



Time in Ethereum

Implications of replacing our dear friend Poisson

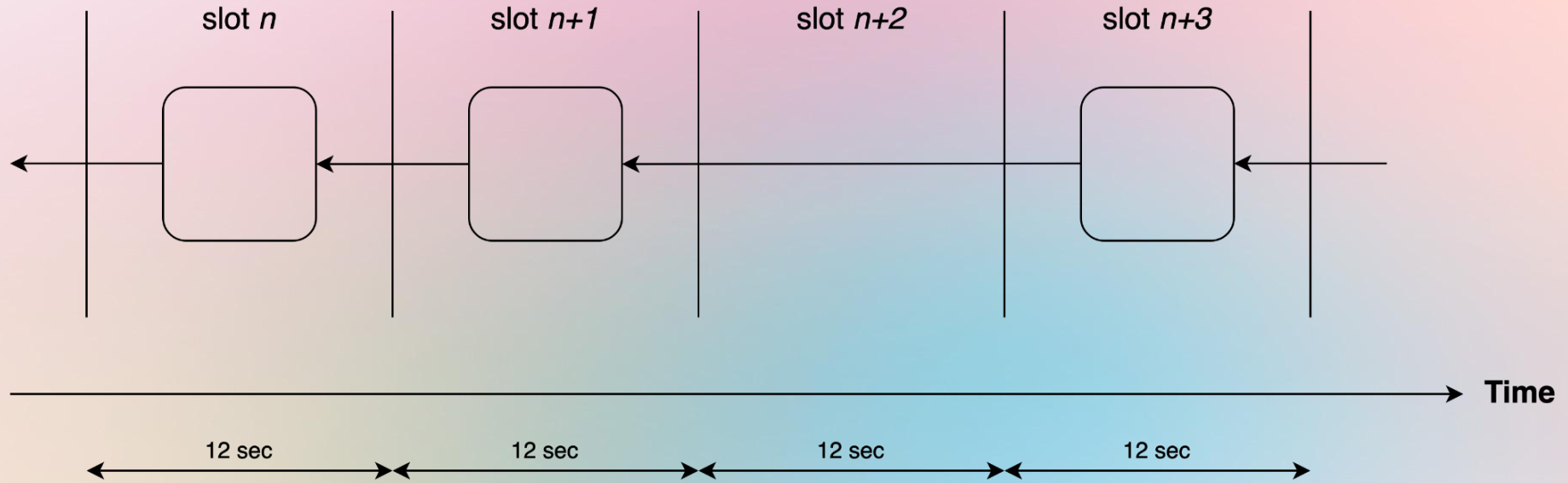
Caspar Schwarz-Schilling

Robust Incentives Group (RIG), Ethereum Foundation

wtf *is* time ?! *

*** next question please...**

Time in Ethereum





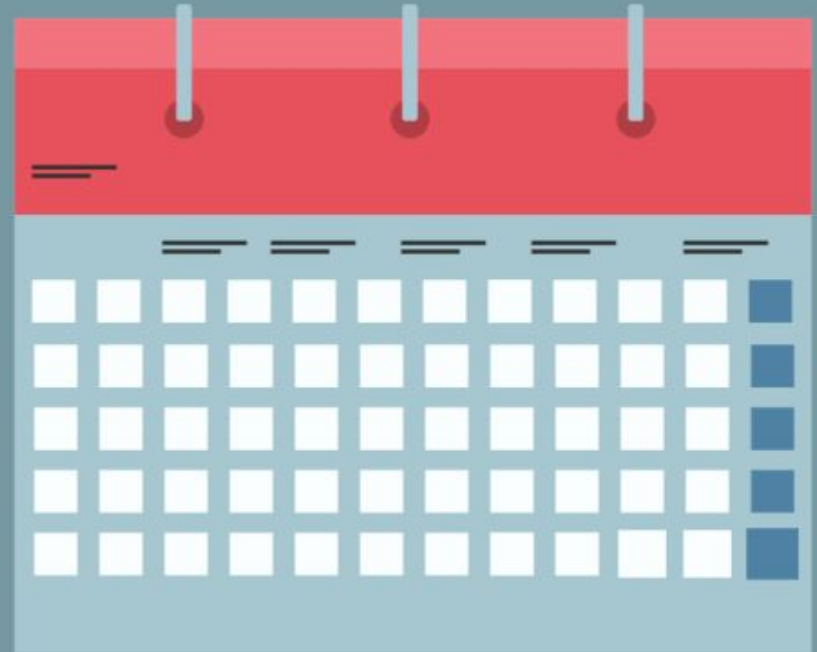
Where does the deterministic nature of time in PoS Ethereum come from?





