

#WWDC19

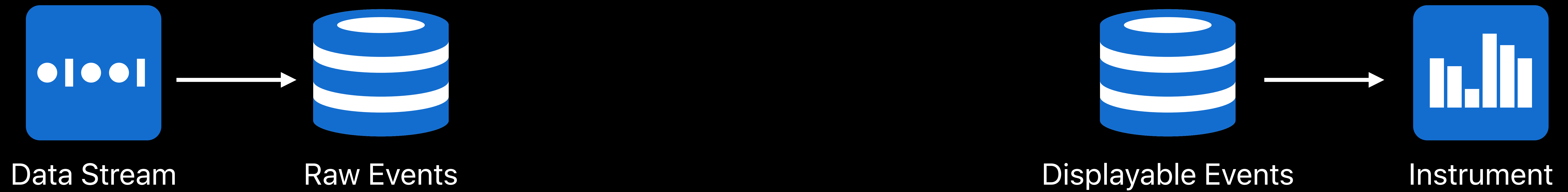
# Modeling in Custom Instruments

Building smarter instruments

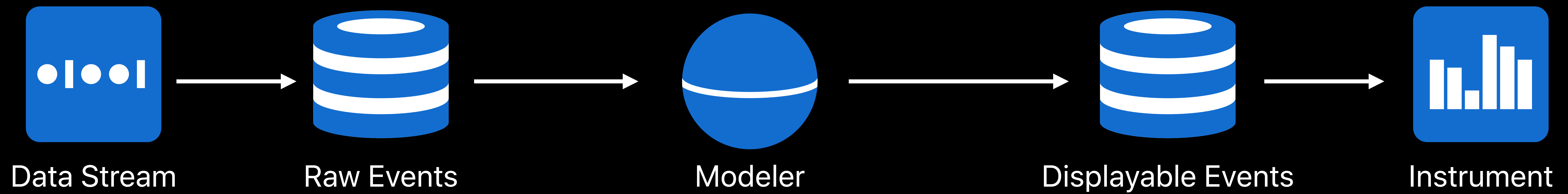
Chad Woolf, Performance Tools

Alejandro Lucena, Performance Tools

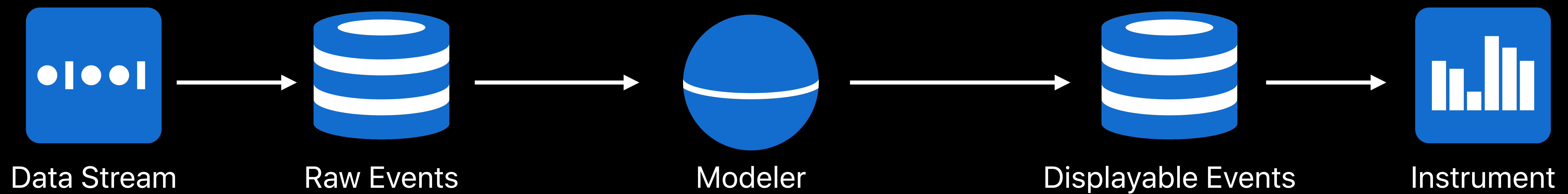
# Instruments Architecture



# Instruments Architecture



# Instruments Architecture



# Time Profiler

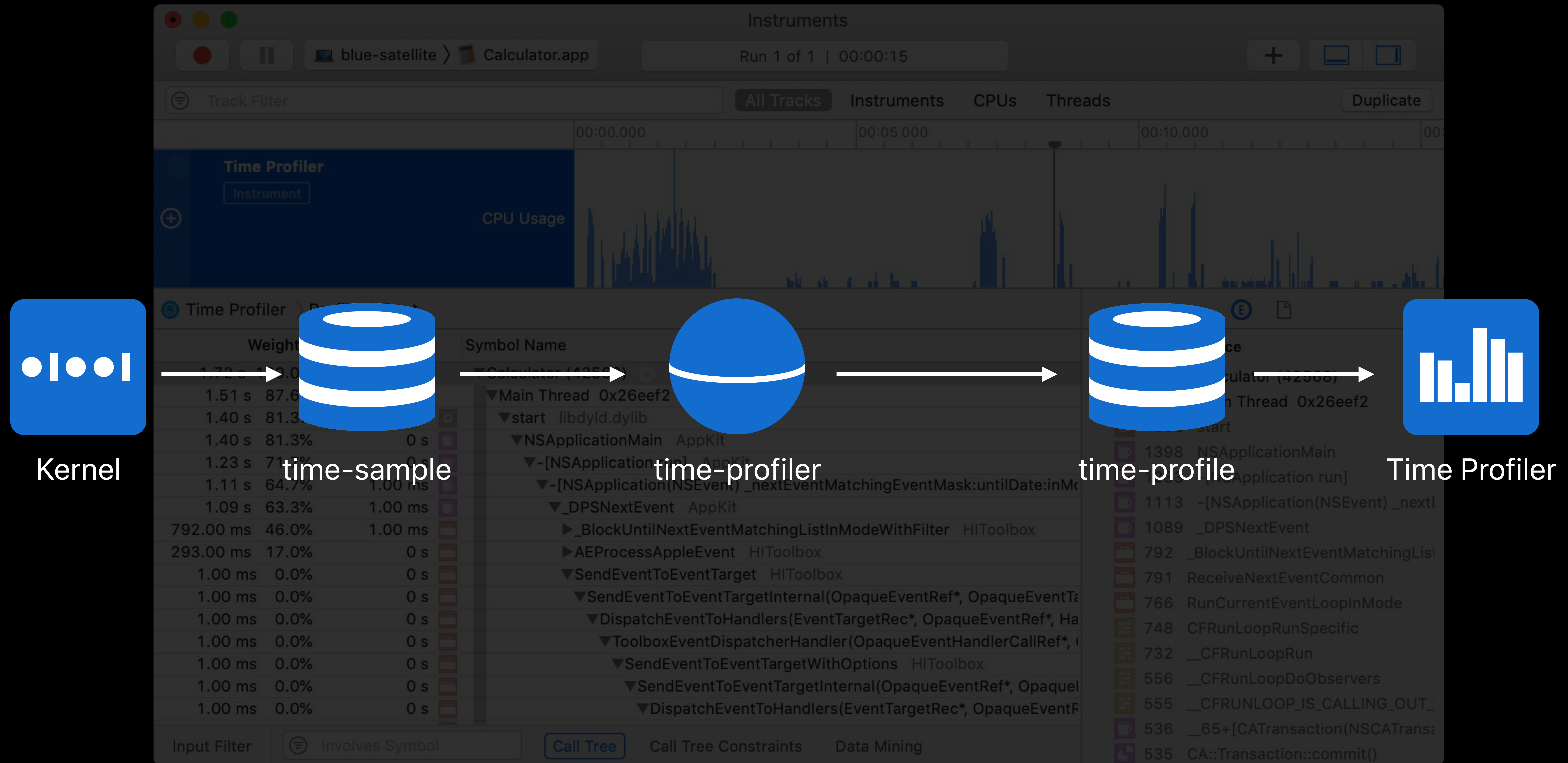
The screenshot shows the Instruments application window with the Time Profiler instrument selected. The top bar indicates the target is 'blue-satellite' running 'Calculator.app' for 'Run 1 of 1' with a duration of '00:00:15'. The 'Track Filter' is set to 'All Tracks'. The main display area shows a 'CPU Usage' graph with a blue bar chart representing CPU activity over time. Below the graph, the 'Time Profiler > Profile > Root' view is active, displaying a table of symbols and their weights.

Weight	Self Weight	Symbol Name	
1.72 s	100.0%	0 s	▼ Calculator (42568)
1.51 s	87.6%	0 s	▼ Main Thread 0x26eef2
1.40 s	81.3%	0 s	▼ start libdyld.dylib
1.40 s	81.3%	0 s	▼ NSApplicationMain AppKit
1.23 s	71.7%	0 s	▼ -[NSApplication run] AppKit
1.11 s	64.7%	1.00 ms	▼ -[NSApplication(NSEvent) _nextEventMatchingEventMask:untilDate:inMode: AppKit
1.09 s	63.3%	1.00 ms	▼ _DPSNextEvent AppKit
792.00 ms	46.0%	1.00 ms	▶ _BlockUntilNextEventMatchingListInModeWithFilter HIToolbox
293.00 ms	17.0%	0 s	▶ AEProcessAppleEvent HIToolbox
1.00 ms	0.0%	0 s	▼ SendEventToEventTarget HIToolbox
1.00 ms	0.0%	0 s	▼ SendEventToEventTargetInternal(OpaqueEventRef*, OpaqueEventTargetRef*, OpaqueEventRef*) HIToolbox
1.00 ms	0.0%	0 s	▼ DispatchEventToHandlers(EventTargetRef*, OpaqueEventRef*, HandlerListRef*) HIToolbox
1.00 ms	0.0%	0 s	▼ ToolboxEventDispatcherHandler(OpaqueEventHandlerCallRef*, OpaqueEventRef*, void*) HIToolbox
1.00 ms	0.0%	0 s	▼ SendEventToEventTargetWithOptions HIToolbox
1.00 ms	0.0%	0 s	▼ SendEventToEventTargetInternal(OpaqueEventRef*, OpaqueEventTargetRef*, OpaqueEventRef*) HIToolbox
1.00 ms	0.0%	0 s	▼ DispatchEventToHandlers(EventTargetRef*, OpaqueEventRef*, HandlerListRef*) HIToolbox

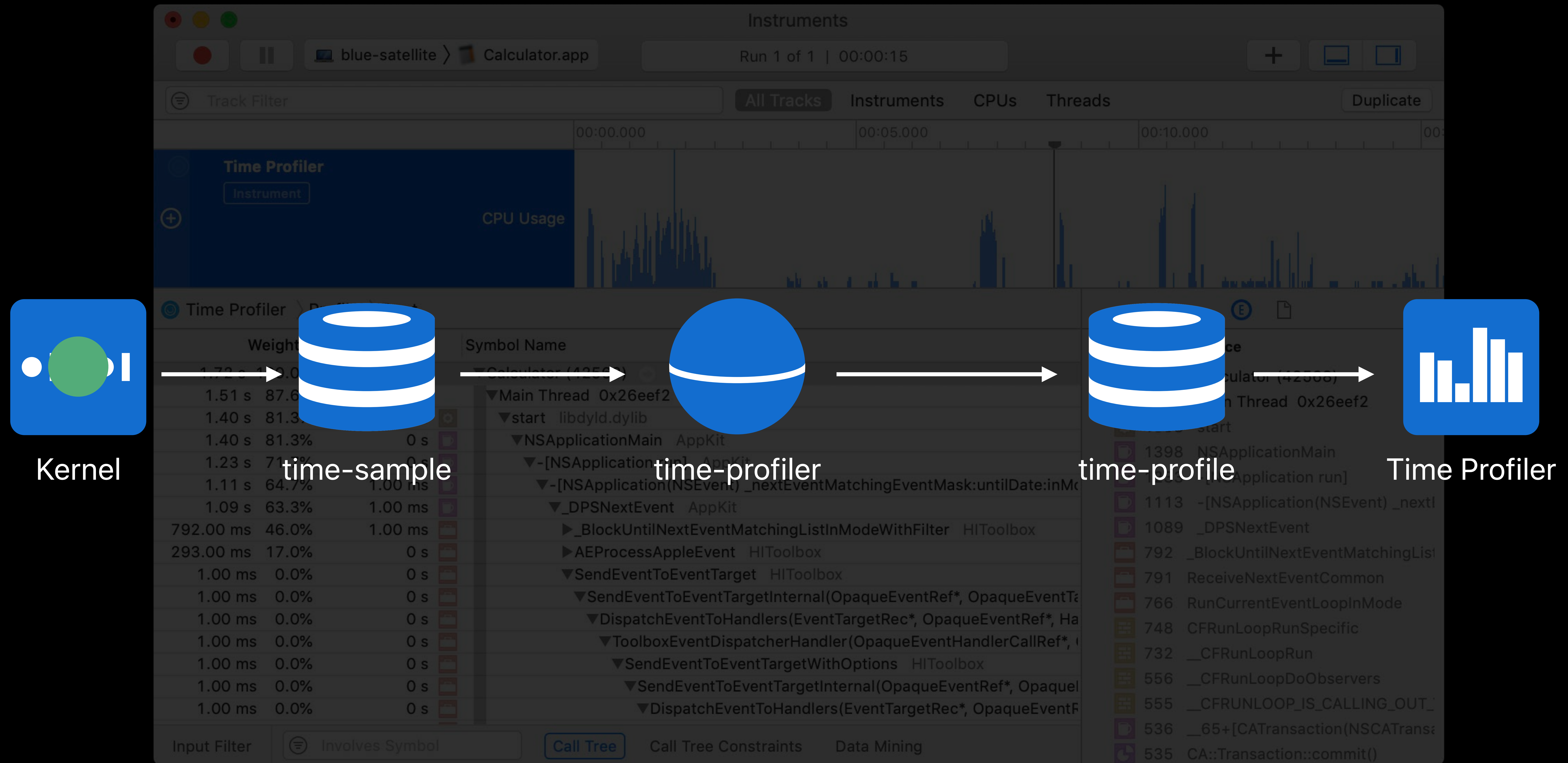
At the bottom, the 'Input Filter' is set to 'Involves Symbol', and the 'Call Tree' view is selected. The 'Heaviest Stack Trace' panel on the right shows the following stack frames:

- 1719 Calculator (42568)
- 1507 Main Thread 0x26eef2
- 1398 start
- 1398 NSApplicationMain
- 1233 -[NSApplication run]
- 1113 -[NSApplication(NSEvent) \_nextEventMatchingEventMask:untilDate:inMode: AppKit]
- 1089 \_DPSNextEvent
- 792 \_BlockUntilNextEventMatchingListInModeWithFilter
- 791 ReceiveNextEventCommon
- 766 RunCurrentEventLoopInMode
- 748 CFRRunLoopRunSpecific
- 732 \_\_CFRunLoopRun
- 556 \_\_CFRunLoopDoObservers
- 555 \_\_CFRUNLOOP\_IS\_CALLING\_OUT\_TO\_A\_CALLBACK\_BLOCK
- 536 \_\_65+[CATransaction(NSCATransaction) initWithBlock:]
- 535 CA::Transaction::commit()

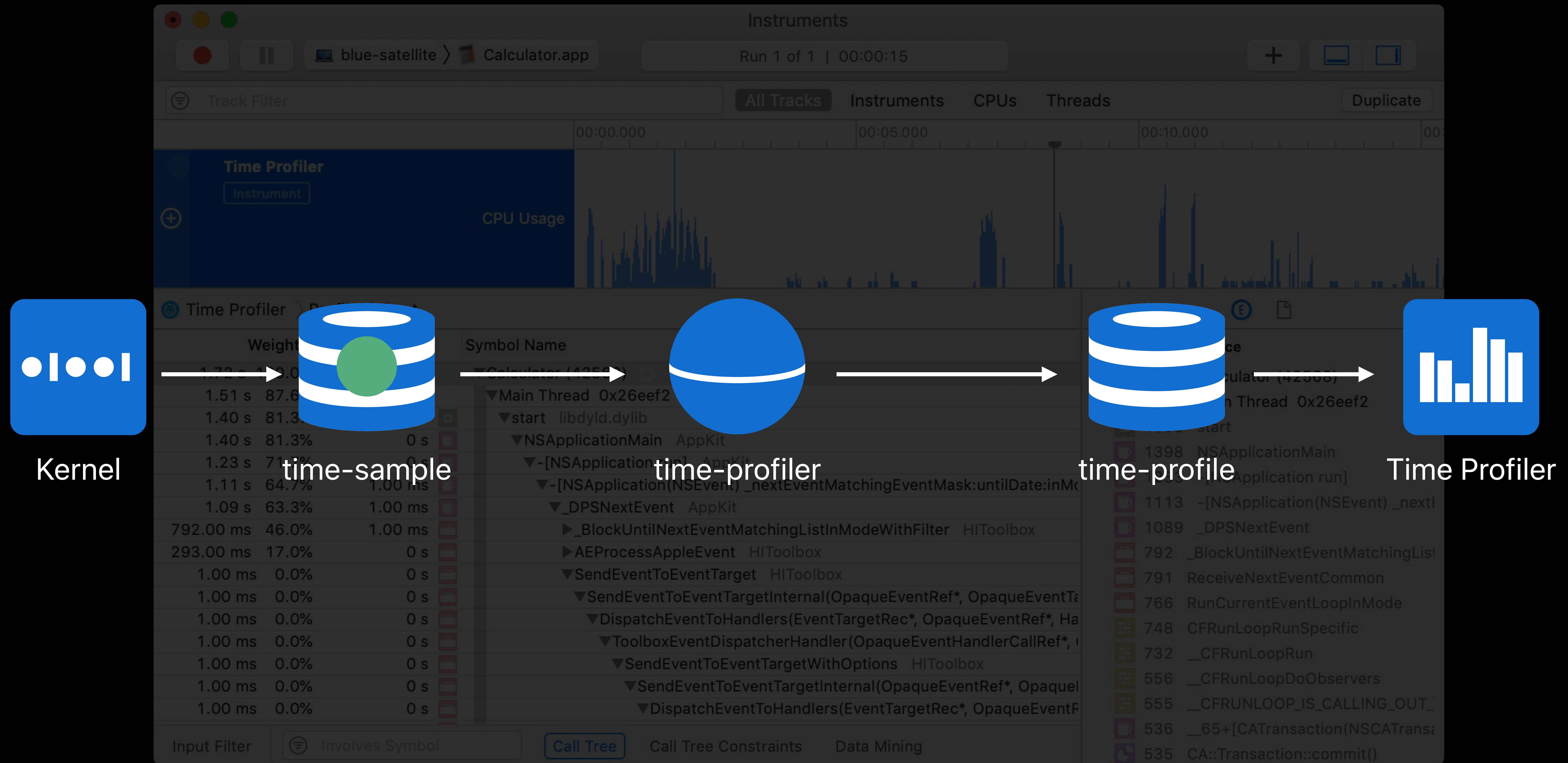
# Transformation



# Transformation

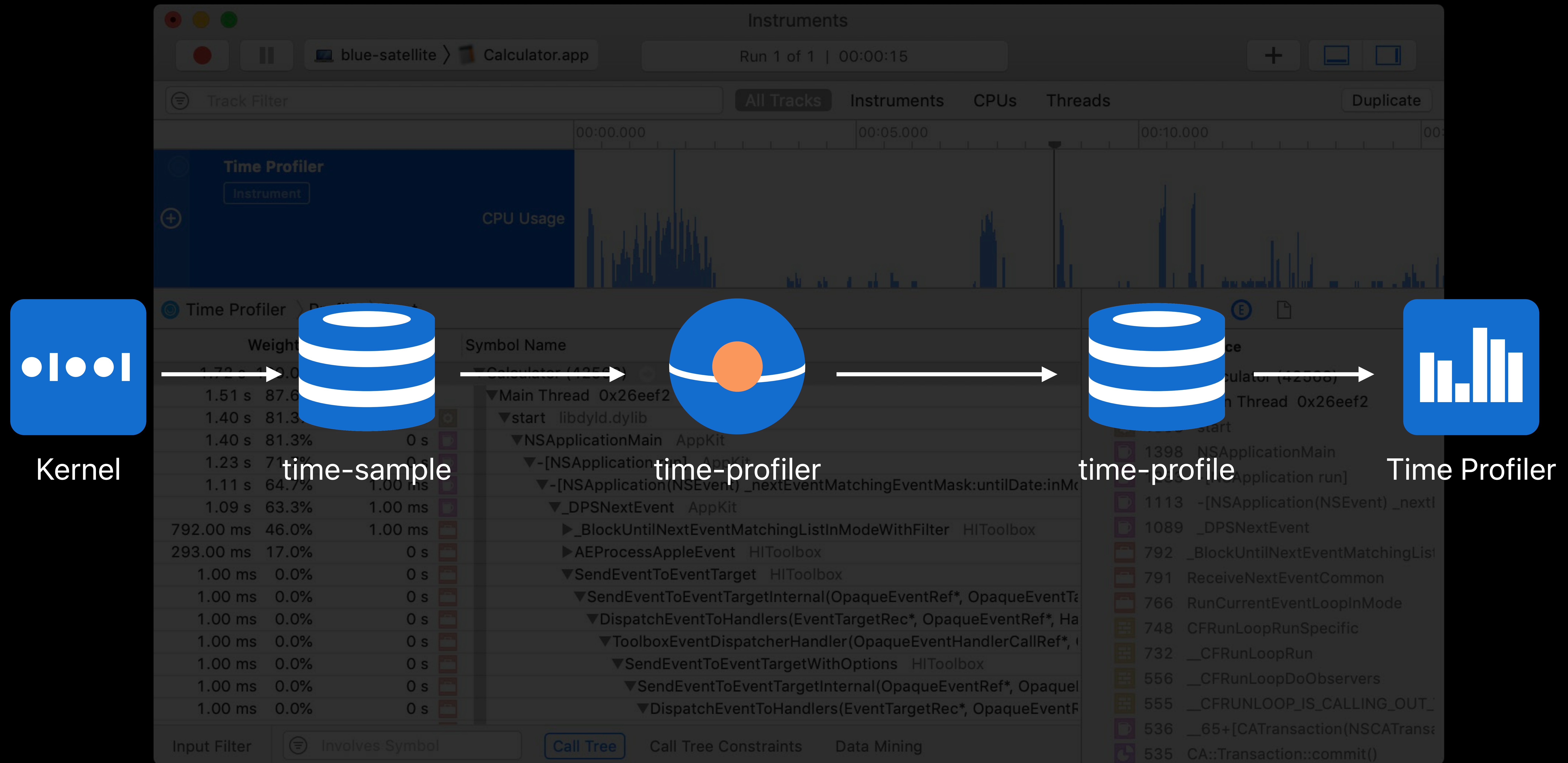


# Transformation

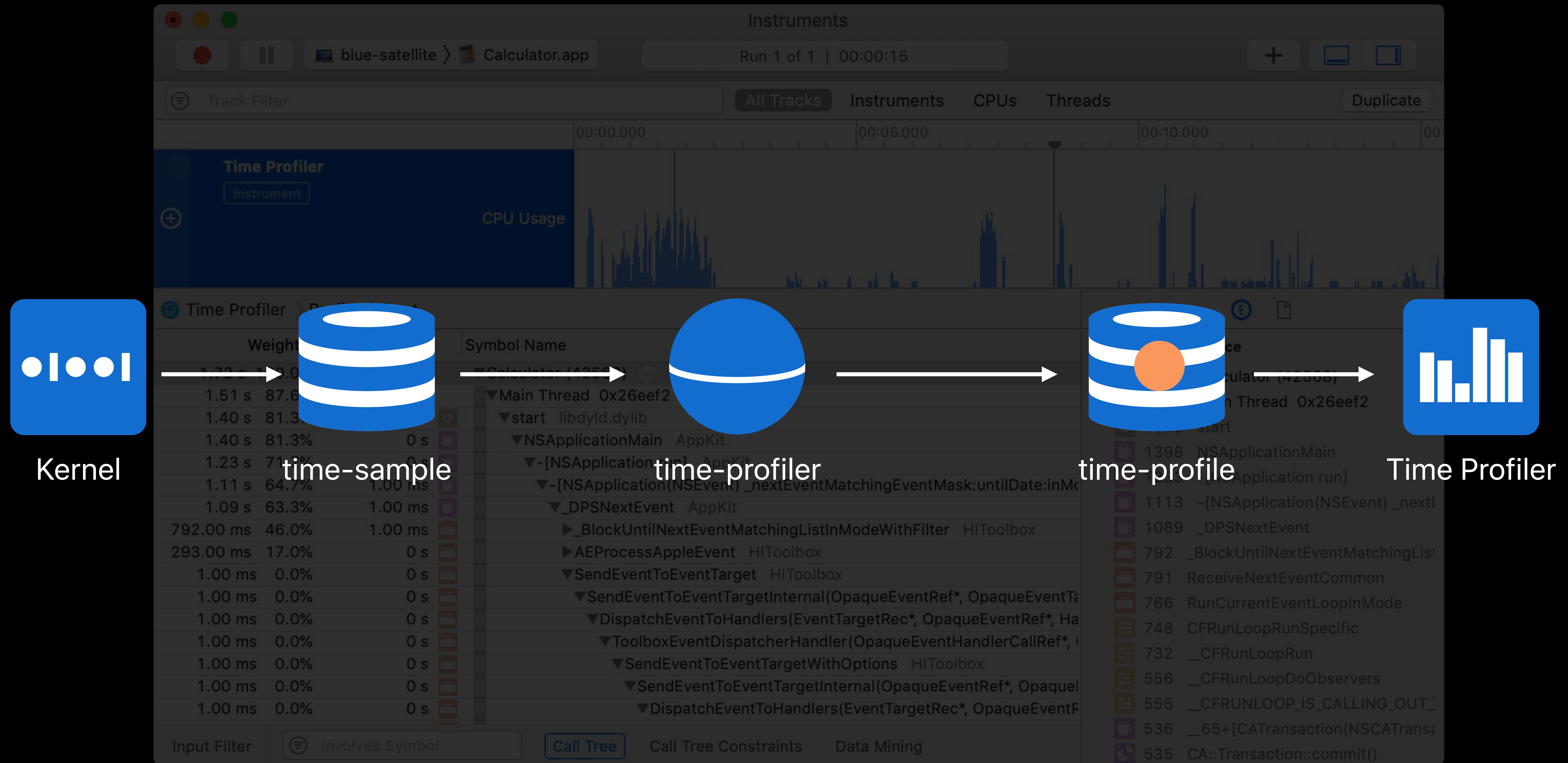




# Transformation



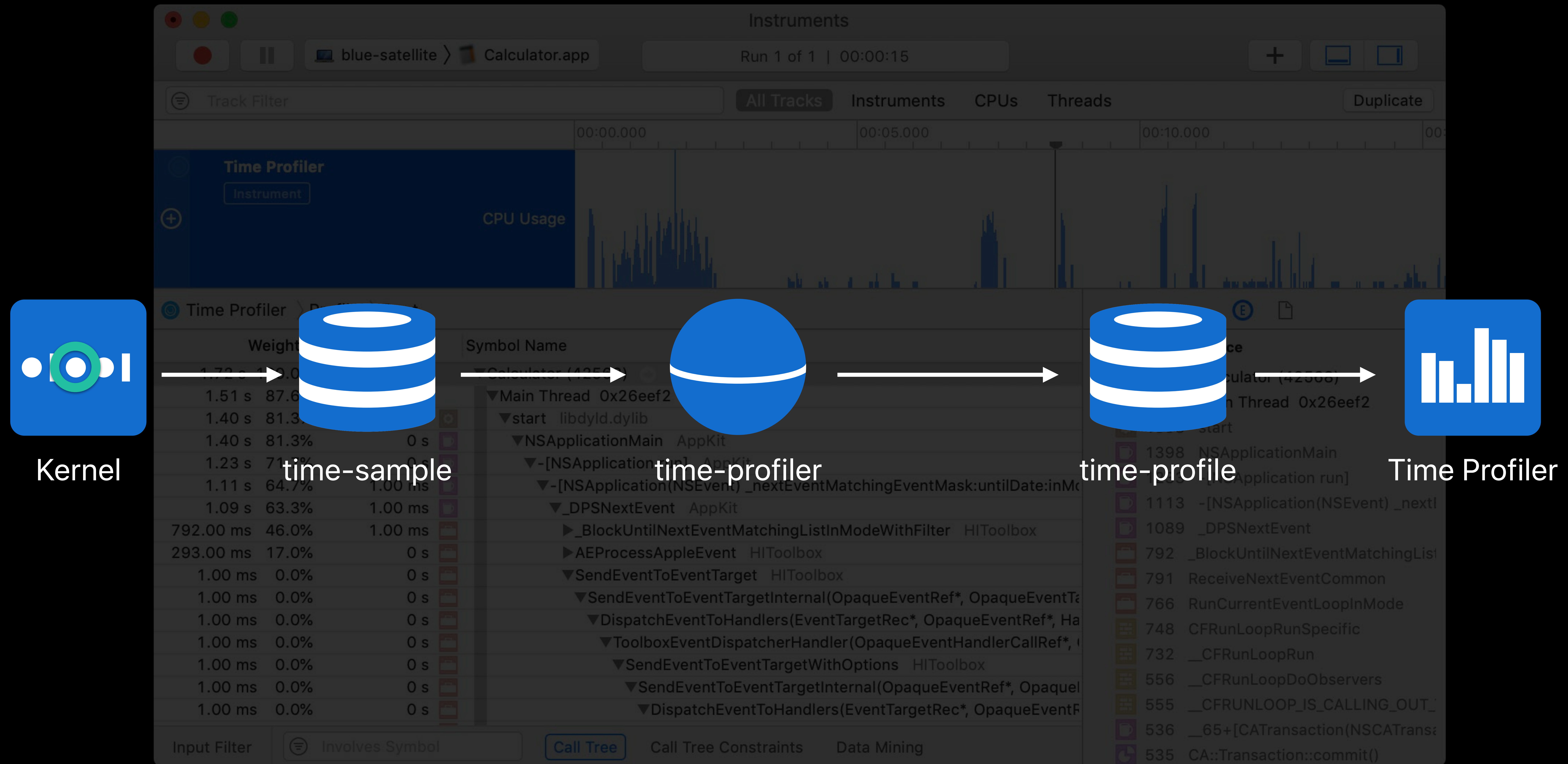
# Transformation



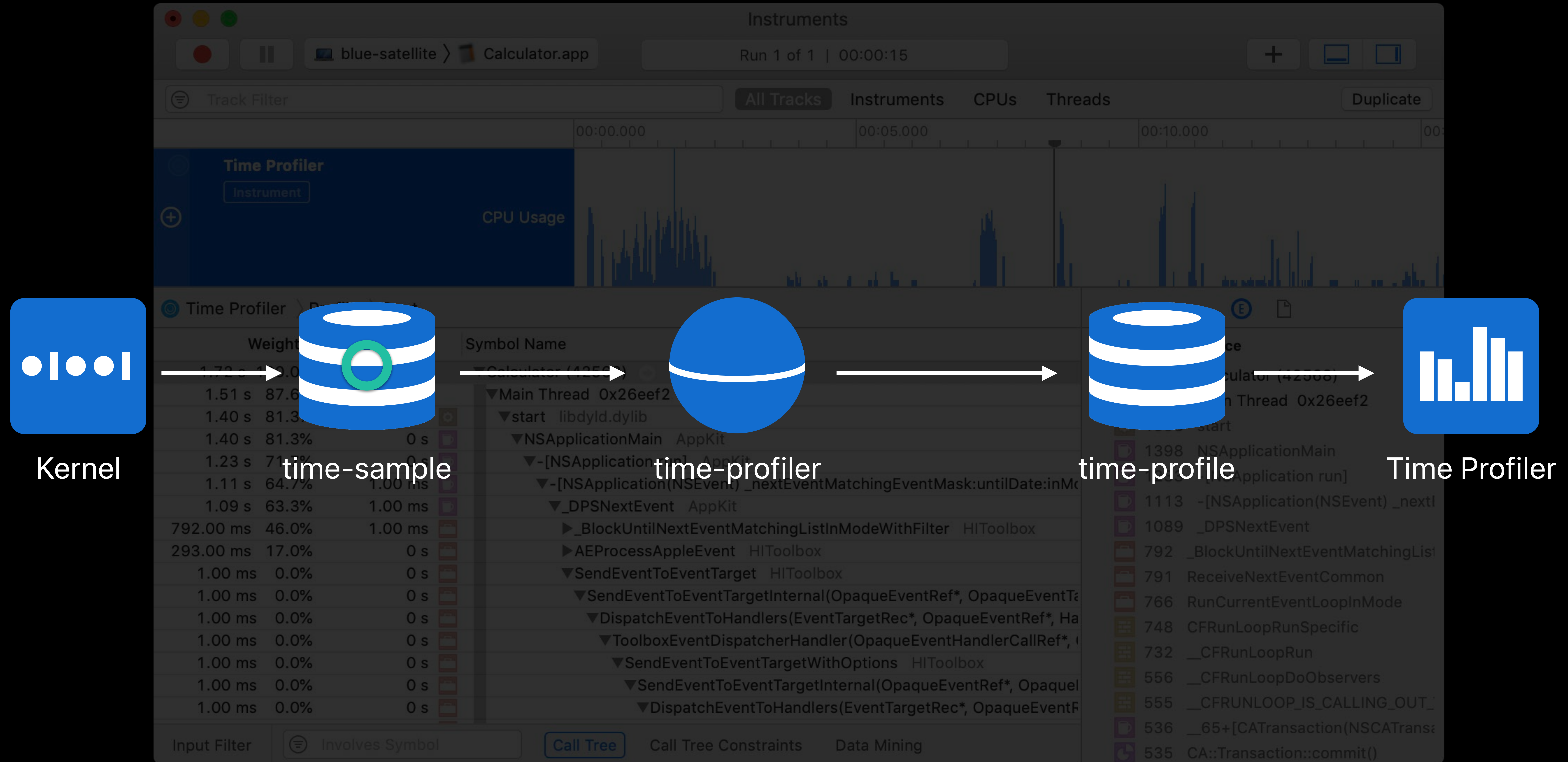




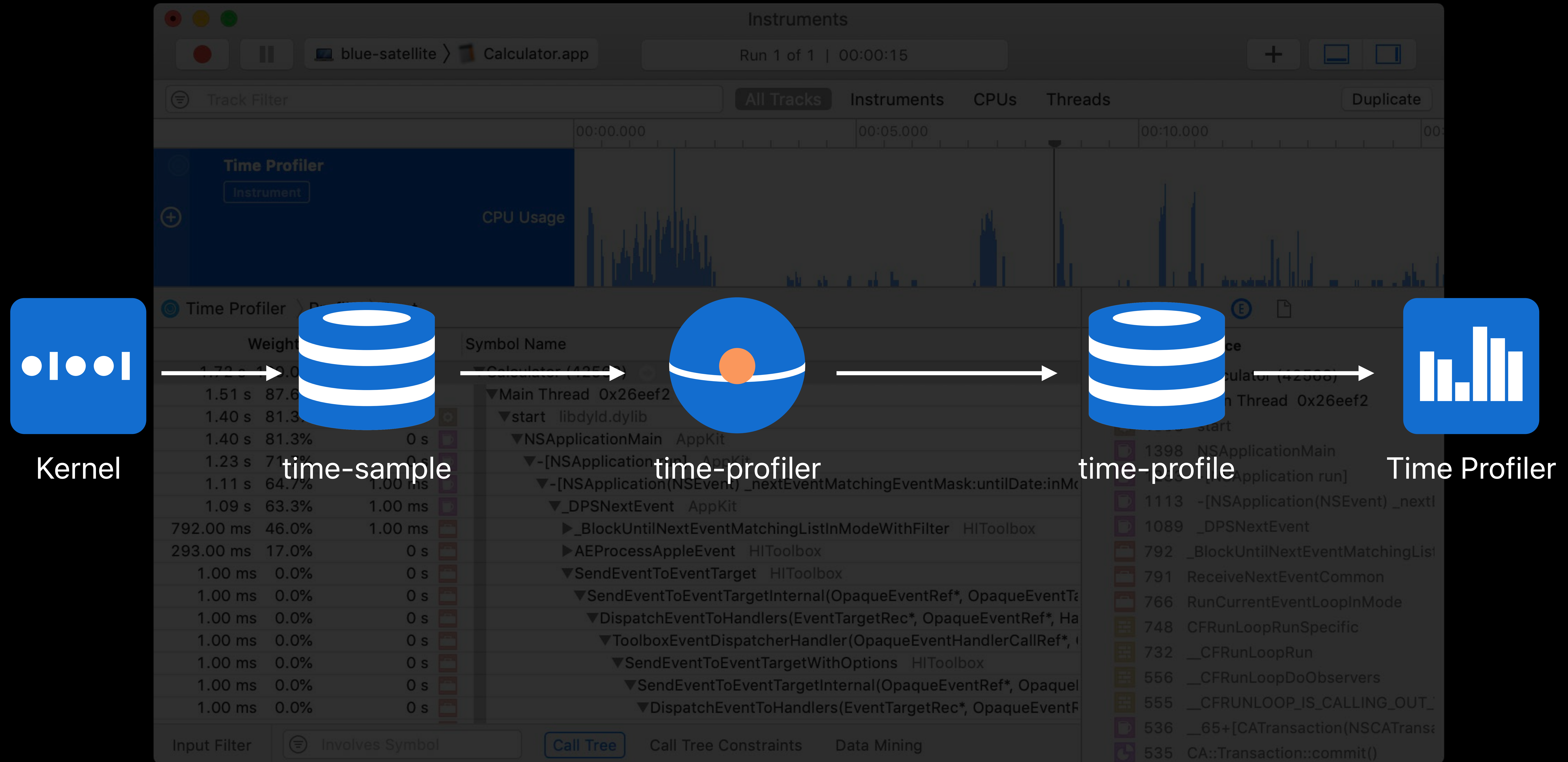
# Expansion



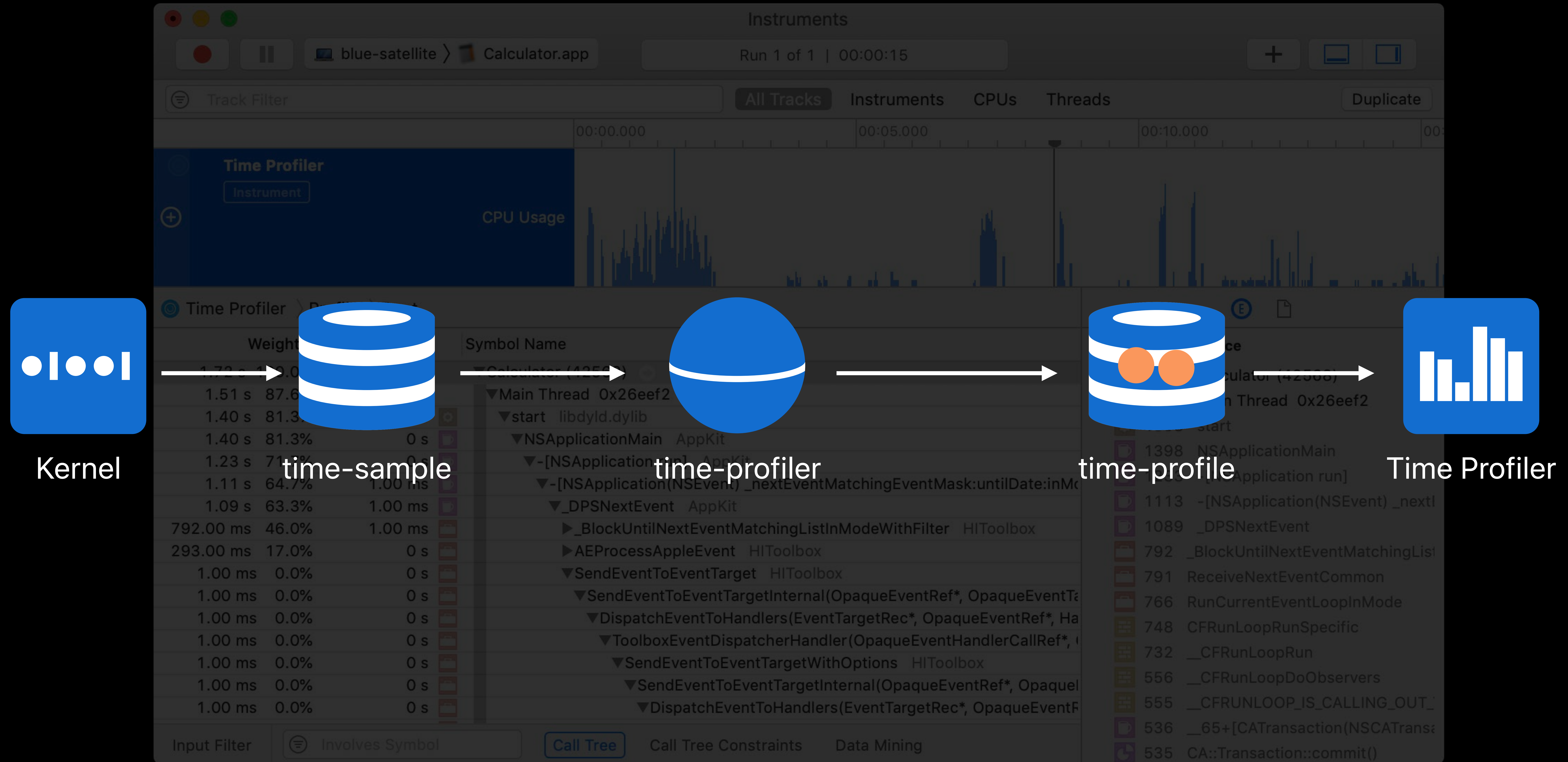
# Expansion



# Expansion



# Expansion





# Themes

# Themes

Absorbing complexity

# Themes

Absorbing complexity

Simplicity elsewhere

Review modeling basics

Build a custom modeler

Rules execution

Speculation

Review modeling basics

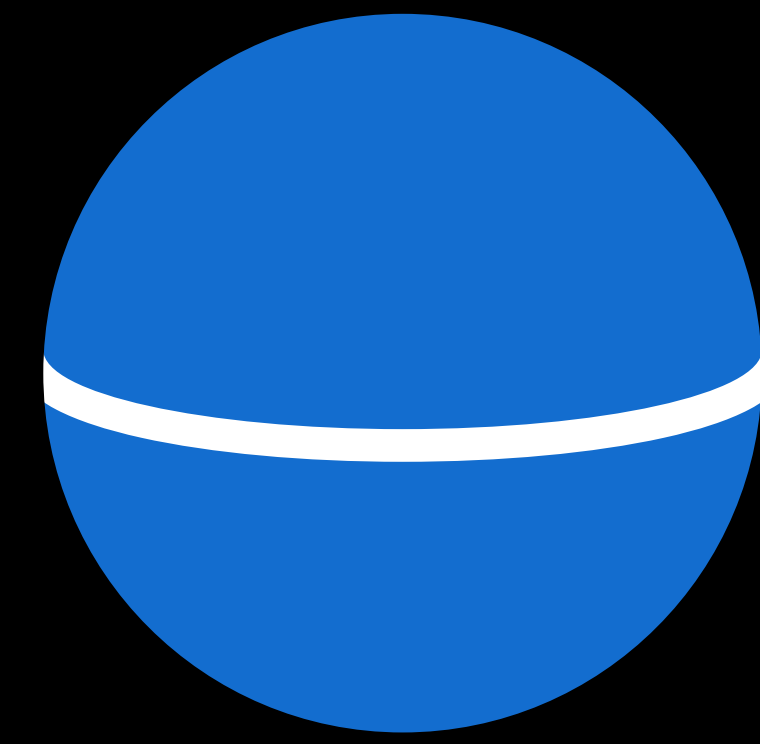
Build a custom modeler

Rules execution

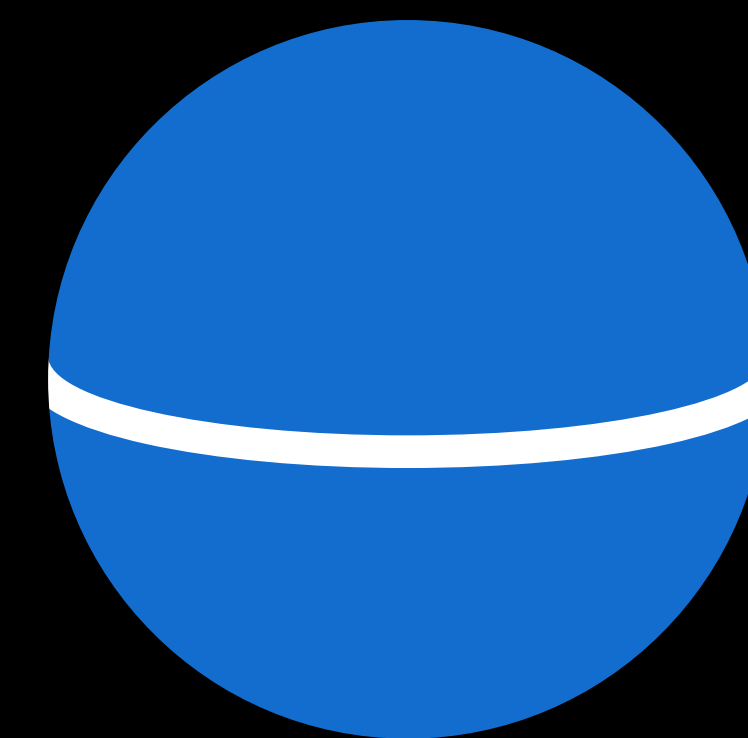
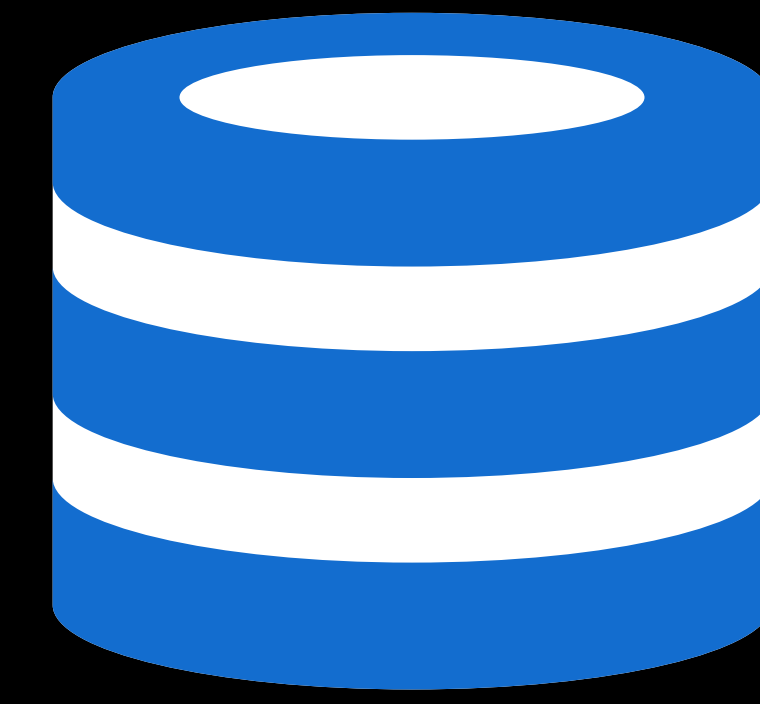
Speculation

