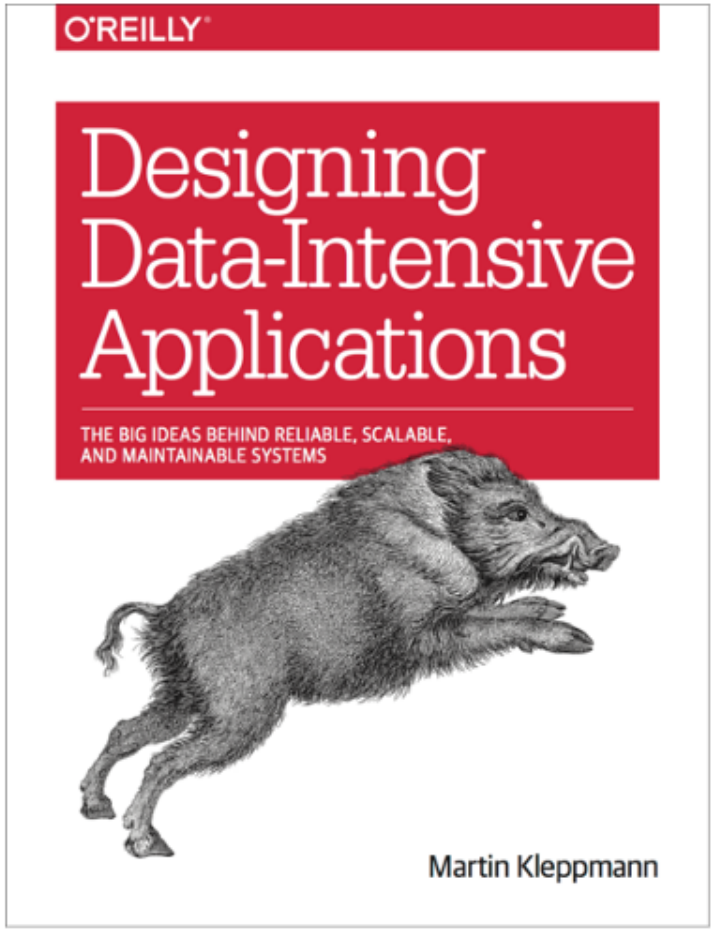
CONVERGENCE versus CONSENSUS

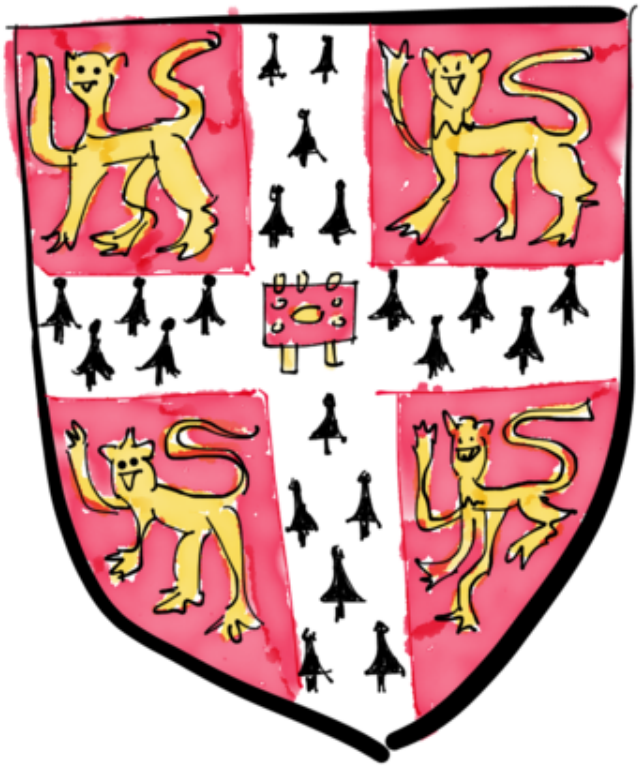
CRDTs and the quest for DISTRIBUTED CONSISTENCY

Martin Kleppmann • University of Cambridge • @martinkl

dataintensive.net



@martinkl



UNIVERSITY OF CAMBRIDGE

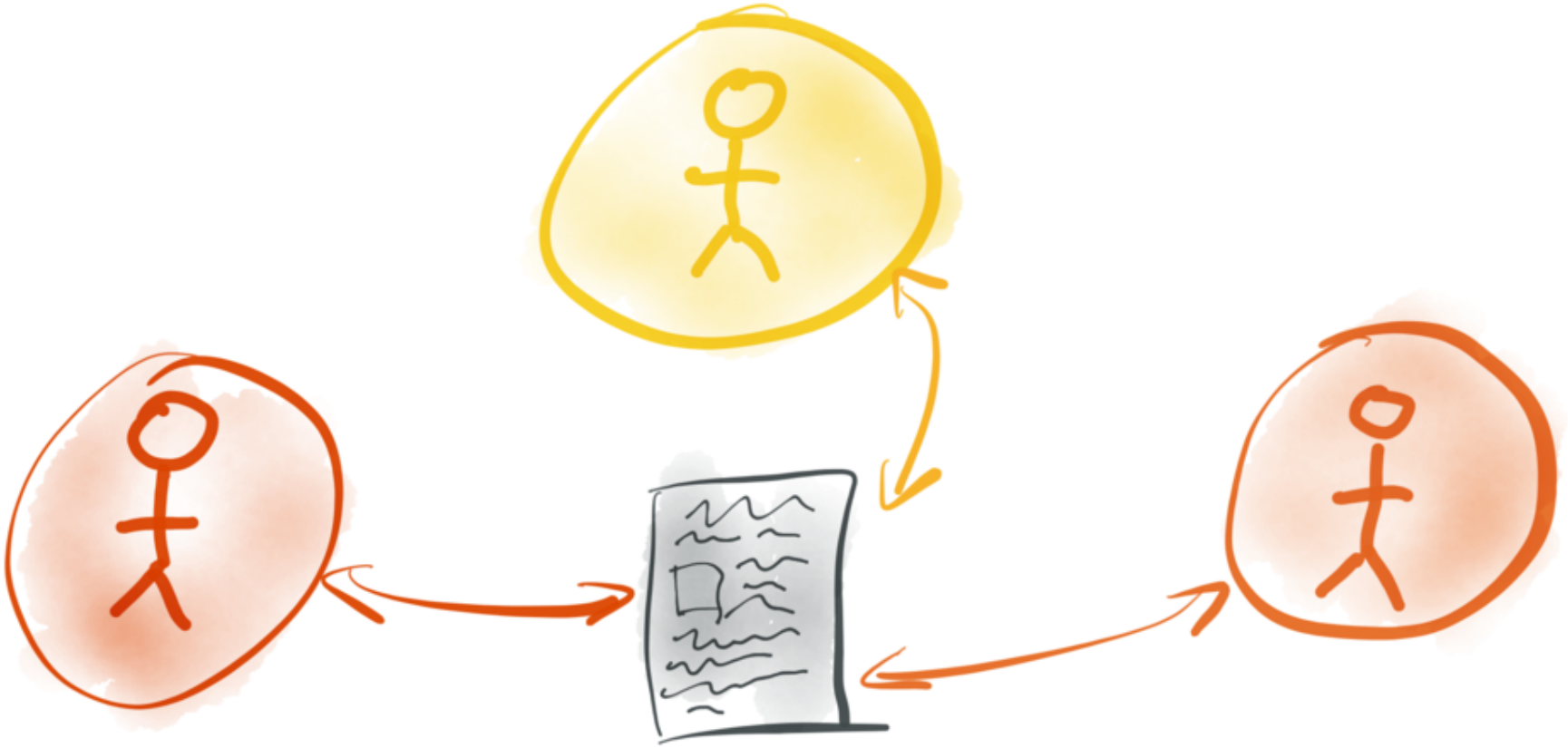
U. TY OF

.MBRIDGE

ON STRIKE TODAY

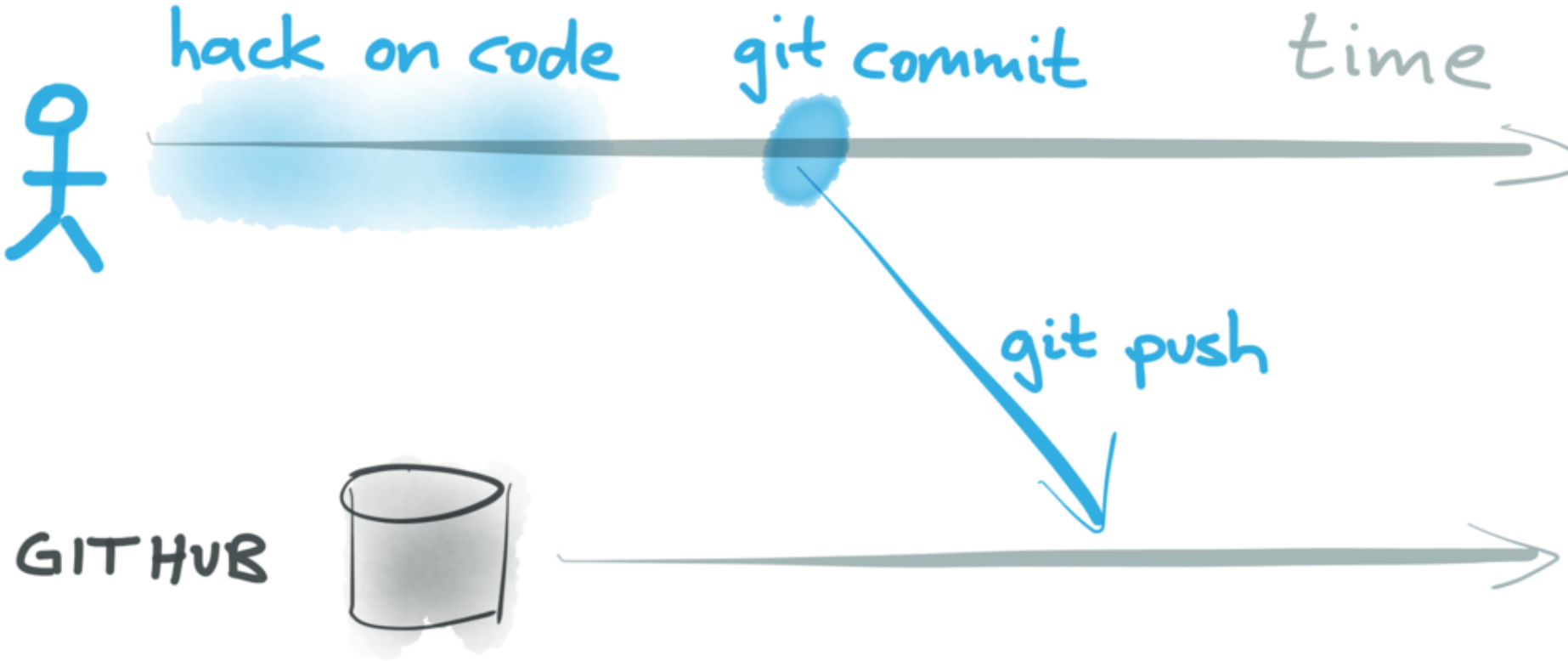
https://www.ucu.org.uk/why-we-are-taking-action-over-USS

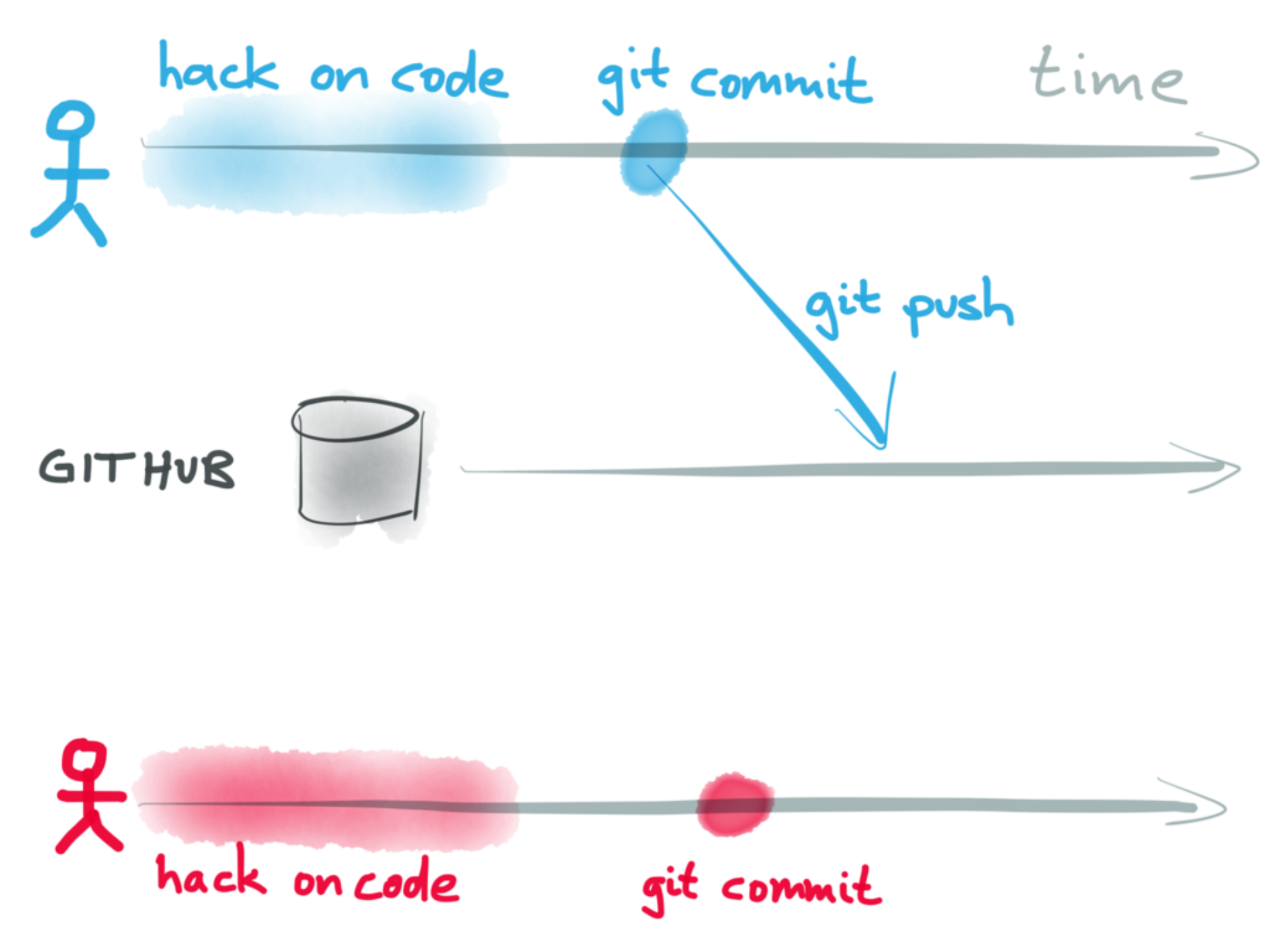
COLLABORATIVE APPLICATIONS

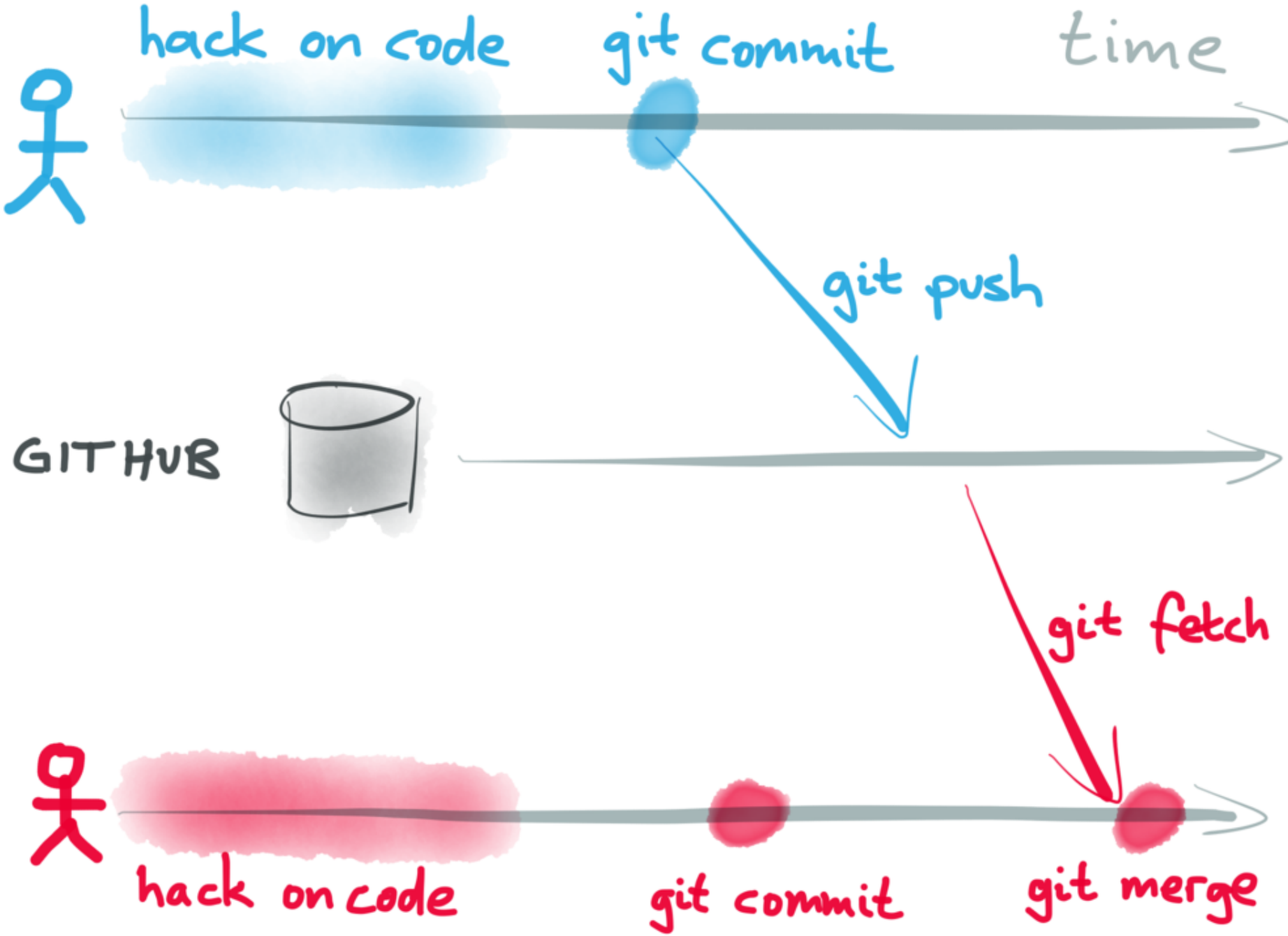



hack on code
time

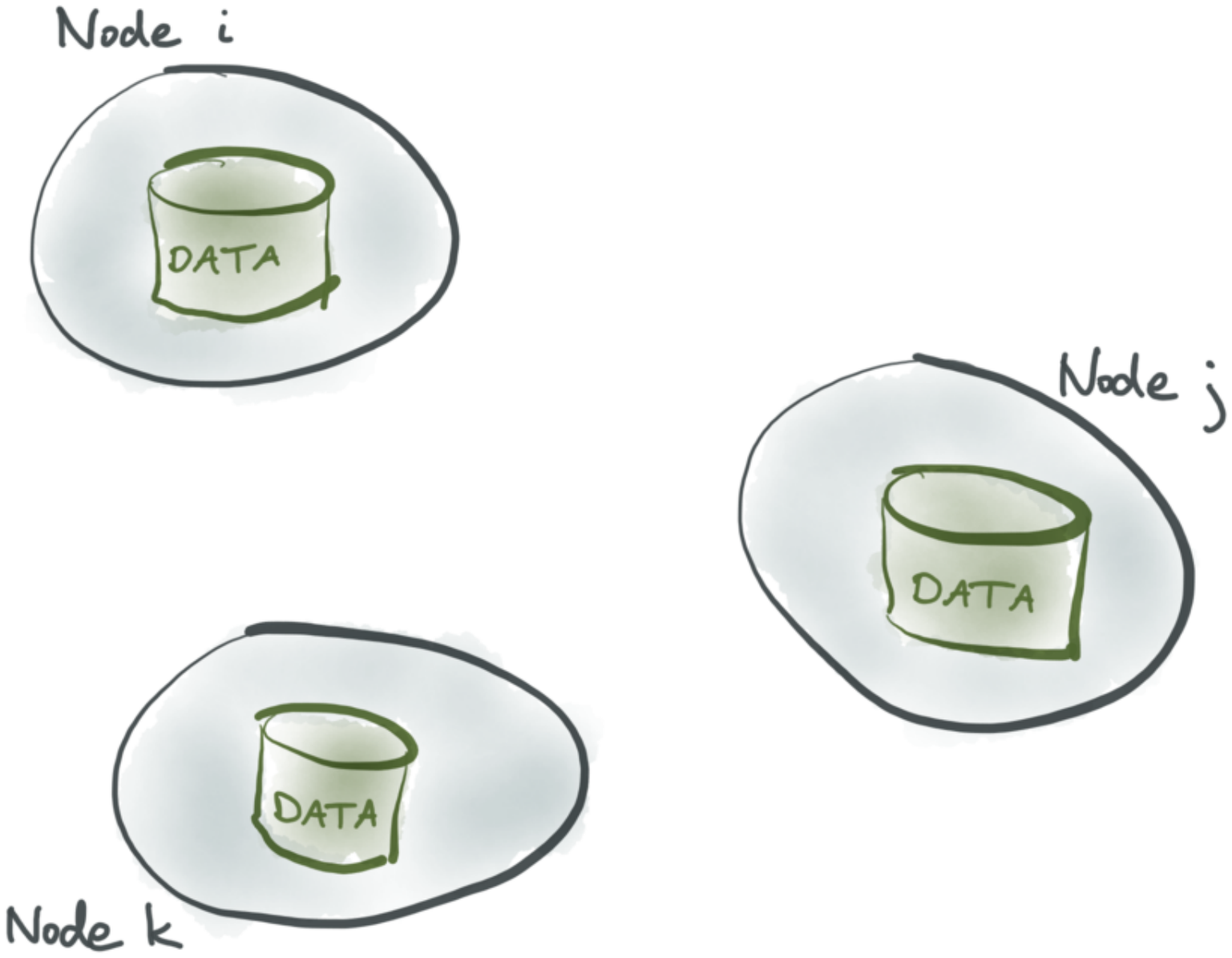

hack on code
git commit
time


hack on code
git commit
time
git push
GITHUB

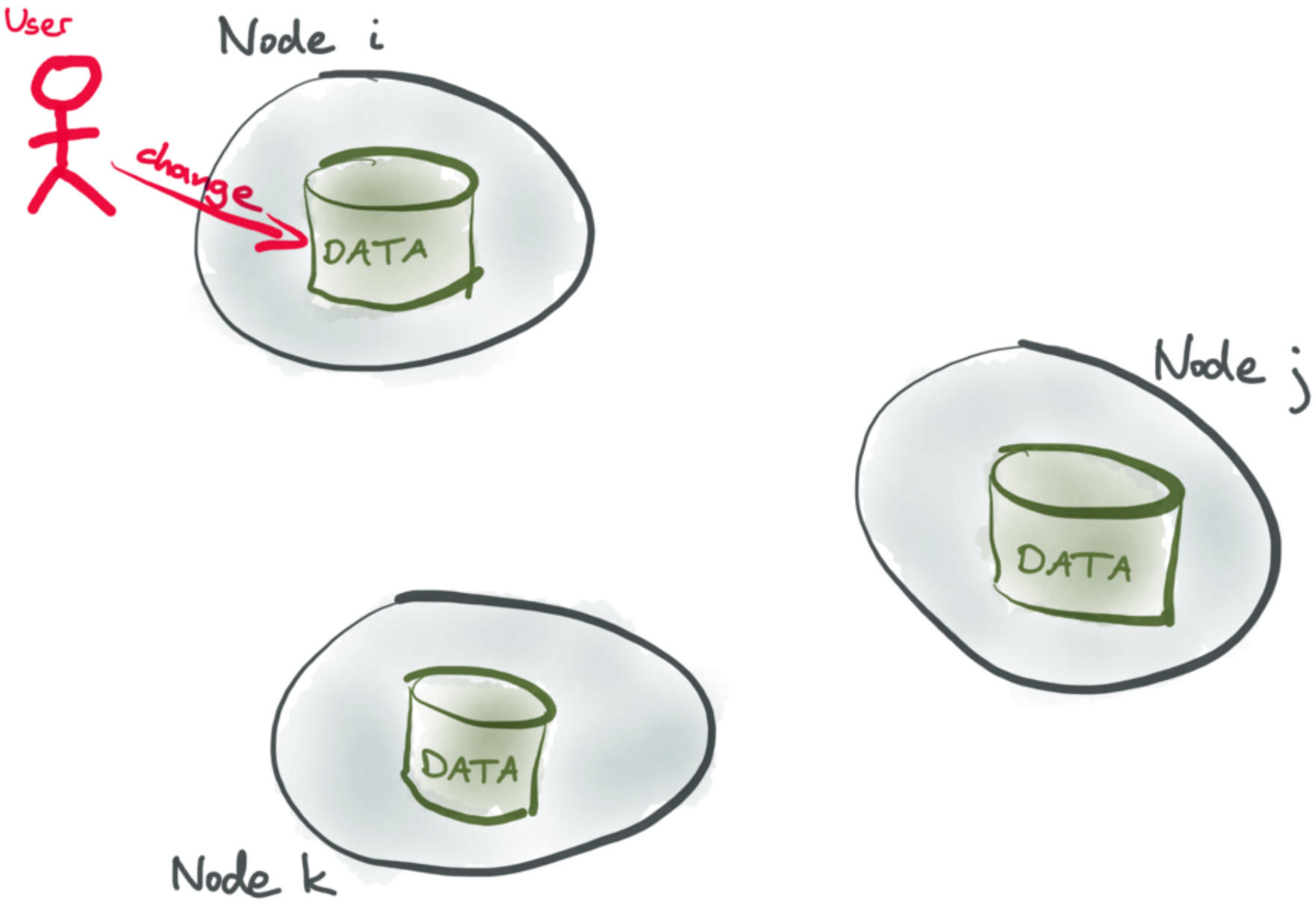

hack on code
git commit
time
git push
GITHUB
hack on code
git commit


hack on code
git commit
time
git push
GITHUB
git fetch
hack on code
git commit
git merge

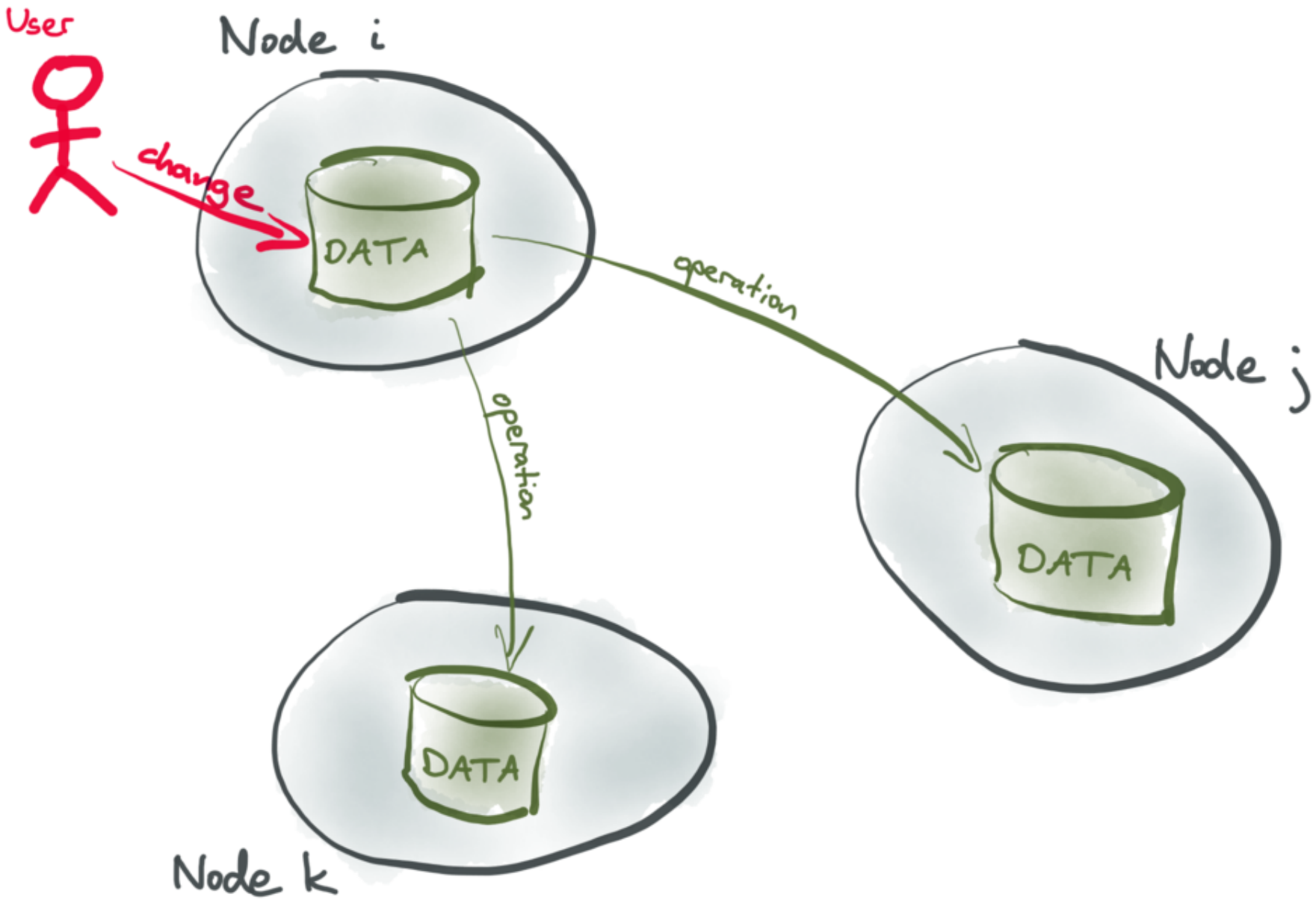
REPLICATION.



