IMS Automated Audit and Configuration: Parameters Audit SmartApp (PAS)

Meltem Yiğit, Onur Görgün, and Hüseyin Gören
Nokia, Istanbul, Turkey
Email: {meltem.yigit, onur.gorgun, huseyin.goren}@nokia.com

***Abstract*—In this study, we propose a brand new automated audit and configuration architecture, Parameter Audit SmartApp (PAS), for IP Multimedia Subsystem (IMS). PAS solution lowers external dependencies to proven development frameworks, and offers new development environment. It also offers selective and critical parameter auditing, configuration abilities, centralized management of Network Element (NE) configuration schemas, and deployment availability on end-user terminals. We deploy PAS on telecommunication giant's IMS network, and share the observed improvements in terms of both processing time and cost-efficiency.**

***Index Terms*—IMS, automated audit, automated network element configuration**

I. Introduction

IP Multimedia Subsystem (IMS) [1] is a network architecture framework developed as an end-to-end seamless infrastructure for transmission of voice, video and data over IP networks. Thanks to the robustness and elasticity provided by IP infrastructure, IMS supports multiple access types – including GSM, WCDMA, CDMA 2000, wire line broadband access, WLAN, 4G and 5G networks and fixed line. IMS has been built upon modular architecture that allows integration of different components and modules from various solution providers. When network infrastructure of a telecom operator considered, IMS would be a prominent solution in terms of variety of service, solution and vendors on continuously growing complex network topology [2]. In this context, IMS can be used to manage all service related issues such as Quality of Service (QoS), Charging, Access Control, User and Services Management [3]-[5].

The novel architecture of IMS system eliminates vendor/product dependency on any network by enabling easy integration of any systems or services from any vendors [6]. Product and vendor variety in IMS network along with increasing number of nodes results network element management problem which composed of network element discovery, management, configuration update processes. Additional tools have been built to tackle this problem and therefore each IMS platform provider either implemented any of well-known framework or developed its own solution.

Manuscript received March 24, 2016; revised June 2, 2016.

Nakina is one of the generic network integrity framework providers which provide on-the shelf network adapters any vendors providing IMS ready network elements. The framework works on "collect", "compare", "configure" and "protect" cycles. The cycle starts with collecting of configuration data from all physical or virtual elements or functions. Then data is compared with desired configuration parameters or the ones in database. If there are any discrepancies in configuration between network elements and database, selected network elements are re-configured with desired parameters as a third step. Finally, it's deployed and then protected to ensure integrity of the system. [7]. Such instant framework enables IMS producers enough agility to develop its own solutions quickly however, it puts some burdens such as customization of the product according to different network architecture, performance on complex network structures and adaptability of the system for future configuration changes.

To sum up, it's impossible to think about IMS services independently from service environments and hence service elements [8]. Having an efficient IMS network strictly depends on efficient network audit platform. Network audit affect performance of IMS product because, complexity of network topology, discovery of redundant network functions and parameters causes performance delays when a complex network considered.

Therefore, an efficient brand new network element auditing application, Parameters Audit SmartApp (PAS) is needed to improve functionality and agility of IMS product. The paper is organized to present the PAS tool. We begin with the current automated audit and configuration solution which is developed by Nokia, formerly Alcatel-Lucent, and, then, present our proposed solution. Next, we provide performance review for the proposed solution tested in another telecommunication giant. Finally, we conclude with critics on the solutions and future work.

II. Automated Audit and Configuration

IP Multimedia Subsystem (IMS) has been specified as service delivery platform for 3G networks that is used to transmit voice, video and data over IP. IMS is an architectural framework to deliver Internet Protocol (IP) Multimedia Services, supports almost all access types which are commonly used in current telecommunication infrastructures such as GSM, WCDMA, CDMA2000,

doi: 10.18178/ijeee.5.1.15-20

wire line broadband access, WLAN, 4G and 5G fixed line. IMS supports specialized network elements that are assembled in different network architectures in order to be adapted to different customers with different network architecture.

To perform configuration from a single point for all network equipment on a telecommunication network is a major problem in terms of value of specified parameters, whether compatible with the topology of the current system and whether configured redundant functions that might affect the performance of the system when each network element has hundreds/thousands of configuration parameters. Hence, these parameters are to be audited to ensure the integrity of the network.

