Picolo Research

Independent research on blockchain startups and ICO's

An independent review of Elrond (ERD)

Updated with Technical Code Review

17 July 2019 - Lennard Neo

Powered By:

ASTRONAUT | CAPITAL

ELROND (ERD)

A Scalable Protocol Technology Utilizing Adaptive State Sharding with SPoS for Practicality

Buy

Summary

Elrond is creating a novel architecture through adaptive state sharding technology and Secure Proof of Stake (SPoS) consensus. This transfer protocol will enable a scalable ecosystem embedded with interconnectivity while maintaining the decentralisation, security and fairness features of a public blockchain. Preliminary test results have reflected an average throughput of 1000x increase as compared to existing solutions that are currently in the market.

Concept | **MVP** | **Established**

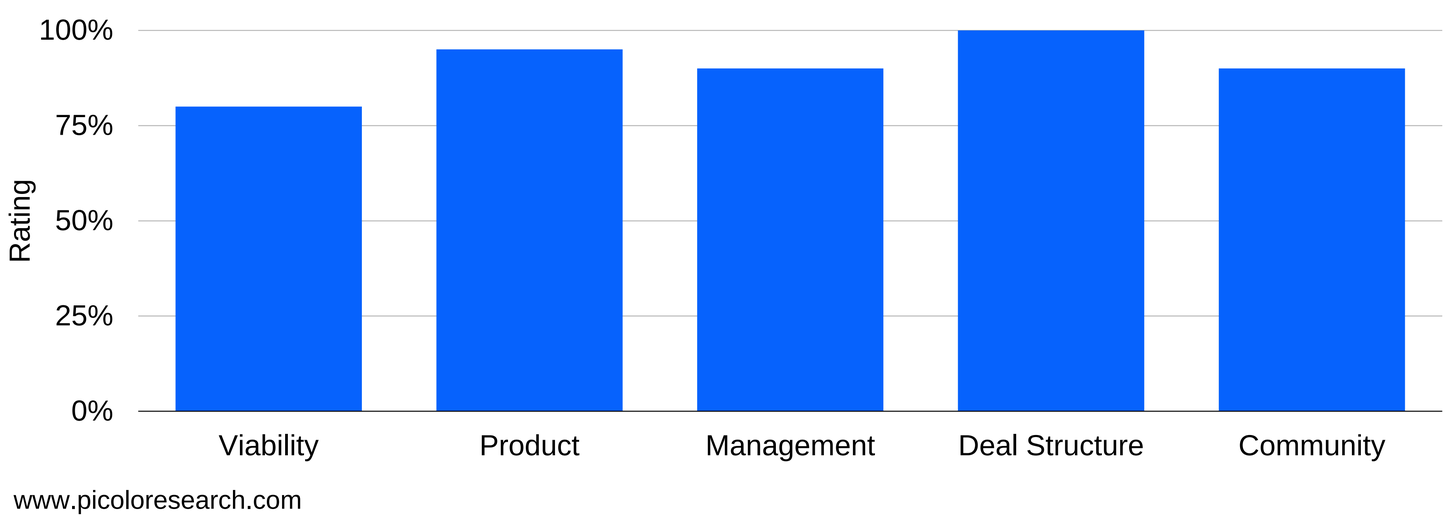
Company Overview

Incorporated in 2017, with its core team in Romania, Elrond seeks to provide a novel architecture that augments current models to improve scalability performance, security and decentralisation. Adopting a two-prong approach through Adaptive State Sharding and Secure Proof of Stake (SPoS) consensus algorithm, Elrond seeks to provide 10k transactions per seconds (TPS), minimize network latency, and reduce fees for the digital economy.

Here are several issues that the project seeks to resolve:

- Full decentralisation to create a trustless ecosystem and remove any single point of failure
- High security to prevent attacks from different vectors
- Scalability to achieve TPS that is higher or at least equivalent to centralised counterparts
- Energy efficient ecosystem that reduces computational requirements
- Bootstrapping and storage enhancement to reduce the time for network synchronization and data storage
- Interoperability to enable communications across networks

Birds-Eye View



www.picoloresearch.com

General

Ticker:	ERD
Website:	Click here
Sale Period:	1 Jul 2019
White Paper:	Click here

Token info

Price: (16 Jul 2019)	$0.003687
Returns since IEO	5.7x
ATH price:	$0.007511
Hard Cap:	$3.25m
Market Cap: (16 Jul 2019)	$27.7m (Based on Initial Cir. Supply)
Initial Circulating supply %:	37.5%

Checklist

Management:	✔
Product:	✔
Commercial:	✔
Interest:	✔
Fulfilment:	⚠

Commercial & Technical Strategy

Elrond is reinventing the public blockchain infrastructure to be more secure, efficient, scalable and interoperable. The project seeks to eliminate energy and computational waste from PoW algorithms through the use of SPoS consensus and sharding technology, which are the two cornerstones of the platform.

