TRIFORK.

An introduction to CQRS and Axon Framework

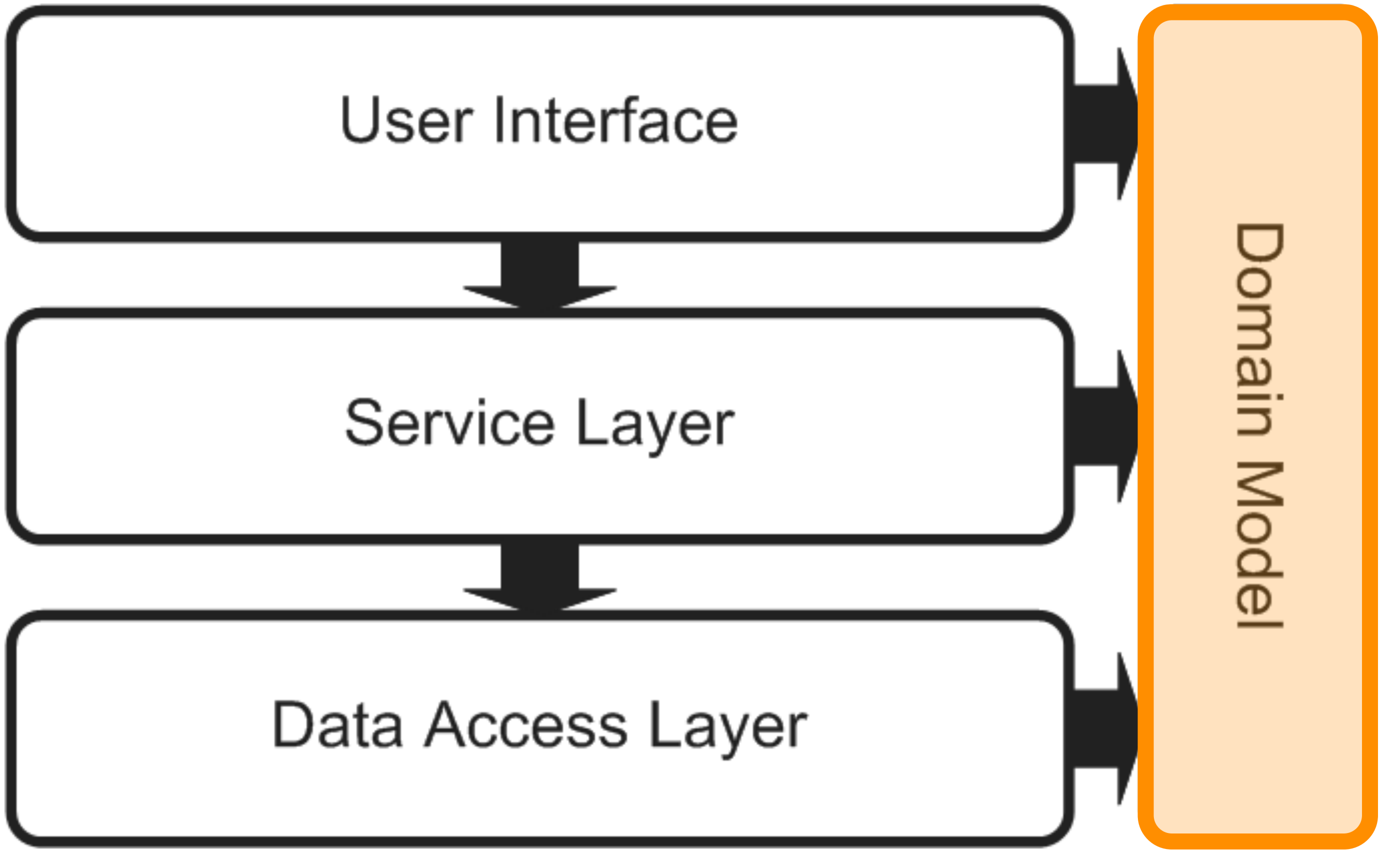
Finance's 'forgotten' treasure

Allard Buijze – allard.buijze@trifork.nl

Allard Buijze

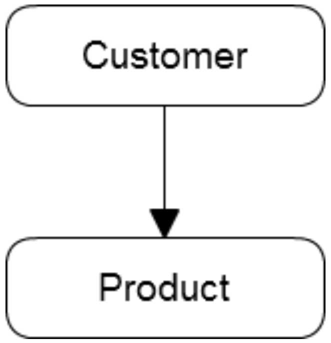
- Software Architect at Trifork Amsterdam
- ~ 15 years of web development experience
- Strong believer in DDD and CQRS
- Developer and initiator of Axon Framework
  - Java Framework for scalability and performance
  - www.axonframework.org

Layered architecture

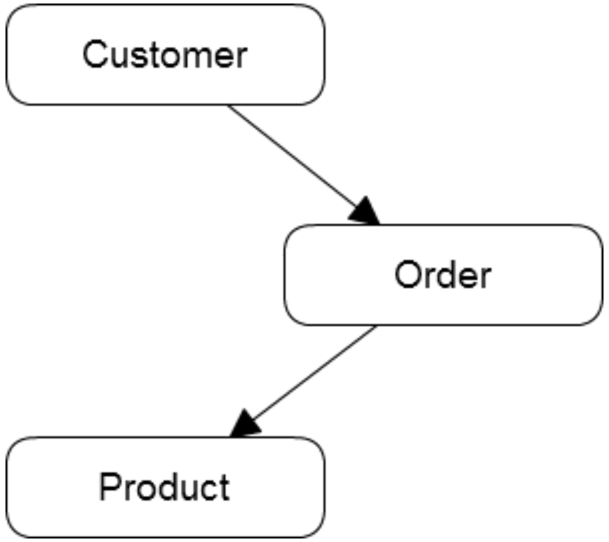


TRIFORK.

