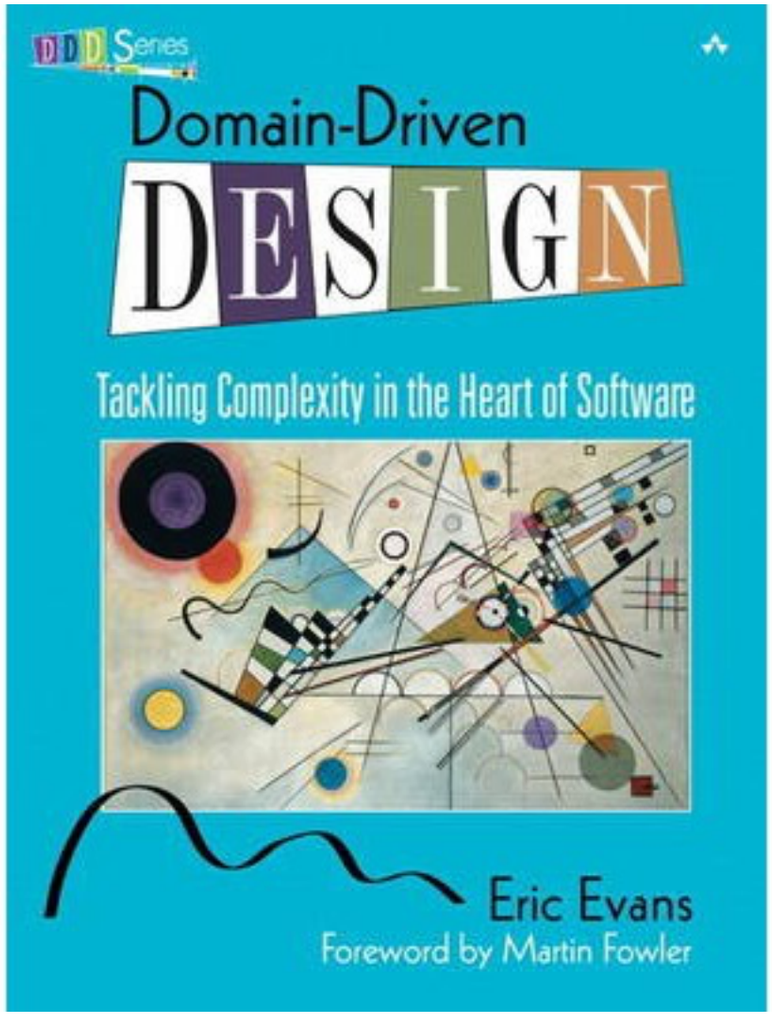
DOMAIN DRIVEN DESIGN


DDD Series
Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

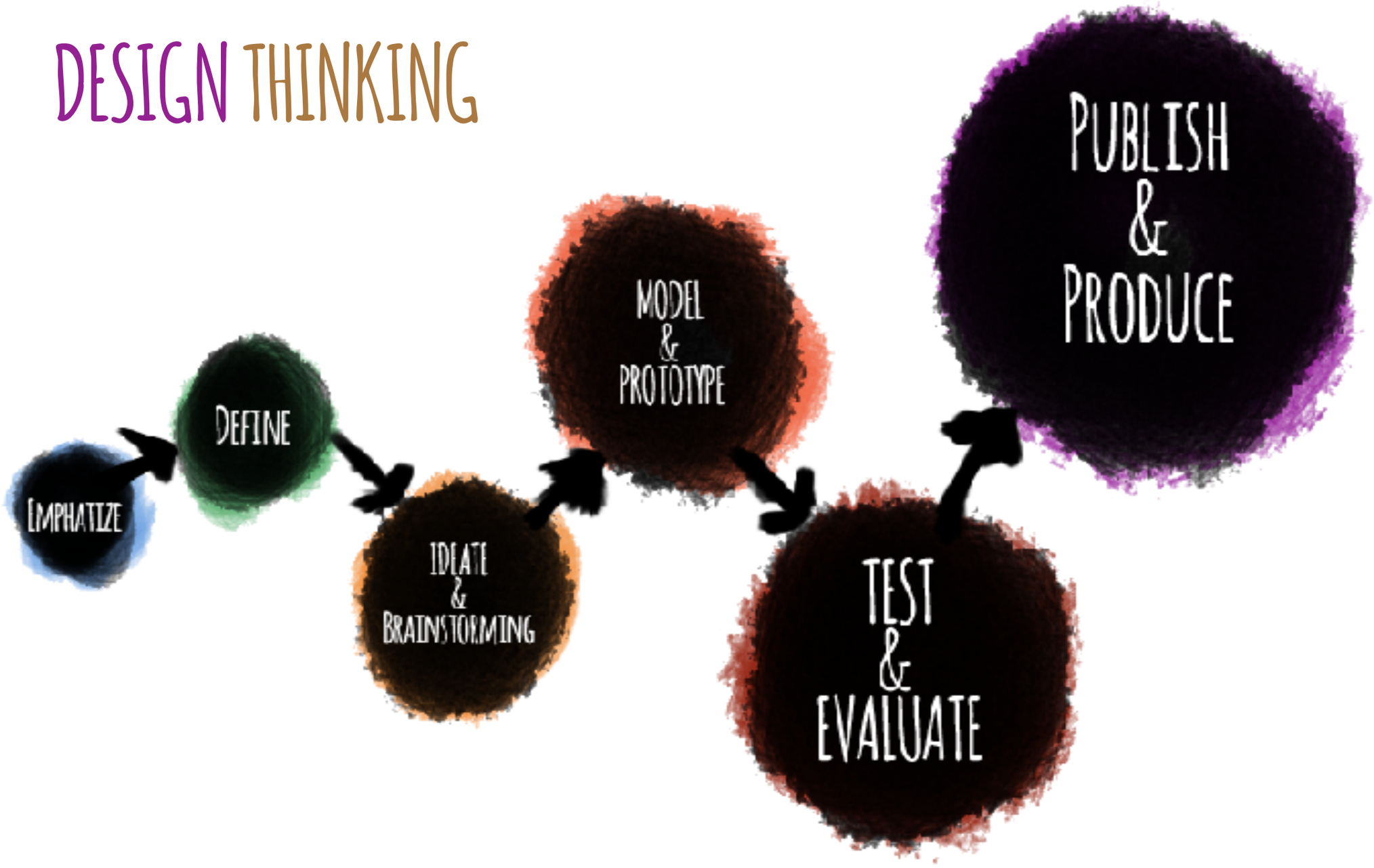

Presentation layer
Business layer
Persistence

Big Ball of Mud

DE-SIGN(NOUN)

