TECHNICAL REPORT

ISO/IEC TR 29181-7

First edition
2013-04-15

Information technology — Future Network — Problem statement and requirements —

Part 7: Service composition

Technologies de l'information — Réseaux du futur — Énoncé du problème et exigences —

Partie 7: Composition des services

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

Reference number
ISO/IEC TR 29181-7:2013(E)

© ISO/IEC 2013

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2013

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

© ISO/IEC 2013 – All rights reserved

Contents

Page

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 29181-7 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

ISO/IEC TR 29181 consists of the following parts, under the general title *Information technology — Future Network — Problem statement and requirements*:

— *Part 1: Overall aspects*

— *Part 3: Switching and routing*

— *Part 4: Mobility*

— *Part 6: Media transport*

— *Part 7: Service composition*

The following parts are under preparation:

— *Part 2: Naming and addressing*

— *Part 5: Security*

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Introduction

The development of the networks during the last years has shown that it becomes harder to integrate new functionality in order to fulfill the demands of new applications and the capabilities of new transport technologies. Especially the core mechanisms are hard to change as it lies in a rigid and ossified architecture.

The current picture of the networks shows a large, heterogeneous, dynamic and complex distributed system. Lots of patches aimed to amend different issues that have arisen during last years. Current networks have to deal with new services, applications and computing paradigms such as new modes of interaction, identification, context-awareness, energy efficiency, seamless service discovery and composition, mobility, ubiquity, etc. At this point, current networks must look for clean solutions to known issues.

The development of a new network architecture has been discussed for some time now. Several proposals are considered in this sense, evolutionary (incremental) approaches and revolutionary (clean-slate). Currently, the general idea in SC6 WG7 is to standardize an architecture to solve current networks faults.

The Future Network (FN) will define a scalable, flexible and robust architecture which will aim at providing services taking into account the changing conditions of the context and thus, offering customized communication and seamless delivery of data. To achieve this, it is necessary to provide service composition capabilities by means of a specific framework that will contribute to create a scalable, modular, and service-aware FN.

The FN introduces a new architecture where the necessary functionality for establishing communications in any node connected to the network (user devices and network elements), is not fixed but dynamically composed, as appropriate to user service requirements, network transfer capabilities and surrounding context in the user and the network environments. In essence, a service-oriented paradigm is followed. Communications are accomplished by assembling appropriate atomic services, each performing a specific communication function. As such, service functionalities can be combined to create higher level communication services, which in turn can be combined with other services as well to enrich existing services or to create new composed ones, until the whole spectrum of required functionality for end-user communications is in place

Service composition is the technology that supports the composition of those activities required to reuse and combine existing services to enrich current services and to create new services. This technology provides a natural way of combining existing services including both atomic and composite services. Such kind of recursive composition of composite services is one of the most attractive and challengeable features of the service composition, allowing to rapidly and easily create new services. Thus, the service composition provides benefits on improved usability of existing services, faster time for service creation and reduced time to market for new services.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

Information technology — Future Network — Problem statement and requirements —

Part 7:
Service composition

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

1 Scope

This part of ISO/IEC TR 29181 describes the problem statement, requirements and a functional building block for the FN from the perspective of service composition. The goal of this part of ISO/IEC TR 29181 is to:

a) analyze and classify problems of the current solutions on the service composition,

b) identify requirements on the service composition for the FN,

c) describe some technical aspects of the service composition for the FN, and

d) propose a functional building block of the service composition including functional components and their reference points among them.

This part of ISO/IEC TR 29181 also introduces various on-going standardization and research activities related to service composition.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC TR 29181-1, *Information technology — Future Network — Problem statement and requirements — Part 1: Overall aspects*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC TR 29181-1 and the following apply.

3.1
Atomic Service (AS)
well-defined and self-contained function or role commonly used in networking protocols (acknowledgments, sequence numbers, flow control, etc.) to establish communications for consuming composite services

3.2
Atomic Mechanism (AM)
specific implementation which provides the desired atomic mechanism functionality

© ISO/IEC 2013 – All rights reserved

3.3
Composite Service (CS)
service that is composed of more than one atomic service

NOTE The composite service logic needs to be specified in a workflow to describe the composition and execution process.

3.4
Service Composition Algorithm (SCA)
mechanism in charge of selecting and associating the specific atomic mechanisms which will create a composite service, specified in the form of a workflow

3.5
Workflow (WF)
formal representation of a CS

NOTE The result of the service composition process is the definition of a set of CSs which will be executed at the nodes involved in a communication. The atomic services are selected according to their specifications and functionalities. Concretely, the algorithm (SCA) will generate the final CS and represented by a WF.

3.6
Patterns or templates
commonly used and well-known WF

NOTE Patterns or templates can improve and speed up the selection process carried out by the execution of SCAs in specific cases.

4 Abbreviations and acronyms

ACCS	Auto-Configuration for Communication Security
AM	Atomic Mechanism
AS	Atomic Service
BPEL	Business Process Execution Language
CBA	Component Based Architecture
CBC	Component Based Computing
CS	Composite Service
DAML	DARPA Agent Markup Language
EBS	Effective Bit Strength
HSP	Heuristic Search Planner
FN	Future Network
IP	Internet Protocol
IPSec	Internet Protocol Security
JSON	JavaScript Object Notation
MOVE	Materialized Ontology View Extractor

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

MPLS	MultiProtocol Label Switching
NAT	Network Address Translator
NGSON	Next Generation Service Overlay Network
OWL	Ontology Web Language
OWL-S	Ontology Web Language for Services
PDDL	Planning Domain Definition Language
QoS	Quality of Service
RNA	Recursive Network Architecture
SDF	Service Delivery Framework
SIDE	Service Integration Development Environment
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SON	Service Oriented Network
SONATE	Service Oriented Network Architecture
TCP	Transmission Control Protocol
TCS	Taxonomical Classification System
TLS	Transport Layer Security
UCPOP	Universal Conditional Partial Order Planner
UDDI	Universal Description Discovery and Integration
WSMO	Web Service Modeling Ontology
WF	Workflow
WS	Web Service
WSDL	Web Service Description Language
XML	eXtensible Markage Language
XSRL	XML Web-services Request Language

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

5 Overview

A service is a set of functions or tasks that provided by software or a system, usually accessible through an application programming interface (API). Considering different types of services, a service can be classified as an atomic service or composite service. Each atomic service provides one concrete and well-defined function.

© ISO/IEC 2013 – All rights reserved

Different implementations of an atomic service may exist in different nodes or co-exist in the same node. The description information of the atomic service should be published and registered to a service registry before providing the service functionality. On the other hand, a composite service is composed of more than one atomic service. Each composite service implies consuming different atomic services and/or sometimes other composite services, with possible dependences appearing between them.

Service composition is the technology that supports the composition of those activities required to reuse and combine existing services to create new services. This technology provides a natural way of combining existing services including both atomic and composite services. Such kind of recursive composition of composite services is one of the most attractive and challengeable features of the service composition, allowing to rapidly and easily create new services. Thus, the service composition provides benefits on improved usability of existing services, faster time for service creation and reduced time to market for new services.

Different design approaches for service composition are used in service oriented computing areas. These approaches can be classified on user defined, semi-automatic or automatic approaches. Under this classification, there are several design mechanisms such as template-based, instance-based, declaration-based, workflow-based, ontology-based, AI planning-based, so on as described in Appendix A.5.5. These design approaches allow building composite services to specify service composition suited for specific service or business needs.

The FN introduces a new architecture where the necessary functionality for establishing communications in any node connected to the network (user devices and network elements), is not fixed but dynamically composed, as appropriate to user service requirements, network transfer capabilities and surrounding context in the user and the network environments. In essence, a service-oriented paradigm is followed. Communications are accomplished by assembling appropriate atomic services, each performing a specific communication function. As such, service functionalities can be combined to create higher level communication services, which in turn can be combined with other services as well to enrich existing services or to create new composed ones, until the whole spectrum of required functionality for end-user communications is in place.

The process of combining available services to create a desired communication service is called service composition.

As opposed to the stringent protocol-oriented approach of current TCP-/IP based communications, the proposed service-oriented and functionality-composed approach adopts a loosely-coupled design. As such, it is beneficial in many aspects:

- It is flexible in building multi-feature and customized communication services
- It allows users to participate in the provision of the services they desire (e.g. user-control routing for performance or cost reasons)
- It provides for adaptation to heterogeneous networks from the very same terminal device
- It facilitates the deployment of new network, service and/or information access technologies from network and user access perspectives
- It avoids redundancy of functionality both in terms of duplication and unnecessary placement

The FN service composition prompts for baring user devices and smart networks. Smart networks would equip on-the-fly with the necessary communication functionalities as appropriate to changing user needs and requirements while, at the same time, they would choreographise themselves to deliver the requested services at the desired quality levels.

Figure 1 illustrates a conceptual architecture of service composition in the FN that is composed by the following functional building blocks.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

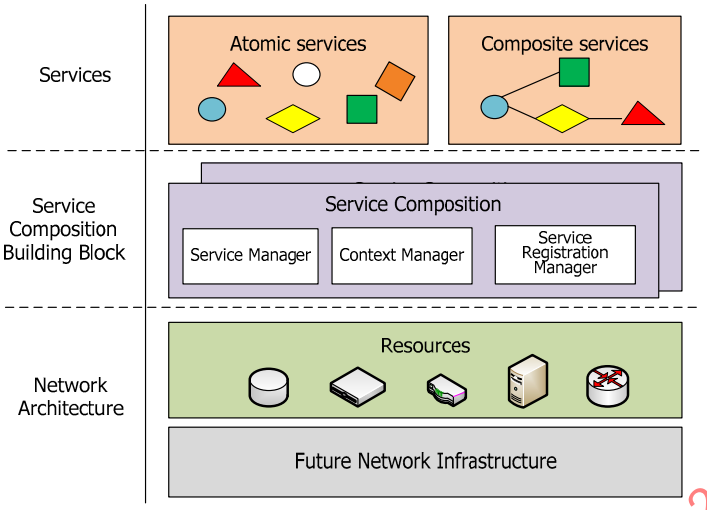

Figure 1 — Conceptual architecture of service composition

- Services

  - From the service composition perspective, services can be classified by atomic services or composite services

  - Atomic service corresponds to a base service that cannot be further decomposed and it does not contain other services

  - Composite service corresponds to a service that is composed of more than one service which can be an atomic service or also composite service itself. It contains an execution sequence of the composite services

  - Figure 2 illustrates a possible taxonomy where services can be classified in different arrangements such as granularity, scope, execution, usage, order, and purpose [1].

    - Granularity: services can be classified as atomic or composite.

      Each atomic service provides one concrete and well-defined networking function (along with the reverse function, if any). Different algorithms and implementations of an atomic service may exist (e.g. different congestion control algorithms), and co-exist in the same node, using attributes to both describe the different possibilities and to tune/configure the atomic service in order to use it to fulfill specific workflows needs. Atomic Services will abstract specific implementations of different functionalities. The specific implementation will be called Atomic Mechanism.

      Each composed service or application implies consuming different atomic and sometimes other composed services, with possible dependences appearing between them. In addition, they can involve one or more nodes, depending on the complexity of the service.

    - Execution/Distribution:

      Isolated: local execution of the service.

      Distributed: execution distributed between two nodes, regardless their location. It includes support for end-to-end, section and hop-by-hop distribution/allocation of services.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

- Scope: services can be applied with different scopes or considerations depending on the desired result.

  Network: services are executed to optimize communication according network context.

  Application: services are executed to optimize application behavior and interconnection to meet application requirements according to context characteristics.

- Usage: rules governing the service usage.

  Mandatory: usage of this service is mandatory as it is basic for establishing a communication (e.g. forwarding).