Evolution of a Domain Model

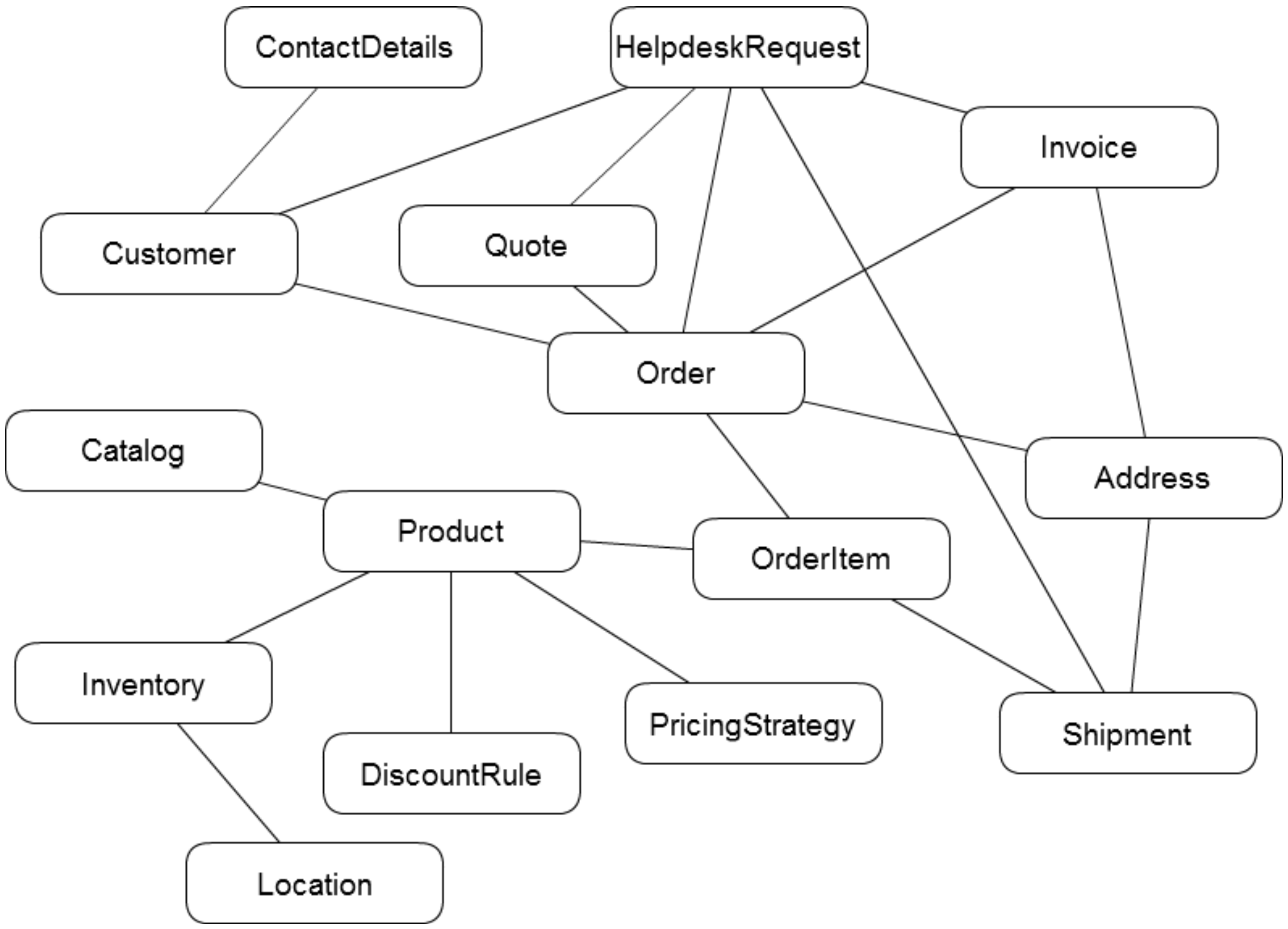


Evolution of a Domain Model



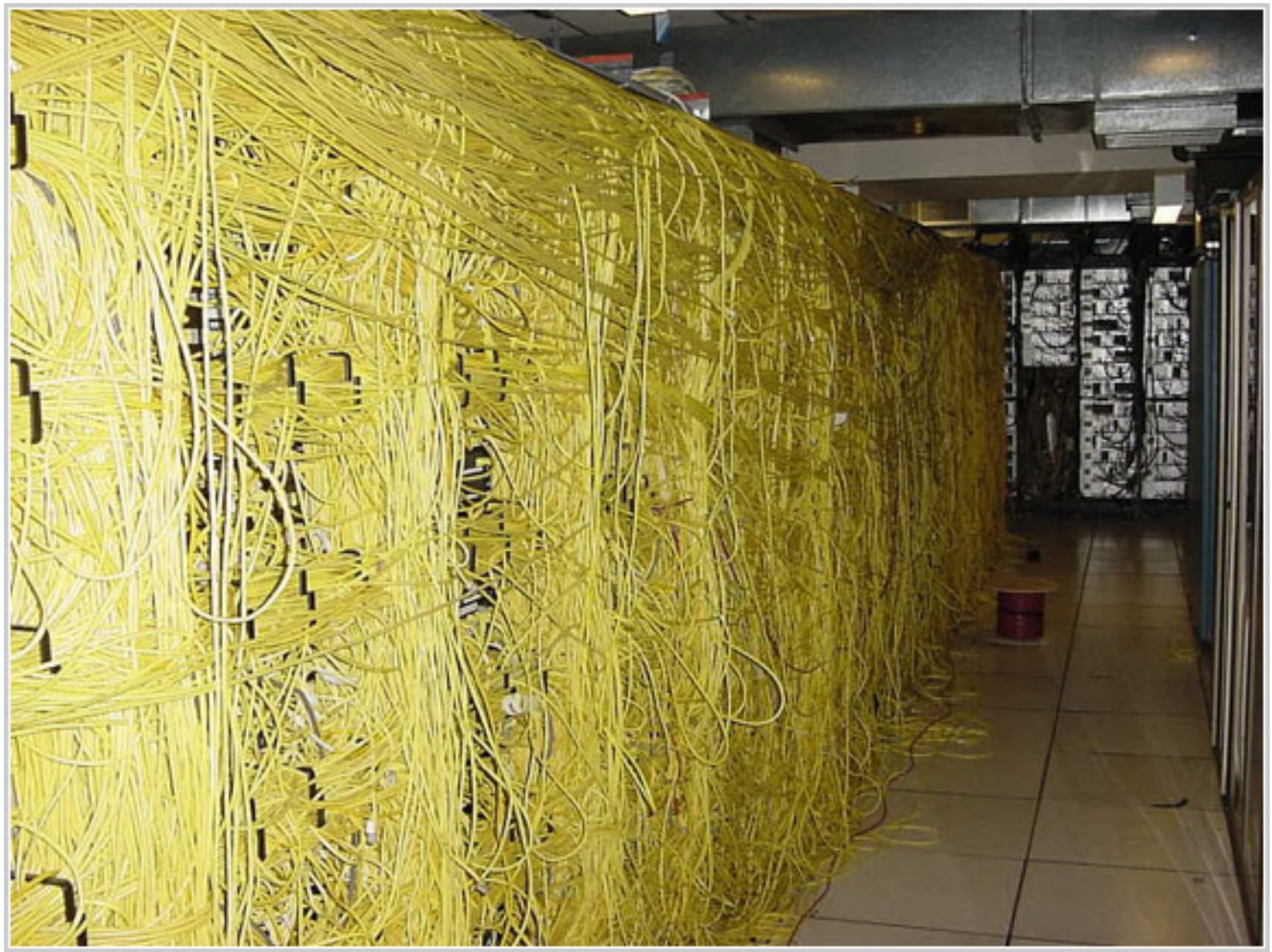
TRIFORK.

Evolution of a Domain Model



TRIFORK.

Evolution of complexity



Source: http://royal.pingdom.com/2008/01/09/the-worst-cable-mess-ever/

Evolution of complexity

```
private static final String PLAYER_COCKPIT_WATERFALL_ITEMS_QUERY =
        "(" +
          "select id, " + EntityType.NEWS_ITEM.ordinal() + " as entity_type, publish_date as sort_date " +
          "from news_item " +
          "where active = true and (" +
              "poster_player_id = :playerId " +
              "or poster_player_id in (" +
                "select destination_friend_id from friendship where origin_friend_id = :playerId " +
              ")" +
              "or project_id in (" +
                "select distinct project_id " +
                "from donation " +
                "where donor_participant_id = :playerId and status = 'OK'" +
              ")" +
              "or project_id in (" +
                "select project_id from ambassador_project where player_id = :playerId " +
              "))" +
        ") union all (" +
          "select id, " + EntityType.DONATION.ordinal() + " as entity_type, approval_date as sort_date " +
          "from donation " +
          "where status = 'OK' and (" +
              "donor_participant_id = :playerId " +
              "or donor_participant_id in (" +
                "select destination_friend_id from friendship where origin_friend_id = :playerId" +
              ")" +
              "or raised_via_player_id = :playerId " +
              "or raised_via_player_id in (" +
                "select destination_friend_id from friendship where origin_friend_id = :playerId" +
              ") " +
            ") " +
        ") union all ( " +
          "select id, " + EntityType.FRIENDSHIP.ordinal() + " as entity_type, created as sort_date " +
          "from friendship " +
          "where origin_friend_id = :playerId or (origin_friend_id in ( " +
            "select destination_friend_id from friendship where origin_friend_id = :playerId " +
          ") and destination_friend_id <> :playerId)" +
        ") ";
```

SELECT

NEWS_ITEM

or project in (…

status = ‘OK’

or project in (…

UNION ALL

DONATION

status = ‘OK’

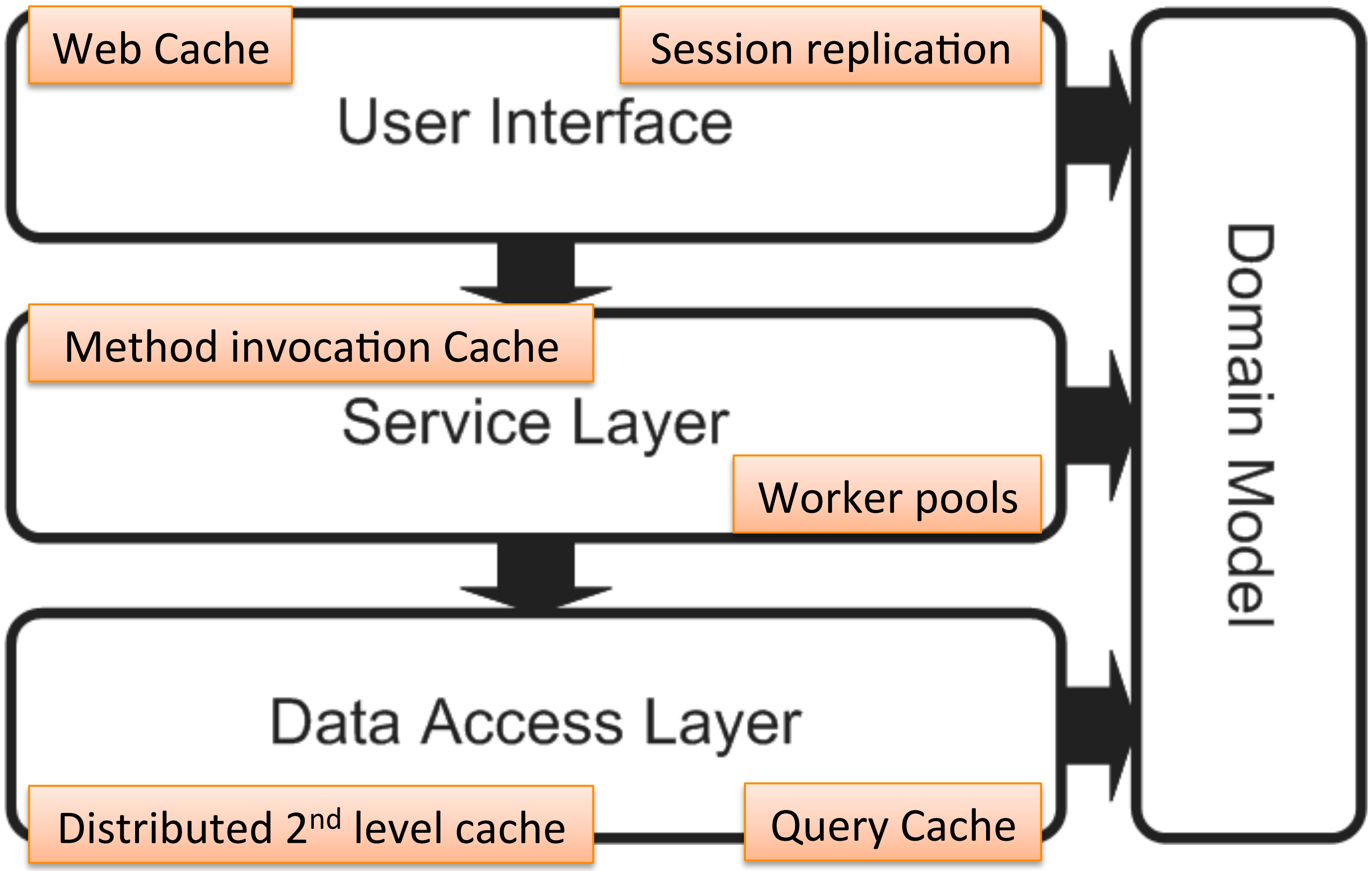
or raised_via_player in (…

UNION ALL

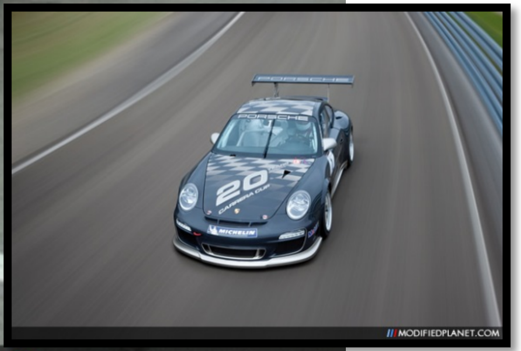
FRIENDSHIP

TRIFORK.

Layered architecture



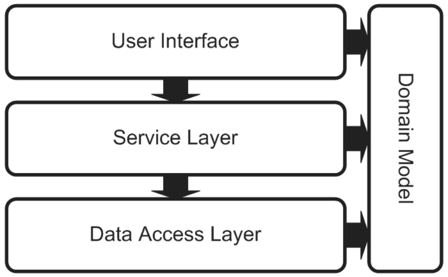
Designed for high performance (?)

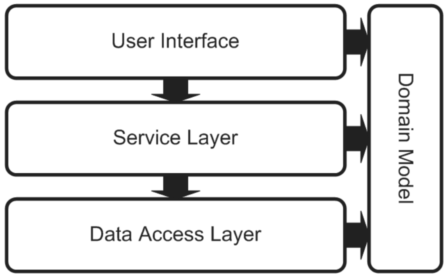


TRIFORK.

Then vs Now

1970's → 2014

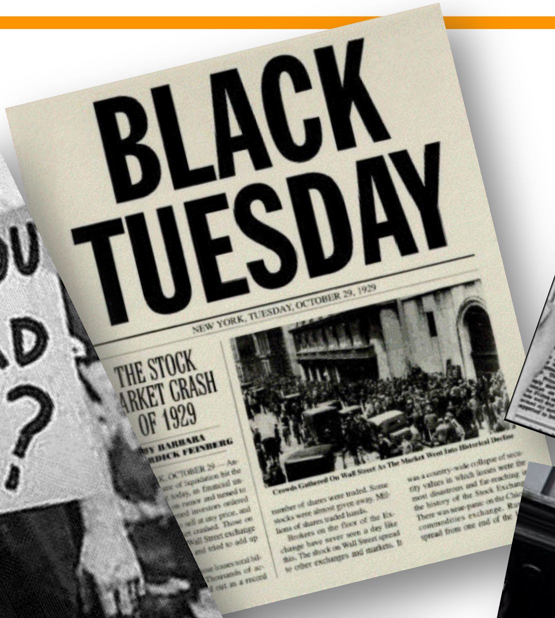




TRIFORK.

Brought to us by the Financial Sector

WHY CAN'T YOU GIVE MY DAD A JOB?
I KID ... N'T STARVE ... WHY SHOULD WE?


BLACK TUESDAY

NEW YORK, TUESDAY, OCTOBER 29, 1929

THE STOCK MARKET CRASH OF 1929

Crowds Gathered On Wall Street As The Market Went Into Historical Decline

BROOKLYN DAILY EAGLE

And Complete Long Island News

NEW YORK CITY, THURSDAY, OCTOBER 24, 1929.