exogenous randomness : random block time ::

on-chain pseudo-randomness : deterministic time

Time in PoW vs. Time in PoS

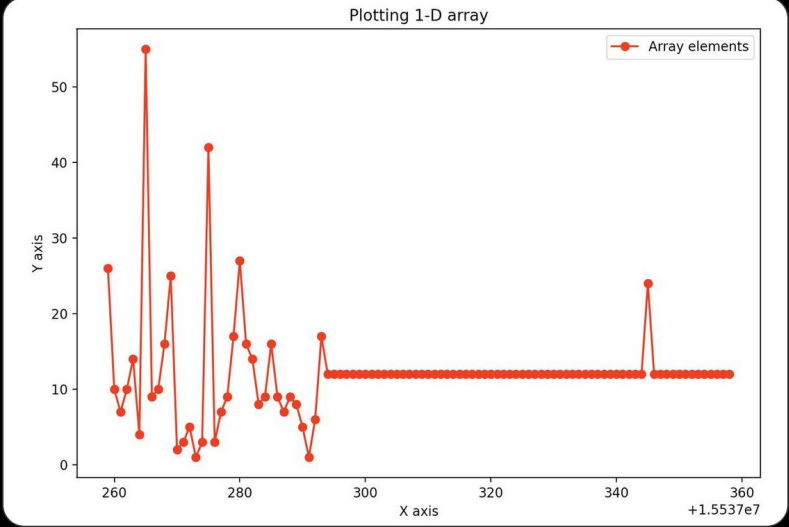


Guess when Ethereum merged ...

 **Martin Köppelmann** 
@koeppelemann

The block times of the last 100 blocks!
Amazing stability after the merge! Only 1 missed block.
This is really the best case scenario!





Plotting 1-D array



| X axis (Block Index) | Y axis (Block Time) |
|----------------------|---------------------|
| 260 | 26 |
| 261 | 10 |
| 262 | 7 |
| 263 | 14 |
| 264 | 4 |
| 265 | 55 |
| 266 | 9 |
| 267 | 25 |
| 268 | 16 |
| 269 | 2 |
| 270 | 5 |
| 271 | 1 |
| 272 | 42 |
| 273 | 3 |
| 274 | 7 |
| 275 | 17 |
| 276 | 27 |
| 277 | 16 |
| 278 | 15 |
| 279 | 9 |
| 280 | 16 |
| 281 | 9 |
| 282 | 10 |
| 283 | 9 |
| 284 | 6 |
| 285 | 1 |
| 286 | 17 |
| 287 | 6 |
| 288 | 12 |
| 289 | 12 |
| 290 | 12 |
| 291 | 12 |
| 292 | 12 |
| 293 | 12 |
| 294 | 12 |
| 295 | 12 |
| 296 | 12 |
| 297 | 12 |
| 298 | 12 |
| 299 | 12 |
| 300 | 12 |
| 301 | 12 |
| 302 | 12 |
| 303 | 12 |
| 304 | 12 |
| 305 | 12 |
| 306 | 12 |
| 307 | 12 |
| 308 | 12 |
| 309 | 12 |
| 310 | 12 |
| 311 | 12 |
| 312 | 12 |
| 313 | 12 |
| 314 | 12 |
| 315 | 12 |
| 316 | 12 |
| 317 | 12 |
| 318 | 12 |
| 319 | 12 |
| 320 | 12 |
| 321 | 12 |
| 322 | 12 |
| 323 | 12 |
| 324 | 12 |
| 325 | 12 |
| 326 | 12 |
| 327 | 12 |
| 328 | 12 |
| 329 | 12 |
| 330 | 12 |
| 331 | 12 |
| 332 | 12 |
| 333 | 12 |
| 334 | 12 |
| 335 | 12 |
| 336 | 12 |
| 337 | 12 |
| 338 | 12 |
| 339 | 12 |
| 340 | 12 |
| 341 | 12 |
| 342 | 12 |
| 343 | 12 |
| 344 | 12 |
| 345 | 25 |
| 346 | 12 |
| 347 | 12 |
| 348 | 12 |
| 349 | 12 |
| 350 | 12 |
| 351 | 12 |
| 352 | 12 |
| 353 | 12 |
| 354 | 12 |
| 355 | 12 |
| 356 | 12 |
| 357 | 12 |
| 358 | 12 |
| 359 | 12 |
| 360 | 12 |

