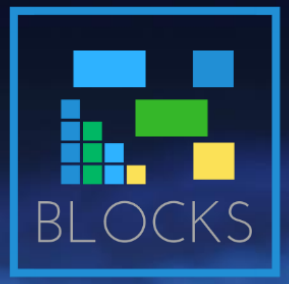


BLOCKCHAIN FOR ENTREPRENEURS - A NON-TRADITIONAL INDUSTRY 4.0 CURRICULUM FOR HIGHER EDUCATION - ERASMUS PLUS –2018-1-RO01-KA203-049510

USECASE X.6

Marian-George CIUCĂ

OVIDIUS UNIVERSITY FROM CONSTANTA/ROMANIA

Internet of Things (IoT)

- Since 1999
- *Consists of networked objects that sense and gather data from their surroundings, which is then used to perform automated functions to aid human users (see 2)*
- Many applications: healthcare, supply chain, pollution, weather etc
- Increasing number of devices: more than 22 billion (2021)
- Big question: Where we can store the data?

BLOCKCHAIN TOPIC

- Since financial crisis of 2007, (re)starting discussions about crytocurrencies (Bitcoin)
- In a distributed ledger, the data is always stored in blocks and the trust is assured by the decentralization structure and the embedded security.
- For Internet of Things, the security and computational power can be issues in implementing blockchain technology

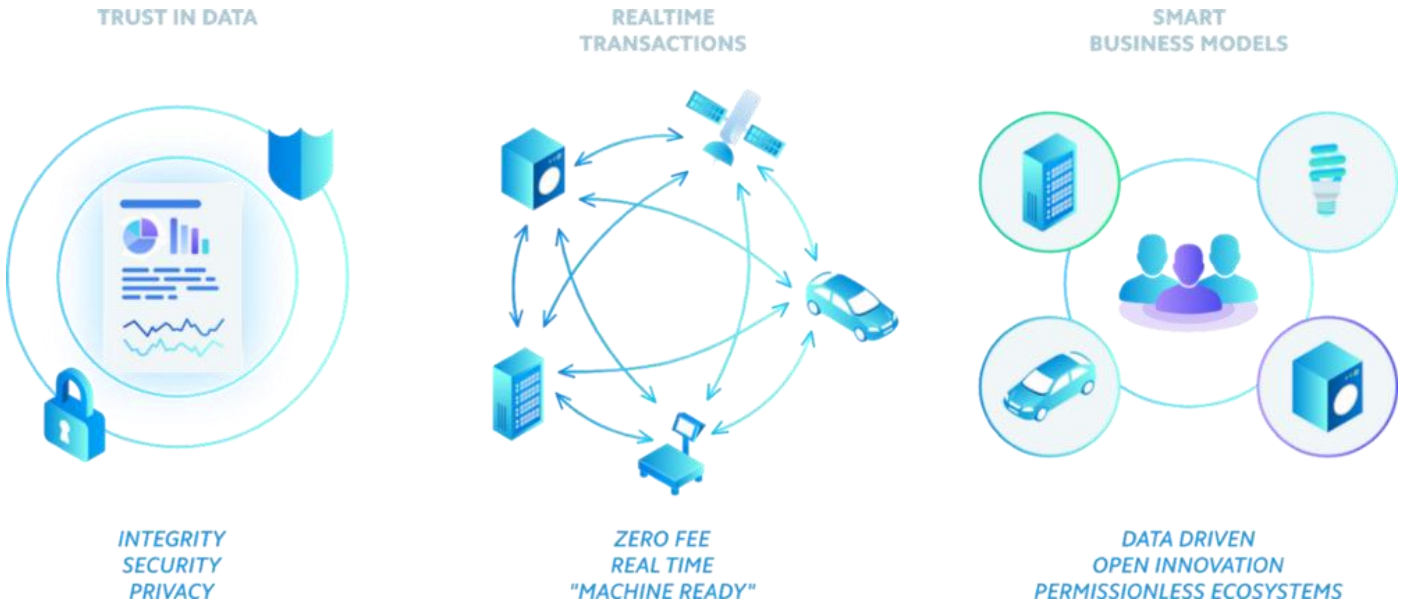
IOTA.ORG

IOTA – WHAT IS

- *Open source, feeless and scalable distributed ledger, designed to support frictionless data and value transfer*.
- As a distributed ledger technology, provides ***a trust layer*** for any devices that are connected to the ***global*** internet.
- Through its open network of nodes, allows you and your devices to:
  - Use the network as a source of truth for stored data
  - Transfer value in IOTA tokens
- IOTA networks are peer-to-peer networks where no central authority controls the data and all nodes hold a copy of it reaching a consensus on its contents.

