3DF

Reactive Datalog for Datomic

by Nikolas Göbel

“We want to make reactive systems that don't poll. And we want those systems to get a consistent view of the world.”

– Rich Hickey, “Deconstructing the Database”

Nikolas Göbel

niko@clockworks.io

in collaboration with:

Frank McSherry (ETH)

David Bach (Clockworks)

Malte Sandstede (Clockworks)

ETH zürich

Systems Group

clockworks

Reactive Systems?

- spreadsheets, chats, stock tickers, real-time dashboards
- alerting and rule engines
- collaboration tools
- stream processors

Reactive Principles

- data coordinates code
- change propagates
- a relationship, once established, is maintained
- results always reflect the current state of the domain

=C4+C5

	A	B	C	D	E	F	G
1							
2							
3							
4		A	12				
5		B	3				
6		**(+ A B)**	**15**				
7							
8							
9							
10							
11							
12							
13							

Data coordinates code, not the other way around.

~~evaluating~~

subscribing to a query

~~re-computing from scratch~~

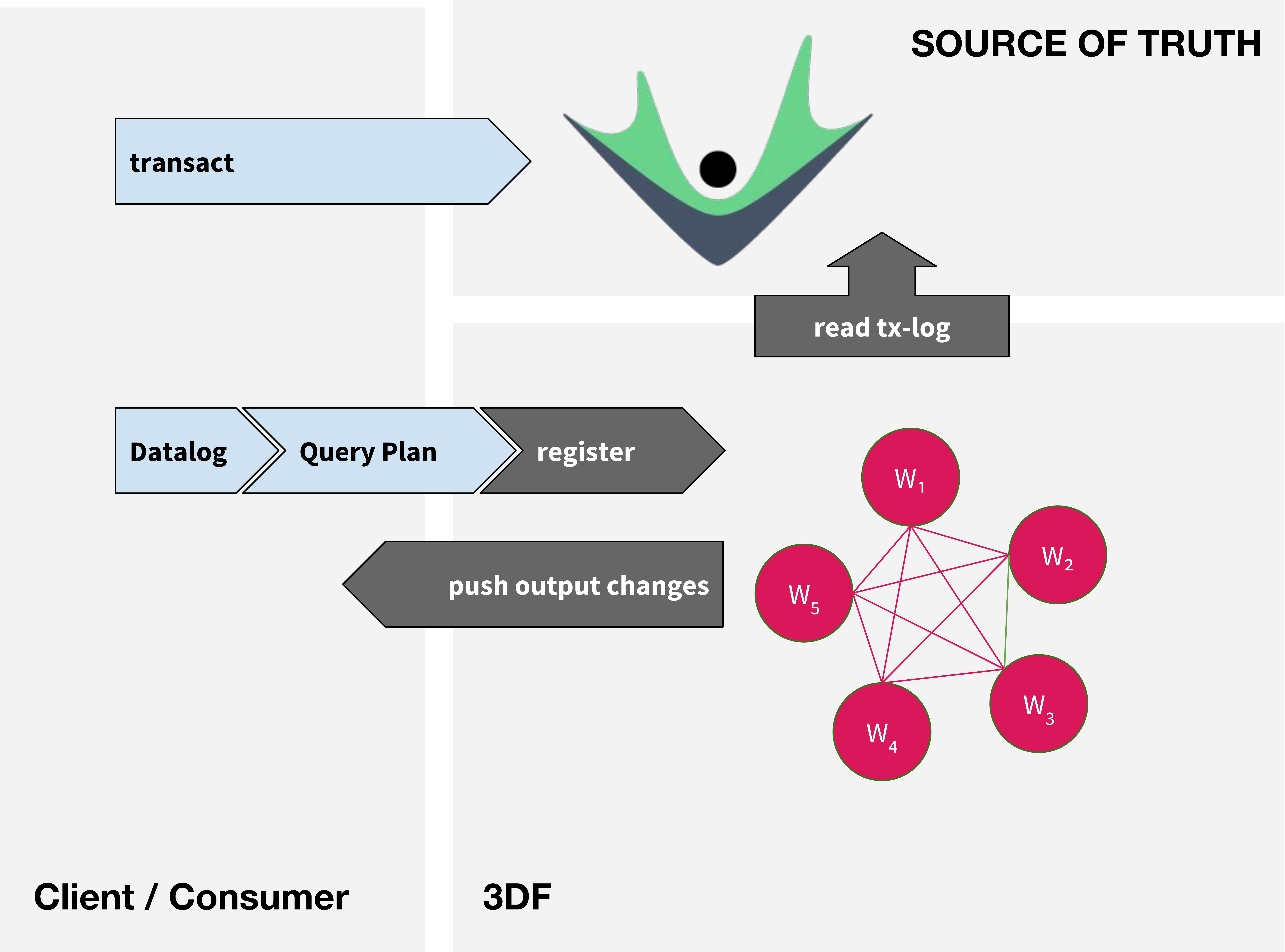
incrementally updating a previous result

<demo>

3DF (github.com/comnik/clj-3df)

