Digital Asset

The Digital Asset Platform

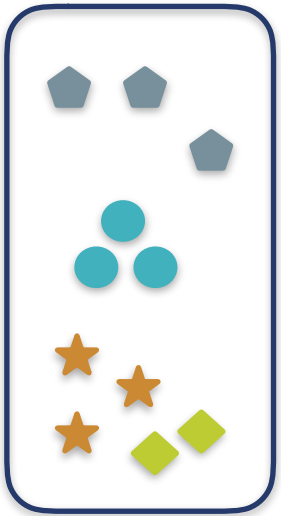
Ben Lippmeier

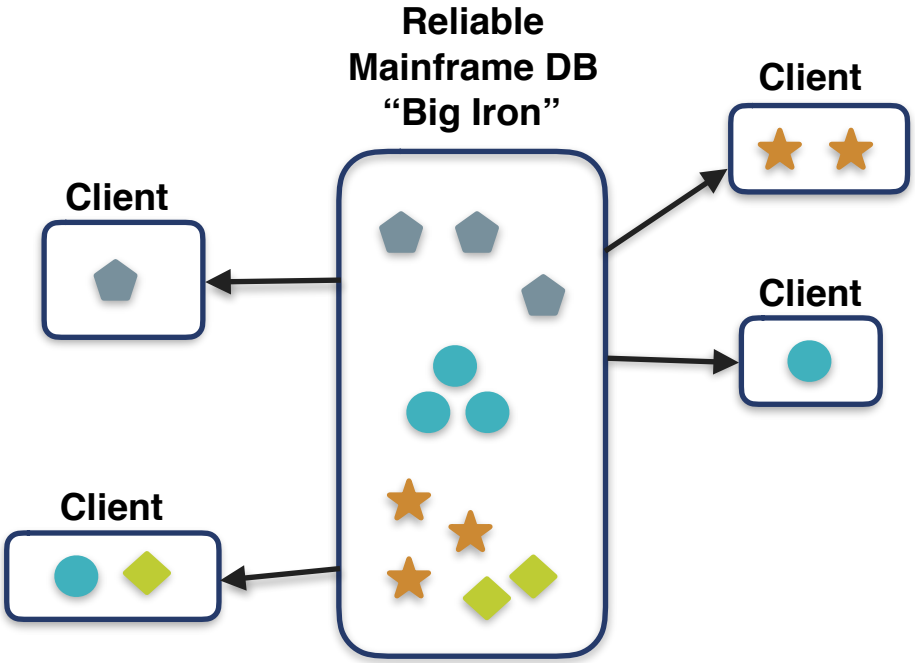
Sydney Blockchain Symposium

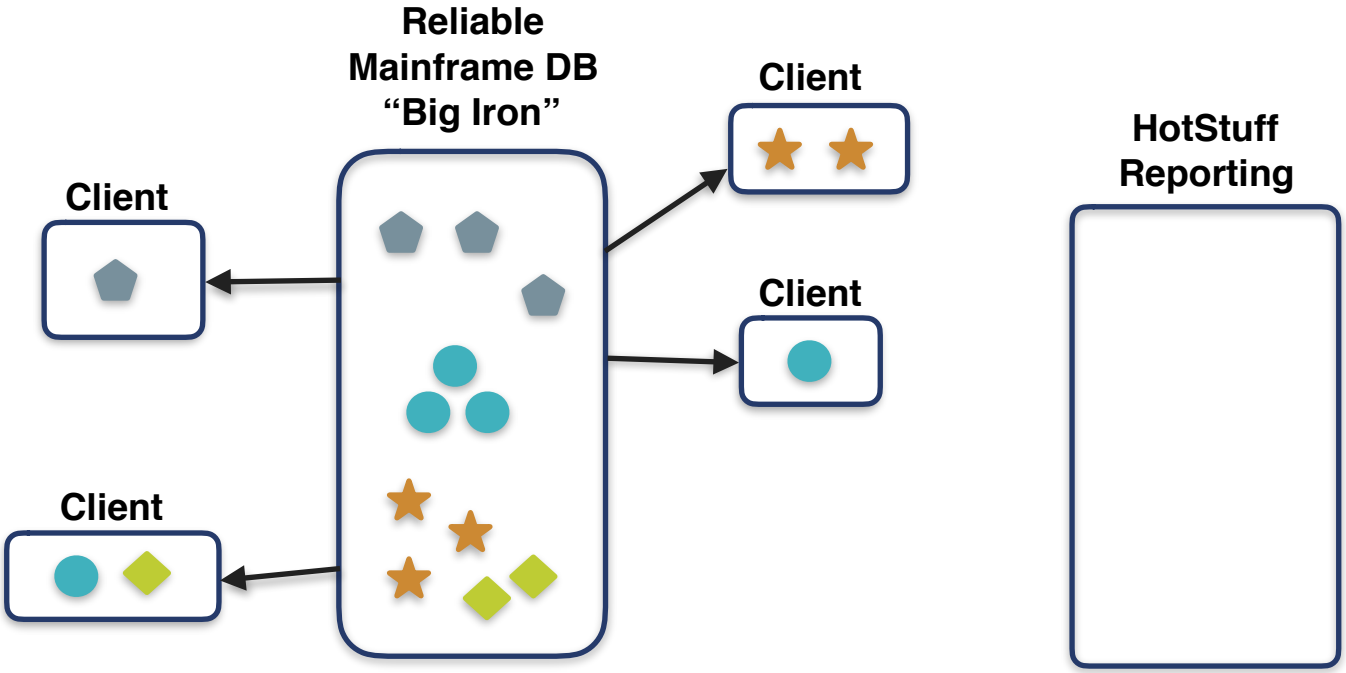
12th February 2018

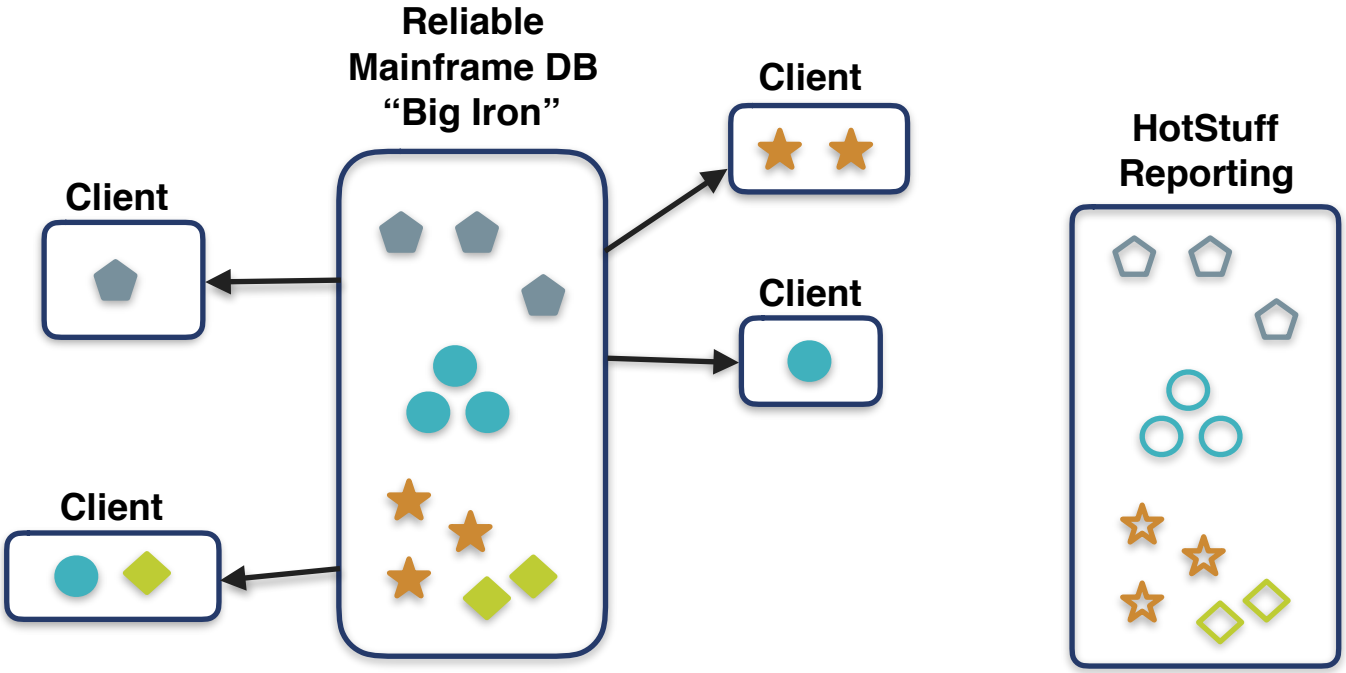
Data Quality

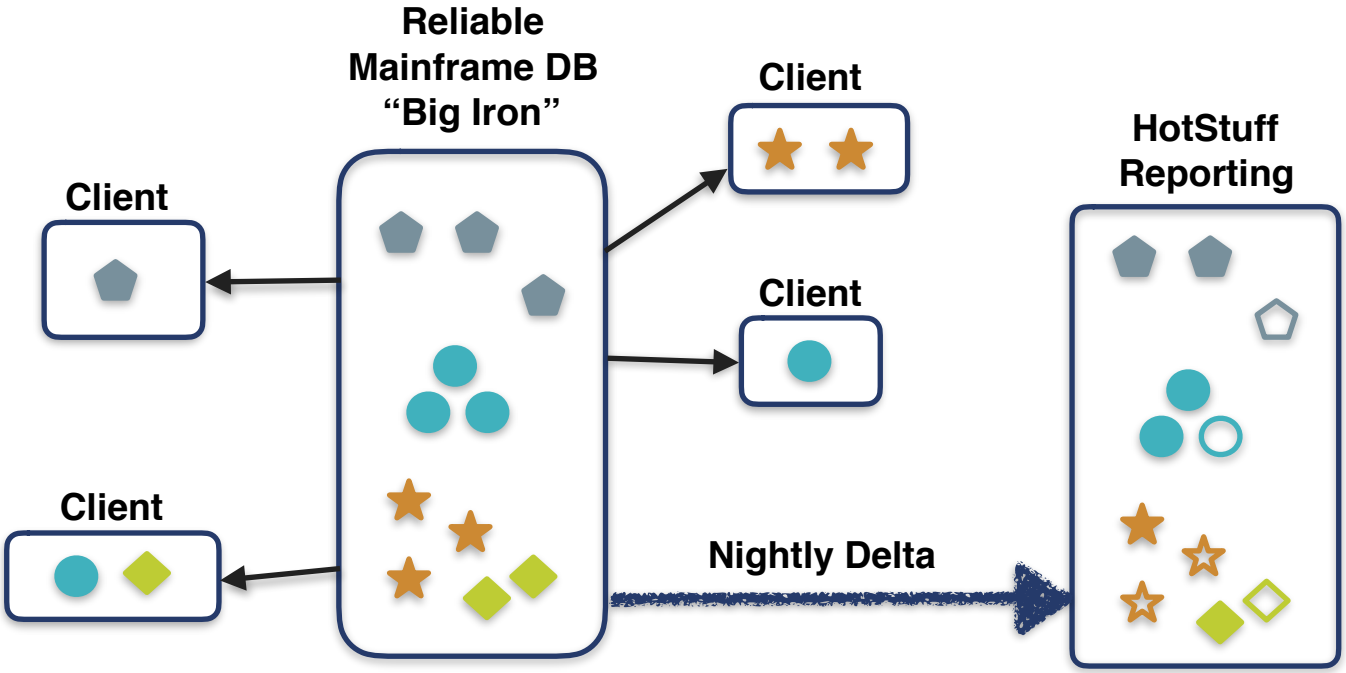
Reliable
Mainframe DB
“Big Iron”

