

IGW related Excerpts from D2.1

1.1.1 The Interconnection Gateway (IGW)

There are a number of problems to be solved for interconnecting the testbeds of different addressing and security schemes; the most important of which is to hide the network topology of a site by applying network addresses and protocol translations. Quote from Panlab SSA D2.2 Section 4.1: “In order to establish connectivity, a Virtual Private Network must be set up [...] hiding the complexities and allowing Panlab partners to dynamically provision multiple overlay networks“

The main purpose of IGW is to interconnect Panlab II testbeds with each other and components inside the testbed with the internet (End User “public access”). Technically seen this component is a Border Gateway Function (BGF) and ingress-egress point in each site for the IP communication (media plane) for the signalling (this node can decide when and where the signalling shall be routed). For a given packet, characterized by a 5-tuple (IP source, port number source, IP destination, port number destination, protocol type), it is allowed or not to pass the border gateway. It furthermore acts as:

- dynamically configurable L2 connection/isolation of testbed devices
- dynamically configurable multi-VPN endpoint and link to central VPN concentrator
- dynamically configurable firewall and filter
- dynamically configurable application proxy and/or network address translation

... and therefore it ...

- configures a stateless Virtual Customer Testbed (VCT) between all neighbour IGWs
- enforces exclusivity of a testbed resource to a VCT
- handles dynamic L2 isolation (VLAN based) of the resource and it's connection to IGW
- processes PTM commands to connect a specific resource to a customer VPN
- makes addresses of devices and networks known to the whole testbed interconnection
- enables, if needed, direct L2 tunnelling between testbeds
- sets up the right IP access- and QoS- rules for that specific resource

From the PTM's point of view the IGW is “just another component” in the testbed. There can be one or more IGWs per testbed. PTM utilizes plain SIP to instruct the IGW about creating or deleting connection states inside the testbed and to other testbeds. The IGW server is expected to have several interfaces, one towards the PTM, another towards the internal network of the partner who is offering the resources of the testbed. Further interfaces can be connected to a demilitarized zone (DMZ) or wireless interfaces (e.g. UMTS). A first approach can be depicted in the following picture that uses as an example of the connectivity method the establishment of a VPN:

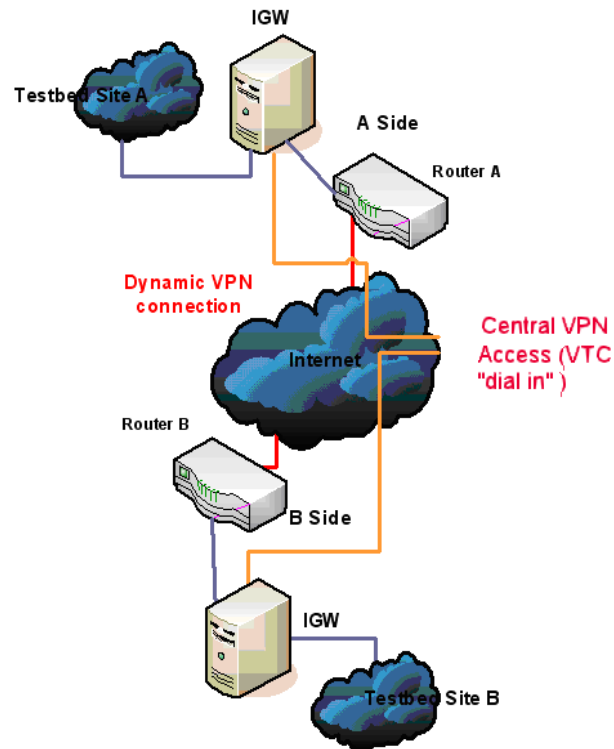


Figure 1: VPN based domain interconnection

The establishment of a VPN up to now has required intervention by specialized network staff for such operations as setting configuration information in a VPN router, setting key information for encrypting the communication between routers, and changing the peer settings at the time of VPN use. As a result, setting up a VPN has been a costly process requiring a certain amount of time from start to finish. Special consideration must be given in the NATing involved at the IGW.

Since Customers should generally only have access to administration tools for their “rented resources” and services he is part of, one VPN (per customer, the customer’s VCT) that connects all these points and protects different customers and the platform against security problems and interference with each other will be established. These functions are dynamically put together and managed by ad-hoc VPN establishment functionality triggered by SIP-based messages.

1.1.2 Virtual Customer Testbeds (VCTs)

Panlab SSA D2.2 Section 4.1 mentioned that “in order to establish connectivity, a Virtual Private Network must be set up **Fehler! Verweisquelle konnte nicht gefunden werden.** hiding the complexities and allowing Panlab partners to dynamically provision multiple overlay networks“.

Customers should generally only have access to administration tools for their “rented resources” and services. To interconnect resources, there are two possibilities that have advantages and disadvantages:

- 1.) Connection of each single lab with a dedicated VPN to one gateway per lab, the gateway/connection point is managing L2 separation of resource (e.g. VPNs)
- 2.) Connection of VPN directly to a resource. A problem is that most of the resources don’t handle VPNs natively so one would need an VPN termination point per resource that could also contain configuration- and/or traffic measurement-functionality

It is very unlikely to terminate a VPN at every kind of lab device so Panlab uses the solution of one VPN per customer (the VCT) connecting all his resources and protects the customer and the platform against security problems and interference with each other.

The per customer's VCT is put together dynamically and managed by IGW ad-hoc VPN establishment functionality, which can be managed by SIP/IMS or other protocols from TEAGLE/PTM. All VCTs are independent VPNs, completely isolated from each other on the same platform. More in-depth description and visualisation of these VCTs can be found in D2.2.

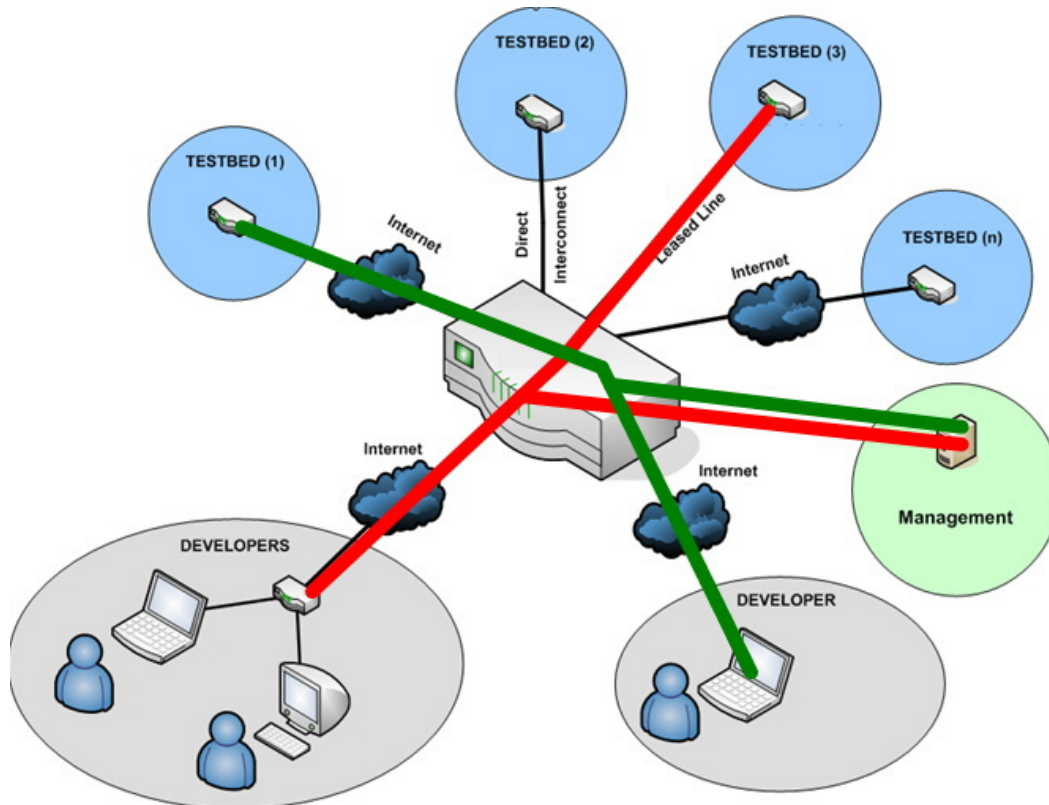


Figure 2: Virtual Customer Testbed constellations

1.2 Functional description of reference points between entities

This section is describing the reference points between the different entities with respect to their functions. A detailed description of how these functions will be realised is to be found in section **Fehler! Verweisquelle konnte nicht gefunden werden.**

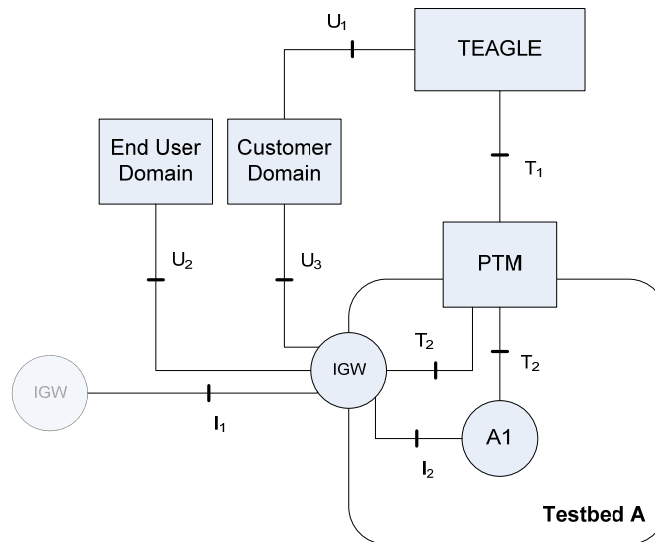


Figure 3: Reference points of PII architecture

1.2.1 The Interconnection Gateway (IGW)

The Interconnection Gateway is the device that interconnects Panlab II testbed with each other and components inside the testbed with the internet (“public access”). For that reason it has to be able to act as a dynamically configurable:

- multi-VPN endpoint
- firewall and filter
- application proxy

From the PTM’s point of view IGW is “just another component” in the testbed. There can be one or more IGW’s per testbed. The PTM utilizes plain SIP to instruct the IGW about creating or deleting connection states inside the testbed and to other testbeds.

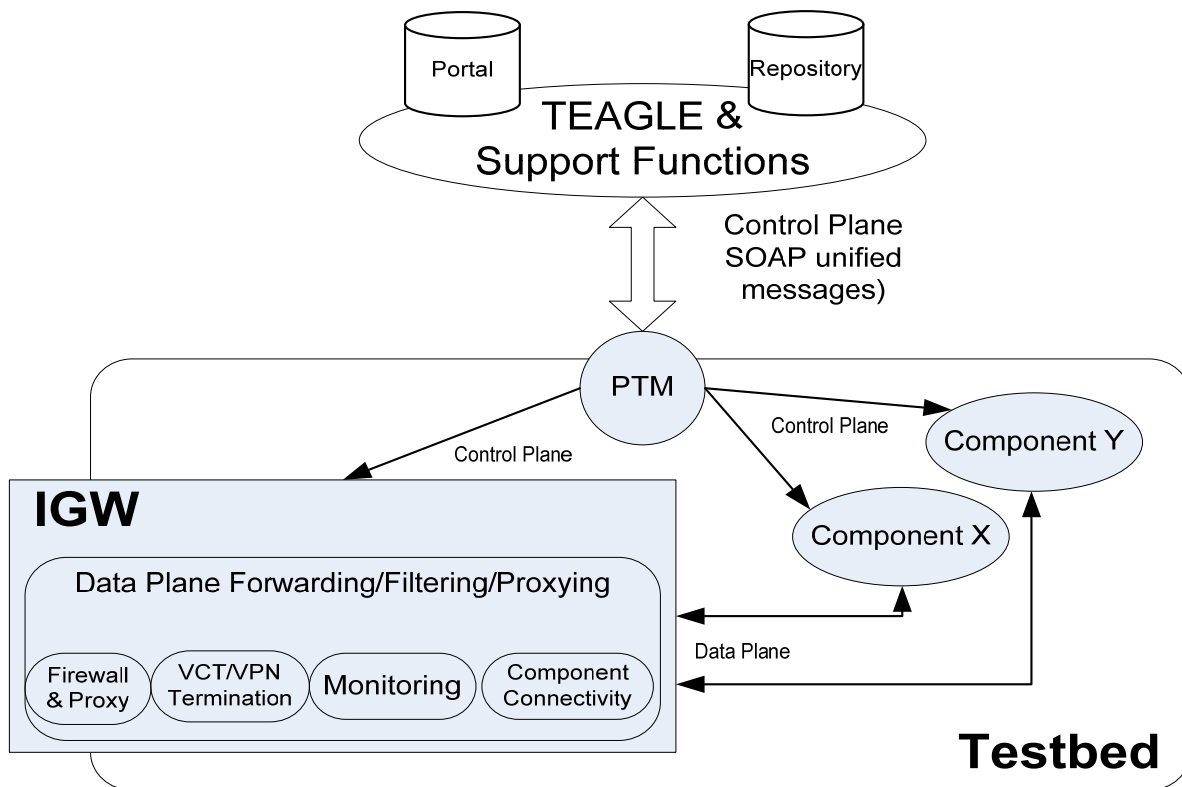


Figure 4: IGW architecture overview

Generally, the following functions and behaviours are required for one VPN per customer, which is dynamically setup by the upper layers of TEAGLE and the PTM:

