

# AGENDA

- What is Tactical Design?
- Aggregates
- Entities vs. Value Objects
- Aggregates Revisited
- Domain Events
- Domain Services
- Application Services

# TACTICAL DESIGN?

Now it's time to start speaking the  
Ubiquitous Language within each Bounded Context.



- Business Invariants
- Policies
- Transactions
- State
- Persistence

# THE CENTRAL CONCEPT

**AGGREGATE**

# THE CENTRAL CONCEPT

A cluster of objects  
treated as a single unit.



AGGREGATE

# THE CENTRAL CONCEPT

A cluster of objects  
treated as a single unit.

**AGGREGATE**

Only accessed through its  
Root Entity.

# THE CENTRAL CONCEPT

A cluster of objects  
treated as a single unit.

**AGGREGATE**

Only accessed through its  
Root Entity.

Responsible for maintaining  
any/all business invariants.

# THE CENTRAL CONCEPT

A cluster of objects  
treated as a single unit.

**AGGREGATE**

Only accessed through its  
Root Entity.

The atomic unit for any  
transactional  
behavior.

Responsible for maintaining  
any/all business invariants.

# THE CENTRAL CONCEPT

A cluster of objects  
treated as a single unit.

Only accessed through its  
Root Entity.

Often modeled as a  
state machine.

The atomic unit for any  
transactional  
behavior.

**AGGREGATE**

Responsible for maintaining  
any/all business invariants.

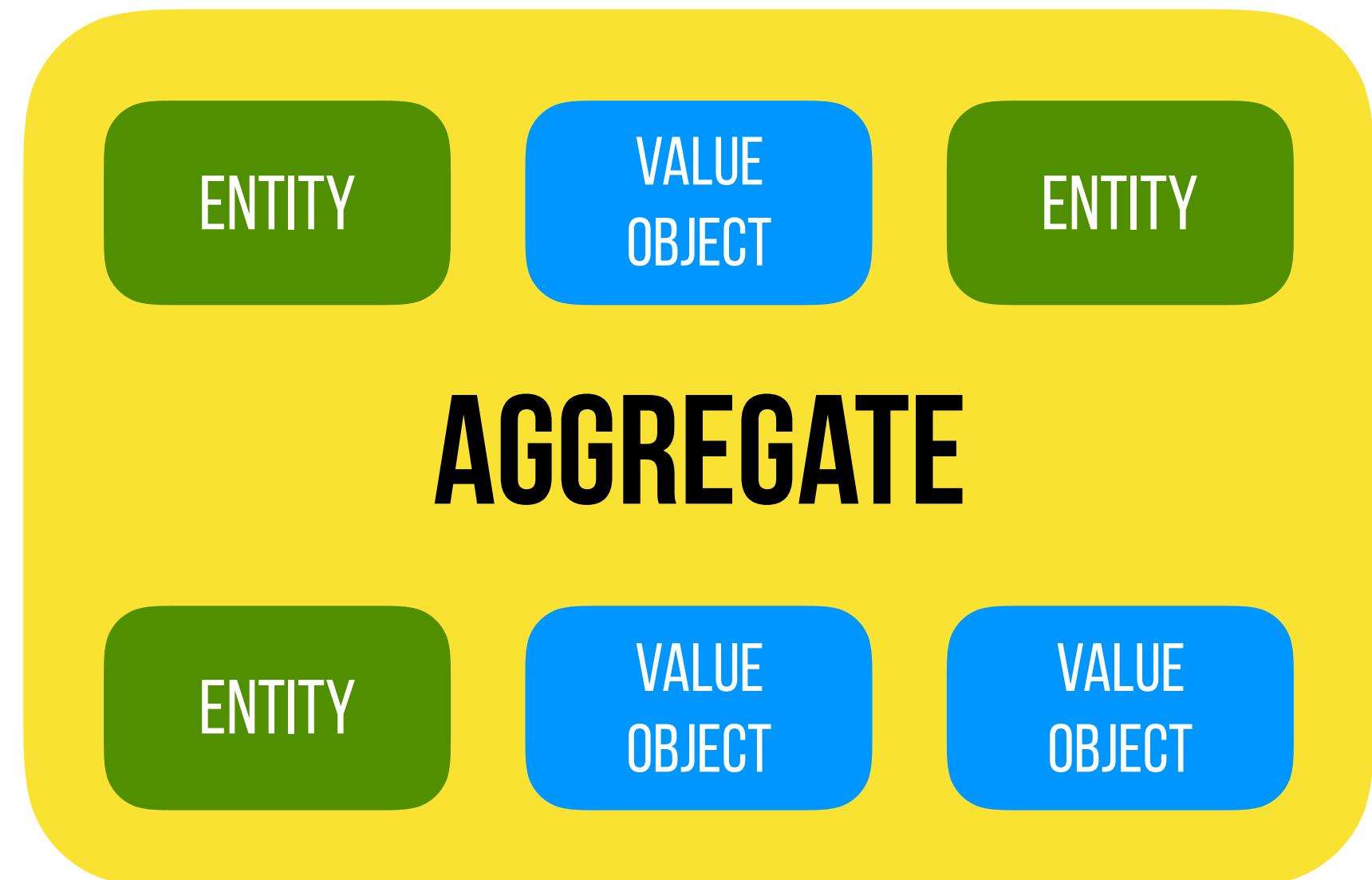
# THE CENTRAL CONCEPT

A cluster of objects  
treated as a single unit.

Only accessed through its  
Root Entity.

Often modeled as a  
state machine.

The atomic unit for any  
transactional  
behavior.



Responsible for maintaining  
any/all business invariants.

**ENTITY**

**An object whose state changes over time,  
but whose identity remains the same.**

ENTITY

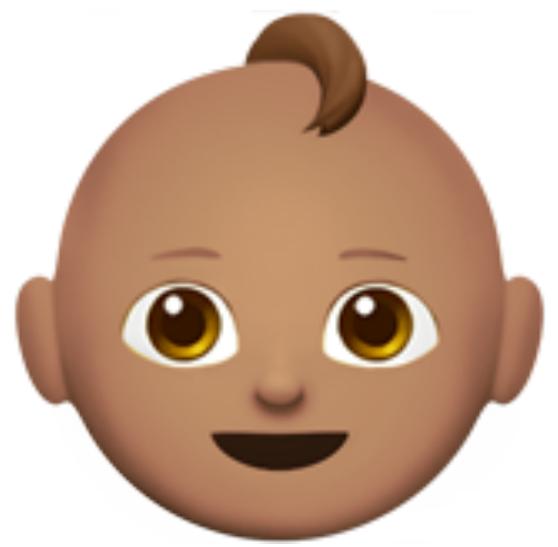
An object whose state changes over time,  
but whose identity remains the same.



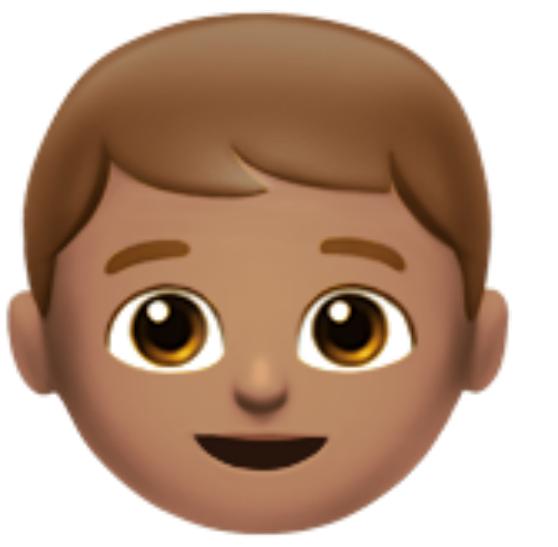
```
{  
  id: 1,  
  name: "Joe",  
  age: 0  
}
```

# ENTITY

An object whose state changes over time,  
but whose identity remains the same.



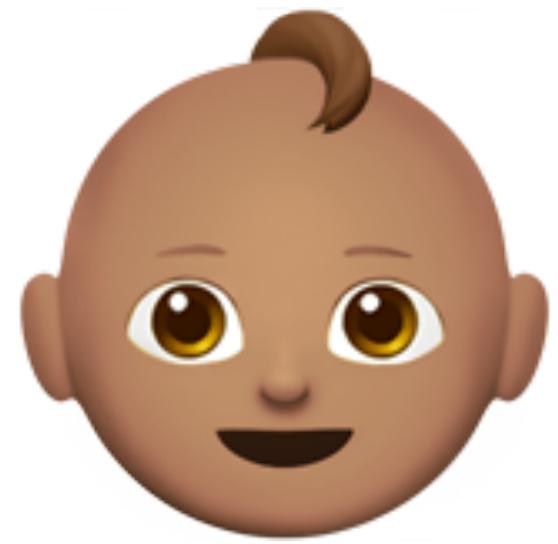
```
{  
  id: 1,  
  name: "Joe",  
  age: 0  
}
```



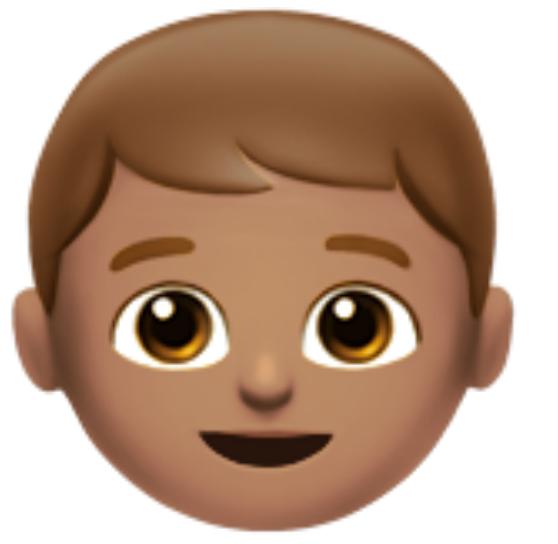
```
{  
  id: 1,  
  name: "Joe",  
  age: 8  
}
```

# ENTITY

An object whose state changes over time,  
but whose identity remains the same.



```
{  
  id: 1,  
  name: "Joe",  
  age: 0  
}
```



```
{  
  id: 1,  
  name: "Joe",  
  age: 8  
}
```



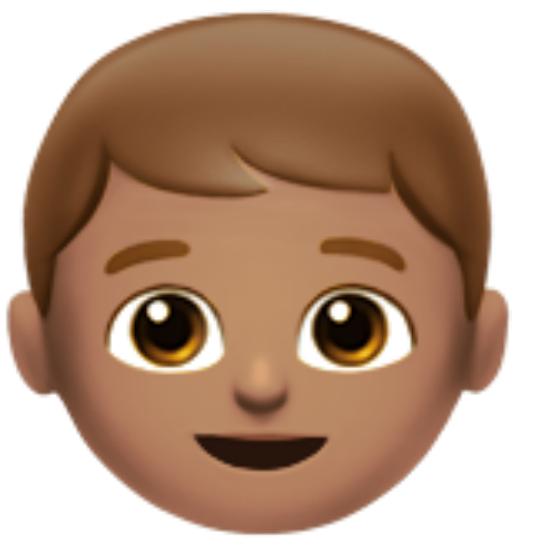
```
{  
  id: 1,  
  name: "Joe",  
  age: 35  
}
```

# ENTITY

An object whose state changes over time,  
but whose identity remains the same.



```
{  
  id: 1,  
  name: "Joe",  
  age: 0  
}
```



```
{  
  id: 1,  
  name: "Joe",  
  age: 8  
}
```



```
{  
  id: 1,  
  name: "Joe",  
  age: 35  
}
```



```
{  
  id: 1,  
  name: "Joe",  
  age: 60  
}
```

## VALUE OBJECT

**An object whose identity is inseparable from its value.**

# VALUE OBJECT

An object whose identity is inseparable from its value.



```
{  
  id: 1,  
  name: "Joe",  
  age: 35,  
  address: <PTR>  
}
```

## VALUE OBJECT



```
{  
  street: "123 Vine St.",  
  city: "Memphis",  
  state: "TN"  
}
```

An object whose identity is inseparable from its value.



```
{  
  id: 1,  
  name: "Joe",  
  age: 35,  
  address: <PTR>  
}
```

## VALUE OBJECT

An object whose identity is inseparable from its value.



```
{  
  street: "123 Vine St.",  
  city: "Memphis",  
  state: "TN"  
}
```

```
{  
  id: 1,  
  name: "Joe",  
  age: 35,  
  address: <PTR>  
}
```

## VALUE OBJECT

An object whose identity is inseparable from its value.



```
{  
  street: "123 Vine St.",  
  city: "Memphis",  
  state: "TN"  
}
```

```
{  
  id: 1,  
  name: "Joe",  
  age: 35,  
  address: <PTR>  
}
```

```
{  
  street: "123 Long Rd.",  
  city: "Memphis",  
  state: "MI"  
}
```

# VALUE OBJECT

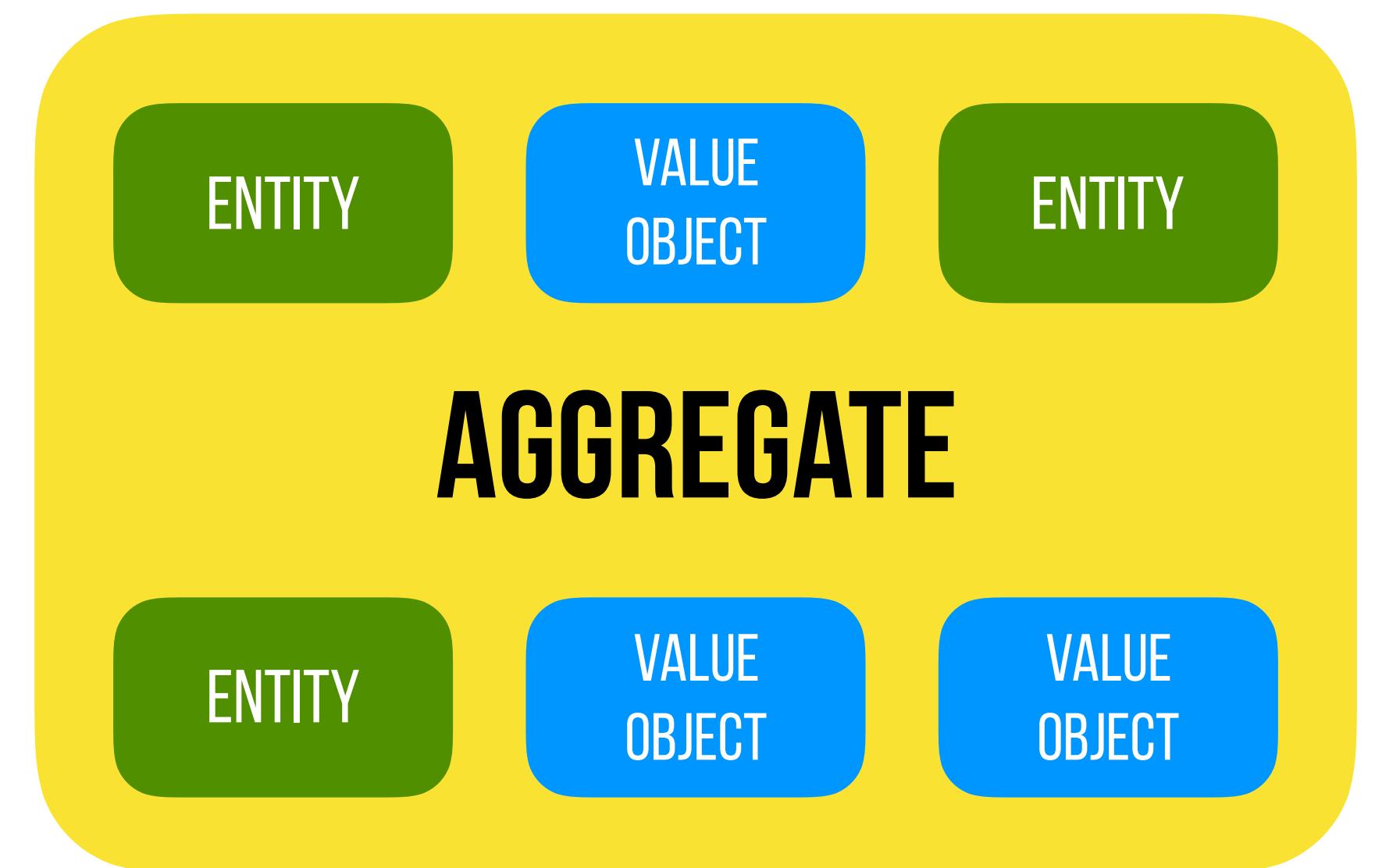
An object whose identity is inseparable from its value.

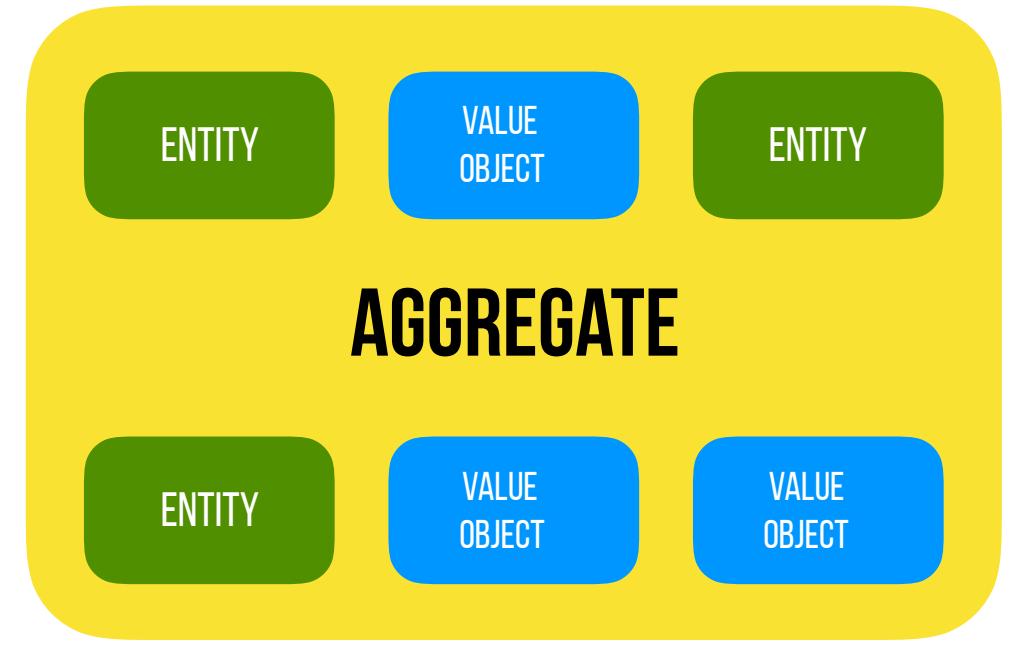


```
{  
  street: "123 Vine St.",  
  city: "Memphis",  
  state: "TN"  
}
```

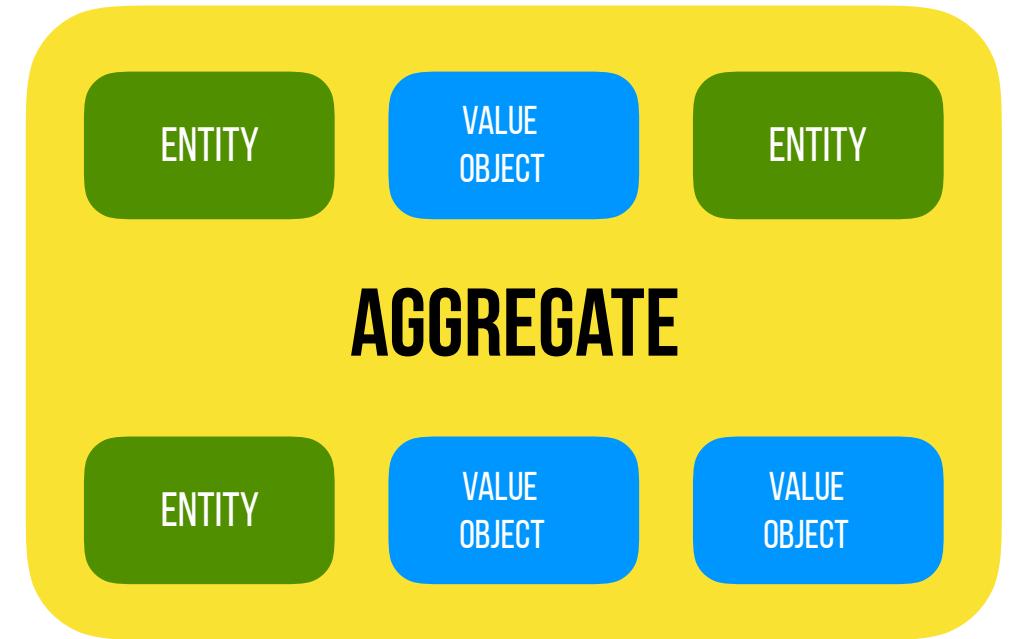
```
{  
  id: 1,  
  name: "Joe",  
  age: 35,  
  address: <PTR>  
}
```

```
{  
  street: "123 Long Rd.",  
  city: "Memphis",  
  state: "MI"  
}
```