- reactive Datalog queries
- feeds from durable sources (Datomic, Datahike, Kafka)
- scales from browsers to clusters

github.com/comnik/clj-3df github.com/comnik/declarative-dataflow



SOURCE OF TRUTH

transact

read tx-log

Datalog

Query Plan

register

push output changes

Client / Consumer

3DF

Making Reactive Practical (I)

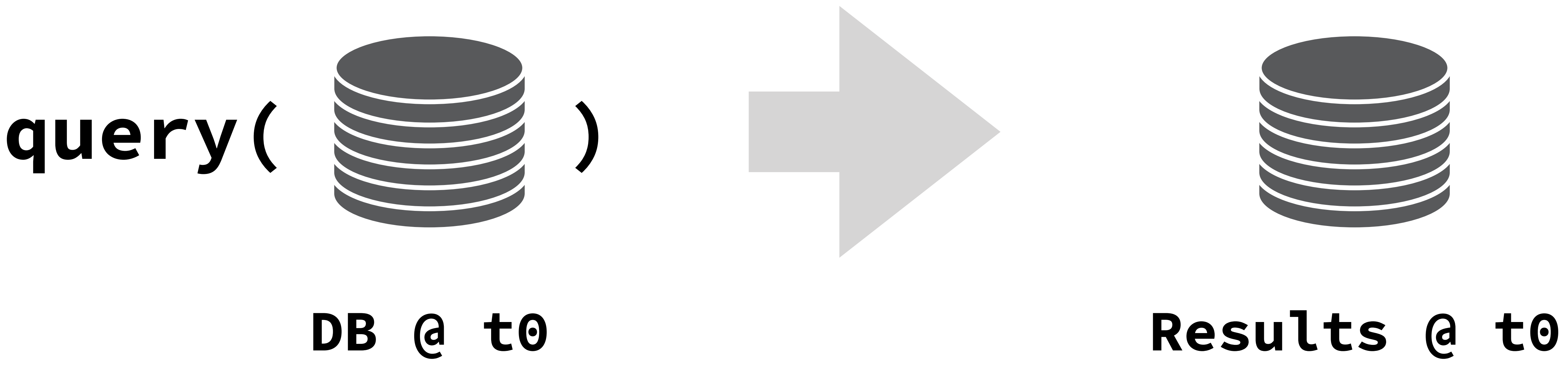
data coordinates code
work like a spreadsheet…

declarative
…without writing joins by hand

Making Reactive Practical (II)

transparent
notion of change shouldn’t leak into Datalog…

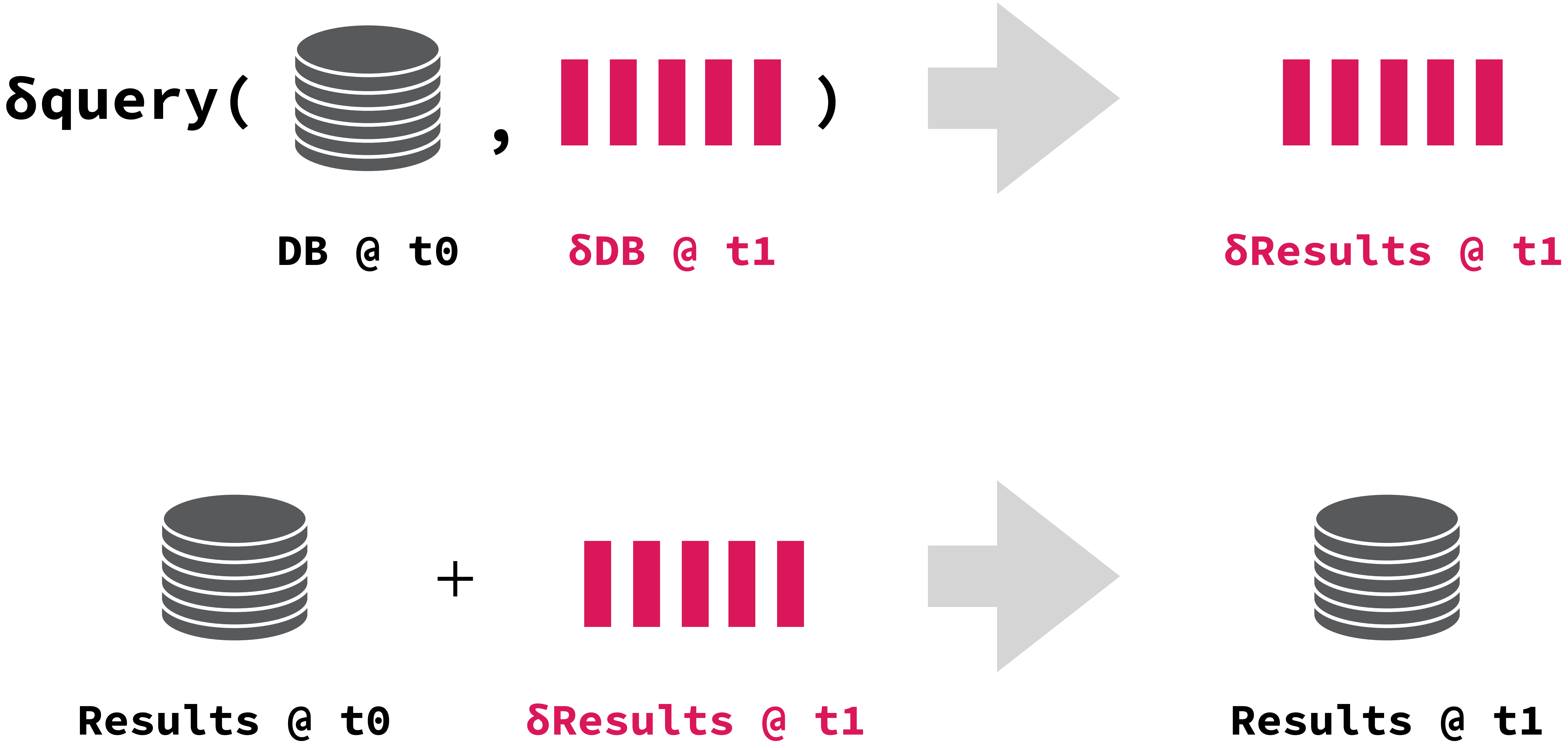
efficient
…but update results incrementally