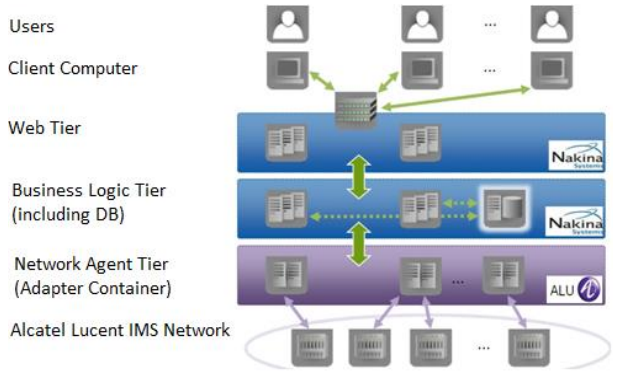

Figure 1. Nokia's current IMS network automated audit architecture.

Current GSAT-IMS solution which is developed by Nokia, supports various network elements with different configuration parameters for recognized telecommunication giant. As depicted Fig. 1. Current GSAT solution resided in Network Agent Tier and basically consists of Network Element (NE) integration points, so called adapters. NE communicators are developed based on the Nakina Framework API and provisioned into the solution.

PAS-IMS solution, developed for the operators looking for quality and efficiency on network, includes a large number of network elements that perform different tasks. These network elements are combined in different network architecture and the parameters they have are checked for providing automation to compare the actual values with the reference values of these parameters and hence increase the network throughput. According to the requirements of the PAS-IMS solution, specific values are assigned for these parameters. The number of parameters, namely configuration parameters should be hundreds of thousands in terms of the complexity of network solution. Controlling of expected values of the parameters takes a long time so that managing control becomes the imperative to automate the software. In addition, compliance with the parameters of the reference value is completed in a very short time in PAS-IMS solution.

Proposed system architecture, Data Collection Module (DCM) integrated GSAT solution, is depicted in Fig. 2.

DCM is the core part of PAS solution which aims to collect data through communicators. In Fig. 2, 3rd party integration interface and new DCM API interface are presented as green and purple indicators, respectively. Thus, the adapter module of the network elements will be harmonized with specifications of our PAS-IMS solution. Therefore, current new GSAT architecture is become independent from Nakina Framework with PAS product. However, the implementation of the integrated GSAT-DCM is expected to improve the customer's operational performance. Proposed architecture has additional development cost and also requires additional efforts of customer on operational processes but maintenance/repair costs are reduced in terms of rapid deployment of communicator on live IMS network.

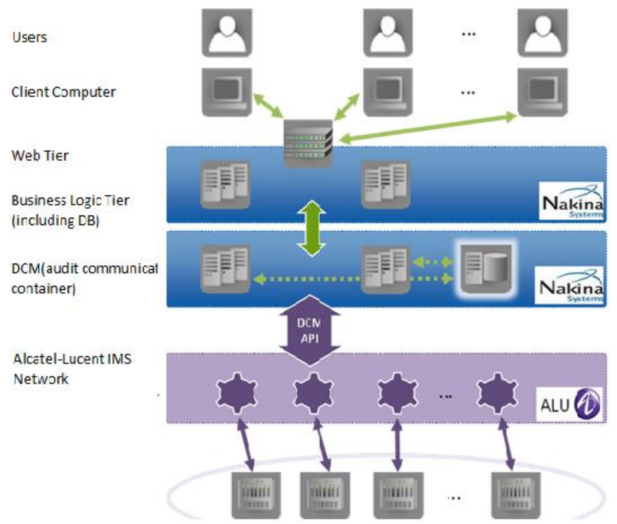

Figure 2. PAS, automated audit architecture for IMS network.

The primary purpose of the Parameter Audit SmartApp (PAS) is to perform audits of configuration parameter values on deployed NEs (Network Elements), to make sure that they are configured as expected. Automation of verification is useful, since the number of configuration parameters per NE typically ranges in the low to mid hundreds, sometimes even in the low thousands, and an entire network represents tens to hundreds of NEs in the core and edge network (thousands or more in the Access). When an audit is launched from PAS, current parameter values are retrieved from the actual deployed NEs through a set of software interfaces, and are compared to reference values, given as Reference Data Sets (RDS) which are the expected values of these parameters, extracted from Reference Data File (RDF) as illustrated in Fig. 3. RDF is per NE low level data design document built using the appropriate Parameter Definition Document (PDD). Each PDD record is used to identify each parameter's name, description, range, default value, and retrieval mechanism. Table I illustrates a snapshot from an example PDD.

