agroString 2.0: A Distributed-Ledger based Smart Agriculture Framework to Ensure Transparency in Food Delivery

S. L. T. Vangipuram[1] , S. P. Mohanty[2] , and E. Kougianos[3]
University of North Texas, Denton, TX 76203, USA.[1,2,3]

Email: lt0264@unt.edu[1] , saraju.mohanty@unt.edu[2], and elias.kougianos@unt.edu[3].

Talk Outline

- Introduction.
- Reasons.
- Solutions.
- Motivation
- Related Works
- Why and what novelty in agroString 2.0.
- Architecture.
- Security in IOTA Tangle.
- Algorithm proposed.
- Implementation.
- Results.
- Conclusion with Future work.

Introduction

Transportation

Unprepared Storage

Food Waste in Harvest

Retail Wastage

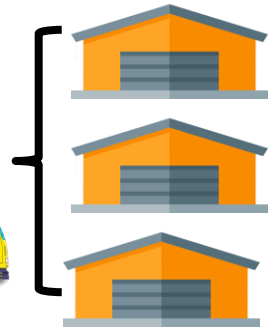
Reasons for Damaged Food Delivery

Raw Produce

Logistics

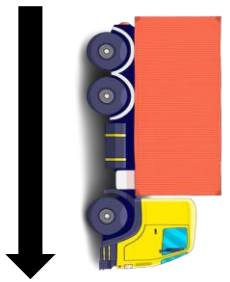
Supplier

Distributor



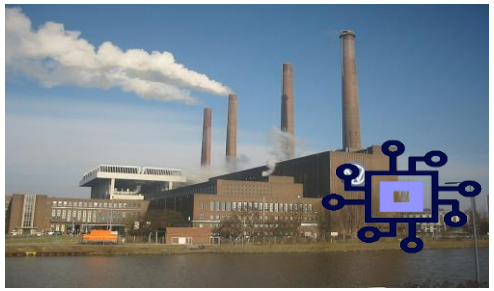
- Conventional/ Zero Tracking.
- Traditional Systems for Communication.
- Traditional Systems for Data Storage.

