
FOR INSTRUCTOR PURPOSES ONLY

MATERIALS

- + [Lab](#)
- + [Lesson](#)
- + Lesson Starter Code

OBJECT DESIGN

Wellington Moreno

Lead iOS Instructor, General Assembly



LESSON



**OUR OVERALL GOAL IS TO
MODEL OBJECTS THAT
REPRESENT THE ACTIVITY OF
MAKING A CUP OF TEA.**

INDEPENDENT PRACTICE: PLAYGROUNDS



CODE

5 mins

DIRECTIONS

Create the following classes:

- | | |
|------------------|--------------|
| + Coffee Machine | + Coffee Pot |
| + Tank of Water | + Water |
| + Cup | + Tea Bag |

NOTES

At this point, you don't have to add any methods or variables.

LESSON

LEARNING OBJECTIVES

- + **Design** object relationships
- + **Articulate** what Delegation is
- + **Write** classes using Object composition

DEFINITION

**OBJECT-ORIENTED DESIGN IS THE
ART OF EFFECTIVELY ORGANIZING
INTERACTING OBJECTS IN A
SOFTWARE SYSTEM.**

QUESTION

WHO CARES?

LESSON

PROFESSIONAL

LESSON

ADAPTABLE

LESSON

INTUITIVE

LESSON

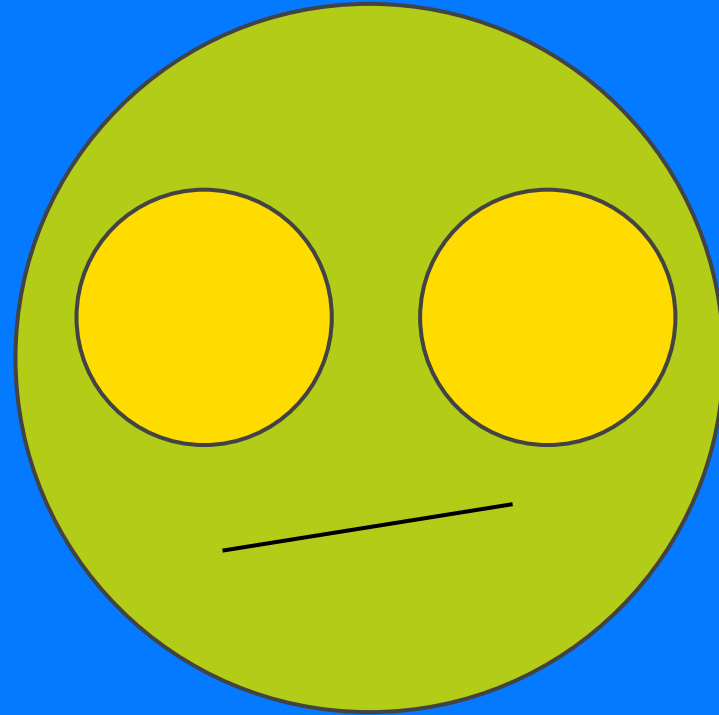
FUN

COMPOSITION

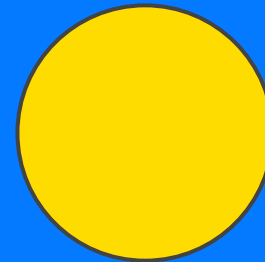
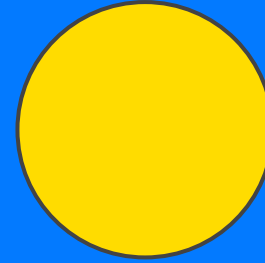
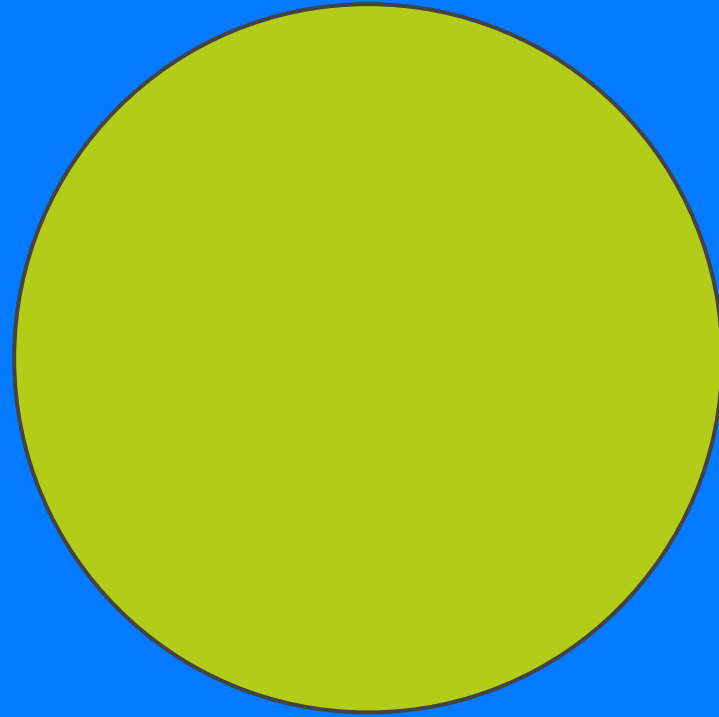
OBJECT COMPOSITION