Computing with Snapshots

```
[:find ?device
 :where
 [?device :settings/speed ?target]
 [?device :device/speed ?speed]
 [(< ?speed ?target)]]
```

✓ transparent ✓ declarative



δquery(,)

DB @ t0

δDB @ t1

δResults @ t1

Results @ t0

+

δResults @ t1

Results @ t1

Computing with Differences

```
[:find ?device
 :in $ ?tx
 :where
 [$ ?device :settings/speed ?target]
 [?tx ?device :device/speed ?speed]
 [(< ?speed ?target)]]
```

✓ incremental

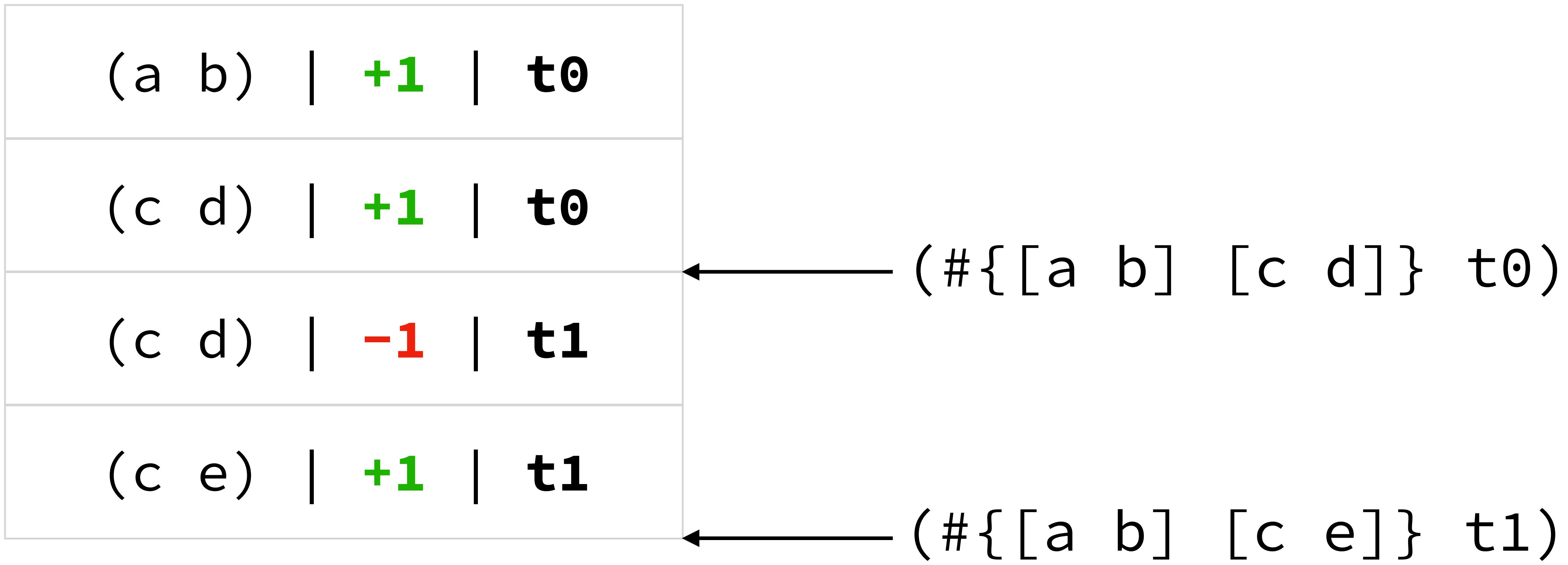
Differential Dataflow

github.com/TimelyDataflow/differential-dataflow

“a data-parallel programming framework designed to quickly respond to arbitrary changes in input”