- the customer can NOT access any other PII resource other than TEAGLE (for configuration purposes) and the ones inside his particular customer VPN
- TEAGLE knows about the name, status, associated resources and connection points of these customer VPNs and initiates/deletes them by communication to the IGWs (over PTMs)
- PTM to IGW communication is done by SIP/IMS
- the customer VPNs stay alive as long as the customer is a valid user of PII
- a central VPN concentrator is optional for the customer VPN and is an entry point from the outside ("dial in) to the customer VPN (data access) or direct control access to the resources
- should the customer wish to make available one or more of his resources to the outside world, this is then a feature that has to be added to the specific component at TEAGLE. If this feature is enabled a connection between the specific component and an external IP/connection pool is switched in a way that only that one resource and not the whole customer VPN is accessible from outside
- each resource bookable by a customer is flagged if direct control/configuration access is allowed by specific protocols (e.g. SNMP, SSH, ...) or not. If not, the IGW will block this access. If yes, the user has absolute access using this specific protocol
- IGWs are responsible for protection and encapsulation of the customer's VPN/resources from any other customer VPN, from the internet or from inside the same testbed
- Customer VPNs - technically - are set up using widely spread VPN technology (IPSec, OpenVPN, for example - to be decided by the IGW Team) and automatically managed by associated IGWs that are instructed by TEAGLE/PTMs

- the customer VPNs are meshed between all IGWs that are responsible for one/more of the customer's resources and the central VPN concentrator
- IGWs terminate the customer VPNs and make sure that inside the testbed no interconnection with other resources other than the rented/bought ones are possible, e.g. by imprisoning all components of a customer VPN into one local Ethernet VLAN
- each IGW has automatic firewalling capabilities that allow only VPN access to the central concentrator, other IGWs and control access to its own PTM

The "public access" feature can be enabled when configuring a resource in TEAGLE. It basically means that a specific service should be accessible from outside, the world wide internet. Of course the additional payment for that should be significant since it should not be the default behaviour to make anything accessible. Making a service public accessible is done by a combination of two things: application proxying (where possible) and automatic filtering/firewalling. Two examples show the behaviour:

- The User has a fancy new system (e.g. mobile web shop) that you want to test in Panlab II and there's a lot going on in the background but at the end of the day the result is presented via a website. What you get is a public IP address that points to a D2 web proxy. This proxy only processes HTTP access to that specific internal server. The proxy itself is just another component in the user's VPN.
- The User may want to use protocols (e.g. VPN) that can't be proxied. In this case the users packets are directly routed from outside to the specific component that requires outside access. The outside connection is done by a router component id "just another component" which is automatically booked when a user sets the "outside flag". For security reasons, only the protocol(s) and IP range that the user specifies are routed.

The IGW "knows" what can be proxied and what cannot be proxied and what has to be firewalled. For the VPN used for testbed-to-testbed firewalling capabilities are also needed since IGW has to secure the testbed anyway.

The IGW will provide status communication updates when either a request is received for such an update or by a notification service triggered when changes occur to the current subject of the connection.

IGW related Excerpts from D2.2

2 Overview of Interoperability Framework in PII

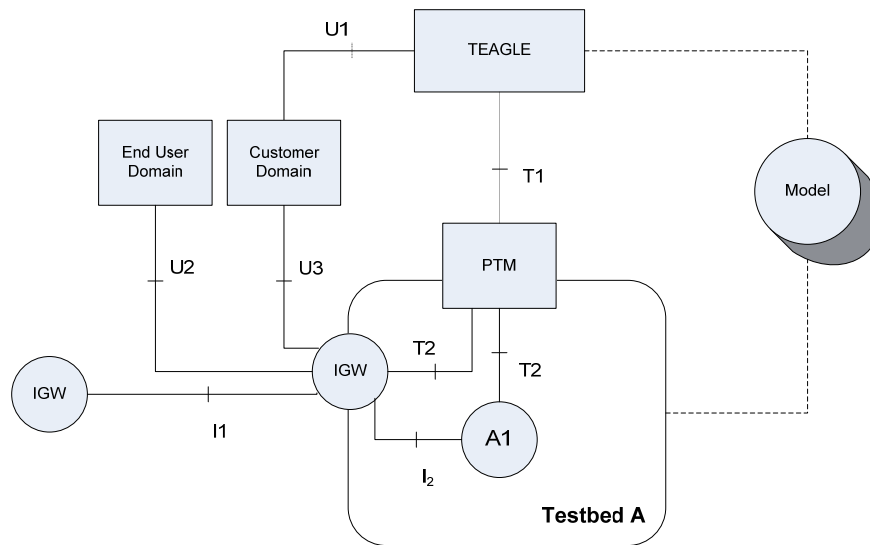


Figure 5: Interoperability in PII architecture

Interoperability emerges across all seven operational stages in various forms and it is mapped onto the PII architecture as defined in D2.1 (see Figure 5).

During customer interaction (stage 1) there must be a well established process and a common “interface” for facilitating interactions between the customer and the Panlab office (Teagle). This “interface” may eventually take the form of a well defined SLA language that must be used in order to unambiguously document the agreements made between the two parties. This is the U1 interface depicted in Figure 1.

In order Teagle to decide whether a customer request can be fulfilled (stage 2) while later to orchestrate the provisioning of the testbed according to the SLA, (stage 3), it must use a “lingua franca” to represent the resources and offerings at large, of the available experimental facilities. This gives rise to the requirement for a model rich in semantics and attributes as well as capable of extending itself when new information and functionality becomes available. Furthermore, this model and the objects it comprises become the parameters to be exchanged between the PII architectural components, e.g. Teagle and PTM, during the relevant operational stages (stages 3-7) through interface T1. As interfaces U1 and T1 as well as the defined model fall within the scope of WP3 their description may be found in the corresponding deliverables of WP3.

During the provisioning of the testbed (stage 3 & 3a) a series of operations must take place which give rise to a number of key architectural interfaces as well as PII specific architectural components that are important not only for the interoperability of the various facilities but also for the overall operation of the provisioned facilities.

Two are the most important PII architectural components, namely, the Interconnection Gateway (IGW) and the Panlab Testbed Manager (PTM) along with their corresponding interoperability interfaces.

More specifically, the interoperability issues that need to be addressed by PTM is the homogeneous treatment of the variety of the configuration interfaces and protocols, proprietary or standard, used by different manufacturer equipments that comprise the experimental facility. This interoperability aspect will be resolved by the resource adaptation layer of the PTM architecture and through the specification of the T2 configuration interfaces. It therefore falls within the scope of Tasks 2.3 & 2.4 and their corresponding deliverables.

In contrast the IGW is the component that provides interconnectivity among experimental facilities that are needed to interconnect on demand as they reside in different locations including customer

premises equipment. Furthermore, IGW components are also subject to configuration and management operations in the same fashion as testbed specific components. To this end, interfaces I1 and I2, T2, and U2 and U3, must be clearly specified. The definition of the IGW architecture and the corresponding interfaces is the main focus of the interoperability framework of Tasks 2.2 and it is described in detail in this deliverable.

Plain interconnectivity is not enough when provisioning and configuring testbeds and components thereof. It is often followed by QoS requirements that must be met by the testbed resources. In the context of QoS, interoperability stems from the fact that individual testbeds are likely to support their own (native) QoS mechanism, namely QoS model, traffic classes and signalling protocol, that may not be applicable outside the specific testbed domain. To this end, there must be a mechanism aiming at identifying the various QoS environments and, in turn, mapping the QoS demands onto the native QoS mechanisms. This issue is addressed in the context of resource provisioning in section **Fehler! Verweisquelle konnte nicht gefunden werden.** which we draw analogies from Next Generation Networks, IMS in particular.

Finally, monitoring and monitoring data manipulation is one of the key services of PII important for supporting a wide spectrum of customer tests as well as providing overall quality assurance. Interoperability emerges from the configuration of monitoring capabilities of the individual components to the collection and transporting of monitoring data to the storage thereof in a common format for reusability purposes. In this context we are looking to adopt and adapt existing engineering practices used in similar environments. One of them is the work by the IETF IPFIX working group that is described later in this deliverable and we propose how it may be used in PI.

Interconnection Gateway

The main purpose of the Interconnection Gateway (IGW) is to handle network setups to separate all communications inside Panlab2 (PII) domain and to communicate with several remote sites. For doing so, there are a number of problems to be solved i.e. to ensure that only specific resources are allowed to communicate to each other or to hide the network topology of a site by applying network addresses and protocol translations.

Quoting from Panlab D2.2¹ Section 4.1 : “*In order to establish connectivity, a Virtual Private Network (VPN) must be set up ... hiding the complexities and allowing Panlab partners to dynamically provision multiple overlay networks*“. The architectural component providing such mechanisms to the PII project is the IGW.

1.3 Introduction and Overview

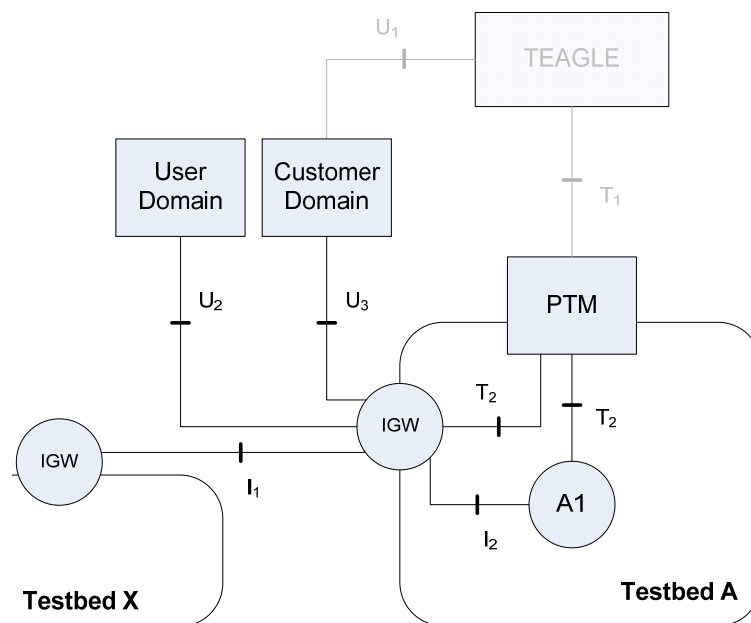


Figure 6: PII testbed environment overview

From the technical viewpoint IGW is the combination of a Border gateway function (BGF) with a Network Access Device (NAD) since it operates as an ingress-egress point at each testbed site while hosting IP communication (data plane) and signalling functionality. Doing so, the IGW interconnects different PII testbeds with each other and also components inside the testbed with the internet (“public access”). Furthermore it implements firewall (IP filtering) and QoS (IP shaping) functionalities for any given packet, characterized by a 5-tuple (IP source, port number source, IP destination, port number destination, protocol type) that passes this border gateway. Furthermore, it may be required to acts as:

- A dynamically configurable L2 connection/isolation of testbed devices
- A dynamically configurable multi-VPN endpoint and central VPN concentrator
- A dynamically configurable IP filter (Firewall) and IP shaper (QoS)
- A dynamically configurable application proxy and/or network address translation

¹ <http://www.panlab.net/fileadmin/documents/Deliverables/Panlab-D2.2-Technical-Infrastructure-V1.0.pdf>

... and therefore it ...

