Jeshka Documentation

Release 1.0

Jeshka

Dec 17, 2018

Contents

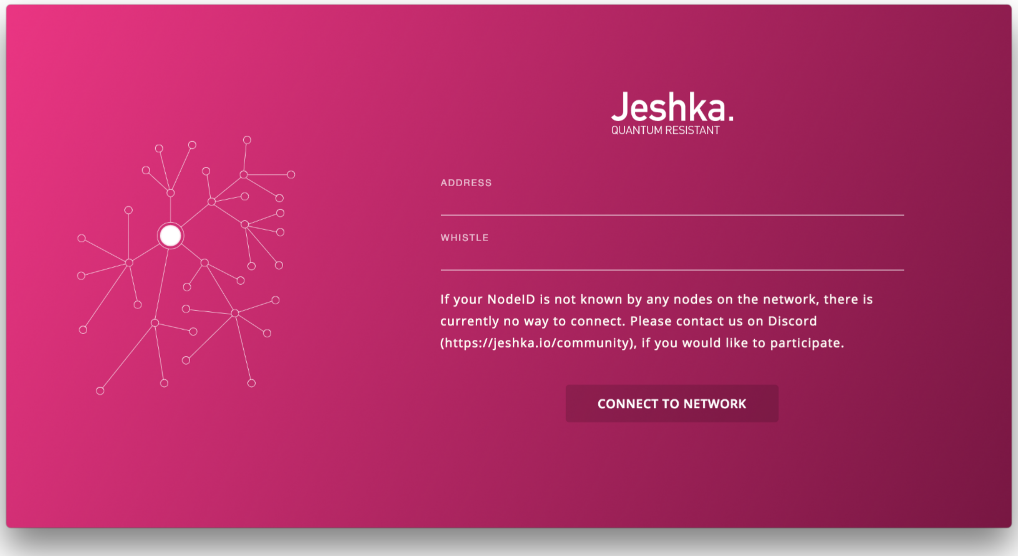
Contents:

CHAPTER 1

Applications

1.1 Jeshka Desktop

1.1.1 Getting Started



If your NodeID is not known by any nodes on the network, there is currently no way to connect. Please contact us on Discord (https://jeshka.io/community), if you would like to participate.

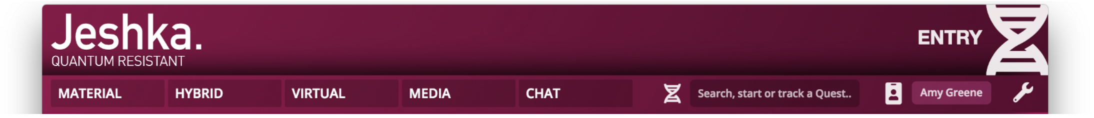
Jeshka Desktop is an interactive experience for all resources available to us in the network. The application seeks to deliver a fresh perspective on life, no matter where you're coming from.

The application is highly gamified, all aspects of life are abstracted to a decentralized and distributed digital consciousness. You can choose to encrypt your interactions.

The entry into the application creates a fictional situation where the user keeps forgetting that life used to be very different with an advanced version of Jeshka, but there was a terrible global disaster and almost all of Jeshka was removed from the knowledge of humanity. Now it's on us, the population of the planet, to rebuild it. The participants only saved a piece of paper with their address and whistle from the disaster.

Release: December 1st, 2018

1.1.2 Control Center



We can skip to other plugins and apps from here. We can start, search or track Quests. We can change account and application preferences and settings.

1.1.3 Quest Machine

Every piece of information that leaves our devices is wrapped up in something we call a Quest. This can be a simple text message, a financial transaction or references to an extremely complicated problem with many different moving parts that you are looking to get solved. Not just theoretical problems that is, we can even start Quests to build and furnish houses! Imagine a Quest like a box with a set of instructions and in some cases with many different (financial) rewards that whoever helps to resolve this Quest or parts of it will be able to remove from my box. Well, maybe this example is not the best. If you can think of a better one, please let me know! ;-)