Several core features of the platform include:

- High overall throughput of 65k TPS in recent testnet performance
- Adaptive State Sharding that splits the blockchain into multiple shards using a binary tree, which reduces latency
- Implementing a technique to balance nodes and rewards to achieve overall network equilibrium
- Automatic transaction routing within corresponding shards
- Shard pruning mechanism to reduce bootstrapping and storage costs, increasing overall throughput
- Secure Proof of Stake (SPoS) consensus mechanism
- A block proposer, which is part of a validator, randomly select consensus group in under 0.1s, enhanced by a robust deterministic function to increase security
- A byzantine adversarial model to prevent attacks
- Interoperability with EVM (Ethereum Virtual Machine) compliant

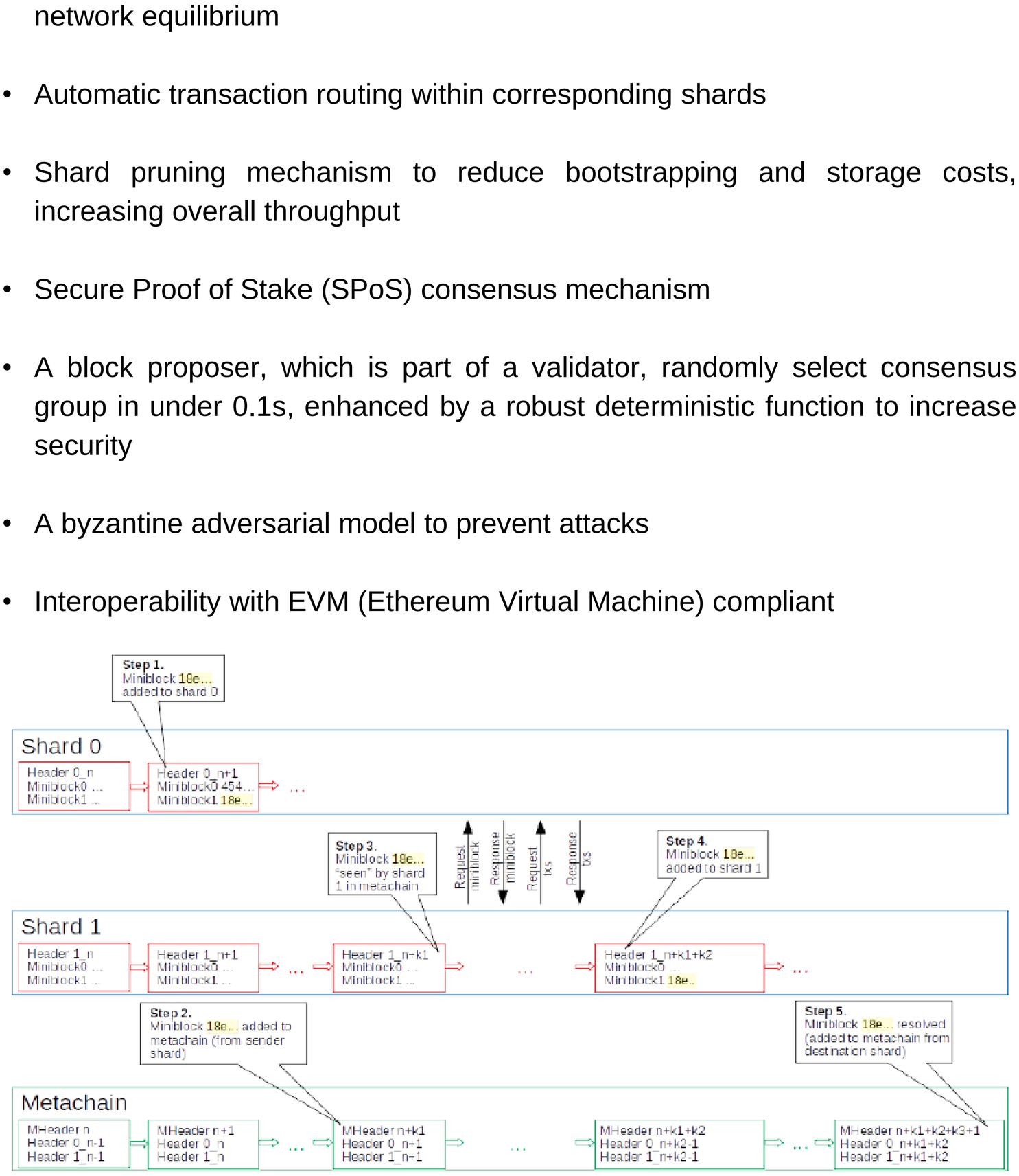


Figure: Cross Shard Transaction Processing

Figure: Network Throughput with Increasing Number of Shards

Technical Code Review

The primary goal of Elrond is to solve several major issues faced by existing blockchain technologies. This technical review will cover 3 major components in detailed, namely:

1. Security through Secure Proof of Stake (SPoS)
2. Scalability via Adaptive State Sharding
3. Interoperability through VM based on K-framework

Security through Secure Proof of Stake (SPoS)

Elrond proposes a unique consensus algorithm called “Secure Proof of Stake” which aims to balance network efficiency with network security. This implementation is an expansion and improvement of Algorand’s concept of random selection mechanism.

1. A random sample from a smaller consensus group will be selected out of all eligible validators in a shard. The randomness is unpredictable before the signing of the previous block and is deterministic once the previous block signature is known. The randomness takes into consideration the stake and rating of a node when selecting validators for the consensus group. The default rating of a node is 0.
2. The hash of the public key and randomization factor of every validator in the consensus group is derived, and the first node will be elected as the block proposer. The block proposer will aggregate transactions to build a new block before broadcasting the block to all validators in the consensus group for verification.
3. Each validator will verify the validity of the block and process its transactions. If there are no problems, it will participate in the pBFT consensus. Every validator will then send a signature to the proposer for a modified BLS multi-signature scheme. If more than (2/3 + 1) signatures are collected, the block is considered validated and will be disseminated to the entire shard.
4. The aggregated signature is used to prove that consensus was reached. Selection of next consensus group will be done through BLS signature over the last randomness source. Both the new and old randomness sources are then added to current block.
5. The rating of the block proposer will increase at the end of the epoch, by analysing the statistical data on the behaviour of each validator.

