The columnar roadmap:

Apache Parquet and Apache Arrow

Julien Le Dem, Principal Architect Dremio,
VP Apache Parquet, Apache Arrow PMC

dremio

Julien Le Dem

@J_

- Architect at @DremioHQ dremio
- Formerly Tech Lead at Twitter on Data Platforms.
- Creator of Parquet
- Apache member
- Apache PMCs: Arrow, Kudu, Incubator, Pig, Parquet

Agenda

- Community Driven Standard
- Benefits of Columnar representation
- Vertical integration: Parquet to Arrow
- Arrow based communication

Community Driven Standard

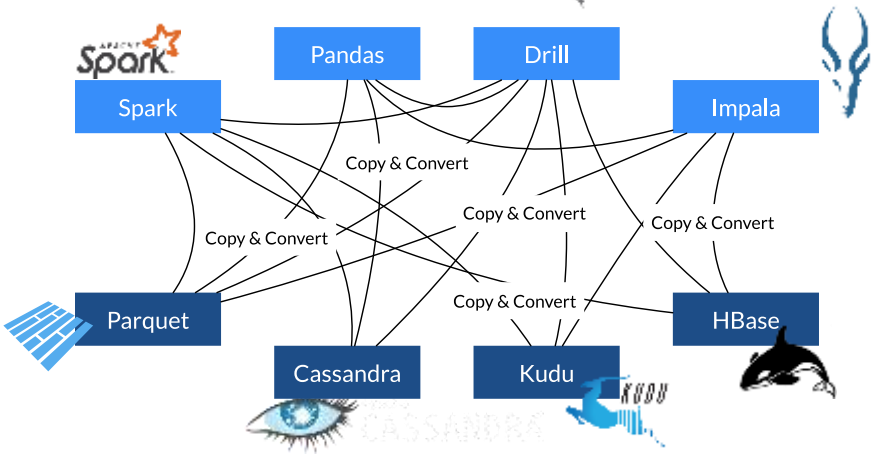
An open source standard

- **Parquet**: Common need for **on disk** columnar.
- **Arrow**: Common need for **in memory** columnar.
- Arrow is building on the success of Parquet.
- Top-level Apache project
- Standard from the start:
  - Members from 13+ major open source projects involved
- Benefits:
  - Share the effort
  - Create an ecosystem

Calcite
Cassandra
Deeplearning4j
Drill
Hadoop
HBase
Ibis
Impala
Kudu
Pandas
Parquet
Phoenix
Spark
Storm
R

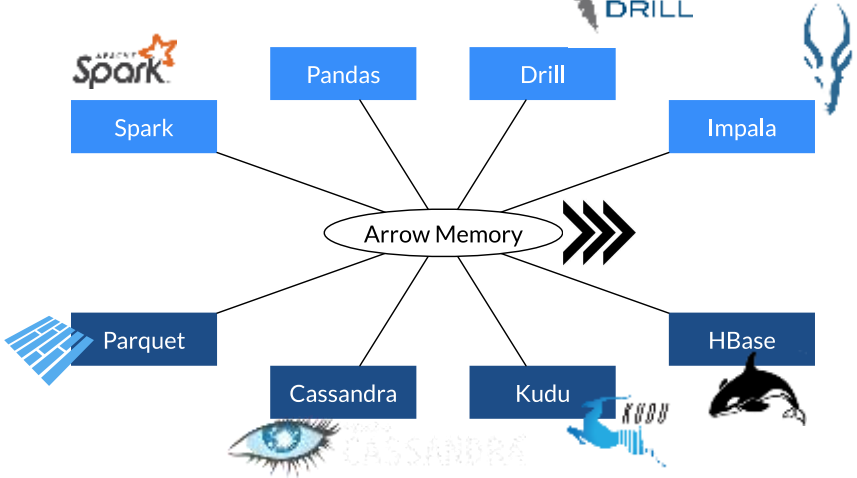
Interoperability and Ecosystem

Before

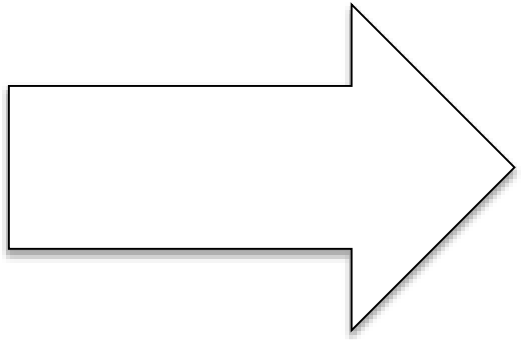


- Each system has its own internal memory format
- 70-80% CPU wasted on serialization and deserialization
- Functionality duplication and unnecessary conversions

With Arrow



- All systems utilize the same memory format
- No overhead for cross-system communication
- Projects can share functionality (eg: Parquet-to-Arrow reader)



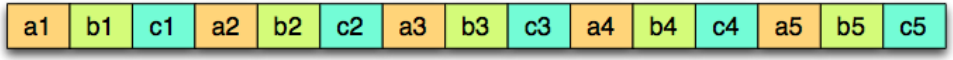
Benefits of Columnar representation

Columnar layout

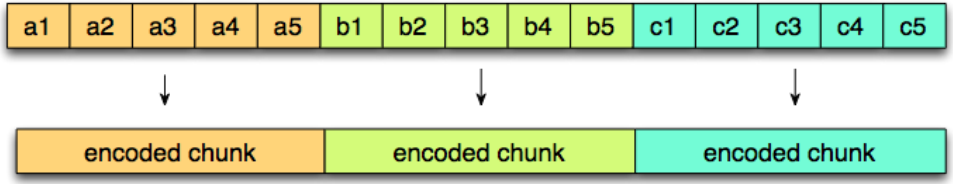
Logical table representation

a	b	c
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c4
a5	b5	c5

Row layout



Column layout



@EmrgencyKittens

On Disk and in Memory

- Different trade offs
  - On disk: Storage.
    - Accessed by multiple queries.
    - Priority to I/O reduction (but still needs good CPU throughput).
    - Mostly Streaming access.
  - In memory: Transient.
    - Specific to one query execution.
    - Priority to CPU throughput (but still needs good I/O).
    - Streaming and Random access.

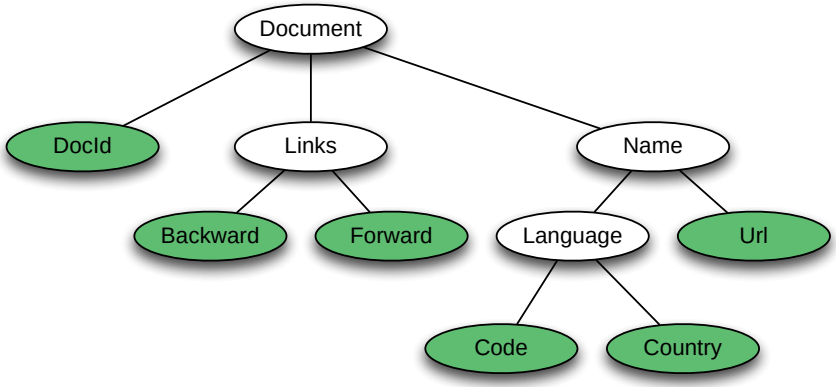
Parquet on disk columnar format

Parquet on disk columnar format

- Nested data structures
- Compact format:
  - type aware encodings
  - better compression
- Optimized I/O:
  - Projection push down (column pruning)
  - Predicate push down (filters based on stats)

Parquet nested representation

Borrowed from the Google Dremel paper



Columns:
docid
links.backward
links.forward
name.language.code
name.language.country
name.url

