The Four Pillars

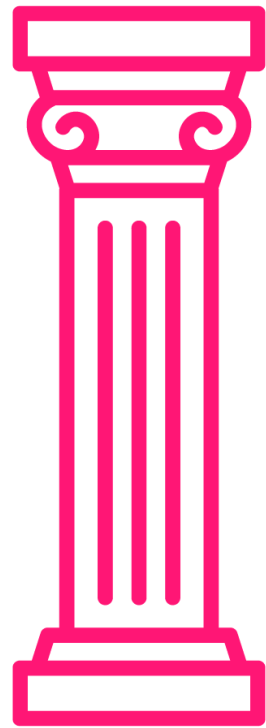**Mel Grubb**

Developer

@melgrubb | www.melgrubb.com

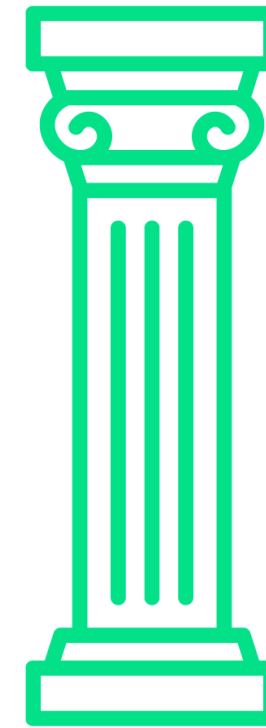# The Four Pillars

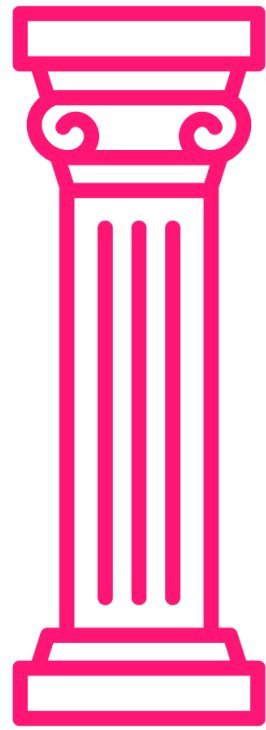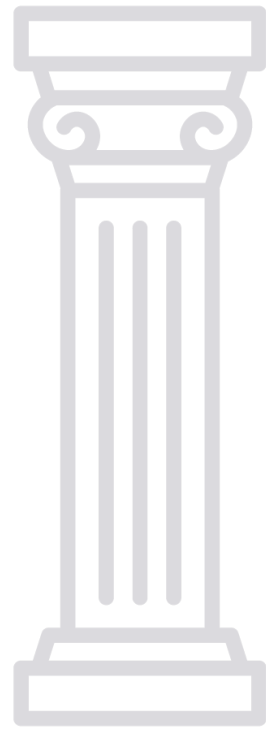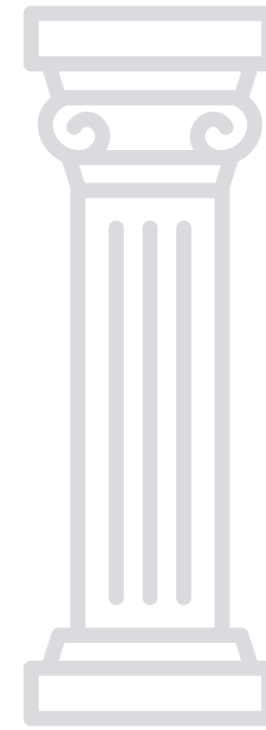**Abstraction**   **Encapsulation**   **Inheritance**   **Polymorphism**
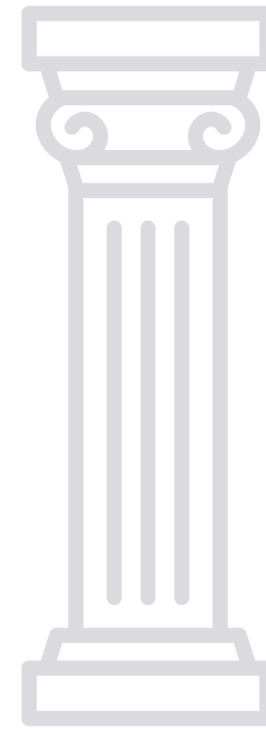
The Four Pillars
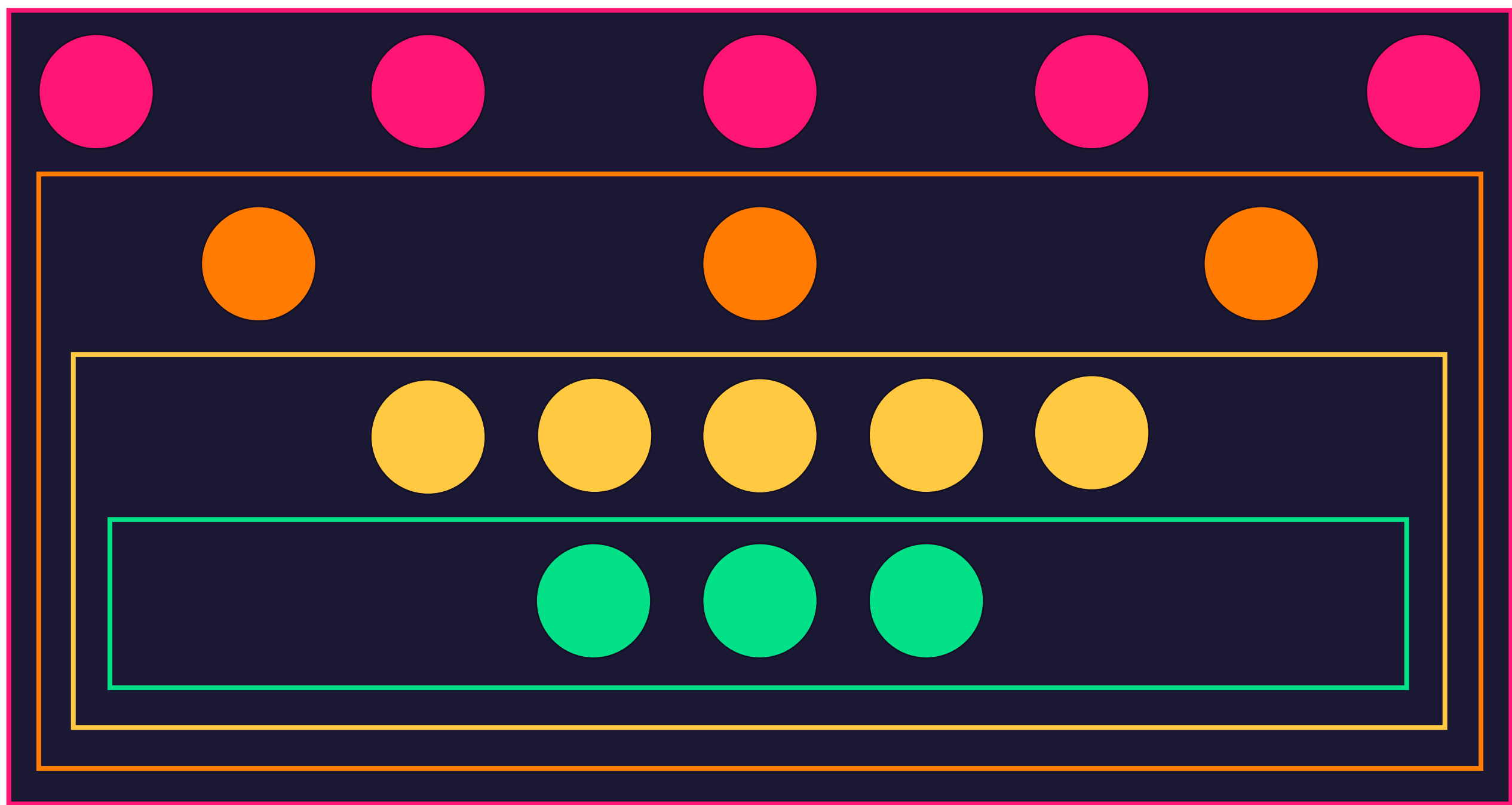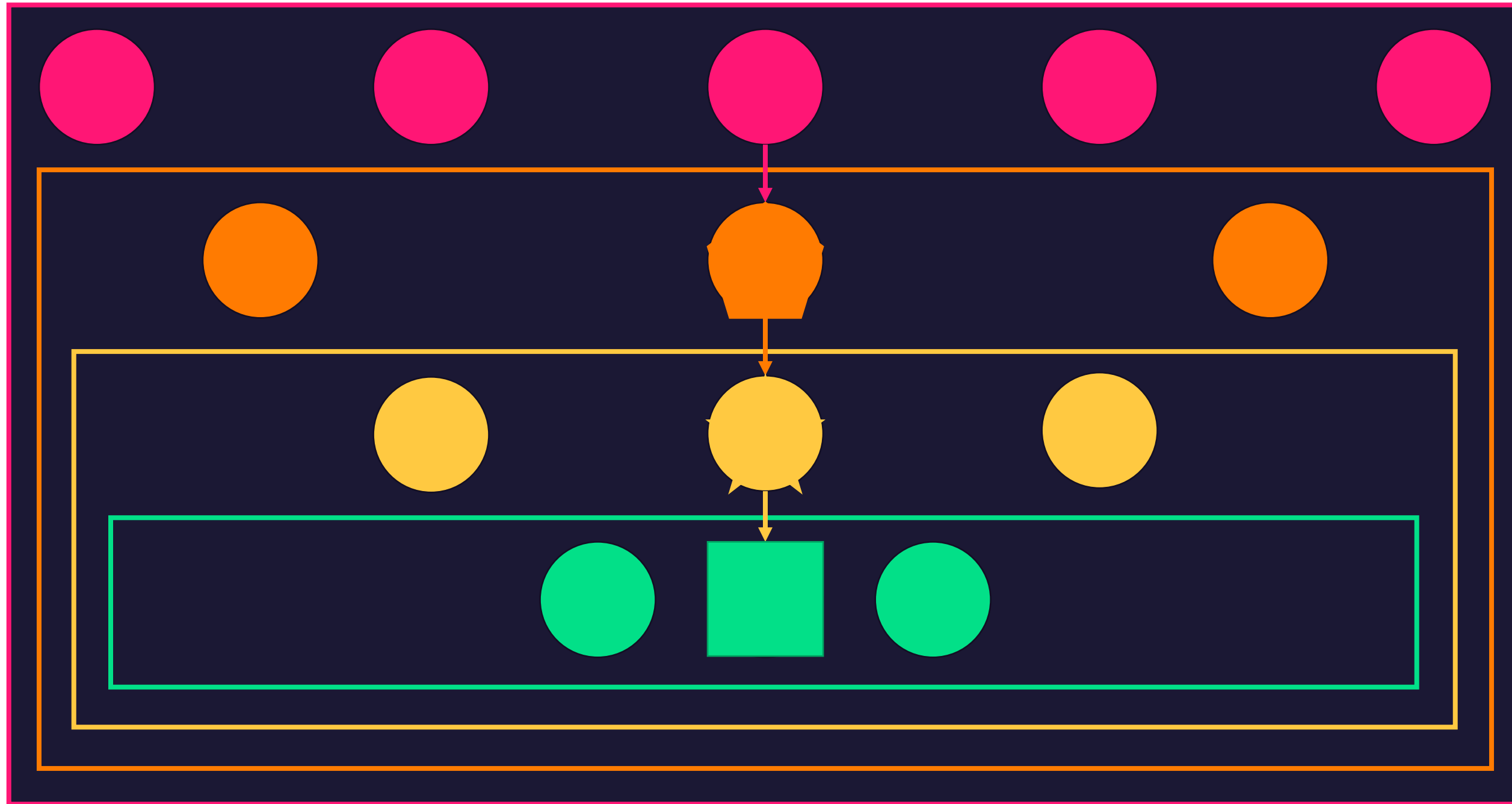
Abstraction Encapsulation Inheritance Polymorphism