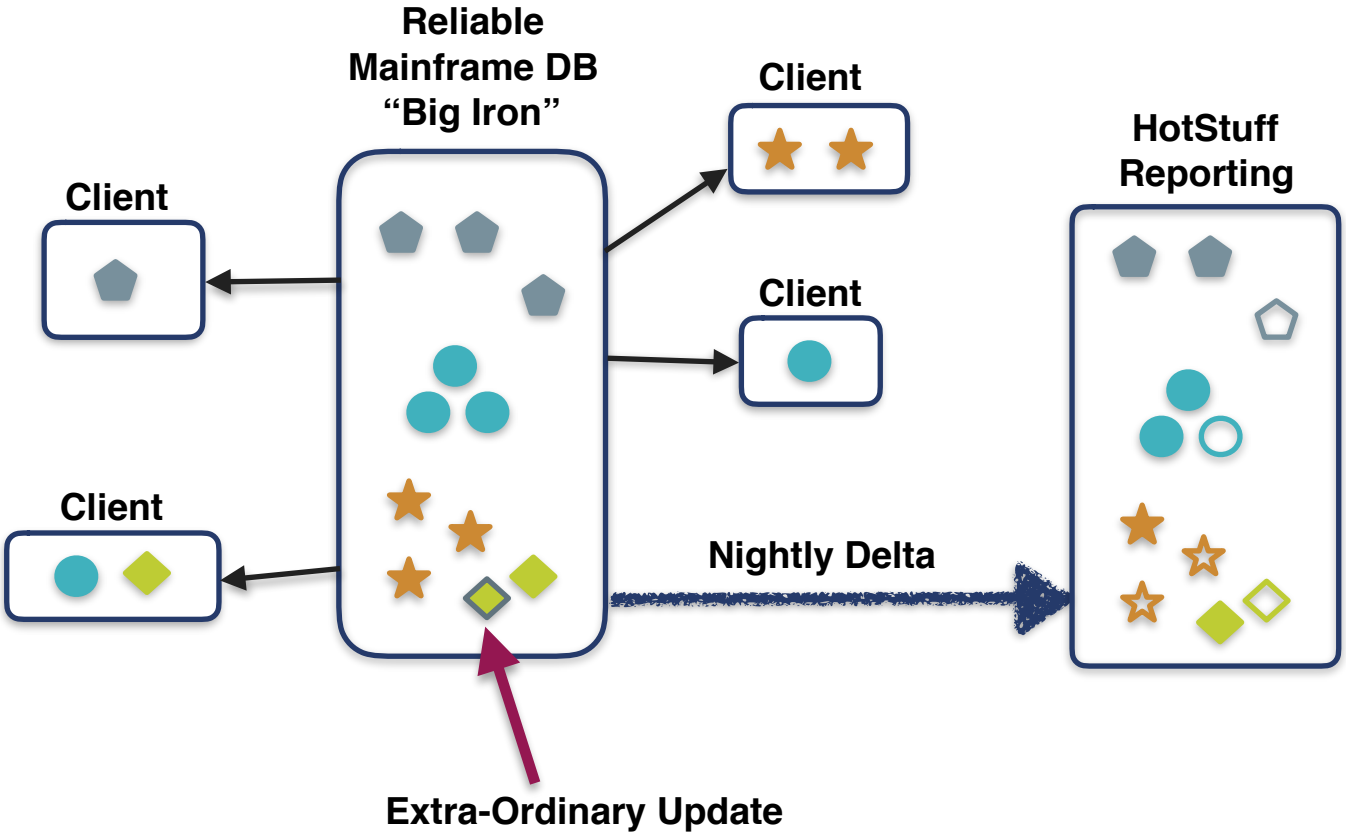


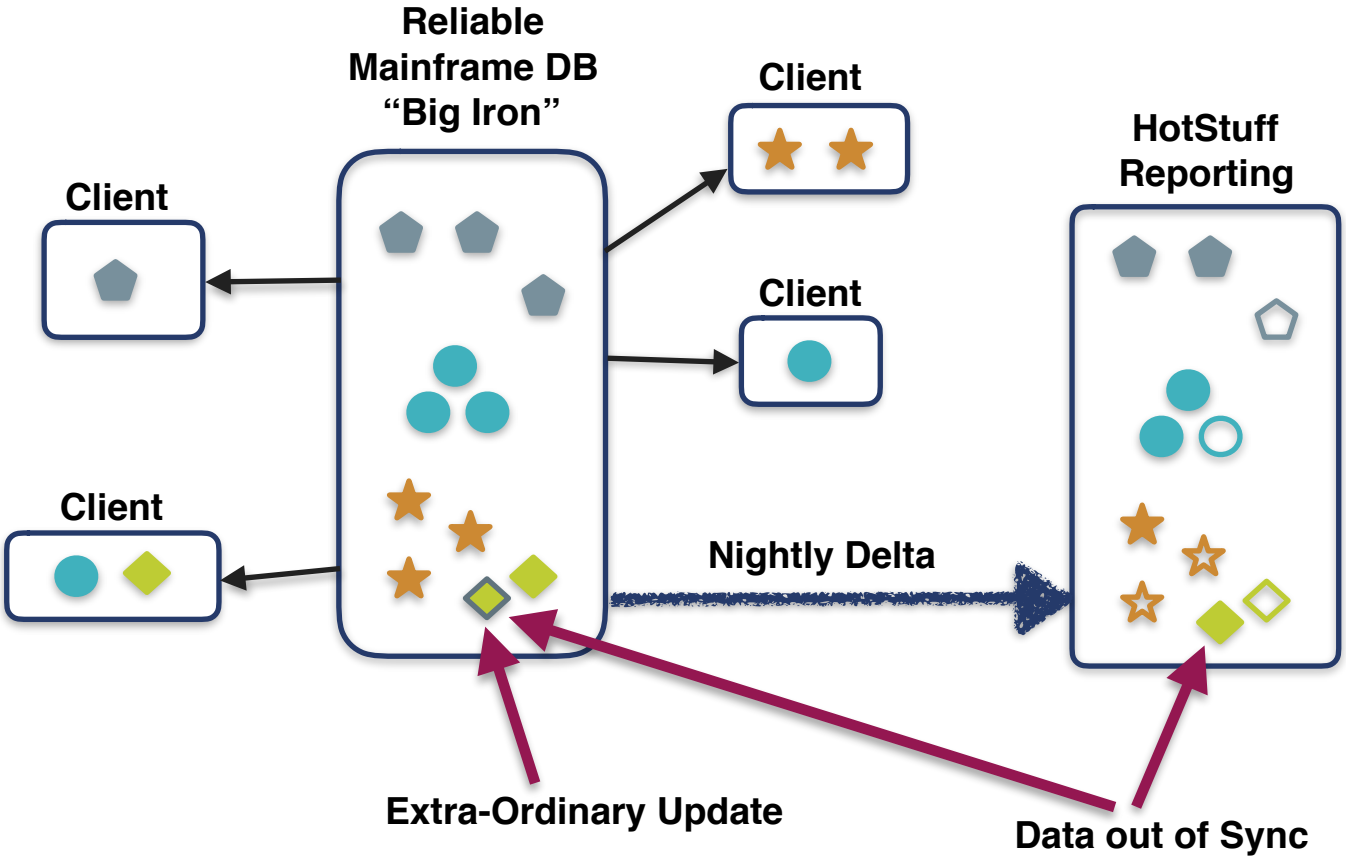

Reliable
Mainframe DB
“Big Iron”
Client
Client
Client
Client


Reliable
Mainframe DB
“Big Iron”
Client
Client
Client
Client
HotStuff
Reporting

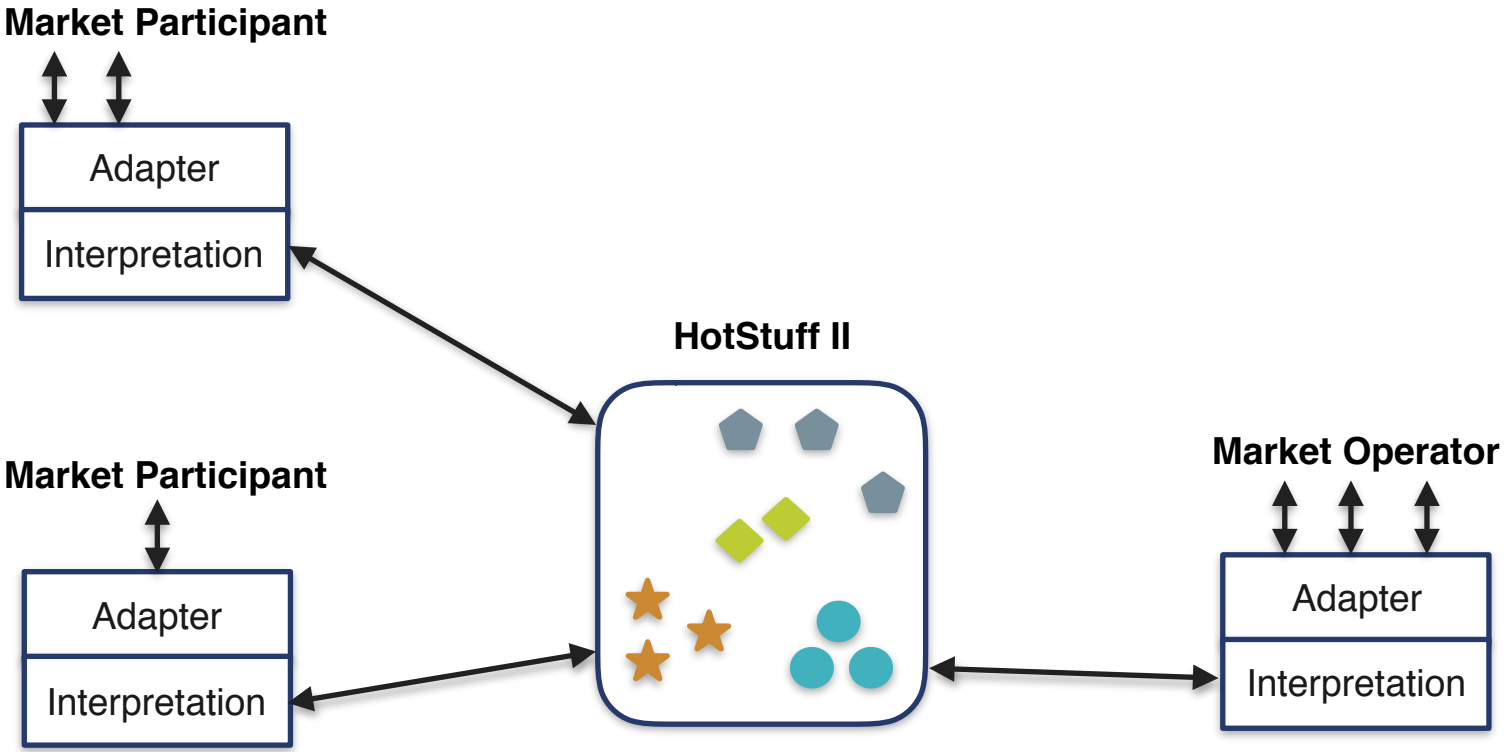

Reliable
Mainframe DB
“Big Iron”
Client
Client
Client
Client
HotStuff
Reporting


Reliable
Mainframe DB
“Big Iron”