- configures Virtual Customer Testbeds (VCT) among all neighbour IGWs
- enforces exclusivity of a testbed resource to a one VCT
- handles dynamic L2 isolation (VLAN based) of the resource and it's connection to IGW
- processes PTM commands to connect a specific resource a customer VPN
- makes addresses, networks and services known to the whole testbed interconnection
- enables - if needed - direct L2 tunnelling between remote testbeds
- sets up the right access restrictions and QoS rules for that specific VCT
- can establish separated special purpose network areas i.e. demilitarized zones DMZ)
- may connect the testbed to wireless network infrastructures (e.g. UMTS, WLAN, etc.)

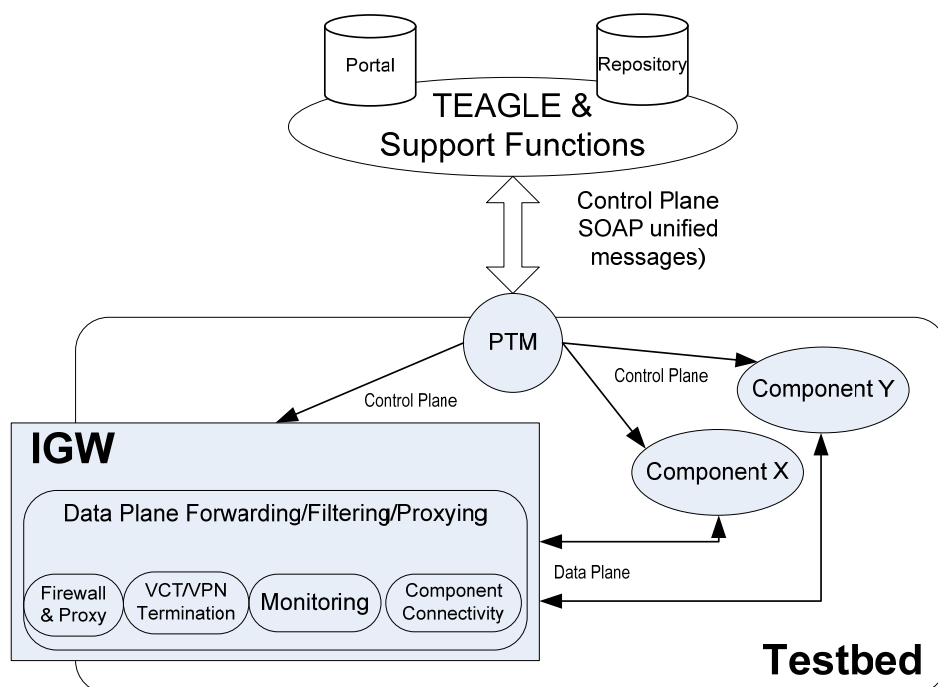


Figure 7: IGW architecture and connection overview

The PII testbed environment overview (Figure 2) shows the separation of command plane (T2) and data plane (I2) communication of testbed components. From the PTM's point of view the IGW is also "just another component" in the testbed and therefore there may be more than one IGW per testbed that may dynamically be deployed too. PTM utilizes the T2 interface to control and configure the IGW for creating or deleting communication states inside the testbed and to other testbeds.

The IGW is expected to have several interfaces, one towards the PTM, another towards the internal network of the partner that is offering resources of the testbed (Figure 3). Of course it is, for different reasons, not common that a PII project partner that contributes network resources is opening up its whole network infrastructure.

For this reason a PII testbed has to be embeddable into a general lab network architecture and separated from the partner's general operation infrastructure. See below an architecture template of a PII partner testbed that fits to the most of today's best-effort based internet related lab infrastructures.

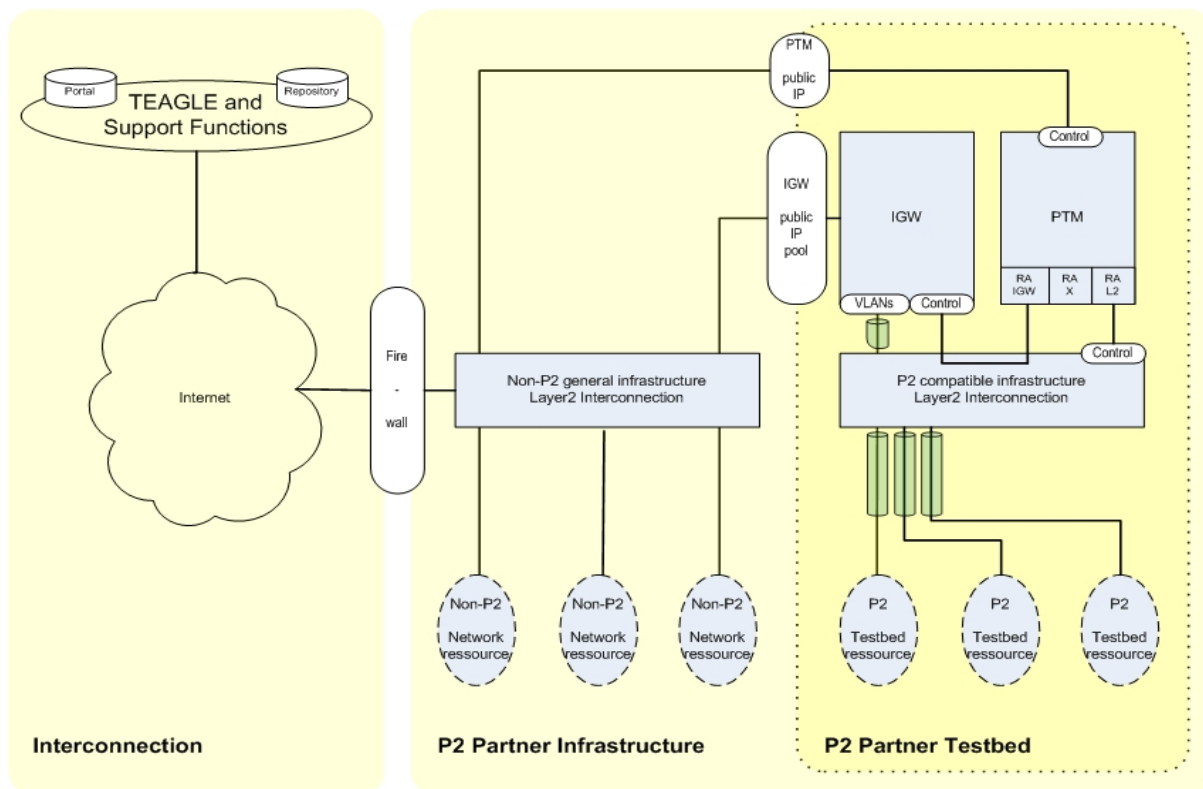


Figure 8: Lab architecture template for PII partner testbed

1.4 The “Default IGW”

A closer view at the PII architecture shows that the operation of a testbed that includes a PTM, it may in principle be possible, since all command plane connections required to run components can be set up without any IGW. Nevertheless, if doing so, there would be no data plane that allows interconnection of components to the customer-/user- domain or to other testbeds and no enforcement of QoS or IP filtering rules. In addition to that, the customer’s VCT would practically not exist since it is maintained by the IGW and is not part of the PTM’s testbed connection.

For that reason it was decided by the Architecture Board (see ABPhoneConf20090210Minutes) that there should be a “Default IGW” existing in every testbed to add the needed data plane and therefore the existence of VCTs. Furthermore the Default IGW will assure a basic best-effort connection from the internet to the testbed components to allow interconnection of customer-/user- domains to VCTs and also optional VPN based interconnections to other testbeds.

The Default IGW is planned to be set up and bootstrapped dynamically by a pre-existing PTM which can access the deployment image of the Default IGW. By using the PTM Resource Adaptors for local Layer2 setup (RA-L2) and computer host control (RA-PC) it is possible to bootstrap such an image and interconnect it on one side to a pool of public IP internet connections and on the other side to the testbed’s VLAN infrastructure. Using the PTM’s Resource Adaptor for IGW communication (RA-IGW) it is possible to completely configure the bootstrapped image for Default IGW operation.

For the initial proof-of-concept implementation of the PII framework, the RA-IGW to IGW communication is done using SNMPv3 since the IGW by definition is a network access- and border gateway- device and therefore may be also supervised using existing network management tools. For later implementations other protocols might be implemented or switched over to these from the initial SNMPv3 based start up.

1.5 Customer testbeds and connection domains

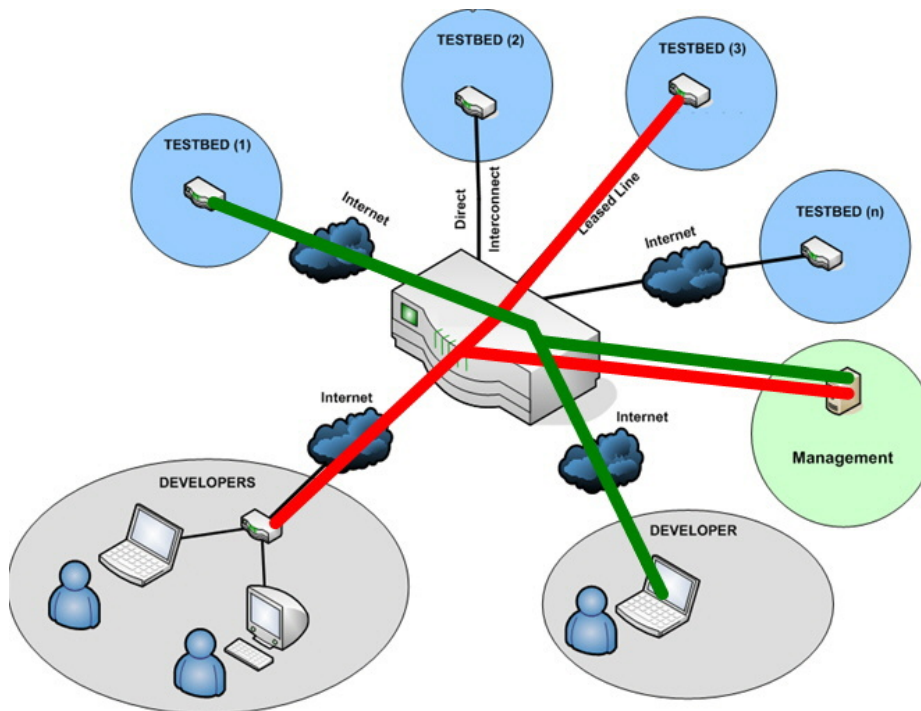


Figure 9: VCT isolation on shared infrastructure

The concept of virtual customer testbeds (VCT) has been introduced to PII define logical separated testbed infrastructures on top and independent from the physical interconnected clusters of the PII partner resources.

Since Customers should generally only have access to administration of his “rented resources“ and services, one separated interconnection layer per customer is needed, the customer’s VCT. It connects all resources of one customer with a common unique IP addressing space and protects different customers and the PII platform itself against security problems and interference with each other.

A VCT is the sum of resources, including interconnections in-between that TEAGLE has assigned to a specific PII customer. VCTs are put together dynamically and managed independently by the IGW’s ad-hoc VPN/VLAN establishment functionalities.

All resources inside a VCT share a common IP addressing scheme and are able to exchange data traffic. The following figure shows an example of logical separation of VCTs (illustrated by tunnel colours) using the same physical infrastructure.