Module – A grouping of related code

Abstraction – Hiding implementation details

"interface" (C# Keyword)

Defines members that must be present on any non-abstract class that implements the interface.
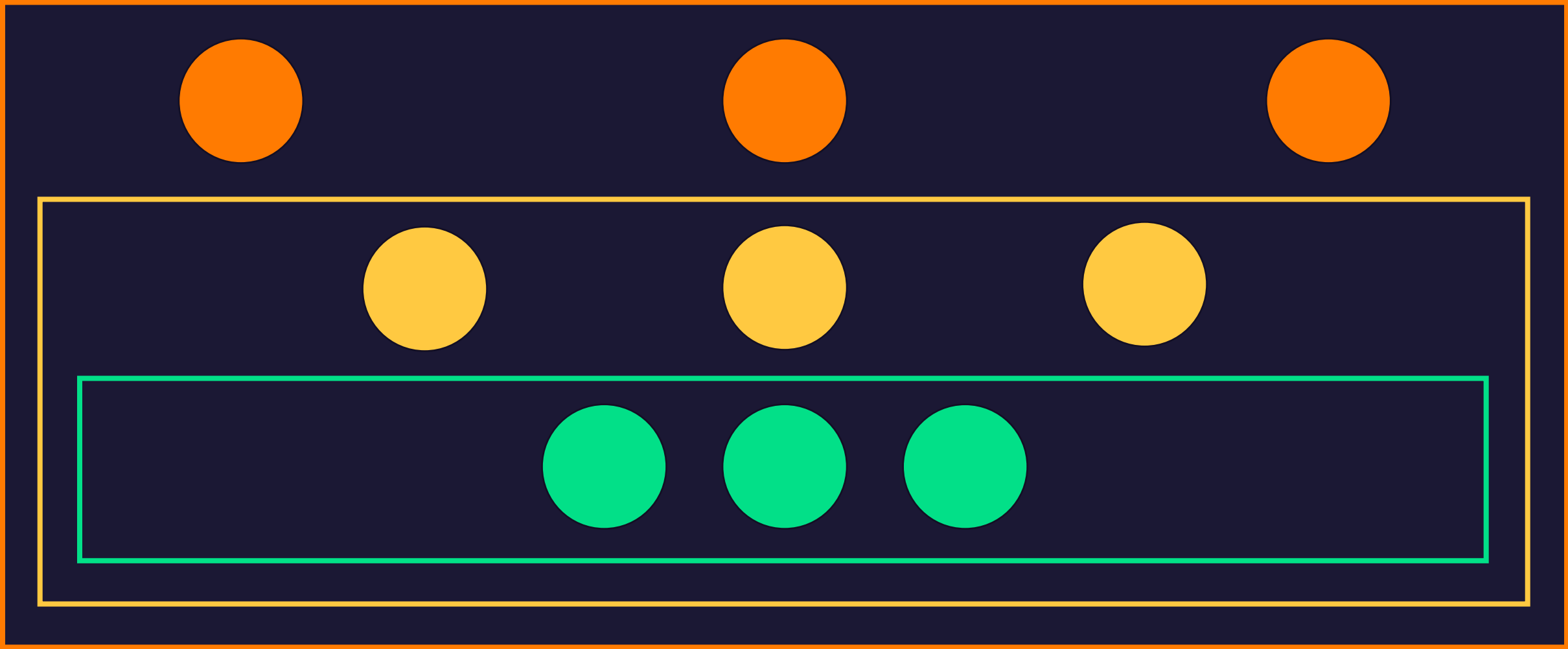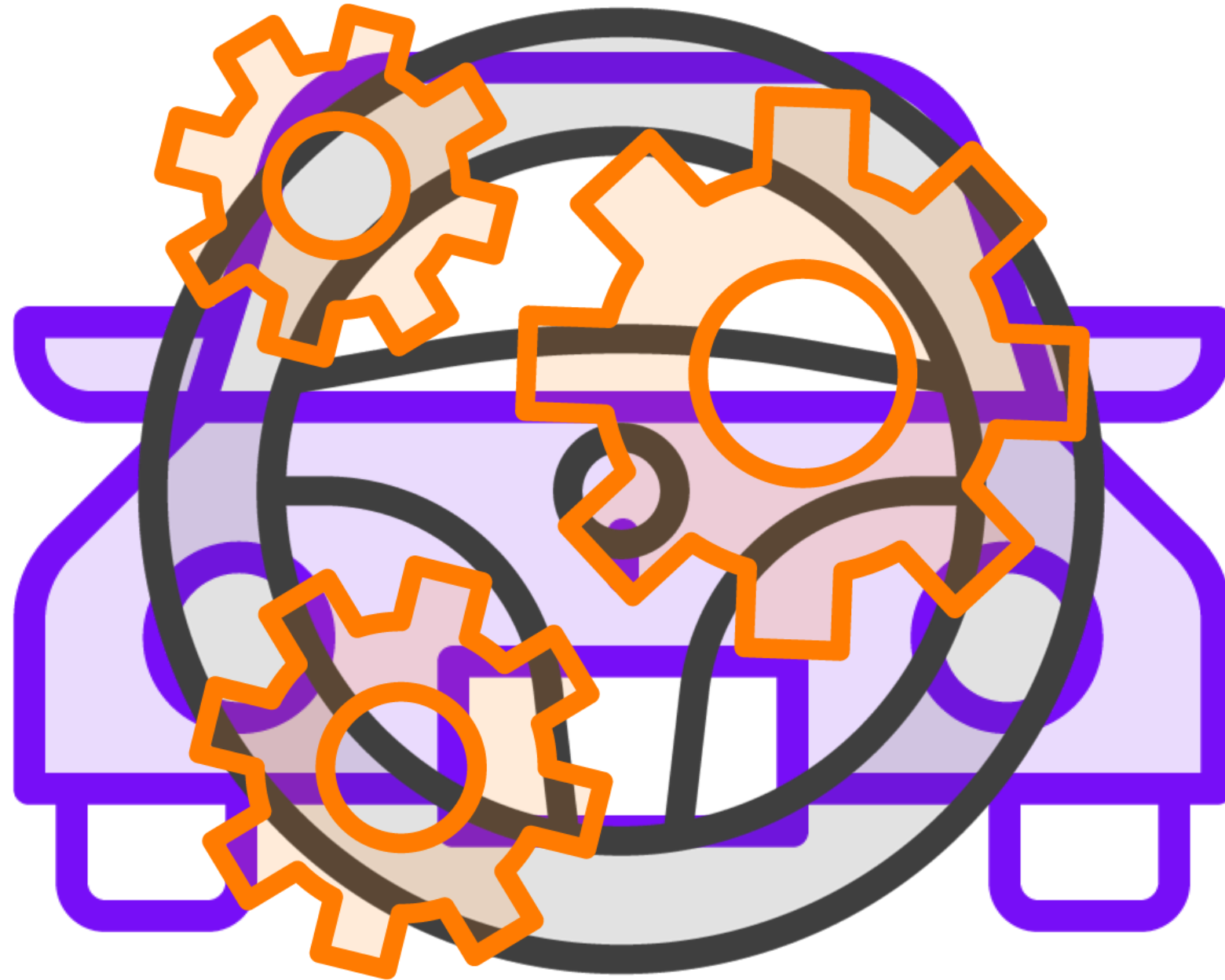
"Interface" (General Concept)