Client
Client
Client
Client
HotStuff
Reporting
Nightly Delta


Reliable
Mainframe DB
“Big Iron”
Client
Client
Client
Client
HotStuff
Reporting
Nightly Delta
Extra-Ordinary Update

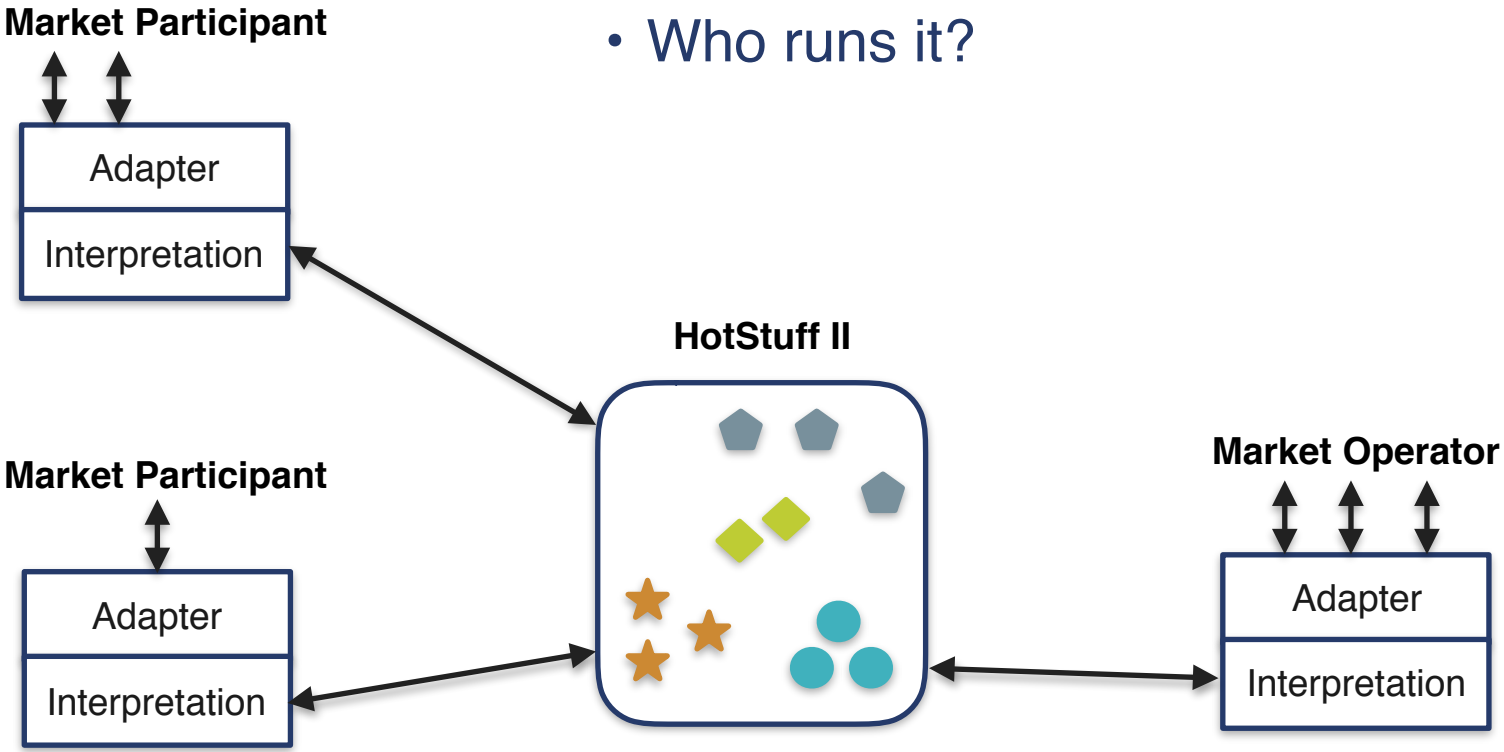

Reliable
Mainframe DB
“Big Iron”
Client
Client
Client
Client
HotStuff
Reporting
Nightly Delta
Extra-Ordinary Update
Data out of Sync

“Let’s just have one database?”



