ToMaTo Documentation

Release 3

Dennis Schwerdel

May 26, 2017

Contents

CHAPTER 1

ToMaTo parts

Hostmanager Documentation

Contents:

Hostmanager Installation

The hostmanager offers the virtualization technology found on the host to backends. For simple installation of the hostmanager, it has been packaged for Debian systems.

Installation on Debian systems

All of the following commands must be issued as **root**.

1. Adding the repository

   ToMaTo has its own repository for debian packages that needs to be added in order to install its packages.

   ```
   echo 'deb http://packages.tomato-lab.org/deb stable main' > /etc/apt/sources.
   ↪list.d/tomato.list
   ```

2. Accepting the repository key

   Since all packages are signed with keys, the repository key must be accepted once, otherwise the Debian package manager will complain on every update that the package is unauthorized.

   ```
   wget http://packages.tomato-lab.org/key.gpg -O - | apt-key add -
   ```

3. Updating the package lists

   ```
   apt-get update
   ```

4. Installing the Hostmanager package

```
apt-get install tomato-hostmanager
```

During the configuration phase of this package, dialogs will appear and propmt for information. All of these prompts can be answered by pressing *enter*.

5. Optional: Install the Updater package

```
apt-get install tomato-updater
```

This package will add a cronjob that keeps your ToMaTo installation automatically up-to-date.

Installation on Proxmox systems

The target platform for the hostmanager is Proxmox VE so there exists a meta-package specifically for Proxmox systems, that installs all additional software that is needed to use the full potential of Proxmox systems.

To install the hostmanager on Proxmox systems, the **steps 1 to 4 from above** have to be executed. Additionally the package `tomato-host-proxmox` has to be installed:

```
apt-get install tomato-host-proxmox
```

After the installation

Some steps are needed to finalize the installation:

- Installation of additional packages so that more *Element types* and *Connection types* become available. (For Proxmox hosts, the package `tomato-host-promxmox` installs all needed software.
- *Hostmanager Configuration*
- *Backend management*

Hostmanager Configuration

Hostmanager API

The hostmanager offers the following methods via an XML-RPC interface. The interface uses an encoding as documented in *XML-RPC Interface*. All of the methods can be called by their method names without modules, etc.

Elements

Connections

Accounting

Resources

Host

Backend management

The hostmanager allows different backends to use it.

Component separation

All components (i.e. elements and connections) of different backends are separated by the hostmanager. Each component will have an owner attribute, that references the backend that created it. The component will only be visible and accessible by that backend.

Resource separation

All resources (i.e. networks and templates) of different backends are separated by the hostmanager. Each resource will have an owner attribute, that references the backend that created it. The resource will only be visible and accessible by that backend.

Access control

The authentication of backends uses SSL keys. Each backend has a key of its own and uses it to authenticate and encrypt connections to hostmanagers. On the side of the hostmanager all backend keys must be present as files in *PEM format* in a specific directory (`/etc/tomato/client_certs` in default config).

Indexing the backend keys

After modifying the SSL keys, the certificate index must be rebuilt.

```
update-tomato-client-certs
```

Note that the hostmanager does not have to be restarted after rebuilding the index. Also note that the hostmanager will issue the command to rebuild the certificate index automatically when it is starting.

Backend identification

The identity of a backend is based on the *common name (CN)* in its certificate. Different certificates with the same common name field will be treated as the same backend and share access to components.

Generating a key-pair

A self-signed key-pair can be created with the following command:

```
openssl req -new -x509 -days 1000 -nodes -out key.pem -keyout cert.pem
```

It is important to create a key without a password if the the key should be used for a backend.

Integrated Fileserver

The backend has an integrated fileserver that allows ToMaTo users to upload and download data files like disk images, packet capture files directly from/to the hostmanager without indirection via the backend.

Access

The fileserver uses the HTTP protocol on a port set in the config file. The public address of the host and the fileserver port can obtained using the API call `hostmanager.tomato.api.host.host_info()`.

Element types

OpenVZ (`openvz`, `openvz_interface`)

KVM with QM frontend (`kvmqm`, `kvmqm_interface`)

KVM with VirSH frontend (`kvm`)

Repy (`repy`, `repy_interface`)

Tinc VPN (`tinc`)

UDP Tunnel (`udp_tunnel`)

Connection types

Bridge (`bridge`)

Fixed bridge (`fixed_bridge`)

Other topics:

XML-RPC Interface

The interface is an RPC interface, i.e. it offers a set of methods that can be called (with parameters) and that return a return value. The internal data format is XML but this should be transparent. The interface adheres the XML-RPC standard with the following modifications:

- It uses an extension for null-value encoding that is not part of the standard. This extension is part of many implementations since the absence of the feature is seen as a flaw in the standard.
- It uses a special parameter encoding that allows for keyword arguments. If exactly two parameters are given, where the first one is a list and the second one is key/value-map the keyword mode is used. In the keyword mode, the first parameter (the list) is expanded and used as normal positional arguments nad the second argument (the key/value-map) is expanded and used as the keyword arguments.

Web-Frontend Documentation

Command-Line-Client Documentation

Basic CLI functionality

`cli.tomato.`**`getLocals`**(*api*)

Combines the api with additional functionalities in one dictionary. It adds a list of all commands, a help method and a method to load and initialize python modules from a file.

Parameter *api*: Connection to a host api

Return value: This method returns a dictionary with a connection to a API, an help method and a method to load and initialize python modules from a file.

`cli.tomato.parseArgs()`

Defines required and optional arguments for the cli and parses them out of sys.argv.

Available Arguments are:

Argument –*help*: Prints a help text for the available arguments

Argument –*url*: The whole URL of the server

Argument –*protocol*: Protocol of the server

Argument –*hostname*: Address of the host of the server

Argument –*port*: Port of the host server

Argument –*ssl*: Whether to use ssl or not

Argument –*client_cert*: Path to the ssl certificate of the client

Argument –*username*: The username to use for login

Argument –*password*: The password to user for login

Argument –*file*: Path to a file to execute

Argument *arguments* Python code to execute directly

Return value: Parsed command-line arguments

`cli.tomato.run()`

Parses the command-line arguments, opens an API connection and creates access to the available commands of the host. It decides based on the options whether to directly execute python code or to execute a file or to grant access to the interactive cli.

`cli.tomato.runFile(locals, file, options)`

Opens a connection to a remote socket at address (host, port) and closes it to open the TCP port.

Parameter *locals*: Dict containing a connection to an API, a help function and a file load function.

Parameter *file*: Path to the file which should be executed

Parameter *options*: Command-line arguments which will be used to create an interactive console which executes the file.

`cli.tomato.runInteractive(locals)`

Creates a interactive console based on the local available methods.

Parameter *locals*: Dict containing a connection to an API, a help function and a file load function.

`cli.tomato.runSource(locals, source)`

Executes a python code using an interpreter based on the methods provided by the API found in locals.

Parameter *locals*: Dict containing a connection to an API, a help function and a file load function.

