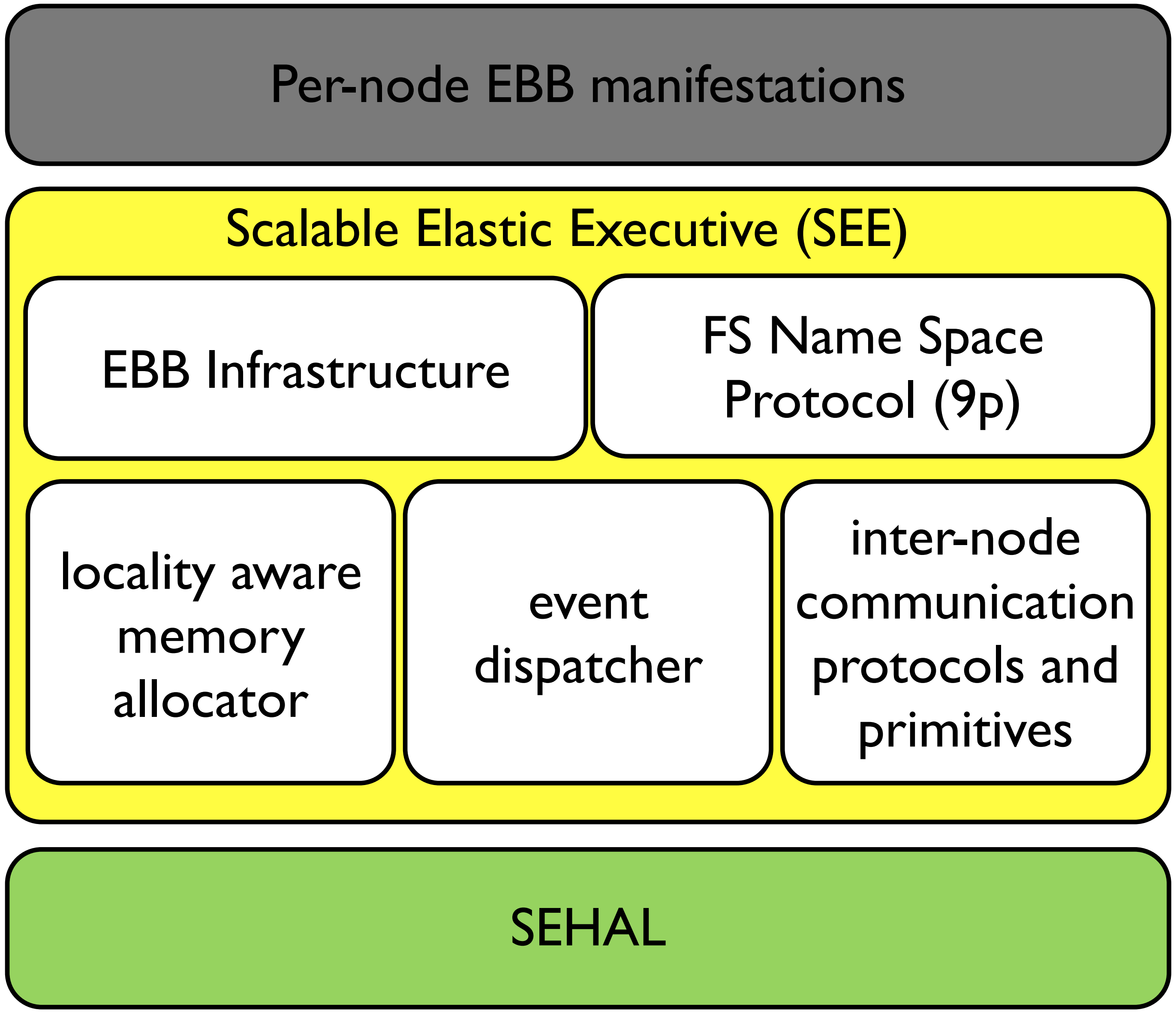
PowerPC Blades: Libra Workers

What did we learn?

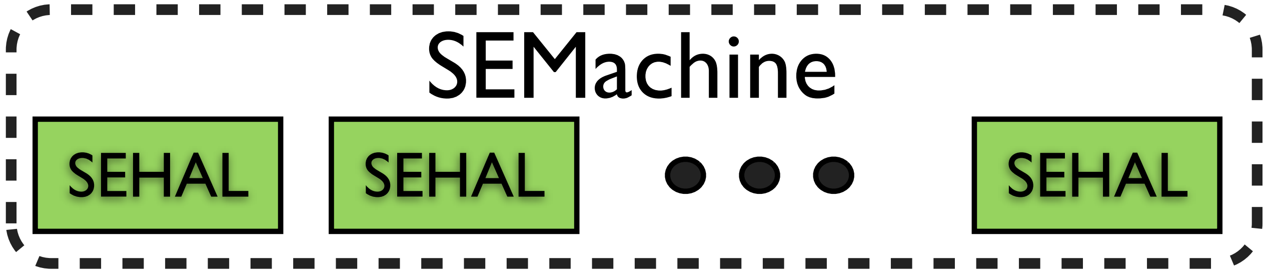
- Specialized environment for each application
- Lightweight system layer implementing services for performance
- General purpose OS for non-performance critical services

SEE :A LibOS for SESA

- Distributed LibOS that can elastically span nodes
- Instances cooperate to support the allocation and deallocation of EBB's
- Enables compatibility with Front End nodes running via unified 9p namespace

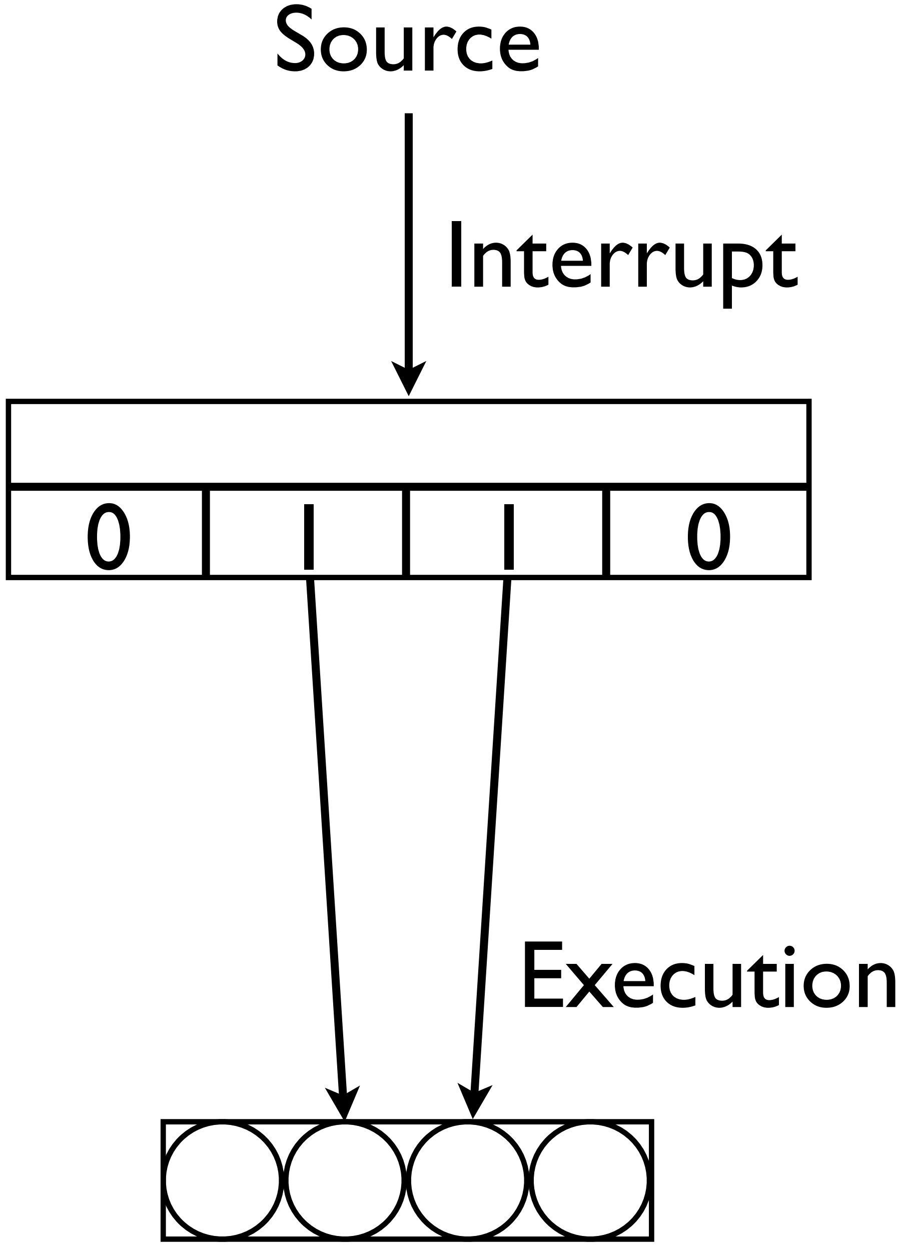

Per-node EBB manifestations
Scalable Elastic Executive (SEE)
EBB Infrastructure
FS Name Space Protocol (9p)
locality aware memory allocator
event dispatcher
inter-node communication protocols and primitives
SEHAL

