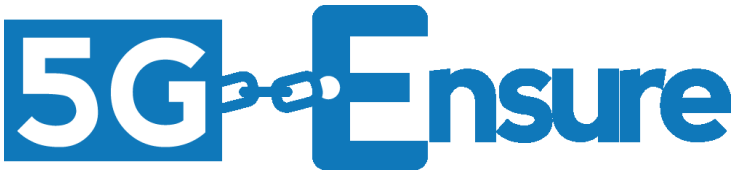

System Security State Repository

Release Note for R1

Project name	5G Enablers for Network and System Security and Resilience	
Short name	5G-ENSURE	
Grant agreement	671562	
Call	H2020-ICT-2014-2	
Delivery date	2016-09-30	
Lead beneficiary	IT Innovation	Mike Surridge, ms@it-innovation.soton.ac.uk
Authors	IT Innovation: Mike Surridge, Stefanie Wiegand, Stephen C Phillips	

Contents

1 Introduction

This release note accompanies the model (ontology) which is the R1 release of the Secure System State Repository. As the R1 release is not software, it does not have a software manual in D3.4. This release note serves to explain the model for those that may wish to make use of it.

2 Asset modelling approach

2.1 Overview of asset types

The high level classification of asset types is shown in Figure 1.

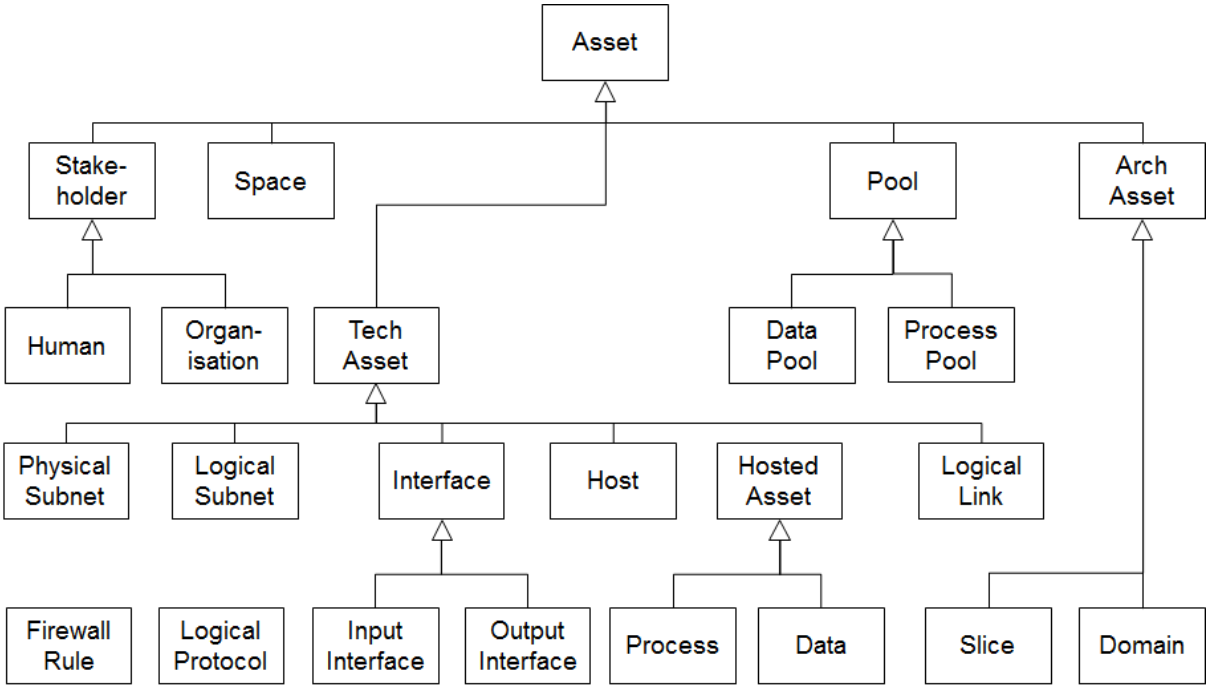

Figure 1. High level asset types.

The top level subclasses capture some basic aspects:

- Stakeholder: represents a controlling interest, e.g. a network operator, technology supplier or network device user, etc.
- Space: represents a region of space from which certain technological assets or network components are accessible.
- Tech Asset: represents a (logical or physical) technology component used to create, operate or use the network.
- Arch Asset: represents an architecturally significant asset collection, at this stage limited to architecturally defined network domains or network slices.
- Pool: represents the notion of run-time choices in processing data.

The last of these is a difficult concept to when considered as a network asset. The need for such a concept was discovered during the FP7 OPTET project, where it proved necessary to fully capture the interactions between processes running in a network. There are two types of Pool:

- Data Pool: represents the choice of which data item to use in a process, e.g. which customer's records to update.
- Process Pool: represents the choice of which other process to initiate an interaction with by exchanging messages with it, e.g. which service a client should bind to.