THE ARRANGEMENT OF ELEMENTS OR DETAILS IN A PRODUCT OR WORK OF ART

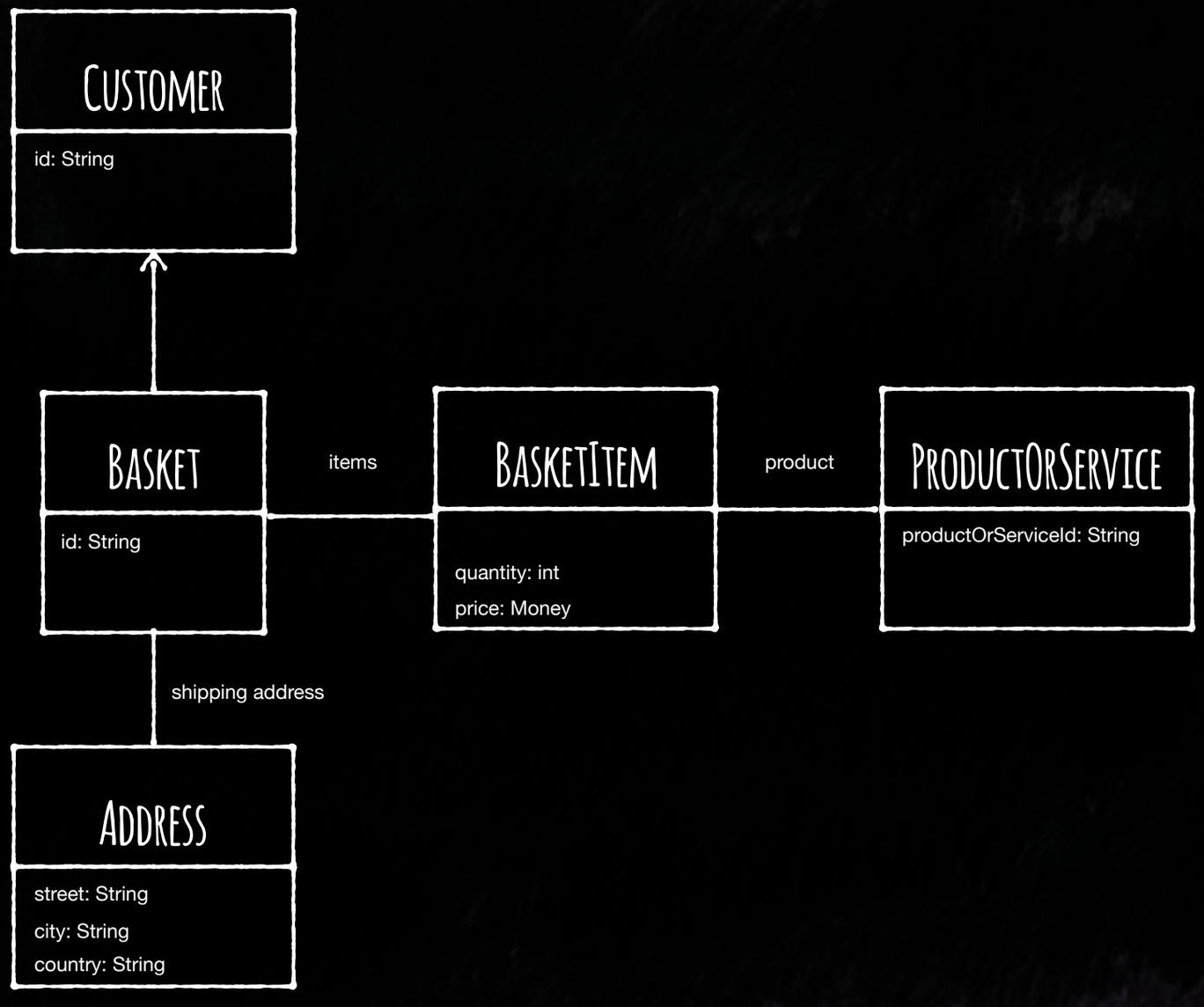
DESIGN THINKING

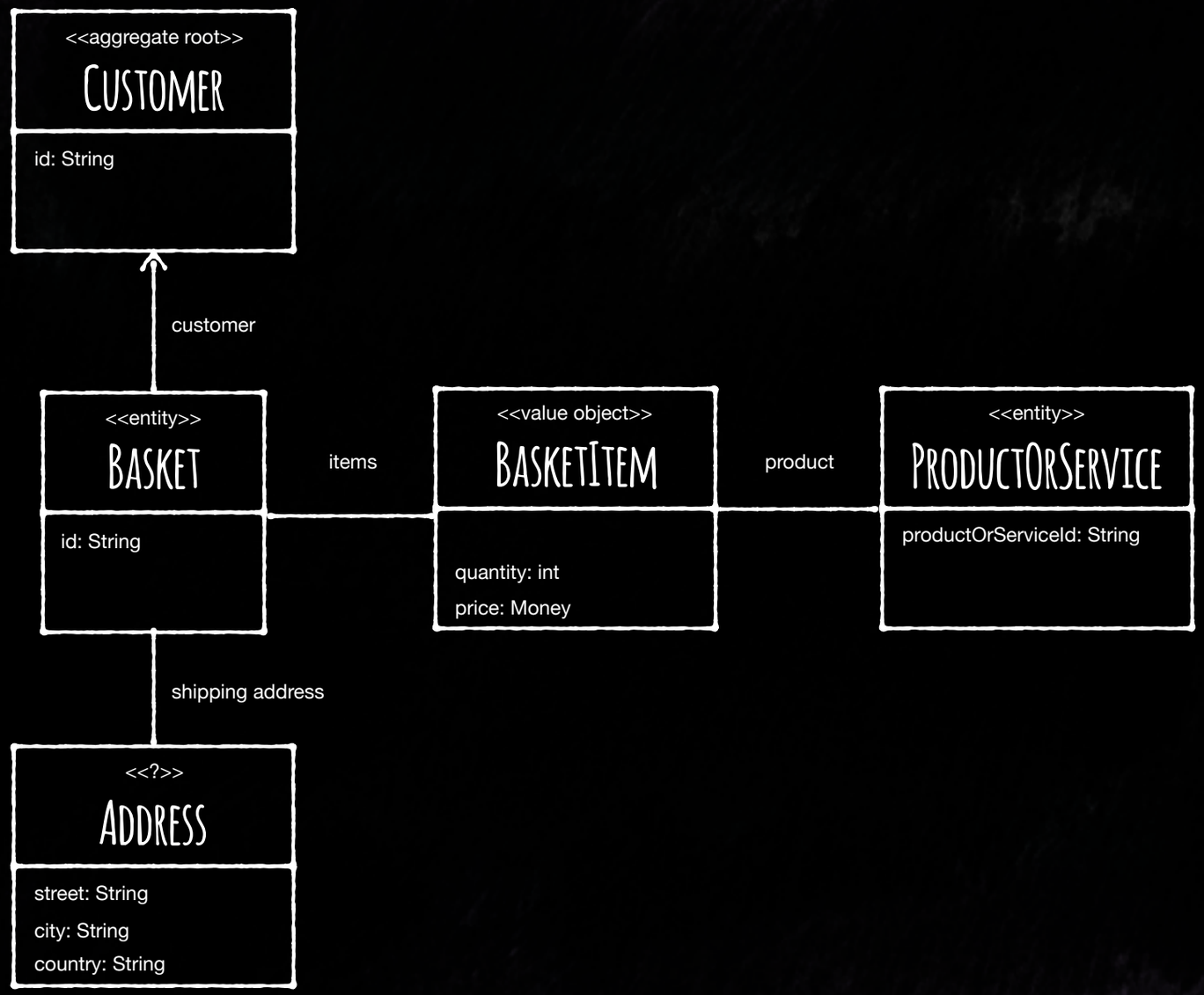


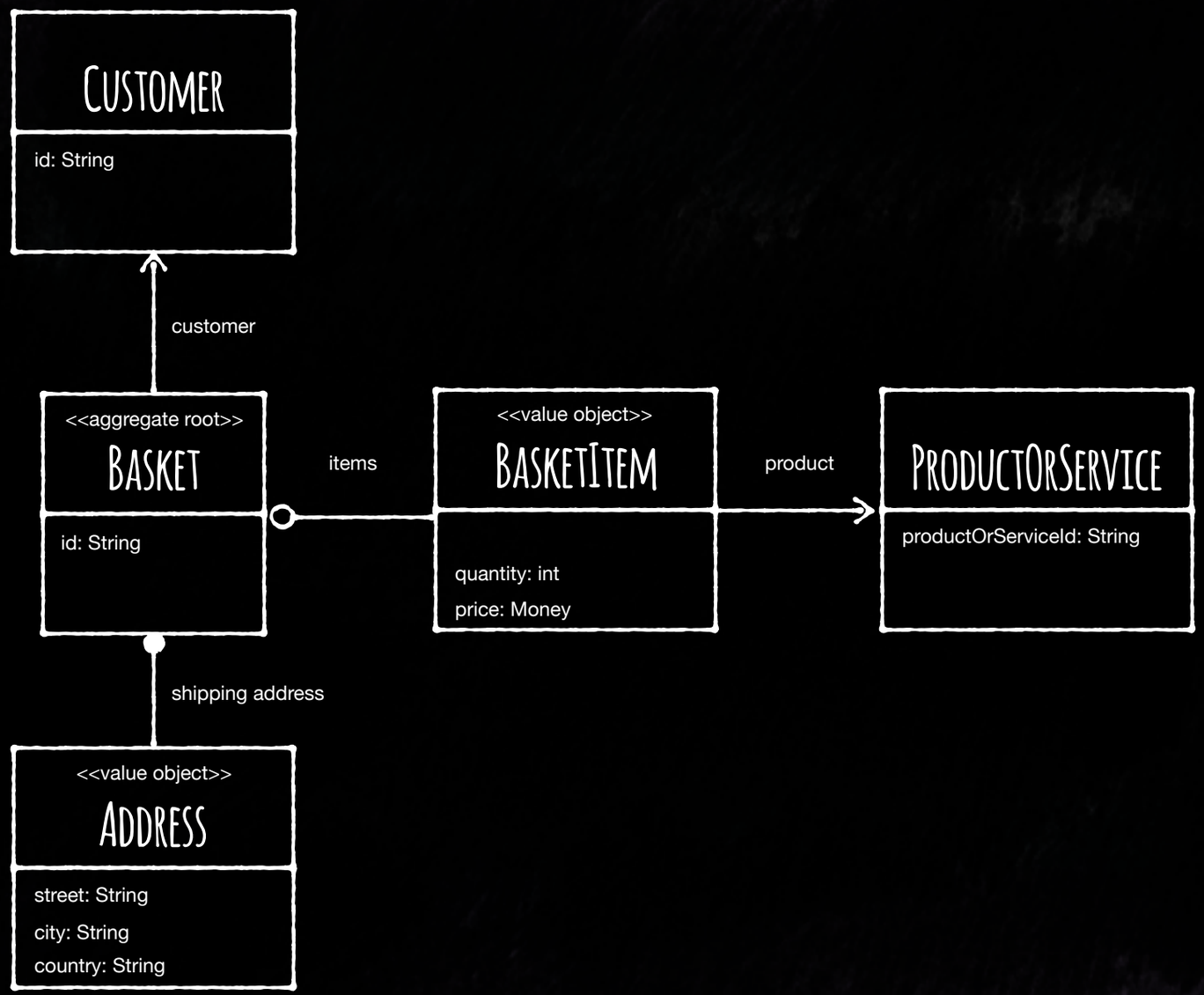
A good design is not the one that correctly predicts the future,
it's one that makes adapting to the future affordable.

