Garuda.AI

FORMAL VERIFICATION OF DLT SYSTEMS

# GARUDA AI PLATFORM

Viktor Radchenko CEO

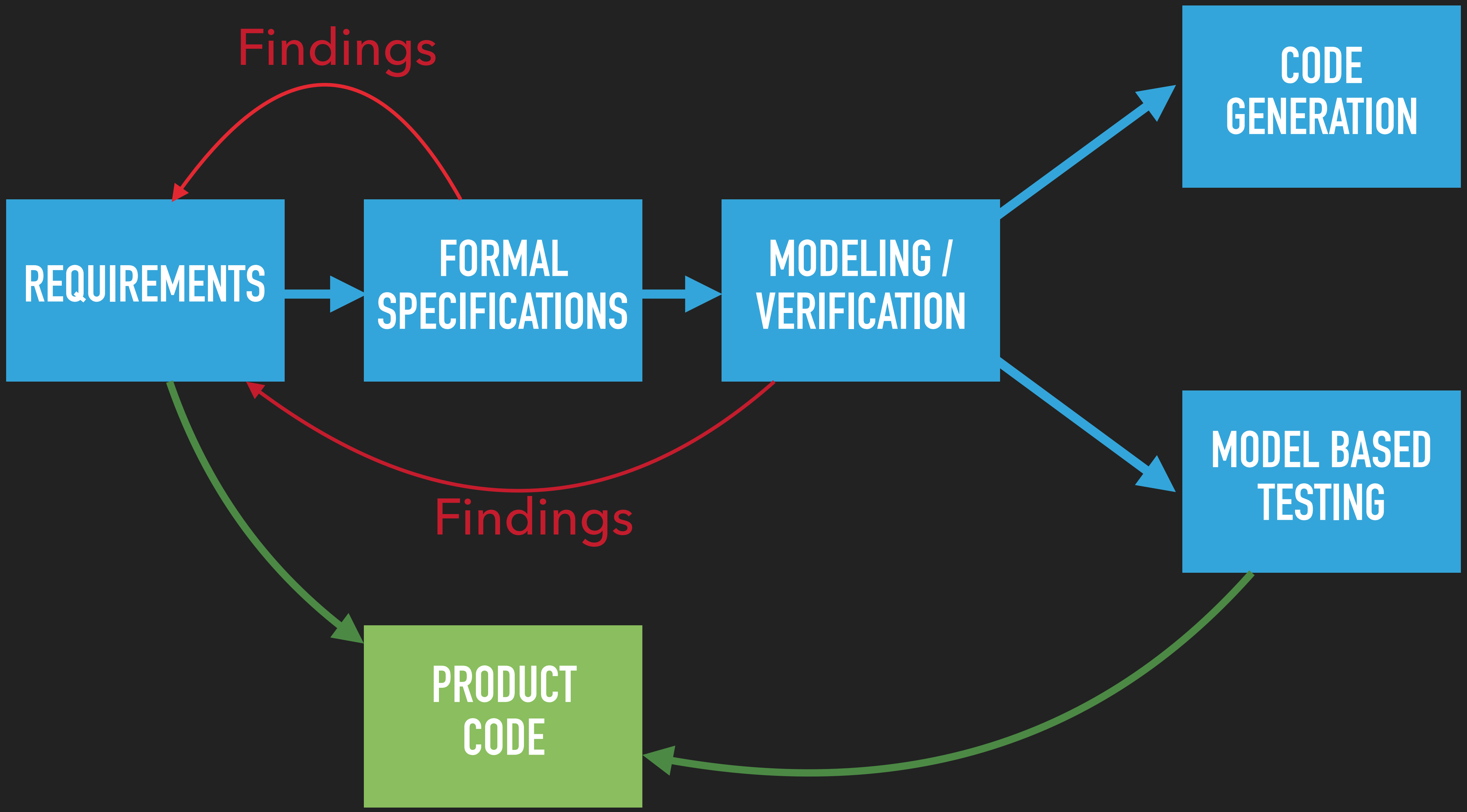
# LET'S DISCUSS

- **The Problem:** DLT Systems Complexity vs. Governance
- **The Solution:** Model Driven Development Approach
- **Garuda AI Platform:**
  - Algebraic modelling
  - History and Formalism
  - Consensus example
  - Mechanism design example
- **What's next?**