Elrond’s approach to randomness selection with consideration based on stake and rating greatly reduces the possibility of collusion and deters bad actors with slashing.

However, this was not Elrond’s original proposal. The team managed to identify key issues with their previous proposal and made modifications to the design of the consensus and randomness source to achieve greater efficiency.

	Previous Proposal	New Proposal
Multi-signature scheme	Belare Neven multi-signature (Schnorr multi-signature scheme)	BLS multi-signature scheme
Communication rounds	5	2

Even though the BLS multi-signature scheme is more time-consuming on both signing and verification, the reduction in communication rounds reduces the overall time spent in consensus.

We review several of Elrond's Github repositories.

consensusState.go

```
95  // GetNextConsensusGroup gets the new consensus group for the current round based on current eligible list and a random
96  // source for the new selection
97  func (cns *ConsensusState) GetNextConsensusGroup(randomSource string, vgs consensus.ValidatorGroupSelector) ([]string,
98      error) {
99      validatorsGroup, err := vgs.ComputeValidatorsGroup([]byte(randomSource))
100
101     if err != nil {
102         return nil, err
103     }
104
105     newConsensusGroup := make([]string, 0)
106
107     for i := 0; i < len(validatorsGroup); i++ {
108         newConsensusGroup = append(newConsensusGroup, string(validatorsGroup[i].PubKey()))
109     }
110
111     return newConsensusGroup, nil
112 }
```

Deriving of next consensus group from random source

In the above code snippet, computing of the next validator group takes in a byte array parameter generated by a random source.

consensusState.go

```
82  // GetLeader method gets the leader of the current round
83  func (cns *ConsensusState) GetLeader() (string, error) {
84      if cns.consensusGroup == nil {
85          return "", ErrNilConsensusGroup
86      }
87
88      if len(cns.consensusGroup) == 0 {
89          return "", ErrEmptyConsensusGroup
90      }
91
92      return cns.consensusGroup[0], nil
93  }
```

First node of the consensus group is elected as the block proposer

The way this function is structured indicates that the leader (or block proposer) is the first node in the consensus group.

blsMultisig.go

```
146 // CreateSignatureShare returns a BLS single signature over the message
147 func (bms *blsMultiSigner) CreateSignatureShare(message []byte, _ []byte) ([]byte, error) {
148     bms.mutSigData.Lock()
149     defer bms.mutSigData.Unlock()
150
151     data := bms.data
152     sigShareBytes, err := bms.llSigner.SignShare(data.privKey, message)
153     if err != nil {
154         return nil, err
155     }
156
157     data.sigShares[data.ownIndex] = sigShareBytes
158
159     return sigShareBytes, nil
160 }
```

Generating signature using BLS multi-signature scheme

Line 152 shows the signing of the share using a private key. Line 157 shows the adding of the signed share into the array of signature shares.

subroundEndRound.go

```
57      bitmap := sr.GenerateBitmap(SrSignature)
58      err := sr.checkSignaturesValidity(bitmap)
59      if err != nil {
60          log.Error(err.Error())
61          return false
62      }
63
64      // Aggregate sig and add it to the block
65      sig, err := sr.MultiSigner().AggregateSigs(bitmap)
66      if err != nil {
67          log.Error(err.Error())
68          return false
69      }
70
71      sr.Header.SetPubKeysBitmap(bitmap)
72      sr.Header.SetSignature(sig)
73
74      timeBefore := time.Now()
75      // Commit the block (commits also the account state)
76      err = sr.BlockProcessor().CommitBlock(sr.Blockchain(), sr.Header, sr.BlockBody)
77      if err != nil {
78          log.Error(err.Error())
79          return false
80      }
81      timeAfter := time.Now()
82
83      log.Info(fmt.Sprintf("time elapsed to commit block: %v sec\n", timeAfter.Sub(timeBefore).Seconds()))
84
85      sr.SetStatus(SrEndRound, spos.SsFinished)
86
87      // broadcast block body and header
88      err = sr.BroadcastMessenger().BroadcastBlock(sr.BlockBody, sr.Header)
89      if err != nil {
90          log.Error(err.Error())
91      }
92
```

Aggregating of signatures and broadcasting the block

Line 57 shows the generating of the bitmap using the signatures obtained, followed by line 58, which checks the validity of the bitmap. Line 65 shows the aggregating of the signatures using the bitmap. Line 71 and 72 show both the bitmap and the signature being added to the next block's header. Finally, line 88 shows the broadcasting of the block.

Overall, the team has completed the codes for this component of Elrond's platform. The implementation of SPoS is elegant yet provides an effortless way to achieve finality and security. Furthermore, the team's flexibility to make modifications to their previous approach shows their ability to identify key issues and develop improvements to increase the platform's efficiency.

Scalability via Adaptive State Sharding

Elrond's approach to scalability is through their proposal of Adaptive State Sharding. This mechanism will adapt and reorganize the shards based on the number of active nodes in the network.

There are usually a few considerations regarding sharding - latency, cross-sharding capabilities and storage.

