GIK INSTITUTE
OF ENGINEERING SCIENCES AND TECHNOLOGY
30 YEARS OF EXCELLENCE

UNIVERSITY OF TARTU
1632
Institute of Computer Science

Practical Exercise:

Building Digital Twins with Microsoft Azure Digital Twin

Mubashar Iqbal

Lecturer in Information Security

Institute of Computer Science, University of Tartu, Estonia

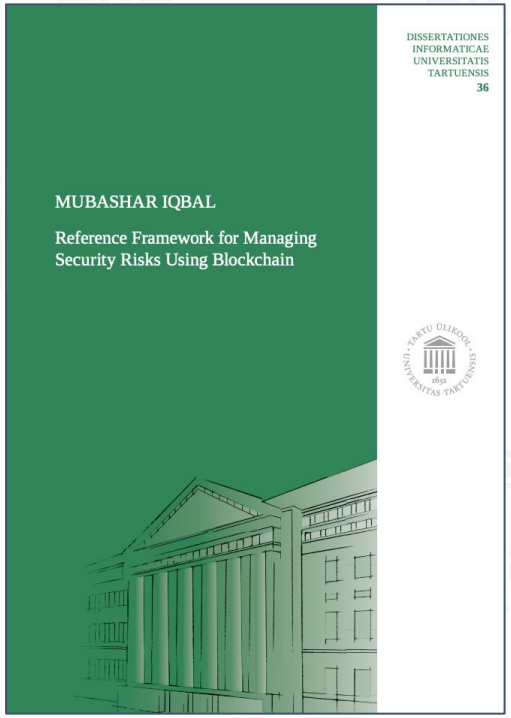
mubashar.iqbal@ut.ee

CHESS

About Me

https://www.linkedin.com/in/**mubashariqbalbajwa/**

https://www.researchgate.net/profile/**Mubashar-Iqbal-2**

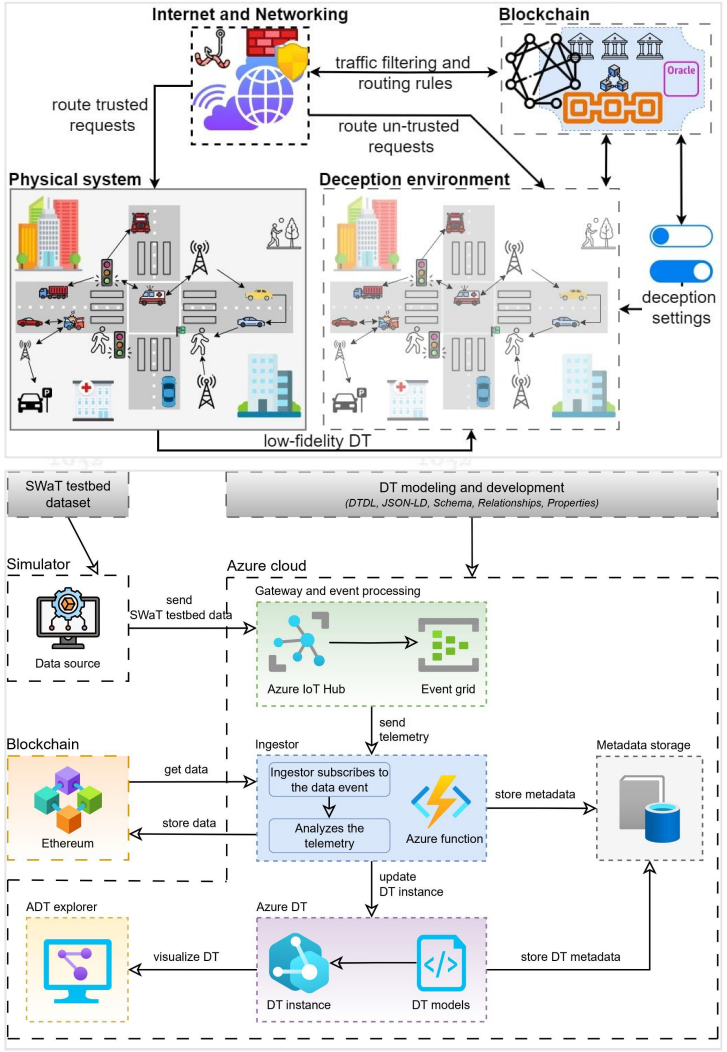


- Mubashar Iqbal
  - PhD in Computer Science
    - **Topic:** Reference Framework For Managing Security Risks Using Blockchain
      - https://dspace.ut.ee/handle/10062/83826
  - Lecturer in Information Security
  - **Research areas:** Blockchain, Digital Twins, Metaverse, Information Security, Ontology, AI for Cybersecurity
- Information Security Research Group
  - https://infosec.cs.ut.ee

Research on DT

(published research papers)