SEMachines and EPICs

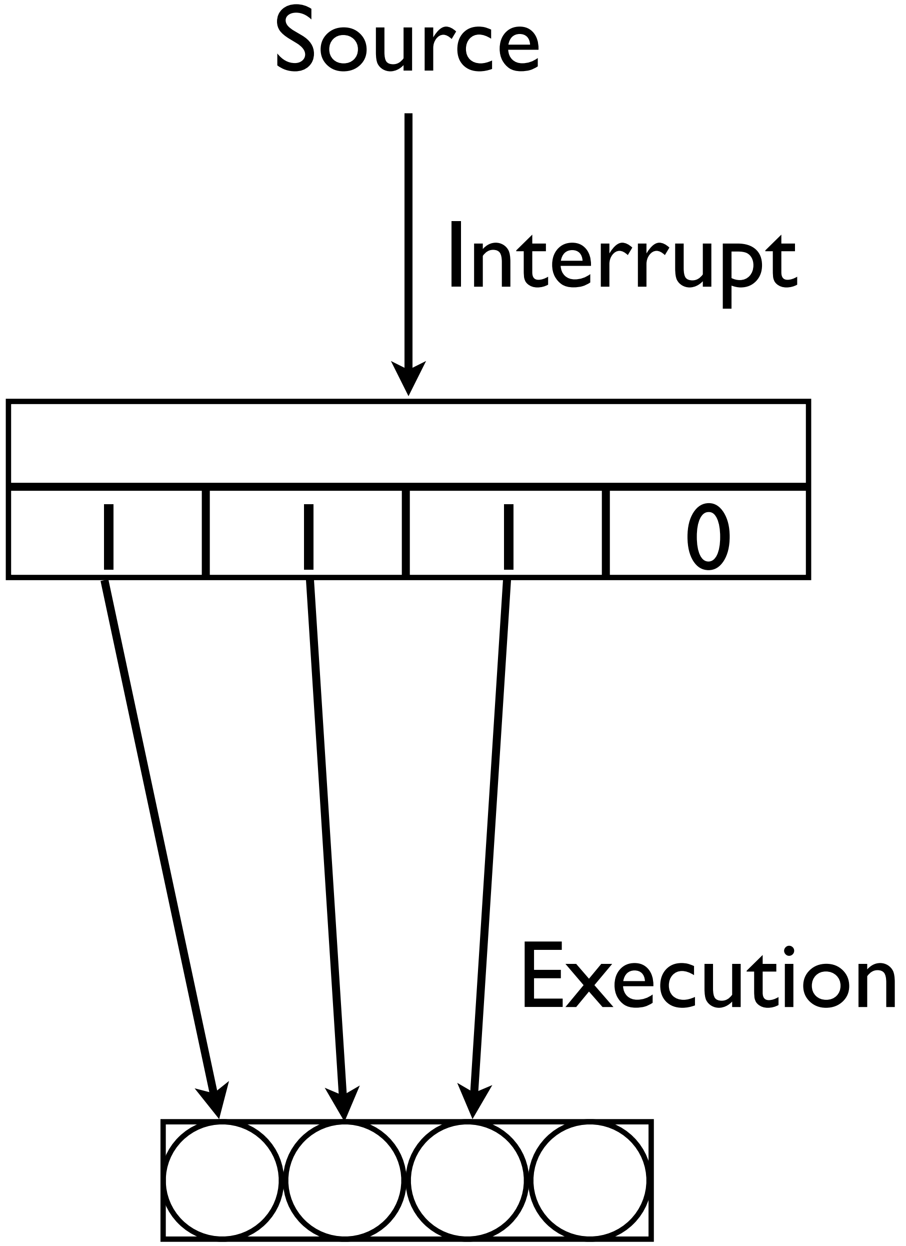


Hardware Abstraction Layer :
EPIC

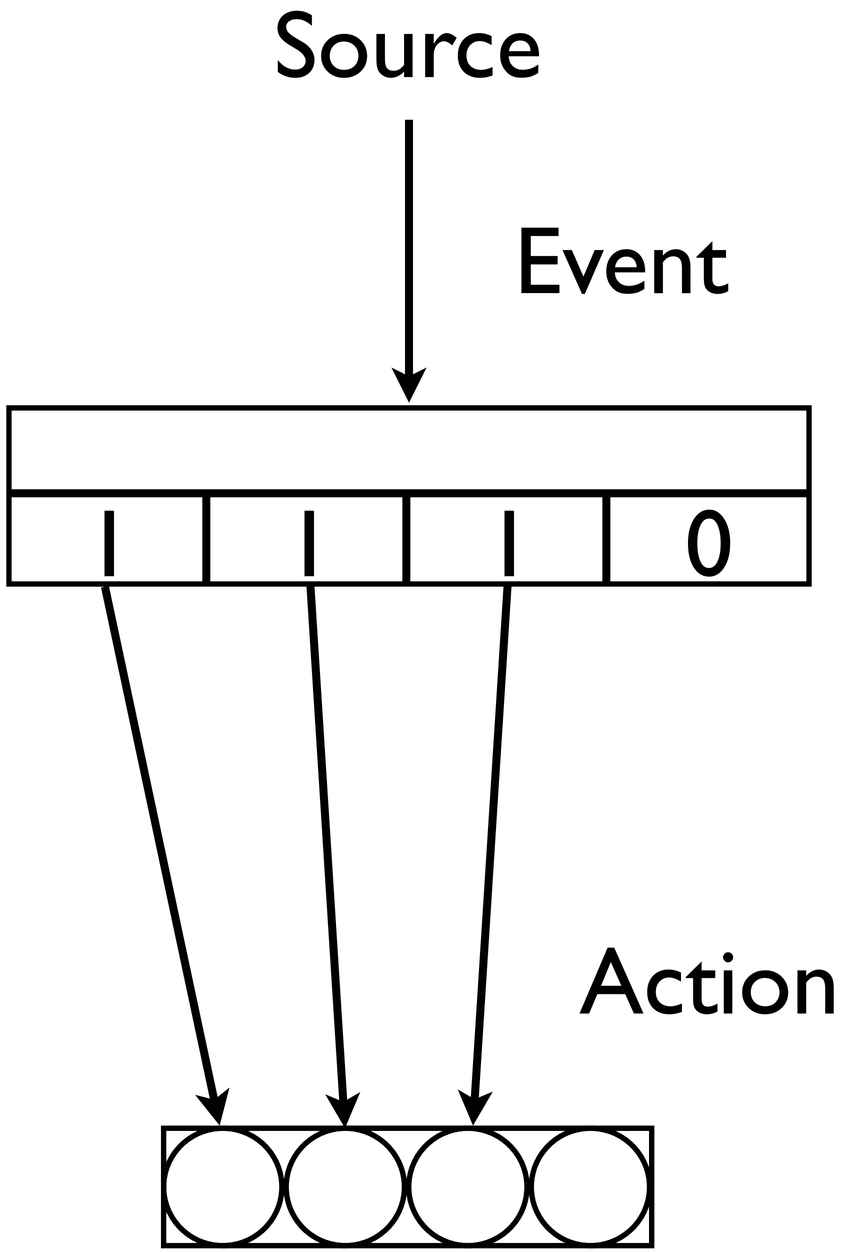
Programmable Interrupt Controller



Programmable Interrupt Controller



Programmable Interrupt Controller



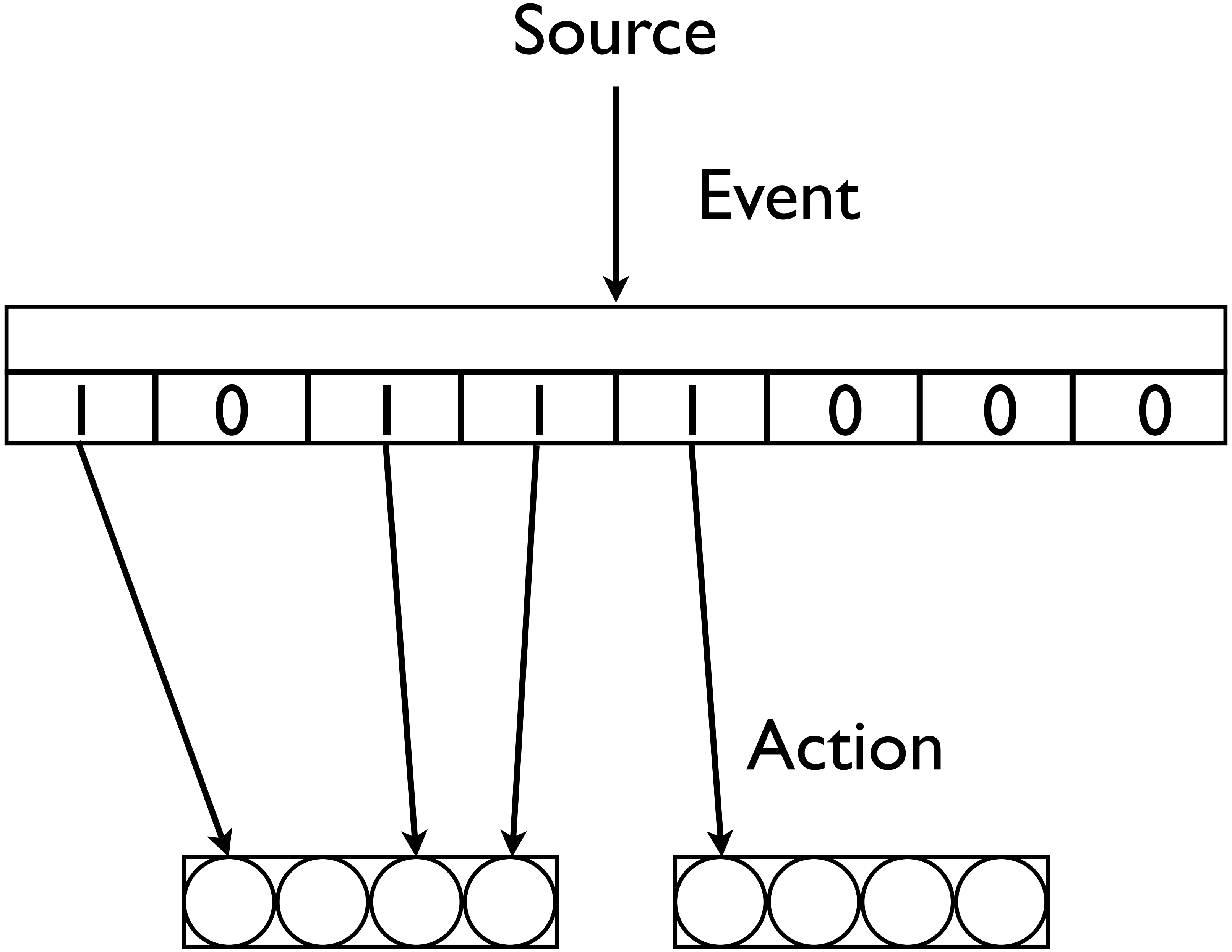
Source

Event

1	1	1	0

Action

Elastic Programmable Interrupt Controller