The Tech Asset subclass represents network technology components, of which there are several subtypes:

- Physical Subnet: represents a L1 + L2 network that can provide communication links between locally connected devices
- Logical Subnet: represents an L3 network segment supporting message routing between connected devices within and beyond the local segment.
- Host: a device that may be connected to a network, capable of sending, receiving, storing and processing data.
- Interface: represents the connection between a Host and a Logical Subnet.
- Logical Link: represents a communication path through a network
- Hosted Asset: represents data items, or processes that can access or alter data items. The name reflects the fact that data is stored and processed by a Host.

These asset types are the ones used to model network connectivity and provisioning.

2.2 Physical Network Provision and Security

Physical network provision and access control are modelled by relating physical subnets to devices that provide physical interconnection mechanism, and also relating the devices and interconnection mechanisms to regions of space in which they are accessible. This is shown in Figure 2.

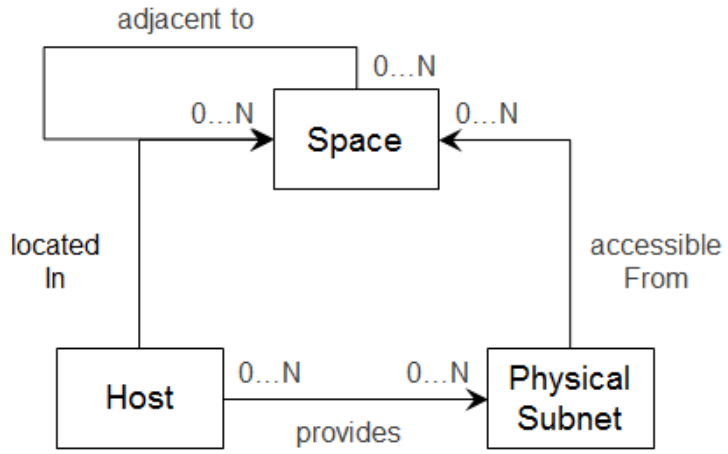

Figure 2. Network provision and accessibility.

Space assets will normally be used to represent:

- regions that are 'within range' of a radio network, where it is possible to connect to the network, and/or interfere with it, e.g. by jamming the network;
- regions within which host devices are located, which must be entered in order to physically access the device, and/or remove or tamper with it.

Spaces may be adjacent, meaning it is in principle possible to move from one to another, if physical security measures don't prevent this. Spaces may also overlap, e.g. the regions in which different radio networks are accessible might overlap. Spaces should not contain any internal boundaries that might restrict movement, so if one can enter a space one may move to any part of the space. Where such boundaries do exist, one should define several Spaces representing the sub-regions within which movement is not restricted. This implies that one cannot use security measures to prevent movement between overlapping spaces.

To give an example, one might model a house that may be entered from a surrounding garden that may be entered from the surrounding world. The house might contain a WiFi broadband router that is only accessible within the house, and provides a WiFi physical subnet that is accessible from the house and surrounding garden, plus a wired physical network that is only accessible inside the house. It would be possible to connect to (or jam) the WiFi from the garden, but to physically interfere with or steal the router or connect to its wired subnet one must enter the house. This may be prevented by installing locks.

The example also shows how physical network provisioning can be modelled in terms of devices. It is important to model at this level as this is where the network might be attacked by interfering with the device providing the network. We assume remote attacks (where the attacker is *not* in a Space giving them physical access) will be possible if the network is provided by a Host, i.e. a device that can store, process and communicate data. Simpler devices are more difficult to attack, and need not be modelled separately from a physical subnet. The broadband WiFi router in the above example can be attacked, and should be modelled as a Host providing WiFi and Wired physical subnets. However, a physical LAN provided by a coax ethernet cable could be modelled using only a Physical Subnet, since there is no way to attack a coax cable without physical access to it.

Note that in most end-to-end networks, it will impractical to model the physical networks and their relationships to secure or insecure spaces for the entire network. This is due to the fact that the physical details will often not be known (or relevant) to the stakeholder creating the model. In the above example, suppose the homeowner also uses a remote backup server accessed over the Internet. They probably could not (and need not) model the physical data centre in which this is hosted, nor the physical layer of the network beyond their own router.

2.3 Physical Subnet and Host Subclasses

Subclasses of the Physical Subnet, Host and Space classes may be used to represent any specialised variants where they are subject to different types of attacks or could be protected by different security measures. There is no need to distinguish between variants where they are subject to the same attacks and can be protected by the same security measures.

At this stage we have not specified classes to represent threats or security measures, so at this stage there is no need to define subclasses of these asset types. They will be added in later versions of the schema

when we have modelled threats and security measures, and it becomes necessary to indicate which affect each type of Host and Physical Subnet.

It seems likely that we may need to use Host subclasses to distinguish personal computers from multi-user servers, mobile from fixed devices, and possibly virtual from physical hosts. Physical Subnet subclasses may be needed to distinguish between wired, WiFi and different generations of cellular networks, and satellite networks. It may also be important to model communication between devices using removable media or by printing data on paper. This may be important in sectors such as health care where information leakage in printed form or on removable media is a concern, and regulations require security measures to prevent this happening. Figure 3 shows some subclasses being considered for inclusion in the next version of this schema.

