

# Event-Based Testing

Sebastian Bergmann





- **Sebastian** since 1978
- **Computers** since 1990
- **PHP** since 1998
- **Open Source** since 1998
- **PHPUnit** since 2000
- **thePHP.cc** since 2009



Jetzt kostenlos und unverbindlich testen  
mit Vertragsnummer 2023-0515-0421

# Fachkräftemangel? Fortbildungsflatrate!

The PHP Consulting Company  
<https://thePHP.cc>  
+49.8171.428058  
team@thePHP.cc



<https://flatrate.thephp.cc>

sharing experience



„An object-oriented system is a web of collaborating objects. A system is built by creating objects and plugging them together so that they can send messages to one another. The behavior of the system is an emergent property of the composition of the objects – the choice of objects and how they are connected.”

Steve Freeman and Nat Pryce

„The big idea is "messaging" [...] The key in making great and growable systems is much more to design how its modules communicate rather than what their internal properties and behaviors should be.”

Alan Kay

„[C]omponents raise events when things change. Other components then listen to events and react appropriately. Event Collaboration leads to some very different ways of thinking about how parts need to think about their interaction with other parts.”

Martin Fowler

# Event

- **Domain Information**
  - What happened?
  - When did it happen?
- **Ordered**
  - One event follows another
- **Immutable**
  - What has happened cannot be changed

# Event Sourcing

System state is kept as sequence of immutable events.

<https://thephp.cc/topics/event-sourcing>



# Event Sourcing

Events are the source of truth.

<https://thephp.cc/topics/event-sourcing>

SHALL WE PLAY A GAME?

CHESS

POKER

FIGHTER COMBAT

GUERRILLA ENGAGEMENT

DESERT WARFARE

AIR-TO-GROUND ACTIONS

THEATERWIDE TACTICAL WARFARE

THEATERWIDE BIOTOXIC AND CHEMICAL WARFARE

GLOBAL THERMONUCLEAR WARFARE

> TIC-TAC-TOE

|   | A | B | C |
|---|---|---|---|
| 1 |   |   |   |
| 2 |   |   |   |
| 3 |   |   |   |

1. B2

|   | A | B | C |
|---|---|---|---|
| 1 |   |   |   |
| 2 |   | X |   |
| 3 |   |   |   |

1. B2

2. C1

|   | A | B | C |
|---|---|---|---|
| 1 |   |   | 0 |
| 2 |   | X |   |
| 3 |   |   |   |

1. B2
2. C1
3. B1

|   | A | B | C |
|---|---|---|---|
| 1 |   | X | O |
| 2 |   | X |   |
| 3 |   |   |   |

1. B2
2. C1
3. B1
4. B3

|   | A | B | C |
|---|---|---|---|
| 1 |   | X | O |
| 2 |   | X |   |
| 3 |   | O |   |

1. B2
2. C1
3. B1
4. B3
5. A2

|   | A | B | C |
|---|---|---|---|
| 1 |   | X | O |
| 2 | X | X |   |
| 3 |   | O |   |



1. B2
2. C1
3. B1
4. B3
5. A2
6. C2

|   | A | B | C |
|---|---|---|---|
| 1 |   | X | O |
| 2 | X | X | O |
| 3 |   | O |   |

1. B2
2. C1
3. B1
4. B3
5. A2
6. C2
7. C3

|   | A | B | C |
|---|---|---|---|
| 1 |   | X | O |
| 2 | X | X | O |
| 3 |   | O | X |

1. B2
2. C1
3. B1
4. B3
5. A2
6. C2
7. C3
8. A1

|   | A | B | C |
|---|---|---|---|
| 1 | O | X | O |
| 2 | X | X | O |
| 3 |   | O | X |

1. B2
2. C1
3. B1
4. B3
5. A2
6. C2
7. C3
8. A1
9. A3

|   | A | B | C |
|---|---|---|---|
| 1 | O | X | O |
| 2 | X | X | O |
| 3 | X | O | X |

1. B2
2. C1
3. B1
4. B3
5. A2
6. C2
7. C3
8. A1
9. A3

|   | A | B | C |
|---|---|---|---|
| 1 | O | X | O |
| 2 | X | X | O |
| 3 | X | O | X |

# Event Storming



<https://thephp.cc/topics/event-storming>



# CQRS

Command/Query Responsibility Segregation

<https://thephp.cc/topics/cqrs>



# CQRS

Use different models for commands and queries.

