International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.4464

(01/2020)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Internet of things and smart cities and communities – Frameworks, architectures and protocols

Framework of blockchain of things as decentralized service platform

Recommendation ITU-T Y.4464

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	**Y.4400–Y.4549**
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.4464

Framework of blockchain of things as decentralized service platform

Summary

Recommendation ITU-T Y.4464 introduces a decentralized IoT service platform, blockchain of things (BoT), which is enabled by blockchain-related technologies. This Recommendation analyses the concept, common characteristics and high-level requirements of BoT, and provides common capabilities and functionalities, general procedures and relevant use cases for BoT.

BoT works in a decentralized service mode and is capable of enhancing many aspects of IoT. It has the advantages of blockchain-related technologies, especially for building decentralized data storage and management, crowding decision-making and automatic interactions.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4464	2020-01-13	20	11.1002/1000/14167

Keywords

Blockchain, blockchain of things, Internet of things.

---

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.4464

Framework of blockchain of things as decentralized service platform

1 Scope

This Recommendation introduces the concept of blockchain of things (BoT), analyses its common characteristics and specifies high-level requirements of it as a decentralized service platform for Internet of things (IoT). It describes BoT-related common capabilities of the IoT reference model specified in [ITU-T Y.4000].

The scope of this Recommendation includes:

– introduction of blockchain of things;

– common characteristics and high-level requirements of BoT;

– relevant common capabilities and functionalities, and general procedures.

In addition, this Recommendation provides in the appendices deployment modes, use cases, technical analysis and comparison, business roles and models to illustrate how BoT can provide alternative approaches for the improvement of IoT applications and services.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of Things*.

[ITU-T Y.4100] Recommendation ITU-T Y.4100/Y.2066 (2014), *Common requirements of the Internet of things*.

[ITU-T Y.4103] Recommendation ITU-T Y.4103/F.748.0 (2014), *Common requirements for Internet of things (IoT) applications*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 address [b-ITU-T Y.2091]: An address is the identifier for a specific termination point and is used for routing to this termination point.

3.1.2 application [b-ITU-T Y.2091]: A structured set of capabilities, which provide value-added functionality supported by one or more services, which may be supported by an API interface.

3.1.3 blockchain [b-FG-DPM TR D3.5]: A peer to peer distributed ledger based on a group of technologies for a new generation of transactional applications which may maintain a continuously growing list of cryptographically secured data records hardened against tampering and revision.

NOTE 1 – Blockchains can help establish trust, accountability and transparency while streamlining business processes.

NOTE 2 – Blockchains can be classified as three types (i.e., public, consortium and private) based on the relationship of the participants and the way to provide services.

3.1.4 consensus [b-FG-DPM TR D3.5]: Agreements to confirm the correctness of the blockchain transaction.

3.1.5 constrained device [b-ITU-T Y.4451]: A device that has constraints on characteristics such as limited processing capability, small memory capability, limited battery power, short range and low bit rate.

3.1.6 device [ITU-T Y.4000]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.7 Internet of things [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – In a broad perspective, the IoT can be perceived as a vision with technological and societal implications.

3.1.8 smart contract [b-FG-DPM TR D3.5]: Embedded logic that encodes the rules for specific types of blockchain transactions. A smart contract can be stored in the blockchain, and can be invoked by specific blockchain applications.

3.1.9 thing [ITU-T Y.4000]: In the Internet of things, object of the physical world (physical things) or of the information world (virtual things), which is capable of being identified and integrated into the communication networks.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 blockchain of things (BoT): An Internet of things (IoT) service platform with a decentralized working mode where IoT applications and services are executed.

NOTE – BoT uses blockchain-related technologies (such as peer to peer communication, decentralized data storage, crowding consensus and transaction) to support a decentralized working mode.

3.2.2 blockchain platform: A platform (or system) that is established based on blockchain-related technologies.

3.2.3 BoT data: The data within a blockchain of things (BoT), besides the traditional Internet of things (IoT) data, includes the data such as distributed append-only ledgers, state information, permission policies, etc.

NOTE – BoT data may be distributed and be stored in BoT peers. A BoT peer may store the whole or a part of the data in a BoT.

3.2.4 BoT entity: A functional entity or physical entity (e.g., Internet of things (IoT) server, IoT device, IoT gateway and end user device) that participates in the activities performed in a blockchain of things (BoT).

NOTE – Physical things with constrained capabilities on computation or communication (e.g., constrained IoT devices, ID tags) are usually not BoT entities, but they can be bound to BoT entities in order to be mapped into BoT. Virtual things can be bound to BoT entities in order to be mapped into BoT.

3.2.5 BoT peer: A logic function of a blockchain of things (BoT) entity to perform BoT-related functionalities (e.g., executing transactions and maintaining BoT data) in proceesses of communications, consensus-making, transactions and management.

3.2.6 BoT transaction: An operation (e.g., reading/writing blockchain of things (BoT) data, deploying/invoking smart contracts, and querying results of smart contracts) that is performed by authorized BoT peers.

NOTE – BoT may provide incentive mechanisms (e.g., rewarding fees or tokens) to encourage the participants to contribute to the transaction (e.g., maintaining BoT data, endorsing the transaction).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

BoT Blockchain of Things

dApp decentralized Application

DDoS Distributed Denial of Service

ID Identity

IoT Internet of Things

ITS Intelligent Transportation System

P2P Peer-to-Peer

VM Virtual Machine

5 Conventions

The following conventions are used in this Recommendation:

– The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed;

– The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

6 Introduction of blockchain of things

IoT service platforms vary widely to meet different demands of networks and services in heterogeneous environments. Typically, traditional IoT service platforms are centralized or distributed. Without loss of generality, a traditional IoT service platform can be illustrated in the diagram shown in Figure 6-1. A traditional IoT service platform, deployed in a centralized or distributed mode, acts as a connection hub for IoT applications and services to access IoT devices, IoT data and things.

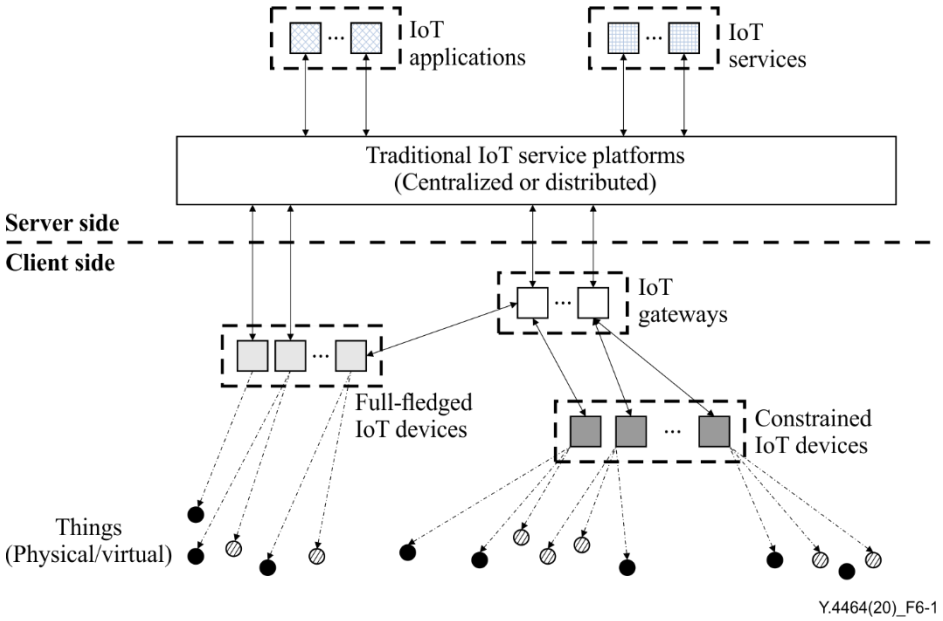

Figure 6-1 – Overview of traditional IoT service platforms

IoT service platforms are categorized in three types of working modes according to their deployment and cooperation mechanisms: centralized, distributed and decentralized (see Figure 6-2). These types of IoT service platforms can work on the same or different underlying infrastructures (e.g., network, cloud, big-data, security, management, accounting, charging).

In a centralized working mode, an IoT service platform is deployed in a single location on the server side (such as in a data centre) and managed by one platform provider, which provides centralized services to IoT applications and services in single location.

In a distributed working mode, an IoT service platform is deployed in multiple locations on the server side and managed by one platform provider, which provides distributed services to IoT applications and services.

In a decentralized working mode, an IoT service platform is established and maintained by a group of independent stakeholders who may be uncorrelated or untrusted. In this working mode, IoT applications and services and IoT devices can be part of the service platform and jointly provide services to one another. In addition, it is not necessary to consider the location of the participants, as these can be deployed freely on the server side or on the client side.

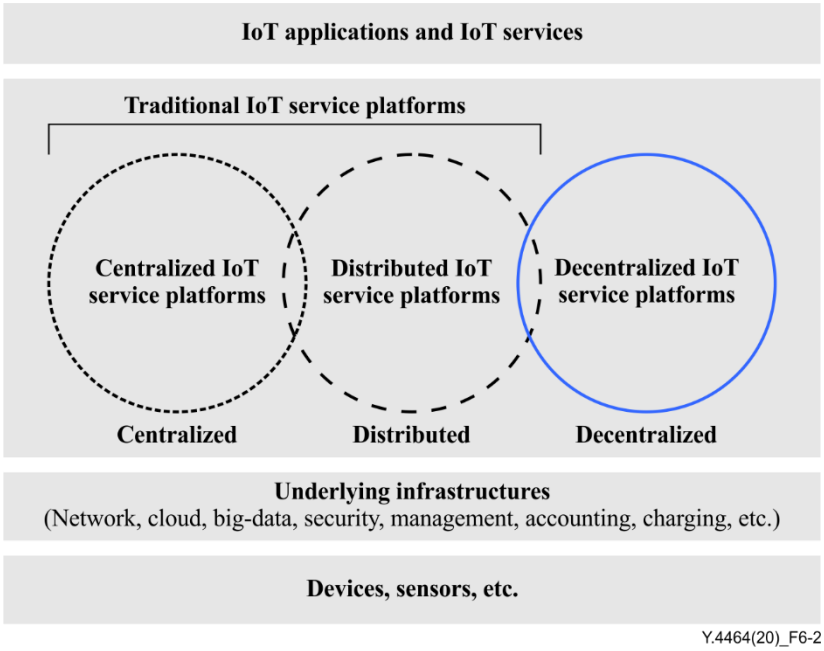

Figure 6-2 – Working modes of IoT service platforms

BoT is one type of a decentralized service platform (see Figure 6-3), based on blockchain-related technologies, enabling BoT entities (e.g., IoT devices, IoT servers, IoT gateways, service gateways and end-user devices) to participate in and perform transactions together. A BoT entity can host one or multiple BoT peer(s) and decentralized application(s) (dApp). A dApp on a BoT entity connects to BoT peers to perform transactions on BoT.

BoT peers interact with one another through peer-to-peer and fully decentralized communication mechanisms over current and future communication networks. Virtual things of the IoT can be mapped into a BoT through BoT entities, and physical things with constrained capabilities on computation or communication (e.g., constrained IoT devices, ID tags) can be bound to BoT entities (e.g., IoT gateways) in order to be mapped into BoT.

BoT works in a decentralized environment established on blockchain-related technologies or is provided by a blockchain platform.

