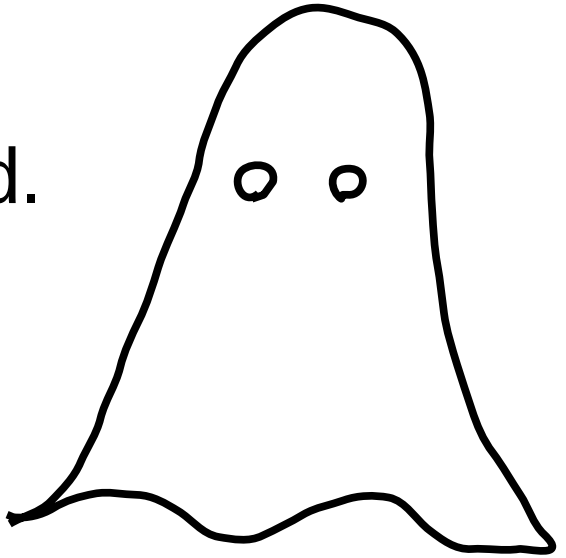
Verifying Casper

Yoichi Hirai
Ethereum Foundation

Cancun, Nov. 1, 2017

The Casper contract

- There will be the Casper contract on each fork
- Validators deposit ETH.
- Casper contract sees validators' signed messages.
- and says which block is finalized.
- Validators get rewards, or get deposits taken away.
- Everything in this talk is seen from one of the Casper contracts.

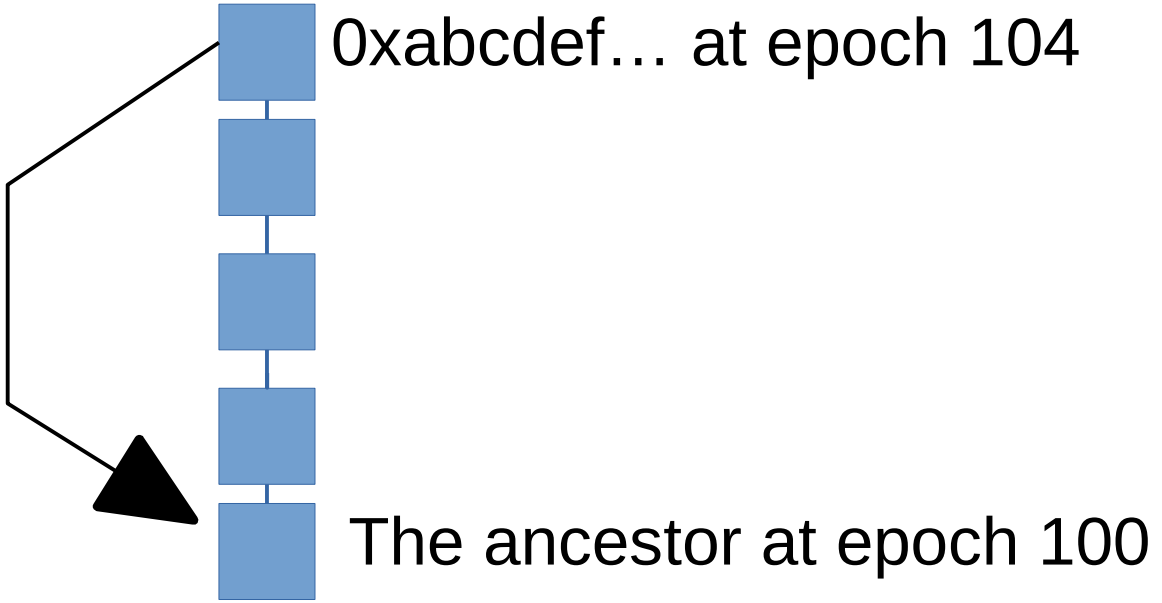


Validators can vote for a block

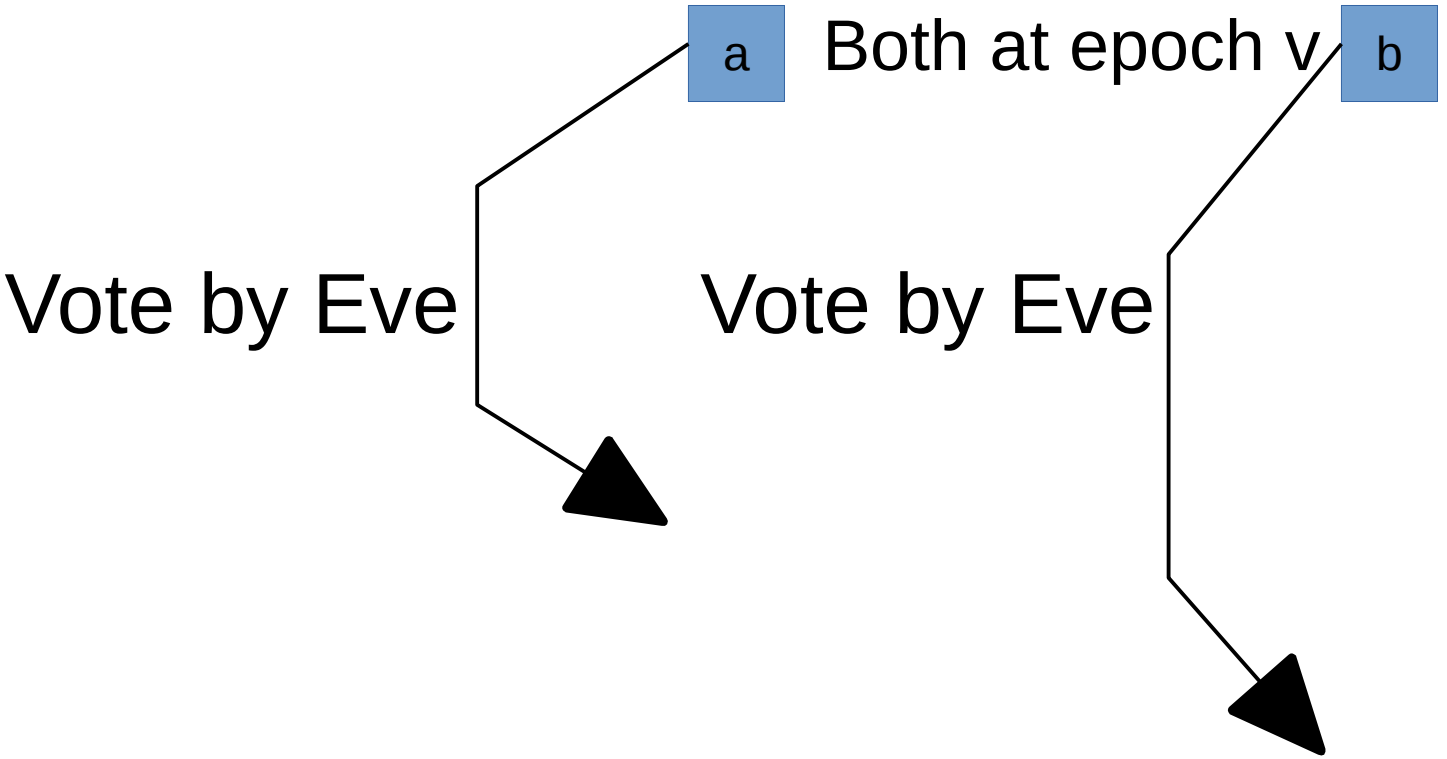
- Vote(epoch, hash, source) by a validator

- Vote(104, 0xabcdef…, 100)

- Voting for 0xabcdef… at epoch 104, citing the ancestor at epoch 100 as a source

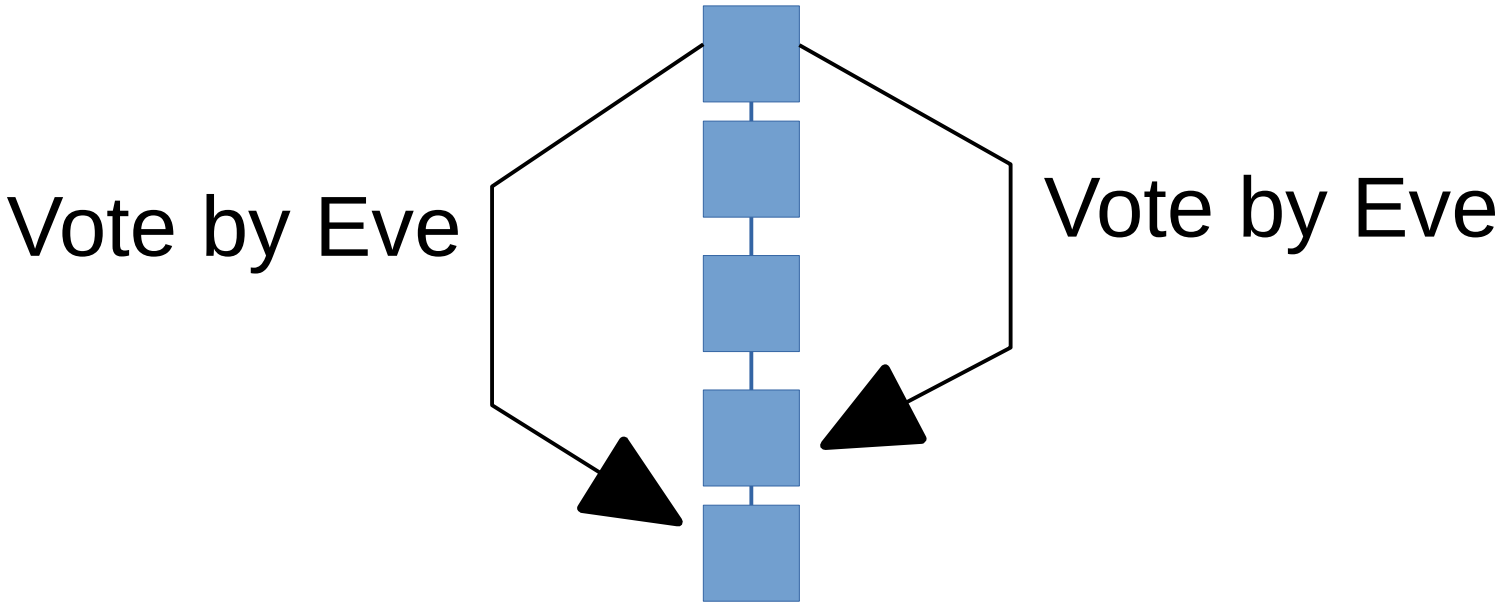


No double voting (diff blocks, same epoch)