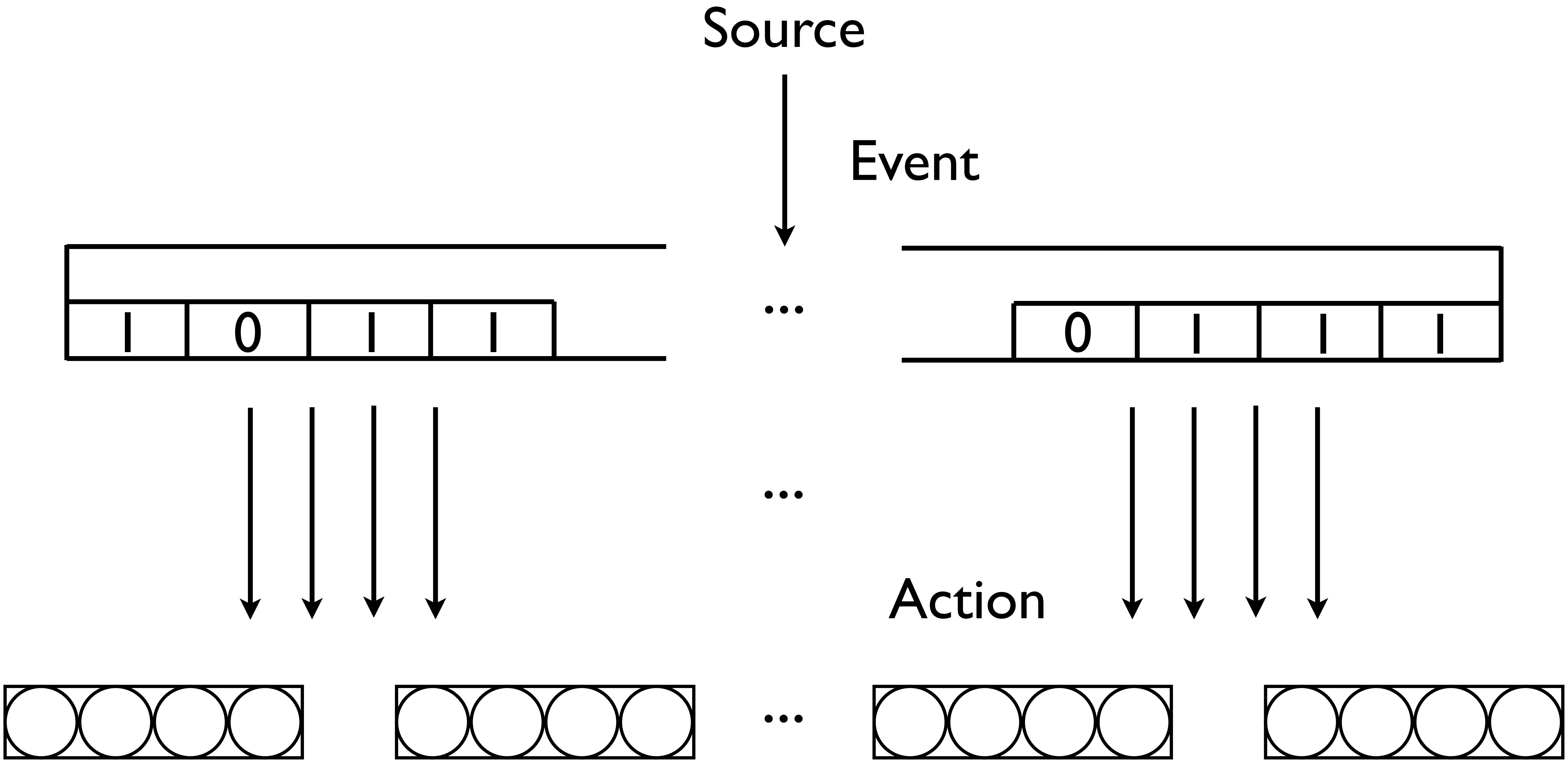
Source

Event

1	0	1	1	1	0	0	0

Action

Elastic Programmable Interrupt Controller



Elastic Programmable Interrupt Controller

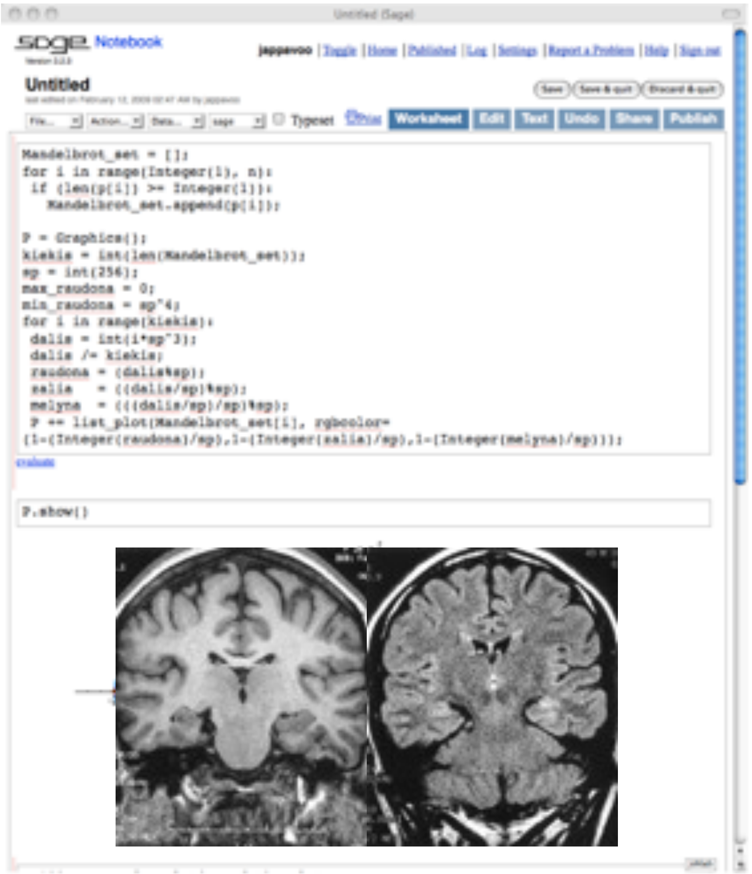
- Programmed by the SEE
- Provides the minimum requirement of elastic applications - mapping load to resources
- Portable layer
- Take advantage of network features such as broadcast and multicast