The underlying infrastructures provide supporting capabilities to BoT, such as general and special supporting capabilities, general and special management capabilities, and general and special security capabilities described in [ITU-T Y.4000].

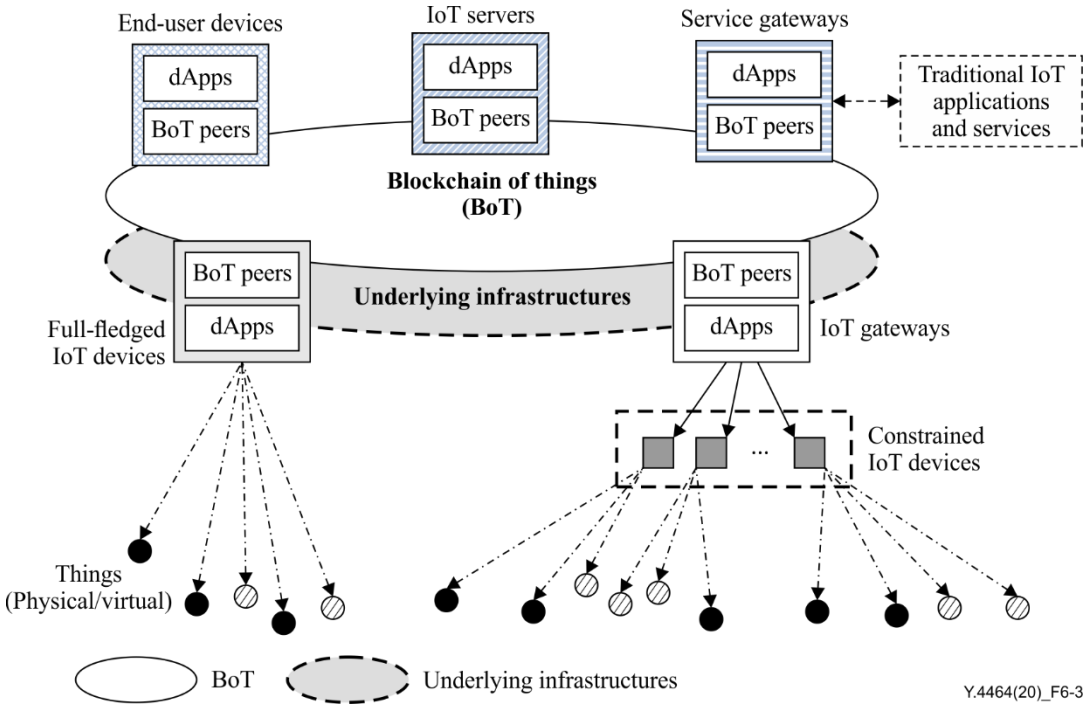

Figure 6-3 – Overview of BoT

BoT has the advantages of blockchains in many fields, especially for building trust, reducing costs, accelerating transactions, and connecting large-scale IoT devices and things. Additionally, the decentralized data management mechanism of BoT makes it easier to create cost-efficient business networks where virtually anything of value can be tracked and traced, without requiring a central point of control.

7 Common characteristics and high-level requirements of BoT

7.1 Common characteristics

[ITU-T Y.4000] describes a group of fundamental characteristics about interconnectivity, services, heterogeneity, dynamic changes and enormous scales of the IoT.

[ITU-T Y.4103] describes another group of fundamental characteristics of the IoT, including common characteristics, social characteristics, autonomy characteristics, and characteristics related to self-replication or control.

Besides the fundamental characteristics that are described in [ITU-T Y.4000] and [ITU-T Y.4103], BoT also has some common characteristics as related to its inherent feature of decentralization.

7.1.1 Supporting decentralized data storage and management

In a traditional IoT service platform, IoT data are usually centralized and/or distributed and stored on the server side and managed by the provider of the service platform. In contrast, in BoT, according to its deployment modes (see Appendix I), entire or parts of the authorized BoT peers contribute to the storage and management of IoT data. BoT provides mechanisms to support authorized BoT peers (which may have no trust relationship with each other) to work together to process IoT data and to maintain data integrity, confidentiality, privacy and accessibility.

7.1.2 Supporting crowding decision-making

In a traditional IoT service platform, decisions and trust are usually performed by providers of the service platform on the server side. In BoT, part or all BoT peers participate in making decisions subject to the deployment mode and policy of BoT. The participants involved in a BoT should be

securely decentralized to make sure that no single participant or set of colluding participants can make up the majority of BoT.

7.1.3 Supporting automatic interactions

BoT is capable of supporting automatic interactions among BoT peers through smart contracts. BoT peers, according to deployment modes and policies of BoT, can deploy smart contracts on BoT, and these smart contracts can be executed automatically when triggered, in order to interact with each other (such as making transactions) among BoT peers involved.

7.2 High-level requirements

[ITU-T Y.4000] provides a group of high-level requirements for IoT, including identification, interoperability, autonomic networking and service provisioning, location, security, privacy, plug and play.

[ITU-T Y.4100] provides a group of common requirements for IoT from the perspectives of application support, service, communication, device, data management, security and privacy protection.

As one type of IoT service platform, BoT supports requirements as specified in [ITU-T Y.4000] and [ITU-T Y.4100]. In addition to these high-level requirements, BoT has the following high-level requirements related to its inherent feature of decentralization:

– It is required to enable all or part of participants of BoT to participate in making consensuses for transactions and storing BoT data according to decentralized management policies;

– It is recommended to support smart contracts to enable BoT peers to automatically perform transactions;

– It is required to support BoT peer management according to decentralized management policies on identification, authentication, authorization and access rights;

– It is required to support BoT data management according to decentralized management policies on identification, storage, transmission, traceability and auditability.

Clauses 7.2.1 to 7.2.4 describe high-level requirements from aspects of different layers of the IoT reference model that is described in clause 8.

7.2.1 Requirements from service and application support aspects

Requirements from aspects of IoT service support and application support include the following:

– It is required to enable BoT peers to deploy, execute and manage smart contracts as per their business purpose and service logic;

– It is required to provide trusted execution environments to perform smart contracts to BoT peers;

– It is recommended to provide adaptive consensus mechanisms to satisfy the different applications to BoT peers;

– It is required to provide access control in BoT to BoT peers, to enable:

  • BoT peers to access services deployed by BoT entities;

  • IoT services to access connected IoT devices in BoT;

  • IoT services to access collected IoT data in BoT;

  • IoT services to control access permissions for services deployed by themselves and for BoT data provided by themselves.

– It is required to enable IoT data to be stored and managed in a decentralized mode in BoT;

– It is required to provide data storage functions including data writing and query services efficiently, safely and stably, and provide data availability by each BoT peer;

– It is required to provide data recording and to maintain mechanisms to guarantee data consistency and security, permit authorized BoT peers to write and query data generated during transaction operations;
– It is recommended to provide support to the peer-to-peer communication network.

7.2.2 Requirements from device aspects

Requirements from device aspects include the following:

– It is required to enable full-fledged IoT devices and gateways to utilize BoT peers to connect to BoT and to participate in and perform transactions, on behalf of themselves or on behalf of the connected constrained IoT devices;
– It is required to enable IoT devices and gateways to be accessed by authorized BoT peers when they are connected to BoT;
– It is required to enable authorized IoT devices and gateways to act as BoT entities to access smart contracts deployed by BoT entities in BoT;
– It is required to enable authorized IoT devices and gateways to act as BoT entities to access BoT data;
– It is recommended to enable IoT devices to act as BoT entities to access other connected IoT devices in BoT;
– It is recommended to enable IoT devices and gateways to control the access permission for themselves and BoT data provided by themselves.

7.2.3 Requirements from security aspects

Requirements from security aspects include the following:

– It is required to provide security attributes such as authentication, authorization, confidentiality, integrity, availability and access control for BoT capabilities in all functional layers;
– It is required to provide applicative cryptographic algorithms related to encryption, digest and digital signature;
– It is required to provide mechanisms to protect the privacy of BoT data and BoT peers.

7.2.4 Requirements from management aspects

Requirements from the aspect of management include the following:

– It is recommended to provide BoT peers capabilities related to regulation, audit, development and operation for BoT;
– It is recommended to enable BoT peers to comply with regulations and rules;
– It is recommended to enable BoT peers to supervise and audit BoT, including the environment, system, availability, disaster recovery, system operation and maintenance, and compliance;
– It is recommended to provide management of BoT peer implementation including performance and availability;
– It is required to enable BoT peers to store, access and manage (e.g., add, delete, update, search) BoT data in a decentralized mode;
– It is required to enable BoT peers to deploy, execute and manage smart contracts in a decentralized mode.

8 BoT-related common capabilities of the IoT reference model

The IoT reference model as defined in [ITU-T Y.4000] is applicable to BoT. Considering the decentralization characteristics and trust mechanisms of BoT, there are some special BoT-related common capabilities of the IoT reference model (see Figure 8-1).

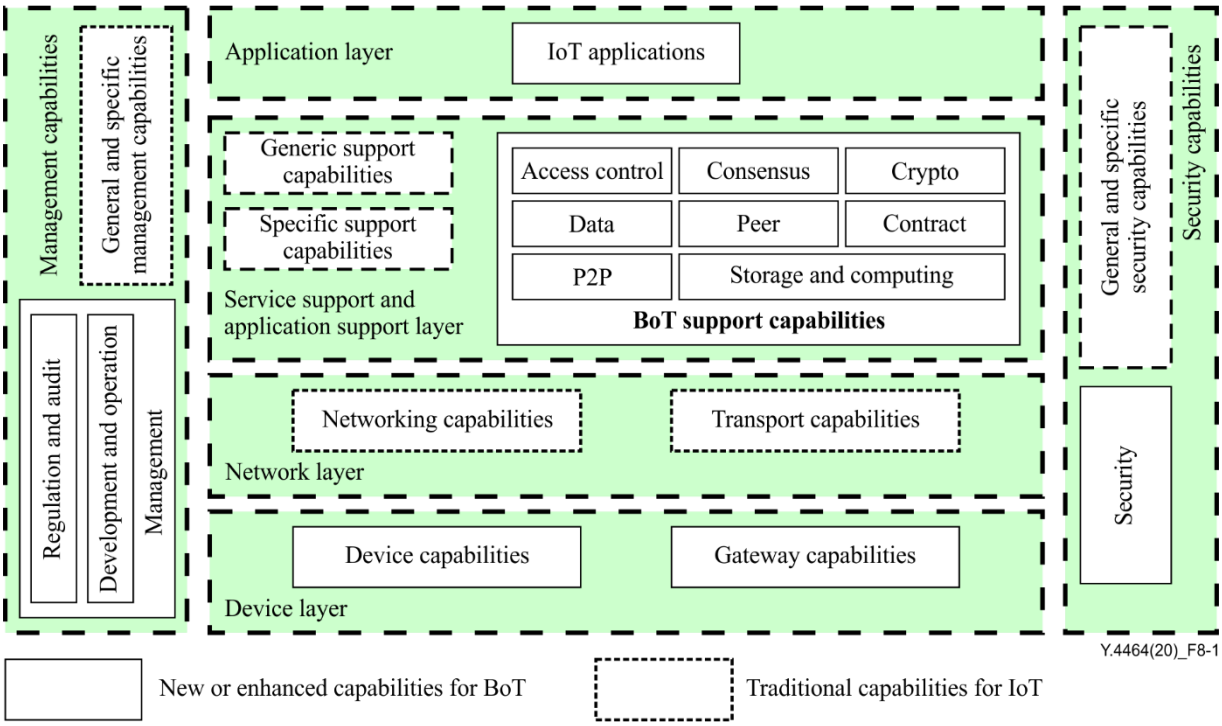

Figure 8-1 – BoT-related common capabilities of the IoT reference model

This clause describes BoT-related common capabilities of the IoT reference model. These new capabilities are enhancements to the capabilities as described in clause 8 of [ITU-T Y.4000].

