ANDREA LATTUADA, PHD CANDIDATE @ ETH SYSTEMS GROUP

PAPERS WE LOVE: ZÜRICH

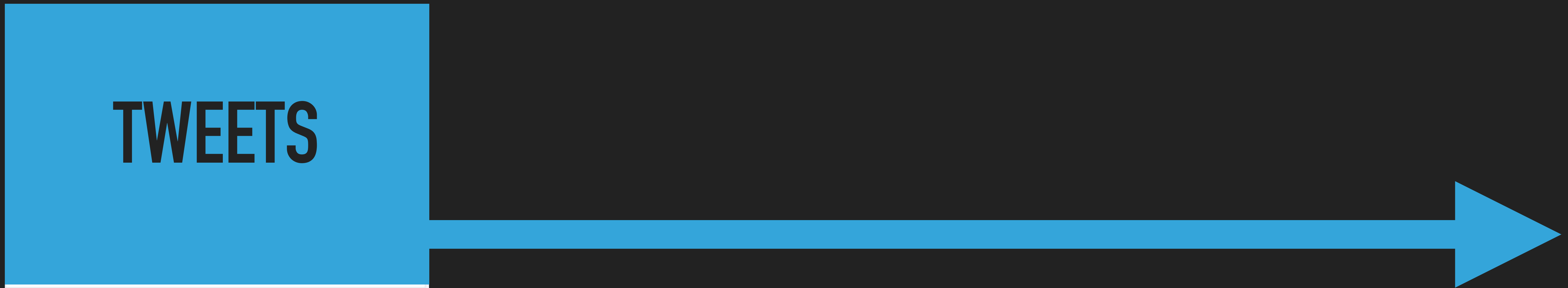
❤️ NAIAD: A TIMELY DATAFLOW SYSTEM

DEREK G. MURRAY, FRANK MCSHERRY, REBECCA ISAACS, MICHAEL ISARD, PAUL BARHAM, MARTÍN ABADI

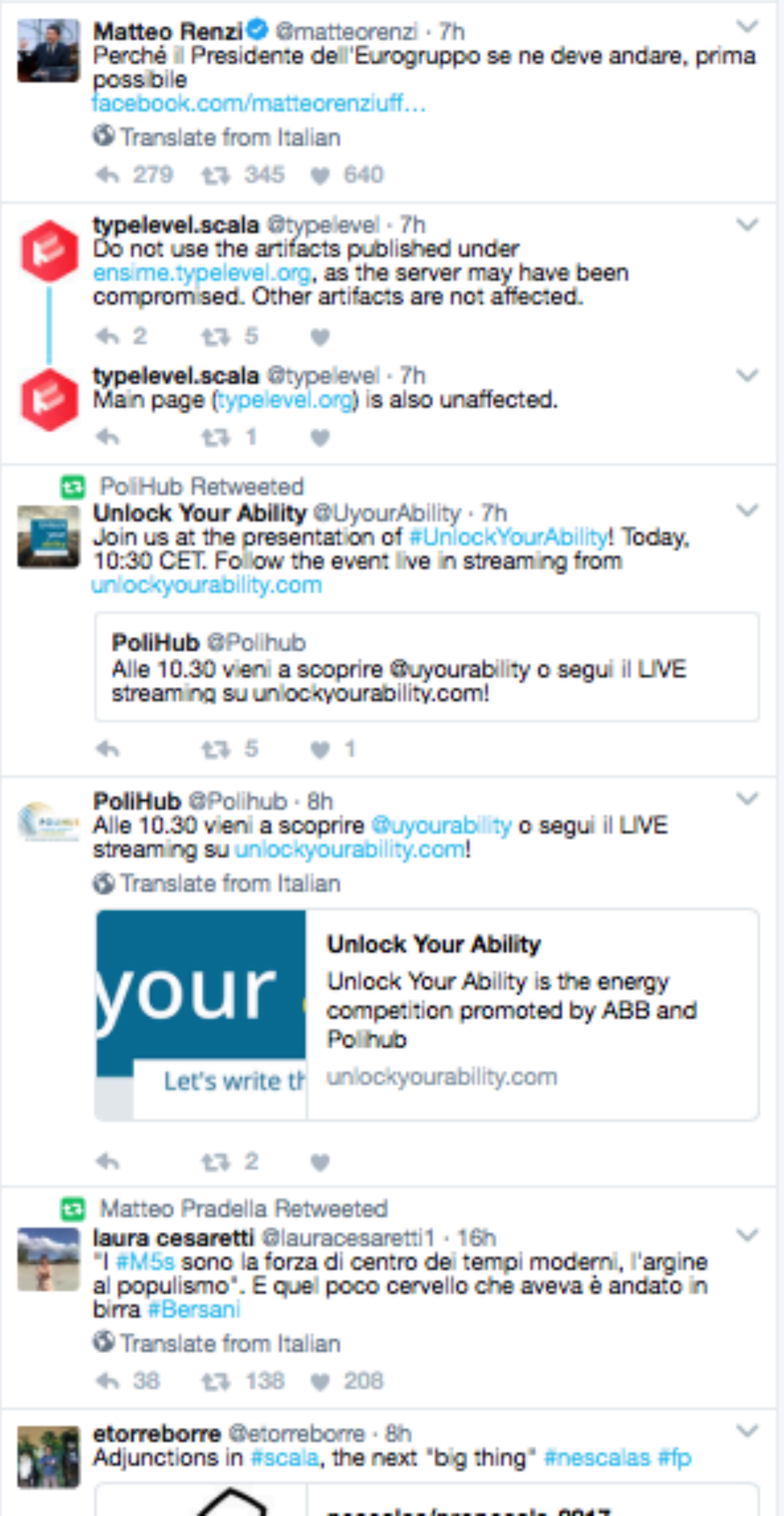
ABSTRACT

NAIAD IS A DISTRIBUTED SYSTEM FOR EXECUTING DATA PARALLEL, CYCLIC DATAFLOW PROGRAMS.

CONTINUOUS INPUT



TWEETS



Matteo Renzi @matteorenzi · 7h
Perché il Presidente dell'Eurogruppo se ne deve andare, prima possibile
facebook.com/matteorenziuff…
Translate from Italian
279 345 640

typelevel.scala @typelevel · 7h
Do not use the artifacts published under ensime.typelevel.org, as the server may have been compromised. Other artifacts are not affected.
2 5

typelevel.scala @typelevel · 7h
Main page (typelevel.org) is also unaffected.
1

PoliHub Retweeted
Unlock Your Ability @UyourAbility · 7h
Join us at the presentation of #UnlockYourAbility! Today, 10:30 CET. Follow the event live in streaming from unlockyourability.com

PoliHub @Polihub
Alle 10.30 vieni a scoprire @uyourability o segui il LIVE streaming su unlockyourability.com!

5 1

PoliHub @Polihub · 8h
Alle 10.30 vieni a scoprire @uyourability o segui il LIVE streaming su unlockyourability.com!
Translate from Italian

your
Let's write th

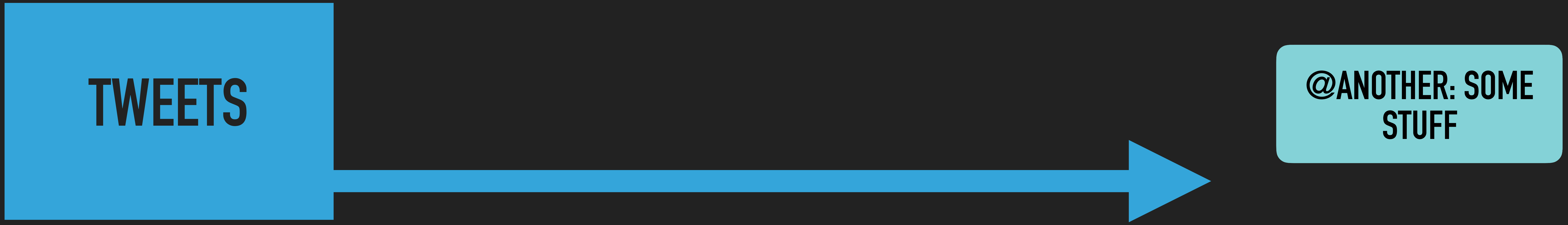
Unlock Your Ability
Unlock Your Ability is the energy competition promoted by ABB and Polihub
unlockyourability.com

2

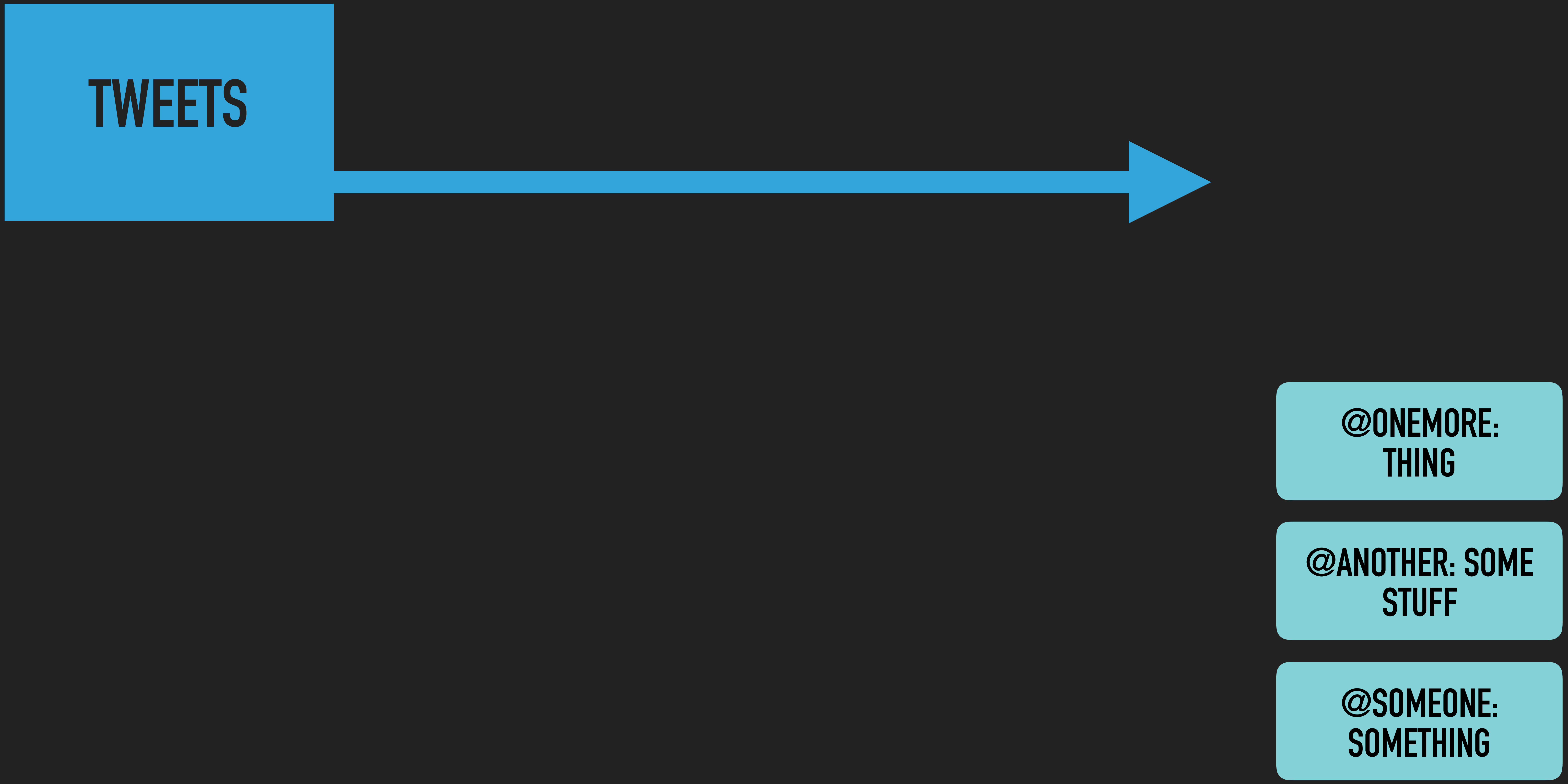
Matteo Pradella Retweeted
laura cesaretti @lauracesaretti1 · 16h
"I #M5s sono la forza di centro dei tempi moderni, l'argine al populismo". E quel poco cervello che aveva è andato in birra #Bersani
Translate from Italian
38 138 208

etorreborre @etorreborre · 8h
Adjunctions in #scala, the next "big thing" #nescalas #fp

