

This document is intended for developers implementing the Test Network. It explores the various different potential interfaces to the Test Network. Each of these covers different use cases and have various different costs and benefits.

Contents

Scope.....	2
Main considerations.....	2
Architecture.....	3
Application Developer → Bubblephone.....	4
VM per AD: One virtual machine per developer.....	4
AD → BP as S/W: Developer delivers software.....	5
AD → BP as H/W: Developer delivers appliance.....	7
AD → BP as VM: Developer delivers Virtual Machine Image.....	8
Bubblephone → Application Developer.....	10
BP → AD as S/W: Bubblephone delivers software.....	10
BP → AD as H/W: Bubblephone delivers appliance.....	11
BP → AD as VM: Bubblephone delivers Virtual Machine Image.....	12
Bubblephone ↔ Application Developer.....	14
BP Hosted: Routing over the Internet.....	14
Cloud Hosted: VM installed in existing cloud services.....	15
Glossary.....	18



Scope

Customer-facing use cases are beyond the scope of this document. Therefore, of the viable implementations, one cannot be preferred over another, and so this document does not state a preference.

This document does categorise types of user interface according to their access method. It does not define the details of these interfaces because these details do not affect functionality.

This document does not propose businesses decisions or provide market research. No priority is placed on any use case.

Main considerations

Although many areas have been explored, the following are of most significant interest for immediate development:

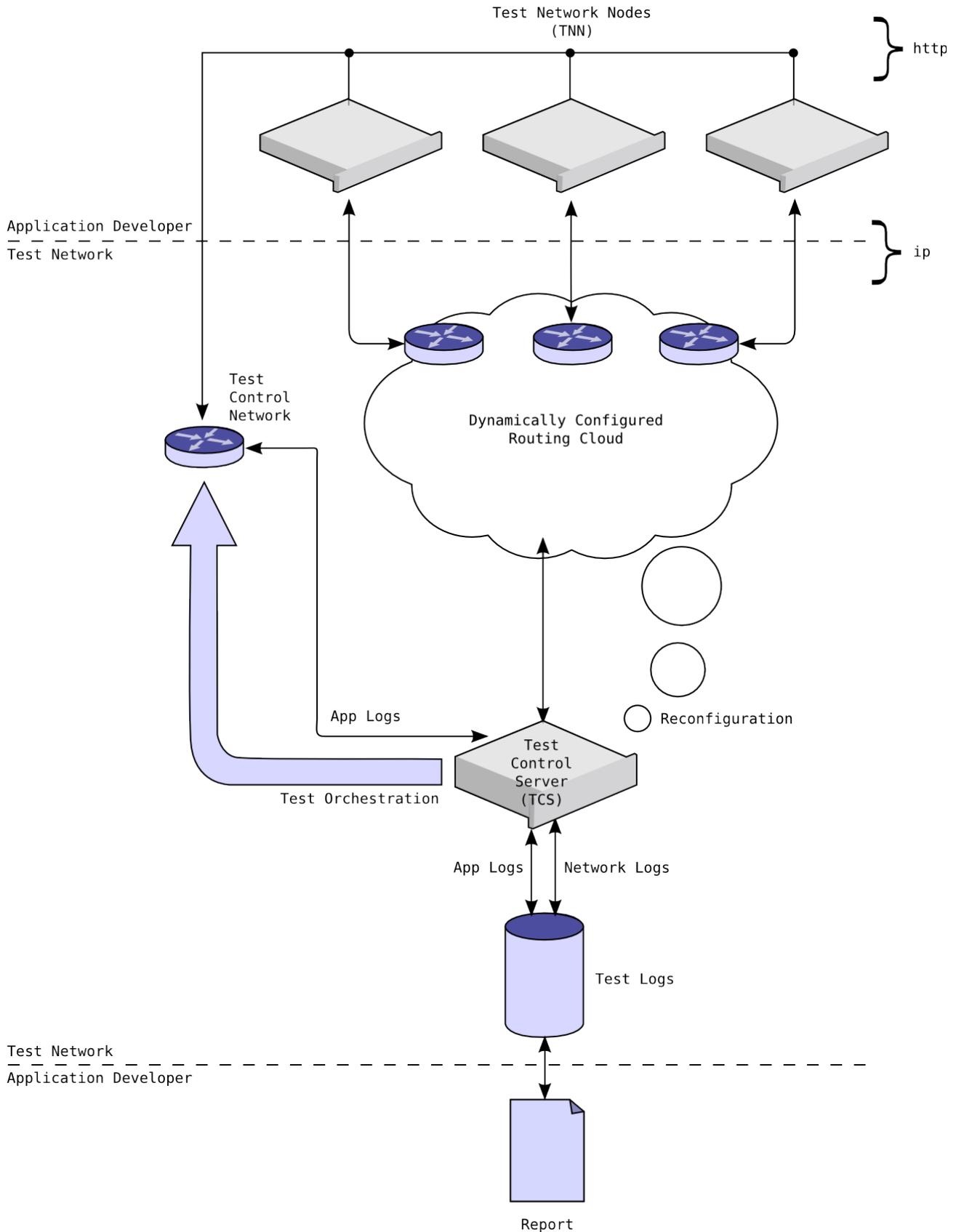
- **Cost to Bubblephone** (for both deployment and development)
- **Customer cost** (for both installation and per-test undertaken)
- **Hassle** (for both development and use)
- **Scalability**
- **Deterministic** (tests must be reproducible)
- **Multiple platform support:** This is the main motivation behind using IP and HTTP as interfaces; these permit the use of any OS without explicit support from Bubblephone.
- **Interfaces:** User Interface; Test orchestration API (TNC API); Tests' network interfaces (Network Interface)

Common to (almost) all approaches are options for an automateable user interface. For example, this might take the form of a website in the spirit of ebay's auction mechanism, to submit tests and review reports. The details of such interfaces are functionally equivalent, and hence beyond the scope of this document.

All of the following interfaces depend on transfer of information between Bubblephone and Application Developers. The bulk of this document categorises different situations as per the direction of this transfer and the nature of the information.

Architecture

The Test Network has an interface for administrative configuration (http), and an arbitrary number of Test Network Nodes, for traffic under test (ip). These nodes are orchestrated by the Test Control Server via HTTP.



