Named Data Networking(NDN) for IoT System (Smart Water Meter Collecting System)

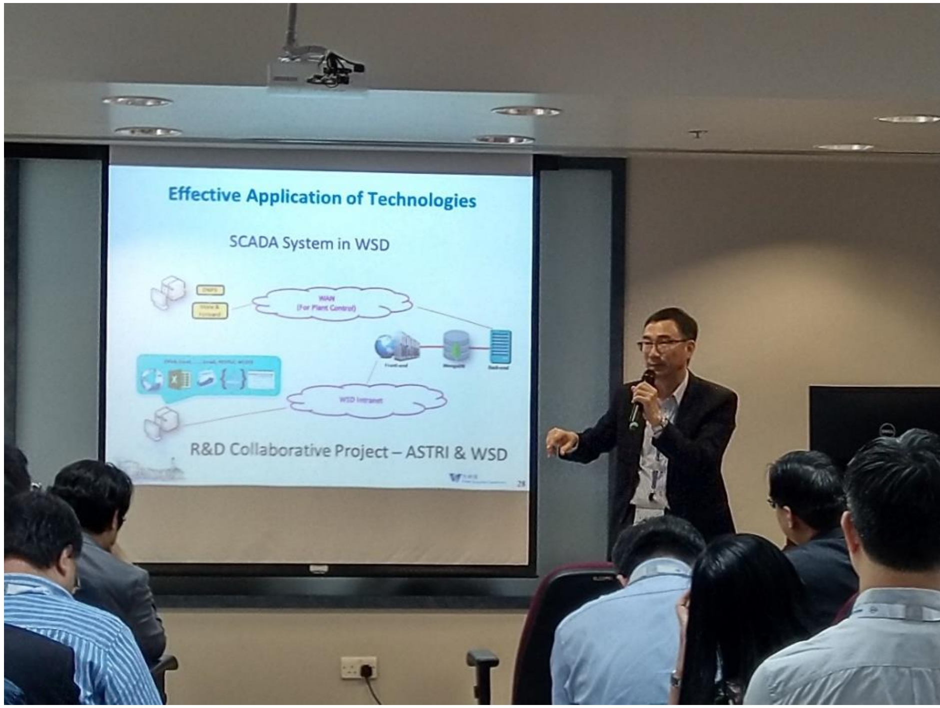
Project Team’s Core Competence

中华人民共和国科学技术部
“十三五”国家重点研发计划

水務署
Water Supplies Department

SAPIENTIA·ET·VIRTUS

- Smart City (Smart Water)
  - Smart Behaviour Analytics Platform with Machine Learning for Utilities Applications (ART/242CP)
    - **Time-Series Data Management** module in WSD’s SCADA (Supervisory Control And Data Acquisition) platform
  - Industrial IoT Platform with DNP3 and LoRa (ART/270CP)
- ICT Awards
  - WITSA Global ICT Excellence Awards 2018
  - Asia Pacific ICT Alliance Awards 2017
  - Hong Kong ICT 2017: Best Business Solution (Application) Bronze Award
  - 2008/2012 Hong Kong Olympics Online Streaming


Effective Application of Technologies
SCADA System in WSD
R&D Collaborative Project – ASTRI & WSD

