AUTOMATING TELECOM EQUIPMENT FOR CLOUD INTEGRATION

Sorin ZAMFIR, Titus BALAN, Florin SANDU
"Transilvania" University, Brasov, Romania
DOI: 10.19062/1842-9238.2015.13.3.19

Abstract: *Integration of equipment in the cellular networks was always considered a challenge mainly due to the heterogeneous nature of architectures and compatibility problems that may arise from them. Testing and prototyping new services is always preceded by a big effort with the environment setup and while communication standards aim to provide a very strict guideline when it comes to offering telecom services they fail to establish a unified way of how these are provisioned within different equipment. As a result most equipment vendors offer custom equipment administration solutions that are better known as Operating Support Systems (OSS) that impose specific know-how and deal with proprietary provisioning workflows. In this article we present a brief survey of the operator challenges identified with the integration of telecom equipment. We explore the possibility of offering a Cloud based model of cellular network equipment, to help third party vendors achieve faster time to market when releasing new services. Lastly, we propose our vision towards a unified provisioning and aggregation system of both new and legacy traditional telecom equipment by abstracting proprietary provisioning workflows in a vendor-agnostic way.*
Keywords: *telecom-cloud, automation, IaaS*

1. INTRODUCTION

The telecom market has evolved to a liberalized model that imposes rough competition between classical service providers and has also a huge impact on the Telco manufacturing ecosystem. While the number of services in the mobile world has exploded in the recent years this has yet the effect that operators would have hoped on the ARPU (average revenue per user).

A recent study by the EU commission shows that the average revenue per user has been dropping on a constant basis in the last 4 years from an average of 211.5€/year in 2010 to 171€/year in 2013. [1]

As a result we cannot fail to observe how operators struggle to find new ways of monetizing the already existing resources (usually characterized by big CAPEX) and how network equipment vendors continue to push to in-house solutions as a method to protect their own market share.

In fact compatibility between network equipment is driven mainly by the 3GPP alliance but this can only address specific parts of a network.

The provisioning and configuration aspect remains something sporadically touched by standard bodies.

There are however some bold initiatives, especially coming from the network operators and test equipment vendors ecosystem (NTAF-Network Test Automation Forum) that propose a uniformed way of administration and configuration of the telecom systems by abstracting equipment based on the functionality and capability they provide [2].

The NTAF initiative relies on the premises that in the future vendors will make their equipment compatible with the standard.

Non-compatible equipment can also be integrated but these must be proxy-ed by 3rd party software that acts as translator between NTAF standard and the non-standard administration interfaces.

In order to fully understand the problem we are trying to address, below one can find a summary of the issues and challenges network operators are facing:

1. Telecom networks are heterogeneous which means that a topology usually contains equipment of different generations from different vendors.

2. Development of Telco equipment is done in a closed-source environment and, as a result, administration and control interfaces are in most cases proprietary.
3. The equipment management is usually done with custom solutions that are provided by the equipment vendor. Even compatibility with different generations of the same vendor is quite problematic.
4. „Leasing" un-used Telco equipment to reduce OPEX and increase the ROI (return of investment) is limited to specific scenarios and introduces security vulnerabilities in the network by allowing direct contact of 3rd parties to the equipment.

In order to address these concerns the authors propose a framework for standardizing the remote configuration and aggregation of Telco equipment so these can be later offered as IaaS (Infrastructure-as-a-Service) by the operators, similar to traditional Cloud services.

The idea behind it is to ensure a generic provisioning interface that hides the implementation complexity through a set of standard capabilities that are vendor independent.

The real challenge is to ensure compatibility between different generations of equipment that even belong to specific functional classes.

In order to integrate "legacy" elements with Cloud telco elements, the actual deployment of network configurations must be done automatically to ensure an end-to-end solution; in the end the user must not be concerned by specific provisioning tasks and should only be focused on the capabilities it requests.

2. INTEGRATING THE EQUIPMENT

Existing network management systems have been developed with focus on accessibility and ease of use in mind to mitigate provisioning tasks. While traditional solutions such as Nokia's NetAct™[3] or Erricson's NetOp™ [4] evolved in such a way that provisioning can be somehow automated they were never conceived for integration within a Cloud environment.

Furthermore these custom solutions only work for custom equipment which makes it difficult for an operator to manage the network from a single point.

To allow future extensibility our solution makes use of dynamic typed languages, most notably Groovy that allows further definition of platform configuration language on the spot without the need to redeploy the whole solution.

This modular approach allows the operator to add new equipment with ease in this ecosystem and just provide the provisioning language adaptations required on the spot.