- ENIGMA: An explainable digital twin security solution for cyber–physical systems
- DECEPTWIN: Proactive security approach for IoV by leveraging deception-based digital twins and blockchain
- TRIPLE: A blockchain-based digital twin framework for cyber–physical systems security
- A framework for enhancing cyber incident response with security-enhancing digital twins in cyber–physical systems



CHESS: CYBER-SECURITY EXCELLENCE HUB IN ESTONIA AND SOUTH MORAVIA

CHESS
Cyber-security Excellence Hub in Estonia and South Moravia

- Estonia (as an advanced digital society) and South Moravia (as a Czech ICT powerhouse) are teaming up to support the Europe's safe transition to a digital society
- Developing a joint cross-border cybersecurity research and innovation strategy
  - Focusing on six Challenge Areas (CA):
    - **CA1:** Internet of Secure Things
    - **CA2:** Security Certification
    - **CA3:** Verification of Trustworthy Software
    - **CA4: Security Preservation in Blockchain**
    - **CA5:** Post-Quantum Cryptography
    - **CA6:** Human-centric Aspects of Cybersecurity

- *Updates: https://chess-eu.cs.ut.ee*

UNIVERSITY OF TARTU · UNIVERSITAS TARTUENSIS · 1632

Funded by the European Union

Today's talk content

- Microsoft Azure Digital Twin (DT)
- Components of Azure DT
- DT Definition Language
- DTDL-based DT models
- Azure DT Explorer
- Practical Exercise

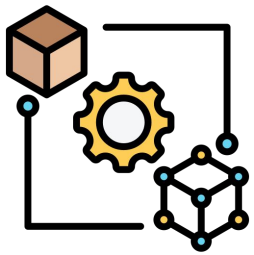
Microsoft Azure

- Cloud Computing Platform developed by Microsoft
- Offers Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)
- Hosts virtual machines, databases, DevOps Tools, IoT Service, etc.