https://blog.twitter.com/2013/dremel-made-simple-with-parquet

Access only the data you need

Columnar

a	b	c
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c4
a5	b5	c5

+

Statistics

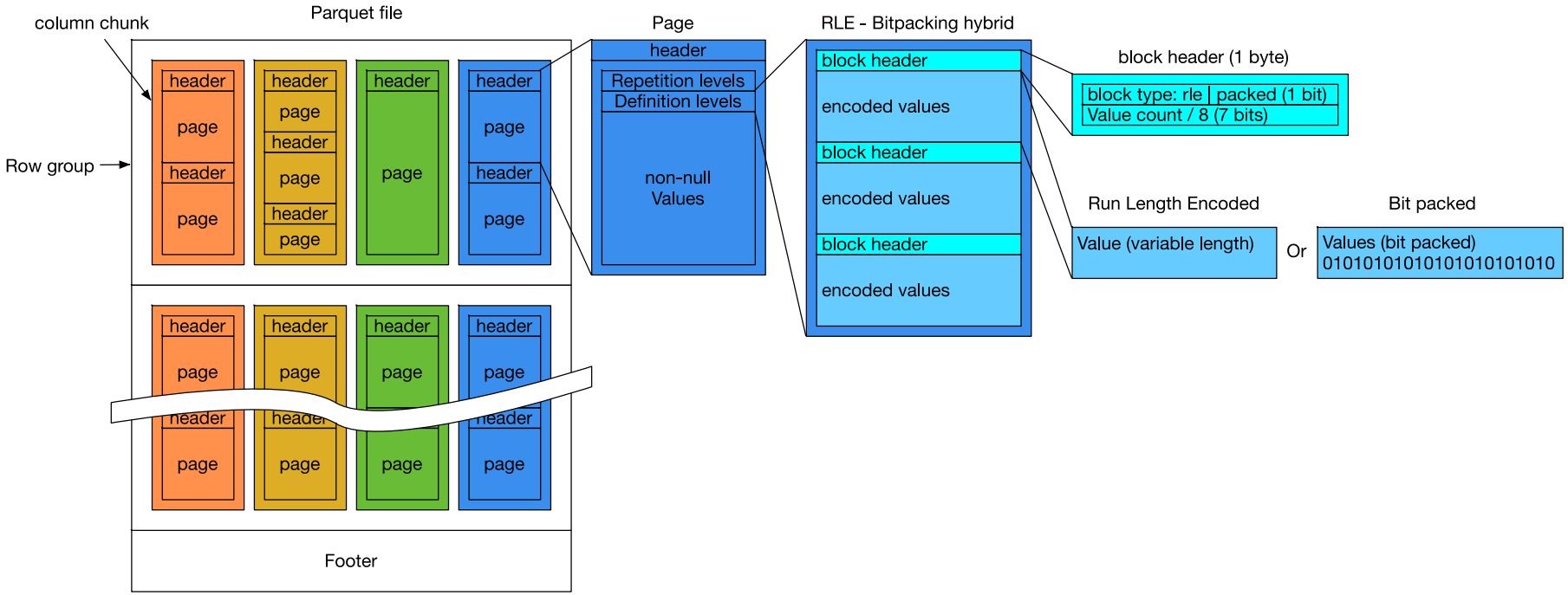
a	b	c
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c4
a5	b5	c5

=

Read only the data you need!

a	b	c
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c4
a5	b5	c5

Parquet file layout



Arrow in memory columnar format

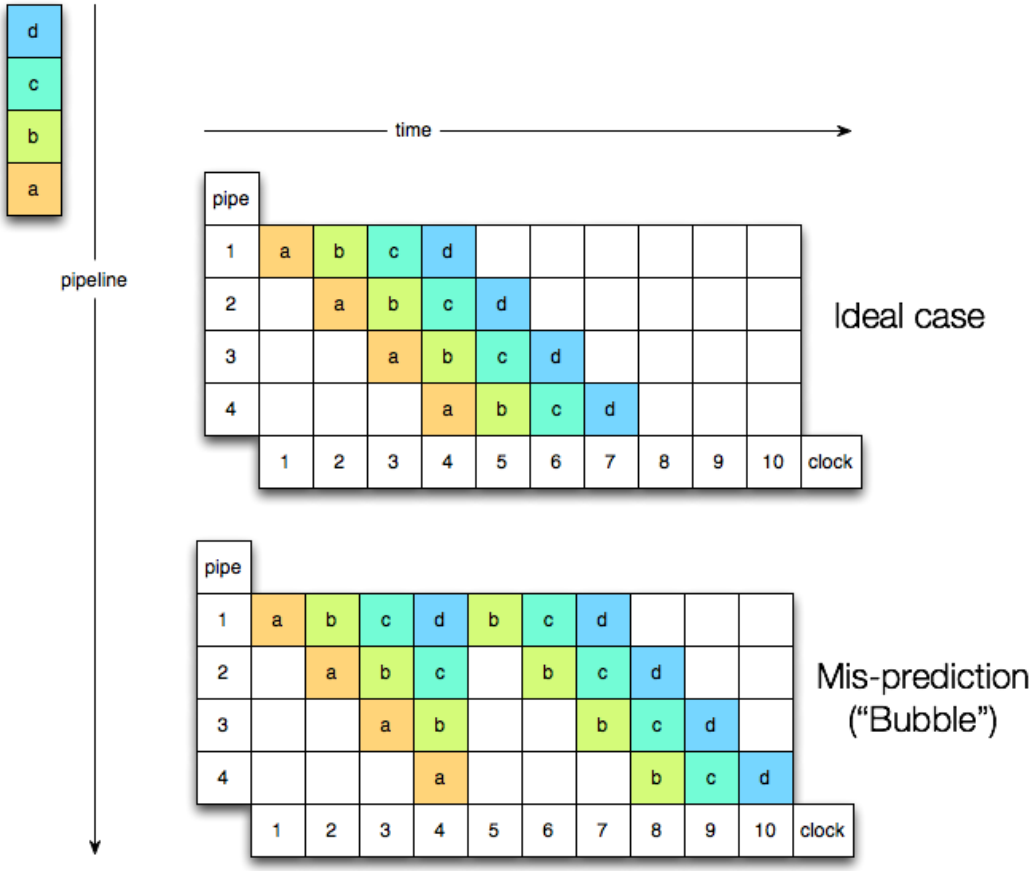
Arrow goals

- Well-documented and cross language compatible
- Designed to take advantage of modern CPU
- Embeddable
  - in execution engines, storage layers, etc.
- Interoperable

Arrow in memory columnar format

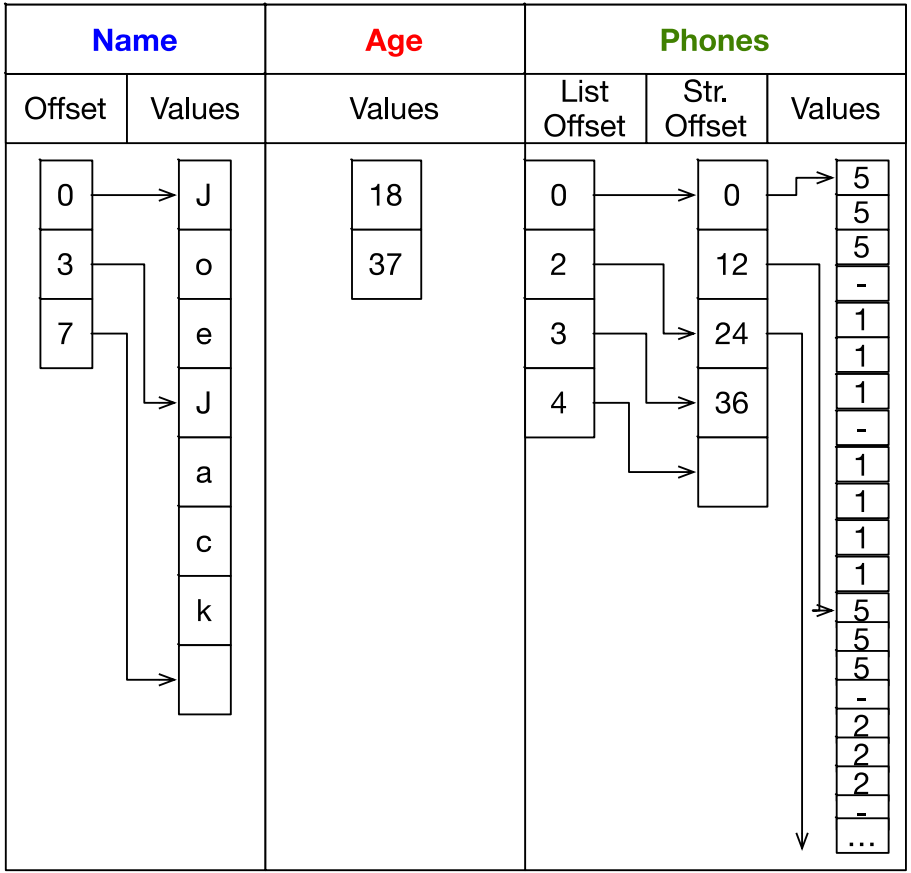
- Nested Data Structures
- Maximize CPU throughput
  - Pipelining
  - SIMD
  - cache locality
- Scatter/gather I/O