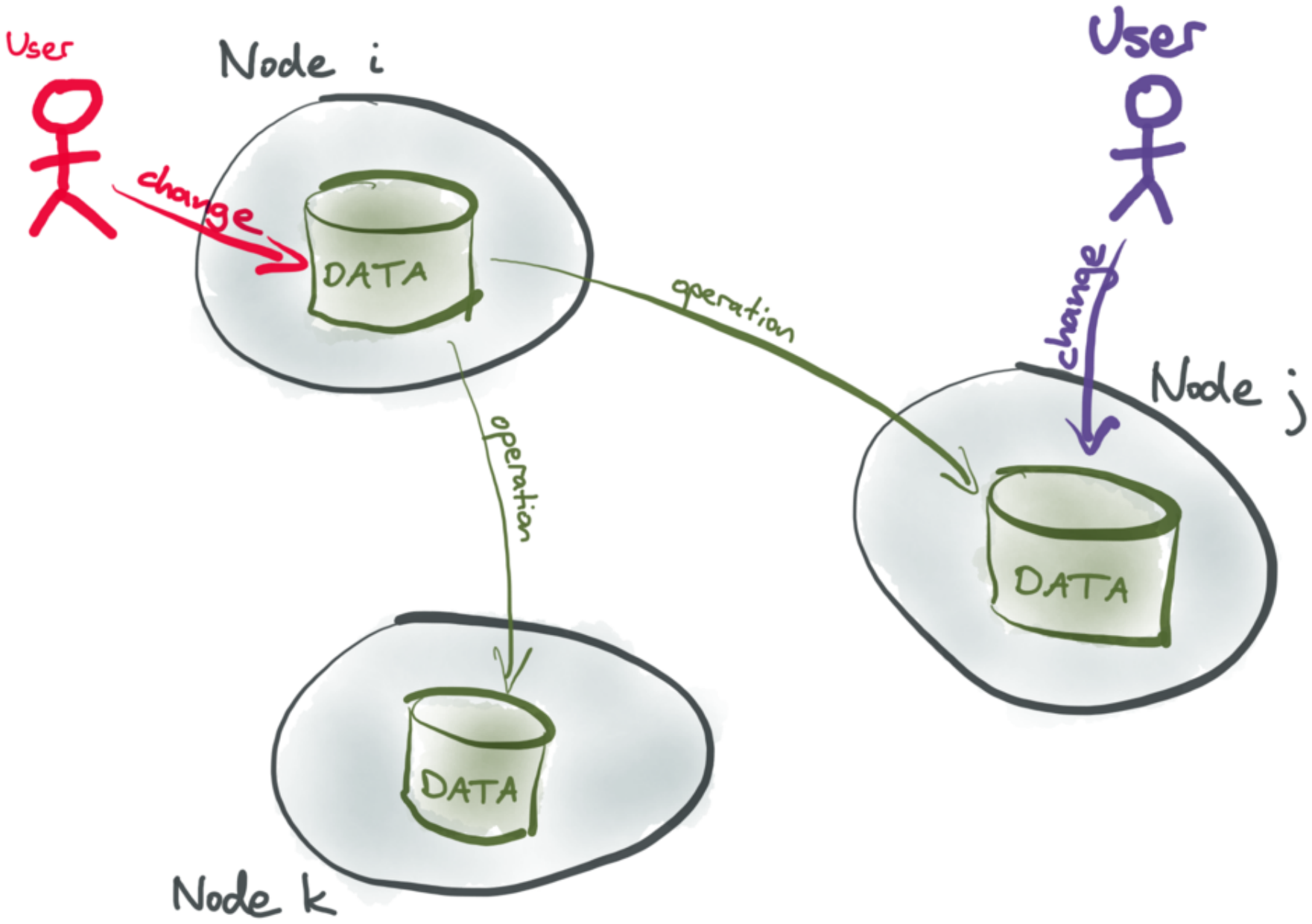
REPLICATION.



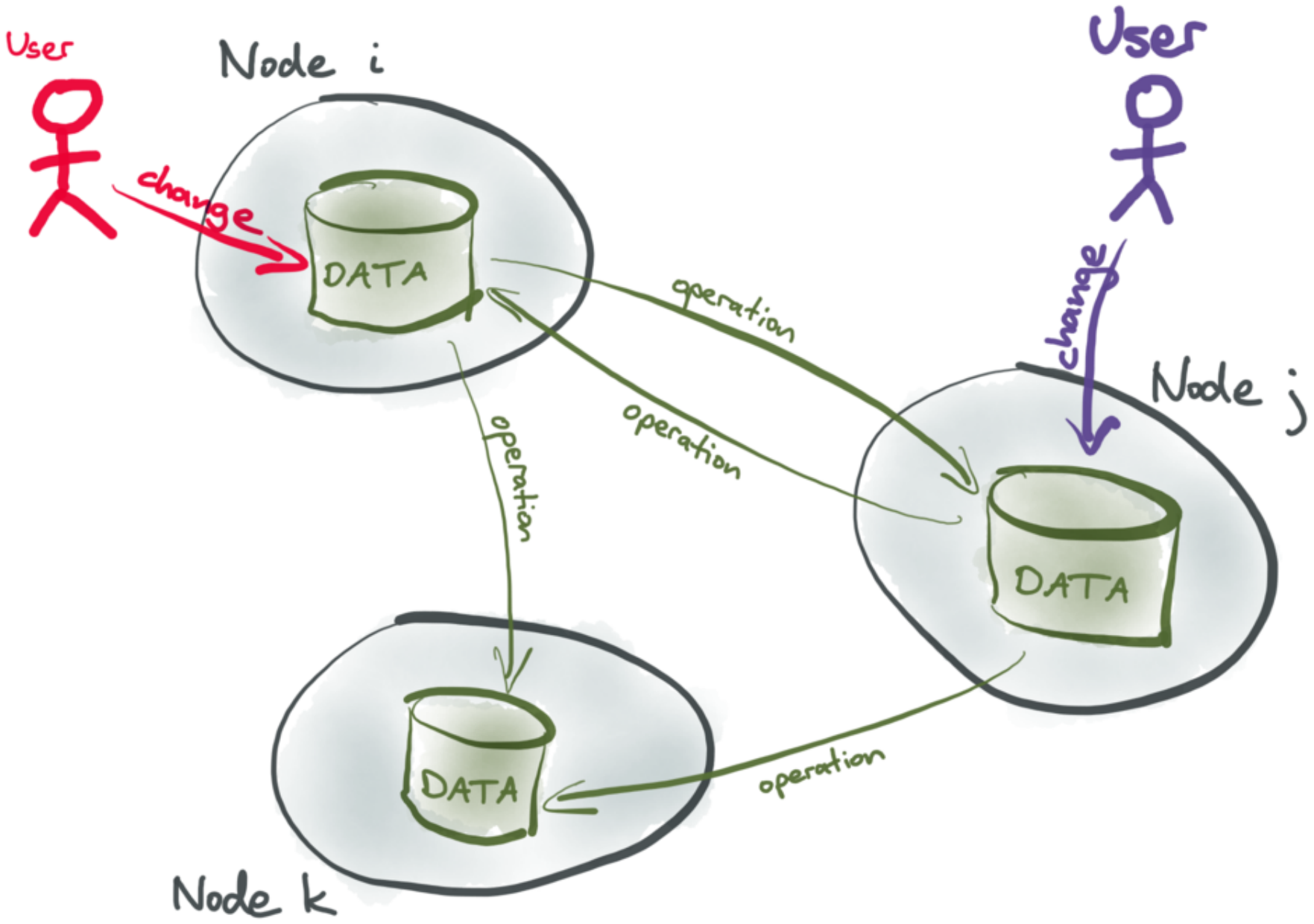
REPLICATION.



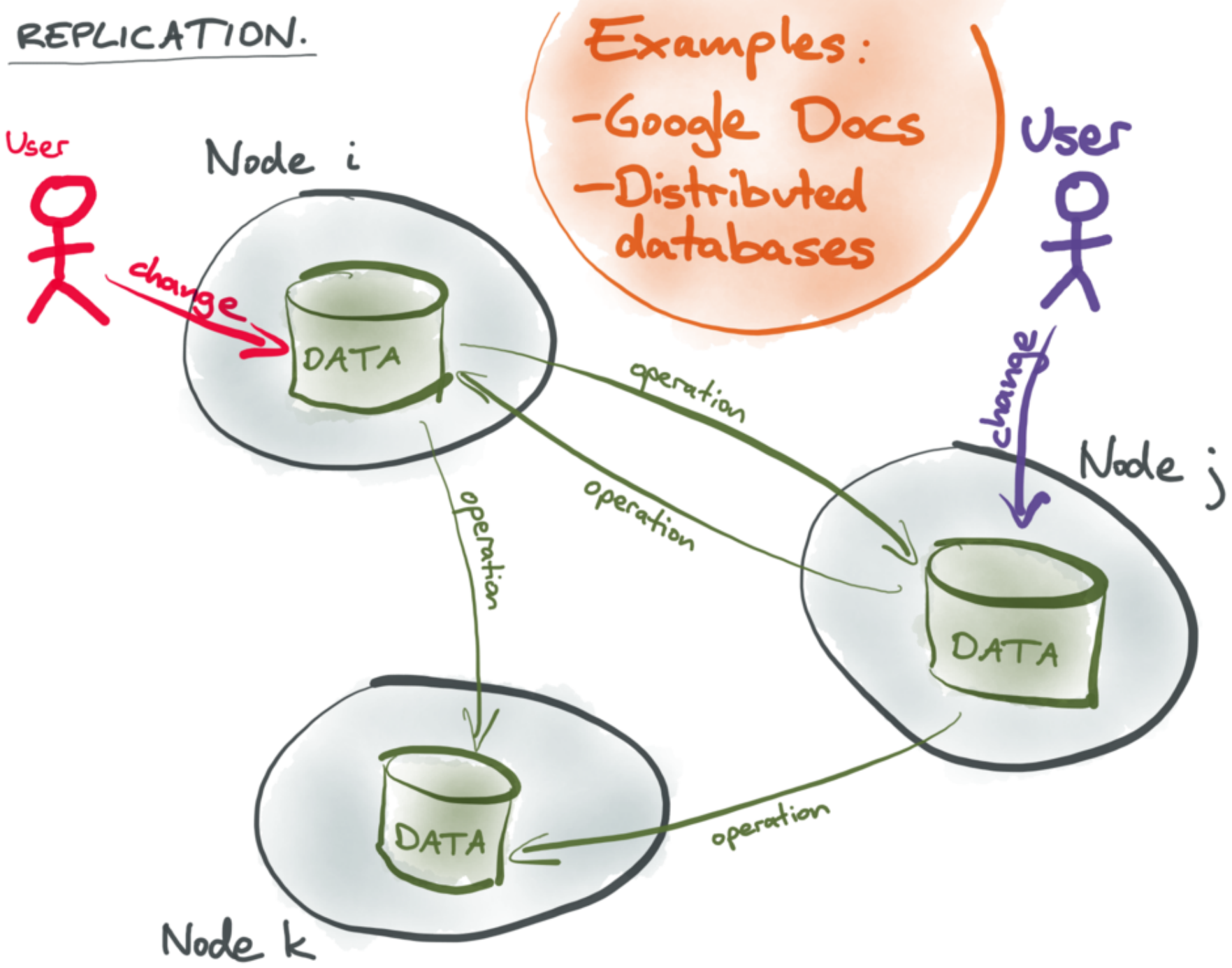
REPLICATION.

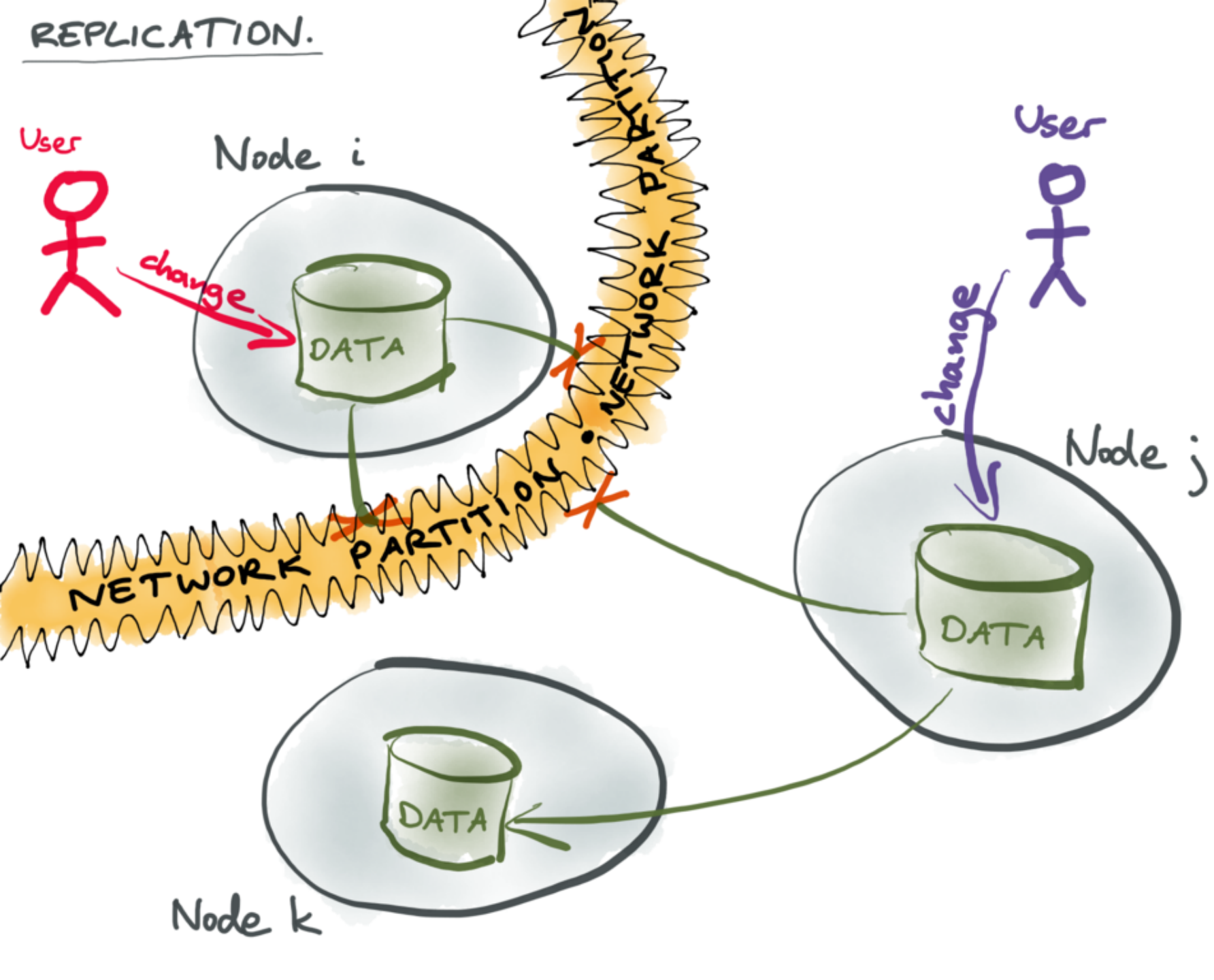


REPLICATION.



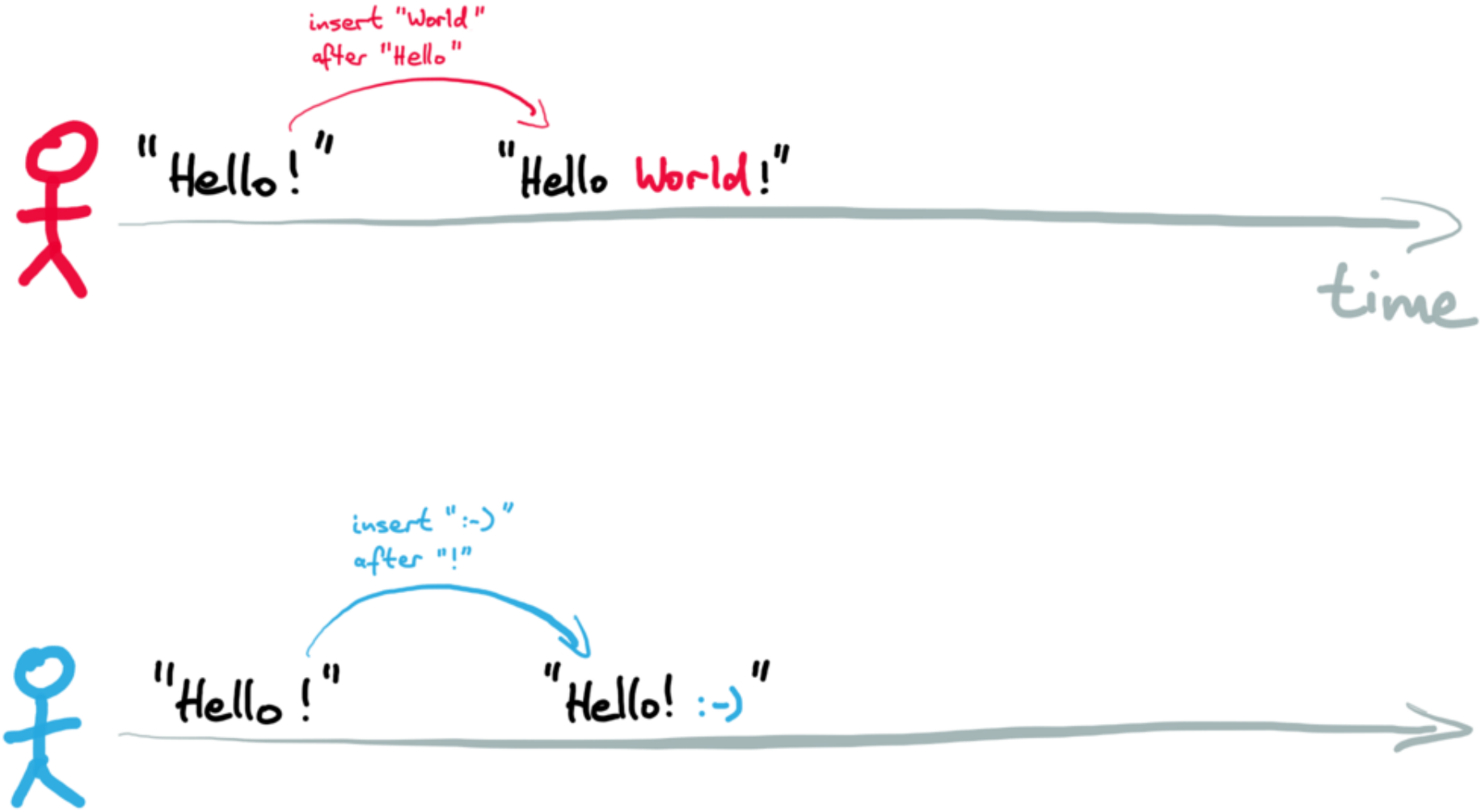
REPLICATION.



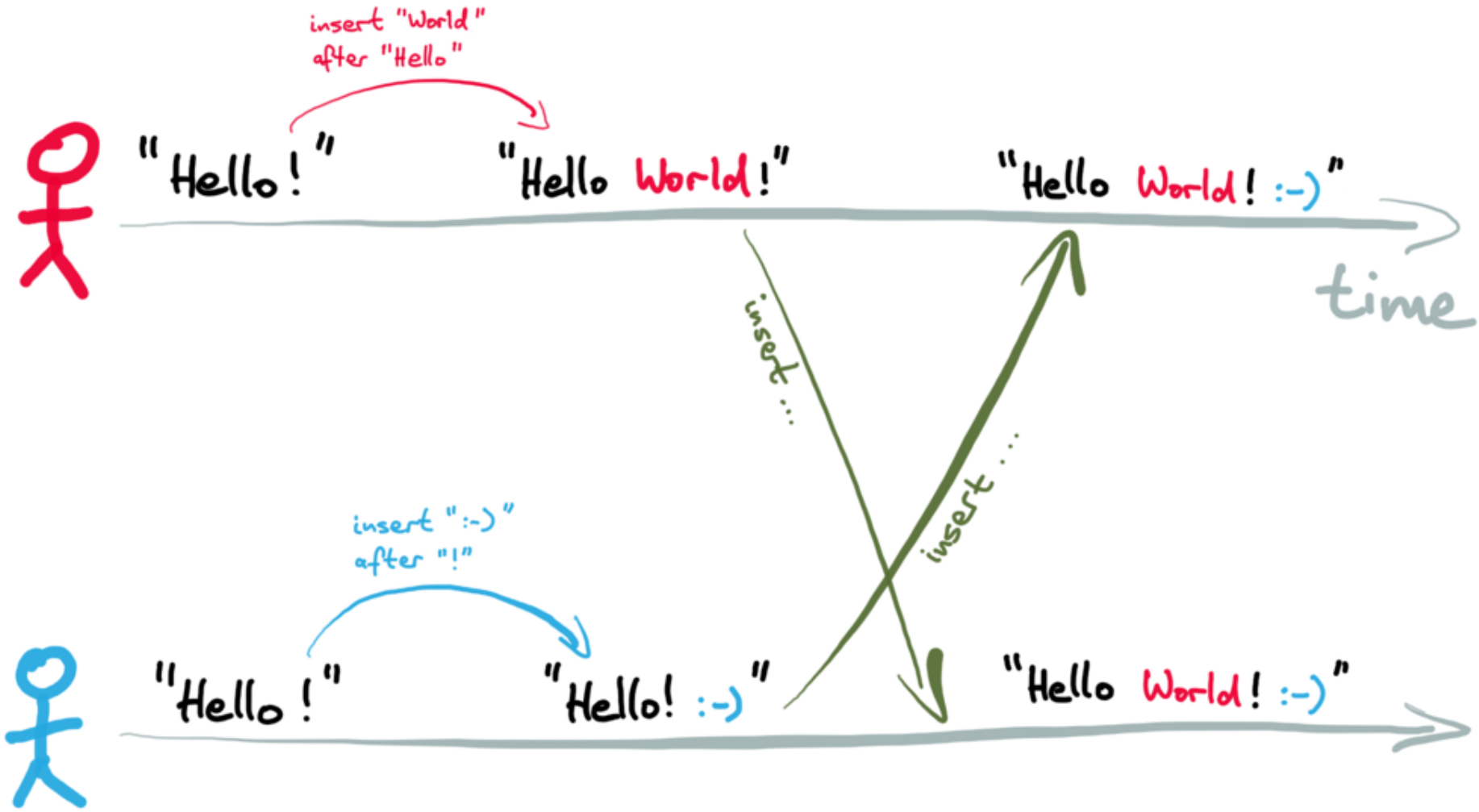

REPLICATION.
User
Node i
change
DATA
NETWORK PARTITION
NETWORK PARTITION
User
change
Node j
DATA
DATA
Node k