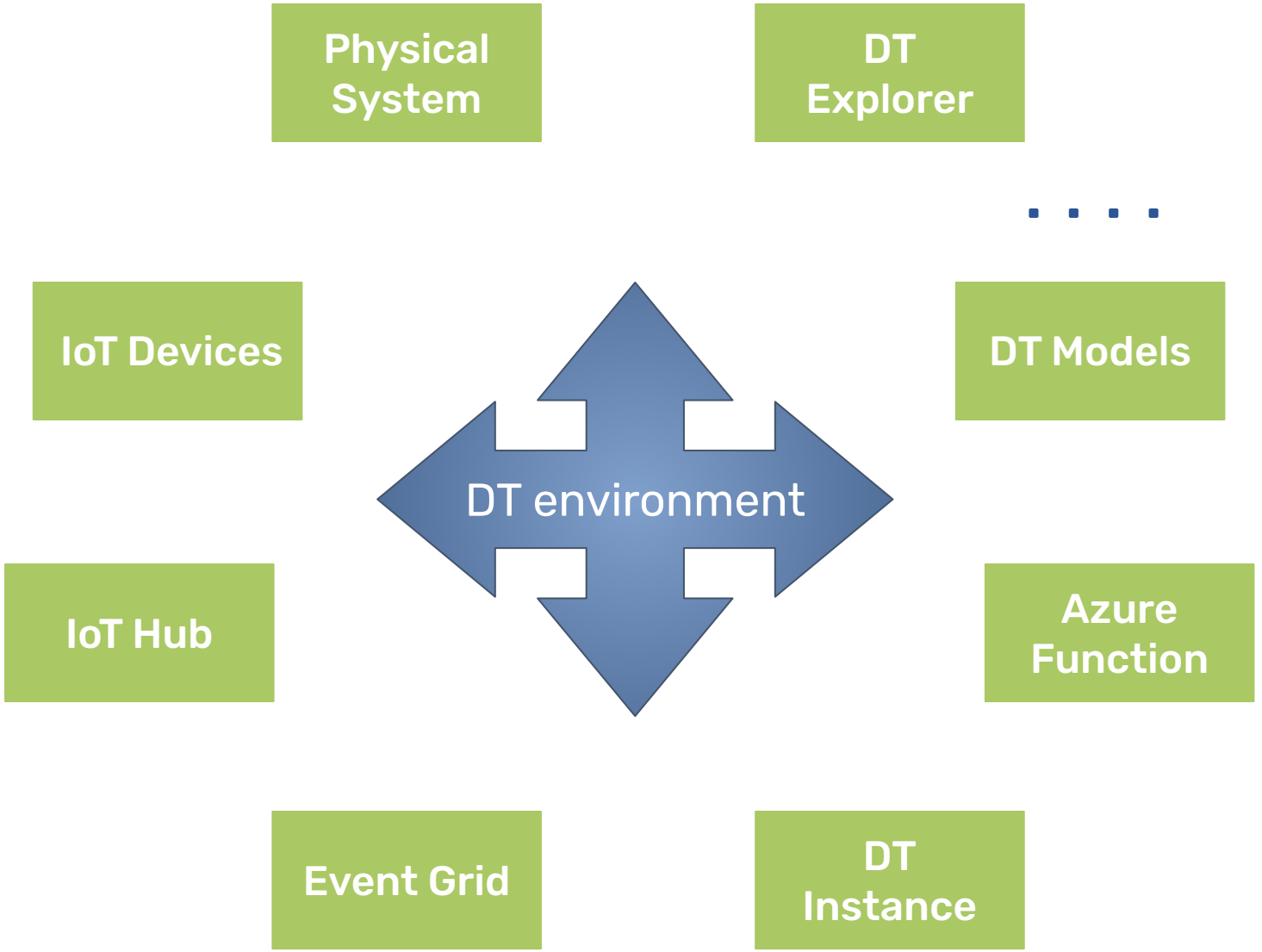
Microsoft Azure Digital Twin

- Create digital replicas of physical environments
- Enables real-time data syncing from IoT devices
- Built on open modeling language
  - Digital Twins Definition Language (DTDL)
- Integrates with Azure **IoT Hub, Event Grid, Functions**
- Supports event-driven architecture using Azure Event Grid

Enables what-if analysis, predictive maintenance, and performance optimization, and so on

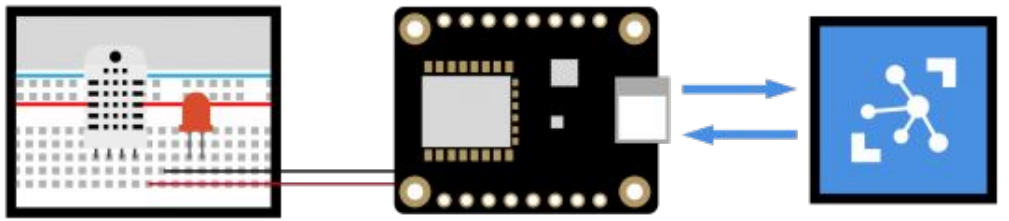


Components of Azure DT

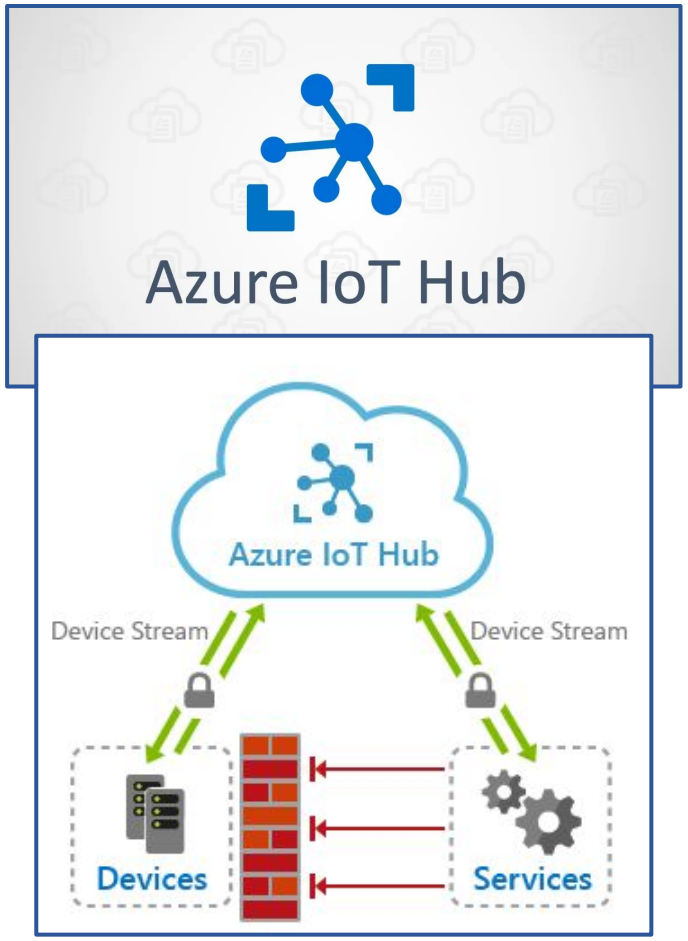


IoT Devices

- Physical hardware
  - Captures real-world data
  - Connect via Azure IoT Hub
- Send telemetry data to the DT
- Enables real-time monitoring, condition-based alerts, and automated actions