CONTINUOUS INPUT



CONTINUOUS INPUT

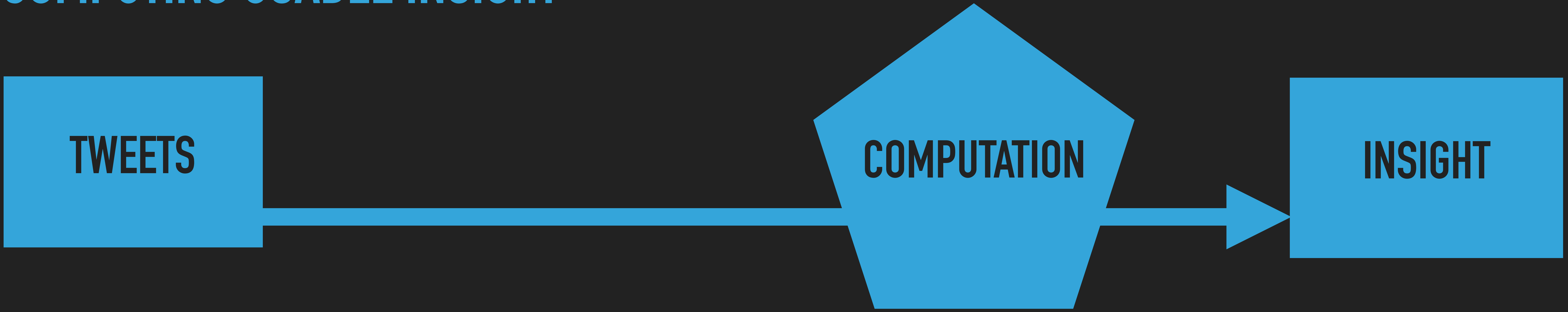

TWEETS

@ONEMORE: THING

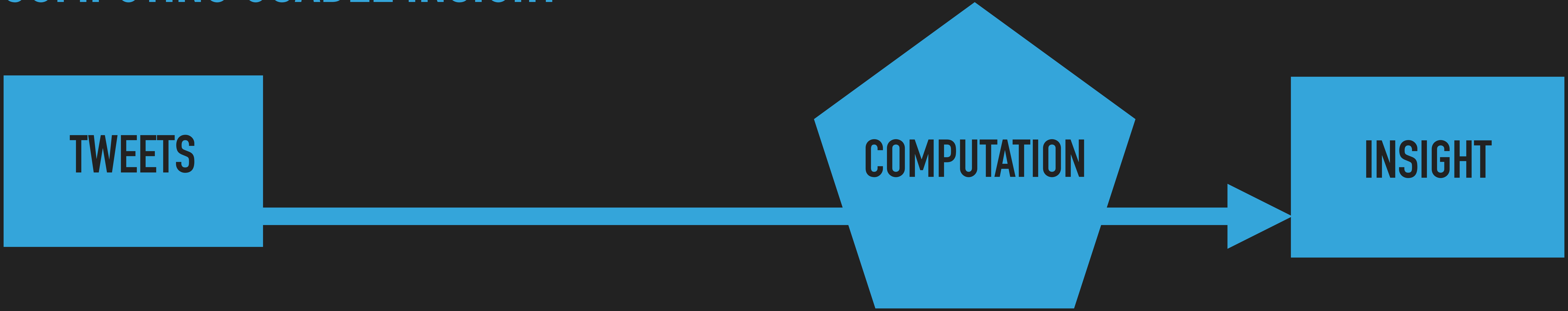
@ANOTHER: SOME STUFF

@SOMEONE: SOMETHING

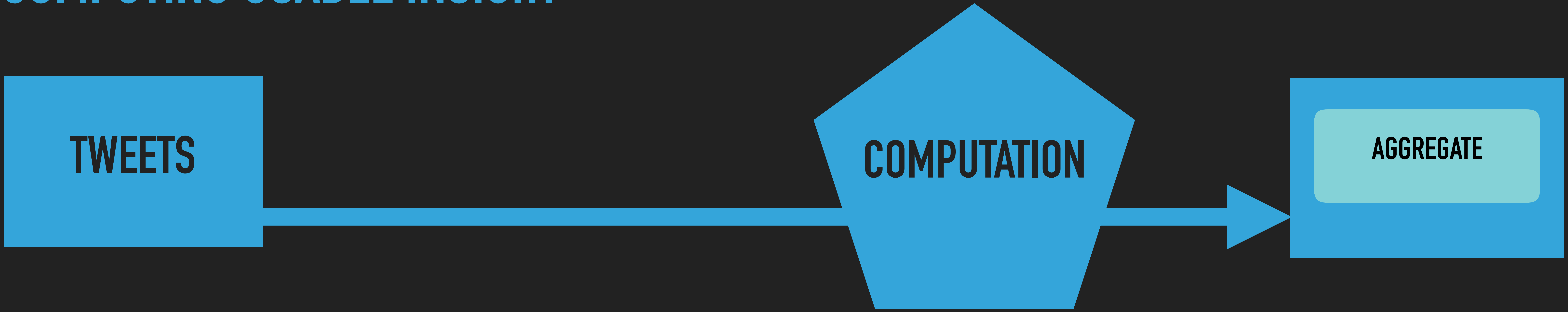
COMPUTING USABLE INSIGHT



COMPUTING USABLE INSIGHT

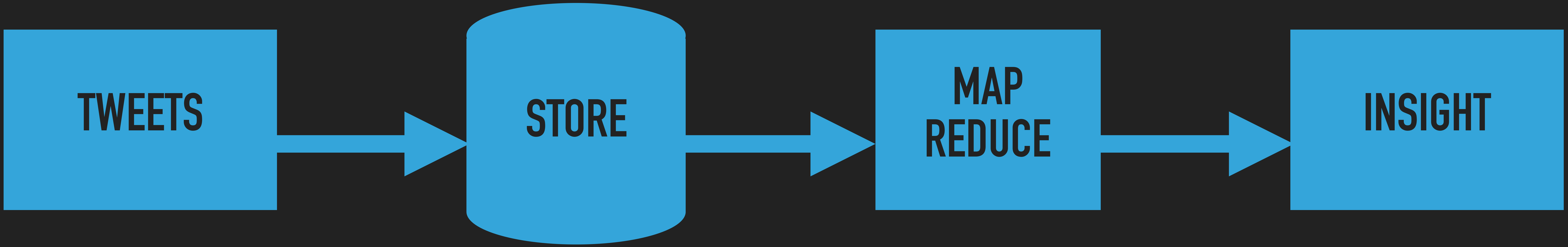


COMPUTING USABLE INSIGHT

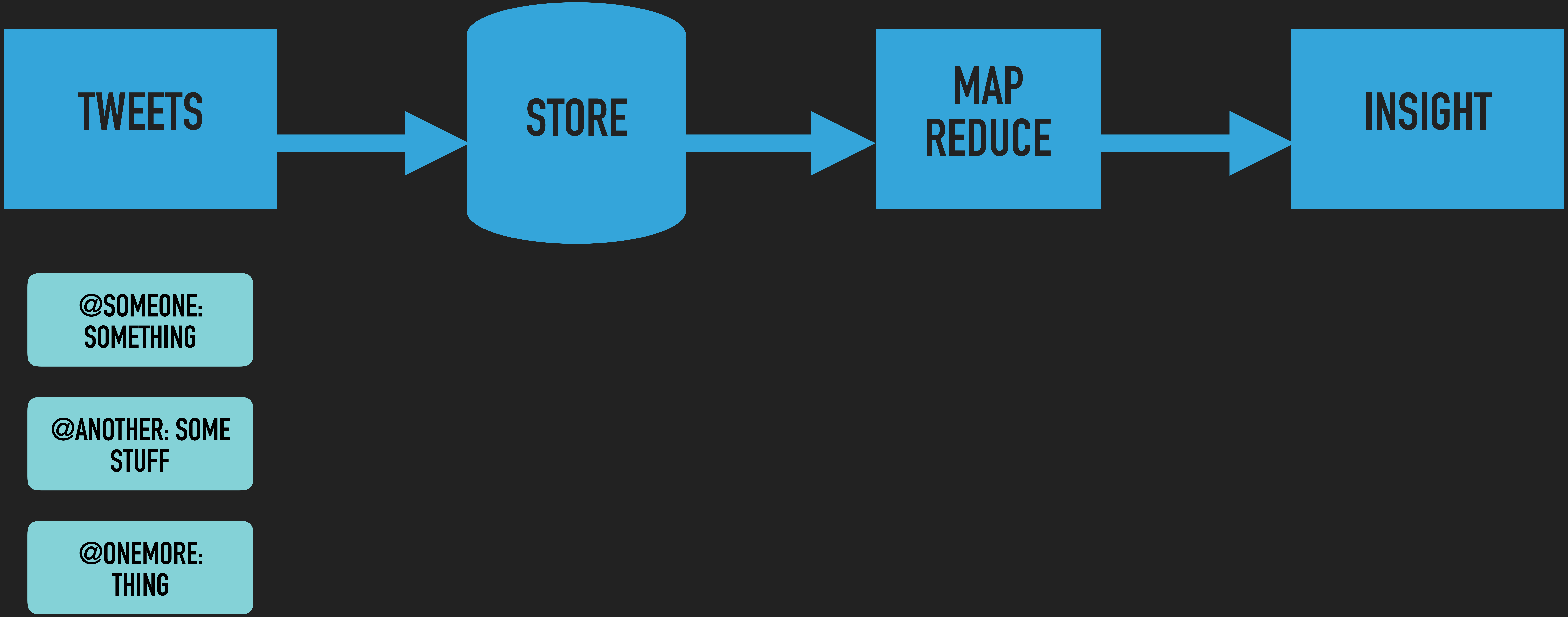


CONTEXT: AGGREGATION

MAP REDUCE



MAP REDUCE



TWEETS

STORE

MAP REDUCE

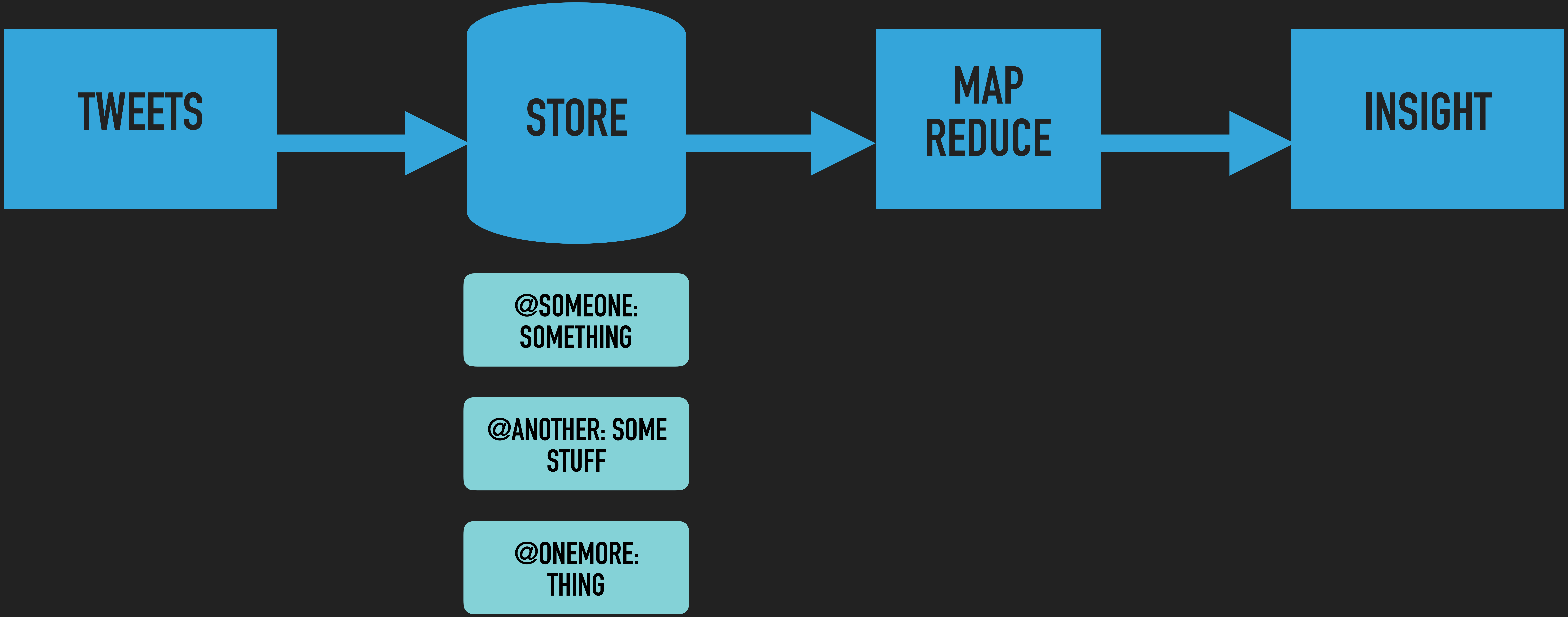
INSIGHT

@SOMEONE: SOMETHING

@ANOTHER: SOME STUFF

@ONEMORE: THING

MAP REDUCE



TWEETS

STORE

MAP REDUCE

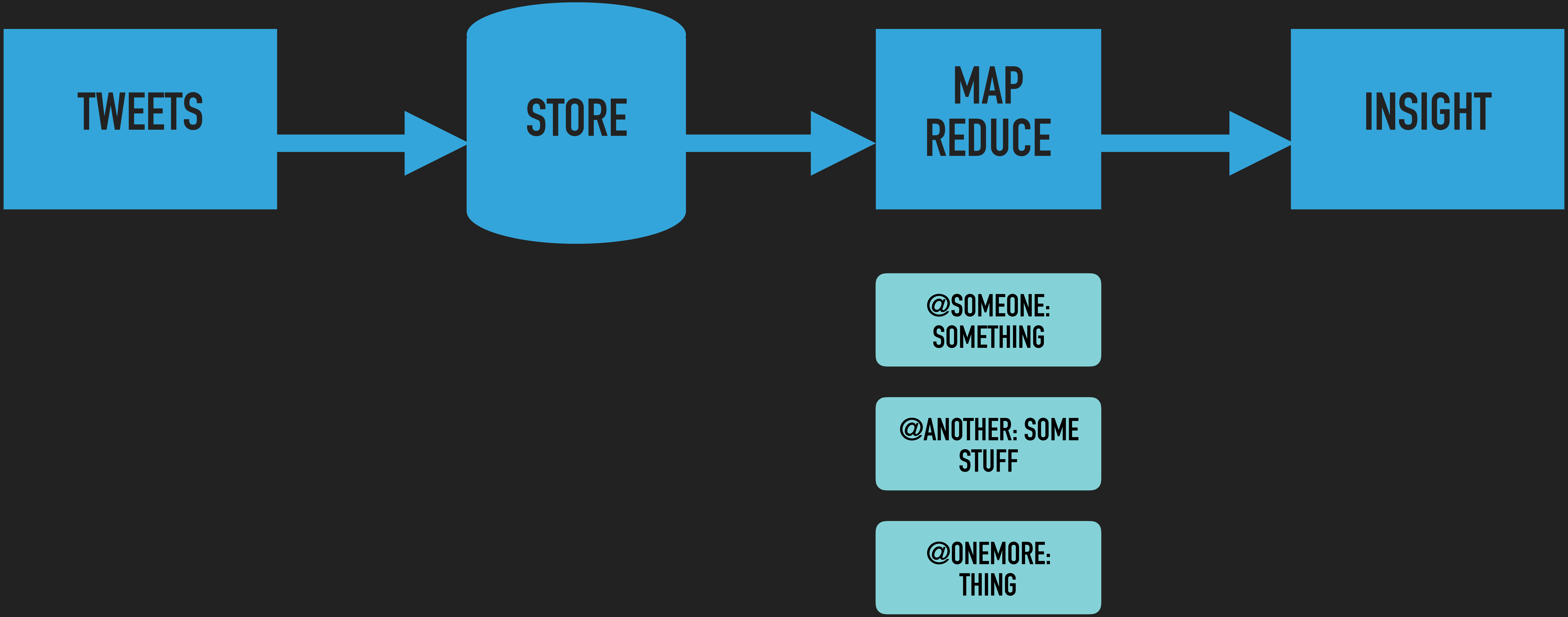
INSIGHT

@SOMEONE: SOMETHING

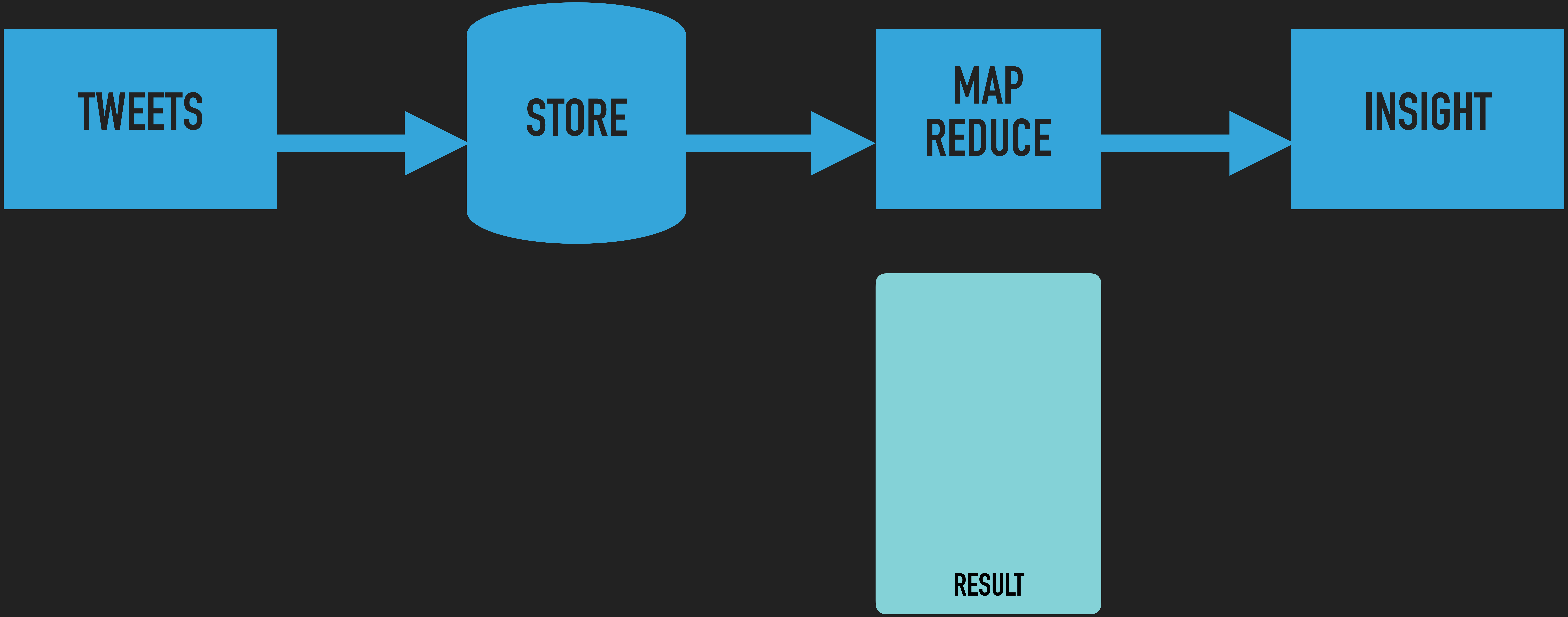
@ANOTHER: SOME STUFF

@ONEMORE: THING

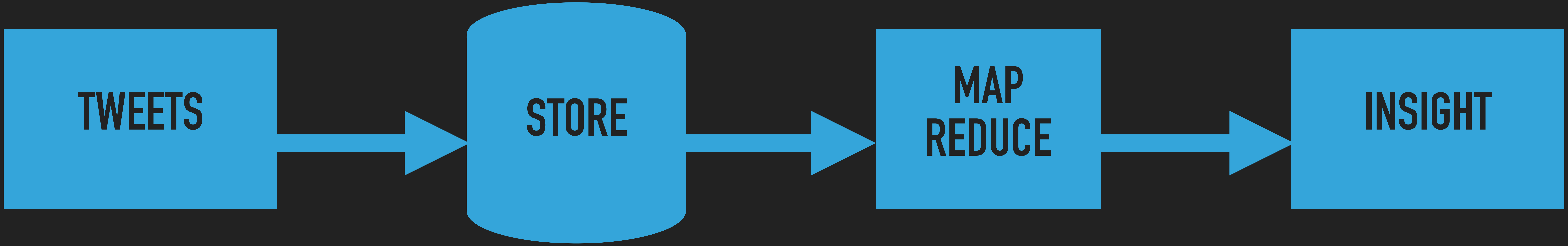
MAP REDUCE


TWEETS
STORE
MAP REDUCE
INSIGHT
@SOMEONE: SOMETHING
@ANOTHER: SOME STUFF
@ONEMORE: THING

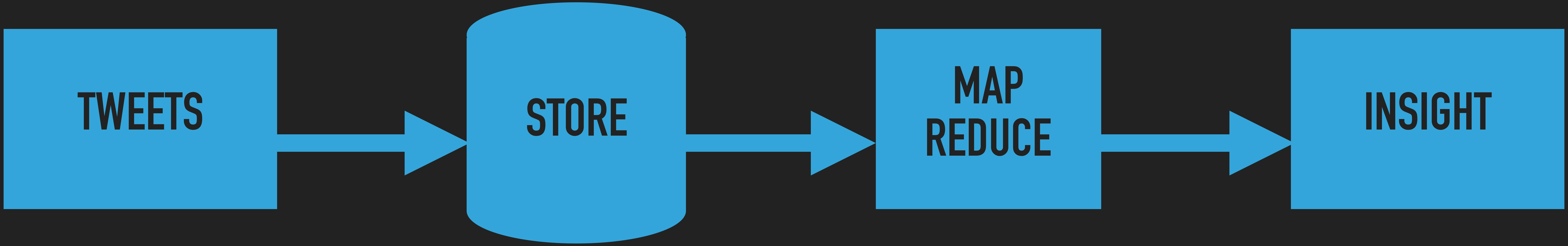
MAP REDUCE



MAP REDUCE



MAP REDUCE

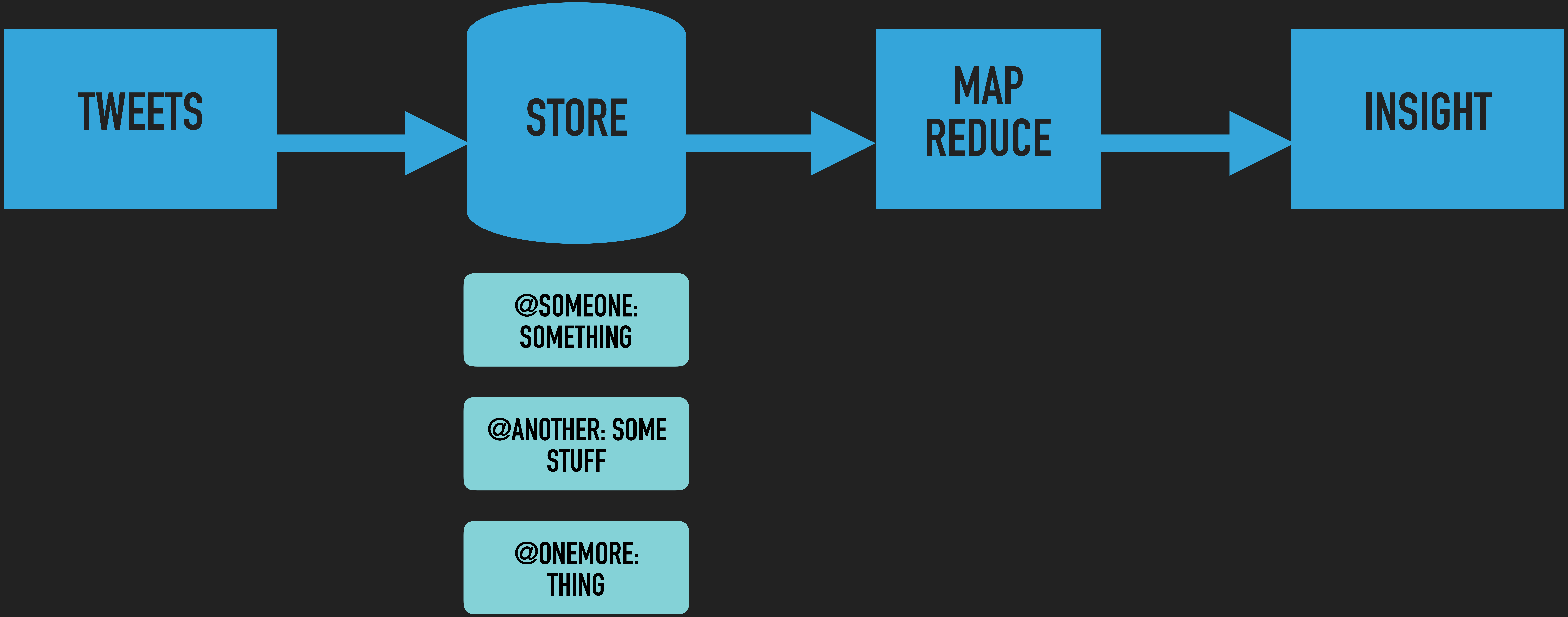


@SOMEONE: SOMETHING

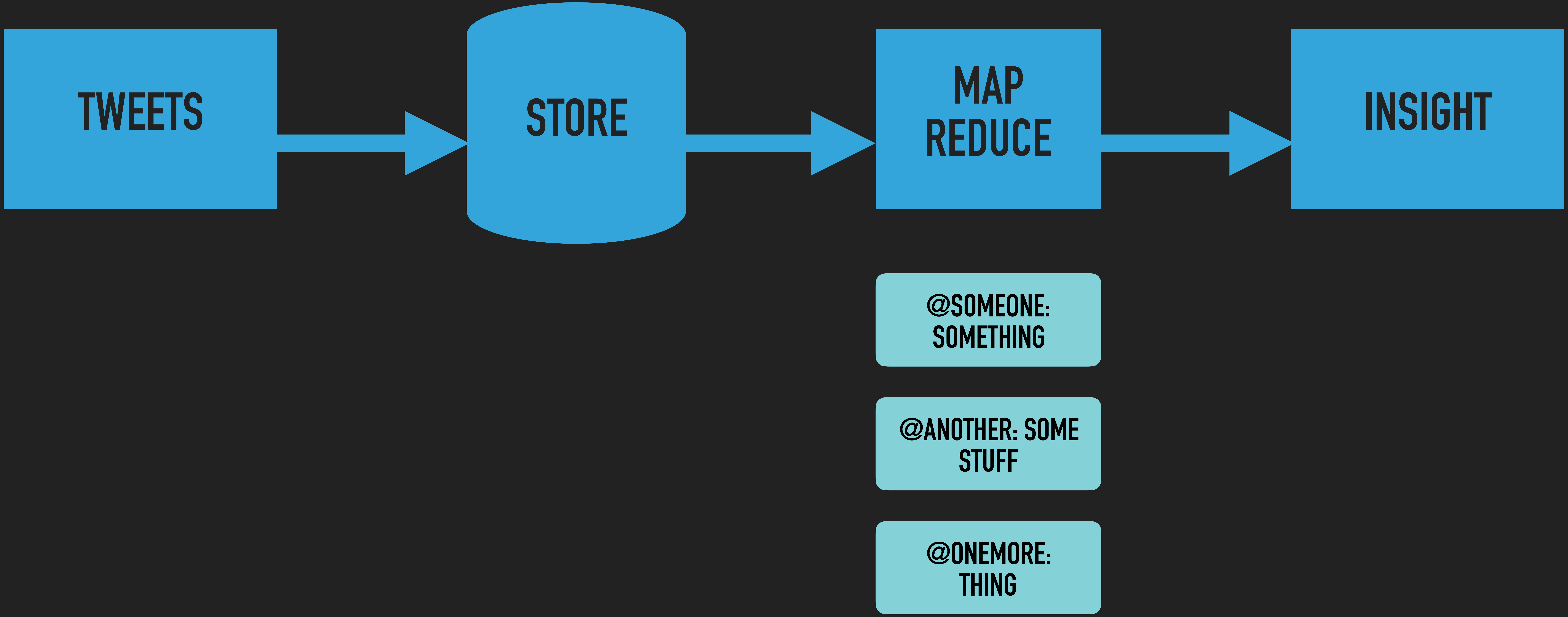
@ANOTHER: SOME STUFF

@ONEMORE: THING

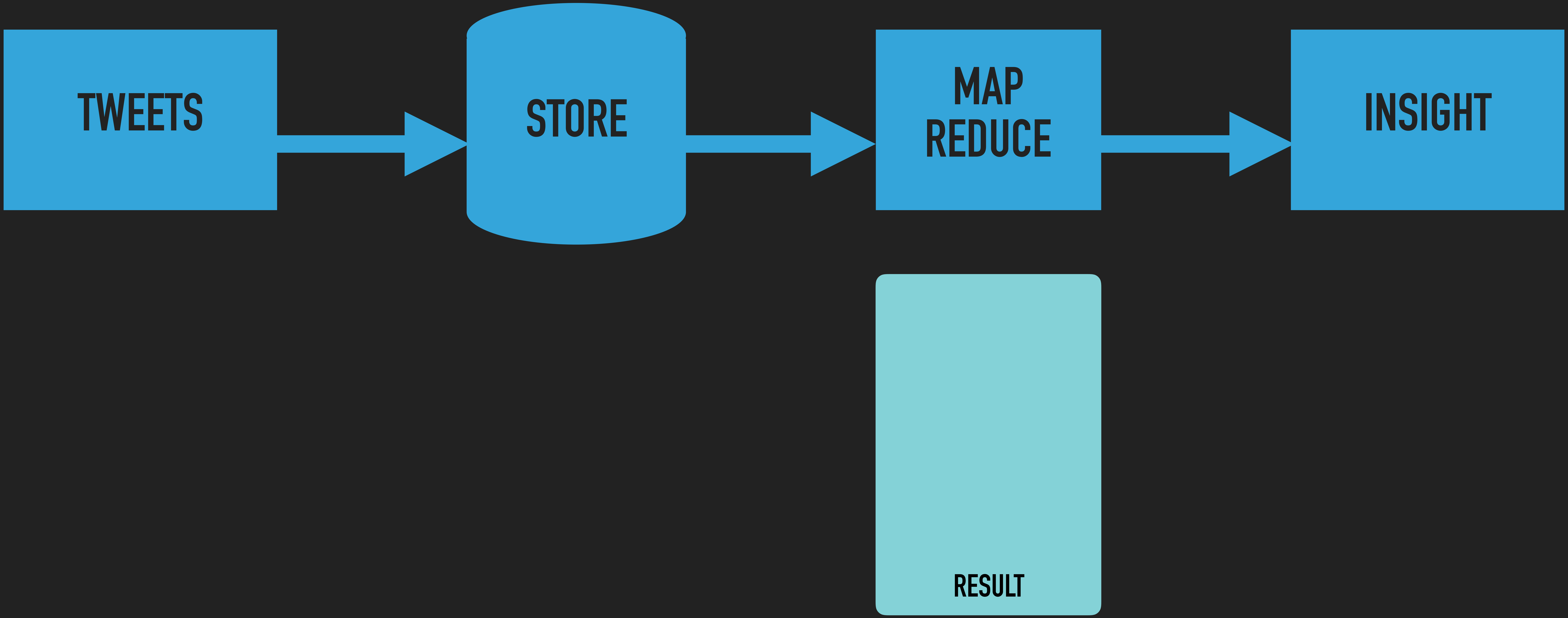
MAP REDUCE


TWEETS
STORE
MAP REDUCE
INSIGHT
@SOMEONE: SOMETHING
@ANOTHER: SOME STUFF
@ONEMORE: THING

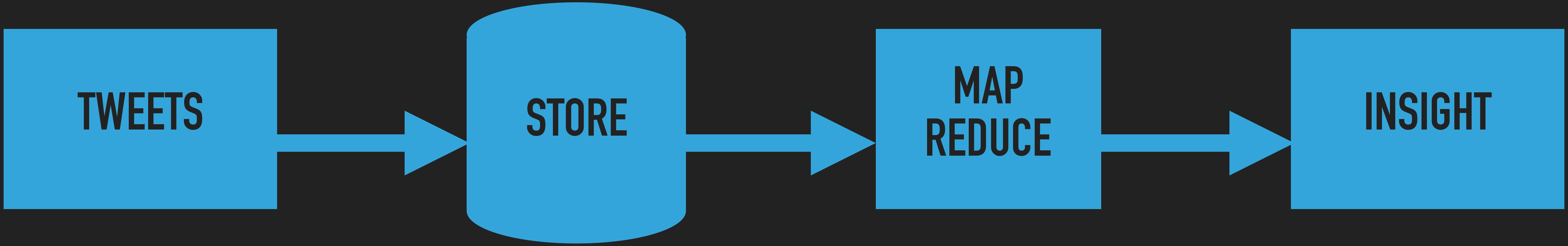
MAP REDUCE


TWEETS
STORE
MAP REDUCE
INSIGHT
@SOMEONE: SOMETHING
@ANOTHER: SOME STUFF
@ONEMORE: THING

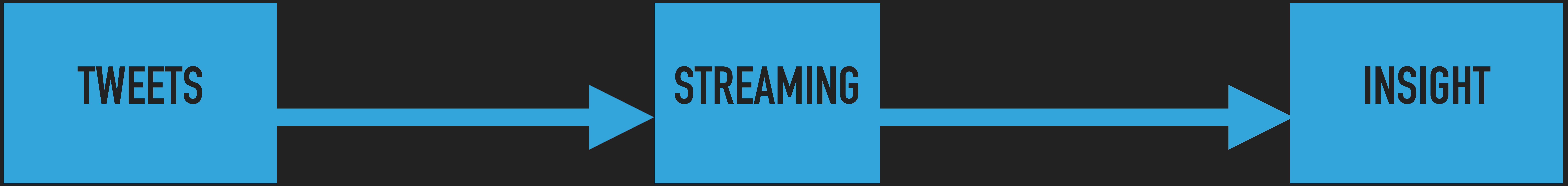
MAP REDUCE



MAP REDUCE



STREAMING

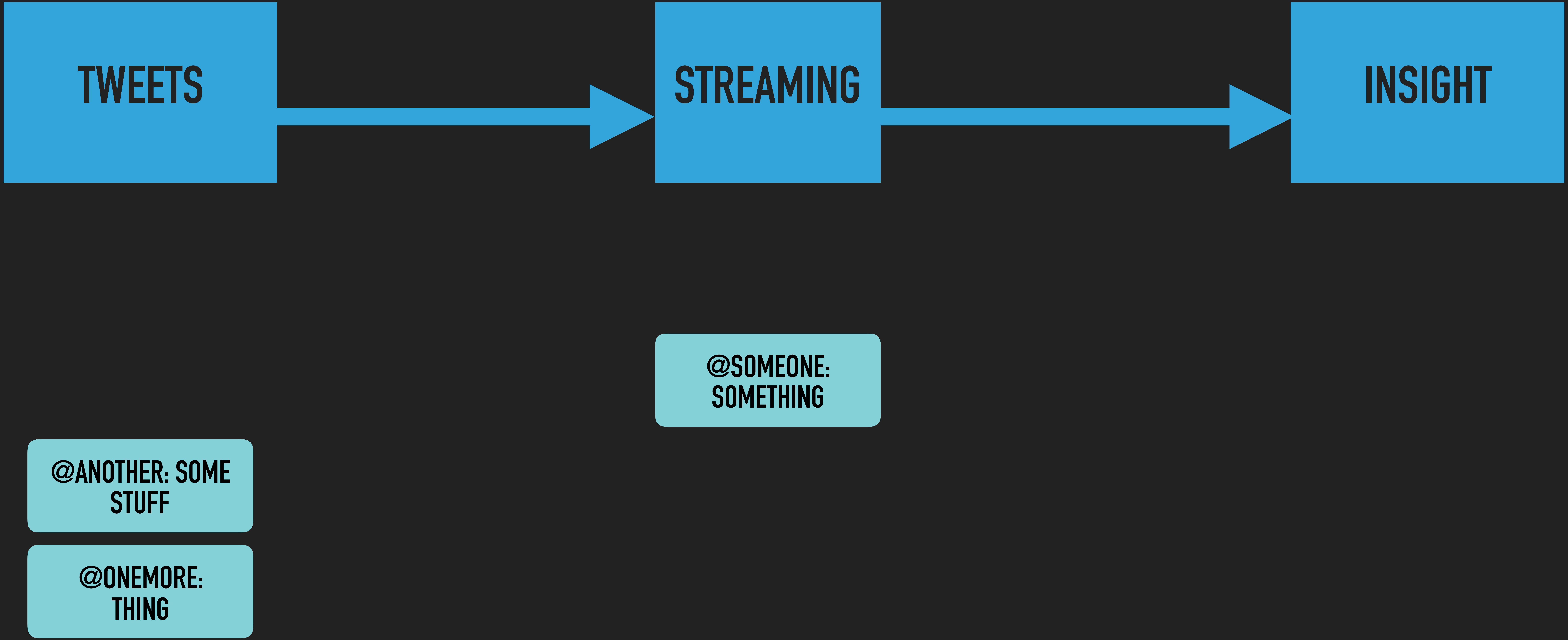


@SOMEONE: SOMETHING

@ANOTHER: SOME STUFF

@ONEMORE: THING

STREAMING



TWEETS

STREAMING

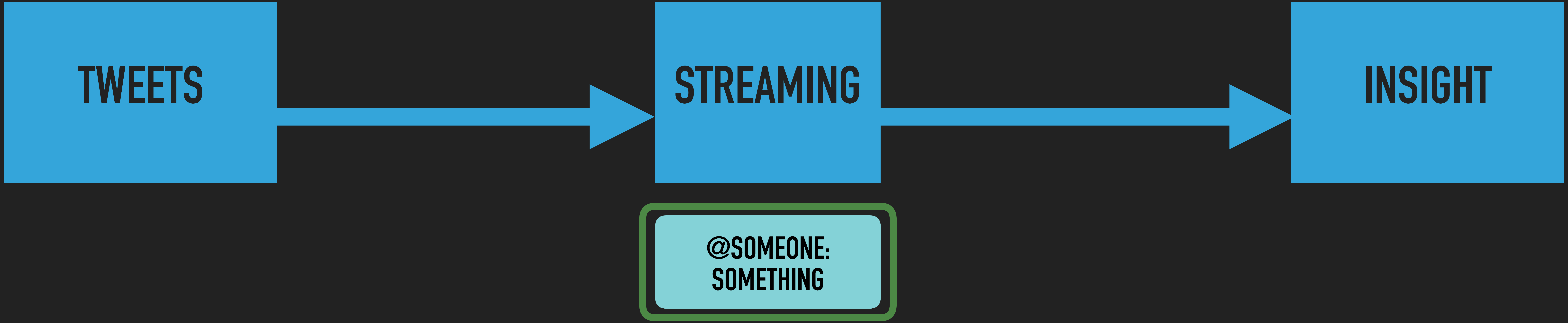
INSIGHT

@SOMEONE: SOMETHING

@ANOTHER: SOME STUFF

@ONEMORE: THING

STREAMING

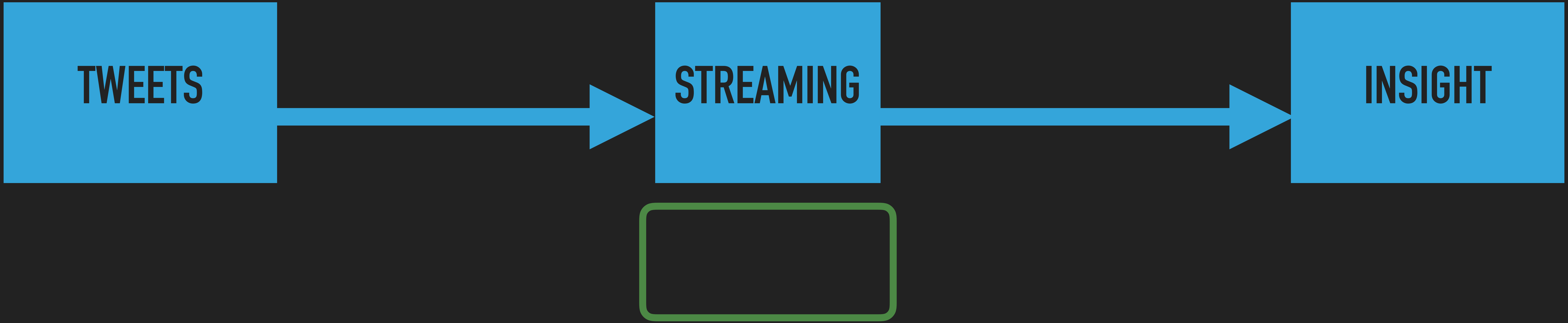


@ANOTHER: SOME STUFF

@ONEMORE: THING

CONTEXT: AGGREGATION

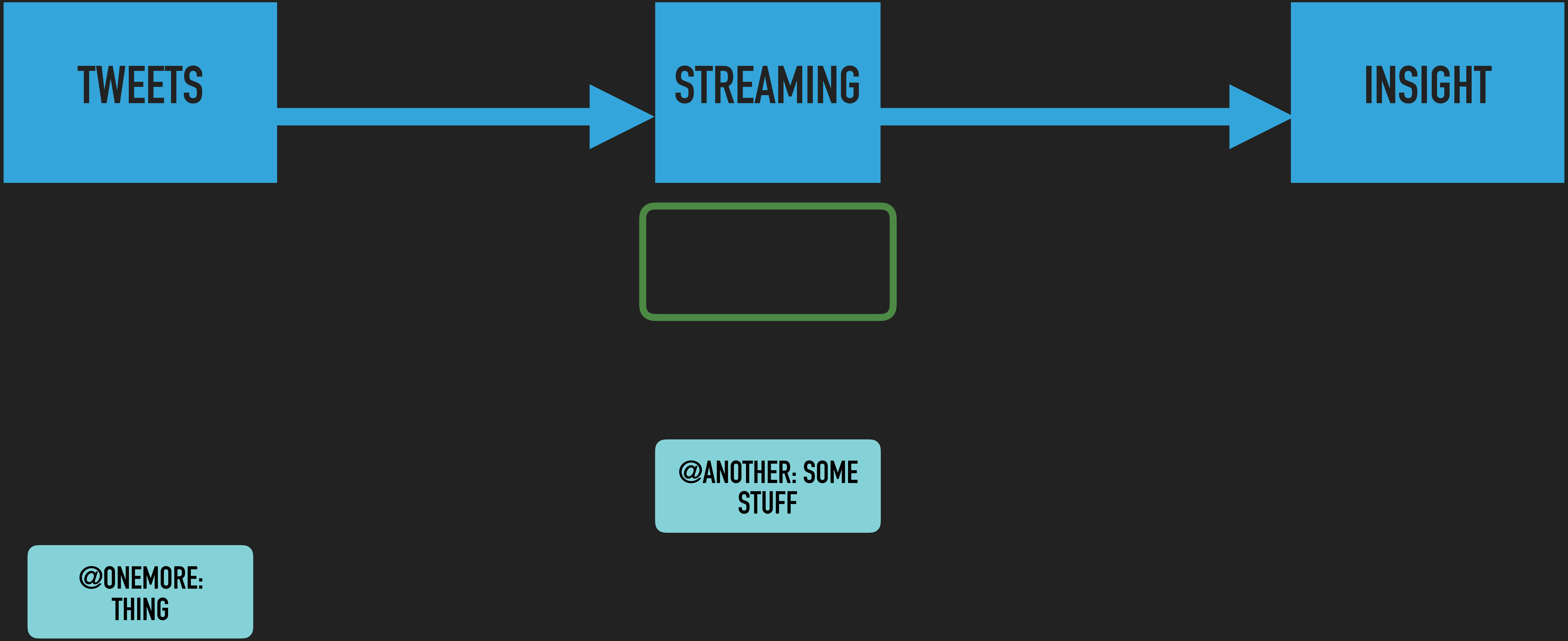
STREAMING



@ANOTHER: SOME STUFF

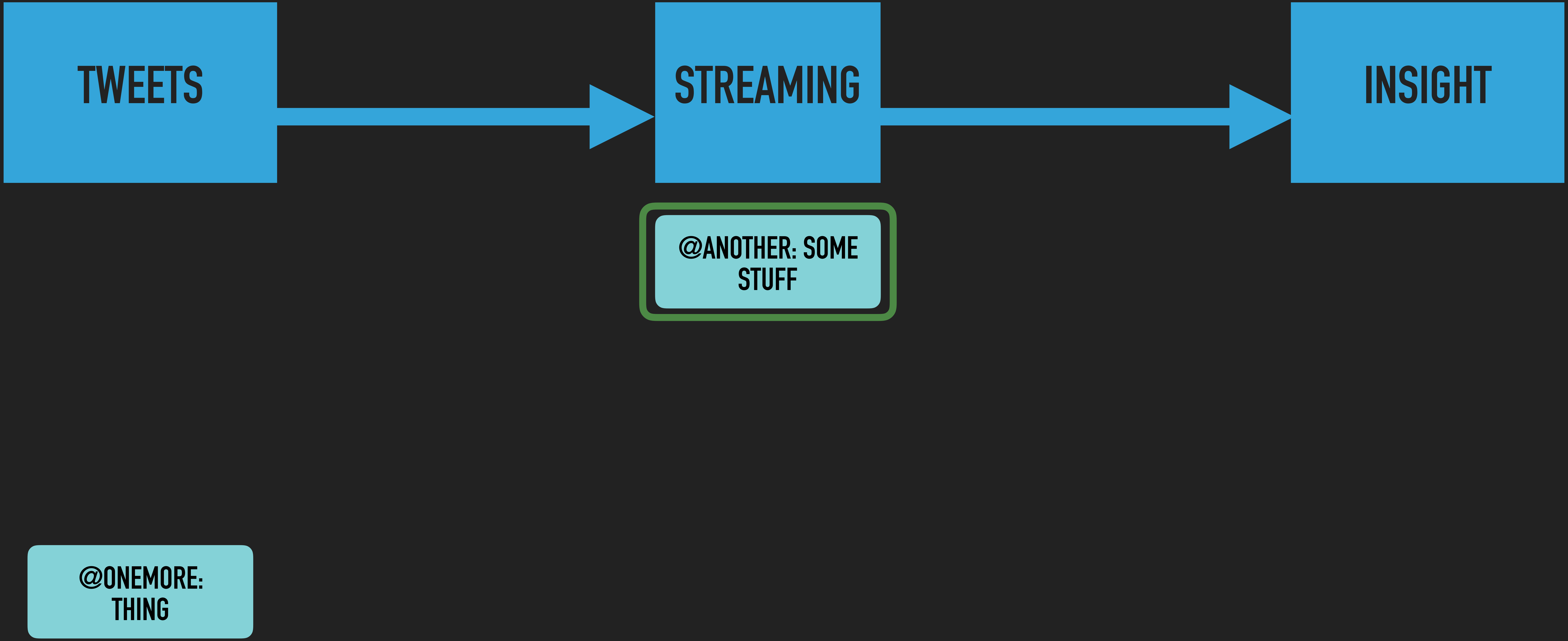
@ONEMORE: THING

STREAMING

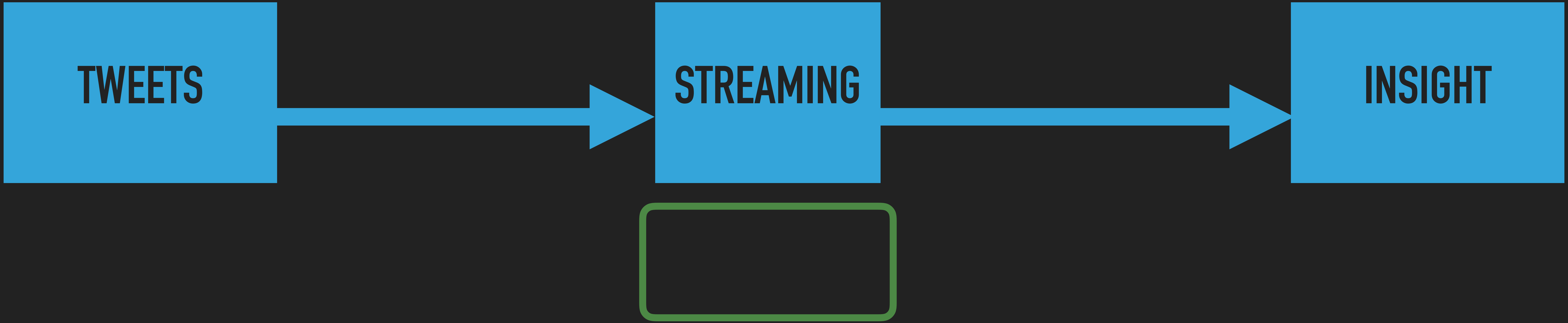


TWEETS
STREAMING
INSIGHT
@ANOTHER: SOME STUFF
@ONEMORE: THING

STREAMING

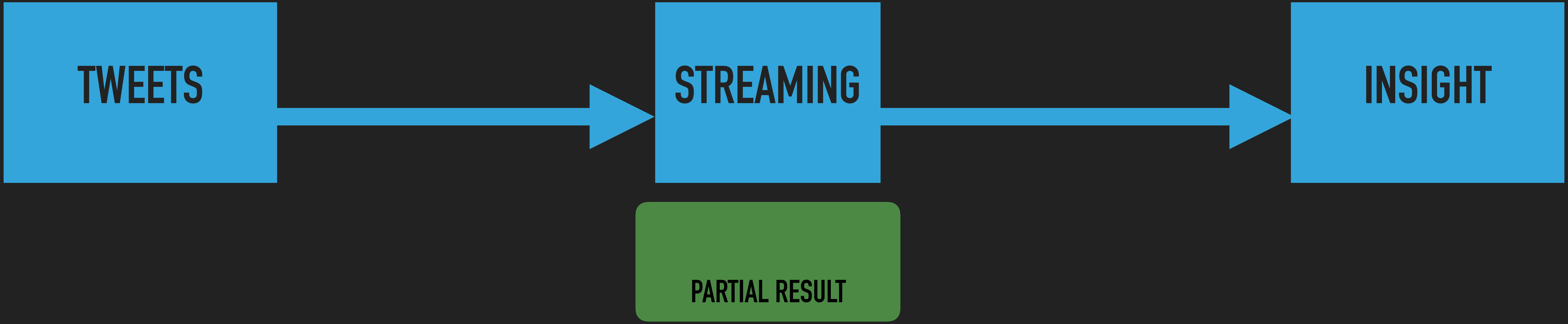

TWEETS
STREAMING
INSIGHT
@ANOTHER: SOME STUFF
@ONEMORE: THING

STREAMING



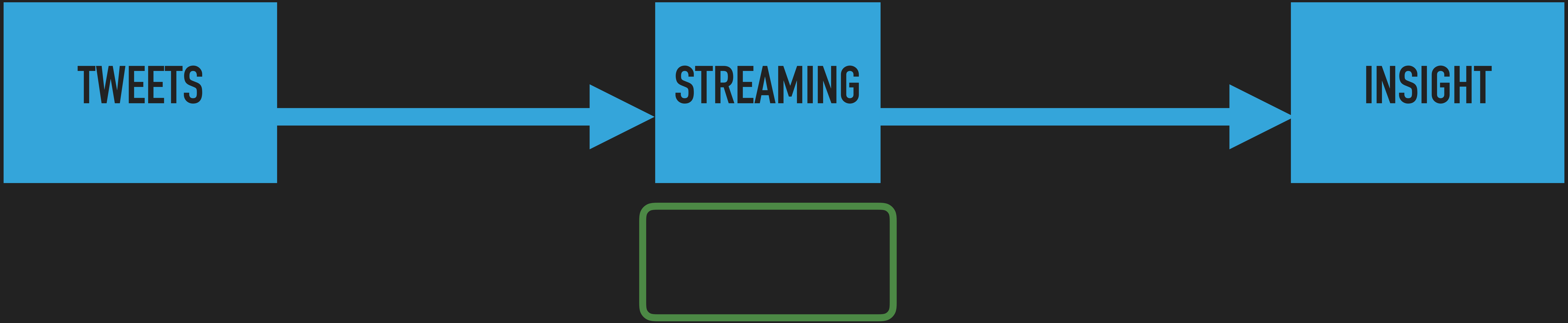
@ONEMORE:
THING

STREAMING



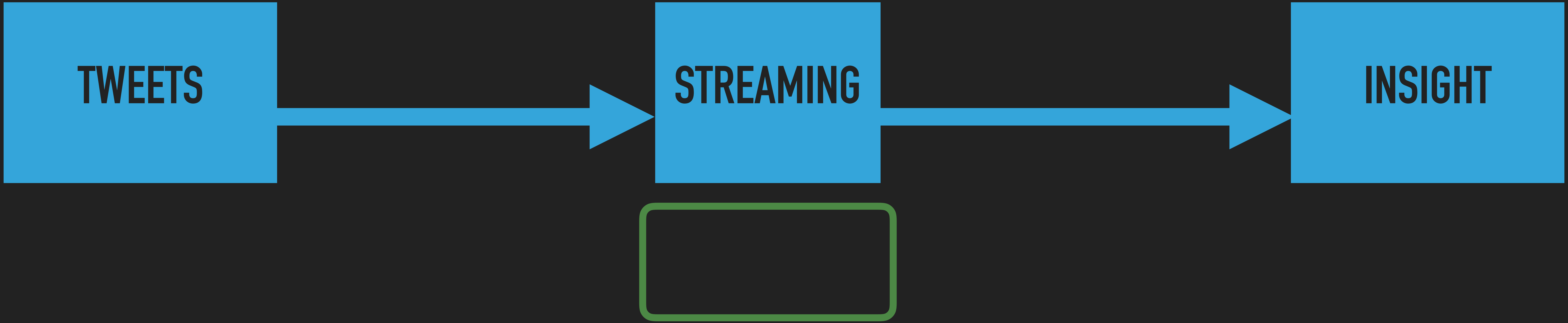
@ONEMORE:
THING

STREAMING



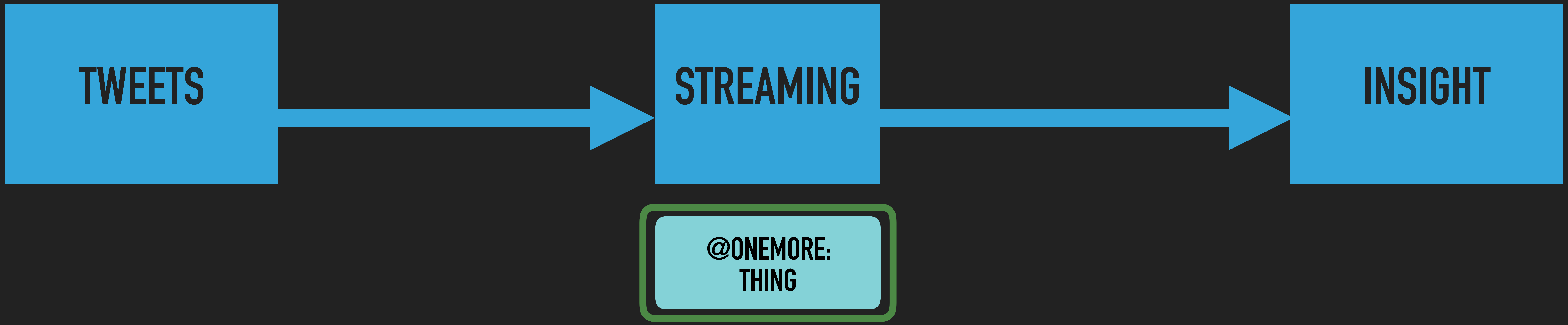
@ONEMORE:
THING

STREAMING



@ONEMORE:
THING

STREAMING



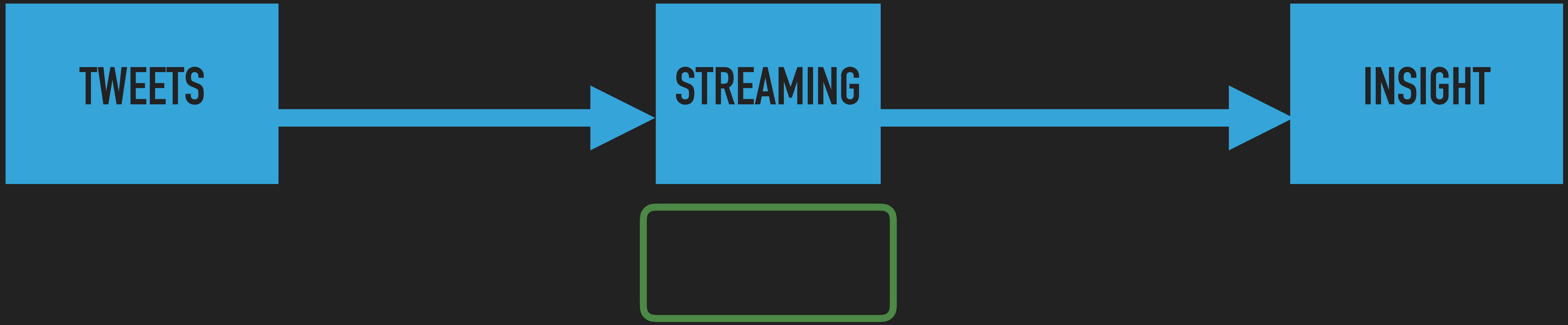
TWEETS

STREAMING

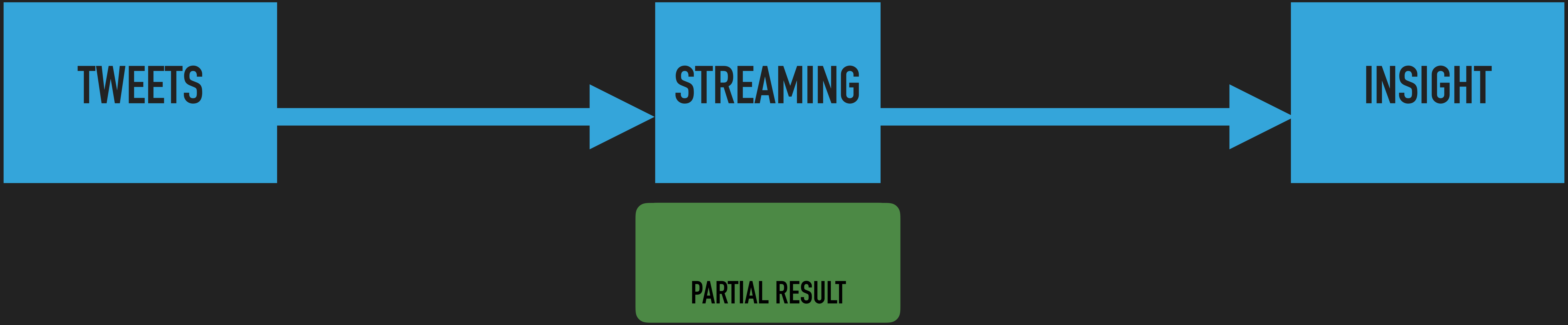
INSIGHT

@ONEMORE:
THING

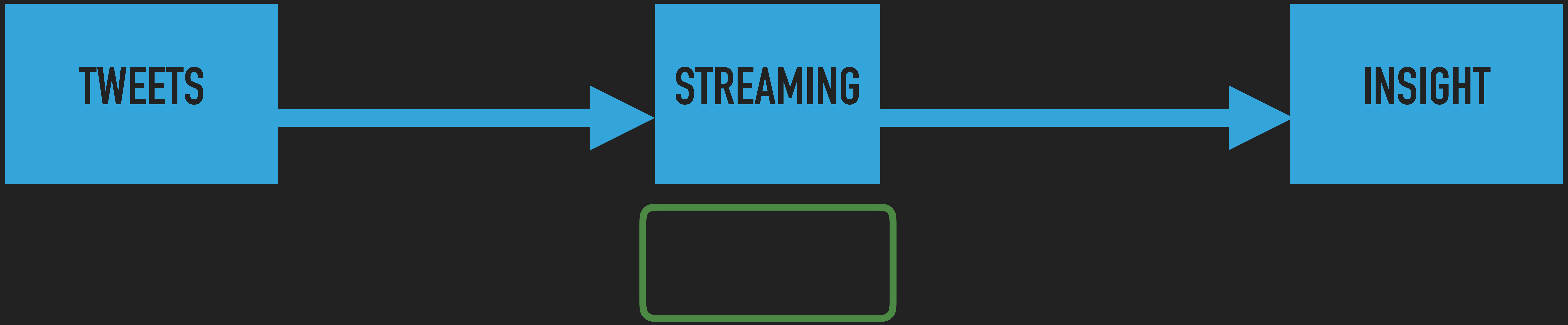
STREAMING



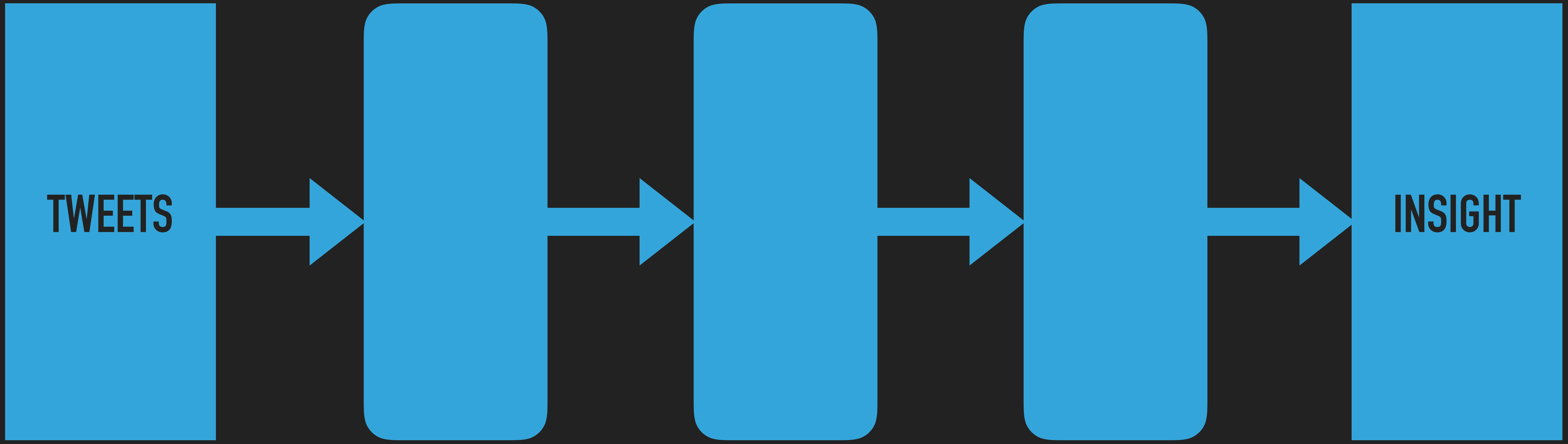
STREAMING



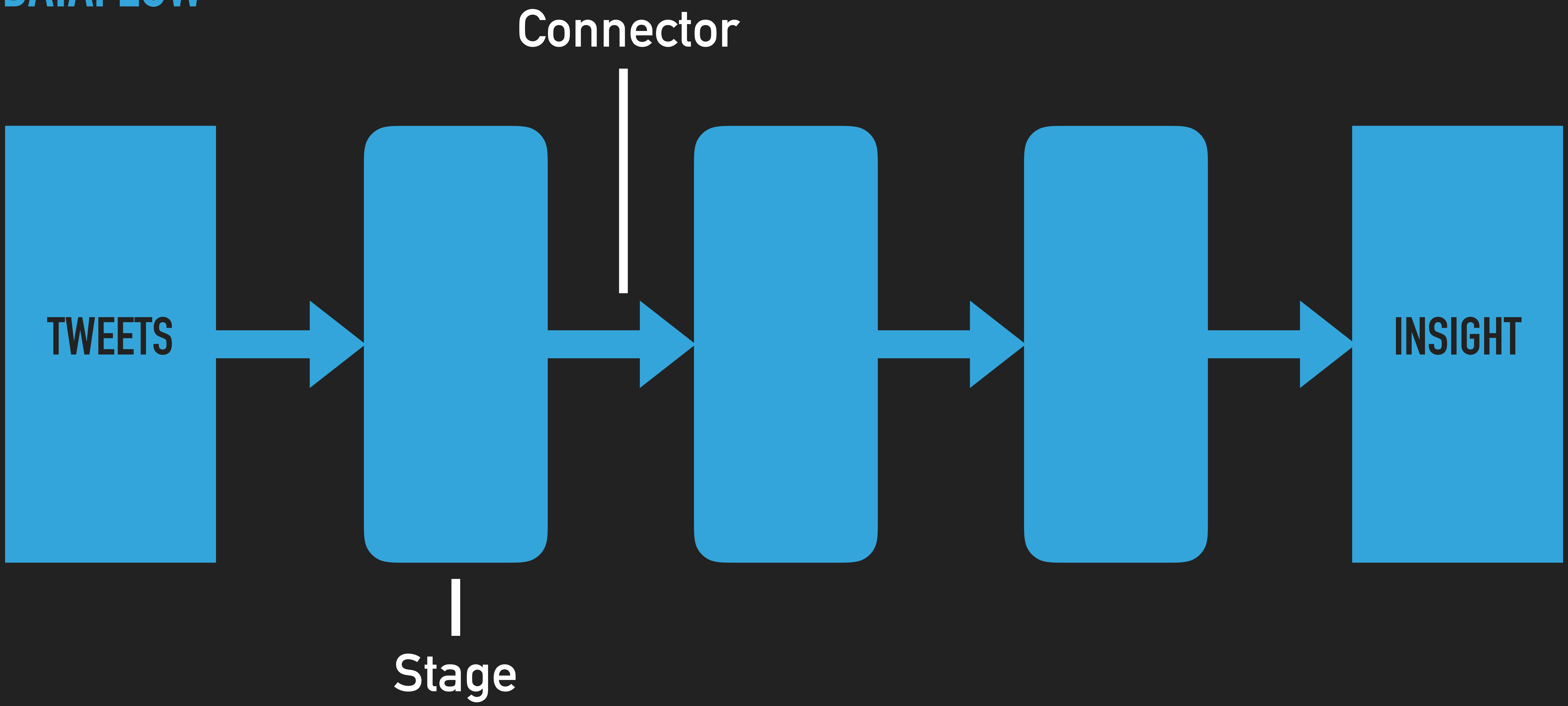
STREAMING



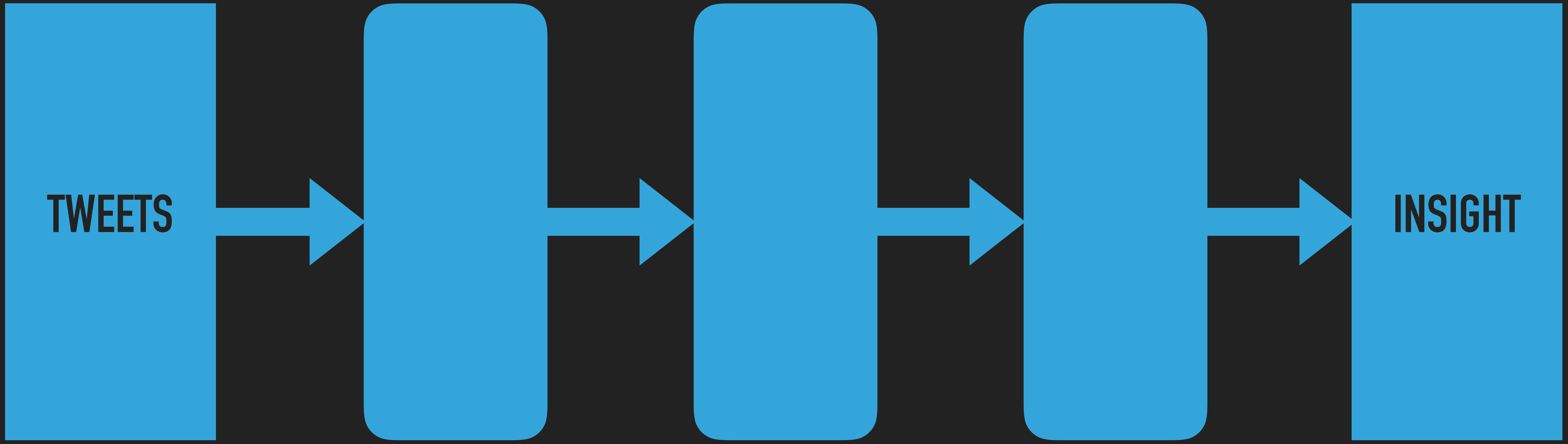
DATAFLOW



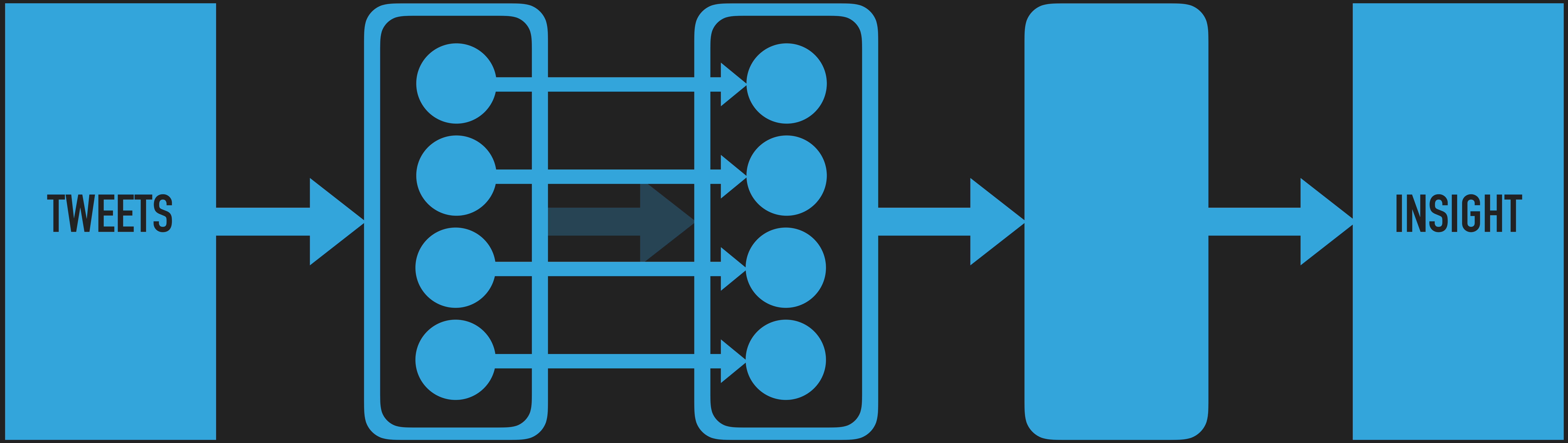
DATAFLOW



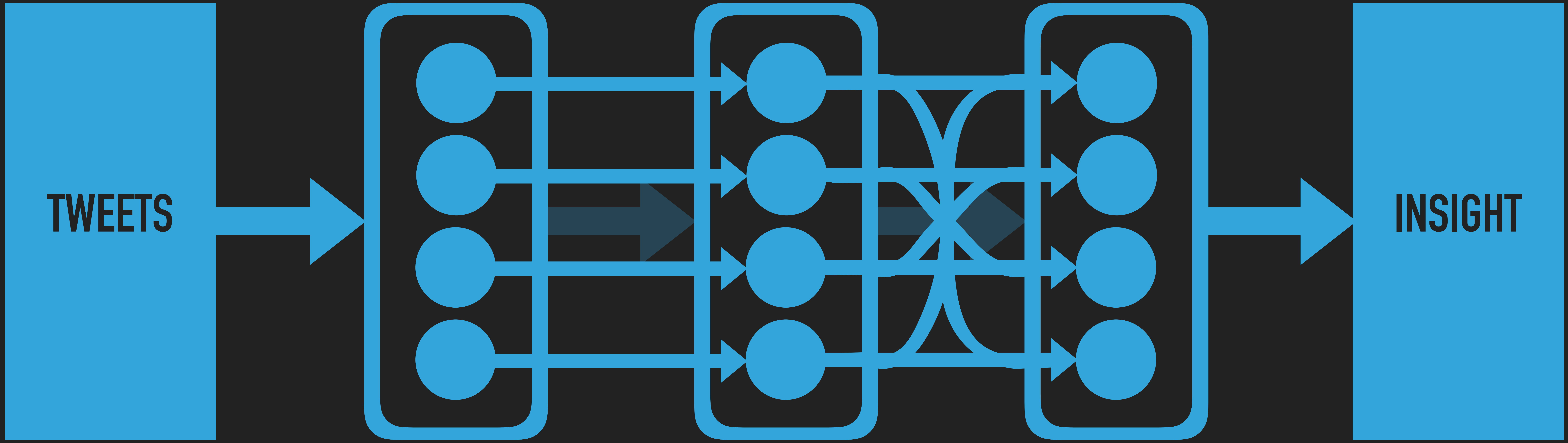
DATAFLOW: PARALLELISM



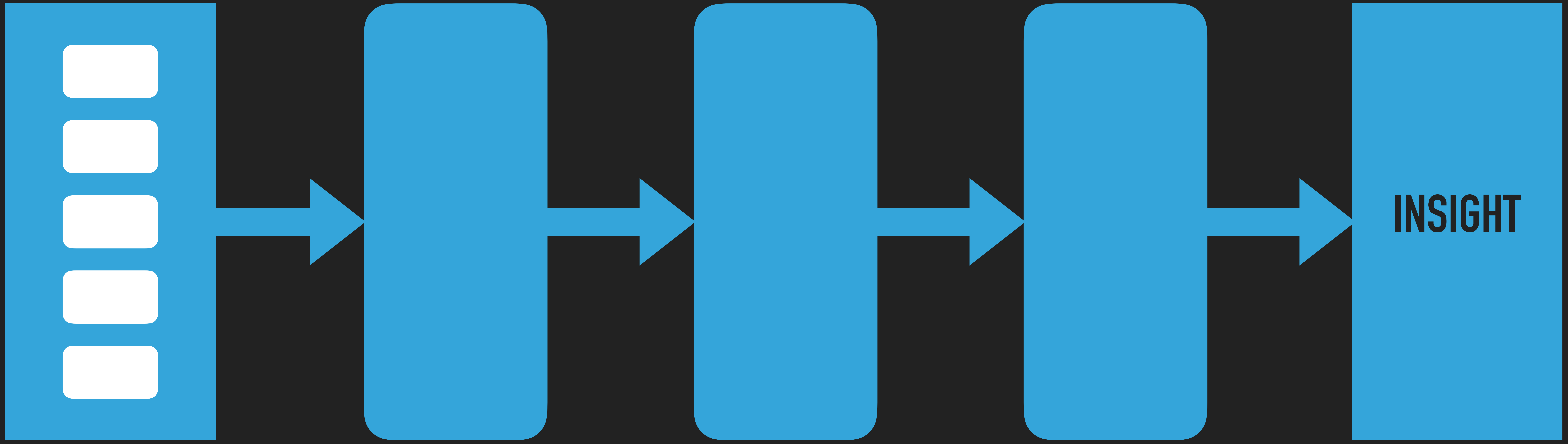
DATAFLOW: PARALLELISM



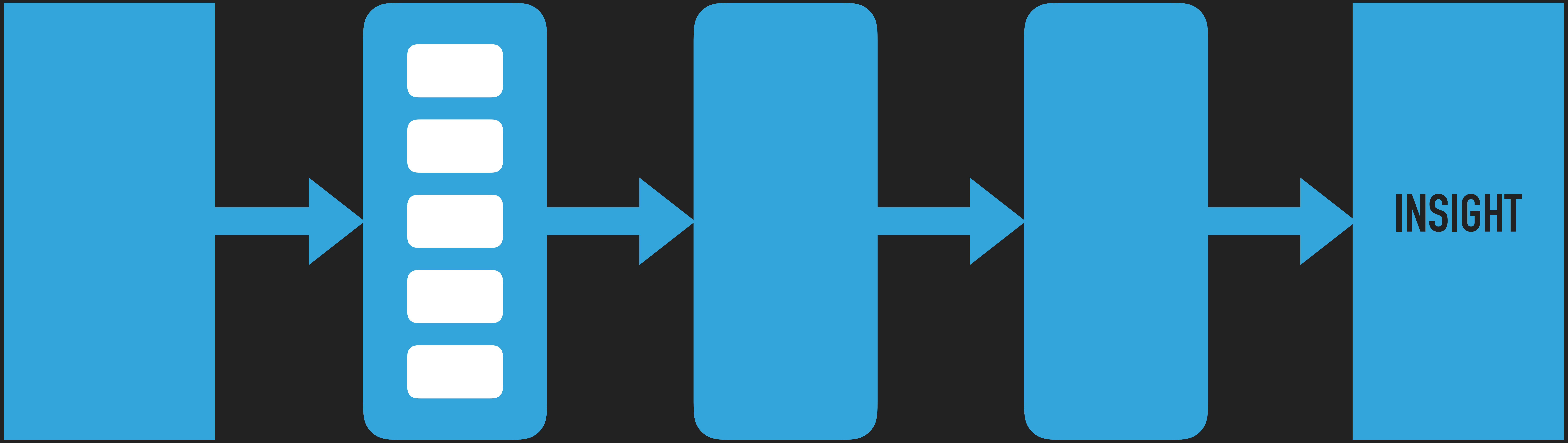
DATAFLOW: PARALLELISM



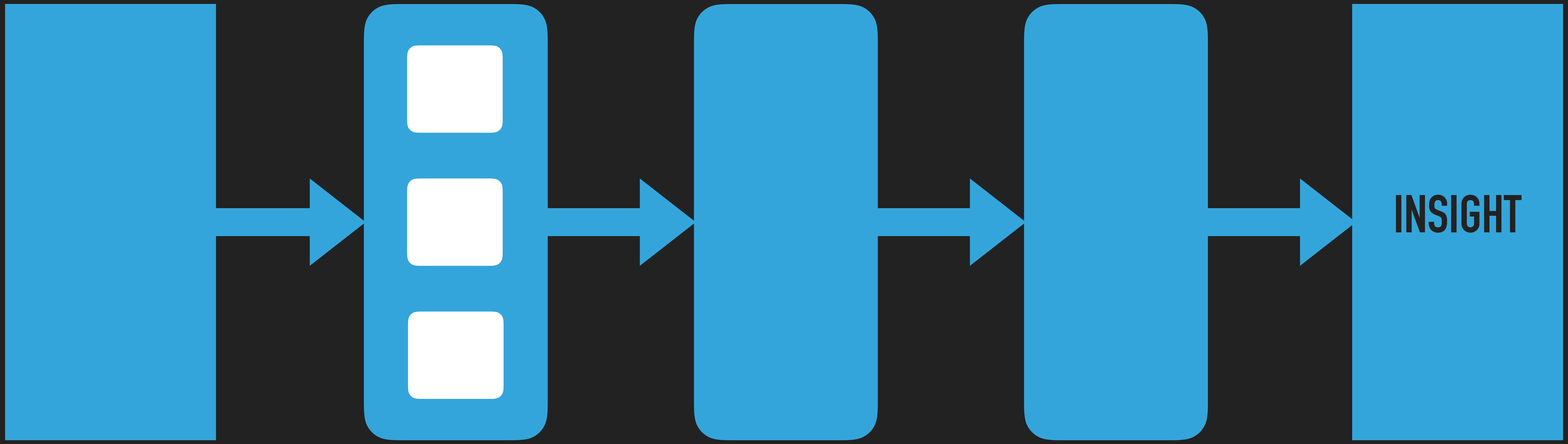
DATAFLOW: BATCH



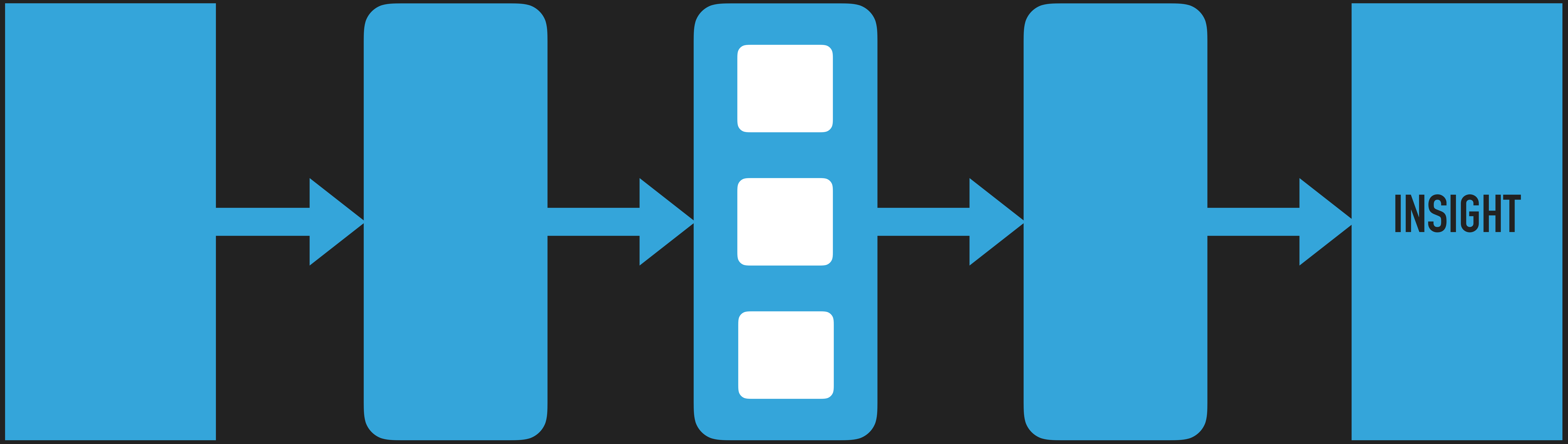
DATAFLOW: BATCH



DATAFLOW: BATCH

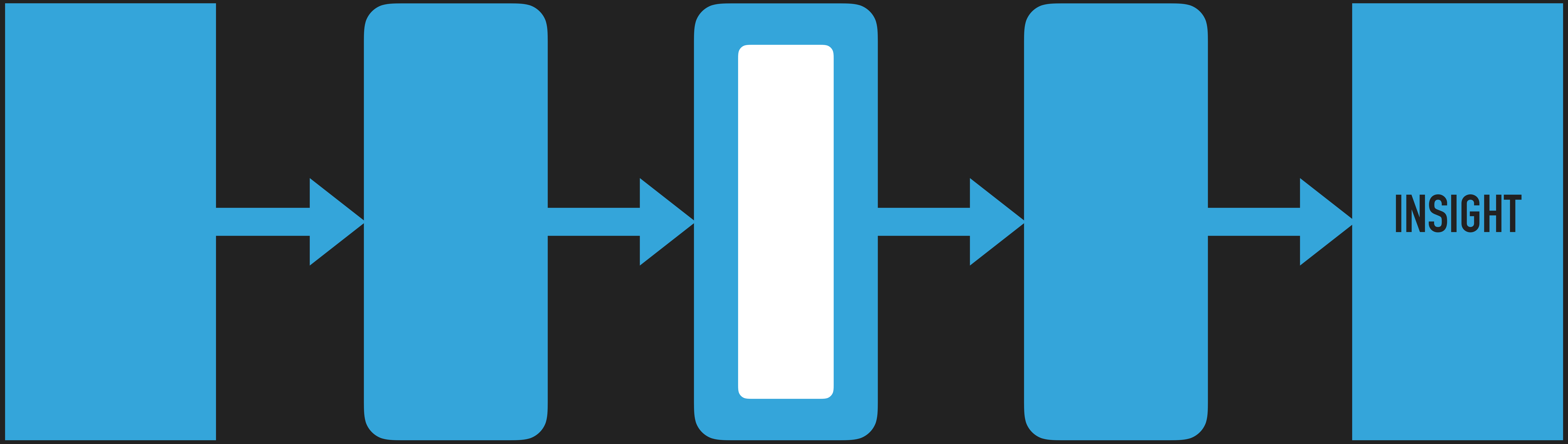


DATAFLOW: BATCH

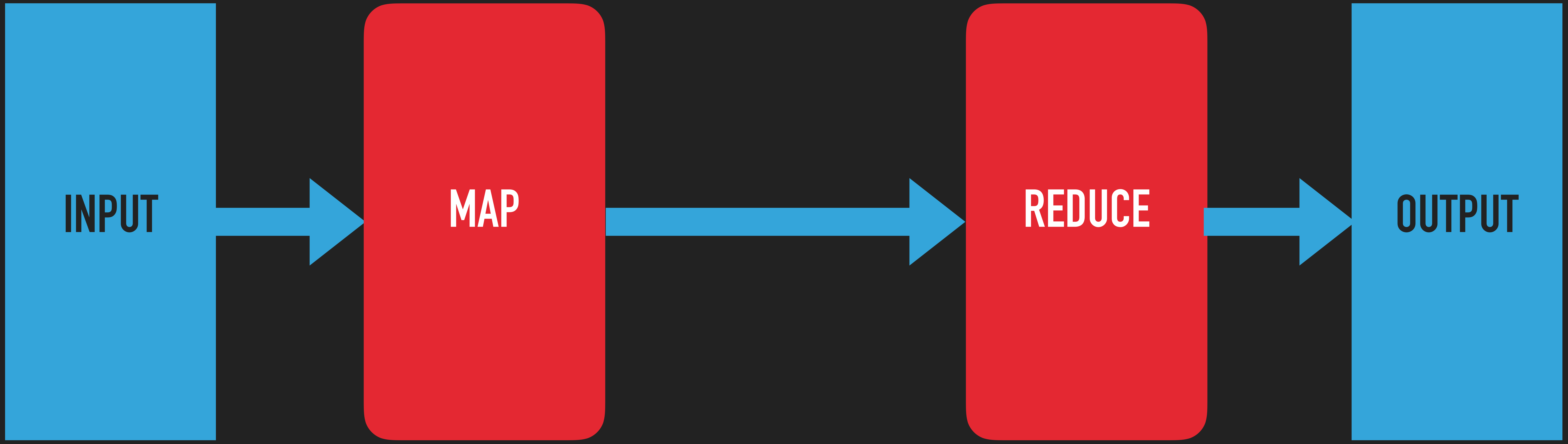