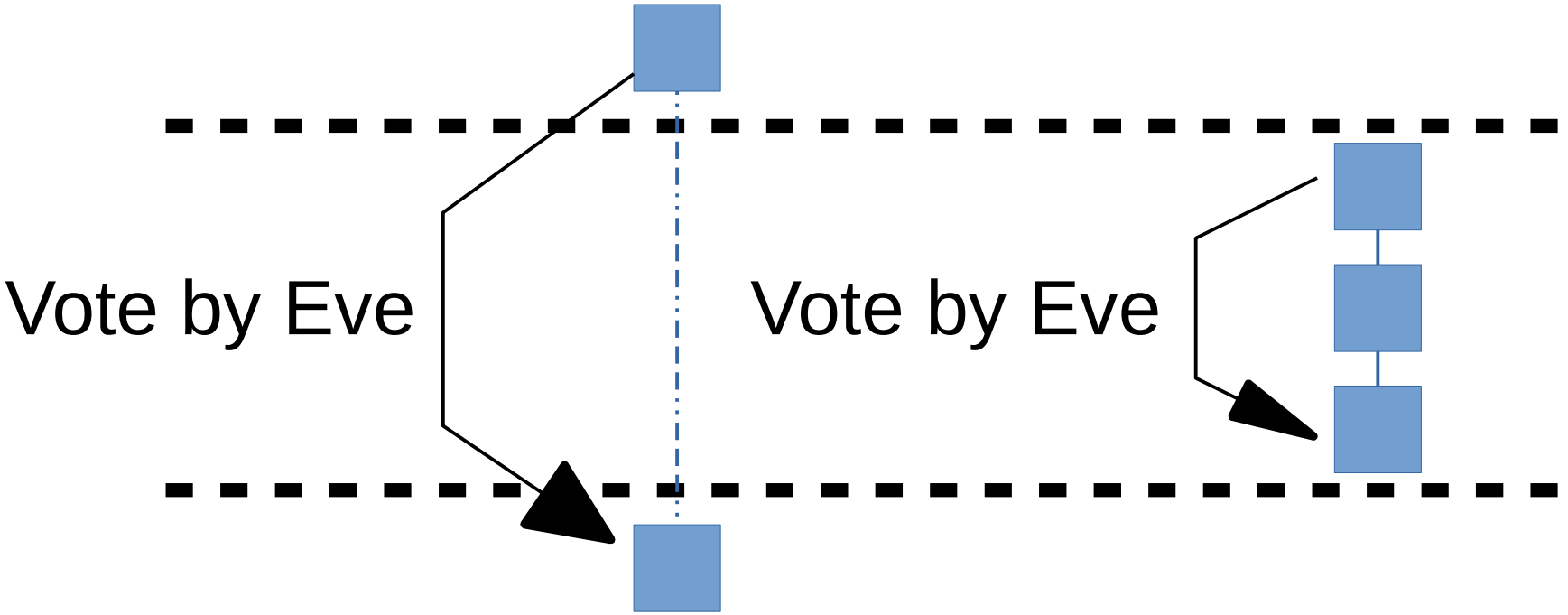
- Eve’s deposits are gone if a and b are different.

No double voting
(same epoch, different source)



- Eve’s deposits are gone
  if sources are different for the same epoch.

No “surrounding”



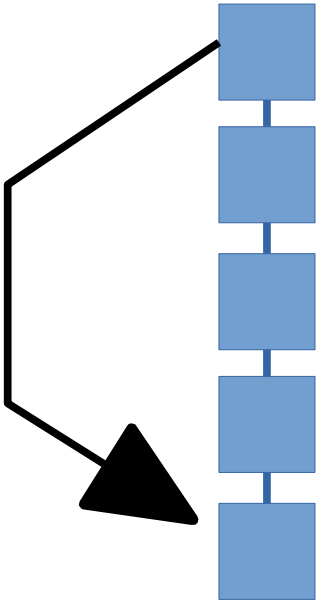
- Eve’s deposits are gone
  if her vote jumps over another
  (judged by epoch heights only)

Sole Responsibility

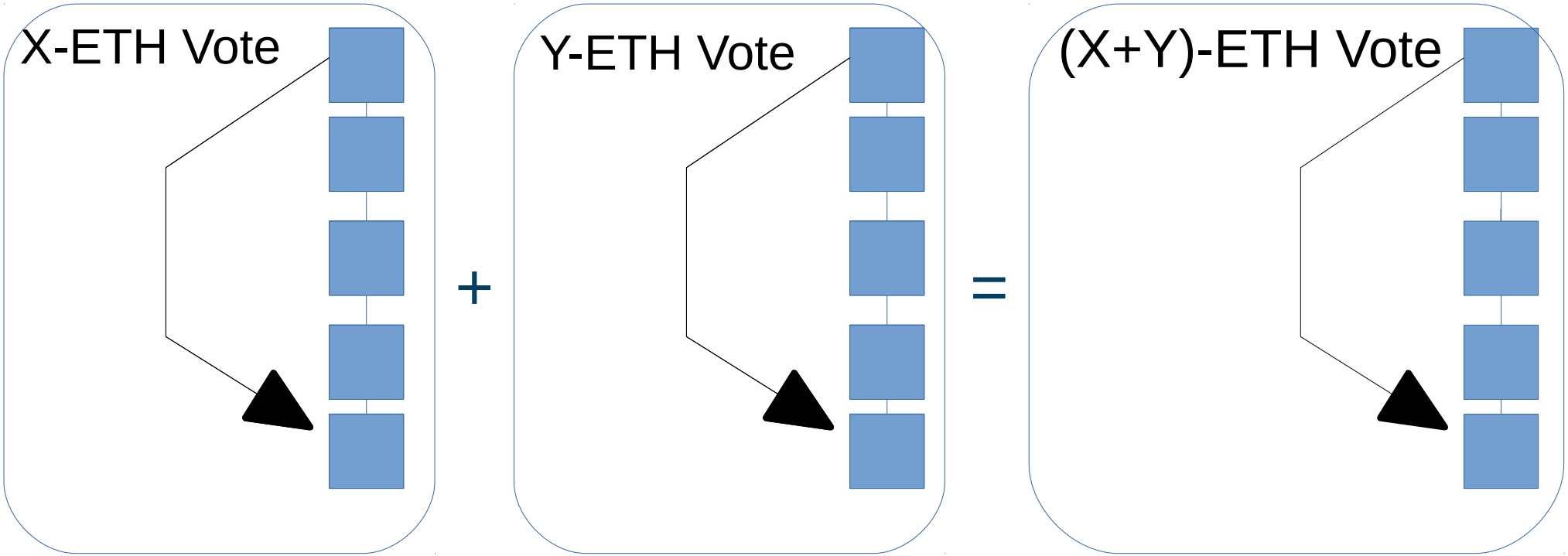
- To avoid being slashed,
  you just need to be careful what you sign.