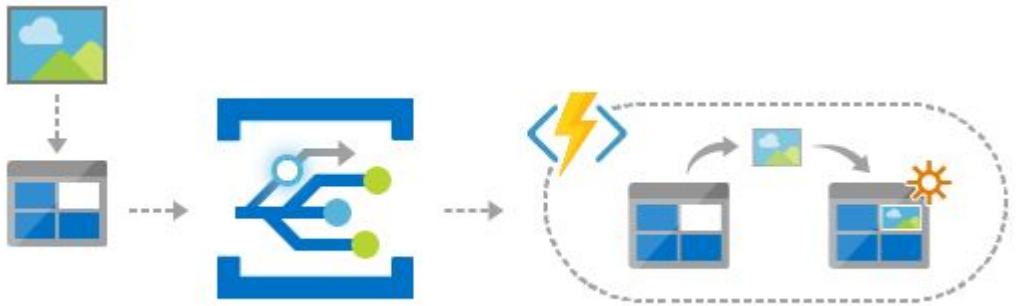
Azure IoT Hub



- Manage bi-directional communication between IoT devices and Azure services
  - e.g., device-to-cloud and cloud-to-device messaging
- Includes device identity, authentication, and status monitoring
- Acts as the bridge between real-world devices and digital models
- Ingests real-time telemetry
- Forwards data to Azure DTs via event routing
  - e.g., Event Grid

Event Grid

- Event routing service
  - Delivers real-time events to various Azure services
  - Messaging backbone for DT event flows
- Routes events from:
  - Sensor data via Azure IoT Hub
  - External systems (e.g., logic apps, functions)



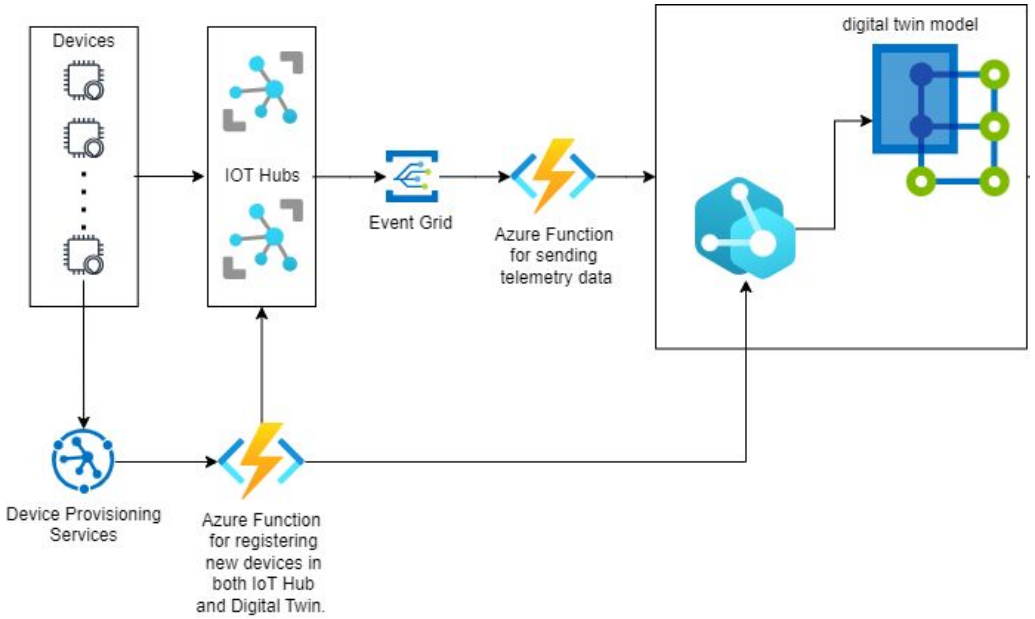
Azure DT Instance

- Azure DTs resource in Azure subscription
- Runtime space where
  - Models are created using DTDL
  - DTs are instantiated and their relationships are defined
- Manages real-time state of each twin

Azure DT instance URL: *https://gikidigitaltwin.api.eus.digitaltwins.azure.net/*
(Only accessible via ADT explorer)

Azure Function

- Serverless compute service
  - Trigger-based execution
  - Automates responses to changes or events in the DT
- Triggered via Azure Event Grid events
  - e.g., updates DT when receive telemetry data from the Azure IoT Hub



DT Definition Language

- JSON-based modeling language
  - Defines DT models in Azure DTs
  - Describes the structure, behavior, and relationships of physical entities
- Standardized for interoperability across IoT ecosystems

DTDL-based DT Models

- DT models are semantic definitions that describe the structure, behavior, and relationships of real-world entities in Azure DTs
  - Schema/blueprint for twins of their physical counterpart
- Mandatory properties *(red box)*
  - Main definition block of the DT model
- Optional properties *(yellow box)*
  - Defines real-time data streams from devices
  - Defines relationships

DTDL-based DT Models:

Mandatory properties

- @context
- @id
- @type
  - Unique identifiers for semantic consistency
- Display name

```
{
    "@context": "dtmi:dtdl:context;2",
    "@id": "dtmi:vehicle:speedometer_sensor;1",
    "@type": "Interface",
    "displayName": "Speedometer Sensor Interface Model",
    "contents": [
        {
            "@type": "Property",
            "name": "Id",
            "schema": "string",
            "description": "Speedometer Sensor Id",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "SensorState",
            "schema": "string",
            "description": "Funcational and Non-Functional State of Speedometer Sensor",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "CanBusPayload",
            "description": "Speedometer Sensor Can Bus Payload",
            "schema": {
                "@id": "dtmi:vehicle:speedometer_sensor:payload;1",
                "@type": "Object",
                "fields": [
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:timestamp;1",
                        "name": "TS",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:id;1",
                        "name": "ID_MF",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:bytes;1",
                        "name": "CAN_PL_DATA_BYTES",
                        "description": "CAN_PL_0 is the first byte, and CAN_PL_7 is the last byte of the payload from CAN bus",
                        "schema": "string"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:timestamp_difference_mf;1",
                        "name": "TD_TS_MF",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:timestamp_difference_id;1",
                        "name": "TD_ID_MF",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:dlc;1",
                        "name": "DLC",
                        "schema": "integer"
                    }
                ]
            }
        },
        {
            "@type": "Property",
            "name": "Classification",
            "schema": "boolean",
            "description": "The classification tells about the request is Attack or Non-Attack",
            "writable": true
        }
    ]
}
```

DTDL-based DT Models:

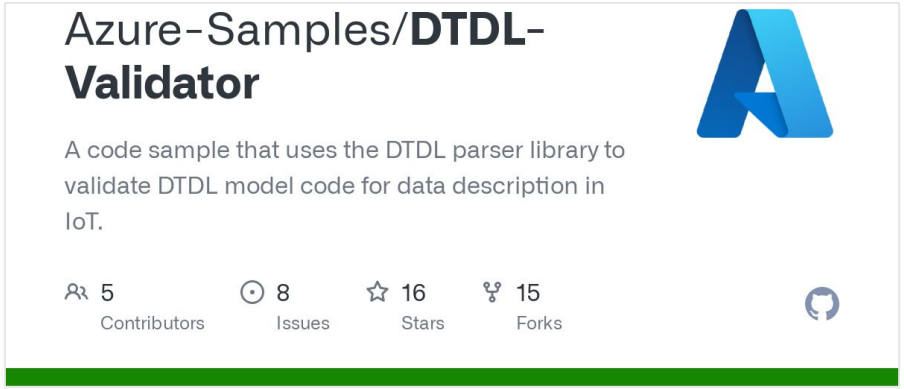
Optional properties

- Contents
- Properties
  - Store data/state (e.g., temperature = 24°C)
- Name
- Schema
- Relationships
  - Links between twin models (e.g., room contains sensor)

```
{
    "@context": "dtmi:dtdl:context;2",
    "@id": "dtmi:vehicle:speedometer_sensor;1",
    "@type": "Interface",
    "displayName": "Speedometer Sensor Interface Model",
    "contents": [
        {
            "@type": "Property",
            "name": "Id",
            "schema": "string",
            "description": "Speedometer Sensor Id",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "SensorState",
            "schema": "string",
            "description": "Funcational and Non-Functional State of Speedometer Sensor",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "CanBusPayload",
            "description": "Speedometer Sensor Can Bus Payload",
            "schema": {
                "@id": "dtmi:vehicle:speedometer_sensor:payload;1",
                "@type": "Object",
                "fields": [
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:timestamp;1",
                        "name": "TS",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:id;1",
                        "name": "ID_MF",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:bytes;1",
                        "name": "CAN_PL_DATA_BYTES",
                        "description": "CAN_PL_0 is the first byte, and CAN_PL_7 is the last byte of the payload from CAN bus",
                        "schema": "string"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:timestamp_difference_mf;1",
                        "name": "TD_TS_MF",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:timestamp_difference_id;1",
                        "name": "TD_ID_MF",
                        "schema": "integer"
                    },
                    {
                        "@id": "dtmi:vehicle:speedometer_sensor:payload:dlc;1",
                        "name": "DLC",
                        "schema": "integer"
                    }
                ]
            }
        },
        {
            "@type": "Property",
            "name": "Classification",
            "schema": "boolean",
            "description": "The classification tells about the request is Attack or Non-Attack",
            "writable": true
        }
    ]
}
```