# THE PROBLEM: DLT SYSTEMS COMPLEXITY VS. GOVERNANCE

- What has to be checked before going live:
  - Consistence and completeness of the specifications
  - Safety property (nothing bad will happen)
  - Liveness property (something good will happen)
  - Security properties
  - Trends, metrics and thresholds

# THE SOLUTION: MODEL DRIVEN DEVELOPMENT APPROACH



# HISTORY AND FORMALISM

| | |
|---|---|
| 2019 | DLT Verification |
| 2010 | Model-based Testing,<br>Revers Engineering,<br>Cyber Security |
| 2000 | Verification of the systems:<br>Telecommunication, Automotive,<br>Hardware spec. etc. |
| 1990 | Algebraic Programming System,<br>Insertion Modeling System |
| 1980 | Automatic theorem prover |

Prof. Alexandr Letichevsky

Dr. Oleksandr Letychevskyi

Dr Volodymyr Peschanenko

# ALGEBRAIC VERIFICATION VS. CONCRETE SIMULATION

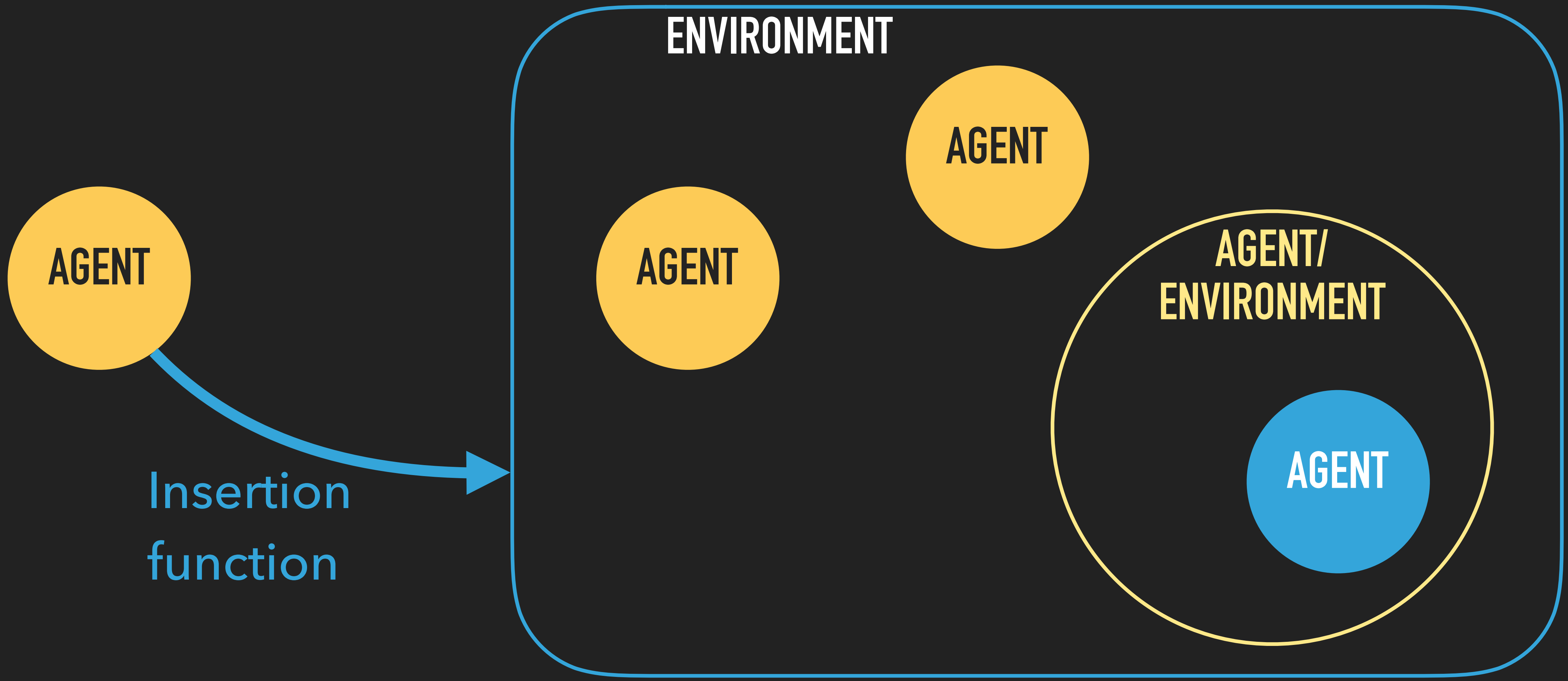
- System behaviour research
- Levels of abstraction and slices
- Proof of properties
- Algebraic behaviours matching
- Model based symbolic testing
- Code generation for the slices
- Garuda AI Platform do both approaches!