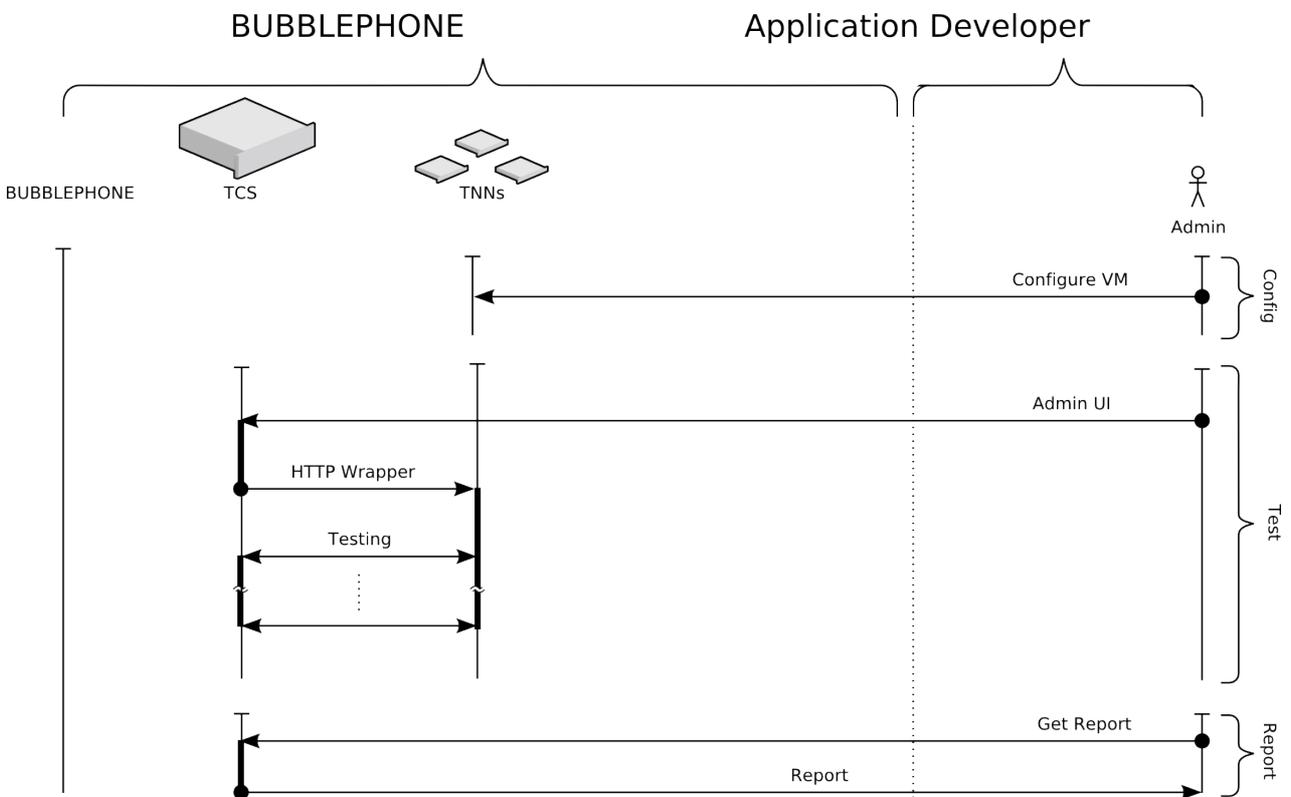
Application Developer → Bubblephone

The following scenarios describe all relevant situations where an application developer sends their software to Bubblephone. This does not necessarily infer that Bubblephone is responsible for performing the test.

- VM per AD: One virtual machine per developer..... 4
- AD → BP as S/W: Developer delivers software..... 5
- AD → BP as H/W: Developer delivers appliance..... 7
- AD → BP as VM: Developer delivers Virtual Machine Image..... 9

VM per AD: One virtual machine per developer

Customer connects to pre-configured VM & is responsible for their own software installation.



Interface	Consumer	Implementation	Description
Delivery	N/A	N/A	Bubblephone hosts the Test Network, and Customers access it remotely in-situ.
Admin UI	Application Developers	ssh, sftp	CLI use via ssh to a non-root account on our VM's OS, assuming a UNIX-like OS.
TNC API	Application Wrappers	route to HTTP REST	REST HTTP API bound to a management interface in BP's VM. (e.g. eth0, for Linux)
Network Interface	Applications	bind to IPs on OS's ifs	IPs on various network interfaces within our VM's OS. (e.g. IPs on eth1, eth2 for Linux)

The customer application would already need to be ported to our virtual machines.

Bubblephone would need to provide multiple operating systems if it is to market the test network to a range of customers. This gives a natural roadmap for supporting more operating systems, prioritised by desirability.

Bubblephone can safely give developers root access within these virtual machines as they can't harm the network topology by reconfiguring interfaces. This is because we can constrain those interfaces using a VLAN or firewall.

This interface supports installing dependencies on the behalf of developers, as for any hosted co-location environment.

