On Telecommunication Software Engineering Education

Manfred Sneps-Sneppe
Ventspils University College
Ventspils, Latvia
manfreds.sneps@gmail.com

Dmitry Namiot
Lomonosov Moscow State University
Moscow, Russia
dnamiot@gmail.com

Vladimir Sukhomlin
Lomonosov Moscow State University
Moscow, Russia
sukhomlin@mail.ru

Abstract— **Communication specialists around the world are facing the same problem: shifting from circuit switching to packet switching. In 2006, the Pentagon adopted a new plan for the next 15 years entitled Joint Vision 2020. The plan announced the Defense Information System Network paradigm shift: the transition from SS7 signaling to IP protocol. It is assumed that the IP protocol will be the only means of communication between the transport layer and applications. The next steps relate to cyberspace operations and Joint Information Enterprise building on the unique model MBSE (Model-based Systems Engineering) and the unique language SysML (Systems Modeling Language). All these expectations depend upon the sophisticated software. We could add to this the newest Software Defined Networking and Network Functions Virtualization concepts. What to teach engineers and programmers – that is the question.**

I. Introduction

Communication specialists around the world are facing the same problem: shifting from circuit switching to packet switching. In 2006, the Pentagon adopted a new plan for the next 15 years entitled Joint Vision 2020. The plan announced the Defense Information System Network (DISN) paradigm shift: the transition from SS7 signaling to IP protocol. It is assumed that the IP protocol will be the only means of communication between the transport layer and applications.

The next step relates to United States Cyber Command (USCYBERCOM) requirements started from 2009. The command centralizes on cyberspace operations, organizes existing cyber resources and synchronizes defense of U.S. military networks. Defense Information Systems Agency (DISA) expects to build Joint Information Enterprise based on the unique model MBSE (Model-based Systems Engineering) and the unique language SysML (Systems Modeling Language). All these expectations depend upon the sophisticated software like the newest conceptions of Software Defined Networking (SDN) and Network Functions Virtualization (NFV) concept. What to teach engineers and programmers – that is the question.

In June 2012, Lockheed Martin got $4.6 billion contracts to operate, secure US military network [1]. Lockheed Martin's Information Systems & Global Solutions Division won a competition, transferring the keystone GSM-O IT services contract away from SAIC, a 15-year incumbent. Under the seven-year Global Systems Management Operations (GSM-O) contract, Lockheed Martin will provide operations, maintenance, and cyber-security for DISN, which is the global telecommunications network for the US military. GSM-O pays for the worldwide support services necessary to carry out day-to-day operations of the Global Information Grid (GIG) networks and related services, and to update them with new technologies.

Defense Information Systems Agency (DISA) expects the GSM-O be carried through five task orders: (1) network operations, (2) network maintenance, (3) cybersecurity, (4) contingency support, and (5) network integration.

The network integration task concerns in moving two Global Network Service Centers (Vaihingen and Bahrain) to the U.S. territory (Fig. 1).

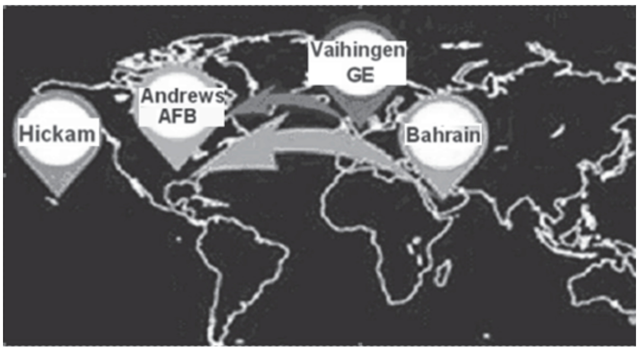

Fig. 1. DISN integration (See Fig. 6 below for detail)

But, in August 2016, Lockheed Martin has sold IT $4.6B business to Leidos [2]. Why? It seems Pentagon cannot upgrade the DISN due to the lack of programmers and telecom analysts. In the long list of vacancies on the Lockheed Martin website in the first place was listed search for analysts of multifunctional information systems for DISA. Applicants are required to develop new services and skills to extend AIN services, from one side, and to have expertise in equipment from CISCO, Juniper, Promina, Safenet, Ciena, Sycamore, Ericsson, from the other. Plus – the highest level of confidentiality. That is, we need specialists to improve the "old" secret core AIN network (already 30 years old) and its docking with the new heterogeneous multi-vendor devices, military equipment. Naturally, Lockheed Martin was not able to arrange an enough great team of such universal specialists, and thus - $4.6B business failed. The similar picture, we afraid, could be for many DISA projects especially due to the extra sophisticated cyber-war requirements.

Industry Job Title
Mult Functional Information Systems Analyst
Job Description
Provides engineering and technical expertise on all issues relating to the specified telecommunications networks/information systems within the DISN.
Applies specialized knowledge of military unique features, specifically built into the network and its subtending hardware and software, to ensure appropriate support to the warfighter's requirements.
Implements or extends advanced intelligent network features into the network/system.
Applies systems engineering disciplines to the provisioning of new service offerings over the network/system, often specifically tailored with military unique features.
Basic Qualifications
Requires expertise in one or more of the following devices/vendors: CISCO, Juniper, Promina, Safenet, Ciena, Sycamore, or Ericsson.
Security Clearance
Top Secret

Fig. 2. Example of Lockheed Martin job vacancies for veterans

Our goal is to discuss one extremely hard question - what to teach engineers and programmers due to increased role of software in modern telecommunications. The rest of the paper is the following. In Section II, we discuss three generations of DISN (from AIN and SS7 to IP protocol and cyber-security). In Section III, we look at the telecom software development history. Section IV gives the insight in SDN and NFV technologies. In Section V, we talk - what to teach engineers and programmers. In Section VI, we discuss some approach to help in telecom software development.