8.1 Application layer

The application layer contains IoT applications.

The dApps and BoT peers can be deployed on BoT entities (e.g., IoT devices, IoT gateways) or on the cloud (e.g., IoT servers).

8.2 Service support and application support layer

For this layer, [ITU-T Y.4000] provides descriptions of two groups of capabilities: generic support capabilities and specific support capabilities.

In addition to the generic and specific support capabilities as defined in [ITU-T Y.4000], BoT needs a new group of BoT support capabilities in this layer to include:

– Access control

  A BoT works in a decentralized mode, operated by all or part of the participants. It needs the capabilities related to access control for BoT, including user account-related information query, ledger-related information query, transaction processing request committing, interface access rights management, etc.

– Consensus

  Different algorithm mechanisms for achieving consensus shall be provided to meet the different needs of applications for BoT. Consensus-related capabilities include: supporting multiple BoT peers to participate in the consensus and confirmation process, supporting

independent BoT peers to validate the relevant information transformed in BoT, preventing any independent BoT peer to record or modify information without other BoT peer confirmation, and possessing a certain fault tolerance capability (including un-malicious failure such as BoT peer physical or malicious failure, and communication malfunction etc.).

– Crypto

  Crypto-related capabilities for BoT include groups of mathematical processes such as encryption, decryption, digest, digital signature, etc.

– Data

  Data-control-related capabilities for BoT include: data residing in BoT peer distribution and exchange, logic validation before consensus and result calculation after consensus, multi-signature permission control to specific transaction processing, and logic execution based on smart contract.

  Data-record-related capabilities for BoT include: persistent storage of BoT data, complete data record among multiple BoT peers, genuine data record to authorized participants, ensuring data consistency among the records of each BoT peer.

– Peer

  Peer-related capabilities for BoT include peer information query, peer start-up and shutdown control, peer configuration, peer status monitoring, peer authorization management.

– Contract

  Contract-related capabilities are optional for BoT and include contract development and running environment, contract storage environment, and contract triggering and terminating mechanisms.

– Storage and computing

  Capabilities of storage for BoT provide storage and query functions of BoT data and BoT transaction information produced in the operation processes. BoT shall be capable of being deployed and utilized by each BoT peer for storage and query information in an effective, secure, and steady way.

  Capabilities of computing for BoT provide the running and computing environments including container, virtual machine (VM) and cloud technologies which can be applied by each BoT peer.

– P2P

  Peer-to-peer (P2P) capabilities are needed for BoT and include: efficient and secure communication between BoT peers, capable of multicast based on P2P communication abilities, and recognition to dynamic addition or reduction among BoT peers.

Specific support capabilities need to be enhanced to provide capabilities for BoT via specific application programming interfaces (APIs).

8.3 Network layer

[ITU-T Y.4000] provides descriptions of two groups of capabilities, networking capabilities and transportation capabilities.

BoT peers usually support P2P communication technologies. BoT is network-independent.

8.4 Device layer

[ITU-T Y.4000] provides descriptions of two groups of capabilities, device capabilities and gateway capabilities.

IoT gateways and full-fledged IoT devices can host dApps and BoT peers, with which they can participate in a BoT.

Constrained IoT devices can be connected to a BoT through IoT gateways or full-fledged IoT devices.

8.5 Management capabilities

[ITU-T Y.4000] provides descriptions of two groups of capabilities, generic management capabilities and specific management capabilities in this layer for traditional IoT service platforms.

In addition to the management capabilities defined in [ITU-T Y.4000], management capabilities for BoT include:

– Regulation and audit

  Regulation capability for BoT ensures tamper-resistant, traceability and inspectability by technical means of access control, supervision, tracing, etc.

  Service performance related BoT data including records and logs of the operating environment condition need to be collected and maintained in a secure way.

  Audit records, audit process and result information for BoT also need to be collected and maintained for audit activity to avoid information leakage.

– Development and operation

  Development capability for BoT supports developer activities, including service component development, test management and service publishing, etc.

  Operation capability for BoT supports abnormal event management, service delivery and deployment, cross chain service management, etc.

8.6 Security capabilities

[ITU-T Y.4000] provides descriptions of two groups of capabilities, generic security capabilities and specific security capabilities for traditional IoT service platforms.

BoTs can fully utilize the security capabilities for the traditional IoT service platforms to enhance their security capabilities, especially in private and consortium BoTs.

In addition, since BoTs work in decentralized and low-trust environments, they need more security-related capabilities for P2P communications, decentralized data storages, crowding consensuses, smart contracts, access permissions and other essential functions.

9 Common functionalities of BoT

This clause provides common functionalities of BoT that correspond to the requirements listed in clause 7.2. The main common functionalities include:

– supporting IoT services to be deployed on BoT;

– supporting IoT devices and IoT services to access services deployed on BoT;

– supporting IoT services to access IoT devices through BoT;

– collecting IoT data from IoT services and IoT devices;

– supporting IoT services and IoT devices to access IoT data on BoT.

There are several common sub-functionalities of BoT, as listed below and described in clauses 9.1 and 9.2, which are usually utilized by other common functionalities:

– crowding consensus on BoT;

– decentralized storage of BoT;

– deployment and execution of smart contracts on BoT.

9.1 Supporting crowding consensus and decentralized storage

In a traditional IoT service platform, the consensuses for decision-making are made by the service platform and the requesting entities (such as IoT devices and IoT servers), and the corresponding data are stored in the service platform or in the requesting entities.

In a BoT, the consensuses for decision-making and trust-making are made by all or part of the authorized participants (BoT peers) of BoT, and the corresponding data are stored by all or part of the authorized participants.

Clause 10.1 provides general procedures of crowding consensus and decentralized storage of BoT. In BoT, crowding consensus and decentralized storage are fundamental sub-procedures. Almost all of the general procedures of BoT include these two sub-procedures.

NOTE – As for a constrained IoT device, if it does not have ability to host dApp and BoT peers, BoT entity (such as IoT gateway, full-fledged IoT device) that it connects to can participate in BoT and make decisions on its behalf.

9.2 Supporting automatic transactions with smart contracts

BoT entities can deploy smart contracts on BoT. When a smart contract is invoked by a BoT entity to make a transaction, the smart contract can be automatically and securely executed by BoT peers that participate in the crowding consensus and decision-making.

Clause 10.2 provides general procedures of deployment and execution of smart contracts on BoT. BoT entities usually use smart contracts to deploy services on BoT and these smart contracts are performed on BoT.

9.3 Supporting IoT services to be deployed and performed in BoT

An IoT service may be deployed on BoT by deploying smart contracts. There are two major approaches to deploy services on BoT, as follows:

1) IoT service is performed on BoT: In this approach, the deployed smart contracts represent the IoT services, and BoT peers can obtain these services through execution of the smart contracts on BoT.
2) IoT service is performed outside of BoT: In this approach, the deployed smart contracts are not used to perform services, but they are used to provide supportive services (e.g., authentication, authorization, charging and recording) for the IoT service.

Clause 10.3 provides general procedures of an IoT service to deploy services on BoT.

If the IoT service is performed on BoT, the deployed smart contract represents the IoT service. Clause 10.4 provides general procedures for one IoT device to access services deployed by one IoT service, and general procedures of these services are performed on BoT.

However, if the IoT service will be performed outside of BoT, the deployed smart contract does not represent the IoT service. Through the deployed smart contract(s), the IoT service can write data into BoT, or authenticate the IoT devices. Clause 10.5 provides general procedures for one IoT device to access services deployed by one IoT service, and general procedures of the services are performed outside of BoT.

NOTE – Besides IoT services, other types of BoT entities (such as full-fledged IoT devices and IoT gateways) can deploy their services on BoT according to their capabilities and the policies of BoT.

9.4 Supporting IoT services to access IoT devices through BoT

BoT is enabled to allow IoT devices to connect to it, and to enable IoT services access to the connected IoT devices.

Clause 10.6 provides general procedures for IoT devices to connect to BoT, and general procedures for IoT services to access the connected IoT devices.

Clause 10.6 also provides general procedures for constrained IoT devices to connect to BoT through an IoT gateway, and general procedures for the IoT services to access the connected IoT devices.

9.5 Supporting IoT services and IoT devices to access IoT data in BoT

BoT can collect IoT data from IoT services and IoT devices, and can enable IoT services and IoT devices to access the collected IoT data.

Clause 10.7 provides general procedures for BoT to collect IoT data, and general procedures of supporting IoT services to access the IoT data in BoT.

10 General procedures of BoT

Without loss of generality, it is assumed that a BoT consists of BoT peers A, B1 to Bk (k is a positive integer), C, D, E and F. It is also assumed there are five BoT entities, including:

1) BoT entity A, is an IoT server that hosts IoT service A and BoT peer A;
2) BoT entity C, is an IoT device that hosts dApp C and BoT peer C;
3) BoT entity D, hosts dApp D and BoT peer D;
4) BoT entity E, hosts dApp E and BoT peer E;
5) BoT entity F, is an IoT gateway that hosts dApp F and BoT peer F.

For simplicity of the descriptions, the deployment mode (see Appendix I) of BoT is not distinguished here.

10.1 Crowding consensus and decentralized storage of BoT

Figure 10-1 depicts the main procedures for crowding consensus and decentralized storage of BoT.

The main procedures for crowding consensus and decentralized storage of BoT are as follows:

– When BoT entity D wants to perform a transaction with BoT entity E, dApp D sends a request to BoT peer D (step 1). Next, BoT peer D and BoT peer E cooperate with all or part of the other BoT peers (such as BoT peers B1 to Bk) to make consensuses for the requested transaction (step 2);
– After the transaction is accepted and performed, and when dApp D requests to store data in BoT (step 3), all or part of BoT peers duplicate and store the transaction data (BoT data) with decentralized mode (step 4);
– Furthermore, when a BoT peer participated in a consensus-making operation or decentralized storage if necessary, it may notify the corresponding dApp; for example, BoT peer E notifies dApp E (step 5).