Example: Text editing

Example: Text editing

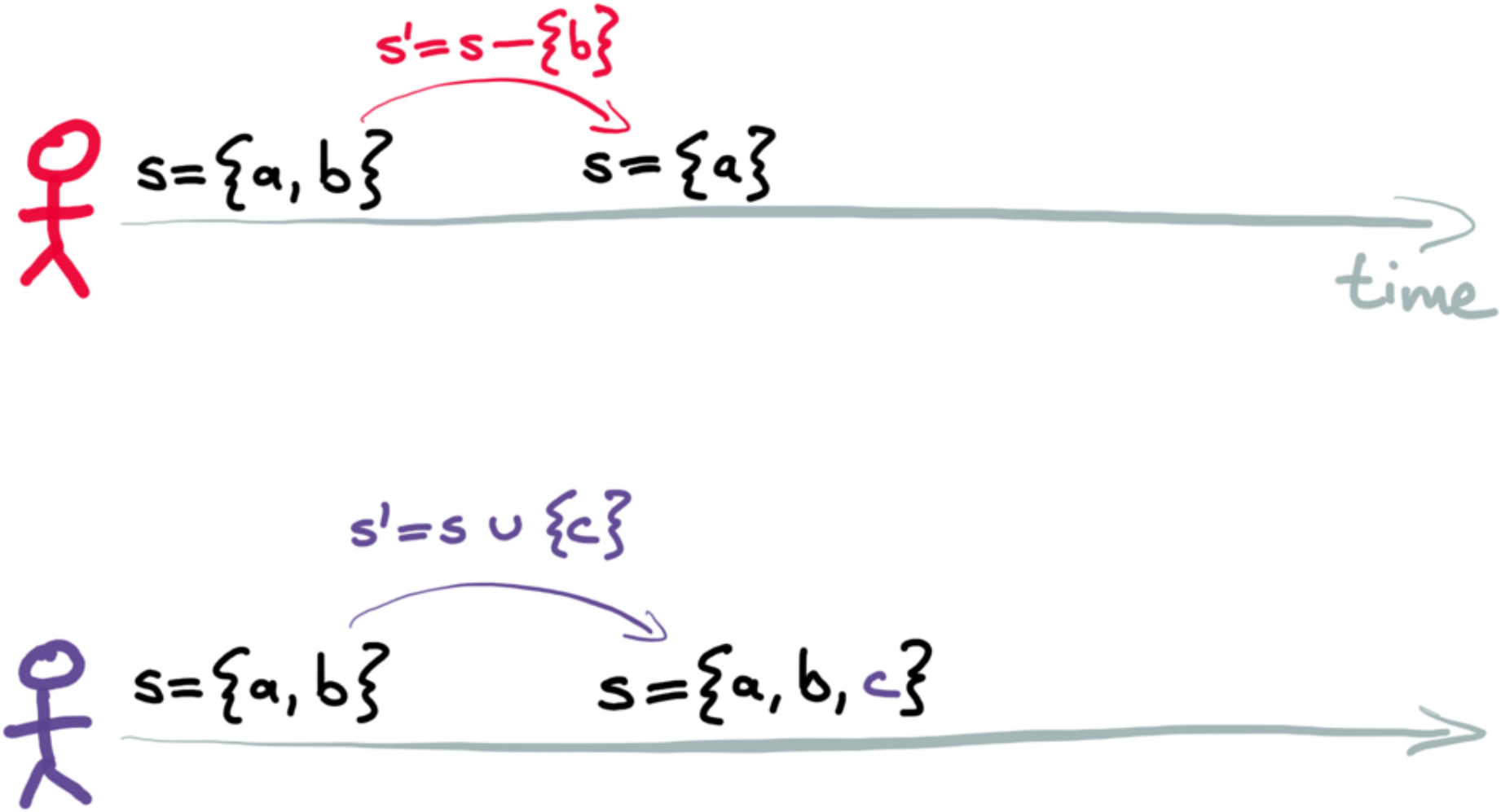


Example: Text editing

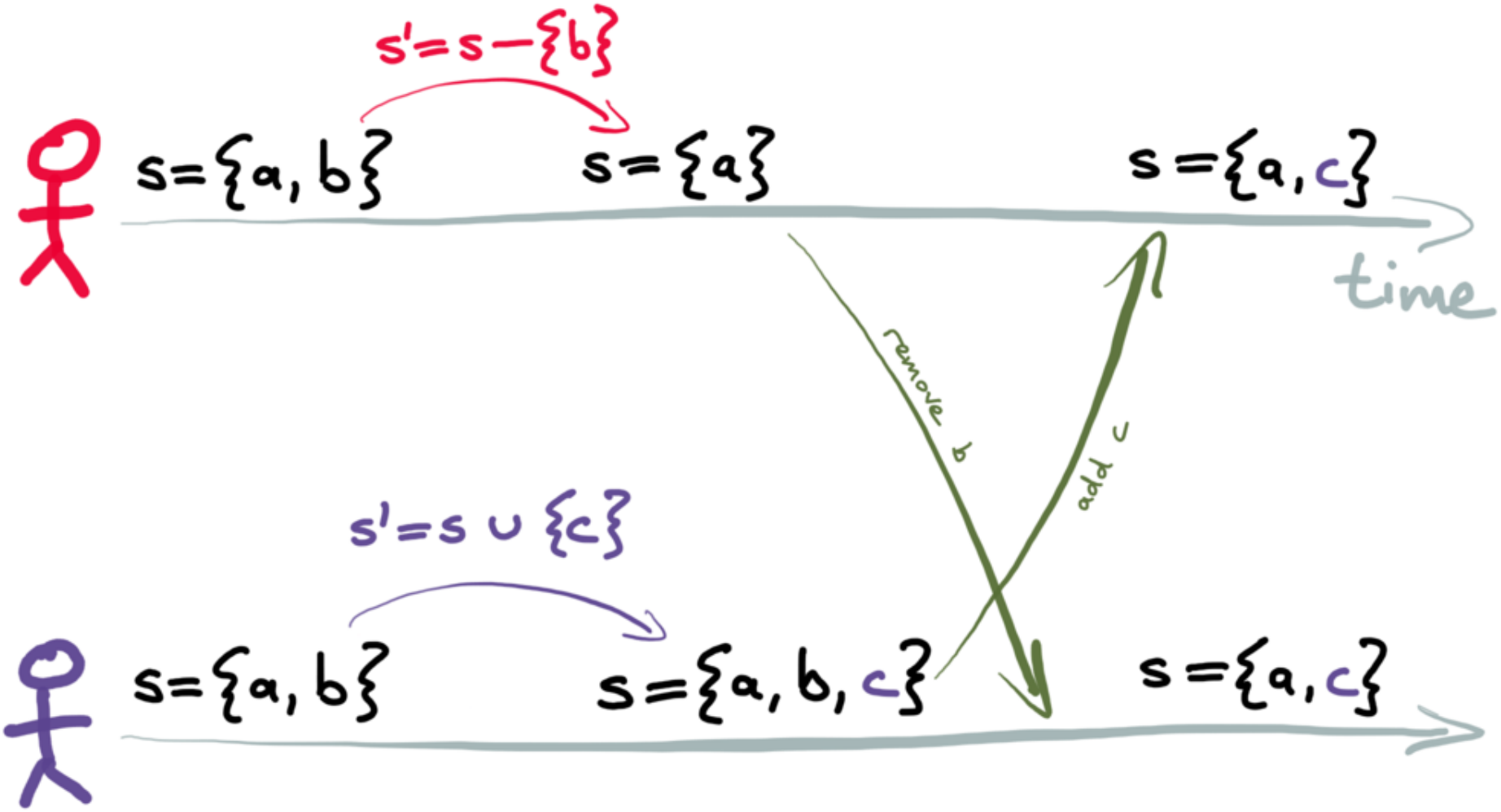


Example: set manipulation

Example: set manipulation

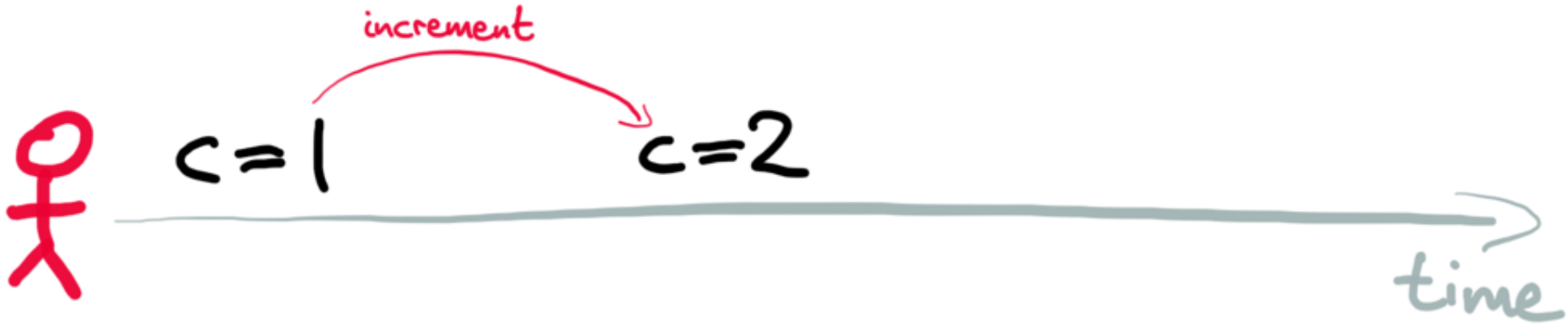


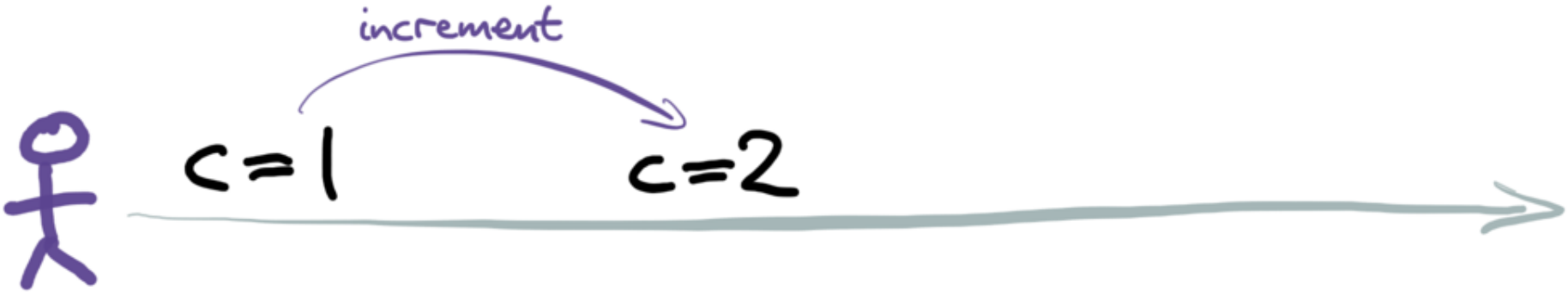
Example: set manipulation



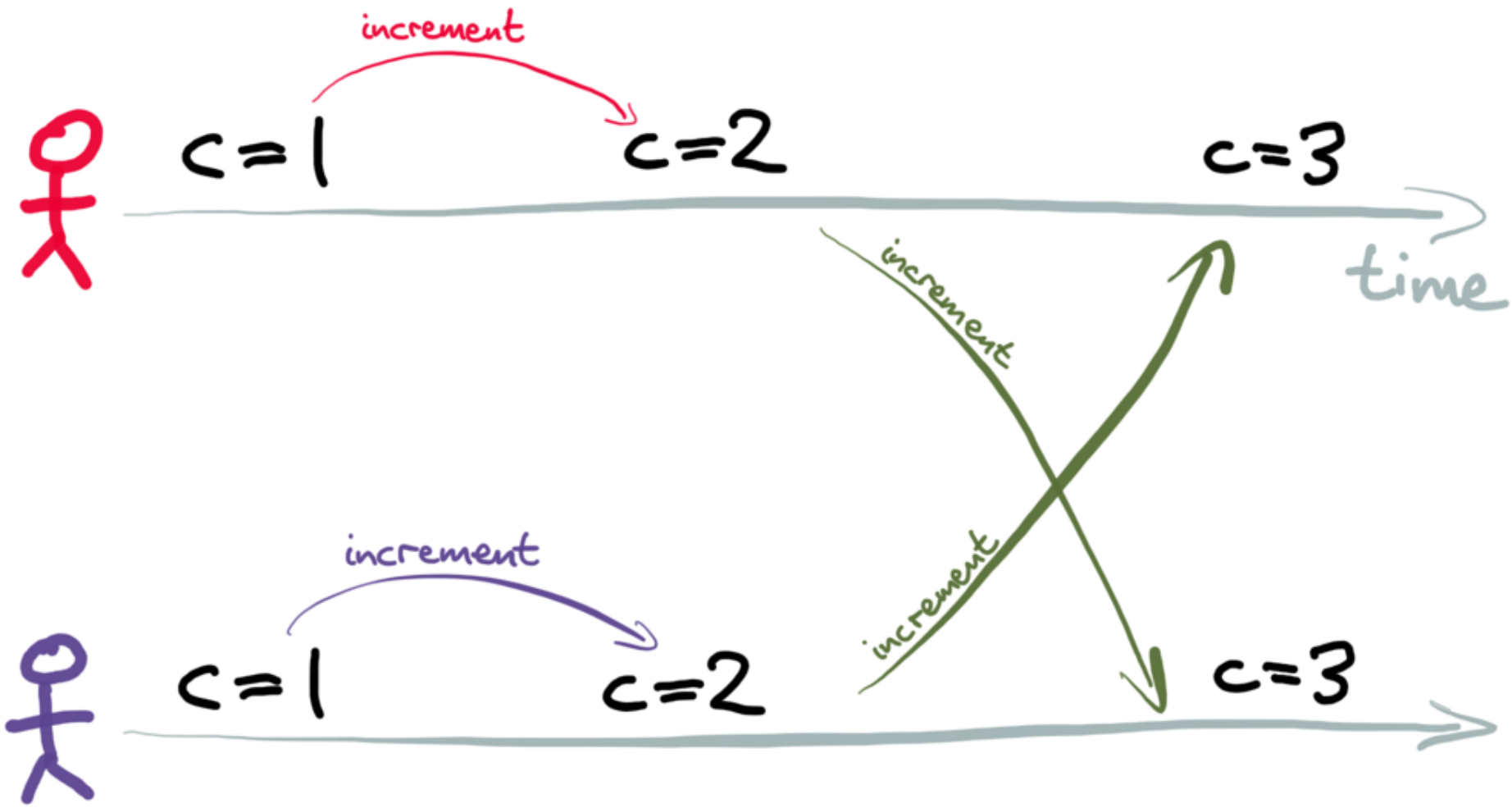
Example : counter

Example : counter





Example : counter



Algorithms for convergence

OPERATIONAL TRANSFORMATION (OT)

- e.g. Google Docs, MS Office Online

1989–2006

CONFLICT-FREE REPLICATED DATA TYPES (CRDTs)

- e.g. Riak, TomTom GPS, Teletype for Atom, ...

2006 – present

FAILURES OF OPERATIONAL TRANSFORM

dOPT
Ellis & Gibbs
1989

adOPTed
Ressel et al.
1996

IMOR
Imine et al.
2003

Jupiter
Nichols et al.
1995

SOCT2
Suleiman et al.
1997

SDT
Li & Li
2004

SOCT3/4
Vidot et al.
2000

TTF
Oster et al.
2006

FAILURES OF OPERATIONAL TRANSFORM

dOPT
~~Ellis & Gibbs~~
1989
Wrong

adOPTed
Ressel et al.
1996

IMOR
Imine et al.
2003

Jupiter
Nichols et al.
1995

SOCT2
Suleiman et al.
1997

SDT
Li & Li
2004

SOCT3/4
Vidot et al.
2000

TTF
Oster et al.
2006

FAILURES OF OPERATIONAL TRANSFORM

~~dOPT~~
~~Ellis & Gibbs~~
1989
Wrong

~~adOPTed~~
~~Ressel et al.~~
1996
Wrong

IMOR
Imine et al.
2003

Jupiter
Nichols et al.
1995

SOCT2
Suleiman et al.
1997

SDT
Li & Li
2004

SOCT3/4
Vidot et al.
2000

TTF
Oster et al.
2006

FAILURES OF OPERATIONAL TRANSFORM

dOPT
Ellis & Gibbs
1989
Wrong

adOPTed
Ressel et al.
1996
Wrong

IMOR
Imine et al.
2003

Jupiter
Nichols et al.
1995

SOCT2
Suleiman et al.
1997
Wrong

SDT
Li & Li
2004

SOCT3/4
Vidot et al.
2000

TTF
Oster et al.
2006

FAILURES OF OPERATIONAL TRANSFORM

dOPT
Ellis & Gibbs
1989
Wrong

adOPTed
Ressel et al.
1996
Wrong

IMOR
Imine et al.
2003
Wrong

Jupiter
Nichols et al.
1995

SOCT2
Suleiman et al.
1997
Wrong

SDT
Li & Li
2004

SOCT3/4
Vidot et al.
2000

TTF
Oster et al.
2006

FAILURES OF OPERATIONAL TRANSFORM

dOPT
Ellis & Gibbs
1989
Wrong

adOPTed
Ressel et al.
1996
Wrong

IMOR
Imine et al.
2003
Wrong

Jupiter
Nichols et al.
1995

SOCT2
Suleiman et al.
1997
Wrong

SDT
Li & Li
2006
Wrong

SOCT3/4
Vidot et al.
2000

TTF
Oster et al.
2006

FAILURES OF OPERATIONAL TRANSFORM

dOPT
Ellis & Gibbs
1989
Wrong

adOPTed
Ressel et al.
1996
Wrong

IMOR
Imine et al.
2003
Wrong

Jupiter
Nichols et al.
1995
require central server

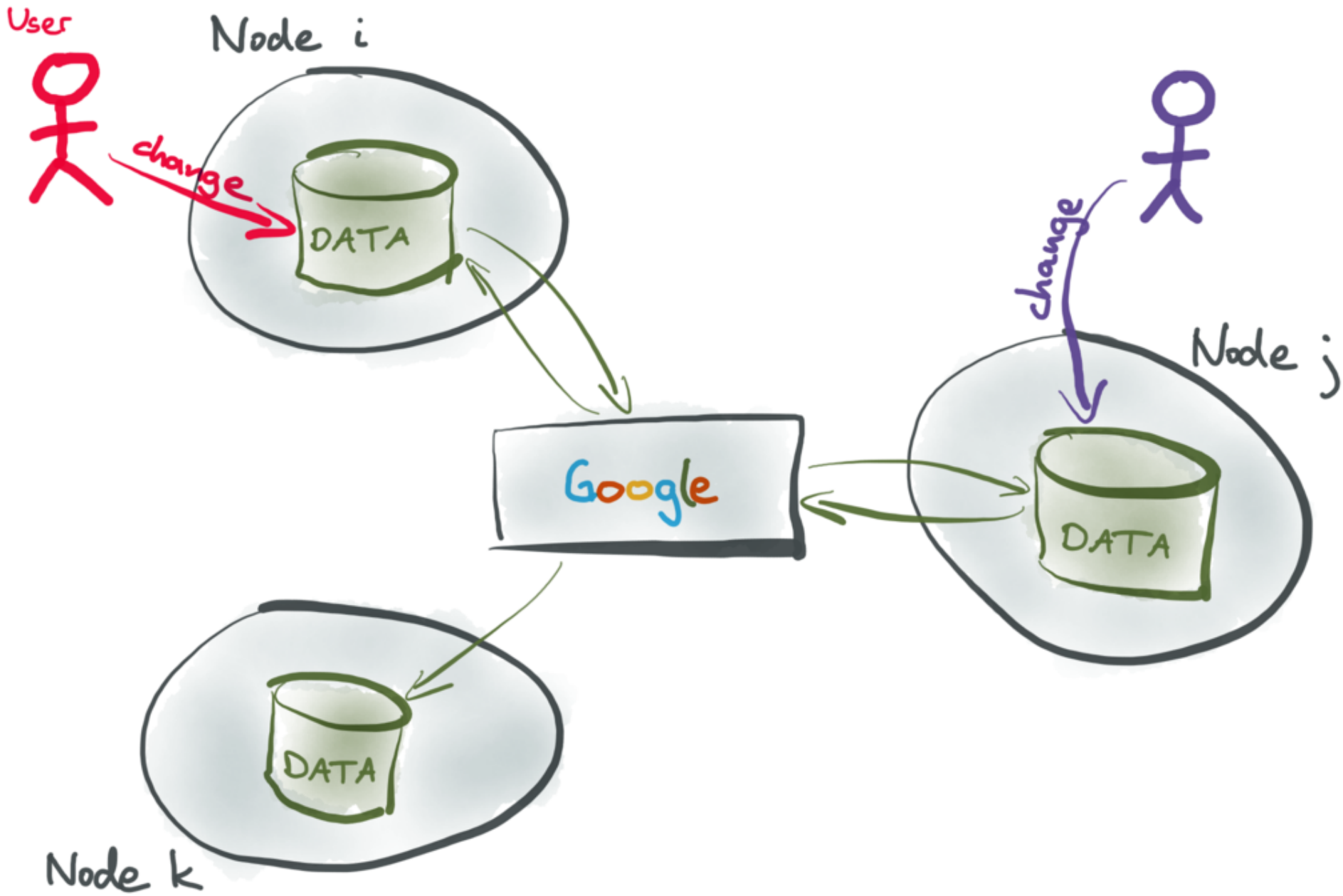
SOCT2
Suleiman et al.
1997
Wrong

SDT
Li & Li
2006
Wrong

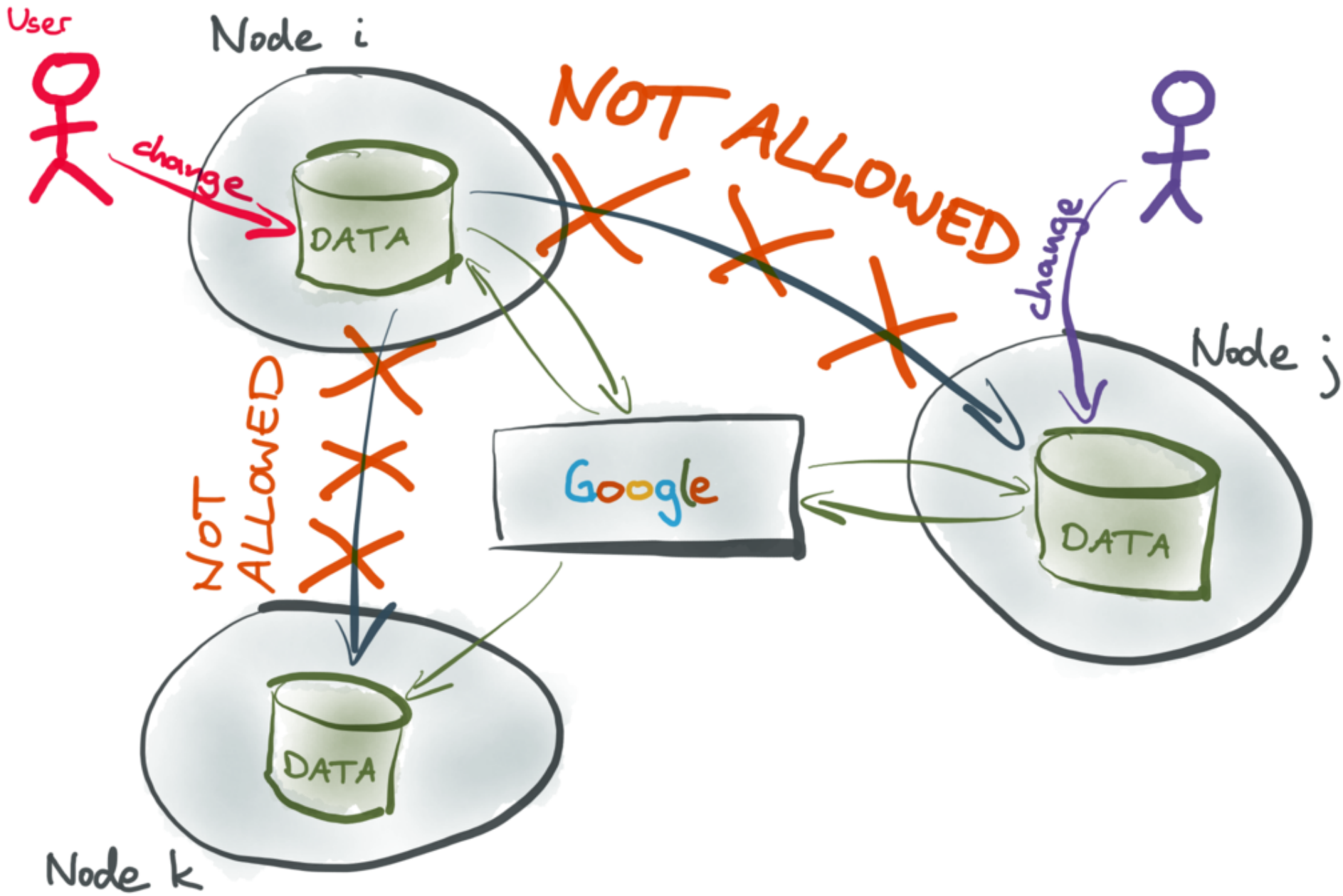
SOCT3/4
Vidot et al.
2000

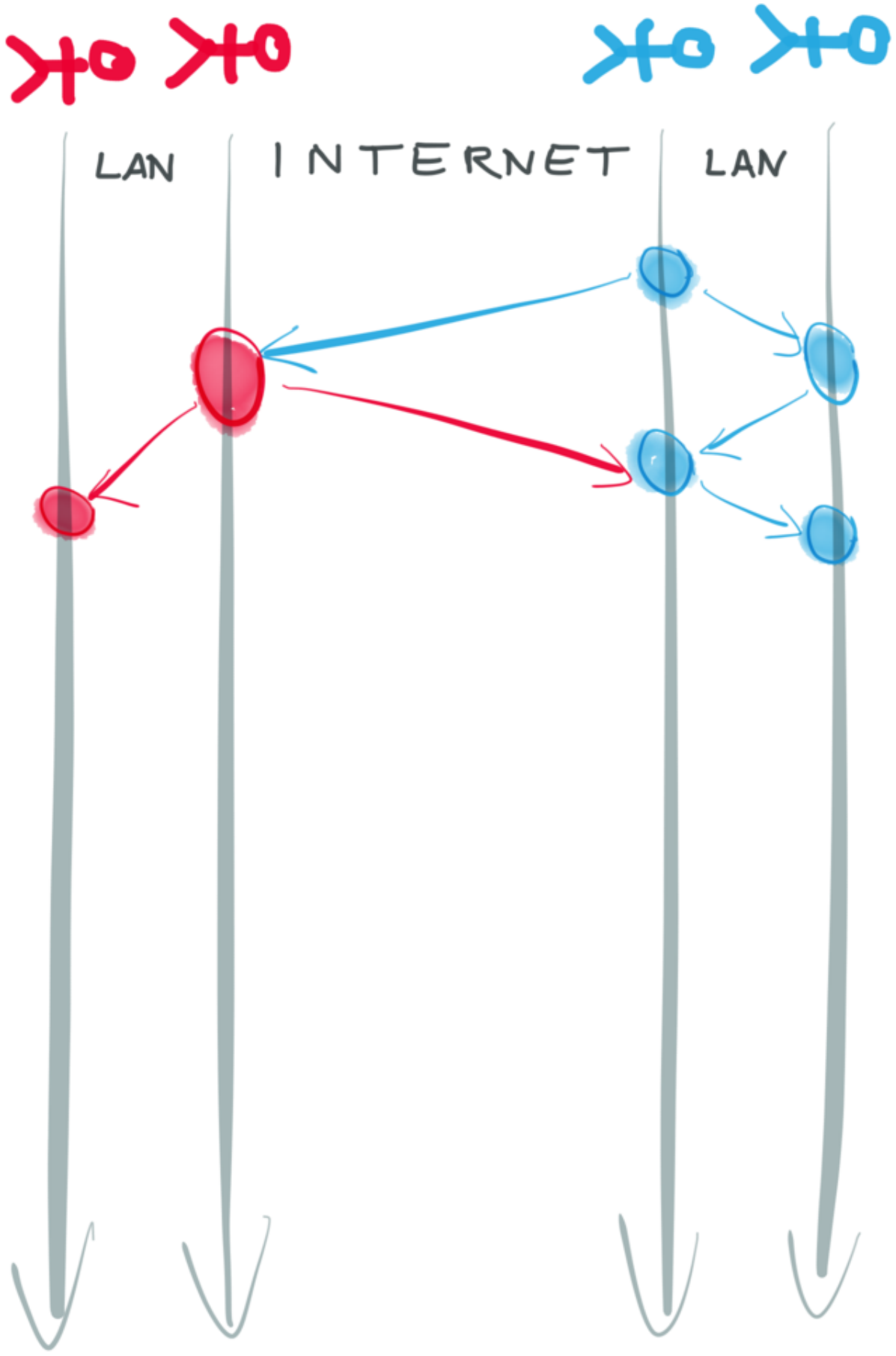
TTF
Oster et al.
2006

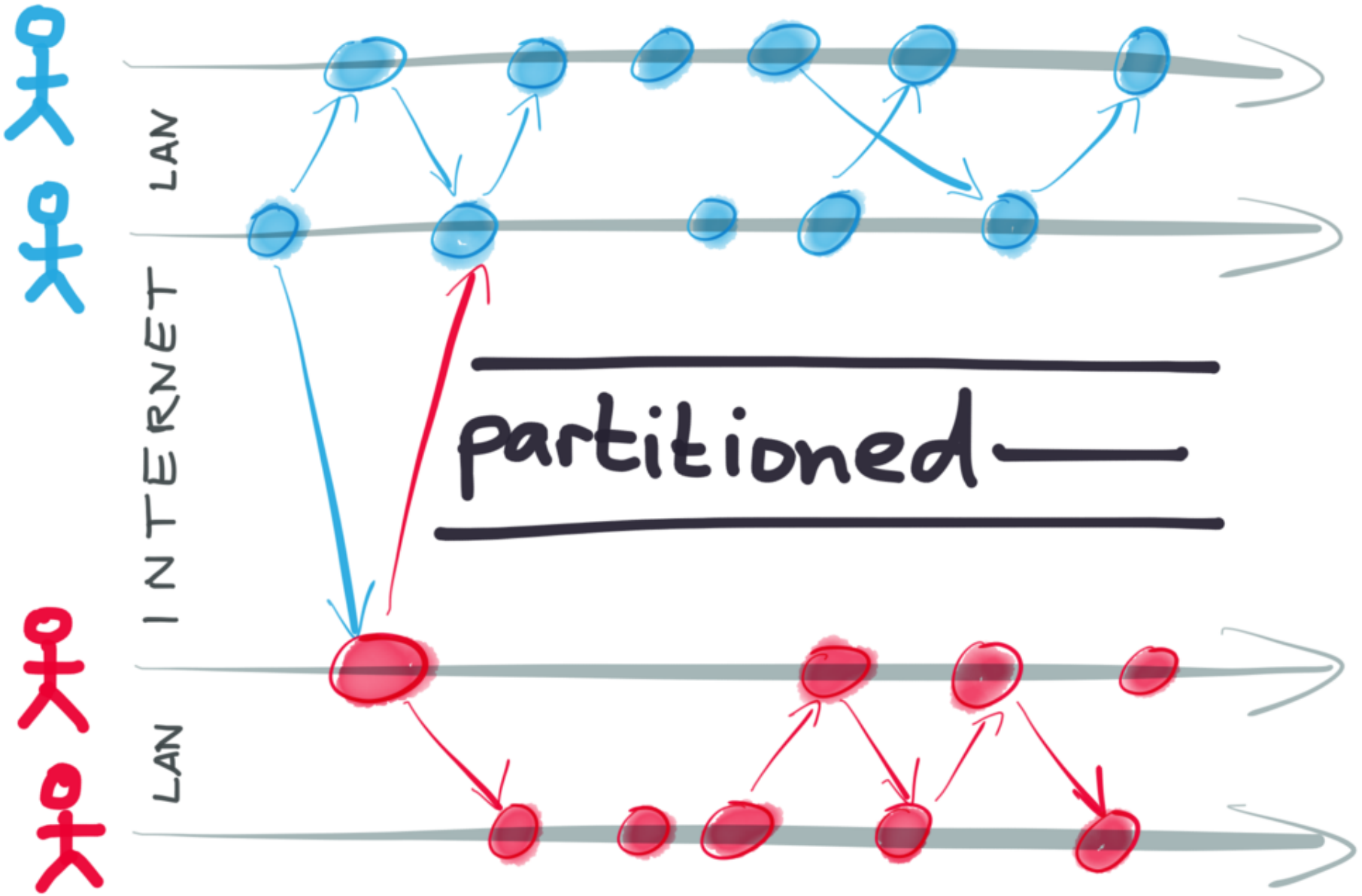
OPERATIONAL TRANSFORMATION IN GOOGLE DOCS.