1.6 Internal architecture and Interfaces

This chapter deals with the internal architecture and function blocks of the Interconnection Gateway (IGW). It consists, as shown in the following function block overview, generally of 5 main elements:

- an internal and an external data path filter (firewall)
- termination stage (server) for VPN connection
- interconnection engine of VCT connections
- link layer interconnection, isolation and QoS shaping
- control and communication layer to PTM

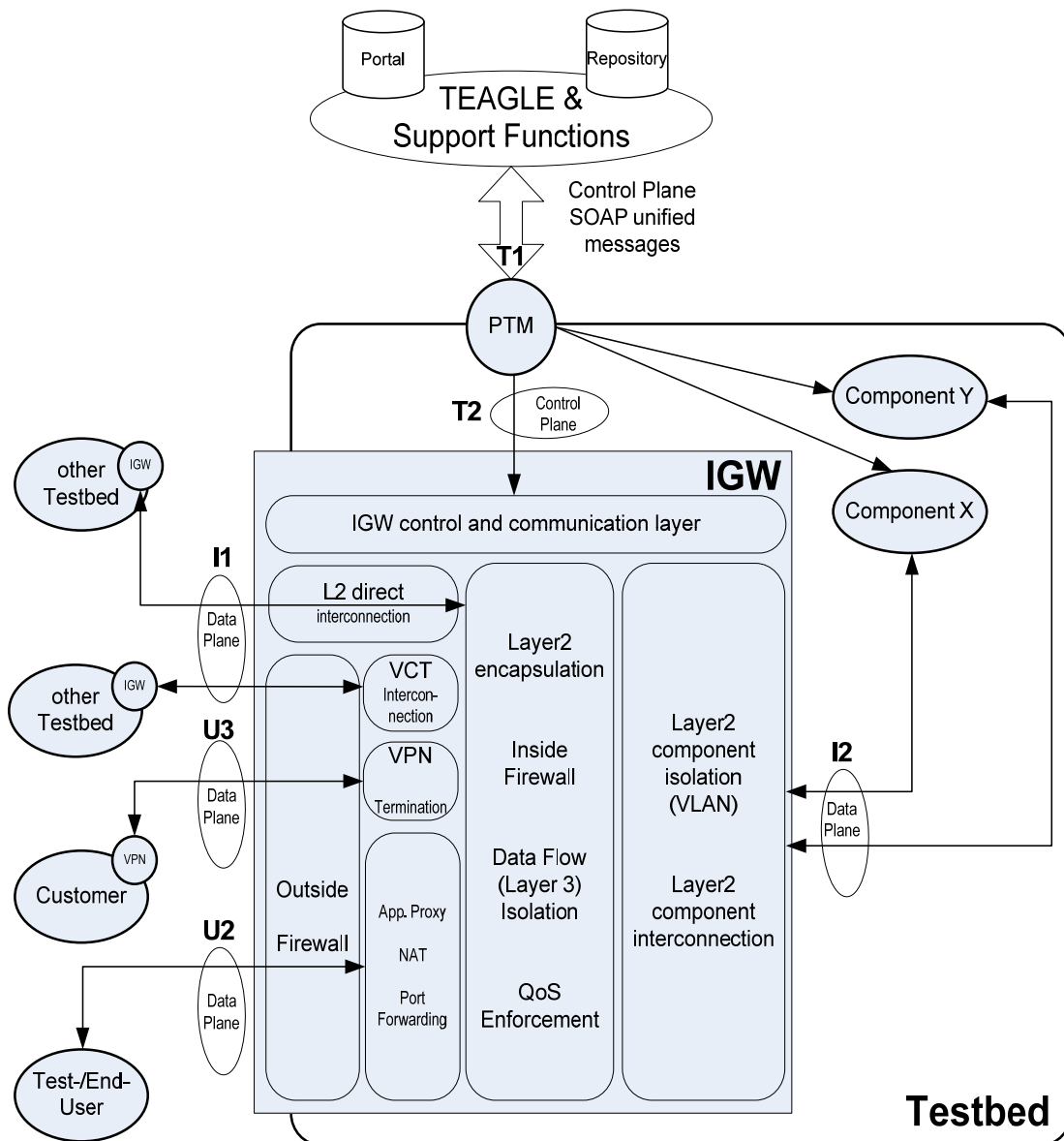


Figure 10 : IGW internal block architecture

The centre of IGW functionalities is to interconnect, keep and protect the mapping of local Layer2 communication channels to external VPN interconnection. It has therefore kind of an IP based trunking function for testbed components communicating on data planes separated by VLANs on the right side and IPsec based VPNs between access domains and other testbeds on the left side. Routing of data plane packets in-between these secure channels is done by the interconnection engine. Furthermore, if ordered by TEAGLE, QoS rules may be enforced on such routing, for instance limiting connection of one VCT to another testbed to a certain maximum throughput rate.

In front and in the back of the interconnection engine the secure channels are being de-encapsulated/decrypted and filtered by a stateful IP packed based firewall. This makes sure that only services and resources may be accessed between testbeds that are allowed so by TEAGLE and that the customer's access is not used for testing nor the test user's access isn't used for configuration purposes.

On the external side of IGW may be also generic Layer2 bridges to other testbeds connected that are not publically accessible and may also be able to perform real QoS reservations such as ATM or fibre optic links. This stage is not firewalled twice (input and output) since it is assumed that an IGW is also available on the other side, so only input firewalling is applied. Technologies and/or devices that

can't be connected or should not be connected directly to the Default IGW can implement their own IGW function and connect to another RA-IGW at the PTM.

The north side of any IGW is the control and communication layer facing towards the PTM's Resource Adapter for the IGW. Communication between IGW and RA-IGW is message based on SNMPv3 and include simple command/reply communication (e.g. for activation of a QoS rule) but also on subscription based event updates (e.g. some security rule was violated). RA-IGW is going to decide which messages are transmitted to TEAGLE and which are just kept internally for statistics- or logging-purposes.

The IP addressing schemes and interconnection parameters (DNS- and NTP- servers, etc) of VCTs are delivered by TEAGLE and announced to all necessary resources by the IGWs. IGWs furthermore take care – if needed - of assigning or relaying dynamic host control parameters and routing information to resources inside a VCT.

1.6.1 Details of component connection and Layer2 isolation (I2 Interface)

The local separation of VCTs on the testbed's Layer2 infrastructure is done by IEEE 802.1q based VLANs. Of course therefore the security of the whole interconnection scheme depends hardly on the implementation of this functionality by the local testbed infrastructure and the command interface of PTM's RA-L2 to it.

The I2 interface between components and IGW is mainly based on numbered VLANs generated by PTM's RA-L2 but interconnected to a data plane by the IGW. PTM itself is not part of such VLAN or VCT at any time

1.6.2 Details of VCT interconnection and Layer3 isolation (I1 Interface)

VCT's are remotely (over "foreign" interconnection) separated by IPSec based VPN tunnels. These are based on pre-shared parameters that were communicated by TEAGLE via the PTMs to the testbed IGWs. The IGW takes care of enumeration of these tunnels and mapping into the right local testbed security architecture.

The I1 interface between testbeds is based on stateless secured separated IP based tunnelling between the Default IGWs of different testbeds. The parameters and vectors for this intercommunication is transmitted by TEAGLE, there is no negotiation or command-based communication between IGWs at any time.

1.6.3 Details of external Layer3 service availability (U2 interface)

Customer are able to allow access to specific resources and services of their VCT from "outside", mainly for testing by external users. These accesses have of course to be hardly restricted on the selected testing purpose and are set up by TEAGLE. Accessing a webserver from outside, for instance, may be needed if the result representation of an internal process is a web site. Consequently only the web site and not all resources that process or collect data represented on that server need to be accessible from outside.

The Default IGW takes care that only the web server machine and its http ports are going to be accessed through a special IP from the public internet. The customer is able to make this IP address available to test users for testing purposes. Protection and enforcement of this access scheme is granted by filtering IP packets and ports as well as including a Layer7 application proxy (http protocol proxy) between the tester and the internal resource. Depending on the service, in some test cases both security mechanisms may work, in other cases only one is usable.

In opposite direction, e.g. an internal resource needs to access external data sources, the same connection scheme is applied reversely. The external site is proxied or accessed by network address translation to protect the internal resource. Since it may be important for the test case which kind of protection is performed, this is selectable by TEAGLE. Because of this the customer is able to test new protocols or connection settings over a variety of border gateway and security mechanisms like NAT.

The U2 interface is used for secured data plane input and output data exchange of a VCT. Therefore this access is limited to the needed resources and services, protecting all other VCT internal entities. Depending on the protocol the different translation- and protection- mechanisms like NAT, port forwarding, proxying and so on are automatically deployed or selected using TEAGLE by the customer.

1.6.4 Details of external Layer3 VPN termination (U3 interface)

For direct configuration or transfer of administrative data between the customer and resources of his VCT is a VPN “dial in” access available to the customer. With this mechanism the customer is practically becoming part of his VCT with a host computer or is able to extend his testbed into his own site, e.g. when using a VPN enabled router to act as a bridge to the customer’s own labs.

In contrast to the U2 interface, the U3 interface is not designed to access VCTs from/to external/public sites and therefore not heavily protected by IP filtering. The IGW will take care that connections between a VCT and a U3 interface are restricted to the customer only and therefore the customer is responsible to restrict access on his end to the VCT as much as possible.

1.6.5 Details of IGW control and communication layer (T2 interface)

Interaction between IGW and PTM is performed on one side by the IGW’s control and communication layer and on the other side by the PTM’s Resource Adapter for the IGW using messages based on the SNMPv3 protocol. Since this signalling is control traffic effecting all VCTs, it is transmitted on the local testbed infrastructure and not inside any VCT.

Messages between IGW and the PTM’s RA-IGW include simple command/reply communication but also subscription based event updates. The RA-IGW is going to decide which messages are transmitted to TEAGLE and which are just kept internally for statistics- or logging-purposes.

1.6.6 Further details and features

In some cases it might be necessary to interconnect two resources across testbeds not only based on Layer3 but also on Layer2, for instance to support local network broadcasting between them. For this reason it is planned to implement on-demand Layer2-in-VCT encapsulation between two IGWs and VLANs. In any case, this should not be the default method of interconnecting resources, it includes a lot of overhead into the payload and since more than one tunnel is used on the actual data plane traffic also lowers the maximum available packet size.

If a customer or operator needs to connect wireless domains of any kind to the physical testbed or to a specific VCT, the IGW is the entity to connect this to. All mentioned filtering and restrictions will be applied to this domain, since wireless networks are handled like an 3rd party sites (please see section 3.6 on this).

1.7 Detailed description of reference points between entities

1.7.1 Reference point U2 - End User Domain to IGW/ Testbed

This interface allows End Users (those actively participating in a given test) to access a PII testbed. The technical details of this interface - with regard to layer 2 technologies, protocols, security constraints, etc – are dynamically defined by the Customer when requesting the establishment of the testing environment to Teagle. Via the PTM, Teagle then configures the IGW to provide and enforce these settings for the U2 interface. In summary, the U2 interface is a test case dependant interface that is provided and enforced by the IGW towards End Users.

For the most common case of using IP connectivity to run the test, this interface must use public, standard IP transport. Therefore, the U2 interface represents the entry point, the “window”, to the public Internet for a specific service provided by testbed resources. These testing resources may be also be pre-established and potentially distributed among several testbeds. A major objective of the U2 interface is to provide an access to the PII infrastructure that does not mandatory impose the support of VPN connectivity, since there may be cases where the terminals/equipments used to run the

test cannot easily support it (e.g. mobile devices, etc). Additionally, this allows End Users to transparently access the PII federated resources without requiring them to implement other mechanisms, protocols or technologies outside those established by the Customer when requesting the setup of the test environment. Thus, the impact on user terminals/equipments for accessing PII is conveniently eliminated.

Despite the use of public IP connectivity, the traffic must be correctly controlled, secured and converted before entering the testbed, i.e. the PII federation. The IGW transparently provides the following functionalities to correctly adapt and control the ingress/egress traffic through the U2 interface:

- User authentication and optional traffic encryption
- Gating: the IGW may need to enforce that only some packet tuples (origin/destination IP address + origin/destination port numbers) are used, as configured for the specific test being executed.
- NAT operations: IP addresses will usually need to be converted from the IP public addressing domain to the IP address space of the local testbed, thus, guarantying correct address interoperability.
- Protocol proxying towards the required testbed resources. This allows End User equipments to interact with testbed resources without knowing the actual testbed topology.
- ALG (Application Level Gateway) operations: in order to insure application-level interoperability between the End User and the testbed resources, the IGW may need to also modify some protocol details, such as connectivity information, protocol headers, etc.