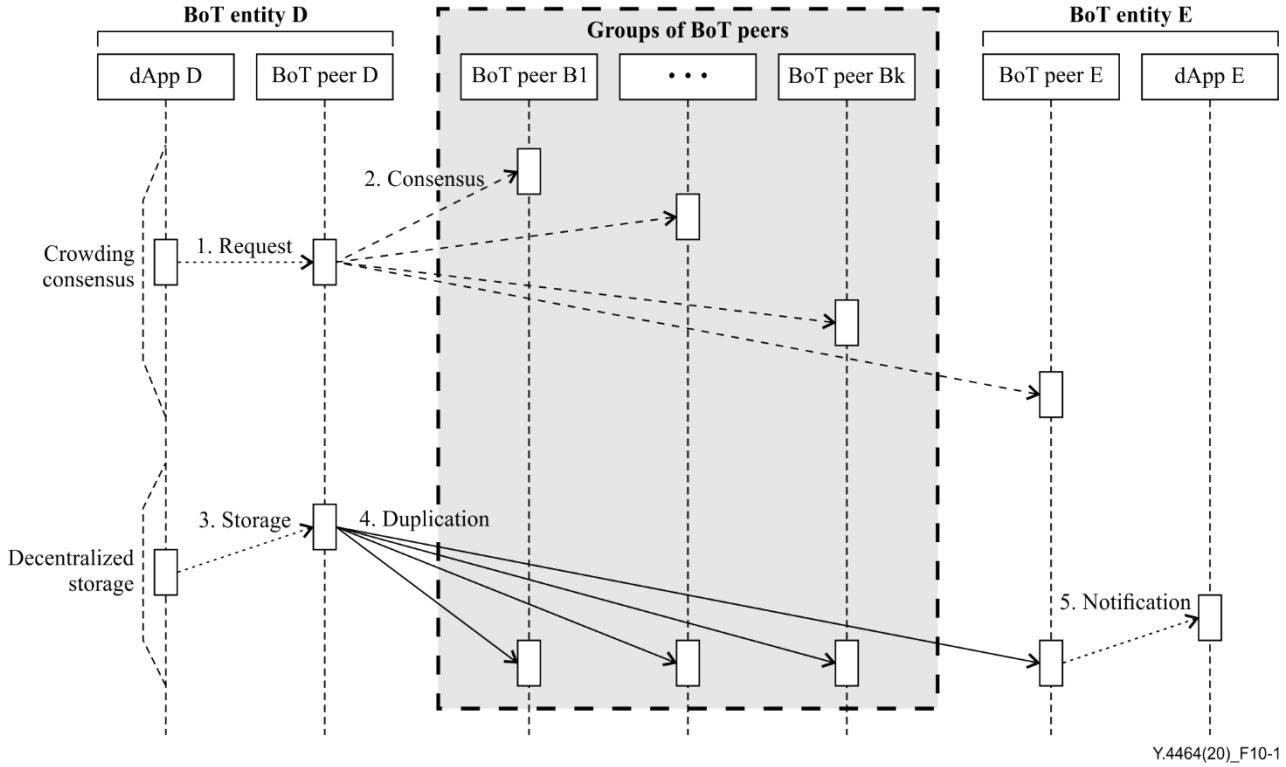

Figure 10-1 – Crowding consensus and decentralized storage of BoT

10.2 Deployment and execution of smart contracts on BoT

Figure 10-2 depicts the main procedures of deployment and execution of smart contracts on BoT.

The main procedures for BoT entity A to deploy smart contracts are as follows:

– Step 1: IoT service A requests to deploy one smart contract on BoT. The request is sent to the corresponding BoT peer (BoT peer A);

– Step 2: BoT peer A validates the request and initializes the information for consensus (such as, which BoT peers can participate in the consensus-making progresses);

– Step 3: BoT peer A interacts with all or part of other BoT peers to make a consensus for the request of IoT service A;

  If a consensus is not achieved and the request is not accepted, the request may be dropped and the process is ended;

– Steps 4 and 5: BoT peer A requests to deploy a smart contract when the request is accepted. All or part of BoT peers duplicate and store the smart contract respectively;

– Step 6: BoT peer A sends IoT service A the response of the results to request to deploy smart contracts on BoT.

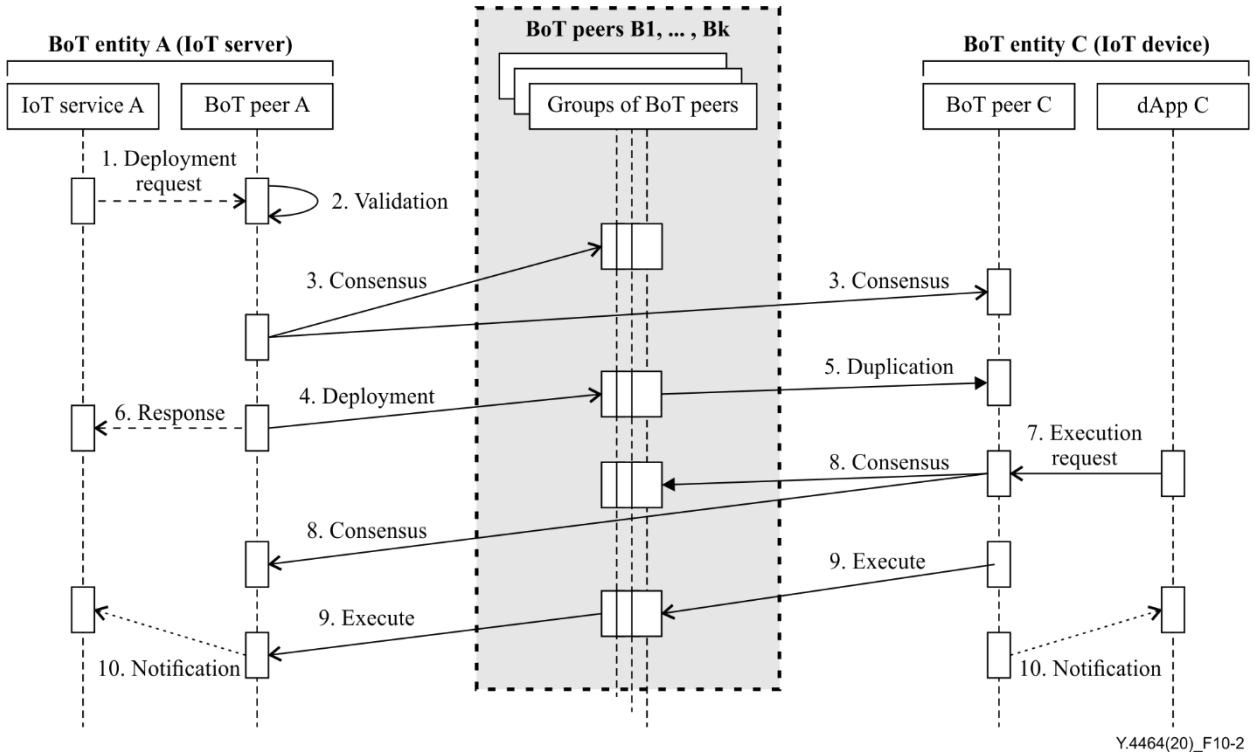

Figure 10-2 – Deployment and execution of smart contracts on BoT

After a smart contract is deployed on BoT, it can be discovered and executed by the authorized BoT entities (such as BoT entity C).

The main procedures for BoT entity C (an IoT device) to execute a deployed smart contract are as follows:

– Step 7: dApp C sends BoT peer C the request to execute a deployed smart contract;

– Step 8: Upon receiving the request, BoT peer C interacts with all or part of BoT peers to make a consensus if supporting the request;

– Step 9: When the request is accepted by BoT, BoT peer C requests to execute the smart contract on BoT;

– Step 10: After execution of the smart contract, BoT peers can send notifications to their corresponding IoT applications (e.g., dApp C) and services (e.g., IoT service A) respectively.

10.3 Deployment of IoT services on BoT

Figure 10-3 depicts the main procedures of an IoT server (such as BoT entity A) to deploy services on BoT. BoT entity A deploys smart contract which presents the IoT service A on BoT.

The main procedures for BoT entity A to deploy IoT service are as below:

– Step 1: Same as the step 1 in Figure 10-2;

– Step 2: Same as the step 2 in Figure 10-2;

– Step 3: Same as the step 3 in Figure 10-2;

– Step 4: Same as the step 4 in Figure 10-2;

– Step 5: Same as the step 5 in Figure 10-2;

– Step 6: Same as the step 6 in Figure 10-2;

– Step 7: When the smart contract is deployed on BoT successfully, BoT peers (such as BoT peer A, BoT peer C) notify their corresponding IoT applications (e.g., dApp C) and services (e.g., IoT service A) respectively.

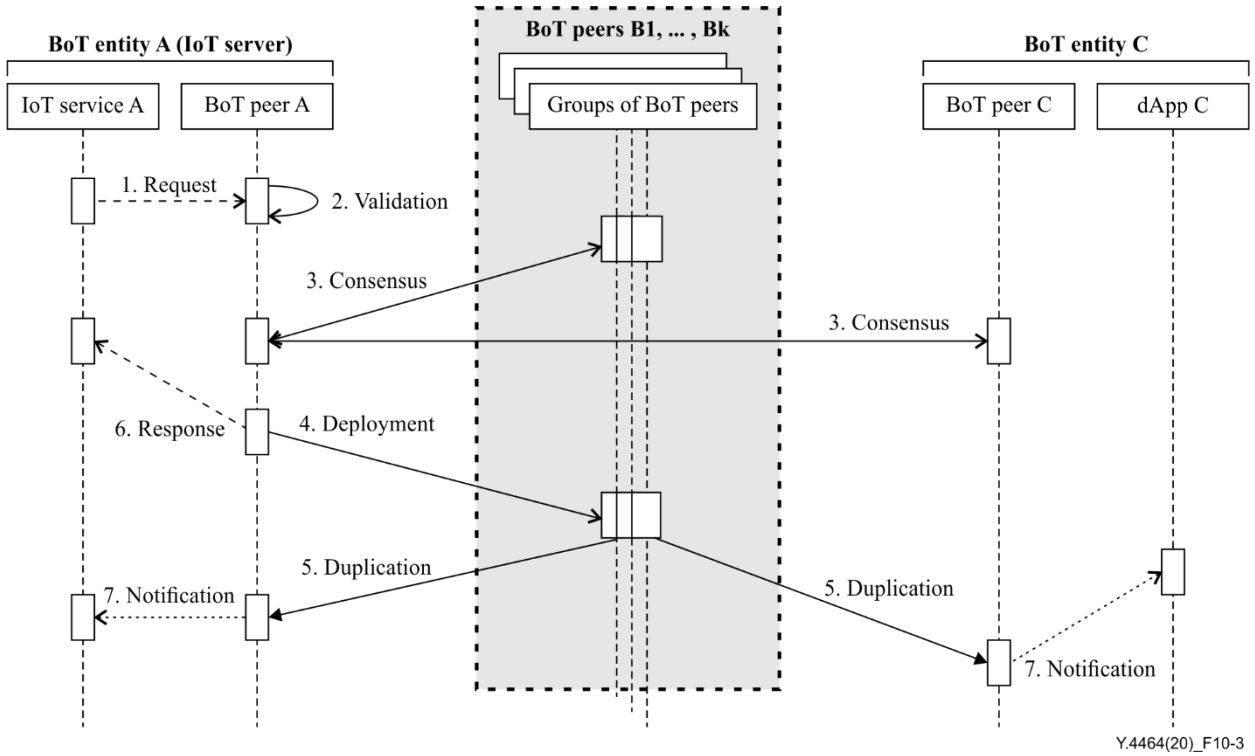

Figure 10-3 – Deployment of IoT services on BoT

10.4 Access to IoT services as performed on BoT

Figure 10-4 depicts the main procedures of an IoT device to access IoT services deployed on BoT.

The main procedures for IoT device to access to IoT service A through BoT are as below:

– Step 1: IoT device requests, through dApp C, to access the deployed IoT service. The request is sent to the corresponding BoT peer C;

– Step 2: All or part of BoT peers participate in the consensus for the request of the IoT device;

– Step 3: When BoT peer A participates in the consensus, it can interact with IoT service A to validate the request;

  The consensus is to check if the IoT device can access the target IoT service. If a consensus is not achieved or the request is not accepted, the request will be dropped and the progress is ended.