Elrond has conceptualized solutions to each of these issues:

1. Latency – Using a binary tree to divide the account address space in shards to reduce latency. This is achieved due to the lack of split overhead as it is predetermined by the hierarchy when using a binary tree.
2. Cross-sharding capabilities – By using miniblocks within shards and a metachain as state sync. Miniblocks containing the transactions in the current shard are sent to the metachain which notarizes the block by creating a new metachain block (metablock). Other shards will then be able to fetch the hash of the miniblock from the metablock, request the miniblock from the original shard, and execute any cross-shard transactions. The resulting block will be sent to the metachain. After notarization, the cross-shard transaction can be considered finalized.

3. Storage – A shard pruning mechanism is implemented to ensure the sustainability of the network. At the end of each epoch, the block proposer will create a state block containing the hash of the Merkle tree's root and balances. When consensus is reached, the block proposer will store the state block in the shard's ledger, making it the genesis block for the next epoch. At the end of the next epoch, the nodes will drop the previous state block and all the blocks before it.

multiShardCoordinator.go

```
47  // ComputeId calculates the shard for a given address used for transaction dispatching
48  func (msc *multiShardCoordinator) ComputeId(address state.AddressContainer) uint32 {
49      bytesNeed := int(msc.numberOfShards/256) + 1
50      startingIndex := 0
51      if len(address.Bytes()) > bytesNeed {
52          startingIndex = len(address.Bytes()) - bytesNeed
53      }
54
55      buffNeeded := address.Bytes()[startingIndex:]
56
57      addr := uint32(0)
58      for i := 0; i < len(buffNeeded); i++ {
59          addr = addr<<8 + uint32(buffNeeded[i])
60      }
61
62      shard := addr & msc.maskHigh
63      if shard > msc.numberOfShards-1 {
64          shard = addr & msc.maskLow
65      }
66
67      return shard
68  }
```

Binary tree splitting of account addresses

This code snippet returns the shard of which the given address belongs to. This is computed using a binary tree. Line 49 computes the number of bytes needed to find out which shard the address belongs to. Line 50 and 52 initializes the starting index of the byte to begin the binary tree split. Line 58 to 60 iterates through the binary tree. Line 62 to 65 returns the shard that the address belongs to.

shardblock.go

```
1190    orderedMetaBlocks, err := sp.getOrderedMetaBlocks(round)
1191    if err != nil {
1192        return nil, 0, err
1193    }
1194
1195    log.Info(fmt.Sprintf("meta blocks ordered: %d \n", len(orderedMetaBlocks)))
1196
1197    lastMetaHdr, err := sp.getLastNotarizedHdr(sharding.MetachainShardId)
1198    if err != nil {
1199        return nil, 0, err
1200    }
1201
1202    // do processing in order
1203    usedMetaHdrsHashes := make([][]byte, 0)
1204    for i := 0; i < len(orderedMetaBlocks); i++ {
1205        if !haveTime() {
1206            log.Info(fmt.Sprintf("time is up after putting %d cross txs with destination to current shard \n", nrTxAdded))
1207            break
1208        }
1209
1210        hdr, ok := orderedMetaBlocks[i].hdr.(*block.MetaBlock)
1211        if !ok {
1212            continue
1213        }
1214
1215        err := sp.isHdrConstructionValid(hdr, lastMetaHdr)
1216        if err != nil {
1217            continue
1218        }
1219
1220        isFinal := sp.isMetaHeaderFinal(hdr, orderedMetaBlocks, i+1)
1221        if !isFinal {
1222            continue
1223        }
1224
1225        if len(hdr.GetMiniBlockHeadersWithDst(sp.shardCoordinator.SelfId())) == 0 {
1226            usedMetaHdrsHashes = append(usedMetaHdrsHashes, orderedMetaBlocks[i].hash)
1227            lastMetaHdr = hdr
1228            continue
1229        }
1230
1231        maxTxRemaining := uint32(maxTxInBlock) - nrTxAdded
1232        currMBProcessed, currTxsAdded, hdrProcessFinished := sp.createAndprocessMiniBlocksFromHeader(hdr, maxTxRemaining, round, haveTime)
1233
1234        // all txs processed, add to processed miniblocks
1235        miniBlocks = append(miniBlocks, currMBProcessed...)
1236        nrTxAdded = nrTxAdded + currTxsAdded
1237
1238        if currTxsAdded > 0 {
1239            usedMetaHdrsHashes = append(usedMetaHdrsHashes, orderedMetaBlocks[i].hash)
1240        }
1241
1242        if !hdrProcessFinished {
1243            break
1244        }
1245
1246        lastMetaHdr = hdr
1247    }
```