32 PAGES

LATE NE

WALL ST. IN PANIC AS STOCKS CRASH

Attempt Made to Kill Italy's Crown Prince

ASSASSIN CAUGHT IN BRUSSELS MOB; PRINCE UNHURT

Hollywood Fire Destroys Films Worth Millions

FEAR 52 PERISHED IN LAKE MICHIGAN; FERRY IS MISSING

PIECE OF PLANE LIKE DITEMAN'S IS FOUND AT SEA

High Duty Group Gave $700,000 to Coolidge Drive

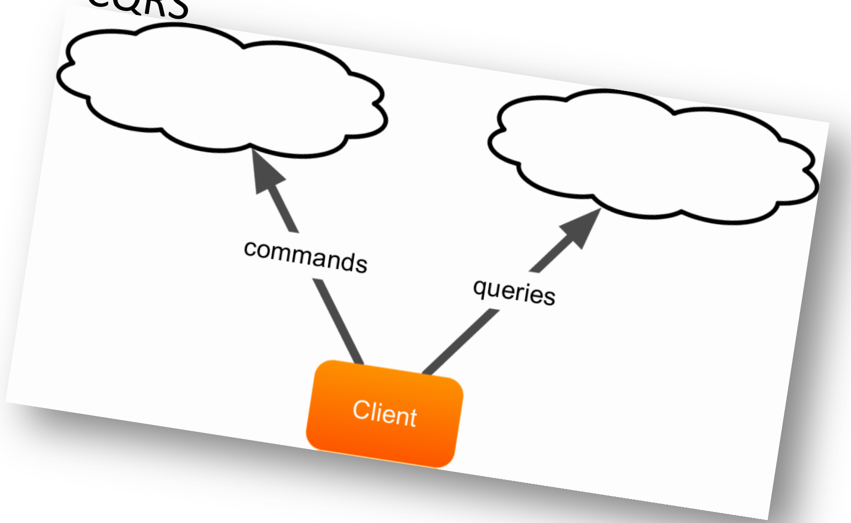
STOCKS CRASH IN RUSH TO SELL; BILLIONS LOST

TRIFORK.

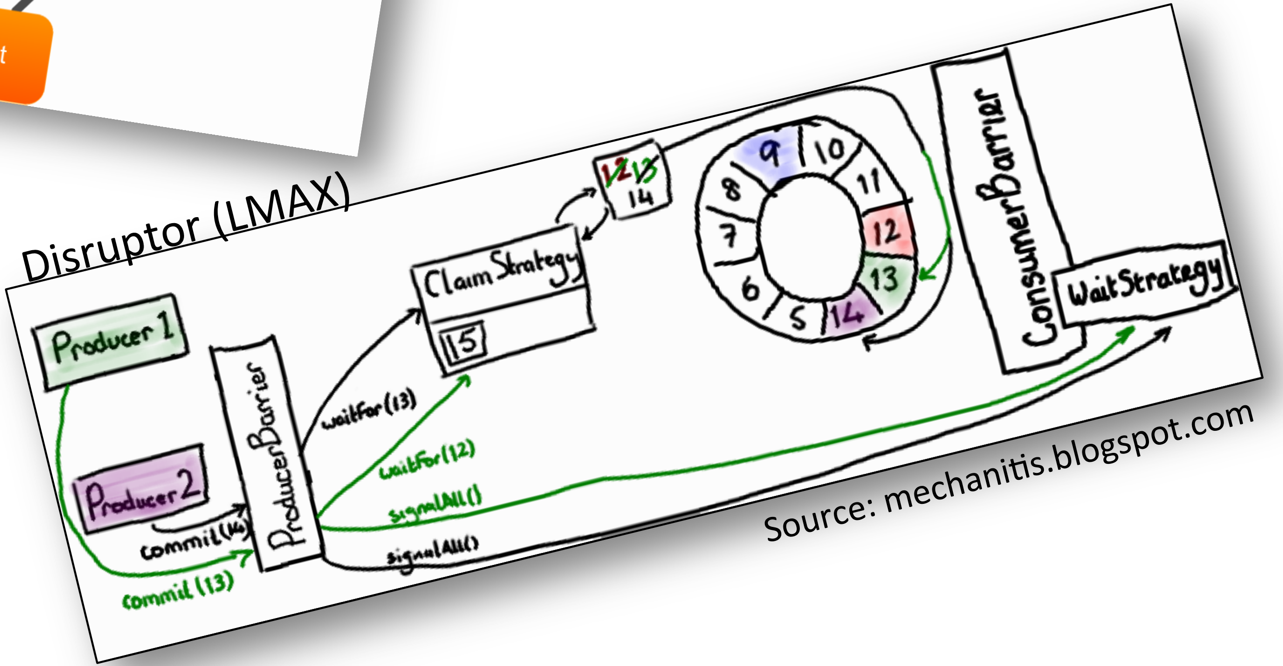
Sources (ltr ttb): guides.wikinut.com (2x), telegraph.co.uk, commons.wikimedia.org, usatoday.com

Brought to us by the Financial Sector

CQRS



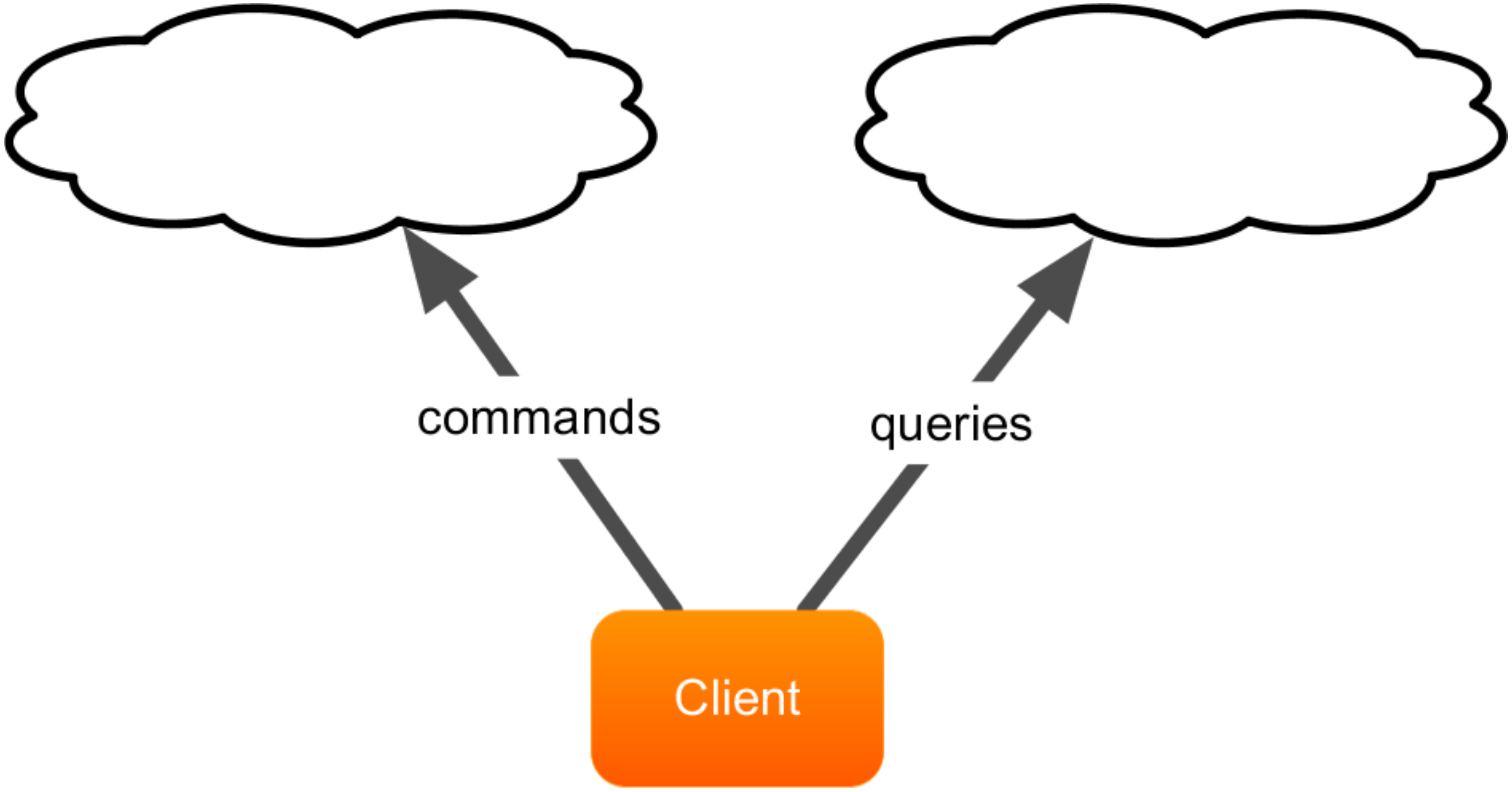
Disruptor (LMAX)



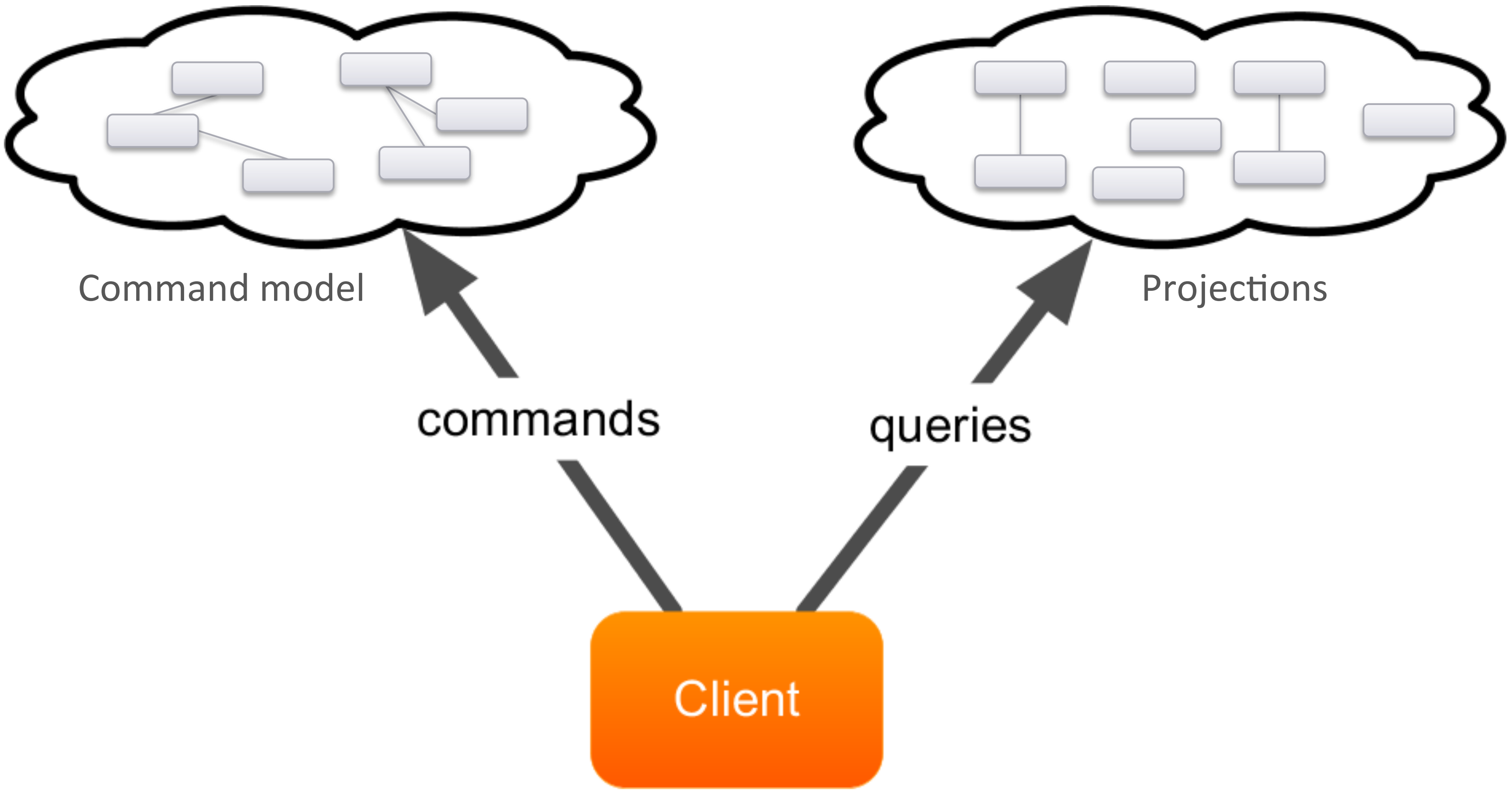
Source: mechanitis.blogspot.com

TRIFORK.