CPU pipeline

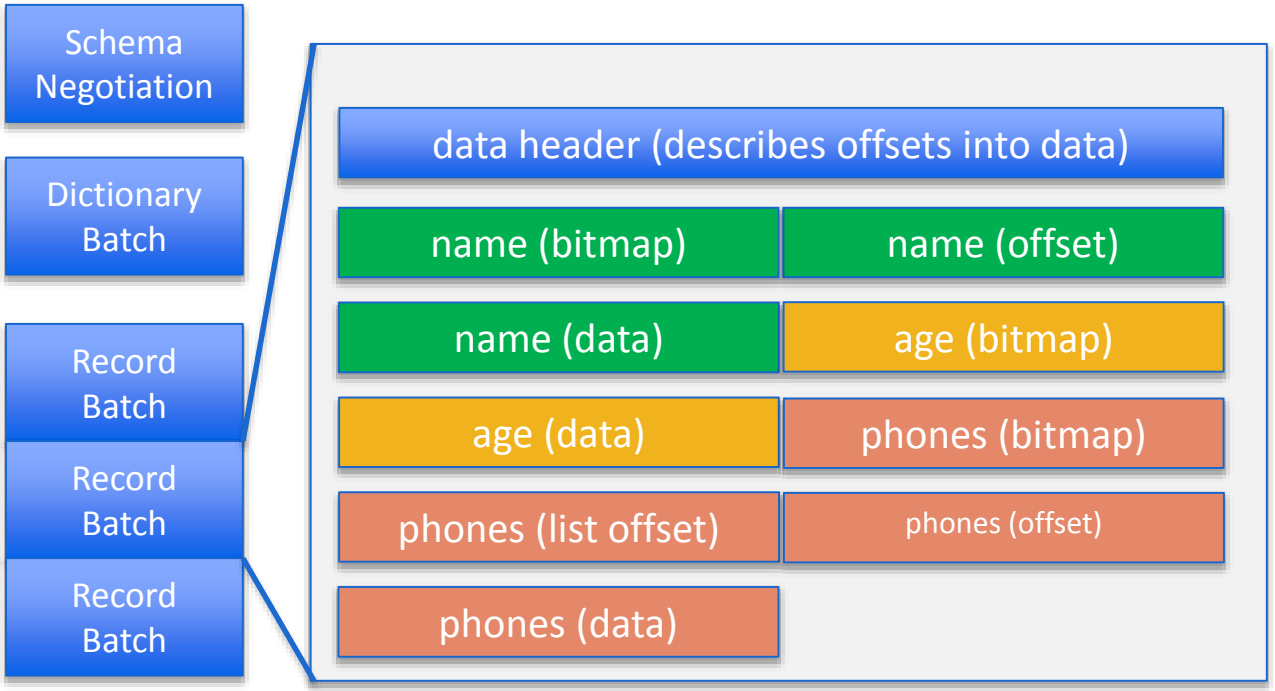


Columnar data

```
persons = [{
    name: ’Joe',
    age: 18,
    phones: [
        ‘555-111-1111’,
        ‘555-222-2222’
    ]
}, {
    name: ’Jack',
    age: 37,
    phones: [ ‘555-333-3333’ ]
}]
```



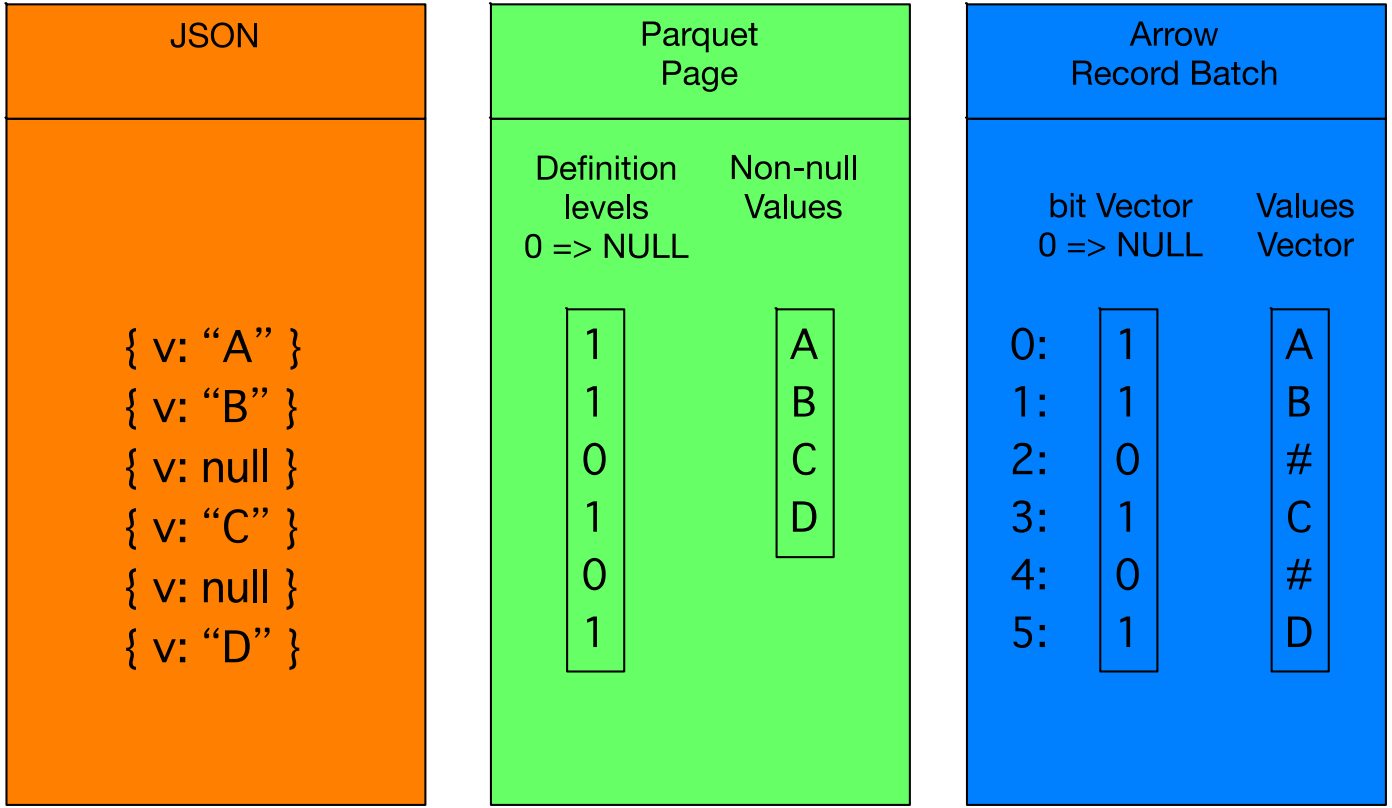
Record Batch Construction



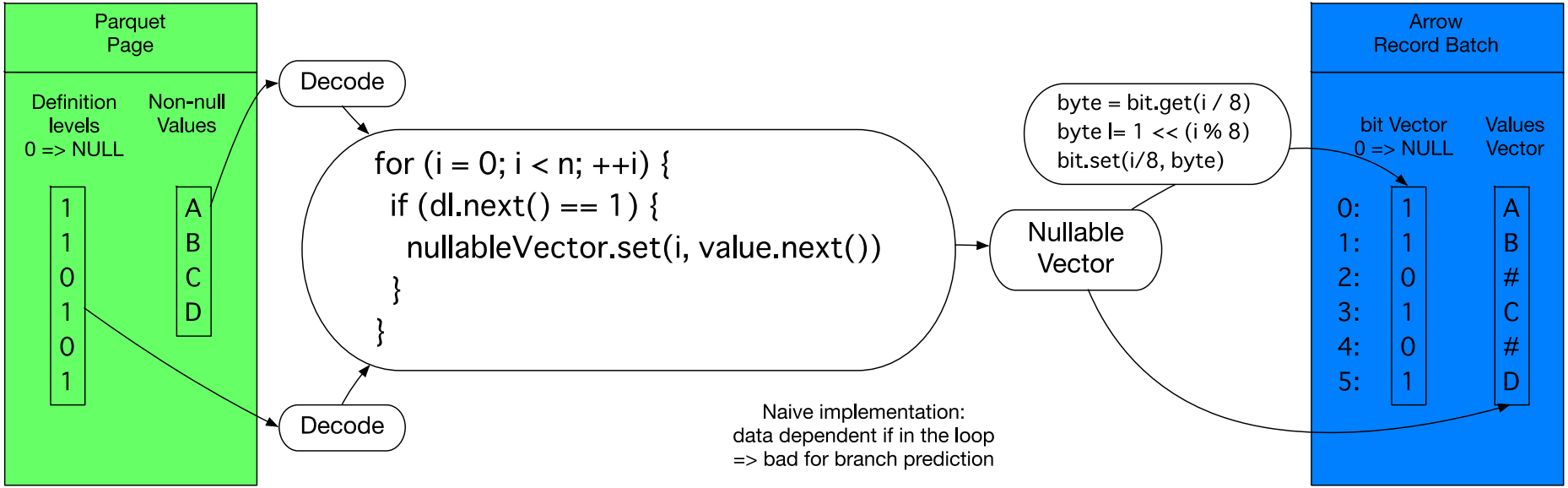
Each box (vector) is contiguous memory
The entire record batch is contiguous on wire

Vertical integration: Parquet to Arrow

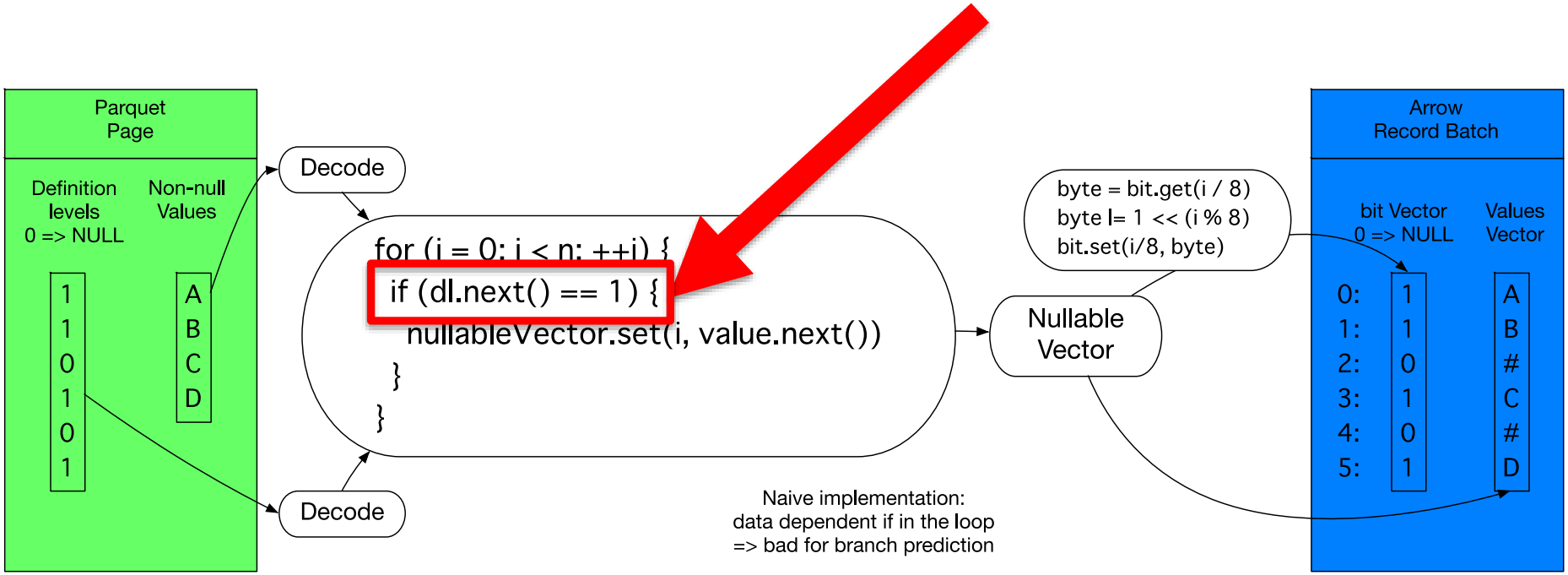
Representation comparison for flat schema