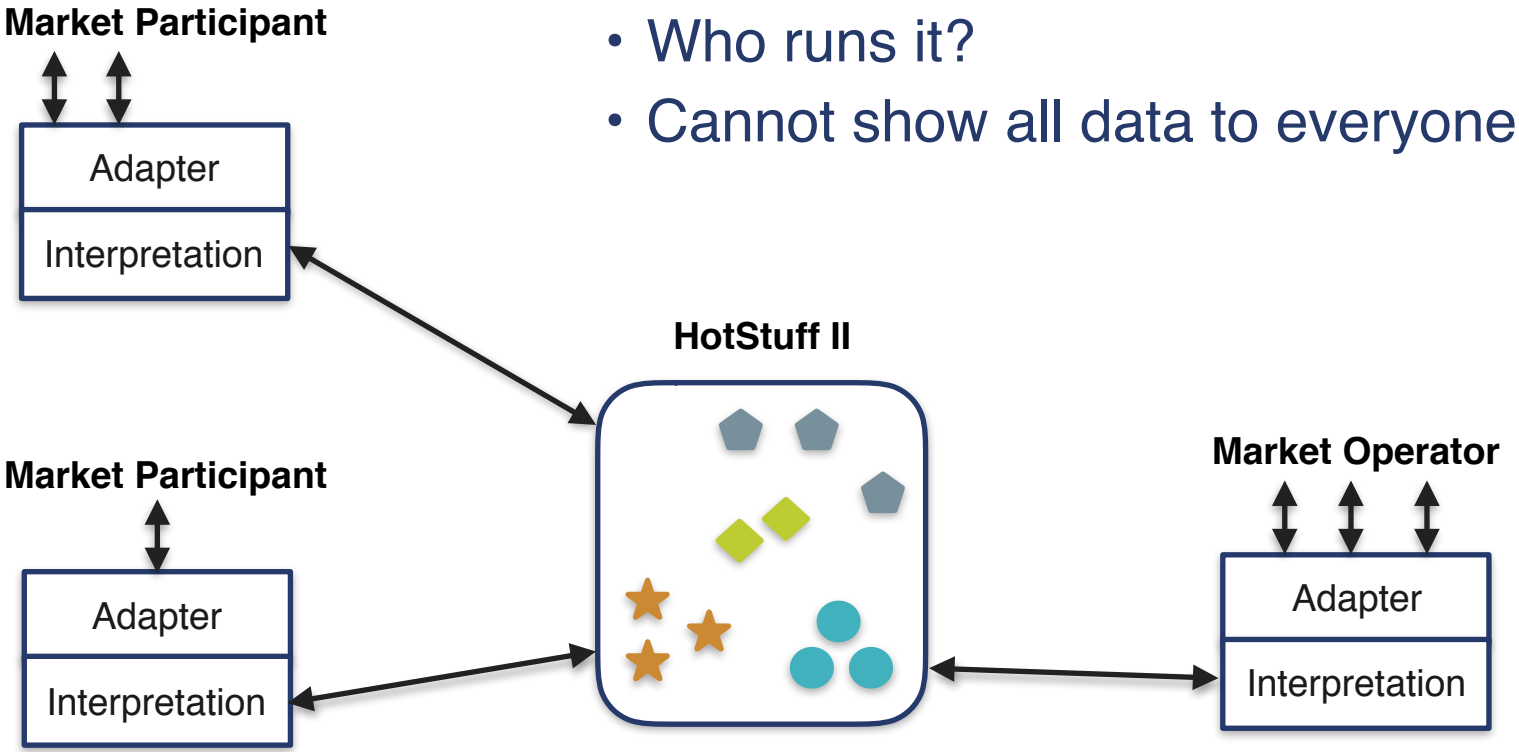
“Let’s just have one database?”

- Who runs it?



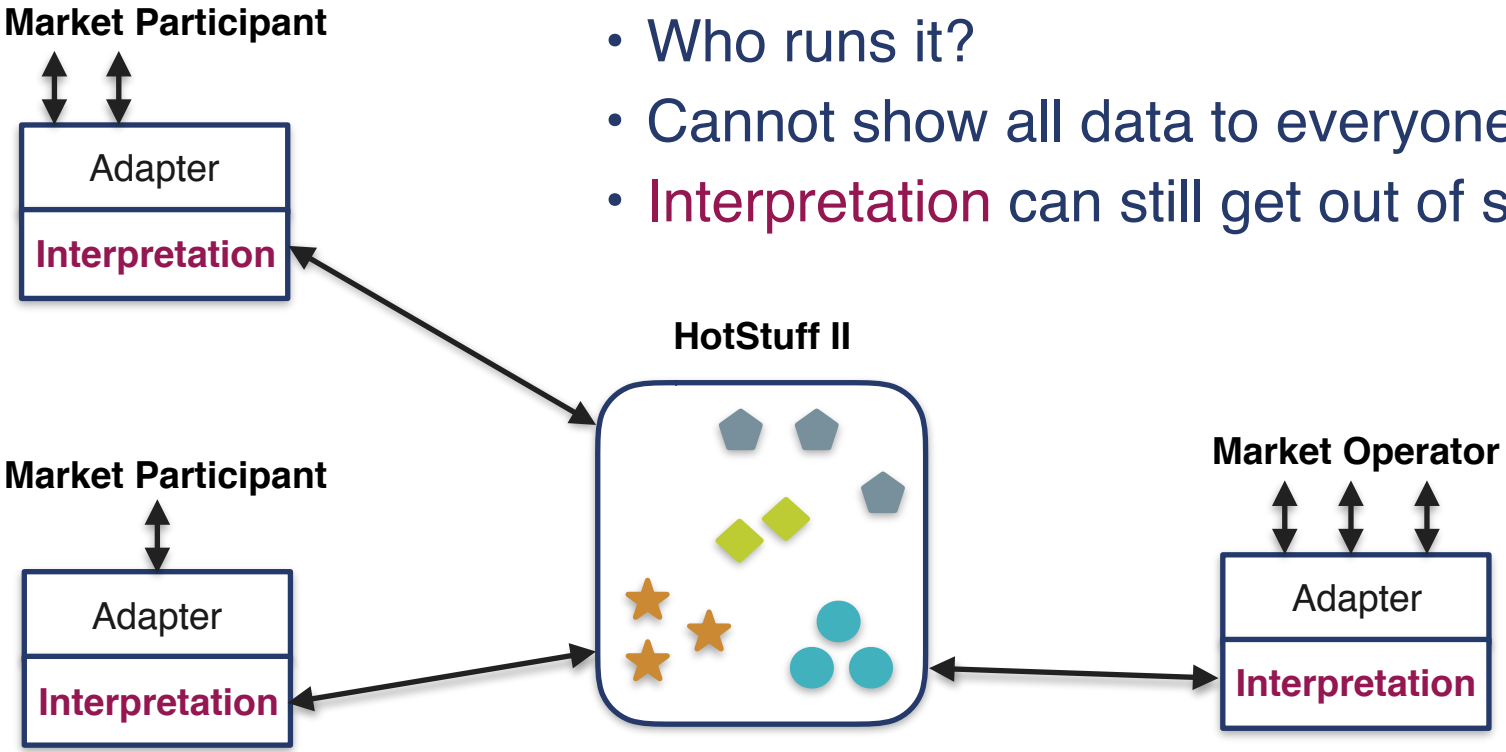
“Let’s just have one database?”

- Who runs it?
- Cannot show all data to everyone.

