Backend System

WS Connection - RealTime Data

HTTP REQ

HTTP RESP

ANGULAR

API Gateway
with Load Balancer

Instance 1

Instance 2

Instance 3

Instance n

Redis Cache

RDBMS
MS SQL M

—————— Encrypted

HTTP REQ

HTTP RESP

External Data
Synchronization

HTTP REQ

HTTP RESP

BURSA
DE VALORI
BUCUREȘTI

Backend System - Monolith WebApp

Logging

| API | RT Comms | Orders Management System | Account Management | Company Inventory |

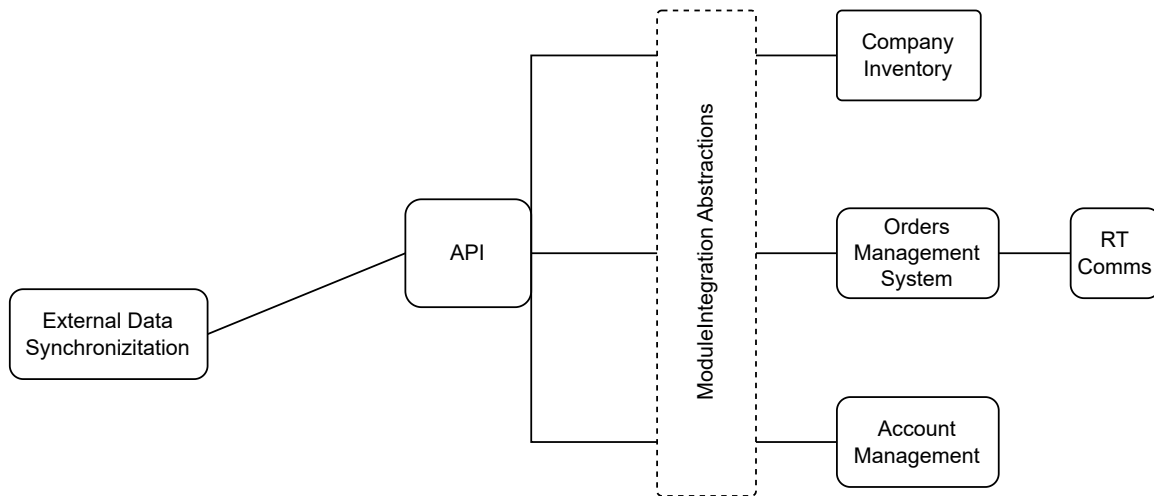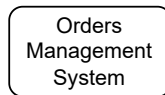Logging
External Data Synchronization

Impelementation Options:
* Console Application,
* WebApp with background processor
* Azure Function
For simplicity in the first phase of the development
process we will use a standalone
ConsoleApplication. The ideal situation for
production environment would be to have a
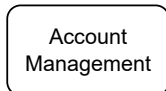scheduled Azure Function

```
                                                  ┌──────────────┐
                                                  │   Company    │
                                                  │  Inventory   │
                                                  └──────────────┘
          ┌──────────────┐          ┌─────┐ ┌──────────────────────┐  ┌──────────────┐  ┌─────────┐
          │ External Data│          │ API │ │                      │  │    Orders    │  │   RT    │
          │Synchronizitation│       │     │ │ ModuleIntegration    │  │  Management  │  │ Comms   │
          └──────────────┘          └─────┘ │   Abstractions       │  │    System    │  └─────────┘
                                            │                      │  └──────────────┘
                                            └──────────────────────┘  ┌──────────────┐
                                                                      │   Account    │
                                                                      │  Management  │
                                                                      └──────────────┘
```
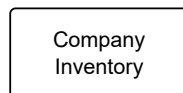
┌─────────┐   ┌──────────────┐   ┌─────────┐
│   API   │   │   Orders   │   │   RT   │
│   │   │ Management │   │ Comms │
└─────────┘   │   System   │   └─────────┘
                               └──────────────┘
Endpoint Hadling   manage connected user sessions
Check Authorization   Keep Order Book   * transmit updated data received
                                                                   from the order management system
                               Manage Orders
                                 * register new orders
                                 * cancel orders
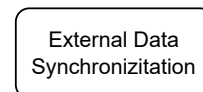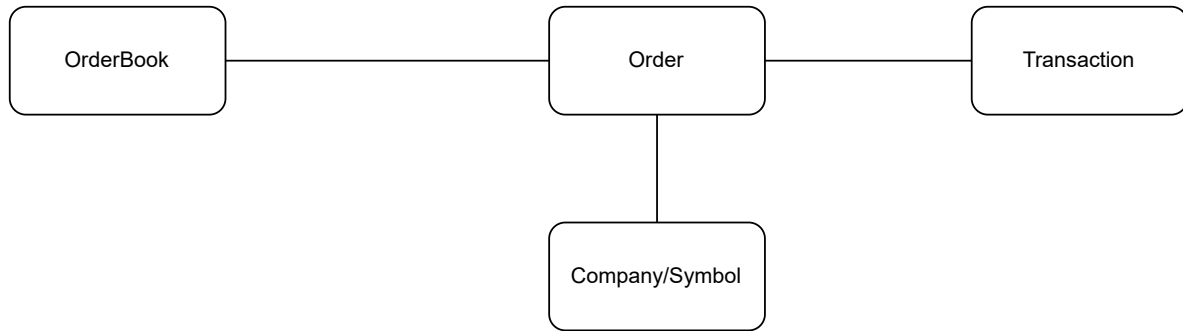   * match orders

