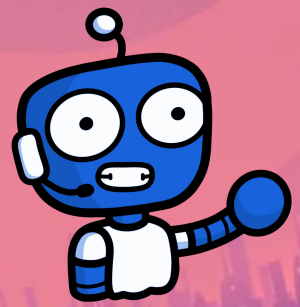
SIREN

Kraken RoboTradeAssistant



The future is Automation

ChatBot = perfect example of Automation

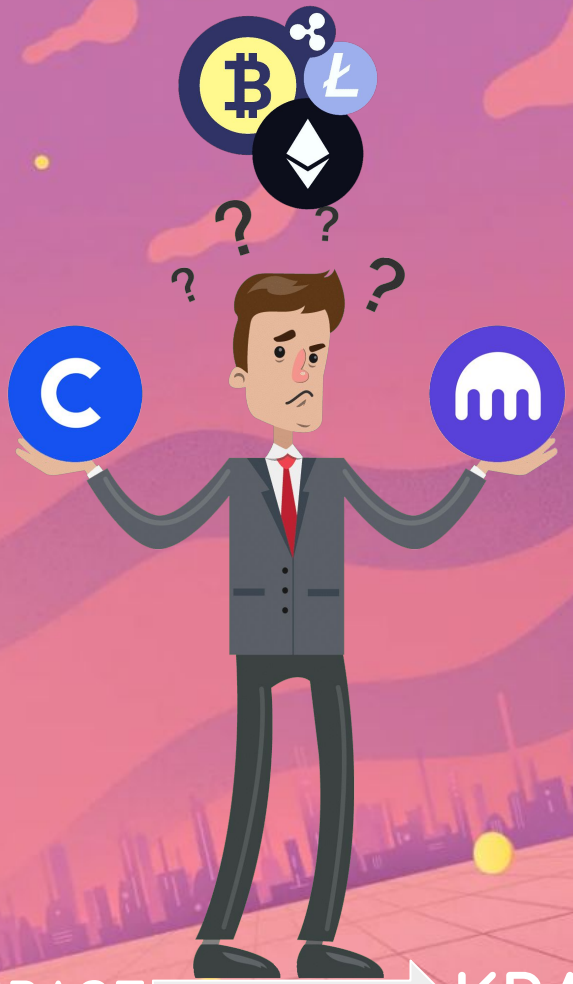
- Artificial intelligence will displace 40% of jobs worldwide within 15 years
- 1.4 billion people use messaging apps and are willing to talk to chatbots
- Chatbots saw a 92% use increase since 2019, making it the brand communication channel with the largest growth
- Growing willingness to engage with chatbots in a variety of ways more than doubling from 2019 to 2020
- 87.2% of consumers rate their typical chatbot experience as within the range of neutral to positive

High Premiums

High exchange fees

Potential delayed transaction times

Limited selection of Alt Coins



Low Premiums

Low exchange fees

Sophisticated trading platform

Greater selection of Alt Coins

Higher interest in staking rewards

COINBASE → KRAKEN

support.kraken.com

Kraken Live Chat

Kai - Virtual Assistant
Ask me something!

Type a message here...

support.kraken.com

Kraken Live Chat

Kai
Virtual Assistant

Note: If you are unable to convert a certain currency directly to your desired currency, it may mean that there is no direct trading pair between the two assets. For example, if you want to use CAD to buy DOGE, you need to convert your CAD to USD, EUR or BTC, and then convert to DOGE.

- I want to create an account.
- I want to trade on the regular exchange.
- Show me the supported trading pairs.
- Tell me about the trading fees.
- Show me how to deposit crypto.
- Show me how to deposit fiat.
- No, it's something else.

Tell me about the trading fees.

Type a message here...