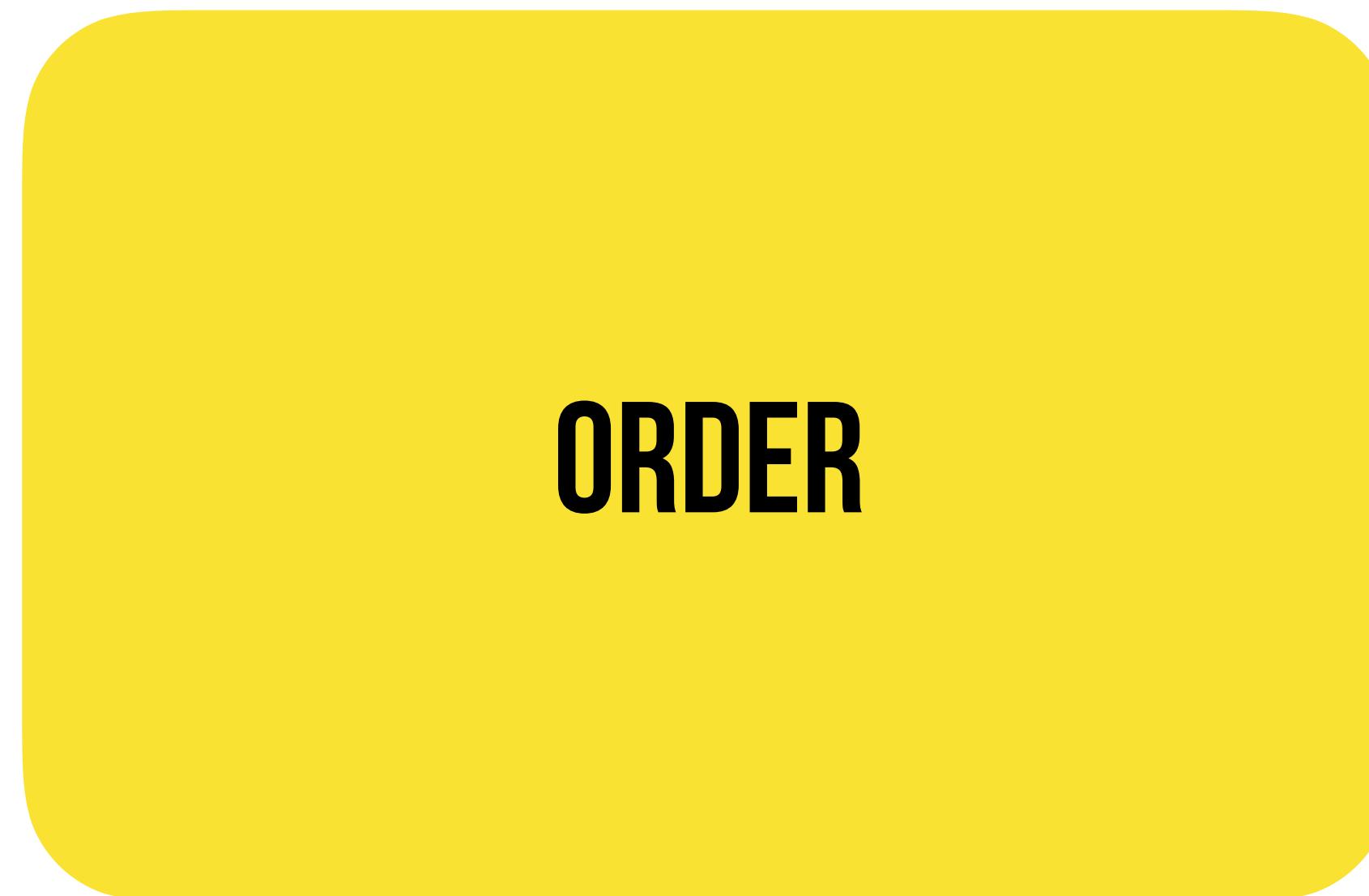




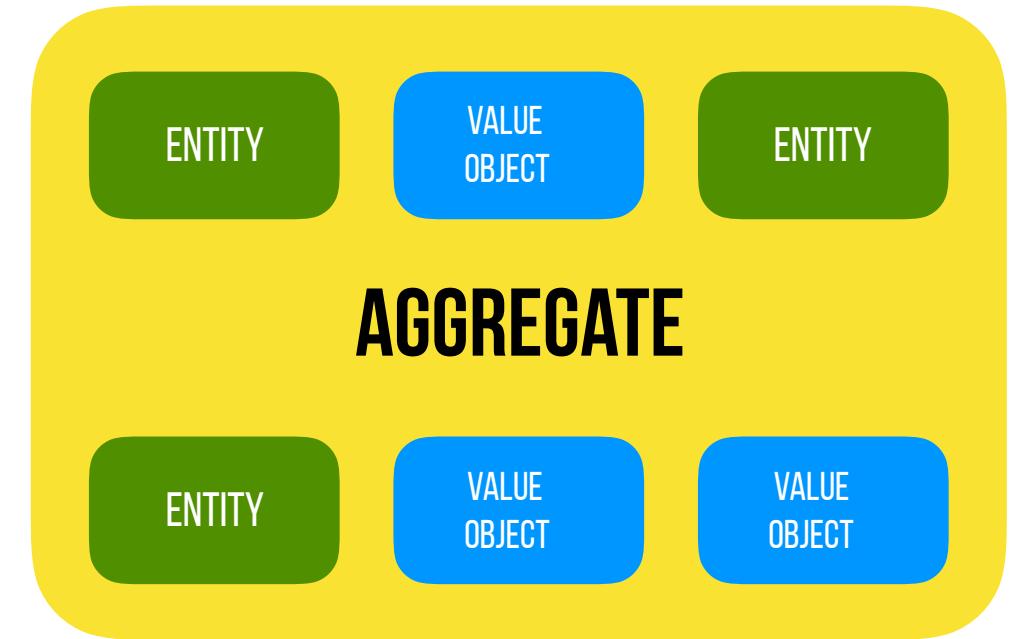
**Protect business invariants  
inside Aggregate boundaries.**



**Protect business invariants  
inside Aggregate boundaries.**



**Business Invariant: “Can only place an order if it contains at least one line item.”**

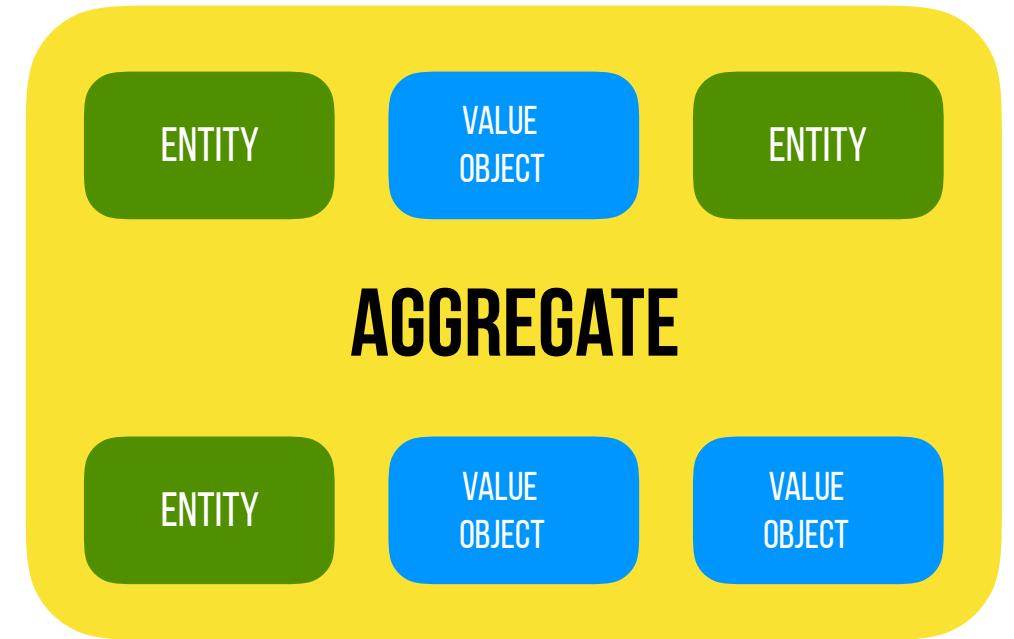


**Protect business invariants  
inside Aggregate boundaries.**

**PLACE  
ORDER**

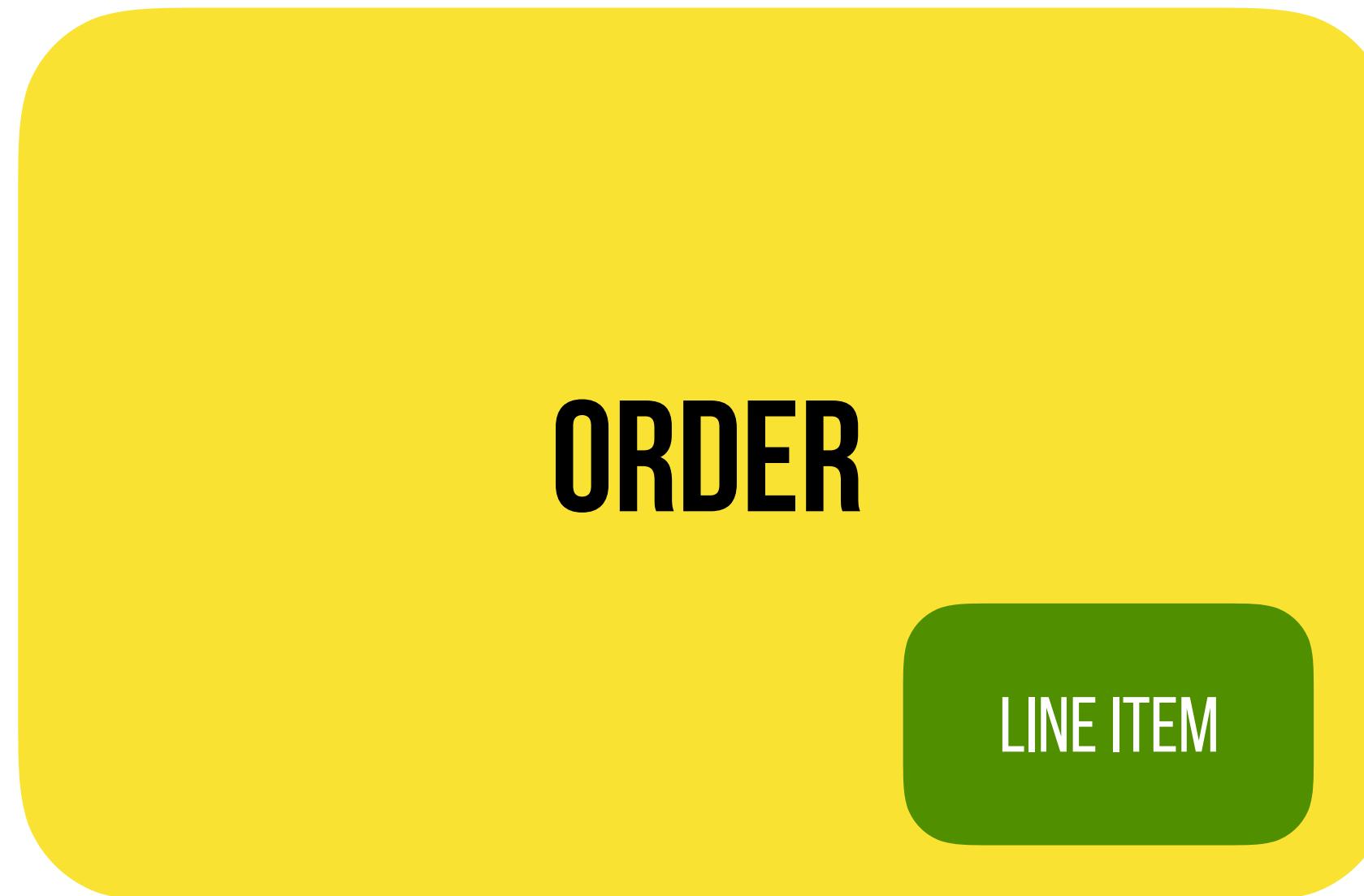
**ORDER**

**Business Invariant: “Can only place an order if it contains at least one line item.”**

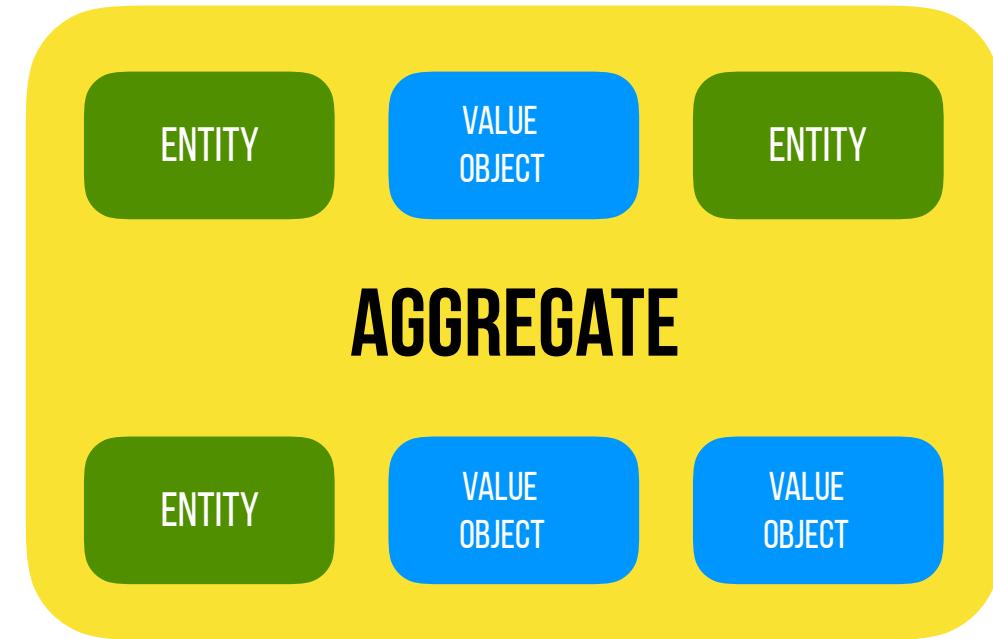


**Protect business invariants  
inside Aggregate boundaries.**

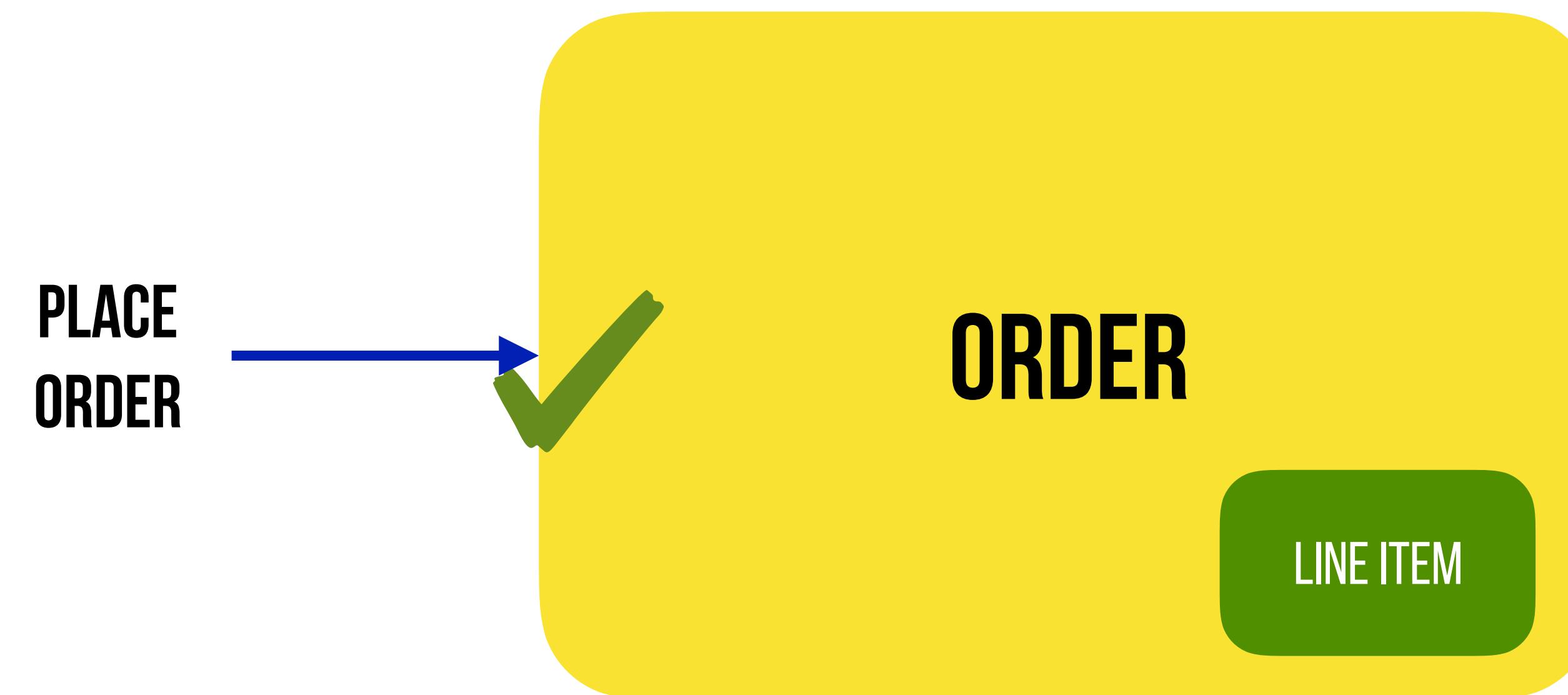
**PLACE  
ORDER**



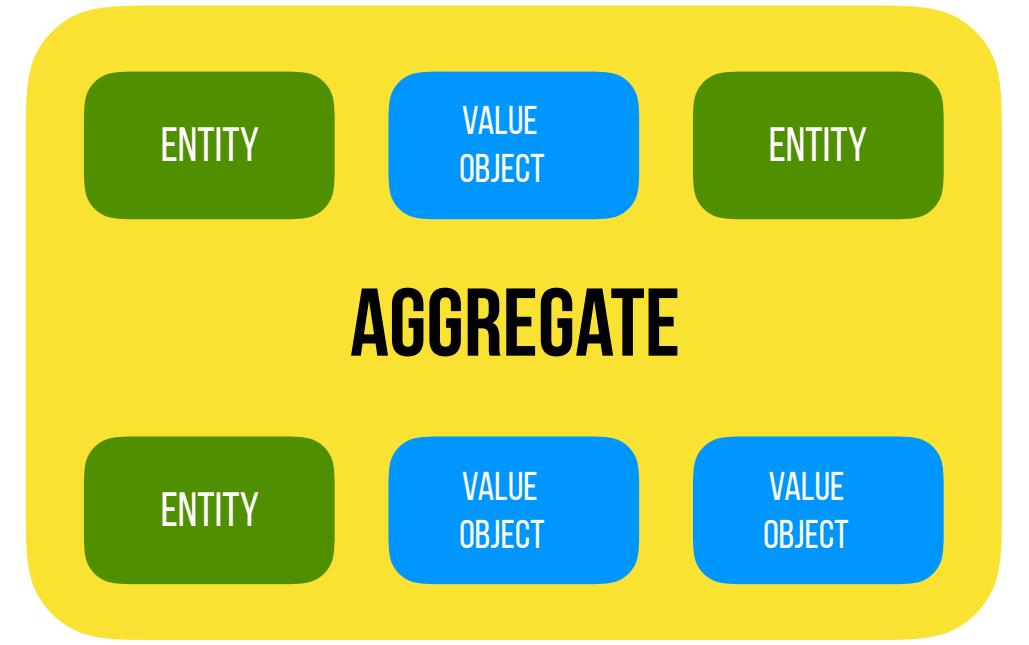
**Business Invariant: “Can only place an order if it contains at least one line item.”**



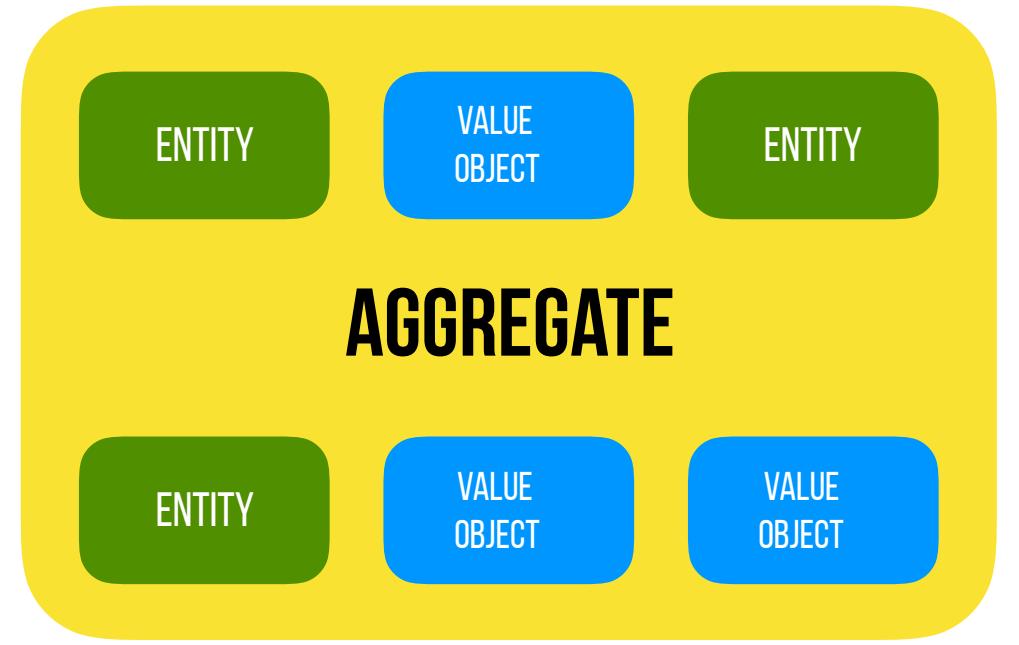
**Protect business invariants  
inside Aggregate boundaries.**