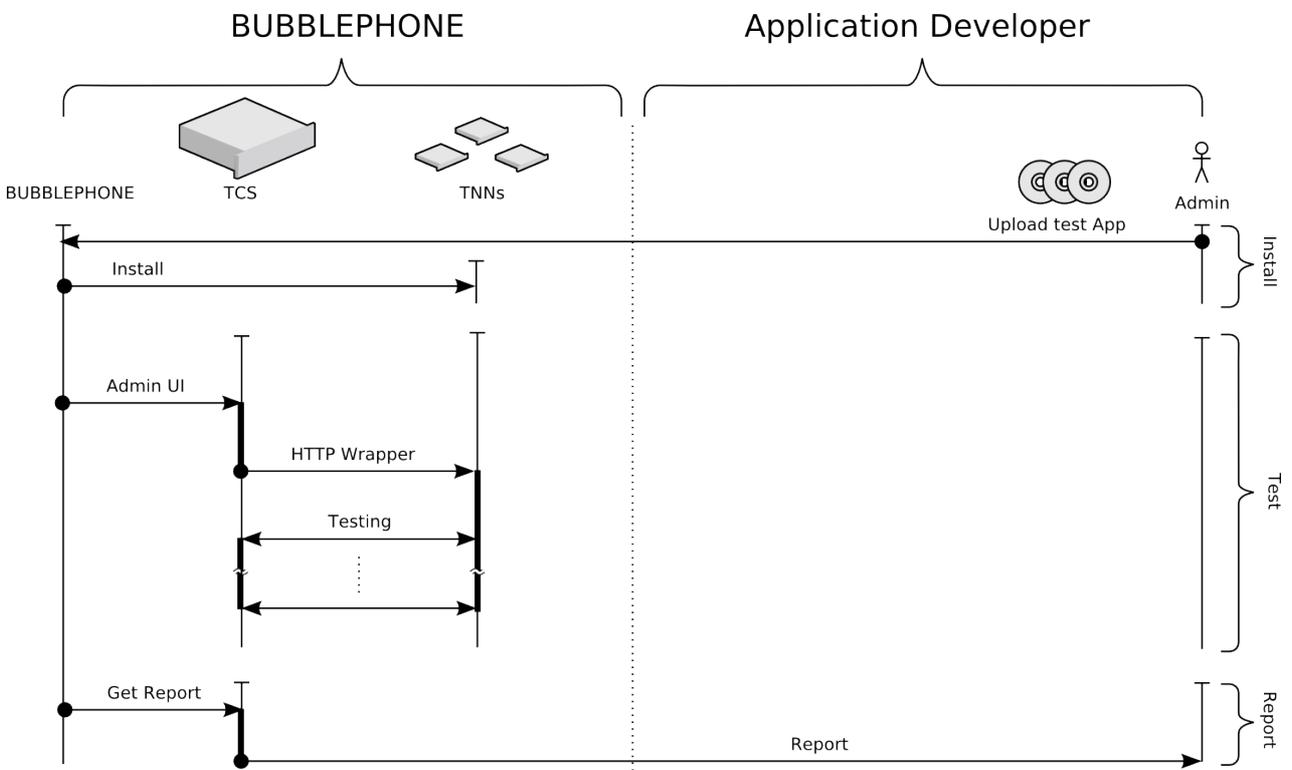
Giving developers complete freedom over their virtual machines means there is much room for error on their part. Time spent fixing misconfigured virtual machines can be costly.

Given that time spent managing virtual machines can be costly, this cost can be recouped by charging customers for support.

Application developers must be given virtual machine specifications, so they can ensure their application will fit comfortably within the virtual machine.

AD → BP as S/W: Developer delivers software

Customer provides their test application as software for Bubblephone to test. This is the same as delivering their application to an end user.



Interface	Consumer	Implementation	Description
Delivery	Bubblephone	email / ftp / post	File transfer (e.g. email, ftp, posting us a CD)
Admin UI	Application Developers	N/A	After sending us an application, the developer cannot access it. Communication is through customer support channels only.
TNC API	Application Wrappers	route to HTTP REST	RESTful HTTP API bound to a management network interface within our machine's OS. (e.g. eth0, for Linux)
Network Interface	Applications	bind to IPs on OS's ifs	IPs on various network interfaces within our machine's OS. (e.g. IPs on eth1, eth2 for Linux)

Application developers would need to deliver binaries for our specific operating system.

The HTTP wrapper may need to know of Bubblephone's addressing scheme in advance.

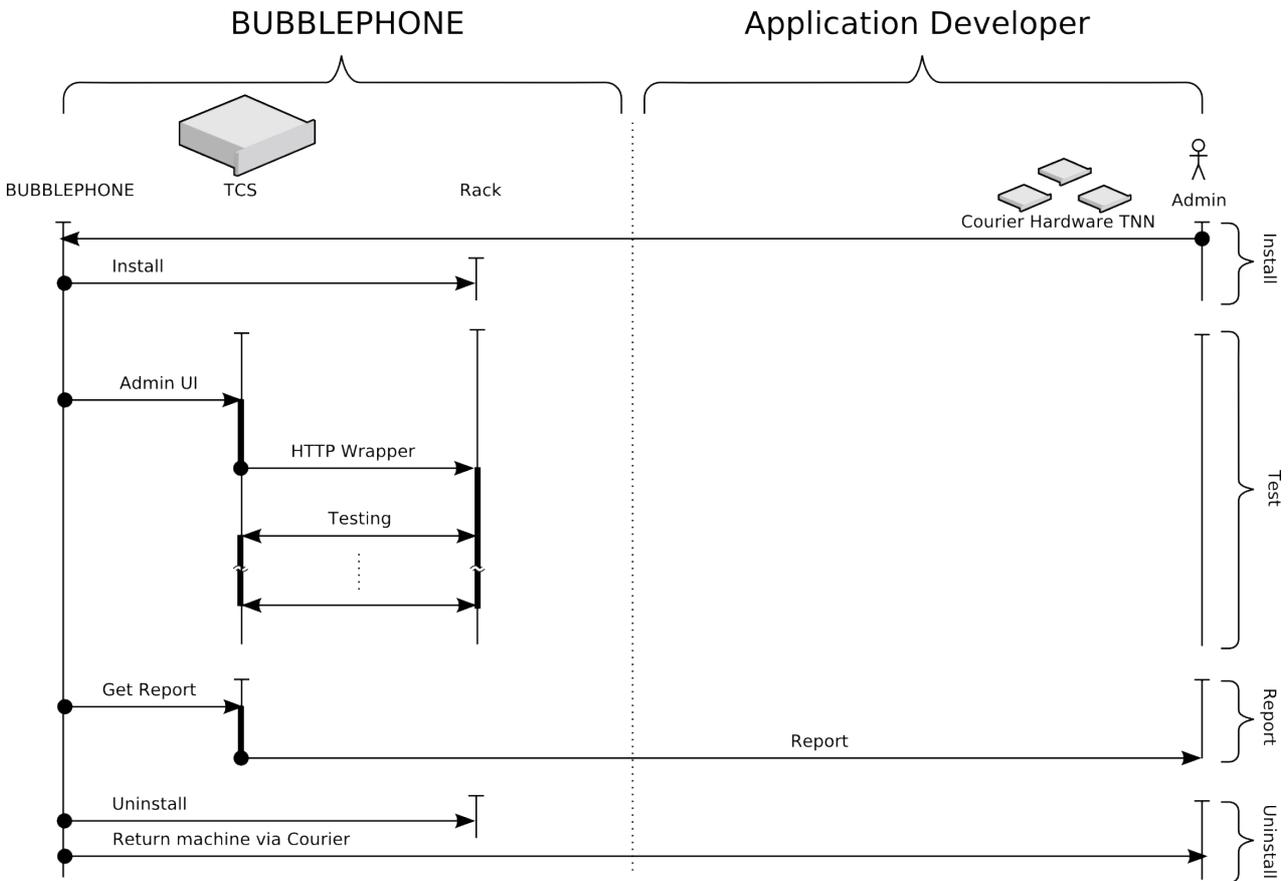
Bubblephone need to provide a range operating systems on popular architectures.

