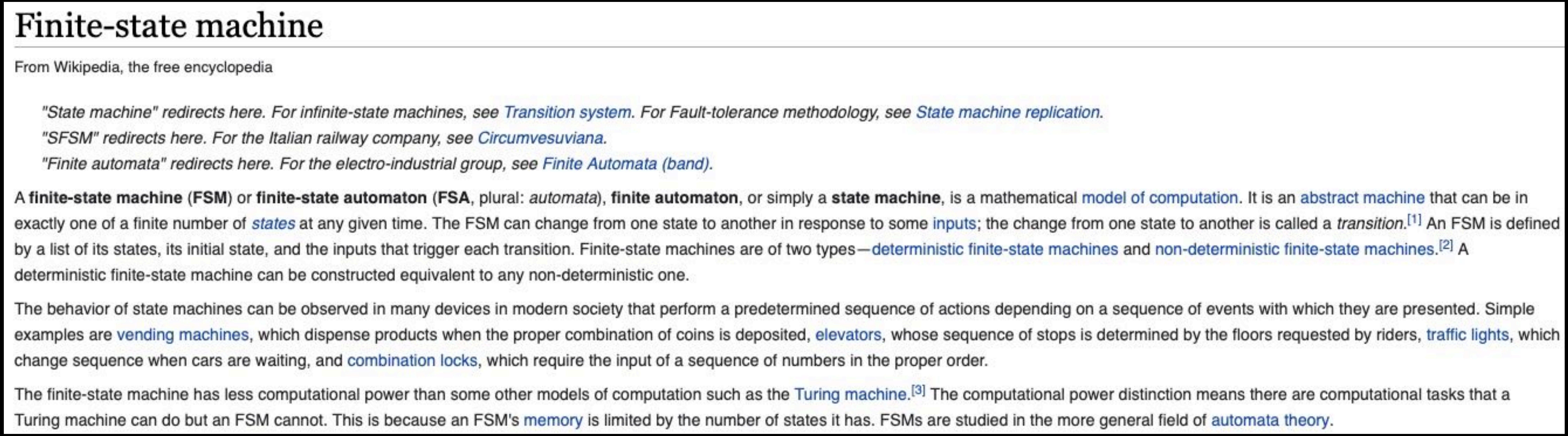
“State Machine”

Improving your code with State Machine

Jason Elian - iOS Developer, Core Squad at LinkAja

What is a State Machine?

What is a State Machine?



Finite-state machine

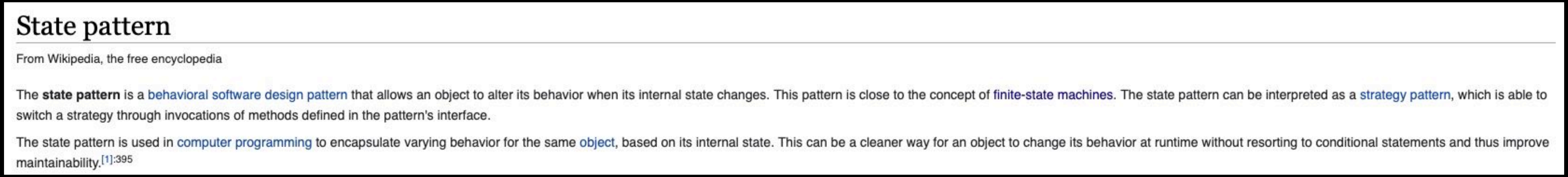
From Wikipedia, the free encyclopedia

"State machine" redirects here. For infinite-state machines, see Transition system. For Fault-tolerance methodology, see State machine replication.
"SFSM" redirects here. For the Italian railway company, see Circumvesuviana.
"Finite automata" redirects here. For the electro-industrial group, see Finite Automata (band).

A **finite-state machine** (**FSM**) or **finite-state automaton** (**FSA**, plural: *automata*), **finite automaton**, or simply a **state machine**, is a mathematical model of computation. It is an abstract machine that can be in exactly one of a finite number of *states* at any given time. The FSM can change from one state to another in response to some inputs; the change from one state to another is called a *transition*.[1] An FSM is defined by a list of its states, its initial state, and the inputs that trigger each transition. Finite-state machines are of two types—deterministic finite-state machines and non-deterministic finite-state machines.[2] A deterministic finite-state machine can be constructed equivalent to any non-deterministic one.