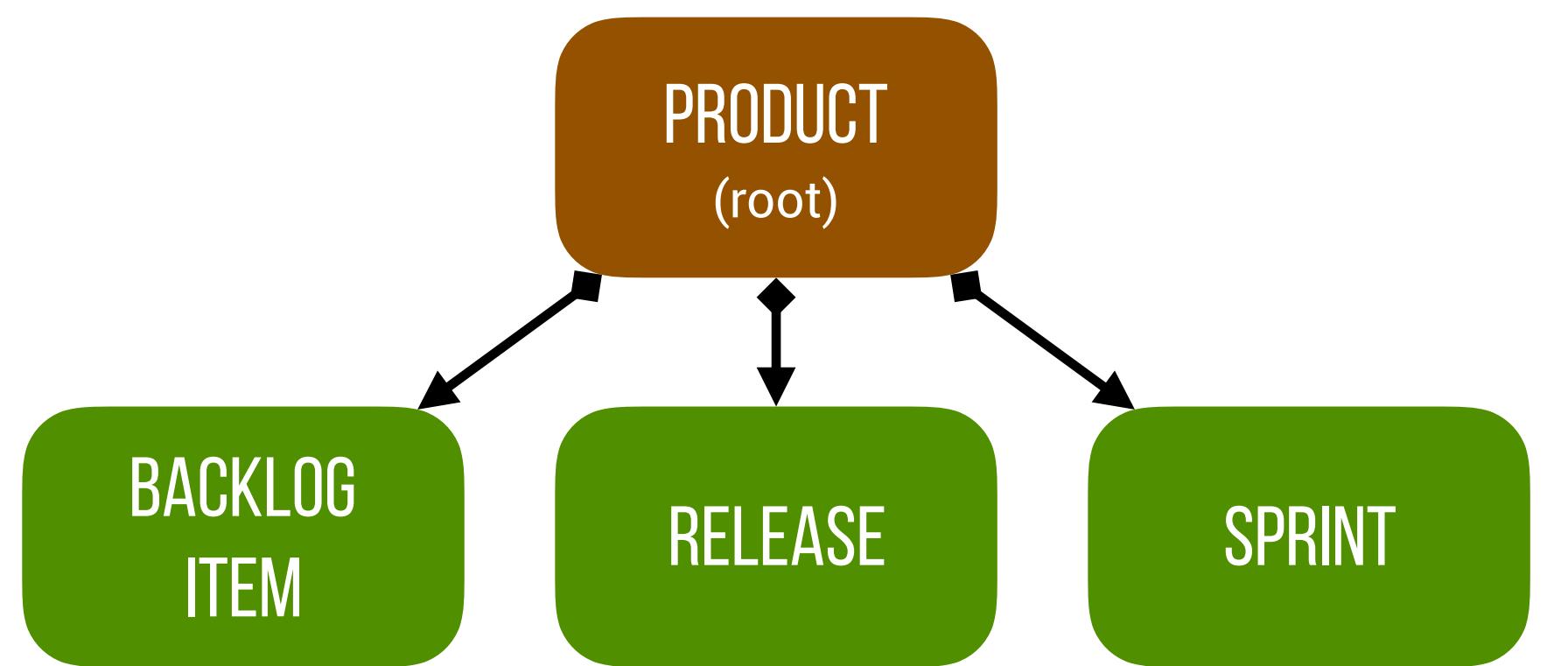
**Business Invariant: “Can only place an order if it contains at least one line item.”**



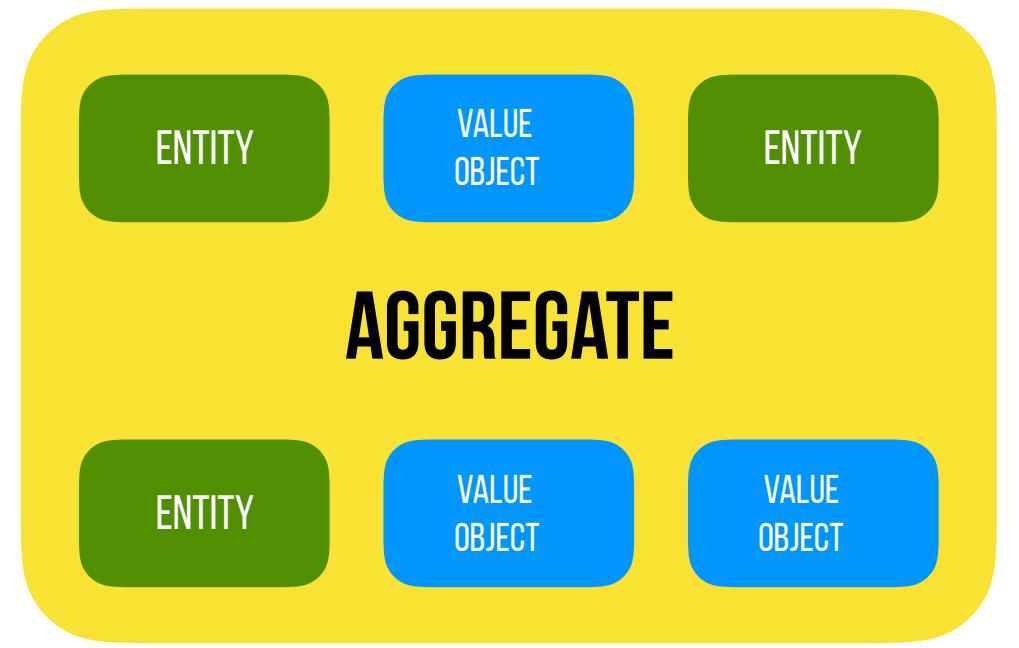
## Design small Aggregates.



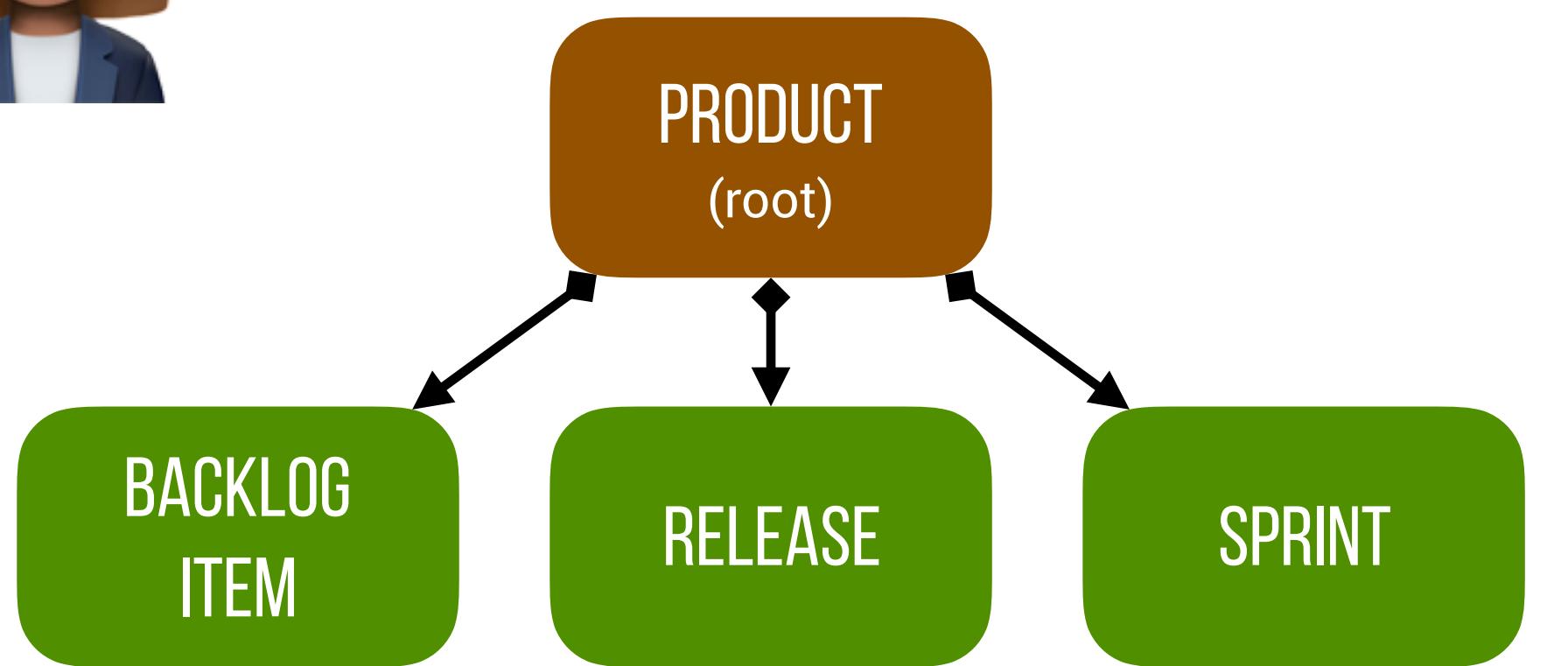
## Design small Aggregates.



**Large Cluster  
Aggregate**

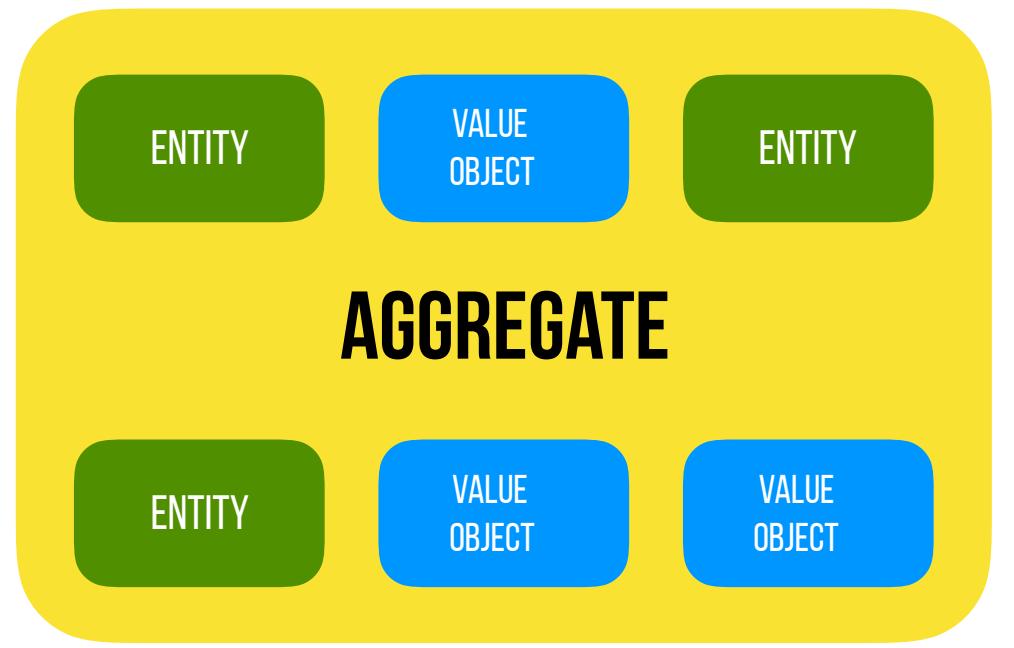


## Design small Aggregates.

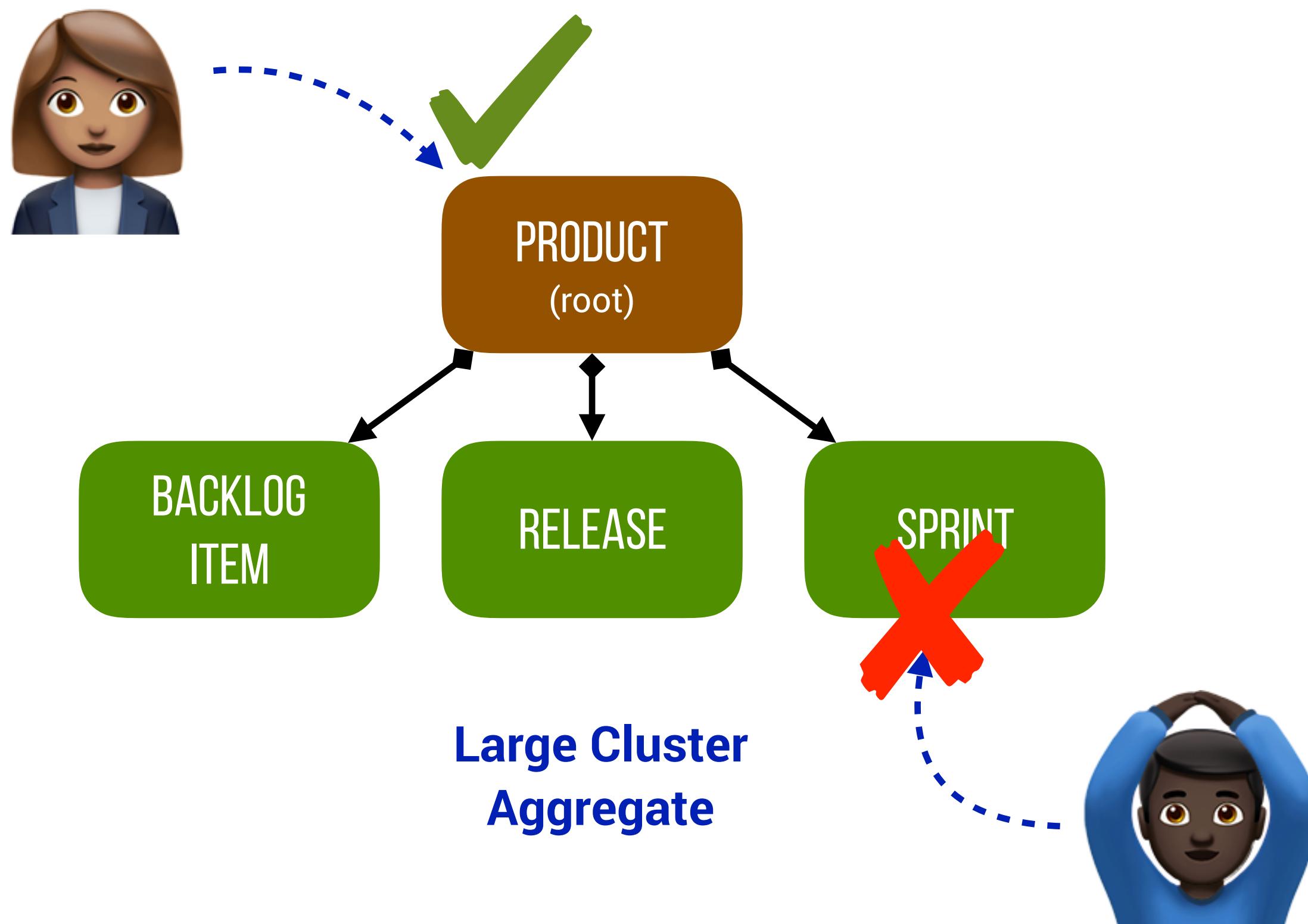


Large Cluster  
Aggregate

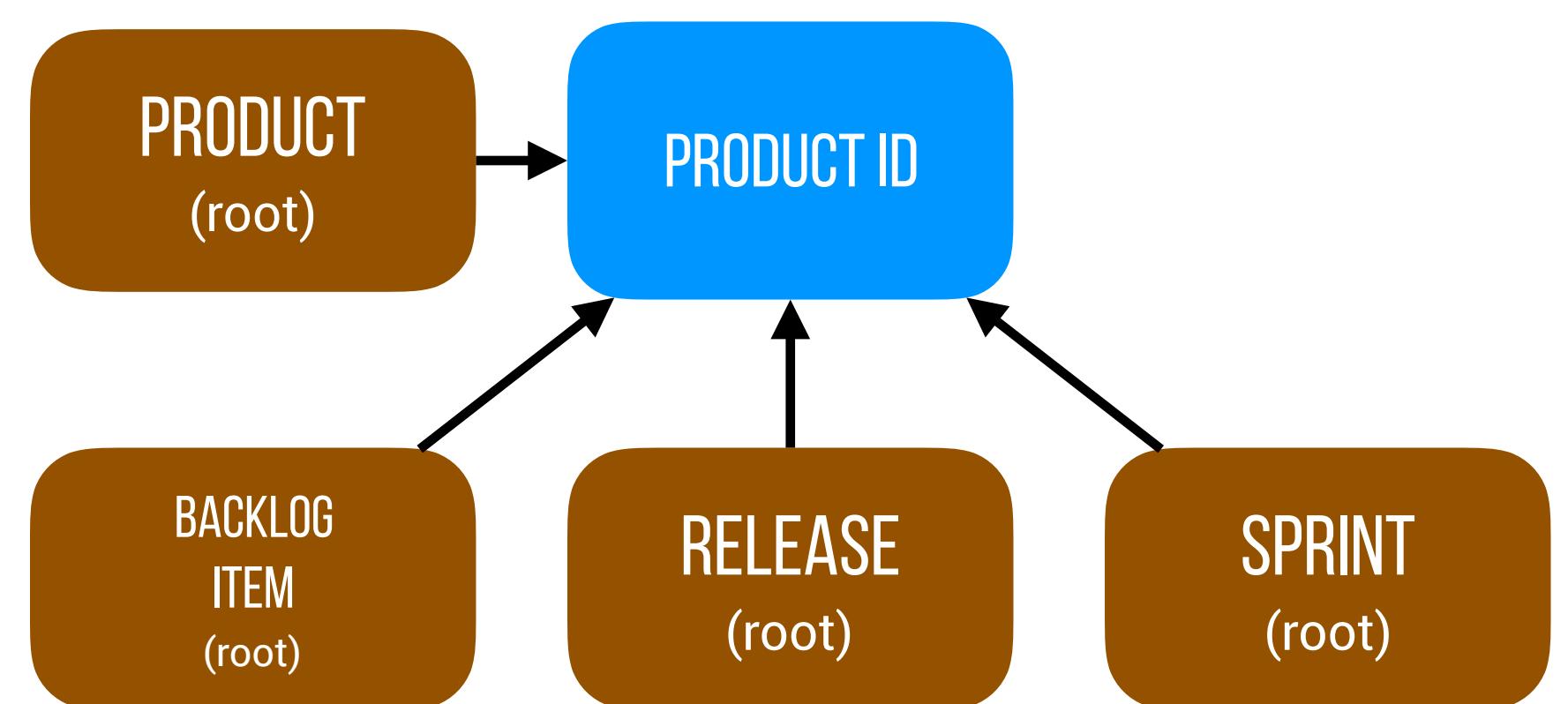
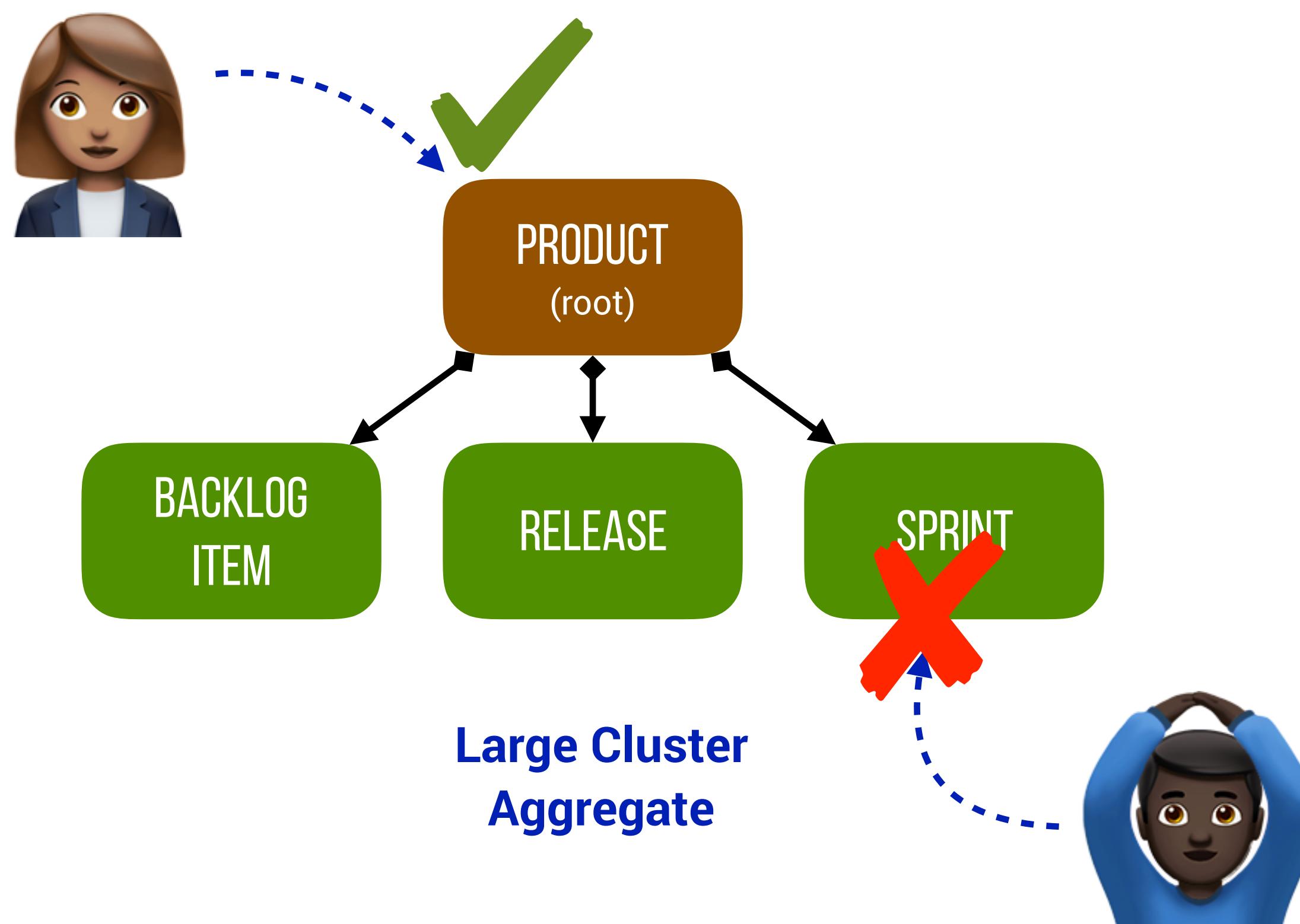
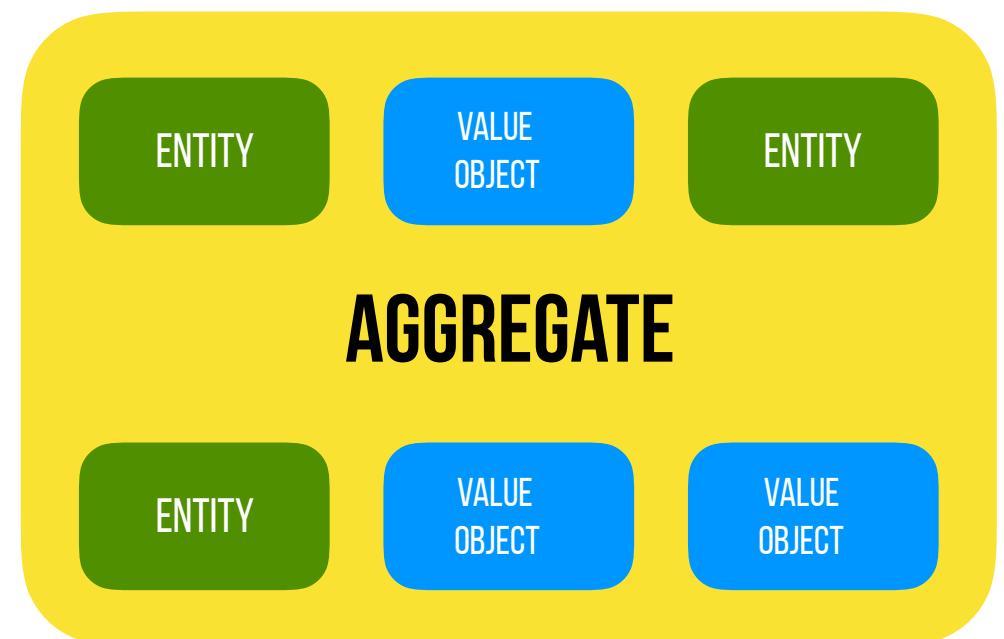