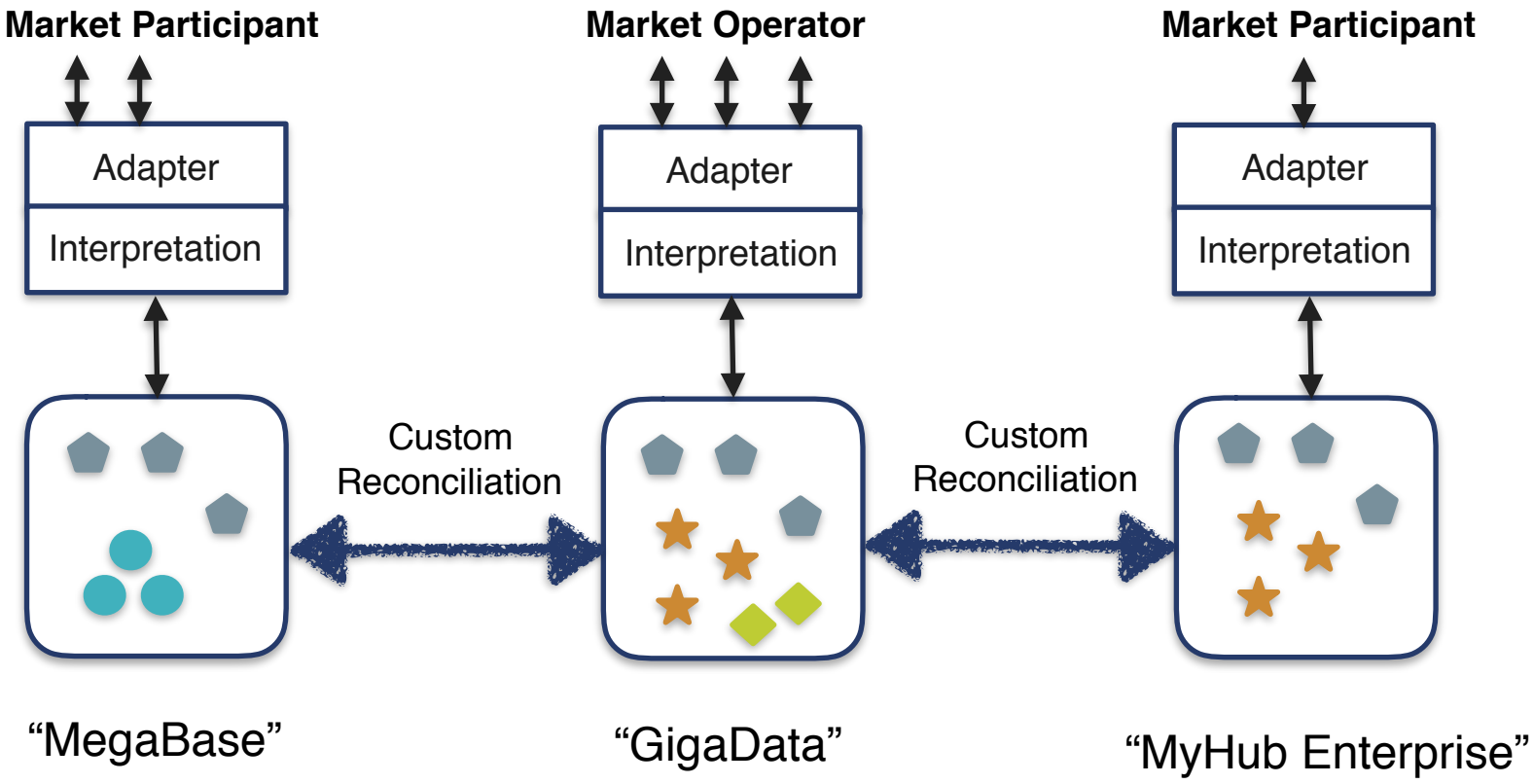


“Let’s just have one database?”

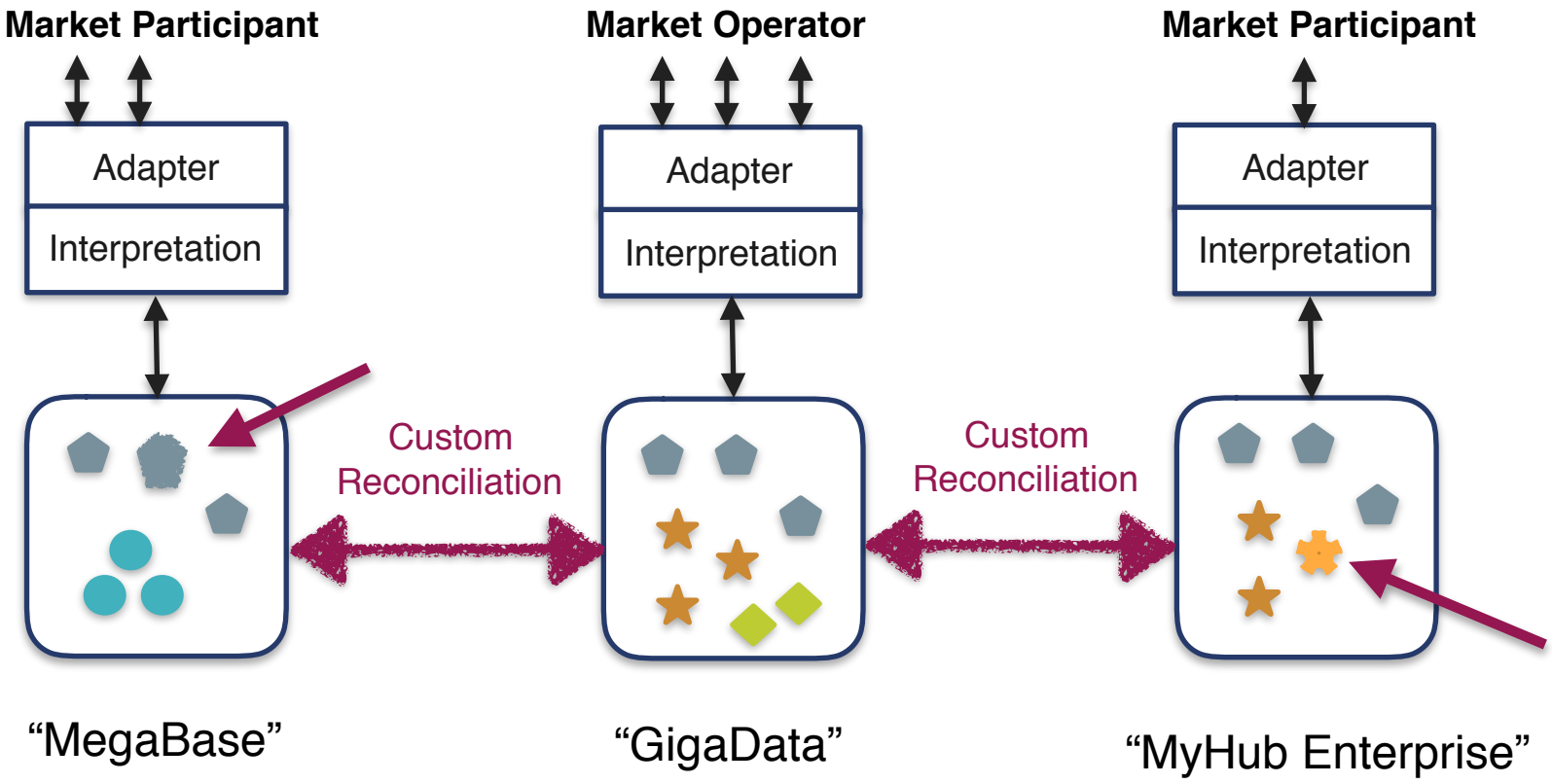
- Who runs it?
- Cannot show all data to everyone.
- Interpretation can still get out of sync.