  Optional: usage of this service is optional, its usage will depend on application requirements and context characteristics.

- Purpose: which is the purpose of the service. Some examples are shown next.

  Delivery: to deliver data between two different entities involved in the delivery chain (they can be adjacent or non-adjacent nodes or they can be two end applications, depending on the scope of the service).

  Mobility: services related to application, user and node mobility.

  Storage: services dealing with the storage of data.

  Security: services dealing with security issues.

  Data Adaptation: services dealing with adapting and transforming data for different objectives, interoperability, customization and optimization of data.

  Addressing: services dealing with the identification and labeling of resources.

  Management: services dealing with the management of the different entities in the network (nodes, applications, services, etc.).

  Signaling: services dealing with interchange of signaling and control data.

  Presentation: services dealing with the presentation of contents and user/application interfaces.

- Order: order/existence of the AS in workflow composition may be dependent of another service.

  Dependent: needs the use of another AS.

  Independent: no need of other AS execution.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

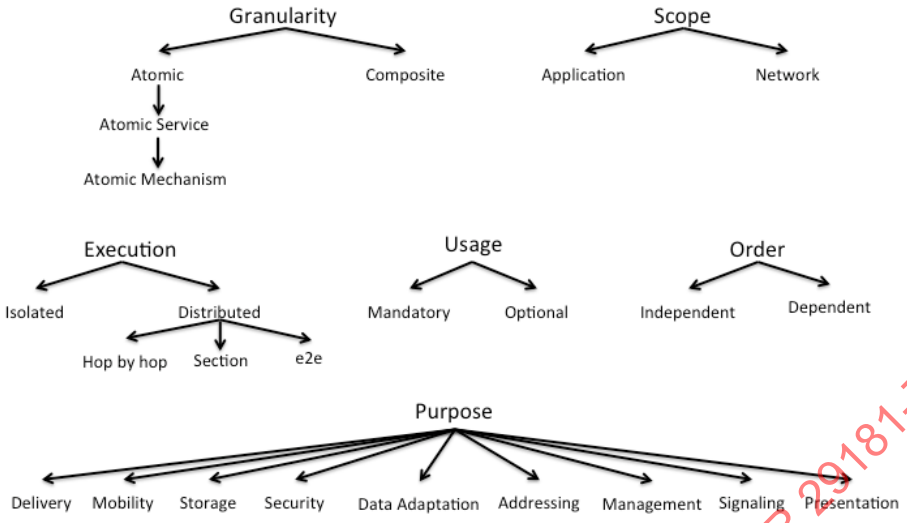

Figure 2 — Service taxonomy

- Service Composition Building Block
  - The FN architecture consists on different architectural components as building blocks that provide a set of supporting technologies such as naming and addressing, switching and routing, media distribution, security, and mobility
  - Service composition is one of the architectural building block identified to support composite services and it is composed by a set of managers such as service manager, context manager, and service registration manager. These managers provide a set of supporting functions such as service selection, service chaining, interpretation of composition description, service execution monitoring, service validation, service adaptation.
  - In the FN, both static and dynamic service composition are supported
- Network Architecture
  - The FN supports the virtualization of different kind of resources that are spread across different locations such as storage, computing power, processing power and network
  - Thus, proper amount of resources can be flexibly dedicated to each atomic service and composite service. For this purpose, the service composition building block should support some mechanisms to coordinate virtualized resources required by composite services.

6 Problem Statement

6.1 General problems

Several challenges might be faced for designing an integral solution for the service composition in the FN that allows to overcome the current technologies and deficiencies.

A new concept and definition of services for the FN will be closely connected to the innovation of heterogeneous environments formed by different kind of networks and users with different requirements. The FN design will allow adopting futuristic capabilities. Complex and personalized users' requirements introduce the need of networks able to be self-configurable and self-evolvable. Considering this, the service composition technology should be also extended to cover possible changes derived from the service and network evolution.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

New distributed software systems have become more dynamic, allowing transparent distribution, self-reconfiguration, portability, etc. Based on that, new paradigms deviated from the end-to-end principle have emerged, such as Pervasive and Ubiquitous Computing or the Internet of the Things.

In addition, the continuous evolution of applications and services are increasing current networks complexity, adding more and diverse requirements (e.g. mobility, security or multihoming) that are not efficiently covered by current TCP/IP protocol stack, as detailed in ISO/IEC TR 29181-2/3/4/5. New features such as data and service identification, context-awareness, seamless service discovery and composition, etc. are required in order to meet the new demanded services and modes of interaction. The lack of these features is withering as well the evolution of networks and slowing down or stopping solutions for known open issues like mobility, flexibility, security, etc. A service-aware architecture should help the deployment of clean-slate solutions in all these aspects.

Nowadays, network level services are executed without taking into account the characteristics of the surrounding context. Consequently, the current architecture fails to provide information of the underlying network technologies, the capabilities of the devices involved in a communication or the characteristics of the users interacting. This provokes that similar or redundant network services (e.g. error correction, retransmission, encryption) are executed at different levels wasting computing resources, introducing unnecessary redundancies and sometimes degrading communication performance. Furthermore, in certain environments, the execution of certain functions can be counterproductive for the correct operation of an application or network service (e.g. TCP's congestion control in wireless networks), making it necessary to modify existing protocols in order to adapt them to environments with restrictions.

This situation is mainly provoked by the rigid and layered communication stack which difficult inter-layer communication. For example, in constrained networks, the need for efficiency in both communication and node performance and energy consumption has led to the prevention of layering and related complex abstractions-by fine-tuning network protocols to application requirements. These approaches result in network architectures that are tightly coupled with the applications, and thus are application-dependent. Therefore, in order to keep the required application-agnostic network architecture simple but flexible, as well as sufficiently optimized, these non-layered solutions should be generalized, thereby converging in modular network architecture able to perform optimally according to different application needs and heterogeneous node capabilities (ISO/IEC TR 29181-1).

However, there are some problems that hindered the deployment of service composition at a macro level. Applying service composition techniques at transport layer means a change of paradigm, which can affect the design of the whole architecture. The inertia caused by the usage of TCP/IP stack and its ossification makes a clean integration of service composition impossible.

Furthermore, a deployment at higher levels is becoming extremely difficult because of the lack of efficient context- and service-aware features and the proliferation of different particular solutions that are not interoperable between different service providers or domains.

Additionally, some scalability aspects may be taken into account, during the service identification and composition process. Depending in which granularity the service is defined, the composition could turn in a high complex process that can last for too much time and this would suppose an unpractical solution. In order to find a feasible solution for service composition, tradeoffs between composition time and optimization should be found.

Currently, services are linked to the physical entities that handle the service. That means that in current set-up, services are addressed through network location of the server and not in a descriptive manner that permit an efficient service discovery. That implies that the service discovery depends basically on the web search engines that permit us do a natural search, or rely on our previous knowledge of where to find the service. Thus, current service discovery processes have a very limited scope. This is especially true for interactive multi-media services and applications, which can dynamically combine different media flows (audio, video, advertisements, etc.) from a multitude of organizationally dispersed sources, content servers, cameras and advertisement/recommendation systems.

For this reason, a shift from current convention towards a more service-oriented architecture is required. Consequently, a generic definition of services and a standard framework to publish and discover them are elementary needed processes. It is important that this definition would be technology and network agnostic.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Thus, services can be discovered and requested at a semantic level, transparently to the exact location of the service and the underlying access and implementation technologies whilst enhancing the interoperability between systems in all kind of environments.

In order to invoke a service available inside a network, a service request and handling mechanisms should be provided by the FN architecture. Ideally, these processes should imply the lightest possible exchange of information between the entities in order to maximize the throughput of networks. Solutions around this issue in current network are performed at an application level and are not suitable as they use heavy communication protocols based on XML (e.g. Web Service technologies). They are not efficient and they are barely prepared for handling a service configuration setup process involving network context and QoS parameters. Service negotiation should be enhanced, allowing users a better control of the services that they are consuming and network administrators of service providers a better management of their resources. Interoperability in service negotiation is a priority as well in order to find mechanisms that alleviate different provider tensions and creating “all party win” situations [2].

A service-oriented landscape where providers can be positioned as competitive collaborators providing multi-provider service compositions is desirable. For that reason, the lack of a negotiation protocol, able to negotiate in-network services between different providers, is an obstacle that should be solved in the FN specification. Furthermore, the introduction of strategies in the network that guarantee a certain level of Quality of Service (QoS) and Quality of Experience (QoE) should be as well a critical need in the FN. Hence, both users and providers, should explicitly specify and agree on the terms of the services that they will receive/provide.

Taking these problems into account, there are some features close related to Service Composition in the FN that should be inherently integrated in its architecture:

- **Dynamic network composition.** Establishing network functions as services that can provide and access easily to all this information should be an essential feature in the FN, in order to have a network that can adapt to current requirements and new ones that eventually rise up. A greater modularity of the network means a better adaptation to new communication paradigms, while decreasing the complexity of the architecture. Unlike static service composition, services can be specified at run time. It means that the capabilities of the service can be extended dynamically, allowing runtime re-composition, decomposition of services, and dynamic adaptation in case of changes in context (services and resources) involved in composite services.

- **Network flexibility.** Network service composition should facilitate in the FN the integration of new functionalities into the network. Instead of tight coupling of functionalities within end-to-end protocols, network services are deployed on arbitrary nodes, are loosely coupled and provide their service to arbitrary other functional blocks. This design originates in the service oriented architecture approach and requires means of service description, discovery and composition. This design approach reduces management efforts and provides a flexible framework to integrate new network services.

- **Inherent cross-layer information exchange.** Cross-layer means that functionalities can be adjusted based on the interaction between different layers. The FN should allow arbitrary composition and information exchange between network services, thus incorporating the benefits of cross-layer design architectures.

- **Context-awareness in service composition.** The FN should support the context management to provide customized and context based services. Thus, different kind of context including user, device, service, resource, and network can be used for discovering, selecting, allocating and composing services to participate in the composition process.

- **Requester empowerment in service choice and routing.** Service requester should have more control over the contents/service that wants to consume. This control must be reflected in flexible routing and service selection according to requester's service definition. Consequently, the FN must build a network architecture that provides more intelligence to the network-side whilst still leaving decision-making processes at the end-points.

- **Semantic Searches oriented to service/resource.** The FN must be focused on a service/data-centric approach that allows executing the search of services and resources based on the requester

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

requirements. This implies that future network must be able to create, discover, negotiate and consume composite services in a flexible and context-aware way.

- **Resources and services identification.** Every flow over the FN must be routed based on its requirements. Therefore, each flow must be identified in order for nodes along the route to cooperate and negotiate autonomously, for guaranteeing the minimum QoS parameters of it.

- **Environmental heterogeneity.** Heterogeneity of nodes, networks and services add another level of complexity to service composition process for the FN. If instances of a service are executed in nodes with different capabilities and network access links, every service instance should be evaluated individually, and attributes of a specific one could not be applied to one of another node.

- **Attribute acquisition.** Composition process should be based on the attributes of the services (and their concrete implementation), but extract the complete and updated information of a service is extremely difficult. It should require a previous empiric process extracting information about how the inclusion of a service or another affects in terms of delay, error rate, and each QoS parameter that are relevant for a complete solution (the whole chain of services from requester to end service provider).

- **Validation and verification.** When multiple service providers interact, it is important to establish trust agreements on service validation and verification to guarantee consistency and reliability of services in the FN in such a way that it does not hamper the entire process of the service composition. Each service needs to be validated its correctness and consistency before registering itself with the FN. Furthermore, the services need to be verified their behavior in terms of functionality, protocols, availability, performance, service price, or other attributes. Moreover, the service composition output should be validated to ensure that the selected services meet users' expectations and, in addition, that the network can provide the requested services. This process can be done once a composition is performed and the output is provided (design-time or run-time). Moreover, service composition validation will evaluate the matching between user requirements, service goals and available resources. The output of the validation process can be a score useful for determining the degree of quality of experience to be achieved.