Depending on the planned test, the U2 provides connectivity for the end user's terminal. It opens a protected window to a VCT that includes the resources that need to be tested. For example, the resource to be tested is able to offers an HTML based user interface, in this case the IWG provides access to this interface for the following purposes:

- allows the end user to see testing purpose/tasks/instructions provided by the customer
- allows the end user to download needed testing client software
- allows the end user to contact support
- allows the end user to contact the Customer
- allows the end user to report observations and opinions about the test
- allows the end user to report starting and ending the test

1.7.2 Reference point U3 - Costumer Domain to IGW/ Testbed

The U3 interface grants access from the customer's machine or lab network to his Panlab virtual customer testbed (VCT) by the IGW. This connection utilizes standard secure "dial in" technologies, such like IPSec, PPTP, L2TP. IGW cares about access imitations to the VCT associated resources only that were set up by the customer using TEAGLE. The customer utilizes a standard VPN client or VPN-aware router to directly connect to an VPN concentrator (usually an IGW chosen by TEAGLE) to become part of his VCT. Once interconnected the customer is able to address all his "rented" resources including configuration front-ends.

Generally, it can be said that this involves several components from both the federation and the user domain than the setup of a VPN that incorporates federation and user domain equipment within the same VPN is likely to be more secure and efficient.

After connecting to the VCT using interface U3, the customer is able to address all resources of his VCT by using the VCT's addressing scheme.

1.7.3 Reference point I1 - IGW to IGW

IGW-to-IGW interconnection is provided through the I1, basically a data path between two testbeds. Communication between IGWs is using the data plane only, there is no control message exchange since the used secure tunnelling methods are stateless and using pre-shared secrets. Routing between testbed components of different testbeds is done by the IGW using intelligent routing protocols like OSPF and topology information – including VPN setup parameters - transmitted by TEAGLE through PTM.

Using I1 interface by all IGWs form a meshed testbed interconnection network (the so-called Panlab connectivity domain) is dynamically created without exchange or stateful control connections between IGWs.

1.7.4 Reference point I2 - IGW to Testbed Components

Interconnection between the local IGW and internal Testbed Components is provided by the I2 interface, main purpose is to enable an isolated (e.g. by VLAN) link layer access to all associated testbed components. It furthermore can provide QoS based traffic shaping for all communications that is routed from the testbed component through the IGW. This communication is data plane payload only and generally configured by PTM.

To archive a secure a secure and isolated connection between IGW and testbed components, the components need to be directly connected to the IGW or link layer connected by a switching device that honours link layer isolation methods like IEEE 802.1q or TDM. As long as the associated testbed component is assigned to a VCT, the associated IGW needs to have exclusive link layer access for reliability and security reasons.

1.8 3rd party sites

3rd party sites are external partners, companies or other organisation who are connected to one of the federated testbeds via fixed connections and are willing to offer resources to the testbed federation but not interested on hosting PTM and Default IWG in own premises.

These 3rd party sites are connected to the federation via an IGW of an existing partner testbed. The existing testbed creates agreements for hosting such testing resource and creates the needed resource descriptions and manages a service by its own PTM. Furthermore the existing partner takes care of and is responsible for security problems related to the hosting of an external partner.

This way the resource offerings in the federation can be extended also by the participants.

IGW related Excerpts from D5.1

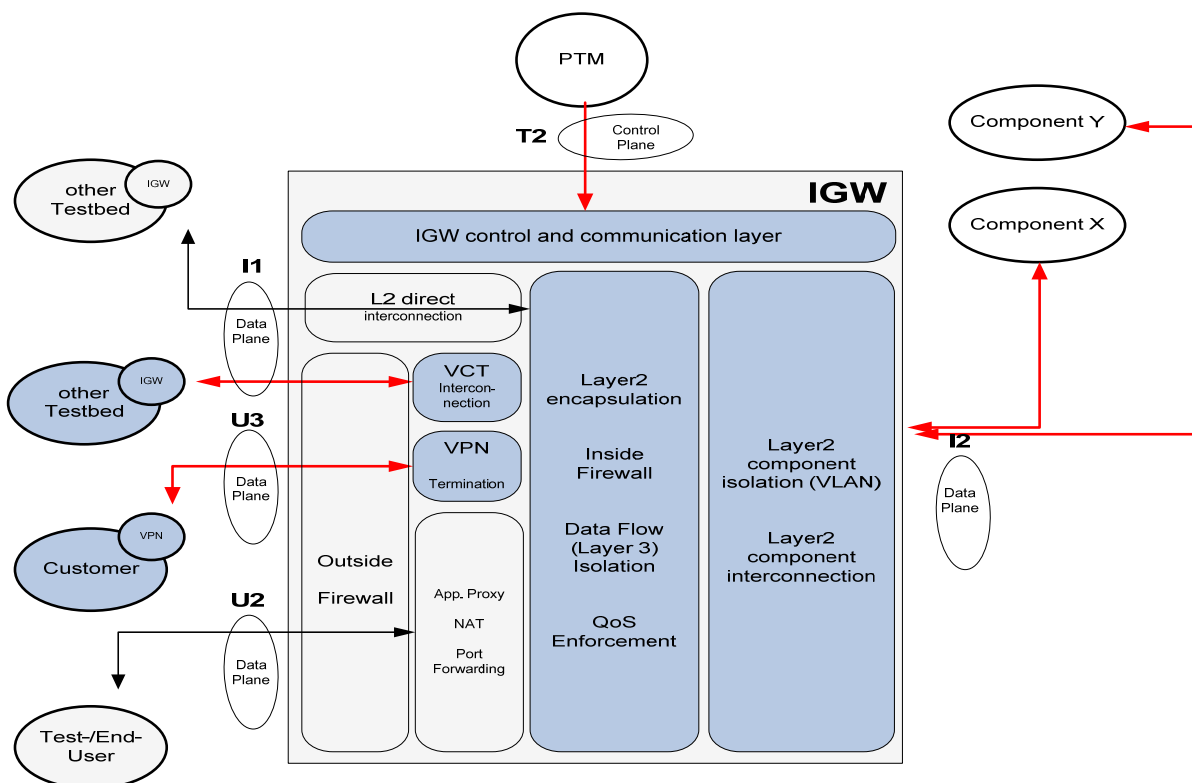
IGW has been implemented by DT for an rpm-based GNU/Linux machine. It currently uses main functionalities of the Linux kernel v2.6 like IPIP tunnelling, packet forwarding, packet filtering, QoS packet queuing, encryption and the like. All parts are managed by an IGW network daemon.

The current release is a connection state machine that interconnects per state 4 network points : an internal VLAN and the public IP address of one IGW to the public address of another IGW including another specific backend VLAN. Such interconnection of VLANs is triggered by TEAGLE using the IGW's Resscource Adapter of the PTM using XML command packets.

Each interconnection state can be expanded by adding more interconnections to one of the attending IGWs. New interconnection states do not interfere with existing ones, they use the same IPIP tunnel but are separated during the routing and filtering process. This guarantees an on-demand automatic IGW-to-IGW meshing of all test sites with stateless low overhead tunnelling without using proprietary inter-IGW protocols.

The interconnection of VLANs is transparent for the customer and uses testbed based IGWs. If the customer wishes to connect own test sites or single devices to his VCT it is possible to access it with the "customer dial in" feature. This L2TP based on-demand tunnel delivers direct access to a specific VCT as if the customer itself had a local IGW.

The following sub-components (marked blue) and interfaces (red lines) have been used up to this point for review demonstration purposes. The next demonstrator is planned to be extended to also use the remaining interfaces and building blocks by adding further scenario details.



For the upcoming release planning are to package the IGW network daemon installable with a pre-packaged reduced Linux distribution. Advantages of this in comparison to a pre-configured virtual machine are the possibilities of using physical machines or other virtualisation techniques that the current one. Furthermore several tunnelling protocols like GRE, SIT or IPSec tunnels are going to be supported.

IGW related Excerpts from D5.2

Testbed and service connectivity scenarios

This chapter talks about static and dynamic networking connectivity scenarios for connecting resources both internally and between testbeds in Panlab federation. The scenarios depict networking from idle testbed when resources are not used to provisioned testbed when one or multiple resources are in use as part of VCT.

The functionality is provided by the Panlab connectivity management framework, which is a mandatory element in any Panlab compliant test site. Its goals are threefold: a) connect resources assigned to a VCT in a virtual network in an automated manner, b) ensure the security and privacy requirements of test sessions executed in parallel by providing an efficient separation of different customer VCTs, c) allow test site providers deployment of multiple connectivity domains using arbitrary layer-2 technologies.

1.1 Resource connectivity in testbed

This section provides a brief high-level overview of the steps required to attach an existing test site or a newly created one to the Panlab federation. One goal of the Panlab architecture is to keep the entrance threshold for attaching new testing systems and services low. This section provides some guidelines for administrators and explains limitations of the model adopted by Panlab for customers and providers.

While the general structure of a PII compliant testbed is in fact determined by the intended service offering, the Panlab architecture consists of two mandatory elements: the Panlab Testbed Manager (PTM) controls resource configurations via a dedicated management network segment and the Interconnection Gateway (IGW) encapsulates VPN functionality in order to establish connectivity with other test sites. Both PTM and IGW functionality may be encapsulated in a single host; or if performance requirements must be met or different interconnecting transport technologies have to be integrated, several IGW instances may be deployed in a test site. PTM and IGW may be installed on dedicated physical servers, though this is not a mandatory requirement. Their functionality is available in packages for installation in already existing hosts or as pre-configured images (XEN **Fehler! Verweisquelle konnte nicht gefunden werden.** or VMware **Fehler! Verweisquelle konnte nicht gefunden werden.**) for direct deployment. A PTM and an IGW deployed on a single physical server define the smallest valid nucleus of a PII compliant testbed.

Besides PTM and IGW as core elements, a test site may comprise additional physical servers for hosting services or providing virtualization or dedicated testing equipment like radio base stations, protocol testers, network equipment (routers, switches) under test, etc. For connecting such physical instances, a connectivity domain is required. While Panlab was designed to support arbitrary connectivity domains, support for IEEE 802.3 Ethernet based systems has been added to the set of core resource adapters due to the wide availability of Ethernet hardware. Note that for isolation of testing sessions IEEE 802.1q support (virtual LANs) is a mandatory requirement and a prerequisite for conducting different test sessions in parallel. If absent, no shared access to the test site will be available.

In the Panlab architecture resource adapters control operation and configuration of individual resources and the core functional elements PTM, IGW, and Ethernet connectivity domain make no exception from this rule. The Panlab software distribution comprises a set of pre-installed core resource adapters and an Ethernet connectivity domain. However, these resource adapters require an initial configuration as explained in Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**

A test site administrator may add additional resources incrementally to a PII compliant test site, thus with the basic set of elements and adding other entities later is a good strategy. The next sections cover the two main building blocks in more detail: adding virtualization services as well as the initial connectivity domain.

1.1.1 Test Session Isolation and Panlab Control Plane

The core connectivity domain adopted for the Panlab architecture is based on IEEE 802.3 Ethernet and the system uses IEEE 802.1q virtual LANs for isolation of different testing sessions and network management functions for configuring testbed resources. While this approach is feasible for virtual network topologies with arbitrary layer 3 protocols and topologies, testing sessions requiring access to layers beyond layer 3 cannot be realized. When a test site owner wants to offer systems for testing on layer 2 or the physical layer, the Ethernet connectivity domain may be used for configuring the deployed endpoints (switches, optical cross connects, ingress devices for GMPLS, etc.) and transfer user traffic to and from these endpoints.

Virtual layer 2 links based on IEEE 802.1q may span multiple test sites or as a shared medium connect multiple resources in a single test site. A customer has full control over the network topology to be deployed, thus a virtualized host resource may act as a software router within a VCT. However, this flexibility comes with a significant complexity in configuring VCT's in the network layer. Panlab does not support automated checking of mis-configured devices and thus, cannot detect any problems arising due to invalid network configurations.

Figure 11 depicts the relevant elements for connecting virtual hosting resources via virtual LANs. In this example three virtual network segments are shown:

1. **VLAN #48** defines network segment #1 and connects virtual bridges *dom0(A):virbr1* and *dom0(B):virbr3*, so that the virtual machines *dom0(A):[vm0, vm4]* and *dom0(B):[vm3,vm5]* are all members of the same Ethernet segment.
2. **VLAN #404** defines network segment #2 and connects virtual bridges *dom0(A):virbr6* and *dom0(B):virbr7*, so that the virtual machines *dom0(A):[vm5]* and *dom0(B):[vm1]* are all members of the same Ethernet segment. Furthermore, the *IGW* is connected to *dom0(B):virbr7* and provides connectivity to other network segment's deployed in remote test sites.
3. **VLAN #10** is a specific network segment that is used as management network segment comprising the PTM as the gateway to PII higher layer entities (Teagle) and all management functions of dom0 hosts and physical servers.