The behavior of state machines can be observed in many devices in modern society that perform a predetermined sequence of actions depending on a sequence of events with which they are presented. Simple examples are vending machines, which dispense products when the proper combination of coins is deposited, elevators, whose sequence of stops is determined by the floors requested by riders, traffic lights, which change sequence when cars are waiting, and combination locks, which require the input of a sequence of numbers in the proper order.

The finite-state machine has less computational power than some other models of computation such as the Turing machine.[3] The computational power distinction means there are computational tasks that a Turing machine can do but an FSM cannot. This is because an FSM's memory is limited by the number of states it has. FSMs are studied in the more general field of automata theory.



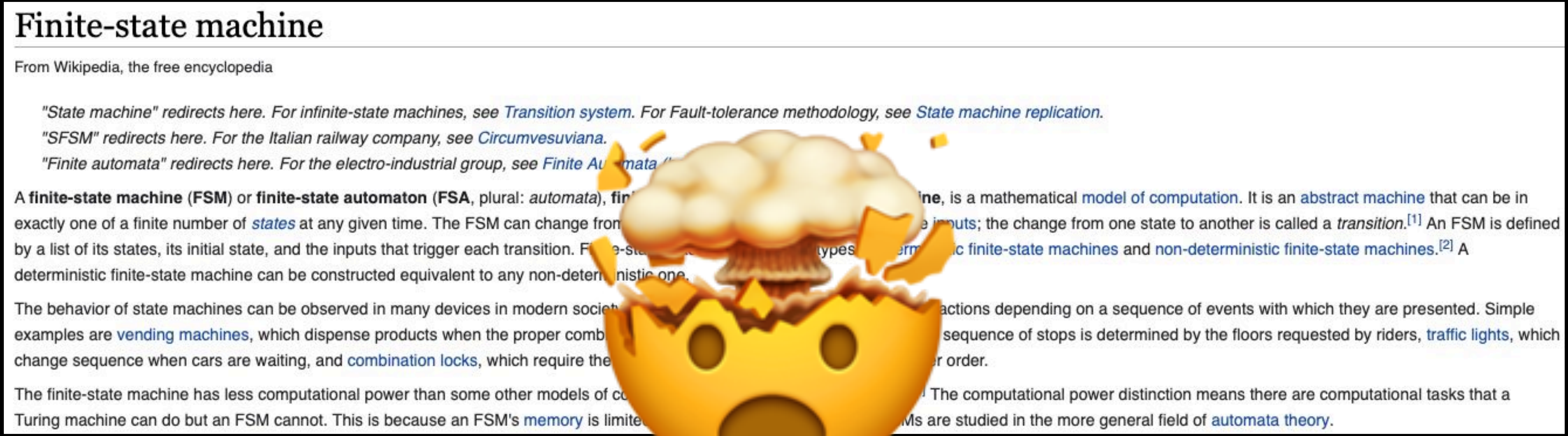
State pattern

From Wikipedia, the free encyclopedia

The **state pattern** is a behavioral software design pattern that allows an object to alter its behavior when its internal state changes. This pattern is close to the concept of finite-state machines. The state pattern can be interpreted as a strategy pattern, which is able to switch a strategy through invocations of methods defined in the pattern's interface.

The state pattern is used in computer programming to encapsulate varying behavior for the same object, based on its internal state. This can be a cleaner way for an object to change its behavior at runtime without resorting to conditional statements and thus improve maintainability.[1]:395

What is a State Machine?



Finite-state machine

From Wikipedia, the free encyclopedia

"State machine" redirects here. For infinite-state machines, see Transition system. For Fault-tolerance methodology, see State machine replication.

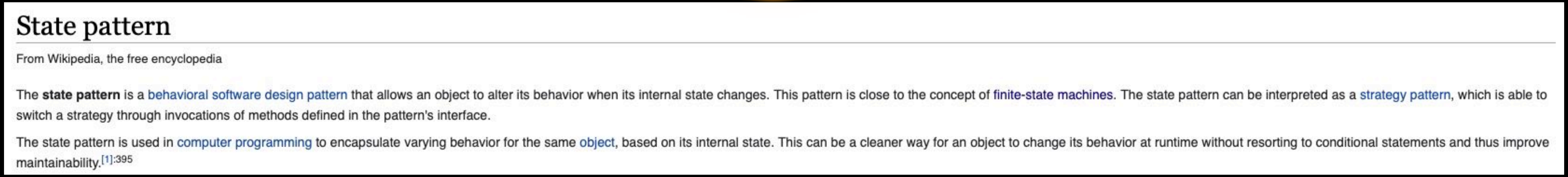
"SFSM" redirects here. For the Italian railway company, see Circumvesuviana.