TABLE I. SNAPSHOT FROM A PDD SAMPLE

Parameter	Description	Range	Default Value
HSS Host Name	Host Name for HSS indetity	5-64 (Varstring)	n/a
Keep CSCF	This data is to determine if HSS should keep stored S-CSCF….	Not Keep Keep	Keep

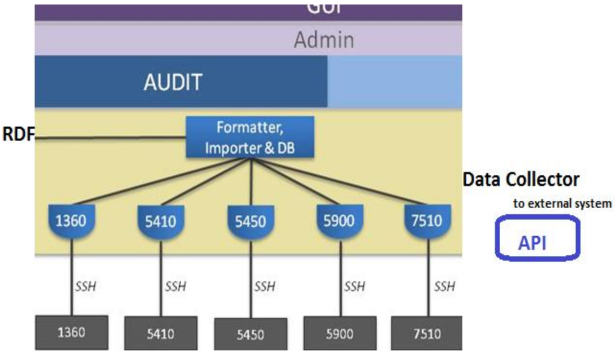

Figure 3. Modular PAS architecture.

RDSs are stored in the PAS database, and written in user-manageable format. Upon audit completion, an audit report is shown, allowing end-users to see all retrieved configuration parameter values, and whether they match the expected reference values in the RDS or not. Audit results can be exported as CSV or HTML files. Also, any previously retrieved data and previous audit report can be browsed at any time via audit history module by using PAS GUI, as long as it is stored in the tool database.

PAS solution provides a one-stop shop for NE configuration verification across the board- multi-NE, multi-vendor, categorization (regions, types). The specific per-NE adaptation work is done in the communicators, and a common view and management style is available for the audits and the corrective actions (when fix feature is available).

PAS solution ensures that the network is functional and optimized: an incorrect parameter value (timer values, filtering options, protocol settings, port maximum bandwidth, etc.) in an NE may lead to customer traffic flows being interrupted or slowed down, especially after a network-wide upgrade. Even worse, an incorrect value could lead to silent outages (with denial of service to subscribers), nationwide outages and/or service interruptions. Regular auditing is an essential outage prevention measure, which avoids (or at least limits) service-impacting misconfigurations. Well-functioning services and networks result in higher customer satisfaction. In addition, a regularly audited network will consume less administration resources (decreased OPEX: less overall troubleshooting effort).

The primary advantages of PAS architecture over the current architecture are as follows:

- *PAS solution automates parameter audit procedure*: this is a time-consuming, tedious and error-prone task if done manually. Network operators can setup their audit scopes and reference values data sets, and then repeatedly run software-based on-demand and recurring audits with PAS.
- *PAS solution allows parameter selection:* After defining the audit, a user is able to select parameters – that can be critical parameters from assigned dataset of each NE in scope. Depending on the preference of user, parameters are listed with available service impacts options. Parameter selection provides time gain and focus on critical parameters.
- *PAS solution enables configuration of NEs:* Users can configure a network element by generating automatic configuration scripts from a reference data set. The reference data must have been imported into the system before running the provisioning service.
- *Gold Standart Services (GSS*): GSS scope includes recommendation, documentation and maintenance of specific, customized parameter settings for IMS NEs from initial deployment through the life of the network/project. PAS benefits from GSS as; i) increase in OPEX savings, ii) optimal settings of network parameters, iii) complete documentation and maintenance, iv) change management process, and v) directly fed to PAS. Fig. 4 illustrates the overview of GSS and role in the proposed process flow.
- *PAS can be installed as standalone software on the local system:* Despite the central installation, PAS is extended to work on client computers for onsite audit and configuration purposes by site-engineers.

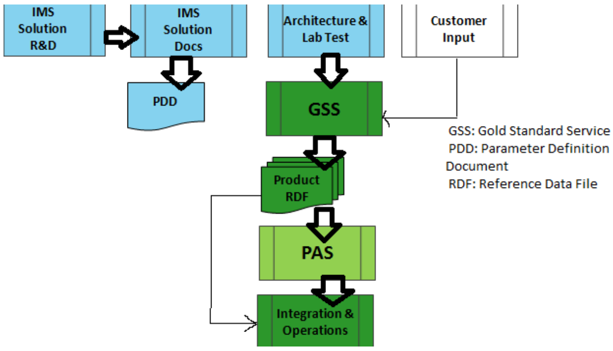

Figure 4. Overview of configuration process flow.