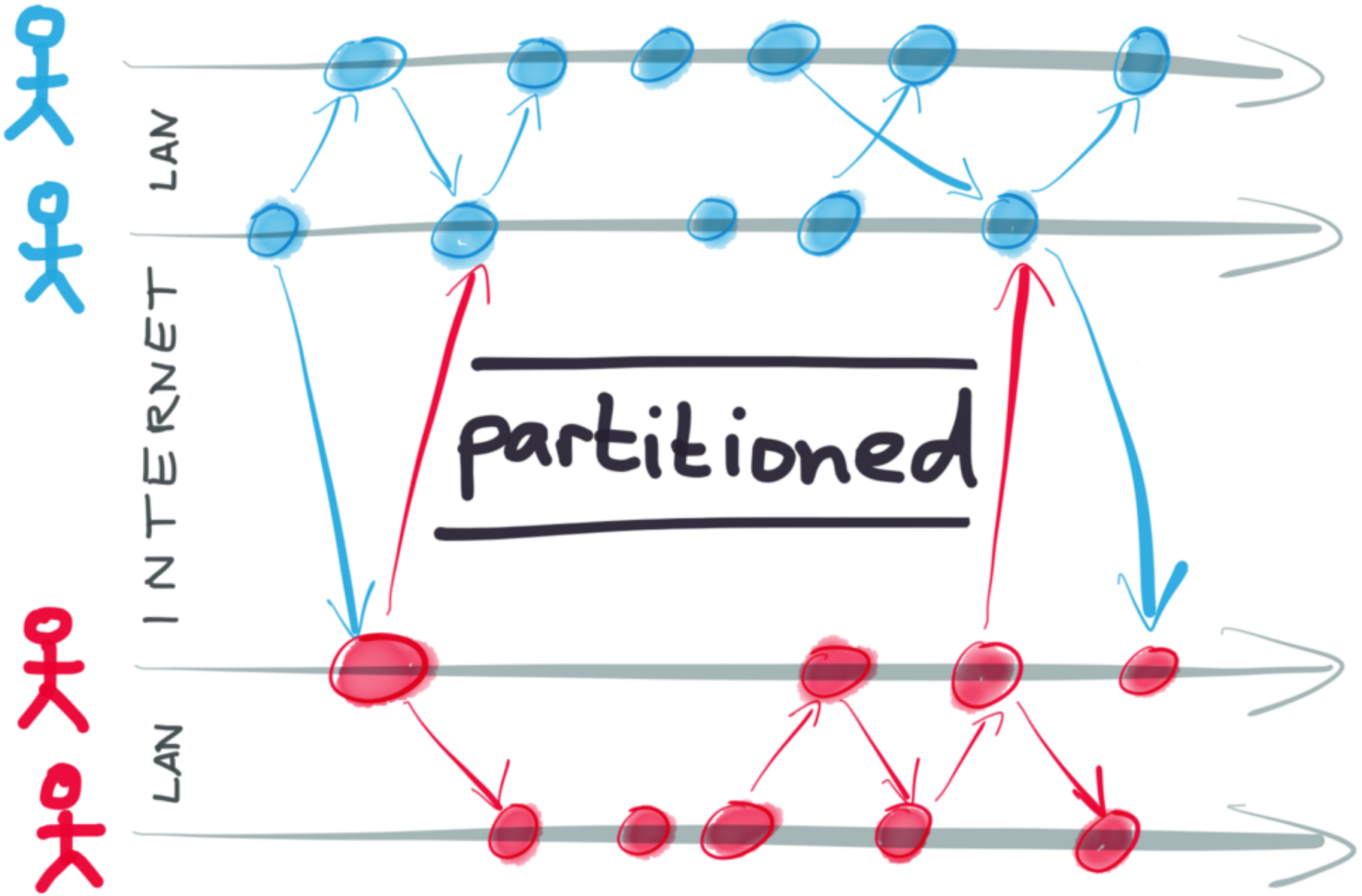


OPERATIONAL TRANSFORMATION IN GOOGLE DOCS.

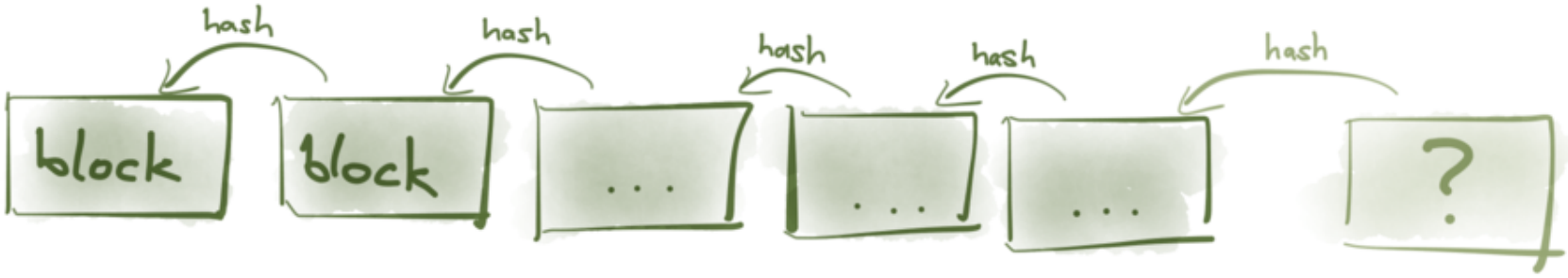



LAN
INTERNET
LAN

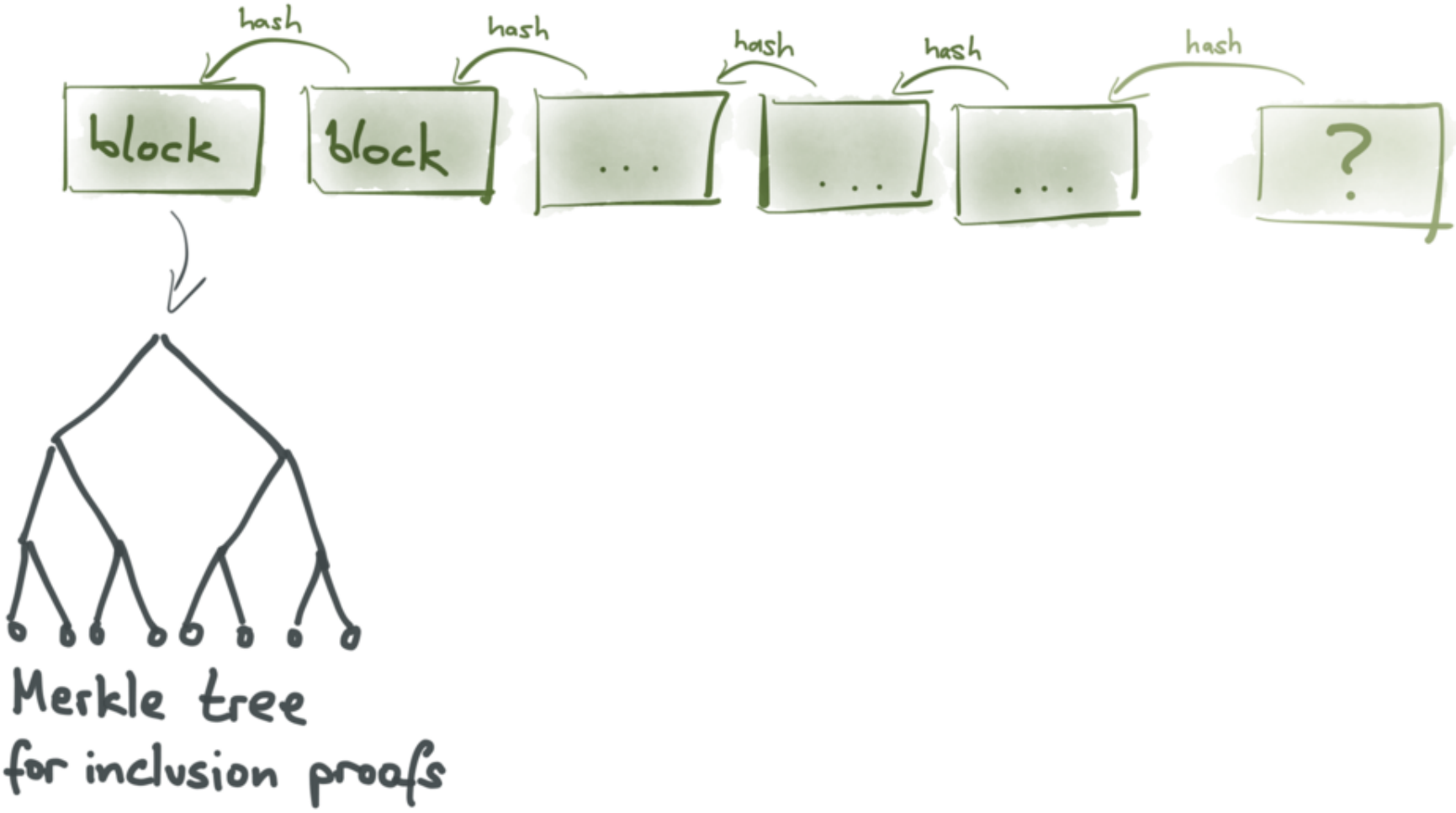

LAN
INTERNET
partitioned
LAN


LAN
INTERNET
partitioned
LAN

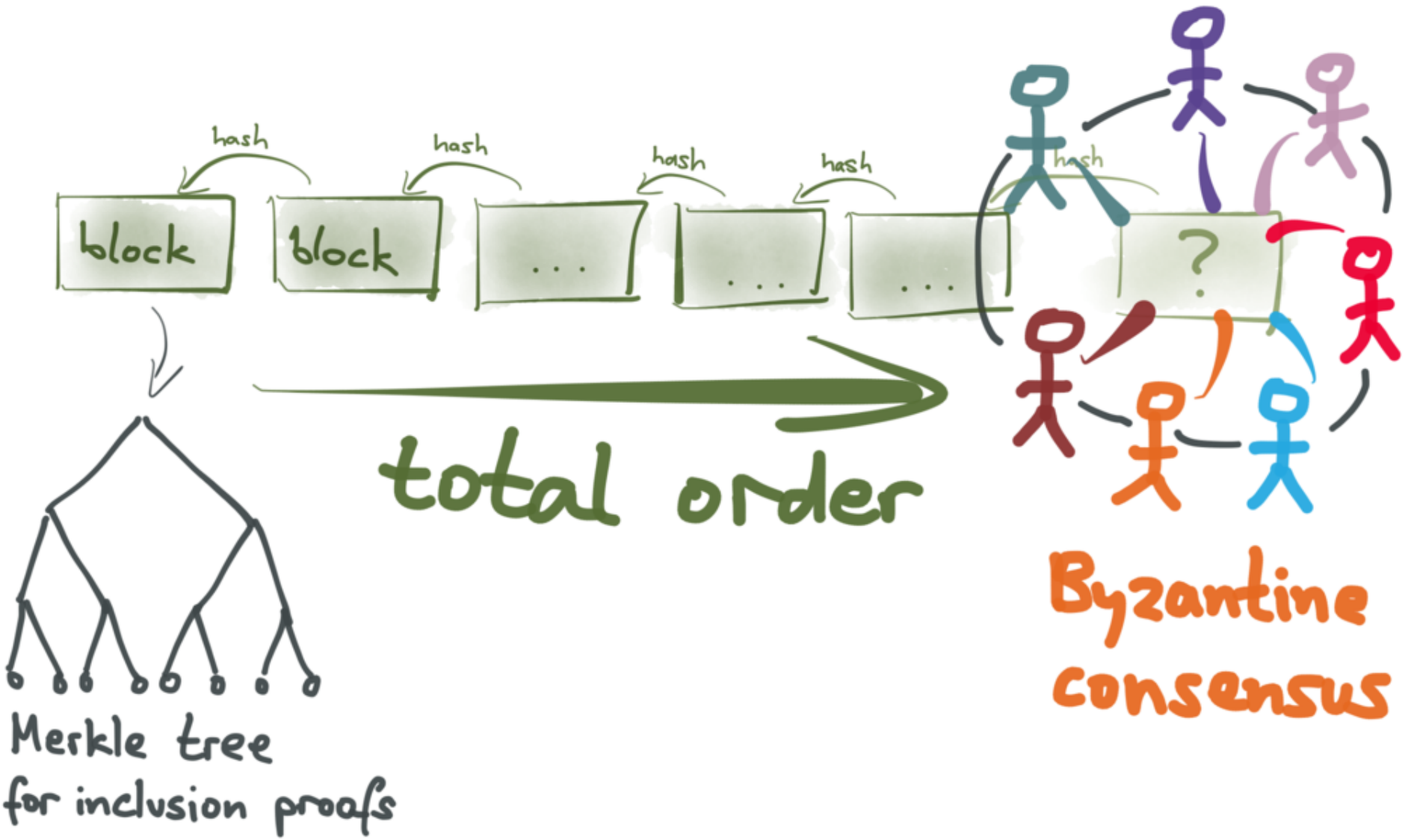
DECENTRALIZATION = BLOCKCHAINS?



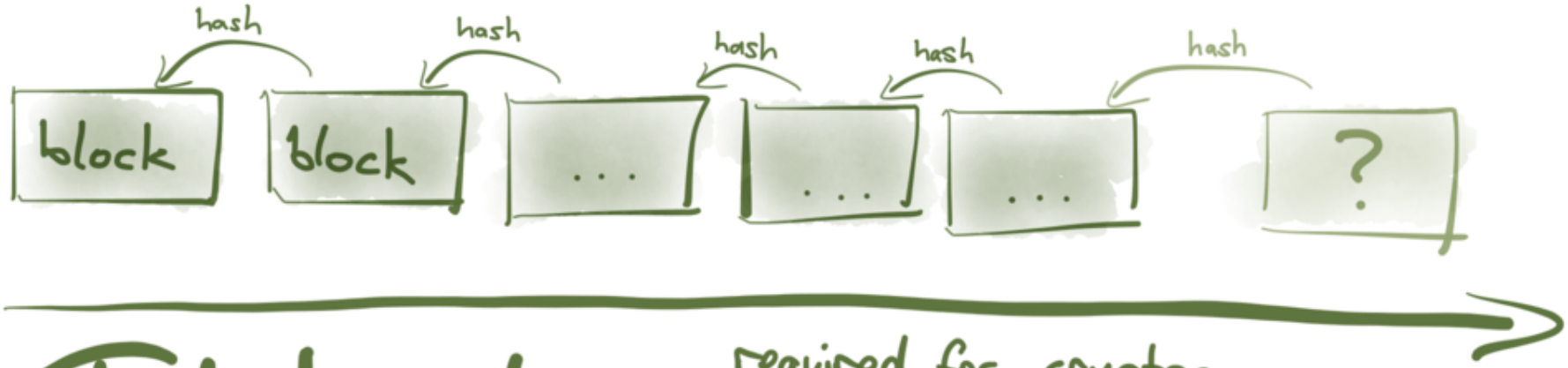
DECENTRALIZATION = BLOCKCHAINS?



DECENTRALIZATION = BLOCKCHAINS?



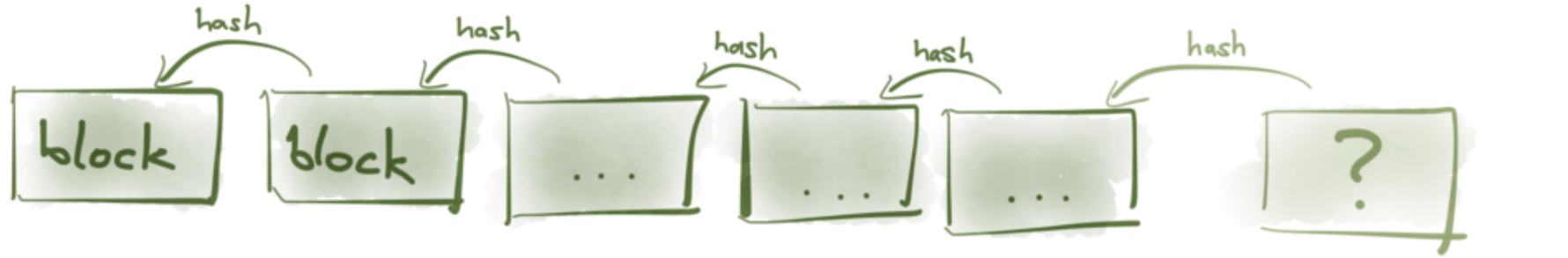
DECENTRALIZATION = BLOCKCHAINS?



Total order

required for cryptocurrencies
(to prevent double-spending)

DECENTRALIZATION = BLOCKCHAINS?



Total order required for cryptocurrencies (to prevent double-spending)

Consensus = pick one of several proposed values

Collaboration = keep all edits and merge them

Algorithms for convergence

OPERATIONAL TRANSFORMATION (OT)

- e.g. Google Docs, MS Office Online

1989–2006

CONFLICT-FREE REPLICATED DATA TYPES (CRDTs)

- e.g. Riak, TomTom GPS, Teletype for Atom, ...

2006 – present

PROVING CRDTs CORRECT

RGA

OpSet

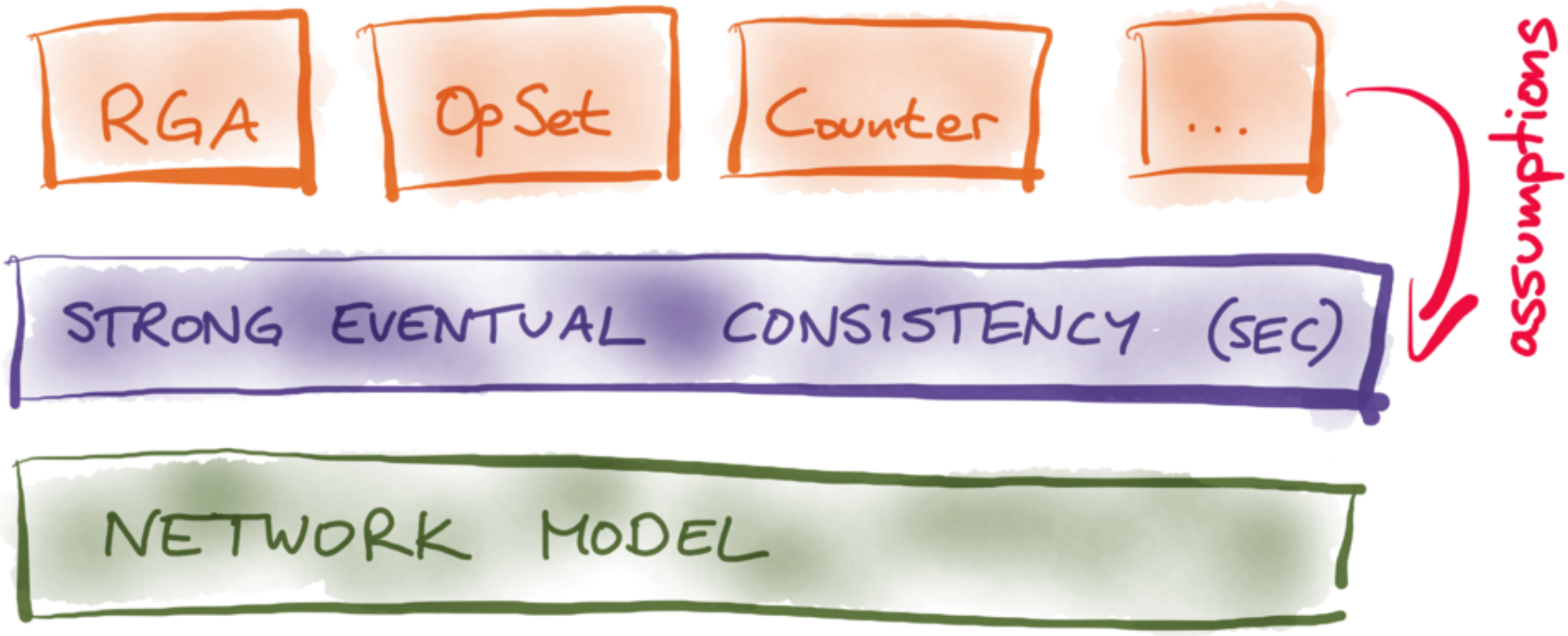
Counter

...

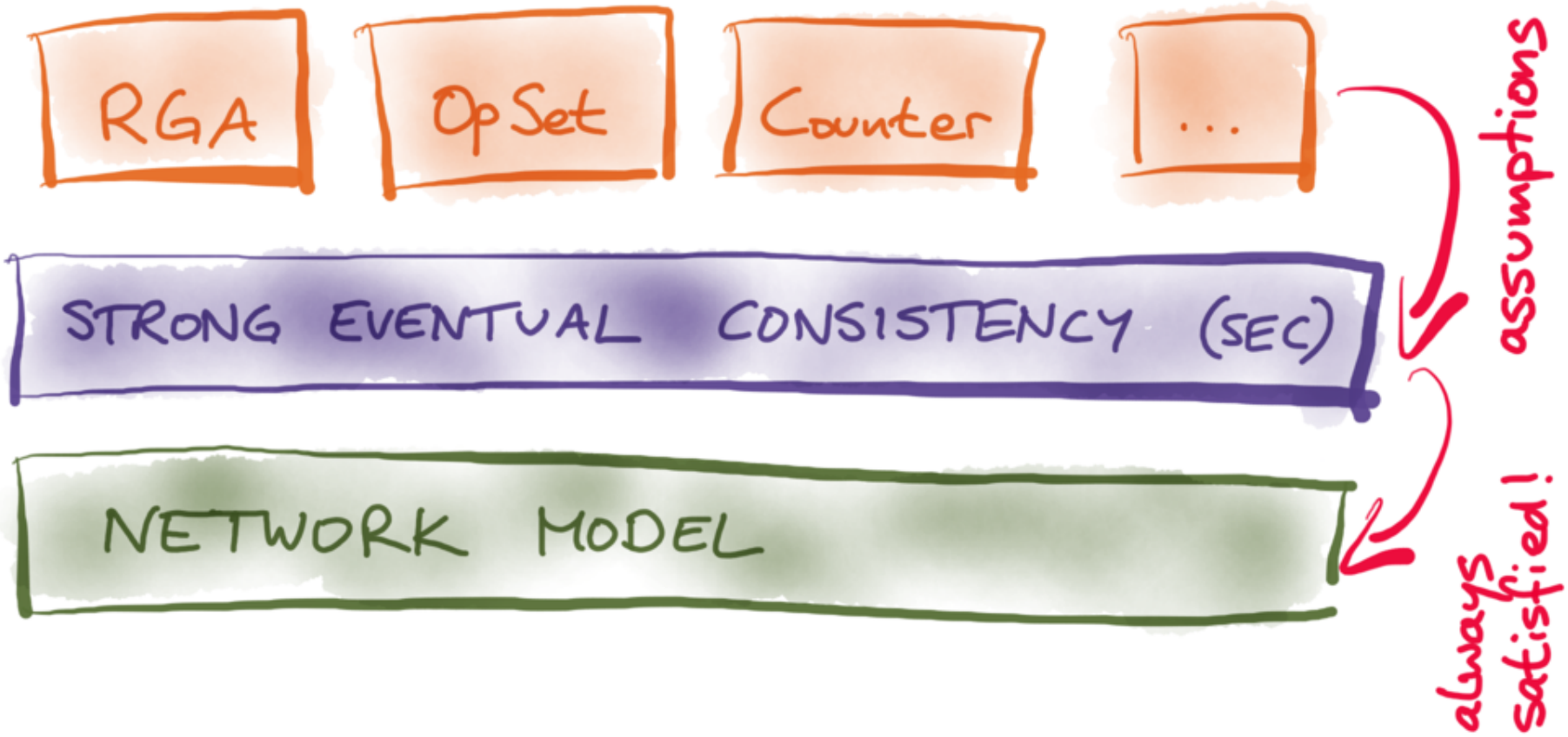
STRONG EVENTUAL CONSISTENCY (SEC)

NETWORK MODEL

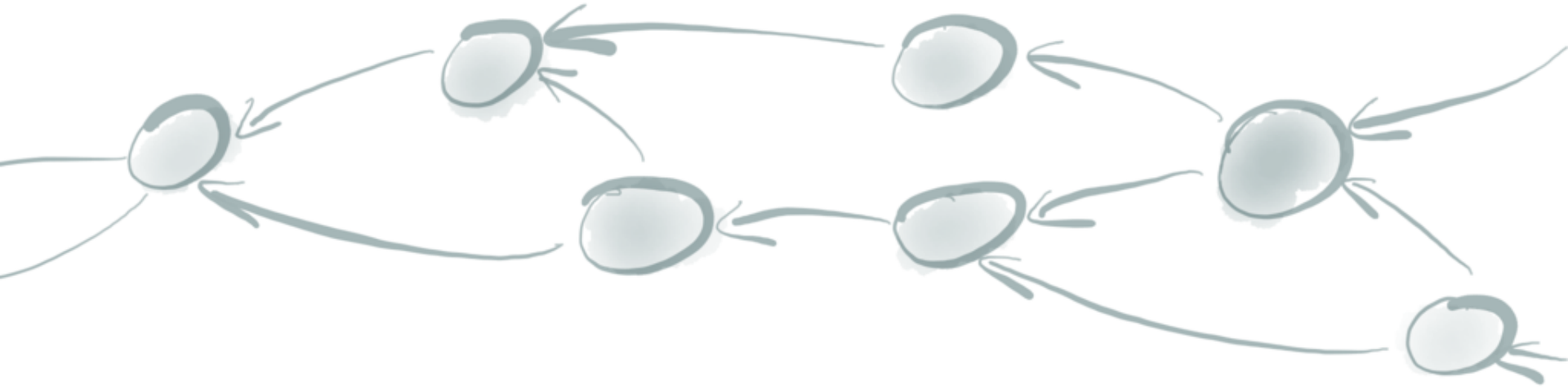
PROVING CRDTs CORRECT



PROVING CRDTs CORRECT



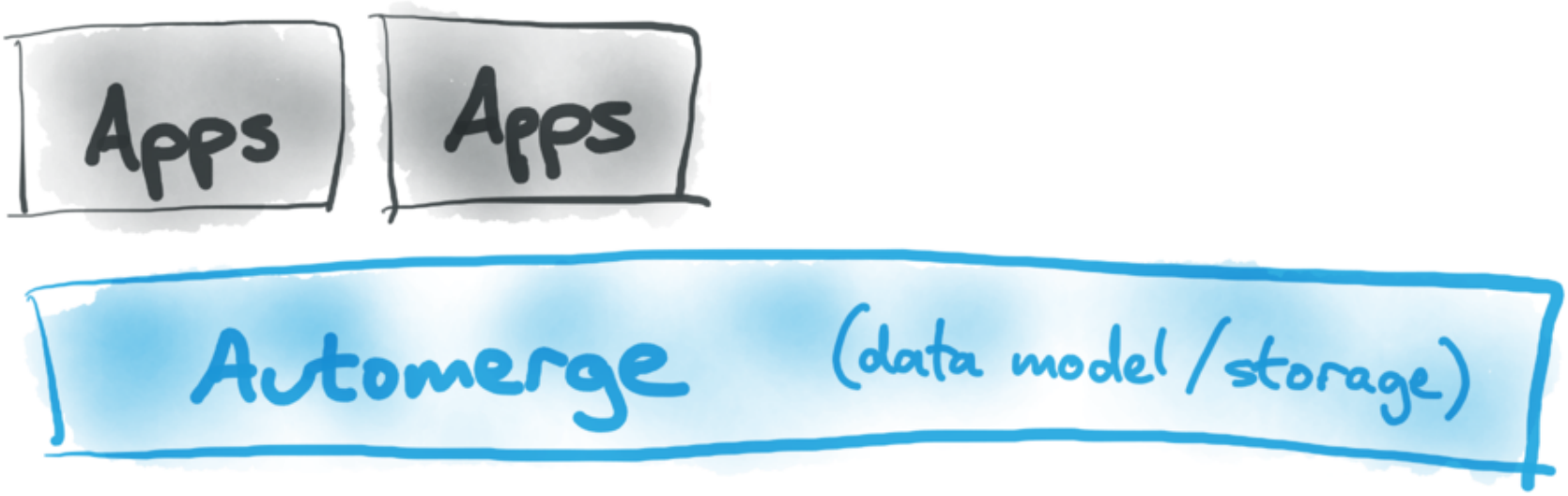
For details, see our paper at https://doi.org/10.1145/3133933
Victor B. F. Gomes, Martin Kleppmann, Dominic P. Mulligan, and Alastair R. Beresford: Verifying Strong Eventual Consistency in Distributed Systems. PACMPL 1(OOPSLA), 2017.