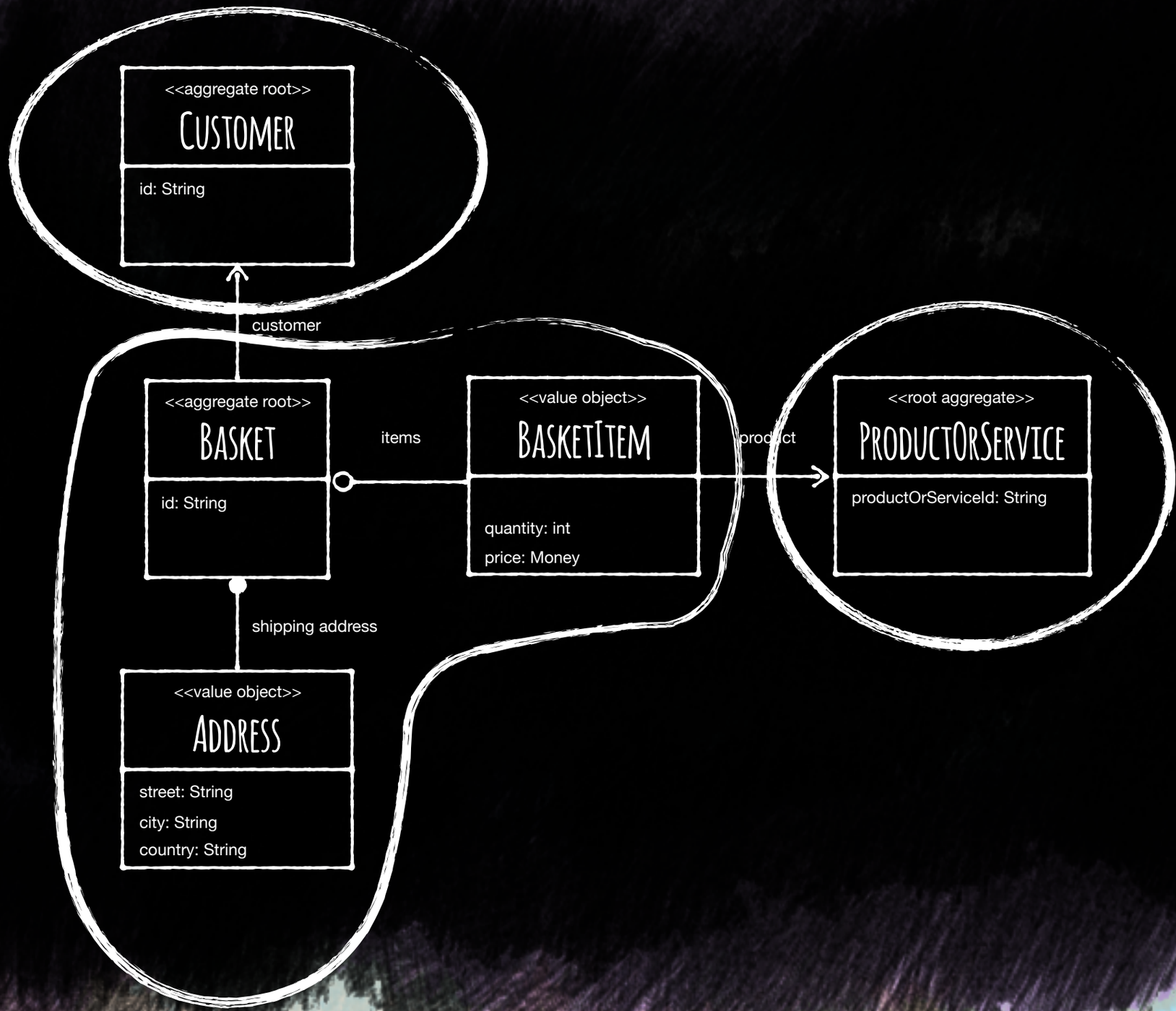
(Venkat Subramaniam)

Common Language


Customer
id: String
Basket
id: String
items
BasketItem
quantity: int
price: Money
product
ProductOrService
productOrServiceId: String
shipping address
Address
street: String
city: String
country: String


<<aggregate root>>
Customer
id: String
customer
<<entity>>
Basket
id: String
items
<<value object>>
BasketItem
quantity: int
price: Money
product
<<entity>>
ProductOrService
productOrServiceId: String
shipping address
<<?>>
Address
street: String
city: String
country: String


Customer
id: String
customer
<<aggregate root>>
Basket
id: String
items
<<value object>>
BasketItem
quantity: int
price: Money
product
ProductOrService
productOrServiceId: String
shipping address
<<value object>>
Address
street: String
city: String
country: String


<<aggregate root>>
Customer
id: String
customer
<<aggregate root>>
Basket
id: String
items
<<value object>>
BasketItem
quantity: int
price: Money
product
<<root aggregate>>
ProductOrService
productOrServiceId: String
shipping address
<<value object>>
Address
street: String
city: String
country: String

Factory

Repository

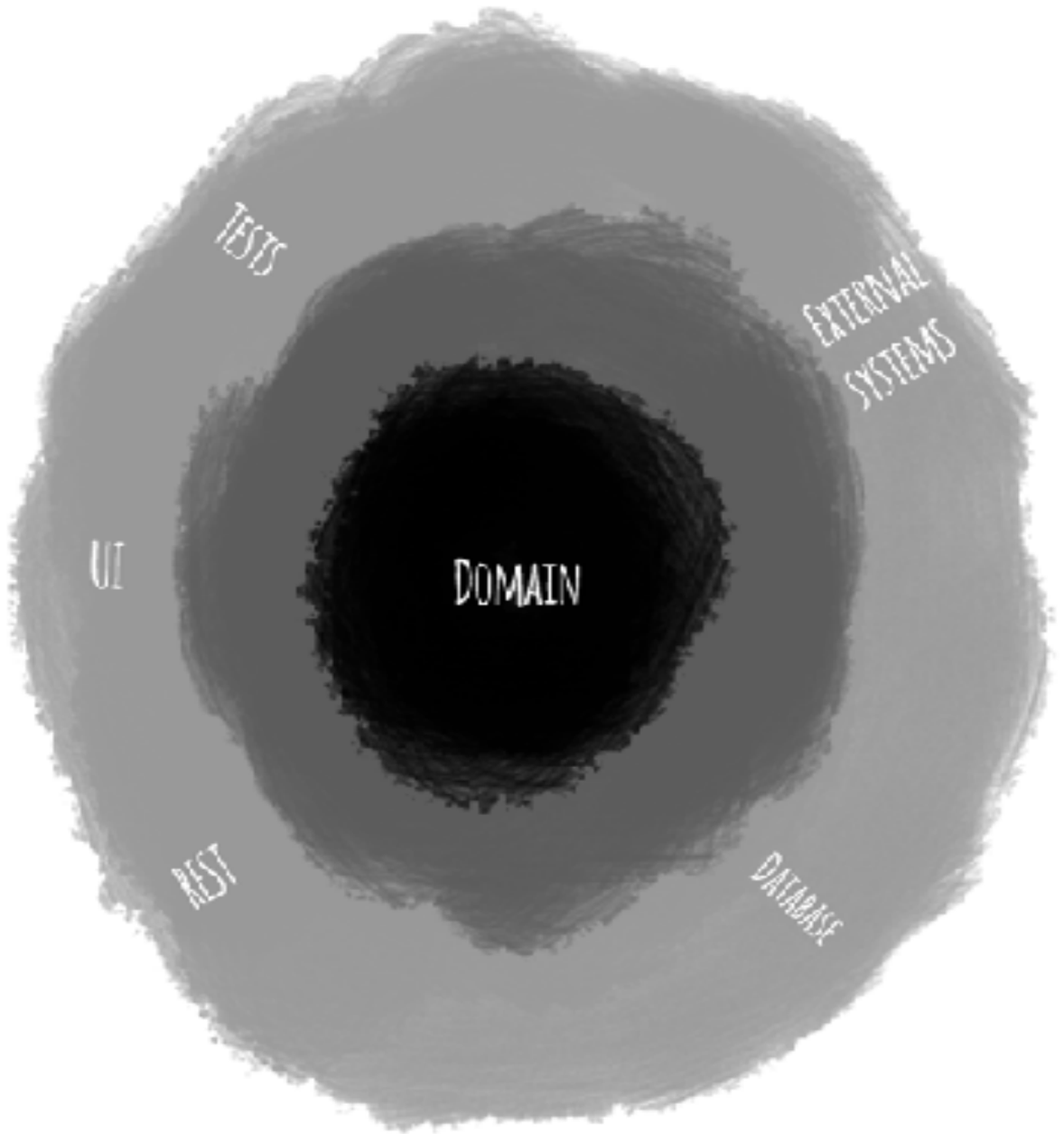
```
public interface OrderRepository {
    Order findBy(String id);
    List<Order> query(Criteria c);
    void save(Order o);
    void remove(Order o);
}
```

Specification

```
public interface Specification<T> {
    bool isSatisfiedBy(T obj);
}
```

