

Rich Queries

Fabric ledger

- **Transaction Log:** Stores the details of the transaction (the transaction logs/history) in a tamper-proof and sequential manner.
- **World state:** Stores the current state of the business object (asset).
 - ❖ **LevelDB** stores chaincode data as simple key-value pairs.
 - ❖ **CouchDB** is an optional, alternate state database that allows you to model data on the ledger as **JSON** and issue queries against data values rather than the keys.

Rich Queries

- **GetQueryResult**
- **GetStateByRange**
- **GetHistoryForKey**
- **GetQueryResultWithPagination**

Rich Queries

GetQueryResult API

we can perform JSON queries against the data in the state database by using the **GetQueryResult** API and passing a **CouchDB query string**

CouchDB query string

selector (*json*) – JSON object describing criteria used to select documents.

GetStateByRange

- ❖ Returns a range of data over a set of keys in the ledger
- ❖ The iterator can be used to iterate over all keys between the start key(inclusive) and end key (exclusive)

getStateByRange(start key, end key)

GetStateByRange

- getStateByRange(CAR-001, CAR-004)
- getStateByRange(CAR-005, " ")
- getStateByRange(" ", " ")

1	{	KEY	VALUE	}	3
		CAR-001	{values:}		
		CAR-002	{values:}		
		CAR-003	{values:}		
2	{	CAR-004	{values:}		
		CAR-005	{values:}		
		CAR-006	{values:}		
		CAR-007	{values:}		

Queries on Private Data

getPrivateDataQueryResult ()

getPrivateDataByRange()

GetHistoryForKey

- ❖ Retrieves the history for a state
- ❖ Returns the set of stored values, including the transaction identifiers that performed the state update
- ❖ For each historic key update, the historic value and associated transaction id and time stamp are returned

getHistoryForKey(Key)

GetQueryResultWithPagination

- It helps to list the records as different sets

Input Arguments:

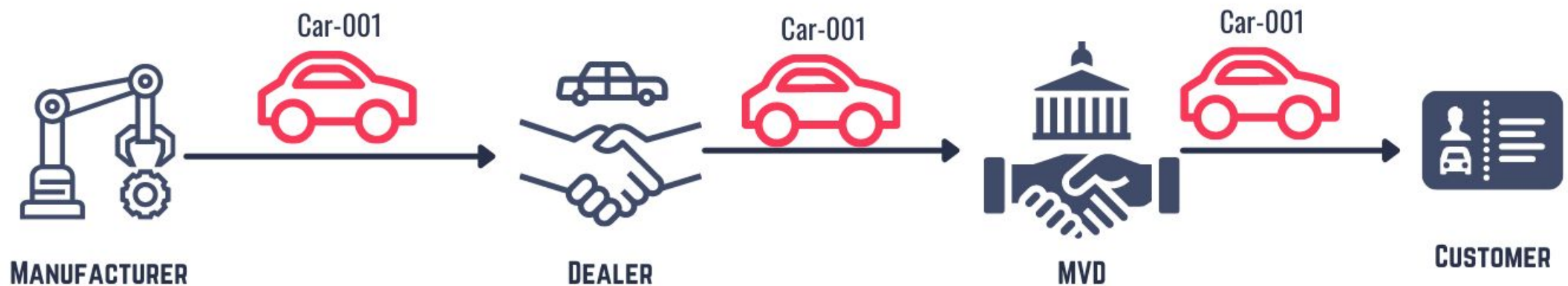
- Query: CouchDB query string
- Pagesize: number of records to be displayed on a single query
- Bookmark: starting point for next set of records

Output:

- Iterator: query result set
- Metadata: fetched Records Count & bookmark

getQueryResultWithPagination (queryString,pageSize,bookmark)

Automobile usecase - Chaincode



1. List matching orders

- ❖ Here we obtain all the orders matching to a car

checkMatchingOrders(ctx, carId)

2. Assigning to dealer

- ❖ Here we assign a car to the dealer based on the order
- ❖ The order is removed from the private data collection

matchOrder(ctx, carId, orderId)

3. Registering Car

- ❖ Register a car to the customer
- ❖ It can be performed by the MVD only
- ❖ Status and ownedBy get updated
- ❖ Car gets registered by a registration number

registerCar(ctx, carId, ownerName, registrationNumber)

THANK YOU