WORLD INFORMATION WITSA TECHNOLOGY AND SERVICES ALLIANCE

APICTA Asia Pacific ICT Alliance AWARDS 2017 Merit Award

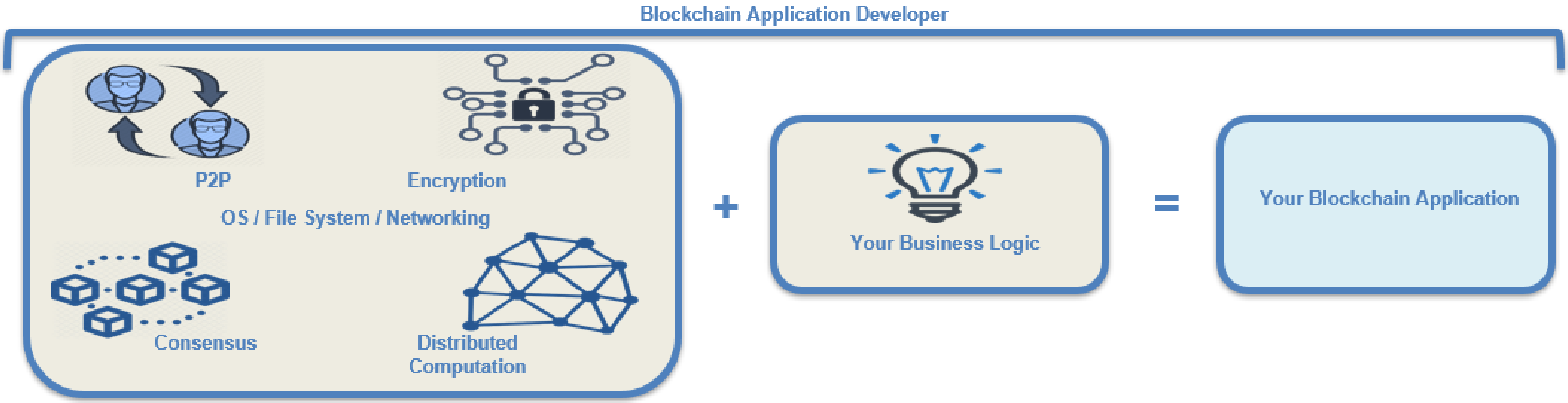
CABLE TV 有線電視

4.a Problem Statement

Current Situation	Out Solution
Centralized, without blockchain certification	Decentralized, with blockchain certification
TCP/IP(low efficiency)	NDN(high efficiency)
3/4/5G(high power)	Lora(low power)
>5000 HKD(High cost)	<1000 HKD(low cost)

Technical Approach – Develop a blockchain application(General)

To Develop an blockchain application (General)



Blockchain Application Developer

P2P

Encryption

OS / File System / Networking

Consensus

Distributed Computation

+

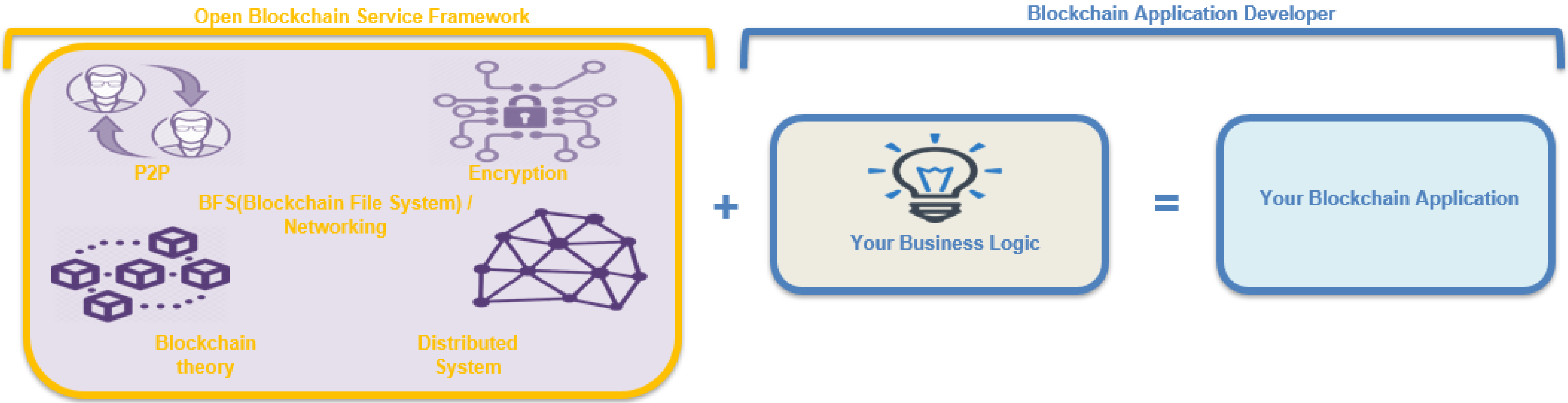
Your Business Logic

=

Your Blockchain Application

Technical Approach – Develop a blockchain application(with BFS & BFC)

To Develop a blockchain application (with BFS&BFC)



