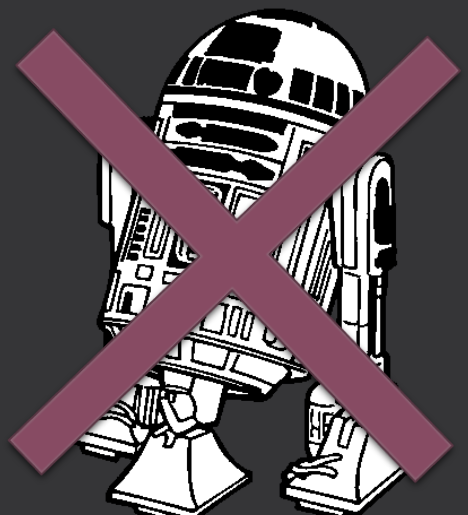
# P2B2

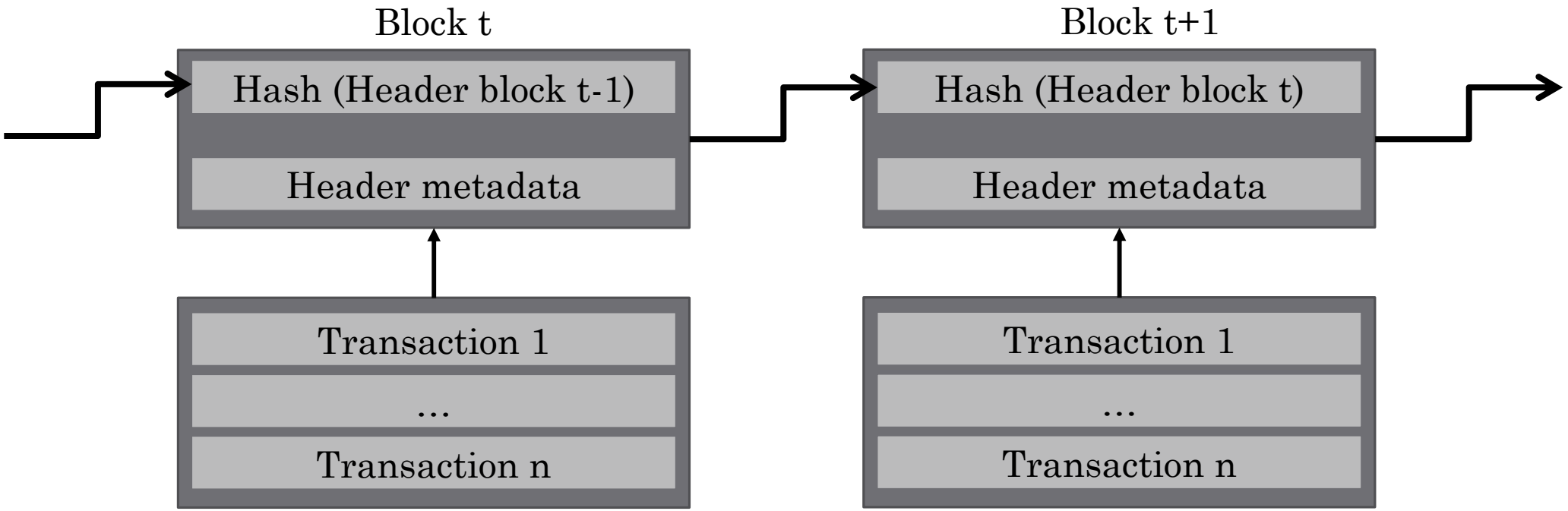
## Polyglot Persistence based Blockchain Analytics

https://github.com/p2b2



# What is Blockchain?

- Blockchain is a distributed P2P database system
- Entries are „blocks“, each chained with the previous block



# What is Ethereum?

- Blockchain ≠ Bitcoin
- Ethereum uses the concept of blockchain to run “smart contracts”
- Smart contracts are pieces of code that run on every peer
- More than just a currency exchange

# Ethereum Blockchain

The Blockchain is basically a database itself, so why would we populate its' data to another database?