VCT virtual links

Thus, a VCT virtual link is mapped on an IEEE 802.1q virtual link and is capable of connecting virtual host resources and physical systems attached to the Ethernet switching domain. The creation of a virtual link is twofold: a virtual LAN is created dynamically (or a manually pre-configured one is assigned to the VCT virtual link) by the Ethernet connectivity domain RA for carrying user traffic among all resources attached to this VCT link. For physical components directly attached to the virtual link, the responsible port on the Ethernet switching domain is added to the VLAN in untagged mode and for virtualized server resources running on a dom0 instance, the VLAN is added to the relevant port for the hosting dom0 in tagged mode. On the dom0 instance, the required steps for decapsulating the tagged VLAN and the setup of an appropriate virtual bridging device is done with the help of the virtualization RA.

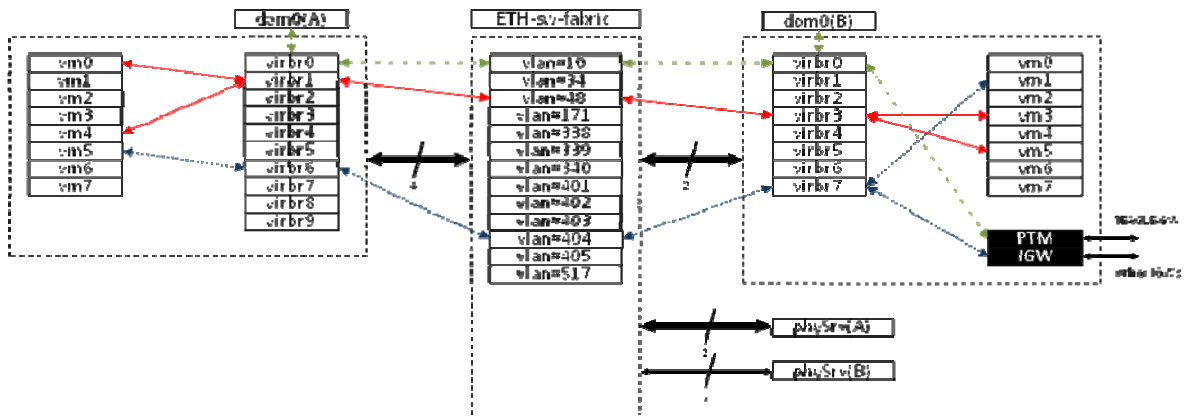


Figure 11: A typical PII testbed structure

While the mapping is done automatically by the Ethernet and virtualization RAs, Panlab provides access to all mapping information to the test site administrator in case that automated management fails and resources have to be released manually.

Note that the Panlab Ethernet connectivity domain RA is capable of blocking specific VLAN ranges and to limit itself on a restricted block of VLAN identifiers. Thus, an existing switching domain may be used in parallel for Panlab as well as other purposes, so no dedicated hardware has to be provided for a PII compliant test site and conflicts with already deployed VLANs should be minimized.

Dynamic vs. static management of resources

Virtual LANs and network setup on the hosting dom0 machines is done automatically by the PII core resource adapters, i.e. a virtual bridge is created dynamically on each dom0 for a new virtual link upon request and connected to a VLAN created with the chosen VLAN identifier. Virtual bridges and virtual LANs are in fact dynamic resources and the core resource adapters provided by the PII framework are capable of handling these resources dynamically, i.e. creation and removal are done automatically during deployment and removal periods of a VCT. However, when the automated RA mechanisms cannot be applied in a specific environment, the test site's administrator may create appropriate virtual bridges and VLANs manually and register these now static resources with the appropriate RAs which will assign these resources like dynamic resources to a virtual link.

Trunking

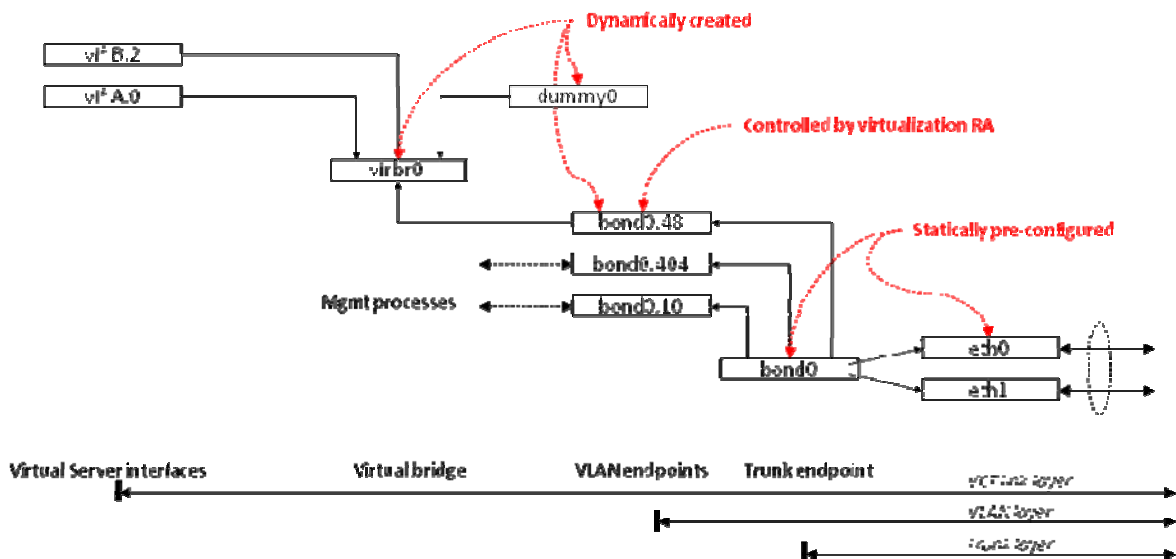


Figure 12: network setup on a XEN dom0

Figure 12 depicts a typical networking setup on a XEN dom0 host: on the right hand side two physical interfaces *eth0* and *eth1* are bound to a single trunk with interface *bond0* acting as endpoint. The trunking protocol used depends on the adopted hardware and thus, on the trunking implementations provided by the switch manufacturer, e.g. Link Aggregation Control Protocol (LACP) according to IEEE 802.3ad, Cisco EtherChannel, or Linux bonding. The test site administrator may choose whatever protocol is available in the local setup. However, when trunking is used, the switching devices may or may not support dynamic adding of tagged VLANs to trunks. In the latter case, when e.g. adding a new tagged VLAN requires unbundling of the trunk, dynamic allocation of VLANs cannot be used. Unbundling may disturb other VCT sessions currently running in parallel. In such cases, either all VLANs must be pre-configured in the switching domain including a registration as static resource with the Ethernet connectivity domain RA or trunking cannot be used at all.

VLAN endpoint management and XEN virtual bridges

The virtualization RA creates dynamically virtual LAN endpoints on the hosting server: in the above example interfaces *bond0.48*, *bond0.404*, and *bond0.10* are created respectively. In order to avoid any interference with XEN provided scripts, the *bond0* interface should not be bound directly to a bridging device. In the recent past the XEN networking code has suffered considerable changes in version 3, 3.1, 3.2, 3.3 and 3.4, so further changes may occur in the future. Nevertheless, we recommend separating management of physical interfaces and virtual bridges and to couple all virtual bridges initially with a dummy interface. In the example, *dummy0* has been coupled with *virbr0*. The virtualization resource adapter creates a new bridging device automatically and once the VLAN endpoint has been created adds this endpoint to the bridge. When the virtual server resource is started, its logical interfaces are also connected to the bridge device.

DomU (and IGW) constraints

In the XEN framework, the number of virtual network interfaces that can be assigned to a virtualized server (domU) is typically limited; in version 3.1 the upper threshold was three network interfaces, later increased to eight interfaces in version 3.2. However, for customer defined host resources and especially for virtualized IGW instances this implies a concrete restriction on virtual network topologies that may be created and in the case of the IGW, the number of VCT testing sessions that can be executed in parallel is limited to eight. Support for carrying tagged VLAN traffic into a domU instance is currently missing in XEN. Furthermore, stable support for adding network interfaces dynamically is missing in XEN. From these restrictions, the following recommendation for setting up a virtualized IGW can be given: add the maximum number of allowed virtual network interfaces to the IGW domU. One network interface (by convention *eth0*) will be used for connecting to the test site's management network; all residual interfaces may be assigned dynamically to VCT testing sessions.²

1.1.2 PII Test Site Management Network Segment

For configuring and controlling resources, a dedicated management virtual LAN must be created. The PTM default configuration comprises a preconfigured DHCP server that provides IP addresses from a pre-defined address space to all resources. Note that, once control is handed to the customer, resources will be disconnected from this management network in order to prevent customers from interfering with other resources and VCTs being active in parallel via the control plane.

The PII Test Site Management network segment is used to allow a Panlab Testbed Management control over resources deployed in a test site. No testing session traffic is allowed within this special virtual LAN. It is the administrator's task to define appropriate layer 3 connectivity among all management endpoints in the PII management VLAN by assigning IP addresses.

The administrator must configure the virtual bridges on the dom0 hosts and the used VLAN manually in order to create the management VLAN. The management virtual bridges must not be registered in the PTM for use by testing sessions.

PII provides a dom0 resource adapter for controlling setup of individual virtual machines as well as the necessary network configuration on the hosting dom0.

² Note that is may change in a future version.

Physical servers are handled differently compared to dom0 entities: a physical server must be connected during configuration periods to the management VLAN for preparing a new testing session. However, such a server must be disconnected from the management VLAN, once control is handed over to the customer in order to prevent the customer to send traffic towards other entities connected to the management network or to gather any information from testing sessions being active in parallel via the management VLAN. Thus, there may be no control over the resource from the PTM during an active testing session. Reestablishing connectivity to the management VLAN involves separating the physical server from the testing session VLAN and readding it to the management VLAN and thus interrupts operation of the physical server device.

1.1.3 PII Endpoint management

A Panlab testing environment consists of two coupled domains: the PII controlled domain comprises all elements deployed and controlled by the Panlab framework, the customer domain comprises all functional elements under control of the customer. The Interconnection Gateways provides adequate means to attach both domains to each other by allowing the export of customer endpoints. Such endpoints secure traffic between the customer domain and the PII domain by using virtual private networks.

1.2 Testbed to testbed connectivity in VCT

IGWs are border gateways of partner's testbed and are able to automatically establish connections to their peer IGWs of other testbeds. They are self-configuring, except the definition of external public IP communications parameters. The following screens show how to install and boot an IGW for the first time and make sure it operates properly.

The IGW's core functionalities are based on Linux Kernel and implemented for Linux RPM-based (e.g. RedHat, CentOS, SuSe) machines. Currently it is delivered inside a virtual image, but upcoming releases are planned to make available a distribution-like installation disk.

Main purpose of an IGW is to act as Panlab testbed border gateways for automatic IGW-to-IGW meshing of all Panlab test sites. For such meshing of all IGWs a stateless low overhead tunneling was chosen, without usage of proprietary inter-IGW protocols.

Site internal 802.1q VLANs keep up the separation of VCTs that are routed site external through IPIP, GRE, SIT or IPSec tunnels. Furthermore an XML-based control signaling and monitoring protocol is used to control the IGW by the PTM or configuration tool using T2 interface.

```

<?xml version="1.0" encoding="UTF-8" ?>
<PTM2IGW>
  <Command>
    <Action> create
    <SubAction> VPN
    <Type> IPIP
    <SubType> IGW2 IGW
    <ID> 1609
  </Command>
  <Local>
    <Type> IPv4
    <PublicIPAddress> 141.39.79.116
    <PublicNetwork> 141.39.79.114
    <PublicNetmask> 255.255.255.224
    <PublicIPGateway> 141.39.79.115
    <InternalIPAddress> 192.168.123.12
  </InternalIPAddress>
    <InternalNetwork> 192.168.123.0
    <InternalNetmask> 255.255.255.224
    <InternalIPGateway> 192.168.123.1
  </Local>
  <Remote>
    <Type> IPv4
    <PublicIPAddress> 141.39.79.116
    <PublicNetwork> 141.39.79.114
    <PublicNetmask> 255.255.255.224
    <PublicIPGateway> 141.39.79.115
    <InternalIPAddress> 192.168.123.12
    <InternalNetwork> 192.168.123.0
    <InternalNetmask> 255.255.255.224
    <InternalIPGateway> 192.168.123.1
  </Remote>
  <Credentials>
    <PSKey> B084D169B084D1
    <Username> IGW
    <Password> secret
  </Credentials>
</PTM2IGW>

```

Figure 13. Network related commands and parameters in IGW.