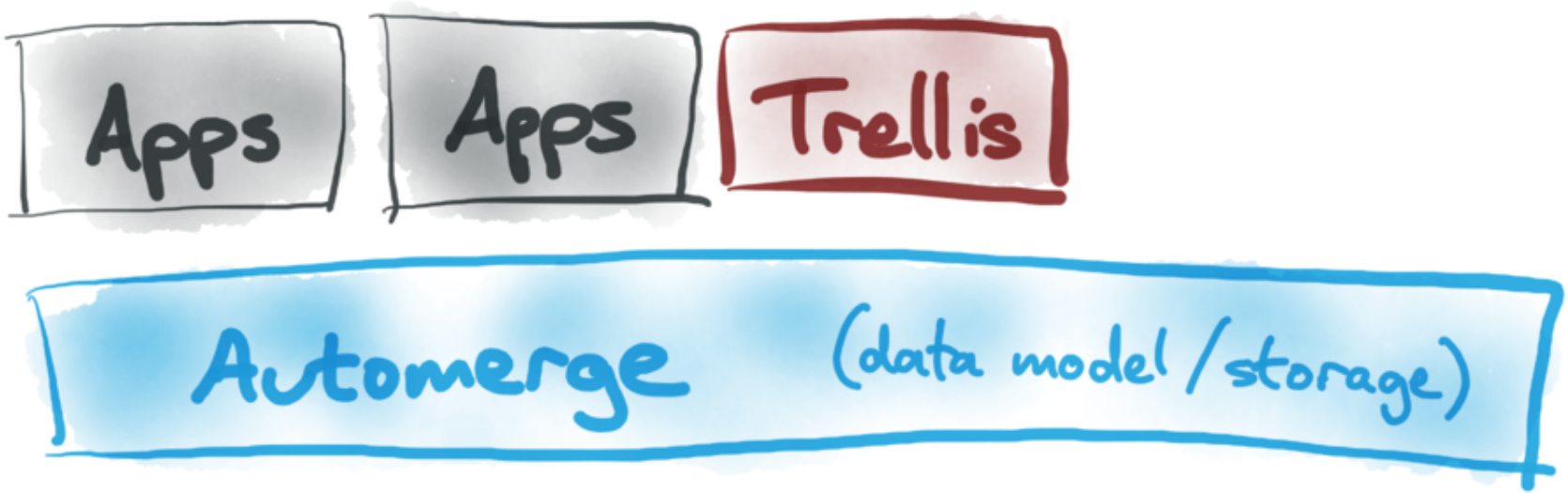


Automerge

https://github.com/automerge/automerge


Automerge
(data model / storage)


Apps
Apps
Automerge
(data model / storage)



Trellis, a Trello clone based on Automerge: https://github.com/automerge/trellis
Joint work with Orion Henry, Peter van Hardenberg, Roshan Choxi, and Adam Wiggins.

The Avengers Initiative

PROSPECTS

- Spiderman
  - 1
- Doctor Strange
- Star Lord
- Groot

Add a card...

RECRUITING

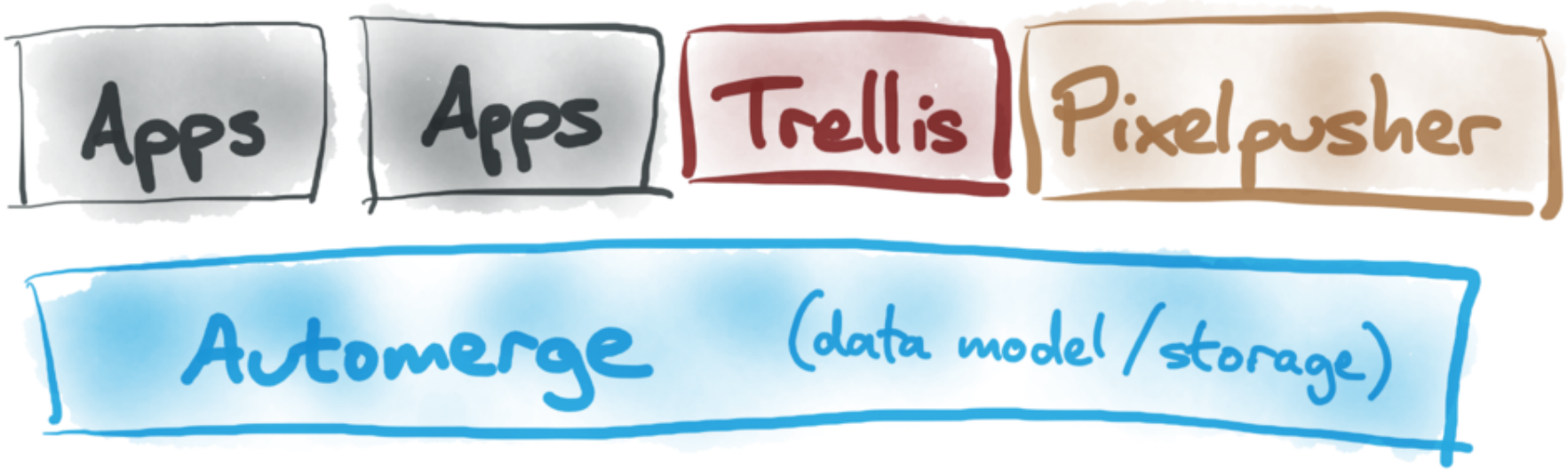
- Iron Man 

Add a card...

JOINED THE TEAM

- Black Widow 
- The Hulk
  - 1 

Add a card...



Pixelpusher, a collaborative pixel art editor: https://github.com/automerge/pixelpusher
Created by Javier Valencia, Jeff Peterson, Peter van Hardenberg, and Jim Pick.

PIXEL ART TO CSS

by JVALEN

Star 324

+ 25 50 75 100

NEW

LOAD SAVE

PREVIEW

RESET

16 + −

16 + −

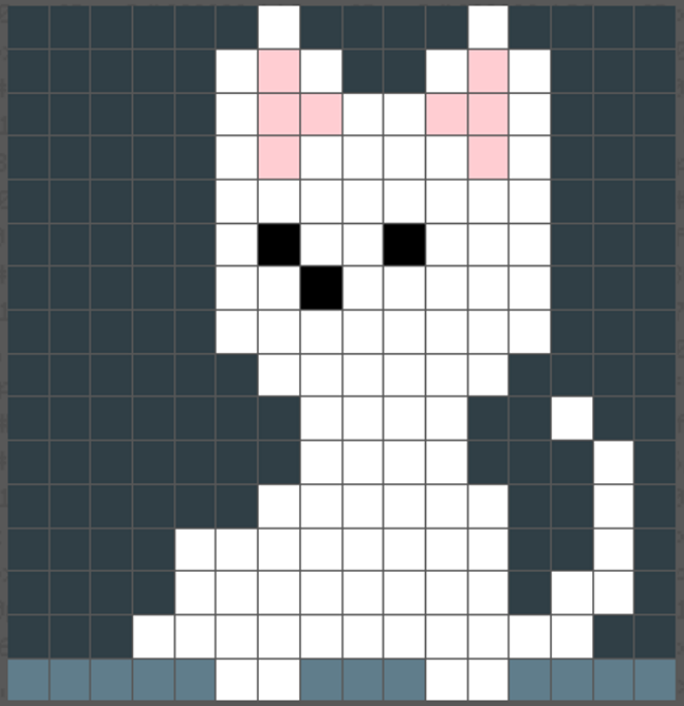
Pixel Size

10

Duration

1

CSS



PIXELPUSHER

pxlpshr://D2Sr5PkQAn7ygXyWcZFC1L9rNnaQNHeGBZgQbZqVw

Projects

Preview

Reset

Pixel Size

10

Duration

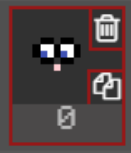
1

16 + −

16 + −

Import Image

Eyeballs

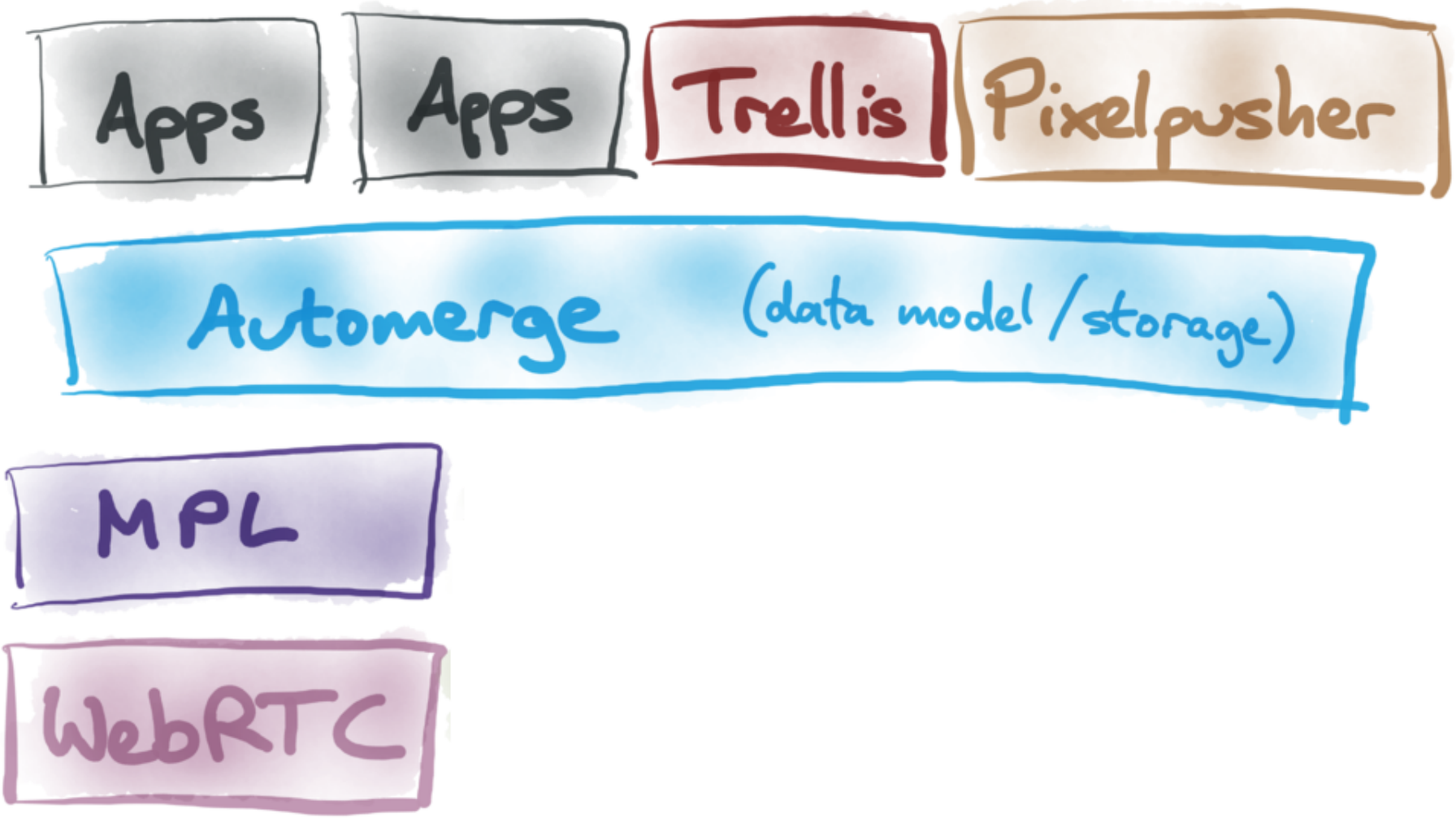


0

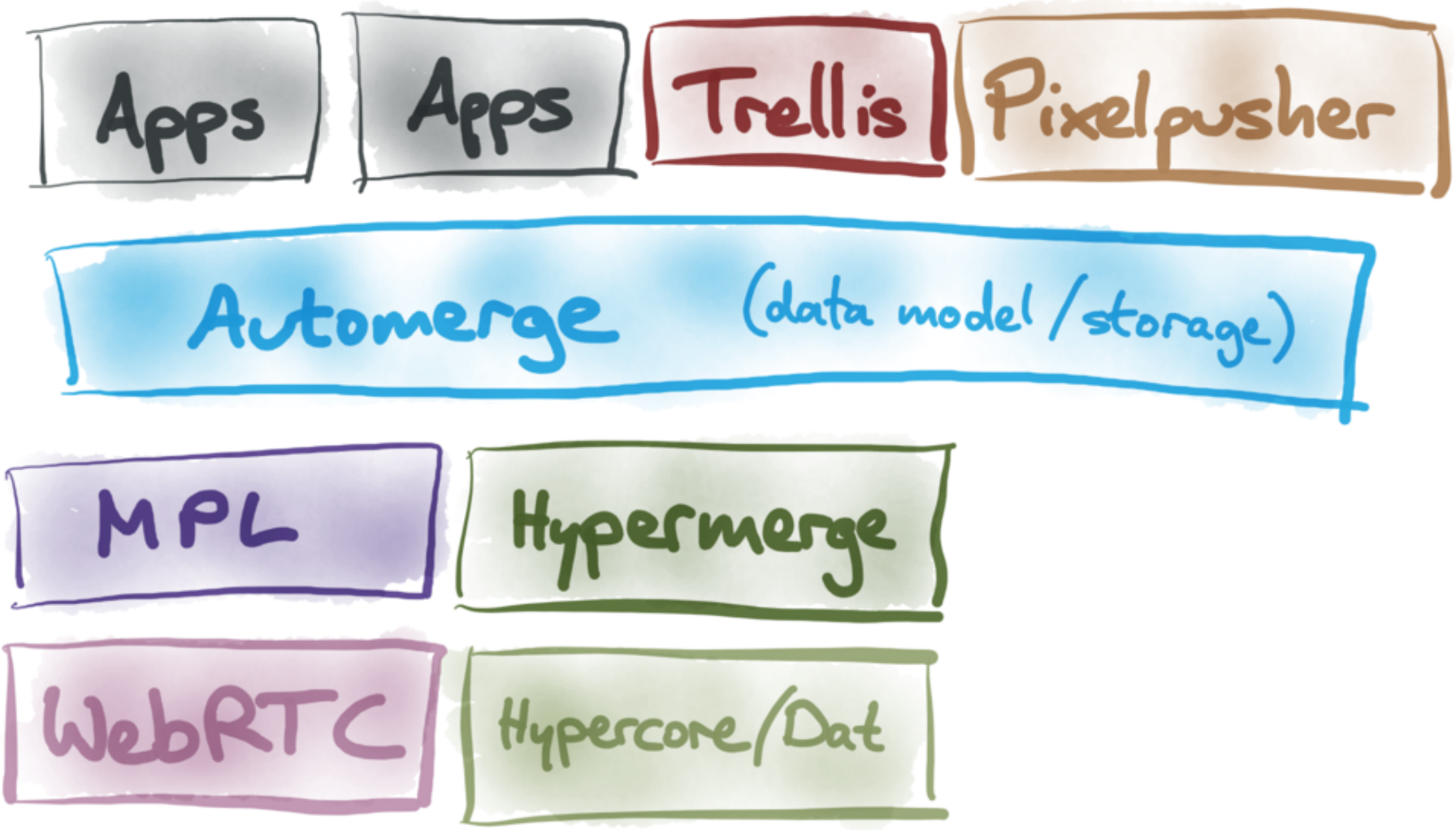
+

Versions

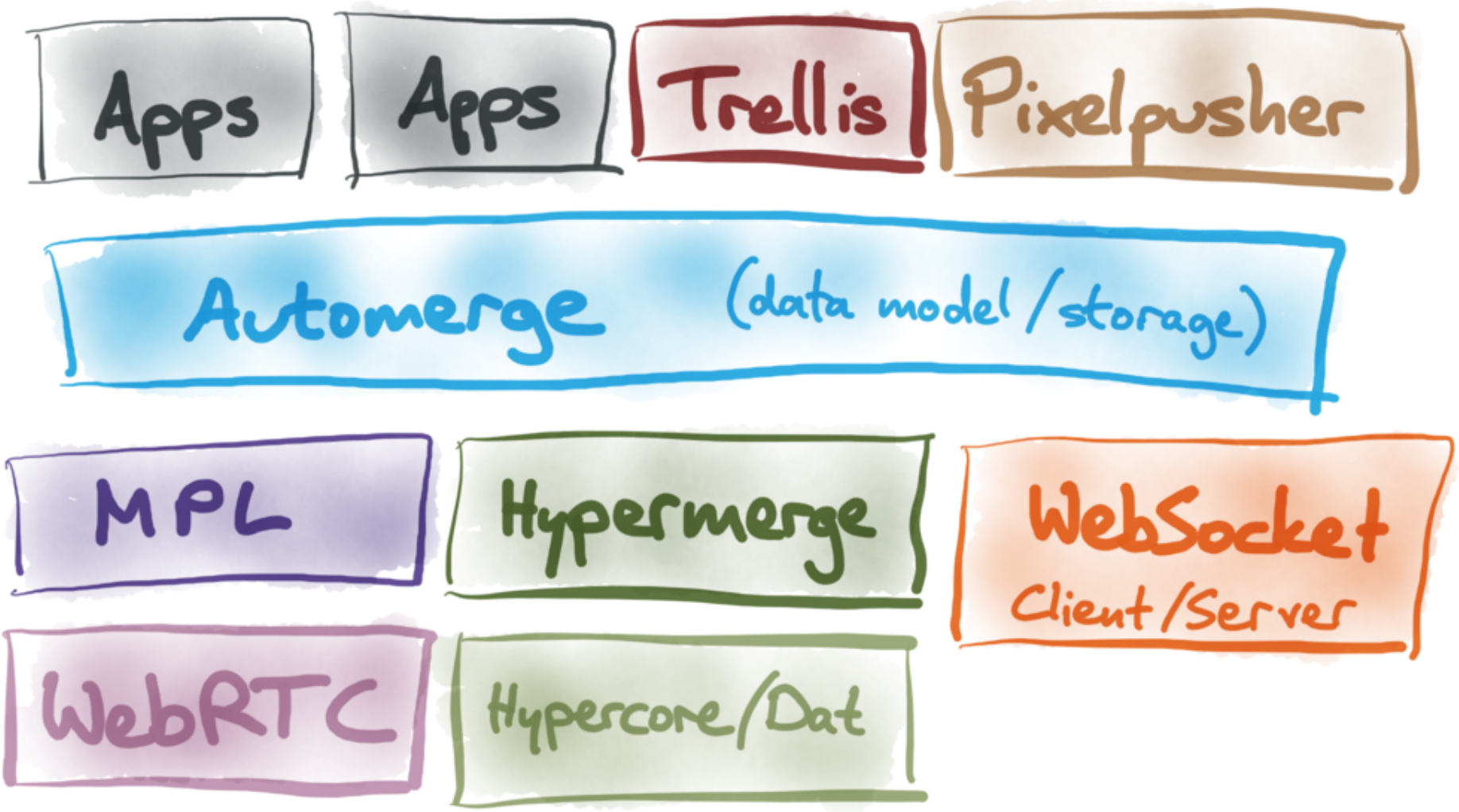
Archivers +



MPL, a WebRTC network layer for Automerge: https://github.com/automerge/mpl
Joint work with Orion Henry, Peter van Hardenberg, Roshan Choxi, and Adam Wiggins.



Hypermerge, a peer-to-peer network layer: https://github.com/automerge/hypermerge
Created by Jim Pick, Jeff Peterson, and Peter van Hardenberg.


Apps
Apps
Trellis
Pixelpusher
Automerge
(data model / storage)
MPL
Hypermerge
WebSocket
Client / Server
WebRTC
Hypercore / Dat

APPLICATION DATA (JSON)

```
{ "to-do": [
    { "title": "buy milk",
      "done": false },
    { "title": "water plants",
      "done": false }
  ],
  "settings": {"alert-sound": "ring", ...}
}
```

APPLICATION DATA (JSON)

```
{ "to-do": [
    { "title": "buy milk",
      "done": false },
    { "title": "water plants",
      "done": false }
  ],
  "settings": {"alert-sound": "ring", ...}
}
```

ordered list

APPLICATION DATA (JSON)

```
{ "to-do": [
    { "title": "buy milk",
      "done": false },
    { "title": "water plants",
      "done": false }
  ],
  "settings": {"alert-sound": "ring", ...}
}
```

ordered list

nested maps

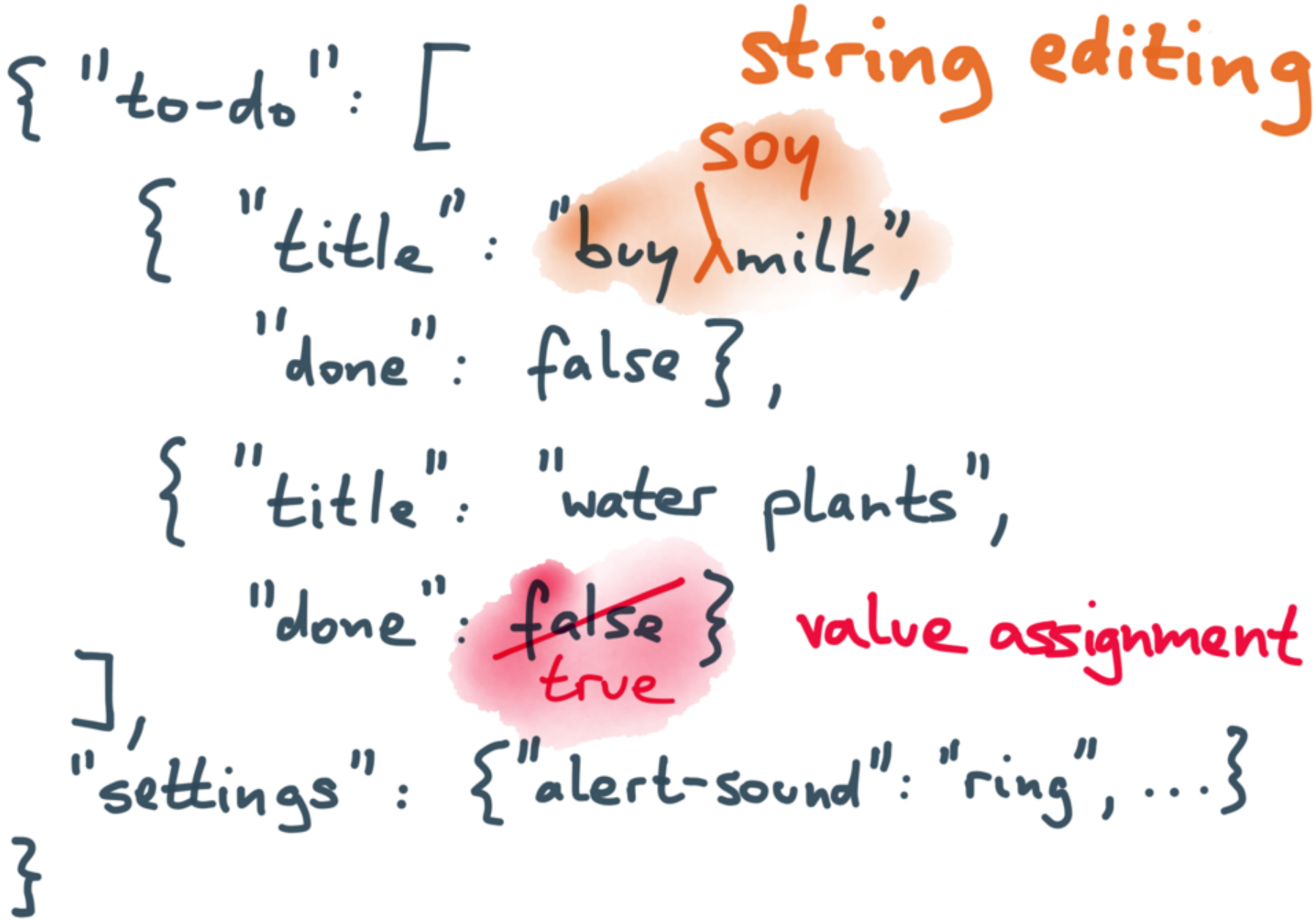
EDITING OPERATIONS

```
{ "to-do": [
    { "title": "buy milk",
      "done": false },
    { "title": "water plants",
      "done": false }
  ],
  "settings": {"alert-sound": "ring", ...}
}
```