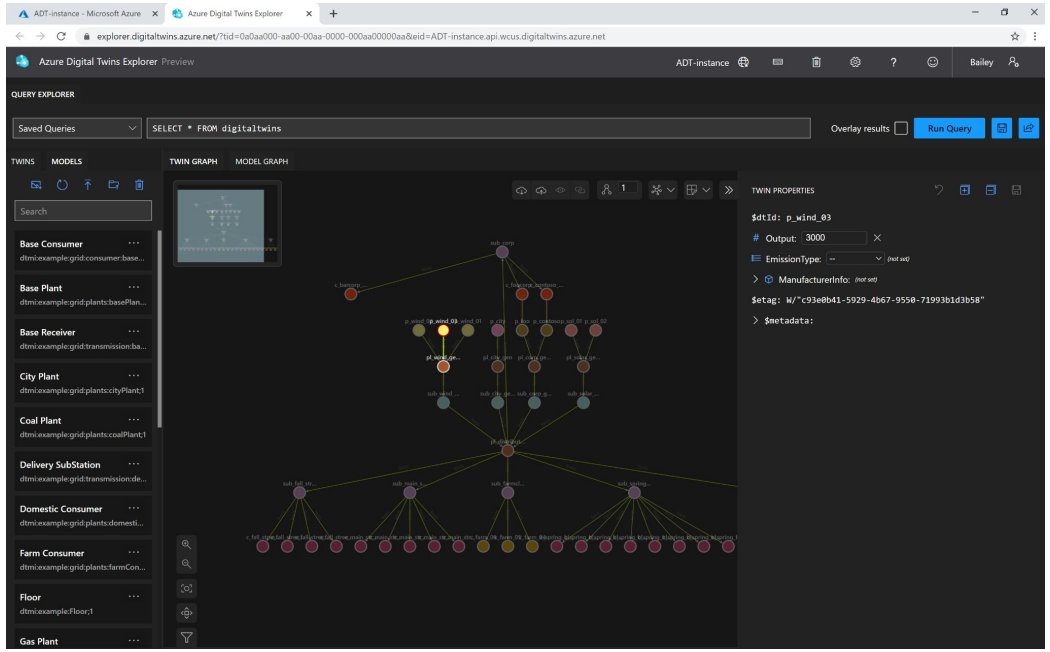
DTDL Models Validator

- Checks models are syntactically and semantically correct
- Checks for:
  - JSON syntax errors
  - Schema compliance with DTDL versions (v2.0+)
  - Duplicate IDs, circular relationships, invalid data types
- Supports bulk validation for DT model libraries



Azure DT Explorer

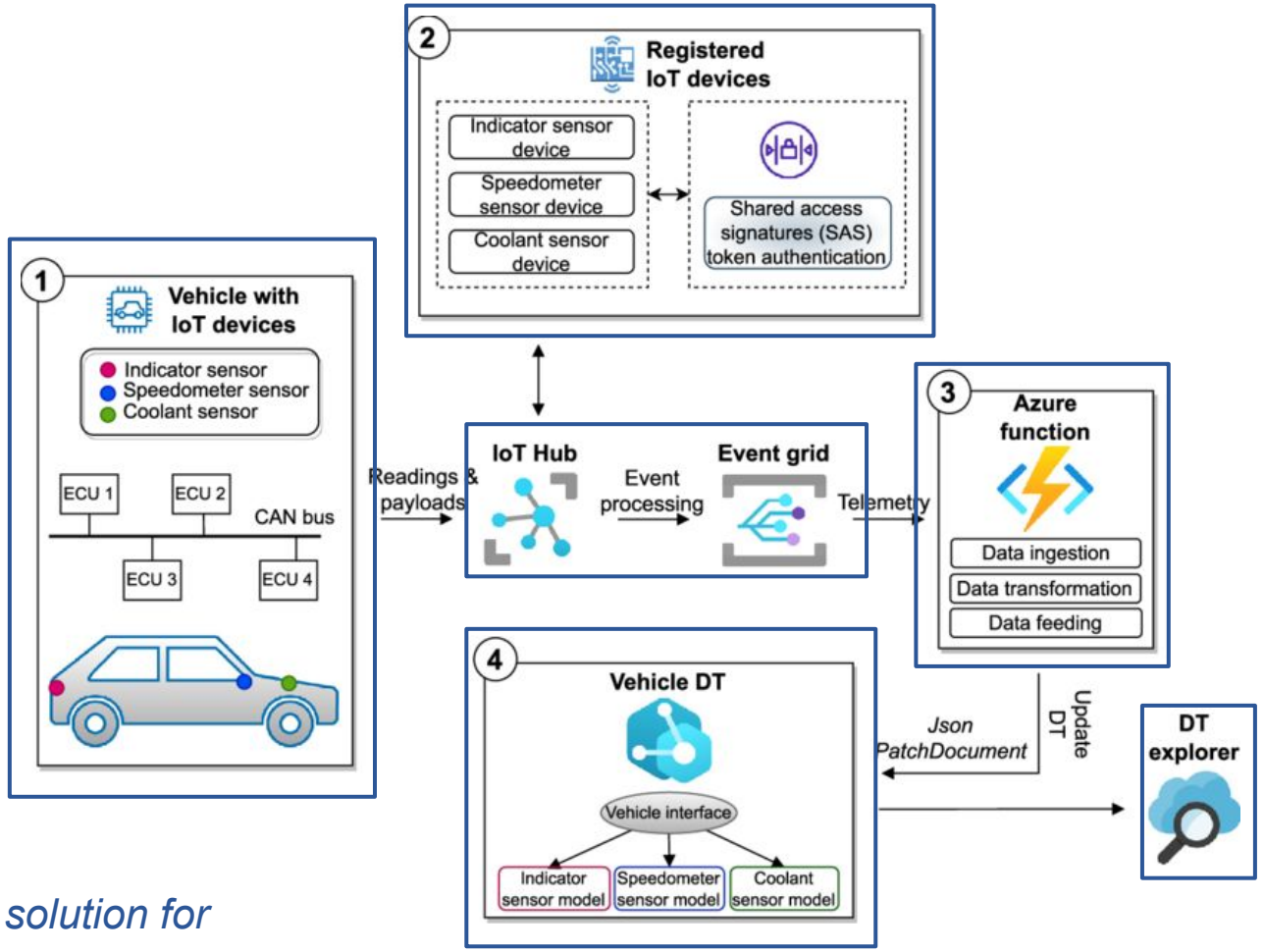
- Web-based interface
  - Explore, visualize, and interact with Azure DTs models
- Displays live data from IoT devices mapped to the DT
- Allows users to view and manage DTDL models
- Supports running queries to filter and analyze DT data



