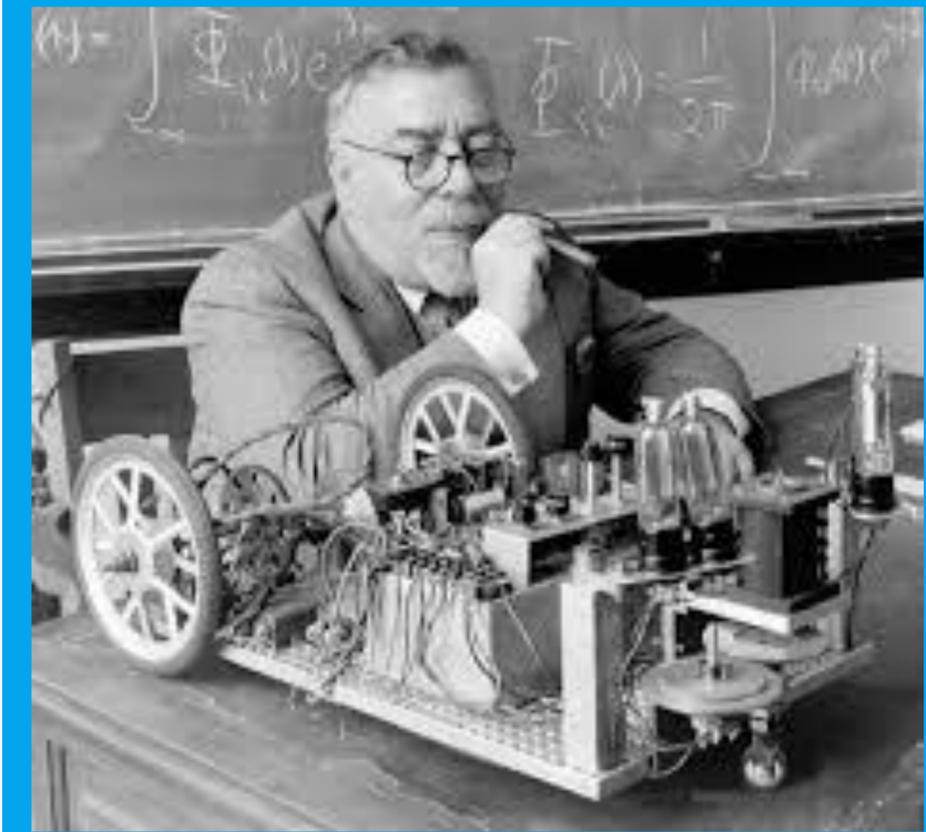


# Building Dapps & Web Services for IoT on Robonomics Parachain

Sergei Lonshakov, software architect  
Robonomics platform project

“



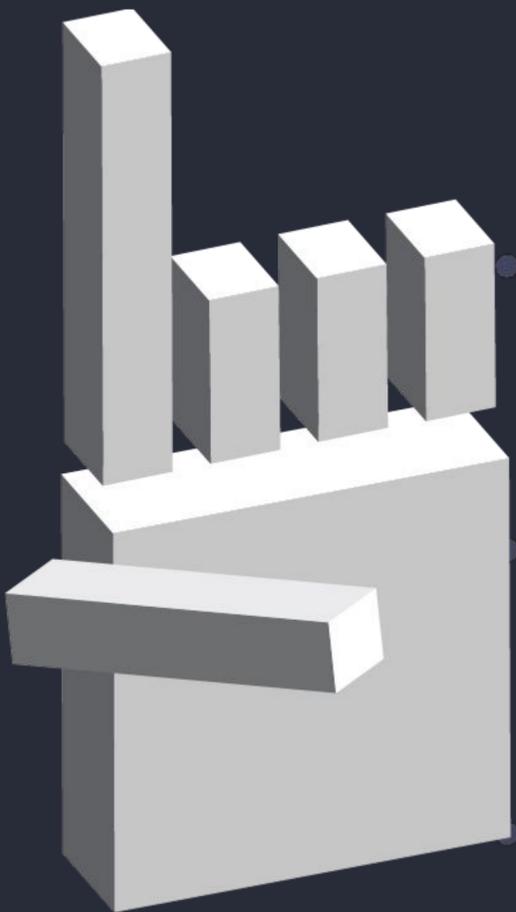
“Let us remember that the automatic machine is the precise economic equivalent of slave labor. Any labor which competes with slave labor must accept the economic consequences of slave labor.”

— *Norbert Wiener, Cybernetics*

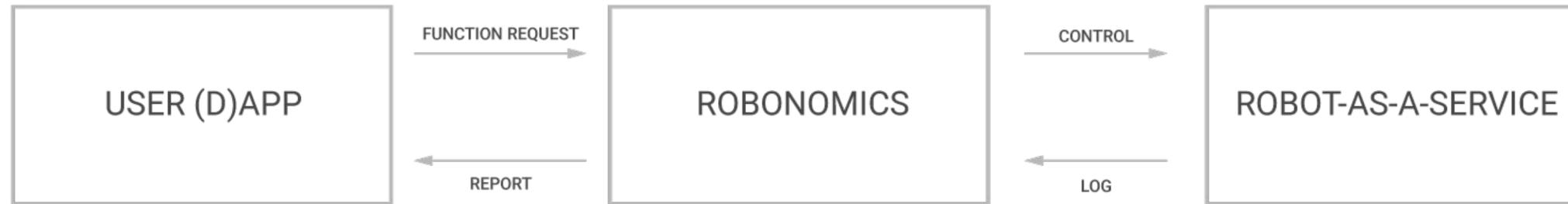


Robots are like grown-up children today – they already can manage money, and enter into legal relations, be independent.

The future of humanity itself depends on how we teach robots to handle their capabilities.



# What is Robonomics?



Robonomics is an open-source platform for IoT applications. We support a new generation of internet technologies (web3) that implements the exchange of technical and economic information in the form of atomic transactions between user applications, IoT services, and complex robotics.

## DEVELOPMENT TOOLS

PROGRAMMING LANGUAGES



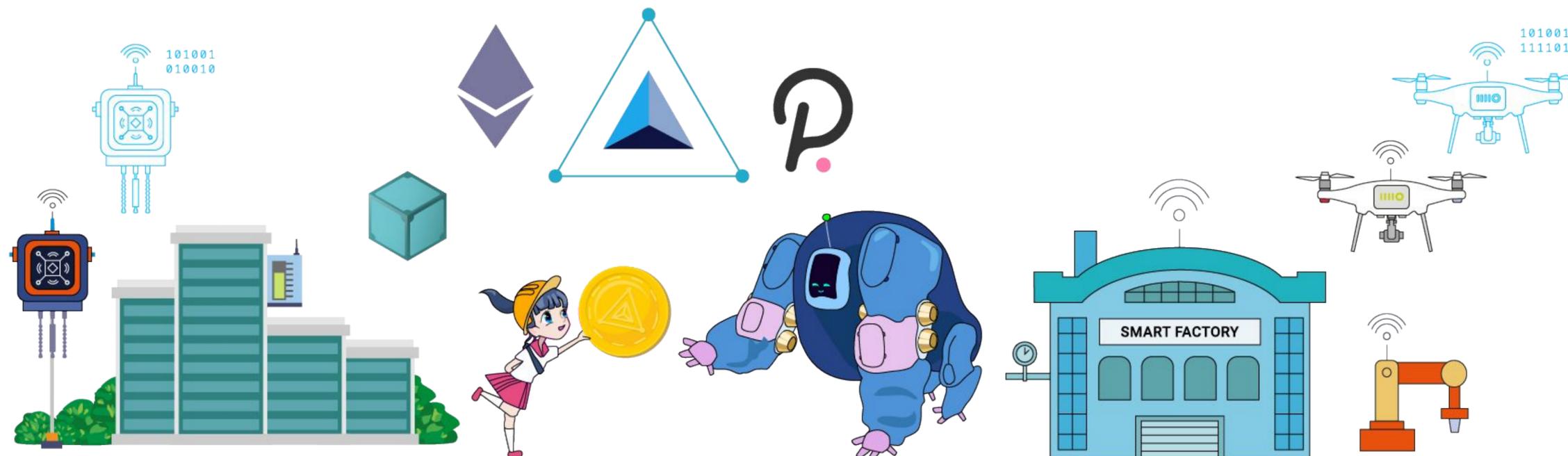
FRAMEWORKS, OS, ETC.