– Step 4: BoT peer A and BoT peer C interact with each other to execute the target smart contracts. All or part of BoT peers can participate in the procedures;

– Step 5: After the target smart contracts are executed, BoT peer C sends the results of the execution to dApp C;

– Steps 6, 7, and 8: After access to the IoT service, the IoT device can write the record in BoT, and then notify the results to corresponding IoT applications (e.g., dApp C) and services (e.g., IoT service A) respectively.

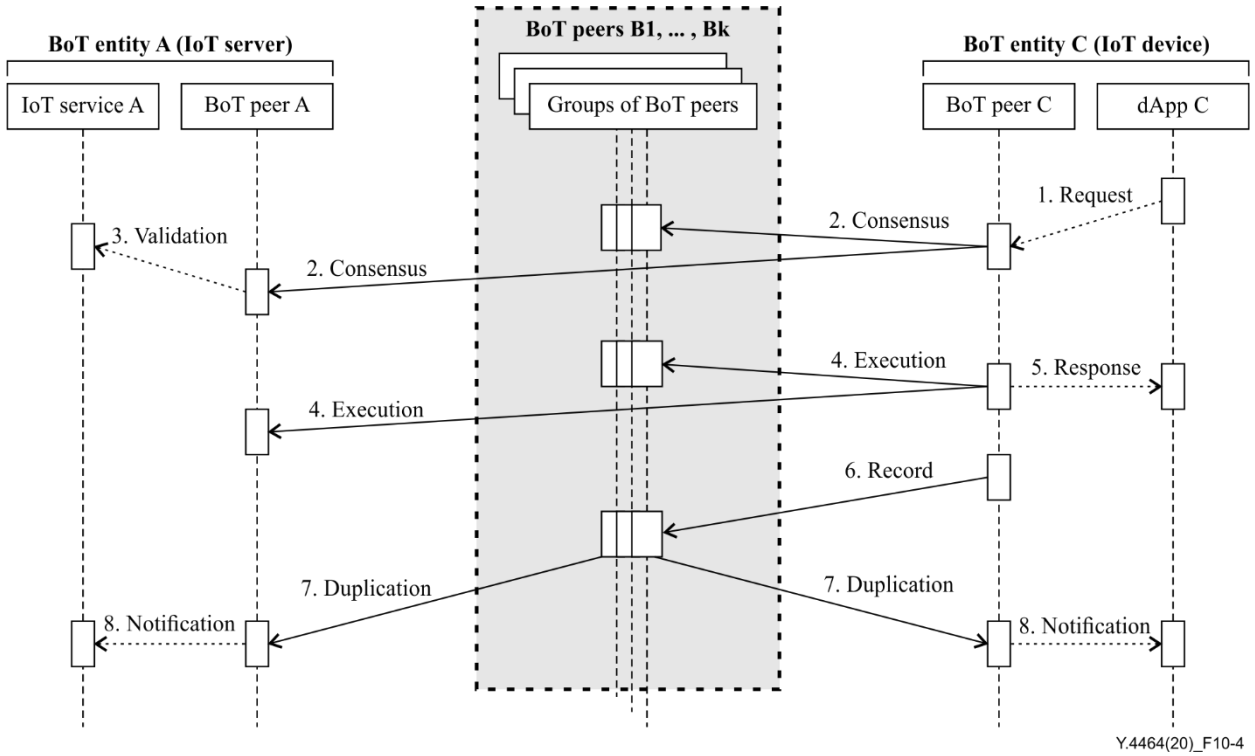

Figure 10-4 – Access to IoT services as performed on BoT

10.5 Access to IoT services as performed outside of BoT

Figure 10-5 depicts the main procedures for an IoT device to access an IoT service as performed outside of BoT.

The main procedures for an IoT device to access an IoT service as performed outside of BoT are as follows:

– Step 1: Same as step 1 in Figure 10-4;

– Step 2: Same as step 2 in Figure 10-4;

– Step 3: Same as step 3 in Figure 10-4;

– Step 4: dApp C accesses IoT service A directly, not through BoT;

  After interactions between dApp C and IoT service A, dApp C or IoT service A can request to write record to BoT;

– Step 5: Same as the step 6 in Figure 10-4;

– Step 6: Same as the step 7 in Figure 10-4;

– Step 7: Same as the step 8 in Figure 10-4.

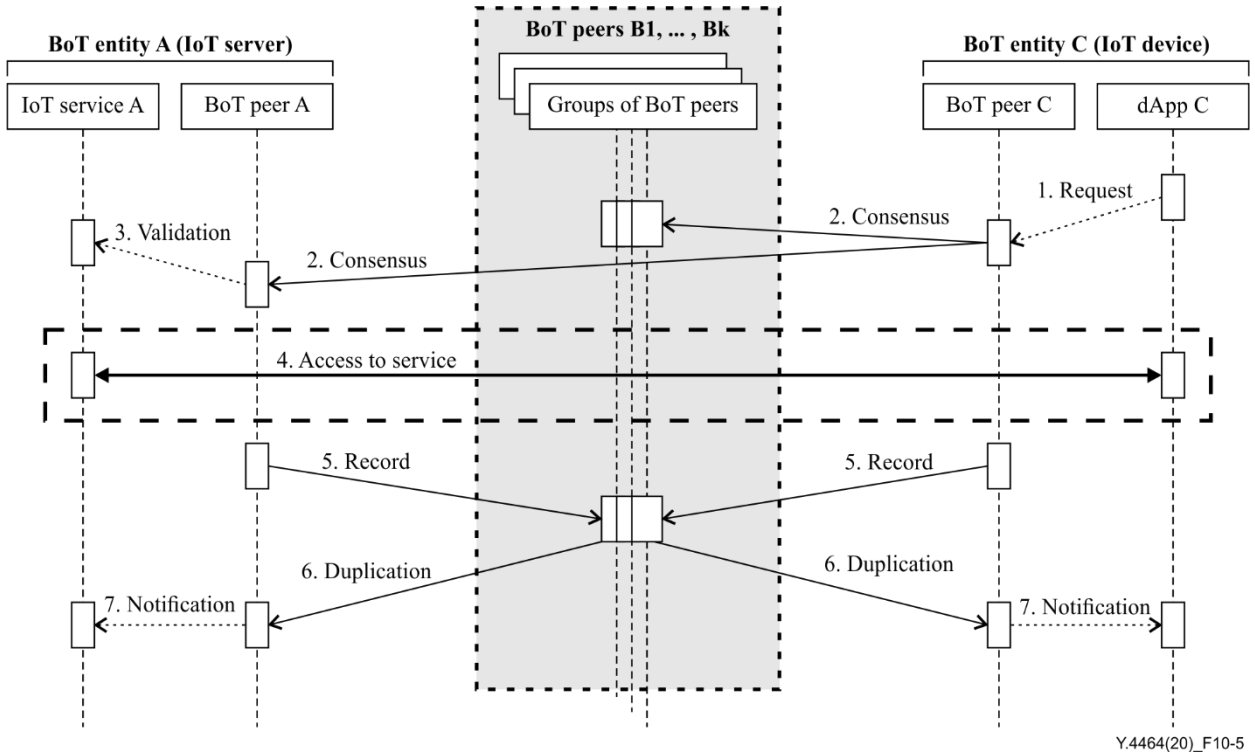

Figure 10-5 – Access to IoT services as performed outside of BoT

10.6 Connection of (constrained) IoT devices to BoT

Figure 10-6 (a) depicts the main procedures of an IoT device to connect to BoT, and of an IoT service to access to the connected IoT device.

The main procedures for an IoT device to connect to BoT and for an IoT service to access the connected IoT device are as follows:

– Step 1: dApp C, on behalf of the IoT device, registers to BoT. The request is sent to corresponding BoT peer C;

– Step 2: BoT peer C and all or part of other BoT peers make a consensus on the request of dApp C;

  If consensus is achieved and the request is accepted by BoT, the connected information about the IoT device will be recorded in BoT.

  When the status of an IoT device is changed, it can update registration information to BoT.

  When an IoT device is registered in BoT, other BoT entities (such as BoT peer A) can discover and subscribe the information of the registered IoT device.

– Step 3: IoT service A sends an access request to BoT peer A;

– Steps 4 and 5: BoT peer A interacts with all or part of BoT peers to make a consensus. In this progress, dApp C validates the access request;

– Step 6: If consensus is achieved and the access request is accepted, IoT service A can directly access the IoT device through dApp C;

– Steps 7 and 8: After the interactions between dApp C and IoT service A, dApp C or IoT service A can request to write record to BoT.

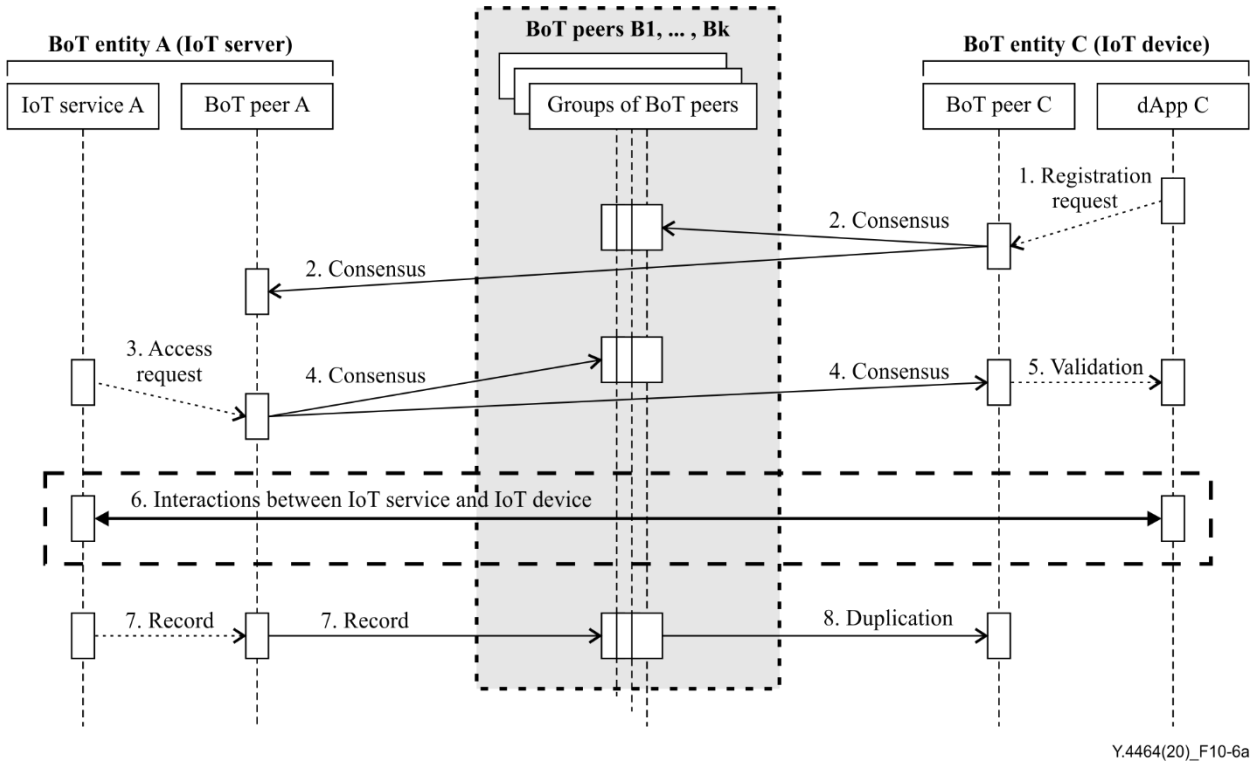

Figure 10-6 (a) – Connection of IoT devices to BoT

Figure 10-6 (b) depicts the main procedures of a constrained IoT device to connect to BoT through an IoT gateway, and of an IoT service to access the constrained IoT device through BoT and the IoT gateway.

