Concepts of Digital Twin Network

draft-zhou-nmrg-digitaltwin-network-concepts-03

Cheng Zhou (zhouchengyjy@chinamobile.com, **Co-presenter**)
Hongwei Yang(yanghongwei@chinamobile.com)
Xiaodong Duan (duanxiaodong@chinamobile.com)
Diego Lopez (diego.r.lopez@telefonica.com, **Co-presenter**)
Antonio Pastor (antonio.pastorperales@telefonica.com)
Qin Wu (bill.wu@huawei.com, **Co-presenter**)
Mohamed Boucadair (mohamed.boucadair@orange.com)
Christian Jacquenet (christian.jacquenet@orange.com)

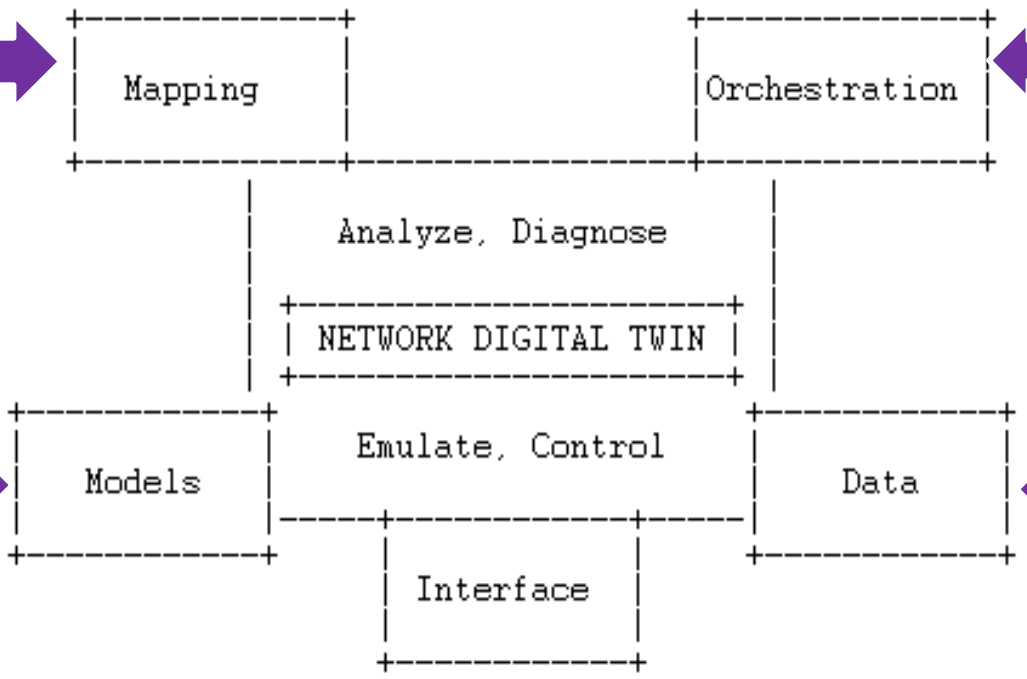
Document Status

- This the third time to present in this RG
  - Basic Idea: Redefine digital twin network in the network field
  - Goals:
    - Define a new tool for efficient and intelligent management
    - Stimulate innovation with optimized cycles
  - Technical Contributions:
    - Sketch a base Reference Architecture
    - Identify challenges to build digital twin networks
    - Clarify the relation with IBN

- Changes in -03 since IETF#109
  - **Key elements definition update**
  - **Add section on application scenarios**
  - **Add open issues section**
  - Split interaction with IBN part as a separate section
  - Clarify the motivation in the introduction section
  - Other editorial changes

Digital Twin Network Composition

Provides real-time interactive mapping between physical network and virtual twin network or mapping between two virtual twin networks



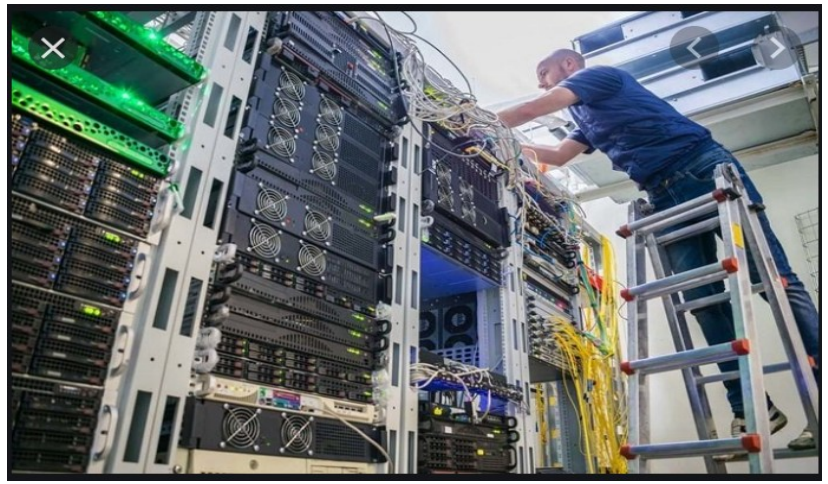
lifecycle management of components

Comprise the models based on physical network simulation, statistical data, performance metrics, inventory information, log information, and Artificial Intelligence