Retrieval of miniblocks from metachain, processing of miniblocks

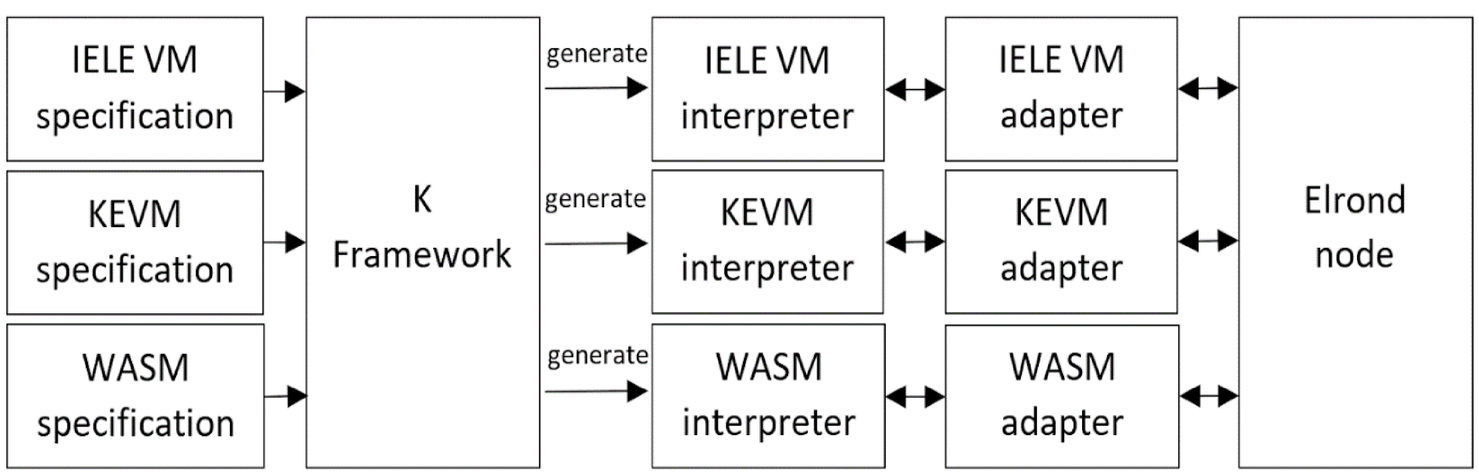
Line 1190 gets the list of metablocks to retrieve. Line 1210 retrieves the headers from the metablock. Line 1232 retrieves the miniblocks using the headers and processes them.

Overall, Elrond's approach is different from the conventional sharding methodology, where the number of shards is static. However, the team has yet to implement the full Adaptive State Sharding mechanism on their platform as the testnet utilizes static state sharding where the number of shards is fixed.

Interoperability through VM based on K-framework

Elrond's approach towards interoperability is to implement an adapter mechanism at the VM level (as proposed by Cosmos). This would require specialized adapters for each chain that will interoperate with Elrond.

Elrond's proposed VM infrastructure is built on top of K Framework. The advantage of using K Framework is that programming languages can be clearly defined, resulting in reliable and safe execution and behaviour. Using a common VM interface, each VM can be plugged into the Elrond network by having an adapter that implements the interface.



testInitializer.go

```
91  func CreateTxProcessorWithOneSCExecutorIeleVM(accnts state.AccountsAdapter) process.TransactionProcessor {
92      blockChainHook, _ := hooks.NewVMAccountsDB(accnts, addrConv)
93      //TODO uncomment the following 2 lines
94      //cryptoHook := &hooks.VMCryptoHook{}
95      //vm := endpoint.NewElrondIeleVM(blockChainHook, cryptoHook, ielecommon.Default)
96      //Uncomment this to enable trace printing of the vm
97      //vm.SetTracePretty()
98      argsParser, _ := smartContract.NewAtArgumentParser()
99      scProcessor, _ := smartContract.NewSmartContractProcessor(
100         nil,
101         argsParser,
102         testHasher,
103         testMarshalizer,
104         accnts,
105         blockChainHook,
106         addrConv,
107         oneShardCoordinator,
108     )
109     txProcessor, _ := transaction.NewTxProcessor(accnts, testHasher, addrConv, testMarshalizer, oneShardCoordinator, scProcessor)
110
111     return txProcessor
112 }
```

Integration tests for IELE VM

Line 95 is the initialization of an IELE VM.

The Elrond team has yet to implement its full interoperability solution, which is slated to launch by Q3 2019. In a recent technical update, the project has implemented VM and protocol integrations with several tests conducted. The team has also open-sourced the codes. However, communications from the team were that work is still in progress, and developments are ongoing internally.

Conclusion - Technical Code Review

Components	In Paper	Proof in History	Code Realised	Code Complete-ness	% Success	Comments
SPoS	Y	N	Y	100%	High	SPoS implements a random selection mechanism based on stake and ratings combined with a pBFT consensus. Elrond's implementation is an elegant yet simple solution to finality and security.
Adaptive State Sharding	Y	N	N	50%	Medium	Adaptive state sharding solves potential problems with sharding due to availability of nodes. This implementation is novel; however, security challenges remain when there are insufficient validators during heavy loads.
VM based on K-framework	Y	Y (Cardano)	N	30%	High	K-framework is a VM infrastructure that supports easy modelling of various VM languages through modifications. K-framework has been around for more than a year, and there have been various implementations of it, including KEVM (A complete formal K semantics of the Ethereum Virtual Machine).

Taking into account all 3 major components, as evident from the technical review, the Elrond team has displayed a clear understanding of their objectives and have conceptualized novel approaches to solving the major issues faced by blockchain technology. The source codes are well documented and adhere to a Test-Driven Development approach.

However, the project is still in the early stages with several repositories due to be completed in the coming months. Solutions to scalability and the implementation of sharding and cross-sharding capabilities, along with the added goal of interoperability between blockchains, are complex in nature and the concepts put forth are still theoretical. At present, the Elrond team has taken considerable steps towards a solution that has a high possibility of success, reaching the required throughput and be the go-to platform for the digital economy.

Roadmap

Elrond has a simple and straight forward roadmap, which has been delivered timely, adding to the team's credibility.