Specification

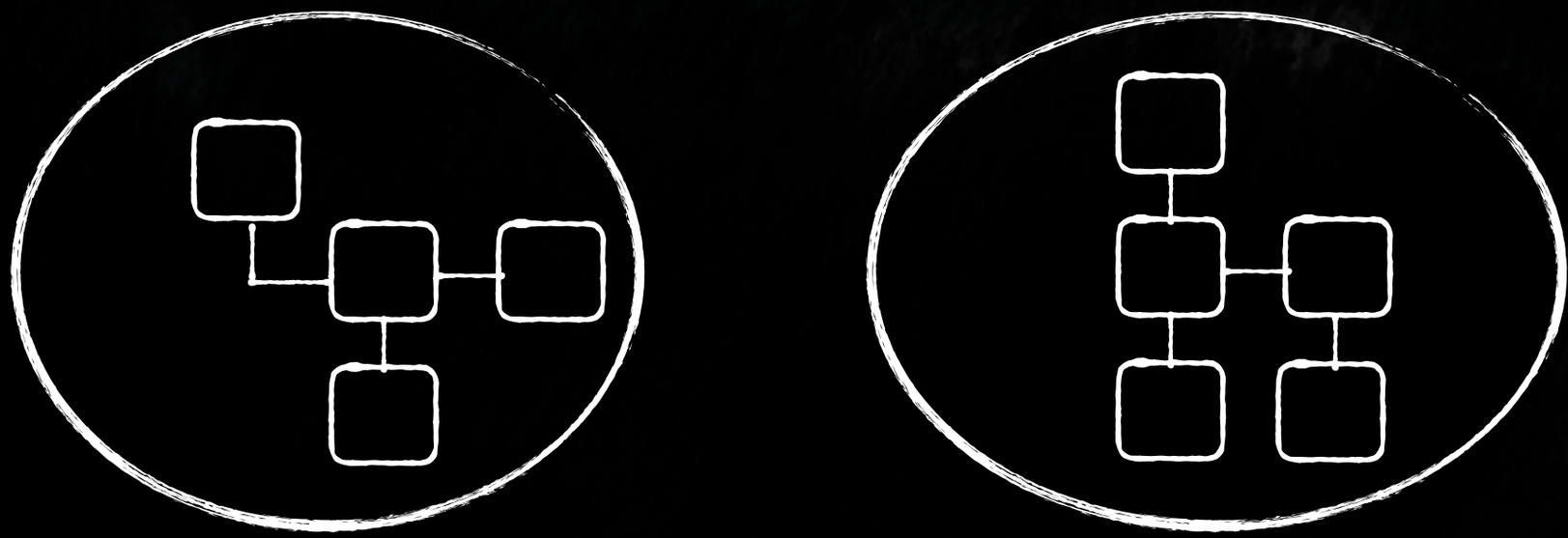
```
public class OverdueInvoiceSpecification {

  private final Date currentDate;

  public OverdueInvoiceSpecification(Date currentDate) {
    this.currentDate = currentDate;
  }

  public boolean isSatisfiedBy(Invoice invoice) {
    int gracePeriod = invoice.customer().getPaymentGracePeriod();
    Date deadline = invoice.dueDate() + gracePeriod;
    return currentDate.after(deadline);
  }
}
```