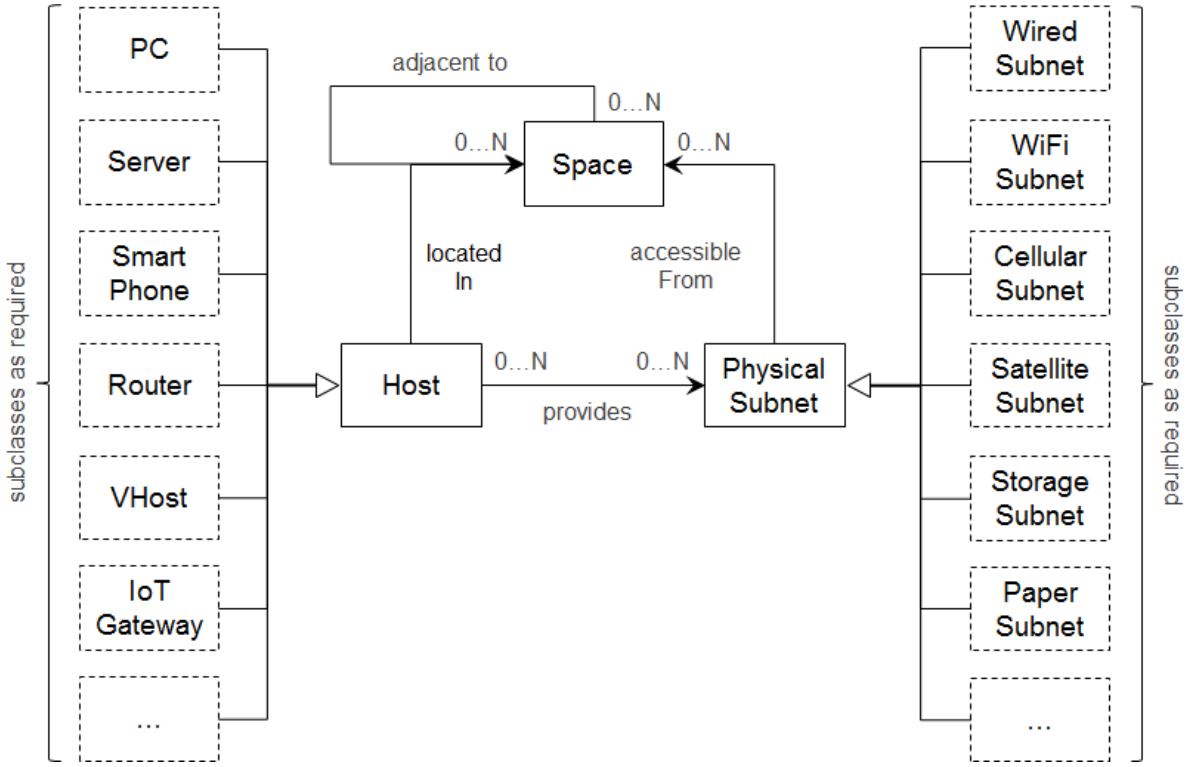

Figure 3. Possible physical network and host subclasses.

2.4 Physical and Logical Network Connectivity

Connectivity is expressed through relationships between Tech Assets. The basic approach uses ideas similar to those from OGF NML[1]:

- Subnets describe parts of the network that support communication between sources and destinations, corresponding to the NML Node concept.
- Interfaces describe possible communication sources and destinations, corresponding to the NML Port concept.
- The NML Topology concept has no direct equivalent here, but could be constructed from a set of subnets and interconnecting Hosts and Interfaces.

[1] Open Grid Forum Network Mark-up Language: https://redmine.ogf.org/projects/nml-wg

The relationships between physical and logical networks and connected devices are shown in Figure 4.

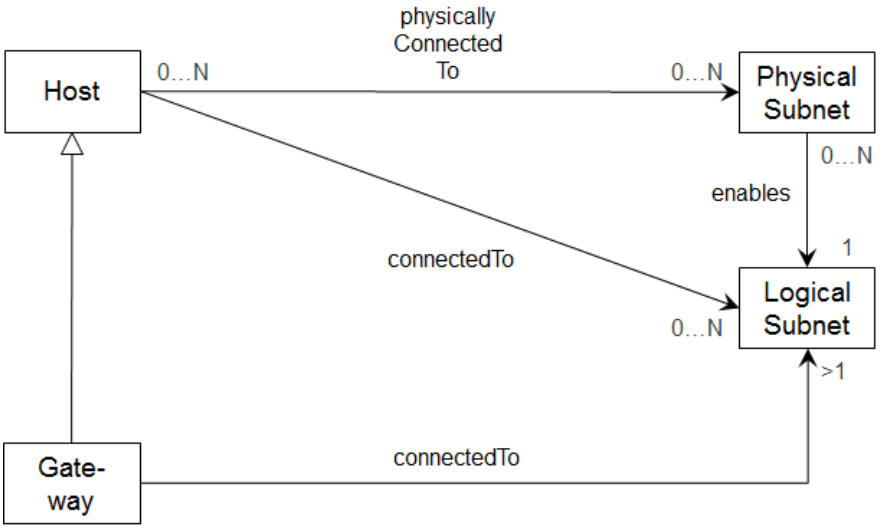

Figure 4. Physical and logical network connectivity.