The externally visible members of a module.

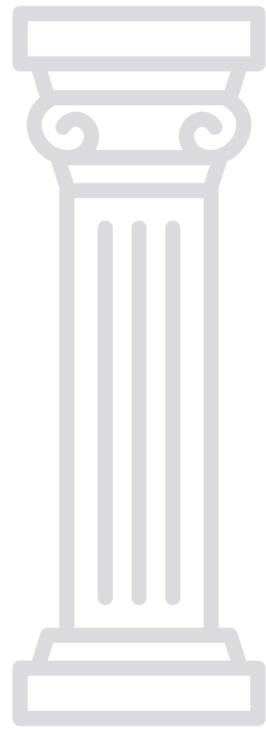# Interfaces

Abstraction

Abstraction

Design Tip

- Look at your modules' surface area
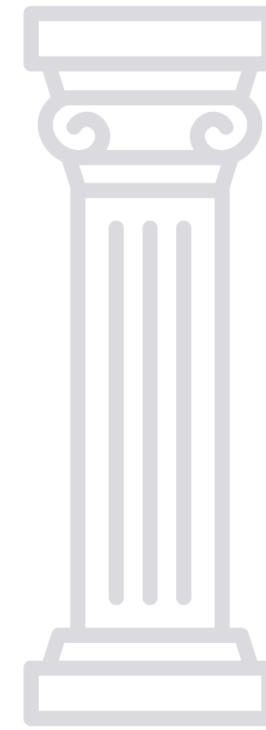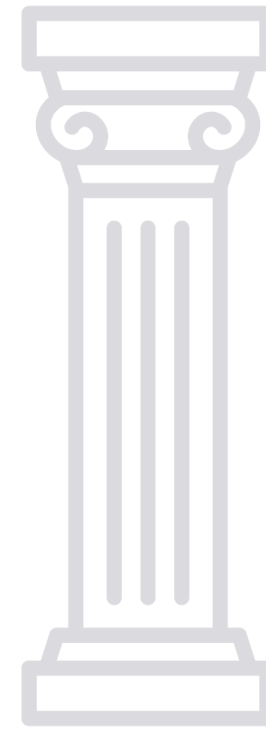- Minimize visibility
- Visibility = Contract
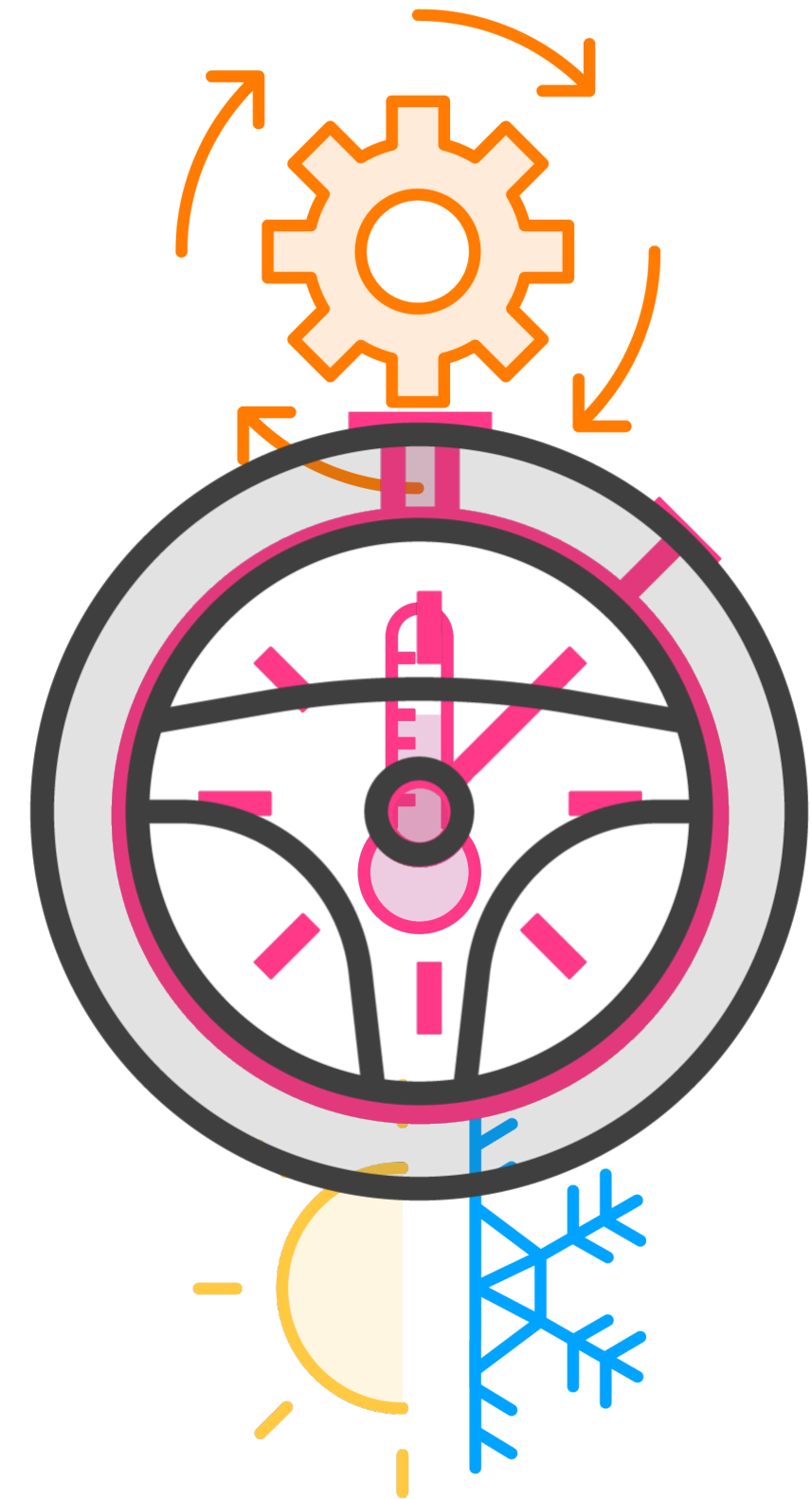