Figure 13 shows the create command from a PTM to the IGW, including a command section, local and remote IP address parameters and a credentials section for authorisation. Each IGW is capable of

- Auto detection of network interface activity
- Auto detection of DHCP (Dynamic Host Configuration Protocol) availability
- Automatic connection test for available tunneling
- Automatic VLAN configuration test
- Automatic communication test to PTM
- Automatic detection of remote IGWs
- Automatic configuration results correctable by hand

The following preparations are necessary to set up an IGW in the local partner's testbed.

- Connect IGW host machine to public internet
- Connect IGW host machine to internal Layer2 testbed switch
- Get the IGW image and start it, make sure all physical network interfaces are connected to all virtual ones
- Be prepared to solve firewall problems for tunneling protocols (e.g. according RFC1853) to other IGWs
- Start the configuration and monitoring tool to verify the state of operation

1.3 Setting up an IWG

Reserve a new machine using VMware Workstation or Server and make sure this machine is guaranteed at least 256MB of RAM. Bridge the virtual "Network Adapter 1" Interface (sometimes just called "Network Adapter") directly to the host system's physical public interface and the virtual "Network Adapter 2" Interface directly to the host system's physical testbed internal interface.

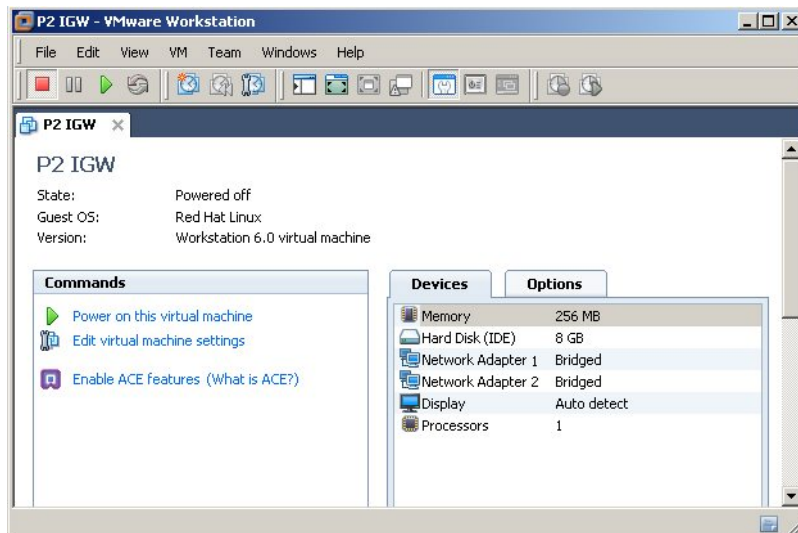


Figure 14. IGW machine in VMware

After that, just boot the virtual machine and all necessary services will be started automatically. At the first boot, or if no meaningful configuration exists, the IGW will prompt an input window for external communication parameters. This is important since IP address, network mask, IP default gateway and domain name system parameters can not be auto-configured. When prompted, choose “Edit Devices”.

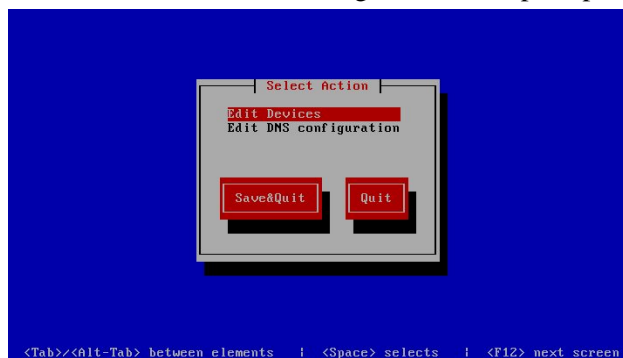


Figure 15. Configuration of IGW at the first time.

Do only select and modify the eth0 interface. The eth1 interface needs to be present but untouched.

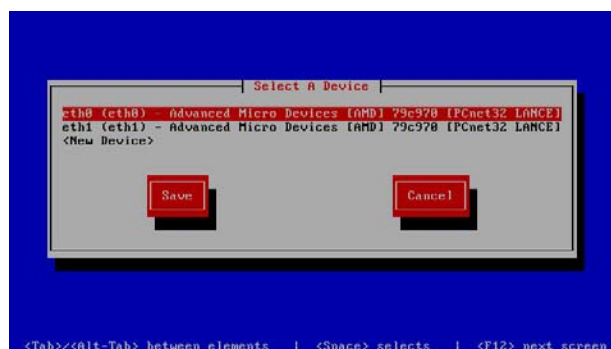


Figure 16. Setting up networking devices.

Insert the usual public IP parameters. The IGW’s external interface is not allowed to use private IP parameters, DHCP configuration or to operate behind a NATing network access device. Furthermore make sure that your firewall allow to the IGW incoming and outgoing the following ports:

- RFC4251 - SSH (TCP, port 22)
- RFC2661 - L2TP (UDP, port 1701)
- RFC2003 - IPIP tunneling (“next level protocol” 4, according to RFC790)

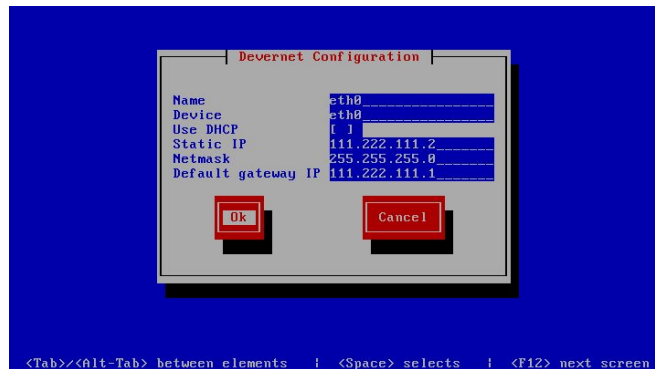


Figure 17. IGW Network setup

After modifying eth0 parameters, save the changes and go back to main menu. As a last step enter the DNS configuration menu.

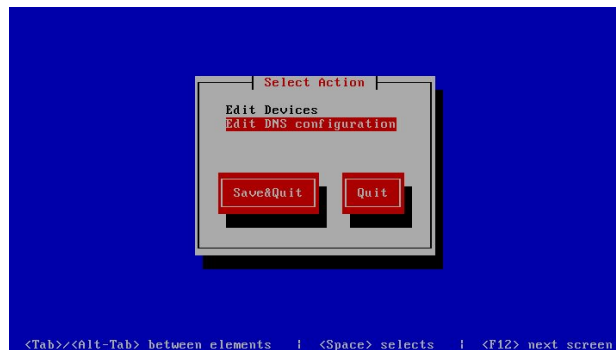


Figure 18. DNS configuration.

Enter the partner name (e.g. EiCT, UoP, TSI, etc.) into the hostname field and up to three existing domain name servers of your choice. Proceed with Ok and go back to main menu.

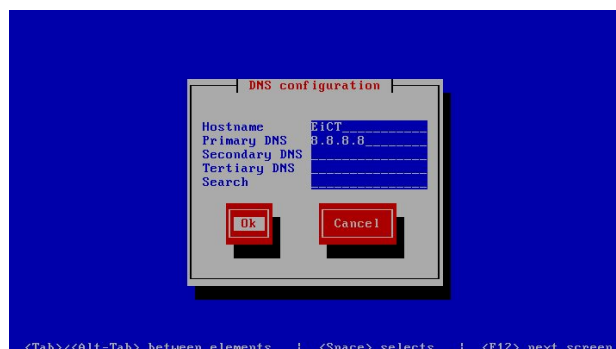


Figure 19. IGW DNS setup.

This procedure is only applied the first time the IGW boots. Thereafter the IGW will try to connect and mesh with other available IGWs as well as the local PTM. The boot process ends showing a configuration and monitoring summary of the internal and external interfaces to overview IGW’s connectivity behavior. See below the automatic configured fields (green) and the ones that can be chosen by and (red and yellow).

```

IGW T2 configuration and monitoring tool v0.79

name      active  VLAN  mode          address          netmask          router
-----
eth0      yes    023   <v4 static>   141.049.016.002 255.255.255.000 141.049.016.001
eth1      yes    042   <v4 DHCP >   010.001.001.002 255.000.000.000 010.001.001.001
eth2      no     -
eth3      no     -

name      resource connection test  mode  address          alive  meshed  list name
-----
eth0      found remote IGW        IP/IP  080.075.105.172  yes   yes    IGW Octo
eth0      found remote IGW        IP/IP  150.140.184.239  yes   yes    IGW UoP
eth1      found remote PTM        local  010.001.001.100  yes   -      PTM BiCT

```

Figure 20. Successful IGW boot.

1.4 User connectivity to the virtual customer testbed

This chapter presents a scenario called “VCT access” in which the customer can connect a terminal or small infrastructure with the created VCT and be part of it. Technically the tunneling protocol L2TP was used to ensure remote access on ISO/OSI layer 2 and upwards. This type of connection should be used if no local IGW connection is available.

See below an installation example for a single Microsoft Windows terminal. Make sure to open UDP Port 1701 on your firewall! Open start menu of the taskbar and select to run the “regedit” tool.

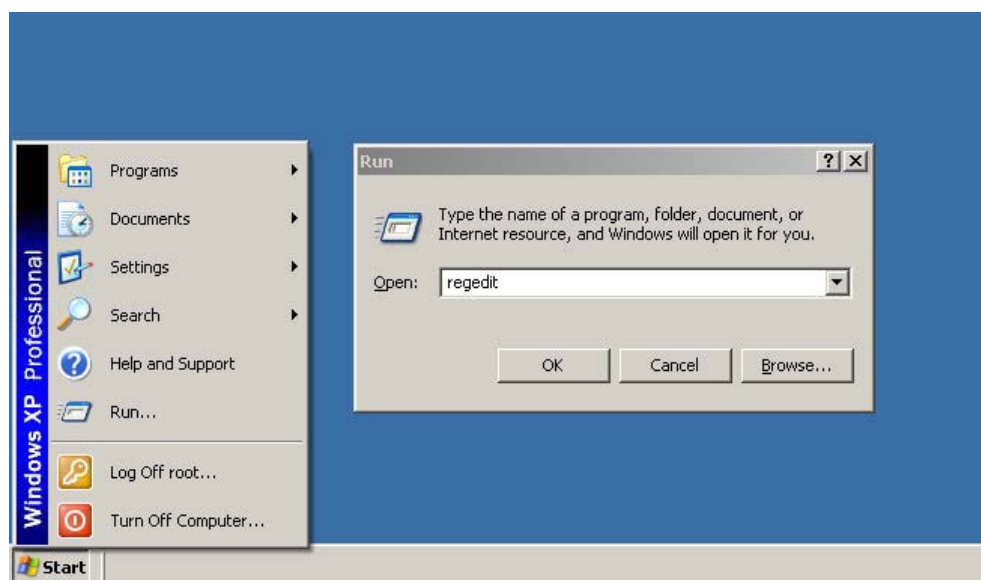


Figure 21. Starting regedit tool.