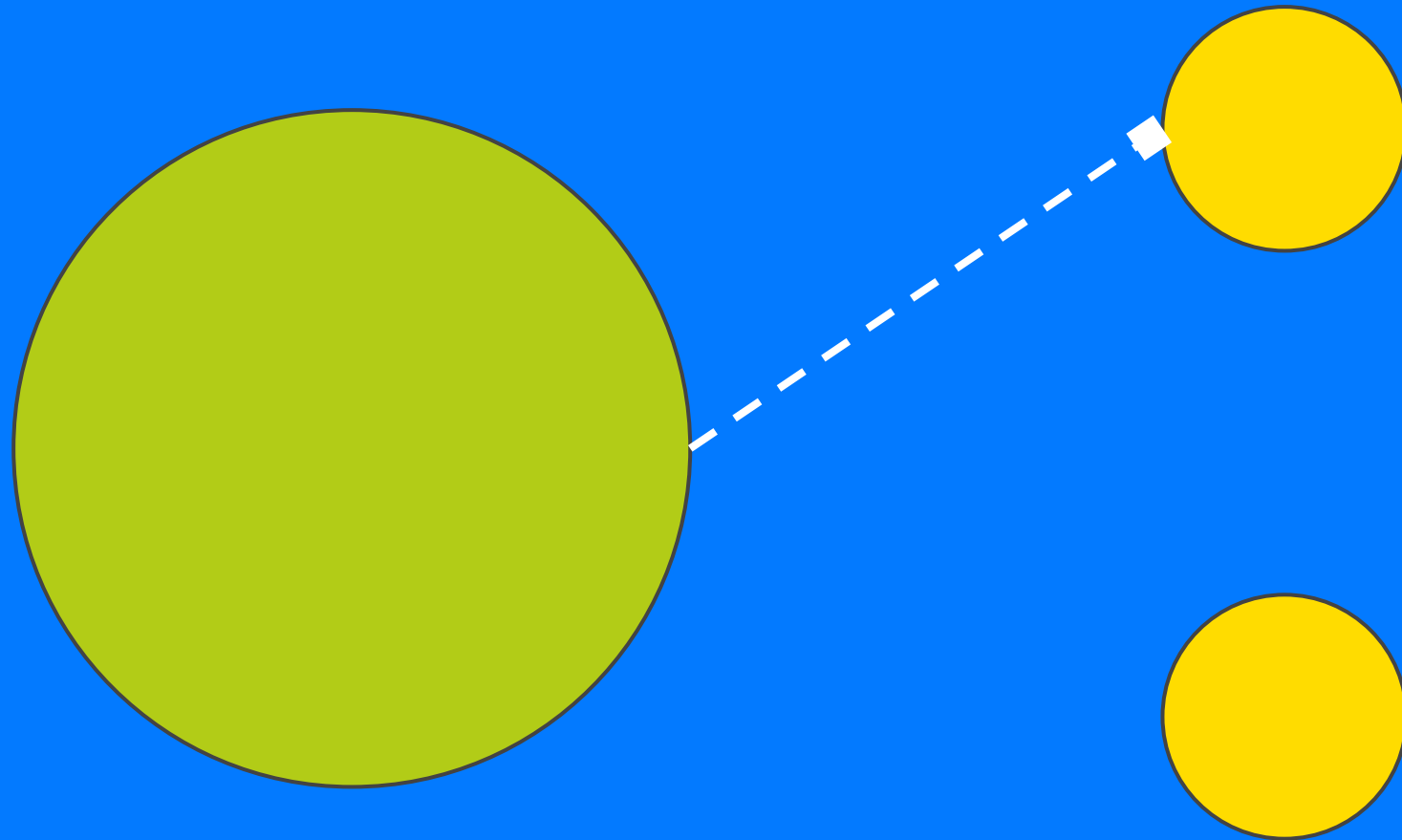
COMPOSITION



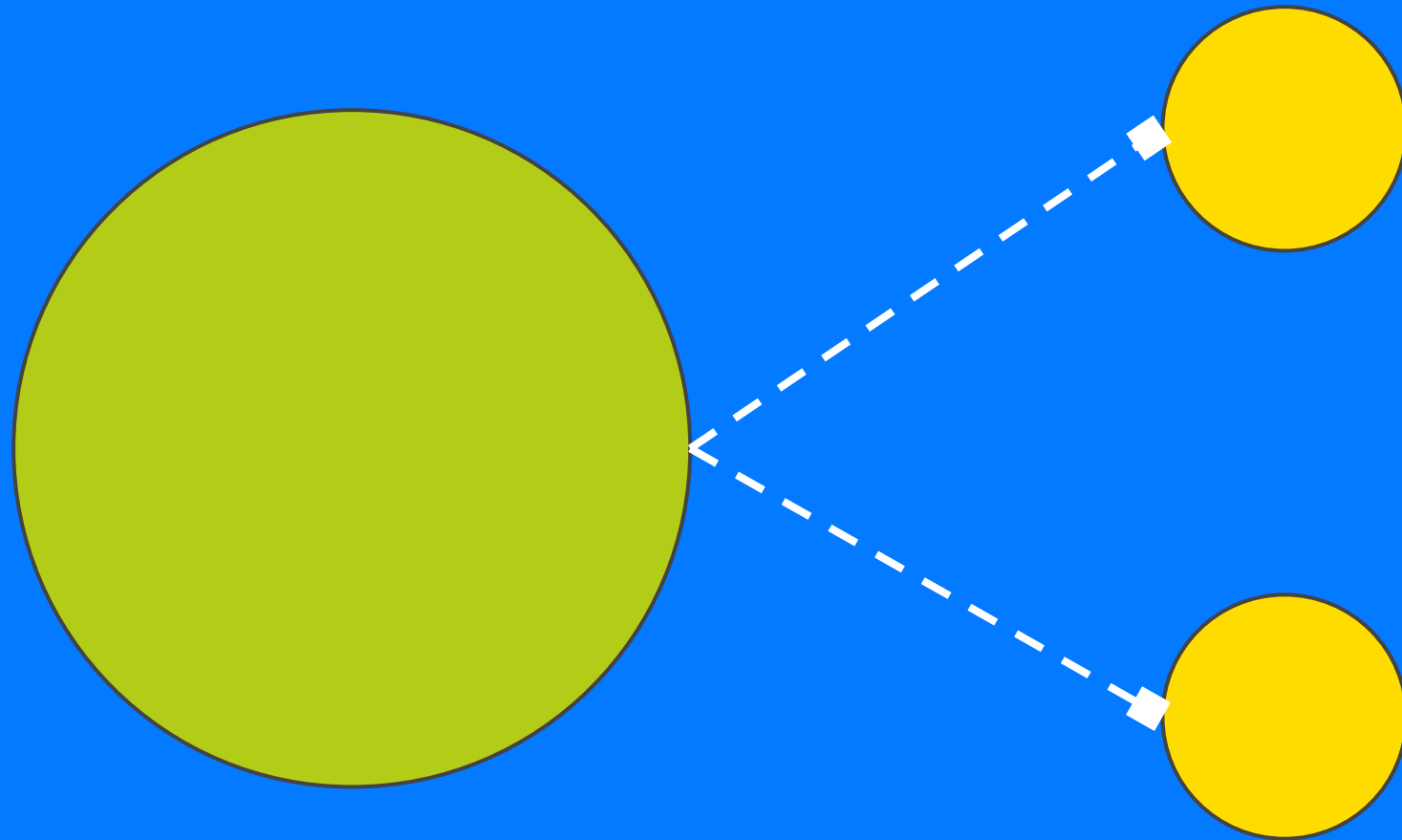
COMPOSITION



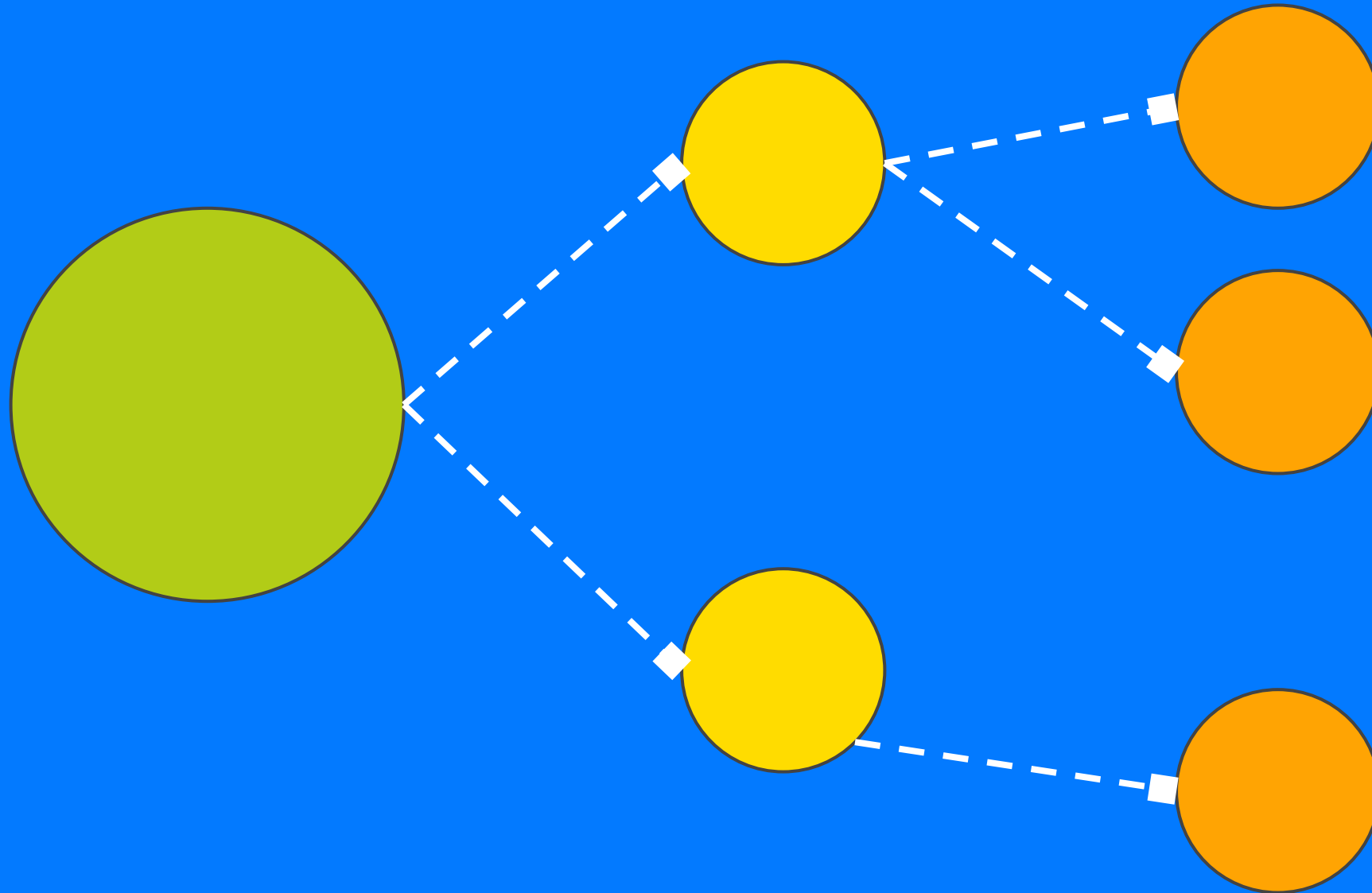
COMPOSITION



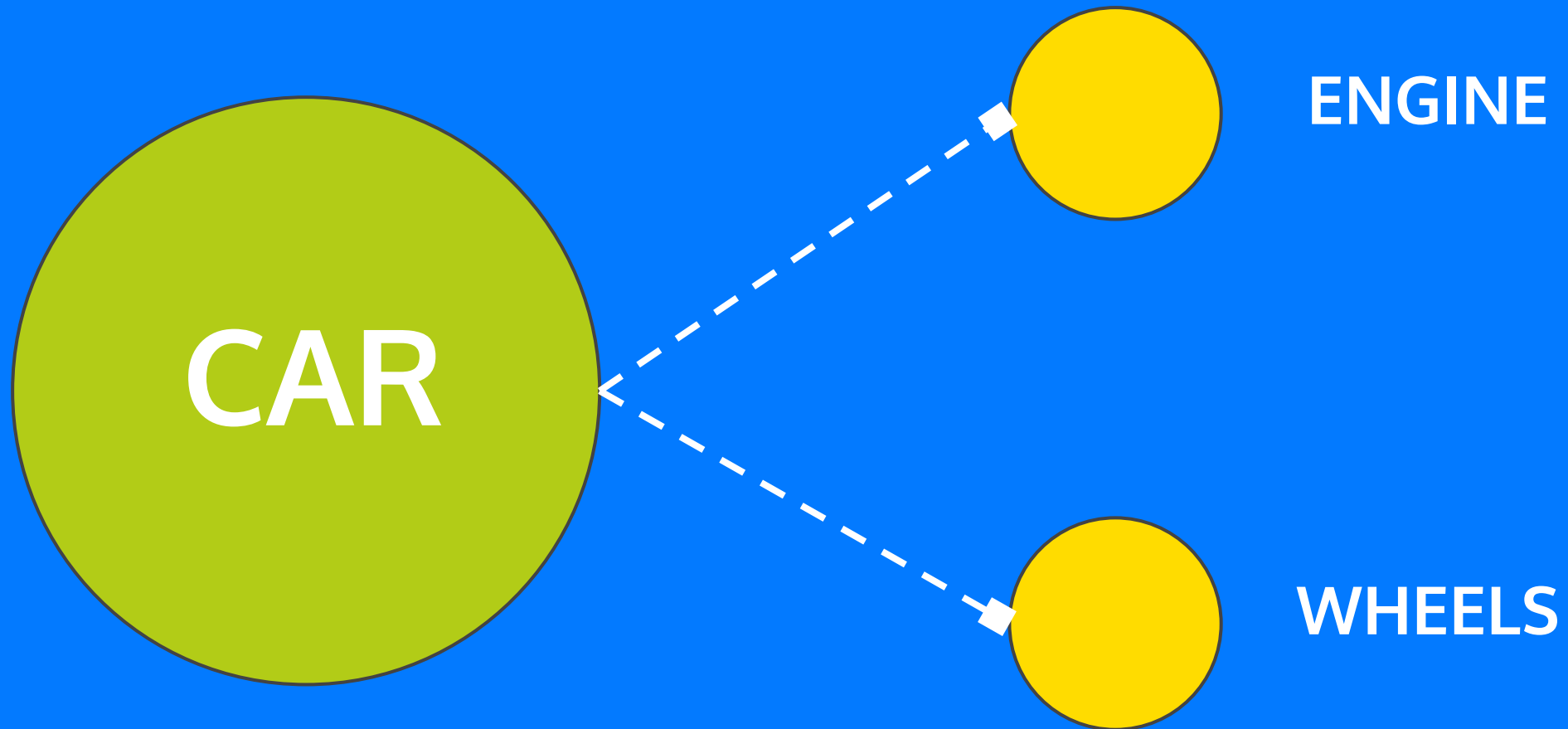
