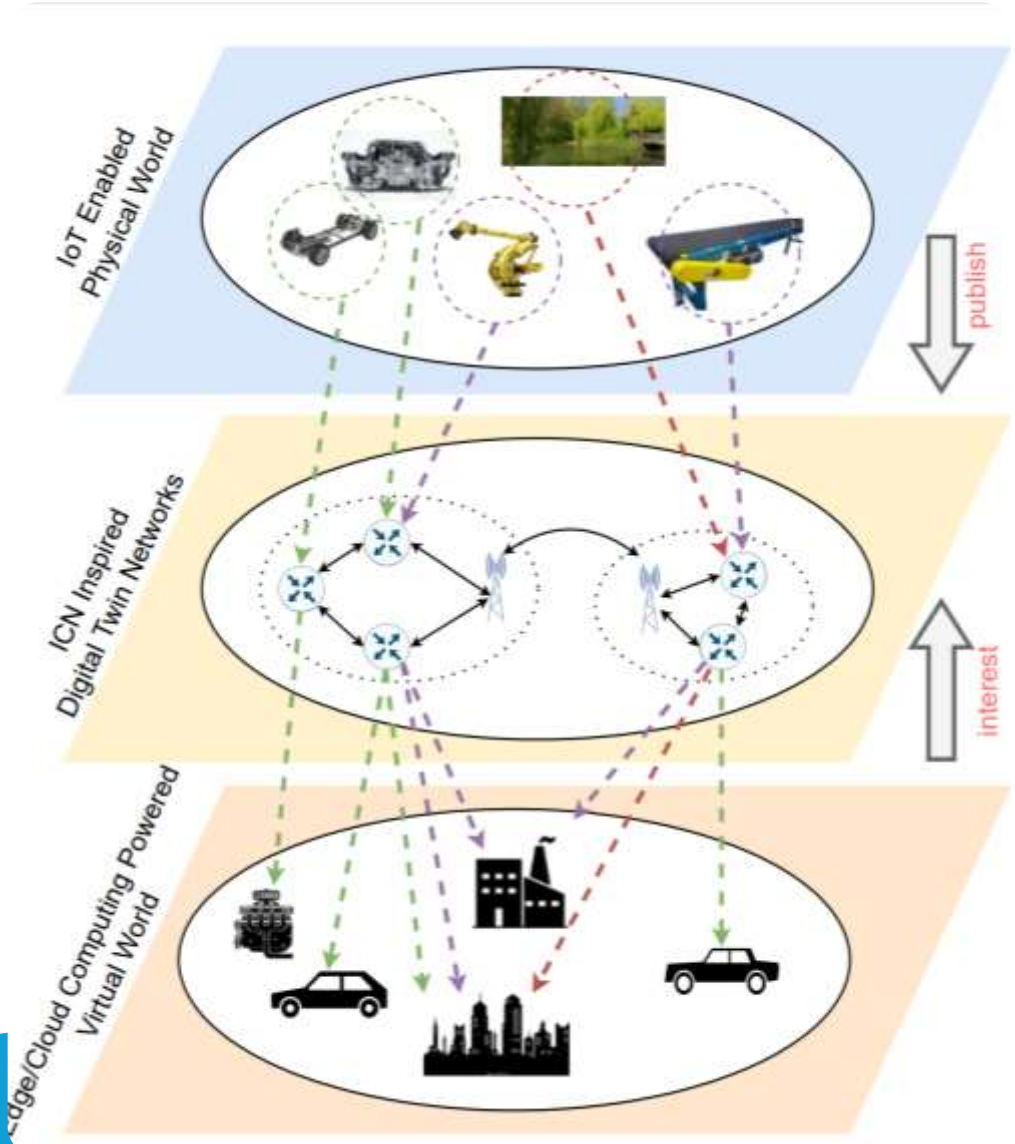
IoDT^2 – Internet of Digital Twin Things