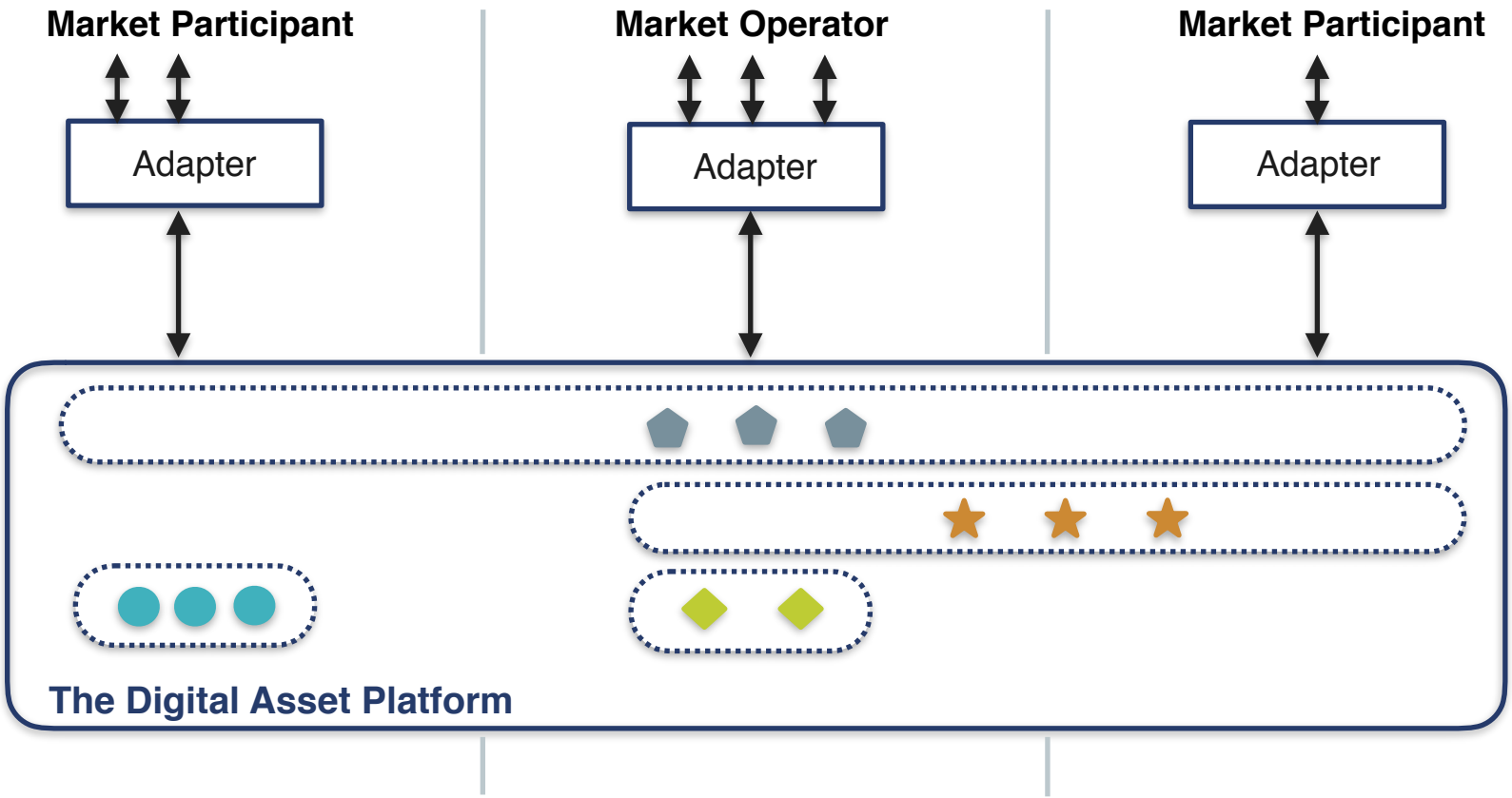
Siloed Data


Market Participant
Adapter
Interpretation
Market Operator
Adapter
Interpretation
Market Participant
Adapter
Interpretation
Custom
Reconciliation
Custom
Reconciliation
“MegaBase”
“GigaData”
“MyHub Enterprise”

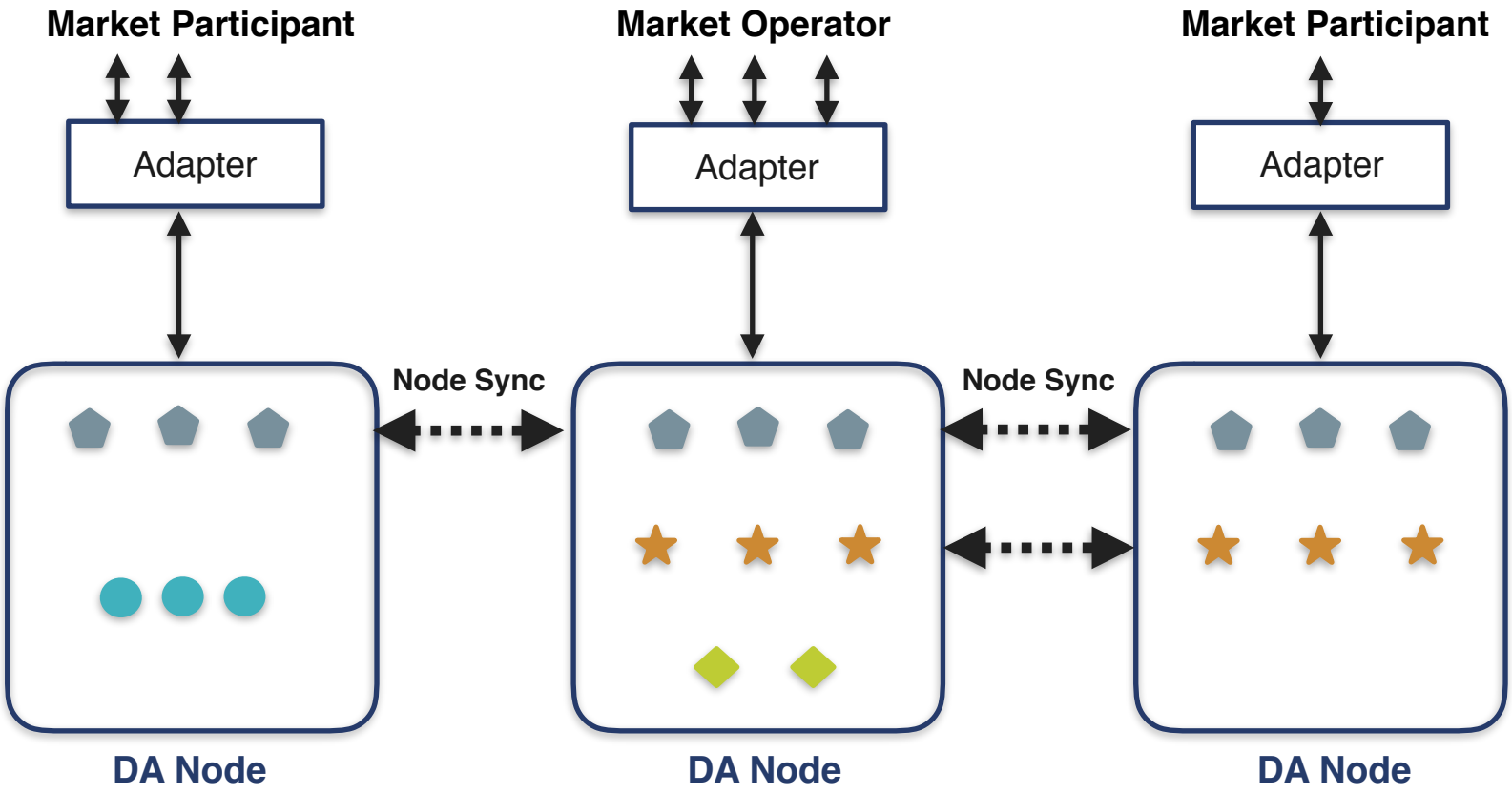
.. is Fractured Data



Distributed Permissioned Ledger (Logical View)



Distributed Permissioned Ledger (Physical View)

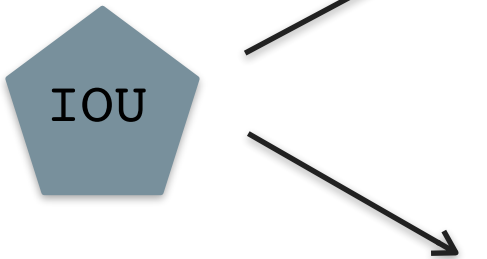


Smart Contracts

and the

Digital **A**sset **M**odelling **L**anguage **(DAML)**

Smart Contracts Provide Interpretation of Data



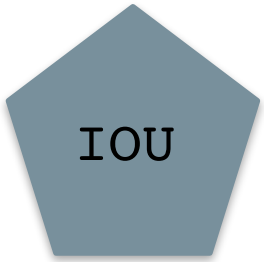
Contract Data (payload)

```
IOU “Alice” “Bob” 100 AUD
```

Contract Choices (methods)

```
call     (acct: AccountId) = …
transfer (party: PartyId)  = …
split    (amount: Decimal) = …
```

DAML: The Digital Asset Modelling Language

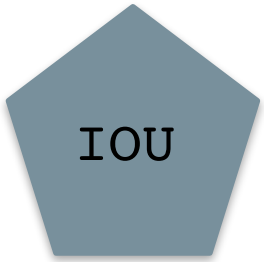



```
call(…)
transfer(…)
split(…)
```


```
buy(…)
sell(…)
distribute(…)
```

- The Rules of the Game are encoded as DAML programs.
- The Rules are distributed and kept in sync along with the data.
- Static program analysis ensures that only authorised parties can exercise contract choices.