CONTEXT: PROGRAMMING MODEL

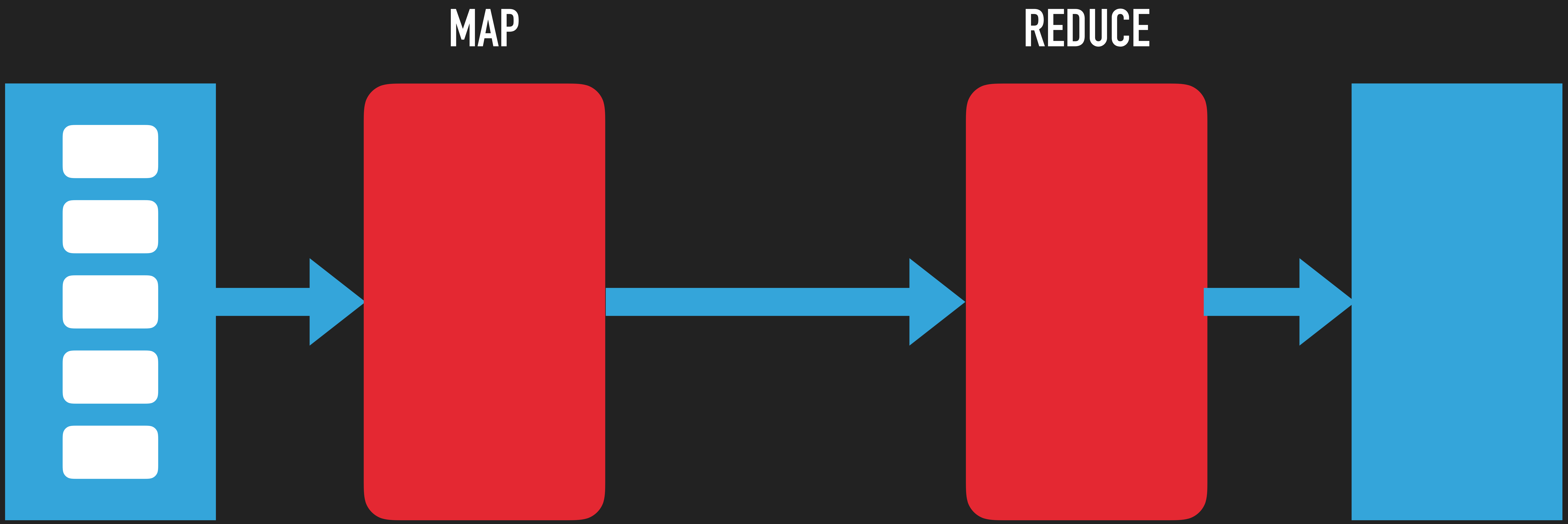
DATAFLOW: BATCH



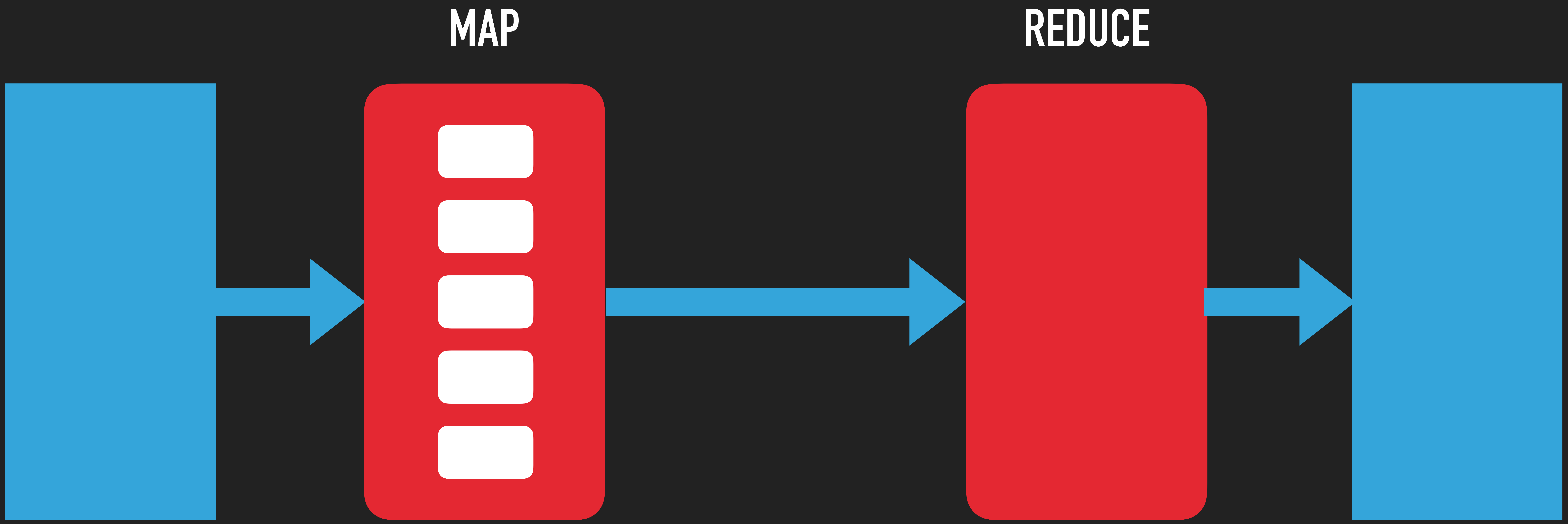
DATAFLOW: BATCH ➡ MAP/REDUCE



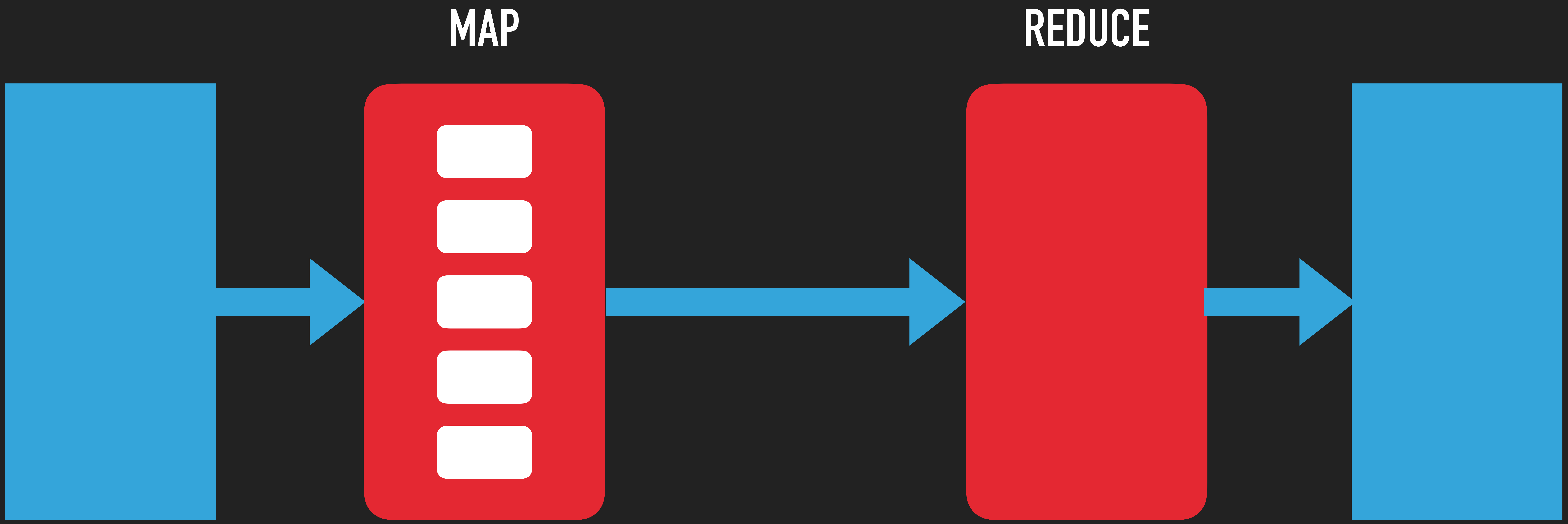
DATAFLOW: BATCH ➡ MAP/REDUCE



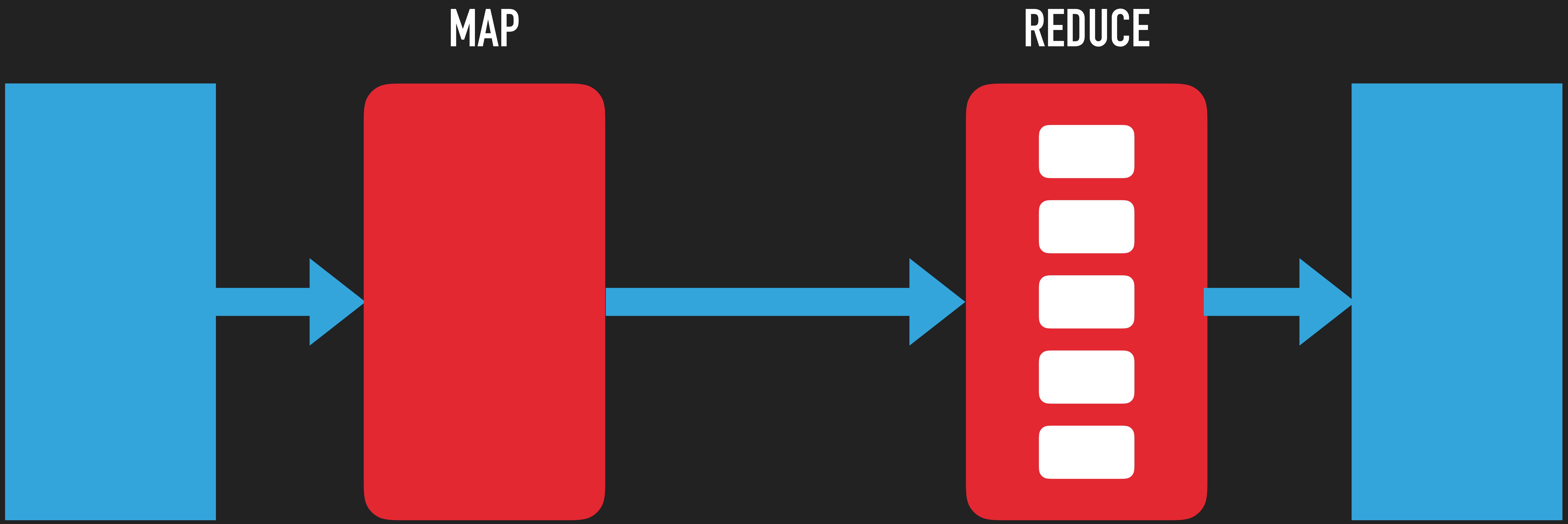
DATAFLOW: BATCH ➡ MAP/REDUCE



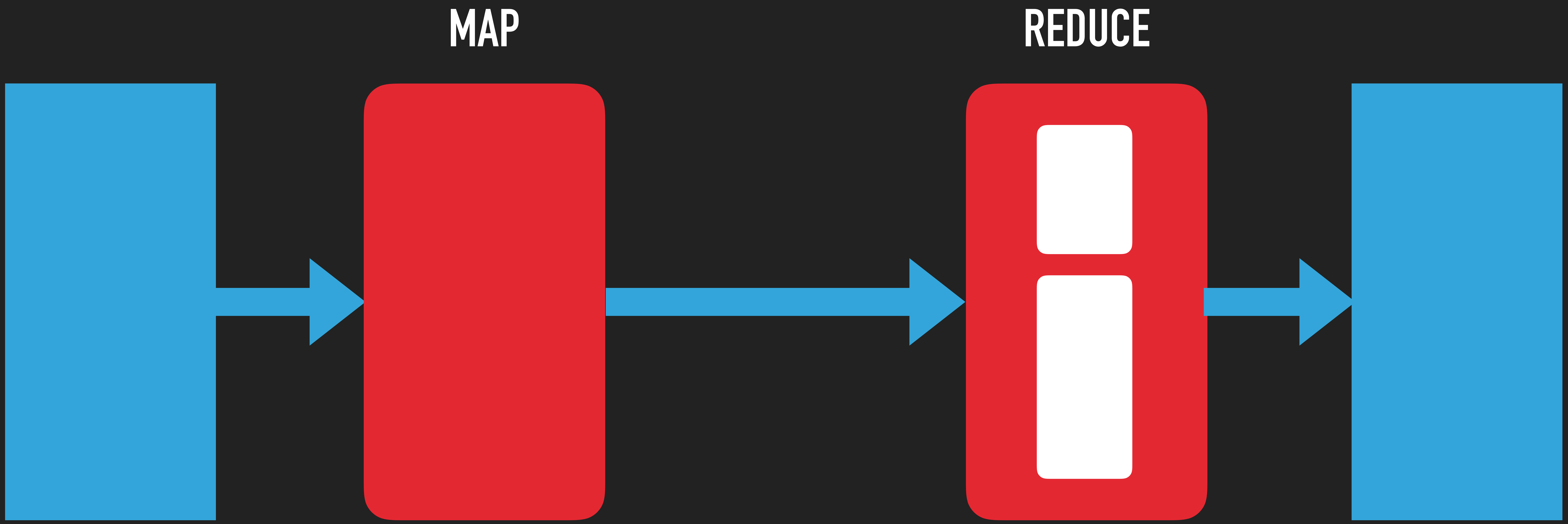
DATAFLOW: BATCH ➡ MAP/REDUCE



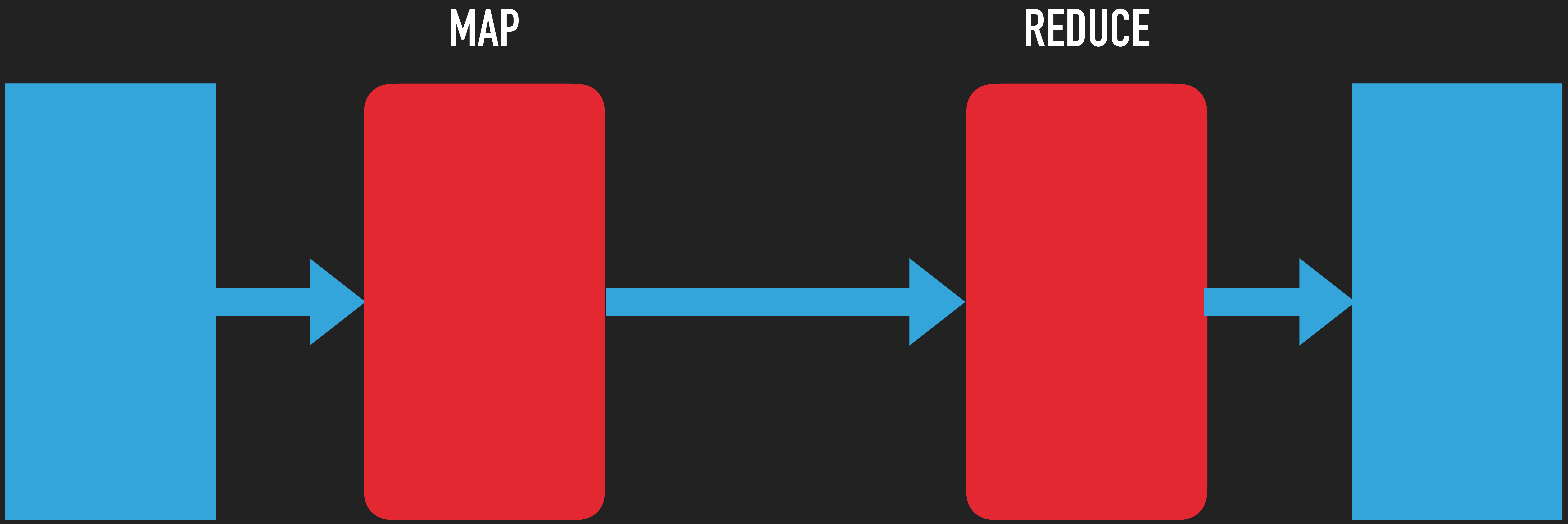
DATAFLOW: BATCH ➡ MAP/REDUCE



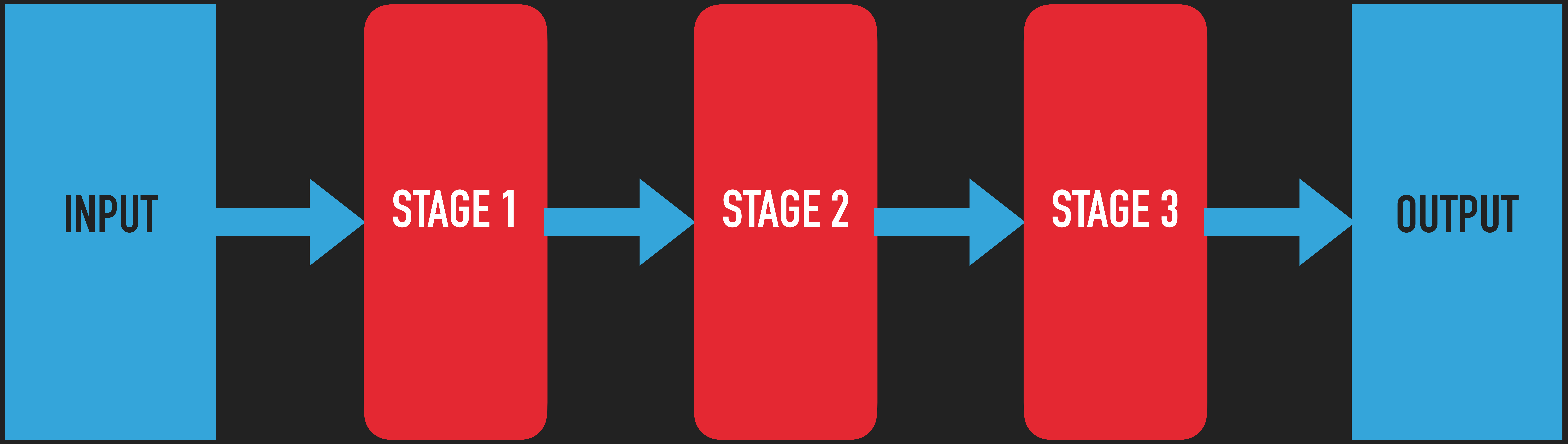
DATAFLOW: BATCH ➡ MAP/REDUCE



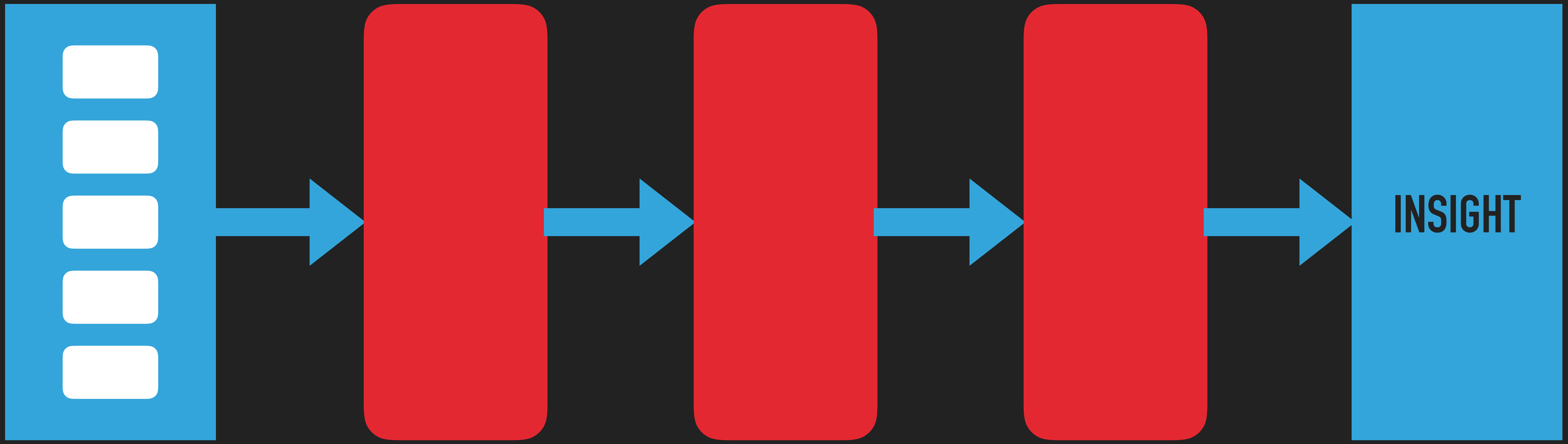
DATAFLOW: BATCH ➡ MAP/REDUCE



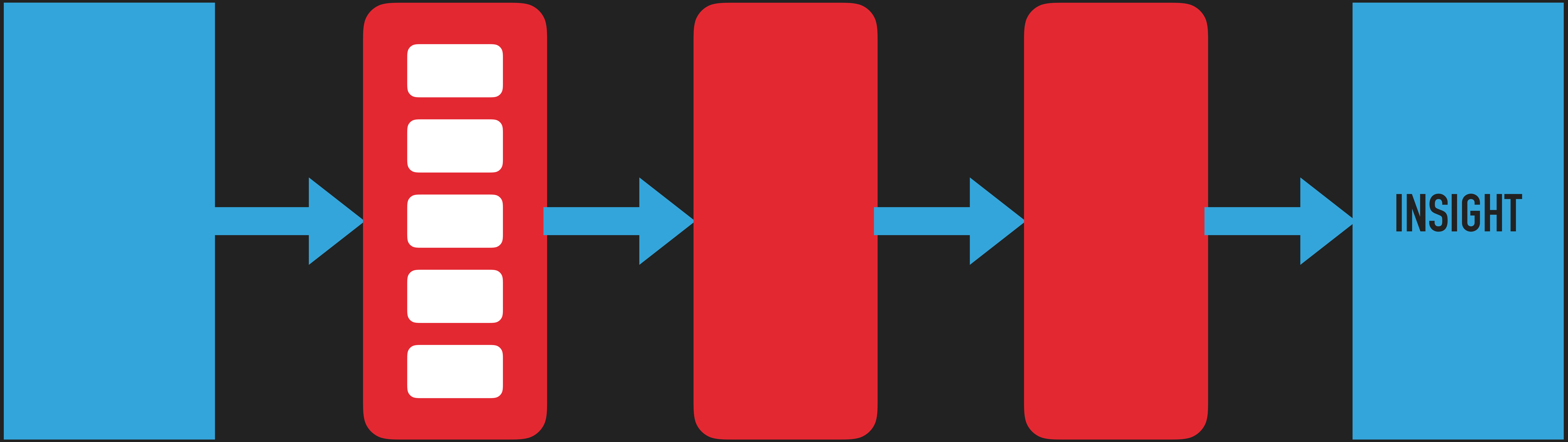
DATAFLOW: BATCH ➡ SPARK



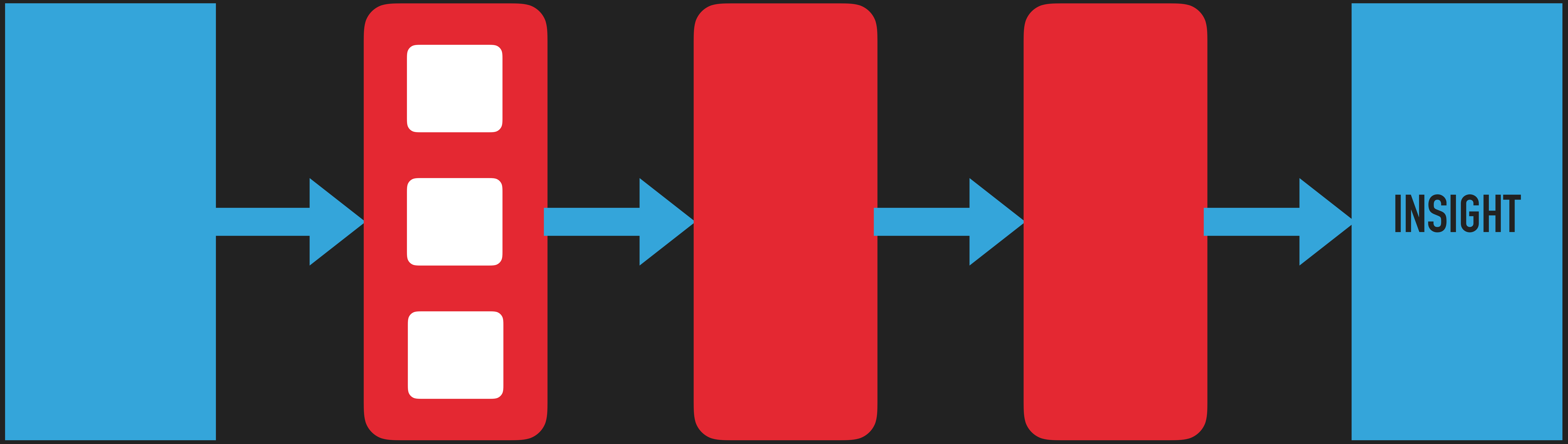
DATAFLOW: BATCH ➡ SPARK



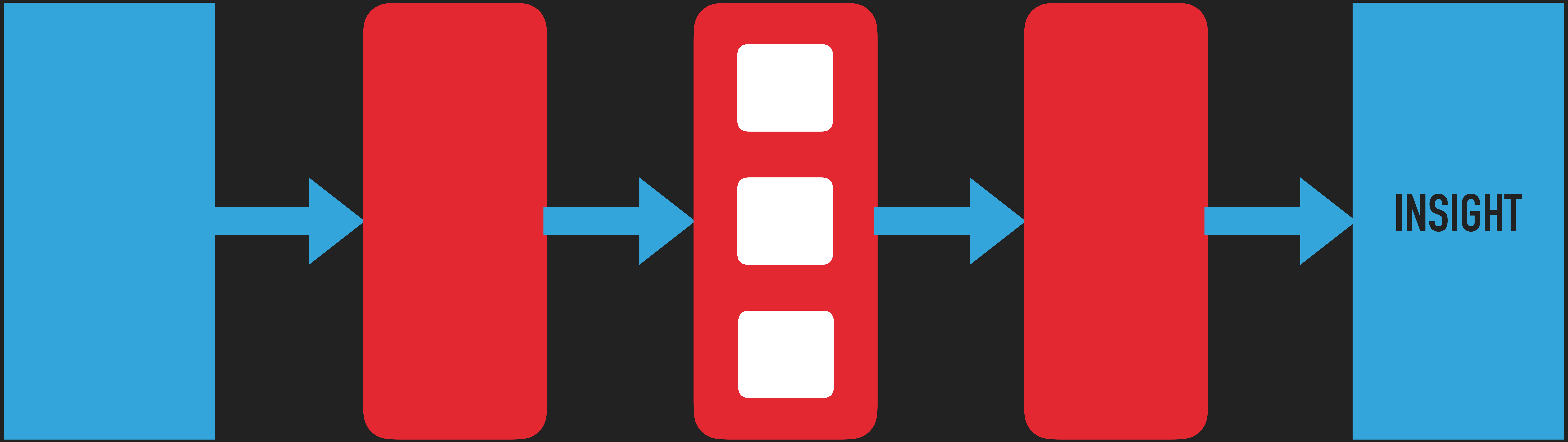
DATAFLOW: BATCH ➡ SPARK



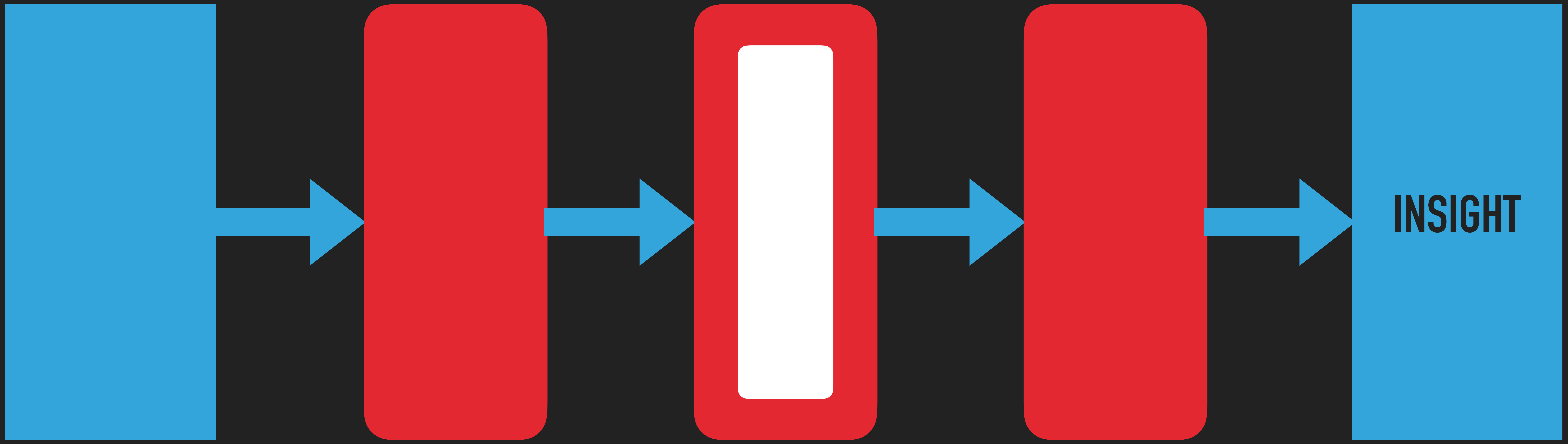
DATAFLOW: BATCH → SPARK



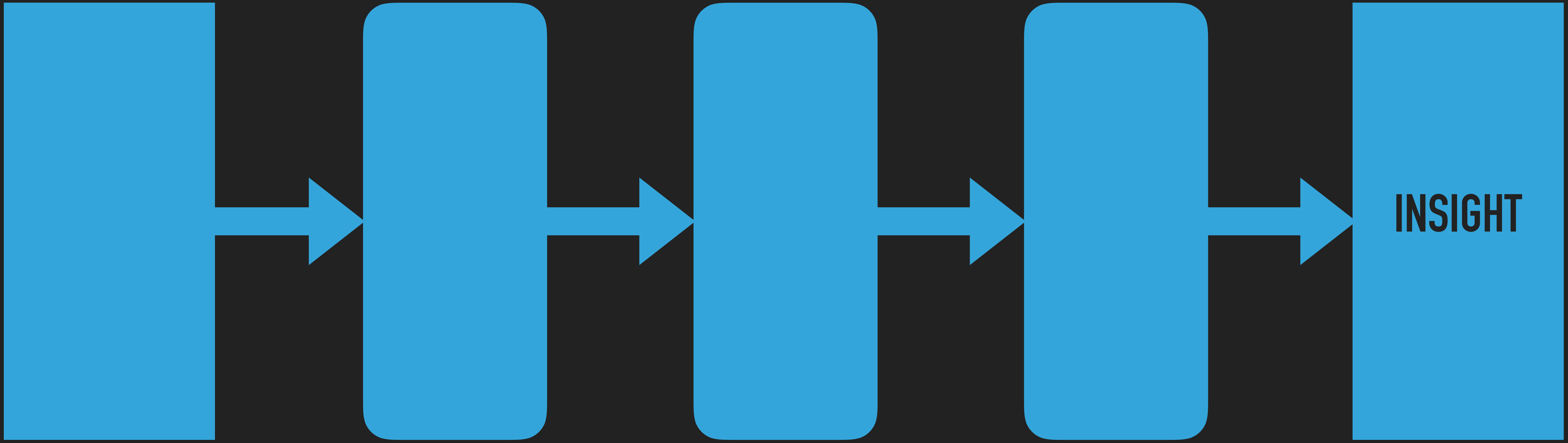
DATAFLOW: BATCH ➡ SPARK



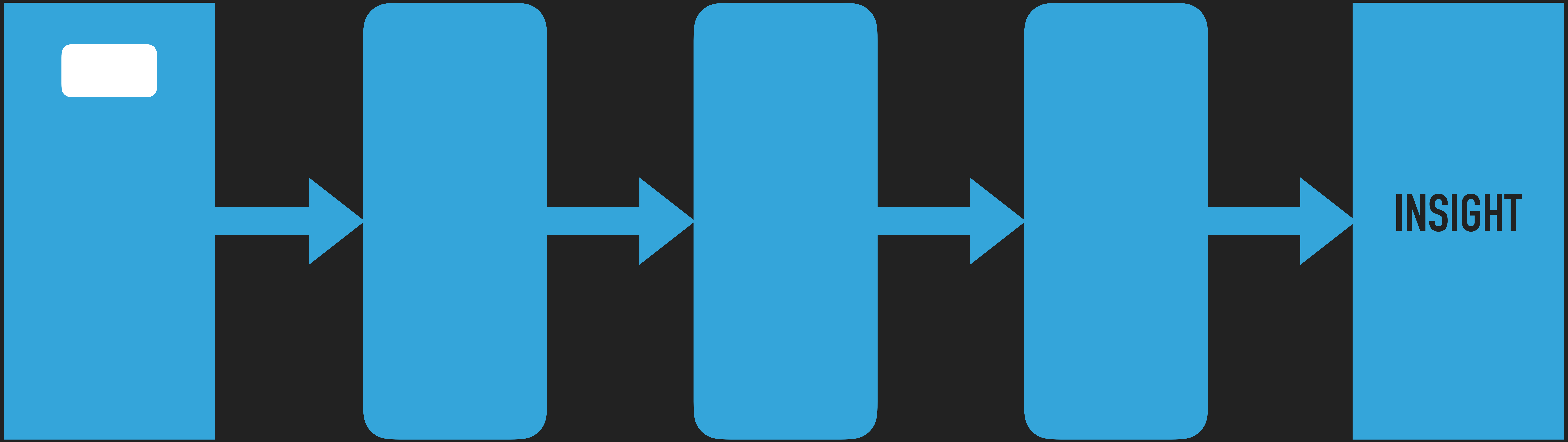
DATAFLOW: BATCH → SPARK



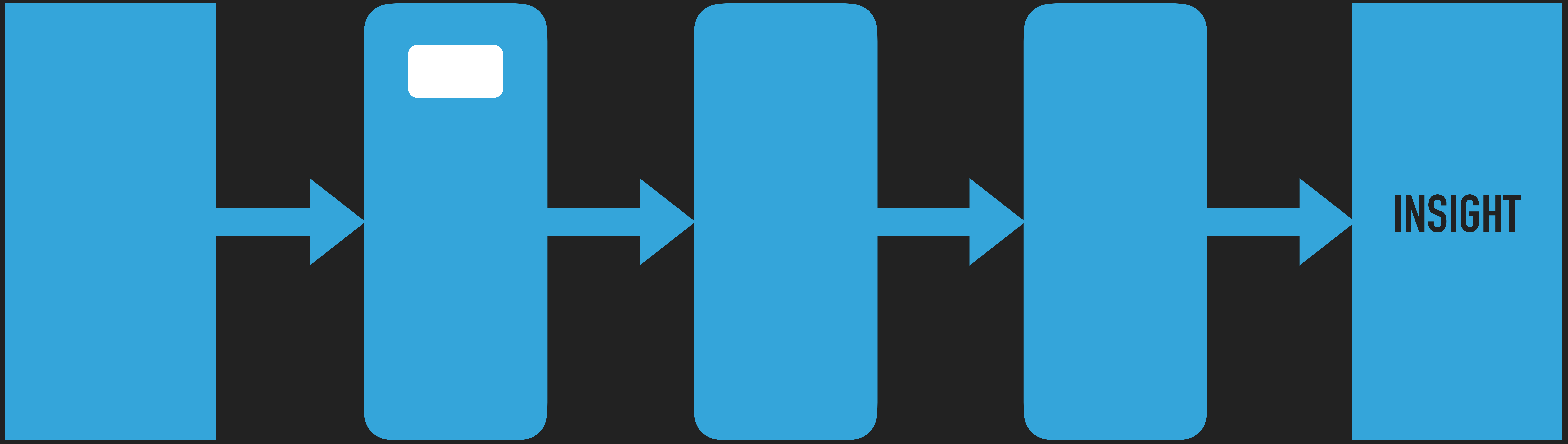
DATAFLOW: STREAMING



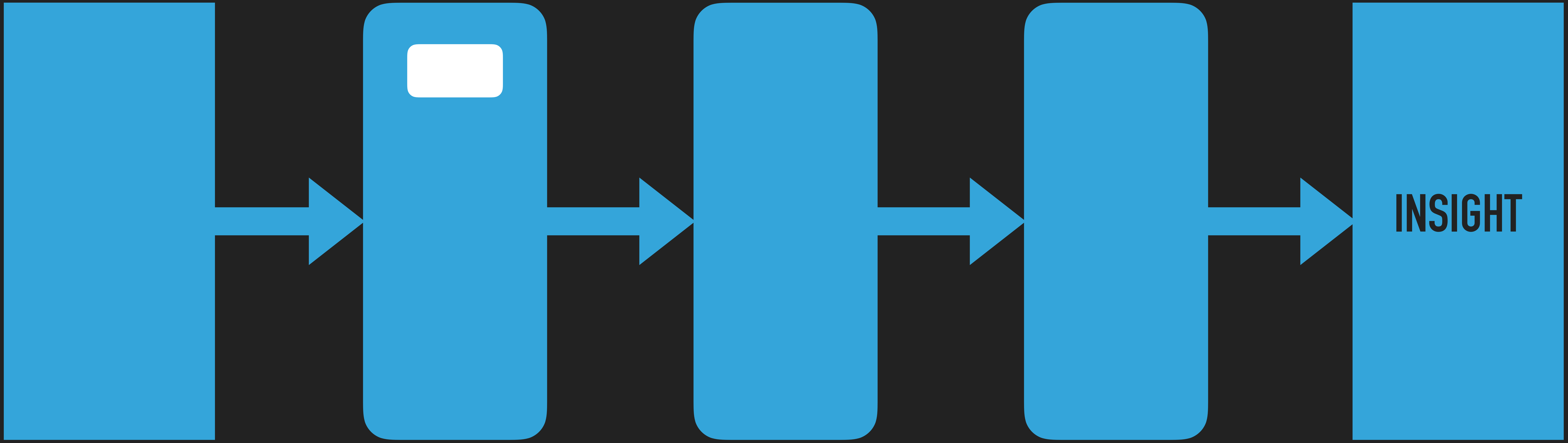
DATAFLOW: STREAMING



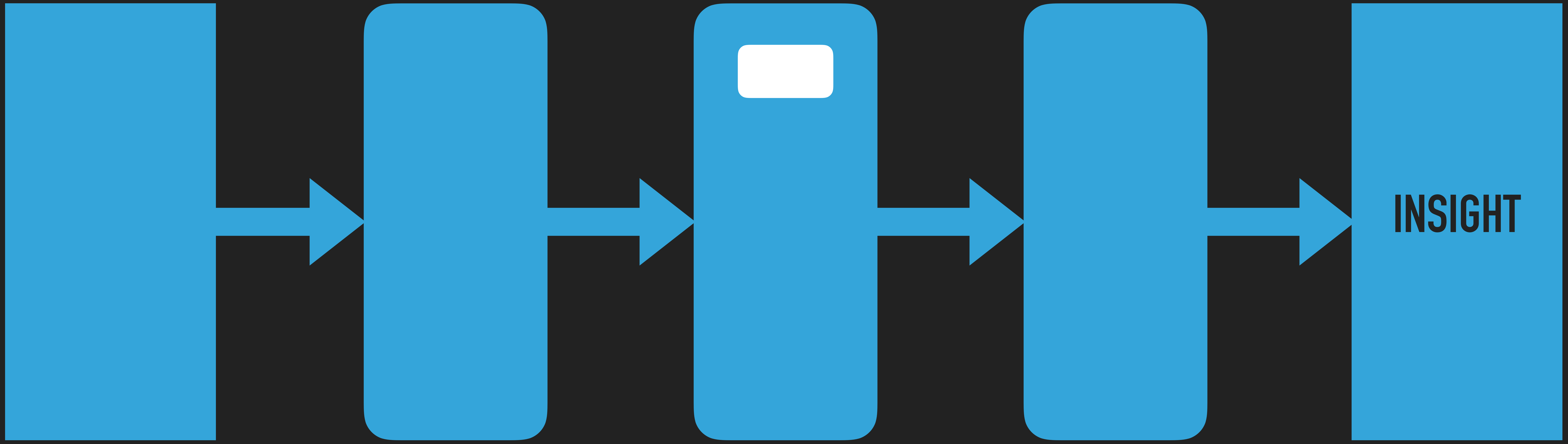
DATAFLOW: STREAMING



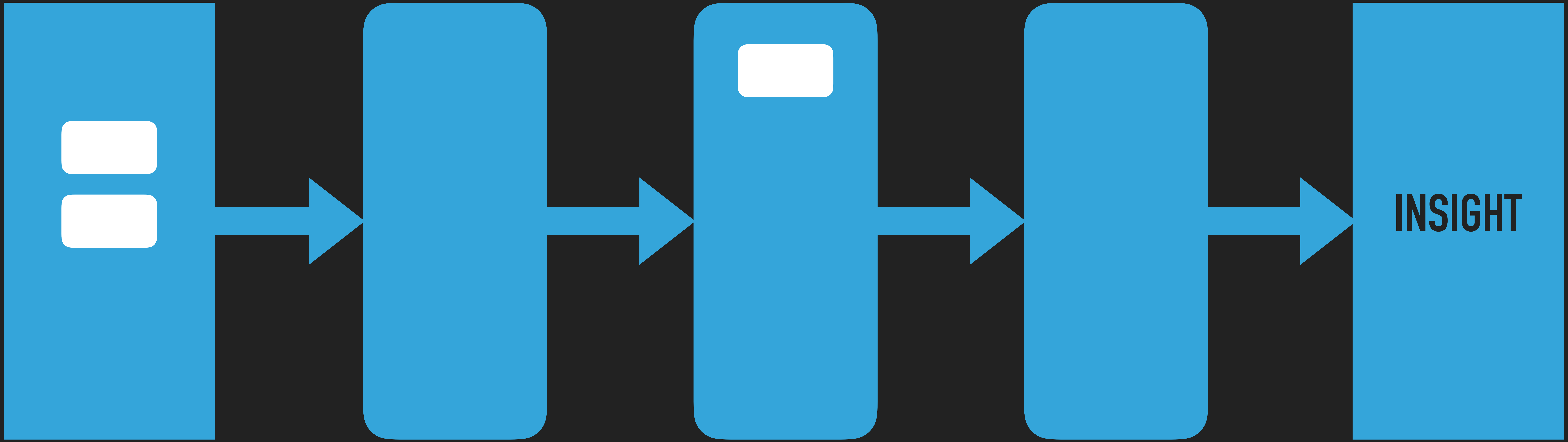
DATAFLOW: STREAMING



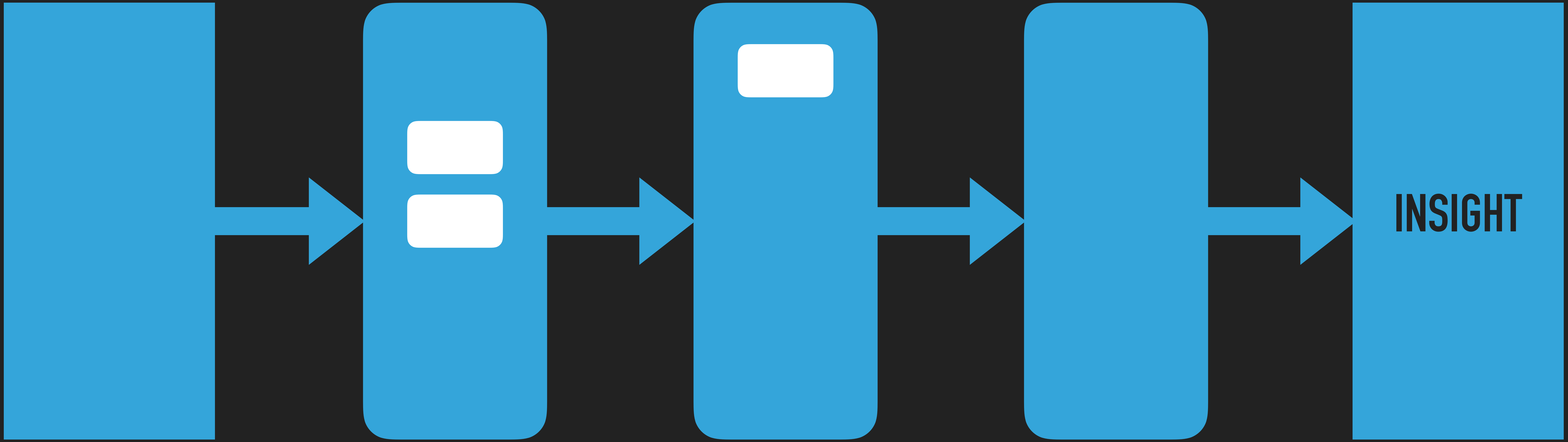
DATAFLOW: STREAMING



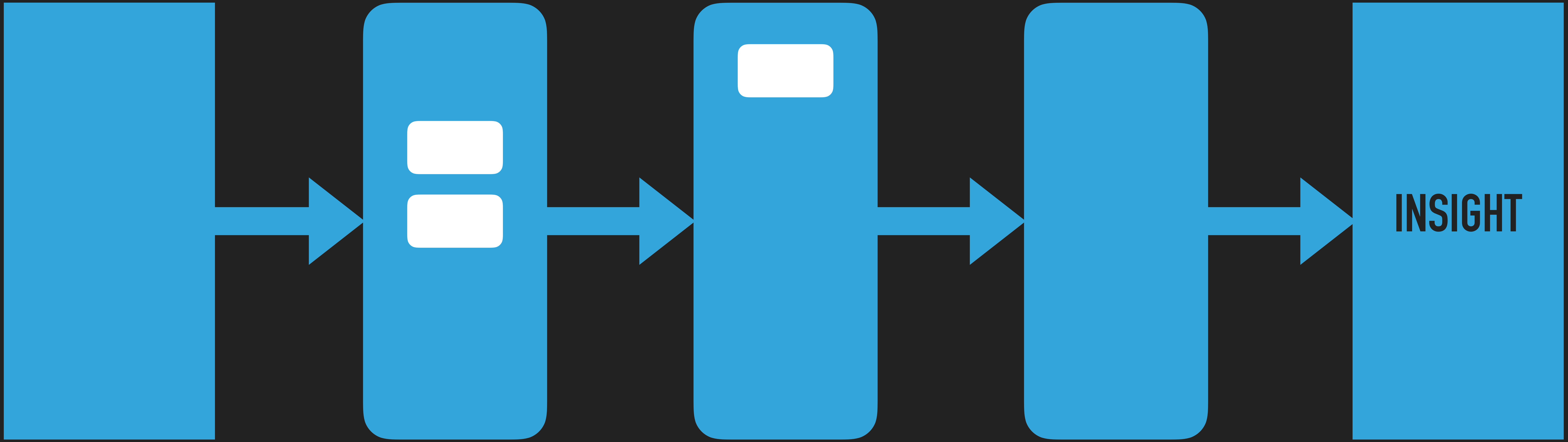
DATAFLOW: STREAMING



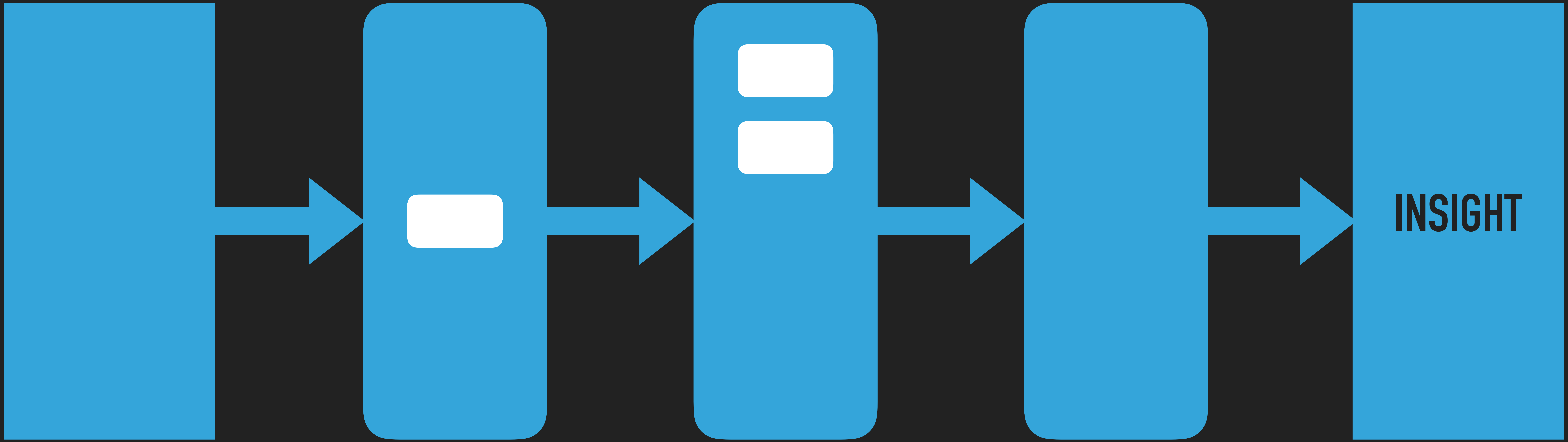
DATAFLOW: STREAMING



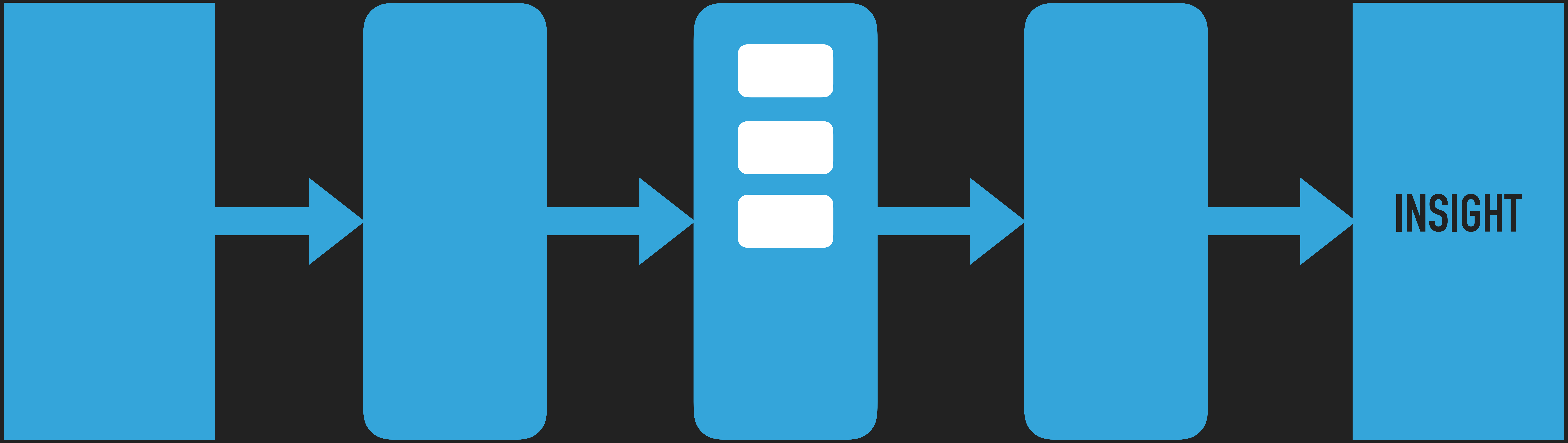
DATAFLOW: STREAMING



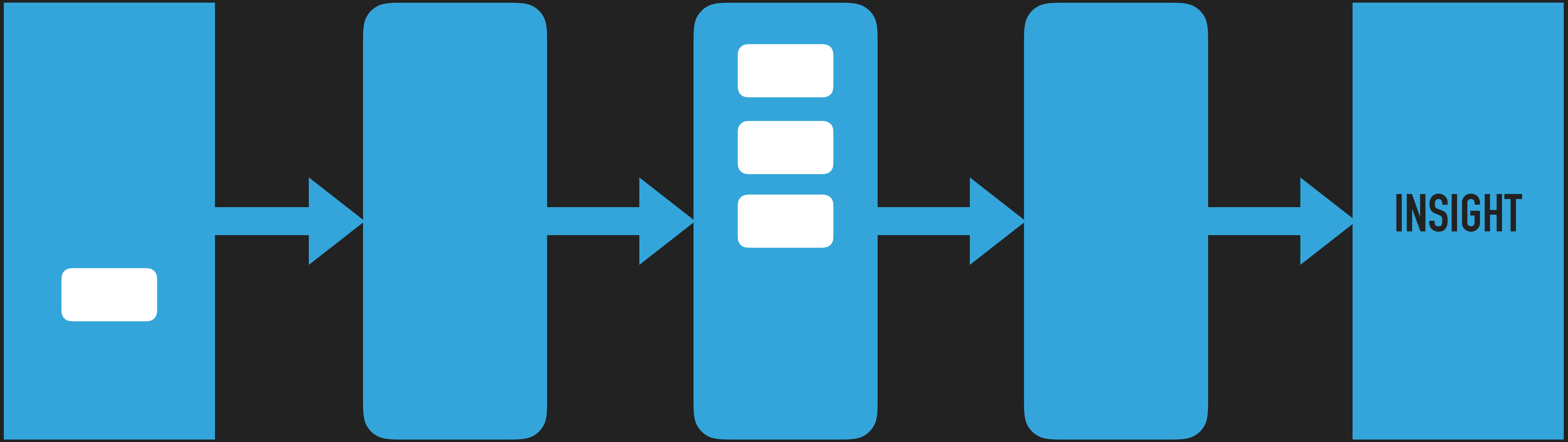
DATAFLOW: STREAMING



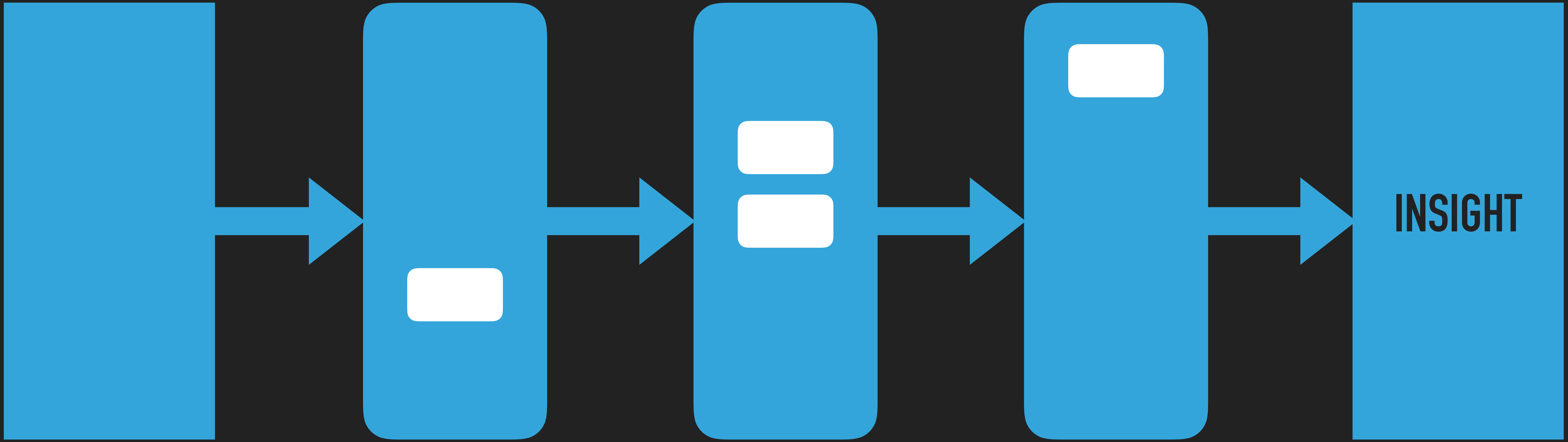
DATAFLOW: STREAMING



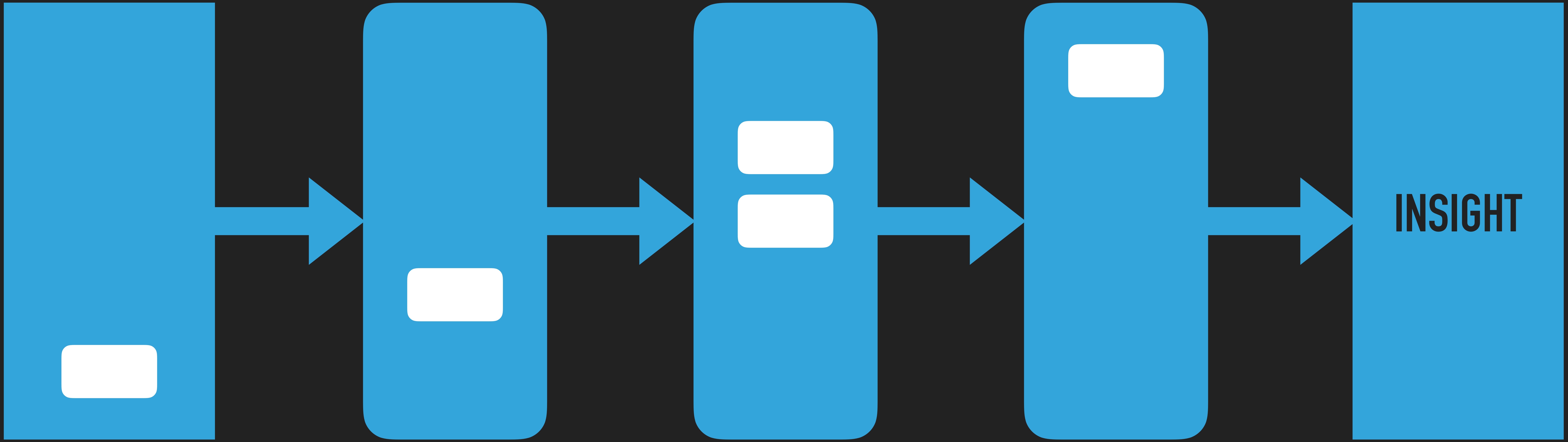
DATAFLOW: STREAMING



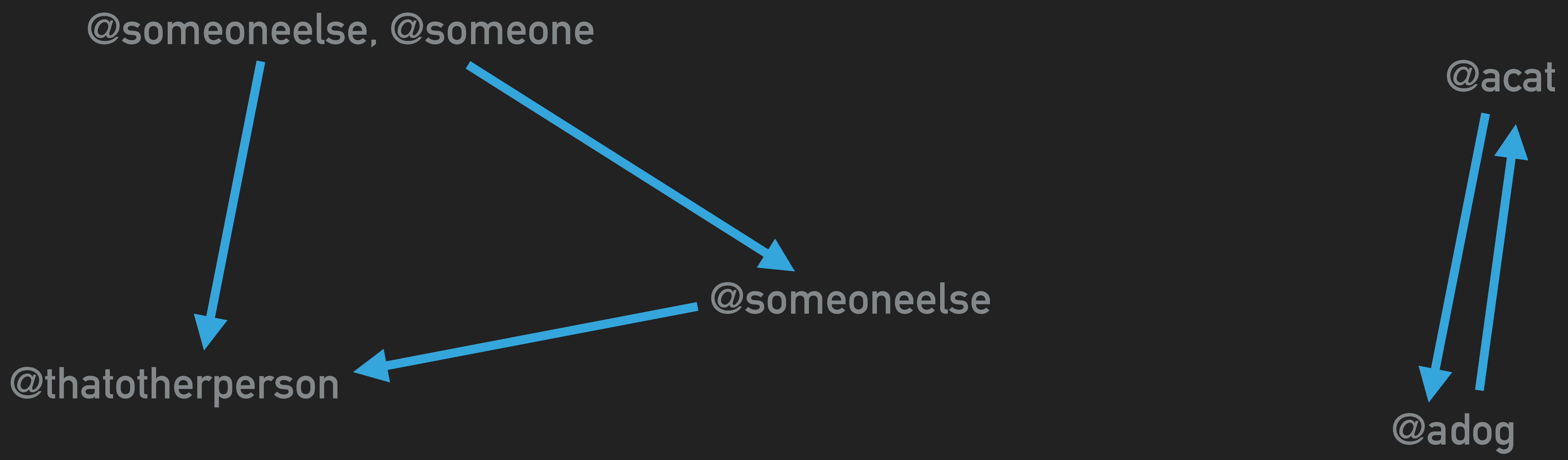
DATAFLOW: STREAMING



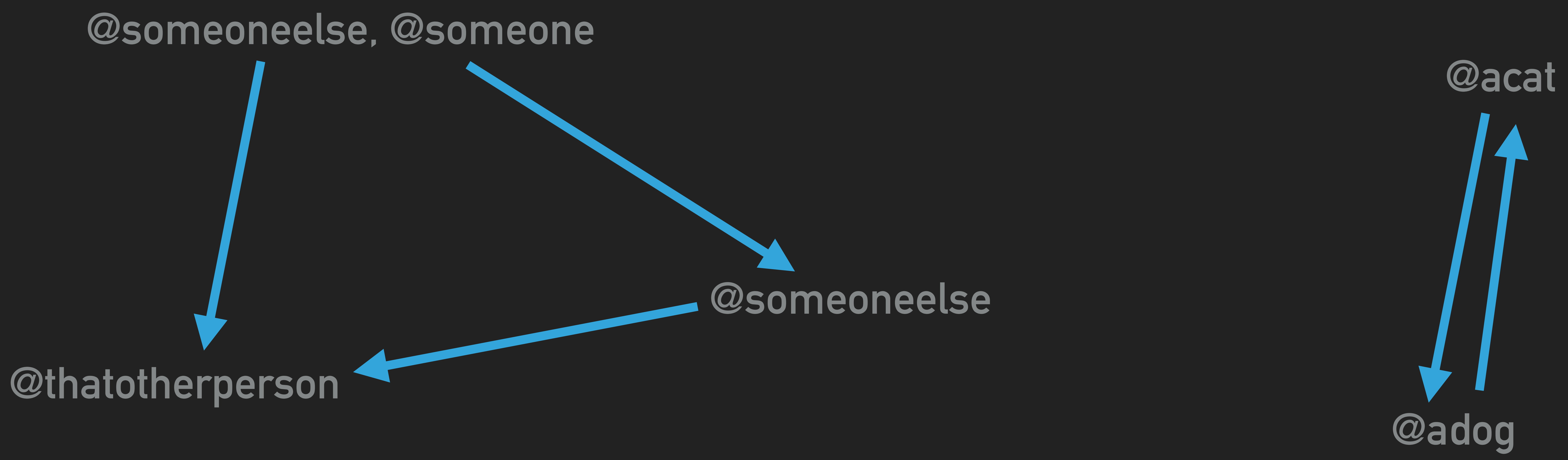
DATAFLOW: STREAMING



GRAPH

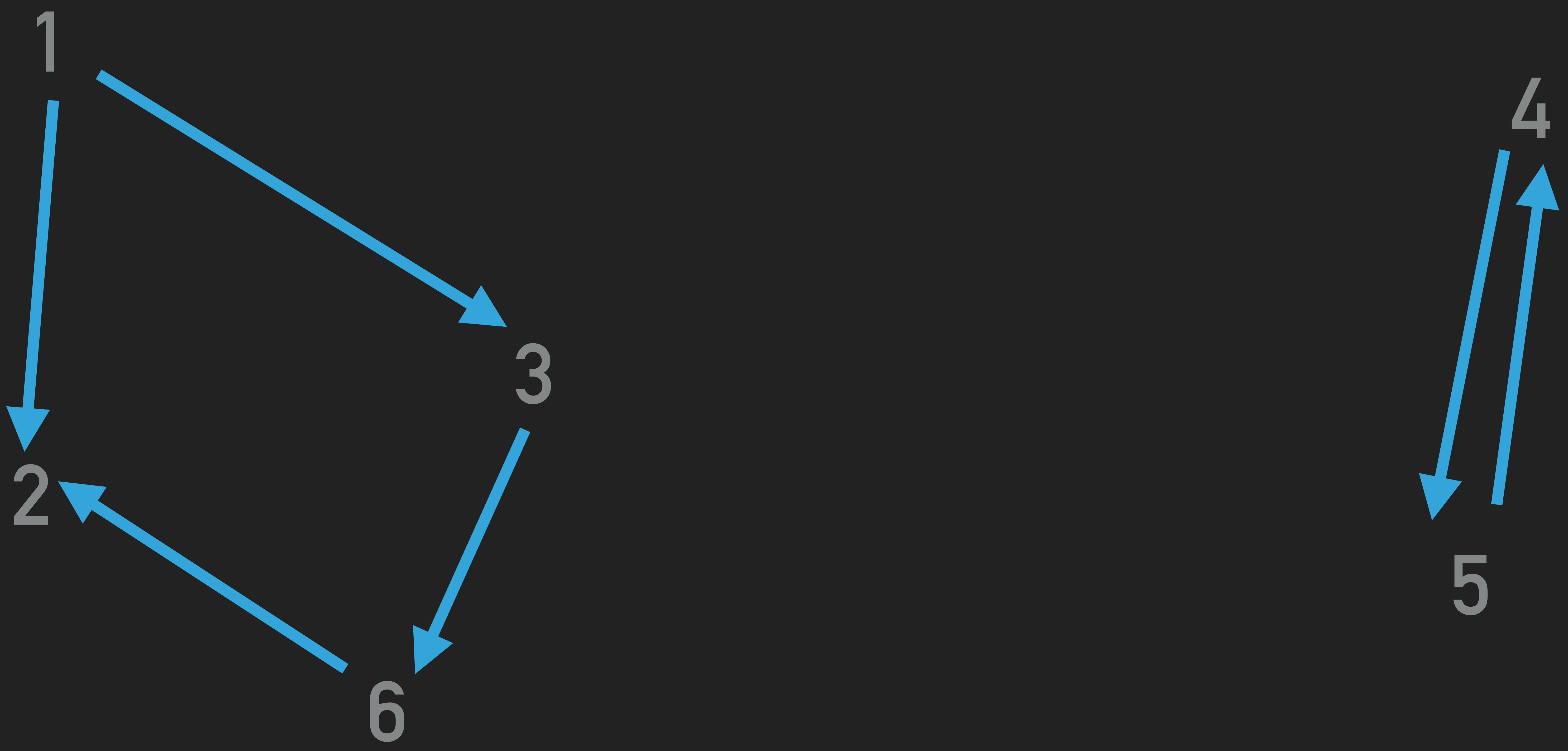

@someoneelse, @someone
@someoneelse
@thatotherperson
@acat
@adog

GRAPH



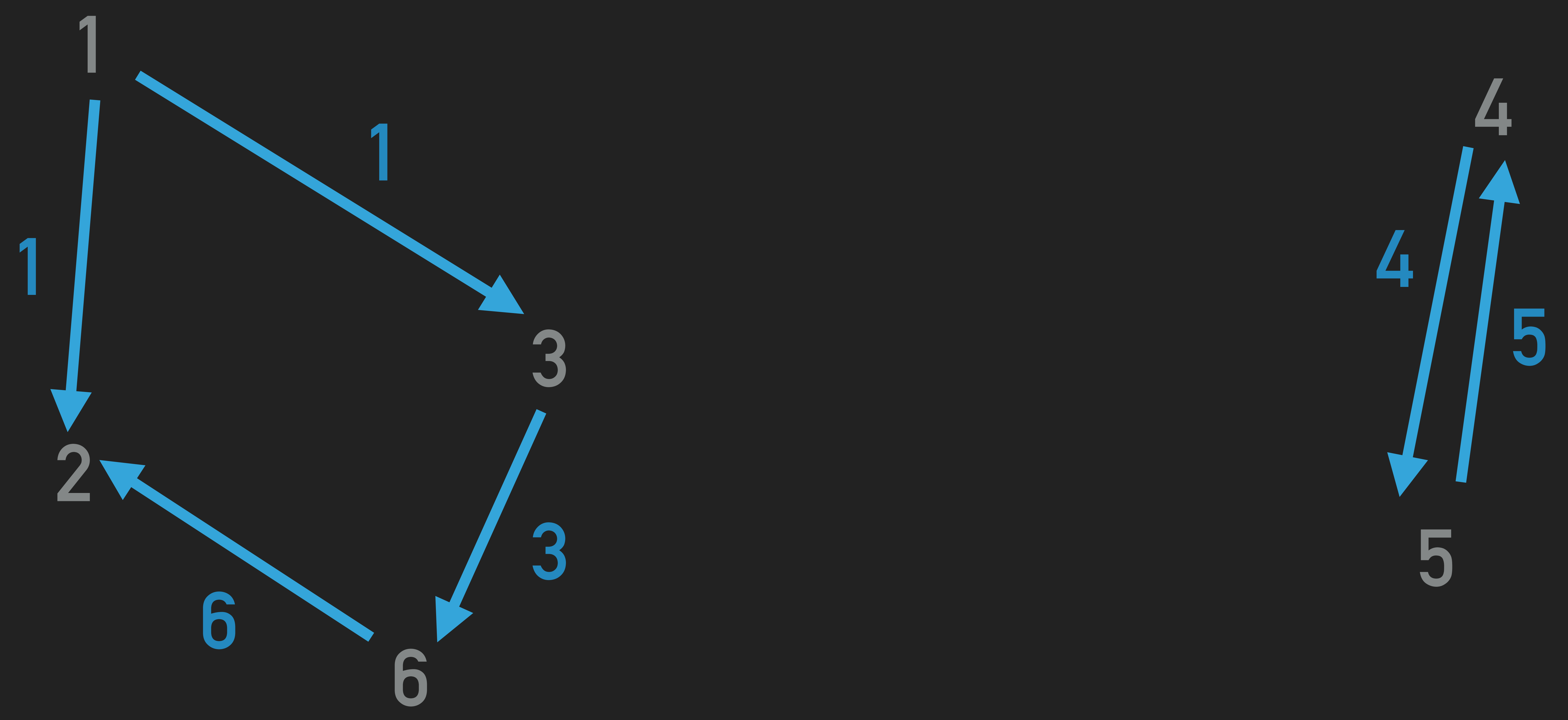
CONNECTED COMPONENTS

GRAPH



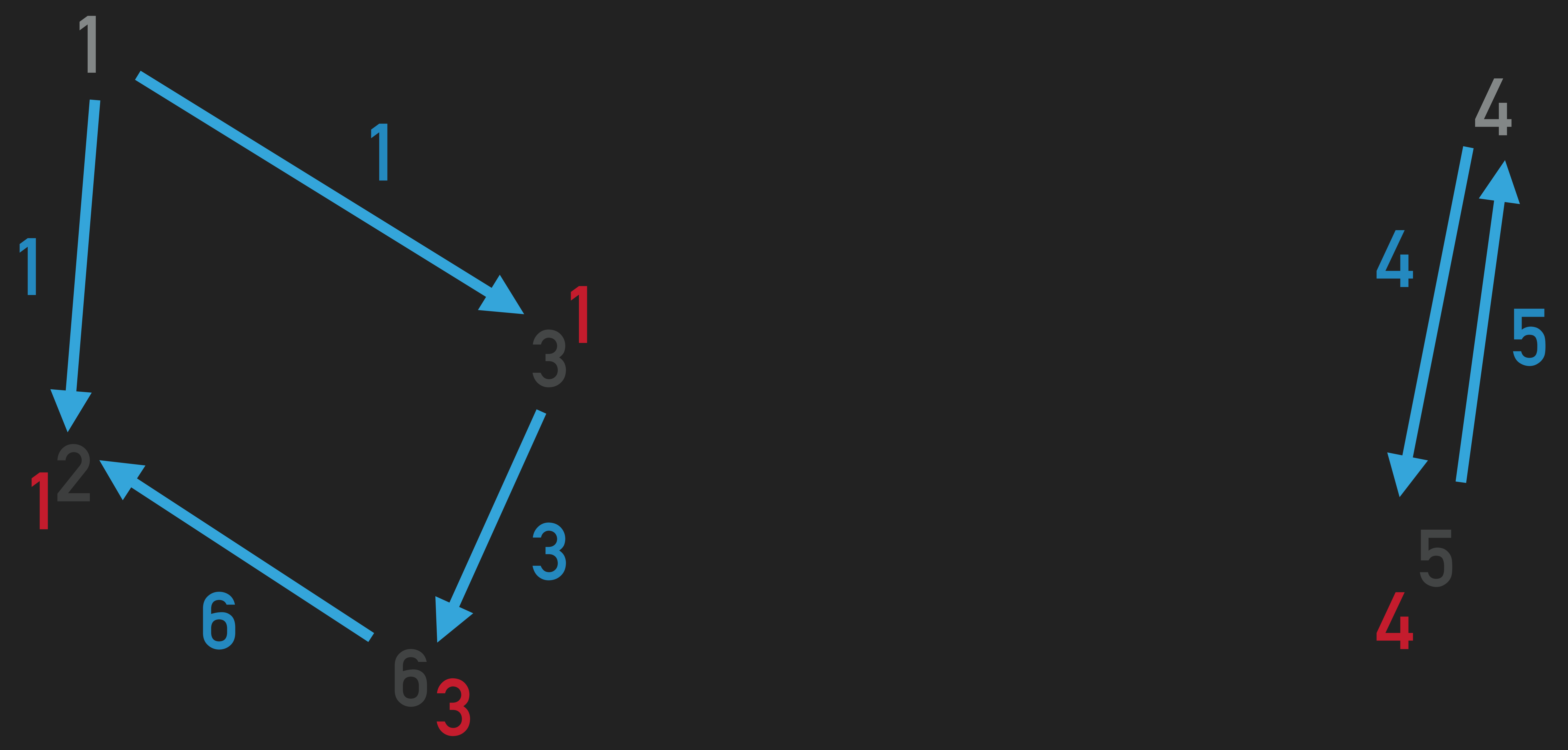
CONNECTED COMPONENTS

GRAPH



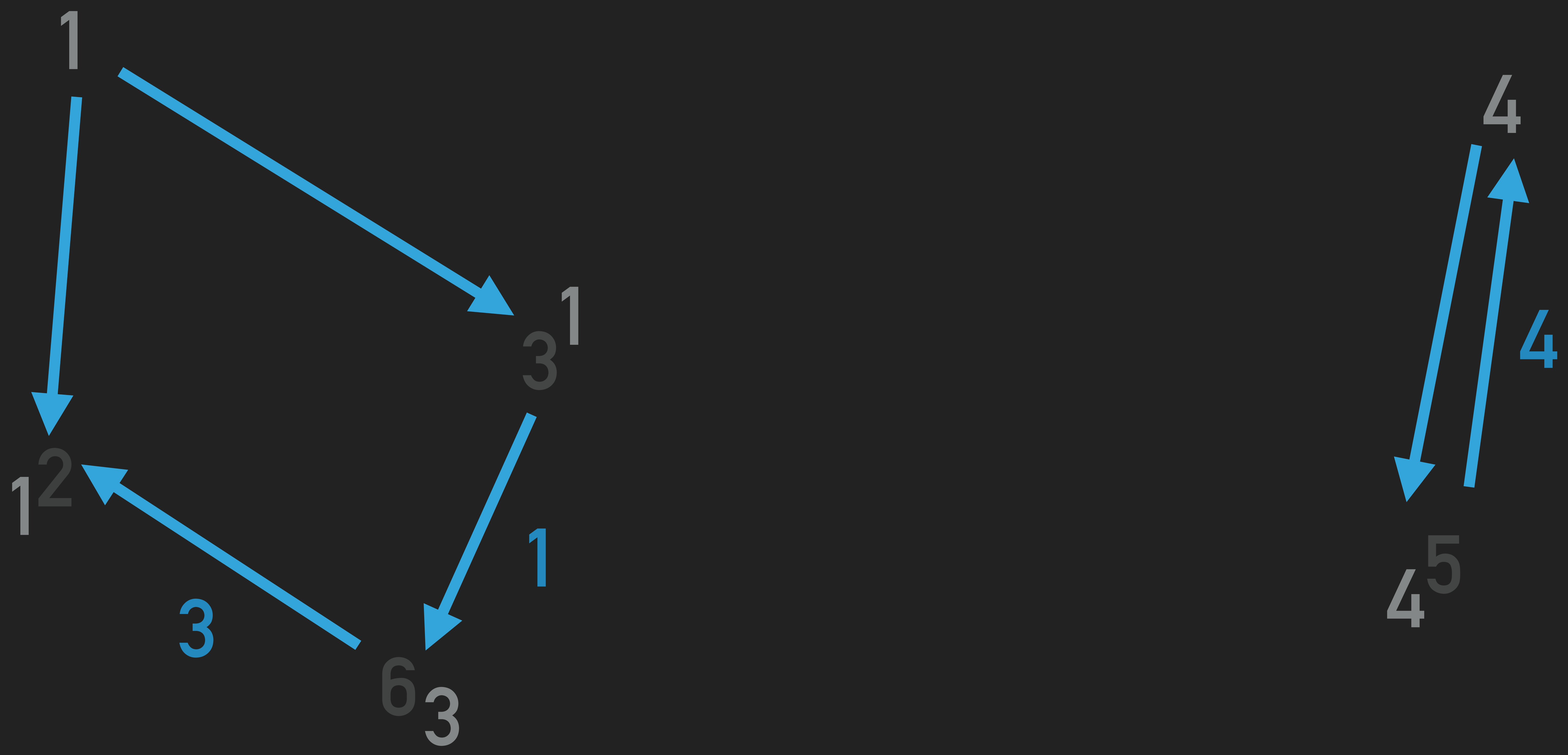
CONNECTED COMPONENTS

GRAPH



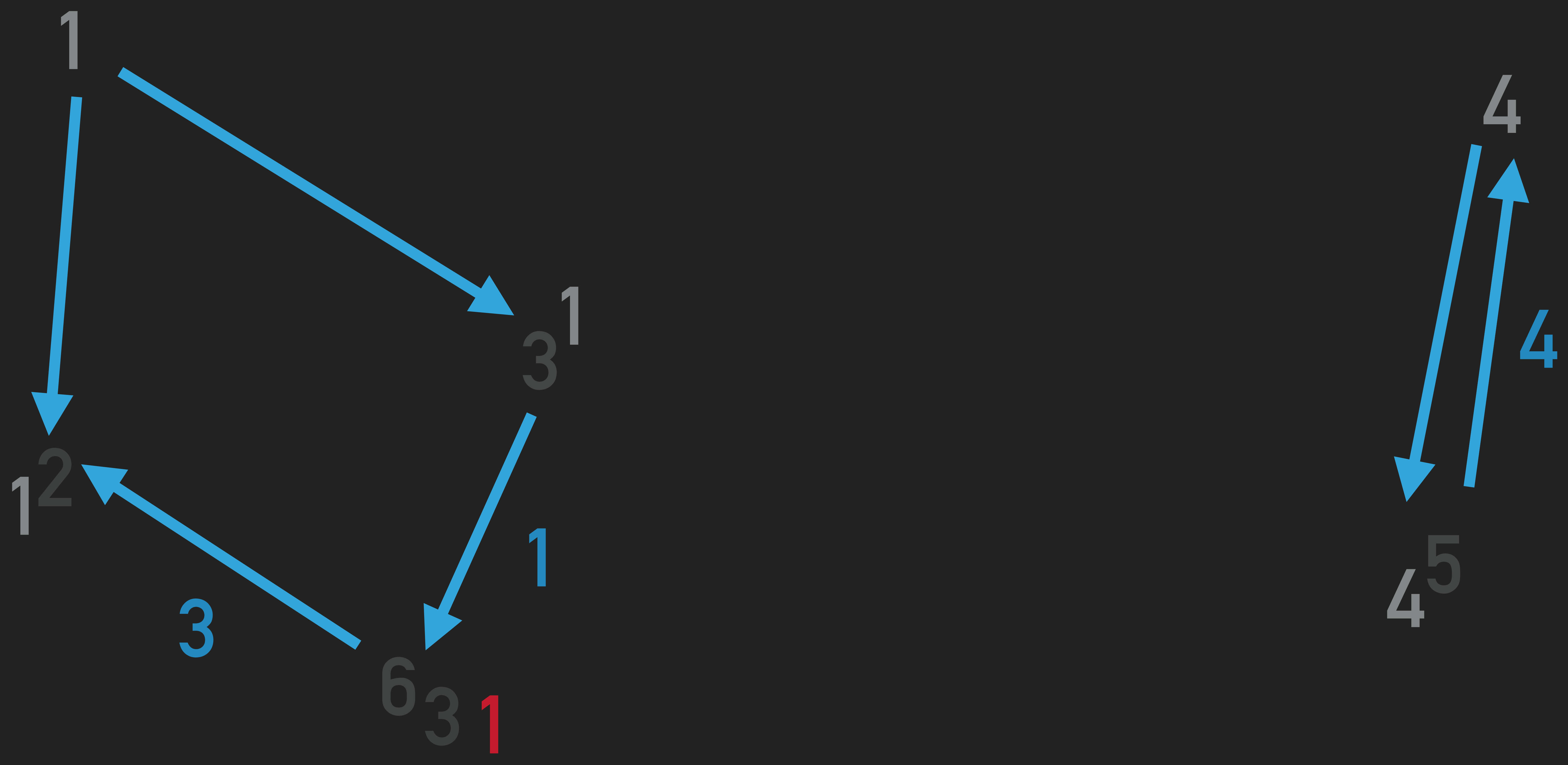
CONNECTED COMPONENTS

GRAPH



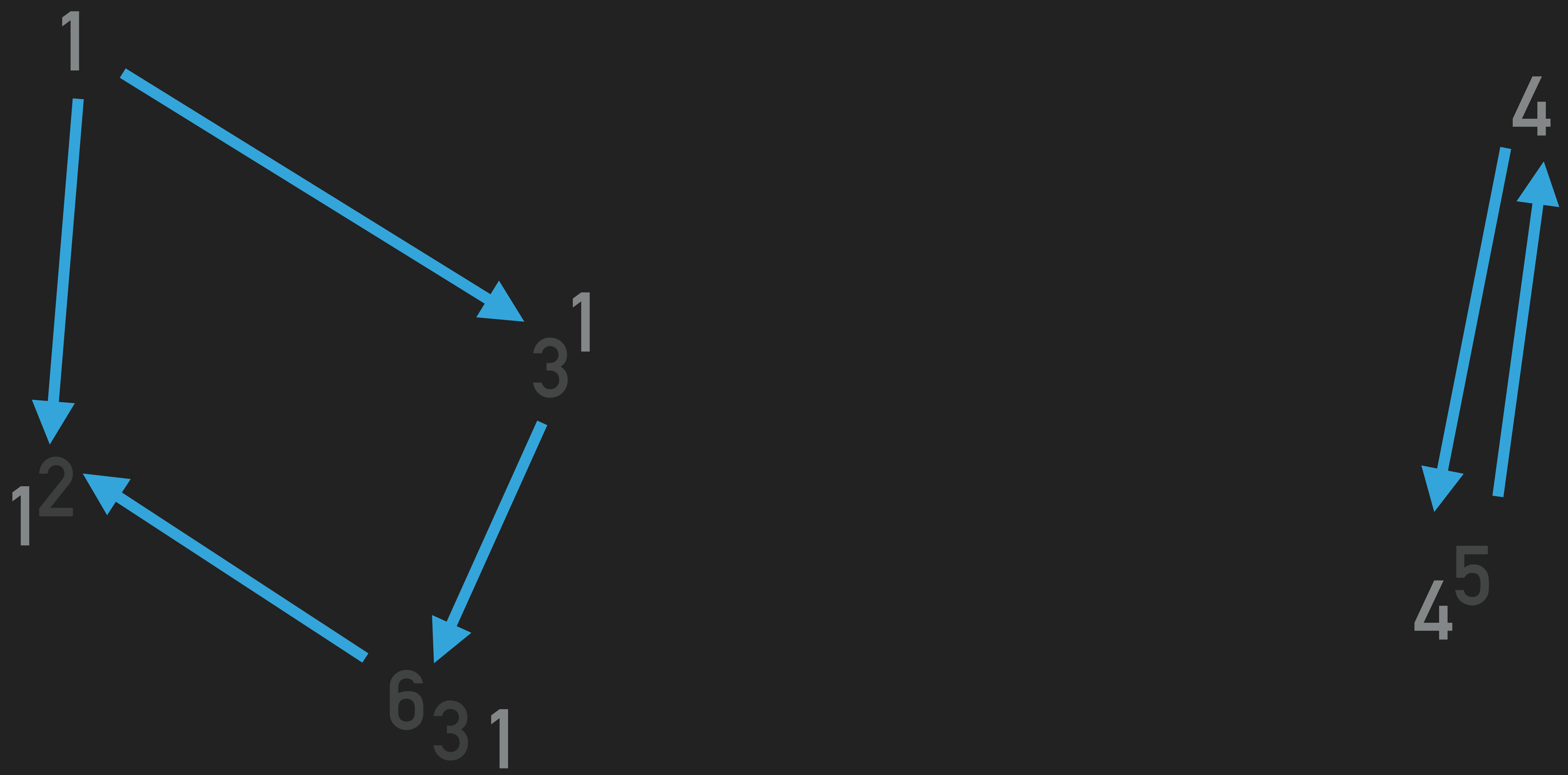
CONNECTED COMPONENTS

GRAPH



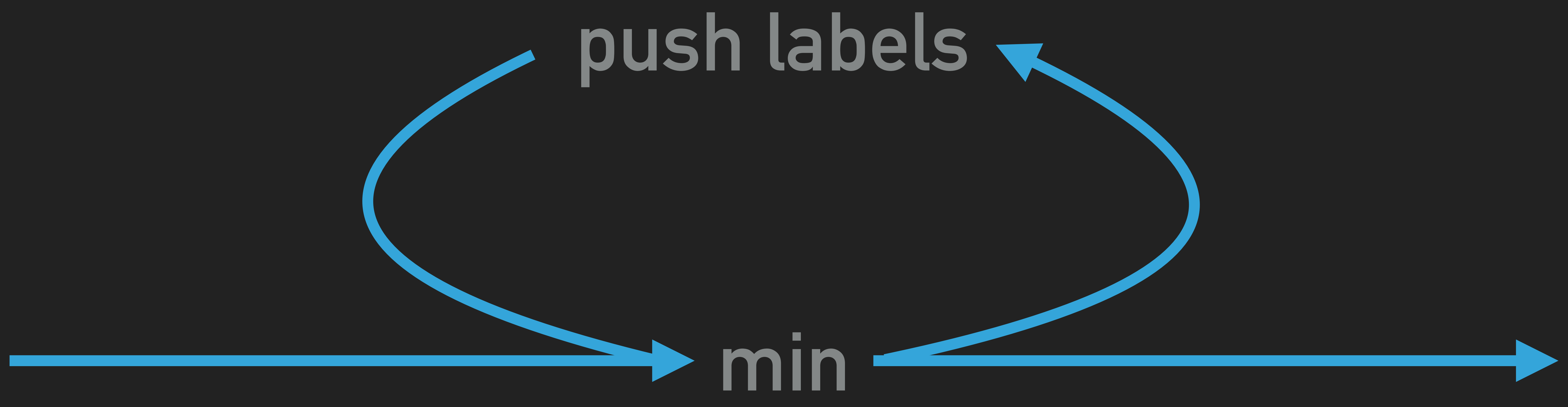
CONNECTED COMPONENTS

GRAPH



CONNECTED COMPONENTS

GRAPH PROCESSING IN DATAFLOW: CYCLES!



push labels

min

CONTEXT: PROGRAMMING MODEL

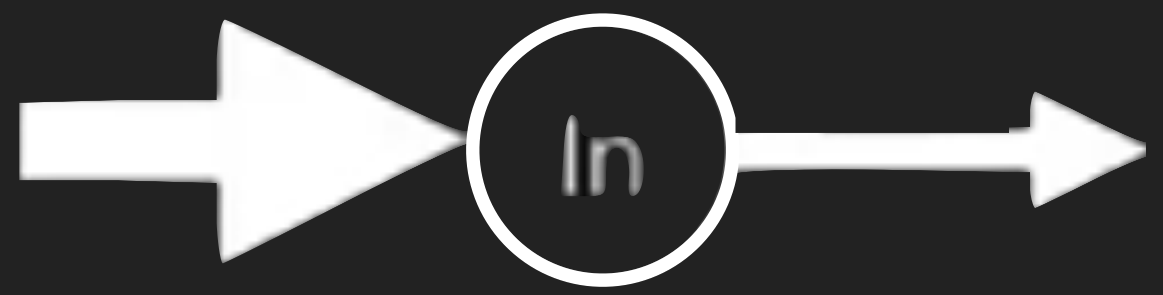
STREAMING

BATCH

GRAPH

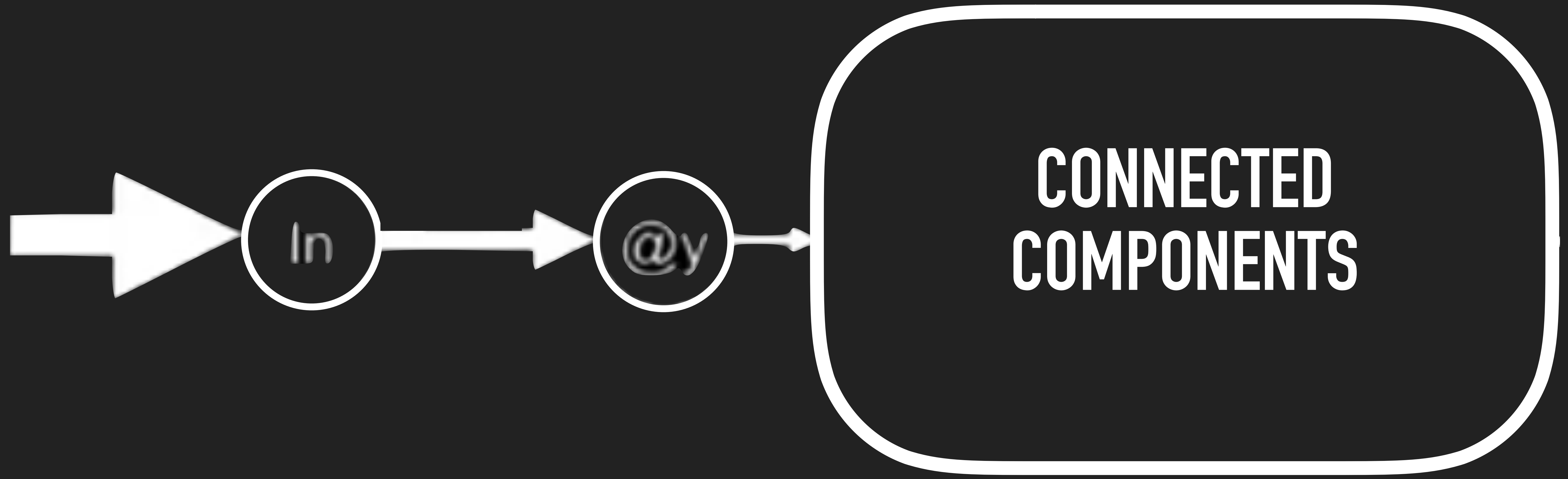
CONTEXT: PROGRAMMING MODEL

STREAMING BATCH GRAPH



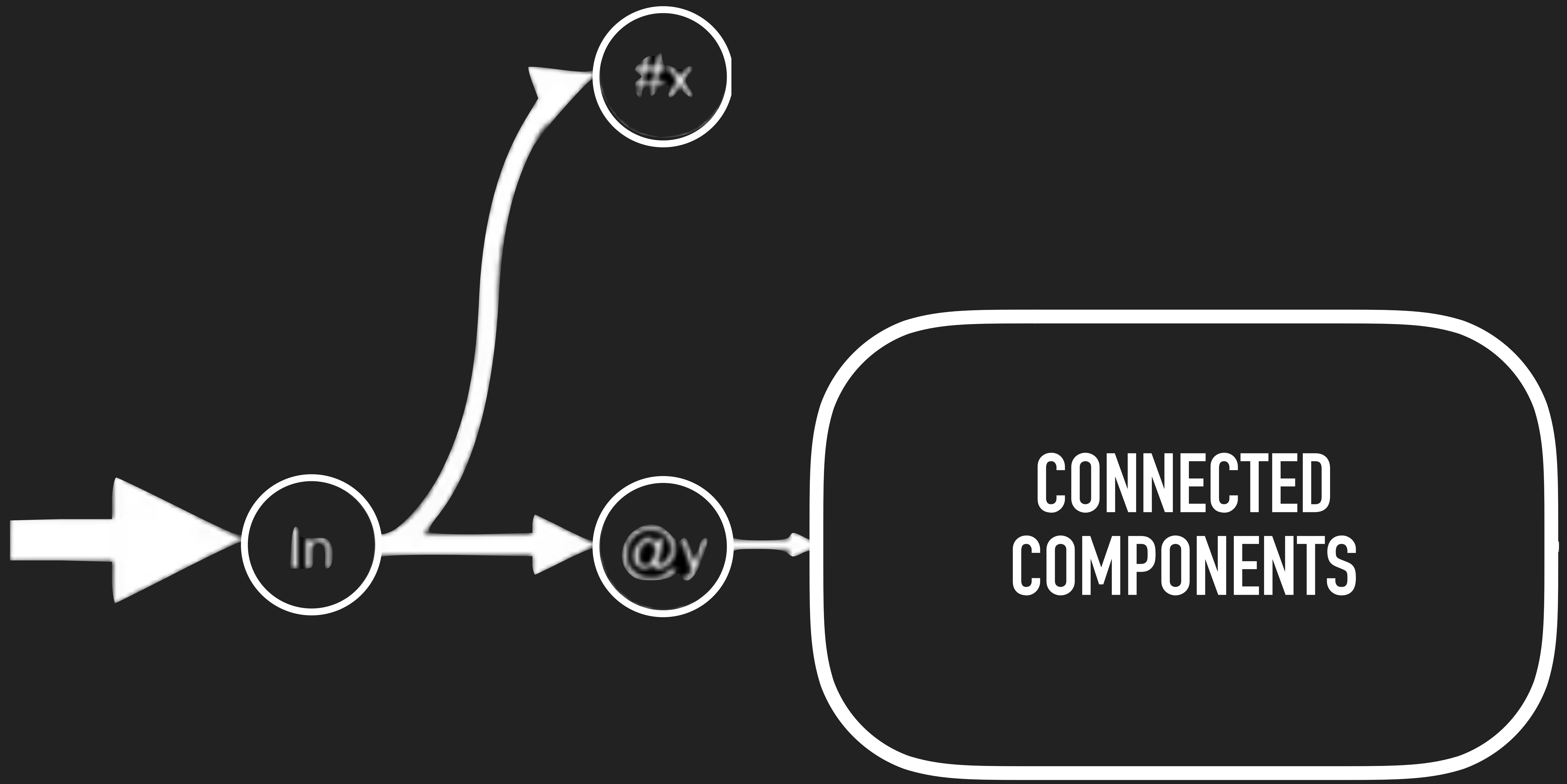
CONTEXT: PROGRAMMING MODEL

STREAMING	BATCH	GRAPH

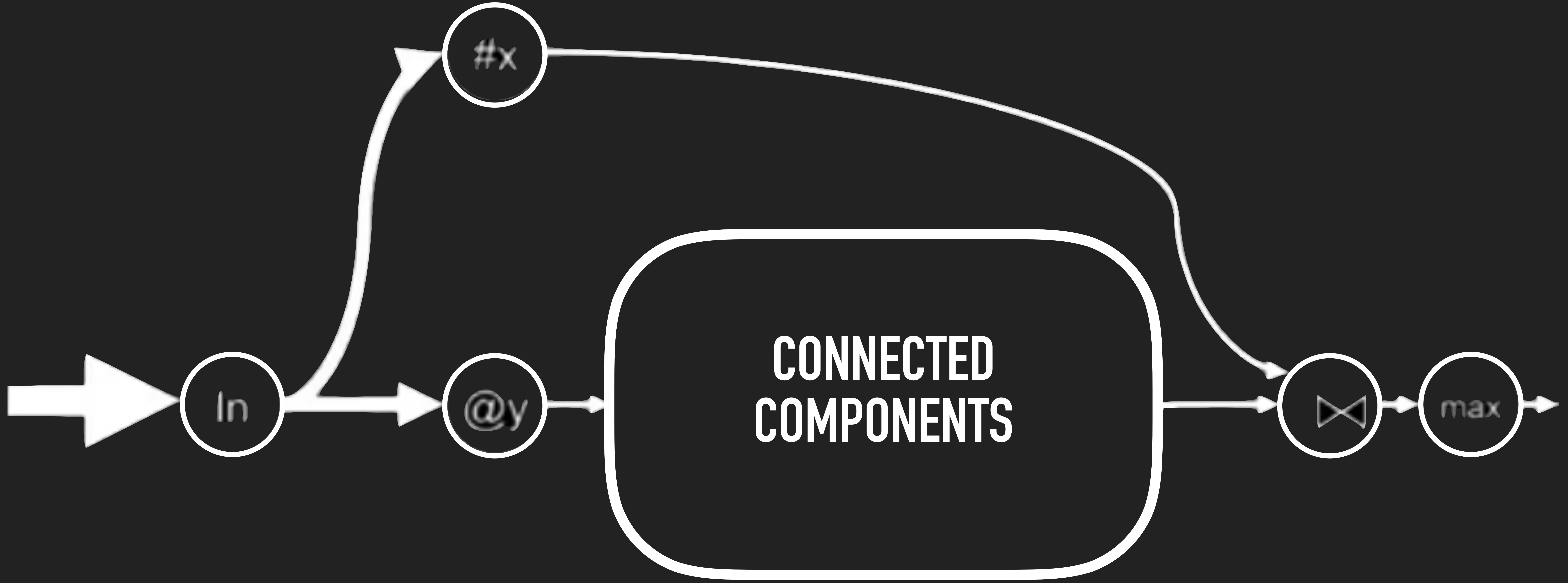


CONTEXT: PROGRAMMING MODEL

STREAMING BATCH GRAPH

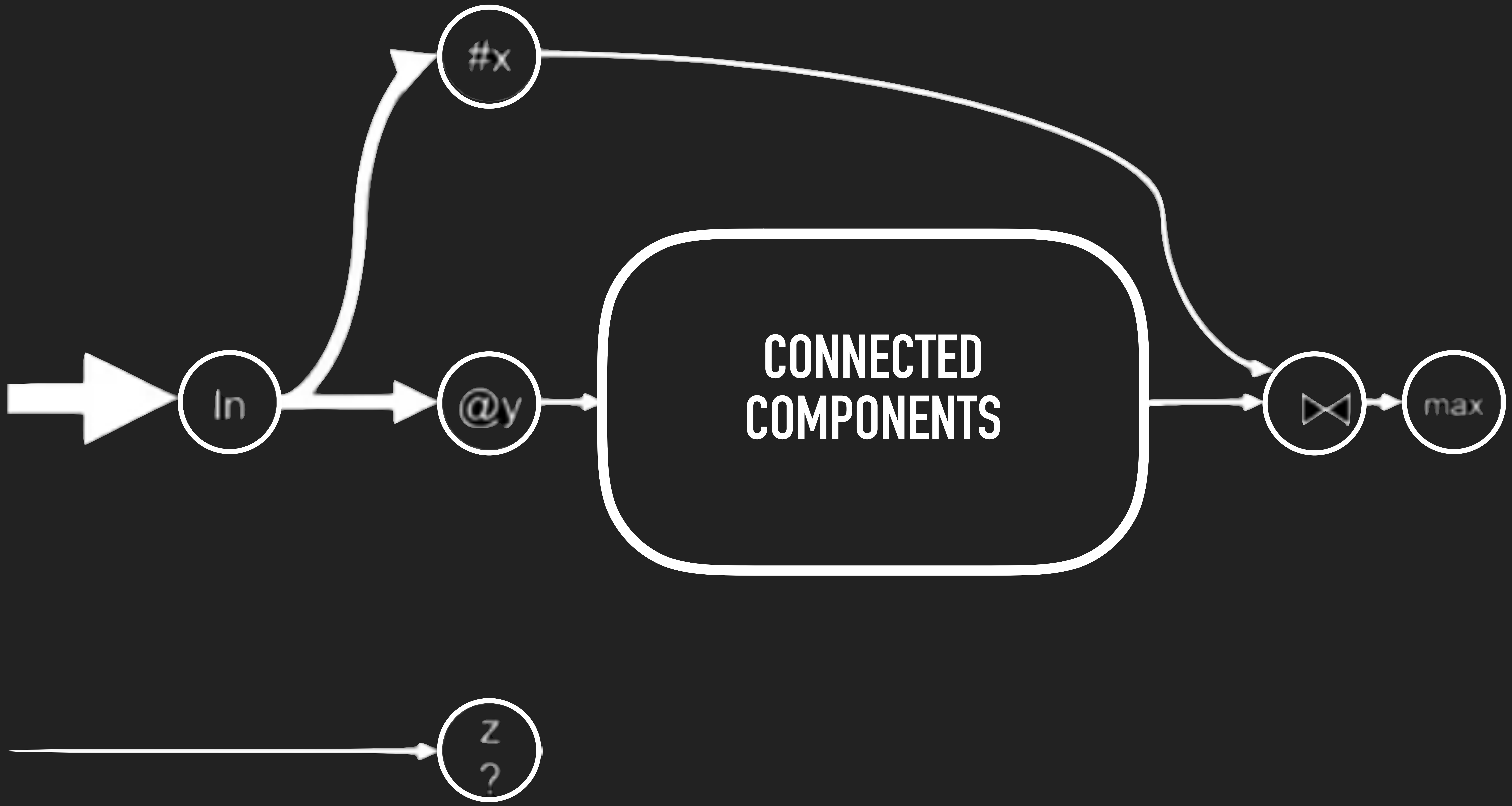


STREAMING BATCH GRAPH

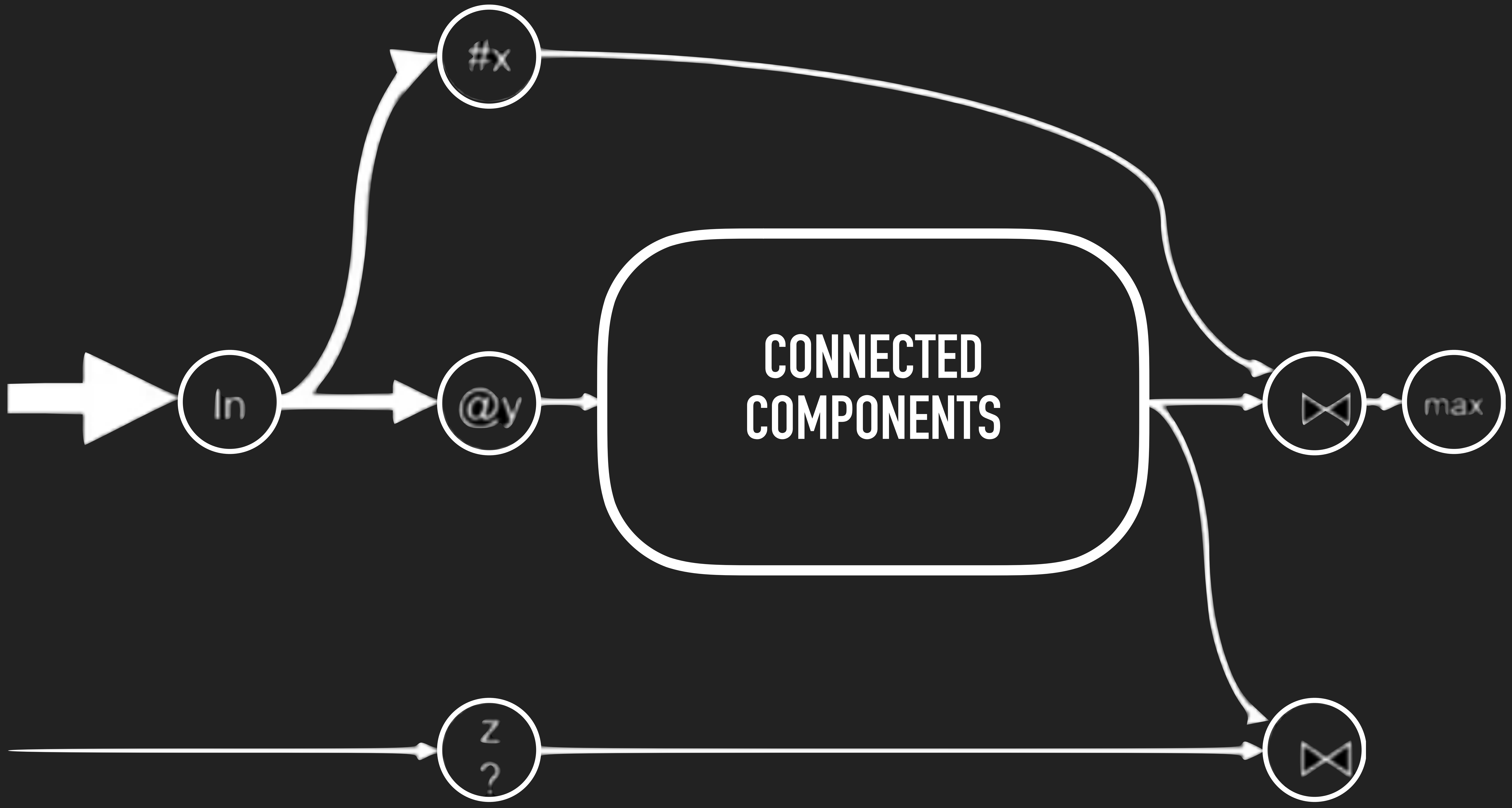


CONTEXT: PROGRAMMING MODEL

STREAMING BATCH GRAPH

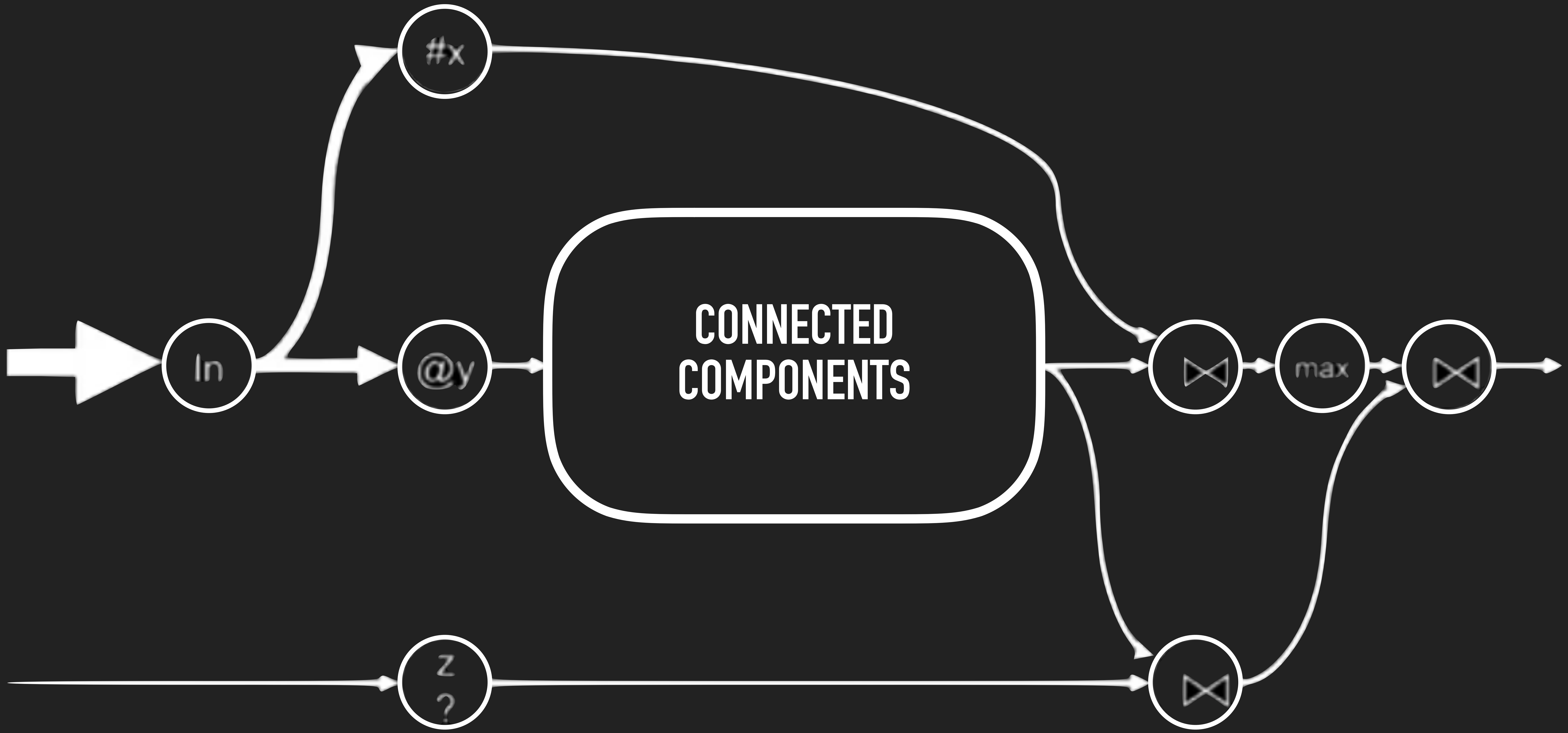


STREAMING BATCH GRAPH



CONTEXT: PROGRAMMING MODEL

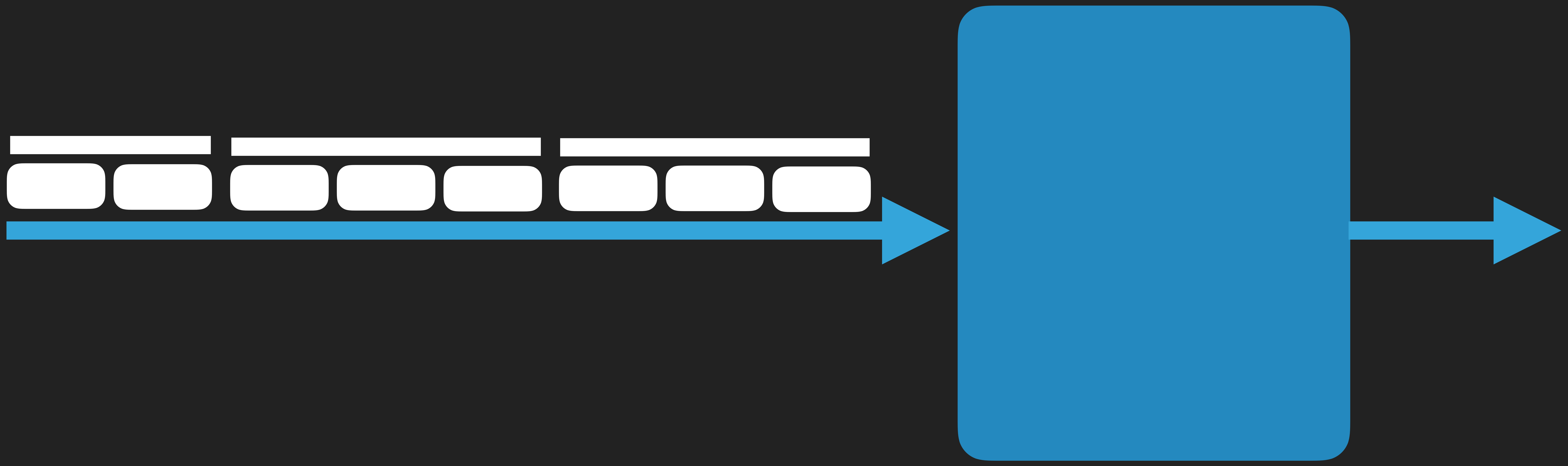
STREAMING BATCH GRAPH



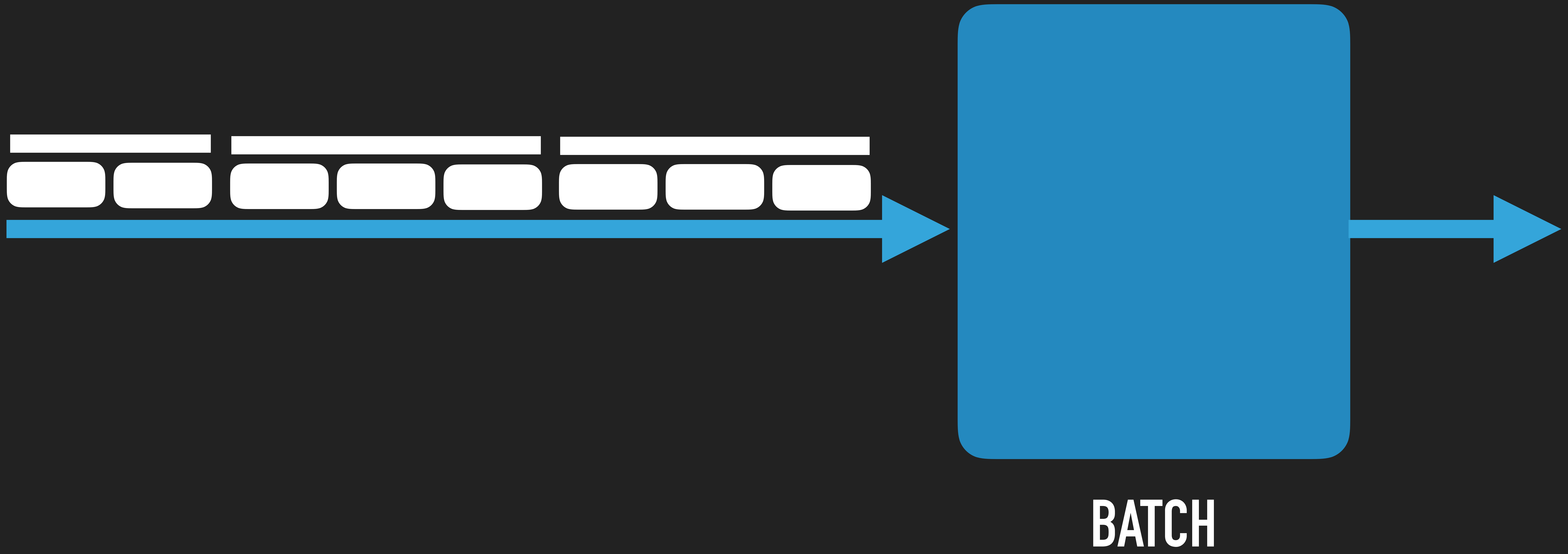
> ACHIEVE THE PERFORMANCE OF SPECIALIZED FRAMEWORK, PROVIDE THE FLEXIBILITY OF A GENERIC FRAMEWORK.