Ziren Xiao, Research Associate
z.xiao@lboro.ac.uk
Loughborough University

Backgrounds: Digital Twins

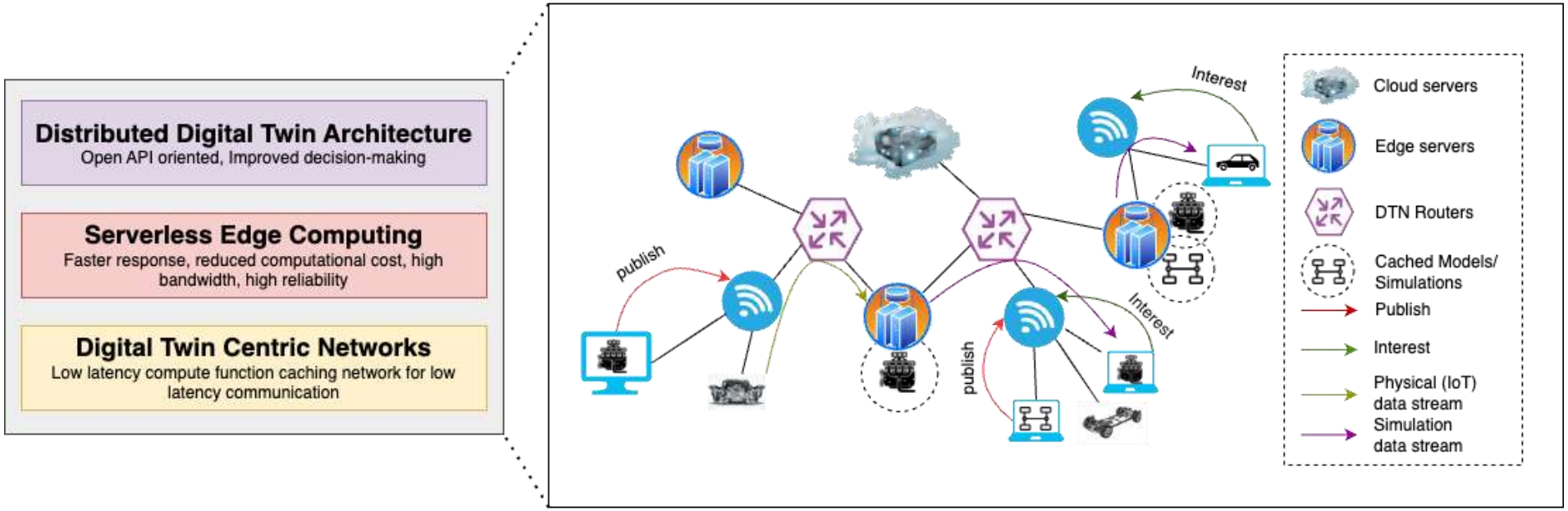
- A virtual representation of a physical object.
  - Smart cities
- It enables real-time monitoring, simulation, and optimisation by integrating data from sensors and other sources



IoDT²: Project Aims

- To overcome limitations of current digital twin practices
  - Centralisation
  - Proprietary solutions
  - Compatibility issues.
- A digital twin framework (IoDT²) that enables:
  - Easy sharing of digital twin models and data across networks.
  - Dynamic allocating computation resources for simulations.
  - Instantiation of models at edge, fog, or cloud levels.
- To empower practitioners to focus on creating necessary core models for their systems, rather than being bogged down by complex software configuration.