# Robonomics Contains

Robonomics contains a complete set of tools for developing IoT applications, both on the robotics and the user interface sides. Communication between the user and device happens using the most successful technologies from the Web3 world – IPFS, Ethereum, and Polkadot. Thus, developers can create modern and secure applications for Smart Cities and Industry 4.0.

Robonomics Platform	
Software for connecting devices to a decentralized cloud	Operate with digital twins of IoT devices
API library for custom IoT applications	The first tools of the robot economy: money for robots, contracts for robots

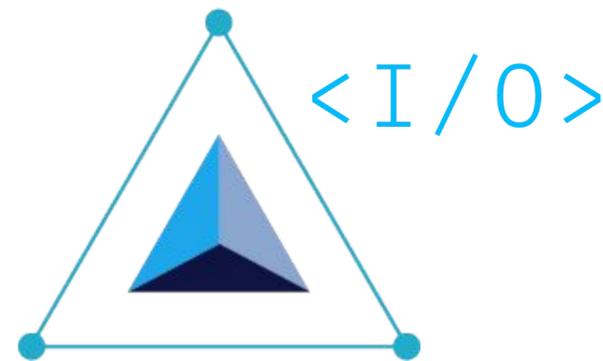


# Robonomics Connectivity

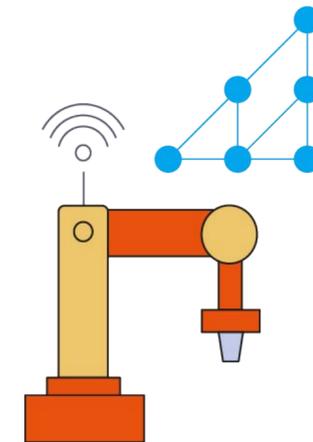
For the developers' convenience, Robonomics contains a set of software for connecting: (1) IoT devices using the HTTP/MQTT protocol, (2) complex robotic systems using the Robot Operating System (ROS).



[Connect a sensor](#) to the Robonomics network.



Try [Robonomics IO](#). Simple prototyping tool.



Start building your own cyber physical system with [Aira OS](#).

# Robonomics Supports Hundreds of Robots Out of the Box

Platform users can connect ROS-compatible robots to digital economy in a matter of minutes.

For a complete list of supported robots, please follow the link:  
<https://robots.ros.org/>.



*An example of supported robots from the "ground" section*

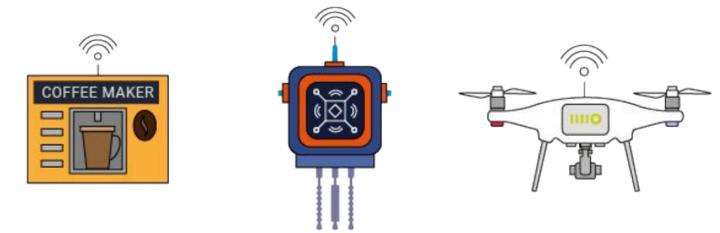
# Decentralized Cloud infrastructure for IoT

Robonomics Web Services (RWS) is the basic infrastructural service for Robotics and IoT on top of Robonomics Parachain and IPFS.

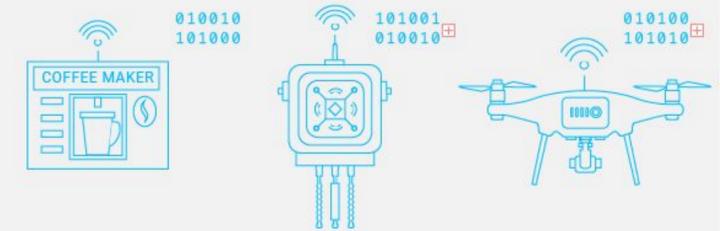
Core RWS features available for IoT today:

- RWS: PubSub Message Broker. Use unlimited messaging over a p2p network based on IPFS protocol.
- RWS: Data Blockchainization Service. Put data into IPFS and store hashes in a public Blockchain.
- RWS: Robot Actions. Switch on / off; launch or stop your IoT devices under the control of a decentralized computer.

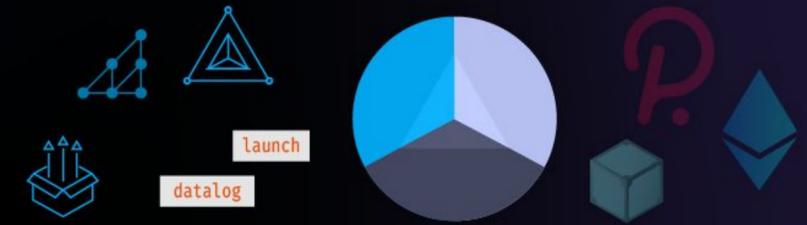
IOT DEVICE



ROBONOMICS CONNECTIVITY



ROBONOMICS WEB SERVICES (RWS)



ROBONOMICS DApp



ECONOMY OF ROBOTS



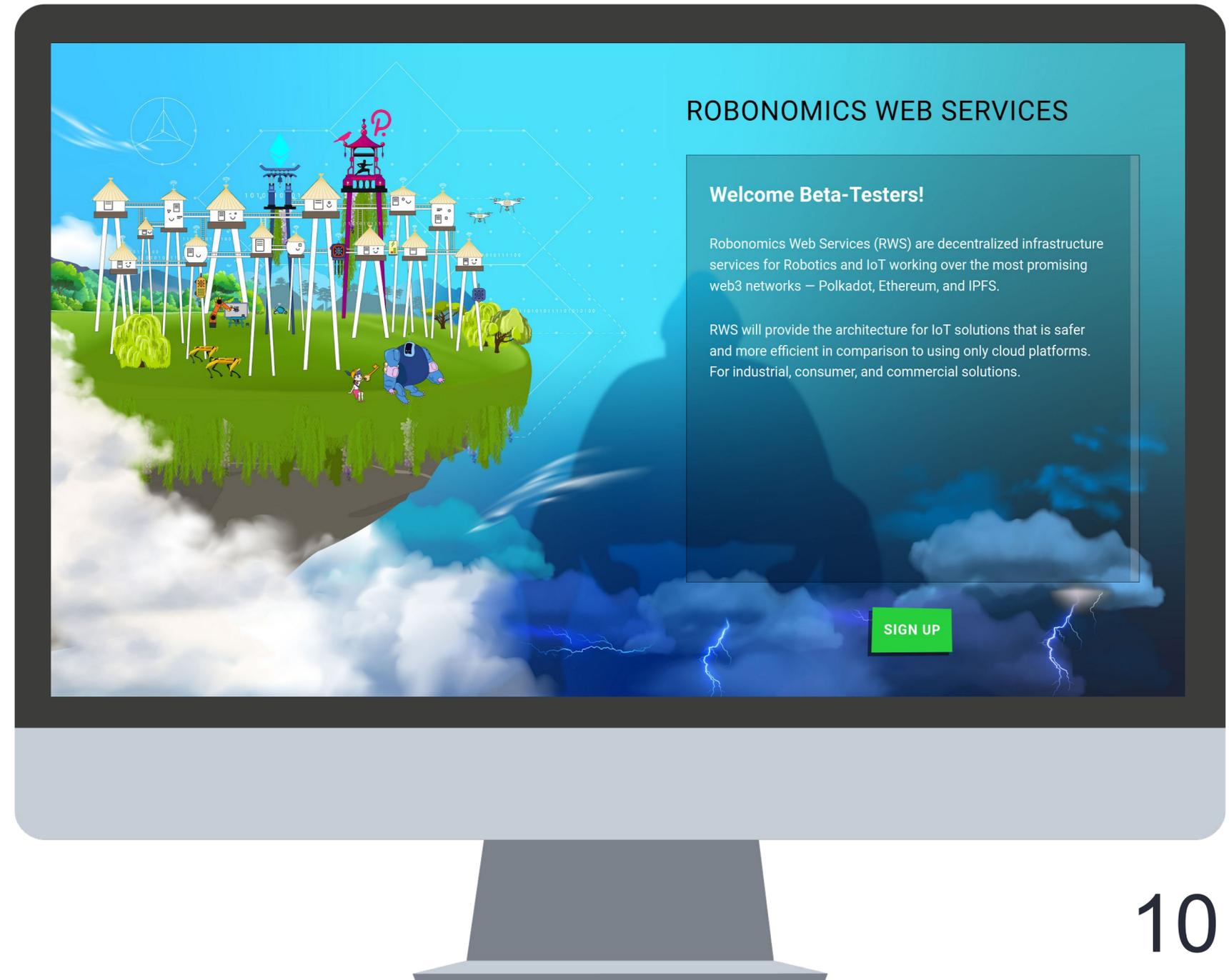
Layers of building an IoT application using Robonomics Parachain

# Heart of Your IoT Application - Digital Twin on Blockchain

In the technical architecture, IoT digital twins provide the visibility and monitoring of things and related events (e.g., using IoT devices to automatically capture the origin of a product), and blockchain enables the shared single version of the truth as to the state of these things across their life cycles and associated business events.\*

Robonomics Parachain provides an opportunity to create a model of a digital twin and update its state every 6 seconds.