II. THREE GENERATIONS OF DISN

A. GIG.1: the orientation towards AIN

The Defense Information Systems Network (DISN) belonging to the Pentagon is the world's largest departmental network. The DISN has been developed since the early 1990s. This is a global network. It is intended to provide communication services by transmitting different types of information (voice, data, video, and multimedia) in order to perform the efficient and secure control of the military, communications, intelligence, and electronic warfare media.

These requirements are reflected in the 15-year program of weapons development entitled Joint Vision 2010, which the United States Joint Chiefs of Staff adopted in October 1996 [3]. Regarding the means of communication, the Advanced Intelligent Network (AIN), the highest achievement in the art of circuit switching, was chosen (Fig. 3). Remember that AIN architecture was developed by Bell Labs in the 1970s and the breakup of the Bell System was in 1982, long time before the DISA solution.

The basic AIN design includes (Fig. 3a): STP (Signaling Transfer Point), SSP (Service Switching Point), SCP-DB (Service Control Point with Database), each End Office (EO) contains Signaling Point (SP). The AIN provides integrated "one stop" end-user services, such as voice, data, video, e-mail, images, office applications, and 800 services. SS7 is a means by which elements of telephone networks exchange information. Information is conveyed in the form of messages. SS7 defines the procedures for the setup, ongoing management, and clearing of a call between users. The key points of AIN are the following: Service Control Point and Database of services, as well as TCAP (Transaction Capabilities Application Part) - a main protocol in the SS7 protocol stack, providing access to databases.

Intelligent Peripheral also plays an important role: its functions include tone generation, voice recognition, speech and data compression, dialing recognition, and much more, including tactical and strategic services for personnel identification. The Adjunct provides the same operation as the SCP but is configured for one or fewer services for a single switch (for deployed forces).

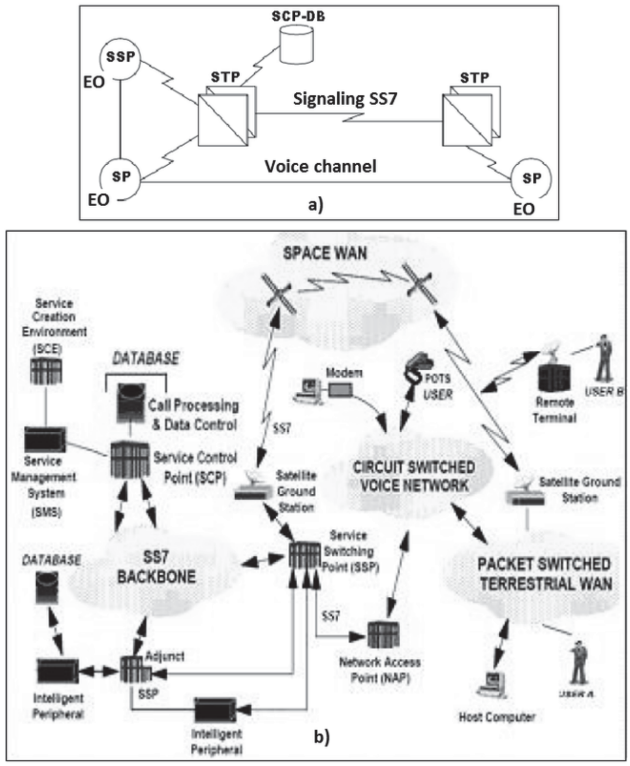

Fig. 3. a) AIN basic design. b) AIN Service Architecture in DISN.

Fig. 4 shows the current state of DISN under testing PBX AVAYA [4]. As it has seen, the SS7 network is, figuratively speaking, the nervous system of a DISN switched network up to recent time: the center of the diagram is occupied by the SS7 network. That is, within the DISN network, the connections are established by means of SS7 signaling and, in the periphery, devices of any type are used. The devices are connected by any protocols: 4-wire (4W); classified LAN (ASLAN); ISDN BRI; Internet telephony (VoIP); video-conferencing (VTC); any proprietary protocol; a link via communication satellites to remote telephone networks and tactical networks at theaters of military operations (STEP/TELEPORT).

From above an important conclusion follows: the DISN network tends to adopt new terminal equipment (to a large extent, this is IP media), but the SS7 network retains its central position till now. The presence of the SS7 network is not an obstacle to the transition to IP protocol.

B. GIG.2: the transition from TDM to IP

In 2006, the Pentagon adopted a new plan for the next 15 years entitled Joint Vision 2020. The plan announced a Defense Information System Network (DISN) paradigm shift: the transition from SS7 signaling to SIP protocol. Note that SIP protocol is regularly deployed alongside SOAP, HTTP, XML, VXML, WSDL, UDDI, SDP, RTP and other protocols (totally 11 RFCs). But SIP, as a signaling protocol, does not have the ability to break into ongoing calls. The support for Multi-Level Precedence and Preemption (MLPP), can be used instead. For this reason, particularly, Assured Services SIP protocol was invented [5].

According to Unified Capabilities Requirements [6], AS-SIP requires support for nearly 200 RFCs. It is the substantially large number of requirements for the end instrument that make AS-SIP different from the typical SIP stack.