Time-varying Collections

```
:edge
```



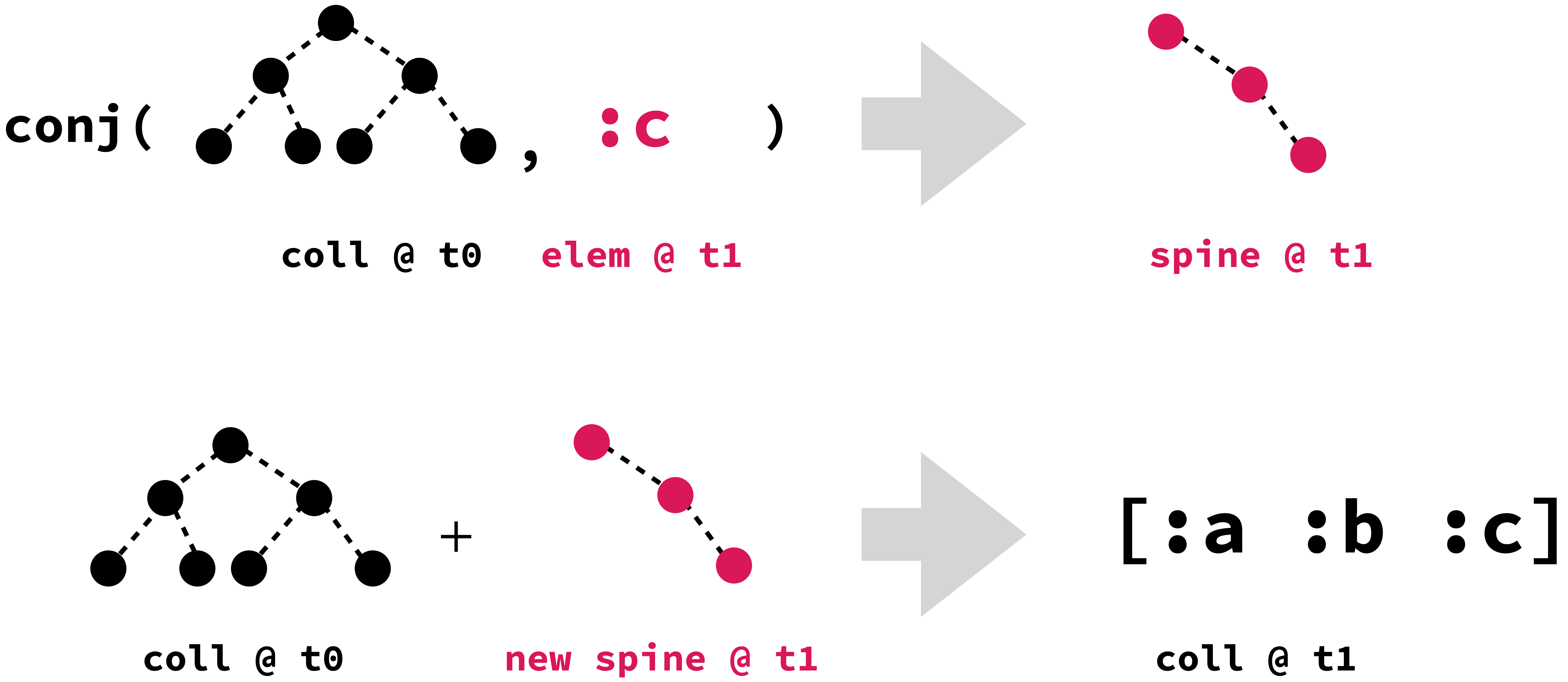
```
(a b) | +1 | t0
(c d) | +1 | t0
(c d) | -1 | t1
(c e) | +1 | t1
```

```
(#{[a b] [c d]} t0)
(#{[a b] [c e]} t1)
```

Partially-ordered Times

```
(bfs :edge a)
```

<table>
<tr><td>(a b) | +1 | (t0 0)</td><td></td><td></td></tr>
<tr><td>(c d) | +1 | (t0 0)</td><td></td><td></td></tr>
<tr><td>(b c) | +1 | (t1 0)</td><td>(a c) | +1 | (t1 1)</td><td>(a d) | +1 | (t1 2)</td></tr>
<tr><td>(c d) | -1 | (t2 0)</td><td>-</td><td>(a d) | -1 | (t2 2)</td></tr>
</table>



conj(, :c)

coll @ t0 elem @ t1

spine @ t1

+

coll @ t0

new spine @ t1

[:a :b :c]

