D4.1

Hybrid Network Manager architecture and functionality

<table>
<thead>
<tr>
<th>Grant Agreement nº:</th>
<th>645047</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Acronym:</td>
<td>SANSA</td>
</tr>
<tr>
<td>Project Title:</td>
<td>Shared Access Terrestrial-Satellite Backhaul Network enabled by Smart Antennas</td>
</tr>
<tr>
<td>Contractual delivery date:</td>
<td>31/10/2016</td>
</tr>
<tr>
<td>Actual delivery date:</td>
<td>07/11/2016</td>
</tr>
<tr>
<td>Contributing WP</td>
<td>4</td>
</tr>
<tr>
<td>Dissemination level:</td>
<td>Public</td>
</tr>
<tr>
<td>Editors:</td>
<td>TAS-E</td>
</tr>
<tr>
<td>Contributors:</td>
<td>TAS-E/OTE/AVA</td>
</tr>
</tbody>
</table>

Abstract:
This Deliverable contains the outcome of Task 4.1 “Interoperability of terrestrial and satellite links” and 4.2 “The Hybrid Network Manager (HNM)”. It provides the high level specification of the network element in charge of the system interoperability between satellite and terrestrial network segments, considered main components of SANSA environment. The Hybrid Network Manager will be in charge of the configuration control over the terrestrial and satellite backhaul elements, covering functionalities as: radio resources optimization, topology management, routing management, quality of services policies control and energy saving, keeping the global system configuration in a modular architecture approach.
Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Editor</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>01/05/2015</td>
<td>TAS-E</td>
<td>Document creation with proposed ToC</td>
</tr>
<tr>
<td>0.2</td>
<td>19/07/2016</td>
<td>TAS-E</td>
<td>Solution Description, Architecture definition for HNM and IBN (intelligent backhauling node), components. Functional, logical and physical initial definition.</td>
</tr>
<tr>
<td>0.3</td>
<td>29/07/2016</td>
<td>TAS-E</td>
<td>HNM and IBN architecture update, presenting commissioning, configuration and event management functionalities. Additional minor changes on formats.</td>
</tr>
<tr>
<td>0.4</td>
<td>19/08/2016</td>
<td>TAS-E</td>
<td>Commissioning Management Module detail definition, data flows and operation description.</td>
</tr>
<tr>
<td>0.5</td>
<td>22/08/2016</td>
<td>TAS-E</td>
<td>Configuration and Event Management Module definition, data flows and operation description.</td>
</tr>
<tr>
<td>0.6</td>
<td>24/08/2016</td>
<td>TAS-E</td>
<td>HNM Core database model for commissioning and configuration operations.</td>
</tr>
<tr>
<td>0.7</td>
<td>26/08/2016</td>
<td>TAS-E</td>
<td>HNM Core database mode for event handling operations. HNM User Interface Definition</td>
</tr>
<tr>
<td>0.8</td>
<td>12/10/2016</td>
<td>AVA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>0.8</td>
<td>13/10/2016</td>
<td>CTTC</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>1.0</td>
<td>07/10/2016</td>
<td>TASE</td>
<td>Final Version released.</td>
</tr>
</tbody>
</table>

Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>Contributions included</th>
</tr>
</thead>
<tbody>
<tr>
<td>Willie Jose Betancourt Simoza</td>
<td>TASE</td>
<td>All document</td>
</tr>
<tr>
<td>Georgios Ziaragkas</td>
<td>AVA</td>
<td>Chapter 3, 4, Quality Assurance</td>
</tr>
<tr>
<td>Jorge Baranda</td>
<td>CTTC</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>Jose Nuñez</td>
<td>CTTC</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>Diana Mathew</td>
<td>AVA</td>
<td>Quality assurance</td>
</tr>
</tbody>
</table>
Table of Contents

List of Figures ................................................................................................................................. 5
List of Acronyms ............................................................................................................................... 7
Executive Summary ............................................................................................................................ 9
1 Introduction ..................................................................................................................................... 10
2 SANSA Solution Description ........................................................................................................ 11
  2.1 Scope ...................................................................................................................................... 11
  2.2 SANSA Transport Architecture ............................................................................................... 11
3 HNM Component Description and Architecture ......................................................................... 13
  3.1 HNM Software Architecture Design ...................................................................................... 15
    3.1.1 Guidelines ....................................................................................................................... 15
  3.2 Commissioning Management Module ..................................................................................... 22
    3.2.1 Initial Routine Subsystem ............................................................................................... 22
    3.2.2 Authentication Management Subsystem ........................................................................... 23
  3.3 Configuration Management Module ......................................................................................... 26
    3.3.1 Radio Resources Management Subsystem ....................................................................... 29
    3.3.2 Topology Management Subsystem .................................................................................. 30
    3.3.3 Routing Management Subsystem .................................................................................... 31
    3.3.4 Quality of Service Management Subsystem .................................................................... 32
    3.3.5 Energy Saving Management Subsystem ........................................................................... 35
  3.4 Events Manager Module ........................................................................................................... 37
    3.4.1 Radio Resources Event Handling Subsystem .................................................................... 37
    3.4.2 Topology Event Handling Subsystem ............................................................................... 38
    3.4.3 Routing Event Handling Subsystem .................................................................................. 39
    3.4.4 Quality of Services Event Handling Subsystem ............................................................... 40
    3.4.5 Energy Saving Event Handling Subsystem ........................................................................ 41
  3.5 HNM Core Module ................................................................................................................... 42
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.1</td>
<td>HNM Core Database</td>
<td>43</td>
</tr>
<tr>
<td>3.6</td>
<td>HNM Graphic User Interface</td>
<td>48</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Interface Capabilities</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>IBN Component Description and Architecture</td>
<td>51</td>
</tr>
<tr>
<td>4.1</td>
<td>iBN Software Architecture Design</td>
<td>51</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Guidelines</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Commissioning Management Module</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Configuration Management Module</td>
<td>54</td>
</tr>
<tr>
<td>4.4</td>
<td>Event Manager Module</td>
<td>54</td>
</tr>
<tr>
<td>4.5</td>
<td>iBN Core Module</td>
<td>55</td>
</tr>
<tr>
<td>4.5.1</td>
<td>iBN Core Database</td>
<td>55</td>
</tr>
<tr>
<td>4.6</td>
<td>iBN User Interface</td>
<td>58</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions</td>
<td>60</td>
</tr>
<tr>
<td>References</td>
<td>61</td>
<td></td>
</tr>
</tbody>
</table>
List of Figures

Figure 1 Work Package 4 ................................................................................................................... 10
Figure 2 Sansa System Environment ................................................................................................. 12
Figure 3 Inter & Intra Node Control Traffic ....................................................................................... 13
Figure 4 Inter & Intra Node Data Traffic ........................................................................................... 14
Figure 5 HNM Modular Architecture ............................................................................................. 17
Figure 6 HNM Subsystems High Level View .................................................................................... 18
Figure 7 HNM Component Architecture ......................................................................................... 21
Figure 8 Interfaces Definition ............................................................................................................ 21
Figure 9 AAA Authentication ............................................................................................................. 24
Figure 10 HNM - IBN AAA Interactions ............................................................................................ 25
Figure 11 HNM Configuration Module ............................................................................................ 26
Figure 12 HNM Configuration Model Process Flow ........................................................................ 27
Figure 13 Radio Resources Process Flow .......................................................................................... 29
Figure 14 Topology Management Process Flow .............................................................................. 30
Figure 15 Routing Management Process Flow ................................................................................ 31
Figure 16 HNM - IBN Traffic Interactions ....................................................................................... 32
Figure 17 QoS Evaluation Process Flow .......................................................................................... 34
Figure 18 Energy Saving Process Flow ............................................................................................ 35
Figure 19 HNM - iBN Energy Saving Data Interactions .................................................................... 36
Figure 20 Radio Resources Event Handing Module ......................................................................... 38
Figure 21 Topology Management Topology Event Handling Process Flow ..................................... 39
D4.1: Hybrid Network Manager architecture and functionality