Parameter *source*: Source code to execute

Additional functionality

Upload / Download commands

`cli.lib.createUrl(protocol, hostname, port, username=None, password=None)`

Creates a URL for connecting to a server.

Parameter *protocol*: Protocol of the server

Parameter *hostname*: Address of the host of the server

Parameter *port*: Port of the host server

Parameter *ssl*: Boolean whether ssl should be used or not

Parameter *username*: The username to use for login

Parameter *password*: The password to user for login

Return value: This method returns a full server URL.

`cli.lib.`**`getConnection`**(*url*, *sslCert=None*)

Creates a server proxy to a host using the given URL.

Parameter *url*: URL of the server

Parameter *sslCert*: SSL certificate to use for a ssl connection

Return value: This method returns a server proxy object.

`cli.lib.`**`tcpPortOpen`**(*host*, *port*)

Opens a connection to a remote socket at address (host, port) and closes it to open the TCP port.

Parameter *host*: Host address of the socket

Parameter *port*: TCP port that will be opened

Return value: This method returns a boolean which is true, if the TCP Port is open and false otherwise.

Advanced link commands

`cli.lib.misc.`**`is_superset`**(*obj1*, *obj2*, *path=''*)

Checks whether obj1 is a superset of obj2.

Parameter *obj1*: Superset object

Parameter *obj2*: Subset object

Parameter *path*: Should be "" in initial call

Return value: Returns a tuple with 2 arguments. The first argument is a boolean which is true, if obj1 is superset of obj2, false otherwise. The second argument returns a string if the first argument is false. The string contains the reason why obj1 is not a superset of obj2.

`cli.lib.misc.`**`link_check`**(*id*, *ip*, *tries=5*, *waitBetween=5*)

Checks the availability of a link by trying to reach him a certain number of tries.

Parameter *id*: ID of device which should be reached

Parameter *ip*: IP address of the ping target

Parameter *tries*: Number of tries

Parameter *waitBetween*: Time between each try

Return value: Returns a boolean which is true, if the link was available within the number of tries, false otherwise.

`cli.lib.misc.`**`link_config`**(*top*, *con*, *c*, *attrs*)

Configures an link by modifying the certain attributes

Parameter *top*: Topology in which the link can be found

Parameter *con*: Link which should be modified

Parameter *c*: Target interface

Parameter *attrs*: Key value pair of attributes which should be configured

`cli.lib.misc.`**`link_info`**(*id*, *ip*, *samples=10*, *maxWait=5*, *oneWayAdapt=False*)

Pings a target IP address from a certain device and returns the results. The number of samples and the maximum wait time for responds can be set. Also a one-way adaption of the results is possible.

Parameter *id*: ID of device which should be used.

Parameter *ip*: IP address of the ping target.

Parameter *samples*: Number of messages to send.

Parameter *maxWait*: Time to wait for a responds in seconds.

Parameter *oneWayAdapt*: Change results to a one-way adaption.

Return value: The return value of this method is a dict containing information about the route between the link and the destination.

`lossratio` The loss ratio of the route between the link and the destination.

`delay` The average round-trip time.

`delay_stddev` The average standard deviation for the delay.

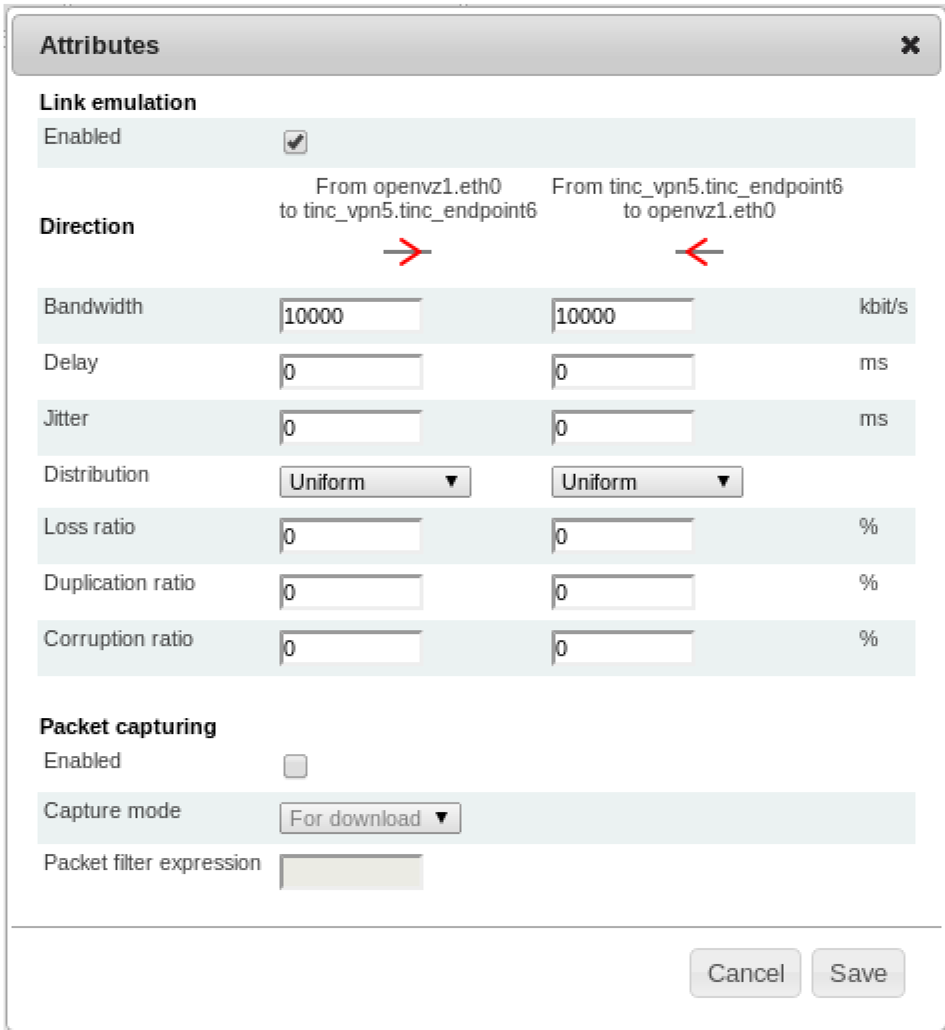
CHAPTER 2

Other topics

Link Emulation

Link emulation is an important feature for some experiments. Link emulation means that conditions of real networks and their effects on traffic are emulated in an environment with different conditions.

ToMaTo supports an extended set of link characteristics based on the tc/netem capabilities. With ToMaTo, users can add *packet loss*, *packet duplication*, *packet corruption*, *delay*, and *bandwith limitation* to existing connections. These conditions are applied to the data that flows over the connection additionally to the conditions that it already possesses. The conditions can be set separately on each direction of a conection. The emulated conditions can be changed any time (even on running connectors) in the graphical editor.



Mode of operation

To emulate the link characteristics ToMaTo uses two combined queues per direction. The queues for the different directions of the connection are completely independent.

The first queue each packet has to go through is a netem queue that emulates all link characteristics except bandwidth limitations. This queue delays packets, drops them depending on the loss and emulates duplication and corruption. Packets that exit this queue (after being delayed) are processed by the second queue.

