Basis of the Vternal Network

Martin Dunsmuir

October 2015

The purpose of the vternal network is to be able to store assets amongst a number of servers in such a way that it is very difficult to find them by random searching (unless they want to be found) and it is very unlikely that they will accidentally be destroyed. The idea is to place k copies of the asset each on one of at least k physically distinct vternal servers[1] chosen from a population N. If k is 3 and N is 1000, for example, then the chance that a copy of the asset will be found on in any particular computer will be 3/1000 but the chance that the 3 repositories where the 3 copies will be found randomly on 3 nodes is much smaller (3/1000)^3 = 27/1000000000 or 1 part in 37M; if there are 1,000,000 nodes then the chance is 1 part in 37,000,000,000,000,000 like finding 3 particular sand grains on a beach. Each node (or server) can only have, at most, one copy of a given asset A (actually any node can make as many backup copies of A as it likes but these are seen externally as ‘one copy of A’). In this document the terms “node” and “server” are used interchangeably.

One of the objections to the Vternal network is that it will become full. Whilst this is theoretically possible there are a number of reasons why this is very unlikely: To store an asset in the network a node must offer to store at least a much data itself; the size of the network and the size of assets stored in it will rise exponentially with time and so yesterday’s assets will be much smaller than today’s (take for example the size of digital images); and, thirdly, if the network is popular then new storage will be added constantly[2].

The system works on the basis that a first copy of the asset is made in one repository and that this copy is copied again and again to randomly chosen repositories until the number of copies reaches k,; at which point the copying process stops. A count of the number of copies is maintained by all the computers containing a copy of A. Periodically each computer with a copy of A checks that the computer which sent it the copy still is accessible. If a copy is inaccessible by the node which has cannot access the server from which it received the asset, then new copies are made on different nodes until k copies are restored.

Assets have a number of attributes, including: a unique hash or key identifying the assets uniquely; the asset data (which may or may not be encrypted); flags & metadata (identifying the asset format etc.); an accessibility date, before which the asset cannot be retrieved; an expiry date, and other data (for example a list of jurisdictions where the data cannot reside). In general assets cannot be deleted but if no-one knows their key then they are effectively lost. Lost assets might clog the network, but in practice this very unlikely for the reasons given above. In the

[1] A given computer can run more than one vternal server (on different ports for example) and although this configuration is useful for testing it defeats the purpose of physically distributing the assets.

[2] When a node joins the network it offers storage to the network, in 2015 this might be 4Gb for a smartphone of over 1Tb for a server.

future assets might include other elements, such as programs to protect them. Global assets will exist in the network for use by all nodes, for example programs to allow nodes to decode different data types.

Another important thing about the vternal network is that assets move around. If an asset were to remain of the original nodes it was stored on then its lifetime would be

Each computer maintains a flat list of all the computers it knows about. This list is maintained by a periodic process of conjugation. During conjugation a computer exchanges lists with another to which it is directly connected and both computers update their lists to match. Computers are removed from the list if they are no longer accessible. It is important to understand that there are no privileged or master nodes on the vternal network. It is a pure peer-to-peer network. Vternal.org may operate nodes which contain commonly used assets and large conjugation lists but this is not a requirement.

The bootstrapping process by which a new computer joins the network is that it must access an existing trusted node which is supplied to the new computer. Once the new node makes a connection to the existing (trusted - see below) node it conjugates with it and so very quickly it becomes aware of other trusted nodes.

To find a given asset A a request is sent to any computer known about by the requesting node. If the node contacted has A then this information is returned and the requestor asks the node where A resides to send it a copy; hence A is retrieved. if the request is sent to a computer which does not have A the request is sent to a computer randomly selected from those computers of which the first computer has knowledge and the process repeats itself until a computer responds to the original requestor and A is retrieved. Asset requests are batched up and exchanged with adjacent nodes during conjugation.

Along with each request is kept and maintained a list of the computers through which the request has passed. If a request visits all the computers on a nodes list without success it is destroyed. It is therefore the responsibility of requestors to refresh their requests if a response is not received in a ‘reasonable’ time. Having multiple requests for the same assets in the network at one time is not a problem.

If a given list entry is inaccessible then its tree is removed from the active list belong to that node but it is put the inaccessible node on a list of ‘missing’ hosts which are periodically polled to see if they exist (have come back online). If they reappear they are put back into the ‘active’ state. Hosts are never forgotten.

Trusted Nodes

It is very important that the network maintains all its nodes as trusted. It is possible that aberrant nodes could break the network by protocol violations, denial of service or other means (such as the delivery of bogus assets in response to requests).

To combat this a variety of safeguards are built into the network. First and foremost is the protocol to establish the trustworthiness of a node. This test is always performed prior to conjugation; second, asset keys are hashes of the original data and metadata associated with them when they first enter the network. Using this hash nodes can immediately tell if an asset is valid.

In order to establish trust the joining node (JN) gets a key from a trusted Key Issuing Authority (KIA). As part of this process the joining node generates a public/private key pair for itself and passes the public key to the KIA. Before issuing a key the KIA uses whatever means it wants (such as a credit card transaction) to establish the identity/owner of the joining node. Once the identity of the joining node has been established the KIA stores key/public key of the joining node and marks the joining node as trusted. The issuing of keys should be a slow process.

To join the network the JN passes its key and public key to the node it wishes to attach to (AN), plus the address of the KIA. The AN passes the key to the KIA and gets back the JN's public key encrypted with the AN's public key (which the KIA must know). The JN decrypts the validation with its private key (unknown by the KIA). If the decryption is successful the AN trusts the JN.

KIA's exchange key/public key pairs but if all the KIA go offline no new nodes can join the network, however the existing nodes can continue to operate normally.

The operation of KIA's is one of the primary functions of the Vternal Brotherhood.