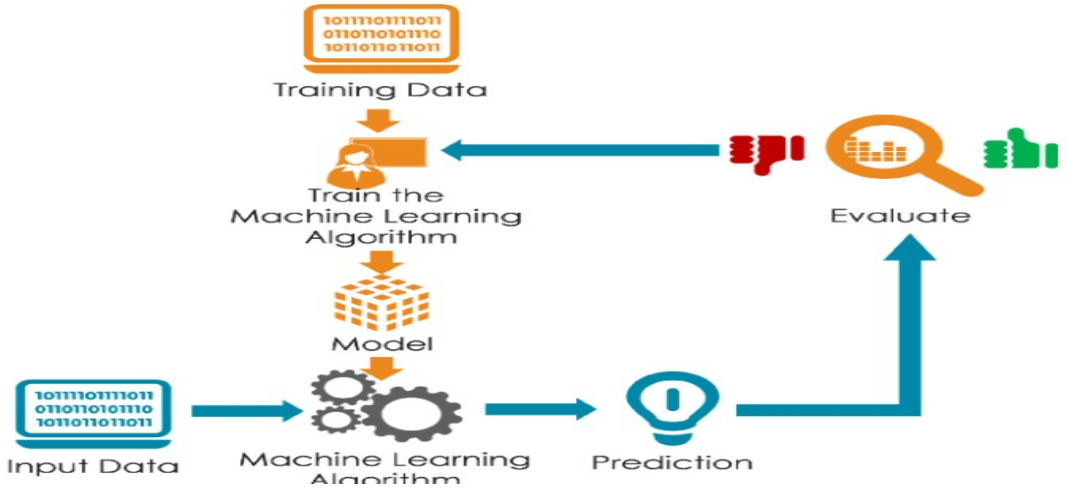
Data about its real-world twin that are required by the models to represent and understand the states and behaviors of the real-world twin

Service interfaces for network applications or other digital twins to access data and invoke capabilities. Telemetry interface for digital twin platform to populate and cache data