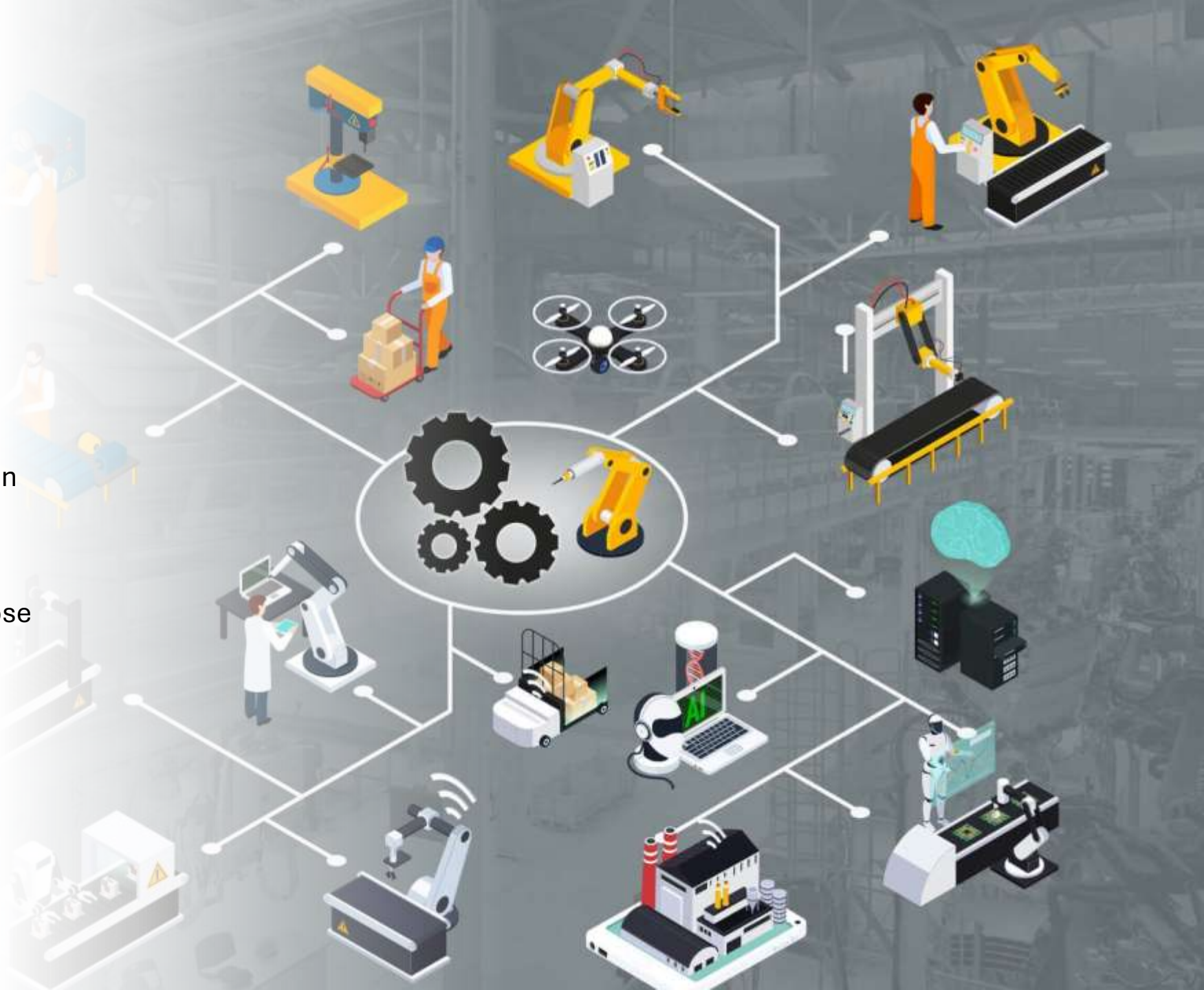
IoDT^2 Architecture



- Distributed Digital Twin Architecture
  - Focus on creating modular and well-defined digital twin models.
  - Enable dynamic composition and chaining from atomic models to create larger models.
  - Explore open descriptive software standards for interfacing with digital twin models to enhance interoperability.
- Serverless Edge Computing
  - Extend serverless computing capabilities from a cloud-centric environment to the edge.
  - Efficiently run digital twin simulations and perform relevant computations at the edge.
- Digital Twin Centric Networking
  - Apply Information-Centric Networking (ICN) principles to create a "digital-twin centric network".
  - Enable users to openly query the network for desired digital twin models and run simulations without specifying locations.

Interoperability: Questions

- Q: Can a digital twin model be automatically deployed without manual configuration?
  - Sub-Q1: Can existing digital twin models be used in other scenarios?
  - Sub-Q2: Is there any model/method that can compose multiple “small” digital twin models together?
  - Sub-Q3: How does the model/method accept various types of input and generate consistent output to external storage?

