Reactor

node.view ≥ node.qc.view
node.payload

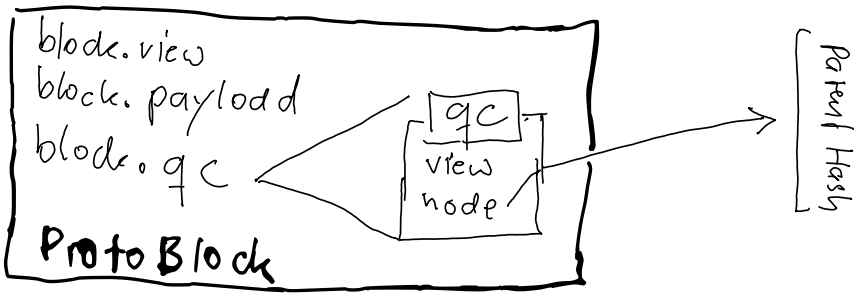
block.qc := qc''
qc''.view = b''.view
qc''.node = b''

broadcast by primary

ProtoBlock Proposal

Algorithmic constraints

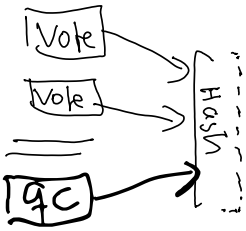
> **Definition** [ChHS Line 10]
block.parent ≟ block.qc.node



Contract (i)

Vote: Signature over ProtoBlock b

Vote: $view \stackrel{!}{=} b.view$



Aggregation $\Rightarrow$ $qc.node = b$

Soure: Utilities Pg 7 function QC

$qc.view = b.view$

$\hookrightarrow$ $qc.view = qc.node.view$

Constraint (ii): Utilities Pg 6

$qc.view \stackrel{!}{=} qc.node.view$

Contract — Line 17 in Utilities

Source: Page 7, top: function QC

Required for validity of QC

Slashable offence for "authorized QC" meaning:

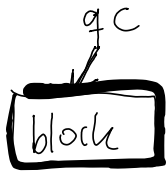
$\Rightarrow$ • QC can only be published as part of Block

or

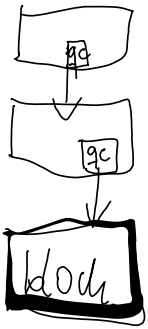
• QC must be signed by primary

Contract (iii)

node.justify.node } is ancestor of node (this)
node.qc.node }

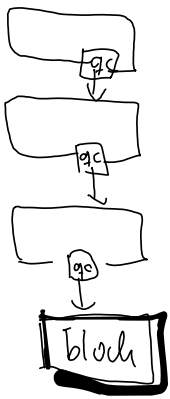


prepare QC: 2/3+ oppove of building on top of this block
⇒ I will build on top



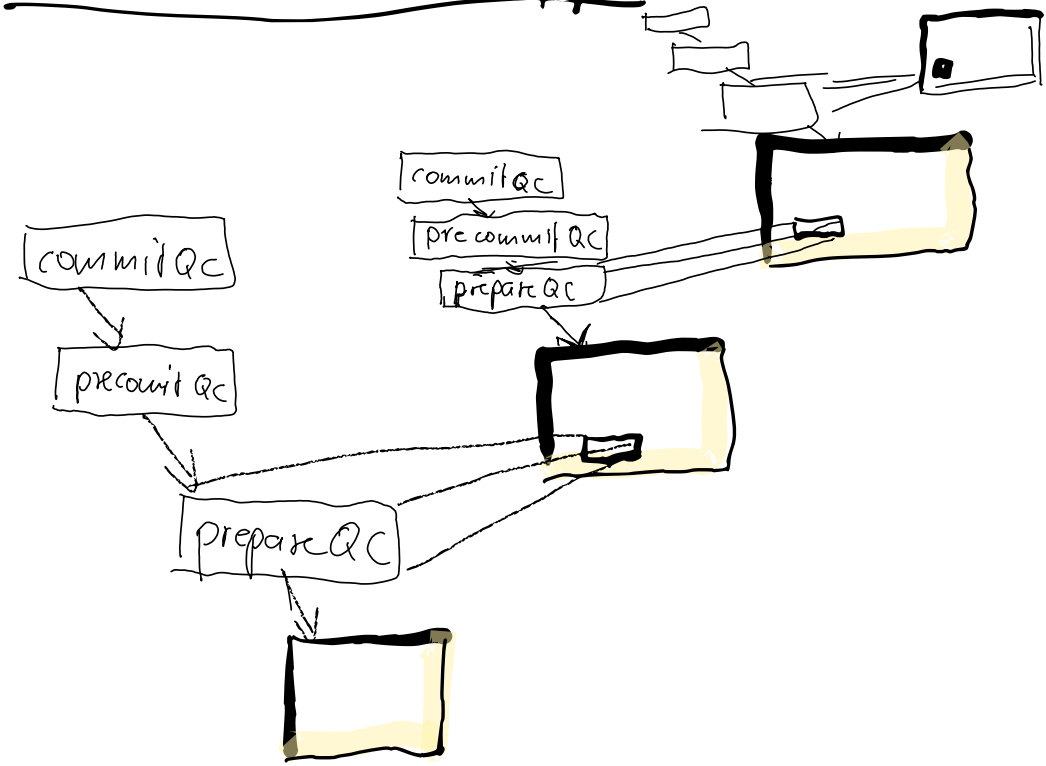
I know that 2/3+ will build on top of block
⇒ I can afford to only vote for extensions on top of "block"
i.e: I lock on 'block'

Note: unlocking is still possible (Liveness cond)



I know that 2/3+ will only vote for decendents of "block"
⇒ "block is finalized"

Basic HotStuff



Chained HotStuff: Skipping Rounds