# FORMALISM

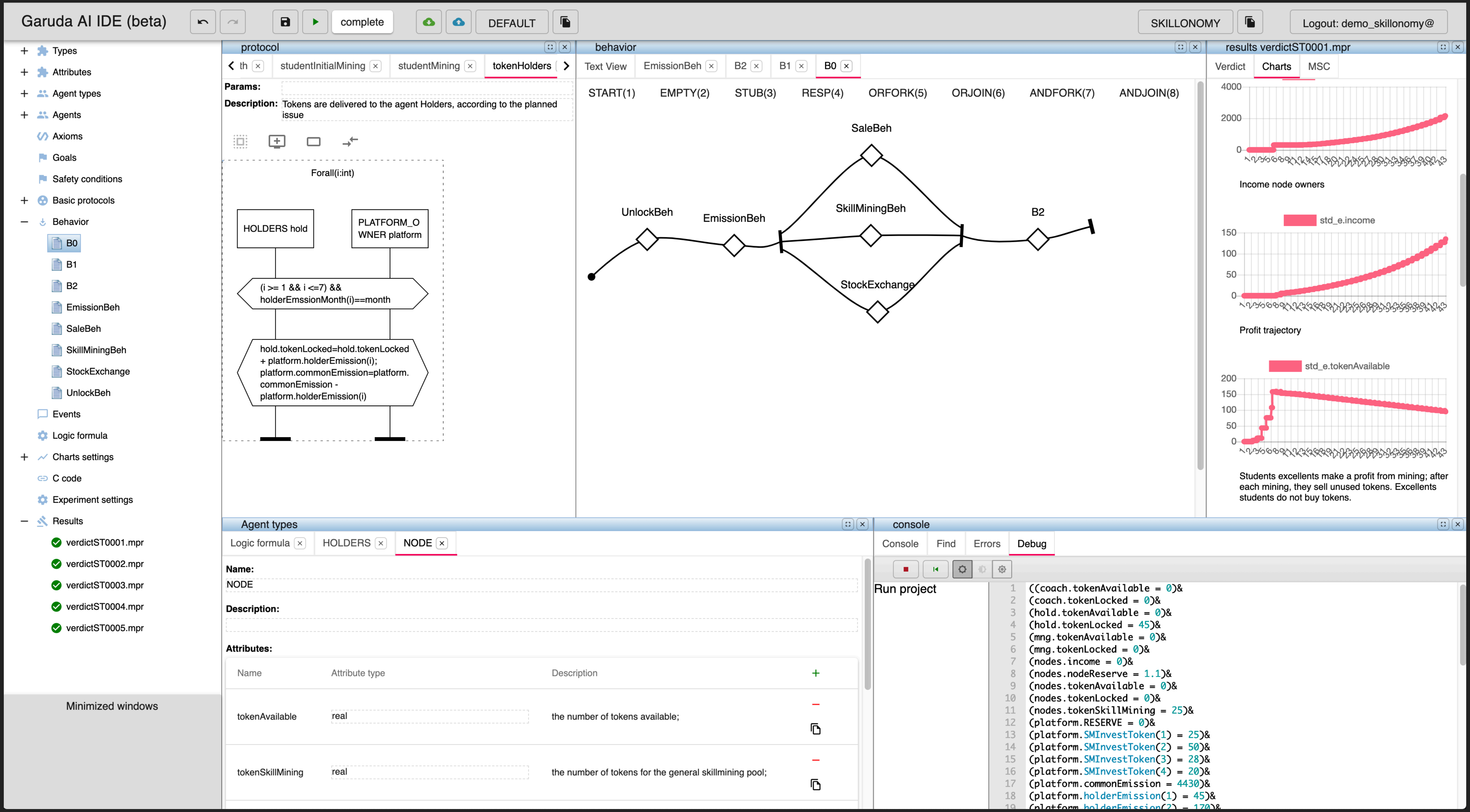
- The history of process algebra begins at the early seventies of the twentieth century.
- **Behaviour Algebra** was developed by Prof. D.Gilbert and Prof. A.Letichevsky in 1997
- Behaviour algebra is a two-sorted universal algebra
- The main sort is a **set of behaviours** and the second sort is a **set of actions**
- The algebra has two operations, three terminal constants, and a relation of approximation
  - The operations are the **prefixing** a.u (where a is an action, and u is a behaviour) and **non-deterministic choice** of behaviours u + v
  - The terminal constants are successful termination Δ, deadlock 0, and non-determinate behaviour ⊥
  - **Sequential** and **parallel** compositions of the behaviours