Date: 07/11/2016
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>Backhaul Node</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CAPEX</td>
<td>Capital Expenditure</td>
</tr>
<tr>
<td>CDN</td>
<td>Content Delivery Network</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Code Point</td>
</tr>
<tr>
<td>DVB-S</td>
<td>Digital Video Broadcasting – Satellite</td>
</tr>
<tr>
<td>EC</td>
<td>European Commission</td>
</tr>
<tr>
<td>eNB</td>
<td>eNodeB</td>
</tr>
<tr>
<td>EPC</td>
<td>Evolved Packet Core</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>HeNB</td>
<td>Home eNodeB</td>
</tr>
<tr>
<td>HNM</td>
<td>Hybrid Network Manager</td>
</tr>
<tr>
<td>iBN</td>
<td>Intelligent Backhaul Node</td>
</tr>
<tr>
<td>IPSec</td>
<td>IP Security</td>
</tr>
<tr>
<td>ISCSI</td>
<td>Internet Small Computer Systems Interface</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MBBS</td>
<td>Macrocell Backhaul Base Station</td>
</tr>
<tr>
<td>MBS</td>
<td>Mobile Base Station</td>
</tr>
<tr>
<td>MCN</td>
<td>Mobile Core Network</td>
</tr>
<tr>
<td>NMS</td>
<td>Network Management System</td>
</tr>
<tr>
<td>O&amp;M</td>
<td>Operation &amp; Maintenance</td>
</tr>
<tr>
<td>OPEX</td>
<td>Operational Expenditure</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Research &amp; Development</td>
</tr>
<tr>
<td>RAN</td>
<td>Radio Access Network</td>
</tr>
<tr>
<td>REM</td>
<td>Radio Environment Map</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RTN</td>
<td>Return (channel)</td>
</tr>
<tr>
<td>SA</td>
<td>Smart Antenna</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>SoTA</td>
<td>State of the Art</td>
</tr>
<tr>
<td>TNA</td>
<td>Transport Network Architecture</td>
</tr>
<tr>
<td>TOS</td>
<td>Type Of Service</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>VSAT</td>
<td>Very Small Aperture Terminal</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
</tbody>
</table>
Executive Summary

This is Deliverable D4.1 of the Shared Access Terrestrial-Satellite Backhaul Network enabled by Smart Antennas (SANSA) project (grant agreement no 645047) which documents the outcome of Tasks 4.1 and 4.2. The objective of this deliverable is to define the functional architecture, internal modules and external interfaces using possible assumptions and constraints in order to define the overall system architecture for the Hybrid Network Manager (HNM) entity. One of the main challenges will be to define the main parameters to be considered between the architecture elements to keep a consistent data model. The material documented in D4.1 will allow the SANSA consortium to collaborate in a coherent way regarding the practical requirements for the devices finally adopted in the proof of concept and present a specific framework/uniform.

This document presents a high level description of the architectural components of the HNM from a functional and logical perspective, detailing functional modules such as configuration management and event handling as well as the interactions with the main components in the solution. The Intelligent Backhauling Node (iBN) is presented in this document to complete the description of the main HNM architectural components, giving a near end-to-end view of the system architecture.
1 Introduction

The objective of this document is to define the architecture of The Hybrid Network Manager (HNM) and its main components, presenting an overall description view about the expected functional and logical capabilities in order to cover the defined requirements for the SANSA system with respect to, spectrum and energy optimizations, Opex reduction and resilience among others. Finally, some specific technical recommendations about the integration of node devices as satellite and terrestrial modems, Home eNodeB (HeNB) will be presented.

This deliverable contains is output of the work done under Tasks 4.1 and 4.2 of WP4 which aims to define the scenarios and network architectures that will be used in SANSA as well as the Key Performance Indicators (KPIs) for evaluating the proposed SANSA solution. Figure 1 illustrates how Tasks 4.1 and 4.2 fit within the WP4 work plan, taking into consideration the state of the art detailed in D2.2 [1] and D2.3 [2].

Figure 1 Work Package 4

The remainder of this deliverable is divided into 3 chapters and is organized as follows:

- Chapter 2 presents a brief description about the SANSA architecture, presenting information about the solution and its main components.
- Chapter 3 defines the HNM architecture including functional and logical modules and subsystems.
Chapter 4 defines the iBN architecture including functional and logical modules and subsystems.

The document ends with Chapter 5, which summarizes the main outcomes of this document in the conclusions.

2 SANSA Solution Description

2.1 Scope

The system architecture in the SANSA solution pretends to provide an end-to-end network solution including access, transport, and core network. This architecture aims to cover the use cases and scenarios defined in deliverable D2.3 [2]. It also details the transport network architecture (TNA), which is the main focus of the SANSA project. Sections 3 and 4 define in detail the architecture of the two key enabling SANSA components - the intelligent Backhaul Node (iBN) and The Hybrid Network Manager (HNM), respectively.

2.2 SANSA Transport Architecture

As mentioned before, the SANSA system focuses on evolving the Transport Network Architecture (TNA) -describing the network elements between the User Equipment (UE) and the Evolved Packet Core (EPC) and combining satellite and terrestrial network resources. The SANSA transport network architecture is composed by the following key elements:

1. The iBN extends the internal architecture of traditional BNs by introducing new functional blocks and interfaces for the proper management of backhaul satellite and terrestrial resources. Amongst other functions, the iBN will control routing, traffic classification, and energy saving functions. The iBN will operate on short to medium timescales and is reconfigurable by the HNM. It will encompass interfaces to other iBNs, and it will be able to reach the EPC either directly (with a radio link) or through other iBNs. Finally, some iBNs in the network may include a direct connection to the EPC through the satellite network.

2. The Backhaul Node (BN) is a legacy entity embedding the H(eNB) in charge of carrying transport traffic to the EPC. It does not possess routing, traffic classification, and energy management functions. A special case of BN a Mobile Base Station (MBS). The MBS is a BN that includes mobility capabilities (e.g. a BS in a train). It is an optional element in the SANSA scenarios.
3. The Satellite (SAT) position is enhanced by SANSA due to its smooth integration in the reconfigurable terrestrial transport network. The SAT will encompass an interface to the EPC, and an interface to iBNs. The interface between the iBN and the SAT allows the system to assess the satellite link status and use the data for traffic classification, routing, topology reconfiguration, and interference management between the satellite and terrestrial links. Information such as satellite carrier frequency, channel bandwidth, available data rate and link availability is constantly monitored by the HNM.

4. The HNM is a new entity introduced by SANSA which includes functionalities to manage not only satellite but also terrestrial backhaul resources. Based on global network information view based on its monitoring capabilities, the HNM is in charge of determining the topology formed between the iBN nodes and their connection and configuration with the satellite resources. In this context, it can configure backhaul resources embedded in terrestrial iBNs, MBSs, and satellite resources. It operates on long and medium timescales.

![Figure 2 Sansa System Environment](image-url)
3 HNM Component Description and Architecture

The Hybrid Network Manager (HNM) and the intelligent backhaul node (iBN), are considered as the main components in SANSA network. The HNM component includes, the calculus algorithms and processing power to determine new possible topologies, routing determination and event or alerts evaluations to take actions under the network devices.

The HNM takes the role of control manager, and based on all event information collected from the iBN nodes and the configuration received from SANSA operator, is able to optimize the traffic and power consumptions at different levels in the network. Actions can include applying a specific energy saving policy in relation to traffic consumption and enhancing the performance in the network using Quality of Services policies, between others. The HNM shall present interfaces to integrate different modules such as externals Radio Resource Managers or Interference analyzers, increasing the capabilities of the system to provide real time analysis and direct actions. This will help to avoid, for example, interference or future link failures, enabling the system to provide real time management in an autonomous way. Figure 3 shows the data control flow between HNM and iBN’s nodes.

![Figure 3 Inter & Intra Node Control Traffic](image)
One of the main challenges for the HNM and iBN components is identified at integration level. In the actual market most of the manufacturers do not provide a complete capacity of integration with external components. iBN should have the capabilities to be easily integrated with different commercial modems or external components such as interference analyzers and radio resources modules in a feasible way. Figure 4 shows a brief view of the data traffic flows between the different elements involved in SANSA environment.

Figure 4 Inter & Intra Node Data Traffic
3.1 HNM Software Architecture Design

HNM shall be considered as an evolution element from a traditional terrestrial backhaul network to the SANSA initiative. The main idea behind the HNM is a redefinition about how the networks are managed, pursuing an initiative to provide autonomous handling against typical network events like congestion, link failures or implementing Quality of services policies and energy saving initiatives on the fly in all the network components using a unique interface but keeping local control inside the HNM management entity.

3.1.1 Guidelines

Software design process is an engineered construction [3], much like an automobile or a building. As such, a software program consists of many parts or components. Like an automobile or a building, software can be said to have an architecture: one or more organizing principles that control how each component relates to and communicates with each other. Architecture structures an unorganized space to serve one or more functions, just as an engine provides power, walls provide privacy and shelter, or doorways orchestrate the flow of motion. The structure provided by architecture for a product line is inherently hierarchical. Power-trains comprise of engines, transmissions, and exhaust systems; buildings consist of floors and floors are divided into rooms.