Kraken RoboTradeAssistant

Siren

Siren Prototype

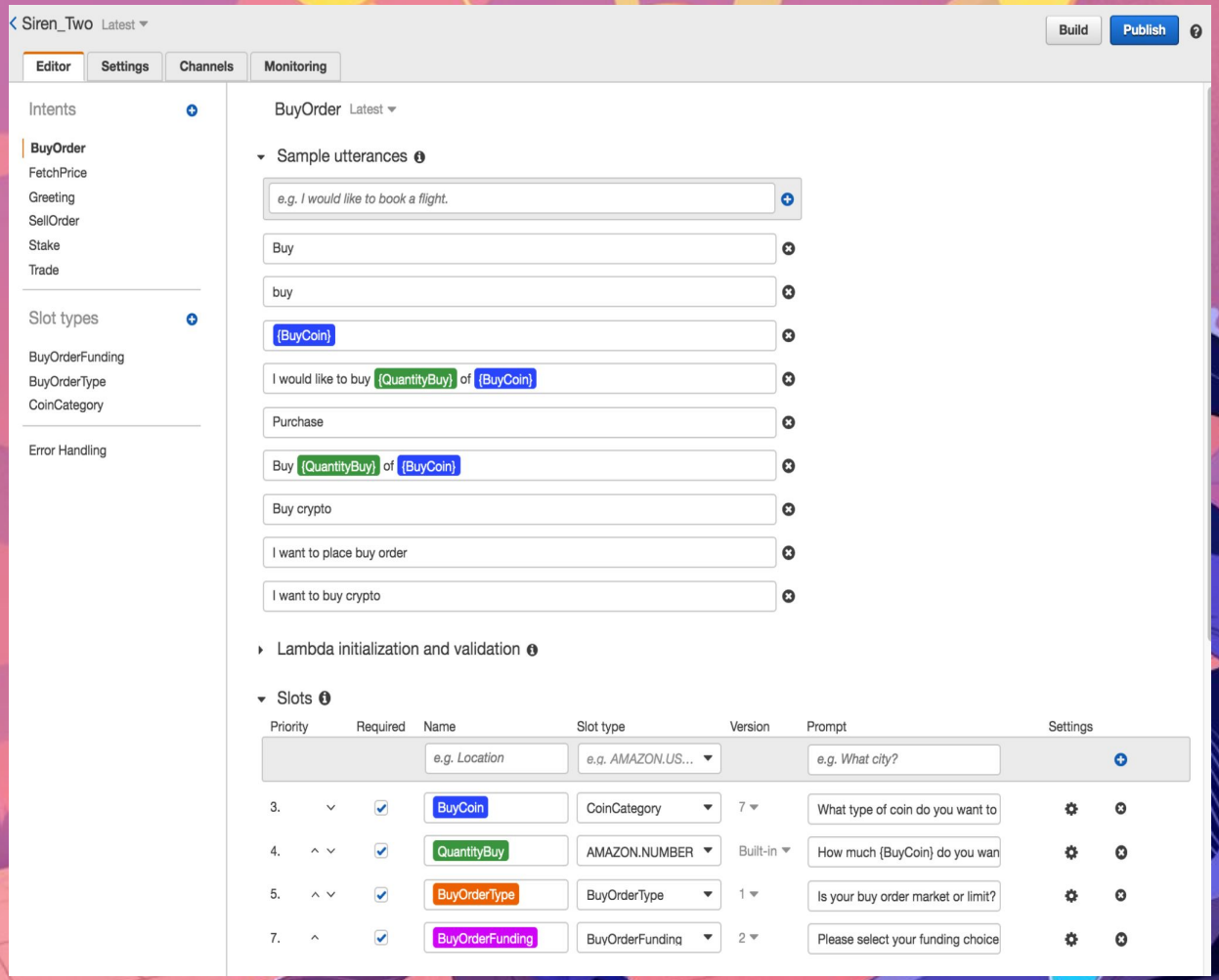
AWS Lex + AWS Lambda

Lex: 6 Intents- Greeting, Trade, BuyOrder, SellOrder, Stake and FetchPrice

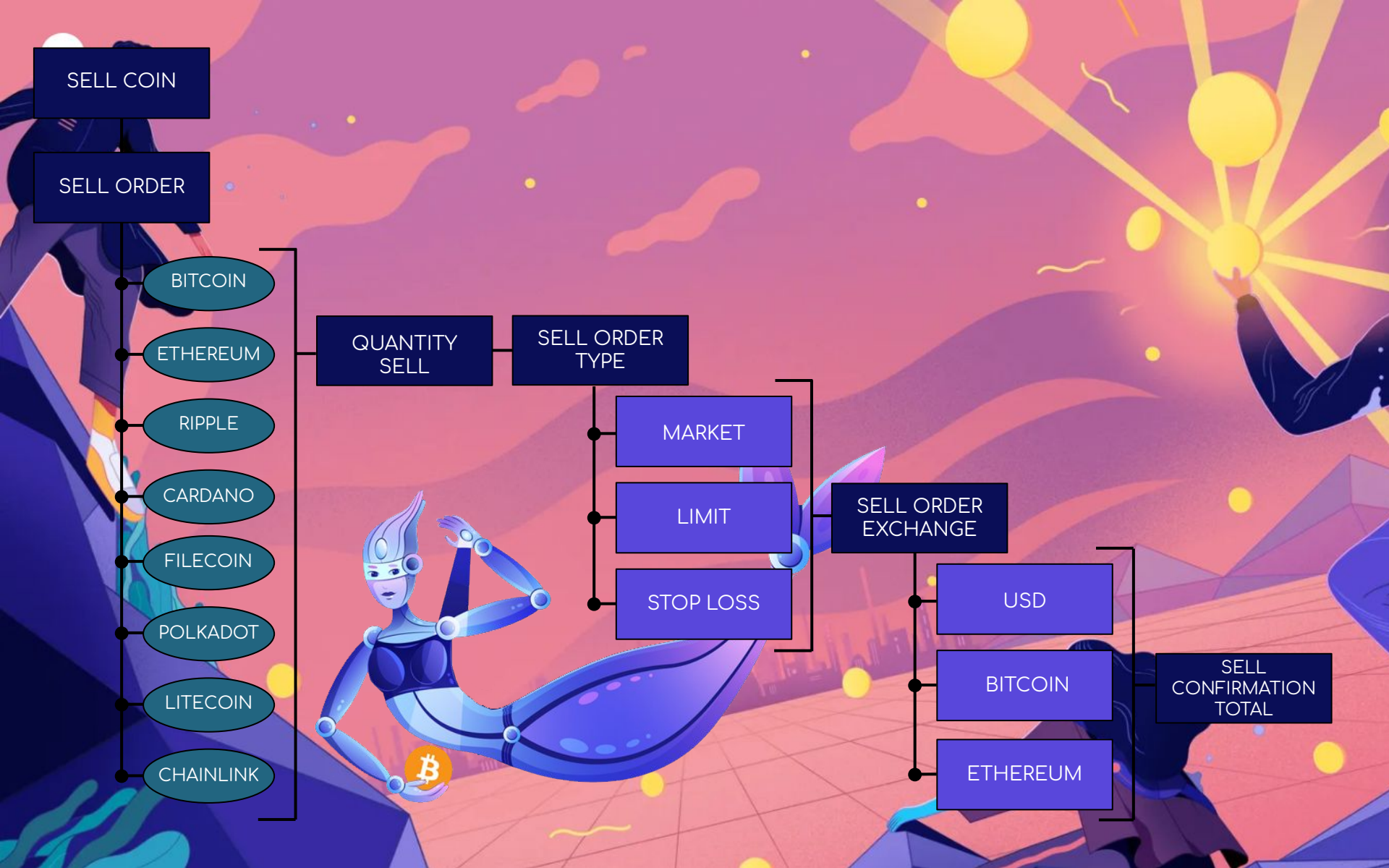
1- 5 slots (each)