<https://thephp.cc/topics/cqrs>

# CQRS

Separate commands (write operations)  
from queries (read operations).

<https://thephp.cc/topics/cqrs>

SHALL WE PLAY A GAME?

ARK NOVA

BRASS BIRMINGHAM

GREAT WESTERN TRAIL

GLIGONG

TEKHENU

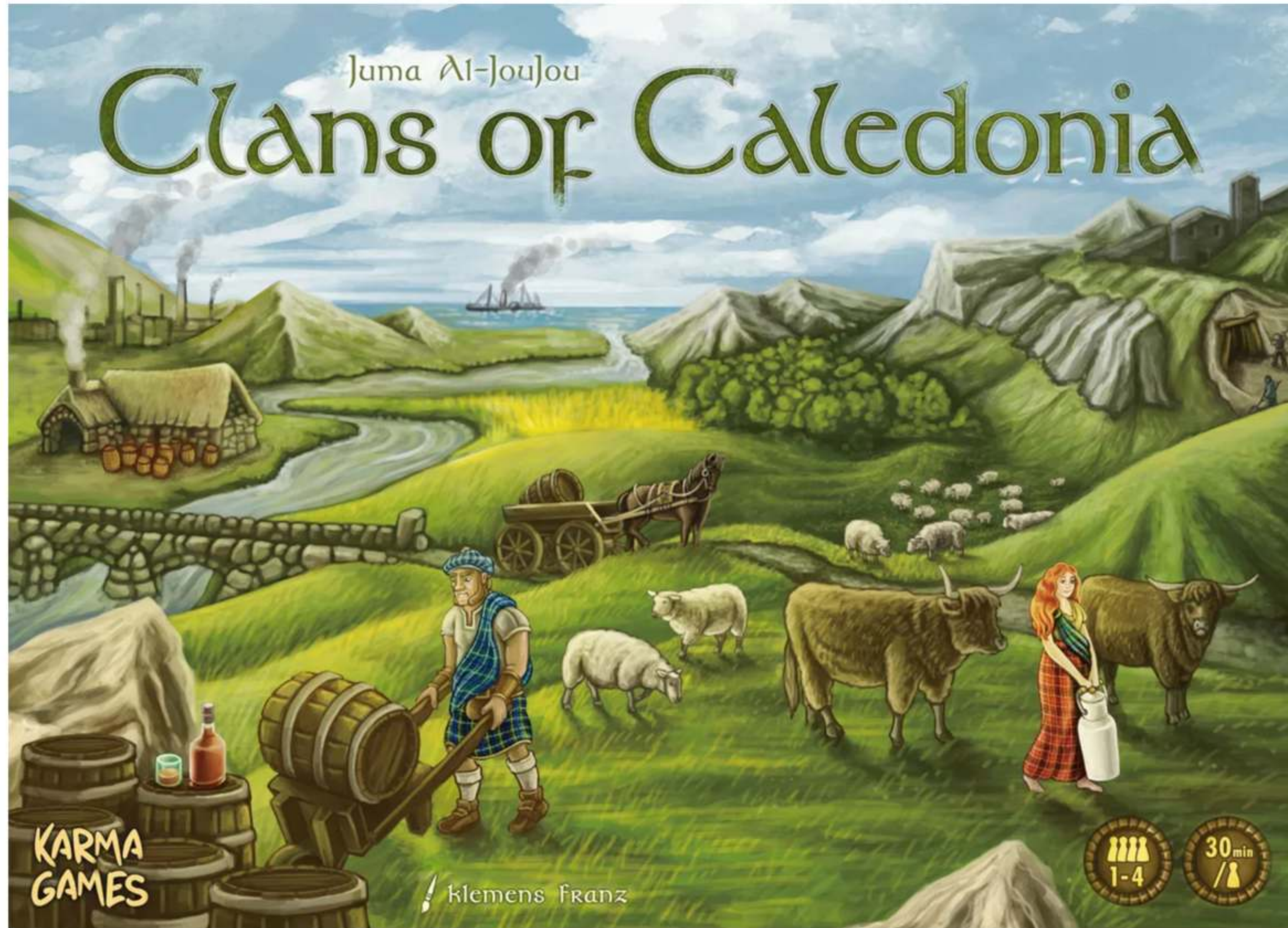
TEOTIHUACAN

TRICKERION

TZOLKIN

LINFAIR

> CLANS OF CALEDONIA



<https://boardgamegeek.com/boardgame/216132/clans-caledonia>

4 4 4 4

+

PURCHASE

|   |   |   |    |    |    |
|---|---|---|----|----|----|
| 8 | 8 | 8 | 15 | 15 | 16 |
| 7 | 8 | 8 | 14 | 14 | 15 |
| 6 | 7 | 7 | 13 | 14 | 14 |
| 6 | 7 | 7 | 12 | 13 | 13 |
| 5 | 6 | 6 | 11 | 12 | 13 |
| 5 | 6 | 6 | 11 | 11 | 12 |
| 4 | 5 | 5 | 10 | 10 | 11 |
| 4 | 4 | 5 | 9  | 9  | 10 |
| 3 | 3 | 4 | 8  | 8  | 9  |
| 3 | 3 | 3 | 7  | 7  | 8  |