Similarly, a software application consists of layers of hierarchy, with each layer containing one or more components. The boundaries, or interfaces, between layers and components, like the walls and doorways in a building, control the motion of information through the application. Of course, there are almost an infinite number and variety of software products, but at the highest level of definition, there are software development tools, software applications developed with these tools, and the systems software, often called operating systems, upon which both tools and applications run. Each one of these software products – operating systems, software development tools, and software applications – has its own layers of technology. Interfaces connect the modules within each layer, and one layer to the layers above and below. A software product – be it an operating system, tool, or application – in which these layers and component boundaries are well-defined is said to be modular.

Using these definitions, we can outline the principles of good software design. A well-designed software solution must:

- Possess clear and explicit organizing principles, expressed as the software product’s architecture. Since architecture is typically developed not just for one product, but also for a stream of related products, the architecture is best referred to as a product line.
architecture, to support customization, this architecture must describe the mechanisms by which the software, and even the architecture itself, are allowed to evolve.

- Be hierarchically organized into layers. While it is perhaps easiest to think of these layers as layers of specific technology, we find it best to view them in terms of function (such as a real-time data acquisition layer, a database layer, logic or algorithm layer, or the graphical user interface). The reason is that multiple technologies may exist within or they can be used to implement the functionality of a particular layer. For example, FibreChannel, InfiniBand, and Internet Small Computer Systems Interface (iSCSI) are all communications technologies used for network connection modules in modern storage array systems.

- Have individual layers consisting of a modular set of components - each with its own single function or purpose, and each offering a single robust interface for other modules to access that functionality. The FibreChannel, InfiniBand, and iSCSI communication protocols mentioned above are implemented as separate components within the communications subsystem of the storage device. Consider a basic report-writing layer, which consists of modules for querying information from a database, for accessing a report template, for pumping the information through that template, and then formatting it for the user’s particular printing device. Interestingly, the output of one module in this layer of functionality becomes the input for another module, and so on. This connection works best if there is only one clear way to pass data into any given module.

Good architecture in software provides modularity across and among all the various components within the software. This means that the components can be combined readily in different ways to meet different purposes without any substantial additional code to make these components work together. With this foundation of flexible components, developers can then add additional modules that provide the functionality needed for a specific application [4]. Good architecture makes software reuse possible. Further, if the foundation of core subsystems is particularly well designed, software developers outside the company – be it other software companies or programmers in end-user companies – can also quickly write their own application modules [5] that plug immediately into that foundation and use its functionality. This is how the most successful software companies have developed extensive collections of specific software applications made by third parties.

Network externalities can play an important role in how users perceive a software product or system. The most common form of network externalities – on the positive side – is when the value of a product for an individual user rises in proportion to the number of other people who use it. The telephone is a classic example. One telephone has little utility, but a worldwide network of telephone users makes each individual telephone a very powerful tool. On the negative side, of course are viruses such as Trojan horses that invade a user’s computer to infect others, and from these others, yet others.
On the basis of this analysis and principles, the Hybrid Network Manager should have a basic modular architecture, where each module shall represent specific functionalities across the main operations required. A good approach could be presenting four different modules covering the commissioning, configuration, event management and Graphic user interface options. This structure could be orchestrated by a Core Module acting as the source of computational power.

The main idea behind this modular model will be to present a coherent structure with a functional separation keeping in mind the main processes expected by the system.

These functionalities will be interacting across the elements until they reach the iBN nodes in tasks like network information gathering, configuration updates or receiving additional information from external sources like interference analyzers. The graphic user interface will allow the SANSA operators to execute manual updates and to evaluate the component performance and events in real time.

Figure 5 illustrates the modular architecture of the HNM showing the interactions between the general logic modules and the data management module.

![Figure 5 HNM Modular Architecture](image)

Detailed definitions about the different modules are presented in the HNM Component Description and Architecture in the following sections.

Once a module definition model is adopted, the next step consist in define the subsystem necessaries to complete the system architecture.

A focused subsystem helps to manage complexity and is more useful over the long term. Without a focused subsystem, changes in one part of the system tend to ripple through and impede functionality in other parts of the system. The goal of modular architecture is that changes at one level of the application do not affect the integrity of modules at other levels.

A classic and still present approach is based on application software containing both logic – such as for business accounting or statistical processing – and data. On one hand, a developer might
correctly choose to have a series of modules each containing a different set of accounting or statistical logic, which then communicate to another module focused on organizing, storing, and accessing the needed data for the calculations. This reflects good architecture. Less desirable is a design where different buckets of logic are combined into one module, and even worse, where data management functions are co-mingled with the logic in that module. This can be viewed as a smokestack architecture where each application is monolithic and shares little, if any, code with other applications in the product line. This type of architecture, lacking clearly focused subsystems, makes the implementation of new technologies (such as a new data management method) labour intensive and error prone. It requires changes to many modules, and extra effort for the developer to ensure that these changes do not impede the functionality of other code containing algorithms and logic. This can produce tremendous problems downstream for integration and maintenance operations in applications.

A high level view of the SANSA subsystems (represented by blue boxes inside each module) is presented in Figure 6. A detailed definition will be presented in the next sub-sections.

![Figure 6 HNM Subsystems High Level View](image)

As the main element in the SANSA Logical architecture vision, the Hybrid Network Manager is, responsible to allocate all the logical modules in charge of the system operations, processing real time events, evaluating alerts statistics, collecting the configuration from all node elements
located in each IBN and controlling the different layers to warranty flexibility, scalability and redundancy at HNM level but keeping the simplicity and a distributed scheme.

In order to accomplish these requirements, it is necessary to evaluate what scheme shall warranty the expected performance. A reasonable comparison between different tiers levels and constraints is presented below to determine what will be the better architecture approach to design the different modules of the HNM entity.

Layers considered in a 3 tier architecture are the Application layer, the Broker layer, and the Database layer. Actual systems usually contain an additional tier which is typically represented by the client browser and is widely known as the Web Client. These 4 basic layers (browser, server, broker, database) are often used in various combinations when describing 2 and 3 tier architectures leading to confusion about what constitutes a tier. Application layer (Tier 1) is typically composed by a group of application components, all of which exchange information with a database. The middle layer (Tier 2) is considered a stand-alone server which creates and manages a pool of database connections (the connection broker). The database layer (Tier 3) could be formed by one or more database instances. Once the system startup, the stand-alone broker creates the requested number of database connections and begins the housekeeping tasks which it continues to run as long as the system is running. Requests for connections are generated from the application threads as they are "hit" from the browser. Each thread requests a connection and then hands the connection back to the broker for replacement in the connection pool.

In the 2-Tier architecture, the single monolithic broker is replaced by a distributed set of internal broker threads. Each application contains one or more broker threads, each of which is configurable for number of connections, database instances, housekeeping characteristics, etc. Upon initialization, each application starts up its set of broker threads and keeps them running as long as the application is running. Since the brokers are running in their own threads, they continue to run in the background handling connections and performing the housekeeping tasks while the core of the servlet is called in successive threads. If a particular application fails, only its own connections are affected allowing the rest of the applications to continue running normally.

There are several important advantages in the 2-Tier approach:

• Simplicity

The 2-Tier approach avoids the complexity involved in designing and implementing a stand-alone broker. Implementing a broker thread is as simple as creating a new instance of a class. In a 3-Tier architecture, the complications of context switching gives rise to complex memory management implementations and/or proprietary communication protocols. Simplicity also shows itself reflected in cost. Most 2 Tier brokers are freely available with various degrees of housekeeping and can be written for specialized situations with relatively few lines of code.
• **Distributed Broker**

Another benefit of the 2-Tier approach is its ability to easily distribute the brokering task between as well as within application components. The broker in a 3 Tier structure is typically a single process or server and relies on a centralized location for broker management. The distributed broker design allows for great flexibility in dealing with multiple, heterogeneous database instances; the decision making process for connection allocations is simple and natural. In a monolithic broker, the same functionality is either absent or obtained through complex algorithms and non-intuitive parameter settings.

• **Fault tolerance**

A stand-alone broker subjects a system to a single point failure. A single failure in a 3-Tier broker process can lock all application components that rely on it. In contrast, a distributed broker design is fault tolerant (a failure in a single application component affects only its own database connection pool).

• **Flexible Load Balancing**

The 2-Tier architecture allows a large degree of flexibility in load balancing - largely due to its distributed nature. Application components that receive a large proportion of overall “hits” can be delegated a larger number of connection resources. Connections can be allocated on the basis of components requirement. Components that are deemed application critical can be given an appropriate proportion of available database resources. In a 3-Tier system, the equivalent granularity of connection allocation is typically absent or implemented through complex algorithms. Load balancing could be typically limited to CPU usage or network traffic alone.