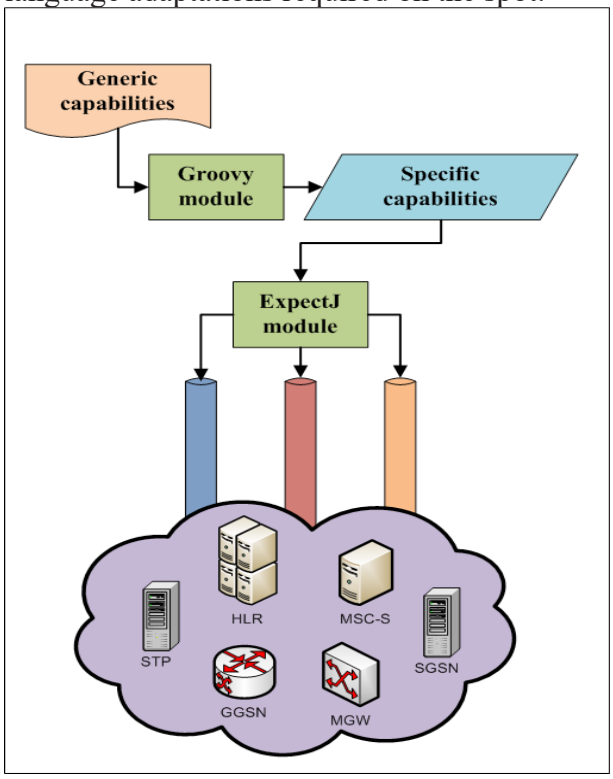

Fig. 1 Overview of the integration framework

As Figure 1 points out , the solution introduces two indirection layers between the end user and the real equipment (in our case telecom network elements form GSM and 3G architecture) with the scope of adapting the vendor independent provisioning to hardware specific semantics and then automating the whole deployment.

```
<Hlrs>
  <Hlr DisplayName="HLR0" HostName="LAB11" ID="1"
      <ProjectData>
          <NetworkSetup MNC="153" MCC="01" MSINR
      </ProjectData>
      <MobileSubscribers>
          <Msub ID="1">
              <AuthenticationData IMSI="01153000"
              <ProfileData Services="TS10,TS21"
          </Msub>
          <Msub ID="2">
              <AuthenticationData IMSI="01153000"
              <ProfileData Services="TS10,TS21"
          </Msub>
      </MobileSubscribers>
  </Hlr>
  <Hlr DisplayName="HLR1" HostName="LAB12" ID="1"
      <ProjectData>
          <NetworkSetup MNC="154" MCC="02" MSINR
      </ProjectData>
      <MobileSubscribers>
          <Msub ID="1">
              <AuthenticationData IMSI="02154000
              <ProfileData Services="TS10,TS21"
          </Msub>
          <Msub ID="2">
              <AuthenticationData IMSI="02154000"
              <ProfileData Services="TS10,TS21"
          </Msub>
      </MobileSubscribers>
  </Hlr>
</Hlrs>
```

Fig. 2 Sample generic configuration

The proposed and implemented concept borrows ideas from the distributed instrumentation ecosystem such as the key aspect of generic equipment „drivers" that can be modified on the spot by the operator. One can observe the resemblance with traditional device drivers from the PC world where a set of common functionalities is abstracted from the actual implementation. This kind of generic middleware offer the user only the basic features that are present on every implementation.

By extension the same pattern can be applied to the Telco ecosystem where a set of basic provisioning services can be offered to the end user.

Figure 2 showcases such a scenario in which the generic capabilities are described through a set of language-independent XML files. These are processed by a domain specific logic and translated into actual configuration commands that are sent to the real network equipment.

The Groovy module implements the domain specific logic responsible for interfaces adaptation. As you would expect the translation could have also been done in a static typed language but the choice of Groovy is not random since it offers a future proof solution.

The idea behind it is that it runs on standard JVM[5] and it binds very well with other parts of the solution that are written in Java.

```
#HLR0
     CRACMSUB:IMSI=011530000000101,A4K
     CRMSUB:MSIN=0000000101,BSNBC=TELE
#HLR1
     ZMAC:IMSI=021540000002101:KI=CF0S
     ZMAI:IMSI=021540000002101,MSISDN=
```

Fig. 3 Specific capabilities syntax

Interaction with the equipment is based on shell services (either Telnet or SSH) because it is the most elementary way in which automation can be achieved.

The purpose is to automate the configuration sessions through means of stream indirection in order to avoid direct dependencies on vendor libraries. In this way we also ensure the all the features of the equipment interface are used.