"Finite automata" redirects here. For the electro-industrial group, see Finite Au mata (

A **finite-state machine** (**FSM**) or **finite-state automaton** (**FSA**, plural: *automata*), **fi** **ne**, is a mathematical model of computation. It is an abstract machine that can be in exactly one of a finite number of *states* at any given time. The FSM can change fron inputs; the change from one state to another is called a *transition*.[1] An FSM is defined by a list of its states, its initial state, and the inputs that trigger each transition. F e-st types erm c finite-state machines and non-deterministic finite-state machines.[2] A deterministic finite-state machine can be constructed equivalent to any non-deter nistic one.

The behavior of state machines can be observed in many devices in modern socie actions depending on a sequence of events with which they are presented. Simple examples are vending machines, which dispense products when the proper comb sequence of stops is determined by the floors requested by riders, traffic lights, which change sequence when cars are waiting, and combination locks, which require the r order.

The finite-state machine has less computational power than some other models of co The computational power distinction means there are computational tasks that a Turing machine can do but an FSM cannot. This is because an FSM's memory is limite Ms are studied in the more general field of automata theory.



State pattern

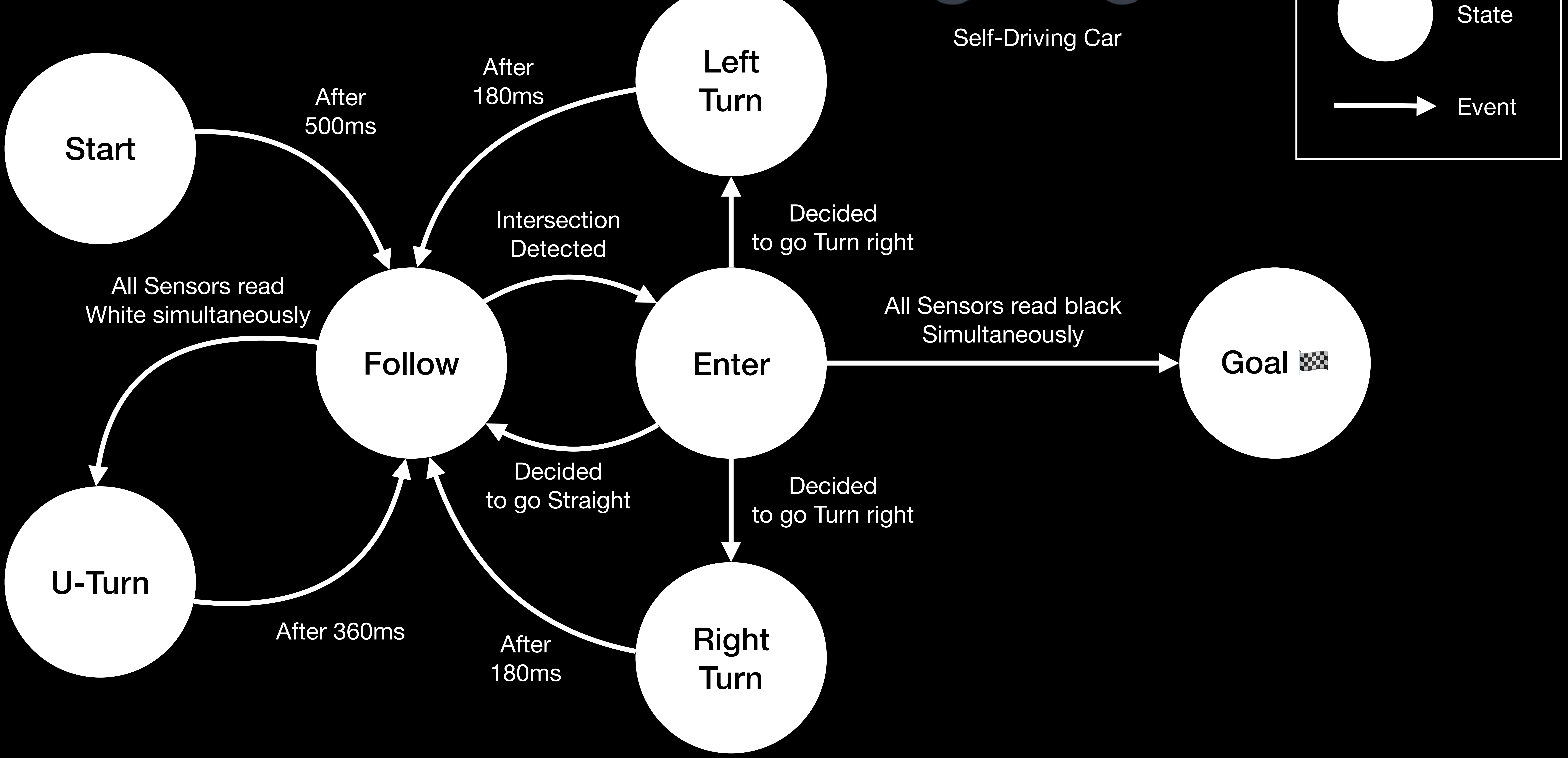
From Wikipedia, the free encyclopedia

The **state pattern** is a behavioral software design pattern that allows an object to alter its behavior when its internal state changes. This pattern is close to the concept of finite-state machines. The state pattern can be interpreted as a strategy pattern, which is able to switch a strategy through invocations of methods defined in the pattern's interface.

The state pattern is used in computer programming to encapsulate varying behavior for the same object, based on its internal state. This can be a cleaner way for an object to change its behavior at runtime without resorting to conditional statements and thus improve maintainability.[1]:395

What is a State Machine?

Self-Driving Car



Legend

State

Event

Start

After
500ms

Follow

After
180ms

Left
Turn

Intersection
Detected

Enter

Decided
to go Turn right

All Sensors read
White simultaneously

All Sensors read black
Simultaneously

Goal 🏁

Decided
to go Straight

Decided
to go Turn right

U-Turn

After 360ms

After
180ms

Right
Turn

What is a State Machine?

A list of the finite states that a machine has.
The states determine how the behavior should be.

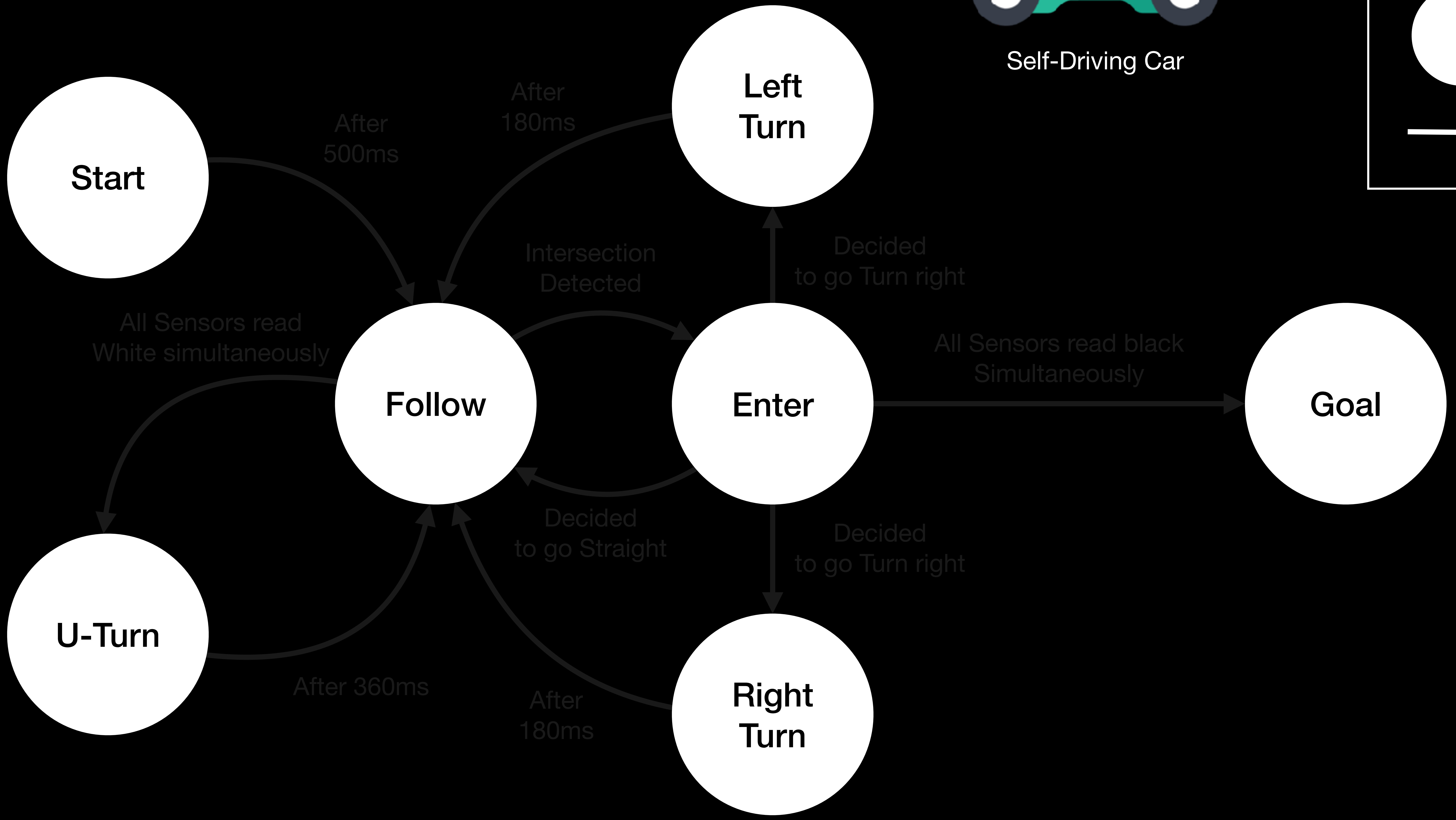
What is a State Machine?