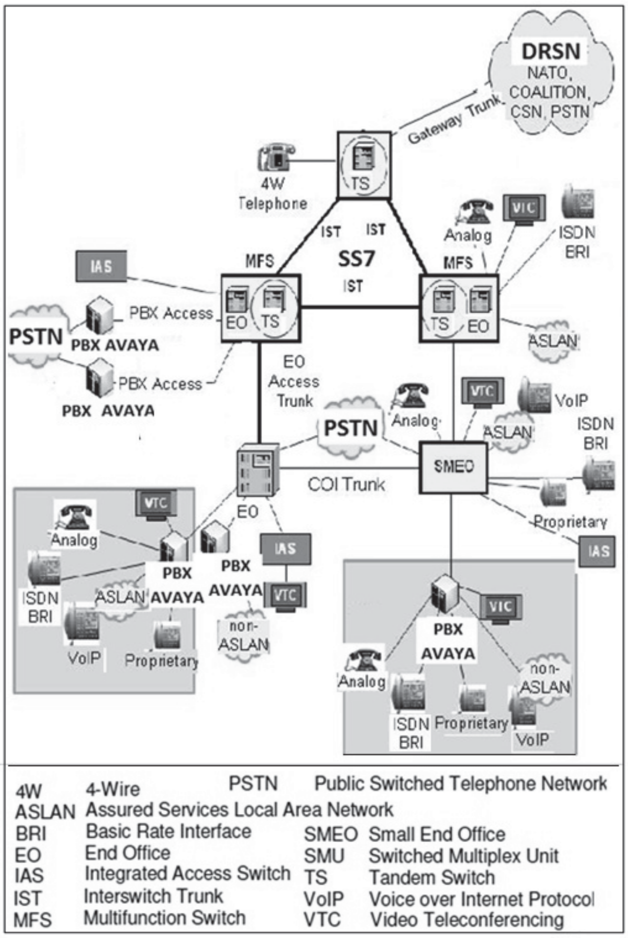

Fig. 4. DISN current state: PBX AVAYA testing

The most important step for DISN modernization is the replacing of channel switching electronic Multifunctional switches (MFS) by packet switching routers under AS-SIP signaling. The transition phase is based on Multifunctional SoftSwiches, MFSS (Fig. 5). According to the DoD Plan [7], there should be only 22 large-scale Cisco Systems multiservice provisioning platforms all around the world (Fig. 6).

The MFSS will be interfacing between the TDM and IP backbone network and will have much more complex circuit-switched based interfaces along with simple packet-switched based IP interfaces. The ISDN network uses ISDN User Part (ISUP) signaling protocol for the session/call control. So, MFSS will also need to provide ISUP-SIP interworking function (IWF). It is expected that TDM switching portion of the MFSS will be retired as soon as all users/systems migrate to IP, excepting H.323 as the leading protocol for video-conferencing and ISDN lines for DRSN.

DRSN (The Defense Red Switch Network) is a dedicated telephone network (Fig. 7), which provides global secure communication services for the command and control structure of the United States Armed Forces. Secure Terminal Equipment (STE) is the encrypted telephone communications system for wired communications and designed to use ISDN telephone lines. It has also a slot for Crypto PC Card and four buttons - to select the four priority levels. This allows users to make phone calls that get precedence over ones with a lower priority.

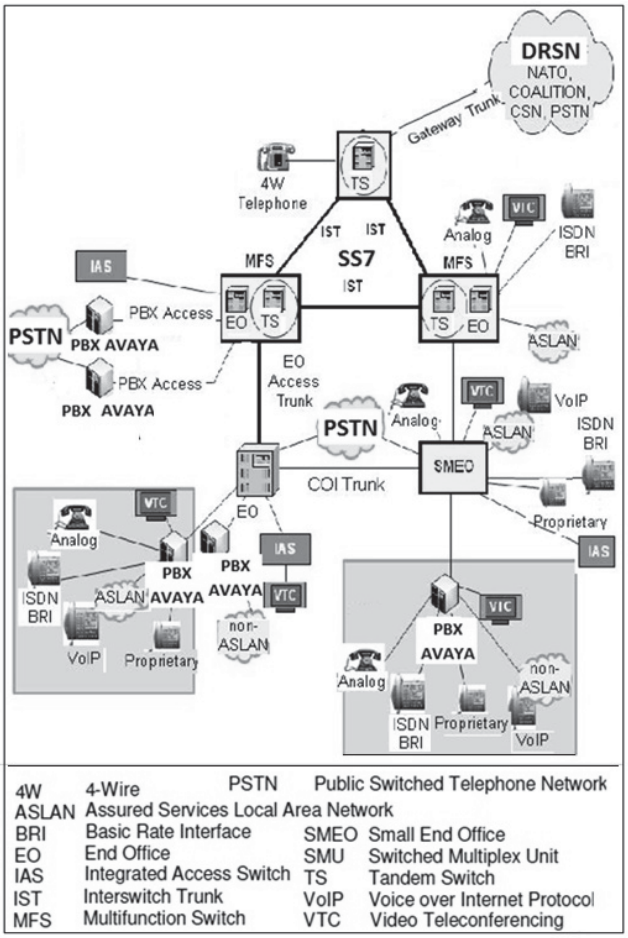

Fig. 5. Reference model for Multifunction SoftSwitch: the future AS-SIP based MFSS, but with the ISDN based DRSN included.

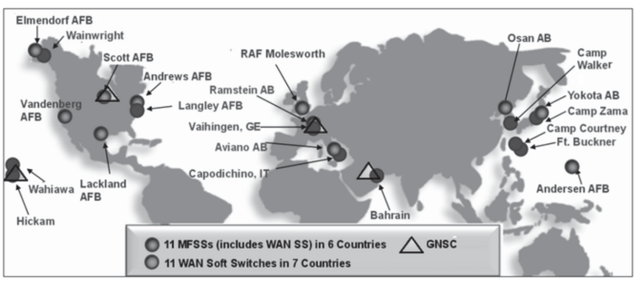

Fig. 6. DoD Plan: 22 large-scale Cisco Systems multiservice provisioning platforms all around the world and 4 Global Network Service Centers (GNSC)

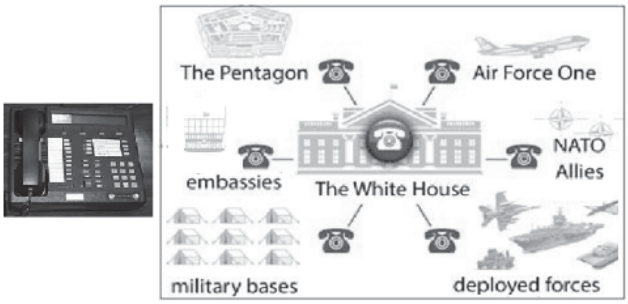

Fig. 7. DRSN: left – Secure Terminal (Red Phone), right – the basic DRSN subscribers.