The main procedures for a constrained IoT device to connect to BoT and for the IoT service to access the constrained IoT device are as follows:

– Step 0: The constrained IoT device first connects to the IoT gateway;

– Steps 1 and 2: Same as steps 1 and 2 in Figure 10-6 (a) respectively;

– Steps 3, 4, 5 and 6: Same as the steps 3, 4, 5 and 6 in Figure 10-6 (a) respectively;

– Step 6.1: The IoT gateway interacts with the constrained IoT device according to the requests of the IoT service and forwards the results to the IoT service;

– Steps 7 and 8: Same as step 7 and 8 in Figure 10-6 (a) respectively.

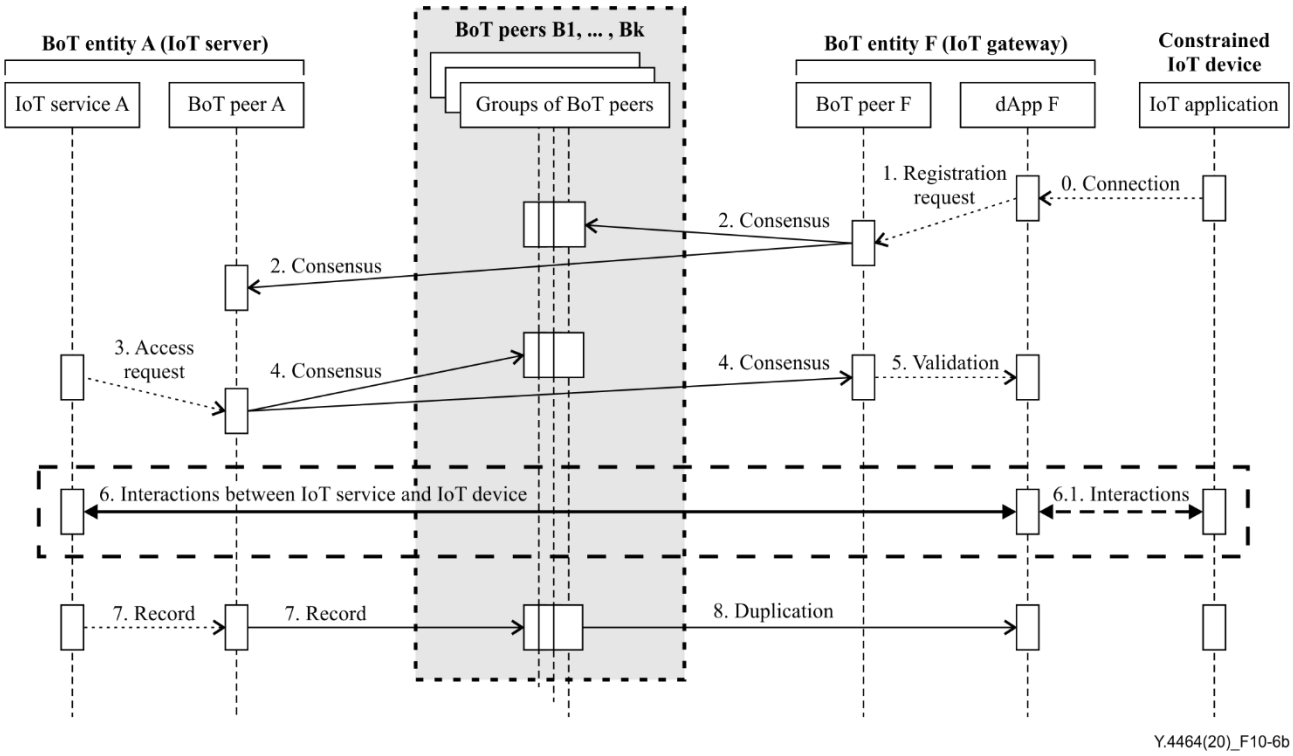

Figure 10-6 (b) – Connection of constrained IoT devices to BoT

10.7 Collection and access to IoT data on BoT

Figure 10-7 depicts the main procedures of BoT to collect IoT data from an IoT device and to support an IoT service to access the collected IoT data.

The main procedures, for BoT to collect IoT data from an IoT device and to support an IoT service to access the collected IoT data, are as follows:

– Step 1: dApp C, on behalf of the IoT device, uploads or updates IoT data to BoT. The request is sent to corresponding BoT peer C;

– Step 2: BoT peer C and all or part of other BoT peers make a consensus on the request from dApp C;

  If consensus is achieved and the request is accepted by BoT, the IoT data will be stored in BoT.

  When IoT data is collected and stored in BoT, all or part of BoT entities may discover and subscribe the collected IoT data.

– Step 3: IoT service A sends access data request to BoT peer A;

– Step 4: BoT peer A interacts with all or part of BoT peers to make a consensus on this request;

  If consensus is achieved and the access request is accepted, IoT service A can access the collected IoT data.

– Steps 5, 6 and 7: After the access, BoT peer A can send record for accessing data to BoT on behalf of the IoT service; and other BoT entities (such as IoT device) can get notifications about the access request if necessary.

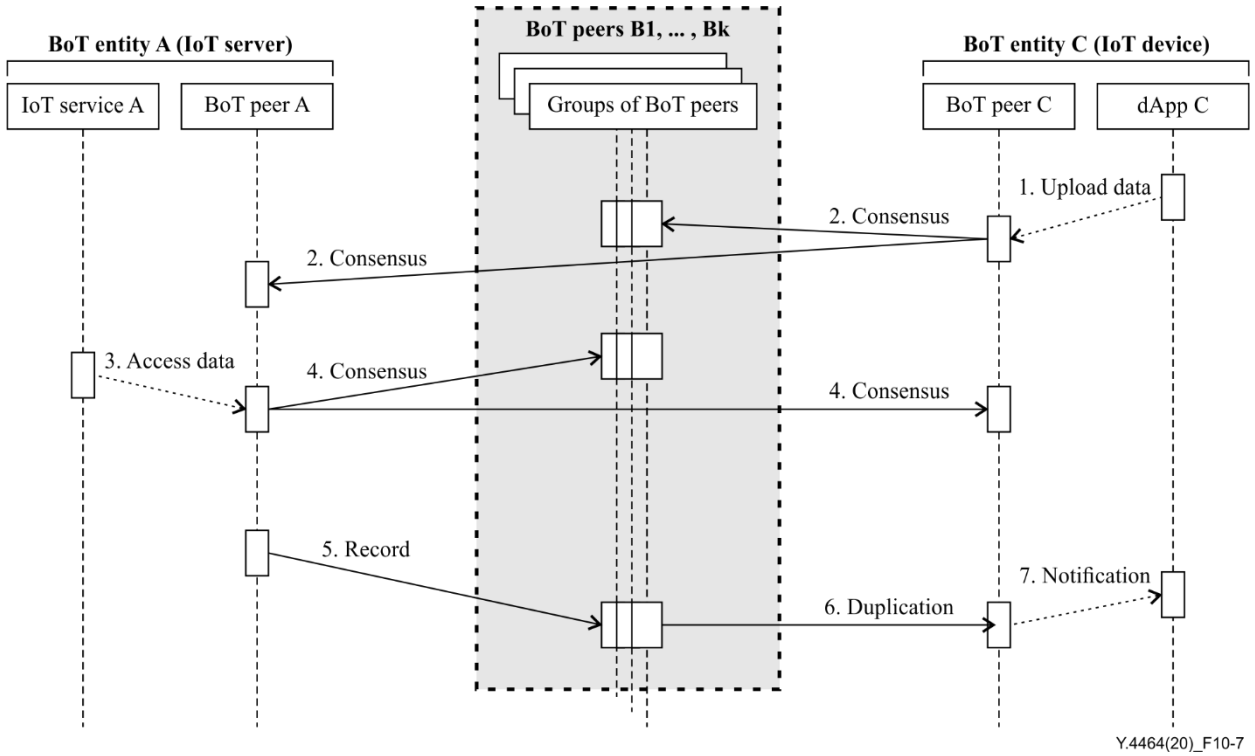

Figure 10-7 – Collection and access to IoT data on BoT

NOTE – The authorized IoT services, IoT gateway and IoT devices can access the IoT data collected and stored on BoT.

Appendix I

Deployment modes of BoT

(This appendix does not form an integral part of this Recommendation.)

According to patterns of deployment and operation, BoT can be of three types: public BoT, consortium BoT and private BoT:

– **Public BoT**: Public BoT is deployed and operated by all participants. In a public BoT, all participants have the same right to participate in the activities of the public BoT, such as making transactions, making consensuses, maintaining BoT data;

– **Consortium BoT**: Consortium BoT is deployed and operated by a consortium (i.e., groups of organizations). The distinction of a consortium BoT is primarily its method of making consensus. The members of the consortium decide which participants on BoT have authority to deploy contracts and make transactions and access BoT data;

– **Private BoT**: Private BoT is deployed and operated by one or multiple private organization(s). Private BoT is opposite to public BoT. Private BoT is usually not open to members outside of the private organization(s).

Table I.1 lists some key characteristics of the three deployment modes for BoTs.

Table I.1 – Key characteristics of three deployment modes of BoT

	Public BoT	**Consortium BoT**	**Private BoT**
Communication	Supporting P2P and traditional communications.		
Storage	BoT data are stored and maintained by all BoT peers.	BoT data are stored and maintained by authorized BoT peers of the consortium.	Customized
Consensus	All BoT peers have same rights to participate in making consensus.	Only the authorized BoT peers of the consortium can participate in making consensus.	Customized
Transaction	Every BoT peer can do (or help do) transactions on BoT.	Only authorized BoT peers can do (or help do) transactions on BoT.	Customized
Smart contract	All support smart contracts		
Crypto	BoT data, communications and transactions are encrypted in BoT.		
Permission	Do not need access permission	Need access permission	Customized
Wallet	Every participant has a wallet to store digital tokens	Do not need	Customized

These three deployment modes of BoTs are not completely distinct and separate. As a reference, Figure I.1 provides guidance to choose a deployment mode for a BoT.