## Design small Aggregates.

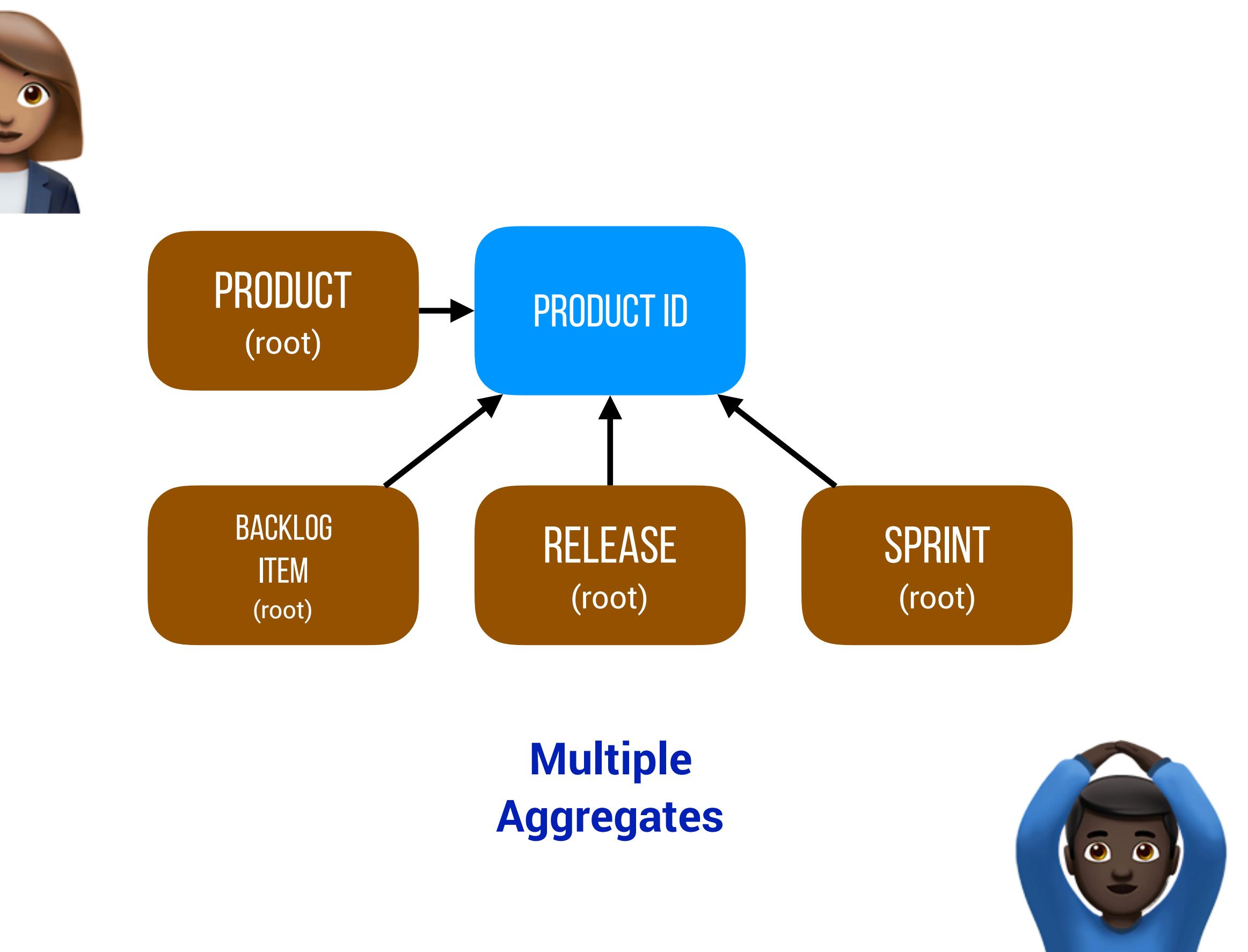
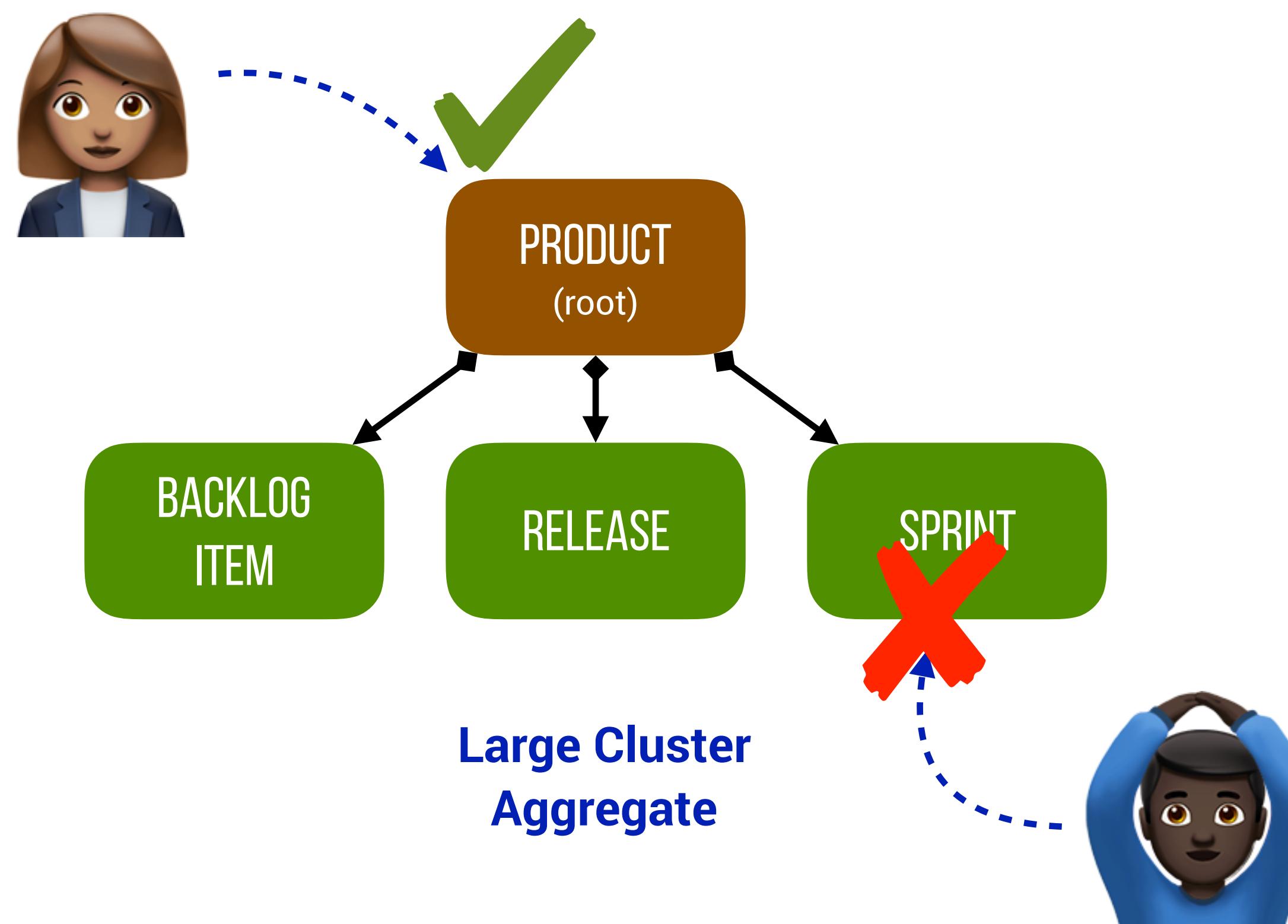
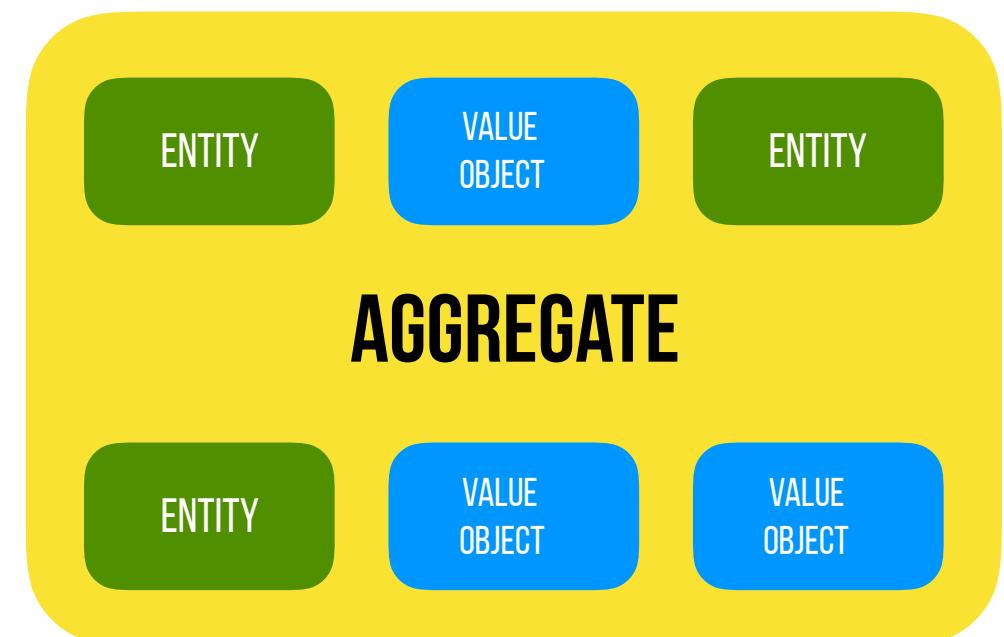


# Design small Aggregates.

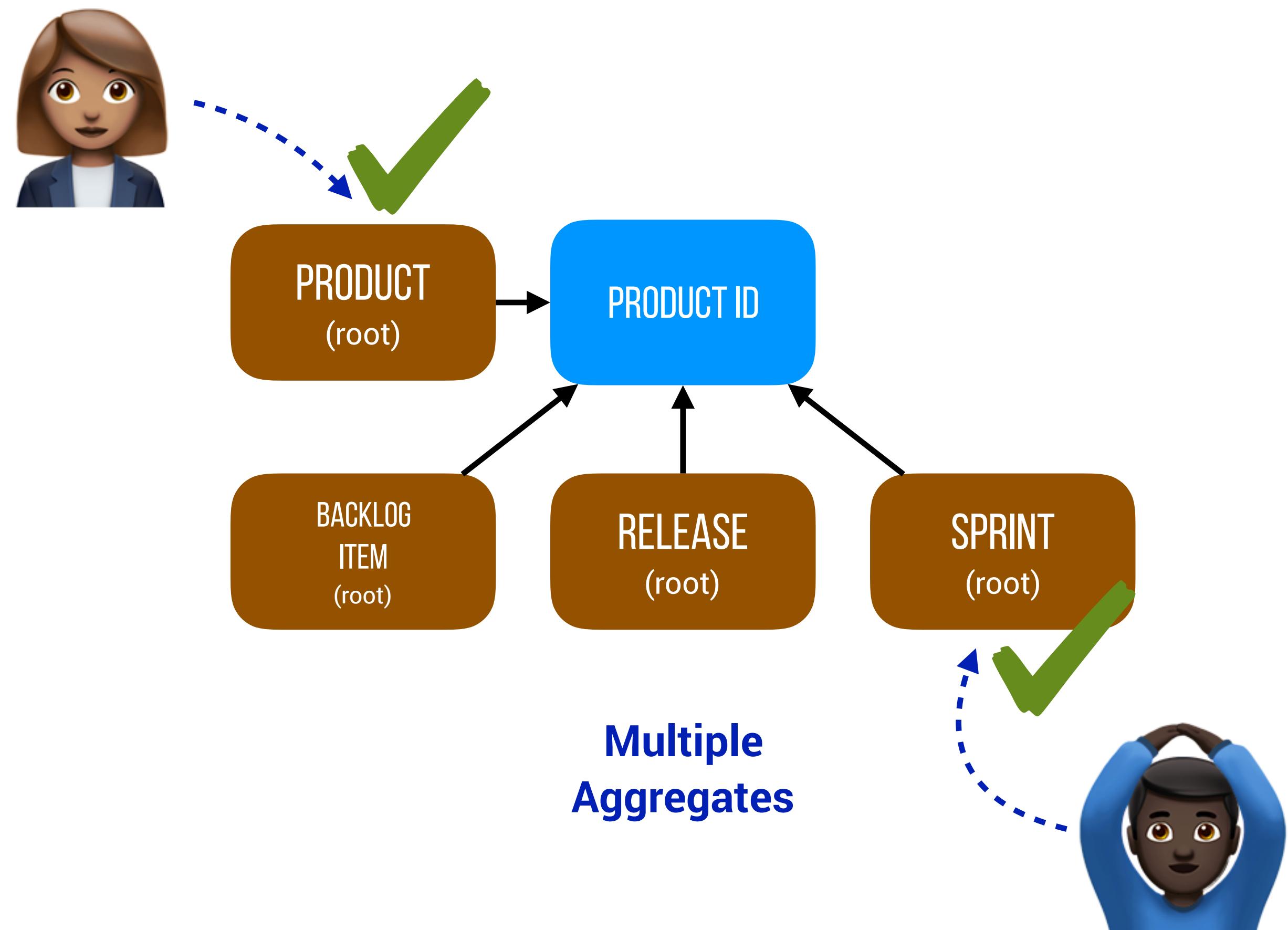
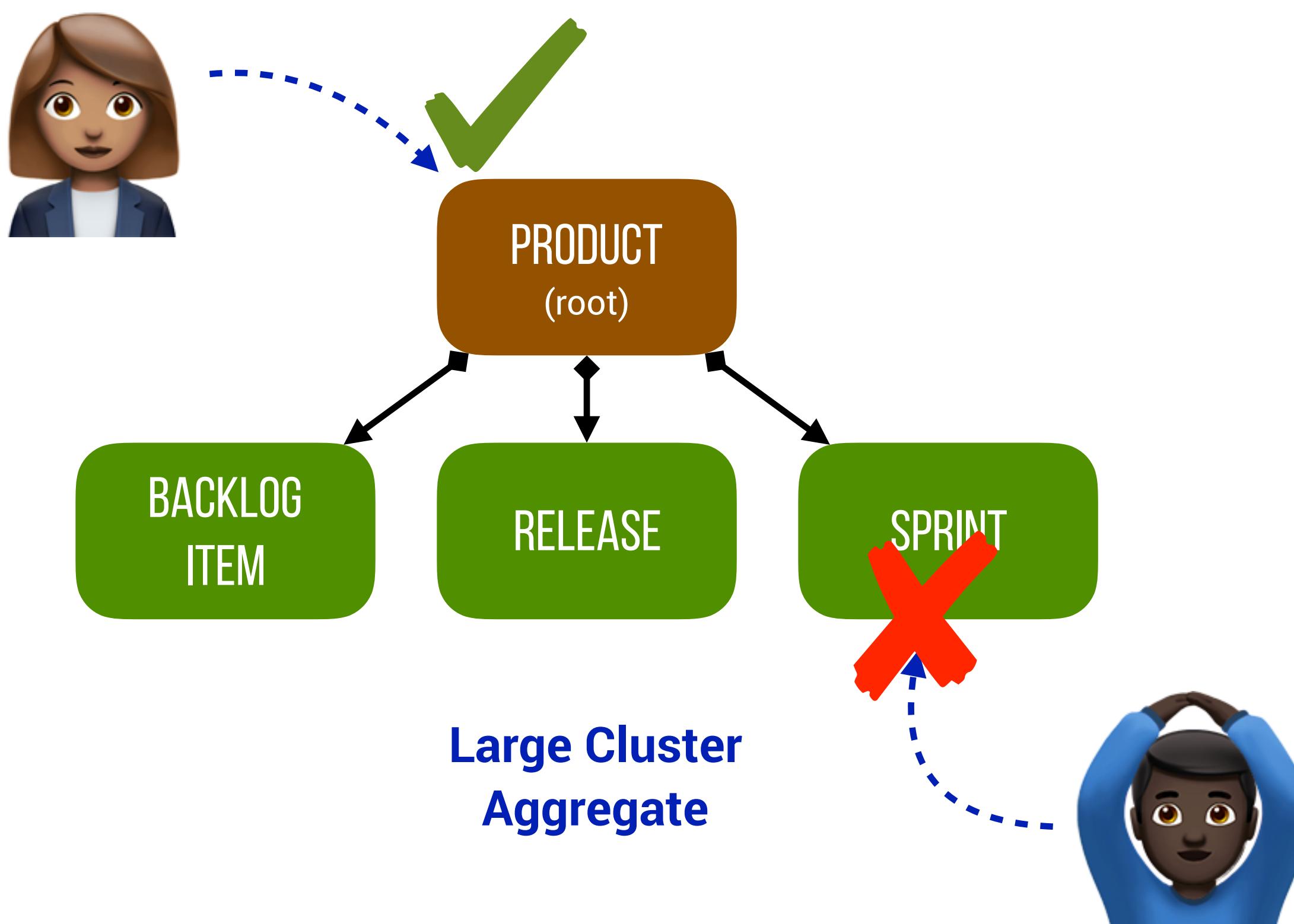
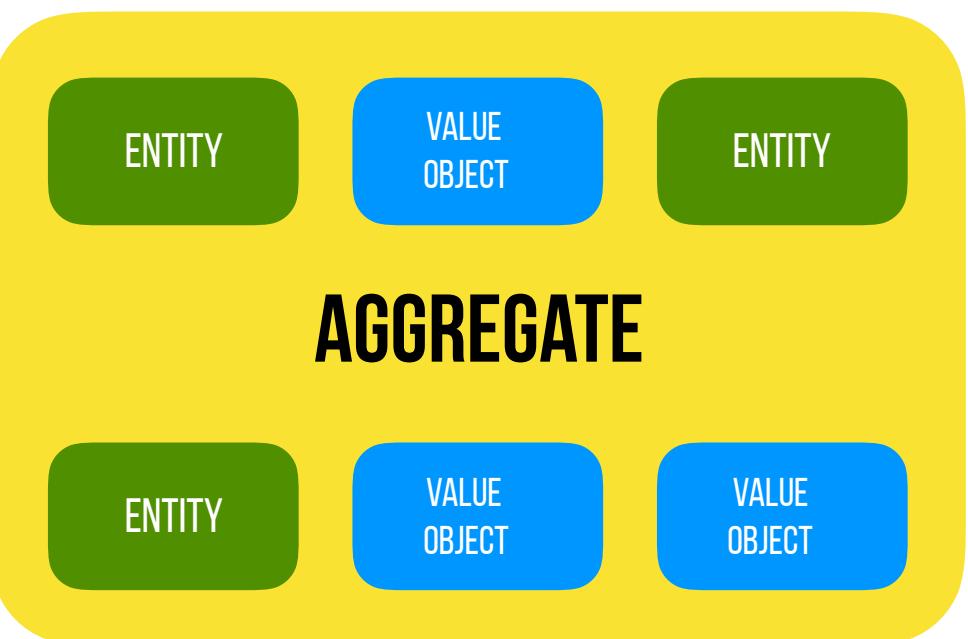