Derek Murray

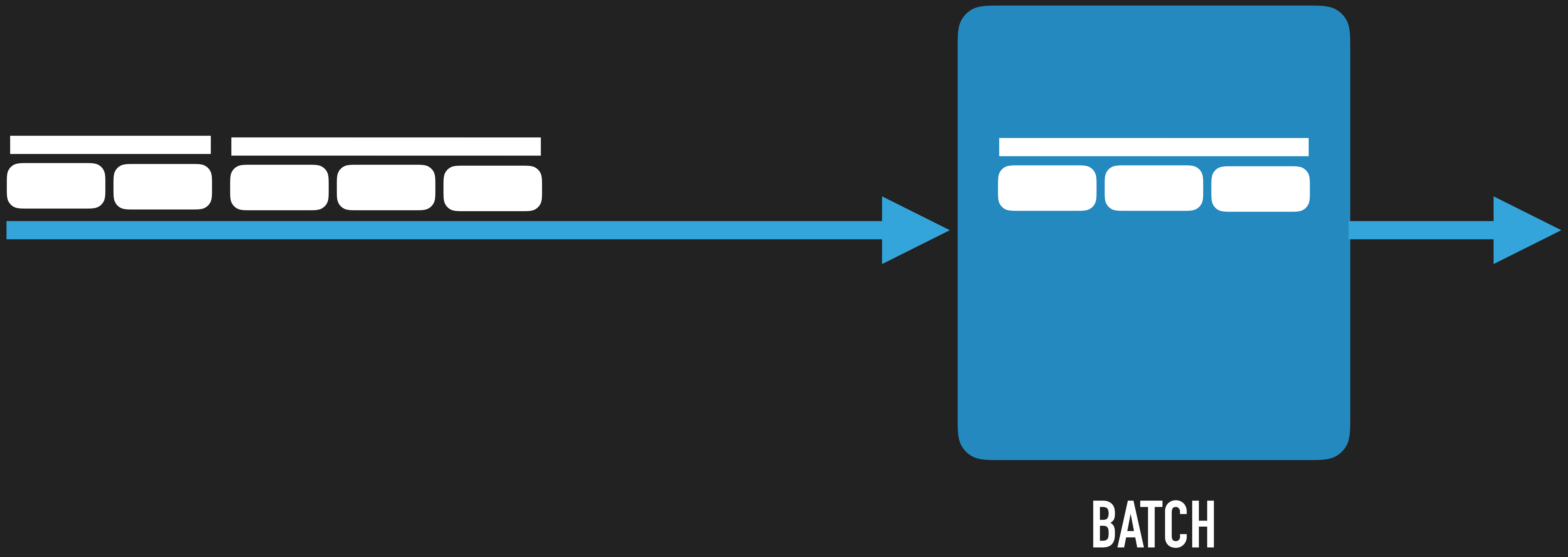
AGGREGATION IN ASYNCHRONOUS MODEL



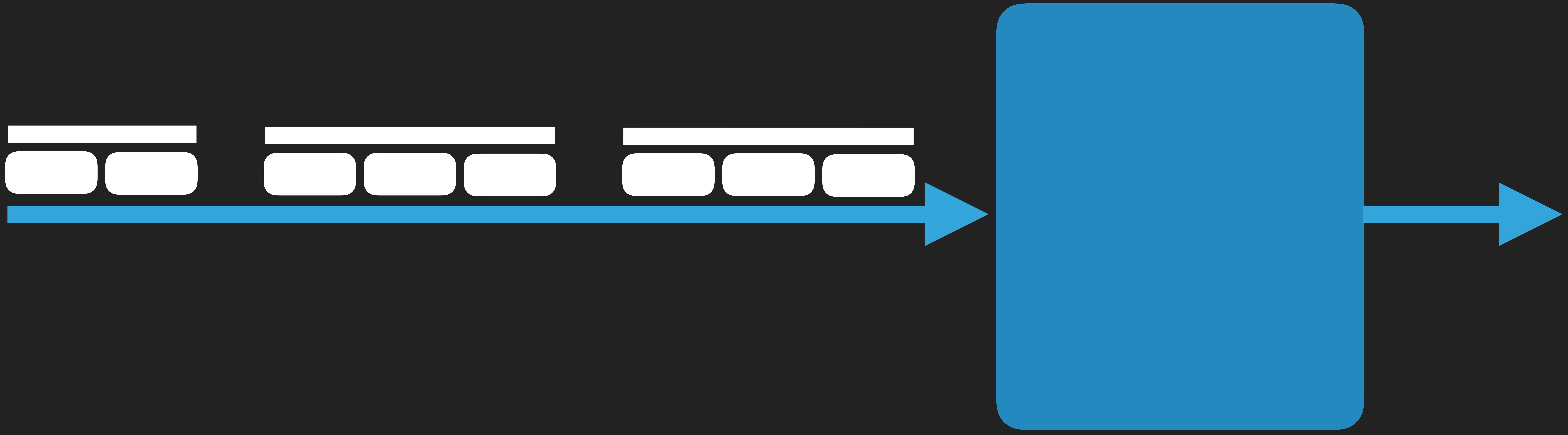
AGGREGATION IN ASYNCHRONOUS MODEL



AGGREGATION IN ASYNCHRONOUS MODEL

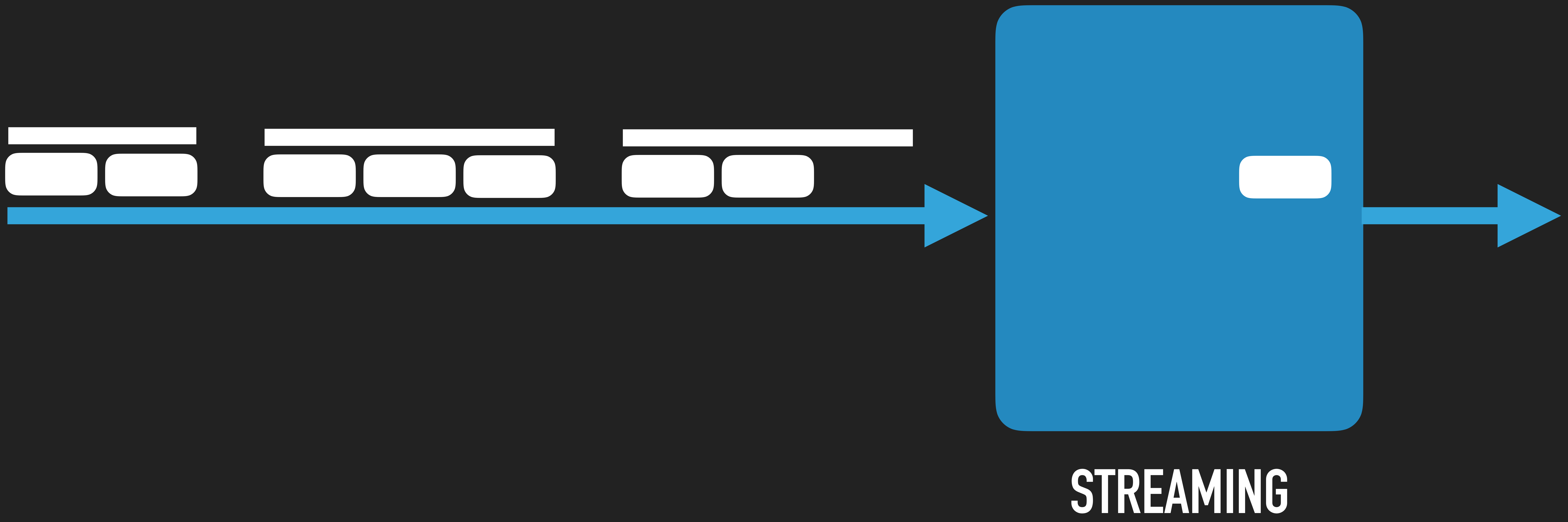


AGGREGATION IN ASYNCHRONOUS MODEL

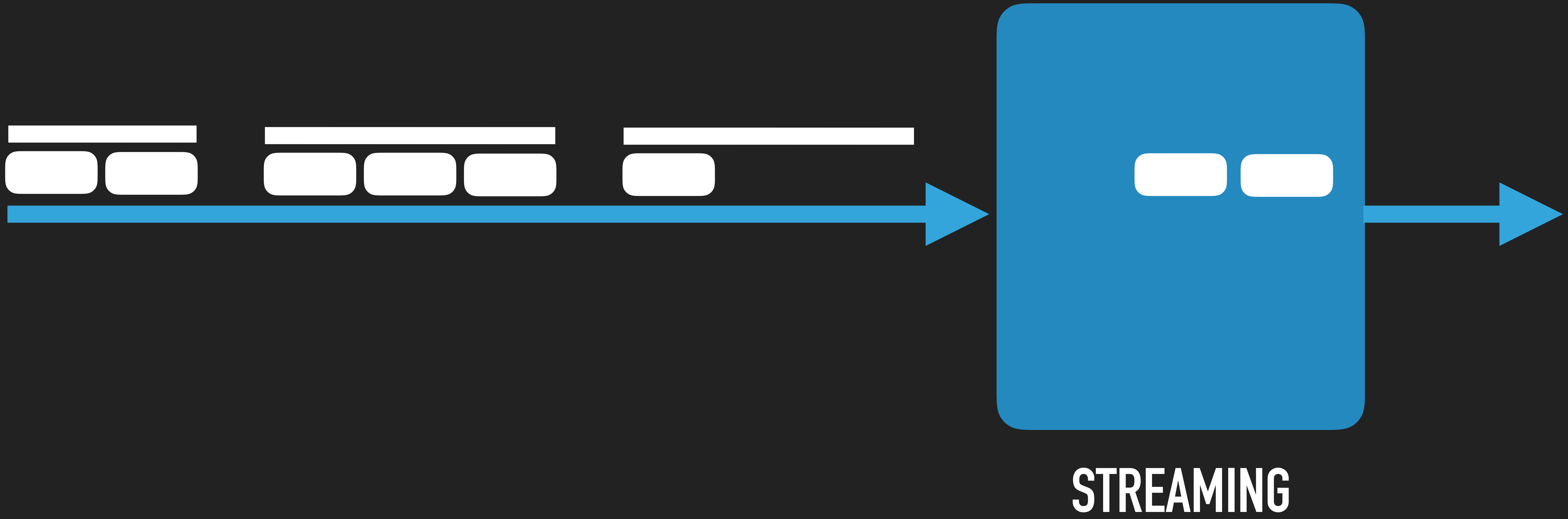


STREAMING

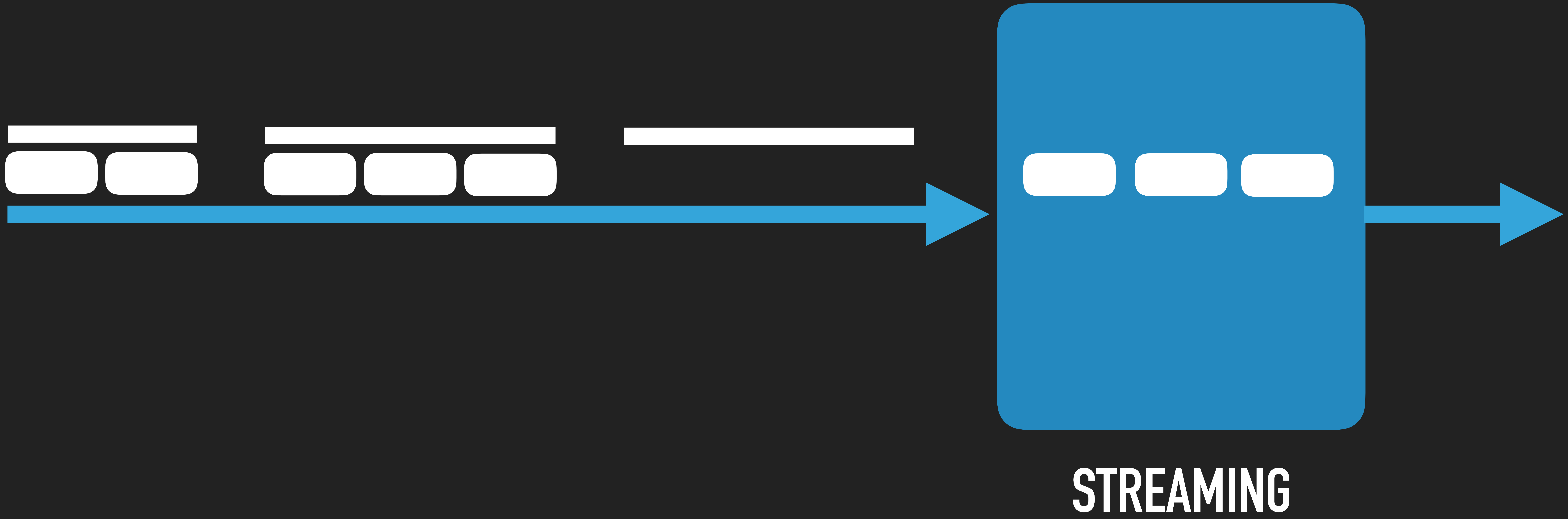
AGGREGATION IN ASYNCHRONOUS MODEL



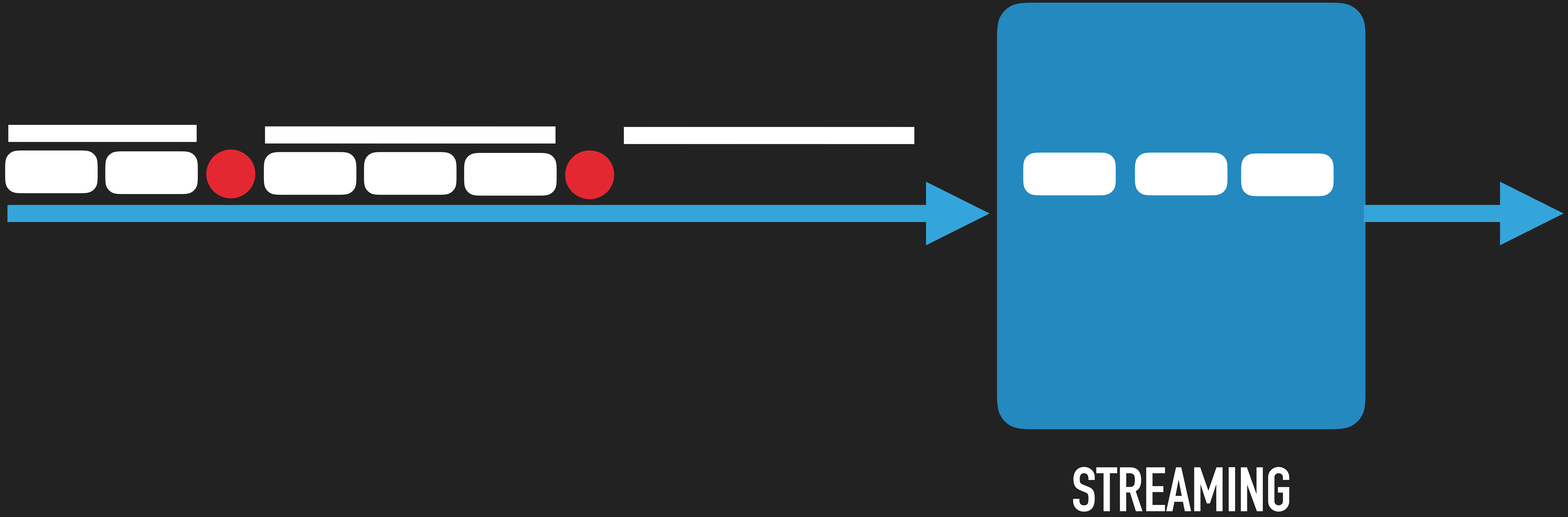
AGGREGATION IN ASYNCHRONOUS MODEL



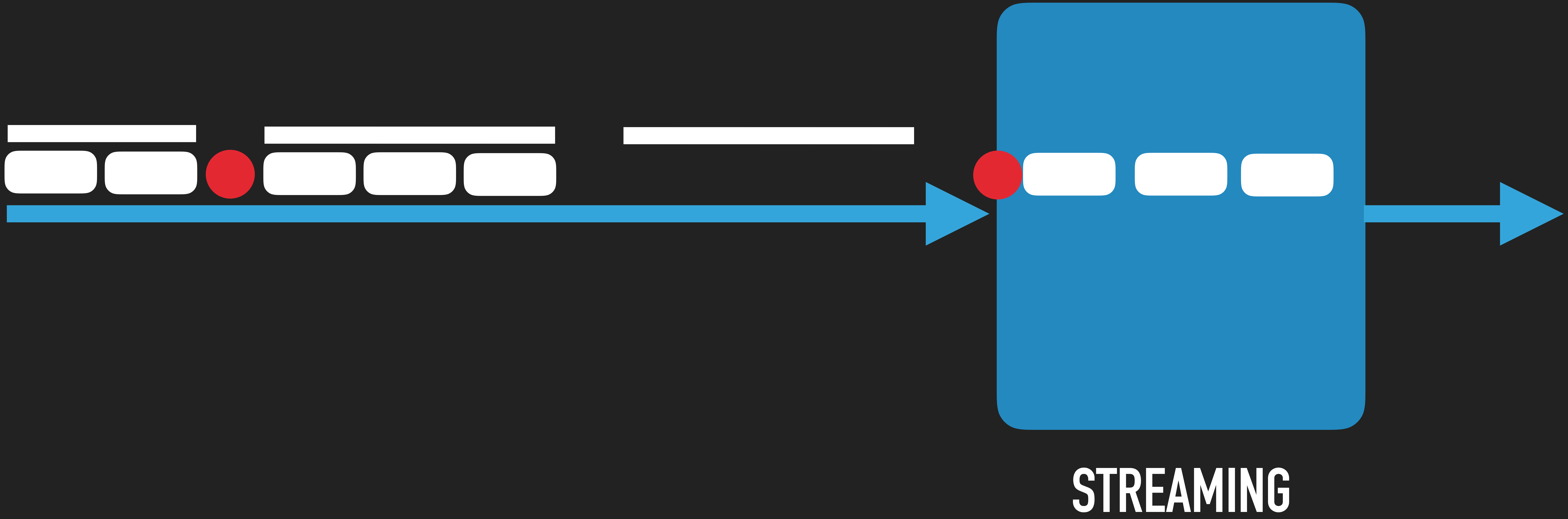
AGGREGATION IN ASYNCHRONOUS MODEL



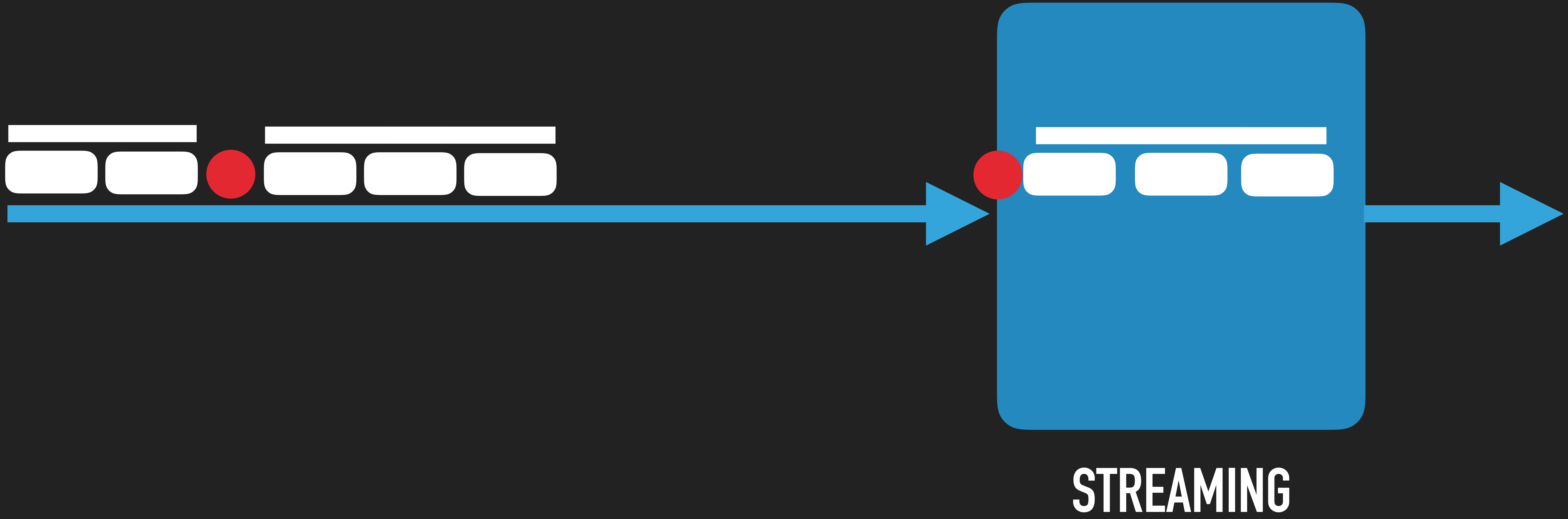
AGGREGATION IN ASYNCHRONOUS MODEL



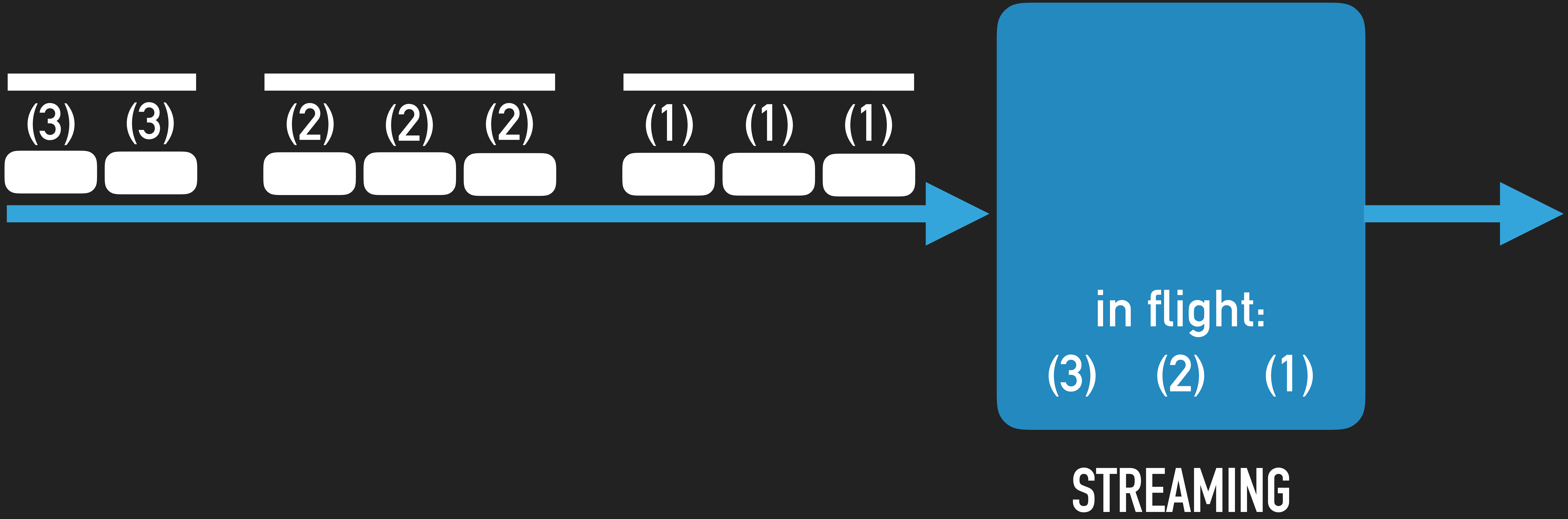
AGGREGATION IN ASYNCHRONOUS MODEL



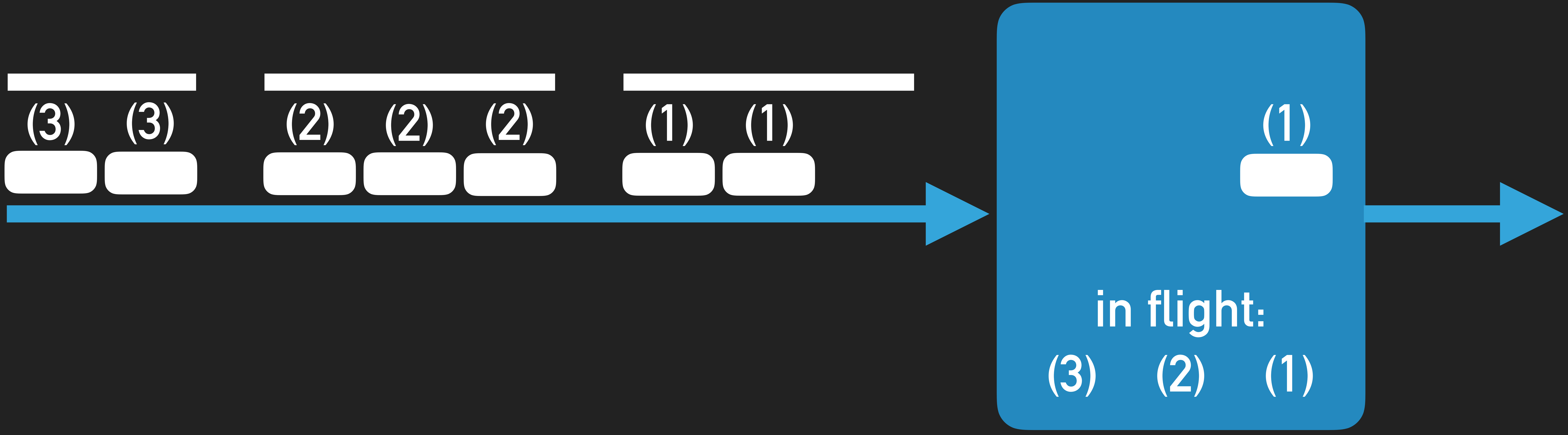
AGGREGATION IN ASYNCHRONOUS MODEL



AGGREGATION IN ASYNCHRONOUS MODEL

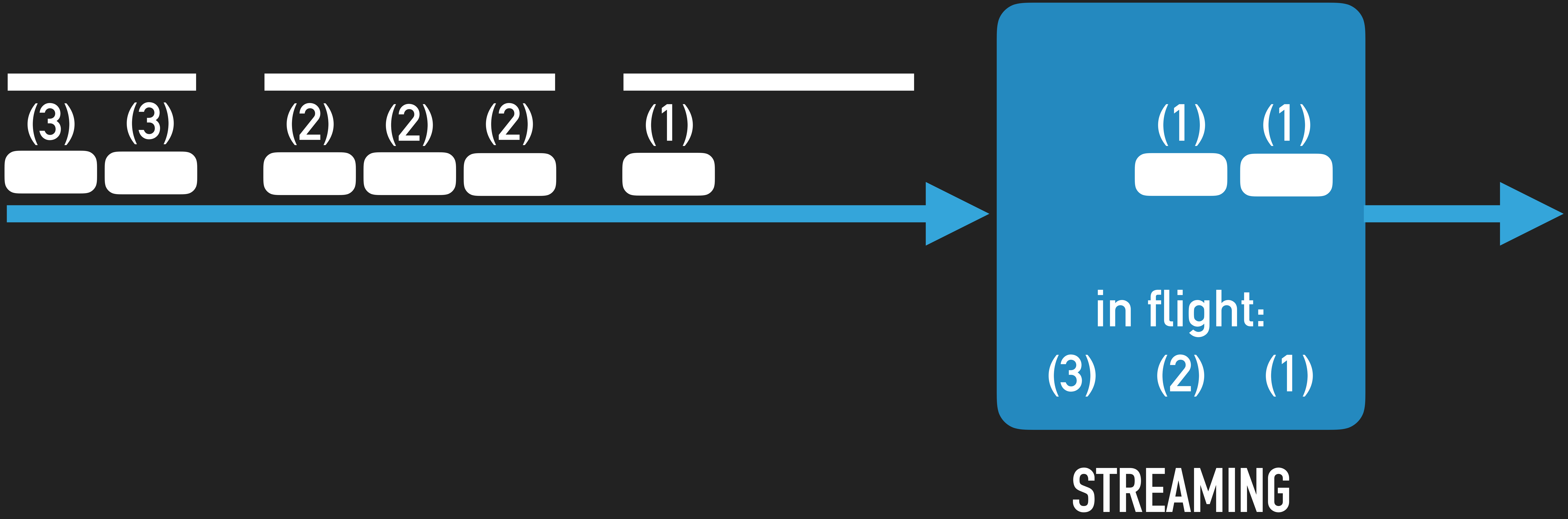


AGGREGATION IN ASYNCHRONOUS MODEL

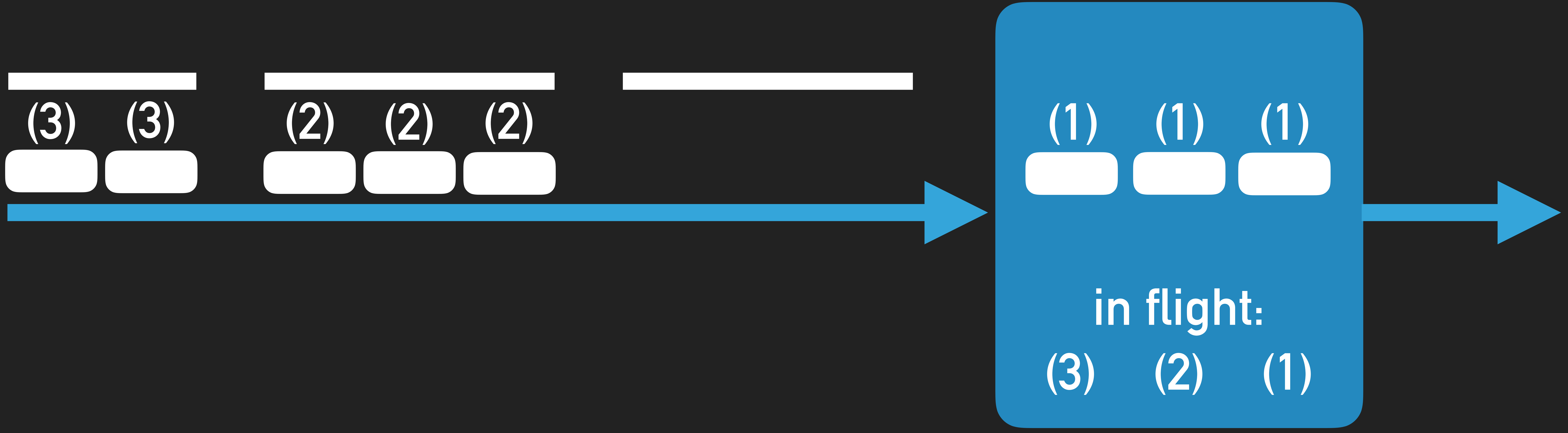


STREAMING

AGGREGATION IN ASYNCHRONOUS MODEL

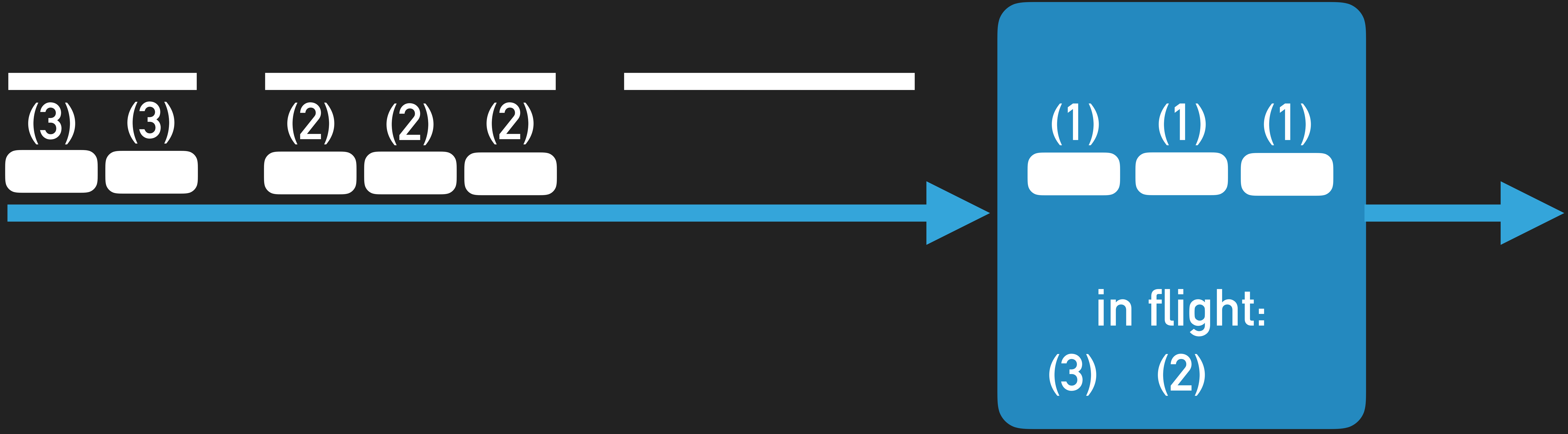


AGGREGATION IN ASYNCHRONOUS MODEL



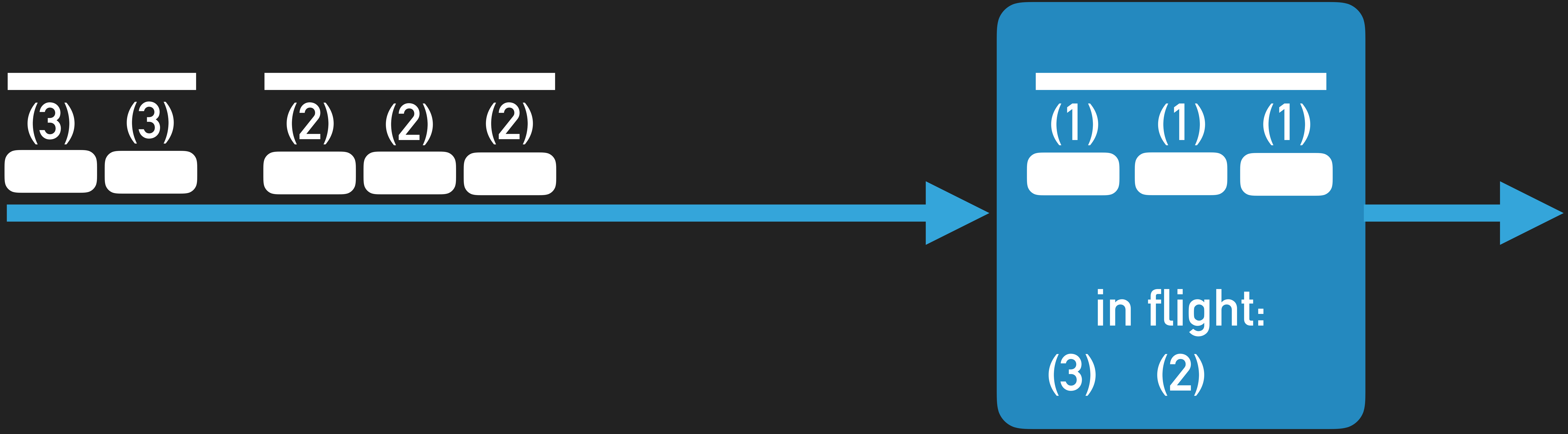
STREAMING

AGGREGATION IN ASYNCHRONOUS MODEL



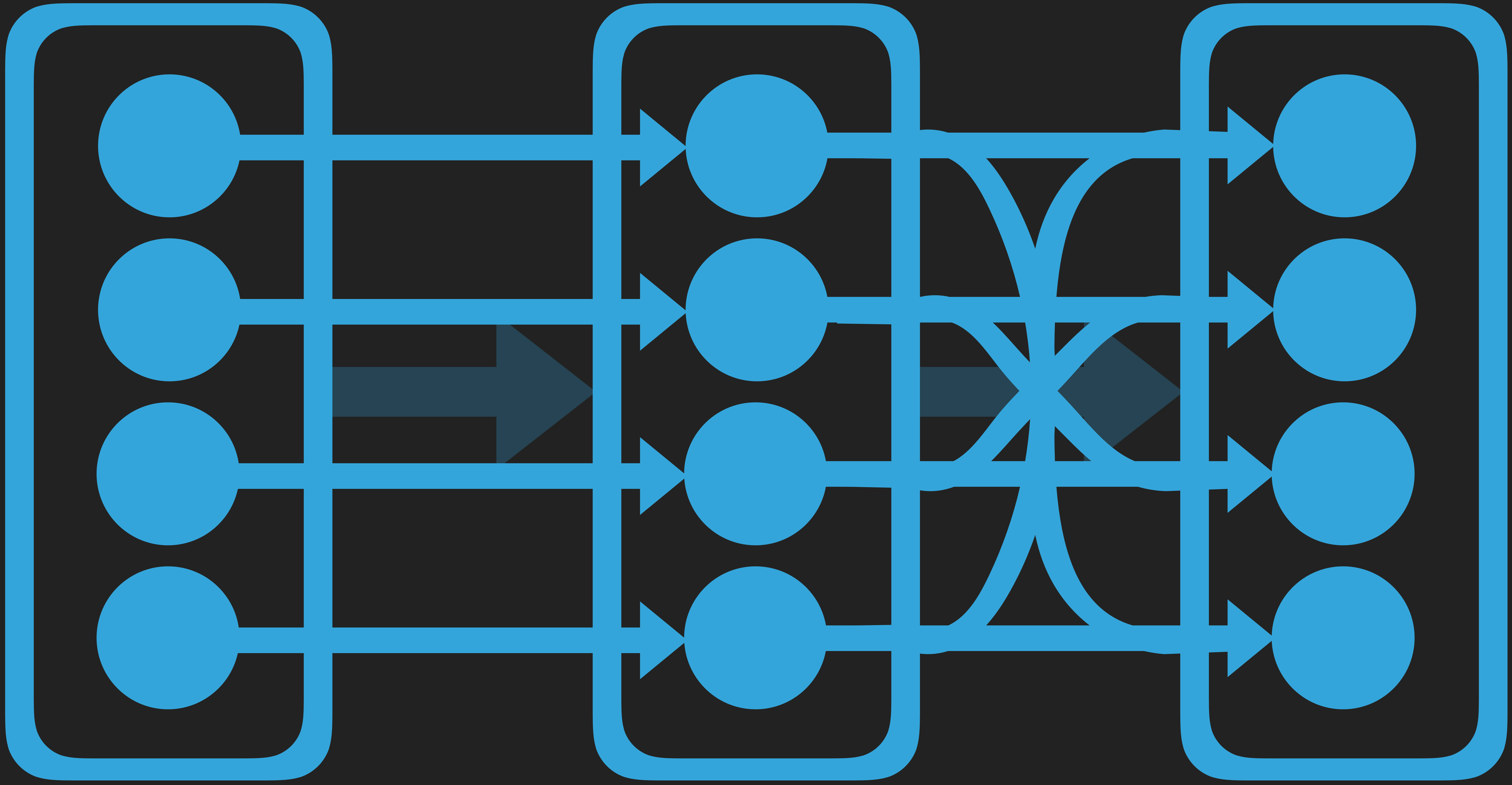
STREAMING

AGGREGATION IN ASYNCHRONOUS MODEL

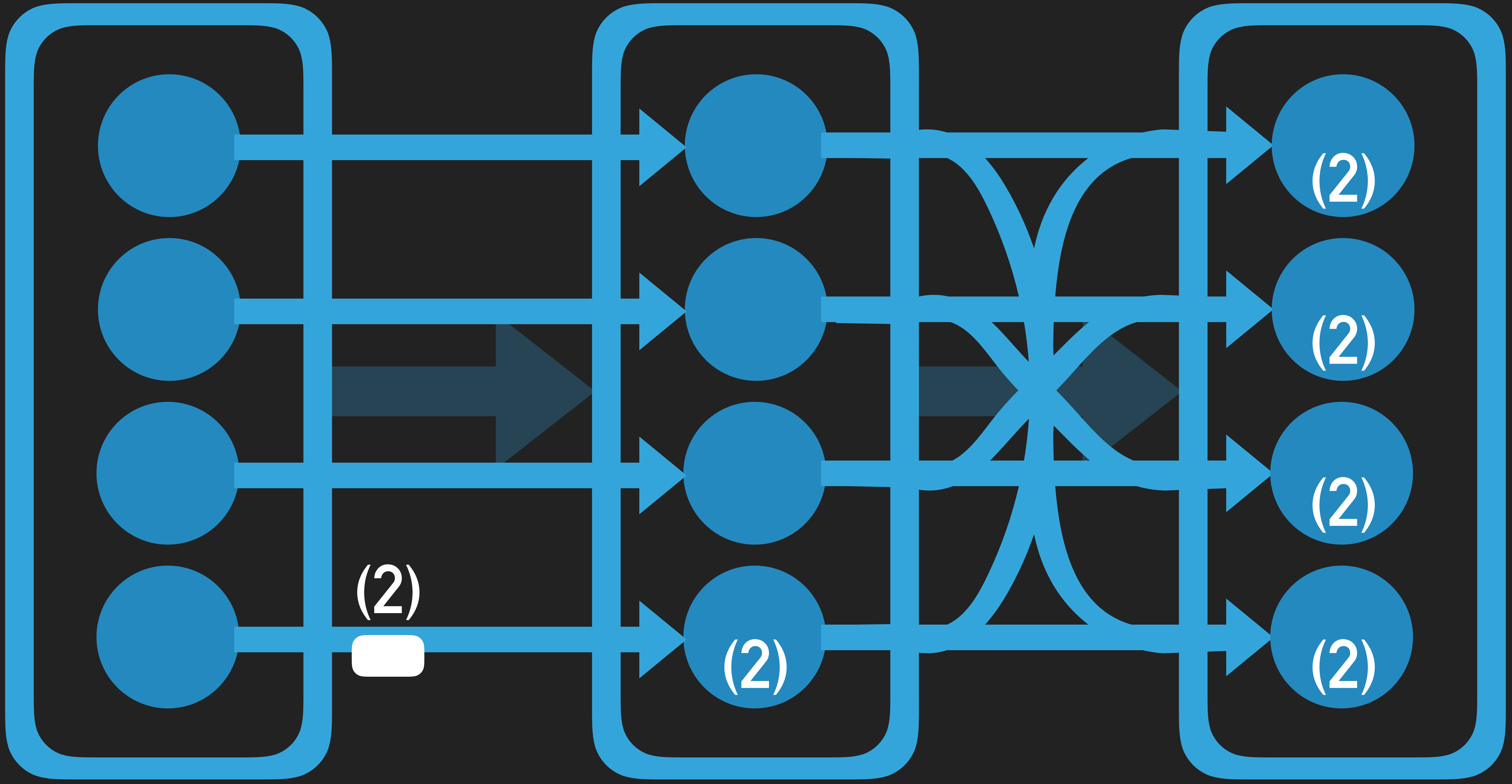


STREAMING

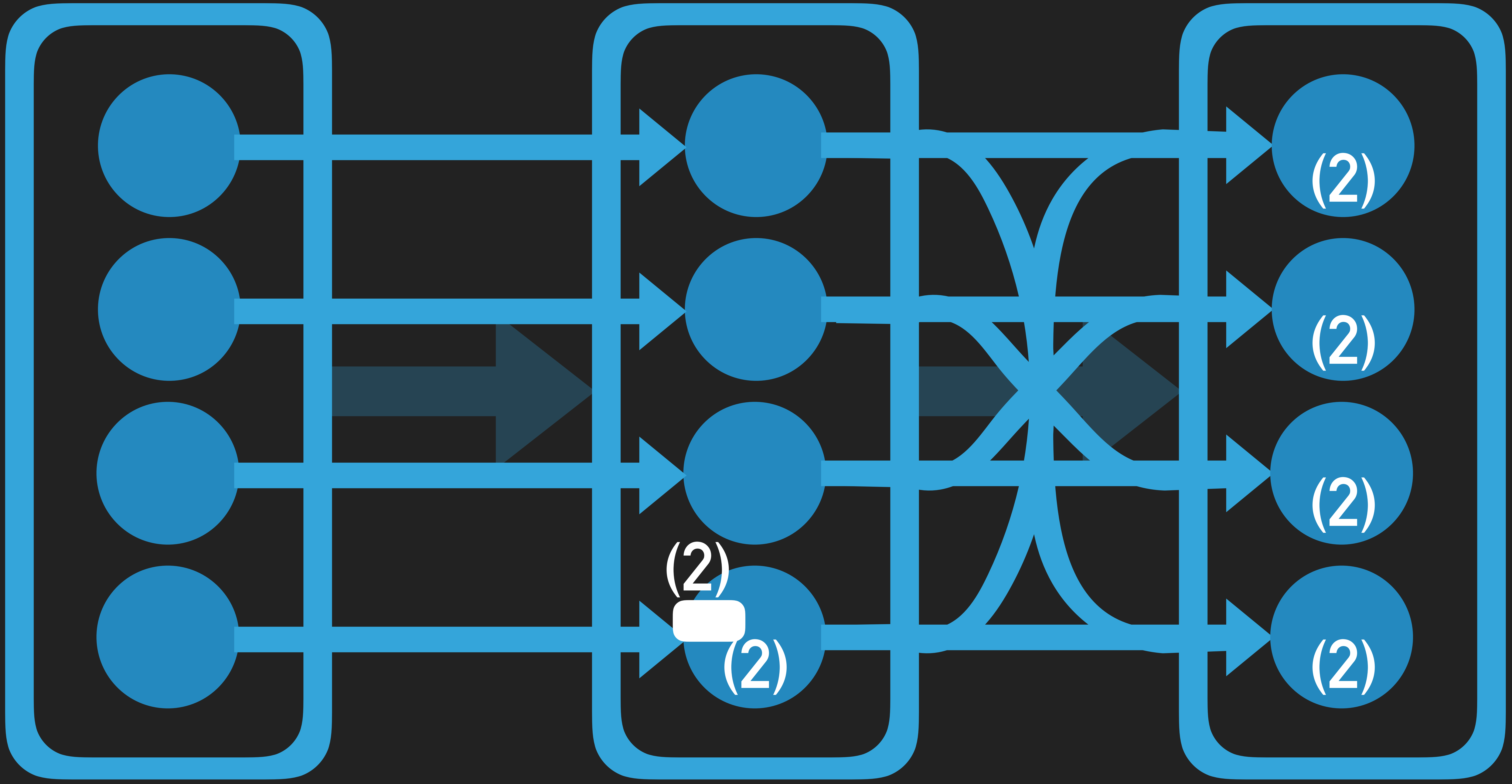
PROGRESS TRACKING



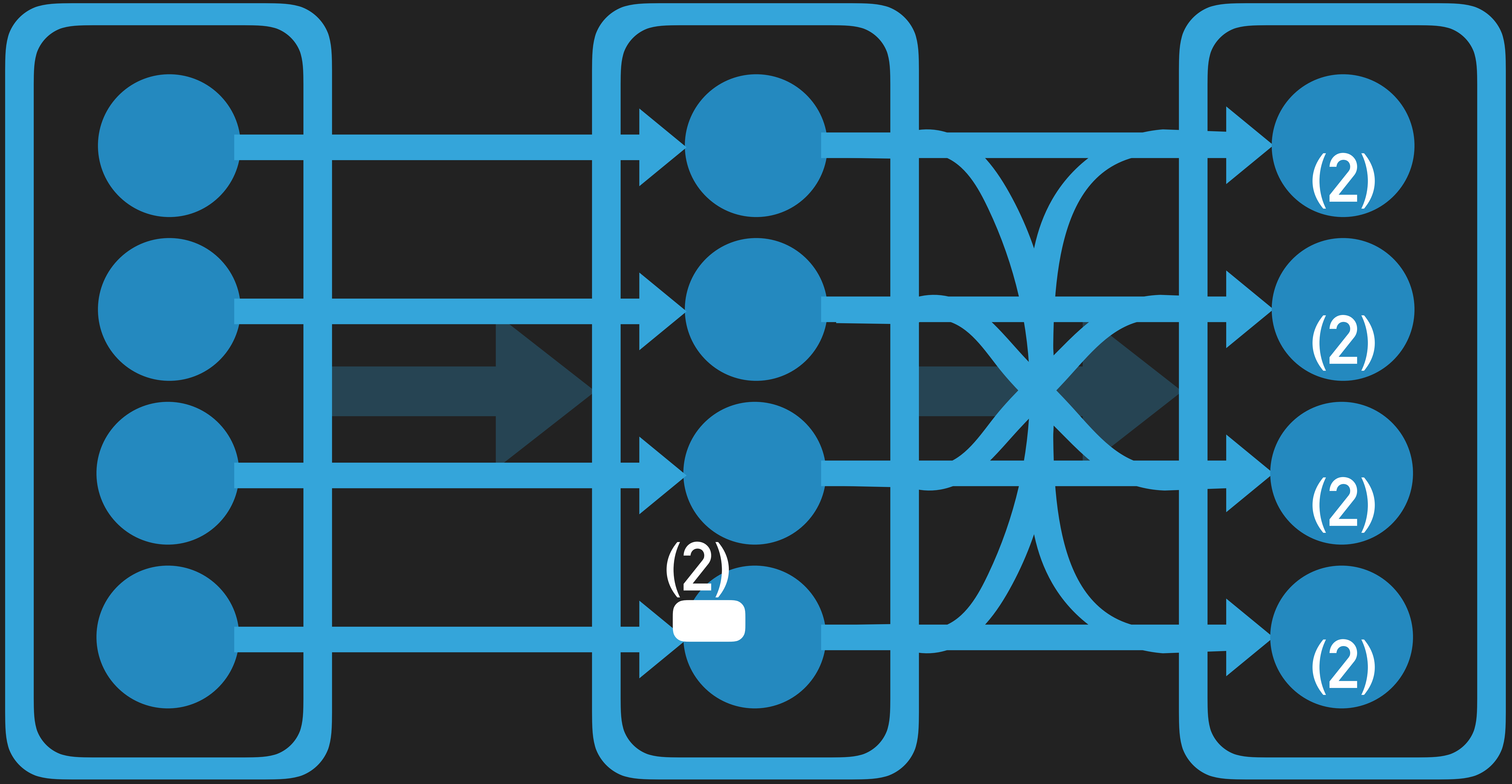
PROGRESS TRACKING



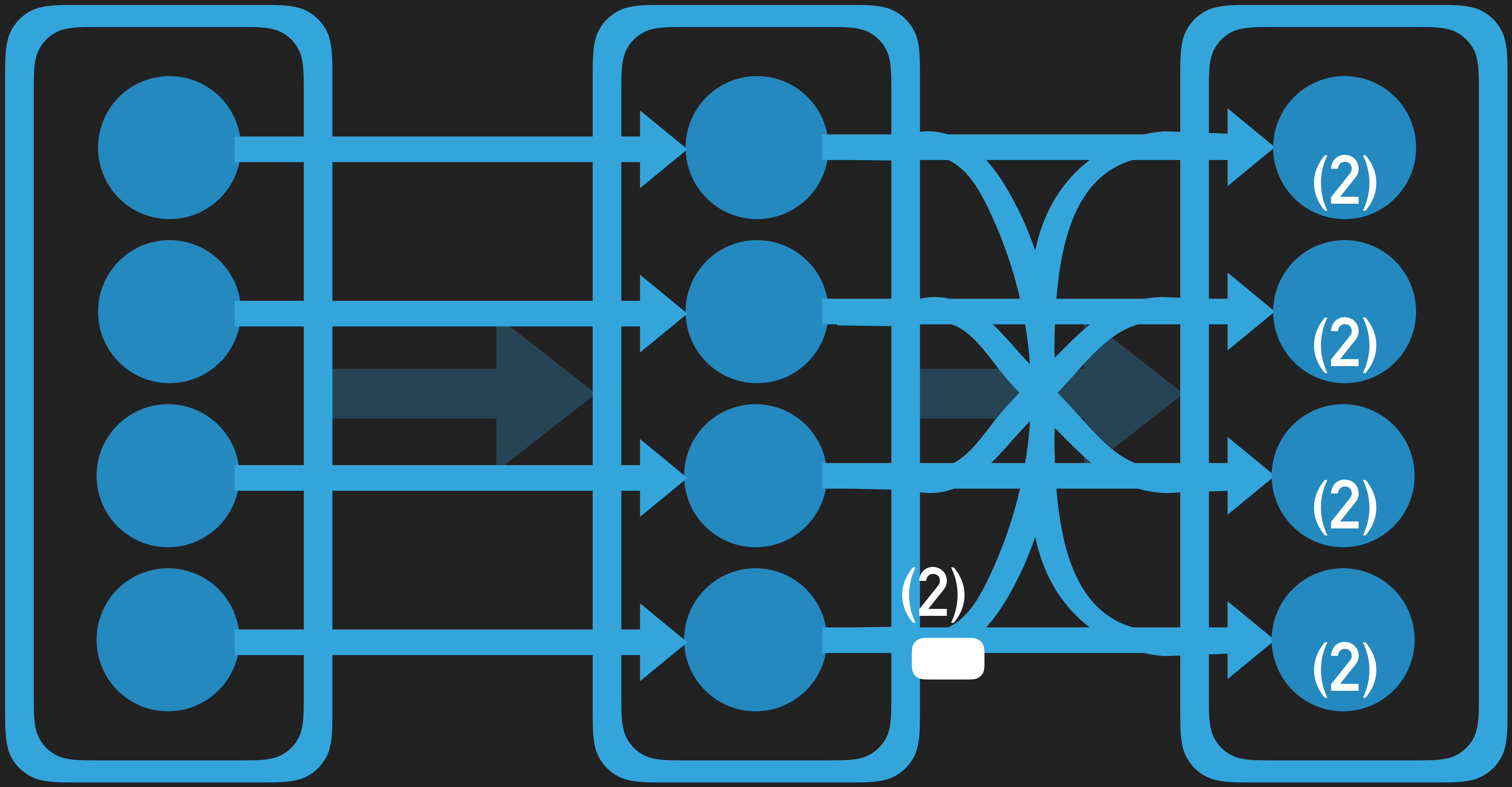
PROGRESS TRACKING



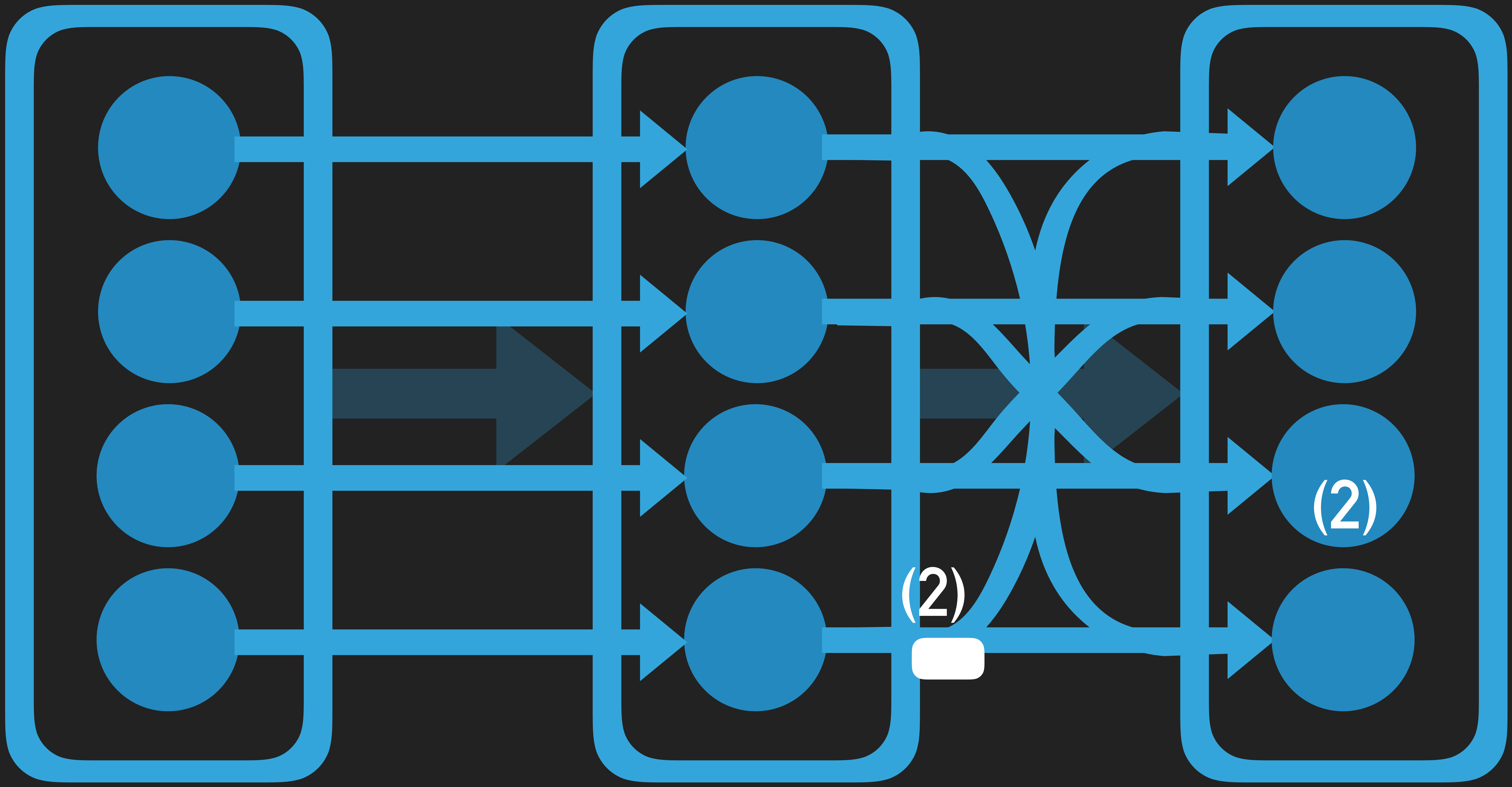
PROGRESS TRACKING



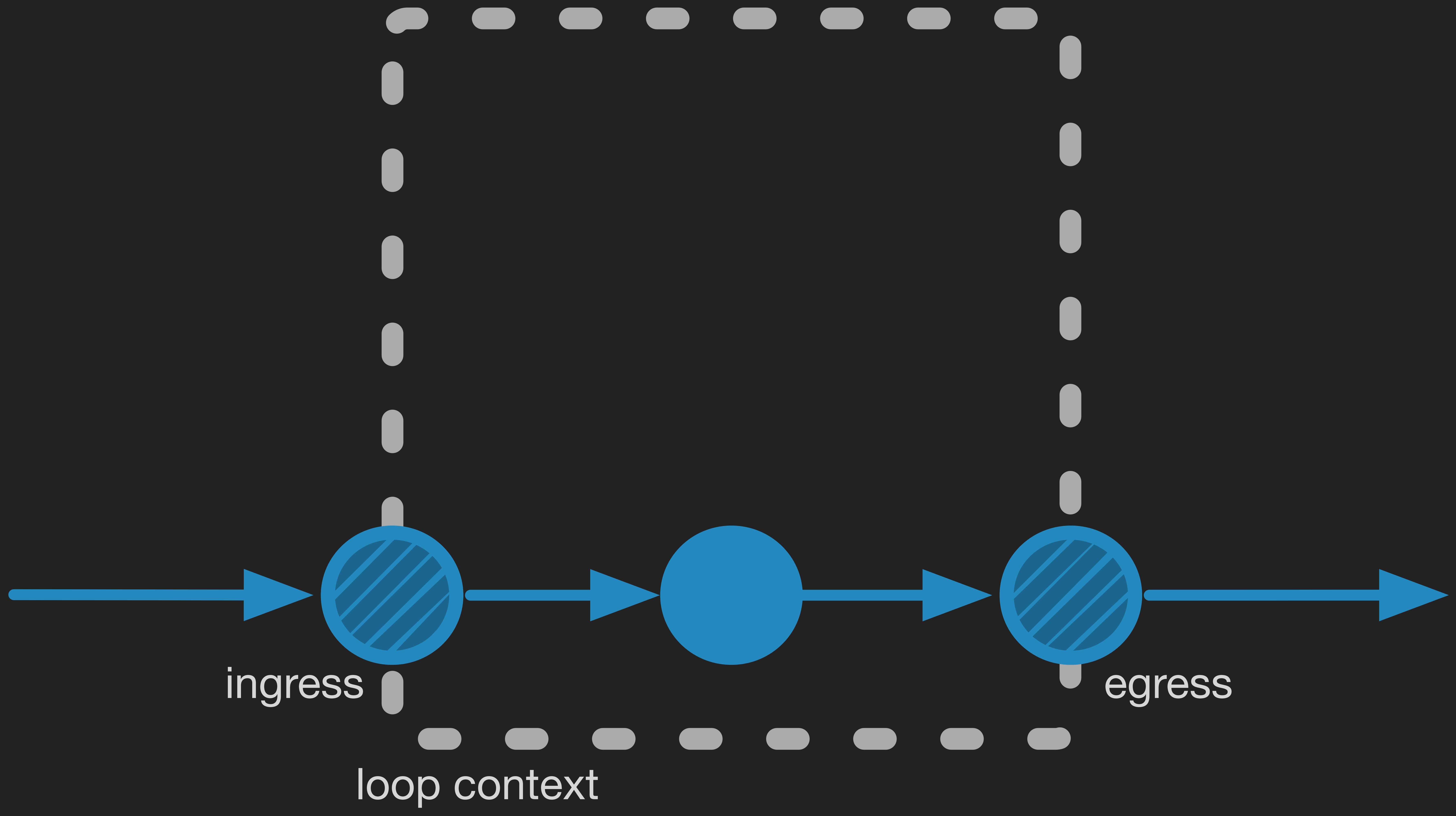
PROGRESS TRACKING



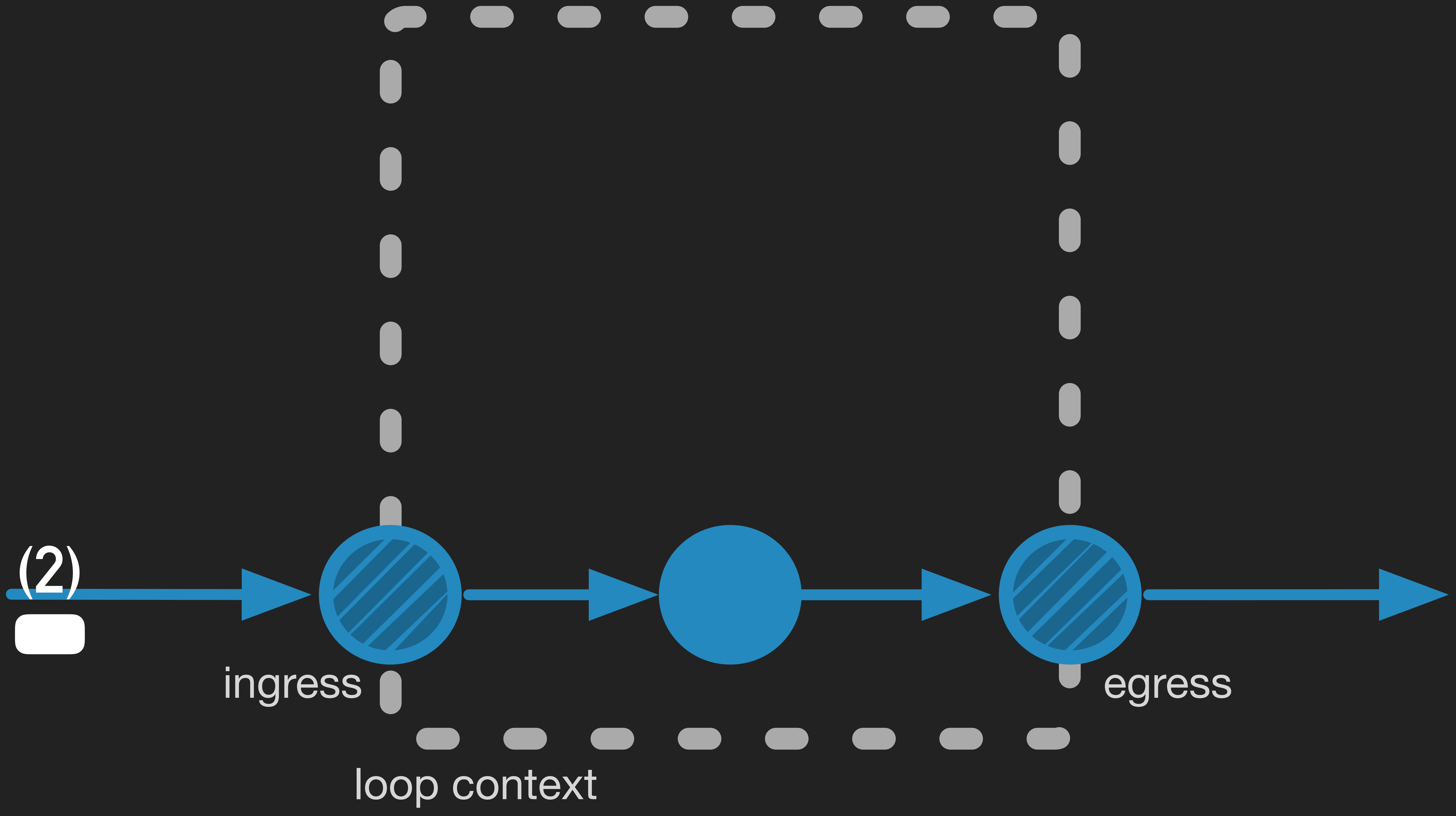
PROGRESS TRACKING



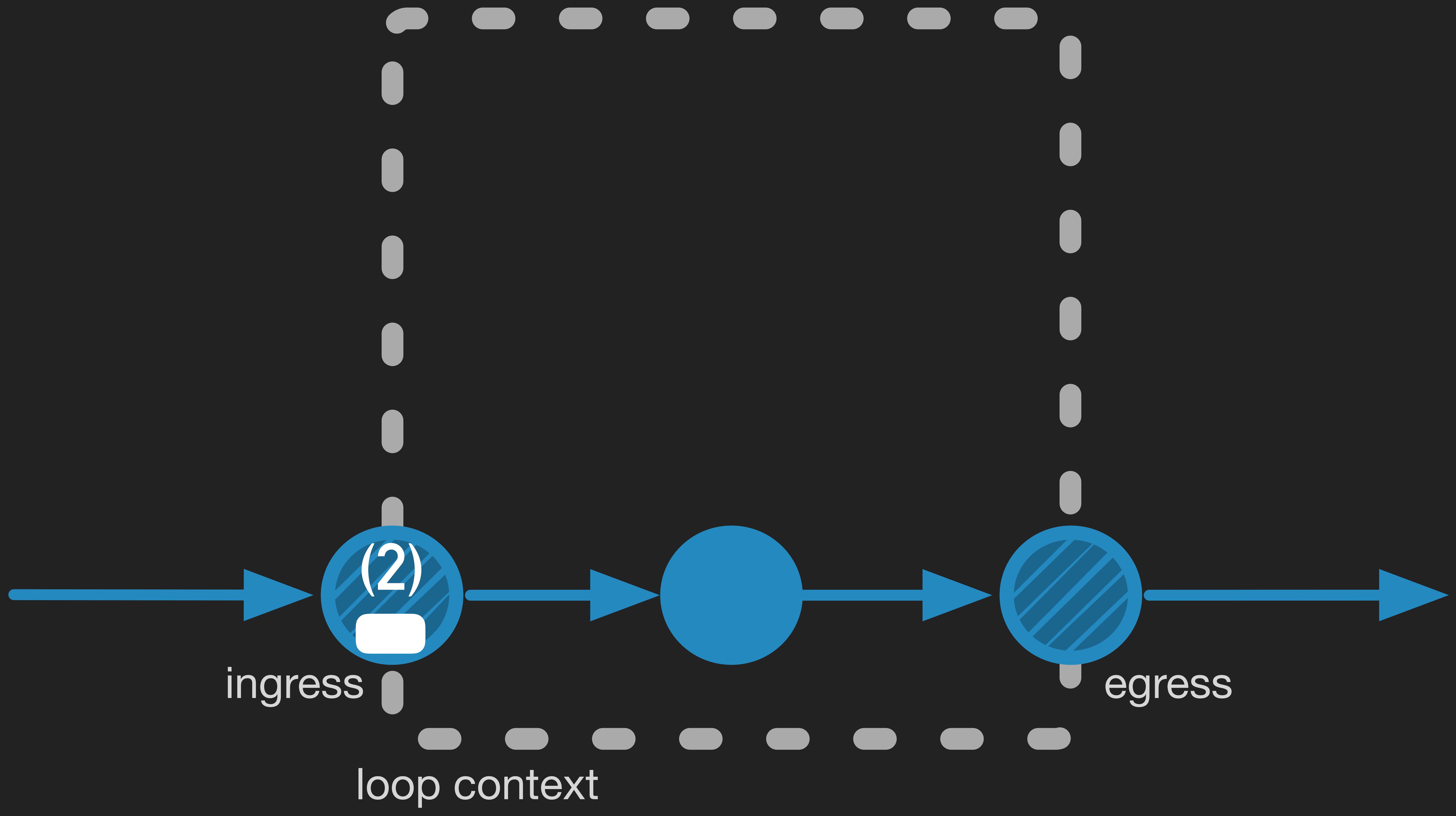
CYCLICAL DATAFLOW: LOOP CONTEXTS



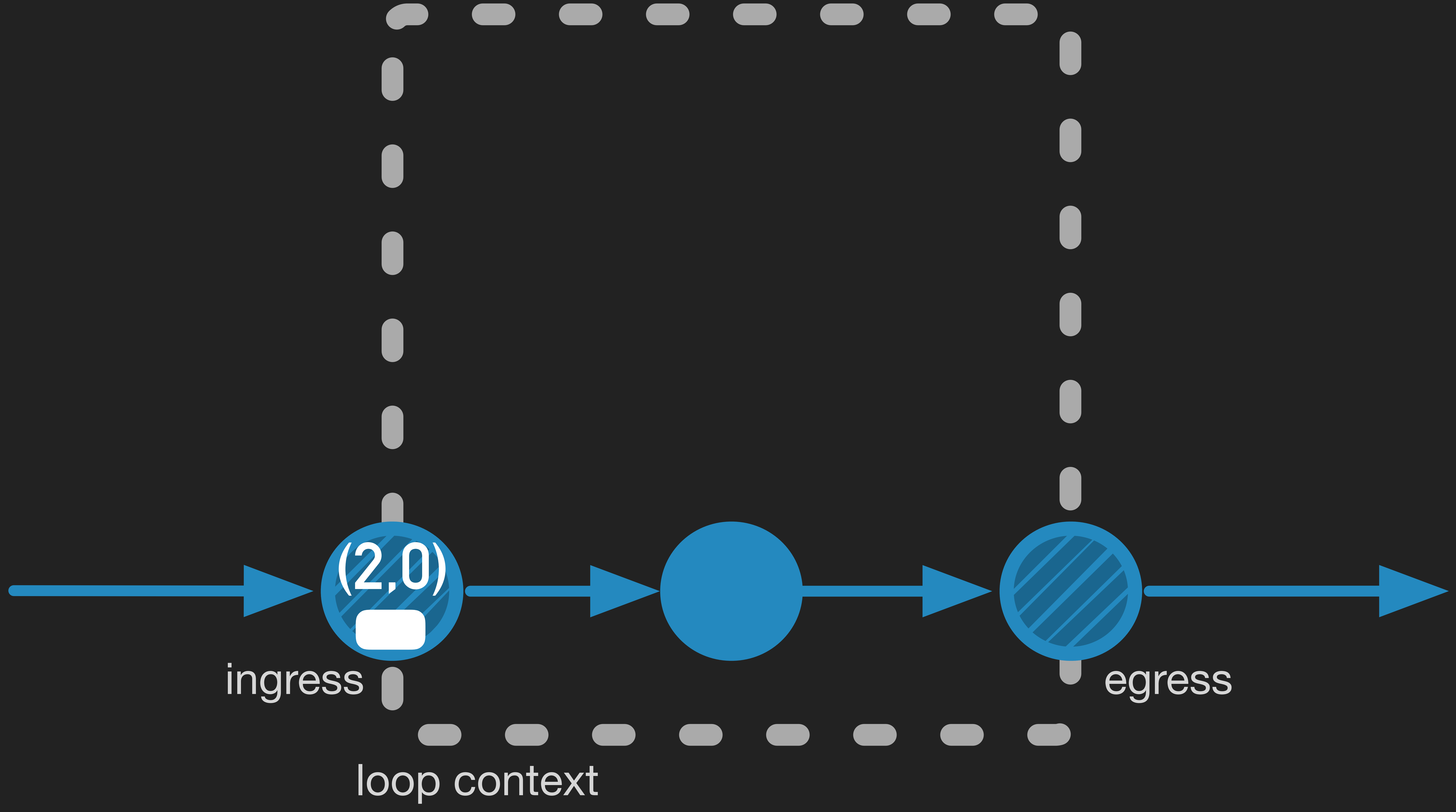
CYCLICAL DATAFLOW: LOOP CONTEXTS



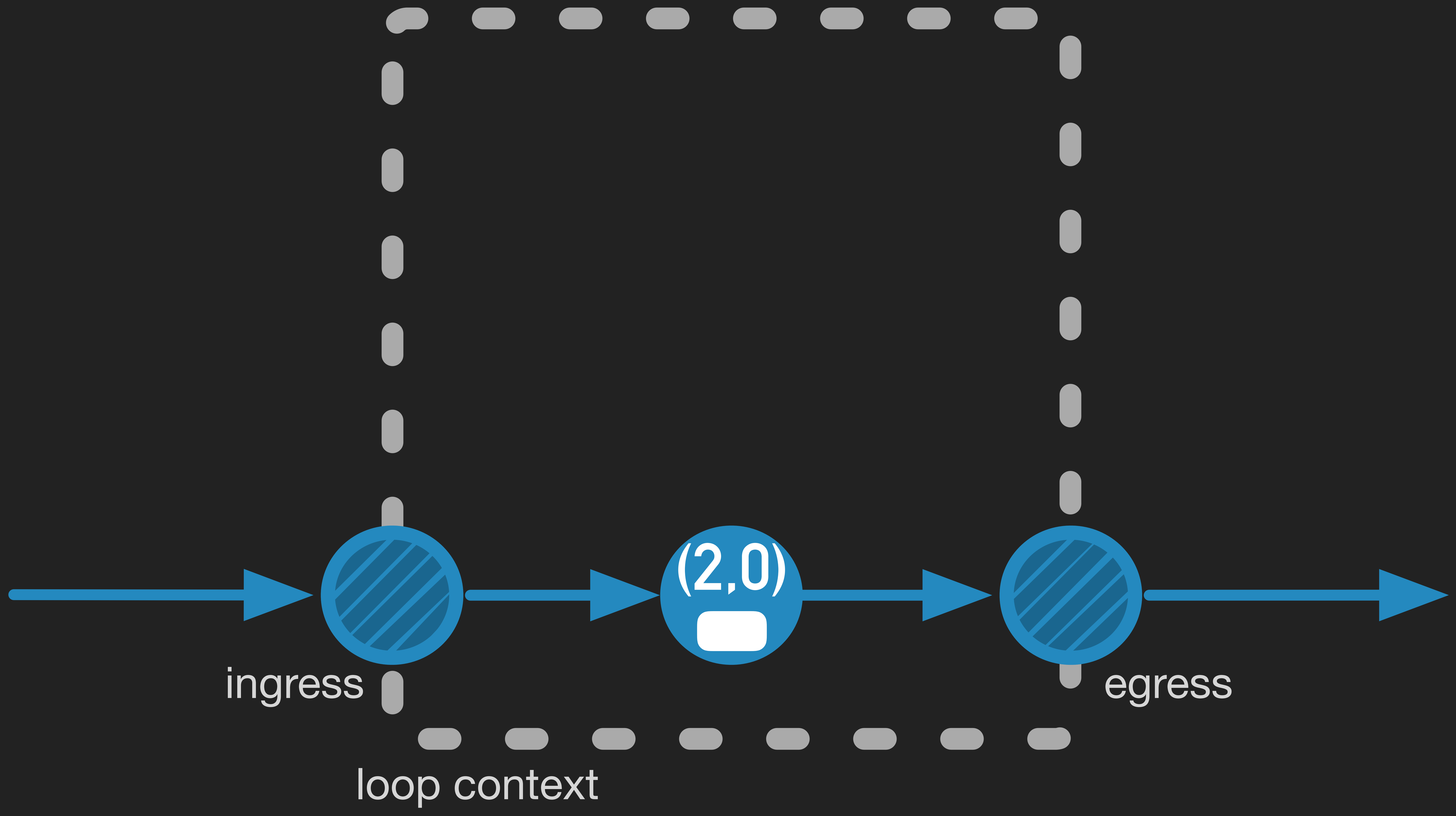
CYCLICAL DATAFLOW: LOOP CONTEXTS



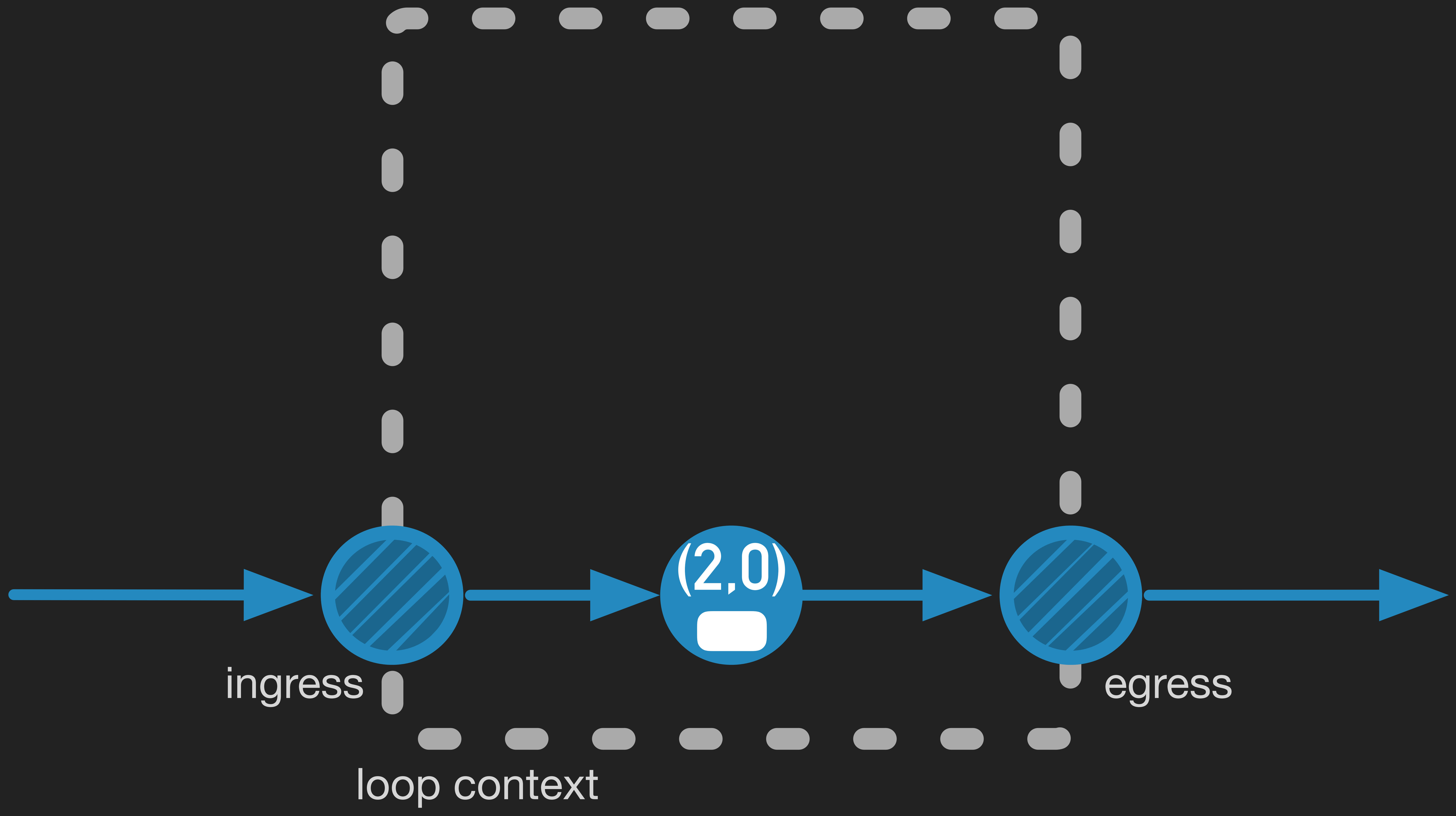