coll @ t1

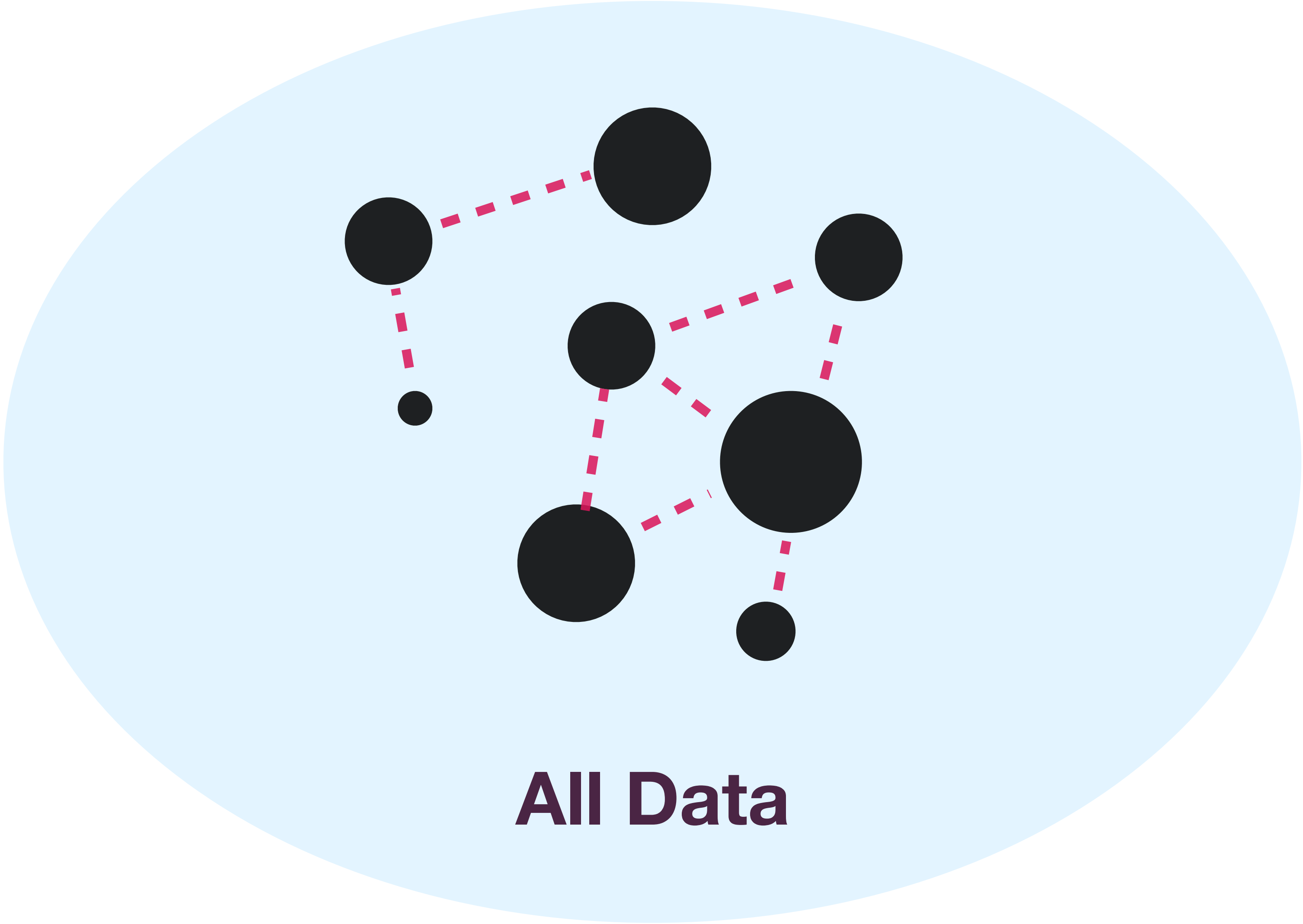
Computing with Differences

Reactive Made Practical

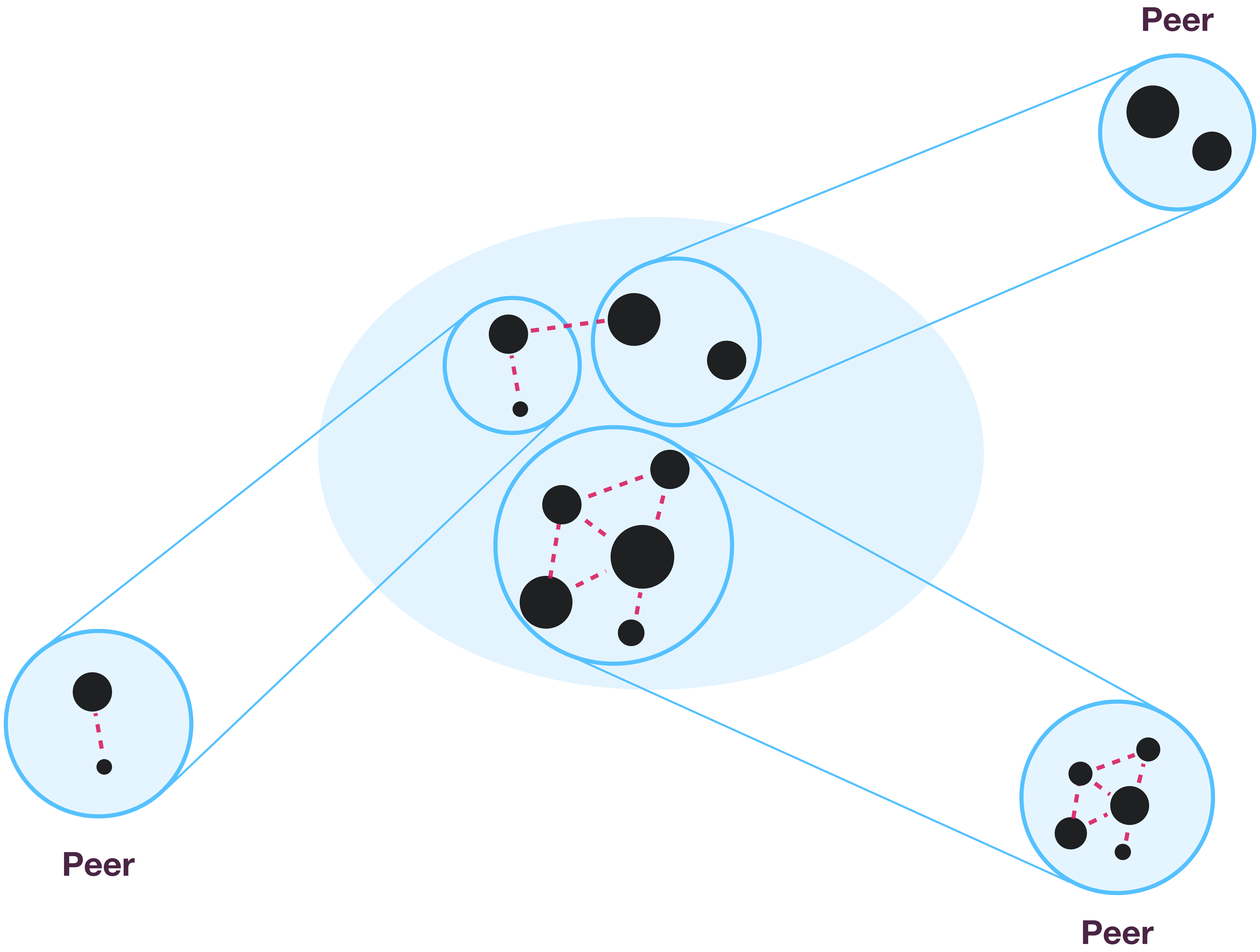
- ✓ data coordinates code
- ✓ declarative joins, filters, and aggregations
- ✓ write Datalog without thinking about change
- ✓ all queries update incrementally / only propagate change