Demand



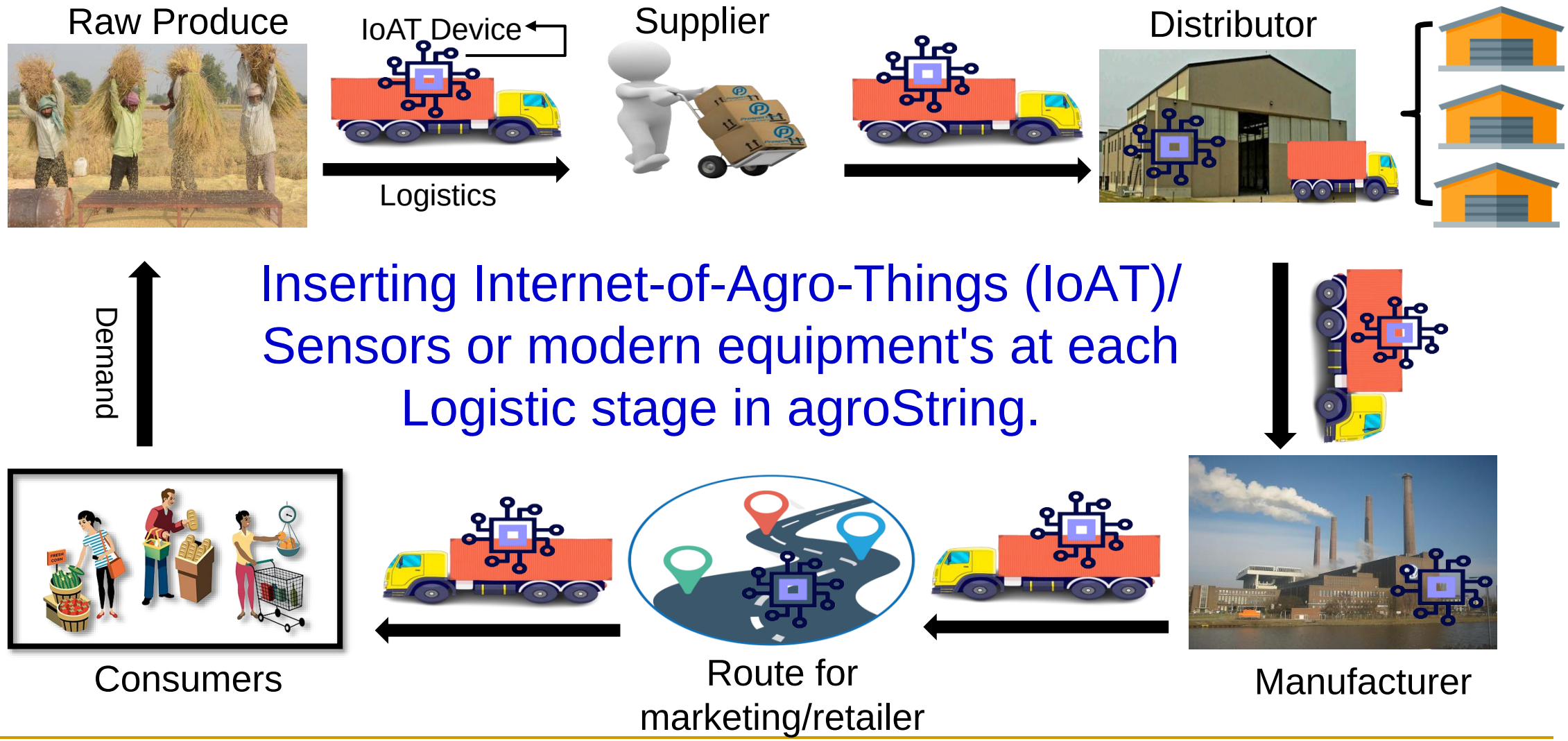
Consumers

Route for marketing/retailer

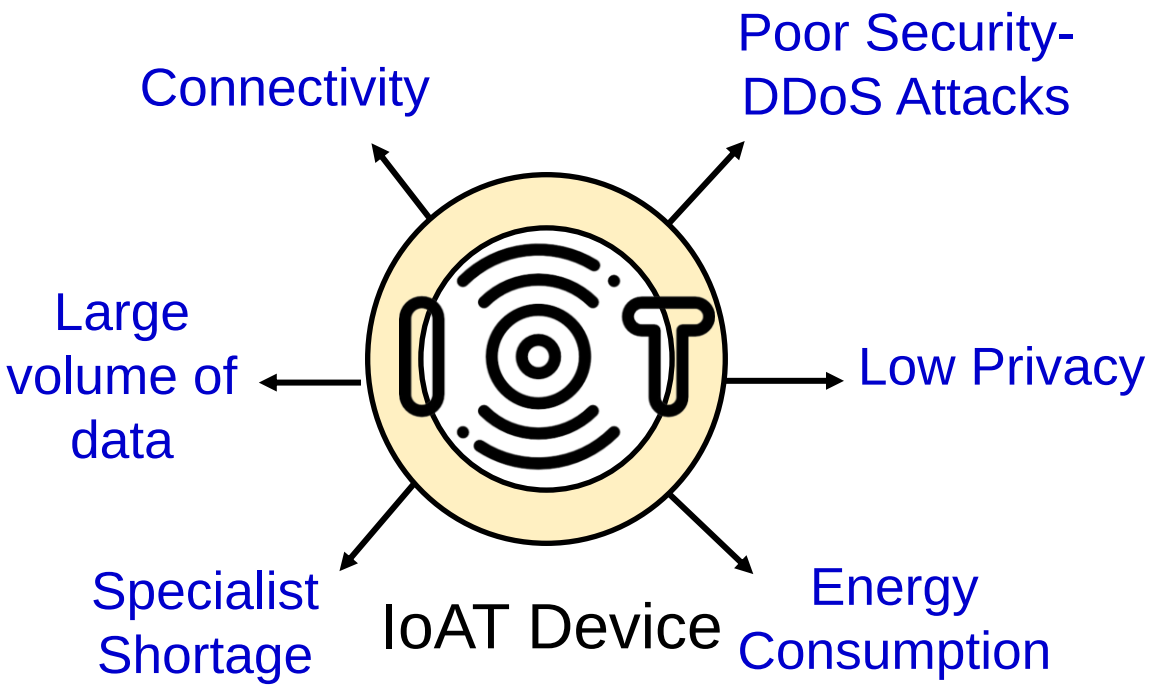


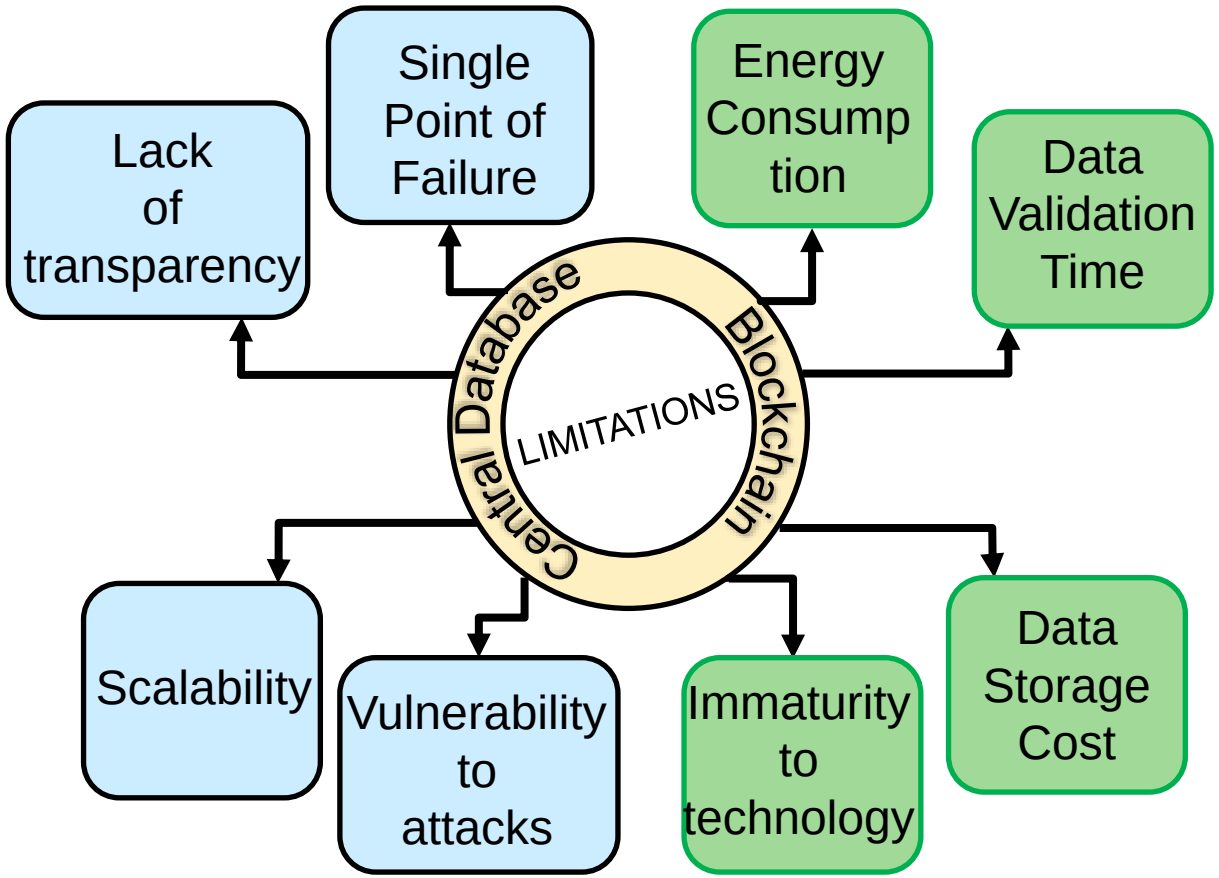
Manufacturer

Solutions with IoAT



Motivation for agroString





Related Works

S.No	Paper	Storage Technology	Security Level	Computation
1.	Traceability System - Zheng et al.[7]	Centralized	High Privacy Breach	Very High near Client
2.	Food Supply Chain - Mohammed and Chopra.[8]	Decentralized-Blockchain	High	Very High near Client
3.	Traceability System - Yang et al.[9]	Partially Centralized	High	High near Client
4.	agroString 1.0 – Vangipuram et al. [10]	Decentralized Corda	High	High near Client
5.	**agroString 2.0 [Current Paper]**	**Distributed Ledger-IOTA Tangle**	**High**	**Very Low**