- The ethereum blockchain is stored in the `chaindata`
  - Currently, depending on the implementation, the Ethereum Blockchain is about 30GB
  - Optimized for validating transactions and maintaining a valid state
  - **NOT good for historical lookups or analytics**

# Our plan

1. Populate all the blocks (including transactions) to document store and Graph-DB
2. Conduct different analytics on the two databases
3. Put the results to a Key-Value Store

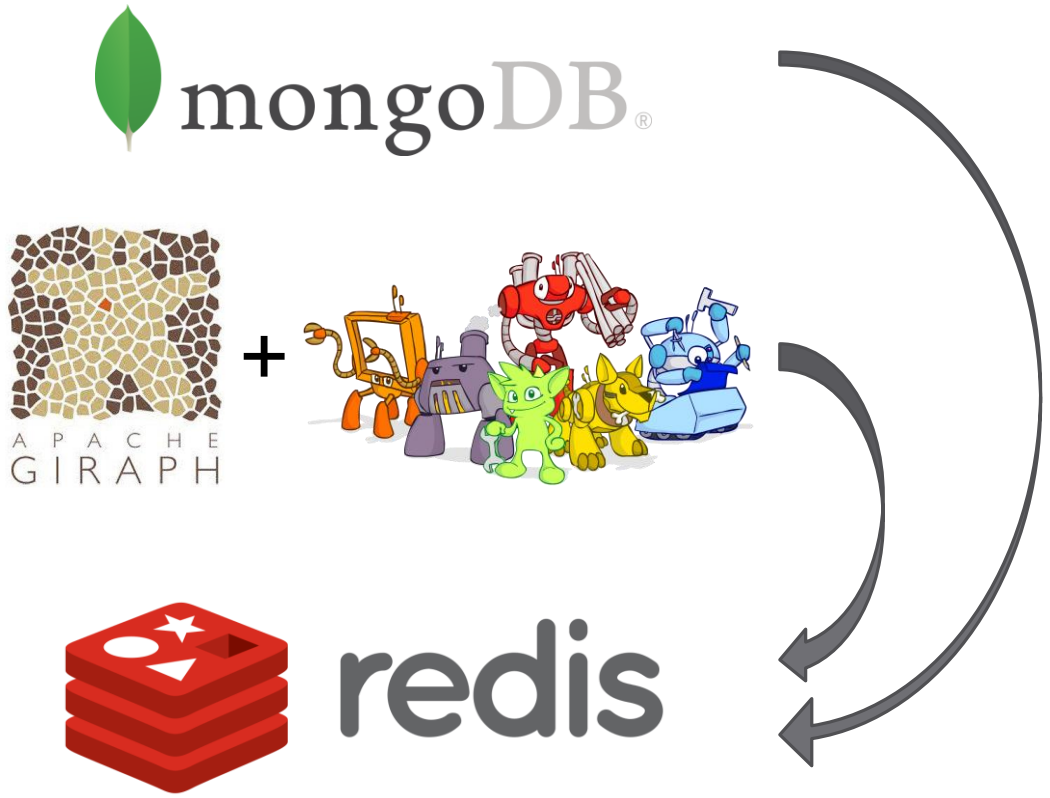
# Scenarios

- Quick lookups of any transaction
  - What are the most used contracts?
  - How high are transaction fees in average?
  - How is the ether distributed?

- All transactions done by an account or contract
  - Given an account address, who sent transactions to that account?
  - Are there different clusters, where most of transactions are done?
  - What are the most important accounts or contracts in the network?

# Technology stack

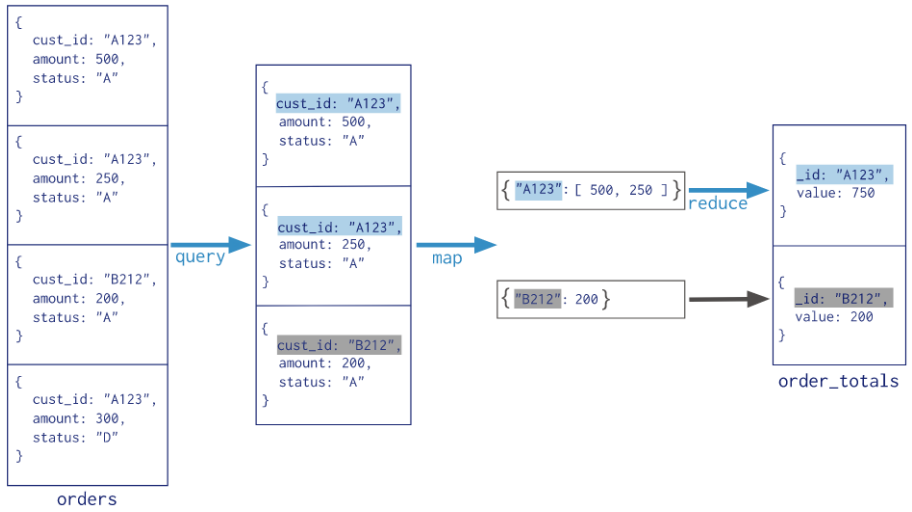
- Analytics: Document Store
  - MongoDB
  - MapReduce
- Analytics: Graph-DB
  - Apache Giraph
  - TinkerPop Graph Computing framework
- Results: Key-Value Store
  - Caching results in Redis



