Using The Graph and Dune to Power on-chain Actions

Christian St.Louis & Mattias Lightstone
Component.fi
Agenda

1. Get to know each other
2. Overview of web3 data stack
3. Description of the phases and simple example
   a. Data fetching
   b. Data processing and modeling
   c. Transaction execution
4. Breakout sessions - Designing your own automation
<table>
<thead>
<tr>
<th>Subject</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>We have closed positions in your account - meet your margin call as per Q...</td>
<td>Sep 20</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Sep 20</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Sep 19</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Sep 16</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Sep 15</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Sep 14</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Sep 7</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Sep 2</td>
</tr>
<tr>
<td>We have closed positions in your account - meet your margin call as per Q...</td>
<td>Aug 30</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Aug 30</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Aug 29</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Aug 26</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Aug 25</td>
</tr>
<tr>
<td>ACTION REQUIRED: your account is in a margin call - in a margin call (your...</td>
<td>Aug 24</td>
</tr>
</tbody>
</table>
Get to know your table

Prompt: a time when things didn’t go your way in web3.
(5 minutes)
Overview of Web3 Data Stack
The Web3 Data Stack

Web3 Data Stack Landscape

Interaction + Data Source
Comes as self-run clients or node endpoints-as-a-service. These both allow you to read data and send transactions (write) to the blockchain.

Query + Data Mapping
Parsing transaction, state, and event log data into a format that can be queried with traditional languages like SQL or GraphQL.

High Touch Data Providers
These are more akin to traditional data providers like Bloomberg or capiQ, where you get to play with data outputs only.

Clients
- ConsenSys/Quorum
- Ethereum/Go-ethereum
- ledgerwatch/erigon

Nodes-as-a-Service
- Infura
- POKT
- alchemy

Key Communities
- flashbets

Transaction Data
- Dune Analytics
- flipsidecrypto
- thegraph

Metadata "Protocols"
- ceramic
- IPFS

Specialty Providers
- blocknative
- parsec
- boardroom
- DeepDAO

"Enterprise" Services
- token terminal
- nansen
- Chainalysis

Enhanced APIs
- alchemy
- Covalent
- Etherscan

OurNetwork
Here’s the process.

Data Fetching

Data Processing & Modeling

Transaction Execution
Using Python to collect data and write a simple prediction model

1. Pulling eth price data from the graph and dune
2. Use ARMA to “predict” price movement
3. If price go up → swap USDC to Eth on uniswap
Section 2

Data Fetching
Available fields
Data format
Scope (Chains, protocols)
Event driven or polled
Flexibility
Stability
Latency
Trust
Reliability
Openness of transformation functions
Openness of the data

Data fetching, Some Considerations
Fetching ETH/USD price data from the Graph and Dune
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Time</td>
<td>Price</td>
</tr>
<tr>
<td>2022-09-23</td>
<td>07:00:00</td>
<td>133257188668</td>
</tr>
<tr>
<td>2022-09-23</td>
<td>08:00:00</td>
<td>131439897822</td>
</tr>
<tr>
<td>2022-09-23</td>
<td>09:00:00</td>
<td>130781000000</td>
</tr>
<tr>
<td>2022-09-23</td>
<td>10:00:00</td>
<td>129334000000</td>
</tr>
<tr>
<td>2022-09-23</td>
<td>11:00:00</td>
<td>128587280014</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>11:00:00</td>
<td>129850000000</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>12:00:00</td>
<td>129445000000</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>13:00:00</td>
<td>129773166484</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>14:00:00</td>
<td>129740000000</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>15:00:00</td>
<td>129980479341</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Time</td>
<td>Price</td>
</tr>
<tr>
<td>2022-09-13</td>
<td>00:00:00+00:00</td>
<td>1696.105333</td>
</tr>
<tr>
<td>2022-09-13</td>
<td>01:00:00+00:00</td>
<td>1692.591667</td>
</tr>
<tr>
<td>2022-09-13</td>
<td>02:00:00+00:00</td>
<td>1695.198667</td>
</tr>
<tr>
<td>2022-09-13</td>
<td>03:00:00+00:00</td>
<td>1706.766000</td>
</tr>
<tr>
<td>2022-09-13</td>
<td>04:00:00+00:00</td>
<td>1715.168333</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>11:00:00+00:00</td>
<td>1303.834833</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>12:00:00+00:00</td>
<td>1301.704167</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>13:00:00+00:00</td>
<td>1297.784000</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>14:00:00+00:00</td>
<td>1297.227833</td>
</tr>
<tr>
<td>2022-10-12</td>
<td>15:00:00+00:00</td>
<td>1298.016667</td>
</tr>
</tbody>
</table>
Data Processing and Modeling
Data Modeling, Considerations

Types of modeling
- Conditional logic (wallet tracking)
- Machine learning algorithms
- Time series predictions
- Etc.
Model Results
Transaction Execution
Considerations
● Which chains are you operating on?
● Is it adversarial?
● What kind of timeline are you operating on?
● Key management?
● Should your logic be transparent?
● What happens if the transaction fails, or is congested?

Options
● TX libs (Ethers, Web3JS) (Cloud, local machine)
● Keeper networks (on or off-chain logic)
● Transaction execution services
● Oracle
● Multi-sig
● L2
● Builder API (MEV)
Explore some data sets, come up with an automation idea. (5-10 minutes)
Pitch your idea in 30 seconds.
Select an idea amongst the table (5 minutes)
Design your automation.

Think adversarially. Where are the attack vectors? (10 minutes)
We’re building tools that make this whole process easier.

@componentfi
Thank you!

Christian St.Louis & Mattias Lightstone
Component.fi
christian@component.fi
@componentfi