SALE

-

3x  
-2

3x  
3

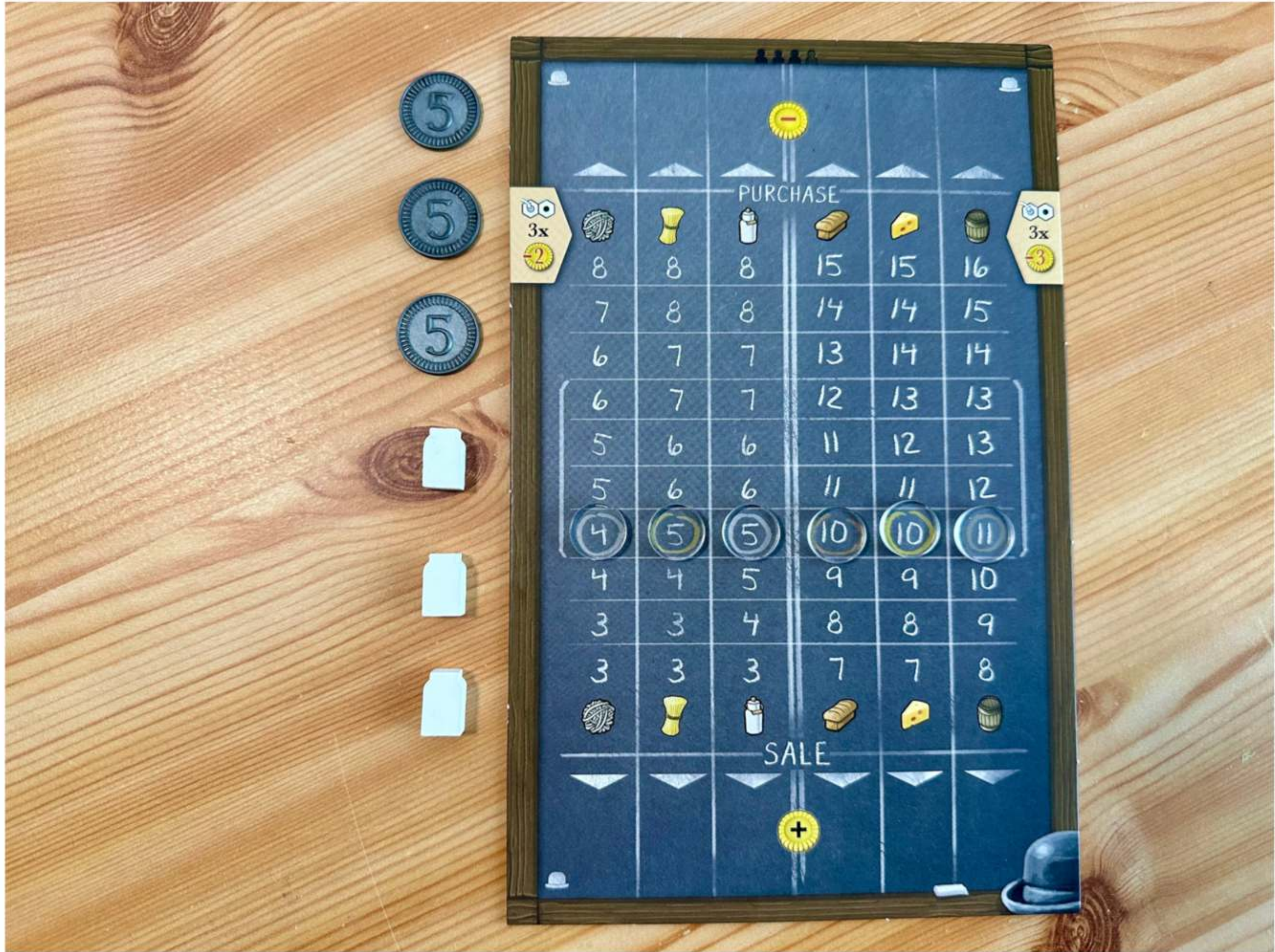


|   | PURCHASE |   |    |    |    |  |
|---|----------|---|----|----|----|--|
|   |          |   |    |    |    |  |
| 8 | 8        | 8 | 15 | 15 | 16 |  |
| 7 | 8        | 8 | 14 | 14 | 15 |  |
| 6 | 7        | 7 | 13 | 14 | 14 |  |
| 6 | 7        | 7 | 12 | 13 | 13 |  |
| 5 | 6        | 6 | 11 | 12 | 13 |  |
| 5 | 6        | 6 | 11 | 11 | 12 |  |
| 4 | 5        | 5 | 10 | 10 | 11 |  |
| 4 | 4        | 5 | 9  | 9  | 10 |  |
| 3 | 3        | 4 | 8  | 8  | 9  |  |
| 3 | 3        | 3 | 7  | 7  | 8  |  |
|   | SALE     |   |    |    |    |  |
|   |          |   |    |    |    |  |

3x

3x





4444

—

PURCHASE

|   |   |   |    |    |    |
|---|---|---|----|----|----|
| 8 | 8 | 8 | 15 | 15 | 16 |
| 7 | 8 | 8 | 14 | 14 | 15 |
| 6 | 7 | 7 | 13 | 14 | 14 |
| 6 | 7 | 7 | 12 | 13 | 13 |
| 5 | 6 | 6 | 11 | 12 | 13 |
| 5 | 6 | 6 | 11 | 11 | 12 |
| 4 | 5 | 5 | 10 | 10 | 11 |
| 4 | 4 | 5 | 9  | 9  | 10 |
| 3 | 3 | 4 | 8  | 8  | 9  |
| 3 | 3 | 3 | 7  | 7  | 8  |

SALE

+



| Wool     | Grain    | Milk     | Bread     | Cheese    | Whisky    |
|----------|----------|----------|-----------|-----------|-----------|
| 8        | 8        | 8        | 15        | 15        | 16        |
| 7        | 8        | 8        | 14        | 14        | 15        |
| 6        | 7        | 7        | 13        | 14        | 14        |
| 6        | 7        | 7        | 12        | 13        | 13        |
| 5        | 6        | 6        | 11        | 12        | 13        |
| 5        | 6        | 6        | 11        | 11        | 12        |
| <b>4</b> | <b>5</b> | <b>5</b> | <b>10</b> | <b>10</b> | <b>11</b> |
| 4        | 4        | 5        | 9         | 9         | 10        |
| 3        | 3        | 4        | 8         | 8         | 9         |

## Purchase 3 milk

| Wool     | Grain    | Milk     | Bread     | Cheese    | Whisky    |
|----------|----------|----------|-----------|-----------|-----------|
| 8        | 8        | 8        | 15        | 15        | 16        |
| 7        | 8        | 8        | 14        | 14        | 15        |
| 6        | 7        | 7        | 13        | 14        | 14        |
| 6        | 7        | 7        | 12        | 13        | 13        |
| 5        | 6        | 6        | 11        | 12        | 13        |
| 5        | 6        | 6        | 11        | 11        | 12        |
| <b>4</b> | <b>5</b> | <b>5</b> | <b>10</b> | <b>10</b> | <b>11</b> |
| 4        | 4        | 5        | 9         | 9         | 10        |
| 3        | 3        | 4        | 8         | 8         | 9         |