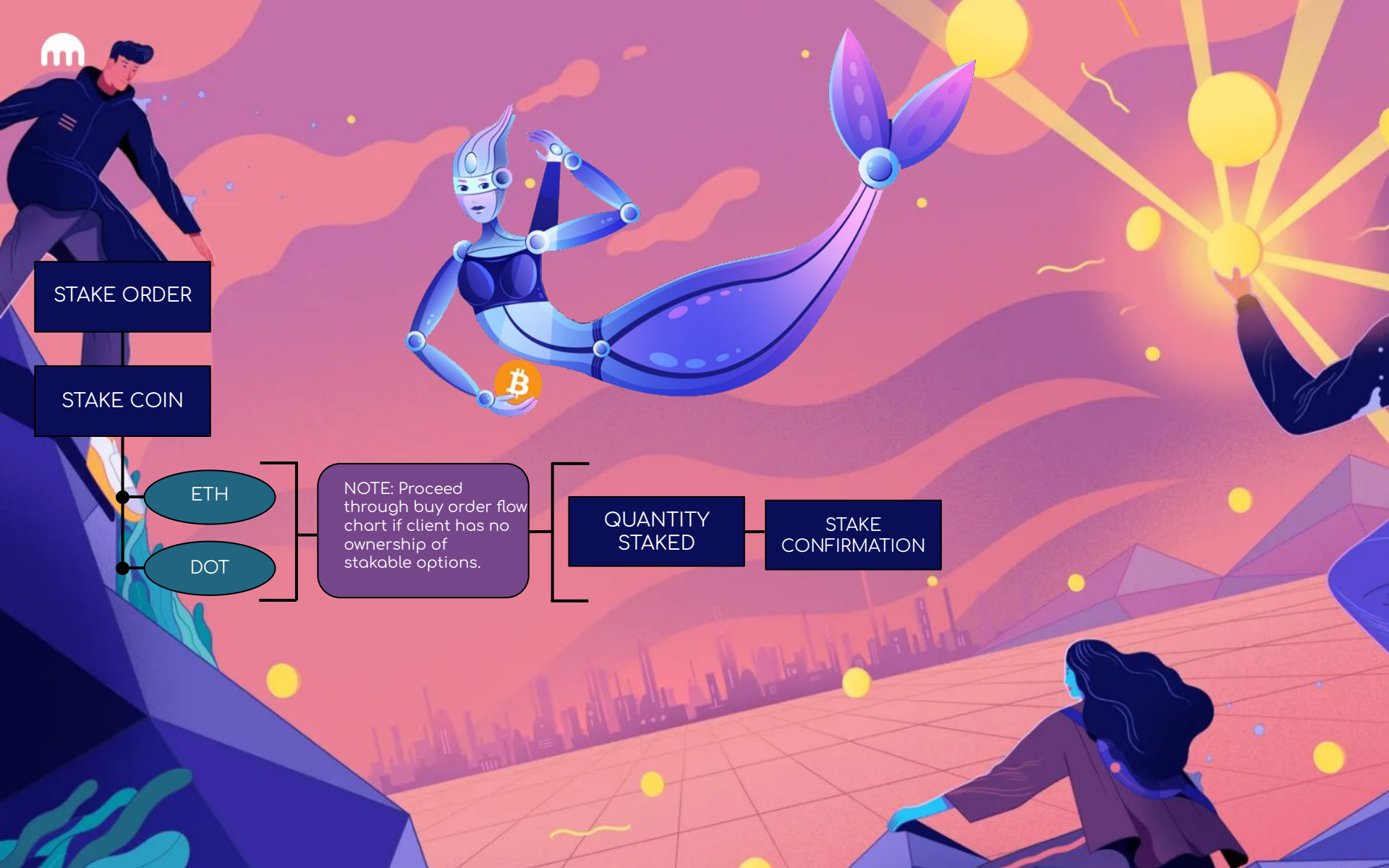
Prompts w/ response cards user's utterances -----> Siren's execution