Self-Driving Car

Legend

● State

→ Event



Start

After 500ms

Follow

After 180ms

Left Turn

Intersection Detected

Enter

Decided to go Turn right

All Sensors read White simultaneously

All Sensors read black Simultaneously

Goal

Decided to go Straight

Decided to go Turn right

Right Turn

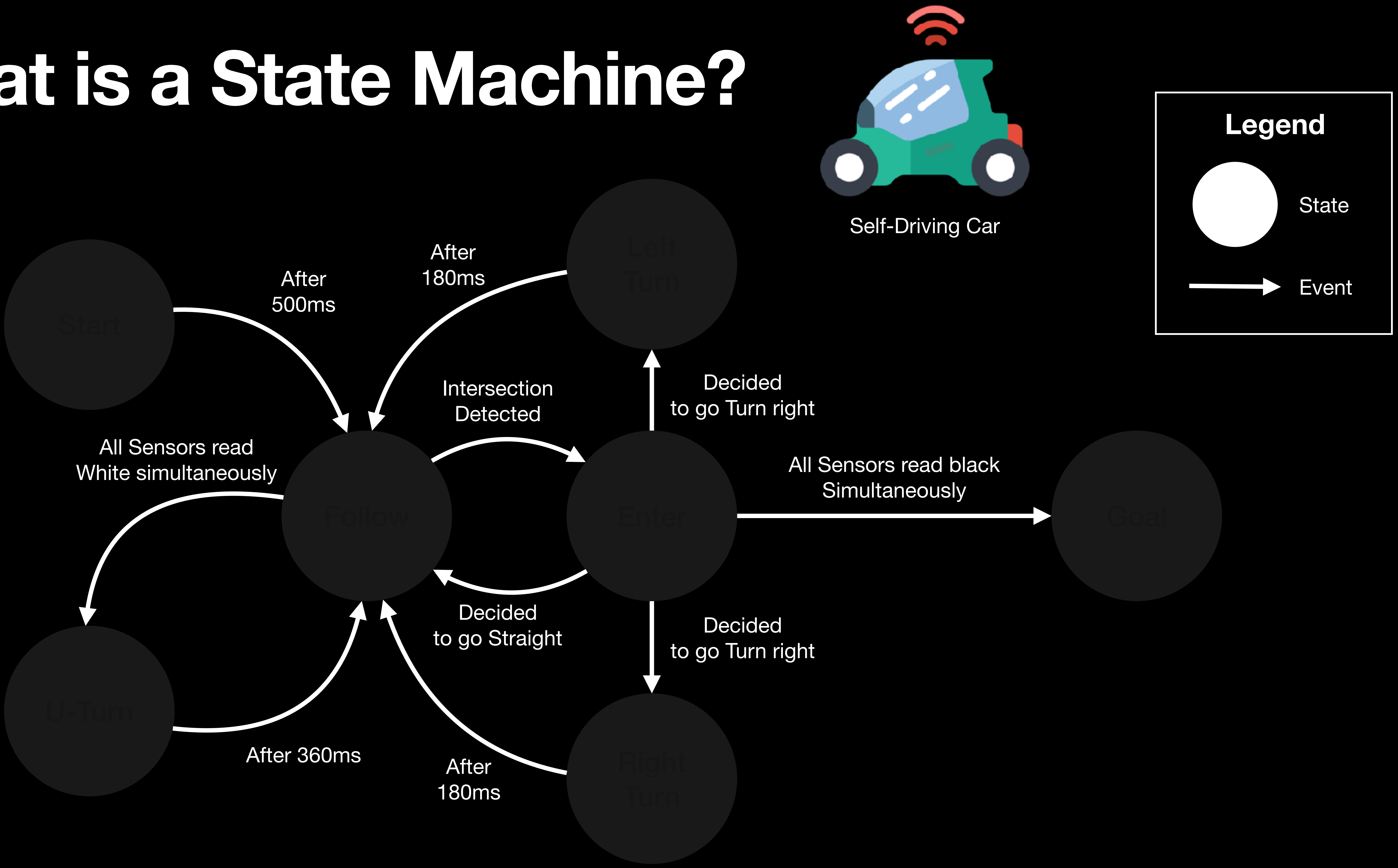
U-Turn

After 360ms

After 180ms

*: https://smashicons.com

What is a State Machine?