Application developers must be given machine specifications, so they can ensure their application will fit comfortably within the machine.

The details of the operating environment are not relevant to the application. Examples includes UML-Linux, OpenVZ etc.

AD → BP as H/W: Developer delivers appliance

Customer provides their test application as a self contained appliance.



Interface	Consumer	Implementation	Description
Delivery	Bubblephone	courier	The application developer's hardware is physically delivered to Bubblephone
Admin UI	Application Developers	N/A	After sending us hardware, the developer cannot access it. Communication is through customer support channels only.
TNC API	Application Wrappers	IP route to machine	RESTful HTTP API bound to their machine's network ports
Network Interface	Applications	IP route to machine	bound to their machine's network ports

Bubblephone needs to install the appliance on their network. The appliance is expected to request its address via DHCP.

Bubblephone needs no knowledge of the customer's application. The appliance is treated as a black box.

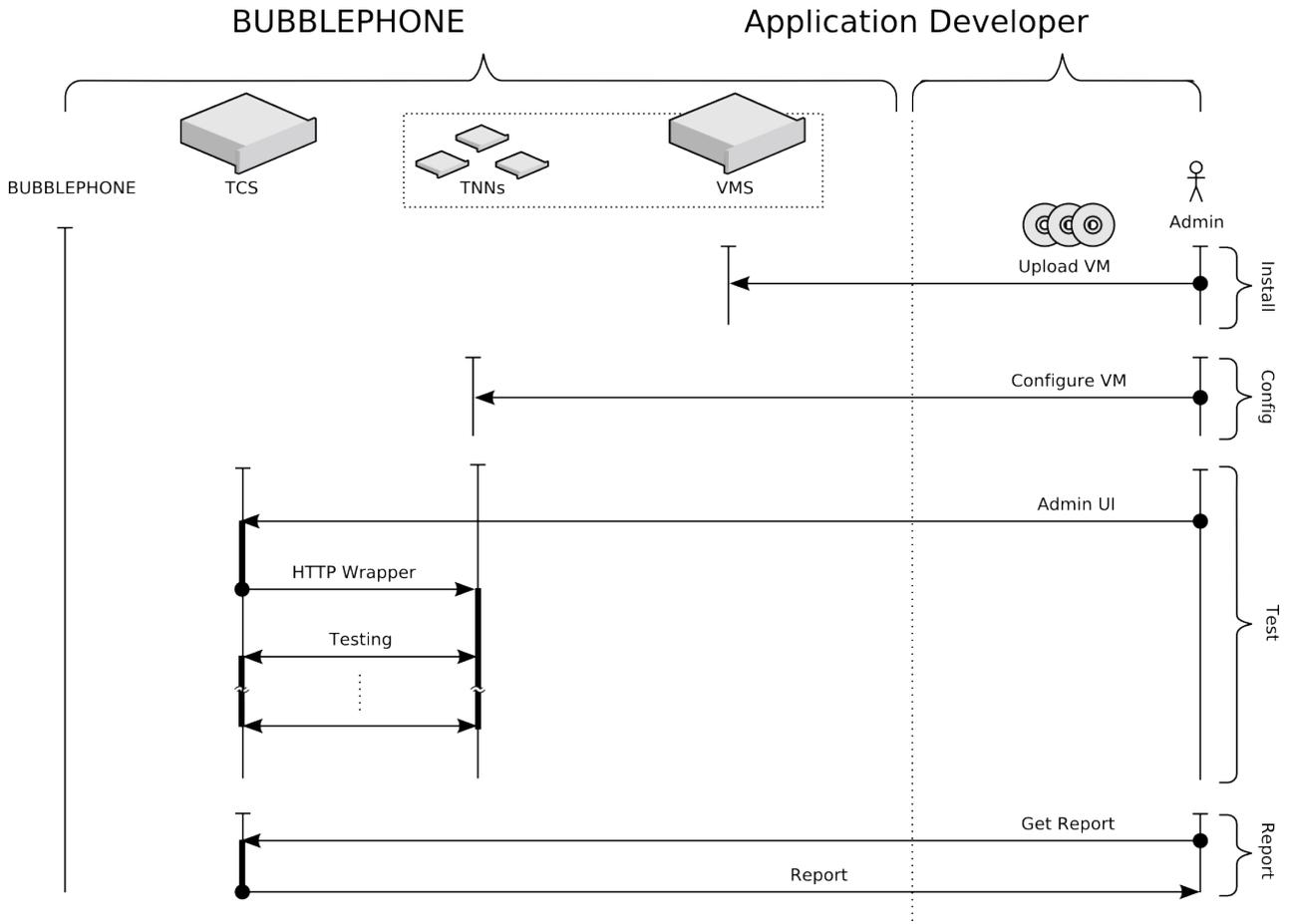
As the customer is providing the test application along with the appliance, there is no concern over the application's resource requirements.

Any interface that requires hardware delivery is inherently non-scalable.

Customer's applications can't adversely influence Bubblephone's network as Bubblephone can constrain these interfaces behind VLANs and firewalls.

AD → BP as VM: Developer delivers Virtual Machine Image

Customer provides an image of a running virtual machine containing their test application as software. Bubblephone deploys this image in order to perform tests against this software.



Interface	Consumer	Implementation	Description
Delivery	Bubblephone	email / ftp / post	File transfer (e.g. email, ftp, posting us a CD)
Admin UI	Application Developers	ssh / sftp / www	After sending us their VM image, developers can access both it and the TNC remotely.
TNC API	Application Wrappers	IP route to VM	RESTful HTTP API bound to their VM's network ports
Network Interface	Applications	IP route to VM	bound to their VM's network ports

Any interface that delivers test applications as virtual machine images is inherently scalable.

Bubblephone host the virtualisation environment, and are responsible for deployment of the customer's machine image. This image is expected to request its address via DHCP.

Application developers must be given environment specifications, so they can ensure their virtual machine image will fit comfortably within the environment. However, there is no concern over resource use, as their virtual machine encapsulates contention.