COMPOSITION



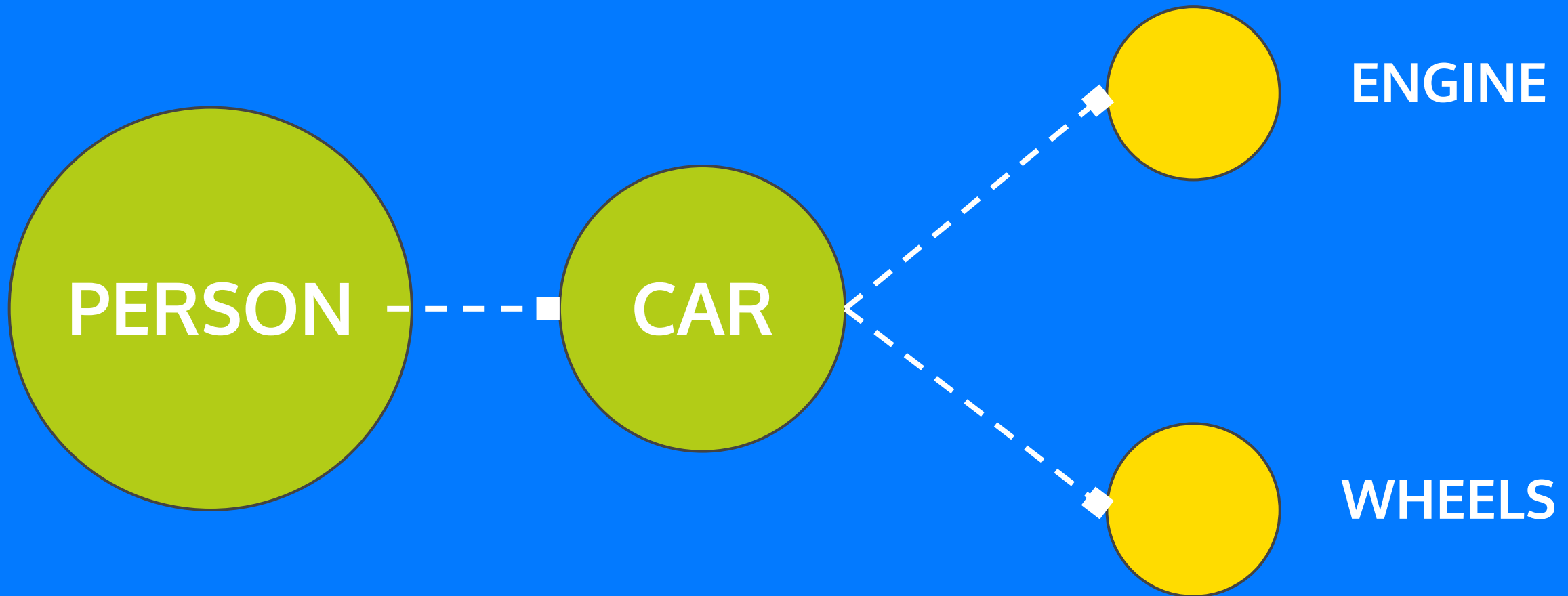
COMPOSITION



COMPOSITION



COMPOSITION



ACTIVITY: STOP-AND-DRAW



5 mins

DIRECTIONS

Draw out the object tree for the classes you wrote.

