Nick Dickson, Christopher Bradshaw, Dustin Watkins
CS 462 - Final Project

Fast Flower Delivery

Problem:

Flower shops receive orders from customers and work with independently contracted drivers to deliver those orders to the customers

Component APIs:

Twilio API: Used to send a text message to the customer, letting them know their delivery is coming

Google Maps Distance Matrix API: Used to calculate how far away each driver is from the store

Actors:

Store PICO: The store PICOs can receive new orders and broadcast those orders to available drivers. Stores receive bids from drivers and choose 1 driver to assign for the delivery.

Driver PICO: The driver PICOs are a network of drivers. New drivers are manually added in a 'Web of Trust'. New drivers must be manually added to the existing network in order to receive new orders. Drivers make use of a gossip protocol to propagate messages to all drivers in the network.

API:

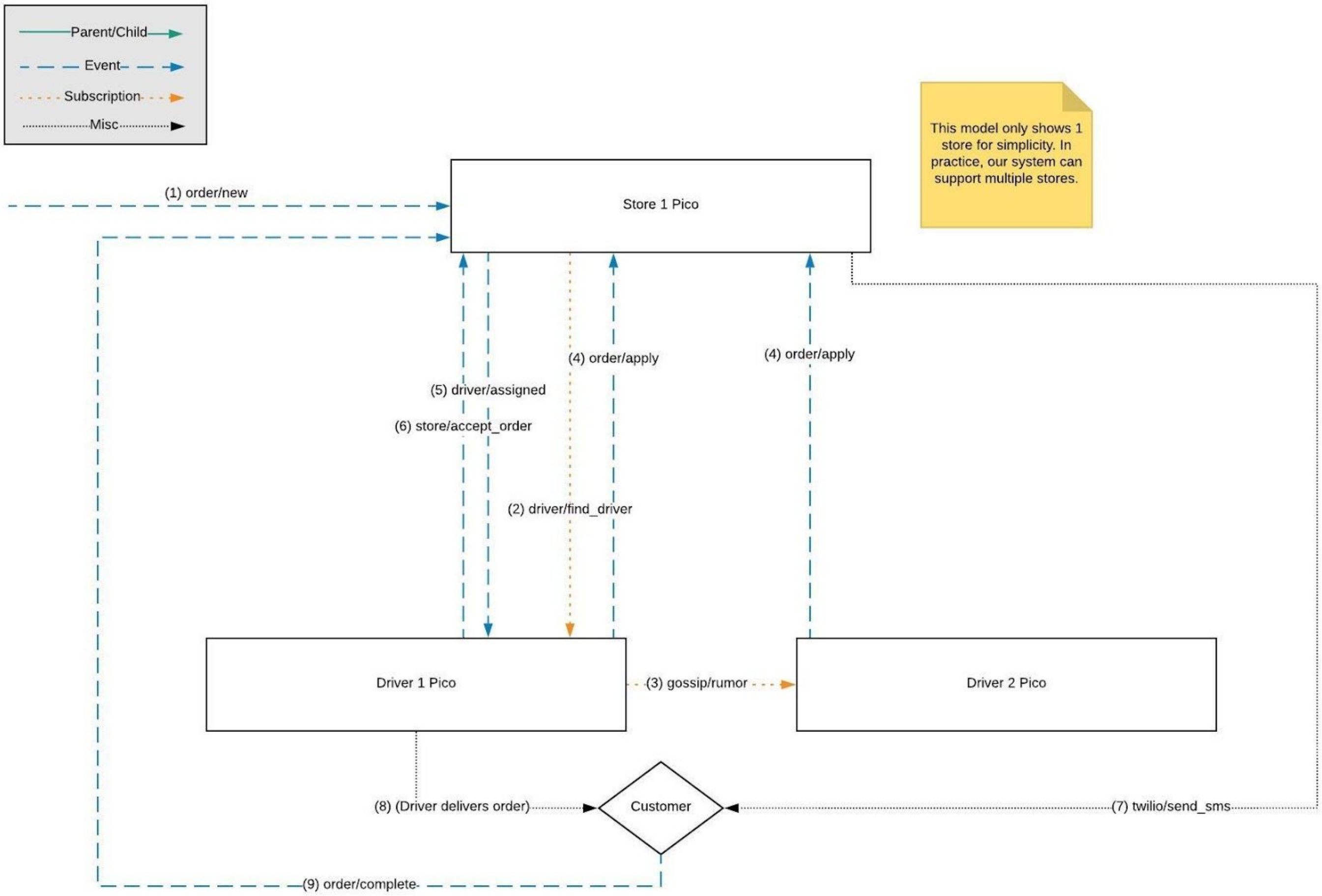
Events:

- **Order/New:** Notifies the store that a new order has been received
- **Driver/Find_driver:** Broadcasts to driver that a new order has been received
- **Order/Apply:** Drivers send this event to 'bid' on orders
- **Driver/Assign:** Once the store has received bids and selected a driver, it sends this event to the driver that it chose
- **Store/Accept_Order:** Drivers send this event to the store when it has received a new assignment and is on-route to deliver the order
- **Order/Complete:** The driver sends this event to the store when the order is complete

Queries:

- **Get_unassigned_orders:** Returns all orders which have not been assigned
- **Get_orders:** Returns all orders that have been received by a store
- **GetCurrentOrder:** Returns the current order if a driver has accepted one; otherwise it returns nothing
- **GetDriverName:** Returns the drivers name
- **GetLocation:** Returns the drivers current location
- **GetPhoneNumber:** Returns the drivers phone number

Architectural Diagram:



Explanation:

1. Store receives new order
2. Store broadcasts new order to subscribed drivers
3. Drivers use gossip protocol to propagate message to all drivers in the network
4. As drivers receive information about new order, they send bids to the store
5. After a delay, the store choses the closet driver out of all of the drivers that applied for the order and assigns them the deliver

6. The driver accepts the order
7. A text is sent to the customer notifying them that their order is on-route
8. The driver delivers the order
9. The customer uses the drivers device to notify the store that the delivery is complete

Pros/Cons:

Pros:

- Events and gossip protocol offer eventual consistency
- Code is naturally compartmentalized when using events and rules

Cons:

- Requires trust when using the gossip protocol.
- Prone to “gaming” the system
- Our current implementation has no way for the driver to constantly update their location
- There is latency in the system while drivers are gossiping about the order and responding
- Race conditions happened often and made it difficult to know what information was where