Naïve conversion



Naïve conversion

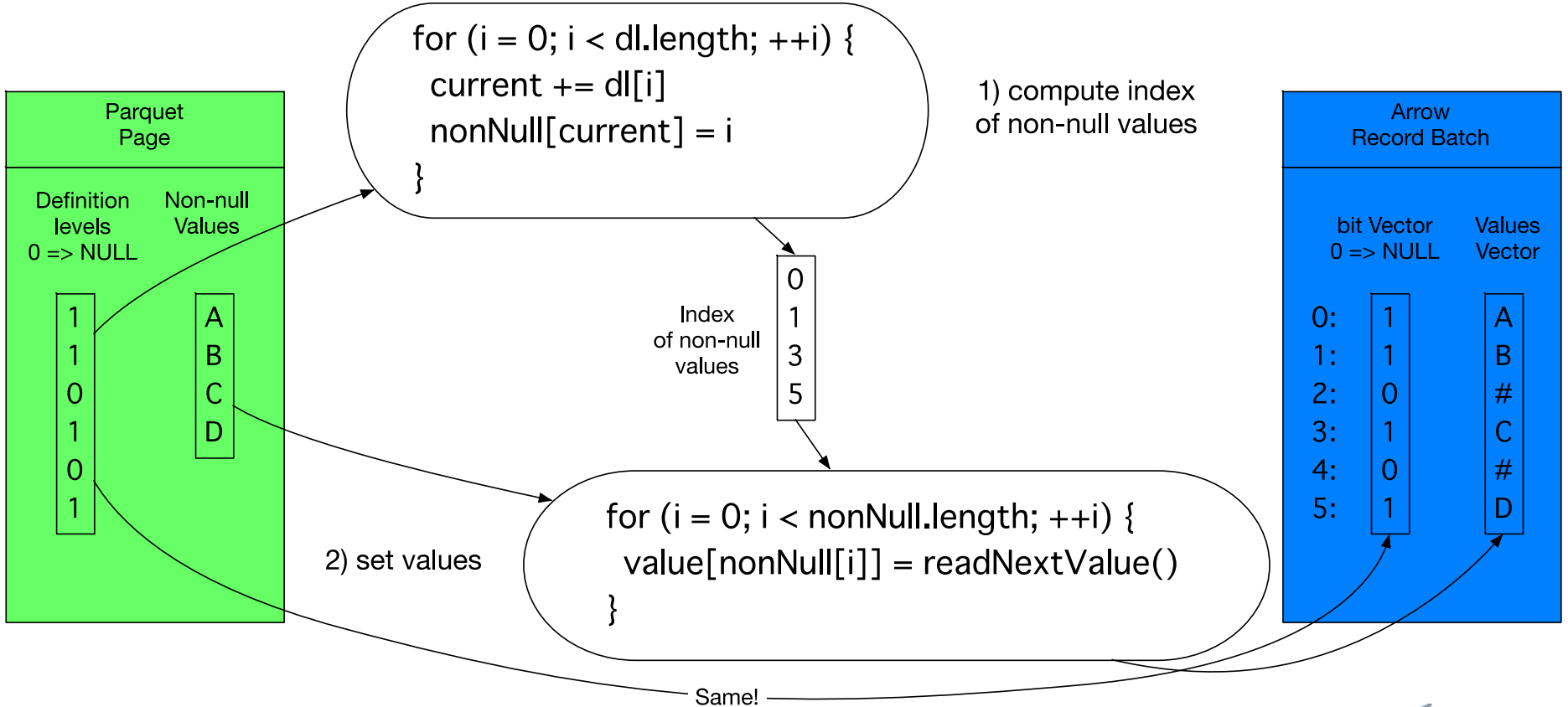


(They have layers)

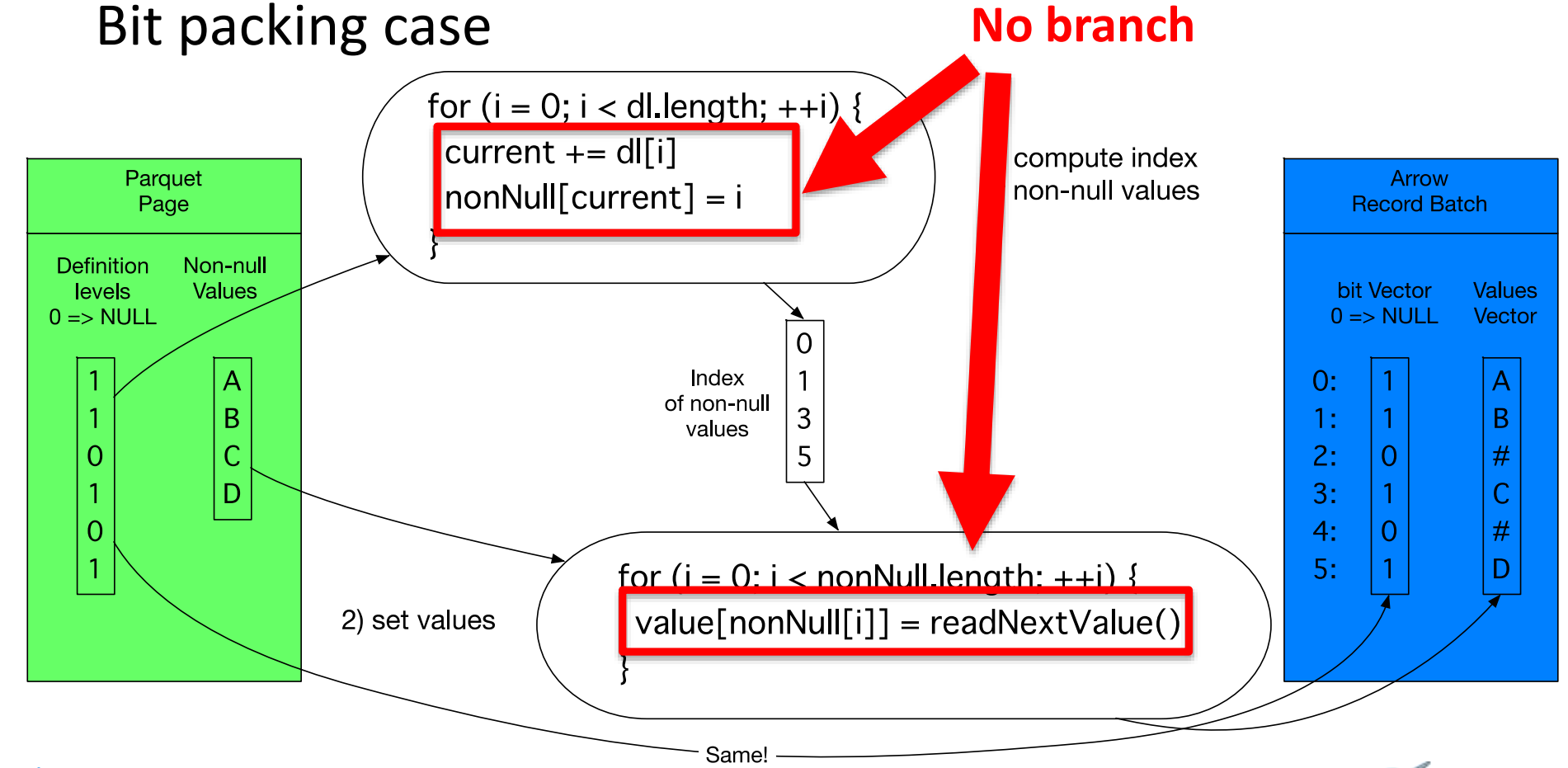
Peeling away abstraction layers
Vectorized read