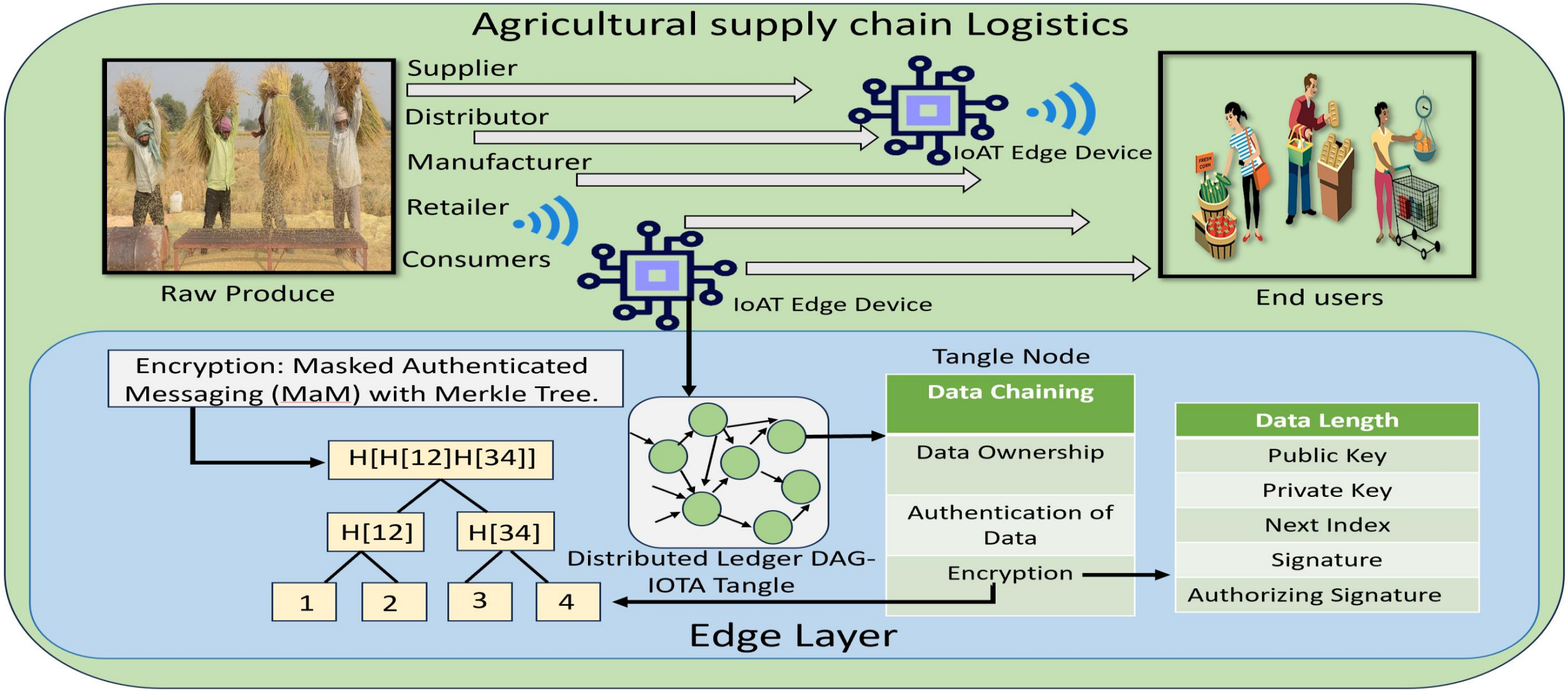
Why agroString 2.0?

- Distributed Ledger security for Internet-of-Agro-Things (IoAT) data in supply chain.
- Authenticity of data through Distributed Ledger System.
- Evading central storage in supply-chain.
- Proposing visibility and provenance design to end con- sumers in the food supply string.