III. Evaluation of the Proposed Solution

A. Evaluation Metrics

We analyze the dimensioning in terms of CPU and Memory load. The following ideas are followed to create meaningful performance evaluation metrics.

- Measurements (or assumptions) on CPU Utilization and Memory for PAS audits (single audit, or multiple audits executed simultaneously), derive the approximate per-Audit:
  - load on one Core
  - Memory load on one server
- Per-server analysis: derive CPU-limited and Memory-limited maximum number of concurrent Audits for one server, and then overall maximum number of concurrent Audits for one server.
- Cluster analysis: assuming a server cluster configuration (multiple servers with multiple cores each) and assuming a per-User Audit load, derive the maximum number of simultaneous Audits and the maximum number of Users for that cluster configuration.

Assuming $U_{srv,aud=1}$ is a measured CPU load (average CPU load across all cores) on a server for the execution of one Audit, the maximum number (limited by Processing) of Audits that could run concurrently on this server is estimated as follows (linear approximation):

$$A_{srv,proc} = \frac{1}{U_{srv,aud=1}}$$

e.g., if one Audit resulted in a 5% CPU load on one Server, that Server could handle 1/0.05 = 20 concurrent Audits, which would theoretically result in a CPU load of 100%.

Number of Cores in a Server (multiply number of CPUs per server by number of Cores per CPU):

$$CORE_{srv} = CPU_{srv} \cdot CORE_{CPU}$$

Theoretical Core load for one Audit (i.e., run on a single Core) would be (number of Cores in a server multiplied by the overall average load):

$$U_{core,aud=1} = CORE_{srv} \cdot U_{srv,aud=1}$$

e.g., if one Audit resulted in a 1.5% overall load on a 8-core server, then one single Core executing the same Audit would have (linear approximation) a Utilization of 8*1.5 = 12%.

Assuming $M_{srv,aud=1}$ is a measured Memory load on a server for the execution of one Audit, the maximum number (limited by Memory) of Audits that could run concurrently on this server is estimated as follows (linear approximation):

$$A_{srv,mem} = \frac{1}{M_{srv,aud=1}}$$

e.g., if one Audit resulted in a 0.7% Memory load on one Server, that Server could handle 1/0.007 ≈ 142.86 concurrent Audits, which would theoretically result in a Memory load of 100%.

For one single Audit with a measured $U_{srv,aud=1}$ CPU load and a measured $M_{srv,aud=1}$ Memory load on a Server, the maximum number of concurrent Audits which could run on this Server (limited by either CPU capacity or by Memory capacity) would be the minimum of the CPU-limited maximum number of Audits ($A_{srv,proc}$) and the Memory-limited maximum number of Audits ($A_{srv,mem}$):

$$A_{srv} = min\left(A_{srv,proc}, A_{srv,mem}\right)$$

e.g., if the CPU-limited maximum number of Audits were 42 and the Memory-limited maximum number of Audits were 155, then the overall maximum would be 42; in this example, the CPU limitation would be more stringent than the Memory limitation.

If measurements are made with multiple concurrent Audits ($aud=N$, assuming these Audits are similar in size and scope), then the previous equations become (generalization for measured $U_{srv,aud=N}$ CPU load and measured $M_{srv,aud=N}$ Memory load):

Maximum processing-limited number of concurrent Audits:

$$A_{srv,proc} = \frac{N}{U_{srv,aud=N}}$$

e.g., if 3 Audits resulted in a measured 6% CPU load on one Server, that Server could handle 3/0.06 = 50 concurrent Audits, which would theoretically result in a CPU load of 100%.

Theoretical Core load for one Audit (number of Cores in a server, multiplied by the overall average load, divided by the number of measured concurrent Audits):

$$U_{core,aud=N} = \frac{CORE_{srv} \cdot U_{srv,aud=N}}{N}$$

e.g., if 3 Audits resulted in a 11.5% overall load on a 4-core server, then one single Core executing one of these Audits would have (linear approximation) a Utilization of 4*0.115/3 ≈ 15.33%.

Per-Audit memory utilization (linear approximation):

$$M_{srv,aud=1} = \frac{M_{srv,aud=N}}{N}$$

Maximum memory-limited number of concurrent Audits:

$$A_{srv,mem} = \frac{N}{M_{srv,aud=N}} = \frac{1}{M_{srv,aud=1}}$$

e.g., if 3 Audits resulted in a 4.2% Memory load on one Server, that Server could handle 3/0.042 ≈ 71.43 concurrent Audits, which would result in a Memory load of 100%.