IOTA – WHO IS FOR

- Anyone who:
  - does no trust centralized networks
  - wants to secure their data
  - values security
  - wants the freedom to transact



IOTA - STATUS QUO

- Manual peering
- Rate control mechanism
- Tip selection strategy
- Consensus
- IRI code optimization

IOTA – MANUAL PEERING

In order to join the Tangle, a node is required to connect to some existing nodes (peering). The current IRI software only permits manual peering, i.e., a node operator has to manually look for the addresses of other Tangle's nodes. Peering is fundamental to propagate transactions and to synchronize to the current status of the ledger. As for the latter, milestones are useful anchors to determine whether two nodes have fallen out of synchronization: If a node's latest solid milestone is much older than its peers', it is probably lagging behind.

IOTA – RATE CONTROL MECHANISM

In order to issue a transaction, a node must solve a cryptographic puzzle (Proof-of-Work). This is necessary to guarantee that nodes do not arbitrarily spam the network, or to avoid that they inject more transactions than the network can handle.

IOTA – TIP SELECTION STRATEGY

Approving transactions is a fundamental procedure which leads to the DAG structure of the Tangle. To approve a transaction, a node must verify that no inconsistencies with respect to the ledger state are introduced. Although it is not possible to enforce which transaction to validate, the original IOTA white paper suggests a tip selection algorithm based on a random walk which:

1. discourages lazy behavior and encourages approving fresh tips;
2. continuously merges small branches into a single large branch, thus increasing confirmation rate;
3. in case of conflicts, kills off all but one of the conflicting branches.

IOTA – CONSENSUS

The main role of milestones is to determine the consensus.

The Tangle applies a simple rule: A transaction is confirmed if and only if it is referenced by a milestone. In IRI, this is reflected in the getBalances and getInclusionStates API calls, which indicate how many tokens an account has and whether a transaction is confirmed, respectively.

IOTA – IRI CODE OPTIMIZATION

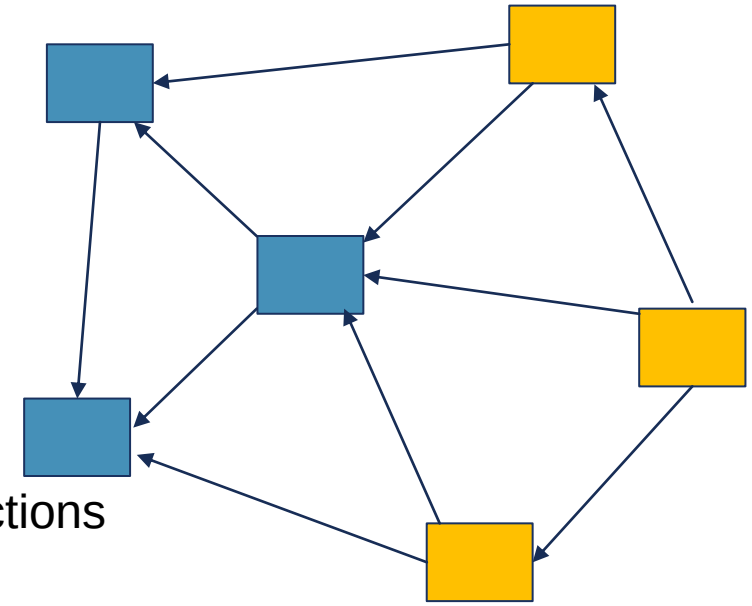
Instead of computing the full ledger state starting from the genesis, an intermediate state is saved for each milestone; similarly, milestones are used in local snapshots, i.e., the IRI pruning mechanism, which allows nodes to avoid storing older parts of the Tangle.

ARCHITECTURE OF IOTA

- **The Tangle**: a public ledger (based on a **Directed Acyclic Graph**) that is replicated across all nodes in an IOTA network. All data in the Tangle is stored in objects (*transactions*).
- **Nodes**: interconnected devices that are responsible for ensuring the integrity of the Tangle
- **Clients**: users of an IOTA network who send transactions to nodes to be attached to the Tangle.