2:00 AM · Sep 15, 2022 · Twitter Web App

459 Retweets 67 Quote Tweets 2,287 Likes



Implications of deterministic time

vitalik.eth @VitalikButerin

One important corollary of this is better EIP 1559 performance (because fewer blocks bump up against the 2x limit). So far, the percentage of full blocks has dropped from ~20% to ~10%.

etherscan.io/blocks?ps=100&...

Martin Köppelmann @koeppelmann · Sep 15

The block times of the last 100 blocks!
Amazing stability after the merge! Only 1 missed block.
This is really the best case scenario!

Plotting 1-D array

2:08 AM · Sep 15, 2022 · Twitter Web App

1,595 Retweets 73 Quote Tweets 6,775 Likes

barnabe.eth @barnabemonnot

Replying to @VitalikButerin

Can you tell when the Merge happened? :) dune.com/barnabe/EIP1559

gas used around Merge

5:50 AM · Sep 15, 2022 · Twitter Web App

1 Retweet 22 Likes

better load balancing!

A historical painting of a man in 17th-century attire, possibly a monarch or high official, with a large black text overlay. The man has long, dark, curly hair and is wearing a white ruffled shirt and a blue and gold patterned coat. He is holding a golden staff or scepter. The background features red and gold draped fabrics. The text is in a bold, red, serif font.

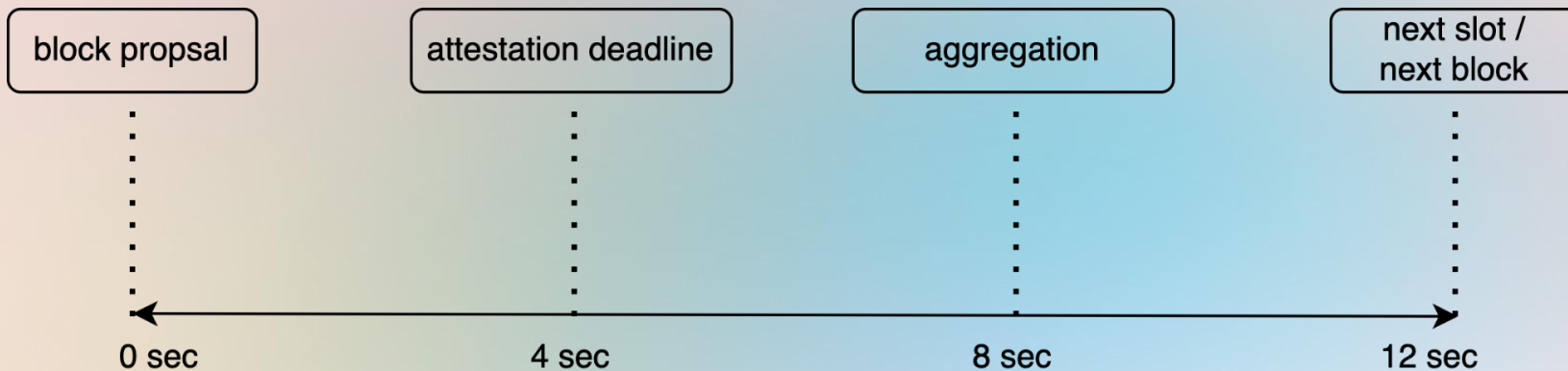
**But proposers can abuse their
guaranteed monopoly power**