# Modeling Basics

# Custom Instruments



# Custom Instruments





# Custom Instruments



# Custom Instruments



# Custom Instruments

# Custom Instruments

# When Do I Need a Custom Modeler?

# When Do I Need a Custom Modeler?

Generated modelers are simplified

# When Do I Need a Custom Modeler?

Generated modelers are simplified

Fuse data from multiple tables

# When Do I Need a Custom Modeler?

Generated modelers are simplified

Fuse data from multiple tables

Maintain a working memory



# When Do I Need a Custom Modeler?

Generated modelers are simplified

Fuse data from multiple tables

Maintain a working memory

Custom graphs

# When Do I Need a Custom Modeler?

Generated modelers are simplified

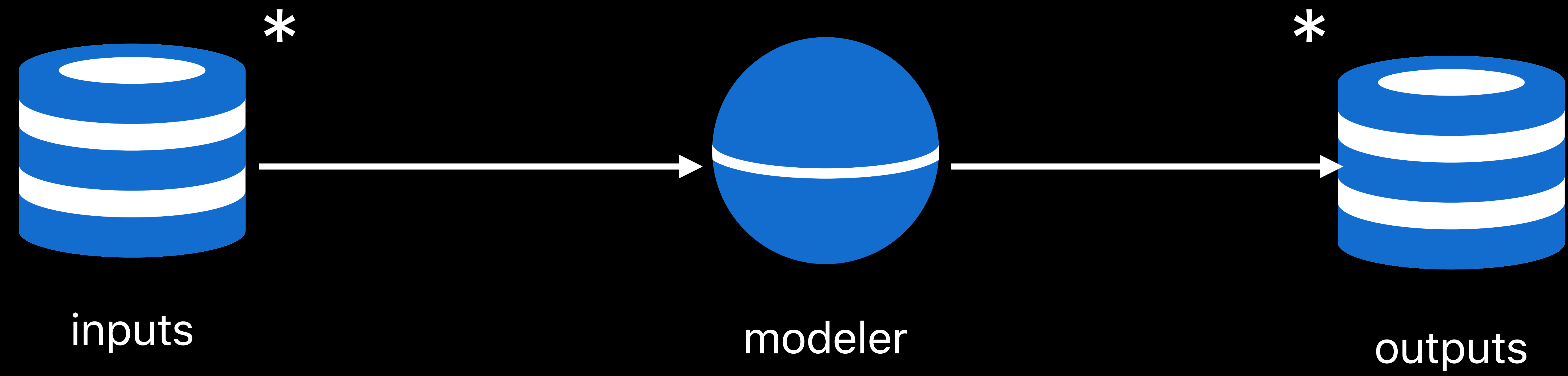
Fuse data from multiple tables

Maintain a working memory

Custom graphs

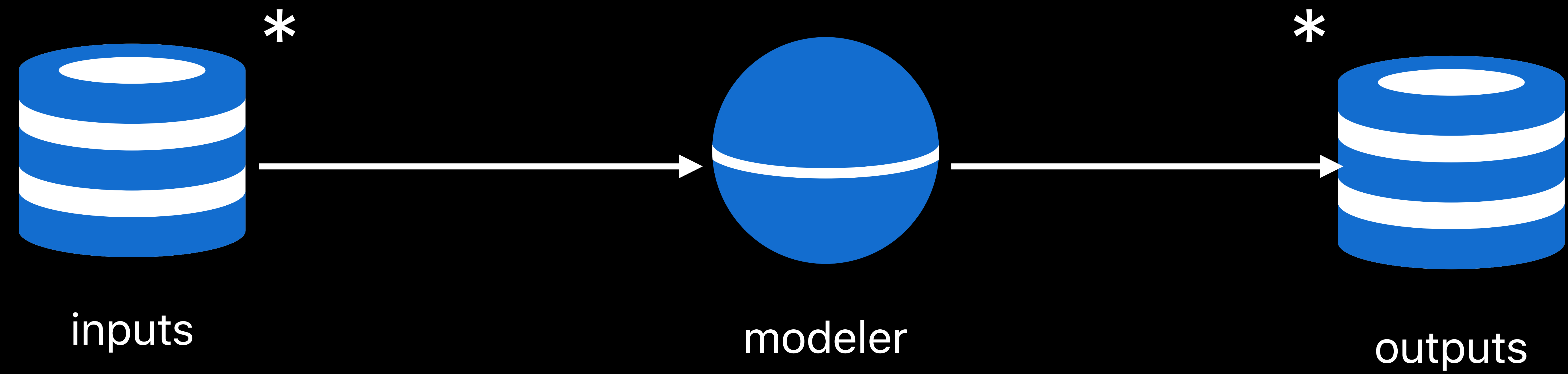
Smarter instruments

# Modelers



# Modelers

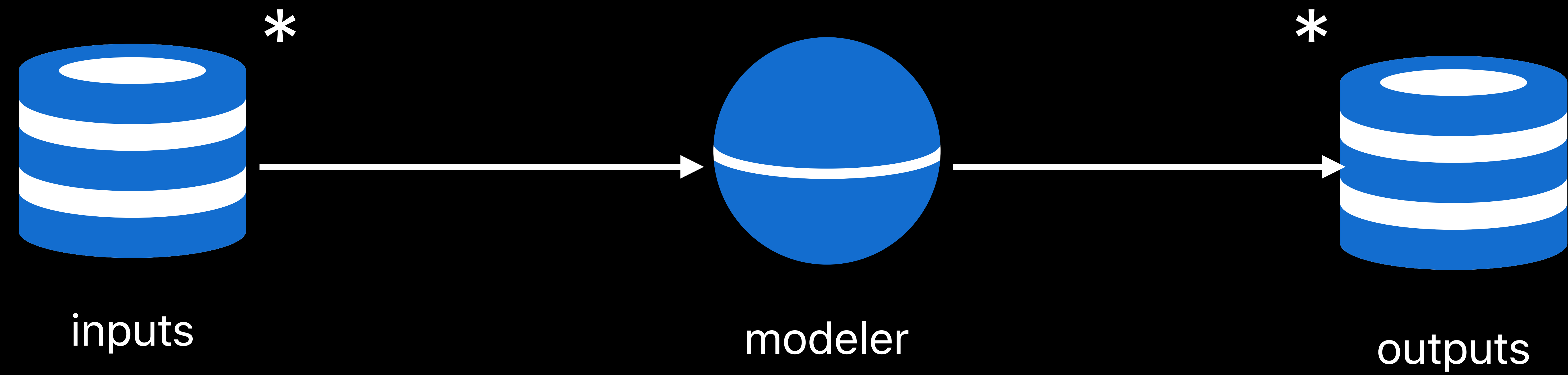
Time ordered



# Modelers

Time ordered

Working memory

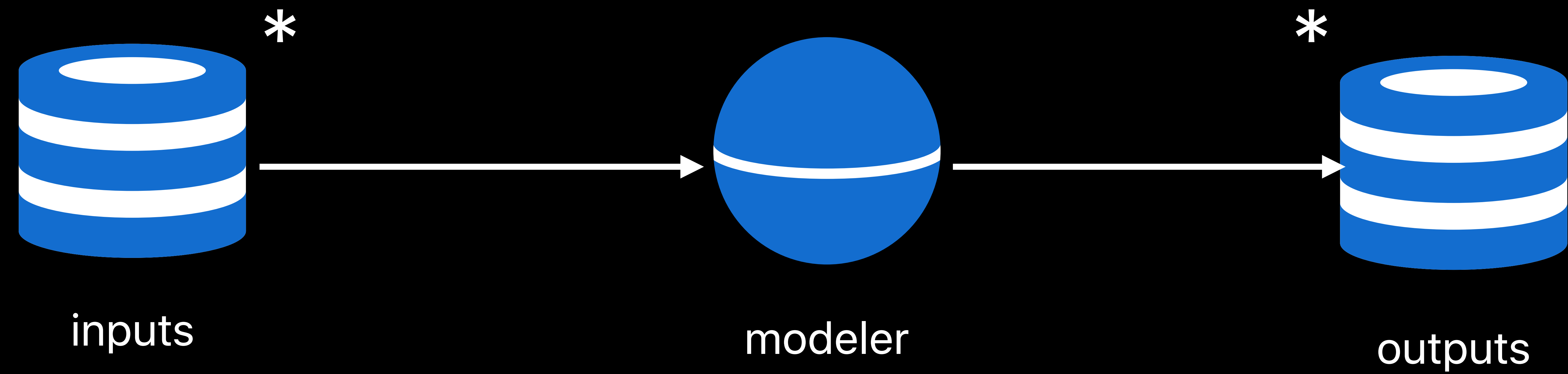


# Modelers

Time ordered

Working memory

CLIPS



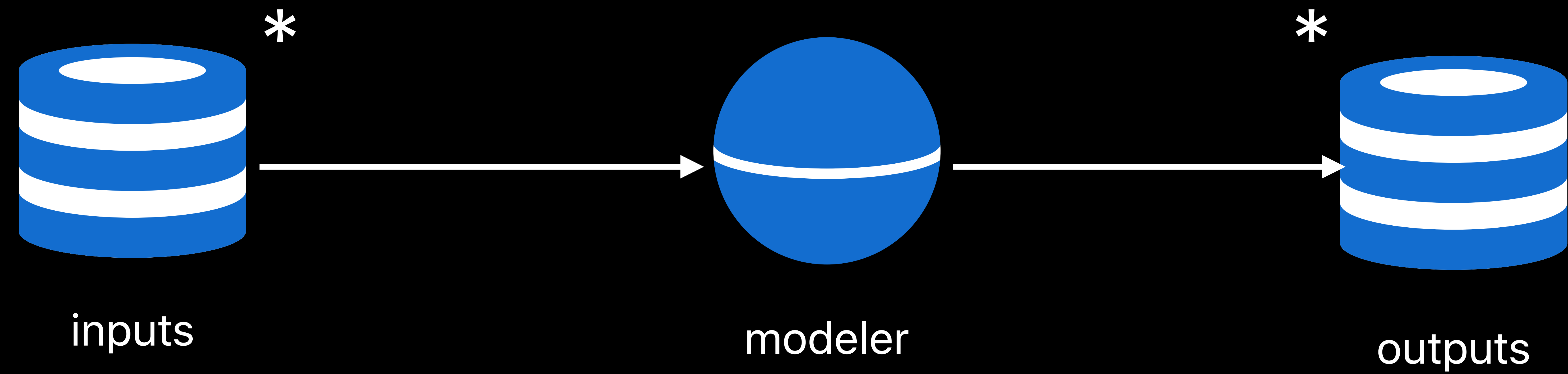
# Modelers

Time ordered

Working memory

CLIPS

Output



# Getting Started



# Getting Started

Decide what you want to model

# Getting Started

Decide what you want to model

# Getting Started

Decide what you want to model

Trace your code with `os_signpost`

# Getting Started

Decide what you want to model

Trace your code with `os_signpost`

Write your rules

# Building a Modeler

Alejandro Lucena, Performance Tools

9:41



Sort

Goats!



Sleepy Goat

Hungry Goat

Happy Goat

Smart Goat

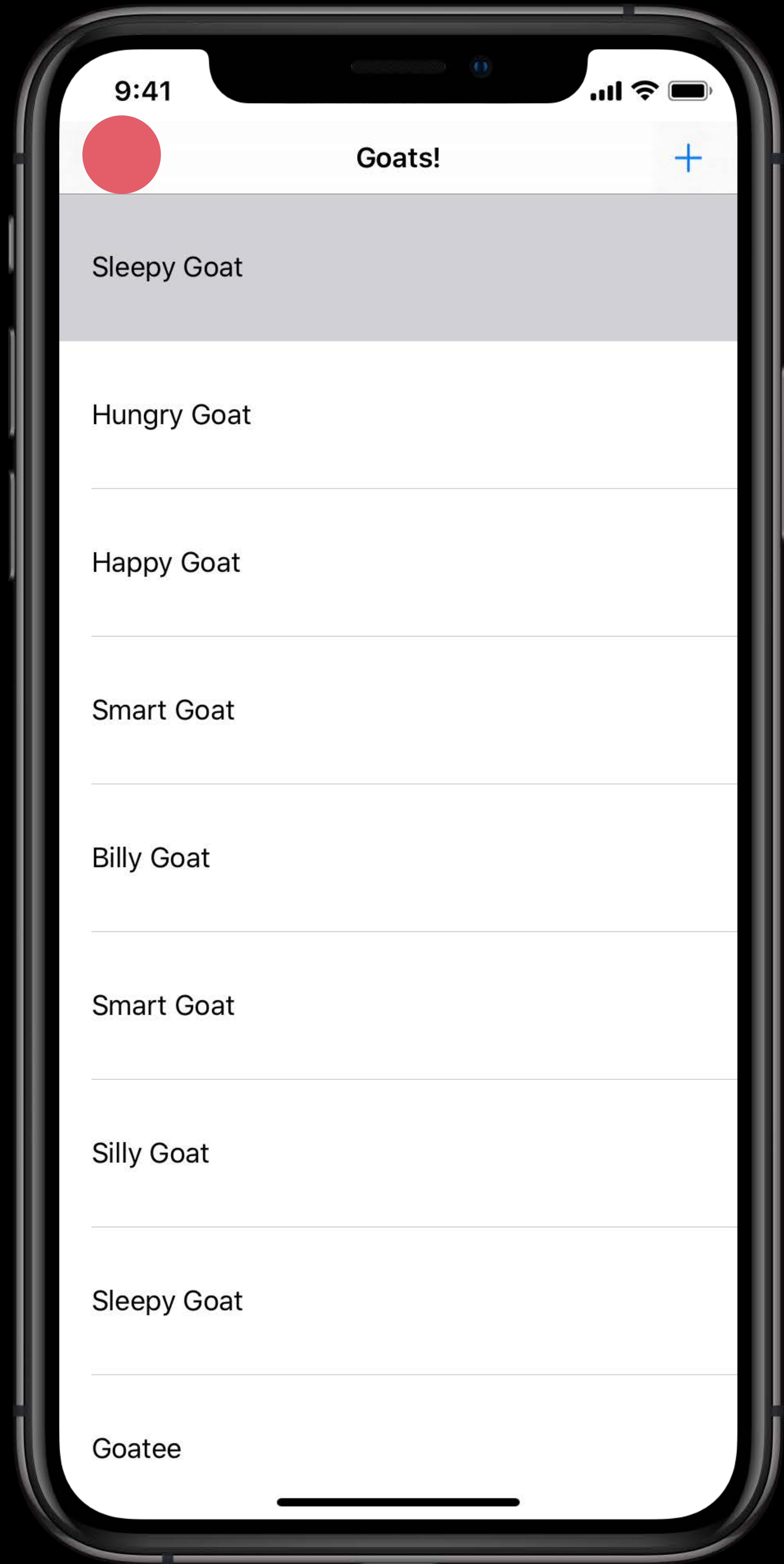
Billy Goat

Smart Goat

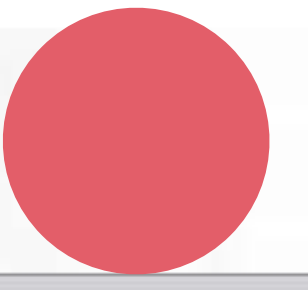
Silly Goat

Sleepy Goat

Goatee



9:41



Goats!



Sleepy Goat

Hungry Goat

Happy Goat

Smart Goat

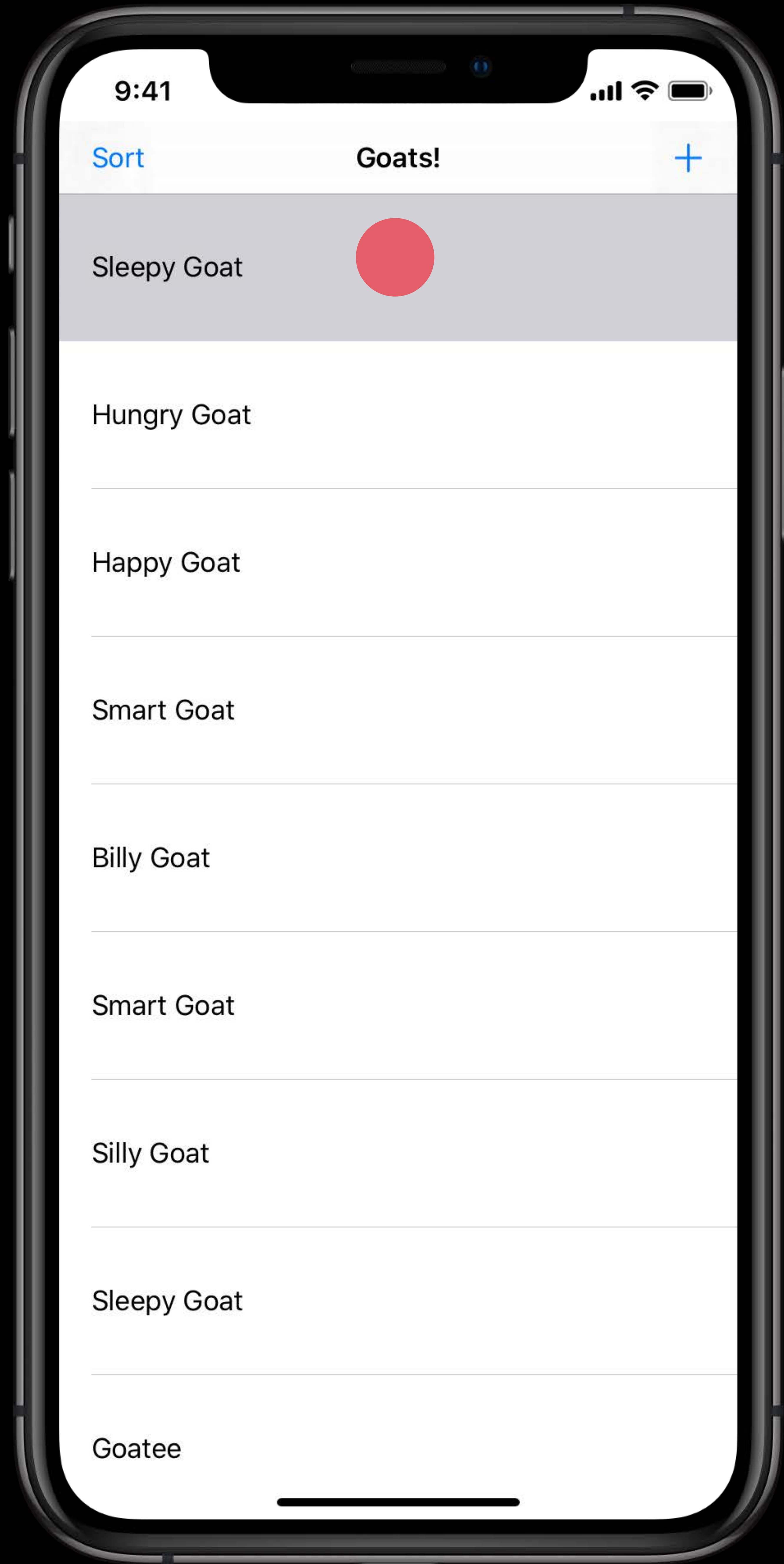
Billy Goat

Smart Goat

Silly Goat

Sleepy Goat

Goatee



9:41

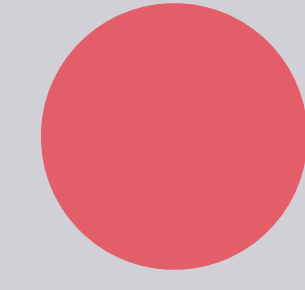


Sort

Goats!



Sleepy Goat



Hungry Goat

Happy Goat

Smart Goat

Billy Goat

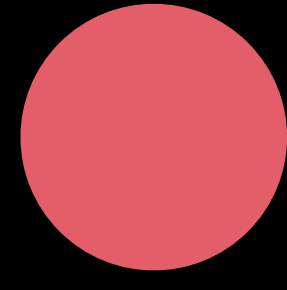
Smart Goat

Silly Goat

Sleepy Goat

Goatee





9:41



Sort

Goats!



Sleepy Goat

Hungry Goat

Happy Goat

Smart Goat

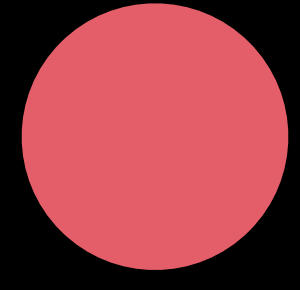
Billy Goat

Smart Goat

Silly Goat

Sleepy Goat

Goatee



9:41



Sort

Goats!



Sleepy Goat

Hungry Goat

Happy Goat

Smart Goat

Billy Goat

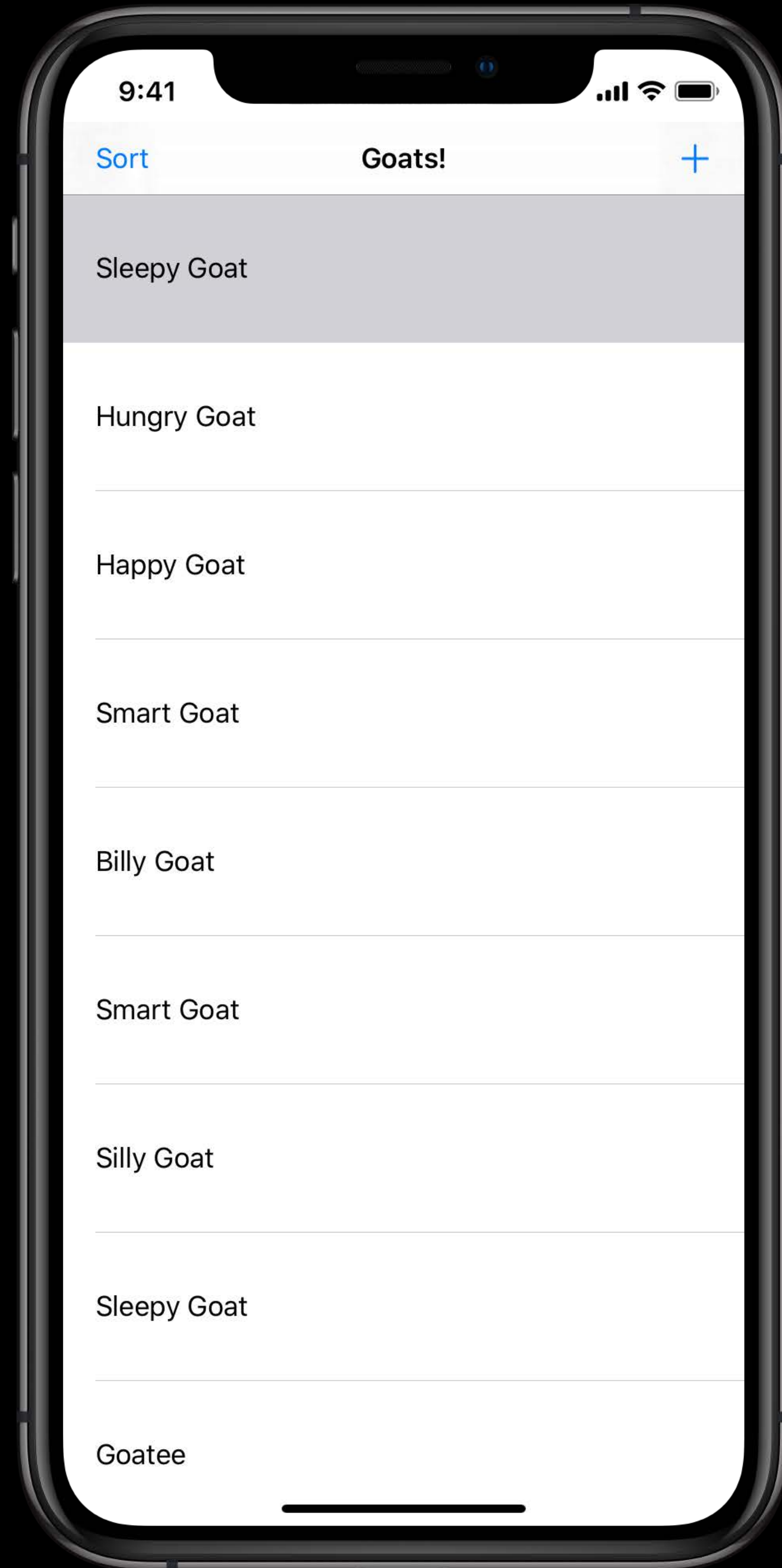
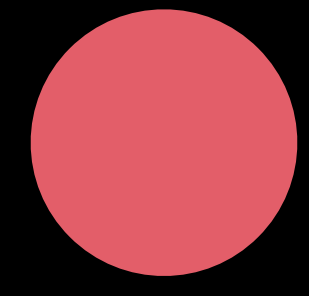
Smart Goat

Silly Goat

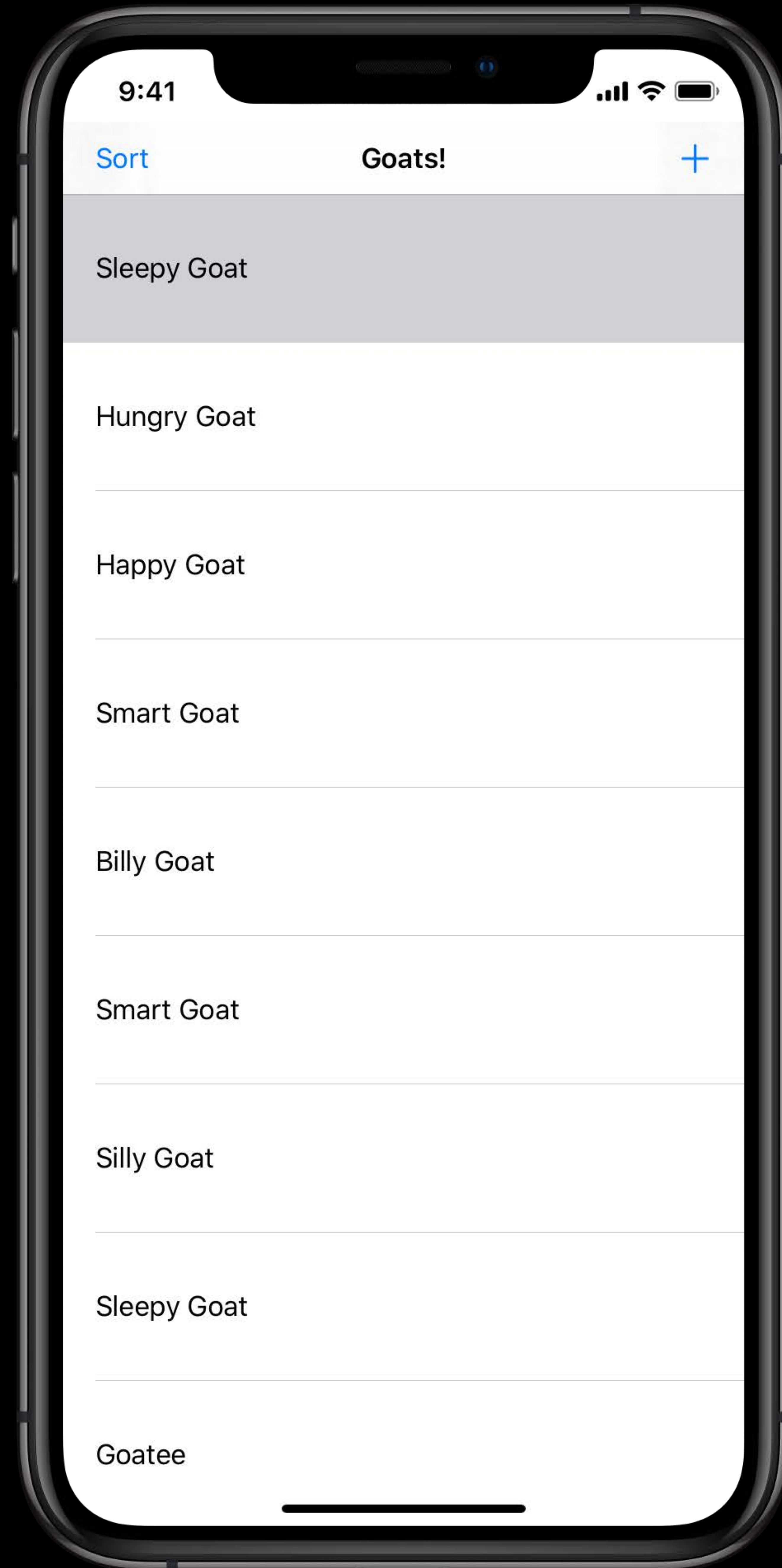
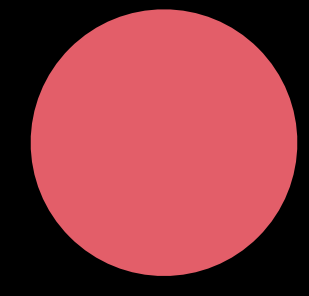
Sleepy Goat

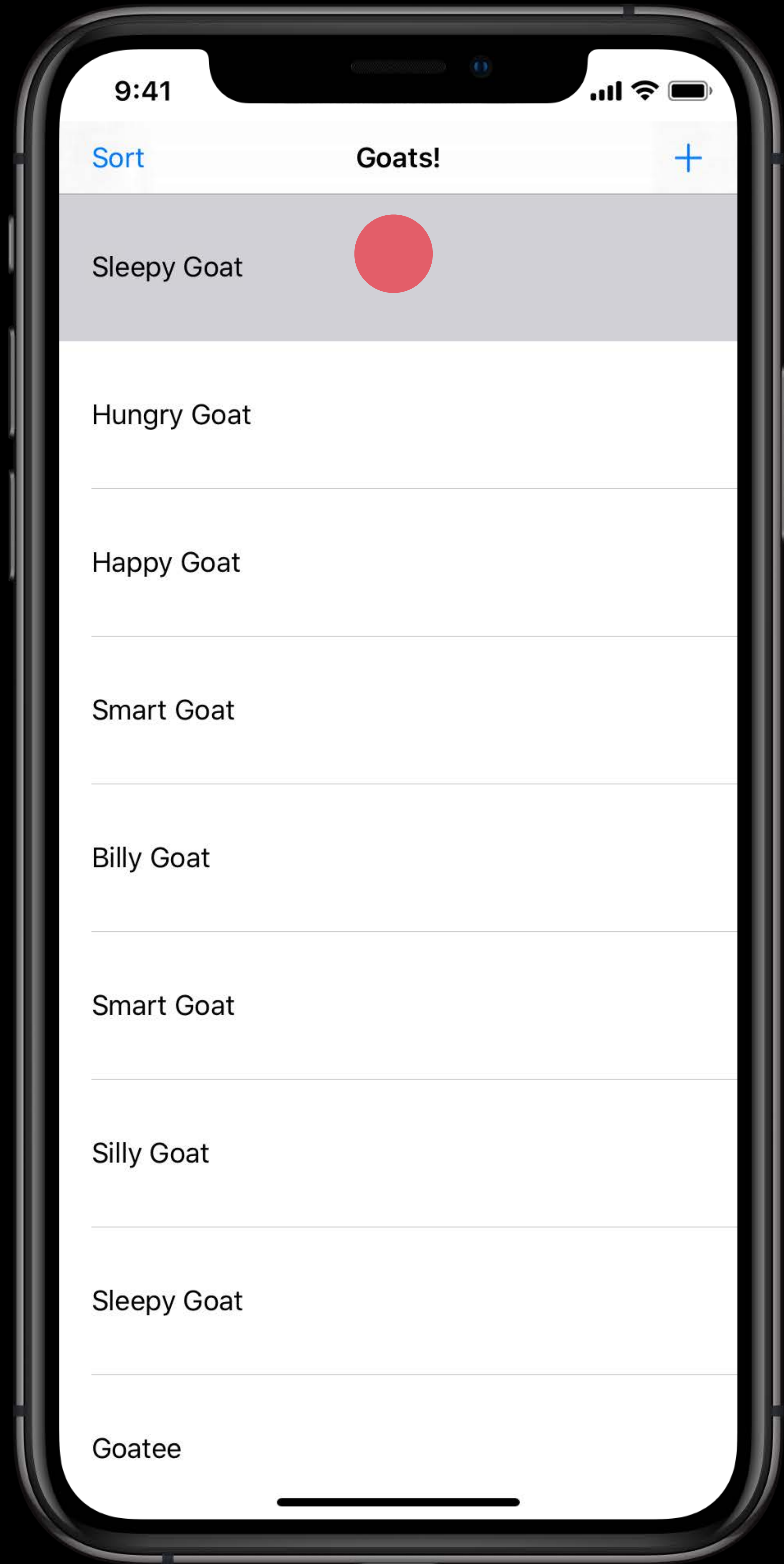
Goatee

# Dispatch Queue



# Dispatch Queue





9:41

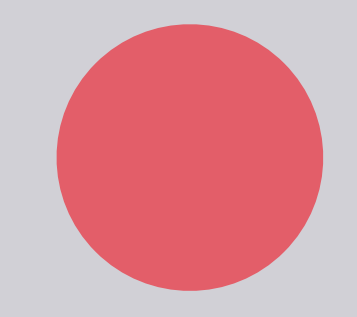


Sort

Goats!



Sleepy Goat



Hungry Goat

Happy Goat

Smart Goat

Billy Goat

Smart Goat

Silly Goat

Sleepy Goat

Goatee

# Dispatch Queue



# Dispatch Queue



# Dispatch Queue

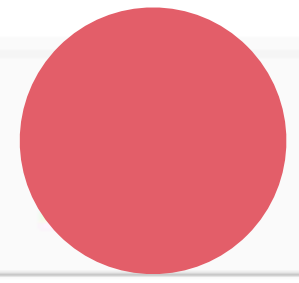




# Dispatch Queue



9:41



Goats!



Silly Goat

---

Mountain Goat

---

Happy Goat

---

Mountain Goat

---

Smart Goat

---

Sleepy Goat

---

Goatee

---

Silly Goat

---

Hungry Goat

9:41

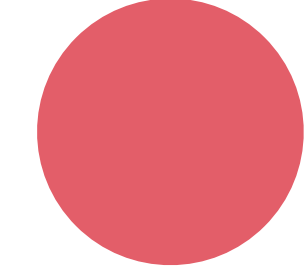


Sort

Goats!