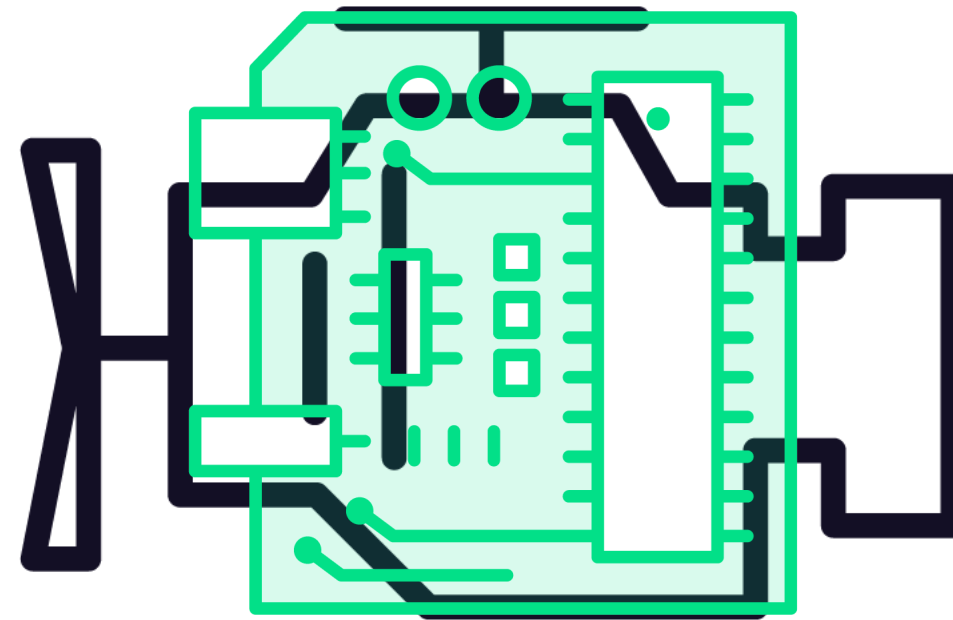# The Four Pillars

Abstraction

Encapsulation

Inheritance

Polymorphism
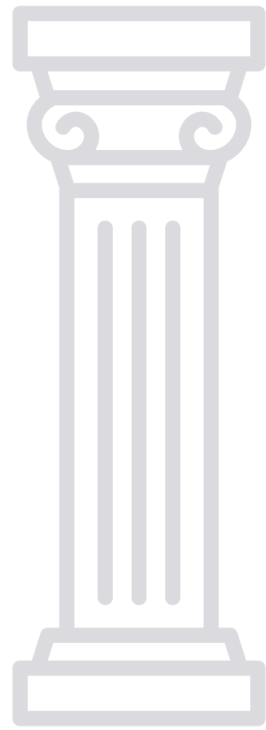
Encapsulation

Composition
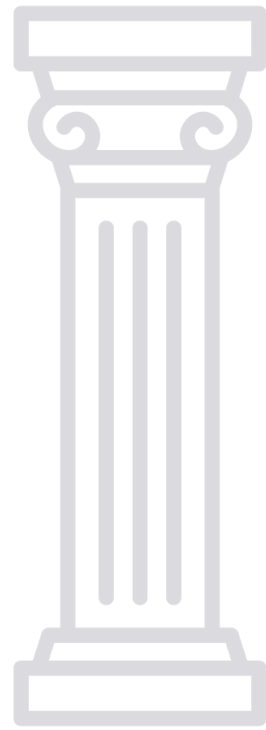
Design Tip

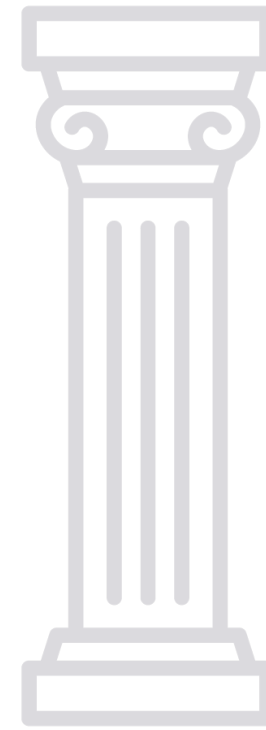- Keep data close to its use
- Minimize visibility