C. GIG.3: Cyber-security

Meetings the challenges of cyber-security are fundamentally changing the plans of constructing DISN network [6]. There are many new connections (Fig. 8) be installed on the requirements of the CYBERCOM, created in 2010. CYBERCOM receives information about the situational safety from two centers: (1) DoD Component Network Operations and Security Center, NOSC and (2) DISA Network Operation Center, NOC. Cyber-security tasks we discuss in more detail in [8].

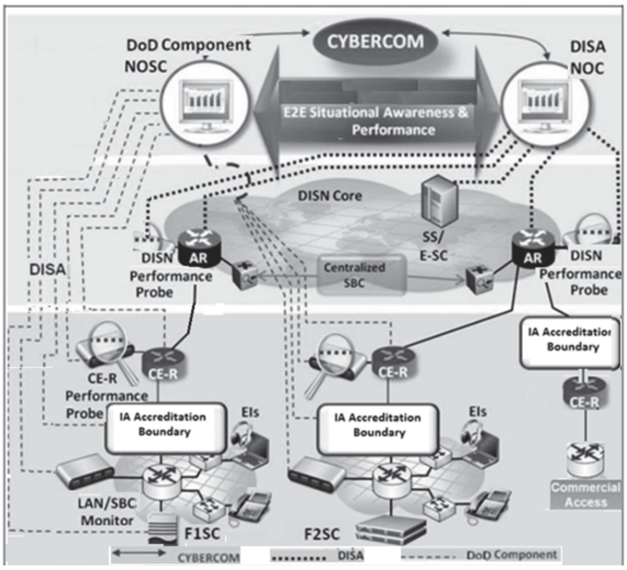

Fig. 8. DISN cyber-security network maintenance: UC services security requirements [6].

III. ON TELECOM SOFTWARE DEVELOPMENT

Let us look at the telecom software development history to follow how the complexity of telecom software is growing.

Service Creation Environment. Point out the attention to the Service Creation Environment (SCE) as a standardized means for AIN service software development (Fig. 3b). According to standards, SCE and SIB (Service Independent Block) library were invented to simplify software development and 3d parties work. There are 17 SIBs (in ITU standard) and 21 SIBs (from ETSI). In reality, telecom vendors had used up to 100 vendors specific SIBs. As a result, the AIN approach had a little success amongst software developers. The very idea failed: software developers were asked to know too many telephony details.

TINA project. The next one was TINA project. TINA Consortium started its work in 1993 and planned to end in 1997. The Consortium was supported by several main actors in the telecommunication world. The aim was to define a new software architecture. According to TINA promoters, the advantages of introducing CORBA / TINA based solutions within the IN are mainly related to the possible rationalization of the service aspects (e.g., integration of service management and control), to a higher level of interoperability between applications, to the ability to extend service related capabilities, scalability of the service platform, vendor independence, etc. In general, TINA concepts had planned for use in the following IN areas: Service Management, Service Data, and Service Control. Unfortunately, TINA concepts ended without implementation.

Parlay. After then was Parlay Group (founded 1998) that specified APIs for the telephone network. Parlay project ended unsuccessfully around 2007. In 2003, the Parlay Group released a new set of web services called Parlay X. These are a much simpler set of APIs intended to be used by a larger community of developers. Unfortunately, Parlay X ended without any wide use also.

FIWARE becomes then a fundamental pillar in the infrastructures of Smart Cities, as the different Generic Enablers build an architecture that can serve most of their needs [9]. Among many initiatives to adopt FIWARE as Smart City platform, 75 cities from 15 countries have joined the initiative "Open & Agile Smart Cities" (Smart City Expo World Congress, 17 to 19 November 2015, Barcelona, [10]). Each Smart City platform contains many GEs as well as some specific enablers (Fig. 9). In practice, the platform seems extremely complicated. Now, it all sounds very laudable, but has anyone ever heard of a successful project to come out of these FIWARE competitions?

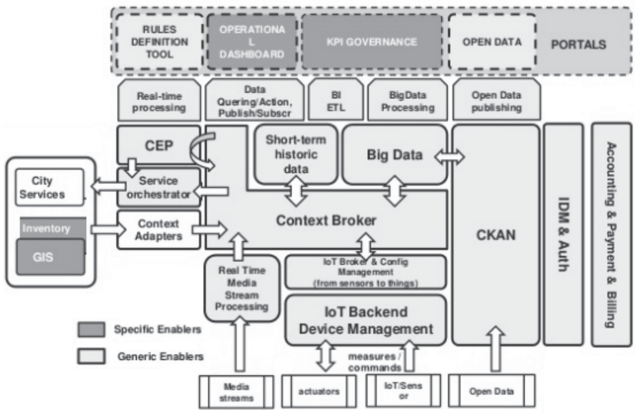

Fig. 9. Target Smart City platform [9]

Alliance for IoT Innovation. The European Commission has adopted on May 2015 the Digital Single Market strategy and has opened the door for large-scale proposals to improve the future of industrial development. The launch of the Alliance for IoT Innovation (AIOTI) in order to develop and support the dialogue and interaction among the various IoT players should be seen as a signal in this direction [11].