Silly Goat



Mountain Goat

Happy Goat

Mountain Goat

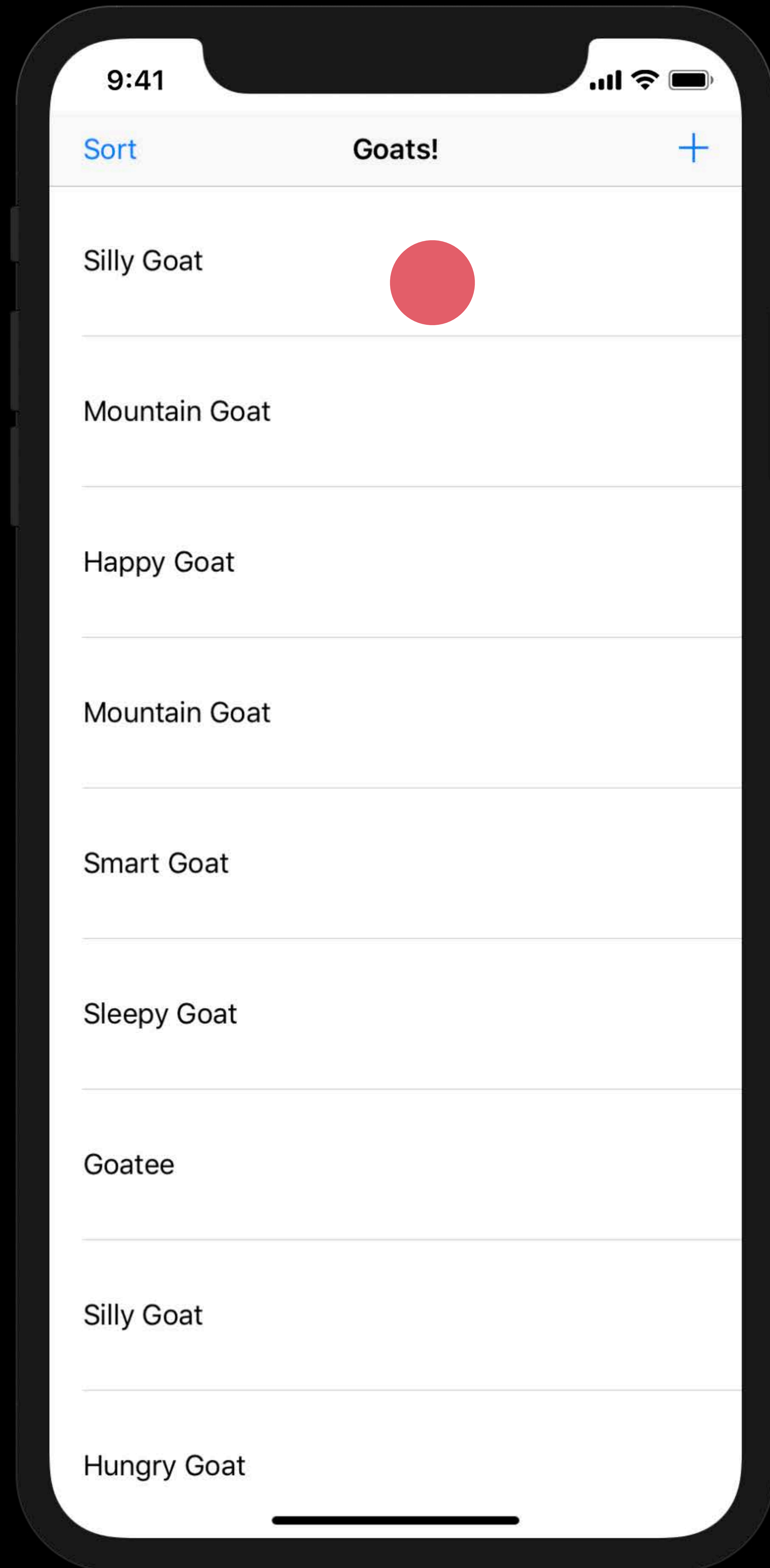
Smart Goat

Sleepy Goat

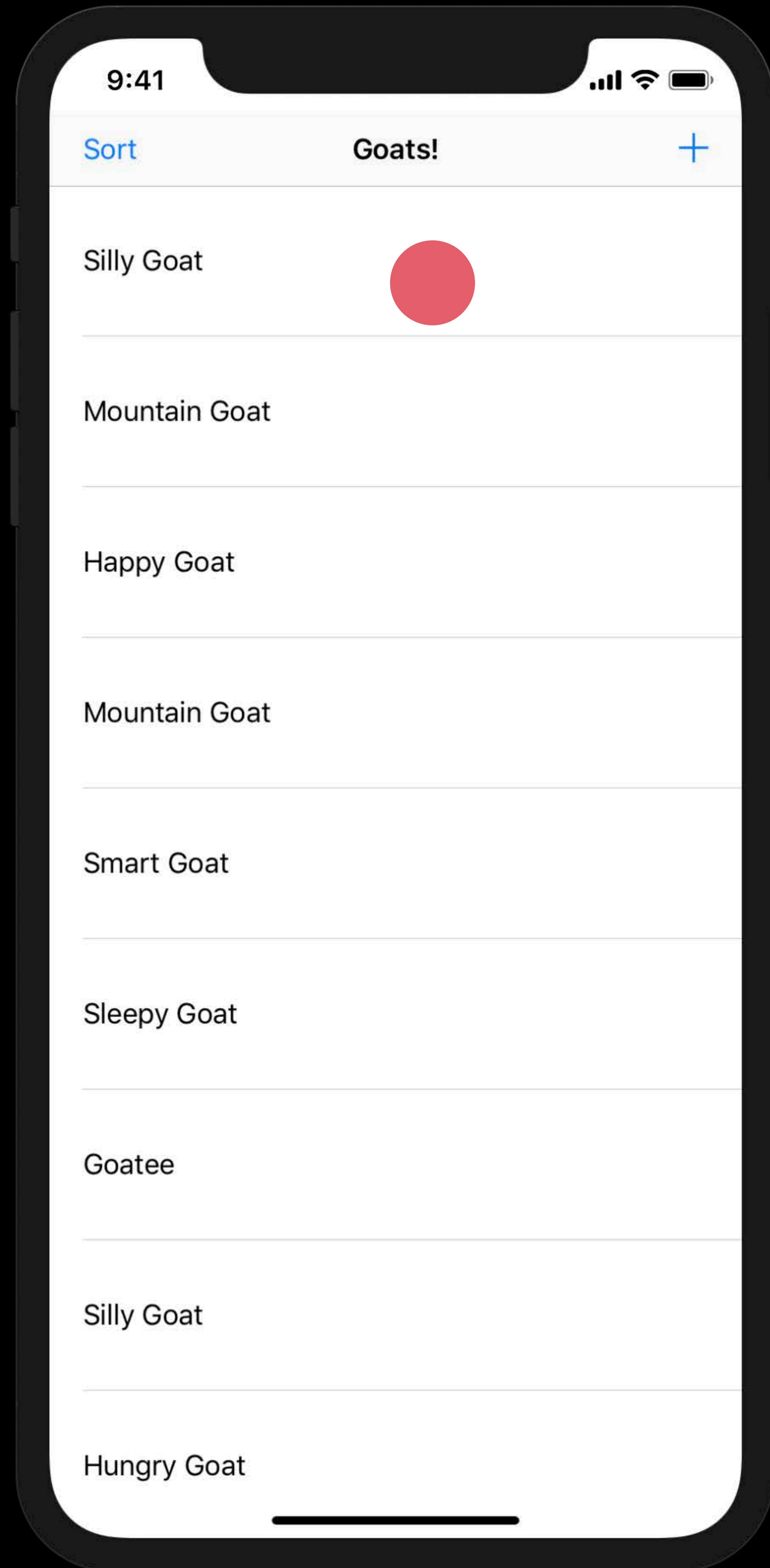
Goatee

Silly Goat

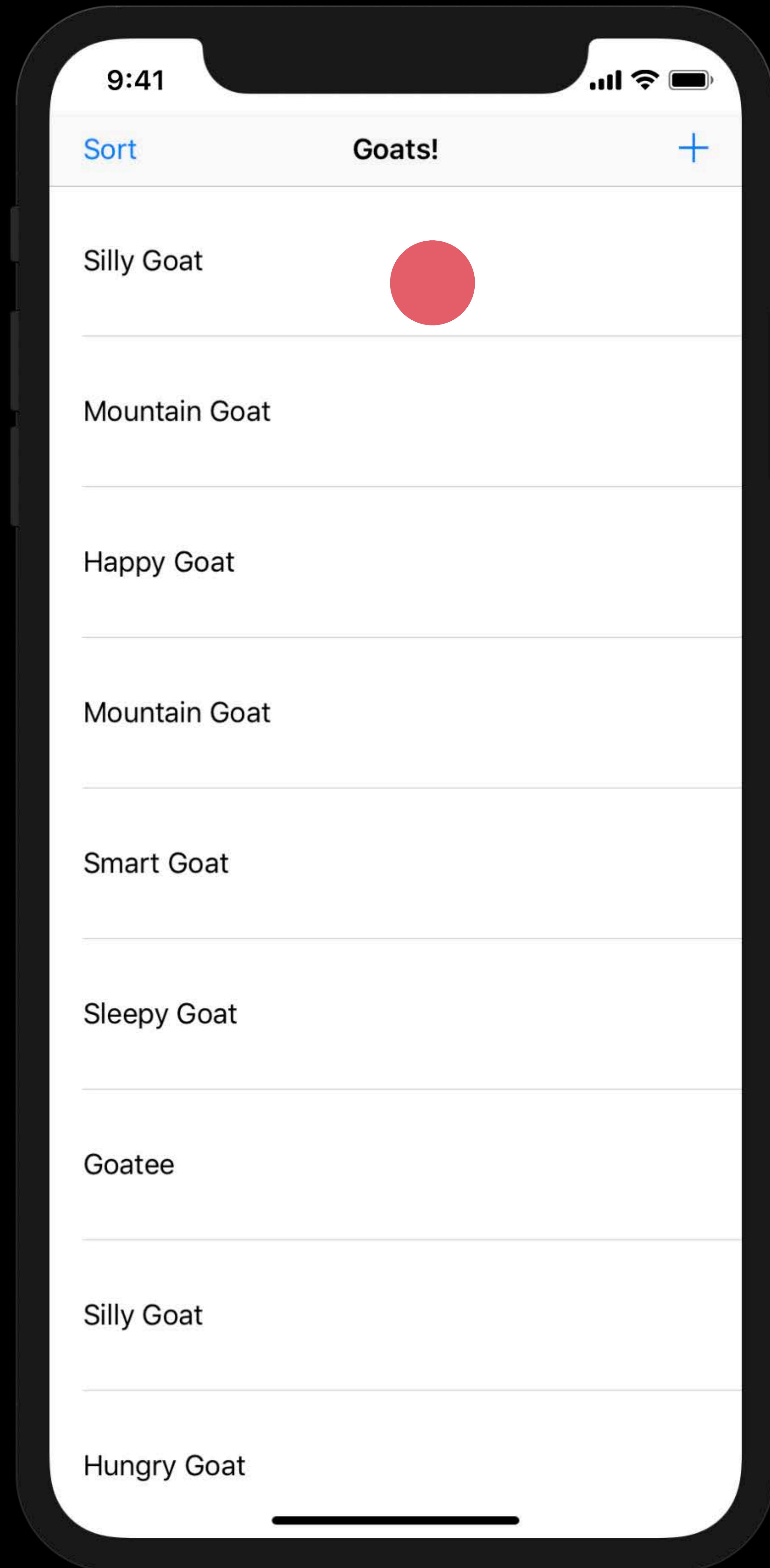
Hungry Goat



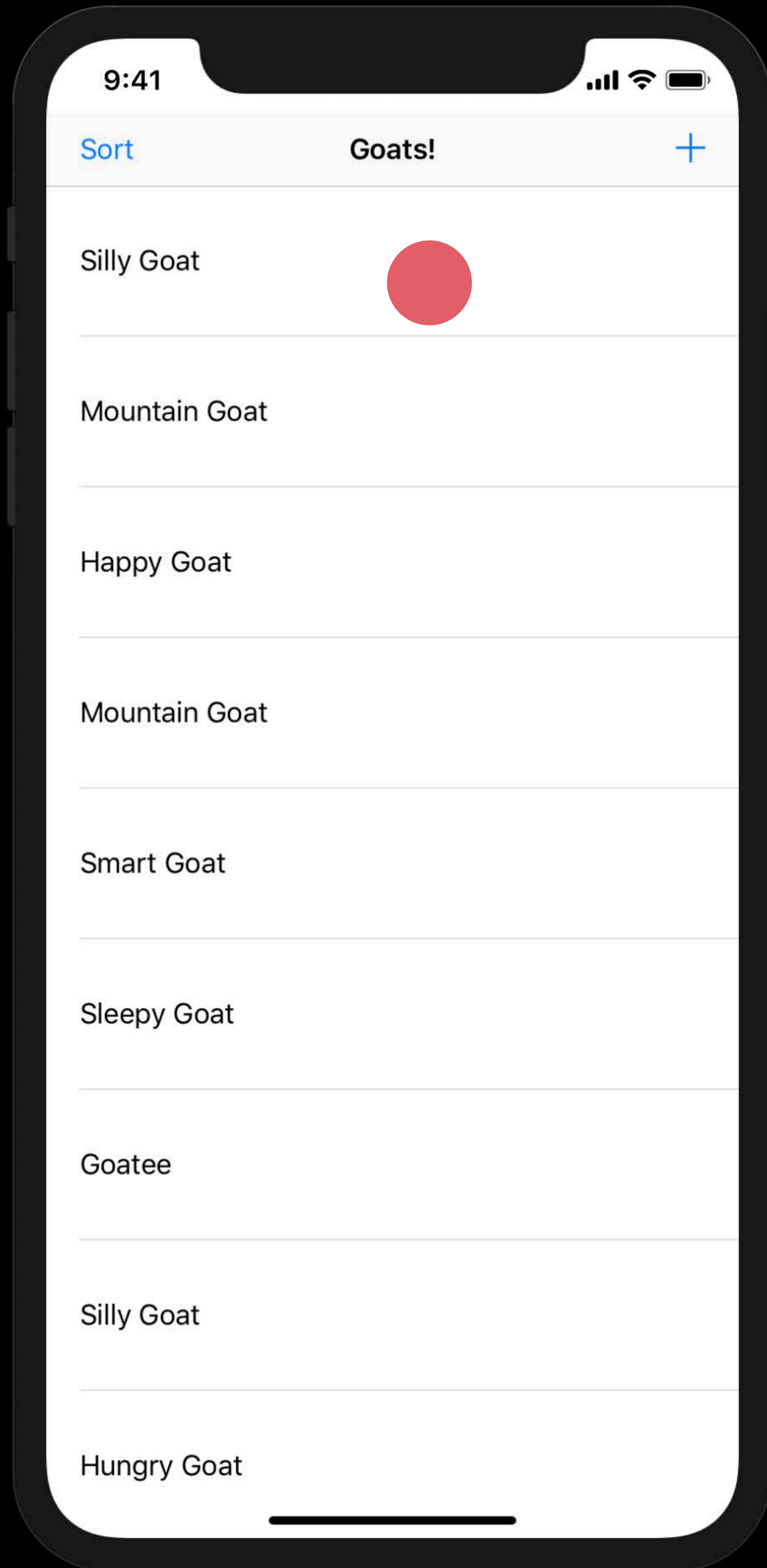
Moving to "UI"



Moving to "UI"



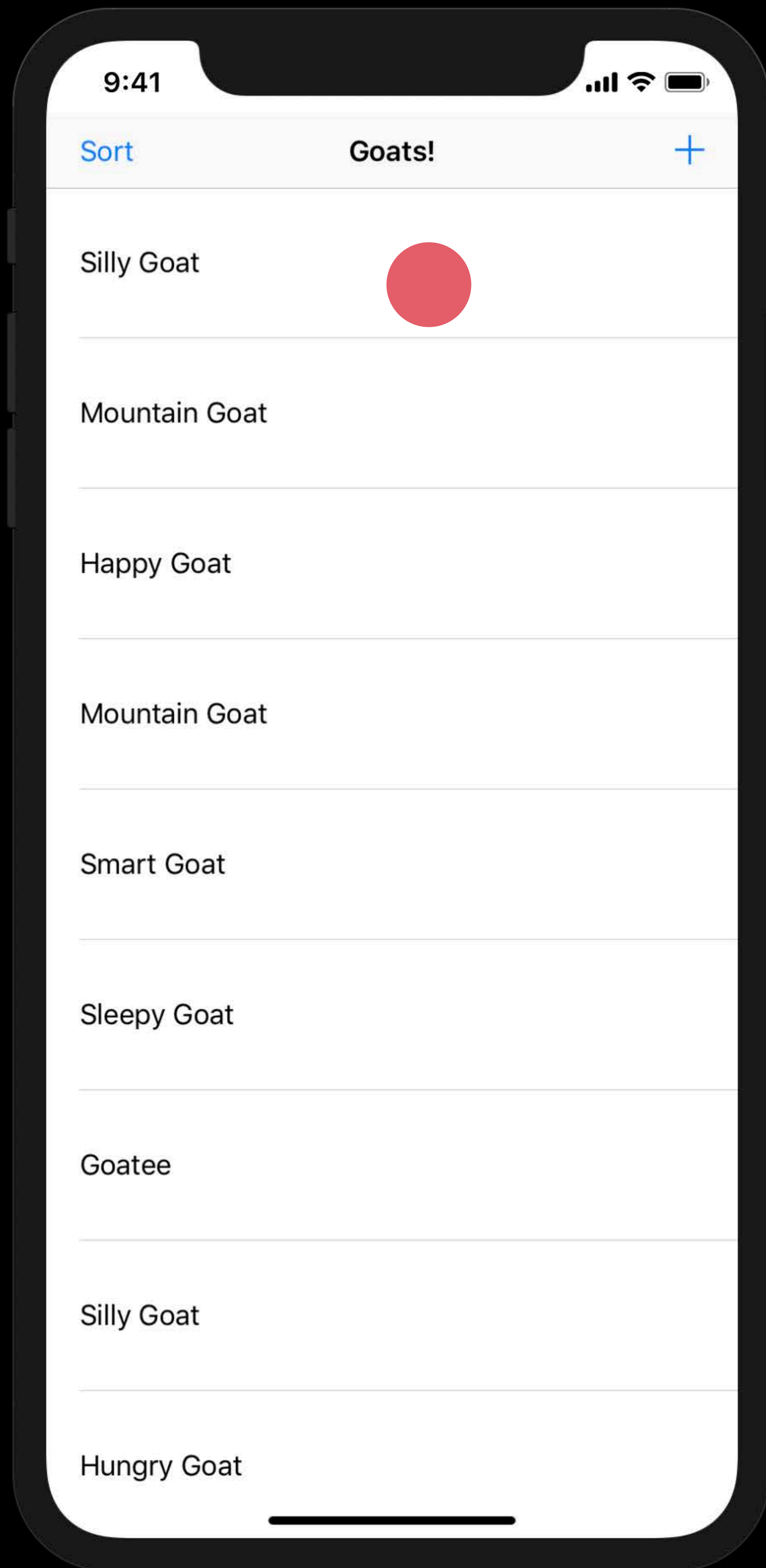
Moving to "UI"



Moving to "UI"

Executing in "Get Goats"

Dispatch Queue

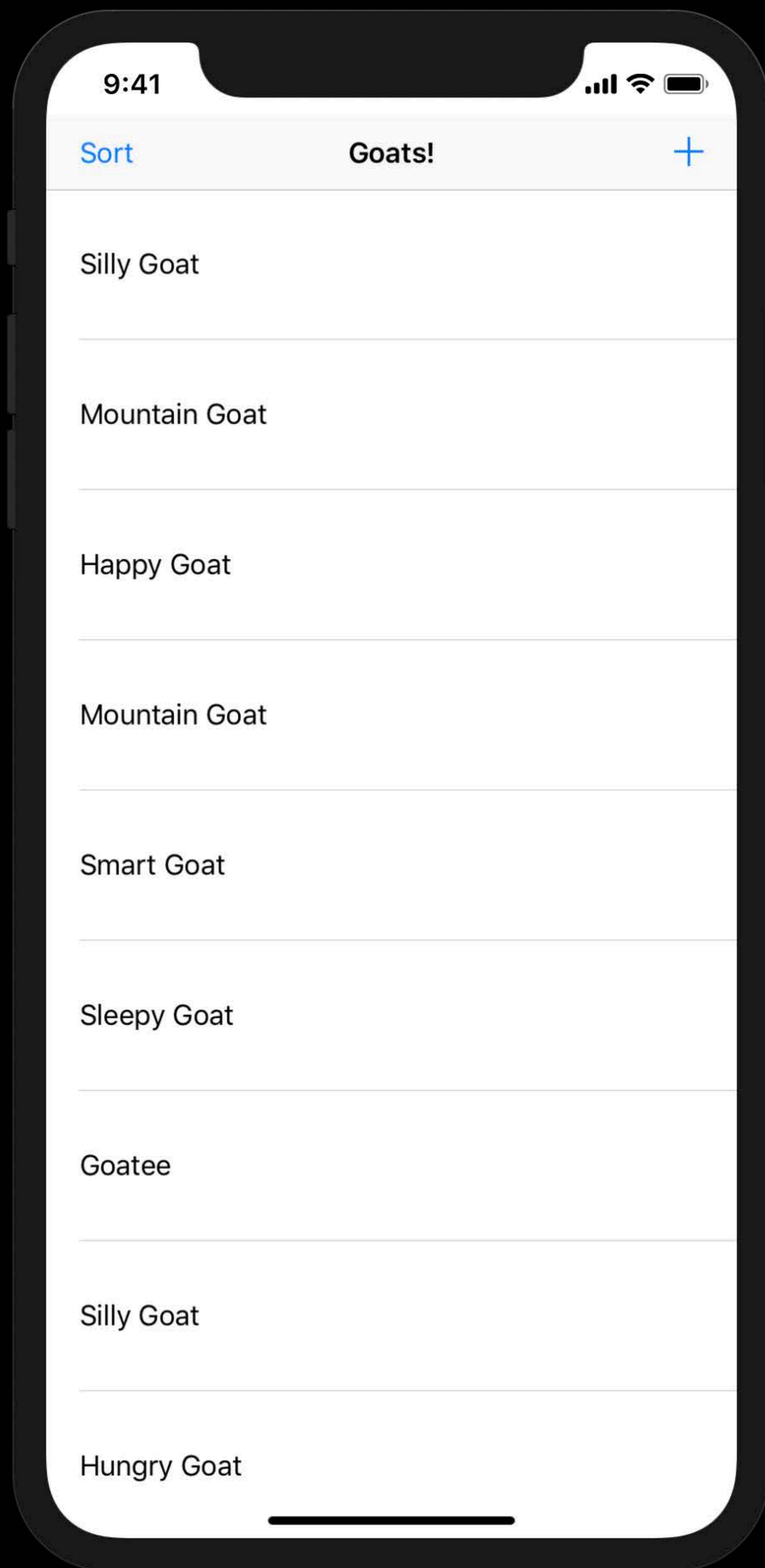
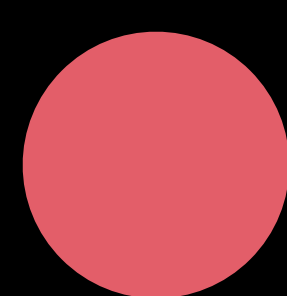


Moving to "UI"

Executing in "Get Goats"



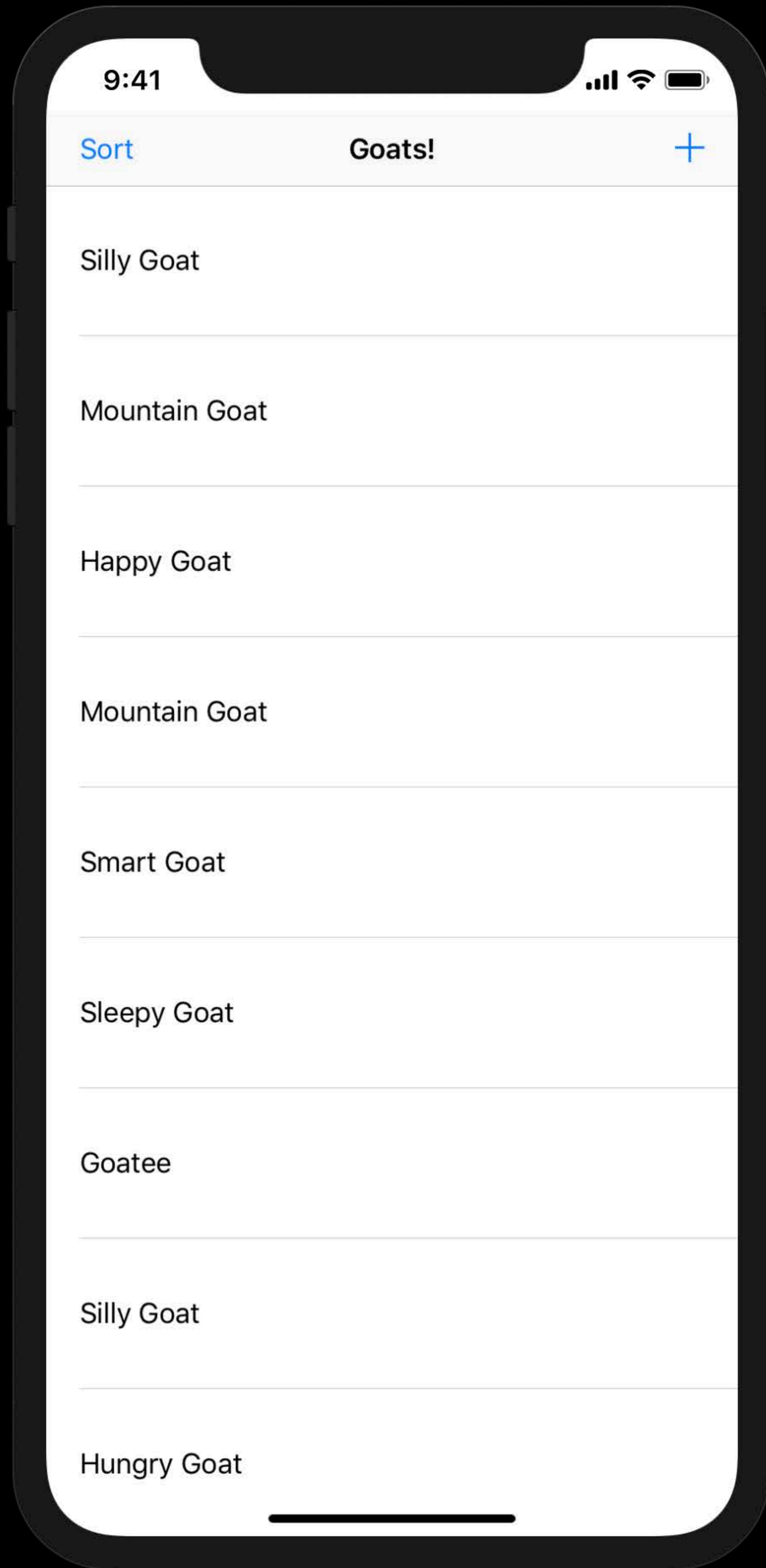
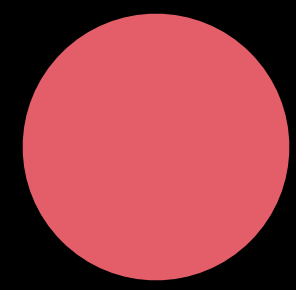
Dispatch Queue



Moving to "UI"

Executing in "Get Goats"

Dispatch Queue

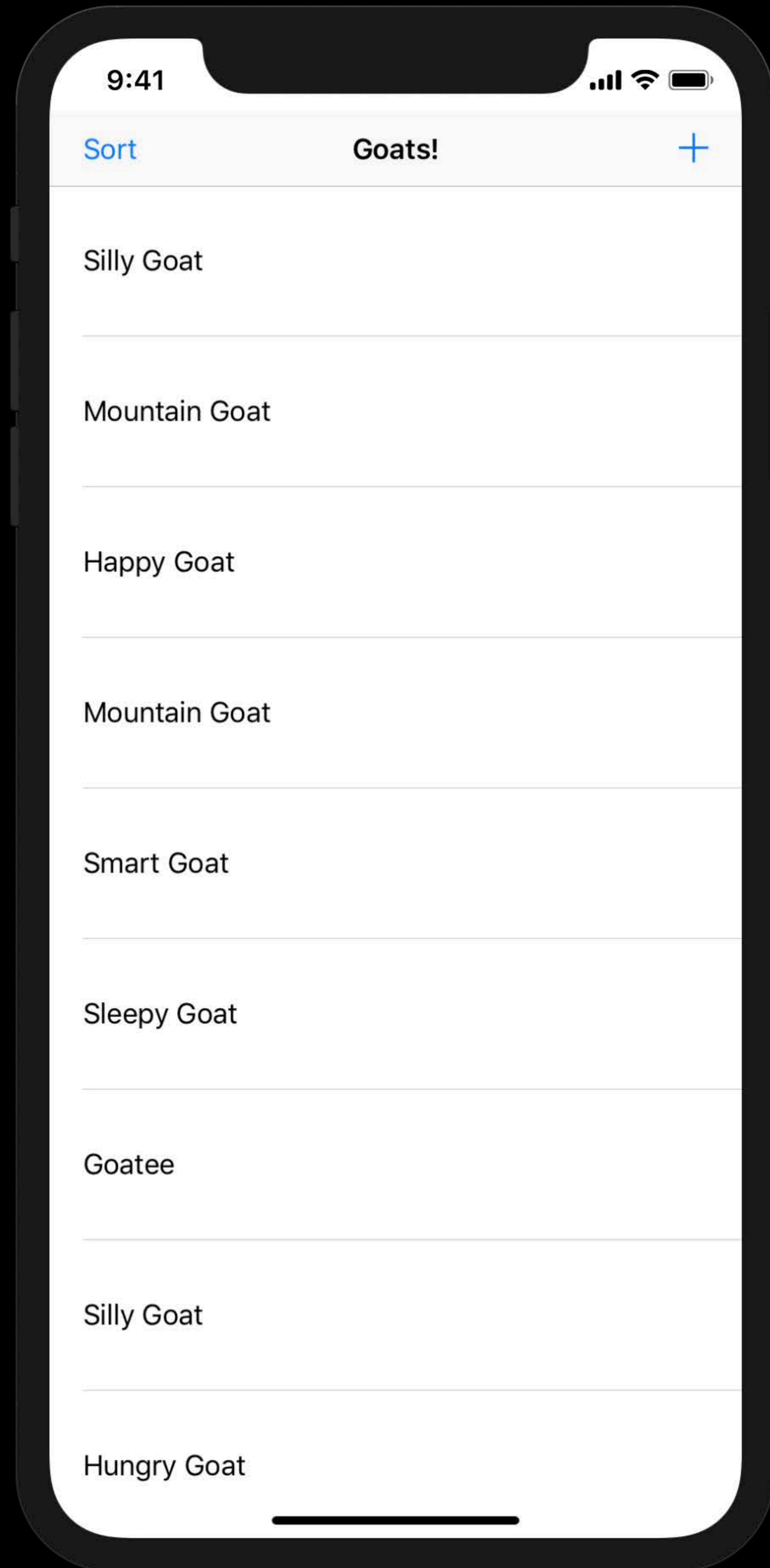


Moving to "UI"

Executing in "Get Goats"

Moving to "Background"

Dispatch Queue

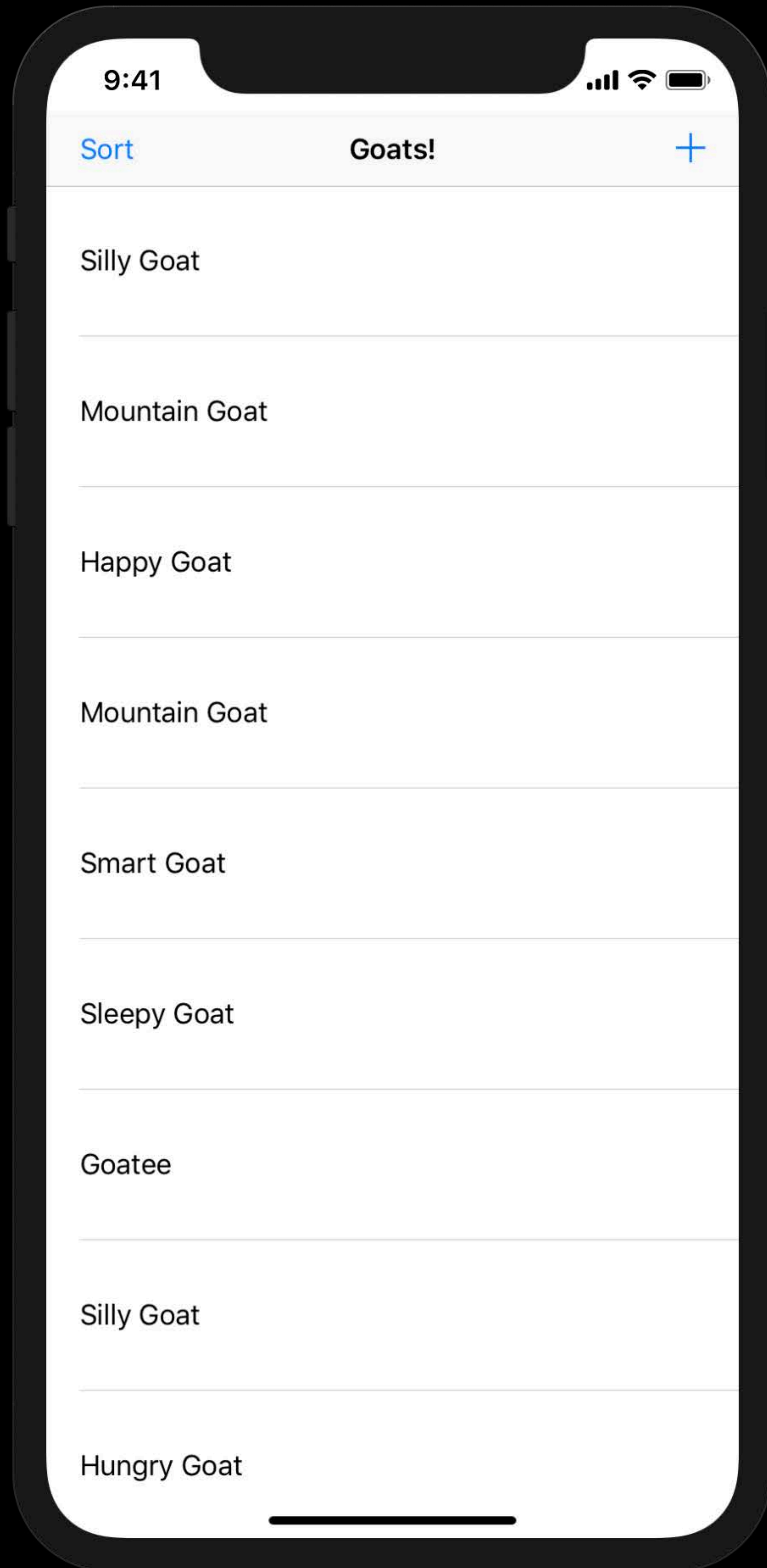
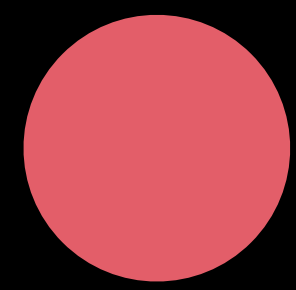


Moving to "UI"

Executing in "Get Goats"

Moving to "Background"

Dispatch Queue

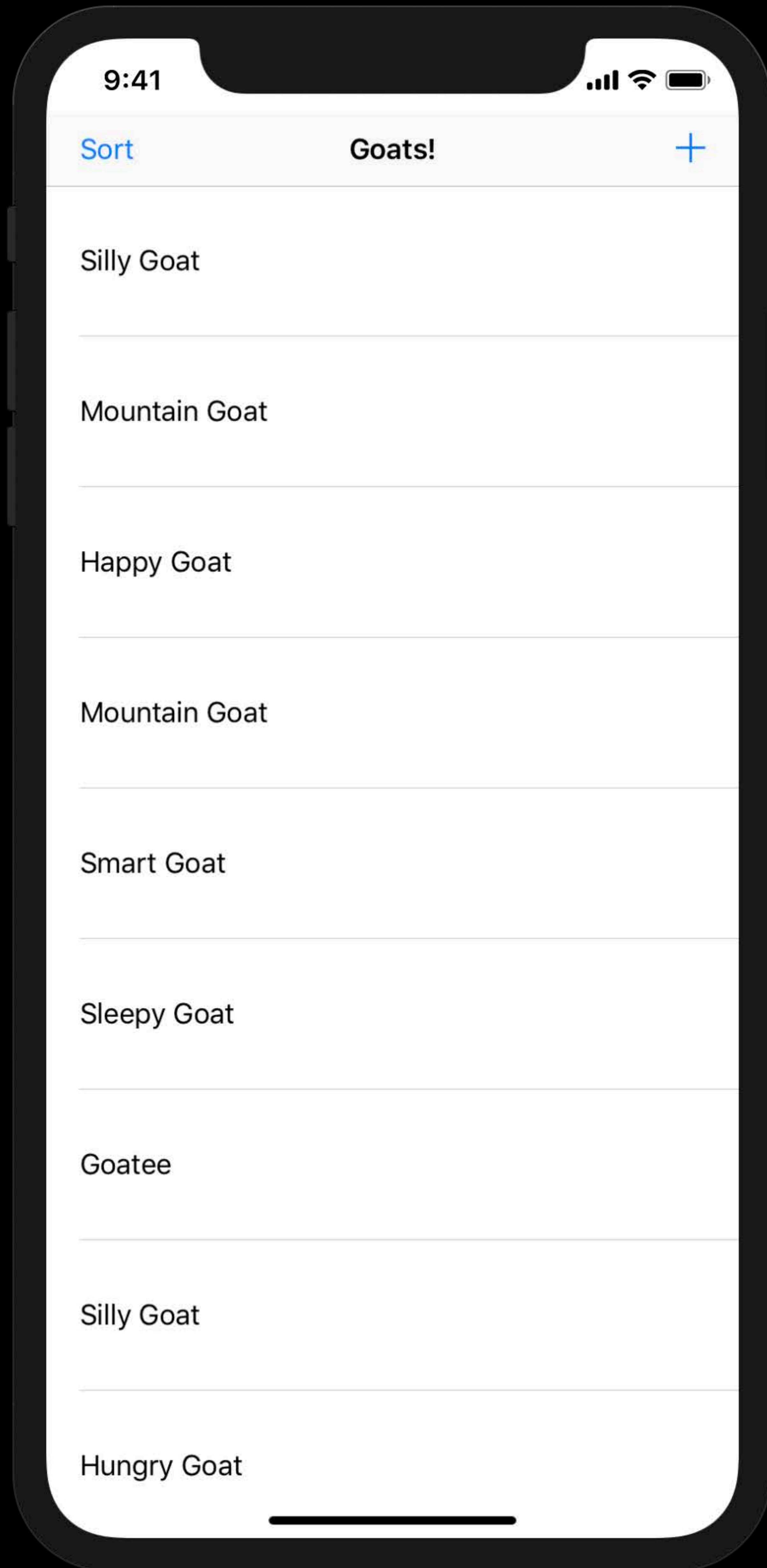
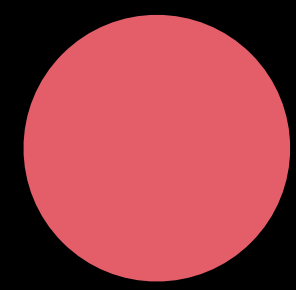


Moving to "UI"

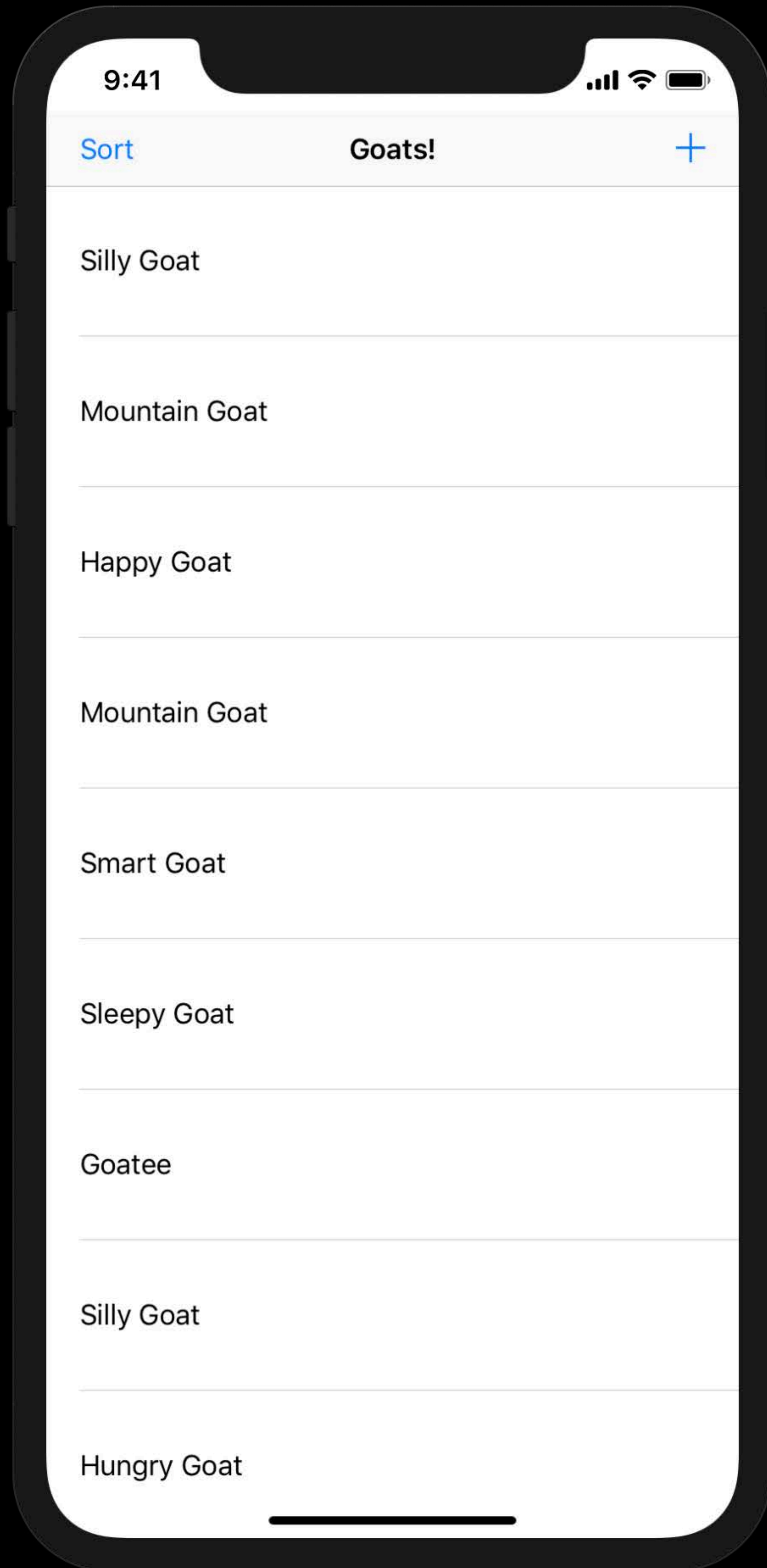
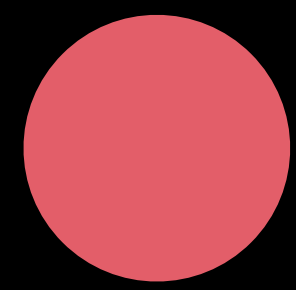
Executing in "Get Goats"

Moving to "Background"