Open Blockchain Service Framework

P2P

Encryption

BFS(Blockchain File System) / Networking

Blockchain theory

Distributed System

+

Blockchain Application Developer

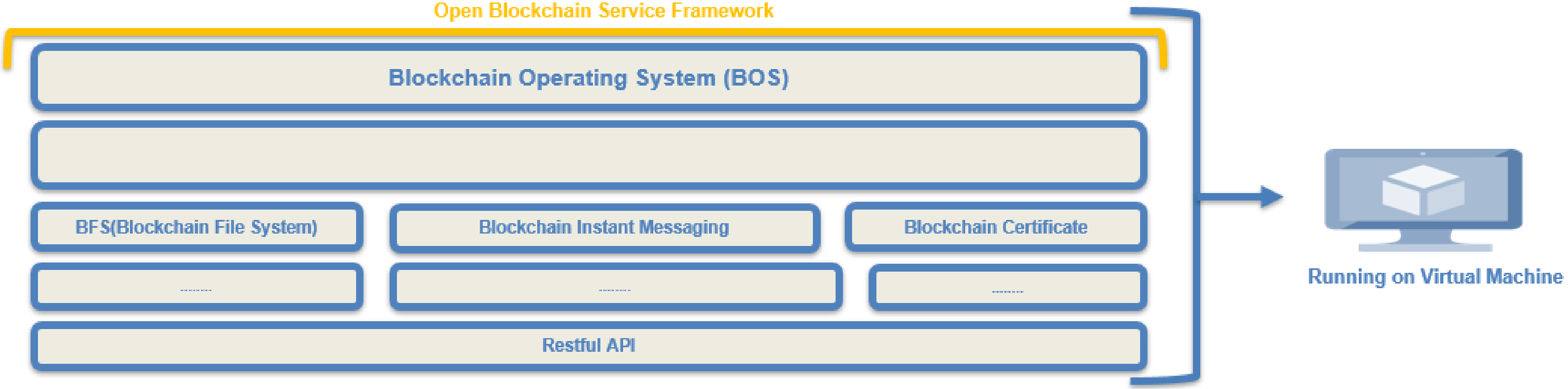
Your Business Logic

=

Your Blockchain Application

Technical Approach – Architecture

BFS & BFC & BOS Service Framework



Open Blockchain Service Framework

Blockchain Operating System (BOS)

BFS(Blockchain File System)

Blockchain Instant Messaging

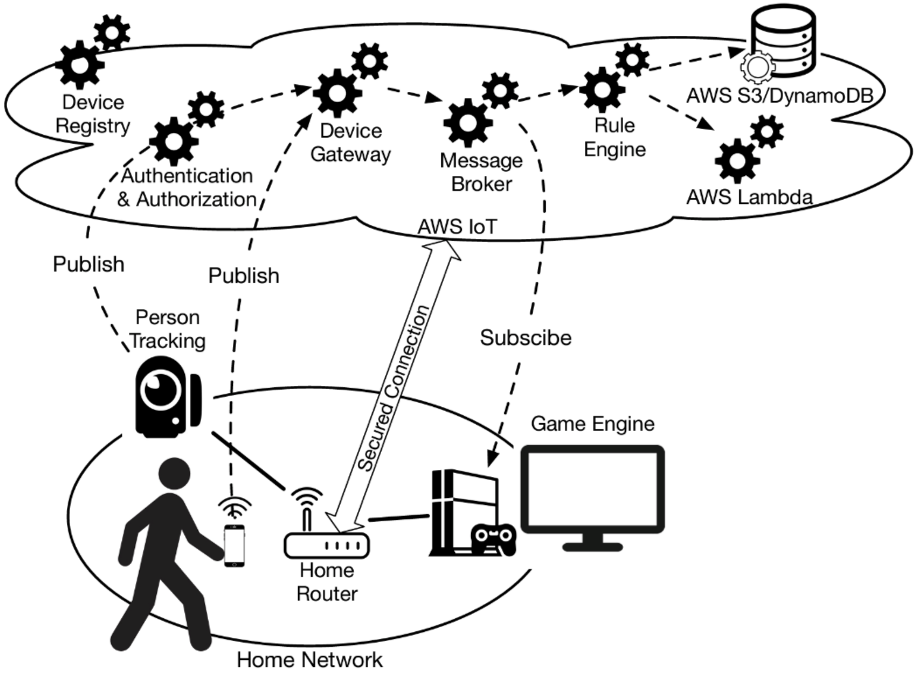
Blockchain Certificate

Restful API

Running on Virtual Machine

Problem Statement — Problems with Cloud-Centric IoT

- Local IoT operations subject to remote failures:
  - Internet connectivity to the cloud may be lost.
  - Cloud services are not immune to failures.
- Poor real-time interactive experience when local interactions have to go through cloud.
- Expose private data to cloud providers.