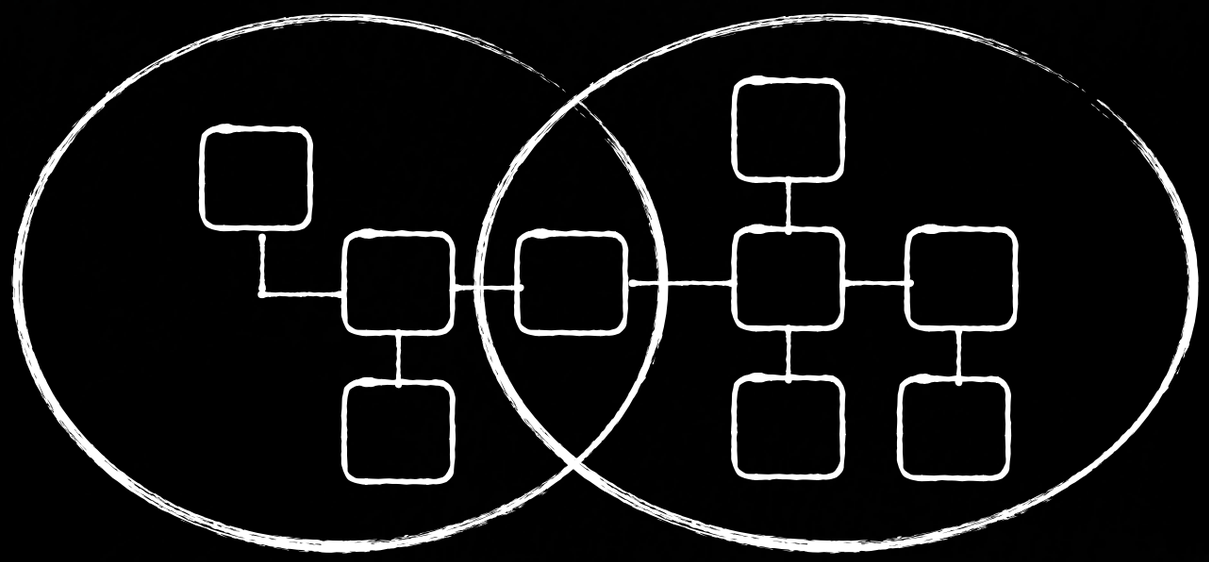

Tests
External systems
UI
Domain
REST
Database

Service

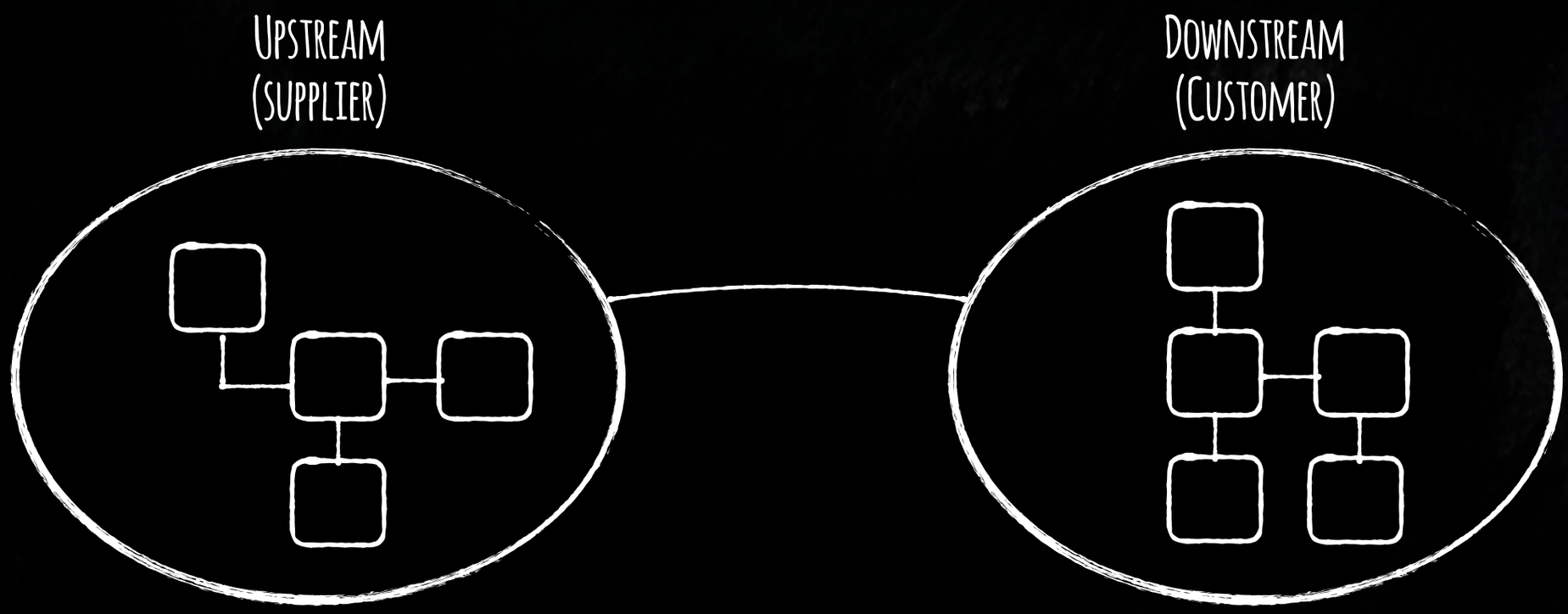
Application / Infrastructure / Domain

Context Mapping



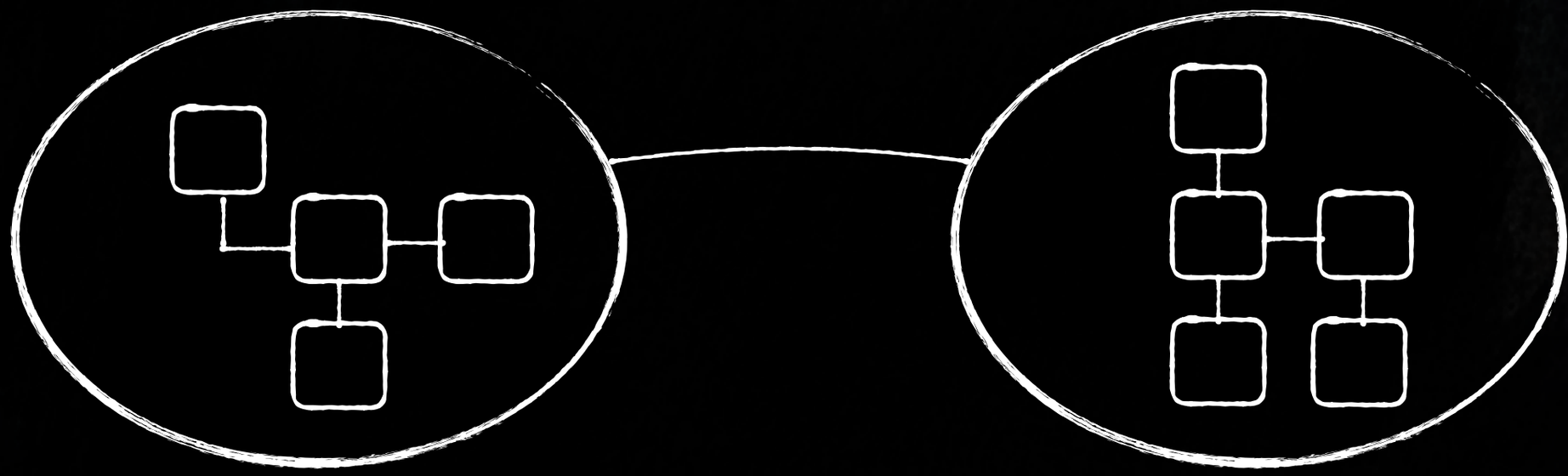


Shared Kernel



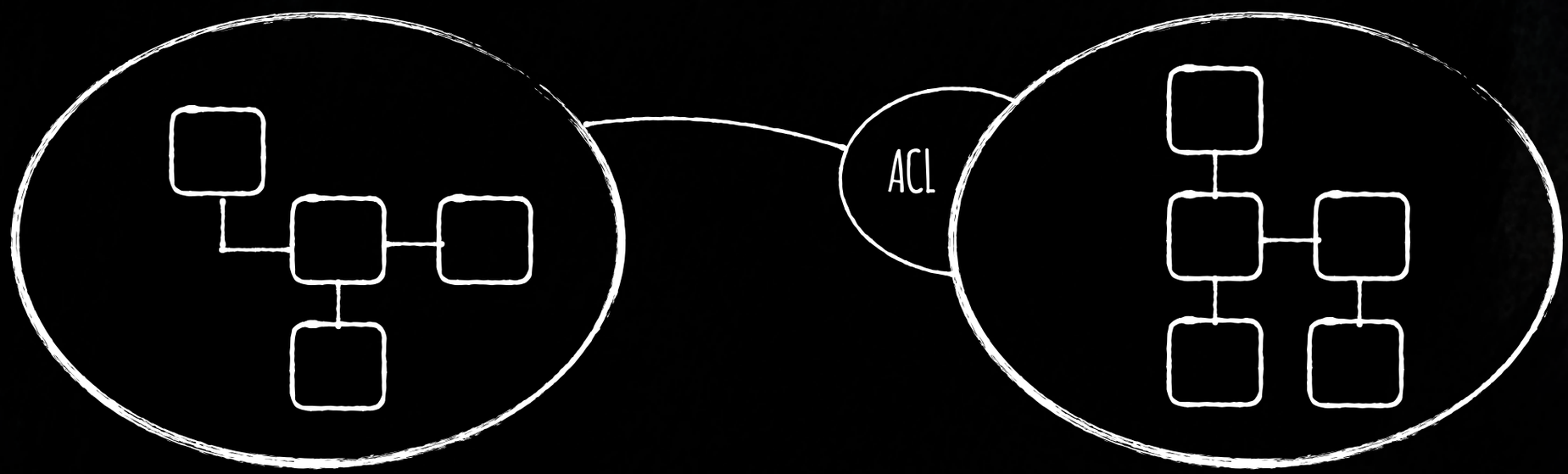