AIOTI is today the largest European IoT ecosystem - with 500 active members and more than 1500 high-level experts split into 11 thematic working groups The AIOTI project is a successor of FIWARE. Could it be more successful than FIWARE - is a question?

Joint Information Environment. The DoD's JIE framework will fundamentally change how implements, operates and defends its information systems. It is vital to the Department's efforts to increase network security, decrease IT costs, and enhance network resiliency. The technical and operational characteristics of the JIE are focused on the following technical characteristics:

Single Security Architecture, Federated Networks, Identity and Access Management, Data Center Consolidation, Software Application Rationalization and Server Virtualization, Desktop Virtualization and Thin-Client Environments, Mobility Services, and Enterprise Services.

The JIE requirements contain 52 volumes [12]:

- All Viewpoint – 2 volumes,
- Capability Viewpoint – 7,
- Data and Information Viewpoint – 3,
- Operational Viewpoint – 9,
- Project Viewpoint – 3,
- Services Viewpoint – 13,
- System Viewpoint – 13,
- Standard Viewpoint – 2.

Fig. 10 shows the common structure – how to develop DISA software (see also [13], [14]). The concept contains two parts: MBSE (Model-based Systems Engineering) and SysML (Systems Modeling Language). The very model MBSE is a collection of language SysML diagrams. As a result, there are three types of SysML documents useful for code translation:

- Service Offering Description,
- Technical Architecture Description,
- Engineering Design Specification.

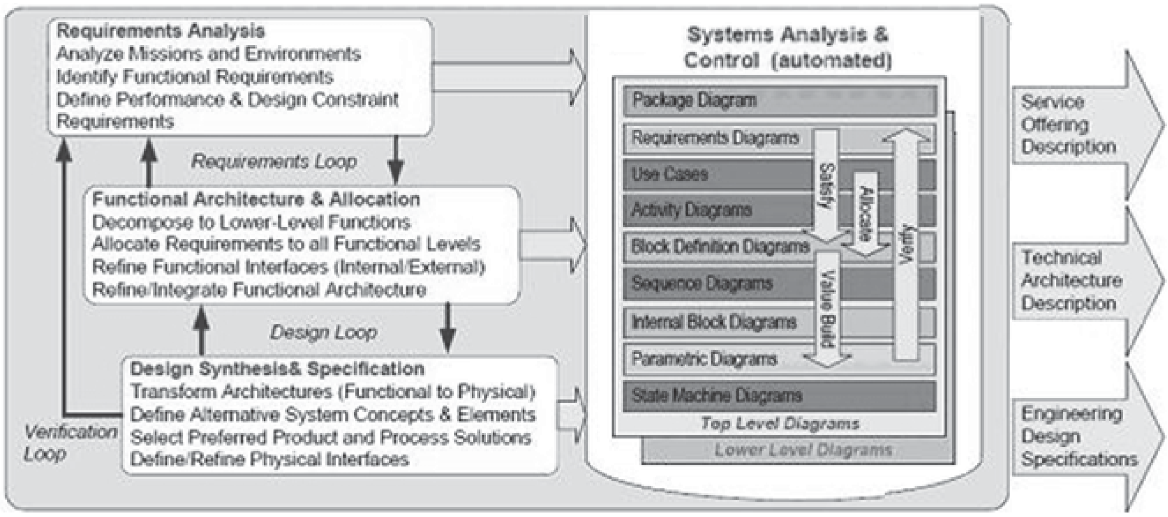

Fig. 10. MBSE (Model-based Systems Engineering) process tailored for DISA [12]

IV. ON SDN AND NVF TECHNOLOGIES

According to Recommendation ITU-T Y.3300, SDN is mapped to the 3-layers reference model (Fig. 11). Application Control Interface (NorthBound Interface) provides an application programmatic control of abstracted network resources. Resource Control Interface (SouthBound Interface) is used to control network resources. The SDN is highly promising now for Internet services and the All-IP move at all (e.g. using OpenFlow protocol). But it is so for SouthBound Interface part only. Meanwhile, the NorthBound Interface issues are not solved up to now.

Due to IT virtualization technology consolidating many network equipment types onto industry standard high volume servers, switches, and storage. Telecommunication network could be located in Datacentres, Network Nodes and in the end user premises, as illustrated in Fig. 12. It involves the implementation of network functions in software that can run on a range of industry standard server hardware [15].