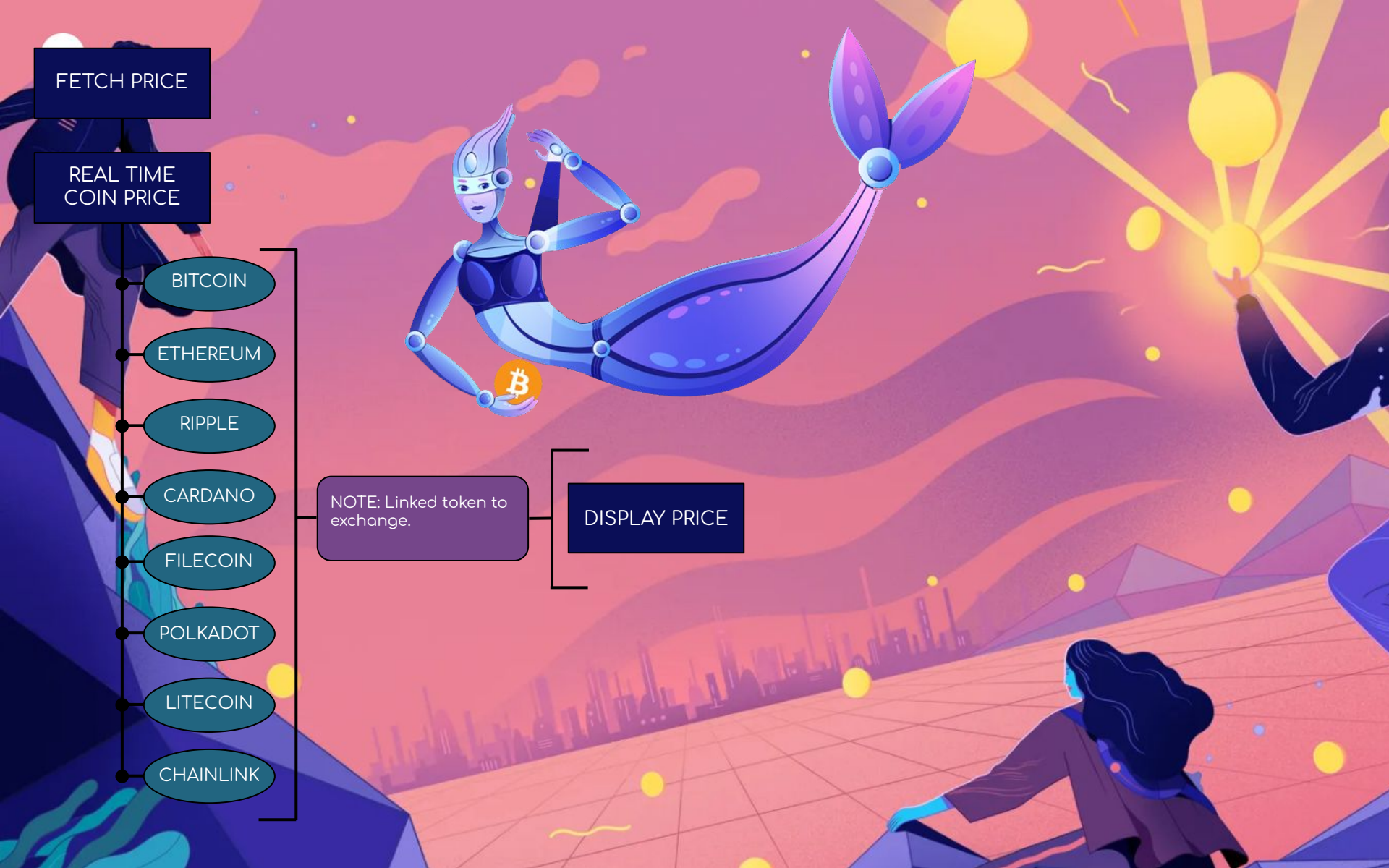
Lambda (Python) allows to implement custom business logic (ex. API call to Kraken fetching current market price)

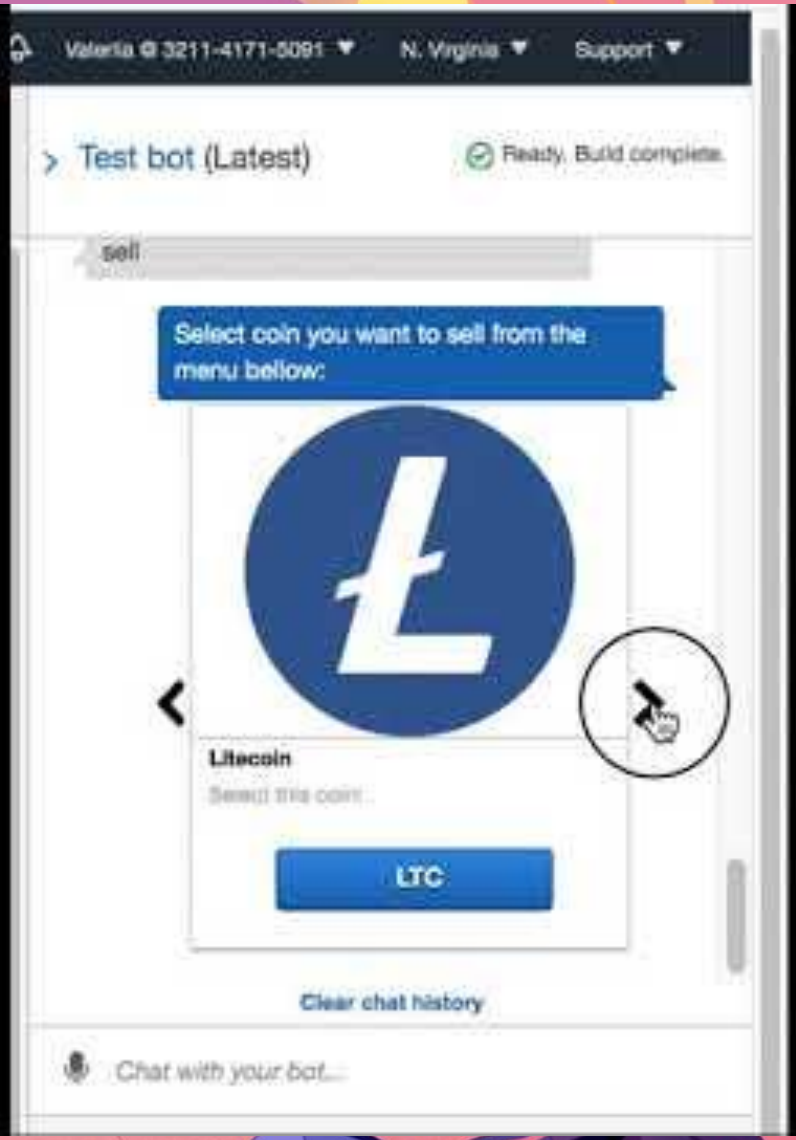



BUY COIN
BUY ORDER
BITCOIN
ETHEREUM
RIPPLE
CARDANO
FILECOIN
POLKADOT
LITECOIN
CHAINLINK
QUANTITY BUY
BUY ORDER TYPE
MARKET
LIMIT
BUY ORDER FUNDING
USD
BITCOIN
ETHEREUM
BUY CONFIRMATION TOTAL