dremio

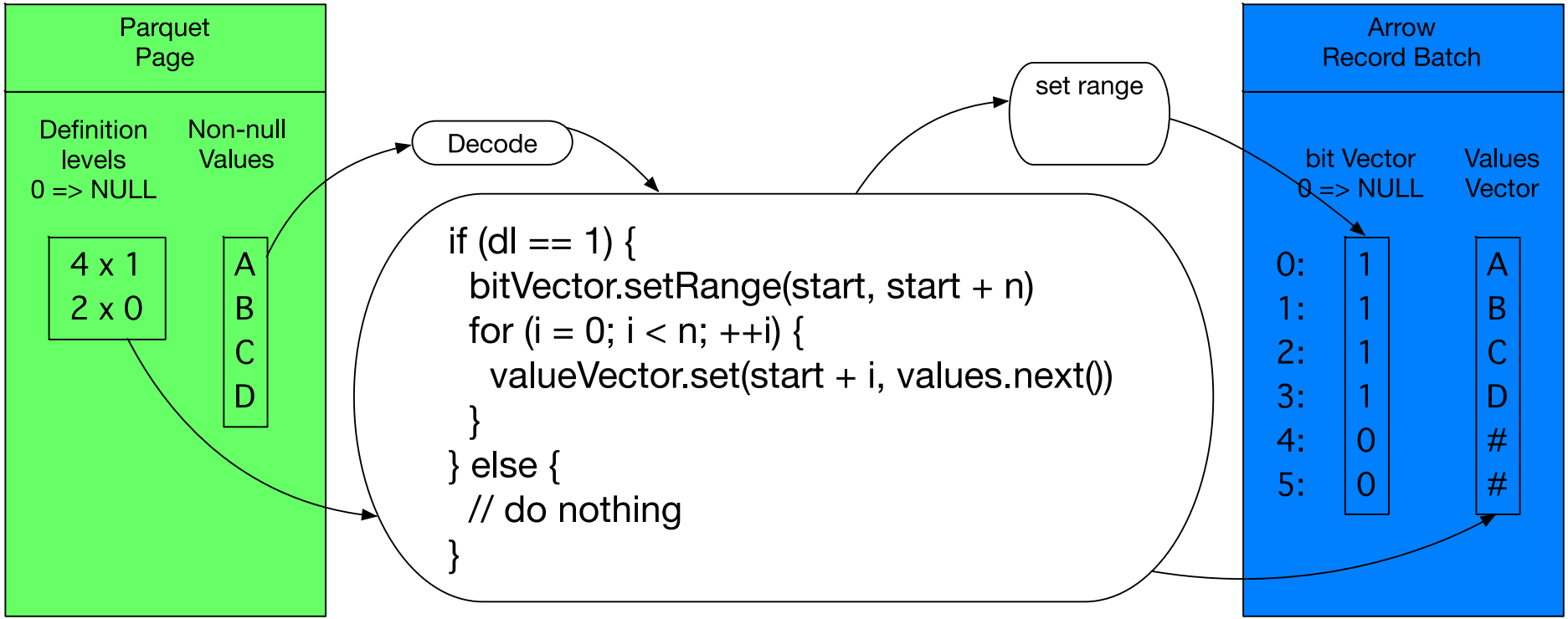
Bit packing case



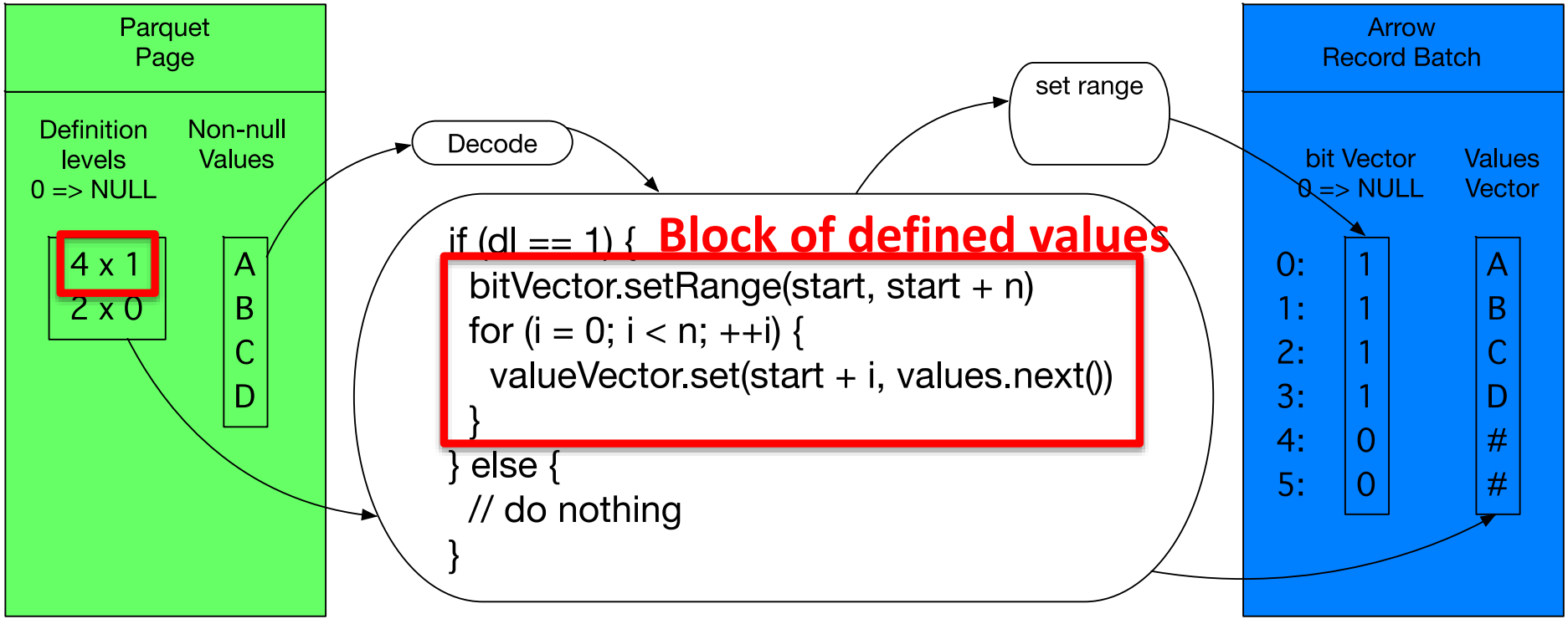
Bit packing case



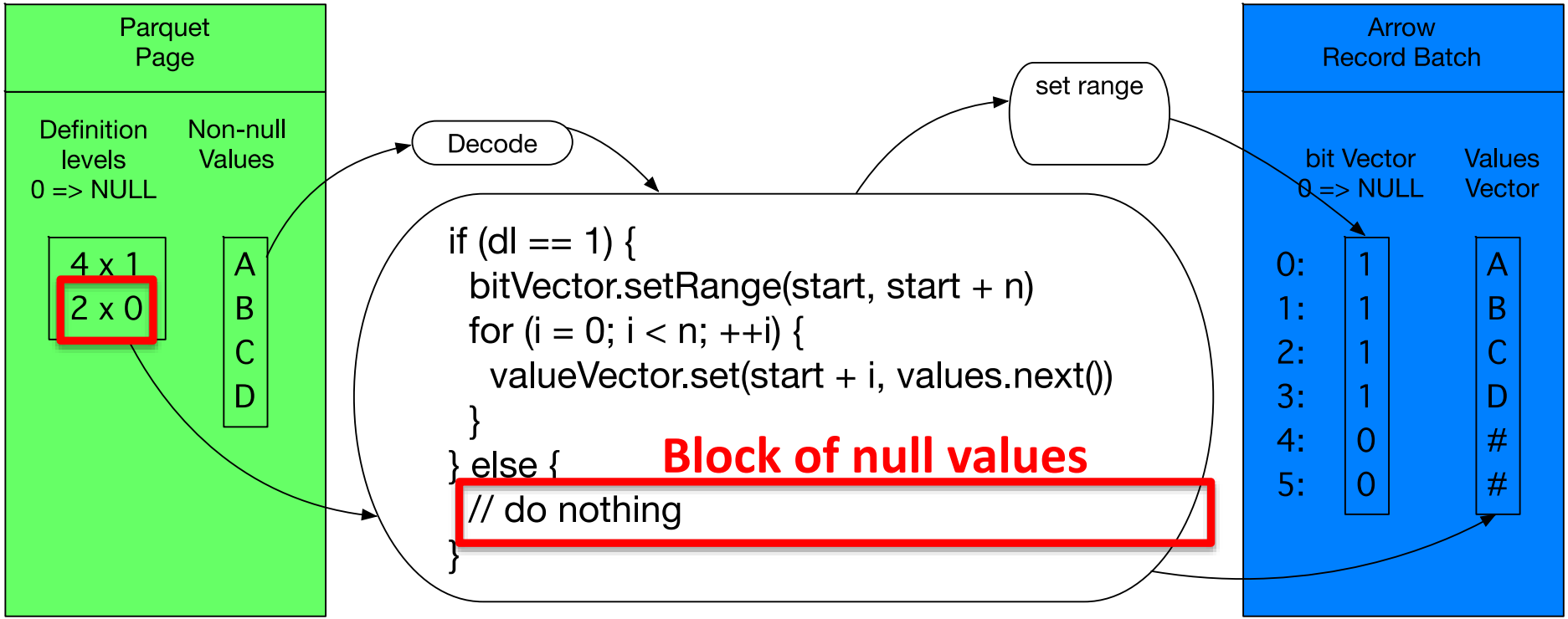
Run length encoding case



Run length encoding case

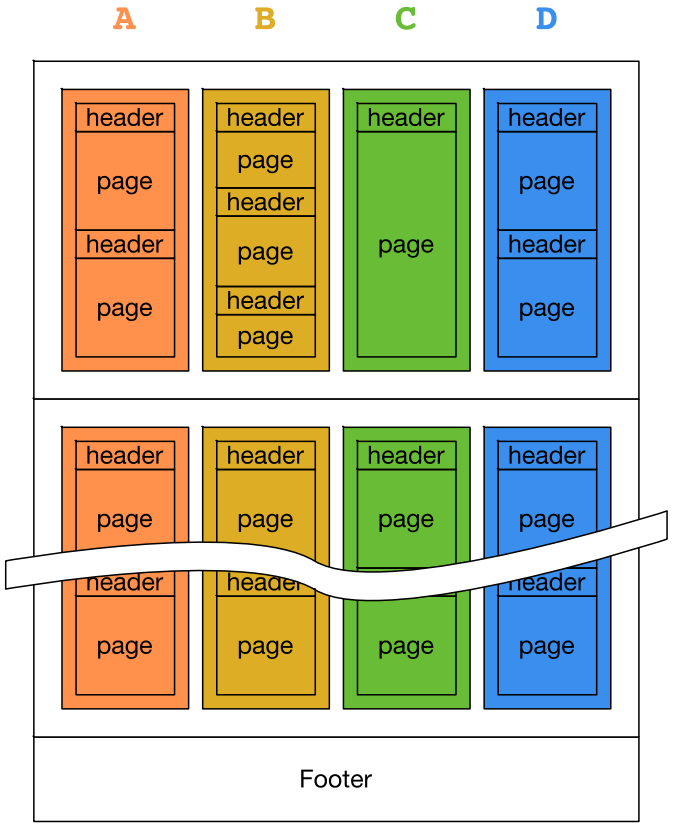


Run length encoding case