Customer - Supplier

Upstream

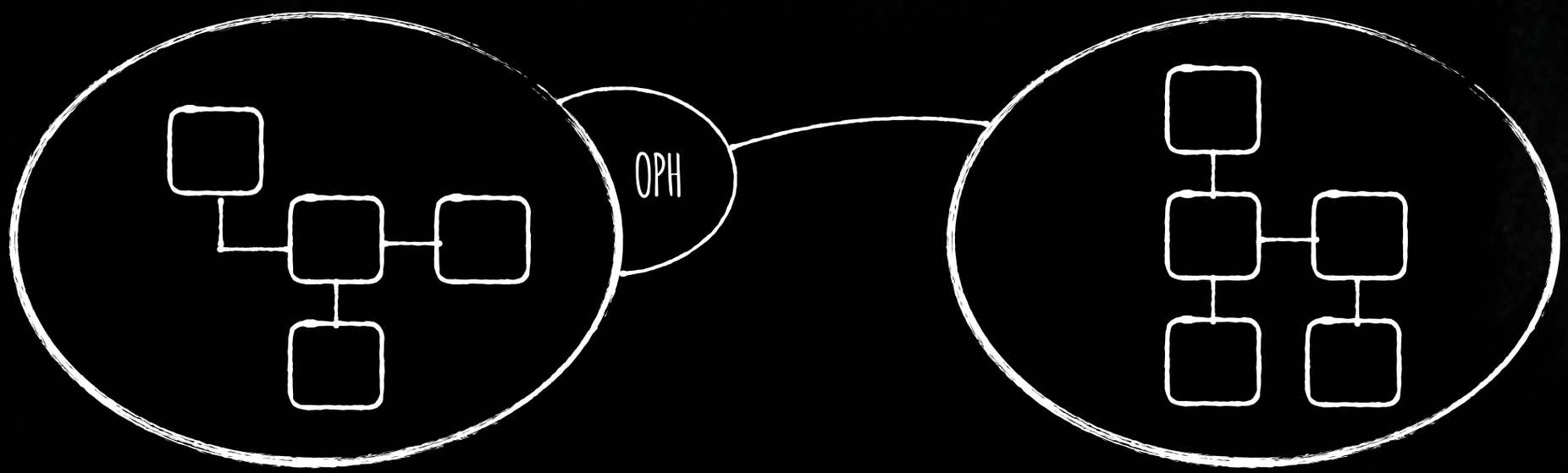


Downstream

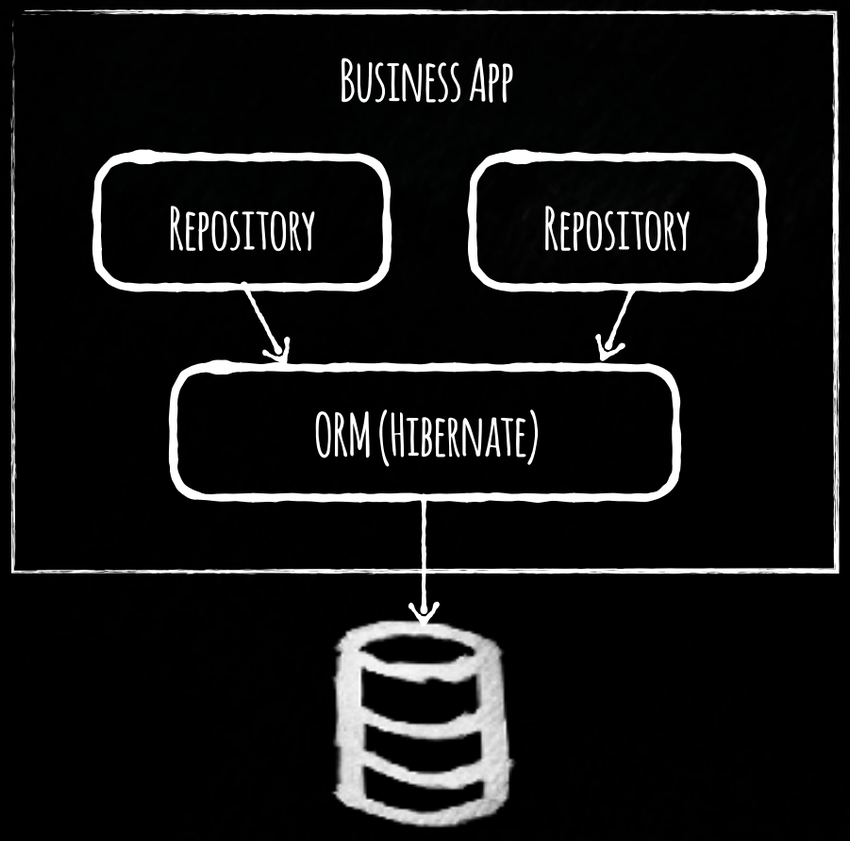
Conformist



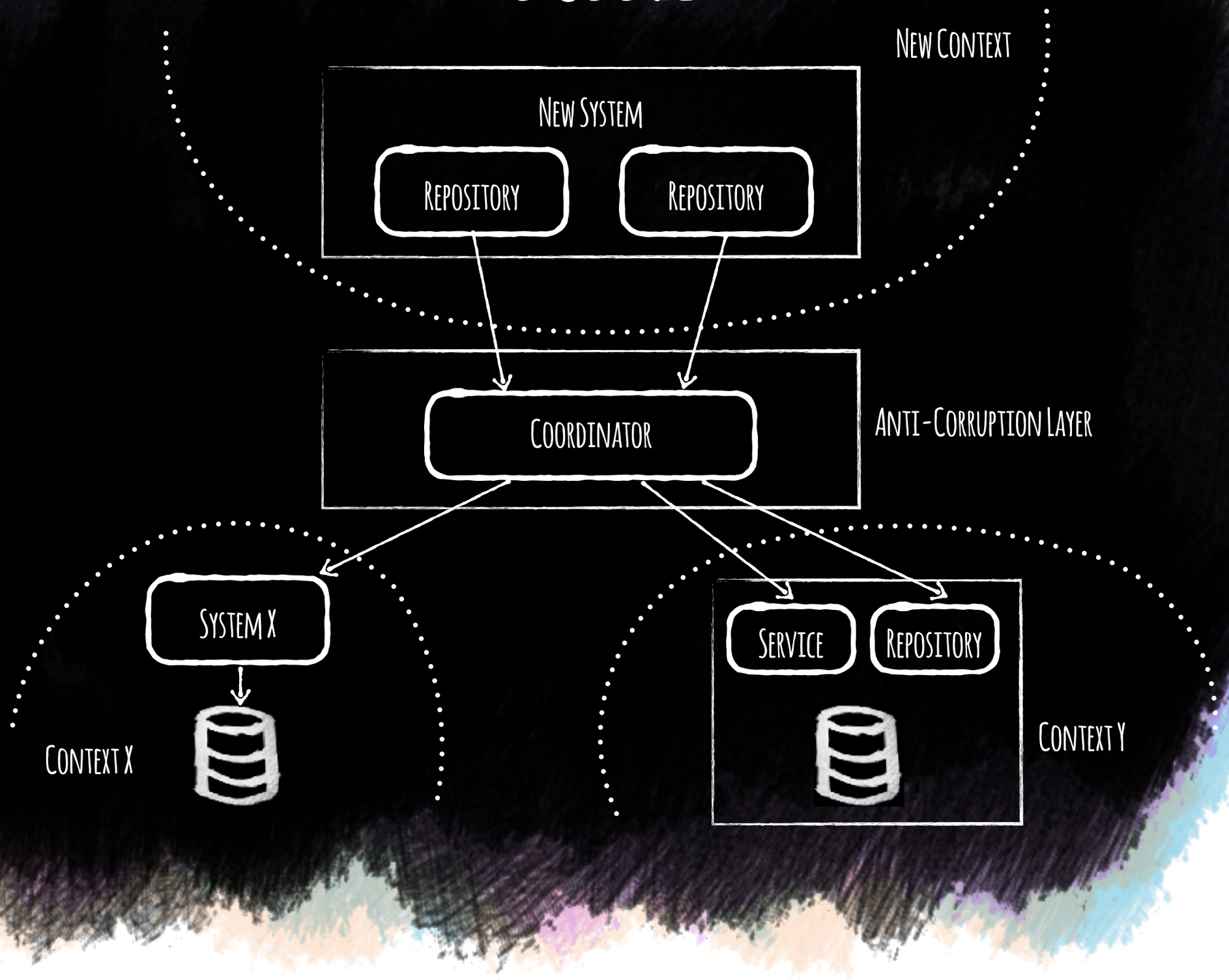
Anti-Corruption Layer



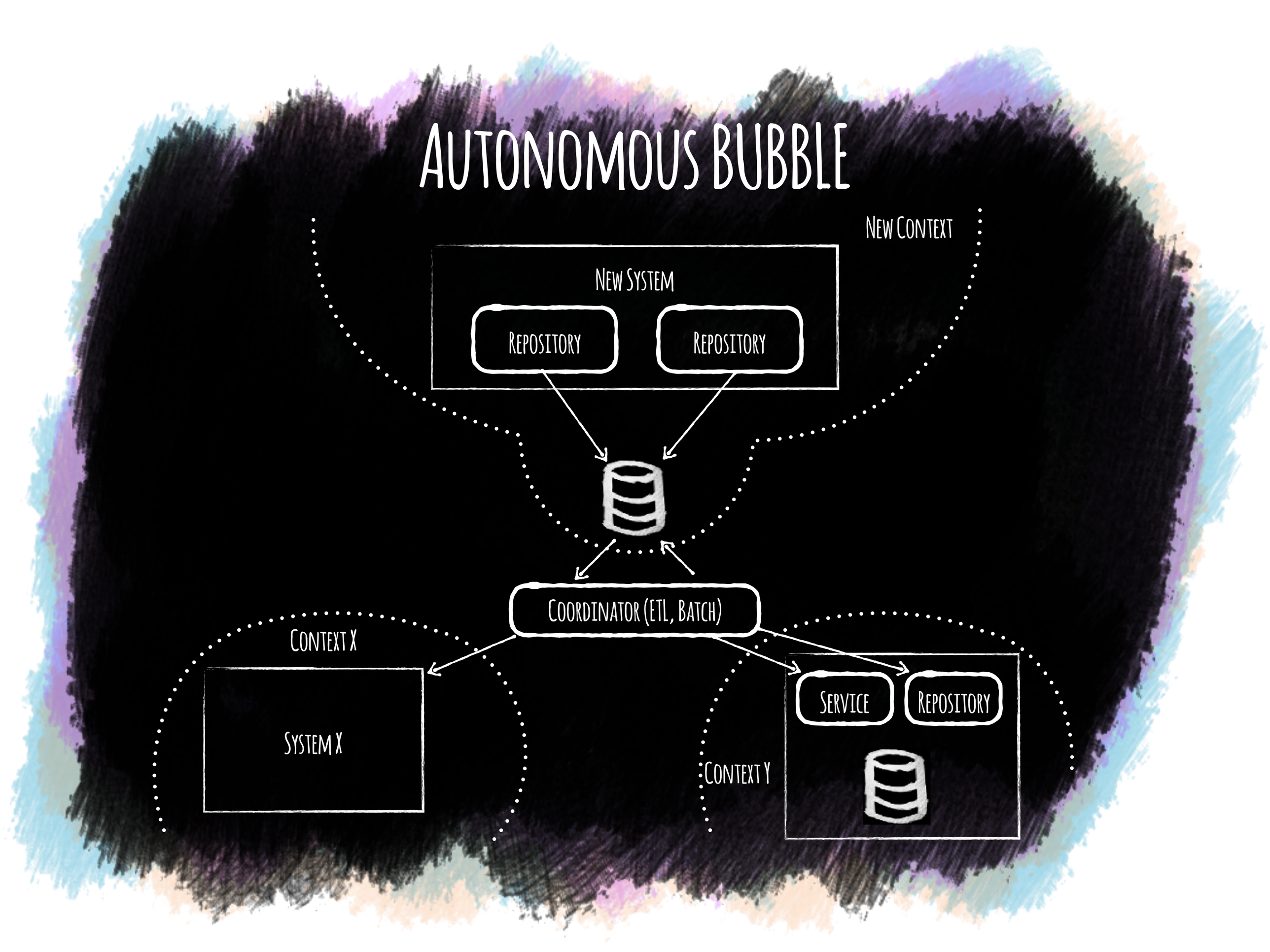
Open Host Service


Business App
Repository
Repository
ORM (Hibernate)