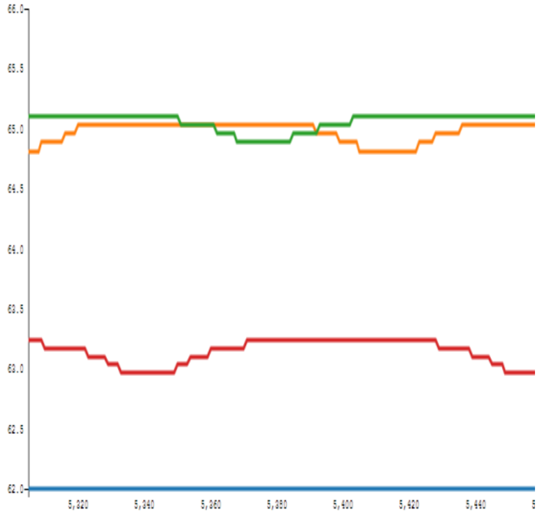


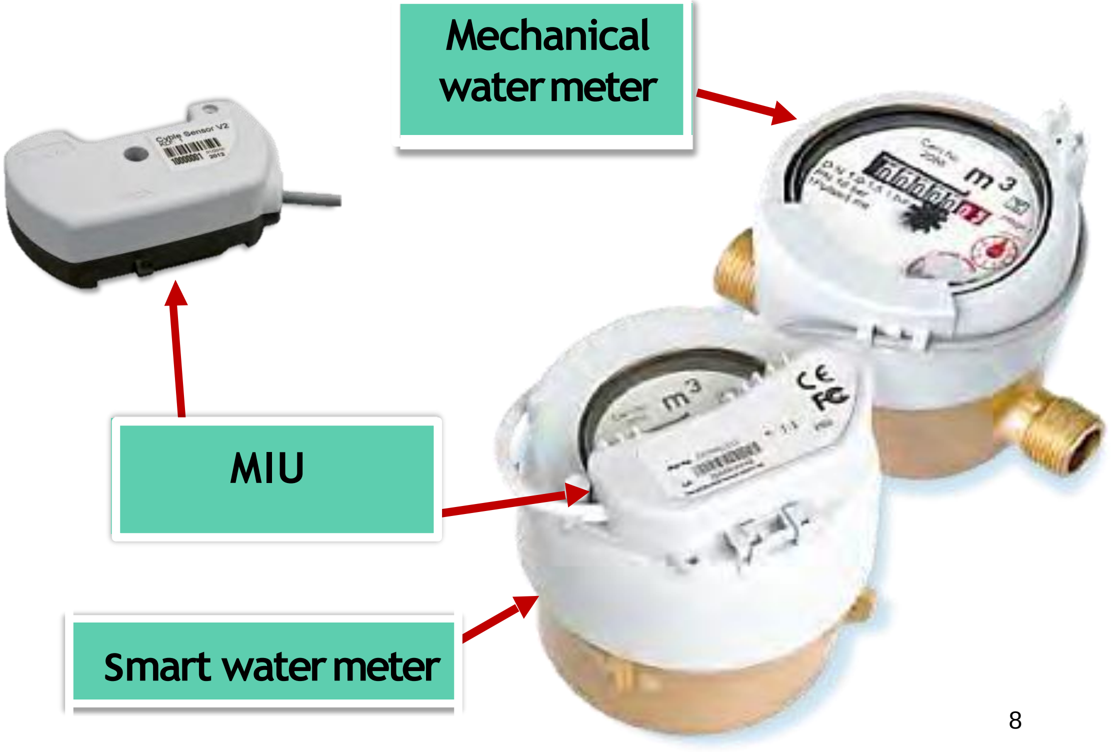
7

Problem Statement — Meter Interface Unit (MIU)

Smart water meter = Mechanical water meter + Meter Interfacing Unit (MIU)

- Measurement error due to banging pipe
  - Computing algorithm to overcome the problem





Mechanical water meter

MIU

Smart water meter

Problem Statement — LoRa low-power star network

- WSD's proposed requirements are challenging

Item	WSD's proposed requirement
Reading frequency	30 minutes
Report frequency	4 hours
Accuracy	Inaccuracy < 3%
Battery life	6 years
LoRa frequency band	920-925MHz

- Find suitable LoRa parameters in HK environment
  - Test Parameters: Bandwidth (kHz), spreading factor, transmit power, coding rate, transmit byte (frame bytes), number of transmit frames (test frame count), with antenna, hardware Type, Tsym, Tpb, Tpl, calculate transmit duration, Estimation sensitivity, total test duration, average transmission time per frame, communication success rate, signal strength M, signal strength S

Technical Approach

Why NDN for IoT?

Simplifying app development and management

- NDN name are semantically meaningful.
- No need to manage IP addresses or map names to addresses
- The Interest-Data exchange model matches application message pattern.

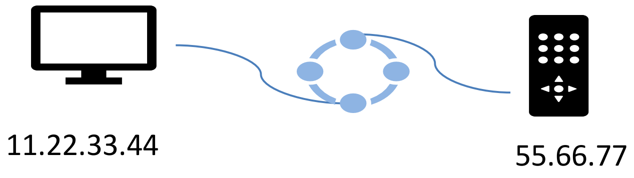
Improving performance

- NDN's data-centric model supports multi-party communication over multicast medium.
- Ubiquitous in-network data caching helps improve the efficiency of information dissemination, especially for resource constrained IoT environments.