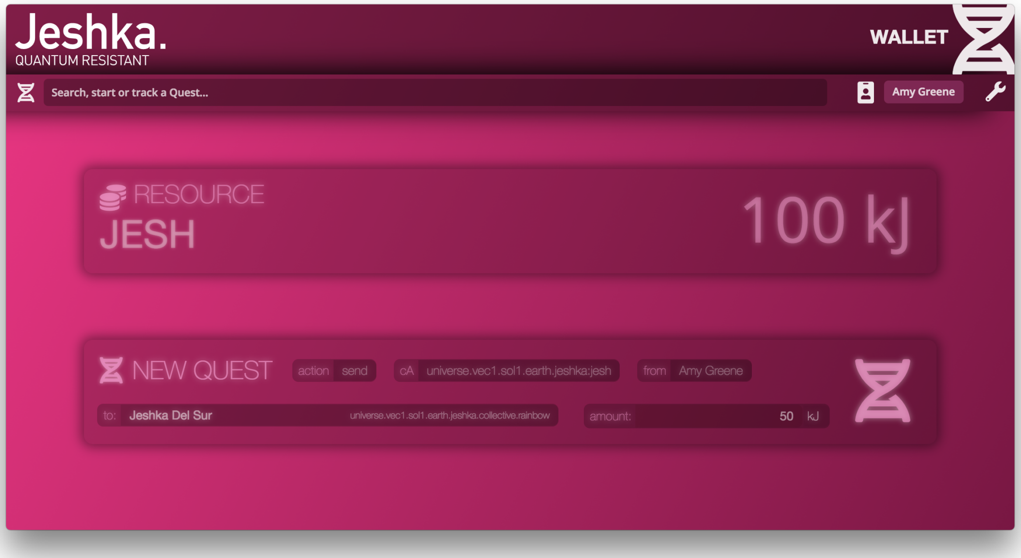
The Quest Machine is a Jeshka Desktop App feature that interactively allows peers to start Quests on the network. It's the core functionality that all the other functionality on Jeshka seeks to support.

The Quest Machine covers the entire flow of dialogs for the creation and management of Quests. It includes the Quest Manager and the Quest Explorer to hunt after unresolving Quests.

1.1.4 Multi Wallet

This feature is an easy to use space that shows our full balances for *all currencies* that we have registered on Jeshka. Internally manages the resource type / contract group **finc** and emits audiovisual events on changes.

This feature is available on almost every screen of the App in some shape or form.

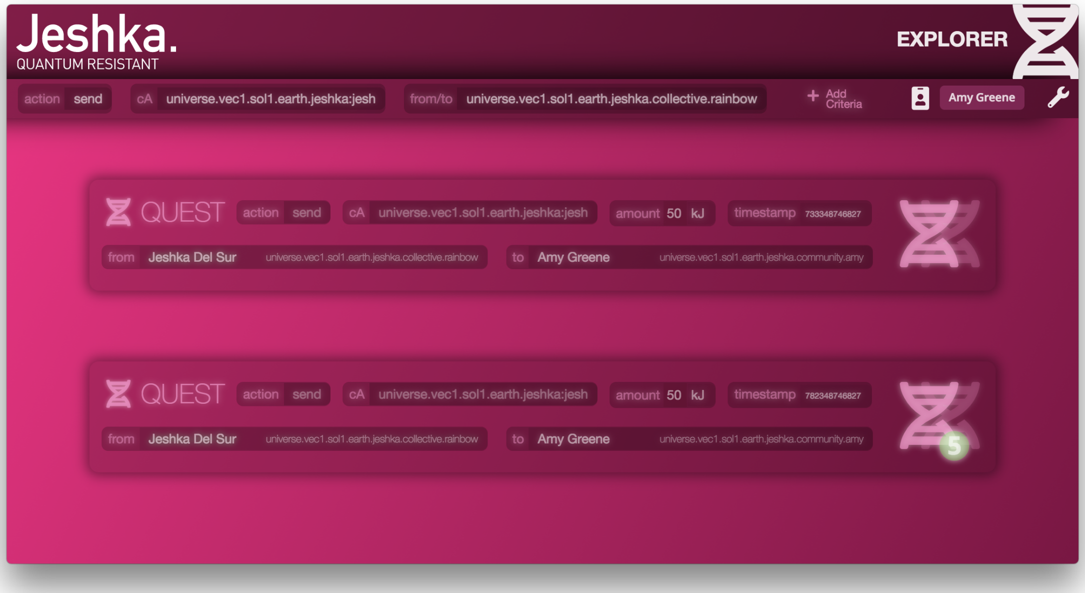


RESOURCE
JESH
100 kJ

NEW QUEST
action send
cA universe.vec1.sol1.earth.jeshka:jesh
from Amy Greene
to: Jeshka Del Sur universe.vec1.sol1.earth.jeshka.collective.rainbow
amount: 50 kJ

1.1.5 Jeshka Quest Explorer

Search and track Quests, check on their status, location, explore streams in certain channels, etc.