Self-Driving Car

Legend

State

Event

Start

Left Turn

Follow

Enter

Goal

U-Turn

Right Turn

After 500ms

After 180ms

Intersection Detected

Decided to go Turn right

All Sensors read White simultaneously

All Sensors read black Simultaneously

Decided to go Straight

Decided to go Turn right

After 360ms

After 180ms

*: https://smashicons.com

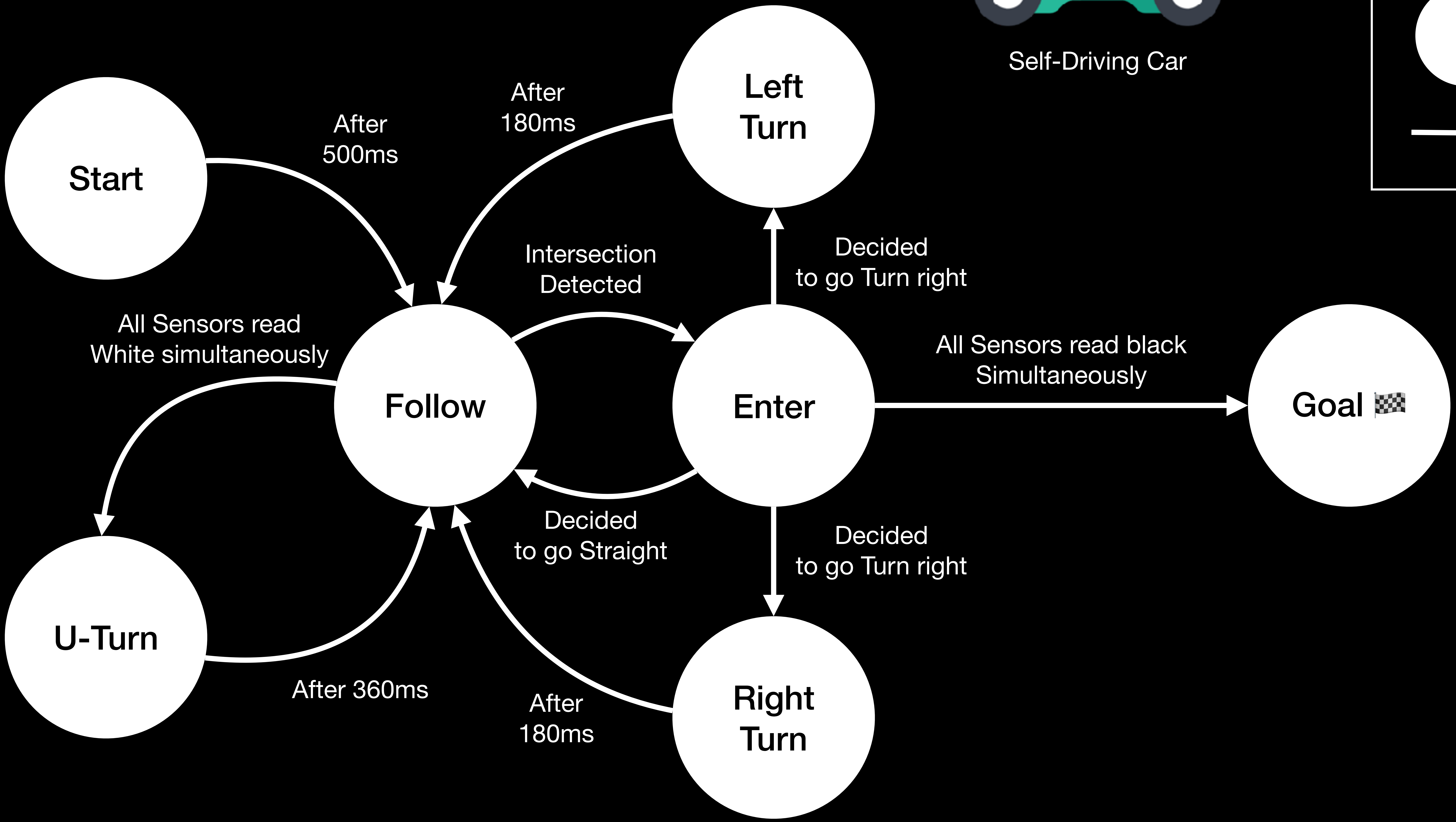
What is a State Machine?

