

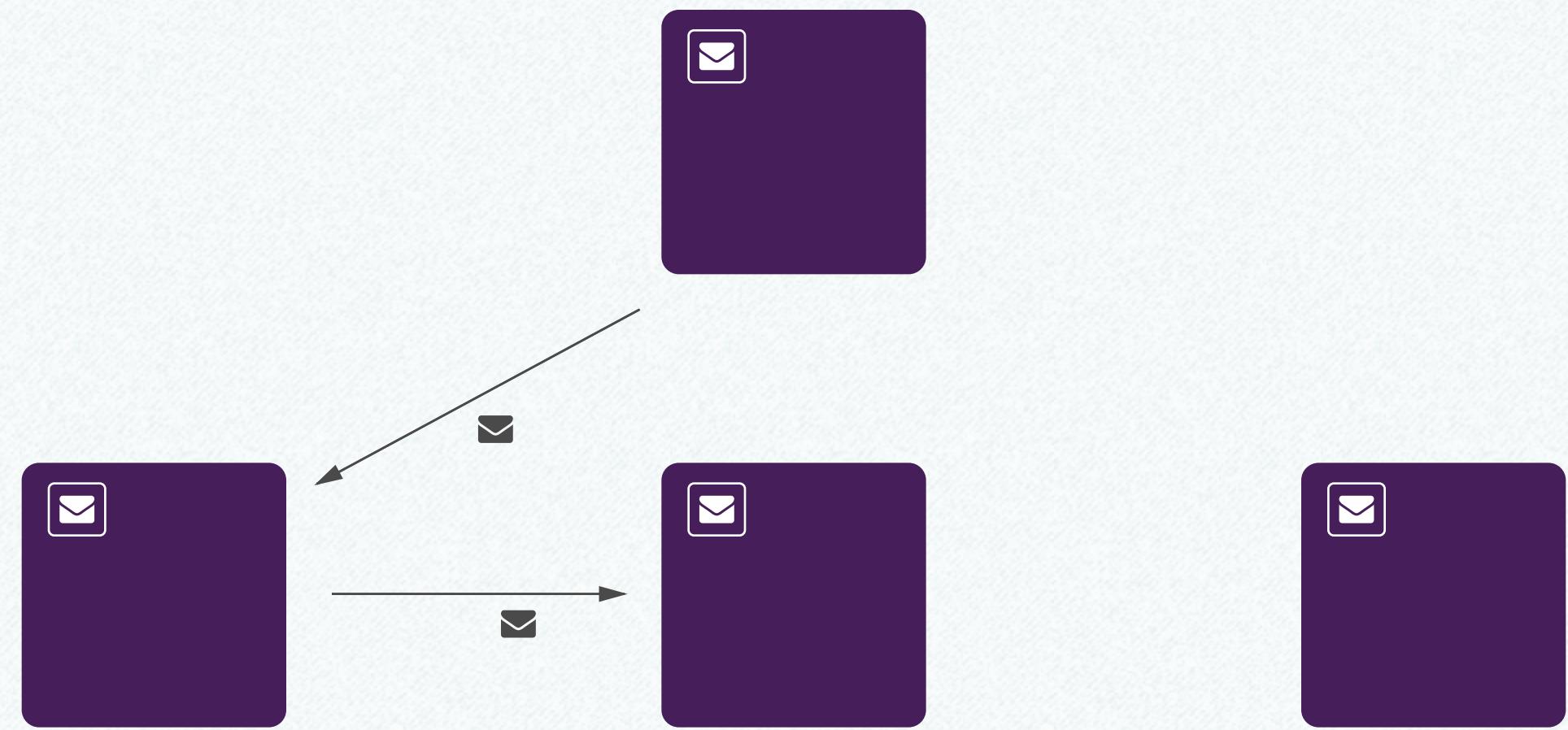


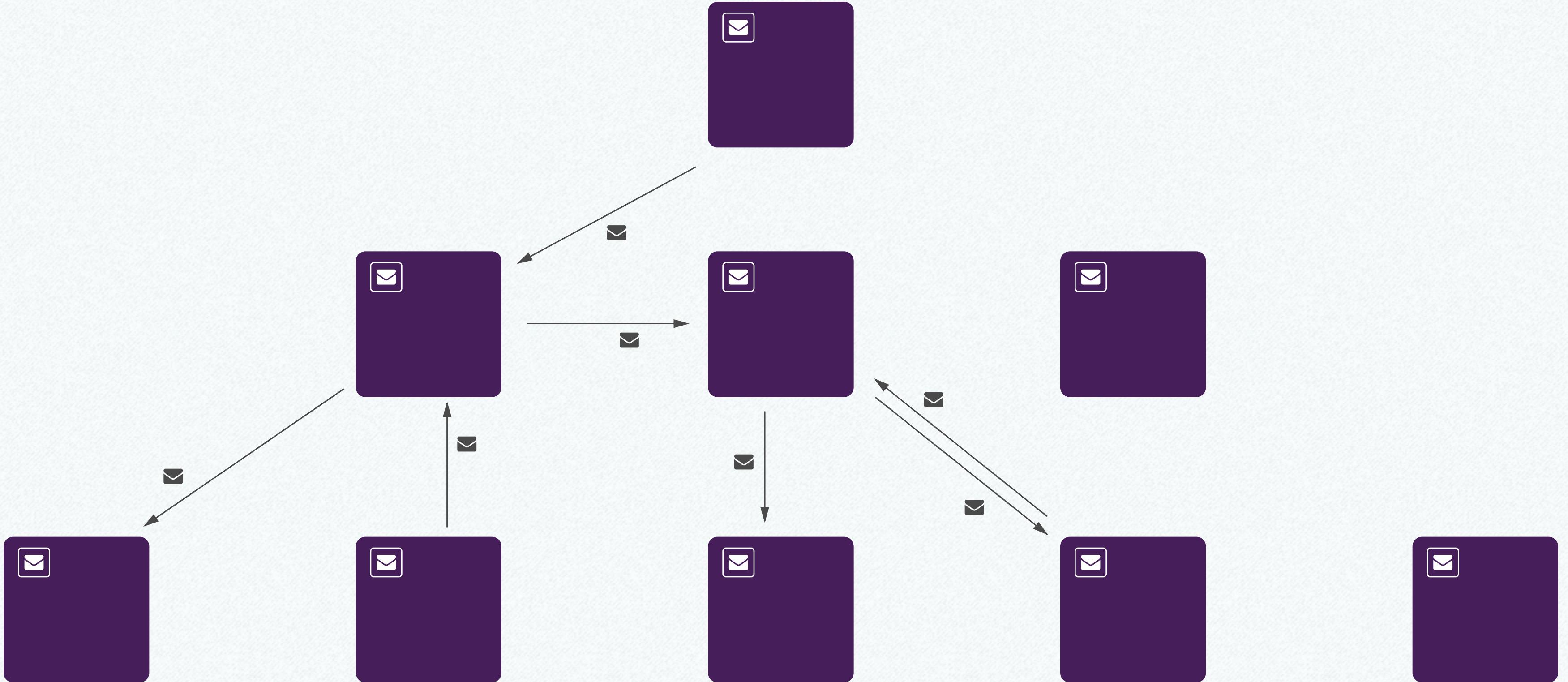