Predicate push down

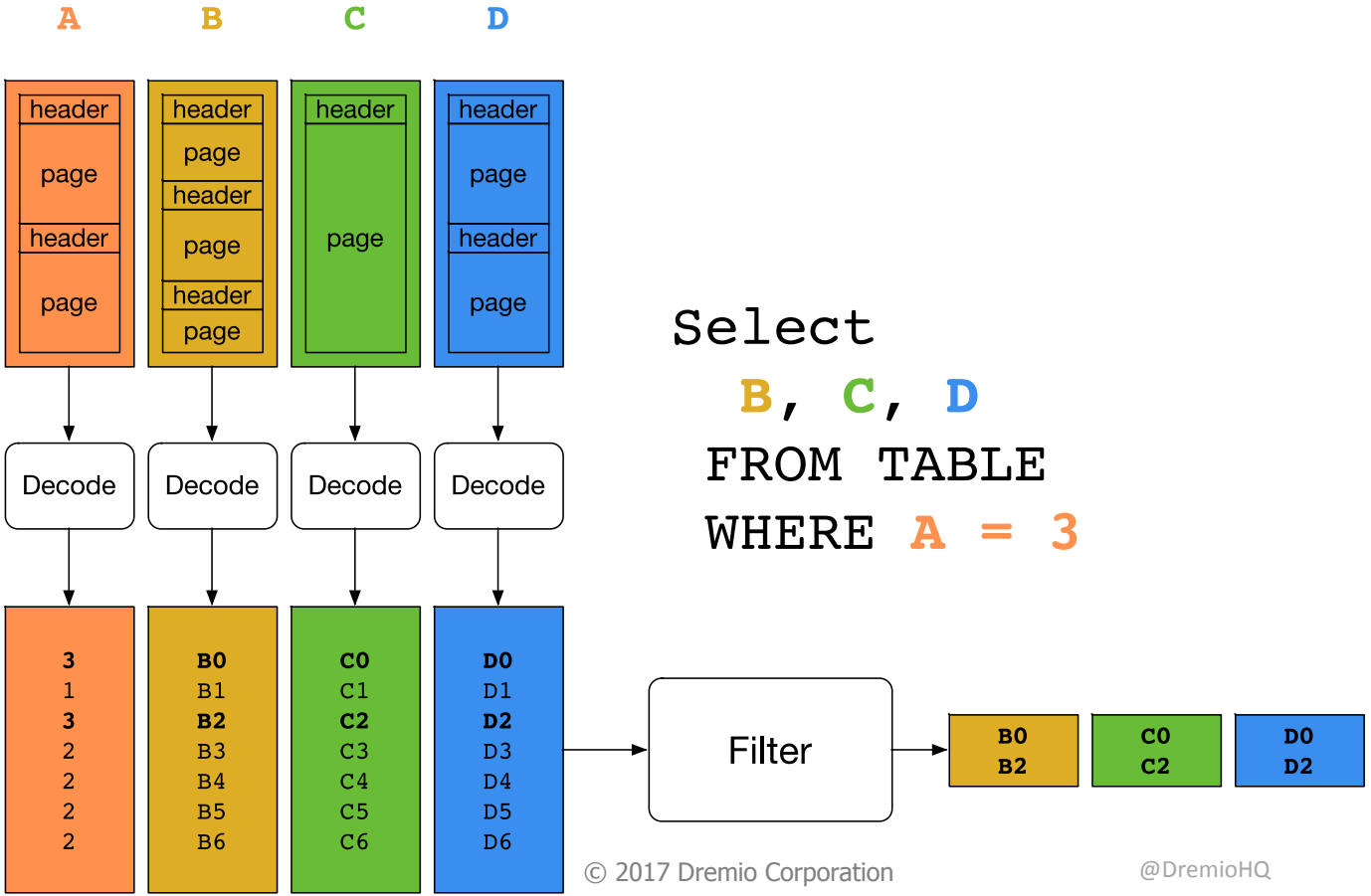
Example: filter and projection




```
Select
   B, C, D
 FROM TABLE
 WHERE A = 3
```


dremio

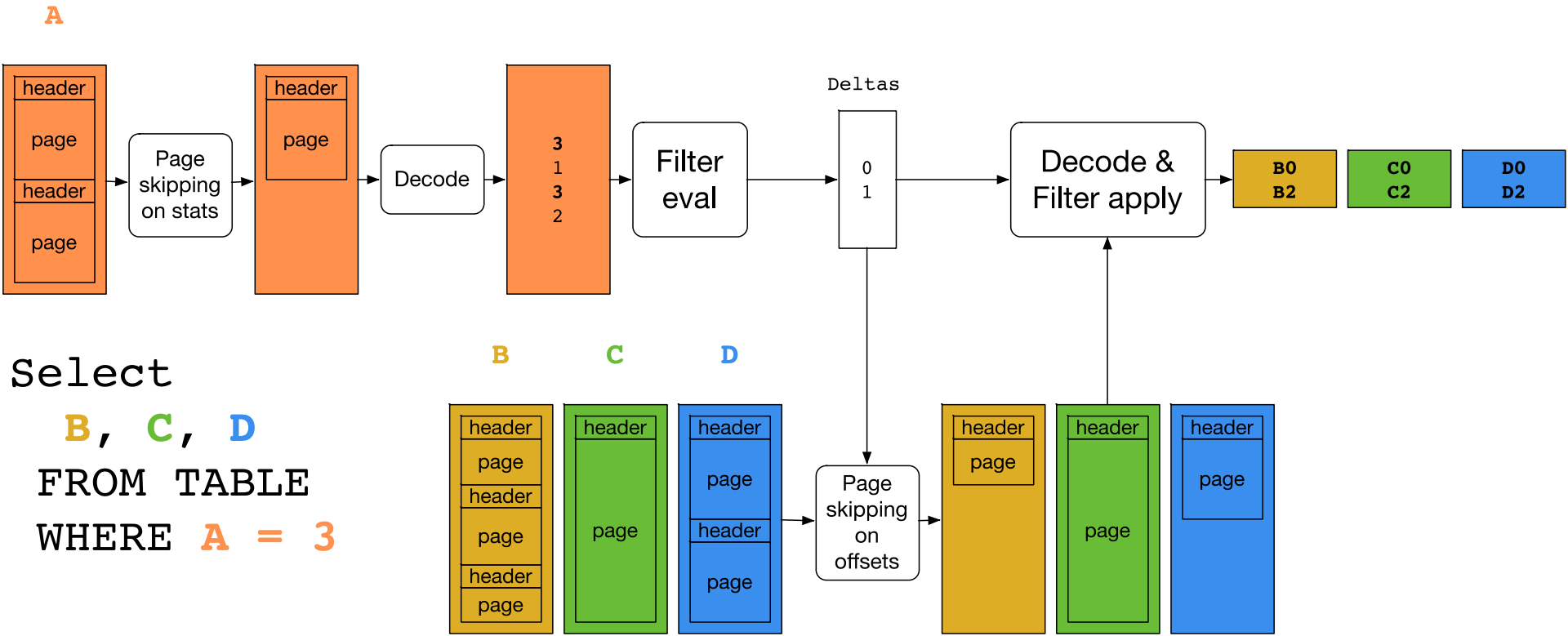
Naive filter and projection implementation




```
Select
  B, C, D
 FROM TABLE
 WHERE A = 3
```