Progression of a slot

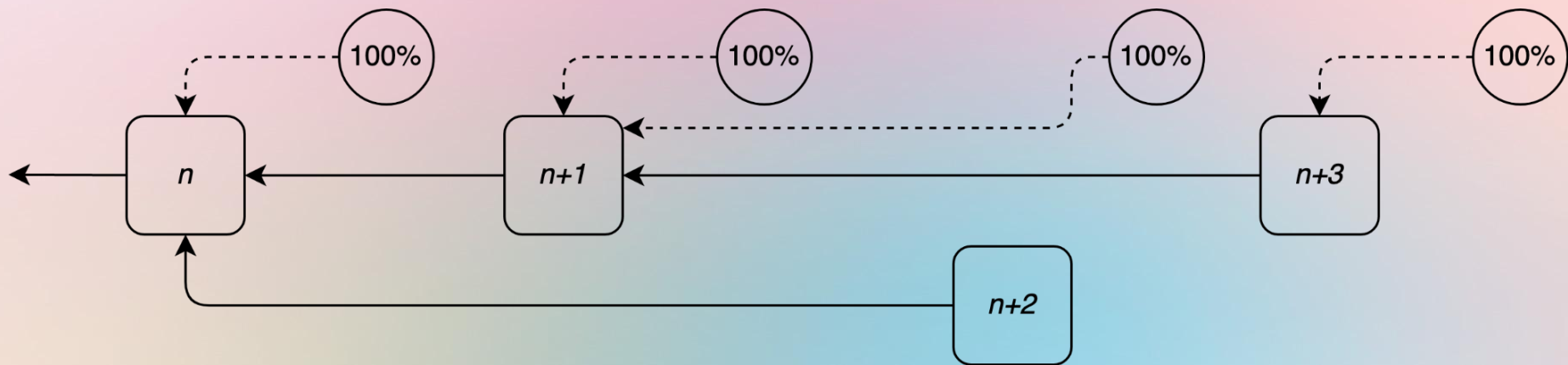
Attesting

A validator is expected to create, sign, and broadcast an attestation during each epoch. The `committee`, assigned `index`, and assigned `slot` for which the validator performs this role during an epoch are defined by `get_committee_assignment(state, epoch, validator_index)`.

A validator should create and broadcast the `attestation` to the associated attestation subnet when either (a) the validator has received a valid block from the expected block proposer for the assigned `slot` or (b) $1 / \text{INTERVALS_PER_SLOT}$ of the `slot` has transpired ($\text{SECONDS_PER_SLOT} / \text{INTERVALS_PER_SLOT}$ seconds after the start of `slot`) -- whichever comes *first*.