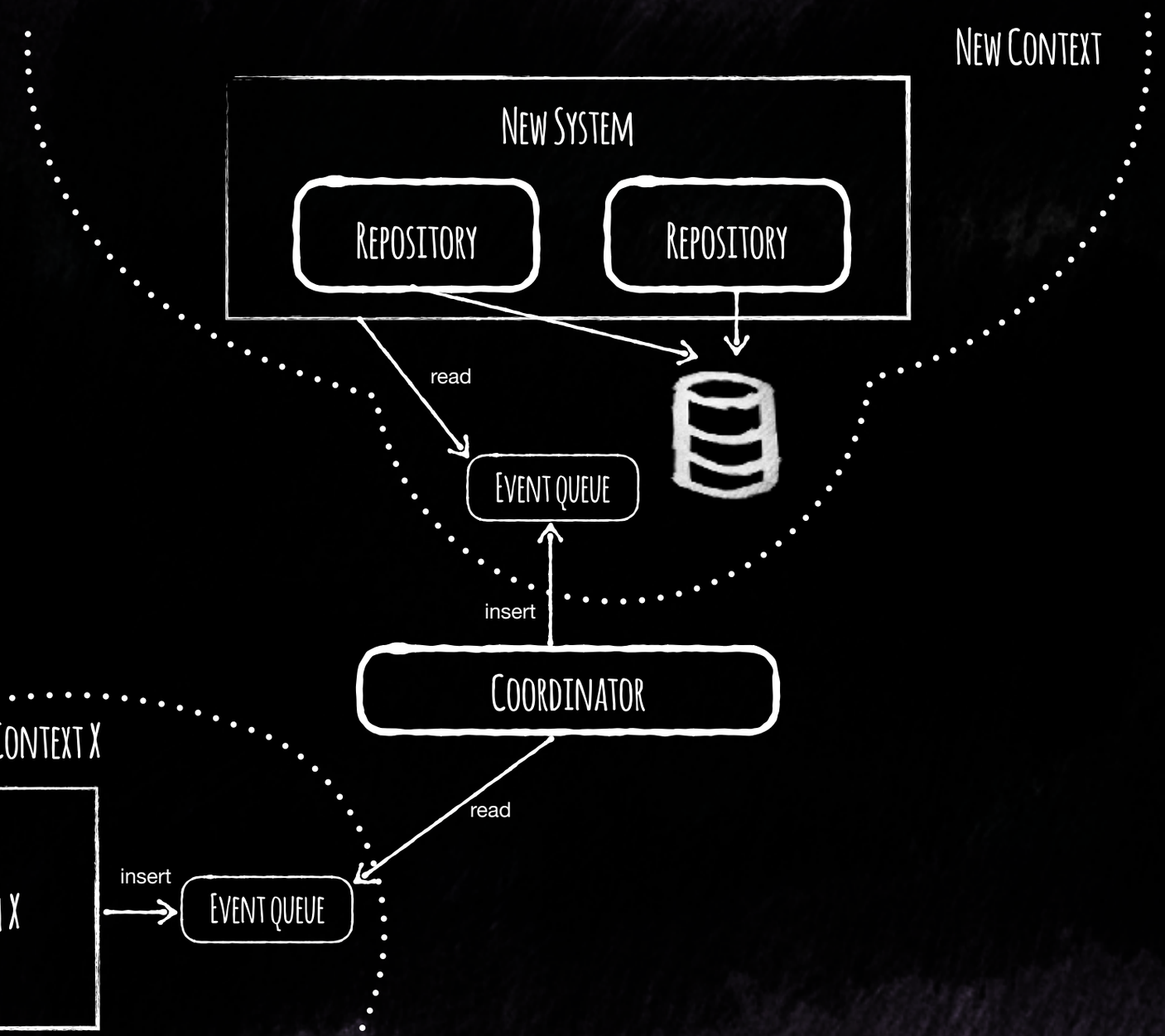
BUBBLE


New Context
New System
Repository
Repository
Coordinator
Anti-Corruption Layer
System X
Context X
Service
Repository
Context Y

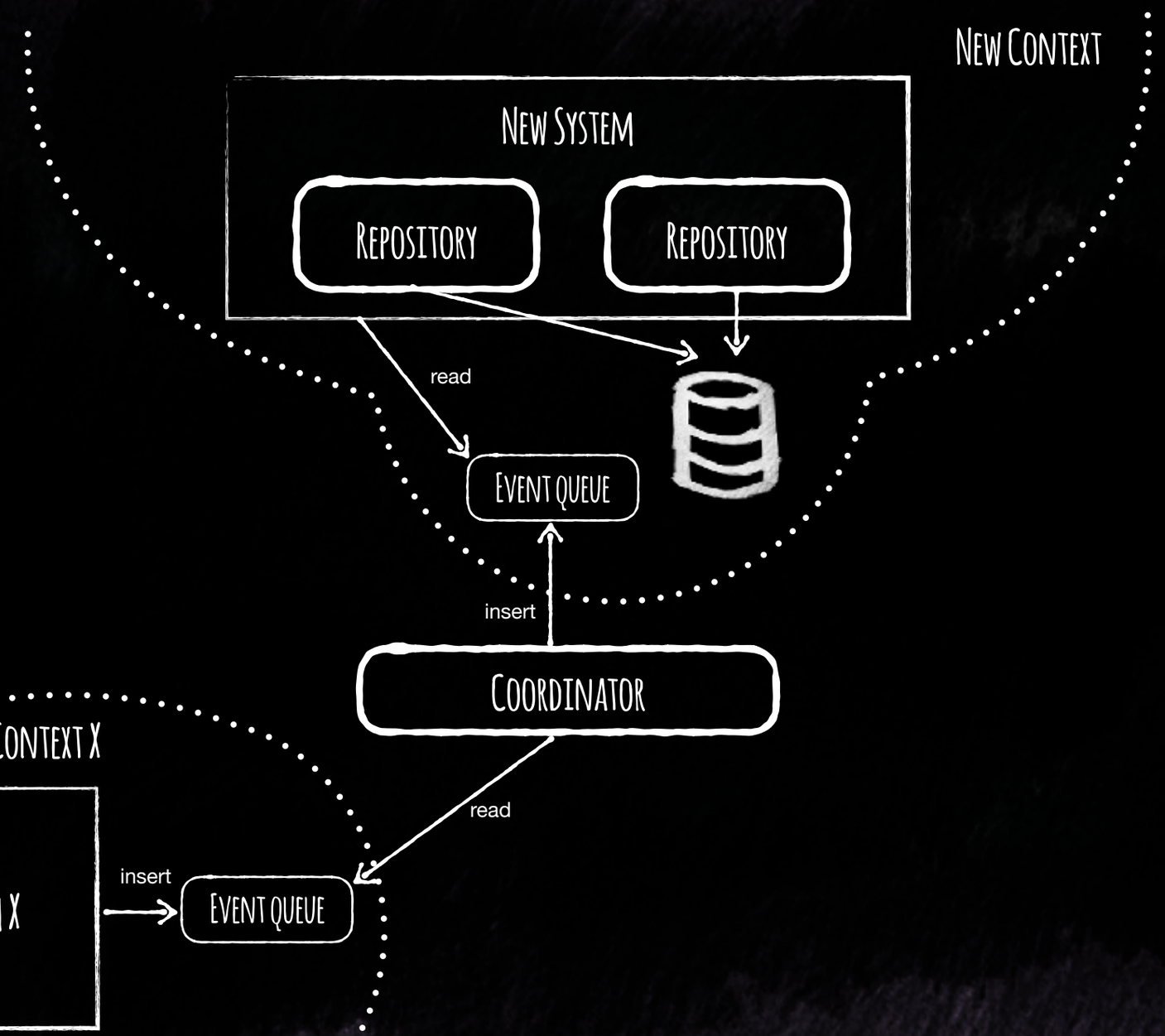
Autonomous Bubble



Autonomous Bubble



Autonomous Bubble



Events?