# Example - MapReduce

- For each ethereum address, we want to know the number of transactions submitted
  - Map a complex transaction object to its sender address
  - Reduce the addresses to the number of their occurences

- For a given ethereum address, get the total amount of currency sent to / received from other addresses

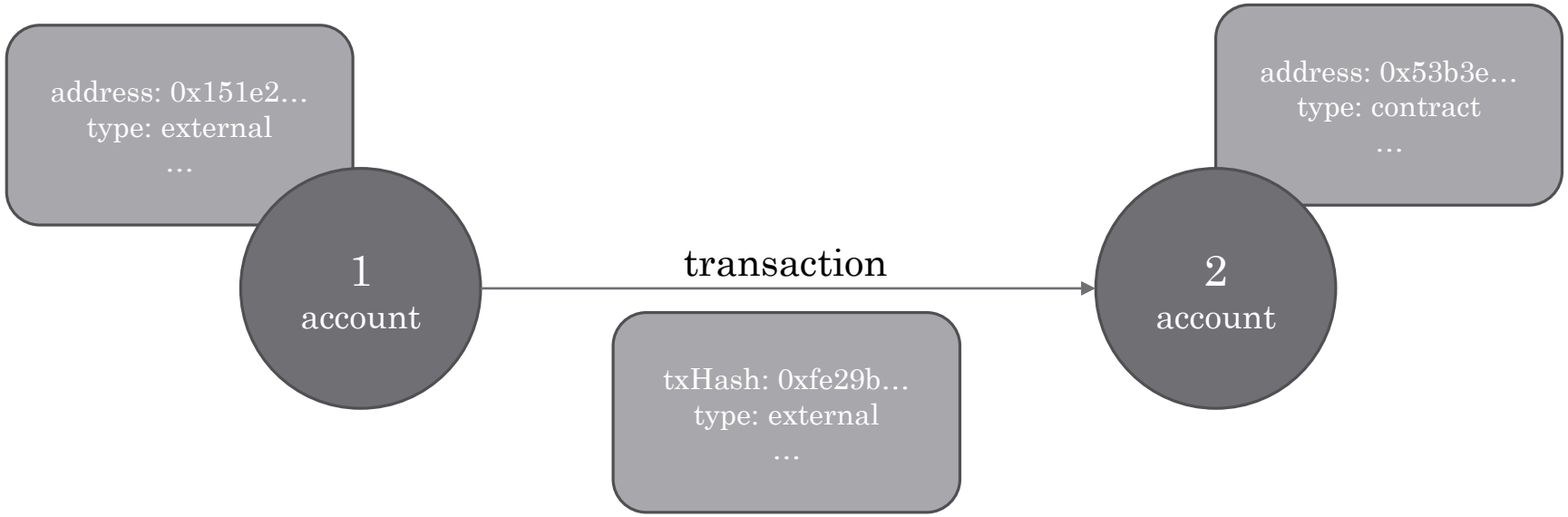



MapReduce example

Source: https://docs.mongodb.com/manual/core/map-reduce/

# Example - Graph DB

- Representation of the Ethereum blockchain in a Graph DB:
  - Vertex: Accounts
  - Edges: Transactions
- For a given account, find all accounts to which it has ever sent a transaction to:
  - Go to the vertex with the given public key and follow all the transaction edges to the vertexes
  - TinkerPop Graph Traversal: g.V().has('address', 0x151e201b90f8…').out('transaction').inV()



# Questions?