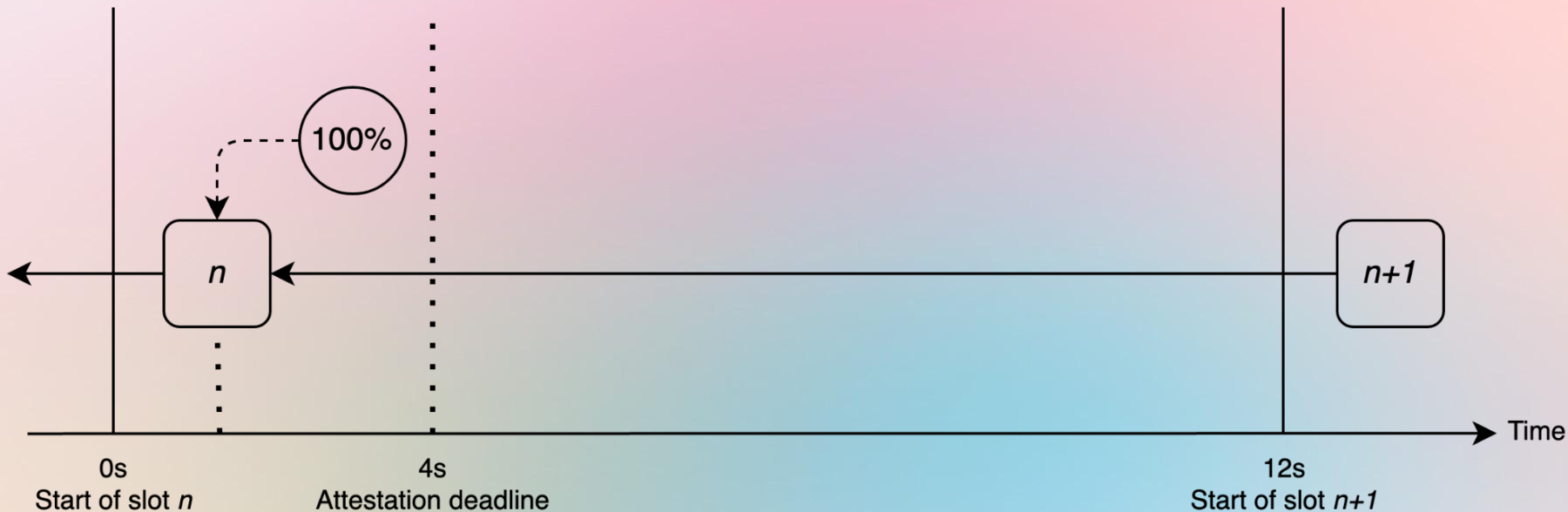
Fork choice rule: LMD GHOST-*ish*



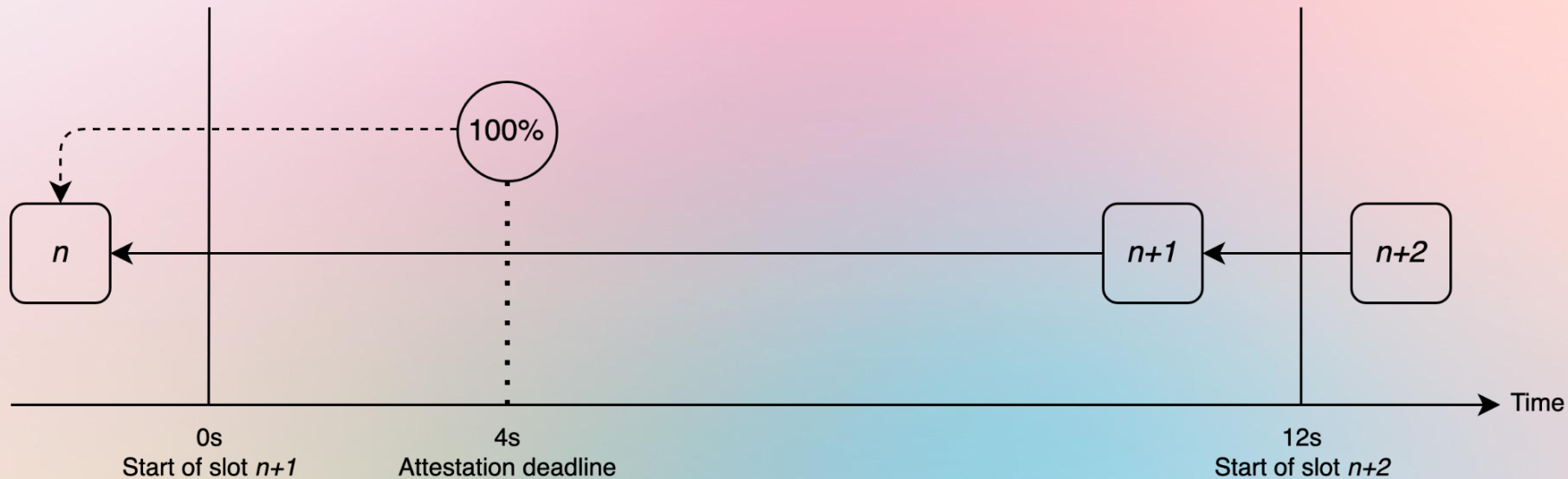
A historical painting of a man in 17th-century attire, possibly a monarch or high official, with a large black text overlay. The man has long, dark, curly hair and is wearing a white ruffled shirt and a blue and gold patterned coat. He is holding a golden staff or scepter. The background features red and gold drapery. The text is in a bold, red, serif font.