• **Customized Housekeeping**

Housekeeping tasks can be tuned for individual calls or on the basis of database brand or location. It might be desirable, for example, to log connection statistics at different rates for databases in different locations. Because broker threads can be scaled and distributed freely, the designer of an application has enormous flexibility in shaping the system to handle new and complex situations.

SANSA project aims to find the advantages present in a 2-Tier architecture, decreasing complexity present in 3-Tier model.

A high level view of the 2-Tier architecture recommended for SANSA is presented in Figure 7. Figure 8 show the reference to the HNM component interfaces definition.
**Figure 7 HNM Component Architecture**

**Figure 8 Interfaces Definition**
3.2 Commissioning Management Module

The commissioning process is one of the most typical activities developed for any element in the SANSA transport network (e.g. Very Small Aperture Terminal (VSAT)) to complete the provisioning process in their own network. Historically, this process has covered all the steps necessary to establish a trusted connection and enroll the elements with the central location element (HUB). Once this process has been completed the new element is considered a new member of the network and is enable to provide the services already defined.

One of the main challenges envisaged in the SANSA project is finding a feasible way to decrease the cost associated in deploying new nodes or elements and offering a solution that gives flexibility and minimized scalability issues. For this reason, two subsystems have been presented next - the initial routine looks to describe the provisioning process that involve the new node allocation process. The authentication management subsystems give a detailed description on how to accomplish the different authentication processes to complete a secure provisioning process across the SANSA and operator network elements.

3.2.1 Initial Routine Subsystem

The state of the art presents the Auto-discovery method as one of the best options to enable the management systems to provision new network elements. Traditionally, the management systems have been focused on the discovery of hosts, servers and network elements in isolation. However with the emergence of new internet services and the necessities to deploy these elements at a rapid pace, it has been essential to find new techniques of implementation that will help to increase the capabilities of the network management systems.

Hewlett Packard was the first to propose the role of Auto discovery capabilities around the year 2000. It was based on the use of SNMP queries directed at the network routers. Different mechanisms to establish discovery process are available nowadays. Most of these mechanisms run active test using a common transport port (UDP or TCP) while other methods can include specific application agent located inside the host directly. However, most of these methods consider isolation networks.

Usually the operator systems are composed of two elements - point of presence (POPs) and Server Farms (typically located on physical datacenters or cloud). The POPs are decentralized locations that enable the end user connection with the operator backbone and server farms, most
commonly known as datacenter or cloud (virtual datacenters), and are considered the heart of the main servers where applications—public or private—are exposed.

In order to define a discovery mechanism different characteristics should be considered. Usually the systems employed by the operators are heterogeneous, with multitude of operating systems (Solaris, Linux, Microsoft, CentOS etc.), different hardware platforms (SUN, HP, Ericsson, Huawei, Nokia, Alcatel-Lucent) and software applications. The solution that will be proposed should enable to function in all these environments without introducing changes in the existing applications or current network equipment deployed.

Discovery mechanisms could be influenced by different dependencies such as application servers, service groups, inter-service dependencies and organizational dependencies related to the organizational levels.

In the case of SANSA, one of the discovery mechanisms will be based on the use of the operator domain name servers (DNS) to resolve a specific path previously loaded in the new iBN node. With this minimum information it is possible to establish a broadcast in the network and determine the initial information exchange between the new element and the network manager.

The use of bootstrap discovery by querying the routing tables and using encrypted connections against the authentication servers is possible to warranty security levels to avoid any kind of breaches during the discovery process.

3.2.2 Authentication Management Subsystem

Commissioning management module should cover all the AAA - authentication, authorization and accounting - mechanisms, capabilities to be used between the network components and operator authentication systems. Authentication, authorization and accounting is a framework for controlling access to network resources, defining a group of policies, auditing options and providing information about the use of services which aids to warrant the network management functions and security.

Typical authentication methods look to provide a way to identify a user. Valid username and a password are considered key to gain access. The AAA server compares the credentials presented within the database looking for a match and if the match is found, the user is granted access to the network, otherwise network access is denied.

After authentication process is completed, during the second phase, authorization determines the kind of actions/tasks defined for the user in relation to the user’s authority or role. Usually a
previous definition on the system defines the activities, resources and services available of specific users or groups.

Furthermore, accounting represents a method to measure the devices the users use to access the system. This capability is especially important to security or audit activities during the sessions. Statistics information about logging operations combined with the authorization process provide important inputs for resources and capacity planning.

Latest tendencies in authentication process are based on multifactor methods. One way, known as two factor authentication method, requires the use of hardware or software-based security tokens that people carry around. This method uses a display of series of numbers that are only valid for a short time and can be generated in a specific device or an application running on a mobile phone. This method is typical for end user devices but is not applicable to server processes.

**Figure 9 AAA Authentication**

Within the ambit of SANSA project, the commissioning operations should be focused on one hand, on establishing the mechanism to authenticate the iBN and HNM Core servers and in the other hand, to authenticate iBN against local node elements (satellite and terrestrial modems and eNB).

The new elements in the operator network, resulting from SANSA, are the HNM and IBNs. These elements should be integrated initially with the operator AAA servers. HNM should be the first new element to establish the connections with the authentication system. Once this process has been completed successfully, the iBN can proceed to complete the AAA process using the HNM and the operator authentication systems. The use of multifactor methods to warrant the AAA
process inside the operator network will be key in offering an extra layer of security and isolated ways to complete the authentication.

Figure 10 HNM - IBN AAA Interactions
3.3 Configuration Management Module

The main functionality of the Configuration Management Module will be to manage the end to end process responsible to generate the configuration updates to be loaded at the iBN’s local elements.

At least two clearly defined inputs can be identified in the HNM configuration module. One of them will be generated by the SANSA operator using the graphic user interface to execute configuration changes over the SANSA environment. The second one will come from iBN events or alerts received and processed by the HNM event manager module. These events will act like a trigger to request configuration updates in order to warranty the performance and efficiency goals of SANSA project.

A high level view about the main process covered by the configuration module is presented in Figure 11 which illustrates the step by step process involved within the module.

Data ingestion process is presented as the first process to be covered by this module, is responsible for handling the data received from the HNM Graphic user interface managed by the SANSA operator and the data allocation process. This process represents the entry interface for all the data received from the HNM graphic user interface module. Data allocation process mainly focuses on providing the interface and the necessary logic definition to apply a correct data
storage scheme. Additional details about the operation and database recommendations will be presented in section 3.5.1(HNM Core Database) and 4.5.1(iBN Core database). A data validation process has been included as part of this module to verify the new configuration promoted. The Evaluation Process is considered the brain of the configuration module and will allocate all the subsystems needed for determining the configuration changes necessary to the iBN nodes in relation to the functionalities implemented in the SANSA environment.

Finally, the propagation process will be responsible to disseminate the configuration updates around the iBN nodes and elements.

The SANSA project defines a subset of functionalities that need to be achieved, mainly comprising radio resource allocation and optimization, topology management to find the best network configuration possible in relation to the active nodes, layer 3 operations based on distributed routing, control of quality of services policies and energy saving management options. Details about the operations will be presented in the next sections.

Figure 12 shows how the changes by the module will be processed in the configuration process.

This scheme suggests using a candidate configuration to present the parameter updates. This proposal finds to reduce human error in any configuration update, hence, this action can significantly improve network uptime. Commit scripts help to control operational practices and enforce operational policy, thereby decreasing the possibility of human error. Restricting device configurations in accordance with custom design rules can vastly improve network reliability and get a better connection control. Additionally this kind of process offers the advantage of making the SANSA operator an important component in the decision making process in order to approve the changes.
A good example of the advantages of using a candidate configuration scheme occurs when this new configuration does not adhere to the defined design rules. A commit script can instruct to generate custom warnings, system log messages, or error messages that block the commit operation from succeeding. In addition, the commit script can change the configuration in accordance with your rules and then proceed with the commit operation.

Another good example can be found when we have enabled the use of MPLS protocols for QoS policies. At commit time, a commit script inspects the configuration and issues an error if a requirement is not met. This error causes the commit operation to fail and forces the SANSA operator to update the configuration to comply. Instead of an error, the commit script can issue a warning about the configuration problem and then automatically correct it by changing the configuration to enable MPLS on all interfaces. A system log message can also be generated, indicating that corrective action was taken.