elixir

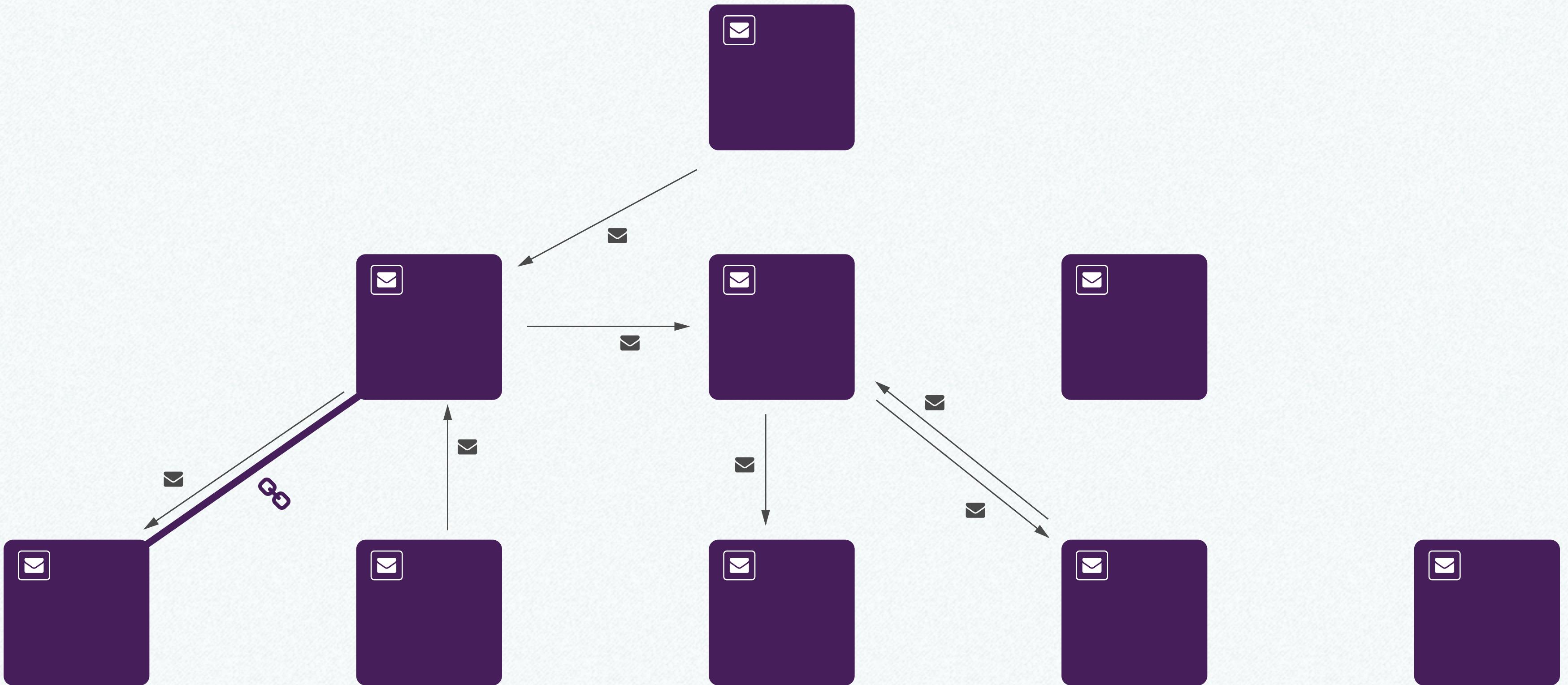
“ Applications will increasingly need to be concurrent if they want to fully exploit continuing exponential CPU throughput gains.