OUTLINE

1. THE PROBLEM
2. OBSERVATIONS
3. OUR TAKE ON A SOLUTION
4. PROTOTYPE & CHALLENGES

PROTOTYPE APP

Sage*

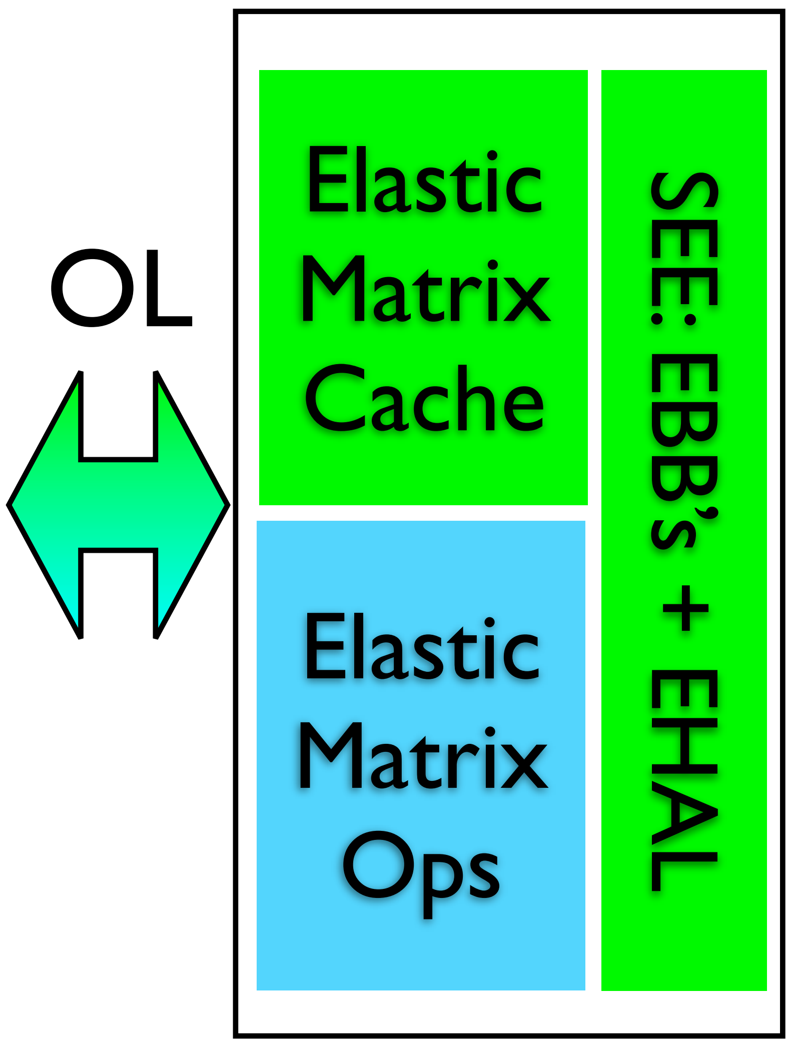


```
Mandelbrot_set = [];
for i in range(Integer(1), n):
 if (len(p[i]) >= Integer(1)):
   Mandelbrot_set.append(p[i]);

P = Graphics();
kiekis = int(len(Mandelbrot_set));
sp = int(256);
max_raudona = 0;
min_raudona = sp^4;
for i in range(kiekis):
 dalis = int(i*sp^3);
 dalis /= kiekis;
 raudona = (dalis%sp);
 zalia   = ((dalis/sp)%sp);
 melyna  = (((dalis/sp)/sp)%sp);
 P += list_plot(Mandelbrot_set[i], rgbcolor=
(1-(Integer(raudona)/sp),1-(Integer(zalia)/sp),1-(Integer(melyna)/sp)));
```

```
P.show()
```

SESA SAGE SERVICES



vmware

Traditional HW

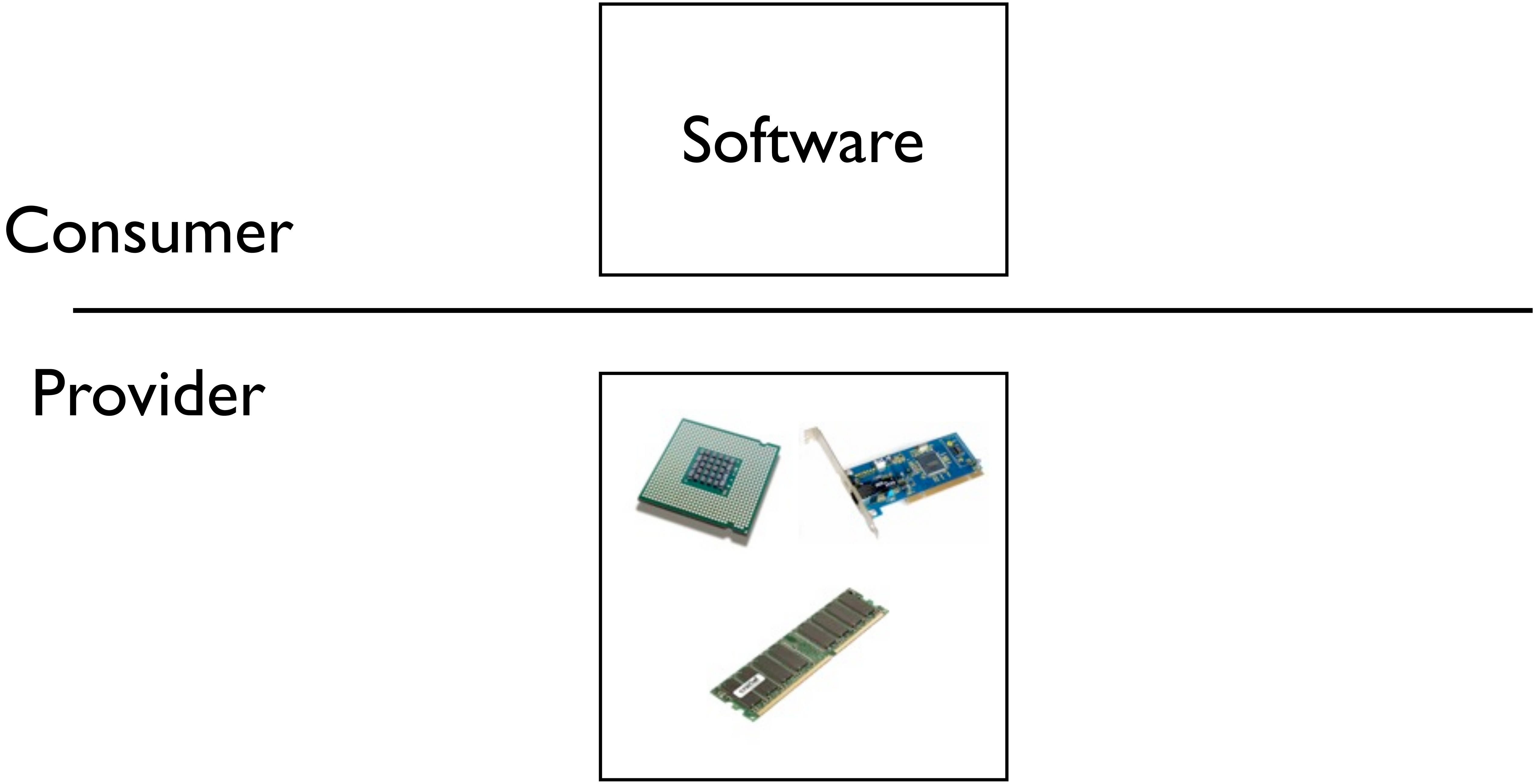
Advanced HW

Challenges and Discussion

OUTLINE

1. THE PROBLEM
    1. Pay as you go computing
    2. Insufficient systems support for elasticity
2. OBSERVATIONS
3. OUR TAKE ON A SOLUTION
4. PROTOTYPE & CHALLENGES