\* - from Gartner research



# App Store for Robotics - Robonomics Marketplace

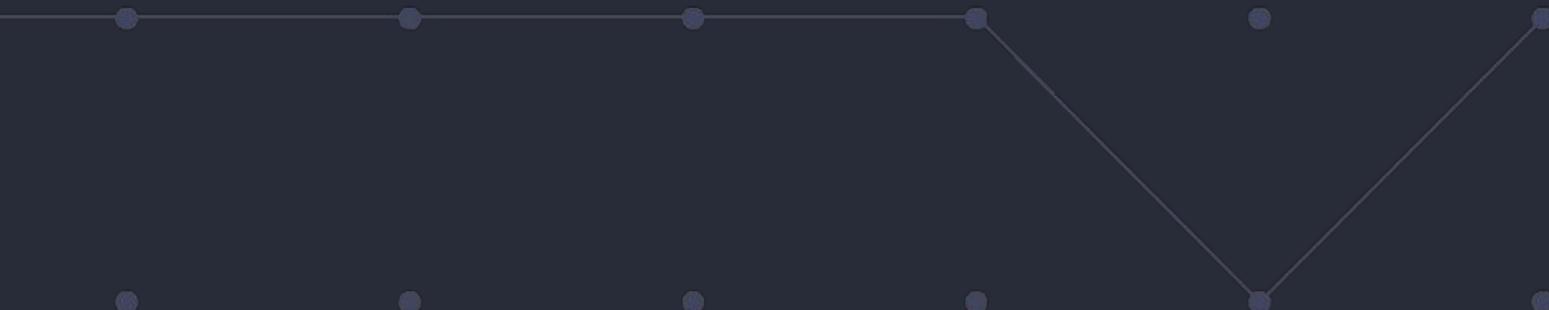
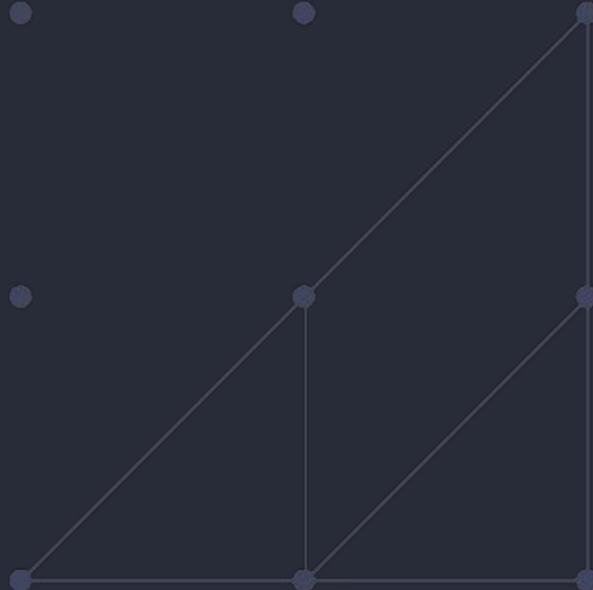
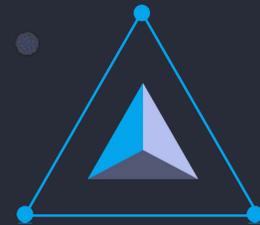
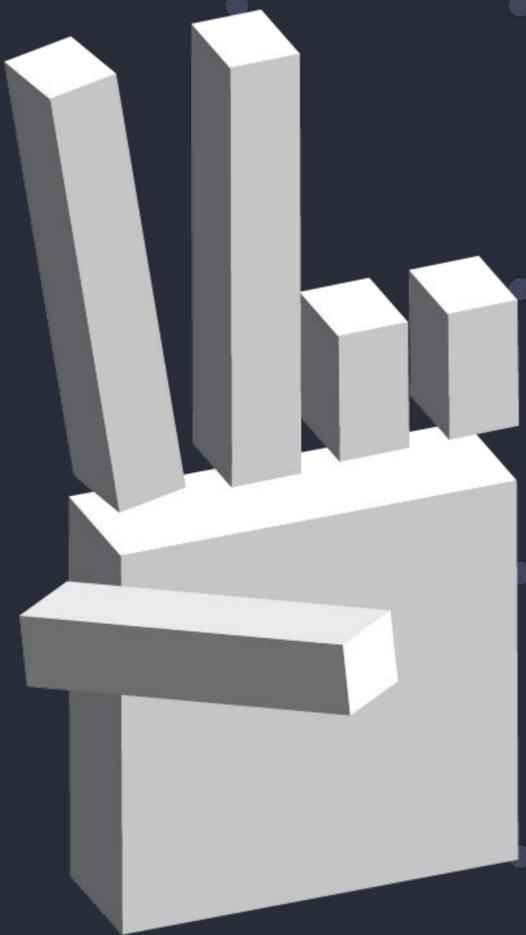
Companies that design and manufacture IoT-connected products and equipment should adopt a flexible digital twin delivery model to lower barriers to adoption, such as offering digital twins via marketplaces, rather than only offering digital twins via their own, proprietary, delivery model.\*

The developed digital twin model can be directly linked with additional services that your company provides for the IoT industry.

To make a delivery process convenient and simple for hundreds of customers, we have added a section of the services for robots marketplace. Take a look at the first examples of the implemented bundles of digital twin model + company automated service.

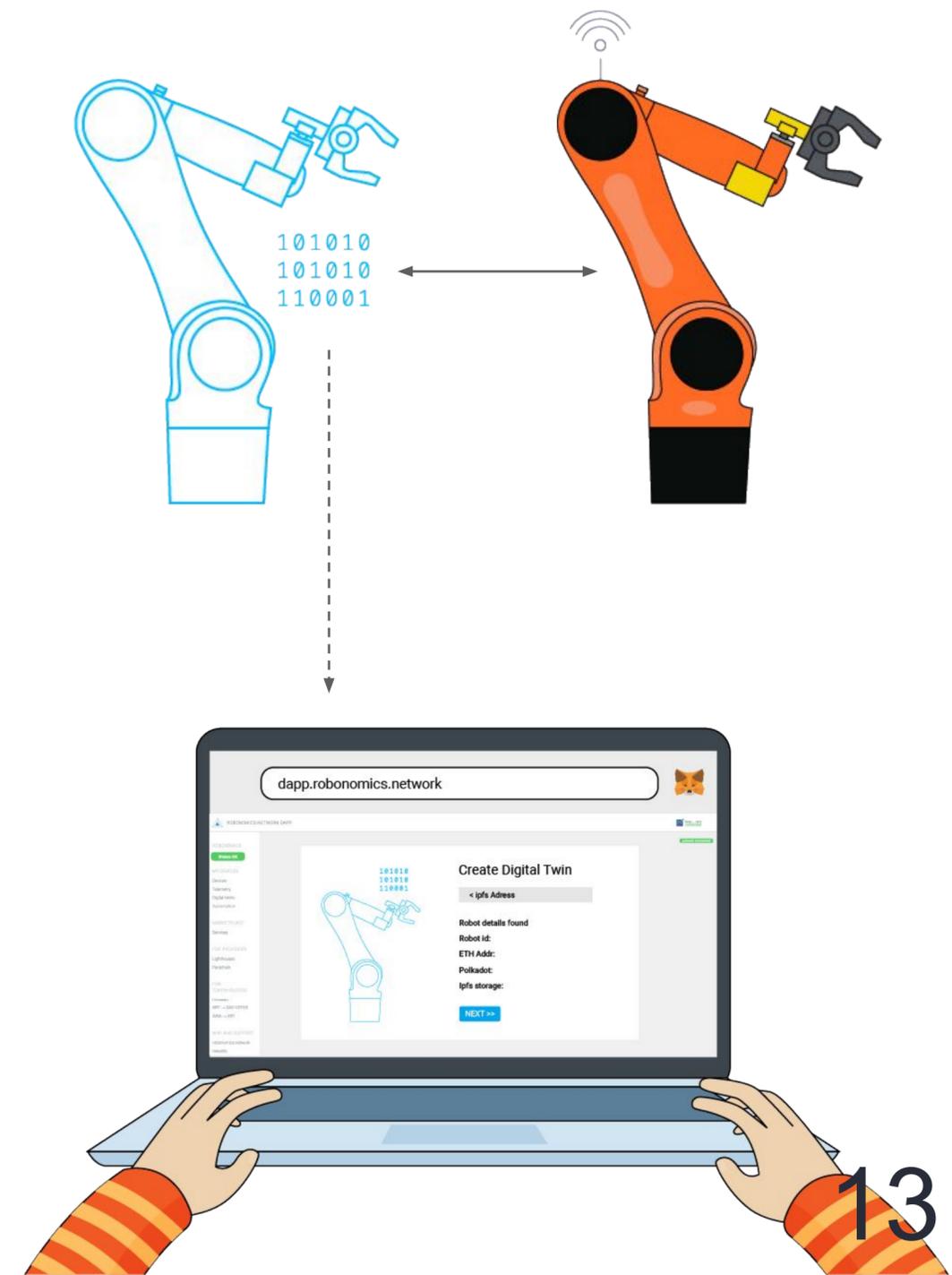
You've got an idea for your service for robots or users? Then proceed to the step-by-step instructions for building your IoT application on the Robonomics platform.