DELIVERABLE

Share your drawing with a partner

PRACTICE: PLAYGROUNDS



5 mins

DIRECTIONS

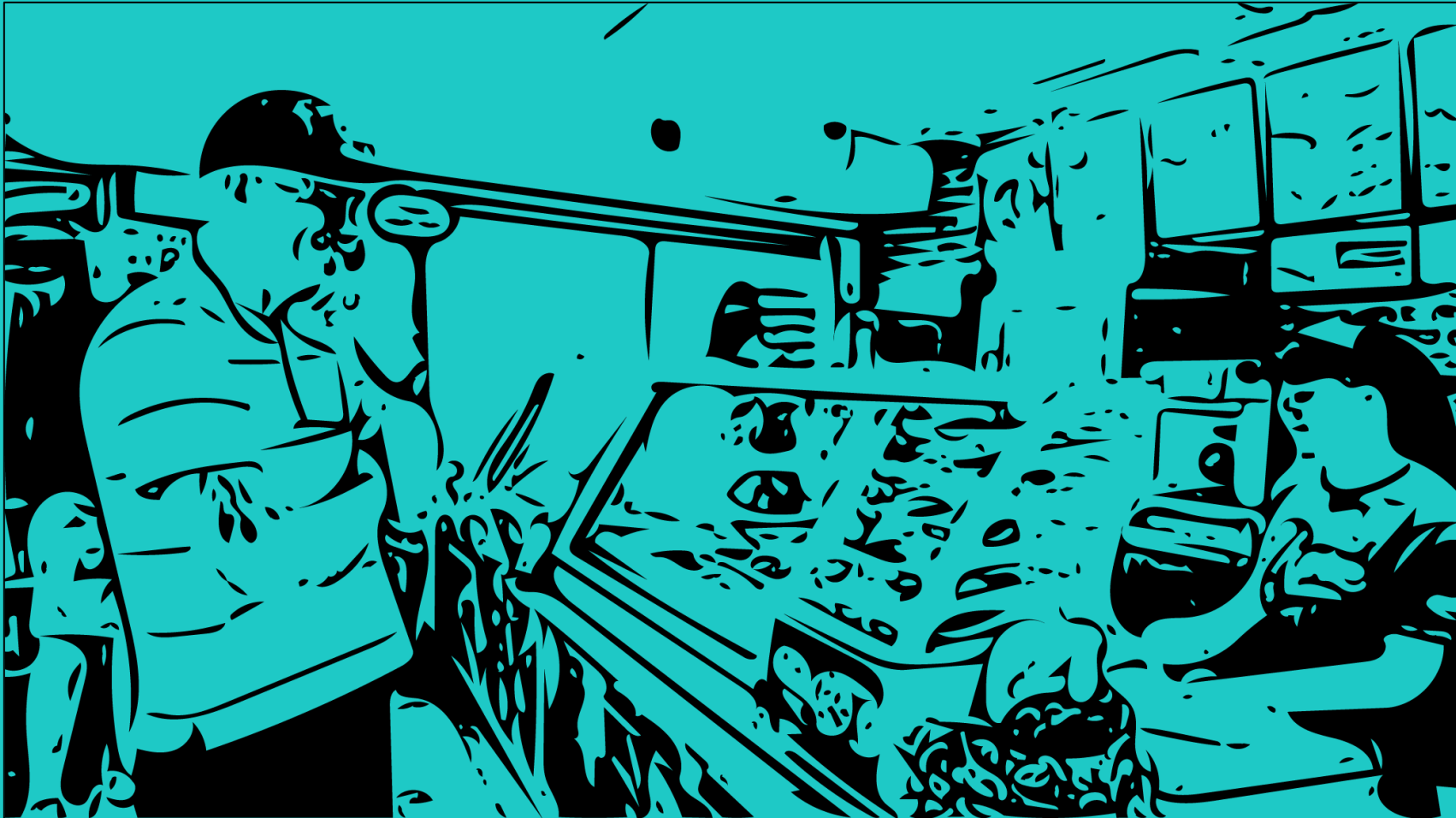
Update your classes using Object Composition.

NOTES

Think about using Optionals
For example, a Cup might have a Tea Bag, or it might not