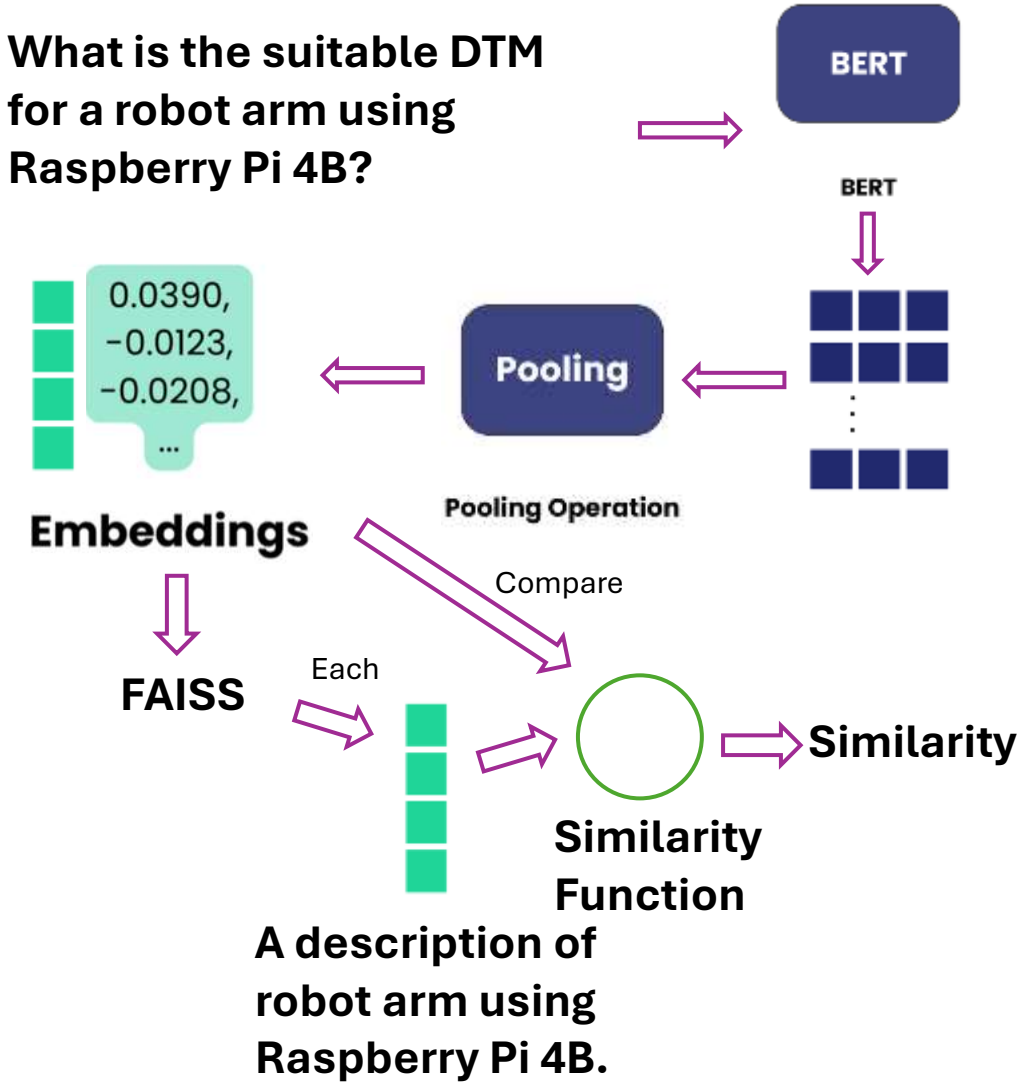


Interoperability: Goals

- Auto-deployment: Automatically deploy models based on use case descriptions, enable sharing, and allow model composition.
- Input & Output: Ensure smooth data flow from sensors to system outputs with consistent formatting and digital twin updates.
- Auto-Configuration: Automate the initialisation of models and networks.
- Scalability: Enable adding new models without retraining.

Stage 1: Digital Twin Model Sharing

- Objective: find the most suitable digital twin model in the database or other edge devices based on the user description
- Use LLM's encoder to encode the description
  - Pass through an LLM encoder
    - BERT
    - Sentence Transformer
  - Store in a vector database
- Compare the similarity of two vectors
  - Cosine similarity



Stage 2: Digital Twin Model Composition

- Objective: find and compose multiple digital twin models to form another model required by the user
- Decompose into multiple sub-models
- Search each sub-model using the model sharing method
- Concatenate the output and put it into the LLM
  - Reinforcement Learning-based training
  - The reward is based on the correct selections and configurations