CYCLICAL DATAFLOW: LOOP CONTEXTS



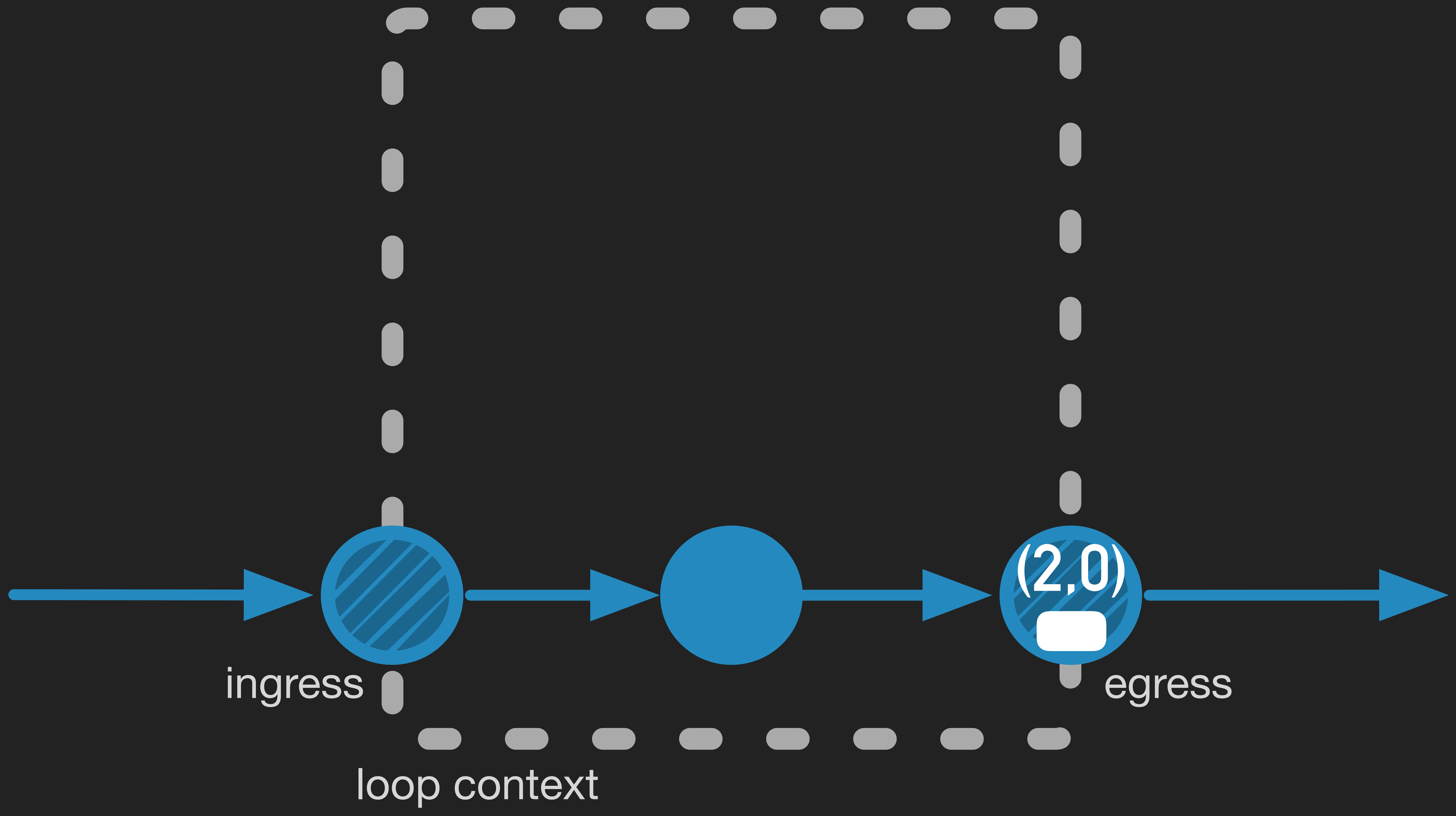
CYCLICAL DATAFLOW: LOOP CONTEXTS



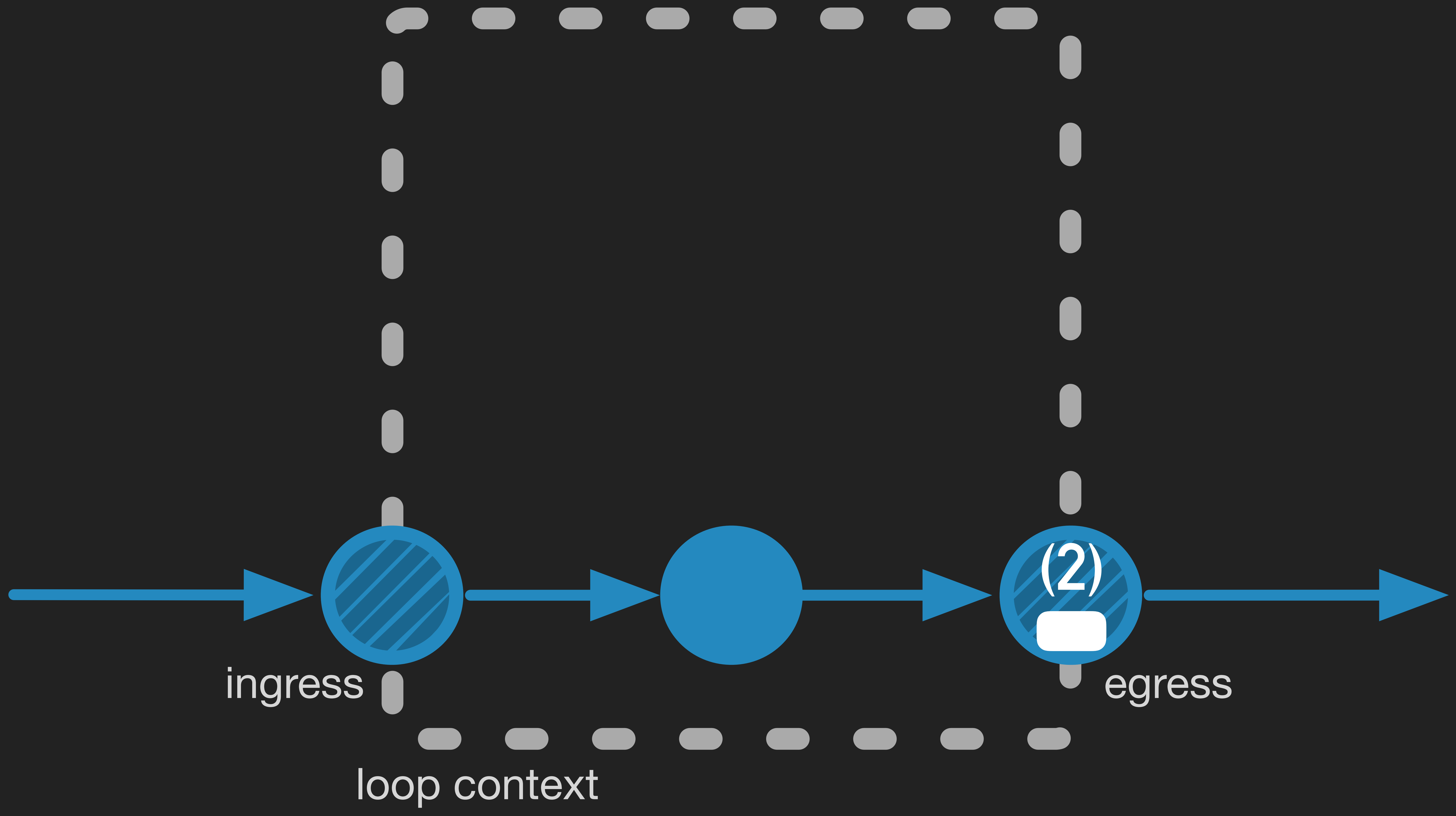
CYCLICAL DATAFLOW: LOOP CONTEXTS



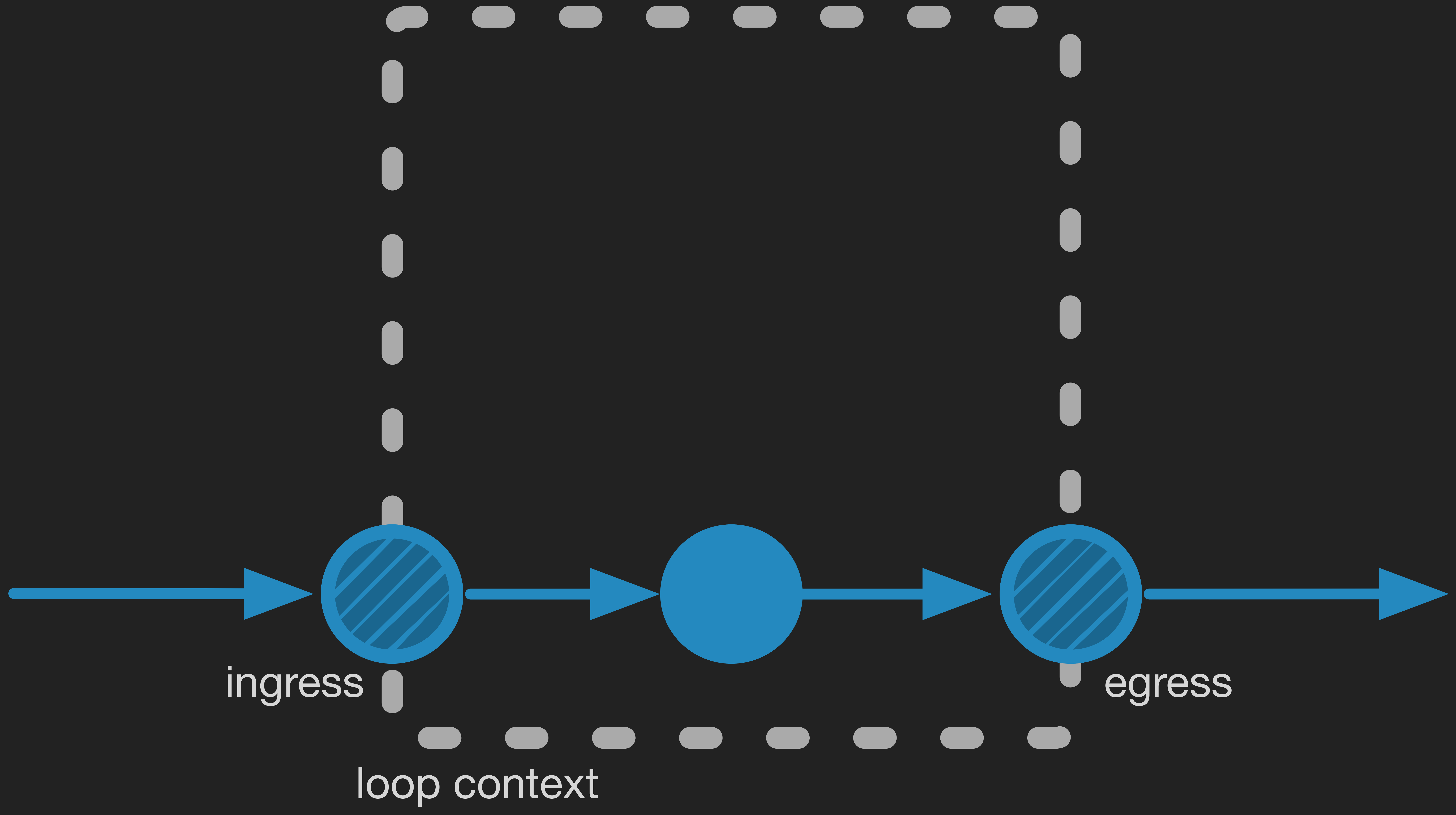
CYCLICAL DATAFLOW: LOOP CONTEXTS



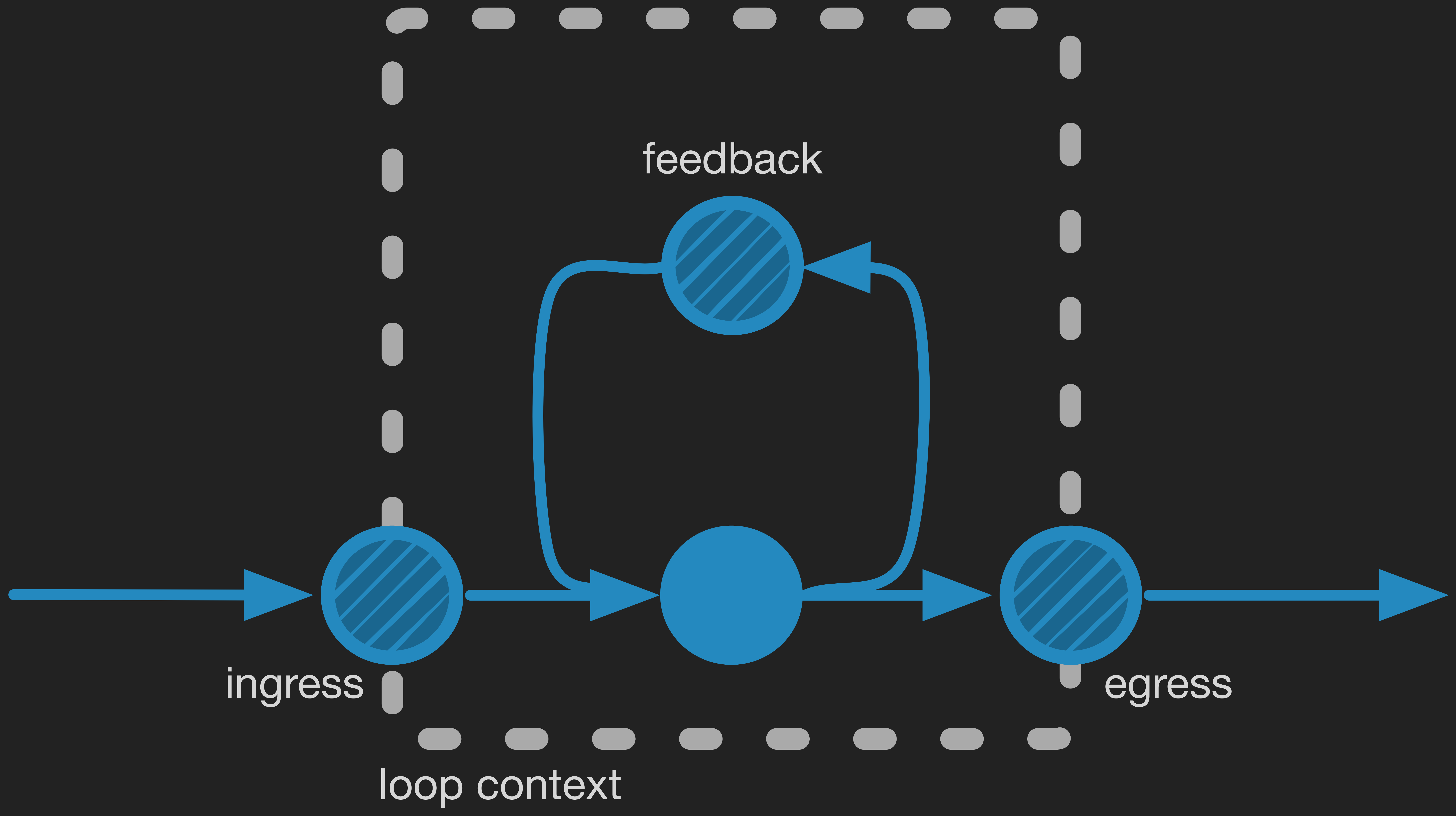
CYCLICAL DATAFLOW: LOOP CONTEXTS



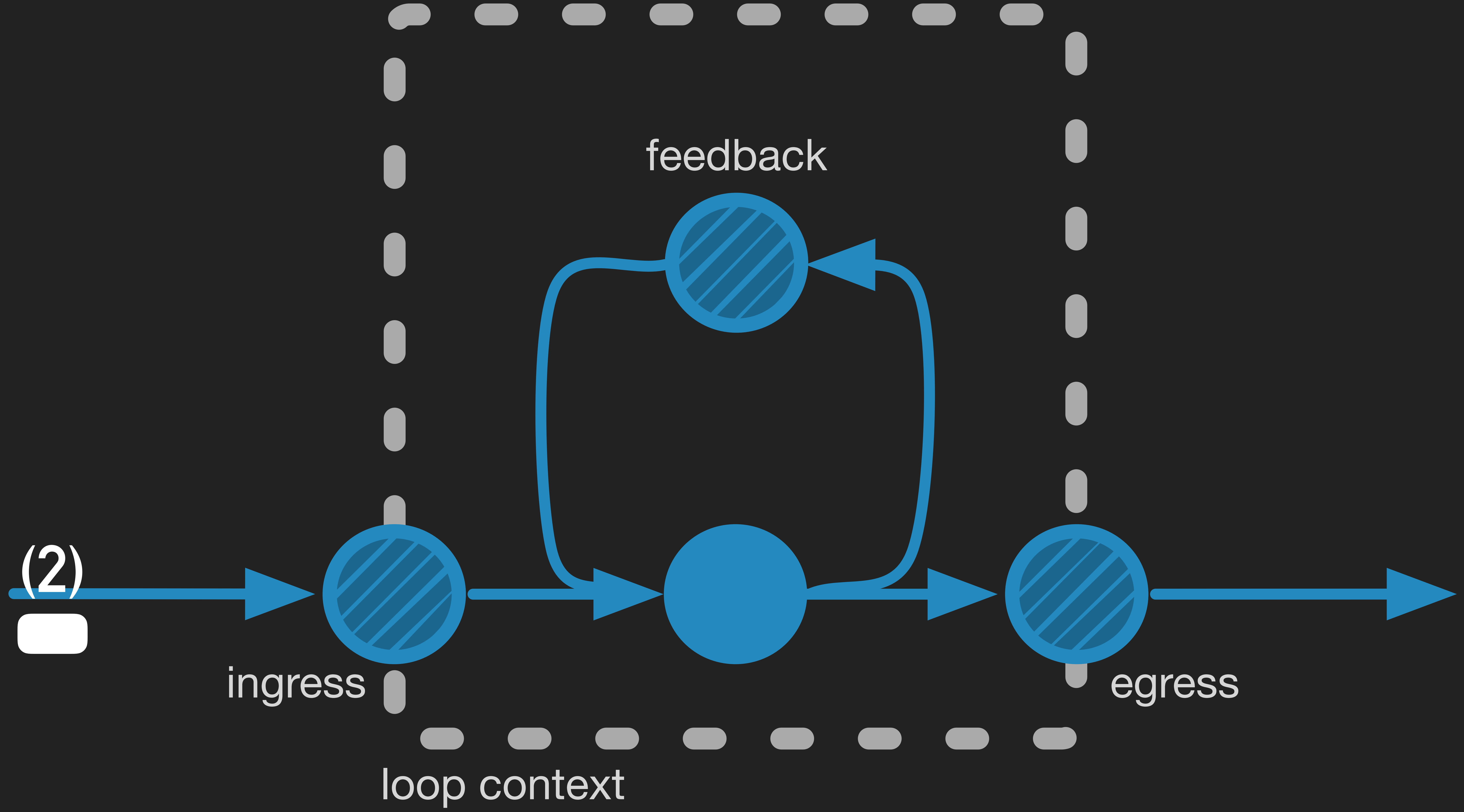
CYCLICAL DATAFLOW: LOOP CONTEXTS



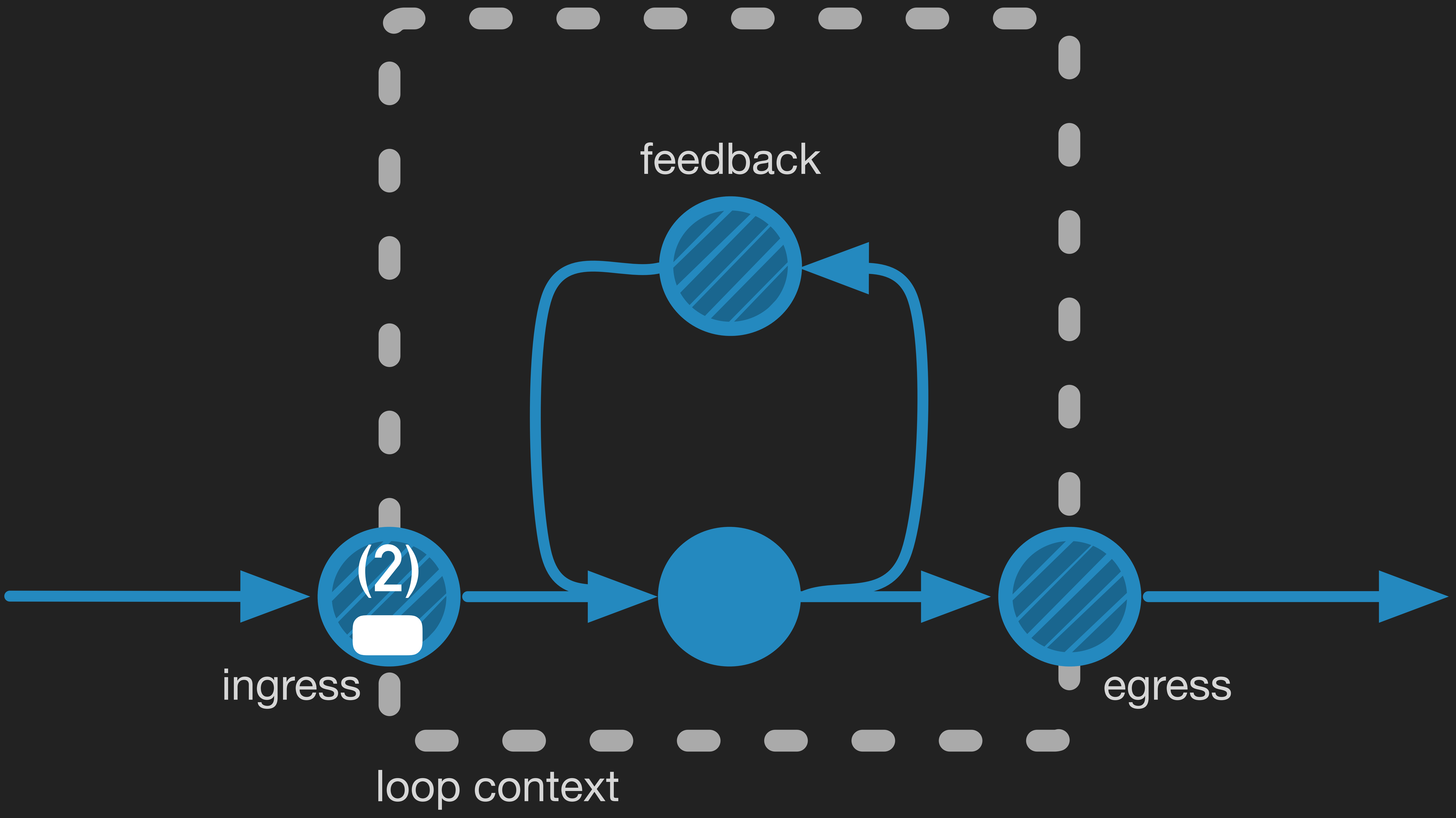
CYCLICAL DATAFLOW: LOOP CONTEXTS



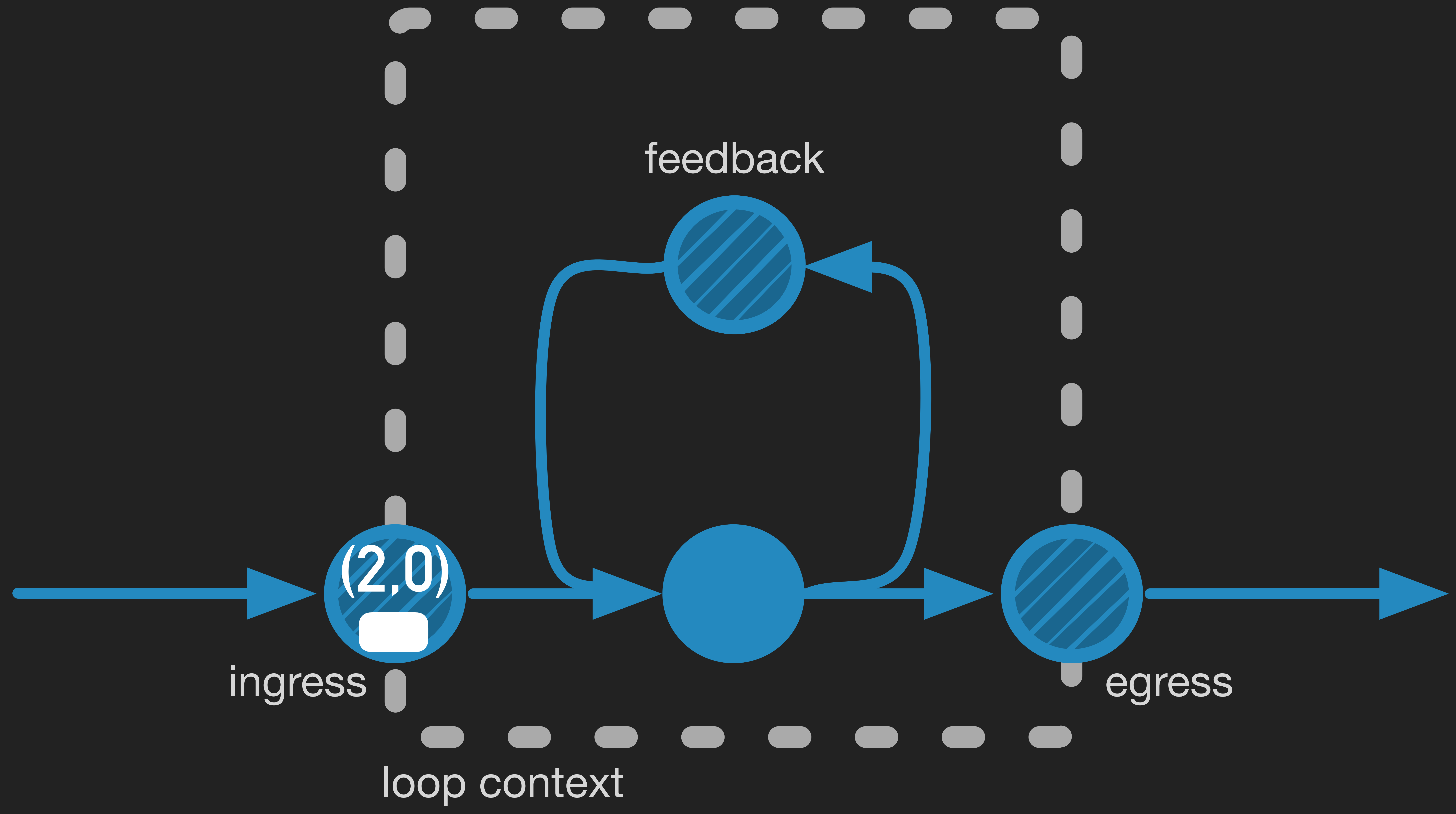
CYCLICAL DATAFLOW: LOOP CONTEXTS



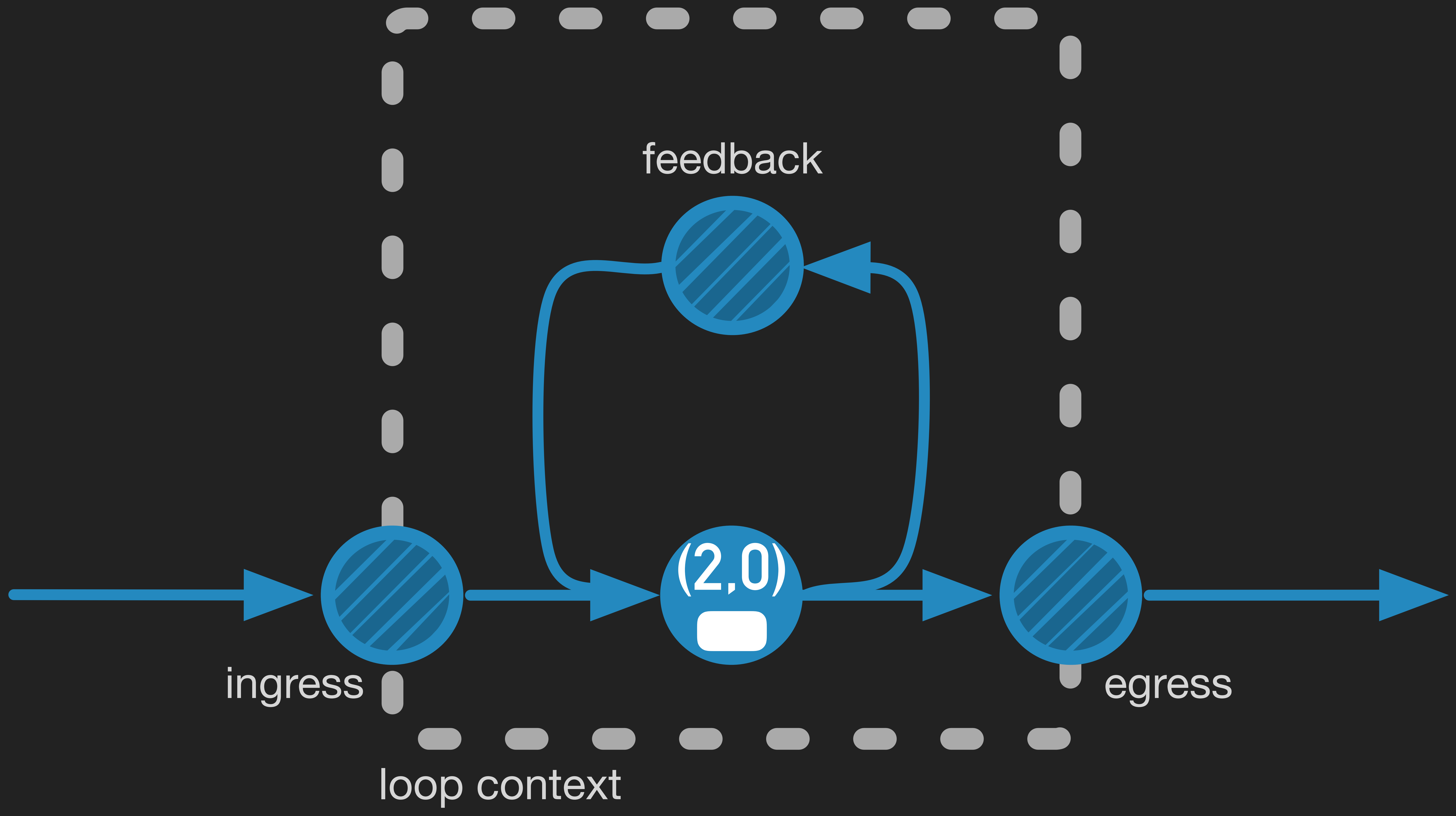
CYCLICAL DATAFLOW: LOOP CONTEXTS



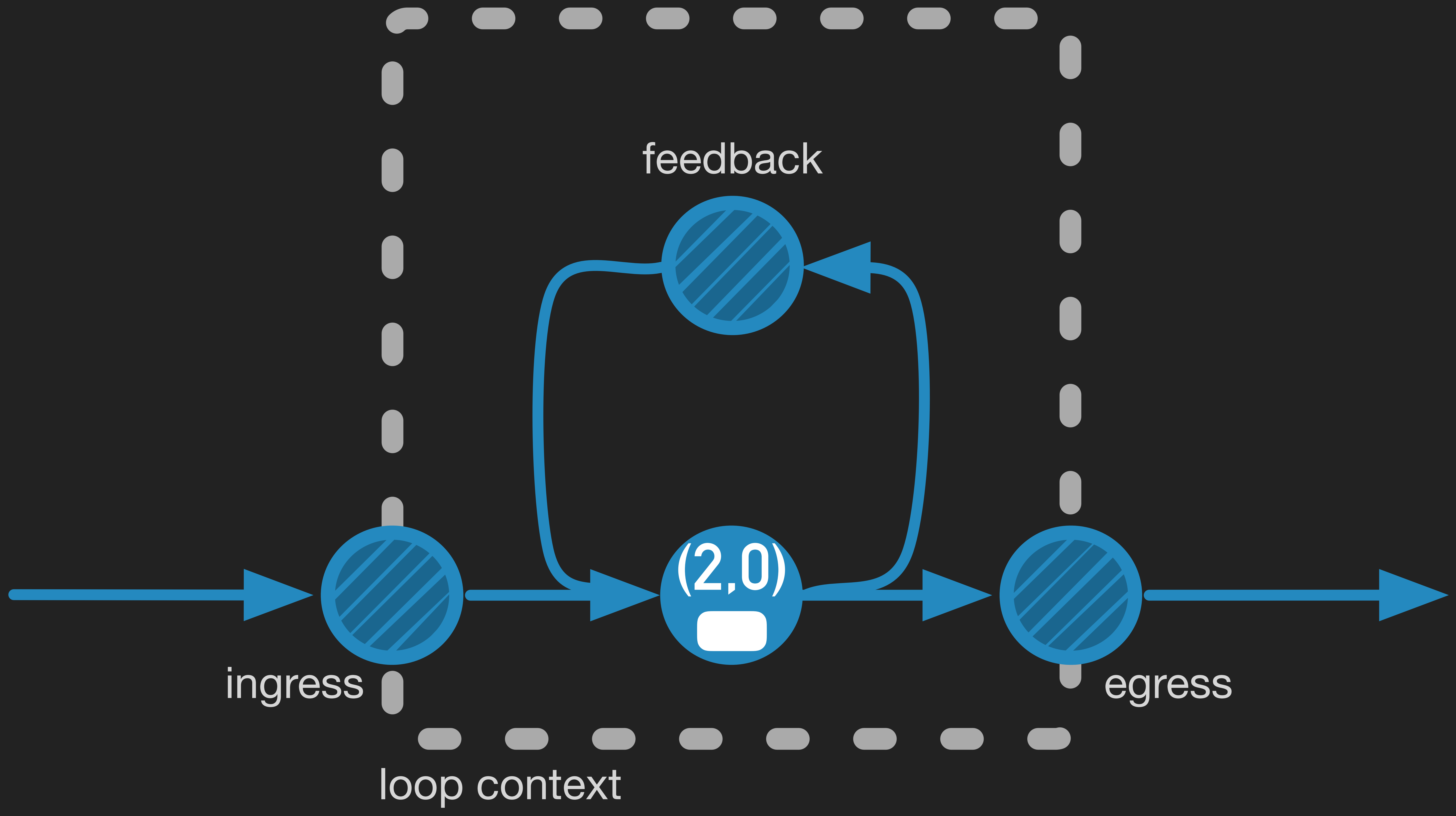
CYCLICAL DATAFLOW: LOOP CONTEXTS



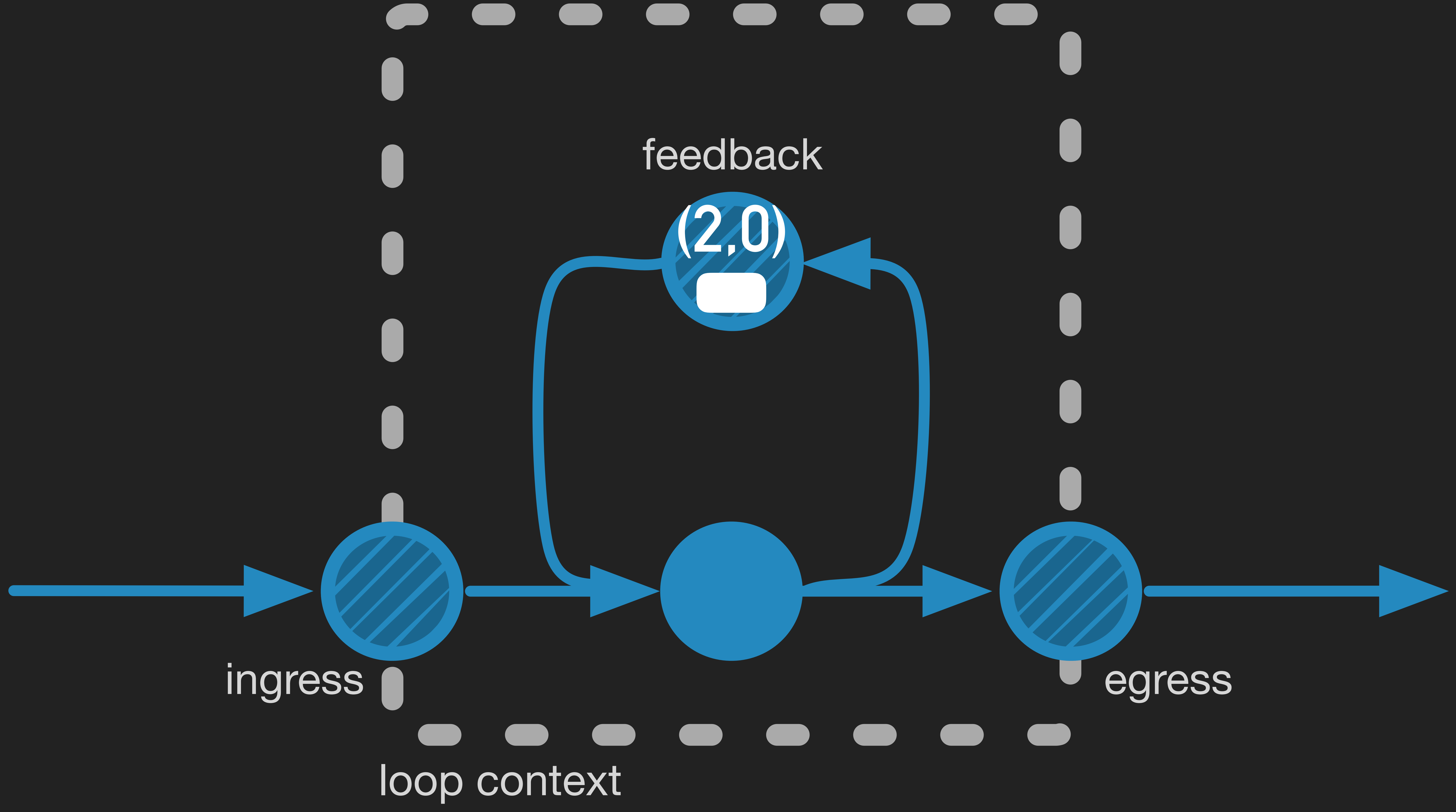
CYCLICAL DATAFLOW: LOOP CONTEXTS



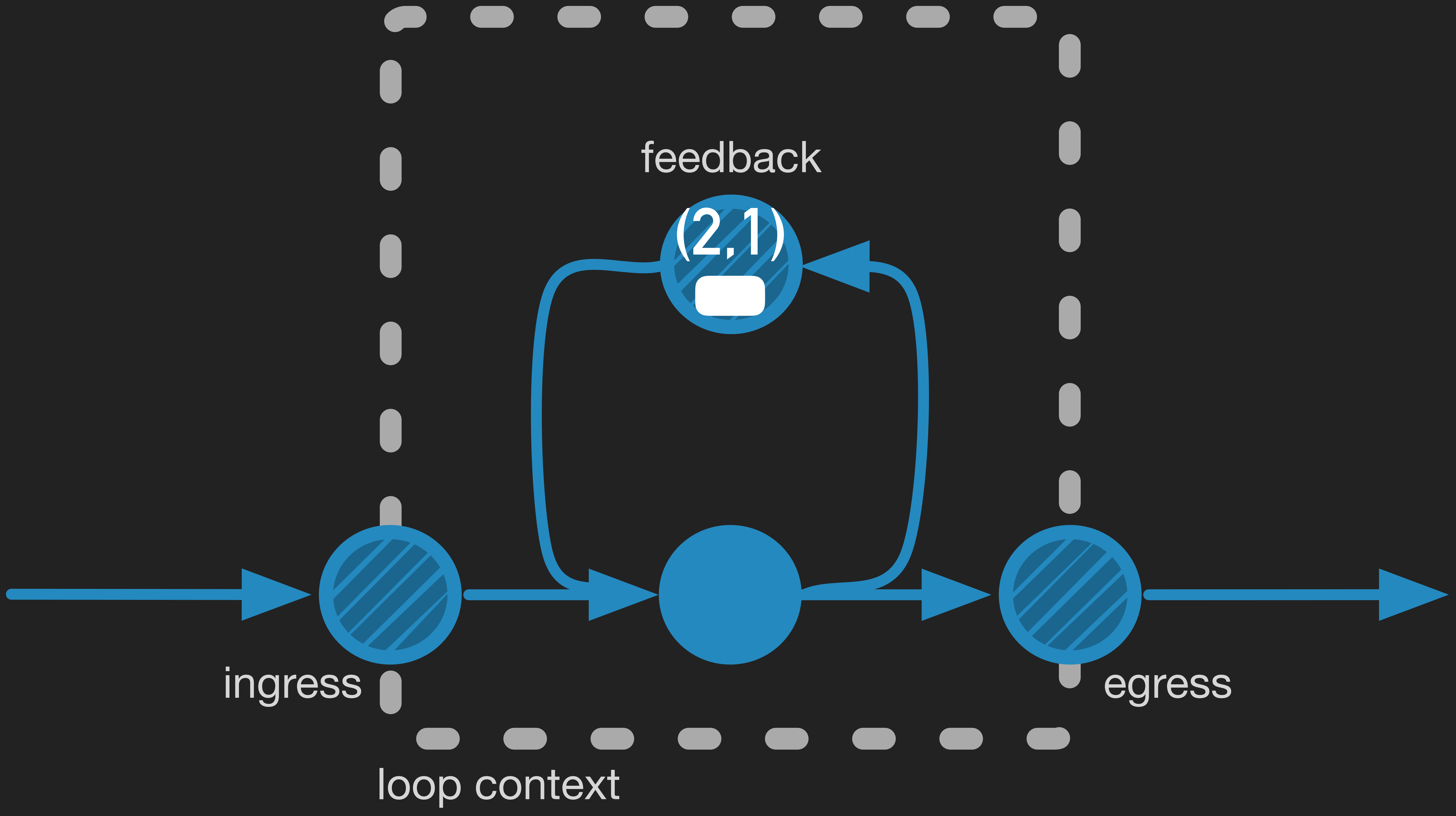
CYCLICAL DATAFLOW: LOOP CONTEXTS



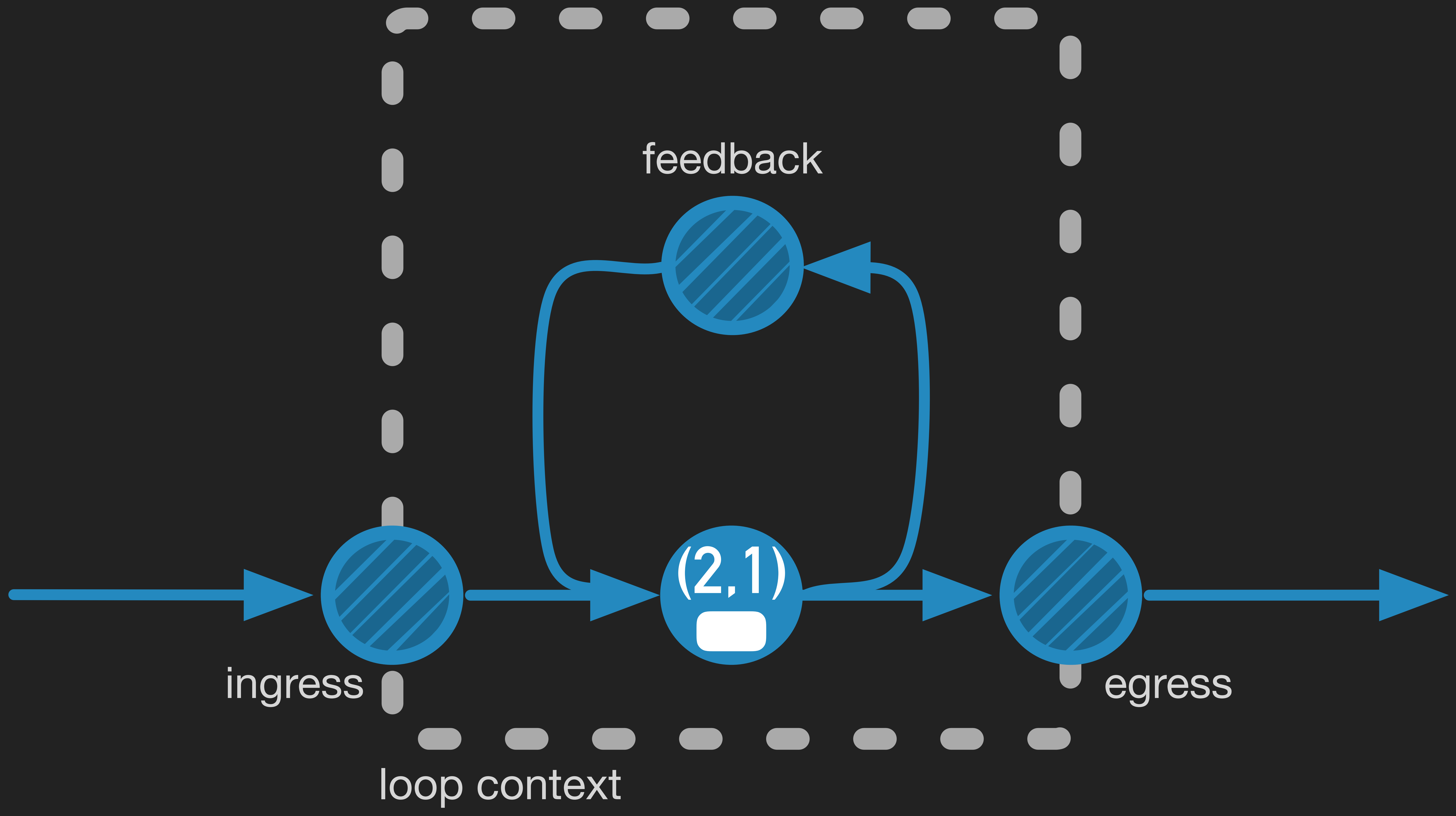
CYCLICAL DATAFLOW: LOOP CONTEXTS



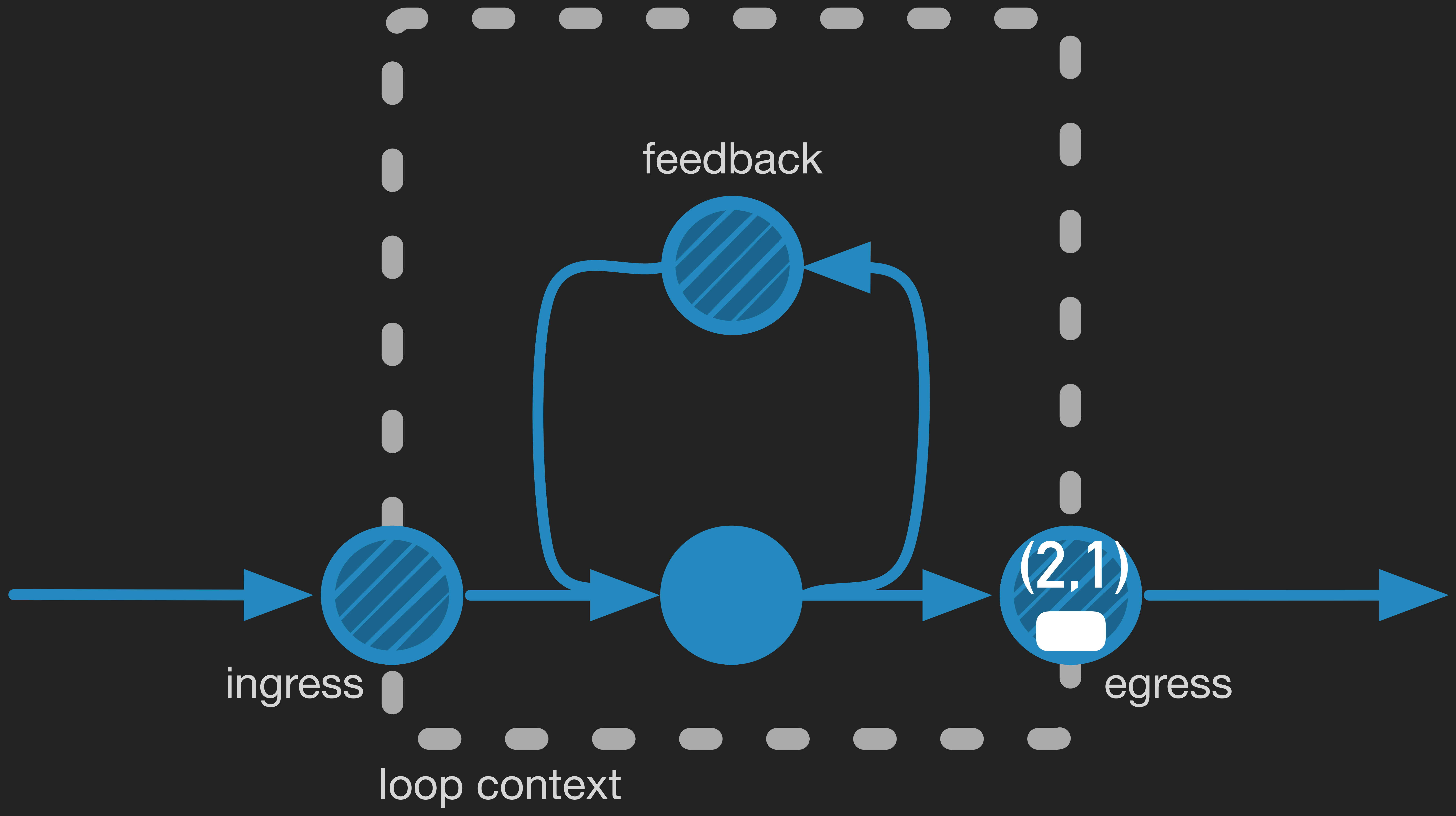
CYCLICAL DATAFLOW: LOOP CONTEXTS



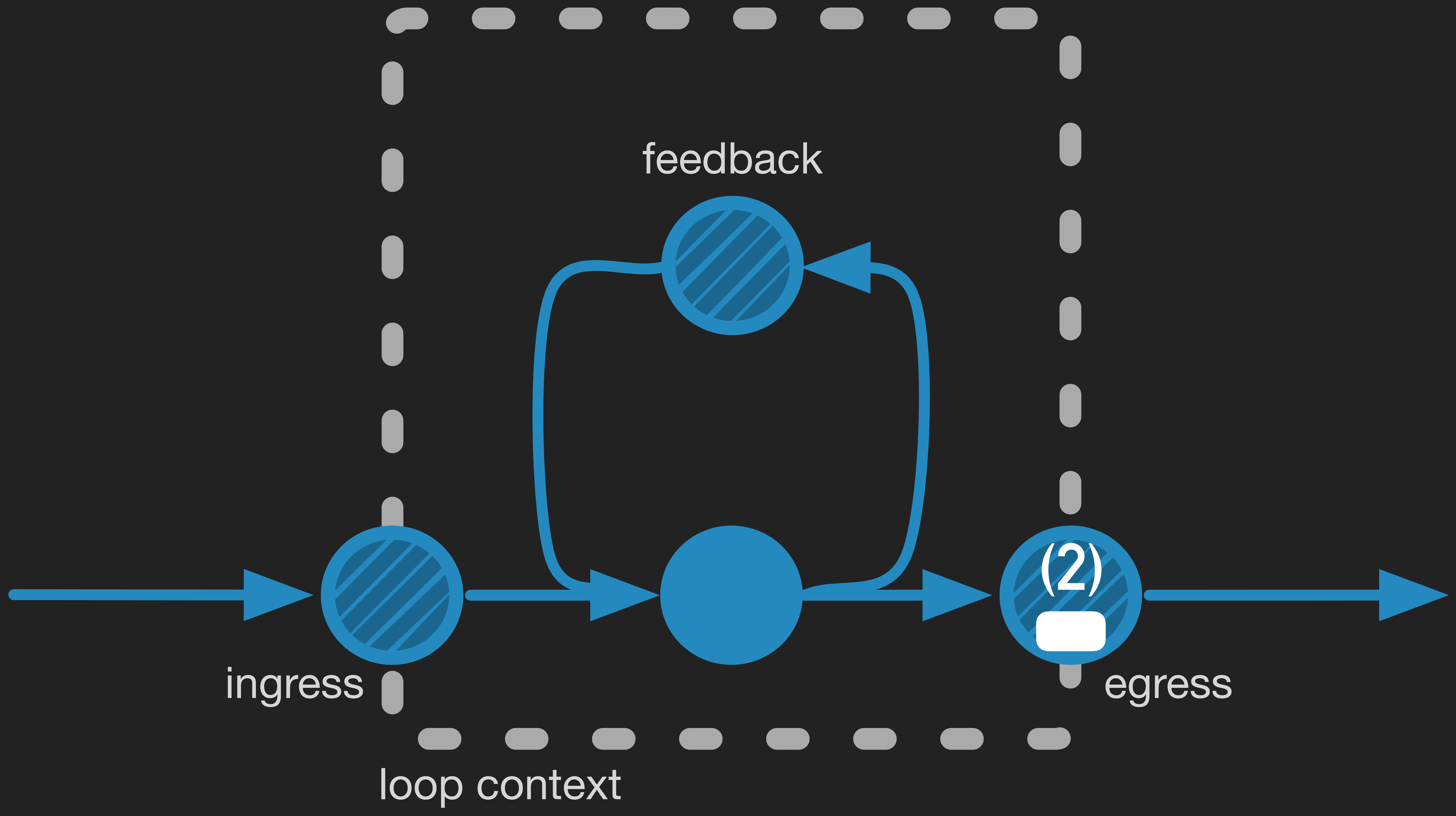
CYCLICAL DATAFLOW: LOOP CONTEXTS



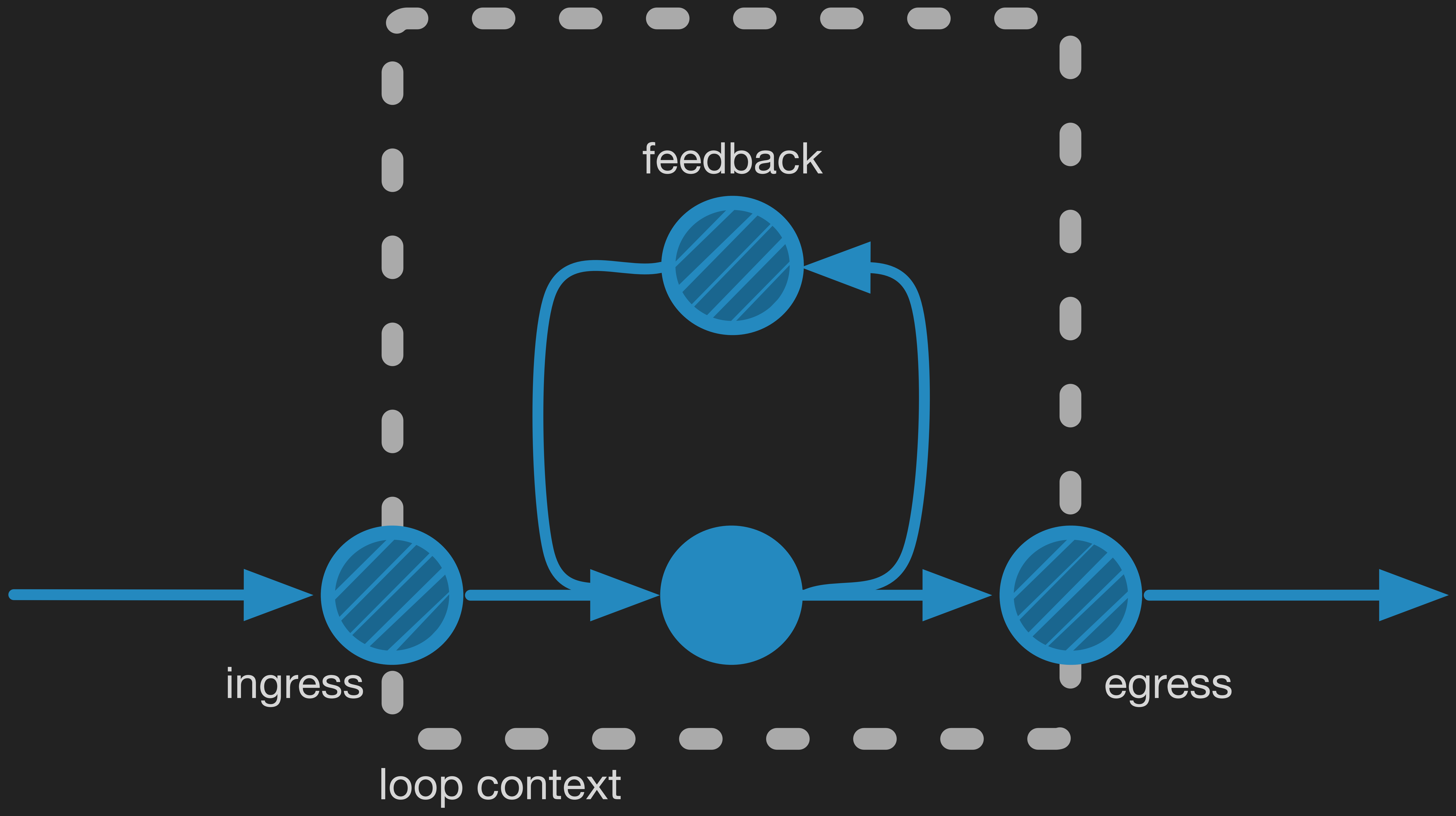
CYCLICAL DATAFLOW: LOOP CONTEXTS



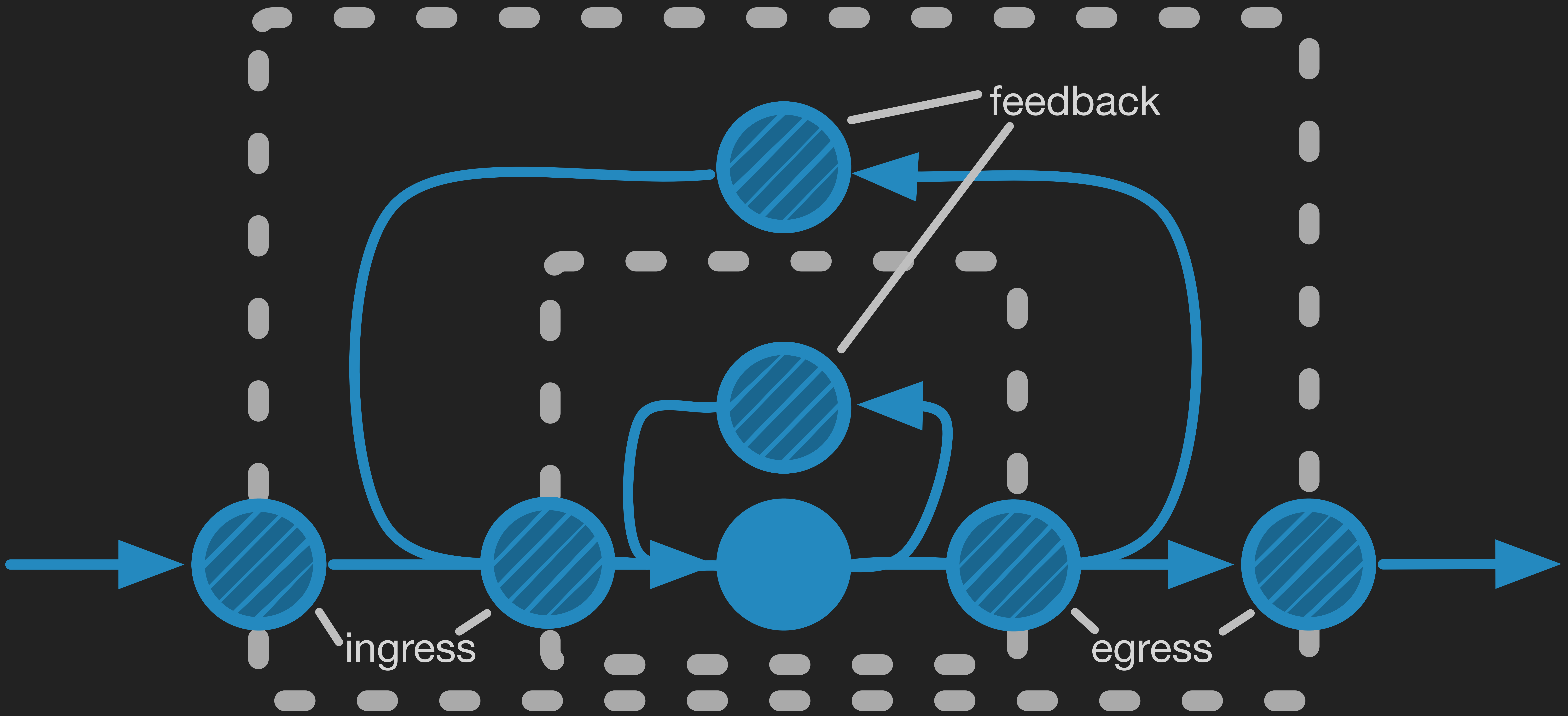
CYCLICAL DATAFLOW: LOOP CONTEXTS



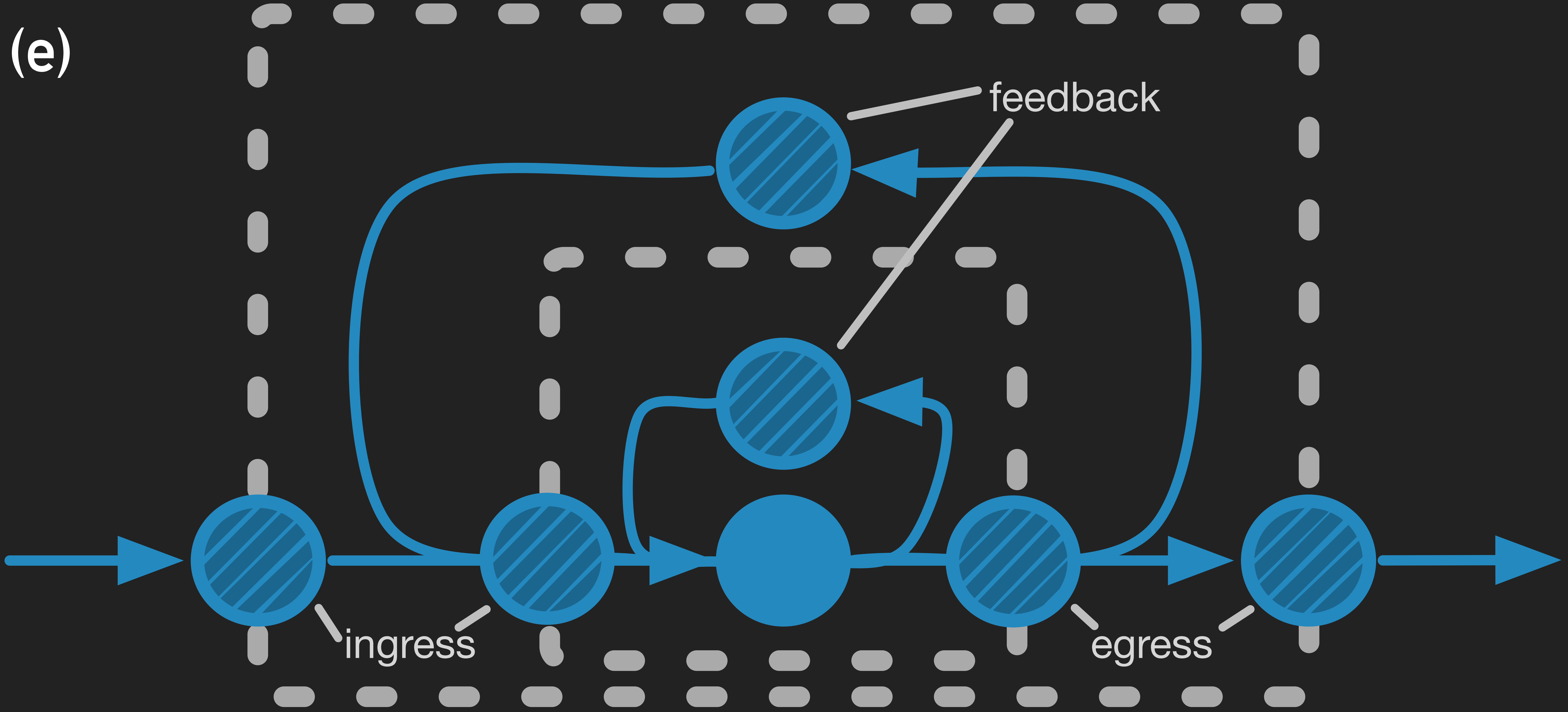
CYCLICAL DATAFLOW: LOOP CONTEXTS



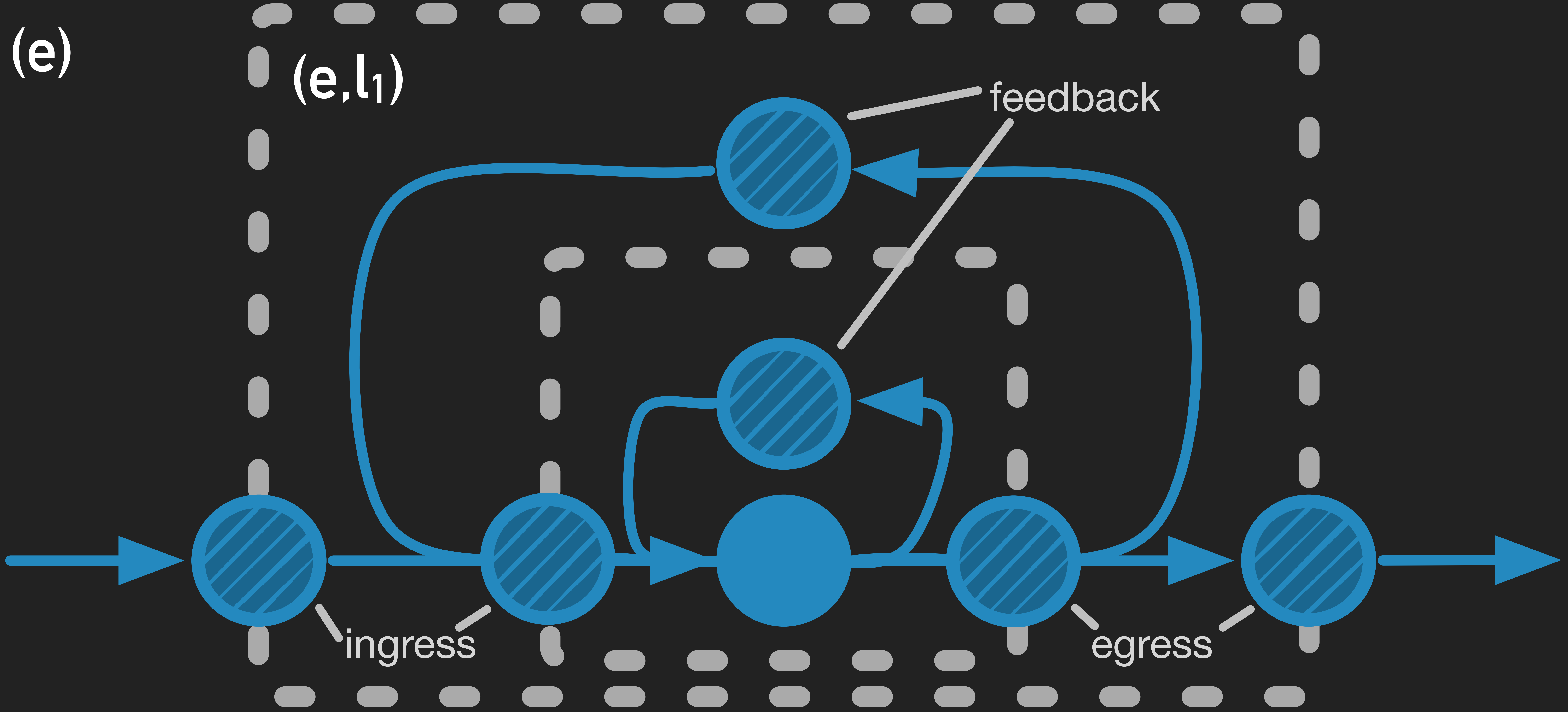
CYCLICAL DATAFLOW: LOOP CONTEXTS



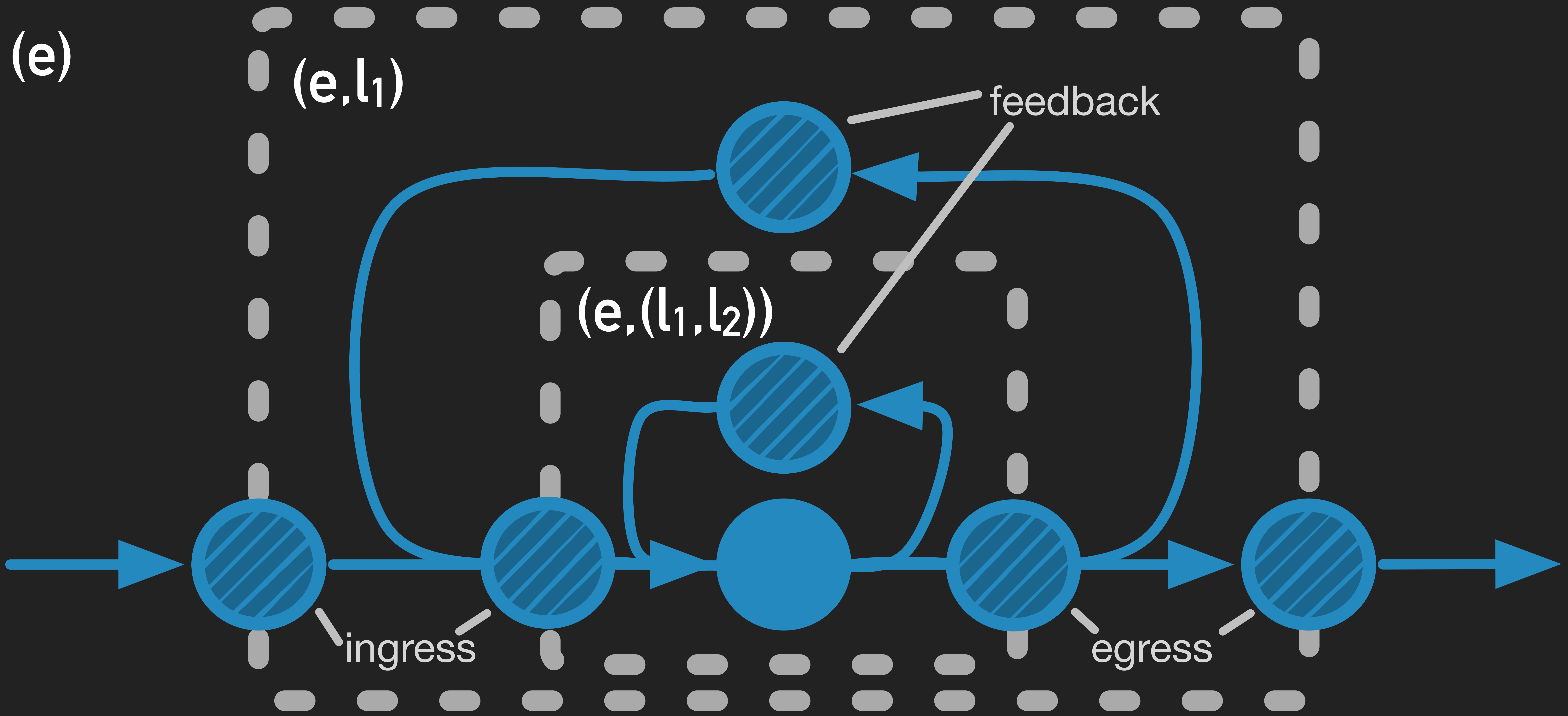
CYCLICAL DATAFLOW: LOOP CONTEXTS



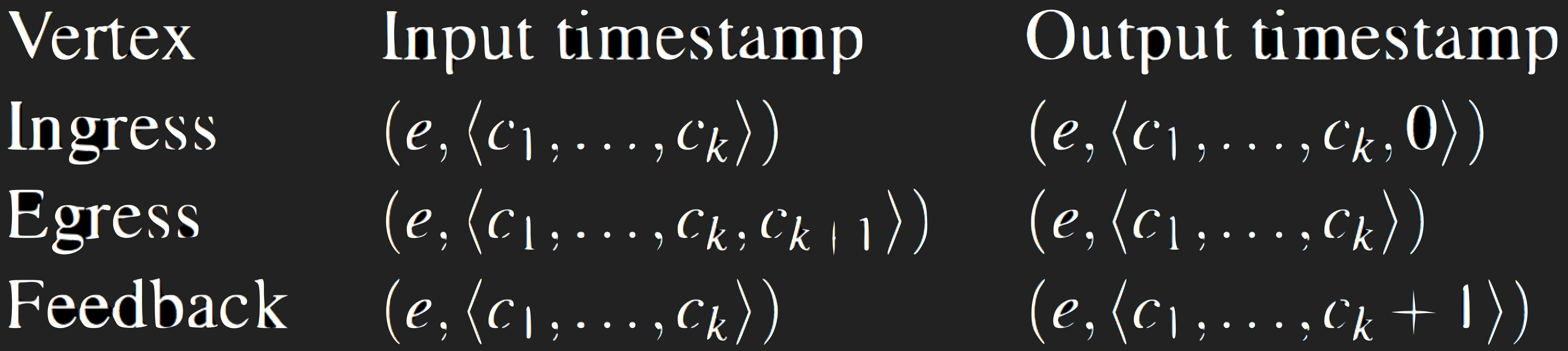
CYCLICAL DATAFLOW: LOOP CONTEXTS



CYCLICAL DATAFLOW: LOOP CONTEXTS



CYCLICAL DATAFLOW: TIMESTAMPS IN LOOP CONTEXTS

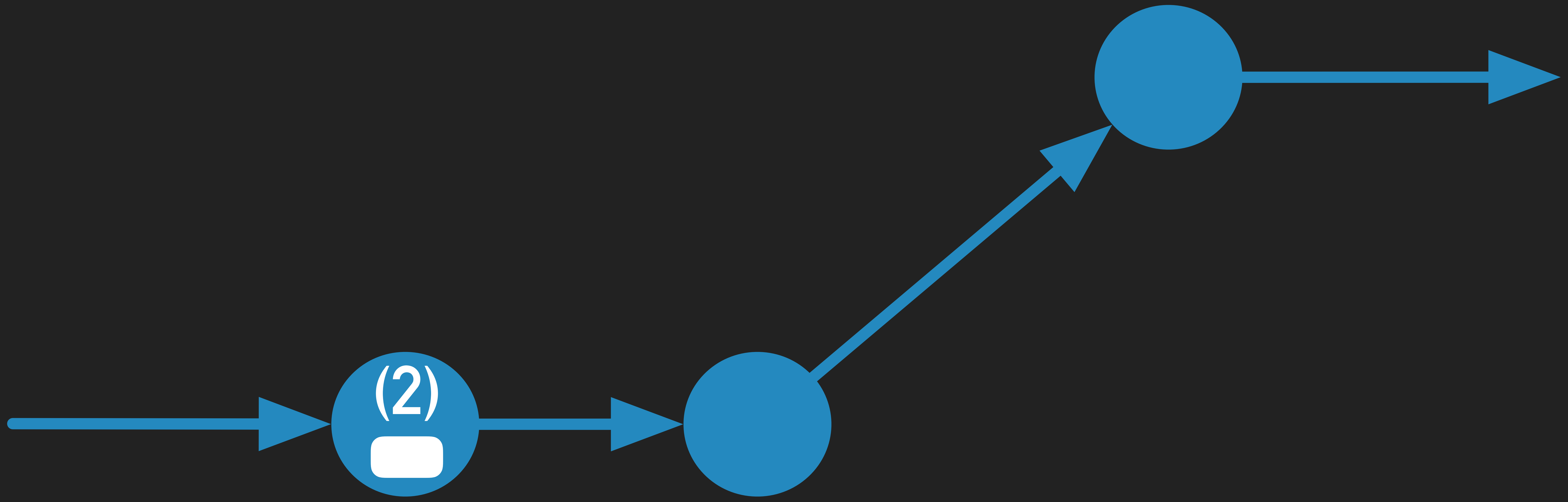


Vertex	Input timestamp	Output timestamp
Ingress	$(e, \langle c_1, \ldots, c_k \rangle)$	$(e, \langle c_1, \ldots, c_k, 0 \rangle)$
Egress	$(e, \langle c_1, \ldots, c_k, c_{k+1} \rangle)$	$(e, \langle c_1, \ldots, c_k \rangle)$
Feedback	$(e, \langle c_1, \ldots, c_k \rangle)$	$(e, \langle c_1, \ldots, c_k + 1 \rangle)$