# FORMALISM

- On the top of Behaviour Algebra, we utilize Agents and Environment Theory and Insertion Modelling approach which fits well for the multi-agent distributed systems



# GARUDA AI PLATFORM OVERVIEW



**Params:**

**Description:** Tokens are delivered to the agent Holders, according to the planned issue

Forall(i:int)

HOLDERS hold

PLATFORM_OWNER platform

(i >= 1 && i <=7) && holderEmssionMonth(i)==month

```
hold.tokenLocked=hold.tokenLocked
+ platform.holderEmission(i);
platform.commonEmission=platform.
commonEmission -
platform.holderEmission(i)
```

Students excellents make a profit from mining; after each mining, they sell unused tokens. Excellents students do not buy tokens.

**Name:** NODE

**Description:**

**Attributes:**

| Name | Attribute type | Description |
|---|---|---|
| tokenAvailable | real | the number of tokens available; |
| tokenSkillMining | real | the number of tokens for the general skillmining pool; |

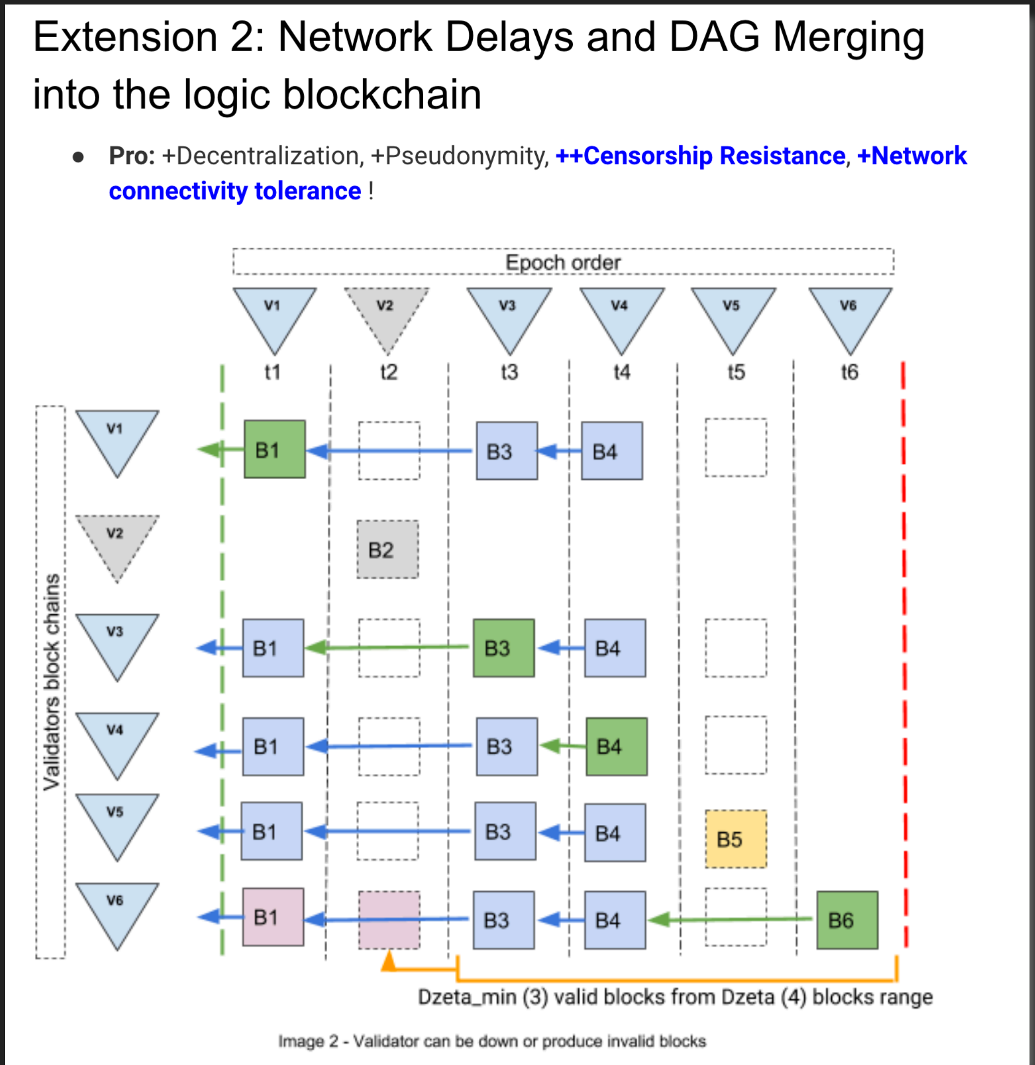
```
((coach.tokenAvailable = 0)&
(coach.tokenLocked = 0)&
(hold.tokenAvailable = 0)&
(hold.tokenLocked = 45)&
(mng.tokenAvailable = 0)&
(mng.tokenLocked = 0)&
(nodes.income = 0)&
(nodes.nodeReserve = 1.1)&