Purchase 3 milk

3 milk purchased

| Wool     | Grain    | Milk     | Bread     | Cheese    | Whisky    |
|----------|----------|----------|-----------|-----------|-----------|
| 8        | 8        | 8        | 15        | 15        | 16        |
| 7        | 8        | 8        | 14        | 14        | 15        |
| 6        | 7        | 7        | 13        | 14        | 14        |
| 6        | 7        | 7        | 12        | 13        | 13        |
| 5        | 6        | 6        | 11        | 12        | 13        |
| 5        | 6        | 6        | 11        | 11        | 12        |
| <b>4</b> | <b>5</b> | <b>5</b> | <b>10</b> | <b>10</b> | <b>11</b> |
| 4        | 4        | 5        | 9         | 9         | 10        |
| 3        | 3        | 4        | 8         | 8         | 9         |

Purchase 3 milk  
3 milk purchased  
Milk price changed to 7

| Wool     | Grain    | Milk     | Bread     | Cheese    | Whisky    |
|----------|----------|----------|-----------|-----------|-----------|
| 8        | 8        | 8        | 15        | 15        | 16        |
| 7        | 8        | 8        | 14        | 14        | 15        |
| 6        | 7        | 7        | 13        | 14        | 14        |
| 6        | 7        | <b>7</b> | 12        | 13        | 13        |
| 5        | 6        | 6        | 11        | 12        | 13        |
| 5        | 6        | 6        | 11        | 11        | 12        |
| <b>4</b> | <b>5</b> | 5        | <b>10</b> | <b>10</b> | <b>11</b> |
| 4        | 4        | 5        | 9         | 9         | 10        |
| 3        | 3        | 4        | 8         | 8         | 9         |

Purchase 3 milk

3 milk purchased

Milk price changed to 7

Sell 2 bread

| Wool     | Grain    | Milk     | Bread     | Cheese    | Whisky    |
|----------|----------|----------|-----------|-----------|-----------|
| 8        | 8        | 8        | 15        | 15        | 16        |
| 7        | 8        | 8        | 14        | 14        | 15        |
| 6        | 7        | 7        | 13        | 14        | 14        |
| 6        | 7        | <b>7</b> | 12        | 13        | 13        |
| 5        | 6        | 6        | 11        | 12        | 13        |
| 5        | 6        | 6        | 11        | 11        | 12        |
| <b>4</b> | <b>5</b> | 5        | <b>10</b> | <b>10</b> | <b>11</b> |
| 4        | 4        | 5        | 9         | 9         | 10        |
| 3        | 3        | 4        | 8         | 8         | 9         |

