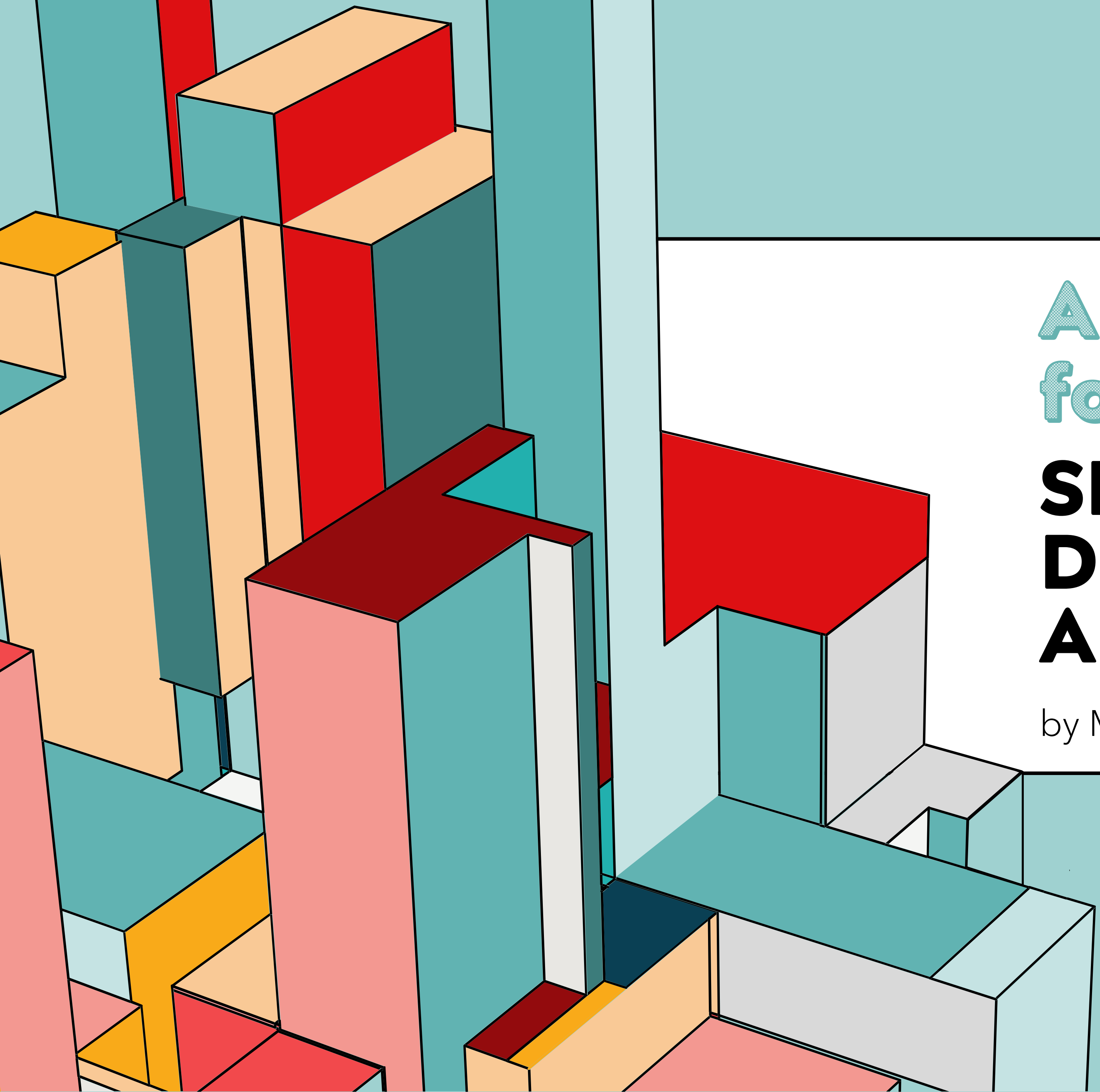


A blockchain pitch for:

SPINDL'S DISTRIBUTED ARCHITECTURE

by Marcus Wong Qi

PROBLEM STATEMENT

Spindl's process, as quoted from Mr. Antonio's blog, will involve "a huge fire hose of data," which SDK collects across every client site.
These sub-transactions are in the millions per second, which is inefficient to store on the smart contract, even with for-loops. Furthermore, the large data sets are ridiculously **costly** to store on-chain as is.
Retrieving and utilizing on-chain data can be executed using web3 libraries, but it is still **unscalable** because of its transactional nature.