Find the “Parameters” entry of the tree HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan

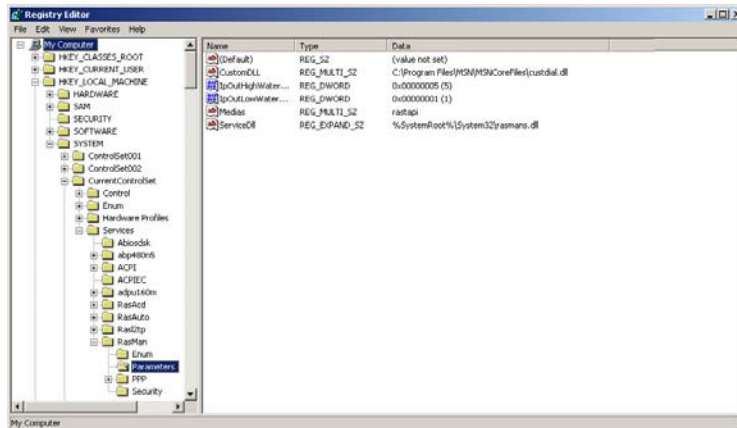


Figure 22. Remote Access Entry in the registry.

Insert a new DWORD named “ProhibitIpSec” and change it’s value to “0x00000001”

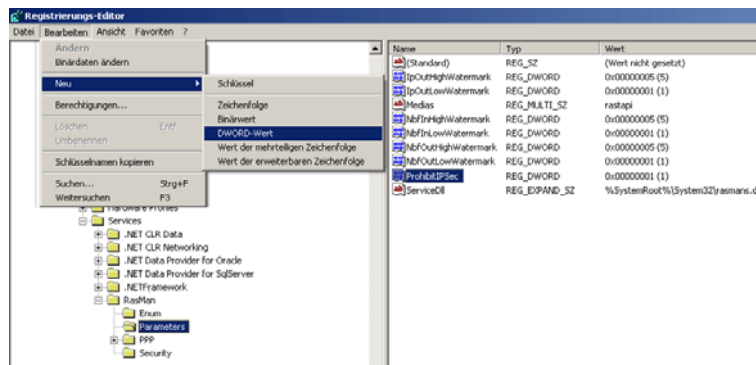


Figure 23. Insert registry configuration.

Close the regedit tool and make sure the value was saved into the registry by re-opening it to have a look if the “ProhibitIpSec” parameter exist.

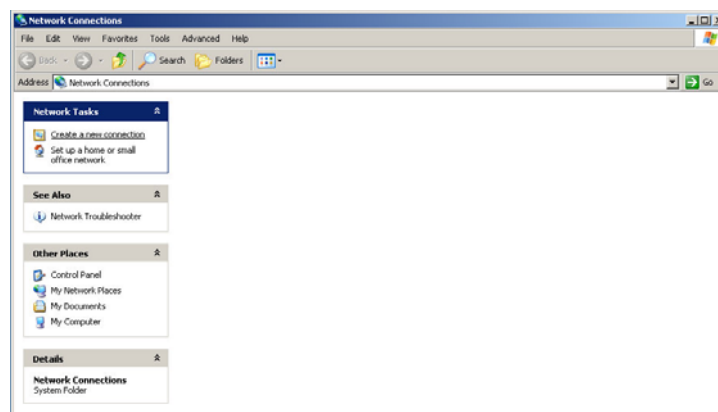


Figure 24. Network settings.

Go to the system’s network settings and start to create a new network connection.

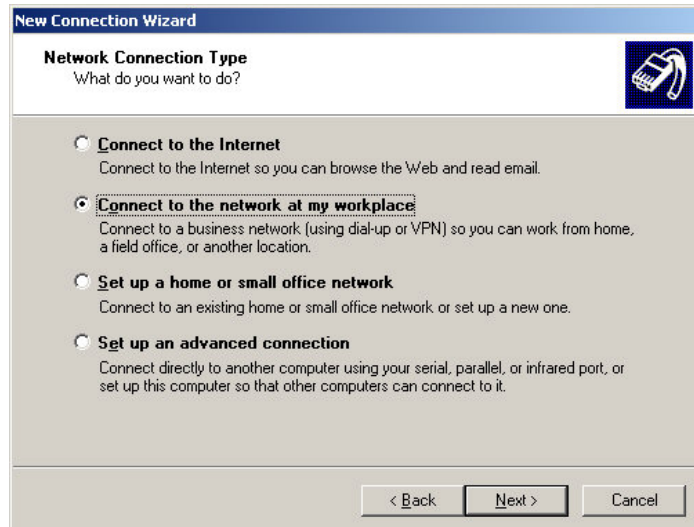


Figure 25. Selecting Network Connection type.

Define the new network to utilize VPN.

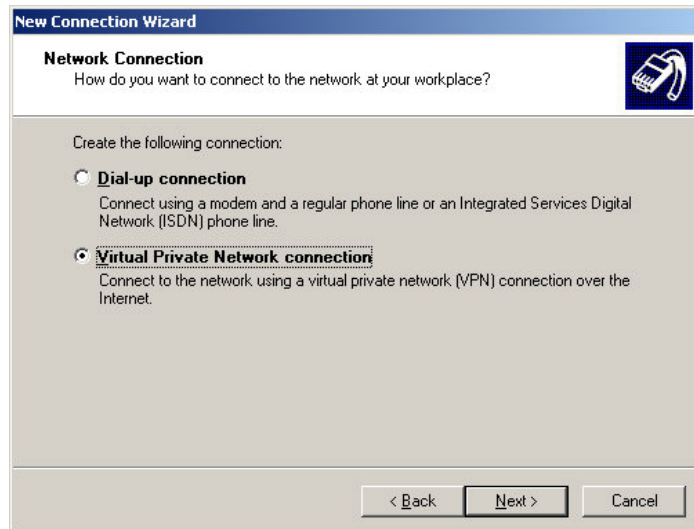


Figure 26. Define a new VPN network

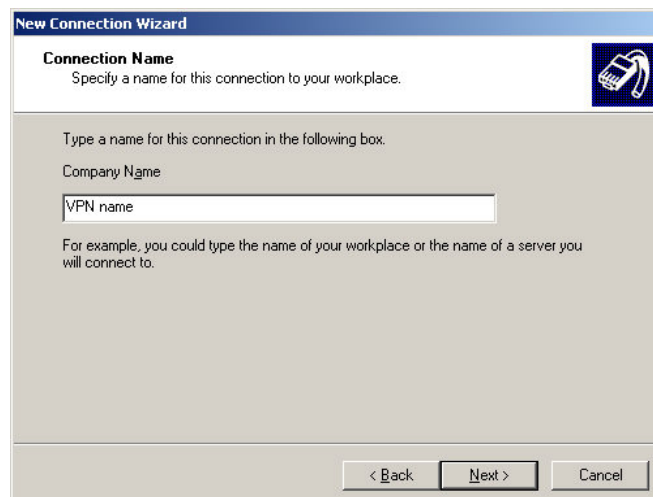


Figure 27. VPN network name.

Name the VPN connection e.g. “P2 VCT access”.

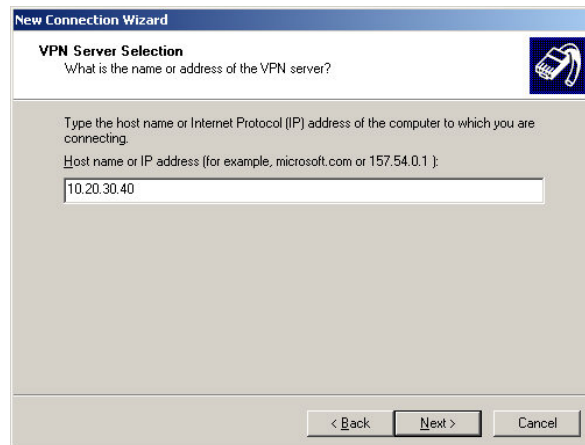


Figure 28. Entering IGW IP address.

Insert the IGW’s or VPN concentrator’s IP address sent to you by email and press next. Now your connection is created, insert username and password sent to you by email.



Figure 29. Enter user credentials.

Select Options and insert a high number of dialing repetitions and a low number of time in between. Select to automatically redial.

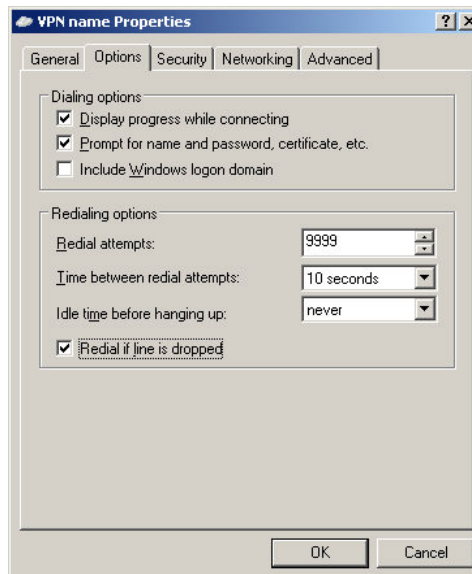


Figure 30. Configure connection options.

In the “Security” tab select typical (preferred) with secure password. Unselect all other options, don’t change any IPSec preferences.

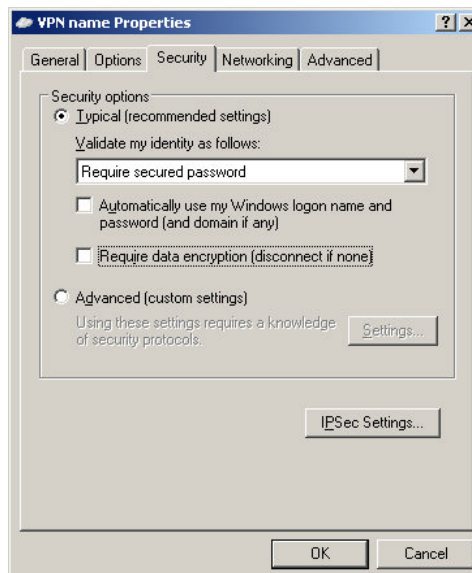


Figure 31. Security settings.

In “network” tab change the VPN type to “L2TP-IPSec”.

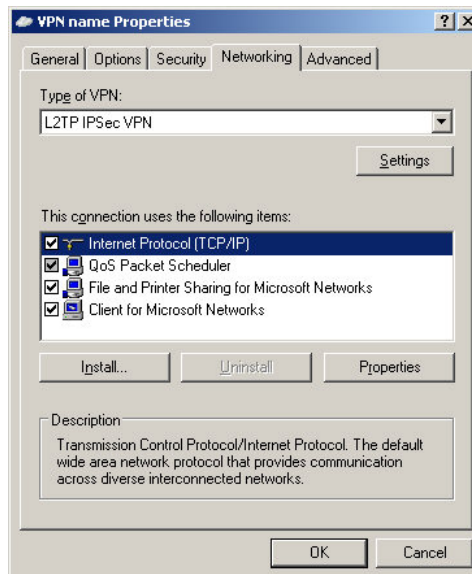


Figure 32. Configuring VPN type.

Now connect ...



Figure 33. Connecting network.

When connection was successfully created, you will find the new VPN connection in the system's network overview. You may create a link to that on the desktop or task bar.

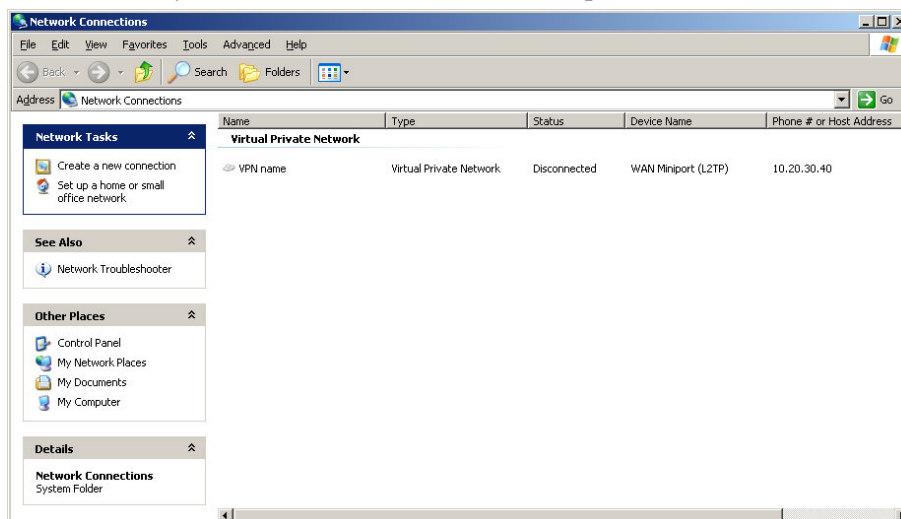


Figure 34. Created VPN network