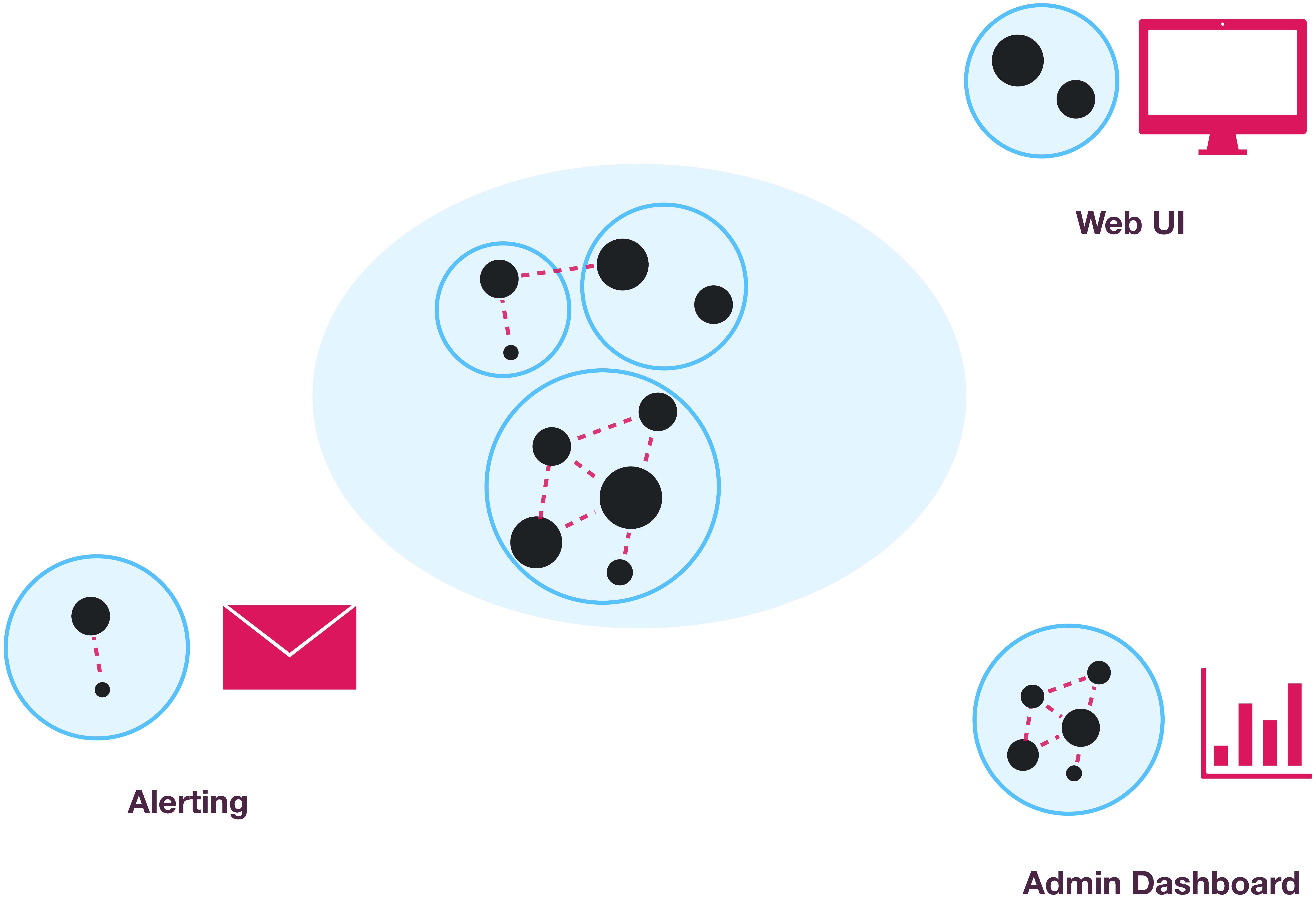
~~rule the world~~

Everybody wants to be a database peer.