Customer's applications can't adversely influence Bubblephone's network as Bubblephone can constrain these interfaces behind VLANs and firewalls.

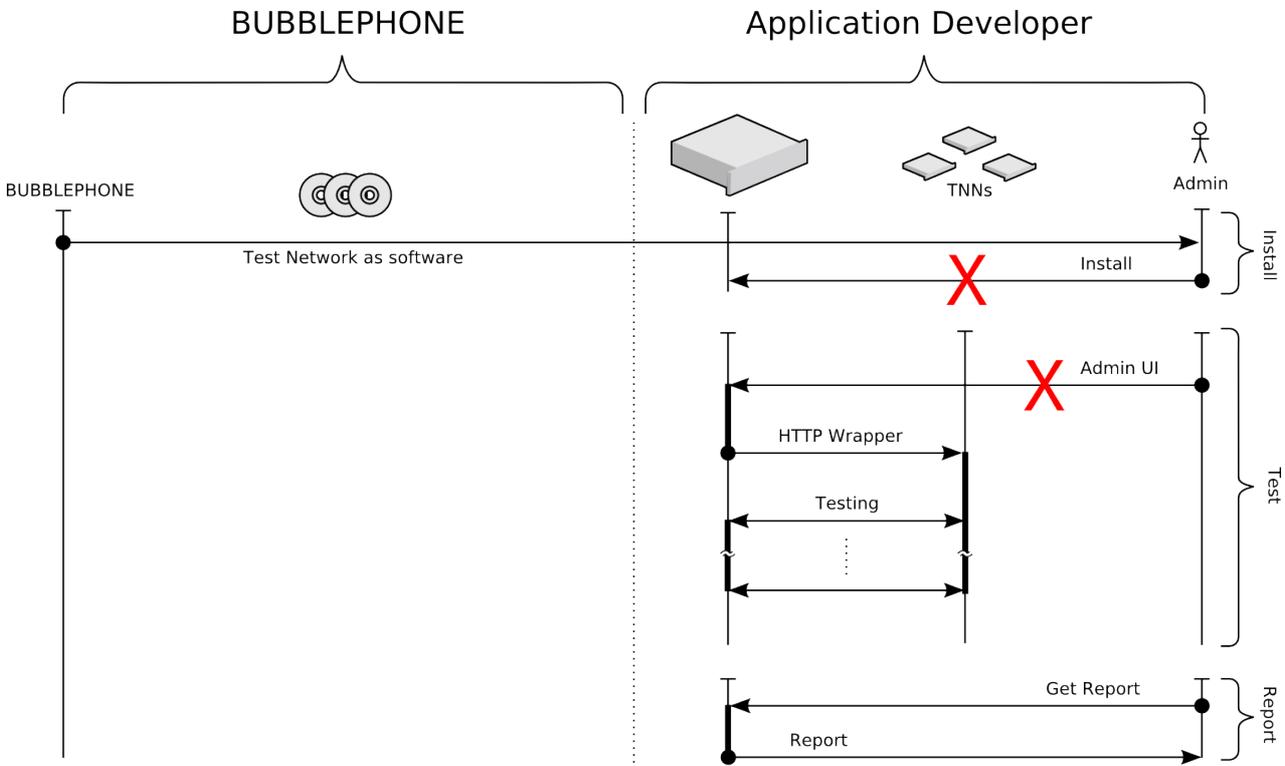
Bubblephone → Application Developer

The following scenarios describe all relevant situations where Bubblephone sends their software to an application developer.

- BP → AD as S/W: Bubblephone delivers software..... 10
- BP → AD as H/W: Bubblephone delivers appliance..... 11
- BP → AD as VM: Bubblephone delivers Virtual Machine Image.....12
- BP Hosted: Routing over the Internet..... 15

BP → AD as S/W: Bubblephone delivers software

Bubblephone provides the Test Network as software for the customer to install. The customer is responsible for installing the Test Network software and performing tests.



Interface	Consumer	Implementation	Description
Delivery	Application Developers	email / ftp / www / post	File transfer (e.g. email, ftp, download from website, posting them a CD)
Admin UI	Application Developers	ssh / web	Orchestration of tests and report generation by connecting to the TNC
TNC API	Application Wrappers	IP route to machine	REST HTTP API bound to AD's network ports
Network Interface	Applications	IP route to machine	bound to their machine's network ports

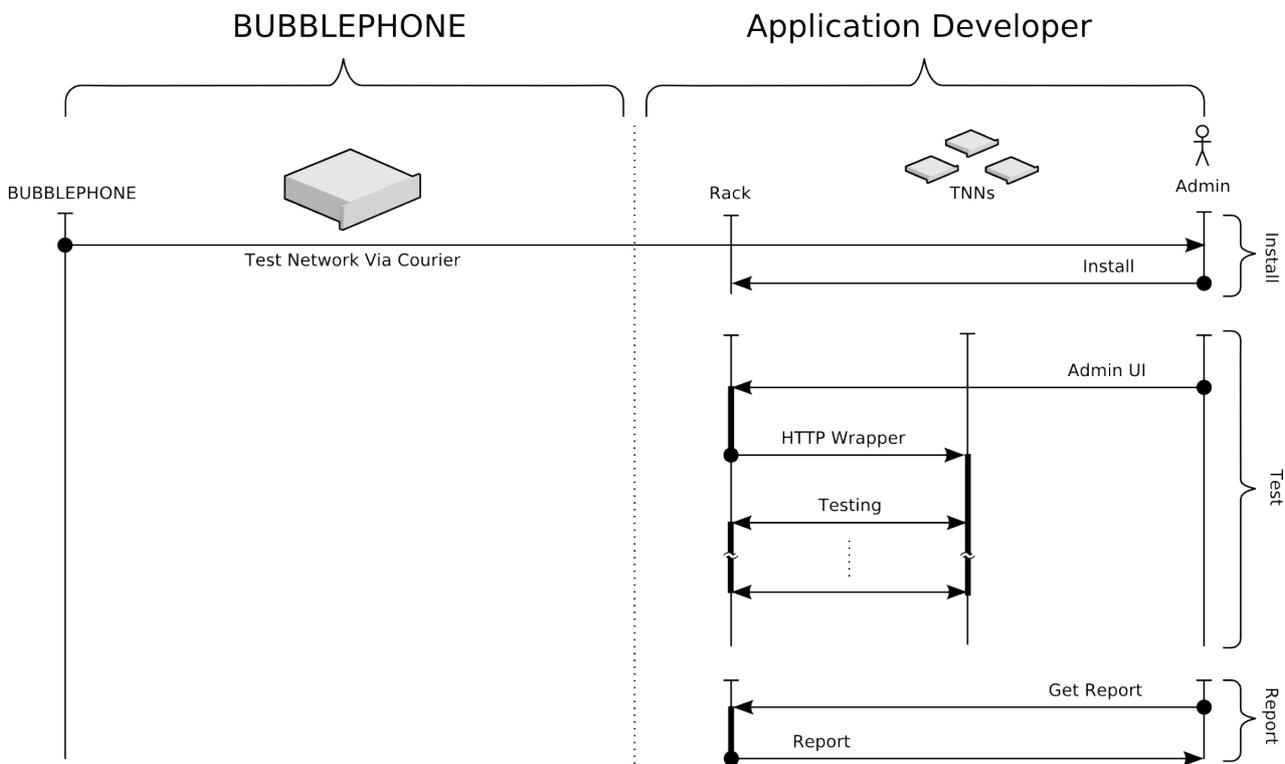
The Test Network has very specific hardware and operating environment requirements.