Architecture of Azure DT:

Instantiation

- Physical system
- IoT Devices
- IoT Hub
- Event Grid
- Azure Function
- DT Instance and Models
- DT Explorer



ENIGMA: An explainable digital twin security solution for cyber–physical systems

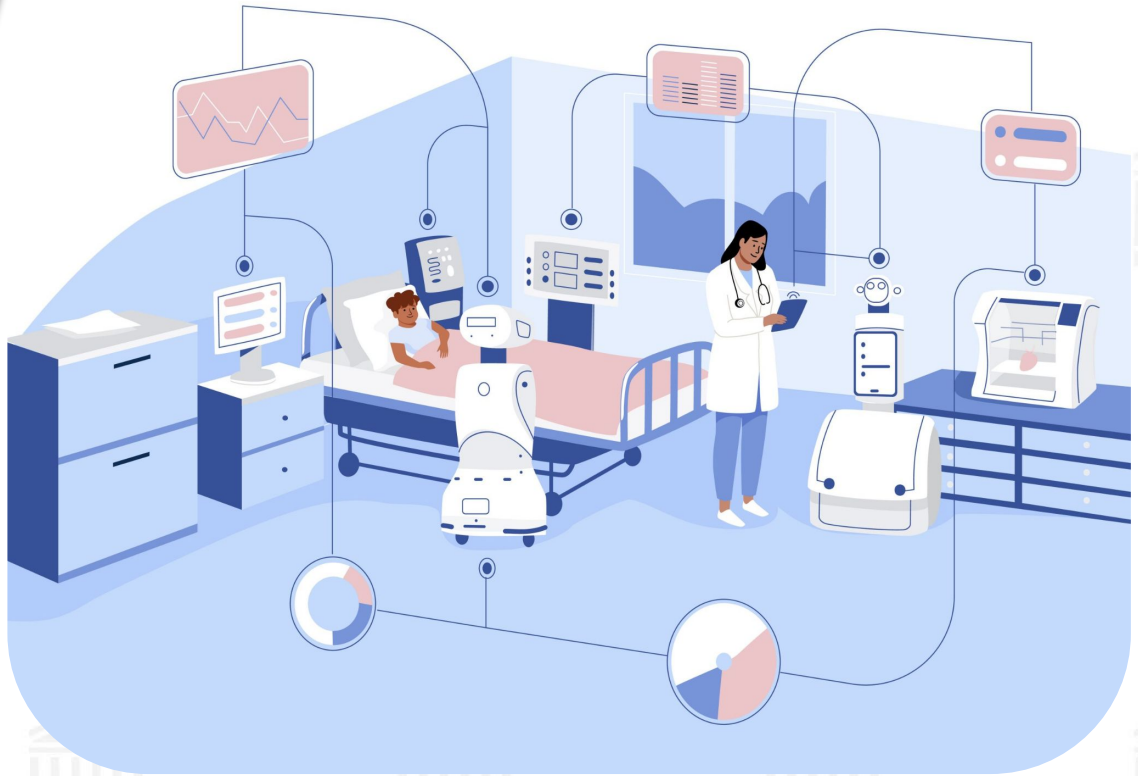
https://doi.org/10.1016/j.compind.2023.103961

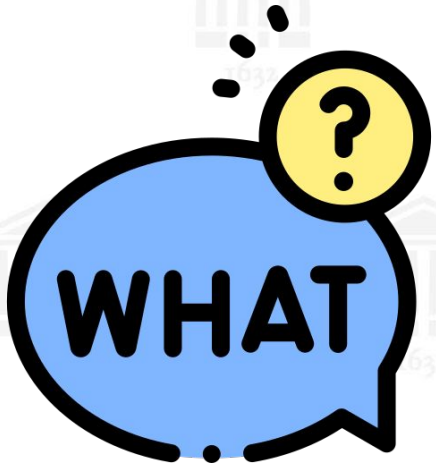
Case Study:

Smart patient room DT

Objectives:

- Real time monitoring of room conditions
- Maintain optimal room conditions for patient recovery