Validators have deposits
Votes can be weighed by deposits

- X-ETH Vote

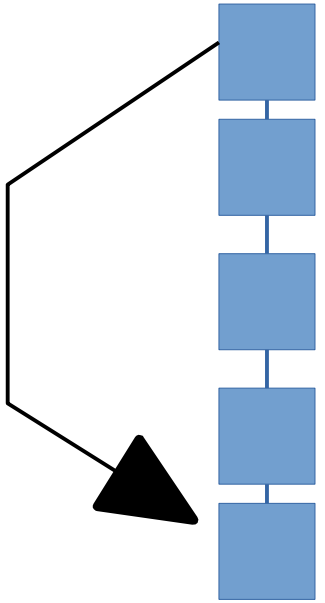


Weights can be added

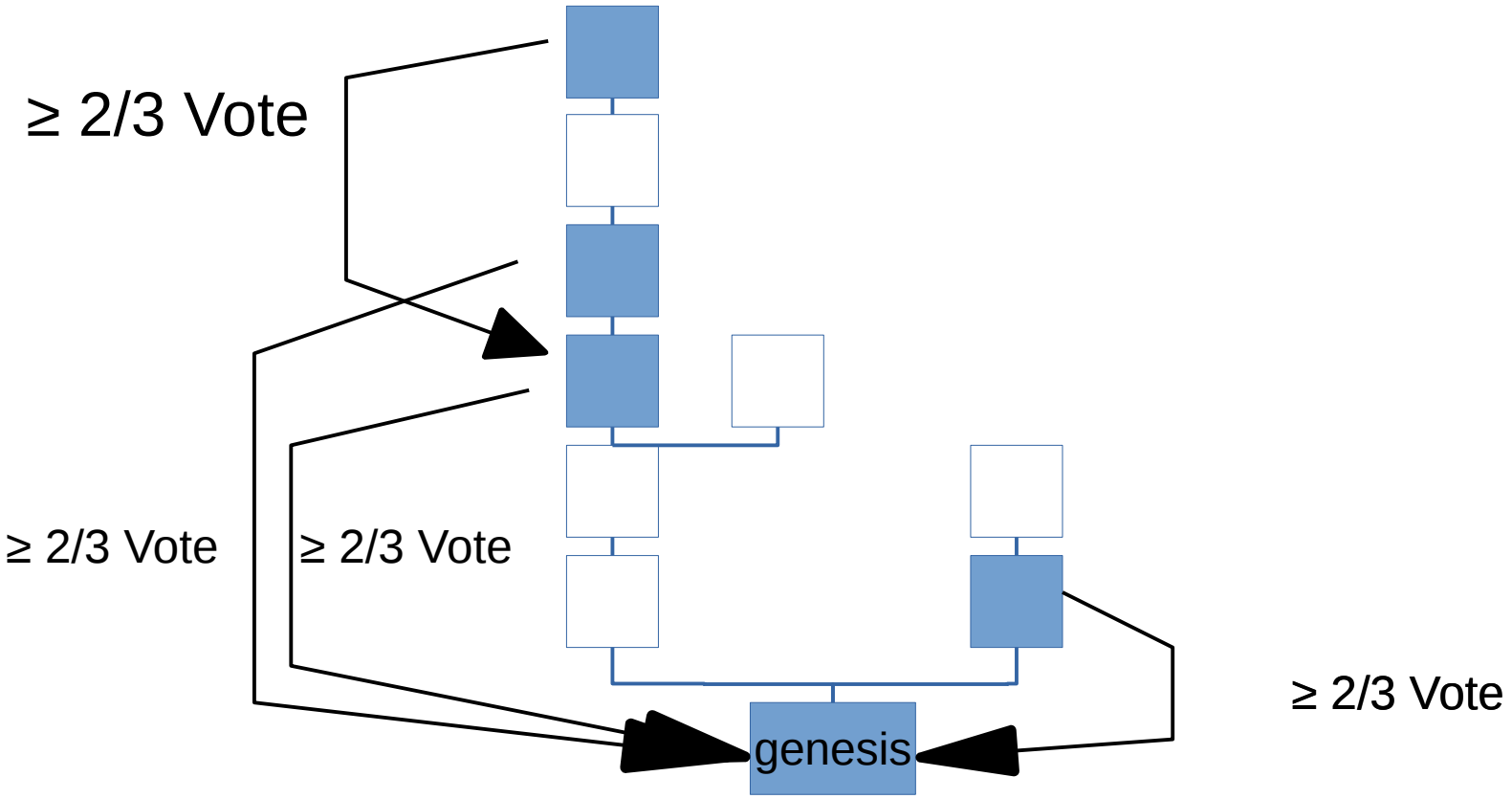


Ratio in the whole deposits

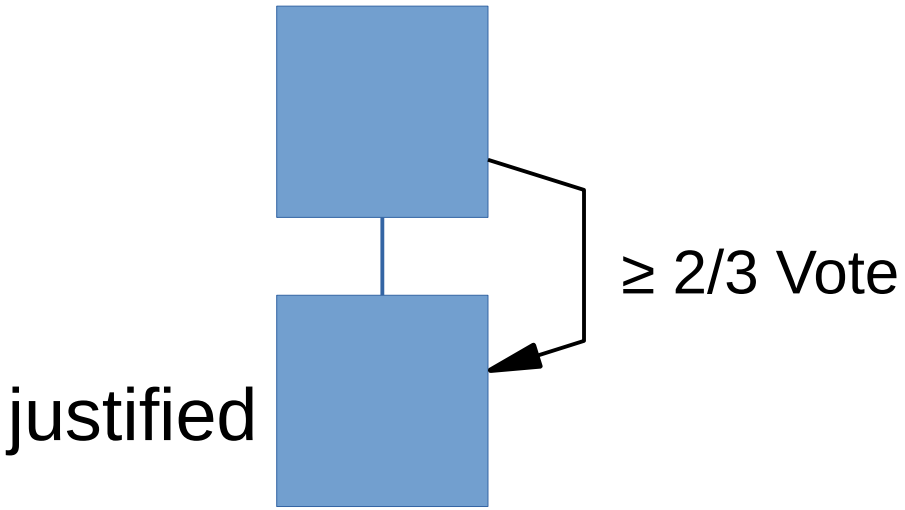
- 2/3 Vote



Justified Blocks



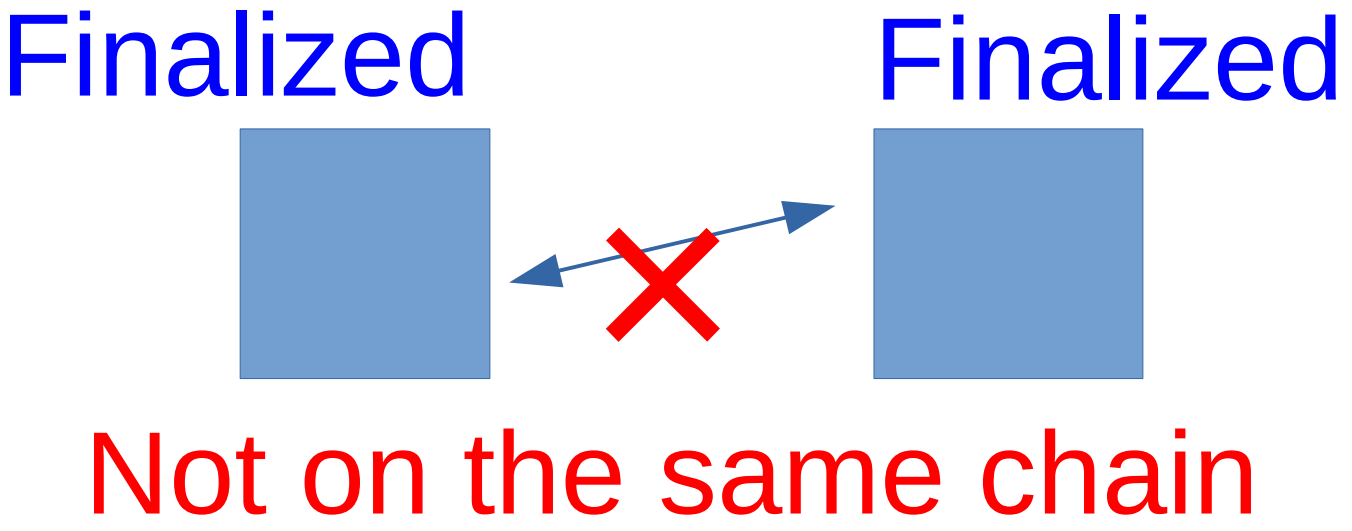
Finalized Blocks



Accoutable Safety

- If finalized blocks fork, we can find 1/3 slashed.