Pay as you go hardware

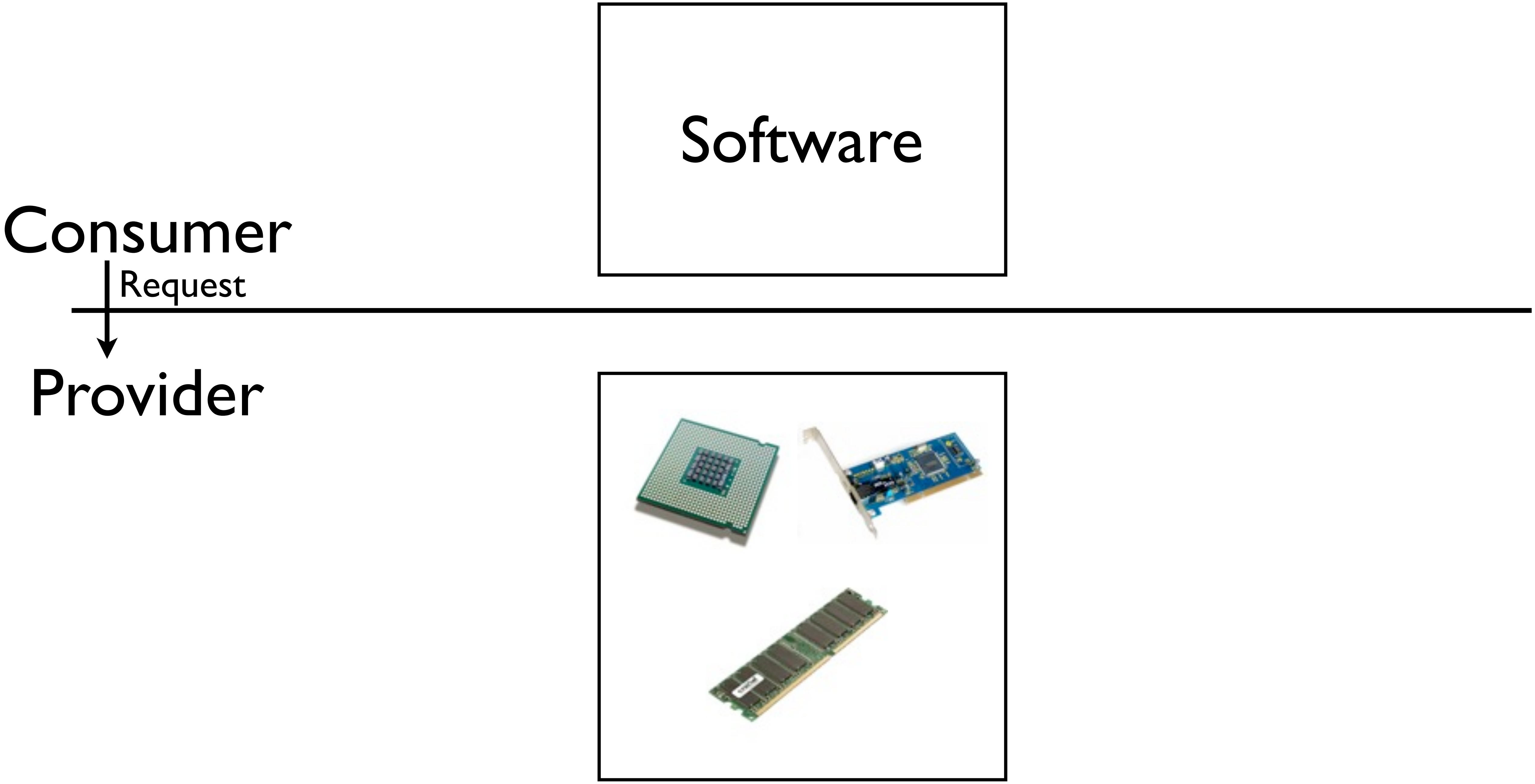


Software

Consumer

Provider

Pay as you go hardware



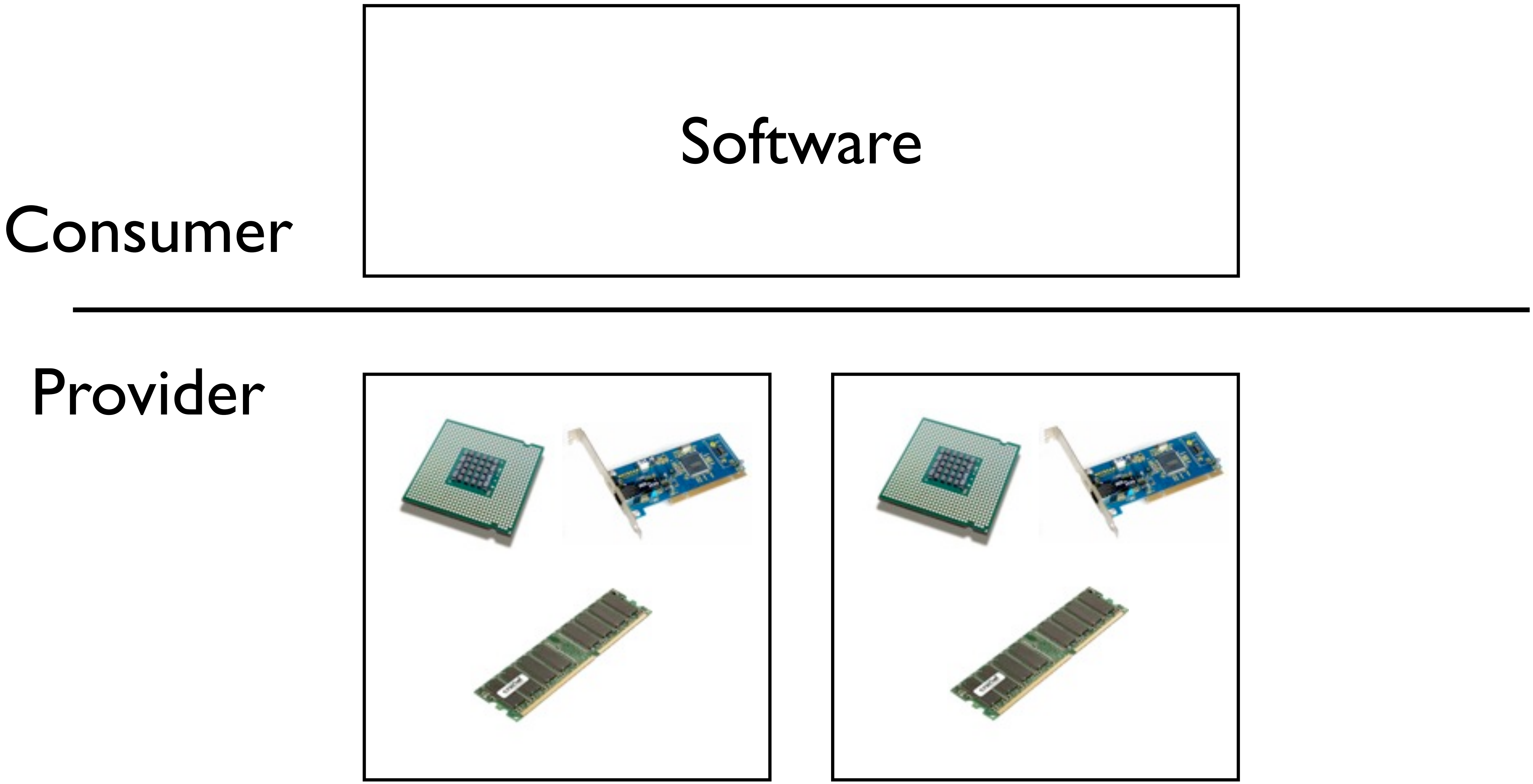
Software

Consumer

Request

Provider

Pay as you go hardware

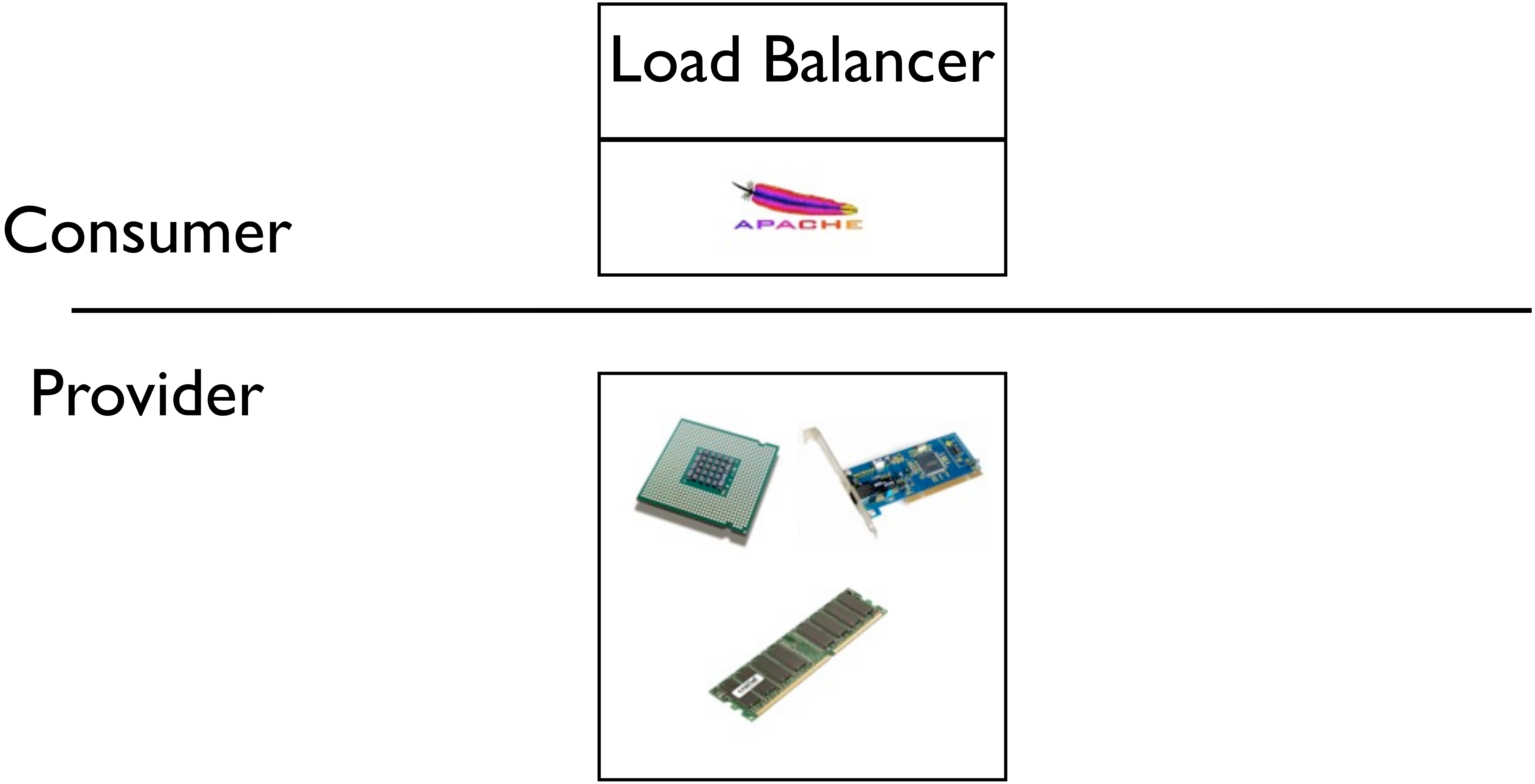


Software

Consumer

Provider

Elastic Website

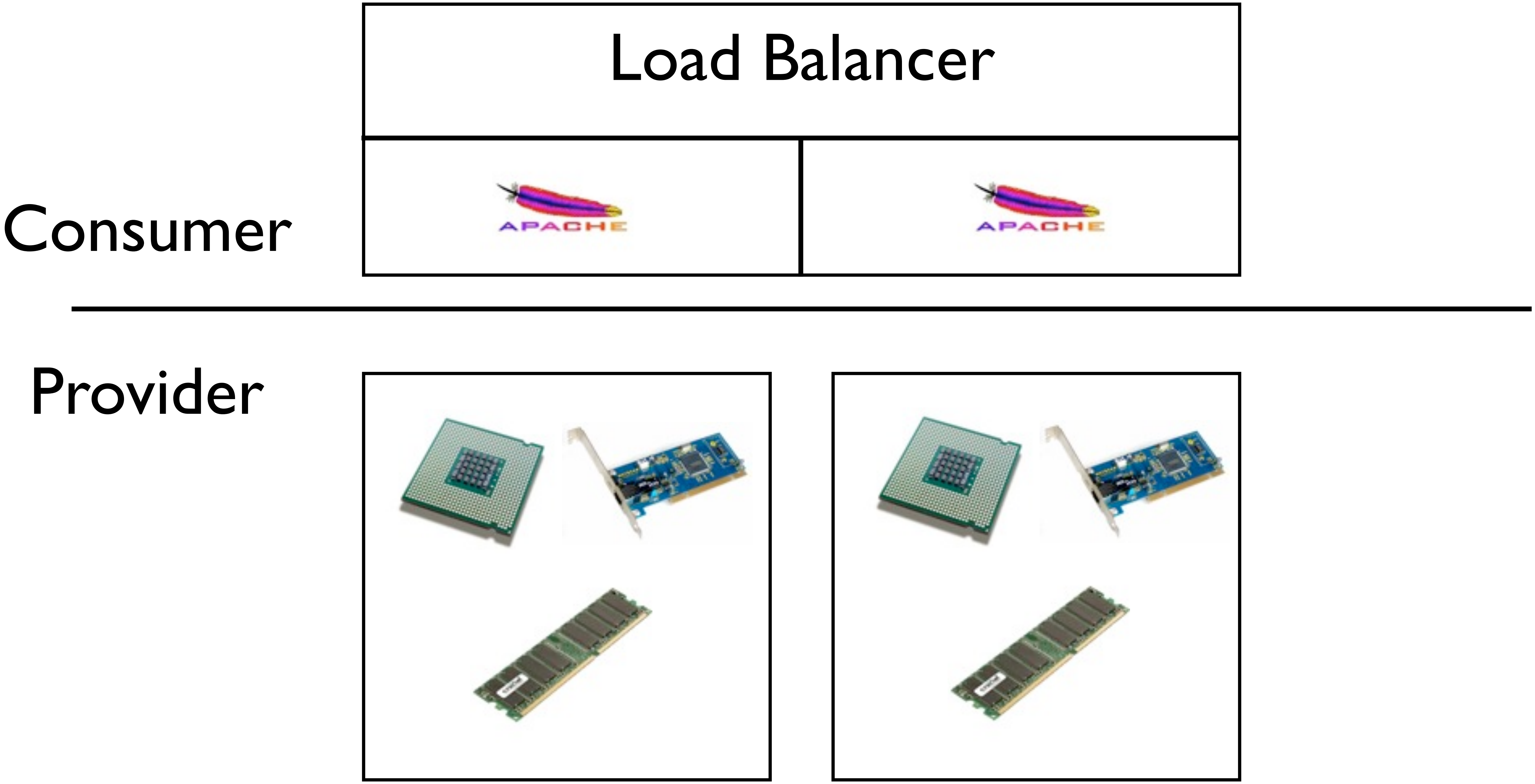