SELL COIN
SELL ORDER
BITCOIN
ETHEREUM
RIPPLE
CARDANO
FILECOIN
POLKADOT
LITECOIN
CHAINLINK
QUANTITY SELL
SELL ORDER TYPE
MARKET
LIMIT
STOP LOSS
SELL ORDER EXCHANGE
USD
BITCOIN
ETHEREUM
SELL CONFIRMATION TOTAL


STAKE ORDER
STAKE COIN
ETH
DOT
NOTE: Proceed through buy order flow chart if client has no ownership of stakable options.
QUANTITY STAKED
STAKE CONFIRMATION


FETCH PRICE
REAL TIME COIN PRICE
BITCOIN
ETHEREUM
RIPPLE
CARDANO
FILECOIN
POLKADOT
LITECOIN
CHAINLINK
NOTE: Linked token to exchange.
DISPLAY PRICE


Valeria @ 3211-4171-5091
N. Virginia
Support
Test bot (Latest)
Ready. Build complete.
sell
Select coin you want to sell from the menu bellow:
Litecoin
LTC
Clear chat history
Chat with your bot...

Backend Coding

lambda_function × Execution results ×

```python
def elicit_slot(session_attributes, intent_name, slots, slot_to_elicit, message):
    """
    Defines an elicit slot type response.
    """

    return {
        "sessionAttributes": session_attributes,
        "dialogAction": {
            "type": "ElicitSlot",
            "intentName": intent_name,
            "slots": slots,
            "slotToElicit": slot_to_elicit,
            "message": message,
        },
    }


def delegate(session_attributes, slots):
    """
    Defines a delegate slot type response.
    """

    return {
        "sessionAttributes": session_attributes,
        "dialogAction": {"type": "Delegate", "slots": slots},
    }


def close(session_attributes, fulfillment_state, message):
    """
    Defines a close slot type response.
    """
```

```python
#-----------------------------------------------------------------------------
### PART 2
### RETRIEVING PRICE OF CRYPTO FROM KRAKEN.COM
""" Retrieves the current price of cryto in US Dollars from the kraken.com API."""
crypto_usd = 'adaxbt'
def price_usd(crypto_usd):
    crypto_map = {
        'xbtusd': 'XXBTZUSD',
        'adausd': 'ADAUSD',
        'ethusd': 'XETHZUSD',
        'xrpusd': 'XXRPZUSD',
        'dotusd': 'DOTUSD',
        'filusd': 'FILUSD',
        'linkusd': 'LINKUSD',
        'ltcusd': 'XLTCZUSD',
    }
    kraken_url = f'https://api.kraken.com/0/public/Ticker?pair={crypto_map[crypto_usd]}'
    response = requests.get(kraken_url)
    response_json = response.json()
    price = response_json['result'][crypto_map[crypto_usd]]['c'][0]
    return price

""" Retrieves price of crypto bought using Bitcoin from the kraken.com API."""
crypto_xbt = 'adaxbt'
def price_xbt(crypto_xbt):
    crypto_xbt_map = {
        'adaxbt': 'ADAXBT',
        'ethxbt': 'XETHXXBT',
        'xrpxbt': 'XXRPXXBT',
        'dotxbt': 'DOTXBT',
        'filxbt': 'FILXBT',
        'linkxbt': 'LINKXBT',
        'ltcxbt': 'XLTCXXBT',
```

Backend Coding

```python
### PART 3 Intents Handlers ###

def Buy_Order(intent_request):
    """
    Performs dialog management and fulfillment for buying order.
    """
    buy_coin = get_slots(intent_request)["BuyCoin"]
    quantity_buy = get_slots(intent_request)["QuantityBuy"]
    buy_order_type = get_slots(intent_request)["BuyOrderType"]
    buy_order_funding = get_slots(intent_request)["BuyOrderFunding"]
    order_price = 0
    if buy_order_funding == "ETH":
        pair = (buy_coin + buy_order_funding).lower()
        order_price = price_eth(pair)
        total_cost = round(float(quantity_buy) * float(order_price) * 1.0016,8)
    elif buy_order_funding == "XBT":
        pair = (buy_coin + buy_order_funding).lower()
        order_price = price_xbt(pair)
        total_cost = round(float(quantity_buy) * float(order_price) * 1.0016,8)
    else:
        pair = (buy_coin + buy_order_funding).lower()
        order_price = price_usd(pair)
        total_cost = round(float(quantity_buy) * float(order_price) * 1.0016,2)
    return close(
        intent_request["sessionAttributes"],
        "Fulfilled",
        {
            "contentType": "PlainText",
            "content": """Great news, your buy order has been filled for {} {} for a total amount {} {}!
            """.format(
                quantity_buy, buy_coin, buy_order_funding, total_cost
            ),
```

Code source Info

Upload from

File Edit Find View Go Tools Window **Test** Deploy Changes deployed

Go to Anything (⌘ P)

Environment

Siren - /

lambda_function.py

lambda_function × Execution result × +

Execution results — Status: Succeeded | Max memory used: 57 MB | Time: 287.14 ms

Response

```
{
  "sessionAttributes": {},
  "dialogAction": {
    "type": "Close",
    "fulfillmentState": "Fulfilled",
    "message": {
      "contentType": "PlainText",
      "content": "Great news, your buy order has been filled for 10 ETH for a total amount USD 25284.07!
    }
  }
}
```

Function Logs

```
START RequestId: 59e85a21-e18b-42f6-b2e4-017e65c28f4c Version: $LATEST
/var/runtime/botocore/vendored/requests/api.py:72: DeprecationWarning: You are using the get() function f
  DeprecationWarning
END RequestId: 59e85a21-e18b-42f6-b2e4-017e65c28f4c
REPORT RequestId: 59e85a21-e18b-42f6-b2e4-017e65c28f4c  Duration: 287.14 ms Billed Duration: 288 ms Memor
```