- **New Business Models.** Internet brings opportunities to create new business models according to novel services, applications and capabilities demanded by users. Hence, it is necessary to introduce innovative models for costing and pricing. The FN architecture should also provide a transparent framework that permits service consumers and providers to interact, establish agreements and satisfy their goals. Furthermore, it will allow network providers to differentiate from competitors by offering services and service composition possibilities at low levels inside the network and guaranteeing them higher margins. A higher accuracy on the use of their resources and a better control of the QoS offered in any case. Because service providers and customers can choose between the services this will also lead to a thriving evolution of network services where the best performing and efficient services will survive [3].

6.2 Scalability

Scalability is always a critical issue, even more considering current evolution of network technologies, types and number of devices, number and heterogeneity of users connected to the network and amount of information exchanged through the network. Scalability can be limited in network domains because, for instance, the use of flooding-based mechanisms that restrict network domain size. Hence, semantic identification will not scale well outside local domains unless some infrastructure support is provided. Besides, clustering strategies are needed to make the semantic routing approach scalable (further detailed in ISO/IEC TR 29181-3 for another part).

Therefore, attributes related to domains, used as labels, or identifiers to distinguish between administrative domains and networks can be used. The main difference with current Internet scheme is that the information used to define and label different domains is not the same as that used nowadays. Currently, network prefixes and addresses are used. They are tightly coupled with routing protocols. Nevertheless, the FN should works with semantic attributes to achieve such flexibility. The FN naming and addressing scheme should not be constrained by their form and meaning as is the case of current IP addresses and domain names.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Hence, any type of attribute is supported and identifiers could take any form. They could be anything that allows identifying and locating a service in a domain. As examples, service identifiers could be a geographical (virtual or not) coordinate bounding the domain area, a random label, an IP address or other legacy/existing addressing schemes requiring no changes in their original architecture. Using one type of attribute or another will depend on the considered scenario and required routing information (e.g. geographical routing will require the use of coordinates).

Thus, high scalability in structured environments could be achieved by means of service discovery that rely on special entities providing the required infrastructure/signaling services whilst avoiding flooding. Some approaches can be adopted such as introducing the use of a semantic resolver instead of performing semantic flooding. The goal of a semantic resolver element would be to process semantic descriptions and map them to identifiers/locators.

6.3 Dynamics

Context dynamics will be a critical issue that affects the validity of the service composition output. Context can be very broad and can contain lots of parameters, static or dynamic. Dealing with high dynamic parameters is a challenging task because depending on the changing metric will imply to recalculate a composition of services. Thus, a service composition able to efficiently deal with context changes will be required. The FN will also need to provide appropriate mechanisms and protocols to gather, represent and share context data.

Regarding network dynamics, the network conditions are the important factors that affect the performance of the service composition process. It means the service composition needs adapt in an intelligent way to diverse conditions of each network domain. For example, the status of workload, traffic, congestion, and the network topology information give benefits to discover and select services in order to maximize the composition performance. However, the collection of such dynamic network information is a challengeable work by the scale and heterogeneity of the networks, which is closely related to the other part ISO/IEC TR 29181-2.

Furthermore, user dynamics in terms of service usage conditions including movement from one place to other, change of devices, different preference, or other environmental conditions require different demand for the service composition. This can be more sophisticated issues in case the user access across multiple service/network providers. Additionally, the user dynamics can cause service/network providers to face difficulties for resource provisioning and management. From the service composition perspective, it is strongly desirable to adapt these user dynamics for the composition process including service discovery, selection, execution, and delivery.

6.4 Security

The trust management and security are critical issues in current-day network (further discussed in another part ISO/IEC TR 29181-5). Internet was created as a network to share knowledge and it was developed by and for researchers with this common goal. Nowadays the situation is really different. Internet is becoming a basic mean for social and commercial interaction. Interests may differ a lot or even, be opposite. This situation has provoked the development of multiple security systems over the network. However, they are so complex and resource consuming to be applied in every situation because of the lack of security mechanisms inherently inside the network. Applying security at service (or content) level will provide a complete solution for protecting not only the entities of the network, but also the information itself. Thus, in the FN heterogeneous scenario, security requirements (e.g. integrity, authenticity, encryption, etc.) need to be introduced from scratch as a main functionality for allowing future networks development and secure access to services.

7 Requirements of service composition for the FN

Provisioning a service efficiently implies a process of selecting ASs (Atomic Services), AMs (Atomic Mechanisms), composing them in a CS (Composite Services), and representing a CS in a WF (Workflow). Then, this WF should be validated. Additionally, if context conditions changes, WF and ASs should be adapted if necessary.

This section identifies a set of requirements of service composition for the FN.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

7.1 General requirements

This TR defines two main basic requirements:

- Future networks must be flexible in order to evolve and allow reuse.
- Future networks must be context-aware in order to adapt to requirements, constraints and context changes.

These requirements yield to the ones shown in 1.

Table 1 — General Requirements

Element	Requirement ID	Description	Main Feature Cathegory
Node	REQ_N_1	Service composition must work despite the underlaying technology or running platform	Technology-agnostic
	REQ_N_2	Service composition and adaptation must work in heterogeneous environments	All-environment operation
	REQ_ N_3	Service composition and adaptation must work in dynamic environments	
	REQ_ N_4	Service composition must deal with failures and changes (network level disconnections, service discovery failures, and service execution failures). Service composition failures can be : • Context not accurate or not reliable --> requires Quality of Information model. If context is not obtained, history can be used. The worst case will imply that the requesters enters the data manually. • Composition request can not be satisfied. Goals may be adapted in order to create a valid service request.	Fault-tolerant and adaptable
	REQ_N_5	The Service Composer must have rules and policies knowledge for guiding the compsition process	Composition rules
	REQ_N_6	Mechanism to guarantee service QoS should be provided to fit QoS requested by CSs. For this, QoS allocation to each AS and QoS coordination of the aggregate ASs QoS is required for CS.	Service QoS reservation and enforcement
	REQ_N_7	Service composition should be validated its correctness and reliability statically or dynamically.	Validation
	REQ_N_8	A service composition process done by another entity should be certified, thus the consumer can validate and accept the service.	Security
	REQ_N_9	Service composition processes should be available with a minimal infrastructure support	Ubiquity and Pervasiveness
	REQ_N_10	Protocols are needed for the negotiation of service composition processes between different providers	Multiprovider cooperation
	REQ_N_11	Methods for abstracting and representing resources as services are needed in order to apply service composition processes at a lower level	Service abstraction

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Element	Requirement ID	Description	Main Feature Cathegory
	REQ_N_12	Node should provide a method to support distributed registration or de-registration of ASs, so that the FN can scale up to an unconstrained number of services that can be supported simultaneously.	Service Reusability
	REQ_N_13	Node must provide the static and dynamic information of ASs including but not limited to service execution status, availability, capability, load, and dynamic service policy.	
AS, AM, CS, WF	REQ_G_1	Descriptive language for ASs, AMs, CSs and WFs are required and basedon formal requirements	Expressive Description
	REQ_G_2	The specification form should not rely on a fixed semantic, since this may not be suitable for the description of newcoming services. A more flexible specification form would also enable later adaptations to yet unknown practical needs.	
	REQ_G_3	Usage or access records should be generated to support charging and accounting for ASs and CS.	Charging and accounting
	REQ_G_4	The level of formality of the descriptive language for ASs, AMs, CSs and WFs needs to be enough to express and validate its timing, concurrency, correctness and consistency.	Validation

7.2 Specific requirements

Table 2 shows specific requirements in terms of AS, WF, SCA, and context.

Table 2 — Specific Requirements

Element	Requirement ID	Description	Main Feature Cathegory	Process stage
AS (Atomic Services)	REQ_AS_1	ASs must be self-descriptive	Atomicity	Service Representation
	REQ_AS_2	ASs must be self-contained		
	REQ_AS_3	ASs must allow loose coupling		
	REQ_AS_4	ASs can be incorporated to the node without affecting the existing ones	Reusability	
	REQ_AS_5	It is required that AS must specify the information of service to be incorporated to FN nodes. The service information can include some attributes or properties such as service name, service ID, provider, and also specific requirements such as SLA, and service policy.		
	REQ_AS_6	ASs' Interfaces must be generic enough to allow their reuse and interaction with the greatest number of components	ASs' Interfaces	
	REQ_AS_7	ASs' Interfaces must hide implementation details. Thus, it is possible to be protocol or mechanism independent.		

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Element	Requirement ID	Description	Main Feature Cathegory	Process stage
	REQ_AS_8	ASs' Interfaces must describe the offered public methods		
	REQ_AS_9	ASs' Interfaces must describe the required input data attributes and the type of output data generated		
	REQ_AS_10	ASs' Interfaces must detail how they can be combined. They must specify restrictions, rules and dependencies		
	REQ_AS_11	Must have a method for getting the list of the supported AMs within a node		
	REQ_AS_12	ASs' interfaces must allow to specify services (ASs and CSs) according to the desired granularity		
	REQ_AS_13	ASs specify the events they are subscribed to		
	REQ_AS_14	ASs recieve data flows with the information to be exchanged and which is understood (processed) by the corresponding AM		
	REQ_AS_15	ASs must have a input and output connections. These are specified in the WF	Interconnection	
	REQ_AS_16	ASs must be combined into specific WFs by mean of compatible interfaces		
	REQ_AS_17	ASs can be orchestrated sequentially and / or in parallel		
	REQ_AS_18	ASs must specify required and optional parameters	Parameters	
	REQ_AS_19	Negotiation mechanism is required to find out the best AS which satisfies the service requirement.	Negotiation	
	REQ_AS_20	ASs should be added, exchanged, removed or dynamically adapted in a executed WF. Depending on the Service Composer implementation, the entire WF may be recalculated in order to fit the new conditions when context changes.	Flexibility	Service Composition (orchestration)
	REQ_AS_21	ASs must be able to be combined with other ASs in order to create CSs and WFs ASs can specify dependencies with other ASs. ASs need to be complementary in their input data and output data in order to be concatenated. ASs recieve and send generic data that will be processed by the corresponding AMs. This implies that AMs must be previously arrenged and must be known by the data generator (source).	Reusability	
	REQ_AS_22	ASs must only identify the data flow which it is targeted. Data will then be processed by the AMs they specifically implement	Data processing	Service Processing

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Element	Requirement ID	Description	Main Feature Cathegory	Process stage
WF (Workflow)	REQ_WF_1	WFs must be scored in order to discern the best ones. The scoring of these WFs must be done taking into consideration several parameters such as: - Network specific parameters: AS's physical location, bandwidth, delay, hop distance, link status, etc. - Service specific parameters: Proximity, energy-efficiency, QoE, etc. - Requester specific parameters: preferences, capabilities, etc. Weights can be applied for each parameter.	Selection	Service Composition (AS selection)
	REQ_WF_2	ASs incorporated in WF must be discovered across FN nodes. Mechanism and discovering criteria for customized AS discovery should be provided for the service composition. Note: REQ_WF_2 is about the AS selection and the requirement for AS discovery is also required before the selection process.	Discovery	
	REQ_WF_3	WFs can be created at design-time	Creation and adaptation	Service Composition (orchestration)
	REQ_WF_4	WFs can be created at run-time		
	REQ_WF_5	WFs specify sequential and/or parallel interactions among ASs		
	REQ_WF_6	WFs must fullfill the requirements specified by the application, the serivce to be consumed, requester preferences, administrators and the network, service and context constraints		
	REQ_WF_7	WFs must be evaluated and validated after their generation in order to verify them and assure that the communication can take place. In order to achieve this, it must be used an appropiate language which allows to verify the correctness in an efficient manner	Consistency	Service Composition (validation)
	REQ_WF_8	Composite services must be represented in a workflow template that include the atomic services used	WF representation	Service Representation
	REQ_WF_9	Atomic mechanisms and input and output configuration parameters of the mechanisms can be added to the workflow information		
	REQ_WF_10	ASs in a WF can be dynamically adapted at run-time according to context changes or the whole WF composition.	Adaptation	Service Adaptation
	REQ_WF_11	Service Requester controls the adaptation process.		