Load Balancer

Consumer

Provider

Elastic Website

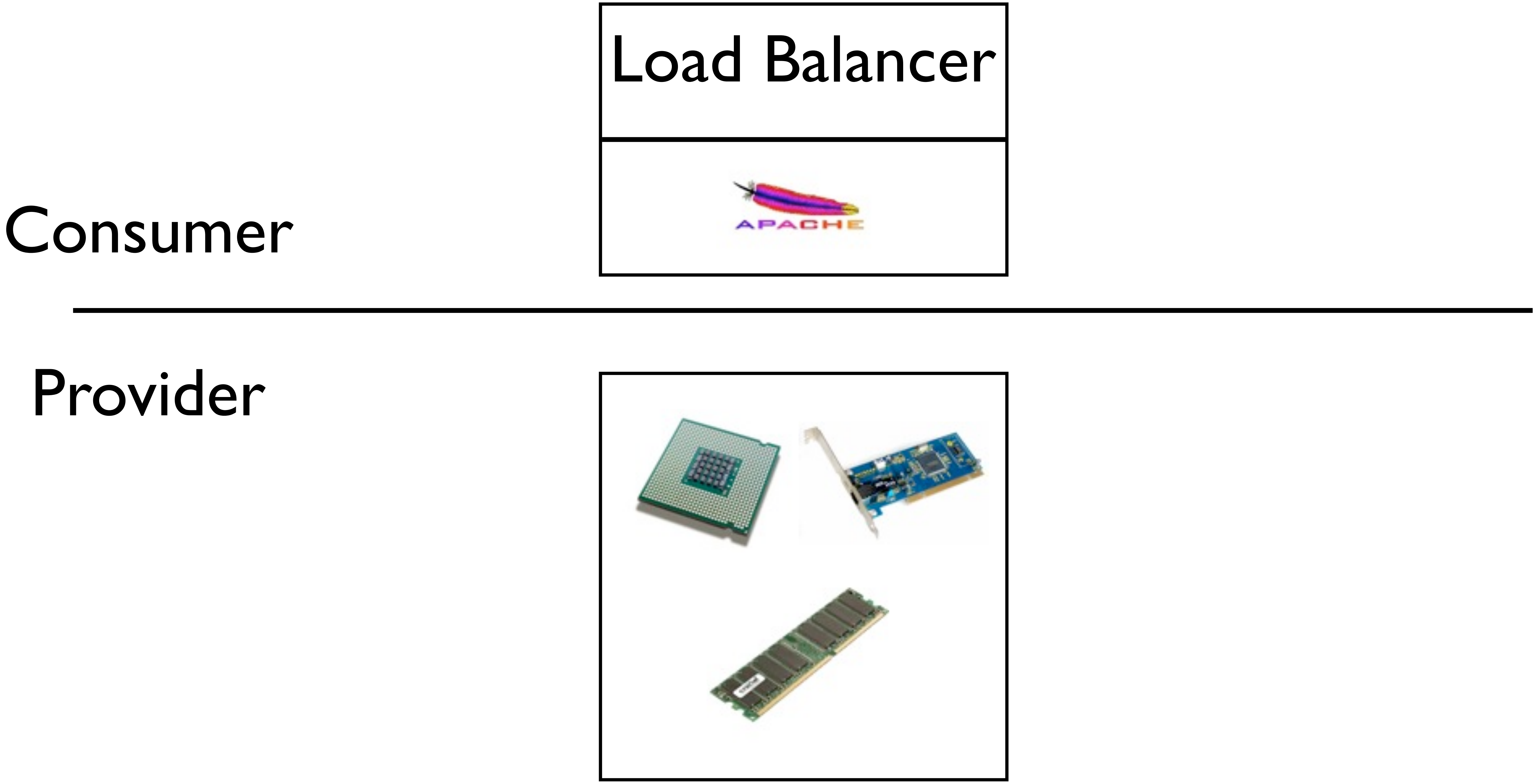


Load Balancer

Consumer

Provider

Elastic Website



Load Balancer

Consumer

Provider

Other Elastic Applications

- Analytics
- Batch computation
- Stream processing

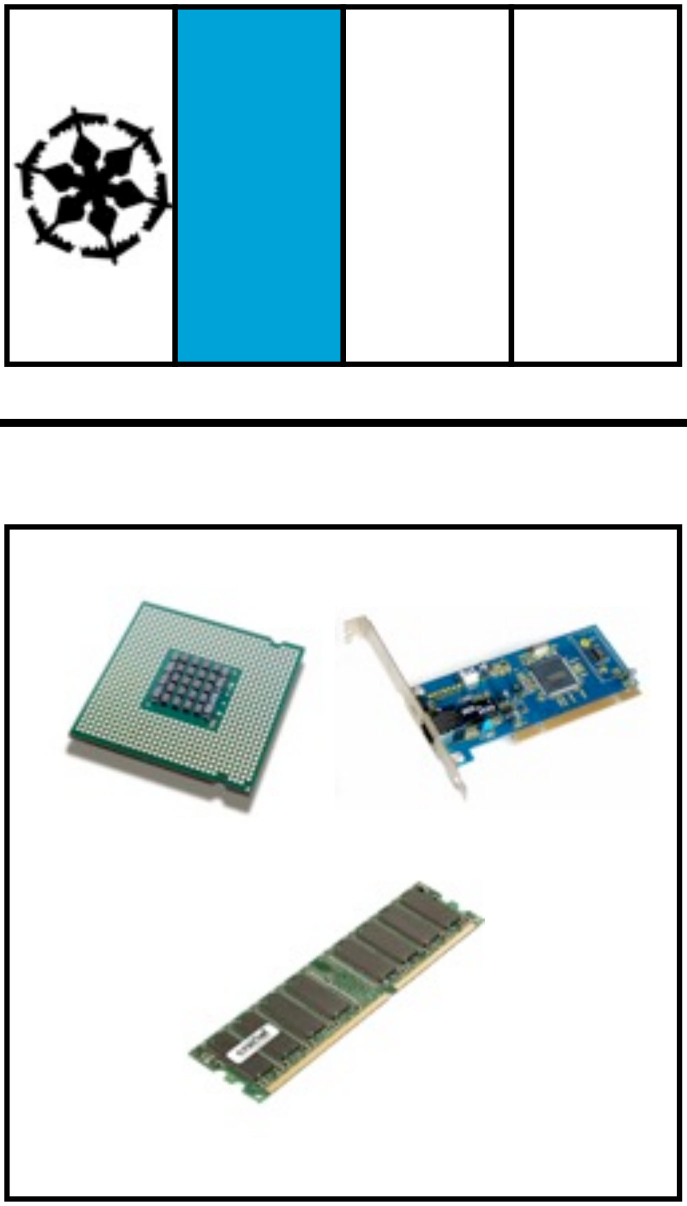
What’s the problem?