Roadmap	Developments	Status
Q4 2017	Elrond Inception	*Completed*
Q2 2018	Technical Whitepaper (release 1)	*Completed*
	Initial formalisation of Adaptive State Sharding and SPoS	*Completed*
Q3 2018	Elrond Prototype Public Release	*Completed*
	Achieve 1k TPS with 250 nodes and 10 shards	*Completed*
Q4 2018	Open Source	*Completed*
	Whitepaper update	*Completed*
Q1 2019	Elrond Testnet v0.5 results	*Completed*
	Validating new implementation in Golang	*Completed*
	Achieve performance increase to 3750 TPS on a single shard	*Completed*
Q2 2019	Testnet Launch	*Completed*
	Elrond Block explorer & Wallet	*Completed*
Q3 2019	Public testnet launch Launch VM integration Develop shard adaptivity and Pruning Elrond Game dApp launch Elrond token economics Target 5 Business Partnerships Execution of Global & PR Marketing plans	
Q4 2019	**Elrond Mainnet launch, initiate token swap and staking** Implement functional payment gateway Launch Elrond dApp store and Elrond name service Target 10 Business Partnerships Supplement local communities PR & Marketing efforts	
Q1 2020	Introduce Elrond digital identity Launch Elrond DEX	

Team

The project comprises a team of 19 employees, many of whom have prior experiences in the blockchain sector.

Beniamin Mincu (CEO) – 8+ years of experience, of which 4 years are within blockchain. Prior experience includes Metachain Capital, NEM, and ICO Market Data. He graduated with a bachelor's in economics from the German University of Sibiu.

Lucian Todea (COO) – 16+ years of experience as a serial entrepreneur within the technology sector. He founded Soft32, ITNT, Travelgator, and mobilPay.com. He majors in Finance from The Bucharest University of Economic Studies.

Lucian Mincu (CIO) – 8+ years of experience in IT. Prior experiences include Computer Troubleshooters, Uhrenwerk24 UG, Cetto Services GmbH, LIEBL Systems GmBH, Metachain Capital, and ICO Market Data. He is an IT Specialist in Computer Science and graduated from Industrie- und Handelskammer.

Felix Crisan (Head of Research) – 20+ years of experience in IT, and specialises in BigData, Machine Learning and AI. Prior experiences include IBM, Hewlett-Packard, Phenomedia, Cybersoft, and founded companies such as mobilPay.com, NETOPIA system, and BTKO. He holds a bachelor's in computer science from the University of Bucharest.

Radu Chris (Head of Technology) – 11+ years of experience as a developer and an expert in Advanced computer architecture. He worked both as a developer and research assistance in the Lucian Blaga University of Sibiu and was a partner in NTT Data Romania. He holds a PhD in computer science from the Lucian Blaga University of Sibiu.

Adrian Dobrita (Head of Engineering) – 10+ years of experience in software engineering. Prior experiences include Intel, Continental Automotive Systems, AUSY, and Pentalog. He holds a master's in advanced computing system from the Lucian Blaga University of Sibiu.

Addition team members include:

- 3 Core Developers
- 6 Software Engineers
- 2 UX Designers
- 2 Business Development & Marketing

Advisors

Alex Iskold (Business Advisor) – Managing Partner @ 2048 Ventures, a seasoned investor with presence in 90+ startups

Alex Tabarrok (Economics Advisor) – Professor @ George Mason University, specialises in economics

Raul Jordan (Technical Advisor) – Co-Lead @ Prysmatic Labs, Ethereum protocol developer, Partner @ zk Capital

Grigore Rosu (Technical Advisor) – Professor @ University of Illinois, specialises in formal methods and programming languages

Fabio C. Canesin (Technical Advisor) – Co-Founder @ Nash, City of Zion

Ethan Fast (Technical Advisor) – CTO @ Nash, Co-Founder @ City of Zion, PhD Computer Science from Stanford University

Andrei Pitis (Business Advisor) – VP Product and Head of European Development Centers @ Fitbit

Investors

Several investors participated in Elrond's previous round. Notable ones include:

Binance Labs – Binance Labs is the venture arm of Binance with a vision to incubate, invest, and empower blockchain projects

Neo Global Capital – NGC invests in inspiring projects related to blockchain. Past investments include Bluzelle, Zilliqa, Trinity, Fortuna, IHT, Ontology, Dekrypt Capital, mainframe, Switcheo Network, Solana, Ankr, Blockcloud, NKN, Oasis Labs, nOS, Certik

Maven11 Capital – A European investment firm solely focused on DLT/blockchain technology and digital assets. Past investments include DUSK, Nash, Ethereum, ICON, Basic Attention Token, SONM, Ontology, 0X, Carry Protocol, Omisego

Electric Capital – The fund invests in crypto companies and protocols through deep due diligence in the technology by compiling codes, security audits and running nodes. Past investments include Coda, Coinlist, Dfinity, Near, Spacemesh, Oasislabs, Thunder Token