Achieving security

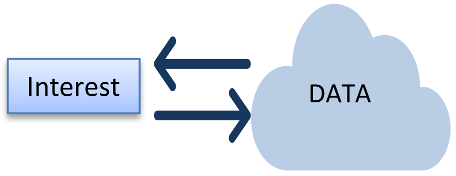
- Data-centric security does not require both ends to be on ensures end-to-end security.

Technical Approach – TCP/IP vs NDN



IP: connect to a server

Telling the network which IP to go to, the network will have a routing protocol and exchange protocol to find a way to pass the package to the destination. If there is a broken road in the middle, you can find it separately.



NDN: retrieve a content

In the NDN network, you only need to send a request, such as: room/temperature. Anyone with this information can return it to the user. You don't need to know how to get this information, and you don't need to know exactly where this information comes from.

Technical Approach – Example: Getting Room Temperature in TCP/IP

TCP/IP

1. Get the name of the device from a local server or cloud — thermostat1.livingroom.<homenet>
2. Do a DNS lookup to find its IP address — 10.0.1.9
3. Connect to the device using DTLS (Datagram Transport Layer Security) over UDP or TLS over TCP
4. Send CoAP (Constrained Application Protocol) message to fetch temperature from the thermostat
5. Receive temperature data

Note: each step may involve multiple messages.

NDN

1. send Interest /<homenet>/livingroom/temp
2. receive data and verify data authenticity using signature and trust schema
3. If data is encrypted with content key
   a. send Interest to fetch content key
   b. receive content key and decrypt data

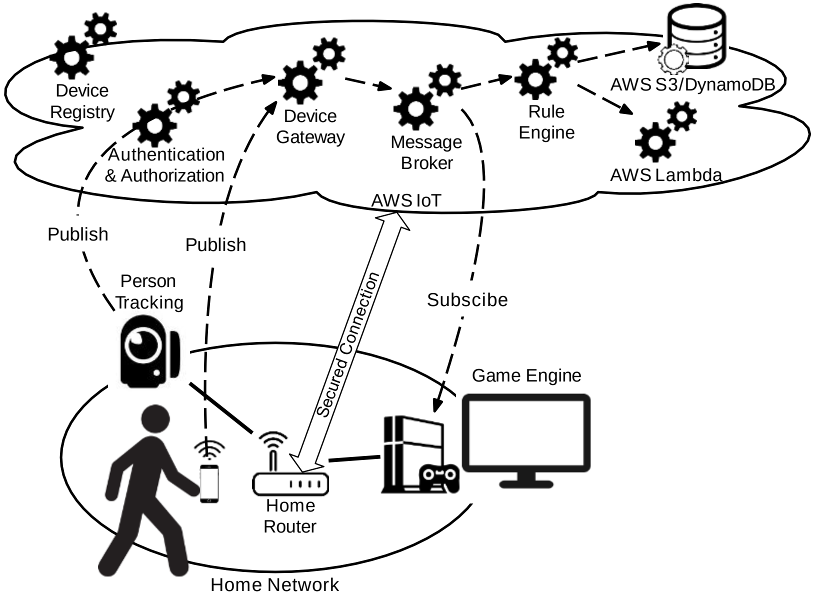
Note:

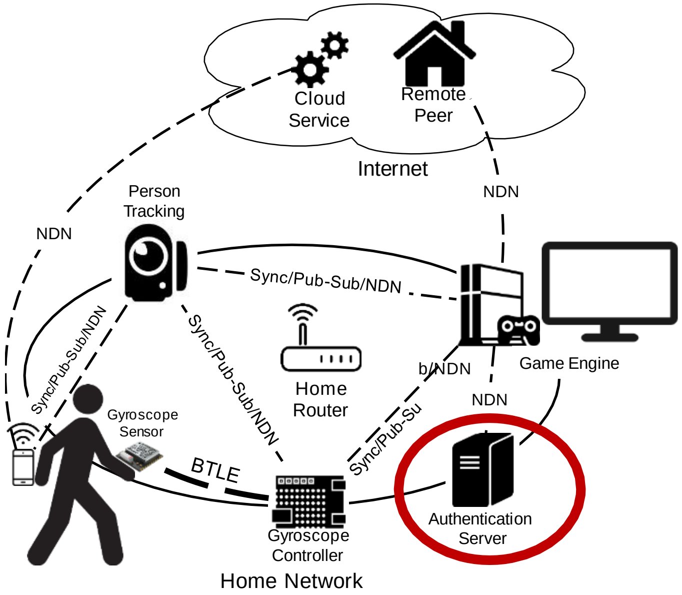
Content key is encrypted with the consumer's public key (or a key encryption key shared with the consumer).

Encrypted content key is published using pre-established naming scheme.