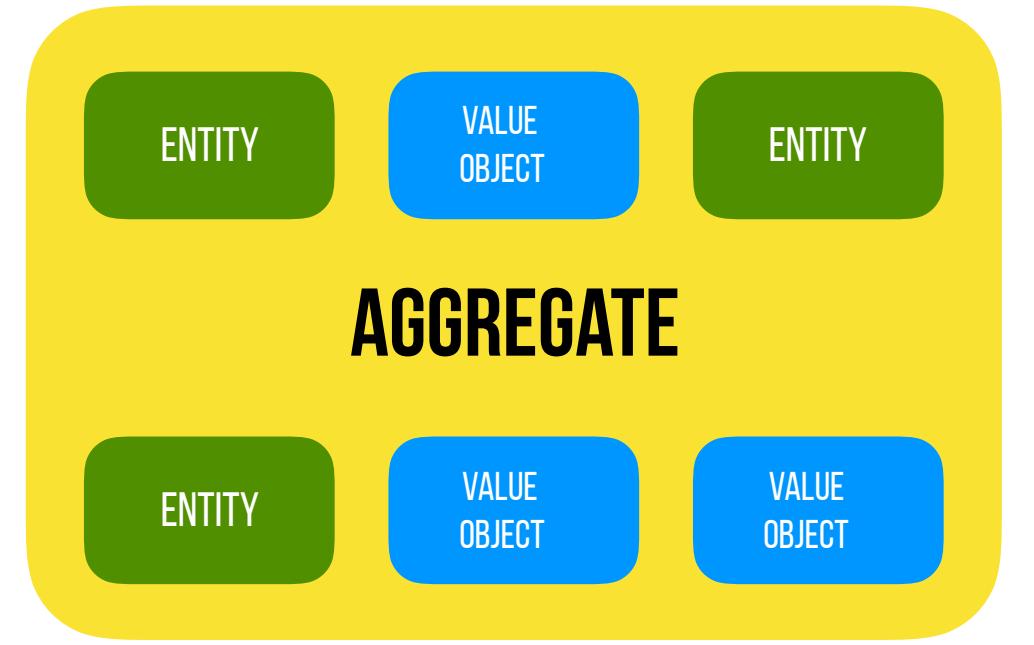
Multiple  
Aggregates

# Design small Aggregates.

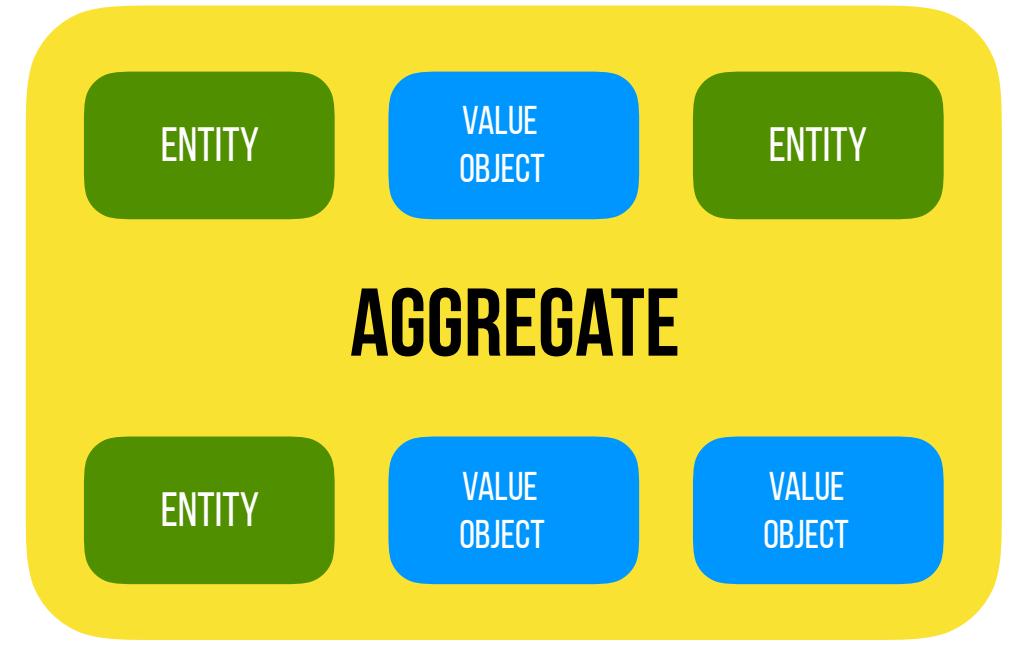


# Design small Aggregates.

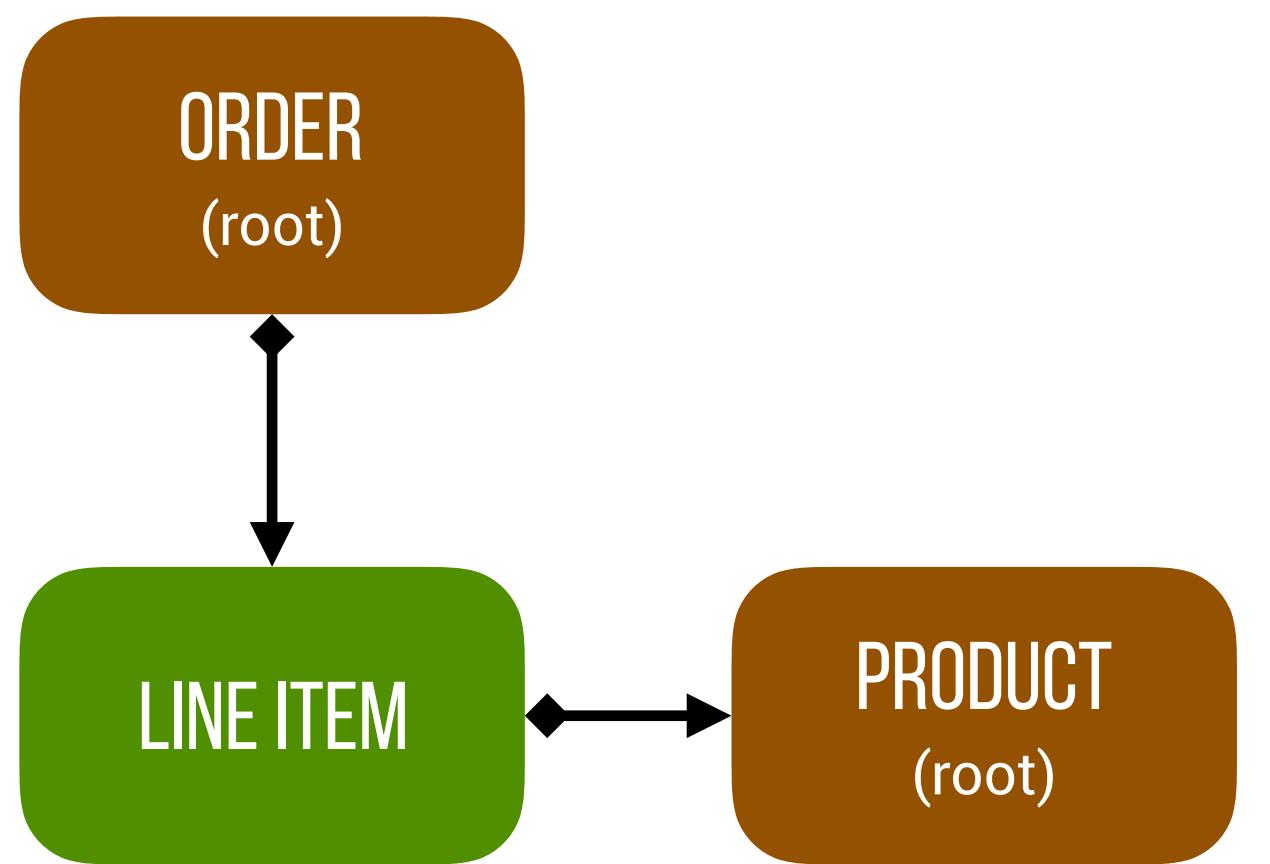




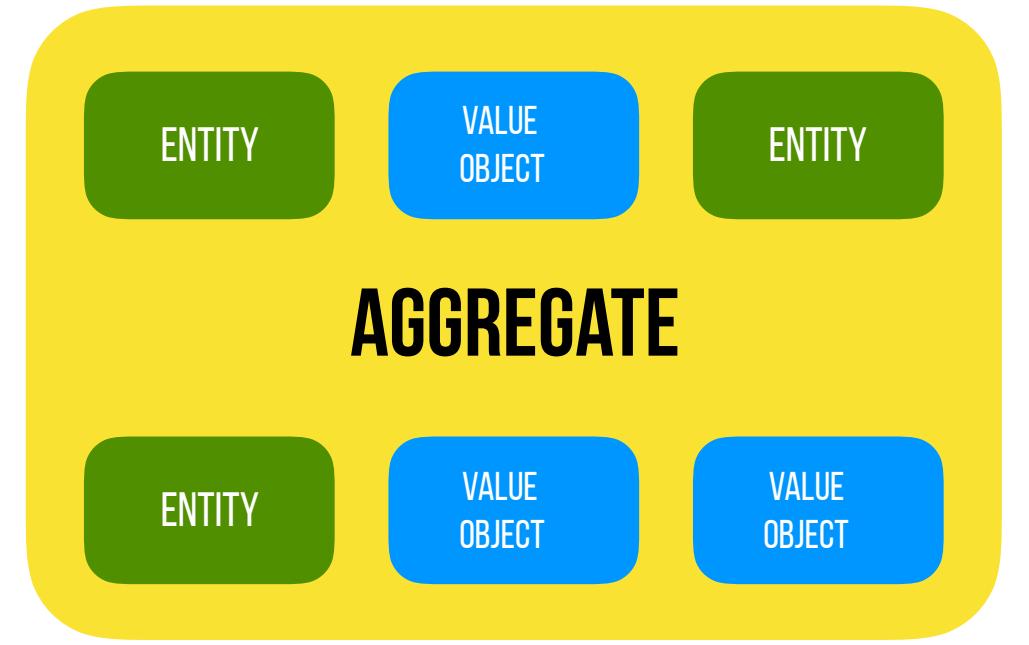
**Reference other Aggregates  
by Identity only.**



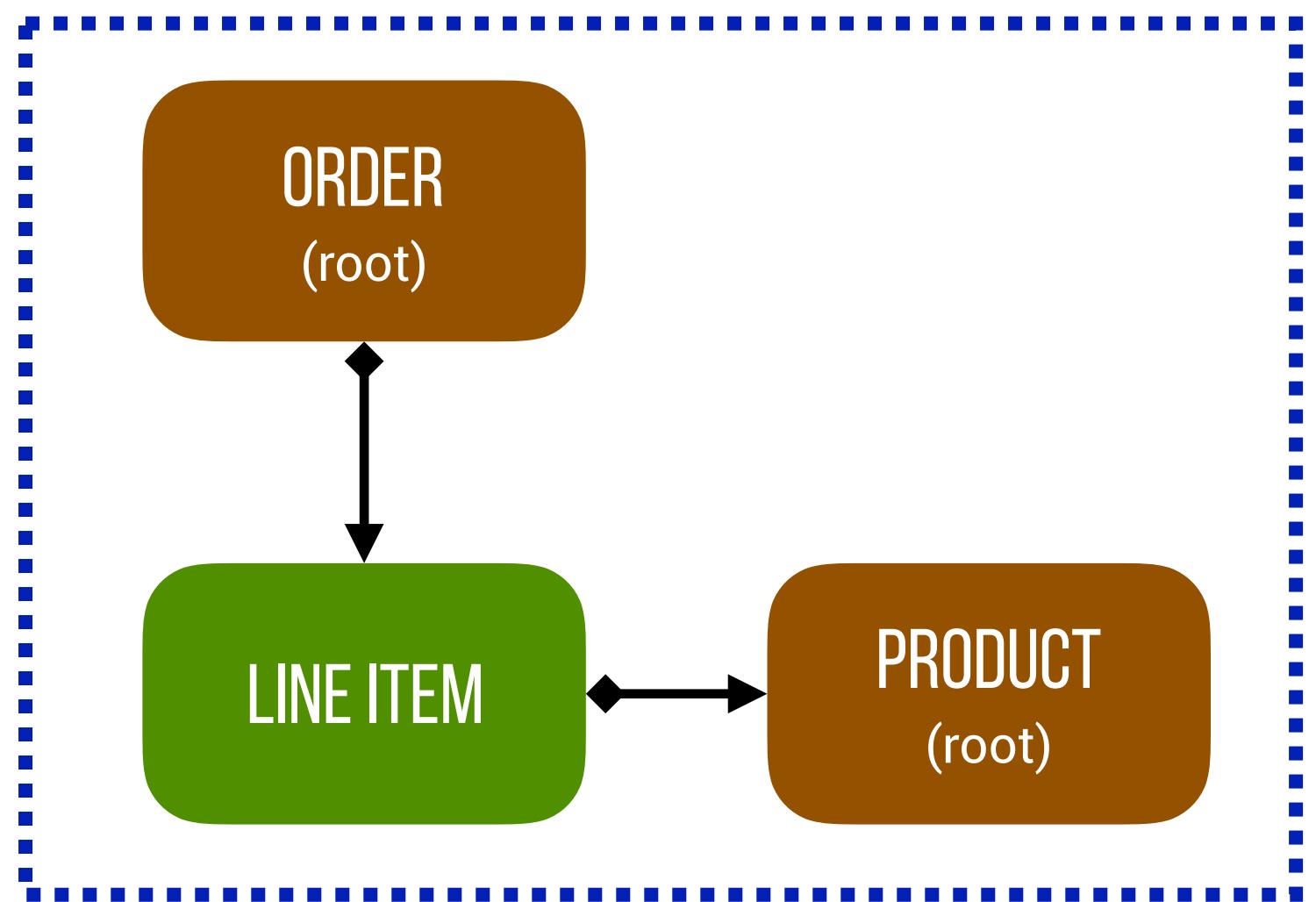
**Reference other Aggregates  
by Identity only.**



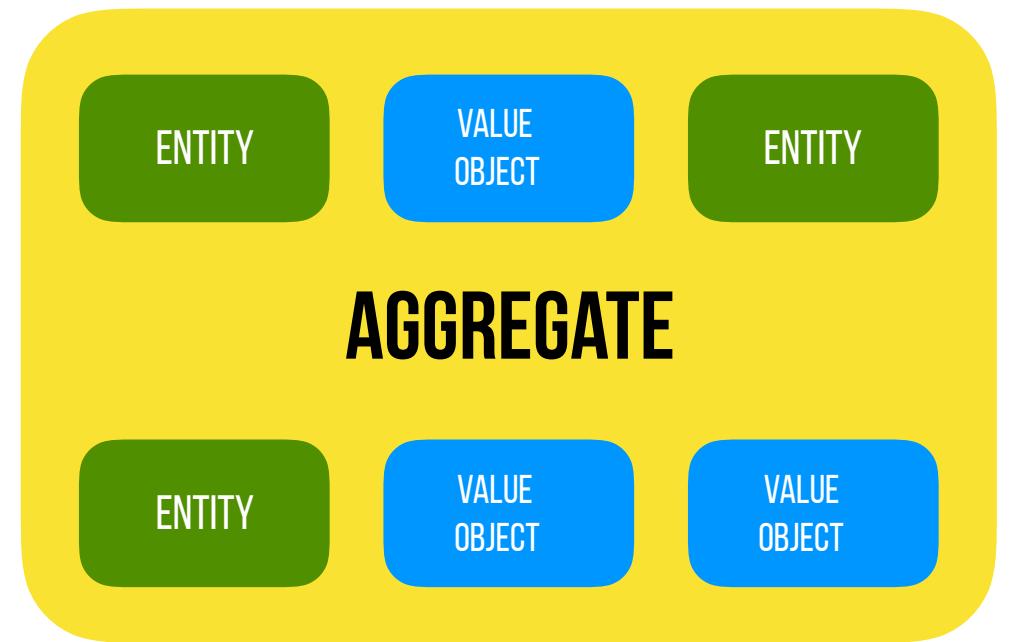
**Composition**



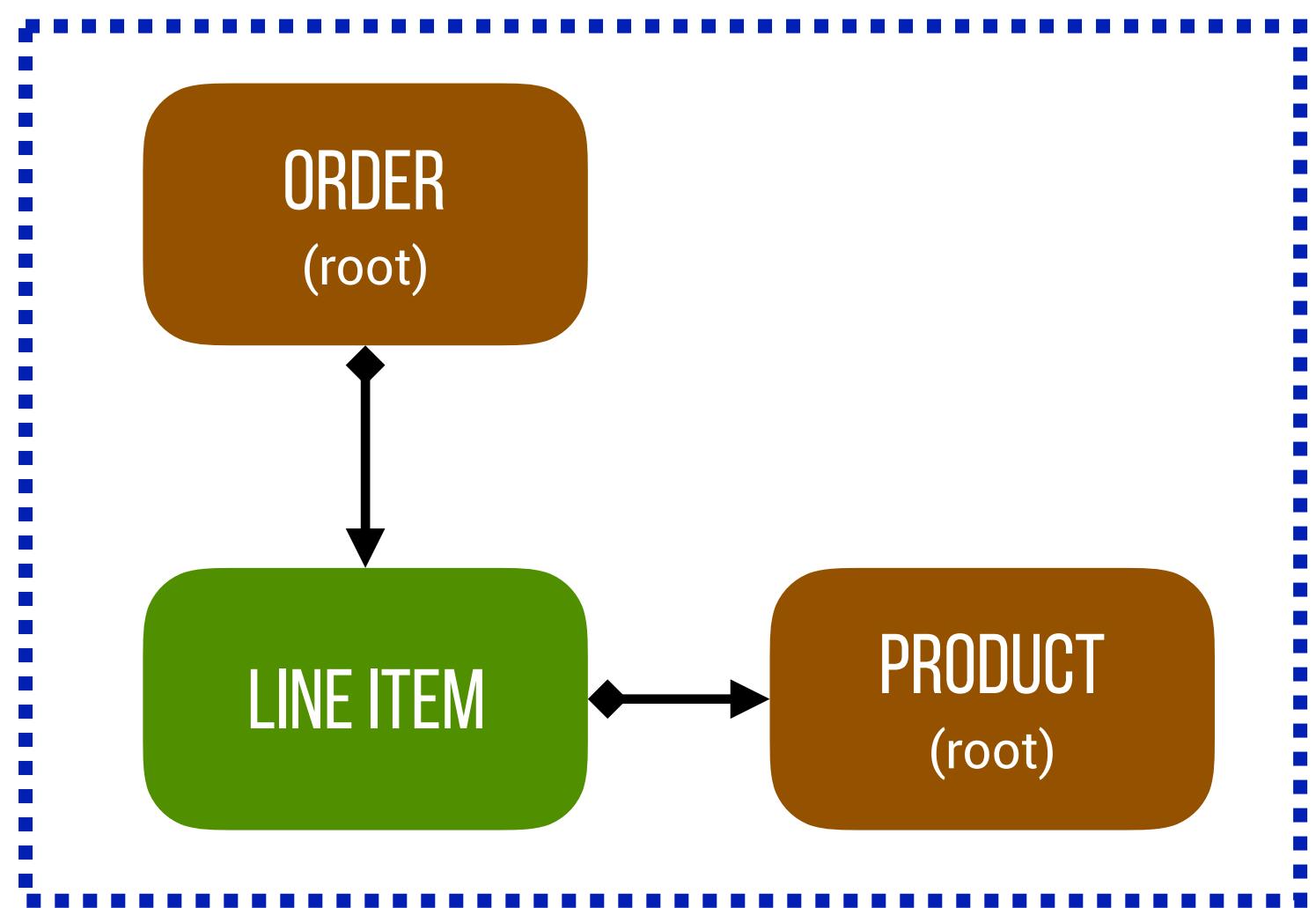
**Reference other Aggregates  
by Identity only.**



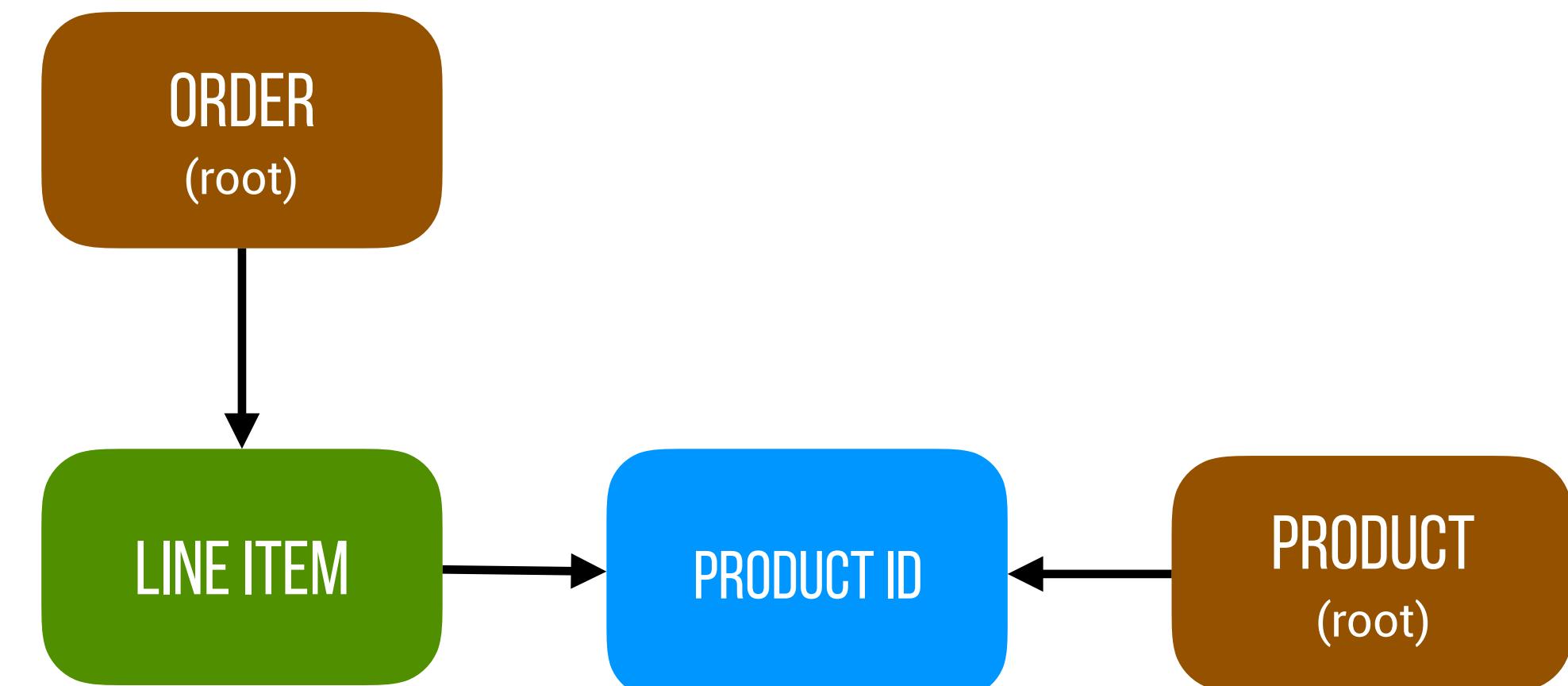
**Composition**



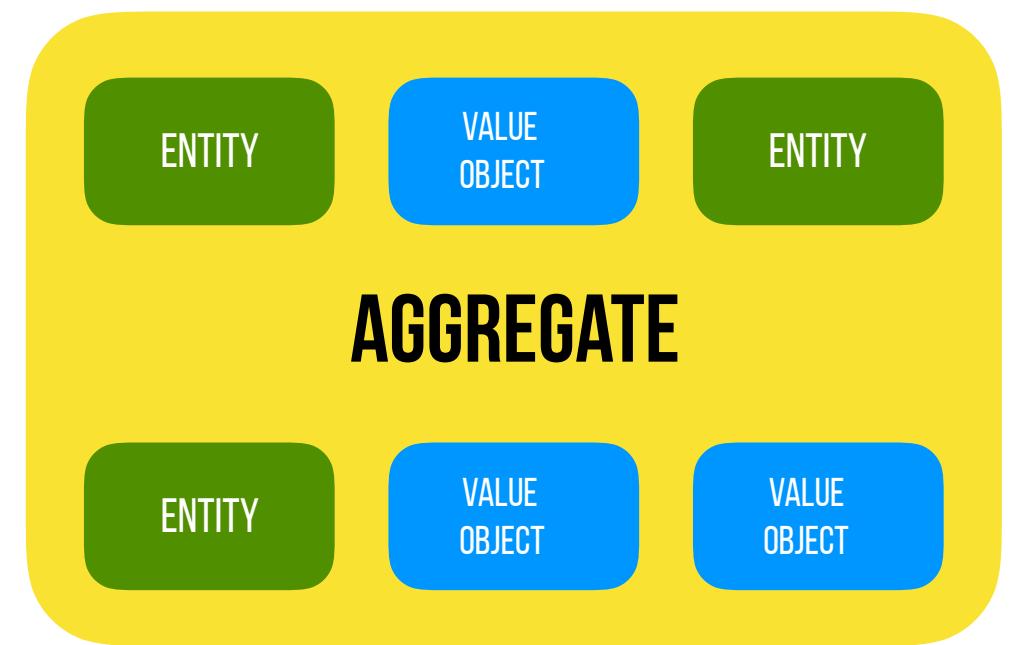
**Reference other Aggregates  
by Identity only.**