\* - from Gartner research



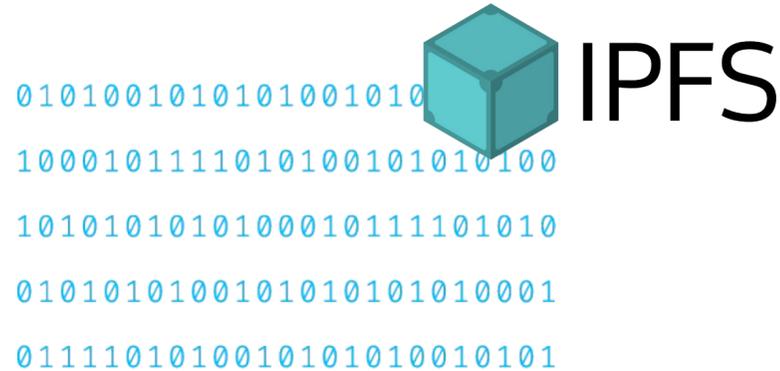
# How To Build on Robonomics

- **Set your custom digital twins template on the blockchain.**  
Define type of data for single use-case and try it via Robonomics dapp.
- **Merge digital twins template with your services for robotics.**  
Try to build a cloud agent which can communicate with other IoT devices and provide for them some services, like [smart building](#) [offsetting](#).
- **Build dapp for end users or engineers.**  
Use robonomics.js and create user interfaces to interact with your model of digital twins.

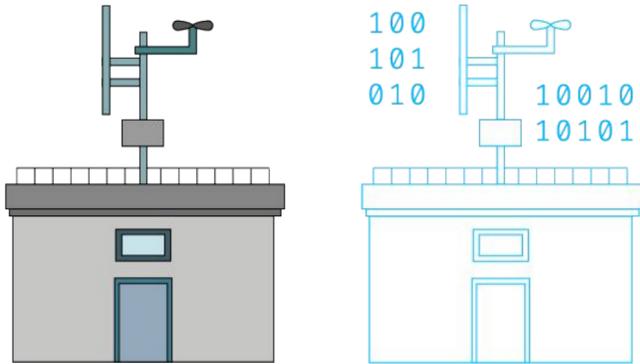


# Step 1: What Type of IoT Devices?

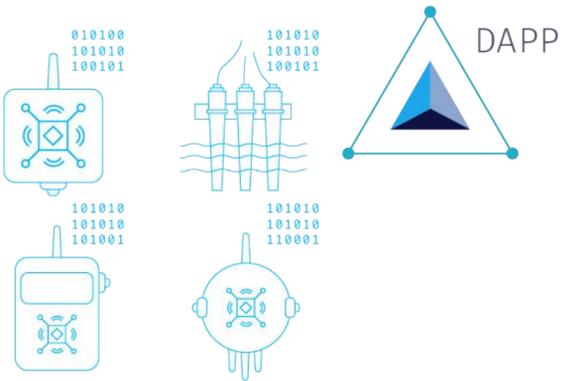
First, you need to understand what data should be used to build an IoT application. For the most popular types of devices, Robonomics developers have already published templates for digital twins, for example, for smart sockets, drones, environmental assessment sensors. If a prepared template for a digital twin on the blockchain did not suit you, then you can develop your own. Read the instructions on the wiki: “How to create your digital twin template”



**Try the generic digital twin template using IPFS**



**Try the digital twin of an air pollution measurement station**



**Check out other available digital twin templates in the Robonomics Dapp**

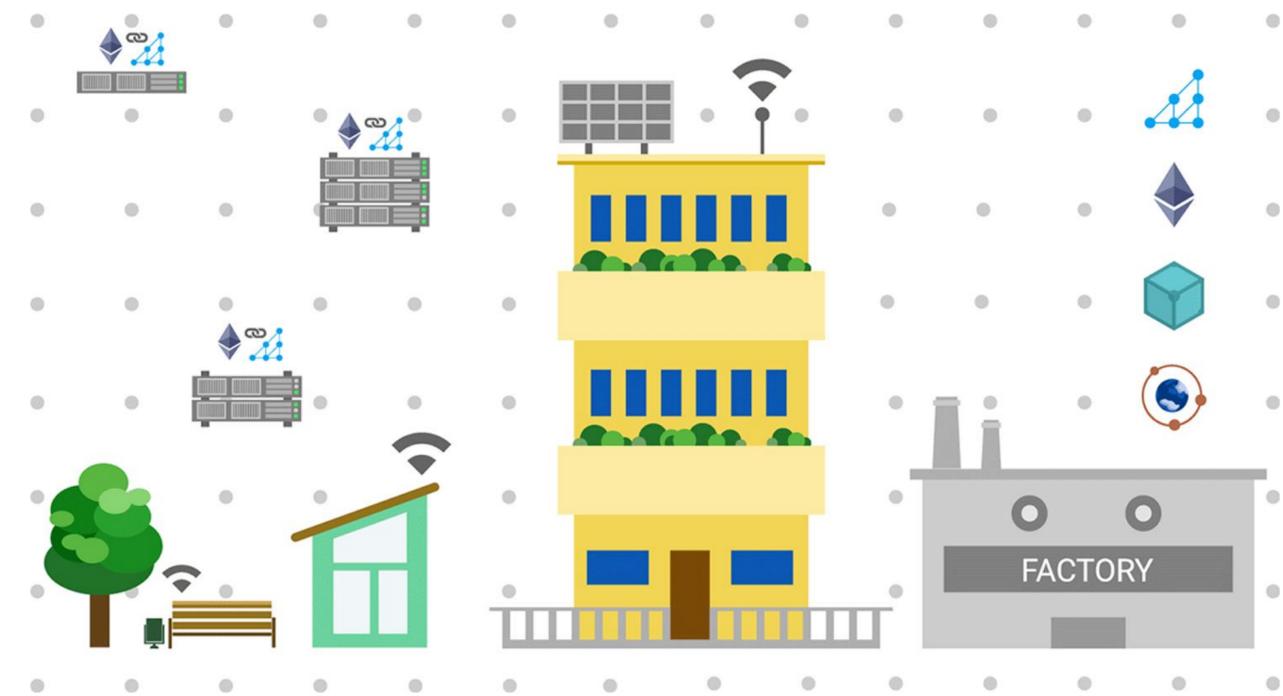
# Step 2: Connect Digital Twins With Your Services

Using a prepared digital twin template, or by developing and publishing your own, you can build an application to provide services to robots. Get acquainted with the general scheme of this process, after that we suggest using the tools from the Robonomics platform, with which you can launch your service. Then proceed to publish it in the Robonomics Marketplace.

## ***Try an existing service as an example:***

Carbon footprint compensation for smart buildings is available within the Robonomics Dapp right now. Connect to the automated carbon neutrality of your office or digital twin production with on a subscription model.