Self-Driving Car

Legend

● State

→ Event

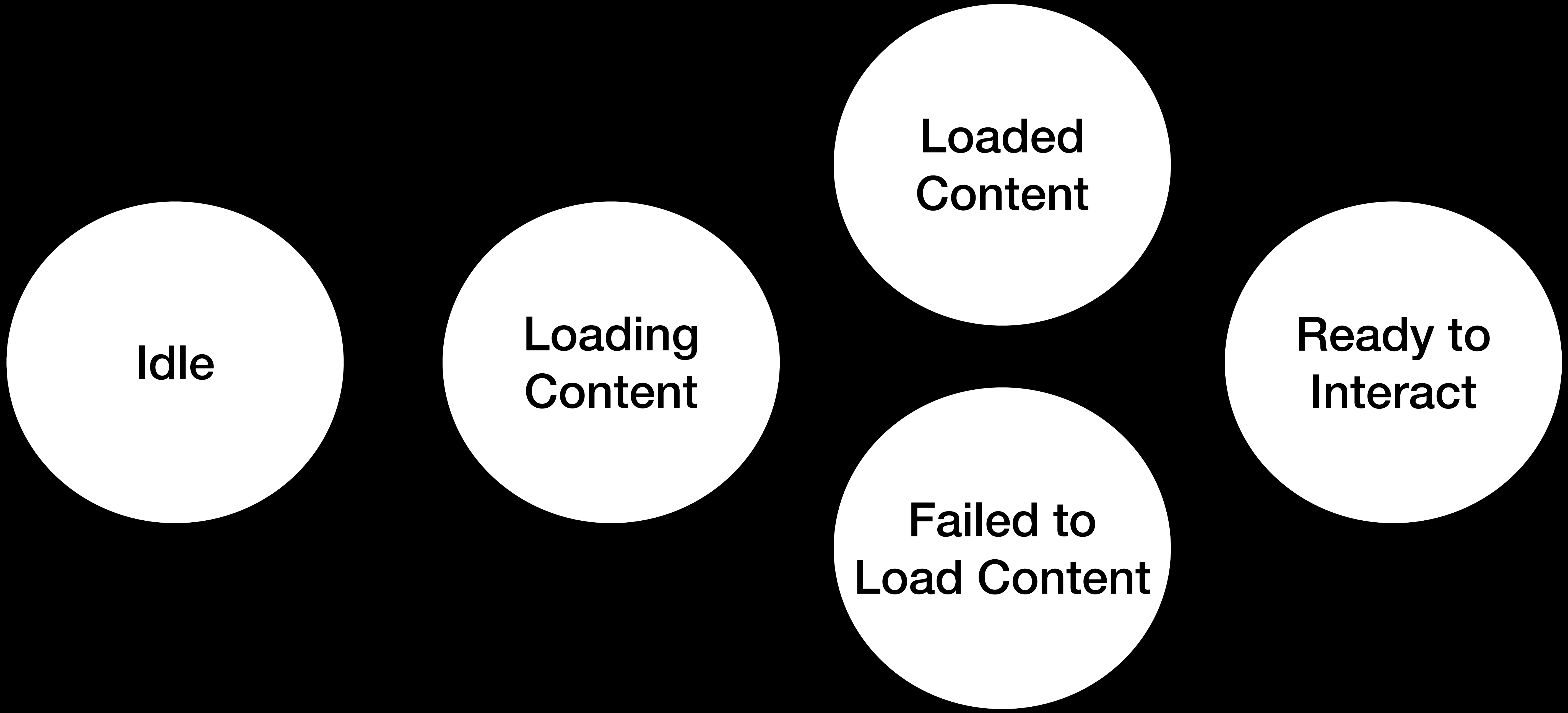


Why State Machine? 🤔

Why State Machine?