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Element	Requirement ID	Description	Main Feature Cathegory	Process stage
	REQ_WF_12	Whenever possible to modify an AM without modifying the WF, will do that		
	REQ_WF_13	Whenever possible to modify a part of a WF without requiring a recomposition (and negotiation) process, will do that		
SCA (Service Composition Algorithms)	REQ_SCA_1	SCA must be time efficient or not very time-consuming	Performance	Service Composition (orchestration)
	REQ_SCA_2	SCA can not be the same at each node. Algorithms must be implemented according to the node capabilites	Diversity	
	REQ_SCA_3	Inference or logic reasoning can be used to optimize the decision making process	Inference	
	REQ_SCA_4	SCA should be as self-explainable as possible in order to identify and clarify its reasoning process	Explainability	
	REQ_SCA_5	The Service Composer can take as inputs the following: - Requeriments (Application, Service) - Constraints (Application, node, context, service), lis of available ASs, lists of available SCAs	Inputs	Service Composition (AS selection, AM selecion, orchestration)
	REQ_SCA_6	Need of a translator between the request to the problem definition, using the format used by the composition methodology		Service Composition
	REQ_SCA_7	Service composition process should not overtake an excessive amount of time that hider service establishment	Time-critical	
	REQ_SCA_8	The increment of the service composition time should be lineal to the increment of nodes in the service provisioning process	Scalable	
	REQ_SCA_9	The service composition process should establish how the services and workflows would be allocated in the network nodes along the communication path.	Allocation	
Context	REQ_C_1	Dynamic service QoS measurement should be provided to make sure that the target service QoS is achieved for the CS request. QoS capabilities of the network must be monitored (performance, reliability, security, costs, etc.)	Network Monitoring	Context Acquisition
	REQ_C_2	Node capabilities should be monitored (static and dynamic or in-time)	Node Monitoring	
	REQ_C_3	ASs within a node should be monitored in order to identify its availability	Node Monitoring	
	REQ_C_4	Requester environmental constraints should be monitored and stored in requester context profiles	Requester Monitoring	

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Element	Requirement ID	Description	Main Feature Cathegory	Process stage
	REQ_C_5	Active ASs should be monitored in order to evaluate its performance, reliability and cost	Service Monitoring	
	REQ_C_6	Context can be acquired by different mechanisms such as: push, pull or publish/subscription[1] to events	Node Monitoring	
	REQ_C_7	Multiple contexts parameters can be monitored together in order to be more efficient	Node Monitoring	
	REQ_C_8	Each context parameter can be acquired separately	Node Monitoring	
	REQ_C_9	Every AS must specify which context parameters it needs to adapt itself	Service Monitoring	
	REQ_C_10	Requester context and preferences should be maintained through different sessions. History of AS and AM usage should be stored in order to extract implicit feedback from user	Requester Monitoring	Context Representation
	REQ_C_11	Context Aware Computing requires suitable context model and quality information model (optional)	Representation	
	REQ_C_12	Different ontologies and description formats should be supported in order to allow interoperability. A specific AS performs the transcoding operation between formats.	Interoperability	
	REQ_C_13	Context information must be able to be updated whenever is necessary	Up to date information	
	REQ_C_14	Context acquisition, management and processing processes should be decoupled in order to ease the development	Management	Context Awareness
	REQ_C_15	Context information can be filtered in order to give only relevant information to the service	Information quality	
	REQ_C_16	Context processing should be a fast and light process	Context Processing performance	

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

1 Publish/Subscribe methods can be configured to operate as pull or push data distribution operations.

© ISO/IEC 2013 – All rights reserved

Annex A
(informative)

Related standardization and research activities

A.1 IEEE P1903 (NGSON)

The IEEE 1903 working group has been proposed and launched by IEEE Communications Society in March 2008. The main goal of this group is to address a new paradigm for a next generation of service overlay networks (NGSON).

This group has specified the functional architecture of NGSON, IEEE Std P1903-2011, that defines a framework of IP-based service overlay networks with advanced service networking capabilities.

The NGSON architecture provides advanced service and transport-related functions to support context-aware, dynamically adaptive, and self-organizing networks. The functional architecture of NGSON consists of a set of functional entities (FEs), their functions, reference points and information flows to illustrate service interaction and media delivery among FEs and external components.

NGSON may operate with different underlying networks such as IMS, NGN, P2P overlay or Web to transmit NGSON signaling messages and/or media among its users and services.

Figure A.1 illustrates a high-level overview of NGSON's functional architecture.

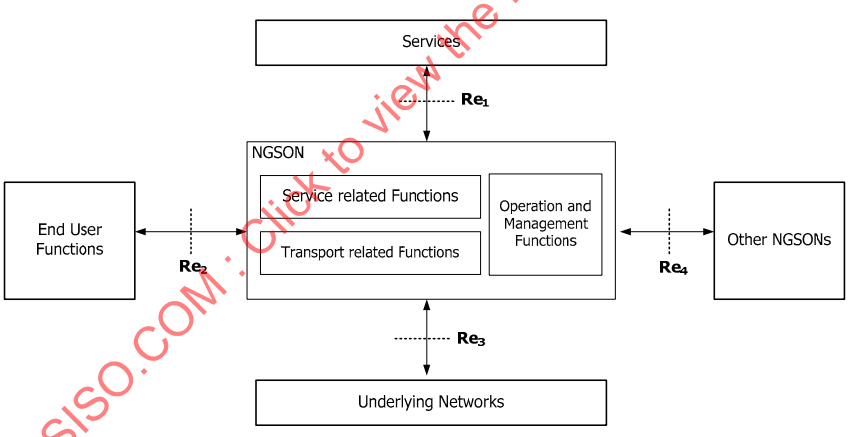

Figure A.1 — Overview of NGSON's functional architecture

Some important features of NGSON correspond to context aware service composition, service discovery and service routing. NGSON defines the functional entity named Service Composition FE that is responsible for execution of composite services. It supports static and/or dynamic service composition with the capabilities for choice of the type of composition, choice among existing services, choice of how to manage the process of composition, and optimize the composition sequence.

The main functions provided to support the service composition are as follows.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

- Service brokering function: It supports static and dynamic service composition which include static service chaining, dynamic service chaining, static service instantiation and dynamic service instantiation based on semantics and context information.
- Specification interpretation Function: It interprets the specification of service composition to perform either static or dynamic composition of base services accordingly.
- Composite service adaptation function: It maintains the functionality of the created service composition using re-composition, including but not limited to, when any included base service fails or when the interfaces of the base services are modified.
- Charging event supporting Function: It supports reporting charging information to the charging system.

NGSON addresses the concept of service composition through service overlay networks that are placed between services and underlying transport networks as an intermediate layer. In other words, NGSON is intended to provide a set of service supporting functions like service composition in the overlay network layer. Comparing some traditional service composition approaches such as Web services, Mashups, NGSON gives benefits of provision of dynamic service composition and utilization of different types of context information including network conditions to support service composition, interaction and service delivery.

After completion of the standard of NGSON functional architecture, this group has initiated new projects to develop additional standards of protocols for content delivery, service composition and self-organizing management.

The approach of NGSON for service composition has made a broad impact to support composite services in an easy and flexible manner. However, this approach has limitations in resource provisioning for QoS and balancing the quality among multiple instances of composite services.

A.2 TMF Service Delivery Framework

TM Forum [5] is a global, non-profit industry association focused on simplifying the complexity of running a service provider's business. This Forum is working on several cooperative activities to address the industry challenge of producing an environment for the management of next generation services. For this, the TM Forum provides frameworks which enable service oriented, highly automated and efficient reference models for next generation services. They published frameworks such as Business Process Framework (eTOM), Information Framework (SID), and Applications Framework (TAM). The TM Forum is also working on the standardization of Service Delivery Framework (SDF) for telecommunication industry. SDF defines the framework that supports and integrates all functions required for the lifecycle of a service delivered to a customer, across all stakeholders in a service provider environment. This forum published TR139 named "Service Delivery Framework Overview Release 2.0" in March 2008 that provides a common logical service delivery framework for design, creation and composition, deployment, activation, execution, charging, monitoring and management of services. SDF can be considered as a super framework that allows many different SDPs to be linked together. In SDF, it is possible to incorporate different architectural domains with different resources, policies, and security. This enables the composition of services that cross operator boundaries using networks, resources, and application components from multiple service providers.

Currently, this forum is developing a management framework enabling effective service delivery in the context of emerging industry trends such as Web 2.0 and converged services.

A.3 ITU-T NGN SIDE

ITU-T SG13 [6] is working on the standardization of Service Integration and Delivery Environment (SIDE) in NGN in order to support integration and adaptation of resources from different domains, resource brokering, service development environment, and context based service delivery. SG13 has developed and published a recommendation for requirements and capabilities for NGN-SIDE as Y.2240 in April 2011. NGN-SIDE can be considered as a service delivery platform with some advanced capabilities such as context and content management. NGN-SIDE has three functional layered views: user layer, SIDE layer and resource layer. The

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

user layer corresponds to those applications or users who utilize services or resources provided by SIDE. The second layer is compromised of integration layer and adaptation layer. This second layer supports creation, execution and delivery management, and adaptation layer, which supports adaptation capabilities for NGN environment. The resource layer provides capabilities of resource brokering among resources and applications located in NGN-SIDE.

One of the important capabilities of NGN-SIDE is the service orchestration that provides mechanism to create new composing resources and executing composing resources. Composing resources is the process of generation of the service logic. Therefore, the composition of resources or services is provided as a service orchestration capability.

After publishing Y.2240, SG13 is currently developing a draft of functional architecture of NGN-SIDE as a sequent work step.

A.4 ATIS SON Forum

The Service Oriented Networks Forum (SON) [7] is addressing work to enable the interoperability and implementation of Service Oriented Network (SON) applications and services by developing standards, providing coordination for the development of standards and practices, and facilitating related technical activities. This forum is placing an emphasis on telecommunications industry needs in collaboration with regional and international standards development programs in the telecommunications, IT and Web industries.

A.5 Related research activities

This section overviews some projects and initiatives that present clean-slate architectures related to SOA, and then an analysis is presented based on the fundamental stages of a service-oriented approach.

One of the focus that 4WARD [8] project emphasizes is a shift from networking nodes to networking information. This information-centric approach is applied to routing and data transport (as structured objects). In this approach, protocol design is focused on analyzing protocol invariants in order to build them from their most basic functional blocks (e.g., error control, encryption, etc.). Thus, similarly to Object Oriented Programming, 4WARD advocates virtualizing the different architectures and protocols using constructions called Netlets made of several functional blocks, in order to meet specific requirements. In this way, nodes instantiate and execute Netlets on top of a kind of protocol virtual machine.

SONATE [9] presents an approach completely based on service-oriented computing, considering the Internet as a large, distributed (software) system. Its goal is to encapsulate micro-protocols by services. Thus, their definition of a service relies on open, standardized and generic service interfaces, permitting the decoupling of logic from implementation, and offering an abstract view on the functionality of these protocols, in contrast to hiding mechanisms by layers. However, this specification should not rely on a fixed semantic, to allow later extension to yet unknown practical needs.