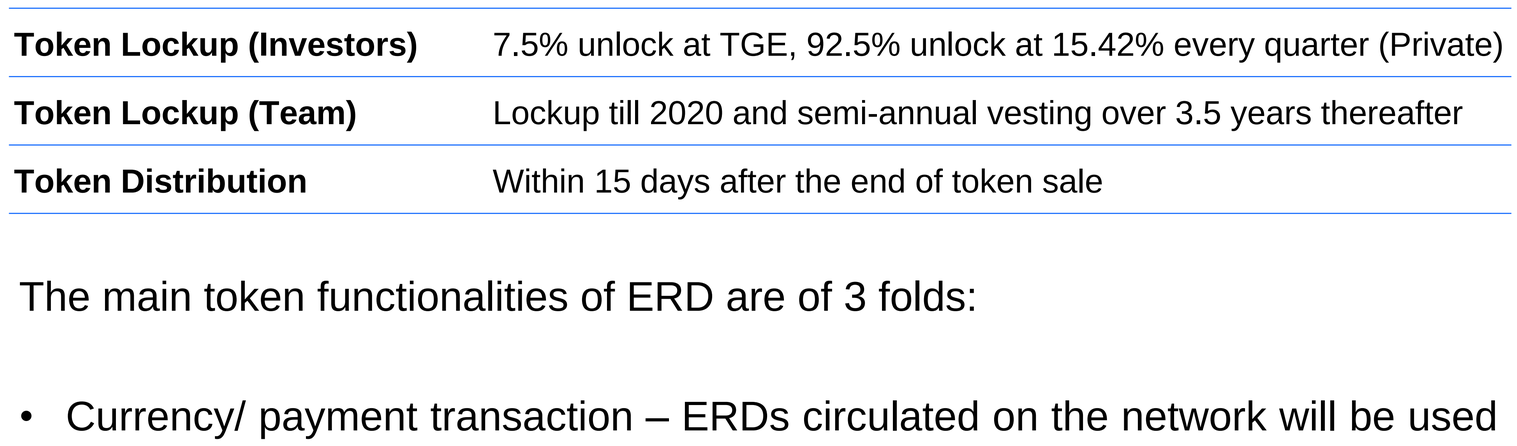
Token Sale

Elrond raised a total of $1.9m in a private sale round for 19% of the tokens. Total token supply is fixed with 44% allocated for all capital raising rounds.

Token Details	Details
Public Sale Date	1 Jul 2019
Sale Platform	Binance Launchpad
Token Symbol	ERD
Token Type	Utility
Issuance Platform	BEP2
Total No. of Tokens	20,000,000,000
Price per Token (Public)	$0.00065
Price per Token (Seed/Private)	$0.00050 (Private)
Accepted Tokens	BNB
Softcap	NA
Hardcap	USD 3.25m (Investors own 25% if Hard Cap reached)
Market Cap Valuation	USD 13m (Fully Diluted basis assuming hard cap reached)
Market Cap (Based on Initial Cir. Supply)	USD 4.875m
Token Lockup (Investors)	7.5% unlock at TGE, 92.5% unlock at 15.42% every quarter (Private)
Token Lockup (Team)	Lockup till 2020 and semi-annual vesting over 3.5 years thereafter
Token Distribution	Within 15 days after the end of token sale

The main token functionalities of ERD are of 3 folds:

- Currency/ payment transaction – ERDs circulated on the network will be used as a unit of exchange to quantify and pay fees
- Staking/ Mining – Users will be able to earn rewards and transaction fees when contributing to the network
- Voting – Tokens will be used for voting governance once this function has been enabled



Partnerships

Nash (formerly NEX) – A decentralised digital asset exchange (DEX). The partnership will explore the integration of Elrond blockchain and ERD into Nash, closing the gap between end users and Elrond's platform

typingdna – A behavioural biometrics company protecting users on how they type on keyboards. Elrond will integrate biometrics to further improve security

Smartbill – Fintech company providing SaaS services to SME for invoicing, accounting and inventory management. Elrond will explore potential integration of its platform to improve trust, transparency and traceability features

DSRL – A research lab with extensive experience across various distributed systems. The collaboration will focus on research-education themes on the applicability of blockchain technology

Netopia – An electronic payment processor in Romania processing more than $400m worth of transactions. The partnership would enable Elrond access to 6000+ merchants to accept ERD as a payment option

Holochain – A scalable distributed app with data integrity. The partnership will explore potential integration of both platforms to boost technology performance

Runtime Verification – A dynamic platform that analyzes programs when executed, providing results and bugs detection. The partnership will work towards better research and development in formal verification methods.

Community Engagement / Social Media

Elrond has hired both a communication and marketing personnel internally to drive community engagement. The team has extensive social media usage across Twitter, Medium, Facebook and Telegram, and constantly provides technical updates, explaining the technology in layman terms to educate users.

Here are the project social media statistics as per the report date:

- Telegram (English) – 13,200+ members
- Twitter – 15,400+ followers
- Medium – 1,900+ Followers
- Facebook – 3,100+ Followers

Competitor Analysis

Project Name	Elrond	Harmony	Zilliqa
Infrastructure Type	Blockchain Sharding Technology	Blockchain Sharding Technology	Blockchain Sharding Technology
Symbol	ERD	ONE	ZIL
Consensus Algorithm	SPoS	PoS	PoW
Programming Language	Golang	Golang	C++
Transaction per Seconds (TPS)	Theoretical: Unlimited Practical: 3,500 per shard (testnet)	Theoretical: 10m Practical: 1,000 per shard (testnet)	Theoretical: Unlimited Practical: ~600 per shard
Latency	5s	8s	120s
Interoperable	Yes	No	No
Product Stage	Testnet	Mainnet	Mainnet
Mainnet Launch	Q4 2019	Jun 2019	Jan 2019
Initiation Year	2017	2018	2016
Current Token Price	$0.003687	$0.0115	$0.0099
Market Cap Valuation	$27.7m (based on initial cir. supply)	$28.4m	$86.3m

Figures as of 16 Jul 2019, Source: Coinmarketcap

Strengths

- Testnet achieved a peak throughput of 65,000 TPS with 3,500 TPS per shard, relatively higher than most sharding projects currently in the market

- Adaptive state sharding with pruning to improve storage and bootstrapping

- SPoS consensus with an improved random function that reduces overall network latency to 5s

- Energy efficient sharding infrastructure that utilises PoS versus PoW

- Team has strong entrepreneurial and blockchain experience

- Great deal structure with low hardcap and low diluted market cap valuation (half that of past launchpad projects)

- Low initial market cap (based on initial circulating supply) of $4.875m provides greater upside potential

Weaknesses

- Roadmap has limited business and marketing plans, coupled with a lower proportion (8.5% versus 11.6% spent prior) of funds raised are reserved for this segment might cause a concern

- Test results of weaknesses are unavailable as no dApp has yet to be deployed to stress test the protocol and network. However, the project will embark on this stress test and will release results in the coming quarters.