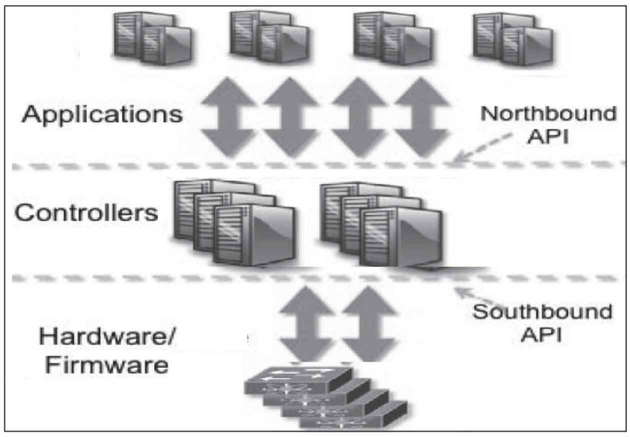

Fig. 11. SDN Stack

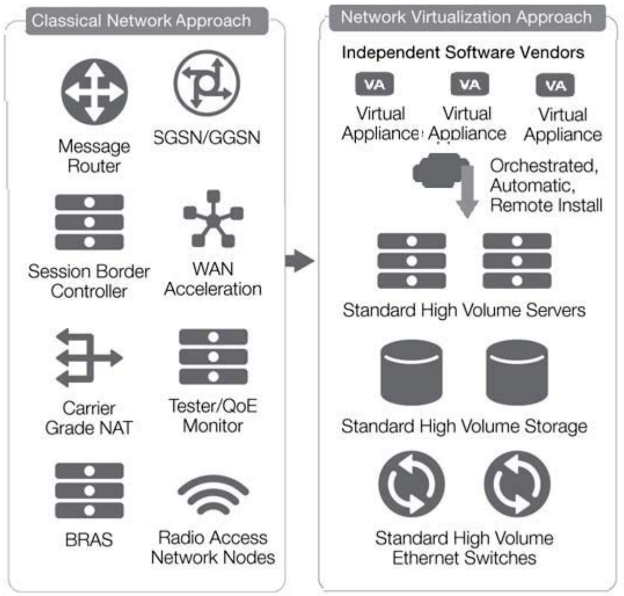

Fig. 12. The ETSI vision for NFV, which relies on COTS hardware and software delivered through the cloud [15]

The key goals of the ETSI NFV Working Group are to:

- Reduce equipment costs and power consumption.
- Improve time to market.
- Enable the availability of multiple applications on a single network appliance with the multi-version and multi-tenancy capabilities.
- Encourage a more dynamic ecosystem through the development and use of software-only solutions.

All of these benefits should come from the use of commercial, off-the-shelf (COTS) hardware that can be purposed for multiple telecom-related services that currently use proprietary hardware.

NFV is taking the software-defined networking (SDN) concept of the virtualization movement and adapting it to benefit the telecommunications application infrastructure. The major components of an NFV architectural framework are:

- Network Functions Virtualization Infrastructure (NFVI): subsystem, which encompasses Compute, Network, and Storage resources.
- Management and Orchestration: subsystem, which includes the Network Functions Virtualization Orchestrator, the Virtualized Infrastructure Manager (VIM) and Virtual Network Function Manager.
- Virtual Network Functions (VNFs): deployed in the NFVI.

What about NFV and SDN relationship, besides two above mentioned interfaces (NorthBound and SouthBound), there is Orchestration Interface – the interface between an SDN controller and an NFV Orchestrator. It might need to pass information between the two entities, such as topology information in both directions. The same kind interface is between an SDN application and an NFV Orchestrator.

From an SDN controller perspective, consider in more detail NorthBound Interface functions. From an NFV architectural framework perspective, the NorthBound Interface is the Application Control Interface provided by the SDN controller if that layer is embedded in the SDN controller, or it could be considered as an SDN application if it seats on top of the SDN controller. Fig. 13 below shows the different combinations of the different components of SDN controller.

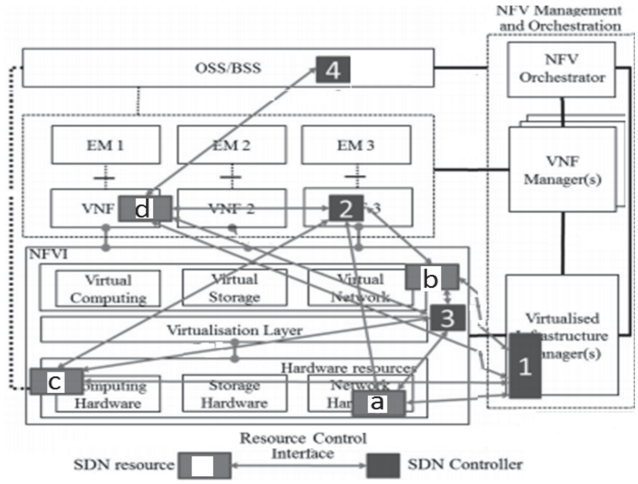

Fig. 13. SDN Resource Control Interface Options in NFV [15]

The similar kind of figures are given in [15] for SDN Controller/Application Orchestration, SDN Application Control, and SDN Controller to Controller Interface Options in NFV. All these many options have been implemented as Virtual Appliances developed by Independent Software Vendors (see Fig. 12). This is the basic idea for SDN and NVF relationship!

The complexity of NFV software is in many degrees more sophisticated than SDN one. Therefore, the future of NFV architecture is quite doubtful.

V. WHAT TO TEACH ENGINEERS AND PROGRAMMERS?

The training of telecommunication engineers for software development focuses on both telecommunication networks, such as the existing telephone network and networks for mobile communications, e.g., GSM and UMTS, and on data communication networks, e.g., local area networks (LANs) and the Internet. Furthermore, the traditional services, such as telephony, email, web-browsing, will be supplemented or even replaced by advanced multimedia services. The design and implementation of these advanced services on future networks, requires engineers who understand the technology involved and can determine the requirements of the services. We consider two MSc programs from many ones.

D. The MSc in Telecommunication program at Denmark Technical University

The MSc Telecommunication program at DTU [16] allows one to specialize in one of the four study lines: Optical Communication, Protocols and Network Technologies, Services and Software, Signals Transmission. We discuss the two of them – more to telecom software oriented - in detail.

Protocols and Network Technologies. In the courses that are part of this study line, the student will acquire a thorough understanding of network technologies and network architectures, e.g., communication protocols and core and access networks, so that he/she can work with the design and implementation of new services, networks, and protocols. The graduates will able to work in areas of network operation and management, capacity planning, network security and others.

Services and Software. This study line is oriented to design and implementation of telecommunication services and applications, with a focus on the enabling telecommunication and software architectures and mechanisms. Traditional software developers focus "just" on the application requirements from a user perspective. Careful design of innovative and advance (information and telecommunication-based) services and applications require a fundamental understanding of the underlying supporting telecommunication infrastructure and mechanisms, in order to streamline the design and implementation of services, and accommodate mutual requirements to fulfill users (final customers, subscribers, operators, or other) expectations.