A token-bucket-filter queue is used as the second queue to enforce bandwidth limitations. Tokens are produced with a fixed rate limited to a certain maximum. If there are tokens left, packets are sent immediately and the tokens are reduced. Otherwise the packets are put on hold and delayed until enough tokens have been produced.

Note: A note on buffering

Both queues need buffers for their normal execution. The netem queue needs the buffer to store packets that have to be delayed and the TBF queue needs the buffer to store packets until enough tokens are ready to send them. ToMaTo tries to automatically calculate good values for these buffers.

The TBF buffer is sized so that 25ms of data (at the maximal rate) can be buffered. This should be enough to actually reach the maximal rate. The minimal buffer size is one MTU of 1540 bytes.

The netem buffer is sized so that enough packets of an average size of 250 bytes can be delayed by the configured delay to reach the configured maximal bandwidth without causing loss.

Note: A note on transitions

Due to technical restrictions changes to the link emulation settings can only be applied in two steps: First the token bucket filter is changed to apply the new bandwidth settings, which results in the Netem queue being temporarily removed. After this the Netem queue that does all the other link emulation is reestablished.

Because of this two-step process there is a short gap between these steps where only the bandwidth is enforced but the delay, loss, etc. are not applied and the link has perfect properties. In most cases this gap is too short to be noticeable but sometimes, depending on load, scheduling can result in the gap length rising to about 10ms (The length of a scheduler time slice).

Bandwidth limitations

Bandwidth limitations can be set on connections to limit the maximal data rate. The bandwidth limitation must be configured in kilobits per second. The default bandwidth limitation for all connections (if no limit is configured) is 10 MBits.

Note that due to the token bucket filter short bursts can have a higher effective bandwidth than the configured limit but the sustained bandwidth is limited by the configured maximum.

Also note that traffic that exceeds the limit first fills the buffers and then is dropped if the buffers are full. See above for an explanation of the buffer sizes.

Delay

If delay is configured for a connection, packets on that connection are stored in a queue and sent when they are due. The timing granularity depends on the capabilities of the realtime clock of the hosts but in general is very precise. Delays must be configured in milliseconds.

Note that delays are only added to the connection and that other delays like processing delays and real transmission delays add to the configured delay.

Jitter

Jitter can be applied as a random modification of the delay per packet. The jitter value must be configured in milliseconds and configures the delay by which the configured delay can be increased or reduced per packet. The way jitter is calculated depends on the chosen distribution.

Note that jitter can cause packets to be reordered if the jitter value has the same order of magnitude as the time between two packets.

Note that if the jitter is higher than the delay all resulting negative per packet delays are treated as zero delay.

Delay distribution

The delay jitter can be allied with a _uniform_ distribution meaning that all values in the jitter range are equally likely (this is the default).

The _normal_ distribution has a higher probability for values near the configured delay but is still symmectric for values higher and lower than the configured delay.

The _pareto_ distribution follows a power law where most of the delays are a little lower than the configured delay and some delays are much higher.

The _paretonormal_ distribution is a mixture of both distributions with 75% pareto and 25% normal distribution.

Note that only the uniform distribution guarantees that the per packet delay will be in [delay-jitter .. delay+jitter] and that the median of the pareto distribution is lower than the configured delay

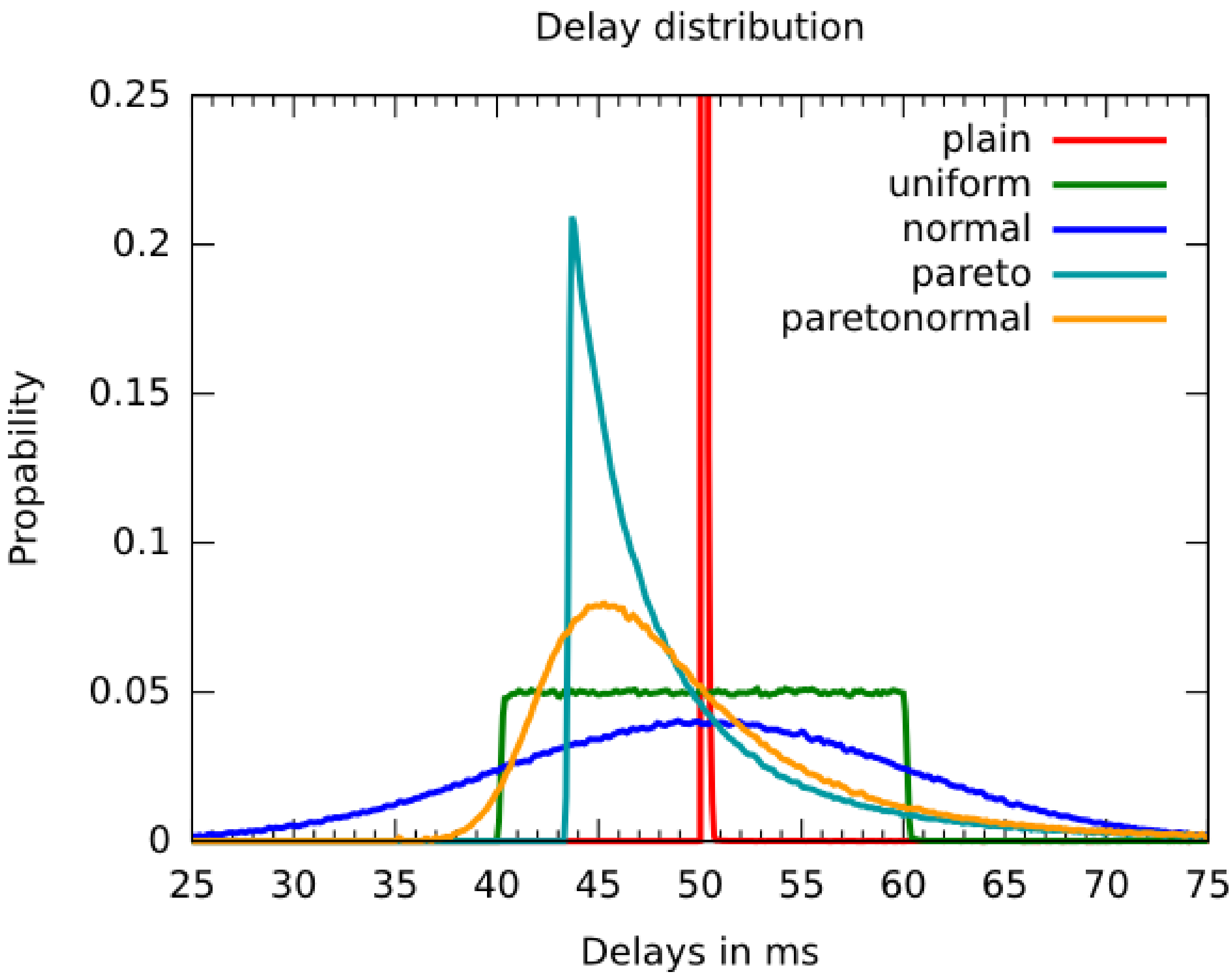


Fig. 2.1: Different delay distributions with delay=50ms and jitter=10ms

Packet loss

Packet loss in ToMaTo in applied as a per-packet probability. This means that for a 20% loss ratio there is no guarantee that one of 5 packets gets lost or that at most one of two packets gets lost. It might take a lot of packets until the number of lost packages is somewhere near to 20%. Packet loss is configured in percent, thus 20% is represented as 20 (not 0.2).