(nodes.tokenAvailable = 0)&
(nodes.tokenLocked = 0)&
(nodes.tokenSkillMining = 25)&
(platform.RESERVE = 0)&
(platform.SMInvestToken(1) = 25)&
(platform.SMInvestToken(2) = 50)&
(platform.SMInvestToken(3) = 28)&
(platform.SMInvestToken(4) = 20)&
(platform.commonEmission = 4430)&
(platform.holderEmission(1) = 45)&
(platform.holderEmission(2) = 170)&
```

# GARUDA AI PLATFORM MAIN FEATURES

- Imitation modelling: trend charts, metrics and thresholds
- Symbolic modelling: properties validation and proofs
  - Forward modelling
  - Backward modeling
  - Static symbolic verification
  - Symbolic tests generation
- Modelling Strategies
- Interactive modelling: Debug, Environment State
- UI tools: Charts, MSC, Blockchain formation

# CONSENSUS: PROMETHEUS POCW + POR

- Prometheus PoR consensus protocol verification results:
  - PoR formal specification:
    - 25 environment attributes
    - 34 basic protocols
  - Findings: approx. 15
  - Safety property violation: 2 times
  - In Progress: UI Tool



Extension 2: Network Delays and DAG Merging into the logic blockchain

- **Pro:** +Decentralization, +Pseudonymity, **++Censorship Resistance**, **+Network connectivity tolerance** !