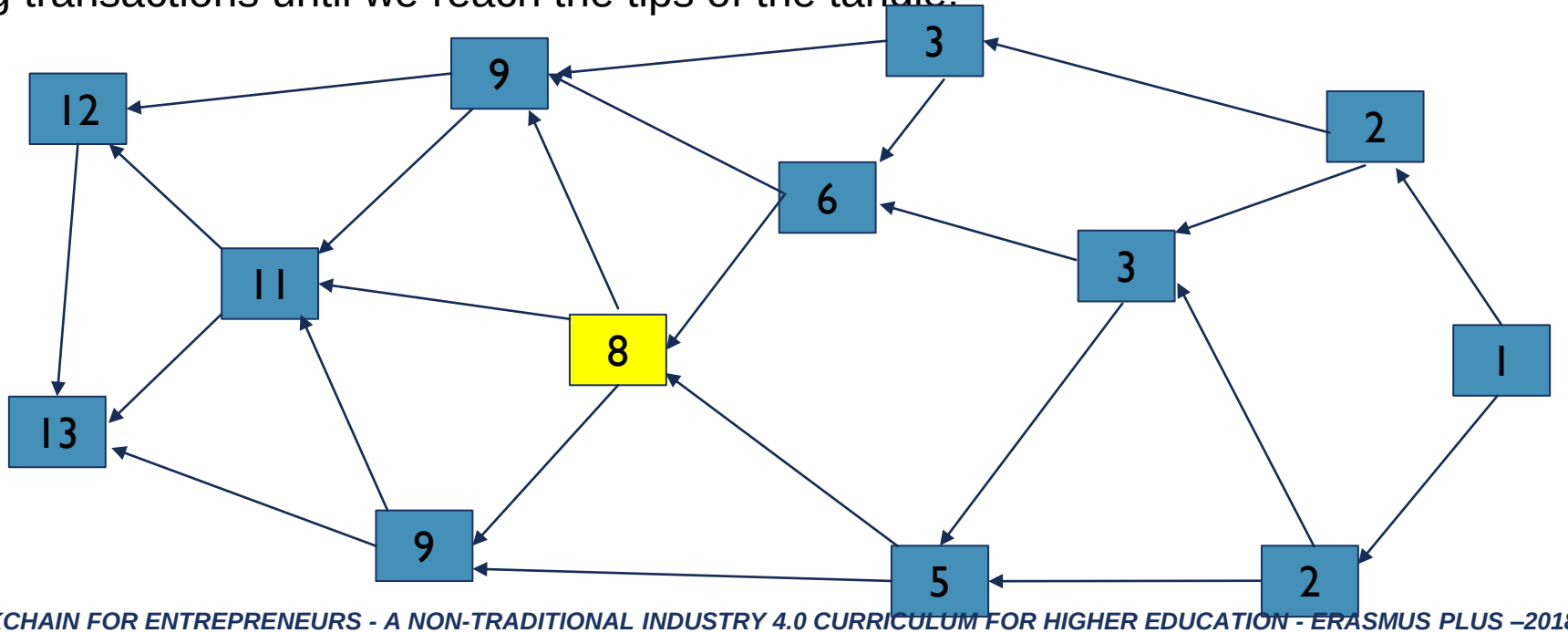
THE TANGLE

- Nodes (vertices): transactions/sites contain
  - Sender
  - Receiver
  - Transferred coins
  - Connection to at least two of other transactions
- Edges: validation/confirmation of transactions
- Tips of the tangle: nodes with less than two incomming transactions



THE TRUST OF A NODE

- Is computed using the cumulative weight of the node
- Every node has a weight of 1
- **Cumulative weight (trust level of the transaction)** of the node is the sum of the weights of the incomming transactions until we reach the tips of the tangle.



DEVELOPER ENVIRONMENT - SETUP

- IOTA supports many programming languages: *Javascript*, C, Go, Java, Python etc.
- Using npm (Node Package Manager from nodejs.org) you can install the IOTA core client libraries: `@iota/core @iota/converter`
- https://legacy.docs.iota.org/docs/getting-started/1.1/first-steps/set-up-env

DEVELOPER ENVIRONMENT - SETUP

- IOTA supports many programming languages: *Javascript*, C, Go, Java, Python etc.
- Using npm (Node Package Manager from nodejs.org) you can install the IOTA core client libraries: `@iota/core @iota/converter`
- https://legacy.docs.iota.org/docs/getting-started/1.1/first-steps/set-up-env

BALANCED TERNARY SYSTEM

Perhaps the prettiest number system of all, is the balanced ternary notation.