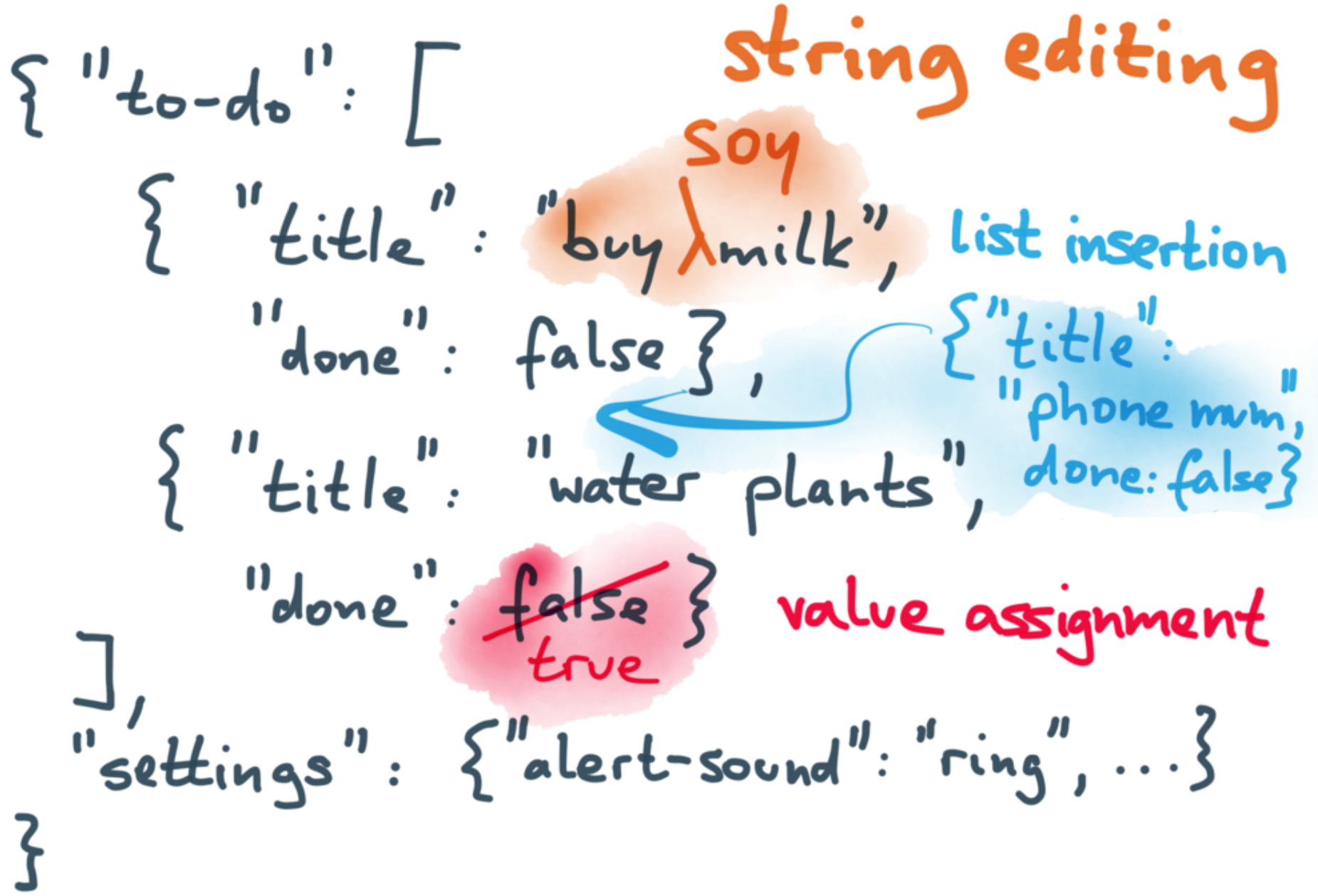
true

value assignment

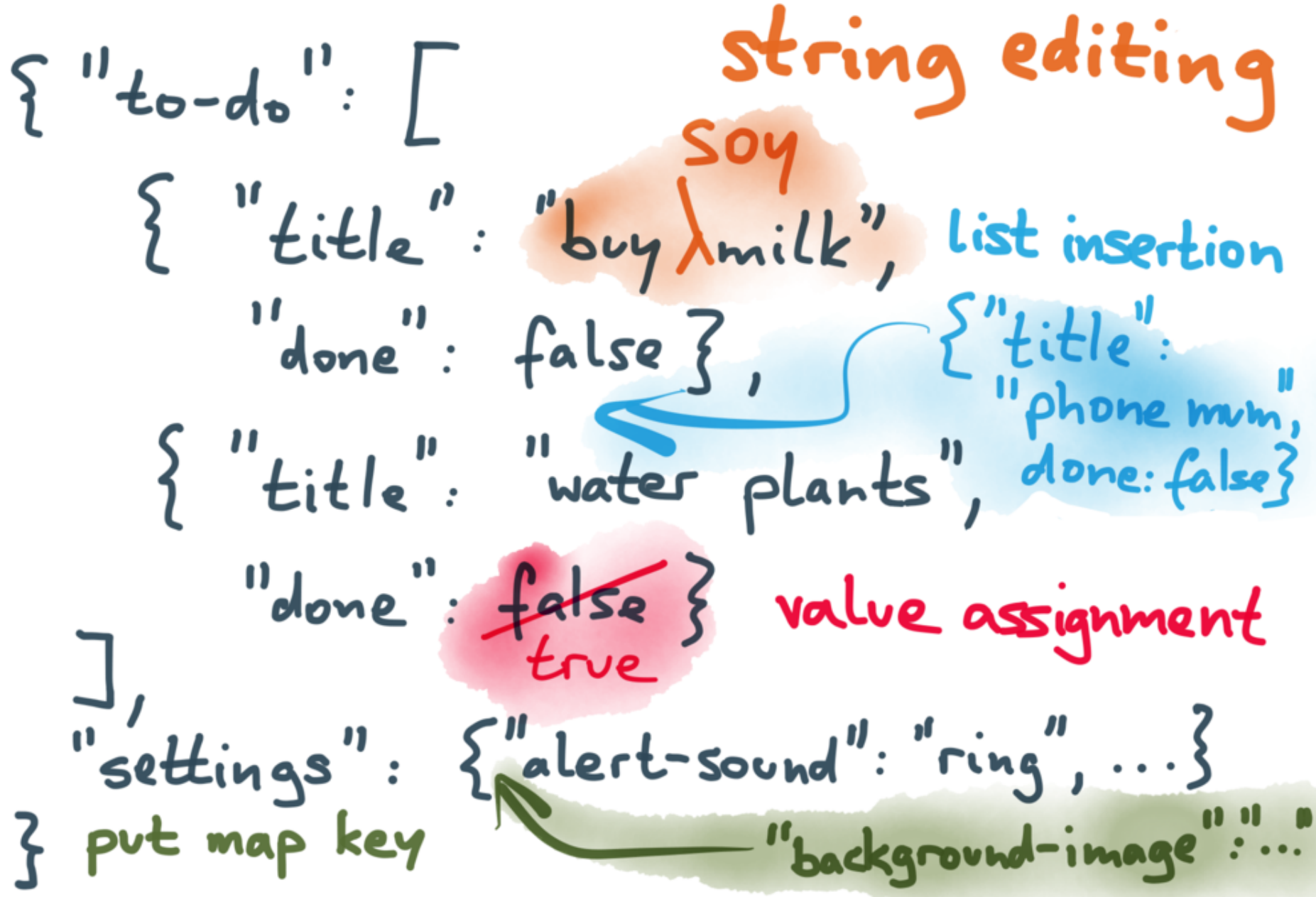
EDITING OPERATIONS



EDITING OPERATIONS



EDITING OPERATIONS



```
state = Automerge.change(state, "Add todo item",
  (doc) => {
    doc.todos.push({
      title: "Buy milk",
      done: false
    })
  })
```

state is immutable
(Automerge.change() returns
new object)

```
state = Automerge.change(state, "Add todo item",
  (doc) => {
    doc.todos.push({
      title: "Buy milk",
      done: false
    })
  })
```

state is immutable (Automerge.change() returns new object)

"Commit message" (optional)

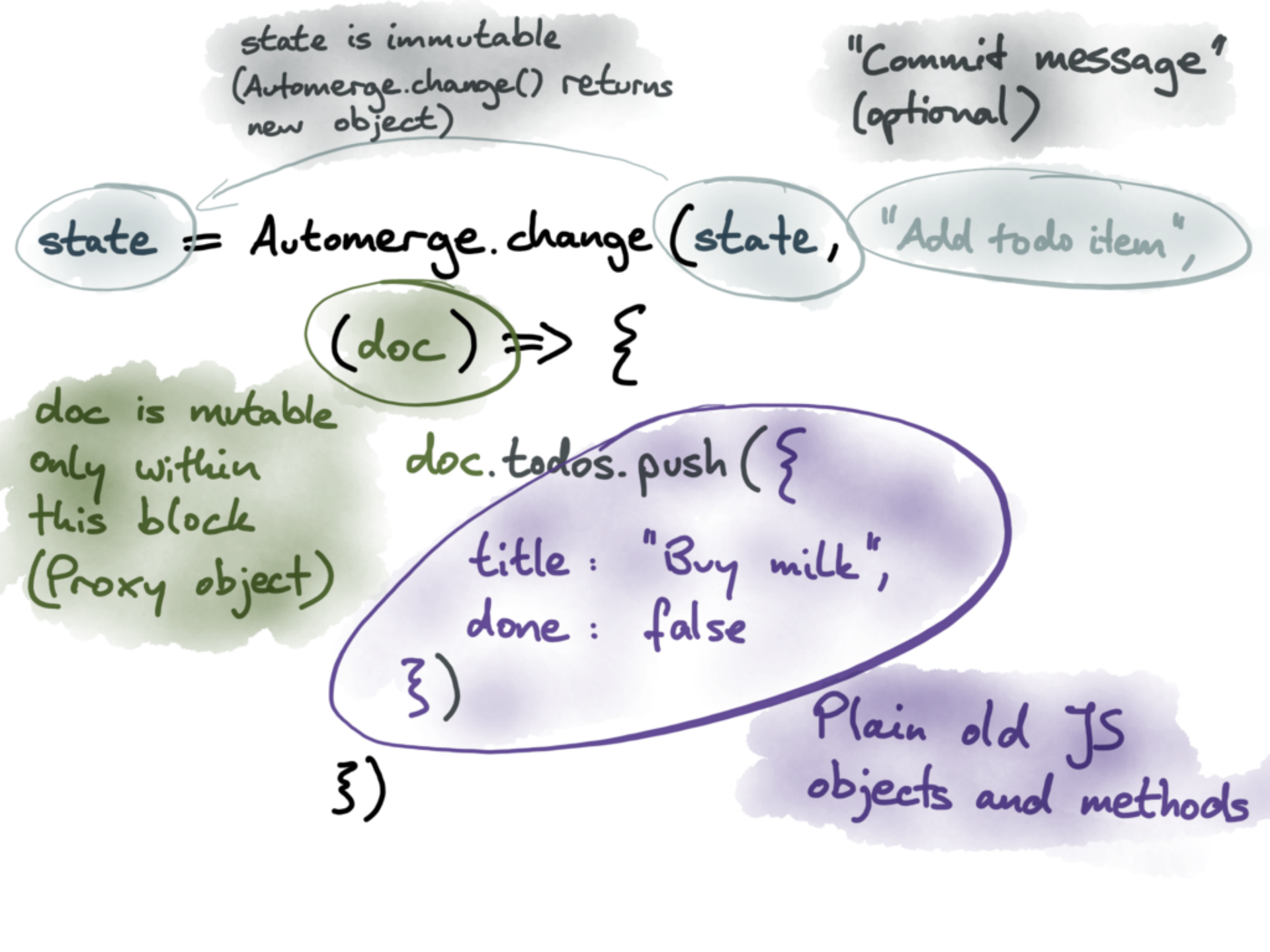
```
state = Automerge.change(state, "Add todo item",
  (doc) => {
    doc.todos.push({
      title: "Buy milk",
      done: false
    })
  })
```

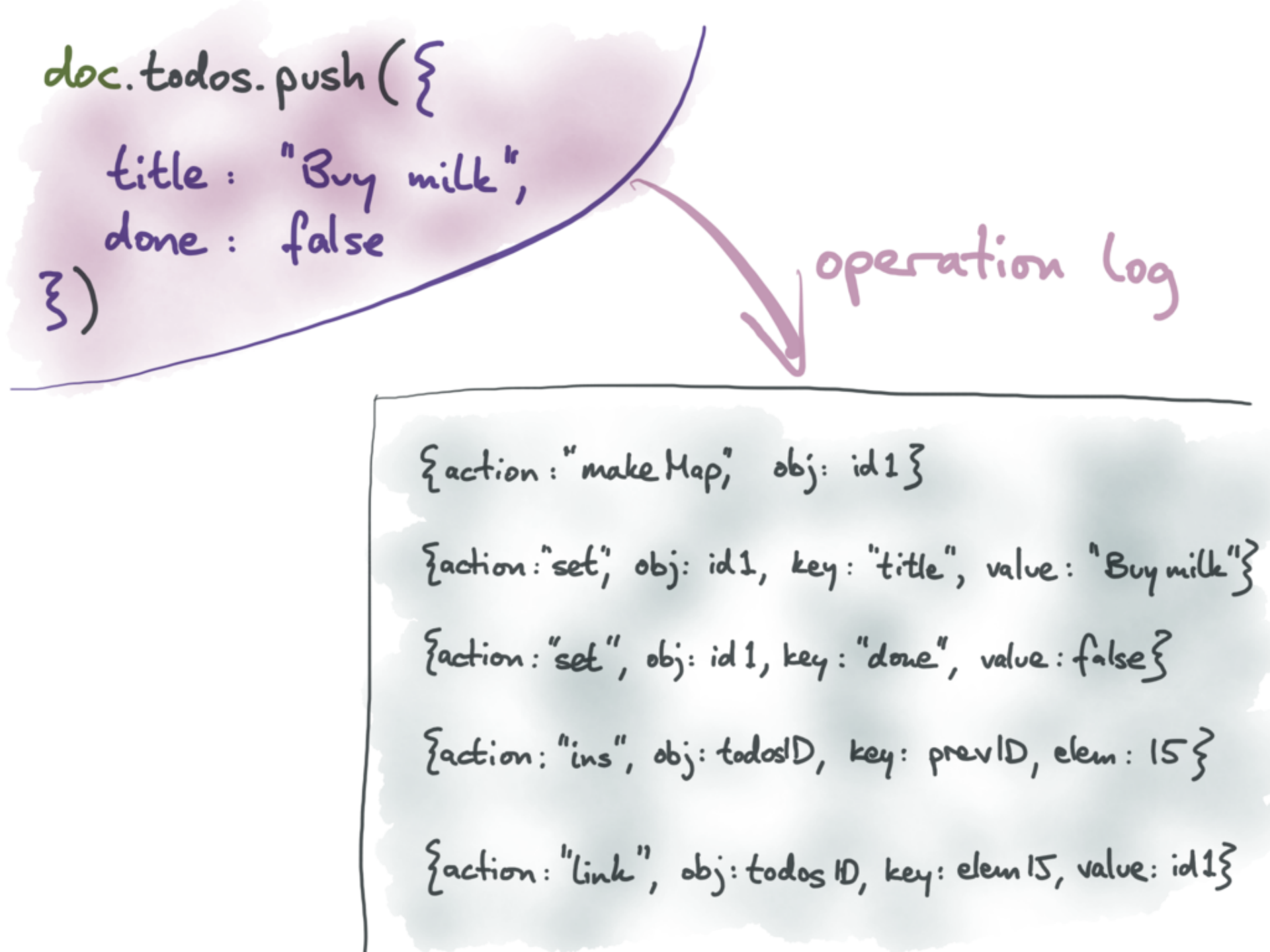
state is immutable
(Automerge.change() returns
new object)

"Commit message"
(optional)

```
state = Automerge.change(state, "Add todo item",
  (doc) => {
    doc.todos.push({
      title: "Buy milk",
      done: false
    })
  })
```

doc is mutable
only within
this block
(Proxy object)


state is immutable
(Automerge.change() returns
new object)
"Commit message"
(optional)
state = Automerge.change(state, "Add todo item",
(doc) => {
doc is mutable
only within
this block
(Proxy object)
doc.todos.push({
title: "Buy milk",
done: false
})
})
Plain old JS
objects and methods


doc.todos.push({
title: "Buy milk",
done: false
})
operation log
{action: "makeMap", obj: id1}
{action: "set", obj: id1, key: "title", value: "Buy milk"}
{action: "set", obj: id1, key: "done", value: false}
{action: "ins", obj: todosID, key: prevID, elem: 15}
{action: "link", obj: todos ID, key: elem 15, value: id1}

TIME TRAVEL

```
Automerge.getHistory(state)
```

TIME TRAVEL

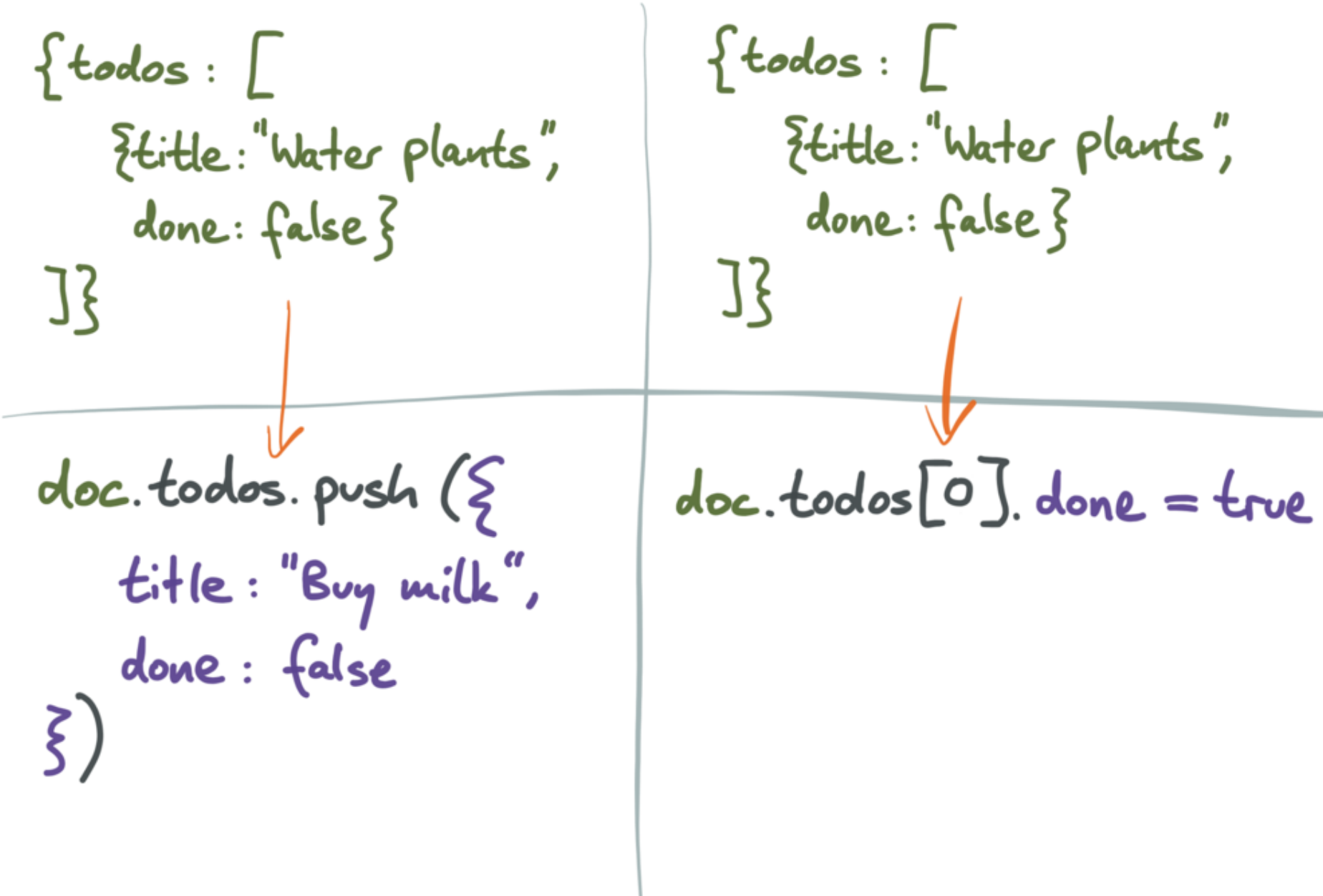
```
Automerge.getHistory(state)

⟹ [ {change: {message: "Add todo item", ...},
      snapshot: {todos: [{title: "Buy milk", ...}, ...]}},

    {change: {message: "Mark item as done", ...},
     snapshot: {todos: [{title: "Buy milk", ...}, ...]}},
    ...
  ]
```

CONCURRENT CHANGES

```
{todos : [
    {title:"Water plants",
     done: false}
]}
```

```
{todos : [
    {title:"Water plants",
     done: false}
]}
```

CONCURRENT CHANGES



CONCURRENT CHANGES

```
doc.todos.push({
    title: "Buy milk",
    done: false
})
```

```
doc.todos[0].done = true
```

CONCURRENT CHANGES

```
doc.todos.push({
    title: "Buy milk",
    done: false
})
```

```
doc.todos[0].done = true
```

```
{todos: [
    {title: "Water plants",
     done: false},
    {title: "Buy milk",
     done: false}
]}
```

```
{todos: [
    {title: "Water plants",
     done: true}
]}
```

CONCURRENT CHANGES

```
doc.todos.push({
    title: "Buy milk",
    done: false
})
```

```
doc.todos[0].done = true
```


```
{todos: [
    {title: "Water plants",
     done: true},
    {title: "Buy milk",
     done: false}
]}
```

```
{todos: [
    {title: "Water plants",
     done: true},
    {title: "Buy milk",
     done: false}
]}
```

Convergence guarantee:

Any two nodes have seen the same set of operations (but maybe in a different order!)

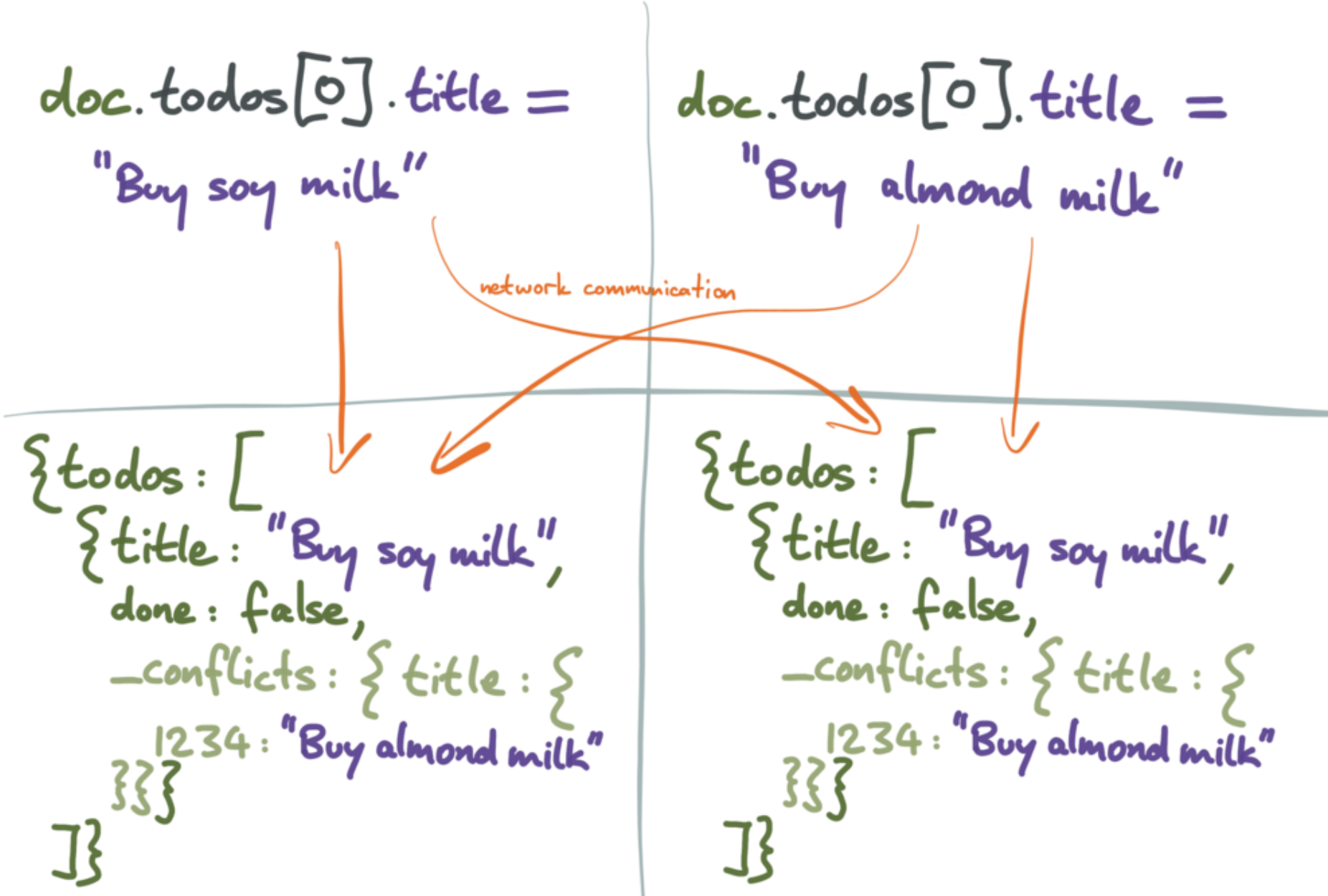
They are in the same state

CONCURRENT CHANGES

```
doc.todos[0].title =
  "Buy soy milk"
```

```
doc.todos[0].title =
  "Buy almond milk"
```