Dispatch Queue



Dispatch Queue



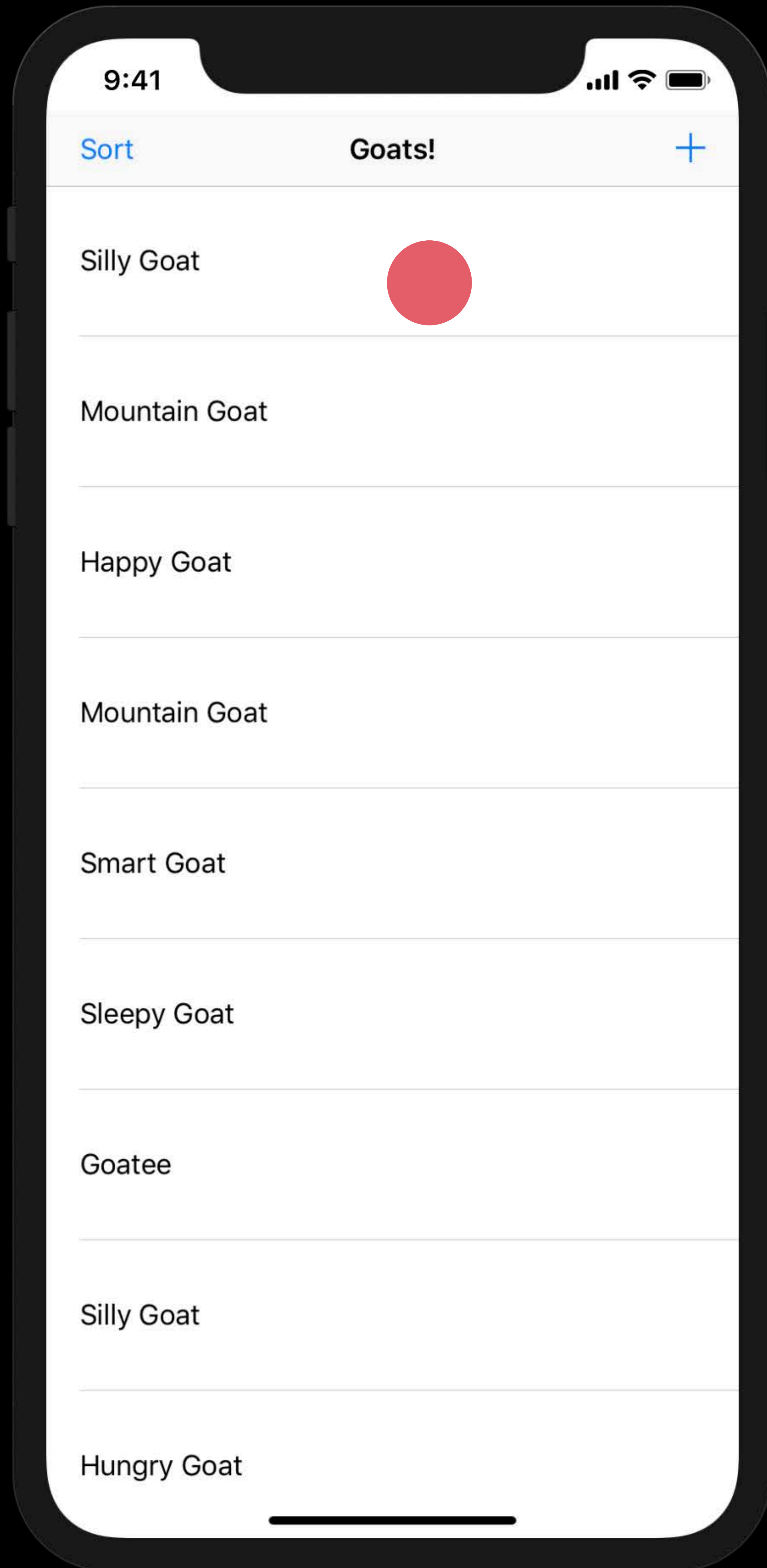
Moving to "UI"

Executing in "Get Goats"

Moving to "Background"

Executing in "Sorting"

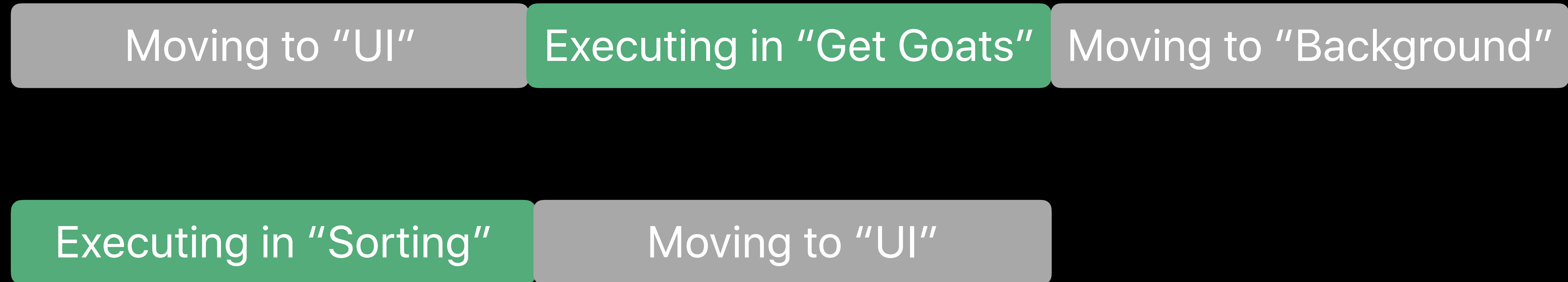
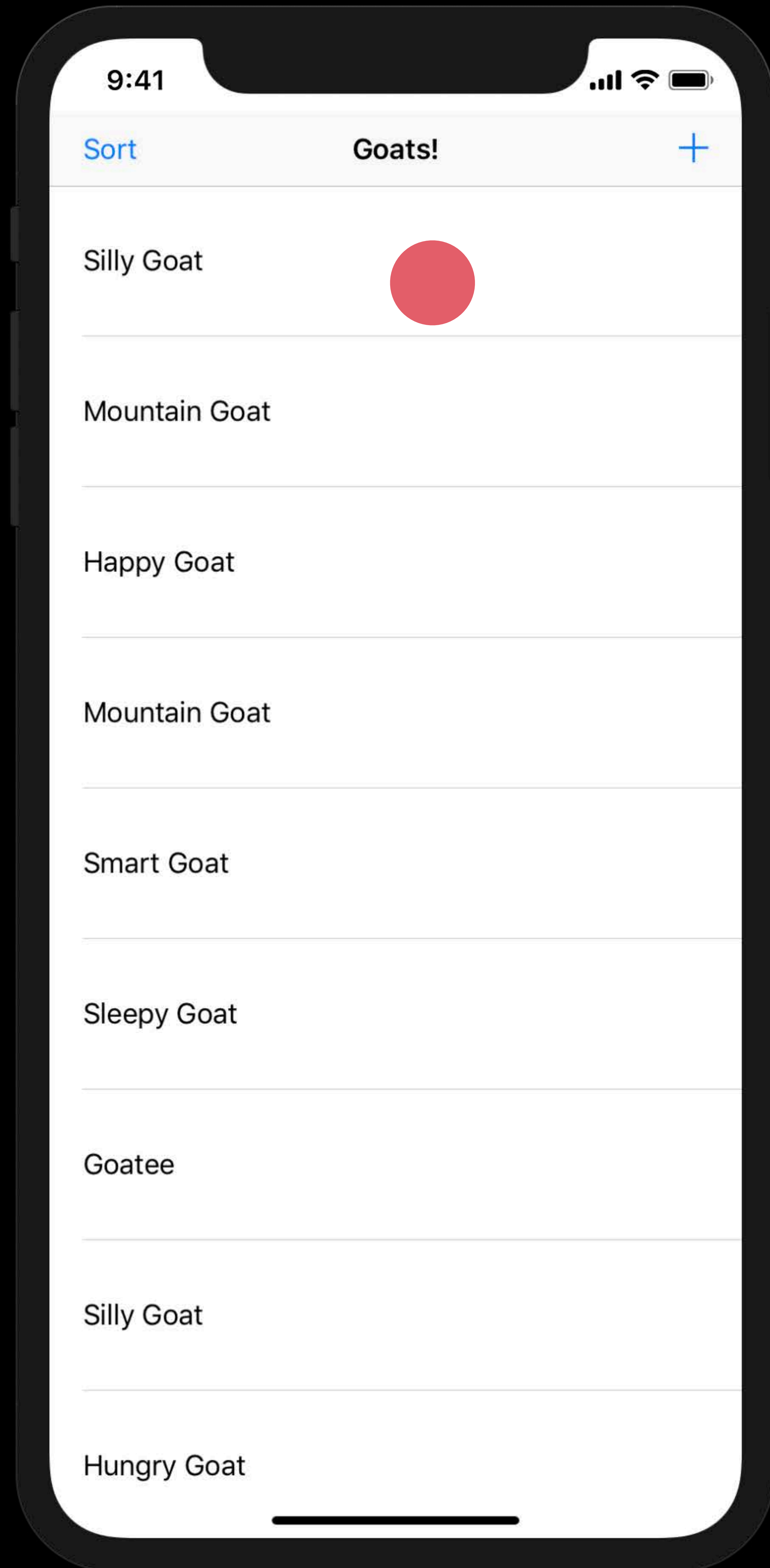
# Dispatch Queue



Moving to "UI"    Executing in "Get Goats"    Moving to "Background"

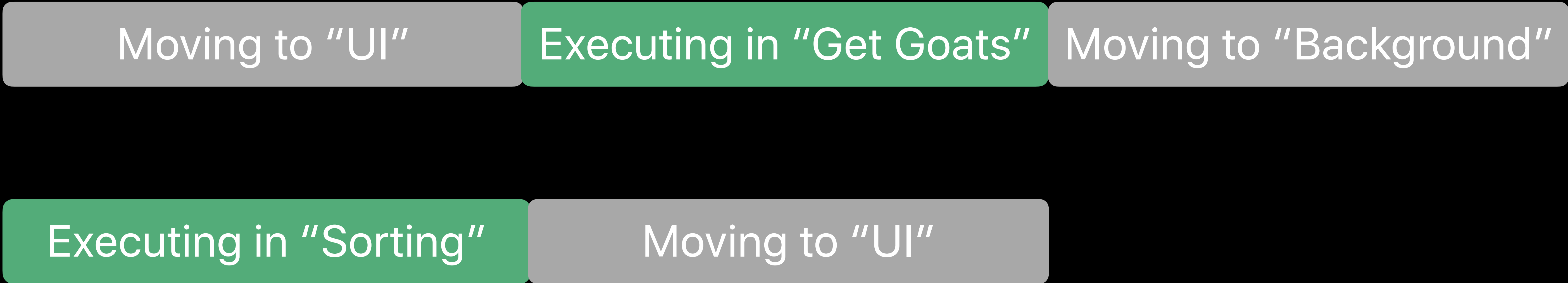
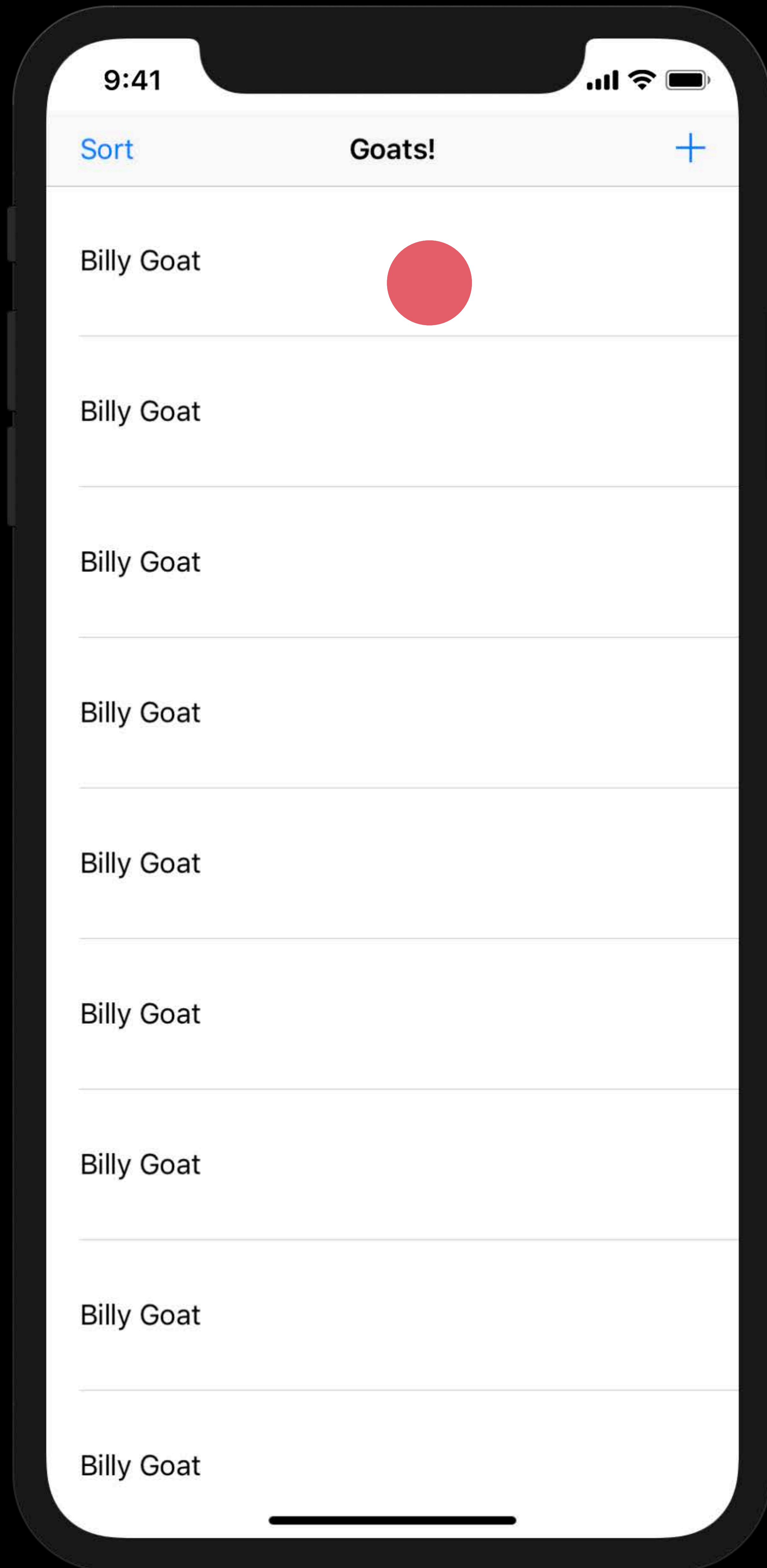
Executing in "Sorting"

# Dispatch Queue

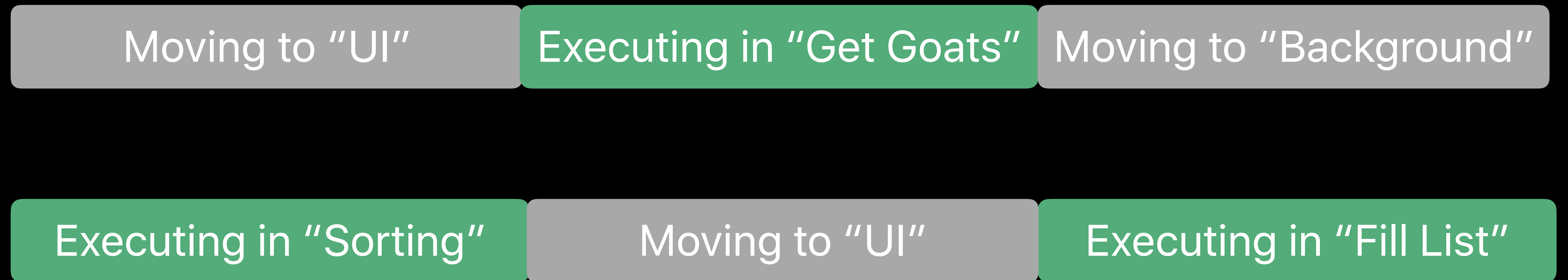
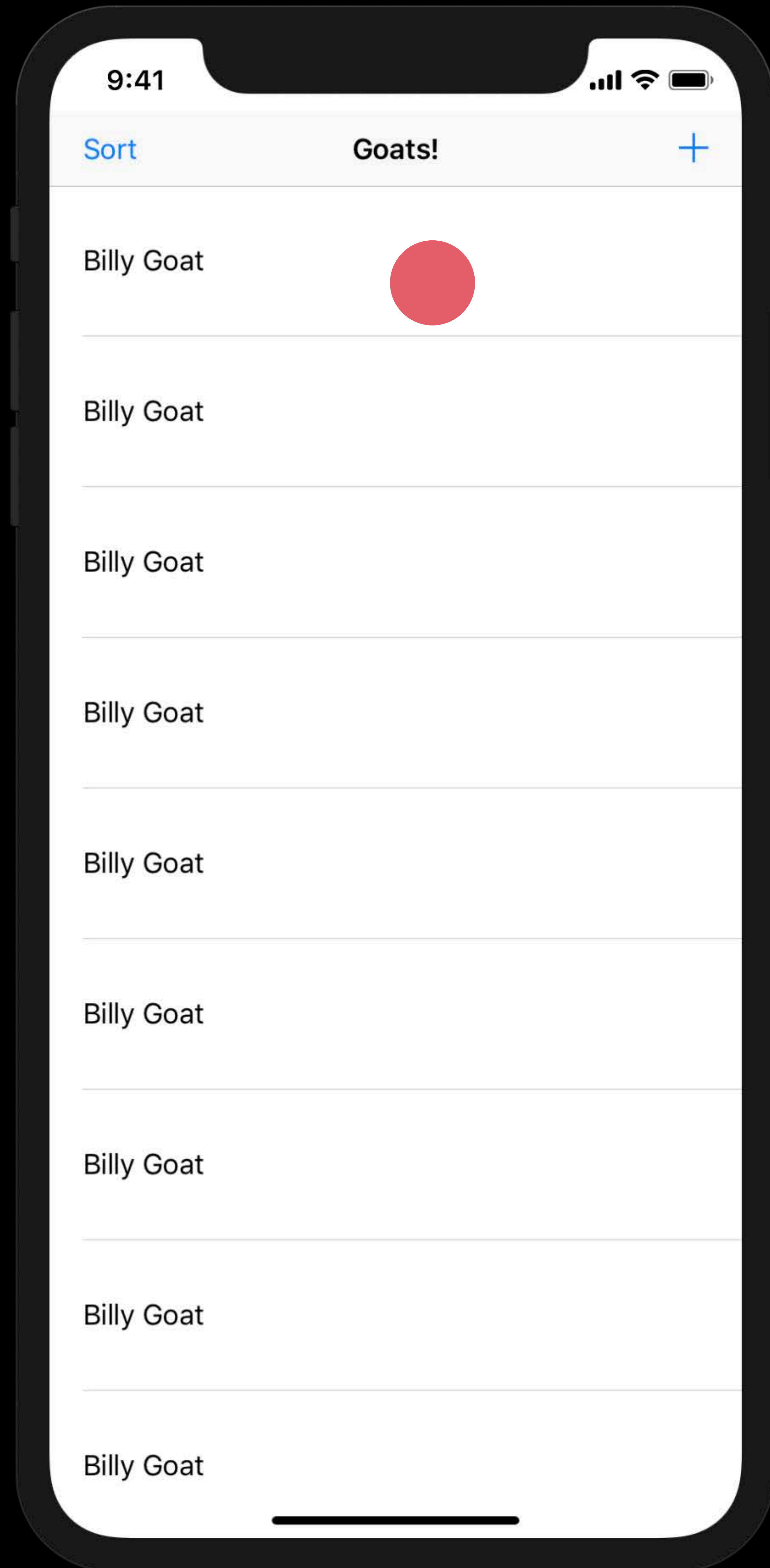




# Dispatch Queue



Dispatch Queue



# Desired Instrument

Instruments

Run 1 of 1 | 00:00:11

Track Filter All Tracks

00:04.800 00:04.900 00:05.000 00:05.100 00:05.200

**Mobile Agent A...ng Type Tracks** Display Agent  
Instrument

Sorting Agent

Moving to "Sor..." "Activating" at stop Sort Stop

**Display Stop** Display Agent  
String

**Goat List Stop** Display Agent  
String

Sorting Agent

**Sort Stop** Sorting Agent  
String

Moving to "Sor..." "Activating" at stop Sort Stop

**Mobile Agent Activity With Engineering Type Tracks** > Activity

Start	Duration	Agent Kind	Stop Kind	State	Activity
00:00.453.672	51.14 ms	Display Agent	Display Stop	Moving to	Moving to "Display Stop" in mode "Activating"
00:00.504.810	200.15 ms	Display Agent	Display Stop	executing	"Activating" at stop Display Stop
00:00.704.959	51.05 ms	Display Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Retrieve Goat List"

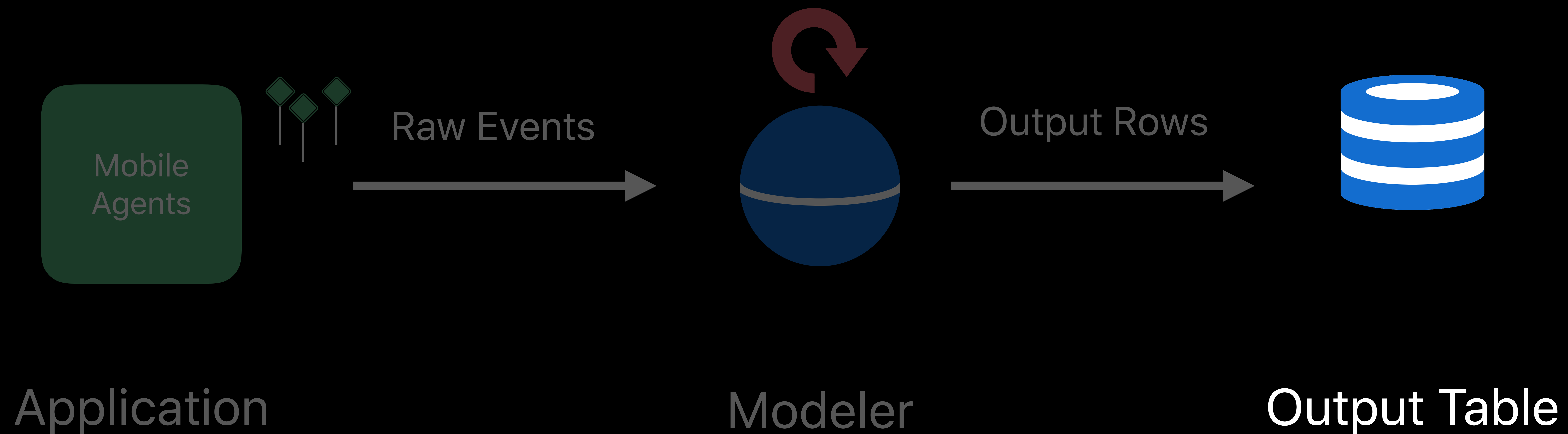
# Desired Instrument

Mobile Agent Activity With Engineering Type Tracks > Activity					
Start^	Duration	Agent Kind	Stop Kind	State	Activity
00:00.453.672	51.14 ms	Display Agent	Display Stop	Moving to	Moving to "Display Stop" in mode "Activating"
00:00.504.810	200.15 ms	Display Agent	Display Stop	executing	"Activating" at stop Display Stop
00:00.704.959	51.05 ms	Display Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Retrieve Goat List"
00:00.756.011	201.15 ms	Display Agent	Goat List Stop	executing	"Retrieve Goat List" at stop Goat List Stop
00:00.957.162	50.57 ms	Display Agent	Display Stop	Moving to	Moving to "Display Stop" in mode "Display Goat List"
00:01.007.737	205.01 ms	Display Agent	Display Stop	executing	"Display Goat List" at stop Display Stop
00:01.212.745	51.07 ms	Display Agent	Display Stop	Parking at	Parking at "Display Stop" in mode "Finished"
00:05.004.023	51.23 ms	Sorting Agent	Sort Stop	Moving to	Moving to "Sort Stop" in mode "Activating"
00:05.055.251	201.22 ms	Sorting Agent	Sort Stop	executing	"Activating" at stop Sort Stop
00:05.256.470	50.28 ms	Sorting Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Retrieve Goat List"
00:05.306.753	200.40 ms	Sorting Agent	Goat List Stop	executing	"Retrieve Goat List" at stop Goat List Stop
00:05.507.150	50.49 ms	Sorting Agent	Sort Stop	Moving to	Moving to "Sort Stop" in mode "Sort Goat List"
00:05.557.641	201.44 ms	Sorting Agent	Sort Stop	executing	"Sort Goat List" at stop Sort Stop
00:05.759.078	50.13 ms	Sorting Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Commit Goat List"
00:05.809.210	200.31 ms	Sorting Agent	Goat List Stop	executing	"Commit Goat List" at stop Goat List Stop
00:06.009.523	51.31 ms	Sorting Agent	Sort Stop	Parking at	Parking at "Sort Stop" in mode "Finished"
00:06.060.993	51.22 ms	Display Agent	Display Stop	Moving to	Moving to "Display Stop" in mode "Activating"
00:06.112.213	201.28 ms	Display Agent	Display Stop	executing	"Activating" at stop Display Stop
00:06.313.492	50.44 ms	Display Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Retrieve Goat List"
00:06.363.934	200.75 ms	Display Agent	Goat List Stop	executing	"Retrieve Goat List" at stop Goat List Stop

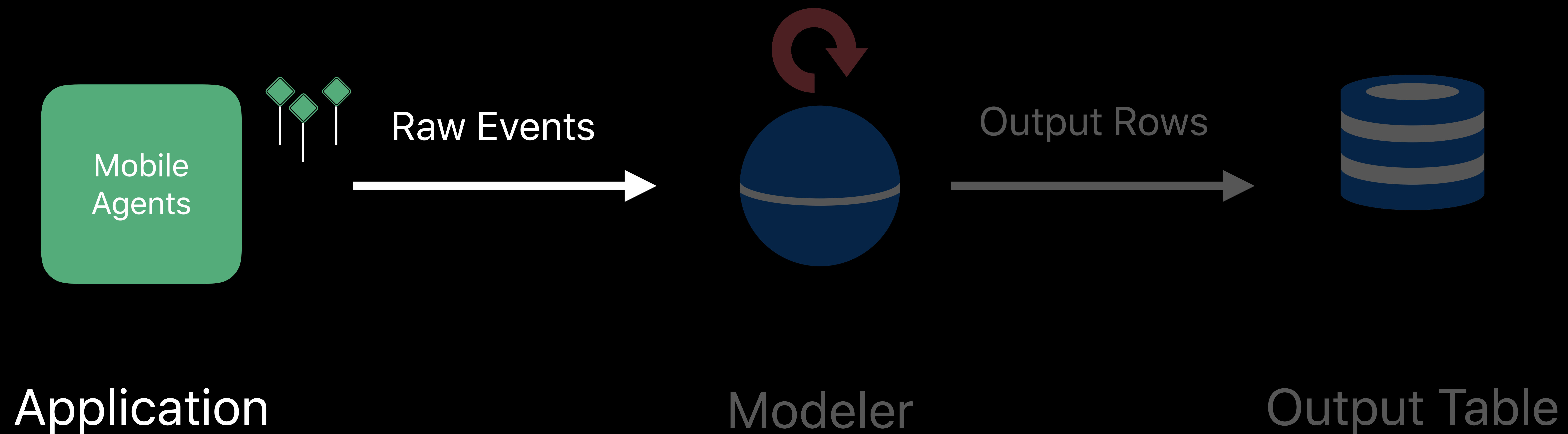
# Desired Instrument

Mobile Agent Activity With Engineering Type Tracks > Activity					
Start^	Duration	Agent Kind	Stop Kind	State	Activity
00:00.453.672	51.14 ms	Display Agent	Display Stop	Moving to	Moving to "Display Stop" in mode "Activating"
00:00.504.810	200.15 ms	Display Agent	Display Stop	executing	"Activating" at stop Display Stop
00:00.704.959	51.05 ms	Display Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Retrieve Goat List"
00:00.756.011	201.15 ms	Display Agent	Goat List Stop	executing	"Retrieve Goat List" at stop Goat List Stop
00:00.957.162	50.57 ms	Display Agent	Display Stop	Moving to	Moving to "Display Stop" in mode "Display Goat List"
00:01.007.737	205.01 ms	Display Agent	Display Stop	executing	"Display Goat List" at stop Display Stop
00:01.212.745	51.07 ms	Display Agent	Display Stop	Parking at	Parking at "Display Stop" in mode "Finished"
00:05.004.023	51.23 ms	Sorting Agent	Sort Stop	Moving to	Moving to "Sort Stop" in mode "Activating"
00:05.055.251	201.22 ms	Sorting Agent	Sort Stop	executing	"Activating" at stop Sort Stop
00:05.256.470	50.28 ms	Sorting Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Retrieve Goat List"
00:05.306.753	200.40 ms	Sorting Agent	Goat List Stop	executing	"Retrieve Goat List" at stop Goat List Stop
00:05.507.150	50.49 ms	Sorting Agent	Sort Stop	Moving to	Moving to "Sort Stop" in mode "Sort Goat List"
00:05.557.641	201.44 ms	Sorting Agent	Sort Stop	executing	"Sort Goat List" at stop Sort Stop
00:05.759.078	50.13 ms	Sorting Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Commit Goat List"
00:05.809.210	200.31 ms	Sorting Agent	Goat List Stop	executing	"Commit Goat List" at stop Goat List Stop
00:06.009.523	51.31 ms	Sorting Agent	Sort Stop	Parking at	Parking at "Sort Stop" in mode "Finished"
00:06.060.993	51.22 ms	Display Agent	Display Stop	Moving to	Moving to "Display Stop" in mode "Activating"
00:06.112.213	201.28 ms	Display Agent	Display Stop	executing	"Activating" at stop Display Stop
00:06.313.492	50.44 ms	Display Agent	Goat List Stop	Moving to	Moving to "Goat List Stop" in mode "Retrieve Goat List"
00:06.363.934	200.75 ms	Display Agent	Goat List Stop	executing	"Retrieve Goat List" at stop Goat List Stop

# Modeling Schematic



# Modeling Schematic



# Event Stream



# Event Stream

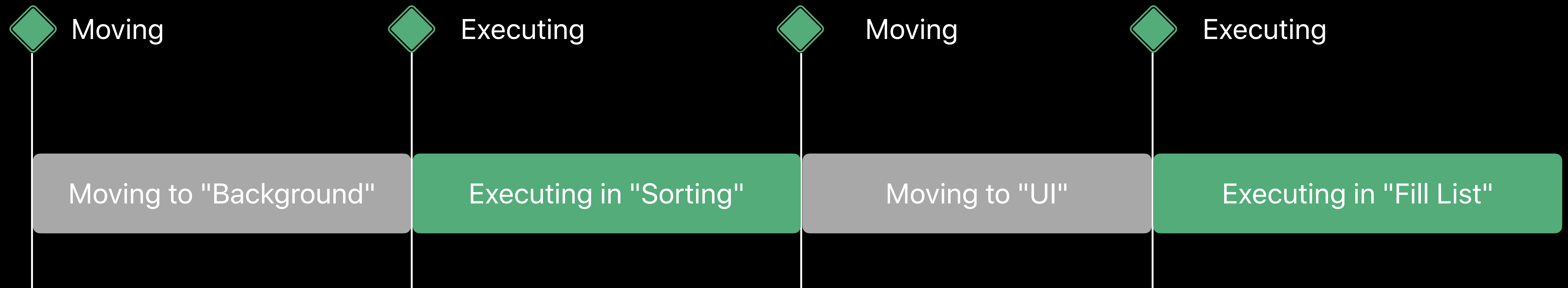
Moving to "Background"

Executing in "Sorting"

Moving to "UI"

Executing in "Fill List"

# Event Stream



```
// Mobile Agent Executing

func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %@ executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())

    execute()
}
```

```
// Mobile Agent Executing

func executeStop() {
    os_signpost(.event,
                log: MobileAgent.signpostHandleID,
                name: "Mobile Agent Exec",
                signpostID: signpostID,
                "Agent of type %@ executing mode %@. At stop %@",
                agentType(), currentMode, stop.agentStopType())

    execute()
}
```

```
// Mobile Agent Executing

func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %@ executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())

    execute()
}
```

```
// Mobile Agent Moving

internal func visitNextStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Moved",
        signpostID: signpostID,
        "Agent of type %@ received by %@ for mode %@",
        agent.agentType(),
        destination.agentStopType(),
        agent.currentMode)

    //...
}
```

```
// Mobile Agent Moving

internal func visitNextStop() {
    os_signpost(.event,
                log: MobileAgent.signpostHandleID,
                name: "Mobile Agent Moved",
                signpostID: signpostID,
                "Agent of type %@ received by %@ for mode %@",
                agent.agentType(),
                destination.agentStopType(),
                agent.currentMode)

    //...
}
```

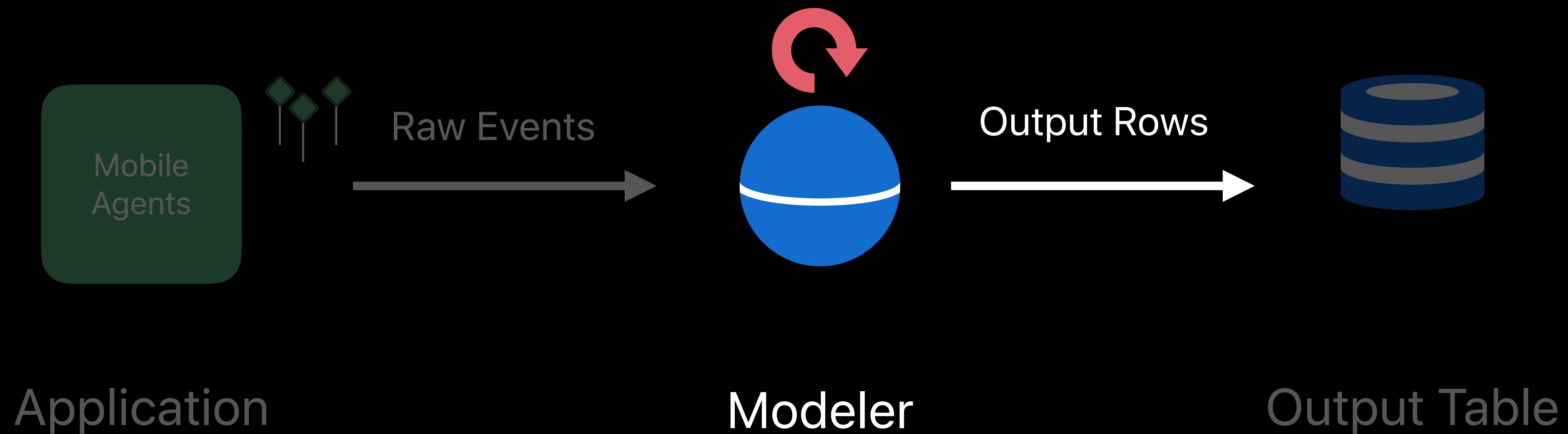
```
// Mobile Agent Moving

internal func visitNextStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Moved",
        signpostID: signpostID,
        "Agent of type %@ received by %@ for mode %@",
        agent.agentType(),
        destination.agentStopType(),
        agent.currentMode)

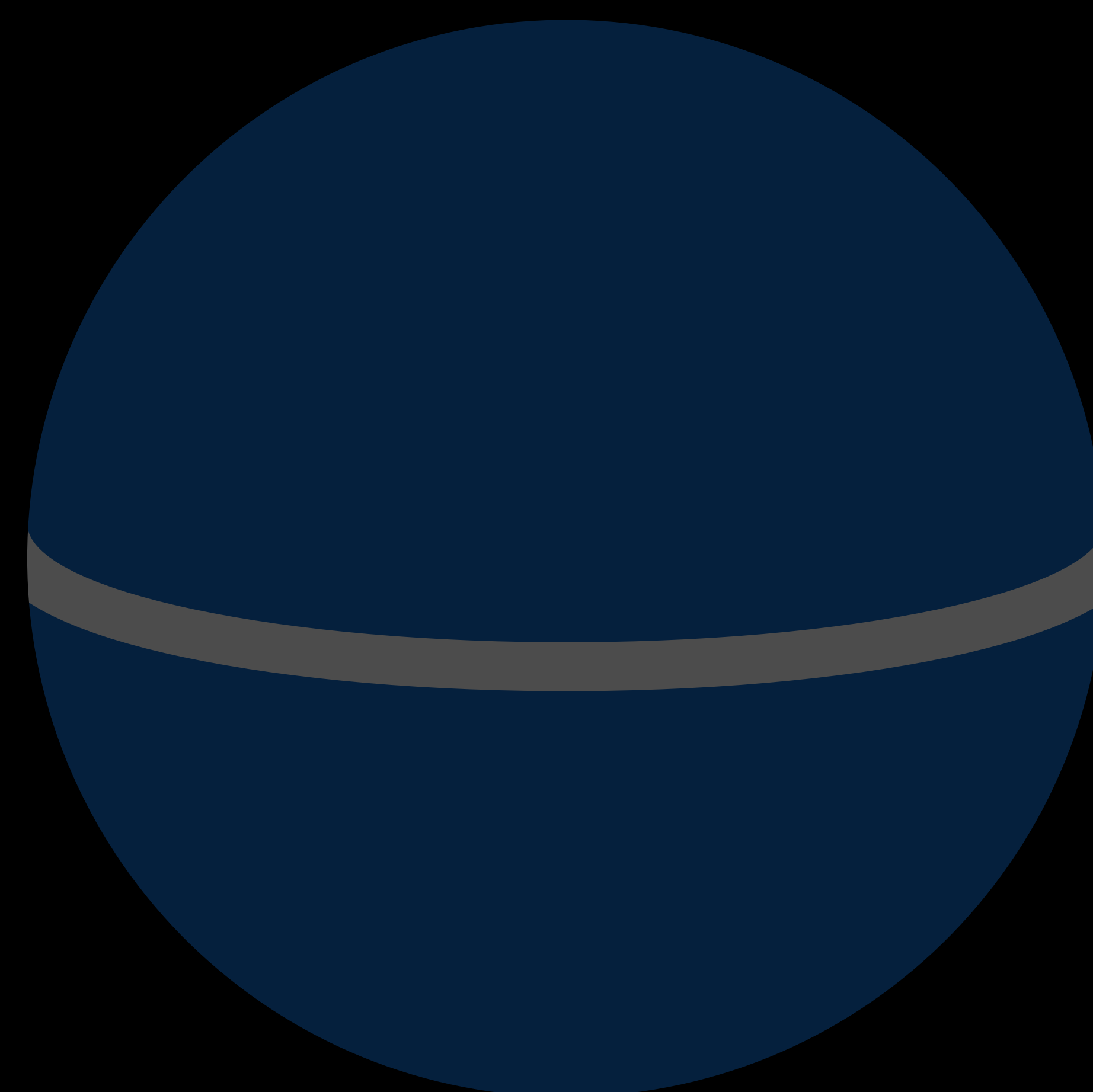
    //...
}
```