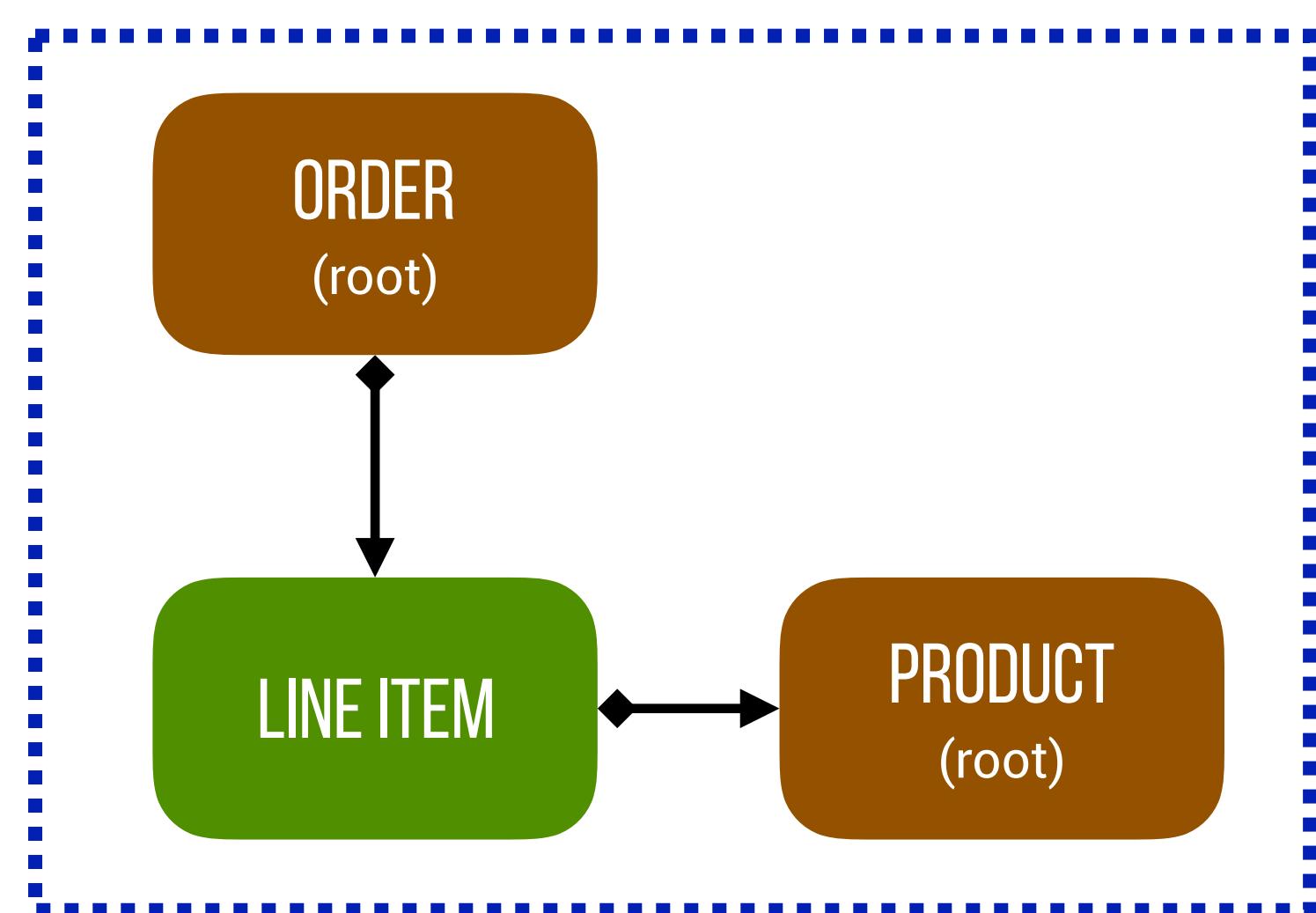
**Composition**



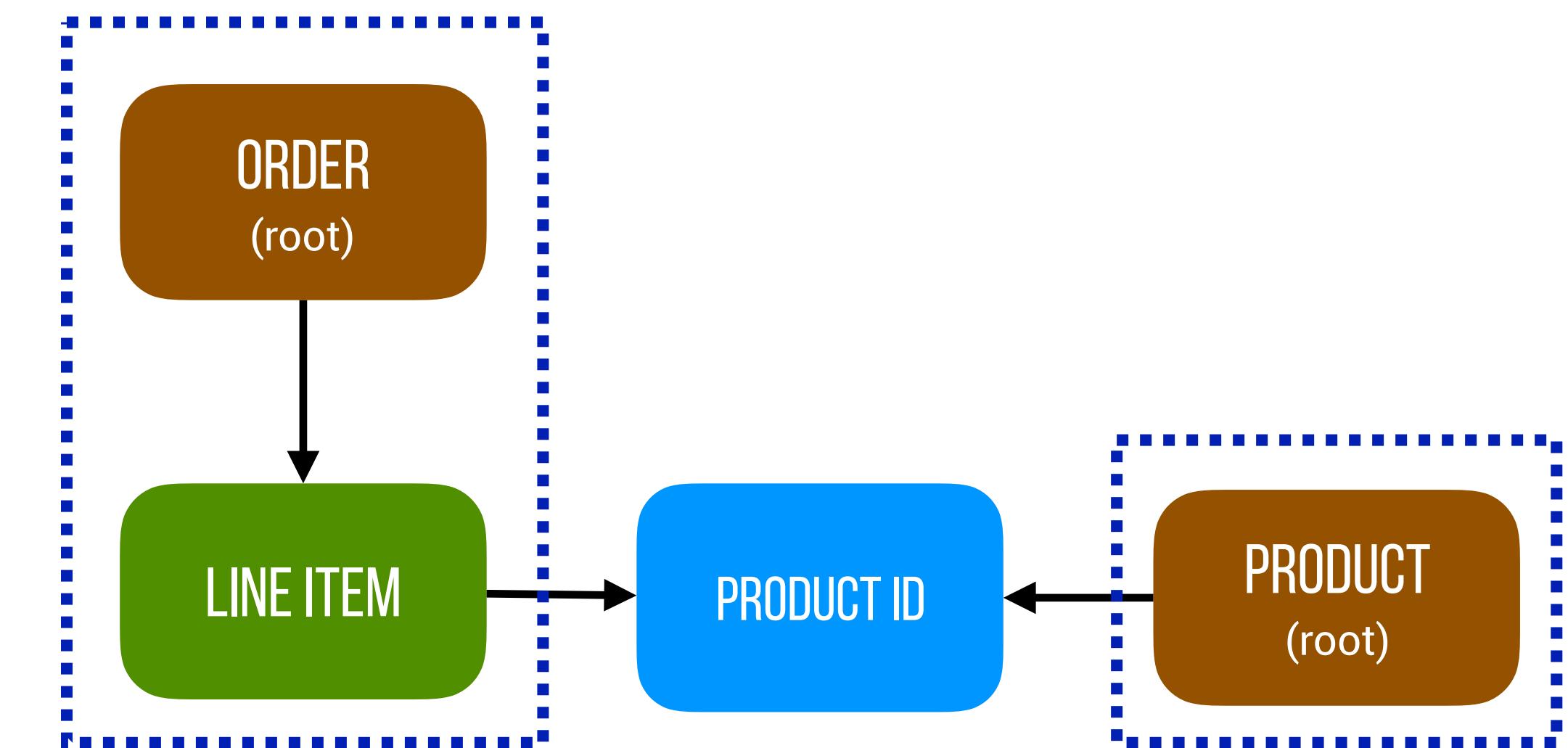
**Reference**



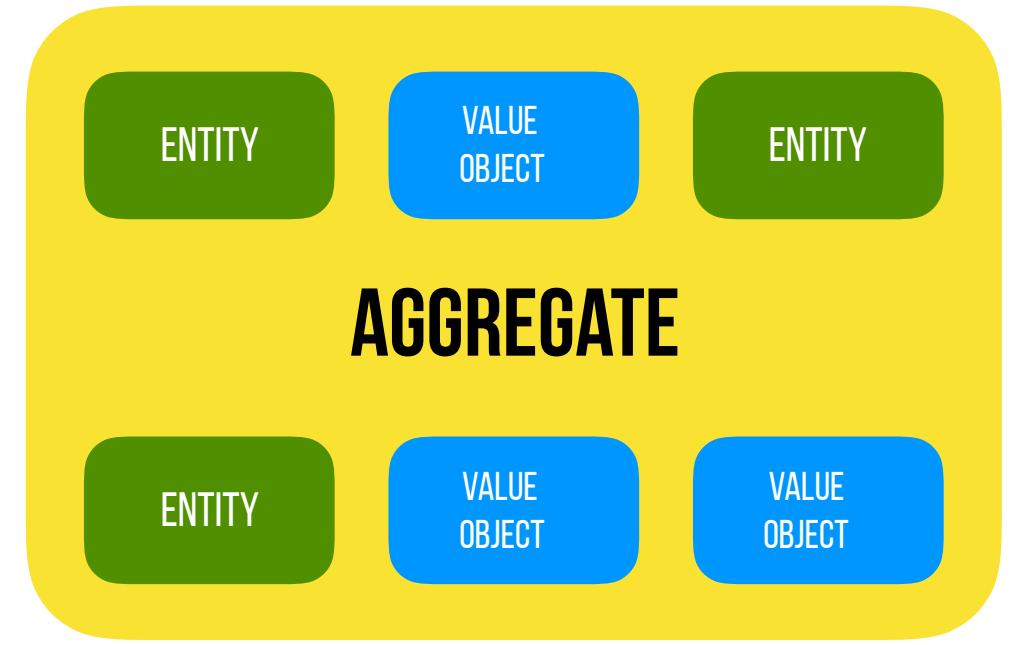
## Reference other Aggregates by Identity only.



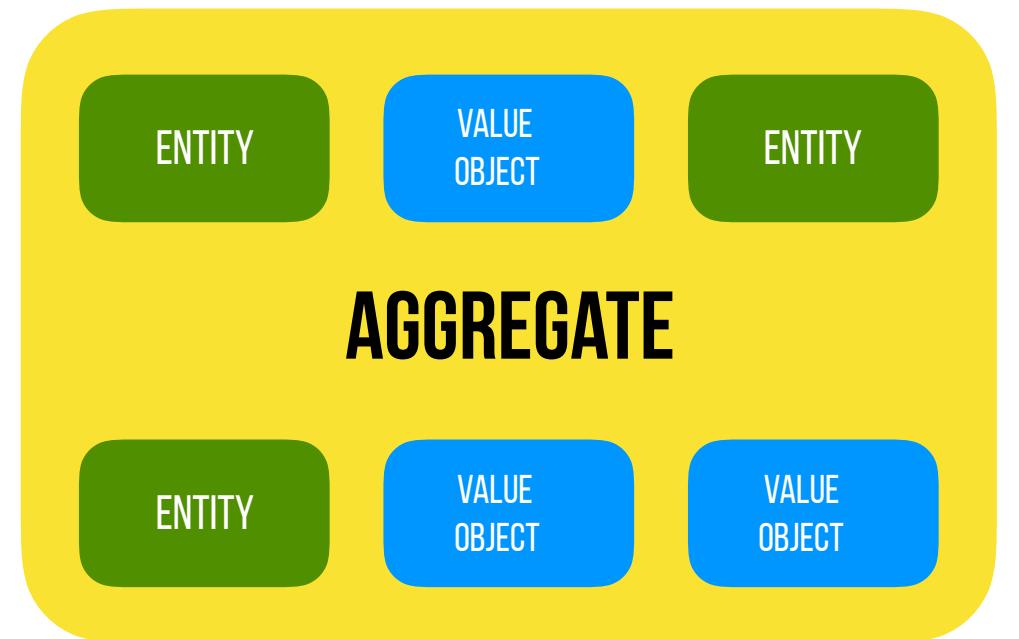
**Composition**



**Reference**



**Update other Aggregates  
using Eventual Consistency.**



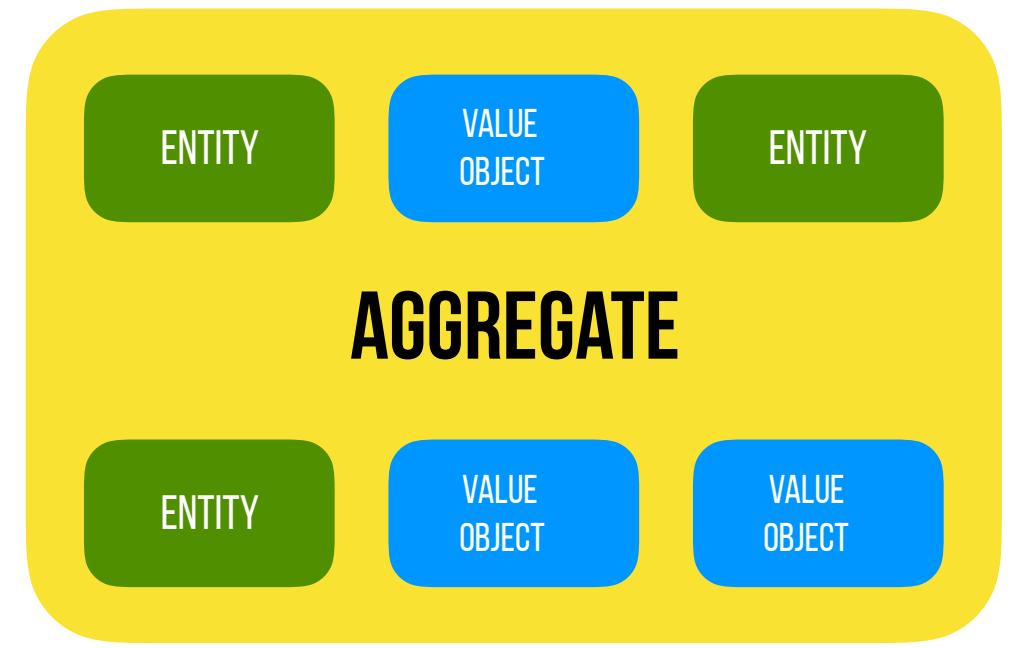
## CUSTOMER

```
{  
  id: 1,  
  first_name: "Joe",  
  last_name: "User"  
}
```

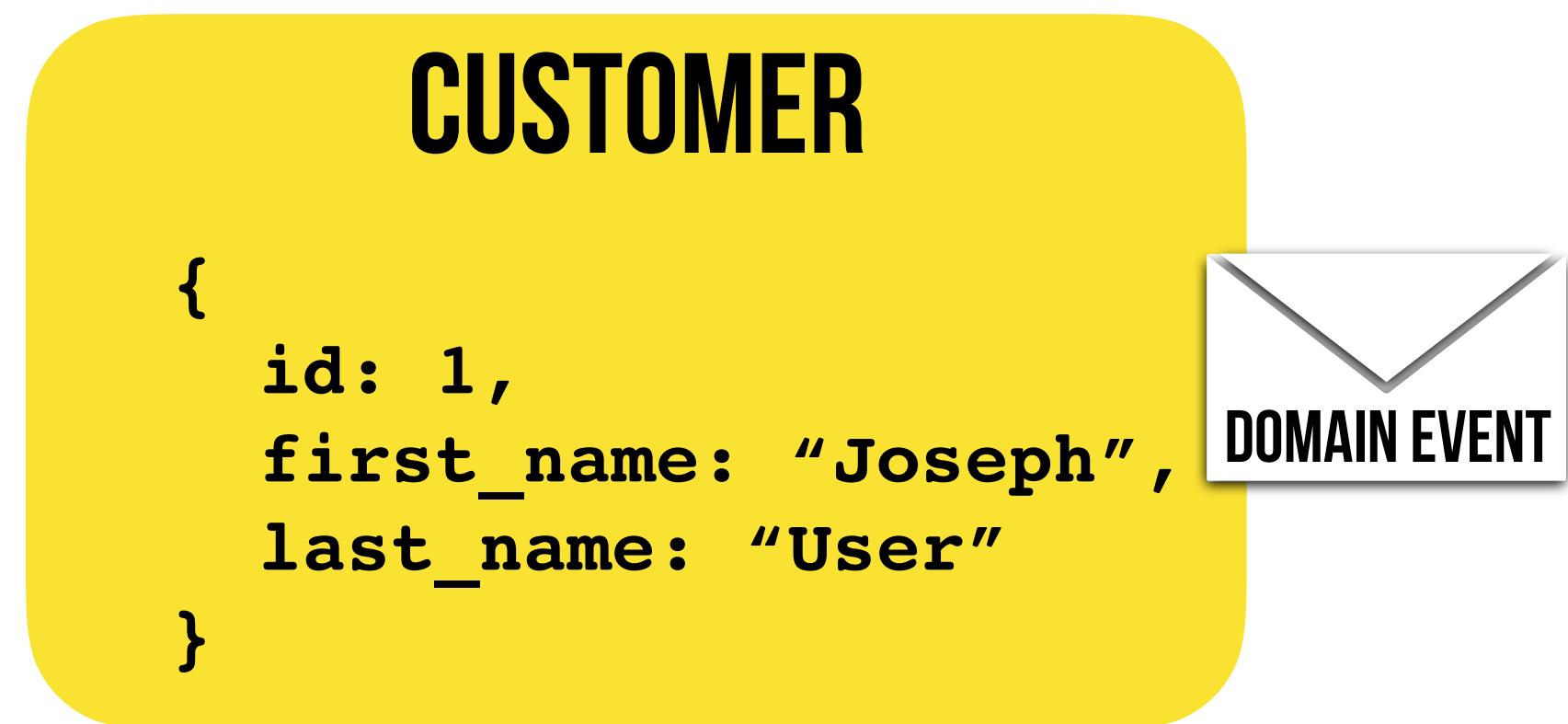
**Update other Aggregates  
using Eventual Consistency.**

## SHIPPING CUSTOMER

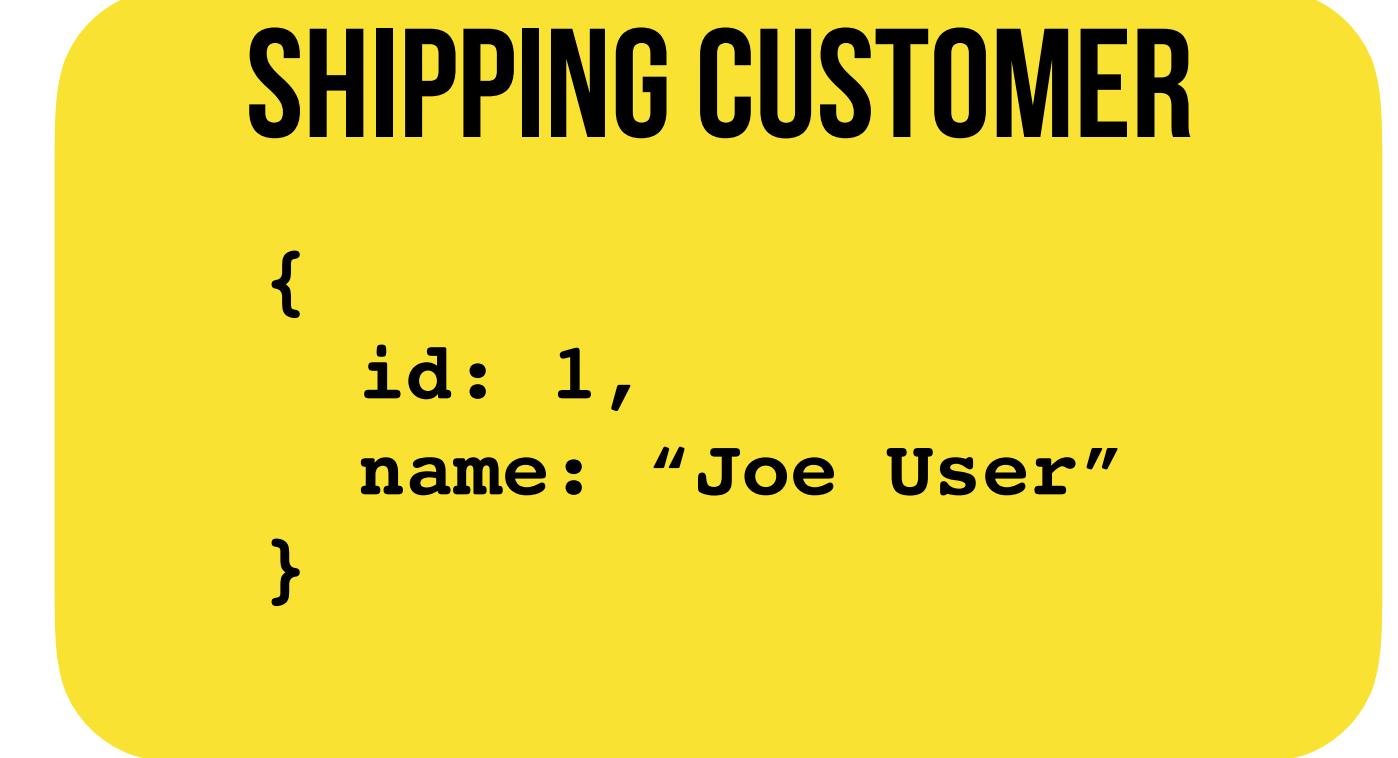
```
{  
  id: 1,  
  name: "Joe User"  
}
```