action send
cA universe.vec1.sol1.earth.jeshka:jesh
from/to universe.vec1.sol1.earth.jeshka.collective.rainbow

QUEST
action send
cA universe.vec1.sol1.earth.jeshka:jesh
amount 50 kJ
timestamp 733348746827
from Jeshka Del Sur universe.vec1.sol1.earth.jeshka.collective.rainbow
to Amy Greene universe.vec1.sol1.earth.jeshka.community.amy

QUEST
action send
cA universe.vec1.sol1.earth.jeshka:jesh
amount 50 kJ
timestamp 782348746827
from Jeshka Del Sur universe.vec1.sol1.earth.jeshka.collective.rainbow
to Amy Greene universe.vec1.sol1.earth.jeshka.community.amy

1.1.6 Text Channels

Decentralized IRC chat, we can import channel specific plugins. Domains can choose Roles for specific addresses and group their channels.

Internally manages the resource type **text** and emits audiovisual events on changes.

The main Jeshka Desktop App uses components to group user activity. This feature is available on almost every screen of the App.

1.1.7 Media Browser

For streaming and archived media objects, images, video, audio.

1.1.8 Consensus Module

Front End qManager

We can use the consensus plugin for a variety of Assembly activities. As a component, it is visible across the application.

1.1.9 Research Center

We can use Jeshka to power our research projects.

1.2 Jeshka Desktop Plugins

While the plugins and libraries might seem isolated, they all contact each other and exchange information within your system. So every part of the Jeshka application knows exactly what's relevant for them.

Unlocking achievements of multiple achievements together can create meta combinations of achievements.

1.2.1 Sponsorships

Sponsor projects on the network with rewards you don't need. Think decentralized crowdfund.

1.2.2 Distributed Git Explorer and Editor

1.2.3 Virtual Space

The virtual **Reward Space** allows most virtual interactions, for example streaming of oracles, unlocking achievements for particular resources, inter-human communication, etc.

Plugin enters with earth view of live network activity.

There are no old-timey prices on Jeshka, we have Rewards in our system the total amount of which is the equivalent to the total estimated energy that is needed to resolve all Quests on the network in a given timeframe. The rewards are calculated by assemblies and all relevant peers on the network, they keep changing.

1.2.4 Material Space

A multi-purpose open space for new and past physical objects (including grocery items) that as part of the specific Quest can also be physically delivered to us. Achievements of individual resources can be autonomously unlocked by the Quest Machine if needed for another Quest.

The nature of the Quests behind these interactions dictates that you can not set old-timey prices to individual goods, you can set a minimum conversation result, but the final exchange rate will be a consensus result with everyone who is offering the same resource under the same conditions and the relevant people who are interested in achieving/unlocking this resource!

1.2.5 Capabilities

An interactive explorer for our currently calculated individual capabilities, which leads to a portal to the Jeshka University module, where we can expand our capabilities while resolving Quests on the Quest Machine.

1.2.6 Combinations

An interactive explorer for combinations of the above, these are meta types and no actual resources, they're for example being used for a Quest that would read "Get me from here to here in less than 20 minutes."

1.2.7 JesHub

A simple platform for sharing of physical spaces. A JesHub domain keeps track of details about their location, their devices and such, also offers JesHub internal encrypted text message channels.

We get dialogs for Quests from others who are looking either to visit or stay at a JesHub or who need storage space, we can set our Jeshka domain to decide about matters in this domain on a detailed level or in full consensus with all peers in this domain.

Internally a JesHub is a resource.

1.2.8 Health

An interactive space for health related subjects. All features currently undisclosed.

1.3 Headless JDK Node

Currently only serves assembly functions. Contact the community if you are interested in this project.

1.4 Network Management

1.4.1 QManager

1.5 Libraries

1.5.1 Jeshka JS Lib

Release Date: December 1st, 2018.

If you want to build an app for Jeshka, don't think twice! We're maintaining a JavaScript library that enables you to start a Quest in 5 easy steps. Just use:

jeshka = new JeshkaNode(); jeshka.startQManager(); jeshka.unlockWallet(seed Whistle); quest = new JeshkaQuest(); and put together your quest with easy setters and getters.

When you're ready, let Jeshka whistle it jeshka.whistle(quest) for you, if you use this method, Jeshka will automatically sign and distribute the Quest. Jeshka is asynchronous, you can await jeshka.whistle(quest). You can also build a Quest that listens to a resource (stream) on the network!