# Modeling Schematic

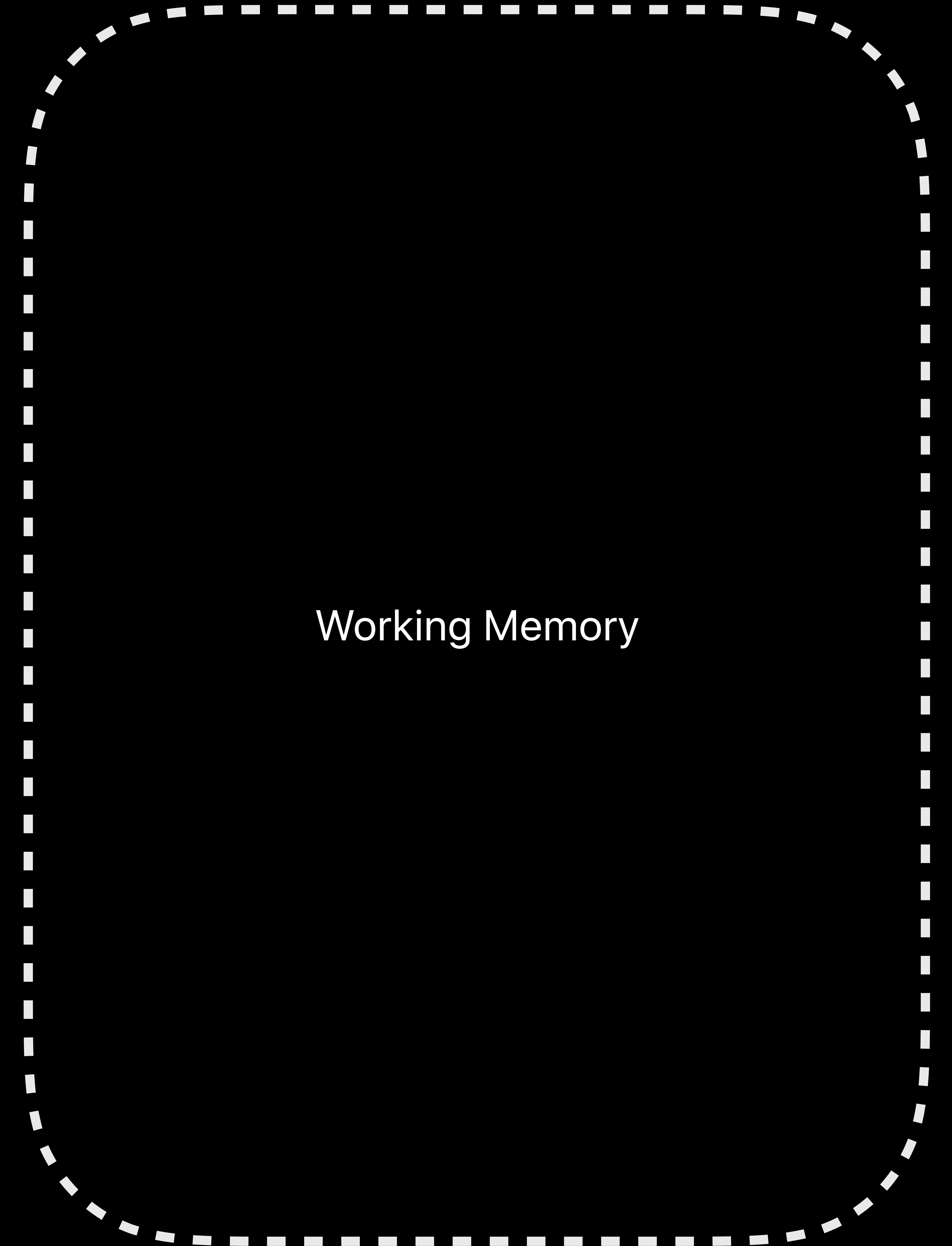
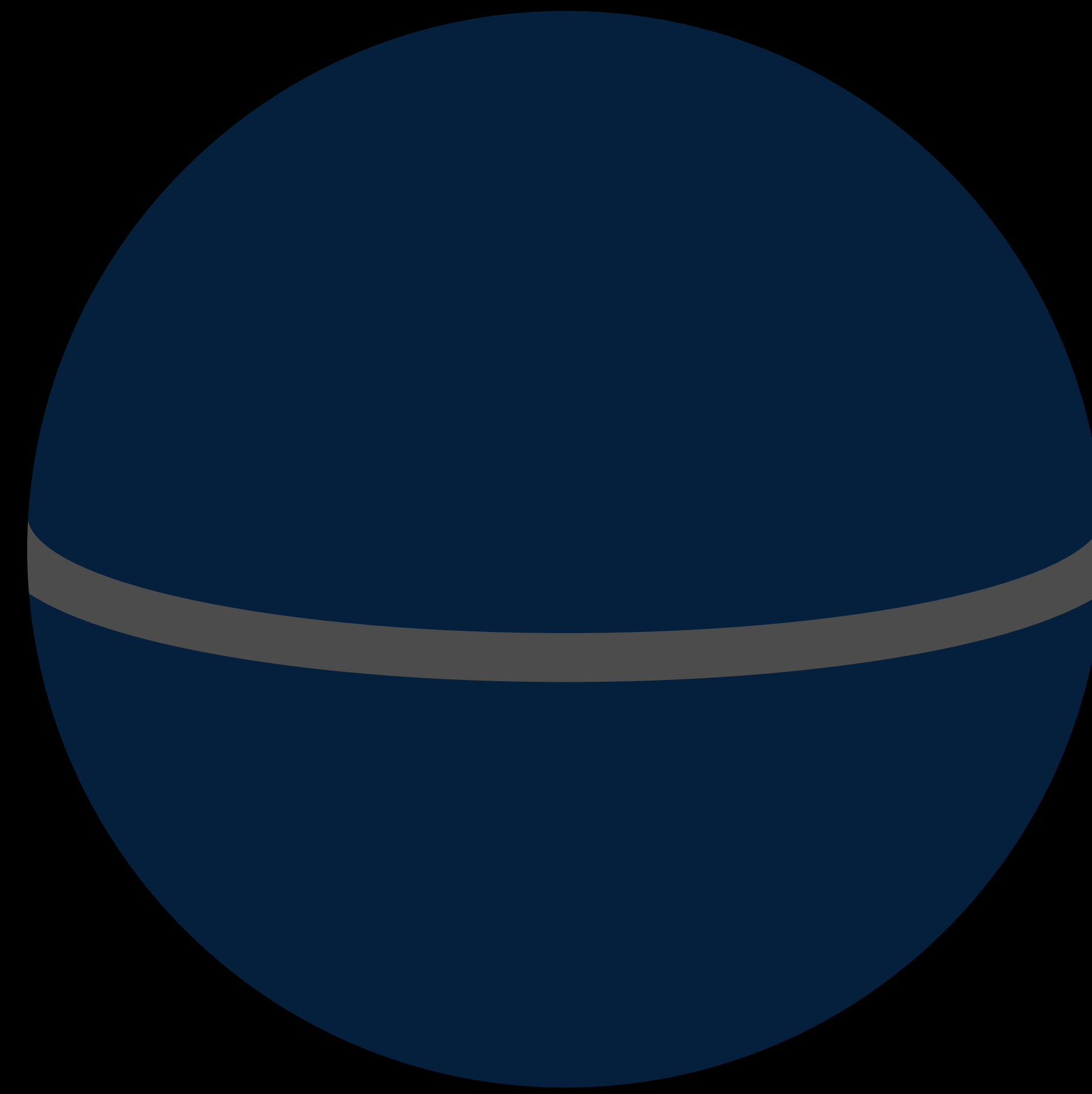
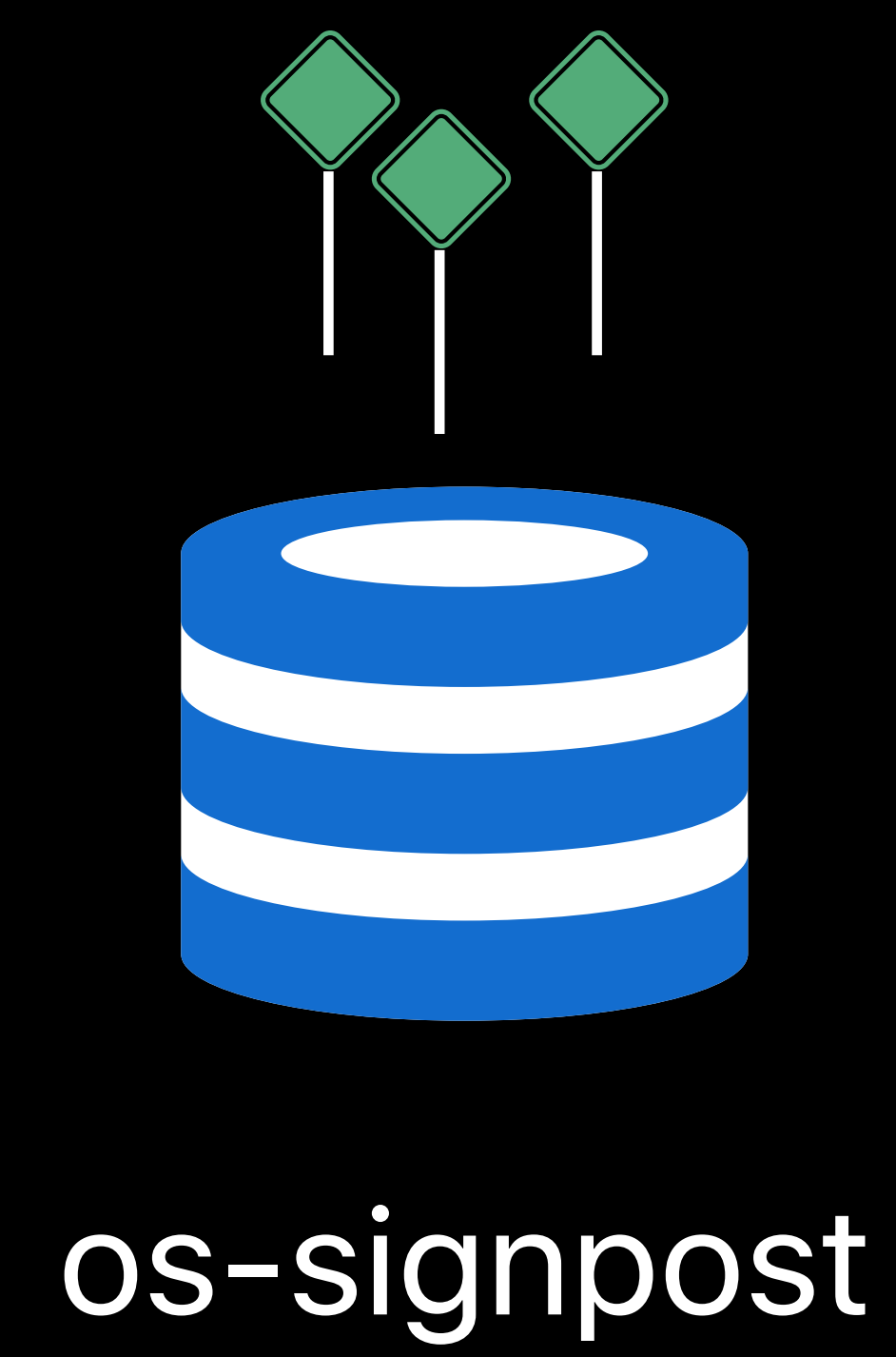


# Example — MobileAgent Activity



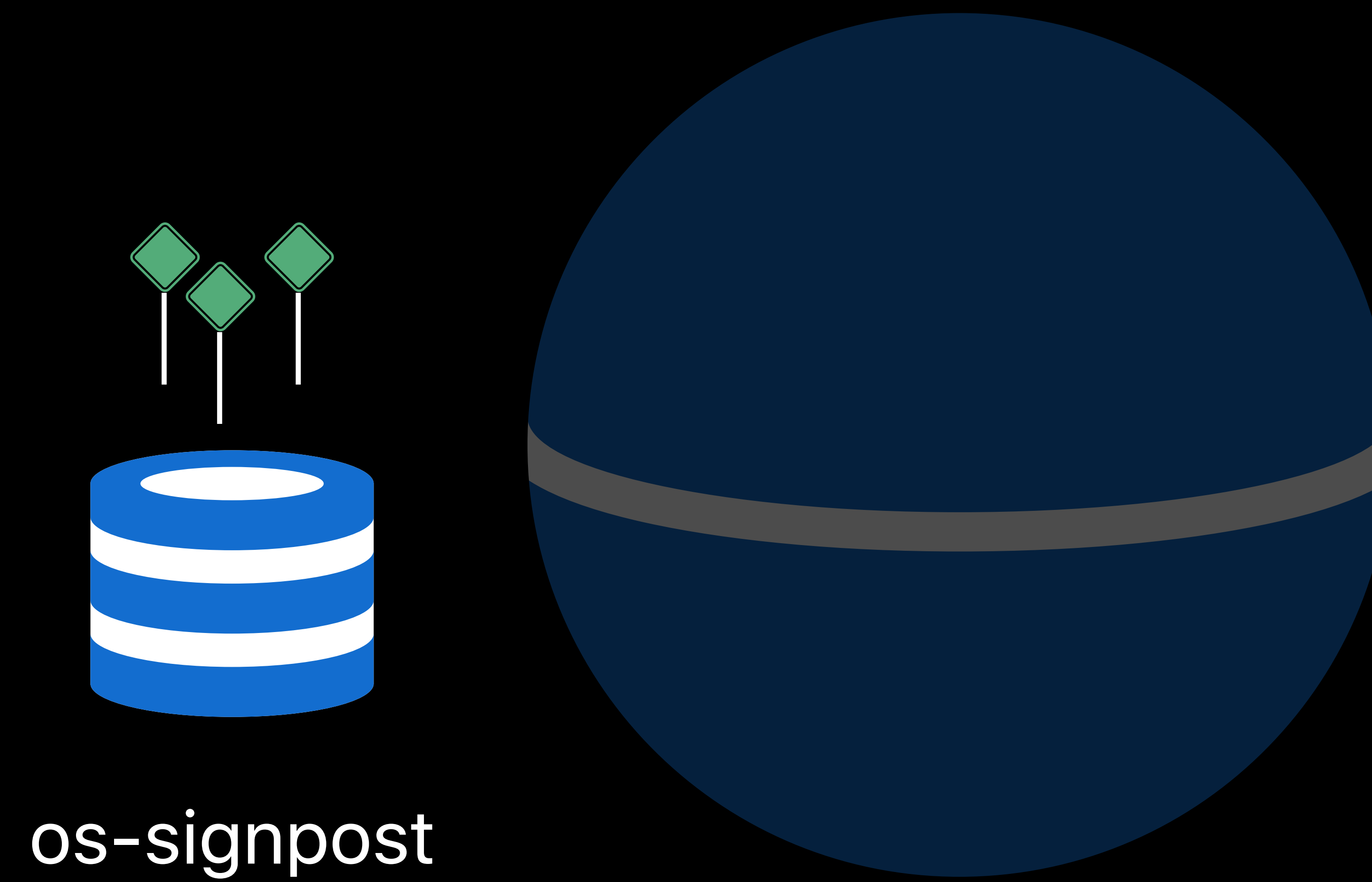
Working Memory

# Example — MobileAgent Activity



# Example — MobileAgent Activity

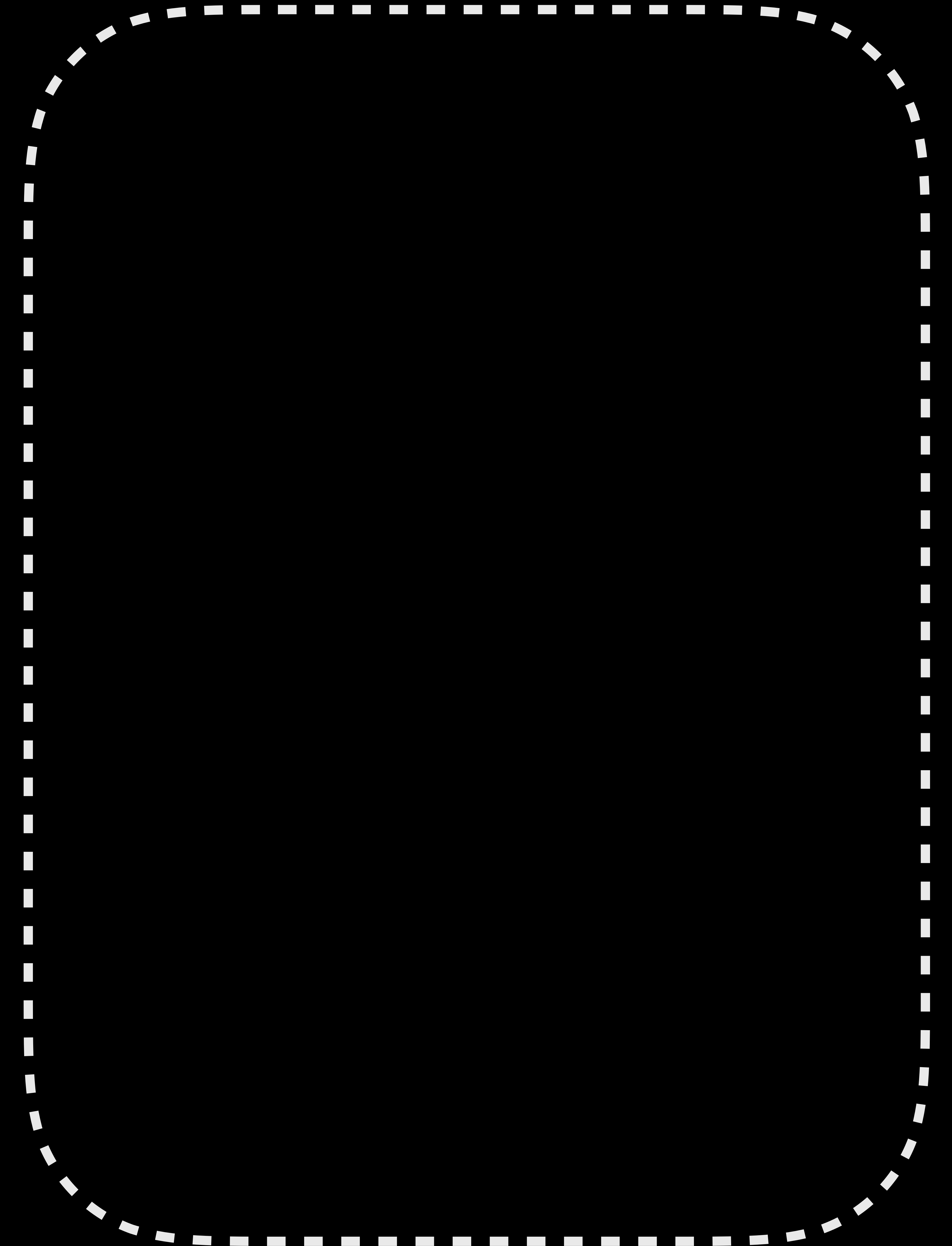
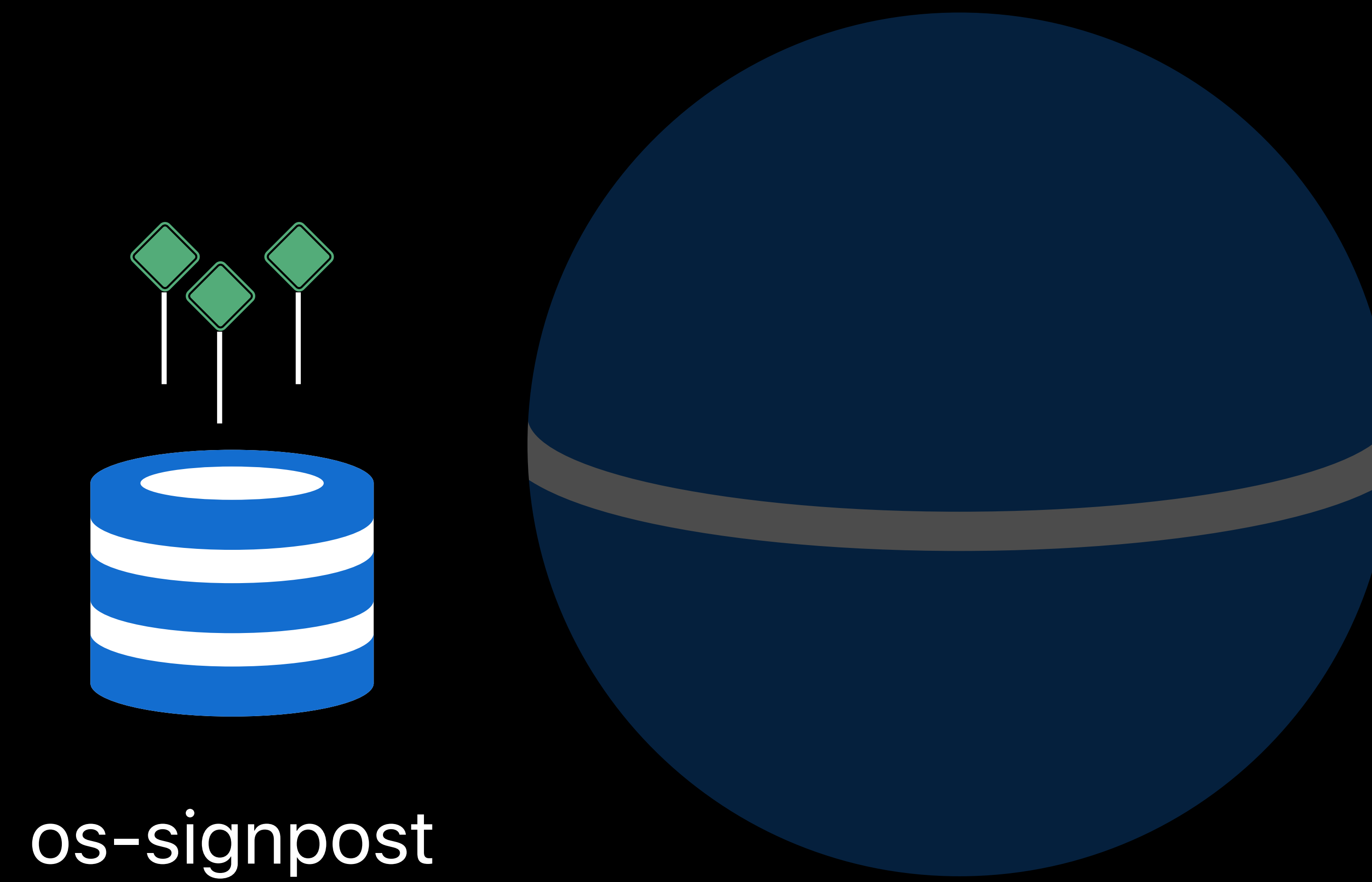
Detect MobileAgent



Working Memory

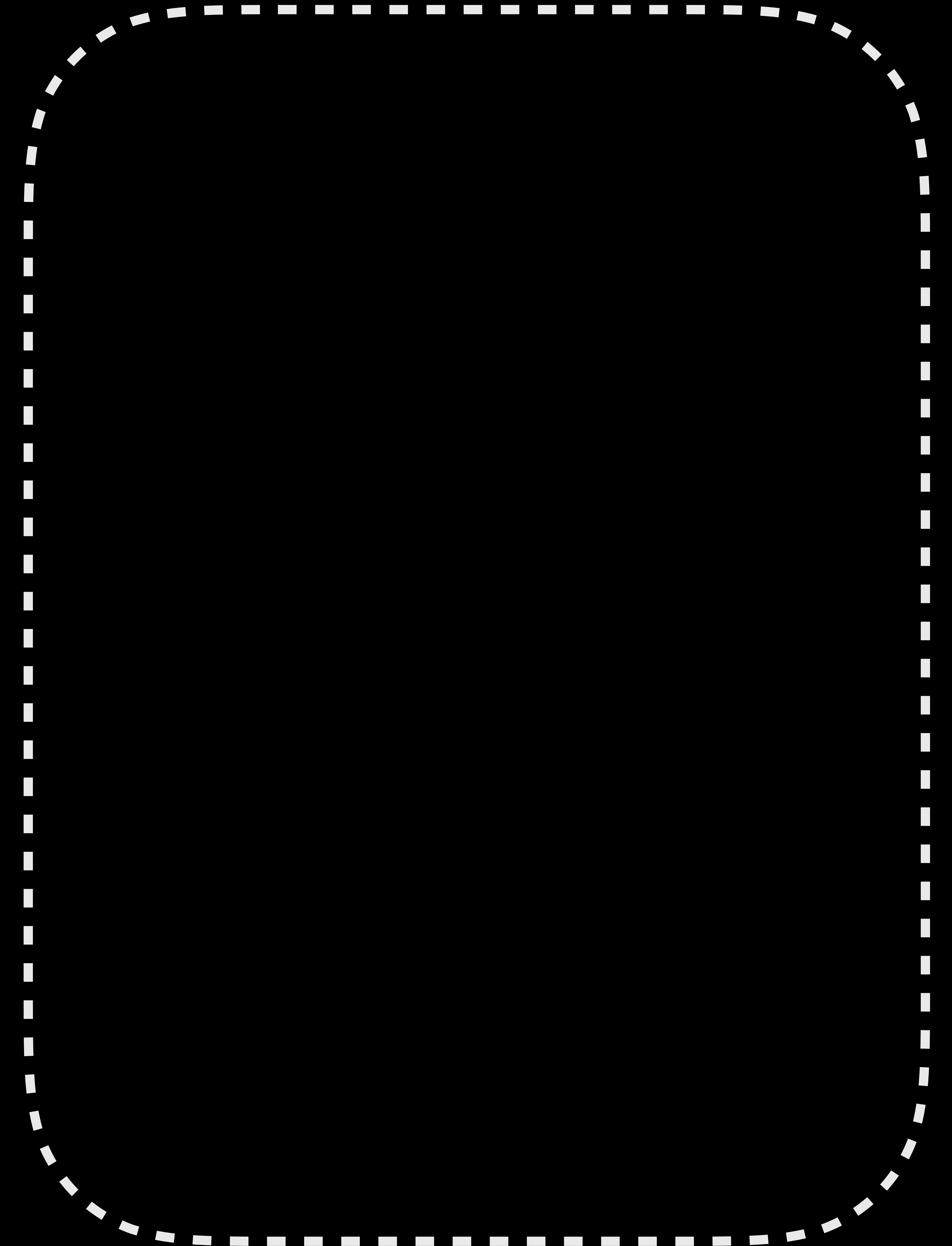
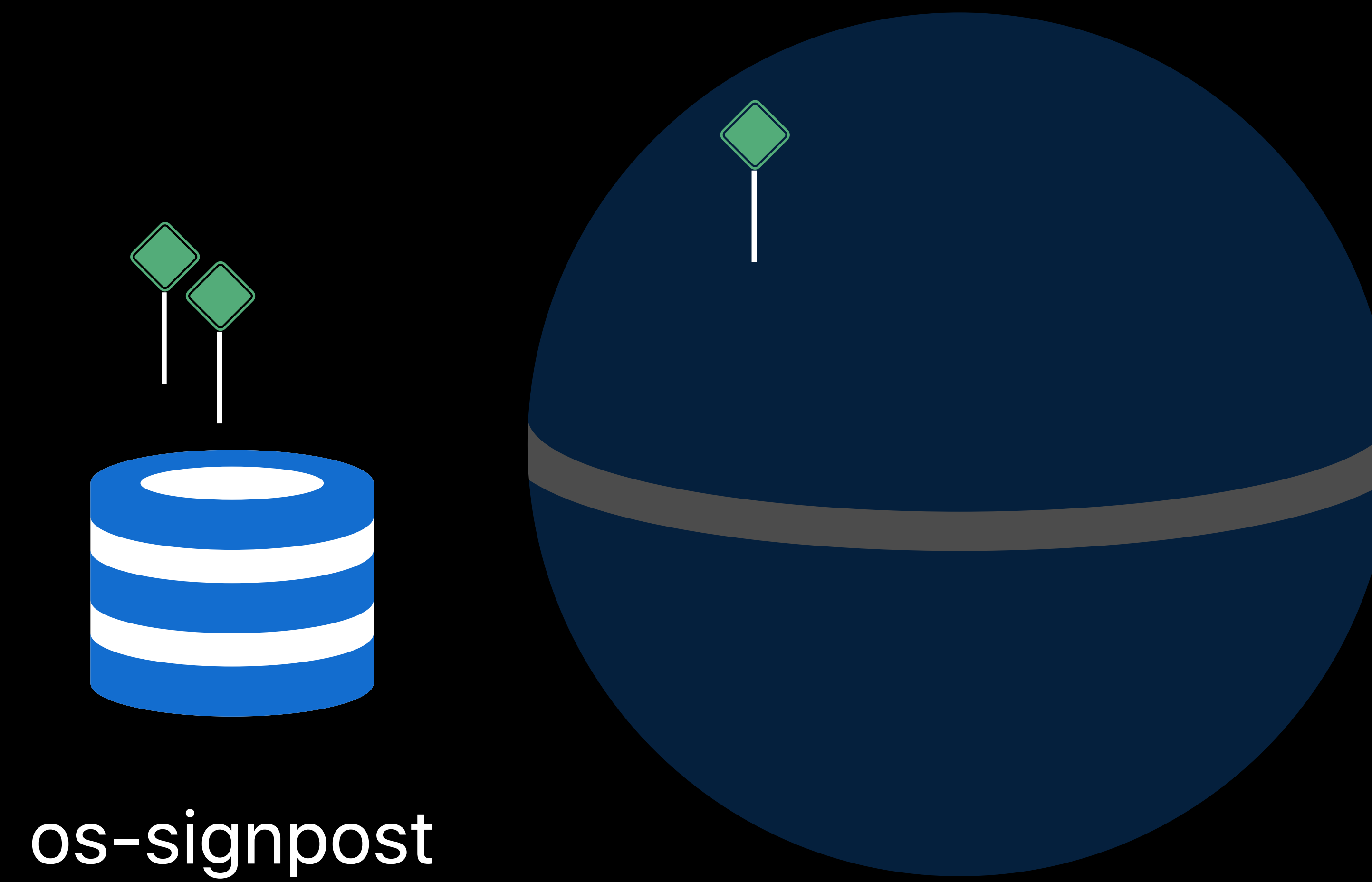
# Example — MobileAgent Activity

Detect MobileAgent



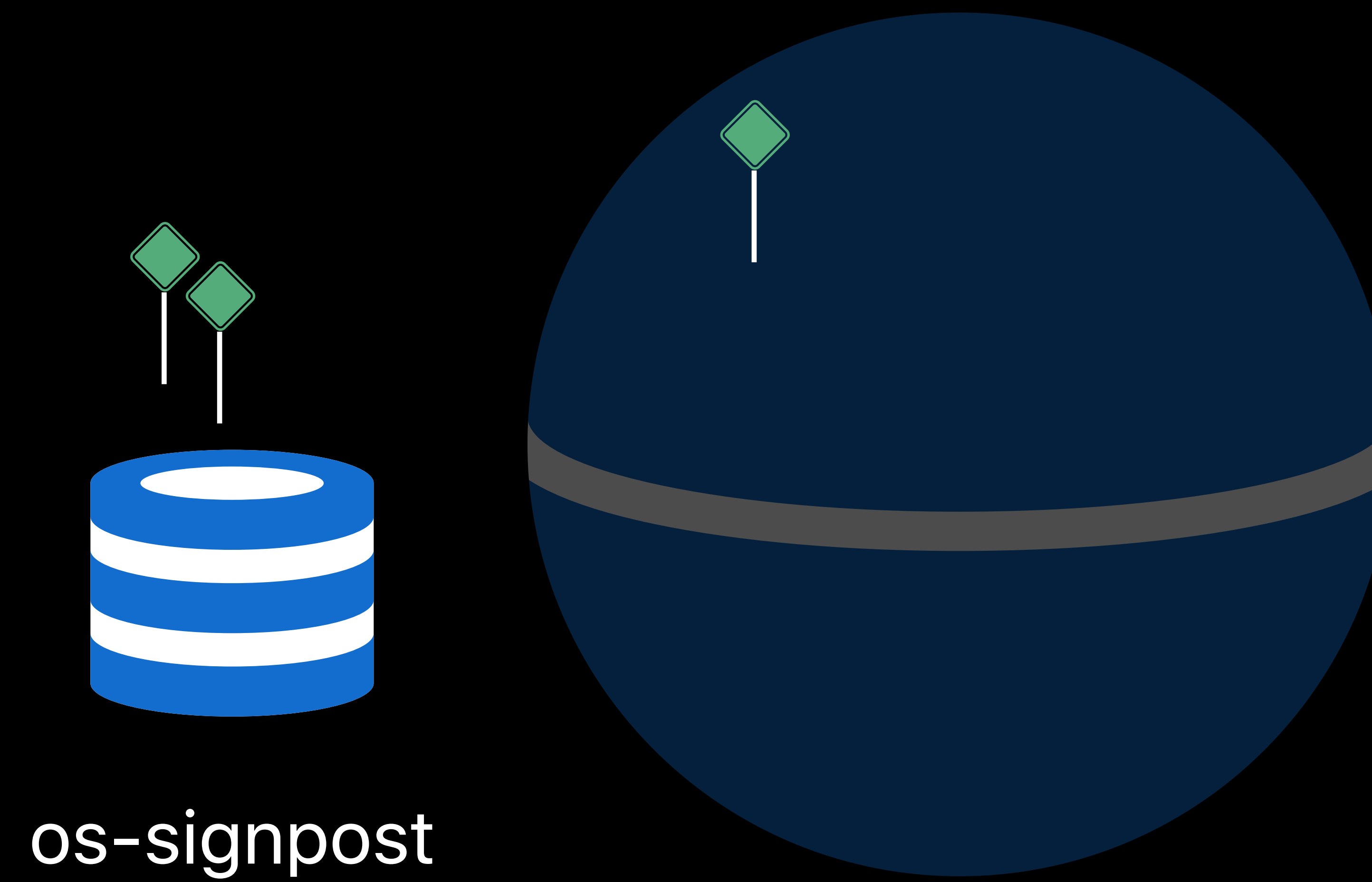
# Example — MobileAgent Activity

Detect MobileAgent



# Example — MobileAgent Activity

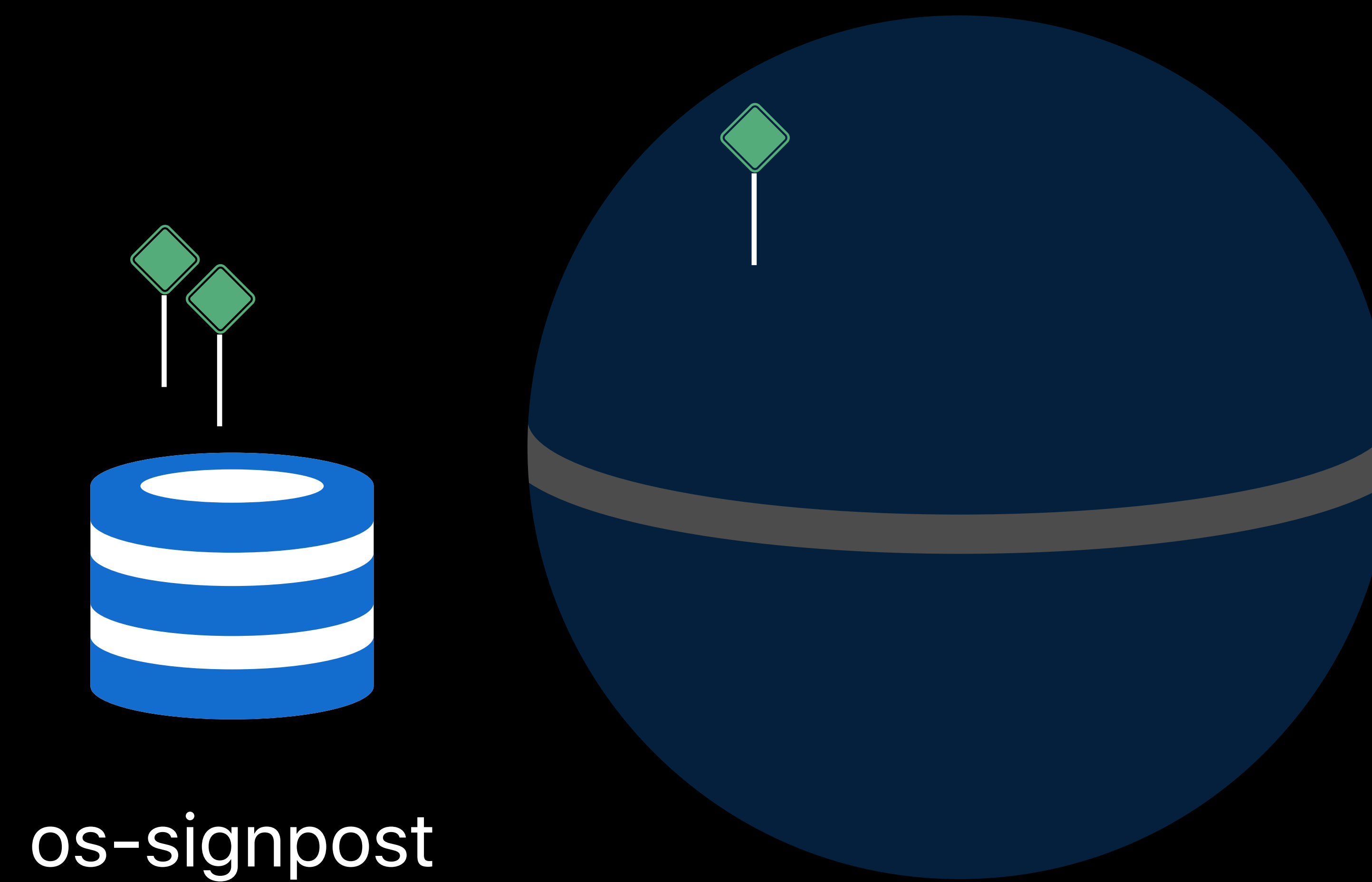
Detect MobileAgent



os-signpost	
time	0:00:42.000
instance	0x234543534
name	"Mobile Agent Moved"
message	"Sort Agent to Background"

# Example — MobileAgent Activity

Detect MobileAgent



os-signpost	
time	0:00:42.000
instance	0x234543534
name	"Mobile Agent Moved"
message	"Sort Agent to Background"

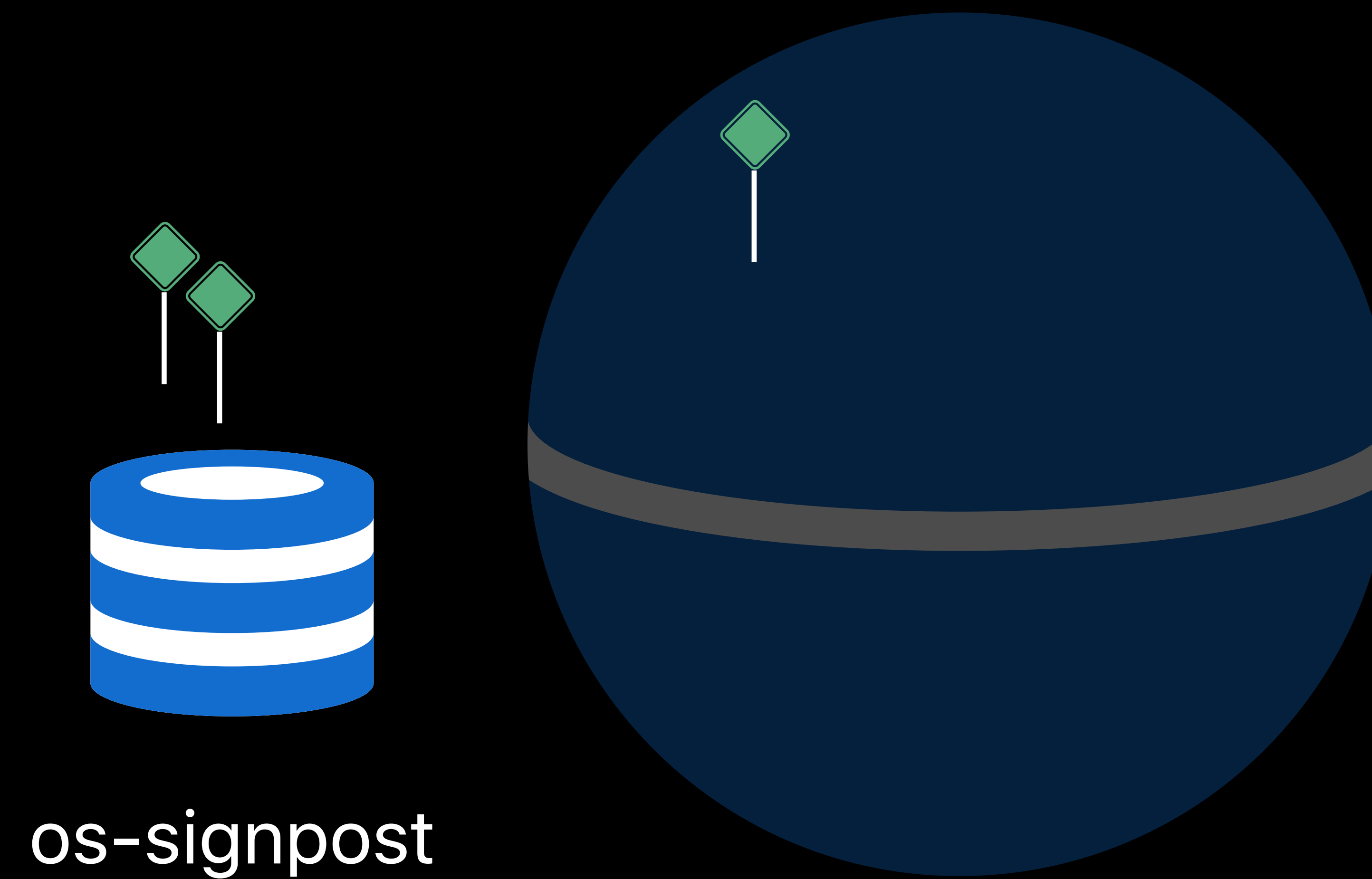
mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"



# Example — MobileAgent Activity

Detect MobileAgent

Determine activity



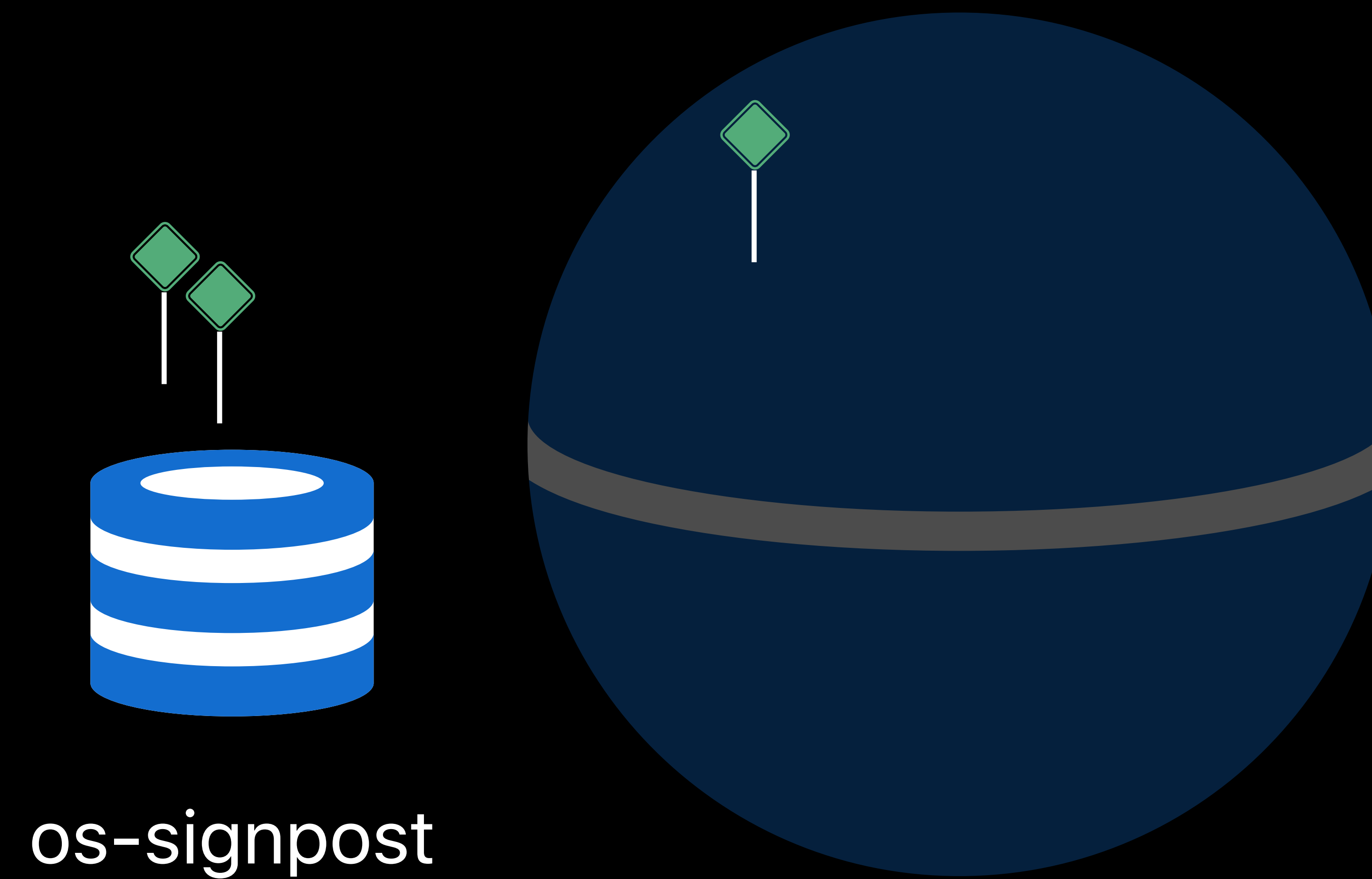
os-signpost	
time	0:00:42.000
instance	0x234543534
name	"Mobile Agent Moved"
message	"Sort Agent to Background"

mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity



os-signpost	
time	0:00:42.000
instance	0x234543534
name	"Mobile Agent Moved"
message	"Sort Agent to Background"

