TECHNICAL DESIGN PROJECT || 2022

DIY mempool

PRESENTATION BY SIERRA CHASTAIN-RITTUE

today's agenda

OUTLINE OF TOPICS

Prompt

Assumptions

Considerations

Risk Assessment

Proposed Design

Quality Assurance

Defining Success

the prompt

DESIGN A MEMPOOL FOR A CONSENSUS ENGINE LIKE TENDERMINT

assumptions

CONSENSUS ENGINE

Modular architecture, BFT-based Proof of Stake, P2P gossip protocol, Separate app interface

CHAOTIC DATA

Mempool is pre-consensus

TRANSACTIONS

Exchange of information, money

assumptions

END USERS

Software engineers across industries

FLEXIBILITY

Transactions can be overwritten & replaced

CONGESTION

Frozen, stuck, & slow transactions impact UX

considerations

INTEROPERABILITY

Flexible design for widespread usage

TRANSACTION TIMES

Utilize average time as benchmark for performance speed

TRANSACTION PRIORITY

Optional, but useful for larger loads and eviction decisions

risk assessment

MISSED TRANSACTIONS

Written transactions not accepted by a mempool

VALIDATORS

Deterministic; round-robin system

FORKING

Inherent risk of PoS design. In theory validators could claim twice the rewards: the 'nothing at stake' problem

proposed mempool design

for Tenderbeast

(Featuring Salvatore, the namesake)

design

PRIORITY

An optional field in the tx;
useful for scalability and widespread usage

SENDER

Required field in the tx;
specifies who sent the transaction

SIZE

5000 transactions, 1GB mempool
20MB blocks

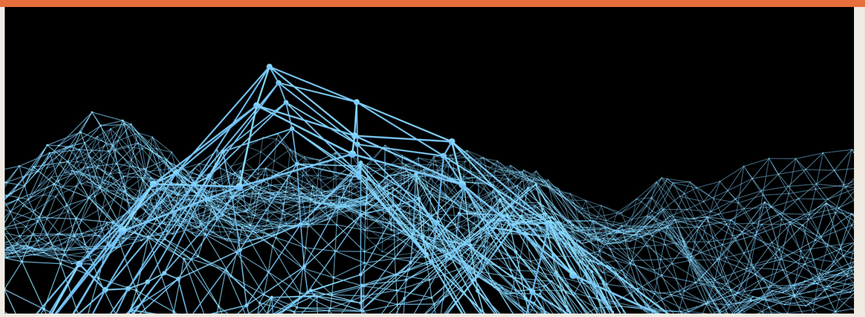
design

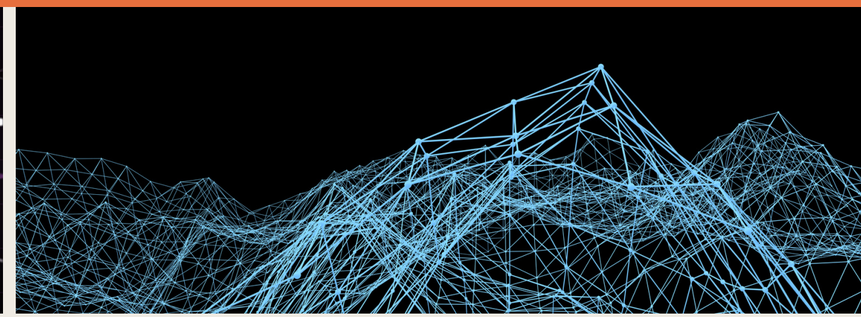
TRANSACTION ALERTS

Configurable. Can notify users when a transaction fails/expires or is sent/received, improving UX

PERFORMANCE DASHBOARD

Presents baseline numbers as well as the current state of the mempool





UNIT TESTING

API TESTING

UI TESTING

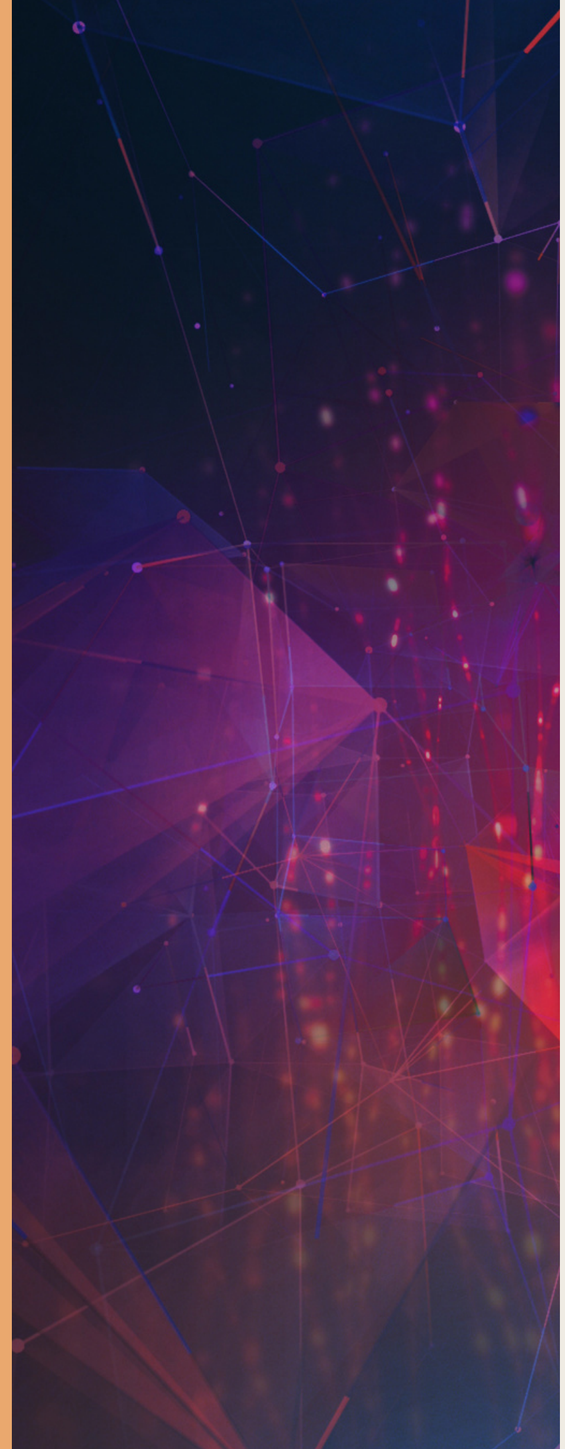
quality assurance

defining success

how do we know it works?

- USER ACCEPTANCE TESTING
- SECURITY TESTING
- INCREASE IN TOTAL NODES

SIERRA CHASTAIN-RITTUE
TECHNICAL DESIGN PROJECT || 2022



Thank You!

QUESTIONS?