DEMO

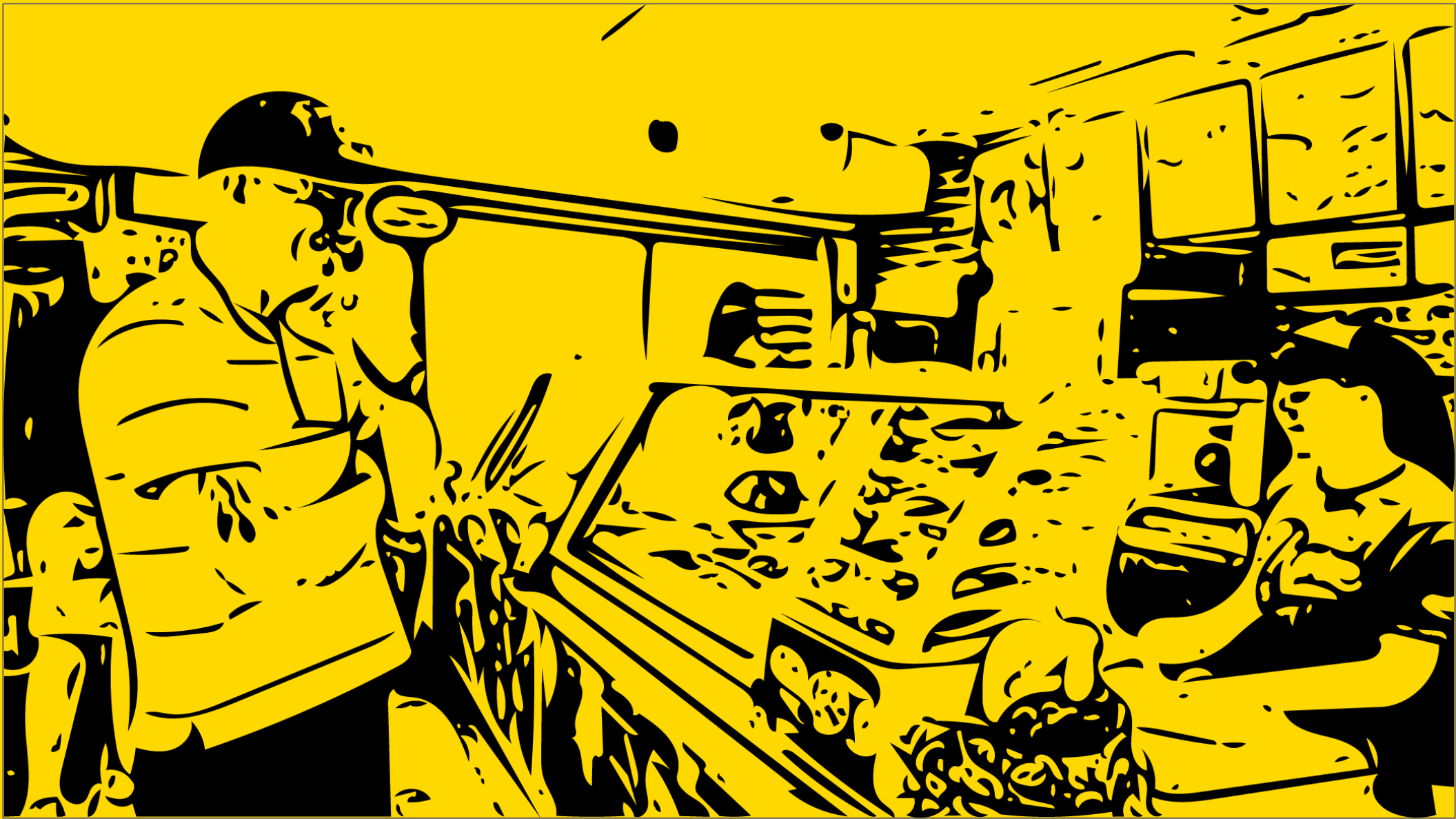
DELEGATION



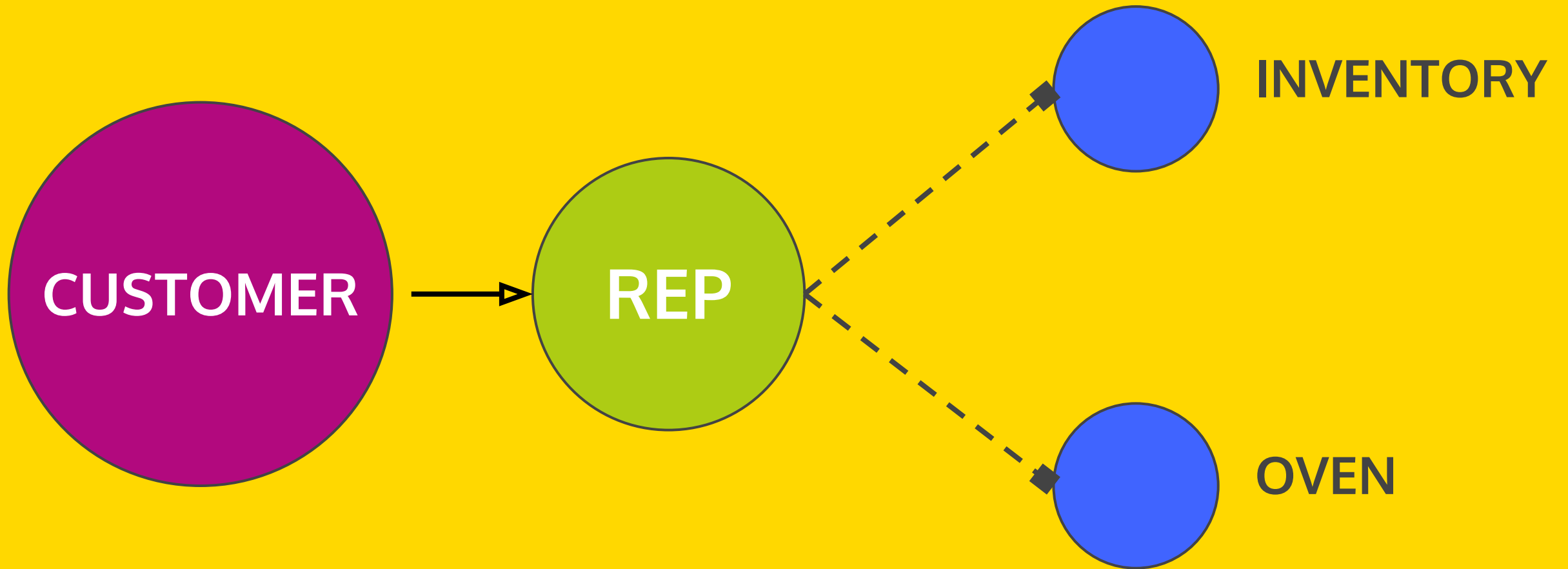
DEFINITION

**DELEGATION IS THE ACT OF
PASSING DOWN RESPONSIBILITY
TO ANOTHER OBJECT.**

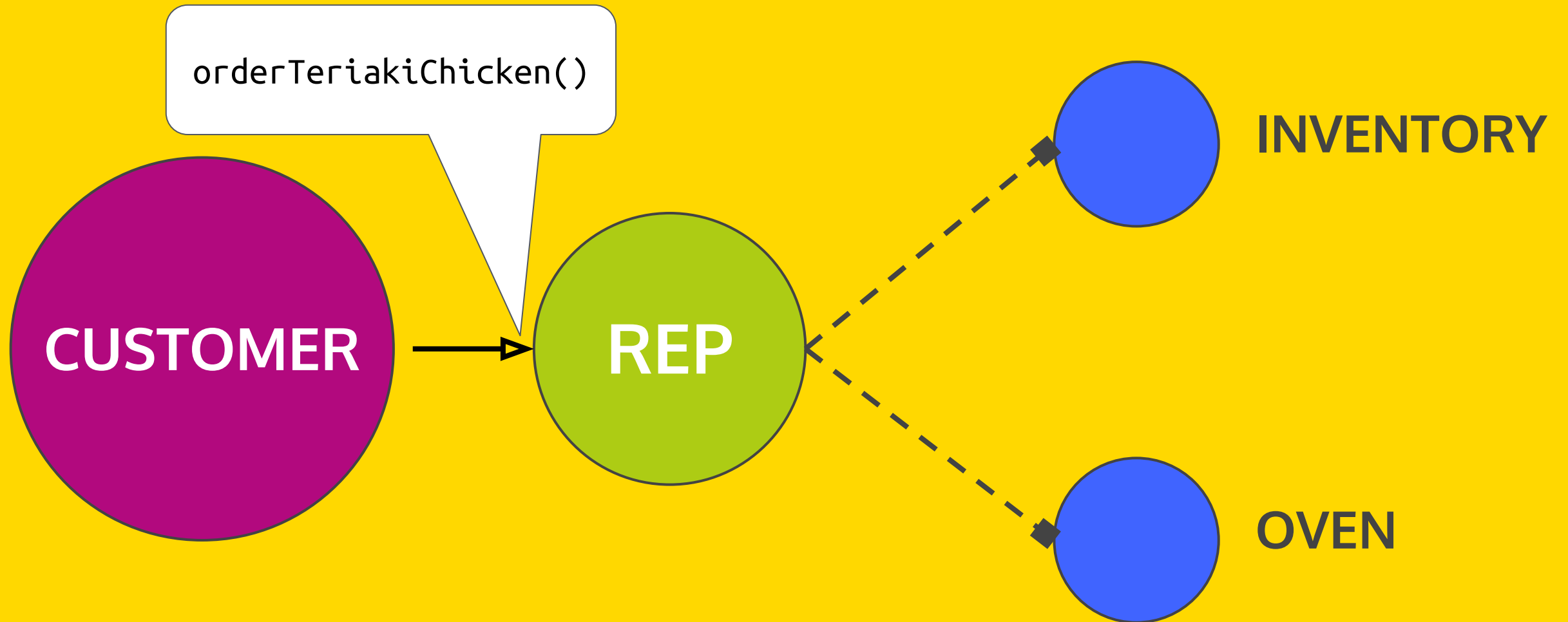
DELEGATION



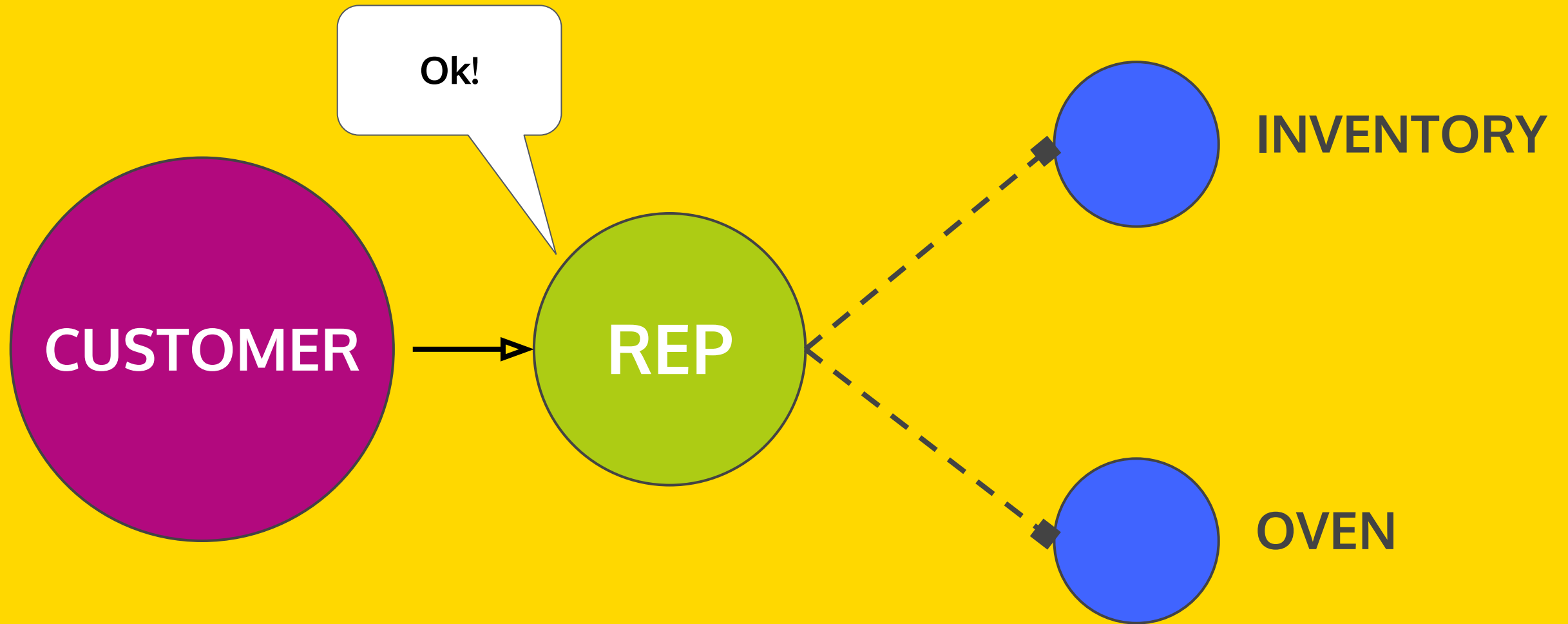
DELEGATION



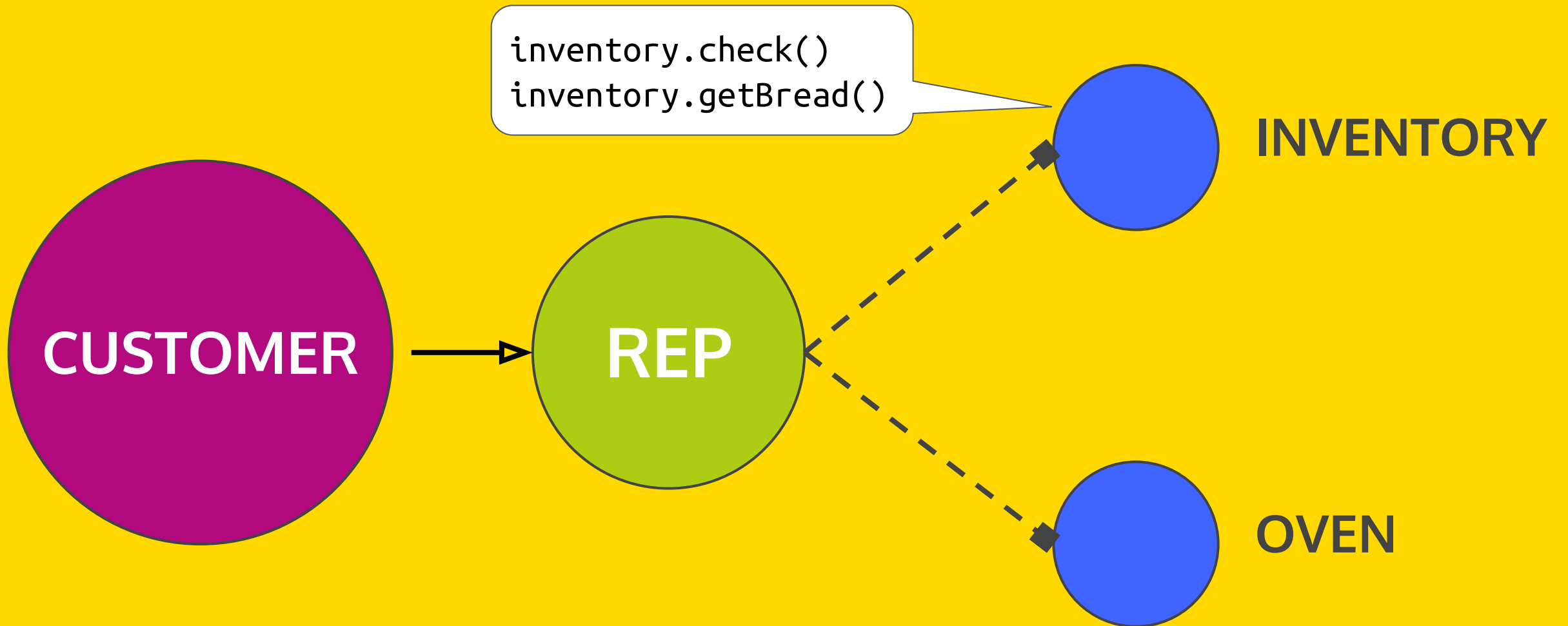
DELEGATION



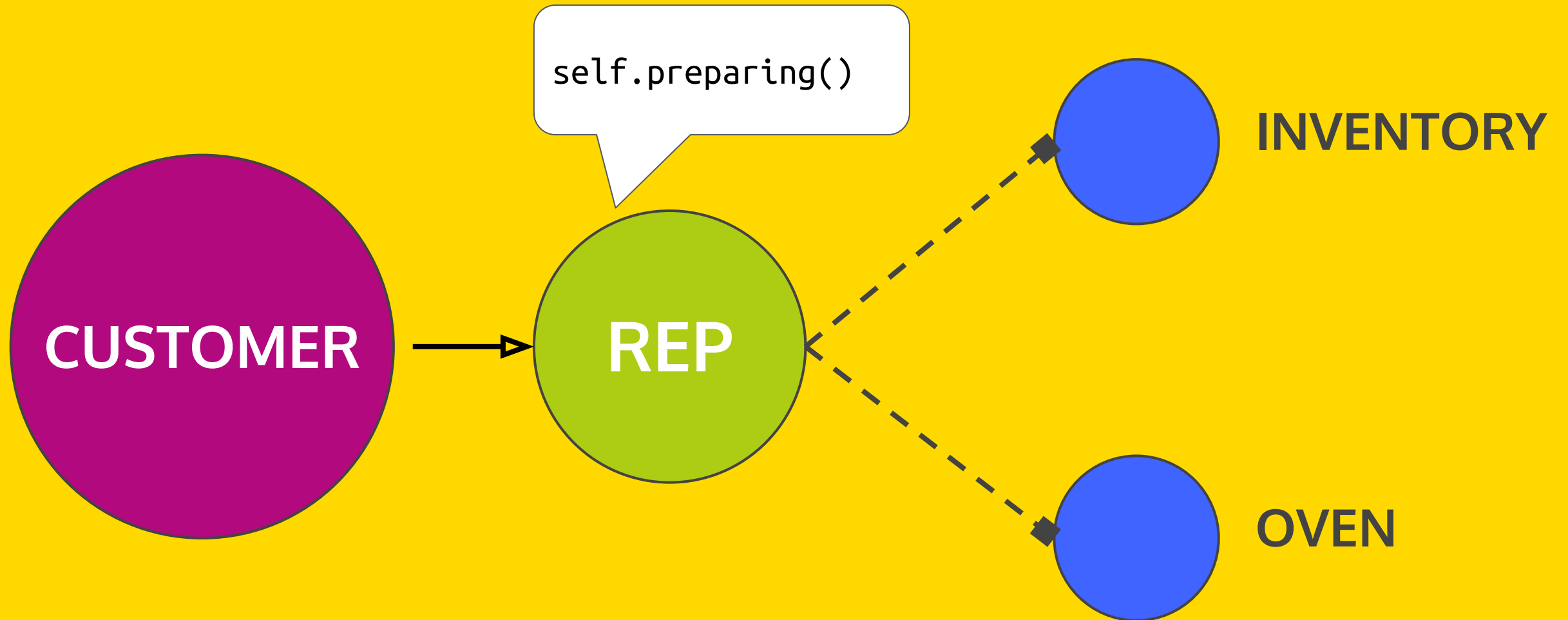