Physical Subnets provide only link-local connectivity. Message routing is considered to take place only over Logical Subnets which are enabled by one or more Physical Subnets. Hosts may be connected to both Physical and Logical Subnets. Message routing between subnets is handled by Gateways, which are modelled simply as Hosts that are connected to multiple Logical Subnets and can route messages between them.

As noted above, it will rarely be practical or necessary to model the physical aspects from Figure 2 in parts of (or sometimes throughout) an end-to-end network. For this reason, it is acceptable for network segments to be represented by a Logical Subnet and connected Hosts, with zero Physical Subnets, as indicated in Figure 4.

2.5 Interfaces and Addressing

The main difference between our approach and NML is that we explicitly represent devices connected to the network that can send, receive and also store and process data (Hosts). The NML Port only represents points where data can enter or leave a network. Our model represents this using an Interface class, which represents a unidirectional connection of a Host to a logical network, as shown in Figure 5:

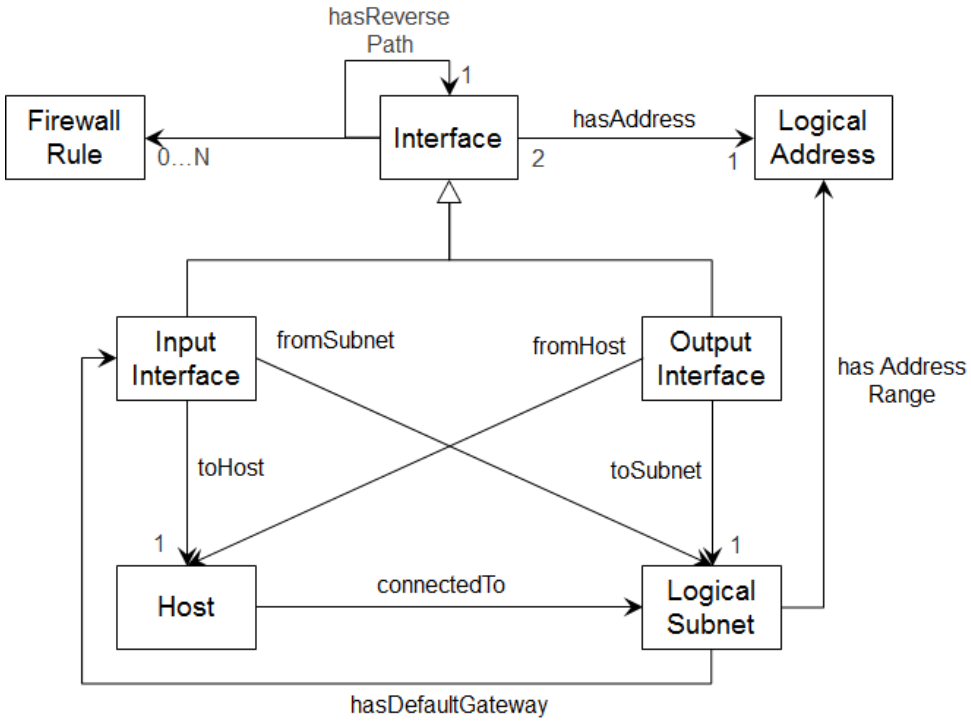

Figure 5. Logical interfaces.

The Interface is needed to support the modelling of threats to hosts that arise in only one specific subnet. In such cases it is helpful to have an asset representing the connection of the host to or from that subnet, so one can assert that such a threat threatens an interface rather than an entire host. For example, a packet flooding DoS attack on a Host from one network would make the inbound interface to that network unavailable, but not the whole Host. At present, Interfaces are only defined to and from Logical Subnets, because that is the layer at which routing between subnets is supported, and most such attacks come from outside the local subnet.

Interfaces can have a logical address attribute associated with the Host on the Logical Subnet to which the Interface is connected. Each Subnet can also have a logical address representing the range of addresses that can be reached directly on that Subnet. These addresses may be technology specific and so can be modelled as String-valued attributes, e.g. as IP addresses or ranges in CIDR notation.

Note that if a Logical Subnet or Interface class is not a singleton then it may have multiple instances. Non-singleton Logical Subnet classes will normally represent private subnets, such as home networks of connected subscribers. Any private address range would be acceptable for such networks, any of which can be used to represent them all as a class. Where non-singleton Host classes are used (e.g. to represent all personal workstations connected to a LAN), the address property of the corresponding Interfaces should be a range. By default this will be the range of addresses used on the corresponding Logical Subnet, but a subrange may also be specified, e.g. to avoid clashes with fixed addresses assigned to singleton hosts (e.g. servers) on that Logical Subnet.