All Data


Peer
Peer
Peer

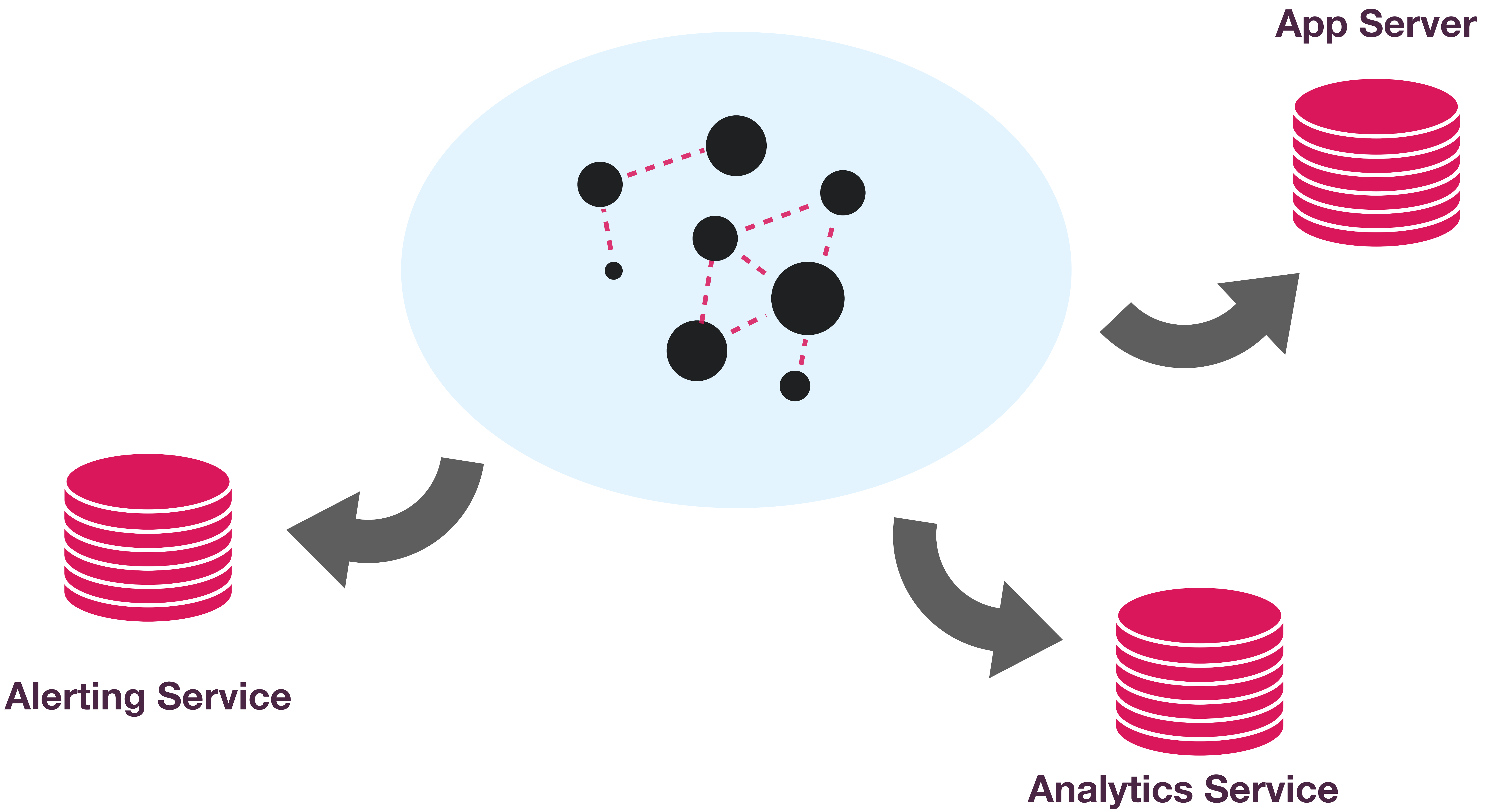


Web UI

Alerting

Admin Dashboard

Datomic:
Broadcast All Novelty To All Peers



Ions:
Move Consumers Into The DB

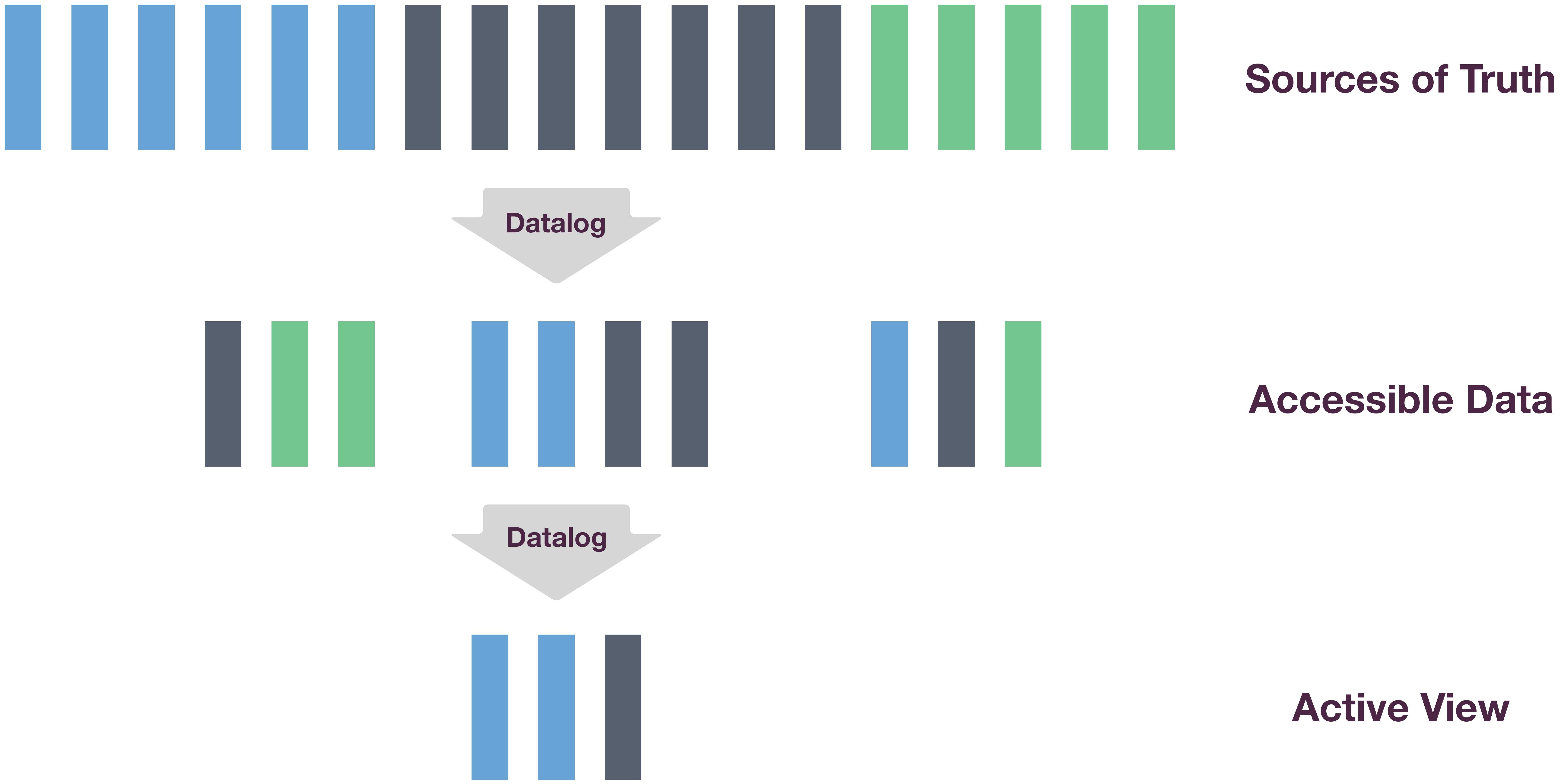
Selective Replication

multiple data sources?
Datomic shards, Kafka, external services…

only send relevant novelty to peers?
for browsers or mobile clients, a LRU cache is not enough

runtimes other than the JVM?
browsers, ML, data science…

3DF:
Declarative Replication via Datalog



Selective Replication

- ✓ open to anyone who speaks Datalog
- ✓ only relevant novelty propagated to each consumer
- ✓ scales to deal with multiple sources

“We want to make reactive systems that don't poll. And we want those systems to get a consistent view of the world.”

– Rich Hickey, “Deconstructing the Database”

3DF lets you…

- …leverage Datalog to build reactive systems
- …treat data consumers as database peers
- …scale queries beyond the limits of a single peer
- …integrate with hand-written Differential Dataflow

Interested? Let’s talk!

@NikolasGoebel | niko@clockworks.io

github.com/comnik/clj-3df

github.com/TimelyDataflow