Loss correlation

Packet loss can optionally be correlated. That means that the loss probability of a packet depends on the loss probability of the packet before. The loss correlation is configured in percent like the loss probability.

Packet duplication

If a packet duplication probability is configured, packets are duplicated with that probability. That means that an exact copy of the packet is constructed and send directly after it. Duplication will create one copy at maximum.

Note that since the packets first pass the duplication and then the rate limiting duplicated packets count twice on the bandwidth usage.

Packet corruption

If a packet corruption probability is configured, packets are corrupted with that probability. That means that a bit at a random position in the packet is flipped. Corruption will flip at most one bit in a packet.

Note that in most cases checksums of the underlying layers prevent corrupted packets to be seen on higher layers.

Physical link statistics

As said before, the link emulation applies additionally to the existing conditions on the underlying links. To set up an experiment it is important to estimate these conditions. ToMaTo supports this by running periodic checks to measure delay and packet loss between all sites. The averages of these measurements can be viewed in the web-frontend in the admin section (accessible by all users).

Other link emulation possibilities

If you need more that the built-in link emulation of ToMaTo you can use a KVM machine as the center of the network and add a link emulation software with your emulation rules.

Another way to do link emulation in ToMaTo is to use an entity outside ToMaTo (on a physical host) to do the link emulation and to get access to it via external network connectors.

Packet Capturing

Packet capturing can help to trace packages through the network and analyze communication streams.

ToMaTo capabilities

ToMaTo supports capturing of packets on connections on Tinc-based connectors. The capturing can be enabled in the graphical editor in the properties panels of the connections. The captured packets are saved to a rotating set of files holding at most 50 MB of data. The capture files can be downloaded by clicking the “download capture” button in the control panel of the connection.

The timestamp in the capture files do not exactly correspond with the time of sending the packet in the virtual machine since the scheduling might introduce a delay. However the timestamp is guaranteed to be between the time of sending and the time of the forwarding to the connection.

Also note that timestamps from different hosts might have a certain offset, depending on how good the clocks of the hosts are synchronized. In the German-Lab testbed currently no actions are taken to synchronize the clocks among the hosts.

Analysis programs

ToMaTo generates capture files in the pcap format. When downloaded from the hosts multiple capture files are packed into a tar.gz archive.

The capture files created by ToMaTo can be used by a lot different programs:

- Wireshark - a graphical pcap explorer an analysis tool
- Cloudshark - a web-based pcp explorer with a similar UI to Wireshark
- tcpreplay - a Linux tool to replay pcap files

Device Templates

Template distribution

The device templates are distributed using the bittorrent protocol. This way the templates can be distributed among the hosts without a noteworthy central component.

For the bittorrent disctribution the backend and all hosts run a bittorrent client that automatically downloads and uploads the contents of torrent files in a certain directory. The backend also runs a so called bittorrent tracker, i.e. a central registry for the bittrorrent protocol that keeps track of all available peers for a torrent file.

The backend periodically checks that all templates are known to all hosts and are up-to-date. Otherwise the backend will create resource entries for the missing templates containing the torrent information. Since the torrent information has a size of several KiB (depending on the content size) the host will include an MD5 hash in its information and the backend will only update the torrent information when it does not match the hash.

The host will periodically check the file size of the templates and compare them to the information given in the torrent file to determine if the download has finished.

Template setup

To make them easier for users, templates should follow some common priciples.

1. Templates should be **secure by default**. This means that by default templates should only run services that are essential to the function of the template. For Linux templates that means that the SSH server will be deactivated by default and has to be activated manually by the user.
2. Templates should **only contain needed modifications**. This means that a template should match the default version of the operating system except where adaptations are needed. This should help users that are already familiar with the operating system to use a template of that OS.
3. Templates should **be as small as possible**. This means that templates will be compressed to save space and only include useful software. For some operating systems this might mean to remove some drivers and software that will never be used, to save space.
4. Templates should be **international but work in Germany**. The language for all templates should be set to american english but the keyboard layout should be set to german.

5. Templates should be **self-explaining and helpful**. That means that templates should contain some documentation on their special features and how to use them.
6. Templates should **not assume internet access**. Without internet access the templates should still work unless they explicitly require an external service in the internet.
7. Templates should **use DHCP**. All existing interfaces should be configured using DHCP. Hostname, DNS and time servers should also be used if included in the DHCP offer.
8. Templates should **require no login for local users and use a default password**. This means that local users (via VNC) should be loggen in directly without entering a username or password. If a password is needed for some actions, the password should be the same for all templates. Note that templates still must be secured against the network and require passwords for non-local login.
9. Templates should **include useful tools**. Not all devices will have internet access so templates should already include the most useful tools that users want installed. There is a clear trade-off between keeping a template small and including useful tools.
10. Templates should be **updated regularely**. This is important in two cases:
    (a) If the device has internet access, it is important that the template is up-to-date so that is initially secure. After device preparation, the user will have the responsibility to keep the system updated but it should be secure to start with.
    (b) If the device does not have internet access, it is important that the template is not outdated because the user can not easily update it.

Template generation

Scripts that can help to create and clean up templates can be found in the repository in the directory `contrib`. The scripts `create_kvm_template.sh` and `create_openvz_template.sh` can be used to create templates for debian-based systems in a semi-automatic way. The script `prepare_vm.sh` can be used to adapt a running system to be a proper template.

Torrent creation

Torrent files for templates can be created using the command

```
btmakemetafile TRACKERURL FILENAME
```

where TRACKERURL is the URL of the tracker and FILENAME is the name of the template file. The result will be a torrent file, that is named like the template file with `.torrent` appended.

Note: The ToMaTo backend includes a tracker that can be used for template torrents. Its URL can be determined by the backend API call `backend.tomato.api.host.server_info()`.

Accounting data types

Usage record

A *usage record* represents a data set with usage statistics of a certain time range. Each usage record has the following fields:

type: The *type* of a usage record describes the time frame which the record covers. Possible values are `single` for a single measurement, `5minutes`, `hour`, `day`, `month` and `year` for aggregated values.

begin and end: The fields *begin* and *end* describe the covered time of the measurement. For a *single* measurement these fields specify the begin and the end of the measurement execution for this single data point. For all other record types these fields specify the time frame of aggregated measurements. Both fields are timestamps in the form if seconds since the epoch (1970-01-01 00:00:00).

measurements: The field *measurements* contains the number of single data points that are combined in the record. This field is `1` for all *single* records and contains the number of aggregated single records for all other types.

usage: This field describes the resource usage during the given time period. For *single* records this time period is the period between the measurement and the last single measurement. For all other types, the time period is the combined time period of the aggregated *single* records. The field *usage* is a dict containing the following fields:

cputime: This field contains the used CPU time in seconds as a float value. CPU time is automatically measured by the operating system, so the measurement results do not depend in measurement timing. Even bigger gaps in measurement do not cause inaccurate values. Note that CPU time is calculated per core, so it is possible to consume several seconds of CPU time during one second.

memory: This field contains the used memory (RAM) in bytes. Memory consumption is measured on certain measurement points and the different data points are averaged into aggregated values.

diskspace: This field contains the used disk space in bytes. The measurement is similar to that of the *memory* field.

traffic: This field contains the traffic volume in bytes. Like *cputime*, traffic is automatically measured by the operating system and the values are very accurate because of this.