Jeshka is non-profit and makes life hard for commercial enterprises. Jeshka is a non-profit organization; we're currently in the process of registering Jeshka as a Foundation in the Netherlands. All parts of Jeshka are released under the AGPL v3.0 license, that means any service that uses Jeshka has to publish the relevant source code.

Jeshka JavaScript Library on npm.

Jeshka JavaScript Library on GitHub.

CHAPTER 2

Devices

2.1 Jeshka Mesh Router

The Jeshka Mesh Router creates an access point to the Jeshka Quest Network, routes and resolves incoming Quests.

Build yourself or order from https://store.jeshka.io/jmr-1

2.2 Jeshka Smart Card

Use the Jeshka Smart Card to store your whistle(s).

Build yourself or order from https://store.jeshka.io/jsc-1

2.3 Jeshka Whistle Chip

Opt in for the fabulous Jeshka Whistle Chip to store your whistle(s).

Warning: Chip implants could cause infections or reactions in the body’s immune system.

Build yourself or order from https://store.jeshka.io/jw-1

2.4 Jeshka Animal Chip

Build yourself or order from https://store.jeshka.io/ja-1

CHAPTER 3

Protocol

This proposed solution makes use of peer to peer networking, cryptographic capabilities and machine learning technologies to create a feeless, trustless and fully decentralized ecosystem of services that enables optimal fragmentation and reuse of work performed by humans and machines across projects and departments. While providing a constantly optimized incentive for fair compensation, allowing for voluntarily unpaid participation and embedded learning experiences. The computer can be used as a tool to teach and reward people, rather than to exploit them.

3.1 Quest

Every piece of information that leaves our devices is wrapped up in something we call a Quest. This can be a simple text message, a financial transaction or references to an extremely complicated problem with many different moving parts that you are looking to get solved. Not just theoretical problems that is, we can even start Quests to build and furnish houses! Imagine a Quest like a box with a set of instructions and in some cases with many different (financial) rewards that whoever helps to resolve this Quest or parts of it will be able to remove from my box. Well, maybe this example is not that great. If you can think of a better one, please let me know! ;-)

A regular financial transaction on Jeshka, “bring this amount of this currency that I own to this other person” is always a Quest signed with a unique one time whistle and your unique and permanent address.

We call the entire existence of information that gets passed on to other peers in a single call ***a Quest Bundle***.

Representatives should check if the quest will get resolved, i.e. if we have permission to listen on a certain channel, request a certain type of information or commit a certain transaction into a channel before routing our quest. It guards the other domains from spam traffic, requesting information that’s not available. If a representative doesn’t check and blindly forwards the request, the representative in the other domain might tell all the representatives on our domain that he is not doing his job and our representative might lose its representative role on the network.

3.1.1 The five-second rule.

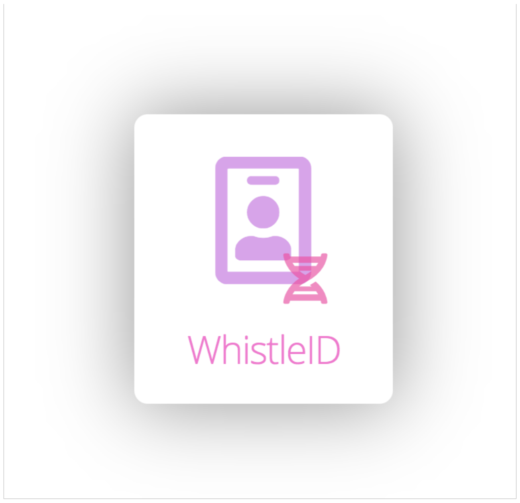
If you start a Quest with a target that’s accessible within your economic cluster/domain, and a financial transaction or text message hasn’t resolved (is not confirmed) within 5 seconds*, we consider it failed or rejected. With a modern laptop, a Quest for a financial transaction usually resolves within 500–2500ms within a given domain/economic cluster.

** refers to the timeframe during which a Quest is considered hot. A Quest is considered hot, starting when it has reached the first peer we connect to. We expect simple Quests to resolve within a maximum of 5000ms from now. Once a Quest has resolved anywhere on the Quest Network it is considered resolved and the countdown stops. It may still take some time until we are informed that it was resolved.*

If for some reason a Basic Quest keeps getting rejected because it exceeds the five seconds, please contact us, this is likely a bug.

3.2 Whistles

3.2.1 WhistleID



Our WhistleID is a collection of important data about us. It is connected to all the whistles we generate to sign Quests on network. While we interact on Jeshka, we collect signatures that prove our WhistleID is indeed ours. The most important piece of data in our WhistleID is our unique and human readable permanent address.