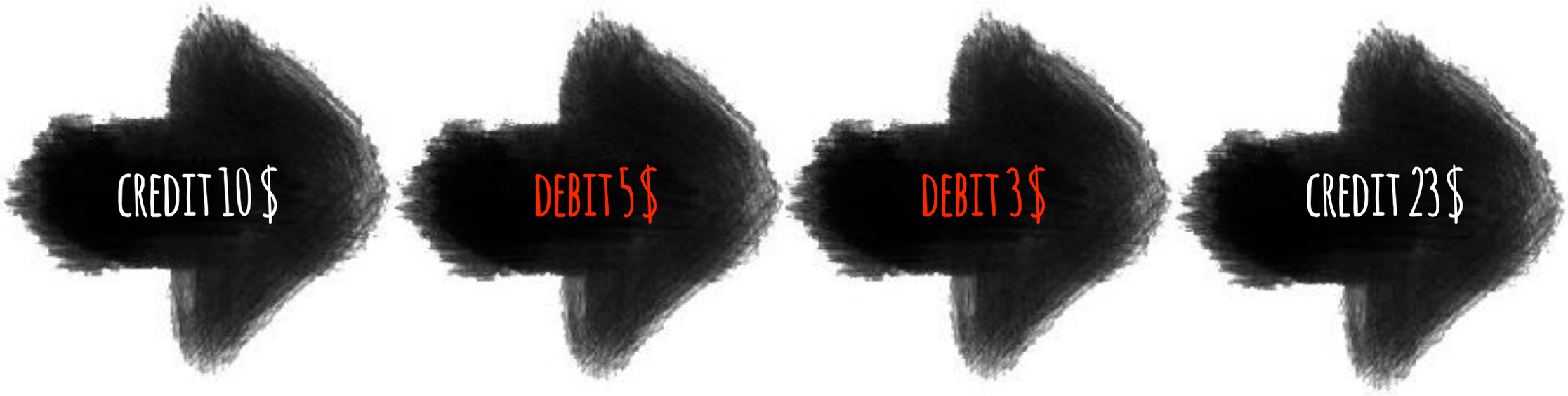
Create Basket

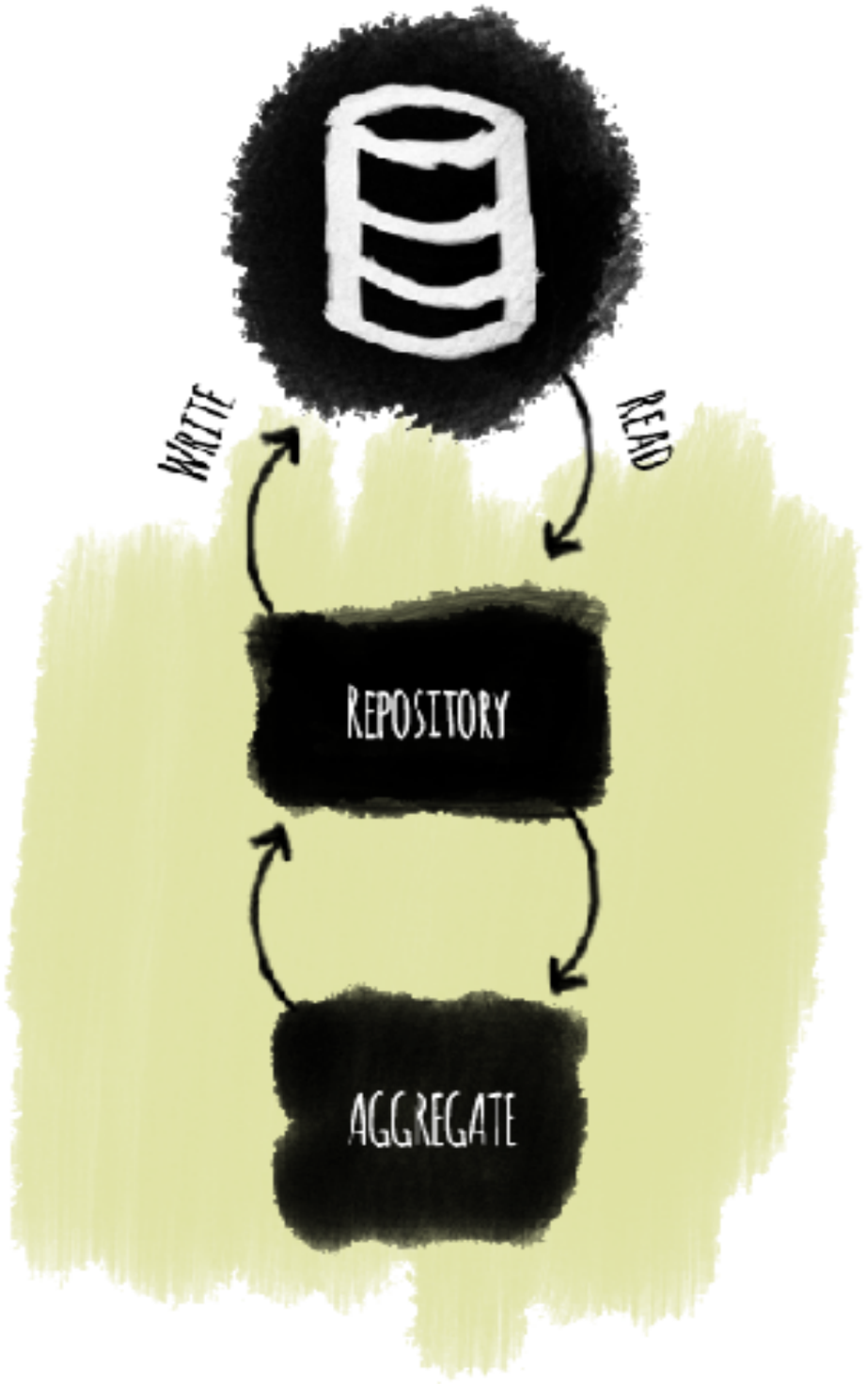
Add Item to Basket

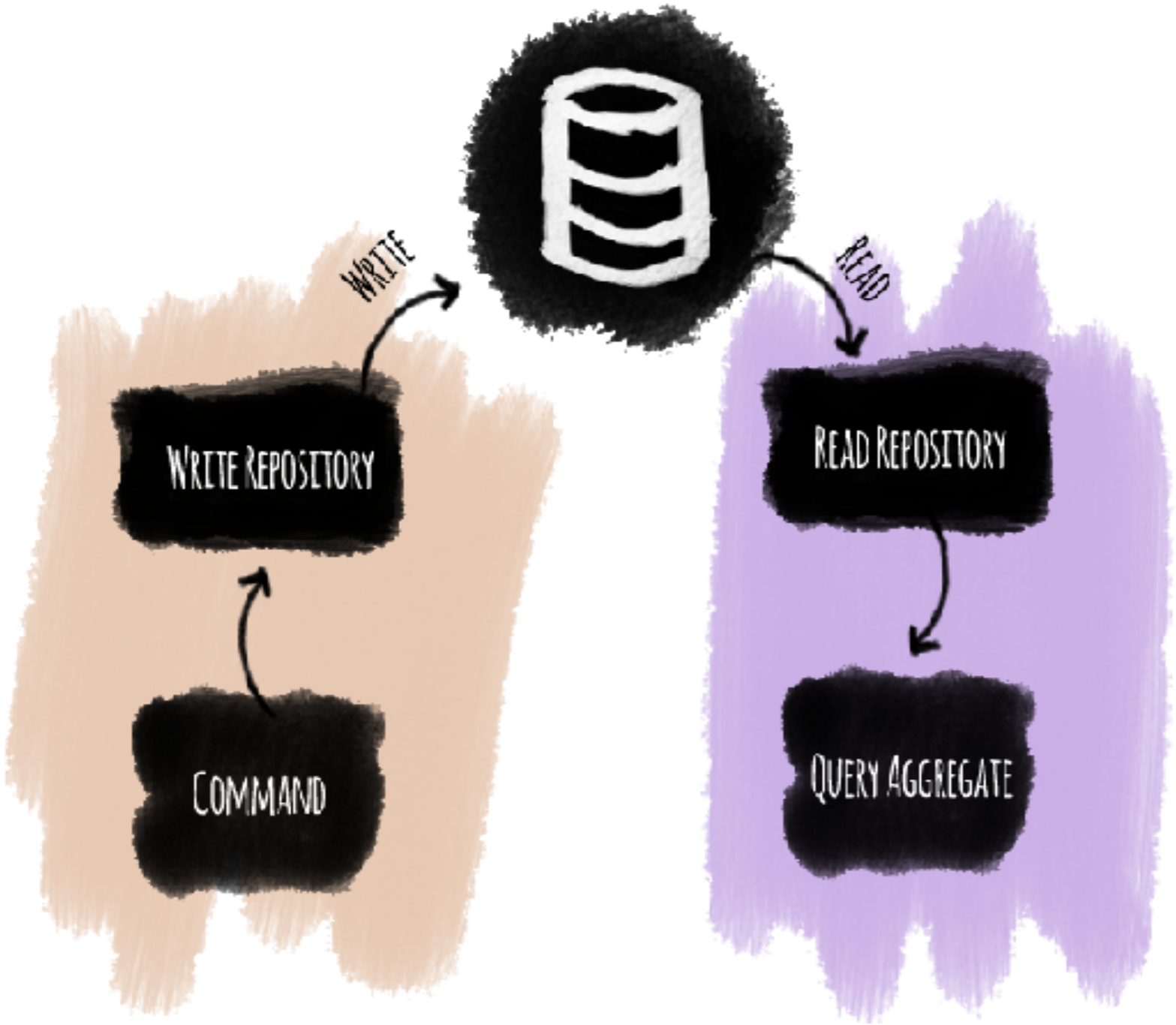
ADD SHIPPING ADDR.

Pay

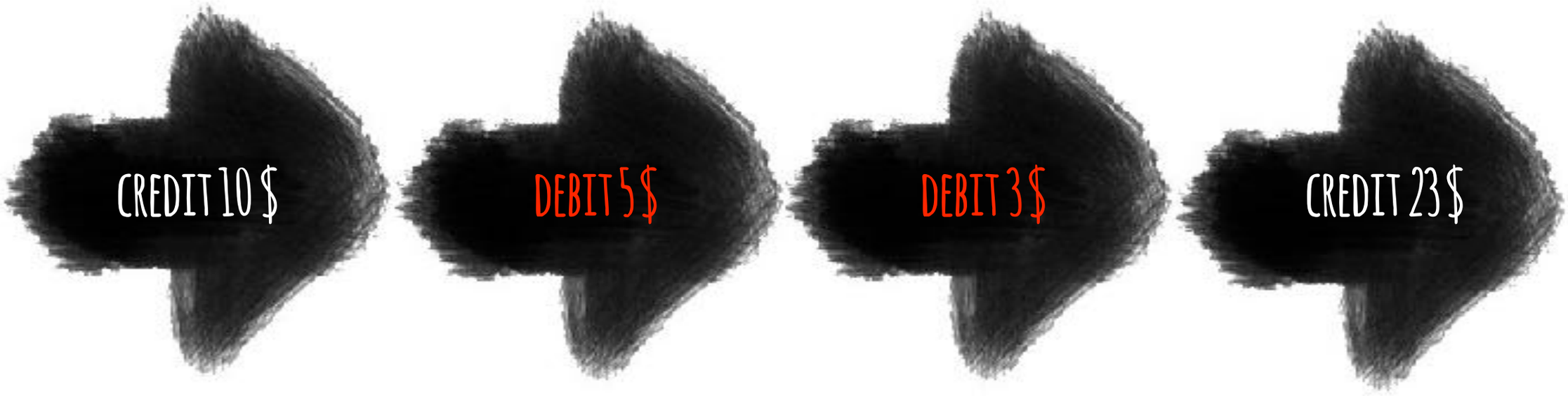
Bank Account




WRITE
READ
REPOSITORY
AGGREGATE


WRITE
READ
WRITE REPOSITORY
READ REPOSITORY
COMMAND
QUERY AGGREGATE

Bank Account



f(state, event) = state

```
class BankAccount(val balance: Int = 0);
```

```
abstract class Event;
case class CreditEvent(sum: Int) extends Event;
case class DebitEvent(sum: Int) extends Event;
```

```
def credit(b: BankAccount, e: CreditEvent)
  = new BankAccount(b.balance + e.sum);


def debit(b: BankAccount, e: DebitEvent)
  = new BankAccount(b.balance - e.sum);
```


def pick(e: Event): (BankAccount) => BankAccount = e match {
case c: CreditEvent => (b:BankAccount) => credit(b, c);
case d: DebitEvent => (b:BankAccount) => debit(b, d);
}


val events:List[Event] = List(CreditEvent(10), DebitEvent(3), …);
val account = new BankAccount(30);
val balance = events.foldLeft(account)(b,e) => pick(e).apply(b);

More?

Functional and Reactive Domain Modeling

Debasish Ghosh

Implementing Domain Driven Design

Vaughn Vernon

Domain Driven Design Distilled

Vaughn Vernon