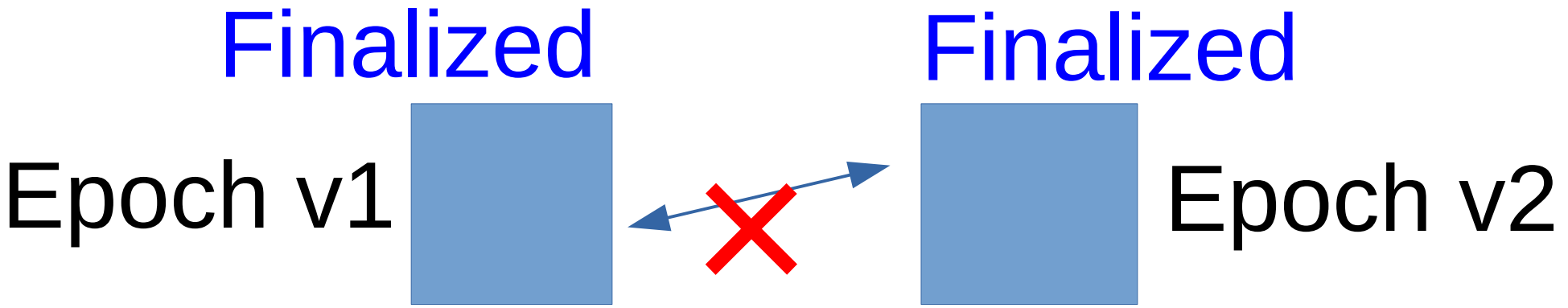
Assumption



Goal

1/3 validators slashed

Assumption



Not on the same chain

v1 = v2 or v1 ≠ v2

Goal

1/3 validators slashed

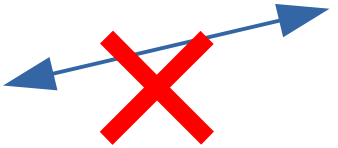
Assumption

https://openclipart.org/detail/191898/bumble-bee

Finalized

Finalized

Epoch v1



Epoch v2

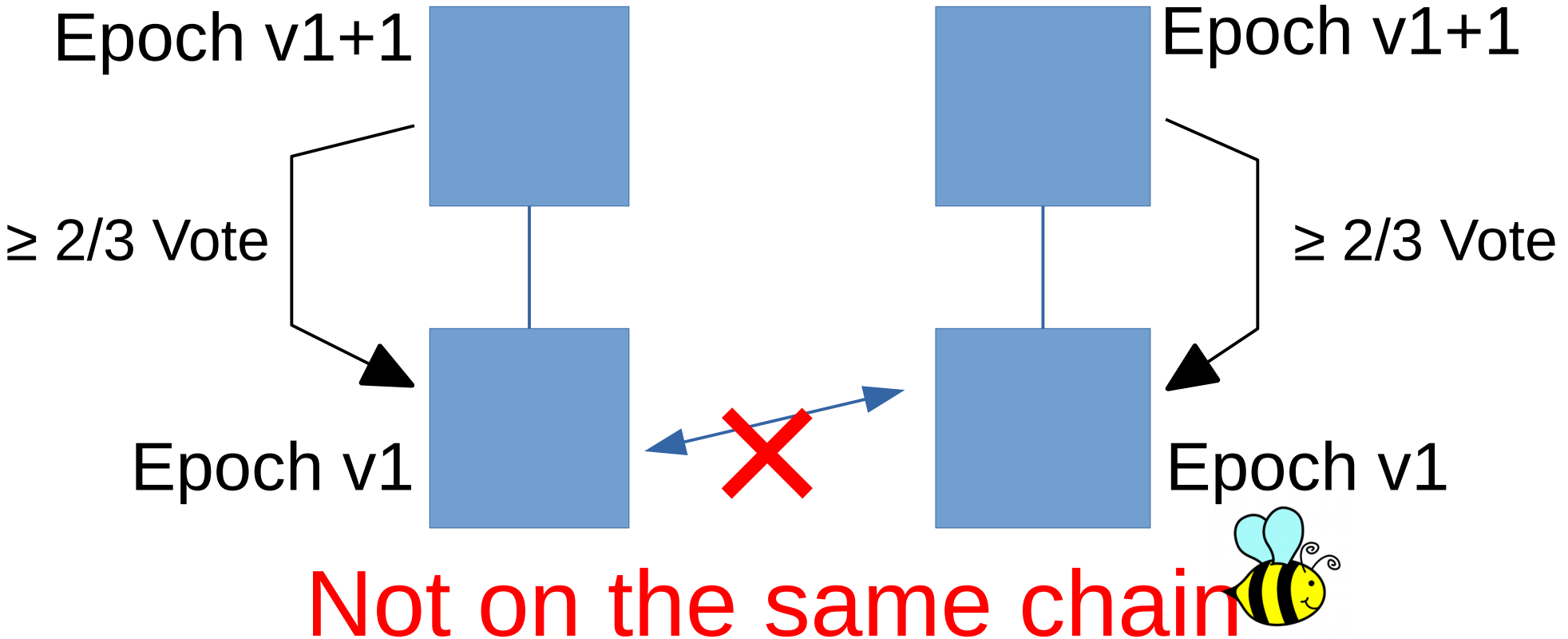
Not on the same chain

v1 = v2

Goal

1/3 validators slashed

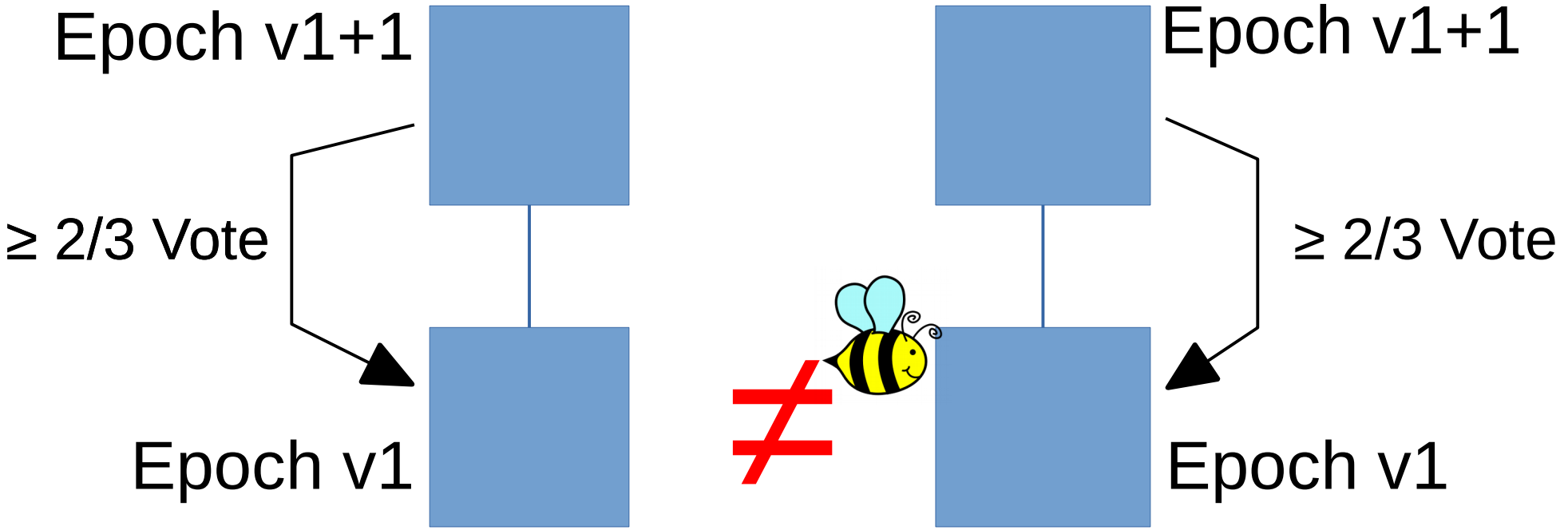
Assumption



Goal

1/3 validators slashed

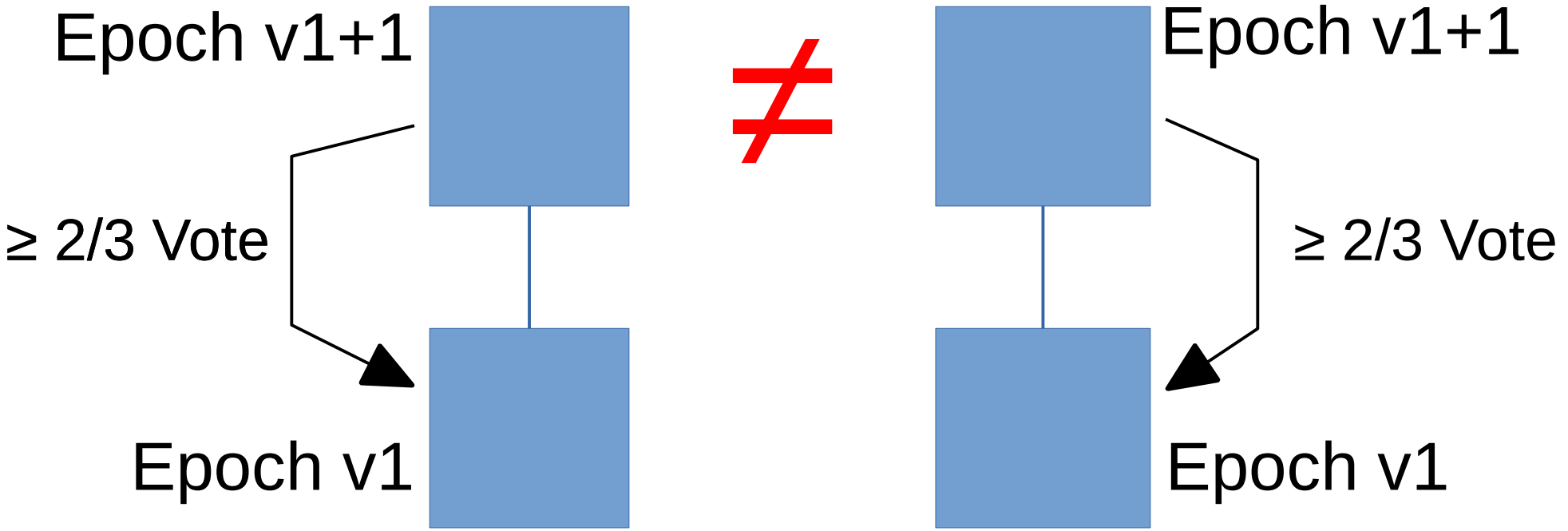
Assumption



Goal

1/3 validators slashed

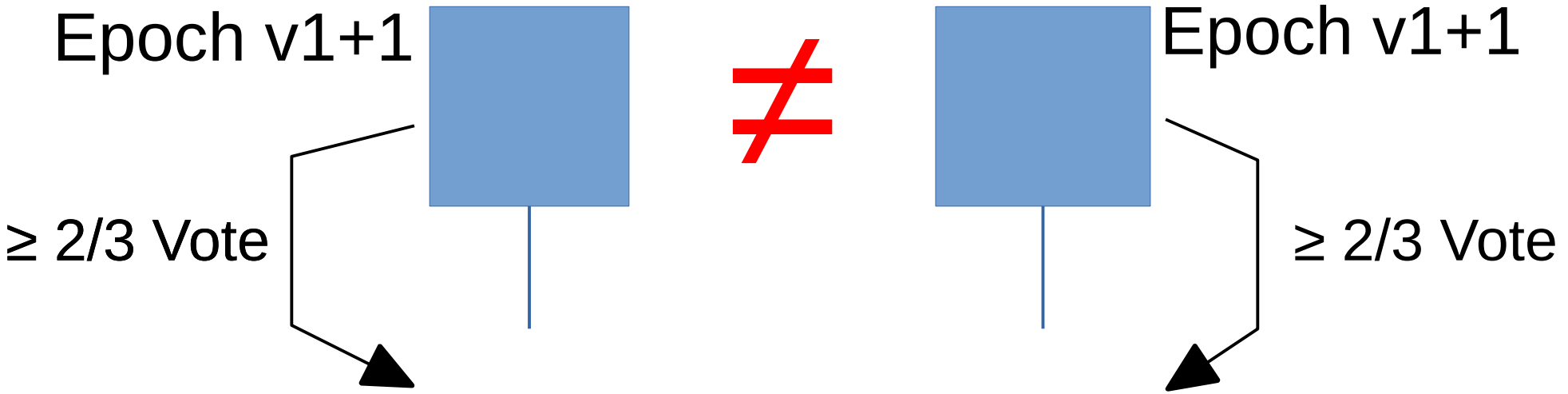
Assumption



Goal

1/3 validators slashed

Assumption

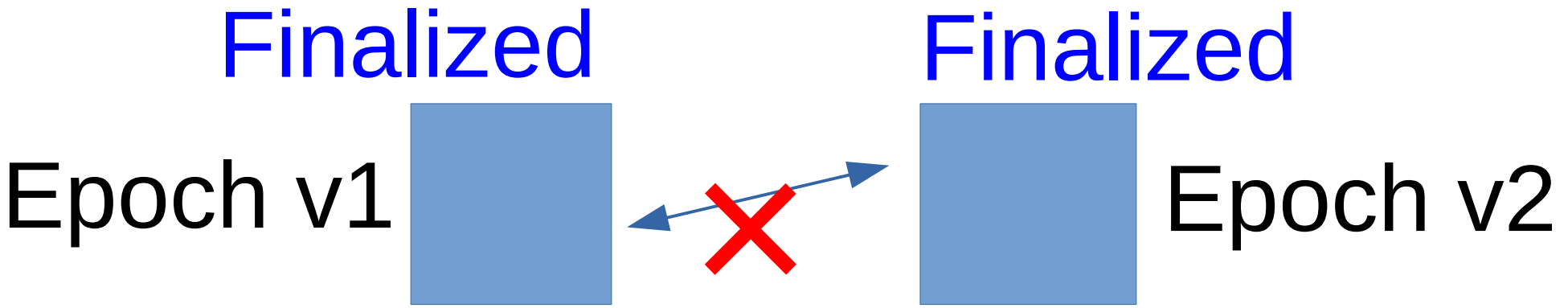


No double voting rule

Goal

1/3 validators slashed

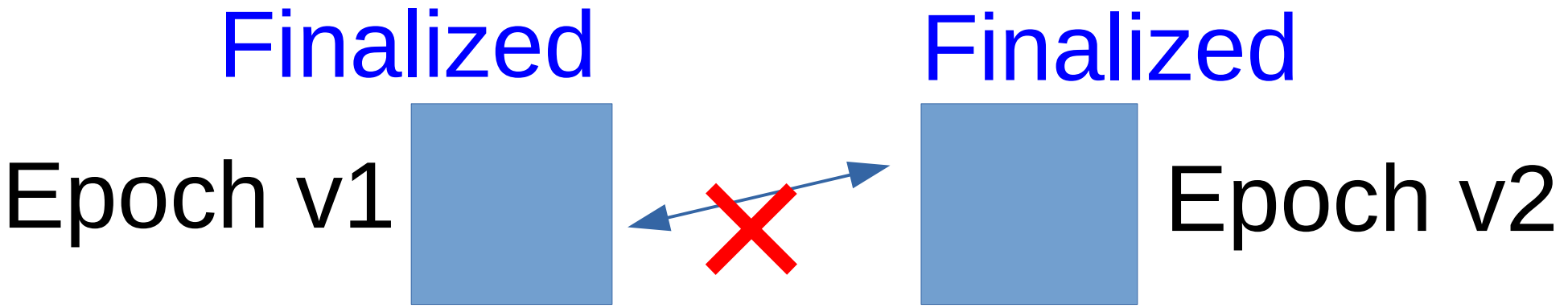
Assumption



Goal

1/3 validators slashed

Assumption



Not on the same chain

v1 > v2 or v1 < v2

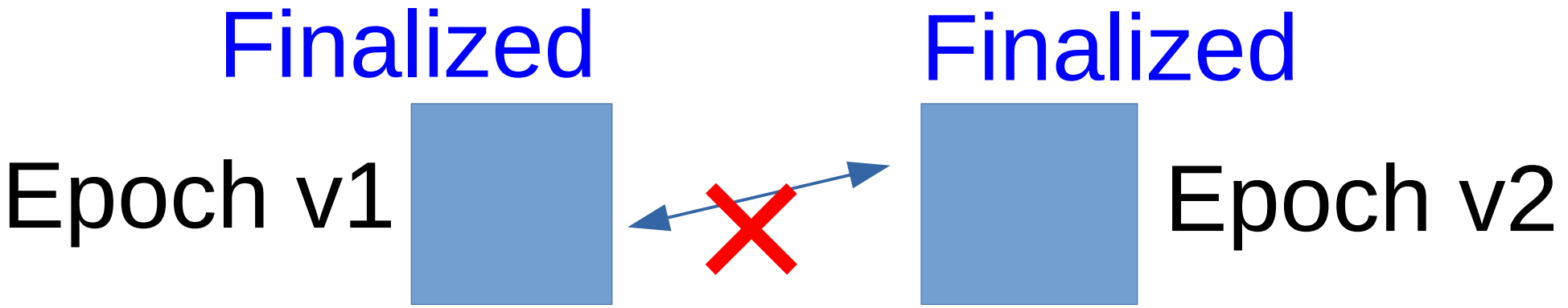
Goal

1/3 validators slashed

WLOG!

