Event Sourcing with Elixir

By Peter Ullrich

Why use Event Sourcing?

Why use Event-Sourcing?

1. Auditing
2. Debugging
3. Historic State (aka. Time Travel)
4. Alternative State
5. Recover lost state
6. Memory Image

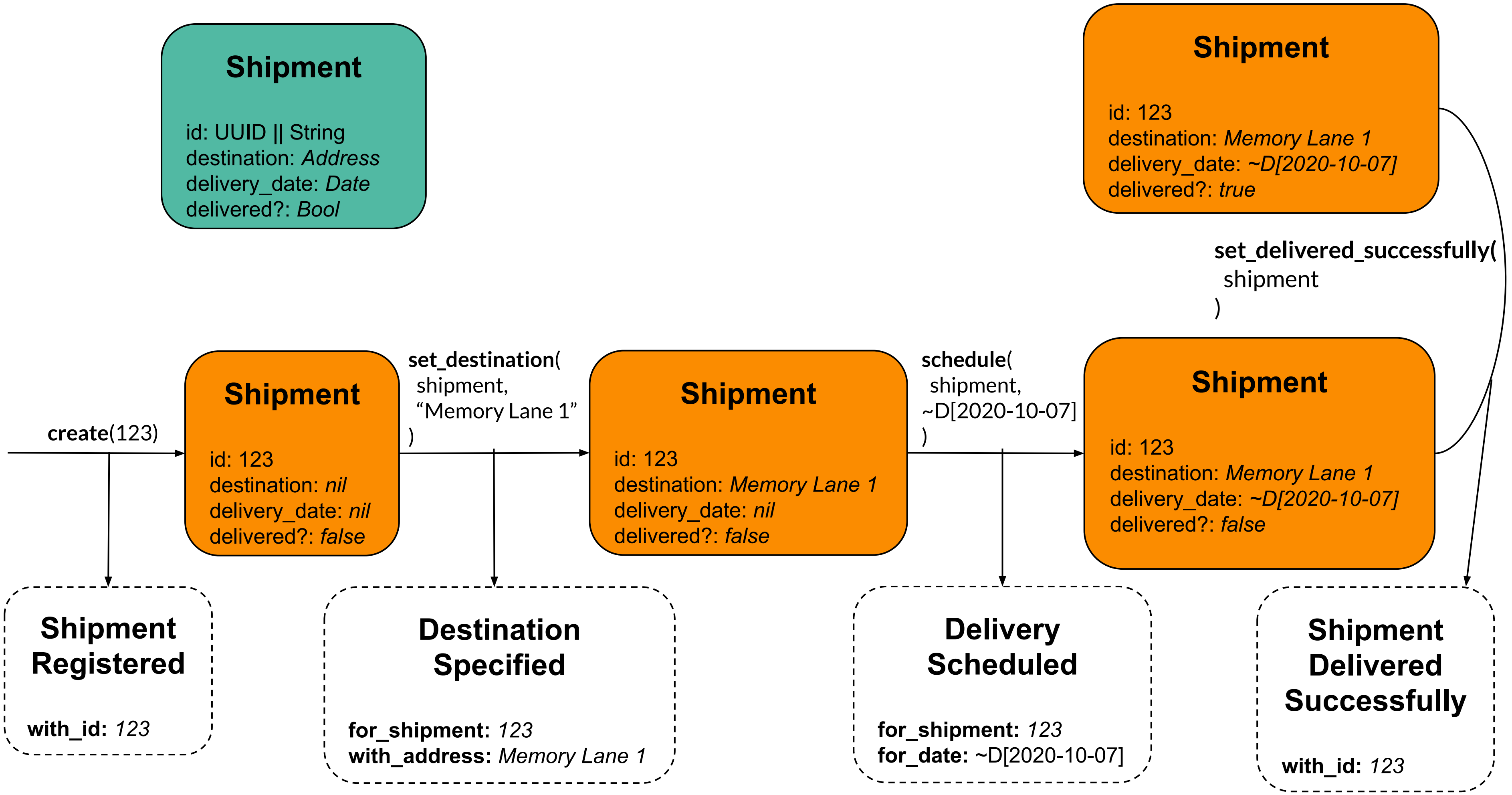
What is “Event Sourcing”?

