

# RASMART

## PITCH DECK





**Rasmart** is a platform based on Blockchain, specializing in expanding the potential of using financial services on the basis of a distributed registry, self-executing smart contracts and cryptocurrencies.

### PROBLEMS

Low network capacity

Small latency

Transaction cost

Smart contacts restrictions

External interaction

### SOLUTIONS

Volume of more than 400 000 transactions per second

Block processed around 0.2 seconds

Fees lesser 0.000005 USD

Advanced smart contracts features (cycles, schedules, etc.)

Advanced API for easy platforms integration

# USE CASES

Blockchain makes it possible to scale not only entire industries, but also applications, services, users who need these extra capabilities. The market potential is estimated at trillions of dollars.



Financial and  
loan services



Internet of  
things (IoT)



Retail



Logistic



Medicine



KYC Exchanges and  
other identification

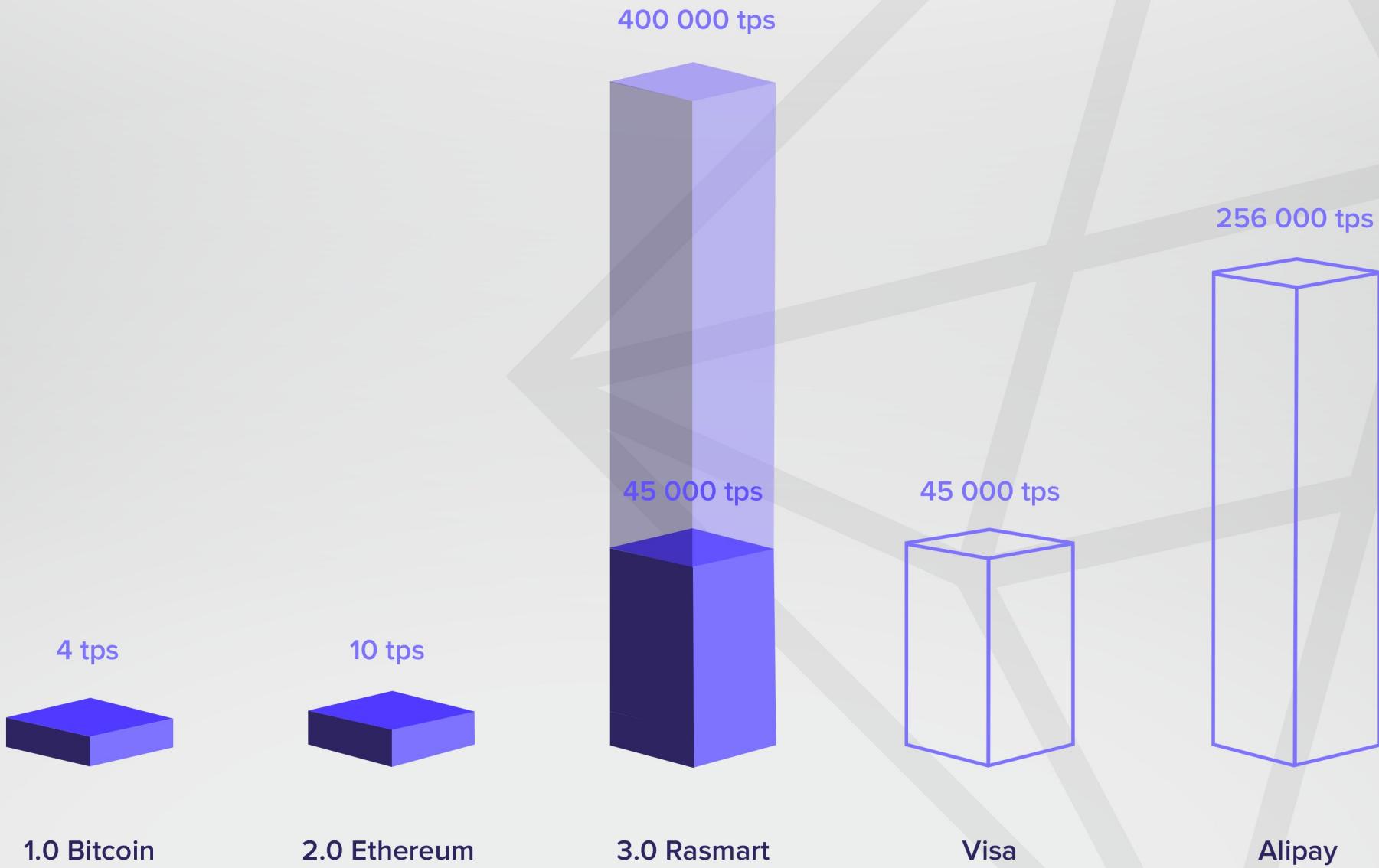


Exchanges



Intellectual  
property

# RASMART PLATFORM



# WHY RASMArt IS SO FAST?

1. Platform optimization for transaction processing speed

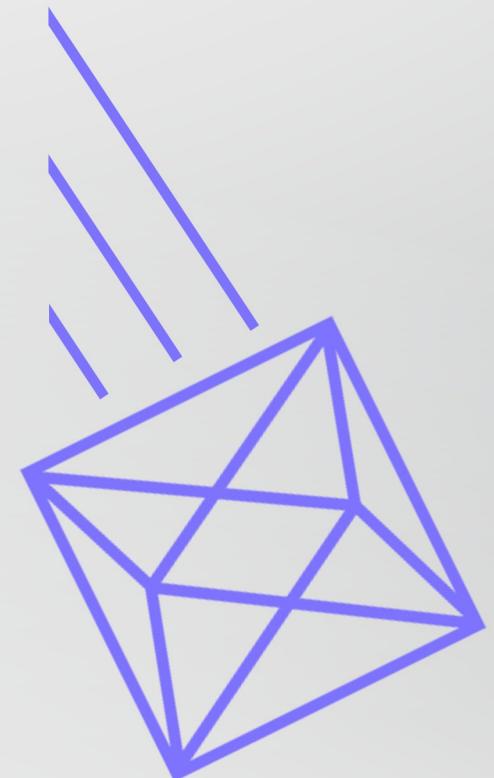
2. Reduction of data packet size

3. Distributed transaction pooling

4. Employment of dynamically optimized data flow in a distributed network

5. Employment of findings from stochastic simulation of similar systems

6. Optimization of the transaction pool access and storage mechanism



# SMART CONTRACT FEATURES



## Cycles and schedules

Smart contracts platform offers truly new features for independent services



## Advanced application programming interface (API)

Allows to connect from smart contracts to other outside systems



## Software development kit (SDK)

SDK provides advanced tools for developers and users