**But proposers can abuse their
guaranteed monopoly power**

Block receipt time: 0s-4s into slot (“on time”-ish)

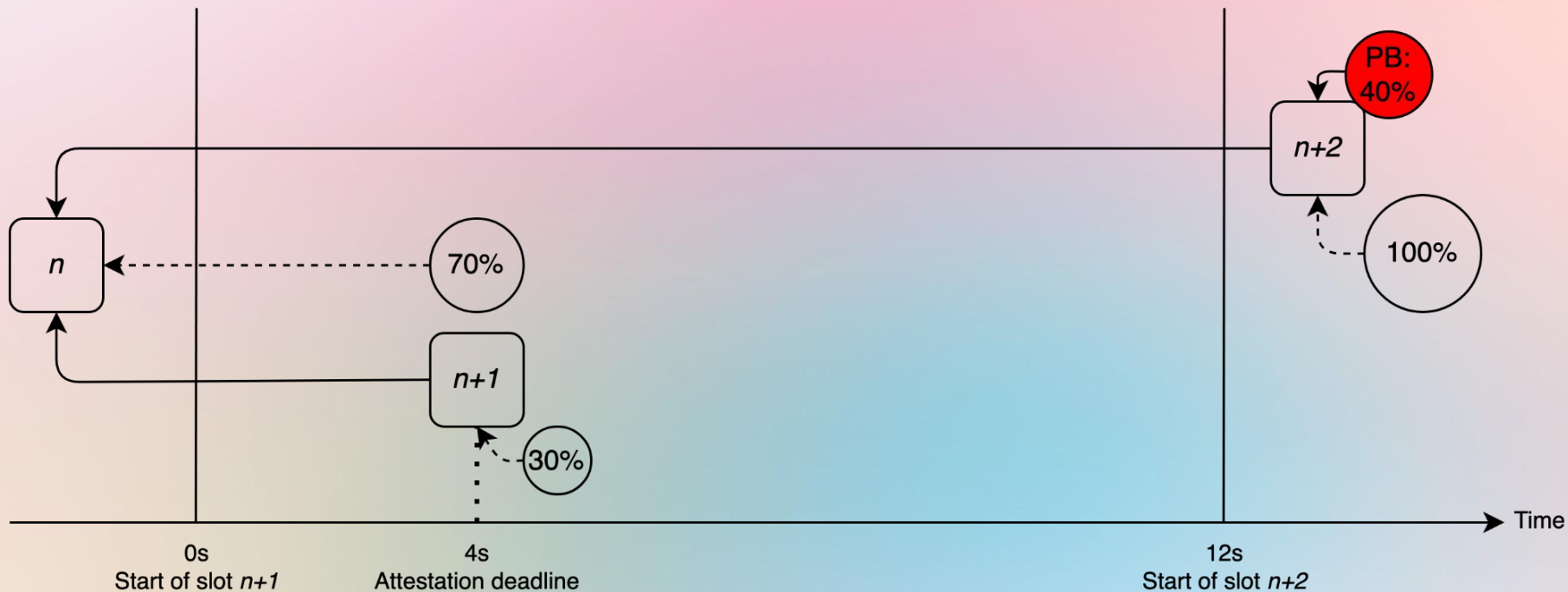


Block receipt time: $>4s$ into slot (“late”)