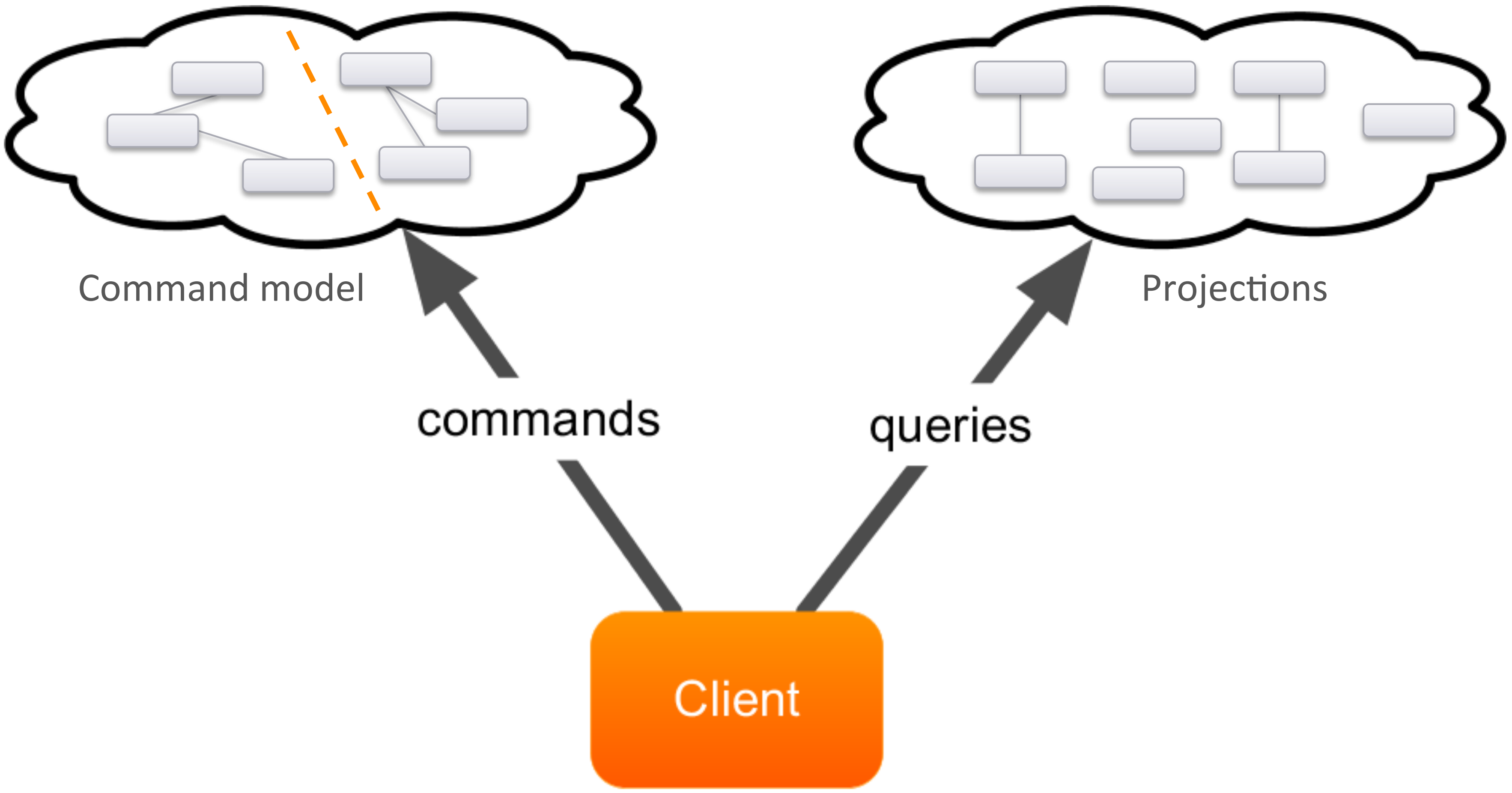
CQRS Based Architecture



CQRS Based Architecture

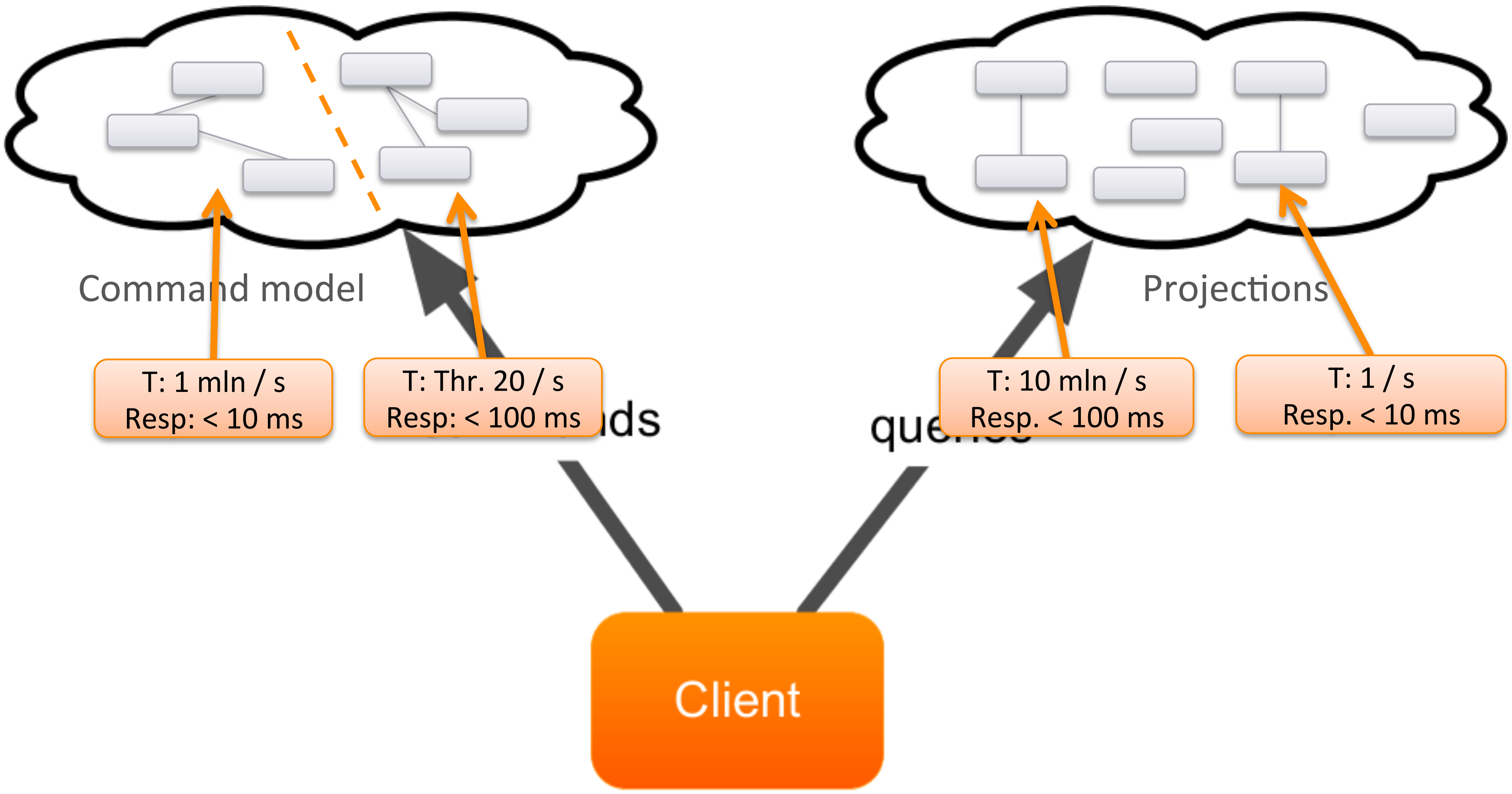


CQRS Based Architecture

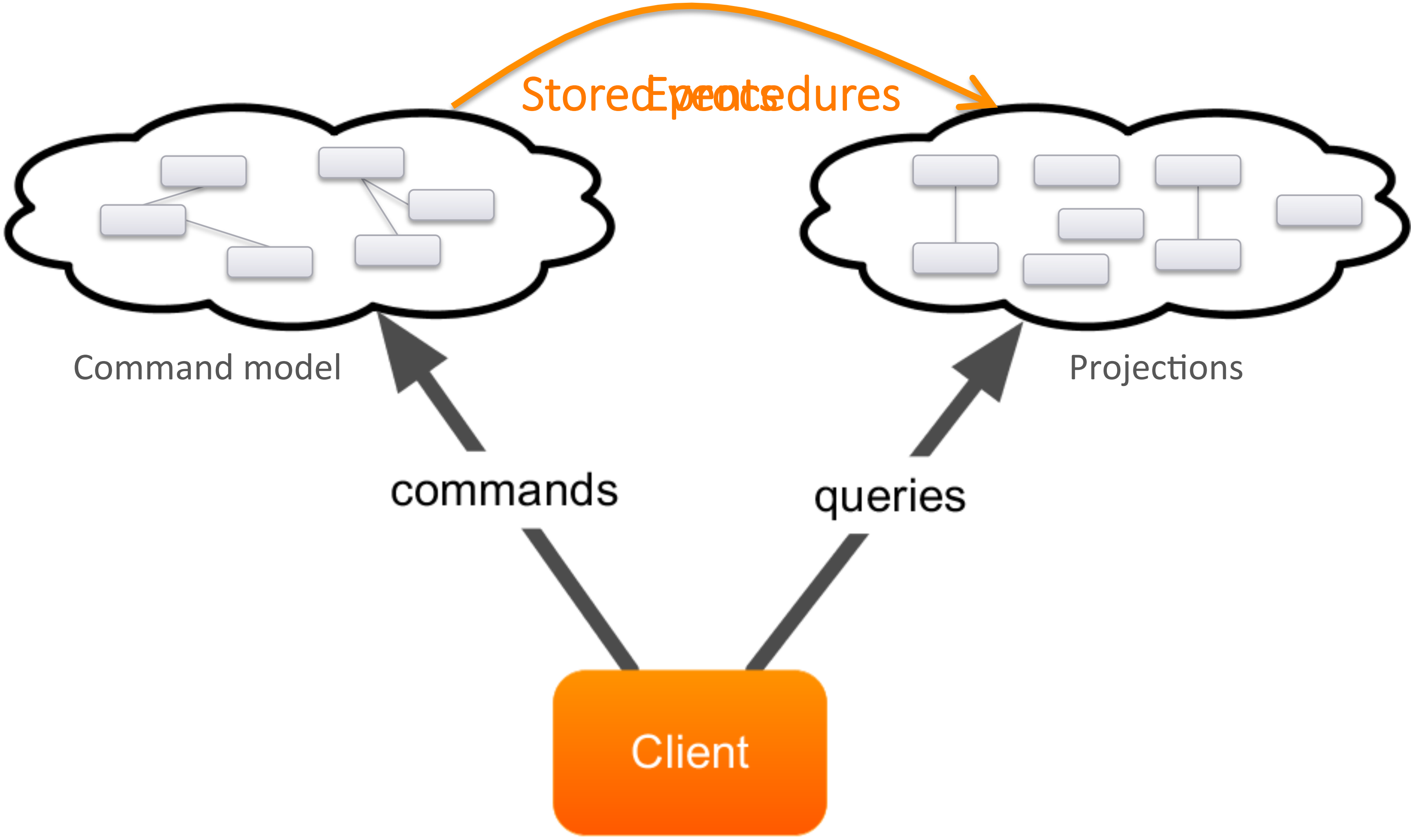


TRIFORK.

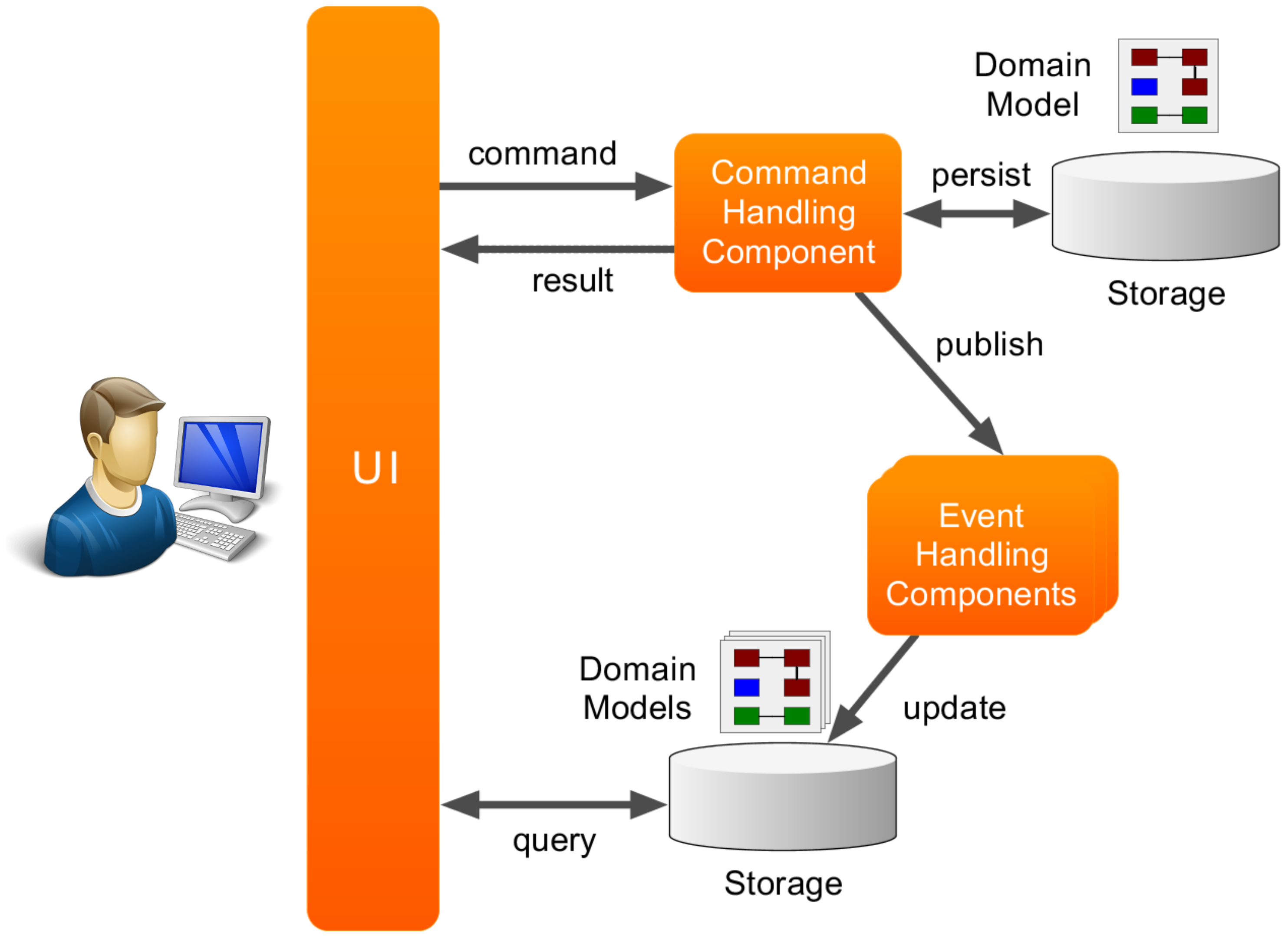
CQRS Based Architecture



Synchronizing models



CQRS Based Architecture



TRIFORK.

The power of ubiquitous events

Reactive

Push

Polling

Cache eviction

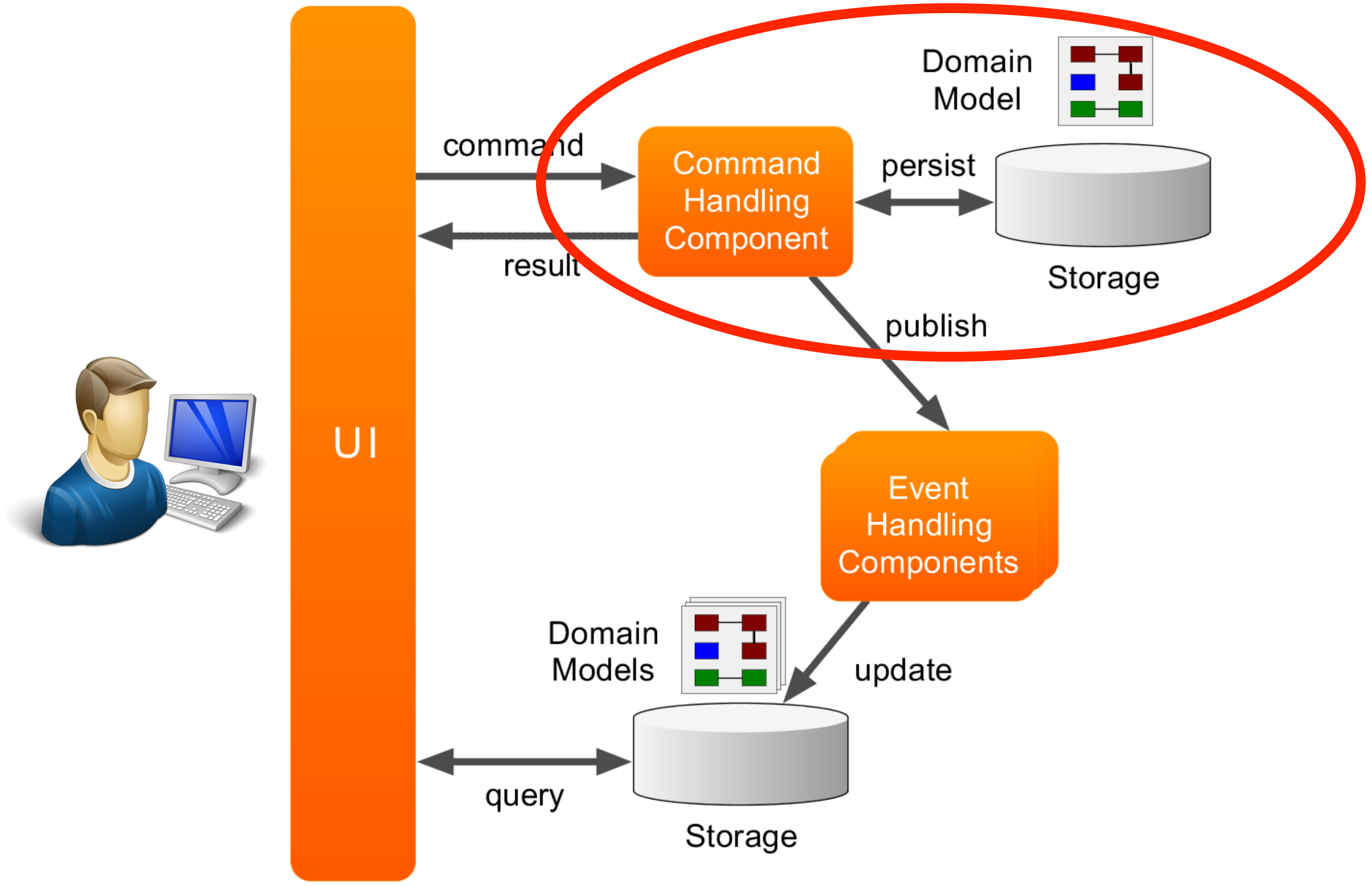
Real-time

Event Sourcing

Location transparency

Systems integration

Event Sourcing



TRIFORK.

Event Sourcing

Orders

ID	Status
1	Return shipment rcvd

OrderItems

ID	OrderID	Product	Count
1	1	Deluxe Chair	1
2	1	...	...

VS

Seq#	Event
0	OrderCreatedEvent
1	ItemAddedEvent (2x Deluxe Chair - € 399)
2	ItemRemovedEvent (1x Deluxe Chair - € 399)
3	OrderConfirmed
4	OrderCancelledByUserEvent
5	ReturnShipmentReceived

TRIFORK.