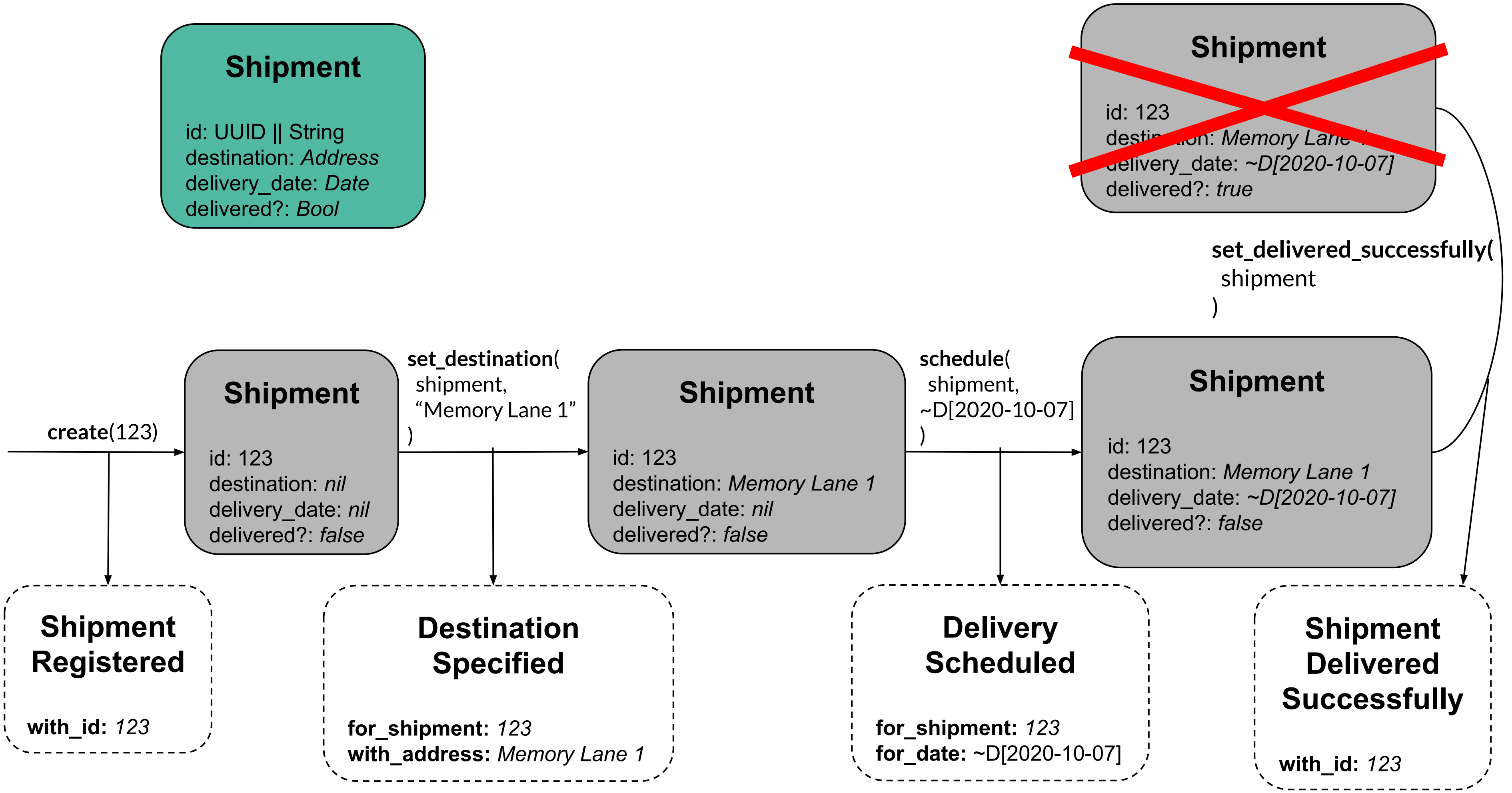
What is Event-Sourcing?