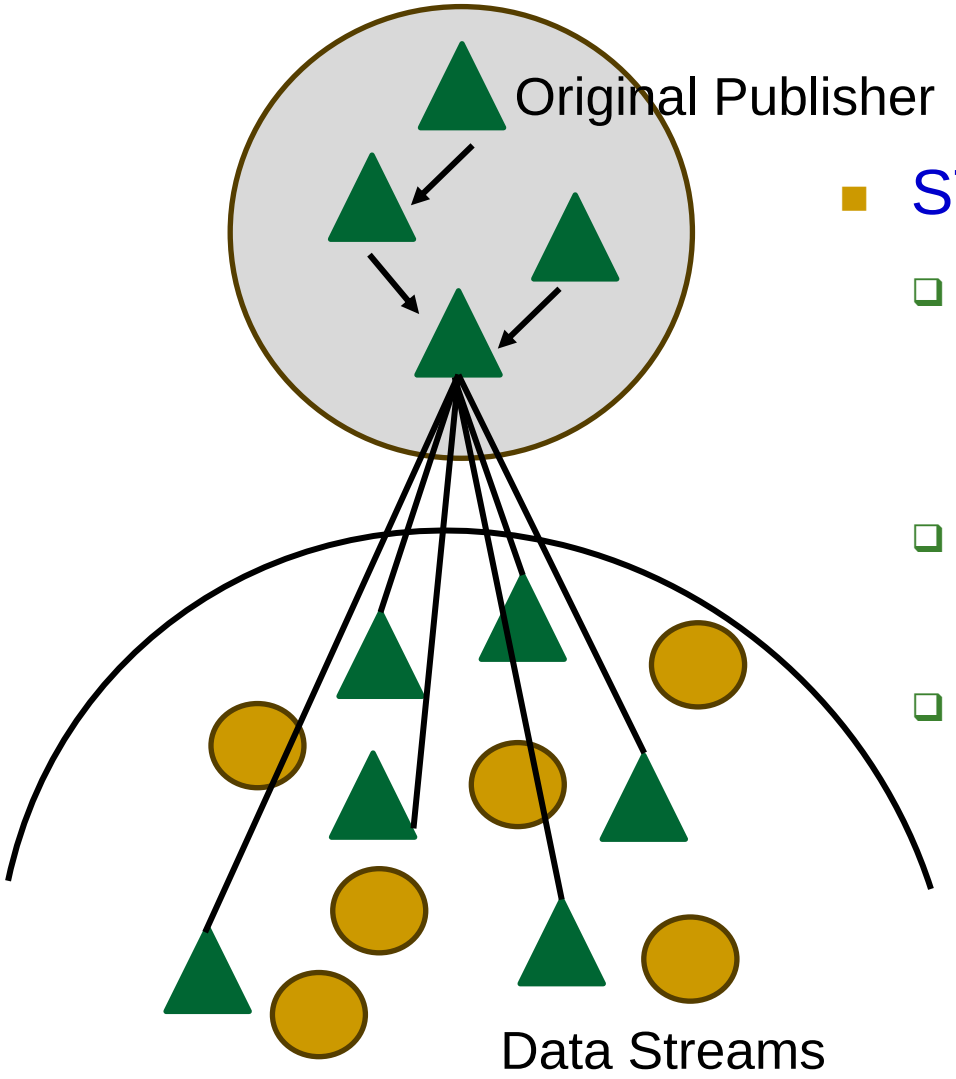
How Novelty in agroString 2.0

- Proposed a ledger storage architecture with IoAT Edge device to avoid central and cloud limitations.
- Implementing the IOTA Tangle with Masked Authenticated Messaging (MaM) for data integrity and validity.
- Showing results with IoAT Edge device and Distributed Ledger-IOTA Tangle with MaM.
- Comparing the results with Public and private distributed ledger systems to the current System.