CONCURRENT CHANGES

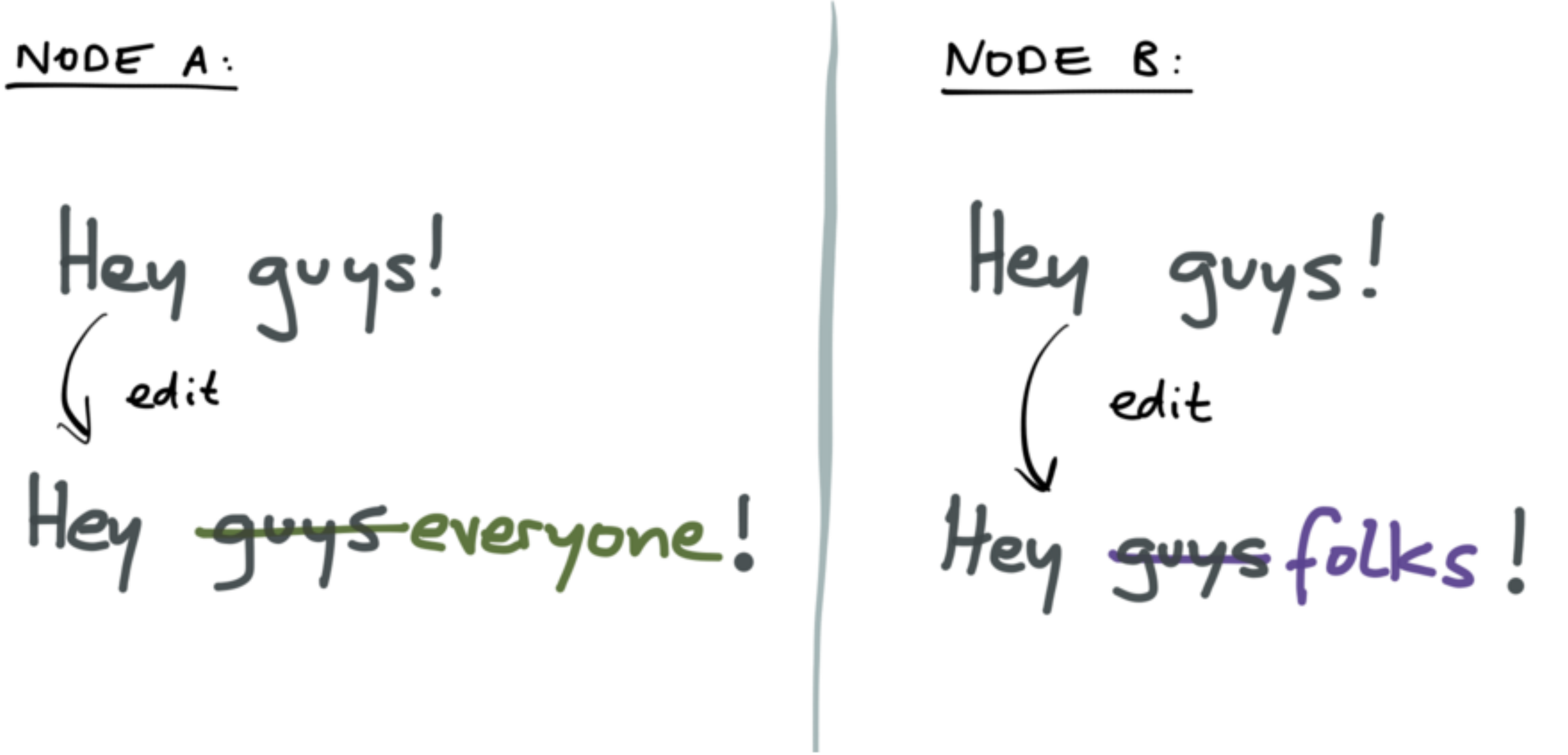


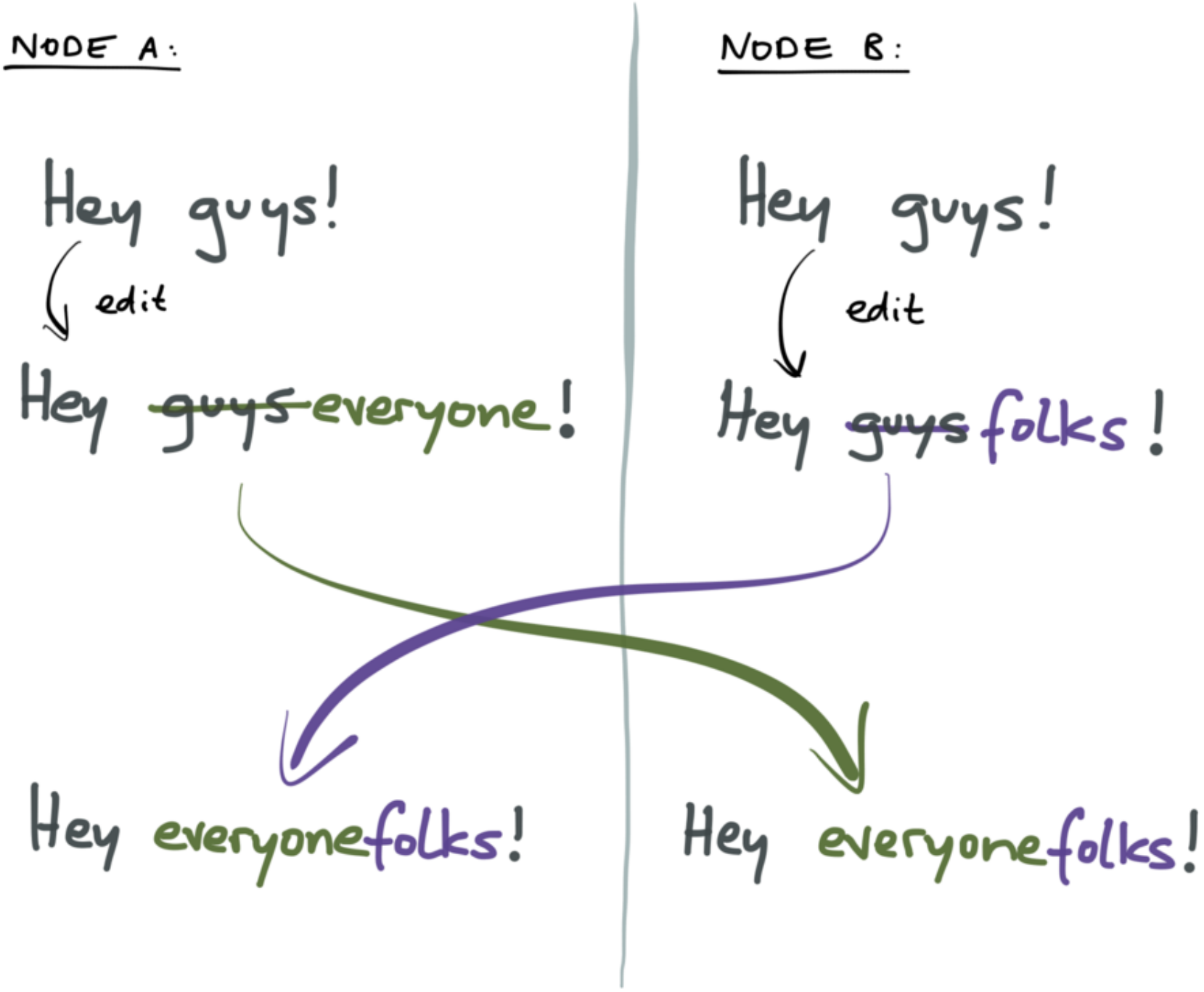
NODE A:

Hey guys!

NODE B:

Hey guys!


NODE A:
Hey guys!
edit
Hey ~~guys~~ everyone!
NODE B:
Hey guys!
edit
Hey ~~guys~~ folks!


NODE A:
Hey guys!
edit
Hey ~~guys~~ everyone!
Hey everyonefolks!
NODE B:
Hey guys!
edit
Hey ~~guys~~ folks!
Hey everyonefolks!

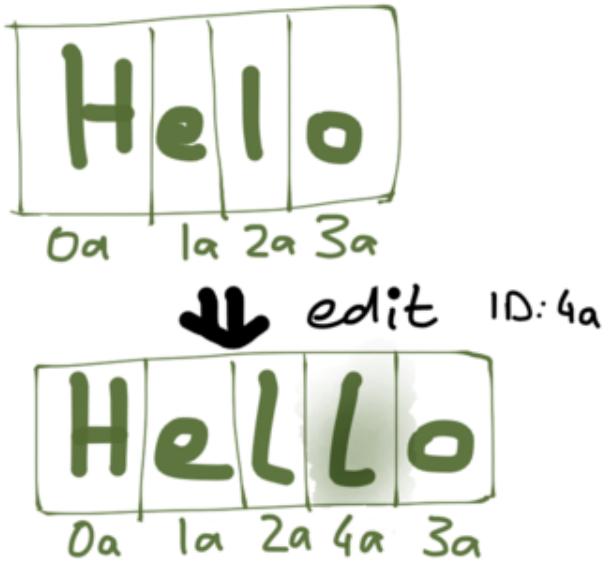
ORDERED LIST CRDT (NUTSHELL)

NODE A:

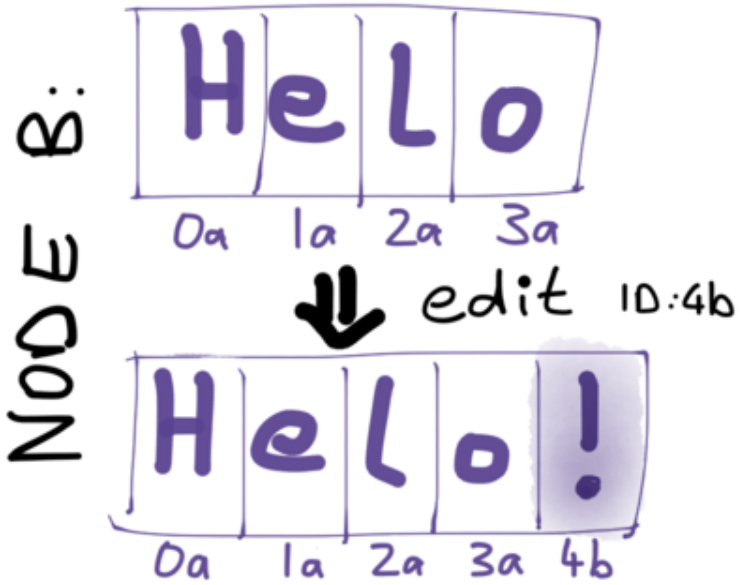
NODE B:

ORDERED LIST CRDT (NUTSHELL)

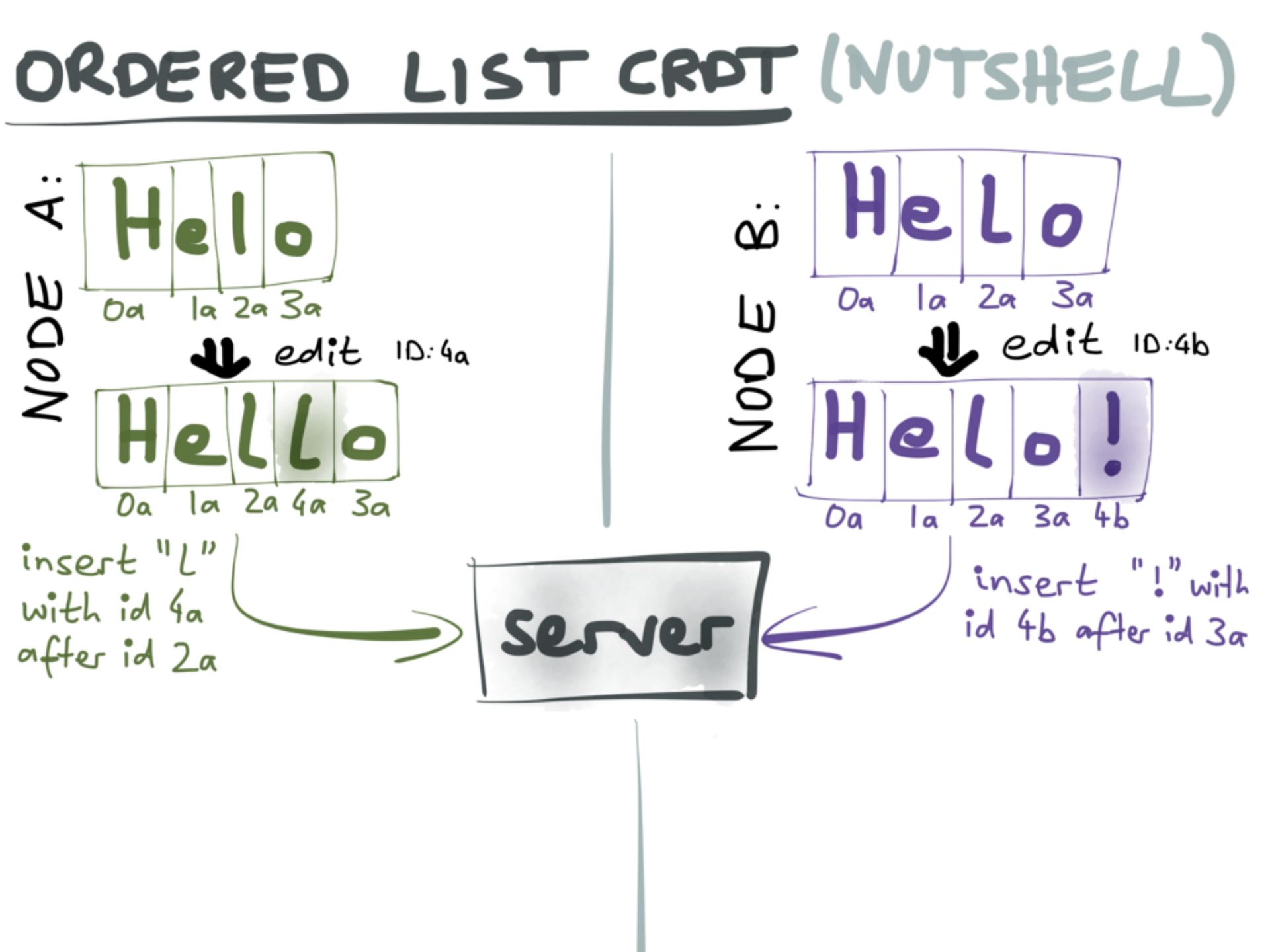
NODE A:



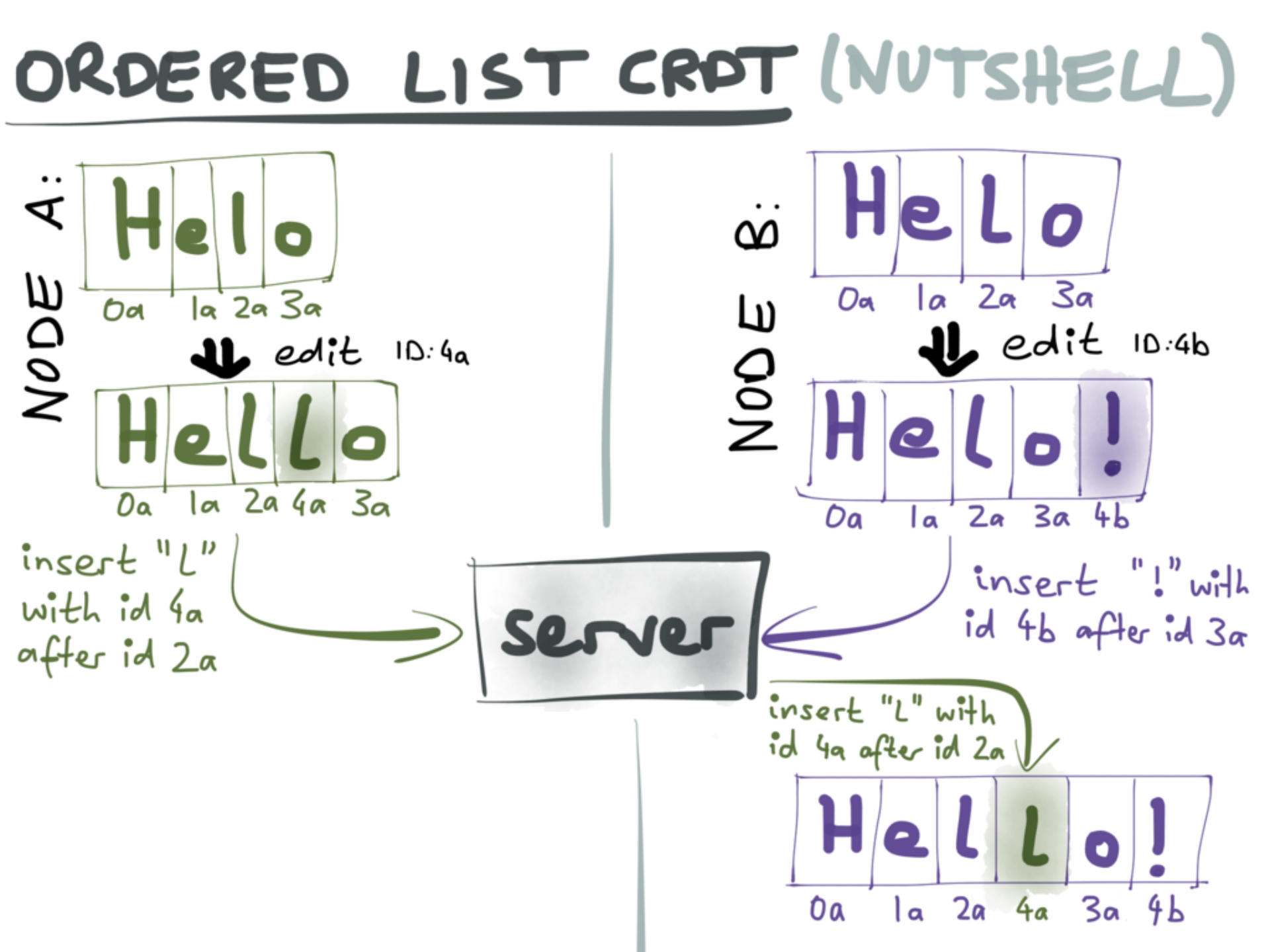
NODE B:



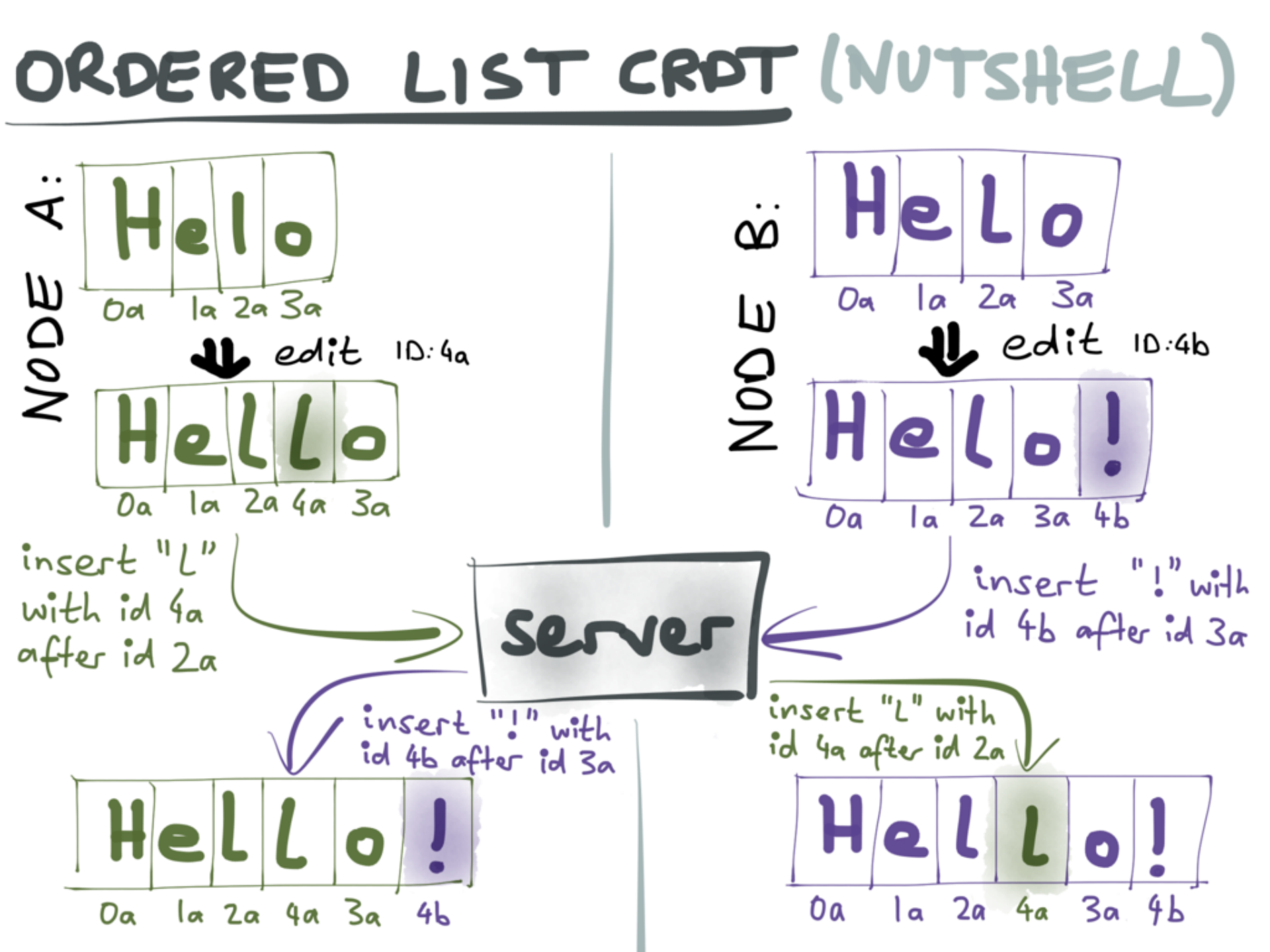
ORDERED LIST CRDT (NUTSHELL)



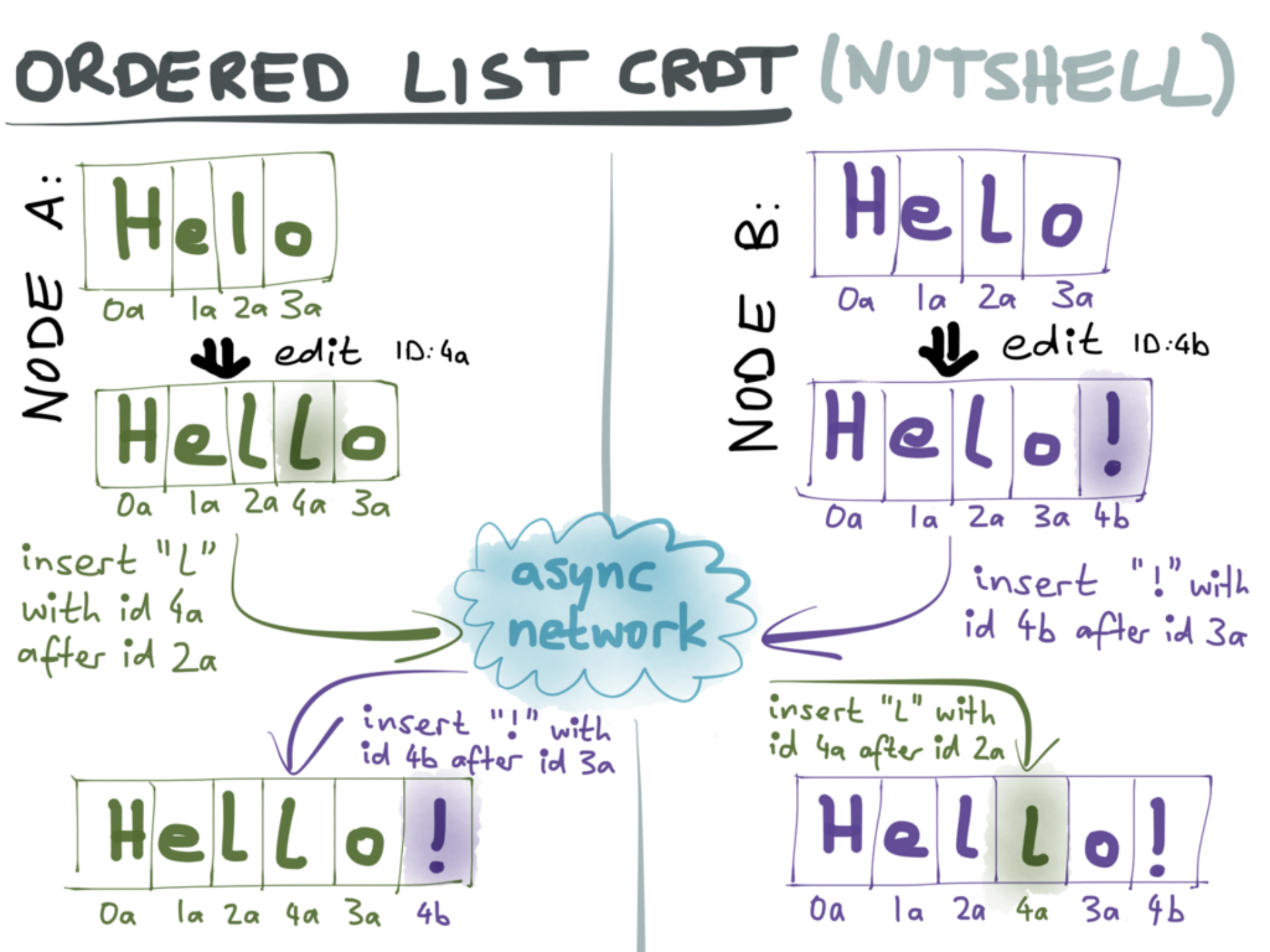
ORDERED LIST CRDT (NUTSHELL)



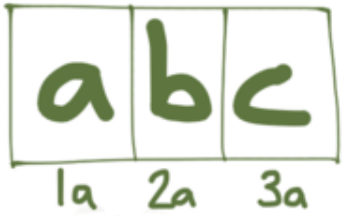
ORDERED LIST CRDT (NUTSHELL)

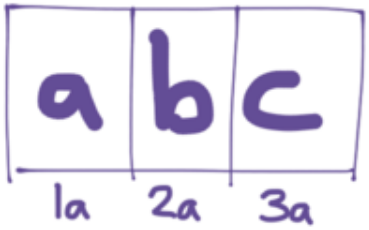


ORDERED LIST CRDT (NUTSHELL)