Our address inherently describes how we can be found if someone loses sight of us and helps to verify our unique one time whistle. Our addresses start with a list of all the parent domains down to our Home Domain from left to right.

3.2.2 Seed Whistle

Our signature scheme is inspired by the way dolphins identify each other. Whistles are the core for authentication and authorization on Jeshka. We all have different seed whistles that nobody else in this universe can ever forge. A seed whistle on Jeshka is a 174 character string of random letters A-Z and the number 9. We can derive everything we'll ever need to participate in the network from this seed whistle. For instance, we sign Quests with quest specific private whistles that we derive from our original seed whistle. It is the first thing we do to prove to everyone in the universe that our signatures are authentic and that by consensus on the network the author of the Quest has to be the WhistleID connected to our permanent address, i.e. us. We can also derive whistles that help us decrypt messages and files. The seed whistle is in the same format as IOTA seeds, so the first 81 characters can just be my IOTA seed, etc, etc. Literally every key we'll ever need can be derived from this seed whistle.

174 chars:

128 chars crypto seed 6 chars pin 28 chars my aes pass (for example to decrypt my ssl whistles for other domains that we gave to our individual home domains) 12 chars home whistle (to get ssl keys from home domain)

3.2.3 Private Whistle

Quests are signed with unique asymmetric one time private whistles that we derive from our seed whistle.

3.2.4 qWhistle

Quests contain a field to announce one upcoming unique asymmetric one time public quest whistle. It is the public counterpart of the private whistle that is used to sign a Quest/transaction.

It can be used to verify the authenticity and origin of an upcoming Quest.

If a Quest is signed with a whistle/key that hasn't been announced in a qWhistle field before, it's invalid.

In other words, the qWhistle field contains the public whistle (key) that has to be used to verify the next or another upcoming Quest in this specific channel or zone.

3.2.5 Home Whistle

A home whistle is used by members of a given domain to roughly identify the origin of an incoming connection, of course this home whistle can be intercepted and used by others, who will then be rejected by the server, because they can't produce the right public whistles. It serves as a good first gateway mechanism, helps others to get a rough idea of who we are and can be changed per our request. For example if it has gotten compromised.

3.2.6 SSL Whistle

OTP to establish SSL connections.

3.2.7 Join Whistle

When connecting to a new flow, we tell our swarm that we are joining this new flow with a Join Quest. If anyone asks around in our swarm, our Join Whistle holds a public whistle as content that other flows are asking about. If unsuccessful the logical conclusion is that the public whistle in combination with the address does not exist, very likely that the particular Quest Bundle is a forgery.

3.3 Addresses

We all identify with a unique permanent address and a unique one time whistle that changes with every Quest. Our address describes how we can be found if someone loses sight of us and helps to verify our unique one time whistle. Our addresses start with a list of all the parent domains down to our Home Domain from left to right. For me this is:

jeshka://universe.vec1.sol1.earth.jeshka.collective.rainbow

and since we are all on the same planet here, most Jeshka apps allow a shorthand version and internally compute the full address, like so: jeshka.collective.rainbow or rainbow@jeshka.collective

This is my receiving address for all Quests directed at me. So people can send anything from encrypted text messages, financial transactions to terabytes of data to my address. It's not to be mixed up with old-timey email addresses.

protocol part jeshka://

home domain part universe.vec1.sol1.earth.jeshka.collective

peer part .rainbow

To identify resources, for example in the cA field we see addresses that refer to resources more than individuals, let's say we're looking up the currency JESH, the full address would be jeshka://universe.vec1.sol1.earth.jeshka:jesh.

Whistleid addresses are relative in places like these, so if I want to look up my balance, I have to: jeshka://universe.vec1.sol1.earth.jeshka:jesh/collective.rainbow (/relative whistle id address)

If we want to look up all balances for JESH, we have to look up the peer map: jeshka://universe.vec1.sol1.earth.jeshka:jesh.p

If we want to look up a specific Quest in a specific channel, we have to look up: jeshka://universe.vec1.sol1.earth.jeshka:jesh::7382e3a8932eb (::qHash)

If we want to look up all Quests in a specific channel, we have to go through the Quest log: jeshka://universe.vec1.sol1.earth.jeshka:jesh.qlog

3.4 Domains

Domains can only be upper and lowercase characters. Minimum 5 Characters. 4 Characters reserved for network functionalities. Once a domain is registered on the network, it's yours.