The overall maximum number of concurrent Audits is still derived using the same formula:

$$A_{srv} = min\left(A_{srv,proc}, A_{srv,mem}\right)$$

For server cluster case, o take advantage (i.e., to increase measurement accuracy) of several CPU and memory measurements, we average them.

For CPU loads, we average the computed per-Core per-Audit Utilizations (i.e., not the measured server Utilizations $U_{srv,aud=1}$ and $U_{srv,aud=N}$).

$$U_{core,aud} = \frac{U_{core,aud=1} + U_{core,aud=3}}{2}$$

e.g. measurements are taken for this case where N=1 and N=3.

For Memory loads, we average the per-Audit Memory utilizations (related to Servers, not per Core):

$$M_{srv,aud} = \frac{M_{srv,aud=1} + \frac{M_{srv,aud=3}}{3}}{2}$$

Number of Cores in the Cluster (multiply number of CPUs per server by number of Cores per CPU and by number of Servers in the Cluster):

$$CORE_{clu} = CPU_{srv} \cdot CORE_{CPU} \cdot NSERV$$

Maximum processing-limited number of concurrent Audits in the Cluster (number of Cores multiplied by the maximum Processing-limited per-Core number of Audits, i.e. divided by the per-Core per-Audit Utilization):

$$A_{clu,proc} = \frac{CORE_{clu}}{U_{core,aud}}$$

e.g., if one Audit resulted in a load of 9.2% on a single Core and if a Cluster has 5 servers, each with two 4-core CPUs, then the maximum Processing-limited number of Audits for that Cluster would be estimated as (2*4*5)/0.092 = 40/0.092 ≈ 434.78.

Maximum memory-limited number of concurrent Audits in the Cluster (number of Servers multiplied by the maximum Memory-limited per-Server number of Audits, i.e. divided by the per-Audit Memory Utilization):

$$A_{clu,mem} = \frac{NSERV}{M_{srv,aud}}$$

e.g., if one Audit resulted in a 2.7% Memory utilization on a Server and if the Cluster had 14 servers, then the maximum Memory-limited number of Audits for that Cluster would be 14/0.027 ≈ 518.52.

The overall maximum number of Audits for the Cluster is derived using the same formula as for a single server:

$$A_{clu} = min(A_{clu,proc}, A_{clu,mem})$$

This is also the maximum number of simultaneous Users executing Audits (if they each execute at most one Audit at a time).

B. Deployment and Results

The proposed PAS solution is deployed to telecommunication giant's IMS network. Target network consists of multiple regions, where they can be reachable through SSH interface. PAS solution has access to them through both HTTP and SFTP interfaces.

TABLE II. MEASUREMENTS WITH ONLY 1 CONCURRENT AUDIT, ON 1 SERVER

CPU Utilization	11.07%
Number of CPUs	1
Number of CPU cores (available)	2
Number of CPU cores (assigned)	2
CPU core utilization	22.14%
Max Number of Audits	9.03
RAM utilization	11.57%
Max Number of Audits (1 server, based on RAM capacity)	8.64
Max Number of Audits(1 server, based on CPU and RAM capacity)	8.64

TABLE III. MEASUREMENTS WITH ONLY 2 OR MORE CONCURRENT AUDIT, ON 1 SERVER

Number of Audits	8
CPU Utilization	35.05%
Number of CPUs	1
Number of CPU cores (available)	2
Number of CPU cores (assigned)	2
CPU core utilization	8.76%
Max Number of Audits	22.82
RAM utilization	15.82%
RAM utilization (Theoretical per-audit average)	1.98%
Max Number of Audits (1 server, based on RAM capacity)	50.57
Max Number of Audits(1 server, based on CPU and RAM capacity)	22.82

TABLE IV. EXTRAPOLATIONS FOR (HORIZONTAL) SCALABILITY ANALYSIS

Number of servers	1
Number of CPUs	4
Number of CPU cores (available)	4
Number of CPU cores (assigned)	16
CPU core utilization	15.45%
Max Number of Audits(based on CPU capacity)	103.55
RAM utilization	6.77%
Max Number of Audits(based on RAM capacity)	14.76
Max Number of Audits (based on CPU and RAM capacity)	14.76
Number of servers	1
Number of CPUs	4

After successful deployment of PAS to customer's network, initial results revealed that approximately 40% and 10% reduction in NE configuration times of SDM & IeCCF, and ICS, ISC & CTS NE which might be considered as remarkable. It is also observed that configuration quality improves noticeably and configuration interval reduces slightly. Table II, Table III, and Table IV illustrate more detailed analysis results.