Other possibilities could be offering a subset of actions to be accomplished for each decision making point and the systems autonomously make the changes over the components. The possibilities to use a mixed scheme between human and system decision making, probably, will be the best approach to optimize the network operations.

Inside the process definition, possibilities to apply the concept of rollback is presented. This option comes from the transactional processing model common to database operations. A database transaction might make a set of changes to a given database table. The SANSA Operator then must choose whether to commit the changes (apply the changes permanently) or to roll back the changes (discard the changes and revert to the previous state of the table). In our context, rollback means that the configuration updates will be discarded, and no changes will be applied. The result of the rollback operation is to revert to the previous state, before any configuration changes are applied.
3.3.1 Radio Resources Management Subsystem

With the objective of optimizing the use of the spectrum resources, but respecting the satellite system frequency plan, a Radio Resources Management function subsystem shall be present at the HNM. The basic and most important role of this subsystem is to coordinate the use of spectrum resources between the satellite and terrestrial links. Some scenarios consider the frequency reuse between both components; therefore there could be limitations for example in the carrier assignments for transmission of satellite links that would be coordinated by the HNM.

One additional feature would be the connection with other interference tools to manage possible interferences in the hybrid SANSA Ka-band system. This sub module, running as a part of the configuration evaluation process, will cover all the frequency management operations product of a direct configuration change requested by the operator like:

- Frequency changes as a result of radio resources operations.
- Frequency changes as a result of topology updates
- Frequency changes as a result of new routing rule applied.
- Frequency changes as a result of energy saving configuration changes applied.
- Frequency changes as a result of Quality of services optimization.

Other subset of operations that originate from the input of the event manager module will be defined in section 3.4.1. A brief description about the algorithm logic behind of this sub-module is presented in the Figure 13.

![Figure 13 Radio Resources Process Flow](image-url)
3.3.2 Topology Management Subsystem

As will be described in following paragraphs, the HNM shall be able to command an iBN to reconfigure the use of its resources, resulting in a new optimized network configuration. In other words, a forced change by an external network event or an iBN reconfiguration event shall be captured by the HNM Events Management module in order to take the appropriate action.

Figure 14 describes how the Topology Management module is in charge of collecting the information related to the nodes topology to update the iBNs. The topology network update will be provided by an advance algorithm in charge of calculating the new topologies available. The input of this algorithm is a matrix describing the current topology and all the possible connections between the nodes of the network. Besides the nodes connection matrix, a set of rules and desired capabilities of network performance are taken into account. As a result, the module generates a new topology configuration that is forwarded to the iBNs after being validated by the Radio Environment Module (REM). A detailed description about the operation of these algorithms and interactions are covered in the Deliverable 4.3

![Figure 14 Topology Management Process Flow](image-url)
3.3.3 Routing Management Subsystem

The HNM shall be responsible of allocating the configuration information related to the routing protocol scheme defined for the SANSA elements. Possible approaches have been suggested in Deliverable 4.2.

Currently routing protocols widely used in commercial networks are based on size, number of hops or links status. Almost all the elements present in SANSA environment could be considered as layer 3 elements with IP capabilities. This fact makes possible to use the routing management subsystem to control and propagate the deciding routing protocol. Figure 15 describes the basic operation. This approach avoids data traffic flows within the HNM and iBN core elements, acting as transparent network elements. However, HNM should be able to allocate all the configuration parameters necessary to apply the routing protocols configuration over any of the devices present in the system. This approach will warrant the compatibility with the protocols used in the operator backbone.

Figure 15 Routing Management Process Flow
The HNM routing management sub module will focus on allocating and propagating routing protocol configuration necessary to establish the inter and intra node communication (Figure 16) between iBN elements. Configuration updates received from SANSA operator or outputs from Event Management Module should be handled by this sub system too when a network protocol configuration is implied.

Figure 16 HNM - IBN Traffic Interactions

3.3.4 Quality of Service Management Subsystem

Due to the dynamic nature of the network, it is not possible to apply QoS Management techniques to negotiate quality between end users and networks. Quality enforcement mechanisms, on the other hand, can be applicable in ad hoc networks. Quality of services tendencies are divided into two: Integrated Services (IntServ) and Differentiated Services (DiffServ). The main difference between these techniques is in the way they treat the packets. In the DiffServ model, a packet's "class" can be marked directly in the packet, which contrasts with the IntServ model where a signaling protocol is required to tell the routers which flows of packets require special QoS treatment. DiffServ achieves better QoS scalability, while IntServ provides a tighter QoS mechanism for real-time traffic. These approaches can be complimentary and are not mutually
exclusive, can be used to manage and control IP flows through queuing, marking and dropping of packets.

The IntServ architecture model (RFC 1633, June 1994) was motivated by the needs of real-time applications such as remote video, multimedia conferencing, visualization, and virtual reality. It provides a way to deliver the end-to-end Quality of Service (QoS) that real-time applications require by explicitly managing network resources to provide QoS to specific user packet streams (flows). It uses "resource reservation" and "admission control" mechanisms as key building blocks to establish and maintain QoS. IntServ uses Resource Reservation Protocol (RSVP) to explicitly signal the QoS needs of an application's traffic along the devices in the end-to-end path through the network. If every network device along the path can reserve the necessary bandwidth, the originating application can begin transmitting.

Besides end-to-end signaling, IntServ requires several functions on routers and switches along the path such as:

- Admission Control: determine whether a new flow can be granted the requested QoS without impacting existing reservations
- Classification: recognize packets that need particular levels of QoS
- Policing: take action, including possibly dropping packets, when traffic does not conform to its specified characteristics
- Queuing and Scheduling: forward packets according to those QoS requests that have been granted

DiffServ enhancements to the Internet Protocol (IP) are intended to enable scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. A variety of services may be built from a small, well-defined set of building blocks which are deployed in network nodes. The services may be either end-to-end or intra-domain; they include both those that can satisfy quantitative performance requirements (e.g., peak bandwidth) and those based on relative performance (e.g., "class" differentiation). Services can be constructed by a combination of:

- Setting bits in an IP header field at network boundaries (autonomous system boundaries, internal administrative boundaries or hosts),
- Using those bits to determine how packets are forwarded by the nodes inside the network, and
- Conditioning the marked packets at network boundaries in accordance with the requirements or rules of each service.

The HNM quality of service subsystem should be able to allocate the traffic method decided by the operator in order to classify the traffic. This classification configuration received from the SANSA operator or as a result of an event manager evaluation will be propagated to the iBN node elements.
Quality of services marking should be compatible with the QoS scheme used in the operator backbone to keep the policy consistent and warranty the performance over the data traffic.

The HNM will be able to allocate, control and propagate traffic classification policies based on:

- Port Types (UDP, TCP)
- VLAN Classification (Outer VLAN)
- IEEE 802.1p: provide QoS at Media Access Control (MAC) Layer.
- VLAN and 802.1p
- Differentiated Services Code Point (DSCP): coarse-grained mechanism employing a field in the IPv4 and IPv6 header, provides QoS at the Network layer.

A high level view about how the QoS evaluation process could be done is presented in Figure 17. It is important to consider all the iBN elements as transport layer elements with capabilities to apply an end to end marking process across the SANSA environment.

![Figure 17 QoS Evaluation Process Flow](image-url)
3.3.5 Energy Saving Management Subsystem

Most commercial manufactures have not yet submitted a specific industry standard to control the energy consumptions of the different devices. This reason makes this sub system even more complex, adding the necessity to implement the configuration changes at different levels in the local elements present at iBN level.

The main idea is to define through the HNM all necessary energy efficiency policies on terrestrial and satellite elements to optimize the power consumption in the system. This subsystem will be responsible to allocate the configuration scheduling for the different energy saving initiatives to be implemented in the network. SANSA operator could decide some specific guidelines to decrease the energy consumptions or could just establish some general KPIs to be achieved at the HNM. After that, the HNM could be able to determine the policies to be achieved in order to reach these goals.

In this case, actions will not be limited to a specific function as can be with a QoS policy update. These actions related to energy saving changes will impact different subsystems such as Radio Resources, Topology and Route management.

HNM should determine all the configuration changes that affect the sub systems and determine the most feasible way to implement. These updates, in order to warrant network availability and transparent application for the end users, are shown in Figure 18.

![Figure 18 Energy Saving Process Flow](image-url)
Other configuration requests could be generated by the event management module as a result of link status changes or bandwidth consumption cycles.

A general diagram about interactions between HNM and iBN nodes is presented in Figure 19. In this case, the configuration updates can apply at different levels in the local elements.