DAML: The Digital Asset Modelling Language

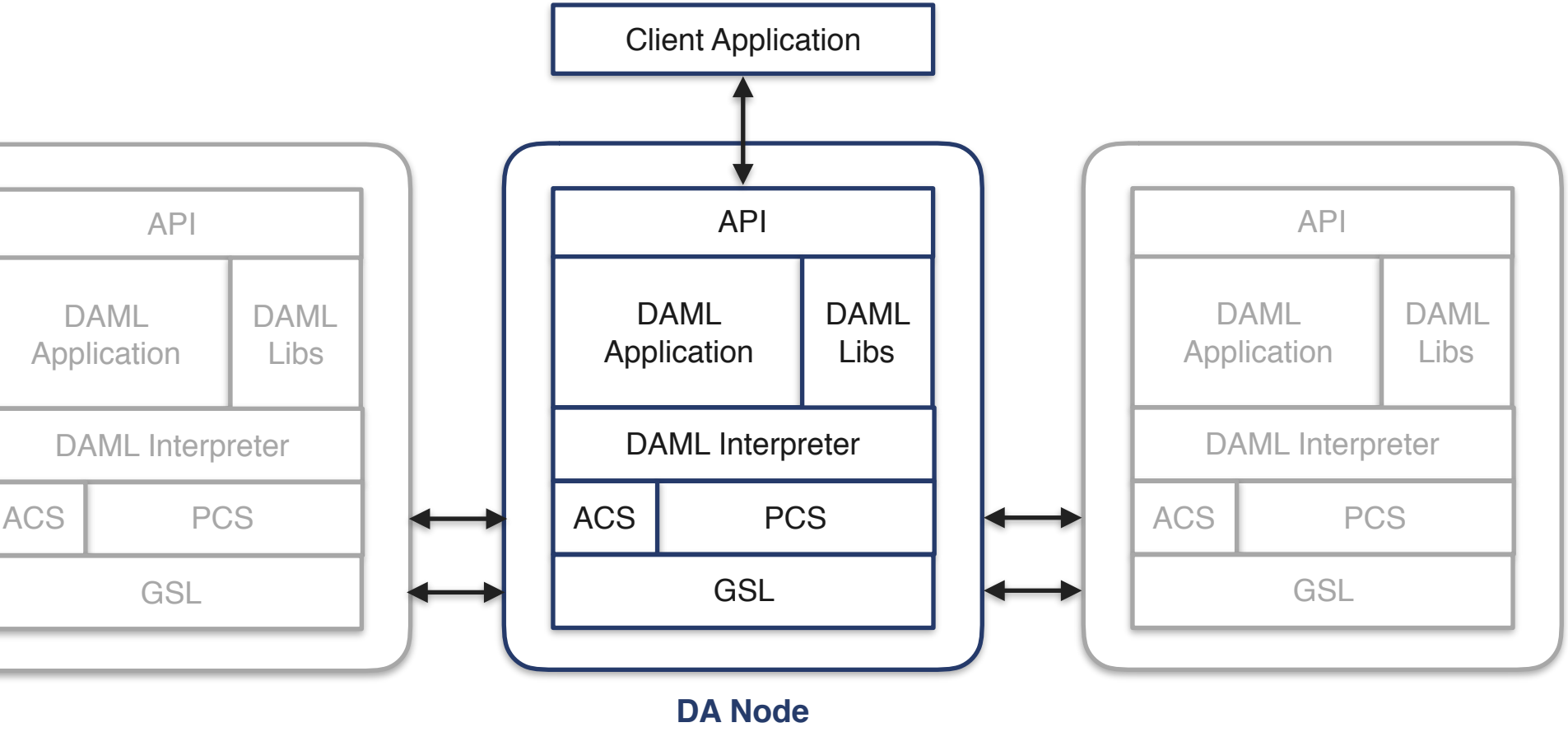



```
call(…)
transfer(…)
split(…)
```

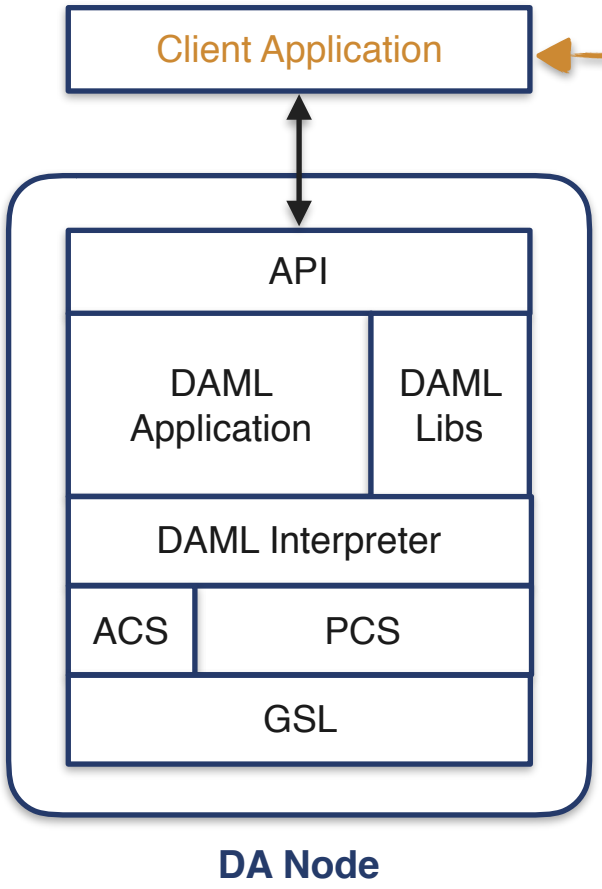

```
buy(…)
sell(…)
distribute(…)
```

- DAML is a typed functional language with UXTO semantics
- The only mutable state is the contract data.
- Exercising a choice on a contract consumes the contract.
- Common programming utilities, workflows and asset models are available as libraries (with no GAS cost, as in Ethereum)

Node Implementation


Client Application
API
DAML Application
DAML Libs
DAML Interpreter
ACS
PCS
GSL
API
DAML Application
DAML Libs
DAML Interpreter
ACS
PCS
GSL
API
DAML Application
DAML Libs
DAML Interpreter
ACS
PCS
GSL