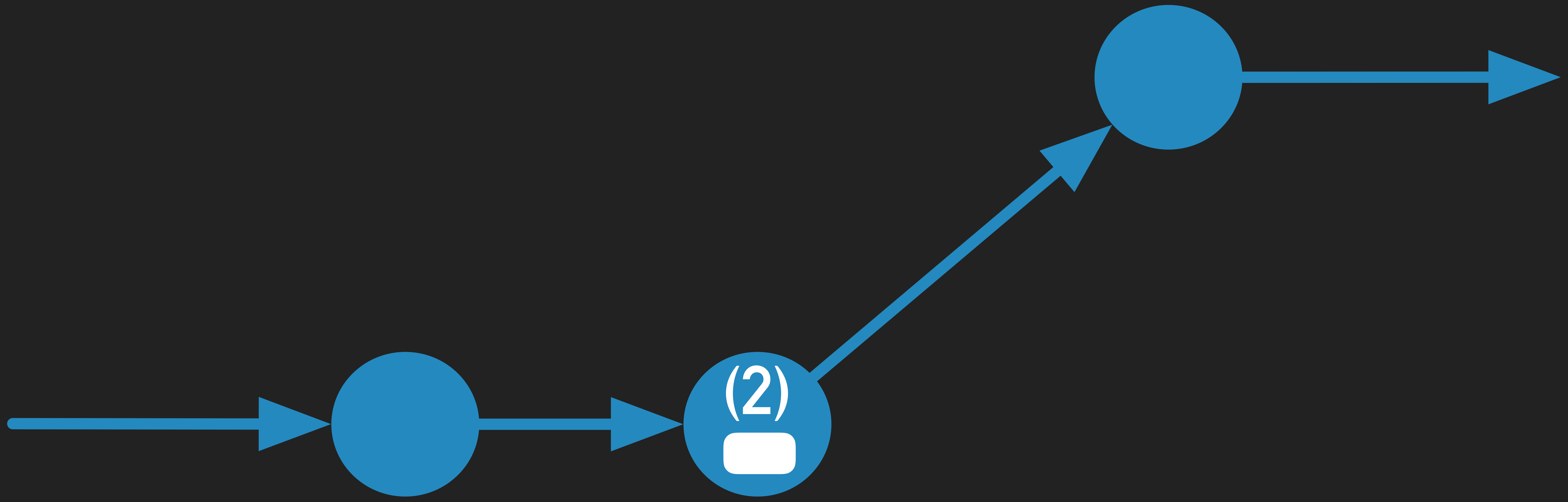
For two timestamps $t_1=(e_1,\mathbf{c}_1)$ and $t_2=(e_2,\mathbf{c}_2)$

$t_1 \leq t_2$ iff. $e_1 \leq e_2$ and $\mathbf{c}_1 \leq \mathbf{c}_2$

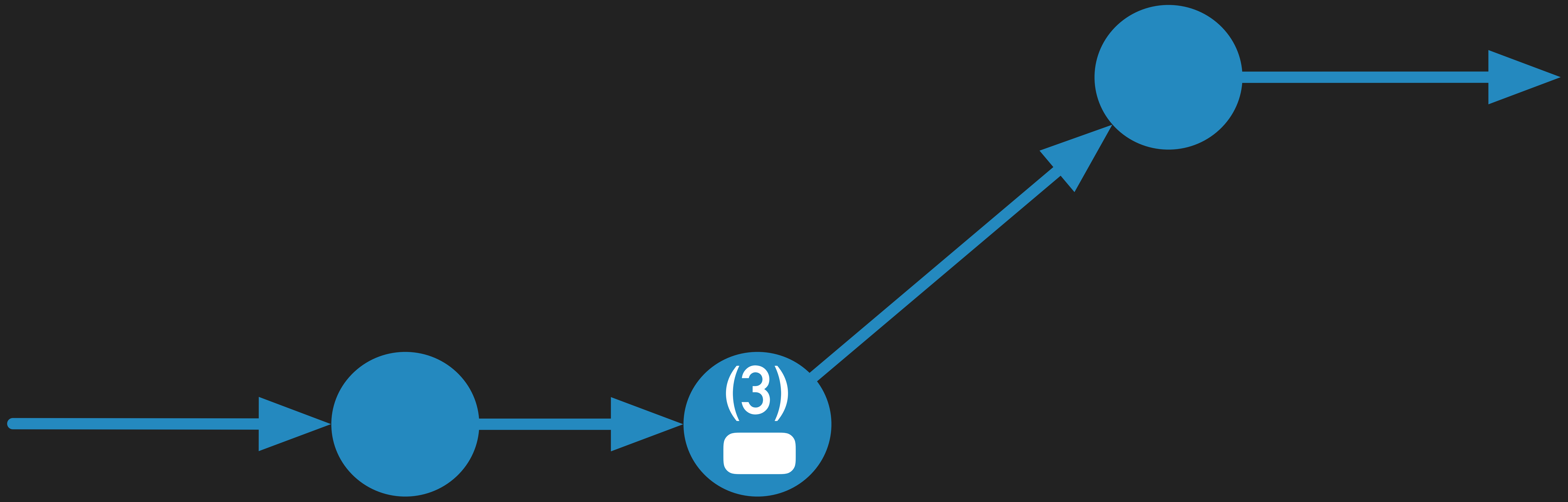
OPERATOR CONTRACT



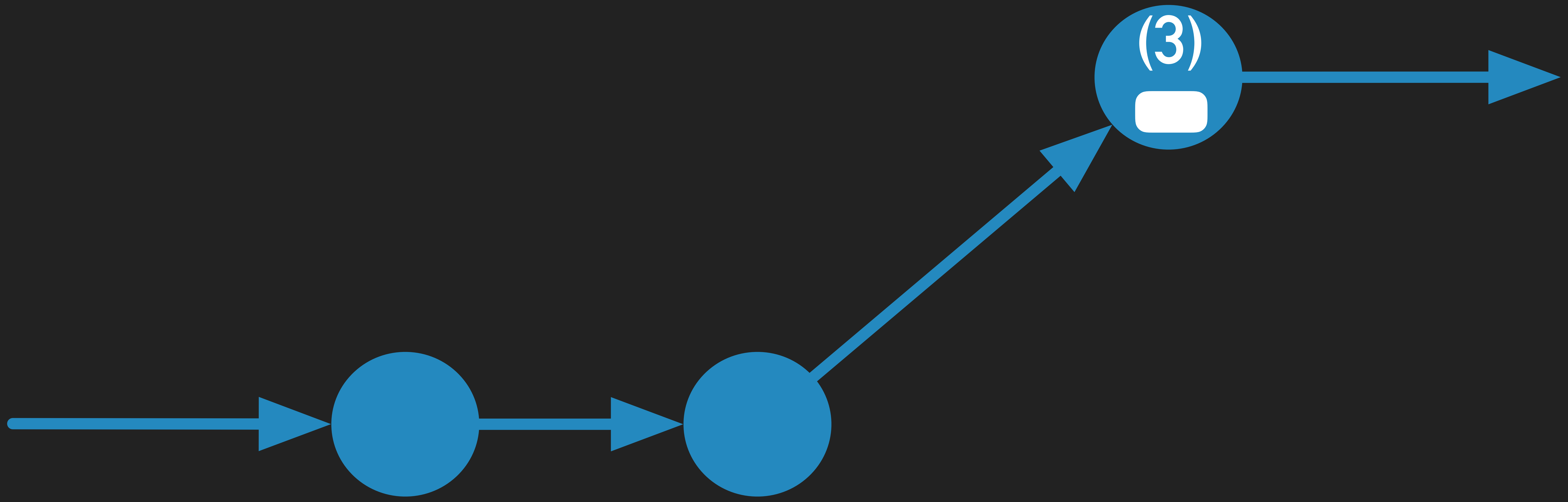
OPERATOR CONTRACT



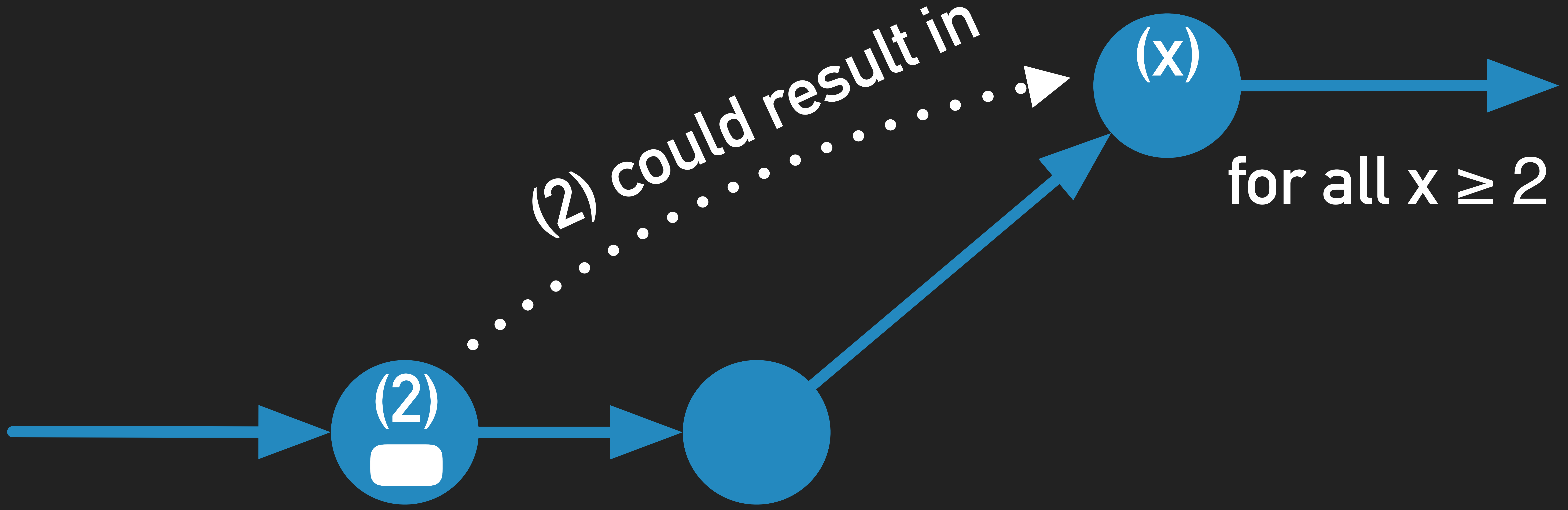
OPERATOR CONTRACT



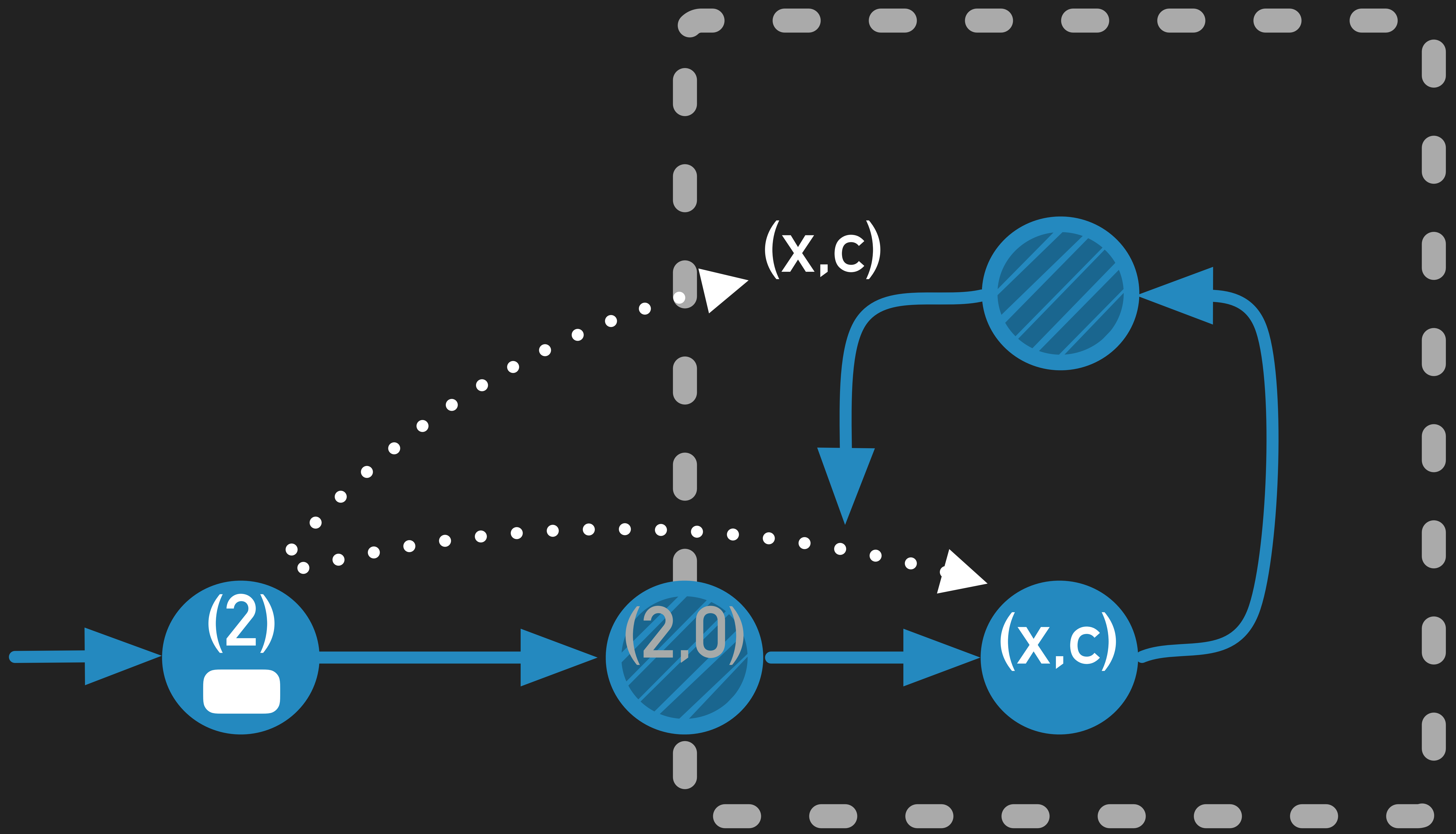
OPERATOR CONTRACT



COULD-RESULT-IN



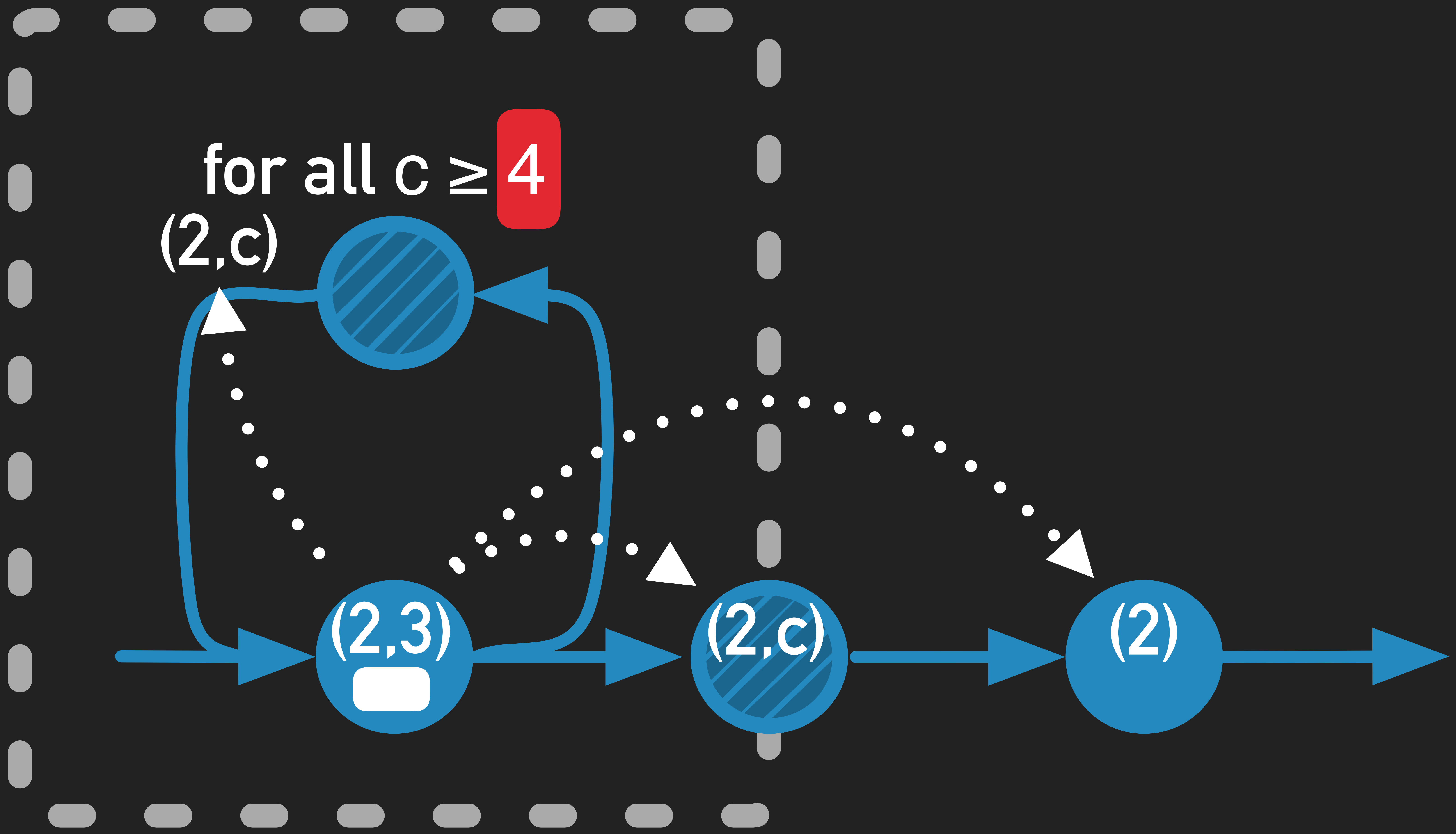
COULD-RESULT-IN



for all x ≥ 2, c ≥ 0

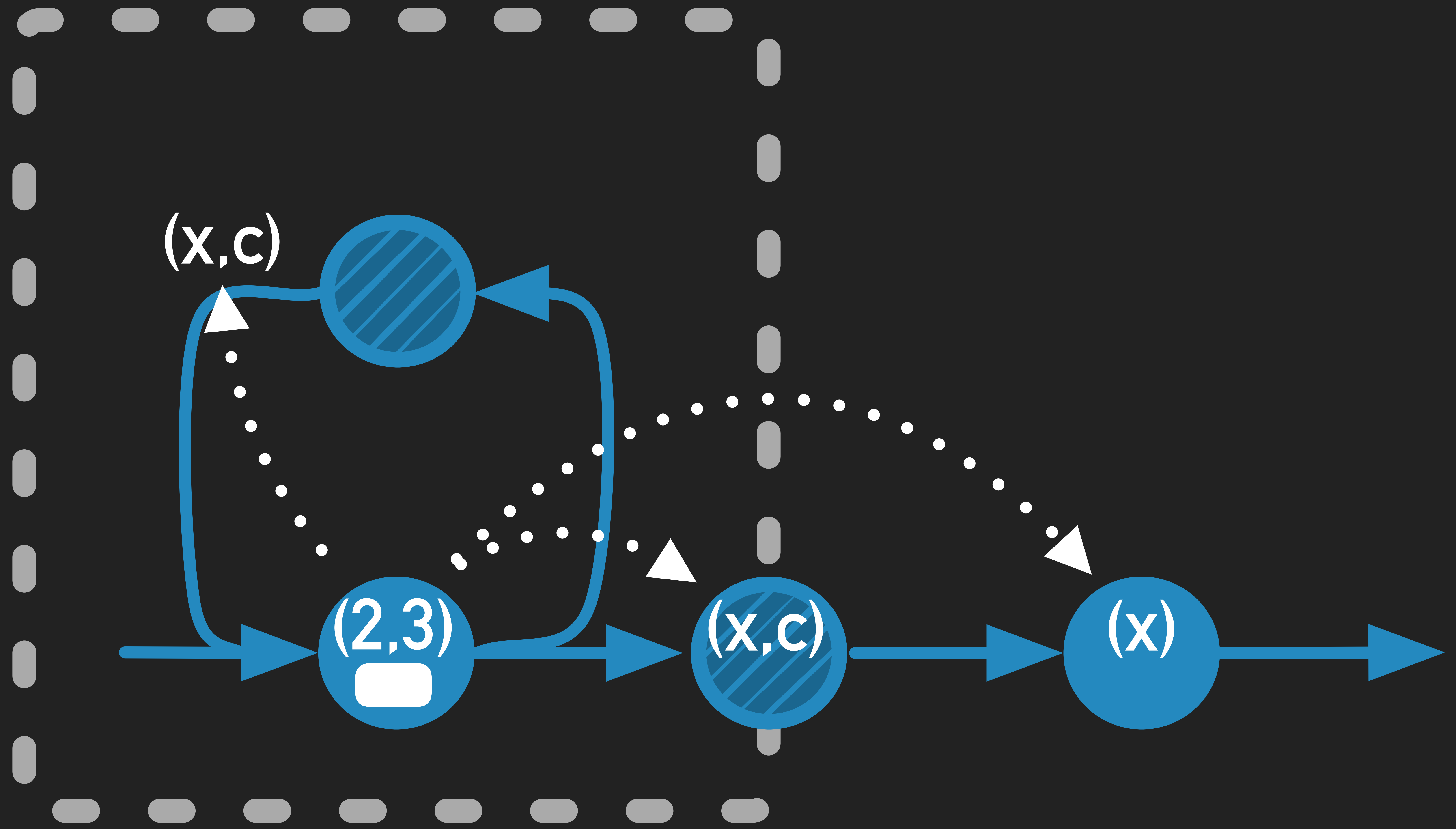
COULD-RESULT-IN

(e = 2, c)



COULD-RESULT-IN

for all c, x so that c ≥ 4 and x ≥ 2



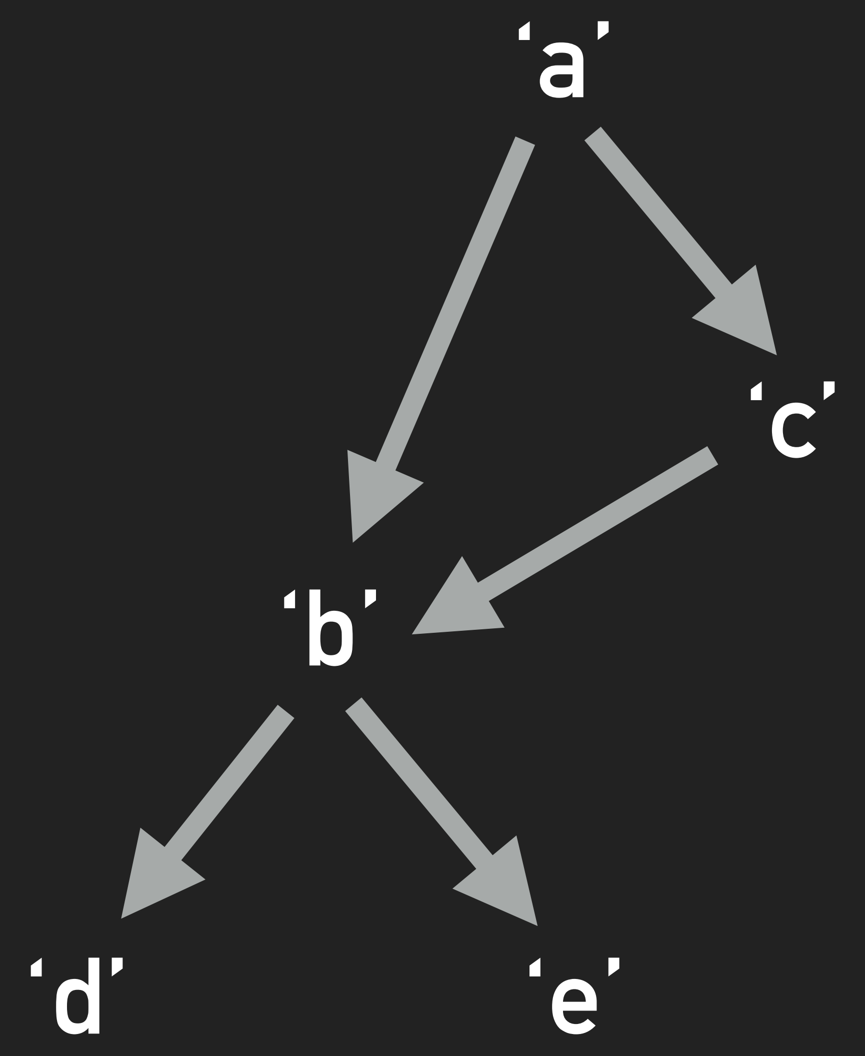
NOT IN THIS TALK

- details about the distributed implementation,
- progress tracking optimization,
- how to prevent micro-stragglers,
- frameworks on top of Naiad / Timely Dataflow,
- performance evaluation.

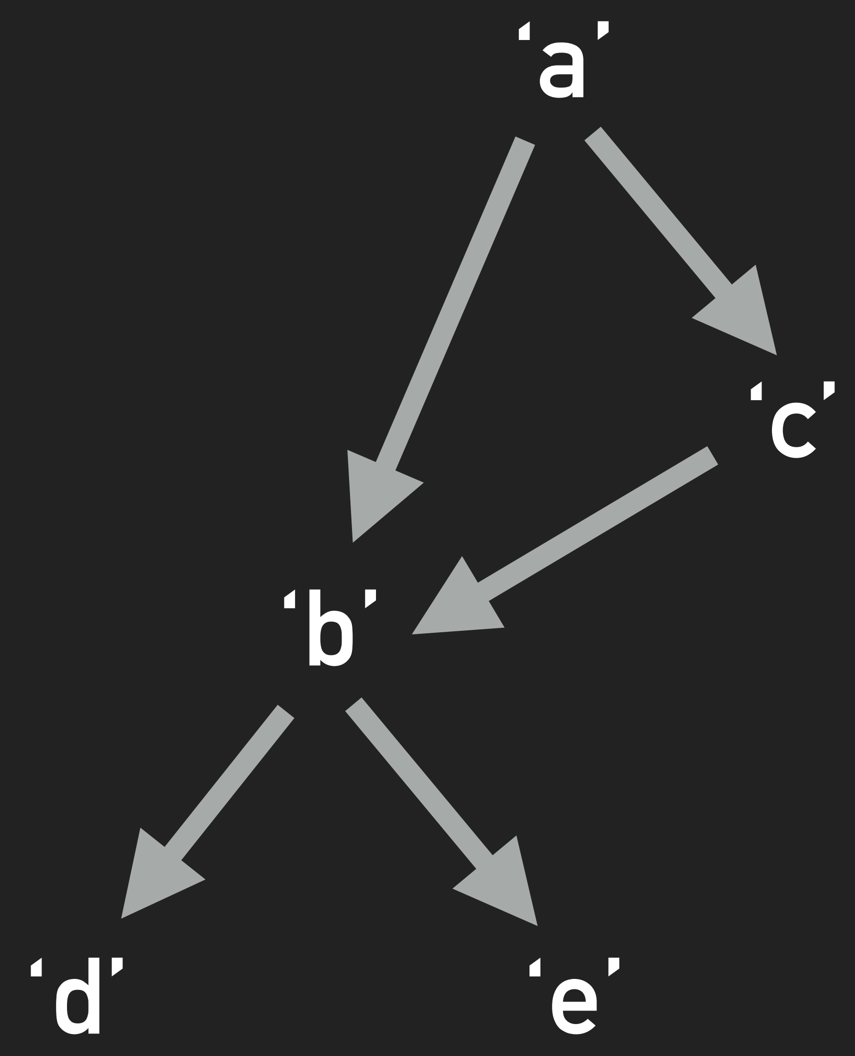
github.com/frankmcsherry/timely-dataflow

TIMELY DATAFLOW IN RUST (❤️).

ITERATIVE GRAPH COMPUTATION IN DATAFLOW.

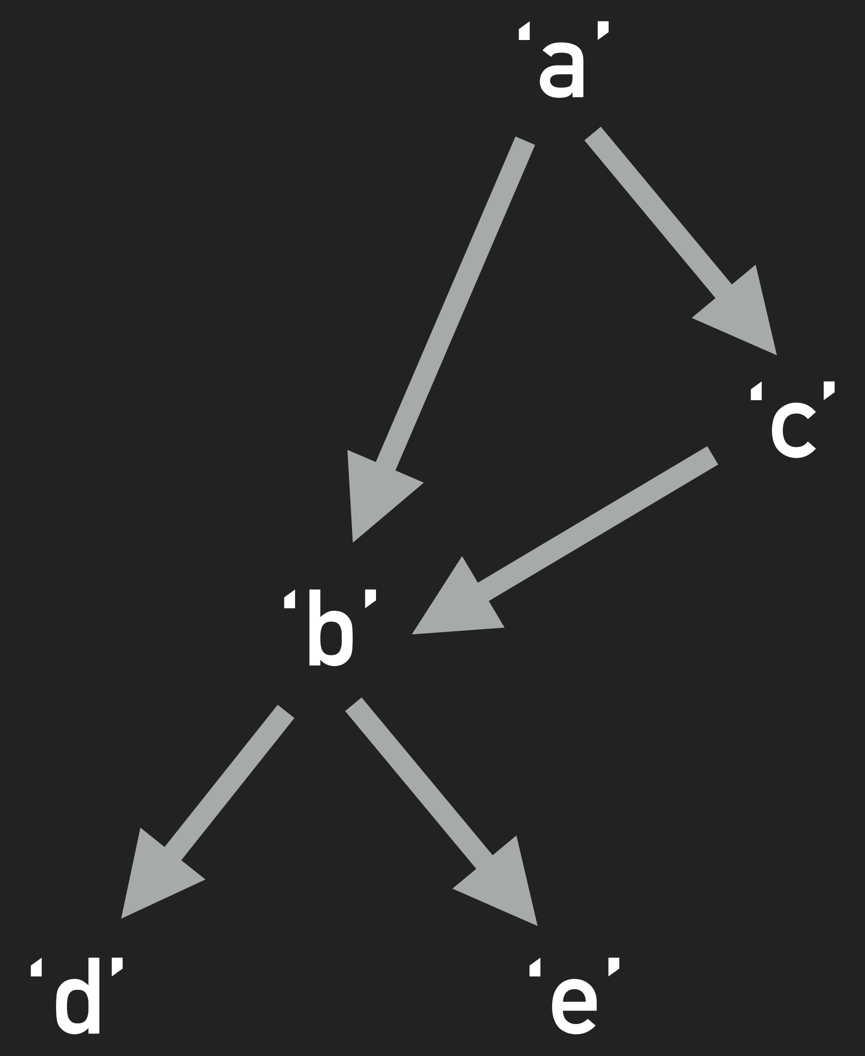


ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)

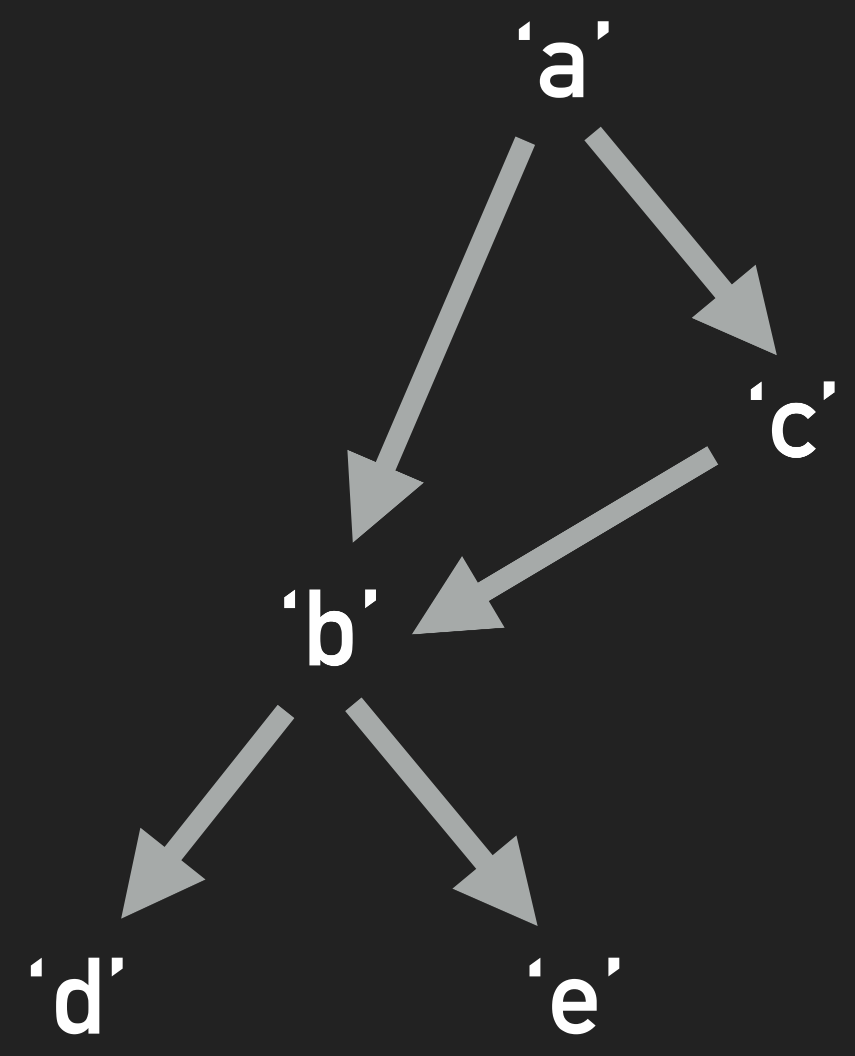
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)

(‘b’, ‘d’)

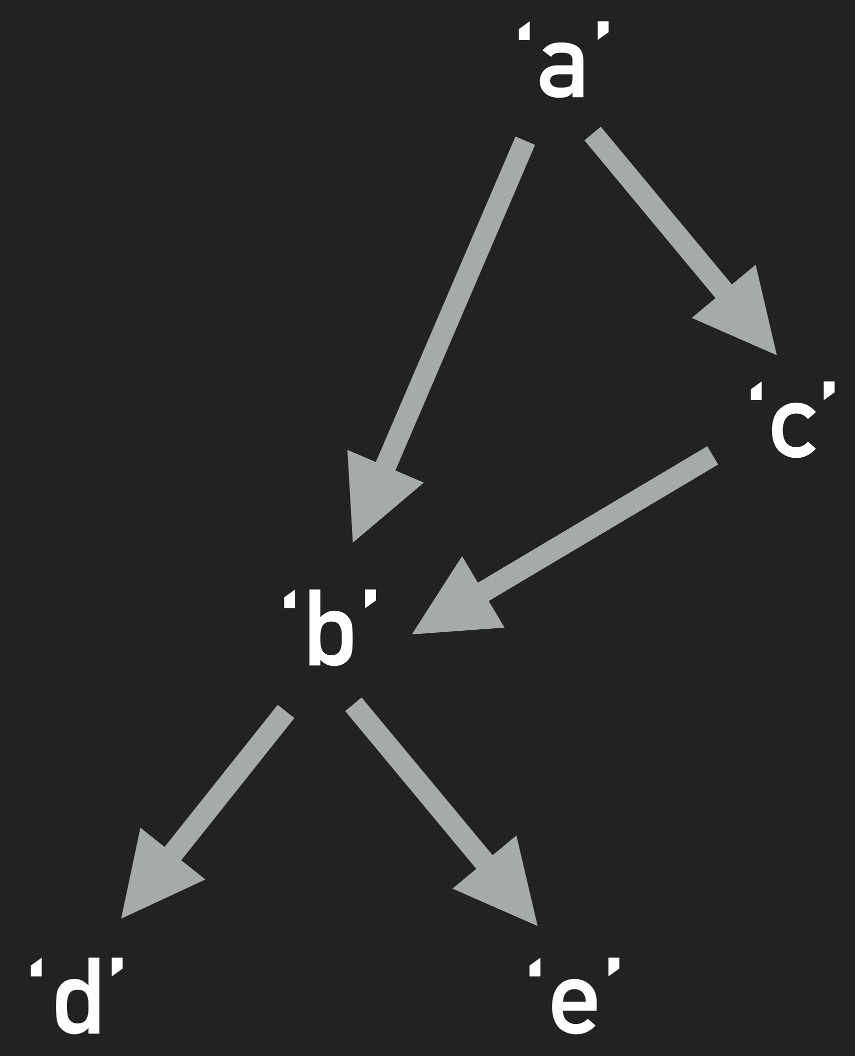
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)

(‘b’, ‘d’)
(‘b’, ‘e’)

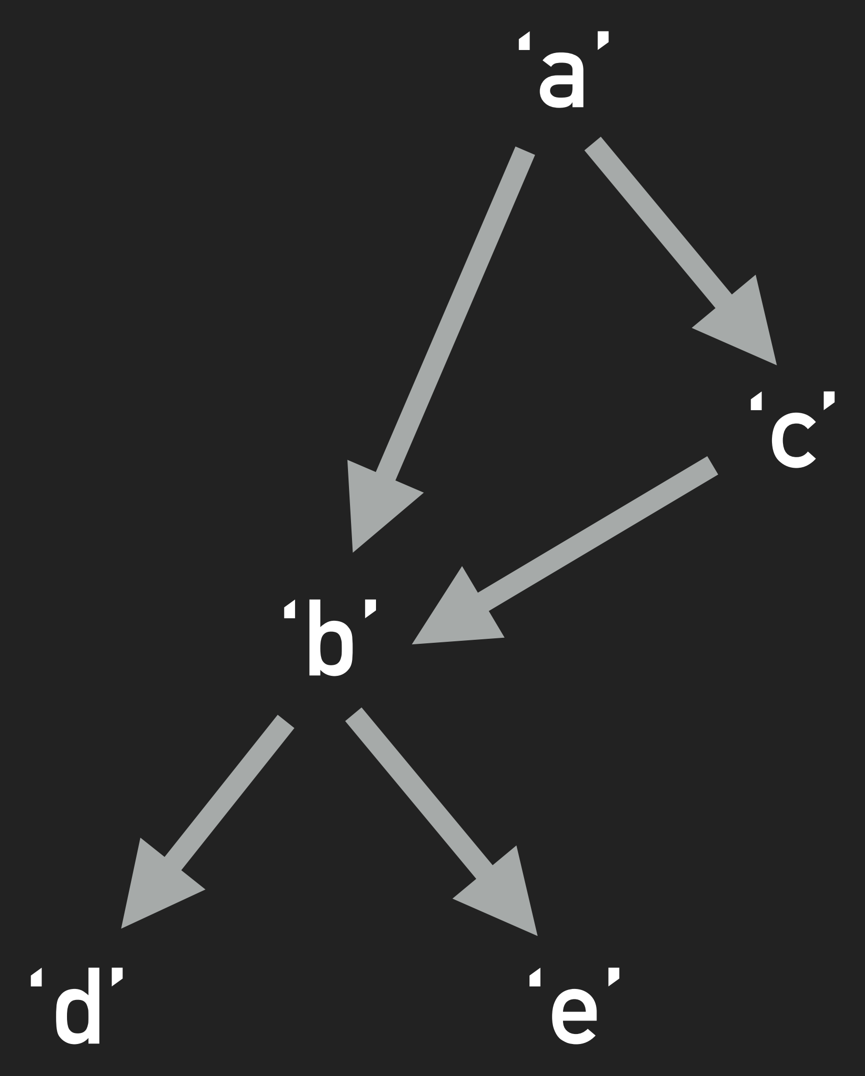
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)   (‘a’, ‘c’)

(‘b’, ‘d’)
(‘b’, ‘e’)

ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)

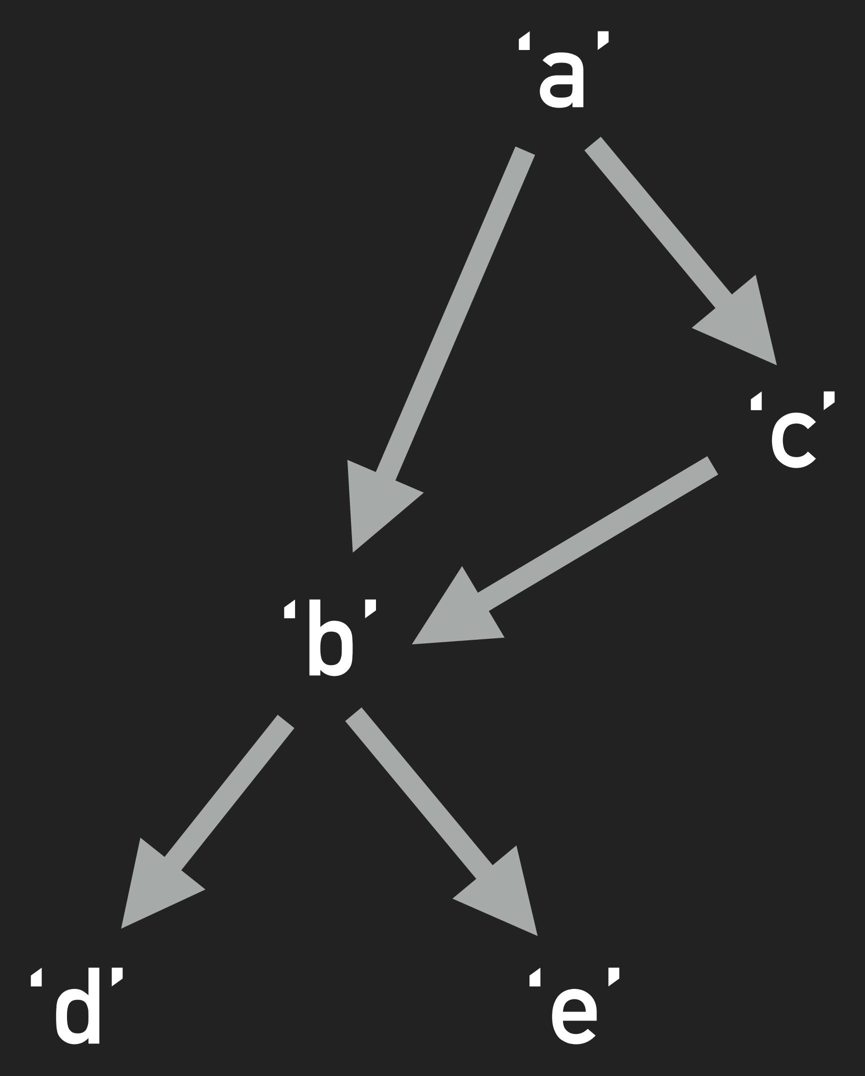
(‘a’, ‘c’)

(‘c’, ‘b’)

(‘b’, ‘d’)

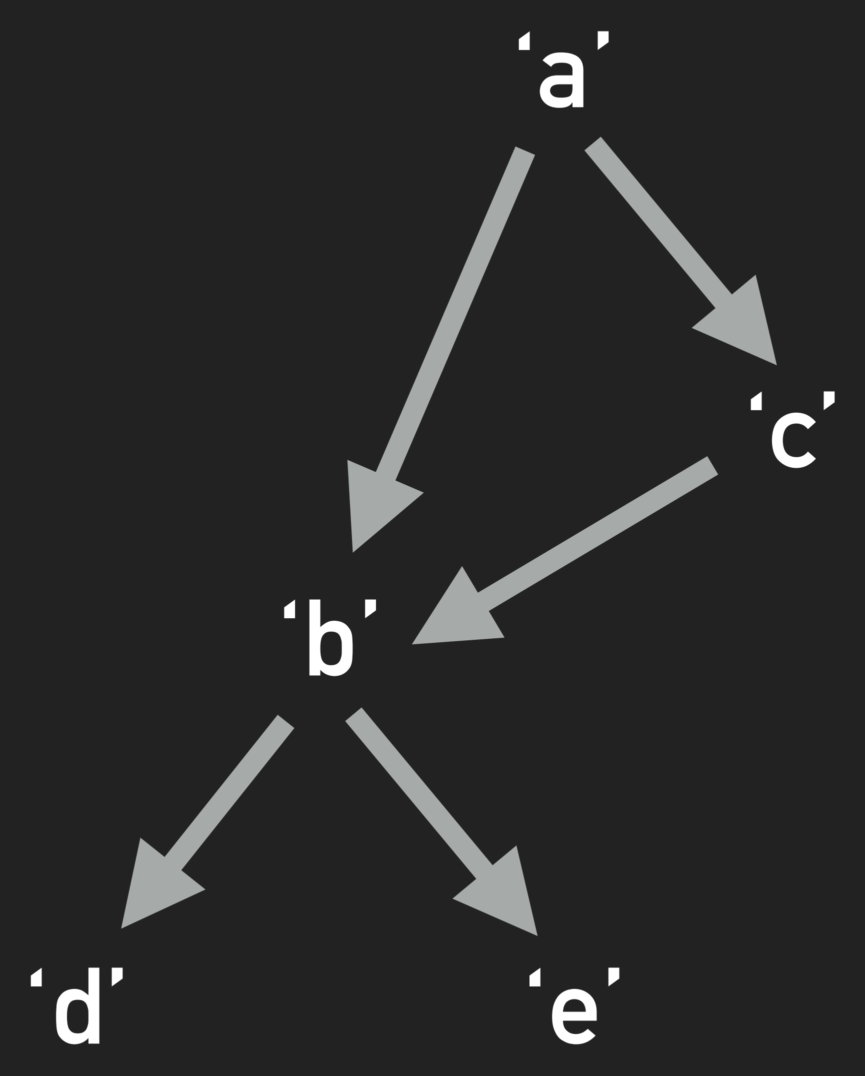
(‘b’, ‘e’)

ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



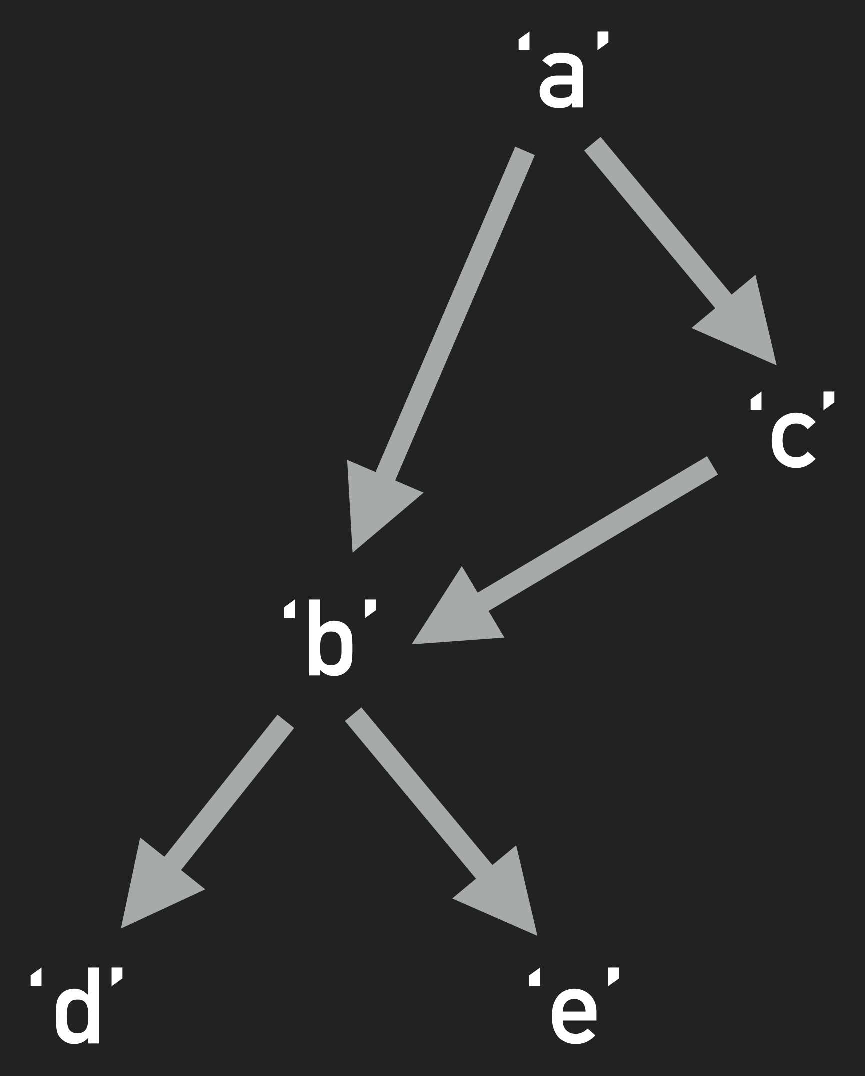
('a', 'b')	('a', 'c')
	('c', 'b')
('b', 'd')	('b', 'd')
('b', 'e')	

ITERATIVE GRAPH COMPUTATION IN DATAFLOW.