Note that because of different nature of the resources *cputime* and *traffic* are summed up during aggregation while *memory* and *diskspace* are averaged.

Usage statistics

The usage statistics data structure contains a set of *usage records*. Usage statistics objects are dict structures that contain the *types* of the usage records as keys and a list of usage records with that type as value.

Example


```
{
  "5minutes": [],
  "hour": [],
  "month": [],
  "single": [
    {
      "usage": {
        "traffic": 0.0,
        "cputime": 0.0,
        "diskspace": 19285.0,
        "memory": 0.0
      },
      "type": "single",
      "begin": 1351241166.88561,
      "measurements": 1,
      "end": 1351241166.89418
    },
    {
```

```
      "usage": {
        "traffic": 0.0,
        "cputime": 0.0,
        "diskspace": 19285.0,
        "memory": 0.0
      },
      "type": "single",
      "begin": 1351241106.80326,
      "measurements": 1,
      "end": 1351241106.81239
    },
    {
      "usage": {
        "traffic": 0.0,
        "cputime": 0.0,
        "diskspace": 19285.0,
        "memory": 0.0
      },
      "type": "single",
      "begin": 1351241226.93197,
      "measurements": 1,
      "end": 1351241226.94053
    },
    {
      "usage": {
        "traffic": 0.0,
        "cputime": 0.0,
        "diskspace": 19285.0,
        "memory": 0.0
      },
      "type": "single",
      "begin": 1351241286.97878,
      "measurements": 1,
      "end": 1351241286.98769
    }
  ],
  "year": [],
  "day": []
}
```


Repy

Repy is a turing-complete subset of Python that allows to run in a sandboxed environment.

Python and Repy

The Python programming language is documented at docs.python.org/reference. Repy is a reduced version of the Python programming language that allows to run scripts in a sandboxed environment. Repy is part of the Seattle Testbed and has an extensive documentation in the Seattle Wiki.

Difference between Repy and Python

- No imports, no external libraries. The `import` statement is forbidden in Repy. Some functionality from Python libraries is made available via special identifiers. (see below)
- No `global` variables. Instead Repy has a dictionary `mycontext` that can be used to store global variables.
- No user input via `input` or `raw_input`.
- **Some Python builtins are not available. The most important are**
  - `print`
  - `eval` and `execfile`
  - `lambda`
  - `reload`
  - `reversed` and `sorted`
  - `staticmethod`
  - `super`
  - `unicode`
  - `yield`
  - `hasattr`, `getattr` and `setattr`
- Parameters are passed as `callargs` instead of `sys.argv` and start with index 0 instead of 1 (`sys.argv[0]` is the script itself).

Methods available to Repy scripts

Output methods

echo(*message*)
will print the message (followed by a newline) to the console.

print_exc(*exception*)
will print an exception and a stack trace to the console.

Threading/Locking methods

createlock()

getthreadname()

createthread()

Misc. methods

exitall()

sleep(*time*)

randombytes()

getruntime()

getlasterror()

Networking methods

tuntap_read(*dev*, *timeout=None*)
will read one packet from the given network device *dev* and return this packet as a byte string. The method will block until a packet arrives at the device but at most *timeout* seconds (forever if *timeout=None*). If no packet has been received before the timeout, *None* will be returned. It is an error if the device does not exist.

tuntap_read_any(*timeout=None*)
will read one packet from any network device and return it. The return value will be a tuple *(dev, packet)* of the incoming device and the packet as a byte string. The first packet that arrives at a network device will be returned. The method will block until a packet arrives at a device but at most *timeout* seconds (forever if *timeout=None*). If no packet has been received before the timeout, *(None, None)* will be returned. It is an error to call this method if no network devices exist.

tuntap_send(*dev*, *data*)
will send the packet *data* via the network device *dev*. The packet must be a byte string. It is an error if the device does not exist.

tuntap_list()
will return a list of all available network devices.

tuntap_info(*dev*)
will return a dictionary containing detailed information about the networking device *dev*.

Struct

The struct library is available via *struct* (no import needed). This library can be used to encode and decode binary data structures.

ToMaTo library

The tomato library contains implementations of protocols and nodes. This library is extensible, so please feel free to contribute.

Databases

The ToMaTo backend needs a database to store information about hosts, topologies and users. The choice of this database is important for the performance of the ToMaTo backend.

ToMaTo uses Django as a database backend so the Django database documentation applies to ToMaTo as well.

SQLite

Note that SQLite lacks some features of real databases and thus is not suitable for running or developing ToMaTo.

PostgreSQL

PostgreSQL is the database that is used in the German-Lab installation. It is a full-featured database with good performance.

Raising the connection limit

The default database connection limit of PostgreSQL is set to 100 which can be reached by ToMaTo if several users are running a lot tasks in parallel. In the config file `postgresql.conf` the value `max_connections` can be raised to allow more concurrent connections. If the postgres server then hits the shared memory limit, the sysctl value `kernel.shmmax` needs to be increased. (See the PostgreSQl documentation for more details.)

Exporting and importing the ToMaTo data

The `manage.py` script that comes with the ToMaTo backend can be used to dump and load the database contents in a generic database-agnostic format. These commands might only work when run as user `tomato`, so `sudo -u tomato ./manage.py ...`. Also note that the commands only work when the database is up-to-date with the current layout in the code. (See migration for details)

Dumping the database

The following command dumps the database to a file named `dump.json` in the current directory:

```
$ ./manage.py dumpdata tomato south > dump.json
```

Loading the data into the database

The following command load a dump from a file named `dump.json` into the database. (Note that the file extension must be skipped in this command)

```
$ ./manage.py loaddata dump
```

Database migrations

As the database layout of ToMaTo changes, the database must be migrated to the new layout. ToMaTo automatically migrates old database schemas to the newest one. (Backups should still be made before the migration.)

Glossary

API API is short for "Application programming interface". Both the *backend* and the *hostmanager* provide APIs with which they can be used.

Backend The backend is one of the parts of ToMaTo. It is the central part that manages all *resources*, *hosts*, *topologies* and users. In a ToMaTo-setup the backend is the only part that can only exist once. The backend uses the capabilities of one or more *hosts* to offer *topologies* to its users through one or more *frontends*.

CLI The command-line interface is a simple way to control both the *backend* and the *hostmanager*. It is one of the *frontends*.