Request ID

59e85a21-e18b-42f6-b2e4-017e65c28f4c

Prospective growth of Siren

Immediate - NLP + Machine Learning = intuitive and advanced Siren

Level Up 1- Execute more complex trades, stop loss and take profit orders etc

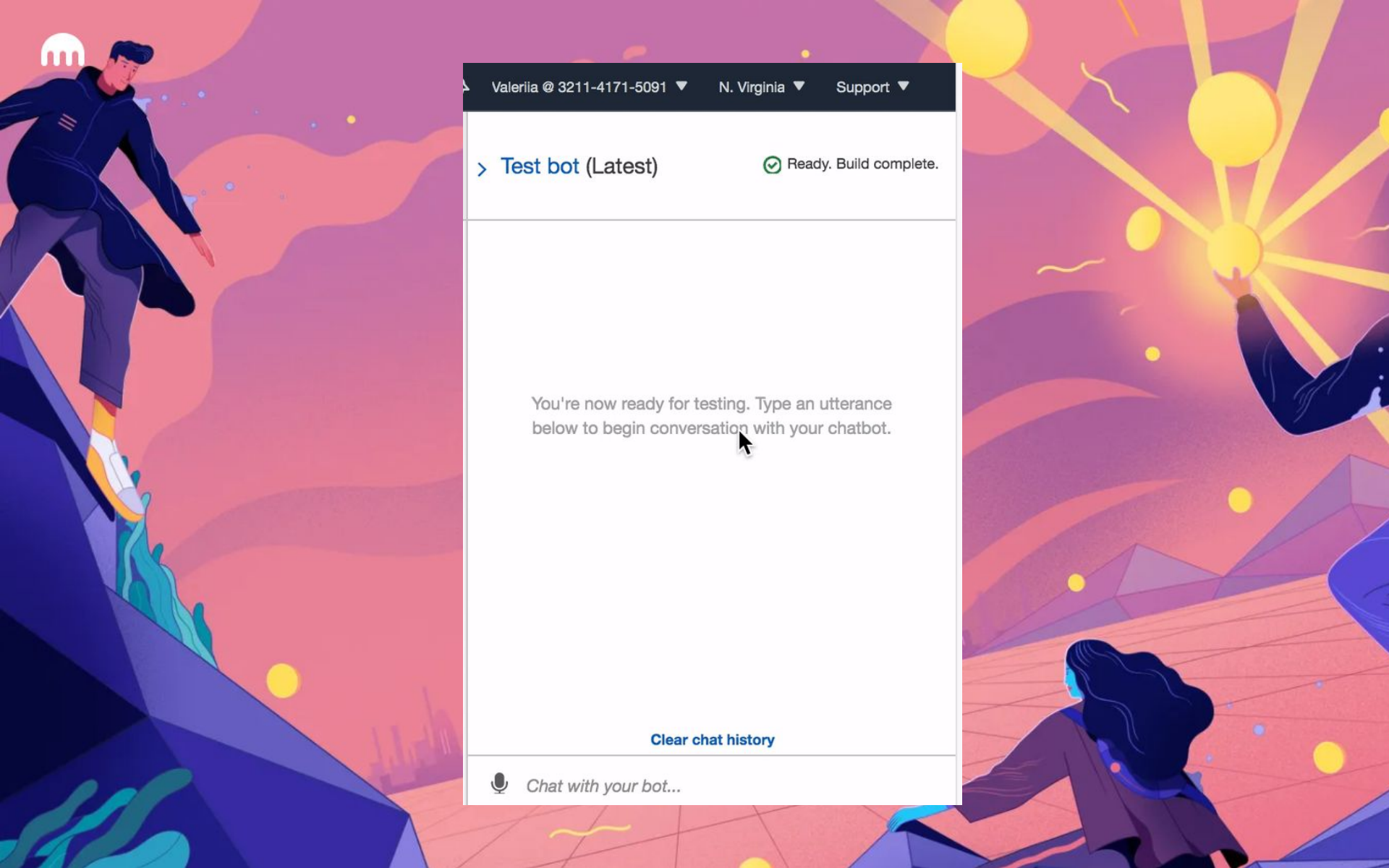
Level Up 2- Make Siren DeFi ready for Uniswap and Polkadex (user navigation)