RINA [10] project proposes a clean slate approach. It proposes a general theory of Inter Process Communication (IPC) where the number of IPC layers (homologous to OSI model) may vary depending on the range of the resource allocation. This network architecture is based on the following principles: (1) Mechanism and policy are separated in the protocols; (2) Connectionless and connection communication modes are unified; and (3) Topological addresses are embedded in a Distributed Inter Process Communication Model.

The Recursive Network Architecture (RNA) [11] explores the relationship of layering to protocol and network architecture. Its primary goal is to encourage cleaner cross-layer interaction, to support dynamic service composition and to gain knowledge on how layering affects architecture. In order to fix the current Internet architecture problems, RNA proposes the use of a generic layer protocol, named metaprotocol. This metaprotocol contains a set of basic services, as well as hooks to configurable capabilities. RNA also notes that the particular services that a protocol provides are dependent on the context (there are environments where a particular mechanism of a protocol is best suited than another one), as well as, on the upper and lower layers that this has.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

TARIFA [12] aims at defining a clean slate approach to a Future Internet architectural redesign, based on a role-based paradigm consisting of non-divisible, or atomic, functions. TARIFA architecture is service-oriented, enabling dynamic composition of services and its adaptation, taking into account context status and its variations. Thus, it allows providing adapted communications according to user needs.

Table A.1 summarizes the features of these projects focusing on service related aspects:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Table A.1 — Service-Oriented approaches comparison

Projects	Service				
	Definition	**Publication**	**Discovery**	**Composition**	**Adaptation**
4WARD	Object-oriented, defining Functional Blocks, Functional Block Mechanisms and Netlets as constructions of them. Additionally, NetInf point of view defines Information Objects.	Maintain a repository containing Building Blocks and Design patterns for guiding the composition process. NetInf proposes an approach that uses DHTs to publish Information Objects.	NetInf proposes a dictionary based on DHTs.	Decomposition and recomposition. Chaining Functional Blocks into Netlets. Done at design time by a network or a service architect.	Dynamic behavior provided by a set of policies that tunes the building blocks at run-time In-network management approach considers building block adaptation.
SONATE	Services as essential building blocks that provide self-contained functionality, well-defined interfaces and loose coupled among them. Also defined by a set of effects.	Service Providers publish services at Service Broker.	Service Consumers ask for a service to Service Broker.	Service Broker is the entity in charge of selecting and composing the service matching to the requirements demanded by the Service Consumer. It relies this process on Analytical Hierarchy Process (AHP).	Not considered
RINA	Services are defined by a set of primitives and arguments, and a set of rules. Associate each protocol to a service definition, but separate its behavior.	Define a Resource Information Exchange Protocol (RIEP) to populate a Resource Information Base (RIB) with application names, addresses, and performance capabilities.	Routing is executed creating the forwarding tables based on RIB information.	RINA bases its composition in an establishment of an Inter Process Communication (IPC) among nodes.	RIEP can be used in both, a request/response mode and a notify/confirm mode, using the same managed objects. This allows IPC processes to notify the other members when there is a significant change or to request information.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Projects	Service				
	Definition	Publication	Discovery	Composition	Adaptation
RNA	Single protocol, (metaprotocol) with a set of basic services (configurable capabilities). Three types of elements: Data element, Control element and Template element.	Not considered.	Not considered.	The metaprotocol is tuned (manual or automatic) depending on the context in which it is located.	Not considered.
TARIFA	Self-contained, self-describing, modular block offering a specific functionality that can be published, located, and invoked across the network. This self-* definition should permit loose-coupling among services	Distributed Plane (DP) used for maintaining services and context information. DP is divided into a Resource Plane (RP) and a Knowledge Plane (KP).	Propagation of semantic requests specifying the requester requirements. Nodes evaluate incoming requests taking into consideration context information.	Nodes can compose services and adapt them in runtime. Compositions are specified in the form of WFs.	Services can be self adapted considering context variations.

Projects like 4WARD or RNA establish a composition based on templates and static processes. In other cases, like SONATE, all nodes and networks are considered homogeneous, because complexity increases considerably when selecting services in heterogeneous environments. Future architectures should go beyond these problems and also empower user decision, by letting composition decisions to requesters. That should also permit a more autonomous network that does not depend always on middle boxes (like SONATE Service Broker).

Additionally, TARIFA links service discovery to routing by means of a semantic-based discovery process, establishing communications easily and adapted to network state in each situation and time.

A.5.1 Service-Oriented Architecture (SOA) paradigm

A good deal of New Future Internet initiatives are based on SOA (Service Oriented Architecture) paradigm. SOA paradigm proposes to organize and to use distributed capabilities operating under the control of different ownership domains. Services are the essential building blocks of SOA, and it is based on three premises:

1. Use self-contained services and data
2. Use explicit descriptions instead of implicit assumptions
3. Use well defined interfaces between services

This paradigm can be related to Component Based Computing (CBC) design, based on the decomposition of a problem into several pieces called components. Then, these components can be composed in order to fulfill specific problems and covering the requirements needed at specific times. This approach gives important benefits in terms of flexibility, scalability and adaptability, as the components are instantiated as needed.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

In addition, SOA defines three basic elements:

- Service User
- Service Broker
- Service Provider

The latter two, create a service oriented communication system. The Service User is composed of an Application and an Application User.

To allow the integration and support of different protocols and formats, an Adapter can be placed in order to make the corresponding mappings.

A.5.2 Service Oriented Network Architecture (SONATE)

SONATE approach is based on principles of service oriented architectures (SOA [13]), foreseeing Future Internet architecture as a large, and distributed (software) system, where a set of services communicate, cooperate, and inter-operate with each other. The goal of this service-oriented networking definition is to offer fine-grained functionality to applications to chose from.

SONATE defines services/mechanisms as an abstraction of specific algorithms and data structures used to implement a functionality. Hence, each service provides a self-contained functionality and a well-defined interface that must not take assumptions about internals of other services in order to loose coupling among them.

According to their scope, the services are conditionally grouped into three service clouds:

- **Application Services Cloud.** The application services cloud offers services related to application functionality. Examples of such services are authentication, application notification service, application information exchange service, etc.
- **Mediation Services Cloud.** The scope of the mediation services cloud incorporates services related to network functionality. Typical example of mediation services are connection establishment and release, flow control and congestion control, etc. The presented approach focuses on this cloud.
- **Connection Services Cloud.** The connection services cloud encompasses services related to data transport. Such services are, for example, modifying network data, signaling, signaling error correction, etc.

As a result to cooperation and composition of these services, SONATE defines workflows, which provide more complex functionalities by selecting sets of services and defining their interaction. Thanks to the loose coupled definition of services and the abstraction of their functionalities SONATE tries to simplify the service composition and orchestration, permitting to add or remove services and change their implementation without affecting the whole workflow.

For building a network based on principles of SOA, SONATE considers specific supporting techniques, since Web Services and XML are inappropriate to implement services on a network level (other light-weighted semantics are needed). However, for primary implementations or proof-of-concept of the efficient behavior of their framework, they specify services and workflows using XML.

Selection and composition is the task of finding a building block graph to fulfill a certain need. A workflow can be defined explicitly by a requester or may be derived from known dependencies. Automatically creating a workflow on demand i.e. connection setup, offers the highest degree of flexibility and workflow optimization, responding in short-term to requests or changing environmental conditions.

Any algorithm that tries to solve the composition problem needs information about which service connections and graphs are valid and which attributes that combination has. This information must be provided in a way that can be processed automatically and it must not restrict service interaction in any way. The easiest approach would be to list all services that are compatible together with the attributes of the combination. For some usage domains this is feasible but not for a future network architecture.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

A crucial point that SONATE tries to enhance is the flexibility of the composition (and thus the network). In this sense, they consider that a global database where all dependencies and combination possibilities of services are to be stored would destroy this flexibility. This means that dependencies and combination possibilities must be expressed in an implicit way that does not require any changes to other services when one service is added, changed or removed.

In service composition process, for each set of requirements of an application, a matching set of services/mechanisms could be assembled. Those mechanisms then form a custom micro-protocol stack (which is more like a mesh in fact).

To dynamically build these micro-protocol stacks mechanism SONATE participants consider necessary to describe their capabilities and requirements in a way that an algorithm can predict the outcome of any combination. To achieve they specify that each mechanism has a set of requirements, a set of provided effects, a description of the costs of the mechanism and a formula how these costs "add up" with the costs of other mechanisms. An algorithm selects mechanisms so that all requirements are met, all desired effects are provided and the combined costs are low.

In service selection process, the goal is to improve the cost of the solution without losing the consistency. Another possible solution is to have a pre-calculated list of solutions and to adapt them in a predefined way, as the Netlets in 4WARD project which is presented in the next sub clause. Or maybe multiple heuristics can calculate solutions concurrently and then the best one is chosen.

A.5.3 4WARD

4WARD [8] was a European research project in the FP7 frame and its goal is to make the development of networks and networked applications faster and easier leading more advanced communication services. 4WARD specifies a framework where the composition of functionality is based on chaining some building or functional blocks which are implemented by specific functional block mechanisms. 4WARD is also based on a Component Based Architecture (CBA), where the decomposition of the engineered systems into functional or logical blocks with well-defined interfaces used for communication across these components. Components are considered to be a higher level abstraction than objects and as such they do not share state and communicate by exchanging messages carrying data.

In 4WARD the dynamic behavior is provided by a set of policies specified by each building block which are evaluated at run-time. An example of their appliance would be when it is necessary to choose a polynomial to calculate the CRC (Cyclic Redundancy Check) in the presence of an alarm or specific event.

4WARD organizes building blocks in different strata (Horizontal, Vertical and Abstract). In addition, 4WARD is able to compose context-aware services thanks to the concatenation and execution of Building Blocks which perform a specific action is contained and specified in a Netlet. 4WARD proposes to use a repository of well-known or best practices to effectively compose different types of functionalities, it is called Design Repository.

In order to cope with the high complexity of communication systems, 4WARD provides two different views on network architectures: the macroscopic view and the microscopic view. The first one is more related to an overall structuring of the network architecture at a rather high level of abstraction in terms of strata. The second one deals with the functionalities needed within network architectures, their composition to so-called Netlets in order to fulfill desired requirements as well as the Node Architecture hosting various Netlets of the same or different families of network architectures.

The microscopic point of view introduced in 4WARD tackles the problem of the composition of basic functionalities or atomic services. The goal of the composition is to build new protocols fulfilling the requirements of each communication.

In 4WARD a functionality can be represented with different functional blocks as illustrated in Figure A.2.

The functionalities used in a network can be part of different categories of functionality. Each category can contain different functional blocks and a functional block can belong to different categories at the same time.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

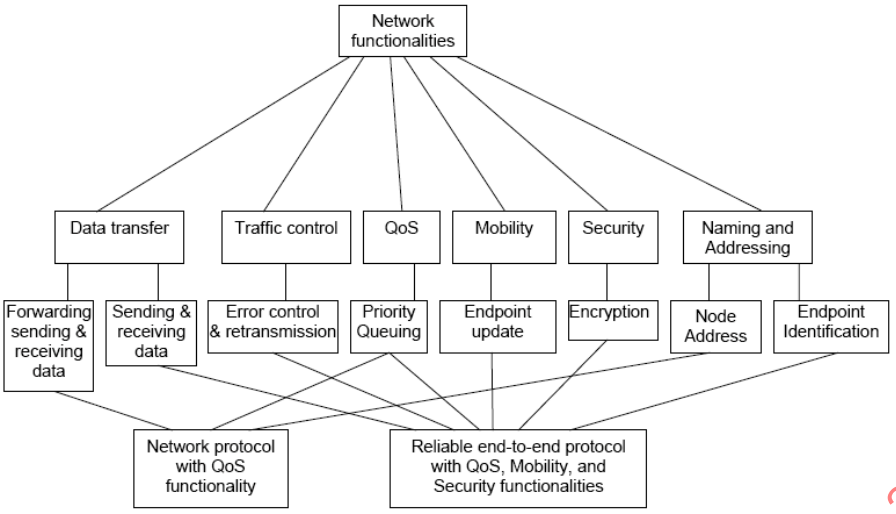

Figure A.2 — Functionalities composition in 4WARD

The ordering of functionalities is important, so it must consider hop-by-hop communications and end-to-end communications restrictions.