From economic point of view, the license cost of the foreign origin third-party software solutions is eliminated with PAS-IMS product. Thus, more economical and more capable solution will be recommended to the IMS customers.

IV. CONCLUSION AND FUTURE WORK

We have proposed a brand new solution for automated network audit and configuration for IMS networks. We have investigated our current IMS solution and removed 3rd party dependencies to external frameworks. Our proposed solution is independent of specific framework and can be integrated to any business flow via API that we have provided. Additionally, the proposed solution offers remarkable performance improvement in terms of audit and configuration time, and per license prices.

Generalization studies have been conducted in order to become independent from the clients of the PAS project. In other words, PAS communicators will be able to work on the same set of devices even if the configuration schema is different.

Data collection, inventory management network element, reference data set management and comparison modules are fully integrated in the current architecture of PAS core software. Due to management of operational challenges in target network, PAS is planned to be available for distributed installation. In distributed installation environment, DCM is planned to be in customer network for data collection. Other operational PAS modules are established in the center of the PAS server in Nokia network. Hence, time and operational sense in workload is expected to be lowered.

REFERENCES

[1] IMS, TS 23.228, 3GPP, Technical Specification Group Services and System Aspects, IP Multimedia Subsystem (IMS), Release 11, June 24, 2013.

[2] H. K. Dialexia and J. C. Gregorie, "IMS for enterprises," *IEEE Communications Magazine*, pp. 68-75, July 2007.

[3] A. Liotta and L. Lin, "Managing P2P services via the IMS," in *Proc. IEEE International Symposium on Integrated Network Management*, May 2007.
[4] G. F. Mao, A. Talevski, and E. Chang, "Voice over internet protocol on mobile devices," in *Proc. 6th IEEE/ACIS International Conference on Computer and Information Science*, July 2007, pp. 163-169.
[5] M. T. Masonta, O. J. Oyedapo, and A. M. Kurien, "Mobile client for the next generation networks," in *Proc. Third International Conference on Broadband Communications, Information Technology & Biomedical Applications*, 2008, pp. 274-279.
[6] D. Geer, "Building converged networks with IMS technology," *Computer*, vol. 38, pp. 14-16, Nov. 2005.
[7] Nakina Systems. Network Integrity Defined. [Online]. Available: http://nakinasystems.com/wp-content/uploads/downloads/2015/01/Nakina_Network_Integrity_Applications_Brief-ver-1.0.pdf
[8] E. Bertin and N. Crespi, *IP Multimedia Subsystem (IMS) Handbook,* CRC Press, 2008, ch. 1, pp. 1-24.

Meltem Yigit in Meltem Yiğit was born in Istanbul, Turkey, in 1987. She graduated from Bahcesehir University with a bachelor's degree in Computer Engineering, in 2011. Since April 2012, she has been working for Nokia formerly Alcatel-Lucent Teletas, as a Senior System Integration Engineer.

Onur Görgün was born in Istanbul, Turkey, in 1980. He received the B.S. and M.Sc. degrees in information technologies from Isik University, Istanbul, Turkey, in 2005, and 2008 respectively, and still continues his Ph.D. studies in computer science in Isik University, Istanbul, Turkey.

In 2005, he joined the Department of Information Technologies, Isik University, as a Research and Teaching Assistant, and in 2008 moved to computer engineering department for the same position. Since December 2012, he has been with the Nokia Research and Development Department, formerly Alcatel-Lucent Teletas, as an Expert Integration Engineer.

Hüseyin Gören was born in Izmir, Turkey in 1981. He received B.S. degree from Industrial Engineering at Bilkent University and obtained M.S degree from Science and Technology Policies at METU.

He has worked for TUBITAK, as junior expert and national project coordinator of Eureka programme between 2004 and 2010. Then, he worked for Avea as research and development manager until 2010. He held several positions at Datateknik and managed project management and pre-sales teams between 2011 and 2014. Huseyin Goren joined Nokia, formerly Alcatel-Lucent-Teletaş in 2015 as research and development manager and still holds this position.