![Figure 19 HNM - iBN Energy Saving Data Interactions](image-url)
3.4 Events Manager Module

The Events Manager Module receives and collects the information generated at the iBN level. These events, which have been previously classified and processed by the iBN’s core server, offer a common data model. Once these events reach the event manager module, they are processed and locally allocated to start the threshold evaluation process separately in each sub modules. Once the threshold evaluation process is finished, a configuration update could be requested to the configuration module in order to realize any additional adjustment necessary in the current network elements.

One of the main goals in SANSA project is to develop a system capability which is able to generate new configurations in relation to events present in the network, optimizing the traffic performance and network availability.

Interactions between Event Management Module and Configuration Module will be fundamental to achieve a real time control of the network elements.

3.4.1 Radio Resources Event Handling Subsystem

Once the network events have been collected from the iBN’s local elements, this information should be sent to the HNM event manager module for evaluation. The radio resources subsystem should contain the logical functions necessary to determine if one of the thresholds defined previously (e.g. reception power level, signal to noise ratio) by the operator has been reached. The subsystem will send a request to the configuration module in order to update the configuration parameters affected. In case the threshold is not reached or a reconfiguration is not possible, this sub module will generate the events or alerts necessary to inform the SANSA operator about the current status of the radio resources.

Radio resources events could impact different functionalities such as topology availability, traffic flows or energy saving policies. Specifics about how the event information could be allocated are presented in section 3.5.1.
Figure 20 shows a high level definition about event handling actions involved in radio resources evaluation.

![Figure 20 Radio Resources Event Handing Module](image)

### 3.4.2 Topology Event Handling Subsystem

This subsystem should contain the logical functions necessary to determine whether or not the thresholds already defined by the operator have been reached. If the threshold limit is met, the subsystem will send a request to the configuration module in order to update the functionalities involved. In the case of the threshold not been reached or a reconfiguration not possible, these subsystems shall be responsible to generate events or alerts to inform the SANSA operator.

Topology resources thresholds used for this sub module should be based on:

- Link failures
- Link congestion
- Link degradation.
- Possible link Interference.

These events could impact different functionalities such as radio resources availability, traffic flows and energy saving policies.
Figure 21 shows a high level definition of the actions involved in the topology event evaluation.

3.4.3 Routing Event Handling Subsystem

The routing event subsystem should have the ability to evaluate the event information related to the performance of the routing protocol applied and determine the network performance, inclusion of new nodes, topology changes or any network actions that can modify the routing status and generate a re-evaluation of the protocol performance and corrective actions to warranty the performance. These logical functions could be based on thresholds defined previously by the operator and once they have been reached, a configuration update request will be sent to the configuration module. In the case of the threshold not been reached or a reconfiguration not possible, this sub system will be responsible to generate the events or alerts necessary to the SANSA operator.

Figure 22 shows a high level definition about event handling actions involved in the routing event evaluation.
3.4.4 Quality of Services Event Handling Subsystem

Quality of services event handling subsystem are based on the results obtained from a sub set of applications already defined in the network to determine how good the packet performance is, how good the compliance of the applied policies is and what possible violations in the QoS are for the thresholds defined.

Event management is a fundamental piece in QoS validation. Traffic measures about how traffic throughput is generated between elements, how this traffic is marked and the efficiency of these marks are considered key to determine the impact of the policies applied and possible optimizations tasks.

Figure 23 shows a high level definition about event handling actions involved in the Quality of services evaluation.
Quality of services events will be allocated at HNM core database. Data model definition for the structure is presented in section 3.5.1.3.4.

### 3.4.5 Energy Saving Event Handling Subsystem

Energy saving events are generated as a result of scheduled actions defined in the configuration management module of these events should be based on the capabilities of the elements to execute configuration changes. Most of the event information received should be based on the performance of the scheduled actions (Figure 24).

Other kind of events will present statistics about power consumption from different elements and components. Some of the most typical indicators could include current transmission power, CPU consumption, active links and energy consumptions by link. The availability of this information will depend on node element capabilities and will offers tools to the operator to determine the trends and new energy efficiency strategies to be implemented in the HNM.
3.5 HNM Core Module

The Core Module should contain all the logical processes in charge of communicating and coordinating the different modules running at the HNM including the access to the data allocation functions. Additionally, it will perform automated tasks based on the configuration defined in the HNM for event management functions and threshold handling. This module will make the decisions such as triggering the respective modules with computation necessary for topology determination based on the collected information of the network status. Additionally, it will be responsible to cover the following actions:

- Control the database access between the different applications modules. These modules based on configuration, commissioning and event handling parameters have been described in the previous sections.
- Network status analysis based on database information collected from iBN in real time to determine possible network bottlenecks or link failures sending traffic routing updates to the iBN’s.
### 3.5.1 HNM Core Database

HNM database will be responsible for allocating all the data support necessary for the commissioning, configuration and event management modules and sub-systems. This information should contain all the system information necessary to execute the SANSA Solution. Based on this, at least two kind of information will be allocated in the database. The first group will consist of configuration and commissioning module, including the parameters necessary to run the SANSA network solution. The first group is composed by the configuration and commissioning module, to support the authentication process and provide the startup data for the system. The second group of information will consist of the event information received from iBNs in real time and will be handled by the Events Management Module and sub modules where the logical algorithm power is located.

In relation to the kind of received information and the high level of liaison between the modules and subsystems the use of a relational database scheme present an efficient way to manage the data information and create a consistent and useful common data model. Use of a relational data model implies an established table scheme. A high level view about the necessary operations to be covered for this data model is presented in Figure 25.
3.5.1.1 Commissioning Operations

Operations related to the commissioning scenario are based on 2 different groups, the initial routine and authentication data model.

3.5.1.1.1 Initial Routine

The initial deployment for the HNM and iBN components will require minimal configuration information. HNM core database should contain a specific data model oriented to support the step by step routine followed to complete the initial bootstrap process, with the database information invoked by the commissioning module. This information should be presented through graphic user interface.

3.5.1.1.2 AAA Operations

AAA operations should be assumed by the HNM in order to warrant inter and intra node security, roles definition, credential information, logging statistics. All this information will be allocated using a data model scheme in the HNM database.

3.5.1.2 Configuration Operations

Brief description about main configuration operations and database interactions are presented in the next sub-sections.

3.5.1.2.1 Radio Resources Operations

Allocation for all the parameters involved in the frequency plan elaboration and implementation for the satellite, terrestrial modems and HeNB will be contained in this subsystem as part of the configuration operation module. All this information related to the radio resources will be summarized in the data model (Deliverable 4.2 will include a detail table definition about the data models used for the different operations defined in SANSA Solution). HNM Radio resources module will be handling this information to determine the possible updates and accomplish the propagation process around the iBN. Additionally, it will collect and manage information sources like interference analyzer to comply with the system requirements.
3.5.1.2.2 Topology Operations

Current topology information provided by the SANSA Operator will be allocated directly in the database. New configurations as a result of network events or changes generated by the calculation of the local algorithm should be present in the form of candidate configuration. Data model used between these configuration options should be the same. Once the configuration has been decided, it will be distributed to all the iBN nodes.

3.5.1.2.3 Routing Operations

SANSA operator will be responsible to define all the network routing protocols present and control the applicable routes to each iBN’s node. This information will be allocated in the database to be distributed around the network. iBNs will be responsible to apply the network routes in all the IP devices inside a node but HNM will be responsible to define the information details about how the traffic will be handled. The data model corresponding to this information will contain the routing information for all the network.

3.5.1.2.4 Quality of Service Operations

Quality of services policies definition will be allocated in these data model. Network operator will be the responsible to generate the rules or policies to follow, depending on the traffic type or class of service, always keeping a common classification to the IP network devices present. Once this QoS has been generated, it will be distributed around all the iBN nodes in the network by the configuration module.

3.5.1.2.5 Energy Saving Operations

The Energy efficiency actions related to the use of the resources such as frequency spectrum, throughput consumption or transmission power, have a direct impact on the overall power requirements, especially in rural environments where power autonomy is considered critical. In this table, we will find a clear definition about how the energy efficiency will be managed in all the devices in the network. SANSA operator will be responsible for this first definition and the HNM configuration module will be responsible to propagate the result of the algorithm calculations located in the database. Subsequent changes in the status of the devices will be based on the
suggestions from event handling operations. Energy saving operations could impact different tables such as Radio Resources configuration, topology or routing operations.