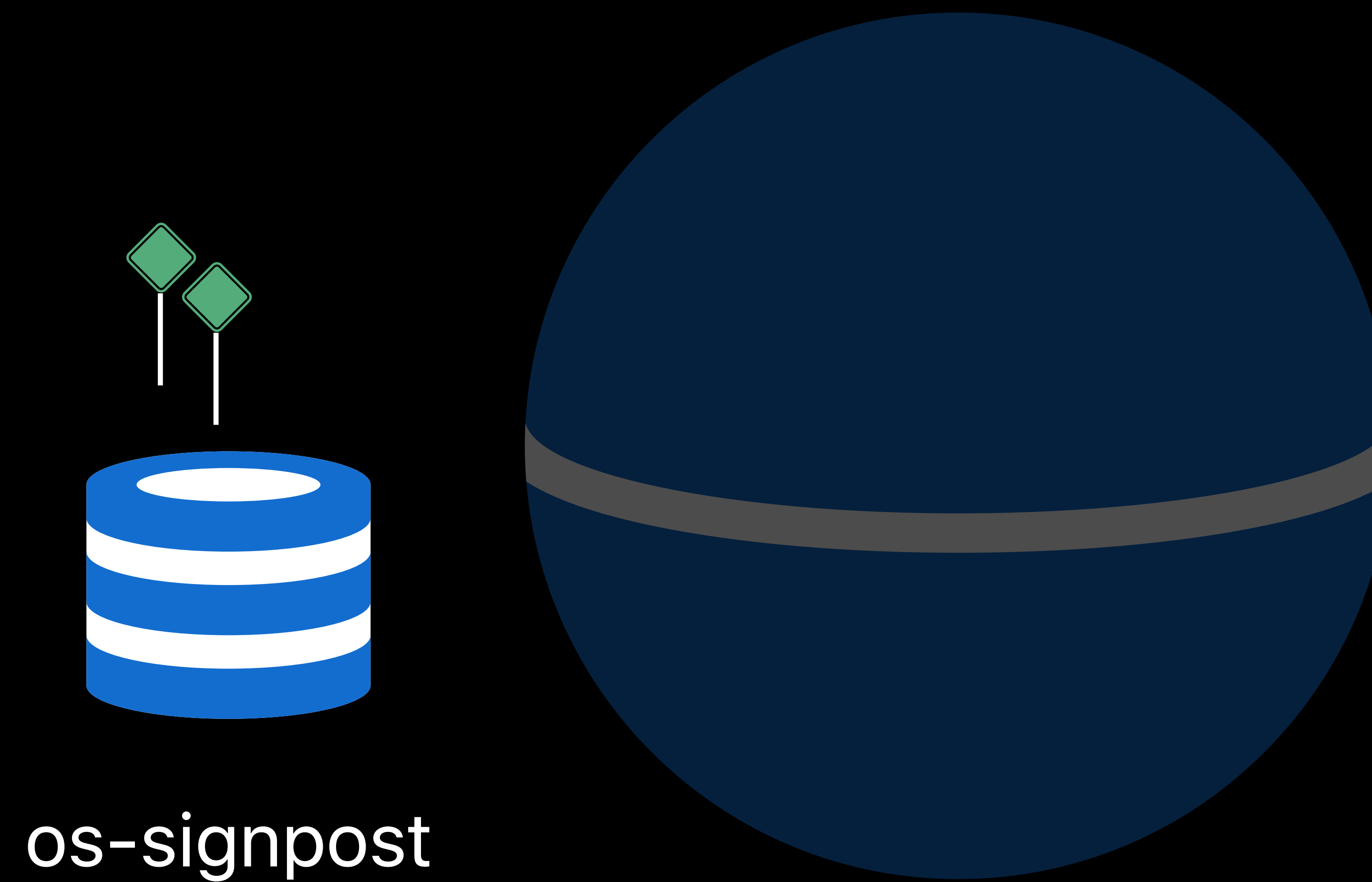
mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity



os-signpost	
time	0:00:42.000
instance	0x234543534
name	"Mobile Agent Moved"
message	"Sort Agent to Background"

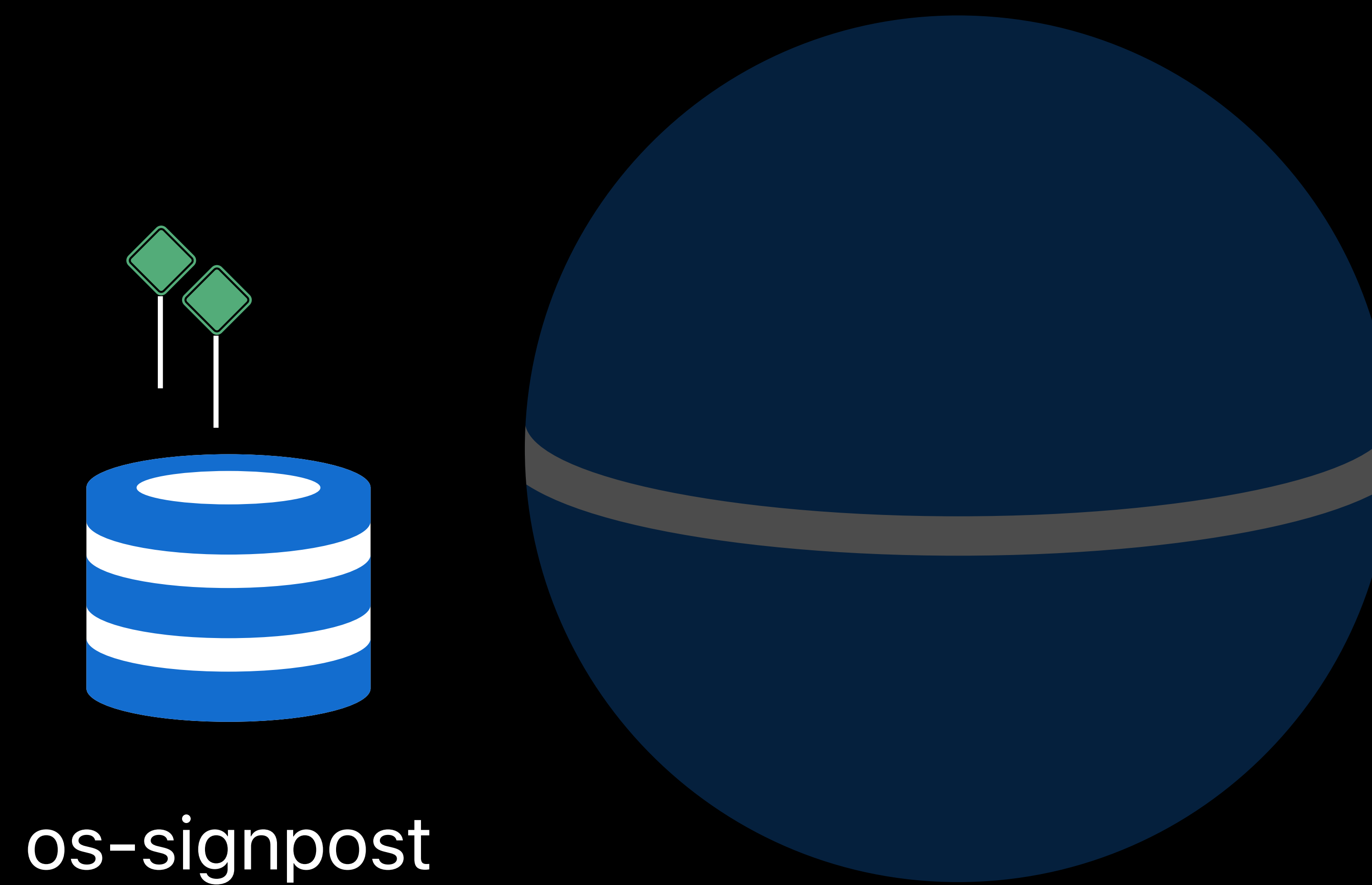
mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity



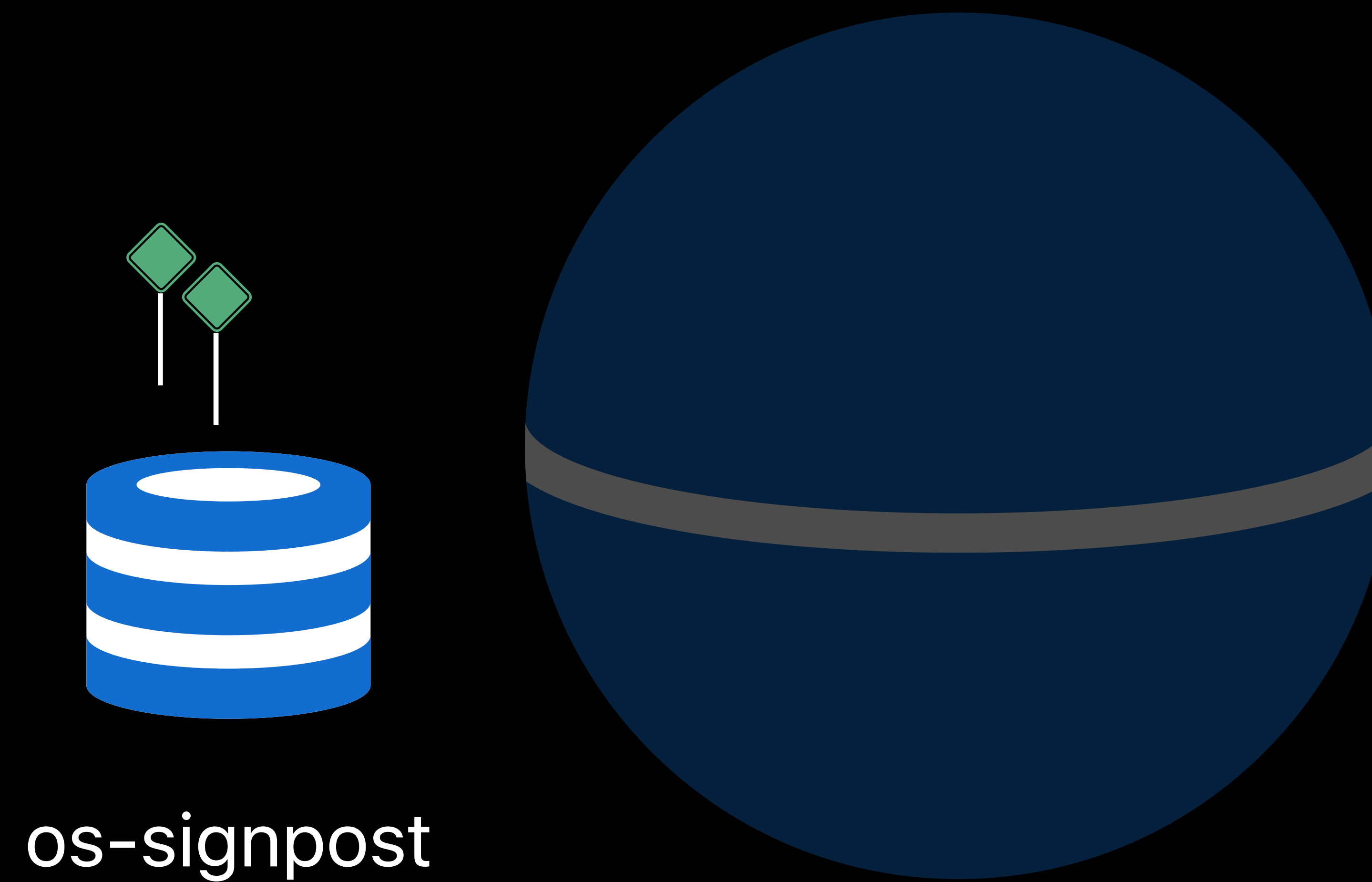
mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

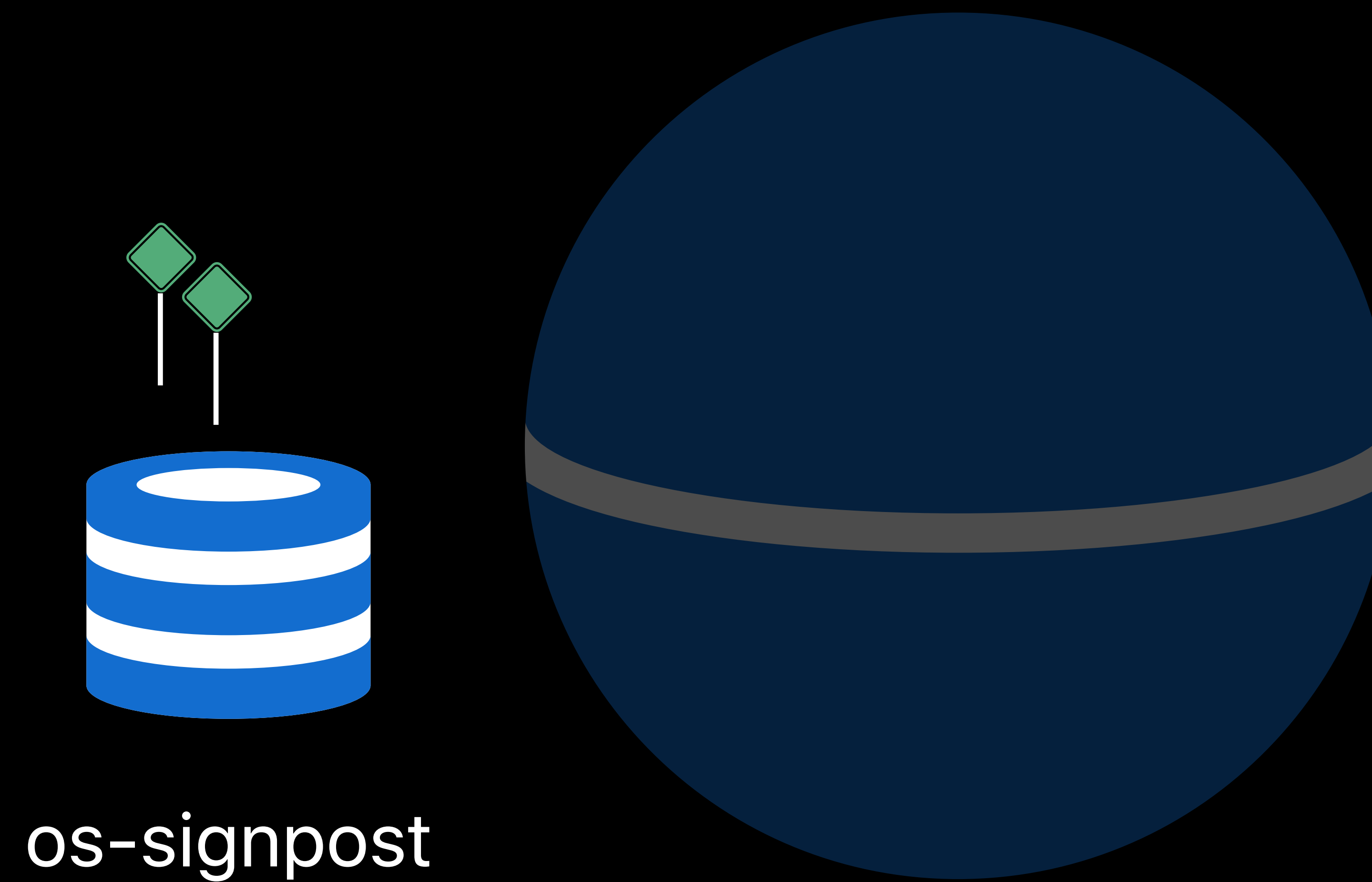
mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

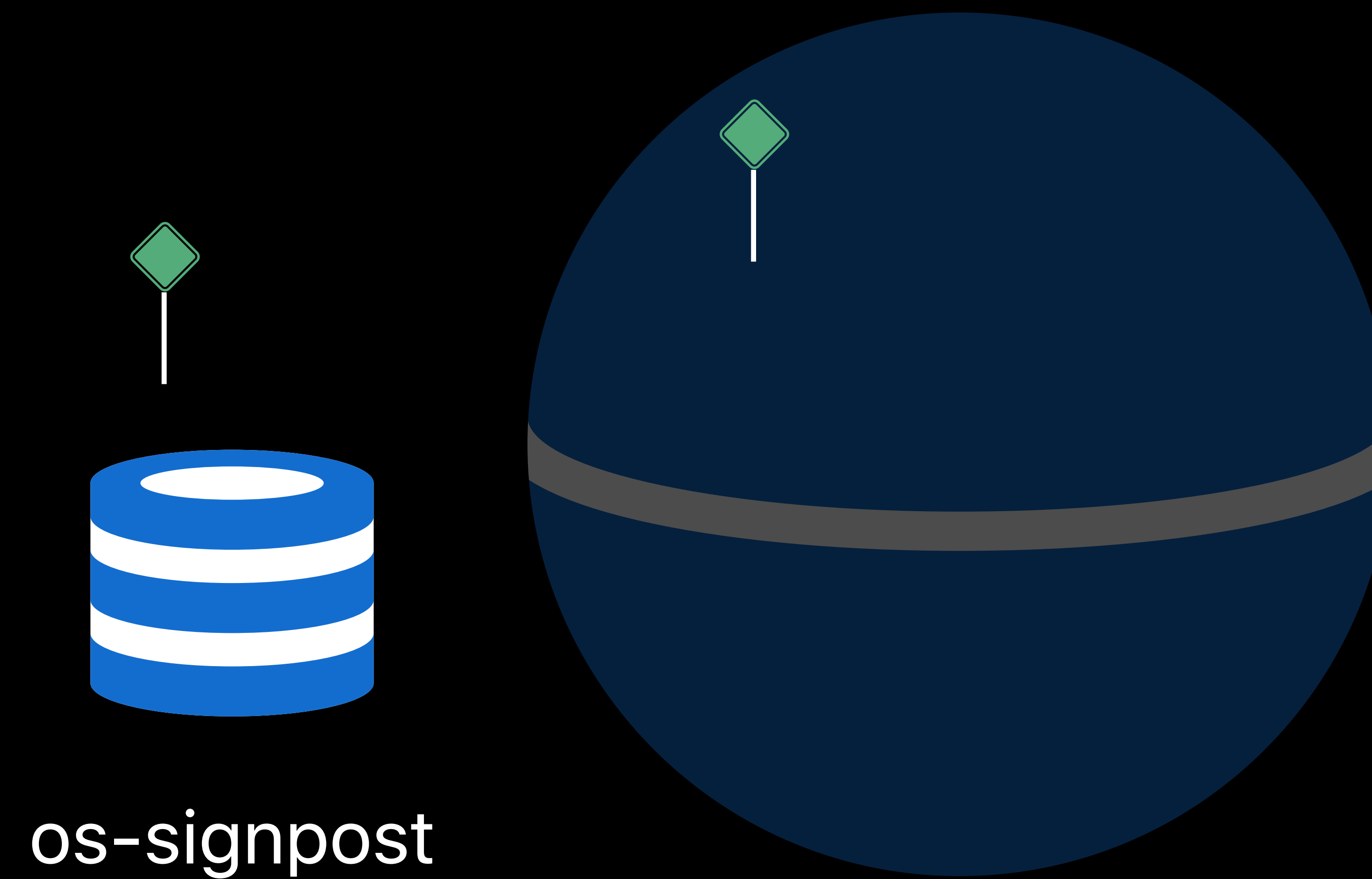
mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

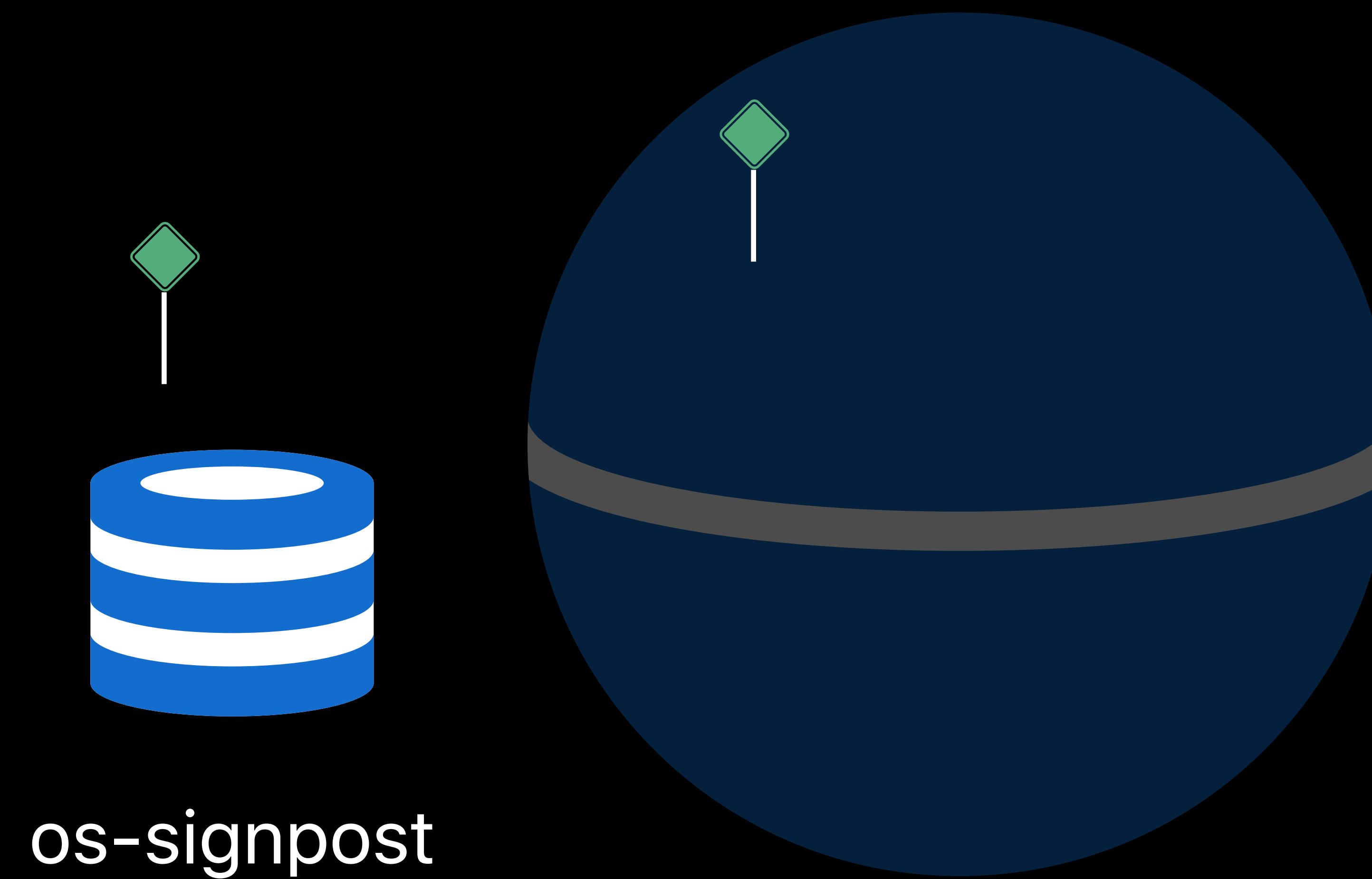
mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

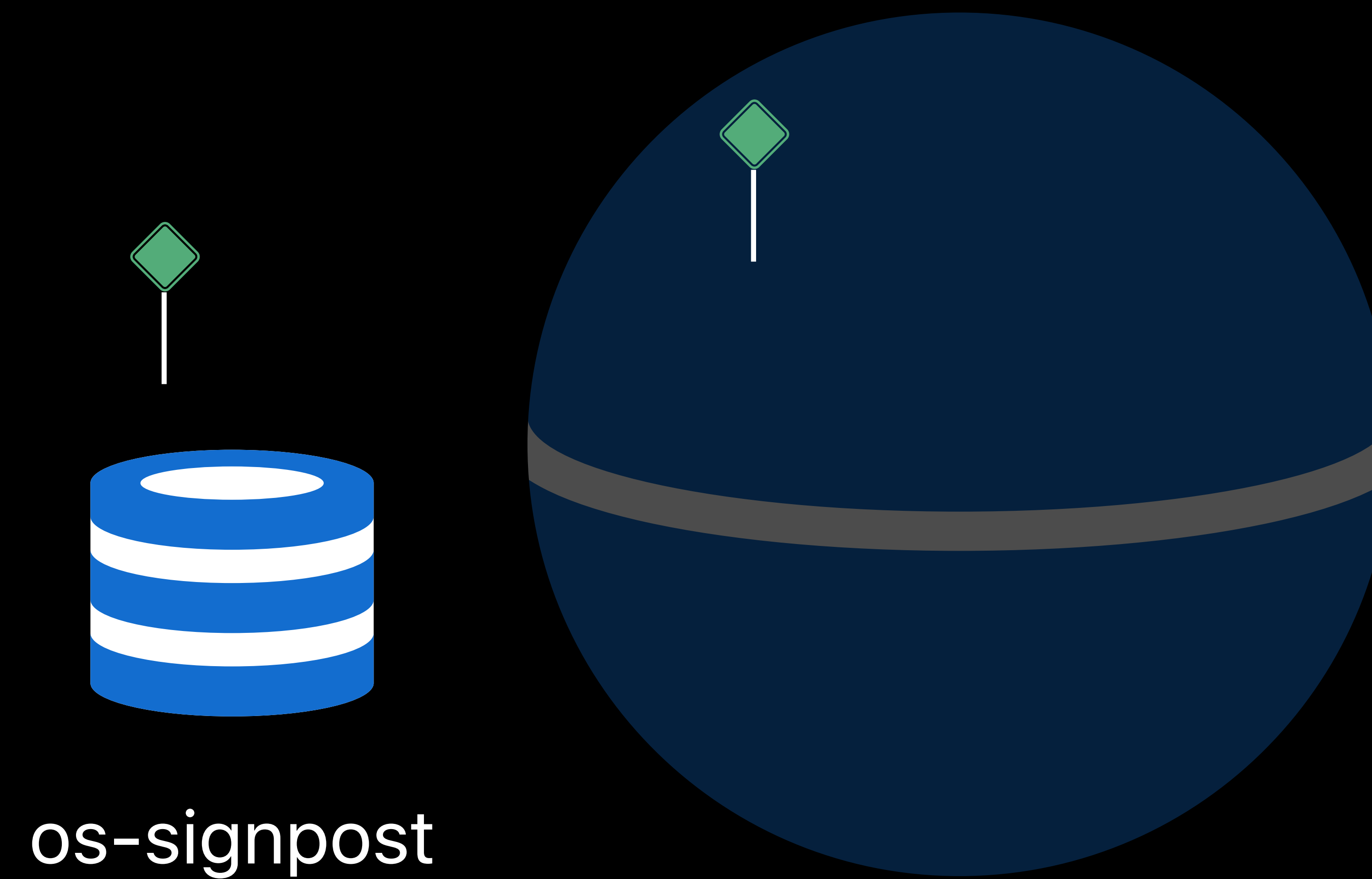


# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

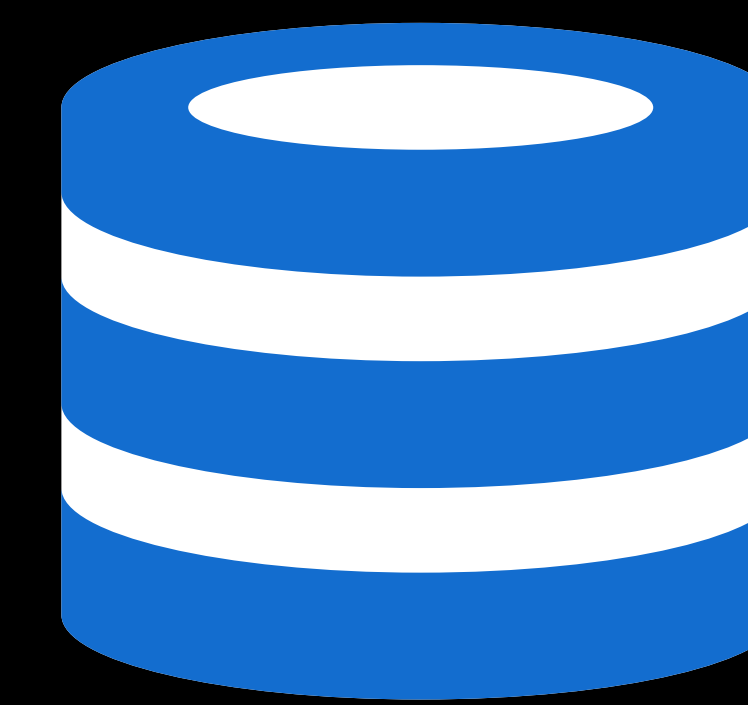
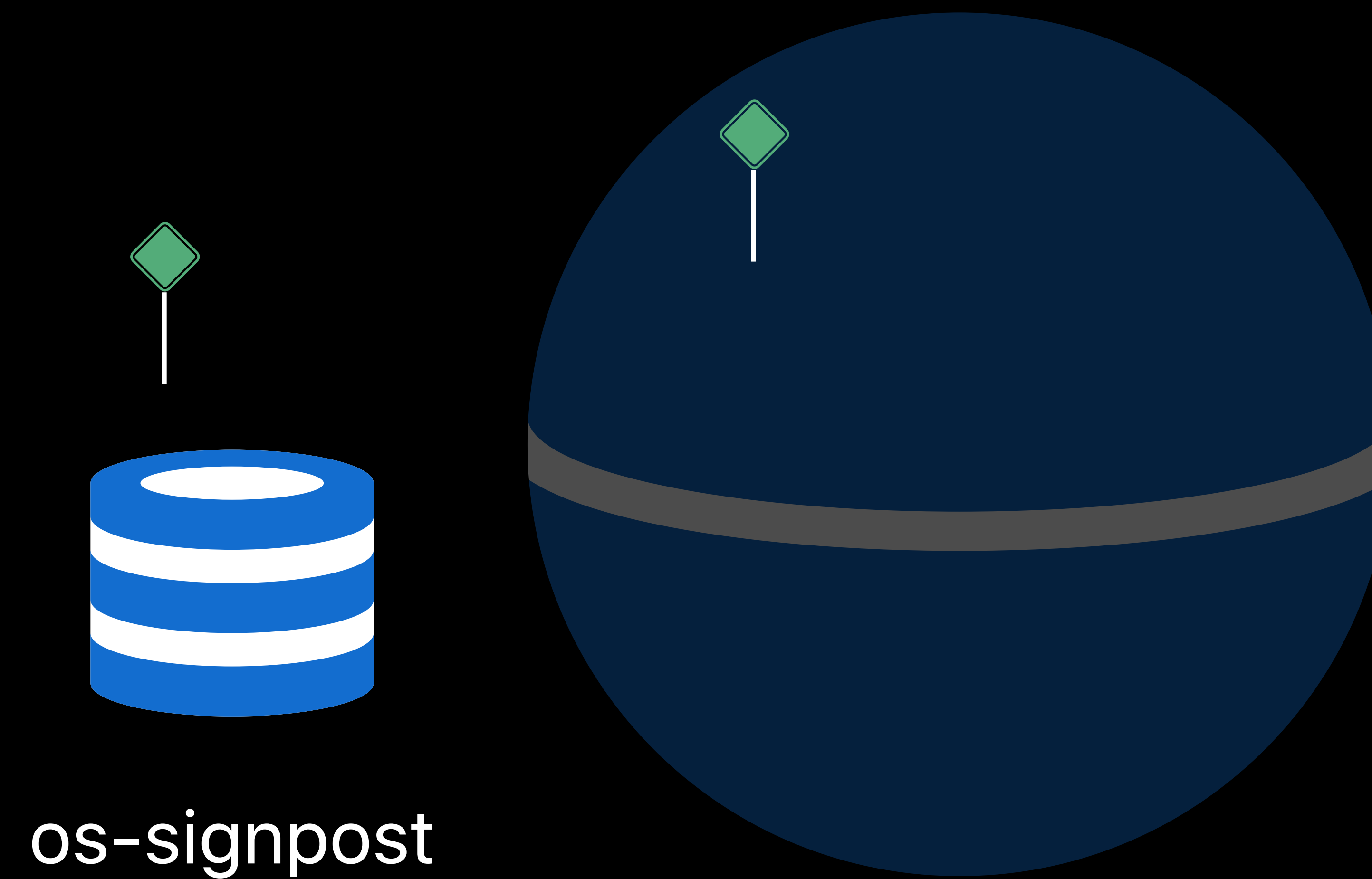
os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



Output Table

mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

mobile-agent-movement	
instance	0x234543534
start	0:00:42.000

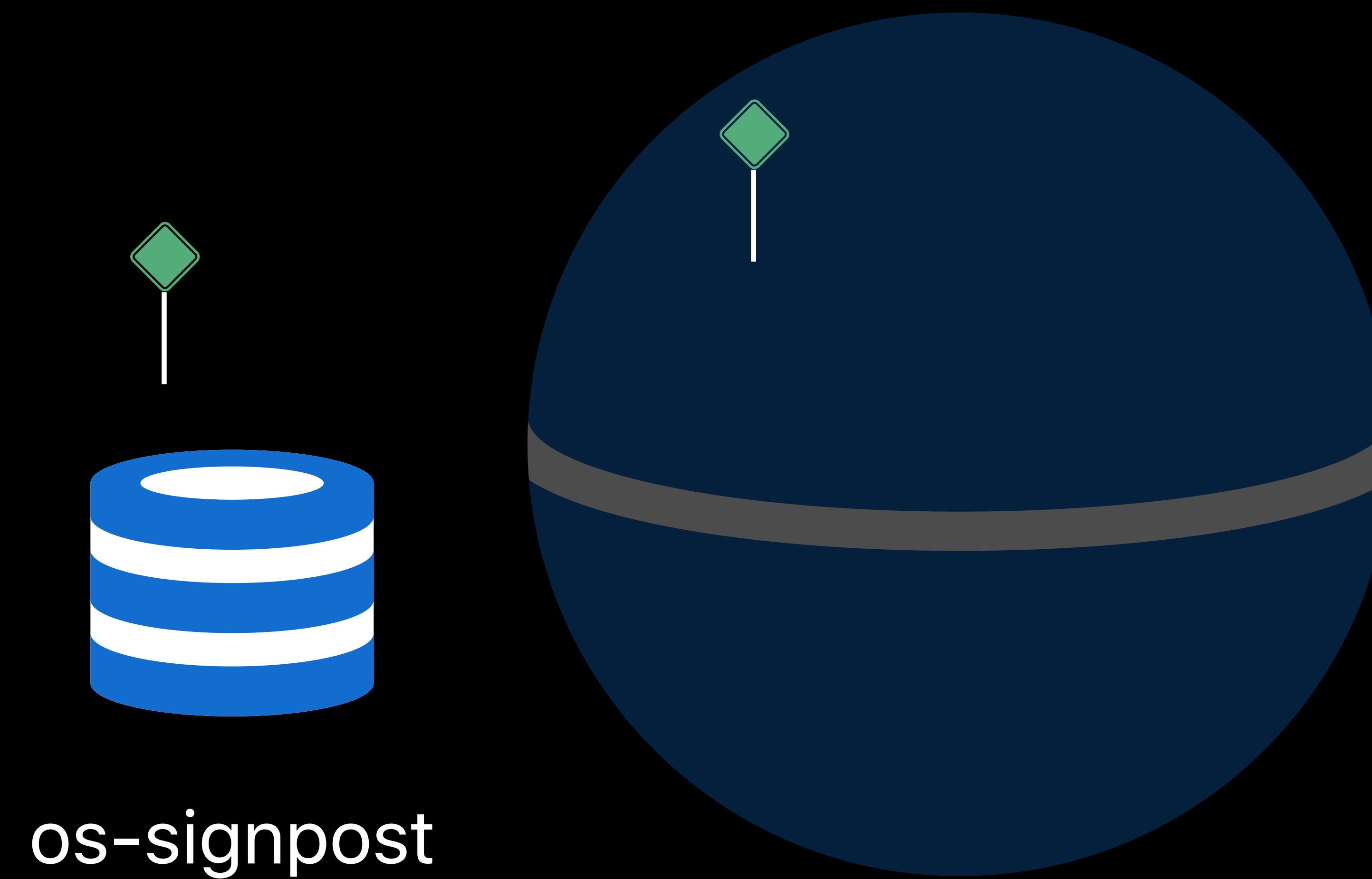
os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving to Background"

os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

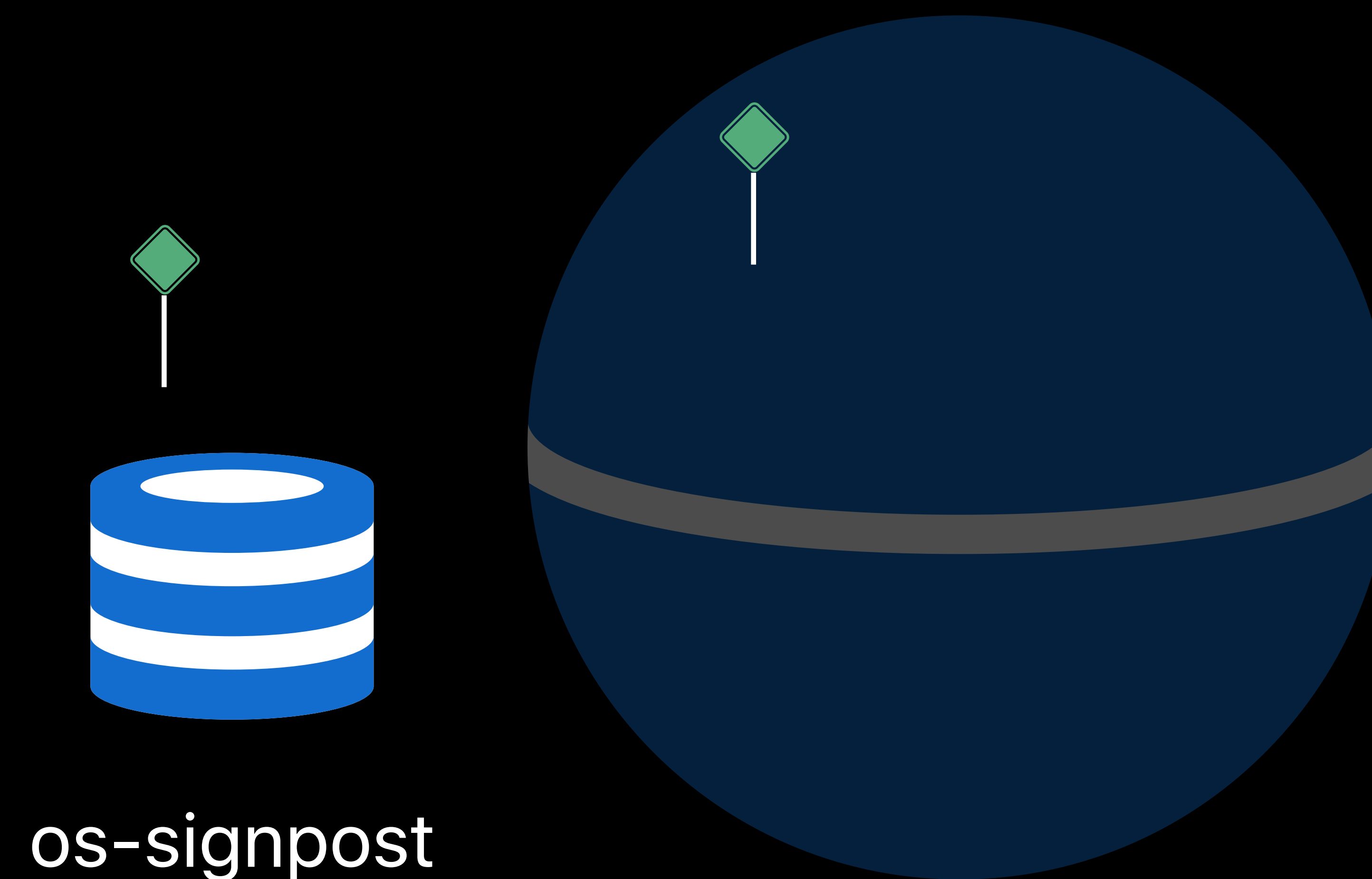
Output Table

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving"

os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

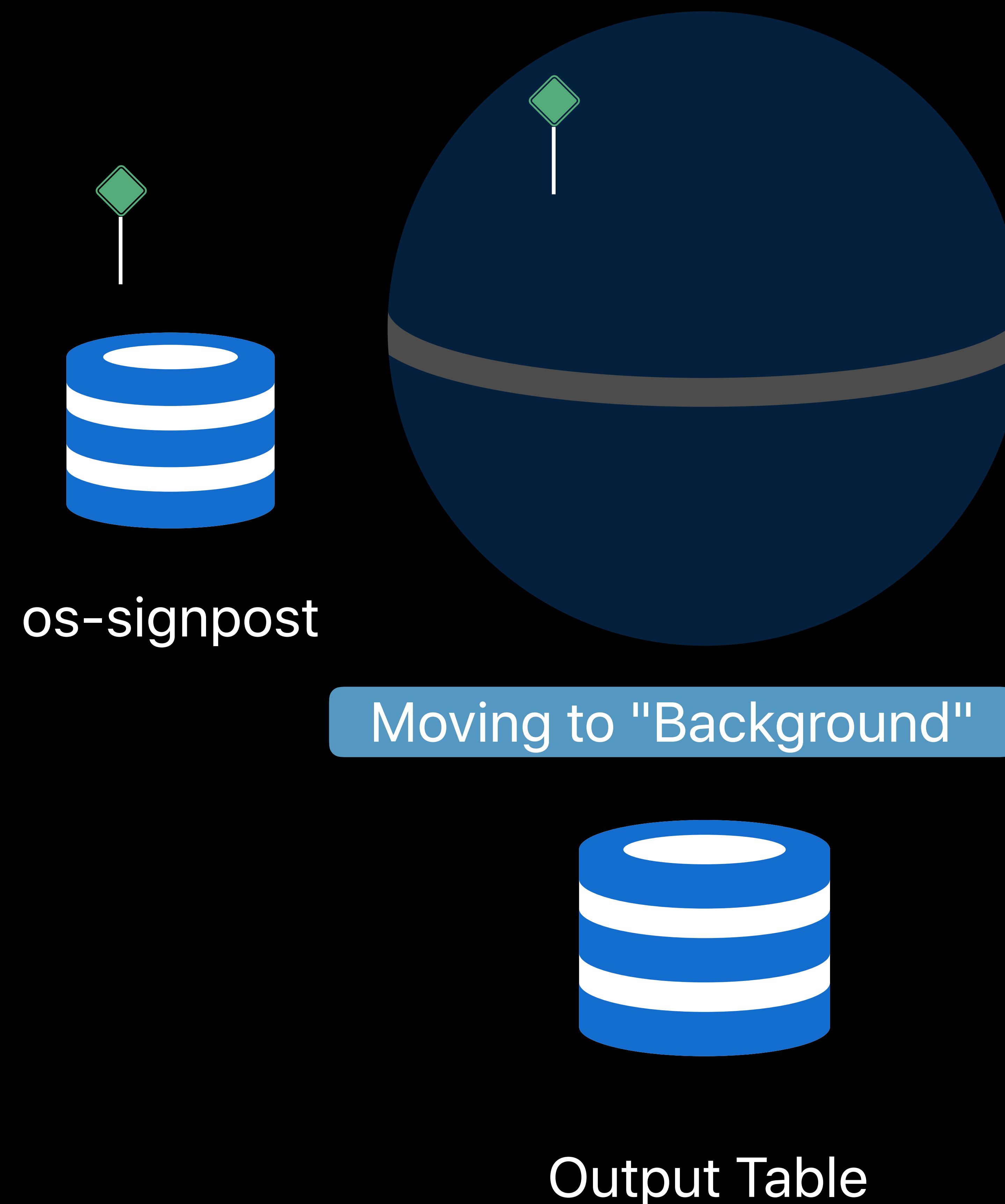
Output Table

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving"