OPPORTUNITY

Spindl will use three components, including its **smart contract, IPFS storage, and a subgraph**:

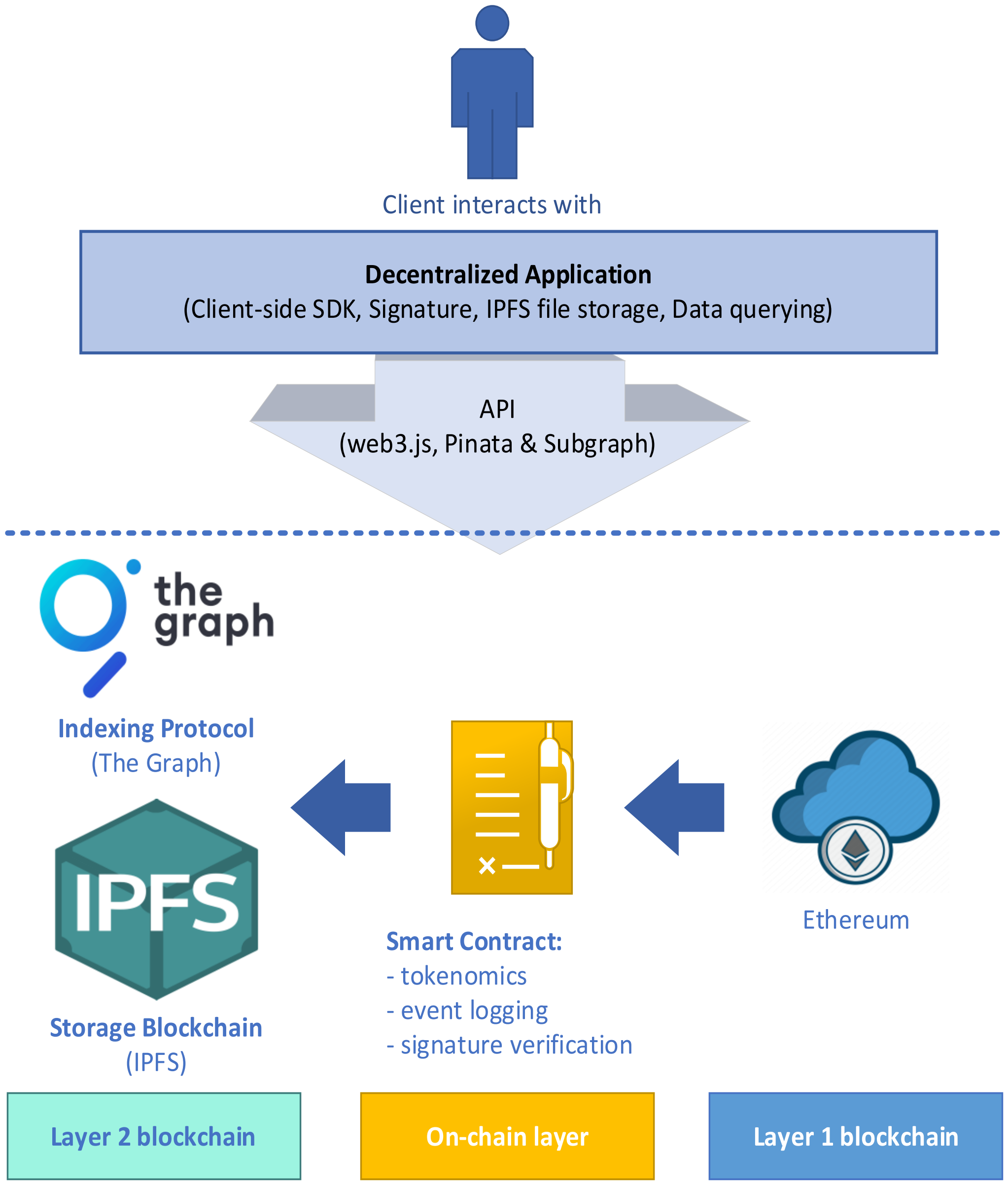
1. Data collected from client-side SDK will be stored on a peer-to-peer IPFS storage. Our client signs a single-bulk transaction, outputting a signature and IPFS file ID
2. The client interacts with the Smart Contract to input the signature and IPFS file ID as an array of transactions. The two data are stored on-chain and is retrievable by event logging
3. Spindl will host a subgraph using the Graph protocol, which enables our clients and ourselves to query the on-chain data. The query is in GraphQL, which can be processed to retrieve the sub-transactions from IPFS for analytics use cases.

POTENTIAL IMPACT

Client enjoys the **flexibility** and **efficiency** to perform bulk transactions on the smart contract because signatures and IPFS files are immutable off-chain entities. IPFS can become a private storage blockchain if data privacy is required.