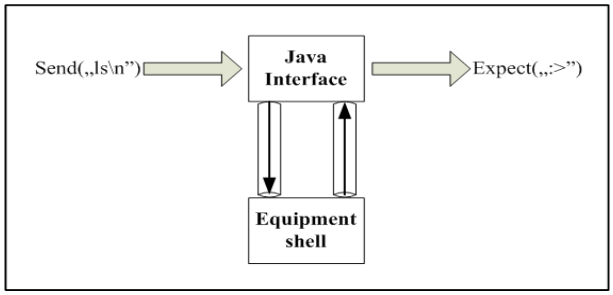

Fig. 4 Interaction with real equipment

Furthermore it is possible to orchestrate complex configuration scenarios with the help of the ExpectJ utility library that implements the Java binding of the well known automation utility Expect [6]. Specific syntax is retained under the form of a collection of Java objects that are passed to Expect utility for shell automated interaction.

3. TESTING AND VALIDATION

In order to validate the solution a test topology was configured with the help of the framework. Initially in this scenario a Tektronix K1297 (telecom network protocol emulator) was used for emulating two network elements HLR (Home Location Register) and a BSC (Base Station Controller) by running two predefined emulation scenarios.

Unfortunately one of the first challenges was to make the host OS (Win NT) to offer some kind of shell process handling. This was achieved by installing PsTools for Windows and by deploying a SSH server to allow secure remote connections.

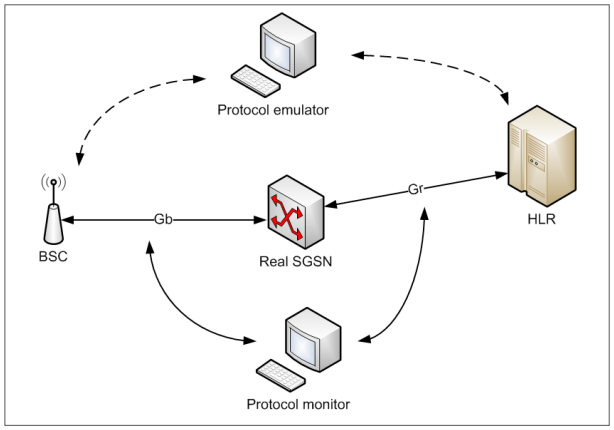

Fig. 5 Test environment

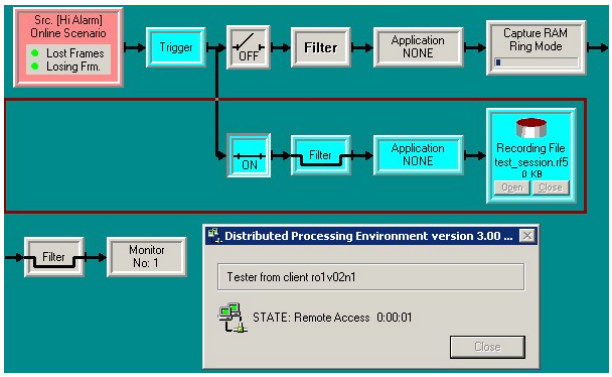

Fig. 6 Remote control of Tektronix K1297

A second Tektronix K1297 (a newer generation) was used for monitoring the messages on the Gb and Gr interfaces thus isolating the device under test which is the real SGSN (Serving GPRS Support Node). In this way the end user is only aware of the high level network topology (figure 5) and doesn't care about actual configuration details (figure 6).

CONCLUSION

One of the biggest challenges of network operators refers to managing a complex topology from a single point of control. While this usually is aimed by classical OSS their approach is limited to a set of equipment of the same generation.

With our solution we succeed in establishing an end-to-end provisioning solution, vendor independent and viable for cloud integration thus opening new service opportunities for operators. Future works could include developments towards network function virtualization (NFV) concept as well software defined networks (SDN) to tighten even more the gap between the telecommunications and Cloud ecosystems.

In the end offering Telco domain specific equipment as a cloud IaaS model aims to decrease time to market, add new revenue streams and increase ROI for network service providers.

BIBLIOGRAPHY

1. EU Comission, *Digital agenda for Europe* [online]. Available: http://digital-agenda-data.eu (2014)
2. Network Test Automation Forum, *NTAF White Paper*. [online] Available: http://www.ntaforum.org (Nov 2012)
3. Nokia Siemens Networks (NSN), *NetAct* Whitepaper, (2011)
4. Ericsson , NetOp EMS Whitepaper, (2012)
5. Konig D., *Groovy in Action*, Manning Publications (2007)
6. Libes D., *Exploring Expect: A Tcl Based Toolkit for Automating Interactive Programs*, O'Reilly (1995)