3.5.1.3 Event Handling Operations

3.5.1.3.1 Radio Resources Operations

HNM will be responsible to receive radio resources information updates from all the iBN nodes. This information, generated in the RF components, will be especially useful for our topology algorithm in order to determine any necessary topology change and to send the specific updates to the involved nodes. The operation of the radio resources subsystem will depend on the thresholds already allocated in the database. The configuration and event handling modules will later on propagate the information over the iBN elements. Additionally, tables related to these operations will contain information about the status of the links. This information could be used to generate mitigation plans to prevent link failures.

3.5.1.3.2 Topology Operations

Network congestion, link failure and operator changes, could provoke topology changes. This table will be collecting real time information from all the iBNs about the topology status, visible nodes, spare links and connections changes between others. This information will be used for the HNM calculation algorithms to identify possible new topology scenarios and generate the configuration changes in almost autonomous mode.

3.5.1.3.3 Routing Operations

All information about the network protocol performance focus on the routing table status and active routes of all the IP devices inside the network. This performance information, located at the database, support the event handling module to determine the configuration updates necessary in the system as a function of the thresholds defined.

3.5.1.3.4 Quality of Service Operations

Statistics, events and possible alerts related to all the nodes will be collected at iBN level. HNM will be responsible for the events and the configuration policies established to generate mitigation actions or possible QoS redefinition policies to optimize the traffic consumption for the applications.
3.5.1.3.5 Energy Saving Operations

Event information and statistics about energy consumption of all devices available in each node, transmission power, CPU use and throughput will be especially located in the database model. This information will be consumed by the configuration management module to determine the changes necessary in the network to warranty the goals established.

In the particular case of energy saving, the information collected at database should be marked in a specific way.
3.6 HNM Graphic User Interface

In the past, software was designed with little consideration towards the user, which meant the user had somehow adapt to the system. This approach to system design is not the case for modern systems that the system must adapt to the user. This is why design principles are so important.

Users should have successful experiences that will allow them to build confidence in themselves and establish self-assurance about how they work with the application interface. Each positive experience with a network management interface allows users to explore outside their area of familiarity and encourages them to expand their knowledge.

HNM graphic user interface aims to present the information necessary to operate the configuration management and event manager modules and the related subsystem to the SANSA operator. The information related to event handling should be presented in real time and should be mapped to the current nodes in the platform. Information about the node status could be presented in a topology diagram geolocated to provide geographic perspective about the system. Next section will describe the several interface capabilities necessary for the SANSA operations.

3.6.1 Interface Capabilities

Interface design principles [6] represent high-level concepts and beliefs that should be used to guide software design. It is necessary to determine which principles are most important and most applicable for the systems and then use these principles to establish guidelines and determine design decisions. The three areas of user interface design principles are:

- Place Users in control of the interface
- Reduce the memory load on users
- Make the user interface consistent

The first set of principles address placing users in control of the interface. The interface should also give them some degree of control and flexibility to do their tasks quickly, comfortably, and efficiently. In this environment, the interface designer must determine which of these principles are most important for the operators of SANSA system.

The user interface is considered as the main interaction point of a software product. If done well, user or operators don’t even feel that it is there. If done poorly, operators can’t get past it to effectively use the product. A goal of the interface is to help operators feel like they are reaching right through the interface and directly manipulating the objects they are working with.
The rich visual and sensory environment of graphical and multimedia user interfaces requires operators to be able to customize the interface.

User interface consistency is a key aspect of usable interfaces. Even though consistency might be a lower priority than other factors. One of the major benefits of consistency is that operators can transfer their knowledge and learning to a new program if it is consistent with other programs they already use. Some of the main principles to create a success user interface and user experience are presenting next.

- Sustain the Context of Operator Tasks.

Operators should be provided by points of reference as they navigate through a product interface. Window titles, navigation maps and trees, and other visual aids give users an immediate, dynamic view of where they are and where they’ve been. Operators should also be able to complete tasks without having to change context or switch between input styles. If operators start a task using the keyboard, they should be able to complete the task using the keyboard as the main interaction style.

- Maintain Consistency within and Across Elements

One of the most important aspects of an interface is to enable the operators to learn general concepts about systems and products and then apply what they’ve learned to new situations in different programs or different parts of the system. This consistency applies at three levels: presentation, behavior, and interaction techniques. Consistency is one of the key issues behind user interface guidelines.

Consistency in presentation means that operators should see information and objects in the same logical, visual, or physical way throughout the product. For example, if the information that users can’t change (static text) is in blue on one screen, then static text on all other screens should also be presented in blue. If a certain type of information is entered using one type of control, then use that same control to capture the same information throughout the product.

Consistency in behavior means that the way an object works is the same everywhere. The behavior of interface controls such as buttons, lists, and menu items should not change within or between programs. Operators should not be surprised by object behaviors in the interface.

SANSA project requirements are focused on the necessity to create more than just an operator user interface. In this case, it will be necessary to generate a network management system that includes some basic features such as:

- Online Monitoring & Control, with an online connection to all the components and software enabled control over the main functionalities.
• Off-line parameter viewing, without an online connection.
• Central Event Log Storage.
• Connection Management.
• Backup and Software Update Management.
4 IBN Component Description and Architecture

Intelligent backhaul Node (iBN) is one of the main components that implement the SANSA ground (or terrestrial) transport network. The iBNs, which are distributed throughout the SANSA backhaul mesh network. They shall be responsible to establish the configurations necessary to distribute and forward the user traffic between the local elements under control. Additionally, it should be able to perform network decisions on a short (e.g. routing) and medium (e.g. energy efficiency) time-scale basis.

4.1 iBN Software Architecture Design

The iBN goal represents more than just an integration between the node elements and the HNM. It should be considered as a control element for satellite modems, terrestrial modems, HeNB smart antennas and communication with EPC between others with focus on configuration management and event handling for all the components involved.

Each iBN node will be formed by at maximum of four components - terrestrial modems, satellite modems, HeNB and the iBN Core server (responsible for the configuration, local routing and event handling functions).

The iBN will be responsible to establish the communications with all these components in order to control the radio access parameters related to the links in the terrestrial and satellite modems, networking configuration between all the elements in the node, event handling for all devices, possible energy saving actions over the devices and execute changes that could impact the topology distribution. Additionally, iBN will have the function to exchange information with neighbor iBNs to establish connections with the HNM. The HNM is represented as the main brain of topology calculations with iBNs possessing the capabilities to establish a stable communications link with the HNM for secure and real time exchange of information about the network status at all levels.
4.1.1 Guidelines

One of the main principles presented in section 3.1.1 for software architecture design in the HNM was to keep the consistency across the solution and the elements involved, especially when operators will play an important role. Keeping this in mind, and taking into consideration the requirements received for SANSA environment, the recommendation is to use a 2 Tier architecture at iBN level. The modular system architecture already defined for the HNM will be a compatible model when extrapolated to the iBN.

The iBN will be composed by the three main functionalities previously defined at HNM level - commissioning, configuration and event handling functions (Figure 26).

Commissioning Management operations shall represent all the necessary actions established with the core module to validate each iBN node against the HNM. Between these operations it shall include the establishment of a connection with the HNM, AAA validation process and configurations download for all iBN node components. This process will be necessary just between the HNM and iBN core modules. A more detailed description will be presented in section 4.2.

Configuration Management, this application running in the iBN will be responsible to apply any possible changes received from the HNM to all the components that are considered as part of the node. The first configuration information should be downloaded at first bootstrap by the commissioning agent and located in a local database. After that, the configuration agent should be responsible to apply any possible changes received from the HNM to all the node components. More information details will be presented in section 4.3.

Event handling functionalities represented by an application running at core module level will be responsible to get all the event or alert messages from the node components, classify these elements and to allocate them in the local database. Additionally, it will send the update messages to the HNM to inform about possible or potential configuration changes necessary to the nodes.
Three logic modules (Figure 27) will be acting at iBN, to control the different sub-systems present. The common definition between this definition and the HNM acts adding architecture consistency across the elements.

Figure 27 iBN Modules and Main Subsystems High Level View

In the particular case of the iBN, an additional module for Data Management has been included at the bottom of the architecture. In this case, the message transformation functionalities will be a necessary step to warranty the integrations between the local elements present in each node. On the other hand, the message routing functionalities will be necessary to route the data transformed to the correct logic modules inside the iBN.
4.2 Commissioning Management Module

This module will be responsible to run all the necessary operations and establish the first connection between iBN nodes and the HNM. This process is denominated as “commissioning process”. In order to establish this connection, it will be necessary to use a preloaded configurations settings provided by the operator. Once the connectivity with the HNM is established, the module will proceed to download all the configuration parameters corresponding to radio resources planning, network configuration, network topology, QoS policies and Energy Saving parameters. This information will be located in the data configurations table present in the local database. As a next step the iBN will proceed to establish the connections with the local elements, get the current configuration and compare the database information to determine inconsistencies. Once, the possible inconsistencies are resolved, the traffic flow between the local elements passing within the iBN will be established, finishing the commissioning process.