Novel Architecture

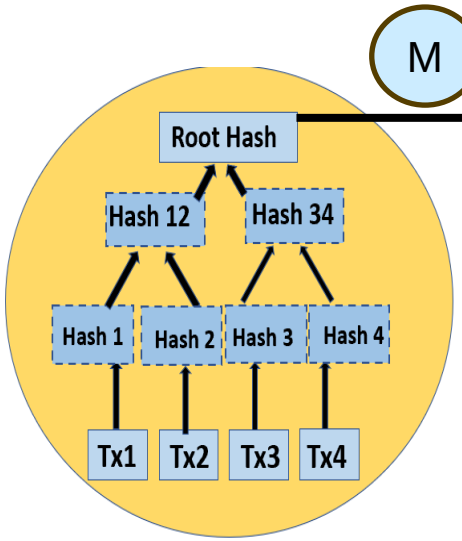


How Security in IOTA Tangle-STREAMS



- STREAMS
  - All branches of data streams reference a common root and state of data belonging to publisher for authenticity.
  - The Tangle data uses streams to always guarantee data integrity.
  - Streams enable users to control the ownership of data and receive payments.

How Security in IOTA Tangle-MaM



Markle Tree

Tree's root as the address of the transaction

Public Mode

Hash of the Tree's root as the address of the transaction

Private Mode

H M

Authorization Key with Hash of the Tree's root as the address of the transaction

Restricted Mode

+ H M

- Three Modes: Public, Private and Restricted.
- MAM fulfills an important need for integrity and privacy.
- MaM uses Merkle tree to hash the messages and give the root Message.
- A MAM publisher can decide to split the channel at any point in time, which means future messages use a new Merkle tree whose root has not been revealed before

Proposed Algorithm- IoAT Sensor Data File in IOTA Tangle.

1: $IoAT_{input} \rightarrow IoAT_{input}, IoAT_{input\text{-}len}, IoAT_{input\text{-}Pr}, IoAT_{input\text{-}Pu}, I_{value}, NI_{value}, S_i, S_{iauth}$.

2: $S_{random} \rightarrow S_{seed}$.

3: $S_{seed} \rightarrow IoAT_{input\text{-}Pr}, IoAT_{input\text{-}Pu}$.

4: $H(IoAT_{input\text{-}Pu}) \longrightarrow I$.

5: Another key pair is generated for the next input data ($IoAT_{nextinput}$) from another random source (S_{random}).

6: The key pairs from the next input data would be ($IoAT_{nextinput\text{-}Pr}$) and ($IoAT_{nextinput\text{-}Pu}$).

7: $H(IoAT_{nextinput\text{-}Pu}) \rightarrow NI_{value}$.

8: A digest D is calculated for signature.

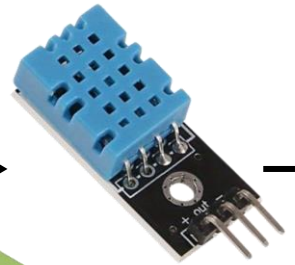
9: $D = H((IoAT_{input}) + (IoAT_{input\text{-}len}) + (IoAT_{input\text{-}Pu}) + (NI_{value}))$.

10: S_i = signature(D + $IoAT_{input\text{-}Pr}$)

Proposed Algorithms –IoAT Sensor Data File in IOTA Tangle.

11: **if** H($IoAT_{input}$)==Si+$IoAT_{input-Pu}$ **then**
12: successful verification.
13: **else**
14: end the process.
15: Si_{auth} = signature($IoAT_{Prkey}$)
16: **if** Si_{auth} == signature($IoAT_{Pukey}$) **then**
17: successful authentication.
18: else
19: end the process.
20: end if
21: end if
22: Repeat the steps from 1 through 21 in each logistic step of the supply chain

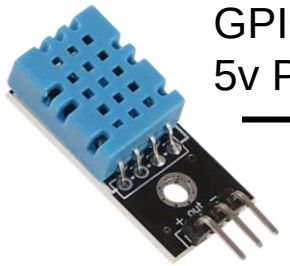
Technologies used for Implementation

- Raspberry Pi 3B+ Model →  → **Edge Layer**
- DHT 11 Sensor →  → **To Read Temperature & Humidity Data**
- NodeJS →
- Application Development →
- IOTA Tangle →