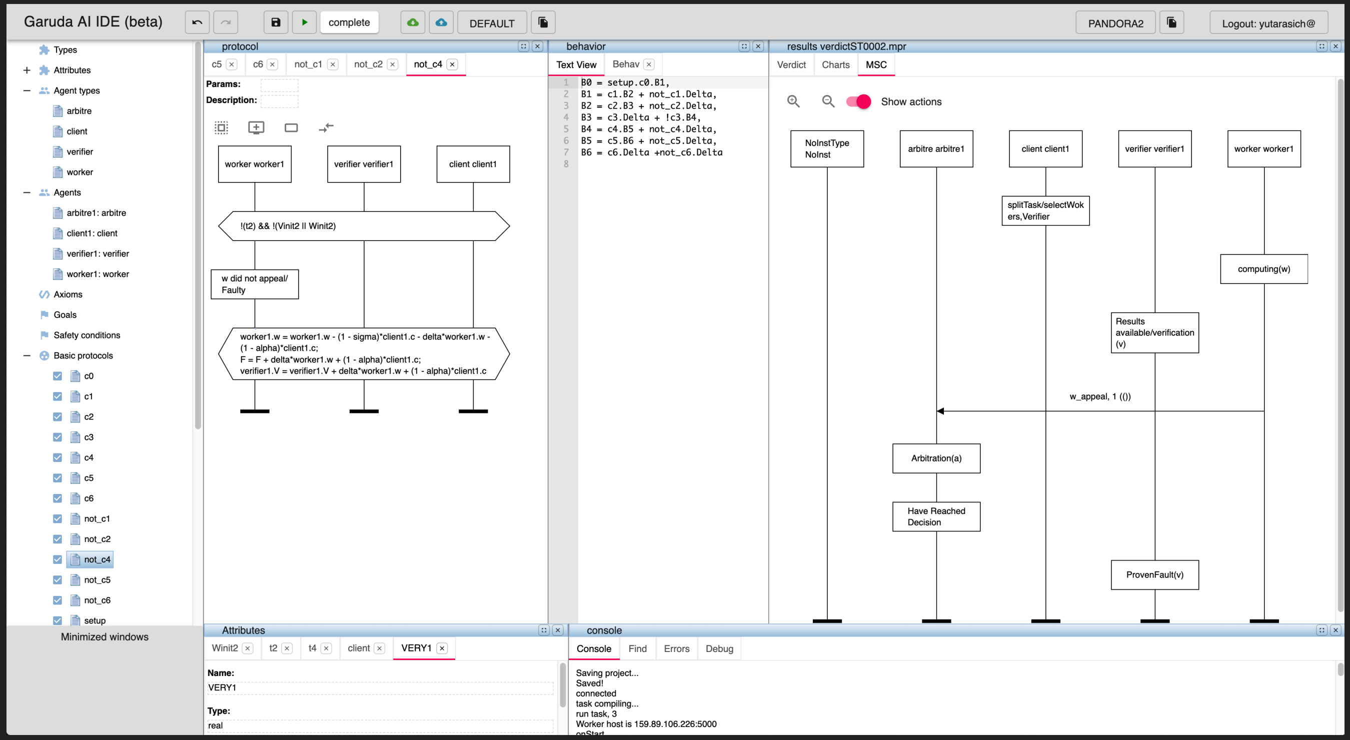
Image 2 - Validator can be down or produce invalid blocks