To select the best functional block among all the possible ones, a model for representing them is needed. 4WARD model tries to describe direct and indirect changes that a building block does to data, thus, they are represented as black boxes.

The model of the functional blocks is used as a basis towards selection and configuration of functionalities (Figure A.3), which is based on [14].

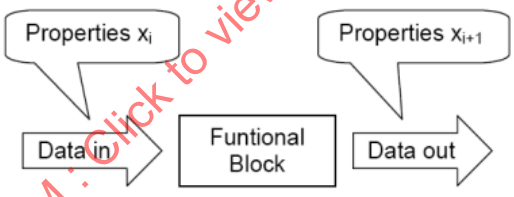

Figure A.3 — Functional Block abstract model in 4WARD

By calculating what each functional block does to the data, a prediction of what a composition does can be obtained. This prediction is precise enough to compare it with the predictions of alternative compositions, therefore allowing us to choose the “best” composition for a given data flow.

Regarding to protocols, they are basically means of data exchange for a communication between nodes or systems. In order to understand each other, the concerned communication parties must agree on syntax and semantic of exchanged data units. Syntax means the data unit’s structure and the fields it contains. Semantic means the meaning of each field of the data unit. Syntax and semantic make one protocol different from another and create a variety of network protocols. Current data units are usually composed of payload and header. The header in turn is composed by fields carrying control information such as network address, data checksum or congestion notification. Based on the header structure presented in Role-based Architecture (RBA), where roles can be equivalent to functionalities in the context of this section, they propose a method to design new protocol and dynamically build the protocol header by functionality composition.

A protocol implements a set of abstract functionalities and is composed by one or many FBs. The header is built based on the functionalities needed by composing the corresponding functional blocks, and finally associated with the corresponding payload to form the whole PDU of the protocol. The separation between the functionality and the functional block can provide the protocol design and header construction with a high

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

flexibility similar to the concept of polymorphism that exists in software engineering. As an example, many addressing schemes may be used to implement the addressing functionality and each provides an address value which can be transported by the header. When switching the address scheme, the address format and value in the header change accordingly. One could achieve this by implementing the addressing scheme as Functional Block Policy.

Regarding the communication related functionalities, different FBs can implement a needed functionality. The protocol specification and the header format will change accordingly to the functionality composition in order to get the required protocol.

A.5.4 Web Service Composition

Besides, although that work in former and current Future Internet projects could be very relevant because of the similarity of the clean-slate vision of the architecture, it is also important to revise service composition in other environments where it is already further studied and deployed. Often related to higher-level services, Web Service composition is an extensive subject in the literature.

Web services are self-described software entities which can be advertised, located, and used across the Internet or a network using a set of standards such as SOAP, WSDL, and UDDI [15].

Web services encapsulate application functionality and information resources, and make them available through programmatic interfaces, as opposed to the interfaces typically provided by traditional Web applications which are intended for manual interactions.

There are two types of Composition: static and dynamic composition

- Static Composition - services to be composed are decided at design time
- Dynamic Composition - services to be composed are decided at run-time

Some languages for web service composition are the following ones:

- BPEL4WS (Business Process Execution Language for Web Services)
- WSFL (Web Services Flow Language)
- XLANG (BizTalk)
- BPML (Business Process Modeling Language)
- ebXMl BPSS (Business Process Specification Schema)

Examples of ontology-based workflow logic for the composition of services include Web Ontology Language for Services (OWL-S) and Web Service Modeling Ontology (WSMO).

Some requirements introduced by Web Services are the following ones:

- Representation of an abstract Web Process
  - Representing/specifying the abstract process in a proper form
- Discovery and Interoperability of Services
  - Need to manually or automatically search for appropriate services
  - The discovered services should interoperate
- Efficiency of a Composed Web Process
  - Need to compose processes which are efficient in terms of performance
- Process Execution
  - Adopting a suitable technique for executing the composed concrete process
- Process Monitoring
  - Using a monitoring technique for run time analysis of the Web process execution

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

A.5.5 Service Composition Approaches

In [4] authors propose a classification system (TCS, Taxonomical Classification System) in the form of taxonomy for semantic web service composition approaches.

TCS provides a classification providing four levels (except workflow based node) of hierarchy in classification taxonomy. The focus of classification at first level is the amount of user involvement in the composition process. It provides three nodes Manual/User-Defined, Semi-Automatic, and Automatic. The second level, which is based on the procedure or steps followed in the composition process, creates template and instance based categories under user-defined node while Declarative, Workflow, Template/Instance, AI Planning [16], Ontology based, and Hybrid categories under Automatic node [17]. The third level of taxonomy is based on the amount of dynamicity involved in the composition process and the technology used to implement the procedure adopted in second level. It creates dynamic and static nodes under each of template based, instance based, and workflow based nodes of second level; chaining, logic and rule based under AI planning based and context and agent based under ontology based node of second level [16]. The fourth level of taxonomy produces the categories and is focused on some variations possible in the technology adopted in the third level. The fifth level of taxonomy in workflow based node is based on the centralization level in the execution process.

A.5.5.1 User Defined/Manual approaches

These approaches do not plan from scratch like other approaches. They are manually configured or described. The most used methods in this area are template or instance-based compositions, which use recommendations methods to choose a specific pre-designed composition among all available templates or instances.

A.5.5.2 Template-based and instance-based approaches

This approach usually takes outlines of permissible workflows (or templates) and instantiates one of them to obtain an executable workflow. This process can be based on a set of rules that covers policies, context events and constraints. Policies are specifications of behavior that the composition should have, usually described as Event-Condition-Action (ECA) formulations, while context parameters and constraints are used to describe conditions that affect these policies.

This method could be applied alone, where the initial template should be hand-coded or can work jointly with other automatic methods generalizing the workflows that these automatic approaches extract, in order to re-apply a known workflow for a new situation.

The policies can also guide plan adaptation to external events (e.g. changes in context constraints or the performance of the service) while a composition is being executed.

The template-based approach works best for environments where compositions follow regular patterns (in general more static networks with infrastructure) and its adaptation is required to be along anticipated context.

However, the complexity of this approach depends on its generalization. This results on the number of instances that could be selected. Given a template of length λ services, where each instance has at most M possibilities, the worst case of complexity to produce executable workflows is O (M^{λ})[18].

Some approaches [19] take this methodology along with templates of services' semantic descriptions, and propose a template-based searching mechanism to allow dynamic searching of business process partners at design or deployment time of a Web process.

In [20], a method is presented to compose Web services by applying logical inference techniques on pre-defined plan templates. The service capabilities are annotated in DAML-S/RDF and then manually translated into Prolog. Now, given a goal description, the logic programming language of Golog [21] (which is implemented over Prolog) is used to instantiate the appropriate plan for composing the Web services. Golog is based on the situation calculus and it supports specification and execution of complex actions in dynamical systems.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Other methods [22] introduce the notion of composition templates only to provide composability soundness; compositions are associated with each composite service and are used to compare values added by different compositions. To check whether a composition is sound or not, stored templates are used to store a composite service's template in the service repository. A service composition is considered as sound if its template is the subset of a stored template.

A.5.5.3 Declarative approach

In declarative composition, composite services are generated from a high-level declarative description. The technique uses composability rules to determine whether two services are composable [23]. Most of the time these rules act as constraints that must be satisfied in order to compose a service. The rules are used to generate composition plans that conform to a service requester's specifications. Techniques that fall under this classification usually tend to reach optimality of composition against some defined objectives (i.e., cost, time…etc.) as they are mathematically modeled. Mostly, the optimality can be achieved by mapping rules to constraints and trying to solve them using operation research methods [24].

FUSION is a software infrastructure system that provides the common infrastructure elements needed to support service portals [25]. Given a user service specification, it automatically generates a correct and optimized execution plan, then executes this plan and verifies the result. The most important advantage of this system is its ability to generate an optimal execution plan automatically from the abstract requirements that a user may specify. Also, this composition system verifies that the result of the execution plan meets the user's requirements, and if not, it immediately recovers this execution plan. However, this system creates a bundle of services and uses it to specify an execution plan by choosing services from this bundle. Web services are evolutionarily increasing and determining which subset of these web services to use will limit the choices for the user and will not guarantee the optimality of the result. Also, there will be a probability of using web services that no longer exist.

A technique suggested in [26], which extends the 'inverse rules' query reformulation algorithm to generate a universal integration plan to answer a range of user queries. The inverse rules algorithm unites the inverse rules with the user query to produce a datalog program. A technique was developed to map datalog programs to integration plans that can be executed by a streaming, highly parallel execution engine called Theseus. Moreover, a mediator system is described that accepts a user query and returns a URL of a new dynamically composed web service that can answer a class of user queries similar to the user query. The system has many limitations. First, the system does not support any monitoring mechanisms that test the validation of the composition. This may result in improper execution of composite web services. Second, there is no mechanism to automatically model the newly generated services as a data source for upcoming compositions. Third, this technique lacks semantic representation and composition of these services, thus resulting in an error prone composition.

SELF-SERV [27] is a framework for dynamic and peer-to-peer provisioning of web services. In SELF-SERV, web services are declaratively composed, and the resulting composite services are executed in a decentralised way within a dynamic environment. The framework uses and adapts the state-charts as a visual declarative language. The significant advantage of SELF-SERV is the peer-to-peer service execution model, whereby the responsibility of coordinating the execution of a composite service is distributed across several peer software components called coordinators. Nevertheless, this system does not provide a method to create a composition at runtime for services. Also, it does not consider any semantics of web services during composition decisions. Moreover, this technique imposes some unrealistic requirements that should be implemented by service providers from the partial point-of-view.

Another significant work was performed in [24] to reduce the complexity and time needed to generate and execute a composition and improve its efficiency by selecting the optimal services at the current time. This research proposed an architecture of dynamic web service composition by runtime searching of registries to find services. Therefore, this technique does not use any service template. Moreover, it reduces the dynamic composition of the web services to a constraint satisfaction problem where any linear programming solver can be used to solve it. Also, it insures the optimality of the web services selection based on domain specific QoS parameters identified by the user. However, this approach does not support user interactive participation in the composition process, which is sometimes important to ensure the user's satisfaction.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

enTish [28] is somewhat different from typical composition platforms. Services are typically created on the fly to realize client requests.

Anyway, most frameworks are based on the assumption that first the business process has to be created. For enTish, a different architecture is needed, since client requests are expressed in a declarative way using formal languages. The declarative approach consists of two phases: the first phase takes an initial situation and the desired goal as starting point and constructs generic plans to reach the goal. The latter one chooses one generic plan, discovers appropriate services and builds a workflow out of them.

The first phase is realized using PDDL (Planning Domain Definition Language) and estimated-regression planning as used in XSRL (XML Web-services Request Language), which must provide machine readable semantics and specify the abstract service behavior. The second phase may be realized by using existing process modeling languages, such as BPEL.