Builds of the Test Network would need to be delivered for various supported operating environments.

Probably not technically feasible, as the Test Network is not intended to be a single deployable application.

BP → AD as H/W: Bubblephone delivers appliance

Bubblephone provides the Test Network application as a self contained appliance. The customer is responsible for installation and all tests.



Interface	Consumer	Implementation	Description
Delivery	Application Developers	courier	The Test Network hardware is physically delivered to to Application Developer
Admin UI	Application Developers	ssh / web	Orchestration of tests and report generation by connecting to the Test Network Control server
TNC API	Application Wrappers	IP route to machine	RESTful HTTP API bound to their machine's network ports
Network Interface	Applications	IP route to machine	bound to their machine's network ports

Nice tangible deliverable .

Delivering the Test Network as an appliance may be relatively expensive.

All Effects used in the Test Network are implemented as external appliances. This includes effects delivered by Bubblephone, bespoke customer-developed Effects, and third party Effects such as firewalls.

As the customer's application is external to the Test Network, there is no concern over the application's resource requirements.

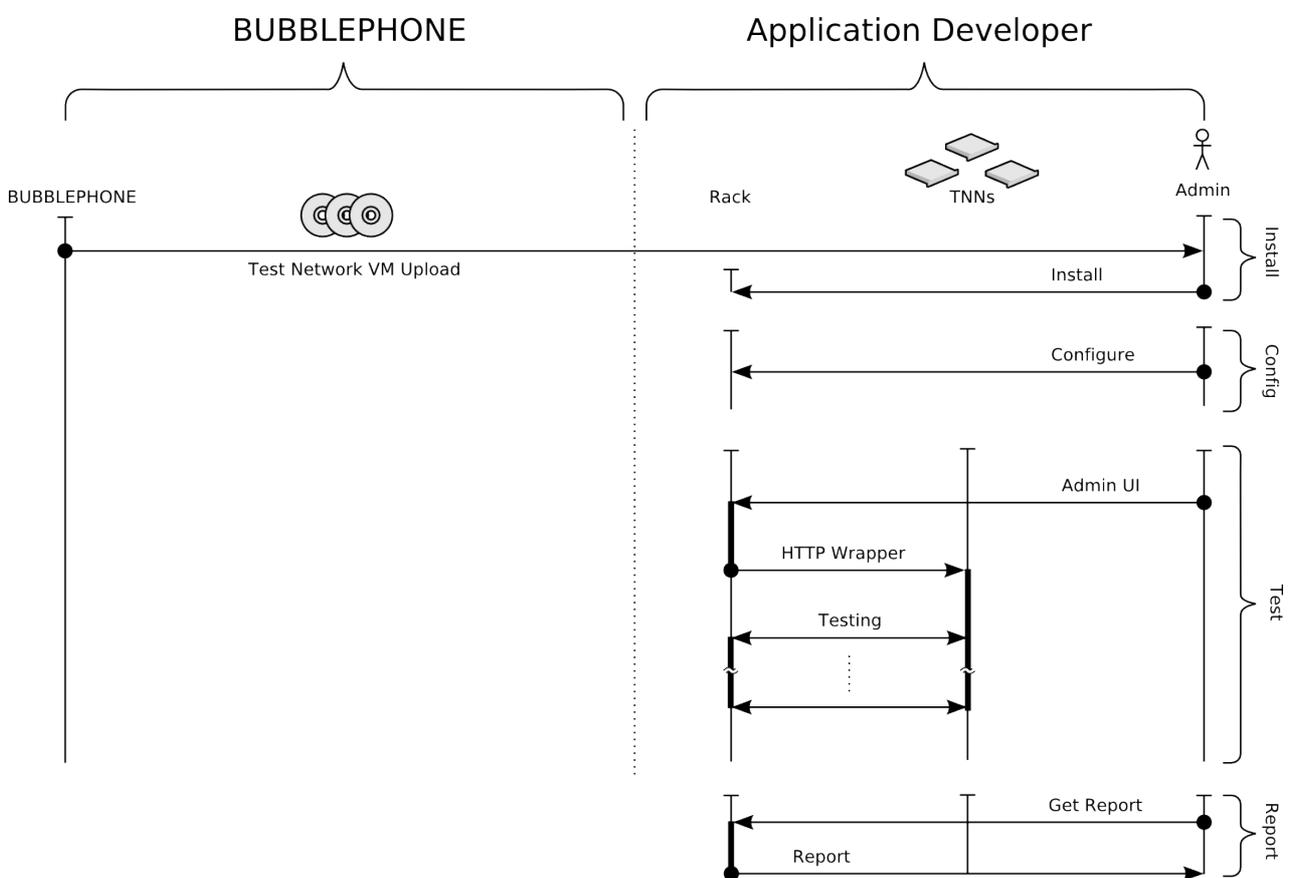
Delivering the Test Network as an appliance would require hardware support contracts with the client, with a channel for reporting bugs and avenues for software updates.

This interface does not require work on the part of the customer to package a formal release of their software. This is convenient because tests can be integrated as part of their development cycle.