Implementation

Communication
BCM2835 Module

GND
GPIO4
5v Power



DHT11 Sensor

Four Javascript Files

Raspberry Pi 3B+
Model

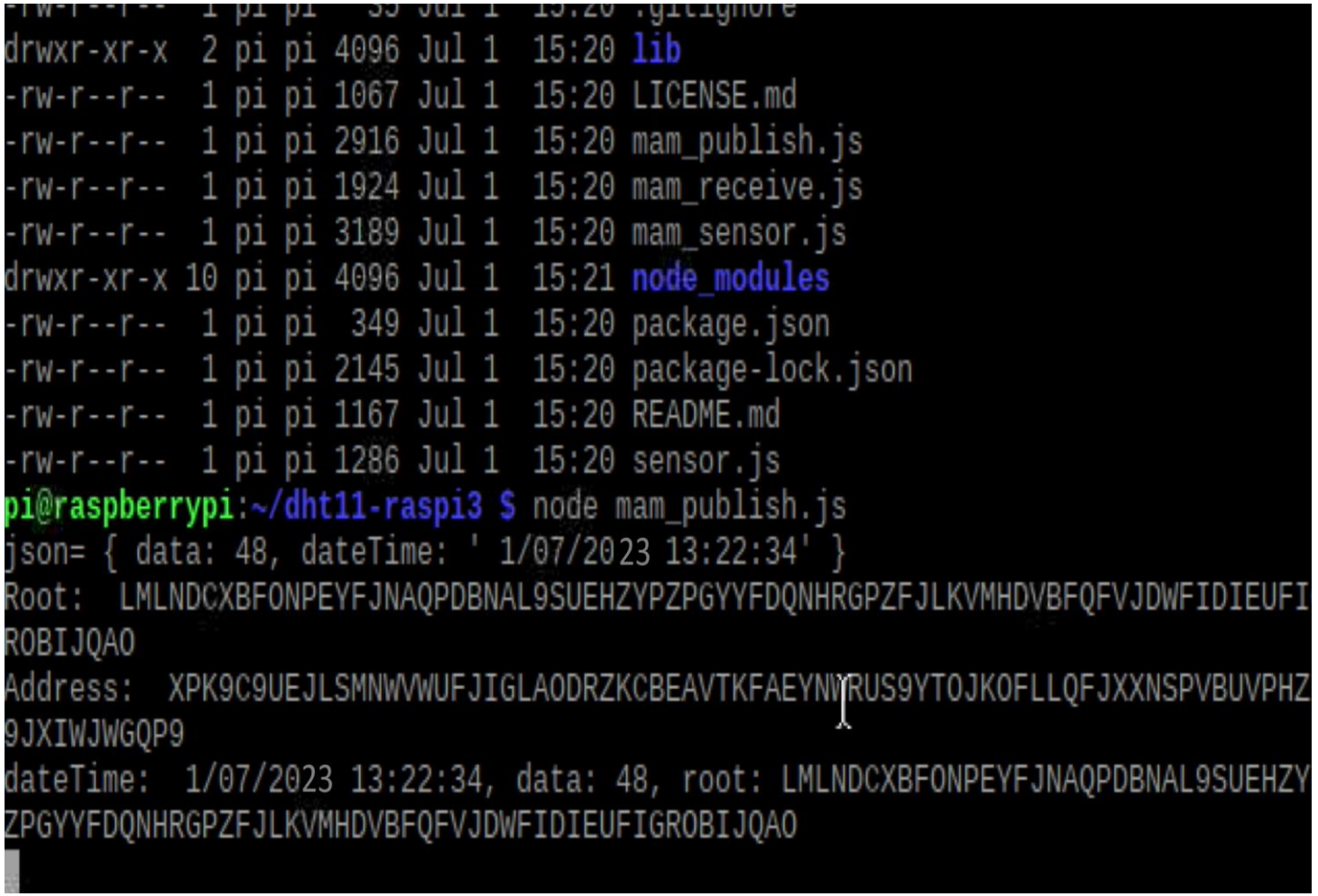
- mam_publish.js
- mam_recieve.js
- sensor.js
- mam_sensor.js
- Installing NodeJS

Backend
IOTA Tangle

Edge Layer

agroString Functional Verification- Publish Data

- Publishes randomly generated numbers to the backend IOTA Tangle using MaM.
- A root hash is given as input in mam-recieve.js to retrieve the sensor data using MaM.