Technical Approach – Architecture: AWS-IoT vs. NDN-IoT

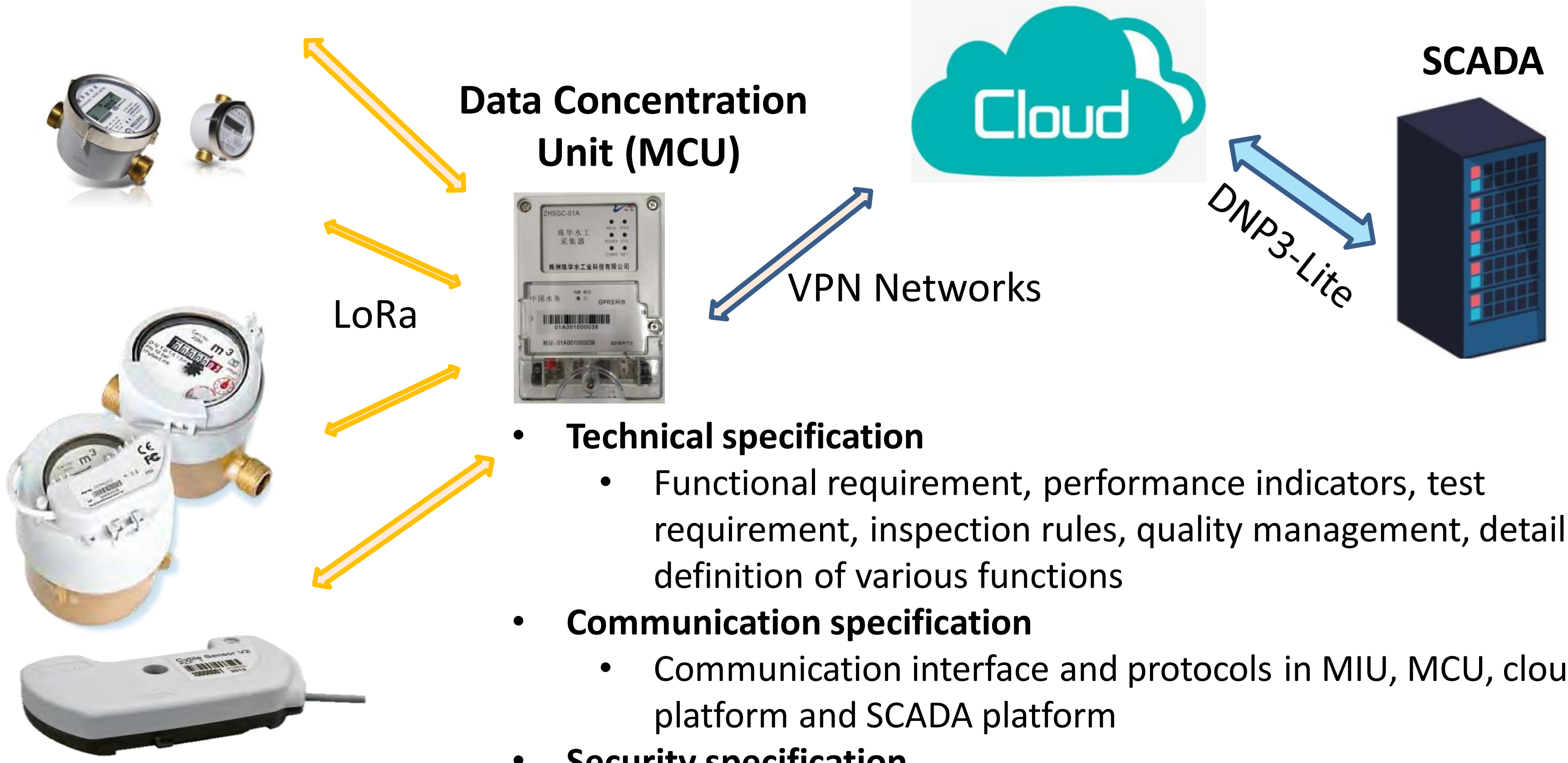
Cloud increases user interaction delay.