The Four Pillars

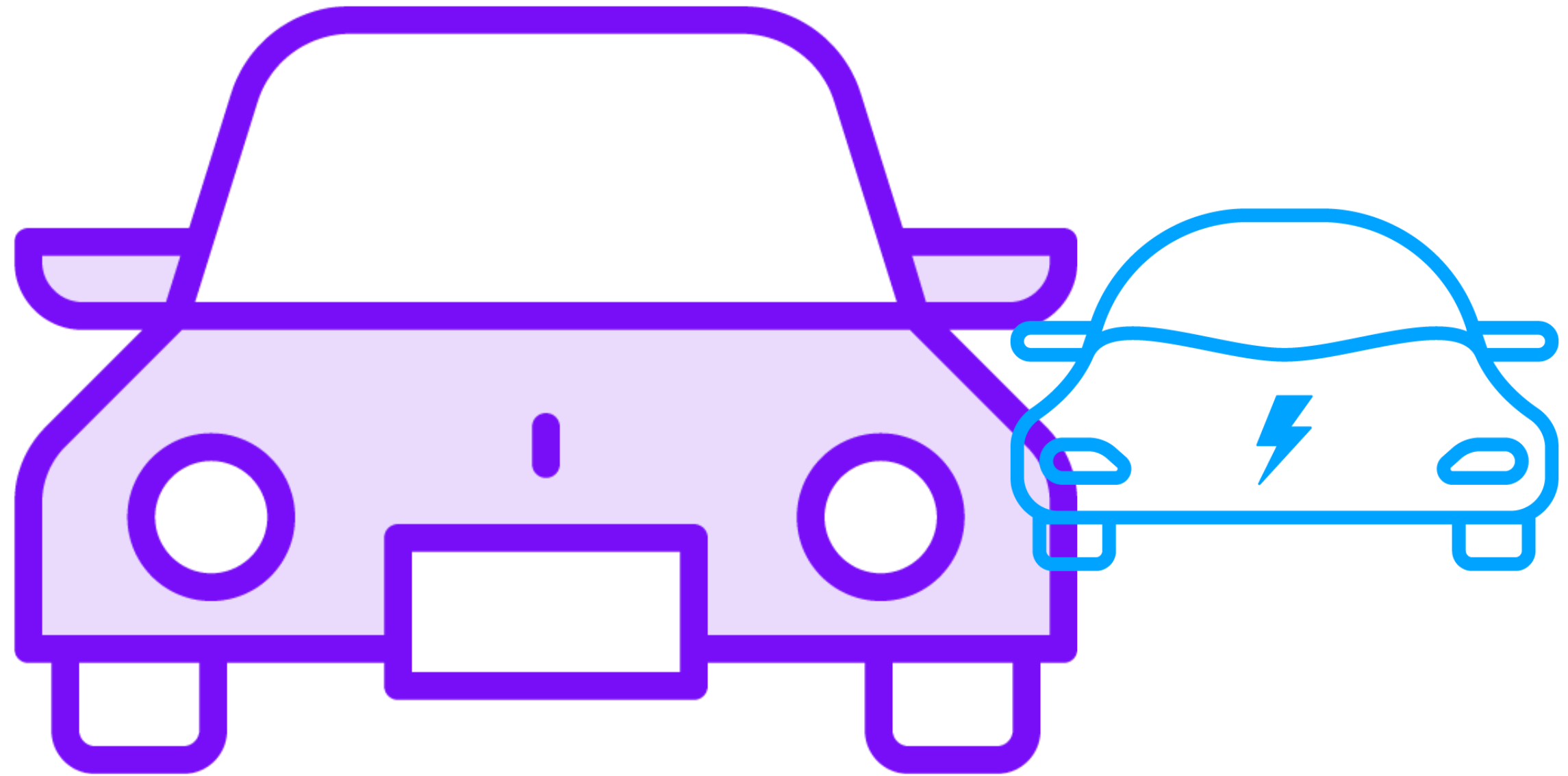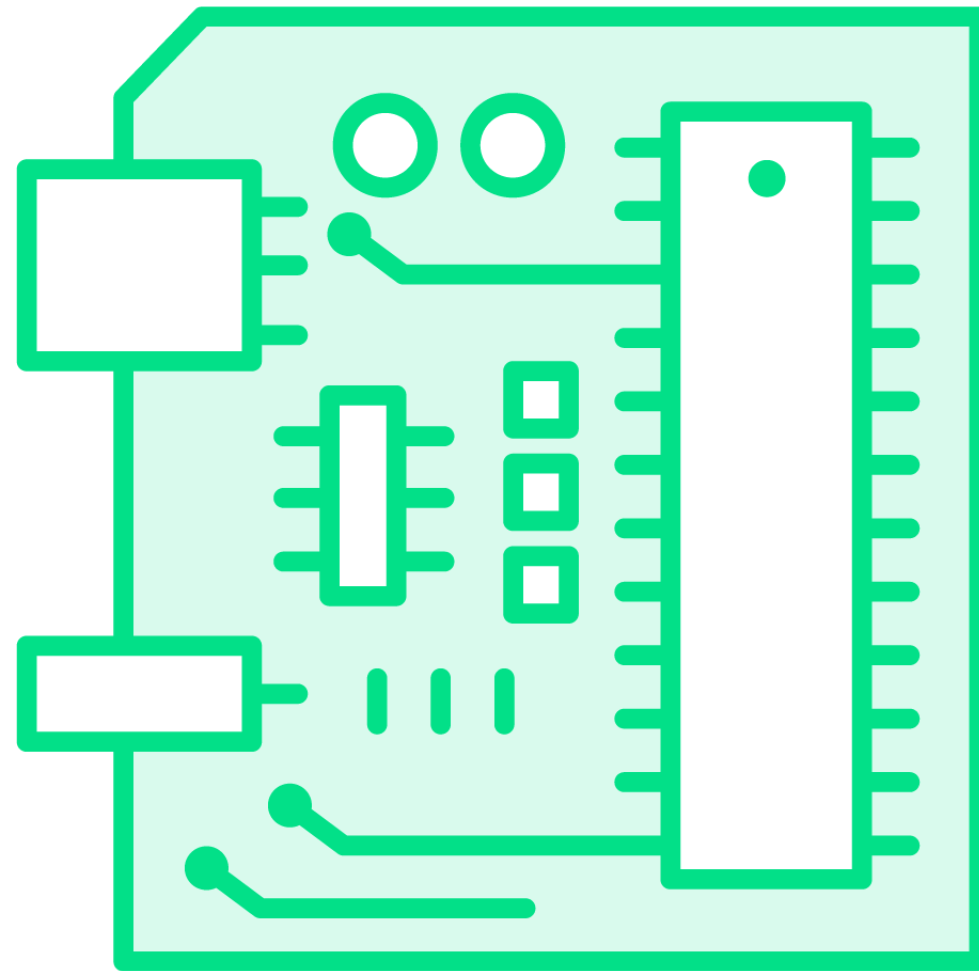Abstraction   Encapsulation   **Inheritance**   Polymorphism