- Without Loss Of Generality

Assumption



Not on the same chain

v1 > v2

Goal

1/3 validators slashed

Assumption

Finalized

Epoch v1

h1

Not ancestor-descendant

h2

Finalized

Epoch v2

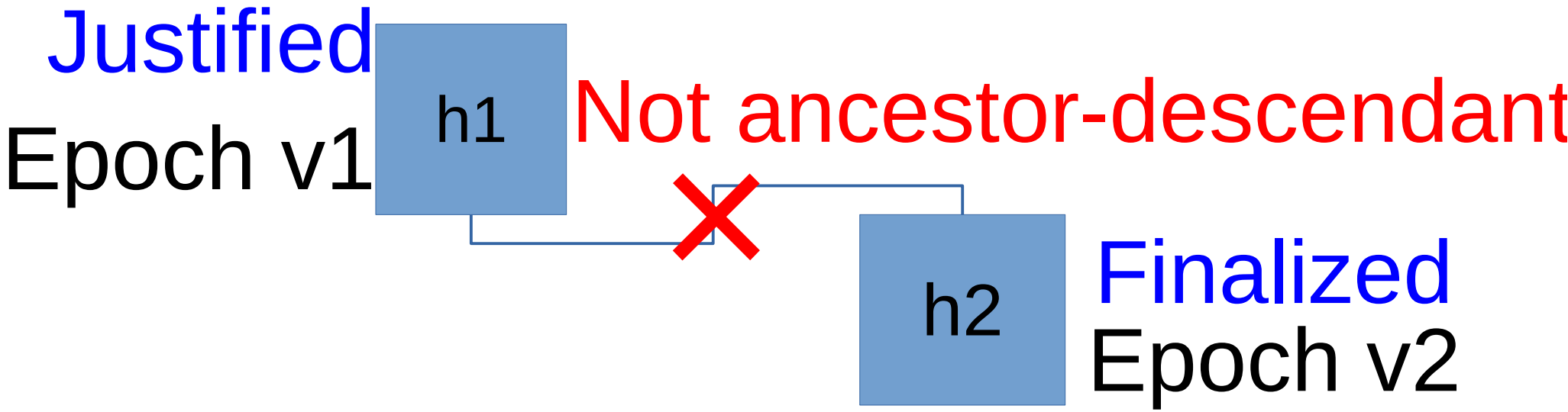
$v1 > v2$

$h1 \neq h2$

Goal

1/3 validators slashed

Assumption



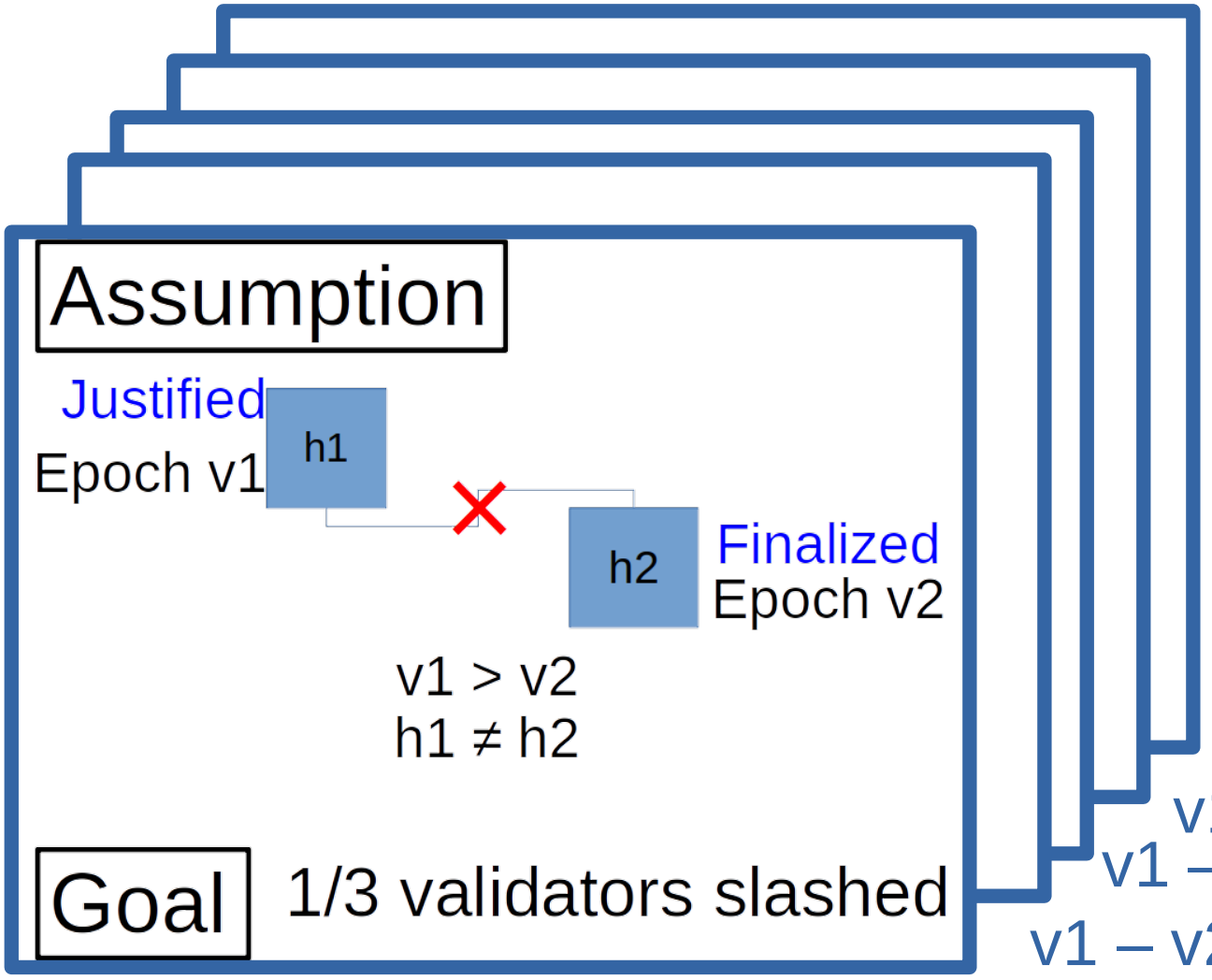
$v1 > v2$

$h1 \neq h2$

Goal

1/3 validators slashed

Induction on v1 - v2



What next?

v1 – v2 = 5 … done

v1 – v2 = 4 … done

v1 – v2 = 3 … done

v1 – v2 = 2 … done

v1 – v2 = 1 … done

Assumption

Justified

Epoch v1

h1

Not ancestor-descendant

h2

Finalized

Epoch v2

$v1 > v2$

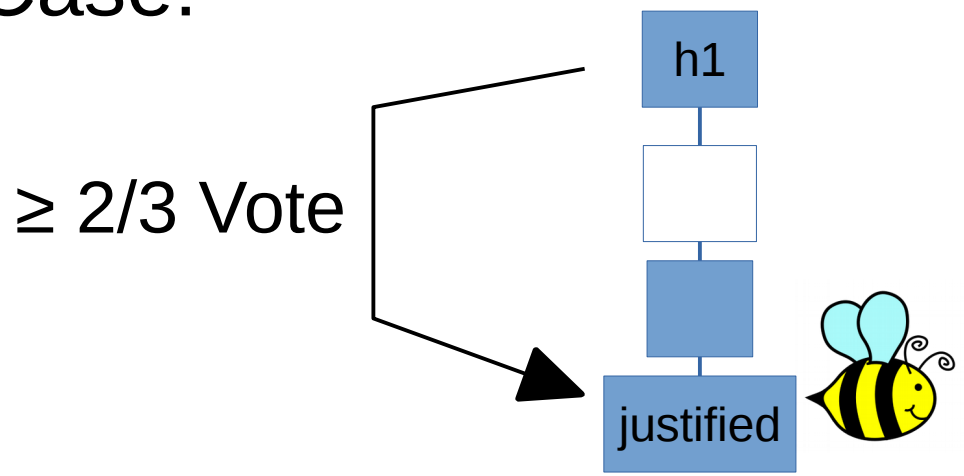
$h1 \neq h2$

Goal

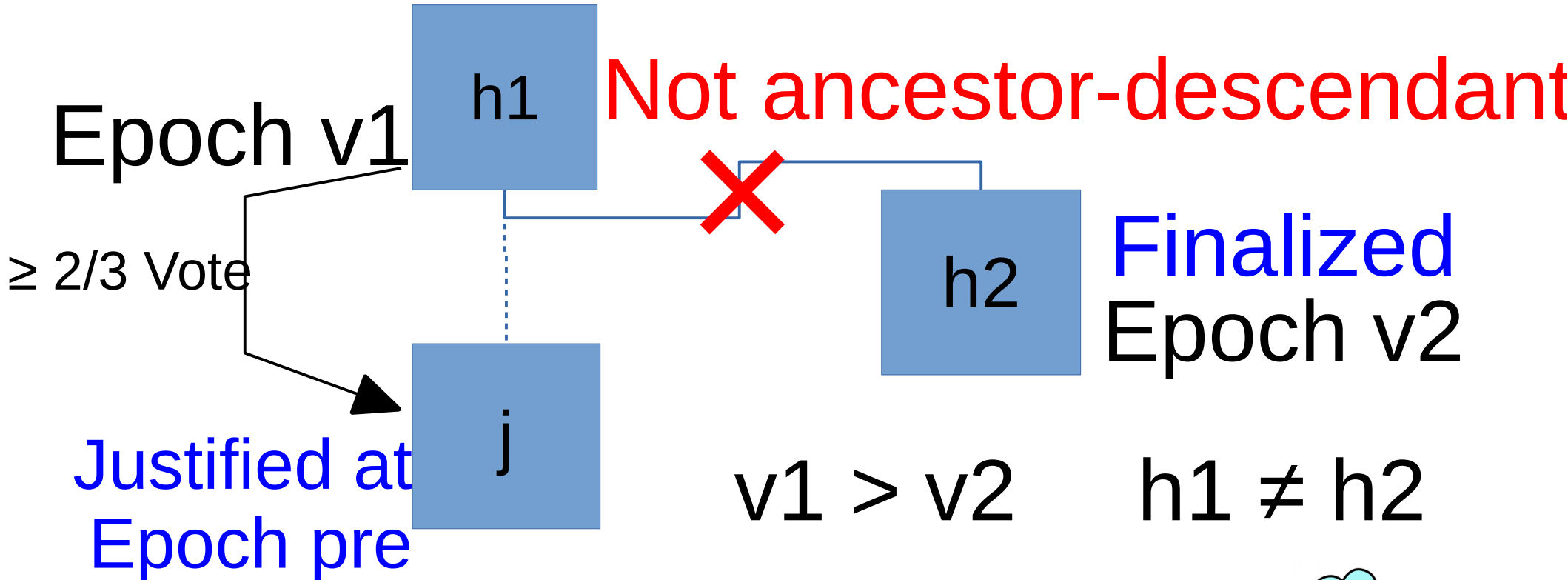
1/3 validators slashed

h1 is justified

- Case: h1 is genesis and v1 = 0
  (cannot be the case because v1 > v2)
- Case:



Assumption



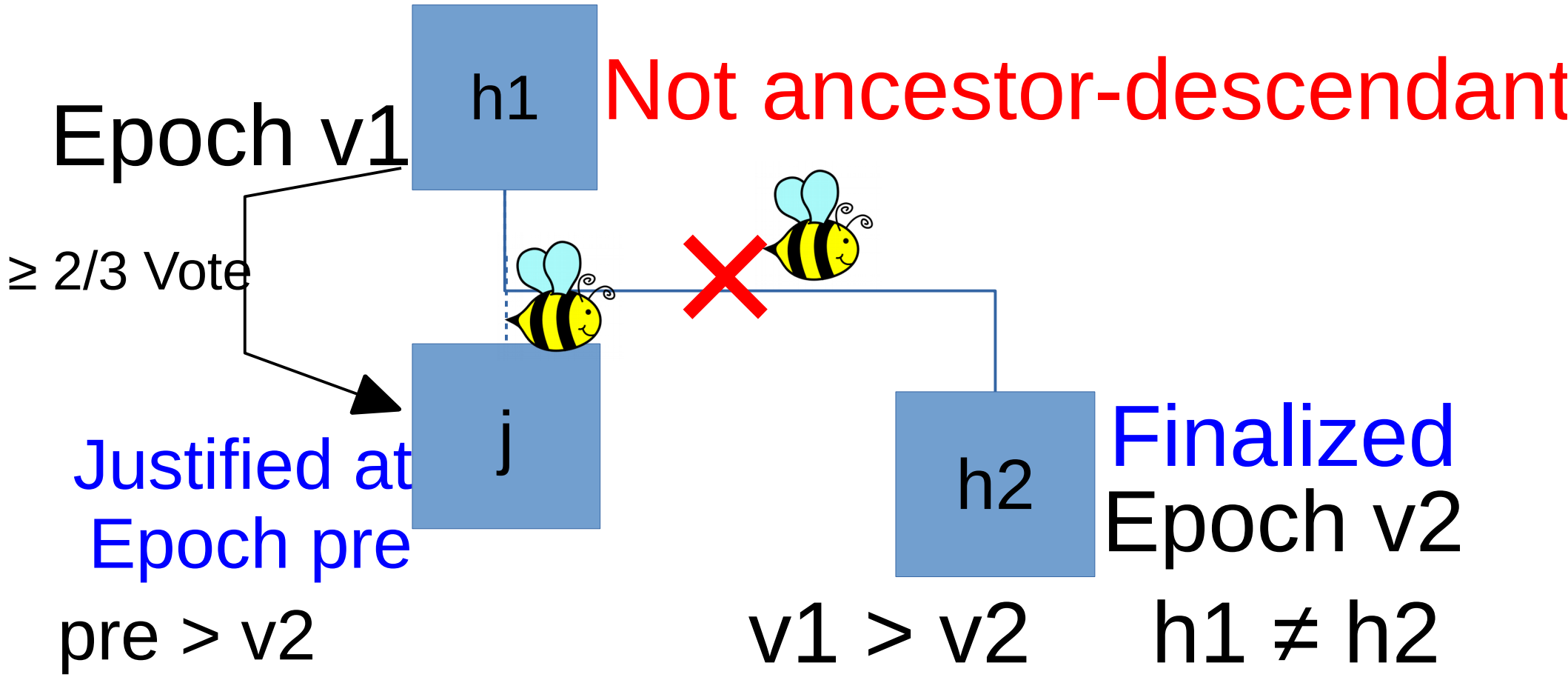
v1 > v2 h1 ≠ h2

pre > v2 or pre = v2 or pre < v2

Goal

1/3 validators slashed

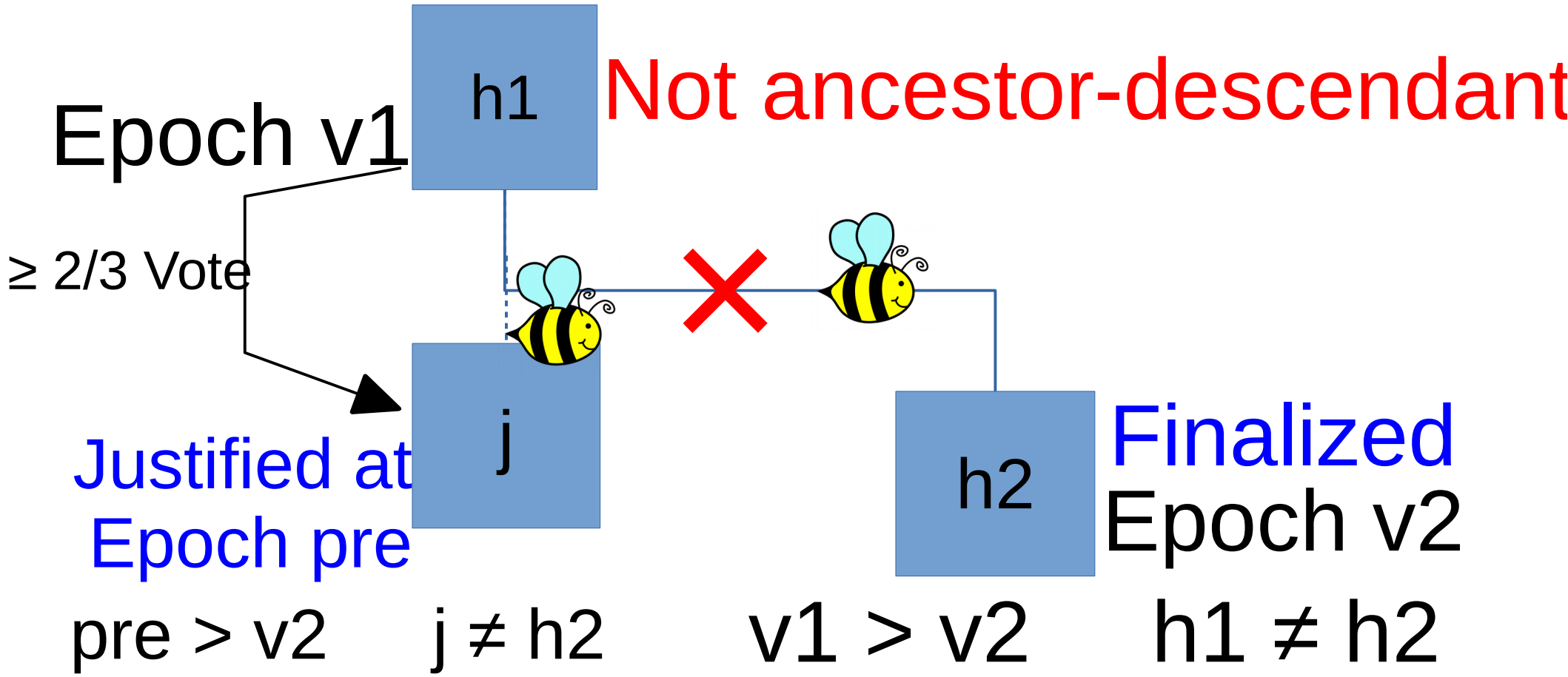
Assumption



Goal

1/3 validators slashed

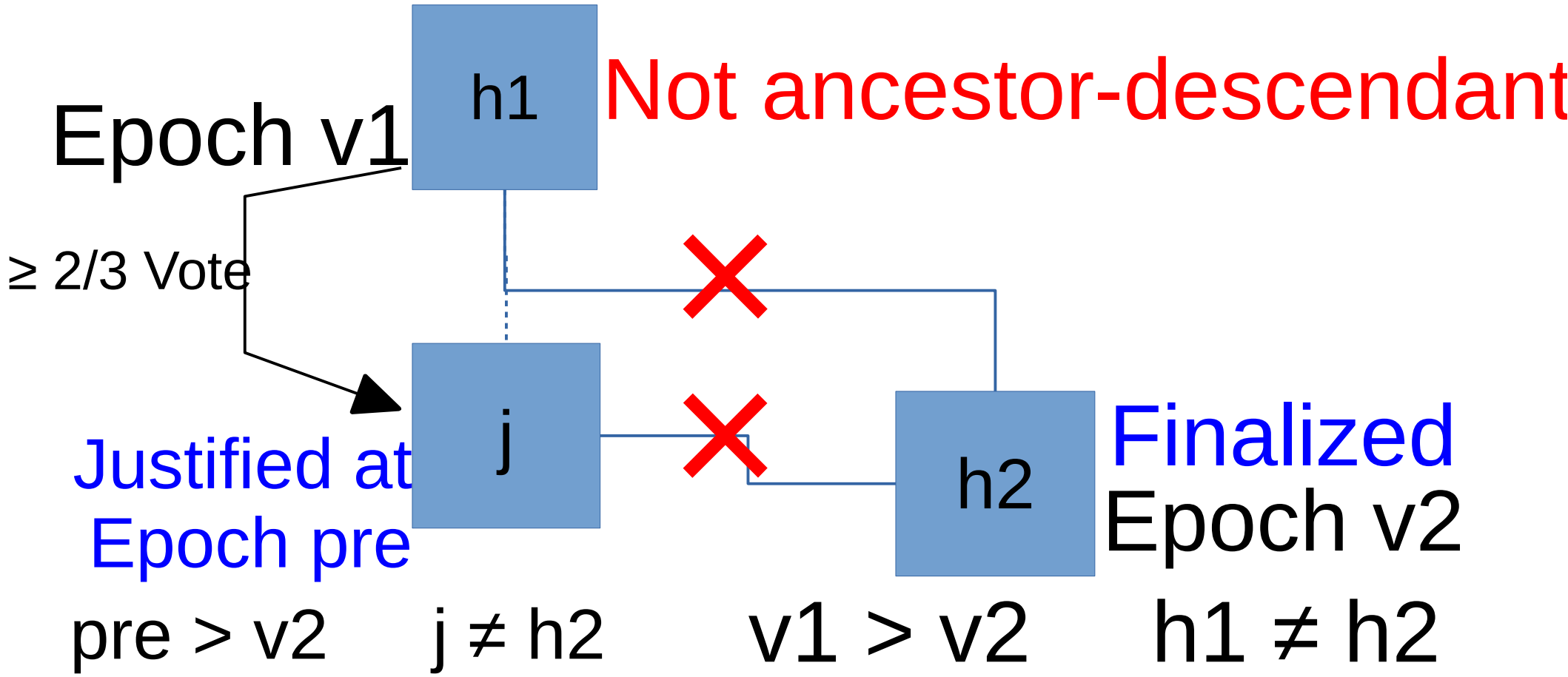
Assumption



Goal

1/3 validators slashed

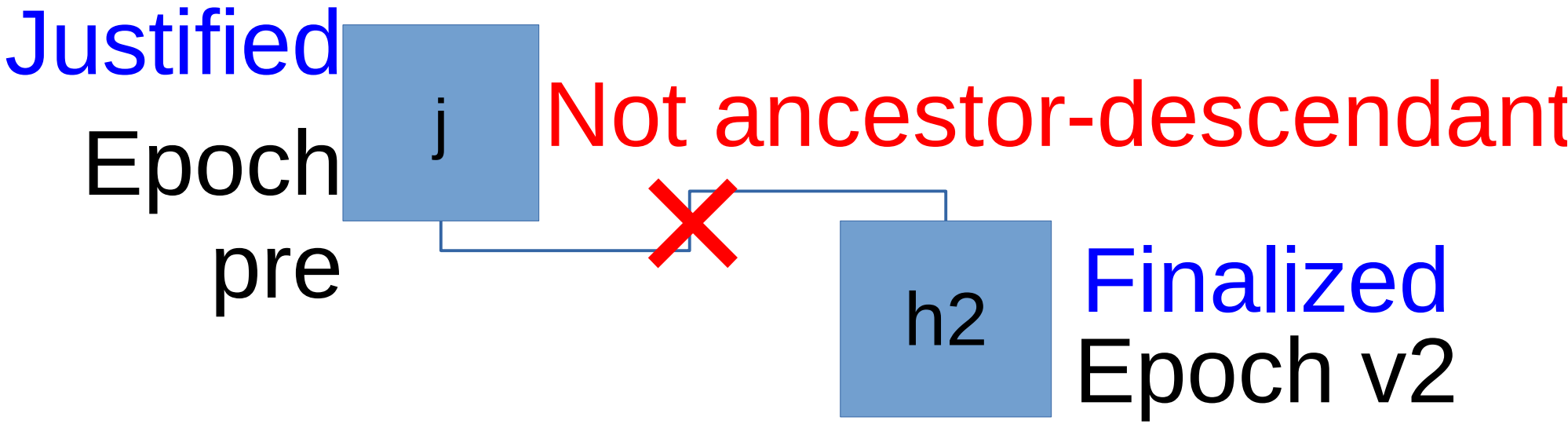
Assumption



Goal