Publishing Data

agroString Functional Verification- Retrieve Data

- Extract the Stored Data from the IOTA Tangle using MaM and displays the Data.
- The node modules of IOTA and http client packages are called to run the code

User Interface of agroString 2.0

```
drwxr-xr-x 22 pi pi 4096 Jul 1  15:20 ..
drwxr-xr-x  8 pi pi 4096 Jul 1  15:20 .git
-rw-r--r--  1 pi pi   35 Jul 1  15:20 .gitignore
drwxr-xr-x  2 pi pi 4096 Jul 1  15:20 lib
-rw-r--r--  1 pi pi 1067 Jul 1  15:20 LICENSE.md
-rw-r--r--  1 pi pi 2916 Jul 1  15:20 mam_publish.js
-rw-r--r--  1 pi pi 1924 Jul 1  15:20 mam_receive.js
-rw-r--r--  1 pi pi 3189 Jul 1  15:20 mam_sensor.js
drwxr-xr-x 10 pi pi 4096 Jul 1  15:21 node_modules
-rw-r--r--  1 pi pi  349 Jul 1  15:20 package.json
-rw-r--r--  1 pi pi 2145 Jul 1  15:20 package-lock.json
-rw-r--r--  1 pi pi 1167 Jul 1  15:20 README.md
-rw-r--r--  1 pi pi 1286 Jul 1  15:20 sensor.js
pi@raspberrypi:~/dht11-raspi3 $ node mam_receive.js LMLNDCXBFONPEYFJNAQPDBNAL9S
EHZYPZPGYYFDQNHRGPZFJLKVMHDVBFQFVJDWFIDIEUFIGROBIJQAO
dateTime:  1/07/2023 13:22:34, data: 48
dateTime:  1/07/2023 13:23:13, data: 20
dateTime:  1/07/2023 13:23:53, data: 63
dateTime:  1/07/2023 13:24:29, data: 60
```

Retrieving Data

Limitations & Challenges

- If the data size increases, there may be multiple issues in loading times and latency.
- The edge device used in the current paper is Raspberry Pi; leaving the power 'ON' all the time leads to the rising temperature of the board.
- The main encounter was the need for knowledge in developing the current Tangle application and the lack of support for DApps with the Tangle. The Tangle network has no such support for these decentralized applications.

Performance Results of CroPAiD

Comparing Previous works

Paper	Data Integrity	Authentication	Double-Spending	Energy Consumption	Cost
Zheng et al. [7]	No	Less	High	High	High
Mohammed and Chopra. [8]	Yes	2-factor Authentication	Less	Very High	Very High
Yang et al. [9]	Yes	Less	High	High	Less
Vangipuram et al. [10] (agroString)	Yes	High	No	High	Less
agroString 2.0 [Current-Paper]	Yes	High	No	Very Less	Zero

Performance Evaluation

Paper	Storage Technology	Edge	Time Taken	Latency
Zheng et al. [7]	Centralized	No	2.23s [15]	Very High
Mohammed and Chopra. [8]	Decentralized	No	8.72s [4]	Very High
Yang et al. [9]	Partially Centralized	No	10.2s [15], [4]	High
Vangipuram et al. [10] (agroString)	Decentralized	Yes-IoAT Data	1 ms [10]	Less
agroString 2.0 [Current-Paper]	Distributed Ledger-Tangle	Yes-IoAT Data	Zero	Less

Performance Calculated for Data Size of 279 Kb File.

Conclusion & Future Direction

- We successfully design an application with the IoAT edge device raspberry pi to sense the data from the DHT1.
- Uses Masked Authenticated Messaging.
- The application we design evades the limitations of the central and blockchain storage platforms and takes in real-time data from the sensor inside the edge layer.
- The system increases the security of sensor data and provides integrity by providing quality food data to end consumers.
- In the future, the current system can be elaborated to other domains for the secure flow of sensitive information.


THANK
YOU