Semantic meaning → local trust anchor → local autonomy

Technical Approach – System Architecture of Smart Water Meter Collecting System

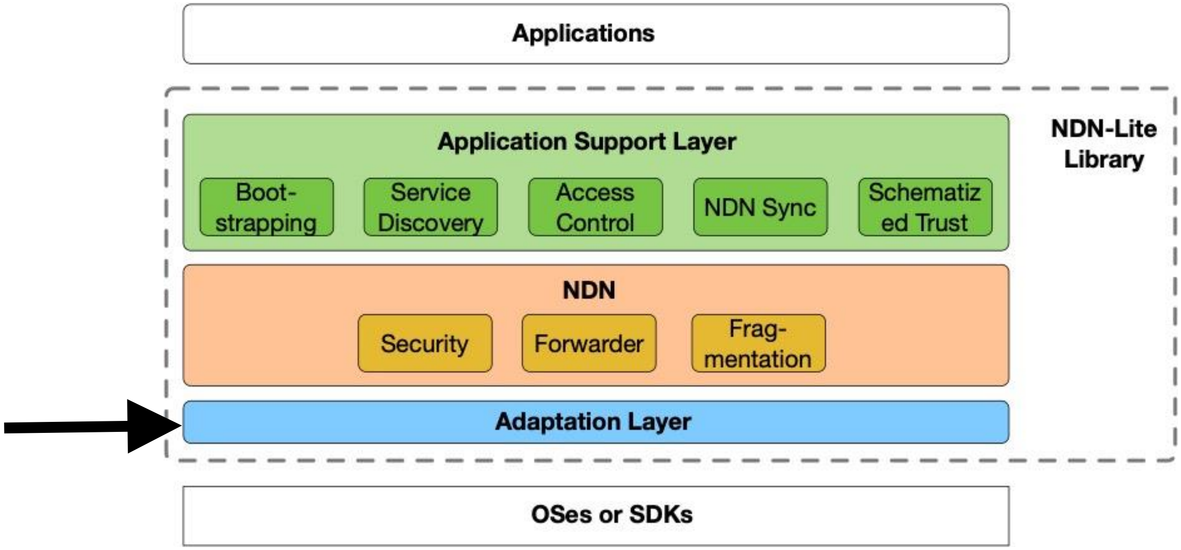


Meter Interface Unit (MIU)

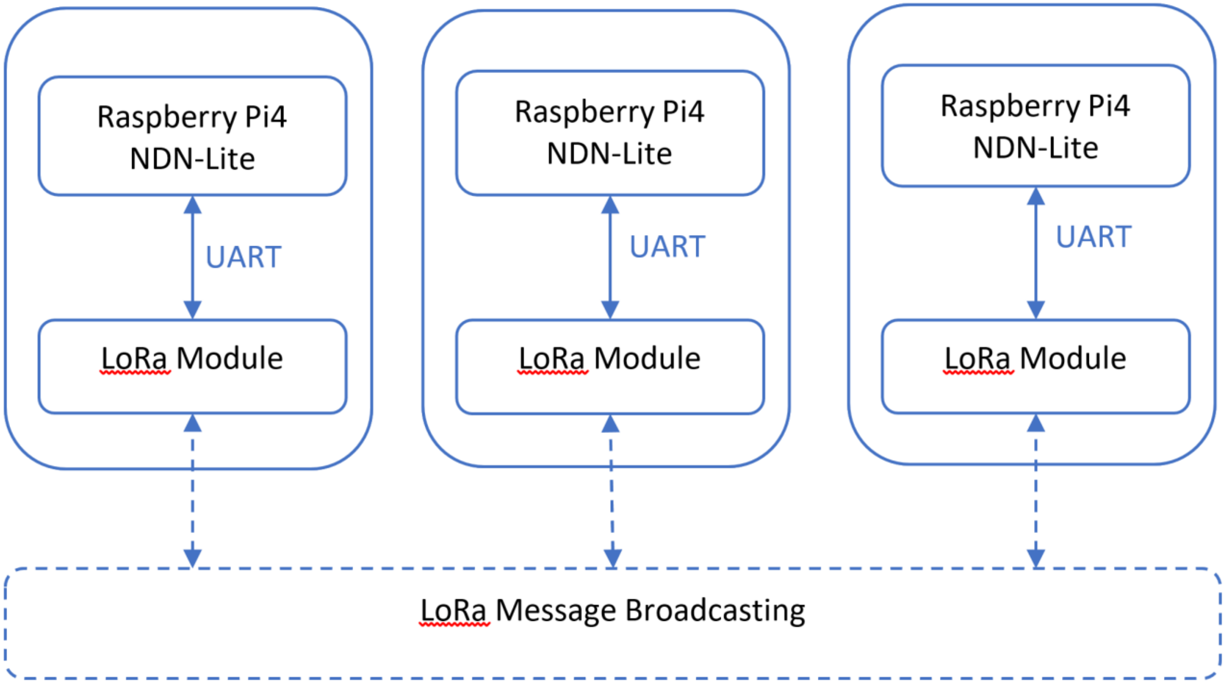
- **Technical specification**
  - Functional requirement, performance indicators, test requirement, inspection rules, quality management, detailed definition of various functions
- **Communication specification**
  - Communication interface and protocols in MIU, MCU, cloud platform and SCADA platform
- **Security specification**
  - Data transmission and storage encryption requirements, and encryption methods
- **Inspection specification**
  - Factory acceptance, acceptance methods, inspection items and inspection methods.
- **Data storage specification**
  - Data storage mode, format, read and write interface.