1/3 validators slashed

Assumption



pre > v2

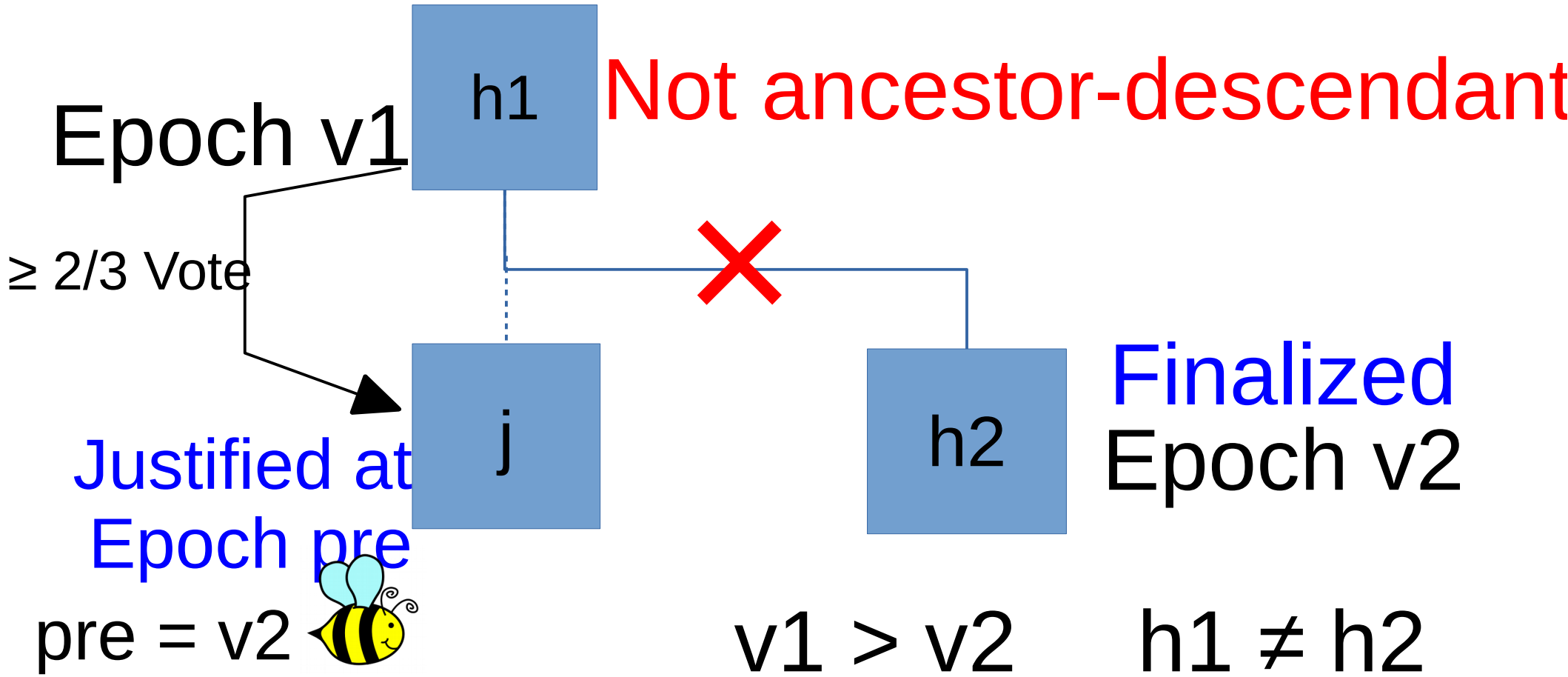
j ≠ h2

The same thing, with a smaller v1 (now pre).

Goal

1/3 validators slashed

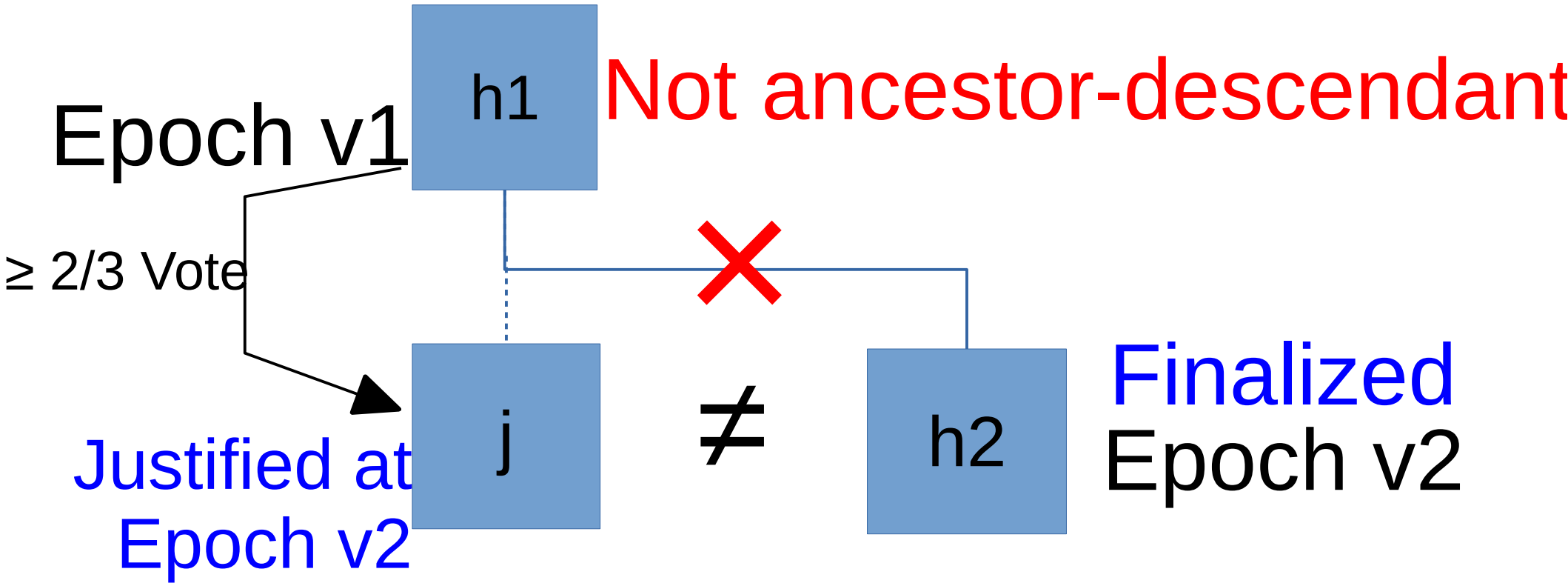
Assumption



Goal

1/3 validators slashed

Assumption

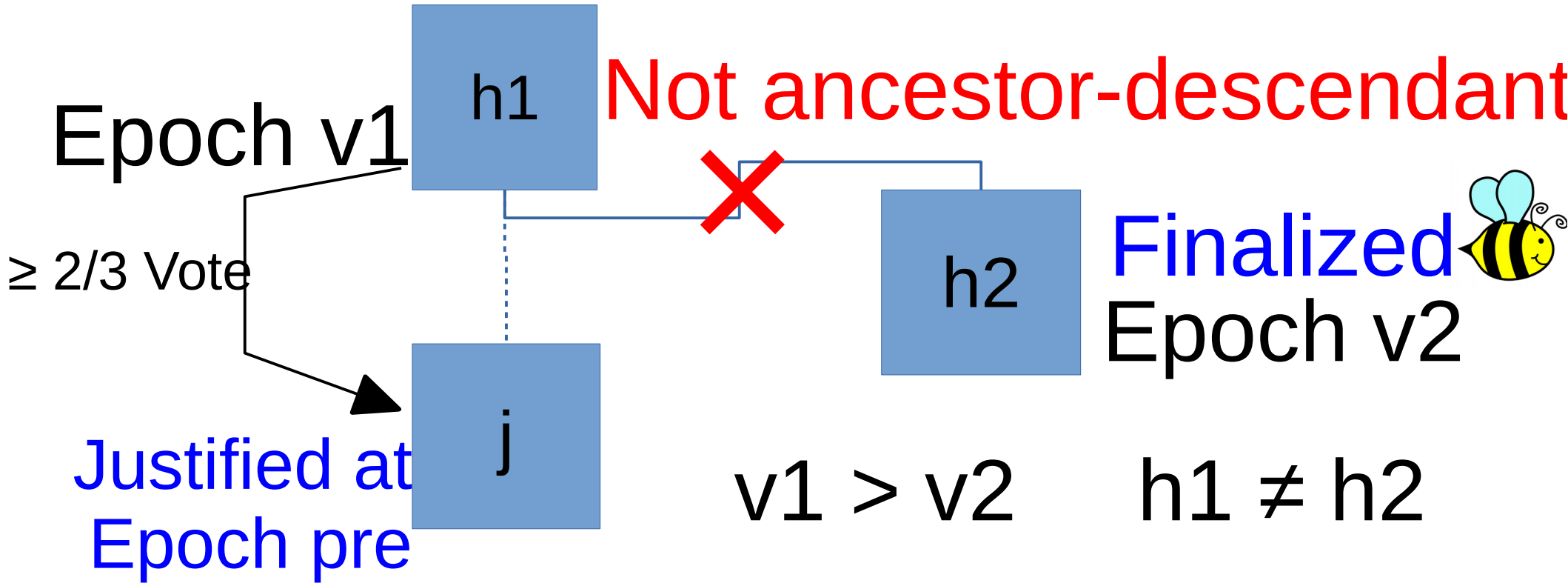


Different blocks are justified on the same block!