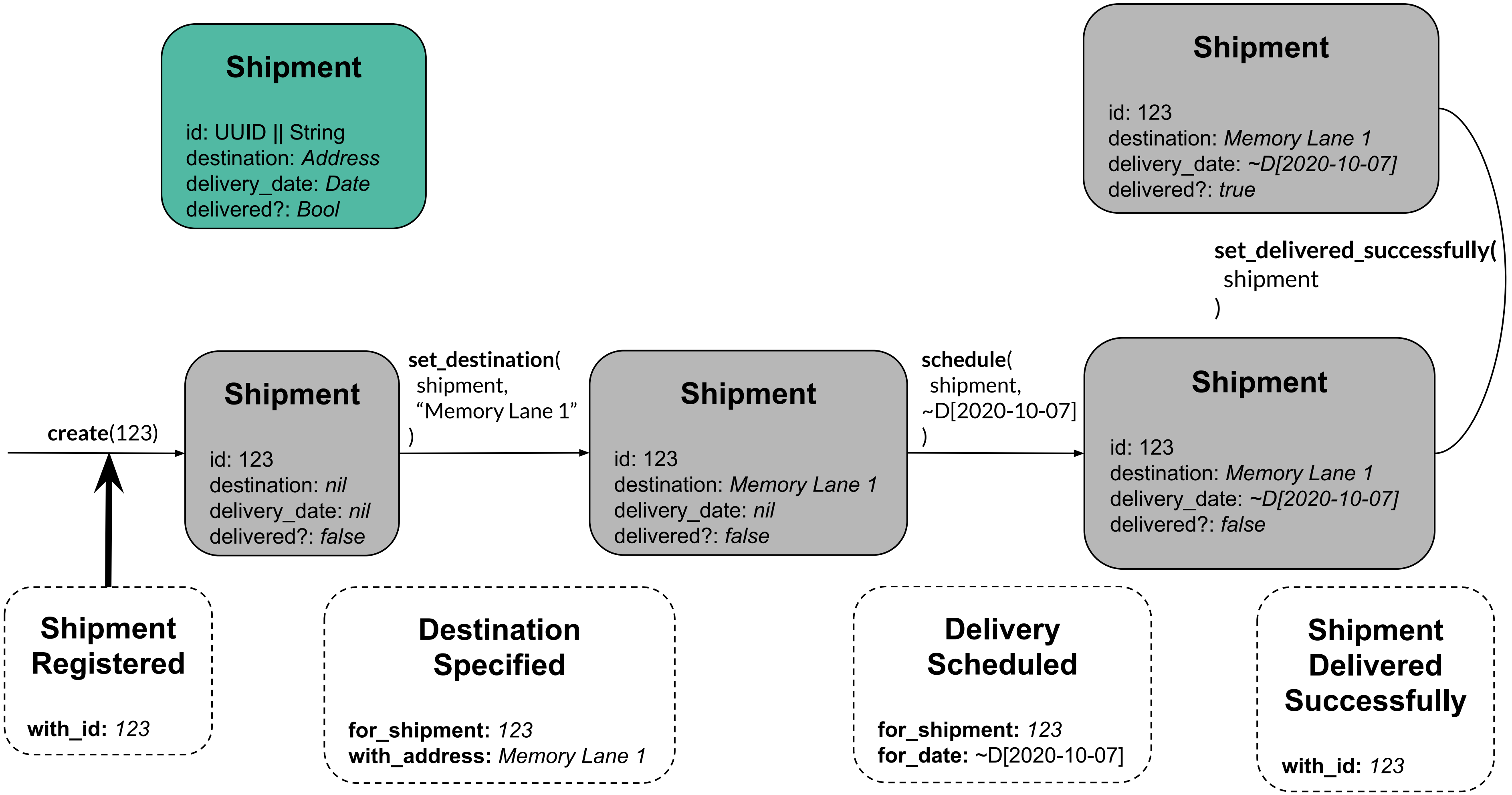
Technique

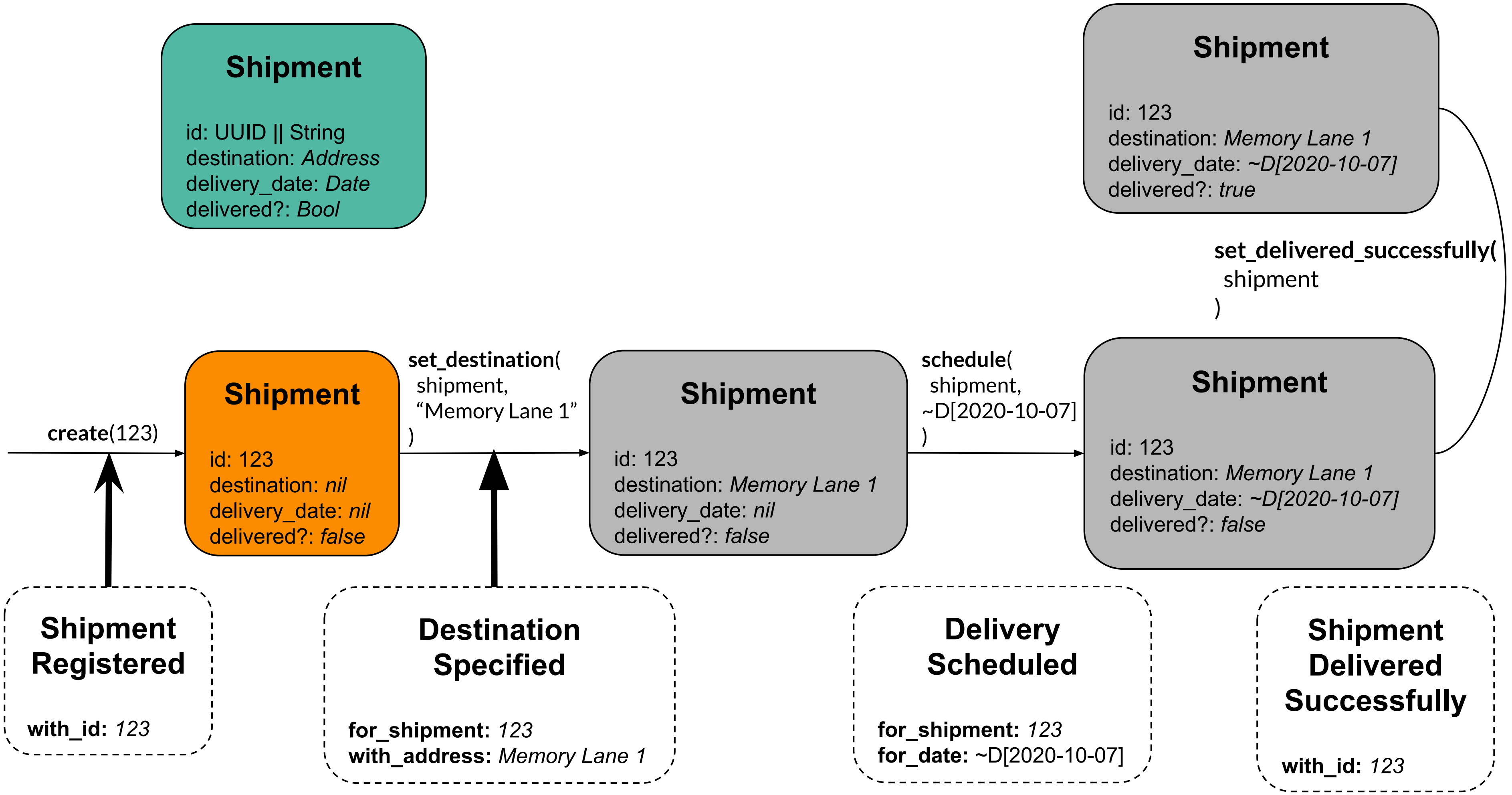
1. Document every Application State Change with an Event
2. Persist all Events in a (separate) storage system
3. That’s it (more or less)