- The project currently has a lower number of partnerships relative to its peers

Opportunities

- **Improvement to current sharding technology** – There are various ways to integrate different types of sharding mechanisms into blockchain. The first being energy efficiency where PoS algorithms are preferred over PoW due to the rising issue of energy wastage incur by the latter. Secondly, the use of chain pruning in adaptive state sharding to reduce storage costs is not widely used with the exception of Omniledger. Third, a robust random function to ensure fairness, at the same time reducing latency to its minimum.

- **Cross chain interoperability** – With the rapid development of blockchain, many varying technologies have emerged, causing interoperability to become more prevalent. The ability to operate across chains would make it easier for users to access the network, eventually increasing adoption. Within the sharding sector, there are limited solutions with interoperable features (with the exception of Dfinity), hence creating the opportunity for Elrond to capture this part of the market.

Threats

- Strong competition within the sharding segment with the likes of Zilliqa, Harmony, Quarkchain, MultiVAC, Dfinity, and Algorand

- Adoption rate within the sharding technology sector has been slow, which could cause drag on the project’s growth

Conclusion

In conclusion, Picolo Research presents a **‘Buy’** and **4.5 stars** rating on Elrond. The project is seeking to scale the limitations of blockchain through adaptive state sharding technology while maintaining features of decentralization, security and fairness. Initial test results have shown an average throughput of 1000x increase over existing solutions at low transaction costs.

Overall, for the reasons listed, Picolo highlights several reasons that affirm our rating:

- Great design architecture of Adaptive state sharding and SPoS that lowers latency, bootstrapping, and storage costs

- Strong differentiating factors with shard pruning, cross chain interoperability that is EVM compliant, and low network latency of 5s

- Testnet throughput achieved 65k peak TPS with 3.5k TPS per shard, which is one of the highest amongst its peers on per shard basis

- Great deal structure with low hard cap and initial market cap, reflecting significant potential upside

- Technical review reveals a novel architecture with a high possibility of success in code implementation

Not withstanding the above, Picolo acknowledges several concerns of Elrond. Strong competition within the sharding sector could stump the platform’s growth rate in adoption. The project’s moderate number of partnerships compared to its peers, with a lower proportion of funds allocated to business and marketing plans and little mention of these in the roadmap, further adds to the aforementioned concern.

However, our communications with the team have clarified that lower marketing budget will not compromise their growth plans as their strong relationships have helped to reduce costs and that greater emphasis is given to operations as that is their immediate focus. In addition, the team also mentioned that several partnerships pipelines are already in place and are working with a PR company to push their communication and marketing efforts further, concurrently releasing a report on the developments within this area shortly. Therefore this mitigates the weaknesses and our concerns to a large extent.

Overall, in light of the preceding, Picolo Research affirms a **‘Buy’** rating on Elrond.

About the Analyst

Lennard specializes in fundamental and technical analysis in digital asset investments. He became acquainted with blockchain, cryptocurrency and ICOs in 2016, and recently decided to take a meaningful step away from traditional banking to join this industry. Previously, Lennard spent 3 years with an investment bank in forex and debt capital markets. Prior to this, he also had entrepreneurship experience working with an e-commerce startup and a local social enterprise. Lennard graduated with a master's degree in Applied Finance and is fluent in English, Chinese with a basic in Korean. He is a CFA level III candidate.

Ratings Definition

Monitor – Continue observation until clarity of information is provided

Sell/ Avoid – Investment is associated with high risk of losing capital

Hold/ Neutral – To maintain current levels of position until the next updated release

Spec Buy – A speculative opportunity for investors with higher risk tolerance

Buy – A high conviction buying opportunity

Disclaimer

This report has been compiled by Picolo Research. Picolo Research is an independent provider of research on cryptocurrency ICO's. In some instance, Picolo Research might be paid or mandated for the preparation of this research report However, the views expressed within this report are Picolo's in its entirety

The contents of this report and its attached documents have been prepared without taking account of your objectives, financial situation or needs. Because of that you should, before taking any action to acquire or deal in, or follow a recommendation (if any) in respect of any of the financial products or information mentioned in or downloaded from or through this website, consult your own investment advisor to consider whether it is appropriate having regard to your own objectives, financial situation and needs.

Whilst Picolo believes the information contained in this report is based on information which is considered to be reliable, its accuracy and completeness are not guaranteed and no warranty of accuracy or reliability is given or implied and no responsibility for any loss or damage arising in any way for any representation, act or omission is accepted by Picolo or by any officer, agent or employee of Picolo or its related entities. Picolo at all times reserves the right to at any time vary, without notice, the range of services offered by Picolo and its subsidiaries, and the terms under which such services are offered. The information in this report may have been used by Astronaut Capital (www.astronaut.capital) in making an investment decision. The information within this report is our own opinion only and is not to be used in making a decision for investment.

Contact us

w: www.astronaut.capital | www.picoloresearch.com

e: admin@astronaut.capital

a: 3 Fraser Street, DUO Tower, Level 5, Singapore 189352