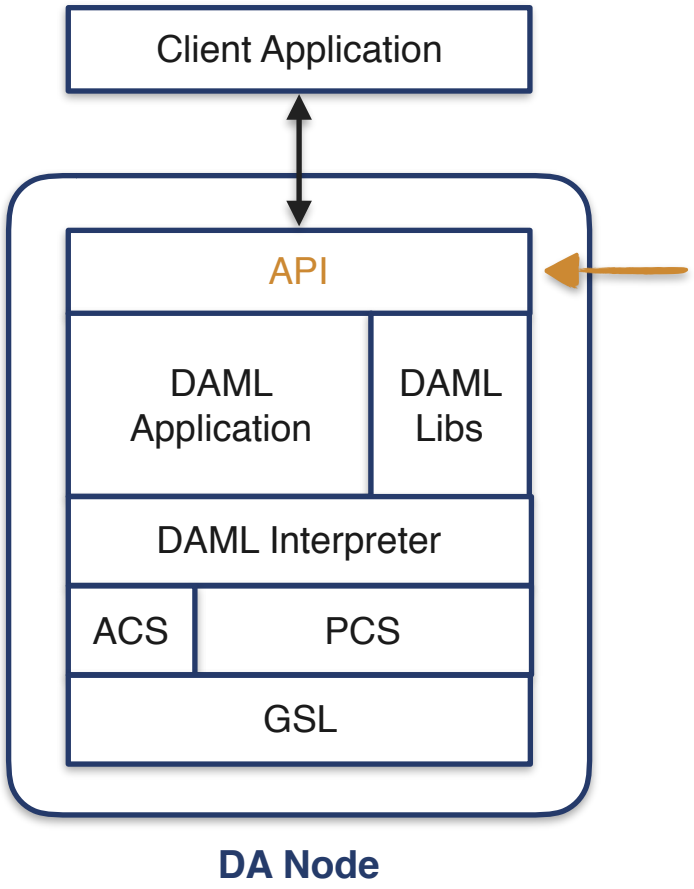
DA Node



Client Application

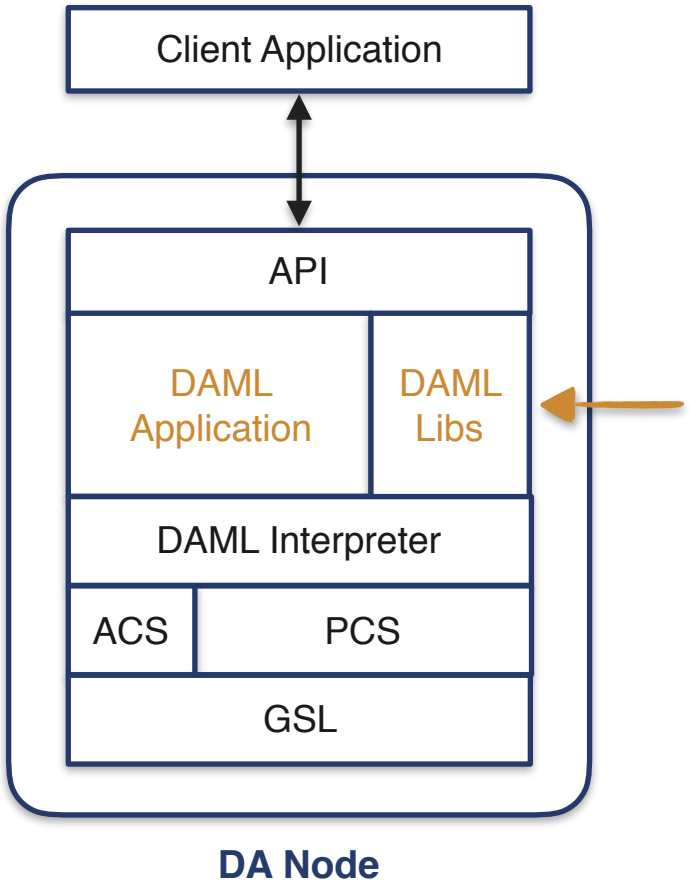
- Interfaces to Client Systems
- Talks to custom GUI's
- Runs the strategy.

The DAML contracts specify
The *Rules of the Game*
The Client App specifies
The *Strategy to Play it.*



App. Prog. Interface (API)

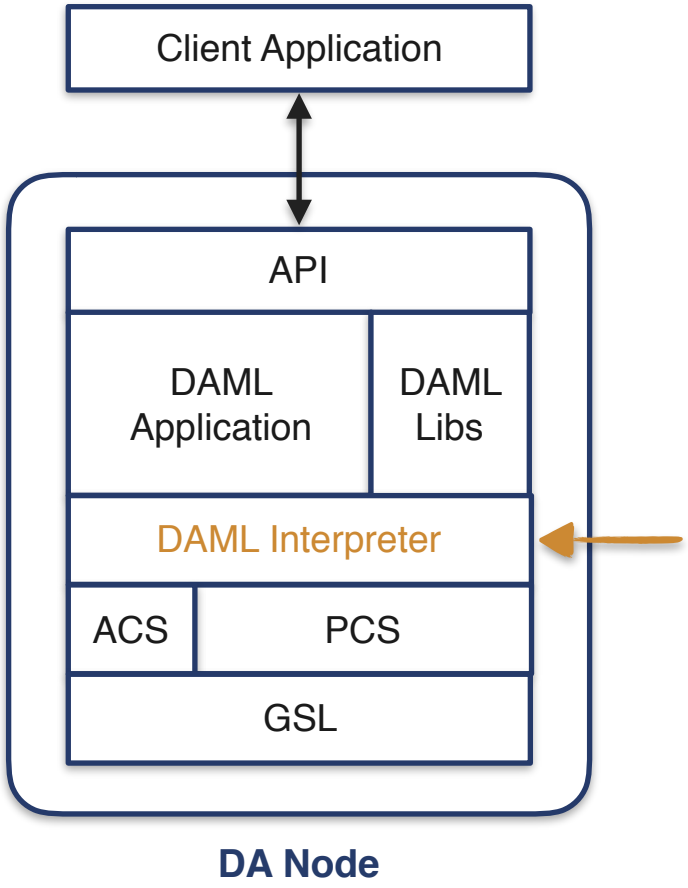
- Provides client an interface to create and exercise contracts.
- Can be as simple as a REST endpoint.



DAML Application and Libs

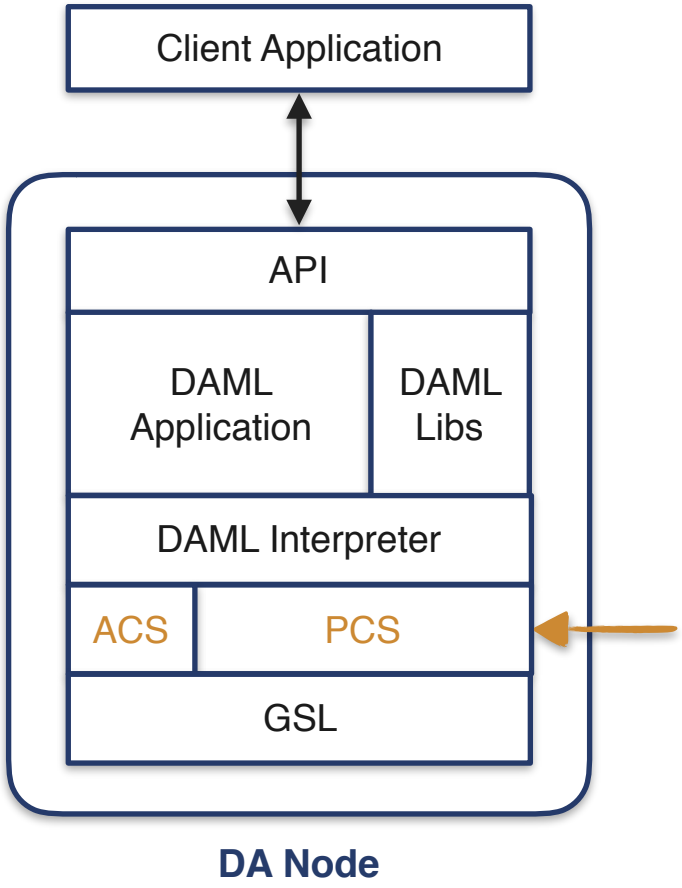
- Defines the rules of the game:
  Who can buy an asset?
  how can it be sold?
  can there be multiple owners?
  do owners receive dividends?

- Libraries provide common infrastructure and patterns.



DAML Interpreter

- Executes the DAML programs.
- Consumes data from the ACS (Active Contract Store).
- Checks that only authorised parties can exercise contract choices.

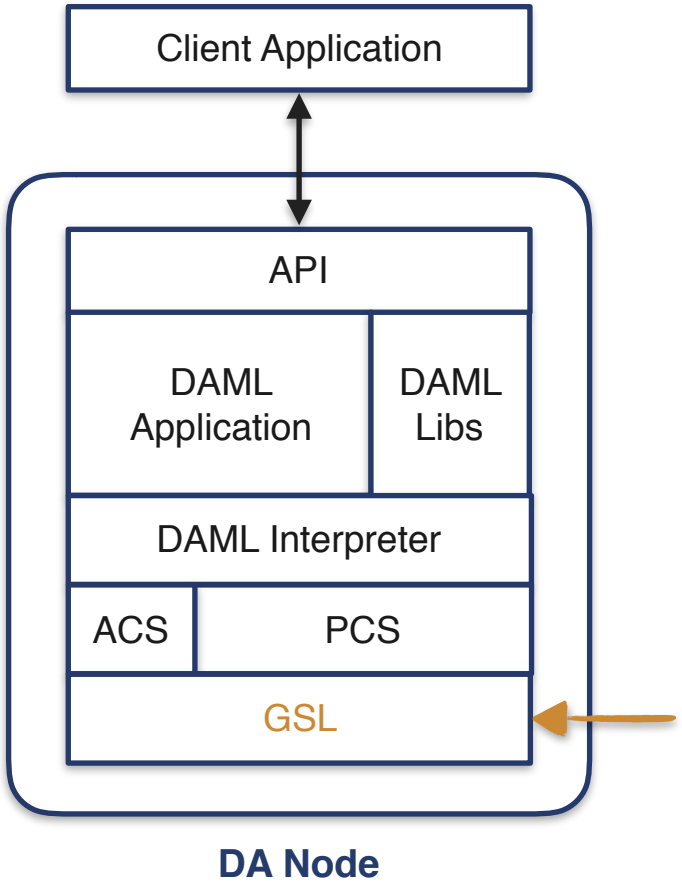


PCS Private Contract Store

- Holds all contracts that have been visible to the node at some point in time.

ACS Active Contract Store

- Holds the active contracts, the ones that are currently still in force and can have their choices exercised.



GSL Global Sync. Log

- Provides a global order of contract execution times.
- Provides blinded event notifications to nodes.
- Notifications are crypto. blinded to avoid leaking information via traffic and timing analysis.

Comparison

- Permissioned Ledger
- Physical data segregation
- “Trust but verify”
- Fewer nodes, high data rate
- No baked-in asset model, custom assets defined in code.
- Contracts written in DAML.

- Public Ledger
- All nodes can see all data
- Actively hostile environment
- Many nodes, lower data rate
- Baked in native asset (ETH), other assets defined in code.
- Contracts compiled to EVM.

Questions?