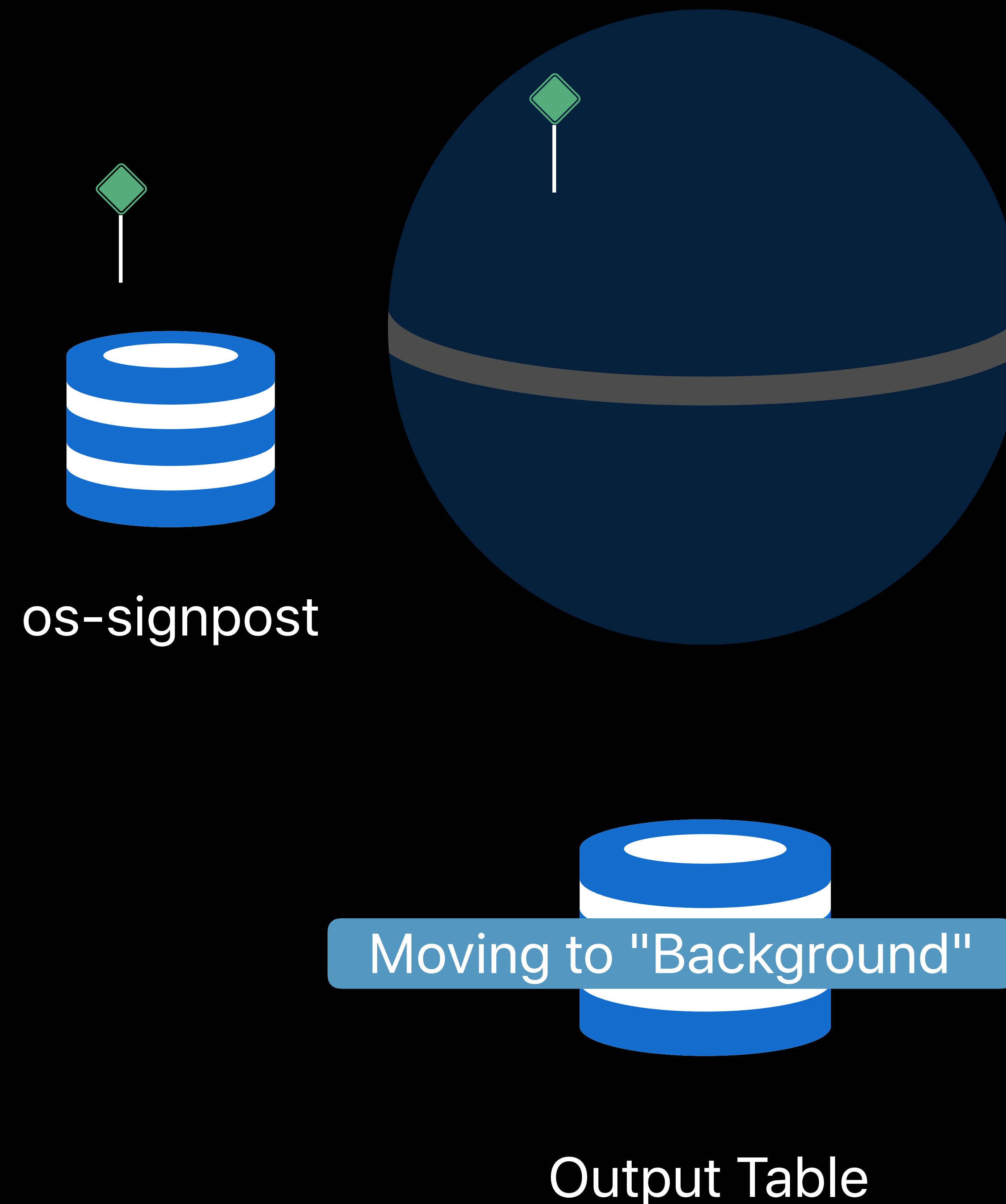
os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving"

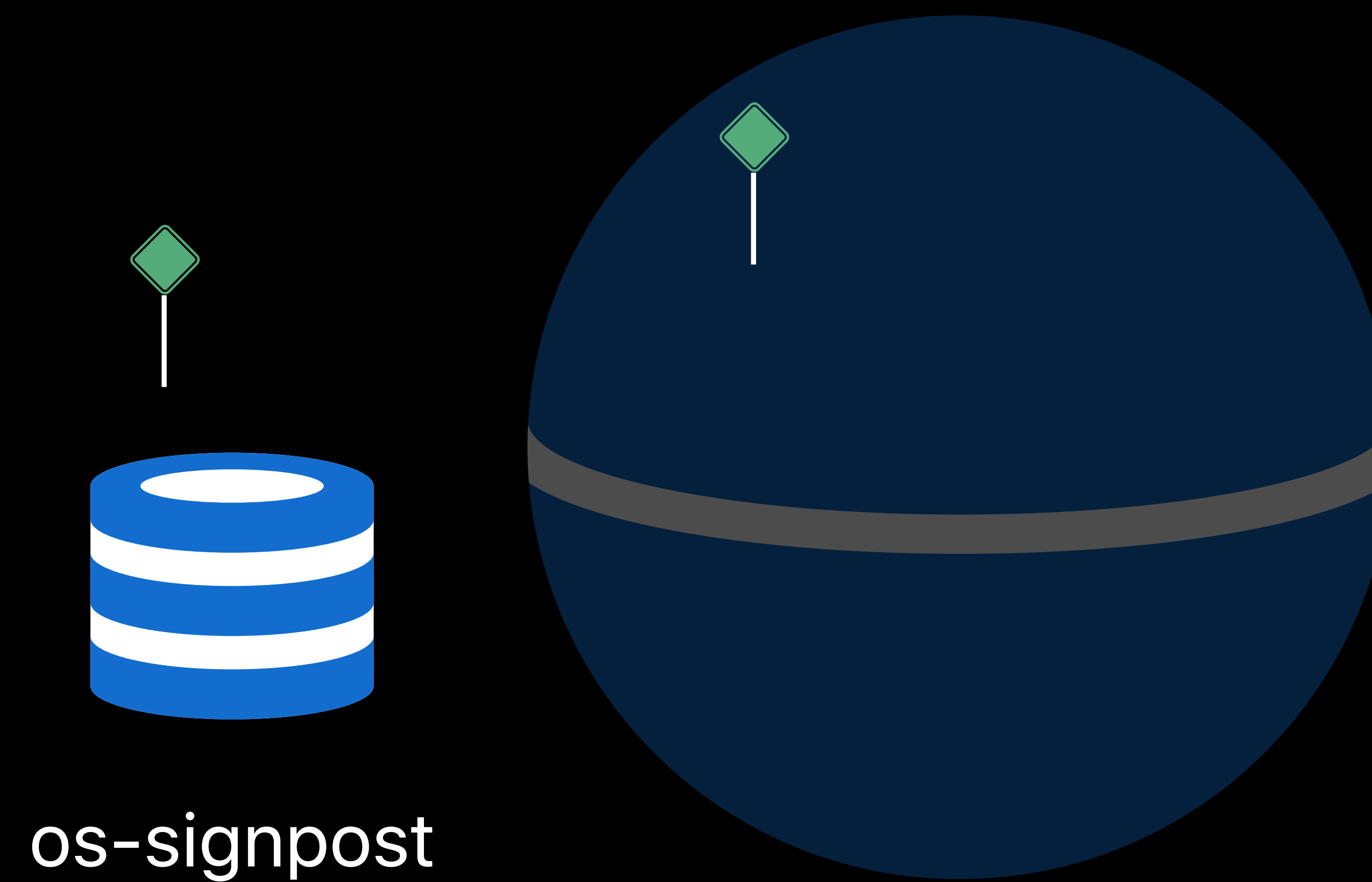
os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving"

os-signpost	
time	0:00:54.000
instance	0x234543534
name	"Mobile Agent Exec"
message	"Sort Agent Sorting"

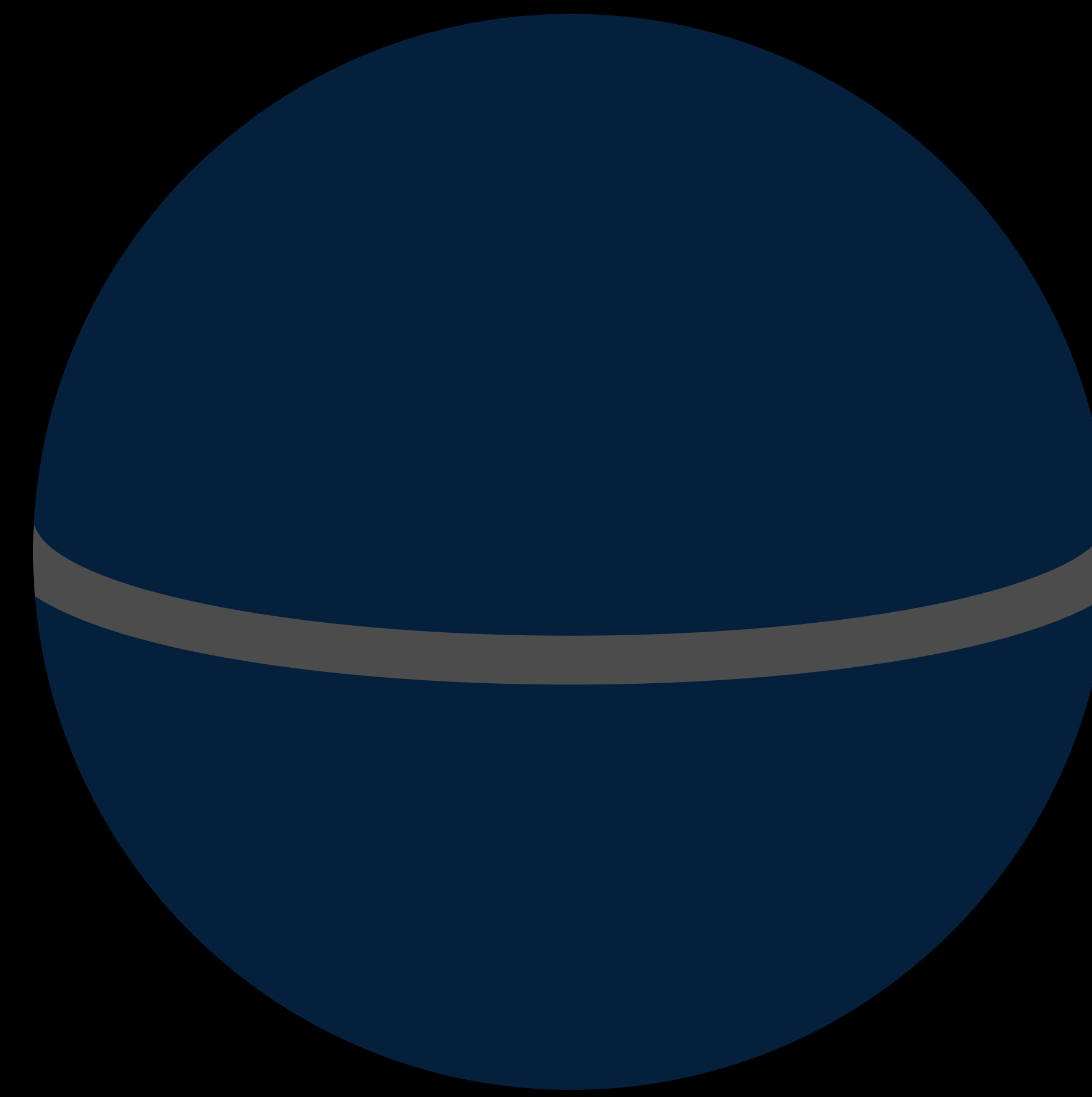
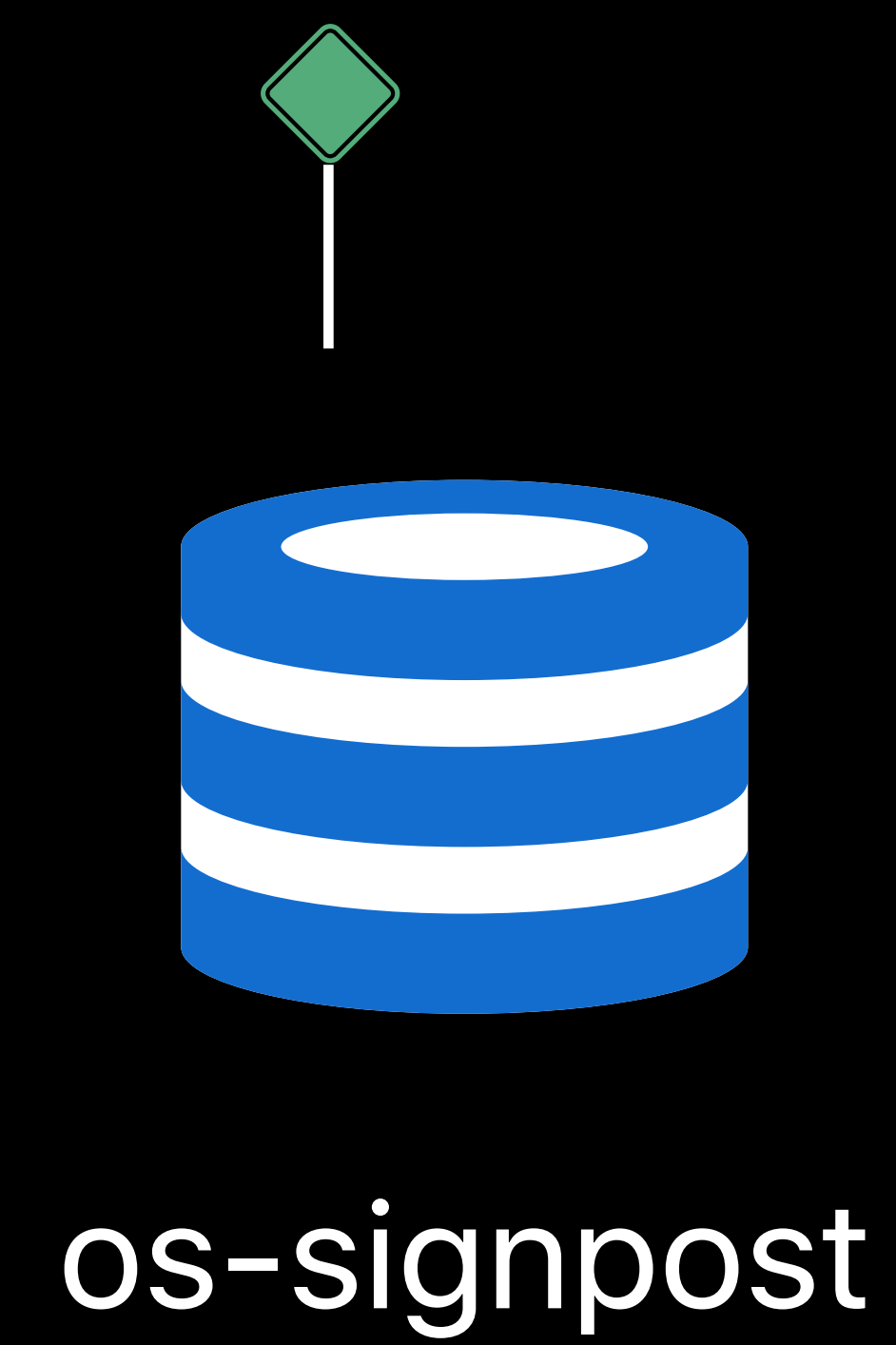
Output Table

# Example — MobileAgent Activity

Detect MobileAgent

Determine activity

Determine interval



Output Table

mobile-agent	
instance	0x234543534
agent-kind	"Sort Agent"
mode	"Moving"



```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```



```
// os-signpost
```

```
(os-signpost
```

```
  (time ?t)
```

```
  (name "Mobile Agent Moved")
```

```
  (event-type "Event")
```

```
  (identifier ?instance)
```

```
  (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
os_signpost(.event,
```

```
  log: MobileAgent.signpostHandleID,
```

```
  name: "Mobile Agent Moved",
```

```
  signpostID: signpostID,
```

```
  "Agent of type %@ received by %@ for mode %@",
```

```
  agent.agentType(),
```

```
  destination.agentStopType(),
```

```
  agent.mode)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
(os-signpost
```

```
(time ?t)
```

```
(name "Mobile Agent Moved")
```

```
(event-type "Event")
```

```
(identifier ?instance)
```

```
(message$ "Agent of type " ?agent-kind " received by " $?)
```

```
)
```

```
(not (mobile-agent (instance ?instance)))
```

```
=>
```

```
(assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```



```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
(not (mobile-agent (instance ?instance)))
```

```
=>
```

```
(assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```



```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
(assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Detection
```

```
(defrule MODELER::detect-new-mobile-agent
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (not (mobile-agent (instance ?instance)))
```

```
=>
```

```
  (assert (mobile-agent (start ?t) (instance ?instance) (agent-kind ?agent-kind)))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
(mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```



```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

```
//MobileAgent Move
```

```
(defrule MODELER::detect-mobile-agent-transition-begin
```

```
  (os-signpost
```

```
    (time ?t)
```

```
    (name "Mobile Agent Moved")
```

```
    (event-type "Event")
```

```
    (identifier ?instance)
```

```
    (message$ "Agent of type " ?agent-kind " received by " $?)
```

```
  )
```

```
  (mobile-agent (instance ?instance))
```

```
=>
```

```
(assert (mobile-agent-transition-started (start ?t) (instance ?instance))
```

```
)
```

# Rule Execution

Chad Woolf, Performance Tools

LHS  $\Rightarrow$  RHS

LHS  $\Rightarrow$  RHS

open-interval

mobile-agent

os-signpost

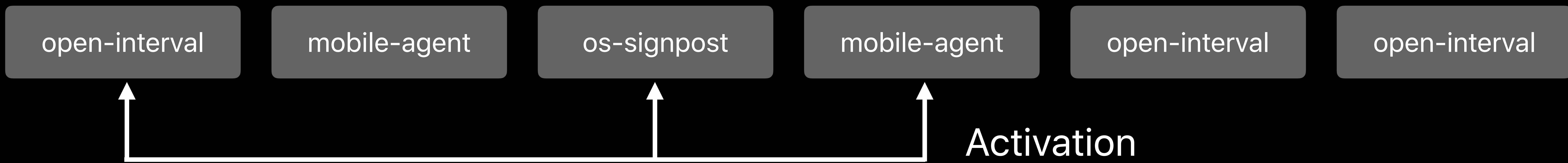
mobile-agent

open-interval

open-interval

LHS  $\Rightarrow$  RHS





LHS => RHS

open-interval

mobile-agent

os-signpost

mobile-agent

open-interval

open-interval

LHS  $\Rightarrow$  RHS

open-interval

mobile-agent

os-signpost

mobile-agent

open-interval

LHS  $\Rightarrow$  RHS

open-interval

mobile-agent

os-signpost

mobile-agent

open-interval

LHS  $\Rightarrow$  RHS

closed-interval

open-interval

mobile-agent

os-signpost

mobile-agent

closed-interval

open-interval

LHS  $\Rightarrow$  RHS

open-interval

mobile-agent

os-signpost

mobile-agent

closed-interval

open-interval

LHS  $\Rightarrow$  RHS



open-interval

mobile-agent

os-signpost

mobile-agent

closed-interval

open-interval

LHS  $\Rightarrow$  RHS



# Facts

Working Memory



# Facts

mobile-agent	
start	0:00:03.000
instance	0x234543534
kind	"Sort Agent"
mode	"Sorting"
state	executing

```
(assert (mobile-agent (start ?start) (instance ?instance ...)))
```

# Facts

f-239

mobile-agent	
start	0:00:03.000
instance	0x234543534
kind	"Sort Agent"
mode	"Sorting"
state	executing

```
(assert (mobile-agent (start ?start) (instance ?instance ...)))
```

# Facts

f-239

mobile-agent	
start	0:00:03.000
instance	0x234543534
kind	"Sort Agent"
mode	"Sorting"
state	executing

# Facts

f-239

mobile-agent	
start	0:00:03.000
instance	0x234543534
kind	"Sort Agent"
mode	"Sorting"
state	executing

```
(modify ?fact-address (mode "Deliver List") (state in-transit))
```

# Facts

Working Memory

```
(modify ?fact-address (mode "Deliver List") (state in-transit))
```

# Facts

f-239

mobile-agent	
start	0:00:03.000
instance	0x234543534
kind	"Sort Agent"
mode	"Deliver List"
state	in-transit

```
(modify ?fact-address (mode "Deliver List") (state in-transit))
```

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance))
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance))
=>
  (modify ?counter (count (+ 1 ?count)))
)
```



```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance))
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance))
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance))
=>
  (modify ?counter (count (+ 1 ?count))))
)
```

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count)) ← f-10
  (mobile-agent (instance ?instance))
=>
  (modify ?counter (count (+ 1 ?count))))
)
```

f-10

mobile-agent-counter	
count	0

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count)) ← f-10
  (mobile-agent (instance ?instance)) ← f-100
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

f-10

mobile-agent-counter	
count	0

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count)) ← f-10
  (mobile-agent (instance ?instance)) ← f-100
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

f-10

mobile-agent-counter	
count	0

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance)) ← f-100
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance)) ← f-100
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

f-10

mobile-agent-counter	
count	1



```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count)) ← f-10
  (mobile-agent (instance ?instance)) ← f-100
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

f-10

mobile-agent-counter	
count	1

```
(defrule MODELER::count-mobile-agent-instances
  ?counter <- (mobile-agent-counter (count ?count))
  (mobile-agent (instance ?instance)) ← f-100
=>
  (modify ?counter (count (+ 1 ?count)))
)
```

# Stuck!

The screenshot shows the Xcode interface for a mobile application named "GoatList" on an iPhone XR (13.0). The interface includes a "Track Filter" search bar, a "Track Filter" button, and a "Track Filter" dropdown menu. The "All Tracks" button is also visible. The "Mobile Agent Logic Loop" track is selected, showing a "Fatal error reported in run 1" at timestamp 00:01.632.475. The error message is "Rules engine appears to be stuck".

Track Filter | All Tracks

00:00.000 | 00:10.000 | 00:

**Mobile Agent Logic Loop** Agents

Instrument

Mobile Agent Logic Loop > Execution

Timestamp ^ Message

! 00:01.632.475 ▼ Fatal error reported in run 1

Rules engine appears to be stuck

# Modeler Console

The screenshot displays the Modeler Console interface. At the top, there are window controls (red, yellow, green buttons) and a title bar with "Run 1". Below the title bar, there are several tabs: "Stores", "Modelers" (selected), and "Schemas".

The main area is divided into two sections. The top section is a table with the following columns: "Stage", "Name", "Debug", and "Class".

Stage	Name	Debug	Class
0	MODELER	3 → (0.1k/s) → 0	COM.APPLE.MOBILEAGENT...

The bottom section is a diagram showing a data flow from a database icon (labeled "os-signpo...t.utilities"}) to a blue circle icon labeled "MODELER".

Below the diagram, there are tabs for "Console" (selected), "Facts", and "RECORDER". The console log shows the following text:

```
=== Loading rules-0003.clp ===
=== Loading rules-0004.clp ===
MODELER> (bind ?*modeler-horizon* 618170976)
Rules engine appears to be stuck
MAIN> (watch all)
FIRE 1 count-mobile-agent-instances: f-18,f-48
<== f-18 (mobile-agent-counter (count 10995))
==> f-18 (mobile-agent-counter (count 10996))
==> Activation 0 count-mobile-agent-instances: f-18,f-48
FIRE 2 count-mobile-agent-instances: f-18,f-48
<== f-18 (mobile-agent-counter (count 10996))
==> f-18 (mobile-agent-counter (count 10997))
==> Activation 0 count-mobile-agent-instances: f-18,f-48
FIRE 3 count-mobile-agent-instances: f-18,f-48
<== f-18 (mobile-agent-counter (count 10997))
--> f-18 (mobile-agent-counter (count 10998))
```

```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)

(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)

(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```

```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)

(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)

(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```

```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)

(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)

(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```

```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)

(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)

(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```



```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)

(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)

(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```

```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)

(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)

(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```

```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)
(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)
(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```

```
(deftemplate MODELER::bump-agent-counter
  (slot instance (type INTEGER))
)

(defrule MODELER::count-mobile-agent-instances
  (mobile-agent (instance ?instance))
  =>
  (assert (bump-agent-counter (instance ?instance)))
)

(defrule MODELER::update-agent-counter
  ?counter <- (mobile-agent-counter (count ?count))
  ?goal <- (bump-agent-counter)
  =>
  (retract ?goal)
  (modify ?counter (count (+ ?count 1)))
)
```

# Firing Order

```
func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %@ executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())
    execute()
}
```

```
func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %@ executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())
    execute()
}
```

```
func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %@ executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())
    execute()
}
```

"Sorting Agent" = 14 bytes



```
func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %d executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())
    execute()
}
```

```
func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %d executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())
    execute()
}
```

```
func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %d executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())
    execute()
}
```

0x00000001 = 4 bytes

```
func executeStop() {
    os_signpost(.event,
        log: MobileAgent.signpostHandleID,
        name: "Mobile Agent Exec",
        signpostID: signpostID,
        "Agent of type %d executing mode %@. At stop %@",
        agentType(), currentMode, stop.agentStopType())
    execute()
}
```

0x00000001 = 4 bytes

Save 10 bytes per event

```
(deffacts MODELER::kind-code-mapping
  (agent-kind-code-to-name (kind-code 1) (kind "Sorting Agent"))
  (agent-kind-code-to-name (kind-code 2) (kind "Display Agent"))
  (agent-kind-code-to-name (kind-code 3) (kind "Editing Agent"))
)
```

```
(deffacts MODELER::kind-code-mapping
  (agent-kind-code-to-name (kind-code 1) (kind "Sorting Agent"))
  (agent-kind-code-to-name (kind-code 2) (kind "Display Agent"))
  (agent-kind-code-to-name (kind-code 3) (kind "Editing Agent"))
)
```

```
(defrule MODELER::detect-new-mobile-agent
  (os-signpost
    (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
                        (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)
```

```
(defrule MODELER::detect-new-mobile-agent
  (os-signpost
    (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
                        (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)
```



```

(defrule MODELER::detect-new-mobile-agent
  (os-signpost
    (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)

```

mobile-agent	
kind	sentinel
kind-code	0x1

```

(defrule MODELER::detect-new-mobile-agent
  (os-signpost
    (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
                        (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)

```

mobile-agent	
kind	sentinel
kind-code	0x1

```

(defrule MODELER::detect-new-mobile-agent
  (os-signpost
    (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)

```

mobile-agent	
kind	sentinel
kind-code	0x1

```

(defrule MODELER::detect-new-mobile-agent
  (os-signpost
    (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)

```

mobile-agent	
kind	sentinel
kind-code	0x1

```

(defrule MODELER::detect-new-mobile-agent
  (os-signpost
    (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
                        (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)

```

mobile-agent	
kind	"Sorting Agent"
kind-code	0x1

```
(defrule MODELER::agent-parked
  (os-signpost (time ?end) (name "Mobile Agent Parked")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Parked in mode " ?mode)
  )
  ?a <- (mobile-agent (instance ?instance) (kind ?kind))
  ?transition-begin <- (mobile-agent-transition-started (start ?start) (instance ?instance)
(stop-kind ?stop-kind&~sentinel) (state ?state))

  =>
  ;; Actions
)
```

```
(defrule MODELER::agent-parked
  (os-signpost (time ?end) (name "Mobile Agent Parked")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Parked in mode " ?mode)
  )
  ?a <- (mobile-agent (instance ?instance) (kind ?kind))
  ?transition-begin <- (mobile-agent-transition-started (start ?start) (instance ?instance)
    (stop-kind ?stop-kind&~sentinel) (state ?state))

  =>
  ;; Actions
)
```

```
(defrule MODELER::agent-parked
  (os-signpost (time ?end) (name "Mobile Agent Parked")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Parked in mode " ?mode)
  )
  ?a <- (mobile-agent (instance ?instance) (kind ?kind&~sentinel))
  ?transition-begin <- (mobile-agent-transition-started (start ?start) (instance ?instance)
    (stop-kind ?stop-kind&~sentinel) (state ?state))

  =>
  ;; Actions
)
```



```
(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
                        (kind sentinel) (kind-code ?kind-code)))
)

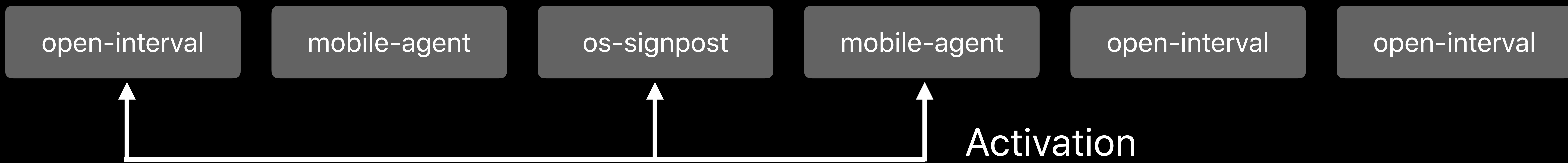
(defrule MODELER::lookup-agent-kind
  (declare (salience 100))
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)
```

```
(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
                        (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  (declare (salience 100))
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)
```

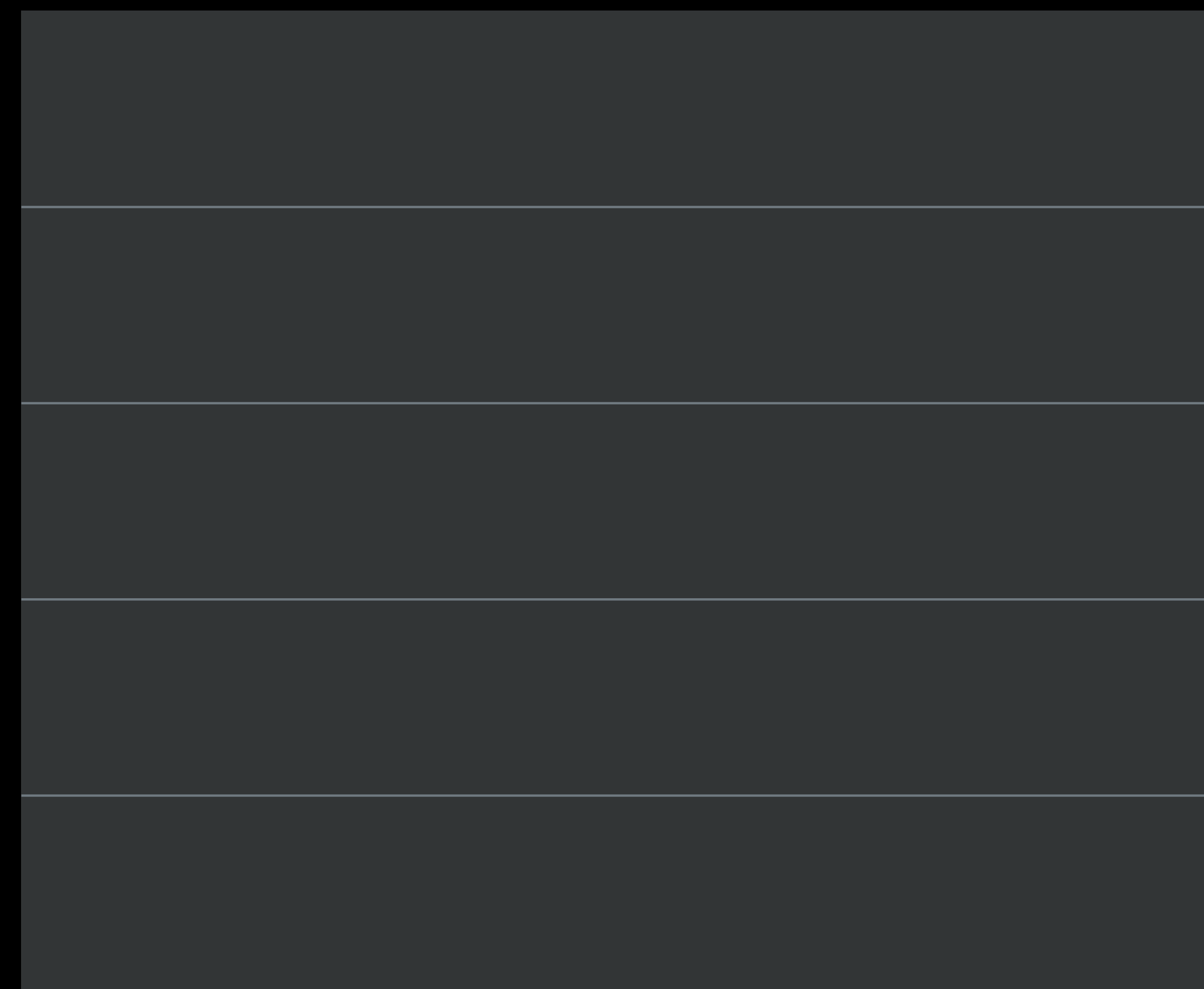
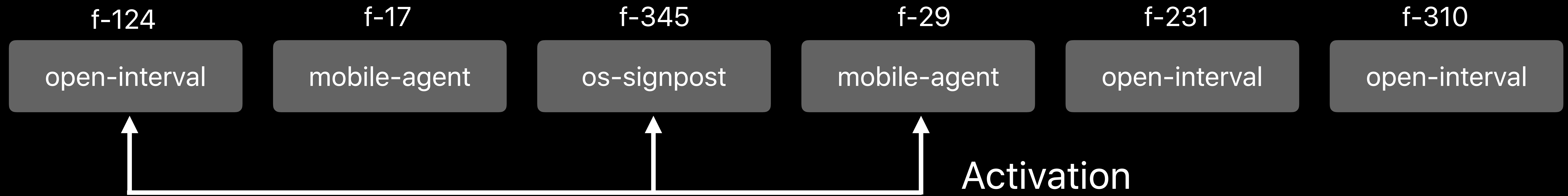
```
(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0)
    (name "Mobile Agent Moved")
    (event-type "Event")
    (identifier ?instance)
    (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
=>
  (assert (mobile-agent (start ?t) (instance ?instance)
                        (kind sentinel) (kind-code ?kind-code)))
)

(defrule MODELER::lookup-agent-kind
  (declare (salience 100))
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
=>
  (modify ?a (kind ?string))
)
```



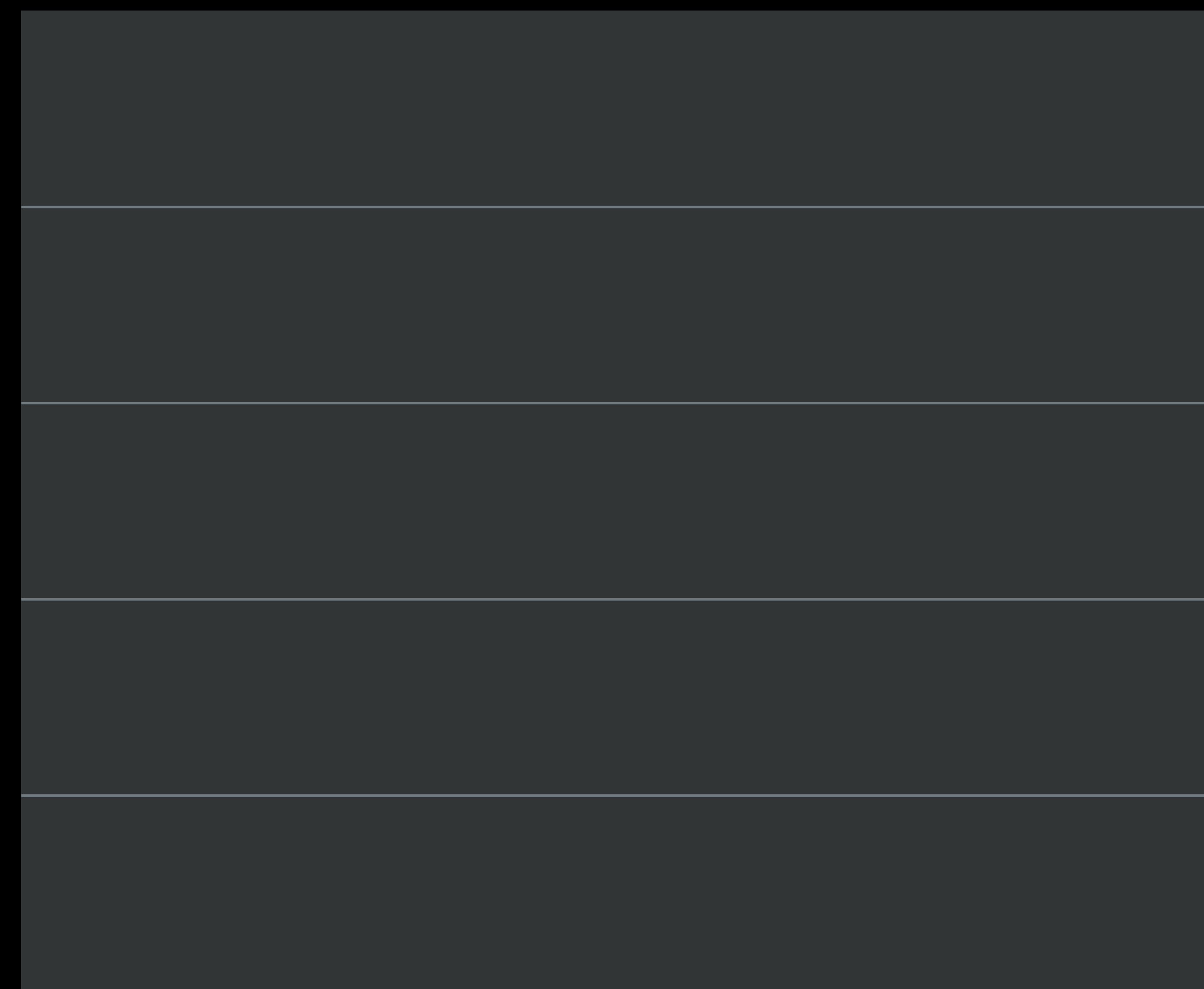
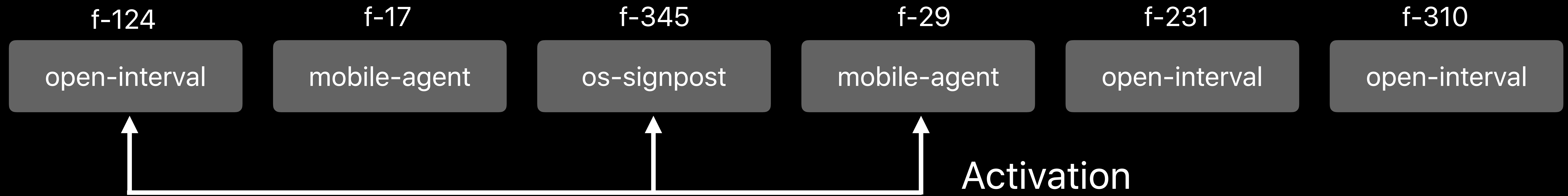
LHS => RHS

# Activation



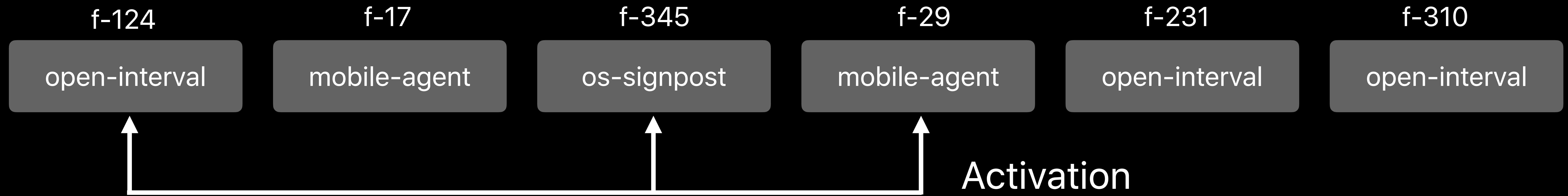
Agenda

# Activation



Agenda

# Activation