The Logical Interface also provides a convenient point where security controls representing traffic restrictions (i.e. firewall rules) can be attached to the model. Strictly speaking the rules are processed at a Host, but different rules may apply to different interfaces depending on which subnets they are on, and whether they represent incoming or outgoing traffic. At present, each Firewall Rule should contains a string

representing the rule (using nftables or iptables syntax) plus a serial number allowing the rules to be organised into a sequence. If we ever need to analyse the rules themselves, non-semantic reasoning can be used.

2.6 Data Storage and Processing

The storage and processing of data is represented by the Data and Process concepts, as shown in Figure 6.

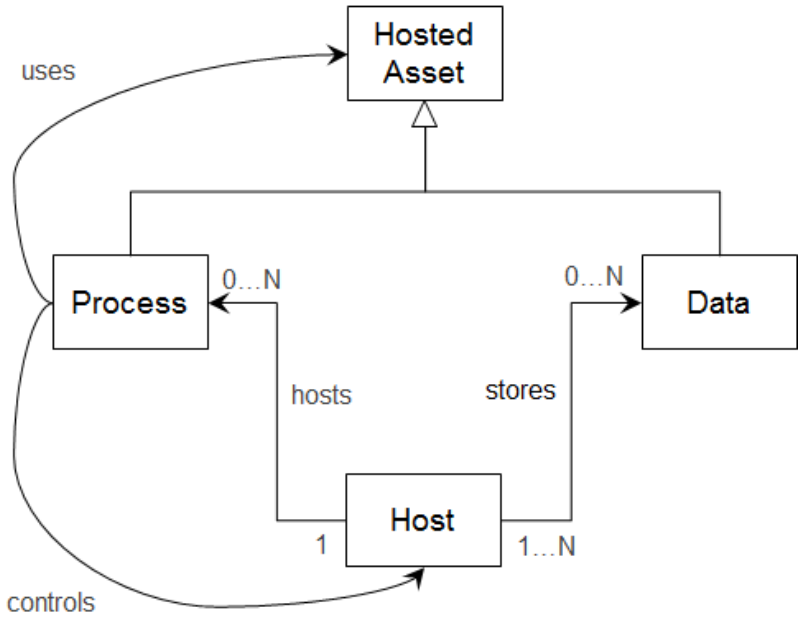

Figure 6. Data Storage and Processing.

The Process represents any process that may be implemented by running software on a Host. The Data concept represents information stored on a Host. Normally, one would only model data items that have security significance, e.g. because it is confidential or sensitive or because security mechanisms depend on it. Data should be modelled in the least detail necessary to capture security requirements to protect data, and the dependence of security measures on data.

A running Process can only be executing on one Host, but because data can be copied, it may be stored at multiple Hosts. One would normally represent copies that are meant to remain consistent as one Data item stored on multiple Hosts, while copies that may be altered independently of each other are represented as multiple Data items each on its own Host.

Processes can directly use and modify data that is available on the same Host. If the data is not local, the only way to access it is by interacting with another process that is co-located with the data. Both these interactions are represented by a 'uses' relationship between the Process and the Hosted Asset (i.e. Data or Process) with which it interacts.

A process can obviously (if sufficiently privileged) modify the configuration of its host. In this sense, processes provide a means to control hosts, including any other processes they host or subnets they provide. Modelling this control makes it possible to model threats designed to target those processes in order to usurp control of specific parts of a network. Modelling management of entire network domains by responsible stakeholders is handled in a simpler way, as described below.

2.7 Run-Time Process Interactions and Delegation

Whenever a Process uses a Data item or another Process, there is an element of choice involved. The Process decides whether to initiate a usage interaction, and also with which candidate process or data item. These choices depend on whether the Process is acting on its own behalf, or being instructed by some other Process. Control of these choices must be captured so we can model threats that disrupt the chain of command and exploit delegated privileges, e.g. in so-called 'confused deputy' attacks. This is captured by using a Pool asset as shown in Figure 7:

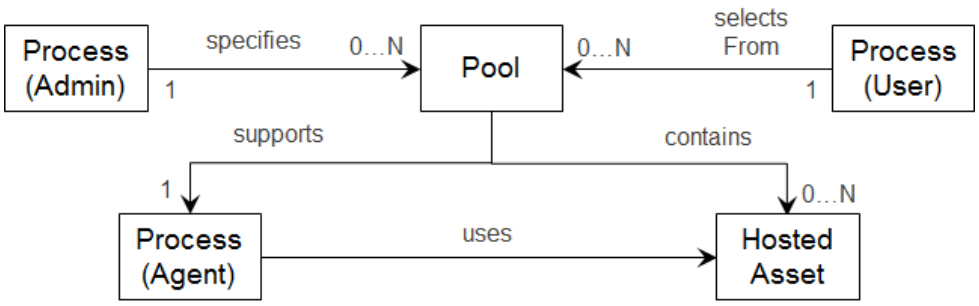

Figure 7. Control of Process usage interactions.