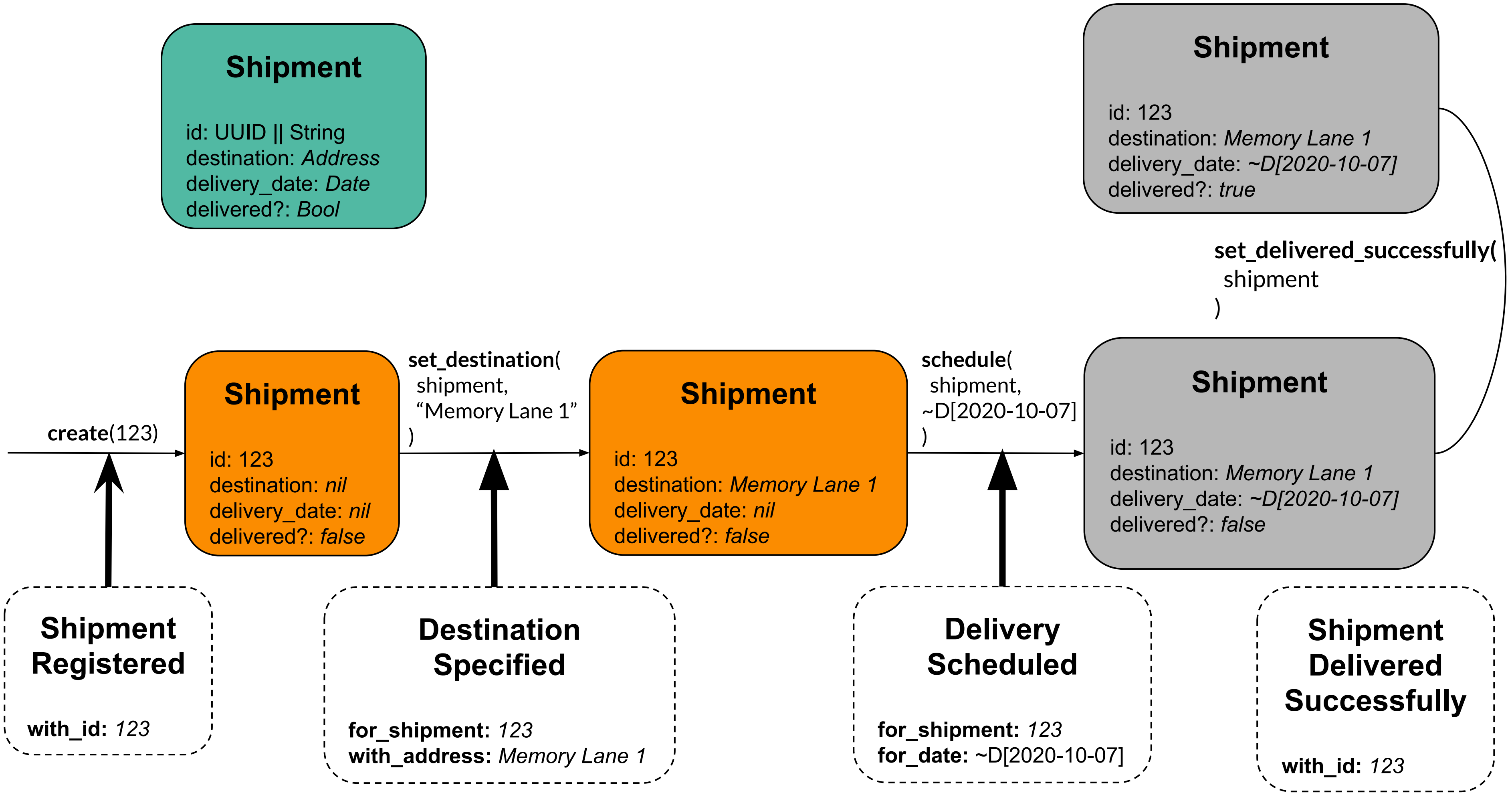
Example

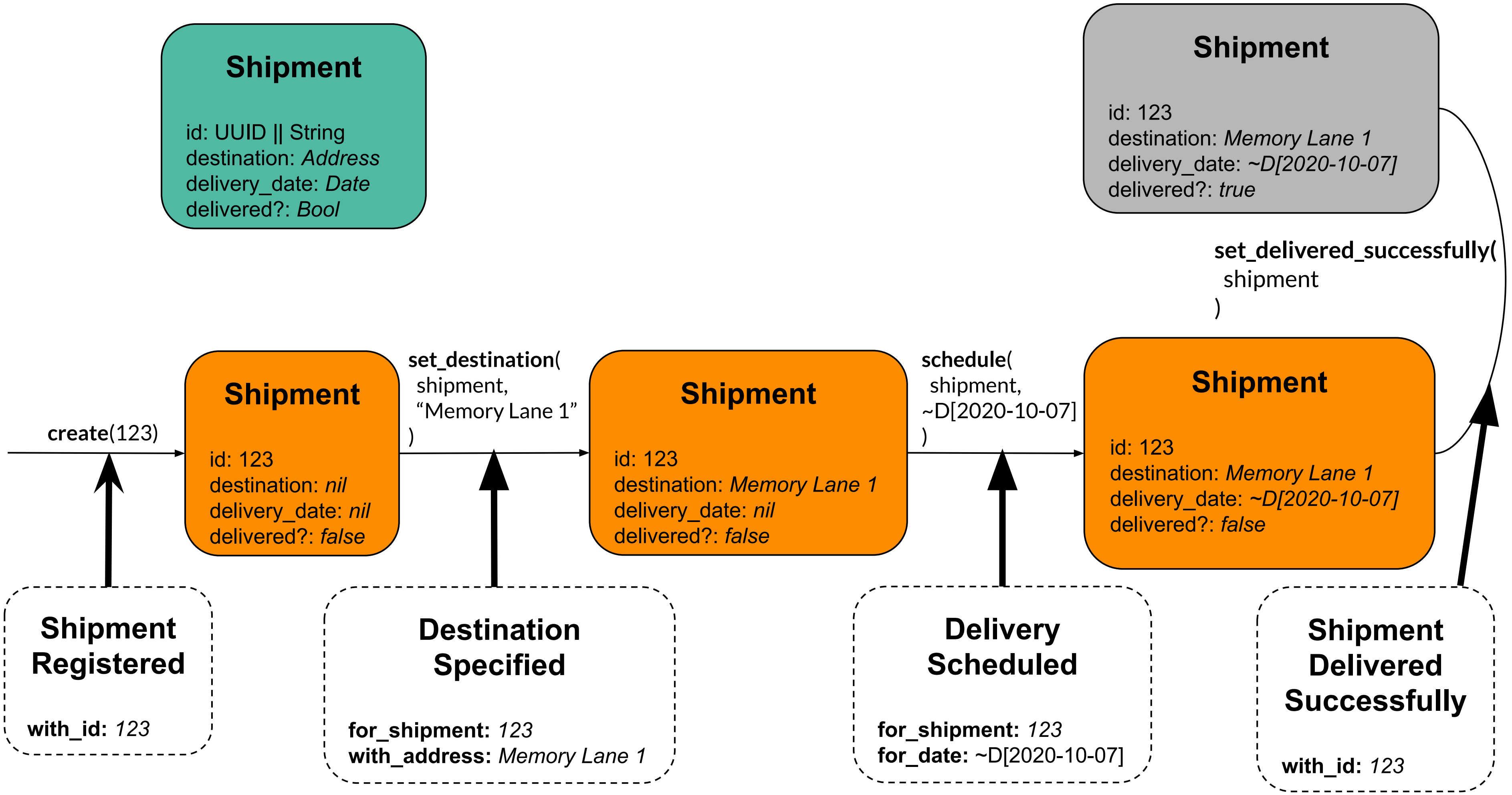

Shipment
id: UUID || String
destination: *Address*
delivery_date: *Date*
delivered?: *Bool*
create(123)
Shipment
id: 123
destination: *nil*
delivery_date: *nil*
delivered?: *false*
set_destination(shipment, "Memory Lane 1")
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *nil*
delivered?: *false*
schedule(shipment, ~D[2020-10-07])
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *false*
set_delivered_successfully(shipment)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *true*
Shipment Registered
with_id: *123*
Destination Specified
for_shipment: *123*
with_address: *Memory Lane 1*
Delivery Scheduled
for_shipment: *123*
for_date: ~D[2020-10-07]
Shipment Delivered Successfully
with_id: *123*