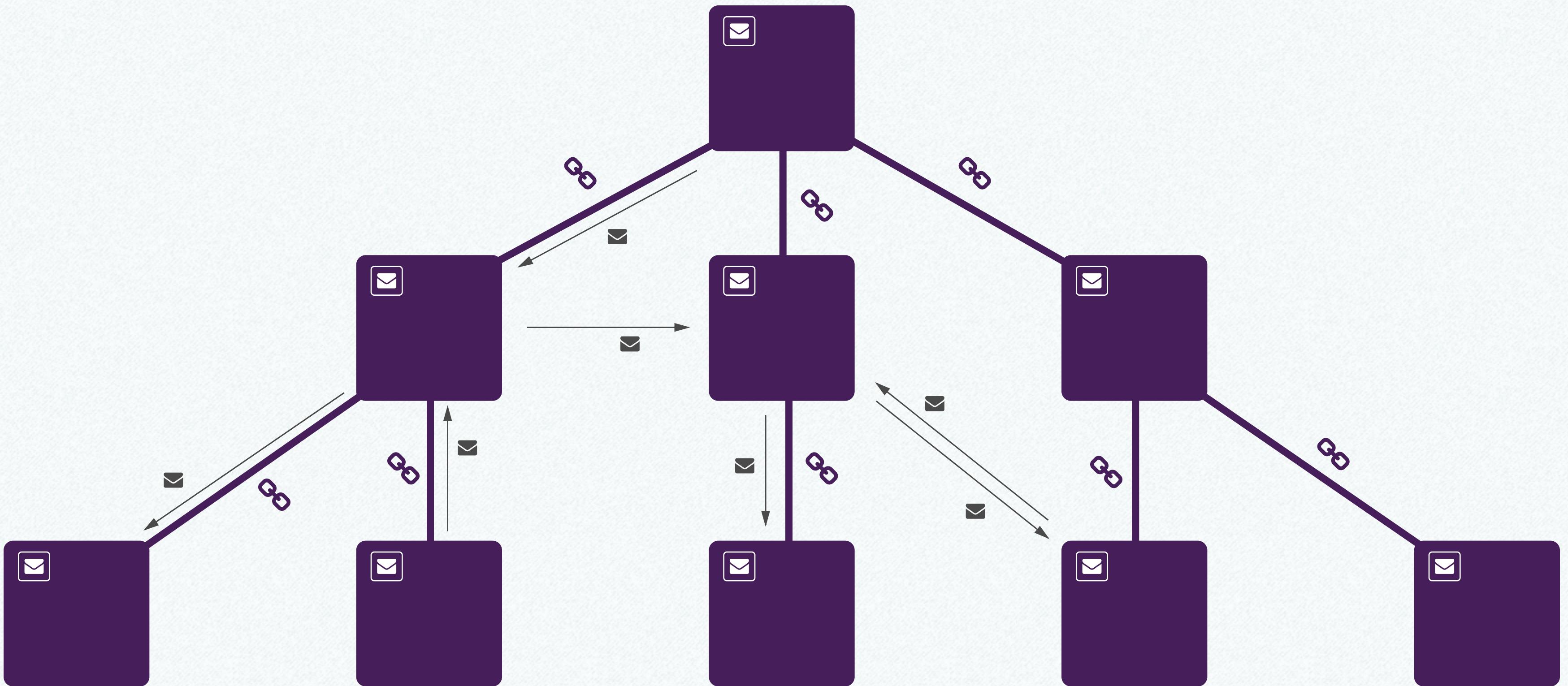
— Herb Sutter,
*The Free Lunch is Over: A Fundamental Turn
Toward Concurrency in Software*











RUBY

```
def sum(*numbers)
  acc = 0;
  numbers.each do |n|
    acc += n
  end
  acc
end
```

RUBY

```
def sum(*numbers)
  acc = 0;
  numbers.each do |n|
    acc += n
  end
  acc
end
```

ELIXIR

```
def sum(numbers) do
  if numbers == [] do
    0
  else
    head = hd(numbers)
    tail = tl(numbers)
    head + sum(tail)
  end
end
```

RUBY

```
def sum(*numbers)
  acc = 0;
  numbers.each do |n|
    acc += n
  end
  acc
end
```

ELIXIR

```
def sum(numbers) do
  if numbers == [] do
    0
  else
    head = hd(numbers)
    tail = tl(numbers)
    head + sum(tail)
  end
end
```

```
sum([1,2,3,4,5])
1 + sum([2,3,4,5])
1 + 2 + sum([3,4,5])
1 + 2 + 3 + sum([4,5])
1 + 2 + 3 + 4 + sum([5])
1 + 2 + 3 + 4 + 5 + sum([])
```

Demo

ELIXIR

```
def sum(numbers) do
  if numbers == [] do
    0
  else
    head = hd(numbers)
    tail = tl(numbers)
    head + sum(tail)
  end
end
```

ELIXIR

```
def sum(numbers) do
  if numbers == [] do
    0
  else
    head = hd(numbers)
    tail = tl(numbers)
    head + sum(tail)
  end
end
```

ALSO ELIXIR

```
def sum([head | tail]) do
  head + sum(tail)
end

def sum([]), do: 0
```

RUBY

```
def fun(x, y)
  if x == 5
    if y == 3
      7
    else
      11
    end
  else
    y
  end
end
```

RUBY

```
def fun(x, y)
  if x == 5
    if y == 3
      7
    else
      11
    end
  else
    y
  end
end
```

ELIXIR

```
def fun(5, 3), do: 7
def fun(5, _), do: 11
def fun(_, y), do: y
```

```
brew install elixir
```

github.com/jswinarton/elixir-talk-may-2016