The Pool allows three distinct roles to be assigned with respect to a usage interaction. The Process that actually interacts with another Process or Data item is considered to have the role of an Agent. The role of User is assigned to the Process (which may be the same or a different Process) that initiates the interaction and decides what to interact with. The other role of Admin is assigned to the Process (which may be either of the first two) that decides which Hosted Assets are candidates from which to choose.

For example, a service for storing and retrieving electronic patient records could potentially use the records of any patient. The service controls which patient records are stored so it would have both the Agent and the Admin roles. However, the service retrieves records as requested by another process (e.g. a record administration client) which has the User role with respect to the selection between patient records.

If the Hosted Asset being used by the Agent is a Data item, the Pool is a Data Pool. If the Hosted Asset being used is another Process, that Process is considered to have the role of a Service. The Pool is a Process Pool which has some extra relationships, as shown in Figure 8:

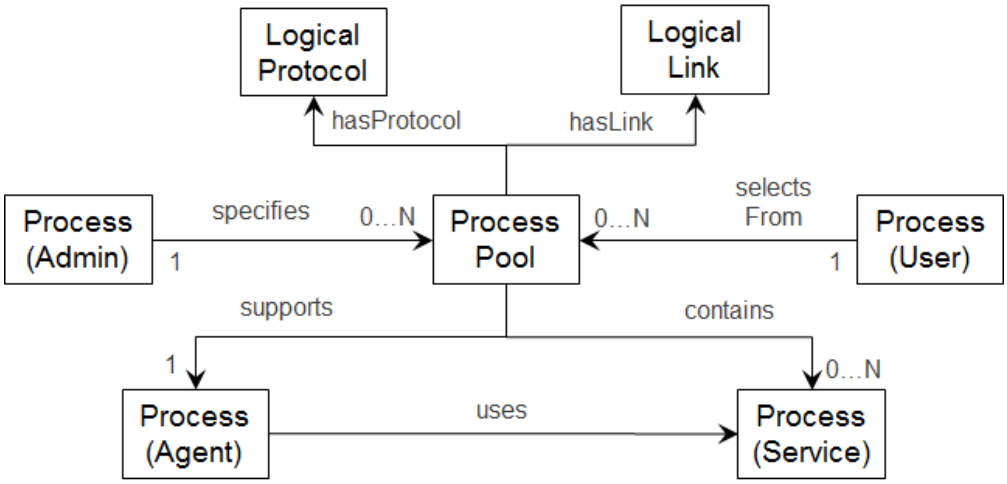

Figure 8. Choice and control of the use of a Process by another Process.

The extra relationships are needed to capture the fact that a usage interaction between two Processes is mediated by communication between their Hosts. This is represented via two additional entities that have relationships with the Service Pool:

- a Logical Protocol entity, consisting of the port number used by the Service in Figure 6, and whether or not the transport used is connection based (like TCP);
- A Logical Link entity, which captures the path through the network over which this protocol will need to pass.

These are shown in more detail in Figure 9:

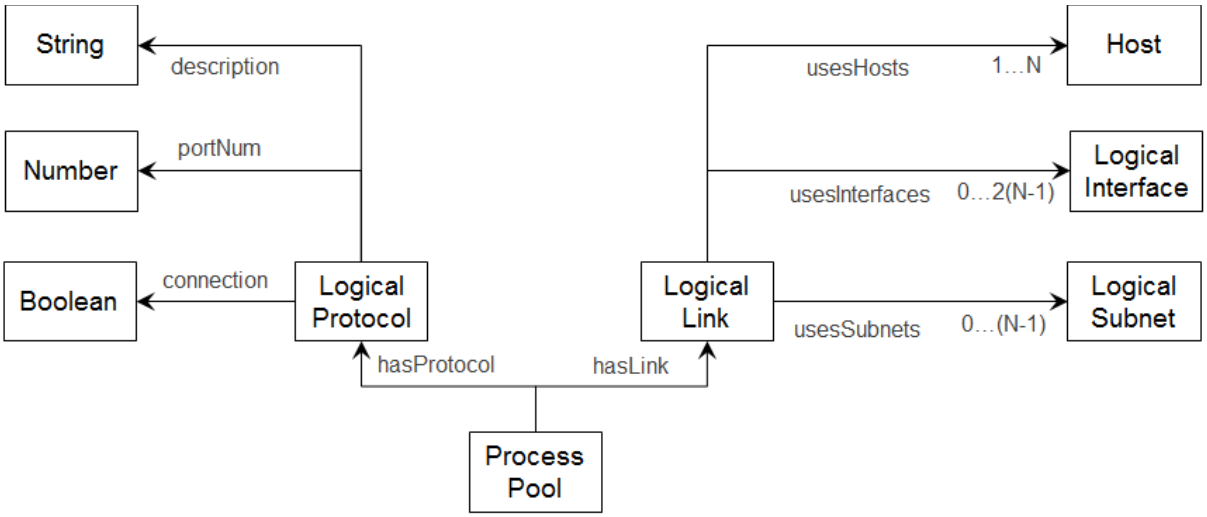

Figure 9. Inter-process communication links and protocols.