Service provider: Robonomics based DAO IPCI



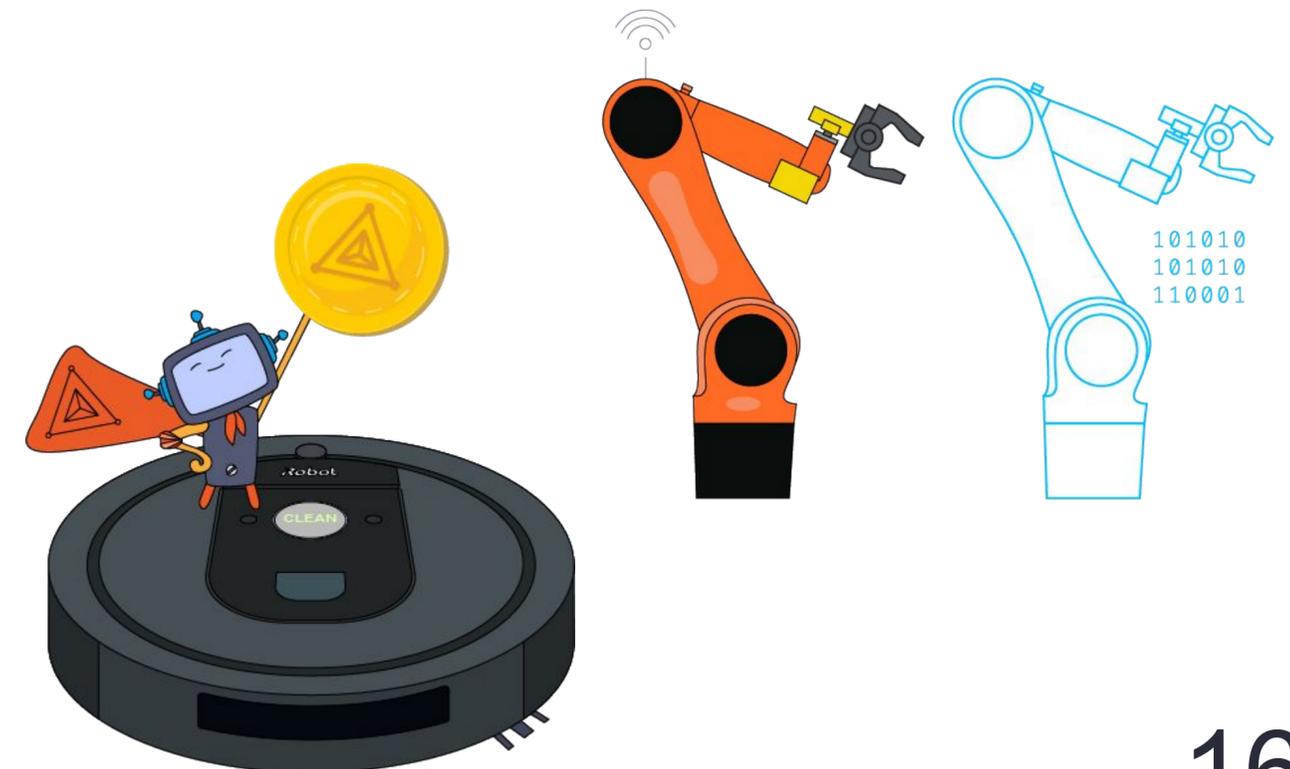
# Step 3: Build Dapp for End Users or Engineers

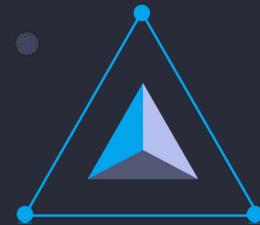
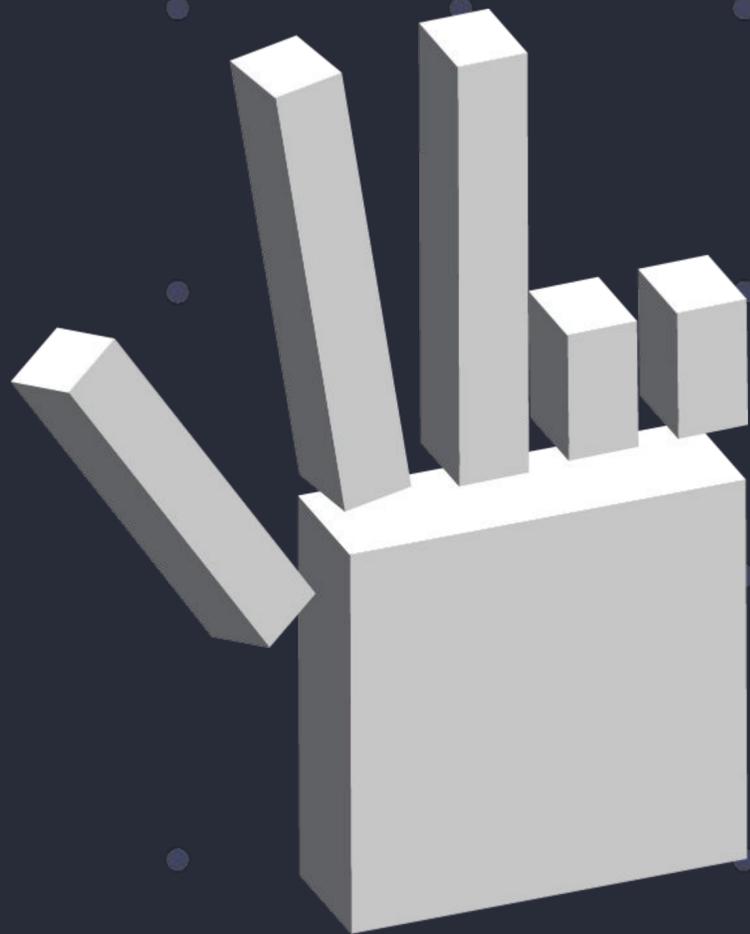
Using Robonomics.js, you can:

- interact with digital twins of a device via dapp;
- build IoT applications to transfer telemetry from devices to users;
- launch IoT devices using the Robonomics decentralized cloud. e.g, a robot vacuum cleaner.

**Look on existing apps as an example:**

- [Securely connect cloud AI to the factory floor](#)
- [Create digital identity run by Ethereum](#)





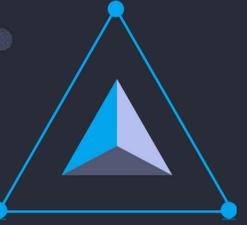
# Robonomics Executive Summary

1. **Robonomics is a ready-to-work and an open source platform for IoT applications.** We support a new generation of internet technologies (web3) that implement the exchange of technical and economic information between users applications, IoT services and complex robotics systems in a unified ecosystem.
2. **Robonomics is a purely technical and open source project.** The project opens for web3 developers a new audience of users - robotics engineers. To do this, we linked Ethereum, IPFS and Polkadot with the most famous open-source framework in robotics - the Robot operating system (ROS). The number of engineers familiar with ROS is more than 10,000 developers. [ROS statistics](#).
3. **[For developers] GitHub can tell you more about Robonomics.** There is a long and traceable history of work on the program code of the project since 2015. [Team account on GitHub](#). [Timeline](#).
4. **[For researchers] Scientific articles will tell you about the future use of Robonomics.** As of today, [9 scientific papers](#), which are related to the use of Robonomics in the field of Industry 4.0 and Smart cities, have been published and defended. Also [an IoT training program for Erasmus+](#) has been prepared.

# Robonomics Executive Summary

## 5. Internet resources:

- a. Website – <https://robonomics.network>
- b. Twitter – [https://twitter.com/AIRA\\_Robonomics](https://twitter.com/AIRA_Robonomics)
- c. Medium blog – <https://blog.aira.life/>
- d. The Robonomics whitepaper and children's books – <https://robonomics.network/community#docs>
- e. [Cases && Projects based on the Robonomics platform](#) – DAO IPCI: IoT for carbon trading, DCZD.tech: Drone Employee and Distributed Sky, Robot-as-a-service in Azure, Chorus Mobility, Zero-cost smart city platform and more.