- Allocation/Boot-time
- Programmability

Medical Imaging Application

- Megapixel image
- Quadratic algorithm
- (1 mil pixels * 4 bytes/pixel)^2 ~ 14 TB
- On Amazon EC2 ~ $8000 per day

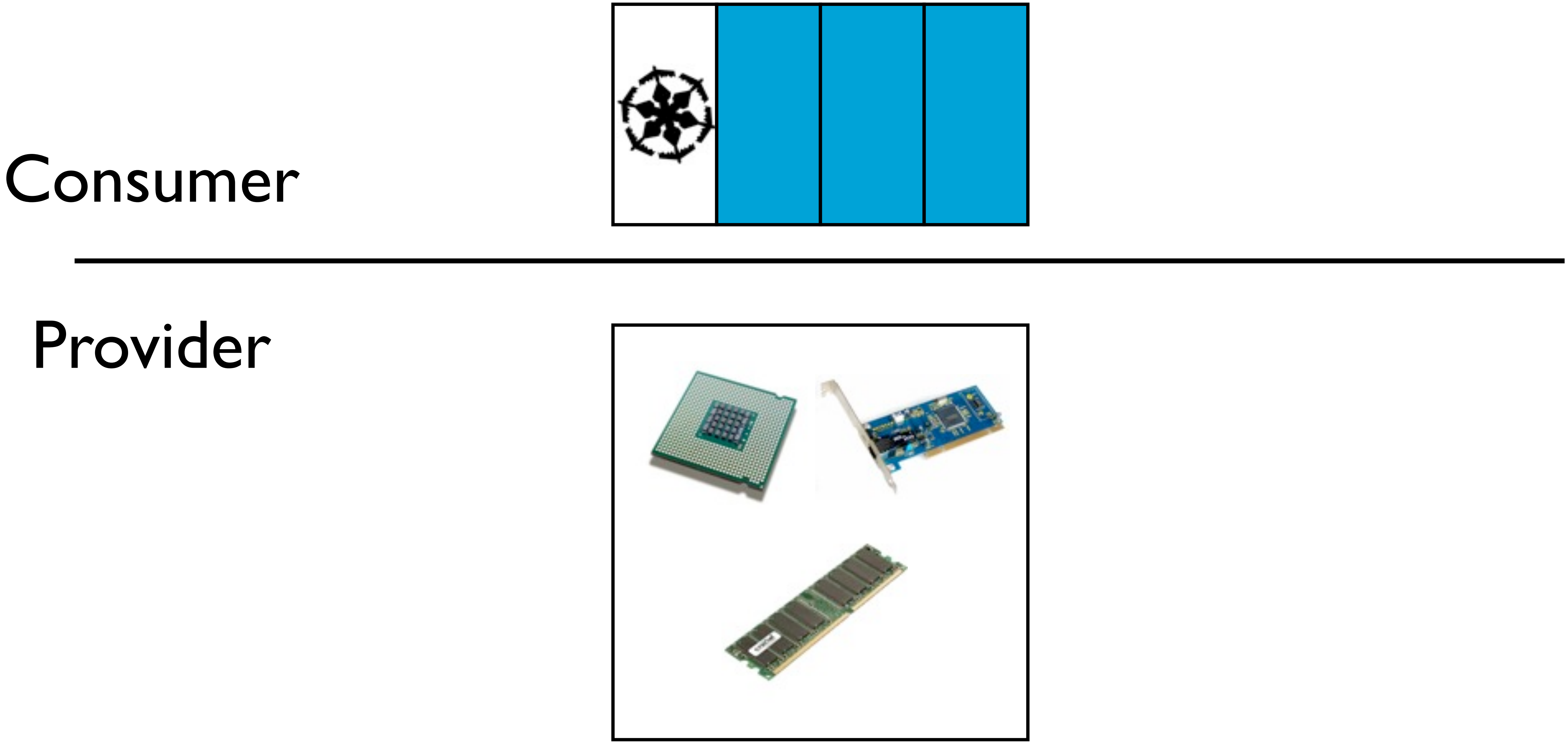
Snowflock



Consumer

Provider

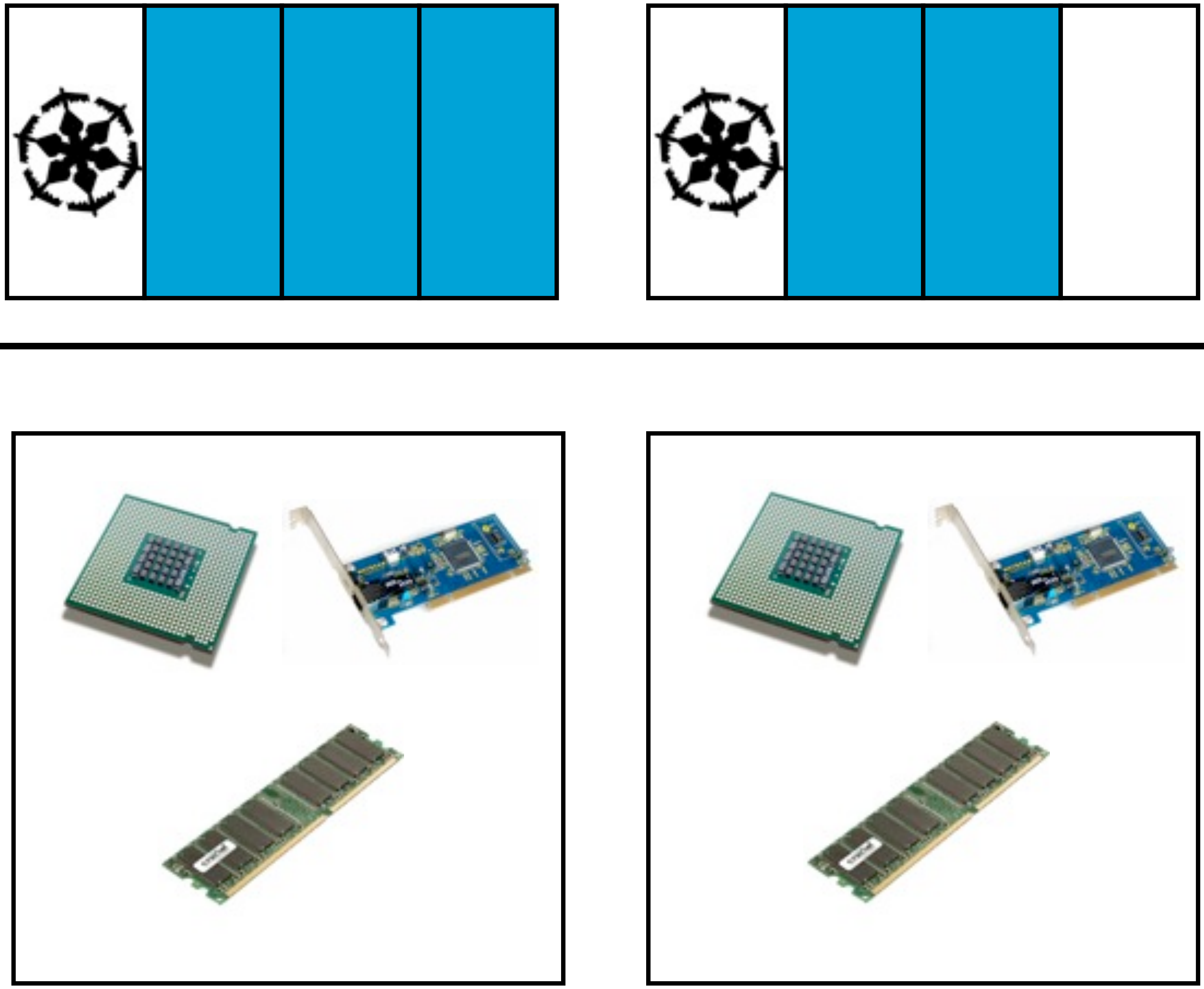
Snowflock



Consumer

Provider