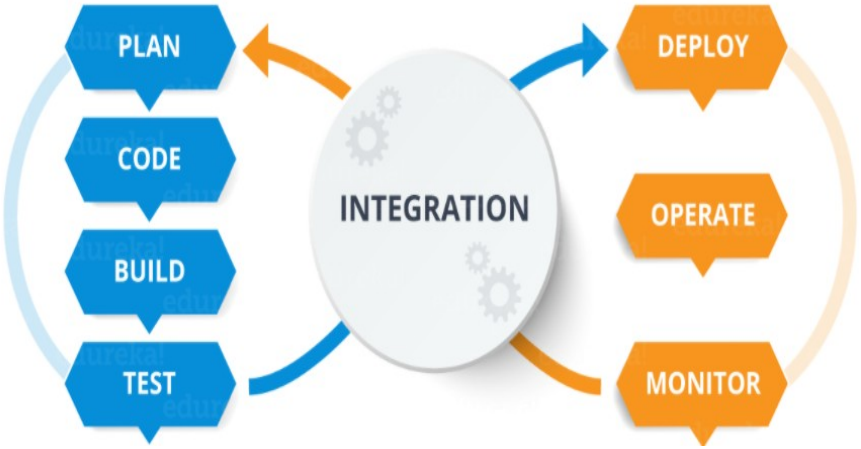
Sample Application Scenarios



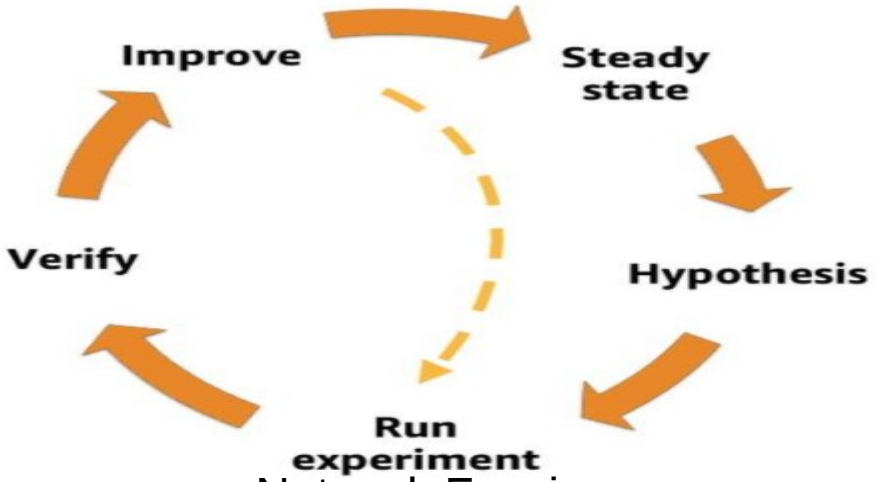
Network Maintenance Engineer Training



Machine Learning Training



DevOps oriented Certification



Network Fuzzing

Issue1: Why distinguish data from model?

- Data and models are usually the two common and separated components/elements in other industrial digital twin entities (e.g., manufacturing, factory-floor).
  - Of course, data can be structured to follow a set of well-known data models.
- Data and Model in digital twin networks
  - *Data*: a digital twin should contain data about its real-world twin (i.e., physical network) that are required by the models to represent and understand the states and behaviors of the real-world twin.
  - *Models*: A digital twin should contain computational or analytic models that are required to describe, understand, and predict the twins' operational states and behaviors, and models that are used to prescribe actions based on service logic and objectives about the corresponding real-world object.
  - In brief, data is cornerstone for constructing a DTN system; and various models are the power and source to Analyze, Diagnose, **Emulate** and Control the physical network.

Issue 2: How Orchestration is different from other components?

- Basically orchestration component aims to control and manage the twin entity, then helps to provide an integrated service to various applications.

- Two main orchestration features can be provided:
  1. Control the digital twin network environment and its components to derive the required/expected behavior
  2. Deal with the dynamic lifecycle management of these components by providing
     - Repeatability: Replicate network conditions on demand
     - Reproducibility: Replay successions of events, possibly under controlled variations

Issue 3: How should the interfaces be defined?

- Three types of interfaces were identified:
  1) Twin interface: between the physical network and its twin entity
     - It can be implemented using a variety of existing tools (telemetry, SNMP, NETCONF, etc.) or new ones.
  2) Application-facing interface: between the network digital twin and applications that make use of the emulated network.
     - For example, Intent, “what-if” planning app, ...
  3) Internal interfaces between components within network digital twin
- We need to first define or choose the first two types of interfaces, then focus on the internal interfaces to build the twin image.
  - The first two interfaces should be open, standardized, real-time, secure, and reliable.
  - Internal interfaces should be with capability of high-speed, high-efficiency, high-concurrency etc.

Issue 4: Which component is responsible for checking for deviation of the underlay network vs. the image?

- Mapping component is responsible for such checking
- From traditional simulation to emulation, with real-time interactive mapping.
  - Digital twin network provides **real-time interactive mapping** between physical network and virtual twin network, that **emulates** the behavior of a network by calculating the deviation between the different network entities (routers, switches, nodes, access points, links, etc.) in the physical network and corresponding entities in the virtual twin network.

- Mapping can be:
  - One to one mapping (pairing, vertical): Synchronize between a physical network and its virtual twin network **with continuous flow**
  - One to many mapping (coupling, horizontal): Synchronize among virtual twin networks with **occasional data exchange**

Issue 5: Continuous Verification vs CI/CD

- Modern DevOps practices involve continuous development/testing/integration, /deployment/monitoring of software applications throughout its development life cycle:
  - ***Continuous Integration (CI)*** allow implement small changes and check in code to version control repositories frequently.
    - E.g., committing all your application code in a single repository
  - ***Continuous Delivery (CD)*** automates the delivery of applications to selected infrastructure environments. (e.g., network digital twin)
    - E.g., Travis CI allows automatically run CI tasks like unit tests and push your code to a hosting platform every time you push new changes to a branch.

- Continuous Verification (CV) is an extension of DevOps practices that are concerned with verifying the system as a whole.
  - The application of CI/CD models in network management operations increases the risk associated to deployment of non-validated updates
  - CV can be used in **DevOps-oriented certification application** to address it.

Next Steps

- **Investigate** related digital twin network work and identify the differences and commonality, e.g.,
  - How this concept and architecture is different from digital twin for industry application
  - How existing network management models can be re-used?
- **Better articulate** the relationship between data, model, mapping.
- **Analyze** requirements for twin and application interfaces, as well as for pairing and coupling mechanisms

- **Address** any issues raised in this meeting