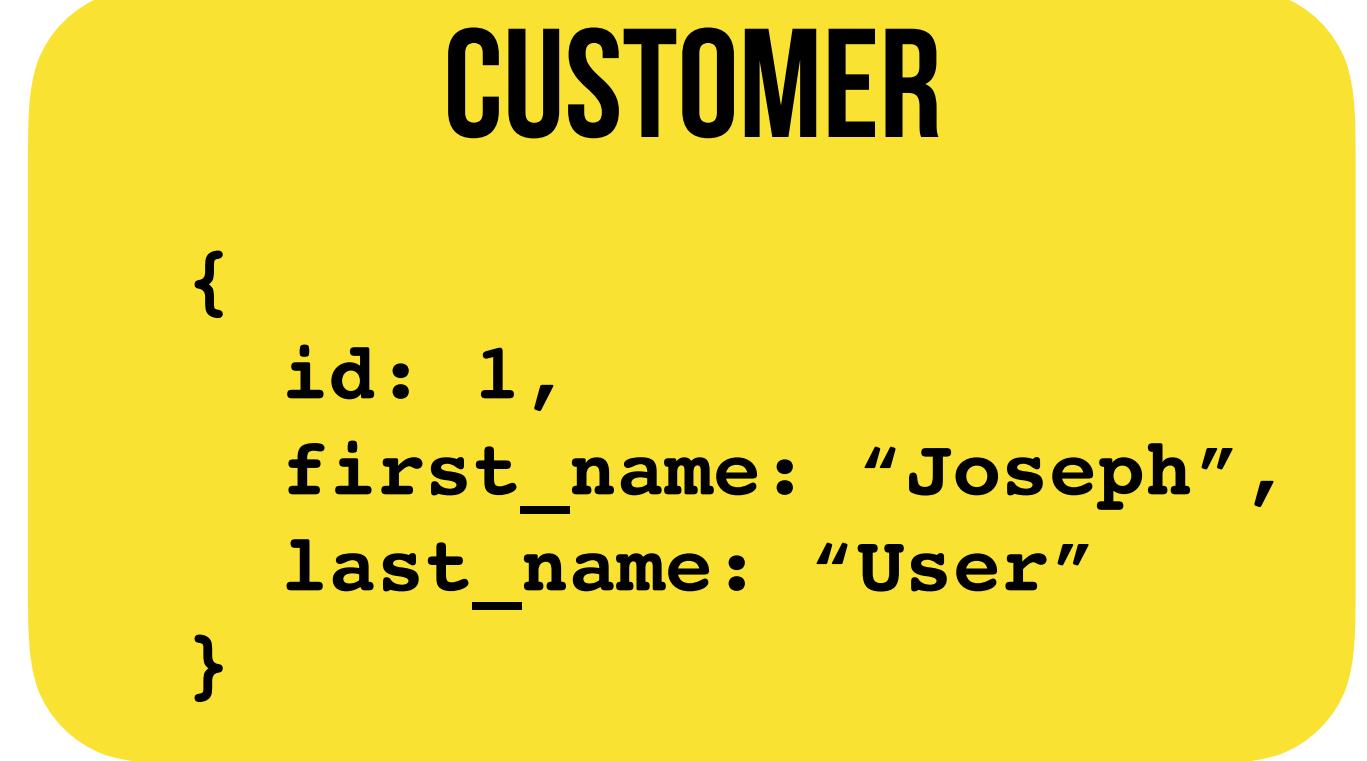
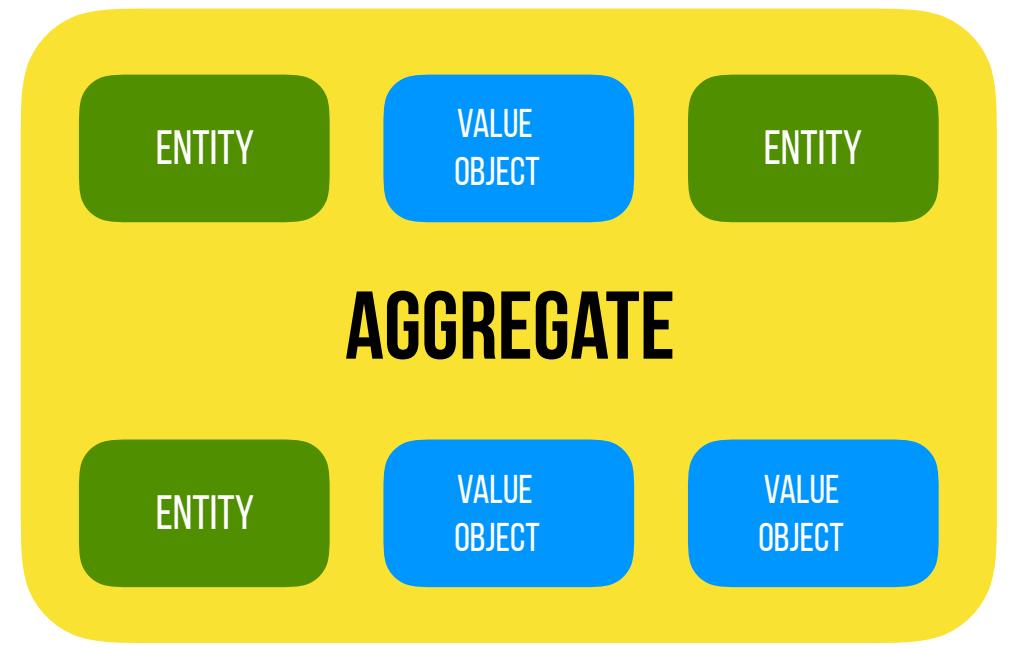


**Update other Aggregates  
using Eventual Consistency.**



**Customer Updated!**





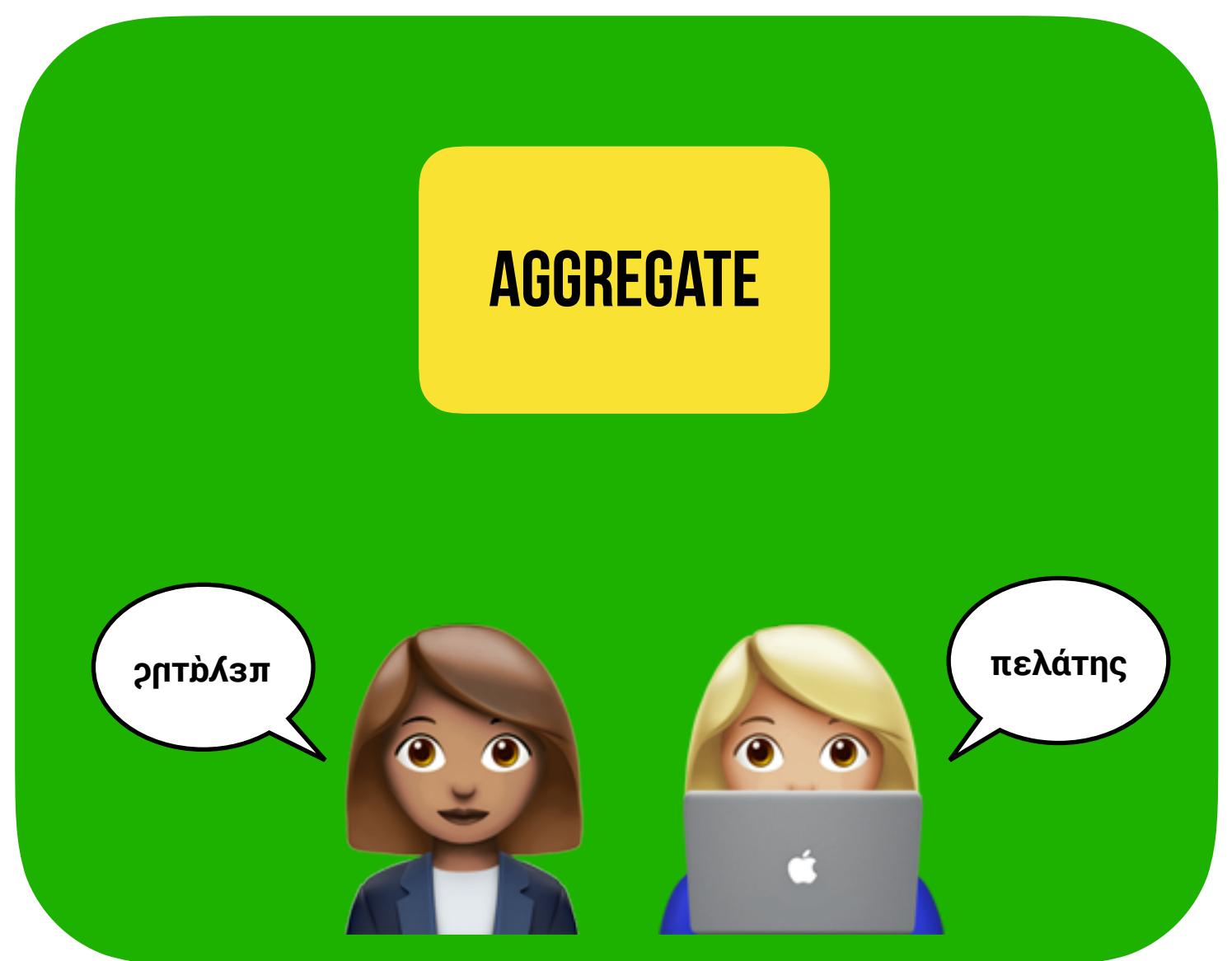
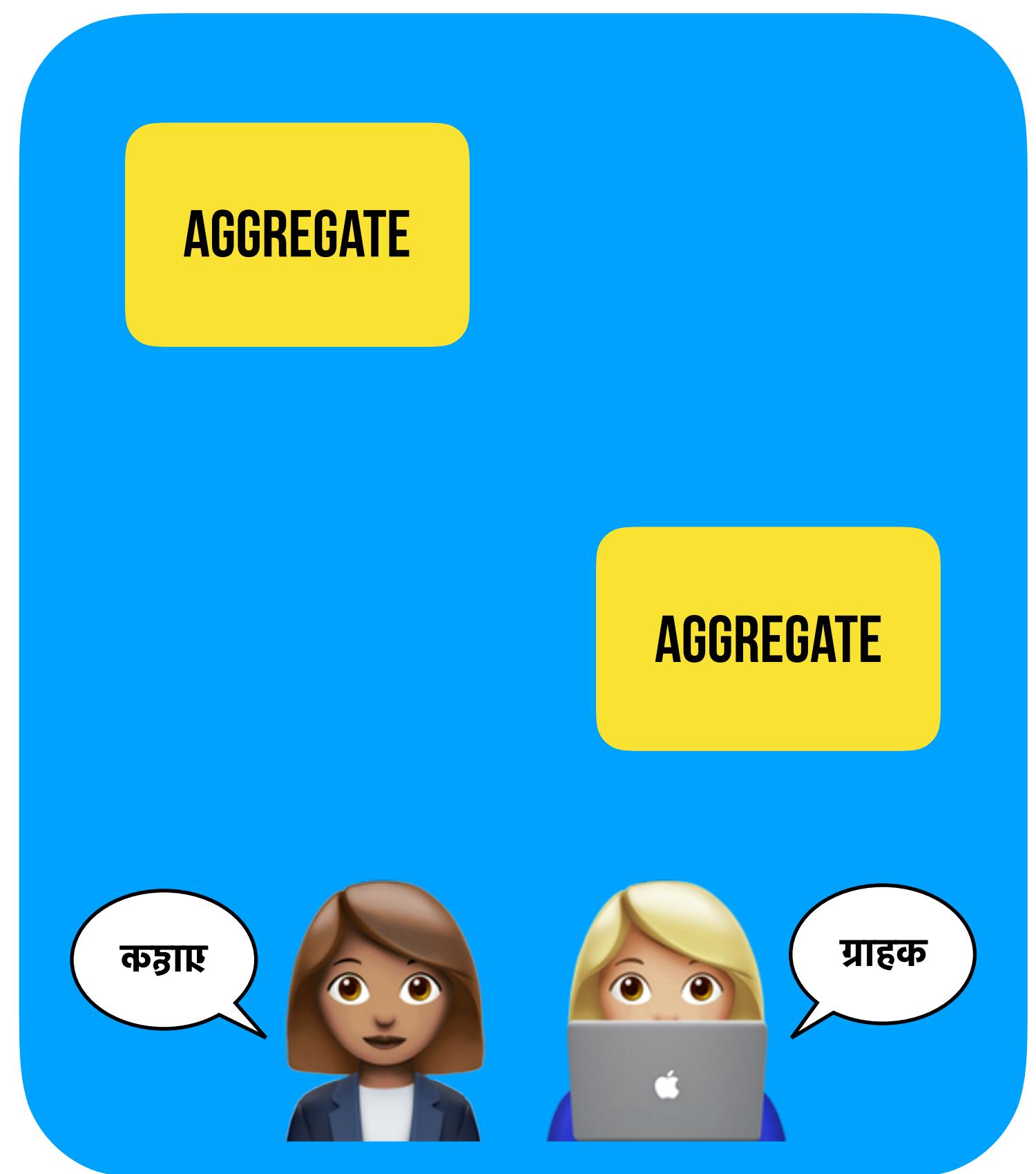
**Customer Updated!**

**Update other Aggregates  
using Eventual Consistency.**



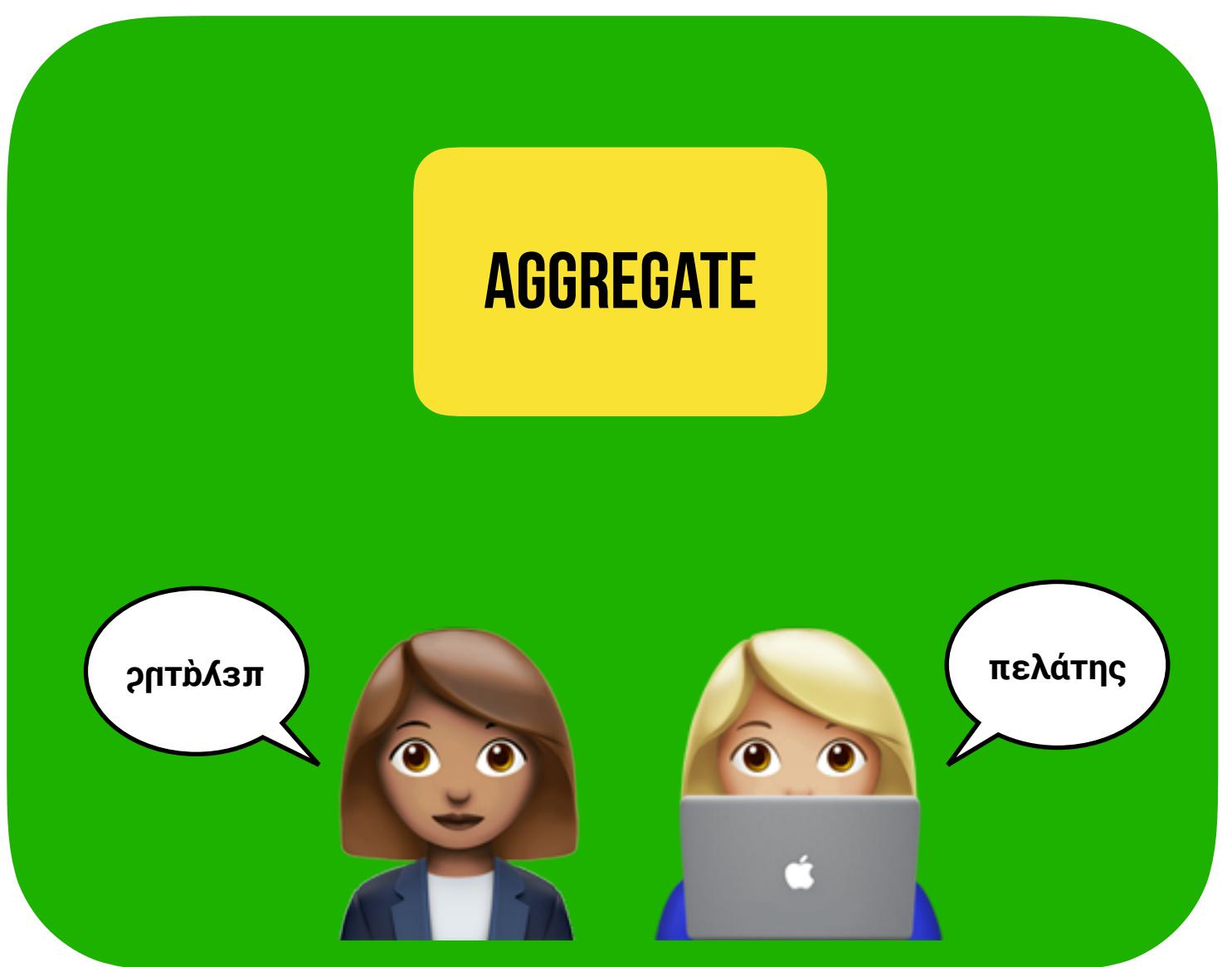
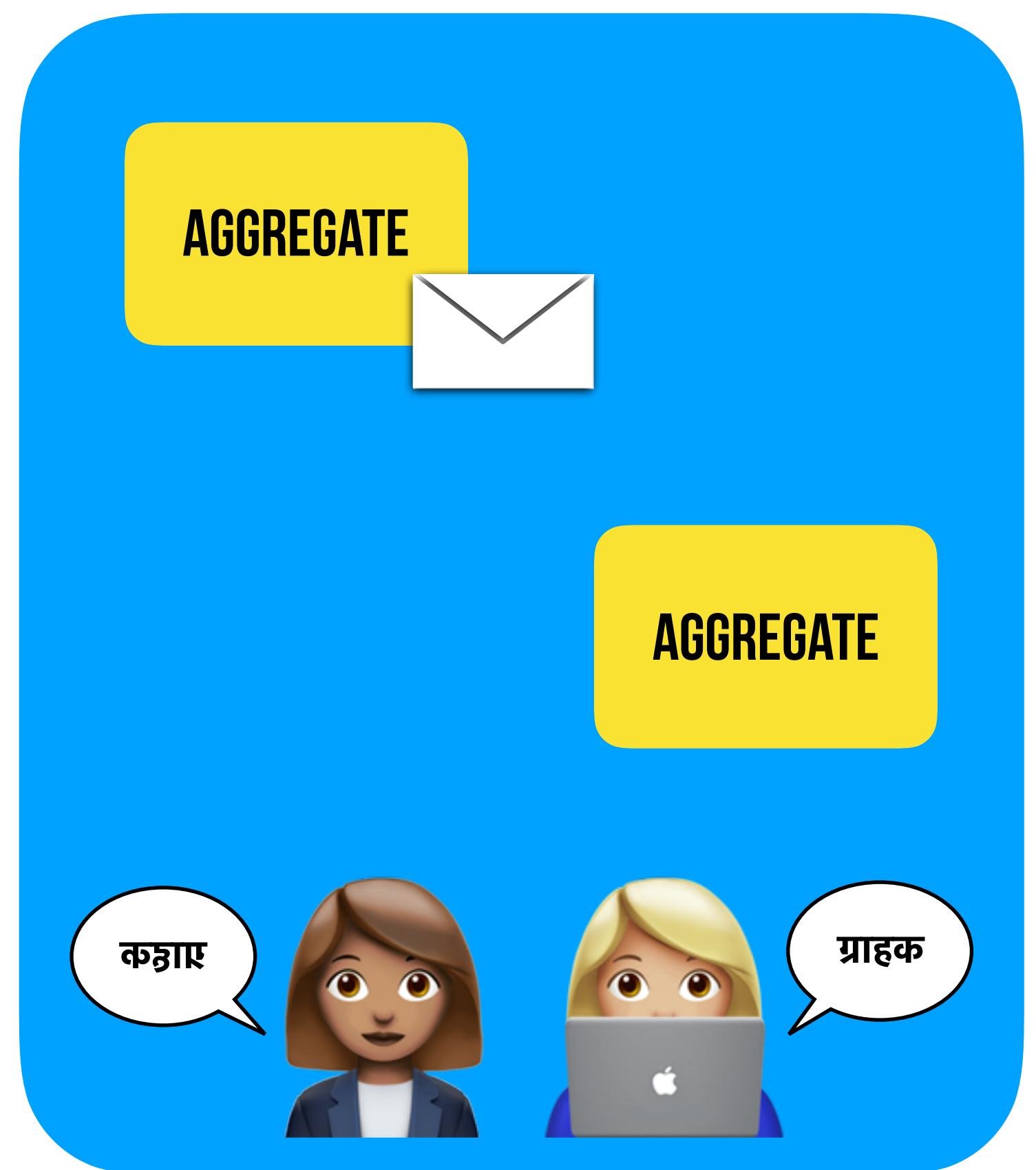


A record of a business-significant occurrence  
within a Bounded Context.