Technical Approach – Design goal

ndn_lora_face_t ()



Technical Approach – Architecture



Sending flow:

1) Encode data and use Lora face to send NDN pakcets
2) Raspberry transmits the packet to Lora module via UART
3) Lora module sends out the Lora message

Receiving flow:

1) Lora module receives the lora message from the another node
2) Use UART to transmit the message to Raspberry
3) Decode the message as the NDN packet

Technical Approach – Implementation-Software

Flag for detecting a full packet**:**

```
static uint8_t magic[4] = {0x80, 0xdb, 0xa9, 0x3e};
```

```
ndn_lora_face_send(ndn_face_intf_t* self, const uint8_t*
packet, uint32_t size){
  ndn_lora_face_t* ptr = (ndn_lora_face_t*)self;
  ssize_t ret = write (ptr->fd, packet, size);
  ret += write(ptr->fd, &magic, sizeof(magic));
  if(ret != size + 4){
    return NDN_LORA_FACE_SOCKET_ERROR;
  }
  else{
    return NDN_SUCCESS;
  }
}
```

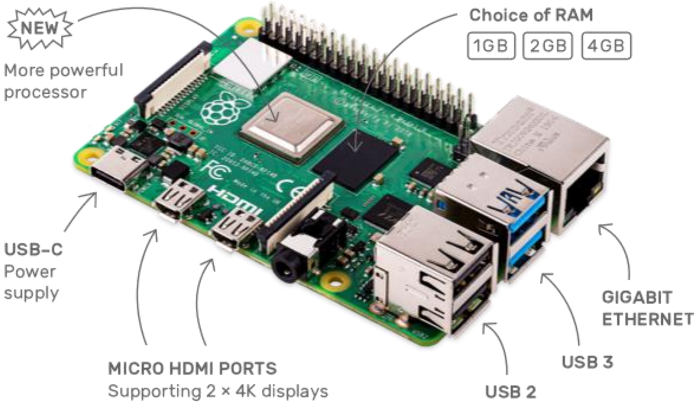
```
ndn_lora_face_recv(void *self, size_t param_len, void *param){
 ssize_t size;
 int ret;
 ndn_lora_face_t* ptr = (ndn_lora_face_t*)self;
while(true){
   size = recvfrom_lora(ptr);
   if(size > 0){
     ret = ndn_forwarder_receive(&ptr->intf, ptr->buf, size);
    }
   else if(size == 0){ break; }
   else{
      ndn_face_down(&ptr->intf); return; }
  }
  ptr->process_event = ndn_msgqueue_post(self,
ndn_lora_face_recv, param_len, param);
}
```

How to detect a full packet:

```
static ssize_t
recvfrom_lora (ndn_lora_face_t* ptr) {
int buffPos = 0;
uint32_t window = 0;
while (true) {
  if (serialDataAvail(ptr->fd) == 0)
    continue;
  ptr->buf[buffPos] = serialGetchar(ptr-
>fd);
  window = byte_shift_left(window, ptr-
>buf[buffPos++]);
  if (memcmp(&window, magic,
sizeof(magic)) == 0) {
    return buffPos – 4;
  }
}
```

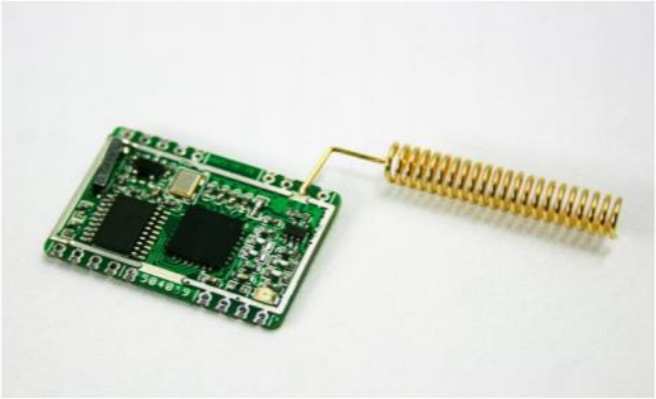
Technical Approach – Implementation-hardware

NDN lite running on Raspberry pi



Raspberry pi 4:
OS: Raspbian (4.19)
Memory: 4GB
CPU: ARM v8 1.5Ghz

Lora module connected with Raspberry pi



GC-TS12

GC-TS12
Programmable bit rates up to 300 kbps
High sensitivity: down to -148 dBm
Long transmit distance, up to 3000 meters in open area
Low power consumption, 3uA stand-by, 12mA in receiving mode