Donald E. Knuth, The Art of Computer Programming

- Binary system (bits): {0,1}, where often 0 stands for Off and 1 stands for On;
- Hexadecimal system: {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}, where A stands for decimal 10, so on;
- Ternary system: {0,1,2};
- **Balanced ternary system (trits)**: {-1,0,1} or {-,0,+}, where -1 stands for negative way, +1 stands for positive way and 0 stands for neutral position.

BALANCED TERNARY SYSTEM

- IOTA uses tryte-encoding
- A tryte is composed from three trits, viewed from left to right: 0, -1, 1 is, for example, the decimal 6;
- There are 3^3=27 trytes, from -13 to 13 (as decimal values)
- All tryte-characters are {9, A, B, C, …, Z}, indexed from 0 to 26, where the tryte 9 is the decimal 0, A is 1, B is 2, …, M is 13, N is -13, O is -12, … and Z is -1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

BALANCED TERNARY SYSTEM

- Encoding data: “Hello World!”
- Each ASCII character is represented as a pair of tryte-characters
- Because the decimal value of H (in the ASCII table) is 72, the first tryte-character is identified by the index 72%27=18 (R) and the second tryte-character is identified by the index (72-18)/27=2 (B). Therefore, RB is the coresponding tryte pair for H.
- Thus, the string “Hello World!” is encoded as “RBTC9D9DCDEAFCCDFD9DSCFA” (see 3).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

IOTA – DEMO (I)

```
const Iota = require('@iota/core');
const Converter = require('@iota/converter');

// Connect to a node
const iota = Iota.composeAPI({
  provider: 'https://nodes.devnet.iota.org:443'
});

const depth = 3;
const minimumWeightMagnitude = 9;
```

IOTA – DEMO (II)

// Define a seed and an address.

// These do not need to belong to anyone or have IOTA tokens.

// They must only contain a maximum of 81 trytes or 90 trytes with a valid checksum

const address = 'HEQLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWORLDHELLOWOR99D';

const seed =

'PUEOTSEITFEVEWCWBTSIZM9NKRGJEIMXTULBACGFRQK9IMGICLBKW9TTEVSDQMGWKBXPVCBMMCXWMNPDX';

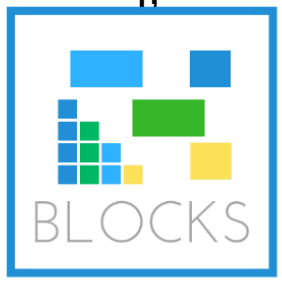
IOTA – DEMO (III)

```
// Define a message to send. This message must include only ASCII characters.
const message = JSON.stringify({"message": "Hello world!"});

// Convert the message to trytes
const messageInTrytes = Converter.asciiToTrytes(message);
```

IOTA – DEMO (IV)

```
// Define a zero-value transaction object that sends the message to the address
const transfers = [
  {
    value: 0,
    address: address,
    message: messageInTrytes
  }
];
```



IOTA – DEMO (V)

```
// Create a bundle from the `transfers` array and send the transaction to the node
iota.prepareTransfers(seed, transfers)
    .then(trytes => {
        return iota.sendTrytes(trytes, depth, minimumWeightMagnitude);
      })
     .then(bundle => {
        console.log(bundle[0].hash);
      })
     .catch(err => {
        console.error(err)
     });
```

IOTA – DEMO (VI)

- https://legacy.docs.iota.org/docs/getting-started/1.1/first-steps/hello-world
- https://explorer.iota.org/legacy-devnet

FUTURE READING

- Coordicide – the central coordinator of the Tangle
- Autopeering
- Mana system
- Fast probabilistic consensus
- IOTA 1.5 (see 5):
  - Chrysalis – the new Tangle
  - Hornet – a powerful, community driven IOTA node software
  - Bee - a framework for building IOTA nodes, clients, and applications in Rust
  - Stronghold - an open-source software library built to protect IOTA Seeds or any digital secret
  - Wallet

RESOURCES

1. https://iota.org
2. Muhammad Salek Ali, Massimo Vecchio, Miguel Pincheira, Koustabh Dolui, Fabio Antonelli, Mubashir Husian Rehnmani - *Applications of Blockchains in the IoT: A comprehensive survey*, IEEE, 2019
3. https://laurencetennant.com/iota-tools/
4. https://www.iota.org/foundation/research-papers
5. https://docs.iota.org/

THANK YOU!

https://blocks.ase.ro/summer-schools/

https://iota.org

mgciuca@365.univ-ovidius.ro

BLOCKS