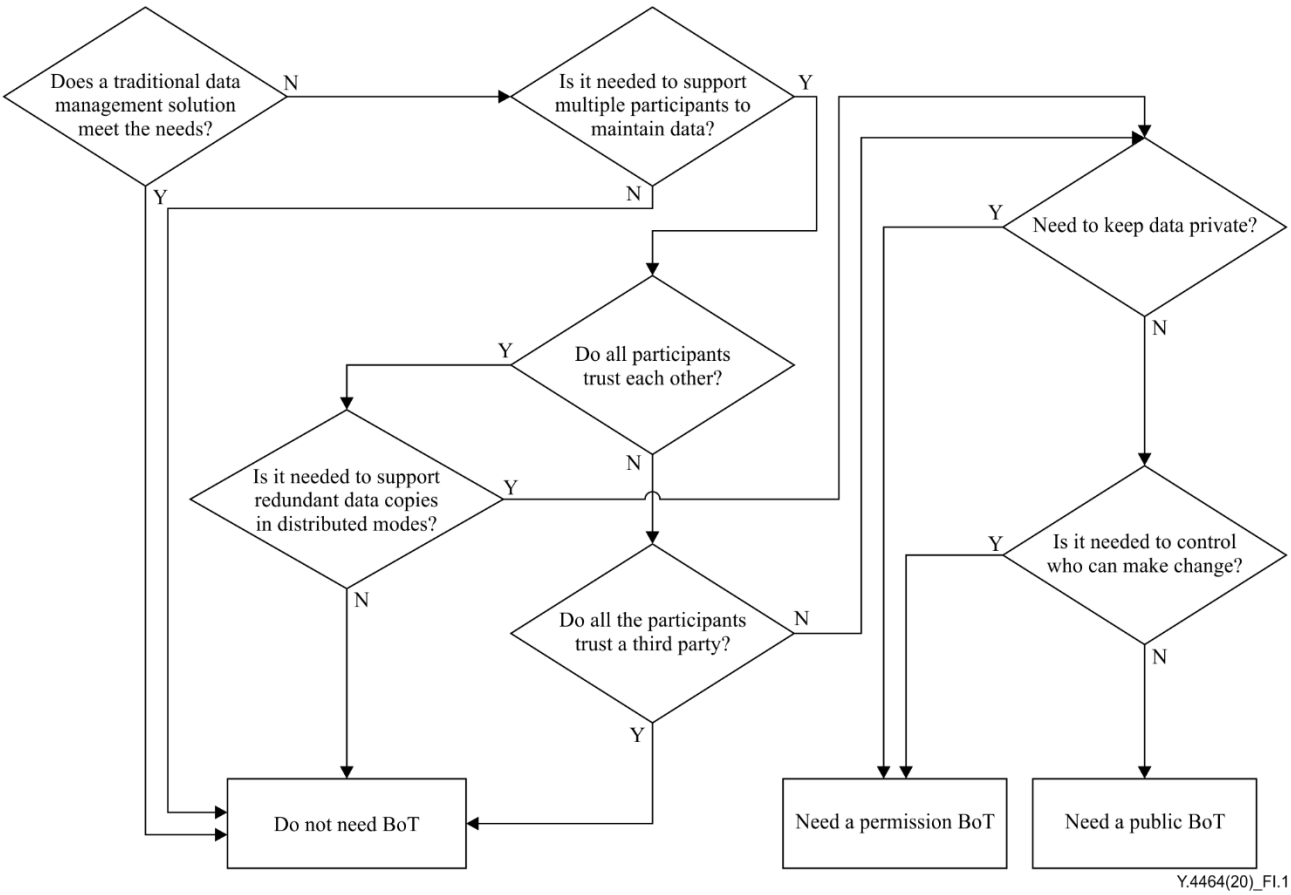

Figure I.1 – Selection of BoT deployment mode

NOTE – Usually, private BoTs and consortium BoTs are permission BoTs. In practice, the detailed deployment mode of BoT is decided by its operation requirements.

Appendix II

Use cases for BoT

(This appendix does not form an integral part of this Recommendation.)

This appendix provides some use cases to illustrate the concept of BoT.

II.1 Use case: Using BoT to enhance supply chains for trust productions

The supply chains represent all the links involved in producing and distributing goods, from raw materials to the finished products that go into the possessions of the consumers. Figure II.1 shows supply chains for bikes, which include links to raw materials suppliers, semi-manufactures, bike manufactures, business channels, retailers and consumers, etc.

Currently, traditional supply chains usually span many separate stages and geographical locations, which makes it very hard to trace part or all of the production processes. Under the traditional supply chains, the productions processes lack transparency, traceability, security and trust.

BoT has the potential to revolutionize the traditional supply chains and improve the trust to the ways for producing, marketing, purchasing and consuming goods. BoT-related supply chains can achieve transparency, traceability and security, and can go a long way toward making our economies safer and much more reliable by promoting trust and honesty, and preventing the implementation of questionable practices.

With the production of bikes as an example, which is shown in Figure II.1, relevant traditional supply chains can be upgraded as BoTs. Those supply chains connect each other. Any of the bodies in the supply chains – in certain authorized circumstances and agreements – can trace the production processes of the bikes. That makes transparency, traceability and security, and trust.

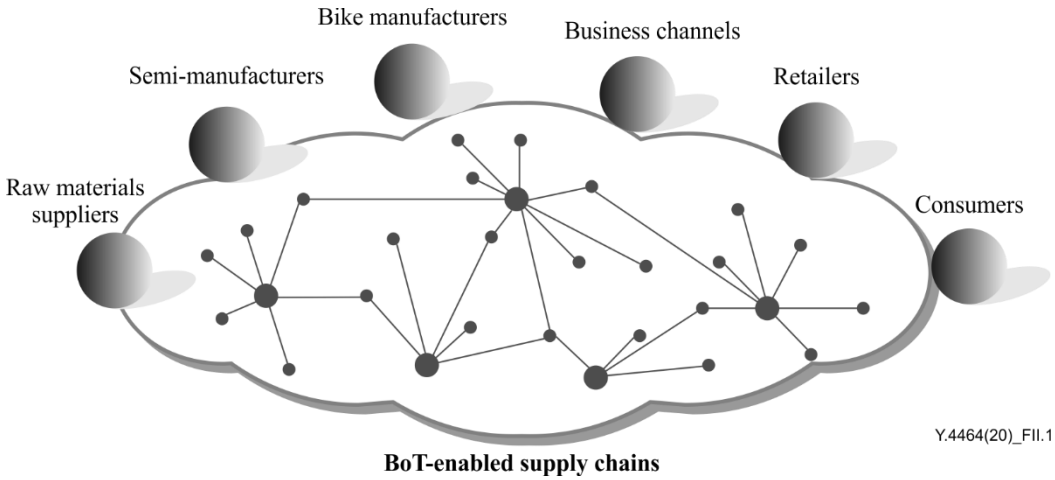

Figure II.1 – Using BoT to enhance supply chains for trust productions

II.2 Use case: Using BoT to mitigate DDoS attacks from hijacked unsecure IoT devices

It is a big problem that many unsecure IoT devices (e.g., home cameras, smart lights, road monitors) are easy to be hijacked silently and become zombies. Those hijacked IoT devices are easily controlled by malware to be used for distributed denial of service (DDoS) attacks on specific services. To remedy this problem the obvious answer is to detect and prohibit the hijacked IoT devices from connecting to the communication networks, and to cut the connections before they access the targeted services.

Most IoT services and unsecure IoT devices are connected to communication networks through IoT gateways. Those IoT gateways can form a BoT and jointly perform those types of prohibitions.

As shown in Figure II.2, IoT devices and IoT services connect to communication networks through their IoT gateways (such as service gateways, home gateways, traffic gateways). Those IoT gateways connect to each other through communication networks and establish a consortium BoT, and deploy smart contracts to validate, record and cancel DDoS attacks related information.

For example, home smart devices can connect to a communication network through the home gateway C shown in Figure II.2. Let's suppose one of the home smart devices (e.g., the smart light) is hijacked and is controlled to perform DDoS attacks to IoT service A. If IoT service A validates the attacks, it can notify the service gateway A. The service gateway A can then connect to the home gateway C to make a transaction according to a deployed smart contract. If the attacks are validated, relevant information could be written into BoT. After that, all the IoT gateways in BoT can drop the connection requests from the hijacked smart device (in this case the smart light). When the hijacked smart light is updated and works correctly, home gateway C can make a new transaction to cancel the prohibition, and all the IoT gateways can acknowledge the cancellation and can serve the connection of the smart light.

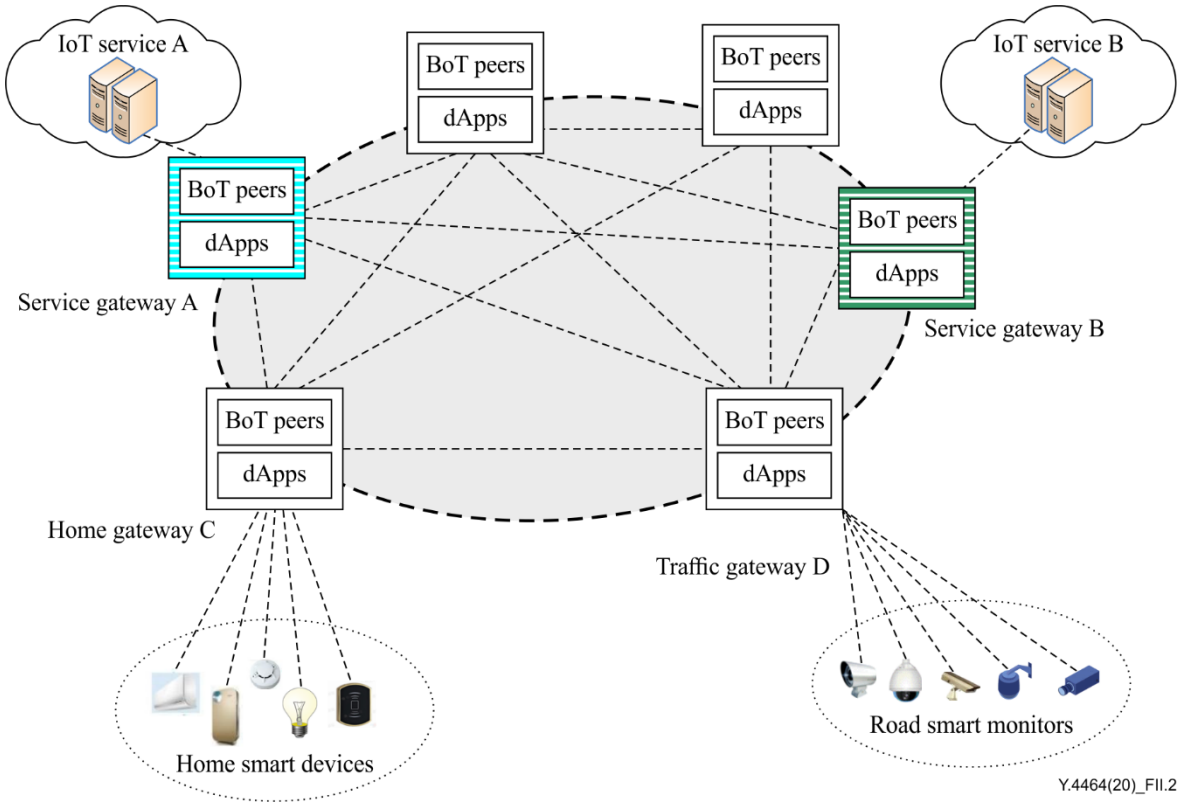

Figure II.2 – Using BoT to mitigate DDoS attacks from hijacked unsecure IoT devices

II.3 Use case: Using BoT to improve ITS for trust data exchanges

Through traditional intelligent transportation systems (ITSs), the vehicles on the road can exchange information of themselves and of their environments (e.g., speed, congestion degree, road humidity) to enhance their navigational abilities. However, there are some issues to be resolved, such as:

1) how to increase service coverage;
2) how to ensure response speed and efficiencies;
3) how to motivate the users of the vehicles to join the activity to exchange information;
4) how to ensure the authenticity and effectiveness of the shared data;
5) how to make the users trust the shared data.