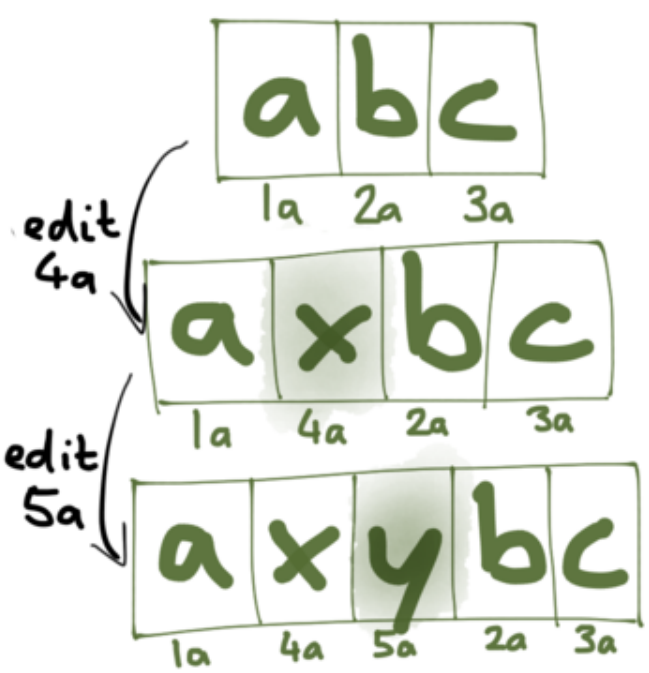


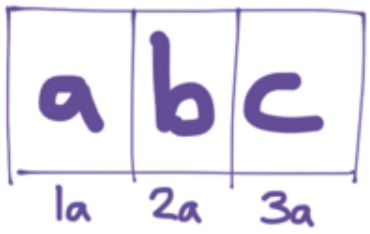
INSERTING IN THE SAME PLACE



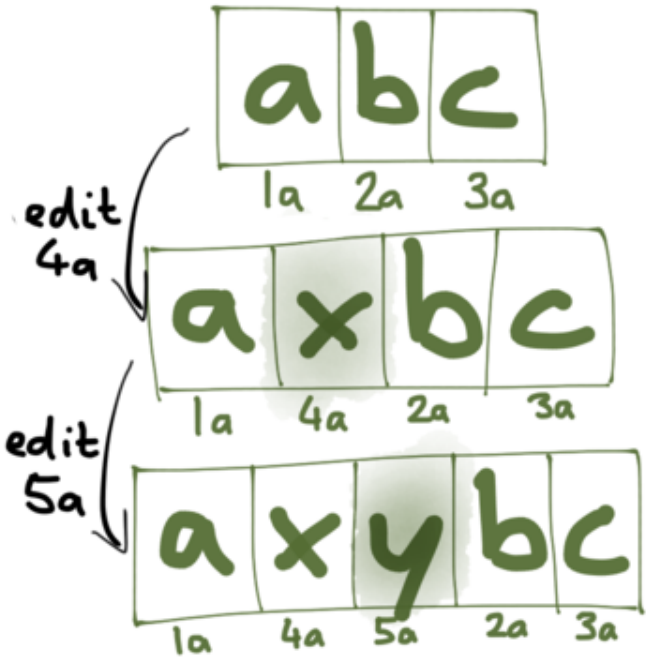


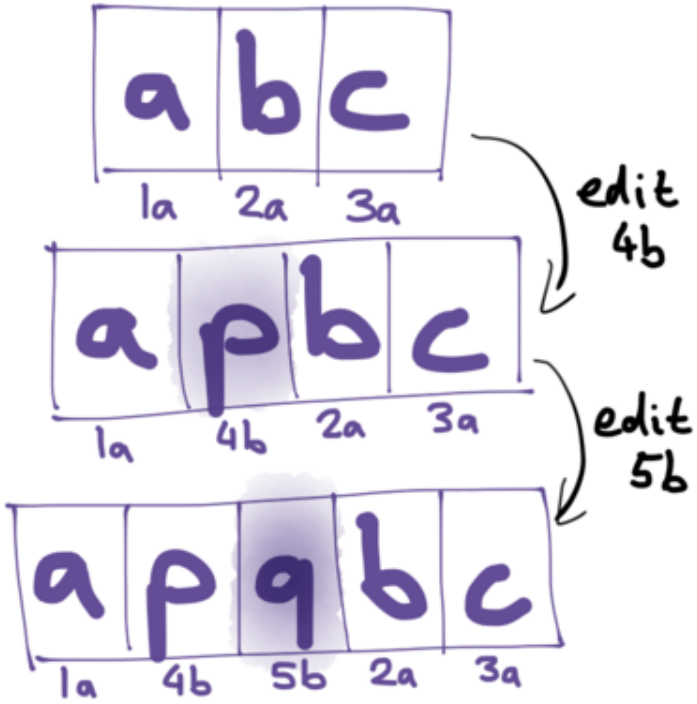
INSERTING IN THE SAME PLACE



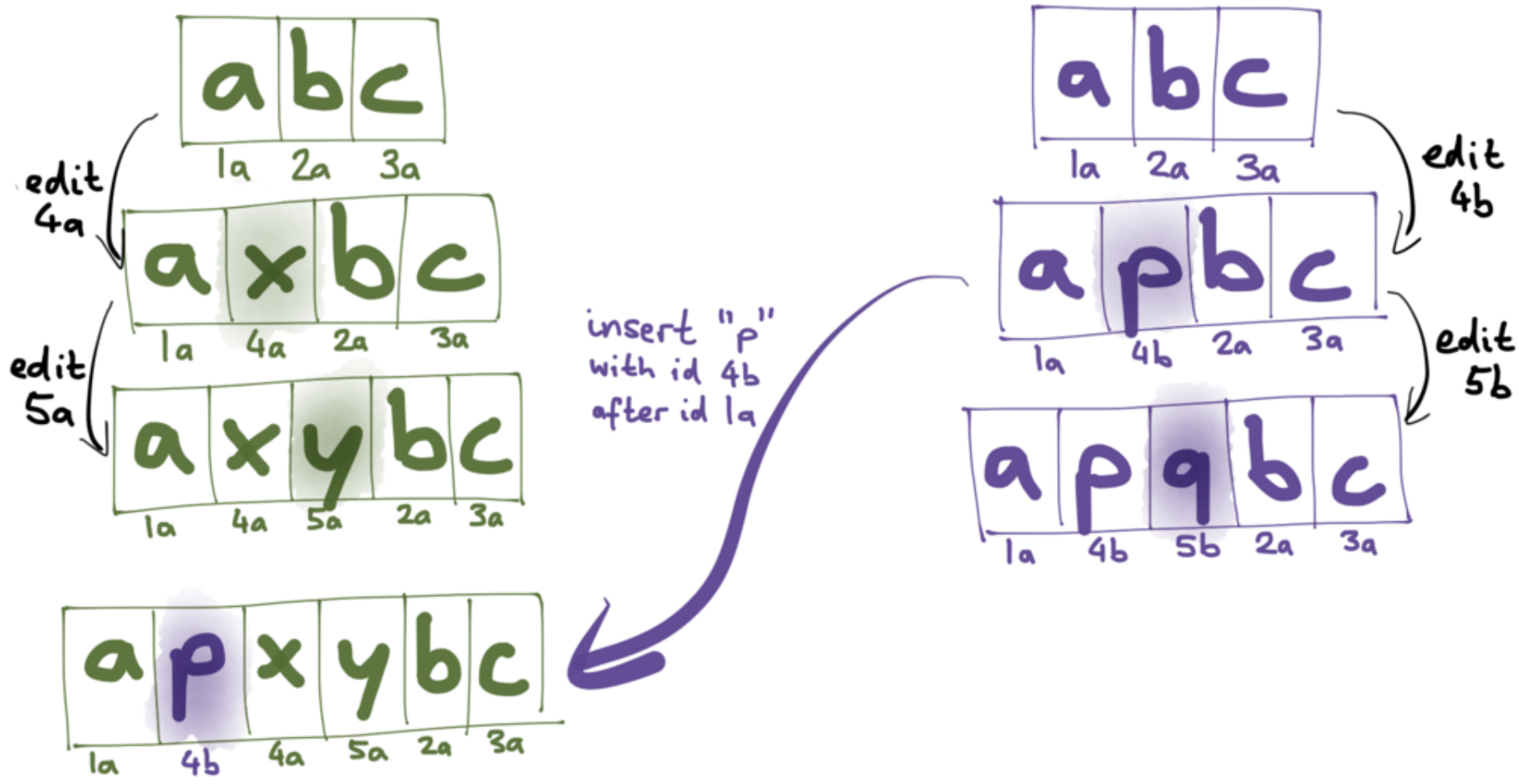


INSERTING IN THE SAME PLACE

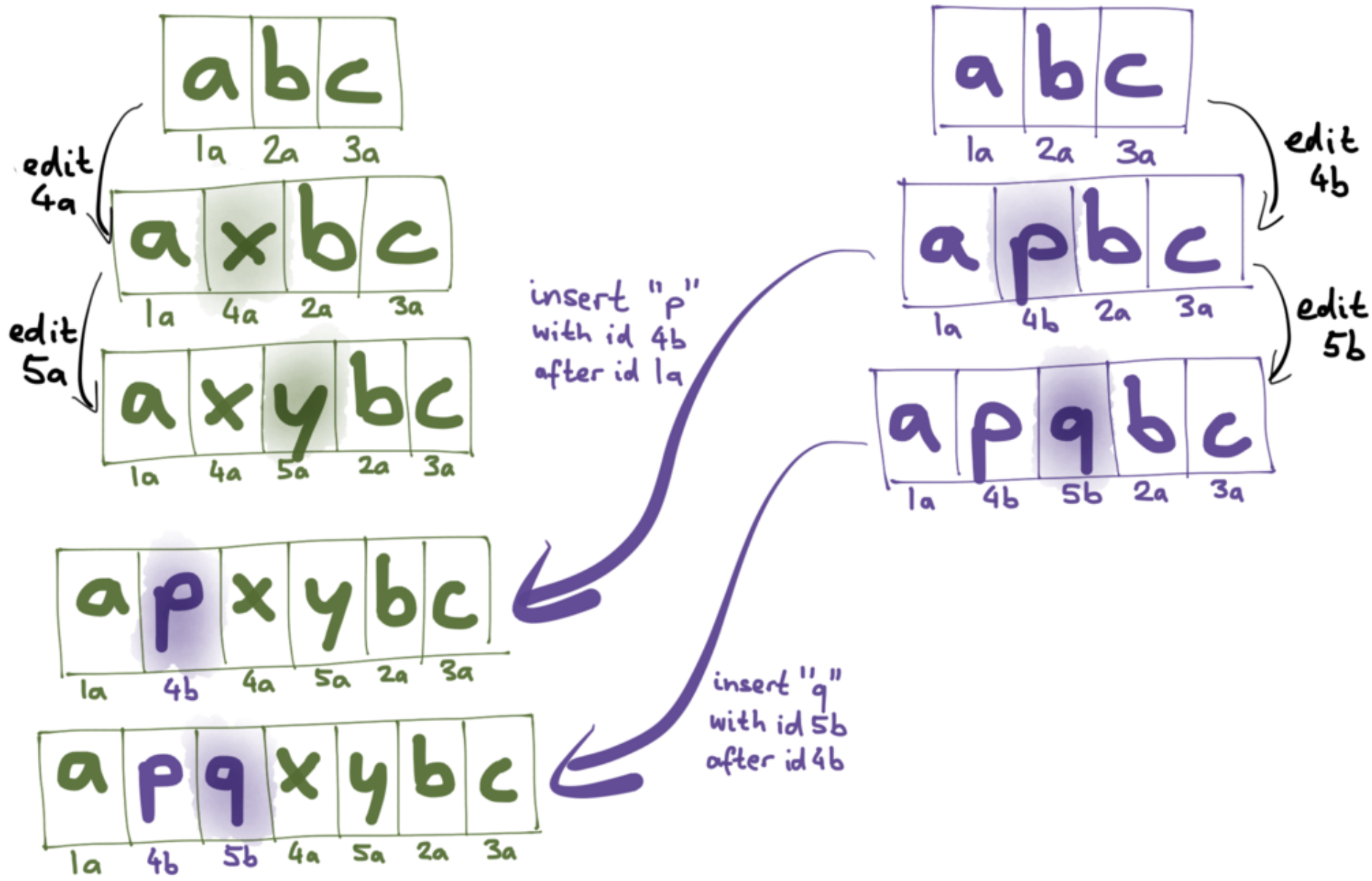




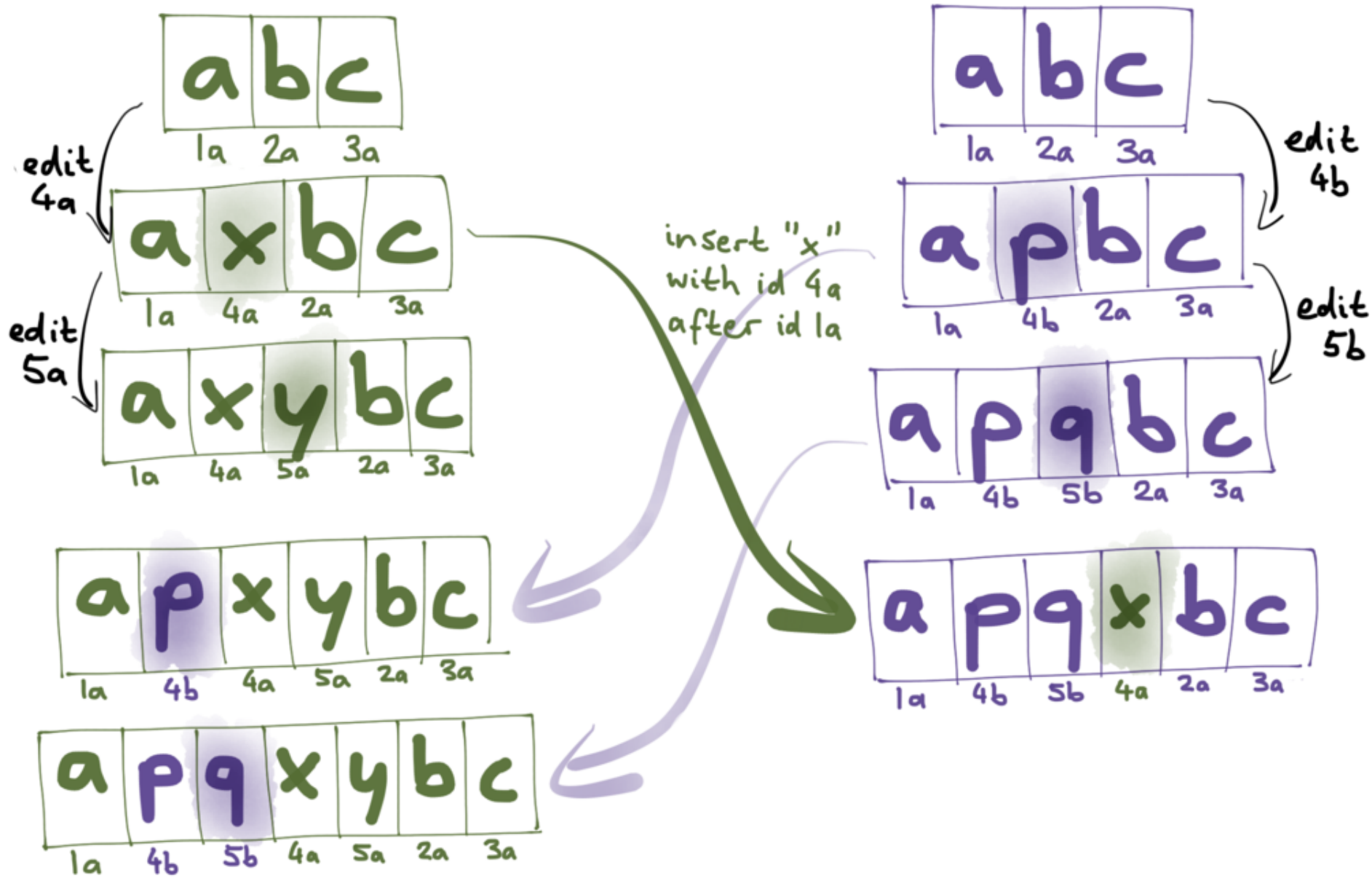
INSERTING IN THE SAME PLACE



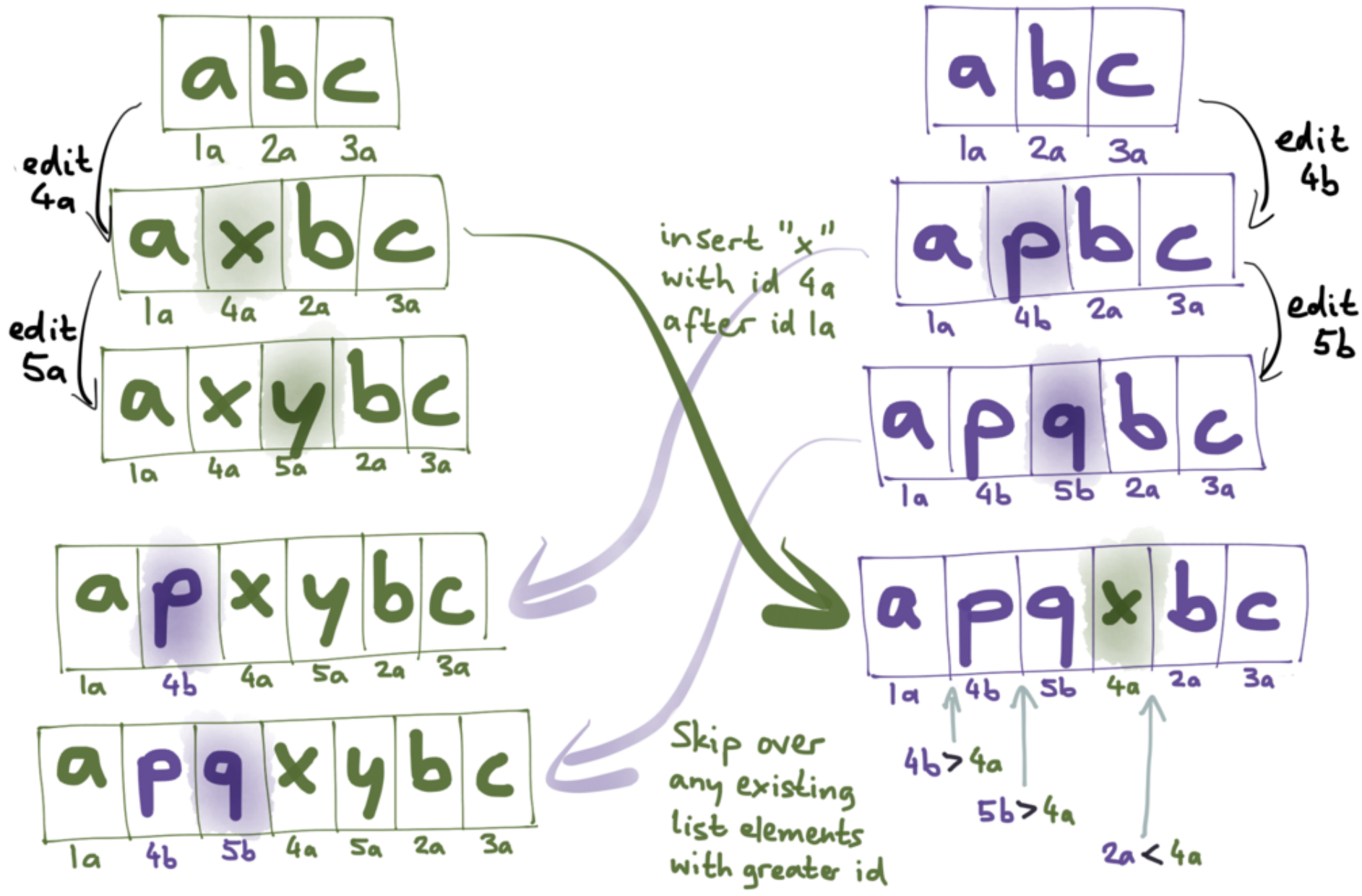
INSERTING IN THE SAME PLACE



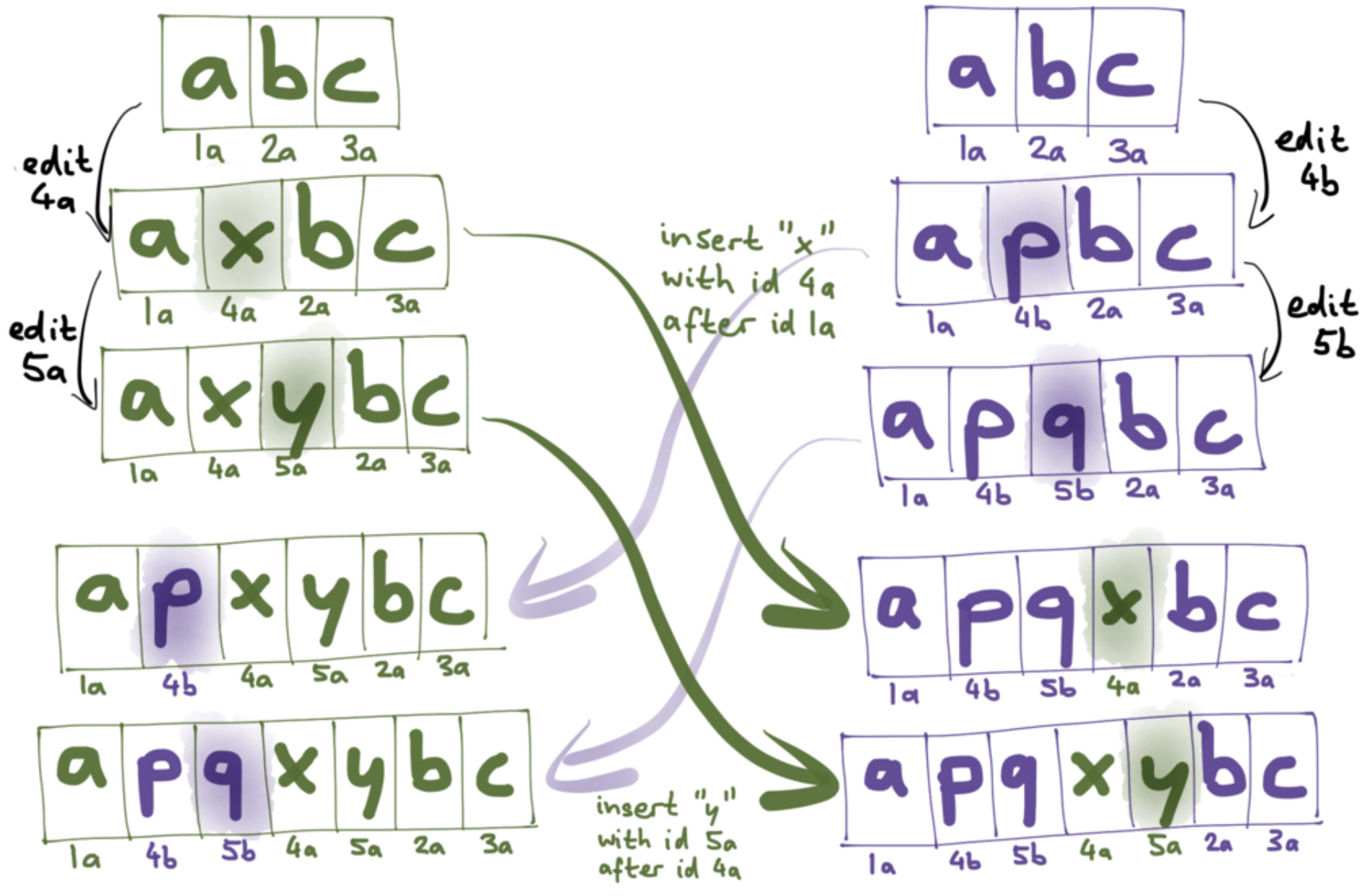
INSERTING IN THE SAME PLACE



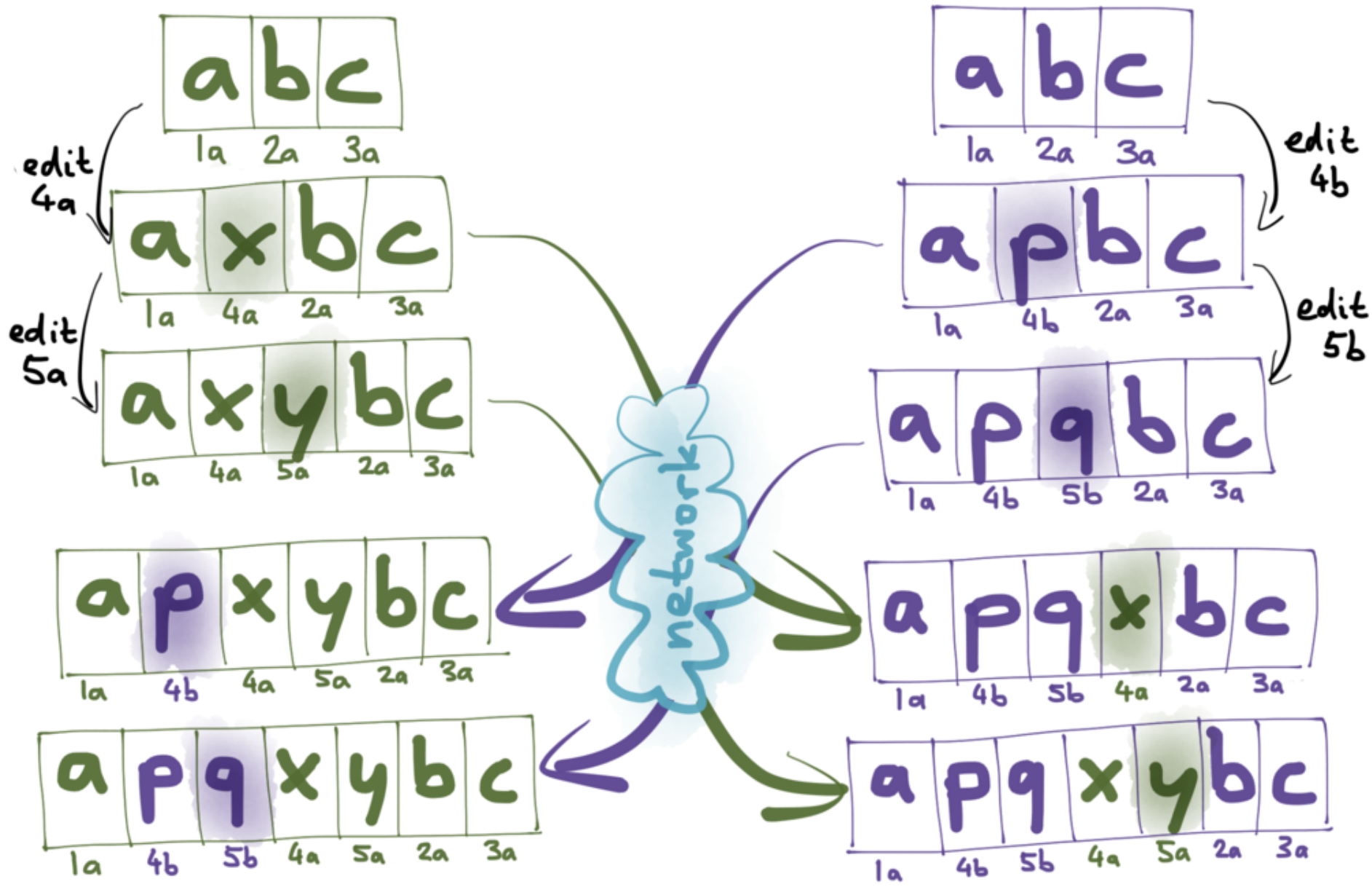
INSERTING IN THE SAME PLACE



INSERTING IN THE SAME PLACE



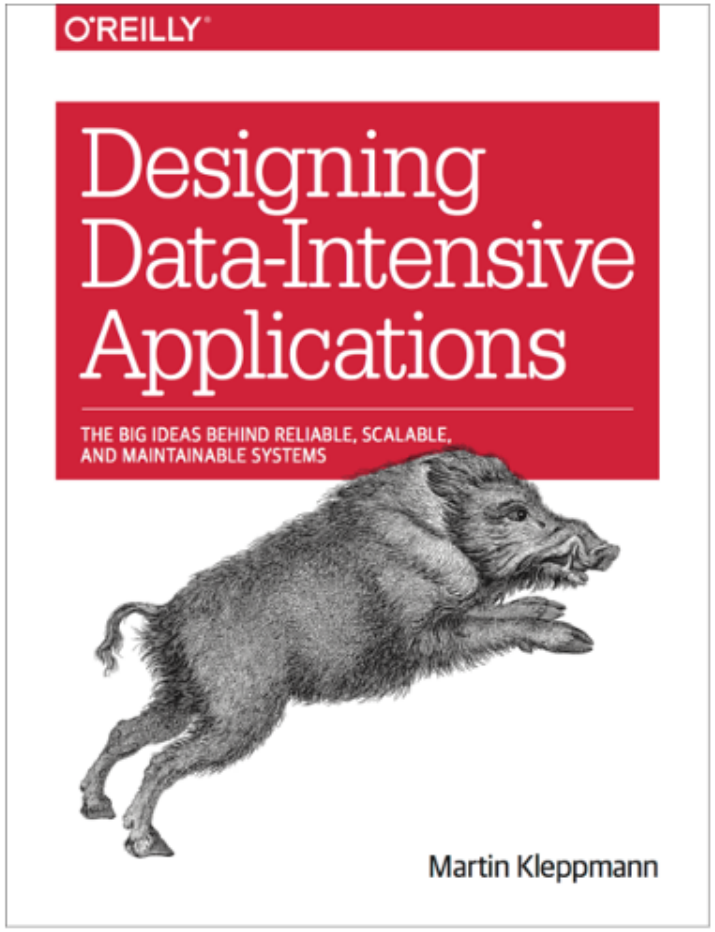
INSERTING IN THE SAME PLACE



Resources

- Automerge: https://github.com/automerge/automerge
- Trellis: https://github.com/automerge/trellis
- Pixelpusher: https://github.com/automerge/pixelpusher
- MPL (WebRTC layer): https://github.com/automerge/mpl
- Hypermerge: https://github.com/automerge/hypermerge
- Dat / Hypercore: https://datproject.org/
- Proving CRDTs correct: https://doi.org/10.1145/3133933
- JSON CRDT: http://arxiv.org/abs/1608.03960
- My book: http://dataintensive.net/

dataintensive.net



@martinkl