A.5.5.4 AI Planning

Planning is a problem solving technique where knowledge about actions and their consequences is used to identify a sequence of actions, which when applied in a given initial state, satisfy a desired goal [29].

A planner can receive three inputs mainly:

- Initial state: describes the starting state of the application domain. It is normally called world.
- Goal state: describes the desired world state.
- Domain description: describes actions that, when invoked, transform the world states.

The output of the planning process is a plan, a sequence of actions that can be executed in order to achieve a desired goal state.

The composition engine may be implemented by a number of different composition methodologies. For example, AI Planning has proven to be an effective tool for service composition [30][31]. Services can be represented in terms of their non-functional and functional properties. Non-functional properties describe service provider details and Quality of Service parameters. Functional properties contain descriptions of service operations in terms of inputs, outputs, preconditions and effects, which make it easy to translate them into planning actions.

Planning systems need semantics for describing the available actions, thus to describe the formalized domain.

The simplest form of domain formalism is based on state-transition models. The state describes the world at a certain point in time, such as an initial state or goal state. Actions perform transitions between states. An action can be described with a list of preconditions, states and post conditions.

Planning technologies differ in the complexity of the problems they can handle and the representations that they use. Moreover, they employ different search algorithms to synthesize plans and the constraints they observe [32].

There are different planning technologies:

a) **State space planning:**
   - Simplest planning algorithm
   - Search space is a subset of the state space
   - There are two types, based on the search starting point.
     - Forward: chaining planner searches in the space generated by applying to each state all actions whose preconditions are satisfied, starting at the initial state.
     - Backward: search algorithms start at the goal and apply inverses of the planning actions to produce sub-goals, stopping if the initial state is reached. The main limitation of state-based planners is that their performance reduces with the size of the search space.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

- To address this performance limitation researchers employ heuristic functions to estimate the usefulness of the alternative actions a planner can choose from. Heuristic methods are found through discovery and observation of the planning process. They guide search algorithms often based on feedback from past executions. This initiated a new type of planning, planning with control knowledge, as discussed later on. Heuristic Search Planner (HSP).

b) Plan space planning:

- Search the space of partially specified possible plans. As a result, the searching process becomes a plan refinement operation.
- There are two kinds of step that can be taken in constructing a plan:
  - adding an action
  - adding an ordering constraint between actions.
- This type of planning is called “partial order planning”, because until the ordering constraints are added, the order in which actions are taken is not specified. This approach avoids extensive backtracking that slows down a state-space planner.
- e.g. Universal Conditional Partial Order Planner (UCPOP)

c) Planning graph techniques:

- Employ graph structures to represent search spaces.
- Given a problem statement, the planning system explicitly constructs and annotates a compact structure called a planning graph.
  - Represents a plan as a flow of truth values through the graph, which has the property that useful information for constraining the search can quickly be propagated through the graph as it is being built.
- Graph-based planners then exploit this information in the search for a plan.

d) Hierarchical Task Network Planning:

- Are based on the notion of hierarchical decomposition, also known as reduction or expansion of an action. This process decomposes an abstract action into a group of steps that form a plan to implement the action.
- The main objective is to produce a sequence of actions that perform some activity or task. The description of a planning domain consists of a set of actions, as well as a set of methods, which prescribe how to decompose a task into subtasks.
- The description of a planning problem contains an initial state; however, instead of a goal formula there is a partially ordered set of tasks to accomplish. Planning progresses as a recursive application of the methods to decompose tasks into smaller and smaller sub-tasks, until primitive tasks, which can be performed directly using planning actions, are reached. For each composite task, the planner selects an applicable method and instantiates it to decompose the task into subtasks. If the plan later turns out to be infeasible, the planner backtracks and tries other applicable methods.

e) Model Based Planning:

- Planning based on model checking is a methodology that aims to address non-determinism, partial observability and extended goals.
- It treats the domain as a nondeterministic state-transition system, where an action may have multiple outcomes. Temporal logic formulas are used to express the set of goal states and the conditions of the final plan execution. Planners use a state transition system and a temporal formula to generate a plan that controls the system evolution so that all of the system’s behaviors make the temporal formula true.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

A.5.5.5 Ontology based

This technique facilitates the semantic dynamic composition of web services. The ontological descriptions and relationships among web services are used to automatically and semi-automatically compose web services (in this case, any service). The ontology-driven approaches mainly compose the services based on the goal-oriented inferring and planning [33].

The fact that web ontologies are becoming too large to be used in a single application has stimulated many researchers. For instance, a distributed architecture – called Materialized Ontology View Extractor (MOVE) – is introduced in [34] and elaborated in Bhatt et al. (2006) for the optimization/extraction of sub-ontology from a large scale base ontology. This work has been extended in [35] to address the issue of semantic correctness of the resulting sub-ontology. Moreover, large distributed ontology framework for tailoring ontologies in the Grid environment has been investigated [36].

Most of the ontology-driven techniques mark-up web service descriptions with ontologies and develop algorithms to match and annotate WSDL files with relevant ontologies. The possible compositions are obtained by checking the semantic similarities between interfaces of individual services (semantic matching) and considering the service quality (QoS matching). Then, these compositions are ranked and presented according to these two dimensions.

A semantic context-based approach for composing web services is proposed in [37]. The composition is performed based on understanding the semantics of interactions/capabilities of the elementary services. A conceptual architecture that enables ontologies to integrate models, languages, infrastructures, and activities to support reuse and composition of semantic web services is introduced in [38].

To achieve semantic composition, these techniques mostly require a domain-specific ontology design that defines explicit formal specifications of the concepts and relationships among the concepts. It might also require an extraction module that helps us in building ontologies from service profiles. The example shows the taxonomic classification of concepts as well as the relationships that exist between entities.

A dynamic web service composition system is proposed in [39] where a service is requested and composed not by its syntax but by its semantic. To satisfy the requirement for semantic support, the system comprises three sub-systems: Component Service Model with Semantic (CoSMoS), Component Runtime Environment (CoRE), and Semantic Graph based Service Composition (SeGSeC). CoSMoS integrates the semantic information and functional information into a single semantic graph representation. CoRE provides a unified interface to discover and access components implemented in various component technologies to make them interoperable with CoSMoS components. SeGSeC is a semantic-based service composition mechanism that allows users to request a service using a natural language sentence and it generates the execution path. The major contribution is that this work is semantic-based. Moreover, the CoRE component enables the proposed system to interoperate with other legacy existing systems without any updates to these systems. In addition, the composition technique is somehow controllable by the end users. However, the set of queries that can be used to test the system's functionality is quite limited (not to mention that the system was designed for very limited scenarios and it needs a lot of add-ons to be able to satisfy other types of queries). Finally, the system does not have any monitoring tools for executing the composite services. This implies that execution failures are not monitored at runtime.

Another ontology-based dynamic service composition research is presented in [40].The authors used the Ontology Web Language (OWL) and DAML-S to provide the semantics needed for web service composition. However, semantic description is considered as a part of the process for another sophisticated technique. They have developed a service composition prototype that has two basic components: a composer and an inference engine. The inference engine stores the information about known services in its Knowledge Base (KB) and it has the capability to find matching services. The inference engine is an OWL reasoner built on Prolog. Ontological information is written in DAML and is converted to RDF triples and loaded to the KB. The engine has built-in axioms for OWL inference rules. These axioms are applied to the facts in the KB to find all relevant entailments. The composer is the user interface that handles the communication between the human operator and the engine. The composer lets the user create a workflow of services by presenting the available choices at each step. The advantage of this work is the use of semantic web for web services composition. Yet, this work suffers from centralization, thus yielding scalability and availability problems. Moreover, it requires passing redundant messages between the coordinator and other parties, which causes an inefficient

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

use of the bandwidth. Also, it uses filtering on a set of pre-discovered services, rather than dynamic matchmaking and loading.

A.5.5.6 Generating Process Templates using Process Mining technologies

Data mining techniques are usually strictly related to process mining problems [41]. In fact, supposing a certain number of tasks within a process, the number of possible different executions is an exponential number (while not all of them have the same probability to be actually executed). Here is where data-mining technology helps.

In [42], while working in the area of software engineering processes analysed three different methods for process discovery: neural networks, Markovian models, and an algorithmic approach. Their work permits to generate explicit process models, and to measure the actual gap between process model and actual observed behavior [43] introduced the idea of extending process mining to workflow management, by analysing the events recorded in a log and by identifying constraints. Their approach is based on an algorithmic approach; by enumerating tasks instances and applying a folding procedure E. M. Gold in [44] shows that the problem of finding a state-machine compatible with a set of recorded data is NP-hard. This problem has an analogy with the process mining problem; even if an important difference is that process mining needs to take into account concurrent tasks. Another important research stream is the one linked with the capability of identifying uncertain variables (either due to low frequency of their observation either due to incomplete datasets). Silve, Zhang and Shanahan [45] propose a probabilistic workflow model and a learning algorithm that is capable to compute in a polynomial time. This model is based on directed acyclic graphs (DAG) where each node represents a task and the arrows represent dependency relationships, so that given the predecessor tasks, a task execution is independent from the other ones. The learning algorithm builds such graph by connecting the various tasks (nodes) on the basis of their joint instances in a log file.

Finally, other approaches of [46] are based on clustering techniques on data contained in log files. The solution is based on dynamic Bayesian networks or Petri stochastic networks. More in particular, factorial Hidden Markov Models are suitable for modeling parallel tasks. The major limitations of these mining approaches when applied to workflow modeling are:

- process instances have a precise start and conclusion time
- there are synchronization constraints

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

Annex B
(informative)

Technical aspects of service composition in FN

Specific technical considerations to enable service composition in the FN, and create a service-aware and flexible network based on principles of service-oriented architectures and service-oriented computing.

Evidently, due to performance and scalability reasons, service composition cannot be feasibly implemented across all communication layers. Performance-critical communication services need to cast to hardware or firmware e.g. those related to transmission; as such, they are better shipped with the hardware interfaces (NICs) than being composed. Services at the network or, even, higher communication layers, being implemented in software, can well be composed rather shipped in monolithic stack-based systems. Thus, to accelerate composition time, predefined service compositions can be stored (e.g. in templates) to be reused avoiding calculating them again.

B.1 A common protocol for supporting service composition

A common generic protocol should provide a mechanism for basic message exchange between instances of services on different nodes that permit the interoperability among heterogeneous nodes. Other techniques are required to enable interaction of different services within the same node.

Communications should be established between instances of the same service that are executed in different nodes. Thus, a generic protocol is needed to separate messages of different services and to identify which service a message belongs to.

This common protocol is a basic mechanism required for discovering available services in the network and, thus, to discover and select the services that will be composed in order to achieve specific communications.