Purchase 3 milk

3 milk purchased

Milk price changed to 7

Sell 2 bread

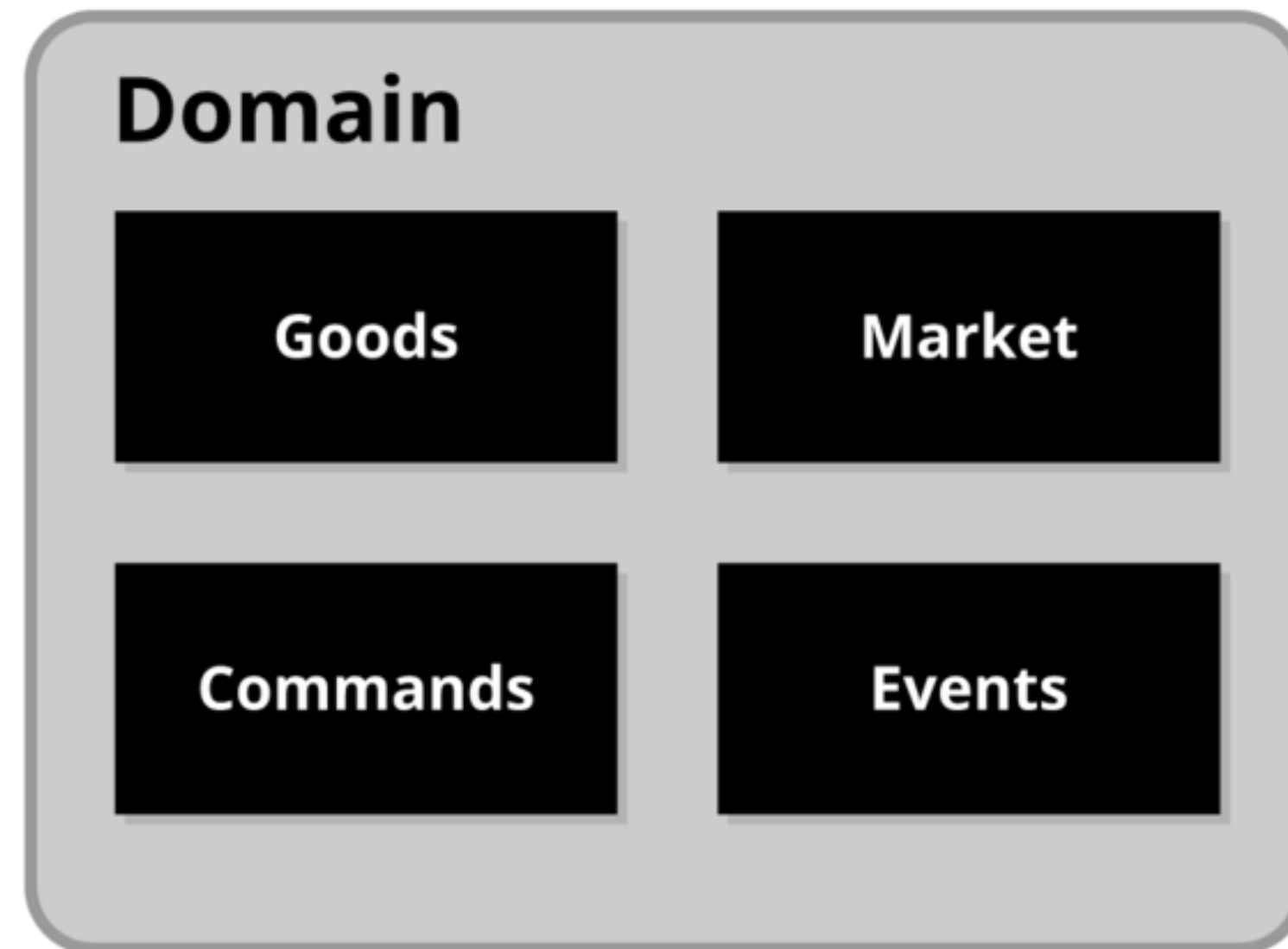
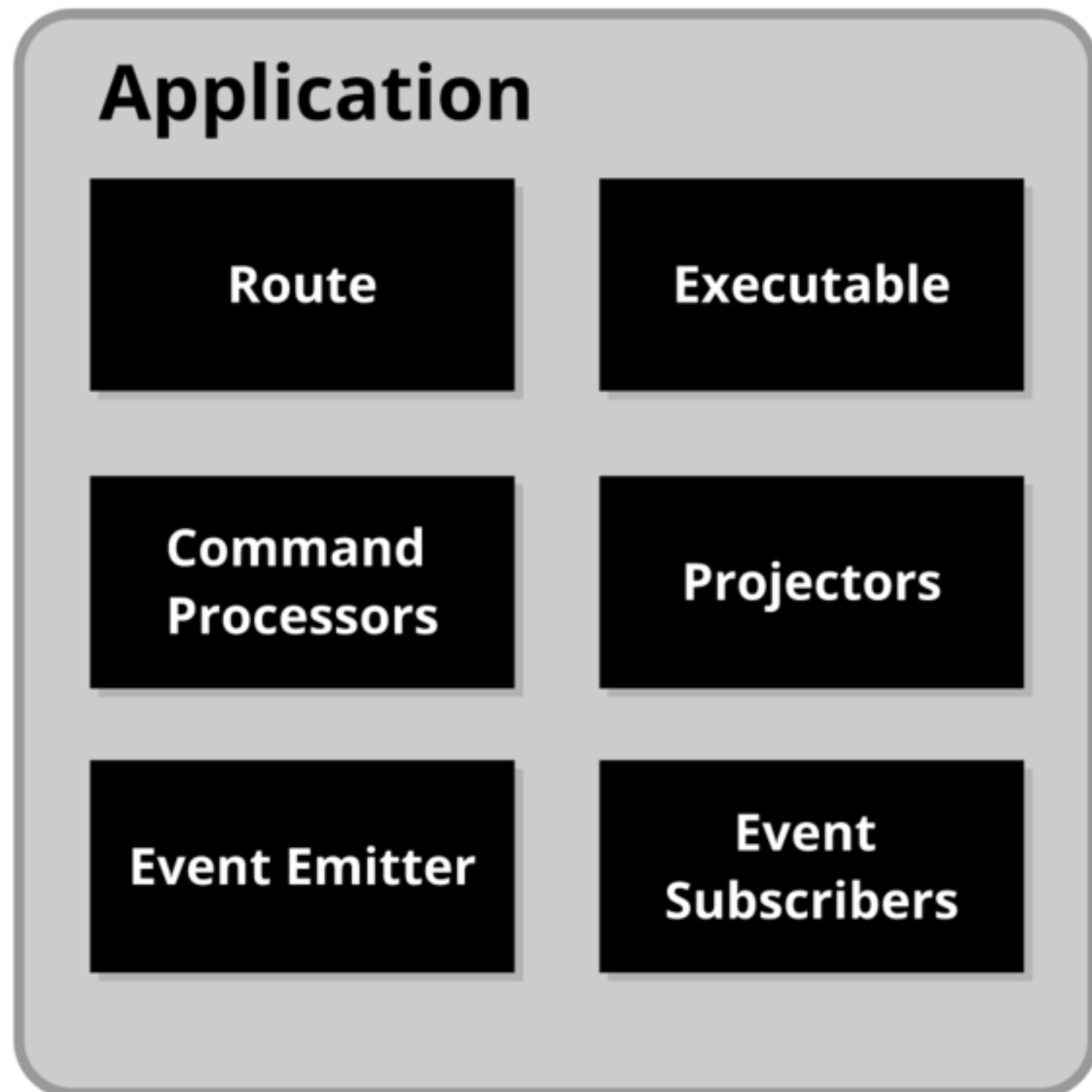
2 bread sold