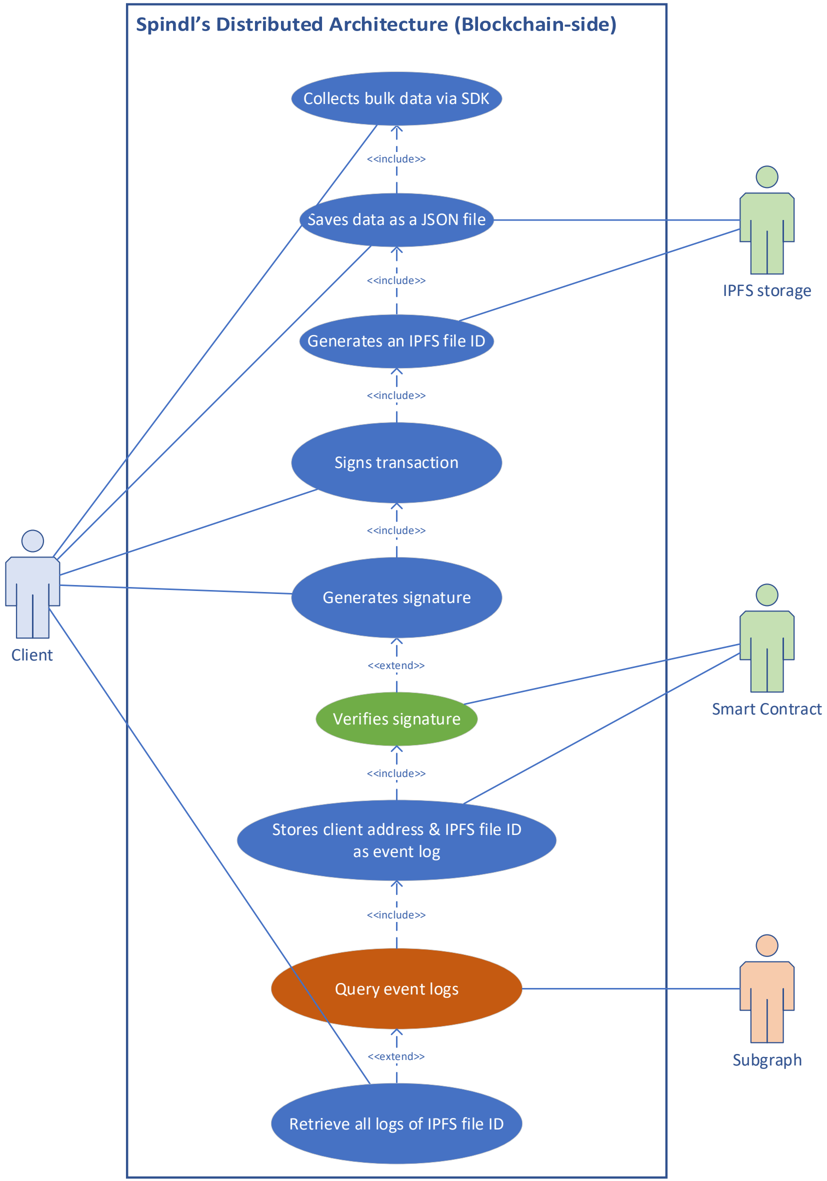
The off-chain process to generate signature has **zero processing time** (compared to on-chain transaction) and can be **automated**.

Streamlined data querying and analytics are interoperable with data visualization tools, such as Power BI and Tableau.