Inheritance

Inheritance

ICE Control Module

Electric Control Module

Design Tip

- Extract commonalities to a base class
- Leave type-specific code in type
- Avoid "multiple personality" code in base class
- Base class should not know descendants

The Four Pillars

Abstraction
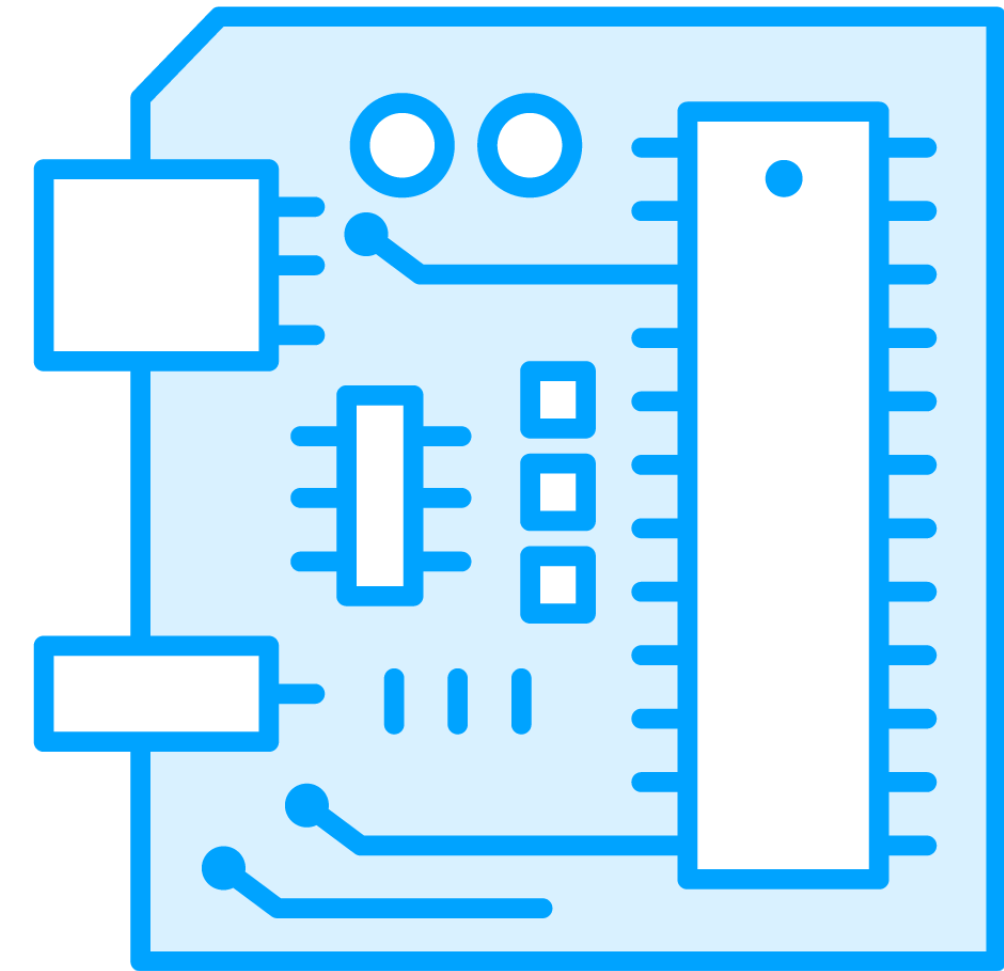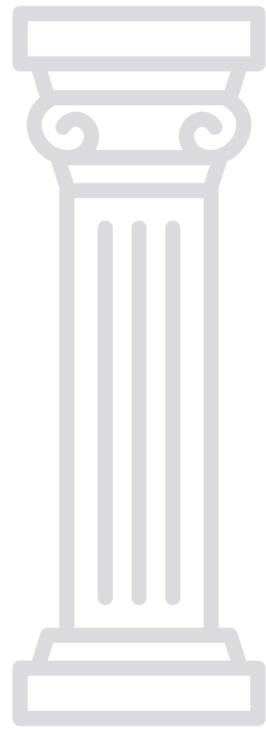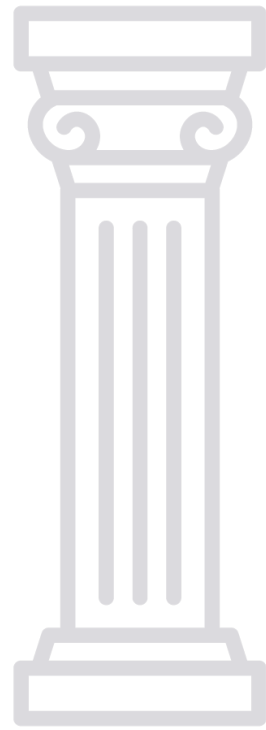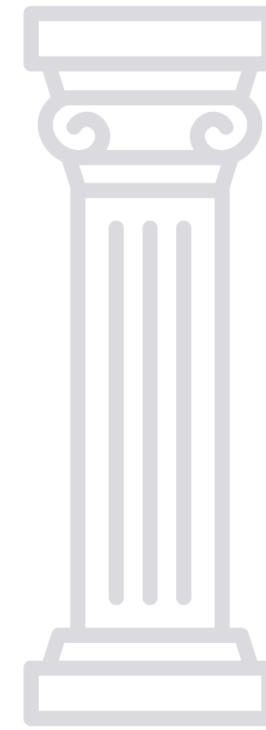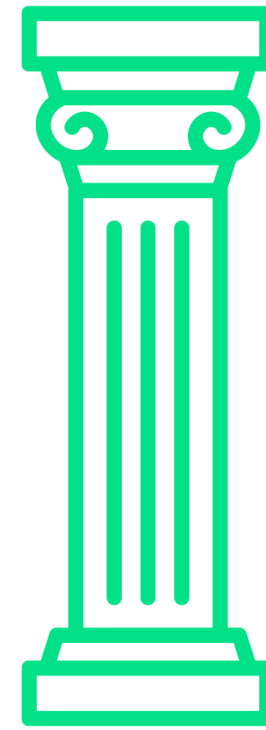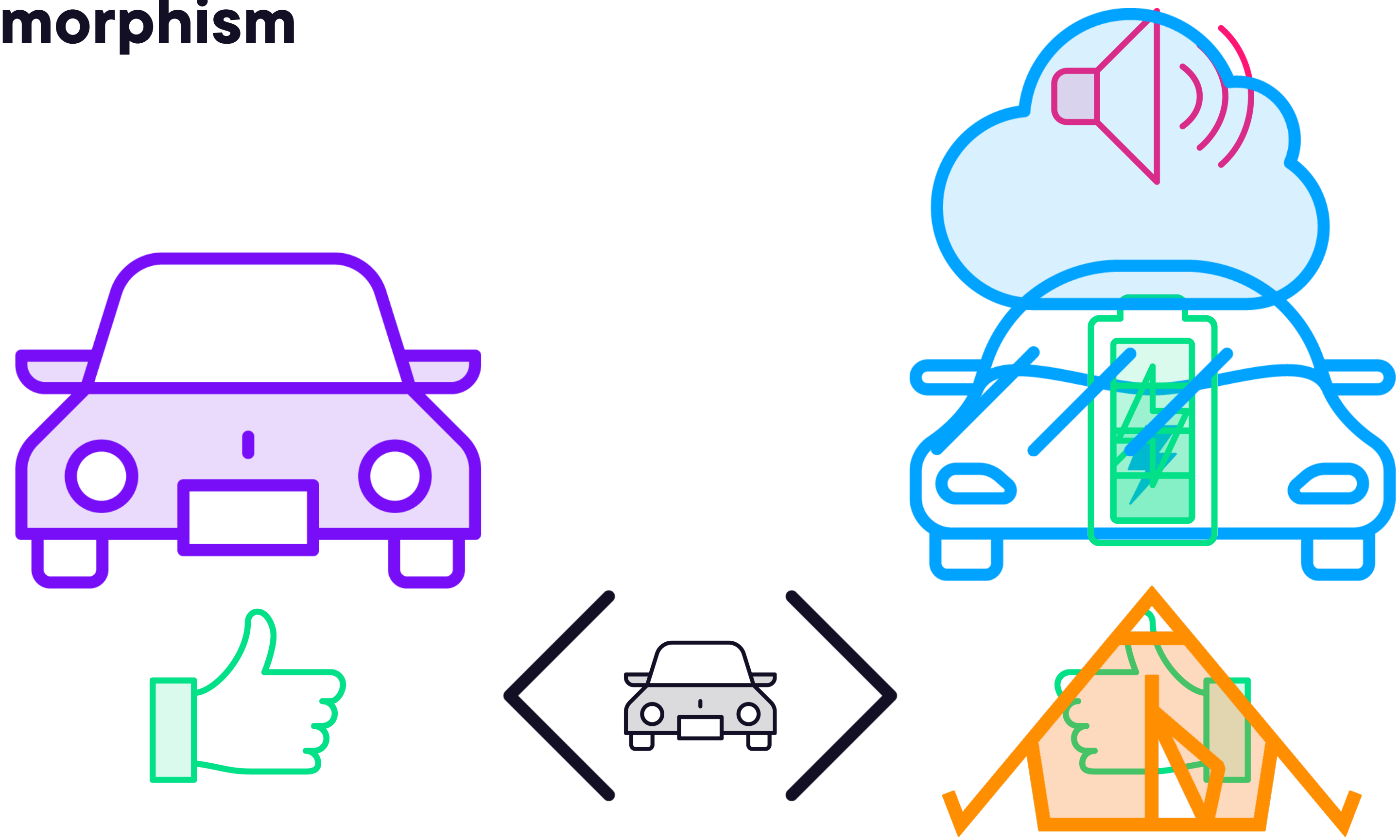
Encapsulation

Inheritance

Polymorphism

Polymorphism

Design Tip

- Divide up responsibilities
- Use inheritance for universal abilities
- Use interfaces for mix & match

Summary

Abstraction

- Hide complexity

Encapsulation

- Hide or move supporting data

Inheritance

- Extract commonalities

Polymorphism

- Objects can have multiple roles
- Mutiple objects can have the same role

Up Next:

The SOLID Principles