Readable and Predicted

Home Screen
e-Commerce



Idle

Loading Content

Loaded Content

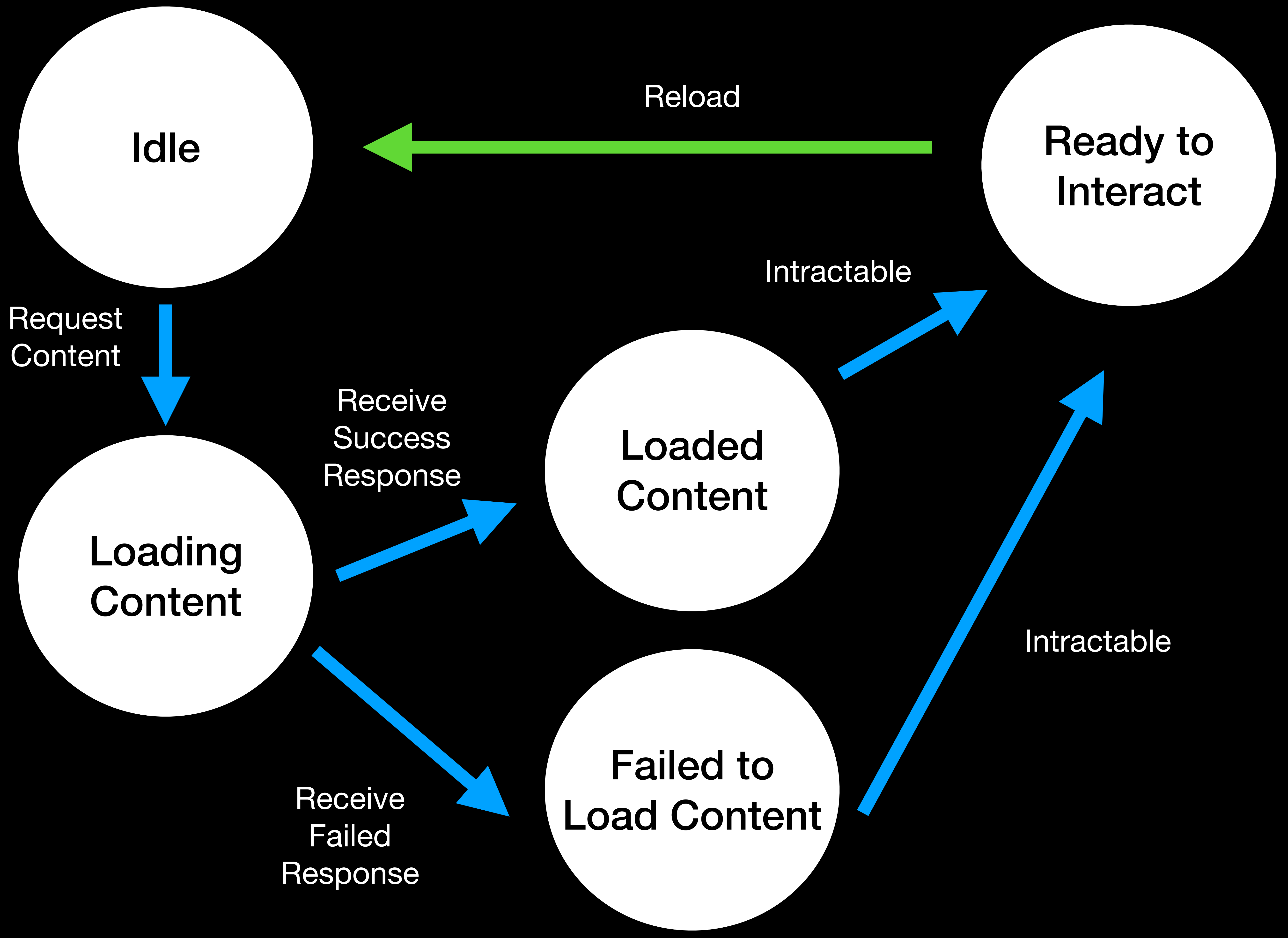
Failed to Load Content

Ready to Interact

Why State Machine?

Readable and Predicted

Home Screen
e-Commerce



How to implement:

1. Design Pattern. Link
2. Essential Implementation.

How to implement:

1. Design Pattern. [Link]

2. Essential Implementation.

How to implement State Machine:

Implementing State Machine as the essential Implementation

1. Determine the state of each components.
2. Determine the event State.
3. Determine the reaction state of each state.

Demo! 👨‍💻

How to implement State Machine:

Implementing State Machine as the essential Implementation

1. Determine the state of each components.
2. Determine the event State.
3. Determine the reaction state of each state.
4. Write enum that readable.

Find this helpful? Do It! 🍾

Thank You! 🎉