The appliance provides a possibility for industry-standard logging interfaces, such as SNMP, and a CLI for administrative use.

BP → AD as VM: Bubblephone delivers Virtual Machine Image

Bubblephone provides an image of the Test Network as a virtual machine. The customer deploys this image in order to perform tests against their application.



Interface	Consumer	Implementation	Description
Delivery	Application Developers	email / ftp / www / post	File transfer (e.g. email, ftp, download from website, posting them a CD)
Admin UI	Application Developers	ssh / web	Orchestration of tests and report generation by connecting to the Test Network Control server
TNC API	Application Wrappers	IP route to machine	RESTful HTTP API bound to their machine's network ports
Network Interface	Applications	IP route to machine	bound to their machine's network ports

Same benefits as hardware, except without the cost of shipping and hardware.

As Bubblephone is delivering a software product, not a service, this still requires a support contract with channels for updates and bug reporting.

The customer would need a server on which to deploy the virtual machine. The customer must be given the virtual machine specifications, so they can ensure the Test Network will fit comfortably within their server.

Bubblephone has the opportunity to market the Test Network to a variety of popular virtualisation technologies.

The virtual machine provides a possibility for industry-standard logging interfaces, such as SNMP, and a CLI for administrative use.

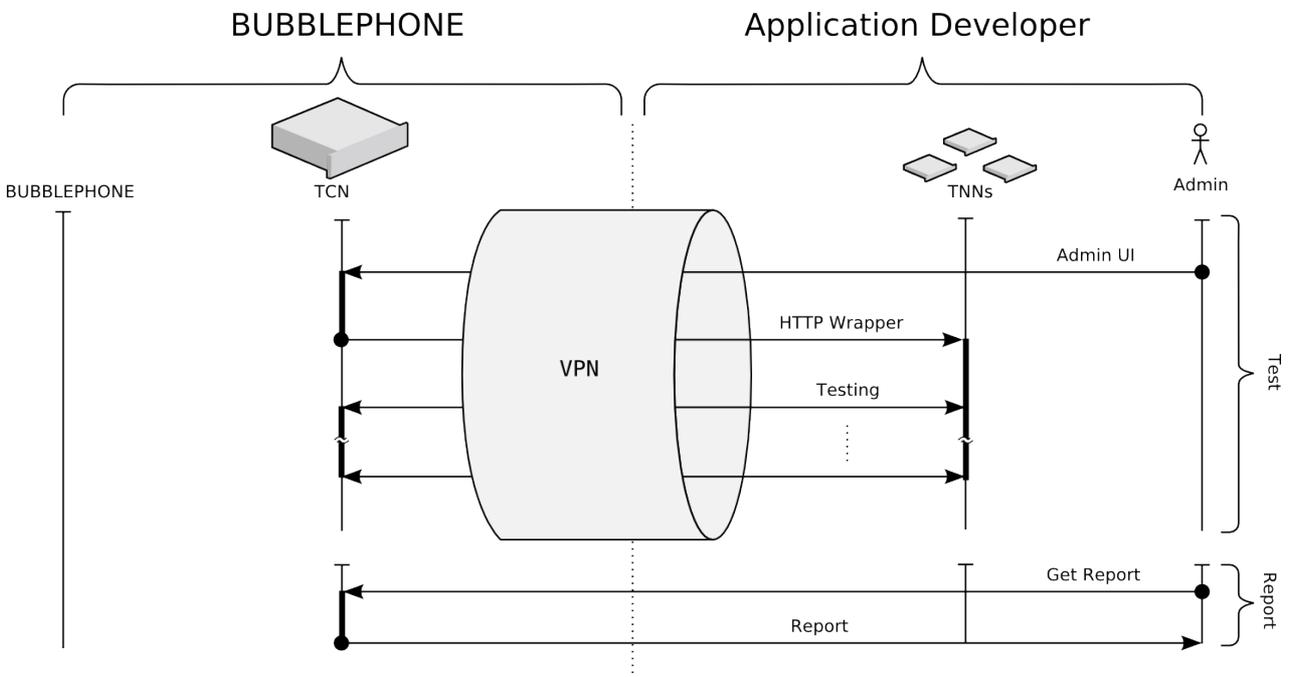
Bubblephone ↔ Application Developer

The following scenarios describe all relevant situations where Bubblephone and the application developer each host their own products respectively.

BP Hosted: Routing over the Internet..... 14
 Cloud Hosted: VM installed in existing cloud services.....15

BP Hosted: Routing over the Internet

Bubblephone provide an interface to the Test Network which is accessible via the internet. Customers are responsible for hosting their own applications. Customers access the Test Network via the internet in order to perform tests.



Interface	Consumer	Implementation	Description
Delivery	Application Developers	N/A	Applications are hosted by developers; the Test Network is hosted by Bubblephone
Admin UI	Application Developers	ssh / web	Orchestration of tests and report generation by connecting to the Test Network Control server
TNC API	Application Wrappers	IP route via VPN	RESTful HTTP API bound to their machine's network ports
Network Interface	Applications	IP route via VPN	bound to their machine's network ports

Routing the Test Network data over the internet can be made secure, but security comes with a performance penalty which may (non-deterministically) influence test results.

Access to the Test Network is provided by a VPN. Access to this VPN incurs overhead, both for test traffic and configuration.