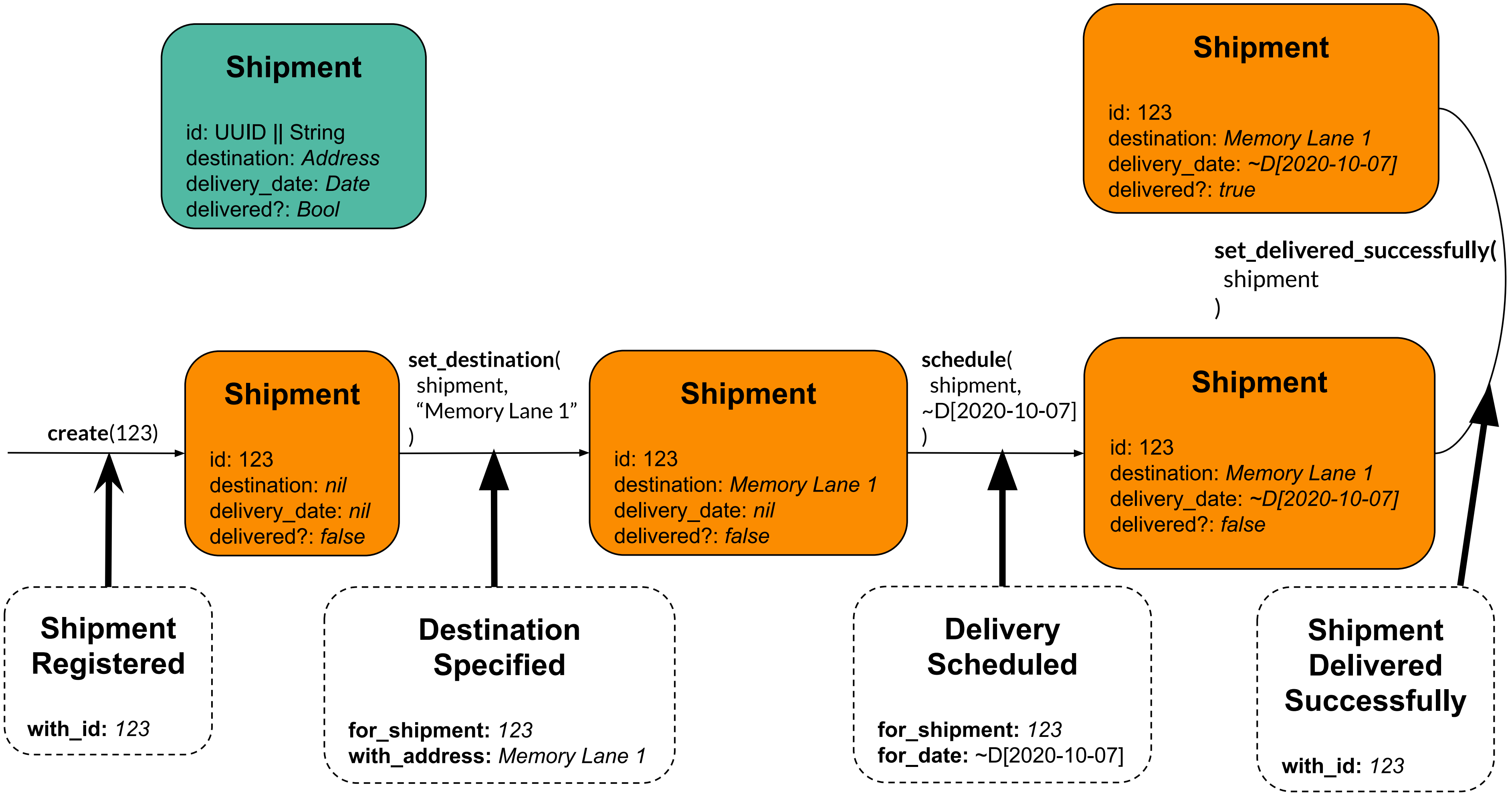

Shipment
id: UUID || String
destination: Address
delivery_date: Date
delivered?: Bool
Shipment
id: 123
destination: Memory Lane 1
delivery_date: ~D[2020-10-07]
delivered?: true
set_delivered_successfully(
shipment
)
create(123)
Shipment
id: 123
destination: nil
delivery_date: nil
delivered?: false
set_destination(
shipment,
“Memory Lane 1”
)
Shipment
id: 123
destination: Memory Lane 1
delivery_date: nil
delivered?: false
schedule(
shipment,
~D[2020-10-07]
)
Shipment
id: 123
destination: Memory Lane 1
delivery_date: ~D[2020-10-07]
delivered?: false
Shipment Registered
with_id: 123
Destination Specified
for_shipment: 123
with_address: Memory Lane 1
Delivery Scheduled
for_shipment: 123
for_date: ~D[2020-10-07]
Shipment Delivered Successfully
with_id: 123


Shipment
id: UUID || String
destination: *Address*
delivery_date: *Date*
delivered?: *Bool*
create(123)
Shipment
id: 123
destination: *nil*
delivery_date: *nil*
delivered?: *false*
set_destination(
shipment,
"Memory Lane 1"
)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *nil*
delivered?: *false*
schedule(
shipment,
~D[2020-10-07]
)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *false*
set_delivered_successfully(
shipment
)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *true*
Shipment Registered
with_id: *123*
Destination Specified
for_shipment: *123*
with_address: *Memory Lane 1*
Delivery Scheduled
for_shipment: *123*
for_date: ~D[2020-10-07]
Shipment Delivered Successfully
with_id: *123*


Shipment
id: UUID || String
destination: *Address*
delivery_date: *Date*
delivered?: *Bool*
create(123)
Shipment
id: 123
destination: *nil*
delivery_date: *nil*
delivered?: *false*
set_destination(shipment, “Memory Lane 1”)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *nil*
delivered?: *false*
schedule(shipment, ~D[2020-10-07])
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *false*
set_delivered_successfully(shipment)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *true*
Shipment Registered
with_id: *123*
Destination Specified
for_shipment: *123*
with_address: *Memory Lane 1*
Delivery Scheduled
for_shipment: *123*
for_date: ~D[2020-10-07]
Shipment Delivered Successfully
with_id: *123*