Those issues can be solved or mitigated when the ITSs employ BoT-related technologies. Figure II.3 provides an illustrative example on how to use BoT to enhance the ITSs. ITS provider (such as A to E) and others make an ITS alliance and establish a consortium BoT. Through the consortium BoT the ITS providers coordinate each other to provide ITS services to the vehicles. The vehicles, as common participants, can produce and consume transportation information through the consortium BoT.

First, the consortium BoT helps the ITS providers solve issues 1) and 2) when they provide ITS services individually. Second, the vehicles become producers and/or consumers of the transportation information, the owners of the vehicles can get benefits from producing transportation information. Under correct strategies, the consortium BoT can motivate the users of the vehicles to join the activity of ITS (issue 3). Due to the inherent properties of BoT, issues 4) and 5) can be easily solved.

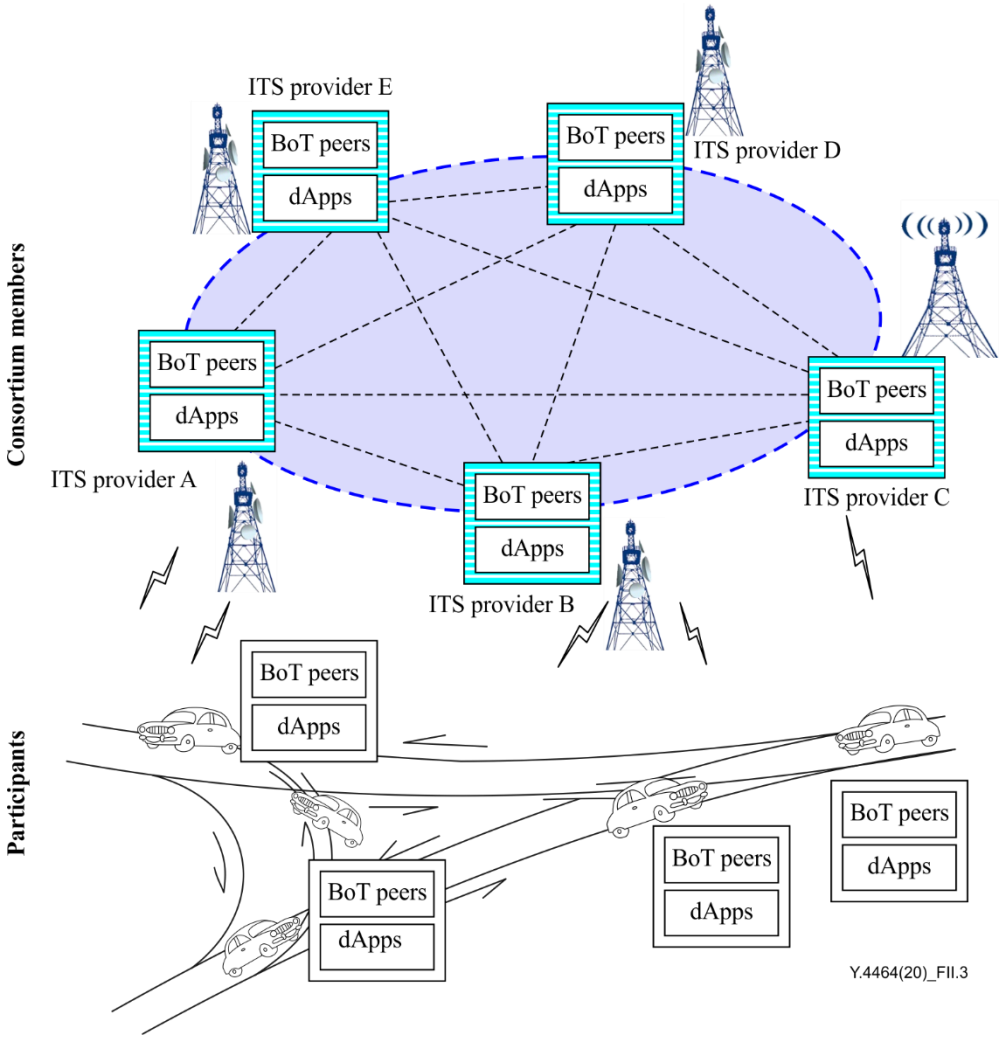

Figure II.3 – Using BoT to enhance ITS for trust data exchanges

II.4 Use case: Using BoT to promote device sociability

Device sociability refers to one type of ability which allows IoT devices to autonomously interact with each other in pre-set conditions.

In the untrusting Internet environment, it is a big issue to establish trust among IoT devices involved when making some social events. Usually, third parties are required to act as trust endorsers. As the number of IoT devices increases, it will become increasingly difficult to coordinate IoT devices to make sociability.

Public BoT may benefit device sociability for IoT devices. BoT can support contracts and transactions. Based on pre-set smart contracts, in BoT, the IoT devices involved directly interact with each other without the need to previously establish trust and without the need of special third parties to endorse the interactions.

For example (see Figure II.4), let us suppose there is a home environment with some home devices (e.g., monitor, printer, washing machine) that are participating in the public BoT. Those home devices can interact with each other and with other IoT devices or services. For example, the homeowner can previously deploy contracts for its washing machine to buy laundry detergents. When needed, based on the pre-set contracts, the washing machine can select a store from which to buy laundry detergents automatically.

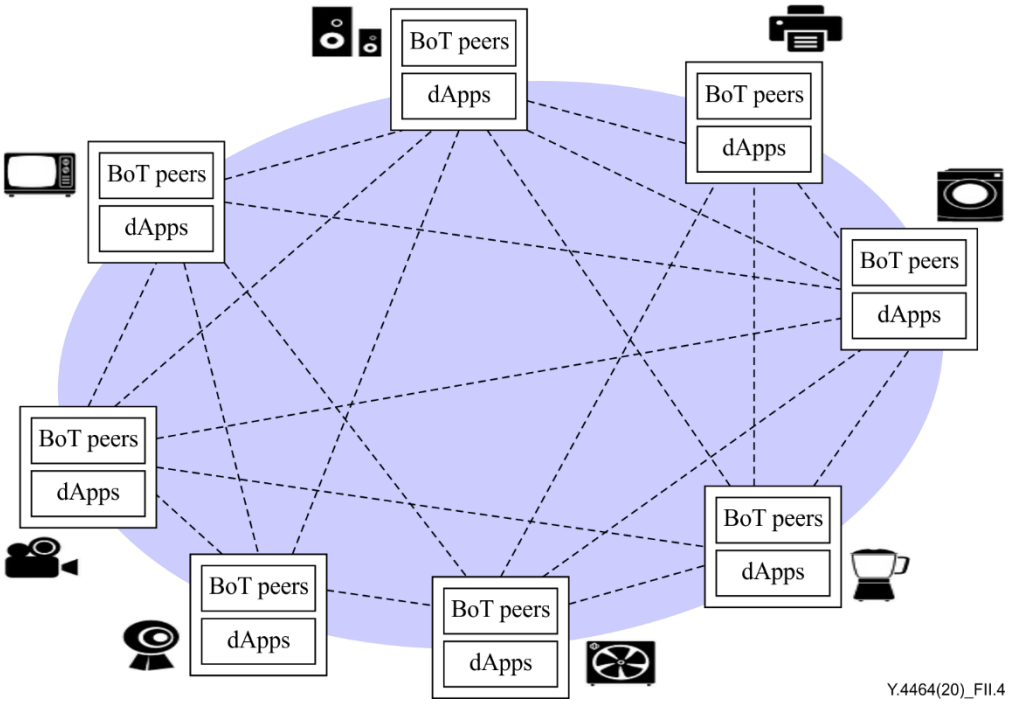

Figure II.4 – Using BoT to promote device sociability

Appendix III

Technical analysis and comparison of BoT

(This appendix does not form an integral part of this Recommendation.)

This appendix provides comparative analysis and investigation on how blockchain compares with alternative approaches for the improvement of IoT services (see Table III.1).

Table III.1 – Comparative analysis of two types of systems

Categories	Non-decentralized solutions	Decentralized solutions (BoT)
Degree of centralization	high	low
System construction	Systems are constructed and maintained by the provider(s).	All participants participate in constructing and maintaining the systems.
Data management	Data are stored by the clients and/or the providers.	All participants can contribute to store and manage the data.
Transaction	Transactions are made by the clients and providers.	All the participants can contribute to transaction-making.
trust	All participants shall trust one or multiple deterministic trust-point(s).	All the (authorized) participants can contribute to trust-making.
Data transparency and reliability	medium	High
System robust	medium	High
Trust dependence	high	Very low
Cost of maintenances	higher	lower

Appendix IV

Business roles and models of BoT

(This appendix does not form an integral part of this Recommendation.)

[ITU-T Y.4000] describes IoT ecosystem, business roles and groups of business models. In view of working on decentralization and crowding decision-making, BoT has some special business models. Figure IV.1 shows two typical business models of BoT.

In the business model A of BoT, a group of players jointly establish and operate the platforms (BoTs), provide applications (dApps and BoT peers) and provide services for themselves. In this business model, those players are providers (for platforms and applications) and are also the application customers. These platforms (BoTs) in this business model usually are public BoTs.

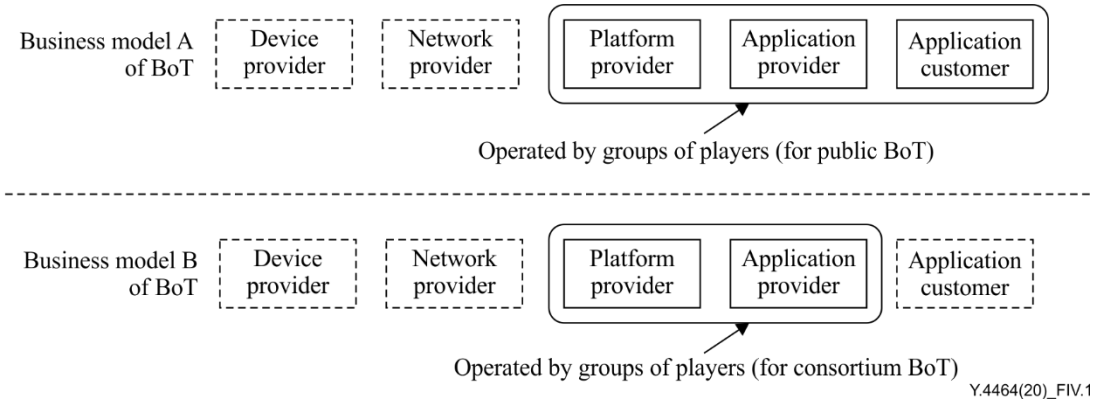

Figure IV.1 – Typical business models of BoT

In the business model B of BoT, a group of players jointly establish and operate the platforms (BoTs), and provide applications (dApps and BoT peers). As opposed to business model A, in this business model those players are usually providers, but not the application customers. These platforms (BoTs) in this business model are usually consortium BoTs.

Bibliography

[b-ITU-T Y.2091] Recommendation ITU-T Y.2091 (2011), *Terms and definitions for next generation networks*.

[b-ITU-T Y.4451] Recommendation ITU-T Y.4451 (2016), *Framework of constrained device networking in the IoT environments*.

[b-FG-DPM TR D3.5] Technical report D3.5 (2019), *Overview of blockchain for supporting IoT and SC&C in DPM aspects*.
http://handle.itu.int/11.1002/pub/812e8ba1-en

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020