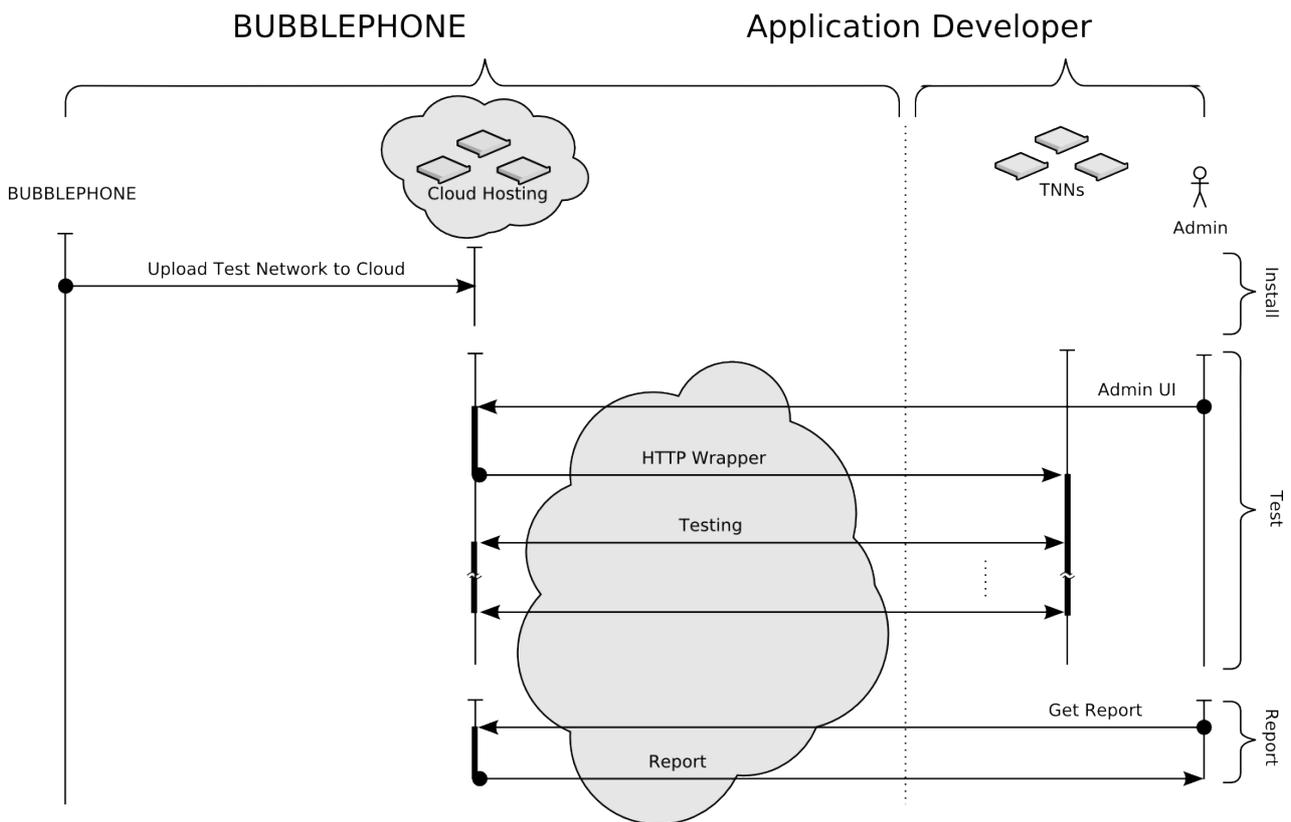
An internet based technology.

Large and variable delays. These can in part be accounted for by finding offsets to subtract, however this can be unreliable and can adversely affect tests results.

Each company (Bubblephone and the client) has expert knowledge in their respective areas. This interface leverages this knowledge by keeping the products of these companies in their own respective knowledge domain environment areas.

Cloud Hosted: VM installed in existing cloud services

Bubblephone provide an interface to the Test Network which is accessible via the internet. Customers deploy their application as Virtual Machine Images to "The Cloud". Customers access the Test Network via the internet in order to perform tests.



Interface	Consumer	Implementation	Description
Delivery	Application Developers	N/A	Applications are hosted by developers; the Test Network is hosted on The Cloud
Admin UI	Application Developers	ssh / web	Orchestration of tests and report generation by connecting to the Test Network Control server
TNC API	Application Wrappers	IP route via VPN	RESTful HTTP API bound to their machine's network ports
Network Interface	Applications	IP route via VPN	bound to their machine's network ports

The routing topology for “The Cloud” is mysterious and unknown, therefore large and variable delays will be experienced from time to time.

A Cloud Technology. The exact cloud technology used is beyond the scope of this document.

The Cloud abstracts scalability.

Feature Matrix	VM per AD	AD → BP as S/W	AD → BP as H/W	AD → BP as VM	BP → AD as S/W	BP → AD as H/W	BP → AD as VM	BP Hosted	Cloud Hosted
Delivery (per test)	N/A	file xfer	courier	file xfer	none	none	none	none	none
Delivery (installation of TN)	none	none	none	none	file xfer	courier	file xfer	none	none
Admin UI	CLI	no	no	CLI / web	CLI	CLI / web	CLI / web	CLI / web	CLI / web
TNC API	HTTP on VM	HTTP on VM	HTTP on H/W	HTTP on VM	HTTP	HTTP	HTTP	HTTP via VPN	HTTP via VPN
Network Interface	IPs on ifs	OS stack	IPs on ifs	IP on VM	infeasible	IP on H/W	IP on VM	IP via VPN	IP via VPN
Who executes/is involved during the tests	AD, Both	BP, BP	BP, BP	AD, Both	AD, AD	AD, AD	AD, AD	AD, Both	AD, Both
App needs porting to set of supported OSes	required	required	none	none	none	none	none	none	none
BP supporting multiple OSes?	many	many	none	none	some	none	none	none	none
Without root, who installs app.s' deps.?	AD	BP	AD	AD	AD	AD	AD	AD	AD
Support overhead (low, medium, high)	medium	high	low	low	medium	low	low	low	low
App runs on the machine it was made for	no	no	yes	yes	yes	yes	yes	yes	yes
Scalability (high, medium, low)	medium	medium	low	high	high	medium	high	high	high
App has no empiric knowledge of addressing	no	no	DHCP	DHCP	N/A	N/A	N/A	N/A	N/A
Potential for industry standard log interfaces	yes	no	no	yes	yes	yes	yes	yes	yes
Software/hardware product or Service	service	service	service	service	software	hardware	software	service	service
Latency on test traffic	constant	constant	constant	constant	constant	constant	constant	variable	variable
TN runs on BP-installed OS	yes	yes	yes	yes	no	yes	yes	yes	yes
App runs on AD-installed OS	no	no	yes	yes	yes	yes	yes	yes	yes
Technically feasible	yes	yes	yes	yes	no	yes	yes	no	no
TCO to AD	low	high	high	low	low	medium	low	low	low
Conclusion	okay	bad	bad	good	bad	good	good	bad	bad

Glossary

Port	Physical network port (e.g. RJ45)
Interface	Software network interface (e.g. eth0)
The Cloud	Abstract migration across dynamic resources accessible via The Internet
BP	Bubblephone
AD	Application Developer
H/W	Hardware
S/W	Software
VM	Virtual Machine