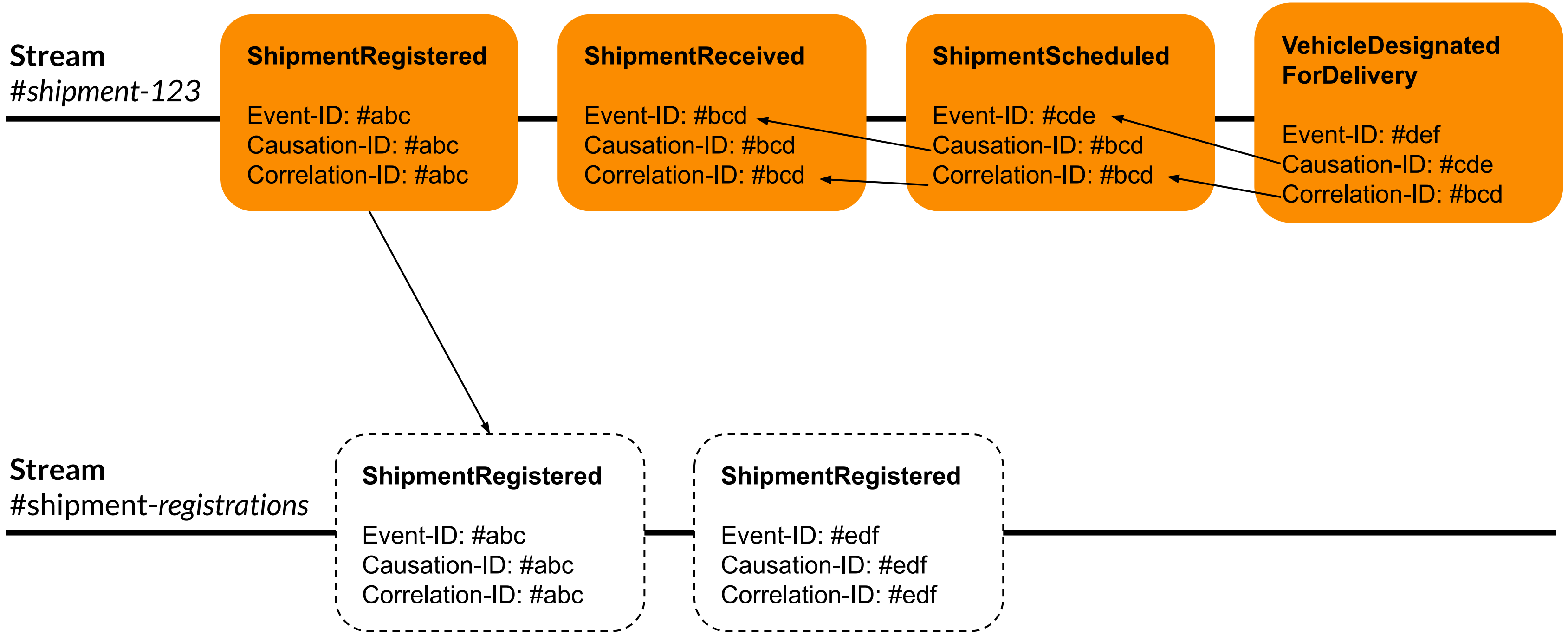

Shipment
id: UUID || String
destination: *Address*
delivery_date: *Date*
delivered?: *Bool*
create(123)
Shipment
id: 123
destination: *nil*
delivery_date: *nil*
delivered?: *false*
set_destination(
shipment,
“Memory Lane 1”
)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *nil*
delivered?: *false*
schedule(
shipment,
~D[2020-10-07]
)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *false*
set_delivered_successfully(
shipment
)
Shipment
id: 123
destination: *Memory Lane 1*
delivery_date: *~D[2020-10-07]*
delivered?: *true*
Shipment Registered
with_id: *123*
Destination Specified
for_shipment: *123*
with_address: *Memory Lane 1*
Delivery Scheduled
for_shipment: *123*
for_date: ~D[2020-10-07]
Shipment Delivered Successfully
with_id: *123*


Shipment
id: UUID || String
destination: Address
delivery_date: Date
delivered?: Bool
Shipment
id: 123
destination: Memory Lane 1
delivery_date: ~D[2020-10-07]
delivered?: true
set_delivered_successfully(shipment)
create(123)
Shipment
id: 123
destination: nil
delivery_date: nil
delivered?: false
set_destination(shipment, "Memory Lane 1")
Shipment
id: 123
destination: Memory Lane 1
delivery_date: nil
delivered?: false
schedule(shipment, ~D[2020-10-07])
Shipment
id: 123