Component A component is either an *element* or a *connection*.

Connection A connection is a relation between exactly two *elements*. The connnection can have attributes of its own.

Connector This is a term from an older version of ToMaTo. Connectors are *elements* that are network elements.

Device This is a term from an older version of ToMaTo. A device is an *element* that is a virtual machine.

Dict Dicts are key-value mappings in the python programming language. In a dict, each key has a value assigned to it. When used in an *API*, the keys are limited to strings and the keys are limited to serializable objects (numbers, strings, booleans, None, lists, dicts).

Element An element refers to virtual objects that the user can control. This includes end systems like virtual machine and scripts as well as networking components like switches, hubs or routers. Each element can have several attributes and child elements (VMs have network interfaces as child elements) and one *connection*.

Entity An entity is a very general term for something that is controlled by either the *hostmanager* or the *backend*. This includes *elements*, *connections*, *topologies*, *users*, *resources* and much more.

Frontend ToMaTo frontends are a part of ToMaTo. Frontends connect to the *backend* and give users access to its capabilties by using the *backend API* and the users credentials. Several frontends exist (e.g. web-frontend and CLI) and can access a *backend* in parallel.

Host A host is a physical machine (computer or server) that hosts parts of *topologies*. Each host is managed by one *host manager*.

Host manager (or Hostmanager) The host manager is one of the parts of ToMaTo. It has exclusive control over one *host* and offers its capabilities to one or more *backends*.

KVM KVM devices are heavy-weight virtual machines that emulate a whole computer with generic hardware. Most things that are possible on physical computers is also possible on KVM. Most operating systems run on KVM.

OpenVZ OpenVZ devices are light-weight virtual machines that translate kernel calls to kernel calls of the host kernel. OpenVZ offers complete usermode access to the virtual machines and a limited kernel-mode access.

Profile Profiles define the resource boundaries for virtual machine *elements*. Depending on the VM technology, profiles define different attributes like RAM limit, disk space and number of CPUs.

Repy Programmable devices are essentially scripts that can work with networking packages. These scripts can be written in a Python dialect called Repy and can read and write raw Ethernet packets to/from their network interfaces. Programmable devices are very light-weight as they are just small Python scripts.

Resource Resources are a generic *entity* type for things that are present at *hosts* and can be used by *elements*. This includes *templates*, external networks but also available port numbers.

Template Templates are pre-installed disk images for virtual machine *elements*. Depending on the VM technology different templates with different operating systems and software exist. For *Repy* the template is the actual script that should be executed.

Topology A topology is a virtual network containing topology *components* (i.e. *elements* and *connections*). For the user, a topology is a virtual world where he can run his experiment.

Licenses & Copyright information

Backend

The backend code is owned by University of Kaiserslautern on behalf of the German-Lab project and licensed under the GNU Affero GPL version 3.

```
Copyright (C) 2010 Dennis Schwerdel, University of Kaiserslautern
```

Frontend

Server code

The frontend code except the graphical topology builder applet is owned by University of Kaiserslautern on behalf of the German-Lab project and licensed under the GNU Affero GPL version 3.

```
Copyright (C) 2010 Dennis Schwerdel, University of Kaiserslautern
```

Design, Layout and Logo

The design, layout and logo have been developed by University of Würzburg and University of Kaiserslautern on behalf of the German-Lab project.

```
Copyright (C) 2010 University of Würzburg, University of Kaiserslautern
```

Icons

Licenses of icons are listed in /web/tomato/img/sources.

Obtaining the code

All the code can be obtained via git from:

```
git://github.com/dswd/ToMaTo.git
```

License Texts

GNU Affero GPL version 3

```
                    GNU AFFERO GENERAL PUBLIC LICENSE
                       Version 3, 19 November 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU Affero General Public License is a free, copyleft license for
software and other kinds of works, specifically designed to ensure
cooperation with the community in the case of network server software.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
our General Public Licenses are intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
```

```
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  Developers that use our General Public Licenses protect your rights
with two steps: (1) assert copyright on the software, and (2) offer
you this License which gives you legal permission to copy, distribute
and/or modify the software.

  A secondary benefit of defending all users' freedom is that
improvements made in alternate versions of the program, if they
receive widespread use, become available for other developers to
incorporate.  Many developers of free software are heartened and
encouraged by the resulting cooperation.  However, in the case of
software used on network servers, this result may fail to come about.
The GNU General Public License permits making a modified version and
letting the public access it on a server without ever releasing its
source code to the public.

  The GNU Affero General Public License is designed specifically to
ensure that, in such cases, the modified source code becomes available
to the community.  It requires the operator of a network server to
provide the source code of the modified version running there to the
users of that server.  Therefore, public use of a modified version, on
a publicly accessible server, gives the public access to the source
code of the modified version.

  An older license, called the Affero General Public License and
published by Affero, was designed to accomplish similar goals.  This is
a different license, not a version of the Affero GPL, but Affero has
released a new version of the Affero GPL which permits relicensing under
this license.

  The precise terms and conditions for copying, distribution and
modification follow.

                       TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU Affero General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
```

```
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.

  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
```

```
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.
```

```
  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

    a) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by the
    Corresponding Source fixed on a durable physical medium
    customarily used for software interchange.

    b) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by a
    written offer, valid for at least three years and valid for as
    long as you offer spare parts or customer support for that product
    model, to give anyone who possesses the object code either (1) a
    copy of the Corresponding Source for all the software in the
    product that is covered by this License, on a durable physical
    medium customarily used for software interchange, for a price no
    more than your reasonable cost of physically performing this
    conveying of source, or (2) access to copy the
    Corresponding Source from a network server at no charge.
```

```
    c) Convey individual copies of the object code with a copy of the
    written offer to provide the Corresponding Source.  This
    alternative is allowed only occasionally and noncommercially, and
    only if you received the object code with such an offer, in accord
    with subsection 6b.

    d) Convey the object code by offering access from a designated
    place (gratis or for a charge), and offer equivalent access to the
    Corresponding Source in the same way through the same place at no
    further charge.  You need not require recipients to copy the
    Corresponding Source along with the object code.  If the place to
    copy the object code is a network server, the Corresponding Source
    may be on a different server (operated by you or a third party)
    that supports equivalent copying facilities, provided you maintain
    clear directions next to the object code saying where to find the
    Corresponding Source.  Regardless of what server hosts the
    Corresponding Source, you remain obligated to ensure that it is
    available for as long as needed to satisfy these requirements.

    e) Convey the object code using peer-to-peer transmission, provided
    you inform other peers where the object code and Corresponding
    Source of the work are being offered to the general public at no
    charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
```

```
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
```

```
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
```

```
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
```

```
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Remote Network Interaction; Use with the GNU General Public License.

  Notwithstanding any other provision of this License, if you modify the
Program, your modified version must prominently offer all users
interacting with it remotely through a computer network (if your version
supports such interaction) an opportunity to receive the Corresponding
```

```
Source of your version by providing access to the Corresponding Source
from a network server at no charge, through some standard or customary
means of facilitating copying of software.  This Corresponding Source
shall include the Corresponding Source for any work covered by version 3
of the GNU General Public License that is incorporated pursuant to the
following paragraph.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the work with which it is combined will remain governed by version
3 of the GNU General Public License.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU Affero General Public License from time to time.  Such new versions
will be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU Affero General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU Affero General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU Affero General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
```

```
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If your software can interact with users remotely through a computer
network, you should also make sure that it provides a way for users to
get its source.  For example, if your program is a web application, its
interface could display a "Source" link that leads users to an archive
of the code.  There are many ways you could offer source, and different
solutions will be better for different programs; see section 13 for the
specific requirements.

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU AGPL, see
<http://www.gnu.org/licenses/>.
```