```
rule 1: f-345, f-29, f-124
```

Agenda

# Agenda

rule 1: f-345, f-29, f-124

rule 99: f-345, f-17, f-231

rule 15: f-93, f-230

rule 9: f-17

rule 12: f-99, f-102, f-231

rule 14: f-99, f-17



# Agenda



rule 1: f-345, f-29, f-124

rule 99: f-345, f-17, f-231

rule 15: f-93, f-230

rule 9: f-17

rule 12: f-99, f-102, f-231

rule 14: f-99, f-17

# Agenda

~~rule 1: f-345, f-29, f-124~~



rule 99: f-345, f-17, f-231

rule 15: f-93, f-230

rule 9: f-17

rule 12: f-99, f-102, f-231

rule 14: f-99, f-17

# Agenda

~~rule 1: f-345, f-29, f-124~~



rule 99: f-345, f-17, f-231

rule 15: f-93, f-230

rule 9: f-17

rule 12: f-99, f-102, f-231

rule 14: f-99, f-17

(retract f-17)

# Agenda

~~rule 1: f-345, f-29, f-124~~



rule 99: f-345, f-17, f-231

rule 15: f-93, f-230

rule 9: f-17

rule 12: f-99, f-102, f-231

rule 14: f-99, f-17

# Agenda

~~rule 1: f-345, f-29, f-124~~

→ rule 99: f-345, f-17, f-231

rule 15: f-93, f-230

← rule 9: f-17

rule 12: f-99, f-102, f-231

← rule 14: f-99, f-17

# Agenda

~~rule 1: f-345, f-29, f-124~~

~~rule 99: f-345, f-17, f-231~~

→ rule 15: f-93, f-230

rule 12: f-99, f-102, f-231

# Separate Agendas

# Separate Agendas

Saliency



# Separate Agendas

Saliency

Modules\*

# Separate Agendas

Saliency

Modules\*

Standard modules

# Separate Agendas

Saliency

Modules\*

Standard modules

MODELER

\*Modules in CLIPS are not related to modules in Swift.

# Separate Agendas

Saliency

Modules\*

Standard modules

MODELER

RECORDER

# Separate Agendas

Saliency

Modules\*

Standard modules

MODELER

RECORDER

```
detect-agent: f-345
```

```
close-interval: f-345, f-10
```

```
open-interval: f-345, f-234
```

MODELER

```
record-interval: f-986
```

RECORDER

\*Modules in CLIPS are not related to modules in Swift.

# Separate Agendas

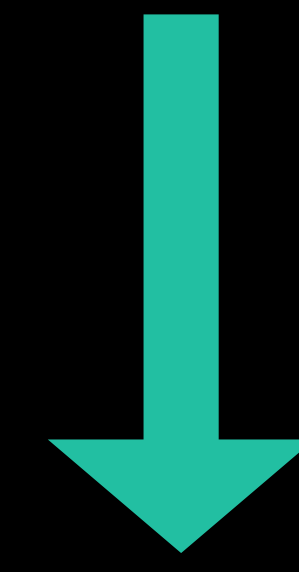
Saliency

Modules\*

Standard modules

MODELER

RECORDER



detect-agent: f-345

close-interval: f-345, f-10

open-interval: f-345, f-234

MODELER

record-interval: f-986

RECORDER

\*Modules in CLIPS are not related to modules in Swift.

# Separate Agendas

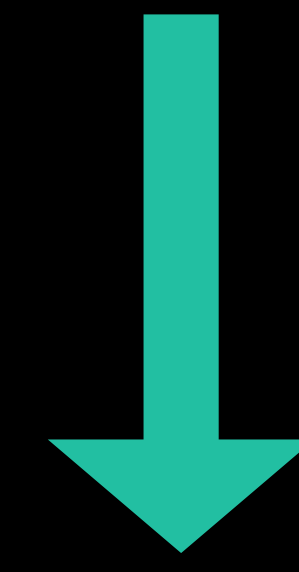
Saliency

Modules\*

Standard modules

MODELER

RECORDER



```
detect-agent: f-345
```

```
close-interval: f-345, f-10
```

```
open-interval: f-345, f-234
```

MODELER

```
record-interval: f-986
```

RECORDER

\*Modules in CLIPS are not related to modules in Swift.

# Separate Agendas

Saliency

Modules\*

Standard modules

MODELER

RECORDER

```
detect-agent: f-345
```

```
close-interval: f-345, f-10
```

```
open-interval: f-345, f-234
```

MODELER



```
record-interval: f-986
```

RECORDER

\*Modules in CLIPS are not related to modules in Swift.



# Separate Agendas

Saliency

Modules\*

Standard modules

MODELER

RECORDER

Custom modules

```
detect-agent: f-345
```

```
close-interval: f-345, f-10
```

```
open-interval: f-345, f-234
```

MODELER



```
record-interval: f-986
```

RECORDER

```
(defmodule LOOKUP
  (import MODELER ?ALL)
)

(defrule LOOKUP::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
  =>
  (modify ?a (kind ?string))
)

(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0) (name "Mobile Agent Moved") (event-type "Event")
    (identifier ?instance) (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
  =>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))

  (focus LOOKUP)
)
```

```
(defmodule LOOKUP
  (import MODELER ?ALL)
)

(defrule LOOKUP::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
  =>
  (modify ?a (kind ?string))
)

(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0) (name "Mobile Agent Moved") (event-type "Event")
    (identifier ?instance) (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
  =>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
  (focus LOOKUP)
)
```

```
(defmodule LOOKUP
  (import MODELER ?ALL)
)

(defrule LOOKUP::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
  =>
  (modify ?a (kind ?string))
)

(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0) (name "Mobile Agent Moved") (event-type "Event")
    (identifier ?instance) (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
  =>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
  (focus LOOKUP)
)
```

```
(defmodule LOOKUP
  (import MODELER ?ALL)
)

(defrule LOOKUP::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
  =>
  (modify ?a (kind ?string))
)

(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0) (name "Mobile Agent Moved") (event-type "Event")
    (identifier ?instance) (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
  =>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
  (focus LOOKUP)
)
```

```
(defmodule LOOKUP
  (import MODELER ?ALL)
)
(defrule LOOKUP::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
  =>
  (modify ?a (kind ?string))
)
(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0) (name "Mobile Agent Moved") (event-type "Event")
    (identifier ?instance) (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
  =>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
  (focus LOOKUP)
)
```

```
(defmodule LOOKUP
  (import MODELER ?ALL)
)

(defrule LOOKUP::lookup-agent-kind
  ?a <- (mobile-agent (kind sentinel) (kind-code ?code))
  (agent-kind-code-to-name (kind-code ?code) (kind ?string))
  =>
  (modify ?a (kind ?string))
)

(defrule MODELER::detect-new-mobile-agent
  (os-signpost (time ?t&~0) (name "Mobile Agent Moved") (event-type "Event")
    (identifier ?instance) (message$ "Agent of type " ?kind-code $?))
  (not (mobile-agent (instance ?instance)))
  =>
  (assert (mobile-agent (start ?t) (instance ?instance)
    (kind sentinel) (kind-code ?kind-code)))
  (focus LOOKUP)
)
```

# Debugging and Profiling

Alejandro Lucena, Performance Tools



# Logging

# Logging

printf style tracing

# Logging

printf style tracing

Dynamically enabled/disabled

```
//Logging
```

```
(defrule MODELER::lookup-known-agent-kind
```

```
  ?agent <- (mobile-agent (instance ?instance) (kind-code ?kind-code))
```

```
  (agent-kind-code-to-name (kind-code ?kind-code) (kind ?kind))
```

```
  =>
```

```
  (modify ?agent (kind ?kind))
```

```
  (log-narrative "Resolved agent kind code %uint64% to %string%" ?kind-code ?kind)
```

```
)
```

# Profiling

# Profiling

Rule activation counts

# Profiling

Rule activation counts

Time distribution

***Demo***

Alejandro Lucena, Performance Tools



# Speculation

Chad Wolf, Performance Tools

# What Happened

# What Happened

Long intervals

# What Happened

Long intervals

Only in working memory

# What Happened

Long intervals

Only in working memory

UI only sees the output tables

# What Happened

Long intervals

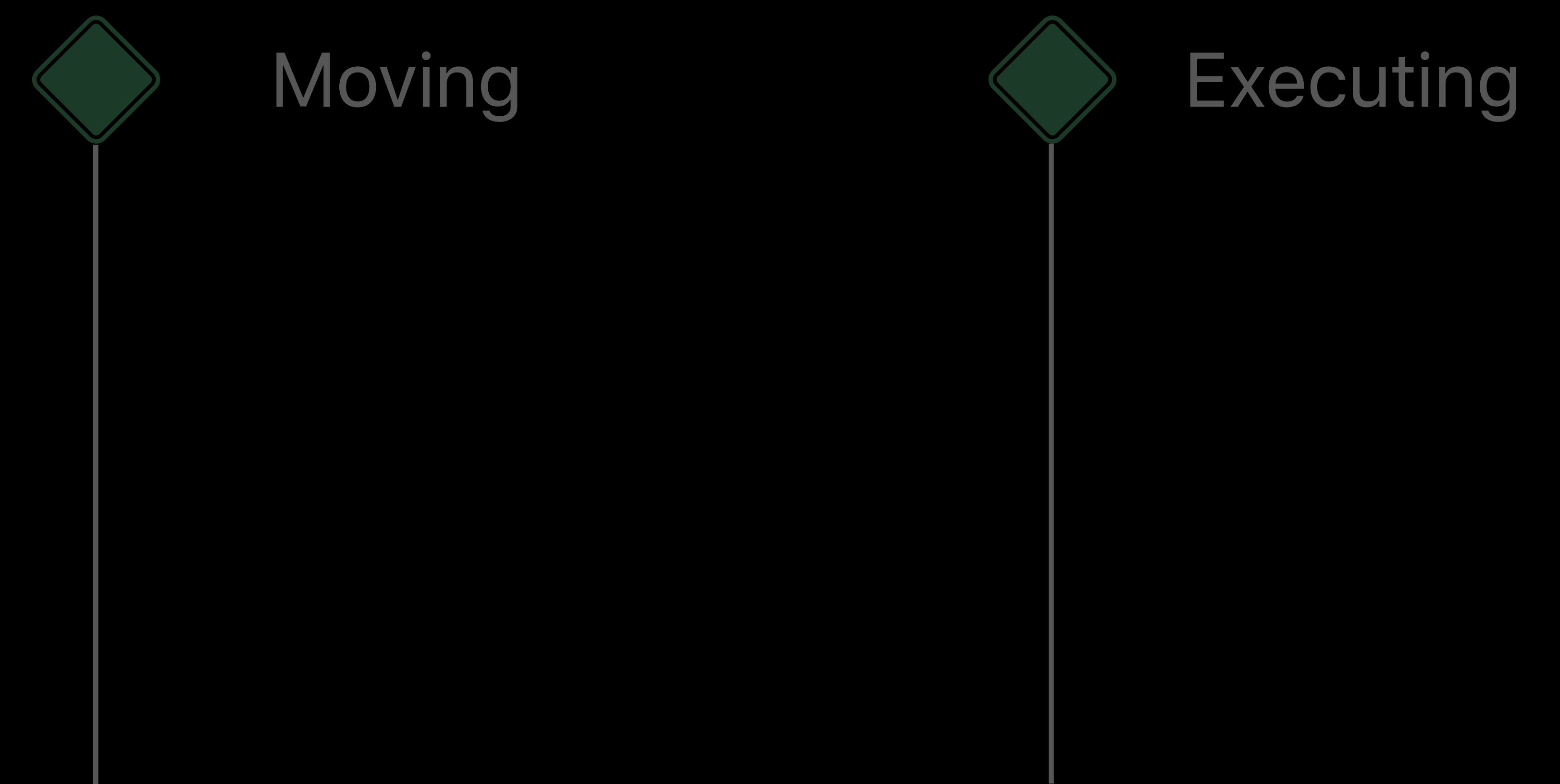
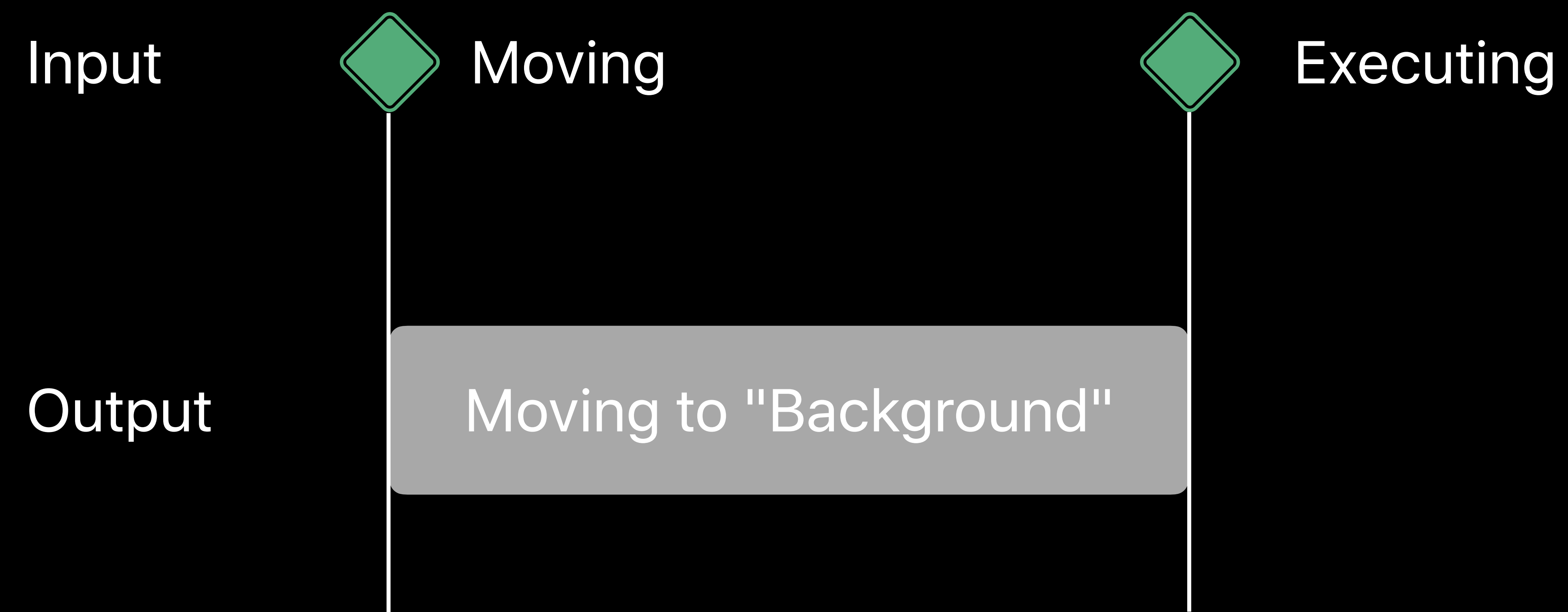
Only in working memory

UI only sees the output tables

Speculation mode

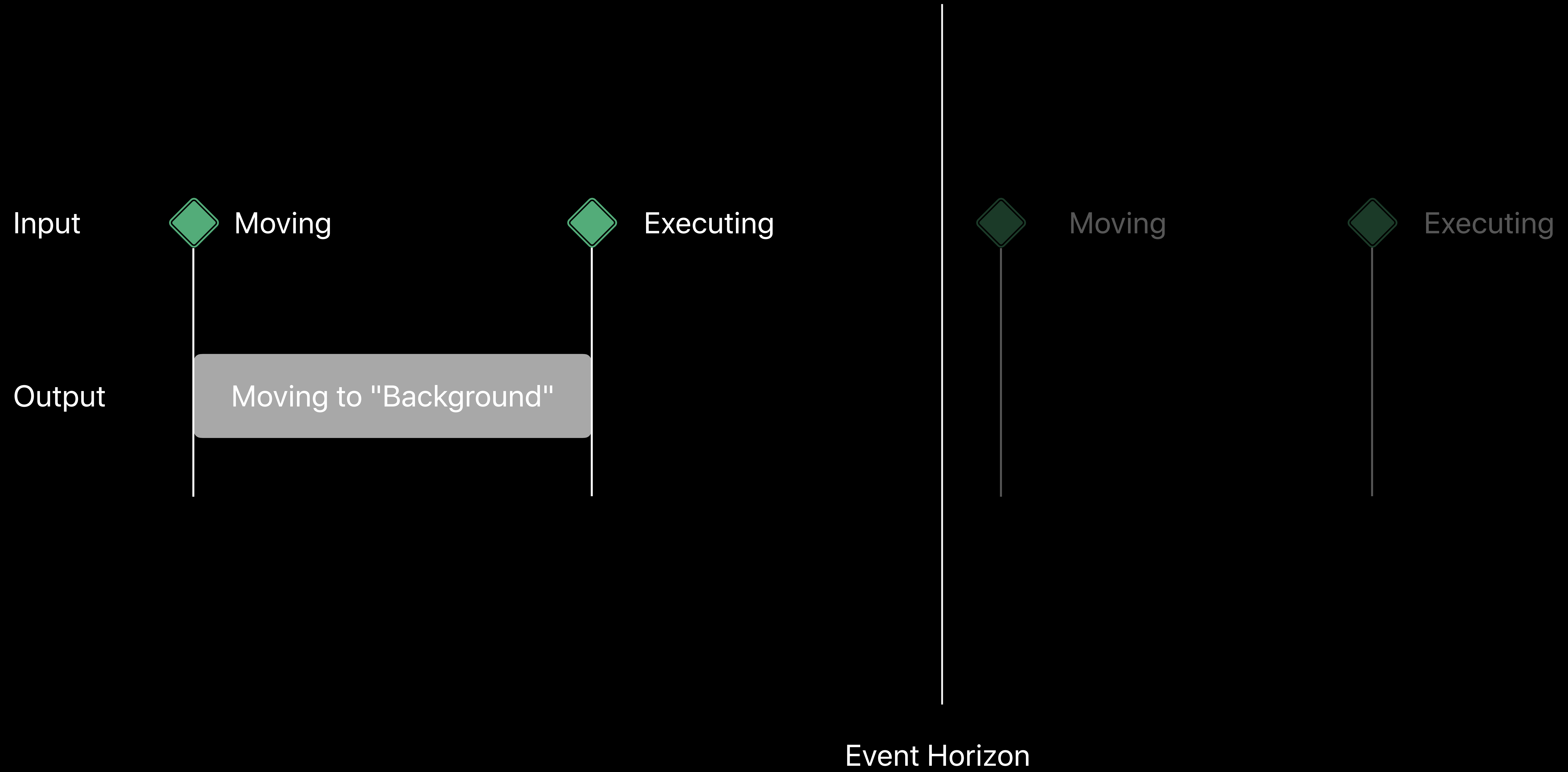
If this were your last chance to write,  
what would you write?

# Speculation

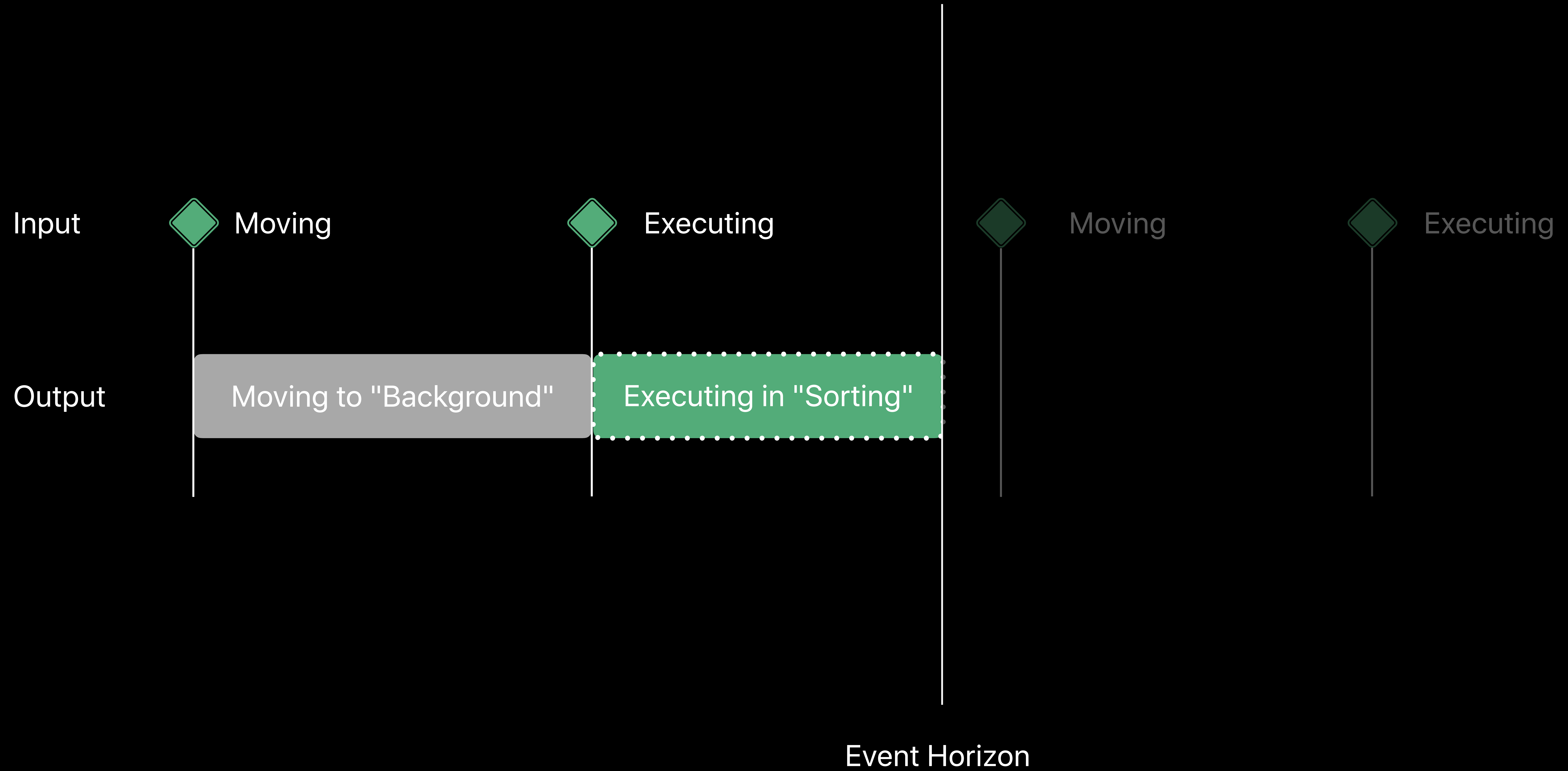




# Speculation



# Speculation



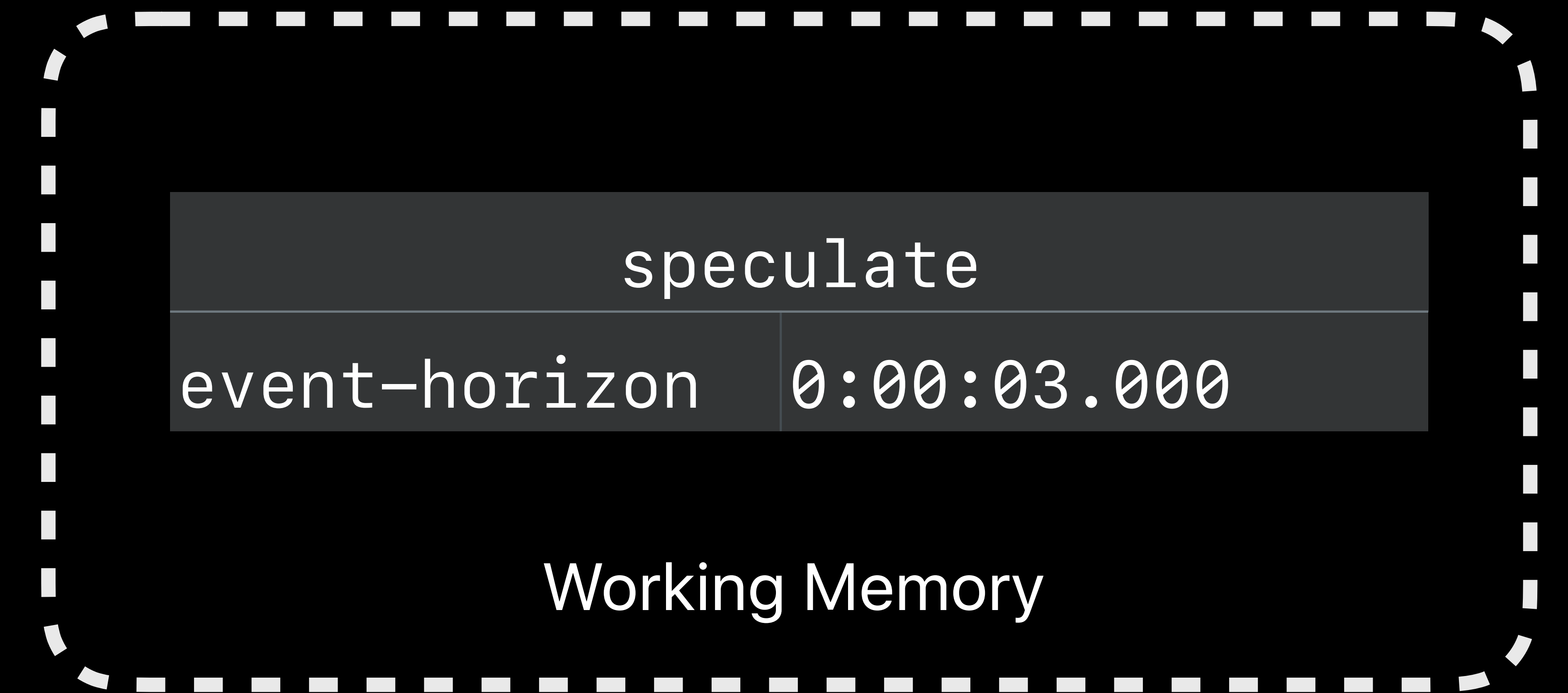
# Speculation Mode

```
speculate  
event-horizon 0:00:03.000
```

Working Memory

# Speculation Mode

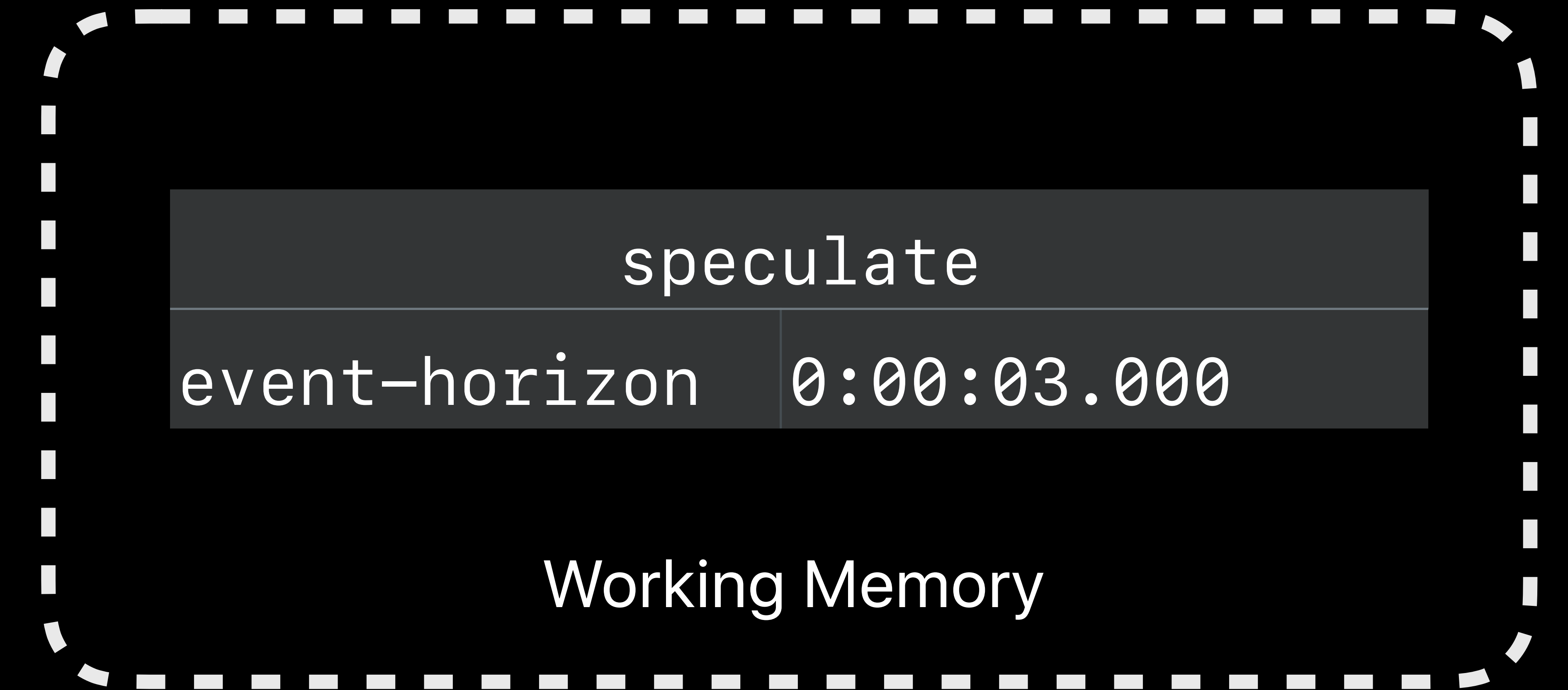
Speculate fact



# Speculation Mode

Speculate fact

Added to rules

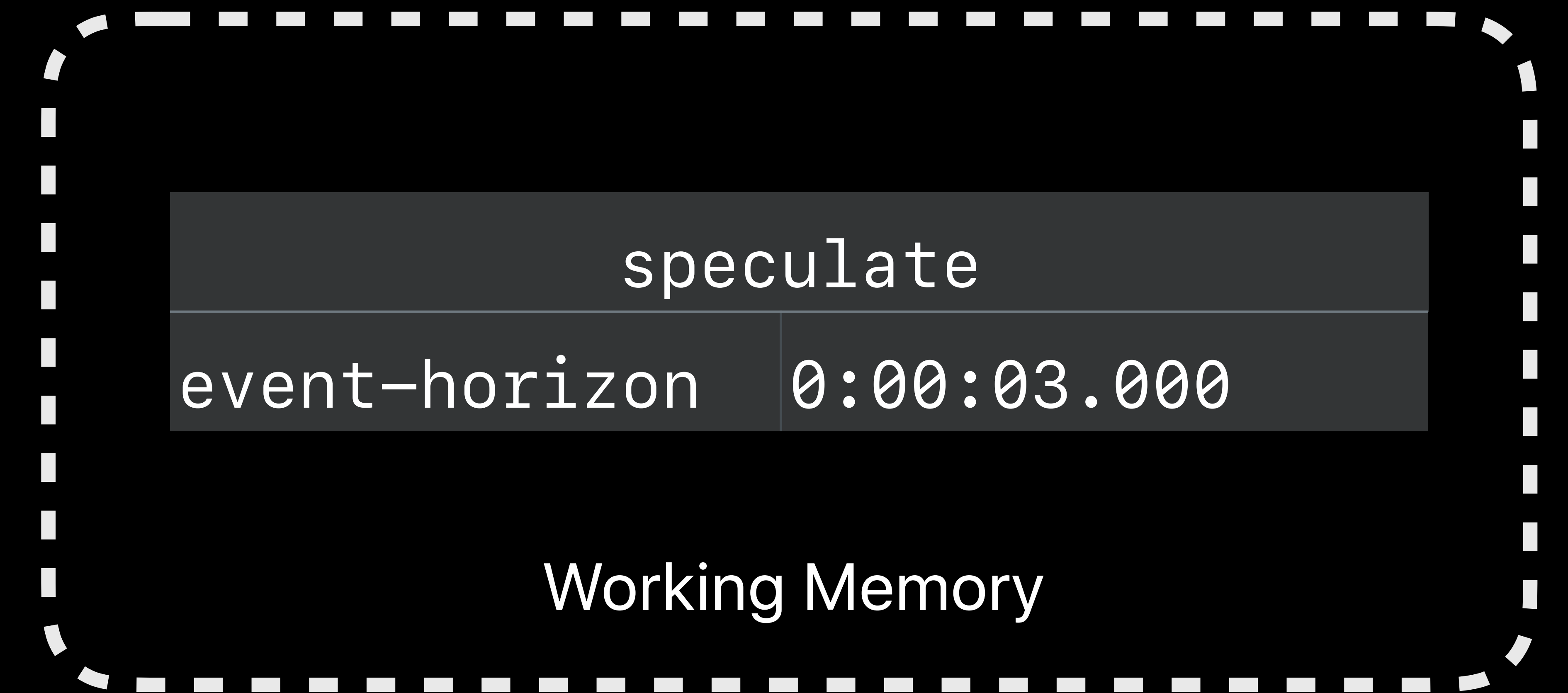


# Speculation Mode

Speculate fact

Added to rules

Write open intervals



```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```



```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```



```
(defrule RECORDER::speculatively-record-execution
  (speculate (event-horizon ?end))
  (table (table-id ?output) (side append))
  (table-attribute (table-id ?output) (has schema mobile-agent-activity))
  (mobile-agent-execution-started (start ?start)
    (instance ?instance) (stop-kind ?stop-kind&~sentinel))
=>
  (bind ?duration (- ?end ?start))
  (create-new-row ?output)
  (set-column start ?start)
  (set-column duration ?duration)
  (set-column instance ?instance)
  (set-column state "Executing")
  (set-column activity-type "Green")
  (set-column stop-kind ?stop-kind)
  (set-column-narrative activity "Executing at stop %string%" ?stop-kind)
)
```

# Immediate Mode

# Immediate Mode

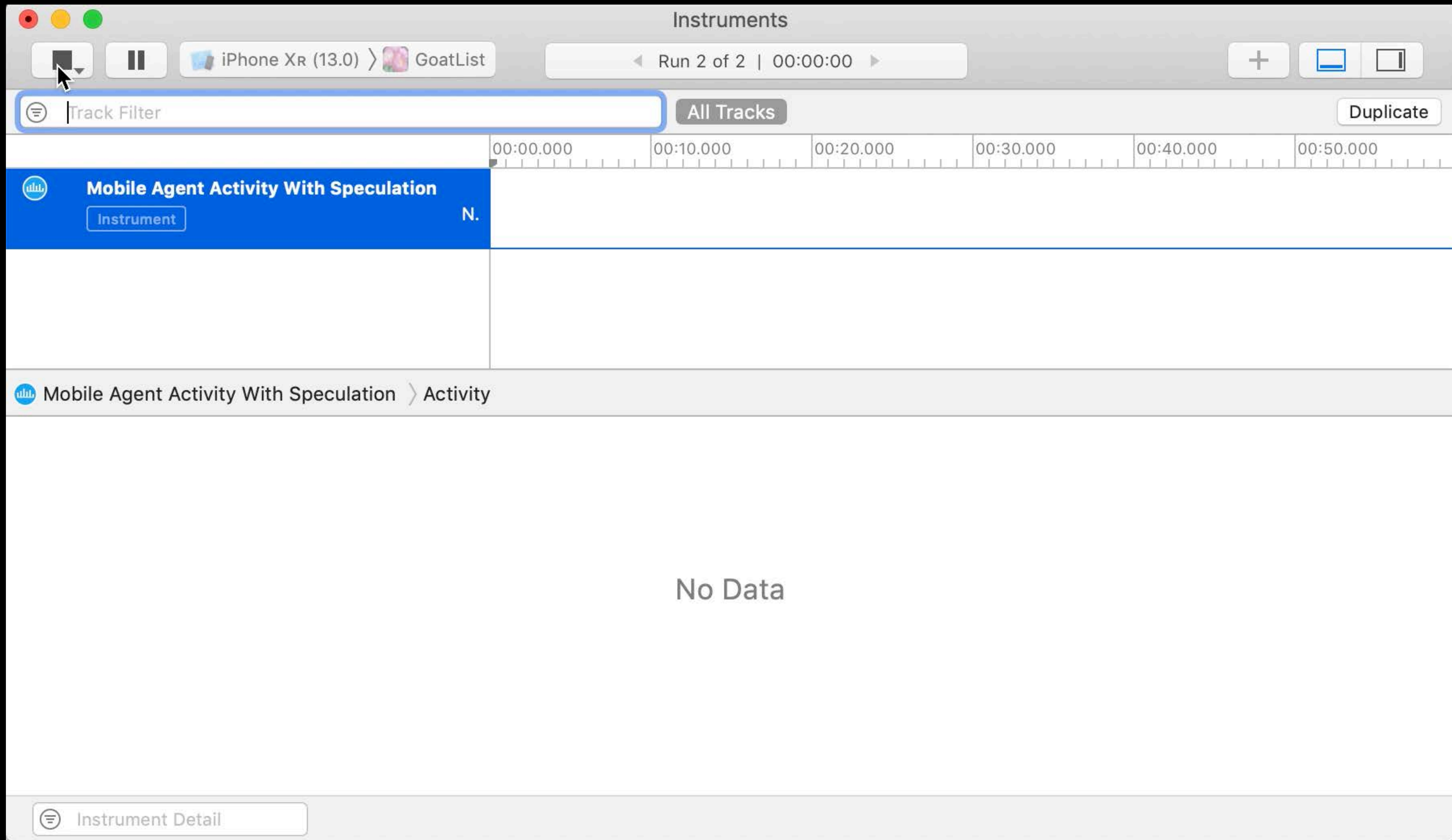
Purged

# Immediate Mode

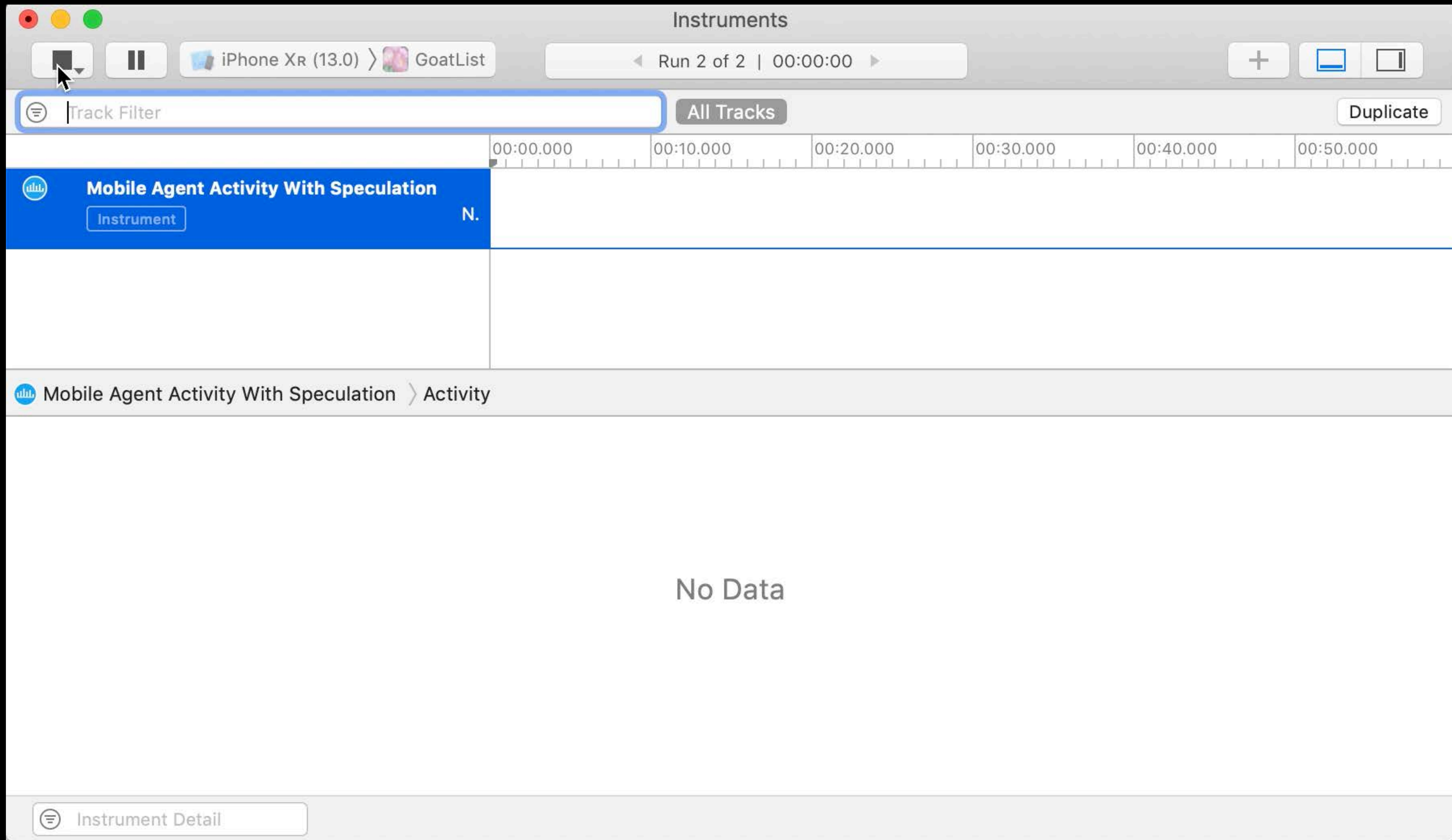
Purged

Become real when trace ends

# Speculation in Action



# Speculation in Action



# Summary

# Summary

Investment



# Summary

Investment

Intelligence

# Summary

Investment

Intelligence

Efficiency

# More Information

[developer.apple.com/wwdc19/421](https://developer.apple.com/wwdc19/421)

 WWDC19