DELEGATION



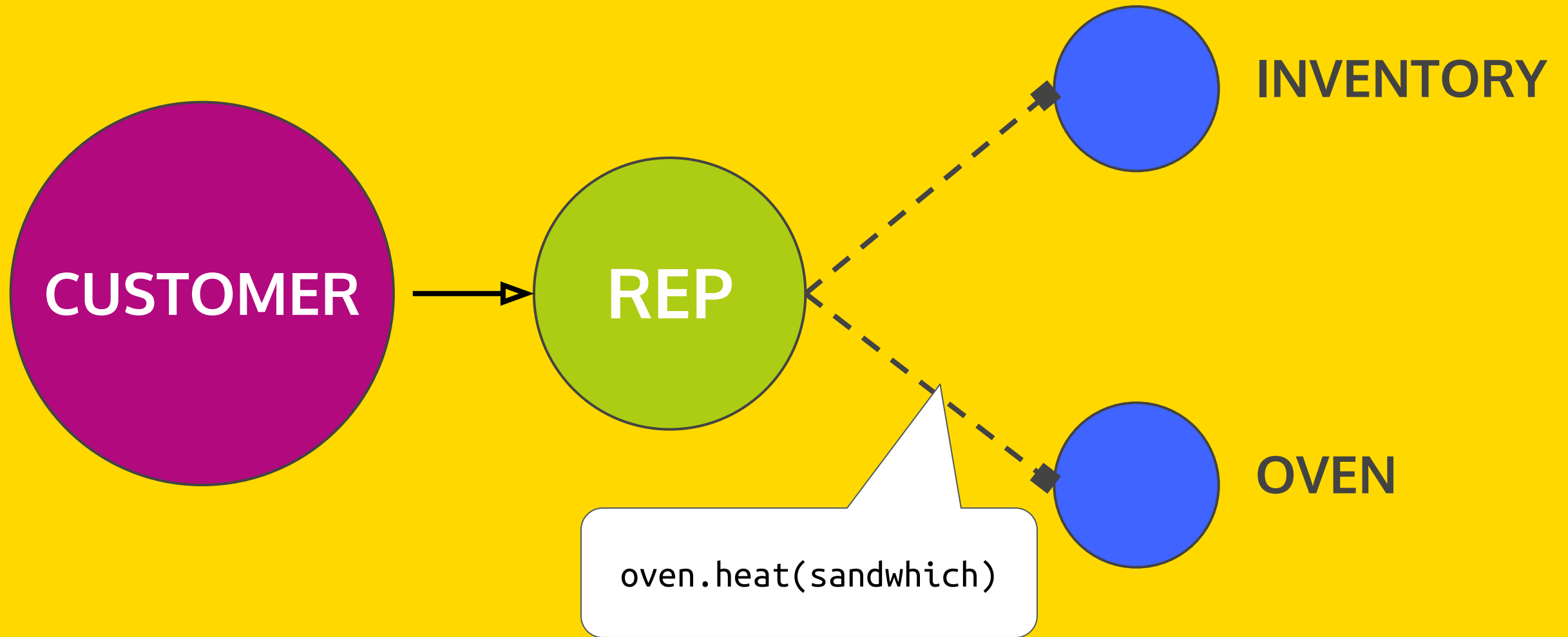
DELEGATION



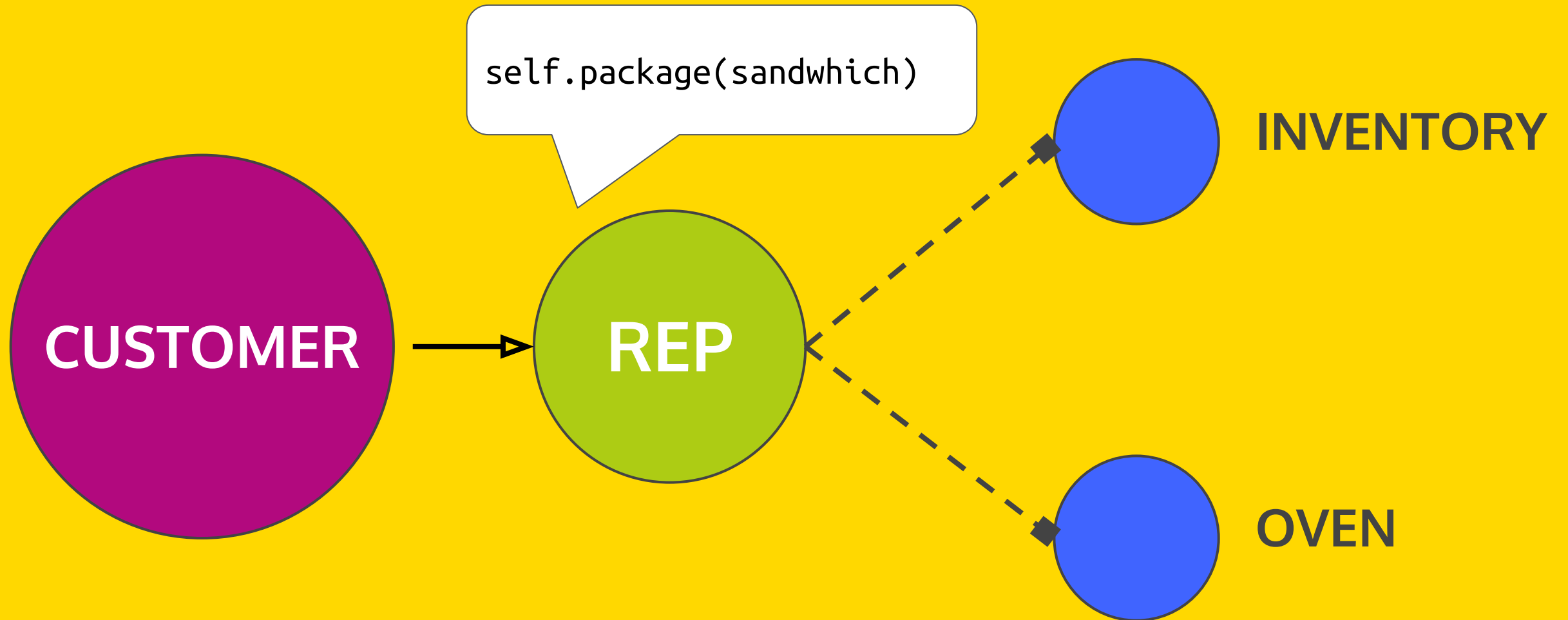
DELEGATION



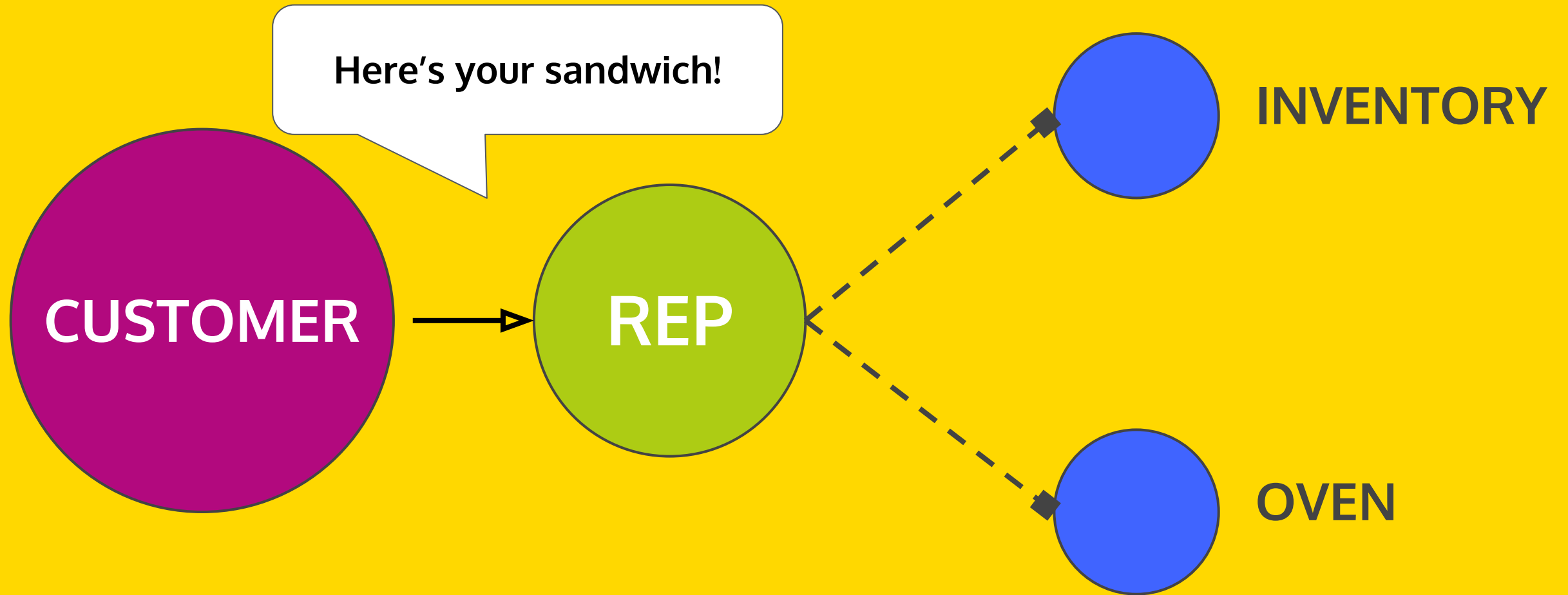
DELEGATION



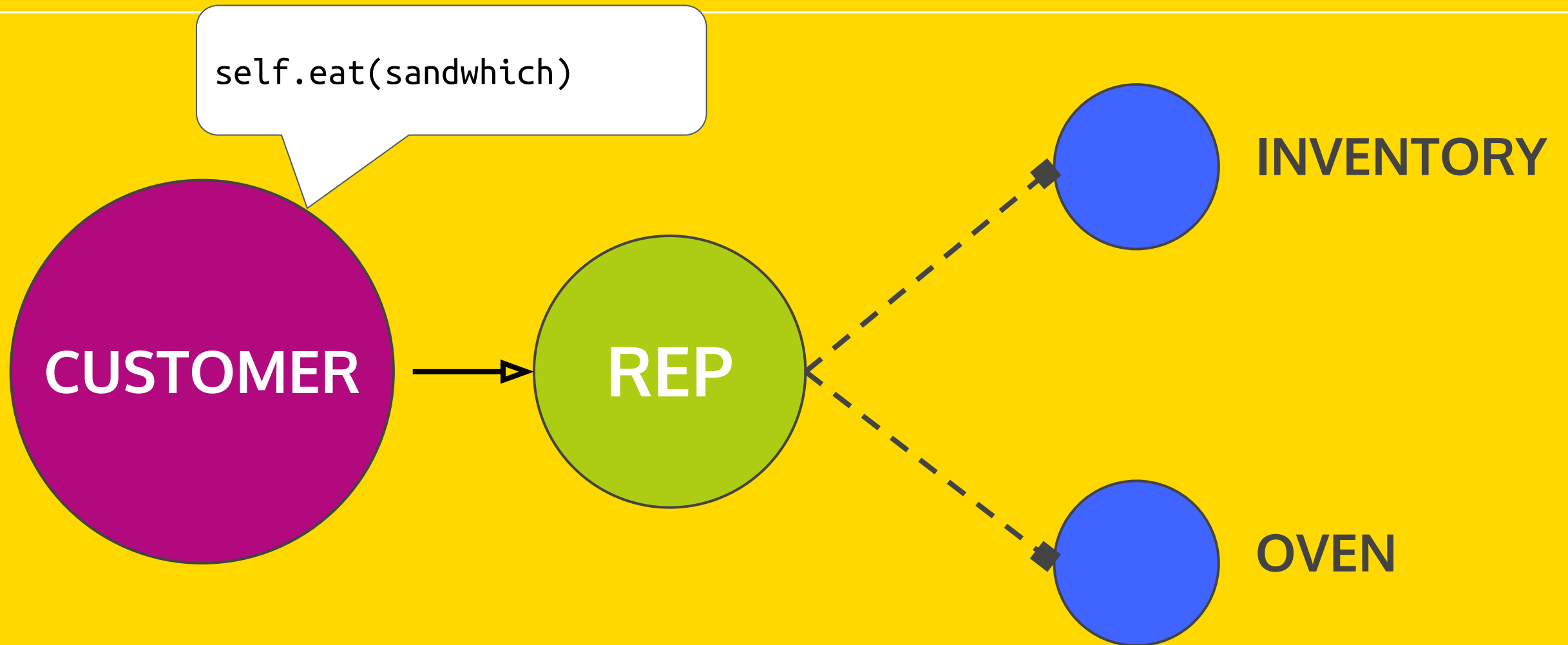
DELEGATION



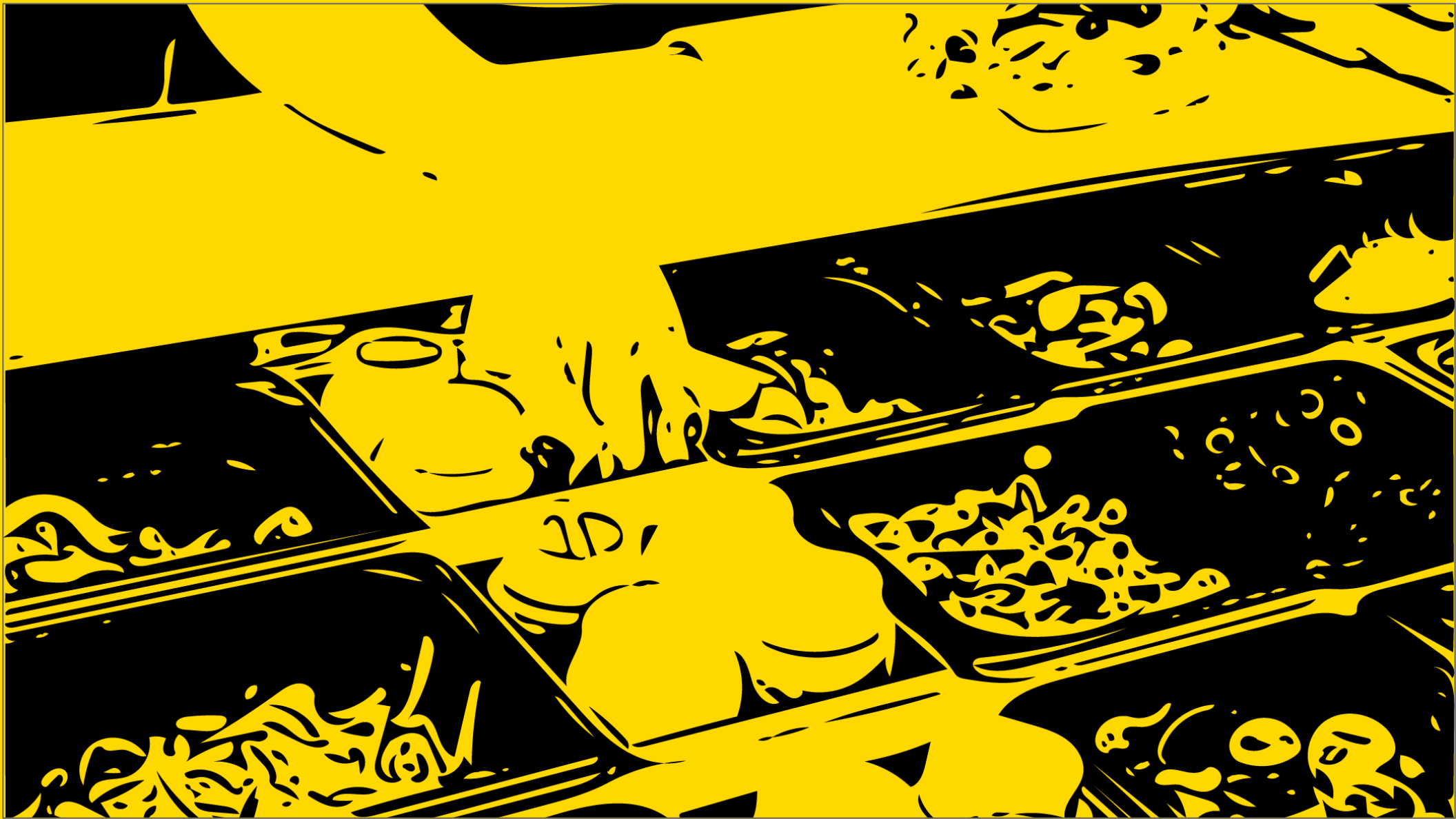
DELEGATION



DELEGATION



DELEGATION



PRACTICE: PLAYGROUNDS



CODE

15 mins

DIRECTIONS

Add the following methods to the right places

+ fillWithWater()	+ turnOn()
+ heat()	+ turnOff()
+ putTeaBag()	+ getTea()
+ putWater()	+ putTea()
+ getWater()	

NOTES

Use Delegation to make each object call the object that it is supposed to.

ACTIVITY: WRITE-PAIR-SHARE



2 mins

DIRECTIONS

Think of a subject where Object Composition make sense (for example, a car).

DELIVERABLE

Share your analogy

CONCLUSION

- + Good Design makes the code easy, fun, and intuitive to work with
- + It is an Artform that takes experience to develop
- + Part of that translation between

BEST PRACTICES

**ONE OBJECT.
ONE THING.**

BEST PRACTICES

**AVOID CIRCULAR
DEPENDENCIES.**

BEST PRACTICES

USE ABSTRACTIONS.

THANKS!

WELLINGTON MORENO

- + GitHub/Slack: **@SirWellington**
- + Twitter: **@SirWellingtonZ**
- + Email: **wellington.moreno@ga.co**