Event Sourcing

- Pros
  - Audit trail
  - Reconstruct query model(s)
  - Management reports since day 1
  - Data analysis

- Cons
  - Maintain history (upcasters)
  - Ever-growing

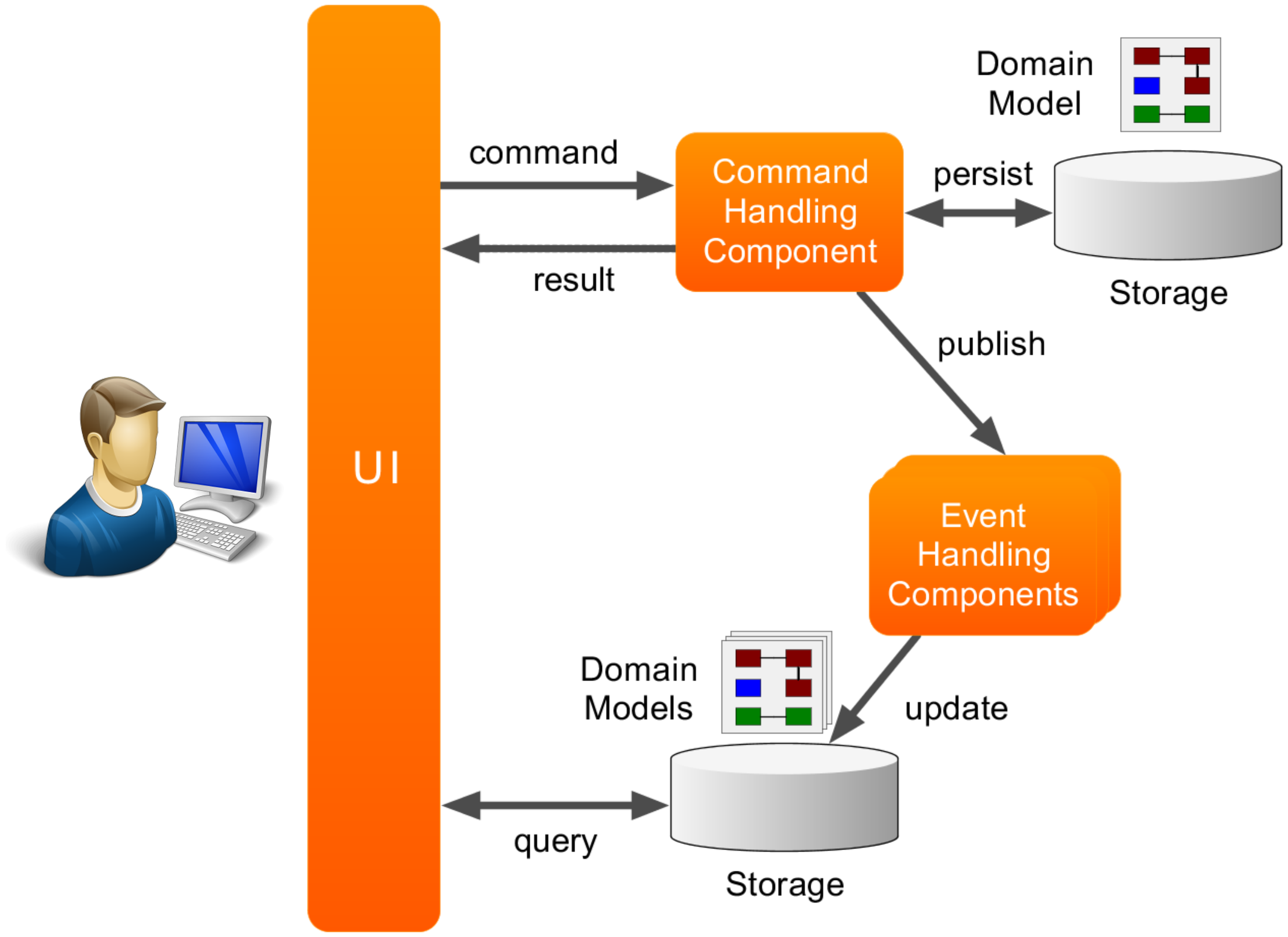
Axon Framework

- “CQRS Framework” for Java
  - Open source under Apache 2 License
- Simplify CQRS based applications
  - Provides building blocks for CQRS applications
- Current version*: 2.1
- More information: www.AxonFramework.org

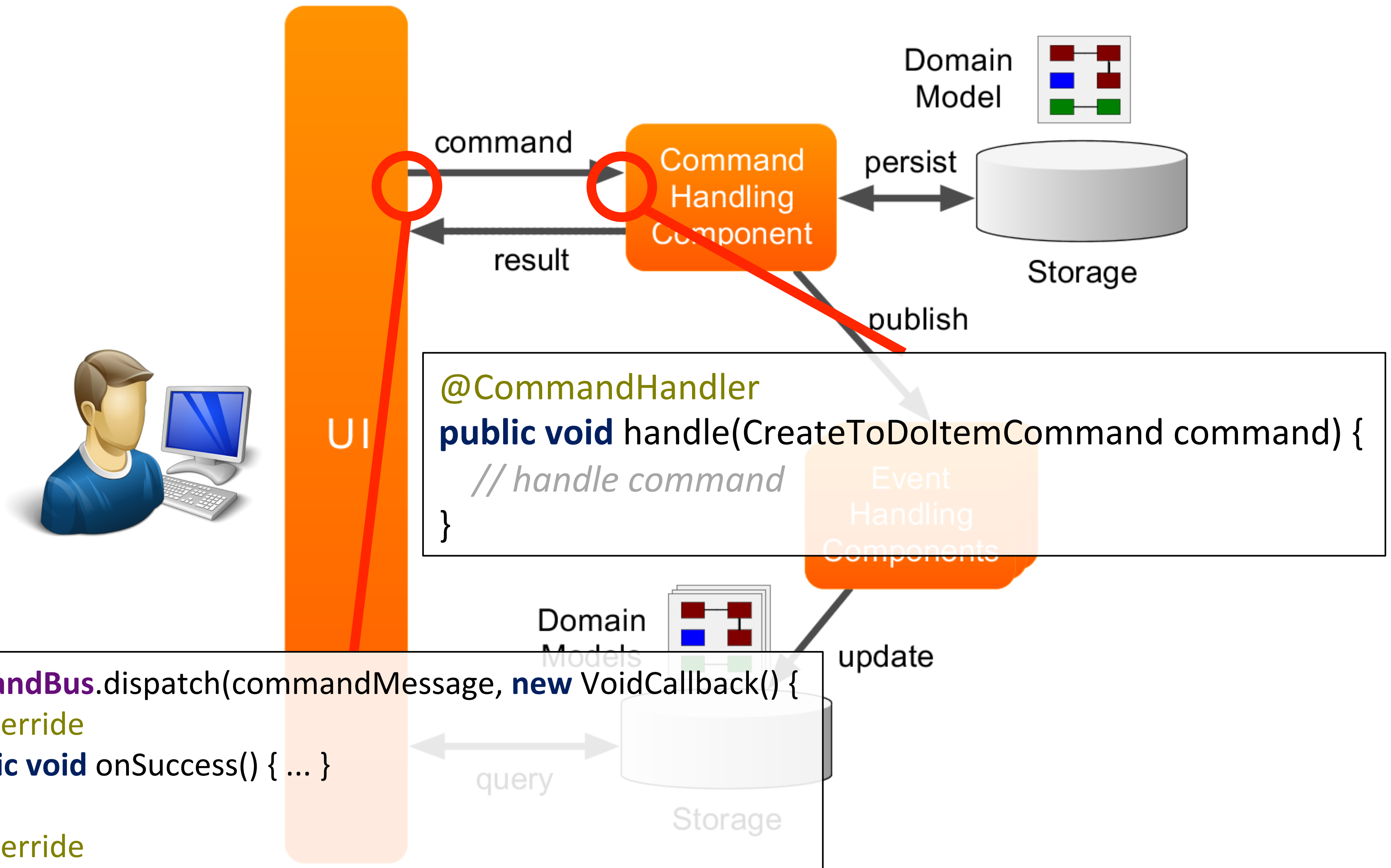
* On January 9th, 2014

TRIFORK.

CQRS Based Architecture



Axon – Command Bus API



```
@CommandHandler
public void handle(CreateToDoItemCommand command) {
   // handle command
}
```

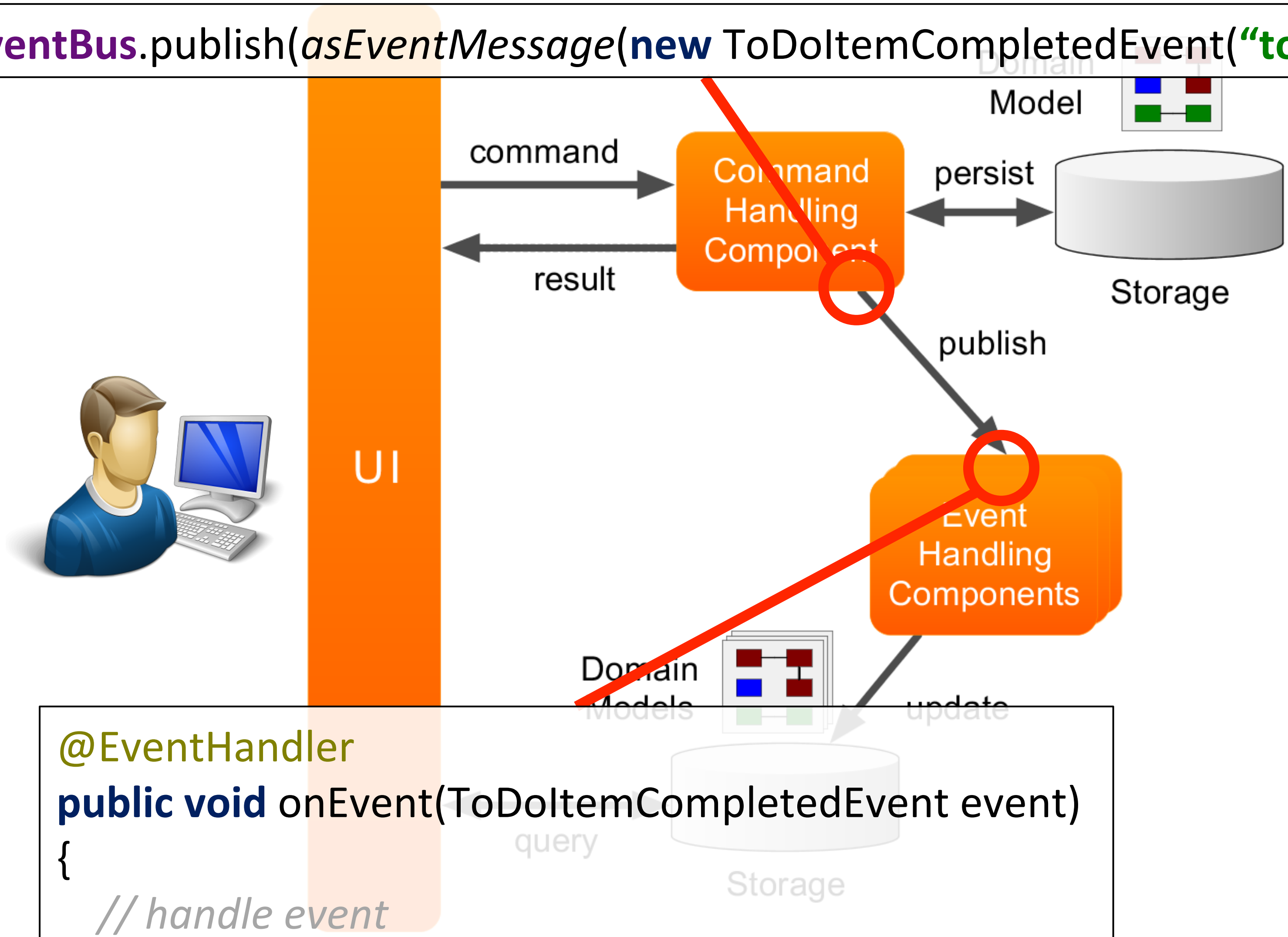
```
commandBus.dispatch(commandMessage, new VoidCallback() {
   @Override
   public void onSuccess() { ... }

   @Override
   public void onFailure(Throwable cause) { ... }
});
```

TRIFORK.