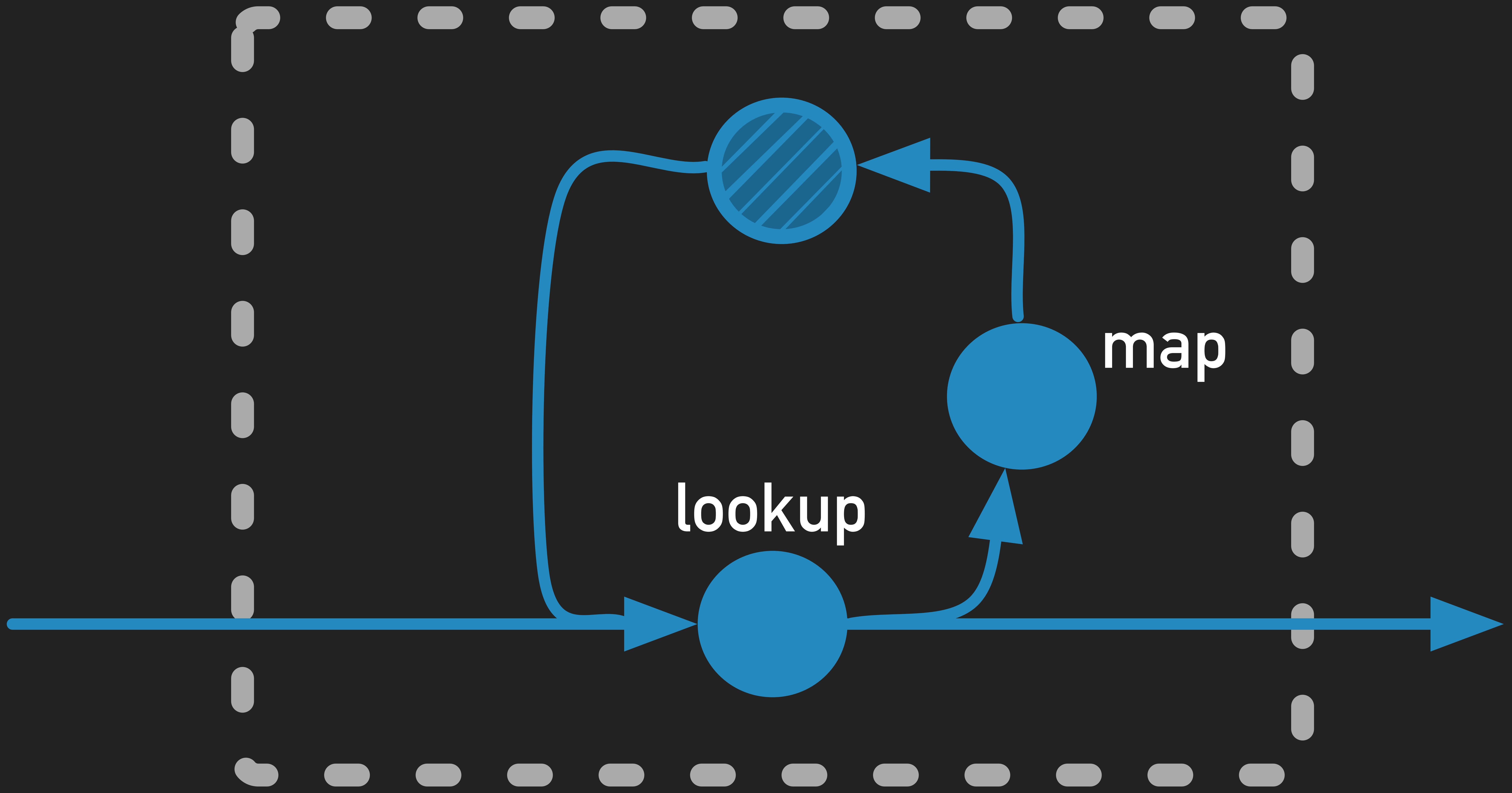


(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)

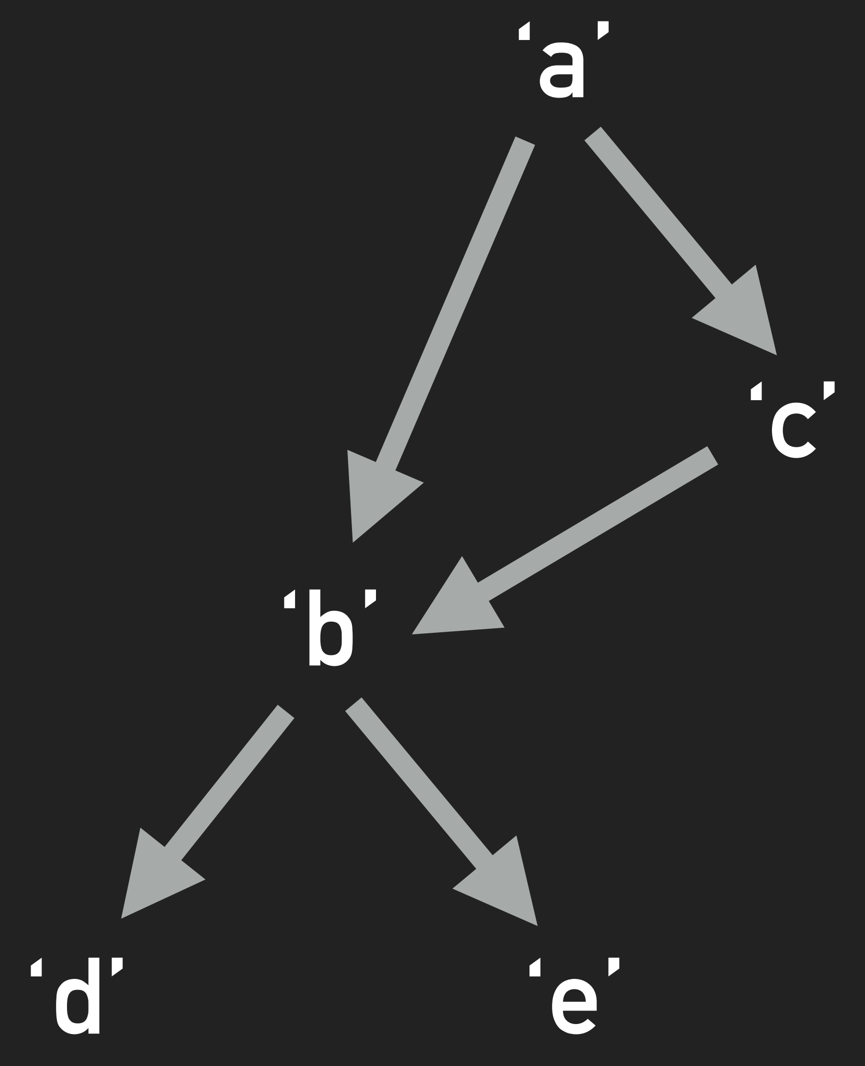
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



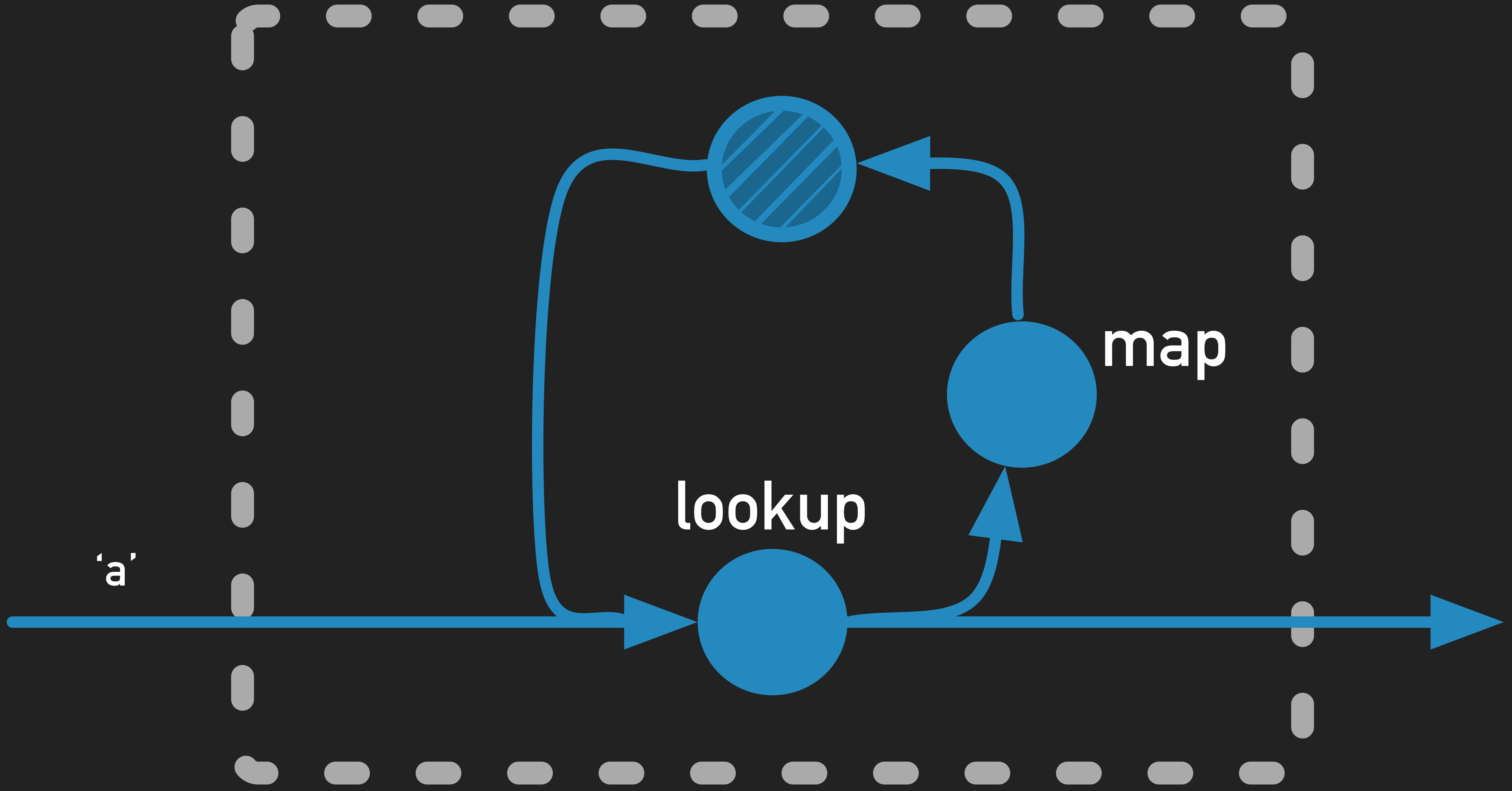
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



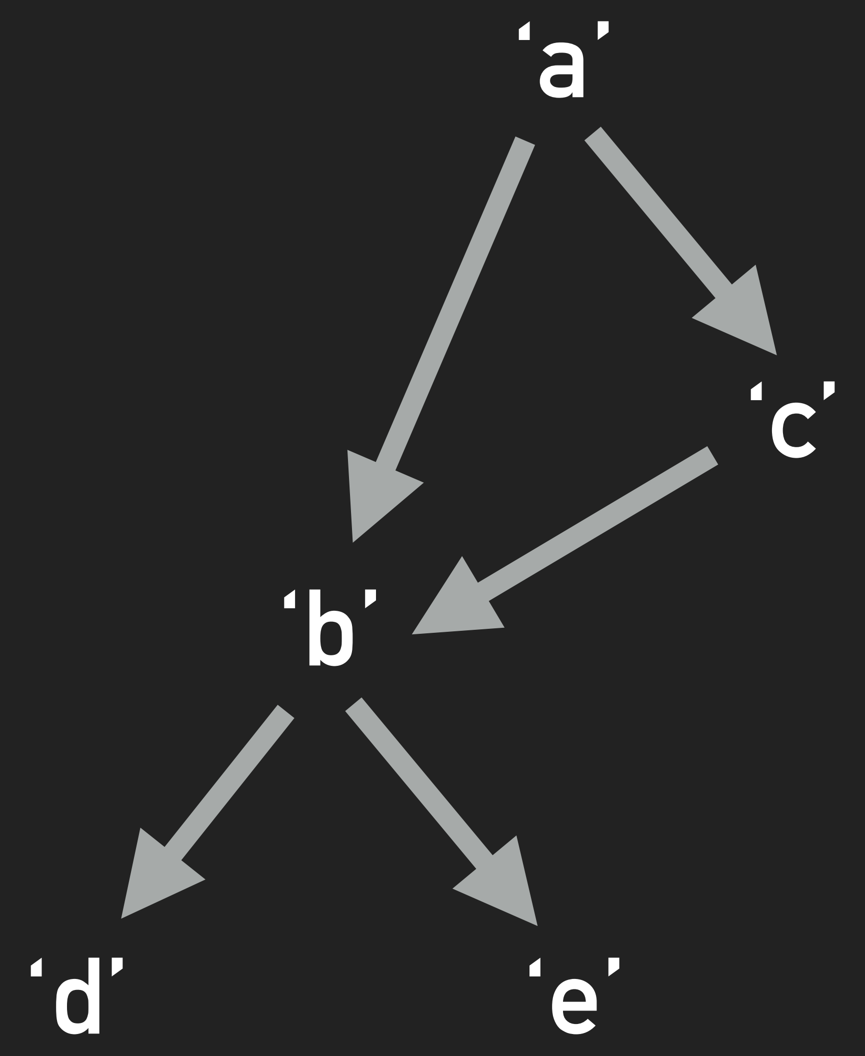
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



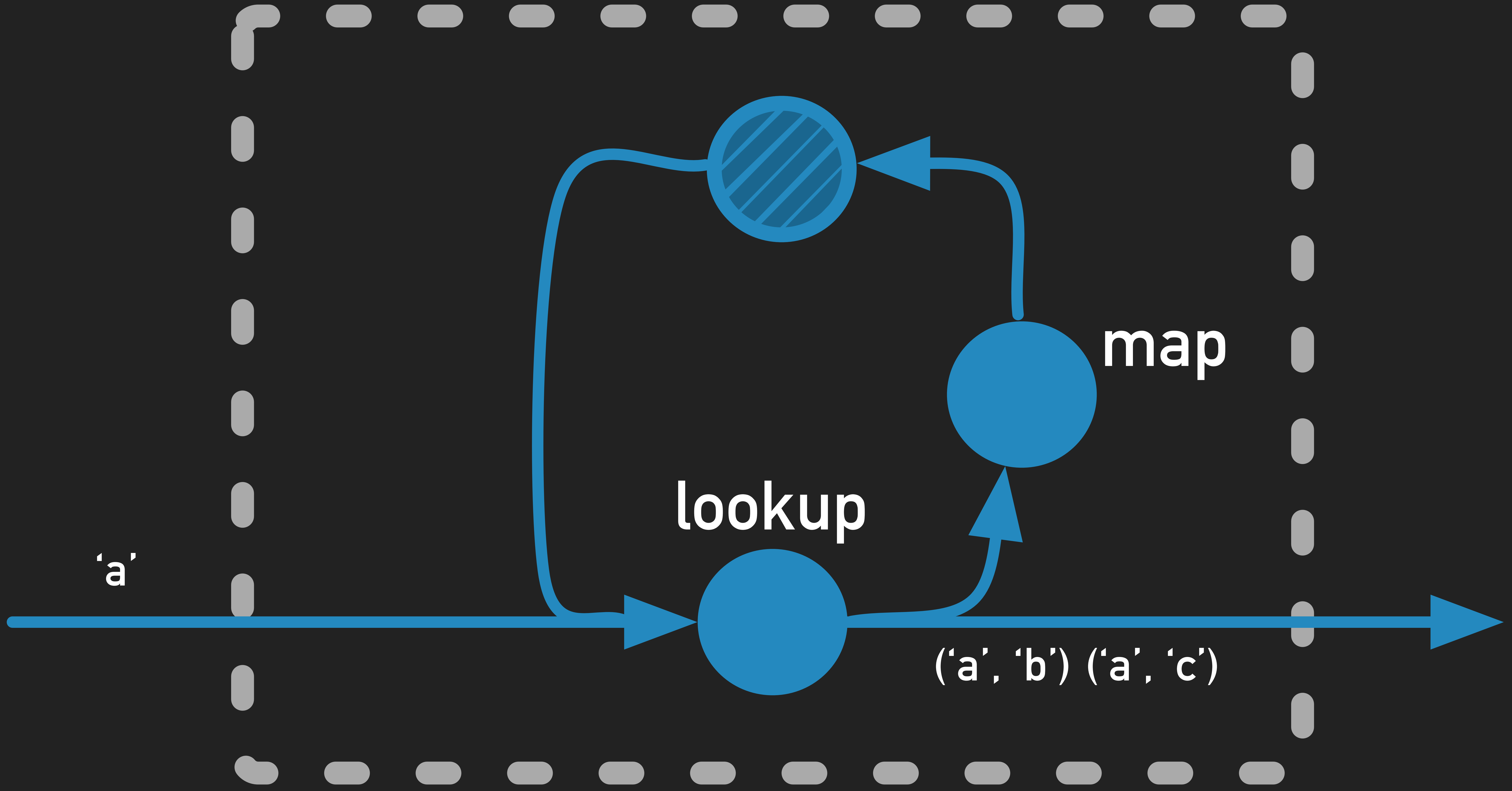
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



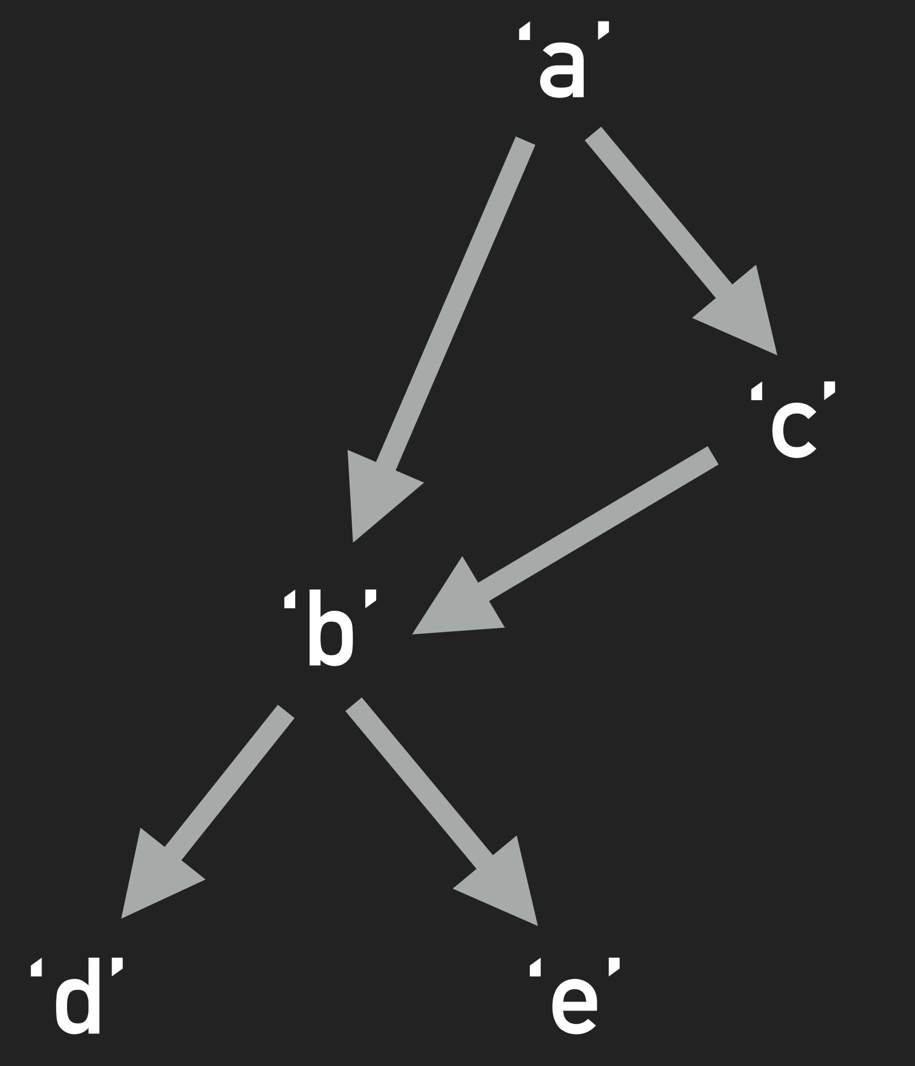
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



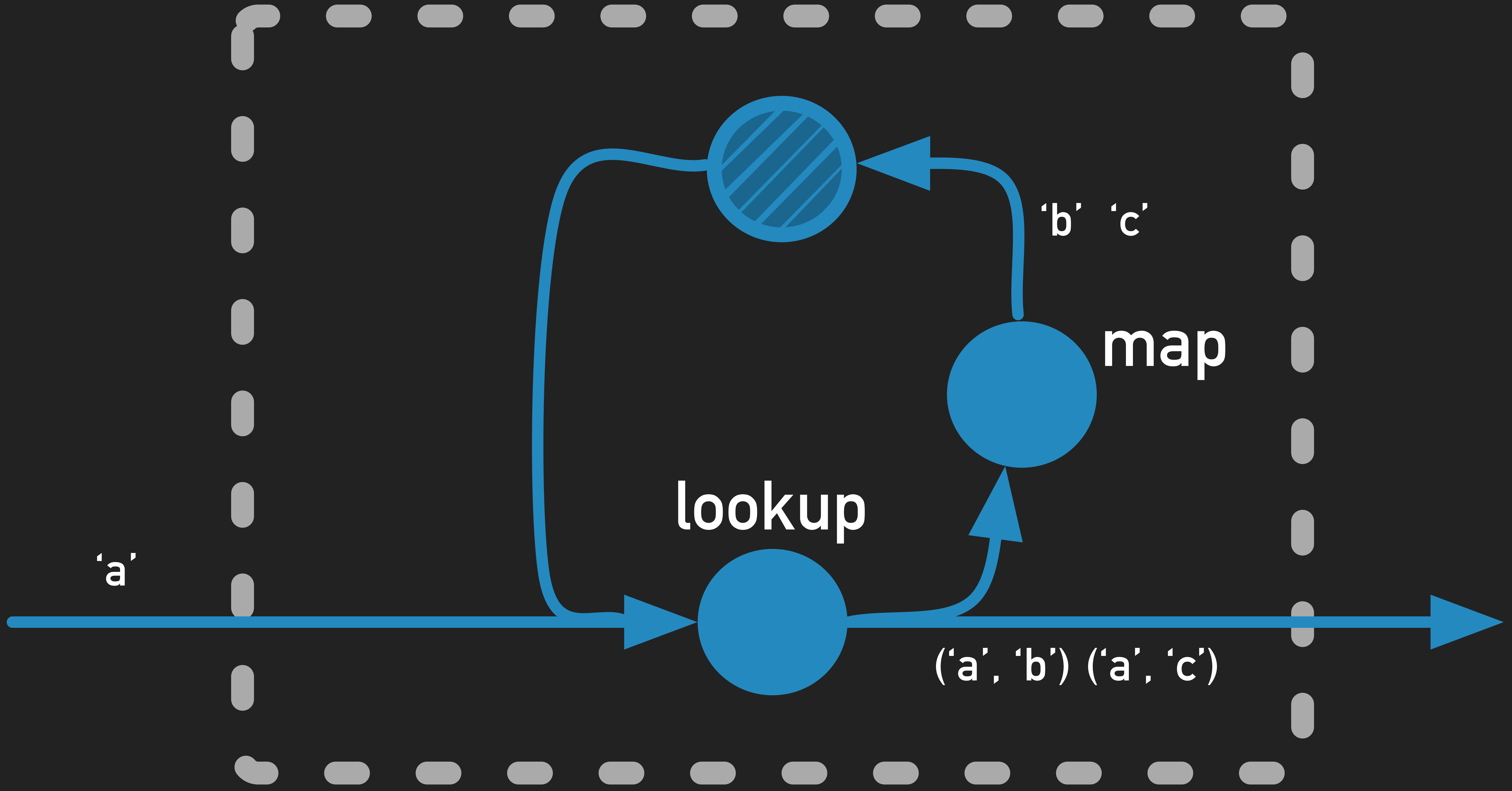
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



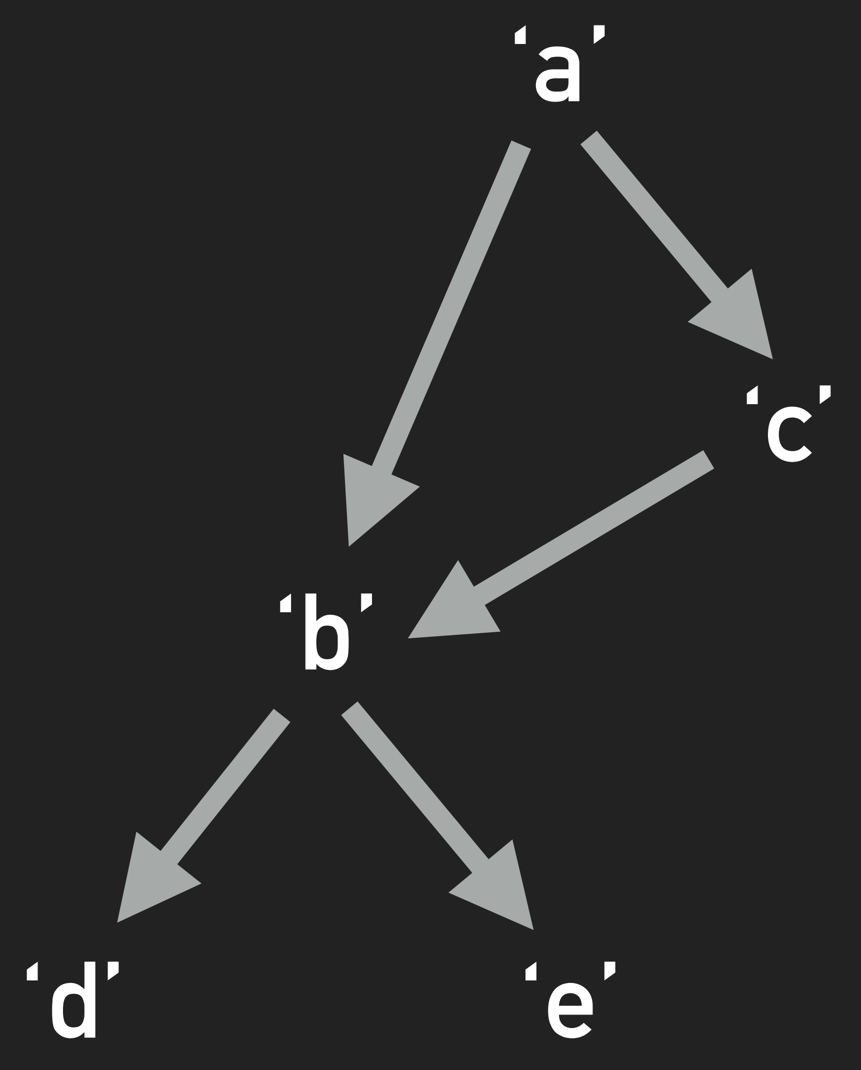
(‘a’, ‘b’)

(‘b’, ‘d’)
(‘b’, ‘e’)

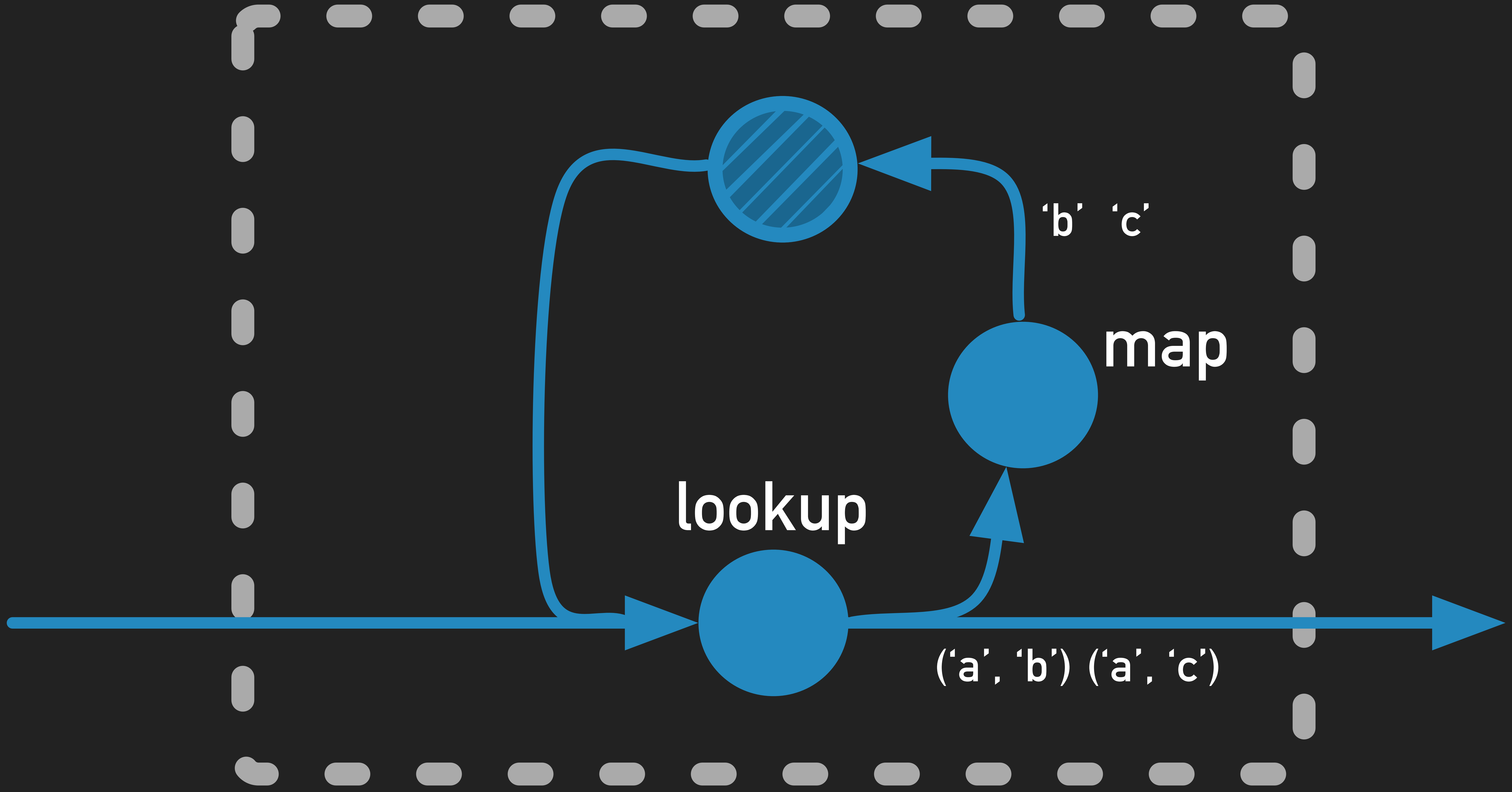
(‘a’, ‘c’)
(‘c’, ‘b’)
(‘b’, ‘d’)
(‘b’, ‘e’)



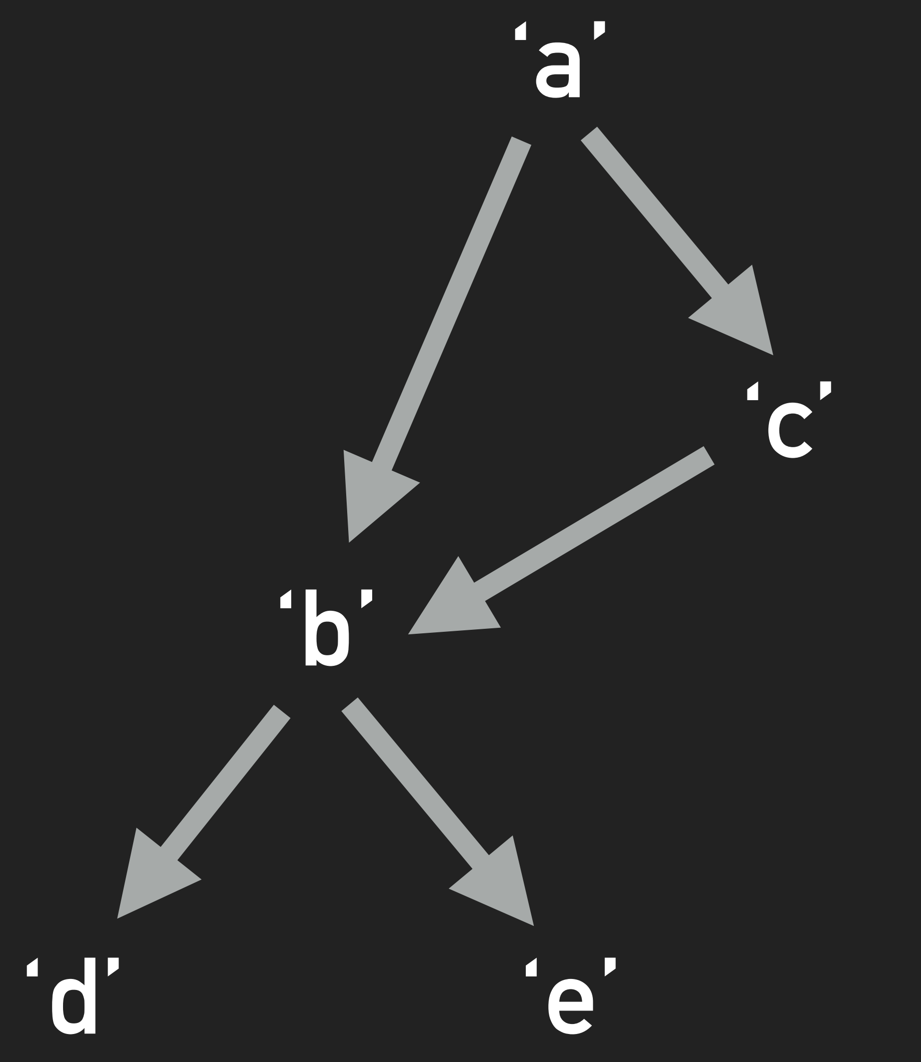
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



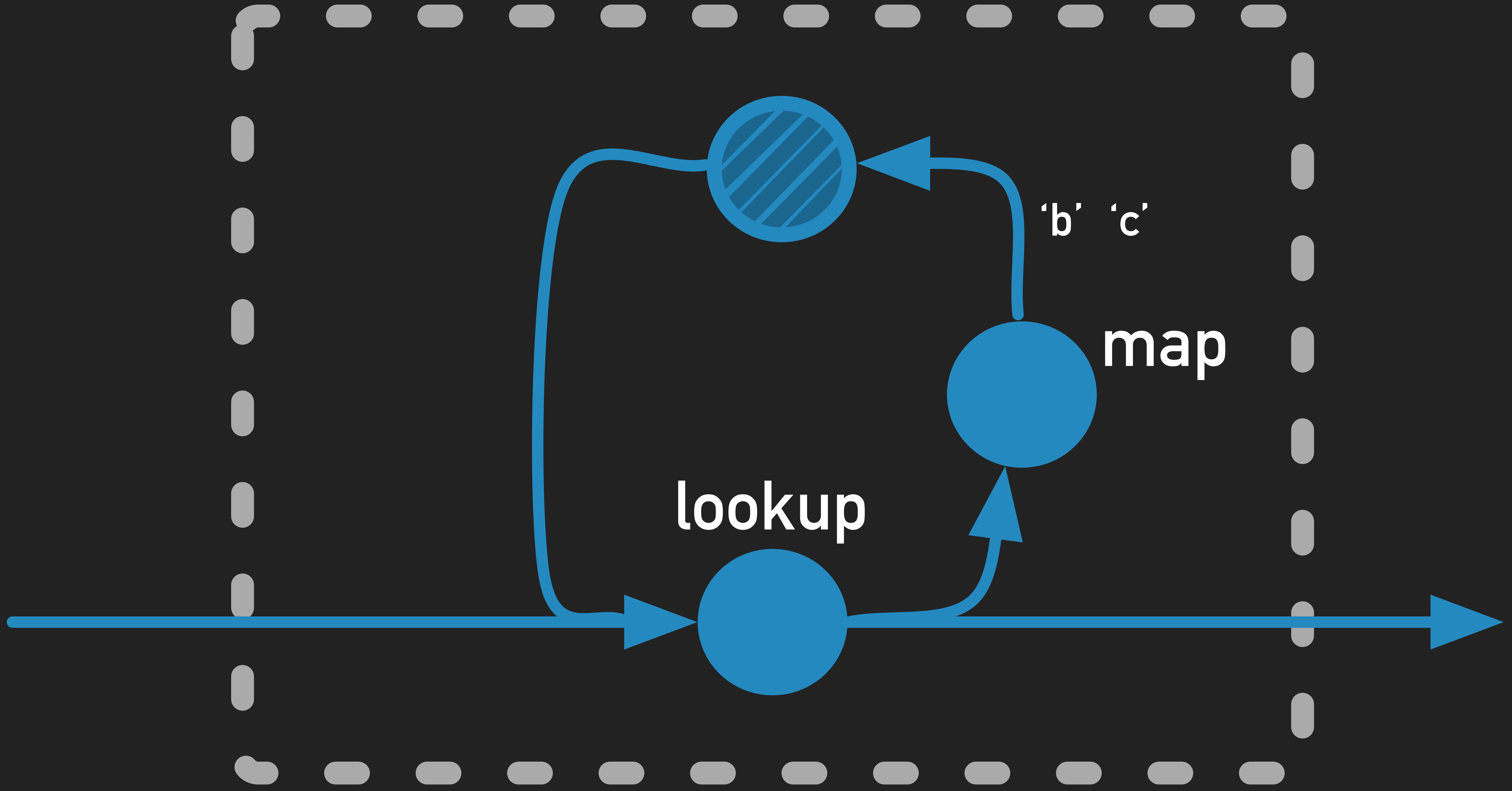
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



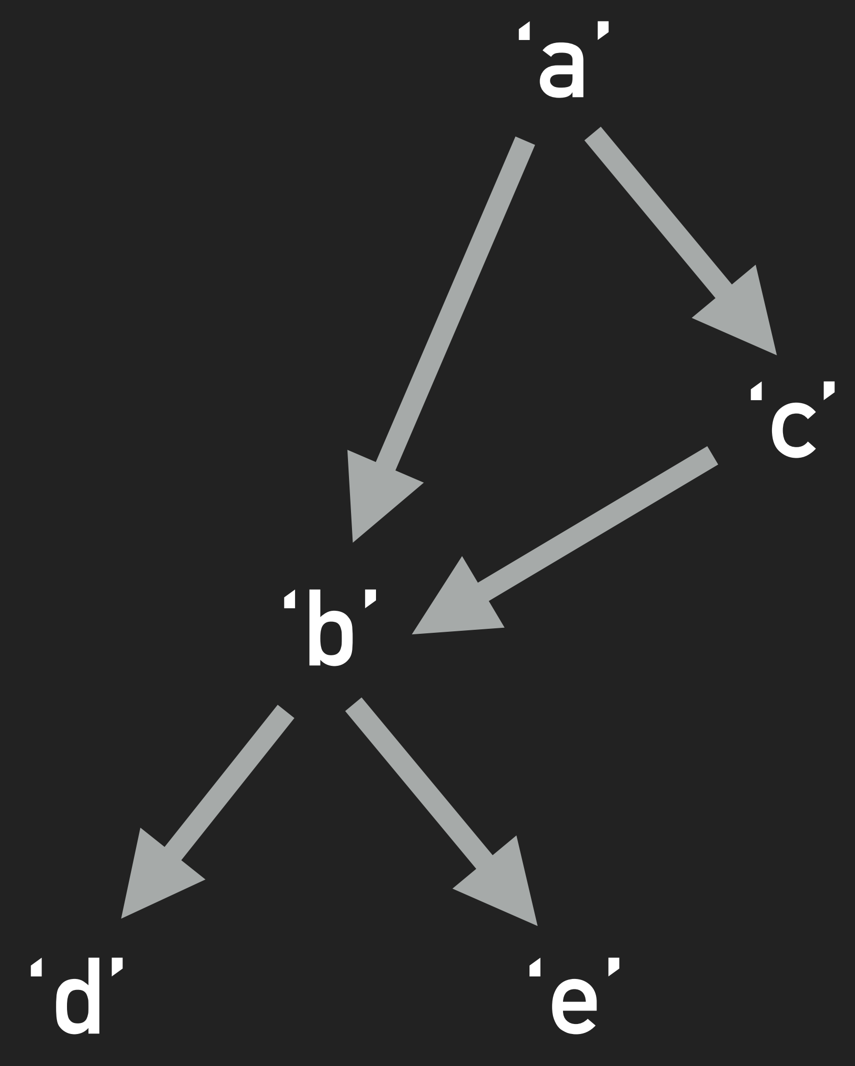
(‘a’, ‘b’)

(‘b’, ‘d’)
(‘b’, ‘e’)

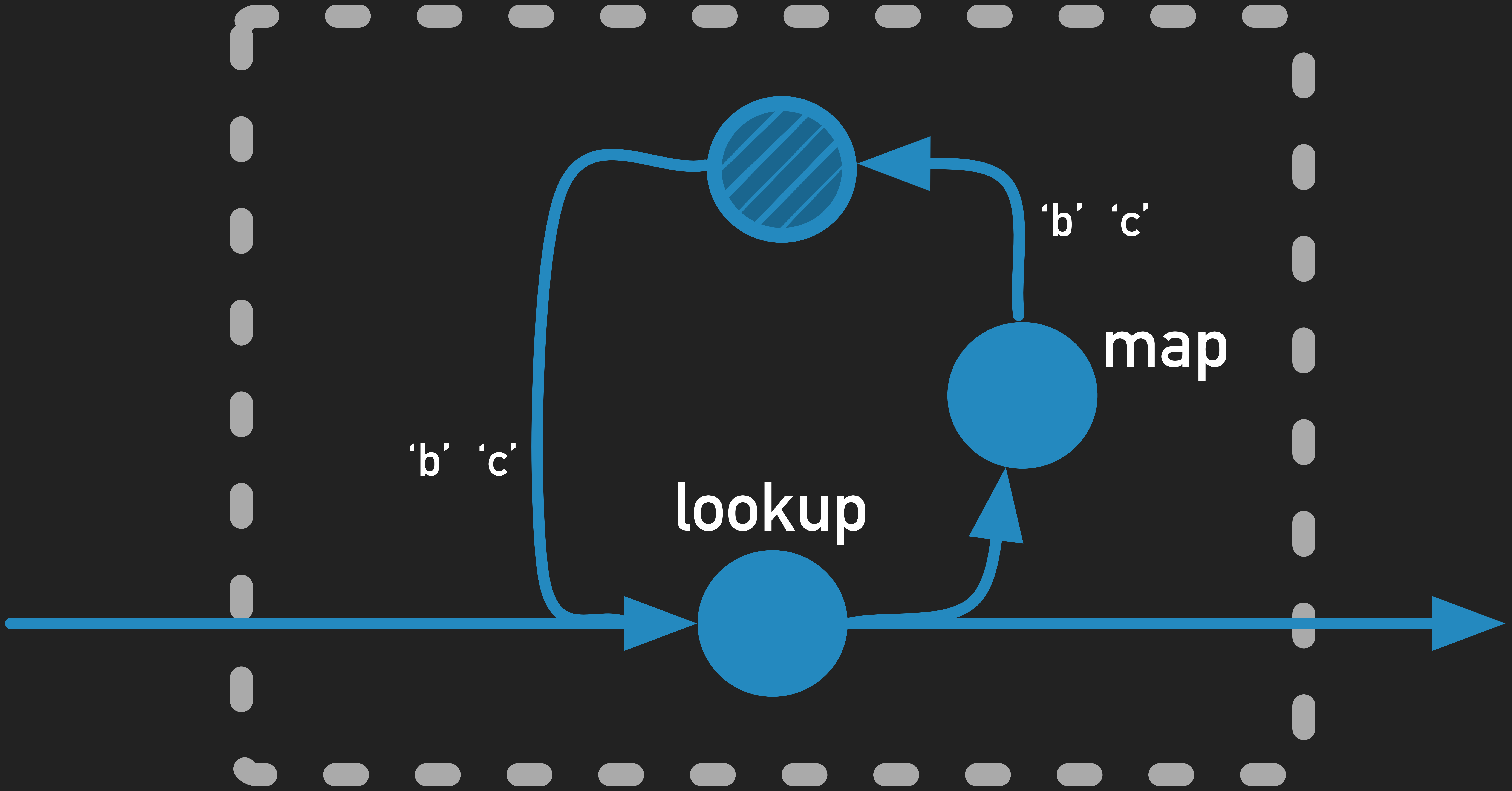
(‘a’, ‘c’)
(‘c’, ‘b’)
(‘b’, ‘d’)
(‘b’, ‘e’)



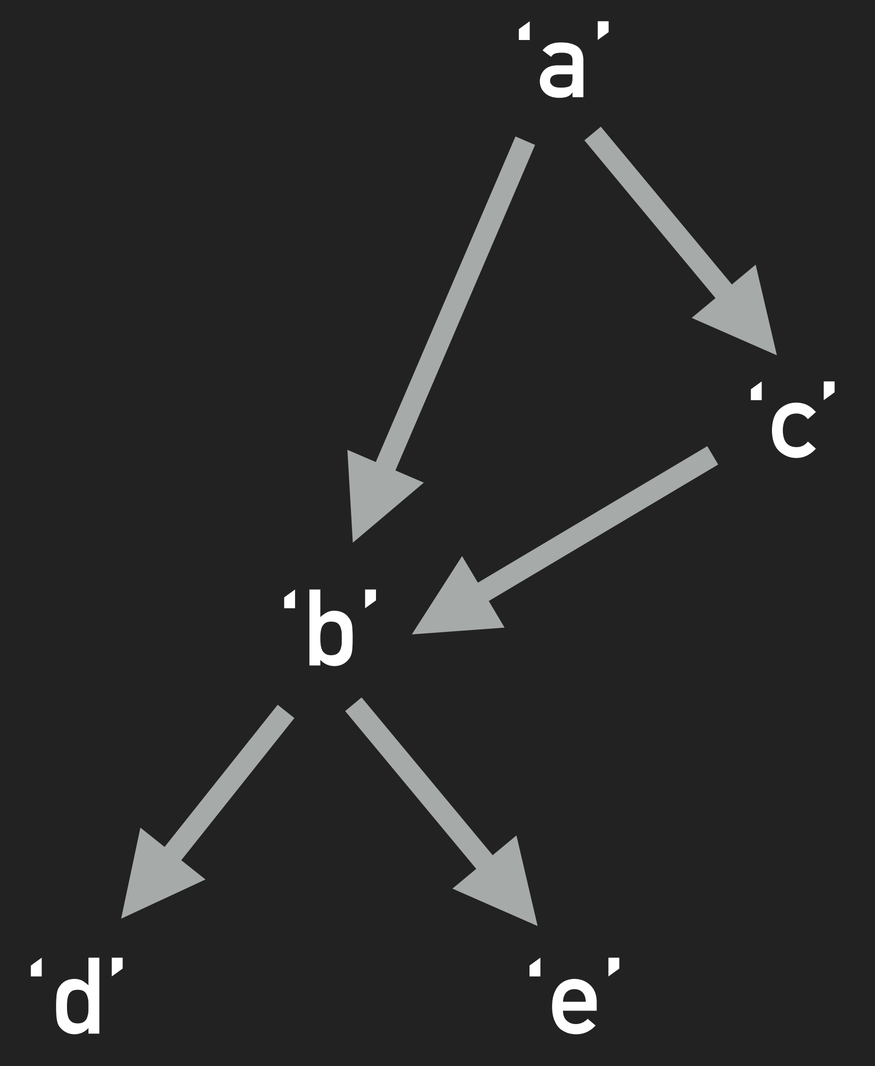
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



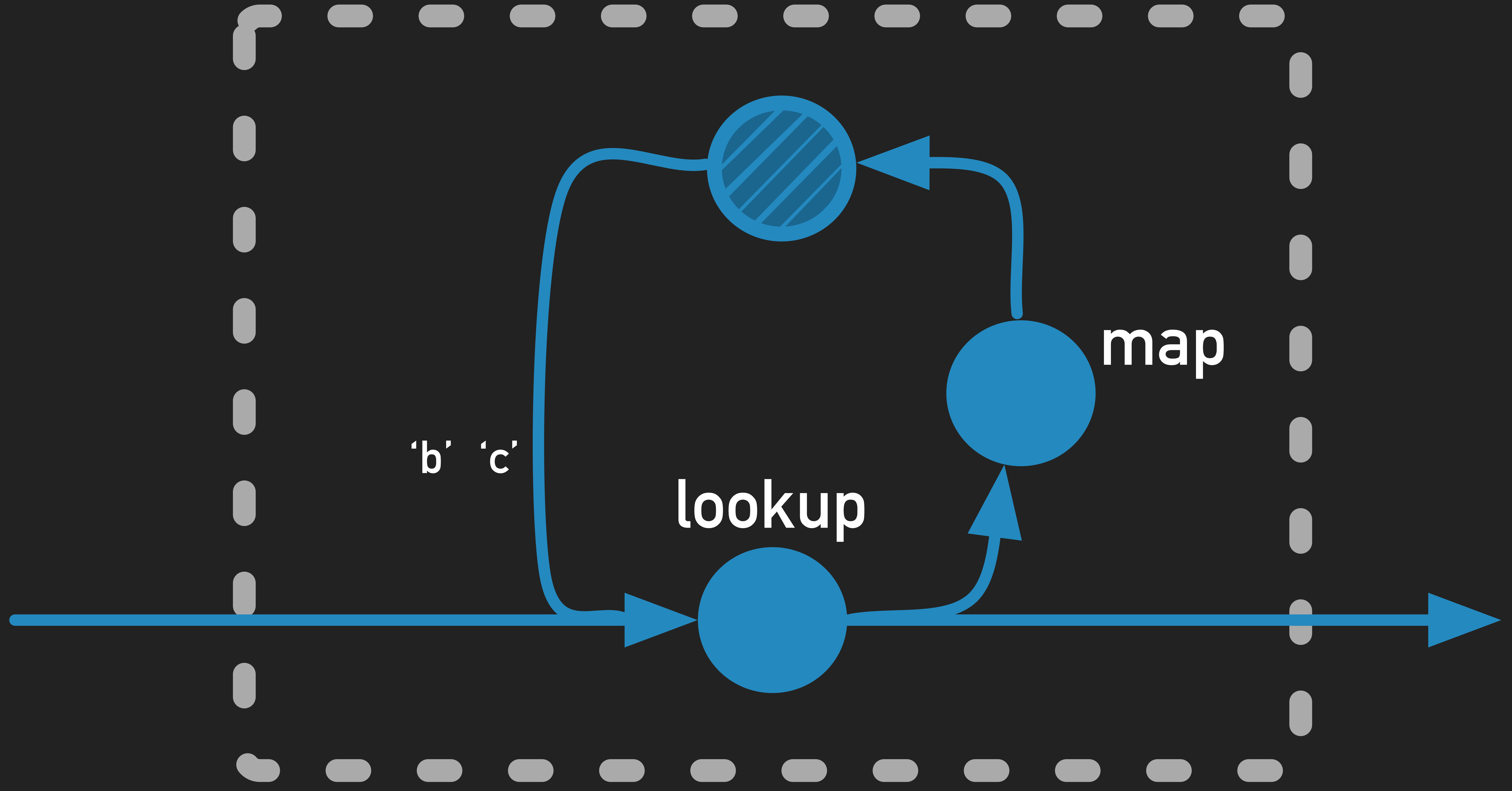
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



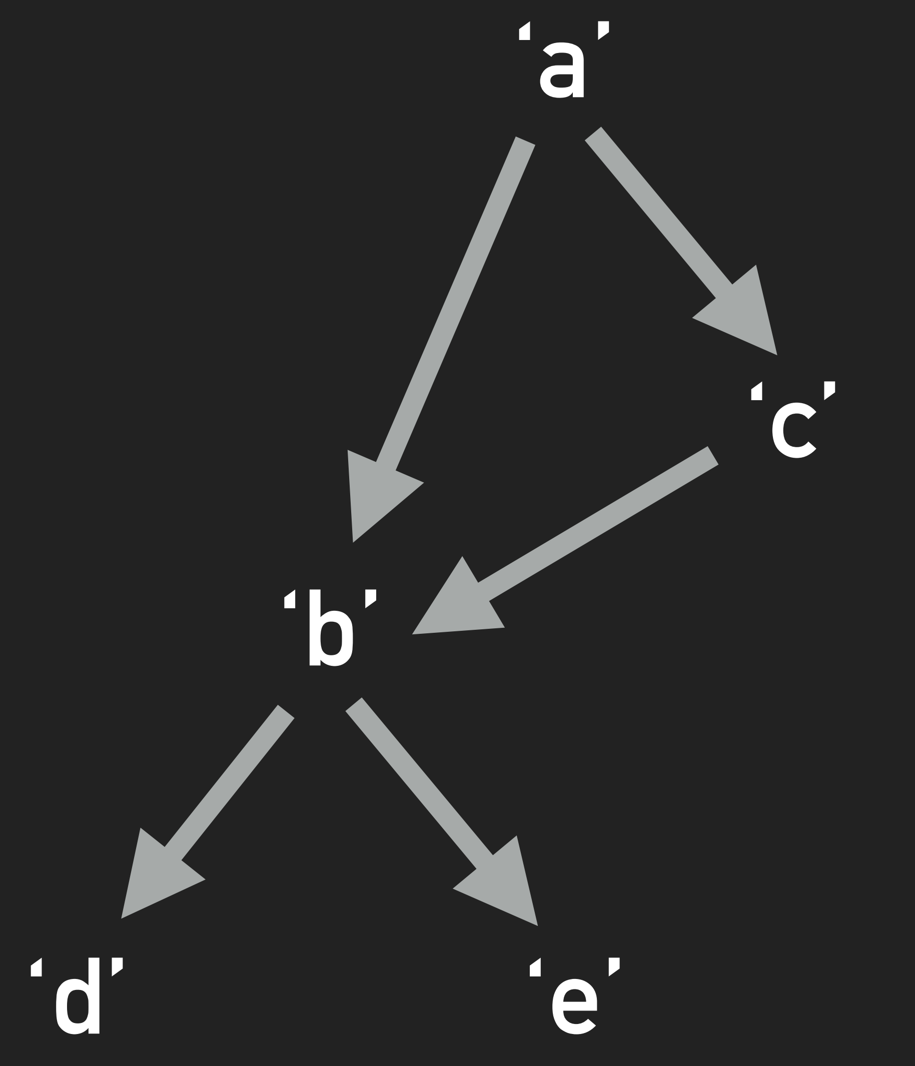
('a', 'b')

('b', 'd')
('b', 'e')

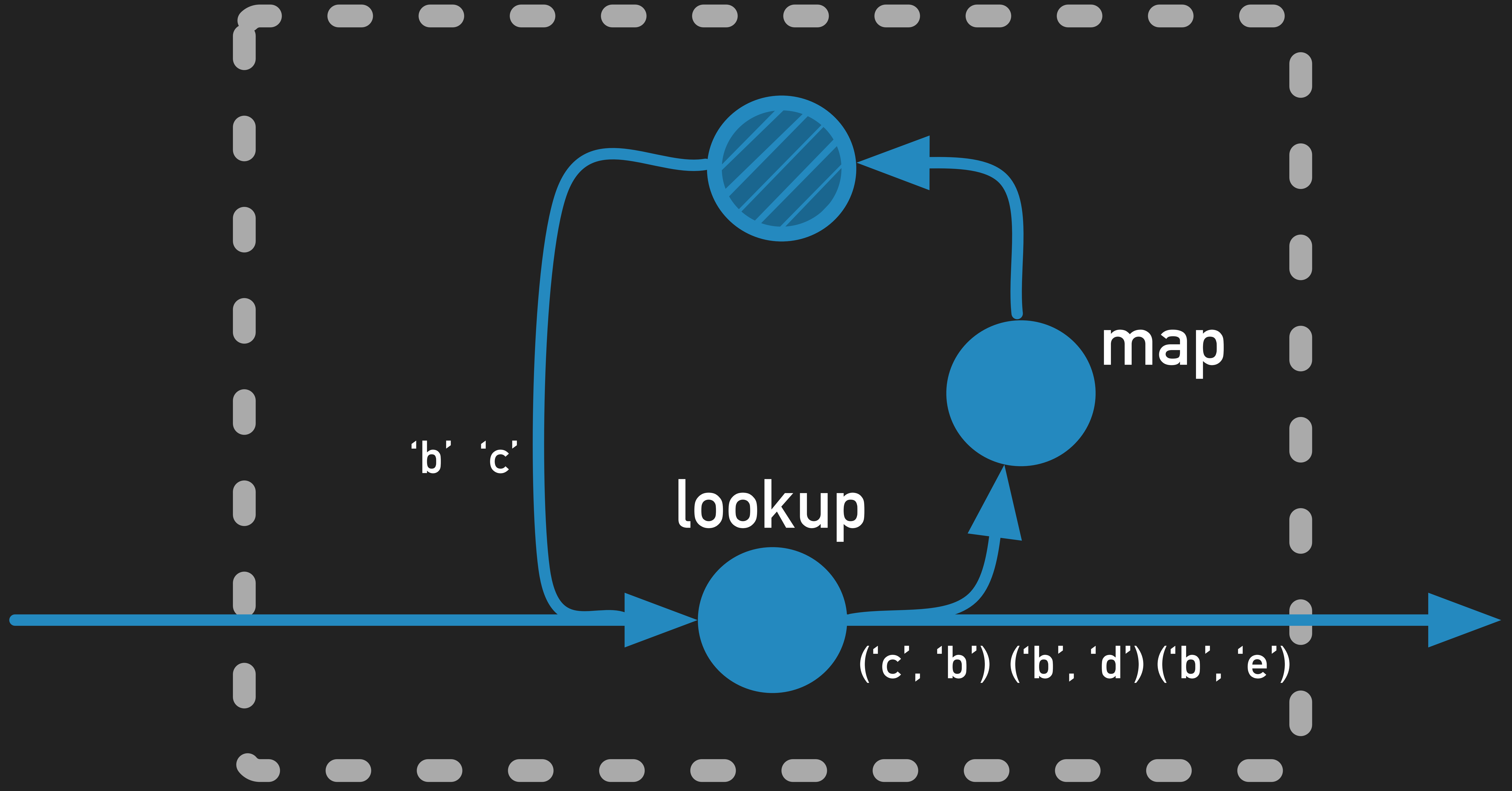
('a', 'c')
('c', 'b')
('b', 'd')
('b', 'e')



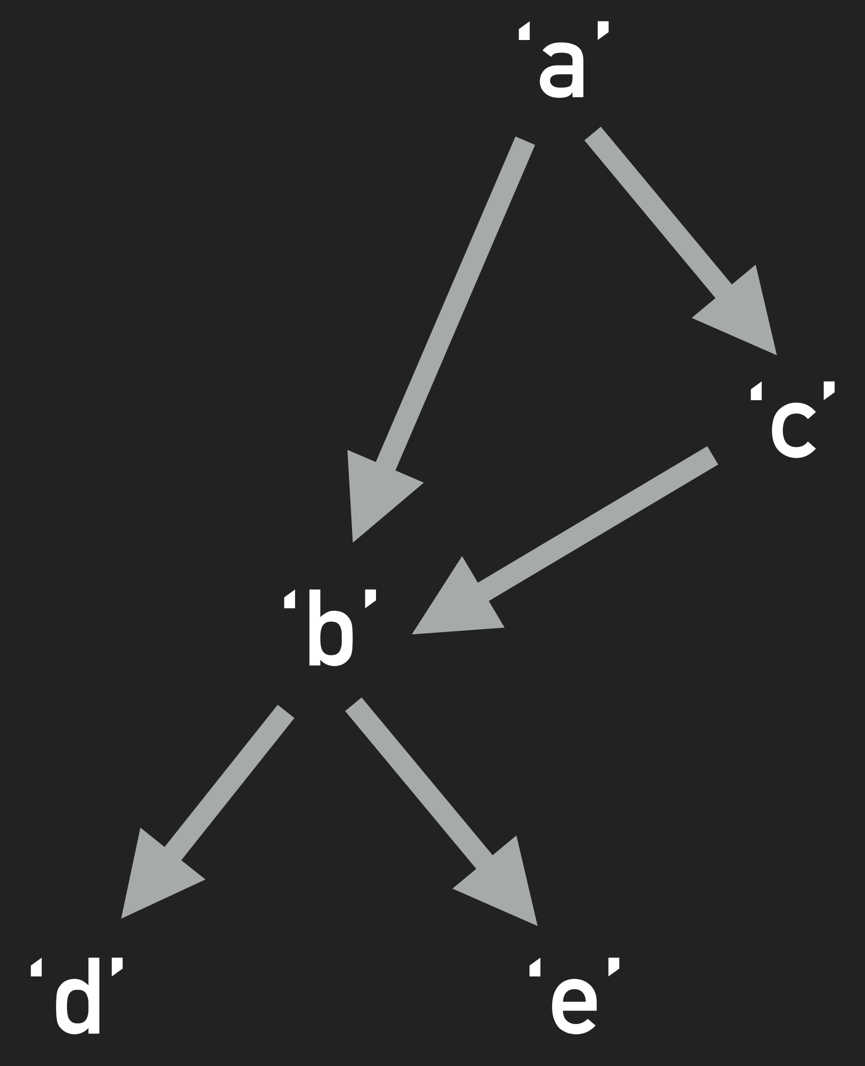
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



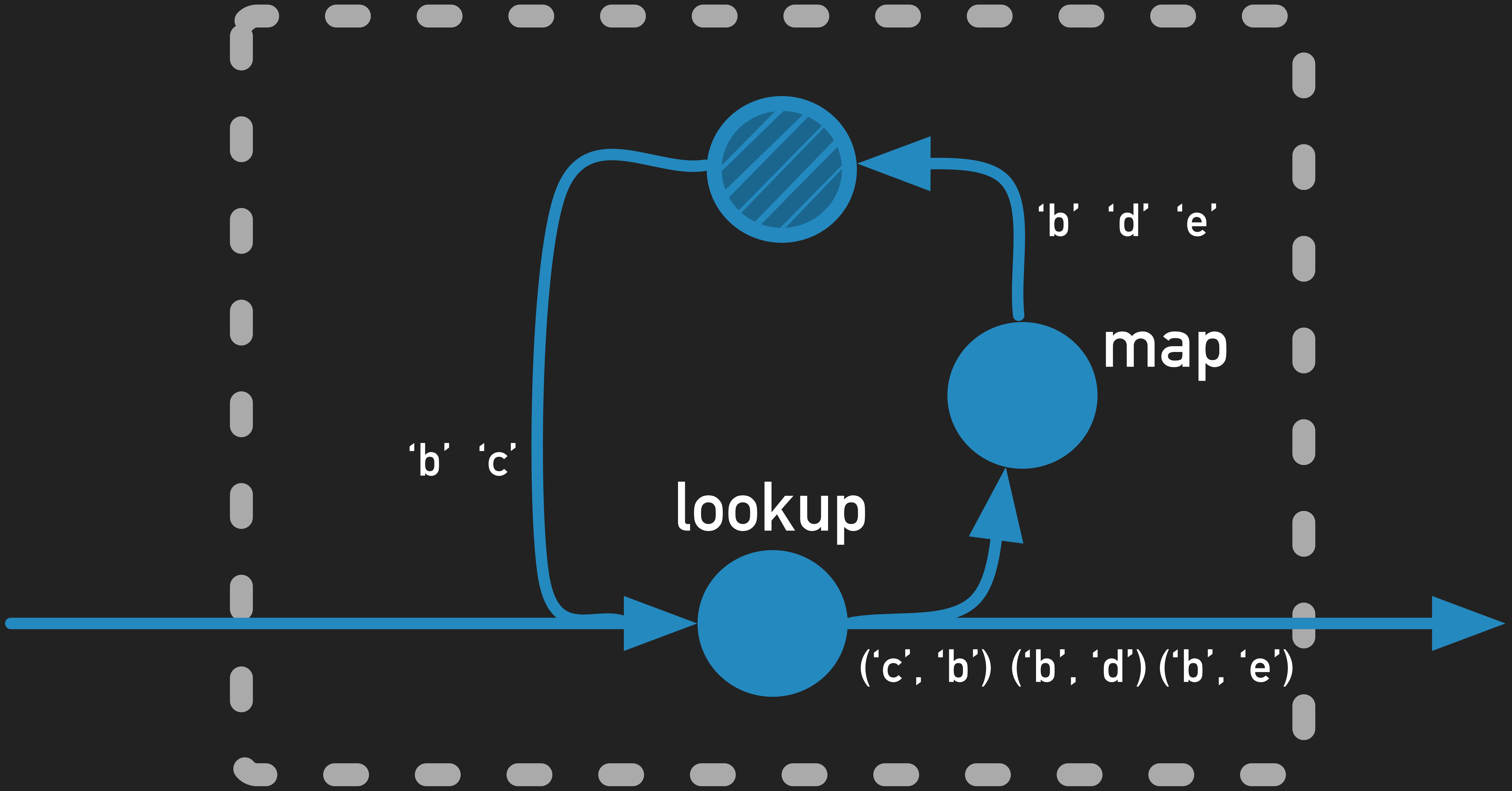
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