| Wool     | Grain    | Milk     | Bread     | Cheese    | Whisky    |
|----------|----------|----------|-----------|-----------|-----------|
| 8        | 8        | 8        | 15        | 15        | 16        |
| 7        | 8        | 8        | 14        | 14        | 15        |
| 6        | 7        | 7        | 13        | 14        | 14        |
| 6        | 7        | <b>7</b> | 12        | 13        | 13        |
| 5        | 6        | 6        | 11        | 12        | 13        |
| 5        | 6        | 6        | 11        | 11        | 12        |
| <b>4</b> | <b>5</b> | 5        | <b>10</b> | <b>10</b> | <b>11</b> |
| 4        | 4        | 5        | 9         | 9         | 10        |
| 3        | 3        | 4        | 8         | 8         | 9         |

Purchase 3 milk  
3 milk purchased  
Milk price changed to 7

Sell 2 bread  
2 bread sold  
Bread price changed to 8

| Wool     | Grain    | Milk     | Bread    | Cheese    | Whisky    |
|----------|----------|----------|----------|-----------|-----------|
| 8        | 8        | 8        | 15       | 15        | 16        |
| 7        | 8        | 8        | 14       | 14        | 15        |
| 6        | 7        | 7        | 13       | 14        | 14        |
| 6        | 7        | <b>7</b> | 12       | 13        | 13        |
| 5        | 6        | 6        | 11       | 12        | 13        |
| 5        | 6        | 6        | 11       | 11        | 12        |
| <b>4</b> | <b>5</b> | 5        | 10       | <b>10</b> | <b>11</b> |
| 4        | 4        | 5        | 9        | 9         | 10        |
| 3        | 3        | 4        | <b>8</b> | 8         | 9         |





## Framework

Database

### Events

Dispatcher

Reader

Writer

### Http

Request

Response

Router

Executable

Kernel

Run application with request received from client

GET request?

Select query to be performed using GET routes

Execute query

Generate response based on projections

Return response generated by query

POST request?