4.3 Configuration Management Module

The iBN configuration module will be responsible to support all necessary configuration processes for local devices operation. HNM is the element responsible to send the configuration updates to the iBN database. This module will check the database to detect any changes on the configuration tables. Once a potential update is detected the agent will send the new parameters to the element involved. Configuration module will be responsible to validate the correct application of the changes in the local elements and confirm to the HNM that the process has been completed.

4.4 Event Manager Module

The iBN event manager module shall be responsible to collect all the events generated by the local devices present as part of the nodes. Additionally, it will have the capabilities to receive information from the neighboring iBN’s. All this information will be received and stored in the local database. The final destination of all this information should be the HNM which will use this information to control the platform and to optimize the use of the resources in each iBN from an RF, networking and energy efficiency perspective.

In order to collect all this events information, the module will connect locally using common protocols with the terrestrial and satellite modems interfaces. During this process, information is
processed and saved in a local database to apply the data model and forward the information translated to the neighbor IBN or the HNM directly.

The events should be classified using the main functionalities defined by SANSA, namely- Radio Resources, Routing, topology changes, Quality of Services and energy saving. Additionally, all the events corresponding to the iBN server itself should be collected.

### 4.5 iBN Core Module

Core module will be considered the main component of the iBN, responsible to allocate the functional intelligence to orchestrate the logical operations. iBN Core module shall be represented in a core server component, enabled to interact between the local components and the backend elements, providing a mediator layer with message translation, database allocation and application functions.

In order to warrant a correct and consistent data exchange between iBNs and HNM, a common specifications related to the data format and values is necessary. Data model should be aligned with the type of data expected to be shared between the components and the interfaces used.

This data model definition should cover the main functions of the Core Module. The interactions and use will be especially important for the agents and the database operations.

#### 4.5.1 iBN Core Database

iBN local database shall be responsible for allocating all the data model necessary to warrant the operations for the already defined system. This local database will contain configuration information received from the HNM and Event information generated by the local elements and transform by the iBN Messages transformation module. Additionally, the information could be consumed by a local user interface for the SANSA operator. The main operations to be covered on this database will be detailed in the next sections.

##### 4.5.1.1 Commissioning Operations

The data model definition for these operations are based on offering coverage for the initial routine operations and the authentication actions needed to provision each iBN node present in the network.
4.5.1.1 Initial Routine

iBN core database should contain a specific data model oriented to support the step by step routine followed to complete the initial bootstrap process for each iBN node. This information will include the system information present in each iBN node and the software protocol necessary to follow to complete the provisioning operation.

4.5.1.2 AAA Operations

AAA operations acts at two different levels, one of them is present between the HNM and IBN the second one will be defined to warrant the security aspect between each IBN node and the local elements such as satellite modems, terrestrial modes and HeNB. IBN local database should contain the configuration necessary to establish the first AAA connections with the local elements present in the nodes.

4.5.1.2 Configuration Operations

Brief description about main configuration operations and database interactions are presented in the next sub-sections.

4.5.1.2.1 Radio Resources Operations

Definition about all the available used resources for the local elements present in the node should be allocated in the database. The iBN will be responsible to keep all this information updated and apply the changes received from the HNM in the devices involved.

4.5.1.2.2 Topology Operations

Using the modules and subsystem architecture, the HNM will define the topology present in the system. This information will be sent to all iBNs in order to keep a local copy and apply possible changes received from the HNM to the involved devices.
4.5.1.2.3 Routing Operations

iBN will take all the routing information from the HNM in order to apply the correct routing rules in the local devices. Additionally, the iBN will be responsible for routing the user traffic between the different components to warranty service availability and the correct network protocol operation.

4.5.1.2.4 Quality of Service Operations

The QoS policies are defined at the HNM but it is the iBN that is responsible to apply these configuration changes over the local elements. A mirror of the configuration should be kept in the local database.

4.5.1.2.5 Energy Saving Operations

This table should keep information related to the energy saving rules that will be applied in the local elements to optimize the energy consumption in relation to the throughput present in the network or the KPIs already defined for this purpose.

4.5.1.3 Event Handling Tables

One of the main functionalities in the IBN have been based on event and alert gathering. In order to warrant a real alignment between the components and to make possible an efficient and correct handling of these events, it is necessary to group and define a minimal structure. One suggested classification is based on the five main functionalities to be covered by the iBN system. Each one of these functionalities will represent a specific table where it is possible to collect all the main information fields necessary to control the network.

4.5.1.3.1 Radio Resources Operations
All the events and alerts received from local components or the HNM indicating a possible change in the radio resources will be located in this table. A basic classification for these events should be agreed and aligned to warrant the correct handling by the HNM in order to define possible updates of radio resources.

4.5.1.3.2 Topology Operations

Examples of topology events include network congestion, link failure and operator changes. These events will be handled by the iBN, receiving the information and sending the alerts information to the HNM. All the events and statistics will be located in this local database.

4.5.1.3.3 Routing Operations

Local information and messages related to the route status will be collected and located in this table. Only update information will be send to the HNM in order to determine possible changes in the routing policies or a topology change.

4.5.1.3.4 Quality of Service Operations

Event collection about status and statistics of policies will be located in this table. A common parameter structure inside the iBN’s about policies applied and packet delivery and marks should be collected. Different groups of alerts definition will be presented at this stage applying a retention policy that enable to optimize the interaction with the HNM in order to determine configurations update and present the most relevant information at the SANSA operator GUI.

4.5.1.3.5 Energy Saving Operations

Device events about power consumption status in real time will be registered in this table. in addition to information related to the local use of CPU, transmission power or general consumption. This information will be sent to the HNM for processing and threshold validations.

4.6 iBN User Interface

The main interface power, described in Section 3.6, should be located at the HNM. This interface should be able to execute any kind of configuration task at the iBN level. However, it would be
recommended to present at least a command line interface, running at iBN level. Some of the advantages in providing this kind of capability is:

- **Scriptability** - specially performed from command line interfaces (CLI). It is easy to create and run every time and results will always be the same, avoiding human error or additional variables that can influence the result.
- **Expressive Potential** - scripts permit to use a CLI and evolve complex programming languages. We can find variables, arithmetic, logical and character computations, support different types of loop.
- **Minimal Environments** - CLI may be offered over GUIs. There are systems that lack either graphical displays or the storage cycles required.

Additionally, CLI presents the availability of an organized two dimensional display with rich content navigation widgets, this makes it possible to make more complex outputs easily understandable.

A CLI shall be highly recommended for the iBN commissioning process. This step is critical for the system start up and containing minimum configuration information could be easily implemented and can offer a local control mechanism for the elements in addition to the main user interface presented by the HNM.
5 Conclusions

Software architecture design principles suggested for SANSA key elements (HNM and iBN) look to provide an effective foundation - to create a layered and modular architecture that offers flexibility, reduces the time to market and to develop and implement new software versions. Additionally, a well interfaced structure definition will make the final solution less complex and more manageable from an integration perspective.

Sharing some main principles in the design of the HNM and iBN components offer advantages, especially in software engineering task. Perhaps, SANSA operator will find consistence and coherence in the operational logic used.

One of the main challenges in SANSA will be located in the integrations tasks. iBN will present the interfaces to the local devices and it should be able to understand the common communication protocol between them and have the capabilities to integrate new interface protocols. The capacity to introduce potential third party plugins will be key to keep the usability of the solution during the time.

SANSA solution description presented in chapter 2 offer a high level view of the environment, necessities and the scope defined in previously in deliverable 2.3. The presented Transport architecture describes the environment elements and the main interaction expected between components.

Chapter 3, navigate deep definition of the hybrid network manager architecture. A structure based on modules and subsystem has been suggested to cover the different required functionalities and follow a coherent distribution between the components. HNM is considered the main element in the SANSA architecture and as such, a detailed definition about the data flow exchange expected between inter and intra node components is provided. Some important guidelines have been postulated for the HNM graphic user interface design, especially oriented to SANSA operator.

In chapter 4, the iBN is presented as a local orchestrator sharing the main part of the development logic with the HNM and taking the role to support the communications, configuration exchange and event gathering with the local elements. Its use allows to optimize the data exchange between the nodes and a central location, the HNM and offers flexibility to the solution.

In summary, this deliverable presented the system architecture for the key elements of the SANSA network architecture, including the components and subsystem description.
References