B.2 Service Composition approaches

Figure B.1 shows different service composition approaches in the form of taxonomy for semantic web service composition approaches. This classification can be applied to the global concept of service composition. However, not all these approaches are suitable for network functionality or in-network service composition because of the special characteristics and constraints in this field [4]. Sub clause A.5.5 describes the detailed information of these approaches.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

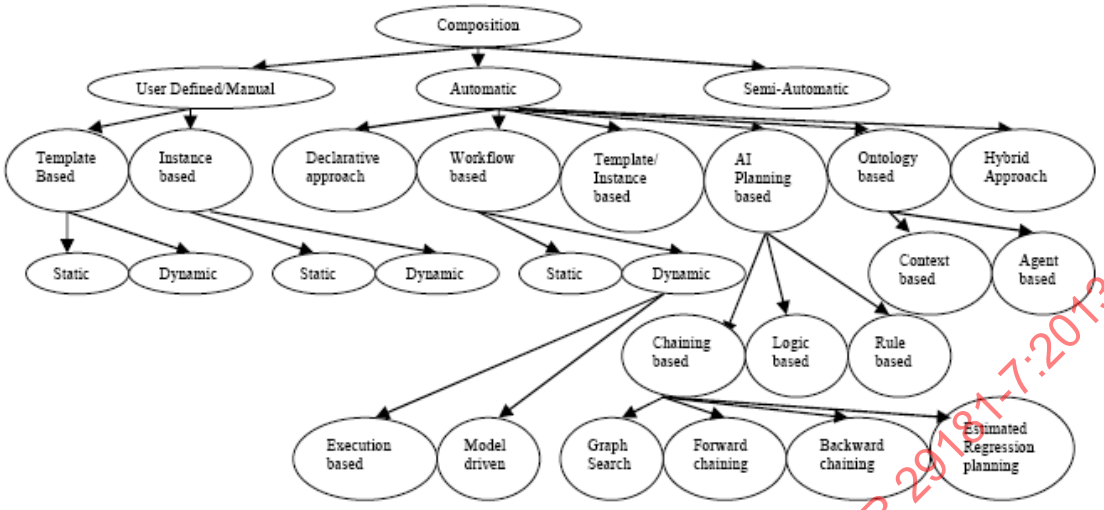

Figure B.1 — Service Composition approaches [4]

B.3 Composing network functionality

A service composition process is always driven by a target/desired communication service. For instance, it can take place when a user requests a service or an in-network QoS-aware connectivity service that needs to be provided under certain conditions and constraints.

The targeted/desired communication service must have specific and well-defined objectives. Its conditions reflect the situational state of the surrounding environment at the time of its usage, which can be modelled through context information. And, its constraints reflect user preferences and network technological capabilities. All this information constitutes the input to the composition process.

No matter where and when executed, a composition process implies the following steps:

- Capture of input information; service objectives, context information and network technology capabilities.
- Identification of the service functionalities that will make up the target service based on information from corresponding to the communication templates and information of the desired service type, taken from matching the required with the offered service properties.
- Discovery the identified service functionalities, within and across provider domains.
- Selection of the most favorable synthesis using the discovered service functionalities, on the basis of current component availability and operational efficiency metrics; in the general case, there may be alternative synthesis patterns and/or alternative service functionalities per synthesis.
- Allocation and realization of the determined synthesis by enforcing the interoperation of the selected service functionalities.

The FN architecture will specify and develop suitable means for effectively and efficiently realize the functionality involved in the above composition functions.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

B.4 Composition scope and service granularity

The FN is concerned with the composition of the required functionality to accomplish user communications, out of a set of existing basic data-networking-level service functionalities e.g. data transfer, forwarding or naming. Hence, the FN focuses on composing services at the connectivity, transport and application communication levels.

Independent of network or service technology, the FN considers basic service functionalities that can be classified as in two levels:

- Network (connectivity services)
- Higher communications levels (transport and application services)

Furthermore, services can be part of two spaces, user device and network spaces. The term in-network services to refer to services/functionalities that reside in the network. An initial view of assumed existing service components is depicted in the following figure.

The FN will be agnostic to the underlying technology of the existing service functionalities to be used in composition, while it does not place any requirements or constraints on the functional granularity of these services. The FN is not focused on standardizing the set of basic networking service functionalities, which is their functional granularity. The FN is concerned on the development of generic means to intelligently bundle together sets of available service components to enable end-to-end communications.

There should be a trade-off associated with the functional granularity of the service functionalities to be synthesized. The finer the granularity, the more flexible the composition outcome, but the more complicated and time consuming the composition process would be. The coarser the granularity, the less flexible the composition outcome, but the more simplified the composition process is.

B.5 Place for composition

For a particular target communication service, such as a required end-to-end user service like a streaming service, composition processes need to take place both at user and network spaces.

At the user space, the composition process is executed at end devices, based on services residing therein and on in-network services that need to be shipped by the network to the user devices. Alternatively, the network or networks where the user is connected to may compose for the user the required communication services and ship the resulted service to the user device. Evidently, composition at the user space enables users to have control over their communications. For instance, this is especially interesting in multihomed scenarios.

In any case, by relying on composition, user devices should not commit to any communications stack, above a basic networking level. Instead, they should be equipped with the necessary functional and execution environment to enable communication service composition or undertake composed services from the network.

At network space, the composition process may be executed centrally or in a distributed fashion within a provider domain, depending on the nature of the communication service to compose and the engineering strategy and architecture of the provider. Composition at different provider domains may also be required, for instance, achieve an end-to-end QoS-aware connectivity service requested by a user.

Once a service is composed, as decided by the composition process, it is realized in a distributed fashion, spanning from user devices to end service nodes across different provider domains, using the constituent services along the way.

The realization of a composite service and its management are important aspects of the composition process, as they affect the performance of the resulted service. Deciding which constituent services to use in the composition process should take into account not only functional relevance but also metrics related to their realization (e.g. set-up time) and its operational efficiency (e.g. robustness, energy consumption, etc.).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

B.6 Composition execution epochs

There should be a trade-off associated with the time required to execute the composition process with respect to the time a requested service is called. Basically, there are two extreme ways to do service composition.

- Design-time execution: the composition process has little information to utilize, therefore its outcome might not meet user expectations, but it has sufficient time to generate a possible solution.

- Run-time execution: the composition process has more information to utilize, therefore its outcome may be more accurate, but it has to be realized in virtually zero time.

Run-time execution might be suitable, within narrow time margins, for explicitly called services which, involve a signaling phase (e.g. voice services). However, it is totally unsuitable for most Internet information access services, where data is transferred in a connection-less mode.

Moreover, the FN architecture would be able to rely on hybrid schemes combining the merits of both approaches design-time and run-time composition execution schemes. Hybrid schemes can be based on a 'prepare-to-fly' approach, versus a truly 'on-the-fly' composition approach, where compositions can be prepared in advance to eliminate additional composition overhead.

To achieve this, a possibility would be to adopt a gradual composition logic evolving over different time epochs.

At the user space, composition may take place gradually at the first time the user end device is connected to the network and at current service demand time, or, even, the first time a service is requested.

At the network space, in-network service composition may take place at various epochs so that to create an adequate readiness level. Appropriate in-network services may be composed at design times to have efficient and reliable realization times.

To practically implement this evolutionary composition approach, we introduce the notion of communication templates. A communication template prescribes a specific set of atomic services to be used in the composition of a particular type of desired service. The constituent services will be completely specified during the composition process, where the current details and characteristics of the requested communication service will be known (e.g. context parameters). These templates are created manually at design time, reflecting the knowledge of the network provider to provide a certain set of service types, based on technologies that employs or can support. In addition to service types, communication templates may be dependent on the epoch to perform composition, prescribing more gross guidelines. Suitable means to describe and efficiently interpret such guiding templates for optimizing the performance of the composition process will be needed.

B.7 An architecture based on services

A unified framework defining the architecture, protocols, and corresponding processes is required for the FN. This framework will stress on the composition, permitting to compose services in three levels: connectivity, transport and application. The FN architecture present both, evolutionary and clean-slate approaches. It can go for evolutionary or incremental approach, composing transport services and assuming that network technology is in place. However, it could be seen as a revolutionary architecture when connectivity services are composed as well.

For example, a three-plane architectural view can be adopted. At the bottom, a data plane, distributed throughout network nodes, is responsible for transferring the data of user services. This plane is built based on existing link transmission and switching technologies. On top of it, a service control plane, at each network node, runs the logic of the distributed service composition protocols and appropriately configures the data plane for constructing the paths along which the end-service data will flow. The protocol messages are transported from node to node by means of the data plane and their routing is determined by the logic of the protocol. Finally, at the top, a context plane, residing in network nodes and/or in an overlay, maintains the information required by the protocol logic to route its messages; the plane is also updated with service-related information from the service control plane (Figure B.2).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved

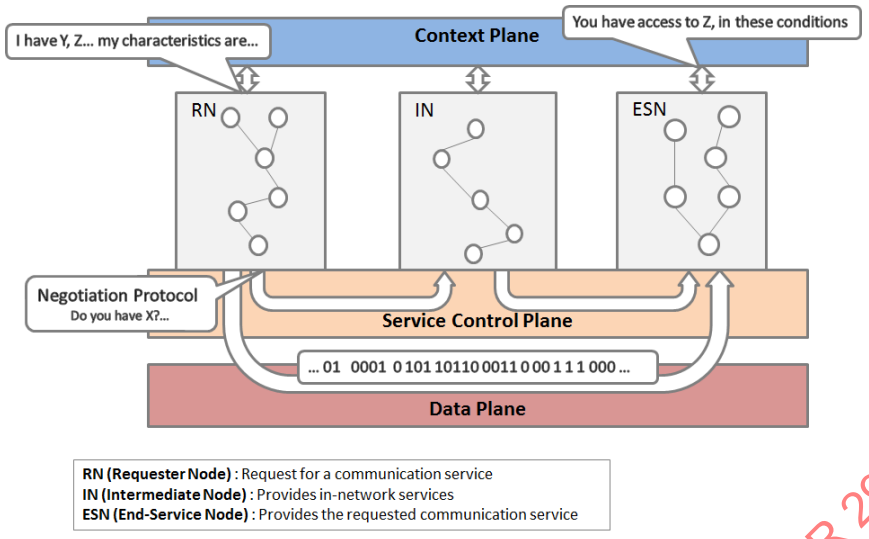

Figure B.2 — FN communication architecture

The FN will need to explicitly address composition in a multi-provider environment, seeing into the necessary interaction between providers and allowing orchestrating services that involve entities of different domains and providers to provide end-to-end services. Some entities will be defined during the project for assuring this functionality. However, an envisioned view of a scenario can be seen in Figure B.3, where manager entities of the domain will share some information such as available services of the domain with associated costs and QoS provided. To achieve this, appropriate SLA agreements must be arranged between the involved and inter-connected entities.

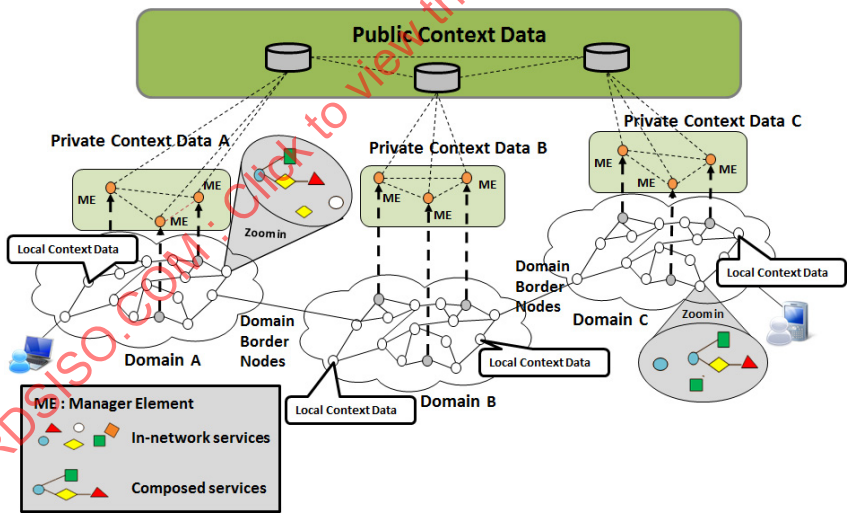

Figure B.3 — FN on an infrastructure multiprovider scenario

In order to operate with such scenario, two kinds of protocols will be needed:

- Distributed service composition protocols for composing services at the connectivity level; the resulting composed service determines a path for specific service data flows (the protocol will be context-aware integrating semantic service discovery capabilities.

- Negotiation protocols that permit providers to negotiate configuration parameters of the in-network services and establish SLAs between them.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC TR 29181-7:2013

© ISO/IEC 2013 – All rights reserved