Select command to be processed using POST routes

Execute command

Source domain objects from events

Perform operations using domain objects

Emit new events

Generate response

Return response generated by command

Send response returned from application to client

```
26 final class PurchaseGoodCommandProcessorTest extends TestCase
27 {
28     #[TestDox('Only a GoodPurchasedEvent is emitted when the price does not change')]
29     public function testEmitsGoodPurchasedEvent(): void
30     {
31         $sourcer = $this->createStub(MarketSourcer::class);
32
33         $sourcer
34             ->method('source')
35             ->willReturn(Market::from(4, 3, 3, 3, 3));
36
37         $emitter = $this->createMock(EventEmitter::class);
38
39         $emitter
40             ->expects($this->once())
41             ->method('goodPurchased')
42             ->with(Good::bread(), Price::from(11), 1);
43
44         $emitter
45             ->expects($this->never())
46             ->method('goodSold');
47
48         $emitter
49             ->expects($this->never())
50             ->method('priceChanged');
51
52         $processor = new PurchaseGoodCommandProcessor($emitter, $sourcer);
53
54         $processor->process(new PurchaseGoodCommand(Good::bread(), 1));
55     }
}

4 use example\caledonia\domain\PurchaseGoodCommand;
5
6 final readonly class PurchaseGoodCommandProcessor
7 {
8     private EventEmitter $emitter;
9     private MarketSourcer $sourcer;
10
11     public function __construct(EventEmitter $emitter, MarketSourcer $sourcer)
12     {
13         $this->emitter = $emitter;
14         $this->sourcer = $sourcer;
15     }
16
17     public function process(PurchaseGoodCommand $command): void
18     {
19         $market = $this->sourcer->source();
20
21         $soldPrice = $market->priceFor($command->good());
22
23         $market = $market->purchase($command->good(), $command->amount());
24
25         $this->emitter->goodPurchased($command->good(), $soldPrice, $command->amount());
26
27         $newPrice = $market->priceFor($command->good());
28
29         if (!$newPrice->equals($soldPrice)) {
30             $this->emitter->priceChanged($command->good(), $soldPrice, $newPrice);
31         }
32     }
33 }
```

Terminal Local

```
caledonia main ./tools/phpunit --no-progress --testdox --filter PurchaseGoodCommandProcessorTest
PHPUnit 10.4.1 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.11
Configuration: /home/sb/examples/caledonia/phpunit.xml

Time: 00:00.006, Memory: 24.52 MB

PurchaseGoodCommandProcessor (example\caledonia\application\PurchaseGoodCommandProcessor)
✓ Only a GoodPurchasedEvent is emitted when the price does not change
✓ A GoodPurchasedEvent and a PriceChangedEvent are emitted when the price changes

OK (2 tests, 2 assertions)
```

Let us have a look into the code ...

Given event(s)

When command

Then event(s)

**Given event(s)**

**When command**

**Then error**

```
1 $this->given();
2
3 $this->when(new PurchaseGoodCommand(Good::bread(), 1));
4
5 $this->then(
6     new GoodPurchasedEvent(
7         Uuid::from('ed486fb4-5b44-4821-8958-1303b30330d3'),
8         Good::bread(),
9         Price::from(10),
10        1,
11    ),
12    new PriceChangedEvent(
13        Uuid::from('f8db80dc-68fe-47b6-866d-18ba8c7c0609'),
14        Good::bread(),
15        Price::from(10)
```

```
1 $this->given();
2
3 $this->when(new PurchaseGoodCommand(Good::bread(), 1));
4
5 $this->then(
6     new GoodPurchasedEvent(
7         Uuid::from('ed486fb4-5b44-4821-8958-1303b30330d3'),
8         Good::bread(),
9         Price::from(10),
10        1,
11    ),
12    new PriceChangedEvent(
13        Uuid::from('f8db80dc-68fe-47b6-866d-18ba8c7c0609'),
14        Good::bread(),
15        Price::from(10)
```

```
1 $this->given();
2
3 $this->when(new PurchaseGoodCommand(Good::bread(), 1));
4
5 $this->then(
6     new GoodPurchasedEvent(
7         Uuid::from('ed486fb4-5b44-4821-8958-1303b30330d3'),
8         Good::bread(),
9         Price::from(10),
10        1,
11    ),
12    new PriceChangedEvent(
13        Uuid::from('f8db80dc-68fe-47b6-866d-18ba8c7c0609'),
14        Good::bread(),
15        Price::from(10)
```



# Domain-Specific Fixture Setup

(based on event log)

# Domain-Specific Verification

(based on event log)

# Domain-Specific Logging

(generate nice images with sticky notes when tests are run)

# Scope-Specific Implementation

(using test doubles for unit tests)

# Scope-Specific Implementation

(using event system for integration tests)

-  <https://thephp.cc>
-  [sebastian@thephp.cc](mailto:sebastian@thephp.cc)
-  [@sebastian@thephpcc.social](#)  
[@sebastian@phpc.social](#)

# Image Credits

- <https://unsplash.com/de/fotos/MEUvVqkU3QI>