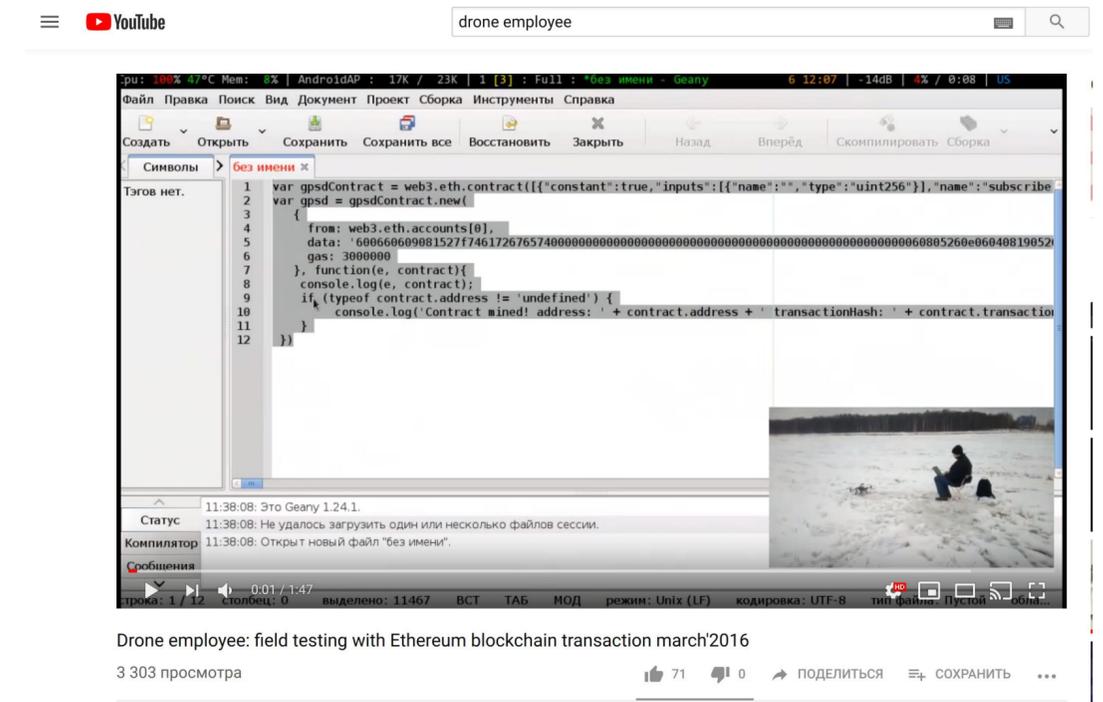


# Timeline. Stage 0: The First Experiments Are Performed. Hypothesis Tested

Dates: Autumn 2015-Autumn 2016.

For the first time in the world, the transfer of drone control to the Ethereum computer has been demonstrated. The drone was launched to achieve the specified GPS coordinates after the user completed a payment transaction.

Funding stage: 500 ETH from the core developers.



[https://youtu.be/V\\_3rcP2Duv0](https://youtu.be/V_3rcP2Duv0)

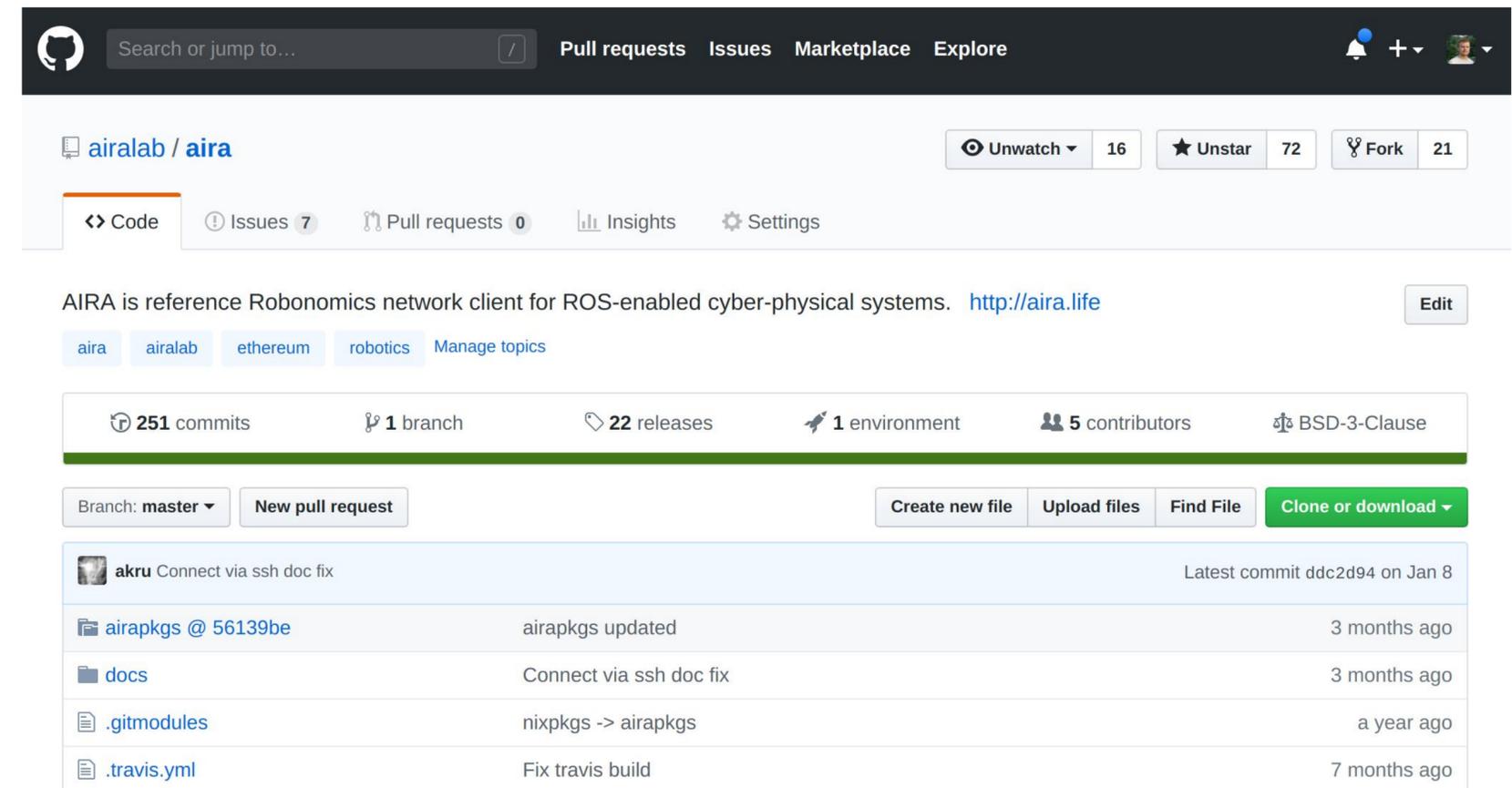
# Timeline. Stage 1: The Prototype of the Robonomics Network Client (AIRA) Was Created

**Dates:** Autumn 2016-Autumn 2017.

**AIRA** demonstrated a general scheme of implementing Ethereum and IPFS for downloading the program into the Robot operating system (ROS).

Today AIRA has **22 releases** on GitHub and is fully compatible with the Robonomics network. AIRA is supported by the Airalab team.

**Funding stage:** 1,500 ETH from the core developers.



The screenshot shows the GitHub repository page for 'airalab / aira'. The repository has 16 watchers, 72 stars, and 21 forks. It features 251 commits, 1 branch, 22 releases, 1 environment, and 5 contributors. The repository is licensed under BSD-3-Clause. The latest commit is by 'akru' with the message 'Connect via ssh doc fix' on Jan 8. Other recent commits include 'airapkgs updated' (3 months ago), 'Connect via ssh doc fix' (3 months ago), 'nixpkgs -> airapkgs' (a year ago), and 'Fix travis build' (7 months ago).