Domains can also keep track of their resources and respond to Quests automatically, manually or let Jeshka decide on a detailed level, i.e. we can request a collection of Jeshka domains to make certain decisions, we can choose our own software to come to individual certain decisions for resources in our domain or we can even manually handle *all* Quests ourselves.

3.4.1 Domain Peers

A peer in a given domain connects to their domain assembly.

Their domain assembly will relay all the information from the domain they represent through the JeshkaNet and the Internet.

When a peer connects to its domain representatives through the internet, they have to trust that behind the https address they find current representative

3.5 Consensus

Every domain can choose their own consensus strategies for different resources and change them over time.

Strong Consistency: Strong consistency means that the transactions in a given resource channel are only considered confirmed, once every representative confirmed them.

In centralized networks the central authority owns or controls all representatives.

3.6 Assemblies

How do peers on Jeshka reach consensus? While Jeshka allows us to establish blazing fast and highly secure peer to peer connections that can not be intercepted, a Quest is never resolved by just one person. We want to make sure the result is what we wanted, that our Quest hasn't been altered and that nobody is trying to remove a reward from the box without having done their part correctly. Some people are better at some things than others and for some parts of our specific Quest, AI may be more suitable. If it doesn't take the entire reward to resolve our Quest, we're always happy to get some back and if the result exceeds all of our expectations, we're free to tip the remainder to the people who did the best job on the entire Quest.

For all of that Jeshka uses Assemblies. The dictionary definition of an Assembly is "a group of people gathered together in one place for a common purpose." and that's just what an Assembly is on Jeshka. An Assembly is a very diverse group of participants, some who are proven to be very good at what they are doing, some have just started out and are resolving parts of our Quest as they are advancing on their learning curve. A Quest can be resolved by different Assemblies for different parts of it or if it involves moving objects across a large geographical distance, for different legs of the journey. An Assembly is a group of people, who are helping us assemble whatever our Quest is about on whichever network is the most suitable.

Every domain and resource has their own quorum strategy and is represented by a series of quorum decisions. The domain assemblies are usually elected from the pool of all peers in a given domain.

A domain representative has to keep records of all of the domain's and it's subdomains and peers resources.

The representatives can choose their own software to connect to the network as long as they follow the communication and encryption protocols.

3.7 Load Balancing

If we are not invited to open connection channels with the representatives in a domain, we are accessing the resources of, we can ask our domain representatives to relay the information for us. They will for example listen on all changes there and update us and all the others who are listening from them about a given resource. These representatives are usually the members of a dynamically formed Assembly for this purpose.

3.8 Connection Channels

Jeshka is available on the internet, to establish a trusted connection, the Jeshka node uses a list of sources (that we can edit) to achieve consensus over who is representative for what domain at the moment.

3.8.1 Initialization

We connect to all the representatives in our home domain, requesting our SSL Whistle keychain, using our peer address and the Home Whistle. The keychain is an aes encrypted JSON Object that holds all of our personal SSL Whistles for different domains. It's on us to keep it updated.

Port 442

3.8.2 Keychain Access

If we have the fitting SSL Whistle, we can use it to establish a direct connection channel with either the representatives of a given domain we are trying to access or with the representatives from our home domain.

3.9 Distributed Gateways

If a given resource domain or domain is not available for us to connect to directly, we ask a dynamically formed assembly of peers to relay the information for us.

What we want these representatives to route to a different network, we call a *Quest*.