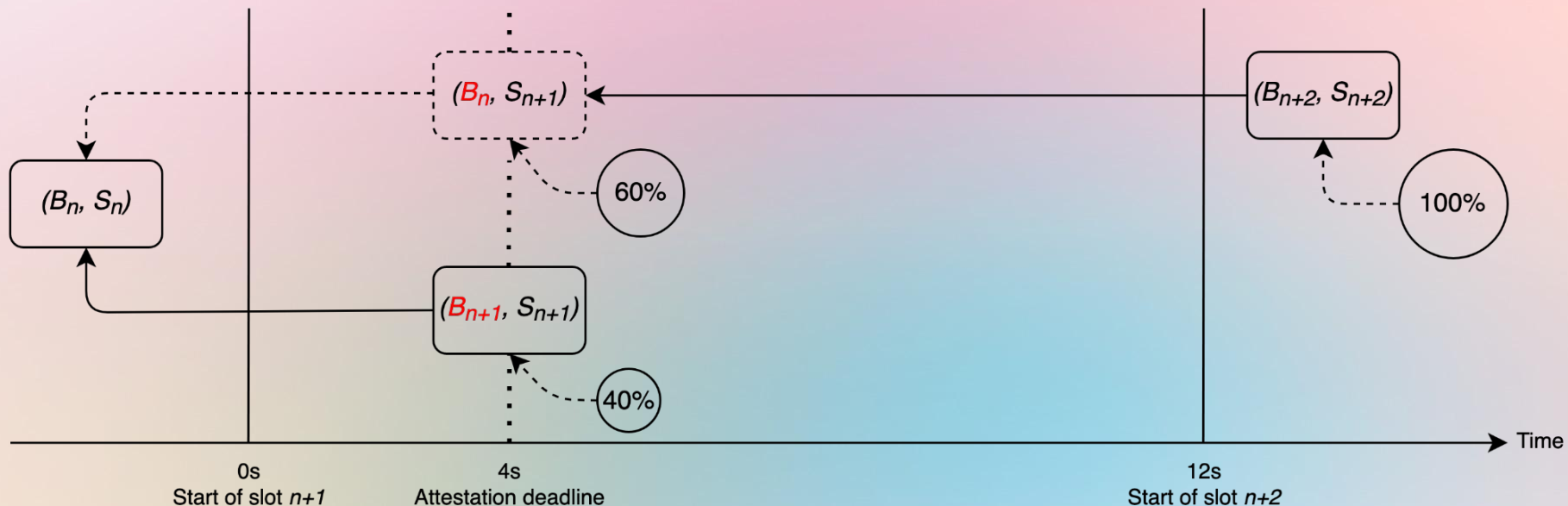


Proposer monopoly... wat do?

Fork choice fun TODAY: proposer boost



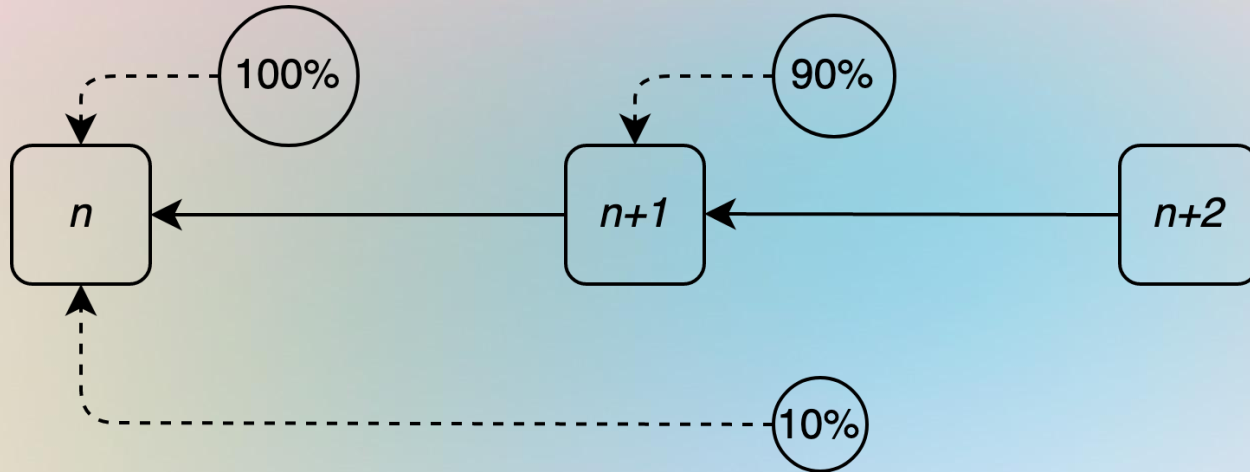
Fork choice fun TOMORROW: (block, slot)-voting



Idea: Incentivizing **timeliness** explicitly

Today: Block proposers are rewarded in proportion to the profitability of attestations they include in their block.

Idea: Scale the proposer's reward by the **share of same-slot committee votes that the block receives and are included in the subsequent block.**





load stability, good.
guaranteed monopoly, bad.

Thank you!

Strong research background?
Mechanism design expert?
Want to help us make sense of it?

Apply to the RIG now!



Caspar Schwarz-Schilling

Robust Incentives Group (RIG), Ethereum Foundation

caspar@ethereum.org



@casparschwa