Axon – Event Bus API

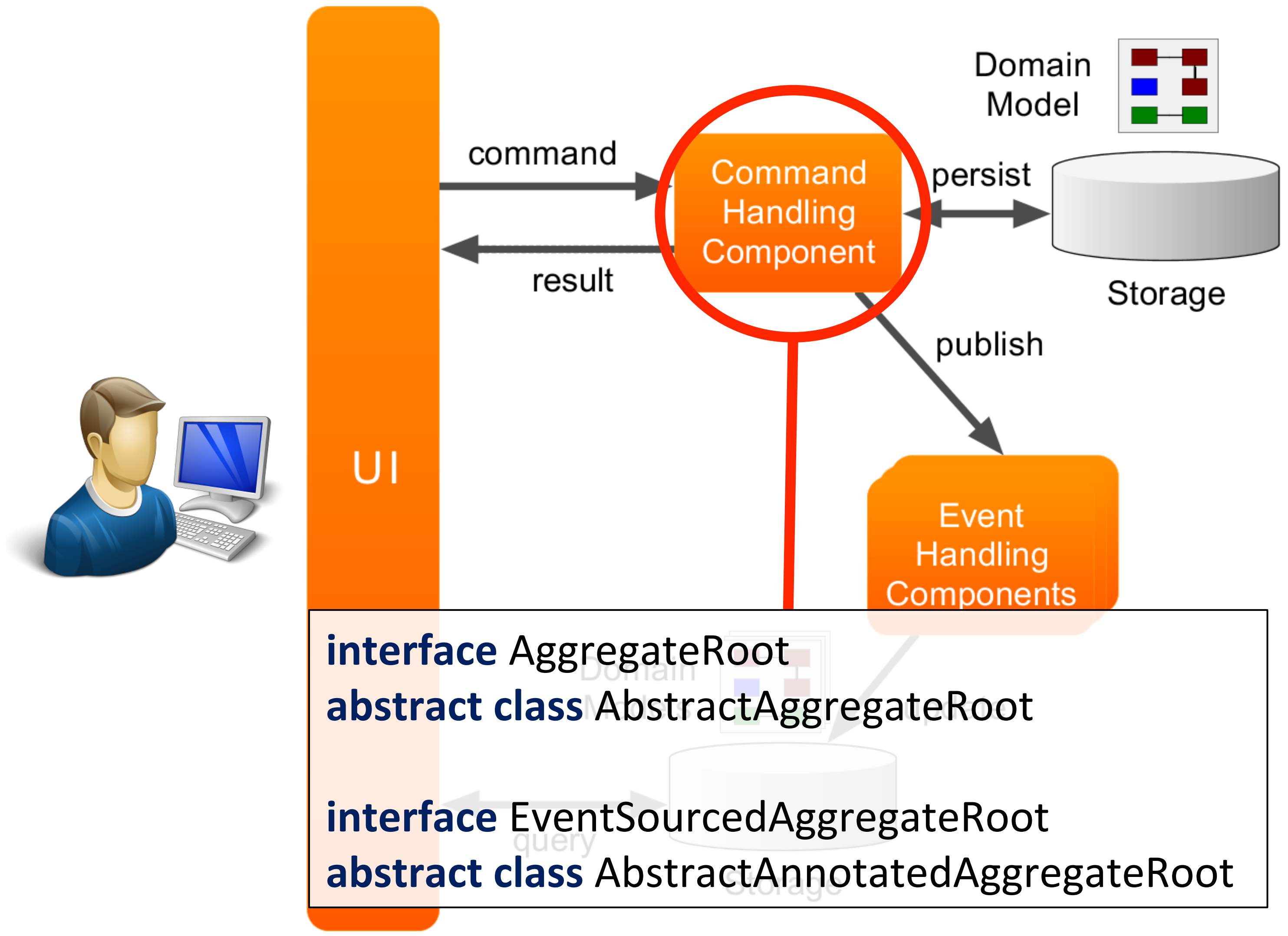
```
eventBus.publish(asEventMessage(new ToDoItemCompletedEvent("todo1")));
```



```
@EventHandler
public void onEvent(ToDoItemCompletedEvent event)
{
    // handle event
}
```

TRIFORK.

CQRS Based Architecture

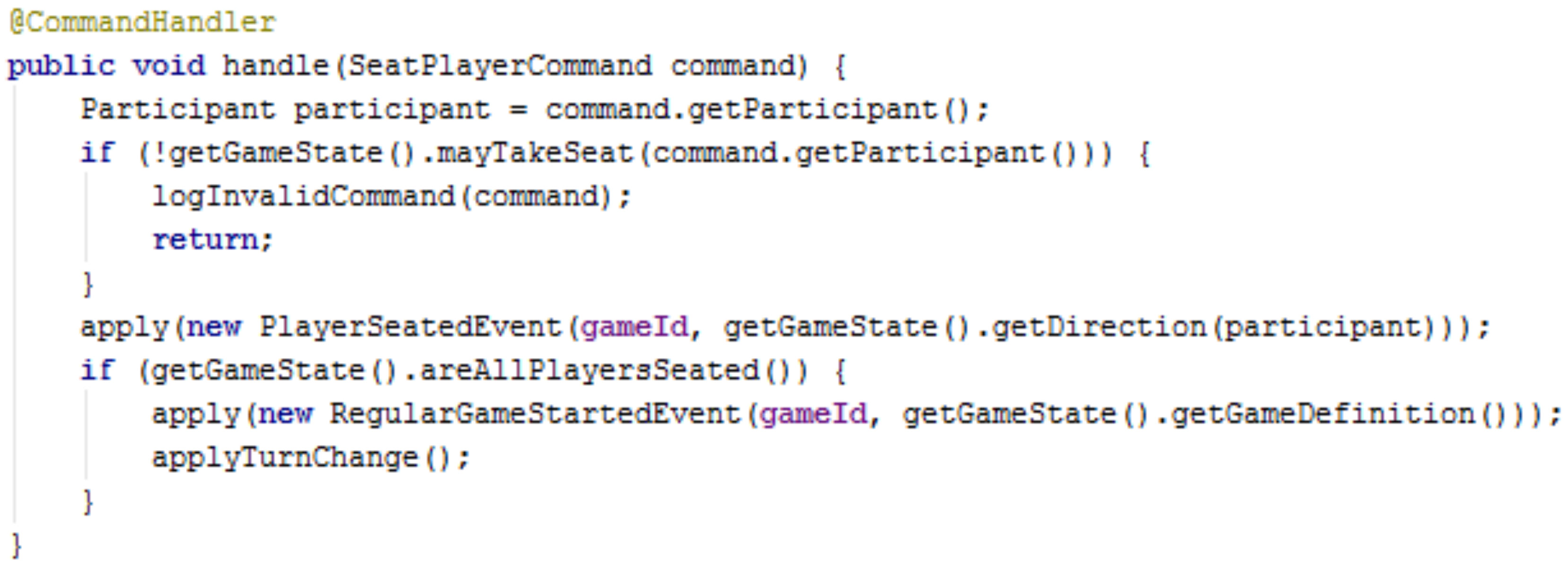


interface AggregateRoot
abstract class AbstractAggregateRoot

interface EventSourcedAggregateRoot
abstract class AbstractAnnotatedAggregateRoot

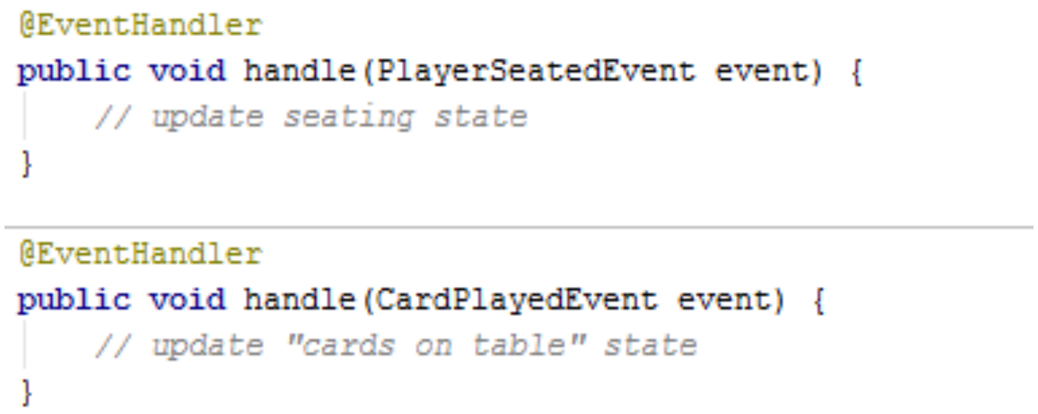
Axon – Event Sourcing

Make decisions



```
@CommandHandler
public void handle(SeatPlayerCommand command) {
    Participant participant = command.getParticipant();
    if (!getGameState().mayTakeSeat(command.getParticipant())) {
        logInvalidCommand(command);
        return;
    }
    apply(new PlayerSeatedEvent(gameId, getGameState().getDirection(participant)));
    if (getGameState().areAllPlayersSeated()) {
        apply(new RegularGameStartedEvent(gameId, getGameState().getGameDefinition()));
        applyTurnChange();
    }
}
```

Apply state



```
@EventHandler
public void handle(PlayerSeatedEvent event) {
    // update seating state
}
```

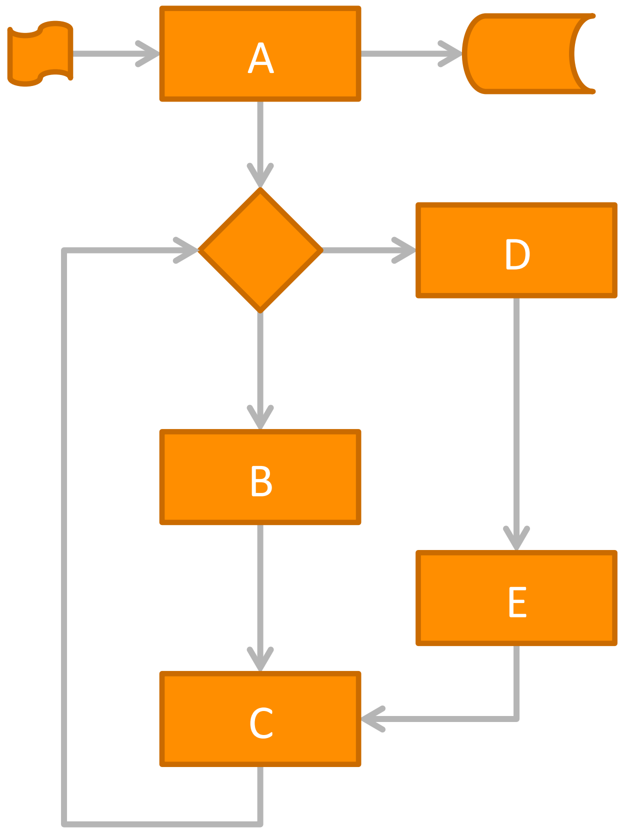
```
@EventHandler
public void handle(CardPlayedEvent event) {
    // update "cards on table" state
}
```

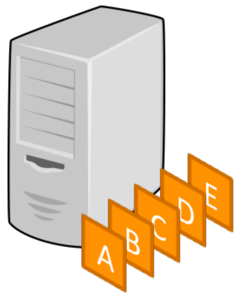
Event Sourcing - Testing

- Given-when-then fixtures
  - Given some past events
  - When I apply a new Command
  - Expect these new Events