┌──────────────┐   ┌──────────────┐   ┌──────────────┐
│   Account   │   │   Company   │   │ External Data│
│ Management │   │ Inventory   │   │Synchronizitation│
└──────────────┘   └──────────────┘   └──────────────┘

 * Authenticate   * keeps the company   * connect to external system
 * Track user trading activity   associated information and   to synchronize market data
   related infromation   historical data
                                                                      * transmit the refreshed
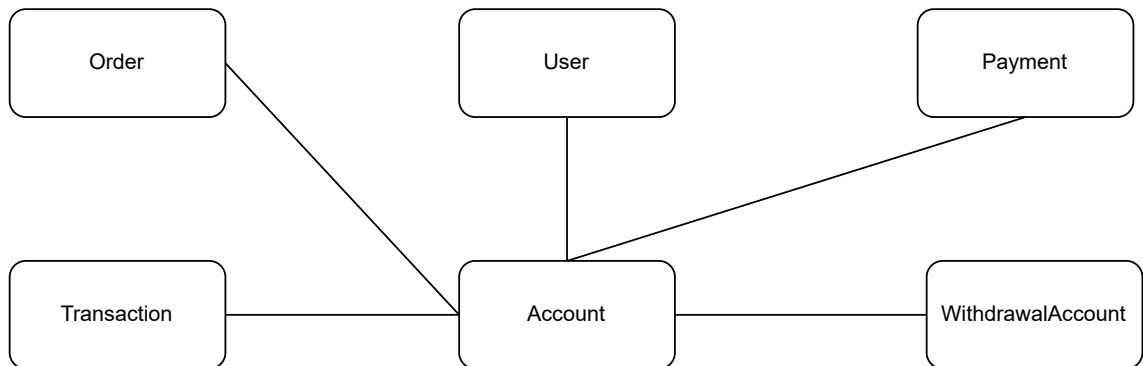 * funds management   market data to the order
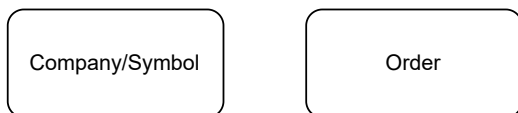   management system

Order Management System

```
OrderBook ———— Order ———— Transaction
                 |
            Company/Symbol
```

Account Management

```
Order            User          Payment
    \             |            /
     \            |           /
Transaction ———— Account ———— WithdrawalAccount
```

External Data Syncthornization

```
Company/Symbol        Order
```

Approach

Vertical Slices Architecture

CQRS - Command and Query Request Segregation

Single Page Application on the frontend