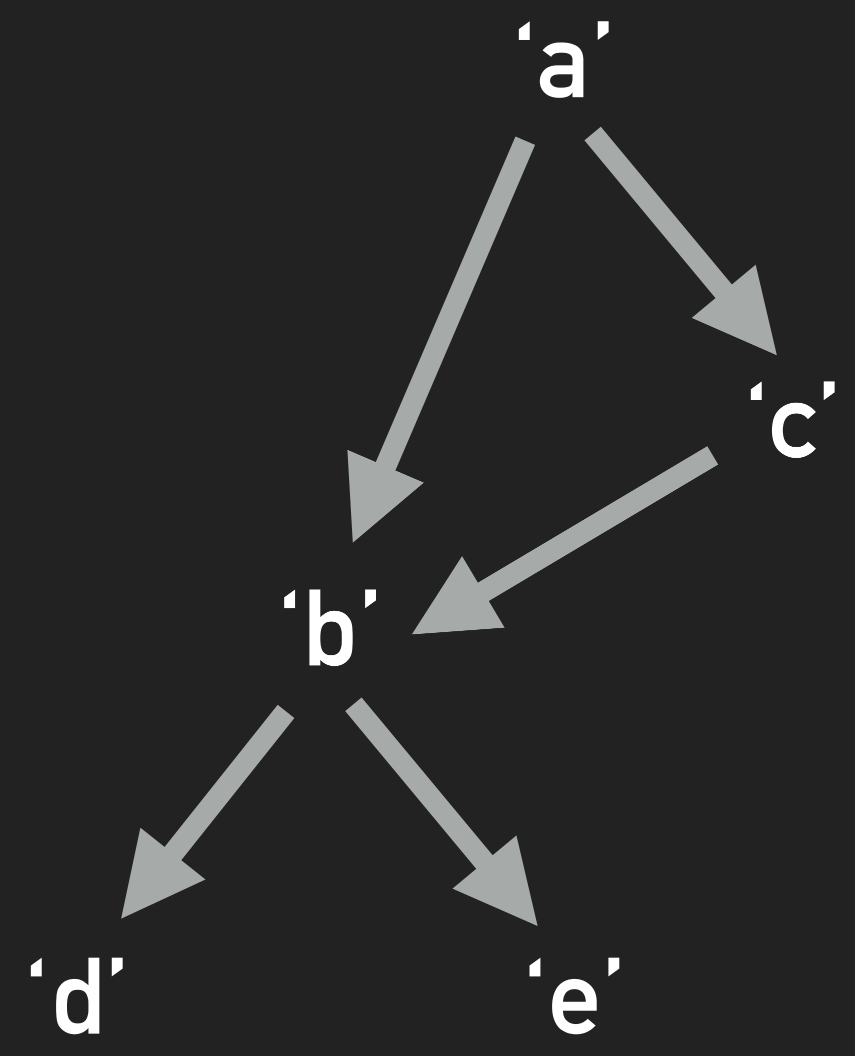
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



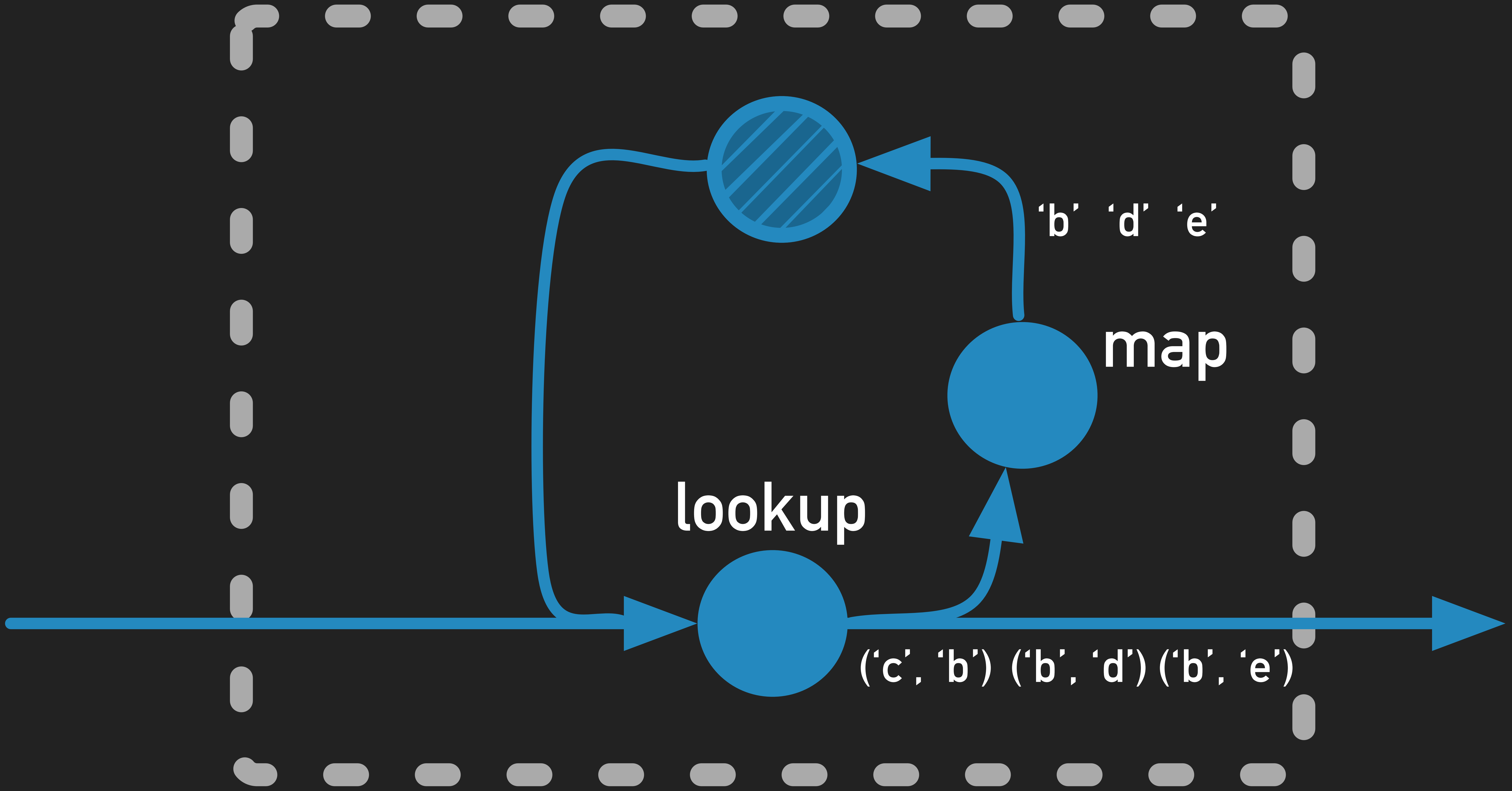
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



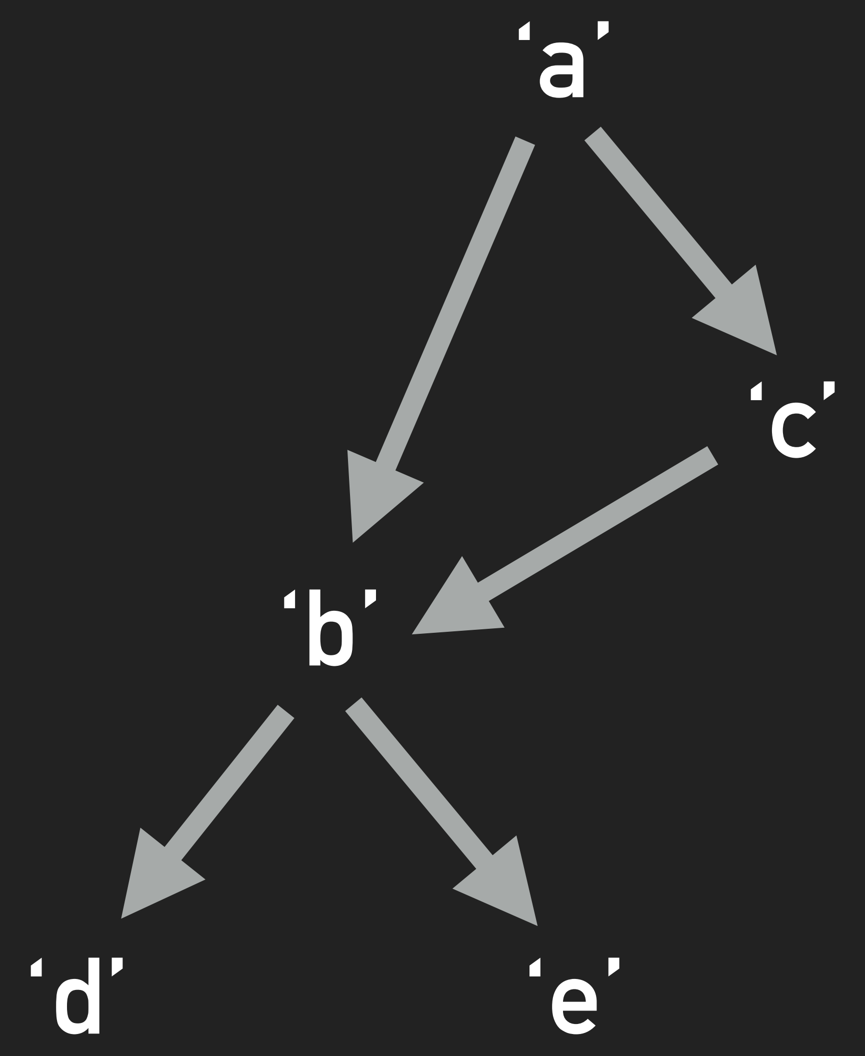
(‘a’, ‘b’)

(‘b’, ‘d’)
(‘b’, ‘e’)

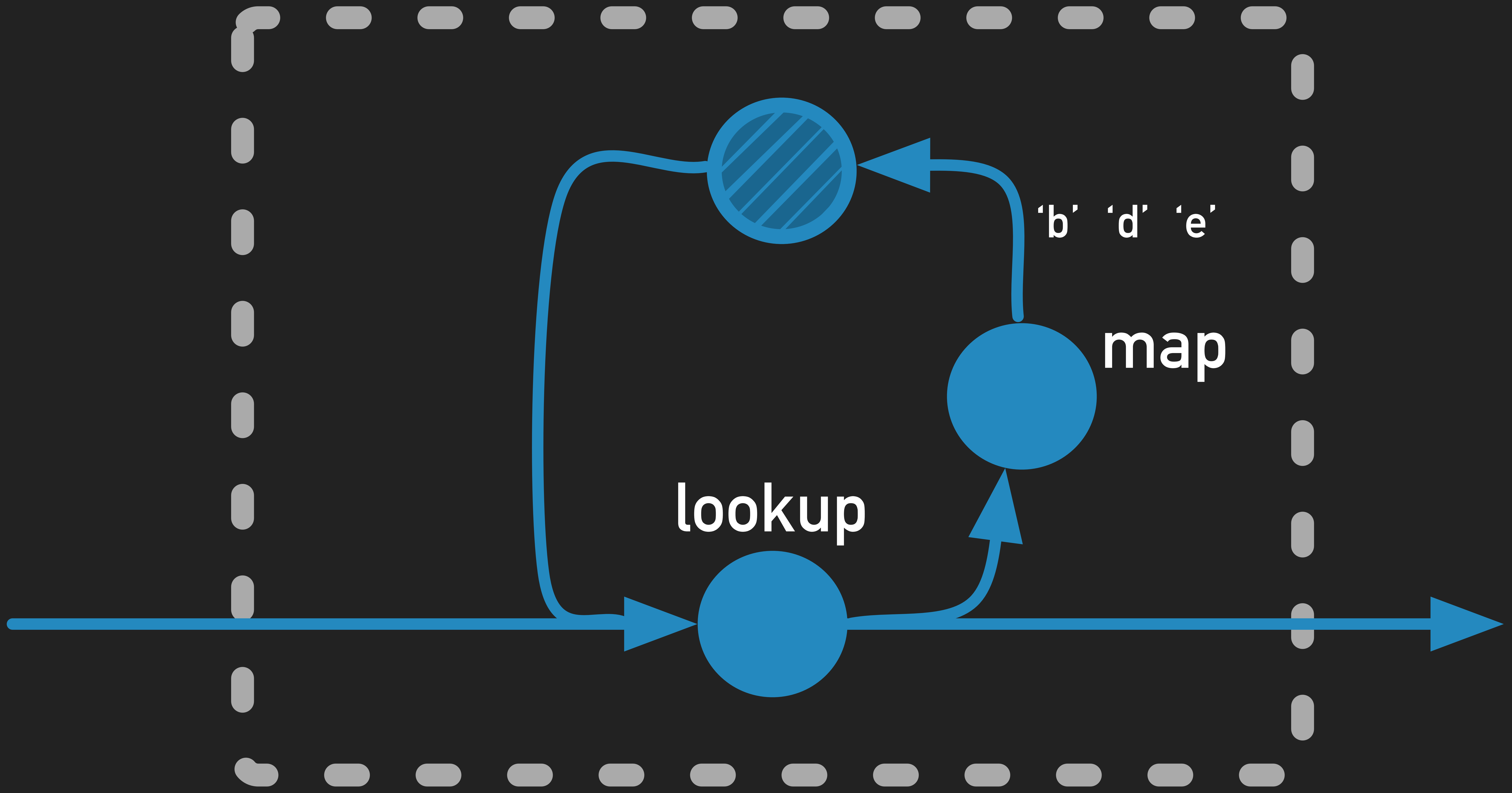
(‘a’, ‘c’)
(‘c’, ‘b’)
(‘b’, ‘d’)
(‘b’, ‘e’)



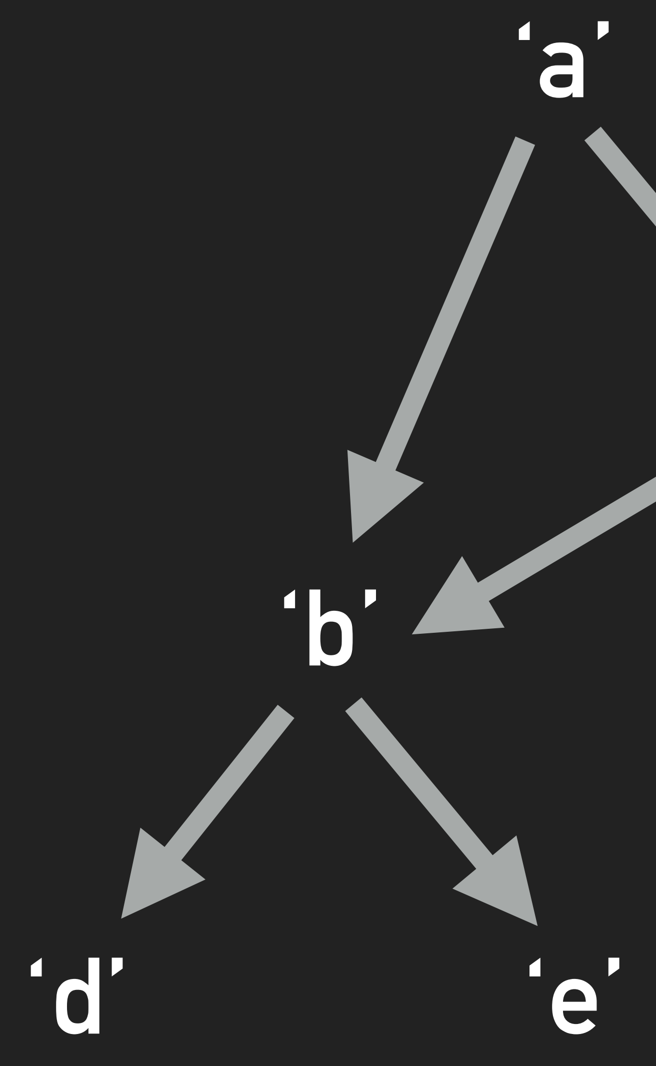
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



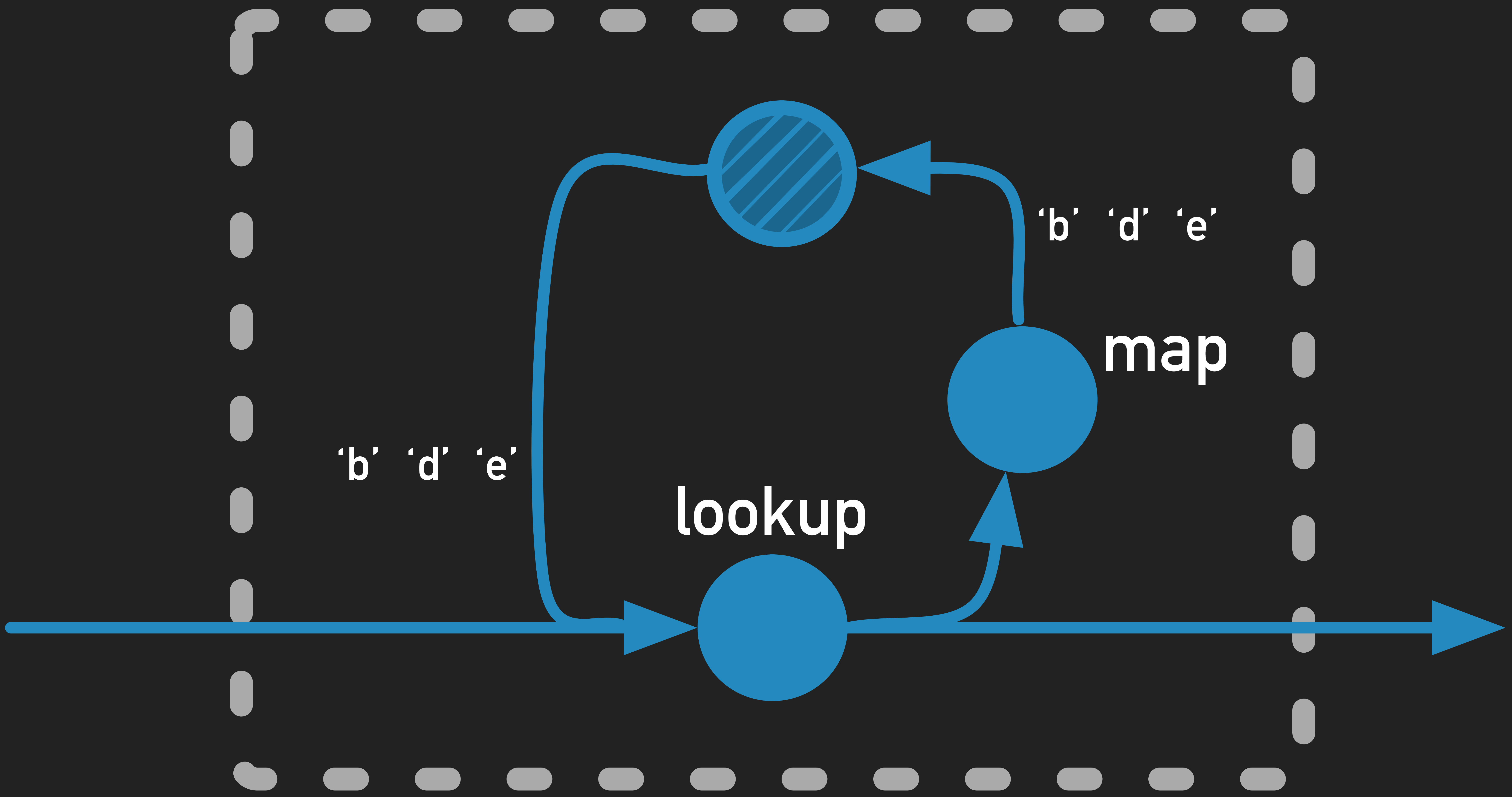
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



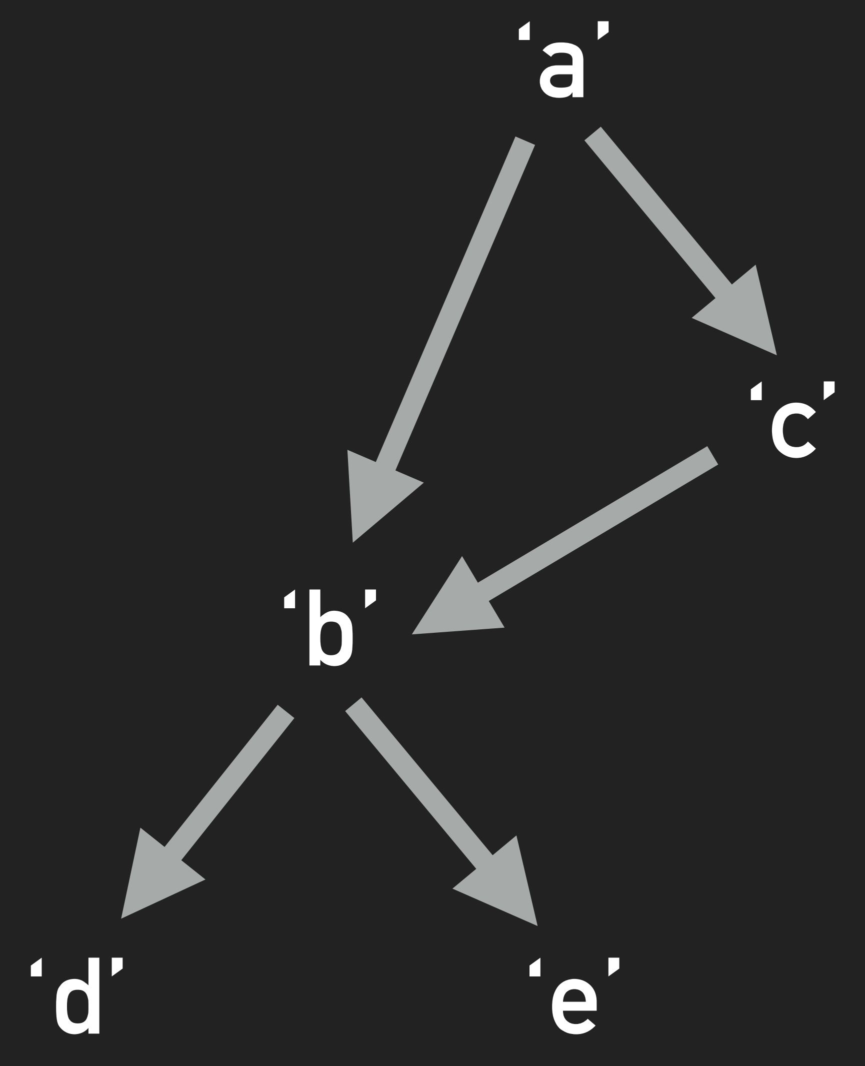
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



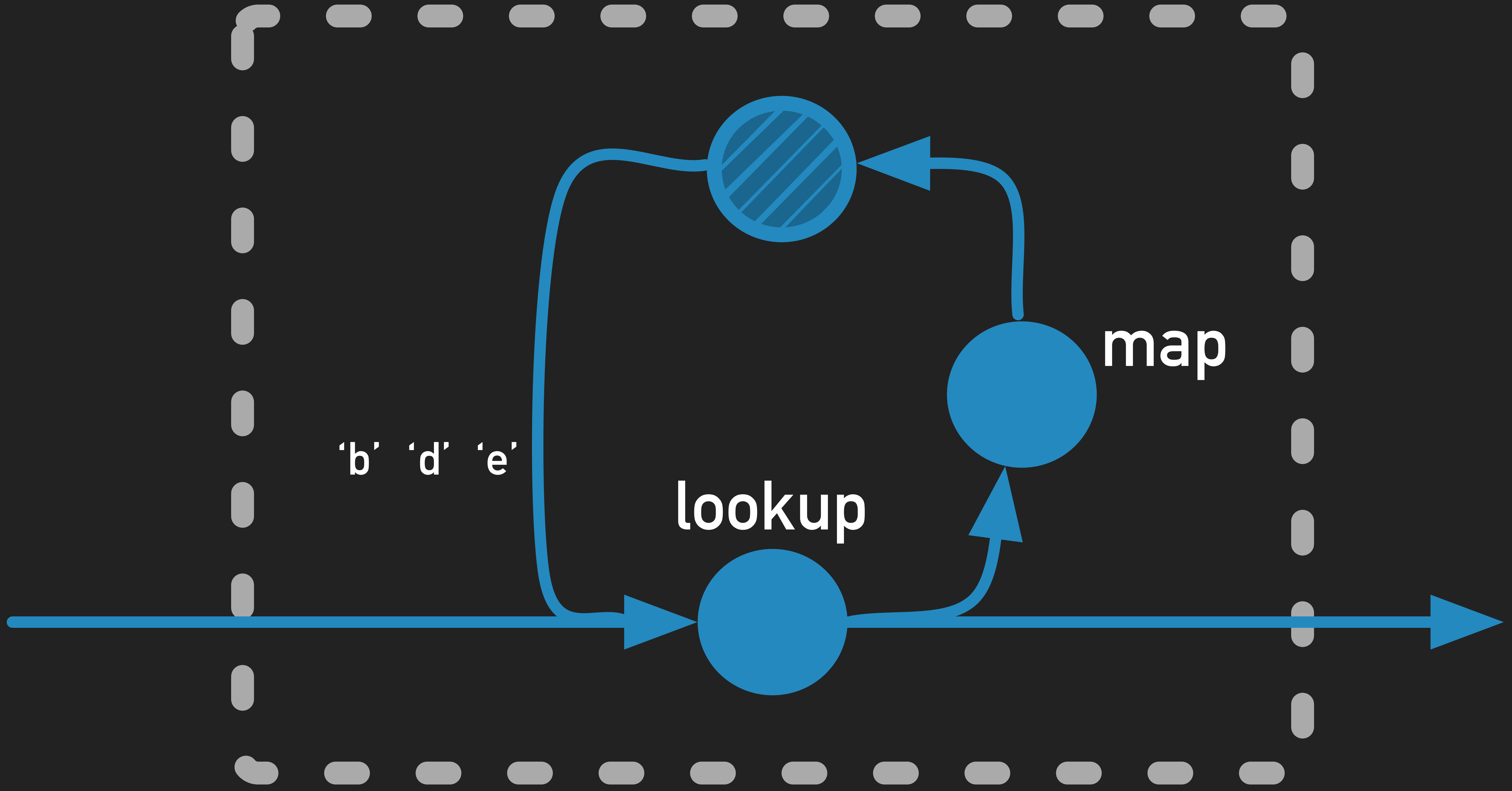
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



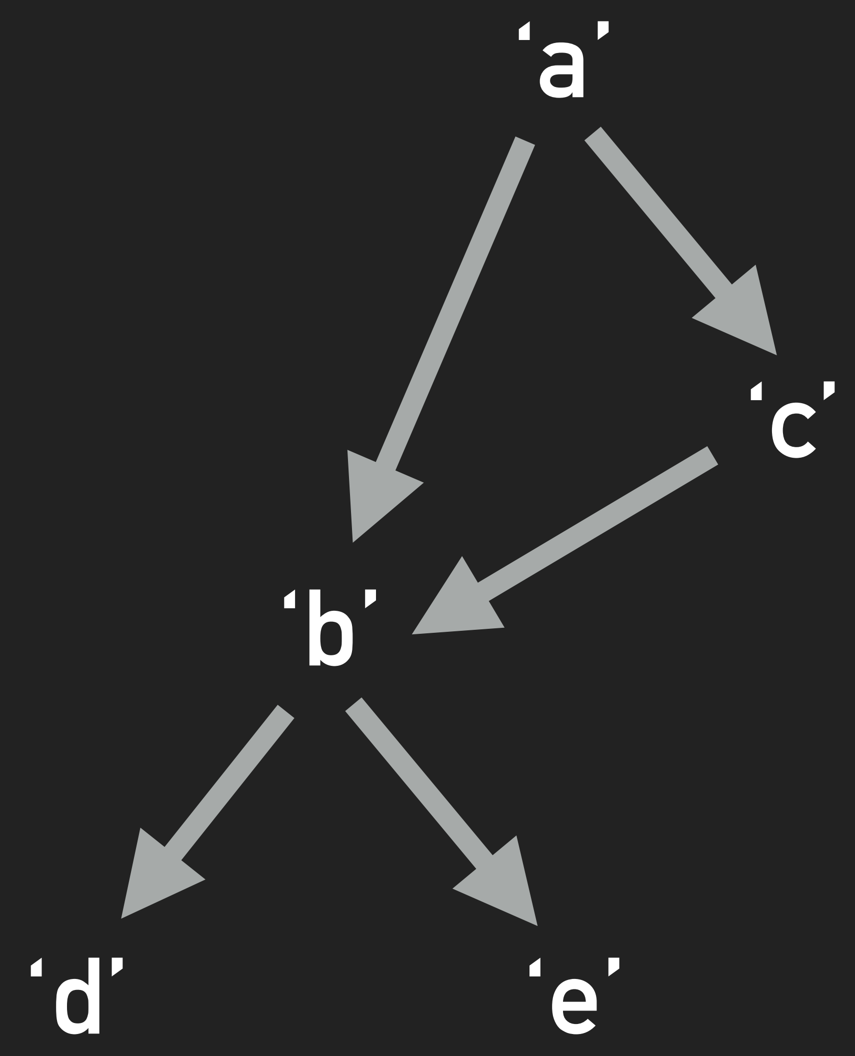
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



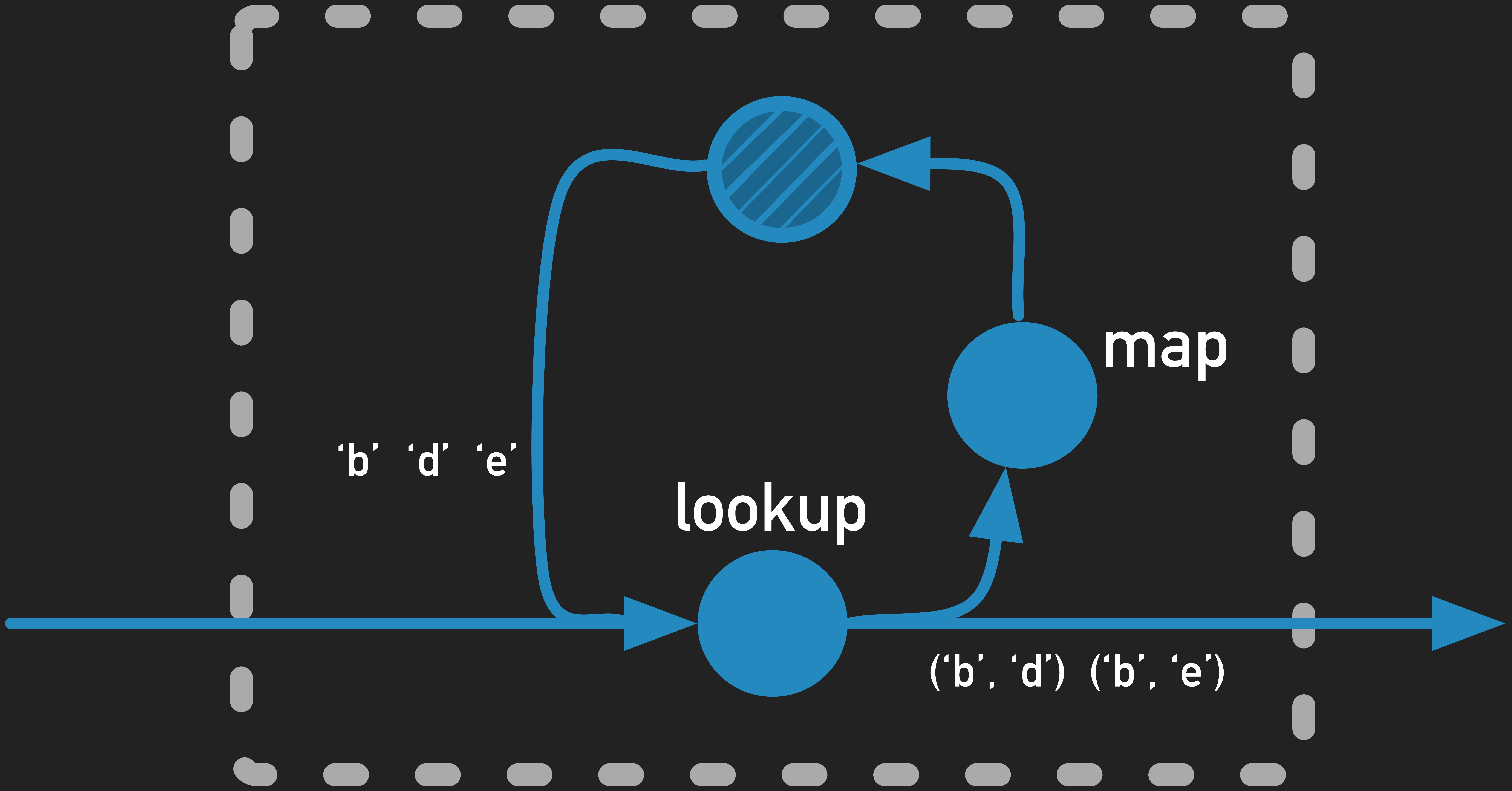
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



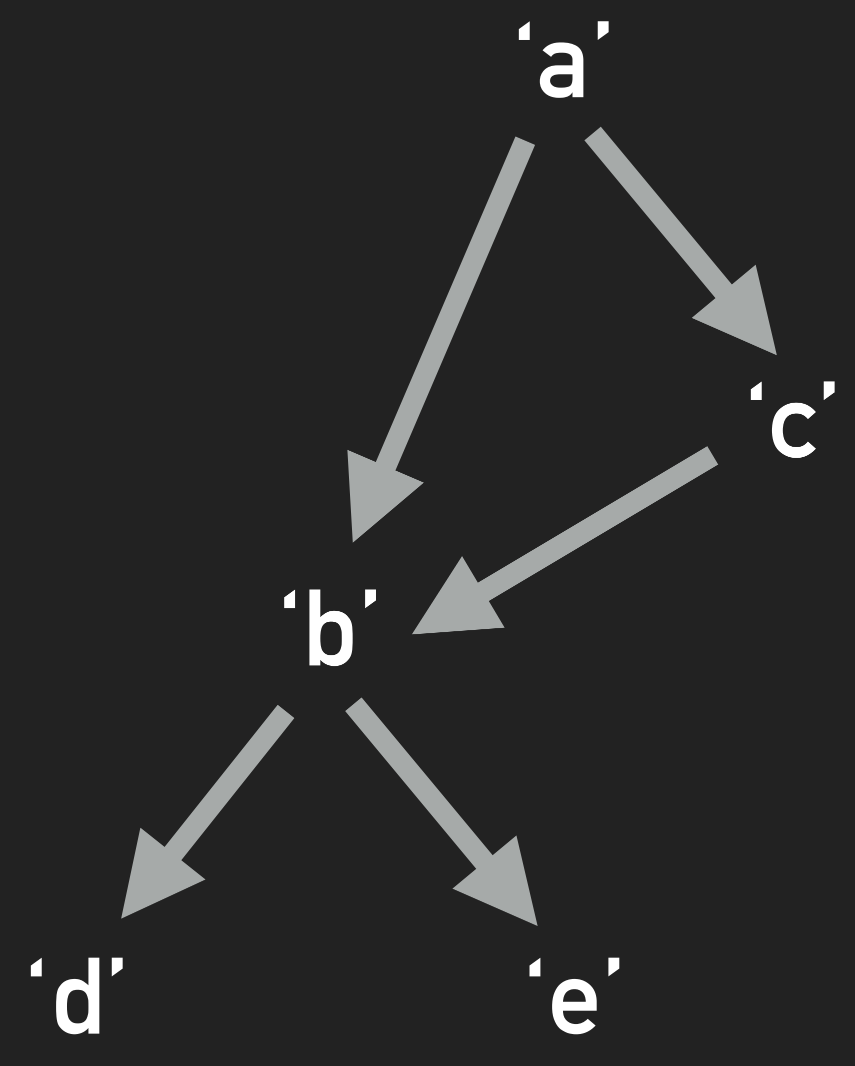
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



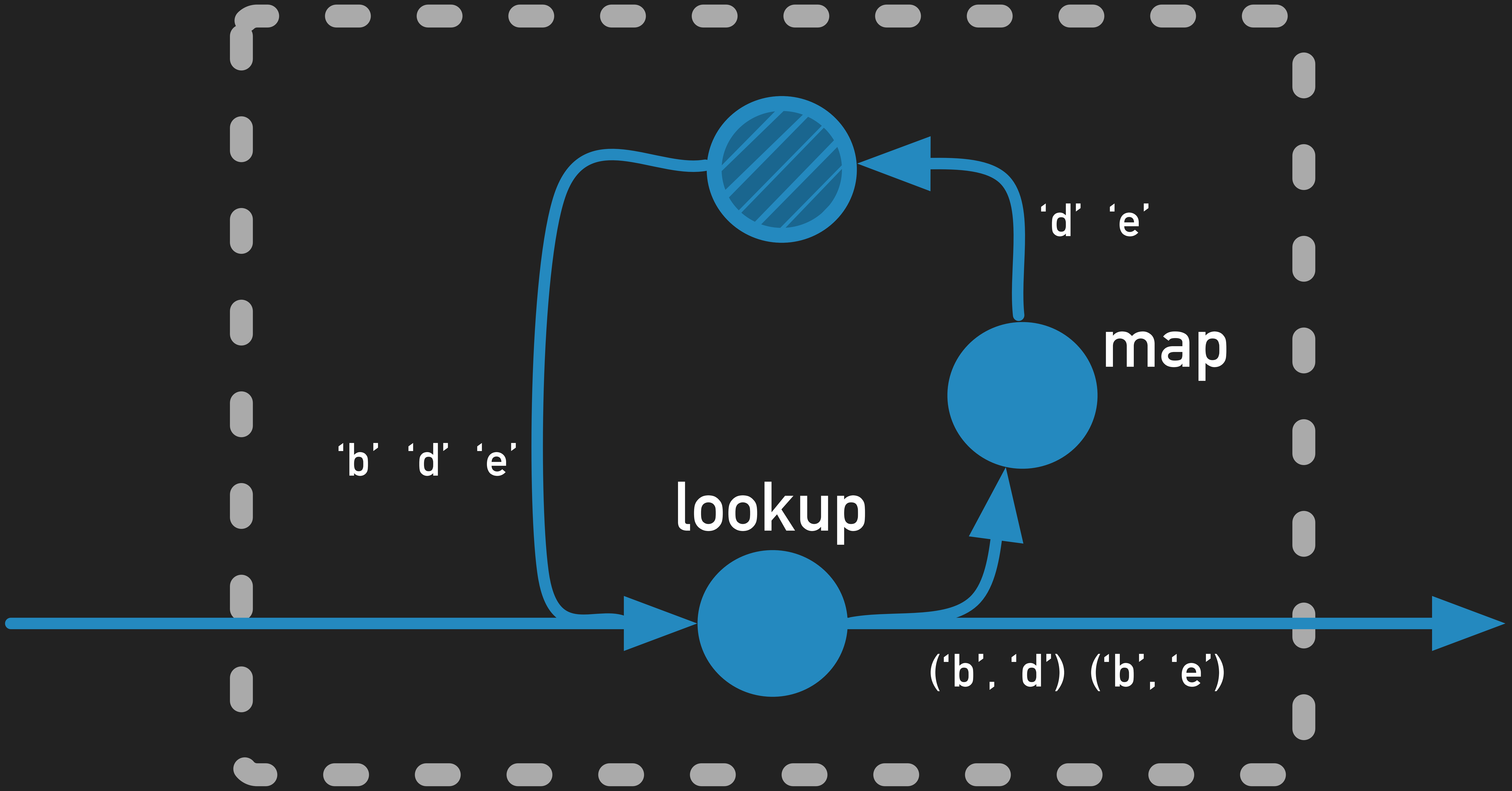
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



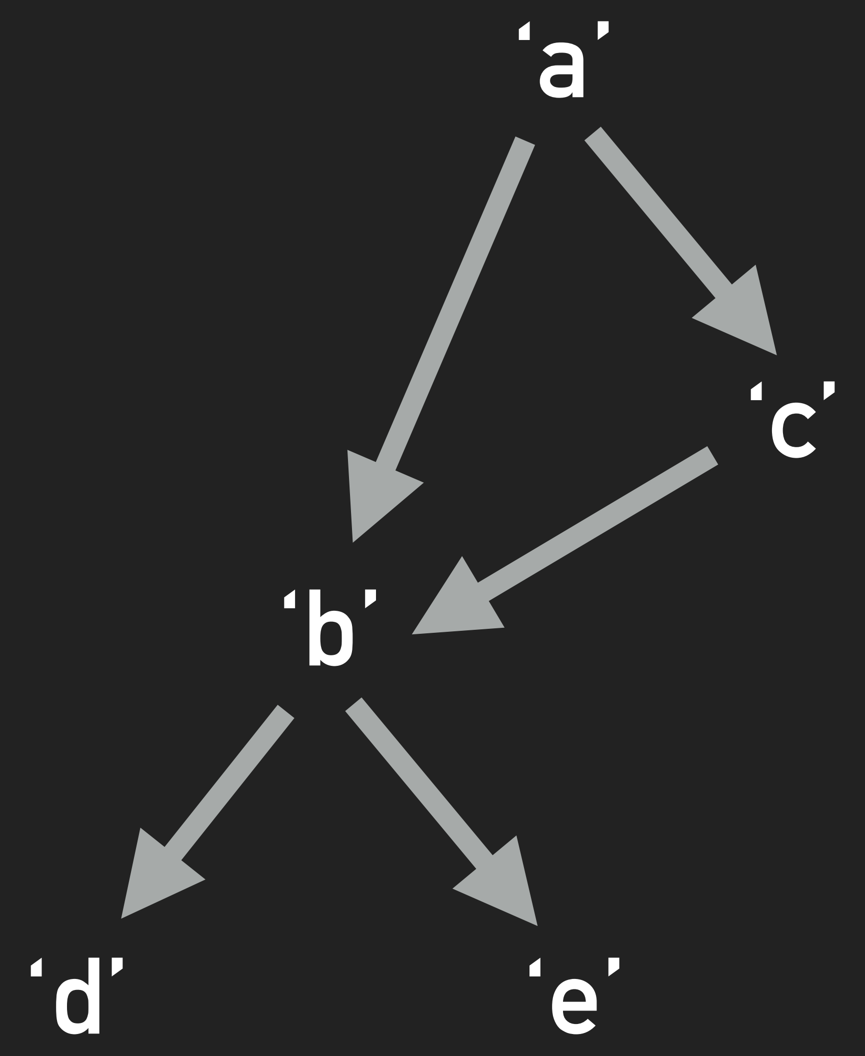
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



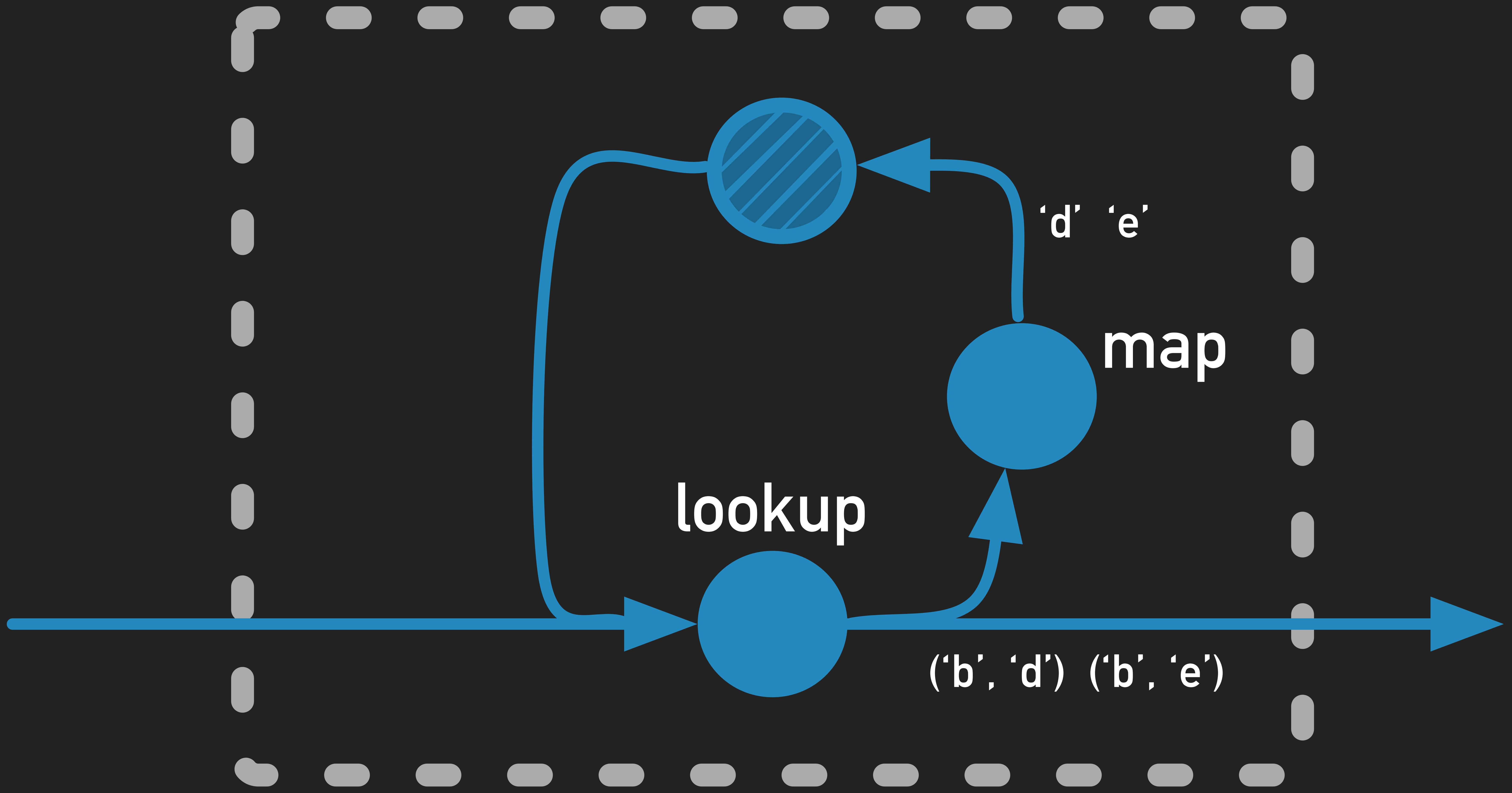
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



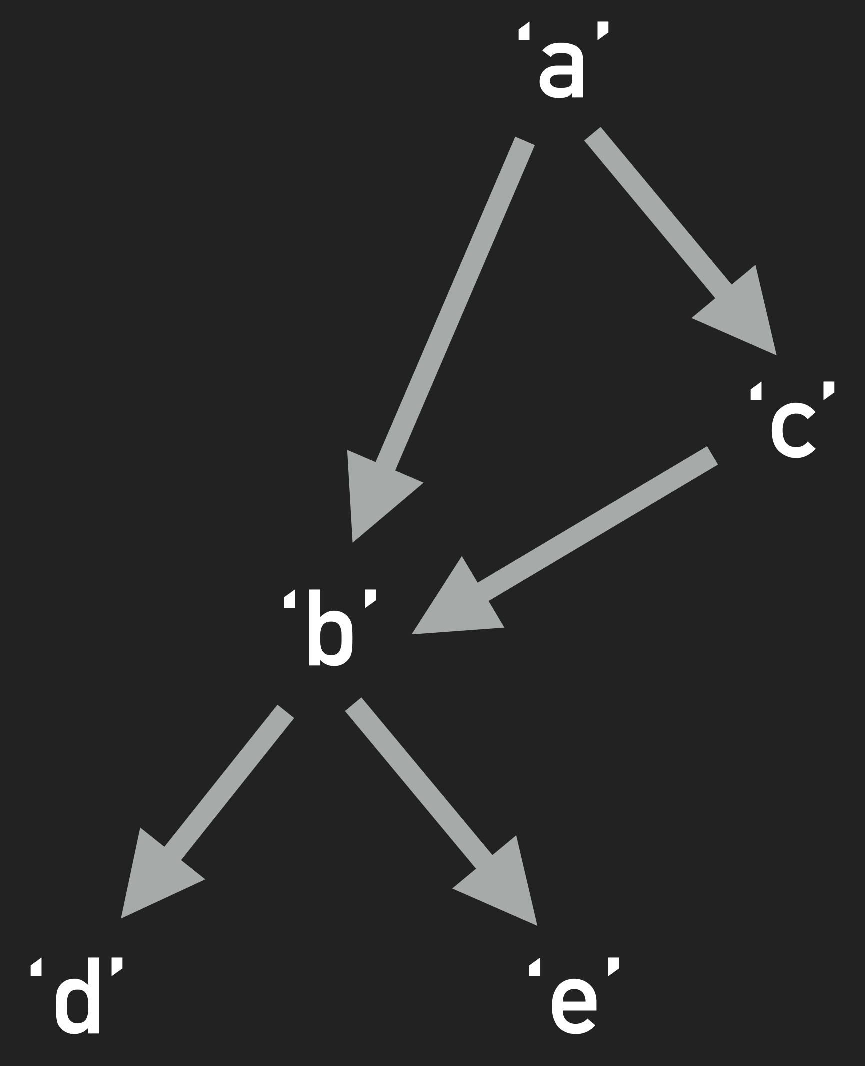
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



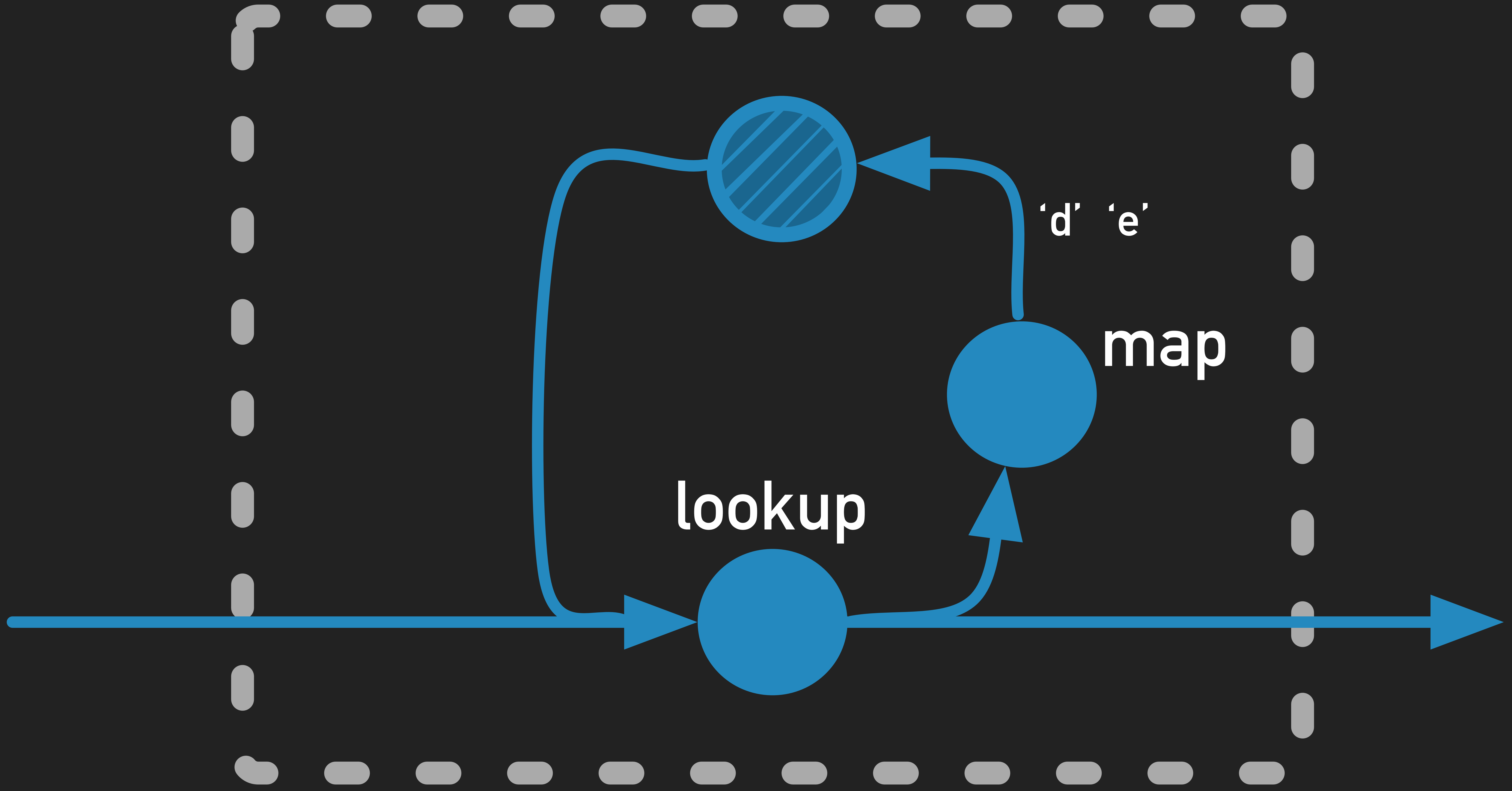
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



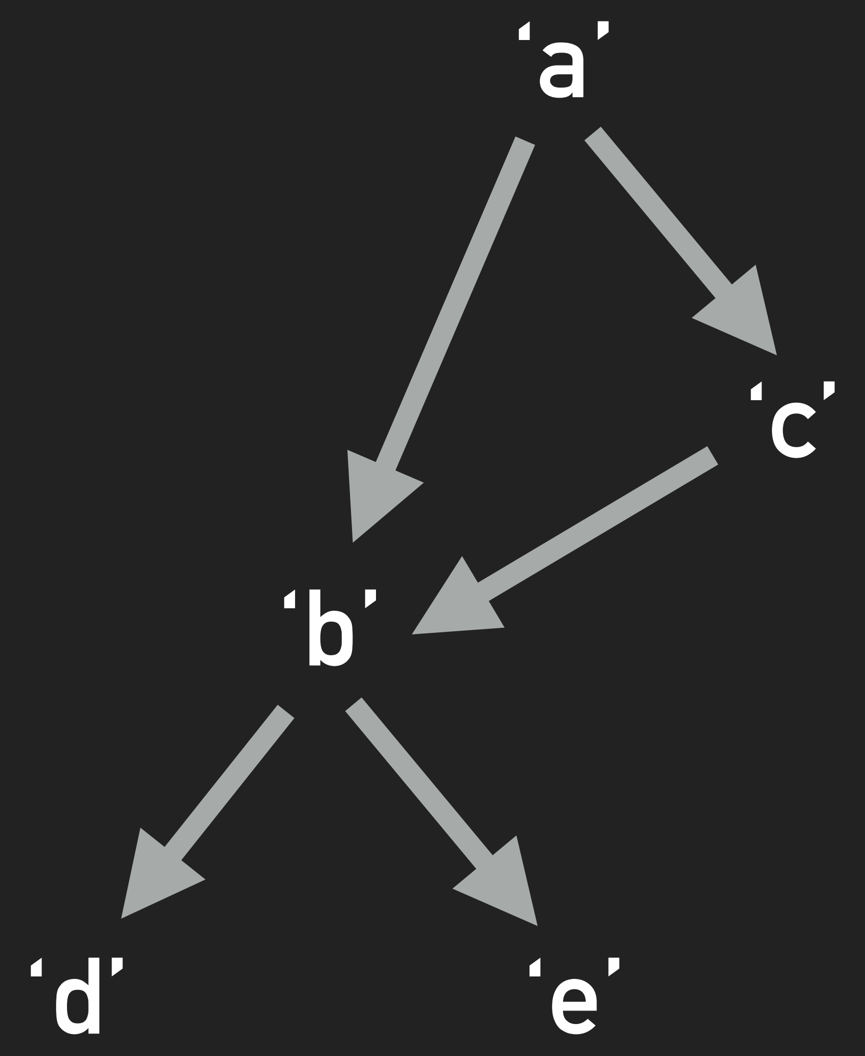
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



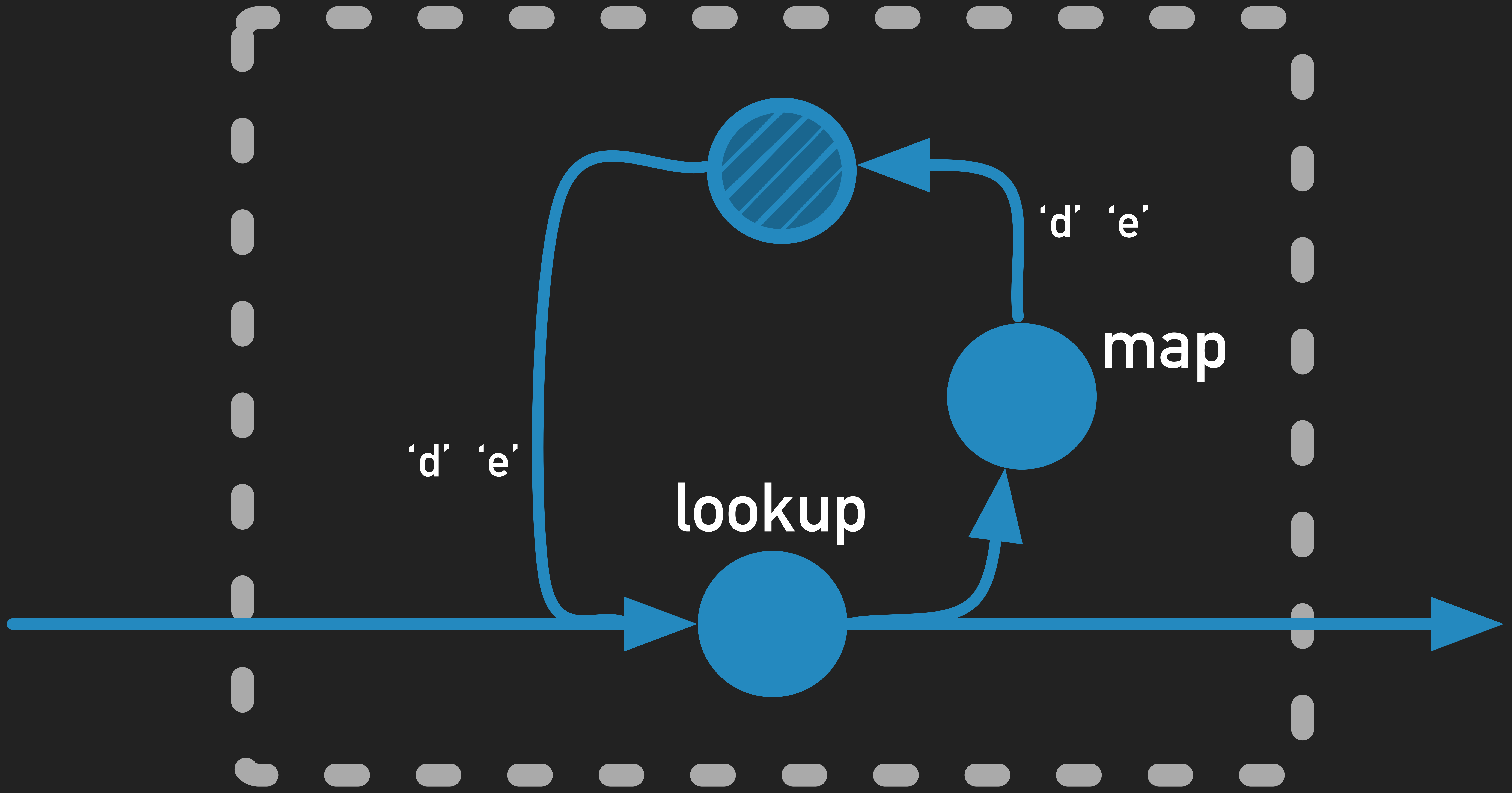
(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



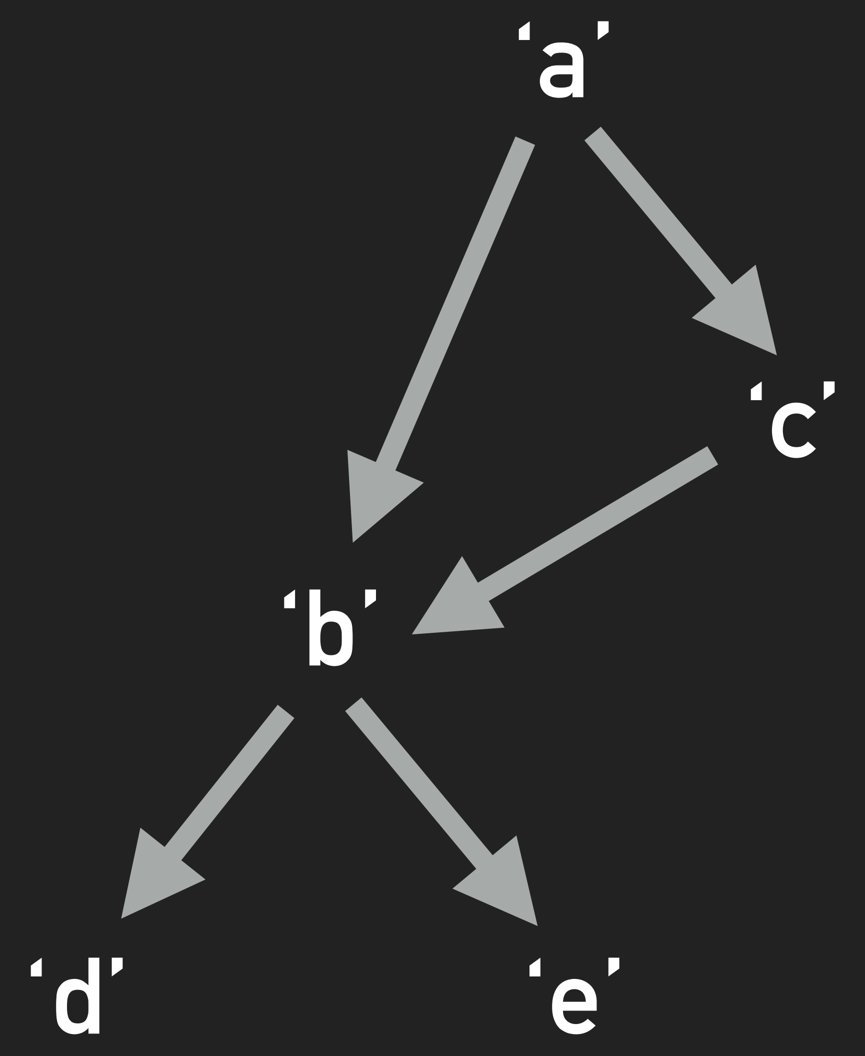
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)



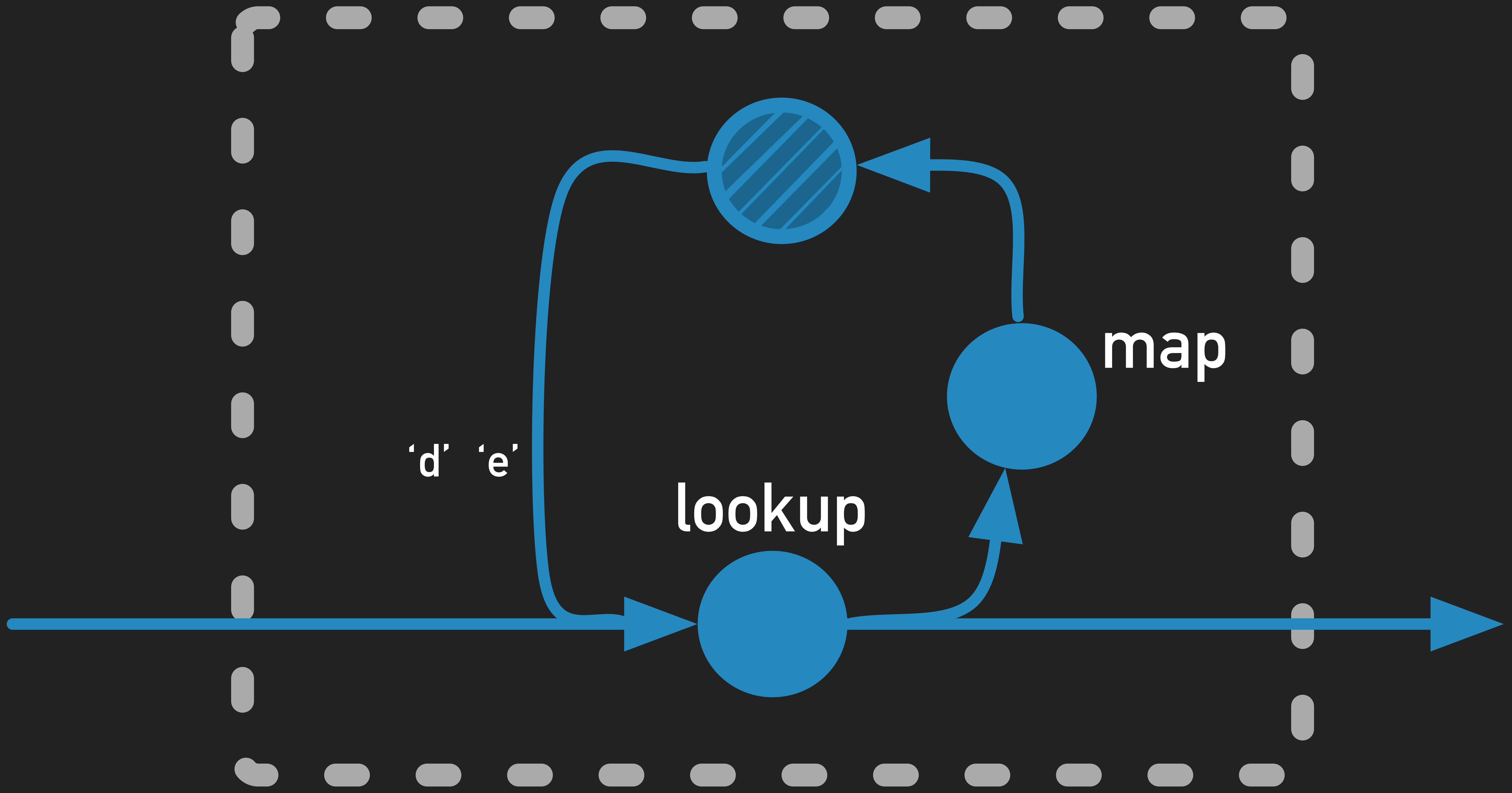
ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



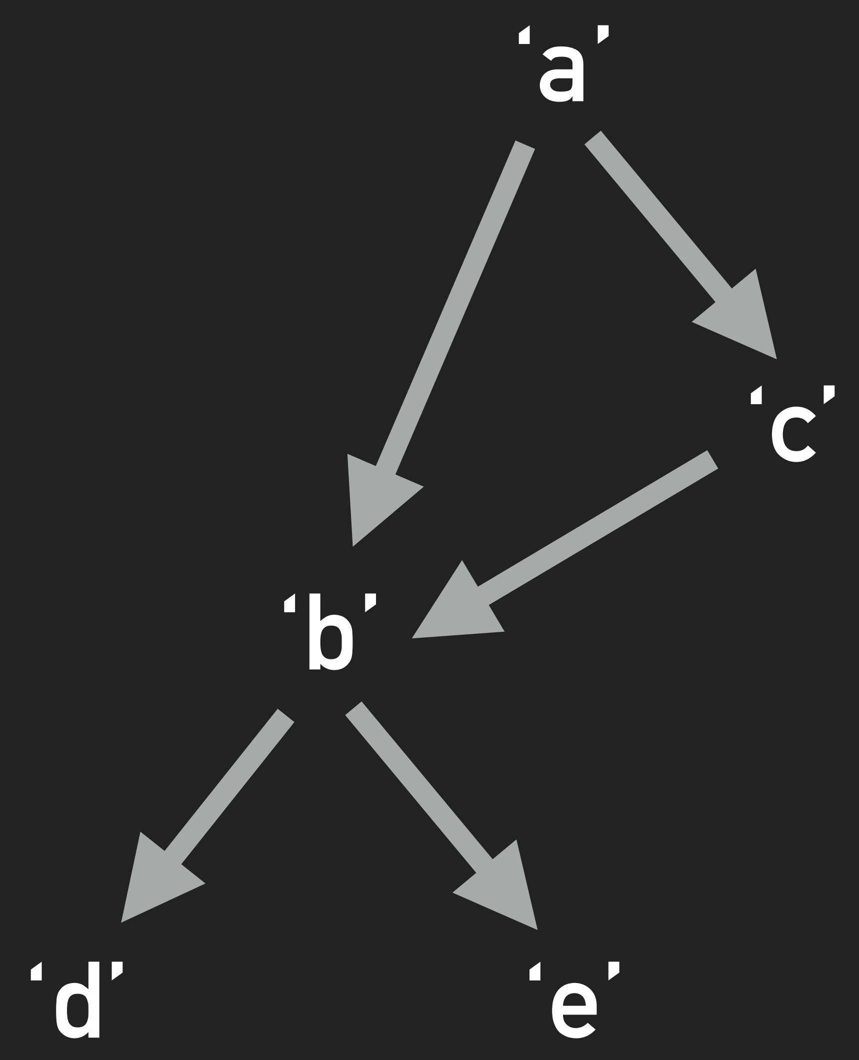
(‘a’, ‘b’)

(‘b’, ‘d’)
(‘b’, ‘e’)

(‘a’, ‘c’)
(‘c’, ‘b’)
(‘b’, ‘d’)
(‘b’, ‘e’)



ITERATIVE GRAPH COMPUTATION IN DATAFLOW.



(‘a’, ‘b’)	(‘a’, ‘c’)
	(‘c’, ‘b’)
(‘b’, ‘d’)	(‘b’, ‘d’)
(‘b’, ‘e’)	(‘b’, ‘e’)