Smart contracts are developed in the **Lua language** and run in a lightweight virtual machine inside the node software. Further transition to LuaJIT will give an additional increase of 1-1.5 if it's necessary

# PLATFORM SECURITY OVERVIEW

Generation of constant chain of pools Due to first-come, first-served basis in the case when permutation of blocks, the sequence of time tags is incorrect and hash sums do not conform to the initial data, which renders them incorrect

Consensus algorithm in networks The Byzantine Fault Tolerances algorithm allows generals to control network status and exchange messages, thus ensuring honesty and making decisions on the transaction and pool validity

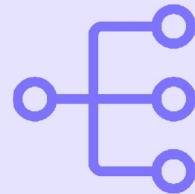
Hash sum generation Platform employs data encryption algorithm known as BLAKE2b which is not sensitive to the size of hash data and is protected against attacks related to the emergence of collisions during hashing

# USAGE OF RAS COIN

1. Payment for the smart contract creation
2. Performing operations on smart contracts
3. Adding information on the operation to blockchain
4. Payment for the transfer of RAS cryptocurrency between the platform participants
5. Purchase of information from third-party sources for services within the system
6. For operations on the exchange of different currencies within the system
7. Fee for the transfer of tokens created on the RAS platform



# PLATFORM COMPONENTS



**Node**  
Desktop Application



**Wallet**  
Webversion



**Monitor**  
NetworkExplorer

# NODE

A Node is a client side application that is installed on the user equipment. The Node performs the processing and storage of transactions, the execution and confirmation of the smart contract rules, request processing from third-party systems, and provides data when requested

# MONITOR

Multifunctional web-based tool for monitoring the history of the transactions of all network members, distribution of nodes around the world and obtaining general statistics.