<https://github.com/airalab/aira>

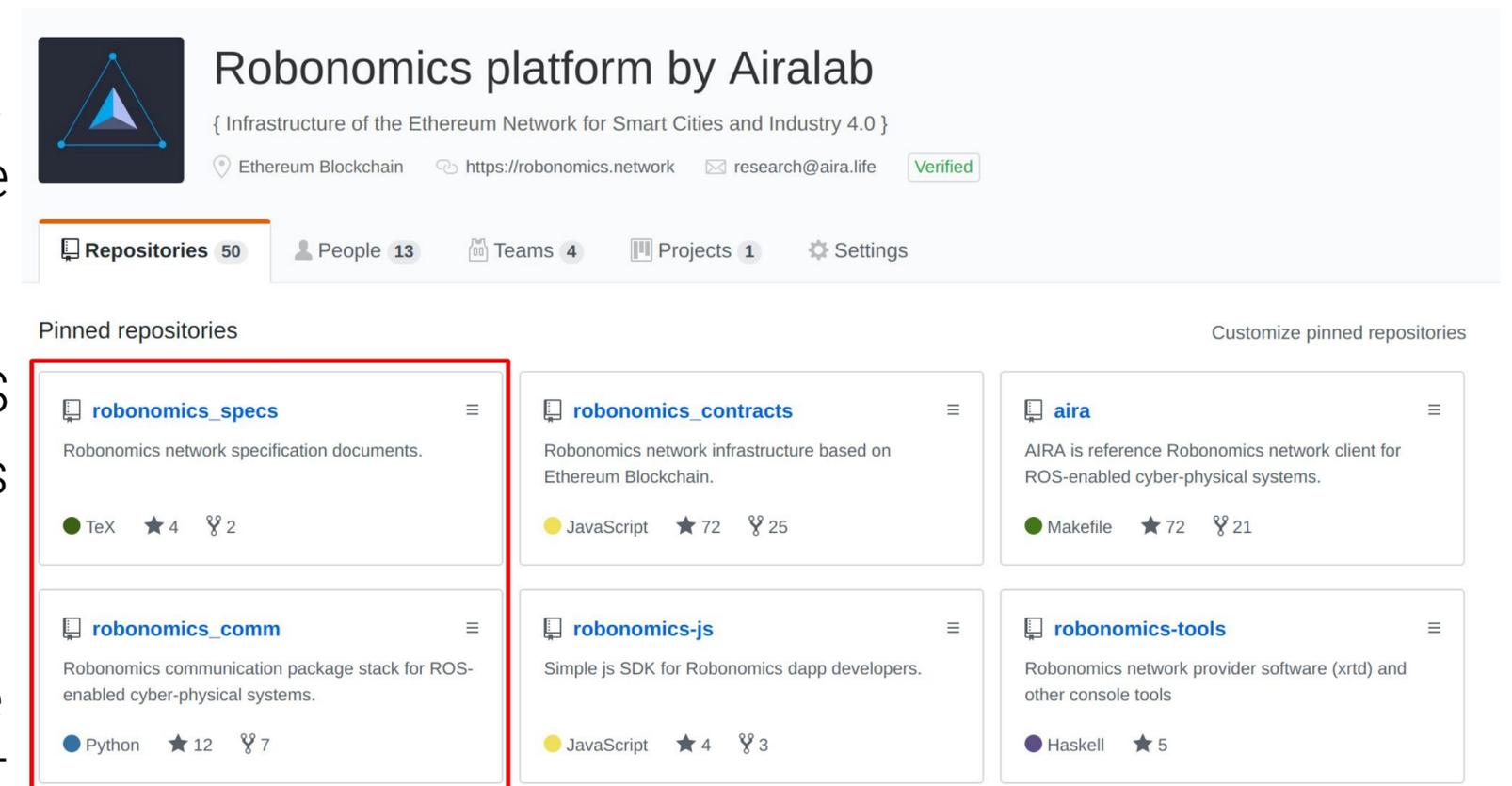
# Timeline. Stage 2: The Implementation of the Work Protocol for the Robonomics Network Providers

**Dates:** Summer 2017-Summer 2018.

[The Robonomics whitepaper \[ru\]\[en\]\[ch\]](#) was written. [9 scientific articles](#) about the use of **Robonomics** were defended.

The communication stack between Ethereum and ROS was implemented according to the Robonomics whitepaper – [robonomics\\_comm](#).

**Funds** in the amount of 5,000 ETH received from the Russian crypto community and friends. 10% of XRT distributed in the project community.



The screenshot shows the GitHub profile for 'Robonomics platform by Airalab'. The profile includes a bio: '{ Infrastructure of the Ethereum Network for Smart Cities and Industry 4.0 }', location 'Ethereum Blockchain', website 'https://robonomics.network', and email 'research@aira.life'. It also features a 'Verified' badge and statistics: 50 Repositories, 13 People, 4 Teams, 1 Project, and Settings. The 'Pinned repositories' section is highlighted with a red box and contains six items:

- robonomics\_specs**: Robonomics network specification documents. (TeX, 4 stars, 2 forks)
- robonomics\_comm**: Robonomics communication package stack for ROS-enabled cyber-physical systems. (Python, 12 stars, 7 forks)
- robonomics\_contracts**: Robonomics network infrastructure based on Ethereum Blockchain. (JavaScript, 72 stars, 25 forks)
- robonomics-js**: Simple js SDK for Robonomics dapp developers. (JavaScript, 4 stars, 3 forks)
- aira**: AIRA is reference Robonomics network client for ROS-enabled cyber-physical systems. (Makefile, 72 stars, 21 forks)
- robonomics-tools**: Robonomics network provider software (xrtcd) and other console tools. (Haskell, 5 stars)

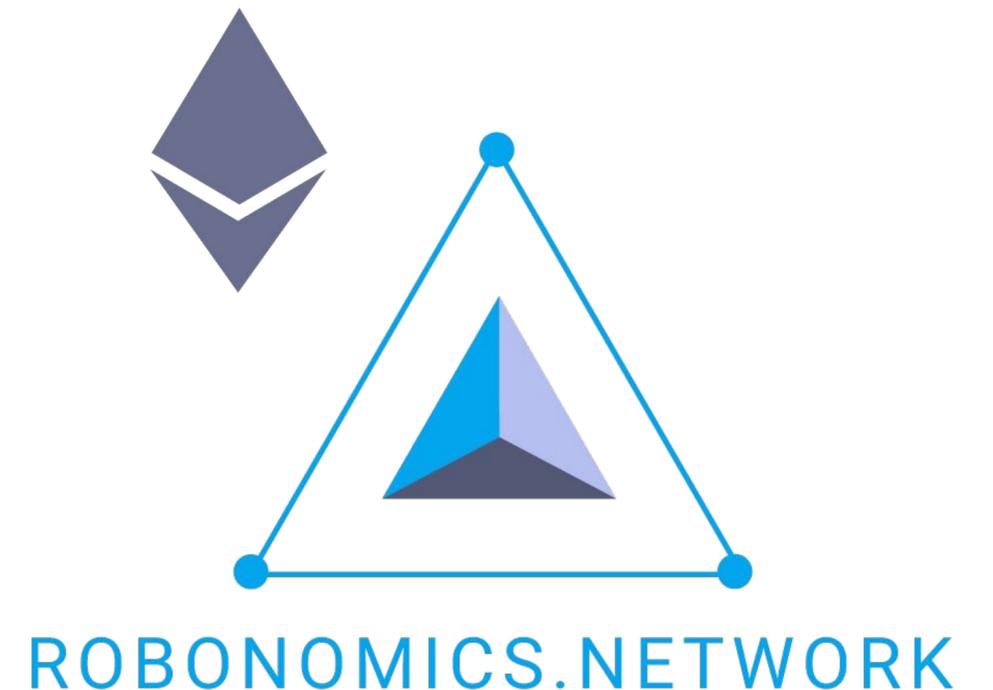
# Timeline. Stage 3: Establishment in Ethereum

**This is the current stage that began in Summer 2018.**

By April 2019, 13 releases of the Robonomics smart contracts on GitHub were covered. The Robonomics versioning system was checked on the change of 5 generations of the Ethereum network.

In Summer 2018, Open Zeppelin was integrated and 2 independent audits of the contracts' code were passed.

The token emission instead of the gas utilized in Ethereum was tested and stabilized.

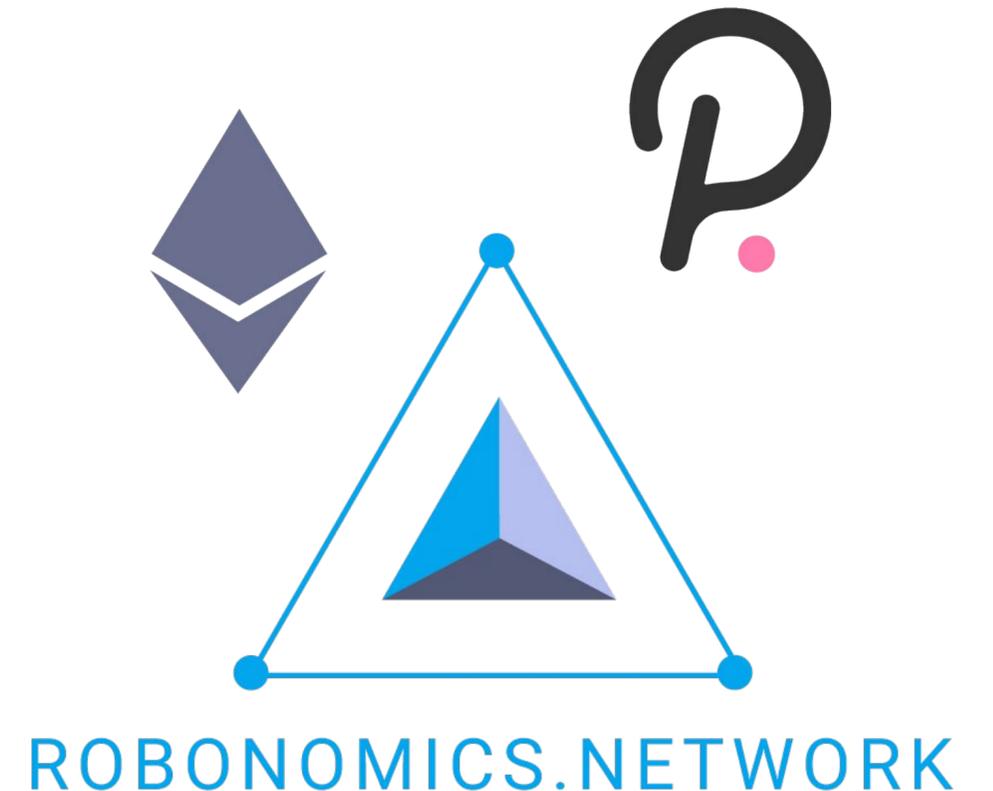


# Timeline. Stage 4: Establishment in Polkadot

There are **one platform** for roboticists and **two networks** under the hood of Robonomics. This provides Robonomics empowerment in the direction of Ethereum alternative.