# CONSENSUS: PROMETHEUS POCW + POR

PANDORA
BOXCHAIN

PROMETHEUS
CONSENSUS

- Prometheus PoCW protocol verification results:
  - PoCW formal specification:
    - 26 environment attributes
    - 13 basic protocols
  - Verification is in progress



!(t2) && !(Vinit2 II Winit2)

worker1.w = worker1.w - (1 - sigma)*client1.c - delta*worker1.w - (1 - alpha)*client1.c;
F = F + delta*worker1.w + (1 - alpha)*client1.c;
verifier1.V = verifier1.V + delta*worker1.w + (1 - alpha)*client1.c

```
B0 = setup.c0.B1,
B1 = c1.B2 + not_c1.Delta,
B2 = c2.B3 + not_c2.Delta,
B3 = c3.Delta + !c3.B4,
B4 = c4.B5 + not_c4.Delta,
B5 = c5.B6 + not_c5.Delta,
B6 = c6.Delta +not_c6.Delta
```

Name:
VERY1

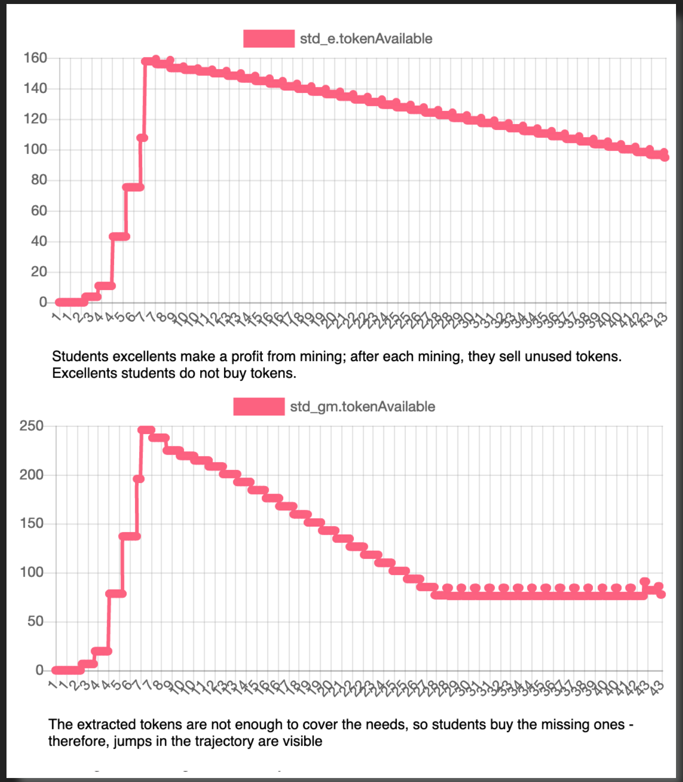
Type:
real

```
Saving project...
Saved!
connected
task compiling...
run task, 3
Worker host is 159.89.106.226:5000
onStart
```

# MECHANISM DESIGN: CRYPTO ECONOMICS

SKILLONOMY
TALENT MANAGEMENT
AND SKILL MONETIZATION
PLATFORM

- Imitation model of token distribution/sales strategy, internal game mechanics vs. token price vs. BTC price
  - Basic protocols: 32
  - Findings: approx. 20
  - Mechanics violation: 1
- Symbolic modeling
  - Forward: Reachability of critical tokens price lowering condition
  - Backward: Initial values to satisfy desired condition(token price)



Students excellents make a profit from mining; after each mining, they sell unused tokens. Excellents students do not buy tokens.

The extracted tokens are not enough to cover the needs, so students buy the missing ones - therefore, jumps in the trajectory are visible

# NEXT STEPS

- Site: garuda.ai
- Telegram channel: t.me/Garuda_AI_Platform
- Demo models: platform.garuda.ai
- Papers and conferences: see on the site
- User manual and how to: see on the site
- Test accounts(request directly)
- Plans: seminars and workshops