Snowflock

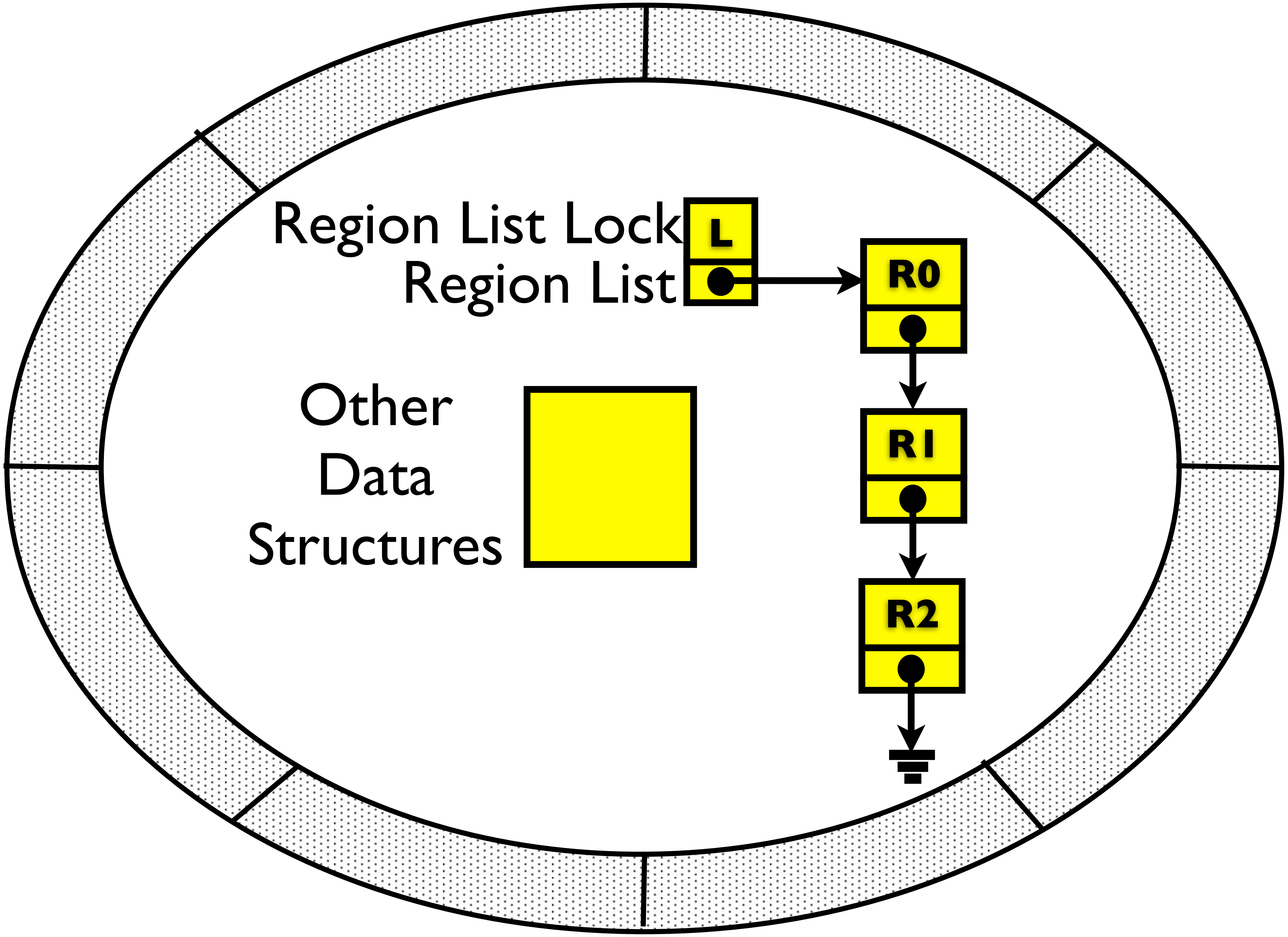


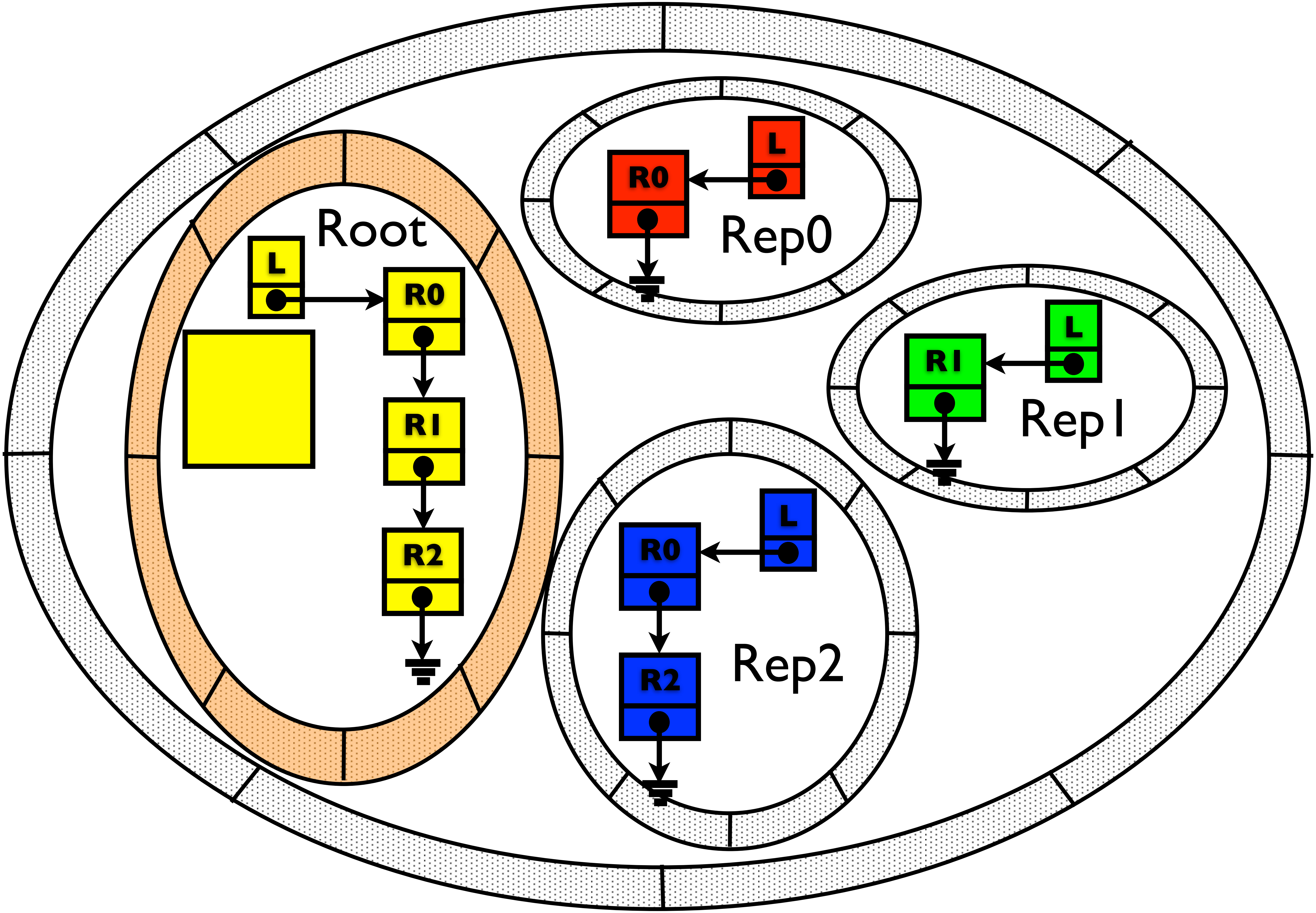
Consumer

Provider

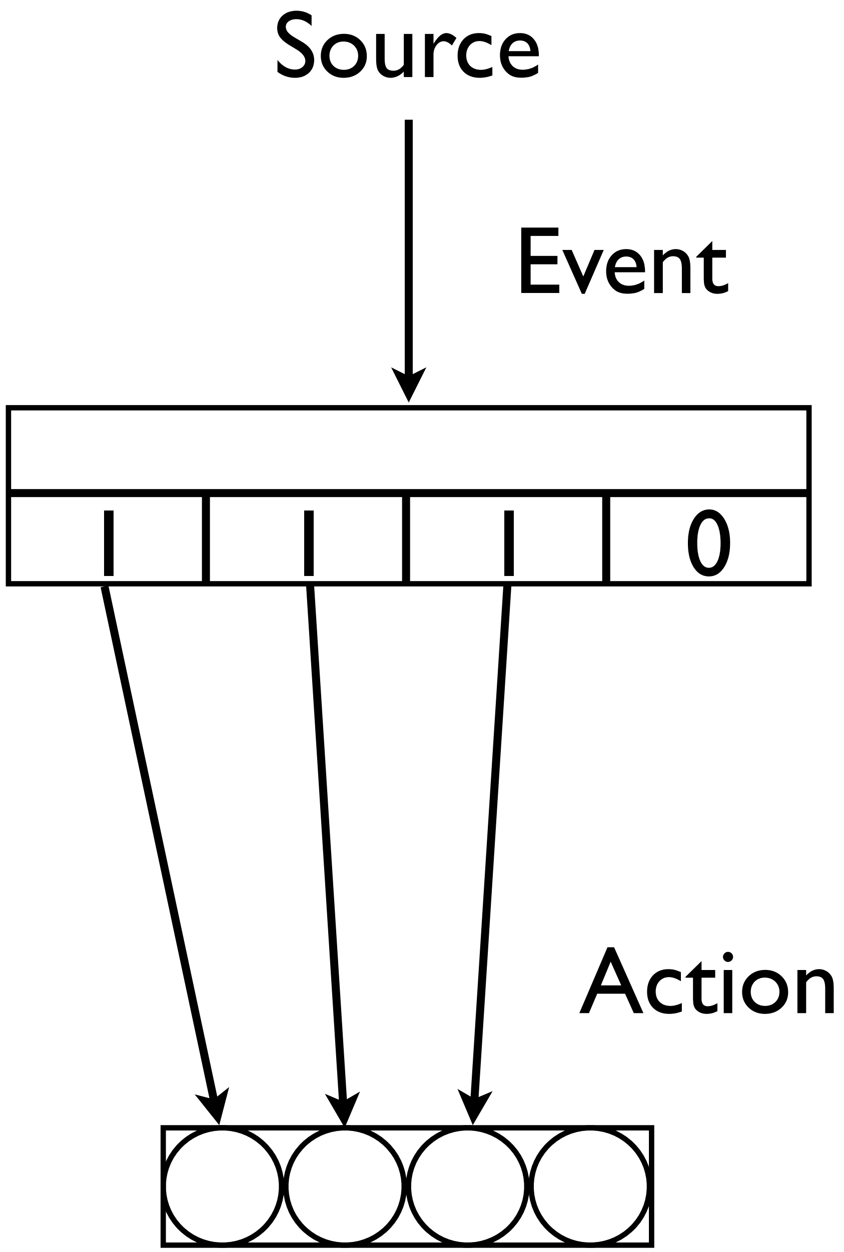
Distributing an Object

Non-Distributed Object Instance




Root
L
R0
R1
R2
Rep0
L
R0
Rep1
L
R1
Rep2
L
R0
R2

Elastic Programmable Interrupt Controller


Source
Event
1
1
1
0
Action