Protocols and Network Technologies	Services and Software courses
Broadband Networks	Broadband networks
Distributed Systems	Social graphs and interactions
Network Security	Distributed systems
Data security	Network Security
Introduction to Optical Communication	Data Security
	User Experience Engineering
Optical Communication Systems	Digital Communication
Principles in Data Communication	Data Compression
Network planning	Digital Video Technology
Mobile Backhaul Networks	Synthesis Project for Telecom
Access and Home Networks	Advanced Image and Video Coding
Advanced Protocols	Network planning
Routing in Data Networks	Mobile Backhaul Networks
Telecommunication Services	Technology for Railways
Optical Networks	Access and Home Networks
	Telecommunication Services
	Programming Services for NGN
	SDN: Software-Defined Networking

Fig. 14. DTU MSc study courses

E. Illinois Institute of Technology. Master of Telecommunications and Software Engineering

A person holding a Bachelor of Science in Electrical Engineering, Computer Engineering, or Computer Science has the necessary broad background to undertake the MTSE-CS program [17].

Computer Science Prerequisites: Introduction to C++ Programming, Introduction to Advanced Studies.

Electrical and Computer Engineering Prerequisites: Circuit Analysis I, Circuit Analysis II, Signals and Systems, Introduction of Differential Equations.

Required Courses are Software Systems Architectures, Software Project Managements, Communication Engineering Fundamentals, Performance Evaluation of Computers and Communications Networks.

Elective Categories. Software Engineering: Object-Oriented Analysis and Design, Software Metrics, Software Testing and Analysis.

Telecommunications Systems: Computer Networks II: Network Services, Broadband Networks, Analytic Models and Simulation of Computer Systems, Computer and Communication Networks.

Telecommunications: Communications Systems Design, Modern Digital Communications, Coding for Reliable Communications.

VI. DISCUSSION AND FUTURE WORK

Telecommunication network architecture is undergoing a massive transformation now, primarily by Software Defined Network (SDN) and Network Functions Virtualization (NFV) technologies. But the hard question arises: is the NFV concept implementable from software developer point of view? Efficient software-based service life cycle depends on two key factors: short time to market and deployment flexibility. Time to market can be minimized through a homogeneous software environment that enables deployment on existing network infrastructure without the need for hardware modification. SDN and NFV software seems extremely sophisticated.

The following are some approaches to help in large software product development.

Metadata in SDN API. The ability to automate the work process for any API directly depends on the ability to obtain programmatically all information about this API. In other words, API should support metadata. In [19], we talked about several aspects of so-called Northbound API (see Fig. 11). As the most of modern APIs, it is some REST based solution (actually, solutions). Metadata support in REST services has its own characteristics.

As per related works, we should note the lack of the common (accepted by developers) standard for meta-data in REST models. Our analysis and our practical experience show that at moment the metadata instruments in REST models are not used. Often REST model operates without metadata, except the manual for developers only.

However, we think the formal description for any public API (including Northbound SDN API) is very important. Of course, the idea of a public API for network nodes is not entirely new. As a direct analogue, we can mention Parlay, for example. Telephone switching system (PBX) is also a network node, and Parlay (Parlay X) API plays the same role as the Northbound SDN API. It is a public API, provides an open programming interface to network hardware for application programs (for applied services). Lessons from Parlay's failure are important. Its main problems had connected with the difficulty of adopting API to the developers. The development of services with Parlay API, in fact, did not save development time. Vice versa, it takes usually more time than the deployment of non-portable (closed) APIs. Time to market is a key indicator for software development tools. The first (and possibly the main) purpose of using the metadata of some API is automation for software development. It directly affects the key indicator (time to market).

On Micro-services Architecture. The micro-services approach is a relatively new term in software architecture patterns. The micro-service architecture is an approach to developing an application as a set of small independent services [20]. Each of the services is running in its own independent process. Services can communicate with some lightweight mechanisms (usually it is something around HTTP) [21]. Such services could be deployed independently. Also, the centralized management of these services is a completely separate service too. It may be written in different programming languages, use own data models, etc. Developers may use, for example, JSON, XML, etc.

Of course, the proposed micro-services approach has got an own set of drawbacks. In practice, micro-services approach means for the developers the additional complexity of creating a distributed system. Testing is more difficult for distributed systems. Probably, it is one of the main problems – we must implement the inter-service communication mechanism. Very often, we will need some form of distributed transactions.

As soon as we talk about distributed systems and remote calls in micro-services architecture, the network part of the system becomes crucial. We would like to present some patterns and discuss the related challenges. The first communication pattern is obvious. Our application can use each service directly.

It is, no doubts, the most flexible way. Think, for example, about web server being able to call various services before rendering the output page for some particular request. The biggest problem, of course, is the potential delays for remote calls. The next step is almost obvious. We need to decrease the number of remote calls. It leads us to the various forms of cache and to the solutions, similar to transaction monitors in databases [22], middleware (3-tier) applications [23], etc. It shows in Fig. 15a. Note, that this pattern is more traditional for M2M (IoT) applications, because this gateway can also hide some limitations for legacy devices.