Creative Commons Attribution 2.5 License


```
Creative Commons Legal Code

Attribution 3.0 Unported

    CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
    LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
    ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
    INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES
    REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR
    DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE
COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY
COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS
AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE
TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY
BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS
CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND
CONDITIONS.

1. Definitions

 a. "Adaptation" means a work based upon the Work, or upon the Work and
    other pre-existing works, such as a translation, adaptation,
    derivative work, arrangement of music or other alterations of a
    literary or artistic work, or phonogram or performance and includes
    cinematographic adaptations or any other form in which the Work may be
    recast, transformed, or adapted including in any form recognizably
    derived from the original, except that a work that constitutes a
    Collection will not be considered an Adaptation for the purpose of
    this License. For the avoidance of doubt, where the Work is a musical
    work, performance or phonogram, the synchronization of the Work in
    timed-relation with a moving image ("synching") will be considered an
    Adaptation for the purpose of this License.
 b. "Collection" means a collection of literary or artistic works, such as
    encyclopedias and anthologies, or performances, phonograms or
    broadcasts, or other works or subject matter other than works listed
    in Section 1(f) below, which, by reason of the selection and
    arrangement of their contents, constitute intellectual creations, in
    which the Work is included in its entirety in unmodified form along
    with one or more other contributions, each constituting separate and
    independent works in themselves, which together are assembled into a
    collective whole. A work that constitutes a Collection will not be
    considered an Adaptation (as defined above) for the purposes of this
    License.
 c. "Distribute" means to make available to the public the original and
    copies of the Work or Adaptation, as appropriate, through sale or
    other transfer of ownership.
 d. "Licensor" means the individual, individuals, entity or entities that
    offer(s) the Work under the terms of this License.
 e. "Original Author" means, in the case of a literary or artistic work,
    the individual, individuals, entity or entities who created the Work
```

```
   or if no individual or entity can be identified, the publisher; and in
   addition (i) in the case of a performance the actors, singers,
   musicians, dancers, and other persons who act, sing, deliver, declaim,
   play in, interpret or otherwise perform literary or artistic works or
   expressions of folklore; (ii) in the case of a phonogram the producer
   being the person or legal entity who first fixes the sounds of a
   performance or other sounds; and, (iii) in the case of broadcasts, the
   organization that transmits the broadcast.
f. "Work" means the literary and/or artistic work offered under the terms
   of this License including without limitation any production in the
   literary, scientific and artistic domain, whatever may be the mode or
   form of its expression including digital form, such as a book,
   pamphlet and other writing; a lecture, address, sermon or other work
   of the same nature; a dramatic or dramatico-musical work; a
   choreographic work or entertainment in dumb show; a musical
   composition with or without words; a cinematographic work to which are
   assimilated works expressed by a process analogous to cinematography;
   a work of drawing, painting, architecture, sculpture, engraving or
   lithography; a photographic work to which are assimilated works
   expressed by a process analogous to photography; a work of applied
   art; an illustration, map, plan, sketch or three-dimensional work
   relative to geography, topography, architecture or science; a
   performance; a broadcast; a phonogram; a compilation of data to the
   extent it is protected as a copyrightable work; or a work performed by
   a variety or circus performer to the extent it is not otherwise
   considered a literary or artistic work.
g. "You" means an individual or entity exercising rights under this
   License who has not previously violated the terms of this License with
   respect to the Work, or who has received express permission from the
   Licensor to exercise rights under this License despite a previous
   violation.
h. "Publicly Perform" means to perform public recitations of the Work and
   to communicate to the public those public recitations, by any means or
   process, including by wire or wireless means or public digital
   performances; to make available to the public Works in such a way that
   members of the public may access these Works from a place and at a
   place individually chosen by them; to perform the Work to the public
   by any means or process and the communication to the public of the
   performances of the Work, including by public digital performance; to
   broadcast and rebroadcast the Work by any means including signs,
   sounds or images.
i. "Reproduce" means to make copies of the Work by any means including
   without limitation by sound or visual recordings and the right of
   fixation and reproducing fixations of the Work, including storage of a
   protected performance or phonogram in digital form or other electronic
   medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce,
limit, or restrict any uses free from copyright or rights arising from
limitations or exceptions that are provided for in connection with the
copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License,
Licensor hereby grants You a worldwide, royalty-free, non-exclusive,
perpetual (for the duration of the applicable copyright) license to
exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more
```

```
   Collections, and to Reproduce the Work as incorporated in the
   Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation,
   including any translation in any medium, takes reasonable steps to
   clearly label, demarcate or otherwise identify that changes were made
   to the original Work. For example, a translation could be marked "The
   original work was translated from English to Spanish," or a
   modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated
   in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

     i. Non-waivable Compulsory License Schemes. In those jurisdictions in
        which the right to collect royalties through any statutory or
        compulsory licensing scheme cannot be waived, the Licensor
        reserves the exclusive right to collect such royalties for any
        exercise by You of the rights granted under this License;
    ii. Waivable Compulsory License Schemes. In those jurisdictions in
        which the right to collect royalties through any statutory or
        compulsory licensing scheme can be waived, the Licensor waives the
        exclusive right to collect such royalties for any exercise by You
        of the rights granted under this License; and,
   iii. Voluntary License Schemes. The Licensor waives the right to
        collect royalties, whether individually or, in the event that the
        Licensor is a member of a collecting society that administers
        voluntary licensing schemes, via that society, from any exercise
        by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now
known or hereafter devised. The above rights include the right to make
such modifications as are technically necessary to exercise the rights in
other media and formats. Subject to Section 8(f), all rights not expressly
granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made
subject to and limited by the following restrictions:

 a. You may Distribute or Publicly Perform the Work only under the terms
    of this License. You must include a copy of, or the Uniform Resource
    Identifier (URI) for, this License with every copy of the Work You
    Distribute or Publicly Perform. You may not offer or impose any terms
    on the Work that restrict the terms of this License or the ability of
    the recipient of the Work to exercise the rights granted to that
    recipient under the terms of the License. You may not sublicense the
    Work. You must keep intact all notices that refer to this License and
    to the disclaimer of warranties with every copy of the Work You
    Distribute or Publicly Perform. When You Distribute or Publicly
    Perform the Work, You may not impose any effective technological
    measures on the Work that restrict the ability of a recipient of the
    Work from You to exercise the rights granted to that recipient under
    the terms of the License. This Section 4(a) applies to the Work as
    incorporated in a Collection, but this does not require the Collection
    apart from the Work itself to be made subject to the terms of this