These two pieces of information make it possible to deduce the firewall rules needed to allow inter-process communication between processes modelled in the network. From this, one can check if the firewall rules actually enforced at each Interface in a network are consistent with the modelled requirements.

Note that the Logical Protocol entity can be reused wherever the same protocol is used (e.g. one Logical Protocol can represent all uses of HTTP within a network model). The Logical Link is usually specific to one

Process Pool corresponding to one uses relationship between two Processes, because it captures the path between the Processes, including their Hosts, the interfaces between these Hosts and their Logical Subnets, and any intervening Logical Subnets and Gateways and their Interfaces.

The 'uses' relationship between two Processes is analogous to the NML Bidirectional Link concept, as it represents an interaction that involves passing information between Hosts, i.e. between Interfaces (analogous to NML Ports) that feed information to or from the network. We relate these data communication paths to the processes that use them.

2.8 Virtualisation

Virtualization of network components is defined as a relationship between one or more processes and an emulated network component. The idea is that a virtualized network component can be provided by executing software (Processes) that use the capabilities provided by their Hosts and interconnecting Subnets to emulate the virtualized network component.

At this stage three types of emulated components are envisaged as shown in Figure 10:

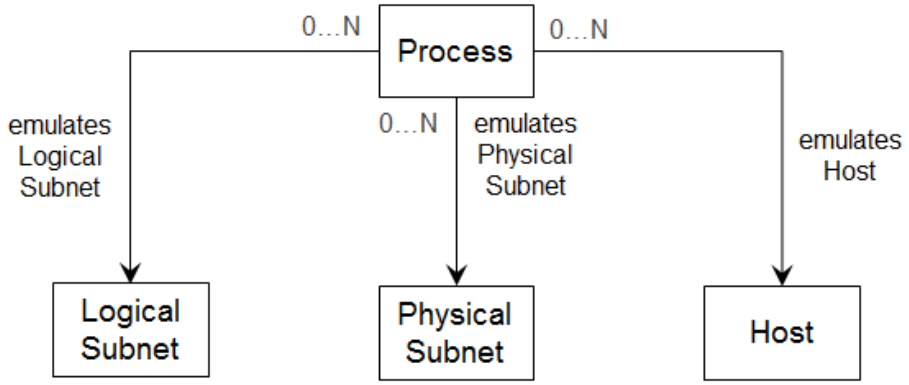

Figure 10. Network asset virtualization.

These correspond to virtual Hosts (i.e. Processes that can host other Processes and provide access to data storage), virtual switches (i.e. Processes that can provide layer 2 connections between Hosts), and virtual routers (i.e. Processes that provide layer 3 messaging between Hosts).

The OGF NML model doesn't include Hosts, of course. It does support switching services, which correspond to Physical or Logical Subnet emulation in our schema. NML also has a concept of "adaptation" and "de-adaptation" services, which handle the multiplexing of many data sources or sinks (Ports) to or from a single source or sink. The analogous concept in our schema is Interface emulation, but this is not necessary because Interfaces only exist where a Host is connected to a Logical Subnet, so if either is emulated by a Process, so is the Interface between them. We do not need to model adaptation and de-adaptation simply because we do model Hosts as well as Interfaces.

We also assume that in some cases a set of processes may be needed to emulate a virtual device. These processes together provide a virtualisation platform based on the capabilities of multiple Hosts and intervening Subnets. This makes it possible to model virtual Hosts that can move between physical Host to Host, e.g. to balance loads between physical Hosts or network connections. However, we assume in such cases it doesn't matter which physical Host is running the device emulator at any given time, so one can only specify which Hosts may be used by the platform processes.

2.9 Stakeholder engagement

The last aspect to be considered is how stakeholders are related to a system and to each other. Several possible relationships may exist, where the stakeholder:

- supplies technology components (assets) used in the network;
- accesses network assets for some application purpose;
- has a stake in the security of data stored or processed in the network; or
- manages part or all of the network.

These relationships are summarised in Figure 11:

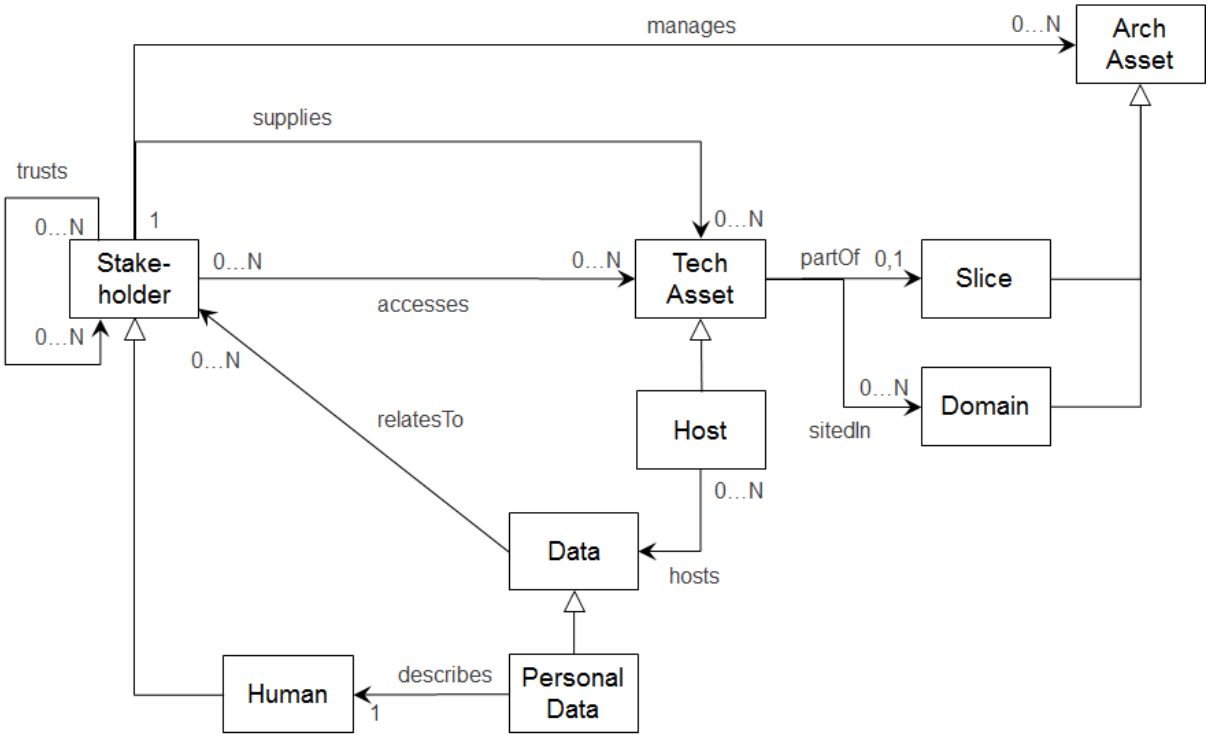

Figure 11. Stakeholder relationships.

The relationships shown in Figure 3 relate to stakeholder roles:

- the supplier of each technology component, responsible for the correctness of its implementation, including
  - the equipment manufacturer for hardware devices
  - the software provider for software used to process data on those devices
- the manager of network domains or slices, which will include
  - mobile network operators (MNO) who operate home or serving domains
  - virtual mobile network operators (VMNO) who may operate a home domain and buy serving domain capacity from other operators
  - network access providers who provide transfer network capacity to other operators
  - enterprise network operators who may manage a network slice based on resources allocated by one or more operators of multiple domains
- end users of networks including

    - individuals who may operate mobile phones and use the network by subscribing to an MNO or VMNO
    - wireless sensor networks (WSN) owners/operators who use
    - enterprises who purchase and use managed slices to provide enterprise networks
    - employees who use networks procured by their enterprise
- data subjects, i.e. individuals whose personal data is stored and may be processed in a 5G network, including users of the network, or of services that use the network.

See 5G-ENSURE Deliverable D2.2 for a more detailed list of stakeholders and descriptions.

At this stage it is assumed that each network component (i.e. technology asset) is supplied by one stakeholder. This stakeholder is responsible for the implementation of the asset, i.e. the software and/or hardware elements from which it is formed. In practice some assets like Hosts may incorporate hardware and software elements from different organisations. These details cannot be included in a network model without making it intractable, so we regard the supplier of the component to a network operator or user as solely responsible for its correct function.

It is also assumed that one stakeholder will be responsible for managing each portion of the network corresponding to either a domain or a slice. This includes all the technology assets that form part of the domain or slice. Management in this sense means they have the ability to configure devices and influence the way they provide their intended functions, within the limits imposed by their implementation. For example, management involves defining access policies, resource allocation policies and priorities, and so forth.

Each technology asset can also be accessed by zero or more stakeholders. They can use functionality provided by the asset but not influence how the functionality is provided by the asset.

Finally, security relevant data items may be related to stakeholders, meaning the stakeholder has a concern for the security of the data. Typically this is because the stakeholder owns the data, or is legally responsible for its security. One special case is the relationship between personal data and the data subject, i.e. the Human stakeholder described by the data. The concerned stakeholder is not necessarily the one managing the domain or slice containing Hosts where the data is to be stored and processed (although usually that stakeholder is also concerned about the data).

One consequence of all these relationships is that stakeholders are likely to be dependent on each other, and this implies that trust relationships must exist between stakeholders, as shown in Figure H. For example, if a technology asset is part of more than one domain or slice, it means it may be managed by more than one stakeholder. These stakeholders will need to trust that the other will not violate their interests. Trust relationships must also exist between the supplier, manager(s) and user(s) of a technology asset, or between managers of a network slice and the managers of domains containing assets hosting the emulator processes providing virtual assets in the slice.