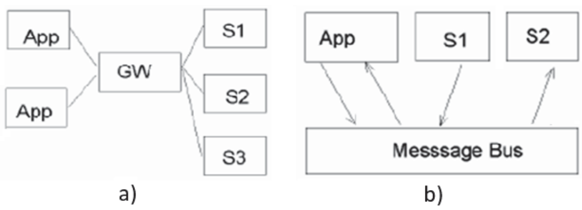

Fig. 15. a) A gateway for micro-services; b) Message Bus

The third pattern is some service-bus. It is suitable for M2M (IoT) applications due to the asynchronous nature of the most of the services. E.g., for the most of the sensors, data reading requests are asynchronous. Service (message) bus lets application post requests and reads response later (Fig. 15b). The service-bus deployment itself can use clustering and load balancing to improve scalability by distributing the workload across nodes.

API or DPI. In [24] there are described Data Program Interface (DPI) as an interface at the edge of an Internet of Things (IoT) device that exposes and consumes data. As seems to us Application Program Interfaces (API) have roots in telecom equipment. Yes, for telecom devices we need a way for transferring some commands (instructions). But IoT devices very often do not support commands (instructions). Many of sensors are just providing some data and nothing more.

Need for collaboration. The full potential of IoT could only be realized if service providers and vendors alike look at it as a customer-centric opportunity while remaining focused on the bigger picture [25]. Without this, IoT growth will be stunted and the market will become heavily fragmented, leading to security issues and vendor lock-in.

Data processing. It is a subject for the separate study, but no doubt that telecom education must include big data processing. We should not here only, that in our opinion data engineering (data architectures) is more important for telecom than data mining. So, in our opinion, big data for telecom is data engineering firstly.

REFERENCE

[1] Lokheed Martin Contarct https://www.infosecurity-magazine.com/news/lockheed-martin-gets-46-billion-contract-to/ Retrieved: Feb, 2017

[2] Leidos https://washingtontechnology.com/blogs/editors-notebook/2016/08/leidos-lockheed-deal-closes.aspx/ Retrieved: Feb, 2017

[3] Shneps-Shneppe, M. A. Namiot D.E. "Telekommunikacii dlja voennyh nuzhd: ot seti GIG1 k seti GIG2", *International Journal of Open Information Technologies* vol.2, no.9, 2014, pp. 9-17.

[4] DISA Joint Interoperability Test Command (JTE). Special Interoperability Test Certification of Avaya S8300D with Gateway 450. 17 Apr 2012.

[5] Department of Defense. Assured Services (AS) Session Initiation Protocol (SIP). Errata-1, July 2013.

[6] U.S. Army Unified Capabilities (UC) Reference Architecture (RA). Version 1.0. 11 October 2013.

[7] Department of Defense. Unified Capabilities Master Plan (UC MP), October 2011.

[8] M. Sneps-Sneppe, S. Seleznev, D. Namiot, V. Kupriyanovsky, "About the status of cybersecurity of critical infrastructure of the state", *International Journal of Open Information Technologies v*ol. 4, no. 7, 2016, pp.22-31

[9] FIWARE https://www.fiware.org/2015/03/25/fiware-a-standard-open-platform-for-smart-cities/ Retrieved: Feb, 2017

[10] J. Villasante. "FIWARE is making cities smarter" 05 January 2016 https://ec.europa.eu/digital-single-market/en/blog/fiware-making-cities-smarter/ Retrieved: Feb, 2017

[11] European Commission. Internet of Things Applications AIOTI. WG01 – IERC. Release 1.0. October 15, 2015.

[12] The Department of Defense. Strategy for Implementing the Joint Information Environment. September 18, 2013.

[13] Department of Defense. Information Enterprise Architecture. Unified Capabilities Reference Architecture. Version 1.0. January 2013.

[14] M. Sneps-Sneppe, D. Namiot, V. Suhomlin, "On creation of an unified information space for the society", *International Journal of Open Information Technologies* vol. 3, no. 2, 2015, pp.1-10.

[15] ETSI GS NFV-EVE 005 V1.1.1 (2015-12) Network Functions Virtualization (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework.

[16] M. Sneps-Sneppe, D. Namiot, "On Telecommunication Architectures: from Intelligent Network to Network Functions Virtualization" In *8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT2016)*, 2016, pp.251-256

[17] Technical University of Denmark http://www.dtu.dk/english/Education/msc/Programmes/ telecommunication/ Retrieved: Feb, 2017

[18] Illinois Institute of Technology https://science.iit.edu/programs/graduate/master-telecommunications-and-software-engineering-mtse-cs/ Retrieved: Feb, 2017

[19] M. Sneps-Sneppe, D. Namiot, "Metadata in SDN API for WSN." In *New Technologies, Mobility and Security (NTMS), 2015 7th International Conference on. IEEE*, 2015, pp.1-5.

[20] D. Namiot, M. Sneps-Sneppe, "On Micro-services Architecture", *International Journal of Open Information Technologies* ISSN vol. 2, no. 9, 2014, pp.24-27.

[21] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things." *Architecting the internet of things*. Springer Berlin Heidelberg, 2011, pp.1-24.

[22] Dayal, U., Garcia-Molina, H., Hsu, M., Kao, B., & Shan, M. C. "Third generation TP monitors: A database challenge". *ACM Sigmod Record* vol. 22, no. 2, 1993, pp. 393-397.

[23] Reijers, N., Lin, K. J., Wang, Y. C., Shih, C. S., & Hsu, J. Y. "Design of an Intelligent Middleware for Flexible Sensor Configuration in M2M Systems", In *SENSORNETS*, 2013, pp. 41-46.

[24] D. Namiot, M.Sneps-Sneppe, "On software standards for smart cities: API or DPI", In *ITU Kaleidoscope Academic Conference: Living in a converged world-Impossible without standards? Proceedings of the 2014*, 2014, pp.169-174.

[25] M. Sneps-Sneppe, D. Namiot, "On Open Source Smart City Platform: How to Get It?" In *Proceedings of the I International scientific conference "Convergent cognitive information technologies" (Convergent'2016)*, 2016, pp. 233-241.