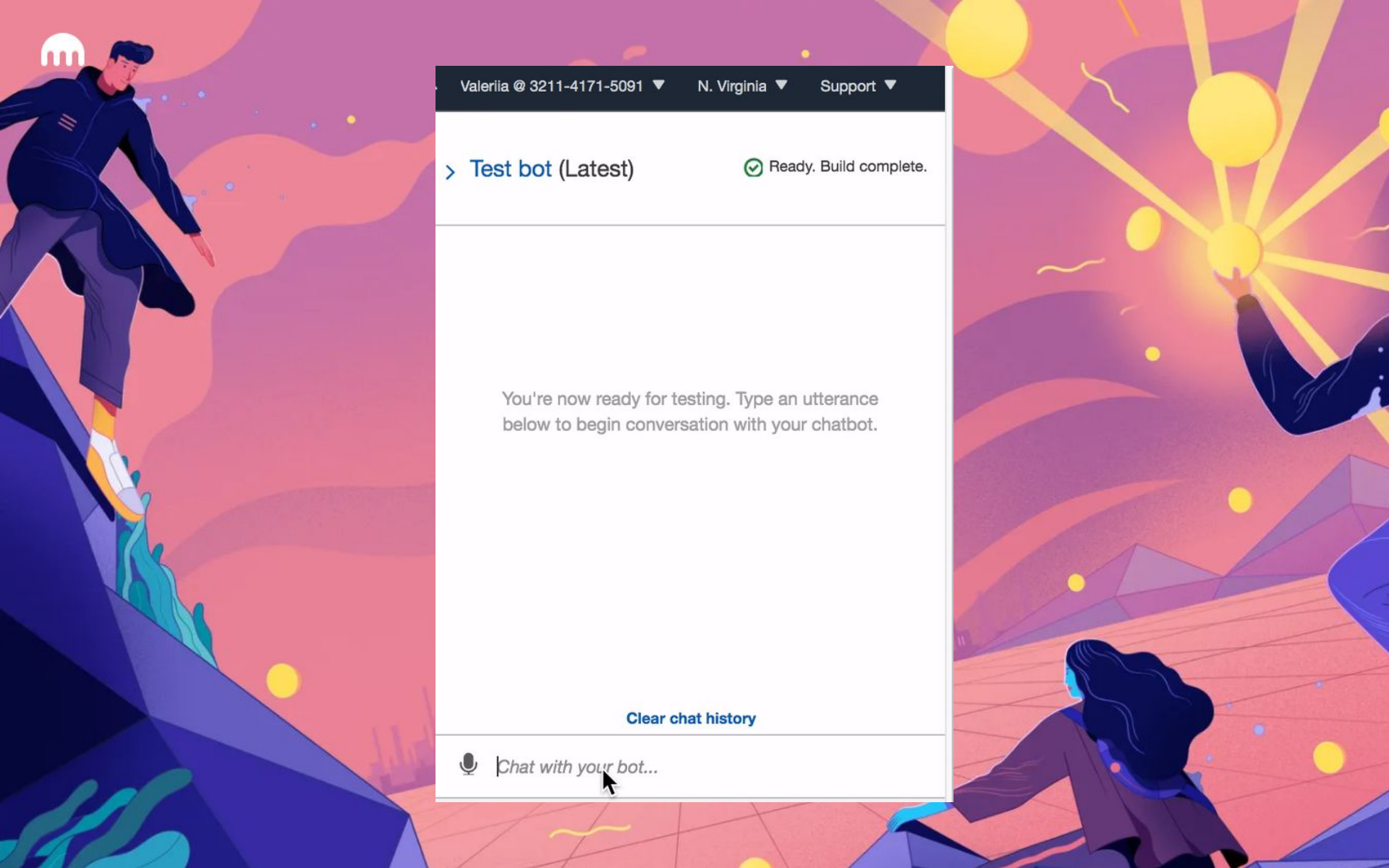
- Defi efficiency to spot and execute on the exchange with the lowest exchange rates
- Spots trade arbitrage opportunity between time zones and exchanges

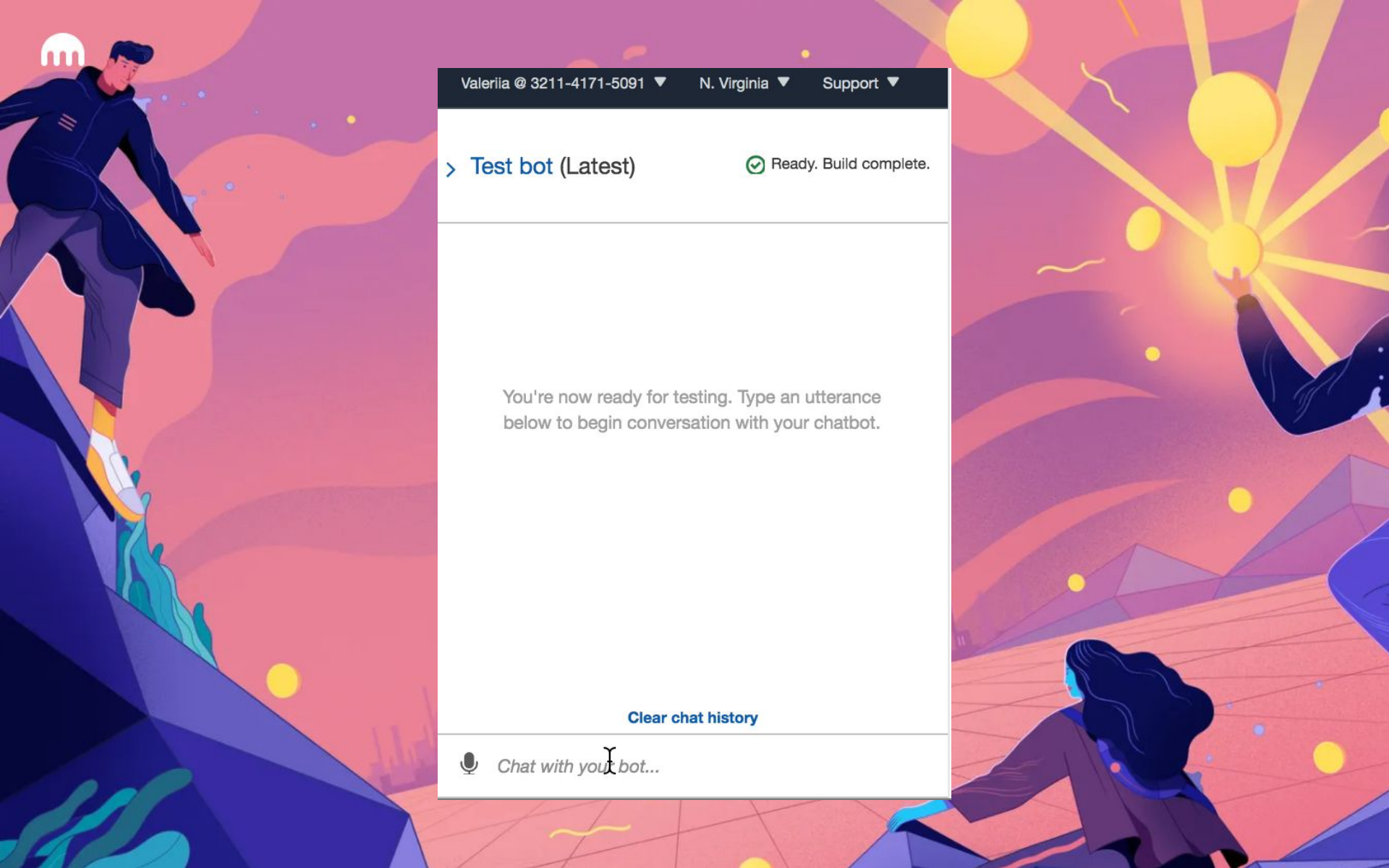
Level Up 3 - Siren AI exchange integrated trading bot for client subscription conducting all work