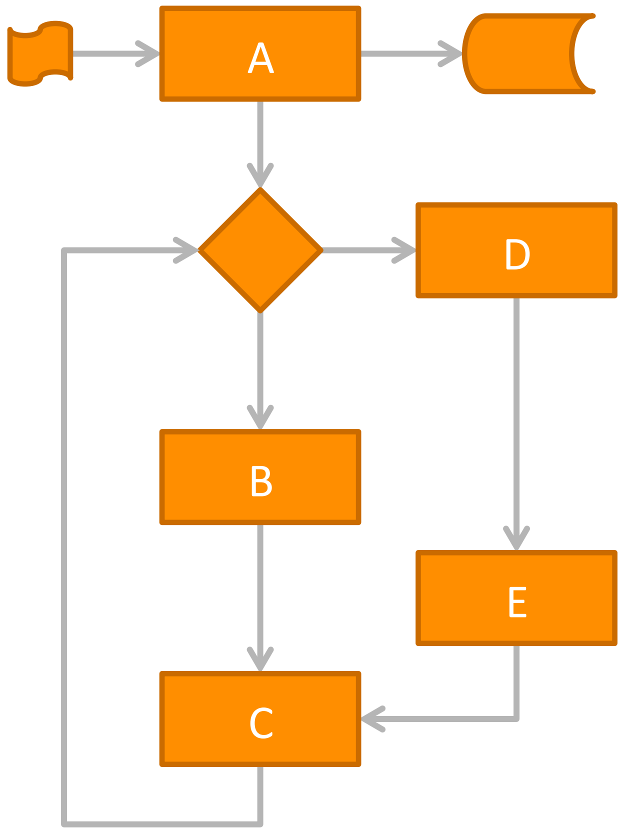
```
fixture.given(new GameStartedEvent(…),
              new CallMadeEvent(…),
              new TurnChangedEvent(…))
       .when(new MakeCallCommand(…))
       .expectEvents(new CallMadeEvent(…),
                     new TurnChangedEvent(…));
```

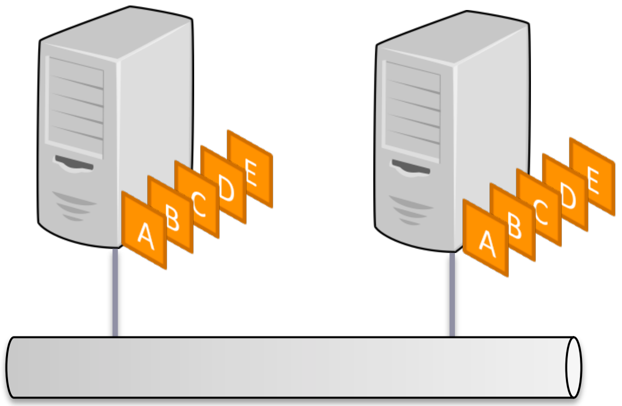
Separate infrastructure from business logic





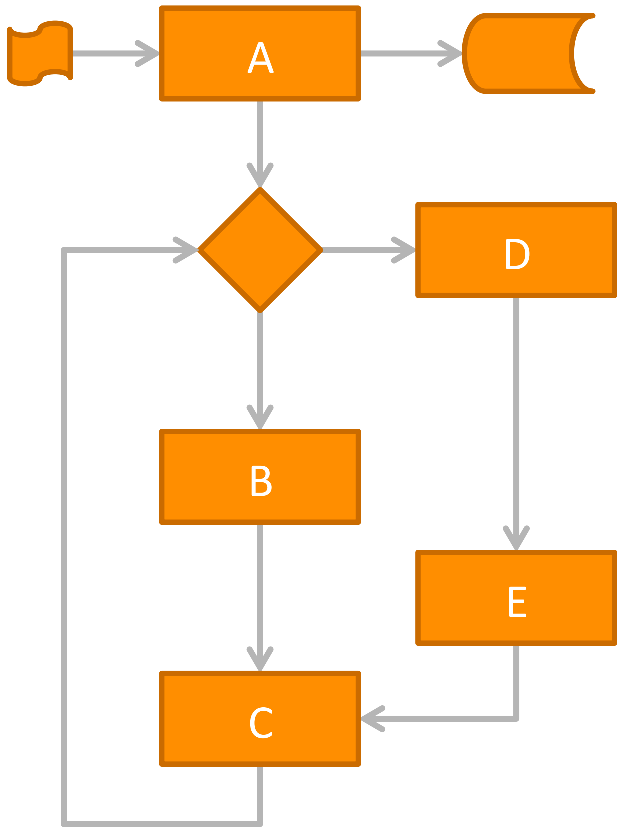
Separate infrastructure from business logic

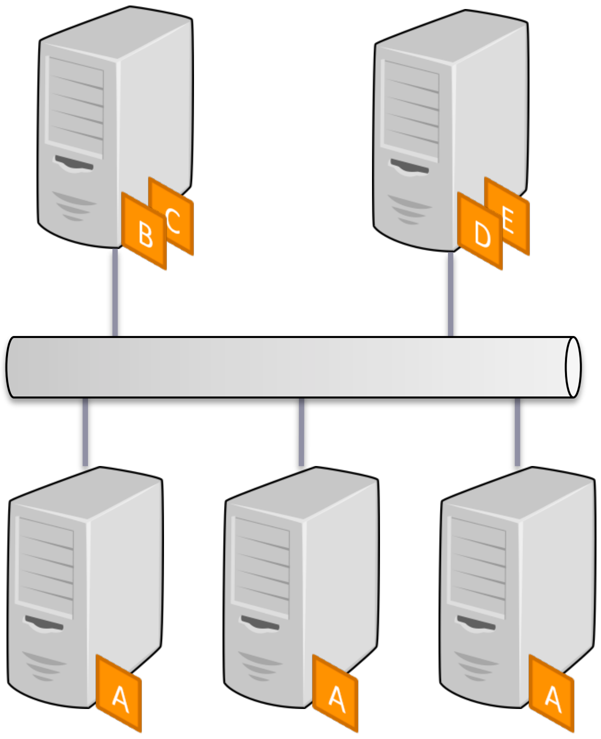




TRIFORK.

Separate infrastructure from business logic





Spring configuration - Simple

```
<axon:event-bus id="eventBus"/>
```

```
<axon:command-bus id="commandBus"/>
```

Spring configuration – High performance

```
<axon:event-bus id="eventBus"/>

<axon:disruptor-command-bus id="commandBus" event-store="eventStore"
                            event-bus="eventBus"
                            transaction-manager="transactionManager">
    <axon:repositories>
        <axon:repository id="gameRepository"
                         aggregate-type="some.sample.engine.game.RegularGame"/>
    </axon:repositories>
</axon:disruptor-command-bus>
```

Spring configuration – Distributed Events

```
<axon:event-bus id="eventBus" terminal="terminal"/>

<axon-amqp:terminal id="terminal" connection-factory="amqpConnection"
                    exchange-name="AxonEventBusExchange">
    <axon-amqp:default-configuration transaction-manager="transactionManager"
                                     transaction-size="25" prefetch="200"
                                     error-handler="loggingErrorHandler"/>
</axon-amqp:terminal>
```

```
<axon:cluster id="gameCluster" order="0" default="true">
    <axon:meta-data>
        <entry key="AMQP.Config">
            <bean class="org.axonframework...SpringAMQPConsumerConfiguration">
                <property name="queueName" value="GameEngineEvents"/>
            </bean>
        </entry>
    </axon:meta-data>
</axon:cluster>
```

TRIFORK.

Spring configuration – Distributed Commands

```
<bean id="commandBus" class="org.axonframework...DistributedCommandBus">
    <constructor-arg ref="jgroupsConnector"/>
</bean>

<bean id="jgroupsConnector"
      class="org.axonframework.commandhandling...JGroupsConnectorFactoryBean">
    <property name="serializer" ref="serializer"/>
    <property name="loadFactor" value="${loadFactor:100}"/>
    <property name="localSegment" ref="localCommandBus"/>
    <property name="configuration" value="tcp_gossip.xml"/>
</bean>


<axon:disruptor-command-bus id="localCommandBus" event-store="eventStore"
                            event-bus="eventBus"
                            transaction-manager="transactionManager">
    <axon:repositories>
        <axon:repository id="gameRepository"
                         aggregate-type="some.sample.engine.game.RegularGame"/>
    </axon:repositories>
</axon:disruptor-command-bus>
```

TRIFORK.

Infrastructure components in Axon

- Single VM
  - SimpleCommandBus
  - SimpleEventBus
- High Performance
  - DisruptorCommandBus
  - ...
- Distributed
  - DistributedCommandBus + JGroupsConnector
  - ClusteringEventBus + AMQP Terminal
  - ...

Axon Roadmap

- More distributed implementations
- Improved OSGi support

- DSL for definition of Command & Events
- IDE Plugins

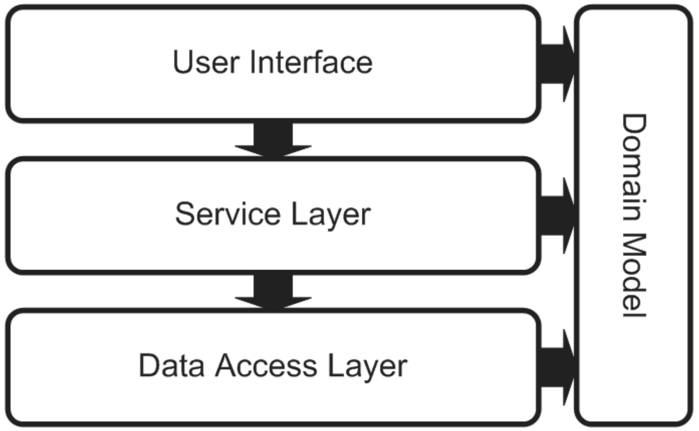
- High performance Event Store

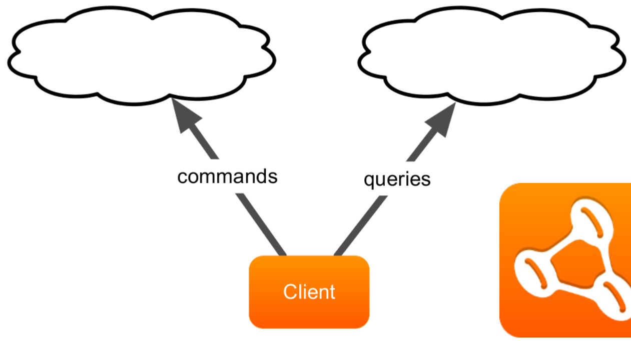
Axon Framework – Some cases

- Finance
  - Process automation in a top 50 bank
  - Trading engine for ETF (index trackers) trading
  - Pension fund calculations at a large bank
  - On-line payment processing
- Gaming
  - On-line bridge platform (bridgebig.com)
  - On-line casino (casumo.com)
- Healthcare
  - Electronic Medical Record for the Geriatric Healthcare
  - Tracking and Tracing of equipment for dental implants
- Aviation
  - Optimizing aircraft movement at a large European airport

TRIFORK.

The next time…





More information: axonframework.org

Allard Buijze

abu@trifork.com

TRIFORK.

QCon

Please evaluate my talk via the mobile app!