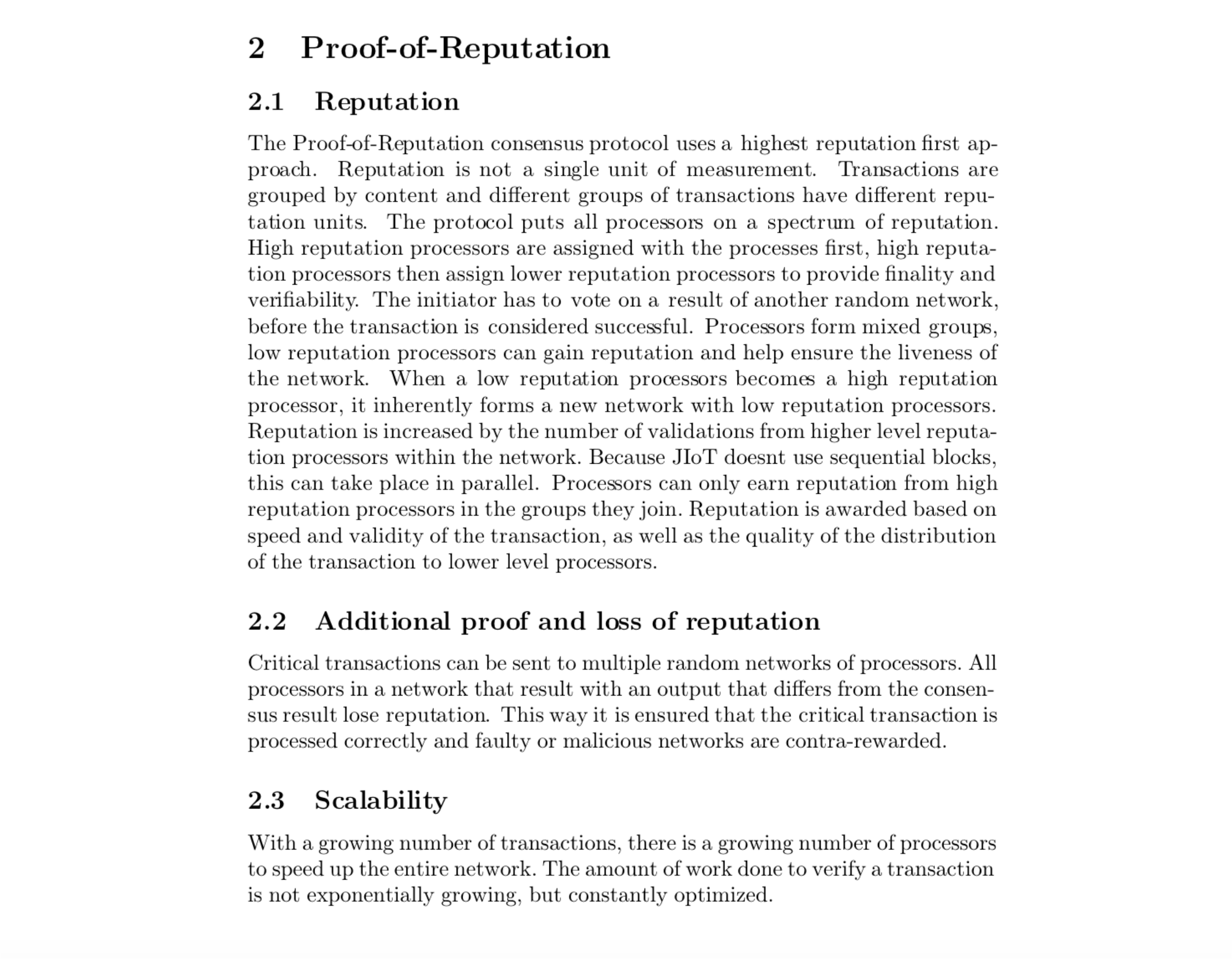
3.10 Network Plugins

Jeshka is network and consensus agnostic. We're aiming to make use of IOTA's capabilities. But really almost every network that meets our security and energy standards can be plugged into Jeshka and Quests as well as Resource Streams can leave the core networks and may get transported in ways we can't even imagine right now. For example, a group of people in Cuba has recently resolved a Jeshka Quest only by using their own physical USB flash drive exchange network. Usually, the first Assembly or Assemblies that receive(s) your Quest decide(s) where and how to transport it from there.

3.11 Meta Conecpts

On a meta layer we can observe many concepts that are emerging from the network as it was theorized by the Jeshka Collective in the original Jeshka Whitepaper.

3.11.1 Proof-of-Reputation



2 Proof-of-Reputation

2.1 Reputation

The Proof-of-Reputation consensus protocol uses a highest reputation first approach. Reputation is not a single unit of measurement. Transactions are grouped by content and different groups of transactions have different reputation units. The protocol puts all processors on a spectrum of reputation. High reputation processors are assigned with the processes first, high reputation processors then assign lower reputation processors to provide finality and verifiability. The initiator has to vote on a result of another random network, before the transaction is considered successful. Processors form mixed groups, low reputation processors can gain reputation and help ensure the liveness of the network. When a low reputation processors becomes a high reputation processor, it inherently forms a new network with low reputation processors. Reputation is increased by the number of validations from higher level reputation processors within the network. Because JIoT doesnt use sequential blocks, this can take place in parallel. Processors can only earn reputation from high reputation processors in the groups they join. Reputation is awarded based on speed and validity of the transaction, as well as the quality of the distribution of the transaction to lower level processors.