dremio

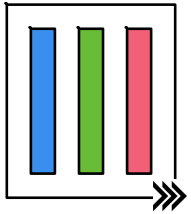
Peeling away abstractions



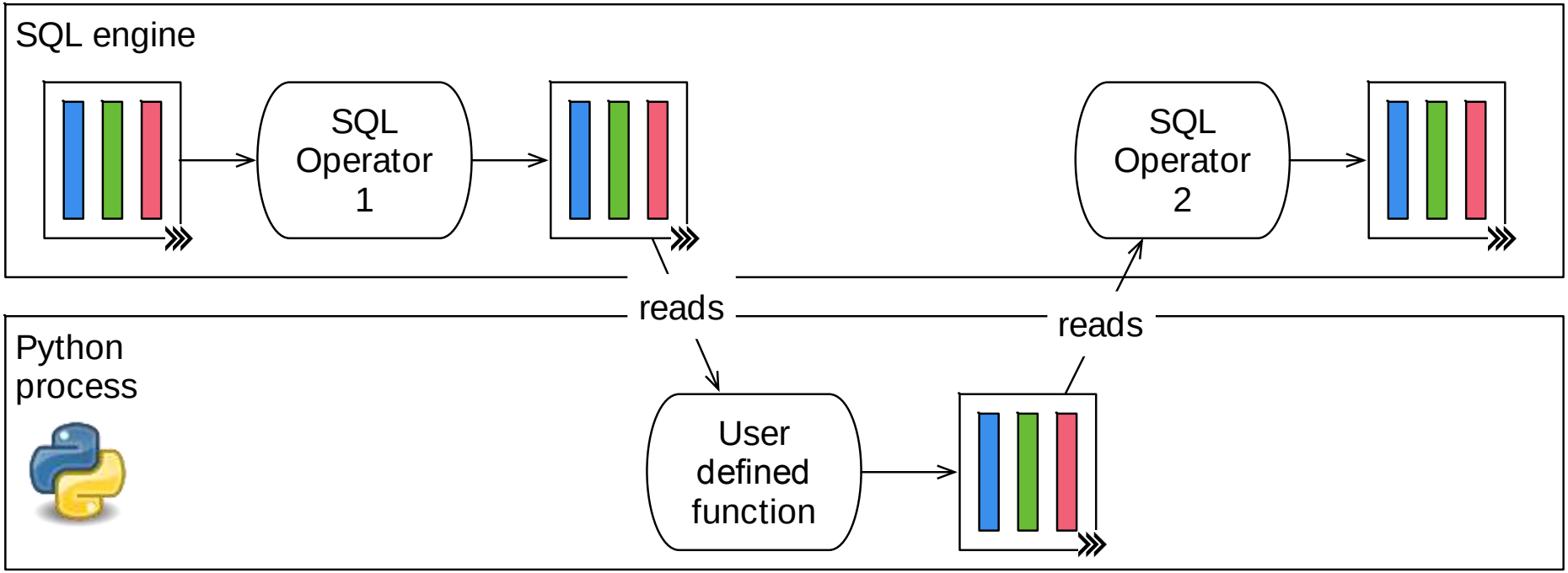

```
Select
  B, C, D
FROM TABLE
WHERE A = 3
```

Arrow based communication

Universal high performance UDFs



Immutable Arrow Batch

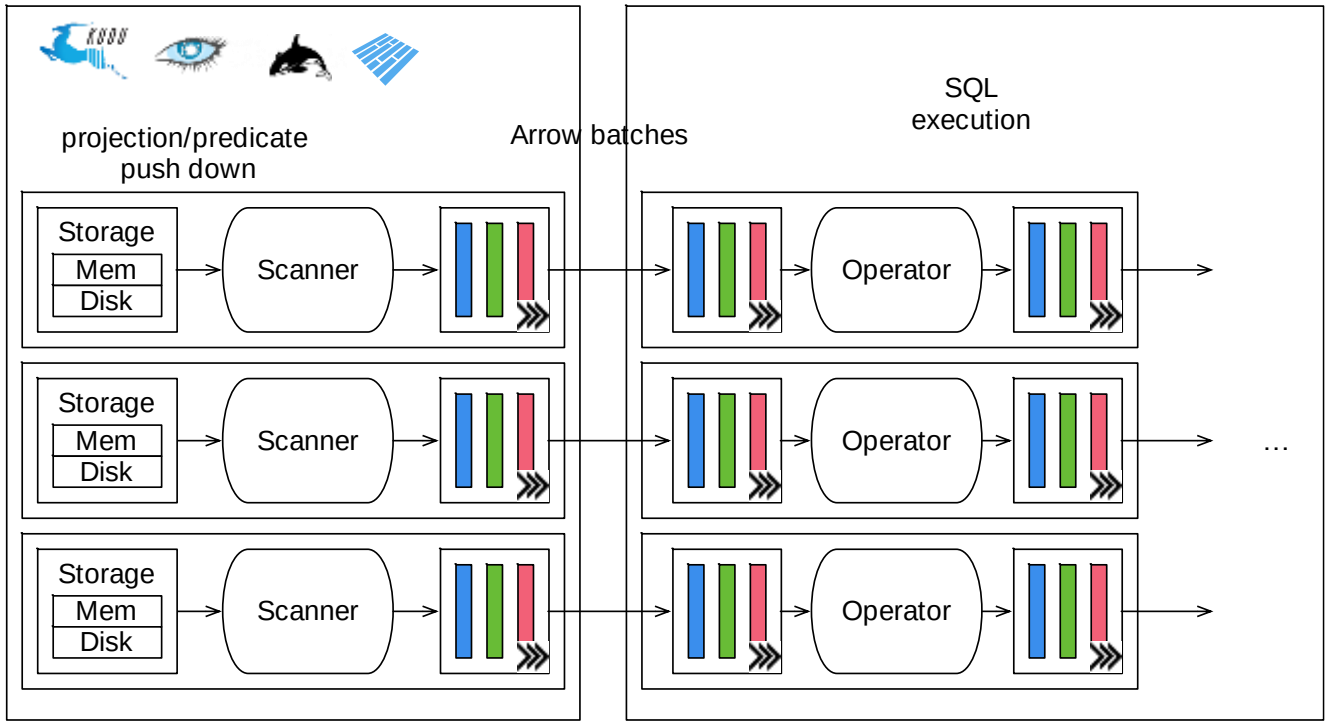


dremio

Arrow RPC/REST API

- Generic way to retrieve data in Arrow format
- Generic way to serve data in Arrow format
- Simplify integrations across the ecosystem
- Arrow based pipe

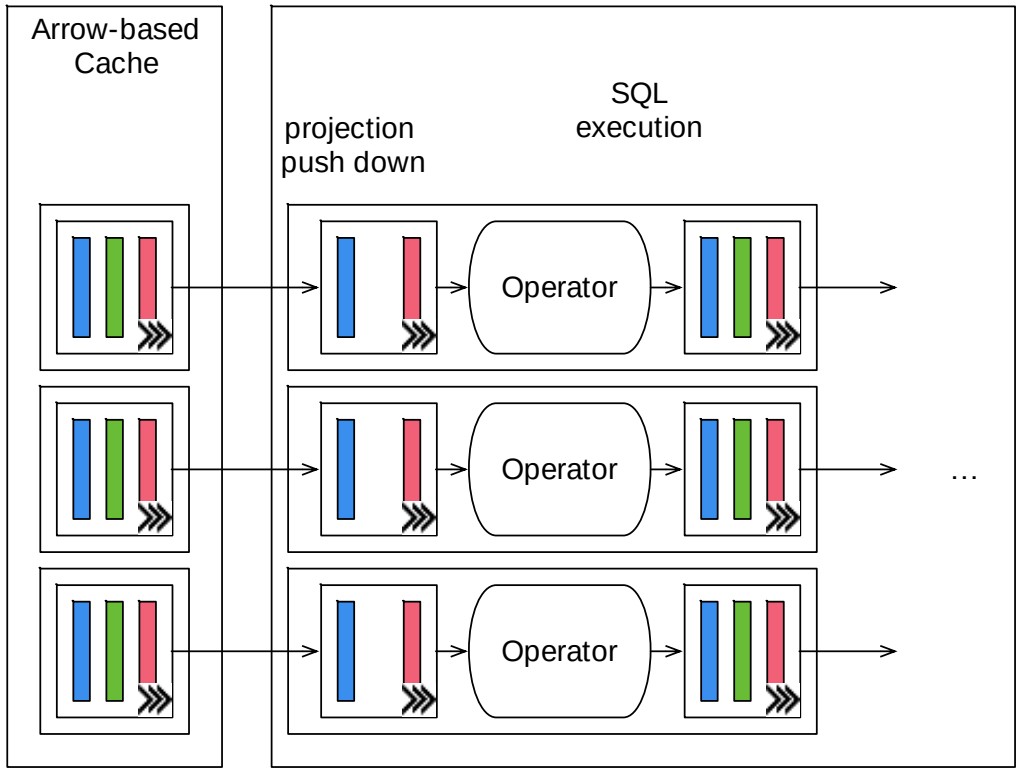
RPC: arrow based storage interchange

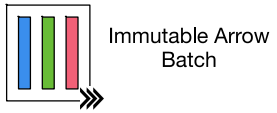


The memory representation is sent over the wire.