destination: Memory Lane 1
delivery_date: ~D[2020-10-07]
delivered?: false
Shipment Registered
with_id: 123
Destination Specified
for_shipment: 123
with_address: Memory Lane 1
Delivery Scheduled
for_shipment: 123
for_date: ~D[2020-10-07]
Shipment Delivered Successfully
with_id: 123


Shipment
id: UUID || String
destination: Address
delivery_date: Date
delivered?: Bool
create(123)
Shipment
id: 123
destination: nil
delivery_date: nil
delivered?: false
set_destination(shipment, “Memory Lane 1”)
Shipment
id: 123
destination: Memory Lane 1
delivery_date: nil
delivered?: false
schedule(shipment, ~D[2020-10-07])
Shipment
id: 123
destination: Memory Lane 1
delivery_date: ~D[2020-10-07]
delivered?: false
set_delivered_successfully(shipment)
Shipment
id: 123
destination: Memory Lane 1
delivery_date: ~D[2020-10-07]
delivered?: true
Shipment Registered
with_id: 123
Destination Specified
for_shipment: 123
with_address: Memory Lane 1
Delivery Scheduled
for_shipment: 123
for_date: ~D[2020-10-07]
Shipment Delivered Successfully
with_id: 123

Streams & Events

Streams & Events



How to get started?
(Demo)

Pros & Cons

Pros & Cons of the Commanded EventStore

Pros

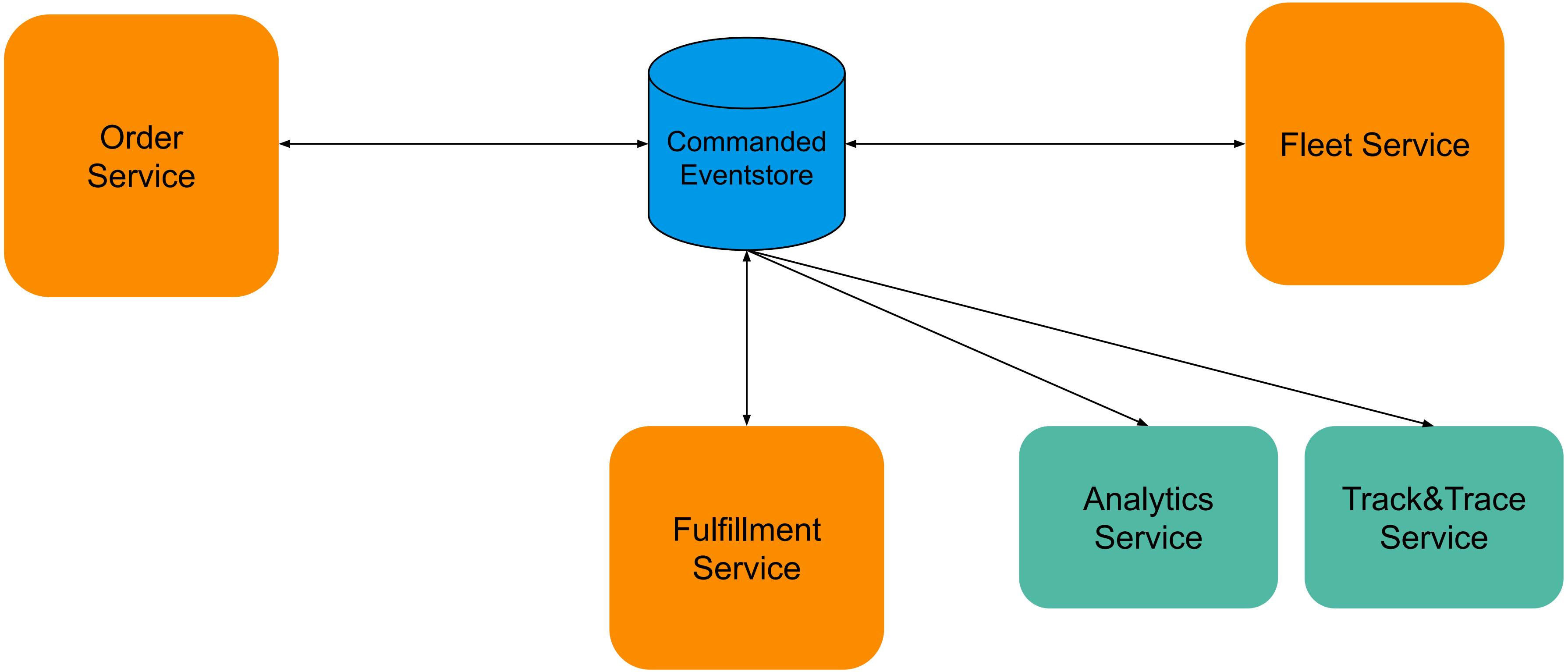
- (Relatively) easy to set up
- Not too hard to understand
- Good documentation