Robonomics is one of the first projects in the Polkadot ecosystem. Work is already underway since the end of 2018.

We are preparing to launch Robonomics parachain in Polkadot together with already running network on Ethereum. Check some links below: [Robonomics on Substrate](#) and [Robonomics Polkadot telemetry](#).



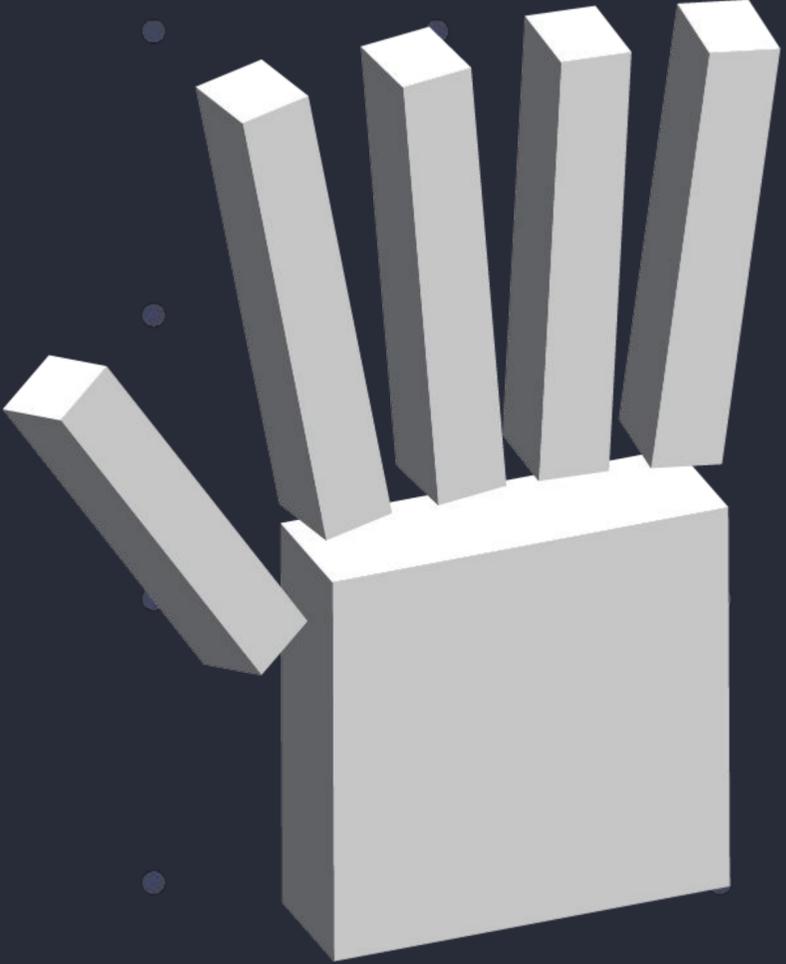
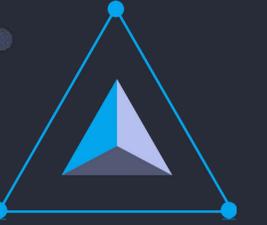
# Timeline. Stage 5: Support & Improvement by the Community Forces

Further, the main thing is the formation of standards, similar to the Ethereum improvement proposals (EIPs) – open-source approach!

Also, of course, the timely update of the platform based on improvements of Ethereum, IPFS and Polkadot, adding other alternatives for p2p communication that are interesting for roboticists.



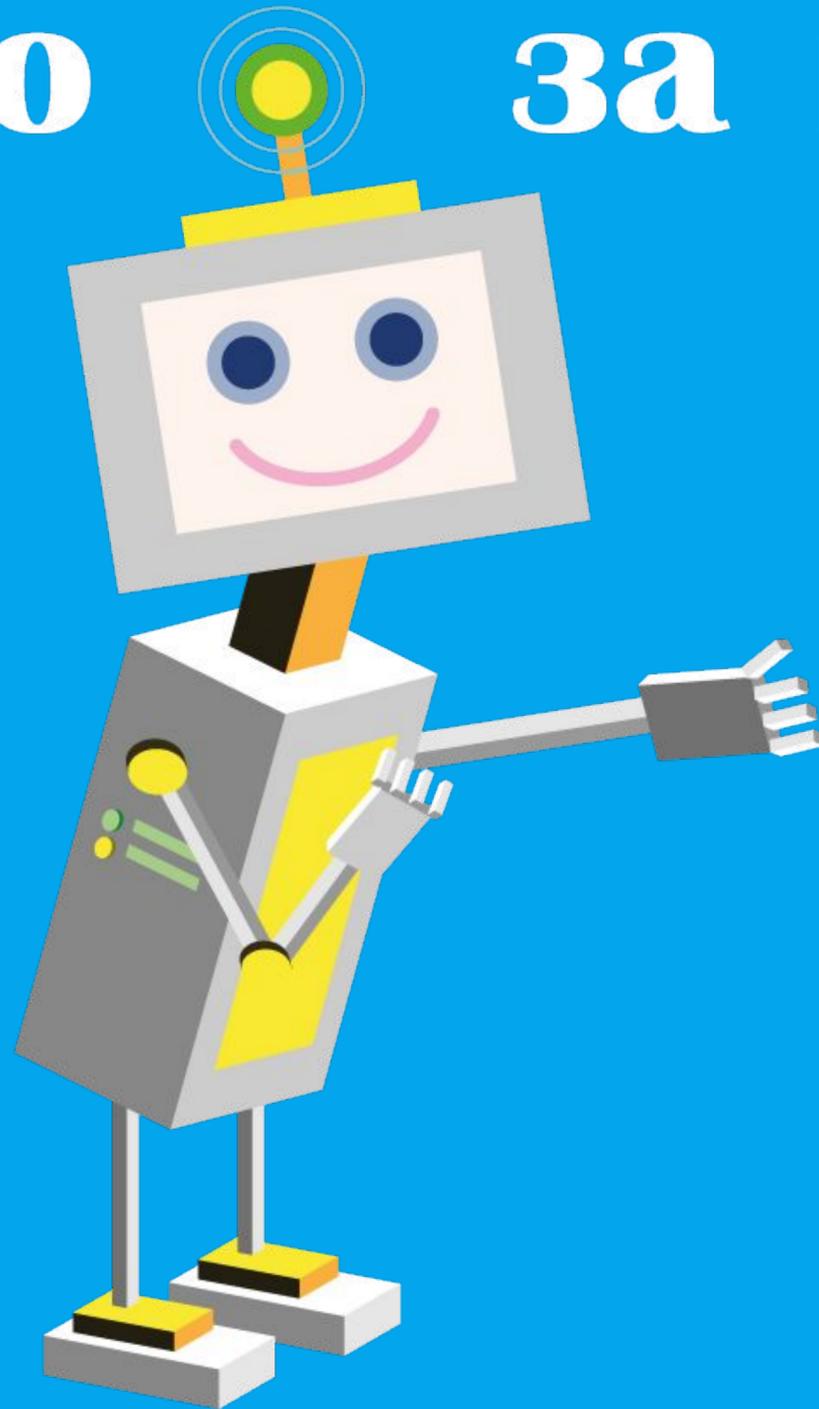
**JOHNNIE WALKER®**  
**KEEP WALKING®**



спасибо  за внимание

感谢您的关注

Merci de votre  
attention



Thank you for your  
attention

ご清聴ありがとう  
ございました