No serialization overhead.

RPC: arrow based cache

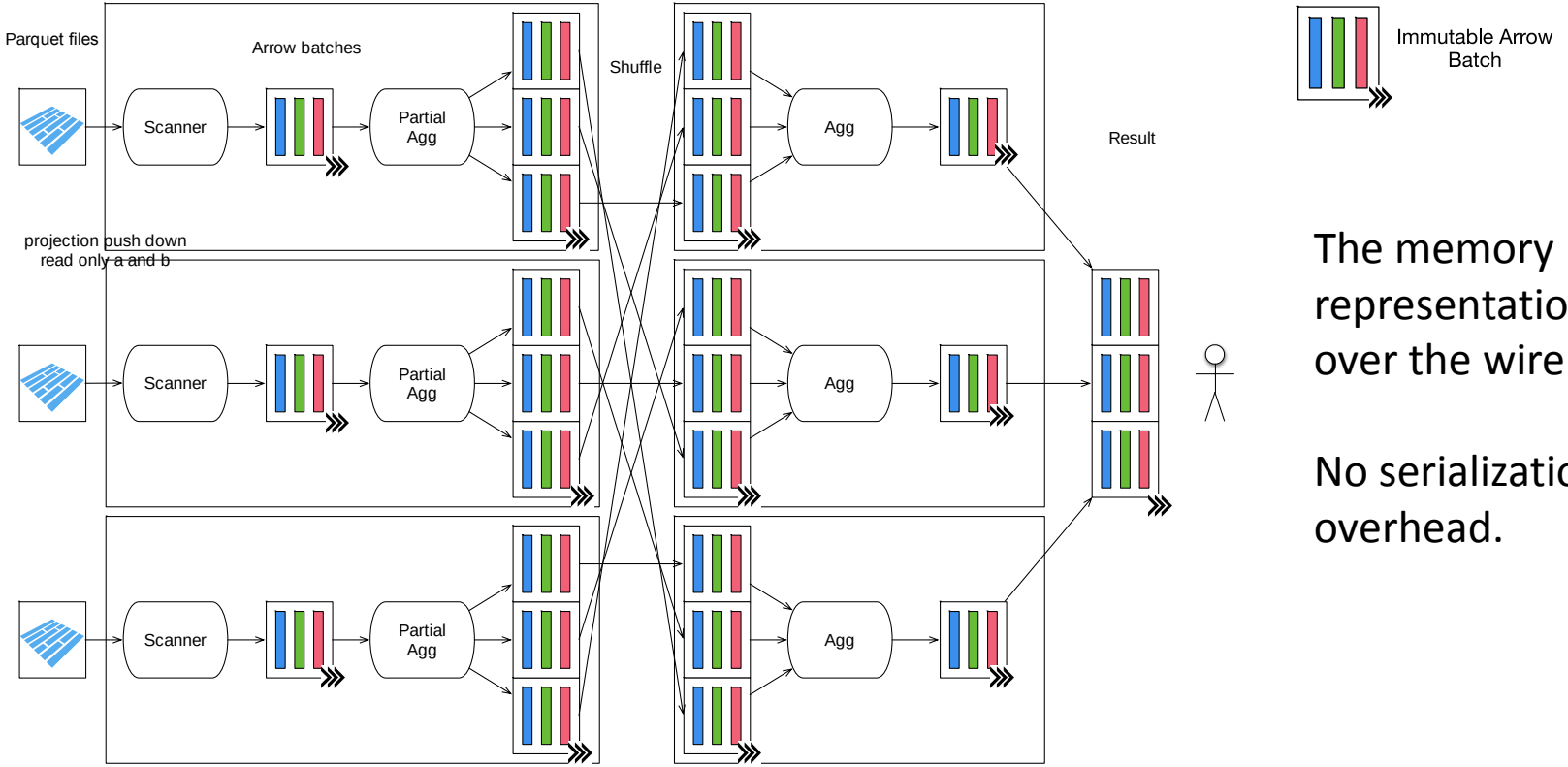




The memory representation is sent over the wire.

No serialization overhead.

RPC: Single system execution



SELECT SUM(a) FROM t GROUP BY b

The memory representation is sent over the wire.

No serialization overhead.

Results

- PySpark Integration:
  53x speedup (IBM spark work on SPARK-13534)
  http://s.apache.org/arrowresult1

- Streaming Arrow Performance
  7.75GB/s data movement
  http://s.apache.org/arrowresult2

- Arrow Parquet C++ Integration
  4GB/s reads
  http://s.apache.org/arrowresult3

- Pandas Integration
  9.71GB/s
  http://s.apache.org/arrowresult4

Language Bindings

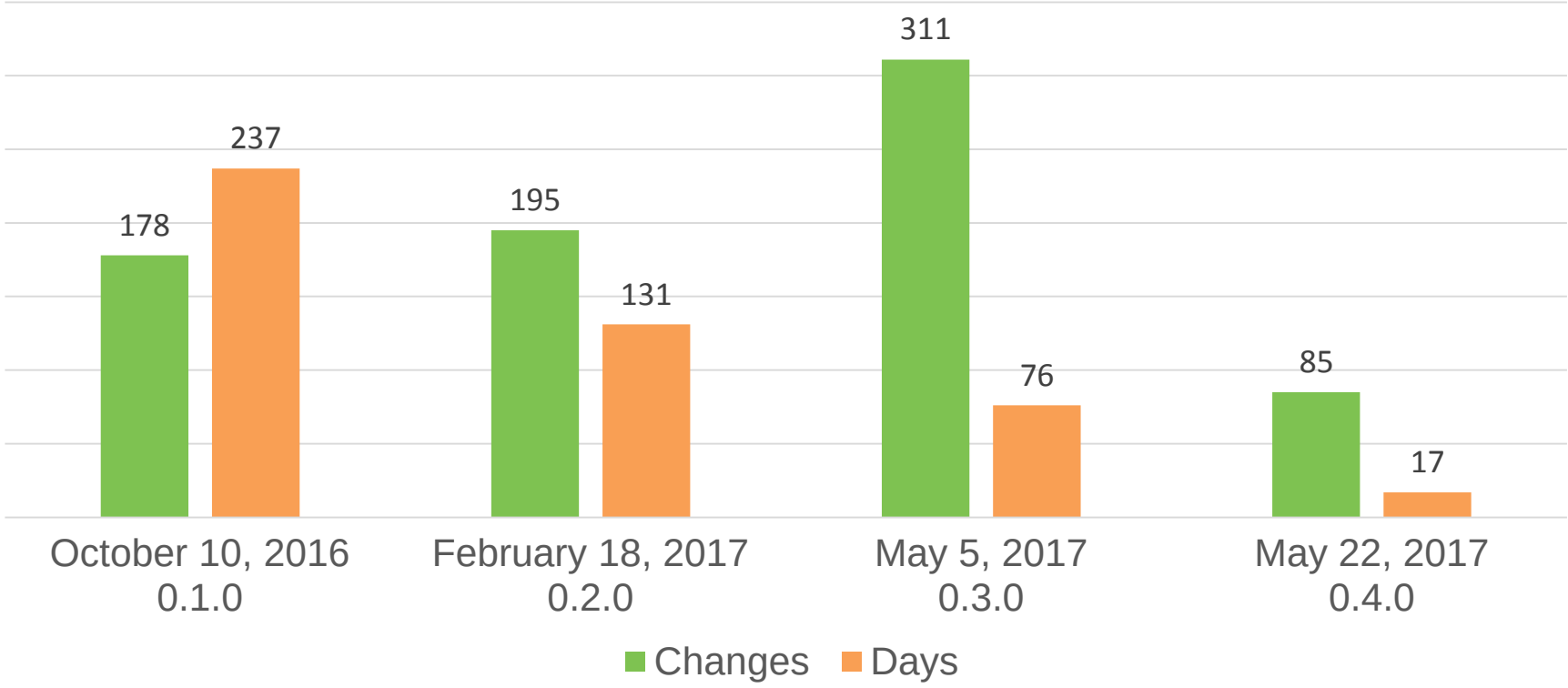
Parquet

- Target Languages
  - Java
  - CPP
  - Python & Pandas
- Engines integration:
  - Many!

Arrow

- Target Languages
  - Java
  - CPP, Python
  - R (underway)
  - C, Ruby, JavaScript
- Engines integration:
  - Drill
  - Pandas, R
  - Spark (underway)

Arrow Releases



dremio

Current activity:

- Spark Integration (SPARK-13534)
- Pages index in Parquet footer (PARQUET-922)
- Arrow REST API (ARROW-1077)
- Bindings:
  - C, Ruby (ARROW-631)
  - JavaScript (ARROW-541)

Get Involved

- Join the community
  - dev@{arrow,parquet}.apache.org
  - Slack:
    - Arrow: https://apachearrowslackin.herokuapp.com/
    - Parquet: https://parquet-slack-invite.herokuapp.com/
  - http://{arrow,parquet}.apache.org
  - Follow @Apache{Parquet,Arrow}

dremio