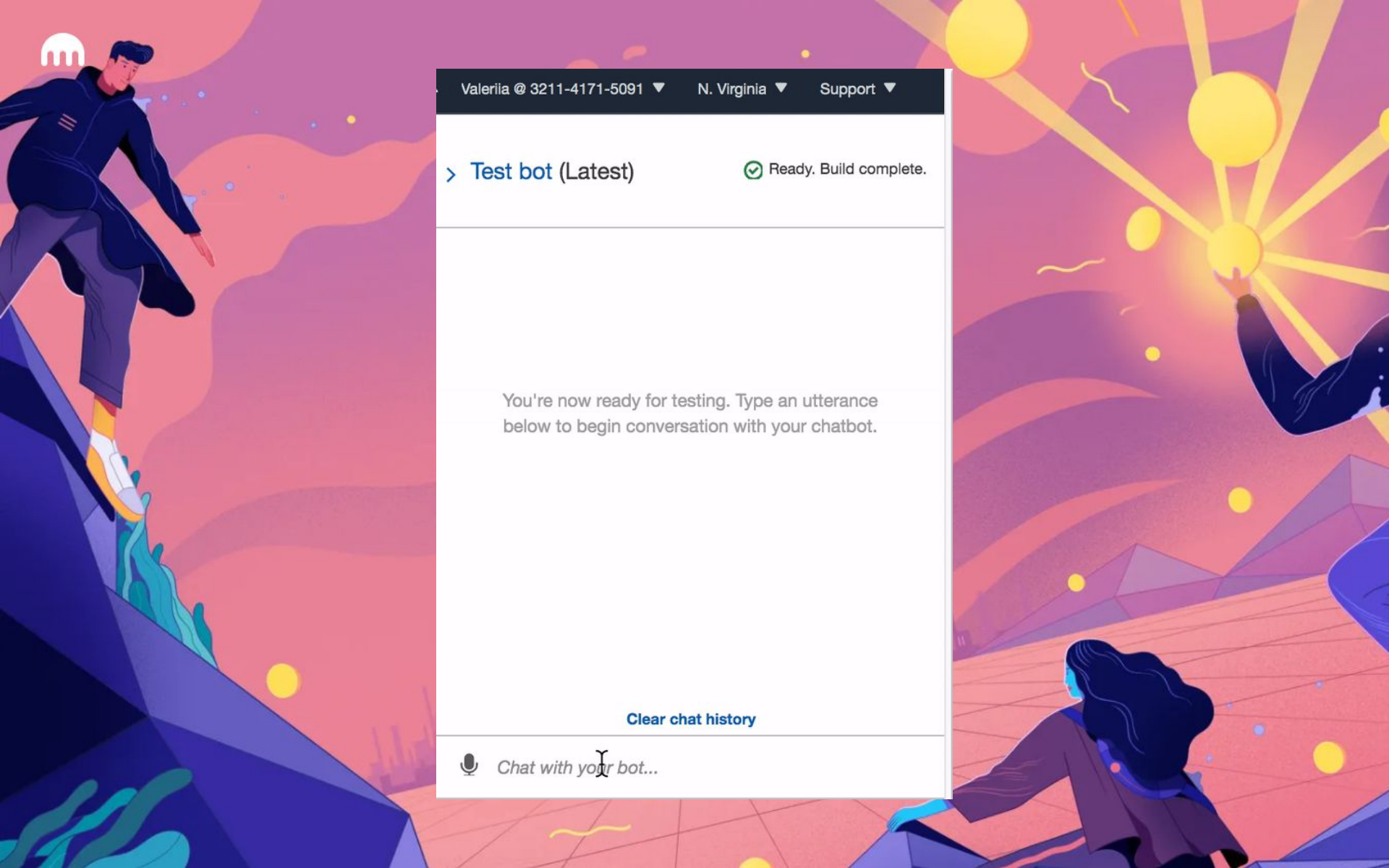
- Data collection over time will inform Siren common trade trends, sentiment, and when to execute an order
- Data collection allows Siren to identify type of market and adjust trading strategy (i.e. bull, bear, wave market)
- While Siren cannot predict the news, recognizing sentiment, it will adjust to market volatility

THANK YOU


Valeriia @ 3211-4171-5091
N. Virginia
Support
Test bot (Latest)
Ready. Build complete.
You're now ready for testing. Type an utterance below to begin conversation with your chatbot.
Clear chat history
Chat with your bot...


Valerlia @ 3211-4171-5091
N. Virginia
Support
Test bot (Latest)
Ready. Build complete.
You're now ready for testing. Type an utterance below to begin conversation with your chatbot.
Clear chat history
Chat with your bot...


Valeriia @ 3211-4171-5091
N. Virginia
Support
Test bot (Latest)
Ready. Build complete.
You're now ready for testing. Type an utterance below to begin conversation with your chatbot.
Clear chat history
Chat with your bot...


Valeriia @ 3211-4171-5091
N. Virginia
Support
Test bot (Latest)
Ready. Build complete.
You're now ready for testing. Type an utterance below to begin conversation with your chatbot.
Clear chat history
Chat with your bot...