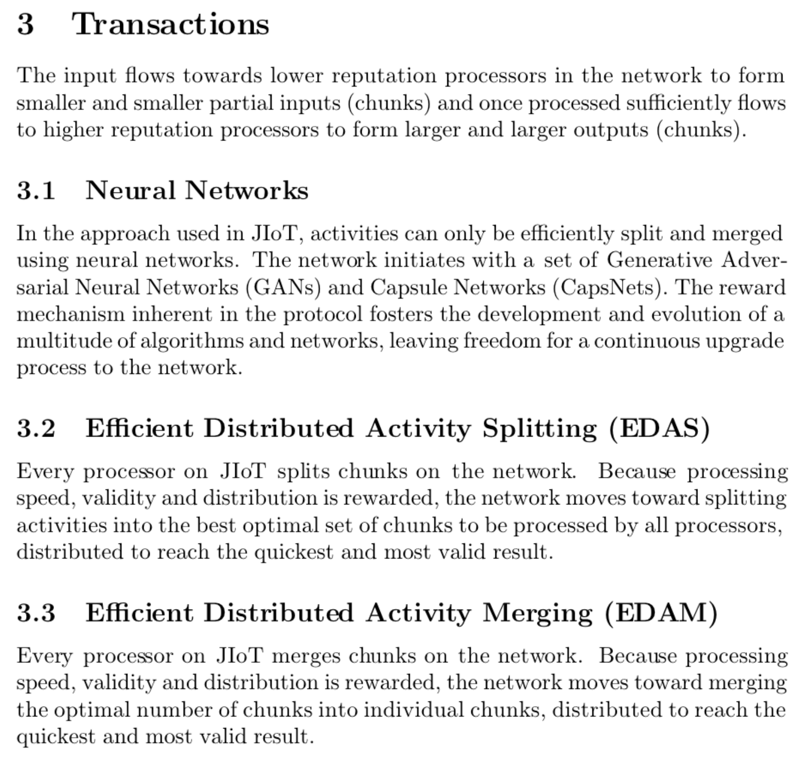
2.2 Additional proof and loss of reputation

Critical transactions can be sent to multiple random networks of processors. All processors in a network that result with an output that differs from the consensus result lose reputation. This way it is ensured that the critical transaction is processed correctly and faulty or malicious networks are contra-rewarded.

2.3 Scalability

With a growing number of transactions, there is a growing number of processors to speed up the entire network. The amount of work done to verify a transaction is not exponentially growing, but constantly optimized.

3.11.2 Transactions - NN, EDAS & EDAM



3 Transactions

The input flows towards lower reputation processors in the network to form smaller and smaller partial inputs (chunks) and once processed sufficiently flows to higher reputation processors to form larger and larger outputs (chunks).

3.1 Neural Networks

In the approach used in JIoT, activities can only be efficiently split and merged using neural networks. The network initiates with a set of Generative Adversarial Neural Networks (GANs) and Capsule Networks (CapsNets). The reward mechanism inherent in the protocol fosters the development and evolution of a multitude of algorithms and networks, leaving freedom for a continuous upgrade process to the network.

3.2 Efficient Distributed Activity Splitting (EDAS)

Every processor on JIoT splits chunks on the network. Because processing speed, validity and distribution is rewarded, the network moves toward splitting activities into the best optimal set of chunks to be processed by all processors, distributed to reach the quickest and most valid result.

3.3 Efficient Distributed Activity Merging (EDAM)

Every processor on JIoT merges chunks on the network. Because processing speed, validity and distribution is rewarded, the network moves toward merging the optimal number of chunks into individual chunks, distributed to reach the quickest and most valid result.

On the local Jeshka Network it has to answer to: req getchannel(peer)historyastxfile for channels in its domain req getlatestwhistleforpeerinchannel for channels in its domain chn open channel with peers from its domain chn open channel with peers from different domains

A representative also **must** forward transactions to the next level up (and sign them) if the receiver is on a different domain. –>

CHAPTER 4

License

GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Copyright (C) 2007 Free Software Foundation, Inc. https://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

1. 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.

1. 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.

1. 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.

1. 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.

1. 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

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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 PAR-

TIES 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.

1. 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.

1. 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)

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 https://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 https://www.gnu.org/licenses/.

CHAPTER 5

Legal Notice

All contents are a work of fiction. Names, characters, businesses, places, events, locales, and incidents are either the products of the authors' imagination or used in a fictitious manner. Any resemblance to actual persons, living or dead, or actual events is purely coincidental.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search