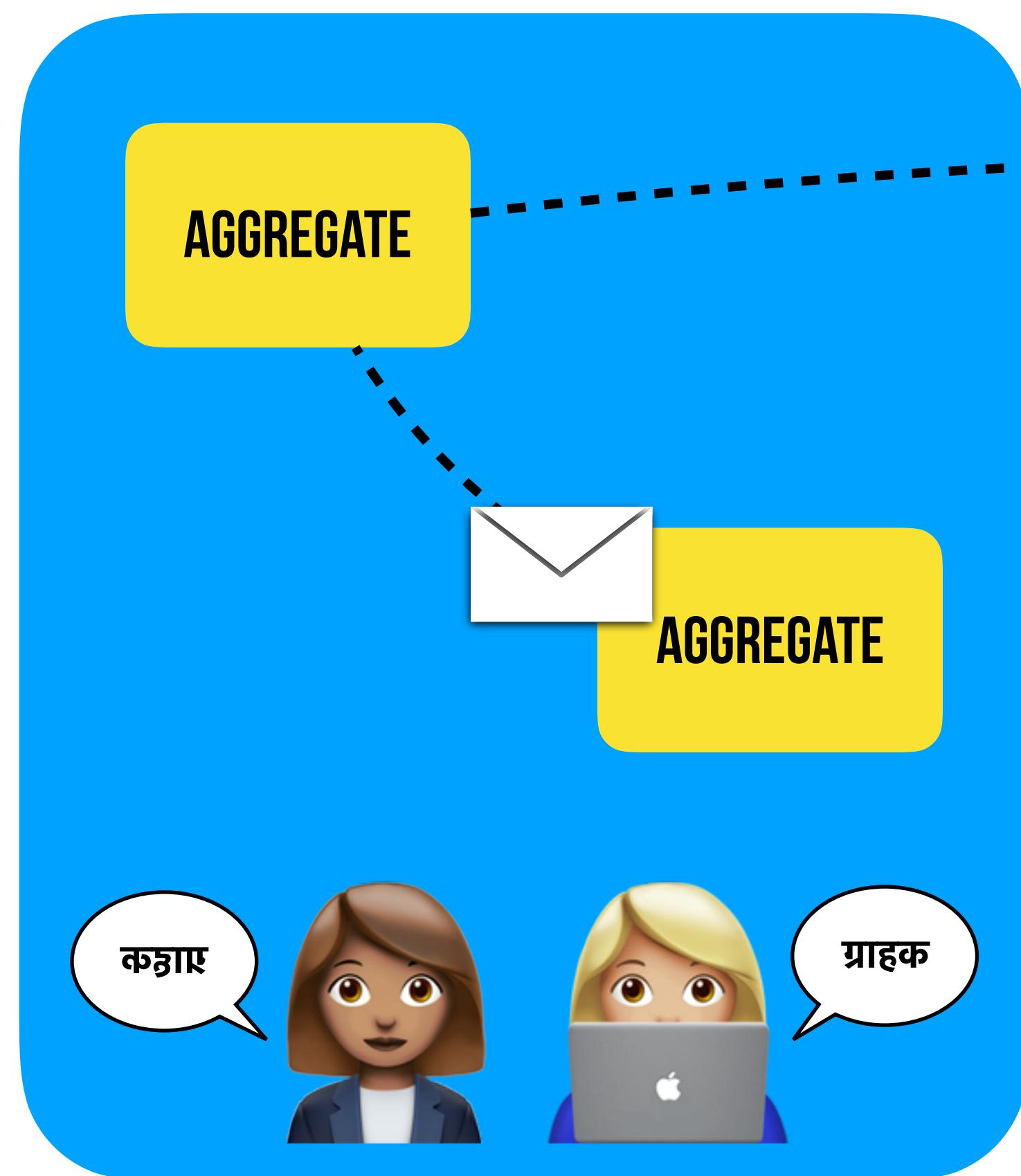


A record of a business-significant occurrence  
within a Bounded Context.

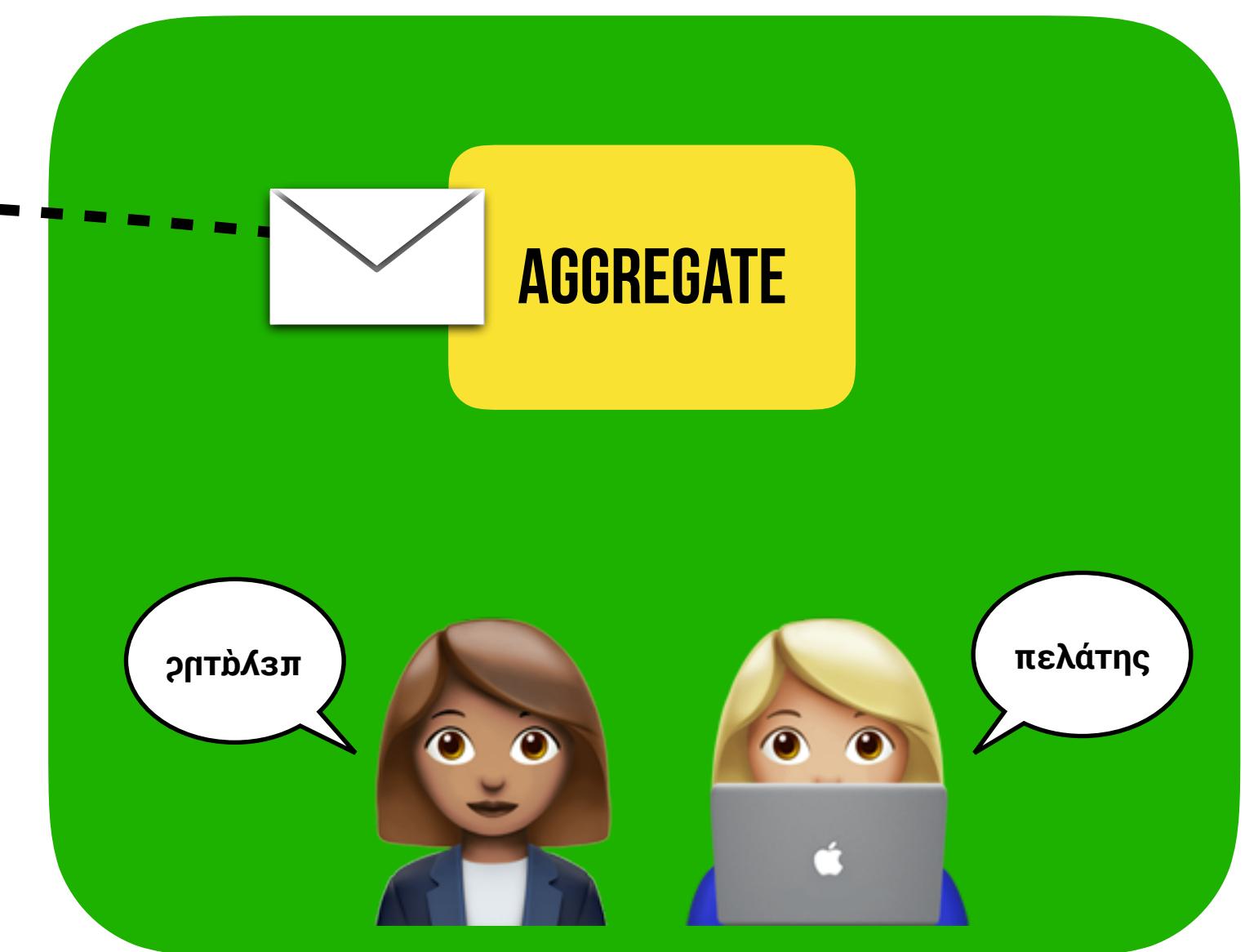




A record of a business-significant occurrence  
within a Bounded Context.



- OrderPlacedEvent
- ItemAddedToOrderEvent
- OrderCanceledEvent
- OrderPaidEvent
- OrderFulfilledEvent
- OrderDeliveredEvent

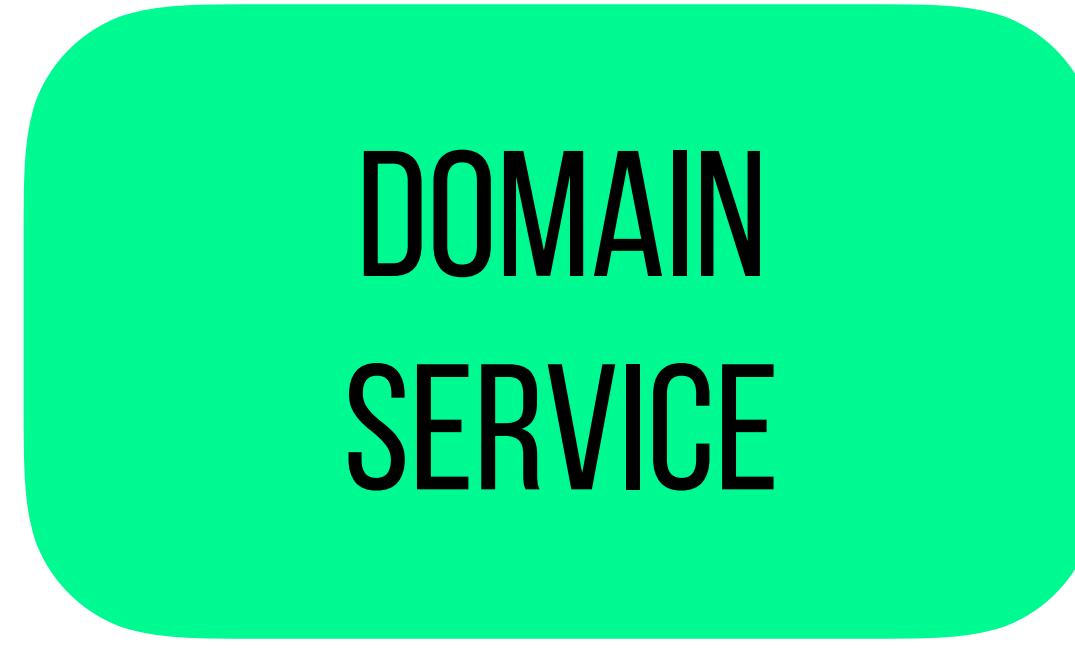


# DOMAIN SERVICES

**Stateless operations that don't belong on an Entity.**

# DOMAIN SERVICES

**Stateless operations that don't belong on an Entity.**



DOMAIN  
SERVICE

[Apply Coupon to Order](#)

# DOMAIN SERVICES

**Stateless operations that don't belong on an Entity.**

**ORDER**

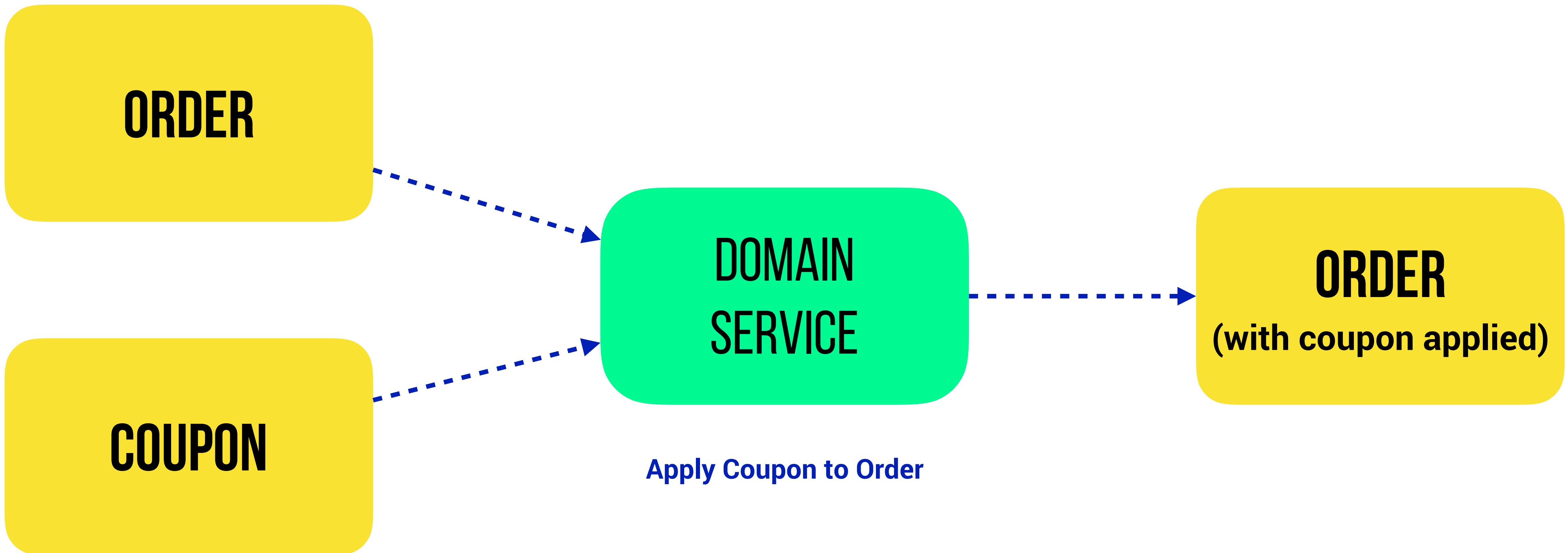
**COUPON**

**DOMAIN  
SERVICE**

[Apply Coupon to Order](#)

# DOMAIN SERVICES

Stateless operations that don't belong on an Entity.



# APPLICATION SERVICES

The first client to your domain model.

# APPLICATION SERVICES

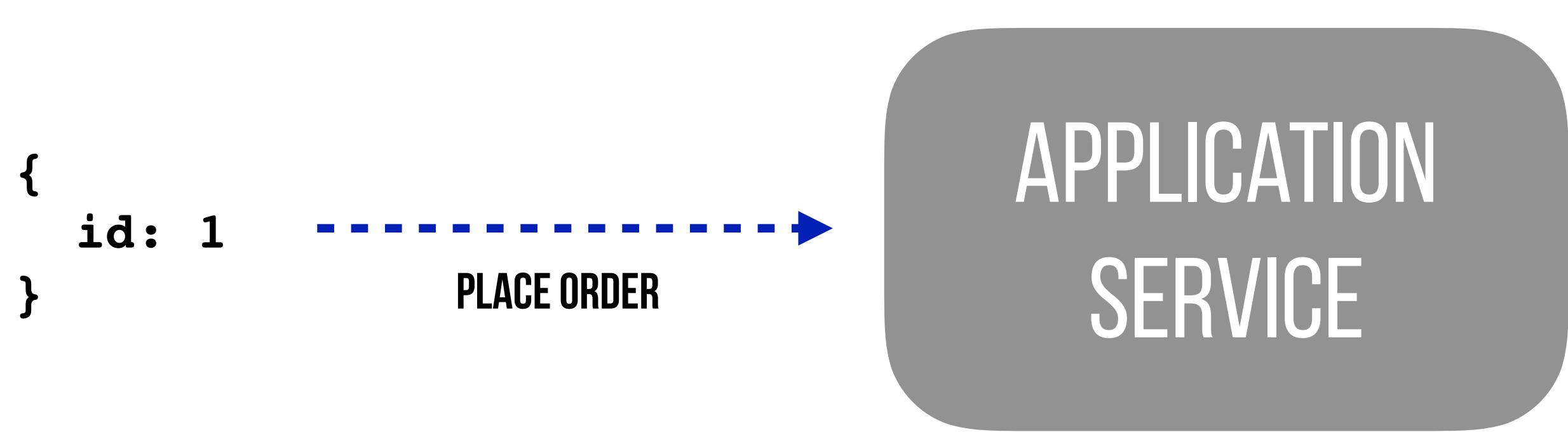
The first client to your domain model.



APPLICATION  
SERVICE

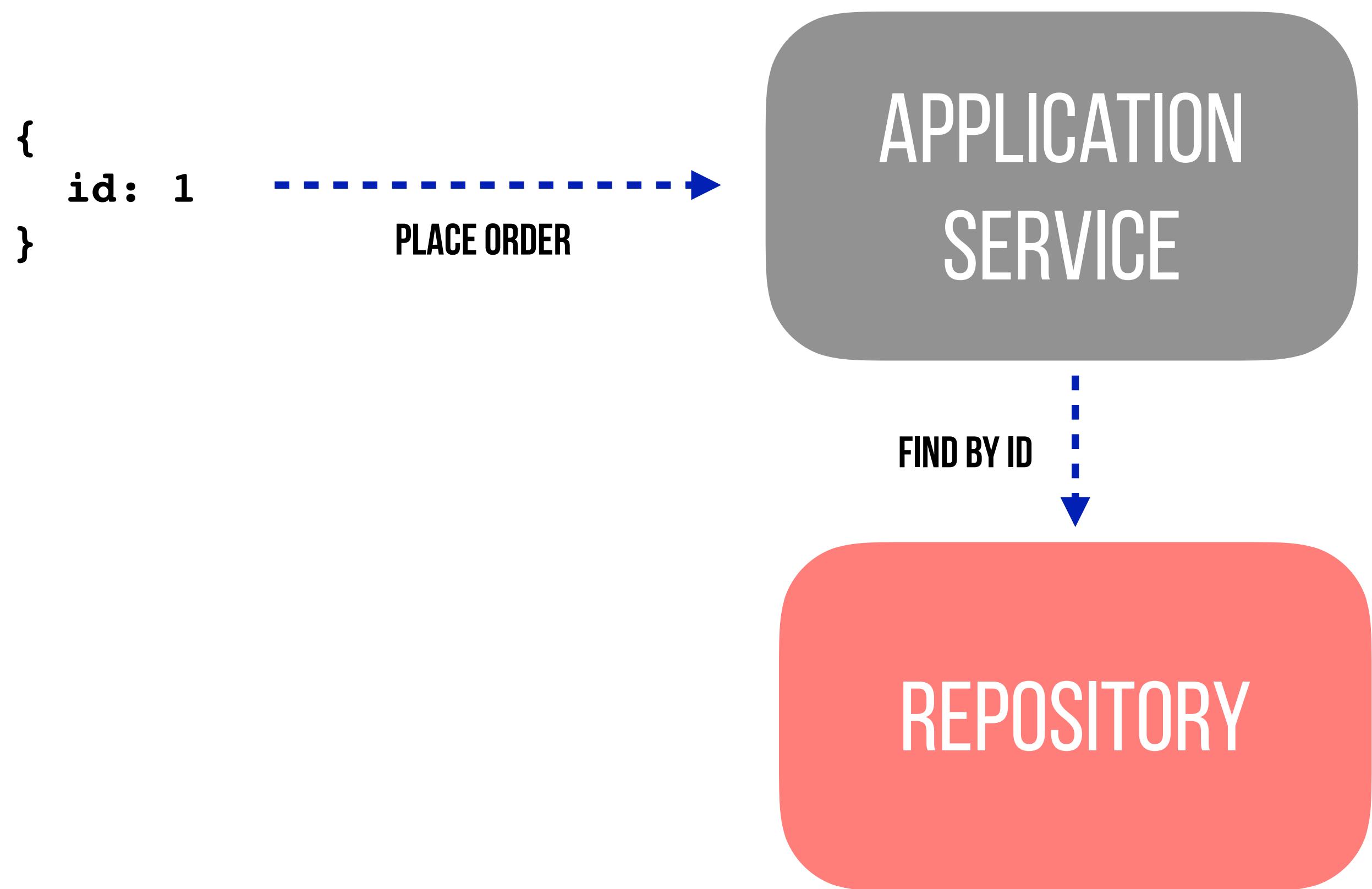
# APPLICATION SERVICES

The first client to your domain model.



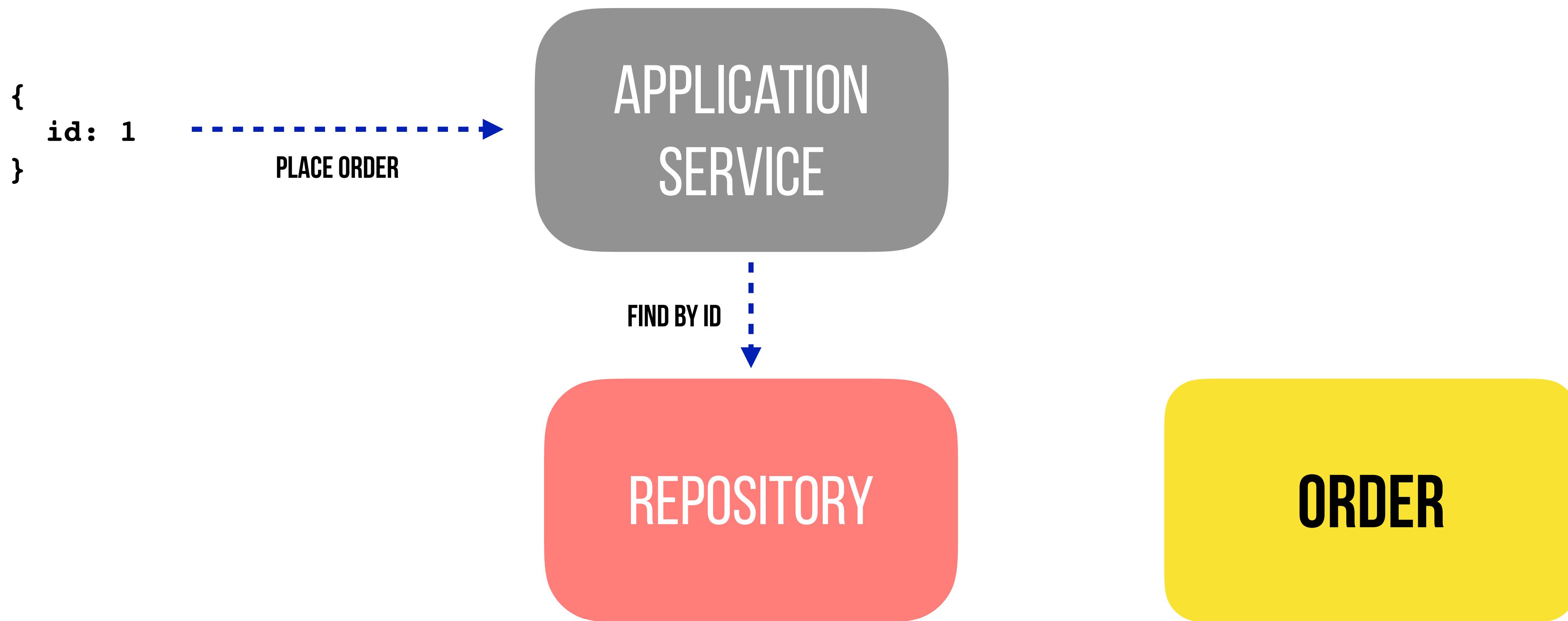
# APPLICATION SERVICES

The first client to your domain model.



# APPLICATION SERVICES

The first client to your domain model.



# APPLICATION SERVICES

The first client to your domain model.