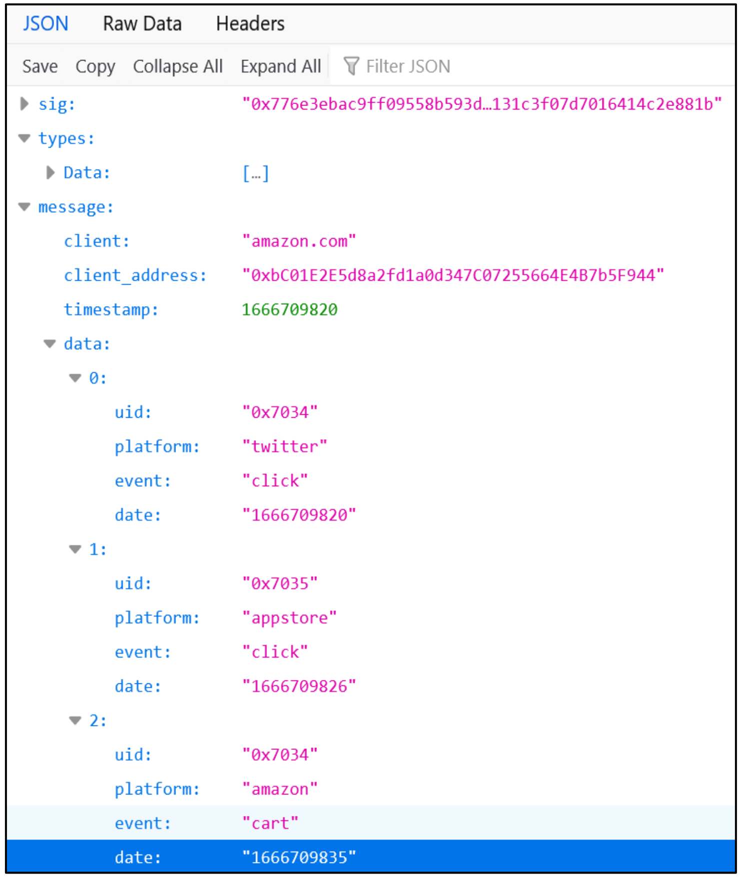
Spindl's Technical Architecture

Use Case diagram:



Spindl's Distributed Architecture (Blockchain-side)
Collects bulk data via SDK
<<include>>
Saves data as a JSON file
<<include>>
Generates an IPFS file ID
<<include>>
Signs transaction
<<include>>
Generates signature
<<extend>>
Verifies signature
<<include>>
Stores client address & IPFS file ID as event log
<<include>>
Query event logs
<<extend>>
Retrieve all logs of IPFS file ID
Client
IPFS storage
Smart Contract
Subgraph

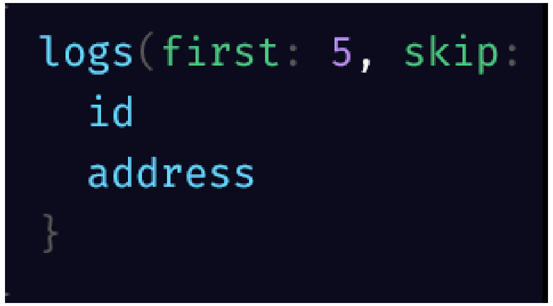
JSON file template for IPFS storage:



```
▶ sig:                "0x776e3ebac9ff09558b593d…131c3f07d7016414c2e881b"
▼ types:
  ▶ Data:             […]
▼ message:
    client:           "amazon.com"
    client_address:   "0xbC01E2E5d8a2fd1a0d347C07255664E4B7b5F944"
    timestamp:        1666709820
  ▼ data:
    ▼ 0:
        uid:          "0x7034"
        platform:     "twitter"
        event:        "click"
        date:         "1666709820"
    ▼ 1:
        uid:          "0x7035"
        platform:     "appstore"
        event:        "click"
        date:         "1666709826"
    ▼ 2:
        uid:          "0x7034"
        platform:     "amazon"
        event:        "cart"
        date:         "1666709835"
```

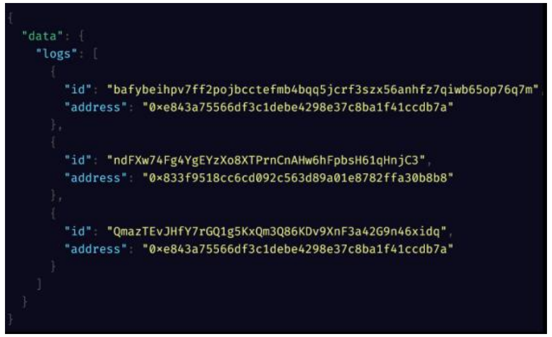
JSON file template to store data attributes

GraphQL query of client address and IPFS file ID:



```
logs(first: 5, skip:
  id
  address
}
```

Query to get first 5 entries of IPFS file ID (id) and client address (address)



```
{
  "data": {
    "logs": [
      {
        "id": "bafybeihpv7ff2pojbcctefmb4bqq5jcrf3szx56anhfz7qiwb65op76q7m",
        "address": "0×e843a75566df3c1debe4298e37c8ba1f41ccdb7a"
      },
      {
        "id": "ndFXw74Fg4YgEYzXo8XTPrnCnAHw6hFpbsH61qHnjC3",
        "address": "0×833f9518cc6cd092c563d89a01e8782ffa30b8b8"
      },
      {
        "id": "QmazTEvJHfY7rGQ1g5KxQm3Q86KDv9XnF3a42G9n46xidq",
        "address": "0×e843a75566df3c1debe4298e37c8ba1f41ccdb7a"
      }
    ]
  }
}
```

Result of the query

Demo subgraph: View link

THANK YOU

Marcus Wong Qi

+6018-355-8288

marcuswongqi@gmail.com

Linkedin