    License. If You create a Collection, upon notice from any Licensor You
    must, to the extent practicable, remove from the Collection any credit
    as required by Section 4(b), as requested. If You create an
    Adaptation, upon notice from any Licensor You must, to the extent
```

```
    practicable, remove from the Adaptation any credit as required by
    Section 4(b), as requested.
 b. If You Distribute, or Publicly Perform the Work or any Adaptations or
    Collections, You must, unless a request has been made pursuant to
    Section 4(a), keep intact all copyright notices for the Work and
    provide, reasonable to the medium or means You are utilizing: (i) the
    name of the Original Author (or pseudonym, if applicable) if supplied,
    and/or if the Original Author and/or Licensor designate another party
    or parties (e.g., a sponsor institute, publishing entity, journal) for
    attribution ("Attribution Parties") in Licensor's copyright notice,
    terms of service or by other reasonable means, the name of such party
    or parties; (ii) the title of the Work if supplied; (iii) to the
    extent reasonably practicable, the URI, if any, that Licensor
    specifies to be associated with the Work, unless such URI does not
    refer to the copyright notice or licensing information for the Work;
    and (iv) , consistent with Section 3(b), in the case of an Adaptation,
    a credit identifying the use of the Work in the Adaptation (e.g.,
    "French translation of the Work by Original Author," or "Screenplay
    based on original Work by Original Author"). The credit required by
    this Section 4 (b) may be implemented in any reasonable manner;
    provided, however, that in the case of a Adaptation or Collection, at
    a minimum such credit will appear, if a credit for all contributing
    authors of the Adaptation or Collection appears, then as part of these
    credits and in a manner at least as prominent as the credits for the
    other contributing authors. For the avoidance of doubt, You may only
    use the credit required by this Section for the purpose of attribution
    in the manner set out above and, by exercising Your rights under this
    License, You may not implicitly or explicitly assert or imply any
    connection with, sponsorship or endorsement by the Original Author,
    Licensor and/or Attribution Parties, as appropriate, of You or Your
    use of the Work, without the separate, express prior written
    permission of the Original Author, Licensor and/or Attribution
    Parties.
 c. Except as otherwise agreed in writing by the Licensor or as may be
    otherwise permitted by applicable law, if You Reproduce, Distribute or
    Publicly Perform the Work either by itself or as part of any
    Adaptations or Collections, You must not distort, mutilate, modify or
    take other derogatory action in relation to the Work which would be
    prejudicial to the Original Author's honor or reputation. Licensor
    agrees that in those jurisdictions (e.g. Japan), in which any exercise
    of the right granted in Section 3(b) of this License (the right to
    make Adaptations) would be deemed to be a distortion, mutilation,
    modification or other derogatory action prejudicial to the Original
    Author's honor and reputation, the Licensor will waive or not assert,
    as appropriate, this Section, to the fullest extent permitted by the
    applicable national law, to enable You to reasonably exercise Your
    right under Section 3(b) of this License (right to make Adaptations)
    but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR
OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY
KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE,
INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY,
FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF
LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS,
WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION
```

```
OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE
LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR
ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES
ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS
BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

 a. This License and the rights granted hereunder will terminate
    automatically upon any breach by You of the terms of this License.
    Individuals or entities who have received Adaptations or Collections
    from You under this License, however, will not have their licenses
    terminated provided such individuals or entities remain in full
    compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will
    survive any termination of this License.
 b. Subject to the above terms and conditions, the license granted here is
    perpetual (for the duration of the applicable copyright in the Work).
    Notwithstanding the above, Licensor reserves the right to release the
    Work under different license terms or to stop distributing the Work at
    any time; provided, however that any such election will not serve to
    withdraw this License (or any other license that has been, or is
    required to be, granted under the terms of this License), and this
    License will continue in full force and effect unless terminated as
    stated above.

8. Miscellaneous

 a. Each time You Distribute or Publicly Perform the Work or a Collection,
    the Licensor offers to the recipient a license to the Work on the same
    terms and conditions as the license granted to You under this License.
 b. Each time You Distribute or Publicly Perform an Adaptation, Licensor
    offers to the recipient a license to the original Work on the same
    terms and conditions as the license granted to You under this License.
 c. If any provision of this License is invalid or unenforceable under
    applicable law, it shall not affect the validity or enforceability of
    the remainder of the terms of this License, and without further action
    by the parties to this agreement, such provision shall be reformed to
    the minimum extent necessary to make such provision valid and
    enforceable.
 d. No term or provision of this License shall be deemed waived and no
    breach consented to unless such waiver or consent shall be in writing
    and signed by the party to be charged with such waiver or consent.
 e. This License constitutes the entire agreement between the parties with
    respect to the Work licensed here. There are no understandings,
    agreements or representations with respect to the Work not specified
    here. Licensor shall not be bound by any additional provisions that
    may appear in any communication from You. This License may not be
    modified without the mutual written agreement of the Licensor and You.
 f. The rights granted under, and the subject matter referenced, in this
    License were drafted utilizing the terminology of the Berne Convention
    for the Protection of Literary and Artistic Works (as amended on
    September 28, 1979), the Rome Convention of 1961, the WIPO Copyright
    Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996
    and the Universal Copyright Convention (as revised on July 24, 1971).
    These rights and subject matter take effect in the relevant
    jurisdiction in which the License terms are sought to be enforced
```

```
    according to the corresponding provisions of the implementation of
    those treaty provisions in the applicable national law. If the
    standard suite of rights granted under applicable copyright law
    includes additional rights not granted under this License, such
    additional rights are deemed to be included in the License; this
    License is not intended to restrict the license of any rights under
    applicable law.


Creative Commons Notice

    Creative Commons is not a party to this License, and makes no warranty
    whatsoever in connection with the Work. Creative Commons will not be
    liable to You or any party on any legal theory for any damages
    whatsoever, including without limitation any general, special,
    incidental or consequential damages arising in connection to this
    license. Notwithstanding the foregoing two (2) sentences, if Creative
    Commons has expressly identified itself as the Licensor hereunder, it
    shall have all rights and obligations of Licensor.

    Except for the limited purpose of indicating to the public that the
    Work is licensed under the CCPL, Creative Commons does not authorize
    the use by either party of the trademark "Creative Commons" or any
    related trademark or logo of Creative Commons without the prior
    written consent of Creative Commons. Any permitted use will be in
    compliance with Creative Commons' then-current trademark usage
    guidelines, as may be published on its website or otherwise made
    available upon request from time to time. For the avoidance of doubt,
    this trademark restriction does not form part of this License.

    Creative Commons may be contacted at http://creativecommons.org/.
```


Missing topics

General

- Usage calculation
- Accounts & Providers
- Logging

Backend

- **API**
  - Host
- Installation
- Configuration
- Permissions

Hostmanager

- **API**
  - Connections

Frontends

CLI

- Usage
- Parameters
- Examples

Webfrontend

- Installation
- Configuration

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

Index