Cons

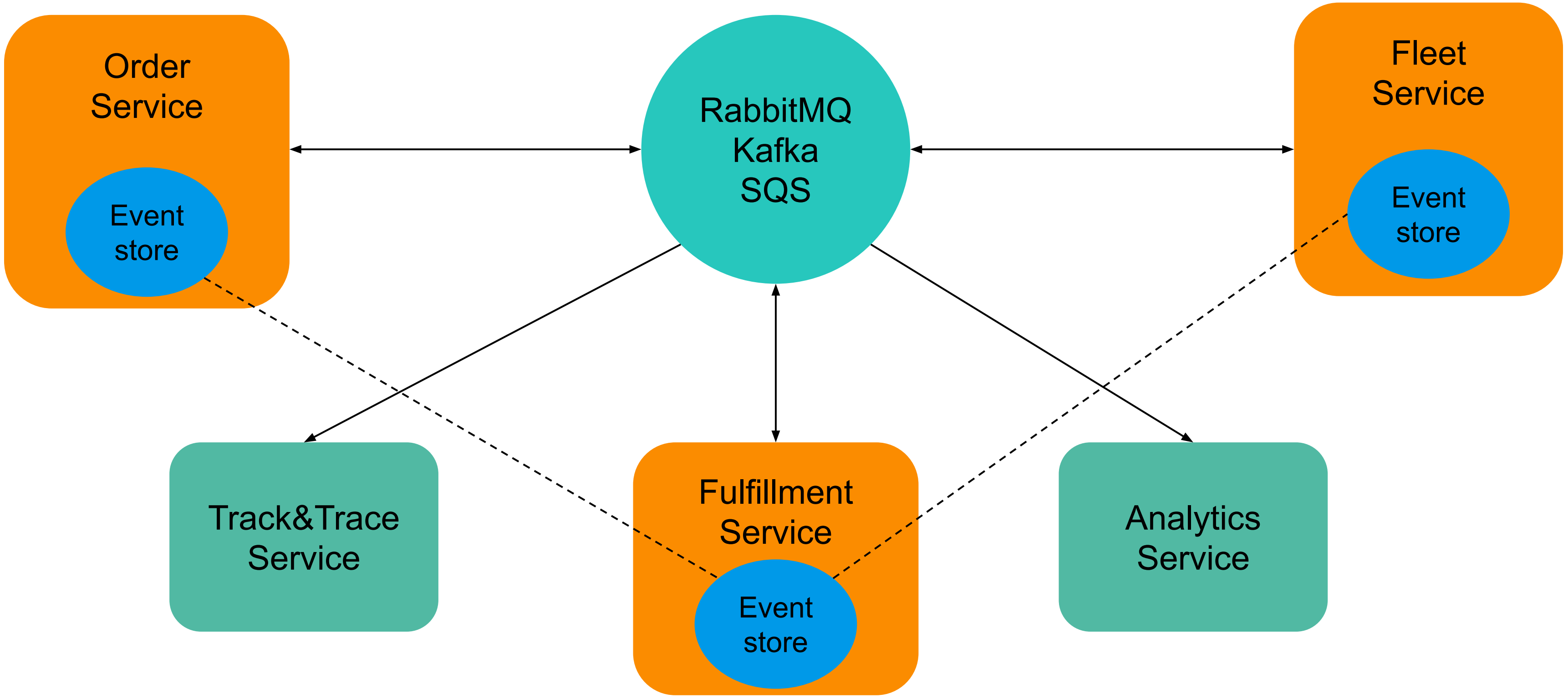
- No “batteries included”

  E.g. *No exponential Back-off*
- Heavy buy-in (Postgres Scheme)

An Example Shipping Architecture



Alternative Architecture



Thank you!

Code

https://github.com/PJUllrich/event-sourcing-with-elixir

Twitter & GitHub

@pjullrich

Resources

- Demo Code: https://github.com/PJUllrich/event-sourcing-with-elixir
- Event-Sourcing by Greg Young
- The Many Meanings of Event-Driven Architecture by Martin Fowler
- DDD, Event-Sourcing, and CQRS by Golo Roden
- Event-driven Architectures by Maciej Kaszubowski
- Event-sourcing Microservises by InfoQ
- Versioning in an Event Store by Greg Young