**Total number of network nodes**

**Statistics about execution of transactions**

**List of transactions for the specific account**

**General information about the smart contract**

**List of transactions for the specific smart contract**

**Results and status of performance of smart contracts**

# CONSENSUS PROTOCOL

**1.** Selection of head and trusted nodes (DPOS)

- A.** Checking last block hash
- B.** Creation proper nodes list

**2.** Transaction voting procedure (BFT)

- A.** Creation of transaction list and sending it to trusted nodes
- B.** Generation of verified transaction list and exchanging between trusted nodes
- C.** Final verified transaction list creation

**3.** Protection against ledger recording by a malicious nodes (BFT)

- A.** Creation of new block
- B.** Sending new data to all network participants then back to 1a

# ROAD MAP

## Stage 1 2017 Q4 - 2018 Q2

- Idea generation
- Organization of the team
- Development of the functions of the decision-making module

## Stage 2 2018 Q3 – 2018 Q4

- Prototyping of the visual part of the project
- Private platform presentation, sale of tokens
- Testnet optimization
- Search for partners

## Stage 3 2019 Q1

- Refinement decision module
- Optimize API usage
- Implementing a cluster of multiple entry points
- Basic functionality of smart contracts
- Rebranding the visual part
- Wallet implementation for various platforms
- Preview version of the service data collection controller
- Public tokensale

## Stage 4 2019 Q2

- Full implementation of the decision module
- Developing a split API model
- Database optimization (big data, high load)
- Implementation of the basic version of the SDK for the development of smart contracts
- New design of loading large amounts of data in explorer
- Implementation of an additional layer of data encryption
- Refinement of the functional of the collection of reliable financial data

## Stage 5 2019 Q2 - 2019

- Decision Module: Parallel Processing of Pre-Transaction Pools
- The release of providing trusted data in a secure module for collecting reliable financial data
- Refinement API functionality
- Release of the release version of the smart contract SDK
- Refinement of the web wallet security system
- Wallet release version

# API

API it is a set of clearly defined methods of communication between various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. To communicate with third-party services on the Rasmart platform, the Apache Thrift technology is used. It combines a software pipeline with a code generation engine for developing services that are more or less efficient and easy to use between languages such as:

C#

Perl

C ++

PHP

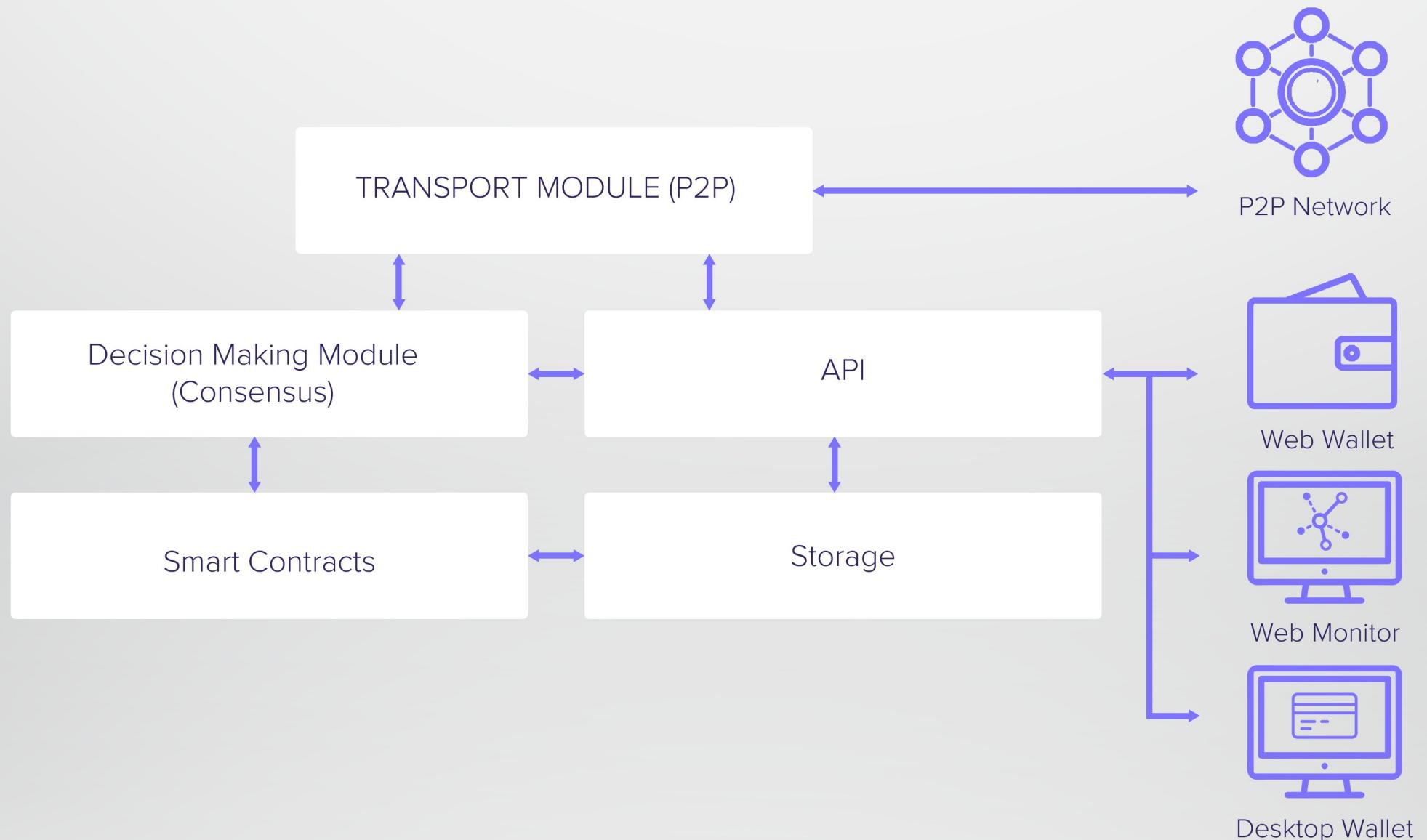
Delphi

Python

Java

JavaScript

# P2P NETWORK



# RASMArt INITIAL COIN OFFERING

## TOKEN DETAILS

Token name: RAS

Token type: Utility

## TOKEN FEATURES:

Token price: \$ 0.23

Hardcap: \$ 50 000 000

Max tokens: 500 000 000 RAS

Minimum buying transaction: \$

Accepted currencies: ETH

## CROWDSALE DETAILS

**500 000 000 RAS** - Initial issue of tokens

**225 000 000 RAS** - ICO

**67 000 000 RAS** - marketing and advisors

**90 000 000 RAS** - platform support

**59 000 000 RAS** - reserve fund

**59 000 000 RAS** - team and associates

## ICO schedule

Stage name	Duration, days	Min. purchase, USD	Bonus, %*
Private sale	60	10 000	30
Pre sale	20	5 000	20
Main sale	10	500	5-10

\* – for large purchases at all stages it's possible to give an extra bonus, for which tokens will be taken from Company fund/Team allocations

# TEAM



**Raag Alexandr**  
**CEO**  
Senior C++ developer, Extensive experience in startup



**Dergachev Boris**  
**CTO**  
Senior C++ developer, have experience in Aeroflot school, MapVis



**Ten Inessa**  
**Community**  
Public Relations in Korea



**Levshina Svetlana**  
**CFO**  
(MSU) Lomonosov Moscow State University, 27 years of successful business experience, Extensive experience in Venture Capital

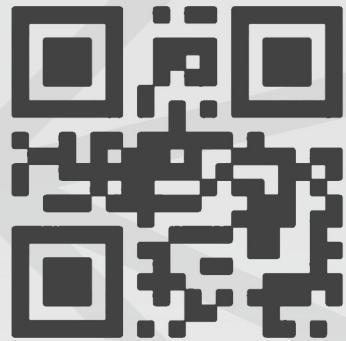


**Safronenko Sergey**  
**COO**  
Russian Academy of National Economy and Public Administration under the President of the Russian Federation



**Cherkasov Vladislav**  
**CMO**  
Crypto-markete with 2 years of experience working with the blockchain startups

CONTACT US



[rasmart.io](http://rasmart.io)