Temperature

Air Quality

Humidity

Lightning

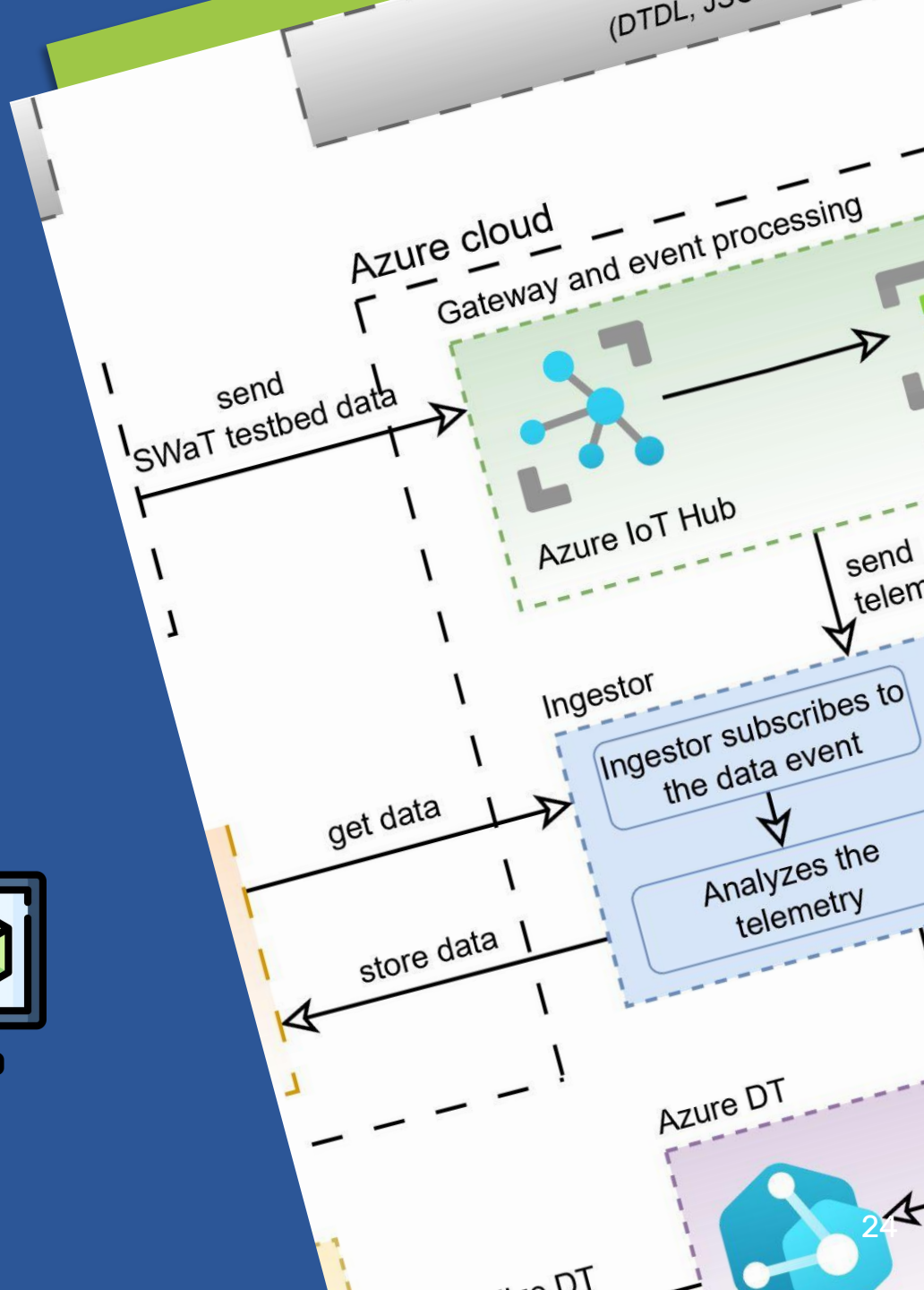
Environment setup

- Create Azure free subscription account
  - *https://azure.microsoft.com/*
- Install Python3
  - *https://www.python.org/downloads/*
- Install Azure-CLI
  - *https://learn.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest*
- Install Azure Identity Module
  - *pip install azure-identity*
- Install Azure IoT explorer *(optional, only available for windows)*
  - *https://learn.microsoft.com/en-us/azure/iot/howto-use-iot-explorer*

2 minutes

Practical Exercise

- Create IoT Hub
  - Add IoT devices
  - Create IoT Devices Simulator
- Create Azure Function
- Create Azure DT Instance
- Create Event Grid
  - Assign to Azure Function
- Create DT Models
  - Use DTDL
  - Validate models
- Deploy Models using DT Explorer
- Create Twins and their Relationships
- Run DT
- Query DT



Thank You!

Mubashar Iqbal
Lecturer of Information Security
mubashar.iqbal@ut.ee
https://infosec.cs.ut.ee

CHESS

GIK INSTITUTE
OF ENGINEERING SCIENCES AND TECHNOLOGY
30 YEARS OF EXCELLENCE