Goal

1/3 validators slashed

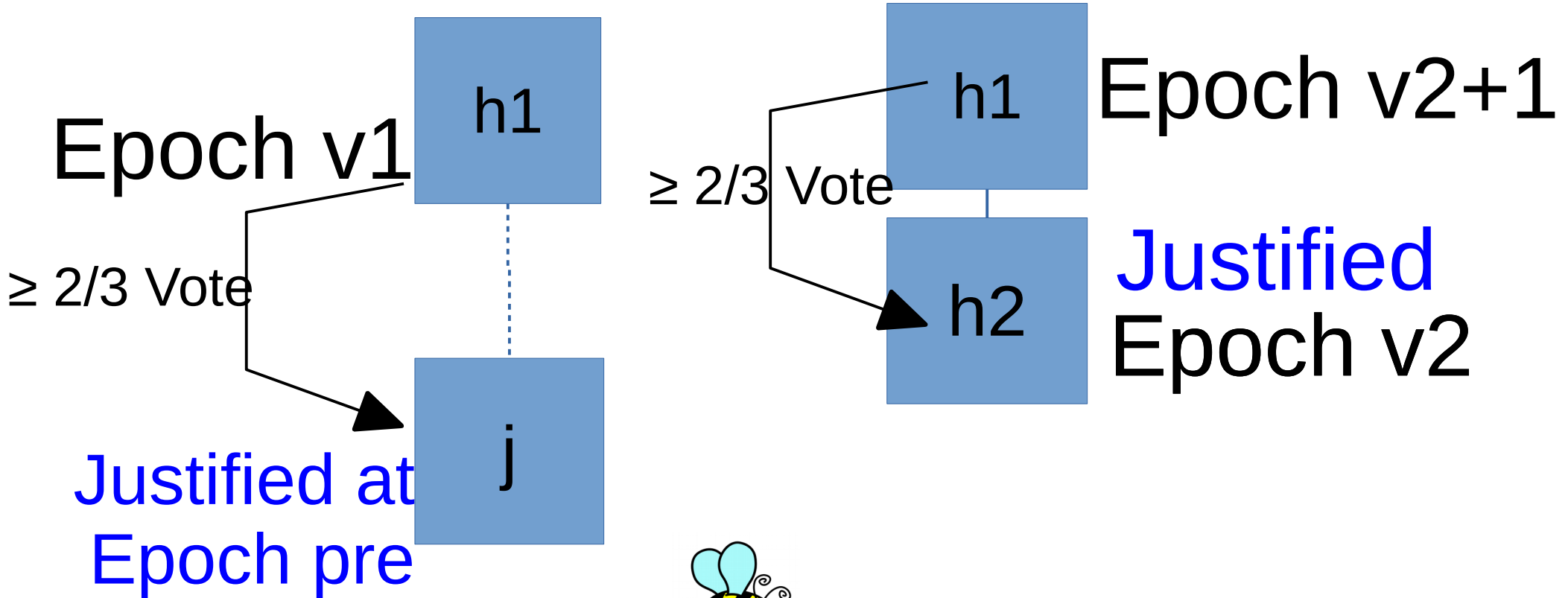
Assumption



Goal

1/3 validators slashed

Assumption

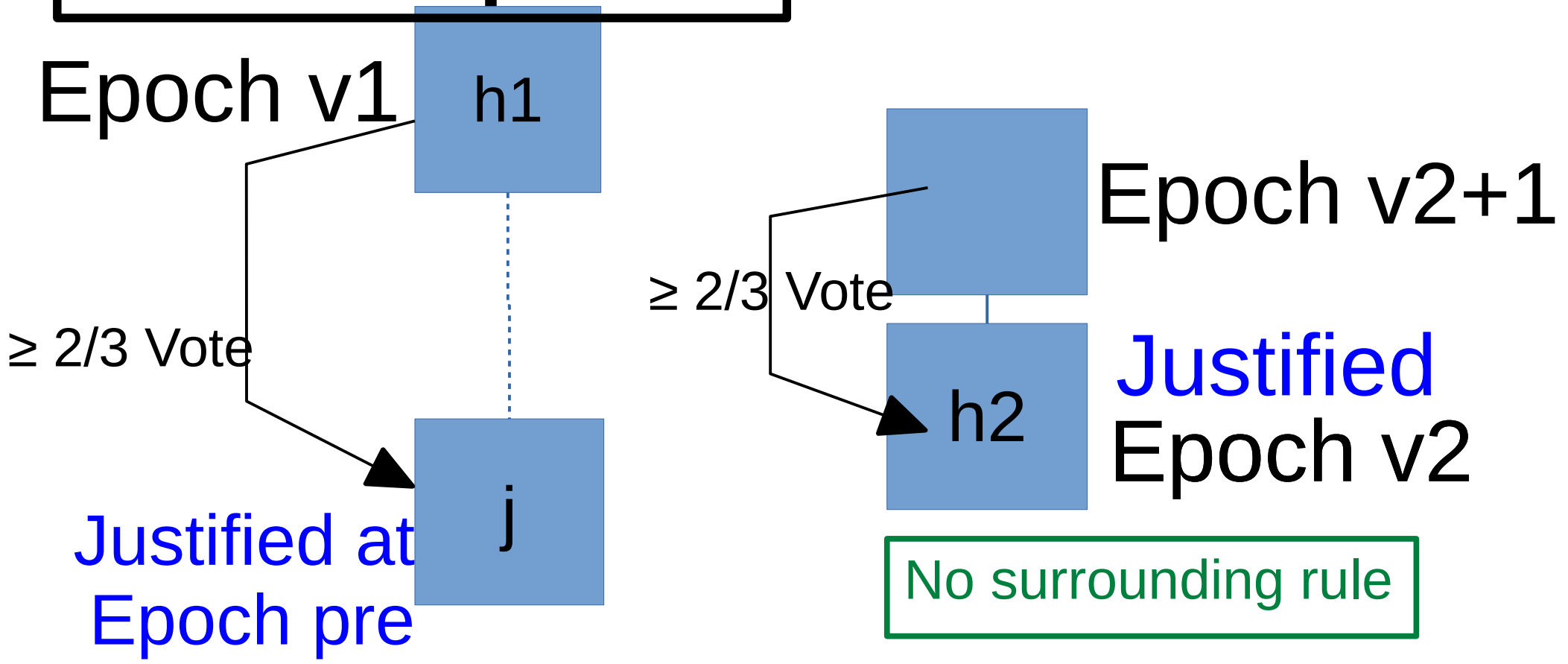


v1 > v2 + 1 or v1 = v2 + 1 pre < v2

Goal

1/3 validators slashed

Assumption



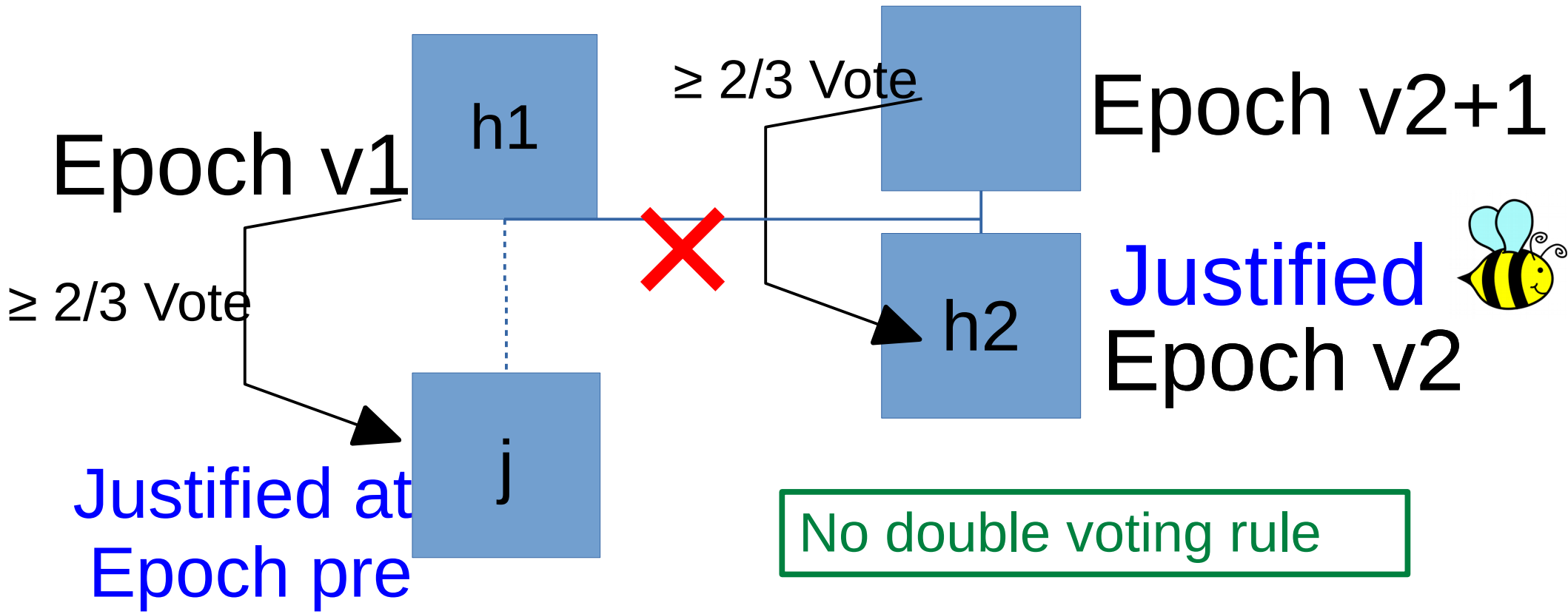
v1 > v2 + 1

pre < v2

Goal

1/3 validators slashed

Assumption



v1 = v2 + 1

pre < v2

Goal

1/3 validators slashed

I checked it with machines first

	Static Validator Set		Dynamic Validator Set	
	Safety	Plausible Liveness	Safety	Plausible Liveness
2-message Casper (obsolete)	done	done	done	obsolete
1-message Casper	Done here	Not yet	Not yet	Not yet

https://github.com/pirapira/pos

- I’ve shown you the rigor a machine forces.